

RX610 群

用户手册 硬件篇

瑞萨 32 位单片机
RX 族 / RX610 系列

本资料所记载的内容，均为本资料发行时的信息，瑞萨电子对于本资料所记载的产品或者规格可能会作改动，恕不另行通知。
请通过瑞萨电子的主页确认发布的最新信息。

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.

产品使用时的注意事项

本文对适用于单片机所有产品的“使用时的注意事项”进行说明。有关个别的使用时的注意事项请参照正文。此外，如果在记载上有与本手册的正文有差异之处，请以正文为准。

1. 未使用的引脚的处理

【注意】将未使用的引脚按照正文的“未使用引脚的处理”进行处理。

CMOS产品的输入引脚的阻抗一般为高阻抗。如果在开路的状态下运行未使用的引脚，由于感应现象，外加LSI周围的噪声，在LSI内部产生穿透电流，有可能被误认为是输入信号而引起误动作。未使用的引脚，请按照正文的“未使用引脚的处理”中的指示进行处理。

2. 通电时的处理

【注意】通电时产品处于不定状态。

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

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

3. 禁止存取保留地址（保留区）

【注意】禁止存取保留地址（保留区）

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

4. 关于时钟

【注意】复位时，请在时钟稳定后解除复位。

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

5. 关于产品间的差异

【注意】在变更不同型号的产品时，请对每一个产品型号进行系统评价测试。

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

本手册的使用方法

1 目的和对象

本手册是一本帮助用户理解本单片机的硬件功能和电特性的手册。它以使用本手册来设计应用系统的用户为对象。在使用本手册时，需要具备电路、逻辑电路以及单片机的基础知识。

本手册由产品概要、CPU、系统控制功能、外围功能、电特性、使用时的注意事项几大部分组成。

必须在充分确认注意事项后使用本单片机。注意事项记录在各章的正文中、各章的最后和注意事项章节中。

修订记录归纳了对旧版本记载内容的更正或追加的主要位置。并不是修订内容的全部记载。详情请确认本手册的正文。

RX610 群准备了以下的文献。请使用最新的文献。最新版本刊登在瑞萨电子的主页上。

| 文献的种类 | 记载内容 | 资料名 | 资料号 |
|--------------------------|---------------------------------------|---------------------------|------------|
| Shortsheet | 硬件的概要 | RX610 Group Shortsheet | — |
| 数据表 | 硬件的概要和电特性 | — | — |
| 硬件手册 | 硬件的说明（引脚配置、存储器映像、外围功能的说明、电特性、时序）和工作说明 | RX610 群用户手册 硬件篇 | 本用户手册 |
| 软件手册 | CPU 指令集的说明 | RX Family Software Manual | REJ09B0435 |
| 应用注意事项 | 应用例子参考程序 | — | — |
| RENESAS TECHNICAL UPDATE | 相关产品说明、文献等的快速公告 | — | — |

2 寄存器的表示

在各章的“寄存器说明”中，有位的排列图和功能表，说明位的设定内容。以下说明所使用的符号和术语。

X.X.X 寄存器

地址 xxxx xxxxxh

| | | | | | | | |
|----|---------|---------|---------|----|----|----|---------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | | | | — | — | — | |

复位后的值 x 0 0 0 0 0 0 0

| 位 | 符号 | 位名 | 功能 | R/W ^(注1) |
|-------|---------------|-------------------|---|---------------------|
| b0 | 0 | 位 (注2) | 0: 1: 不能设定。 (注3) | R/W |
| b3-b1 | — | 保留位 | 读写值都为“0”。 | R/W |
| b4 | 4 | 位 | 0: 1: | R |
| b6-b5 | [1:0] | 位 | 0 0: 0 1: 上述以外: 不能设定。 (注3) | R(W) (注1) |
| b7 | — | 保留位 | 读取值为不定值, 写无效。 | R |

- 注1. R/W : 可读写。
R(W) : 可读写, 但是有写数据的限制。
限制内容请参照各寄存器的说明和注释。
R : 可写, 写数据无效。
- 注2. 保留位。只能写指定值, 如果写的值不是指定的值, 就不保证运行。
- 注3. 不能设定。如果设定, 就不保证运行。

3 省略语及简称的说明

| 省略 / 简称 | 全称 | 备注 |
|---------|--|-------------------|
| ACIA | Asynchronous Communication Interface Adapter | 异步通信接口适配器 |
| bps | bits per second | 位 / 秒; 每秒传送位数 |
| 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 | DMA 控制器 |
| GSM | Global System for Mobile Communications | 全球数字移动电话系统 |
| Hi-Z | High Impedance | 高阻抗 |
| IEBus | Inter Equipment bus | - |
| I/O | Input/Output | 输入 / 输出 |
| IrDA | Infrared Data Association | 红外线数据协会 |
| 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 |
| PWM | Pulse Width Modulation | 脉宽调制 |
| SFR | Special Function Registers | 特殊功能寄存器 |
| SIM | Subscriber Identity Module | ISO-7816 规定的 IC 卡 |
| UART | Universal Asynchronous Receiver/Transmitter | 通用异步收发器 |
| VCO | Voltage Controlled Oscillator | 电压控制振荡器 |

所有商标及注册商标分别归属于其所有者。

目 录

| | |
|----------------------------------|----|
| 1. 概要 | 1 |
| 1.1 特点 | 1 |
| 1.1.1 用途 | 1 |
| 1.1.2 规格概要 | 2 |
| 1.2 产品一览表 | 5 |
| 1.3 框图 | 6 |
| 1.4 引脚排列图 | 7 |
| 1.5 引脚功能 | 21 |
| 2. CPU | 25 |
| 2.1 特点 | 25 |
| 2.2 CPU 寄存器组 | 26 |
| 2.2.1 通用寄存器 (R0 ~ R15) | 27 |
| 2.2.2 控制寄存器 | 27 |
| 2.3 处理器模式 | 33 |
| 2.3.1 管理模式 | 33 |
| 2.3.2 用户模式 | 33 |
| 2.3.3 特权指令 | 33 |
| 2.3.4 处理器模式之间的转移 | 33 |
| 2.4 数据类型 | 34 |
| 2.4.1 整数 | 34 |
| 2.4.2 浮点数 | 34 |
| 2.4.3 位 | 35 |
| 2.4.4 字符串 | 35 |
| 2.5 字节序 | 36 |
| 2.5.1 字节序的转换 | 36 |
| 2.5.2 I/O 寄存器的存取 | 39 |
| 2.5.3 存取 I/O 寄存器时的注意事项 | 39 |
| 2.5.4 数据排列 | 40 |
| 2.6 向量表 | 41 |
| 2.6.1 固定向量表 | 41 |
| 2.6.2 可向量表 | 41 |
| 2.7 指令操作 | 42 |
| 2.7.1 RMPA 指令和字符串操作指令的数据预取 | 42 |
| 2.8 流水线 | 43 |
| 2.8.1 概要 | 43 |
| 2.8.2 指令和流水线处理 | 44 |
| 2.8.3 指令处理时间的计算方法 | 51 |
| 2.8.4 中断响应周期数 | 51 |
| 3. 运行模式 | 52 |
| 3.1 运行模式的种类和选择 | 52 |
| 3.2 寄存器说明 | 53 |
| 3.2.1 模式监视寄存器 (MDMONR) | 53 |
| 3.2.2 模式状态寄存器 (MDSR) | 54 |
| 3.2.3 系统控制寄存器 0 (SYSCR0) | 54 |
| 3.2.4 系统控制寄存器 1 (SYSCR1) | 55 |

| | | |
|-------|---------------------------|-----|
| 3.3 | 运行模式的说明 | 56 |
| 3.3.1 | 单芯片模式 | 56 |
| 3.3.2 | 内部 ROM 有效扩展模式 | 56 |
| 3.3.3 | 内部 ROM 无效扩展模式 | 56 |
| 3.3.4 | 引导模式 | 56 |
| 3.3.5 | 用户引导模式 | 56 |
| 3.4 | 运行模式的转移 | 57 |
| 3.4.1 | 通过模式引脚进行运行模式的转移 | 57 |
| 3.4.2 | 通过寄存器进行运行模式的转移 | 58 |
| 4. | 地址空间 | 59 |
| 4.1 | 地址空间 | 59 |
| 4.2 | 外部地址空间 | 63 |
| 5. | I/O 寄存器 | 64 |
| 5.1 | I/O 寄存器地址一览表（地址顺序） | 66 |
| 5.2 | I/O 寄存器位一览表 | 86 |
| 6. | 复位 | 112 |
| 6.1 | 概要 | 112 |
| 6.2 | 寄存器说明 | 113 |
| 6.2.1 | 复位状态寄存器（RSTSR） | 113 |
| 6.2.2 | 复位的控制/状态寄存器（RSTCSR） | 114 |
| 6.3 | 运行说明 | 115 |
| 6.3.1 | 引脚复位 | 115 |
| 6.3.2 | 深度软件待机复位 | 115 |
| 6.3.3 | 看门狗定时器复位 | 115 |
| 6.4 | 复位发生源的判断 | 115 |
| 6.5 | 使用时的注意事项 | 115 |
| 6.5.1 | 设计电路板时的注意事项 | 115 |
| 7. | 时钟发生电路 | 116 |
| 7.1 | 概要 | 116 |
| 7.2 | 寄存器说明 | 117 |
| 7.2.1 | 系统时钟控制寄存器（SCKCR） | 118 |
| 7.3 | 主时钟振荡器 | 120 |
| 7.3.1 | 连接晶体谐振器的方法 | 120 |
| 7.3.2 | 输入外部时钟的方法 | 121 |
| 7.4 | PLL 电路 | 121 |
| 7.5 | 分频器 | 121 |
| 7.6 | 内部时钟 | 121 |
| 7.6.1 | 系统时钟（ICLK） | 122 |
| 7.6.2 | 外围模块时钟（PCLK） | 122 |
| 7.6.3 | 外部总线时钟（BCLK） | 122 |
| 7.7 | 使用时的注意事项 | 122 |
| 7.7.1 | 有关时钟发生电路的注意事项 | 122 |
| 7.7.2 | 有关振荡器的注意事项 | 123 |
| 7.7.3 | 设计电路板时的注意事项 | 123 |

| | | |
|--------|---------------------------------------|-----|
| 8. | 低功耗功能 | 124 |
| 8.1 | 概要 | 124 |
| 8.2 | 寄存器说明 | 127 |
| 8.2.1 | 待机控制寄存器 (SBYCR) | 128 |
| 8.2.2 | 模块停止控制寄存器 A (MSTPCRA) | 129 |
| 8.2.3 | 模块停止控制寄存器 B (MSTPCRB) | 131 |
| 8.2.4 | 模块停止控制寄存器 C (MSTPCRC) | 132 |
| 8.2.5 | 深度待机控制寄存器 (DPSBYCR) | 133 |
| 8.2.6 | 深度待机等待控制寄存器 (DPSWCR) | 134 |
| 8.2.7 | 深度待机中断允许寄存器 (DPSIER) | 135 |
| 8.2.8 | 深度待机中断标志寄存器 (DPSIFR) | 136 |
| 8.2.9 | 深度待机中断边沿寄存器 (DPSIEGR) | 137 |
| 8.2.10 | 复位状态寄存器 (RSTSR) | 138 |
| 8.2.11 | 深度待机备用寄存器 (DPSBKRY) (y=0 ~ 31) | 138 |
| 8.3 | 多时钟功能 | 139 |
| 8.4 | 模块停止功能 | 139 |
| 8.5 | 低功耗状态 | 139 |
| 8.5.1 | 睡眠模式 | 139 |
| 8.5.2 | 全模块时钟停止模式 | 140 |
| 8.5.3 | 软件待机模式 | 141 |
| 8.5.4 | 深度软件待机模式 | 144 |
| 8.6 | BCLK 输出的控制 | 148 |
| 8.7 | 使用时的注意事项 | 148 |
| 8.7.1 | I/O 端口状态 | 148 |
| 8.7.2 | DMAC 和 DTC 的模块停止 | 148 |
| 8.7.3 | 内部外围模块的中断 | 148 |
| 8.7.4 | MSTPCRA、MSTPCRB、MSTPCRC 寄存器的写操作 | 148 |
| 8.7.5 | 通过 DIRQnE 位 (n=0 ~ 3) 控制输入缓冲器 | 148 |
| 8.7.6 | 深度软件待机模式的转移和中断的竞争 | 149 |
| 8.7.7 | WAIT 指令的执行时序 | 149 |
| 9. | 异常处理 | 150 |
| 9.1 | 异常事件 | 150 |
| 9.1.1 | 未定义指令异常 | 150 |
| 9.1.2 | 特权指令异常 | 150 |
| 9.1.3 | 浮点异常 | 150 |
| 9.1.4 | 复位 | 150 |
| 9.1.5 | 非屏蔽中断 | 150 |
| 9.1.6 | 中断 | 151 |
| 9.1.7 | 无条件陷阱 | 151 |
| 9.2 | 异常处理步骤 | 151 |
| 9.3 | 异常事件的接受 | 152 |
| 9.3.1 | 接受时序和被保存的 PC 值 | 152 |
| 9.3.2 | 向量和 PC、PSW 的保存场所 | 153 |
| 9.4 | 接受异常 / 从异常返回时的硬件处理 | 153 |
| 9.5 | 硬件预处理 | 154 |
| 9.5.1 | 未定义指令异常 | 154 |
| 9.5.2 | 特权指令异常 | 154 |
| 9.5.3 | 浮点异常 | 154 |
| 9.5.4 | 复位 | 154 |
| 9.5.5 | 非屏蔽中断 | 155 |
| 9.5.6 | 中断 | 155 |

| | | |
|---------|--|-----|
| 9.5.7 | 无条件陷阱 | 155 |
| 9.6 | 从异常处理程序的返回 | 155 |
| 9.7 | 异常事件的优先级 | 156 |
| 10. | 中断控制器 (ICU) | 157 |
| 10.1 | 概要 | 157 |
| 10.2 | 寄存器说明 | 159 |
| 10.2.1 | 中断请求寄存器 i (IRi) (i= 中断向量号) | 167 |
| 10.2.2 | 中断请求目标设定寄存器 i (ISELRi) (i= 中断向量号) | 168 |
| 10.2.3 | 中断请求允许寄存器 i (IERi) (i=02h ~ 1Fh) | 169 |
| 10.2.4 | 中断源优先级寄存器 i (IPRi) (i=00h~8Fh) | 170 |
| 10.2.5 | 高速中断设定寄存器 (FIR) | 171 |
| 10.2.6 | IRQ 检测允许寄存器 n (IRQERn) (n=0 ~ 15) | 172 |
| 10.2.7 | IRQ 控制寄存器 n (IRQCRn) (n=0 ~ 15) | 172 |
| 10.2.8 | 非屏蔽中断允许寄存器 (NMIER) | 173 |
| 10.2.9 | NMI 引脚中断控制寄存器 (NMICR) | 173 |
| 10.2.10 | 非屏蔽中断状态寄存器 (NMISR) | 174 |
| 10.2.11 | 非屏蔽中断清除寄存器 (NMICLR) | 174 |
| 10.2.12 | 软件待机解除 IRQ 允许寄存器 (SSIER) | 175 |
| 10.3 | 向量表 | 176 |
| 10.3.1 | 中断向量表 | 176 |
| 10.3.2 | 高速中断的向量地址 | 181 |
| 10.3.3 | 非屏蔽中断的向量地址 | 181 |
| 10.4 | 运行说明 | 182 |
| 10.4.1 | 中断的允许 / 禁止 | 182 |
| 10.4.2 | 中断状态标志 | 182 |
| 10.4.3 | 中断请求目标的选择 | 184 |
| 10.4.4 | 优先级的判断 | 186 |
| 10.4.5 | 高速中断 | 186 |
| 10.4.6 | 外部中断 | 186 |
| 10.5 | 非屏蔽中断 | 187 |
| 10.6 | 从低功耗状态的返回 | 188 |
| 10.6.1 | 从睡眠模式和全模块时钟停止模式的返回 | 188 |
| 10.6.2 | 从软件待机模式的返回 | 188 |
| 10.7 | 使用注意事项 | 189 |
| 10.7.1 | 写中断控制器的寄存器时的注意事项 | 189 |
| 10.7.2 | 使用 NMI 引脚中断时的 WAIT 指令的注意事项 | 189 |
| 11. | 总线 | 190 |
| 11.1 | 概要 | 190 |
| 11.2 | 总线说明 | 191 |
| 11.2.1 | CPU 总线 | 191 |
| 11.2.2 | 内部主总线 | 191 |
| 11.2.3 | 内部外围总线 | 191 |
| 11.2.4 | 外部总线 | 192 |
| 11.2.5 | 并行运行 | 194 |
| 11.3 | 寄存器说明 | 195 |
| 11.3.1 | CSi 控制寄存器 (CSiCNT) (i=0 ~ 7) | 196 |
| 11.3.2 | CSi 恢复周期设定寄存器 (CSiREC) (i=0 ~ 7) | 197 |
| 11.3.3 | CSi 模式寄存器 (CSiMOD) (i=0 ~ 7) | 199 |
| 11.3.4 | CSi 等待控制寄存器 1 (CSiWCNT1) (i=0 ~ 7) | 201 |
| 11.3.5 | CSi 等待控制寄存器 2 (CSiWCNT2) (i=0 ~ 7) | 204 |

| | | |
|---------|---------------------------|-----|
| 11.3.6 | 总线错误源清除寄存器 (BERCLR) | 206 |
| 11.3.7 | 总线错误监视允许寄存器 (BEREN) | 207 |
| 11.3.8 | 总线错误中断允许寄存器 (BERIE) | 207 |
| 11.4 | 字节序和数据对齐 | 208 |
| 11.4.1 | 16 位总线空间 | 208 |
| 11.4.2 | 8 位总线空间 | 210 |
| 11.5 | 运行说明 | 212 |
| 11.5.1 | 外部总线的存取时序 | 212 |
| 11.5.2 | 外部等待功能 | 220 |
| 11.5.3 | 恢复周期的插入 | 221 |
| 11.5.4 | 写缓冲器功能 | 222 |
| 11.5.5 | 限制事项 | 223 |
| 11.6 | 总线错误监视部 | 224 |
| 11.6.1 | 总线错误的种类 | 224 |
| 11.6.2 | 发生总线错误时的运行 | 224 |
| 11.6.3 | 总线错误的发生条件 | 225 |
| 12. | DMA 控制器 (DMAC) | 226 |
| 12.1 | 概要 | 226 |
| 12.2 | 寄存器说明 | 227 |
| 12.2.1 | DMA 模式寄存器 (DMMOD) | 229 |
| 12.2.2 | DMA 控制寄存器 A (DMCRA) | 231 |
| 12.2.3 | DMA 控制寄存器 B (DMCRB) | 234 |
| 12.2.4 | DMA 控制寄存器 C (DMCRC) | 234 |
| 12.2.5 | DMA 控制寄存器 D (DMCRD) | 235 |
| 12.2.6 | DMA 控制寄存器 E (DMCRE) | 236 |
| 12.2.7 | DMA 当前传送源的地址寄存器 (DMCSA) | 237 |
| 12.2.8 | DMA 当前传送目标的地址寄存器 (DMCDA) | 238 |
| 12.2.9 | DMA 当前传送字节的计数寄存器 (DMCBC) | 239 |
| 12.2.10 | DMA 重加载传送源的地址寄存器 (DMRSA) | 240 |
| 12.2.11 | DMA 重加载传送目标的地址寄存器 (DMRDA) | 240 |
| 12.2.12 | DMA 重加载传送字节的计数寄存器 (DMRBC) | 241 |
| 12.2.13 | DMA 中断控制寄存器 (DMICNT) | 241 |
| 12.2.14 | DMA 启动控制寄存器 (DMSCNT) | 242 |
| 12.2.15 | DMA 仲裁状态寄存器 (DMASTS) | 243 |
| 12.2.16 | DMA 传送结束的检测寄存器 (DMEDET) | 244 |
| 12.3 | 运行说明 | 245 |
| 12.3.1 | 总线权的释放时序 | 245 |
| 12.3.2 | 传送方式 | 245 |
| 12.3.3 | DMAC 的启动 | 247 |
| 12.3.4 | DMA 传送的开始 | 248 |
| 12.3.5 | DMA 传送的结束 | 248 |
| 12.3.6 | DMA 传送的暂停、重新开始和中止 | 248 |
| 12.3.7 | DMA 启动源 | 249 |
| 12.3.8 | 通道仲裁 | 249 |
| 12.3.9 | 重加载功能 | 250 |
| 12.3.10 | 循环 | 251 |
| 12.4 | 中断 | 251 |
| 12.5 | 低功耗功能 | 252 |
| 12.6 | 使用时的注意事项 | 253 |
| 12.6.1 | 寄存器的设定 | 253 |
| 12.6.2 | 使用外部设备的情况 | 253 |

| | |
|-------------------------------|-----|
| 13. 数据传送控制器 (DTC) | 254 |
| 13.1 概要 | 254 |
| 13.2 寄存器说明 | 256 |
| 13.2.1 DTC 模式寄存器 A (MRA) | 257 |
| 13.2.2 DTC 模式寄存器 B (MRB) | 258 |
| 13.2.3 DTC 源地址寄存器 (SAR) | 259 |
| 13.2.4 DTC 目标地址寄存器 (DAR) | 259 |
| 13.2.5 DTC 传送计数寄存器 A (CRA) | 260 |
| 13.2.6 DTC 传送计数寄存器 B (CRB) | 261 |
| 13.2.7 DTC 控制寄存器 (DTCCR) | 262 |
| 13.2.8 DTC 向量基址寄存器 (DTCVBR) | 263 |
| 13.2.9 DTC 地址模式寄存器 (DTCADM0D) | 263 |
| 13.2.10 DTC 模块启动寄存器 (DTCST) | 264 |
| 13.3 启动源 | 265 |
| 13.3.1 传送信息的分配和 DTC 向量表 | 265 |
| 13.3.2 启动源和向量地址 | 267 |
| 13.4 运行说明 | 269 |
| 13.4.1 传送信息的跳读功能 | 272 |
| 13.4.2 传送信息的回写省略功能 | 273 |
| 13.4.3 正常传送模式 | 274 |
| 13.4.4 重复传送模式 | 275 |
| 13.4.5 块传送模式 | 276 |
| 13.4.6 链传送 | 277 |
| 13.4.7 运行时序 | 278 |
| 13.4.8 DTC 的执行状态 | 280 |
| 13.4.9 DTC 的总线权释放时序 | 280 |
| 13.5 DTC 的设定步骤 | 281 |
| 13.6 DTC 的使用例子 | 282 |
| 13.6.1 正常传送 | 282 |
| 13.6.2 链传送 | 282 |
| 13.6.3 计数器为“0”时的链传送 | 283 |
| 13.7 中断源 | 284 |
| 13.8 低功耗功能 | 284 |
| 13.8.1 DTC 模块启动寄存器的设定 | 284 |
| 13.9 使用时的注意事项 | 285 |
| 13.9.1 传送信息起始地址、传送源地址和传送目标地址 | 285 |
| 13.9.2 传送信息的分配 | 285 |
| 14. I/O 端口 | 286 |
| 14.1 概要 | 286 |
| 14.2 寄存器说明 | 291 |
| 14.2.1 数据方向寄存器 (DDR) | 294 |
| 14.2.2 数据寄存器 (DR) | 295 |
| 14.2.3 端口寄存器 (PORT) | 296 |
| 14.2.4 输入缓冲控制寄存器 (ICR) | 297 |
| 14.2.5 上拉电阻控制寄存器 (PCR) | 298 |
| 14.2.6 漏极开路控制寄存器 (ODR) | 299 |
| 14.2.7 端口功能控制寄存器 0 (PFCR0) | 299 |
| 14.2.8 端口功能控制寄存器 1 (PFCR1) | 300 |
| 14.2.9 端口功能控制寄存器 2 (PFCR2) | 302 |
| 14.2.10 端口功能控制寄存器 3 (PFCR3) | 303 |
| 14.2.11 端口功能控制寄存器 4 (PFCR4) | 304 |

| | | |
|---------|--|-----|
| 14.2.12 | 端口功能控制寄存器 5 (PFCR5) | 305 |
| 14.2.13 | 端口功能控制寄存器 6 (PFCR6) | 306 |
| 14.2.14 | 端口功能控制寄存器 7 (PFCR7) | 309 |
| 14.2.15 | 端口功能控制寄存器 8 (PFCR8) | 312 |
| 14.2.16 | 端口功能控制寄存器 9 (PFCR9) | 313 |
| 14.3 | 端口的设定 | 314 |
| 14.3.1 | 端口 0 (P0) | 314 |
| 14.3.2 | 端口 1 (P1) | 315 |
| 14.3.3 | 端口 2 (P2) | 317 |
| 14.3.4 | 端口 3 (P3) | 320 |
| 14.3.5 | 端口 4 (P4) | 322 |
| 14.3.6 | 端口 5 (P5) | 323 |
| 14.3.7 | 端口 6 (P6) | 325 |
| 14.3.8 | 端口 7 (P7) | 328 |
| 14.3.9 | 端口 8 (P8) | 329 |
| 14.3.10 | 端口 9 (P9) | 331 |
| 14.3.11 | 端口 A (PA) | 333 |
| 14.3.12 | 端口 B (PB) | 335 |
| 14.3.13 | 端口 C (PC) | 338 |
| 14.3.14 | 端口 D (PD) | 341 |
| 14.3.15 | 端口 E (PE) | 341 |
| 14.3.16 | 端口 F (PF) | 341 |
| 14.3.17 | 端口 G (PG) | 343 |
| 14.3.18 | 端口 H (PH) | 345 |
| 14.4 | 输出信号有效的设定一览表 | 347 |
| 14.5 | 未使用引脚的处理 | 353 |
| 14.6 | 使用时的注意事项 | 354 |
| 14.6.1 | 输入缓冲控制寄存器 (Pm.ICR) 的设定 | 354 |
| 14.6.2 | 端口功能控制寄存器 (PFCRm) 的设定 | 354 |
| 14.6.3 | 使用 A/D 转换器输入时的端口设定 | 354 |
| 15. | 16 位定时器脉冲单元 (TPU) | 355 |
| 15.1 | 概要 | 355 |
| 15.2 | 寄存器说明 | 363 |
| 15.2.1 | 定时器的控制寄存器 (TCR) | 366 |
| 15.2.2 | 定时器的模式寄存器 (TMDR) | 370 |
| 15.2.3 | 定时器的 I/O 控制寄存器 (TIORH、TIORL、TIOR) | 372 |
| 15.2.4 | 定时器的中断允许寄存器 (TIER) | 381 |
| 15.2.5 | 定时器的状态寄存器 (TSR) | 382 |
| 15.2.6 | 定时器的计数器 (TCNT) | 383 |
| 15.2.7 | 定时器的通用寄存器 A (TGRA) 定时器的通用寄存器 B (TGRB) 定时器的通用寄存器 C (TGRC) 定时器的通用寄存器 D (TGRD) | 383 |
| 15.2.8 | 定时器的启动寄存器 (TSTRA、TSTRB) | 384 |
| 15.2.9 | 定时器的同步寄存器 (TSYRA、TSYRB) | 385 |
| 15.3 | 运行说明 | 386 |
| 15.3.1 | 概要 | 386 |
| 15.3.2 | 同步运行 | 391 |
| 15.3.3 | 缓冲运行 | 393 |
| 15.3.4 | 级联运行 | 396 |
| 15.3.5 | PWM 模式 | 398 |

| | | |
|---------|---|-----|
| 15.3.6 | 相位计数模式 | 403 |
| 15.4 | 中断源 | 409 |
| 15.5 | DTC 的启动 | 411 |
| 15.6 | DMAC 的启动 | 411 |
| 15.7 | A/D 转换器的启动 | 411 |
| 15.8 | 运行时序 | 412 |
| 15.8.1 | 输入 / 输出时序 | 412 |
| 15.8.2 | 中断信号的时序 | 416 |
| 15.9 | 使用时的注意事项 | 418 |
| 15.9.1 | 模块停止功能的设定 | 418 |
| 15.9.2 | 输入时钟的限制事项 | 418 |
| 15.9.3 | 设定周期时的注意事项 | 418 |
| 15.9.4 | TPUm.TCNT 计数器的写和清除的竞争 | 419 |
| 15.9.5 | TPUm.TCNT 计数器的写和递增计数的竞争 | 419 |
| 15.9.6 | TPUm.TGRy 寄存器的写和比较匹配的竞争 | 420 |
| 15.9.7 | 缓冲寄存器的写和比较匹配的竞争 | 420 |
| 15.9.8 | TPUm.TGRy 寄存器的读和输入捕捉的竞争 | 421 |
| 15.9.9 | TPUm.TGRy 寄存器的写和输入捕捉的竞争 | 421 |
| 15.9.10 | 缓冲寄存器的写和输入捕捉的竞争 | 422 |
| 15.9.11 | 上溢 / 下溢和计数器清除的竞争 | 422 |
| 15.9.12 | TPUm.TCNT 计数器的写和上溢 / 下溢的竞争 | 423 |
| 15.9.13 | 输入 / 输出引脚的兼用 | 423 |
| 16. | 可编程脉冲发生器 (PPG) | 424 |
| 16.1 | 概要 | 424 |
| 16.2 | 寄存器说明 | 428 |
| 16.2.1 | PPG 触发选择寄存器 (PTRSLR) | 429 |
| 16.2.2 | 下一个数据允许寄存器 H (NDERH)、下一个数据允许寄存器 L (NDERL) | 430 |
| 16.2.3 | 输出数据寄存器 H (PODRH)、输出数据寄存器 L (PODRL) | 433 |
| 16.2.4 | 下一个数据寄存器 H (NDRH)、下一个数据寄存器 L (NDRL) | 435 |
| 16.2.5 | PPG 输出控制寄存器 (PCR) | 439 |
| 16.2.6 | PPG 输出模式寄存器 (PMR) | 441 |
| 16.3 | 运行说明 | 443 |
| 16.3.1 | 输出时序 | 444 |
| 16.3.2 | 脉冲输出正常运行的设定步骤例子 | 445 |
| 16.3.3 | 脉冲输出正常运行的例子 (5 相脉冲输出例子) | 447 |
| 16.3.4 | 脉冲输出非重叠运行 | 448 |
| 16.3.5 | 脉冲输出非重叠运行的设定步骤例子 | 449 |
| 16.3.6 | 脉冲输出非重叠运行的例子 (4 相互补的非重叠输出例子) | 451 |
| 16.3.7 | 脉冲的反相输出 | 452 |
| 16.3.8 | 通过输入捕捉进行的脉冲输出 | 453 |
| 16.4 | 使用时的注意事项 | 453 |
| 16.4.1 | 模块停止功能的设定 | 453 |
| 17. | 8 位定时器 (TMR) | 454 |
| 17.1 | 概要 | 454 |
| 17.2 | 寄存器说明 | 458 |
| 17.2.1 | 定时器的计数器 (TCNT) | 459 |
| 17.2.2 | 时间常数寄存器 A (TCORA) | 459 |
| 17.2.3 | 时间常数寄存器 B (TCORB) | 460 |
| 17.2.4 | 定时器的控制寄存器 (TCR) | 461 |
| 17.2.5 | 定时器的计数器控制寄存器 (TCCR) | 462 |

| | | |
|--------|-------------------------------|-----|
| 17.2.6 | 定时器的控制 / 状态寄存器 (TCSR) | 464 |
| 17.3 | 运行说明 | 466 |
| 17.3.1 | 脉冲输出 | 466 |
| 17.3.2 | 复位输入 | 466 |
| 17.4 | 运行时序 | 467 |
| 17.4.1 | TCNT 计数器的计数时序 | 467 |
| 17.4.2 | 比较匹配时的中断标志变为“1”的时序 | 468 |
| 17.4.3 | 比较匹配时的定时器输出时序 | 468 |
| 17.4.4 | 通过比较匹配进行的计数器清除时序 | 469 |
| 17.4.5 | TCNT 计数器的外部复位时序 | 469 |
| 17.4.6 | 通过上溢信号使中断标志变为“1”的时序 | 470 |
| 17.5 | 级联时的运行 | 470 |
| 17.5.1 | 16 位计数模式 | 470 |
| 17.5.2 | 比较匹配计数模式 | 470 |
| 17.6 | 中断源 | 471 |
| 17.6.1 | 中断源和 DTC 启动 | 471 |
| 17.6.2 | A/D 转换器的启动 | 471 |
| 17.7 | 使用时的注意事项 | 472 |
| 17.7.1 | 模块停止功能的设定 | 472 |
| 17.7.2 | 设定周期时的注意事项 | 472 |
| 17.7.3 | TCNT 计数器的写和计数器清除的竞争 | 472 |
| 17.7.4 | TCNT 计数器的写和递增计数的竞争 | 472 |
| 17.7.5 | TCORA 寄存器、TCORB 寄存器的写和比较匹配的竞争 | 473 |
| 17.7.6 | 比较匹配 A 和比较匹配 B 的竞争 | 473 |
| 17.7.7 | 内部时钟的转换和 TCNT 计数器的运行 | 474 |
| 17.7.8 | 级联时的时钟源设定 | 475 |
| 18. | 比较匹配定时器 (CMT) | 476 |
| 18.1 | 概要 | 476 |
| 18.2 | 寄存器说明 | 477 |
| 18.2.1 | 比较匹配定时器的启动寄存器 0 (CMSTR0) | 477 |
| 18.2.2 | 比较匹配定时器的启动寄存器 1 (CMSTR1) | 478 |
| 18.2.3 | 比较匹配定时器的控制寄存器 (CMCR) | 478 |
| 18.2.4 | 比较匹配定时器的计数器 (CMCNT) | 479 |
| 18.2.5 | 比较匹配定时器的常数寄存器 (CMCOR) | 479 |
| 18.3 | 运行说明 | 480 |
| 18.3.1 | 周期计数 | 480 |
| 18.3.2 | CMCNT 计数器的计数时序 | 480 |
| 18.4 | 中断 | 481 |
| 18.4.1 | 中断源 | 481 |
| 18.4.2 | 比较匹配中断的发生时序 | 481 |
| 18.5 | 使用时的注意事项 | 482 |
| 18.5.1 | 模块停止功能的设定 | 482 |
| 18.5.2 | CMCNT 计数器的写和比较匹配的竞争 | 482 |
| 18.5.3 | CMCNT 计数器的写和递增计数的竞争 | 482 |
| 19. | 看门狗定时器 (WDT) | 483 |
| 19.1 | 概要 | 483 |
| 19.2 | 寄存器说明 | 485 |
| 19.2.1 | 定时器的计数器 (TCNT) | 485 |
| 19.2.2 | 定时器的控制 / 状态寄存器 (TCSR) | 486 |
| 19.2.3 | 复位的控制 / 状态寄存器 (RSTCSR) | 487 |

| | | |
|---------|---------------------------------|-----|
| 19.2.4 | 写窗口 A 寄存器 (WINA) | 488 |
| 19.2.5 | 写窗口 B 寄存器 (WINB) | 488 |
| 19.3 | 运行说明 | 489 |
| 19.3.1 | 看门狗定时器模式 | 489 |
| 19.3.2 | 间隔定时器模式 | 490 |
| 19.4 | 中断源 | 490 |
| 19.5 | 使用时的注意事项 | 491 |
| 19.5.1 | 存取寄存器时的注意事项 | 491 |
| 19.5.2 | 定时器的计数器 (TCNT) 的写和递增计数的竞争 | 492 |
| 19.5.3 | CKS[2:0] 位的改写 | 492 |
| 19.5.4 | 看门狗定时器模式和间隔定时器模式的转换 | 493 |
| 19.5.5 | 看门狗定时器模式的内部复位 | 493 |
| 19.5.6 | WDTOVF# 信号引起的系统复位 | 493 |
| 19.5.7 | 向看门狗定时器模式和软件待机模式的转移 | 493 |
| 20. | 串行通信接口 (SCI) | 494 |
| 20.1 | 概要 | 494 |
| 20.2 | 寄存器说明 | 498 |
| 20.2.1 | 接收移位寄存器 (RSR) | 500 |
| 20.2.2 | 接收数据寄存器 (RDR) | 500 |
| 20.2.3 | 发送数据寄存器 (TDR) | 500 |
| 20.2.4 | 发送移位寄存器 (TSR) | 500 |
| 20.2.5 | 串行模式寄存器 (SMR) | 501 |
| 20.2.6 | 串行控制寄存器 (SCR) | 505 |
| 20.2.7 | 串行状态寄存器 (SSR) | 510 |
| 20.2.8 | 智能卡模式寄存器 (SCMR) | 514 |
| 20.2.9 | 位速率寄存器 (BRR) | 515 |
| 20.2.10 | 串行扩展模式寄存器 (SEMR) | 522 |
| 20.3 | 异步模式的运行 | 524 |
| 20.3.1 | 串行发送 / 接收格式 | 525 |
| 20.3.2 | 异步模式的接收数据采样时序和接收容限 | 526 |
| 20.3.3 | 时钟 | 527 |
| 20.3.4 | SCI 的初始化 (异步模式) | 528 |
| 20.3.5 | 串行数据的发送 (异步模式) | 529 |
| 20.3.6 | 串行数据的接收 (异步模式) | 531 |
| 20.4 | 时钟同步模式的运行 | 534 |
| 20.4.1 | 时钟 | 534 |
| 20.4.2 | SCI 的初始化 (时钟同步模式) | 535 |
| 20.4.3 | 串行数据的发送 (时钟同步模式) | 536 |
| 20.4.4 | 串行数据的接收 (时钟同步模式) | 538 |
| 20.4.5 | 串行数据的全双工通信 (时钟同步模式) | 540 |
| 20.5 | 智能卡接口模式的运行 | 541 |
| 20.5.1 | 连接例子 | 541 |
| 20.5.2 | 数据格式 (块传送模式除外) | 542 |
| 20.5.3 | 块传送模式 | 543 |
| 20.5.4 | 接收数据的采样时序和接收容限 | 543 |
| 20.5.5 | SCI 的初始化 | 544 |
| 20.5.6 | 串行数据的发送 (块传送模式除外) | 545 |
| 20.5.7 | 串行接收 (块传送模式除外) | 547 |
| 20.5.8 | 时钟的输出控制 | 548 |

| | | |
|---------|--------------------------------------|-----|
| 20.6 | 中断源 | 550 |
| 20.6.1 | 串行通信接口模式的中断 | 550 |
| 20.6.2 | 智能卡接口模式的中断 | 551 |
| 20.7 | 使用时的注意事项 | 551 |
| 20.7.1 | 模块停止功能的设定 | 551 |
| 20.7.2 | 有关中止的检测和处理 | 551 |
| 20.7.3 | 标记状态和中止的发送 | 551 |
| 20.7.4 | 有关接收错误标志和发送（只限于时钟同步模式） | 552 |
| 20.7.5 | 有关写 TDR | 552 |
| 20.7.6 | 时钟同步发送时的限制事项 | 552 |
| 20.7.7 | 使用 DMAC 或者 DTC 时的限制事项 | 552 |
| 20.7.8 | 有关低功耗状态时的运行 | 552 |
| 20.7.9 | 时钟同步模式的外部时钟输入 | 554 |
| 21. | CRC 运算器 (CRC) | 555 |
| 21.1 | 概要 | 555 |
| 21.2 | 寄存器说明 | 556 |
| 21.2.1 | CRC 控制寄存器 (CRCCR) | 556 |
| 21.2.2 | CRC 数据输入寄存器 (CRCDIR) | 557 |
| 21.2.3 | CRC 数据输出寄存器 (CRCDOR) | 557 |
| 21.3 | CRC 运算器的运行说明 | 558 |
| 21.4 | 使用时的注意事项 | 561 |
| 21.4.1 | 模块待机功能的设定 | 561 |
| 21.5 | 传送时的注意事项 | 561 |
| 22. | I ² C 总线接口 (RIIC) | 562 |
| 22.1 | 概要 | 562 |
| 22.2 | 寄存器说明 | 565 |
| 22.2.1 | I ² C 总线控制寄存器 1 (ICCR1) | 566 |
| 22.2.2 | I ² C 总线控制寄存器 2 (ICCR2) | 568 |
| 22.2.3 | I ² C 总线模式寄存器 1 (ICMR1) | 571 |
| 22.2.4 | I ² C 总线模式寄存器 2 (ICMR2) | 572 |
| 22.2.5 | I ² C 总线模式寄存器 3 (ICMR3) | 574 |
| 22.2.6 | I ² C 总线功能允许寄存器 (ICFER) | 576 |
| 22.2.7 | I ² C 总线状态允许寄存器 (ICSER) | 578 |
| 22.2.8 | I ² C 总线中断允许寄存器 (ICIER) | 579 |
| 22.2.9 | I ² C 总线状态寄存器 1 (ICSR1) | 581 |
| 22.2.10 | I ² C 总线状态寄存器 2 (ICSR2) | 584 |
| 22.2.11 | 从属地址寄存器 Lm (SARLy) (m=0 ~ 2) | 587 |
| 22.2.12 | 从属地址寄存器 Uy (SARUy) (y=0 ~ 2) | 588 |
| 22.2.13 | I ² C 总线位速率低电平寄存器 (ICBRL) | 589 |
| 22.2.14 | I ² C 总线位速率高电平寄存器 (ICBRH) | 590 |
| 22.2.15 | I ² C 总线发送数据寄存器 (ICDRT) | 592 |
| 22.2.16 | I ² C 总线接收数据寄存器 (ICDRR) | 592 |
| 22.2.17 | I ² C 总线移位寄存器 (ICDRS) | 592 |
| 22.3 | 运行说明 | 593 |
| 22.3.1 | 通信数据的格式 | 593 |
| 22.3.2 | 初始设定 | 594 |
| 22.3.3 | 主控发送 | 595 |
| 22.3.4 | 主控接收 | 599 |
| 22.3.5 | 从属发送 | 602 |
| 22.3.6 | 从属接收 | 605 |

| | | |
|---------|---------------------------------|-----|
| 22.4 | SCL 同步电路 | 607 |
| 22.5 | SDA 输出延迟功能 | 608 |
| 22.6 | 数字噪声滤波器电路 | 609 |
| 22.7 | 地址匹配检测功能 | 610 |
| 22.7.1 | 从属地址匹配检测功能 | 610 |
| 22.7.2 | 全呼地址检测功能 | 612 |
| 22.7.3 | 设备 ID 地址检测功能 | 612 |
| 22.7.4 | 主机地址检测功能 | 614 |
| 22.8 | SCL 的 Low 电平自动保持功能 | 615 |
| 22.8.1 | 发送数据的误发送防止功能 | 615 |
| 22.8.2 | NACK 接收传送中止功能 | 616 |
| 22.8.3 | 接收数据漏取防止功能 | 616 |
| 22.9 | 仲裁失败检测功能 | 618 |
| 22.9.1 | 主控仲裁失败检测功能 (MALE 位) | 618 |
| 22.9.2 | NACK 发送仲裁失败检测功能 (NALE 位) | 620 |
| 22.9.3 | 从属仲裁失败检测功能 (SALE 位) | 621 |
| 22.10 | 开始条件、重新开始条件和停止条件的发行功能 | 622 |
| 22.10.1 | 开始条件的发行 | 622 |
| 22.10.2 | 重新开始条件的发行 | 622 |
| 22.10.3 | 停止条件的发行 | 623 |
| 22.11 | 总线意外停机 | 623 |
| 22.11.1 | 超时检测功能 | 624 |
| 22.11.2 | SCL 时钟追加输出功能 | 625 |
| 22.11.3 | RIIC/ 内部复位 | 625 |
| 22.12 | SMBus 运行 | 626 |
| 22.12.1 | SMBus 超时测量 | 626 |
| 22.12.2 | 数据包错误码 (PEC) | 627 |
| 22.12.3 | SMBus 主机通知协议 /Notify ARP master | 627 |
| 22.13 | 中断源 | 628 |
| 22.14 | 复位状况 | 629 |
| 22.15 | 使用时的注意事项 | 630 |
| 22.15.1 | 模块停止功能的设定 | 630 |
| 22.15.2 | 输入缓冲控制寄存器的设定 | 630 |
| 22.15.3 | 发送应答位的写和输出时序 | 630 |
| 22.15.4 | 主控发送时的停止条件发行请求和发送数据写时序的限制事项 | 630 |
| 22.15.5 | 主控模式中接收到 NACK 时的重新开始通信的注意事项 | 630 |
| 22.15.6 | RDRF 标志的置位时序选择位 (RDRFS 位) 的注意事项 | 630 |
| 23. | A/D 转换器 | 631 |
| 23.1 | 概要 | 631 |
| 23.2 | 寄存器说明 | 638 |
| 23.2.1 | A/D 数据寄存器 y (ADDRy) (y=A ~ D) | 639 |
| 23.2.2 | A/D 控制 / 状态寄存器 (ADCSR) | 640 |
| 23.2.3 | A/D 控制寄存器 (ADCR) | 642 |
| 23.2.4 | ADDRy 格式选择寄存器 (ADDPR) | 644 |
| 23.2.5 | A/D 采样状态寄存器 (ADSSTR) | 644 |
| 23.3 | 运行说明 | 645 |
| 23.3.1 | 单次模式 | 645 |
| 23.3.2 | 扫描模式 | 646 |
| 23.3.3 | 输入采样和 A/D 转换时间 | 648 |
| 23.3.4 | 通过外部触发的启动 | 650 |
| 23.3.5 | 通过 TPU0 的比较匹配 / 输入捕捉 A ~ D 的启动 | 651 |

| | | |
|---------|---|-----|
| 23.3.6 | 通过 TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A 的启动 | 652 |
| 23.3.7 | 通过 TMR 的比较匹配的启动 | 653 |
| 23.4 | 中断源 | 654 |
| 23.5 | A/D 转换精度的定义 | 654 |
| 23.6 | 使用时的注意事项 | 656 |
| 23.6.1 | 模块停止功能的设定 | 656 |
| 23.6.2 | 停止 A/D 转换时的注意事项 | 656 |
| 23.6.3 | 重新开始 A/D 转换时的注意事项 | 656 |
| 23.6.4 | 向低功耗状态转移时的注意事项 | 656 |
| 23.6.5 | 有关容许信号源阻抗 | 656 |
| 23.6.6 | 对绝对精度的影响 | 657 |
| 23.6.7 | 模拟电源引脚等的设定范围 | 657 |
| 23.6.8 | 设计电路板时的注意事项 | 658 |
| 23.6.9 | 噪声对策的注意事项 | 658 |
| 23.6.10 | 高速转换的实现 | 659 |
| 24. | D/A 转换器 | 660 |
| 24.1 | 概要 | 660 |
| 24.2 | 寄存器说明 | 661 |
| 24.2.1 | D/A 数据寄存器 y (DADRy) (y=0,1) | 661 |
| 24.2.2 | D/A 控制寄存器 (DACR) | 662 |
| 24.2.3 | DADRy 格式选择寄存器 (DADPR) | 663 |
| 24.3 | 运行说明 | 664 |
| 24.4 | 使用时的注意事项 | 664 |
| 24.4.1 | 模块停止功能的设定 | 664 |
| 24.4.2 | 模块停止时的 D/A 转换器的运行 | 664 |
| 24.4.3 | 软件待机模式时的 D/A 转换器的运行 | 664 |
| 24.4.4 | 深度软件待机模式时的注意事项 | 664 |
| 25. | RAM | 665 |
| 25.1 | 概要 | 665 |
| 25.2 | 运行说明 | 665 |
| 25.2.1 | 数据的保持 | 665 |
| 25.2.2 | 低功耗功能 | 665 |
| 26. | ROM (保存代码的闪存) | 666 |
| 26.1 | 概要 | 666 |
| 26.2 | 寄存器说明 | 668 |
| 26.2.1 | 闪存模式寄存器 (FMODR) | 669 |
| 26.2.2 | 闪存存取状态寄存器 (FASTAT) | 670 |
| 26.2.3 | 闪存存取错误中断允许寄存器 (FAEINT) | 671 |
| 26.2.4 | FCU RAM 允许寄存器 (FCURAME) | 672 |
| 26.2.5 | 闪存状态寄存器 0 (FSTATR0) | 673 |
| 26.2.6 | 闪存状态寄存器 1 (FSTATR1) | 675 |
| 26.2.7 | 闪存就绪中断允许寄存器 (FRDYIE) | 676 |
| 26.2.8 | 闪存 P/E 模式入口寄存器 (FENTRYR) | 677 |
| 26.2.9 | 闪存保护寄存器 (FPROTR) | 680 |
| 26.2.10 | 闪存复位寄存器 (FRESETR) | 681 |
| 26.2.11 | FCU 命令寄存器 (FCMDR) | 682 |
| 26.2.12 | FCU 处理转换寄存器 (FCPSR) | 683 |
| 26.2.13 | 闪存 P/E 状态寄存器 (FPESTAT) | 683 |
| 26.2.14 | 外围时钟通知寄存器 (PCKAR) | 684 |

| | | |
|---------|---------------------------------|-----|
| 26.2.15 | 闪存编程 / 擦除保护寄存器 (FWEPROR) | 685 |
| 26.3 | ROM 的存储器 MAT 结构 | 685 |
| 26.4 | 块结构 | 686 |
| 26.5 | ROM 相关的运行模式 | 687 |
| 26.6 | ROM 的编程 / 擦除 | 689 |
| 26.6.1 | FCU 的模式 | 689 |
| 26.6.2 | FCU 命令一览表 | 691 |
| 26.6.3 | FCU 的模式和命令的关系 | 693 |
| 26.6.4 | FCU 命令的使用方法 | 694 |
| 26.7 | 挂起 | 711 |
| 26.7.1 | 编程时的挂起 | 711 |
| 26.7.2 | 擦除时的挂起 (挂起优先模式) | 712 |
| 26.7.3 | 擦除时的挂起 (擦除优先模式) | 713 |
| 26.8 | 保护 | 713 |
| 26.8.1 | 软件保护 | 713 |
| 26.8.2 | 错误保护 | 714 |
| 26.9 | 用户引导模式 | 716 |
| 26.10 | 引导模式 | 716 |
| 26.10.1 | 系统结构 | 716 |
| 26.10.2 | ID 码保护 | 716 |
| 26.10.3 | 引导模式的状态转移 | 718 |
| 26.10.4 | 位速率的自动调整 | 720 |
| 26.10.5 | 查询设定主机命令的等待状态 | 721 |
| 26.10.6 | ID 码等待状态 | 731 |
| 26.10.7 | 编程 / 擦除主机命令等待状态 | 732 |
| 26.11 | on-chip 调试器的 ID 码保护 | 740 |
| 26.12 | ROM 码保护 | 740 |
| 26.13 | 使用时的注意事项 | 741 |
| 27. | 数据闪存 (保存数据的闪存) | 742 |
| 27.1 | 概要 | 742 |
| 27.2 | 寄存器说明 | 744 |
| 27.2.1 | 闪存模式寄存器 (FMODR) | 745 |
| 27.2.2 | 闪存存取状态寄存器 (FASTAT) | 746 |
| 27.2.3 | 闪存存取错误中断允许寄存器 (FAEINT) | 748 |
| 27.2.4 | 数据闪存读允许寄存器 (DFLRE) | 749 |
| 27.2.5 | 数据闪存编程 / 擦除允许寄存器 (DFLWE) | 750 |
| 27.2.6 | 闪存 P/E 模式入口寄存器 (FENTRYR) | 751 |
| 27.2.7 | 数据闪存空白检查控制寄存器 (DFLBCCNT) | 752 |
| 27.2.8 | 数据闪存空白检查状态寄存器 (DFLBCSTAT) | 752 |
| 27.3 | 数据闪存的存储器 MAT 结构 | 753 |
| 27.4 | 块结构 | 753 |
| 27.5 | 数据闪存相关的运行模式 | 754 |
| 27.6 | 数据闪存的编程 / 擦除 | 755 |
| 27.6.1 | FCU 的模式 | 755 |
| 27.6.2 | FCU 命令一览表 | 756 |
| 27.6.3 | FCU 的模式和命令的关系 | 757 |
| 27.6.4 | FCU 命令的使用方法 | 759 |
| 27.7 | 保护 | 763 |
| 27.7.1 | 软件保护 | 763 |
| 27.7.2 | 错误保护 | 763 |

| | | |
|------------|-------------------------|------------|
| 27.8 | 引导模式 | 764 |
| 27.8.1 | 查询设定主机命令 | 764 |
| 27.8.2 | 编程 / 擦除主机命令 | 766 |
| 27.9 | 使用时的注意事项 | 767 |
| 28. | 边界扫描 | 768 |
| 28.1 | 概要 | 768 |
| 28.2 | 寄存器说明 | 769 |
| 28.2.1 | 指令寄存器 (JTIR) | 770 |
| 28.2.2 | 旁路寄存器 (JTBPR) | 770 |
| 28.2.3 | 边界扫描寄存器 (JTBSR) | 770 |
| 28.2.4 | IDCODE 寄存器 (JTID) | 776 |
| 28.3 | 运行说明 | 777 |
| 28.3.1 | TAP 控制器 | 777 |
| 28.3.2 | 命令一览 | 778 |
| 28.4 | 使用时的注意事项 | 779 |
| 29. | 电特性 | 781 |
| 29.1 | 绝对最大额定值 | 781 |
| 29.2 | DC 特性 | 782 |
| 29.3 | AC 特性 | 785 |
| 29.3.1 | 时钟时序 | 785 |
| 29.3.2 | 控制信号的时序 | 789 |
| 29.3.3 | 总线时序 | 790 |
| 29.3.4 | 内部外围模块的时序 | 795 |
| 29.4 | A/D 转换特性 | 802 |
| 29.5 | D/A 转换特性 | 802 |
| 29.6 | ROM (保存代码的闪存) 特性 | 803 |
| 29.7 | 数据闪存 (保存数据的闪存) 特性 | 804 |
| 附录 | | 806 |
| 附录 1. | 各运行模式中的端口状态 | 806 |
| 附录 2. | 封装尺寸图 | 809 |

1. 概要

1.1 特点

RX610 群是以高速高性能 RX CPU 为内核的单片机。

以 1 条指令 1 个时钟执行基本指令。强化了运算功能，除了 32 位乘法器和除法器以外，还装载了单精度浮点运算单元。另外，通过以字节为单位的可变长指令以及强化寻址方式，提高编码效率。

作为嵌入式设备所需的外围功能，RX610 群内置了定时器、串行通信接口、I²C 总线接口、A/D 转换器和 D/A 转换器等。

因为 RX610 群具备外部存储器连接功能，所以能直接连接存储器和外围 LSI。内部存储器装载了大容量并且能高速运行的闪存。

1.1.1 用途

OA 设备和数字民用设备等。

1.1.2 规格概要

规格概要如表 1.1 所示。

表 1.1 规格概要 (1/3)

| 分类 | 模块 / 功能 | 说明 |
|----------|---------|--|
| CPU | 中央处理器 | <ul style="list-style-type: none"> 最大工作频率：100MHz 32 位 RX CPU 指令的最短执行时间：1 条指令 1 个时钟 地址空间：4G 字节、线性地址 寄存器 <ul style="list-style-type: none"> 通用寄存器：32 位 × 16 个 控制寄存器：32 位 × 9 个 累加器：64 位 × 1 个 基本指令：73 种 浮点运算指令：8 种 DSP 功能指令：9 种 寻址方式：10 种 数据排列 <ul style="list-style-type: none"> 指令：小端法 数据：可选择小端法或者大端法 32 位乘法器：32 位 × 32 位 → 64 位 除法器：32 位 ÷ 32 位 → 32 位 桶式移位器：32 位 |
| | FPU | <ul style="list-style-type: none"> 单精度浮点数（32 位） 符合 IEEE754 规格的数据类型以及对应浮点异常 |
| 存储器 | ROM | <ul style="list-style-type: none"> ROM 容量：最多 2M 字节 3 种板上编程模式 <ul style="list-style-type: none"> SCI 引导模式、用户编程模式、用户引导模式 |
| | RAM | RAM 容量：128K 字节 |
| | 数据闪存 | 数据 ROM 容量：32K 字节 |
| MCU 运行模式 | | 单芯片模式、内部 ROM 有效扩展模式、内部 ROM 无效扩展模式 |
| 时钟 | 时钟发生电路 | <ul style="list-style-type: none"> 1 个电路：主时钟振荡器 由 PLL 和分频器构成，可选择工作频率。 能分别设定系统时钟、外围模块时钟和外部总线时钟。 <ul style="list-style-type: none"> CPU、DMAC、DTC、ROM、RAM 与系统时钟（ICLK）同步：8 ~ 100MHz 外围模块与外围模块时钟（PCLK）同步：8 ~ 50MHz 连接外部总线的设备与外部总线时钟（BCLK）同步：8 ~ 25MHz |
| 低功耗 | 低功耗功能 | <ul style="list-style-type: none"> 模块停止功能 4 种低功耗状态 <ul style="list-style-type: none"> 睡眠模式、全模块时钟停止模式、软件待机模式、深度软件待机模式 |
| 中断 | 中断控制器 | <ul style="list-style-type: none"> 外围功能中断：116 个中断源 外部中断：16 个中断源（IRQ15 ~ IRQ0 引脚） 非屏蔽中断：1 个中断源（NMI 引脚） 能设定 8 个中断优先级。 |

表 1.1 规格概要 (2/3)

| 分类 | 模块 / 功能 | 说明 |
|--------|--------------|---|
| 外部总线扩展 | | <ul style="list-style-type: none"> 将外部地址空间分为 8 个区域 (CS0 ~ CS7) 进行管理。 各区域: 16M 字节 各区域可输出片选 (CS0# ~ CS7#)。 各区域可选择 8 位总线空间或者 16 位总线空间。 各区域可设定字节序 (只限于数据)。 总线格式: 分离总线 可控制等待。 写缓冲器功能 |
| DMA | DMA 控制器 | <ul style="list-style-type: none"> 4 个通道 启动源: 软件触发、外部中断、外围功能中断 |
| | 数据传送控制器 | <ul style="list-style-type: none"> 传送模式: 正常传送模式、重复传送模式、块传送模式 启动源: 通过中断源来启动 (可进行链传送)。 |
| I/O 端口 | 可编程输入 / 输出端口 | <ul style="list-style-type: none"> 输入 / 输出: 117 个 (144 引脚 LQFP)、140 个 (176 引脚 LFBGA) 上拉电阻: 40 个 漏极开路输出: 16 个 5V 容许: 10 个 |
| 定时器 | 16 位定时器脉冲单元 | <ul style="list-style-type: none"> (16 位 × 6 个通道) × 2 个单元 最多能输入 / 输出 16 个脉冲。 各通道可选择 7 种或者 8 种计数时钟。 输入捕捉 / 输出比较功能 输出最多 15 相的 PWM 波形的 PWM 模式 各通道的缓冲运行、相位计数模式 (2 相编码器输入)、级联运行 (32 位 × 2 个通道)。 能生成 PPG 的输出触发。 能生成 A/D 转换器的转换开始触发。 |
| | 可编程脉冲发生器 | <ul style="list-style-type: none"> (4 位 × 4 组) × 2 个单元 将 TPU 的输出作为触发信号, 输出脉冲。 最多输出 32 位脉冲。 |
| | 8 位定时器 | <ul style="list-style-type: none"> (8 位 × 2 个通道) × 2 个单元 可选择外部时钟和 7 种内部时钟 能进行任意占空比的脉冲输出和 PWM 输出。 能将 2 个通道进行级联运行, 用作 16 位定时器。 能生成 A/D 转换器的转换开始触发。 能生成 SCI5 和 SCI6 的波特率时钟。 |
| | 比较匹配定时器 | <ul style="list-style-type: none"> (16 位 × 2 个通道) × 2 个单元 可选择 4 种计数时钟。 |
| 看门狗定时器 | | <ul style="list-style-type: none"> 8 位 × 1 个通道 可选择 8 种计数时钟。 能转换使用看门狗定时器模式和间隔定时器模式。 |

表 1.1 规格概要 (3/3)

| 分类 | 模块 / 功能 | 说明 |
|---------|-----------------------|---|
| 通信功能 | 串行通信接口 | <ul style="list-style-type: none"> • 7 个通道 • 串行通信方式: 异步 / 时钟同步 / 智能卡接口 • 可通过内部波特率发生器选择任意的位速率。 • 可选择 LSB first 或者 MSB first。 • 能输入 TMR 的平均传送率时钟 (SCI5 和 SCI6)。 |
| | I ² C 总线接口 | <ul style="list-style-type: none"> • 2 个通道 • 通信格式 I²C 总线格式 / SMBus 格式 • 可选择主控模式或者从属模式 (对应多主控)。 • 最大传送速度: 1Mbps |
| A/D 转换器 | | <ul style="list-style-type: none"> • 4 个单元 (单元 × 4 个通道) • 分辨率: 10 位 • 转换时间: 每个通道 1.0μs (以 PCLK=50MHz 运行时) • 2 种运行模式 单次模式、扫描模式 (1 个周期扫描模式 / 连续扫描模式) • 有采样 & 保持功能 • 3 种 A/D 转换的开始方法 软件触发、定时器 (TPU 和 TMR) 的触发、外部触发 |
| D/A 转换器 | | <ul style="list-style-type: none"> • 2 个通道 • 分辨率: 10 位 • 输出电压: 0V ~ VREFH |
| CRC 运算器 | | <ul style="list-style-type: none"> • 对以 8 位为单位的任意数据长度, 生成 CRC 码。 • 可从 3 个多项式中选择: X^8+X^2+X+1、$X^{16}+X^{15}+X^2+1$、$X^{16}+X^{12}+X^5+1$ • 可选择是生成 LSB first 通行的 CRC 码, 还是生成 MSB first 通信的 CRC 码。 |
| 工作频率 | | 8 ~ 100MHz |
| 电源电压 | | VCC=PLL VCC=AVCC=3.0 ~ 3.6V, VREFH=3.0 ~ AVCC |
| 消耗电流 | | 50mA(typ) (普通规格产品) |
| 工作环境温度 | | -20 ~ +85°C (普通规格产品)、-40 ~ +85°C (大温度范围的规格产品) |
| 封装 | | 176 引脚 LFBGA (PLBG0176GA-A) 144 引脚 LQFP (PLQP0144KA-A) |

1.2 产品一览表

产品一览表如表 1.2 所示，产品型号、存储容量和封装如图 1.1 所示。

表 1.2 产品一览表

| 产品型号 | 封装 | ROM 容量 | RAM 容量 | 数据闪存 | 工作频率 (max) |
|--------------|---------------|---------|---------|--------|------------|
| R5F56108VNFP | PLQP0144KA-A | 2M 字节 | 128K 字节 | 32K 字节 | 100MHz |
| R5F56108WNBG | PLBGA0176GA-A | 2M 字节 | 128K 字节 | 32K 字节 | 100MHz |
| R5F56107VNFP | PLQP0144KA-A | 1.5M 字节 | 128K 字节 | 32K 字节 | 100MHz |
| R5F56107WNBG | PLBG0176GA-A | 1.5M 字节 | 128K 字节 | 32K 字节 | 100MHz |
| R5F56106VNFP | PLQP0144KA-A | 1M 字节 | 128K 字节 | 32K 字节 | 100MHz |
| R5F56106WNBG | PLBG0176GA-A | 1M 字节 | 128K 字节 | 32K 字节 | 100MHz |
| R5F56104VNFP | PLQP0144KA-A | 768K 字节 | 128K 字节 | 32K 字节 | 100MHz |
| R5F56104WNBG | PLBG0176GA-A | 768K 字节 | 128K 字节 | 32K 字节 | 100MHz |

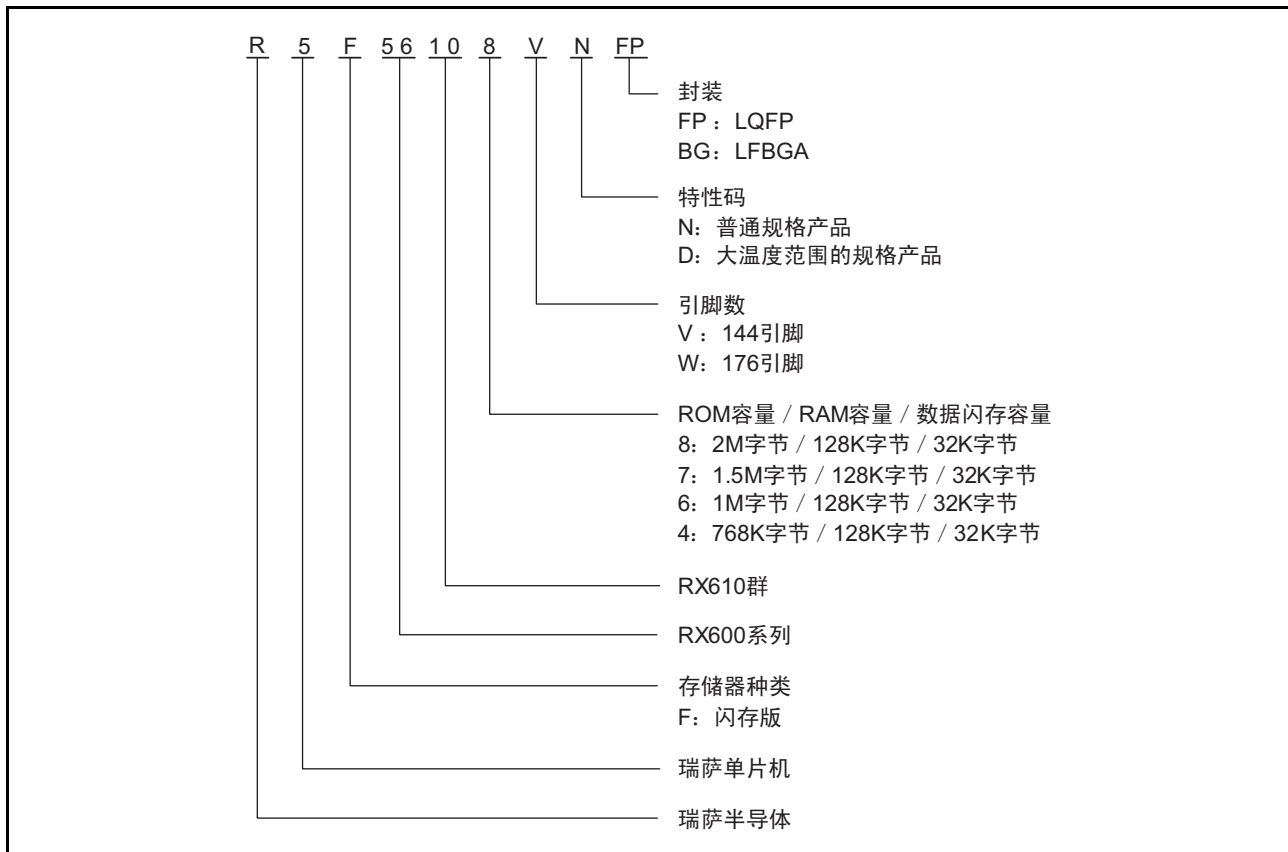


图 1.1 产品型号、存储容量和封装

1.3 框图

框图如图 1.2 所示。

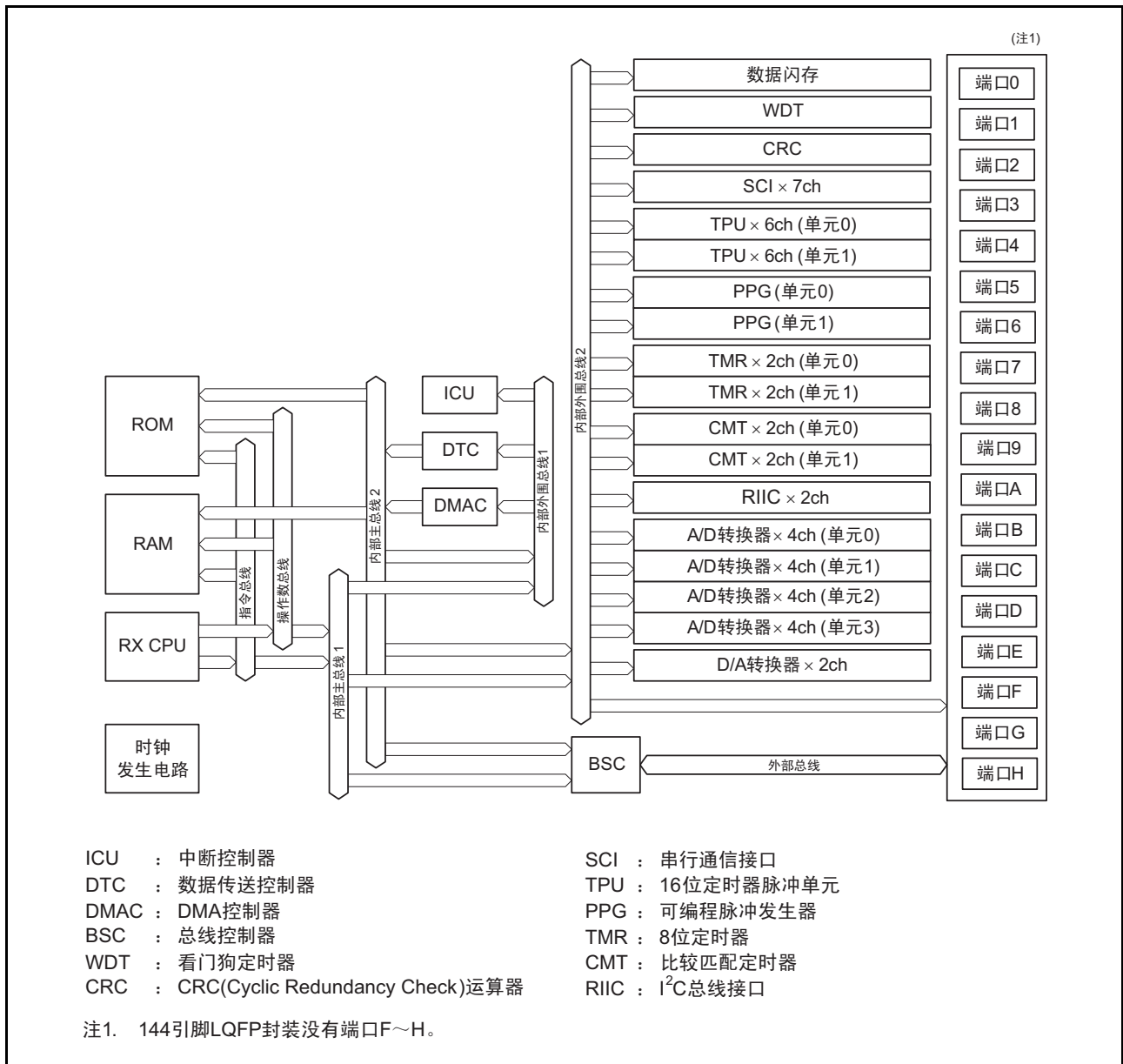


图 1.2 框图

1.4 引脚排列图

176 引脚 LFBGA 的引脚排列图、144 引脚 LQFP 的引脚排列图和 144 引脚 LQFP 的引脚排列图（辅助图）分别如图 1.3、图 1.4 和图 1.5 所示，176 引脚 LFBGA 的各功能引脚一览表和 144 引脚 LQFP 的各功能引脚一览表分别如表 1.3 和表 1.4 所示。

| | A | B | C | D | E | F | G | H | J | K | L | M | N | P | R | | |
|----|-------|-------|--------|------|---|-----|------|-------|-----|-----|-----|-----|-----|--------|-------|-----|-----|
| 15 | PE0 | PE2 | PE5 | PG5 | VSS | PA1 | PA5 | PH1 | P70 | P74 | PB3 | PB6 | PC1 | VCC | PC3 | 15 | |
| 14 | PD6 | PE1 | PE3 | PE7 | PG6 | PA0 | PA4 | PH0 | VCC | P73 | PB4 | PC0 | PC2 | PC4 | PC5 | 14 | |
| 13 | PD4 | PD5 | PD7 | PE6 | PG7 | PA2 | PA6 | VSS | P71 | PB1 | PB5 | VSS | PH2 | PC6 | P75 | 13 | |
| 12 | P63 | VCC | VSS | PE4 | VCC | PA3 | PA7 | PB0 | P72 | PB2 | PB7 | PC7 | P76 | P77 | PH3 | 12 | |
| 11 | P60 | P61 | P62 | P64 | RX610群 PLBG0176GA-A (176引脚LFBGA) (俯视透视图) | | | | | | | | PH4 | VSS | VCC | PH5 | 11 |
| 10 | PD1 | PD0 | PD2 | PD3 | | | | | | | | | P51 | P50 | PH6 | PH7 | 10 |
| 9 | PG2 | PG1 | PG3 | PG4 | | | | | | | | | P81 | P80 | P52 | P53 | 9 |
| 8 | P97 | P96 | BSCANP | PG0 | | | | | | | | | P83 | VSS | VCC | P82 | 8 |
| 7 | P93 | P92 | P94 | P95 | | | | | | | | | P57 | P56 | P54 | P55 | 7 |
| 6 | P90 | VCC | VSS | P91 | | | | | | | | | P37 | P36 | P84 | P35 | 6 |
| 5 | P46 | P45 | P47 | P44 | | | | | | | | | P14 | P12 | P11 | P10 | 5 |
| 4 | P43 | P42 | P41 | P40 | | | | | | | | | P00 | MDE | P86 | VSS | P34 |
| 3 | VREFL | VREFH | P03 | AVSS | EMLE | VCL | P85 | EXTAL | PF6 | P32 | PF3 | VCC | P20 | PLLVCC | PLLSS | 3 | |
| 2 | AVCC | P05 | P66 | P01 | WDTOVF# | MD0 | XTAL | NMI | PF4 | P30 | PF1 | P26 | P24 | P22 | P17 | 2 | |
| 1 | P04 | P67 | P02 | P65 | VSS | MD1 | RESN | VCC | PF5 | P31 | PF2 | P27 | P25 | P23 | P21 | 1 | |
| | A | B | C | D | E | F | G | H | J | K | L | M | N | P | R | | |

图 1.3 176 引脚 LFBGA 的引脚排列图

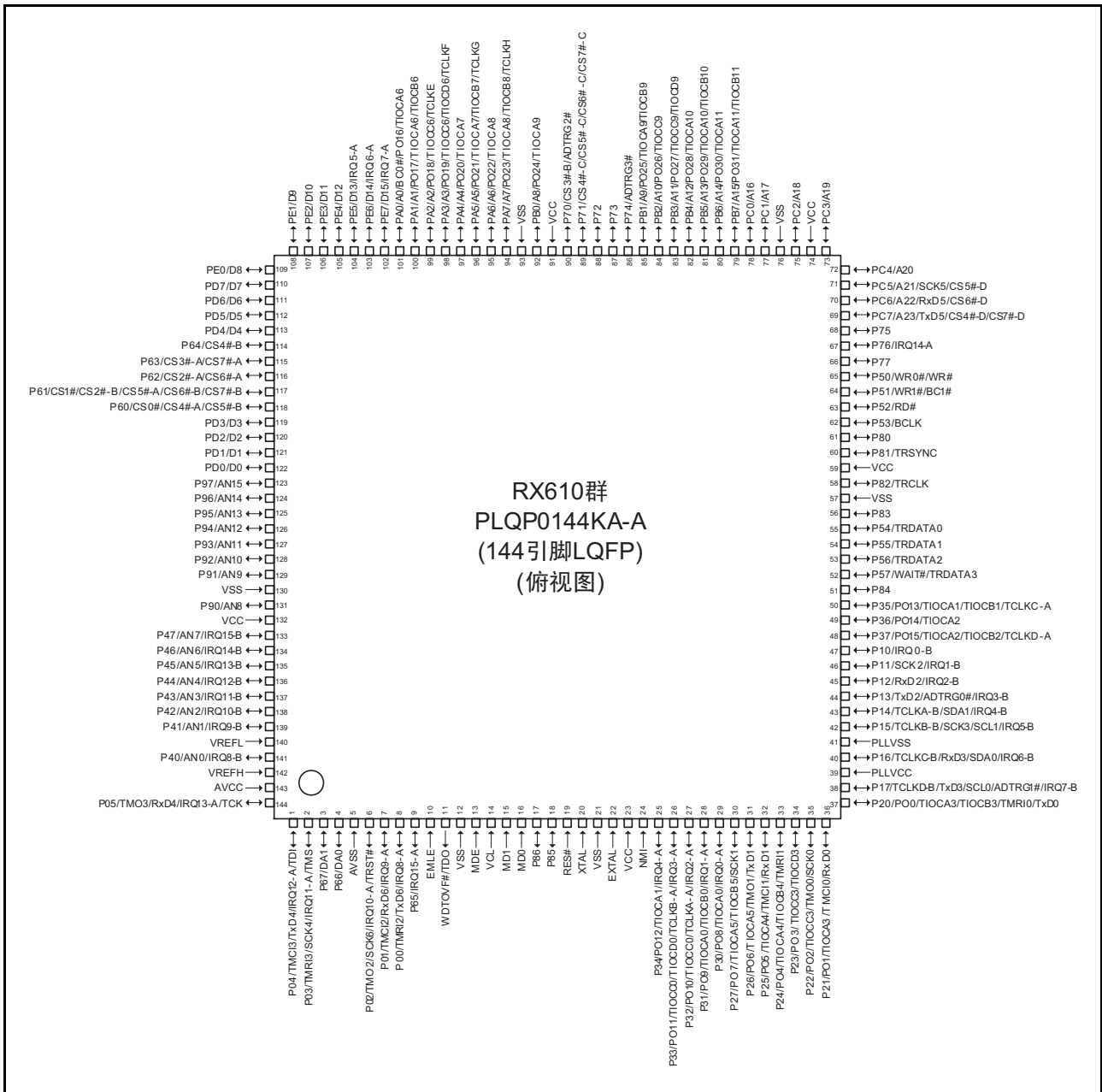


图 1.4 144 引脚 LQFP 的引脚排列图

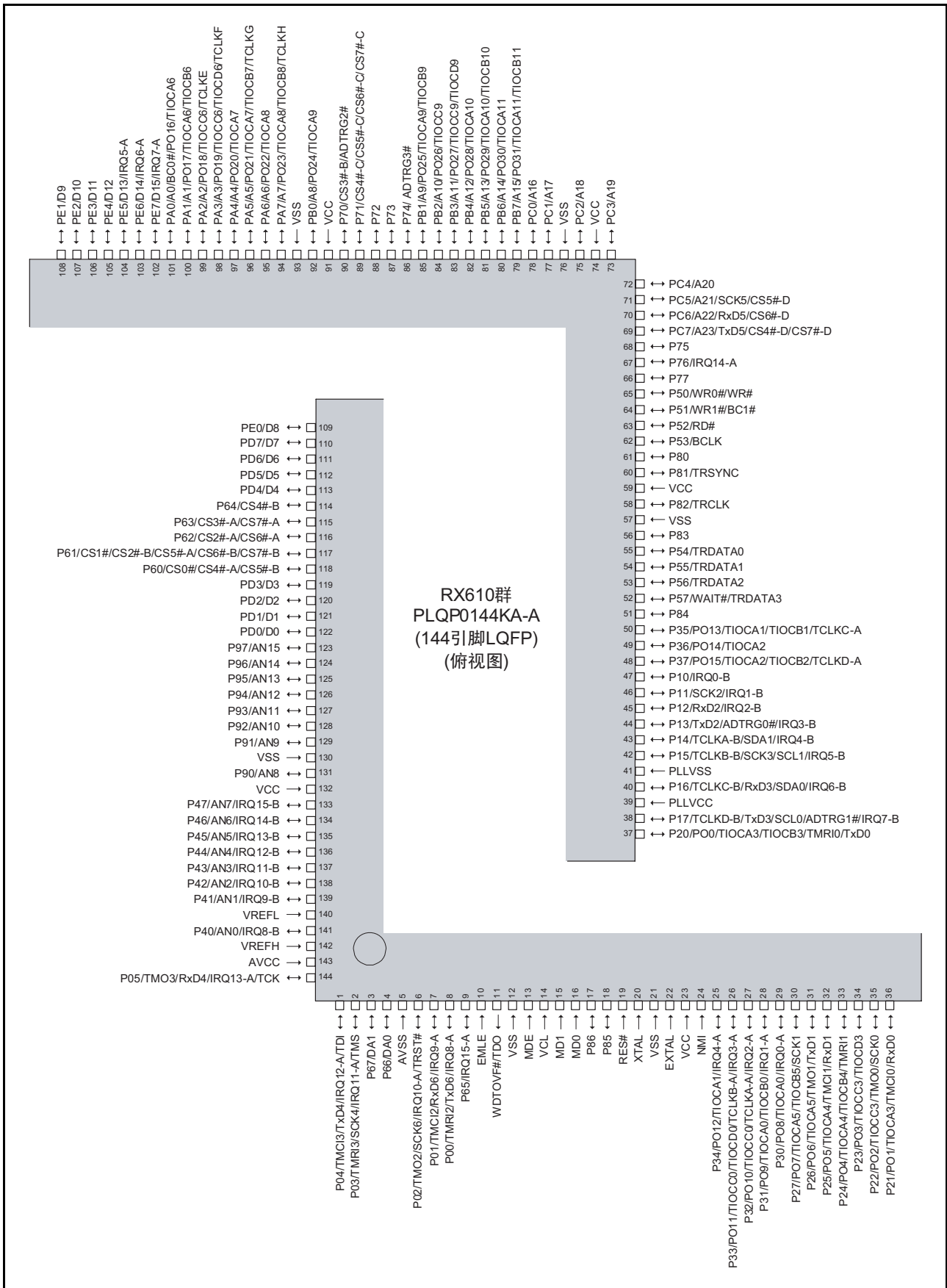


图 1.5 144 引脚 LQFP 的引脚排列图 (辅助图)

表 1.3 各功能引脚一览表 (176 引脚 LFBGA) (1/6)

| 引脚序号 | 电源时钟 系统控制 | I/O 端口 | 中断 | 外部总线 | 定时器 | 通信 | 模拟 | 内部仿真器 |
|------|--------------|--------|---------|--|-------|------|------|-------|
| A1 | | P04 | IRQ12-A | | TMCI3 | TxD4 | | TDI |
| A2 | AVCC | | | | | | | |
| A3 | VREFL | | | | | | | |
| A4 | | P43 | IRQ11-B | | | | AN3 | |
| A5 | | P46 | IRQ14-B | | | | AN6 | |
| A6 | | P90 | | | | | AN8 | |
| A7 | | P93 | | | | | AN11 | |
| A8 | | P97 | | | | | AN15 | |
| A9 | | PG2 | | | | | | |
| A10 | | PD1 | | D1 | | | | |
| A11 | | P60 | | CS0#/ CS4#-A/ CS5#-B | | | | |
| A12 | | P63 | | CS3#-A/ CS7#-A | | | | |
| A13 | | PD4 | | D4 | | | | |
| A14 | | PD6 | | D6 | | | | |
| A15 | | PE0 | | D8 | | | | |
| B1 | | P67 | | | | | DA1 | |
| B2 | | P05 | IRQ13-A | | TMO3 | RxD4 | | TCK |
| B3 | VREFH | | | | | | | |
| B4 | | P42 | IRQ10-B | | | | AN2 | |
| B5 | | P45 | IRQ13-B | | | | AN5 | |
| B6 | VCC | | | | | | | |
| B7 | | P92 | | | | | AN10 | |
| B8 | | P96 | | | | | AN14 | |
| B9 | | PG1 | | | | | | |
| B10 | | PD0 | | D0 | | | | |
| B11 | | P61 | | CS1#/ CS2#-B/ CS5#-A/ CS6#-B/ CS7#-B | | | | |
| B12 | VCC | | | | | | | |
| B13 | | PD5 | | D5 | | | | |
| B14 | | PE1 | | D9 | | | | |
| B15 | | PE2 | | D10 | | | | |
| C1 | | P02 | IRQ10-A | | TMO2 | SCK6 | | TRST# |
| C2 | | P66 | | | | | DA0 | |
| C3 | | P03 | IRQ11-A | | TMRI3 | SCK4 | | TMS |
| C4 | | P41 | IRQ9-B | | | | AN1 | |
| C5 | | P47 | IRQ15-B | | | | AN7 | |
| C6 | VSS | | | | | | | |
| C7 | | P94 | | | | | AN12 | |

表 1.3 各功能引脚一览表 (176 引脚 LFBGA) (2/6)

| 引脚序号 176 引脚 LFBGA | 电源时钟 系统控制 | I/O 端口 | 中断 | 外部总线 | 定时器 | 通信 | 模拟 | 内部仿真器 |
|-------------------------|--------------|--------|---------|-------------------|--------------------------------------|------|------|-------|
| C8 | BSCANP | | | | | | | |
| C9 | | PG3 | | | | | | |
| C10 | | PD2 | | D2 | | | | |
| C11 | | P62 | | CS2#-A/ CS6#-A | | | | |
| C12 | VSS | | | | | | | |
| C13 | | PD7 | | D7 | | | | |
| C14 | | PE3 | | D11 | | | | |
| C15 | | PE5 | IRQ5-A | D13 | | | | |
| D1 | | P65 | IRQ15-A | | | | | |
| D2 | | P01 | IRQ9-A | | TMC12 | RxD6 | | |
| D3 | AVSS | | | | | | | |
| D4 | | P40 | IRQ8-B | | | | AN0 | |
| D5 | | P44 | IRQ12-B | | | | AN4 | |
| D6 | | P91 | | | | | AN9 | |
| D7 | | P95 | | | | | AN13 | |
| D8 | | PG0 | | | | | | |
| D9 | | PG4 | | | | | | |
| D10 | | PD3 | | D3 | | | | |
| D11 | | P64 | | CS4#-B | | | | |
| D12 | | PE4 | | D12 | | | | |
| D13 | | PE6 | IRQ6-A | D14 | | | | |
| D14 | | PE7 | IRQ7-A | D15 | | | | |
| D15 | | PG5 | | | | | | |
| E1 | VSS | | | | | | | |
| E2 | WDTOVF# | | | | | | | TDO |
| E3 | EMLE | | | | | | | |
| E4 | | P00 | IRQ8-A | | TMR12 | TxD6 | | |
| E12 | VCC | | | | | | | |
| E13 | | PG7 | | | | | | |
| E14 | | PG6 | | | | | | |
| E15 | VSS | | | | | | | |
| F1 | MD1 | | | | | | | |
| F2 | MD0 | | | | | | | |
| F3 | VCL | | | | | | | |
| F4 | MDE | | | | | | | |
| F12 | | PA3 | | A3 | PO19/ TIOCC6/ TIOCD6/ TCLKF | | | |
| F13 | | PA2 | | A2 | PO18/ TIOCC6/ TCLKE | | | |
| F14 | | PA0 | | A0/BC0# | PO16/ TIOCA6 | | | |

表 1.3 各功能引脚一览表 (176 引脚 LFBGA) (3 / 6)

| 引脚序号 176 引脚 LFBGA | 电源时钟 系统控制 | I/O 端口 | 中断 | 外部总线 | 定时器 | 通信 | 模拟 | 内部仿真器 |
|-------------------------|--------------|--------|--------|---|--------------------------------------|----|---------|-------|
| F15 | | PA1 | | A1 | PO17/ TIOCA6/ TIOCB6 | | | |
| G1 | RES# | | | | | | | |
| G2 | XTAL | | | | | | | |
| G3 | | P85 | | | | | | |
| G4 | | P86 | | | | | | |
| G12 | | PA7 | | A7 | PO23/ TIOCA8/ TIOCB8/ TCLKH | | | |
| G13 | | PA6 | | A6 | PO22/ TIOCA8 | | | |
| G14 | | PA4 | | A4 | PO20/ TIOCA7 | | | |
| G15 | | PA5 | | A5 | PO21/ TIOCA7/ TIOCB7/ TCLKG | | | |
| H1 | VCC | | | | | | | |
| H2 | | | NMI | | | | | |
| H3 | EXTAL | | | | | | | |
| H4 | VSS | | | | | | | |
| H12 | | PB0 | | A8 | PO24/ TIOCA9 | | | |
| H13 | VSS | | | | | | | |
| H14 | | PH0 | | | | | | |
| H15 | | PH1 | | | | | | |
| J1 | | PF5 | | | | | | |
| J2 | | PF4 | | | | | | |
| J3 | | PF6 | | | | | | |
| J4 | | P34 | IRQ4-A | | PO12/ TIOCA1 | | | |
| J12 | | P72 | | | | | | |
| J13 | | P71 | | CS4#-C/ CS5#-C/ CS6#-C/ CS7#-C | | | | |
| J14 | VCC | | | | | | | |
| J15 | | P70 | | CS3#-B | | | ADTRG2# | |
| K1 | | P31 | IRQ1-A | | PO9/ TIOCA0/ TIOCB0 | | | |
| K2 | | P30 | IRQ0-A | | PO8/ TIOCA0 | | | |

表 1.3 各功能引脚一览表 (176 引脚 LFBGA) (4/6)

| 引脚序号 176 引脚 LFBGA | 电源时钟 系统控制 | I/O 端口 | 中断 | 外部总线 | 定时器 | 通信 | 模拟 | 内部仿真器 |
|-------------------------|--------------|--------|--------|-------|--|------|---------|---------|
| K3 | | P32 | IRQ2-A | | PO10/ TIOCC0/ TCLKA-A | | | |
| K4 | | P33 | IRQ3-A | | PO11/ TIOCC0/ TIOCD0/ TCLKB-A | | | |
| K12 | | PB2 | | A10 | PO26/ TIOCC9 | | | |
| K13 | | PB1 | | A9 | PO25/ TIOCA9/ TIOCB9 | | | |
| K14 | | P73 | | | | | | |
| K15 | | P74 | | | | | ADTRG3# | |
| L1 | | PF2 | | | | | | |
| L2 | | PF1 | | | | | | |
| L3 | | PF3 | | | | | | |
| L4 | | PF0 | | | | | | |
| L12 | | PB7 | | A15 | PO31/ TIOCA11/ TIOCB11 | | | |
| L13 | | PB5 | | A13 | PO29/ TIOCA10/ TIOCB10 | | | |
| L14 | | PB4 | | A12 | PO28/ TIOCA10 | | | |
| L15 | | PB3 | | A11 | PO27/ TIOCC9/ TIOCD9 | | | |
| M1 | | P27 | | | PO7/ TIOCA5/ TIOCB5 | SCK1 | | |
| M2 | | P26 | | | PO6/ TIOCA5/ TMO1 | TxD1 | | |
| M3 | VCC | | | | | | | |
| M4 | VSS | | | | | | | |
| M5 | | P14 | IRQ4-B | | TCLKA-B | SDA1 | | |
| M6 | | P37 | | | PO15/ TIOCA2/ TIOCB2/ TCLKD-A | | | |
| M7 | | P57 | | WAIT# | | | | TRDATA3 |
| M8 | | P83 | | | | | | |
| M9 | | P81 | | | | | | TRSYNC |

表 1.3 各功能引脚一览表 (176 引脚 LFBGA) (5 / 6)

| 引脚序号 176 引脚 LFBGA | 电源时钟 系统控制 | I/O 端口 | 中断 | 外部总线 | 定时器 | 通信 | 模拟 | 内部仿真器 |
|-------------------------|--------------|--------|---------|---------------------------|-------------------------------------|-----------|----|---------|
| M10 | | P51 | | WR1#/ BC1# | | | | |
| M11 | | PH4 | | | | | | |
| M12 | | PC7 | | A23/ CS4#-D/ CS7#-D | | TxD5 | | |
| M13 | VSS | | | | | | | |
| M14 | | PC0 | | A16 | | | | |
| M15 | | PB6 | | A14 | PO30/ TIOCA11 | | | |
| N1 | | P25 | | | PO5/ TIOCA4/ TMC11 | RxD1 | | |
| N2 | | P24 | | | PO4/ TIOCA4/ TIOCB4/ TMR11 | | | |
| N3 | | P20 | | | PO0/ TIOCA3/ TIOCB3/ TMR10 | TxD0 | | |
| N4 | | P16 | IRQ6-B | | TCLKC-B | RxD3/SDA0 | | |
| N5 | | P12 | IRQ2-B | | | RxD2 | | |
| N6 | | P36 | | | PO14/ TIOCA2 | | | |
| N7 | | P56 | | | | | | TRDATA2 |
| N8 | VSS | | | | | | | |
| N9 | | P80 | | | | | | |
| N10 | | P50 | | WR0#/WR# | | | | |
| N11 | VSS | | | | | | | |
| N12 | | P76 | IRQ14-A | | | | | |
| N13 | | PH2 | | | | | | |
| N14 | | PC2 | | A18 | | | | |
| N15 | | PC1 | | A17 | | | | |
| P1 | | P23 | | | PO3/ TIOCC3/ TIOCD3 | | | |
| P2 | | P22 | | | PO2/ TIOCC3/ TMO0 | SCK0 | | |
| P3 | PLLVC | | | | | | | |
| P4 | | P15 | IRQ5-B | | TCLKB-B | SCK3/SCL1 | | |
| P5 | | P11 | IRQ1-B | | | SCK2 | | |
| P6 | | P84 | | | | | | |
| P7 | | P54 | | | | | | TRDATA0 |

表 1.3 各功能引脚一览表 (176 引脚 LFBGA) (6 / 6)

| 引脚序号 176 引脚 LFBGA | 电源时钟 系统控制 | I/O 端口 | 中断 | 外部总线 | 定时器 | 通信 | 模拟 | 内部仿真器 |
|-------------------------|--------------|--------|--------|----------------|--|-----------|---------|---------|
| P8 | VCC | | | | | | | |
| P9 | | P52 | | RD# | | | | |
| P10 | | PH6 | | | | | | |
| P11 | VSS | | | | | | | |
| P12 | | P77 | | | | | | |
| P13 | | PC6 | | A22/ CS6#-D | | RxD5 | | |
| P14 | | PC4 | | A20 | | | | |
| P15 | VCC | | | | | | | |
| R1 | | P21 | | | PO1/ TIOCA3/ TMCIO | RxD0 | | |
| R2 | | P17 | IRQ7-B | | TCLKD-B | TxD3/SCL0 | ADTRG1# | |
| R3 | PLLSS | | | | | | | |
| R4 | | P13 | IRQ3-B | | | TxD2 | ADTRG0# | |
| R5 | | P10 | IRQ0-B | | | | | |
| R6 | | P35 | | | PO13/ TIOCA1/ TIOCB1/ TCLKC-A | | | |
| R7 | | P55 | | | | | | TRDATA1 |
| R8 | | P82 | | | | | | TRCLK |
| R9 | BCLK | P53 | | | | | | |
| R10 | | PH7 | | | | | | |
| R11 | | PH5 | | | | | | |
| R12 | | PH3 | | | | | | |
| R13 | | P75 | | | | | | |
| R14 | | PC5 | | A21/ CS5#-D | | SCK5 | | |
| R15 | | PC3 | | A19 | | | | |

表 1.4 各功能引脚一览表 (144 引脚 LQFP) (1/5)

| 引脚序号 | 电源时钟 系统控制 | I/O 端口 | 中断 | 外部总线 | 定时器 | 通信 | 模拟 | 内部仿真器 |
|------|--------------|--------|---------|------|--|------|-----|-------|
| 1 | | P04 | IRQ12-A | | TMCI3 | TxD4 | | TDI |
| 2 | | P03 | IRQ11-A | | TMRI3 | SCK4 | | TMS |
| 3 | | P67 | | | | | DA1 | |
| 4 | | P66 | | | | | DA0 | |
| 5 | AVSS | | | | | | | |
| 6 | | P02 | IRQ10-A | | TMO2 | SCK6 | | TRST# |
| 7 | | P01 | IRQ9-A | | TMCI2 | RxD6 | | |
| 8 | | P00 | IRQ8-A | | TMRI2 | TxD6 | | |
| 9 | | P65 | IRQ15-A | | | | | |
| 10 | EMLE | | | | | | | |
| 11 | WDTOVF# | | | | | | | TDO |
| 12 | VSS | | | | | | | |
| 13 | MDE | | | | | | | |
| 14 | VCL | | | | | | | |
| 15 | MD1 | | | | | | | |
| 16 | MD0 | | | | | | | |
| 17 | | P86 | | | | | | |
| 18 | | P85 | | | | | | |
| 19 | RES# | | | | | | | |
| 20 | XTAL | | | | | | | |
| 21 | VSS | | | | | | | |
| 22 | EXTAL | | | | | | | |
| 23 | VCC | | | | | | | |
| 24 | | | NMI | | | | | |
| 25 | | P34 | IRQ4-A | | PO12/ TIOCA1 | | | |
| 26 | | P33 | IRQ3-A | | PO11/ TIOCC0/ TIOCD0/ TCLKB-A | | | |
| 27 | | P32 | IRQ2-A | | PO10/ TIOCC0/ TCLKA-A | | | |
| 28 | | P31 | IRQ1-A | | PO9/ TIOCA0/ TIOCB0 | | | |
| 29 | | P30 | IRQ0-A | | PO8/ TIOCA0 | | | |
| 30 | | P27 | | | PO7/ TIOCA5/ TIOCB5 | SCK1 | | |
| 31 | | P26 | | | PO6/ TIOCA5/ TMO1 | TxD1 | | |

表 1.4 各功能引脚一览表 (144 引脚 LQFP) (2 / 5)

| 引脚序号 144 引脚 LQFP | 电源时钟 系统控制 | I/O 端口 | 中断 | 外部总线 | 定时器 | 通信 | 模拟 | 内部仿真器 |
|------------------------|--------------|--------|--------|-------|--|-----------|---------|---------|
| 32 | | P25 | | | PO5/ TIOCA4/ TMCI1 | RxD1 | | |
| 33 | | P24 | | | PO4/ TIOCA4/ TIOCB4/ TMR1 | | | |
| 34 | | P23 | | | PO3/ TIOCC3/ TIOCD3 | | | |
| 35 | | P22 | | | PO2/ TIOCC3/ TMO0 | SCK0 | | |
| 36 | | P21 | | | PO1/ TIOCA3/ TMCI0 | RxD0 | | |
| 37 | | P20 | | | PO0/ TIOCA3/ TIOCB3/ TMR0 | TxD0 | | |
| 38 | | P17 | IRQ7-B | | TCLKD-B | TxD3/SCL0 | ADTRG1# | |
| 39 | PLLVC | | | | | | | |
| 40 | | P16 | IRQ6-B | | TCLKC-B | RxD3/SDA0 | | |
| 41 | PLLVS | | | | | | | |
| 42 | | P15 | IRQ5-B | | TCLKB-B | SCK3/SCL1 | | |
| 43 | | P14 | IRQ4-B | | TCLKA-B | SDA1 | | |
| 44 | | P13 | IRQ3-B | | | TxD2 | ADTRG0# | |
| 45 | | P12 | IRQ2-B | | | RxD2 | | |
| 46 | | P11 | IRQ1-B | | | SCK2 | | |
| 47 | | P10 | IRQ0-B | | | | | |
| 48 | | P37 | | | PO15/ TIOCA2/ TIOCB2/ TCLKD-A | | | |
| 49 | | P36 | | | PO14/ TIOCA2 | | | |
| 50 | | P35 | | | PO13/ TIOCA1/ TIOCB1/ TCLKC-A | | | |
| 51 | | P84 | | | | | | |
| 52 | | P57 | | WAIT# | | | | TRDATA3 |
| 53 | | P56 | | | | | | TRDATA2 |
| 54 | | P55 | | | | | | TRDATA1 |
| 55 | | P54 | | | | | | TRDATA0 |

表 1.4 各功能引脚一览表 (144 引脚 LQFP) (3/5)

| 引脚序号 144 引脚 LQFP | 电源时钟 系统控制 | I/O 端口 | 中断 | 外部总线 | 定时器 | 通信 | 模拟 | 内部仿真器 |
|------------------------|--------------|--------|---------|---------------------------|------------------------------|------|----|--------|
| 56 | | P83 | | | | | | |
| 57 | VSS | | | | | | | |
| 58 | | P82 | | | | | | TRCLK |
| 59 | VCC | | | | | | | |
| 60 | | P81 | | | | | | TRSYNC |
| 61 | | P80 | | | | | | |
| 62 | BCLK | P53 | | | | | | |
| 63 | | P52 | | RD# | | | | |
| 64 | | P51 | | WR1#/ BC1# | | | | |
| 65 | | P50 | | WR0#/WR# | | | | |
| 66 | | P77 | | | | | | |
| 67 | | P76 | IRQ14-A | | | | | |
| 68 | | P75 | | | | | | |
| 69 | | PC7 | | A23/ CS4#-D/ CS7#-D | | TxD5 | | |
| 70 | | PC6 | | A22/ CS6#-D | | RxD5 | | |
| 71 | | PC5 | | A21/ CS5#-D | | SCK5 | | |
| 72 | | PC4 | | A20 | | | | |
| 73 | | PC3 | | A19 | | | | |
| 74 | VCC | | | | | | | |
| 75 | | PC2 | | A18 | | | | |
| 76 | VSS | | | | | | | |
| 77 | | PC1 | | A17 | | | | |
| 78 | | PC0 | | A16 | | | | |
| 79 | | PB7 | | A15 | PO31/ TIOCA11/ TIOCB11 | | | |
| 80 | | PB6 | | A14 | PO30/ TIOCA11 | | | |
| 81 | | PB5 | | A13 | PO29/ TIOCA10/ TIOCB10 | | | |
| 82 | | PB4 | | A12 | PO28/ TIOCA10 | | | |
| 83 | | PB3 | | A11 | PO27/ TIOCC9/ TIOCD9 | | | |
| 84 | | PB2 | | A10 | PO26/ TIOCC9 | | | |

表 1.4 各功能引脚一览表 (144 引脚 LQFP) (4 / 5)

| 引脚序号 144 引脚 LQFP | 电源时钟 系统控制 | I/O 端口 | 中断 | 外部总线 | 定时器 | 通信 | 模拟 | 内部仿真器 |
|------------------------|--------------|--------|--------|---|--------------------------------------|----|---------|-------|
| 85 | | PB1 | | A9 | PO25/ TIOCA9/ TIOCB9 | | | |
| 86 | | P74 | | | | | ADTRG3# | |
| 87 | | P73 | | | | | | |
| 88 | | P72 | | | | | | |
| 89 | | P71 | | CS4#-C/ CS5#-C/ CS6#-C/ CS7#-C | | | | |
| 90 | | P70 | | CS3#-B | | | ADTRG2# | |
| 91 | VCC | | | | | | | |
| 92 | | PB0 | | A8 | PO24/ TIOCA9 | | | |
| 93 | VSS | | | | | | | |
| 94 | | PA7 | | A7 | PO23/ TIOCA8/ TIOCB8/ TCLKH | | | |
| 95 | | PA6 | | A6 | PO22/ TIOCA8 | | | |
| 96 | | PA5 | | A5 | PO21/ TIOCA7/ TIOCB7/ TCLKG | | | |
| 97 | | PA4 | | A4 | PO20/ TIOCA7 | | | |
| 98 | | PA3 | | A3 | PO19/ TIOCC6/ TIOCD6/ TCLKF | | | |
| 99 | | PA2 | | A2 | PO18/ TIOCC6/ TCLKE | | | |
| 100 | | PA1 | | A1 | PO17/ TIOCA6/ TIOCB6 | | | |
| 101 | | PA0 | | A0/BC0# | PO16/ TIOCA6 | | | |
| 102 | | PE7 | IRQ7-A | D15 | | | | |
| 103 | | PE6 | IRQ6-A | D14 | | | | |
| 104 | | PE5 | IRQ5-A | D13 | | | | |
| 105 | | PE4 | | D12 | | | | |
| 106 | | PE3 | | D11 | | | | |
| 107 | | PE2 | | D10 | | | | |

表 1.4 各功能引脚一览表 (144 引脚 LQFP) (5/5)

| 引脚序号 144 引脚 LQFP | 电源时钟 系统控制 | I/O 端口 | 中断 | 外部总线 | 定时器 | 通信 | 模拟 | 内部仿真器 |
|------------------------|--------------|--------|---------|--|------|------|------|-------|
| 108 | | PE1 | | D9 | | | | |
| 109 | | PE0 | | D8 | | | | |
| 110 | | PD7 | | D7 | | | | |
| 111 | | PD6 | | D6 | | | | |
| 112 | | PD5 | | D5 | | | | |
| 113 | | PD4 | | D4 | | | | |
| 114 | | P64 | | CS4#-B | | | | |
| 115 | | P63 | | CS3#-A/ CS7#-A | | | | |
| 116 | | P62 | | CS2#-A/ CS6#-A | | | | |
| 117 | | P61 | | CS1#/ CS2#-B/ CS5#-A/ CS6#-B/ CS7#-B | | | | |
| 118 | | P60 | | CS0#/ CS4#-A/ CS5#-B | | | | |
| 119 | | PD3 | | D3 | | | | |
| 120 | | PD2 | | D2 | | | | |
| 121 | | PD1 | | D1 | | | | |
| 122 | | PD0 | | D0 | | | | |
| 123 | | P97 | | | | | AN15 | |
| 124 | | P96 | | | | | AN14 | |
| 125 | | P95 | | | | | AN13 | |
| 126 | | P94 | | | | | AN12 | |
| 127 | | P93 | | | | | AN11 | |
| 128 | | P92 | | | | | AN10 | |
| 129 | | P91 | | | | | AN9 | |
| 130 | VSS | | | | | | | |
| 131 | | P90 | | | | | AN8 | |
| 132 | VCC | | | | | | | |
| 133 | | P47 | IRQ15-B | | | | AN7 | |
| 134 | | P46 | IRQ14-B | | | | AN6 | |
| 135 | | P45 | IRQ13-B | | | | AN5 | |
| 136 | | P44 | IRQ12-B | | | | AN4 | |
| 137 | | P43 | IRQ11-B | | | | AN3 | |
| 138 | | P42 | IRQ10-B | | | | AN2 | |
| 139 | | P41 | IRQ9-B | | | | AN1 | |
| 140 | VREFL | | | | | | | |
| 141 | | P40 | IRQ8-B | | | | AN0 | |
| 142 | VREFH | | | | | | | |
| 143 | AVCC | | | | | | | |
| 144 | | P05 | IRQ13-A | | TMO3 | RxD4 | | TCK |

1.5 引脚功能

引脚功能一览表如表 1.5 所示。

表 1.5 引脚功能一览表 (1/4)

| 分类 | 引脚名 | 输入 / 输出 | 功 能 |
|---------|-------------------|---------|--|
| 电源 | VCC | 输入 | 电源引脚 必须连接系统电源。 |
| | VCL | 输入 | 必须通过 0.1 μ F 的电容器连接 VSS，并且使电容器靠近引脚。 |
| | VSS | 输入 | 接地引脚 必须连接系统电源 (0V)。 |
| | PLLVCC | 输入 | 用于 PLL 电路的电源引脚。 必须连接系统电源。 |
| | PLLVSS | 输入 | 用于 PLL 电路的接地引脚。 |
| 时钟 | XTAL | 输入 | 连接晶体谐振器的引脚。 |
| | EXTAL | 输入 | EXTAL 引脚也能输入外部时钟。 |
| | BCLK | 输出 | 给外部设备提供系统时钟。 |
| 运行模式的控制 | MD0、MD1、MDE | 输入 | 设定运行模式。不能在运行中改变这些引脚。 |
| 系统控制 | RES# | 输入 | 复位引脚 当此引脚为 Low 电平时，进入复位状态。 |
| | EMLE | 输入 | 允许内部仿真器的引脚。 当使用内部仿真器时，必须置为 High 电平；否则，必须置为 Low 电平。 |
| | BSCANP | 输入 | 允许边界扫描的引脚。 当此引脚为 High 电平时，边界扫描有效。当不使用边界扫描时，必须置为 Low 电平。 |
| 内部仿真器 | TRST# | 输入 | 用于内部仿真器的引脚。 |
| | TMS | 输入 | 如果将 EMLE 引脚置为 High 电平，就变为内部仿真器的专用引脚。 |
| | TDI | 输入 | |
| | TCK | 输入 | |
| | TDO | 输出 | |
| | TRCLK | 输出 | 输出用于取得与跟踪数据同步的时钟。 |
| | TRSYNC | 输出 | 表示 TRDATA0 ~ TRDATA3 引脚的输出为有效数据。 |
| | TRDATA0 ~ TRDATA3 | 输出 | 输出跟踪信息。 |
| 地址总线 | A0 ~ A23 | 输出 | 地址的输出引脚。 |
| 数据总线 | D0 ~ D15 | 输入 / 输出 | 双向数据总线。 |

表 1.5 引脚功能一览表 (2/4)

| 分类 | 引脚名 | 输入 / 输出 | 功 能 |
|--|---|---------|--|
| 总线控制 | RD# | 输出 | 表示正在读外部总线接口空间的选通信号。 |
| | WR0# | 输出 | 表示在字节选通模式中正在写外部总线接口空间并且低位字节 (D0 ~ D7) 有效的选通信号。 |
| | WR1# | 输出 | 表示在字节选通模式中正在写外部总线接口空间并且高位字节 (D8 ~ D15) 有效的选通信号。 |
| | WR# | 输出 | 表示在 1 次写选通模式中正在写外部总线接口空间的选通信号。 |
| | BC0# | 输出 | 表示在 1 次写选通模式中正在写外部总线接口空间并且低位字节 (D0 ~ D7) 有效的选通信号。 |
| | BC1# | 输出 | 表示在 1 次写选通模式中正在写外部总线接口空间并且高位字节 (D8 ~ D15) 有效的选通信号。 |
| | CS0#、CS1# CS2#-A/CS2#-B CS3#-A/CS3#-B CS4#-A/CS4#-B/ CS4#-C/CS4#-D CS5#-A/CS5#-B/ CS5#-C/CS5#-D CS6#-A/CS6#-B/ CS6#-C/CS6#-D CS7#-A/CS7#-B/ CS7#-C/CS7#-D | 输出 | 区域 0 ~ 7 的选择信号。 |
| | WAIT# | 输入 | 存取外部空间时的等待请求信号。 |
| | 中断 | NMI | 输入 |
| IRQ0-A/IRQ0-B IRQ1-A/IRQ1-B IRQ2-A/IRQ2-B IRQ3-A/IRQ3-B IRQ4-A/IRQ4-B IRQ5-A/IRQ5-B IRQ6-A/IRQ6-B IRQ7-A/IRQ7-B IRQ8-A/IRQ8-B IRQ9-A/IRQ9-B IRQ10-A/IRQ10-B IRQ11-A/IRQ11-B IRQ12-A/IRQ12-B IRQ13-A/IRQ13-B IRQ14-A/IRQ14-B IRQ15-A/IRQ15-B | | 输入 | 中断请求的引脚。 |

表 1.5 引脚功能一览表 (3/4)

| 分类 | 引脚名 | 输入 / 输出 | 功 能 |
|--|--|---------|---|
| 16 位定时器 脉冲单元 | TIOCA0、TIOCB0 TIOCC0、TIOCD0 | 输入 / 输出 | TGRA0 ~ TGRD0 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| | TIOCA1、TIOCB1 | 输入 / 输出 | TGRA1 和 TGRB1 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| | TIOCA2、TIOCB2 | 输入 / 输出 | TGRA2 和 TGRB2 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| | TIOCA3、TIOCB3 TIOCC3、TIOCD3 | 输入 / 输出 | TGRA3 ~ TGRD3 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| | TIOCA4、TIOCB4 | 输入 / 输出 | TGRA4 和 TGRB4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| | TIOCA5、TIOCB5 | 输入 / 输出 | TGRA5 和 TGRB5 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| | TIOCA6、TIOCB6 TIOCC6、TIOCD6 | 输入 / 输出 | TGRA6 ~ TGRD6 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| | TIOCA7、TIOCB7 | 输入 / 输出 | TGRA7 和 TGRB7 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| | TIOCA8、TIOCB8 | 输入 / 输出 | TGRA8 和 TGRB8 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| | TIOCA9、TIOCB9 TIOCC9、TIOCD9 | 输入 / 输出 | TGRA9 ~ TGRD9 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| | TIOCA10、TIOCB10 | 输入 / 输出 | TGRA10 和 TGRB10 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| | TIOCA11、TIOCB11 | 输入 / 输出 | TGRA11 和 TGRB11 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 |
| TCLKA-A/TCLKA-B TCLKB-A/TCLKB-B TCLKC-A/TCLKC-B TCLKD-A/TCLKD-B TCLKE、TCLKF TCLKG、TCLKH | 输入 | 输入外部时钟。 | |
| 可编程脉冲发生器 | PO0 ~ PO31 | 输出 | 脉冲输出引脚。 |
| 8 位定时器 | TMO0 ~ TMO3 | 输出 | 比较匹配的输出引脚。 |
| | TMC10 ~ TMC13 | 输入 | 输入到计数器的外部时钟的输入引脚。 |
| | TMRI0 ~ TMRI3 | 输入 | 计数器复位的输入引脚。 |
| 看门狗定时器 | WDTOVF# | 输出 | 看门狗定时器模式中的计数器溢出信号的输出引脚。 |
| 串行通信接口 | TxD0、TxD1、TxD2、 TxD3、TxD4、TxD5、 TxD6 | 输出 | 发送数据的输出引脚。 |
| | RxD0、RxD1、RxD2、 RxD3、RxD4、RxD5、 RxD6 | 输入 | 接收数据的输入引脚。 |
| | SCK0、SCK1、SCK2、 SCK3、SCK4、SCK5、 SCK6 | 输入 / 输出 | 时钟的输入 / 输出引脚。 |

表 1.5 引脚功能一览表 (4/4)

| 分类 | 引脚名 | 输入 / 输出 | 功 能 |
|-----------------------|-------------------|---------------|---|
| I ² C 总线接口 | SCL0、SCL1 | 输入 / 输出 | RIIC 的时钟输入 / 输出引脚 能通过 NMOS 漏极开路输出，直接驱动总线。 |
| | SDA0、SDA1 | 输入 / 输出 | RIIC 的数据输入 / 输出引脚 能通过 NMOS 漏极开路输出，直接驱动总线。 |
| A/D 转换器 | AN0 ~ AN15 | 输入 | A/D 转换器的模拟输入引脚 |
| | ADTRG0# ~ ADTRG3# | 输入 | A/D 转换开始的外部触发输入引脚 |
| D/A 转换器 | DA0、DA1 | 输出 | D/A 转换器的模拟输出引脚 |
| 模拟电源 | AVCC | 输入 | A/D 转换器和 D/A 转换器的模拟电源引脚 在不使用 A/D 转换器和 D/A 转换器时，必须连接系统电源。 |
| | AVSS | 输入 | A/D 转换器和 D/A 转换器的接地引脚 必须连接系统电源 (0V)。 |
| | VREFH | 输入 | A/D 转换器和 D/A 转换器的基准电源引脚 在不使用 A/D 转换器和 D/A 转换器时，必须连接系统电源。 |
| | VREFL | 输入 | A/D 转换器和 D/A 转换器的基准接地引脚 必须连接模拟基准电源 (0V)。在不使用 A/D 转换器和 D/A 转换器时，必须连接系统电源 (0V)。详细内容请参照“23.6.7 模拟电源引脚等的设定范围”。 |
| I/O 端口 | P00 ~ P05 | 输入 / 输出 | 6 位的输入 / 输出引脚 |
| | P10 ~ P17 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | P20 ~ P27 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | P30 ~ P37 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | P40 ~ P47 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | P50 ~ P57 | 输入 / 输出 | 8 位的输入 / 输出引脚 (P53 为输入专用) |
| | P60 ~ P67 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | P70 ~ P77 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | P80 ~ P86 | 输入 / 输出 | 7 位的输入 / 输出引脚 |
| | P90 ~ P97 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | PA0 ~ PA7 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | PB0 ~ PB7 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | PC0 ~ PC7 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | PD0 ~ PD7 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | PE0 ~ PE7 | 输入 / 输出 | 8 位的输入 / 输出引脚 |
| | PF0 ~ PF6 | 输入 / 输出 | 7 位的输入 / 输出引脚 |
| PG0 ~ PG7 | 输入 / 输出 | 8 位的输入 / 输出引脚 | |
| PH0 ~ PH7 | 输入 / 输出 | 8 位的输入 / 输出引脚 | |

2. CPU

RX610 群是装载了 RX CPU 的处理器。

RX CPU 采用了可变长指令格式。通过将常用指令分配为较短的指令长度，能开发出存储容量小而且效率高的程序。

RX CPU 有 73 种基本指令、8 种浮点运算指令、9 种 DSP 功能指令共 90 种指令和 10 种寻址方式，进行寄存器 - 寄存器之间、寄存器 - 存储器之间、立即数 - 寄存器、立即数 - 存储器的运算以及位操作、存储器 - 存储器之间的传送。通过 1 个时钟执行寄存器之间的运算指令和多条复合指令，实现了快速运算处理。因为内置了乘法器和除法器，所以能进行快速的乘除法运算处理。

RX CPU 通过取指令、解码、执行、存储器的存取、回写这 5 个阶段的流水线处理，进行指令的处理。如果存储器存取引起流水线的延长，也有可能先执行后面的运算。RX CPU 采用这种乱序执行进行节省时钟周期数的指令执行控制。

2.1 特点

- 指令的最短执行时间：1 个时钟执行 1 条指令
- 地址空间：4G 字节、线性地址
- CPU 寄存器组
 - 通用寄存器：32 位×16 个
 - 控制寄存器：32 位×9 个
 - 累加器：64 位×1 个
- 基本指令：73 种（算术/逻辑指令、传送指令、转移指令、位操作指令、字符串操作指令、系统操作指令）
 - 对应转移距离的相对转移指令
 - 可变长指令格式（1 字节～8 字节）
 - 常用指令有缩短格式
- 浮点运算指令：8 种
- DSP 功能指令：9 种
 - 对应 16 位×16 位的乘法指令和乘加指令
 - 对应累加器的舍入指令
- 寻址方式：10 种
- 5 个阶段的流水线
 - 采用乱序执行
- 处理器模式
 - 管理模式、用户模式
- 浮点运算单元
 - 对应单精度浮点数（32 位）
 - 符合 IEEE754 规格的数据类型以及对应浮点异常
- 数据排列
 - 可选择小端法或者大端法

2.2 CPU 寄存器组

RX CPU 寄存器有 16 个通用寄存器、9 个控制寄存器和 1 个由 DSP 功能指令使用的累加器。

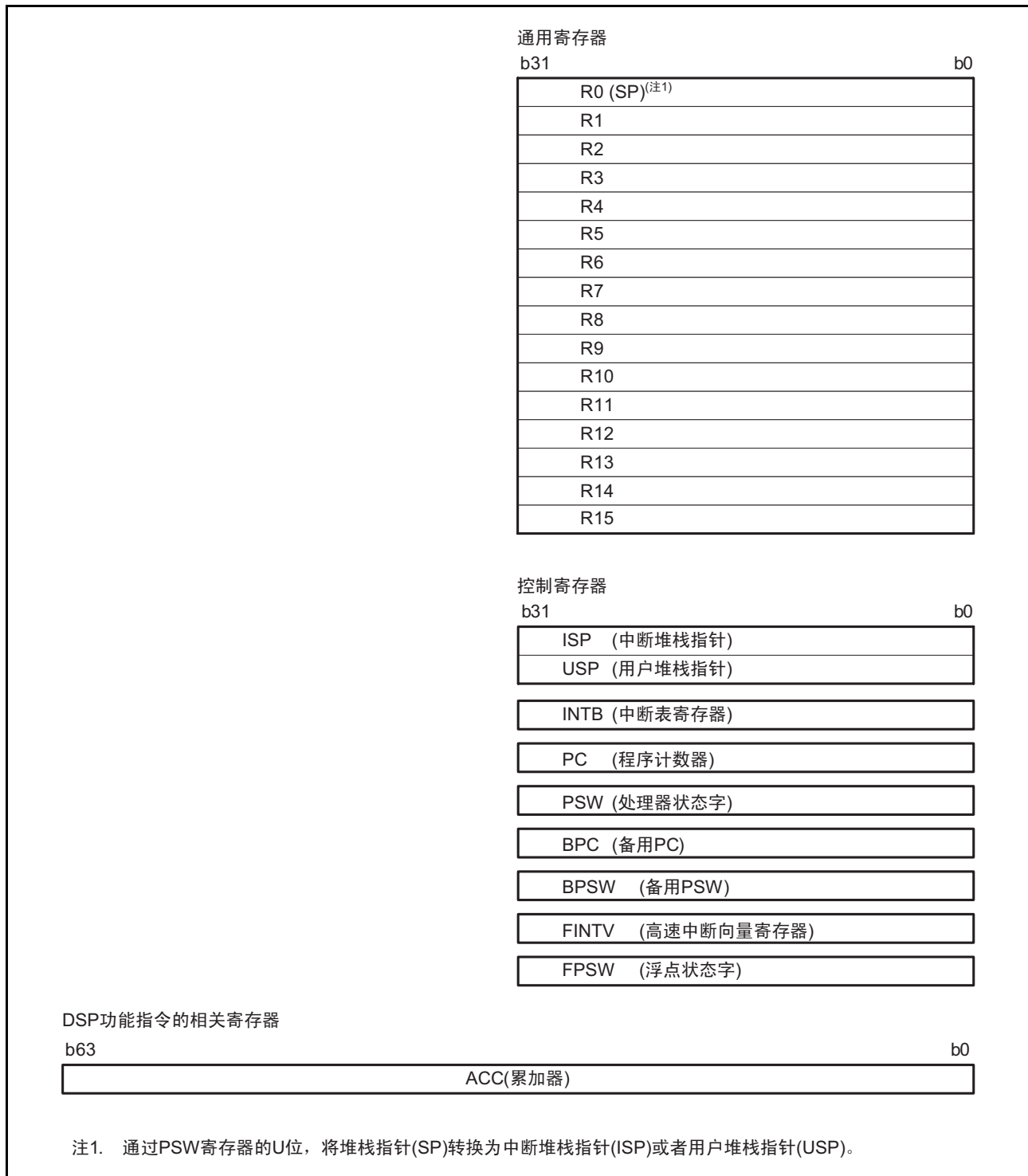


图 2.1 CPU 寄存器组

2.2.1 通用寄存器 (R0 ~ R15)

通用寄存器有 16 个 (R0 ~ R15)，用作数据寄存器和地址寄存器。

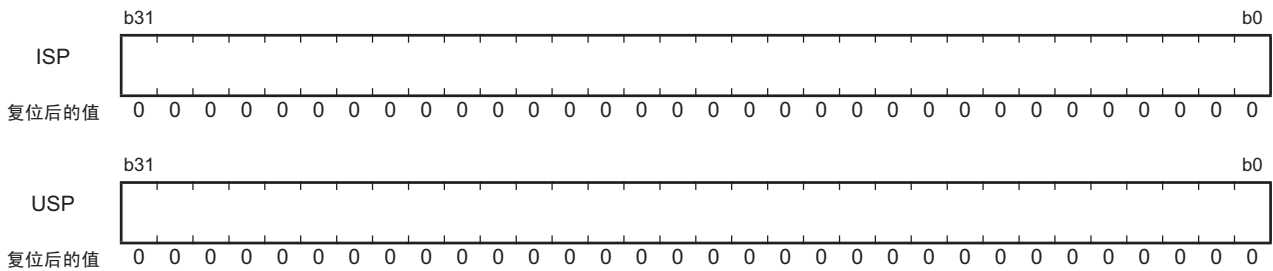
通用寄存器 R0 除了具有通用寄存器的功能以外，还有堆栈指针 (SP) 的功能。通过处理器状态字 (PSW) 的堆栈指针指定位 (U)，将 SP 转换为中断堆栈指针 (ISP) 或者用户堆栈指针 (USP)。

2.2.2 控制寄存器

控制寄存器有以下 9 个：

- 中断堆栈指针 (ISP)
- 用户堆栈指针 (USP)
- 中断表寄存器 (INTB)
- 程序计数器 (PC)
- 处理器状态字 (PSW)
- 备用 PC (BPC)
- 备用 PSW (BPSW)
- 高速中断向量寄存器 (FINTV)
- 浮点状态字 (FPSW)

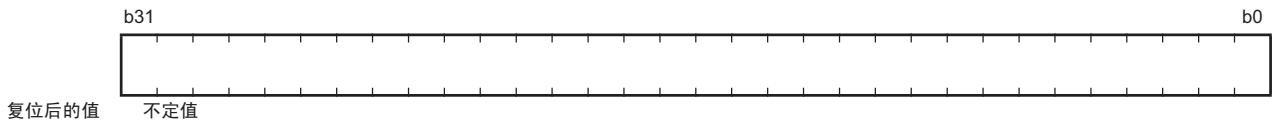
2.2.2.1 中断堆栈指针 (ISP) / 用户堆栈指针 (USP)



堆栈指针 (SP) 有中断堆栈指针 (ISP) 和用户堆栈指针 (USP) 两种，通过处理器状态字 (PSW) 的堆栈指针指定位 (U) 转换要使用的堆栈指针 (ISP/USP)。

如果给 ISP 和 USP 设定 4 的倍数，带有堆栈操作的指令和中断响应顺序的周期数就会变短。

2.2.2.2 中断表寄存器 (INTB)



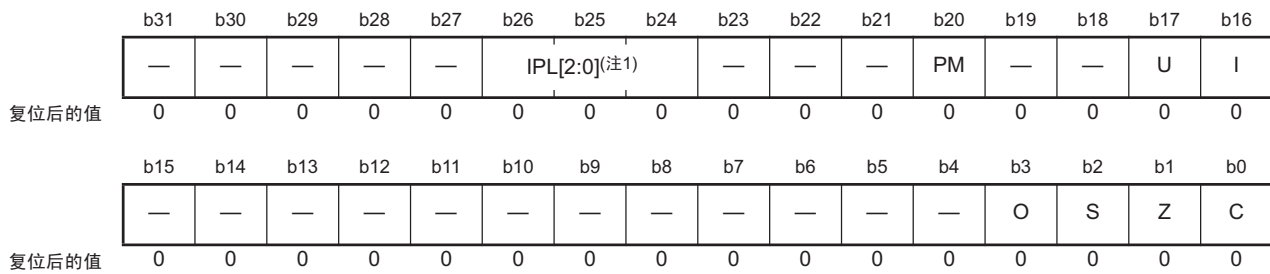
必须给中断表寄存器 (INTB) 设定可变向量表的起始地址。

2.2.2.3 程序计数器 (PC)



程序计数器 (PC) 表示正在执行的指令的地址。

2.2.2.4 处理器状态字 (PSW)



注1. RX610群不支持MVTIPL指令。
必须使用MVTC指令写PSW.IPL[2:0]位。

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|---------------------|----------|---|-----|
| b0 | C | 进位标志 | 0: 未发生进位 1: 发生进位 | R/W |
| b1 | Z | 零标志 | 0: 运算结果不为“0” 1: 运算结果为“0” | R/W |
| b2 | S | 符号标志 | 0: 运算结果为正数或者“0” 1: 运算结果为负数 | R/W |
| b3 | O | 上溢标志 | 0: 未发生上溢 1: 发生上溢 | R/W |
| b15-b4 | — | 保留位 | 读写值都为“0”。 | R/W |
| b16 | I (注1) | 中断允许位 | 0: 禁止中断 1: 允许中断 | R/W |
| b17 | U (注1) | 堆栈指针指定位 | 0: 指定中断堆栈指针 (ISP) 1: 指定用户堆栈指针 (USP) | R/W |
| b19-b18 | — | 保留位 | 读写值都为“0”。 | R/W |
| b20 | PM (注1、注2、注3) | 处理器模式设定位 | 0: 设定为管理模式 1: 设定为用户模式 | R/W |
| b23-b21 | — | 保留位 | 读写值都为“0”。 | R/W |
| b26-b24 | IPL[2:0] (注1、注4) | 处理器中断优先级 | b26 b24 0 0 0: 0级 (最低) 0 0 1: 1级 0 1 0: 2级 0 1 1: 3级 1 0 0: 4级 1 0 1: 5级 1 1 0: 6级 1 1 1: 7级 (最高) | R/W |
| b31-b27 | — | 保留位 | 读写值都为“0”。 | R/W |

- 注1. 当设定为用户模式时, 忽视用 MVTC 指令和 POPC 指令对 IPL[2:0] 位、PM 位、U 位和 I 位的写操作。
注2. 当设定为管理模式时, 忽视用 MVTC 指令和 POPC 指令对 PM 位的写操作, 但是能写除 PM 位以外的其他位。
注3. 要从管理模式转换为用户模式时, 必须在将被压栈的 PSW.PM 位置“1”后执行 RTE 指令、或者在将 BPSW.PM 位置“1”后执行 RTFI 指令。
注4. RX610 群不支持 MVTIPL 指令。必须使用 MVTC 指令写 PSW.IPL[2:0] 位。

处理器状态字 (PSW) 表示指令的执行结果和 CPU 的状态。

C 标志（进位标志）

表示运算结果发生进位、借位或者移出。

Z 标志（零标志）

表示运算结果为“0”。

S 标志（符号标志）

表示运算结果为负数。

O 标志（上溢标志）

表示运算中发生上溢。

I 位（中断允许位）

此位是允许接受中断请求的位。如果接受异常处理，此位就变为“0”。

U 位（堆栈指针指定位）

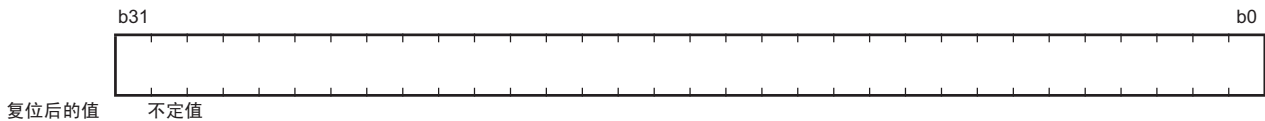
此位是指定要使用的堆栈指针（ISP/USP）的位。如果接受异常处理，此位就变为“0”。如果从管理模式转移到用户模式，此位就变为“1”。

PM 位（处理器模式设定位）

此位是设定处理器模式的位。如果接受异常处理，此位就变为“0”。

IPL[2:0] 位（处理器中断优先级）

IPL[2:0] 位指定 0 级（最低）～7 级（最高）的 8 个处理器中断优先级。如果发生请求的中断的优先级高于处理器中断优先级，就允许该中断。在将 IPL[2:0] 位设定为“7 级”（111b）时，禁止全部的中断。如果发生非屏蔽中断，IPL[2:0] 位就变为“7 级”（111b）。如果发生中断，这些位就为所接受中断的优先级。

2.2.2.5 备用 PC（BPC）

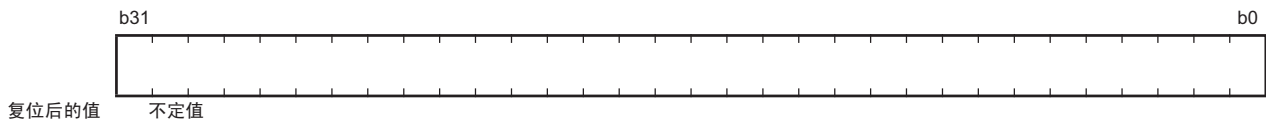
备用 PC（BPC）是为实现中断响应高速化而设计的寄存器。如果发生高速中断，就将程序计数器（PC）的内容保存到 BPC。

2.2.2.6 备用 PSW（BPSW）

备用 PSW（BPSW）是为实现中断响应高速化而设计的寄存器。

如果发生高速中断，就将处理器状态字（PSW）的内容保存到 BPSW。BPSW 的位配置与 PSW 对应。

2.2.2.7 高速中断向量寄存器 (FINTV)



高速中断向量寄存器 (FINTV) 是为实现中断响应高速化而设计的寄存器。必须设定发生高速中断时的转移目标地址。

2.2.2.8 浮点状态字 (FPSW)

| | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---------|-----|
| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| | FS | FX | FU | FZ | FO | FV | — | — | — | — | — | — | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | EX | EU | EZ | EO | EV | — | DN | CE | CX | CU | CZ | CO | CV | RM[1:0] | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|--------------|---|----------------|
| b1-b0 | RM[1:0] | 浮点舍入模式设定位 | b1 b0 0 0: 向最接近的值舍入 0 1: 向 0 方向舍入 1 0: 向 +∞ 方向舍入 1 1: 向 -∞ 方向舍入 | R/W |
| b2 | CV | 无效运算源标志 | 0: 未发生无效运算 1: 发生无效运算 | R/(W) (注 1) |
| b3 | CO | 上溢源标志 | 0: 未发生上溢 1: 发生上溢 | R/(W) (注 1) |
| b4 | CZ | 被零除源标志 | 0: 未发生被零除 1: 发生被零除 | R/(W) (注 1) |
| b5 | CU | 下溢源标志 | 0: 未发生下溢 1: 发生下溢 | R/(W) (注 1) |
| b6 | CX | 精度异常源标志 | 0: 未发生精度异常 1: 发生精度异常 | R/(W) (注 1) |
| b7 | CE | 非安装处理源标志 | 0: 未发生非安装处理 1: 发生非安装处理 | R/(W) (注 1) |
| b8 | DN | 非规格化数的 0 刷新位 | 0: 将非规格化数作为非规格化数处理 1: 将非规格化数作为 0 处理 (注 2) | R/W |
| b9 | — | 保留位 | 读写值都为“0”。 | R/W |
| b10 | EV | 无效运算异常处理允许位 | 0: 禁止无效运算引起的异常处理 1: 允许无效运算引起的异常处理 | R/W |
| b11 | EO | 上溢异常处理允许位 | 0: 禁止上溢引起的异常处理 1: 允许上溢引起的异常处理 | R/W |
| b12 | EZ | 被零除异常处理允许位 | 0: 禁止被零除引起的异常处理 1: 允许被零除引起的异常处理 | R/W |

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|---------|-----------|--------------------------------------|-----|
| b13 | EU | 下溢异常处理允许位 | 0: 禁止下溢引起的异常处理 1: 允许下溢引起的异常处理 | R/W |
| b14 | EX | 精度异常处理允许位 | 0: 禁止精度异常引起的异常处理 1: 允许精度异常引起的异常处理 | R/W |
| b25-b15 | — | 保留位 | 读写值都为“0”。 | R/W |
| b26 | FV (注3) | 无效运算标志 | 0: 未发生无效运算 1: 发生无效运算 (注8) | R/W |
| b27 | FO (注4) | 上溢标志 | 0: 未发生上溢 1: 发生上溢 (注8) | R/W |
| b28 | FZ (注5) | 被零除标志 | 0: 未发生被零除 1: 发生被零除 (注8) | R/W |
| b29 | FU (注6) | 下溢标志 | 0: 未发生下溢 1: 发生下溢 (注8) | R/W |
| b30 | FX (注7) | 精度异常标志 | 0: 未发生精度异常 1: 发生精度异常 (注8) | R/W |
| b31 | FS | 浮点错误概要标志 | 反映FU、FZ、FO、FV标志的逻辑或。 | R |

注1. 如果写“0”，此位就变为“0”。如果写“1”，就保持原来的值。

注2. 正的非规格化数作为+0处理，负的非规格化数作为-0处理。

注3. 当EV位为“0”时，FV标志有效。

注4. 当EO位为“0”时，FO标志有效。

注5. 当EZ位为“0”时，FZ标志有效。

注6. 当EU位为“0”时，FU标志有效。

注7. 当EX位为“0”时，FX标志有效。

注8. 一旦该位变为“1”，就在通过软件置“0”前一直保持“1”。

浮点状态字 (FPSW) 表示浮点运算结果。

如果通过异常处理允许位 E_j 允许异常处理 ($E_j=1$)，就能通过异常处理程序检查对应的 C_j 标志，判断异常的发生源。如果禁止异常处理 ($E_j=0$)，就能在一连串处理的最后检查 F_j 标志，确认是否发生异常。 F_j 标志是累积标志 ($j=X,U,Z,O,V$)。

RM[1:0] 位 (浮点舍入模式设定位)

这些位是设定浮点舍入模式的位。

【浮点舍入模式的说明】

- 向最接近的值舍入 (默认) : 向接近以无限有效位数进行计算时的结果的值舍入。
如果为中间值，就向结果为偶数的方向舍入。
- 向0方向舍入 : 向结果的绝对值变小的方向舍入 (单纯的舍去)。
- 向 $+\infty$ 方向舍入 : 向结果值变大的方向舍入。
- 向 $-\infty$ 方向舍入 : 向结果值变小的方向舍入。

1. “向最接近的值舍入”是默认模式，返回最正确的值。

2. “向0方向舍入”、“向 $+\infty$ 方向舍入”、“向 $-\infty$ 方向舍入”用于保证使用了区间运算 (Interval arithmetic) 的精度。

CV 标志（无效运算源标志）、CO 标志（上溢源标志）

CZ 标志（被零除源标志）、CU 标志（下溢源标志）

CX 标志（精度异常源标志）、CE 标志（非安装处理源标志）

除 IEEE754 规格规定的 5 种异常（上溢、下溢、精度异常、被零除、无效运算）以外，如果发生非安装处理，对应的标志就变为“1”。

- 在对应的标志为“1”的情况下执行 FPU 运算指令时，该标志就变为“0”。
- 如果使用 MVTTC 指令和 POPC 指令给对应的标志写“0”，该标志就变为“0”；如果写“1”，就保持原来的值。

DN 位（非规格化数的 0 刷新位）

当此位为“0”时，将非规格化数作为非规格化数处理；当此位为“1”时，将非规格化数作为 0 处理。

EV 位（无效运算异常处理允许位）、EO 位（上溢异常处理允许位）

EZ 位（被零除异常处理允许位）、EU 位（下溢异常处理允许位）

EX 位（精度异常处理允许位）

由于执行 FPU 运算指令而发生了 IEEE754 规格规定的 5 种异常时，控制 CPU 是否转移到异常处理。

如果此位为“0”，就禁止异常处理；如果为“1”，就允许异常处理。

FV 标志（无效运算标志）、FO 标志（上溢标志）、FZ 标志（被零除标志）

FU 标志（下溢标志）、FX 标志（精度异常标志）

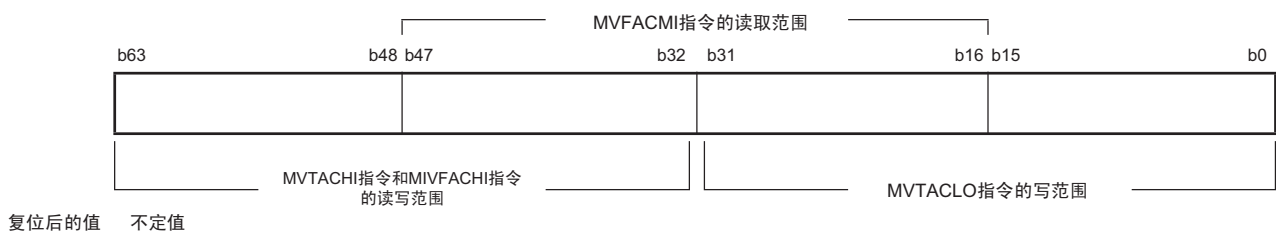
在异常处理允许位 E_j 为“0”（禁止异常处理）时，如果发生 IEEE754 规格规定的 5 种异常，对应的标志就变为“1”。

- 当 E_j 为“1”（允许异常处理）时，此标志不变。
- 如果该标志变为“1”，就在通过软件置“0”前一直保持“1”（累积标志）。

FS 标志（浮点错误概要标志）

此标志反映 FU、FZ、FO、FV 标志的逻辑或。

2.2.2.9 累加器（ACC）



累加器（ACC）是 64 位寄存器，用于 DSP 功能指令，也用于乘法指令（EMUL、EMULU、FMUL、MUL）和乘加运算指令（RMPA）。在执行这些指令时，ACC 的值被更改。

使用 MVTACHI 指令和 MVTACLO 指令写 ACC。通过 MVTACHI 指令将数据写到高 32 位（b63 ~ b32），通过 MVTACLO 指令，将数据写到低 32 位（b31 ~ b0）。

使用 MVFACHI 指令和 MVFACMI 指令读 ACC。通过 MVFACHI 指令读高 32 位（b63 ~ b32）的数据，通过 MVFACMI 指令读中间 32 位（b47 ~ b16）的数据。

2.3 处理器模式

RX CPU 有管理模式和用户模式两种处理器模式。能通过使用处理器模式，实现对 CPU 资源的阶层保护结构。

各处理器模式对能执行的指令以及能存取的 CPU 资源规定了权限，管理模式的权限高于用户模式。复位后，以管理模式运行。

2.3.1 管理模式

在管理模式中，能存取全部的 CPU 资源，还能执行全部的指令。但是，忽视通过 MVTC 指令和 POPC 指令写处理器状态字（PSW）的处理器模式设定位（PM）。有关写 PM 位的方法，请参照“2.2.2.4 处理器状态字（PSW）”。

2.3.2 用户模式

在用户模式中，限制部分 CPU 资源的写存取。被限制写存取的 CPU 资源如下，限制对象为全部指令的存取。

- 处理器状态字（PSW）的部分位（IPL[2:0]、PM、U、I）
- 中断堆栈指针（ISP）
- 中断表寄存器（INTB）
- 备用 PSW（BPSW）
- 备用 PC（BPC）
- 高速中断向量寄存器（FINTV）

2.3.3 特权指令

特权指令是只能在管理模式中执行的指令。如果在用户模式中执行特权指令，就会发生特权指令异常。特权指令有 RTFI、RTE、WAIT 指令。

2.3.4 处理器模式之间的转移

通过处理器状态字（PSW）的处理器模式设定位（PM）转换处理器模式。但是，通过 MVTC 指令和 POPC 指令对 PM 位的改写无效。必须通过以下所示的方法进行转换。

(1) 用户模式向管理模式的转移

如果发生异常，PSW 的 PM 位就变为“0”，CPU 转移到管理模式。在管理模式中执行硬件的预处理。被保存的 PSW 的 PM 位保持发生异常前的处理器模式。

(2) 管理模式向用户模式的转移

在被压栈的 PSW 的 PM 位为“1”时执行 RTE 指令，或者在被保存到备用 PSW（BPSW）中的 PSW 的 PM 位为“1”时执行 RTFI 指令，向用户模式转移。一旦转移到用户模式，PSW 的堆栈指针指定位（U）就变为“1”。

2.4 数据类型

RX CPU 能处理整数、浮点数、位、字符串共 4 种数据。

2.4.1 整数

整数有带符号整数和不带符号整数，带符号整数的负值用 2 的补数表现。

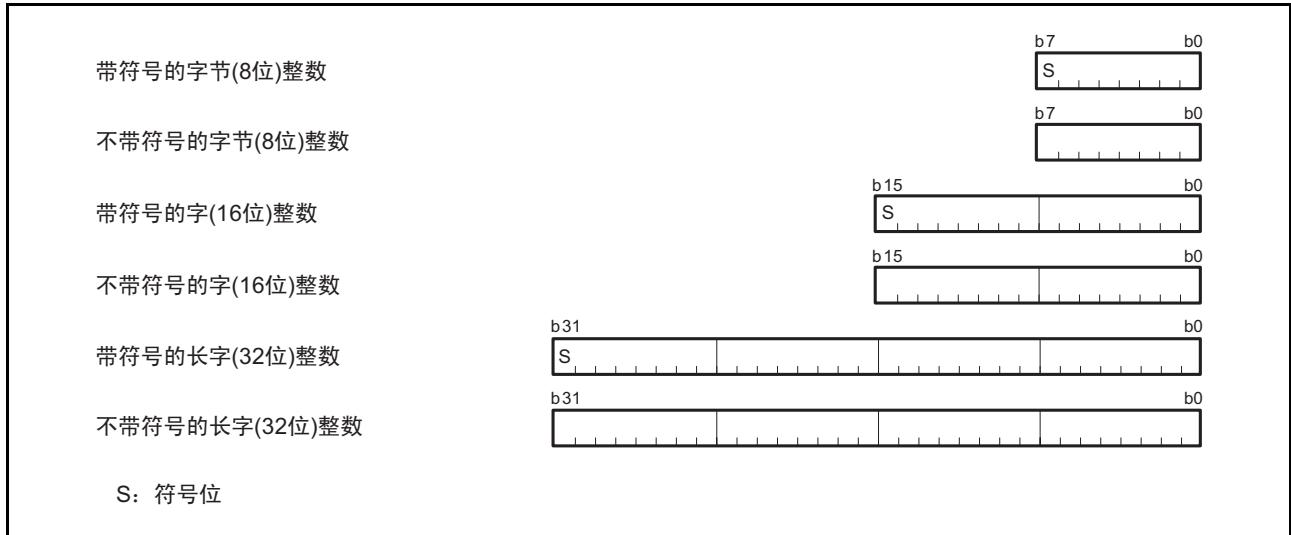


图 2.2 整数

2.4.2 浮点数

浮点数对应 IEEE754 规定的单精度浮点数。浮点数能用于浮点运算指令 FADD、FCMP、FDIV、FMUL、FSUB、FTOI、ITOF、ROUND 共 8 种指令。

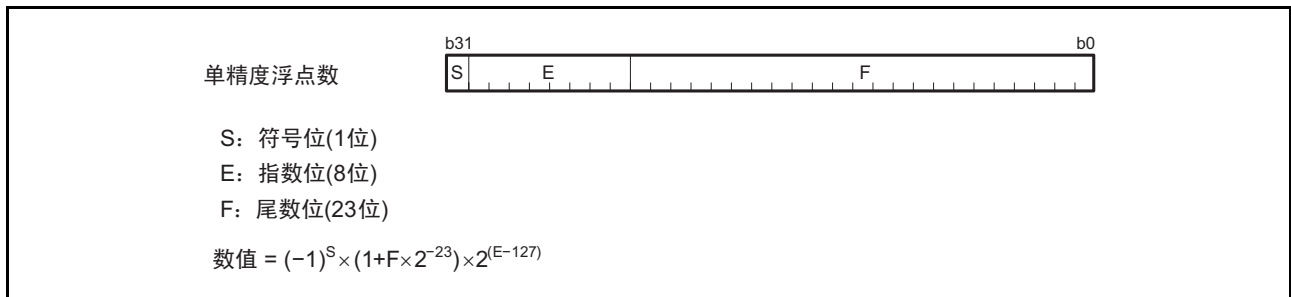


图 2.3 浮点数

浮点数支持以下数值：

- 0<E<255 （规格化数-Normal Numbers）
- E=0 并且 F=0 （零-Signed Zero）
- E=0 并且 F>0 （非规格化数-Subnormal Numbers）

注 1. 当 FPSW.DN 位为 “1” 时，作为 0 处理；当 DN 位为 “0” 时，发生非安装处理。

- E=255 并且 F=0 （无穷大-Infinity）
- E=255 并且 F>0 （非数值-NaN: Not a Number）

2.4.3 位

位用于位操作指令 **BCLR**、**BMCnd**、**BNOT**、**BSET**、**BTST** 共 5 种指令。

通过对象寄存器和 31 ~ 0 的位号指定寄存器的位。

通过对象地址和 7 ~ 0 的位号指定存储器的位。能用于地址指定的寻址方式有寄存器间接和寄存器相对两种。

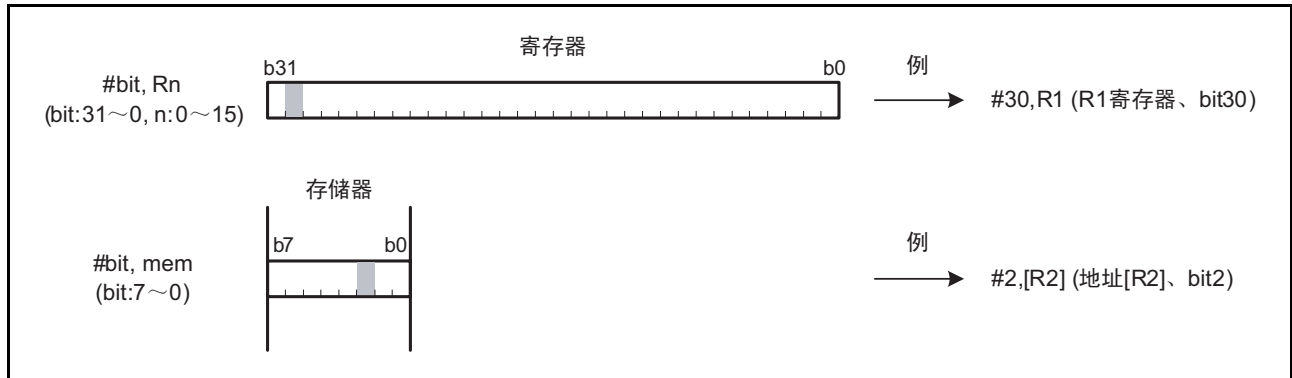


图 2.4 位

2.4.4 字符串

字符串是指只连续排列任意个数的字节（8 位）、字（16 位）或者长字（32 位）数据的数据类型。字符串能用于字符串操作指令 **SCMPU**、**SMOVB**、**SMOVF**、**SMOVU**、**SSTR**、**SUNTIL**、**SWHILE** 共 7 种指令。

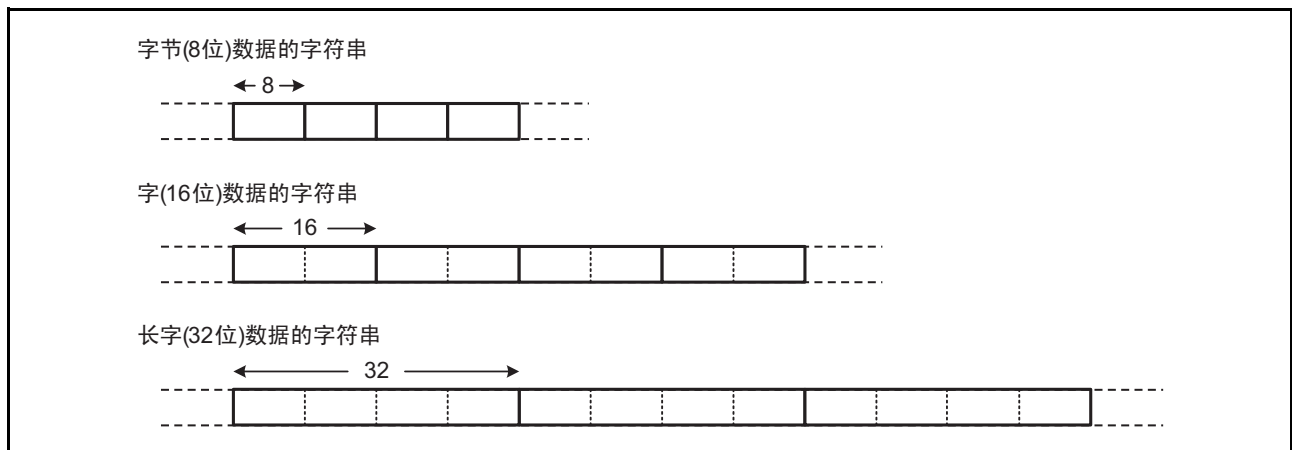


图 2.5 字符串

2.5 字节序

数据排列能选择小端法或者大端法。

2.5.1 字节序的转换

在 RX610 群中，能使用高位字节（MSB）为地址 0 的大端法和低位字节（LSB）为地址 0 的小端法两种字节数据的排列方法。

通过模式引脚（MDE）转换字节序。有关模式引脚的设定，请参照“3. 运行模式”。

使用指令选择 8 位、16 位或者 32 位的存取，并且存取因小端法或者大端法的设定而不同，各存取如表 2.1 ~ 表 2.12 所示。

表中的

LL 表示通用寄存器的 D7 ~ D0；

LH 表示通用寄存器的 D15 ~ D8；

HL 表示通用寄存器的 D23 ~ D16；

HH 表示通用寄存器的 D31 ~ D24。

| | | | | |
|----------|-----------|-----------|----------|---------|
| | D31 ~ D24 | D23 ~ D16 | D15 ~ D8 | D7 ~ D0 |
| 通用寄存器 Rm | HH | HL | LH | LL |

表 2.1 设定为小端法时的 32 位读操作

| 操作 地址 src | 用 32 位读地址 0 | 用 32 位读地址 1 | 用 32 位读地址 2 | 用 32 位读地址 3 | 用 32 位读地址 4 |
|--------------|-------------|-------------|-------------|-------------|-------------|
| 地址 0 | 传送到 LL | — | — | — | — |
| 地址 1 | 传送到 LH | 传送到 LL | — | — | — |
| 地址 2 | 传送到 HL | 传送到 LH | 传送到 LL | — | — |
| 地址 3 | 传送到 HH | 传送到 HL | 传送到 LH | 传送到 LL | — |
| 地址 4 | — | 传送到 HH | 传送到 HL | 传送到 LH | 传送到 LL |
| 地址 5 | — | — | 传送到 HH | 传送到 HL | 传送到 LH |
| 地址 6 | — | — | — | 传送到 HH | 传送到 HL |
| 地址 7 | — | — | — | — | 传送到 HH |

表 2.2 设定为大端法时的 32 位读操作

| 操作 地址 src | 用 32 位读地址 0 | 用 32 位读地址 1 | 用 32 位读地址 2 | 用 32 位读地址 3 | 用 32 位读地址 4 |
|--------------|-------------|-------------|-------------|-------------|-------------|
| 地址 0 | 传送到 HH | — | — | — | — |
| 地址 1 | 传送到 HL | 传送到 HH | — | — | — |
| 地址 2 | 传送到 LH | 传送到 HL | 传送到 HH | — | — |
| 地址 3 | 传送到 LL | 传送到 LH | 传送到 HL | 传送到 HH | — |
| 地址 4 | — | 传送到 LL | 传送到 LH | 传送到 HL | 传送到 HH |
| 地址 5 | — | — | 传送到 LL | 传送到 LH | 传送到 HL |
| 地址 6 | — | — | — | 传送到 LL | 传送到 LH |
| 地址 7 | — | — | — | — | 传送到 LL |

表 2.3 设定为小端法时的 32 位写操作

| 操作 地址 dest | 用 32 位写地址 0 | 用 32 位写地址 1 | 用 32 位写地址 2 | 用 32 位写地址 3 | 用 32 位写地址 4 |
|---------------|-------------|-------------|-------------|-------------|-------------|
| 地址 0 | 从 LL 传送 | — | — | — | — |
| 地址 1 | 从 LH 传送 | 从 LL 传送 | — | — | — |
| 地址 2 | 从 HL 传送 | 从 LH 传送 | 从 LL 传送 | — | — |
| 地址 3 | 从 HH 传送 | 从 HL 传送 | 从 LH 传送 | 从 LL 传送 | — |
| 地址 4 | — | 从 HH 传送 | 从 HL 传送 | 从 LH 传送 | 从 LL 传送 |
| 地址 5 | — | — | 从 HH 传送 | 从 HL 传送 | 从 LH 传送 |
| 地址 6 | — | — | — | 从 HH 传送 | 从 HL 传送 |
| 地址 7 | — | — | — | — | 从 HH 传送 |

表 2.4 设定为大端法时的 32 位写操作

| 操作 地址 dest | 用 32 位写地址 0 | 用 32 位写地址 1 | 用 32 位写地址 2 | 用 32 位写地址 3 | 用 32 位写地址 4 |
|---------------|-------------|-------------|-------------|-------------|-------------|
| 地址 0 | 从 HH 传送 | — | — | — | — |
| 地址 1 | 从 HL 传送 | 从 HH 传送 | — | — | — |
| 地址 2 | 从 LH 传送 | 从 HL 传送 | 从 HH 传送 | — | — |
| 地址 3 | 从 LL 传送 | 从 LH 传送 | 从 HL 传送 | 从 HH 传送 | — |
| 地址 4 | — | 从 LL 传送 | 从 LH 传送 | 从 HL 传送 | 从 HH 传送 |
| 地址 5 | — | — | 从 LL 传送 | 从 LH 传送 | 从 HL 传送 |
| 地址 6 | — | — | — | 从 LL 传送 | 从 LH 传送 |
| 地址 7 | — | — | — | — | 从 LL 传送 |

表 2.5 设定为小端法时的 16 位读操作

| 操作 地址 src | 用 16 位 读地址 0 | 用 16 位 读地址 1 | 用 16 位 读地址 2 | 用 16 位 读地址 3 | 用 16 位 读地址 4 | 用 16 位 读地址 5 | 用 16 位 读地址 6 |
|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 地址 0 | 传送到 LL | — | — | — | — | — | — |
| 地址 1 | 传送到 LH | 传送到 LL | — | — | — | — | — |
| 地址 2 | — | 传送到 LH | 传送到 LL | — | — | — | — |
| 地址 3 | — | — | 传送到 LH | 传送到 LL | — | — | — |
| 地址 4 | — | — | — | 传送到 LH | 传送到 LL | — | — |
| 地址 5 | — | — | — | — | 传送到 LH | 传送到 LL | — |
| 地址 6 | — | — | — | — | — | 传送到 LH | 传送到 LL |
| 地址 7 | — | — | — | — | — | — | 传送到 LH |

表 2.6 设定为大端法时的 16 位读操作

| 操作 地址 src | 用 16 位 读地址 0 | 用 16 位 读地址 1 | 用 16 位 读地址 2 | 用 16 位 读地址 3 | 用 16 位 读地址 4 | 用 16 位 读地址 5 | 用 16 位 读地址 6 |
|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 地址 0 | 传送到 LH | — | — | — | — | — | — |
| 地址 1 | 传送到 LL | 传送到 LH | — | — | — | — | — |
| 地址 2 | — | 传送到 LL | 传送到 LH | — | — | — | — |
| 地址 3 | — | — | 传送到 LL | 传送到 LH | — | — | — |
| 地址 4 | — | — | — | 传送到 LL | 传送到 LH | — | — |
| 地址 5 | — | — | — | — | 传送到 LL | 传送到 LH | — |
| 地址 6 | — | — | — | — | — | 传送到 LL | 传送到 LH |
| 地址 7 | — | — | — | — | — | — | 传送到 LL |

表 2.7 设定为小端法时的 16 位写操作

| 操作 地址 dest | 用 16 位 写地址 0 | 用 16 位 写地址 1 | 用 16 位 写地址 2 | 用 16 位 写地址 3 | 用 16 位 写地址 4 | 用 16 位 写地址 5 | 用 16 位 写地址 6 |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 地址 0 | 从 LL 传送 | — | — | — | — | — | — |
| 地址 1 | 从 LH 传送 | 从 LL 传送 | — | — | — | — | — |
| 地址 2 | — | 从 LH 传送 | 从 LL 传送 | — | — | — | — |
| 地址 3 | — | — | 从 LH 传送 | 从 LL 传送 | — | — | — |
| 地址 4 | — | — | — | 从 LH 传送 | 从 LL 传送 | — | — |
| 地址 5 | — | — | — | — | 从 LH 传送 | 从 LL 传送 | — |
| 地址 6 | — | — | — | — | — | 从 LH 传送 | 从 LL 传送 |
| 地址 7 | — | — | — | — | — | — | 从 LH 传送 |

表 2.8 设定为大端法时的 16 位写操作

| 操作 地址 dest | 用 16 位 写地址 0 | 用 16 位 写地址 1 | 用 16 位 写地址 2 | 用 16 位 写地址 3 | 用 16 位 写地址 4 | 用 16 位 写地址 5 | 用 16 位 写地址 6 |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 地址 0 | 从 LH 传送 | — | — | — | — | — | — |
| 地址 1 | 从 LL 传送 | 从 LH 传送 | — | — | — | — | — |
| 地址 2 | — | 从 LL 传送 | 从 LH 传送 | — | — | — | — |
| 地址 3 | — | — | 从 LL 传送 | 从 LH 传送 | — | — | — |
| 地址 4 | — | — | — | 从 LL 传送 | 从 LH 传送 | — | — |
| 地址 5 | — | — | — | — | 从 LL 传送 | 从 LH 传送 | — |
| 地址 6 | — | — | — | — | — | 从 LL 传送 | 从 LH 传送 |
| 地址 7 | — | — | — | — | — | — | 从 LL 传送 |

表 2.9 设定为小端法时的 8 位读操作

| 操作 地址 src | 用 8 位读地址 0 | 用 8 位读地址 1 | 用 8 位读地址 2 | 用 8 位读地址 3 |
|--------------|------------|------------|------------|------------|
| 地址 0 | 传送到 LL | — | — | — |
| 地址 1 | — | 传送到 LL | — | — |
| 地址 2 | — | — | 传送到 LL | — |
| 地址 3 | — | — | — | 传送到 LL |

表 2.10 设定为大端法时的 8 位读操作

| 操作 地址 src | 用 8 位读地址 0 | 用 8 位读地址 1 | 用 8 位读地址 2 | 用 8 位读地址 3 |
|--------------|------------|------------|------------|------------|
| 地址 0 | 传送到 LL | — | — | — |
| 地址 1 | — | 传送到 LL | — | — |
| 地址 2 | — | — | 传送到 LL | — |
| 地址 3 | — | — | — | 传送到 LL |

表 2.11 设定为小端法时的 8 位写操作

| 操作 地址 dest | 用 8 位写地址 0 | 用 8 位写地址 1 | 用 8 位写地址 2 | 用 8 位写地址 3 |
|---------------|------------|------------|------------|------------|
| 地址 0 | 从 LL 传送 | — | — | — |
| 地址 1 | — | 从 LL 传送 | — | — |
| 地址 2 | — | — | 从 LL 传送 | — |
| 地址 3 | — | — | — | 从 LL 传送 |

表 2.12 设定为大端法时的 8 位写操作

| 操作 地址 dest | 用 8 位写地址 0 | 用 8 位写地址 1 | 用 8 位写地址 2 | 用 8 位写地址 3 |
|---------------|------------|------------|------------|------------|
| 地址 0 | 从 LL 传送 | — | — | — |
| 地址 1 | — | 从 LL 传送 | — | — |
| 地址 2 | — | — | 从 LL 传送 | — |
| 地址 3 | — | — | — | 从 LL 传送 |

2.5.2 I/O 寄存器的存取

I/O 寄存器分配在固定地址，与 MDE 引脚设定的大端法或者小端法无关。因此，I/O 寄存器的存取不受字节序变更的影响。有关 I/O 寄存器的分配，请参照各章的寄存器说明。

2.5.3 存取 I/O 寄存器时的注意事项

必须按照以下的规则存取 I/O 寄存器：

- 8 位总线宽度的 I/O 寄存器必须使用 8 位总线宽度的指令。
- 16 位总线宽度的 I/O 寄存器必须使用 16 位总线宽度的指令。
- 32 位总线宽度的 I/O 寄存器必须使用 32 位总线宽度的指令。

2.5.4 数据排列

2.5.4.1 寄存器的数据排列

寄存器的数据长度和位号的关系如图 2.6 所示。

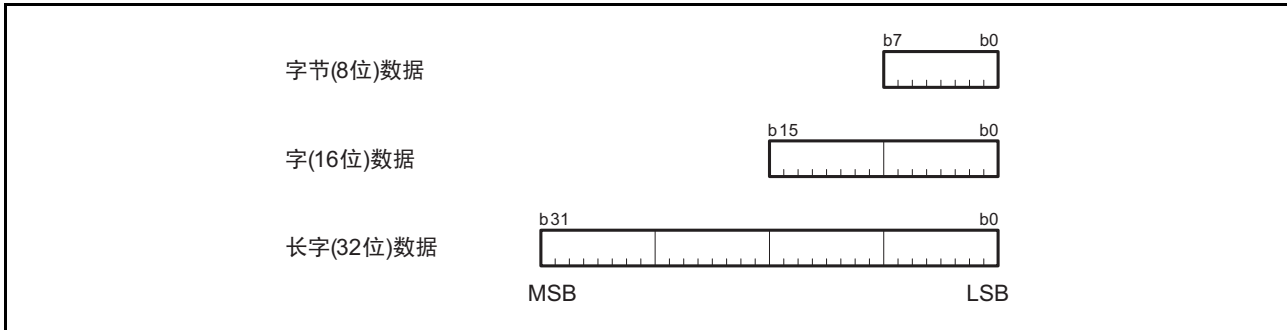


图 2.6 寄存器的数据排列

2.5.4.2 存储器的数据排列

存储器的数据长度有字节（8 位）、字（16 位）和长字（32 位）共 3 种，能选择小端法或者大端法的数据排列方式。存储器的数据排列如图 2.7 所示。

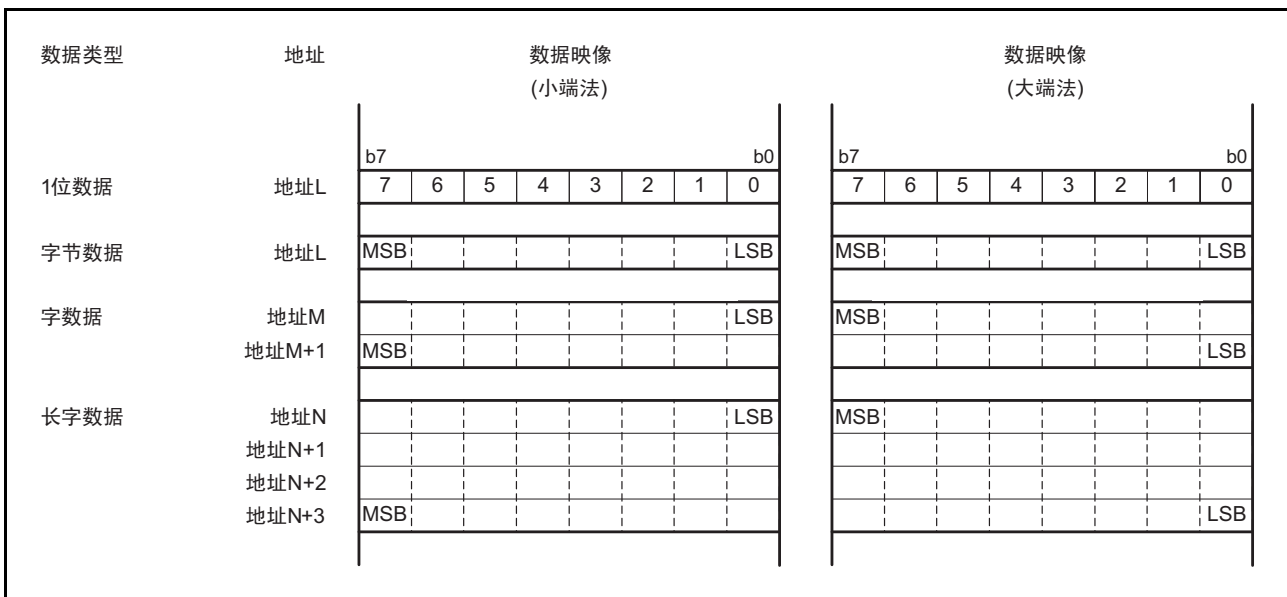


图 2.7 存储器的数据排列

2.6 向量表

向量表有固定向量表和可变向量表。向量表的 1 个向量由 4 字节构成，各向量设有对应的异常处理程序的起始地址。

2.6.1 固定向量表

固定向量表是表的分配地址被固定的向量表。特权指令异常、未定义指令异常、浮点异常、非屏蔽中断、复位的各向量分配在地址 FFFFFFF80h ~ FFFFFFFFh。固定向量表如图 2.8 所示。

| | MSB | LSB |
|------------|---------|-----|
| FFFFFFF80h | (保留区) | |
| : | : | |
| FFFFFFFCCh | (保留区) | |
| FFFFFFFD0h | 特权指令异常 | |
| FFFFFFFD4h | (保留区) | |
| FFFFFFFD8h | (保留区) | |
| FFFFFFDCh | 未定义指令异常 | |
| FFFFFFE0h | (保留区) | |
| FFFFFFE4h | 浮点异常 | |
| FFFFFFE8h | (保留区) | |
| FFFFFFECh | (保留区) | |
| FFFFFFF0h | (保留区) | |
| FFFFFFF4h | (保留区) | |
| FFFFFFF8h | 非屏蔽中断 | |
| FFFFFFFCh | 复位 | |

图 2.8 固定向量表

2.6.2 可变向量表

可变向量表是能改变表的分配地址的向量表。无条件陷阱和中断的各向量分配在以中断表寄存器 (INTB) 的内容所示的值为起始地址 (IntBase) 的 1024 字节的区域。可变向量表如图 2.9 所示。

可变向量表中的每个向量带有序号 (0 ~ 255)。在无条件陷阱发生源的 INT 指令时，分配了与 INT 指令号 (0 ~ 255) 对应的向量，而在 BRK 指令时，分配了向量号 0 的向量。

另外，在中断源时，分配了各产品规定的向量号 (0 ~ 255)。有关中断的向量号，请参照“10.3.1 中断向量表”。

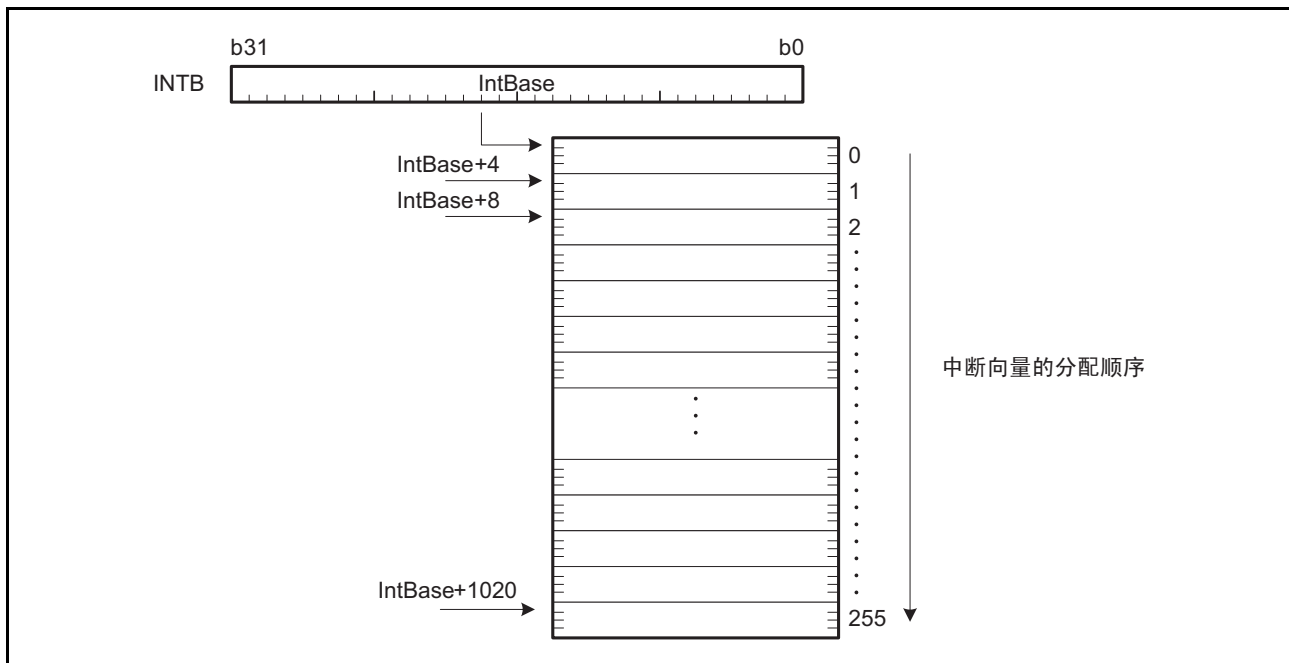


图 2.9 可变向量表

2.7 指令操作

2.7.1 RMPA 指令和字符串操作指令的数据预取

为了从存储器快速读取数据，RMPA 指令和字符串操作指令（SCMPU、SMOVB、SMOVF、SMOVU、SUNTIL、SWHILE。SSTR 指令除外）有可能进行数据预取。对应数据读取位置，能进行最多到该位置之后的 3 字节为止的数据预取。各指令的数据读取位置如下：

- RMPA 指令：R1 指定的被乘数地址以及 R2 指定的乘数地址
- SCMPU 指令：R1 指定的比较源地址以及 R2 指定的比较目标地址
- SUNTIL 指令和 SWHILE 指令：R1 指定的比较目标地址
- SMOVB、SMOVF、SMOVU 指令：R2 指定的传送源地址

2.8 流水线

2.8.1 概要

RX CPU 由 5 个阶段的流水线构成。RX CPU 的指令转换为 1 个或者多个微操作，RX CPU 对微操作进行流水线处理。流水线 IF 阶段以指令为单位进行操作，D 以后的阶段以微操作为单位进行操作。

流水线的操作和各阶段的概要如下所示：

(1) IF 阶段（取指令阶段）

这是从存储器取指令的阶段。RX CPU 有 4 个 8 字节的指令队列，与 D（解码）阶段的解码处理结束无关，继续取指令，直到指令队列满为止。

(2) D 阶段（解码阶段）

D 阶段进行指令的解码处理（DEC），并且将指令转换为微操作。如果在此阶段读寄存器（RF）并且要参照前面指令的运算结果，就进行旁路处理（BYP）。通过旁路处理，能将运算结果写到寄存器（RW），同时在 D 阶段参照寄存器。

(3) E 阶段（执行阶段）

进行运算和地址计算等（OP）。

(4) M 阶段（存储器存取阶段）

此阶段进行操作数的存储器存取（OA1 和 OA2），只在存储器存取时使用此阶段。此阶段又分为 M1 和 M2 两个子阶段。RX CPU 中 M1 阶段和 M2 阶段各有 1 个存储器存取。

- M1 阶段（存储器存取阶段 1）

此阶段进行操作数的存储器存取（OA1）。

存储时：一旦总线接受写请求，就结束流水线处理。

加载时：一旦总线接受读请求，就进入 M2 阶段。如果请求的接受和加载数据的到达同时（无等待的存储器存取），就进入 WB 阶段。

- M2 阶段（存储器存取阶段 2）

此阶段进行操作数的存储器存取（OA2）。这是等待加载数据到达的阶段。一旦加载数据到达，就进入 WB 阶段。

(5) WB 阶段（回写阶段）

将运算结果以及从存储器读取的数据写到寄存器（RW）。能将从存储器读取的数据以及其他运算结果同时（相同周期）写到寄存器。

流水线结构及其运行如图 2.10 所示。

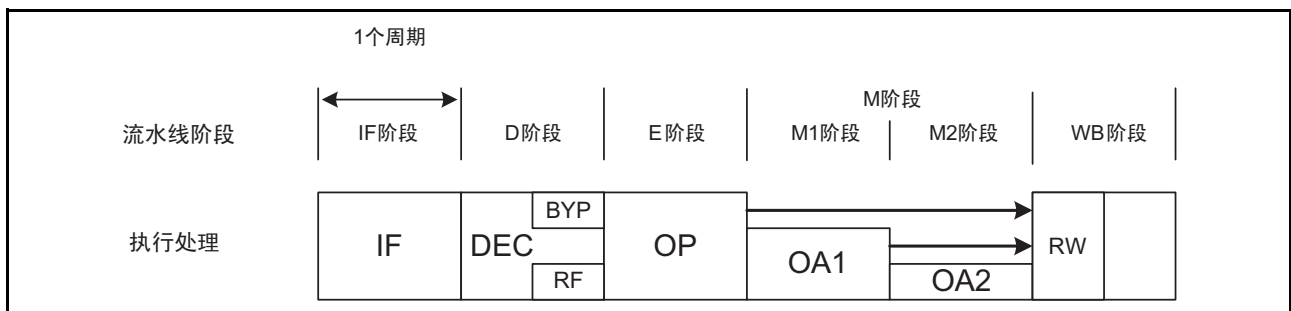


图 2.10 流水线结构及其运行

2.8.2 指令和流水线处理

表中操作数的表示法如下：

#IMM: 立即数

Rs、Rs2、Rd、Rd2、Ri、Rb: 通用寄存器

CR: 控制寄存器

dsp: dsp5、dsp8、dsp16、dsp24

pcdsp: pcdsp3、pcdsp8、pcdsp16、pcdsp24

2.8.2.1 转换为单一微操作的指令和流水线处理

转换为单一微操作的指令如下所示，周期数表示无等待存储器存取时的周期数。

表 2.13 转换为单一微操作的指令

| 指令 | 助记符（当省略指令长度时，为全长度共同的操作） | 参照图 | 周期数 |
|--|---|--------|-----------------------|
| 算术 / 逻辑运算指令 （寄存器之间、立即数 - 寄存器） EMUL、EMULU、RMPA、DIV 和 DIVU 除外 | • {ABS,ADC……（省略）,XOR} “#IMM,Rd” / “Rd” / “Rs,Rd” / “Rs,Rs2,Rd” | 图 2.11 | 1 |
| 算术 / 逻辑运算指令（除法） | • DIV “#IMM,Rd” / “Rs,Rd” | 图 2.11 | 3 ~ 20（注1） |
| | • DIVU “#IMM,Rd” / “Rs,Rd” | 图 2.11 | 2 ~ 18（注1） |
| 传送指令 （寄存器之间、立即数 - 寄存器） | • {MOV,MOVU,REVL,REVV} “#IMM,Rd” / “Rs,Rd” • SCCnd “Rd” • {STNZ,STZ} “#IMM,Rd” | 图 2.11 | 1 |
| 传送指令（加载） | • {MOV,MOVU} “[Rs],Rd” / “dsp[Rs],Rd” / “[Rs+],Rd” / “[−Rs],Rd” / “Rs,[Ri,Rb]” • • POP “Rd” | 图 2.12 | 吞吐量: 1 等待时间: 2（注2） |
| 传送指令（存储） | • MOV “Rs,[Rd]” / “Rs,dsp[Rd]” / “Rs,[Rd+]” / “Rs,[−Rd]” / “Rs,[Ri,Rb]” • PUSH “Rs” • PUSHC “CR” | 图 2.13 | 1 |
| 位操作指令（寄存器） | • {BCLR,BNOT,BSET,BTST} “#IMM,Rd” / “Rs,Rd” • BMCnd “#IMM,Rd” | 图 2.11 | 1 |
| 转移指令 | • BCnd “pcdsp” • {BRA,BSR} “pcdsp” / “Rs” • {JMP,JSR} “Rs” | 图 2.22 | 转移成立: 3 转移不成立: 1 |
| 浮点运算指令 （寄存器之间、立即数 - 寄存器） | • FCMP “#IMM,Rd” / “Rs,Rd” | 图 2.11 | 1 |
| 系统控制指令 | • CLRPSW,SETPSW “#IMM” • MVTC “#IMM,CR” / “Rs,CR” • MVFC “CR,Rd” | — | 1 |

注 1. 除法指令的周期数因除数和被除数的值而不同。

注 2. 有关吞吐量以及表示等待时间的周期数，请参照“2.8.3 指令处理时间的计算方法”。

转换为单一微操作的指令操作如图 2.11 ~图 2.13 所示。

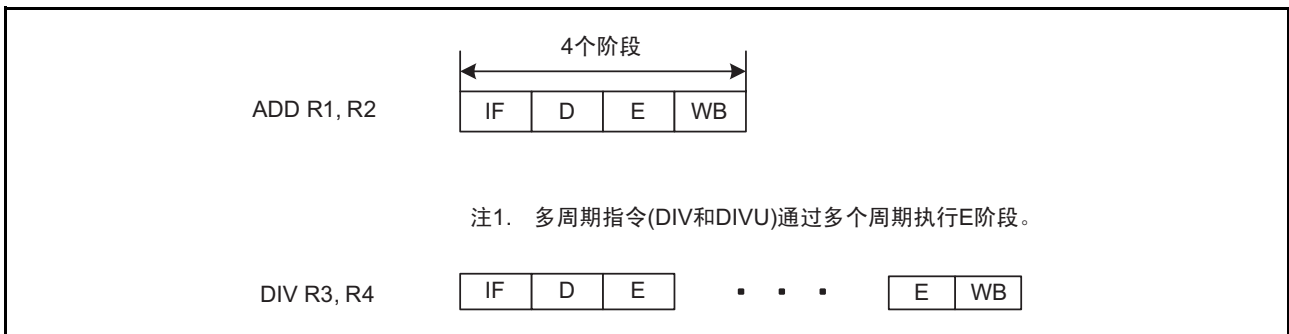


图 2.11 寄存器之间、立即数 - 寄存器的运算

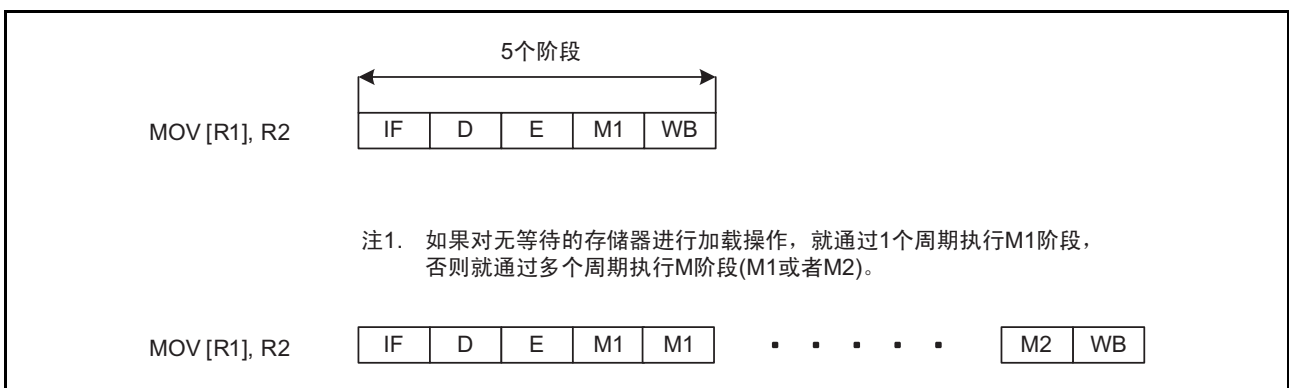


图 2.12 加载操作

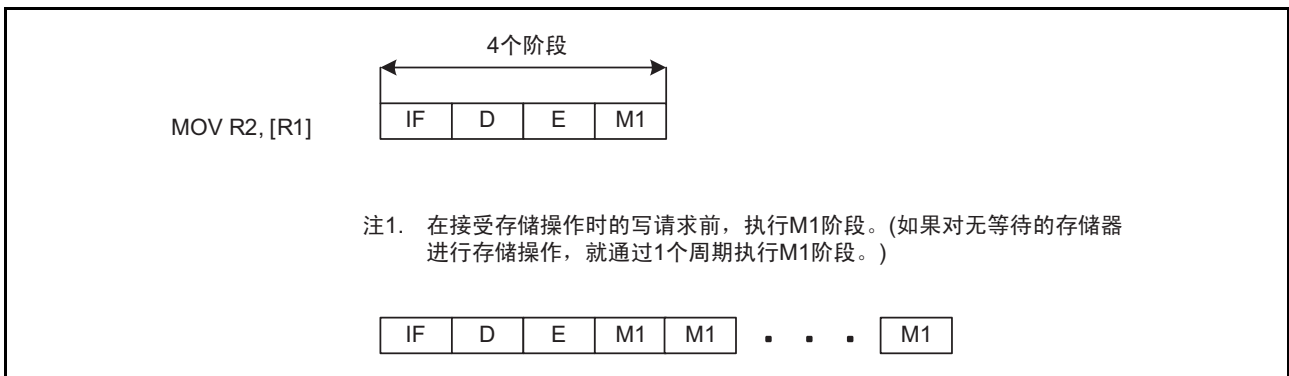


图 2.13 存储操作

2.8.2.2 转换为多个微操作的指令和流水线处理

转换为多个微操作的指令如下所示，周期数表示无等待存储器存取时的周期数。

表 2.14 转换为多个微操作的指令 (1/2)

| 指令 | 助记符 (当省略指令长度时, 为全长度共同的操作) | 参照图 | 周期数 |
|---|--|--------|--|
| 算术 / 逻辑运算指令 (寄存器之间、立即数 - 寄存器) | • ADC, ADD…… (省略), XOR “ [Rs], Rd” / “ dsp[Rs], Rd” | 图 2.14 | 3 |
| 算术 / 逻辑运算指令 (除法) | • DIV “ [Rs], Rd/dsp[Rs], Rd” | — | 5 ~ 22 |
| | • DIVU “ [Rs], Rd/dsp[Rs], Rd” | — | 4 ~ 20 |
| 算术 / 逻辑运算指令 (乘法 32 位 × 32 位 → 64 位) (寄存器之间、寄存器 - 立即数) | • {EMUL, EMULU} “ #IMM, Rd” / “ Rs, Rd” | 图 2.16 | 2 |
| 算术 / 逻辑运算指令 (乘加运算) | • RMPA.B | — | $6+7 \times \text{floor}(n/4)+4 \times (n\%4)$ n 为处理字节数 (注 1) |
| | • RMPA.W | — | $6+5 \times \text{floor}(n/2)+4 \times (n\%2)$ n 为处理字数 (注 1) |
| | • RMPA.L | — | $6+4n$ n 为处理长字数 (注 1) |
| 传送指令 (存储器之间的传送) | • MOV “ [Rs], [Rd]” / “ dsp[Rs], [Rd]” / “ [Rs], dsp[Rd]” / “ dsp[Rs], [Rd]” • PUSH “ [Rs]” / “ dsp[Rs]” | 图 2.15 | 3 |
| 位操作指令 (存储器源操作数) | • {BCLR, BNOT, BSET, BTST} “ #IMM, [Rd]” / “ #IMM, dsp[Rd]” • BMCnd “ #IMM, [Rd]” / “ #IMM, dsp[Rd]” | 图 2.15 | 3 |
| 传送指令 (加载指令) | • POPC “ CR” | — | 吞吐量: 3 等待时间: 4 (注 2) |
| 传送指令 (多个寄存器的保存) | • PUSHM “ Rs-Rs2” | — | n n 为寄存器的个数 (注 3) |
| 传送指令 (多个寄存器的恢复) | • POPM “ Rs-Rs2” | — | 吞吐量: n 等待时间: n+1 n 为寄存器的个数 (注 2、注 4) |
| 传送指令 (寄存器之间的交换) | • XCHG “ Rs, Rd” | 图 2.17 | 2 |
| 传送指令 (存储器 - 寄存器的交换) | • XCHG “ [Rs], Rd” / “ dsp[Rs], Rd” | 图 2.18 | 2 |
| 转移指令 | • RTS | — | 5 |
| | • RTSD “ #IMM” | — | 5 |
| | • RTSD “ #IMM, Rd-Rd2” | — | 吞吐量: $n < 5 ? 5 : 1+n$ 等待时间: $n < 4 ? 5 : 2+n$ n 为寄存器的个数 (注 2) |

表 2.14 转换为多个微操作的指令 (2/2)

| 指令 | 助记符 (当省略指令长度时, 为全长度共同的操作) | 参照图 | 周期数 |
|---------------------------|---|--------|--|
| 字符串操作指令 (注5) | • SCMPU | — | $2+4 \times \text{floor}(n/4)+4 \times (n\%4)$ n 为比较字节数 (注1) |
| | • SMOVB | — | $n > 3?$ $6+3 \times \text{floor}(n/4)+3 \times (n\%4):$ $2+3n$ n 为传送字节数 (注1) |
| | • SMOVF,SMOVU | — | $2+3 \times \text{floor}(n/4)+3 \times (n\%4)$ n 为传送字节数 (注1) |
| | • SSTR.B | — | $2+\text{floor}(n/4)+n\%4$ n 为传送字节数 (注1) |
| | • SSTR.W | — | $2+\text{floor}(n/2)+n\%2$ n 为传送字数 (注1) |
| | • SSTR.L | — | $2+n$ n 为传送长字数 |
| | • SUNTIL.B,SWHILE.B | — | $3+3 \times \text{floor}(n/4)+3 \times (n\%4)$ n 为比较字节数 (注1) |
| | • SUNTIL.W,SWHILE.W | — | $3+3 \times \text{floor}(n/2)+3 \times (n\%2)$ n 为比较字数 (注1) |
| | • SUNTIL.L,SWHILE.L | — | $3+3 \times n$ n 为比较长字数 |
| 浮点运算指令 (寄存器之间、立即数-寄存器) | • {FADD,FSUB} “#IMM,Rd” / “Rs,Rd” | 图 2.19 | 4 |
| | • FMUL “#IMM,Rd” / “Rs,Rd” | — | 3 |
| | • FDIV “#IMM,Rd” / “Rs,Rd” | — | 16 |
| | • {FTOI,ROUND,I TOF} “Rs,Rd” | — | 2 |
| 浮点运算指令 (存储器源操作数) | • {FADD,FSUB} “[Rs],Rd” / “dsp[Rs],Rd” | — | 6 |
| | • FMUL “[Rs],Rd” / “dsp[Rs],Rd” | — | 5 |
| | • FDIV “[Rs],Rd” / “dsp[Rs],Rd” | — | 18 |
| | • {FTOI,ROUND,I TOF} “[Rs],Rd” / “dsp[Rs],Rd” | — | 4 |
| 系统控制指令 | • RTE | — | 6 |
| | • RTFI | — | 3 |

注 1. floor(x): 小于等于 x 的最大整数

注 2. 有关吞吐量以及表示等待时间的周期数, 请参照“2.8.3 指令处理时间的计算方法”。

注 3. 将 PUSHM 指令转换为多个存储操作。MOV 指令的存储操作和指定寄存器次数的重复是相同的流水线处理。

注 4. 将 POPM 指令转换为多个加载操作。MOV 指令的加载操作和指定寄存器次数的重复是相同的流水线处理。

注 5. 如果在执行 SCMPU、SMOVU、SWHILE、SUNTIL 各指令时满足结束条件, 就结束指令的执行, 与记载的周期无关。

转换为多个微操作的指令操作如图 2.14 ~ 图 2.19 所示，图中的小写字母表示微操作。

注 1. mop: 微操作; stall: 流水线延迟

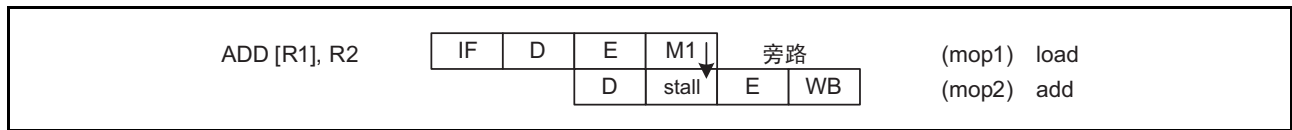


图 2.14 算术 / 逻辑运算指令 (存储器源操作数)

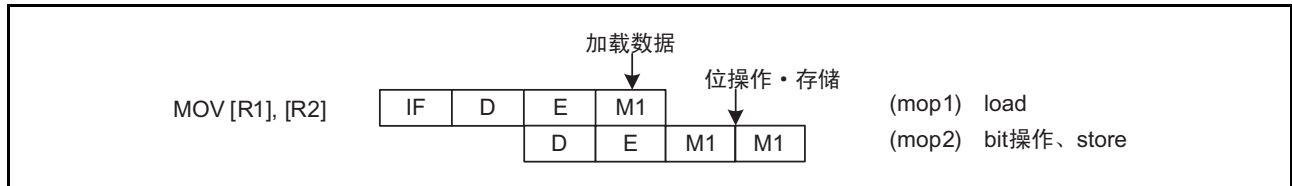


图 2.15 MOV 指令 (存储器之间的传送) 和位操作指令 (存储器源操作数)

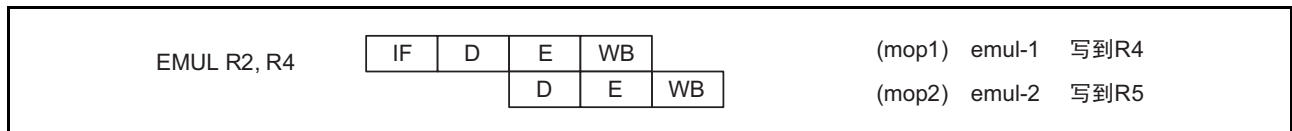


图 2.16 EMUL 指令和 EMULU 指令 (寄存器之间、寄存器 - 立即数)

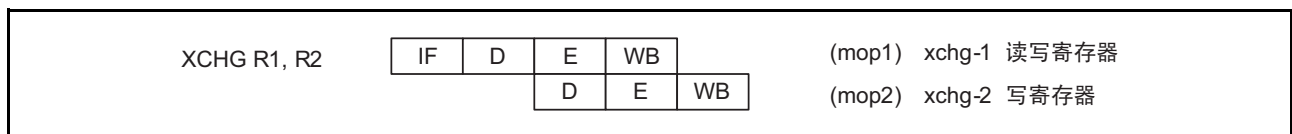


图 2.17 XCHG 指令 (寄存器)

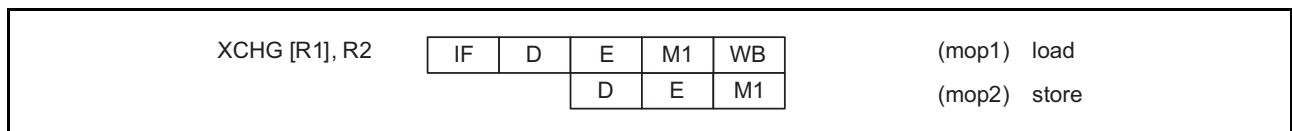


图 2.18 XCHG 指令 (存储器源操作数)

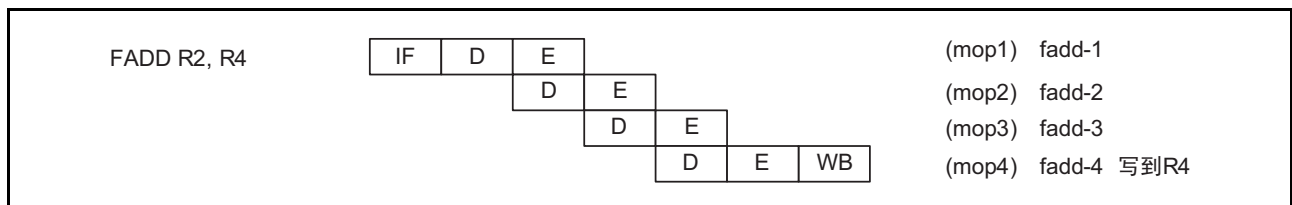


图 2.19 浮点运算指令 (寄存器之间、立即数 - 寄存器)

2.8.2.3 流水线的基本操作

在理想的流水线处理中，各阶段的执行周期数为“1”，但是流水线处理有可能因各阶段中的处理以及转移执行等而发生混乱。

CPU 在 IF 阶段以指令为单位进行流水线阶段控制，在 D 以后的阶段以微操作为单位进行流水线阶段控制。

典型的流水线处理状况如下所示，图中的小写字母表示微操作。

注 1. mop: 微操作; stall: 流水线延迟

(1) 流水线处理混乱的情况

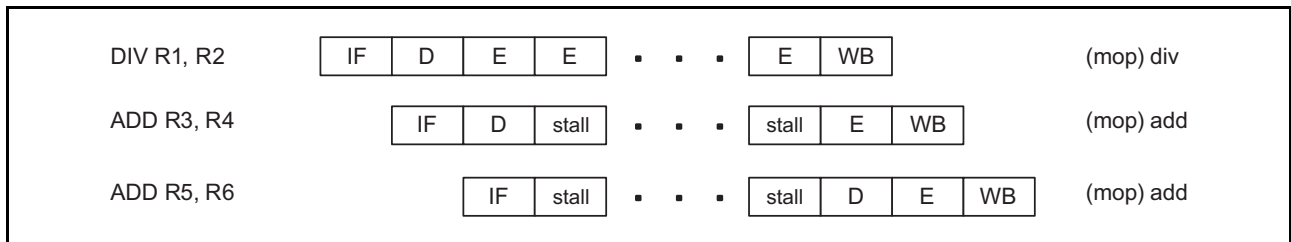


图 2.20 在执行 E 阶段过程中执行需要多个周期的指令的情况

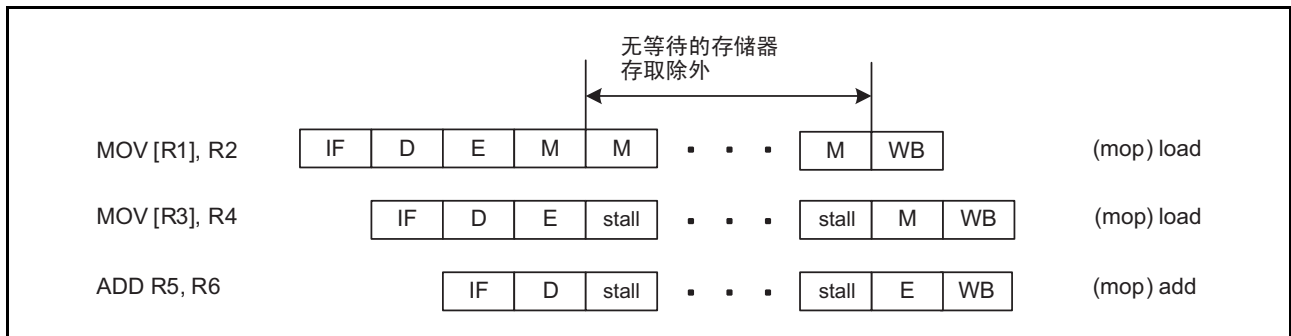


图 2.21 操作数存取没有在 1 个周期中结束的情况

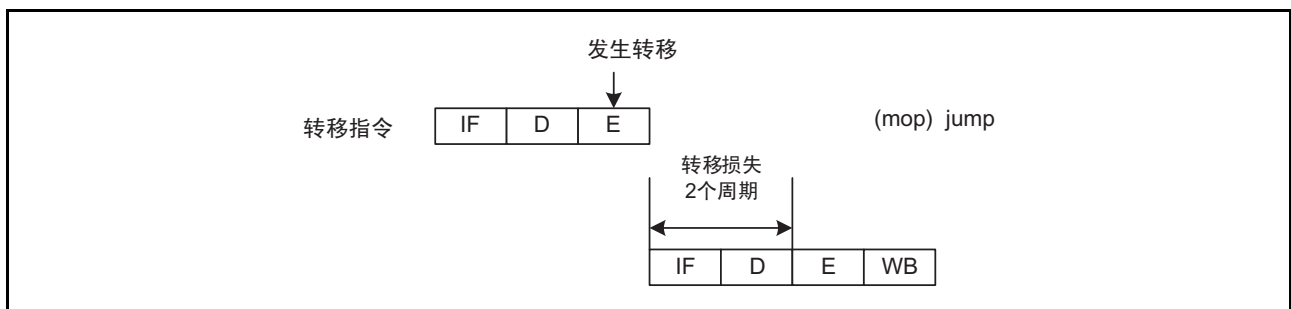


图 2.22 转移（在无条件的转移或者条件转移中条件成立的情况）

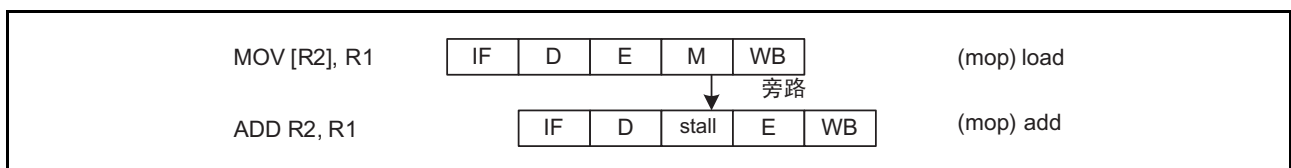


图 2.23 后续指令使用从存储器读取的操作数的情况

(2) 流水线处理不混乱的情况

(a) 旁路

即使后续指令使用前面的指令写的寄存器，在进行寄存器之间的运算时，流水线处理也不会因旁路而发生混乱。

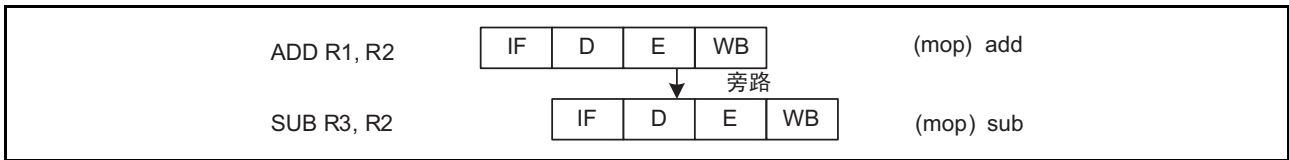


图 2.24 旁路

(b) 存储器加载和运算的 WB 阶段重叠的情况

因为加载数据和运算结果能同时写到寄存器，所以即使存储器加载和运算的 WB 阶段重叠，流水线处理也不会混乱。

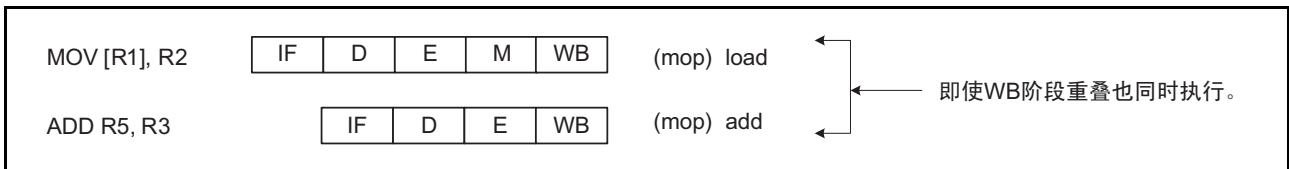


图 2.25 存储器加载和运算的 WB 阶段重叠的情况

(c) 在存储器加载结束前后续指令写相同寄存器的情况

在存储器加载结束前，即使后续指令写相同的寄存器，因为取消了存储器加载的 WB 阶段，所以流水线处理不会混乱。

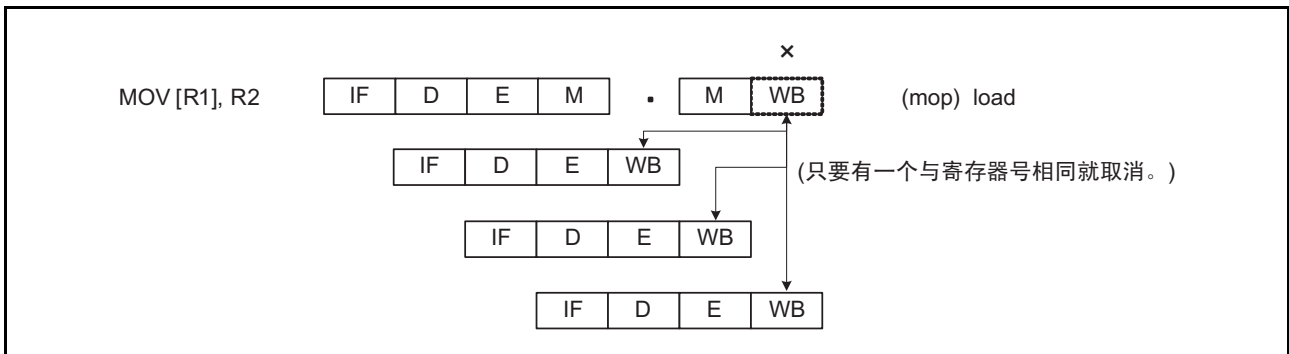


图 2.26 在结束存储器加载前，后续指令写相同寄存器的情况

(d) 后续指令不参照存储器加载的数据的情况

当后续指令不参照存储器加载的数据时，先执行后续指令，然后结束。
(Out-of-Order Completion)

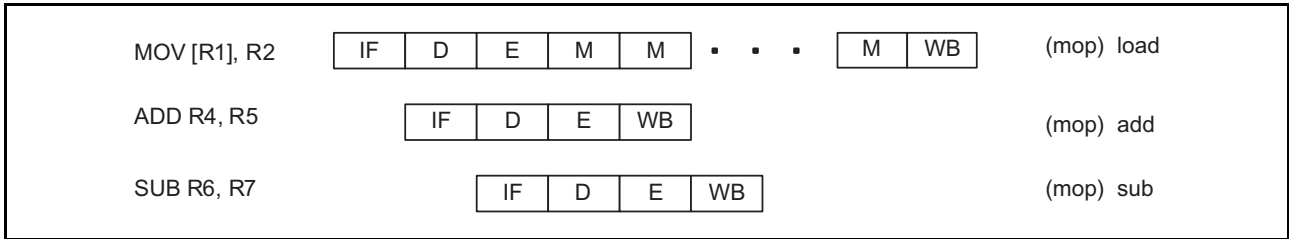


图 2.27 后续指令不参照存储器加载的数据的情况

2.8.3 指令处理时间的计算方法

CPU 的指令处理时间因流水线处理而不同，但是能使用以下的计算方法概算指令的处理时间。

- 计算周期数（参照表 2.13 和表 2.14）。
- 如果后续指令参照存储器加载的结果，存储器加载的指令周期数就为“等待时间”的周期数。否则，就为“吞吐量”的周期数。
- 如果引起取指令的延迟，就再追加周期数。
- 根据系统结构，存储器存取有时需要多个周期。RX610 群的存储器存取周期数取决于机种。

2.8.4 中断响应周期数

中断响应处理的周期数如表 2.15 所示。

表 2.15 中断响应周期数

| 中断请求的种类 / 处理内容 | 高速中断 | 高速中断以外的中断 |
|---|-------------------------|-----------|
| ICU 判断优先级。 | 2 个周期 | |
| CPU 从通知中断请求到接受中断的周期数。 | N 个周期 (因正在执行的指令而不同。) | |
| CPU 硬件预处理 保存到 PC 或者 PSW 的 RAM。 (将高速中断保存到控制寄存器。) 读向量。 转移到处理器的起始位置。 | 4 个周期 | 6 个周期 |

表 2.15 是 CPU 的存储器存取全部以无等待处理时的中断响应时间。RX610 群装载了能无等待存取的内部 ROM 和内部 RAM。通过将程序（包括向量）分配到内部 ROM，将堆栈区分配到内部 RAM，能将中断响应周期数缩到最短。而且，异常处理程序的起始地址必须指定 8 字节边界。

有关从通知中断请求到接受中断的周期数 N，请参照“表 2.13 转换为单一微操作的指令”和“表 2.14 转换为多个微操作的指令”。

中断接受时序取决于 CPU 的流水线状态。有关中断的接受时序，请参照“9.3.1 接受时序和被保存的 PC 值”。

3. 运行模式

3.1 运行模式的种类和选择

通过 MD1 引脚和 MD0 引脚以及系统控制寄存器 0 (SYSCR0) 的 ROME 位和 EXBE 位设定运行模式。在各运行模式中，能选择字节序。通过 MDE 引脚设定字节序，有关字节序请参照“11. 总线”。不能在 LSI 运行时更改 MDE、MD1、MD0 引脚，也不能设定表 3.1 中没有的组合。

表 3.1 通过模式引脚选择的运行模式

| 模式引脚 | | SYSCR0 寄存器的初始状态 | | 运行模式 | 内部 ROM (注 1) | 外部总线 |
|------|-----|-----------------|------|--------|-----------------|------|
| MD1 | MD0 | ROME | EXBE | | | |
| 0 | 1 | 1 | 0 | 引导模式 | 有效 | 无效 |
| 1 | 0 | 1 | 0 | 用户引导模式 | 有效 | 无效 |
| 1 | 1 | 1 | 0 | 单芯片模式 | 有效 | 无效 |

注 1. 内部 ROM 有 ROM 和数据闪存，详细内容请参照“26. ROM (保存代码的闪存)”和“27. 数据闪存 (保存数据的闪存)”。

表 3.2 通过寄存器选择的运行模式

| SYSCR0 寄存器 | | 运行模式 | 内部 ROM (注 1) | 外部总线 |
|------------|------|---------------|-----------------|------|
| ROME | EXBE | | | |
| 0 | 0 | 单芯片模式 | 无效 | 无效 |
| 1 | 0 | | 有效 | 无效 |
| 0 | 1 | 内部 ROM 无效扩展模式 | 无效 | 有效 |
| 1 | 1 | 内部 ROM 有效扩展模式 | 有效 | 有效 |

注 1. 内部 ROM 有 ROM 和数据闪存，详细内容请参照“26. ROM (保存代码的闪存)”和“27. 数据闪存 (保存数据的闪存)”。

表 3.3 字节序的选择

| 模式引脚 | 字节序 |
|------|-----|
| MDE | |
| 0 | 小端法 |
| 1 | 大端法 |

3.2 寄存器说明

运行模式的相关寄存器一览表如表 3.4 所示。

表 3.4 运行模式的相关寄存器一览表

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|-----------|--------|--------------------|------------|------|
| 模式监视寄存器 | MDMONR | 10000000 x00000xxb | 0008 0000h | 16 |
| 模式状态寄存器 | MDSR | 00000000 00001001b | 0008 0002h | 16 |
| 系统控制寄存器 0 | SYSCR0 | 00000000 00000001b | 0008 0006h | 16 |
| 系统控制寄存器 1 | SYSCR1 | 00000000 00000001b | 0008 0008h | 16 |

3.2.1 模式监视寄存器 (MDMONR)

地址 0008 0000h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|-------|-----|-----|-----|-----|-----|-----|----|----|-------|----|----|----|----|----|-------|-------|
| | — | — | — | — | — | — | — | — | MDE | — | — | — | — | — | MD1 | MD0 |
| 复位后的值 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x(注1) | 0 | 0 | 0 | 0 | 0 | x(注1) | x(注1) |

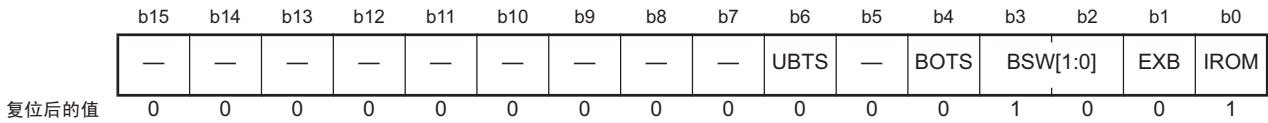
注 1. 因模式引脚(MDE、MD1和MD0)的设定而不同。

| 位 | 符号 | 位名 | 功能 | R/W |
|--------|-----|------------|--|-----|
| b0 | MD0 | MD0 引脚状态标志 | 0: MD0 引脚为 “0” 1: MD0 引脚为 “1” | R |
| b1 | MD1 | MD1 引脚状态标志 | 0: MD1 引脚为 “0” 1: MD1 引脚为 “1” | R |
| b6-b2 | — | 保留位 | 读取值为 “0”，写操作无效。 | R |
| b7 | MDE | MDE 引脚状态标志 | 0: MDE 引脚为 “0” (小端法) 1: MDE 引脚为 “1” (大端法) | R |
| b14-b8 | — | 保留位 | 读取值为 “0”，写操作无效。 | R |
| b15 | — | 保留位 | 读取值为 “1”，写操作无效。 | R |

MDMONR 寄存器是监视模式引脚的寄存器。

3.2.2 模式状态寄存器 (MDSR)

地址 0008 0002h

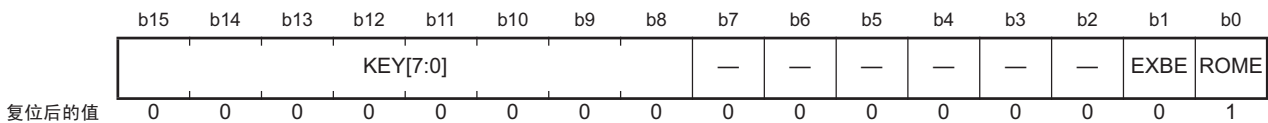


| 位 | 符号 | 位名 | 功能 | R/W |
|--------|----------|----------------|--|-----|
| b0 | IROM | 内部 ROM 的启动状态标志 | 0: 启动时, 内部 ROM 无效。 1: 启动时, 内部 ROM 有效。 | R |
| b1 | EXB | 外部总线的启动状态标志 | 0: 启动时, 外部总线无效。 1: 启动时, 外部总线有效。 | R |
| b3-b2 | BSW[1:0] | 外部总线宽度的启动标志 | b3 b2 0 0: 启动 16 位总线 0 1: 不能设定 1 0: 启动 8 位总线 1 1: 不能设定 | R |
| b4 | BOTS | 引导模式的启动标志 | 0: 非引导模式启动 1: 引导模式启动 | R |
| b5 | — | 保留位 | 读取值为“0”, 写操作无效。 | R |
| b6 | UBTS | 用户引导模式的启动标志 | 0: 非用户引导模式启动 1: 用户引导模式启动 | R |
| b15-b7 | — | 保留位 | 读取值为“0”, 写操作无效。 | R |

MDSR 寄存器是监视启动时的内部状态的寄存器。

3.2.3 系统控制寄存器 0 (SYSCR0)

地址 0008 0006h



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|----------|------------|---|-----|
| b0 | ROME | 内部 ROM 允许位 | 0: 内部 ROM 无效 1: 内部 ROM 有效 | R/W |
| b1 | EXBE | 外部总线允许位 | 0: 外部总线无效 1: 外部总线有效 | R/W |
| b7-b2 | — | 保留位 | 读写值都为“0”。 | R/W |
| b15-b8 | KEY[7:0] | SYSCR0 键码 | 5Ah: 允许写 SYSCR0 寄存器 上述以外: 禁止写 SYSCR0 寄存器 读取值为“00h”。 | R/W |

SYSCR0 寄存器是选择内部 ROM 和外部总线的有效或者无效的寄存器。

ROME 位（内部 ROM 允许位）

此位选择内部 ROM（ROM 和数据闪存）的有效或者无效。

当 ROME 位为“1”时，能写“0”；当 ROME 位为“0”时，不能写“1”。如果将内部 ROM 从有效改为无效，就不能使用 ROME 位将内部 ROM 置为有效。

不能在存取内部 ROM 时给 ROME 位写“0”。在给 ROME 位写“0”并且将内部 ROM 设定为无效后，必须先确认 ROME 位已被改写为“0”，然后再进行以后的处理。

EXBE 位（外部总线允许位）

此位选择外部总线的有效或者无效。

必须在未执行外部总线周期的状态下给 EXBE 位写“0”。

因为外部总线和内部总线有可能同时运行，所以必须注意将外部总线置为无效的情况。要改写 EXBE 位时，必须在写寄存器后确认 EXBE 位已被改写，然后再进行总线存取。

需要同时设定 EXBE 位和 I/O 端口，详细内容请参照“14. I/O 端口”。

KEY[7:0] 位（SYSCR0 键码）

这些位选择允许或者禁止写 SYSCR0 寄存器。

在给 ROME 位和 EXBE 位写值时，也必须给 KEY[7:0] 位写“5Ah”。如果给 KEY[7:0] 位写“5Ah”以外的值，即使写 SYSCR0 寄存器，ROME 位和 EXBE 位的值也不变。

3.2.4 系统控制寄存器 1（SYSCR1）

地址 0008 0008h

| | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|------|
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RAME |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| 位 | 符号 | 位名 | 功能 | R/W |
|--------|------|---------|------------------------------|-----|
| b0 | RAME | RAM 允许位 | 0: 内部 RAM 无效 1: 内部 RAM 有效 | R/W |
| b15-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

SYSCR1 寄存器是选择内部 RAM 的有效或者无效的寄存器。

RAME 位（RAM 允许位）

此位选择内部 RAM 的有效或者无效。

一旦解除复位，RAME 位就被初始化为“1”。

不能在存取内部 RAM 时写“0”。在将内部 RAM 从无效改为有效后紧接着存取内部 RAM 时，必须在确认 RAME 位已被改写为“1”后进行存取。

即使将 RAME 位置“0”，也保持内部 RAM 的值，而且 RAM 待机电压（VRAM）也必须保持规定值。详细内容请参照“29. 电特性”。

3.3 运行模式的说明

3.3.1 单芯片模式

在此模式中，内部 ROM 有效或者无效而外部总线无效（SYSCR0.EXBE 位 =0）。全部 I/O 端口都能用作输入 / 输出端口。

启动时的内部 ROM 有效。当内部 ROM 有效（SYSCR0.ROME 位 =1）时，能将内部 ROM 设定为无效（ROME 位 =0）；当内部 ROM 无效（ROME 位 =0）时，不能将内部 ROM 设定为有效（ROME 位 =1）。

能通过将 SYSCR0.EXBE 位置“1”并且转移到内部 ROM 有效扩展模式或者内部 ROM 无效扩展模式，使用外部总线。

3.3.2 内部 ROM 有效扩展模式

这是内部 ROM 有效（SYSCR0.ROME 位 =1）并且能使用外部总线的外部扩展模式（SYSCR0.EXBE 位 =1）。能将部分 I/O 端口用作数据总线的输入 / 输出、地址总线的输出和总线控制信号的输入 / 输出，详细内容请参照“14. I/O 端口”。

能通过设定总线的外部总线宽度选择位（CSiCNT.BSIZE[1:0] 位（i=0 ~ 7））更改外部总线宽度，详细内容请参照“11. 总线”。

能通过将 EXBE 位置“0”转移到单芯片模式（内部 ROM 有效）。

能通过将 ROME 位置“0”转移到内部 ROM 无效扩展模式。

3.3.3 内部 ROM 无效扩展模式

这是内部 ROM 无效（SYSCR0.ROME 位 =0）并且能使用外部总线的外部扩展模式（SYSCR0.EXBE 位 =1）。能将部分 I/O 端口用作数据总线的输入 / 输出、地址总线的输出和总线控制信号的输入 / 输出，详细内容请参照“14. I/O 端口”。

能通过设定总线的外部总线宽度选择位（CSiCNT.BSIZE[1:0] 位（i=0 ~ 7））更改外部总线宽度，详细内容请参照“11. 总线”。

不能将内部 ROM 设定为有效（ROME 位 =1）。

能通过将 EXBE 位置“0”转移到单芯片模式（内部 ROM 无效）。

3.3.4 引导模式

这是闪存的引导模式，除了闪存的编程 / 擦除以外，和单芯片模式中的运行相同。详细内容请参照“26. ROM（保存代码的闪存）”和“27. 数据闪存（保存数据的闪存）”。

3.3.5 用户引导模式

这是闪存的用户引导模式，除了闪存的编程 / 擦除以外，和单芯片模式中的运行相同。详细内容请参照“26. ROM（保存代码的闪存）”和“27. 数据闪存（保存数据的闪存）”。

3.4 运行模式的转移

3.4.1 通过模式引脚进行运行模式的转移

通过设定 MD1 引脚和 MD0 引脚进行运行模式的转移，如图 3.1 所示，能将运行模式转移到图中箭头所指的方向。

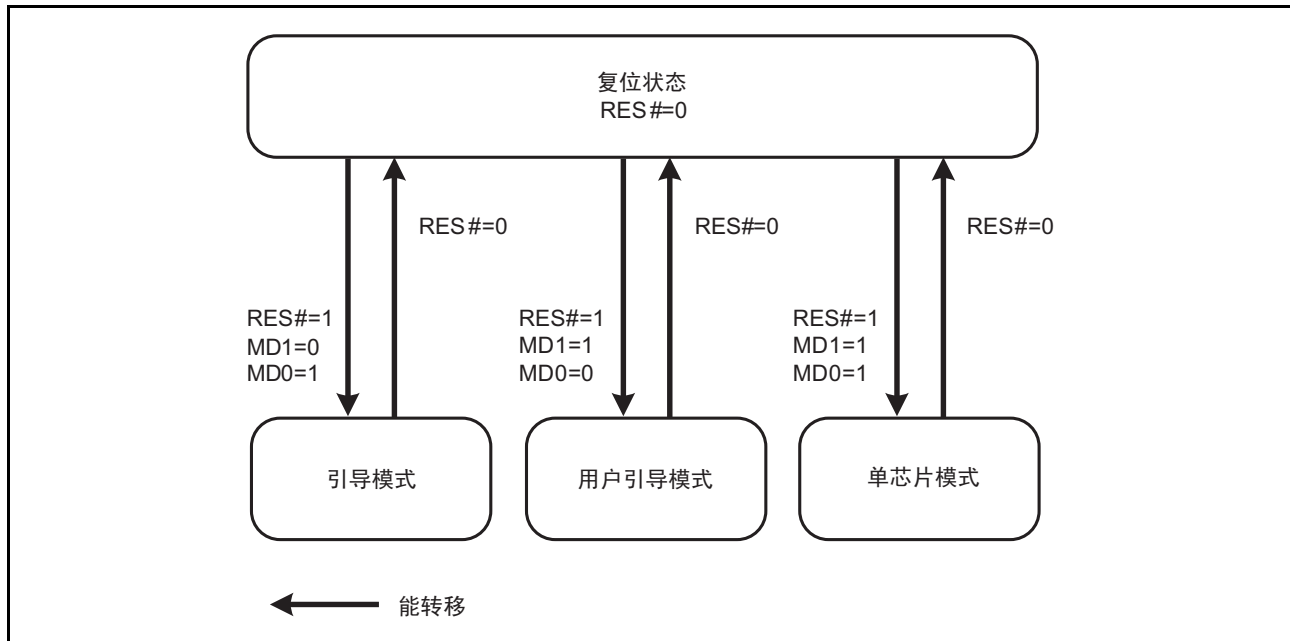


图 3.1 MD1 引脚、MD0 引脚的设定和运行模式

3.4.2 通过寄存器进行运行模式的转移

通过设定 SYSCR0.ROME 位和 SYSCR0.EXBE 位进行运行模式的转移，如图 3.2 所示，能将运行模式转移到图中箭头所指的方向。

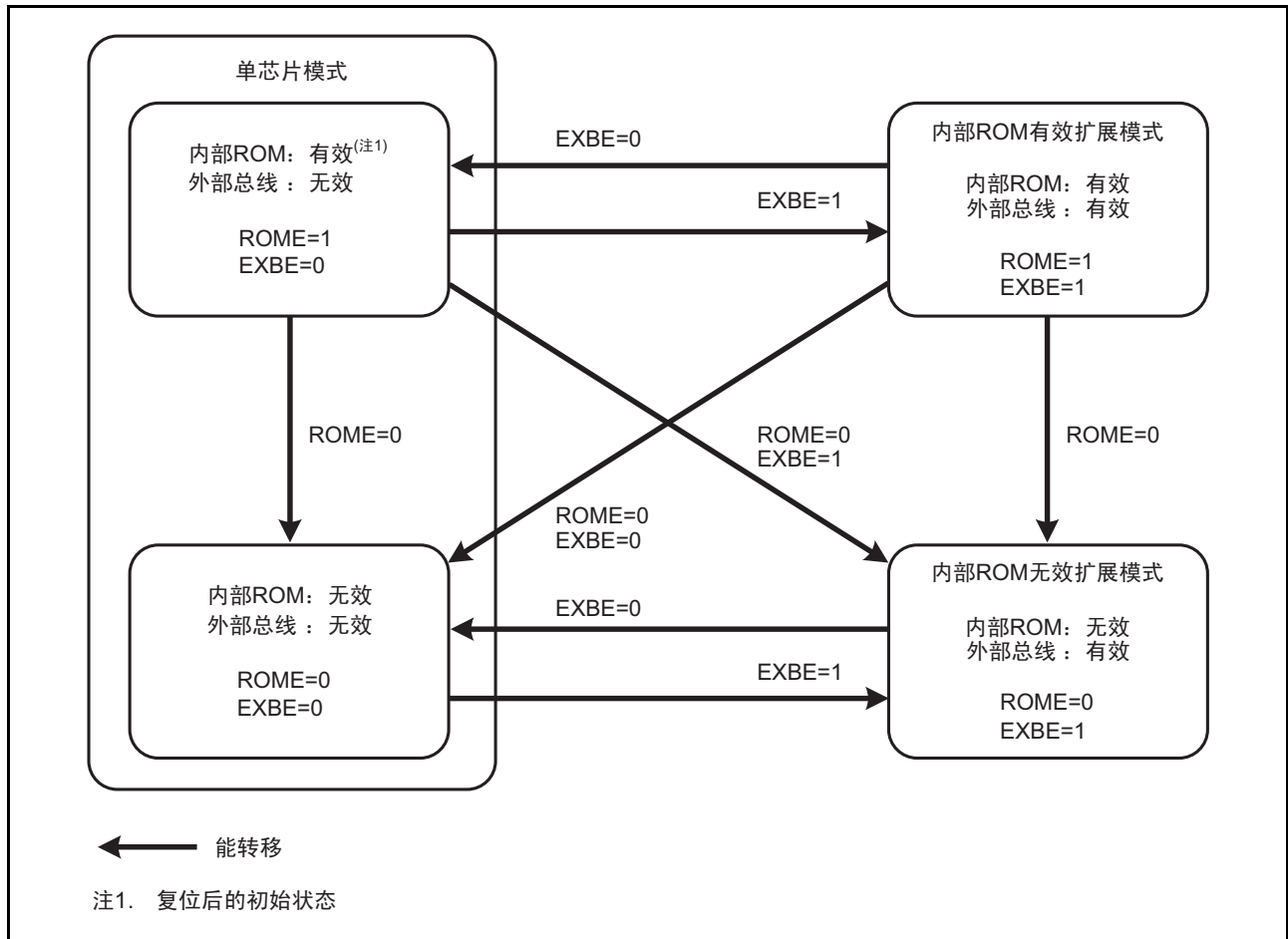


图 3.2 ROME 位、EXBE 位的设定和运行模式

4. 地址空间

4.1 地址空间

地址空间有地址 0000 0000h ~ 地址 FFFF FFFFh 的 4G 字节，能对程序区和数据区共计最多 4G 字节的空间进行线性存取。

能存取的区域因运行模式和各控制位的状态而不同，并且 ROM 容量因产品而不同。各产品和各运行模式的存储器映像如图 4.1 ~ 图 4.4 所示。

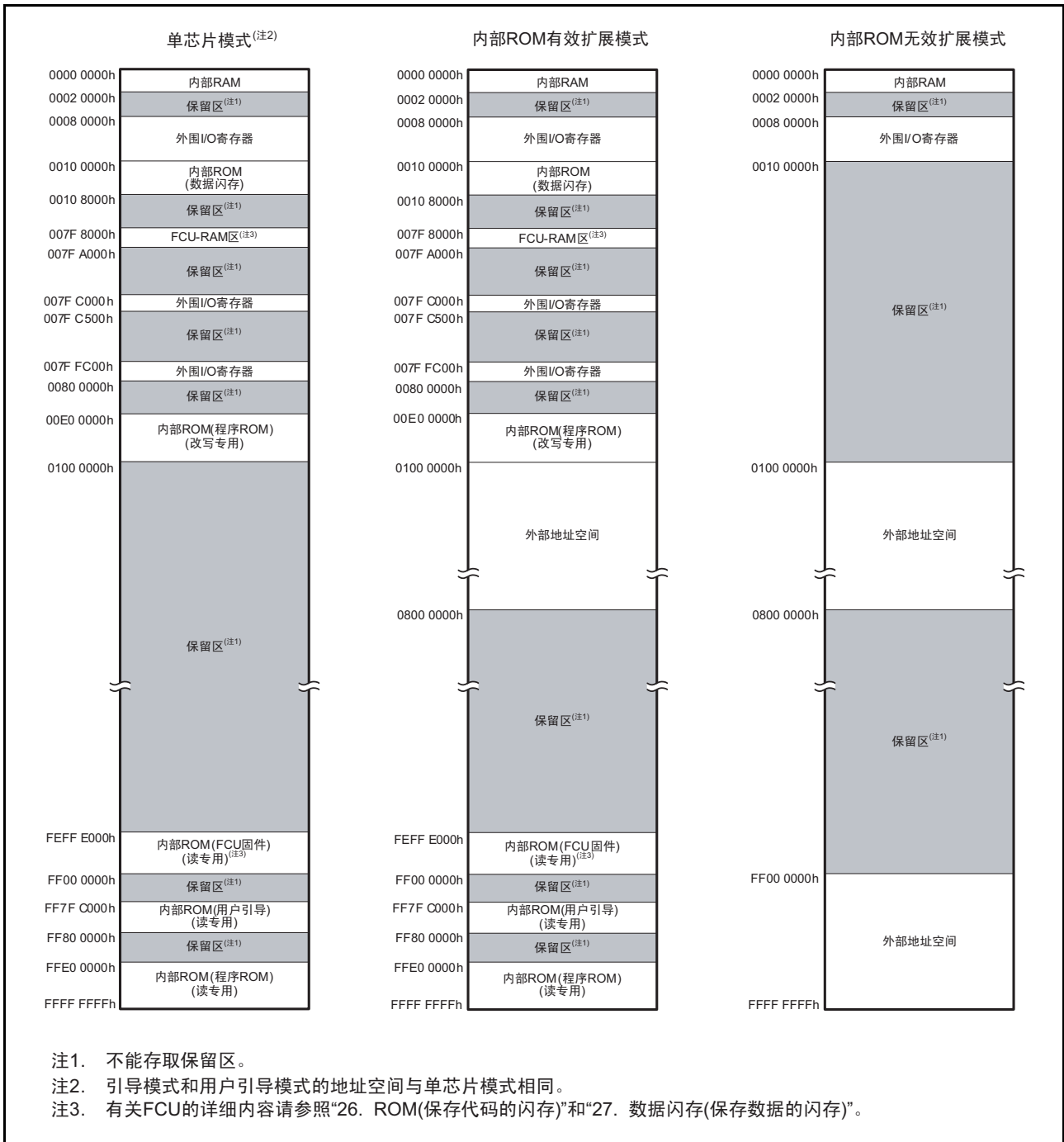


图 4.1 R5F56108 的存储器映像

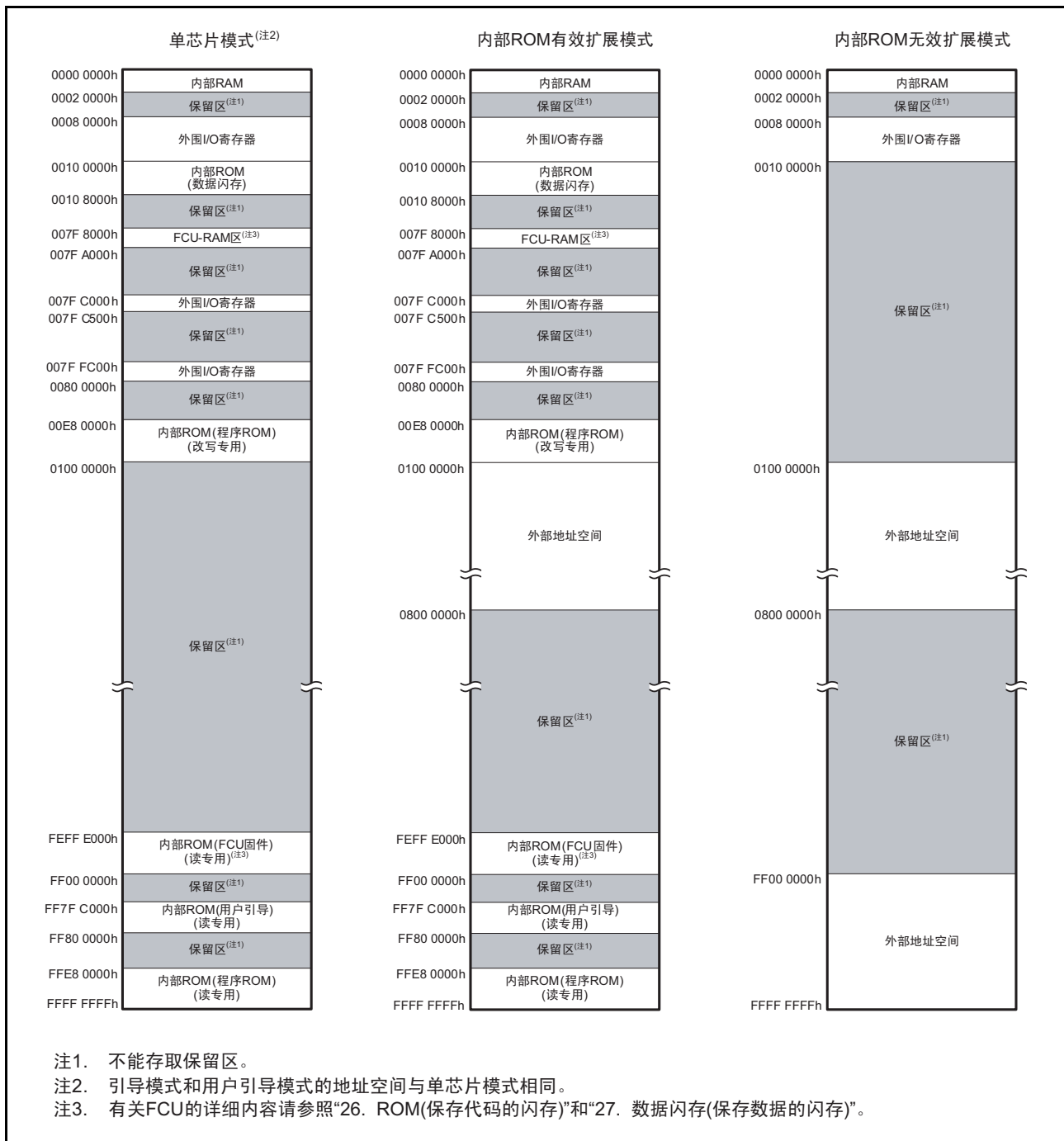


图 4.2 R5F56107 的存储器映像

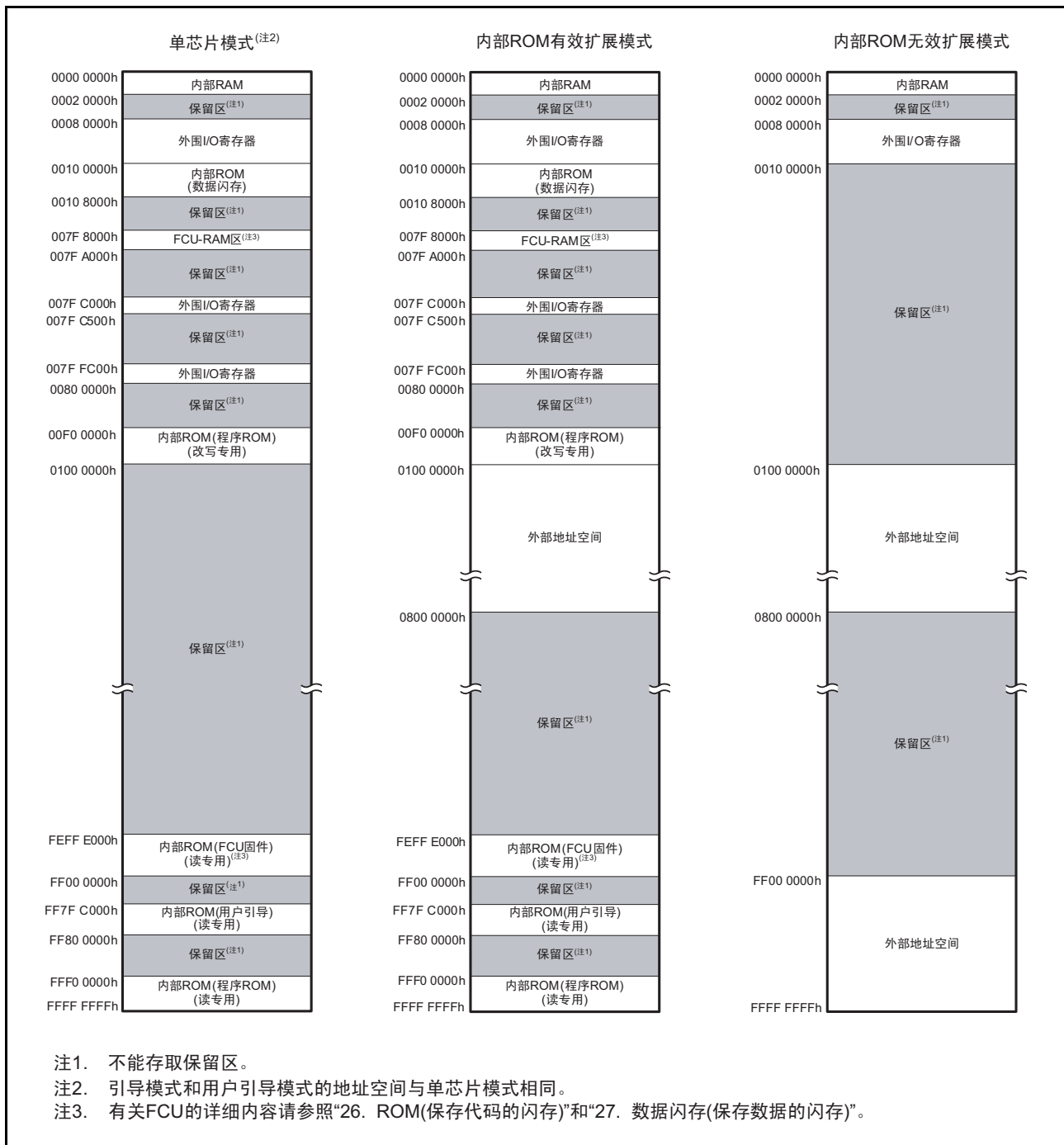


图 4.3 R5F56106 的存储器映像

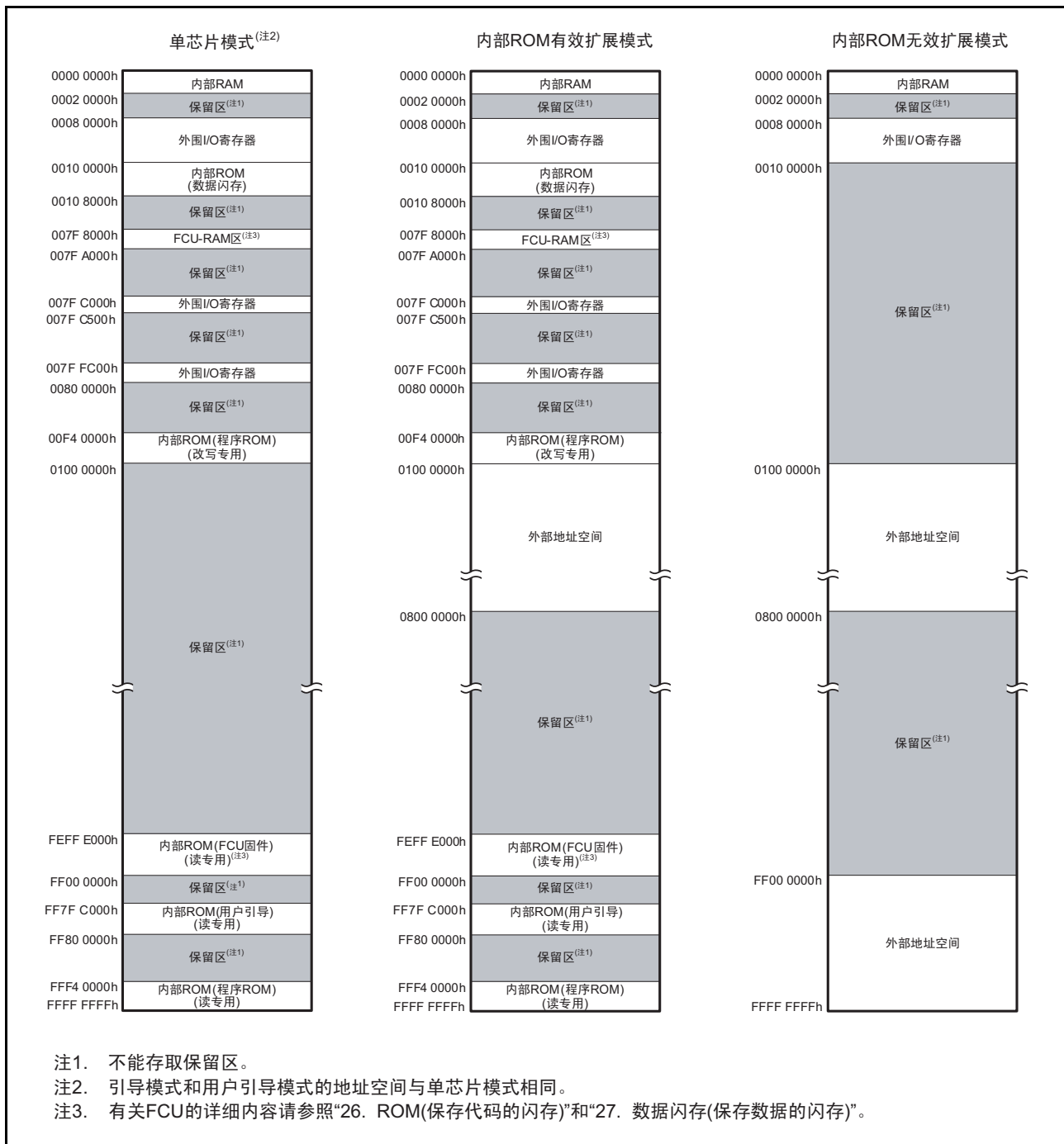


图 4.4 R5F56104 的存储器映像

4.2 外部地址空间

能通过 CSn# 引脚 (n=0 ~ 7) 输出的 CSn# 信号, 将外部地址空间最多分为 8 个区域。内部 ROM 无效扩展模式中的 CSn# 信号 (CSi 区域) (i=0 ~ 7) 和地址的对应如图 4.5 所示。

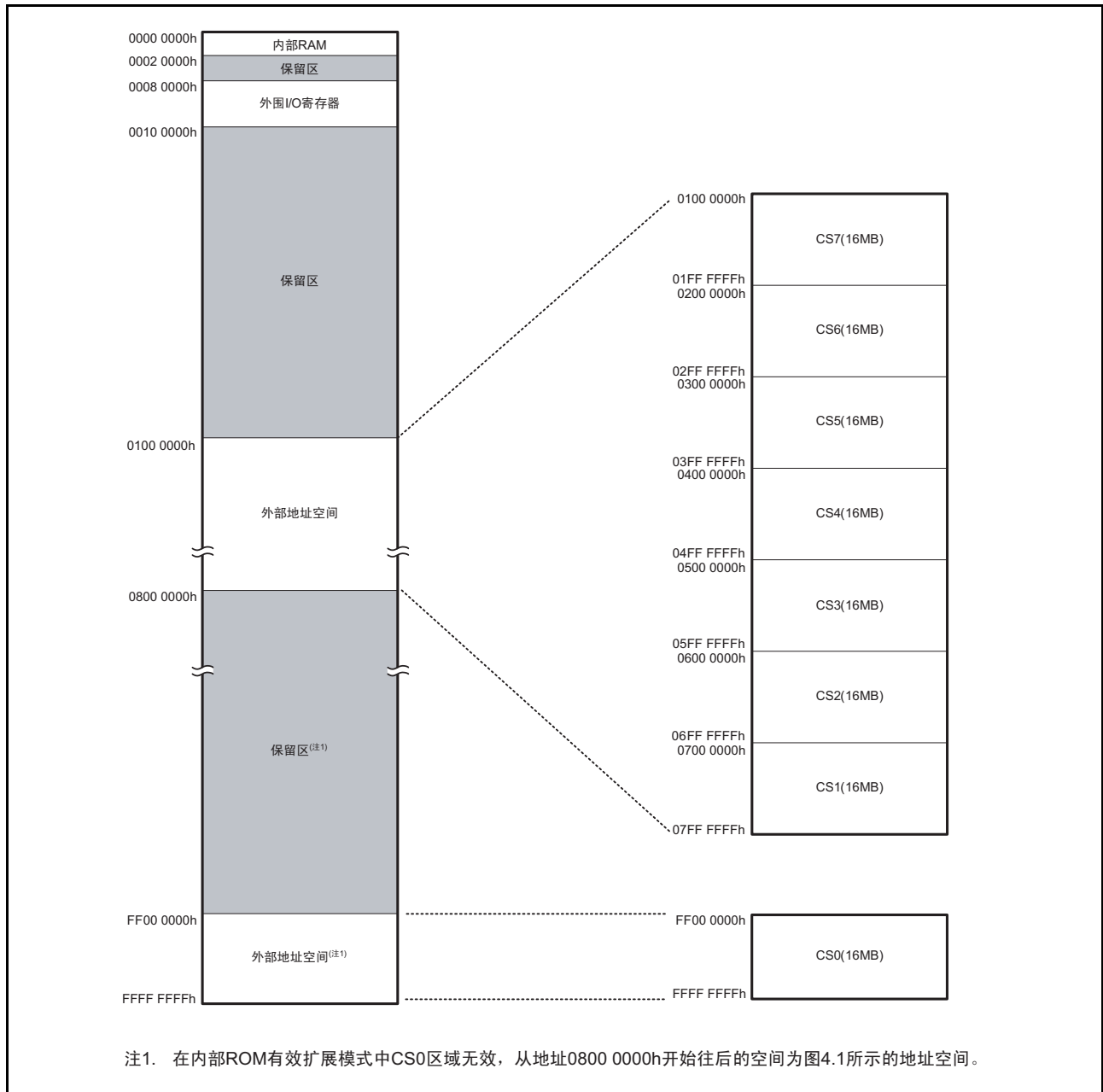


图 4.5 外部地址空间和 CSi 区域 (内部 ROM 无效扩展模式的情况)

5. I/O 寄存器

I/O 寄存器一览表汇总了有关内部寄存器的地址和位结构的信息。表中的记述方法以及写寄存器时的注意事项如下所示。

(1) I/O 寄存器地址一览表（地址顺序）

- 从分配地址小的寄存器开始依次记载。
- 根据模块的符号进行分类。
- 存取状态数表示指定的基准时钟的状态数。
- 在内部 I/O 寄存器的区域，寄存器一览表中没有记载的地址区为保留区。禁止存取保留区，因为无法保证存取这些寄存器时的运行及其后续运行，所以不能存取。

(2) I/O 寄存器位一览表

- 按照“寄存器地址一览表（地址顺序）”的顺序，记载位结构。
- 在位名部分用“—”表示保留位。
- 表中位名为空白的栏表示该寄存器全部被分配给计数器或者数据。
- 在 16 位或者 32 位寄存器的情况下，从 MSB 侧的位开始记载。

(3) 写 I/O 寄存器时的注意事项

在写 I/O 寄存器时，CPU 不等到写结束就执行后续指令。因此，在通过写 I/O 寄存器进行的设定变更被反映到运行前，有可能执行后续指令。

在以下例子中，必须在 I/O 寄存器的设定变更被反映后执行后续指令。

[需要注意的运行例子]

- 将中断请求的允许位（ICU.IERm.IENj 位）置“0”，然后要在禁止中断的状态下执行后续指令时
- 在转移到低功耗状态的预处理后继续执行 WAIT 指令时

在这些情况下，必须在写 I/O 寄存器后按照以下步骤等到写结束，然后执行后续指令。

1. 写 I/O 寄存器。
2. 将所写的 I/O 寄存器的值读到通用寄存器。
3. 用读取值进行运算。
4. 执行后续指令。

[指令例子]

- I/O 寄存器为字节的情况


```
MOV.L ADDR,R1
MOV.B DATA,[R1]
CMP [R1].UB,R1
;; 以后的处理
```
- I/O 寄存器为字的情况


```
MOV.L ADDR,R1
MOV.W DATA,[R1]
CMP [R1].W,R1
;; 以后的处理
```

- I/O 寄存器为长字的情况
MOV.L ADDR,R1
MOV.L DATA,[R1]
CMP [R1].L,R1
;; 以后的处理

另外，在写多个寄存器并且等到写结束后执行后续指令时，必须将最后写的 I/O 寄存器作为对象进行读操作和运算，而不需要将所写的全部寄存器作为对象进行读操作和运算。

(4) I/O 寄存器的存取周期数

用以下表达式表示存取 I/O 寄存器时的存取周期数（注 1）。

$$\begin{aligned} \text{I/O 寄存器的存取周期数} = & \text{内部主总线 1 的总线周期数} + \\ & \text{分频时钟的同步周期数} + \\ & \text{内部外围总线 1（或者 2）的总线周期数} \end{aligned}$$

内部外围总线 1（或者 2）的总线周期数因存取的目标寄存器而不同。各寄存器的 I/O 寄存器的存取周期数请参照“表 5.1 I/O 寄存器地址一览表”。

在存取内部外围总线 2 连接的外围功能和外部总线控制部的寄存器（总线错误的相关寄存器除外）时，增加分频时钟的同步周期。

分频时钟的同步周期数因 ICLK 和 PCLK（或者 BCLK）的频率比和总线存取的时序而不同，但是因为内部主总线 1 的总线周期数加上分频时钟的同步周期数最多为 1 个 PCLK（或者 BCLK）周期，所以在表 5.1 的记载中增加了 1 个 PCLK（或者 1 个 BCLK）。

注 1. 这是为了和外部存储器的取指令或者不同总线主控器（DMAC、DTC）的总线存取不发生竞争，CPU 存取寄存器时的周期数。

5.1 I/O 寄存器地址一览表（地址顺序）

表 5.1 I/O 寄存器地址一览表（1 / 20）

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|--------|------------------|---------|----|------|-------------------|
| 0008 0000h | SYSTEM | 模式监视寄存器 | MDMONR | 16 | 16 | 3 个 ICLK |
| 0008 0002h | SYSTEM | 模式状态寄存器 | MDSR | 16 | 16 | 3 个 ICLK |
| 0008 0006h | SYSTEM | 系统控制寄存器 0 | SYSCR0 | 16 | 16 | 3 个 ICLK |
| 0008 0008h | SYSTEM | 系统控制寄存器 1 | SYSCR1 | 16 | 16 | 3 个 ICLK |
| 0008 000Ch | SYSTEM | 待机控制寄存器 | SBYCR | 16 | 16 | 3 个 ICLK |
| 0008 0010h | SYSTEM | 模块停止控制寄存器 A | MSTPCRA | 32 | 32 | 3 个 ICLK |
| 0008 0014h | SYSTEM | 模块停止控制寄存器 B | MSTPCRB | 32 | 32 | 3 个 ICLK |
| 0008 0018h | SYSTEM | 模块停止控制寄存器 C | MSTPCRC | 32 | 32 | 3 个 ICLK |
| 0008 0020h | SYSTEM | 系统时钟控制寄存器 | SCKCR | 32 | 32 | 3 个 ICLK |
| 0008 1300h | BSC | 总线错误源清除寄存器 | BERCLR | 8 | 8 | 2 个 ICLK |
| 0008 1304h | BSC | 总线错误监视允许寄存器 | BEREN | 8 | 8 | 2 个 ICLK |
| 0008 1306h | BSC | 总线错误中断允许寄存器 | BERIE | 8 | 8 | 2 个 ICLK |
| 0008 2000h | DMAC0 | DMA 当前传送源地址寄存器 | DMCSA | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2004h | DMAC0 | DMA 当前传送目标地址寄存器 | DMCDA | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2008h | DMAC0 | DMA 当前传送字节计数寄存器 | DMCBC | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 200Ch | DMAC0 | DMA 模式寄存器 | DMMOD | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2010h | DMAC1 | DMA 当前传送源地址寄存器 | DMCSA | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2014h | DMAC1 | DMA 当前传送目标地址寄存器 | DMCDA | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2018h | DMAC1 | DMA 当前传送字节计数寄存器 | DMCBC | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 201Ch | DMAC1 | DMA 模式寄存器 | DMMOD | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2020h | DMAC2 | DMA 当前传送源地址寄存器 | DMCSA | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2024h | DMAC2 | DMA 当前传送目标地址寄存器 | DMCDA | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2028h | DMAC2 | DMA 当前传送字节计数寄存器 | DMCBC | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 202Ch | DMAC2 | DMA 模式寄存器 | DMMOD | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2030h | DMAC3 | DMA 当前传送源地址寄存器 | DMCSA | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2034h | DMAC3 | DMA 当前传送目标地址寄存器 | DMCDA | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2038h | DMAC3 | DMA 当前传送字节计数寄存器 | DMCBC | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 203Ch | DMAC3 | DMA 模式寄存器 | DMMOD | 32 | 32 | 4 ~ 5 个 ICLK |
| 0008 2200h | DMAC0 | DMA 重加载传送源地址寄存器 | DMRSA | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2204h | DMAC0 | DMA 重加载传送目标地址寄存器 | DMRDA | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2208h | DMAC0 | DMA 重加载传送字节计数寄存器 | DMRBC | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2210h | DMAC1 | DMA 重加载传送源地址寄存器 | DMRSA | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2214h | DMAC1 | DMA 重加载传送目标地址寄存器 | DMRDA | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2218h | DMAC1 | DMA 重加载传送字节计数寄存器 | DMRBC | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2220h | DMAC2 | DMA 重加载传送源地址寄存器 | DMRSA | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2224h | DMAC2 | DMA 重加载传送目标地址寄存器 | DMRDA | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2228h | DMAC2 | DMA 重加载传送字节计数寄存器 | DMRBC | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2230h | DMAC3 | DMA 重加载传送源地址寄存器 | DMRSA | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2234h | DMAC3 | DMA 重加载传送目标地址寄存器 | DMRDA | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2238h | DMAC3 | DMA 重加载传送字节计数寄存器 | DMRBC | 32 | 32 | 4 ~ 5 个 ICLK (注8) |
| 0008 2400h | DMAC0 | DMA 控制寄存器 A | DMCRA | 32 | 32 | 3 个 ICLK |
| 0008 2404h | DMAC0 | DMA 控制寄存器 B | DMCRB | 8 | 8 | 3 个 ICLK |
| 0008 2405h | DMAC0 | DMA 控制寄存器 C | DMCRC | 8 | 8 | 3 个 ICLK |
| 0008 2406h | DMAC0 | DMA 控制寄存器 D | DMCRD | 8 | 8 | 3 个 ICLK |
| 0008 2407h | DMAC0 | DMA 控制寄存器 E | DMCRE | 8 | 8 | 3 个 ICLK |

表 5.1 I/O 寄存器地址一览表 (2 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|---------|---------------|----------|----|------|-----------------|
| 0008 2408h | DMAC1 | DMA 控制寄存器 A | DMCRA | 32 | 32 | 3 个 ICLK |
| 0008 240Ch | DMAC1 | DMA 控制寄存器 B | DMCRB | 8 | 8 | 3 个 ICLK |
| 0008 240Dh | DMAC1 | DMA 控制寄存器 C | DMCRC | 8 | 8 | 3 个 ICLK |
| 0008 240Eh | DMAC1 | DMA 控制寄存器 D | DMCRD | 8 | 8 | 3 个 ICLK |
| 0008 240Fh | DMAC1 | DMA 控制寄存器 E | DMCRE | 8 | 8 | 3 个 ICLK |
| 0008 2410h | DMAC2 | DMA 控制寄存器 A | DMCRA | 32 | 32 | 3 个 ICLK |
| 0008 2414h | DMAC2 | DMA 控制寄存器 B | DMCRB | 8 | 8 | 3 个 ICLK |
| 0008 2415h | DMAC2 | DMA 控制寄存器 C | DMCRC | 8 | 8 | 3 个 ICLK |
| 0008 2416h | DMAC2 | DMA 控制寄存器 D | DMCRD | 8 | 8 | 3 个 ICLK |
| 0008 2417h | DMAC2 | DMA 控制寄存器 E | DMCRE | 8 | 8 | 3 个 ICLK |
| 0008 2418h | DMAC3 | DMA 控制寄存器 A | DMCRA | 32 | 32 | 3 个 ICLK |
| 0008 241Ch | DMAC3 | DMA 控制寄存器 B | DMCRB | 8 | 8 | 3 个 ICLK |
| 0008 241Dh | DMAC3 | DMA 控制寄存器 C | DMCRC | 8 | 8 | 3 个 ICLK |
| 0008 241Eh | DMAC3 | DMA 控制寄存器 D | DMCRD | 8 | 8 | 3 个 ICLK |
| 0008 241Fh | DMAC3 | DMA 控制寄存器 E | DMCRE | 8 | 8 | 3 个 ICLK |
| 0008 2502h | DMAC 通用 | DMA 启动控制寄存器 | DMSCNT | 8 | 8 | 3 个 ICLK |
| 0008 250Bh | DMAC 通用 | DMA 中断控制寄存器 | DMICNT | 8 | 8 | 3 个 ICLK |
| 0008 2517h | DMAC 通用 | DMA 传送结束检测寄存器 | DMEDET | 8 | 8 | 3 个 ICLK |
| 0008 251Bh | DMAC 通用 | DMA 仲裁状态寄存器 | DMASTS | 8 | 8 | 3 个 ICLK |
| 0008 3002h | BSC | CS0 模式寄存器 | CS0MOD | 16 | 16 | 1~2 个 BCLK (注7) |
| 0008 3004h | BSC | CS0 等待控制寄存器 1 | CS0WCNT1 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3008h | BSC | CS0 等待控制寄存器 2 | CS0WCNT2 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3012h | BSC | CS1 模式寄存器 | CS1MOD | 16 | 16 | 1~2 个 BCLK (注7) |
| 0008 3014h | BSC | CS1 等待控制寄存器 1 | CS1WCNT1 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3018h | BSC | CS1 等待控制寄存器 2 | CS1WCNT2 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3022h | BSC | CS2 模式寄存器 | CS2MOD | 16 | 16 | 1~2 个 BCLK (注7) |
| 0008 3024h | BSC | CS2 等待控制寄存器 1 | CS2WCNT1 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3028h | BSC | CS2 等待控制寄存器 2 | CS2WCNT2 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3032h | BSC | CS3 模式寄存器 | CS3MOD | 16 | 16 | 1~2 个 BCLK (注7) |
| 0008 3034h | BSC | CS3 等待控制寄存器 1 | CS3WCNT1 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3038h | BSC | CS3 等待控制寄存器 2 | CS3WCNT2 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3042h | BSC | CS4 模式寄存器 | CS4MOD | 16 | 16 | 1~2 个 BCLK (注7) |
| 0008 3044h | BSC | CS4 等待控制寄存器 1 | CS4WCNT1 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3048h | BSC | CS4 等待控制寄存器 2 | CS4WCNT2 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3052h | BSC | CS5 模式寄存器 | CS5MOD | 16 | 16 | 1~2 个 BCLK (注7) |
| 0008 3054h | BSC | CS5 等待控制寄存器 1 | CS5WCNT1 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3058h | BSC | CS5 等待控制寄存器 2 | CS5WCNT2 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3062h | BSC | CS6 模式寄存器 | CS6MOD | 16 | 16 | 1~2 个 BCLK (注7) |
| 0008 3064h | BSC | CS6 等待控制寄存器 1 | CS6WCNT1 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3068h | BSC | CS6 等待控制寄存器 2 | CS6WCNT2 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3072h | BSC | CS7 模式寄存器 | CS7MOD | 16 | 16 | 1~2 个 BCLK (注7) |
| 0008 3074h | BSC | CS7 等待控制寄存器 1 | CS7WCNT1 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3078h | BSC | CS7 等待控制寄存器 2 | CS7WCNT2 | 32 | 32 | 1~2 个 BCLK (注7) |
| 0008 3802h | BSC | CS0 控制寄存器 | CS0CNT | 16 | 16 | 1~2 个 BCLK (注7) |
| 0008 380Ah | BSC | CS0 恢复周期设定寄存器 | CS0REC | 16 | 16 | 1~2 个 BCLK (注7) |
| 0008 3812h | BSC | CS1 控制寄存器 | CS1CNT | 16 | 16 | 1~2 个 BCLK (注7) |
| 0008 381Ah | BSC | CS1 恢复周期设定寄存器 | CS1REC | 16 | 16 | 1~2 个 BCLK (注7) |

表 5.1 I/O 寄存器地址一览表 (3 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|------|---------------|--------|----|------|----------------|
| 0008 3822h | BSC | CS2 控制寄存器 | CS2CNT | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 382Ah | BSC | CS2 恢复周期设定寄存器 | CS2REC | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 3832h | BSC | CS3 控制寄存器 | CS3CNT | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 383Ah | BSC | CS3 恢复周期设定寄存器 | CS3REC | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 3842h | BSC | CS4 控制寄存器 | CS4CNT | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 384Ah | BSC | CS4 恢复周期设定寄存器 | CS4REC | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 3852h | BSC | CS5 控制寄存器 | CS5CNT | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 385Ah | BSC | CS5 恢复周期设定寄存器 | CS5REC | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 3862h | BSC | CS6 控制寄存器 | CS6CNT | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 386Ah | BSC | CS6 恢复周期设定寄存器 | CS6REC | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 3872h | BSC | CS7 控制寄存器 | CS7CNT | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 387Ah | BSC | CS7 恢复周期设定寄存器 | CS7REC | 16 | 16 | 1~2个 BCLK (注7) |
| 0008 7010h | ICU | 中断请求寄存器 016 | IR016 | 8 | 8 | 2个 ICLK |
| 0008 7015h | ICU | 中断请求寄存器 021 | IR021 | 8 | 8 | 2个 ICLK |
| 0008 7017h | ICU | 中断请求寄存器 023 | IR023 | 8 | 8 | 2个 ICLK |
| 0008 701Ch | ICU | 中断请求寄存器 028 | IR028 | 8 | 8 | 2个 ICLK |
| 0008 701Dh | ICU | 中断请求寄存器 029 | IR029 | 8 | 8 | 2个 ICLK |
| 0008 701Eh | ICU | 中断请求寄存器 030 | IR030 | 8 | 8 | 2个 ICLK |
| 0008 701Fh | ICU | 中断请求寄存器 031 | IR031 | 8 | 8 | 2个 ICLK |
| 0008 7040h | ICU | 中断请求寄存器 064 | IR064 | 8 | 8 | 2个 ICLK |
| 0008 7041h | ICU | 中断请求寄存器 065 | IR065 | 8 | 8 | 2个 ICLK |
| 0008 7042h | ICU | 中断请求寄存器 066 | IR066 | 8 | 8 | 2个 ICLK |
| 0008 7043h | ICU | 中断请求寄存器 067 | IR067 | 8 | 8 | 2个 ICLK |
| 0008 7044h | ICU | 中断请求寄存器 068 | IR068 | 8 | 8 | 2个 ICLK |
| 0008 7045h | ICU | 中断请求寄存器 069 | IR069 | 8 | 8 | 2个 ICLK |
| 0008 7046h | ICU | 中断请求寄存器 070 | IR070 | 8 | 8 | 2个 ICLK |
| 0008 7047h | ICU | 中断请求寄存器 071 | IR071 | 8 | 8 | 2个 ICLK |
| 0008 7048h | ICU | 中断请求寄存器 072 | IR072 | 8 | 8 | 2个 ICLK |
| 0008 7049h | ICU | 中断请求寄存器 073 | IR073 | 8 | 8 | 2个 ICLK |
| 0008 704Ah | ICU | 中断请求寄存器 074 | IR074 | 8 | 8 | 2个 ICLK |
| 0008 704Bh | ICU | 中断请求寄存器 075 | IR075 | 8 | 8 | 2个 ICLK |
| 0008 704Ch | ICU | 中断请求寄存器 076 | IR076 | 8 | 8 | 2个 ICLK |
| 0008 704Dh | ICU | 中断请求寄存器 077 | IR077 | 8 | 8 | 2个 ICLK |
| 0008 704Eh | ICU | 中断请求寄存器 078 | IR078 | 8 | 8 | 2个 ICLK |
| 0008 704Fh | ICU | 中断请求寄存器 079 | IR079 | 8 | 8 | 2个 ICLK |
| 0008 7060h | ICU | 中断请求寄存器 096 | IR096 | 8 | 8 | 2个 ICLK |
| 0008 7062h | ICU | 中断请求寄存器 098 | IR098 | 8 | 8 | 2个 ICLK |
| 0008 7063h | ICU | 中断请求寄存器 099 | IR099 | 8 | 8 | 2个 ICLK |
| 0008 7064h | ICU | 中断请求寄存器 100 | IR100 | 8 | 8 | 2个 ICLK |
| 0008 7065h | ICU | 中断请求寄存器 101 | IR101 | 8 | 8 | 2个 ICLK |
| 0008 7068h | ICU | 中断请求寄存器 104 | IR104 | 8 | 8 | 2个 ICLK |
| 0008 7069h | ICU | 中断请求寄存器 105 | IR105 | 8 | 8 | 2个 ICLK |
| 0008 706Ah | ICU | 中断请求寄存器 106 | IR106 | 8 | 8 | 2个 ICLK |
| 0008 706Bh | ICU | 中断请求寄存器 107 | IR107 | 8 | 8 | 2个 ICLK |
| 0008 706Ch | ICU | 中断请求寄存器 108 | IR108 | 8 | 8 | 2个 ICLK |
| 0008 706Fh | ICU | 中断请求寄存器 111 | IR111 | 8 | 8 | 2个 ICLK |
| 0008 7070h | ICU | 中断请求寄存器 112 | IR112 | 8 | 8 | 2个 ICLK |

表 5.1 I/O 寄存器地址一览表 (4 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|------|-------------|-------|----|------|----------|
| 0008 7073h | ICU | 中断请求寄存器 115 | IR115 | 8 | 8 | 2 个 ICLK |
| 0008 7074h | ICU | 中断请求寄存器 116 | IR116 | 8 | 8 | 2 个 ICLK |
| 0008 7075h | ICU | 中断请求寄存器 117 | IR117 | 8 | 8 | 2 个 ICLK |
| 0008 7076h | ICU | 中断请求寄存器 118 | IR118 | 8 | 8 | 2 个 ICLK |
| 0008 7078h | ICU | 中断请求寄存器 120 | IR120 | 8 | 8 | 2 个 ICLK |
| 0008 7079h | ICU | 中断请求寄存器 121 | IR121 | 8 | 8 | 2 个 ICLK |
| 0008 707Ah | ICU | 中断请求寄存器 122 | IR122 | 8 | 8 | 2 个 ICLK |
| 0008 707Bh | ICU | 中断请求寄存器 123 | IR123 | 8 | 8 | 2 个 ICLK |
| 0008 707Ch | ICU | 中断请求寄存器 124 | IR124 | 8 | 8 | 2 个 ICLK |
| 0008 707Dh | ICU | 中断请求寄存器 125 | IR125 | 8 | 8 | 2 个 ICLK |
| 0008 707Eh | ICU | 中断请求寄存器 126 | IR126 | 8 | 8 | 2 个 ICLK |
| 0008 707Fh | ICU | 中断请求寄存器 127 | IR127 | 8 | 8 | 2 个 ICLK |
| 0008 7080h | ICU | 中断请求寄存器 128 | IR128 | 8 | 8 | 2 个 ICLK |
| 0008 7083h | ICU | 中断请求寄存器 131 | IR131 | 8 | 8 | 2 个 ICLK |
| 0008 7084h | ICU | 中断请求寄存器 132 | IR132 | 8 | 8 | 2 个 ICLK |
| 0008 7085h | ICU | 中断请求寄存器 133 | IR133 | 8 | 8 | 2 个 ICLK |
| 0008 7086h | ICU | 中断请求寄存器 134 | IR134 | 8 | 8 | 2 个 ICLK |
| 0008 7088h | ICU | 中断请求寄存器 136 | IR136 | 8 | 8 | 2 个 ICLK |
| 0008 7089h | ICU | 中断请求寄存器 137 | IR137 | 8 | 8 | 2 个 ICLK |
| 0008 708Ah | ICU | 中断请求寄存器 138 | IR138 | 8 | 8 | 2 个 ICLK |
| 0008 708Bh | ICU | 中断请求寄存器 139 | IR139 | 8 | 8 | 2 个 ICLK |
| 0008 708Ch | ICU | 中断请求寄存器 140 | IR140 | 8 | 8 | 2 个 ICLK |
| 0008 708Dh | ICU | 中断请求寄存器 141 | IR141 | 8 | 8 | 2 个 ICLK |
| 0008 708Eh | ICU | 中断请求寄存器 142 | IR142 | 8 | 8 | 2 个 ICLK |
| 0008 7091h | ICU | 中断请求寄存器 145 | IR145 | 8 | 8 | 2 个 ICLK |
| 0008 7092h | ICU | 中断请求寄存器 146 | IR146 | 8 | 8 | 2 个 ICLK |
| 0008 7095h | ICU | 中断请求寄存器 149 | IR149 | 8 | 8 | 2 个 ICLK |
| 0008 7096h | ICU | 中断请求寄存器 150 | IR150 | 8 | 8 | 2 个 ICLK |
| 0008 7097h | ICU | 中断请求寄存器 151 | IR151 | 8 | 8 | 2 个 ICLK |
| 0008 7098h | ICU | 中断请求寄存器 152 | IR152 | 8 | 8 | 2 个 ICLK |
| 0008 709Ah | ICU | 中断请求寄存器 154 | IR154 | 8 | 8 | 2 个 ICLK |
| 0008 709Bh | ICU | 中断请求寄存器 155 | IR155 | 8 | 8 | 2 个 ICLK |
| 0008 709Ch | ICU | 中断请求寄存器 156 | IR156 | 8 | 8 | 2 个 ICLK |
| 0008 709Dh | ICU | 中断请求寄存器 157 | IR157 | 8 | 8 | 2 个 ICLK |
| 0008 709Eh | ICU | 中断请求寄存器 158 | IR158 | 8 | 8 | 2 个 ICLK |
| 0008 709Fh | ICU | 中断请求寄存器 159 | IR159 | 8 | 8 | 2 个 ICLK |
| 0008 70A0h | ICU | 中断请求寄存器 160 | IR160 | 8 | 8 | 2 个 ICLK |
| 0008 70A1h | ICU | 中断请求寄存器 161 | IR161 | 8 | 8 | 2 个 ICLK |
| 0008 70A2h | ICU | 中断请求寄存器 162 | IR162 | 8 | 8 | 2 个 ICLK |
| 0008 70A5h | ICU | 中断请求寄存器 165 | IR165 | 8 | 8 | 2 个 ICLK |
| 0008 70A6h | ICU | 中断请求寄存器 166 | IR166 | 8 | 8 | 2 个 ICLK |
| 0008 70A7h | ICU | 中断请求寄存器 167 | IR167 | 8 | 8 | 2 个 ICLK |
| 0008 70A8h | ICU | 中断请求寄存器 168 | IR168 | 8 | 8 | 2 个 ICLK |
| 0008 70AAh | ICU | 中断请求寄存器 170 | IR170 | 8 | 8 | 2 个 ICLK |
| 0008 70ABh | ICU | 中断请求寄存器 171 | IR171 | 8 | 8 | 2 个 ICLK |
| 0008 70AEh | ICU | 中断请求寄存器 174 | IR174 | 8 | 8 | 2 个 ICLK |
| 0008 70AFh | ICU | 中断请求寄存器 175 | IR175 | 8 | 8 | 2 个 ICLK |

表 5.1 I/O 寄存器地址一览表 (5 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|------|-------------|-------|----|------|----------|
| 0008 70B0h | ICU | 中断请求寄存器 176 | IR176 | 8 | 8 | 2 个 ICLK |
| 0008 70B1h | ICU | 中断请求寄存器 177 | IR177 | 8 | 8 | 2 个 ICLK |
| 0008 70B2h | ICU | 中断请求寄存器 178 | IR178 | 8 | 8 | 2 个 ICLK |
| 0008 70B3h | ICU | 中断请求寄存器 179 | IR179 | 8 | 8 | 2 个 ICLK |
| 0008 70B4h | ICU | 中断请求寄存器 180 | IR180 | 8 | 8 | 2 个 ICLK |
| 0008 70B5h | ICU | 中断请求寄存器 181 | IR181 | 8 | 8 | 2 个 ICLK |
| 0008 70B6h | ICU | 中断请求寄存器 182 | IR182 | 8 | 8 | 2 个 ICLK |
| 0008 70B7h | ICU | 中断请求寄存器 183 | IR183 | 8 | 8 | 2 个 ICLK |
| 0008 70B8h | ICU | 中断请求寄存器 184 | IR184 | 8 | 8 | 2 个 ICLK |
| 0008 70B9h | ICU | 中断请求寄存器 185 | IR185 | 8 | 8 | 2 个 ICLK |
| 0008 70C6h | ICU | 中断请求寄存器 198 | IR198 | 8 | 8 | 2 个 ICLK |
| 0008 70C7h | ICU | 中断请求寄存器 199 | IR199 | 8 | 8 | 2 个 ICLK |
| 0008 70C8h | ICU | 中断请求寄存器 200 | IR200 | 8 | 8 | 2 个 ICLK |
| 0008 70C9h | ICU | 中断请求寄存器 201 | IR201 | 8 | 8 | 2 个 ICLK |
| 0008 70D6h | ICU | 中断请求寄存器 214 | IR214 | 8 | 8 | 2 个 ICLK |
| 0008 70D7h | ICU | 中断请求寄存器 215 | IR215 | 8 | 8 | 2 个 ICLK |
| 0008 70D8h | ICU | 中断请求寄存器 216 | IR216 | 8 | 8 | 2 个 ICLK |
| 0008 70D9h | ICU | 中断请求寄存器 217 | IR217 | 8 | 8 | 2 个 ICLK |
| 0008 70DAh | ICU | 中断请求寄存器 218 | IR218 | 8 | 8 | 2 个 ICLK |
| 0008 70DBh | ICU | 中断请求寄存器 219 | IR219 | 8 | 8 | 2 个 ICLK |
| 0008 70DCh | ICU | 中断请求寄存器 220 | IR220 | 8 | 8 | 2 个 ICLK |
| 0008 70DDh | ICU | 中断请求寄存器 221 | IR221 | 8 | 8 | 2 个 ICLK |
| 0008 70DEh | ICU | 中断请求寄存器 222 | IR222 | 8 | 8 | 2 个 ICLK |
| 0008 70DFh | ICU | 中断请求寄存器 223 | IR223 | 8 | 8 | 2 个 ICLK |
| 0008 70E0h | ICU | 中断请求寄存器 224 | IR224 | 8 | 8 | 2 个 ICLK |
| 0008 70E1h | ICU | 中断请求寄存器 225 | IR225 | 8 | 8 | 2 个 ICLK |
| 0008 70E2h | ICU | 中断请求寄存器 226 | IR226 | 8 | 8 | 2 个 ICLK |
| 0008 70E3h | ICU | 中断请求寄存器 227 | IR227 | 8 | 8 | 2 个 ICLK |
| 0008 70E4h | ICU | 中断请求寄存器 228 | IR228 | 8 | 8 | 2 个 ICLK |
| 0008 70E5h | ICU | 中断请求寄存器 229 | IR229 | 8 | 8 | 2 个 ICLK |
| 0008 70E6h | ICU | 中断请求寄存器 230 | IR230 | 8 | 8 | 2 个 ICLK |
| 0008 70E7h | ICU | 中断请求寄存器 231 | IR231 | 8 | 8 | 2 个 ICLK |
| 0008 70E8h | ICU | 中断请求寄存器 232 | IR232 | 8 | 8 | 2 个 ICLK |
| 0008 70E9h | ICU | 中断请求寄存器 233 | IR233 | 8 | 8 | 2 个 ICLK |
| 0008 70EAh | ICU | 中断请求寄存器 234 | IR234 | 8 | 8 | 2 个 ICLK |
| 0008 70EBh | ICU | 中断请求寄存器 235 | IR235 | 8 | 8 | 2 个 ICLK |
| 0008 70ECh | ICU | 中断请求寄存器 236 | IR236 | 8 | 8 | 2 个 ICLK |
| 0008 70EDh | ICU | 中断请求寄存器 237 | IR237 | 8 | 8 | 2 个 ICLK |
| 0008 70EEh | ICU | 中断请求寄存器 238 | IR238 | 8 | 8 | 2 个 ICLK |
| 0008 70EFh | ICU | 中断请求寄存器 239 | IR239 | 8 | 8 | 2 个 ICLK |
| 0008 70F0h | ICU | 中断请求寄存器 240 | IR240 | 8 | 8 | 2 个 ICLK |
| 0008 70F1h | ICU | 中断请求寄存器 241 | IR241 | 8 | 8 | 2 个 ICLK |
| 0008 70F6h | ICU | 中断请求寄存器 246 | IR246 | 8 | 8 | 2 个 ICLK |
| 0008 70F7h | ICU | 中断请求寄存器 247 | IR247 | 8 | 8 | 2 个 ICLK |
| 0008 70F8h | ICU | 中断请求寄存器 248 | IR248 | 8 | 8 | 2 个 ICLK |
| 0008 70F9h | ICU | 中断请求寄存器 249 | IR249 | 8 | 8 | 2 个 ICLK |
| 0008 70FAh | ICU | 中断请求寄存器 250 | IR250 | 8 | 8 | 2 个 ICLK |

表 5.1 I/O 寄存器地址一览表 (6 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|------|-----------------|----------|----|------|----------|
| 0008 70FBh | ICU | 中断请求寄存器 251 | IR251 | 8 | 8 | 2 个 ICLK |
| 0008 70FCh | ICU | 中断请求寄存器 252 | IR252 | 8 | 8 | 2 个 ICLK |
| 0008 70FDh | ICU | 中断请求寄存器 253 | IR253 | 8 | 8 | 2 个 ICLK |
| 0008 711Ch | ICU | 中断请求目标设定寄存器 028 | ISELR028 | 8 | 8 | 2 个 ICLK |
| 0008 711Dh | ICU | 中断请求目标设定寄存器 029 | ISELR029 | 8 | 8 | 2 个 ICLK |
| 0008 711Eh | ICU | 中断请求目标设定寄存器 030 | ISELR030 | 8 | 8 | 2 个 ICLK |
| 0008 711Fh | ICU | 中断请求目标设定寄存器 031 | ISELR031 | 8 | 8 | 2 个 ICLK |
| 0008 7140h | ICU | 中断请求目标设定寄存器 064 | ISELR064 | 8 | 8 | 2 个 ICLK |
| 0008 7141h | ICU | 中断请求目标设定寄存器 065 | ISELR065 | 8 | 8 | 2 个 ICLK |
| 0008 7142h | ICU | 中断请求目标设定寄存器 066 | ISELR066 | 8 | 8 | 2 个 ICLK |
| 0008 7143h | ICU | 中断请求目标设定寄存器 067 | ISELR067 | 8 | 8 | 2 个 ICLK |
| 0008 7144h | ICU | 中断请求目标设定寄存器 068 | ISELR068 | 8 | 8 | 2 个 ICLK |
| 0008 7145h | ICU | 中断请求目标设定寄存器 069 | ISELR069 | 8 | 8 | 2 个 ICLK |
| 0008 7146h | ICU | 中断请求目标设定寄存器 070 | ISELR070 | 8 | 8 | 2 个 ICLK |
| 0008 7147h | ICU | 中断请求目标设定寄存器 071 | ISELR071 | 8 | 8 | 2 个 ICLK |
| 0008 7148h | ICU | 中断请求目标设定寄存器 072 | ISELR072 | 8 | 8 | 2 个 ICLK |
| 0008 7149h | ICU | 中断请求目标设定寄存器 073 | ISELR073 | 8 | 8 | 2 个 ICLK |
| 0008 714Ah | ICU | 中断请求目标设定寄存器 074 | ISELR074 | 8 | 8 | 2 个 ICLK |
| 0008 714Bh | ICU | 中断请求目标设定寄存器 075 | ISELR075 | 8 | 8 | 2 个 ICLK |
| 0008 714Ch | ICU | 中断请求目标设定寄存器 076 | ISELR076 | 8 | 8 | 2 个 ICLK |
| 0008 714Dh | ICU | 中断请求目标设定寄存器 077 | ISELR077 | 8 | 8 | 2 个 ICLK |
| 0008 714Eh | ICU | 中断请求目标设定寄存器 078 | ISELR078 | 8 | 8 | 2 个 ICLK |
| 0008 714Fh | ICU | 中断请求目标设定寄存器 079 | ISELR079 | 8 | 8 | 2 个 ICLK |
| 0008 7162h | ICU | 中断请求目标设定寄存器 098 | ISELR098 | 8 | 8 | 2 个 ICLK |
| 0008 7163h | ICU | 中断请求目标设定寄存器 099 | ISELR099 | 8 | 8 | 2 个 ICLK |
| 0008 7164h | ICU | 中断请求目标设定寄存器 100 | ISELR100 | 8 | 8 | 2 个 ICLK |
| 0008 7165h | ICU | 中断请求目标设定寄存器 101 | ISELR101 | 8 | 8 | 2 个 ICLK |
| 0008 7168h | ICU | 中断请求目标设定寄存器 104 | ISELR104 | 8 | 8 | 2 个 ICLK |
| 0008 7169h | ICU | 中断请求目标设定寄存器 105 | ISELR105 | 8 | 8 | 2 个 ICLK |
| 0008 716Ah | ICU | 中断请求目标设定寄存器 106 | ISELR106 | 8 | 8 | 2 个 ICLK |
| 0008 716Bh | ICU | 中断请求目标设定寄存器 107 | ISELR107 | 8 | 8 | 2 个 ICLK |
| 0008 716Fh | ICU | 中断请求目标设定寄存器 111 | ISELR111 | 8 | 8 | 2 个 ICLK |
| 0008 7170h | ICU | 中断请求目标设定寄存器 112 | ISELR112 | 8 | 8 | 2 个 ICLK |
| 0008 7175h | ICU | 中断请求目标设定寄存器 117 | ISELR117 | 8 | 8 | 2 个 ICLK |
| 0008 7176h | ICU | 中断请求目标设定寄存器 118 | ISELR118 | 8 | 8 | 2 个 ICLK |
| 0008 717Ah | ICU | 中断请求目标设定寄存器 122 | ISELR122 | 8 | 8 | 2 个 ICLK |
| 0008 717Bh | ICU | 中断请求目标设定寄存器 123 | ISELR123 | 8 | 8 | 2 个 ICLK |
| 0008 717Ch | ICU | 中断请求目标设定寄存器 124 | ISELR124 | 8 | 8 | 2 个 ICLK |
| 0008 717Dh | ICU | 中断请求目标设定寄存器 125 | ISELR125 | 8 | 8 | 2 个 ICLK |
| 0008 717Fh | ICU | 中断请求目标设定寄存器 127 | ISELR127 | 8 | 8 | 2 个 ICLK |
| 0008 7180h | ICU | 中断请求目标设定寄存器 128 | ISELR128 | 8 | 8 | 2 个 ICLK |
| 0008 7185h | ICU | 中断请求目标设定寄存器 133 | ISELR133 | 8 | 8 | 2 个 ICLK |
| 0008 7186h | ICU | 中断请求目标设定寄存器 134 | ISELR134 | 8 | 8 | 2 个 ICLK |
| 0008 718Ah | ICU | 中断请求目标设定寄存器 138 | ISELR138 | 8 | 8 | 2 个 ICLK |
| 0008 718Bh | ICU | 中断请求目标设定寄存器 139 | ISELR139 | 8 | 8 | 2 个 ICLK |
| 0008 718Ch | ICU | 中断请求目标设定寄存器 140 | ISELR140 | 8 | 8 | 2 个 ICLK |
| 0008 718Dh | ICU | 中断请求目标设定寄存器 141 | ISELR141 | 8 | 8 | 2 个 ICLK |

表 5.1 I/O 寄存器地址一览表 (7 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|------|-----------------|----------|----|------|----------|
| 0008 7191h | ICU | 中断请求目标设定寄存器 145 | ISELR145 | 8 | 8 | 2 个 ICLK |
| 0008 7192h | ICU | 中断请求目标设定寄存器 146 | ISELR146 | 8 | 8 | 2 个 ICLK |
| 0008 7197h | ICU | 中断请求目标设定寄存器 151 | ISELR151 | 8 | 8 | 2 个 ICLK |
| 0008 7198h | ICU | 中断请求目标设定寄存器 152 | ISELR152 | 8 | 8 | 2 个 ICLK |
| 0008 719Ch | ICU | 中断请求目标设定寄存器 156 | ISELR156 | 8 | 8 | 2 个 ICLK |
| 0008 719Dh | ICU | 中断请求目标设定寄存器 157 | ISELR157 | 8 | 8 | 2 个 ICLK |
| 0008 719Eh | ICU | 中断请求目标设定寄存器 158 | ISELR158 | 8 | 8 | 2 个 ICLK |
| 0008 719Fh | ICU | 中断请求目标设定寄存器 159 | ISELR159 | 8 | 8 | 2 个 ICLK |
| 0008 71A1h | ICU | 中断请求目标设定寄存器 161 | ISELR161 | 8 | 8 | 2 个 ICLK |
| 0008 71A2h | ICU | 中断请求目标设定寄存器 162 | ISELR162 | 8 | 8 | 2 个 ICLK |
| 0008 71A7h | ICU | 中断请求目标设定寄存器 167 | ISELR167 | 8 | 8 | 2 个 ICLK |
| 0008 71A8h | ICU | 中断请求目标设定寄存器 168 | ISELR168 | 8 | 8 | 2 个 ICLK |
| 0008 71AEh | ICU | 中断请求目标设定寄存器 174 | ISELR174 | 8 | 8 | 2 个 ICLK |
| 0008 71AFh | ICU | 中断请求目标设定寄存器 175 | ISELR175 | 8 | 8 | 2 个 ICLK |
| 0008 71B1h | ICU | 中断请求目标设定寄存器 177 | ISELR177 | 8 | 8 | 2 个 ICLK |
| 0008 71B2h | ICU | 中断请求目标设定寄存器 178 | ISELR178 | 8 | 8 | 2 个 ICLK |
| 0008 71B4h | ICU | 中断请求目标设定寄存器 180 | ISELR180 | 8 | 8 | 2 个 ICLK |
| 0008 71B5h | ICU | 中断请求目标设定寄存器 181 | ISELR181 | 8 | 8 | 2 个 ICLK |
| 0008 71B7h | ICU | 中断请求目标设定寄存器 183 | ISELR183 | 8 | 8 | 2 个 ICLK |
| 0008 71B8h | ICU | 中断请求目标设定寄存器 184 | ISELR184 | 8 | 8 | 2 个 ICLK |
| 0008 71C6h | ICU | 中断请求目标设定寄存器 198 | ISELR198 | 8 | 8 | 2 个 ICLK |
| 0008 71C7h | ICU | 中断请求目标设定寄存器 199 | ISELR199 | 8 | 8 | 2 个 ICLK |
| 0008 71C8h | ICU | 中断请求目标设定寄存器 200 | ISELR200 | 8 | 8 | 2 个 ICLK |
| 0008 71C9h | ICU | 中断请求目标设定寄存器 201 | ISELR201 | 8 | 8 | 2 个 ICLK |
| 0008 71D7h | ICU | 中断请求目标设定寄存器 215 | ISELR215 | 8 | 8 | 2 个 ICLK |
| 0008 71D8h | ICU | 中断请求目标设定寄存器 216 | ISELR216 | 8 | 8 | 2 个 ICLK |
| 0008 71DBh | ICU | 中断请求目标设定寄存器 219 | ISELR219 | 8 | 8 | 2 个 ICLK |
| 0008 71DCh | ICU | 中断请求目标设定寄存器 220 | ISELR220 | 8 | 8 | 2 个 ICLK |
| 0008 71DFh | ICU | 中断请求目标设定寄存器 223 | ISELR223 | 8 | 8 | 2 个 ICLK |
| 0008 71E0h | ICU | 中断请求目标设定寄存器 224 | ISELR224 | 8 | 8 | 2 个 ICLK |
| 0008 71E3h | ICU | 中断请求目标设定寄存器 227 | ISELR227 | 8 | 8 | 2 个 ICLK |
| 0008 71E4h | ICU | 中断请求目标设定寄存器 228 | ISELR228 | 8 | 8 | 2 个 ICLK |
| 0008 71E7h | ICU | 中断请求目标设定寄存器 231 | ISELR231 | 8 | 8 | 2 个 ICLK |
| 0008 71E8h | ICU | 中断请求目标设定寄存器 232 | ISELR232 | 8 | 8 | 2 个 ICLK |
| 0008 71EBh | ICU | 中断请求目标设定寄存器 235 | ISELR235 | 8 | 8 | 2 个 ICLK |
| 0008 71ECh | ICU | 中断请求目标设定寄存器 236 | ISELR236 | 8 | 8 | 2 个 ICLK |
| 0008 71EFh | ICU | 中断请求目标设定寄存器 239 | ISELR239 | 8 | 8 | 2 个 ICLK |
| 0008 71F0h | ICU | 中断请求目标设定寄存器 240 | ISELR240 | 8 | 8 | 2 个 ICLK |
| 0008 71F7h | ICU | 中断请求目标设定寄存器 247 | ISELR247 | 8 | 8 | 2 个 ICLK |
| 0008 71F8h | ICU | 中断请求目标设定寄存器 248 | ISELR248 | 8 | 8 | 2 个 ICLK |
| 0008 71FBh | ICU | 中断请求目标设定寄存器 251 | ISELR251 | 8 | 8 | 2 个 ICLK |
| 0008 71FCh | ICU | 中断请求目标设定寄存器 252 | ISELR252 | 8 | 8 | 2 个 ICLK |
| 0008 7202h | ICU | 中断请求允许寄存器 02 | IER02 | 8 | 8 | 2 个 ICLK |
| 0008 7203h | ICU | 中断请求允许寄存器 03 | IER03 | 8 | 8 | 2 个 ICLK |
| 0008 7208h | ICU | 中断请求允许寄存器 08 | IER08 | 8 | 8 | 2 个 ICLK |
| 0008 7209h | ICU | 中断请求允许寄存器 09 | IER09 | 8 | 8 | 2 个 ICLK |
| 0008 720Ch | ICU | 中断请求允许寄存器 0C | IER0C | 8 | 8 | 2 个 ICLK |

表 5.1 I/O 寄存器地址一览表 (8 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|------|--------------|-------|----|------|----------|
| 0008 720Dh | ICU | 中断请求允许寄存器 0D | IER0D | 8 | 8 | 2 个 ICLK |
| 0008 720Eh | ICU | 中断请求允许寄存器 0E | IER0E | 8 | 8 | 2 个 ICLK |
| 0008 720Fh | ICU | 中断请求允许寄存器 0F | IER0F | 8 | 8 | 2 个 ICLK |
| 0008 7210h | ICU | 中断请求允许寄存器 10 | IER10 | 8 | 8 | 2 个 ICLK |
| 0008 7211h | ICU | 中断请求允许寄存器 11 | IER11 | 8 | 8 | 2 个 ICLK |
| 0008 7212h | ICU | 中断请求允许寄存器 12 | IER12 | 8 | 8 | 2 个 ICLK |
| 0008 7213h | ICU | 中断请求允许寄存器 13 | IER13 | 8 | 8 | 2 个 ICLK |
| 0008 7214h | ICU | 中断请求允许寄存器 14 | IER14 | 8 | 8 | 2 个 ICLK |
| 0008 7215h | ICU | 中断请求允许寄存器 15 | IER15 | 8 | 8 | 2 个 ICLK |
| 0008 7216h | ICU | 中断请求允许寄存器 16 | IER16 | 8 | 8 | 2 个 ICLK |
| 0008 7217h | ICU | 中断请求允许寄存器 17 | IER17 | 8 | 8 | 2 个 ICLK |
| 0008 7218h | ICU | 中断请求允许寄存器 18 | IER18 | 8 | 8 | 2 个 ICLK |
| 0008 7219h | ICU | 中断请求允许寄存器 19 | IER19 | 8 | 8 | 2 个 ICLK |
| 0008 721Ah | ICU | 中断请求允许寄存器 1A | IER1A | 8 | 8 | 2 个 ICLK |
| 0008 721Bh | ICU | 中断请求允许寄存器 1B | IER1B | 8 | 8 | 2 个 ICLK |
| 0008 721Ch | ICU | 中断请求允许寄存器 1C | IER1C | 8 | 8 | 2 个 ICLK |
| 0008 721Dh | ICU | 中断请求允许寄存器 1D | IER1D | 8 | 8 | 2 个 ICLK |
| 0008 721Eh | ICU | 中断请求允许寄存器 1E | IER1E | 8 | 8 | 2 个 ICLK |
| 0008 721Fh | ICU | 中断请求允许寄存器 1F | IER1F | 8 | 8 | 2 个 ICLK |
| 0008 7300h | ICU | 中断源优先级寄存器 00 | IPR00 | 8 | 8 | 2 个 ICLK |
| 0008 7301h | ICU | 中断源优先级寄存器 01 | IPR01 | 8 | 8 | 2 个 ICLK |
| 0008 7302h | ICU | 中断源优先级寄存器 02 | IPR02 | 8 | 8 | 2 个 ICLK |
| 0008 7304h | ICU | 中断源优先级寄存器 04 | IPR04 | 8 | 8 | 2 个 ICLK |
| 0008 7305h | ICU | 中断源优先级寄存器 05 | IPR05 | 8 | 8 | 2 个 ICLK |
| 0008 7306h | ICU | 中断源优先级寄存器 06 | IPR06 | 8 | 8 | 2 个 ICLK |
| 0008 7307h | ICU | 中断源优先级寄存器 07 | IPR07 | 8 | 8 | 2 个 ICLK |
| 0008 7320h | ICU | 中断源优先级寄存器 20 | IPR20 | 8 | 8 | 2 个 ICLK |
| 0008 7321h | ICU | 中断源优先级寄存器 21 | IPR21 | 8 | 8 | 2 个 ICLK |
| 0008 7322h | ICU | 中断源优先级寄存器 22 | IPR22 | 8 | 8 | 2 个 ICLK |
| 0008 7323h | ICU | 中断源优先级寄存器 23 | IPR23 | 8 | 8 | 2 个 ICLK |
| 0008 7324h | ICU | 中断源优先级寄存器 24 | IPR24 | 8 | 8 | 2 个 ICLK |
| 0008 7325h | ICU | 中断源优先级寄存器 25 | IPR25 | 8 | 8 | 2 个 ICLK |
| 0008 7326h | ICU | 中断源优先级寄存器 26 | IPR26 | 8 | 8 | 2 个 ICLK |
| 0008 7327h | ICU | 中断源优先级寄存器 27 | IPR27 | 8 | 8 | 2 个 ICLK |
| 0008 7328h | ICU | 中断源优先级寄存器 28 | IPR28 | 8 | 8 | 2 个 ICLK |
| 0008 7329h | ICU | 中断源优先级寄存器 29 | IPR29 | 8 | 8 | 2 个 ICLK |
| 0008 732Ah | ICU | 中断源优先级寄存器 2A | IPR2A | 8 | 8 | 2 个 ICLK |
| 0008 732Bh | ICU | 中断源优先级寄存器 2B | IPR2B | 8 | 8 | 2 个 ICLK |
| 0008 732Ch | ICU | 中断源优先级寄存器 2C | IPR2C | 8 | 8 | 2 个 ICLK |
| 0008 732Dh | ICU | 中断源优先级寄存器 2D | IPR2D | 8 | 8 | 2 个 ICLK |
| 0008 732Eh | ICU | 中断源优先级寄存器 2E | IPR2E | 8 | 8 | 2 个 ICLK |
| 0008 732Fh | ICU | 中断源优先级寄存器 2F | IPR2F | 8 | 8 | 2 个 ICLK |
| 0008 7340h | ICU | 中断源优先级寄存器 40 | IPR40 | 8 | 8 | 2 个 ICLK |
| 0008 7344h | ICU | 中断源优先级寄存器 44 | IPR44 | 8 | 8 | 2 个 ICLK |
| 0008 7345h | ICU | 中断源优先级寄存器 45 | IPR45 | 8 | 8 | 2 个 ICLK |
| 0008 7346h | ICU | 中断源优先级寄存器 46 | IPR46 | 8 | 8 | 2 个 ICLK |
| 0008 7347h | ICU | 中断源优先级寄存器 47 | IPR47 | 8 | 8 | 2 个 ICLK |

表 5.1 I/O 寄存器地址一览表 (9 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|------|--------------|-------|----|------|----------|
| 0008 734Ch | ICU | 中断源优先级寄存器 4C | IPR4C | 8 | 8 | 2 个 ICLK |
| 0008 734Dh | ICU | 中断源优先级寄存器 4D | IPR4D | 8 | 8 | 2 个 ICLK |
| 0008 734Eh | ICU | 中断源优先级寄存器 4E | IPR4E | 8 | 8 | 2 个 ICLK |
| 0008 734Fh | ICU | 中断源优先级寄存器 4F | IPR4F | 8 | 8 | 2 个 ICLK |
| 0008 7350h | ICU | 中断源优先级寄存器 50 | IPR50 | 8 | 8 | 2 个 ICLK |
| 0008 7351h | ICU | 中断源优先级寄存器 51 | IPR51 | 8 | 8 | 2 个 ICLK |
| 0008 7352h | ICU | 中断源优先级寄存器 52 | IPR52 | 8 | 8 | 2 个 ICLK |
| 0008 7353h | ICU | 中断源优先级寄存器 53 | IPR53 | 8 | 8 | 2 个 ICLK |
| 0008 7354h | ICU | 中断源优先级寄存器 54 | IPR54 | 8 | 8 | 2 个 ICLK |
| 0008 7355h | ICU | 中断源优先级寄存器 55 | IPR55 | 8 | 8 | 2 个 ICLK |
| 0008 7356h | ICU | 中断源优先级寄存器 56 | IPR56 | 8 | 8 | 2 个 ICLK |
| 0008 7357h | ICU | 中断源优先级寄存器 57 | IPR57 | 8 | 8 | 2 个 ICLK |
| 0008 7358h | ICU | 中断源优先级寄存器 58 | IPR58 | 8 | 8 | 2 个 ICLK |
| 0008 7359h | ICU | 中断源优先级寄存器 59 | IPR59 | 8 | 8 | 2 个 ICLK |
| 0008 735Ah | ICU | 中断源优先级寄存器 5A | IPR5A | 8 | 8 | 2 个 ICLK |
| 0008 735Bh | ICU | 中断源优先级寄存器 5B | IPR5B | 8 | 8 | 2 个 ICLK |
| 0008 735Ch | ICU | 中断源优先级寄存器 5C | IPR5C | 8 | 8 | 2 个 ICLK |
| 0008 735Dh | ICU | 中断源优先级寄存器 5D | IPR5D | 8 | 8 | 2 个 ICLK |
| 0008 735Eh | ICU | 中断源优先级寄存器 5E | IPR5E | 8 | 8 | 2 个 ICLK |
| 0008 735Fh | ICU | 中断源优先级寄存器 5F | IPR5F | 8 | 8 | 2 个 ICLK |
| 0008 7360h | ICU | 中断源优先级寄存器 60 | IPR60 | 8 | 8 | 2 个 ICLK |
| 0008 7361h | ICU | 中断源优先级寄存器 61 | IPR61 | 8 | 8 | 2 个 ICLK |
| 0008 7362h | ICU | 中断源优先级寄存器 62 | IPR62 | 8 | 8 | 2 个 ICLK |
| 0008 7363h | ICU | 中断源优先级寄存器 63 | IPR63 | 8 | 8 | 2 个 ICLK |
| 0008 7368h | ICU | 中断源优先级寄存器 68 | IPR68 | 8 | 8 | 2 个 ICLK |
| 0008 7369h | ICU | 中断源优先级寄存器 69 | IPR69 | 8 | 8 | 2 个 ICLK |
| 0008 736Ah | ICU | 中断源优先级寄存器 6A | IPR6A | 8 | 8 | 2 个 ICLK |
| 0008 736Bh | ICU | 中断源优先级寄存器 6B | IPR6B | 8 | 8 | 2 个 ICLK |
| 0008 7370h | ICU | 中断源优先级寄存器 70 | IPR70 | 8 | 8 | 2 个 ICLK |
| 0008 7371h | ICU | 中断源优先级寄存器 71 | IPR71 | 8 | 8 | 2 个 ICLK |
| 0008 7372h | ICU | 中断源优先级寄存器 72 | IPR72 | 8 | 8 | 2 个 ICLK |
| 0008 7373h | ICU | 中断源优先级寄存器 73 | IPR73 | 8 | 8 | 2 个 ICLK |
| 0008 7380h | ICU | 中断源优先级寄存器 80 | IPR80 | 8 | 8 | 2 个 ICLK |
| 0008 7381h | ICU | 中断源优先级寄存器 81 | IPR81 | 8 | 8 | 2 个 ICLK |
| 0008 7382h | ICU | 中断源优先级寄存器 82 | IPR82 | 8 | 8 | 2 个 ICLK |
| 0008 7383h | ICU | 中断源优先级寄存器 83 | IPR83 | 8 | 8 | 2 个 ICLK |
| 0008 7384h | ICU | 中断源优先级寄存器 84 | IPR84 | 8 | 8 | 2 个 ICLK |
| 0008 7385h | ICU | 中断源优先级寄存器 85 | IPR85 | 8 | 8 | 2 个 ICLK |
| 0008 7386h | ICU | 中断源优先级寄存器 86 | IPR86 | 8 | 8 | 2 个 ICLK |
| 0008 7388h | ICU | 中断源优先级寄存器 88 | IPR88 | 8 | 8 | 2 个 ICLK |
| 0008 7389h | ICU | 中断源优先级寄存器 89 | IPR89 | 8 | 8 | 2 个 ICLK |
| 0008 738Ah | ICU | 中断源优先级寄存器 8A | IPR8A | 8 | 8 | 2 个 ICLK |
| 0008 738Bh | ICU | 中断源优先级寄存器 8B | IPR8B | 8 | 8 | 2 个 ICLK |
| 0008 738Ch | ICU | 中断源优先级寄存器 8C | IPR8C | 8 | 8 | 2 个 ICLK |
| 0008 738Dh | ICU | 中断源优先级寄存器 8D | IPR8D | 8 | 8 | 2 个 ICLK |
| 0008 738Eh | ICU | 中断源优先级寄存器 8E | IPR8E | 8 | 8 | 2 个 ICLK |
| 0008 738Fh | ICU | 中断源优先级寄存器 8F | IPR8F | 8 | 8 | 2 个 ICLK |

表 5.1 I/O 寄存器地址一览表 (10/20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|------------|-----------------|----------|----|------|-----------------|
| 0008 73F0h | ICU | 高速中断设定寄存器 | FIR | 16 | 16 | 2 个 ICLK |
| 0008 7400h | DTC | DTC 控制寄存器 | DTCCR | 8 | 8 | 2 个 ICLK |
| 0008 7404h | DTC | DTC 向量基址寄存器 | DTCVBR | 32 | 32 | 2 个 ICLK |
| 0008 7408h | DTC | DTC 地址模式寄存器 | DTCADMOD | 8 | 8 | 2 个 ICLK |
| 0008 740Ch | DTC | DTC 模块启动寄存器 | DTCST | 8 | 8 | 2 个 ICLK |
| 0008 8000h | CMT (单元 0) | 比较匹配定时器的启动寄存器 0 | CMSTR0 | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8002h | CMT0 | 比较匹配定时器的控制寄存器 | CMCR | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8004h | CMT0 | 比较匹配定时器的计数器 | CMCNT | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8006h | CMT0 | 比较匹配定时器的常数寄存器 | CMCOR | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8008h | CMT1 | 比较匹配定时器的控制寄存器 | CMCR | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 800Ah | CMT1 | 比较匹配定时器的计数器 | CMCNT | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 800Ch | CMT1 | 比较匹配定时器的常数寄存器 | CMCOR | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8010h | CMT (单元 1) | 比较匹配定时器的启动寄存器 1 | CMSTR1 | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8012h | CMT2 | 比较匹配定时器的控制寄存器 | CMCR | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8014h | CMT2 | 比较匹配定时器的计数器 | CMCNT | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8016h | CMT2 | 比较匹配定时器的常数寄存器 | CMCOR | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8018h | CMT3 | 比较匹配定时器的控制寄存器 | CMCR | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 801Ah | CMT3 | 比较匹配定时器的计数器 | CMCNT | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 801Ch | CMT3 | 比较匹配定时器的常数寄存器 | CMCOR | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8028h | WDT | 定时器的控制 / 状态寄存器 | TCSR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8028h | WDT | 写窗口 A 寄存器 | WINA | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8029h | WDT | 定时器的计数器 | TCNT | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 802Ah | WDT | 写窗口 B 寄存器 | WINB | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 802Bh | WDT | 复位的控制 / 状态寄存器 | RSTCSR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8040h | AD0 | A/D 数据寄存器 A | ADDRA | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8042h | AD0 | A/D 数据寄存器 B | ADDRB | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8044h | AD0 | A/D 数据寄存器 C | ADDRC | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8046h | AD0 | A/D 数据寄存器 D | ADDRD | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8050h | AD0 | A/D 控制 / 状态寄存器 | ADCSR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8051h | AD0 | A/D 控制寄存器 | ADCR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8052h | AD0 | ADDRy 格式选择寄存器 | ADDPR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8053h | AD0 | A/D 采样状态寄存器 | ADSSTR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8060h | AD1 | A/D 数据寄存器 A | ADDRA | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8062h | AD1 | A/D 数据寄存器 B | ADDRB | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8064h | AD1 | A/D 数据寄存器 C | ADDRC | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8066h | AD1 | A/D 数据寄存器 D | ADDRD | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8070h | AD1 | A/D 控制 / 状态寄存器 | ADCSR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8071h | AD1 | A/D 控制寄存器 | ADCR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8072h | AD1 | ADDRy 格式选择寄存器 | ADDPR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8073h | AD1 | A/D 采样状态寄存器 | ADSSTR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8080h | AD2 | A/D 数据寄存器 A | ADDRA | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8082h | AD2 | A/D 数据寄存器 B | ADDRB | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8084h | AD2 | A/D 数据寄存器 C | ADDRC | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8086h | AD2 | A/D 数据寄存器 D | ADDRD | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 8090h | AD2 | A/D 控制 / 状态寄存器 | ADCSR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8091h | AD2 | A/D 控制寄存器 | ADCR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8092h | AD2 | ADDRy 格式选择寄存器 | ADDPR | 8 | 8 | 2~3 个 PCLK (注7) |

表 5.1 I/O 寄存器地址一览表 (11 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|------------|------------------|--------|----|------|---------------|
| 0008 8093h | AD2 | A/D 采样状态寄存器 | ADSSTR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 80A0h | AD3 | A/D 数据寄存器 A | ADDRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 80A2h | AD3 | A/D 数据寄存器 B | ADDRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 80A4h | AD3 | A/D 数据寄存器 C | ADDRC | 16 | 16 | 2~3个PCLK (注7) |
| 0008 80A6h | AD3 | A/D 数据寄存器 D | ADDRD | 16 | 16 | 2~3个PCLK (注7) |
| 0008 80B0h | AD3 | A/D 控制 / 状态寄存器 | ADCSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 80B1h | AD3 | A/D 控制寄存器 | ADCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 80B2h | AD3 | ADDRy 格式选择寄存器 | ADDPY | 8 | 8 | 2~3个PCLK (注7) |
| 0008 80B3h | AD3 | A/D 采样状态寄存器 | ADSSTR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 80C0h | D/A | D/A 数据寄存器 0 | DADR0 | 16 | 16 | 2~3个PCLK (注7) |
| 0008 80C2h | D/A | D/A 数据寄存器 1 | DADR1 | 16 | 16 | 2~3个PCLK (注7) |
| 0008 80C4h | D/A | D/A 控制寄存器 | DACR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 80C5h | D/A | DADRy 格式选择寄存器 | DADPY | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8100h | TPU (单元 0) | 定时器的启动寄存器 | TSTRA | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8101h | TPU (单元 0) | 定时器的同步寄存器 | TSYRA | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8110h | TPU0 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8111h | TPU0 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8112h | TPU0 | 定时器的 I/O 控制寄存器 H | TIORH | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8113h | TPU0 | 定时器的 I/O 控制寄存器 L | TIORL | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8114h | TPU0 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8115h | TPU0 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8116h | TPU0 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8118h | TPU0 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 811Ah | TPU0 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 811Ch | TPU0 | 定时器的通用寄存器 C | TGRC | 16 | 16 | 2~3个PCLK (注7) |
| 0008 811Eh | TPU0 | 定时器的通用寄存器 D | TGRD | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8120h | TPU1 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8121h | TPU1 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8122h | TPU1 | 定时器的 I/O 控制寄存器 | TIOR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8124h | TPU1 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8125h | TPU1 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8126h | TPU1 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8128h | TPU1 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 812Ah | TPU1 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8130h | TPU2 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8131h | TPU2 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8132h | TPU2 | 定时器的 I/O 控制寄存器 | TIOR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8134h | TPU2 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8135h | TPU2 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8136h | TPU2 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8138h | TPU2 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 813Ah | TPU2 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8140h | TPU3 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8141h | TPU3 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8142h | TPU3 | 定时器的 I/O 控制寄存器 H | TIORH | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8143h | TPU3 | 定时器的 I/O 控制寄存器 L | TIORL | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8144h | TPU3 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |

表 5.1 I/O 寄存器地址一览表 (12 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|-----------|------------------|-------|----|------|---------------|
| 0008 8145h | TPU3 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8146h | TPU3 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8148h | TPU3 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 814Ah | TPU3 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 814Ch | TPU3 | 定时器的通用寄存器 C | TGRC | 16 | 16 | 2~3个PCLK (注7) |
| 0008 814Eh | TPU3 | 定时器的通用寄存器 D | TGRD | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8150h | TPU4 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8151h | TPU4 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8152h | TPU4 | 定时器的 I/O 控制寄存器 | TIOR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8154h | TPU4 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8155h | TPU4 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8156h | TPU4 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8158h | TPU4 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 815Ah | TPU4 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8160h | TPU5 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8161h | TPU5 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8162h | TPU5 | 定时器的 I/O 控制寄存器 | TIOR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8164h | TPU5 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8165h | TPU5 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8166h | TPU5 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8168h | TPU5 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 816Ah | TPU5 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8170h | TPU(单元 1) | 定时器的启动寄存器 | TSTRB | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8171h | TPU(单元 1) | 定时器的同步寄存器 | TSYRB | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8180h | TPU6 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8181h | TPU6 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8182h | TPU6 | 定时器的 I/O 控制寄存器 H | TIORH | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8183h | TPU6 | 定时器的 I/O 控制寄存器 L | TIORL | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8184h | TPU6 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8185h | TPU6 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8186h | TPU6 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8188h | TPU6 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 818Ah | TPU6 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 818Ch | TPU6 | 定时器的通用寄存器 C | TGRC | 16 | 16 | 2~3个PCLK (注7) |
| 0008 818Eh | TPU6 | 定时器的通用寄存器 D | TGRD | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8190h | TPU7 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8191h | TPU7 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8192h | TPU7 | 定时器的 I/O 控制寄存器 | TIOR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8194h | TPU7 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8195h | TPU7 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8196h | TPU7 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8198h | TPU7 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 819Ah | TPU7 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81A0h | TPU8 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81A1h | TPU8 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81A2h | TPU8 | 定时器的 I/O 控制寄存器 | TIOR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81A4h | TPU8 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |

表 5.1 I/O 寄存器地址一览表 (13 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|-----------------|-------|------------------|--------|----|------|---------------|
| 0008 81A5h | TPU8 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81A6h | TPU8 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81A8h | TPU8 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81AAh | TPU8 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81B0h | TPU9 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81B1h | TPU9 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81B2h | TPU9 | 定时器的 I/O 控制寄存器 H | TIORH | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81B3h | TPU9 | 定时器的 I/O 控制寄存器 L | TIORL | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81B4h | TPU9 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81B5h | TPU9 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81B6h | TPU9 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81B8h | TPU9 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81BAh | TPU9 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81BCh | TPU9 | 定时器的通用寄存器 C | TGRC | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81BEh | TPU9 | 定时器的通用寄存器 D | TGRD | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81C0h | TPU10 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81C1h | TPU10 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81C2h | TPU10 | 定时器的 I/O 控制寄存器 | TIOR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81C4h | TPU10 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81C5h | TPU10 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81C6h | TPU10 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81C8h | TPU10 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81CAh | TPU10 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81D0h | TPU11 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81D1h | TPU11 | 定时器的模式寄存器 | TMDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81D2h | TPU11 | 定时器的 I/O 控制寄存器 | TIOR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81D4h | TPU11 | 定时器的中断允许寄存器 | TIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81D5h | TPU11 | 定时器的状态寄存器 | TSR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81D6h | TPU11 | 定时器的计数器 | TCNT | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81D8h | TPU11 | 定时器的通用寄存器 A | TGRA | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81DAh | TPU11 | 定时器的通用寄存器 B | TGRB | 16 | 16 | 2~3个PCLK (注7) |
| 0008 81E6h | PPG0 | PPG 输出控制寄存器 | PCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81E7h | PPG0 | PPG 输出模式寄存器 | PMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81E8h | PPG0 | 下一个数据允许寄存器 H | NDERH | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81E9h | PPG0 | 下一个数据允许寄存器 L | NDERL | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81EAh | PPG0 | 输出数据寄存器 H | PODRH | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81EBh | PPG0 | 输出数据寄存器 L | PODRL | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81ECh (注1) | PPG0 | 下一个数据寄存器 H | NDRH | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81EDh (注2) | PPG0 | 下一个数据寄存器 L | NDRL | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81EEh (注1) | PPG0 | 下一个数据寄存器 H | NDRH | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81EFh (注2) | PPG0 | 下一个数据寄存器 L | NDRL | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81F0h | PPG1 | PPG 触发选择寄存器 | PTRSLR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81F6h | PPG1 | PPG 输出控制寄存器 | PCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81F7h | PPG1 | PPG 输出模式寄存器 | PMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81F8h | PPG1 | 下一个数据允许寄存器 H | NDERH | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81F9h | PPG1 | 下一个数据允许寄存器 L | NDERL | 8 | 8 | 2~3个PCLK (注7) |
| 0008 81FAh | PPG1 | 输出数据寄存器 H | PODRH | 8 | 8 | 2~3个PCLK (注7) |

表 5.1 I/O 寄存器地址一览表 (14 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|-----------------|------|----------------|----------|----|--------------|-----------------|
| 0008 81FBh | PPG1 | 输出数据寄存器 L | PODRL | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 81FCh (注3) | PPG1 | 下一个数据寄存器 H | NDRH | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 81FDh (注4) | PPG1 | 下一个数据寄存器 L | NDRL | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 81FEh (注3) | PPG1 | 下一个数据寄存器 H | NDRH | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 81FFh (注4) | PPG1 | 下一个数据寄存器 L | NDRL | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8200h | TMR0 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8201h | TMR1 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8202h | TMR0 | 定时器的控制 / 状态寄存器 | TCSR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8203h | TMR1 | 定时器的控制 / 状态寄存器 | TCSR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8204h | TMR0 | 时间常数寄存器 A | TCORA | 8 | 8 或者 16 | 2~3 个 PCLK (注7) |
| 0008 8205h | TMR1 | 时间常数寄存器 A | TCORA | 8 | 8 或者 16 (注5) | 2~3 个 PCLK (注7) |
| 0008 8206h | TMR0 | 时间常数寄存器 B | TCORB | 8 | 8 或者 16 | 2~3 个 PCLK (注7) |
| 0008 8207h | TMR1 | 时间常数寄存器 B | TCORB | 8 | 8 或者 16 (注5) | 2~3 个 PCLK (注7) |
| 0008 8208h | TMR0 | 定时器的计数器 | TCNT | 8 | 8 或者 16 | 2~3 个 PCLK (注7) |
| 0008 8209h | TMR1 | 定时器的计数器 | TCNT | 8 | 8 或者 16 (注5) | 2~3 个 PCLK (注7) |
| 0008 820Ah | TMR0 | 定时器的计数器控制寄存器 | TCCR | 8 | 8 或者 16 | 2~3 个 PCLK (注7) |
| 0008 820Bh | TMR1 | 定时器的计数器控制寄存器 | TCCR | 8 | 8 或者 16 | 2~3 个 PCLK (注7) |
| 0008 8210h | TMR2 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8211h | TMR3 | 定时器的控制寄存器 | TCR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8212h | TMR2 | 定时器的控制 / 状态寄存器 | TCSR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8213h | TMR3 | 定时器的控制 / 状态寄存器 | TCSR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8214h | TMR2 | 时间常数寄存器 A | TCORA | 8 | 8 或者 16 | 2~3 个 PCLK (注7) |
| 0008 8215h | TMR3 | 时间常数寄存器 A | TCORA | 8 | 8 或者 16 (注5) | 2~3 个 PCLK (注7) |
| 0008 8216h | TMR2 | 时间常数寄存器 B | TCORB | 8 | 8 或者 16 | 2~3 个 PCLK (注7) |
| 0008 8217h | TMR3 | 时间常数寄存器 B | TCORB | 8 | 8 或者 16 (注5) | 2~3 个 PCLK (注7) |
| 0008 8218h | TMR2 | 定时器的计数器 | TCNT | 8 | 8 或者 16 | 2~3 个 PCLK (注7) |
| 0008 8219h | TMR3 | 定时器的计数器 | TCNT | 8 | 8 或者 16 (注5) | 2~3 个 PCLK (注7) |
| 0008 821Ah | TMR2 | 定时器的计数器控制寄存器 | TCCR | 8 | 8 或者 16 | 2~3 个 PCLK (注7) |
| 0008 821Bh | TMR3 | 定时器的计数器控制寄存器 | TCCR | 8 | 8 或者 16 | 2~3 个 PCLK (注7) |
| 0008 8240h | SCI0 | 串行模式寄存器 | SMR (注6) | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8241h | SCI0 | 位速率寄存器 | BRR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8242h | SCI0 | 串行控制寄存器 | SCR (注6) | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8243h | SCI0 | 发送数据寄存器 | TDR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8244h | SCI0 | 串行状态寄存器 | SSR (注6) | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8245h | SCI0 | 接收数据寄存器 | RDR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8246h | SCI0 | 智能卡模式寄存器 | SCMR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8247h | SCI0 | 串行扩展模式寄存器 | SEMR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8248h | SCI1 | 串行模式寄存器 | SMR (注6) | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8249h | SCI1 | 位速率寄存器 | BRR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 824Ah | SCI1 | 串行控制寄存器 | SCR (注6) | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 824Bh | SCI1 | 发送数据寄存器 | TDR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 824Ch | SCI1 | 串行状态寄存器 | SSR (注6) | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 824Dh | SCI1 | 接收数据寄存器 | RDR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 824Eh | SCI1 | 智能卡模式寄存器 | SCMR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 824Fh | SCI1 | 串行扩展模式寄存器 | SEMR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8250h | SCI2 | 串行模式寄存器 | SMR (注6) | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 8251h | SCI2 | 位速率寄存器 | BRR | 8 | 8 | 2~3 个 PCLK (注7) |

表 5.1 I/O 寄存器地址一览表 (15 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|-------|----------------------------|----------|----|------|---------------|
| 0008 8252h | SCI2 | 串行控制寄存器 | SCR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8253h | SCI2 | 发送数据寄存器 | TDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8254h | SCI2 | 串行状态寄存器 | SSR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8255h | SCI2 | 接收数据寄存器 | RDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8256h | SCI2 | 智能卡模式寄存器 | SCMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8257h | SCI2 | 串行扩展模式寄存器 | SEMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8258h | SCI3 | 串行模式寄存器 | SMR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8259h | SCI3 | 位速率寄存器 | BRR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 825Ah | SCI3 | 串行控制寄存器 | SCR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 825Bh | SCI3 | 发送数据寄存器 | TDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 825Ch | SCI3 | 串行状态寄存器 | SSR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 825Dh | SCI3 | 接收数据寄存器 | RDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 825Eh | SCI3 | 智能卡模式寄存器 | SCMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 825Fh | SCI3 | 串行扩展模式寄存器 | SEMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8260h | SCI4 | 串行模式寄存器 | SMR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8261h | SCI4 | 位速率寄存器 | BRR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8262h | SCI4 | 串行控制寄存器 | SCR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8263h | SCI4 | 发送数据寄存器 | TDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8264h | SCI4 | 串行状态寄存器 | SSR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8265h | SCI4 | 接收数据寄存器 | RDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8266h | SCI4 | 智能卡模式寄存器 | SCMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8267h | SCI4 | 串行扩展模式寄存器 | SEMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8268h | SCI5 | 串行模式寄存器 | SMR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8269h | SCI5 | 位速率寄存器 | BRR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 826Ah | SCI5 | 串行控制寄存器 | SCR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 826Bh | SCI5 | 发送数据寄存器 | TDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 826Ch | SCI5 | 串行状态寄存器 | SSR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 826Dh | SCI5 | 接收数据寄存器 | RDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 826Eh | SCI5 | 智能卡模式寄存器 | SCMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 826Fh | SCI5 | 串行扩展模式寄存器 | SEMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8270h | SCI6 | 串行模式寄存器 | SMR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8271h | SCI6 | 位速率寄存器 | BRR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8272h | SCI6 | 串行控制寄存器 | SCR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8273h | SCI6 | 发送数据寄存器 | TDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8274h | SCI6 | 串行状态寄存器 | SSR (注6) | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8275h | SCI6 | 接收数据寄存器 | RDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8276h | SCI6 | 智能卡模式寄存器 | SCMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8277h | SCI6 | 串行扩展模式寄存器 | SEMR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8280h | CRC | CRC 控制寄存器 | CRCCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8281h | CRC | CRC 数据输入寄存器 | CRCDIR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8282h | CRC | CRC 数据输出寄存器 | CRCDOR | 16 | 16 | 2~3个PCLK (注7) |
| 0008 8300h | RIIC0 | I ² C 总线控制寄存器 1 | ICCR1 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8301h | RIIC0 | I ² C 总线控制寄存器 2 | ICCR2 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8302h | RIIC0 | I ² C 总线模式寄存器 1 | ICMR1 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8303h | RIIC0 | I ² C 总线模式寄存器 2 | ICMR2 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8304h | RIIC0 | I ² C 总线模式寄存器 3 | ICMR3 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8305h | RIIC0 | I ² C 总线功能允许寄存器 | ICFER | 8 | 8 | 2~3个PCLK (注7) |

表 5.1 I/O 寄存器地址一览表 (16 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|-------|------------------------------|-------|----|------|---------------|
| 0008 8306h | RIIC0 | I ² C 总线状态允许寄存器 | ICSER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8307h | RIIC0 | I ² C 总线中断允许寄存器 | ICIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8308h | RIIC0 | I ² C 总线状态寄存器 1 | ICSR1 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8309h | RIIC0 | I ² C 总线状态寄存器 2 | ICSR2 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 830Ah | RIIC0 | 从属地址寄存器 L0 | SARL0 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 830Bh | RIIC0 | 从属地址寄存器 U0 | SARU0 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 830Ch | RIIC0 | 从属地址寄存器 L1 | SARL1 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 830Dh | RIIC0 | 从属地址寄存器 U1 | SARU1 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 830Eh | RIIC0 | 从属地址寄存器 L2 | SARL2 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 830Fh | RIIC0 | 从属地址寄存器 U2 | SARU2 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8310h | RIIC0 | I ² C 总线位速率低电平寄存器 | ICBRL | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8311h | RIIC0 | I ² C 总线位速率高电平寄存器 | ICBRH | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8312h | RIIC0 | I ² C 总线发送数据寄存器 | ICDRT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8313h | RIIC0 | I ² C 总线接收数据寄存器 | ICDRR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8320h | RIIC1 | I ² C 总线控制寄存器 1 | ICCR1 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8321h | RIIC1 | I ² C 总线控制寄存器 2 | ICCR2 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8322h | RIIC1 | I ² C 总线模式寄存器 1 | ICMR1 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8323h | RIIC1 | I ² C 总线模式寄存器 2 | ICMR2 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8324h | RIIC1 | I ² C 总线模式寄存器 3 | ICMR3 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8325h | RIIC1 | I ² C 总线功能允许寄存器 | ICFER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8326h | RIIC1 | I ² C 总线状态允许寄存器 | ICSER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8327h | RIIC1 | I ² C 总线中断允许寄存器 | ICIER | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8328h | RIIC1 | I ² C 总线状态寄存器 1 | ICSR1 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8329h | RIIC1 | I ² C 总线状态寄存器 2 | ICSR2 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 832Ah | RIIC1 | 从属地址寄存器 L0 | SARL0 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 832Bh | RIIC1 | 从属地址寄存器 U0 | SARU0 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 832Ch | RIIC1 | 从属地址寄存器 L1 | SARL1 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 832Dh | RIIC1 | 从属地址寄存器 U1 | SARU1 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 832Eh | RIIC1 | 从属地址寄存器 L2 | SARL2 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 832Fh | RIIC1 | 从属地址寄存器 U2 | SARU2 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8330h | RIIC1 | I ² C 总线位速率低电平寄存器 | ICBRL | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8331h | RIIC1 | I ² C 总线位速率高电平寄存器 | ICBRH | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8332h | RIIC1 | I ² C 总线发送数据寄存器 | ICDRT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 8333h | RIIC1 | I ² C 总线接收数据寄存器 | ICDRR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C000h | P0 | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C001h | P1 | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C002h | P2 | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C003h | P3 | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C004h | P4 | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C005h | P5 | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C006h | P6 | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C007h | P7 | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C008h | P8 | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C009h | P9 | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C00Ah | PA | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C00Bh | PB | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C00Ch | PC | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |

表 5.1 I/O 寄存器地址一览表 (17 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|------|-----------|-------|----|------|---------------|
| 0008 C00Dh | PD | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C00Eh | PE | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C00Fh | PF | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C010h | PG | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C011h | PH | 数据方向寄存器 | DDR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C020h | P0 | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C021h | P1 | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C022h | P2 | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C023h | P3 | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C024h | P4 | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C025h | P5 | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C026h | P6 | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C027h | P7 | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C028h | P8 | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C029h | P9 | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C02Ah | PA | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C02Bh | PB | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C02Ch | PC | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C02Dh | PD | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C02Eh | PE | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C02Fh | PF | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C030h | PG | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C031h | PH | 数据寄存器 | DR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C040h | P0 | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C041h | P1 | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C042h | P2 | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C043h | P3 | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C044h | P4 | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C045h | P5 | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C046h | P6 | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C047h | P7 | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C048h | P8 | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C049h | P9 | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C04Ah | PA | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C04Bh | PB | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C04Ch | PC | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C04Dh | PD | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C04Eh | PE | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C04Fh | PF | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C050h | PG | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C051h | PH | 端口寄存器 | PORT | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C060h | P0 | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C061h | P1 | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C062h | P2 | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C063h | P3 | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C064h | P4 | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C065h | P5 | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |

表 5.1 I/O 寄存器地址一览表 (18 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|----------|----------------|----------|----|------|---------------|
| 0008 C066h | P6 | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C067h | P7 | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C068h | P8 | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C069h | P9 | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C06Ah | PA | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C06Bh | PB | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C06Ch | PC | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C06Dh | PD | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C06Eh | PE | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C06Fh | PF | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C070h | PG | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C071h | PH | 输入缓冲控制寄存器 | ICR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C082h | P2 | 漏极开路控制寄存器 | ODR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C08Ch | PC | 漏极开路控制寄存器 | ODR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C0CAh | PA | 上拉电阻控制寄存器 | PCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C0CBh | PB | 上拉电阻控制寄存器 | PCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C0CCh | PC | 上拉电阻控制寄存器 | PCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C0CDh | PD | 上拉电阻控制寄存器 | PCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C0CEh | PE | 上拉电阻控制寄存器 | PCR | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C100h | I/O PORT | 端口功能控制寄存器 0 | PFCR0 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C101h | I/O PORT | 端口功能控制寄存器 1 | PFCR1 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C102h | I/O PORT | 端口功能控制寄存器 2 | PFCR2 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C103h | I/O PORT | 端口功能控制寄存器 3 | PFCR3 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C104h | I/O PORT | 端口功能控制寄存器 4 | PFCR4 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C105h | I/O PORT | 端口功能控制寄存器 5 | PFCR5 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C106h | I/O PORT | 端口功能控制寄存器 6 | PFCR6 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C107h | I/O PORT | 端口功能控制寄存器 7 | PFCR7 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C108h | I/O PORT | 端口功能控制寄存器 8 | PFCR8 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C109h | I/O PORT | 端口功能控制寄存器 9 | PFCR9 | 8 | 8 | 2~3个PCLK (注7) |
| 0008 C280h | SYSTEM | 深度待机控制寄存器 | DPSBYCR | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C281h | SYSTEM | 深度待机等待控制寄存器 | DPSWCR | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C282h | SYSTEM | 深度待机中断允许寄存器 | DPSIER | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C283h | SYSTEM | 深度待机中断标志寄存器 | DPSIFR | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C284h | SYSTEM | 深度待机中断边沿寄存器 | DPSIEGR | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C285h | SYSTEM | 复位状态寄存器 | RSTSR | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C289h | FLASH | 闪存编程 / 擦除保护寄存器 | FWEPROR | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C290h | SYSTEM | 深度待机备用寄存器 0 | DPSBKR0 | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C291h | SYSTEM | 深度待机备用寄存器 1 | DPSBKR1 | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C292h | SYSTEM | 深度待机备用寄存器 2 | DPSBKR2 | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C293h | SYSTEM | 深度待机备用寄存器 3 | DPSBKR3 | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C294h | SYSTEM | 深度待机备用寄存器 4 | DPSBKR4 | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C295h | SYSTEM | 深度待机备用寄存器 5 | DPSBKR5 | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C296h | SYSTEM | 深度待机备用寄存器 6 | DPSBKR6 | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C297h | SYSTEM | 深度待机备用寄存器 7 | DPSBKR7 | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C298h | SYSTEM | 深度待机备用寄存器 8 | DPSBKR8 | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C299h | SYSTEM | 深度待机备用寄存器 9 | DPSBKR9 | 8 | 8 | 4~5个PCLK (注7) |
| 0008 C29Ah | SYSTEM | 深度待机备用寄存器 10 | DPSBKR10 | 8 | 8 | 4~5个PCLK (注7) |

表 5.1 I/O 寄存器地址一览表 (19/20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|--------|----------------|----------|----|------|----------------|
| 0008 C29Bh | SYSTEM | 深度待机备用寄存器 11 | DPSBKR11 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C29Ch | SYSTEM | 深度待机备用寄存器 12 | DPSBKR12 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C29Dh | SYSTEM | 深度待机备用寄存器 13 | DPSBKR13 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C29Eh | SYSTEM | 深度待机备用寄存器 14 | DPSBKR14 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C29Fh | SYSTEM | 深度待机备用寄存器 15 | DPSBKR15 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2A0h | SYSTEM | 深度待机备用寄存器 16 | DPSBKR16 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2A1h | SYSTEM | 深度待机备用寄存器 17 | DPSBKR17 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2A2h | SYSTEM | 深度待机备用寄存器 18 | DPSBKR18 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2A3h | SYSTEM | 深度待机备用寄存器 19 | DPSBKR19 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2A4h | SYSTEM | 深度待机备用寄存器 20 | DPSBKR20 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2A5h | SYSTEM | 深度待机备用寄存器 21 | DPSBKR21 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2A6h | SYSTEM | 深度待机备用寄存器 22 | DPSBKR22 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2A7h | SYSTEM | 深度待机备用寄存器 23 | DPSBKR23 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2A8h | SYSTEM | 深度待机备用寄存器 24 | DPSBKR24 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2A9h | SYSTEM | 深度待机备用寄存器 25 | DPSBKR25 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2AAh | SYSTEM | 深度待机备用寄存器 26 | DPSBKR26 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2ABh | SYSTEM | 深度待机备用寄存器 27 | DPSBKR27 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2ACh | SYSTEM | 深度待机备用寄存器 28 | DPSBKR28 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2ADh | SYSTEM | 深度待机备用寄存器 29 | DPSBKR29 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2AEh | SYSTEM | 深度待机备用寄存器 30 | DPSBKR30 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C2AFh | SYSTEM | 深度待机备用寄存器 31 | DPSBKR31 | 8 | 8 | 4~5个 PCLK (注7) |
| 0008 C300h | ICU | IRQ 检测允许寄存器 0 | IRQER0 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C301h | ICU | IRQ 检测允许寄存器 1 | IRQER1 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C302h | ICU | IRQ 检测允许寄存器 2 | IRQER2 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C303h | ICU | IRQ 检测允许寄存器 3 | IRQER3 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C304h | ICU | IRQ 检测允许寄存器 4 | IRQER4 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C305h | ICU | IRQ 检测允许寄存器 5 | IRQER5 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C306h | ICU | IRQ 检测允许寄存器 6 | IRQER6 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C307h | ICU | IRQ 检测允许寄存器 7 | IRQER7 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C308h | ICU | IRQ 检测允许寄存器 8 | IRQER8 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C309h | ICU | IRQ 检测允许寄存器 9 | IRQER9 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C30Ah | ICU | IRQ 检测允许寄存器 10 | IRQER10 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C30Bh | ICU | IRQ 检测允许寄存器 11 | IRQER11 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C30Ch | ICU | IRQ 检测允许寄存器 12 | IRQER12 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C30Dh | ICU | IRQ 检测允许寄存器 13 | IRQER13 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C30Eh | ICU | IRQ 检测允许寄存器 14 | IRQER14 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C30Fh | ICU | IRQ 检测允许寄存器 15 | IRQER15 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C320h | ICU | IRQ 控制寄存器 0 | IRQCR0 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C321h | ICU | IRQ 控制寄存器 1 | IRQCR1 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C322h | ICU | IRQ 控制寄存器 2 | IRQCR2 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C323h | ICU | IRQ 控制寄存器 3 | IRQCR3 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C324h | ICU | IRQ 控制寄存器 4 | IRQCR4 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C325h | ICU | IRQ 控制寄存器 5 | IRQCR5 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C326h | ICU | IRQ 控制寄存器 6 | IRQCR6 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C327h | ICU | IRQ 控制寄存器 7 | IRQCR7 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C328h | ICU | IRQ 控制寄存器 8 | IRQCR8 | 8 | 8 | 2~3个 PCLK (注7) |
| 0008 C329h | ICU | IRQ 控制寄存器 9 | IRQCR9 | 8 | 8 | 2~3个 PCLK (注7) |

表 5.1 I/O 寄存器地址一览表 (20 / 20)

| 地址 | 模块符号 | 寄存器名 | 寄存器符号 | 位数 | 存取长度 | 存取周期数 |
|------------|-------|------------------|-----------|----|------|-----------------|
| 0008 C32Ah | ICU | IRQ 控制寄存器 10 | IRQCR10 | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 C32Bh | ICU | IRQ 控制寄存器 11 | IRQCR11 | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 C32Ch | ICU | IRQ 控制寄存器 12 | IRQCR12 | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 C32Dh | ICU | IRQ 控制寄存器 13 | IRQCR13 | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 C32Eh | ICU | IRQ 控制寄存器 14 | IRQCR14 | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 C32Fh | ICU | IRQ 控制寄存器 15 | IRQCR15 | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 C340h | ICU | 软件待机解除 IRQ 允许寄存器 | SSIER | 16 | 16 | 2~3 个 PCLK (注7) |
| 0008 C350h | ICU | 非屏蔽中断允许寄存器 | NMIER | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 C351h | ICU | NMI 引脚中断控制寄存器 | NMICR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 C352h | ICU | 非屏蔽中断状态寄存器 | NMISR | 8 | 8 | 2~3 个 PCLK (注7) |
| 0008 C353h | ICU | 非屏蔽中断清除寄存器 | NMICLR | 8 | 8 | 2~3 个 PCLK (注7) |
| 007F C402h | FLASH | 闪存模式寄存器 | FMODR | 8 | 8 | 2~3 个 PCLK (注7) |
| 007F C410h | FLASH | 闪存存取状态寄存器 | FASTAT | 8 | 8 | 2~3 个 PCLK (注7) |
| 007F C411h | FLASH | 闪存存取错误中断允许寄存器 | FAEINT | 8 | 8 | 2~3 个 PCLK (注7) |
| 007F C412h | FLASH | 闪存就绪中断允许寄存器 | FRDYIE | 8 | 8 | 2~3 个 PCLK (注7) |
| 007F C440h | FLASH | 数据闪存读允许寄存器 | DFLRE | 16 | 16 | 2~3 个 PCLK (注7) |
| 007F C450h | FLASH | 数据闪存编程 / 擦除允许寄存器 | DFLWE | 16 | 16 | 2~3 个 PCLK (注7) |
| 007F C454h | FLASH | FCU RAM 允许寄存器 | FCURAME | 16 | 16 | 2~3 个 PCLK (注7) |
| 007F FFB0h | FLASH | 闪存状态寄存器 0 | FSTATR0 | 8 | 8 | 2~3 个 PCLK (注7) |
| 007F FFB1h | FLASH | 闪存状态寄存器 1 | FSTATR1 | 8 | 8 | 2~3 个 PCLK (注7) |
| 007F FFB2h | FLASH | 闪存 P/E 模式入口寄存器 | FENTRYR | 16 | 16 | 2~3 个 PCLK (注7) |
| 007F FFB4h | FLASH | 闪存保护寄存器 | FPROTR | 16 | 16 | 2~3 个 PCLK (注7) |
| 007F FFB6h | FLASH | 闪存复位寄存器 | FRESETR | 16 | 16 | 2~3 个 PCLK (注7) |
| 007F FFBAh | FLASH | FCU 命令寄存器 | FCMDR | 16 | 16 | 2~3 个 PCLK (注7) |
| 007F FFC8h | FLASH | FCU 处理转换寄存器 | FCPSR | 16 | 16 | 2~3 个 PCLK (注7) |
| 007F FFCAh | FLASH | 数据闪存空白检查控制寄存器 | DFLBCCNT | 16 | 16 | 2~3 个 PCLK (注7) |
| 007F FFCh | FLASH | 闪存 P/E 状态寄存器 | FPESTAT | 16 | 16 | 2~3 个 PCLK (注7) |
| 007F FFCEh | FLASH | 数据闪存空白检查状态寄存器 | DFLBCSTAT | 16 | 16 | 2~3 个 PCLK (注7) |
| 007F FFE8h | FLASH | 外围时钟通知寄存器 | PCKAR | 16 | 16 | 2~3 个 PCLK (注7) |

- 注 1. 通过设定 PPG0.PCR, 当脉冲输出组 2 和脉冲输出组 3 的输出触发设定值相同时, PPG0.NDRH 的地址为 000881ECh; 当输出触发设定值不同时, 对应脉冲输出组 2 的 PPG0.NDRH 的地址为 000881EEh, 对应脉冲输出组 3 的 PPG0.NDRH 的地址为 000881ECh。
- 注 2. 通过设定 PPG0.PCR, 当脉冲输出组 0 和脉冲输出组 1 的输出触发设定值相同时, PPG0.NDRL 的地址为 000881EDh; 当输出触发设定值不同时, 对应脉冲输出组 0 的 PPG0.NDRL 的地址为 000881EFh, 对应脉冲输出组 1 的 PPG0.NDRL 的地址为 000881EDh。
- 注 3. 通过设定 PPG1.PCR, 当脉冲输出组 6 和脉冲输出组 7 的输出触发设定值相同时, PPG1.NDRH 的地址为 000881FCh; 当输出触发设定值不同时, 对应脉冲输出组 6 的 PPG1.NDRH 的地址为 000881FEh, 对应脉冲输出组 7 的 PPG1.NDRH 的地址为 000881FCh。
- 注 4. 通过设定 PPG1.PCR, 当脉冲输出组 4 和脉冲输出组 5 的输出触发设定值相同时, PPG1.NDRL 的地址为 000881FDh; 当输出触发设定值不同时, 对应脉冲输出组 4 的 PPG1.NDRL 的地址为 000881FFh, 对应脉冲输出组 5 的 PPG1.NDRL 的地址为 000881FDh。
- 注 5. 不能对奇数地址进行 16 位的存取。要对寄存器进行 16 位的存取时, 必须对 TMR0 或者 TMR2 的寄存器的地址进行存取。
- 注 6. 在串行通信模式和智能卡接口模式中, 部分位的功能不同。
- 注 7. 根据分频时钟的同步周期 (0~1 个 PCLK) 发生变化。
- 注 8. 如果在 DMAC 运行中存取寄存器, 就有可能变为 5 个 ICLK。

5.2 I/O 寄存器位一览表

外围模块的寄存器的地址和位名如下所示。

16 位或者 32 位的寄存器分别以 8 位分 2 段或者 4 段表示。

表 5.2 I/O 寄存器位一览表 (1 / 26)

| 模块符号 | 寄存器符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|--------|---------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| SYSTEM | MDMONR | — | — | — | — | — | — | — | — |
| | | MDE | — | — | — | — | — | MD1 | MD0 |
| SYSTEM | MDSR | — | — | — | — | — | — | — | — |
| | | — | UBTS | — | BOTS | BSW[1:0] | | EXB | IROM |
| SYSTEM | SYSCR0 | KEY[7:0] | | | | | | | |
| | | — | — | — | — | — | — | EXBE | ROME |
| SYSTEM | SYSCR1 | — | — | — | — | — | — | — | — |
| | | — | — | — | — | — | — | — | RAME |
| SYSTEM | SBYCR | SSBY | OPE | — | STS[4:0] | | | | |
| | | — | — | — | — | — | — | — | — |
| SYSTEM | MSTPCRA | ACSE | — | — | MSTPA28 | MSTPA27 | — | — | — |
| | | MSTPA23 | MSTPA22 | MSTPA21 | MSTPA20 | MSTPA19 | — | — | — |
| | | MSTPA15 | MSTPA14 | MSTPA13 | MSTPA12 | MSTPA11 | MSTPA10 | — | — |
| | | — | — | MSTPA5 | MSTPA4 | — | — | — | — |
| SYSTEM | MSTPCRB | MSTPB31 | MSTPB30 | MSTPB29 | MSTPB28 | MSTPB27 | MSTPB26 | MSTPB25 | — |
| | | MSTPB23 | — | MSTPB21 | MSTPB20 | — | — | — | — |
| | | — | — | — | — | — | — | — | — |
| | | — | — | — | — | — | — | — | — |
| SYSTEM | MSTPCRC | — | — | — | — | — | — | — | — |
| | | — | — | — | — | — | — | — | — |
| | | — | — | — | — | — | — | — | — |
| | | — | — | — | — | — | — | MSTPC1 | MSTPC0 |
| SYSTEM | SCKCR | — | — | — | — | ICK[3:0] | | | |
| | | PSTOP1 | — | — | — | BCK[3:0] | | | |
| | | — | — | — | — | PCK[3:0] | | | |
| | | — | — | — | — | — | — | — | — |
| BSC | BERCLR | — | — | — | — | — | — | — | STSCLR |
| BSC | BEREN | — | — | — | — | — | — | TOEN | IGAEN |
| BSC | BERIE | — | — | — | — | — | — | — | CPEN |
| DMAC0 | DMCSA | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC0 | DMCDA | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC0 | DMCBC | — | — | — | — | — | — | — | — |
| | | | | | | | | | |
| | | | | | | | | | |

表 5.2 I/O 寄存器位一览表 (2 / 26)

| 模块符号 | 寄存器符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|-------|-------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| DMAC0 | DMMOD | — | — | — | — | OPSEL[3:0] | | | |
| | | — | — | — | — | — | SZSEL[2:0] | | |
| | | — | SMOD[2:0] | | | — | DMOD[2:0] | | |
| | | — | — | — | — | — | — | — | — |
| DMAC1 | DMCSA | | | | | | | | |
| DMAC1 | DMCDA | | | | | | | | |
| DMAC1 | DMCBC | — | — | — | — | — | — | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC1 | DMMOD | — | — | — | — | OPSEL[3:0] | | | |
| | | — | — | — | — | — | SZSEL[2:0] | | |
| | | — | SMOD[2:0] | | | — | DMOD[2:0] | | |
| | | — | — | — | — | — | — | — | — |
| DMAC2 | DMCSA | | | | | | | | |
| DMAC2 | DMCDA | | | | | | | | |
| DMAC2 | DMCBC | — | — | — | — | — | — | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC2 | DMMOD | — | — | — | — | OPSEL[3:0] | | | |
| | | — | — | — | — | — | SZSEL[2:0] | | |
| | | — | SMOD[2:0] | | | — | DMOD[2:0] | | |
| | | — | — | — | — | — | — | — | — |
| DMAC3 | DMCSA | | | | | | | | |
| DMAC3 | DMCDA | | | | | | | | |
| DMAC3 | DMCBC | — | — | — | — | — | — | | |
| | | | | | | | | | |
| | | | | | | | | | |

表 5.2 I/O 寄存器位一览表 (3 / 26)

| 模块符号 | 寄存器符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|-------|-------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| DMAC3 | DMMOD | — | — | — | — | OPSEL[3:0] | | | |
| | | — | — | — | — | — | SZSEL[2:0] | | |
| | | — | SMOD[2:0] | | | — | DMOD[2:0] | | |
| | | — | — | — | — | — | — | — | — |
| DMAC0 | DMRSA | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC0 | DMRDA | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC0 | DMRBC | — | — | — | — | — | — | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC1 | DMRSA | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC1 | DMRDA | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC1 | DMRBC | — | — | — | — | — | — | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC2 | DMRSA | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC2 | DMRDA | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC2 | DMRBC | — | — | — | — | — | — | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC3 | DMRSA | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |
| DMAC3 | DMRDA | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

表 5.2 I/O 寄存器位一览表 (4 / 26)

| 模块符号 | 寄存器符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 | |
|---------|----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|-------|
| DMAC3 | DMRBC | — | — | — | — | — | — | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| DMAC0 | DMCRA | — | — | — | — | — | — | DSEL[1:0] | | |
| | | — | — | — | — | — | — | — | — | |
| | | — | — | — | — | — | — | BRL0D | SRLOD | DRLOD |
| | | — | — | DCTG[5:0] | | | | | | |
| DMAC0 | DMCRB | — | — | — | — | — | — | — | DSCLR | |
| DMAC0 | DMCRC | — | — | — | — | — | — | — | ECLR | |
| DMAC0 | DMCRD | — | — | — | — | — | — | — | DREQ | |
| DMAC0 | DMCRE | — | — | — | — | — | — | — | DEN | |
| DMAC1 | DMCRA | — | — | — | — | — | — | DSEL[1:0] | | |
| | | — | — | — | — | — | — | — | — | |
| | | — | — | — | — | — | — | BRL0D | SRLOD | DRLOD |
| | | — | — | DCTG[5:0] | | | | | | |
| DMAC1 | DMCRB | — | — | — | — | — | — | — | DSCLR | |
| DMAC1 | DMCRC | — | — | — | — | — | — | — | ECLR | |
| DMAC1 | DMCRD | — | — | — | — | — | — | — | DREQ | |
| DMAC1 | DMCRE | — | — | — | — | — | — | — | DEN | |
| DMAC2 | DMCRA | — | — | — | — | — | — | DSEL[1:0] | | |
| | | — | — | — | — | — | — | — | — | |
| | | — | — | — | — | — | — | BRL0D | SRLOD | DRLOD |
| | | — | — | DCTG[5:0] | | | | | | |
| DMAC2 | DMCRB | — | — | — | — | — | — | — | DSCLR | |
| DMAC2 | DMCRC | — | — | — | — | — | — | — | ECLR | |
| DMAC2 | DMCRD | — | — | — | — | — | — | — | DREQ | |
| DMAC2 | DMCRE | — | — | — | — | — | — | — | DEN | |
| DMAC3 | DMCRA | — | — | — | — | — | — | DSEL[1:0] | | |
| | | — | — | — | — | — | — | — | — | |
| | | — | — | — | — | — | — | BRL0D | SRLOD | DRLOD |
| | | — | — | DCTG[5:0] | | | | | | |
| DMAC3 | DMCRB | — | — | — | — | — | — | — | DSCLR | |
| DMAC3 | DMCRC | — | — | — | — | — | — | — | ECLR | |
| DMAC3 | DMCRD | — | — | — | — | — | — | — | DREQ | |
| DMAC3 | DMCRE | — | — | — | — | — | — | — | DEN | |
| DMAC 通用 | DMSCNT | — | — | — | — | — | — | — | DMST | |
| DMAC 通用 | DMICNT | DINTM0 | DINTM1 | DINTM2 | DINTM3 | — | — | — | — | |
| DMAC 通用 | DMEDET | DEDET0 | DEDET1 | DEDET2 | DEDET3 | — | — | — | — | |
| DMAC 通用 | DMASTS | DASTS0 | DASTS1 | DASTS2 | DASTS3 | — | — | — | — | |
| BSC | CS0MOD | PRMOD | — | — | — | — | — | PWENB | PRENB | |
| | | — | — | — | — | EWENB | — | — | WRMOD | |
| BSC | CS0WCNT1 | — | — | — | CSRWAIT[4:0] | | | | | |
| | | — | — | — | CSWWAIT[4:0] | | | | | |
| | | — | — | — | — | — | CSPRWAIT[2:0] | | | |
| | | — | — | — | — | — | CSPWWAIT[2:0] | | | |

表 5.2 I/O 寄存器位一览表 (5 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| BSC | CS0WCNT2 | — | CSON[2:0] | | | — | WDON[2:0] | | |
| | | — | WRON[2:0] | | | — | RDON[2:0] | | |
| | | — | — | — | — | — | WDOFF[2:0] | | |
| | | — | CSWOFF[2:0] | | | — | CSROFF[2:0] | | |
| BSC | CS1MOD | PRMOD | — | — | — | — | — | PWENB | PRENB |
| | | — | — | — | — | EWENB | — | — | WRMOD |
| BSC | CS1WCNT1 | — | — | — | CSRWAIT[4:0] | | | | |
| | | — | — | — | CSWWAIT[4:0] | | | | |
| | | — | — | — | — | — | CSPRWAIT[2:0] | | |
| | | — | — | — | — | — | CSPWWAIT[2:0] | | |
| BSC | CS1WCNT2 | — | CSON[2:0] | | | — | WDON[2:0] | | |
| | | — | WRON[2:0] | | | — | RDON[2:0] | | |
| | | — | — | — | — | — | WDOFF[2:0] | | |
| | | — | CSWOFF[2:0] | | | — | CSROFF[2:0] | | |
| BSC | CS2MOD | PRMOD | — | — | — | — | — | PWENB | PRENB |
| | | — | — | — | — | EWENB | — | — | WRMOD |
| BSC | CS2WCNT1 | — | — | — | CSRWAIT[4:0] | | | | |
| | | — | — | — | CSWWAIT[4:0] | | | | |
| | | — | — | — | — | — | CSPRWAIT[2:0] | | |
| | | — | — | — | — | — | CSPWWAIT[2:0] | | |
| BSC | CS2WCNT2 | — | CSON[2:0] | | | — | WDON[2:0] | | |
| | | — | WRON[2:0] | | | — | RDON[2:0] | | |
| | | — | — | — | — | — | WDOFF[2:0] | | |
| | | — | CSWOFF[2:0] | | | — | CSROFF[2:0] | | |
| BSC | CS3MOD | PRMOD | — | — | — | — | — | PWENB | PRENB |
| | | — | — | — | — | EWENB | — | — | WRMOD |
| BSC | CS3WCNT1 | — | — | — | CSRWAIT[4:0] | | | | |
| | | — | — | — | CSWWAIT[4:0] | | | | |
| | | — | — | — | — | — | CSPRWAIT[2:0] | | |
| | | — | — | — | — | — | CSPWWAIT[2:0] | | |
| BSC | CS3WCNT2 | — | CSON[2:0] | | | — | WDON[2:0] | | |
| | | — | WRON[2:0] | | | — | RDON[2:0] | | |
| | | — | — | — | — | — | WDOFF[2:0] | | |
| | | — | CSWOFF[2:0] | | | — | CSROFF[2:0] | | |
| BSC | CS4MOD | PRMOD | — | — | — | — | — | PWENB | PRENB |
| | | — | — | — | — | EWENB | — | — | WRMOD |
| BSC | CS4WCNT1 | — | — | — | CSRWAIT[4:0] | | | | |
| | | — | — | — | CSWWAIT[4:0] | | | | |
| | | — | — | — | — | — | CSPRWAIT[2:0] | | |
| | | — | — | — | — | — | CSPWWAIT[2:0] | | |
| BSC | CS4WCNT2 | — | CSON[2:0] | | | — | WDON[2:0] | | |
| | | — | WRON[2:0] | | | — | RDON[2:0] | | |
| | | — | — | — | — | — | WDOFF[2:0] | | |
| | | — | CSWOFF[2:0] | | | — | CSROFF[2:0] | | |
| BSC | CS5MOD | PRMOD | — | — | — | — | — | PWENB | PRENB |
| | | — | — | — | — | EWENB | — | — | WRMOD |

表 5.2 I/O 寄存器位一览表 (6 / 26)

| 模块符号 | 寄存器符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 | |
|------|----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|-------|
| BSC | CS5WCNT1 | — | — | — | CSRWAIT[4:0] | | | | | |
| | | — | — | — | CSWWAIT[4:0] | | | | | |
| | | — | — | — | — | — | CSPRWAIT[2:0] | | | |
| | | — | — | — | — | — | CSPWWAIT[2:0] | | | |
| BSC | CS5WCNT2 | — | CSON[2:0] | | | — | WDON[2:0] | | | |
| | | — | WRON[2:0] | | | — | RDON[2:0] | | | |
| | | — | — | — | — | — | WDOFF[2:0] | | | |
| | | — | CSWOFF[2:0] | | | — | CSROFF[2:0] | | | |
| BSC | CS6MOD | PRMOD | — | — | — | — | — | PWENB | PRENB | |
| | | — | — | — | — | EWENB | — | — | WRMOD | |
| BSC | CS6WCNT1 | — | — | — | CSRWAIT[4:0] | | | | | |
| | | — | — | — | CSWWAIT[4:0] | | | | | |
| | | — | — | — | — | — | CSPRWAIT[2:0] | | | |
| | | — | — | — | — | — | CSPWWAIT[2:0] | | | |
| BSC | CS6WCNT2 | — | CSON[2:0] | | | — | WDON[2:0] | | | |
| | | — | WRON[2:0] | | | — | RDON[2:0] | | | |
| | | — | — | — | — | — | WDOFF[2:0] | | | |
| | | — | CSWOFF[2:0] | | | — | CSROFF[2:0] | | | |
| BSC | CS7MOD | PRMOD | — | — | — | — | — | PWENB | PRENB | |
| | | — | — | — | — | EWENB | — | — | WRMOD | |
| BSC | CS7WCNT1 | — | — | — | CSRWAIT[4:0] | | | | | |
| | | — | — | — | CSWWAIT[4:0] | | | | | |
| | | — | — | — | — | — | CSPRWAIT[2:0] | | | |
| | | — | — | — | — | — | CSPWWAIT[2:0] | | | |
| BSC | CS7WCNT2 | — | CSON[2:0] | | | — | WDON[2:0] | | | |
| | | — | WRON[2:0] | | | — | RDON[2:0] | | | |
| | | — | — | — | — | — | WDOFF[2:0] | | | |
| | | — | CSWOFF[2:0] | | | — | CSROFF[2:0] | | | |
| BSC | CS0CNT | — | — | — | — | — | — | — | EMODE | |
| | | — | — | BSIZE[1:0] | | | — | — | — | EXENB |
| BSC | CS0REC | — | — | — | — | WRCV[3:0] | | | | |
| | | — | — | — | — | RRCV[3:0] | | | | |
| BSC | CS1CNT | — | — | — | — | — | — | — | EMODE | |
| | | — | — | BSIZE[1:0] | | | — | — | — | EXENB |
| BSC | CS1REC | — | — | — | — | WRCV[3:0] | | | | |
| | | — | — | — | — | RRCV[3:0] | | | | |
| BSC | CS2CNT | — | — | — | — | — | — | — | EMODE | |
| | | — | — | BSIZE[1:0] | | | — | — | — | EXENB |
| BSC | CS2REC | — | — | — | — | WRCV[3:0] | | | | |
| | | — | — | — | — | RRCV[3:0] | | | | |
| BSC | CS3CNT | — | — | — | — | — | — | — | EMODE | |
| | | — | — | BSIZE[1:0] | | | — | — | — | EXENB |
| BSC | CS3REC | — | — | — | — | WRCV[3:0] | | | | |
| | | — | — | — | — | RRCV[3:0] | | | | |
| BSC | CS4CNT | — | — | — | — | — | — | — | EMODE | |
| | | — | — | BSIZE[1:0] | | | — | — | — | EXENB |
| BSC | CS4REC | — | — | — | — | WRCV[3:0] | | | | |
| | | — | — | — | — | RRCV[3:0] | | | | |

表 5.2 I/O 寄存器位一览表 (7 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| BSC | CS5CNT | — | — | — | — | — | — | — | EMODE |
| | | — | — | BSIZE[1:0] | | — | — | — | EXENB |
| BSC | CS5REC | — | — | — | — | WRCV[3:0] | | | |
| | | — | — | — | — | RRCV[3:0] | | | |
| BSC | CS6CNT | — | — | — | — | — | — | — | EMODE |
| | | — | — | BSIZE[1:0] | | — | — | — | EXENB |
| BSC | CS6REC | — | — | — | — | WRCV[3:0] | | | |
| | | — | — | — | — | RRCV[3:0] | | | |
| BSC | CS7CNT | — | — | — | — | — | — | — | EMODE |
| | | — | — | BSIZE[1:0] | | — | — | — | EXENB |
| BSC | CS7REC | — | — | — | — | WRCV[3:0] | | | |
| | | — | — | — | — | RRCV[3:0] | | | |
| ICU | IR016 | — | — | — | — | — | — | — | IR |
| ICU | IR021 | — | — | — | — | — | — | — | IR |
| ICU | IR023 | — | — | — | — | — | — | — | IR |
| ICU | IR028 | — | — | — | — | — | — | — | IR |
| ICU | IR029 | — | — | — | — | — | — | — | IR |
| ICU | IR030 | — | — | — | — | — | — | — | IR |
| ICU | IR031 | — | — | — | — | — | — | — | IR |
| ICU | IR064 | — | — | — | — | — | — | — | IR |
| ICU | IR065 | — | — | — | — | — | — | — | IR |
| ICU | IR066 | — | — | — | — | — | — | — | IR |
| ICU | IR067 | — | — | — | — | — | — | — | IR |
| ICU | IR068 | — | — | — | — | — | — | — | IR |
| ICU | IR069 | — | — | — | — | — | — | — | IR |
| ICU | IR070 | — | — | — | — | — | — | — | IR |
| ICU | IR071 | — | — | — | — | — | — | — | IR |
| ICU | IR072 | — | — | — | — | — | — | — | IR |
| ICU | IR073 | — | — | — | — | — | — | — | IR |
| ICU | IR074 | — | — | — | — | — | — | — | IR |
| ICU | IR075 | — | — | — | — | — | — | — | IR |
| ICU | IR076 | — | — | — | — | — | — | — | IR |
| ICU | IR077 | — | — | — | — | — | — | — | IR |
| ICU | IR078 | — | — | — | — | — | — | — | IR |
| ICU | IR079 | — | — | — | — | — | — | — | IR |
| ICU | IR096 | — | — | — | — | — | — | — | IR |
| ICU | IR098 | — | — | — | — | — | — | — | IR |
| ICU | IR099 | — | — | — | — | — | — | — | IR |
| ICU | IR100 | — | — | — | — | — | — | — | IR |
| ICU | IR101 | — | — | — | — | — | — | — | IR |
| ICU | IR104 | — | — | — | — | — | — | — | IR |
| ICU | IR105 | — | — | — | — | — | — | — | IR |
| ICU | IR106 | — | — | — | — | — | — | — | IR |
| ICU | IR107 | — | — | — | — | — | — | — | IR |
| ICU | IR108 | — | — | — | — | — | — | — | IR |
| ICU | IR111 | — | — | — | — | — | — | — | IR |
| ICU | IR112 | — | — | — | — | — | — | — | IR |
| ICU | IR115 | — | — | — | — | — | — | — | IR |

表 5.2 I/O 寄存器位一览表 (8 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| ICU | IR116 | — | — | — | — | — | — | — | IR |
| ICU | IR117 | — | — | — | — | — | — | — | IR |
| ICU | IR118 | — | — | — | — | — | — | — | IR |
| ICU | IR120 | — | — | — | — | — | — | — | IR |
| ICU | IR121 | — | — | — | — | — | — | — | IR |
| ICU | IR122 | — | — | — | — | — | — | — | IR |
| ICU | IR123 | — | — | — | — | — | — | — | IR |
| ICU | IR124 | — | — | — | — | — | — | — | IR |
| ICU | IR125 | — | — | — | — | — | — | — | IR |
| ICU | IR126 | — | — | — | — | — | — | — | IR |
| ICU | IR127 | — | — | — | — | — | — | — | IR |
| ICU | IR128 | — | — | — | — | — | — | — | IR |
| ICU | IR131 | — | — | — | — | — | — | — | IR |
| ICU | IR132 | — | — | — | — | — | — | — | IR |
| ICU | IR133 | — | — | — | — | — | — | — | IR |
| ICU | IR134 | — | — | — | — | — | — | — | IR |
| ICU | IR136 | — | — | — | — | — | — | — | IR |
| ICU | IR137 | — | — | — | — | — | — | — | IR |
| ICU | IR138 | — | — | — | — | — | — | — | IR |
| ICU | IR139 | — | — | — | — | — | — | — | IR |
| ICU | IR140 | — | — | — | — | — | — | — | IR |
| ICU | IR141 | — | — | — | — | — | — | — | IR |
| ICU | IR142 | — | — | — | — | — | — | — | IR |
| ICU | IR145 | — | — | — | — | — | — | — | IR |
| ICU | IR146 | — | — | — | — | — | — | — | IR |
| ICU | IR149 | — | — | — | — | — | — | — | IR |
| ICU | IR150 | — | — | — | — | — | — | — | IR |
| ICU | IR151 | — | — | — | — | — | — | — | IR |
| ICU | IR152 | — | — | — | — | — | — | — | IR |
| ICU | IR154 | — | — | — | — | — | — | — | IR |
| ICU | IR155 | — | — | — | — | — | — | — | IR |
| ICU | IR156 | — | — | — | — | — | — | — | IR |
| ICU | IR157 | — | — | — | — | — | — | — | IR |
| ICU | IR158 | — | — | — | — | — | — | — | IR |
| ICU | IR159 | — | — | — | — | — | — | — | IR |
| ICU | IR160 | — | — | — | — | — | — | — | IR |
| ICU | IR161 | — | — | — | — | — | — | — | IR |
| ICU | IR162 | — | — | — | — | — | — | — | IR |
| ICU | IR165 | — | — | — | — | — | — | — | IR |
| ICU | IR166 | — | — | — | — | — | — | — | IR |
| ICU | IR167 | — | — | — | — | — | — | — | IR |
| ICU | IR168 | — | — | — | — | — | — | — | IR |
| ICU | IR170 | — | — | — | — | — | — | — | IR |
| ICU | IR171 | — | — | — | — | — | — | — | IR |
| ICU | IR174 | — | — | — | — | — | — | — | IR |
| ICU | IR175 | — | — | — | — | — | — | — | IR |
| ICU | IR176 | — | — | — | — | — | — | — | IR |
| ICU | IR177 | — | — | — | — | — | — | — | IR |

表 5.2 I/O 寄存器位一览表 (9 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| ICU | IR178 | — | — | — | — | — | — | — | IR |
| ICU | IR179 | — | — | — | — | — | — | — | IR |
| ICU | IR180 | — | — | — | — | — | — | — | IR |
| ICU | IR181 | — | — | — | — | — | — | — | IR |
| ICU | IR182 | — | — | — | — | — | — | — | IR |
| ICU | IR183 | — | — | — | — | — | — | — | IR |
| ICU | IR184 | — | — | — | — | — | — | — | IR |
| ICU | IR185 | — | — | — | — | — | — | — | IR |
| ICU | IR198 | — | — | — | — | — | — | — | IR |
| ICU | IR199 | — | — | — | — | — | — | — | IR |
| ICU | IR200 | — | — | — | — | — | — | — | IR |
| ICU | IR201 | — | — | — | — | — | — | — | IR |
| ICU | IR214 | — | — | — | — | — | — | — | IR |
| ICU | IR215 | — | — | — | — | — | — | — | IR |
| ICU | IR216 | — | — | — | — | — | — | — | IR |
| ICU | IR217 | — | — | — | — | — | — | — | IR |
| ICU | IR218 | — | — | — | — | — | — | — | IR |
| ICU | IR219 | — | — | — | — | — | — | — | IR |
| ICU | IR220 | — | — | — | — | — | — | — | IR |
| ICU | IR221 | — | — | — | — | — | — | — | IR |
| ICU | IR222 | — | — | — | — | — | — | — | IR |
| ICU | IR223 | — | — | — | — | — | — | — | IR |
| ICU | IR224 | — | — | — | — | — | — | — | IR |
| ICU | IR225 | — | — | — | — | — | — | — | IR |
| ICU | IR226 | — | — | — | — | — | — | — | IR |
| ICU | IR227 | — | — | — | — | — | — | — | IR |
| ICU | IR228 | — | — | — | — | — | — | — | IR |
| ICU | IR229 | — | — | — | — | — | — | — | IR |
| ICU | IR230 | — | — | — | — | — | — | — | IR |
| ICU | IR231 | — | — | — | — | — | — | — | IR |
| ICU | IR232 | — | — | — | — | — | — | — | IR |
| ICU | IR233 | — | — | — | — | — | — | — | IR |
| ICU | IR234 | — | — | — | — | — | — | — | IR |
| ICU | IR235 | — | — | — | — | — | — | — | IR |
| ICU | IR236 | — | — | — | — | — | — | — | IR |
| ICU | IR237 | — | — | — | — | — | — | — | IR |
| ICU | IR238 | — | — | — | — | — | — | — | IR |
| ICU | IR239 | — | — | — | — | — | — | — | IR |
| ICU | IR240 | — | — | — | — | — | — | — | IR |
| ICU | IR241 | — | — | — | — | — | — | — | IR |
| ICU | IR246 | — | — | — | — | — | — | — | IR |
| ICU | IR247 | — | — | — | — | — | — | — | IR |
| ICU | IR248 | — | — | — | — | — | — | — | IR |
| ICU | IR249 | — | — | — | — | — | — | — | IR |
| ICU | IR250 | — | — | — | — | — | — | — | IR |
| ICU | IR251 | — | — | — | — | — | — | — | IR |
| ICU | IR252 | — | — | — | — | — | — | — | IR |
| ICU | IR253 | — | — | — | — | — | — | — | IR |

表 5.2 I/O 寄存器位一览表 (10 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| ICU | ISELR028 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR029 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR030 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR031 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR064 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR065 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR066 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR067 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR068 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR069 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR070 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR071 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR072 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR073 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR074 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR075 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR076 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR077 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR078 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR079 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR098 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR099 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR100 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR101 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR104 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR105 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR106 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR107 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR111 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR112 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR117 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR118 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR122 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR123 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR124 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR125 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR127 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR128 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR133 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR134 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR138 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR139 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR140 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR141 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR145 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR146 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR151 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR152 | — | — | — | — | — | — | ISEL[1:0] | |

表 5.2 I/O 寄存器位一览表 (11 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| ICU | ISELR156 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR157 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR158 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR159 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR161 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR162 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR167 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR168 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR174 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR175 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR177 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR178 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR180 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR181 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR183 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR184 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR198 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR199 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR200 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR201 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR215 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR216 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR219 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR220 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR223 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR224 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR227 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR228 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR231 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR232 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR235 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR236 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR239 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR240 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR247 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR248 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR251 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | ISELR252 | — | — | — | — | — | — | ISEL[1:0] | |
| ICU | IER02 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER03 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER08 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER09 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER0C | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER0D | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER0E | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER0F | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER10 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER11 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |

表 5.2 I/O 寄存器位一览表 (12 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| ICU | IER12 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER13 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER14 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER15 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER16 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER17 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER18 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER19 | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER1A | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER1B | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER1C | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER1D | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER1E | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IER1F | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| ICU | IPR00 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR01 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR02 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR04 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR05 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR06 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR07 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR20 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR21 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR22 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR23 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR24 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR25 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR26 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR27 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR28 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR29 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR2A | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR2B | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR2C | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR2D | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR2E | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR2F | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR40 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR44 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR45 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR46 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR47 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR4C | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR4D | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR4E | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR4F | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR50 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR51 | — | — | — | — | — | IPR[2:0] | | |

表 5.2 I/O 寄存器位一览表 (13 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| ICU | IPR52 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR53 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR54 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR55 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR56 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR57 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR58 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR59 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR5A | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR5B | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR5C | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR5D | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR5E | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR5F | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR60 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR61 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR62 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR63 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR68 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR69 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR6A | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR6B | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR70 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR71 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR72 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR73 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR80 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR81 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR82 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR83 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR84 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR85 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR86 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR89 | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR8A | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR8B | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR8C | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR8D | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR8E | — | — | — | — | — | IPR[2:0] | | |
| ICU | IPR8F | — | — | — | — | — | IPR[2:0] | | |
| ICU | FIR | FIEN | — | — | — | — | — | — | — |
| | | FVCT[7:0] | | | | | | | |
| DTC | DTCCR | — | — | — | RRS | RCHNE | — | — | ERR |
| DTC | DTCVBR | | | | | | | | |
| | | | | | | 0 | 0 | 0 | 0 |
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| DTC | DTCADMOD | — | — | — | — | — | — | — | SHORT |

表 5.2 I/O 寄存器位一览表 (14 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|---------------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| DTC | DTCST | — | — | — | — | — | — | — | DTCST |
| CMT (单元 0) | CMSTR0 | — | — | — | — | — | — | — | — |
| | | — | — | — | — | — | — | STR1 | STR0 |
| CMT0 | CMCR | — | — | — | — | — | — | — | — |
| | | — | CMIE | — | — | — | — | CKS[1:0] | |
| CMT0 | CMCNT | | | | | | | | |
| CMT0 | CMCOR | | | | | | | | |
| CMT1 | CMCR | — | — | — | — | — | — | — | — |
| | | — | CMIE | — | — | — | — | CKS[1:0] | |
| CMT1 | CMCNT | | | | | | | | |
| CMT1 | CMCOR | | | | | | | | |
| CMT (单元 1) | CMSTR1 | — | — | — | — | — | — | — | — |
| | | — | — | — | — | — | — | STR3 | STR2 |
| CMT2 | CMCR | — | — | — | — | — | — | — | — |
| | | — | CMIE | — | — | — | — | CKS[1:0] | |
| CMT2 | CMCNT | | | | | | | | |
| CMT2 | CMCOR | | | | | | | | |
| CMT3 | CMCR | — | — | — | — | — | — | — | — |
| | | — | CMIE | — | — | — | — | CKS[1:0] | |
| CMT3 | CMCNT | | | | | | | | |
| CMT3 | CMCOR | | | | | | | | |
| WDT | TCSR | — | TMS | TME | — | — | CKS[2:0] | | |
| WDT | WINA | | | | | | | | |
| WDT | TCNT | | | | | | | | |
| WDT | WINB | | | | | | | | |
| WDT | RSTCSR | WOVF | RSTE | — | — | — | — | — | — |
| AD0 | ADDRA | | | | | | | | |
| AD0 | ADDRB | | | | | | | | |
| AD0 | ADDRC | | | | | | | | |
| AD0 | ADDRD | | | | | | | | |
| AD0 | ADCSR | — | ADIE | ADST | — | CH[3:0] | | | |
| AD0 | ADCR | TRGS[2:0] | | | — | CKS[1:0] | | MODE[1:0] | |
| AD0 | ADDPR | DPSEL | — | — | — | — | — | — | — |
| AD0 | ADSSTR | | | | | | | | |

表 5.2 I/O 寄存器位一览表 (15 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|---------------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| AD1 | ADDRA | | | | | | | | |
| AD1 | ADDRB | | | | | | | | |
| AD1 | ADDRC | | | | | | | | |
| AD1 | ADDRD | | | | | | | | |
| AD1 | ADCSR | — | ADIE | ADST | — | CH[3:0] | | | |
| AD1 | ADCR | TRGS[2:0] | | | — | CKS[1:0] | | MODE[1:0] | |
| AD1 | ADDPR | DPSEL | — | — | — | — | — | — | — |
| AD1 | ADSSTR | | | | | | | | |
| AD2 | ADDRA | | | | | | | | |
| AD2 | ADDRB | | | | | | | | |
| AD2 | ADDRC | | | | | | | | |
| AD2 | ADDRD | | | | | | | | |
| AD2 | ADCSR | — | ADIE | ADST | — | CH[3:0] | | | |
| AD2 | ADCR | TRGS[2:0] | | | — | CKS[1:0] | | MODE[1:0] | |
| AD2 | ADDPR | DPSEL | — | — | — | — | — | — | — |
| AD2 | ADSSTR | | | | | | | | |
| AD3 | ADDRA | | | | | | | | |
| AD3 | ADDRB | | | | | | | | |
| AD3 | ADDRC | | | | | | | | |
| AD3 | ADDRD | | | | | | | | |
| AD3 | ADCSR | — | ADIE | ADST | — | CH[3:0] | | | |
| AD3 | ADCR | TRGS[2:0] | | | — | CKS[1:0] | | MODE[1:0] | |
| AD3 | ADDPR | DPSEL | — | — | — | — | — | — | — |
| AD3 | ADSSTR | | | | | | | | |
| D/A | DADR0 | | | | | | | | |
| D/A | DADR1 | | | | | | | | |
| D/A | DACR | DAOE1 | DAOE0 | DAE | — | — | — | — | — |
| D/A | DADPR | DPSEL | — | — | — | — | — | — | — |
| TPU (单元 0) | TSTRA | — | — | CST5 | CST4 | CST3 | CST2 | CST1 | CST0 |
| TPU (单元 0) | TSYRA | — | — | SYNC5 | SYNC4 | SYNC3 | SYNC2 | SYNC1 | SYNC0 |
| TPU0 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | |
| TPU0 | TMDR | ICSELD | ICSELB | BFB | BFA | MD[3:0] | | | |
| TPU0 | TIORH | IOB[3:0] | | | | IOA[3:0] | | | |

表 5.2 I/O 寄存器位一览表 (16 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 | |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|--|
| TPU0 | TIORL | IOD[3:0] | | | | IOC[3:0] | | | | |
| TPU0 | TIER | TTGE | — | — | TCIEV | TGIED | TGIEC | TGIEB | TGIEA | |
| TPU0 | TSR | — | — | — | — | — | — | — | — | |
| TPU0 | TCNT | | | | | | | | | |
| TPU0 | TGRA | | | | | | | | | |
| TPU0 | TGRB | | | | | | | | | |
| TPU0 | TGRC | | | | | | | | | |
| TPU0 | TGRD | | | | | | | | | |
| TPU1 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | | |
| TPU1 | TMDR | — | ICSELB | — | — | MD[3:0] | | | | |
| TPU1 | TIOR | IOB[3:0] | | | | IOA[3:0] | | | | |
| TPU1 | TIER | TTGE | — | TCIEU | TCIEV | — | — | TGIEB | TGIEA | |
| TPU1 | TSR | TCFD | — | — | — | — | — | — | — | |
| TPU1 | TCNT | | | | | | | | | |
| TPU1 | TGRA | | | | | | | | | |
| TPU1 | TGRB | | | | | | | | | |
| TPU2 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | | |
| TPU2 | TMDR | — | ICSELB | — | — | MD[3:0] | | | | |
| TPU2 | TIOR | IOB[3:0] | | | | IOA[3:0] | | | | |
| TPU2 | TIER | TTGE | — | TCIEU | TCIEV | — | — | TGIEB | TGIEA | |
| TPU2 | TSR | TCFD | — | — | — | — | — | — | — | |
| TPU2 | TCNT | | | | | | | | | |
| TPU2 | TGRA | | | | | | | | | |
| TPU2 | TGRB | | | | | | | | | |
| TPU3 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | | |
| TPU3 | TMDR | ICSELD | ICSELB | BFB | BFA | MD[3:0] | | | | |
| TPU3 | TIORH | IOB[3:0] | | | | IOA[3:0] | | | | |
| TPU3 | TIORL | IOD[3:0] | | | | IOC[3:0] | | | | |
| TPU3 | TIER | TTGE | — | — | TCIEV | TGIED | TGIEC | TGIEB | TGIEA | |
| TPU3 | TSR | — | — | — | — | — | — | — | — | |
| TPU3 | TCNT | | | | | | | | | |
| TPU3 | TGRA | | | | | | | | | |
| TPU3 | TGRB | | | | | | | | | |

表 5.2 I/O 寄存器位一览表 (17 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|---------------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| TPU3 | TGRC | | | | | | | | |
| TPU3 | TGRD | | | | | | | | |
| TPU4 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | |
| TPU4 | TMDR | — | ICSELB | — | — | MD[3:0] | | | |
| TPU4 | TIOR | IOB[3:0] | | | | IOA[3:0] | | | |
| TPU4 | TIER | TTGE | — | TCIEU | TCIEV | — | — | TGIEB | TGIEA |
| TPU4 | TSR | TCFD | — | — | — | — | — | — | — |
| TPU4 | TCNT | | | | | | | | |
| TPU4 | TGRA | | | | | | | | |
| TPU4 | TGRB | | | | | | | | |
| TPU5 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | |
| TPU5 | TMDR | — | ICSELB | — | — | MD[3:0] | | | |
| TPU5 | TIOR | IOB[3:0] | | | | IOA[3:0] | | | |
| TPU5 | TIER | TTGE | — | TCIEU | TCIEV | — | — | TGIEB | TGIEA |
| TPU5 | TSR | TCFD | — | — | — | — | — | — | — |
| TPU5 | TCNT | | | | | | | | |
| TPU5 | TGRA | | | | | | | | |
| TPU5 | TGRB | | | | | | | | |
| TPU (单元 1) | TSTRB | — | — | CST5 | CST4 | CST3 | CST2 | CST1 | CST0 |
| TPU (单元 1) | TSYRB | — | — | SYNC5 | SYNC4 | SYNC3 | SYNC2 | SYNC1 | SYNC0 |
| TPU6 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | |
| TPU6 | TMDR | ICSELD | ICSELB | BFB | BFA | MD[3:0] | | | |
| TPU6 | TIORH | IOB[3:0] | | | | IOA[3:0] | | | |
| TPU6 | TIORL | IOD[3:0] | | | | IOC[3:0] | | | |
| TPU6 | TIER | TTGE | — | — | TCIEV | TGIED | TGIEC | TGIEB | TGIEA |
| TPU6 | TSR | — | — | — | — | — | — | — | — |
| TPU6 | TCNT | | | | | | | | |
| TPU6 | TGRA | | | | | | | | |
| TPU6 | TGRB | | | | | | | | |
| TPU6 | TGRC | | | | | | | | |
| TPU6 | TGRD | | | | | | | | |
| TPU7 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | |
| TPU7 | TMDR | — | ICSELB | — | — | MD[3:0] | | | |
| TPU7 | TIOR | IOB[3:0] | | | | IOA[3:0] | | | |

表 5.2 I/O 寄存器位一览表 (18 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| TPU7 | TIER | TTGE | — | TCIEU | TCIEV | — | — | TGIEB | TGIEA |
| TPU7 | TSR | TCFD | — | — | — | — | — | — | — |
| TPU7 | TCNT | | | | | | | | |
| TPU7 | TGRA | | | | | | | | |
| TPU7 | TGRB | | | | | | | | |
| TPU8 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | |
| TPU8 | TMDR | — | ICSELB | — | — | MD[3:0] | | | |
| TPU8 | TIOR | IOB[3:0] | | | | IOA[3:0] | | | |
| TPU8 | TIER | TTGE | — | TCIEU | TCIEV | — | — | TGIEB | TGIEA |
| TPU8 | TSR | TCFD | — | — | — | — | — | — | — |
| TPU8 | TCNT | | | | | | | | |
| TPU8 | TGRA | | | | | | | | |
| TPU8 | TGRB | | | | | | | | |
| TPU9 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | |
| TPU9 | TMDR | ICSELD | ICSELB | BFB | BFA | MD[3:0] | | | |
| TPU9 | TIORH | IOB[3:0] | | | | IOA[3:0] | | | |
| TPU9 | TIORL | IOD[3:0] | | | | IOC[3:0] | | | |
| TPU9 | TIER | TTGE | — | — | TCIEV | TGIED | TGIEC | TGIEB | TGIEA |
| TPU9 | TSR | — | — | — | — | — | — | — | — |
| TPU9 | TCNT | | | | | | | | |
| TPU9 | TGRA | | | | | | | | |
| TPU9 | TGRB | | | | | | | | |
| TPU9 | TGRC | | | | | | | | |
| TPU9 | TGRD | | | | | | | | |
| TPU10 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | |
| TPU10 | TMDR | — | ICSELB | — | — | MD[3:0] | | | |
| TPU10 | TIOR | IOB[3:0] | | | | IOA[3:0] | | | |
| TPU10 | TIER | TTGE | — | TCIEU | TCIEV | — | — | TGIEB | TGIEA |
| TPU10 | TSR | TCFD | — | — | — | — | — | — | — |
| TPU10 | TCNT | | | | | | | | |
| TPU10 | TGRA | | | | | | | | |
| TPU10 | TGRB | | | | | | | | |
| TPU11 | TCR | CCLR[2:0] | | | CKEG[1:0] | | TPSC[2:0] | | |
| TPU11 | TMDR | — | ICSELB | — | — | MD[3:0] | | | |

表 5.2 I/O 寄存器位一览表 (19 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 | |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|--|
| TPU11 | TIOR | IOB[3:0] | | | | IOA[3:0] | | | | |
| TPU11 | TIER | TTGE | — | TCIEU | TCIEV | — | — | TGIEB | TGIEA | |
| TPU11 | TSR | TCFD | — | — | — | — | — | — | — | |
| TPU11 | TCNT | | | | | | | | | |
| TPU11 | TGRA | | | | | | | | | |
| TPU11 | TGRB | | | | | | | | | |
| PPG0 | PCR | G3CMS[1:0] | | G2CMS[1:0] | | G1CMS[1:0] | | G0CMS[1:0] | | |
| PPG0 | PMR | G3INV | G2INV | G1INV | G0INV | G3NOV | G2NOV | G1NOV | G0NOV | |
| PPG0 | NDERH | NDER15 | NDER14 | NDER13 | NDER12 | NDER11 | NDER10 | NDER9 | NDER8 | |
| PPG0 | NDERL | NDER7 | NDER6 | NDER5 | NDER4 | NDER3 | NDER2 | NDER1 | NDER0 | |
| PPG0 | PODRH | POD15 | POD14 | POD13 | POD12 | POD11 | POD10 | POD9 | POD8 | |
| PPG0 | PODRL | POD7 | POD6 | POD5 | POD4 | POD3 | POD2 | POD1 | POD0 | |
| PPG0 | NDRH (注1) | NDR15 | NDR14 | NDR13 | NDR12 | NDR11 | NDR10 | NDR9 | NDR8 | |
| | | | | | | (—) | (—) | (—) | (—) | |
| PPG0 | NDRL (注2) | NDR7 | NDR6 | NDR5 | NDR4 | NDR3 | NDR2 | NDR1 | NDR0 | |
| | | | | | | (—) | (—) | (—) | (—) | |
| PPG0 | NDRH (注1) | — | — | — | — | NDR11 | NDR10 | NDR9 | NDR8 | |
| PPG0 | NDRL (注2) | — | — | — | — | NDR3 | NDR2 | NDR1 | NDR0 | |
| PPG1 | PTRSLR | — | — | — | — | — | — | — | PTRSL | |
| PPG1 | PCR | G3CMS[1:0] | | G2CMS[1:0] | | G1CMS[1:0] | | G0CMS[1:0] | | |
| PPG1 | PMR | G3INV | G2INV | G1INV | G0INV | G3NOV | G2NOV | G1NOV | G0NOV | |
| PPG1 | NDERH | NDER31 | NDER30 | NDER29 | NDER28 | NDER27 | NDER26 | NDER25 | NDER24 | |
| PPG1 | NDERL | NDER23 | NDER22 | NDER21 | NDER20 | NDER19 | NDER18 | NDER17 | NDER16 | |
| PPG1 | PODRH | POD31 | POD30 | POD29 | POD28 | POD27 | POD26 | POD25 | POD24 | |
| PPG1 | PODRL | POD23 | POD22 | POD21 | POD20 | POD19 | POD18 | POD17 | POD16 | |
| PPG1 | NDRH (注3) | NDR31 | NDR30 | NDR29 | NDR28 | NDR27 | NDR26 | NDR25 | NDR24 | |
| | | | | | | (—) | (—) | (—) | (—) | |
| PPG1 | NDRL (注4) | NDR23 | NDR22 | NDR21 | NDR20 | NDR19 | NDR18 | NDR17 | NDR16 | |
| | | | | | | (—) | (—) | (—) | (—) | |
| PPG1 | NDRH (注3) | — | — | — | — | NDR27 | NDR26 | NDR25 | NDR24 | |
| PPG1 | NDRL (注4) | — | — | — | — | NDR19 | NDR18 | NDR17 | NDR16 | |
| TMR0 | TCR | CMIEB | CMIEA | OVIE | CCLR[1:0] | | — | — | — | |
| TMR1 | TCR | CMIEB | CMIEA | OVIE | CCLR[1:0] | | — | — | — | |
| TMR0 | TCSR | — | — | — | ADTE | OSB[1:0] | | OSA[1:0] | | |
| TMR1 | TCSR | — | — | — | — | OSB[1:0] | | OSA[1:0] | | |
| TMR0 | TCORA | | | | | | | | | |
| TMR1 | TCORA | | | | | | | | | |
| TMR0 | TCORB | | | | | | | | | |
| TMR1 | TCORB | | | | | | | | | |
| TMR0 | TCNT | | | | | | | | | |
| TMR1 | TCNT | | | | | | | | | |
| TMR0 | TCCR | TMRIS | — | — | CSS[1:0] | | CKS[2:0] | | | |
| TMR1 | TCCR | TMRIS | — | — | CSS[1:0] | | CKS[2:0] | | | |
| TMR2 | TCR | CMIEB | CMIEA | OVIE | CCLR[1:0] | | — | — | — | |
| TMR3 | TCR | CMIEB | CMIEA | OVIE | CCLR[1:0] | | — | — | — | |

表 5.2 I/O 寄存器位一览表 (20 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|--------------------|-------------------|------------------------|------------------|
| TMR2 | TCSR | — | — | — | ADTE | OSB[1:0] | | OSA[1:0] | |
| TMR3 | TCSR | — | — | — | — | OSB[1:0] | | OSA[1:0] | |
| TMR2 | TCORA | | | | | | | | |
| TMR3 | TCORA | | | | | | | | |
| TMR2 | TCORB | | | | | | | | |
| TMR3 | TCORB | | | | | | | | |
| TMR2 | TCNT | | | | | | | | |
| TMR3 | TCNT | | | | | | | | |
| TMR2 | TCCR | TMRIS | — | — | CSS[1:0] | | CKS[2:0] | | |
| TMR3 | TCCR | TMRIS | — | — | CSS[1:0] | | CKS[2:0] | | |
| SCI0 | SMR (注5) | CM (GM) | CHR (BLK) | PE (PE) | PM (PM) | STOP (BCP[1:0]) | — | CKS[1:0] (CKS[1:0]) | |
| SCI0 | BRR | | | | | | | | |
| SCI0 | SCR (注5) | TIE | RIE | TE | RE | — | TEIE | CKE[1:0] | |
| SCI0 | TDR | | | | | | | | |
| SCI0 | SSR (注5) | — (—) | — (—) | ORER (ORER) | FER (ERS) | PER (PER) | TEND (TEND) | — (—) | — (—) |
| SCI0 | RDR | | | | | | | | |
| SCI0 | SCMR | BCP2 | — | — | — | SDIR | SINV | — | SMIF |
| SCI0 | SEMR | — | — | — | ABCS | — | — | — | ACS0 |
| SCI1 | SMR (注5) | CM (GM) | CHR (BLK) | PE (PE) | PM (PM) | STOP (BCP[1:0]) | — | CKS[1:0] (CKS[1:0]) | |
| SCI1 | BRR | | | | | | | | |
| SCI1 | SCR (注5) | TIE | RIE | TE | RE | — | TEIE | CKE[1:0] | |
| SCI1 | TDR | | | | | | | | |
| SCI1 | SSR (注5) | — (—) | — (—) | ORER (ORER) | FER (ERS) | PER (PER) | TEND (TEND) | — (—) | — (—) |
| SCI1 | RDR | | | | | | | | |
| SCI1 | SCMR | BCP2 | — | — | — | SDIR | SINV | — | SMIF |
| SCI1 | SEMR | — | — | — | ABCS | — | — | — | ACS0 |
| SCI2 | SMR (注5) | CM (GM) | CHR (BLK) | PE (PE) | PM (PM) | STOP (BCP[1:0]) | — | CKS[1:0] (CKS[1:0]) | |
| SCI2 | BRR | | | | | | | | |
| SCI2 | SCR (注5) | TIE | RIE | TE | RE | — | TEIE | CKE[1:0] | |
| SCI2 | TDR | | | | | | | | |
| SCI2 | SSR (注5) | — (—) | — (—) | ORER (ORER) | FER (ERS) | PER (PER) | TEND (TEND) | — (—) | — (—) |
| SCI2 | RDR | | | | | | | | |
| SCI2 | SCMR | BCP2 | — | — | — | SDIR | SINV | — | SMIF |
| SCI2 | SEMR | — | — | — | ABCS | — | — | — | ACS0 |
| SCI3 | SMR (注5) | CM (GM) | CHR (BLK) | PE (PE) | PM (PM) | STOP (BCP[1:0]) | — | CKS[1:0] (CKS[1:0]) | |
| SCI3 | BRR | | | | | | | | |
| SCI3 | SCR (注5) | TIE | RIE | TE | RE | — | TEIE | CKE[1:0] | |
| SCI3 | TDR | | | | | | | | |
| SCI3 | SSR (注5) | — (—) | — (—) | ORER (ORER) | FER (ERS) | PER (PER) | TEND (TEND) | — (—) | — (—) |
| SCI3 | RDR | | | | | | | | |
| SCI3 | SCMR | BCP2 | — | — | — | SDIR | SINV | — | SMIF |

表 5.2 I/O 寄存器位一览表 (21 / 26)

| 模块符号 | 寄存器符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 | |
|-------|----------|-------------------|-------------------|-------------------|-------------------|--------------------|-------------------|------------------------|------------------|--|
| SCI3 | SEMR | — | — | — | ABCS | — | — | — | ACS0 | |
| SCI4 | SMR (注5) | CM (GM) | CHR (BLK) | PE (PE) | PM (PM) | STOP (BCP[1:0]) | — | CKS[1:0] (CKS[1:0]) | | |
| SCI4 | BRR | | | | | | | | | |
| SCI4 | SCR (注5) | TIE | RIE | TE | RE | — | TEIE | CKE[1:0] | | |
| SCI4 | TDR | | | | | | | | | |
| SCI4 | SSR (注5) | — (—) | — (—) | ORER (ORER) | FER (ERS) | PER (PER) | TEND (TEND) | — (—) | — (—) | |
| SCI4 | RDR | | | | | | | | | |
| SCI4 | SCMR | BCP2 | — | — | — | SDIR | SINV | — | SMIF | |
| SCI4 | SEMR | — | — | — | ABCS | — | — | — | ACS0 | |
| SCI5 | SMR (注5) | CM (GM) | CHR (BLK) | PE (PE) | PM (PM) | STOP (BCP[1:0]) | — | CKS[1:0] (CKS[1:0]) | | |
| SCI5 | BRR | | | | | | | | | |
| SCI5 | SCR (注5) | TIE | RIE | TE | RE | — | TEIE | CKE[1:0] | | |
| SCI5 | TDR | | | | | | | | | |
| SCI5 | SSR (注5) | — (—) | — (—) | ORER (ORER) | FER (ERS) | PER (PER) | TEND (TEND) | — (—) | — (—) | |
| SCI5 | RDR | | | | | | | | | |
| SCI5 | SCMR | BCP2 | — | — | — | SDIR | SINV | — | SMIF | |
| SCI5 | SEMR | — | — | — | ABCS | — | — | — | ACS0 | |
| SCI6 | SMR (注5) | CM (GM) | CHR (BLK) | PE (PE) | PM (PM) | STOP (BCP[1:0]) | — | CKS[1:0] (CKS[1:0]) | | |
| SCI6 | BRR | | | | | | | | | |
| SCI6 | SCR (注5) | TIE | RIE | TE | RE | — | TEIE | CKE[1:0] | | |
| SCI6 | TDR | | | | | | | | | |
| SCI6 | SSR (注5) | — (—) | — (—) | ORER (ORER) | FER (ERS) | PER (PER) | TEND (TEND) | — (—) | — (—) | |
| SCI6 | RDR | | | | | | | | | |
| SCI6 | SCMR | BCP2 | — | — | — | SDIR | SINV | — | SMIF | |
| SCI6 | SEMR | — | — | — | ABCS | — | — | — | ACS0 | |
| CRC | CRCCR | DORCLR | — | — | — | — | LMS | GPS[1:0] | | |
| CRC | CRCDIR | | | | | | | | | |
| CRC | CRCDOR | | | | | | | | | |
| RIIC0 | ICCR1 | ICE | IICRST | CLO | SOWP | SCLO | SDAO | SCLI | SDAI | |
| RIIC0 | ICCR2 | BBSY | MST | TRS | — | SP | RS | ST | — | |
| RIIC0 | ICMR1 | MTWP | CKS[2:0] | | | BCWP | BC[2:0] | | | |
| RIIC0 | ICMR2 | DLCS | SDDL[2:0] | | | — | TMOH | TMOL | TMOS | |
| RIIC0 | ICMR3 | SMBS | WAIT | RDRFS | ACKWP | ACKBT | ACKBR | NF[1:0] | | |
| RIIC0 | ICFER | FMPE | SCLE | NFE | NACKE | SALE | NALE | MALE | TMOE | |
| RIIC0 | ICSER | HOAE | — | DIDE | — | GCAE | SAR2E | SAR1E | SAR0E | |
| RIIC0 | ICIER | TIE | TEIE | RIE | NAKIE | SPIE | STIE | ALIE | TMOIE | |
| RIIC0 | ICSR1 | HOA | — | DID | — | GCA | AAS2 | AAS1 | AAS0 | |
| RIIC0 | ICSR2 | TDRE | TEND | RDRF | NACKF | STOP | START | AL | TMOF | |
| RIIC0 | SARL0 | SVA[7:1] | | | | | | | SVA0 | |
| RIIC0 | SARU0 | — | — | — | — | — | SVA[9:8] | | FS | |
| RIIC0 | SARL1 | SVA[7:1] | | | | | | | SVA0 | |

表 5.2 I/O 寄存器位一览表 (22 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| RIIC0 | SARU1 | — | — | — | — | — | SVA[9:8] | | FS |
| RIIC0 | SARL2 | SVA[7:1] | | | | | | | SVA0 |
| RIIC0 | SARU2 | — | — | — | — | — | SVA[9:8] | | FS |
| RIIC0 | ICBRL | — | — | — | BRL[4:0] | | | | |
| RIIC0 | ICBRH | — | — | — | BRH[4:0] | | | | |
| RIIC0 | ICDRT | | | | | | | | |
| RIIC0 | ICDRR | | | | | | | | |
| RIIC1 | ICCR1 | ICE | IICRST | CLO | SOWP | SCLO | SDAO | SCLI | SDAI |
| RIIC1 | ICCR2 | BBSY | MST | TRS | — | SP | RS | ST | — |
| RIIC1 | ICMR1 | MTWP | CKS[2:0] | | | BCWP | BC[2:0] | | |
| RIIC1 | ICMR2 | DLCS | SDDL[2:0] | | | — | TMOH | TMOL | TMOS |
| RIIC1 | ICMR3 | SMBS | WAIT | RDRFS | ACKWP | ACKBT | ACKBR | NF[1:0] | |
| RIIC1 | ICFER | FMPE | SCLE | NFE | NACKE | SALE | NALE | MALE | TMOE |
| RIIC1 | ICSER | HOAE | — | DIDE | — | GCAE | SAR2E | SAR1E | SAR0E |
| RIIC1 | ICIER | TIE | TEIE | RIE | NAKIE | SPIE | STIE | ALIE | TMOIE |
| RIIC1 | ICSR1 | HOA | — | DID | — | GCA | AAS2 | AAS1 | AAS0 |
| RIIC1 | ICSR2 | TDRE | TEND | RDRF | NACKF | STOP | START | AL | TMOF |
| RIIC1 | SARL0 | SVA[7:1] | | | | | | | SVA0 |
| RIIC1 | SARU0 | — | — | — | — | — | SVA[9:8] | | FS |
| RIIC1 | SARL1 | SVA[7:1] | | | | | | | SVA0 |
| RIIC1 | SARU1 | — | — | — | — | — | SVA[9:8] | | FS |
| RIIC1 | SARL2 | SVA[7:1] | | | | | | | SVA0 |
| RIIC1 | SARU2 | — | — | — | — | — | SVA[9:8] | | FS |
| RIIC1 | ICBRL | — | — | — | BRL[4:0] | | | | |
| RIIC1 | ICBRH | — | — | — | BRH[4:0] | | | | |
| RIIC1 | ICDRT | | | | | | | | |
| RIIC1 | ICDRR | | | | | | | | |
| P0 | DDR | — | — | B5 | B4 | B3 | B2 | B1 | B0 |
| P1 | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P2 | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P3 | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P4 | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P5 | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P6 | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P7 | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P8 | DDR | — | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P9 | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PA | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PB | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PC | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PD | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PE | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PF | DDR | — | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PG | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PH | DDR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P0 | DR | — | — | B5 | B4 | B3 | B2 | B1 | B0 |
| P1 | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P2 | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

表 5.2 I/O 寄存器位一览表 (23 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| P3 | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P4 | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P5 | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P6 | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P7 | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P8 | DR | — | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P9 | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PA | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PB | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PC | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PD | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PE | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PF | DR | — | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PG | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PH | DR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P0 | PORT | — | — | B5 | B4 | B3 | B2 | B1 | B0 |
| P1 | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P2 | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P3 | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P4 | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P5 | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P6 | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P7 | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P8 | PORT | — | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P9 | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PA | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PB | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PC | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PD | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PE | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PF | PORT | — | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PG | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PH | PORT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P0 | ICR | — | — | B5 | B4 | B3 | B2 | B1 | B0 |
| P1 | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P2 | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P3 | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P4 | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P5 | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P6 | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P7 | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P8 | ICR | — | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| P9 | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PA | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PB | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PC | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PD | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| PE | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

表 5.2 I/O 寄存器位一览表 (24 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 | | |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|---|---|
| PF | ICR | — | B6 | B5 | B4 | B3 | B2 | B1 | B0 | | |
| PG | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | | |
| PH | ICR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | | |
| P2 | ODR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | | |
| PC | ODR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | | |
| PA | PCR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | | |
| PB | PCR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | | |
| PC | PCR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | | |
| PD | PCR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | | |
| PE | PCR | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | | |
| I/O PORT | PFCR0 | CS7E | CS6E | CS5E | CS4E | CS3E | CS2E | CS1E | CS0E | | |
| I/O PORT | PFCR1 | CS7S[1:0] | | CS6S[1:0] | | CS5S[1:0] | | CS4S[1:0] | | | |
| I/O PORT | PFCR2 | CS3S | CS2S | — | — | — | — | — | — | | |
| I/O PORT | PFCR3 | A23E | A22E | A21E | A20E | A19E | A18E | A17E | A16E | | |
| I/O PORT | PFCR4 | A15E | A14E | A13E | A12E | A11E | A10E | A9E | A8E | | |
| I/O PORT | PFCR5 | — | WR1BC1E | — | DHE | TCLKS | — | — | — | | |
| I/O PORT | PFCR6 | TPUMS5 | TPUMS4 | TPUMS3A | TPUMS3B | TPUMS2 | TPUMS1 | TPUMS0A | TPUMS0B | | |
| I/O PORT | PFCR7 | TPUMS11 | TPUMS10 | TPUMS9A | TPUMS9B | TPUMS8 | TPUMS7 | TPUMS6A | TPUMS6B | | |
| I/O PORT | PFCR8 | ITS15 | ITS14 | ITS13 | ITS12 | ITS11 | ITS10 | ITS9 | ITS8 | | |
| I/O PORT | PFCR9 | ITS7 | ITS6 | ITS5 | ITS4 | ITS3 | ITS2 | ITS1 | ITS0 | | |
| SYSTEM | DPSBYCR | DPSBY | IOKEEP | RAMCUT2 | RAMCUT1 | — | — | — | RAMCUT0 | | |
| SYSTEM | DPSWCR | — | — | WTSTS[5:0] | | | | | | — | — |
| SYSTEM | DPSIER | DNMIE | — | — | — | DIRQ3E | DIRQ2E | DIRQ1E | DIRQ0E | | |
| SYSTEM | DPSIFR | DNMIF | — | — | — | DIRQ3F | DIRQ2F | DIRQ1F | DIRQ0F | | |
| SYSTEM | DPSIEGR | DNMIEG | — | — | — | DIRQ3EG | DIRQ2EG | DIRQ1EG | DIRQ0EG | | |
| SYSTEM | RSTSR | DPSRSTF | — | — | — | — | — | — | — | | |
| FLASH | FWEPROR | — | — | — | — | — | — | FLWE[1:0] | | | |
| SYSTEM | DPSBKR0 | BKUP07 | BKUP06 | BKUP05 | BKUP04 | BKUP03 | BKUP02 | BKUP01 | BKUP00 | | |
| SYSTEM | DPSBKR1 | BKUP17 | BKUP16 | BKUP15 | BKUP14 | BKUP13 | BKUP12 | BKUP11 | BKUP10 | | |
| SYSTEM | DPSBKR2 | BKUP27 | BKUP26 | BKUP25 | BKUP24 | BKUP23 | BKUP22 | BKUP21 | BKUP20 | | |
| SYSTEM | DPSBKR3 | BKUP37 | BKUP36 | BKUP35 | BKUP34 | BKUP33 | BKUP32 | BKUP31 | BKUP30 | | |
| SYSTEM | DPSBKR4 | BKUP47 | BKUP46 | BKUP45 | BKUP44 | BKUP43 | BKUP42 | BKUP41 | BKUP40 | | |
| SYSTEM | DPSBKR5 | BKUP57 | BKUP56 | BKUP55 | BKUP54 | BKUP53 | BKUP52 | BKUP51 | BKUP50 | | |
| SYSTEM | DPSBKR6 | BKUP67 | BKUP66 | BKUP65 | BKUP64 | BKUP63 | BKUP62 | BKUP61 | BKUP60 | | |
| SYSTEM | DPSBKR7 | BKUP77 | BKUP76 | BKUP75 | BKUP74 | BKUP73 | BKUP72 | BKUP71 | BKUP70 | | |
| SYSTEM | DPSBKR8 | BKUP87 | BKUP86 | BKUP85 | BKUP84 | BKUP83 | BKUP82 | BKUP81 | BKUP80 | | |
| SYSTEM | DPSBKR9 | BKUP97 | BKUP96 | BKUP95 | BKUP94 | BKUP93 | BKUP92 | BKUP91 | BKUP90 | | |
| SYSTEM | DPSBKR10 | BKUP107 | BKUP106 | BKUP105 | BKUP104 | BKUP103 | BKUP102 | BKUP101 | BKUP100 | | |
| SYSTEM | DPSBKR11 | BKUP117 | BKUP116 | BKUP115 | BKUP114 | BKUP113 | BKUP112 | BKUP111 | BKUP110 | | |
| SYSTEM | DPSBKR12 | BKUP127 | BKUP126 | BKUP125 | BKUP124 | BKUP123 | BKUP122 | BKUP121 | BKUP120 | | |
| SYSTEM | DPSBKR13 | BKUP137 | BKUP136 | BKUP135 | BKUP134 | BKUP133 | BKUP132 | BKUP131 | BKUP130 | | |
| SYSTEM | DPSBKR14 | BKUP147 | BKUP146 | BKUP145 | BKUP144 | BKUP143 | BKUP142 | BKUP141 | BKUP140 | | |
| SYSTEM | DPSBKR15 | BKUP157 | BKUP156 | BKUP155 | BKUP154 | BKUP153 | BKUP152 | BKUP151 | BKUP150 | | |
| SYSTEM | DPSBKR16 | BKUP167 | BKUP166 | BKUP165 | BKUP164 | BKUP163 | BKUP162 | BKUP161 | BKUP160 | | |
| SYSTEM | DPSBKR17 | BKUP177 | BKUP176 | BKUP175 | BKUP174 | BKUP173 | BKUP172 | BKUP171 | BKUP170 | | |
| SYSTEM | DPSBKR18 | BKUP187 | BKUP186 | BKUP185 | BKUP184 | BKUP183 | BKUP182 | BKUP181 | BKUP180 | | |
| SYSTEM | DPSBKR19 | BKUP197 | BKUP196 | BKUP195 | BKUP194 | BKUP193 | BKUP192 | BKUP191 | BKUP190 | | |
| SYSTEM | DPSBKR20 | BKUP207 | BKUP206 | BKUP205 | BKUP204 | BKUP203 | BKUP202 | BKUP201 | BKUP200 | | |

表 5.2 I/O 寄存器位一览表 (25 / 26)

| 模块 符号 | 寄存器 符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|----------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| SYSTEM | DPSBKR21 | BKUP217 | BKUP216 | BKUP215 | BKUP214 | BKUP213 | BKUP212 | BKUP211 | BKUP210 |
| SYSTEM | DPSBKR22 | BKUP227 | BKUP226 | BKUP225 | BKUP224 | BKUP223 | BKUP222 | BKUP221 | BKUP220 |
| SYSTEM | DPSBKR23 | BKUP237 | BKUP236 | BKUP235 | BKUP234 | BKUP233 | BKUP232 | BKUP231 | BKUP230 |
| SYSTEM | DPSBKR24 | BKUP247 | BKUP246 | BKUP245 | BKUP244 | BKUP243 | BKUP242 | BKUP241 | BKUP240 |
| SYSTEM | DPSBKR25 | BKUP257 | BKUP256 | BKUP255 | BKUP254 | BKUP253 | BKUP252 | BKUP251 | BKUP250 |
| SYSTEM | DPSBKR26 | BKUP267 | BKUP266 | BKUP265 | BKUP264 | BKUP263 | BKUP262 | BKUP261 | BKUP260 |
| SYSTEM | DPSBKR27 | BKUP277 | BKUP276 | BKUP275 | BKUP274 | BKUP273 | BKUP272 | BKUP271 | BKUP270 |
| SYSTEM | DPSBKR28 | BKUP287 | BKUP286 | BKUP285 | BKUP284 | BKUP283 | BKUP282 | BKUP281 | BKUP280 |
| SYSTEM | DPSBKR29 | BKUP297 | BKUP296 | BKUP295 | BKUP294 | BKUP293 | BKUP292 | BKUP291 | BKUP290 |
| SYSTEM | DPSBKR30 | BKUP307 | BKUP306 | BKUP305 | BKUP304 | BKUP303 | BKUP302 | BKUP301 | BKUP300 |
| SYSTEM | DPSBKR31 | BKUP317 | BKUP316 | BKUP315 | BKUP314 | BKUP313 | BKUP312 | BKUP311 | BKUP310 |
| ICU | IRQER0 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER1 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER2 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER3 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER4 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER5 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER6 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER7 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER8 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER9 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER10 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER11 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER12 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER13 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER14 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQER15 | — | — | — | — | — | — | — | IRQEN |
| ICU | IRQCR0 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR1 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR2 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR3 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR4 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR5 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR6 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR7 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR8 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR9 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR10 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR11 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR12 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR13 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR14 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | IRQCR15 | — | — | — | — | IRQMD[1:0] | | — | — |
| ICU | SSIER | SSI15 | SSI14 | SSI13 | SSI12 | SSI11 | SSI10 | SSI9 | SSI8 |
| | | SSI7 | SSI6 | SSI5 | SSI4 | SSI3 | SSI2 | SSI1 | SSI0 |
| ICU | NMIER | — | — | — | — | — | — | — | NMIEN |
| ICU | NMICR | — | — | — | — | NMIMD | — | — | — |
| ICU | NMISR | — | — | — | — | — | — | — | NMIST |

表 5.2 I/O 寄存器位一览表 (26 / 26)

| 模块符号 | 寄存器符号 | bit 31/23/15/7 | bit 30/22/14/6 | bit 29/21/13/5 | bit 28/20/12/4 | bit 27/19/11/3 | bit 26/18/10/2 | bit 25/17/9/1 | bit 24/16/8/0 |
|-------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|------------------|------------------|
| ICU | NMICLR | — | — | — | — | — | — | — | NMICLR |
| FLASH | FMODR | — | — | — | FRDMD | — | — | — | — |
| FLASH | FASTAT | ROMAE | — | — | CMDLK | DFLAE | — | DFLRPE | DFLWPE |
| FLASH | FAEINT | ROMAEIE | — | — | CMDLKIE | DFLAEIE | — | DFLRPEIE | DFLWPEIE |
| FLASH | FRDYIE | — | — | — | — | — | — | — | FRDYIE |
| FLASH | DFLRE | KEY[7:0] | | | | | | | |
| | | — | — | — | — | DBRE3 | DBRE2 | DBRE1 | DBRE0 |
| FLASH | DFLWE | KEY[7:0] | | | | | | | |
| | | — | — | — | — | DBWE3 | DBWE2 | DBWE1 | DBWE0 |
| FLASH | FCURAME | KEY[7:0] | | | | | | | |
| | | — | — | — | — | — | — | — | FCRME |
| FLASH | FSTATR0 | FRDY | ILGLERR | ERSERR | PRGERR | SUSRDY | — | ERSSPD | PRGSPD |
| FLASH | FSTATR1 | FCUERR | — | — | FLOCKST | — | — | — | — |
| FLASH | FENTRYR | FEKEY[7:0] | | | | | | | |
| | | FENTRYD | — | — | — | — | — | — | FENTRY1 |
| FLASH | FPROTR | FPKEY[7:0] | | | | | | | |
| | | — | — | — | — | — | — | — | FPROTCN |
| FLASH | FRESETR | FRKEY[7:0] | | | | | | | |
| | | — | — | — | — | — | — | — | FRESET |
| FLASH | FCMDR | CMDR[7:0] | | | | | | | |
| | | PCMDR[7:0] | | | | | | | |
| FLASH | FCPSR | — | — | — | — | — | — | — | — |
| | | — | — | — | — | — | — | — | ESUSPMD |
| FLASH | DFLBCCNT | BCADR[9:0] | | | | | | | |
| | | BCADR[9:0] | | | | | | | — |
| FLASH | FPESTAT | — | — | — | — | — | — | — | — |
| | | PEERRST[7:0] | | | | | | | |
| FLASH | DFLBCSTAT | — | — | — | — | — | — | — | — |
| | | — | — | — | — | — | — | — | BCST |
| FLASH | PCKAR | — | — | — | — | — | — | — | — |
| | | PCKA[7:0] | | | | | | | |

- 注 1. 通过设定 PPG0.PCR, 当脉冲输出组 2 和脉冲输出组 3 的输出触发设定值相同时, PPG0.NDRH 的地址为 000881ECh; 当输出触发设定值不同时, 对应脉冲输出组 2 的 PPG0.NDRH 的地址为 000881EEh, 对应脉冲输出组 3 的 PPG0.NDRH 的地址为 000881ECh。
- 注 2. 通过设定 PPG0.PCR, 当脉冲输出组 0 和脉冲输出组 1 的输出触发设定值相同时, PPG0.NDRL 的地址为 000881EDh; 当输出触发设定值不同时, 对应脉冲输出组 0 的 PPG0.NDRL 的地址为 000881EFh, 对应脉冲输出组 1 的 PPG0.NDRL 的地址为 000881EDh。
- 注 3. 通过设定 PPG1.PCR, 当脉冲输出组 6 和脉冲输出组 7 的输出触发设定值相同时, PPG1.NDRH 的地址为 000881FCh; 当输出触发设定值不同时, 对应脉冲输出组 6 的 PPG1.NDRH 的地址为 000881FEh, 对应脉冲输出组 7 的 PPG1.NDRH 的地址为 000881FCh。
- 注 4. 通过设定 PPG1.PCR, 当脉冲输出组 4 和脉冲输出组 5 的输出触发设定值相同时, PPG1.NDRL 的地址为 000881FDh; 当输出触发设定值不同时, 对应脉冲输出组 4 的 PPG1.NDRL 的地址为 000881FFh, 对应脉冲输出组 5 的 PPG1.NDRL 的地址为 000881FDh。
- 注 5. 在串行通信模式和智能卡接口模式中, 部分位的功能不同。

6. 复位

6.1 概要

复位有引脚复位、深度软件待机复位和看门狗定时器复位，复位名称和复位源如表 6.1 所示。

表 6.1 复位名称和复位源

| 复位名称 | 复位源 |
|----------|-----------------------|
| 引脚复位 | RES# 引脚的输入电压为 Low 电平。 |
| 深度软件待机复位 | 通过中断来解除深度软件待机模式。 |
| 看门狗定时器复位 | 看门狗定时器发生上溢。 |

通过复位对内部状态进行初始化，引脚为初始状态。通过各复位进行初始化的对象如图 6.1 所示。

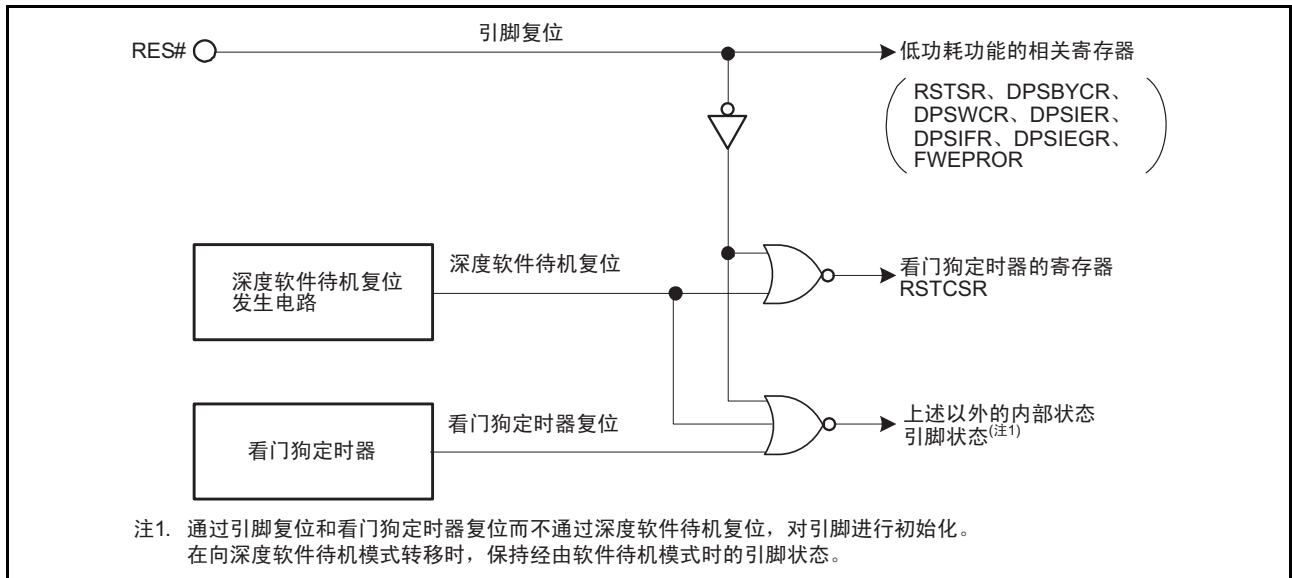


图 6.1 复位电路的框图

表 6.2 各复位种类的初始化对象

| 复位对象 | 复位种类 | | |
|---|------|----------|----------|
| | 引脚复位 | 深度软件待机复位 | 看门狗定时器复位 |
| 低功耗功能的相关寄存器 (RSTSR、DPSBYCR、DPSWCR、 DPSIER、DPSIFR、DPSIEGR、 FWEPROR) | ○ | — | — |
| 看门狗定时器的寄存器 RSTCSR | ○ | ○ | — |
| 上述以外的寄存器和内部状态 | ○ | ○ | ○ |
| 引脚的状态 | ○ | — | ○ |

如果解除复位，就开始复位异常处理。有关复位异常处理，请参照“9. 异常处理”。

复位的相关输入 / 输出引脚如表 6.3 所示。

表 6.3 复位的相关输入 / 输出引脚

| 引脚名 | 输入 / 输出 | 功能 |
|------|---------|------|
| RES# | 输入 | 复位引脚 |

6.2 寄存器说明

复位的相关寄存器一览表如表 6.4 所示。

表 6.4 复位的相关寄存器一览表

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|---------------|--------|-------|------------|------|
| 复位状态寄存器 | RSTSR | 00h | 0008 C285h | 8 |
| 复位的控制 / 状态寄存器 | RSTCSR | 1Fh | 0008 802Bh | 8 |

6.2.1 复位状态寄存器 (RSTSR)

地址 0008 C285h

| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---------|----|----|----|----|----|----|----|----|
| DPSRSTF | — | — | — | — | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|------------|---|----------------|
| b6-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | DPSRSTF | 深度软件待机复位标志 | 0: 不产生外部中断引起的深度软件待机模式的解除请求 1: 产生外部中断引起的深度软件待机模式的解除请求 | R/(W) (注 1) |

注 1. 只能写“0”。

RSTSR 寄存器是表示内部复位的发生源的寄存器。

DPSRSTF 标志 (深度软件待机复位标志)

此标志表示通过 DPSIER 寄存器和 DPSIEGR 寄存器设定的外部中断，解除了深度软件待机模式并产生内部复位。

通过 RES# 引脚的复位信号而不通过解除深度软件待机模式的内部复位信号，对 DPSRSTF 标志进行初始化。

[为“1”的条件]

- 通过外部中断解除了深度软件待机模式时

[为“0”的条件]

- 读“1”后写“0”时

6.2.2 复位的控制 / 状态寄存器 (RSTCSR)

地址 0008 802Bh

| | | | | | | | | |
|-------|------|------|----|----|----|----|----|----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | WOVF | RSTE | — | — | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|------------|--|---------------|
| b4-b0 | — | 保留位 | 读写值都为“1”。 | R/W |
| b5 | — | 保留位 | 读写值都为“0”。 | R/W |
| b6 | RSTE | 复位允许位 | 0: 在看门狗定时器模式中, 即使 TCNT 计数器发生上溢, LSI 内部也不被复位 (WDT 的 TCNT 计数器和 TCSR 寄存器被复位)。 1: 在看门狗定时器模式中, 当 TCNT 计数器发生上溢时, LSI 内部被复位。 | R/W |
| b7 | WOVF | 看门狗定时器上溢标志 | 0: 在看门狗定时器模式中, TCNT 计数器没有发生上溢。 1: 在看门狗定时器模式中, TCNT 计数器发生上溢。 | R(W) (注 1) |

注 1. 只能写“0”。

RSTCSR 寄存器控制 TCNT 计数器的上溢引起内部复位信号的产生, 并且选择内部复位信号的种类。

通过 RES# 引脚的复位信号和深度软件待机复位, 将 RSTCSR 寄存器初始化为“1Fh”, 而不通过 WDT 的上溢产生的内部复位信号进行初始化。

必须以 8 位为单位读 RSTCSR 寄存器。

写 RSTCSR 寄存器时, 必须以 16 位为单位写 WINB 寄存器。

详细内容请参照“19.5.1 存取寄存器时的注意事项”。

RSTE 位 (复位允许位)

在看门狗定时器模式中, 此位选择是否通过 TCNT 计数器的上溢对 LSI 内部进行复位。

WOVF 标志 (看门狗定时器上溢标志)

在看门狗定时器模式中, WOVF 标志表示 TCNT 计数器发生上溢。在间隔定时器模式中, 此标志不变为“1”。

[为“1”的条件]

- 在看门狗定时器模式中, TCNT 计数器发生上溢 (“FFh” → “00h”) 时

[为“0”的条件]

- 读“1”后写“0”时

6.3 运行说明

6.3.1 引脚复位

这是 RES# 引脚引起的复位。

当 RES# 引脚为 Low 电平时，停止正在执行中的全部处理，进入复位状态。

为了确保复位，必须在接通电源时根据规定的振荡稳定时间，使 RES# 引脚保持 Low 电平。必须在运行中按照规定的复位脉宽，使 RES# 引脚保持 Low 电平。详细内容请参照“29. 电特性”。

6.3.2 深度软件待机复位

这是通过中断来解除深度软件待机模式时产生的内部复位。

如果解除深度软件待机模式，就在时钟开始振荡的同时产生深度软件待机复位。在经过由深度软件待机等待时间的设定值（DPSWCR.WTSTS[5:0]）选择的时间后，解除深度软件待机复位。

深度软件待机复位的详细内容请参照“8. 低功耗功能”。

6.3.3 看门狗定时器复位

这是看门狗定时器引起的内部复位。

如果将 RSTCSR.RSTE 位置“1”，就在看门狗定时器发生上溢时产生看门狗定时器复位。然后，在经过一定的时间后，解除看门狗定时器复位。

看门狗定时器复位的详细内容请参照“19. 看门狗定时器（WDT）”。

6.4 复位发生源的判断

能通过读 RSTCSR 寄存器和 RSTSR 寄存器，确认是因发生哪种复位而执行了复位异常处理。复位发生源的判断流程例子如图 6.2 所示。

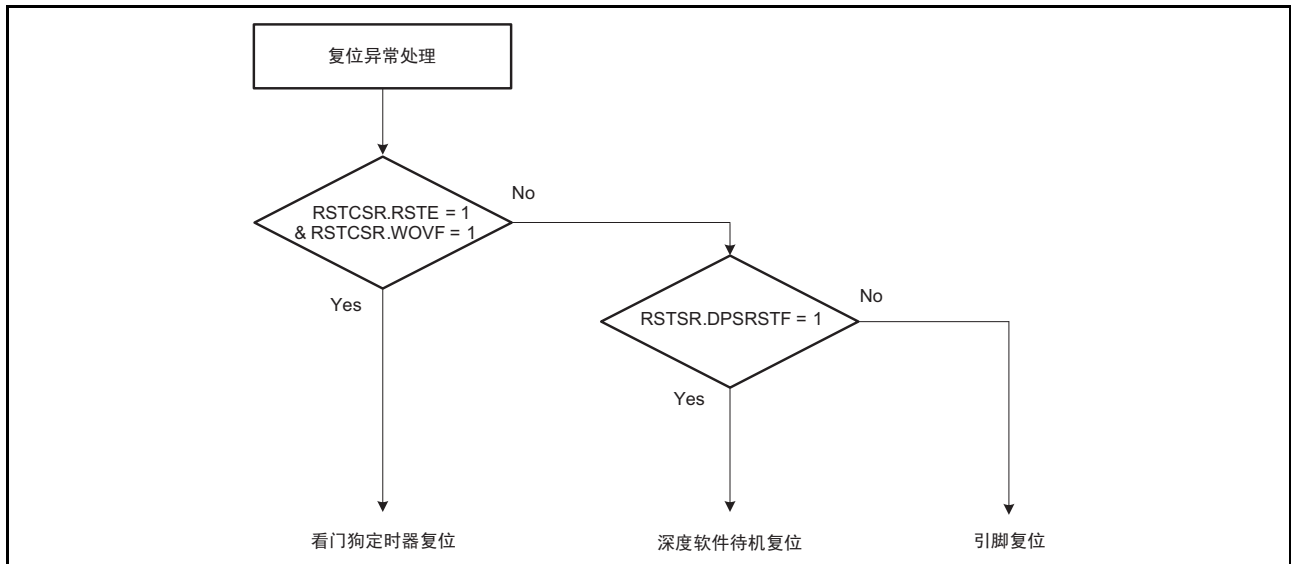


图 6.2 复位发生源的判断流程例子

6.5 使用时的注意事项

6.5.1 设计电路板时的注意事项

RX610 群的 XTAL 引脚邻接复位引脚。因此，必须通过 GND 保护复位信号，以免受到时钟信号的影响。

7. 时钟发生电路

7.1 概要

RX610 群内置了时钟发生电路，生成系统时钟（ICLK）、外围模块时钟（PCLK）和外部总线时钟（BCLK）。

时钟发生电路由主时钟振荡器、PLL（Phase Locked Loop）电路、分频器和选择电路构成。

时钟发生电路的规格和框图分别如表 7.1 和图 7.1 所示。

表 7.1 时钟发生电路的规格

| 项目 | 规格 |
|----------------------|---|
| 用途 | <ul style="list-style-type: none"> 生成给 CPU、DTC、DMAC、ROM 和 RAM 提供的系统时钟（ICLK）。 生成给外围模块提供的外围模块时钟（PCLK）。 生成给外部总线提供的外部总线时钟（BCLK）。 |
| 输入时钟（EXTAL）频率 | 8MHz ~ 14MHz |
| ICLK/PCLK/BCLK 时钟的选择 | 可给 ICLK/PCLK/BCLK 分别选择 EXTAL×8、×4、×2、×1。 |
| 工作频率 | ICLK: 8MHz ~ 100MHz PCLK: 8MHz ~ 50MHz BCLK: 8MHz ~ 25MHz 时钟频率的设定限制：维持 ICLK ≥ PCLK、ICLK ≥ BCLK。 |
| 能连接的振荡器或者附加电路 | 晶体谐振器 |
| 振荡器或者附加电路的连接引脚 | EXTAL、XTAL |
| BCLK 输出控制功能 | 可选择 BCLK 输出或者 High 输出。 |

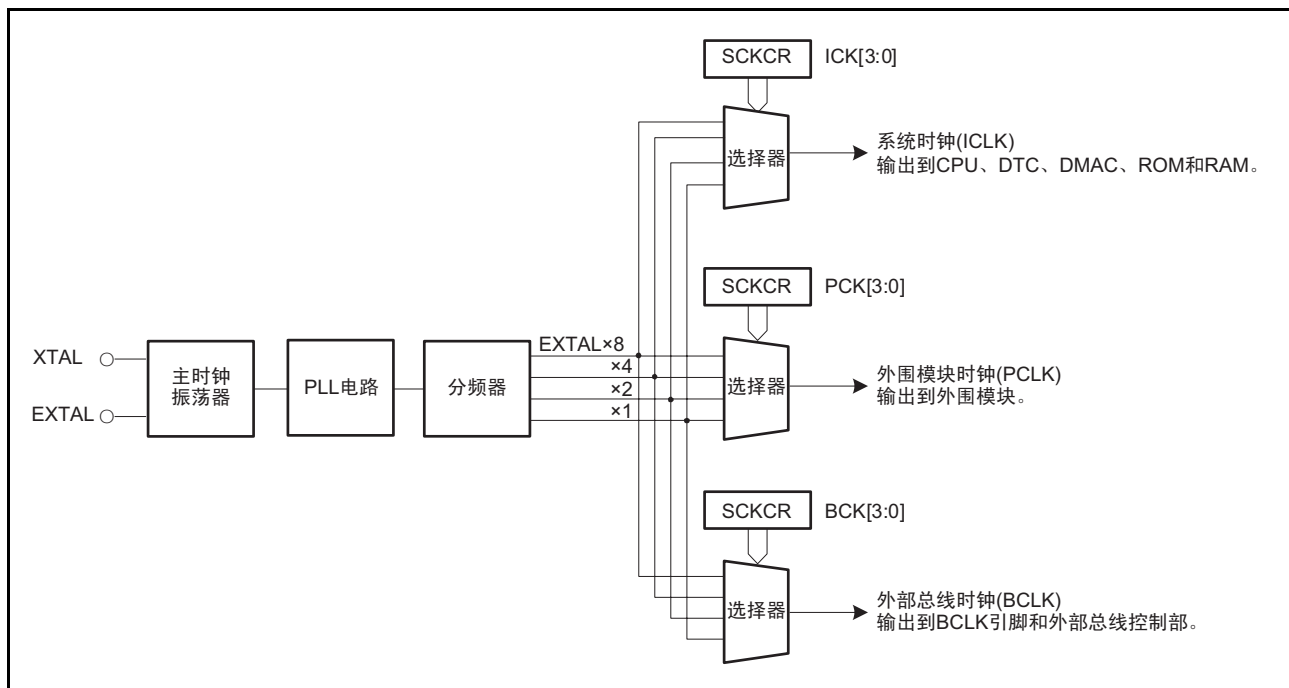


图 7.1 时钟发生电路的框图

时钟发生电路的输入 / 输出引脚如表 7.2 所示。

表 7.2 时钟发生电路的输入 / 输出引脚

| 引脚名 | 输入 / 输出 | 功能 |
|-------|---------|---|
| XTAL | 输入 | 连接晶体谐振器的引脚 |
| EXTAL | 输入 | EXTAL 引脚也可以输入外部时钟，详细内容请参照“7.3.2 输入外部时钟的方法”。 |
| BCLK | 输出 | 给外部设备提供外部总线时钟（BCLK）。 |

7.2 寄存器说明

时钟发生电路的寄存器一览表如表 7.3 所示。

表 7.3 时钟发生电路的寄存器一览表

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|-----------|-------|------------|------------|------|
| 系统时钟控制寄存器 | SCKCR | 0202 0200h | 0008 0020h | 32 |

7.2.1 系统时钟控制寄存器 (SCKCR)

地址 0008 0020h

| | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|----------|-----|-----|-----|--------|-----|-----|-----|----------|-----|-----|-----|---|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 | |
| — | — | — | — | ICK[3:0] | | | | PSTOP1 | — | — | — | BCK[3:0] | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |
| 复位后的值 | | | | | | | | | | | | | | | | |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | |
| — | | | | PCK[3:0] | | | | — | — | — | — | — | — | — | — | — |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 复位后的值 | | | | | | | | | | | | | | | | |

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|------------------|-------------------|---|-----|
| b7-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b11-b8 | PCK[3:0] (注1) | 外围模块时钟 (PCLK) 选择位 | b11 b8 0000: ×8 0001: ×4 0010: ×2 0011: ×1 上述以外: 不能设定 | R/W |
| b15-b12 | — | 保留位 | 读写值都为“0”。 | R/W |
| b19-b16 | BCK[3:0] (注1) | 外部总线时钟 (BCLK) 选择位 | b19 b16 0000: ×8 0001: ×4 0010: ×2 0011: ×1 上述以外: 不能设定 | R/W |
| b22-b20 | — | 保留位 | 读写值都为“0”。 | R/W |
| b23 | PSTOP1 | BCLK 输出停止位 | 0: BCLK 输出 1: 固定为 High 电平 | R/W |
| b27-b24 | ICK[3:0] (注2) | 系统时钟 (ICLK) 选择位 | b27 b24 0000: ×8 0001: ×4 0010: ×2 0011: ×1 上述以外: 不能设定 | R/W |
| b31-b28 | — | 保留位 | 读写值都为“0”。 | R/W |

注 1. 不能设定大于系统时钟 (ICLK) 的频率, 否则尽管能设定寄存器, 但是频率和 ICLK 的频率相同。

注 2. 不能设定低于外围模块时钟 (PCLK) 和外部总线时钟 (BCLK) 的频率, 否则尽管能设定寄存器, 但是系统时钟 (ICLK) 的频率和 PCLK 或者 BCLK 的频率相同。

SCKCR 寄存器选择 BCLK 输出控制以及系统时钟 (ICLK)、外围模块时钟 (PCLK) 和外部总线时钟 (BCLK) 的频率。

PCK[3:0] 位（外围模块时钟（PCLK）选择位）

这些位选择外围模块时钟（PCLK）的频率，并且表示输入时钟（EXTAL）的倍率。

BCK[3:0] 位（外部总线时钟（BCLK）选择位）

这些位选择外部总线时钟（BCLK）的频率，并且表示输入时钟（EXTAL）的倍率。

PSTOP1 位（BCLK 输出停止位）

此位控制 P53 的 BCLK 输出。

ICK[3:0] 位（系统时钟（ICLK）选择位）

这些位选择 CPU、DMAC、DTC 和系统时钟（ICLK）的频率，并且表示输入时钟（EXTAL）的倍率。

7.3 主时钟振荡器

提供时钟的方法有连接晶体谐振器和输入外部时钟两种。

7.3.1 连接晶体谐振器的方法

晶体谐振器的连接例子和阻尼电阻 R_d （参考值）分别如图 7.2 和表 7.4 所示。晶体谐振器必须使用 AT-cut 并联谐振型。

在连接晶体谐振器后提供时钟时，连接的晶体谐振器必须为 8 ~ 14MHz。

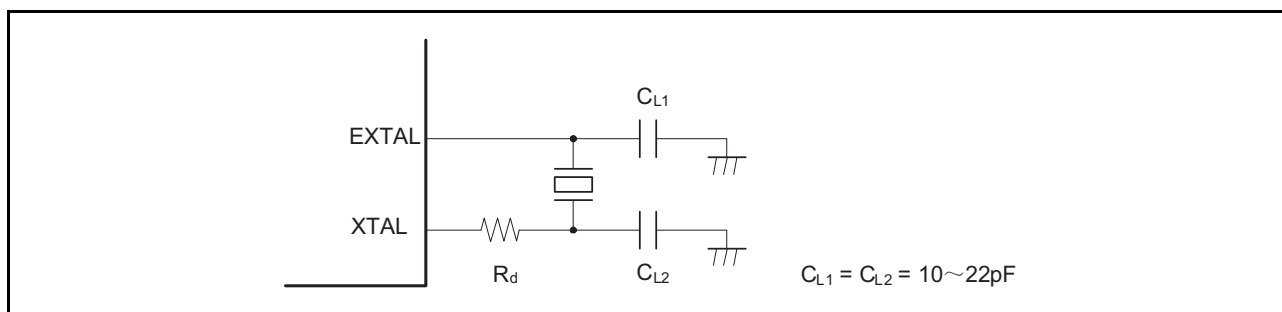


图 7.2 晶体谐振器的连接例子

表 7.4 阻尼电阻（参考值）

| 频率 (MHz) | 8 | 10 | 12 | 14 |
|--------------------|-----|-----|----|----|
| R_d (Ω) | 200 | 100 | 0 | 0 |

晶体谐振器的等效电路如图 7.3 所示，必须使用表 7.5 所示特性的晶体谐振器。

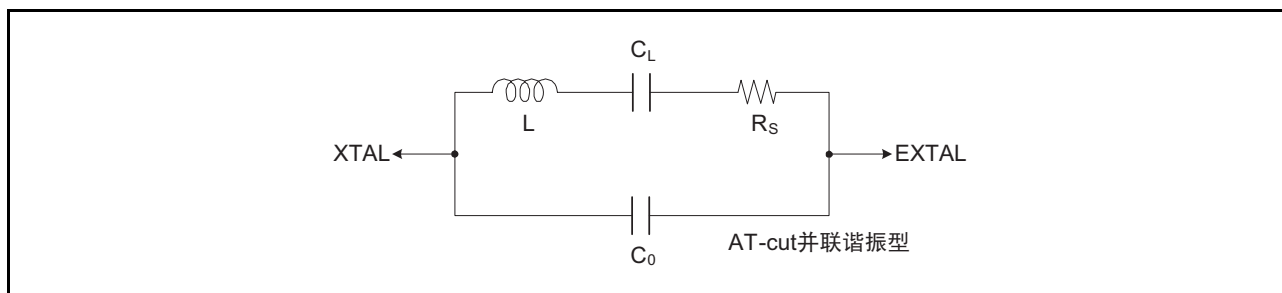


图 7.3 晶体谐振器的等效电路

表 7.5 晶体谐振器的特性（参考值）

| 频率 (MHz) | 8 | 10 | 12 | 14 |
|------------------------|----|----|----|----|
| R_s max (Ω) | 80 | 70 | 60 | 50 |
| C_0 max (pF) | 7 | | | |

7.3.2 输入外部时钟的方法

外部时钟输入的连接例子如图 7.4 所示。在将 XTAL 引脚置为开路时，寄生电容不能超过 10pF。在给 XTAL 引脚输入反相时钟时，必须在待机模式中将外部时钟置为 High 电平。

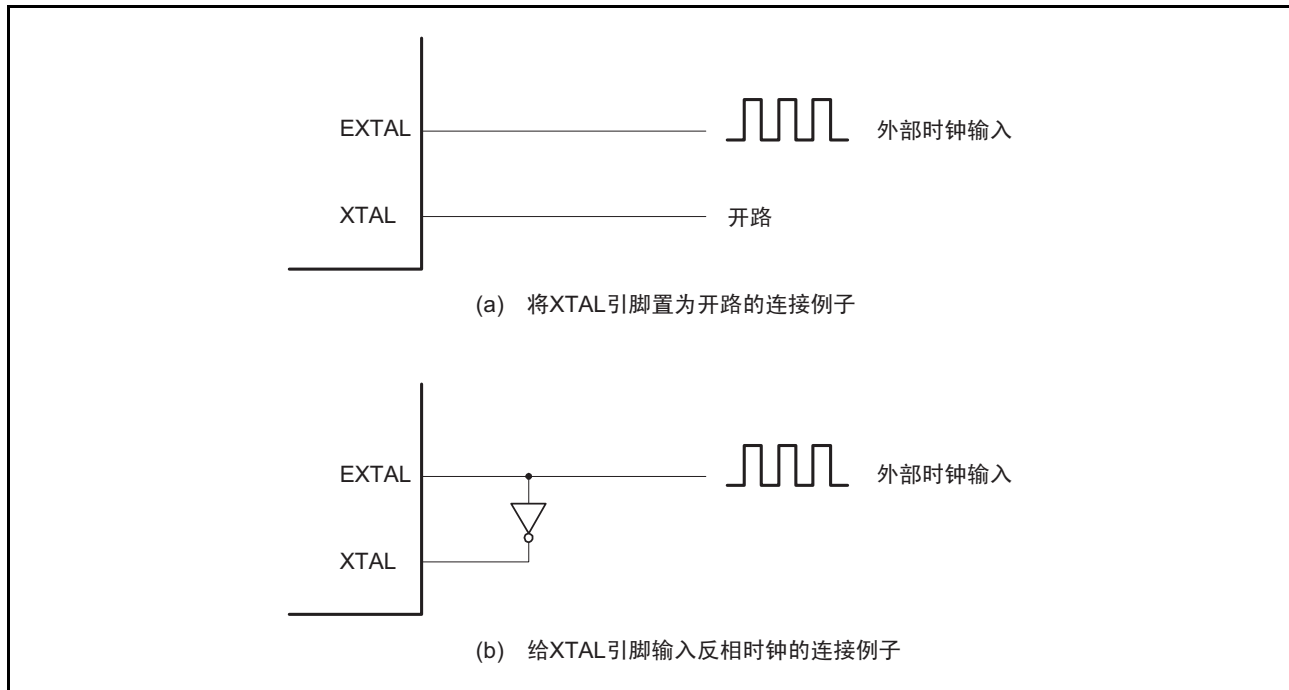


图 7.4 外部时钟输入的连接例子

7.4 PLL 电路

PLL 电路具有将振荡器的频率最大递增 8 倍的功能。

7.5 分频器

分频器对 PLL 时钟进行分频，生成 1/2、1/4、1/8 的时钟。如果改写 SCKCR.ICK[3:0] 位、SCKCR.PCK[3:0] 位和 SCKCR.BCK[3:0] 位，就以该频率进行工作。

7.6 内部时钟

内部时钟是通过 PLL 电路将外部的输入时钟 (EXTAL) 递增 8 倍，并且通过分频器进行 1 分频、2 分频、4 分频或者 8 分频后的时钟。

内部时钟有以下 3 种时钟：

- CPU、DMAC、DTC 的运行时钟：系统时钟 (ICLK)
- 外围模块的运行时钟：外围模块时钟 (PCLK)
- 外部总线控制部、用于外部引脚输出的时钟：外部总线时钟 (BCLK)

通过 SCKCR.ICK[3:0] 位、SCKCR.PCK[3:0] 位和 SCKCR.BCK[3:0] 位的组合来设定频率。

7.6.1 系统时钟 (ICLK)

系统时钟 (ICLK) 是 CPU、DMAC、DTC、ROM 和 RAM 的运行时钟。

通过 SCKCR.ICK[3:0] 位设定 ICLK 的频率。

不能将 ICLK 的频率设定为低于外围模块时钟 (PCLK) 和外部总线时钟 (BCLK) 的频率, 否则该频率就和 PCLK 或者 BCLK 的频率相同。

7.6.2 外围模块时钟 (PCLK)

外围模块时钟 (PCLK) 是外围模块的运行时钟。

通过 SCKCR.PCK[3:0] 位设定 PCLK 的频率。

不能将 PCLK 的频率设定为高于系统时钟 (ICLK) 的频率, 否则该频率就和 ICLK 的频率相同。

7.6.3 外部总线时钟 (BCLK)

外部总线时钟 (BCLK) 是连接外部总线的外部引脚输出时钟。

如果将 SCKCR.PSTOP1 位置 “0” (BCLK 输出) 并且将 P5.DDR.B3 位置 “1” (输出端口), 就能从 BCLK 输出引脚输出 BCLK。但是, 需要注意 SCKCR.PSTOP1 位和 P5.DDR.B3 位的设定顺序, 必须在 SCKCR.PSTOP1 位为 “1” (固定为 High 电平) 的状态下更改 P5.DDR.B3 位的值。

通过 SCKCR.BCK[3:0] 位设定 BCLK 的频率。不能将 BCLK 的频率设定为高于系统时钟 (ICLK) 的频率, 否则该频率就和 ICLK 的频率相同。

7.7 使用时的注意事项

7.7.1 有关时钟发生电路的注意事项

1. 通过 SCKCR 寄存器选择给各模块提供的系统时钟 (ICLK)、外围模块时钟 (PCLK) 和外部总线时钟 (BCLK) 的频率。各频率必须设定在电特性的 AC 特性的时钟周期时间 (tcyc) 的运行保证范围内, 如下所示:

$$\text{ICLK}=8\sim 100\text{MHz}、\text{PCLK}=8\sim 50\text{MHz}、\text{BCLK}=8\sim 25\text{MHz}$$

外围模块 (DMAC 和 DTC 除外) 全部以 PCLK 为基准运行。因此必须注意: 在更改频率的前后, 定时器和 SCI 等的运行速度将发生变化。

另外, 用于解除软件待机模式的待机时间也因更改频率而发生变化, 详细内容请参照 “8.5.3.3 解除软件待机模式后的振荡稳定时间的设定”。

2. 在系统时钟 (ICLK)、外围模块时钟 (PCLK) 和外部总线时钟 (BCLK) 之间, $\text{ICLK} \geq \text{PCLK}$ 、 $\text{ICLK} \geq \text{BCLK}$ 的关系成立, 并且优先 ICLK 的设定。因此, 如果设定没有满足此条件, SCKCR.PCK[3:0] 位和 SCKCR.BCK[3:0] 位的设定就无效, PCLK 和 BCLK 为 SCKCR.ICK[3:0] 位设定的时钟频率。
3. 在更改时钟频率时, 必须注意在存取外部总线时不要改变频率。
4. 在写 SCKCR 寄存器后到频率变更结束的期间, 如果重写 SCKCR 寄存器, 就忽视此写操作。如果继续写 SCKCR 寄存器, 就必须确认是否能从 SCKCR 寄存器中读到最后写的值。
5. 在写 SCKCR 寄存器后到频率变更结束的期间, 禁止转移到软件待机模式。如果在更改频率的中途转移到软件待机模式, 就无法保证以后的运行。在写 SCKCR 寄存器后到发行 WAIT 指令的期间, 必须通过系统时钟至少间隔 11 个周期。

7.7.2 有关振荡器的注意事项

因为振荡器的各种特性与用户的电路板设计密切相关，所以请用户参考本章介绍的振荡器的连接例子，在给予充分的评估后使用。振荡器的电路常数因振荡器和安装电路的寄生电容而不同，因此必须和振荡器厂商仔细商谈后作决定。外加在振荡引脚的电压不能超过绝对最大额定值。

7.7.3 设计电路板时的注意事项

在使用晶体谐振器时，必须尽量将振荡器和电容器排列在 XTAL 引脚和 EXTAL 引脚附近。如图 7.5 所示，在振荡电路的附近不能通过信号线，否则有可能因电感而不能正常振荡。

RX610 群的 XTAL 引脚邻接复位引脚。因此，必须通过 GND 保护复位信号，以免受到时钟信号的影响。

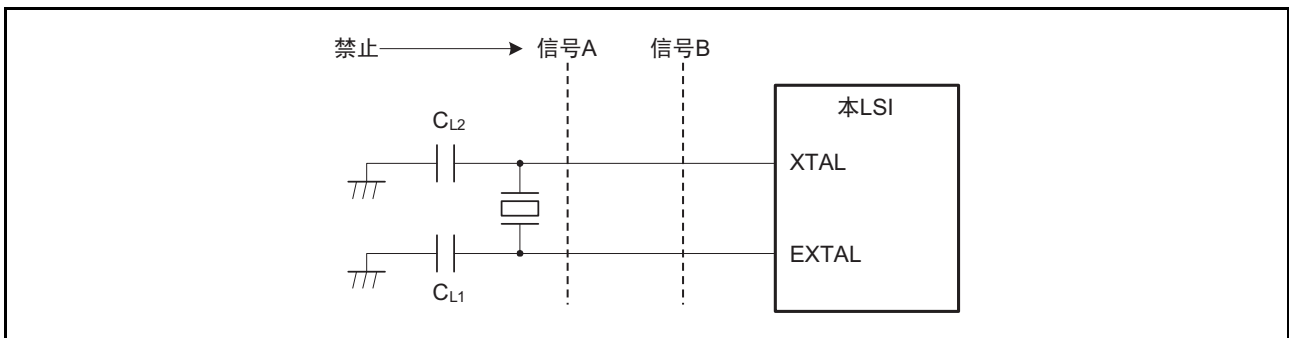


图 7.5 有关振荡电路的电路板设计的注意事项

外接 PLL 电路的推荐电路如图 7.6 所示，必须将 PLLVCC、PLLVSS 和 VCC、VSS 远离电路板的电源供给源，并且在引脚附近插入旁路电容器 CPB 和 CB。

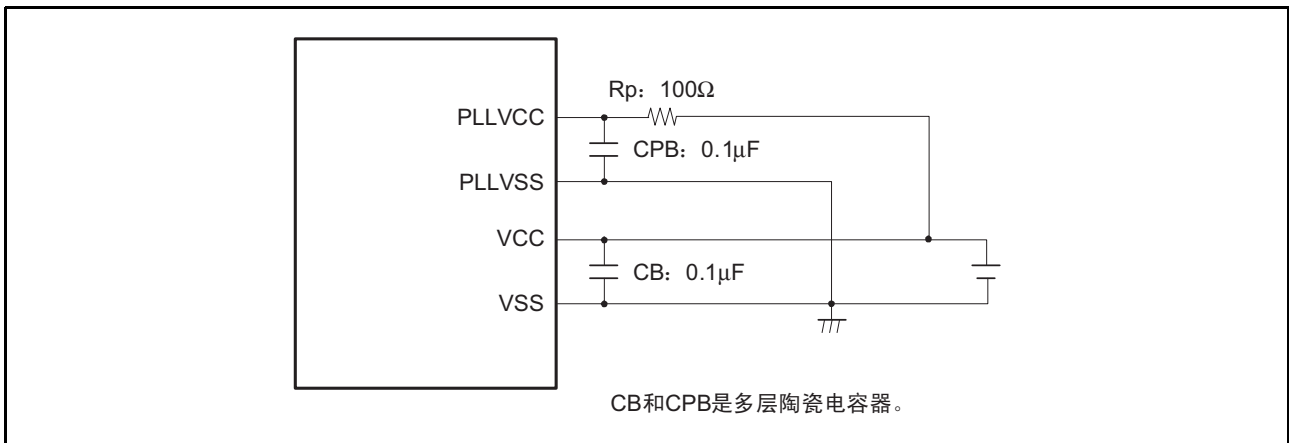


图 7.6 外接 PLL 电路的推荐电路

8. 低功耗功能

8.1 概要

RX610 群的低功耗功能有多时钟功能、BCLK 输出停止功能、模块停止功能和向低功耗状态转移的功能。

低功耗功能的规格如表 8.1 所示，向低功耗状态转移的条件、CPU 和外围模块等的状态以及各模式的解除方法如表 8.2 所示。

复位后，在正常的程序运行中，DTC 和 DMAC 以外的模块处于停止状态。

表 8.1 低功耗功能的规格

| 项目 | 内容 |
|-------------|--|
| 多时钟功能 | 能给系统时钟（ICLK）、外围模块时钟（PCLK）和外部总线时钟（BCLK）分别设定分频比。 |
| BCLK 输出控制功能 | 可选择 BCLK 输出或者 High 电平输出。 |
| 模块停止功能 | 能停止各外围模块的功能。 |
| 向低功耗状态转移的功能 | 能进入使 CPU、外围模块和振荡器停止运行的低功耗状态。 |
| 低功耗状态 | 睡眠模式 全模块时钟停止模式 软件待机模式 深度软件待机模式 |

表 8.2 各模式的转移方法、解除方法和运行状态

| 转移方法、解除方法和运行状态 | 睡眠模式 | 全模块时钟停止模式 | 软件待机模式 | 深度软件待机模式 |
|--------------------------------------|-------------------|-------------------|-------------------|---------------------|
| 转移方法 | 控制寄存器 + 指令 | 控制寄存器 + 指令 | 控制寄存器 + 指令 | 控制寄存器 + 指令 |
| 解除方法 | 中断 | 中断 (注1) | 外部中断 | 外部中断 (注2) |
| 解除后的状态 (注3) | 程序的执行状态 (中断处理) | 程序的执行状态 (中断处理) | 程序的执行状态 (中断处理) | 程序的执行状态 (复位处理) |
| 振荡器 | 运行 | 运行 | 停止 | 停止 |
| CPU | 停止 (保持) | 停止 (保持) | 停止 (保持) | 停止 (不定值) |
| 内部 RAM1 (0001 0000h ~ 0001 FFFFh) | 运行 (保持) | 停止 (保持) | 停止 (保持) | 停止 (不定值) |
| 内部 RAM0 (0000 0000h ~ 0000 FFFFh) | 运行 (保持) | 停止 (保持) | 停止 (保持) | 停止 (保持/不定值) (注4) |
| 看门狗定时器 | 运行 | 运行 | 停止 (保持) | 停止 (不定值) |
| 8 位定时器 (单元 0 和单元 1) | 运行 | 运行 (注5) | 停止 (保持) | 停止 (不定值) |
| 外围模块 | 运行 | 停止 (注6) | 停止 (注6) | 停止 (不定值) |
| I/O 端口 | 运行 | 保持 (注7) | 保持 (注8) | 保持 (注8) |

停止 (保持) 表示保持内部寄存器的值并且内部状态为运行中止状态。

停止 (不定值) 表示内部寄存器的值为不定值并且内部状态为断电状态。

注 1. 外部中断和部分内部中断 (8 位定时器和看门狗定时器)

注 2. 只限于 NMI 和 IRQ0 ~ IRQ3 的 A 侧。但是, 只在深度待机中断允许寄存器 (DPSIER) 的该位为 “1” 时, NMI 和 IRQ 有效。

注 3. 通过 RES# 引脚进行的解除除外。当通过 RES# 引脚进行解除时, 就转移到复位状态。

注 4. 能通过设定深度待机控制寄存器的内部 RAM 断开 2 的位、内部 RAM 断开 1 的位和内部 RAM 断开 0 的位 (DPSBYCR.RAMCUT2、DPSBYCR.RAMCUT1 和 DPSBYCR.RAMCUT0), 选择保持或者不定值。

注 5. 能通过设定模块停止控制寄存器 A 的 8 位定时器 3、定时器 2 (单元 1) 的模块停止设定位以及 8 位定时器 1、定时器 0 (单元 0) 的模块停止设定位 (MSTPCRA.MSTPA4 和 MSTPCRA.MSTPA5), 选择运行或者停止。

注 6. 外围模块保持原来的状态。

注 7. 如果将 P53 用作 BCLK 输出, 就继续进行 BCLK 输出, 详细内容请参照 “8.6 BCLK 输出的控制”。

注 8. 能通过设定待机控制寄存器的输出端口允许位 (SBYCR.OPE), 选择地址总线和总线控制信号 (CS0# ~ CS7#、RD#、WR0#、WR1#、WR#、BC0#、BC1#) 为保持状态或者高阻抗状态。

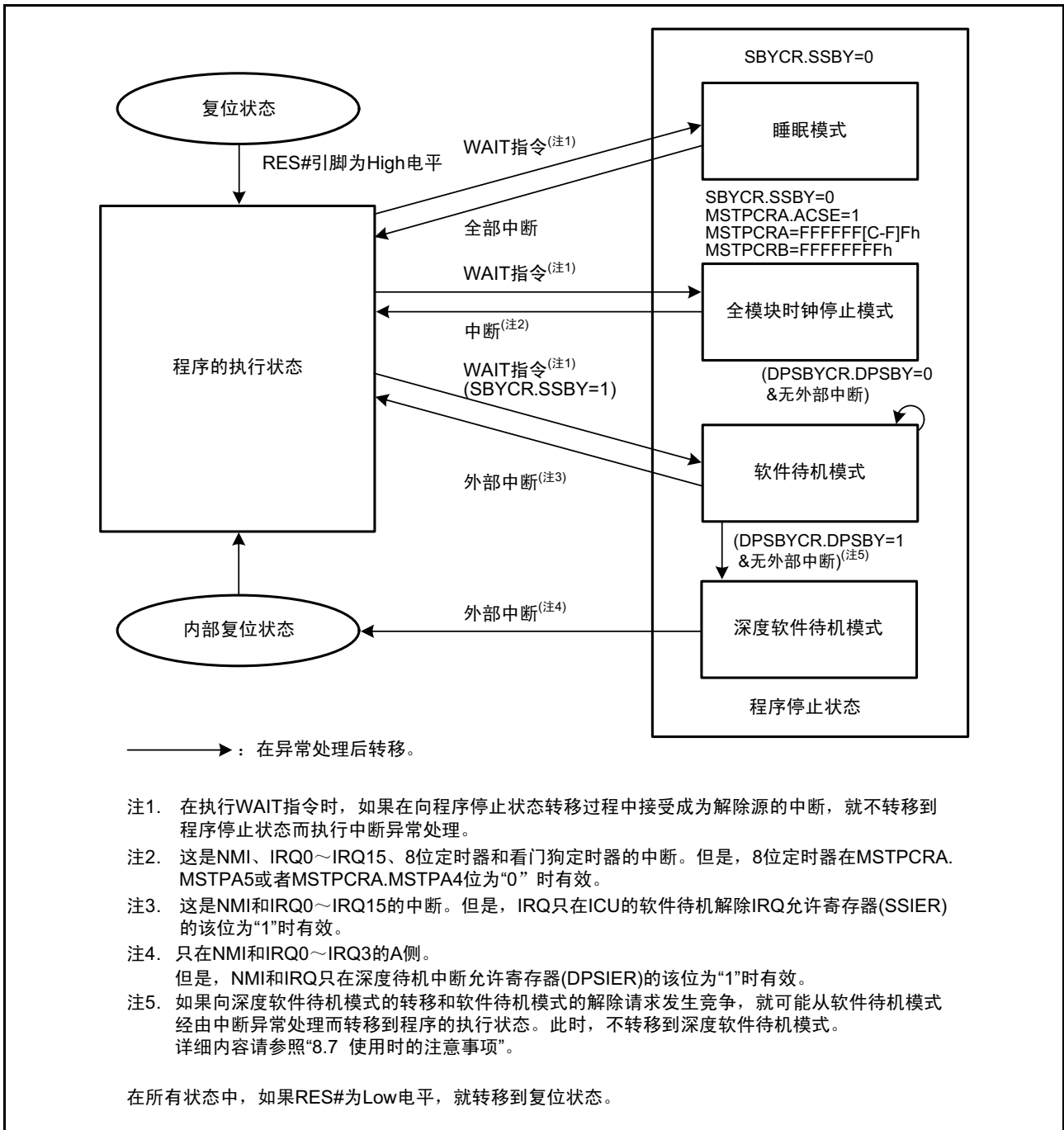


图 8.1 模式转移

8.2 寄存器说明

低功耗的相关寄存器如表 8.3 所示。有关系统时钟控制寄存器（SCKCR），请参照“8.2.1 待机控制寄存器（SBYCR）”。

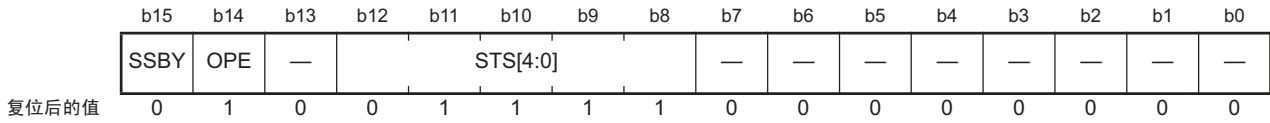
表 8.3 低功耗的相关寄存器一览表

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|--------------|----------|------------|------------|------|
| 待机控制寄存器 | SBYCR | 4F00h | 0008 000Ch | 16 |
| 模块停止控制寄存器 A | MSTPCRA | 67FF FFFFh | 0008 0010h | 32 |
| 模块停止控制寄存器 B | MSTPCRB | FFFF FFFFh | 0008 0014h | 32 |
| 模块停止控制寄存器 C | MSTPCRC | FFFF 0000h | 0008 0018h | 32 |
| 深度待机控制寄存器 | DPSBYCR | 31h | 0008 C280h | 8 |
| 深度待机等待控制寄存器 | DPSWCR | 0Fh | 0008 C281h | 8 |
| 深度待机中断允许寄存器 | DPSIER | 00h | 0008 C282h | 8 |
| 深度待机中断标志寄存器 | DPSIFR | 00h | 0008 C283h | 8 |
| 深度待机中断边沿寄存器 | DPSIEGR | 00h | 0008 C284h | 8 |
| 复位状态寄存器 | RSTSR | 00h | 0008 C285h | 8 |
| 深度待机备用寄存器 0 | DPSBKR0 | xxh (注 1) | 0008 C290h | 8 |
| 深度待机备用寄存器 1 | DPSBKR1 | xxh (注 1) | 0008 C291h | 8 |
| 深度待机备用寄存器 2 | DPSBKR2 | xxh (注 1) | 0008 C292h | 8 |
| 深度待机备用寄存器 3 | DPSBKR3 | xxh (注 1) | 0008 C293h | 8 |
| 深度待机备用寄存器 4 | DPSBKR4 | xxh (注 1) | 0008 C294h | 8 |
| 深度待机备用寄存器 5 | DPSBKR5 | xxh (注 1) | 0008 C295h | 8 |
| 深度待机备用寄存器 6 | DPSBKR6 | xxh (注 1) | 0008 C296h | 8 |
| 深度待机备用寄存器 7 | DPSBKR7 | xxh (注 1) | 0008 C297h | 8 |
| 深度待机备用寄存器 8 | DPSBKR8 | xxh (注 1) | 0008 C298h | 8 |
| 深度待机备用寄存器 9 | DPSBKR9 | xxh (注 1) | 0008 C299h | 8 |
| 深度待机备用寄存器 10 | DPSBKR10 | xxh (注 1) | 0008 C29Ah | 8 |
| 深度待机备用寄存器 11 | DPSBKR11 | xxh (注 1) | 0008 C29Bh | 8 |
| 深度待机备用寄存器 12 | DPSBKR12 | xxh (注 1) | 0008 C29Ch | 8 |
| 深度待机备用寄存器 13 | DPSBKR13 | xxh (注 1) | 0008 C29Dh | 8 |
| 深度待机备用寄存器 14 | DPSBKR14 | xxh (注 1) | 0008 C29Eh | 8 |
| 深度待机备用寄存器 15 | DPSBKR15 | xxh (注 1) | 0008 C29Fh | 8 |
| 深度待机备用寄存器 16 | DPSBKR16 | xxh (注 1) | 0008 C2A0h | 8 |
| 深度待机备用寄存器 17 | DPSBKR17 | xxh (注 1) | 0008 C2A1h | 8 |
| 深度待机备用寄存器 18 | DPSBKR18 | xxh (注 1) | 0008 C2A2h | 8 |
| 深度待机备用寄存器 19 | DPSBKR19 | xxh (注 1) | 0008 C2A3h | 8 |
| 深度待机备用寄存器 20 | DPSBKR20 | xxh (注 1) | 0008 C2A4h | 8 |
| 深度待机备用寄存器 21 | DPSBKR21 | xxh (注 1) | 0008 C2A5h | 8 |
| 深度待机备用寄存器 22 | DPSBKR22 | xxh (注 1) | 0008 C2A6h | 8 |
| 深度待机备用寄存器 23 | DPSBKR23 | xxh (注 1) | 0008 C2A7h | 8 |
| 深度待机备用寄存器 24 | DPSBKR24 | xxh (注 1) | 0008 C2A8h | 8 |
| 深度待机备用寄存器 25 | DPSBKR25 | xxh (注 1) | 0008 C2A9h | 8 |
| 深度待机备用寄存器 26 | DPSBKR26 | xxh (注 1) | 0008 C2AAh | 8 |
| 深度待机备用寄存器 27 | DPSBKR27 | xxh (注 1) | 0008 C2ABh | 8 |
| 深度待机备用寄存器 28 | DPSBKR28 | xxh (注 1) | 0008 C2ACh | 8 |
| 深度待机备用寄存器 29 | DPSBKR29 | xxh (注 1) | 0008 C2ADh | 8 |
| 深度待机备用寄存器 30 | DPSBKR30 | xxh (注 1) | 0008 C2AEh | 8 |
| 深度待机备用寄存器 31 | DPSBKR31 | xxh (注 1) | 0008 C2AFh | 8 |

注 1. 不对 DPSBKR0 ~ DPSBKR31 寄存器进行初始化，刚接通电源后的寄存器的值为不定值。

8.2.1 待机控制寄存器 (SBYCR)

地址 0008 000Ch



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|----------|----------|---|-----|
| b7-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b12-b8 | STS[4:0] | 待机定时器选择位 | b12 b8 0 0 1 0 1: 待机时间 =64 个状态 0 0 1 1 0: 待机时间 =512 个状态 0 0 1 1 1: 待机时间 =1024 个状态 0 1 0 0 0: 待机时间 =2048 个状态 0 1 0 0 1: 待机时间 =4096 个状态 0 1 0 1 0: 待机时间 =16384 个状态 0 1 0 1 1: 待机时间 =32768 个状态 0 1 1 0 0: 待机时间 =65536 个状态 0 1 1 0 1: 待机时间 =131072 个状态 0 1 1 1 0: 待机时间 =262144 个状态 0 1 1 1 1: 待机时间 =524288 个状态 上述以外: 不能设定 | R/W |
| b13 | — | 保留位 | 读写值都为“0”。 | R/W |
| b14 | OPE | 输出端口允许位 | 0: 在软件待机模式和深度软件待机模式中, 地址总线和总线控制信号为高阻抗状态。 1: 在软件待机模式和深度软件待机模式中, 地址总线和总线控制信号保持输出状态。 | R/W |
| b15 | SSBY | 软件待机位 | 0: 在执行 WAIT 指令后转移到睡眠模式或者全模块时钟停止模式。 1: 在执行 WAIT 指令后转移到软件待机模式。 | R/W |

SBYCR 寄存器是控制软件待机模式的寄存器。

STS[4:0] 位 (待机定时器选择位)

在通过外部中断解除软件待机模式时, 此位选择时钟稳定前的待机时间。

在使用晶体振荡的情况下, 必须参照表 8.4 并且根据工作频率选择大于等于振荡稳定时间的待机时间; 在使用外部时钟的情况下, 也需要 PLL 电路的稳定时间, 请参照表 8.4 设定待机时间。

必须注意: 在使用多时钟功能的情况下, 在振荡稳定期间, 以外围模块时钟 (PCLK) 的频率进行计数。

OPE 位 (输出端口允许位)

在软件待机模式和深度软件待机模式中, 此位能选择是将地址总线和总线控制信号 (CS0# ~ CS7#、RD#、WR0#、WR1#、WR#、BC0#、BC1#) 保持为输出状态还是置为高阻抗状态。

SSBY 位（软件待机位）

此位设定执行 WAIT 指令后的转移目标。

当 SSBY 位为“0”时，根据 MSTPCRA 寄存器和 MSTPCRB 寄存器的设定，在执行 WAIT 指令后转移到睡眠模式或者全模块时钟停止模式；当 SSBY 位为“1”时，在执行 WAIT 指令后转移到软件待机模式。此时，如果 DPSBYCR.DPSBY 位为“1”，就经由软件待机模式转移到深度软件待机模式，详细内容请参照“8.5 低功耗状态”。

如果在通过外部中断解除软件待机模式后转移到正常模式，此位就保持“1”。要将此位置“0”时，必须写“0”。

在看门狗定时器模式中使用 WDT 时，此位的设定值无效，并且在执行 WAIT 指令后总是转移到睡眠模式或者全模块时钟停止模式。

8.2.2 模块停止控制寄存器 A（MSTPCRA）

地址 0008 0010h

| | | | | | | | | | | | | | | | | |
|-------|-------------|-------------|-------------|-------------|-------------|-------------|-----|-----|-------------|-------------|-------------|-------------|-------------|-----|-----|-----|
| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| | ACSE | — | — | MSTPA 28 | MSTPA 27 | — | — | — | MSTPA 23 | MSTPA 22 | MSTPA 21 | MSTPA 20 | MSTPA 19 | — | — | — |
| 复位后的值 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | MSTPA 15 | MSTPA 14 | MSTPA 13 | MSTPA 12 | MSTPA 11 | MSTPA 10 | — | — | — | — | MSTPA 5 | MSTPA 4 | — | — | — | — |
| 复位后的值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|------------------------------|--|-----|
| b3-b0 | — | 保留位 | 读写值都为“1”。 | R/W |
| b4 | MSTPA4 | 8 位定时器 3、定时器 2（单元 1）的模块停止设定位 | 对象模块：TMR3、TMR2 0：解除模块停止状态 1：转移到模块停止状态 | R/W |
| b5 | MSTPA5 | 8 位定时器 1、定时器 0（单元 0）的模块停止设定位 | 对象模块：TMR1、TMR0 0：解除模块停止状态 1：转移到模块停止状态 | R/W |
| b9-b6 | — | 保留位 | 读写值都为“1”。 | R/W |
| b10 | MSTPA10 | 可编程脉冲发生器 1（单元 1）的模块停止设定位 | 对象模块：PPG1 0：解除模块停止状态 1：转移到模块停止状态 | R/W |
| b11 | MSTPA11 | 可编程脉冲发生器 0（单元 0）的模块停止设定位 | 对象模块：PPG0 0：解除模块停止状态 1：转移到模块停止状态 | R/W |
| b12 | MSTPA12 | 16 位定时器脉冲单元 1（单元 1）的模块停止设定位 | 对象模块：TPU 单元 1（TPU6 ~ TPU11） 0：解除模块停止状态 1：转移到模块停止状态 | R/W |
| b13 | MSTPA13 | 16 位定时器脉冲单元 0（单元 0）的模块停止设定位 | 对象模块：TPU 单元 0（TPU0 ~ TPU5） 0：解除模块停止状态 1：转移到模块停止状态 | R/W |

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|---------|---------------------------|---|-----|
| b14 | MSTPA14 | 比较匹配定时器 1 (单元 1) 的模块停止设定位 | 对象模块: CMT 单元 1 (CMT2 ~ CMT3) 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b15 | MSTPA15 | 比较匹配定时器 0 (单元 0) 的模块停止设定位 | 对象模块: CMT 单元 0 (CMT0 ~ CMT1) 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b18-b16 | — | 保留位 | 读写值都为“1”。 | R/W |
| b19 | MSTPA19 | D/A 转换器的模块停止设定位 | 对象模块: DA 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b20 | MSTPA20 | A/D 转换器 (单元 3) 的模块停止设定位 | 对象模块: AD3 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b21 | MSTPA21 | A/D 转换器 (单元 2) 的模块停止设定位 | 对象模块: AD2 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b22 | MSTPA22 | A/D 转换器 (单元 1) 的模块停止设定位 | 对象模块: AD1 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b23 | MSTPA23 | A/D 转换器 (单元 0) 的模块停止设定位 | 对象模块: AD0 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b26-b24 | — | 保留位 | 读写值都为“1”。 | R/W |
| b27 | MSTPA27 | 数据传送控制器的模块停止设定位 | 对象模块: DTC 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b28 | MSTPA28 | DMA 控制器的模块停止设定位 | 对象模块: DMAC 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b30-b29 | — | 保留位 | 读写值都为“1”。 | R/W |
| b31 | ACSE | 全模块时钟停止模式允许位 | 0: 禁止全模块时钟停止模式 1: 允许全模块时钟停止模式 | R/W |

MSTPCRA 寄存器是控制模块停止状态的寄存器。

ACSE 位 (全模块时钟停止模式允许位)

如果在将 MSTPCRA 寄存器和 MSTPCRB 寄存器控制的全部模块 (注 1) 设定为模块停止状态后 CPU 执行 WAIT 指令, 就在停止总线控制部和 I/O 端口的运行后, 设定允许或者禁止降低消耗电流的全模块时钟停止模式。

注 1. 能通过设定 MSTPA5 位和 MSTPA4 位, 选择 8 位定时器的运行或者停止。

8.2.3 模块停止控制寄存器 B (MSTPCRB)

地址 0008 0014h

| | | | | | | | | | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|-----|---------|-----|---------|---------|-----|-----|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| MSTPB31 | MSTPB30 | MSTPB29 | MSTPB28 | MSTPB27 | MSTPB26 | MSTPB25 | — | MSTPB23 | — | MSTPB21 | MSTPB20 | — | — | — | — |
| 复位后的值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 复位后的值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| 位 | 符号 | 位名 | 功能 | R/W |
|--------|---------|---|--|-----|
| b19-b0 | — | 保留位 | 读写值都为“1”。 | R/W |
| b20 | MSTPB20 | I ² C 总线接口 1 (单元 1) 的模块停止设定位 | 对象模块: RIIC1 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b21 | MSTPB21 | I ² C 总线接口 0 (单元 0) 的模块停止设定位 | 对象模块: RIIC0 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b22 | — | 保留位 | 读写值都为“1”。 | R/W |
| b23 | MSTPB23 | CRC 运算器的模块停止设定位 | 对象模块: CRC 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b24 | — | 保留位 | 读写值都为“1”。 | R/W |
| b25 | MSTPB25 | 串行通信接口 6 的模块停止设定位 | 对象模块: SCI6 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b26 | MSTPB26 | 串行通信接口 5 的模块停止设定位 | 对象模块: SCI5 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b27 | MSTPB27 | 串行通信接口 4 的模块停止设定位 | 对象模块: SCI4 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b28 | MSTPB28 | 串行通信接口 3 的模块停止设定位 | 对象模块: SCI3 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b29 | MSTPB29 | 串行通信接口 2 的模块停止设定位 | 对象模块: SCI2 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b30 | MSTPB30 | 串行通信接口 1 的模块停止设定位 | 对象模块: SCI1 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |
| b31 | MSTPB31 | 串行通信接口 0 的模块停止设定位 | 对象模块: SCI0 0: 解除模块停止状态 1: 转移到模块停止状态 | R/W |

MSTPCRB 寄存器是控制模块停止状态的寄存器。

8.2.4 模块停止控制寄存器 C (MSTPCRC)

地址 0008 0018h

| | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------|-------|
| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 复位后的值 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | MSTPC | MSTPC |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|----------------|---------------|--|-----|
| b0 | MSTPC0 (注1) | RAM0 的模块停止设定位 | 对象模块: RAM0 (0000 0000h ~ 0000 FFFFh) 0: RAM0 运行 1: RAM0 停止 | R/W |
| b1 | MSTPC1 (注1) | RAM1 的模块停止设定位 | 对象模块: RAM1 (0001 0000h ~ 0001 FFFFh) 0: RAM1 运行 1: RAM1 停止 | R/W |
| b15-b2 | — | 保留位 | 读写值都为“0”。 | R/W |
| b31-b16 | — | 保留位 | 读写值都为“1”。 | R/W |

注 1. 不能在存取内部 RAM 时将对应的 MSTPC1 位和 MSTPC0 位置“1”，而且不能在 MSTPC1 位和 MSTPC0 位为“1”的状态下存取对应的内部 RAM。

MSTPCRC 寄存器是控制模块停止状态的寄存器。

8.2.5 深度待机控制寄存器 (DPSBYCR)

地址 0008 C280h

| | | | | | | | | |
|-------|-------|--------|-------------|-------------|----|----|----|-------------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | DPSBY | IOKEEP | RAMCUT 2 | RAMCUT 1 | — | — | — | RAMCUT 0 |
| 复位后的值 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|---------------|---|-----|
| b0 | RAMCUT0 | 内部 RAM 断开 0 位 | b5 b4 b0 0 0 0: 在深度软件待机模式中, 给内部 RAM (RAM0) (注1) 供电。 1 1 1: 在深度软件待机模式中, 不给内部 RAM (RAM0) (注1) 供电。 上述以外: 不能设定 | R/W |
| b3-b1 | — | 保留位 | 读写值都为“0”。 | R/W |
| b4 | RAMCUT1 | 内部 RAM 断开 1 位 | 请参照 RAMCUT0 位的功能。 | R/W |
| b5 | RAMCUT2 | 内部 RAM 断开 2 位 | 请参照 RAMCUT0 位的功能。 | R/W |
| b6 | IOKEEP | I/O 端口保持位 | 0: 在解除深度软件待机模式的同时, 解除 I/O 端口的保持状态。 1: 如果在解除深度软件待机模式后给 IOKEEP 位写“0”, 就解除 I/O 端口的保持状态。 | R/W |
| b7 | DPSBY | 深度软件待机位 | SSBY b7 0 0: 在执行 WAIT 指令后转移到睡眠模式。 0 1: 在执行 WAIT 指令后转移到睡眠模式。 1 0: 在执行 WAIT 指令后转移到软件待机模式。 1 1: 在执行 WAIT 指令后转移到深度软件待机模式。 | R/W |

注 1. 有关内部 RAM 的地址空间, 请参照表 8.2。

DPSBYCR 寄存器是控制深度软件待机模式的寄存器。

通过 RES# 引脚的复位信号而不通过解除深度软件待机模式的内部复位信号, 对 DPSBYCR 寄存器进行初始化。

RAMCUTj 位 (内部 RAM 断开 j 位) (j=0 ~ 2)

在深度软件待机模式中, 此位控制给内部 RAM 提供的内部电源。

内部 RAM 的地址空间分为 RAM0 区和 RAM1 区。有关地址空间, 请参照表 8.2。

只能通过设定 RAMCUT0 位、RAMCUT1 位和 RAMCUT2 位对 RAM0 的内部电源进行控制。

RAM1 的内部电源与 RAMCUT0 位、RAMCUT1 位和 RAMCUT2 位的设定无关, 在深度软件待机模式中停止工作。

IOKEEP 位 (I/O 端口保持位)

在深度软件待机模式中, I/O 端口保持和软件待机模式相同的状态。通过 IOKEEP 位选择在解除深度软件待机模式后是继续保持还是解除在深度软件待机模式中保持的 I/O 端口状态。

DPSBY 位（深度软件待机位）

此位控制深度软件待机模式的转移。

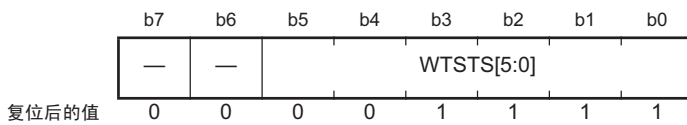
如果在 SBYCR.SSBY 位和 DPSBY 位都为“1”状态下执行 WAIT 指令，就经由软件待机模式转移到深度软件待机模式。如果通过外部中断引脚解除深度软件待机模式，DPSBY 位就保持“1”。要将此位置“0”时，必须写“0”。

在看门狗定时器模式中使用 WDT 时，此位的设定值无效。

此时，即使在 SBYCR.SSBY 位和 DPSBY 位都为“1”的状态下，也总是在执行 WAIT 指令后转移到睡眠模式或者全模块时钟停止模式。

8.2.6 深度待机等待控制寄存器（DPSWCR）

地址 0008 C281h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------------|----------------|---|-----|
| b5-b0 | WTSTS[5:0] | 深度软件待机的等待时间设定位 | b5 b0 0 0 0 1 0 1: 待机时间 =64 个状态 0 0 0 1 1 0: 待机时间 =512 个状态 0 0 0 1 1 1: 待机时间 =1024 个状态 0 0 1 0 0 0: 待机时间 =2048 个状态 0 0 1 0 0 1: 待机时间 =4096 个状态 0 0 1 0 1 0: 待机时间 =16384 个状态 0 0 1 0 1 1: 待机时间 =32768 个状态 0 0 1 1 0 0: 待机时间 =65536 个状态 0 0 1 1 0 1: 待机时间 =131072 个状态 0 0 1 1 1 0: 待机时间 =262144 个状态 0 0 1 1 1 1: 待机时间 =524288 个状态 | R/W |
| b7-b6 | — | 保留位 | 读写值都为“0”。 | R/W |

在通过外部中断引脚解除深度软件待机模式时，通过 DPSWCR 寄存器选择时钟稳定前的 LSI 待机时间。

通过 RES# 引脚的复位信号而不通过解除深度软件待机模式的内部复位信号，对 DPSWCR 寄存器进行初始化。

WTSTS[5:0] 位（深度软件待机的等待时间设定位）

在通过外部中断引脚解除深度软件待机模式时，此位选择时钟稳定前的 LSI 待机时间。如果使用深度软件待机模式，就必须在转移到深度软件待机模式前设定 WTSTS[5:0] 位。

在使用晶体振荡的情况下，必须参照表 8.5 并且根据工作频率选择大于等于振荡稳定时间的待机时间。在使用外部时钟的情况下，也需要 PLL 电路的稳定时间，请参照表 8.5 设定待机时间。

在振荡稳定期间，以 EXTAL 输入时钟的频率进行计数。

8.2.7 深度待机中断允许寄存器 (DPSIER)

地址 0008 C282h

| | | | | | | | |
|-------|----|----|----|--------|--------|--------|--------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| DNMIE | — | — | — | DIRQ3E | DIRQ2E | DIRQ1E | DIRQ0E |

复位后的值 0 0 0 0 0 0 0 0

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|--------|------------|--|-------------|
| b0 | DIRQ0E | IRQ0 引脚允许位 | 0: 禁止通过 IRQ0 引脚解除深度软件待机模式 1: 允许通过 IRQ0 引脚解除深度软件待机模式 | R/W |
| b1 | DIRQ1E | IRQ1 引脚允许位 | 0: 禁止通过 IRQ1 引脚解除深度软件待机模式 1: 允许通过 IRQ1 引脚解除深度软件待机模式 | R/W |
| b2 | DIRQ2E | IRQ2 引脚允许位 | 0: 禁止通过 IRQ2 引脚解除深度软件待机模式 1: 允许通过 IRQ2 引脚解除深度软件待机模式 | R/W |
| b3 | DIRQ3E | IRQ3 引脚允许位 | 0: 禁止通过 IRQ3 引脚解除深度软件待机模式 1: 允许通过 IRQ3 引脚解除深度软件待机模式 | R/W |
| b6-b4 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | DNMIE | NMI 引脚允许位 | 0: 禁止通过 NMI 引脚解除深度软件待机模式 1: 允许通过 NMI 引脚解除深度软件待机模式 | R/W (注1) |

注1. 只能写1次“1”，以后的写操作无效。

DPSIER 寄存器选择允许或者禁止作为深度软件待机模式解除源的外部中断引脚。

通过 RES# 引脚的复位信号而不通过深度软件待机模式解除源的内部复位信号，对 DPSIER 寄存器进行初始化。

8.2.8 深度待机中断标志寄存器 (DPSIFR)

地址 0008 C283h

| | | | | | | | | |
|-------|-------|----|----|----|--------|--------|--------|--------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | DNMIF | — | — | — | DIRQ3F | DIRQ2F | DIRQ1F | DIRQ0F |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|--------|---------------|---|--------------|
| b0 | DIRQ0F | IRQ0 深度待机解除标志 | 0: 未发生由 IRQ0 引脚引起的解除请求 1: 发生由 IRQ0 引脚引起的解除请求 | R(W) (注1) |
| b1 | DIRQ1F | IRQ1 深度待机解除标志 | 0: 未发生由 IRQ1 引脚引起的解除请求 1: 发生由 IRQ1 引脚引起的解除请求 | R(W) (注1) |
| b2 | DIRQ2F | IRQ2 深度待机解除标志 | 0: 未发生由 IRQ2 引脚引起的解除请求 1: 发生由 IRQ2 引脚引起的解除请求 | R(W) (注1) |
| b3 | DIRQ3F | IRQ3 深度待机解除标志 | 0: 未发生由 IRQ3 引脚引起的解除请求 1: 发生由 IRQ3 引脚引起的解除请求 | R(W) (注1) |
| b6-b4 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | DNMIF | NMI 深度待机解除标志 | 0: 未发生由 NMI 引脚引起的解除请求 1: 发生由 NMI 引脚引起的解除请求 | R(W) (注1) |

注1. 只能写“0”。

DPSIFR 寄存器是保持深度软件待机模式的解除请求的寄存器。

如果发生深度待机中断沿寄存器 (DPSIEGR) 设定的解除请求, DPSIFR 寄存器就变为“1”。即使在非深度软件待机模式的状态下发生解除请求, DPSIFR 寄存器也变为“1”, 因此必须在将 DPSIFR 寄存器置“00h”后转移到深度软件待机模式。而且, 有可能因更改 P3.ICR 寄存器和 DPSIER 寄存器的设定而使 DPSIFR 寄存器变为“1”。如果在更改 P3.ICR 寄存器和 DPSIER 寄存器的设定后将 DPSIFR 寄存器置“00h”, 就必须在至少经过 6 个 PCLK 周期并且读 DPSIFR 寄存器后, 给此寄存器写“0”。例如, 能通过读 DPSIER 寄存器至少确保 6 个 PCLK 周期。

通过 RES# 引脚的复位信号而不通过深度软件待机模式解除源的内部复位信号, 对 DPSIFR 寄存器进行初始化。

DIRQnF 标志 (IRQn 深度待机解除标志) (n=0 ~ 3)

此标志表示发生由 IRQn 引脚引起的解除请求。

[为“1”的条件]

- 发生由 DPSIEGR 寄存器选择的 IRQn 引脚引起的解除请求时

[为“0”的条件]

- 读“1”后写“0”时

DNMIF 标志 (NMI 深度待机解除标志)

此标志表示发生由 NMI 引脚引起的解除请求。

[为“1”的条件]

- 发生由 DPSIEGR 寄存器设定的 NMI 引脚引起的解除请求时

[为“0”的条件]

- 读“1”后写“0”时

8.2.9 深度待机中断边沿寄存器 (DPSIEGR)

地址 0008 C284h

| | | | | | | | |
|------------|----|----|----|-------------|-------------|-------------|-------------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| DNMI EG | — | — | — | DIRQ3 EG | DIRQ2 EG | DIRQ1 EG | DIRQ0 EG |

复位后的值 0 0 0 0 0 0 0 0

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|------------|--------------------------------|-----|
| b0 | DIRQ0EG | IRQ0 边沿选择位 | 0: 在下降沿产生解除请求 1: 在上升沿产生解除请求 | R/W |
| b1 | DIRQ1EG | IRQ1 边沿选择位 | 0: 在下降沿产生解除请求 1: 在上升沿产生解除请求 | R/W |
| b2 | DIRQ2EG | IRQ2 边沿选择位 | 0: 在下降沿产生解除请求 1: 在上升沿产生解除请求 | R/W |
| b3 | DIRQ3EG | IRQ3 边沿选择位 | 0: 在下降沿产生解除请求 1: 在上升沿产生解除请求 | R/W |
| b6-b4 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | DNMIEG | NMI 边沿选择位 | 0: 在下降沿产生解除请求 1: 在上升沿产生解除请求 | R/W |

DPSIEGR 寄存器选择用于解除深度软件待机模式的解除信号的边沿。

通过 RES# 引脚的复位信号而不通过深度软件待机模式解除源的内部复位信号，对 DPSIEGR 寄存器进行初始化。

8.2.10 复位状态寄存器 (RSTSR)

地址 0008 C285h

| | | | | | | | | |
|-------|-------------|----|----|----|----|----|----|----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | DPSR STF | — | — | — | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|------------|---|--------------|
| b6-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | DPSRSTF | 深度软件待机复位标志 | 0: 未产生由外部中断引起的深度软件待机模式的解除请求 1: 产生由外部中断引起的深度软件待机模式的解除请求 | R(W) (注1) |

注1. 只能写“0”。

RSTSR 寄存器是表示内部复位发生源的寄存器。

DPSRSTF 标志 (深度软件待机复位标志)

此标志表示通过 DPSIER 寄存器和 DPSIEGR 寄存器设定的外部中断源，解除了深度软件待机模式并且产生了内部复位。

通过 RES# 引脚的复位信号而不通过解除深度软件待机模式的内部复位信号，对 DPSRSTF 标志进行初始化。

[为“1”的条件]

- 通过外部中断解除了深度软件待机模式时

[为“0”的条件]

- 读“1”后写“0”时

8.2.11 深度待机备用寄存器 (DPSBKRY) (y=0 ~ 31)

地址 0008 C290h ~ 0008 C2AFh

| | | | | | | | | |
|-------|------------|------------|------------|------------|------------|------------|------------|------------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | BKUPm 7 | BKUPm 6 | BKUPm 5 | BKUPm 4 | BKUPm 3 | BKUPm 2 | BKUPm 1 | BKUPm 0 |
| 复位后的值 | x | x | x | x | x | x | x | x |

DPSBKRY 寄存器是能通过深度软件待机模式保存数据的 32 字节可读写寄存器。

即使在不保持内部 RAM 数据的深度软件待机模式中，也保持此寄存器的值。

不对 DPSBKRY 寄存器进行初始化，刚接通电源后的寄存器的值为不定值。

8.3 多时钟功能

如果设定 SCKCR.ICK[3:0] 位、SCKCR.BCK[3:0] 位和 SCKCR.PCK[3:0] 位，时钟频率就随之改变。

CPU 和总线主控通过 ICK[3:0] 位设定的运行时钟工作，外围模块通过 PCK[3:0] 位设定的运行时钟工作，外部总线时钟通过 BCK[3:0] 位设定的运行时钟工作，详细内容请参照“7. 时钟发生电路”。

8.4 模块停止功能

能以内部外围模块为单位设定模块停止功能。

如果将 MSTPCRA ~ MSTPCRC 寄存器对应的 MSTPyj 位 (y=A ~ C, j=0 ~ 31) 置“1”，模块就停止运行并且转移到模块停止状态。此时，CPU 继续单独运行。通过将对应的 MSTPyj 位置“0”，解除模块停止状态并且模块在总线周期结束时重新开始运行。

在模块停止状态下，保持模块的内部状态。

复位后，DMAC、DTC 和内部 RAM 以外的全部模块处于模块停止状态。不能读写被设定为模块停止状态的模块的寄存器。

8.5 低功耗状态

8.5.1 睡眠模式

8.5.1.1 向睡眠模式的转移

如果在 SBYCR.SSBY 位为“0”的状态下执行 WAIT 指令，CPU 就进入睡眠模式。

CPU 在睡眠模式中停止运行，但是保持 CPU 内部寄存器的值。是 CPU 以外的外围功能不停止运行。

8.5.1.2 睡眠模式的解除

通过全部中断、RES# 引脚的复位或者看门狗定时器的上溢复位来解除睡眠模式。

- 通过中断进行的解除

如果发生中断，睡眠模式就被解除并且开始中断异常处理。如果可屏蔽中断被 CPU 屏蔽（中断优先级（注1）小于等于 CPU 的 PSW.IPL[2:0] 位（注2）的优先级），就不解除睡眠模式。

注 1. 详细内容请参照“10. 中断控制器 (ICU)”。

注 2. 详细内容请参照“2. CPU”。

- 通过 RES# 引脚进行的解除

如果将 RES# 引脚置为 Low 电平，就进入复位状态。如果在经过规定的复位输入期间后将 RES# 引脚置为 High 电平，CPU 就开始复位异常处理。

- 通过看门狗定时器上溢复位进行的解除

通过看门狗定时器上溢的内部复位来解除睡眠模式。

8.5.2 全模块时钟停止模式

8.5.2.1 向全模块时钟停止模式的转移

在 SBYCR.SSBY 位置“0”的状态下执行 WAIT 指令时，如果以下 2 个条件成立，就在总线周期结束时向全模块时钟停止模式转移（注1）。

- MSTPCRA.ACSE 位为“1”。
- 在由 MSTPCRA 寄存器和 MSTPCRB 寄存器控制的全部模块中，将 8 位定时器（单元 0 和单元 1）以外的全部模块设定为模块停止状态（MSTPCRA=FFFFFF[C~F]Fh，MSTPCRB=FFFFFFFh）。

在全模块时钟停止模式中，除 8 位定时器（注2）和看门狗定时器以外的全部外围模块、CPU、总线控制器和 I/O 端口停止运行。

如果在全模块时钟停止模式中需要进一步降低消耗电流，就必须将由 MSTPCRC 寄存器控制的模块设定为模块停止状态。

在使用全模块时钟停止模式时，必须在进行以下的设定后执行 WAIT 指令：

1. 将 CPU 的 PSW.I 位（注3）置“0”。
2. 将用于从全模块时钟停止模式返回的中断的优先级（注4）设定为高于 CPU 的 PSW.IPL[2:0] 位的优先级（注3）。
3. 将用于从全模块时钟停止模式返回的中断的 IERi.IENj 位（注4）置“1”。
4. 对不用于从全模块时钟停止模式返回的中断进行以下的任何一种设定：
 - 将不用于全模块时钟停止模式返回的中断（注5）的优先级设定为小于等于 CPU 的 PSW.IPL[2:0] 位（注3）的优先级（注4）。
 - 将不用于全模块时钟停止模式返回的中断（注5）的 IERi.IENj 位（注4）置“0”。
5. 执行 WAIT 指令（通过执行 WAIT 指令，CPU.PSW.I 位（注3）自动变为“1”）。

注 1. 有时因 DTC 和 DMAC 的运行状态而无法转移到全模块时钟停止模式。因此，必须在将 MSTPCRA.MSTPA28 位和 MSTPCRA.MSTPA27 位置“1”前，将 DMAC 的 DMSCNT.DMST 位和 DTC 的 DTCST.DTCST 位置“0”，并且在启动 DTC 和 DMAC 的状态下进行转移。

注 2. 能通过 MSTPCRA.MSTPA5 位和 MSTPCRA.MSTPA4 位选择运行或者停止。

注 3. 详细内容请参照“2. CPU”。

注 4. 详细内容请参照“10. 中断控制器（ICU）”。

注 5. 如果在外围功能运行的状态下执行 WAIT 指令，就有可能通过不能作为返回源的中断进行返回。因此除了能作为返回源的中断以外，还包括能通过 IERi.IENj 位和 PSW.IPL[2:0] 位设定的全部中断。

8.5.2.2 全模块时钟停止模式的解除

在通过外部中断引脚（NMI 引脚和 IRQ0 ~ IRQ15 引脚）、RES# 引脚或者内部中断（8 位定时器（注1）、看门狗定时器）解除全模块时钟停止模式后，经过异常处理状态转移到正常的程序执行状态。如果可屏蔽中断被 CPU 屏蔽（中断优先级（注2）小于等于 CPU 的 PSW.IPL[2:0] 位（注3）的优先级），或者被设定为 DTC、DMAC 的启动源，就不解除全模块时钟停止模式。

注 1. 能通过 MSTPCRA.MSTPA5 位和 MSTPCRA.MSTPA4 位选择运行或者停止。

注 2. 详细内容请参照“10. 中断控制器（ICU）”。

注 3. 详细内容请参照“2. CPU”。

8.5.3 软件待机模式

8.5.3.1 向软件待机模式的转移

如果在将 SBYCR.SSBY 位置“1”，将 DPSBYCR.DPSBY 位置“0”的状态下执行 WAIT 指令，就转移到软件待机模式。在此模式中，CPU、内部外围功能和振荡器的全部功能停止运行，但是保持 CPU 内部寄存器的值、内部 RAM 的数据、内部外围功能和 I/O 端口状态。能通过 SBYCR.OPE 位选择是将地址总线和总线控制信号设定为高阻抗状态还是保持为输出状态。因为振荡器停止振荡，所以功耗明显降低。

在向软件待机模式转移时，系统时钟（ICLK）和外围模块时钟（PCLK）的工作频率必须相同。如果要在不同工作频率的状态下向软件待机模式转移，就必须在执行 WAIT 指令前更改系统时钟或者外围模块时钟的设定。

另外，必须在执行 WAIT 指令前将 DMAC 的 DMSCNT.DMST 位和 DTC 的 DTCST.DTCST 位置“0”。在看门狗定时器模式中使用 WDT 时，不能转移到软件待机模式，必须在执行 WAIT 指令前停止 WDT。在使用软件待机模式时，必须在进行以下的设定后执行 WAIT 指令：

1. 将 CPU 的 PSW.I 位（注1）置“0”。
2. 将用于从软件待机模式返回的中断的优先级（注2）设定为高于 CPU.PSW.IPL[2:0] 位（注1）的优先级。
3. 将用于从软件待机模式返回的中断的 IERi.IENj 位（注2）置“1”。
4. 对不用于从软件待机模式返回的中断进行以下的任何一种设定：
 - 将不用于从软件待机模式返回的中断（注3）的优先级（注2）设定为小于等于 CPU.PSW.IPL[2:0] 位（注1）的优先级。
 - 将不用于从软件待机模式返回的中断（注3）的 IERi.IENj 位（注2）置“0”。
5. 执行 WAIT 指令（通过执行 WAIT 指令，CPU.PSW.I 位（注1）自动变为“1”）。

注 1. 详细内容请参照“2. CPU”。

注 2. 详细内容请参照“10. 中断控制器（ICU）”。

注 3. 如果在外围功能运行的状态下执行 WAIT 指令，就有可能通过不能作为返回源的中断进行返回。而且，在软件待机模式中，因为未被设定为返回源的 IRQ0 ~ IRQ15 中断的中断状态标志有可能被置位，所以可能在返回后开始不同于返回源的中断异常处理。因此除了能作为返回源的中断以外，还包括能通过 IERi.IENj 位和 PSW.IPL[2:0] 位设定的全部中断。

8.5.3.2 软件待机模式的解除

通过外部中断引脚（NMI 引脚和 IRQ0 ~ IRQ15 引脚（注1））或者 RES# 引脚的复位来解除软件待机模式。

1. 通过中断进行的解除

如果输入 NMI 中断、IRQ0 ~ IRQ15（注1）的中断请求信号，时钟就开始振荡，在经过 SBYCR.STS[4:0] 位选择的时间后给整个 LSI 提供稳定的时钟，然后解除软件待机模式，开始中断异常处理。

在通过 IRQ0 ~ IRQ15（注1）中断解除软件待机模式时，必须将对应的允许位（IERi.IENj 位（注2））置“1”并且不能发生优先级高于 IRQ0 ~ IRQ15（注1）中断优先级的中断。

在通过设定为边沿检测的 IRQ0 ~ IRQ15（注1）中断解除软件待机模式时，必须在作为解除源的中断异常处理程序的起始位置，将该中断的状态标志（ICU.IRi.IR）置“0”。

在软件待机模式中，因为不作为软件待机模式解除源的 IRQ0 ~ IRQ15 中断的中断状态标志有可能被置位（注3），所以必须在从软件待机模式返回后清除 IR 标志。

注 1. 当 ICU 的 SSIER.SSIj 位为“0”时，不能通过对应的 IRQ0 ~ IRQ15 中断解除软件待机模式，详细内容请参照“10. 中断控制器（ICU）”。

注 2. 详细内容请参照“10. 中断控制器（ICU）”。

注 3. 详细内容请参照“10.6.2 从软件待机模式的返回”。

2. 通过 RES# 引脚进行的解除

如果将 RES# 引脚置为 Low 电平，时钟就开始振荡，同时给 LSI 提供时钟。此时，RES# 引脚必须保持 Low 电平直到时钟振荡稳定为止。如果将 RES# 引脚置为 High 电平，CPU 就开始复位异常处理。

8.5.3.3 解除软件待机模式后的振荡稳定时间的设定

必须通过 SBYCR.STS[4:0] 位进行以下的设定：

1. 使用晶体振荡的情况
设定 STS[4:0] 位，使待机时间大于等于振荡稳定时间。
工作频率、STS[4:0] 位的设定与待机时间的对应如表 8.4 所示。
2. 使用外部时钟的情况
需要 PLL 电路的稳定时间，请参照表 8.4 设定待机时间。

表 8.4 振荡稳定时间的设定

| STS4 | STS3 | STS2 | STS1 | STS0 | 待机时间 (状态) | PCLK (注1) (MHz) | | | 单位 |
|------|------|------|------|--------|--------------|-----------------|-------|-------|----|
| | | | | | | 50 | 25 | 8 | |
| 0 | 0 | 0 | 0 | 0 | (保留) | — | — | — | μs |
| | | | | 1 | (保留) | — | — | — | |
| | | | 1 | 0 | (保留) | — | — | — | |
| | | | | 1 | (保留) | — | — | — | |
| | | 1 | 0 | 0 | (保留) | — | — | — | |
| | | | | 1 | 64 | 1.3 | 2.6 | 8.0 | |
| | | | | 1 | 512 | 10.25 | 20.5 | 64.0 | |
| | | | 1 | 0 | 1024 | 20.5 | 41.0 | 128.0 | |
| | 1 | | | 0 | 2048 | 40.95 | 81.9 | 256.0 | |
| | | | | 1 | 4096 | 0.08 | 0.16 | 0.51 | |
| | 1 | 0 | 0 | 0 | 16384 | 0.33 | 0.66 | 2.05 | ms |
| | | | | 1 | 32768 | 0.655 | 1.31 | 4.10 | |
| | | | 1 | 0 | 65536 | 1.31 | 2.62 | 8.19 | |
| | | | | 1 | 131072 | 2.62 | 5.24 | 16.38 | |
| 1 | | 0 | 0 | 262144 | 5.25 | 10.49 | 32.77 | | |
| | | | 1 | 524288 | 10.49 | 20.97 | 65.54 | | |
| 1 | x | x | x | x | (保留) | — | — | — | |

■：使用外部时钟时的推荐设定时间

■：使用晶体振荡时的推荐设定时间

注 1. PCLK 是外围模块分频器的输出时钟。

因为振荡稳定等待时间包括振荡器的振荡尚未稳定的期间，所以受谐振器特性的影响。
上述数值是参考值。

8.5.3.4 软件待机模式的应用例子

在 IRQ 引脚的下降沿转移到软件待机模式并且在 IRQ 引脚的上升沿解除软件待机模式的运行例子如图 8.2 所示。

此例中，在 ICU 的 IRQCRn.IRQMD[1:0] 位为 “01b”（下降沿）的状态下接受 IRQ 中断后，先将 IRQMD[1:0] 位置 “10b”（上升沿），再将 ICU 的 SSIER.SSIj 位和 SBYCR.SSBY 位置 “1”，然后执行 WAIT 指令，转移到软件待机模式。

此后，在 IRQ 引脚的上升沿解除软件待机模式。

要从软件待机模式返回时，还需要设定中断控制器（ICU），详细内容请参照“10. 中断控制器（ICU）”。

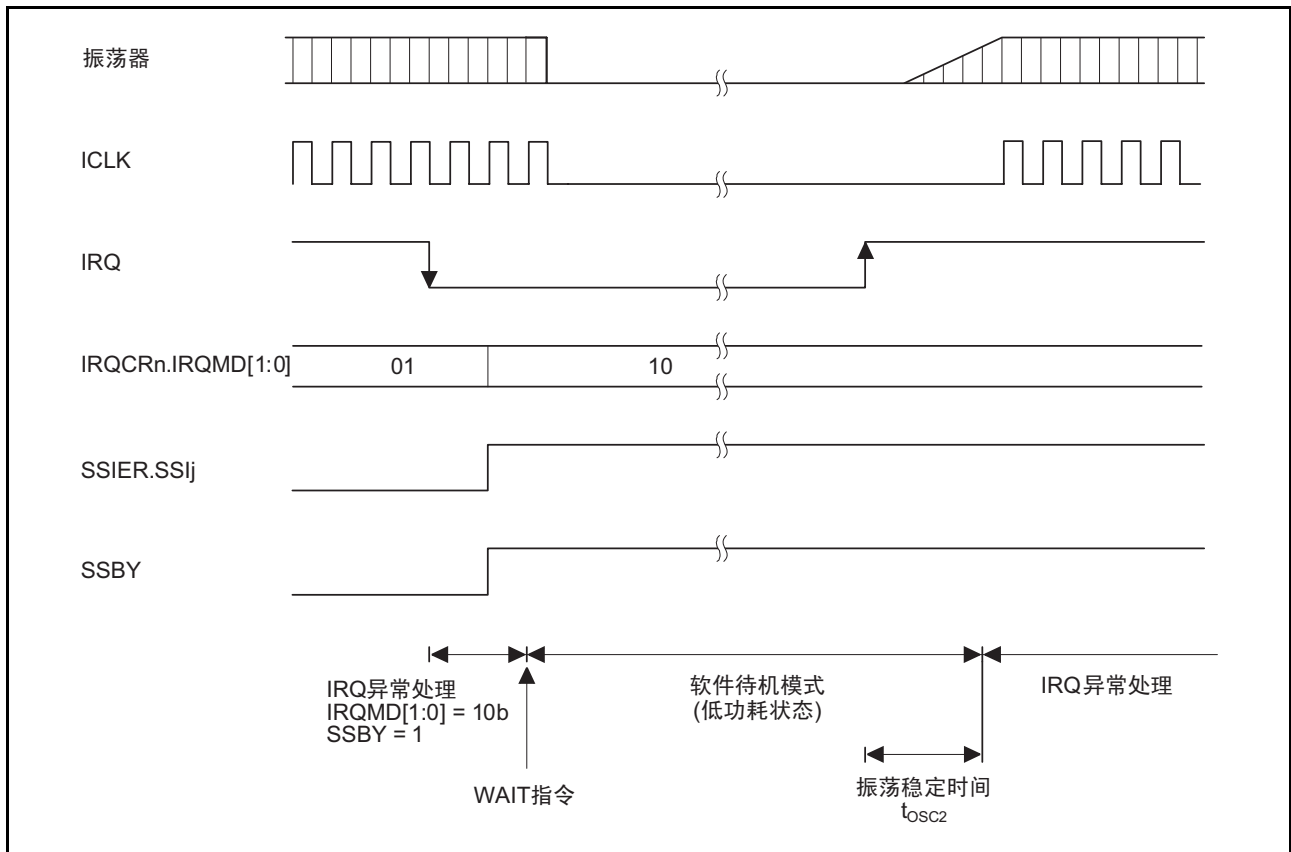


图 8.2 软件待机模式的应用例子

8.5.4 深度软件待机模式

8.5.4.1 向深度软件待机模式的转移

如果在 SBYCR.SSBY 位为“1”的状态下执行 WAIT 指令，就转移到软件待机模式（注1）。此时，如果 DPSBYCR.DPSBY 位为“1”，就转移到深度软件待机模式。如果在转移到软件待机模式时发生和软件待机模式解除请求（NMI 和 IRQ0 ~ IRQ15 的中断请求）的竞争，就解除软件待机模式，与 DPSBY 位的设定无关。在经过 SBYCR.STS[4:0] 位选择的软件待机模式的振荡稳定时间后，开始中断异常处理（注2）。

如果在 SSBY 位和 DPSBY 位都为“1”的状态下没有产生软件待机模式的解除请求，就在转移到软件待机模式后立即转移到深度软件待机模式。

在深度软件待机模式中，因为 CPU、内部外围功能、内部 RAM1（注3）和振荡器的全部功能停止运行，而且停止提供内部电源，所以功耗明显降低。此时，CPU 和内部外围功能的寄存器内容全部为不定值，内部 RAM1（注3）的数据与 DPSBYCR.RAMCUT2 ~ RAMCUT0 位的设定无关，也全部为不定值。

能通过将 RAMCUT2 ~ RAMCUT0 位全部置“0”，保持内部 RAM0（注3）的数据。在 RAMCUT2 ~ RAMCUT0 位全部被置“1”的情况下，因为停止给内部 RAM0（注3）提供内部电源，所以进一步降低了功耗。此时，内部 RAM0（注3）的数据为不定值。

端口保持软件待机模式中的状态。

- 注 1. 必须在执行 WAIT 指令前满足向软件待机模式转移时的 DTC、DMAC 和 WDT 的相关条件，详细内容请参照“8.5.3 软件待机模式”。
- 注 2. 在通过设定为边沿检测的 IRQ0 ~ IRQ15 中断解除软件待机模式时，必须在作为解除源的中断异常处理程序的起始位置，将该中断的状态标志（ICU.IRi.IR）置“0”。
- 注 3. 内部 RAM 的地址空间分为 RAM0 区和 RAM1 区，地址空间请参照表 8.2。

8.5.4.2 深度软件待机模式的解除

通过外部中断引脚（NMI 引脚和 IRQ0-A ~ IRQ3-A 引脚）或者 RES# 引脚的复位来解除深度软件待机模式。

1. 通过外部中断进行的解除

如果 DPSIFR.DNMIF 标志或者 DPSIFR.DIRQnF 标志（n=0~3）变为“1”，就解除深度软件待机模式。如果 DPSIER.DNMIE 位或者 DPSIER.DIRQnE 位（n=0~3）允许的 NMI 引脚或者 IRQ0-A ~ IRQ3-A 引脚产生边沿，DNMIF 标志或者 DIRQnF 标志就变为“1”。能通过 DPSIEGR 寄存器选择各引脚的上升沿和下降沿。

如果产生深度软件待机模式的解除请求，就在开始时钟振荡的同时开始提供内部电源，并且对整个 LSI 产生内部复位信号。在经过 DPSWCR.WTSTS[5:0] 位选择的待机时间后，给整个 LSI 提供稳定的时钟并且解除内部复位。如果在解除内部复位的同时解除深度软件待机模式，就开始复位异常处理。

如果通过外部中断解除深度软件待机模式，RSTSR.DPSRSTF 标志就变为“1”。

2. 通过 RES# 引脚进行的解除

如果将 RES# 引脚置为 Low 电平，时钟就开始振荡并且同时开始提供内部电源。在时钟开始振荡的同时，给 LSI 提供时钟。此时，RES# 引脚必须保持 Low 电平直到时钟振荡稳定为止。如果将 RES# 引脚置为 High 电平，CPU 就开始复位异常处理。

8.5.4.3 解除深度软件待机模式时的引脚状态

在深度软件待机模式中，I/O 端口保持软件待机模式中的状态。通过深度软件待机模式的内部复位对 LSI 内部进行初始化，如果解除深度软件待机模式，就立即开始复位异常处理。此时的端口状态如下所示。

能通过 DPSBYCR.IOKEEP 位选择是将 I/O 端口设定为初始状态还是继续保持软件待机模式中的 I/O 端口状态。

- 当 IOKEEP 位为 “0” 时
通过深度软件待机模式的内部复位对 I/O 端口进行初始化。
- 当 IOKEEP 位为 “1” 时
通过深度软件待机模式的内部复位对 LSI 内部进行初始化，与 LSI 的内部状态无关，I/O 端口继续保持软件待机模式中的状态。此时，即使设定 I/O 端口或者外围模块，I/O 端口也继续保持软件待机模式中的状态。然后，通过将 IOKEEP 位置 “0”，解除 I/O 端口的状态保持，并且根据内部状态运行。

不通过解除深度软件待机模式的内部复位对 IOKEEP 位进行初始化。

8.5.4.4 解除深度软件待机模式后的振荡稳定时间的设定

必须对 DPSWCR.WTSTS[5:0] 位进行以下的设定：

1. 使用晶体振荡的情况
设定 WTSTS[5:0] 位，使待机时间大于等于振荡稳定时间。
EXTAL 输入时钟的频率、WTSTS[5:0] 位的设定与待机时间的对应如表 8.5 所示。
2. 使用外部时钟的情况
需要 PLL 电路的稳定时间，请参照表 8.5 设定待机时间。

表 8.5 振荡稳定时间的设定

| WTSTS5 | WTSTS4 | WTSTS3 | WTSTS2 | WTSTS1 | WTSTS0 | 待机时间 (状态) | EXTAL 输入时钟的频率 (注1) (MHz) | | 单位 |
|--------|--------|--------|--------|--------|--------|--------------|-----------------------------|-------|----|
| | | | | | | | 12 | 8 | |
| 0 | 0 | 0 | 0 | 0 | 0 | (保留) | — | — | μs |
| | | | | | 1 | (保留) | — | — | |
| | | | | 1 | 0 | (保留) | — | — | |
| | | | | | 1 | (保留) | — | — | |
| | | | 1 | 0 | 0 | (保留) | — | — | |
| | | | | | 1 | 64 | 5.3 | 8.0 | |
| | | | | 1 | 0 | 512 | 42.7 | 64.0 | |
| | | | | | 1 | 1024 | 85.3 | 128.0 | |
| | | 1 | | | 0 | 2048 | 170.7 | 256.0 | |
| | | | | | 1 | 4096 | 0.34 | 0.51 | |
| | | 1 | 0 | 1 | 0 | 16384 | 1.37 | 2.05 | |
| | | | | | 1 | 32768 | 2.73 | 4.10 | |
| | | | | 1 | 0 | 65536 | 5.46 | 8.19 | |
| | | | | | 1 | 131072 | 10.92 | 16.38 | |
| 1 | 1 | | 0 | 262144 | 21.85 | 32.77 | | | |
| | | | 1 | 524288 | 43.69 | 65.54 | | | |
| 1 | 1 | x | x | x | x | (保留) | — | — | |
| 1 | x | x | x | x | x | (保留) | — | — | |

■：使用外部时钟时的推荐设定时间

■：使用晶体振荡时的推荐设定时间

注 1. 因为振荡稳定等待时间包括振荡器的振荡尚未稳定的期间，所以受谐振器特性的影响。
上述数值是参考值。

8.5.4.5 深度软件待机模式的应用例子

在 IRQ 引脚的下降沿转移到深度软件待机模式，并且在 IRQ 引脚的上升沿解除深度软件待机模式的运行例子如图 8.3 所示。

此例中，在 ICU 的 IRQCRn.IRQMD[1:0] 位为 “01b”（下降沿）的状态下接受 IRQ 中断后，先将 DPSIEGR.DIRQnEG 位置 “1”（上升沿），再将 SBYCR.SSBY 位和 DPSBYCR.DPSBY 位置 “1”，然后执行 WAIT 指令，转移到深度软件待机模式。

此后，在 IRQ 引脚的上升沿解除深度软件待机模式。

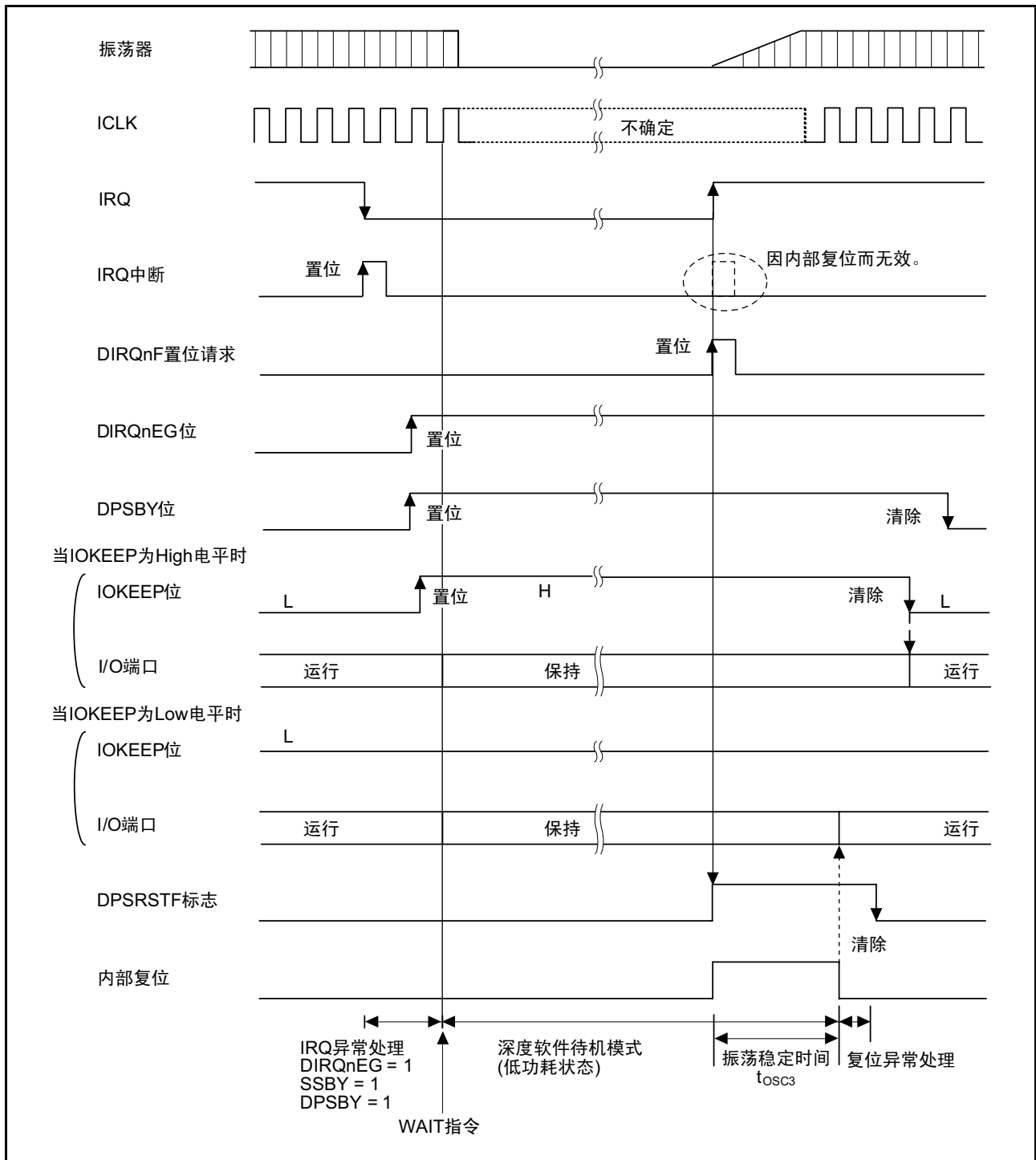


图 8.3 深度软件待机模式的应用例子

8.5.4.6 深度软件待机模式的流程图

使用深度软件待机模式时的流程图例子如图 8.4 所示。

此例中，在复位异常处理后，根据复位的 RSTSR.DPSRSTF 标志判断是通过 RES# 引脚进行的复位还是通过解除深度软件待机模式进行的复位。

在通过 RES# 引脚进行复位时，在进行各种设定后转移到深度软件待机模式。

在通过解除深度软件待机模式进行复位时，在设定 I/O 端口后将 DPSBYCR.IOKEEP 位置 “0”。

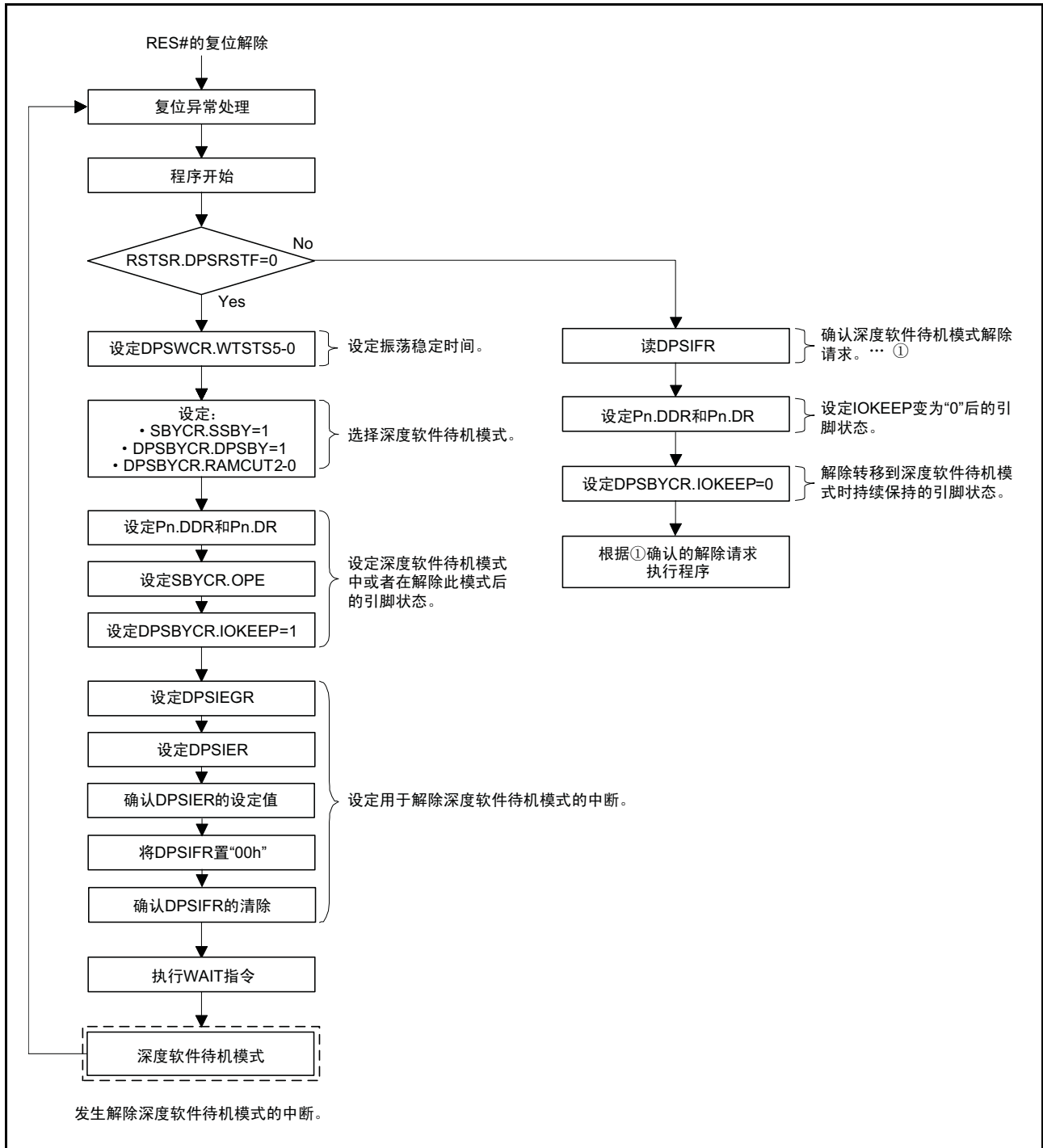


图 8.4 深度软件待机模式的流程图例子

8.6 BCLK 输出的控制

能通过 SCKCR.PSTOP1 位和对应的 P53 的 P5.DDR.B3 位控制 BCLK 输出。

如果将 PSTOP1 位置“0”，P53 就为 BCLK 输出；如果将 PSTOP1 位置“1”，就在总线周期结束时停止 BCLK 输出，BCLK 输出变为 High 电平。如果将 P53 的 P5.DDR.B3 位置“0”，BCLK 输出就被禁止，变为输入端口。

各运行模式的 BCLK 引脚状态如表 8.6 所示。

表 8.6 各运行模式的 BCLK 引脚（P53）状态

| 寄存器的设定值 | | 正常运行 状态 | 睡眠 | 全模块时钟 停止 | 软件待机 | | 深度软件待机 | |
|---------|--------|------------|---------|-------------|-------|-------|----------|----------|
| DDR | PSTOP1 | | | | OPE=0 | OPE=1 | IOKEEP=0 | IOKEEP=1 |
| 0 | x | Hi-Z | Hi-Z | Hi-Z | Hi-Z | Hi-Z | Hi-Z | Hi-Z |
| 1 | 0 | BCLK 输出 | BCLK 输出 | BCLK 输出 | High | High | High | High |
| 1 | 1 | High | High | High | High | High | High | High |

8.7 使用时的注意事项

8.7.1 I/O 端口状态

在软件待机模式和深度软件待机模式中保持 I/O 端口状态。如果输出 High 电平，就无法降低输出电流的消耗电流。

8.7.2 DMAC 和 DTC 的模块停止

必须在将 MSTPCRA.MSTPA28 和 MSTPCRA.MSTPA27 位置“1”前，将 DMAC 的 DMSCNT.DMST 位和 DTC 的 DTCST.DTCST 位置“0”，并且将 DTC 和 DMAC 设定为未启动状态。详细内容请参照“12. DMA 控制器（DMAC）”和“13. 数据传送控制器（DTC）”。

8.7.3 内部外围模块的中断

此中断不能在模块停止状态下运行。如果在产生中断请求的状态下停止模块，就不能清除 CPU 的中断源或者 DMAC、DTC 的启动源。必须事先禁止中断，然后进入模块停止状态。

8.7.4 MSTPCRA、MSTPCRB、MSTPCRC 寄存器的写操作

只能通过 CPU 写 MSTPCRA、MSTPCRB、MSTPCRC 寄存器。

8.7.5 通过 DIRQnE 位（n=0～3）控制输入缓冲器

能通过将 DPSIER.DIRQnE 位（n=0～3）位置“1”，将 P30/IRQ0-A～P33/IRQ3-A 引脚的输入缓冲器设定为有效。因此必须注意：这些引脚的输入反映到 DPSIFR.DIRQnF 标志（n=0～3）而不反映到中断控制器、外围模块和 I/O 端口等。

必须通过 Pm.ICR 寄存器控制中断控制器、外围模块和 I/O 端口等的输入。

8.7.6 深度软件待机模式的转移和中断的竞争

如果在向深度软件待机模式转移时发生和软件待机模式解除请求的竞争，就不向深度软件待机模式转移而开始软件待机模式的解除响应顺序。然后，在经过 SBYCR.STS[4:0] 位选择的软件待机模式的振荡稳定时间后，开始中断异常处理。

必须注意：在向深度软件待机模式的转移和 NMI 中断发生竞争时，需要执行 NMI 中断异常处理程序。

在相深度软件待机模式的转移和 IRQ0 ~ IRQ15 中断发生竞争时，能通过将 ICU 的 SSIER.SSIj 位 (j=0 ~ 15) 置“0”，转移到深度软件待机模式而不开始中断异常处理。

8.7.7 WAIT 指令的执行时序

不等到先执行的寄存器的写操作结束就执行 WAIT 指令。有可能在通过写寄存器进行的设定变更被反映前执行 WAIT 指令，而导致意想不到的运行。因此，必须在确认已经写完最后的寄存器后执行 WAIT 指令。

9. 异常处理

9.1 异常事件

在 CPU 正常执行程序的过程中，有可能因某个事件的发生而中止正在执行的程序并且需要执行其他程序。此类事件统称为异常事件。

RX CPU 对应 7 种异常，异常的种类如图 9.1 所示。

如果发生异常，处理器模式就转移到管理模式。

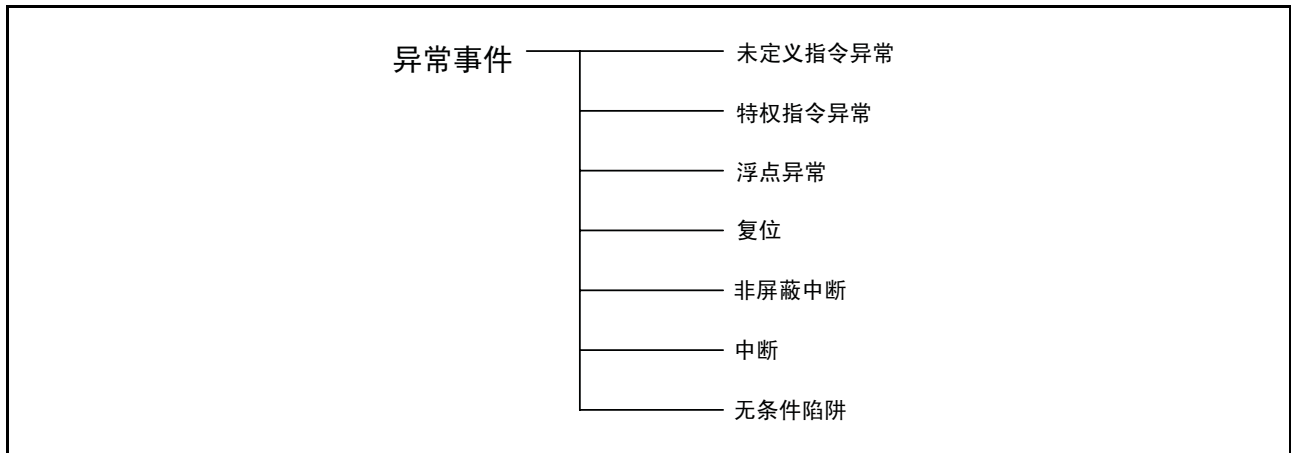


图 9.1 异常事件的种类

9.1.1 未定义指令异常

在检测到执行未定义指令（未安装的指令）时发生未定义指令异常。

9.1.2 特权指令异常

在用户模式中检测到已执行特权指令时发生特权指令异常。只能在管理模式中执行特权指令。

9.1.3 浮点异常

在检测到 IEEE754 规格规定的 5 种异常事件（上溢、下溢、精度异常、被零除、无效运算）以及非安装处理时发生浮点异常。当 FPSW 的 EX 位、EU 位、EZ 位、EO 位和 EV 位为“0”时，禁止浮点异常处理。

9.1.4 复位

在给 CPU 输入复位信号时发生复位。因为复位的优先级最高，所以随时被接受。

9.1.5 非屏蔽中断

在给 CPU 输入非屏蔽中断信号时发生非屏蔽中断。只在认为是对系统造成致命的故障时使用此中断。使用条件是必须在异常处理程序的处理后不返回到发生异常时正在执行的程序。

9.1.6 中断

在给 CPU 输入中断信号时发生中断。能将高速中断选择为优先级最高的中断。当 PSW 的 I 位为 “0” 时，禁止中断异常处理。

9.1.7 无条件陷阱

如果执行 INT 指令和 BRK 指令，就产生无条件陷阱。

9.2 异常处理步骤

异常处理包括硬件自动处理的部分以及通过用户记述的程序（异常处理程序）进行处理的部分。除复位以外，接受异常时的处理步骤如图 9.2 所示。

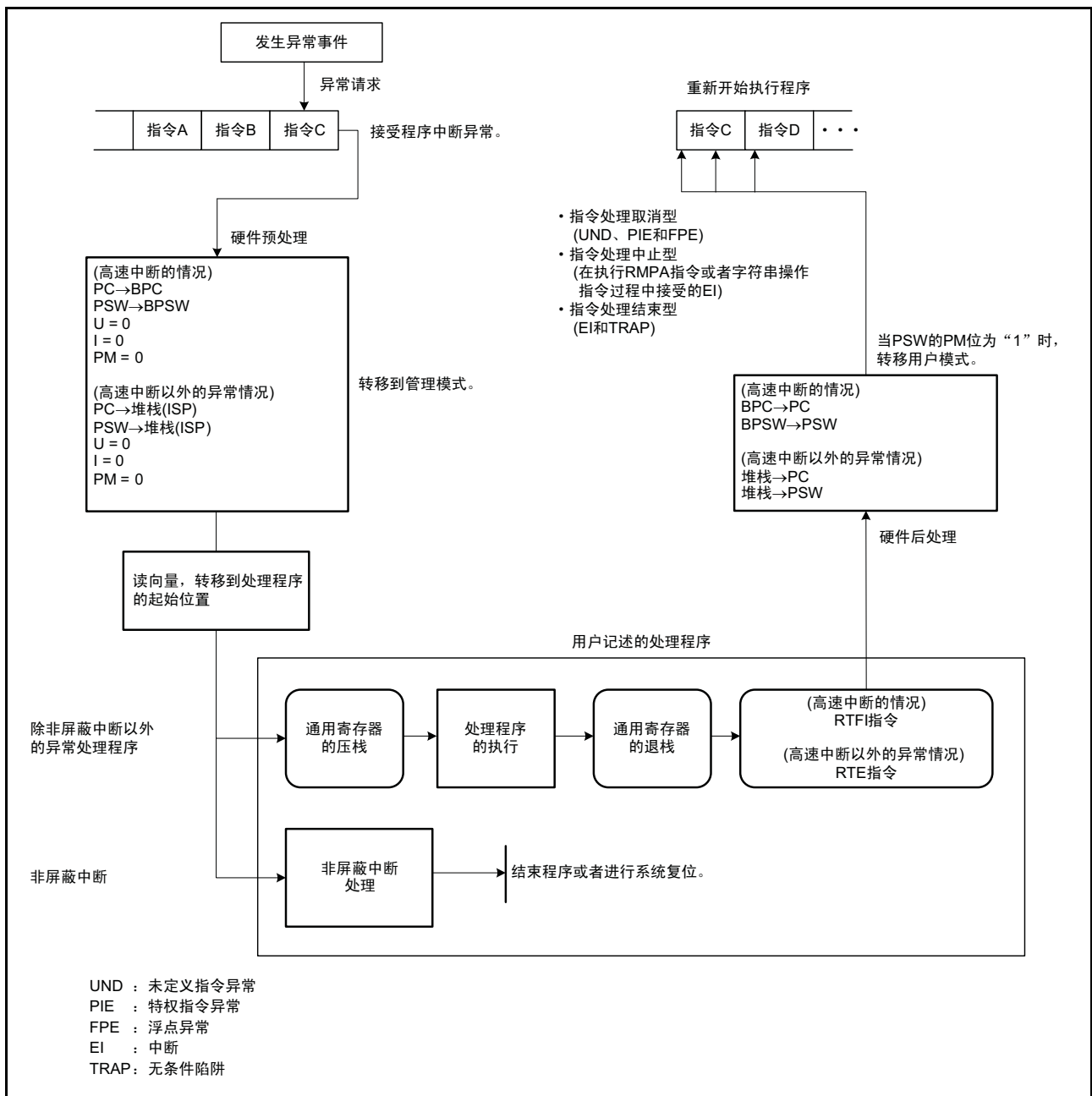


图 9.2 异常处理步骤的概要

一旦接受异常，RX CPU 就在硬件处理后，存取向量并且取得转移目标地址。按各异常给向量分配了向量地址，写异常处理程序的转移目标地址。

有关 RX CPU 的硬件预处理，在高速中断的情况下，将程序计数器（PC）的内容保存到备用 PC（BPC），处理器状态字（PSW）的内容保存到备用 PSW（BPSW）；在非高速中断异常的情况下，将 PC 和 PSW 压栈。

对于异常处理程序中使用的通用寄存器以及 PC 和 PSW 以外的控制寄存器，必须在异常处理程序的起始位置，通过用户程序将这些寄存器压栈。

在异常处理程序处理结束后，通过在恢复被压栈的寄存器后执行 RTE 指令，从异常处理返回到原来的程序。只在高速中断的情况下执行 RTFI 指令。但是，在非屏蔽中断的情况下，不返回到原来的程序而必须结束程序或者进行系统复位。

有关 RX CPU 的硬件后处理，在高速中断的情况下，将 BPC 的值恢复到 PC，BPSW 的值恢复到 PSW。在非高速中断异常的情况下，从堆栈恢复 PC 和 PSW 的值。

9.3 异常事件的接受

如果发生异常事件，就在中止目前执行的程序后转移到异常处理程序的处理。

9.3.1 接受时序和被保存的 PC 值

各异常事件的接受时序以及被保存的程序计数器（PC）的值如表 9.1 所示。

表 9.1 接受时序和被保存的 PC 值

| 异常事件 | | 处理型 | 接受时序 | 被保存到 BPC/ 堆栈的 PC 值 |
|---------|--|---------|----------|--------------------|
| 未定义指令异常 | | 指令处理取消型 | 正在执行指令 | 发生异常的指令的 PC 值 |
| 特权指令异常 | | 指令处理取消型 | 正在执行指令 | 发生异常的指令的 PC 值 |
| 浮点异常 | | 指令处理取消型 | 正在执行指令 | 发生异常的指令的 PC 值 |
| 复位 | | 指令处理放弃型 | 各机器周期 | 无 |
| 非屏蔽中断 | 正在执行 RMPA、SCMPU、SMOVB、SMOVF、SMOVU、SSTR、SUNTIL、SWHILE 各指令 | 指令处理中止型 | 正在执行指令 | 正在执行的指令的 PC 值 |
| | 上述以外的状态 | 指令处理结束型 | 在指令和指令之间 | 下一条指令的 PC 值 |
| 中断 | 正在执行 RMPA、SCMPU、SMOVB、SMOVF、SMOVU、SSTR、SUNTIL、SWHILE 各指令 | 指令处理中止型 | 正在执行指令 | 正在执行的指令的 PC 值 |
| | 上述以外的状态 | 指令处理结束型 | 在指令和指令之间 | 下一条指令的 PC 值 |
| 无条件陷阱 | | 指令处理结束型 | 在指令和指令之间 | 下一条指令的 PC 值 |

9.3.2 向量和 PC、PSW 的保存场所

各异常事件的向量、程序计数器（PC）和处理器状态字（PSW）的保存场所如表 9.2 所示。

表 9.2 向量和 PC、PSW 的保存场所

| 异常事件 | | 向量 | PC 和 PSW 的保存场所 |
|---------|--------|-------------|----------------|
| 未定义指令异常 | | 固定向量表 | 堆栈 |
| 特权指令异常 | | 固定向量表 | 堆栈 |
| 浮点异常 | | 固定向量表 | 堆栈 |
| 复位 | | 固定向量表 | 无 |
| 非屏蔽中断 | | 固定向量表 | 堆栈 |
| 中断 | 高速中断 | FINTV | BPC、BPSW |
| | 高速中断以外 | 可变向量表（INTB） | 堆栈 |
| 无条件陷阱 | | 可变向量表（INTB） | 堆栈 |

9.4 接受异常 / 从异常返回时的硬件处理

以下说明接受异常以及从异常返回时的硬件处理（复位除外）。

(1) 接受异常时的硬件预处理

(a) PSW 的保存

- 高速中断的情况
PSW→BPSW
- 高速中断以外的异常情况
PSW→堆栈

注 1. 在硬件预处理中不保存 FPSW。如果在异常处理程序内使用浮点运算指令，用户就必须在异常处理程序内将 FPSW 压栈。

(b) PSW 的 PM 位、U 位和 I 位的更新

I : 置“0”
U : 置“0”
PM : 置“0”

(c) PC 的保存

- 高速中断的情况
PC→BPC
- 高速中断以外的异常情况
PC→堆栈

(d) 给 PC 设定异常处理程序的转移目标地址

通过取得对应各异常的向量，转移到异常处理程序的处理。

(2) 执行 RTE 指令和 RTFI 指令时的硬件后处理

(a) PSW 的恢复

- 高速中断的情况
BPSW→PSW
- 高速中断以外的异常情况
堆栈→PSW

(b) PC 的恢复

- 高速中断的情况
BPC→PC
- 高速中断以外的异常情况
堆栈→PC

9.5 硬件预处理

以下说明从接受异常请求到执行异常处理程序的硬件预处理。

9.5.1 未定义指令异常

1. 将处理器状态字 (PSW) 的内容压栈 (ISP)。
2. 将PSW的处理器模式设定位 (PM)、堆栈指针指定位 (U) 和中断允许位 (I) 置“0”。
3. 将程序计数器 (PC) 的内容压栈 (ISP)。
4. 从地址FFFF FFDCh取向量。
5. 将取到的向量设定到PC后转移到异常处理程序。

9.5.2 特权指令异常

1. 将处理器状态字 (PSW) 的内容压栈 (ISP)。
2. 将PSW的处理器模式设定位 (PM)、堆栈指针指定位 (U) 和中断允许位 (I) 置“0”。
3. 将程序计数器 (PC) 的内容压栈 (ISP)。
4. 从地址FFFF FFD0h取向量。
5. 将取到的向量设定到PC后转移到异常处理程序。

9.5.3 浮点异常

1. 将处理器状态字 (PSW) 的内容压栈 (ISP)。
2. 将PSW的处理器模式设定位 (PM)、堆栈指针指定位 (U) 和中断允许位 (I) 置“0”。
3. 将程序计数器 (PC) 的内容压栈 (ISP)。
4. 从地址FFFF FFE4h取向量。
5. 将取到的向量设定到PC后转移到异常处理程序。

9.5.4 复位

1. 对控制进行初始化。
2. 从地址FFFF FFFCh取向量。
3. 将取到的向量设定到程序计数器 (PC)。

9.5.5 非屏蔽中断

1. 将处理器状态字（PSW）的内容压栈（ISP）。
2. 将PSW的处理器模式设定位（PM）、堆栈指针指定位（U）和中断允许位（I）置“0”。
3. 当正在执行RMPA、SCMPU、SMOVB、SMOVF、SMOVU、SSTR、SUNTIL、SWHILE各指令时，将正在执行的指令的程序计数器（PC）内容压栈（ISP）；而在其他状态下，将下一条指令的PC内容压栈（ISP）。
4. 将PSW的处理器中断优先级（IPL[2:0]）置“111b”。
5. 从地址FFFF FFF8h取向量。
6. 将取到的向量设定到PC后转移到异常处理程序。

9.5.6 中断

1. 将处理器状态字（PSW）的内容压栈（ISP）。在高速中断的情况下，保存到备用PSW（BPSW）。
2. 将PSW的处理器模式设定位（PM）、堆栈指针指定位（U）和中断允许位（I）置“0”。
3. 当正在执行RMPA、SCMPU、SMOVB、SMOVF、SMOVU、SSTR、SUNTIL、SWHILE各指令时，将正在执行的指令的程序计数器（PC）内容压栈（ISP）；而在其他状态下，将下一条指令的PC内容压栈（ISP）。在高速中断的情况下，保存到备用PC（BPC）。
4. 给PSW的处理器中断优先级（IPL[2:0]）设定已接受中断的中断优先级。
5. 从可变量量表取已接受中断源的向量。在高速中断的情况下，从高速中断向量寄存器（FINTV）取向量。
6. 将取到的向量设定到PC后转移到异常处理程序。

9.5.7 无条件陷阱

1. 将处理器状态字（PSW）的内容压栈（ISP）。
2. 将PSW的处理器模式设定位（PM）、堆栈指针指定位（U）和中断允许位（I）置“0”。
3. 将下一条指令的程序计数器（PC）内容压栈（ISP）。
4. 在使用INT指令时，从可变量量表取对应INT指令号的向量。
在使用BRK指令时，从可变量量表的起始地址取向量。
5. 将取到的向量设定到PC后转移到异常处理程序。

9.6 从异常处理程序的返回

如果在异常处理程序的最后执行表 9.3 所示的指令，就恢复异常处理顺序前被压栈或者保存到控制寄存器（BPC 和 BPSW）的程序计数器（PC）和处理器状态字（PSW）的内容。

表 9.3 异常处理程序的返回指令

| 异常事件 | | 返回指令 |
|---------|--------|------|
| 未定义指令异常 | | RTE |
| 特权指令异常 | | RTE |
| 浮点异常 | | RTE |
| 复位 | | 不能返回 |
| 非屏蔽中断 | | 不能返回 |
| 中断 | 高速中断 | RTFI |
| | 高速中断以外 | RTE |
| 无条件陷阱 | | RTE |

9.7 异常事件的优先级

异常事件的优先级如表 9.4 所示。如果同时发生多个异常，就先接受优先级高的事件。

表 9.4 中断优先级

| 优先级 | | 异常事件 |
|---|---|-------------------|
| 高  低 | 1 | 复位 |
| | 2 | 非屏蔽中断 |
| | 3 | 中断 |
| | 4 | 未定义指令异常 特权指令异常 |
| | 5 | 无条件陷阱 |
| | 6 | 浮点异常 |

10. 中断控制器 (ICU)

10.1 概要

中断控制器通过外围模块和外部引脚的中断源，向 CPU 输出中断请求以及启动 DTC 和 DMAC。
中断控制器的规格和框图分别如表 10.1 和图 10.1 所示。

表 10.1 中断控制器的规格

| 项目 | | 内容 |
|----------------|----------|--|
| 中断 | 外围功能中断 | 外围模块的中断 <ul style="list-style-type: none"> 中断源数：116 个 中断检测：边沿检测 / 电平检测 连接外围模块的各中断源的检测方法已有规定。 |
| | 外部中断 | IRQ15 ~ IRQ0 引脚的中断 <ul style="list-style-type: none"> 中断源数：16 个 中断检测：各中断源能设定 Low 电平、下降沿、上升沿或者双边沿。 |
| 非屏蔽中断 | NMI 引脚中断 | NMI 引脚的中断 <ul style="list-style-type: none"> 非屏蔽中断的中断源数：1 个 中断检测：下降沿 / 上升沿 |
| 中断优先级 | | 通过寄存器设定优先级。 |
| 向量地址 | | 因为给各中断源分配了向量地址，所以不需要通过软件判断中断源。 |
| 高速中断功能 | | 能实现 CPU 中断处理的高速化。只能给 1 个中断源设定此功能。 |
| DTC 和 DMAC 的控制 | | 能通过中断源启动 DTC 和 DMAC。 <ul style="list-style-type: none"> 能启动 DTC 的中断源：86 个（70 个外围功能中断 + 16 个外部中断） 能启动 DMAC 的中断源：42 个（38 个外围功能中断 + 4 个外部中断） |
| 从低功耗状态的返回 | | <ul style="list-style-type: none"> 睡眠模式：通过 NMI 引脚中断或者中断返回。 全模块时钟停止模式：通过 NMI 引脚中断、外部中断或者外围功能中断（WDT 和 TMR）返回。 软件待机模式：通过 NMI 引脚中断或者外部中断返回。 |

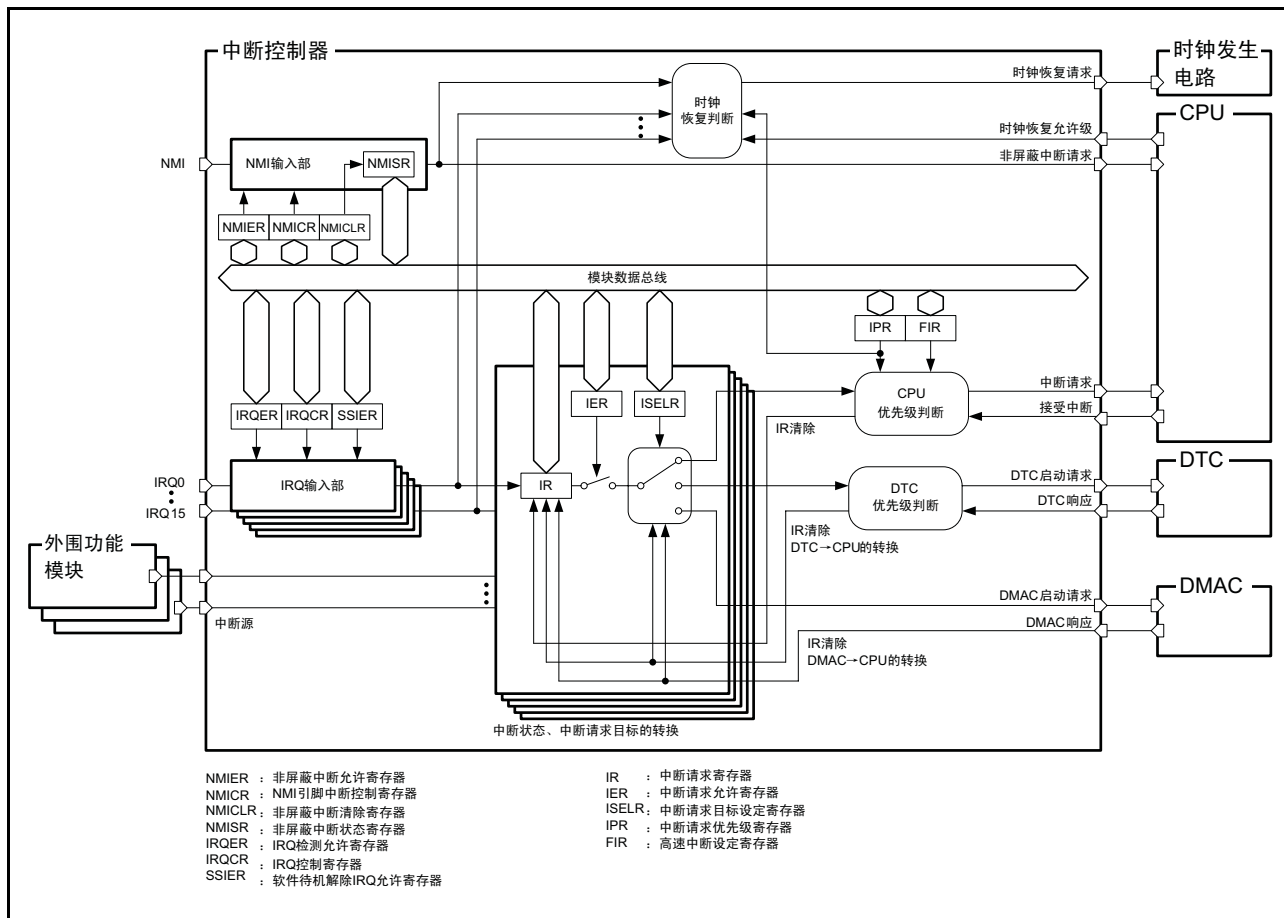


图 10.1 中断控制器的框图

中断控制器使用的输入 / 输出引脚如表 10.2 所示。

表 10.2 中断控制器的输入 / 输出引脚

| 引脚名 | 输入 / 输出 | 功能 |
|--------------|---------|-----------|
| NMI | 输入 | 非屏蔽中断请求引脚 |
| IRQ15 ~ IRQ0 | 输入 | 外部中断请求引脚 |

10.2 寄存器说明

中断控制器的寄存器一览表如表 10.3 所示。

表 10.3 中断控制器的寄存器一览表 (1/8)

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|-------------|-------|-------|------------|------|
| 中断请求寄存器 016 | IR016 | 00h | 0008 7010h | 8 |
| 中断请求寄存器 021 | IR021 | 00h | 0008 7015h | 8 |
| 中断请求寄存器 023 | IR023 | 00h | 0008 7017h | 8 |
| 中断请求寄存器 028 | IR028 | 00h | 0008 701Ch | 8 |
| 中断请求寄存器 029 | IR029 | 00h | 0008 701Dh | 8 |
| 中断请求寄存器 030 | IR030 | 00h | 0008 701Eh | 8 |
| 中断请求寄存器 031 | IR031 | 00h | 0008 701Fh | 8 |
| 中断请求寄存器 064 | IR064 | 00h | 0008 7040h | 8 |
| 中断请求寄存器 065 | IR065 | 00h | 0008 7041h | 8 |
| 中断请求寄存器 066 | IR066 | 00h | 0008 7042h | 8 |
| 中断请求寄存器 067 | IR067 | 00h | 0008 7043h | 8 |
| 中断请求寄存器 068 | IR068 | 00h | 0008 7044h | 8 |
| 中断请求寄存器 069 | IR069 | 00h | 0008 7045h | 8 |
| 中断请求寄存器 070 | IR070 | 00h | 0008 7046h | 8 |
| 中断请求寄存器 071 | IR071 | 00h | 0008 7047h | 8 |
| 中断请求寄存器 072 | IR072 | 00h | 0008 7048h | 8 |
| 中断请求寄存器 073 | IR073 | 00h | 0008 7049h | 8 |
| 中断请求寄存器 074 | IR074 | 00h | 0008 704Ah | 8 |
| 中断请求寄存器 075 | IR075 | 00h | 0008 704Bh | 8 |
| 中断请求寄存器 076 | IR076 | 00h | 0008 704Ch | 8 |
| 中断请求寄存器 077 | IR077 | 00h | 0008 704Dh | 8 |
| 中断请求寄存器 078 | IR078 | 00h | 0008 704Eh | 8 |
| 中断请求寄存器 079 | IR079 | 00h | 0008 704Fh | 8 |
| 中断请求寄存器 096 | IR096 | 00h | 0008 7060h | 8 |
| 中断请求寄存器 098 | IR098 | 00h | 0008 7062h | 8 |
| 中断请求寄存器 099 | IR099 | 00h | 0008 7063h | 8 |
| 中断请求寄存器 100 | IR100 | 00h | 0008 7064h | 8 |
| 中断请求寄存器 101 | IR101 | 00h | 0008 7065h | 8 |
| 中断请求寄存器 104 | IR104 | 00h | 0008 7068h | 8 |
| 中断请求寄存器 105 | IR105 | 00h | 0008 7069h | 8 |
| 中断请求寄存器 106 | IR106 | 00h | 0008 706Ah | 8 |
| 中断请求寄存器 107 | IR107 | 00h | 0008 706Bh | 8 |
| 中断请求寄存器 108 | IR108 | 00h | 0008 706Ch | 8 |
| 中断请求寄存器 111 | IR111 | 00h | 0008 706Fh | 8 |
| 中断请求寄存器 112 | IR112 | 00h | 0008 7070h | 8 |
| 中断请求寄存器 115 | IR115 | 00h | 0008 7073h | 8 |
| 中断请求寄存器 116 | IR116 | 00h | 0008 7074h | 8 |
| 中断请求寄存器 117 | IR117 | 00h | 0008 7075h | 8 |
| 中断请求寄存器 118 | IR118 | 00h | 0008 7076h | 8 |
| 中断请求寄存器 120 | IR120 | 00h | 0008 7078h | 8 |
| 中断请求寄存器 121 | IR121 | 00h | 0008 7079h | 8 |
| 中断请求寄存器 122 | IR122 | 00h | 0008 707Ah | 8 |

表 10.3 中断控制器的寄存器一览表 (2/8)

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|-------------|-------|-------|------------|------|
| 中断请求寄存器 123 | IR123 | 00h | 0008 707Bh | 8 |
| 中断请求寄存器 124 | IR124 | 00h | 0008 707Ch | 8 |
| 中断请求寄存器 125 | IR125 | 00h | 0008 707Dh | 8 |
| 中断请求寄存器 126 | IR126 | 00h | 0008 707Eh | 8 |
| 中断请求寄存器 127 | IR127 | 00h | 0008 707Fh | 8 |
| 中断请求寄存器 128 | IR128 | 00h | 0008 7080h | 8 |
| 中断请求寄存器 131 | IR131 | 00h | 0008 7083h | 8 |
| 中断请求寄存器 132 | IR132 | 00h | 0008 7084h | 8 |
| 中断请求寄存器 133 | IR133 | 00h | 0008 7085h | 8 |
| 中断请求寄存器 134 | IR134 | 00h | 0008 7086h | 8 |
| 中断请求寄存器 136 | IR136 | 00h | 0008 7088h | 8 |
| 中断请求寄存器 137 | IR137 | 00h | 0008 7089h | 8 |
| 中断请求寄存器 138 | IR138 | 00h | 0008 708Ah | 8 |
| 中断请求寄存器 139 | IR139 | 00h | 0008 708Bh | 8 |
| 中断请求寄存器 140 | IR140 | 00h | 0008 708Ch | 8 |
| 中断请求寄存器 141 | IR141 | 00h | 0008 708Dh | 8 |
| 中断请求寄存器 142 | IR142 | 00h | 0008 708Eh | 8 |
| 中断请求寄存器 145 | IR145 | 00h | 0008 7091h | 8 |
| 中断请求寄存器 146 | IR146 | 00h | 0008 7092h | 8 |
| 中断请求寄存器 149 | IR149 | 00h | 0008 7095h | 8 |
| 中断请求寄存器 150 | IR150 | 00h | 0008 7096h | 8 |
| 中断请求寄存器 151 | IR151 | 00h | 0008 7097h | 8 |
| 中断请求寄存器 152 | IR152 | 00h | 0008 7098h | 8 |
| 中断请求寄存器 154 | IR154 | 00h | 0008 709Ah | 8 |
| 中断请求寄存器 155 | IR155 | 00h | 0008 709Bh | 8 |
| 中断请求寄存器 156 | IR156 | 00h | 0008 709Ch | 8 |
| 中断请求寄存器 157 | IR157 | 00h | 0008 709Dh | 8 |
| 中断请求寄存器 158 | IR158 | 00h | 0008 709Eh | 8 |
| 中断请求寄存器 159 | IR159 | 00h | 0008 709Fh | 8 |
| 中断请求寄存器 160 | IR160 | 00h | 0008 70A0h | 8 |
| 中断请求寄存器 161 | IR161 | 00h | 0008 70A1h | 8 |
| 中断请求寄存器 162 | IR162 | 00h | 0008 70A2h | 8 |
| 中断请求寄存器 165 | IR165 | 00h | 0008 70A5h | 8 |
| 中断请求寄存器 166 | IR166 | 00h | 0008 70A6h | 8 |
| 中断请求寄存器 167 | IR167 | 00h | 0008 70A7h | 8 |
| 中断请求寄存器 168 | IR168 | 00h | 0008 70A8h | 8 |
| 中断请求寄存器 170 | IR170 | 00h | 0008 70AAh | 8 |
| 中断请求寄存器 171 | IR171 | 00h | 0008 70ABh | 8 |
| 中断请求寄存器 174 | IR174 | 00h | 0008 70AEh | 8 |
| 中断请求寄存器 175 | IR175 | 00h | 0008 70AFh | 8 |
| 中断请求寄存器 176 | IR176 | 00h | 0008 70B0h | 8 |
| 中断请求寄存器 177 | IR177 | 00h | 0008 70B1h | 8 |
| 中断请求寄存器 178 | IR178 | 00h | 0008 70B2h | 8 |
| 中断请求寄存器 179 | IR179 | 00h | 0008 70B3h | 8 |
| 中断请求寄存器 180 | IR180 | 00h | 0008 70B4h | 8 |

表 10.3 中断控制器的寄存器一览表 (3 / 8)

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|-------------|-------|-------|------------|------|
| 中断请求寄存器 181 | IR181 | 00h | 0008 70B5h | 8 |
| 中断请求寄存器 182 | IR182 | 00h | 0008 70B6h | 8 |
| 中断请求寄存器 183 | IR183 | 00h | 0008 70B7h | 8 |
| 中断请求寄存器 184 | IR184 | 00h | 0008 70B8h | 8 |
| 中断请求寄存器 185 | IR185 | 00h | 0008 70B9h | 8 |
| 中断请求寄存器 198 | IR198 | 00h | 0008 70C6h | 8 |
| 中断请求寄存器 199 | IR199 | 00h | 0008 70C7h | 8 |
| 中断请求寄存器 200 | IR200 | 00h | 0008 70C8h | 8 |
| 中断请求寄存器 201 | IR201 | 00h | 0008 70C9h | 8 |
| 中断请求寄存器 214 | IR214 | 00h | 0008 70D6h | 8 |
| 中断请求寄存器 215 | IR215 | 00h | 0008 70D7h | 8 |
| 中断请求寄存器 216 | IR216 | 00h | 0008 70D8h | 8 |
| 中断请求寄存器 217 | IR217 | 00h | 0008 70D9h | 8 |
| 中断请求寄存器 218 | IR218 | 00h | 0008 70DAh | 8 |
| 中断请求寄存器 219 | IR219 | 00h | 0008 70DBh | 8 |
| 中断请求寄存器 220 | IR220 | 00h | 0008 70DCh | 8 |
| 中断请求寄存器 221 | IR221 | 00h | 0008 70DDh | 8 |
| 中断请求寄存器 222 | IR222 | 00h | 0008 70DEh | 8 |
| 中断请求寄存器 223 | IR223 | 00h | 0008 70DFh | 8 |
| 中断请求寄存器 224 | IR224 | 00h | 0008 70E0h | 8 |
| 中断请求寄存器 225 | IR225 | 00h | 0008 70E1h | 8 |
| 中断请求寄存器 226 | IR226 | 00h | 0008 70E2h | 8 |
| 中断请求寄存器 227 | IR227 | 00h | 0008 70E3h | 8 |
| 中断请求寄存器 228 | IR228 | 00h | 0008 70E4h | 8 |
| 中断请求寄存器 229 | IR229 | 00h | 0008 70E5h | 8 |
| 中断请求寄存器 230 | IR230 | 00h | 0008 70E6h | 8 |
| 中断请求寄存器 231 | IR231 | 00h | 0008 70E7h | 8 |
| 中断请求寄存器 232 | IR232 | 00h | 0008 70E8h | 8 |
| 中断请求寄存器 233 | IR233 | 00h | 0008 70E9h | 8 |
| 中断请求寄存器 234 | IR234 | 00h | 0008 70EAh | 8 |
| 中断请求寄存器 235 | IR235 | 00h | 0008 70EBh | 8 |
| 中断请求寄存器 236 | IR236 | 00h | 0008 70ECh | 8 |
| 中断请求寄存器 237 | IR237 | 00h | 0008 70EDh | 8 |
| 中断请求寄存器 238 | IR238 | 00h | 0008 70EEh | 8 |
| 中断请求寄存器 239 | IR239 | 00h | 0008 70EFh | 8 |
| 中断请求寄存器 240 | IR240 | 00h | 0008 70F0h | 8 |
| 中断请求寄存器 241 | IR241 | 00h | 0008 70F1h | 8 |
| 中断请求寄存器 246 | IR246 | 00h | 0008 70F6h | 8 |
| 中断请求寄存器 247 | IR247 | 00h | 0008 70F7h | 8 |
| 中断请求寄存器 248 | IR248 | 00h | 0008 70F8h | 8 |
| 中断请求寄存器 249 | IR249 | 00h | 0008 70F9h | 8 |
| 中断请求寄存器 250 | IR250 | 00h | 0008 70FAh | 8 |
| 中断请求寄存器 251 | IR251 | 00h | 0008 70FBh | 8 |
| 中断请求寄存器 252 | IR252 | 00h | 0008 70FCh | 8 |
| 中断请求寄存器 253 | IR253 | 00h | 0008 70FDh | 8 |

表 10.3 中断控制器的寄存器一览表 (4/8)

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|-----------------|----------|-------|------------|------|
| 中断请求目标设定寄存器 028 | ISELR028 | 00h | 0008 711Ch | 8 |
| 中断请求目标设定寄存器 029 | ISELR029 | 00h | 0008 711Dh | 8 |
| 中断请求目标设定寄存器 030 | ISELR030 | 00h | 0008 711Eh | 8 |
| 中断请求目标设定寄存器 031 | ISELR031 | 00h | 0008 711Fh | 8 |
| 中断请求目标设定寄存器 064 | ISELR064 | 00h | 0008 7140h | 8 |
| 中断请求目标设定寄存器 065 | ISELR065 | 00h | 0008 7141h | 8 |
| 中断请求目标设定寄存器 066 | ISELR066 | 00h | 0008 7142h | 8 |
| 中断请求目标设定寄存器 067 | ISELR067 | 00h | 0008 7143h | 8 |
| 中断请求目标设定寄存器 068 | ISELR068 | 00h | 0008 7144h | 8 |
| 中断请求目标设定寄存器 069 | ISELR069 | 00h | 0008 7145h | 8 |
| 中断请求目标设定寄存器 070 | ISELR070 | 00h | 0008 7146h | 8 |
| 中断请求目标设定寄存器 071 | ISELR071 | 00h | 0008 7147h | 8 |
| 中断请求目标设定寄存器 072 | ISELR072 | 00h | 0008 7148h | 8 |
| 中断请求目标设定寄存器 073 | ISELR073 | 00h | 0008 7149h | 8 |
| 中断请求目标设定寄存器 074 | ISELR074 | 00h | 0008 714Ah | 8 |
| 中断请求目标设定寄存器 075 | ISELR075 | 00h | 0008 714Bh | 8 |
| 中断请求目标设定寄存器 076 | ISELR076 | 00h | 0008 714Ch | 8 |
| 中断请求目标设定寄存器 077 | ISELR077 | 00h | 0008 714Dh | 8 |
| 中断请求目标设定寄存器 078 | ISELR078 | 00h | 0008 714Eh | 8 |
| 中断请求目标设定寄存器 079 | ISELR079 | 00h | 0008 714Fh | 8 |
| 中断请求目标设定寄存器 098 | ISELR098 | 00h | 0008 7162h | 8 |
| 中断请求目标设定寄存器 099 | ISELR099 | 00h | 0008 7163h | 8 |
| 中断请求目标设定寄存器 100 | ISELR100 | 00h | 0008 7164h | 8 |
| 中断请求目标设定寄存器 101 | ISELR101 | 00h | 0008 7165h | 8 |
| 中断请求目标设定寄存器 104 | ISELR104 | 00h | 0008 7168h | 8 |
| 中断请求目标设定寄存器 105 | ISELR105 | 00h | 0008 7169h | 8 |
| 中断请求目标设定寄存器 106 | ISELR106 | 00h | 0008 716Ah | 8 |
| 中断请求目标设定寄存器 107 | ISELR107 | 00h | 0008 716Bh | 8 |
| 中断请求目标设定寄存器 111 | ISELR111 | 00h | 0008 716Fh | 8 |
| 中断请求目标设定寄存器 112 | ISELR112 | 00h | 0008 7170h | 8 |
| 中断请求目标设定寄存器 117 | ISELR117 | 00h | 0008 7175h | 8 |
| 中断请求目标设定寄存器 118 | ISELR118 | 00h | 0008 7176h | 8 |
| 中断请求目标设定寄存器 122 | ISELR122 | 00h | 0008 717Ah | 8 |
| 中断请求目标设定寄存器 123 | ISELR123 | 00h | 0008 717Bh | 8 |
| 中断请求目标设定寄存器 124 | ISELR124 | 00h | 0008 717Ch | 8 |
| 中断请求目标设定寄存器 125 | ISELR125 | 00h | 0008 717Dh | 8 |
| 中断请求目标设定寄存器 127 | ISELR127 | 00h | 0008 717Fh | 8 |
| 中断请求目标设定寄存器 128 | ISELR128 | 00h | 0008 7180h | 8 |
| 中断请求目标设定寄存器 133 | ISELR133 | 00h | 0008 7185h | 8 |
| 中断请求目标设定寄存器 134 | ISELR134 | 00h | 0008 7186h | 8 |
| 中断请求目标设定寄存器 138 | ISELR138 | 00h | 0008 718Ah | 8 |
| 中断请求目标设定寄存器 139 | ISELR139 | 00h | 0008 718Bh | 8 |
| 中断请求目标设定寄存器 140 | ISELR140 | 00h | 0008 718Ch | 8 |
| 中断请求目标设定寄存器 141 | ISELR141 | 00h | 0008 718Dh | 8 |
| 中断请求目标设定寄存器 145 | ISELR145 | 00h | 0008 7191h | 8 |

表 10.3 中断控制器的寄存器一览表 (5/8)

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|-----------------|----------|-------|------------|------|
| 中断请求目标设定寄存器 146 | ISELR146 | 00h | 0008 7192h | 8 |
| 中断请求目标设定寄存器 151 | ISELR151 | 00h | 0008 7197h | 8 |
| 中断请求目标设定寄存器 152 | ISELR152 | 00h | 0008 7198h | 8 |
| 中断请求目标设定寄存器 156 | ISELR156 | 00h | 0008 719Ch | 8 |
| 中断请求目标设定寄存器 157 | ISELR157 | 00h | 0008 719Dh | 8 |
| 中断请求目标设定寄存器 158 | ISELR158 | 00h | 0008 719Eh | 8 |
| 中断请求目标设定寄存器 159 | ISELR159 | 00h | 0008 719Fh | 8 |
| 中断请求目标设定寄存器 161 | ISELR161 | 00h | 0008 71A1h | 8 |
| 中断请求目标设定寄存器 162 | ISELR162 | 00h | 0008 71A2h | 8 |
| 中断请求目标设定寄存器 167 | ISELR167 | 00h | 0008 71A7h | 8 |
| 中断请求目标设定寄存器 168 | ISELR168 | 00h | 0008 71A8h | 8 |
| 中断请求目标设定寄存器 174 | ISELR174 | 00h | 0008 71AEh | 8 |
| 中断请求目标设定寄存器 175 | ISELR175 | 00h | 0008 71AFh | 8 |
| 中断请求目标设定寄存器 177 | ISELR177 | 00h | 0008 71B1h | 8 |
| 中断请求目标设定寄存器 178 | ISELR178 | 00h | 0008 71B2h | 8 |
| 中断请求目标设定寄存器 180 | ISELR180 | 00h | 0008 71B4h | 8 |
| 中断请求目标设定寄存器 181 | ISELR181 | 00h | 0008 71B5h | 8 |
| 中断请求目标设定寄存器 183 | ISELR183 | 00h | 0008 71B7h | 8 |
| 中断请求目标设定寄存器 184 | ISELR184 | 00h | 0008 71B8h | 8 |
| 中断请求目标设定寄存器 198 | ISELR198 | 00h | 0008 71C6h | 8 |
| 中断请求目标设定寄存器 199 | ISELR199 | 00h | 0008 71C7h | 8 |
| 中断请求目标设定寄存器 200 | ISELR200 | 00h | 0008 71C8h | 8 |
| 中断请求目标设定寄存器 201 | ISELR201 | 00h | 0008 71C9h | 8 |
| 中断请求目标设定寄存器 215 | ISELR215 | 00h | 0008 71D7h | 8 |
| 中断请求目标设定寄存器 216 | ISELR216 | 00h | 0008 71D8h | 8 |
| 中断请求目标设定寄存器 219 | ISELR219 | 00h | 0008 71DBh | 8 |
| 中断请求目标设定寄存器 220 | ISELR220 | 00h | 0008 71DCh | 8 |
| 中断请求目标设定寄存器 223 | ISELR223 | 00h | 0008 71DFh | 8 |
| 中断请求目标设定寄存器 224 | ISELR224 | 00h | 0008 71E0h | 8 |
| 中断请求目标设定寄存器 227 | ISELR227 | 00h | 0008 71E3h | 8 |
| 中断请求目标设定寄存器 228 | ISELR228 | 00h | 0008 71E4h | 8 |
| 中断请求目标设定寄存器 231 | ISELR231 | 00h | 0008 71E7h | 8 |
| 中断请求目标设定寄存器 232 | ISELR232 | 00h | 0008 71E8h | 8 |
| 中断请求目标设定寄存器 235 | ISELR235 | 00h | 0008 71EBh | 8 |
| 中断请求目标设定寄存器 236 | ISELR236 | 00h | 0008 71ECh | 8 |
| 中断请求目标设定寄存器 239 | ISELR239 | 00h | 0008 71EFh | 8 |
| 中断请求目标设定寄存器 240 | ISELR240 | 00h | 0008 71F0h | 8 |
| 中断请求目标设定寄存器 247 | ISELR247 | 00h | 0008 71F7h | 8 |
| 中断请求目标设定寄存器 248 | ISELR248 | 00h | 0008 71F8h | 8 |
| 中断请求目标设定寄存器 251 | ISELR251 | 00h | 0008 71FBh | 8 |
| 中断请求目标设定寄存器 252 | ISELR252 | 00h | 0008 71FCh | 8 |
| 中断请求允许寄存器 02 | IER02 | 00h | 0008 7202h | 8 |
| 中断请求允许寄存器 03 | IER03 | 00h | 0008 7203h | 8 |
| 中断请求允许寄存器 08 | IER08 | 00h | 0008 7208h | 8 |
| 中断请求允许寄存器 09 | IER09 | 00h | 0008 7209h | 8 |

表 10.3 中断控制器的寄存器一览表 (6 / 8)

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|--------------|-------|-------|------------|------|
| 中断请求允许寄存器 0C | IER0C | 00h | 0008 720Ch | 8 |
| 中断请求允许寄存器 0D | IER0D | 00h | 0008 720Dh | 8 |
| 中断请求允许寄存器 0E | IER0E | 00h | 0008 720Eh | 8 |
| 中断请求允许寄存器 0F | IER0F | 00h | 0008 720Fh | 8 |
| 中断请求允许寄存器 10 | IER10 | 00h | 0008 7210h | 8 |
| 中断请求允许寄存器 11 | IER11 | 00h | 0008 7211h | 8 |
| 中断请求允许寄存器 12 | IER12 | 00h | 0008 7212h | 8 |
| 中断请求允许寄存器 13 | IER13 | 00h | 0008 7213h | 8 |
| 中断请求允许寄存器 14 | IER14 | 00h | 0008 7214h | 8 |
| 中断请求允许寄存器 15 | IER15 | 00h | 0008 7215h | 8 |
| 中断请求允许寄存器 16 | IER16 | 00h | 0008 7216h | 8 |
| 中断请求允许寄存器 17 | IER17 | 00h | 0008 7217h | 8 |
| 中断请求允许寄存器 18 | IER18 | 00h | 0008 7218h | 8 |
| 中断请求允许寄存器 19 | IER19 | 00h | 0008 7219h | 8 |
| 中断请求允许寄存器 1A | IER1A | 00h | 0008 721Ah | 8 |
| 中断请求允许寄存器 1B | IER1B | 00h | 0008 721Bh | 8 |
| 中断请求允许寄存器 1C | IER1C | 00h | 0008 721Ch | 8 |
| 中断请求允许寄存器 1D | IER1D | 00h | 0008 721Dh | 8 |
| 中断请求允许寄存器 1E | IER1E | 00h | 0008 721Eh | 8 |
| 中断请求允许寄存器 1F | IER1F | 00h | 0008 721Fh | 8 |
| 中断源优先级寄存器 00 | IPR00 | 00h | 0008 7300h | 8 |
| 中断源优先级寄存器 01 | IPR01 | 00h | 0008 7301h | 8 |
| 中断源优先级寄存器 02 | IPR02 | 00h | 0008 7302h | 8 |
| 中断源优先级寄存器 04 | IPR04 | 00h | 0008 7304h | 8 |
| 中断源优先级寄存器 05 | IPR05 | 00h | 0008 7305h | 8 |
| 中断源优先级寄存器 06 | IPR06 | 00h | 0008 7306h | 8 |
| 中断源优先级寄存器 07 | IPR07 | 00h | 0008 7307h | 8 |
| 中断源优先级寄存器 20 | IPR20 | 00h | 0008 7320h | 8 |
| 中断源优先级寄存器 21 | IPR21 | 00h | 0008 7321h | 8 |
| 中断源优先级寄存器 22 | IPR22 | 00h | 0008 7322h | 8 |
| 中断源优先级寄存器 23 | IPR23 | 00h | 0008 7323h | 8 |
| 中断源优先级寄存器 24 | IPR24 | 00h | 0008 7324h | 8 |
| 中断源优先级寄存器 25 | IPR25 | 00h | 0008 7325h | 8 |
| 中断源优先级寄存器 26 | IPR26 | 00h | 0008 7326h | 8 |
| 中断源优先级寄存器 27 | IPR27 | 00h | 0008 7327h | 8 |
| 中断源优先级寄存器 28 | IPR28 | 00h | 0008 7328h | 8 |
| 中断源优先级寄存器 29 | IPR29 | 00h | 0008 7329h | 8 |
| 中断源优先级寄存器 2A | IPR2A | 00h | 0008 732Ah | 8 |
| 中断源优先级寄存器 2B | IPR2B | 00h | 0008 732Bh | 8 |
| 中断源优先级寄存器 2C | IPR2C | 00h | 0008 732Ch | 8 |
| 中断源优先级寄存器 2D | IPR2D | 00h | 0008 732Dh | 8 |
| 中断源优先级寄存器 2E | IPR2E | 00h | 0008 732Eh | 8 |
| 中断源优先级寄存器 2F | IPR2F | 00h | 0008 732Fh | 8 |
| 中断源优先级寄存器 40 | IPR40 | 00h | 0008 7340h | 8 |
| 中断源优先级寄存器 44 | IPR44 | 00h | 0008 7344h | 8 |

表 10.3 中断控制器的寄存器一览表 (7/8)

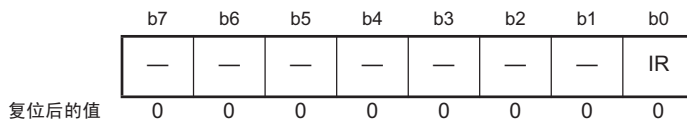
| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|--------------|-------|-------|------------|------|
| 中断源优先级寄存器 45 | IPR45 | 00h | 0008 7345h | 8 |
| 中断源优先级寄存器 46 | IPR46 | 00h | 0008 7346h | 8 |
| 中断源优先级寄存器 47 | IPR47 | 00h | 0008 7347h | 8 |
| 中断源优先级寄存器 4C | IPR4C | 00h | 0008 734Ch | 8 |
| 中断源优先级寄存器 4D | IPR4D | 00h | 0008 734Dh | 8 |
| 中断源优先级寄存器 4E | IPR4E | 00h | 0008 734Eh | 8 |
| 中断源优先级寄存器 4F | IPR4F | 00h | 0008 734Fh | 8 |
| 中断源优先级寄存器 50 | IPR50 | 00h | 0008 7350h | 8 |
| 中断源优先级寄存器 51 | IPR51 | 00h | 0008 7351h | 8 |
| 中断源优先级寄存器 52 | IPR52 | 00h | 0008 7352h | 8 |
| 中断源优先级寄存器 53 | IPR53 | 00h | 0008 7353h | 8 |
| 中断源优先级寄存器 54 | IPR54 | 00h | 0008 7354h | 8 |
| 中断源优先级寄存器 55 | IPR55 | 00h | 0008 7355h | 8 |
| 中断源优先级寄存器 56 | IPR56 | 00h | 0008 7356h | 8 |
| 中断源优先级寄存器 57 | IPR57 | 00h | 0008 7357h | 8 |
| 中断源优先级寄存器 58 | IPR58 | 00h | 0008 7358h | 8 |
| 中断源优先级寄存器 59 | IPR59 | 00h | 0008 7359h | 8 |
| 中断源优先级寄存器 5A | IPR5A | 00h | 0008 735Ah | 8 |
| 中断源优先级寄存器 5B | IPR5B | 00h | 0008 735Bh | 8 |
| 中断源优先级寄存器 5C | IPR5C | 00h | 0008 735Ch | 8 |
| 中断源优先级寄存器 5D | IPR5D | 00h | 0008 735Dh | 8 |
| 中断源优先级寄存器 5E | IPR5E | 00h | 0008 735Eh | 8 |
| 中断源优先级寄存器 5F | IPR5F | 00h | 0008 735Fh | 8 |
| 中断源优先级寄存器 60 | IPR60 | 00h | 0008 7360h | 8 |
| 中断源优先级寄存器 61 | IPR61 | 00h | 0008 7361h | 8 |
| 中断源优先级寄存器 62 | IPR62 | 00h | 0008 7362h | 8 |
| 中断源优先级寄存器 63 | IPR63 | 00h | 0008 7363h | 8 |
| 中断源优先级寄存器 68 | IPR68 | 00h | 0008 7368h | 8 |
| 中断源优先级寄存器 69 | IPR69 | 00h | 0008 7369h | 8 |
| 中断源优先级寄存器 6A | IPR6A | 00h | 0008 736Ah | 8 |
| 中断源优先级寄存器 6B | IPR6B | 00h | 0008 736Bh | 8 |
| 中断源优先级寄存器 70 | IPR70 | 00h | 0008 7370h | 8 |
| 中断源优先级寄存器 71 | IPR71 | 00h | 0008 7371h | 8 |
| 中断源优先级寄存器 72 | IPR72 | 00h | 0008 7372h | 8 |
| 中断源优先级寄存器 73 | IPR73 | 00h | 0008 7373h | 8 |
| 中断源优先级寄存器 80 | IPR80 | 00h | 0008 7380h | 8 |
| 中断源优先级寄存器 81 | IPR81 | 00h | 0008 7381h | 8 |
| 中断源优先级寄存器 82 | IPR82 | 00h | 0008 7382h | 8 |
| 中断源优先级寄存器 83 | IPR83 | 00h | 0008 7383h | 8 |
| 中断源优先级寄存器 84 | IPR84 | 00h | 0008 7384h | 8 |
| 中断源优先级寄存器 85 | IPR85 | 00h | 0008 7385h | 8 |
| 中断源优先级寄存器 86 | IPR86 | 00h | 0008 7386h | 8 |
| 中断源优先级寄存器 88 | IPR88 | 00h | 0008 7388h | 8 |
| 中断源优先级寄存器 89 | IPR89 | 00h | 0008 7389h | 8 |
| 中断源优先级寄存器 8A | IPR8A | 00h | 0008 738Ah | 8 |

表 10.3 中断控制器的寄存器一览表 (8/8)

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|------------------|---------|-------|------------|------|
| 中断源优先级寄存器 8B | IPR8B | 00h | 0008 738Bh | 8 |
| 中断源优先级寄存器 8C | IPR8C | 00h | 0008 738Ch | 8 |
| 中断源优先级寄存器 8D | IPR8D | 00h | 0008 738Dh | 8 |
| 中断源优先级寄存器 8E | IPR8E | 00h | 0008 738Eh | 8 |
| 中断源优先级寄存器 8F | IPR8F | 00h | 0008 738Fh | 8 |
| 高速中断设定寄存器 | FIR | 0000h | 0008 73F0h | 16 |
| IRQ 检测允许寄存器 0 | IRQER0 | 00h | 0008 C300h | 8 |
| IRQ 检测允许寄存器 1 | IRQER1 | 00h | 0008 C301h | 8 |
| IRQ 检测允许寄存器 2 | IRQER2 | 00h | 0008 C302h | 8 |
| IRQ 检测允许寄存器 3 | IRQER3 | 00h | 0008 C303h | 8 |
| IRQ 检测允许寄存器 4 | IRQER4 | 00h | 0008 C304h | 8 |
| IRQ 检测允许寄存器 5 | IRQER5 | 00h | 0008 C305h | 8 |
| IRQ 检测允许寄存器 6 | IRQER6 | 00h | 0008 C306h | 8 |
| IRQ 检测允许寄存器 7 | IRQER7 | 00h | 0008 C307h | 8 |
| IRQ 检测允许寄存器 8 | IRQER8 | 00h | 0008 C308h | 8 |
| IRQ 检测允许寄存器 9 | IRQER9 | 00h | 0008 C309h | 8 |
| IRQ 检测允许寄存器 10 | IRQER10 | 00h | 0008 C30Ah | 8 |
| IRQ 检测允许寄存器 11 | IRQER11 | 00h | 0008 C30Bh | 8 |
| IRQ 检测允许寄存器 12 | IRQER12 | 00h | 0008 C30Ch | 8 |
| IRQ 检测允许寄存器 13 | IRQER13 | 00h | 0008 C30Dh | 8 |
| IRQ 检测允许寄存器 14 | IRQER14 | 00h | 0008 C30Eh | 8 |
| IRQ 检测允许寄存器 15 | IRQER15 | 00h | 0008 C30Fh | 8 |
| IRQ 控制寄存器 0 | IRQCR0 | 00h | 0008 C320h | 8 |
| IRQ 控制寄存器 1 | IRQCR1 | 00h | 0008 C321h | 8 |
| IRQ 控制寄存器 2 | IRQCR2 | 00h | 0008 C322h | 8 |
| IRQ 控制寄存器 3 | IRQCR3 | 00h | 0008 C323h | 8 |
| IRQ 控制寄存器 4 | IRQCR4 | 00h | 0008 C324h | 8 |
| IRQ 控制寄存器 5 | IRQCR5 | 00h | 0008 C325h | 8 |
| IRQ 控制寄存器 6 | IRQCR6 | 00h | 0008 C326h | 8 |
| IRQ 控制寄存器 7 | IRQCR7 | 00h | 0008 C327h | 8 |
| IRQ 控制寄存器 8 | IRQCR8 | 00h | 0008 C328h | 8 |
| IRQ 控制寄存器 9 | IRQCR9 | 00h | 0008 C329h | 8 |
| IRQ 控制寄存器 10 | IRQCR10 | 00h | 0008 C32Ah | 8 |
| IRQ 控制寄存器 11 | IRQCR11 | 00h | 0008 C32Bh | 8 |
| IRQ 控制寄存器 12 | IRQCR12 | 00h | 0008 C32Ch | 8 |
| IRQ 控制寄存器 13 | IRQCR13 | 00h | 0008 C32Dh | 8 |
| IRQ 控制寄存器 14 | IRQCR14 | 00h | 0008 C32Eh | 8 |
| IRQ 控制寄存器 15 | IRQCR15 | 00h | 0008 C32Fh | 8 |
| 软件待机解除 IRQ 允许寄存器 | SSIER | 0000h | 0008 C340h | 16 |
| 非屏蔽中断允许寄存器 | NMIER | 00h | 0008 C350h | 8 |
| NMI 引脚中断控制寄存器 | NMICR | 00h | 0008 C351h | 8 |
| 非屏蔽中断状态寄存器 | NMISR | 00h | 0008 C352h | 8 |
| 非屏蔽中断清除寄存器 | NMICLR | 00h | 0008 C353h | 8 |

10.2.1 中断请求寄存器 i (IRi) (i= 中断向量号)

地址 0008 7010h ~ 0008 70FDh



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----|--------|----------------------|---------------|
| b0 | IR | 中断状态标志 | 0: 无中断请求 1: 有中断请求 | R/(W) (注1) |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

注1. 只能写“0”，不能写“1”。

IRi 寄存器是中断请求的状态寄存器。

各中断源都有 IRi 寄存器，i 对应中断向量号。

有关中断源和中断向量号的对应，请参照“表 10.4 中断向量表”。

IR 标志 (中断状态标志)

这是中断请求的状态标志。

当此标志为“1”时，如果通过 IERi.IENj 位允许中断，就将中断请求输出到 ISELRi.ISEL[1:0] 位设定的中断请求目标。

如果在中断请求的发生源检测到中断，此标志就变为“1”。为了检测中断，需要通过外围功能模块的中断允许位允许中断的输出，或者通过 IRQn 引脚的 IRQERn.IRQEN 位允许外部中断的检测。

中断的检测方法有边沿检测和电平检测两种，详细内容请参照“10.4.2 中断状态标志”。

●外围功能中断 (IR064 ~ IR079 除外)

边沿检测的情况

[为“1”的条件]

- 产生中断请求时

[为“0”的条件]

- 写“0”时
- 在 ISELRi.ISEL[1:0] 位为“00b”的状态下，CPU 执行了异常处理时
- 在 ISELRi.ISEL[1:0] 位为“01b”并且 DTC 的 MRB.DISEL 位为“0”的状态下，启动了 DTC 时
- 在 ISELRi.ISEL[1:0] 位为“10b”的状态下，启动了 DMAC 时

电平检测的情况

[为“1”的条件]

- 产生中断请求时

[为“0”的条件]

- 将中断请求源的状态标志置“0”时
- 通过中断请求源的中断允许位禁止了中断请求的输出时

●外部中断 (IR064~IR079)

边沿检测的情况 (IRQCRn.IRQMD[1:0] 位 = 01b/10b/11b)

[为“1”的条件]

- 产生中断请求时

[为“0”的条件]

- 写“0”时
- 在 ISEL_{Ri}.ISEL[1:0] 位为“00b”的状态下，CPU 执行了异常处理时
- 在 ISEL_{Ri}.ISEL[1:0] 位为“01b”并且 DTC 的 MRB.DISEL 位为“0”的状态下，启动了 DTC 时
- 在 ISEL_{Ri}.ISEL[1:0] 位为“10b”的状态下，启动了 DMAC 时

电平检测的情况 (IRQCR_i.IRQMD[1:0] 位 = 00b)

[为“1”的条件]

- 产生中断请求时

[为“0”的条件]

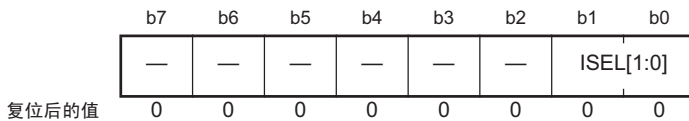
- 在 IRQ_n 引脚为 High 电平的状态下写“0” (注1、注2) 时
- 将 IRQ_n 引脚的 IRQER_n.IRQEN 位置“0” (禁止中断的检测) 时

注 1. 有关外部中断为电平检测时的清除方法的详细内容，请参照“10.4.2.2 电平检测的中断状态标志”。

注 2. 不能在 IRQ_n 引脚为 Low 电平的状态下将 IR 标志置“0”，否则 IR 标志就变为“0”并且中断请求被取消，在经过 2 个 PCLK 时钟后，IR 标志将再次变为“1”并且产生中断请求。

10.2.2 中断请求目标设定寄存器 i (ISELR_i) (i= 中断向量号)

地址 0008 711Ch ~ 0008 71FCh



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-----------|-----------|---|--------------|
| b1-b0 | ISEL[1:0] | 中断请求目标设定位 | b1 b0 0 0: 向 CPU 请求中断。 0 1: 在启动 DTC 并且数据传送结束后，向 CPU 请求中断 (注 1)。 1 0: 启动 DMAC。 1 1: 在启动 DMAC 后，向 CPU 请求中断 (注 1)。 | R/W (注 2) |
| b7-b2 | — | 保留位 | 读写值都为“0”。 | R/W |

注 1. 在结束 DTC 的数据传送后或者启动 DMAC 后，ISEL[1:0] 位的值自动变为“00b”。详细内容请参照“10.4.3 中断请求目标的选择”。

注 2. 对于不能选择 DMAC 为中断请求目标源的寄存器，低 1 位有效，高 7 位为保留位并且读写值都为“0”。

ISELR_i 寄存器是设定中断请求目标的寄存器。

ISEL[1:0] 位 (中断请求目标设定位)

这些位是设定中断请求目标的位。

必须在通过 IER_i.IEN_j 位禁止中断后设定这些位。

能通过 ISEL[1:0] 位设定的中断请求目标取决于各中断源，中断请求目标和中断源的对应请参照“表 10.4 中断向量表”。

如果在 IR_i.IR 标志为“1”时通过 IER_i.IEN_j 位允许中断，就将中断请求输出到 ISEL[1:0] 位设定的请求目标。详细内容请参照“10.4.3 中断请求目标的选择”。

10.2.3 中断请求允许寄存器 i (IERi) (i=02h ~ 1Fh)

地址 0008 7202h ~ 0008 721Fh

| | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | IEN7 | IEN6 | IEN5 | IEN4 | IEN3 | IEN2 | IEN1 | IEN0 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W (注1) |
|----|------|---------|--------------------|-------------|
| b0 | IEN0 | 中断允许位 0 | 0: 禁止中断 1: 允许中断 | R/W |
| b1 | IEN1 | 中断允许位 1 | | R/W |
| b2 | IEN2 | 中断允许位 2 | | R/W |
| b3 | IEN3 | 中断允许位 3 | | R/W |
| b4 | IEN4 | 中断允许位 4 | | R/W |
| b5 | IEN5 | 中断允许位 5 | | R/W |
| b6 | IEN6 | 中断允许位 6 | | R/W |
| b7 | IEN7 | 中断允许位 7 | | R/W |

注 1. 对应被保留的中断源的位的读写值都为“0”。

IERm 寄存器是设定允许或者禁止中断的寄存器。

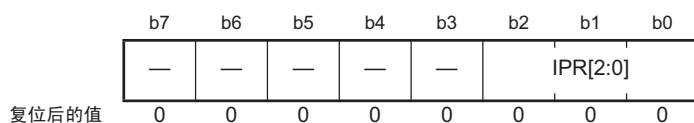
IENj 位 (中断允许位) (j = 0 ~ 7)

各中断源都有中断允许位，中断源和中断允许位的对应请参照“表 10.4 中断向量表”。

当此位为“1”时，允许中断；当此位为“0”时，禁止中断。

10.2.4 中断源优先级寄存器 i (IPRi) (i=00h~8Fh)

地址 0008 7300h~0008 738Fh



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|----------|---|-----|
| b2-b0 | IPR[2:0] | 中断优先级设定位 | b2 b0 0 0 0: 0 级 (禁止中断) 0 0 1: 1 级 0 1 0: 2 级 0 1 1: 3 级 1 0 0: 4 级 1 0 1: 5 级 1 1 0: 6 级 1 1 1: 7 级 (最高) | R/W |
| b7-b3 | — | 保留位 | 读写值都为“0”。 | R/W |

IPRi 寄存器是设定中断优先级的寄存器。

各中断组都有寄存器，i 为 00 ~ 8F (用 16 进制数表示) 的连号。

中断和中断组的对应请参照“表 10.4 中断向量表”。

IPR[2:0] 位 (中断优先级设定位)

这些位是选择中断请求优先级的位。

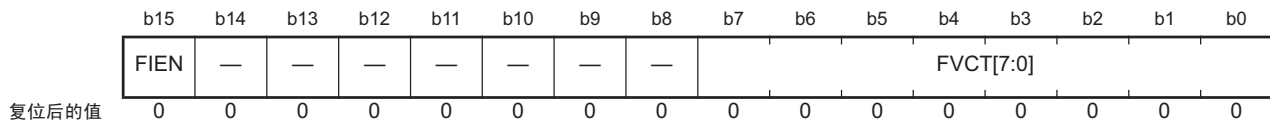
只在判断向 CPU 请求的中断优先级时参照 IPR[2:0] 位选择的优先级，不影响 DTC 和 DMAC 的传送请求。

CPU 只接受优先级高于 PSW.IPL[2:0] 位所示优先级的中断请求并且进行中断处理。

如果同时发生多个中断请求，就通过 IPR[2:0] 位的设定值比较优先级。如果同时发生相同优先级的中断请求，就优先接受向量号小的中断请求。

10.2.5 高速中断设定寄存器 (FIR)

地址 0008 73F0h



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|-----------|-----------|------------------------|-----|
| b7-b0 | FVCT[7:0] | 高速中断向量设定位 | 指定要设定为高速中断的中断向量号。 | R/W |
| b14-b8 | — | 保留位 | 读写值都为“0”。 | R/W |
| b15 | FIEN | 高速中断允许位 | 0: 禁止高速中断 1: 允许高速中断 | R/W |

FIR 寄存器是设定高速中断的寄存器。

只能将向 CPU 请求的中断设定为高速中断，不影响 DTC 和 DMAC 的传送请求。

FVCT[7:0] 位 (高速中断向量设定位)

这些位指定要设定为高速中断的中断向量号。

如果在 FIEN 位为“1”时产生所设向量号的中断请求，就作为高速中断请求将中断请求输出到 CPU，与 IPRi 寄存器的设定无关。

在通过 IERi.IENj 位禁止中断请求时，不将中断请求输出到 CPU。

能设定的向量号请参照“表 10.4 中断向量表”。

不能指定被保留的向量号。

FIEN 位 (高速中断允许位)

此位是允许高速中断的位。

通过将此位置“1”，FVCT[7:0] 位指定的向量号的中断变为高速中断。

高速中断的详细内容请参照“9. 异常处理”。

10.2.6 IRQ 检测允许寄存器 n (IRQERn) (n=0 ~ 15)

地址 0008 C300h ~ 0008 C30Fh

| | | | | | | | |
|----|----|----|----|----|----|----|-------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | — | — | — | IRQEN |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-----------|--|-----|
| b0 | IRQEN | IRQ 检测允许位 | 0: 禁止检测由 IRQn 引脚引起的外部中断 1: 允许检测由 IRQn 引脚引起的外部中断 (n=0 ~ 15) | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

IRQERn 寄存器禁止或者允许检测由 IRQn 引脚 (n=0 ~ 15) 引起的外部中断。

IRQEN 位 (IRQ 允许位)

此位禁止或者允许检测由 IRQn 引脚引起的外部中断。

10.2.7 IRQ 控制寄存器 n (IRQCRn) (n=0 ~ 15)

地址 0008 C320h ~ 0008 C32Fh

| | | | | | | | |
|----|----|----|----|------------|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | IRQMD[1:0] | — | — | — |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------------|-----------|--|-----|
| b1-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b3-b2 | IRQMD[1:0] | IRQ 检测设定位 | b3 b2 0 0: Low 电平 0 1: 下降沿 1 0: 上升沿 1 1: 双边沿 | R/W |
| b7-b4 | — | 保留位 | 读写值都为“0”。 | R/W |

IRQCRn 寄存器是设定外部中断引脚 IRQn (n=0 ~ 15) 的寄存器。

IRQMD[1:0] 位 (IRQ 检测设定位)

这些位设定外部中断 (IRQ0 ~ IRQ15) 的检测方法。

有关外部中断检测方法的设定, 请参照“10.4.6 外部中断”。

10.2.8 非屏蔽中断允许寄存器 (NMIER)

地址 0008 C350h

| | | | | | | | |
|----|----|----|----|----|----|----|-------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | — | — | — | NMIEN |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|---------|----------------------------------|---------------|
| b0 | NMIEN | NMI 允许位 | 0: 禁止 NMI 引脚中断 1: 允许 NMI 引脚中断 | R/(W) (注1) |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

注1. 只能写1次“1”，以后的写操作无效。

NMIER 寄存器是允许使用非屏蔽中断的寄存器。

NMIEN 位 (NMI 允许位)

此位是允许使用 NMI 引脚中断的位。

只能写1次“1”，以后的写操作无效。

不能写“0”。不能禁止已经允许的 NMI 引脚中断。

10.2.9 NMI 引脚中断控制寄存器 (NMICR)

地址 0008 C351h

| | | | | | | | |
|----|----|----|----|-------|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | NMIMD | — | — | — |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-----------|------------------|-----|
| b2-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b3 | NMIMD | NMI 检测设定位 | 0: 下降沿 1: 上升沿 | R/W |
| b7-b4 | — | 保留位 | 读写值都为“0”。 | R/W |

NMICR 寄存器是设定 NMI 引脚中断的寄存器。

必须在允许使用 NMI 引脚中断 (将 NMIER.NMIEN 位置“1”)前更改设定。

NMIMD 位 (NMI 检测设定位)

此位是设定 NMI 引脚中断检测方法的位。

10.2.10 非屏蔽中断状态寄存器 (NMISR)

地址 0008 C352h

| | | | | | | | |
|----|----|----|----|----|----|----|-------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | — | — | — | NMIST |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|----------|------------------------------------|-----|
| b0 | NMIST | NMI 状态标志 | 0: 无 NMI 引脚中断请求 1: 有 NMI 引脚中断请求 | R |
| b7-b1 | — | 保留位 | 读取值为“0”，写操作无效。 | R |

NMISR 寄存器是非屏蔽中断的状态寄存器。

NMIST 标志 (NMI 状态标志)

此标志表示 NMI 引脚的中断请求。

只能读 NMIST 标志，而且通过 NMICLR.NMICR 位将此标志置“0”。

[为“1”的条件]

- 如果在将 NMICR.NMIEN 位置“1”（允许 NMI 引脚中断）后将 NMICR.NMIMD 位选择的边沿输入到 NMI 引脚，此标志就变为“1”

[为“0”的条件]

- 给 NMICLR.NMICLR 位写“1”时

10.2.11 非屏蔽中断清除寄存器 (NMICLR)

地址 0008 C353h

| | | | | | | | |
|----|----|----|----|----|----|----|--------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | — | — | — | NMICLR |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|--------|---------|---|--------------|
| b0 | NMICLR | NMI 清除位 | 读取值为“0”。如果写“1”，NMISR.NMIST 标志就被置“0”。写“0”无效。 | R(W) (注1) |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

注 1. 只能写“1”。

NMICLR 寄存器是将非屏蔽中断状态寄存器 (NMISR) 置“00h”的寄存器。

NMICLR 位 (NMI 清除位)

如果给此位写“1”，NMISR.NMIST 标志就变为“0”。不保持“1”的状态，读取值总是“0”。

10.2.12 软件待机解除 IRQ 允许寄存器 (SSIER)

地址 0008 C340h

| | | | | | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | SSI15 | SSI14 | SSI13 | SSI12 | SSI11 | SSI10 | SSI9 | SSI8 | SSI7 | SSI6 | SSI5 | SSI4 | SSI3 | SSI2 | SSI1 | SSI0 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-----|-------|-----------------|--|-----|
| b0 | SSI0 | 软件待机解除的 IRQ 设定位 | 0: 不在软件待机状态下对 IRQ _n 引脚产生的外部中断源进行采样。 1: 如果在软件待机状态下发生由 IRQ _i 引脚引起的外部中断, 就在经过振荡稳定时间后从软件待机状态返回。(n=0 ~ 15) | R/W |
| b1 | SSI1 | | | R/W |
| b2 | SSI2 | | | R/W |
| b3 | SSI3 | | | R/W |
| b4 | SSI4 | | | R/W |
| b5 | SSI5 | | | R/W |
| b6 | SSI6 | | | R/W |
| b7 | SSI7 | | | R/W |
| b8 | SSI8 | | | R/W |
| b9 | SSI9 | | | R/W |
| b10 | SSI10 | | | R/W |
| b11 | SSI11 | | | R/W |
| b12 | SSI12 | | | R/W |
| b13 | SSI13 | | | R/W |
| b14 | SSI14 | | | R/W |
| b15 | SSI15 | | | R/W |

SSIER 寄存器选择是否将 IRQ_n (n=0 ~ 15) 引脚用作软件待机状态的返回源。

SSI_j 位 (j=0 ~ 15) (软件待机解除的 IRQ 设定位)

这些位设定是否将对位号的 IRQ_n 引脚用于从软件待机状态的返回。

有关从软件待机状态返回的设定, 请参照“10.6.2 从软件待机模式的返回”。

10.3 向量表

通过中断控制器检测的异常有中断和非屏蔽中断。如果 CPU 接受中断或者非屏蔽中断，就从向量表取 4 字节的向量地址。

10.3.1 中断向量表

中断向量表连续分配在从 CPU 的中断表寄存器 (INTB) 设定地址开始的 1024 字节 (4 字节 × 256 个中断源) 的区域。必须在允许中断前设定 INTB 寄存器，如果给 INTB 寄存器设定 4 的倍数，中断异常处理的执行速度就会变快。

中断向量表如表 10.4 所示，表中记载了中断源的发生源及其对应的向量号、向量地址以及可选择的中断请求目标。

表 10.4 中断向量表 (1/6)

| 优先级 | 中断请求发生源 | 名称 | 向量号 | 向量地址的偏移 | 中断的检测方法 | 可选择的中断请求目标 | | | IER | IPR |
|--------|-------------|---------|---------------|---------------|---------|------------|-----|------------|------------|-------|
| | | | | | | CPU | DTC | DMAC | | |
| ↑ 高 | — | 保留 | 0 | 0000h | — | × | × | × | — | — |
| | | 保留 | 1 | 0004h | — | × | × | × | — | — |
| | | 保留 | 2 | 0008h | — | × | × | × | — | — |
| | | 保留 | 3 | 000Ch | — | × | × | × | — | — |
| | | 保留 | 4 | 0010h | — | × | × | × | — | — |
| | | 保留 | 5 | 0014h | — | × | × | × | — | — |
| | | 保留 | 6 | 0018h | — | × | × | × | — | — |
| | | 保留 | 7 | 001Ch | — | × | × | × | — | — |
| | | 保留 | 8 | 0020h | — | × | × | × | — | — |
| | | 保留 | 9 ~ 15 | 0024h ~ 003Ch | — | × | × | × | — | — |
| ↓ 低 | 总线错误 | BUSERR | 16 | 0040h | 电平 | ○ | × | × | IER02.IEN0 | IPR00 |
| | — | 保留 | 17 | 0044h | — | × | × | × | IER02.IEN1 | — |
| | | 保留 | 18 | 0048h | — | × | × | × | IER02.IEN2 | — |
| | | 保留 | 19 | 004Ch | — | × | × | × | IER02.IEN3 | — |
| | | 保留 | 20 | 0050h | — | × | × | × | IER02.IEN4 | — |
| | FCU | FIFERR | 21 | 0054h | 电平 | ○ | × | × | IER02.IEN5 | IPR01 |
| | | 保留 | 22 | 0058h | — | × | × | × | IER02.IEN6 | — |
| | | FRDYI | 23 | 005Ch | 边沿 | ○ | × | × | IER02.IEN7 | IPR02 |
| | — | 保留 | 24 | 0060h | — | × | × | × | IER03.IEN0 | — |
| | | 保留 | 25 | 0064h | — | × | × | × | IER03.IEN1 | — |
| | | 保留 | 26 | 0068h | — | × | × | × | IER03.IEN2 | — |
| | | 保留 | 27 | 006Ch | — | × | × | × | IER03.IEN3 | — |
| | CMT 单元 0 | CMT0 | 28 | 0070h | 边沿 | ○ | ○ | ○ | IER03.IEN4 | IPR04 |
| | | CMT1 | 29 | 0074h | 边沿 | ○ | ○ | ○ | IER03.IEN5 | IPR05 |
| | CMT 单元 1 | CMT2 | 30 | 0078h | 边沿 | ○ | ○ | ○ | IER03.IEN6 | IPR06 |
| CMT3 | | 31 | 007Ch | 边沿 | ○ | ○ | ○ | IER03.IEN7 | IPR07 | |
| — | 保留 | 32 ~ 63 | 0080h ~ 00FCh | — | × | × | × | — | — | |

表 10.4 中断向量表 (2/6)

| 优先级 | 中断请求发生源 | 名称 | 向量号 | 向量地址的偏移 | 中断的检测方法 | 可选择的 中断请求目标 | | | IER | IPR |
|--------|---------|---------|---------------|---------|---------|----------------|-----|------------|------------|-------|
| | | | | | | CPU | DTC | DMAC | | |
| 高 ↑ | 外部引脚 | IRQ0 | 64 | 0100h | 边沿/电平 | ○ | ○ | ○ | IER08.IEN0 | IPR20 |
| | | IRQ1 | 65 | 0104h | 边沿/电平 | ○ | ○ | ○ | IER08.IEN1 | IPR21 |
| | | IRQ2 | 66 | 0108h | 边沿/电平 | ○ | ○ | ○ | IER08.IEN2 | IPR22 |
| | | IRQ3 | 67 | 010Ch | 边沿/电平 | ○ | ○ | ○ | IER08.IEN3 | IPR23 |
| | | IRQ4 | 68 | 0110h | 边沿/电平 | ○ | ○ | × | IER08.IEN4 | IPR24 |
| | | IRQ5 | 69 | 0114h | 边沿/电平 | ○ | ○ | × | IER08.IEN5 | IPR25 |
| | | IRQ6 | 70 | 0118h | 边沿/电平 | ○ | ○ | × | IER08.IEN6 | IPR26 |
| | | IRQ7 | 71 | 011Ch | 边沿/电平 | ○ | ○ | × | IER08.IEN7 | IPR27 |
| | | IRQ8 | 72 | 0120h | 边沿/电平 | ○ | ○ | × | IER09.IEN0 | IPR28 |
| | | IRQ9 | 73 | 0124h | 边沿/电平 | ○ | ○ | × | IER09.IEN1 | IPR29 |
| | | IRQ10 | 74 | 0128h | 边沿/电平 | ○ | ○ | × | IER09.IEN2 | IPR2A |
| | | IRQ11 | 75 | 012Ch | 边沿/电平 | ○ | ○ | × | IER09.IEN3 | IPR2B |
| | | IRQ12 | 76 | 0130h | 边沿/电平 | ○ | ○ | × | IER09.IEN4 | IPR2C |
| | | IRQ13 | 77 | 0134h | 边沿/电平 | ○ | ○ | × | IER09.IEN5 | IPR2D |
| | | IRQ14 | 78 | 0138h | 边沿/电平 | ○ | ○ | × | IER09.IEN6 | IPR2E |
| | | IRQ15 | 79 | 013Ch | 边沿/电平 | ○ | ○ | × | IER09.IEN7 | IPR2F |
| — | 保留 | 80 ~ 95 | 0140h ~ 017Ch | — | × | × | × | — | — | |
| WDT | WOVI | 96 | 0180h | 边沿 | ○ | × | × | IER0C.IEN0 | IPR40 | |
| | 保留 | 97 | 0184h | — | × | × | × | IER0C.IEN1 | — | |
| AD0 | ADI0 | 98 | 0188h | 边沿 | ○ | ○ | ○ | IER0C.IEN2 | IPR44 | |
| AD1 | ADI1 | 99 | 018Ch | 边沿 | ○ | ○ | ○ | IER0C.IEN3 | IPR45 | |
| AD2 | ADI2 | 100 | 0190h | 边沿 | ○ | ○ | ○ | IER0C.IEN4 | IPR46 | |
| AD3 | ADI3 | 101 | 0194h | 边沿 | ○ | ○ | ○ | IER0C.IEN5 | IPR47 | |
| — | 保留 | 102 | 0198h | — | × | × | × | IER0C.IEN6 | — | |
| | 保留 | 103 | 019Ch | — | × | × | × | IER0C.IEN7 | — | |
| TPU0 | TGI0A | 104 | 01A0h | 边沿 | ○ | ○ | ○ | IER0D.IEN0 | IPR4C | |
| | TGI0B | 105 | 01A4h | 边沿 | ○ | ○ | × | IER0D.IEN1 | | |
| | TGI0C | 106 | 01A8h | 边沿 | ○ | ○ | × | IER0D.IEN2 | | |
| | TGI0D | 107 | 01ACh | 边沿 | ○ | ○ | × | IER0D.IEN3 | | |
| | TCI0V | 108 | 01B0h | 边沿 | ○ | × | × | IER0D.IEN4 | IPR4D | |
| | 保留 | 109 | 01B4h | — | × | × | × | IER0D.IEN5 | — | |
| | 保留 | 110 | 01B8h | — | × | × | × | IER0D.IEN6 | — | |
| TPU1 | TGI1A | 111 | 01BCh | 边沿 | ○ | ○ | ○ | IER0E.IEN7 | IPR4E | |
| | TGI1B | 112 | 01C0h | 边沿 | ○ | ○ | × | IER0E.IEN0 | | |
| | 保留 | 113 | 01C4h | — | × | × | × | IER0E.IEN1 | — | |
| | 保留 | 114 | 01C8h | — | × | × | × | IER0E.IEN2 | — | |
| | TCI1V | 115 | 01CCh | 边沿 | ○ | × | × | IER0E.IEN3 | IPR4F | |
| | TCI1U | 116 | 01D0h | 边沿 | ○ | × | × | IER0E.IEN4 | | |
| 低 | | | | | | | | | | |

表 10.4 中断向量表 (5/6)

| 优先级 | 中断请求 发生源 | 名称 | 向量号 | 向量地址的 偏移 | 中断的 检测方法 | 可选择的 中断请求目标 | | | IER | IPR |
|--------|-------------|---------|-------|-------------|-------------|----------------|-----|------------|------------|-------|
| | | | | | | CPU | DTC | DMAC | | |
| 高 ↑ | DMAC | DMTEND0 | 198 | 0318h | 边沿 | ○ | ○ | × | IER18.IEN6 | IPR70 |
| | | DMTEND1 | 199 | 031Ch | 边沿 | ○ | ○ | × | IER18.IEN7 | IPR71 |
| | | DMTEND2 | 200 | 0320h | 边沿 | ○ | ○ | × | IER19.IEN0 | IPR72 |
| | | DMTEND3 | 201 | 0324h | 边沿 | ○ | ○ | × | IER19.IEN1 | IPR73 |
| | — | 保留 | 202 | 0328h | — | × | × | × | IER19.IEN2 | — |
| | | 保留 | 203 | 032C | — | × | × | × | IER19.IEN3 | — |
| | | 保留 | 204 | 0330h | — | × | × | × | IER19.IEN4 | — |
| | | 保留 | 205 | 0334h | — | × | × | × | IER19.IEN5 | — |
| | | 保留 | 206 | 0338h | — | × | × | × | IER19.IEN6 | — |
| | | 保留 | 207 | 033Ch | — | × | × | × | IER19.IEN7 | — |
| | | 保留 | 208 | 0340h | — | × | × | × | IER1A.IEN0 | — |
| | | 保留 | 209 | 0344h | — | × | × | × | IER1A.IEN1 | — |
| | | 保留 | 210 | 0348h | — | × | × | × | IER1A.IEN2 | — |
| | | 保留 | 211 | 034Ch | — | × | × | × | IER1A.IEN3 | — |
| | | 保留 | 212 | 0350h | — | × | × | × | IER1A.IEN4 | — |
| | | 保留 | 213 | 0354h | — | × | × | × | IER1A.IEN5 | — |
| | SCI0 | ERI0 | 214 | 0358h | 电平 | ○ | × | × | IER1A.IEN6 | IPR80 |
| | | RX10 | 215 | 035Ch | 边沿 | ○ | ○ | ○ | IER1A.IEN7 | |
| | | TX10 | 216 | 0360h | 边沿 | ○ | ○ | ○ | IER1B.IEN0 | |
| | | TEI0 | 217 | 0364h | 电平 | ○ | × | × | IER1B.IEN1 | |
| | SCI1 | ERI1 | 218 | 0368h | 电平 | ○ | × | × | IER1B.IEN2 | IPR81 |
| | | RX11 | 219 | 036Ch | 边沿 | ○ | ○ | ○ | IER1B.IEN3 | |
| | | TX11 | 220 | 0370h | 边沿 | ○ | ○ | ○ | IER1B.IEN4 | |
| | | TEI1 | 221 | 0374h | 电平 | ○ | × | × | IER1B.IEN5 | |
| | SCI2 | ERI2 | 222 | 0378h | 电平 | ○ | × | × | IER1B.IEN6 | IPR82 |
| | | RX12 | 223 | 037Ch | 边沿 | ○ | ○ | ○ | IER1B.IEN7 | |
| TX12 | | 224 | 0380h | 边沿 | ○ | ○ | ○ | IER1C.IEN0 | | |
| TEI2 | | 225 | 0384h | 电平 | ○ | × | × | IER1C.IEN1 | | |
| SCI3 | ERI3 | 226 | 0388h | 电平 | ○ | × | × | IER1C.IEN2 | IPR83 | |
| | RX13 | 227 | 038Ch | 边沿 | ○ | ○ | ○ | IER1C.IEN3 | | |
| | TX13 | 228 | 0390h | 边沿 | ○ | ○ | ○ | IER1C.IEN4 | | |
| | TEI3 | 229 | 0394h | 电平 | ○ | × | × | IER1C.IEN5 | | |
| SCI4 | ERI4 | 230 | 0398h | 电平 | ○ | × | × | IER1C.IEN6 | IPR84 | |
| | RX14 | 231 | 039Ch | 边沿 | ○ | ○ | ○ | IER1C.IEN7 | | |
| | TX14 | 232 | 03A0h | 边沿 | ○ | ○ | ○ | IER1D.IEN0 | | |
| | TEI4 | 233 | 03A4h | 电平 | ○ | × | × | IER1D.IEN1 | | |
| SCI5 | ERI5 | 234 | 03A8h | 电平 | ○ | × | × | IER1D.IEN2 | IPR85 | |
| | RX15 | 235 | 03ACh | 边沿 | ○ | ○ | ○ | IER1D.IEN3 | | |
| | TX15 | 236 | 03B0h | 边沿 | ○ | ○ | ○ | IER1D.IEN4 | | |
| | TEI5 | 237 | 03B4h | 电平 | ○ | × | × | IER1D.IEN5 | | |

表 10.4 中断向量表 (6/6)

| 优先级 | 中断请求发生源 | 名称 | 向量号 | 向量地址的偏移 | 中断的检测方法 | 可选择的 中断请求目标 | | | IER | IPR |
|-------------|---------|--------|-----|---------|---------|----------------|-----|------|------------|-------|
| | | | | | | CPU | DTC | DMAC | | |
| 高 ↑ 低 | SCI6 | ERI6 | 238 | 03B8h | 电平 | ○ | × | × | IER1D.IEN6 | IPR86 |
| | | RXI6 | 239 | 03BCh | 边沿 | ○ | ○ | ○ | IER1D.IEN7 | |
| | | TXI6 | 240 | 03C0h | 边沿 | ○ | ○ | ○ | IER1E.IEN0 | |
| | | TEI6 | 241 | 03C4h | 电平 | ○ | × | × | IER1E.IEN1 | |
| | — | 保留 | 242 | 03C8h7 | — | × | × | × | IER1E.IEN2 | — |
| | | 保留 | 243 | 03CCh | — | × | × | × | IER1E.IEN3 | — |
| | | 保留 | 244 | 03CDh | — | × | × | × | IER1E.IEN4 | — |
| | | 保留 | 245 | 03D4h | — | × | × | × | IER1E.IEN5 | — |
| | RIIC0 | ICEE10 | 246 | 03D8h | 电平 | ○ | × | × | IER1E.IEN6 | IPR88 |
| | | ICRXI0 | 247 | 03DCh | 边沿 | ○ | ○ | ○ | IER1E.IEN7 | IPR89 |
| | | ICTXI0 | 248 | 03E0h | 边沿 | ○ | ○ | ○ | IER1F.IEN0 | IPR8A |
| | | ICTEI0 | 249 | 03E4h | 电平 | ○ | × | × | IER1F.IEN1 | IPR8B |
| | RIIC1 | ICEE11 | 250 | 03E8h | 电平 | ○ | × | × | IER1F.IEN2 | IPR8C |
| | | ICRXI1 | 251 | 03ECh | 边沿 | ○ | ○ | ○ | IER1F.IEN3 | IPR8D |
| | | ICTXI1 | 252 | 03F0h | 边沿 | ○ | ○ | ○ | IER1F.IEN4 | IPR8E |
| | | ICTEI1 | 253 | 03F4h | 电平 | ○ | × | × | IER1F.IEN5 | IPR8F |
| | — | 保留 | 254 | 03F8h | — | × | × | × | IER1F.IEN6 | — |
| | | 保留 | 255 | 03FCh | — | × | × | × | IER1F.IEN7 | — |

○：可选择 ×：不可选择

10.3.2 高速中断的向量地址

通过 CPU 的高速中断向量寄存器 (FINTV) 指定被设定为高速中断的中断向量地址。

10.3.3 非屏蔽中断的向量地址

非屏蔽中断的向量地址分配在“FFFF FFF8h”。

10.4 运行说明

中断控制器判断中断和非屏蔽中断的优先级并且向 CPU、DTC 和 DMAC 输出中断请求。

如果产生中断请求，对应的 $IR_i.IR$ 标志就变为“1”，并且向中断请求目标输出中断请求。为了输出中断请求，需要通过 $IER_i.IEN_j$ 位允许中断。如果多个 $IR_i.IR$ 标志同时变为“1”，就分别向 CPU 和 DTC 输出优先级最高的中断源的中断请求（注1）。

注1. 如果同时产生多个 DMAC 启动请求，就由 DMAC 判断优先级。

10.4.1 中断的允许 / 禁止

要允许中断时，需要进行以下的设定：

- 在外围功能中断的情况下，通过外围模块的中断允许位允许输出中断请求。
- 在外部中断的情况下，通过 $IRQER_n.IRQEN$ 位允许检测由 IRQ_n 引脚引起的外部中断。
- 通过 $IER_i.IEN_j$ 位允许中断。

如果在中断请求的发生源并且在允许输出中断请求的状态下产生中断请求，对应的 $IR_i.IR$ 标志就变为“1”。通过 $IER_i.IEN_j$ 位允许中断，向中断请求目标输出 $IR_i.IR$ 标志为“1”的中断请求；通过 $IER_i.IEN_j$ 位禁止中断，保留 $IR_i.IR$ 标志为“1”的中断请求。

$IR_i.IR$ 标志不受 $IER_i.IEN_j$ 位的影响。

10.4.2 中断状态标志

中断状态标志（ $IR_i.IR$ 标志）是检测中断并且保持中断请求的标志。

中断的检测方法有电平检测和边沿检测两种。

给外围模块的中断的各中断请求规定了边沿检测或者电平检测。

IRQ_n 引脚（ $n=0 \sim 15$ ）中断的检测方法能通过设定 $IRQCR_n.IRQMD[1:0]$ 位，转换边沿检测和电平检测。

中断请求与中断检测方法的对应请参照“表 10.4 中断向量表”。

10.4.2.1 边沿检测的中断状态标志

外围功能中断以及在进行外部中断的边沿检测时的 $IR_i.IR$ 标志的操作如图 10.2 所示。

在产生中断请求时的中断信号的变化位置，向中断请求目标输出 $IR_i.IR$ 标志为“1”的中断请求。如果中断请求目标接受中断请求， $IR_i.IR$ 标志就自动变为“0”，而不需要通过软件将 $IR_n.IR$ 标志置“0”。

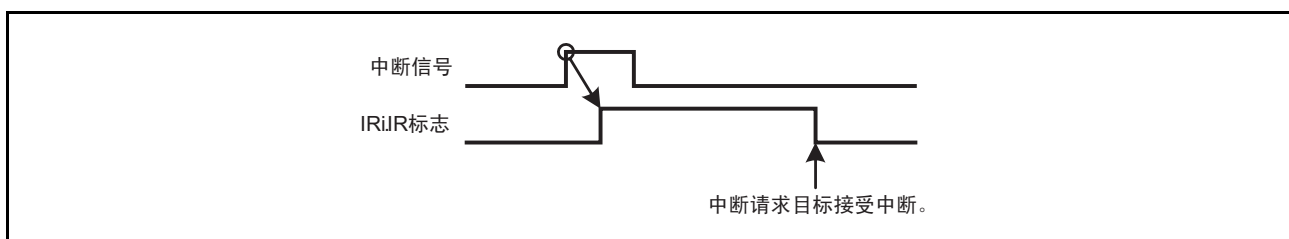


图 10.2 边沿检测的 $IR_i.IR$ 标志的操作

在产生中断请求并且 $IRi.IR$ 标志为“1”的状态下，忽视再次产生的中断请求。如果在 $IRi.IR$ 标志变为“0”后产生中断请求， $IRi.IR$ 标志就重新变为“1”。 $IRi.IR$ 标志被重新置位的时序如图 10.3 所示。

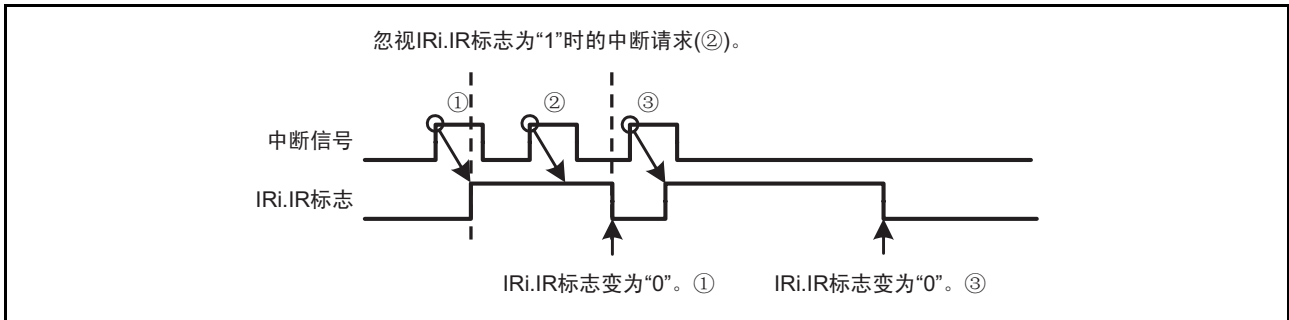


图 10.3 $IRi.IR$ 标志被重新置位的时序

在 $IRi.IR$ 标志变为“1”后，即使禁止中断（通过外围模块的中断允许位禁止输出中断请求，或者通过 $IRQERn.IRQEN$ 位禁止检测由 $IRQn$ 引脚引起的外部中断）， $IRi.IR$ 标志也不受影响而保持“1”的状态。禁止中断时的 $IRi.IR$ 标志的操作如图 10.4 所示。

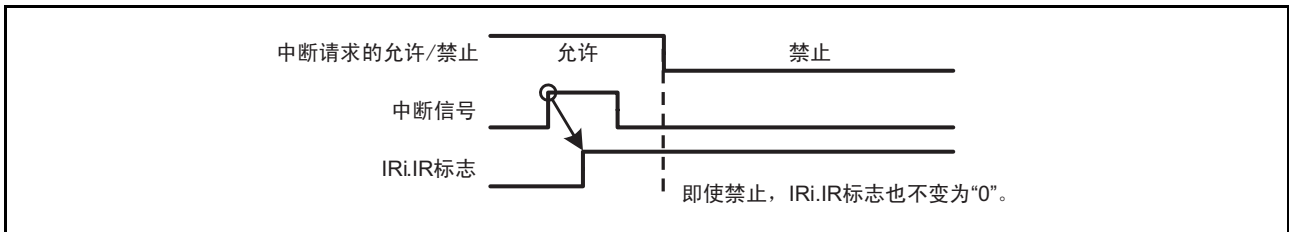


图 10.4 中断请求的禁止和 $IRi.IR$ 标志的关系

10.4.2.2 电平检测的中断状态标志

外围功能中断以及进行外部中断的电平检测时的运行如下所示。

在外围功能中断进行电平检测时的 $IRi.IR$ 标志的操作如图 10.5 所示。

在外围模块中断的情况下， $IRi.IR$ 标志在发生中断请求期间保持为“1”。要将 $IRi.IR$ 标志置“0”时，必须通过软件将中断请求发生源的状态标志置“0”，或者通过中断请求发生源的中断允许位禁止中断。在将中断请求发生源置“0”后，必须先确认 $IRi.IR$ 标志已变为“0”，然后执行下一条指令。

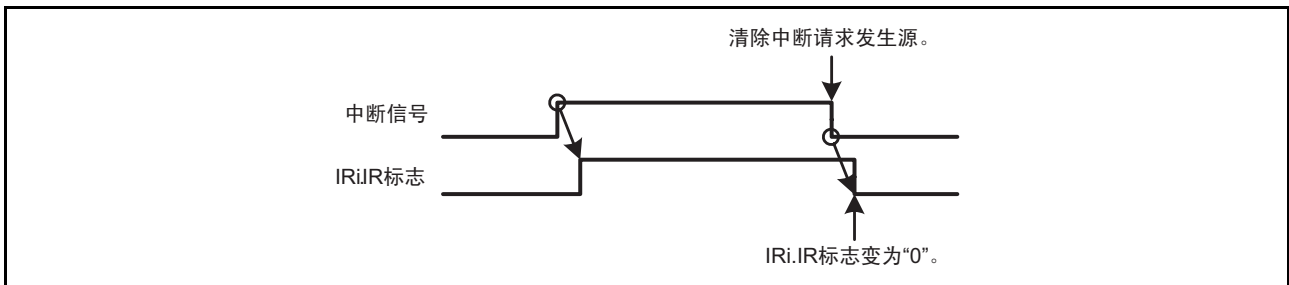


图 10.5 在外围功能中断进行电平检测时的 $IRi.IR$ 标志的操作

在外部中断进行电平检测时的 $IRi.IR$ 标志的操作如图 10.6 所示。

如果通过 $IRQCRn.IRQMD[1:0]$ 位将外部中断设定为 Low 电平检测，就必须使 $IRQn$ 引脚的输入中断请求信号保持 Low 电平直到开始进行该中断处理为止。然后，必须在中断异常处理程序内将该 $IRQn$ 引脚的输入恢复为 High 电平，并且在 4 个 PCLK 周期后将 $IRi.IR$ 标志置“0”。例如，能通过读 I/O 端口的 $Pm.PORT.Bj$ 位并且重复 2 次确认该 IRQ 引脚的 High 电平，至少确保 4 个 PCLK 周期。

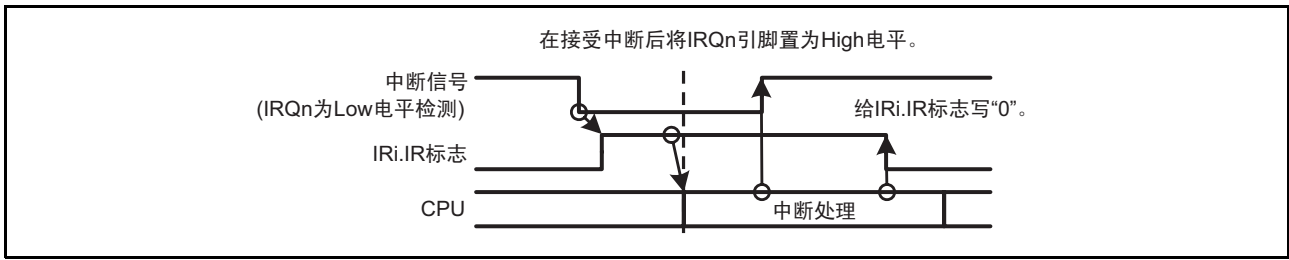


图 10.6 在外部中断进行电平检测时的 IRi.IR 标志的操作

如果在将 IRi.IR 标志置“1”后禁止中断（通过外围模块的中断允许位禁止输出中断请求，或者通过 IRQERn.IRQEN 位禁止检测由 IRQn 引脚引起的外部中断），IRi.IR 标志就变为“0”。禁止中断时的 IRi.IR 标志的操作如图 10.7 所示。

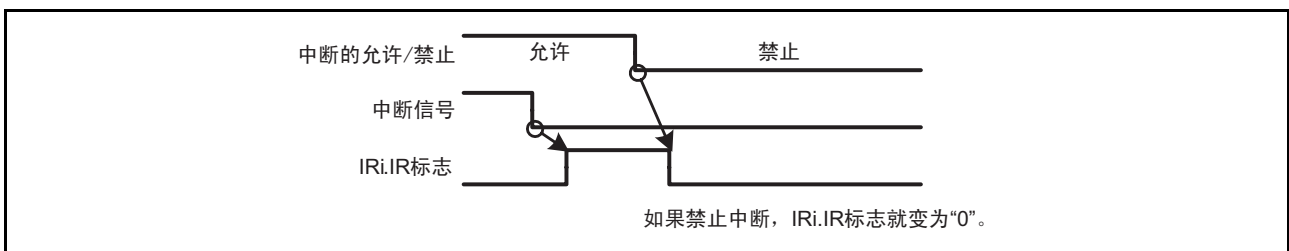


图 10.7 中断的禁止和 IRi.IR 标志的关系

10.4.3 中断请求目标的选择

能启动 DTC 和 DMAC 的中断源有给各中断源设定中断请求目标的 ISELRi.ISEL[1:0] 位，这些位决定了能各中断源设定的中断请求目标，可选择的中断请求目标如“表 10.4 中断向量表”所示。不能选择此表中没有记载的中断请求目标。

能通过 ISELRi.ISEL[1:0] 位选择的中断请求目标有以下 4 种：

- 向 CPU 请求的中断。
- 在启动 DTC 并且在数据传送结束后，向 CPU 请求的中断。
- 向 DMAC 请求的启动。
- 在启动 DMAC 后，向 CPU 请求的中断（不等到 DMA 数据的传送结束就向 CPU 请求中断）。

在启动 DTC 和 DMAC 时，需要使用边沿检测的中断检测方法。

在外围功能中断的情况下，只能选择边沿检测的中断请求。

在外部中断的情况下，必须通过 IRQCRn.IRQMD[1:0] 位设定为边沿检测。

如果将 ISELRi.ISEL[1:0] 位置“00b”，就向 CPU 输出中断请求。

如果将 ISELRi.ISEL[1:0] 位置“01b”，就向 DTC 输出启动请求，并且通过设定 DTC 的 MRB.DISEL 位进行以下的操作：

- 当 DTC 的 MRB.DISEL 为“0”时，DTC 进行所定次数的数据传送，在传送计数器变为“0”前 ISELRi.ISEL[1:0] 位保持“01b”，并且在传送计数器变为“0”时的数据传送结束后，ISELRi.ISEL[1:0] 位自动更新为“00b”。此时，向 CPU 输出中断请求。
- 当 DTC 的 MRB.DISEL 为“1”时，与 DTC 的传送计数器无关，在各数据传送结束后 ISELRi.ISEL[1:0] 位自动更新为“00b”。此时，向 CPU 输出中断请求。

如果在 ISELRi.ISEL[1:0] 位更新为“00b”后要重新置“01b”，就必须在该中断的异常处理程序内进行更改。如果要重新启动 DTC，就必须在将 ISELRi.ISEL[1:0] 位置“01b”后产生中断请求。

如果将 ISEL_{Ri}.ISEL[1:0] 位置 “10b”，就向 DMAC 输出启动请求。

在启动 DMAC 后，ISEL_{Ri}.ISEL[1:0] 位保持 “10b”。

当 DMA 传送方式为连续操作数传送时，或者当与 DMA 传送方式无关而通过相同的中断请求启动多个通道时，必须在以下的时序产生中断请求。

- 当 DMA 传送方式为连续操作数传送时，必须在启动通道的全部 DMA 传送结束后产生下一个中断请求（注1）。
- 当与 DMA 传送方式无关而通过相同中断请求启动多个通道时，必须在全部通道的传送结束后产生下一个中断请求（注2）。

- 注 1. 在单个操作数传送中，1 次启动请求进行 1 个操作数的传送。在启动 1 个操作数传送时，IR_i.IR 标志变为 “0”。在无停止传送中，1 次启动请求进行 1 次 DMA 传送。在启动 DMA 传送时，IR_i.IR 标志变为 “0”。在连续操作数传送中，1 次启动请求进行多个操作数的传送。在每次操作数传送时，IR_i.IR 标志变为 “0”。如果在全部操作数传送结束前产生中断请求，IR_i.IR 标志就在变为 “1” 后因各操作数的传送而变为 “0”。
- 注 2. 如果通过相同的中断请求启动 DMAC 的多个通道，1 次启动请求就启动所设定的全部通道，与 DMA 传送方式无关。在每次各通道的传送时，IR_i.IR 标志变为 “0”。如果在全部通道的全部 DMA 传送结束前产生中断请求，IR_i.IR 标志就在变为 “1” 后因各通道的传送而变为 “0”。

如果将 ISEL_{Ri}.ISEL[1:0] 位置 “11b”，就在启动 DMAC 后，不等到传送结束，ISEL_{Ri}.ISEL[1:0] 位就自动更新为 “00b”。此时，IR_i.IR 标志不变为 “0” 而向 CPU 输出中断请求。

如果在 ISEL_{Ri}.ISEL[1:0] 位更新为 “00b” 后要重新置 “11b”，就必须在以下的时序进行设定：

- 当 DMA 传送方式为单个操作数传送并且进行无停止传送时，必须在该中断的异常处理程序内进行设定。
- 当 DMA 传送方式为连续操作数传送时，必须在启动通道的全部 DMA 传送结束后进行设定（注1）。
- 当与 DMA 传送方式无关而通过相同中断源启动多个通道时，必须在全部通道的传送结束后进行设定（注2）。

如果要重新启动 DMAC，就必须在将 ISEL_{Ri}.ISEL[1:0] 位置 “11b” 后产生中断请求。

- 注 1. 在单个操作数传送中，1 次启动请求进行 1 个操作数的传送。在启动 1 个操作数传送时，ISEL_{Ri}.ISEL[1:0] 位自动更新为 “00b”。在无停止传送中，1 次启动请求进行 1 次 DMA 传送。在启动 DMA 传送时，ISEL_{Ri}.ISEL[1:0] 位自动更新为 “00b”。在连续操作数传送中，1 次启动请求进行多个操作数的传送。在每次操作数传送时，ISEL_{Ri}.ISEL[1:0] 位自动更新为 “00b”。如果在全部操作数传送结束前重新将 ISEL_{Ri}.ISEL[1:0] 位置 “11b”，这些位就因各操作数的传送而被更新为 “00b”。
- 注 2. 如果通过相同的中断源启动 DMAC 的多个通道，1 次启动请求就启动所设定的全部通道，与 DMA 传送方式无关。在各通道的传送时，ISEL_{Ri}.ISEL[1:0] 位自动更新为 “00b”。如果在全部通道的全部 DMA 传送结束前重新将 ISEL_{Ri}.ISEL[1:0] 位置 “11b”，这些位就因各通道的传送而被更新为 “00b”。

10.4.4 优先级的判断

中断控制器判断各中断请求目标的优先级。

如果对同一个中断请求目标产生多个中断请求，就接受优先级高的中断源。优先级的判断方法因各中断请求目标而不同。

(1) 中断请求目标为 CPU 时的优先级判断

在 ISEL_{Ri}.ISEL[1:0] 位为 “00b” 的中断组中，优先接受中断优先级选择位 (IPR_i.IPR[2:0]) 值大的请求。如果 IPR_i.IPR[2:0] 位的值为多个相同优先级的请求，就优先接受向量号小的请求。

如果发生被设定为高速中断的请求，就向 CPU 输出优先级最高 (7 级) 的中断请求，与 IPR_i.IPR[2:0] 位和向量号无关。

(2) 中断请求目标为 DTC 时的优先级判断

ISEL_{Ri}.ISEL[1:0] 位为 “01b” 的中断组不受 IPR_i.IPR[2:0] 位的影响，优先接受向量号小的请求。

(3) 中断请求目标为 DMAC 时的优先级判断

ISEL_{Ri}.ISEL[1:0] 位为 “10b” 或者 “11b” 的中断组不受 IPR_i.IPR[2:0] 位的影响，而取决于 DMAC 的设定，详细内容请参照 “12. DMA 控制器 (DMAC)”。

10.4.5 高速中断

高速中断是高速进行 CPU 中断处理的功能，只对 CPU 的中断请求有效，不影响 DTC 和 DMAC 的传送请求。

在通过 FIR.FVCT[7:0] 位给要设定为高速中断的中断源设定向量号并且通过 FIR.FIEN 位允许高速中断的状态下，如果发生对应的中断，就能作为高速中断输出到 CPU。

被设定为高速中断的中断源与 IPR_i.IPR[2:0] 位的设定无关，是优先级最高的中断源。如果 CPU 正在进行非屏蔽中断或者优先级为 “7 级” 的中断处理，就在处理结束后接受此高速中断。

高速中断的详细内容请参照 “9. 异常处理”。

10.4.6 外部中断

外部中断是 IRQ_n 引脚 (n=0 ~ 15) 引起的中断，外部中断的框图如图 10.8 所示。

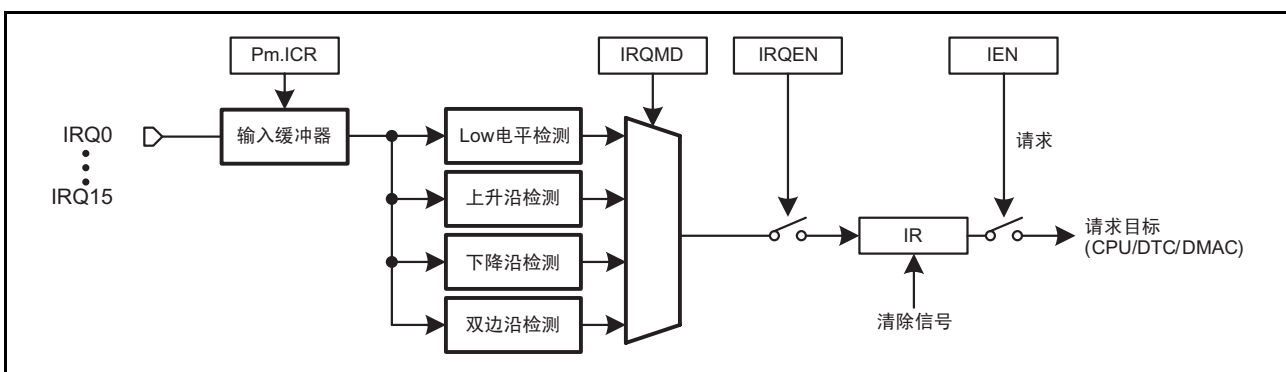


图 10.8 外部中断的框图

外部中断能通过 IRQCRn.IRQMD[1:0] 位选择电平检测 (Low) 或者边沿检测 (下降沿、上升沿、双边沿) 的中断检测方法。

必须按照以下的步骤设定 IRQCRn.IRQMD[1:0] 位:

1. 通过 IRQERn.IRQEN 位和 IERi.IENj 位禁止外部中断的检测以及中断的发生。
2. 设定 IRQCRn.IRQMD[1:0] 位。
3. 将 IRn.IR 标志置 “0”。
4. 通过 IRQERn.IRQEN 位和 IERi.IENj 位允许外部中断的检测以及中断的发生。

外部中断的边沿检测和电平检测的运行请参照 “10.4.2 中断状态标志”。

在使用外部中断时, 必须通过 Pm.ICR.Bj 位将 IRQn 引脚的输入缓冲器设定为有效。

必须按照以下的步骤更改 Pm.ICR.Bj 位的设定:

1. 通过 IRQERn.IRQEN 位和 IERi.IENj 位禁止外部中断的检测以及中断的发生。
2. 更改 Pm.ICR.Bj 位的设定。
3. 在 4 个 PCLK 周期后, 将 IRi.IR 标志置 “0”。
4. 通过 IRQERn.IRQEN 位和 IERi.IENj 位允许外部中断的检测以及中断的发生。

IRQn 引脚与 Pm.ICR.Bj 位的对应请参照 “14. I/O 端口”。

10.5 非屏蔽中断

NMI 引脚中断通过给 NMI 引脚输入下降沿或者上升沿, 向 CPU 请求非屏蔽中断, 但是不能启动 DTC 和 DMAC。在包括高速中断在内的所有中断中, NMI 引脚中断是最优先的中断。

如果检测到 NMI 引脚中断, NMISR.NMIST 标志就变为 “1”, 并且将非屏蔽中断请求输出到 CPU。与 CPU 的 PSW.I 位 (中断允许位) 和 PSW.IPL[2:0] 位 (处理器中断优先级) 的状态无关, 总是接受此非屏蔽中断请求。

要将 NMISR.NMIST 标志置 “0” 时, 必须给 NMICLR.NMICLR 位写 “1”。此后, 必须在确认 NMISR.NMIST 标志已变为 “0” 后执行下一条指令。

为了避免在不需要 NMI 引脚中断的系统发生误动作, 必须将初始状态设定为禁止 NMI 引脚中断。在使用 NMI 引脚中断的系统中, 必须在全部程序处理的起始位置按照以下的步骤进行设定。

NMI 引脚中断的使用步骤:

1. 设定堆栈指针 (SP)。
2. 设定 NMI 引脚的检测方法 (NMICR.NMIMD 位)。
3. 在给 NMICLR.NMICLR 位写 “1”, NMISR.NMIST 标志写 “0” 后, 确认 NMISR.NMIST 标志已变为 “0”。
4. 给 NMIER.NMIEN 位写 “1”, 允许使用 NMI 引脚中断。

如果给 NMIER.NMIEN 位写 “1”, 就忽视以后的 NMIER.NMIEN 位的写操作而不能禁止 NMI 引脚中断。此功能是为了避免在使用 NMI 引脚中断过程中因程序失控而导致意想不到的 “禁止 NMI 引脚中断”。非屏蔽中断的处理流程请参照 “9. 异常处理”。

10.6 从低功耗状态的返回

有时能通过中断源从低功耗状态返回。

低功耗状态与能用作返回源的中断源的对应如表 10.5 所示。

表 10.5 返回源一览表

| 低功耗状态 | 能用于返回的中断源 | 中断控制器的时钟 |
|-----------|---|----------|
| 睡眠模式 | 包括 NMI 引脚中断在内的全部中断 | 运行 |
| 全模块时钟停止模式 | 外围功能中断 (WDT、TMR (注1))、 外部中断、NMI 引脚中断 | 运行 |
| 软件待机模式 | 外部中断、NMI 引脚中断 | 停止 |

注 1. 详细内容请参照“8. 低功耗功能”。

10.6.1 从睡眠模式和全模块时钟停止模式的返回

能通过全部中断从睡眠模式和全模块时钟停止模式返回。

从睡眠模式和全模块时钟停止模式返回的条件如下：

1. 中断请求目标为 CPU。
2. 通过 IERi.IENj 位允许对应的中断。
3. 优先级高于 CPU 的 PSW.IPL[2:0] 位的优先级。

10.6.2 从软件待机模式的返回

能通过 NMI 引脚中断和 IRQn 引脚 (n=0 ~ 15) 的外部中断从软件待机模式返回。

如果发生 NMI 引脚中断或者 IRQn 引脚 (n=0 ~ 15) 的外部中断，就给整个 LSI 提供已经停止运行的时钟并且开始中断处理。

要通过 NMI 引脚中断从软件待机模式返回时，必须将 NMIE.NMIEN 位置“1”（允许 NMI 引脚中断）。

通过 IRQn 引脚 (n=0 ~ 15) 的外部中断从软件待机模式返回的条件如下所示：

1. 通过 SSIER.SSIj 位允许其作为返回源。
2. 通过 IRQERn.IRQEN 位允许检测对应的 IRQn 引脚的外部中断。
3. 通过 IERi.IENj 位允许对应的 IRQn 引脚的中断。
4. 通过 ISELr.ISEL[1:0] 位将中断请求目标指定为 CPU。
5. IPRi.IPR[2:0] 位的中断优先级高于 CPU 的 PSW.IPL[2:0] 位的中断优先级 (注1)。

注 1. 与 IPRi.IPR[2:0] 位的设定无关，被设定为高速中断的中断源最优先。但是，如果将被设定为高速中断的中断源作为软件待机返回源，就需要根据此条件设定 IPRi.IPR[2:0] 位。

如果通过被设定为边沿检测的 IRQ0 ~ IRQ15 中断解除软件待机模式，就必须在作为解除源的中断异常处理程序的起始位置将该中断的状态标志 (ICU.IRi.IR) 置“0”。

在不将 IRQ0 ~ IRQ15 中断设定为软件待机模式的解除源的情况下，因为在软件待机模式中对引脚的输入缓冲器无效，所以 LSI 内部的输入信号被固定为 High 电平。因此，中断状态标志 (ICU.IRi.IR) 有可能因引脚状态而变为“1”。在向软件待机模式转移时，必须先通过 IERi.IENj 位禁止不作为解除源的 IRQ0 ~ IRQ15 中断或者通过 IPRi.IPR 位降低中断优先级，然后执行 WAIT 指令。另外，从软件待机模式返回后，必须清除中断状态标志。

低功耗状态的详细内容请参照“8. 低功耗功能”。

10.7 使用注意事项

10.7.1 写中断控制器的寄存器时的注意事项

在从 CPU 写中断控制器的寄存器时，因为 CPU 不等到写操作结束就执行后续的指令，所以有可能在写入值被保存到寄存器前就执行后续的指令。为了避免此操作，必须确认已经写完寄存器。如果连续写多个寄存器，就必须确认最后的写操作已经结束。有关写操作结束的确认方法，请参照“5. I/O 寄存器”。

例如，向软件待机状态转移前先禁止与返回无关的中断的情况，给 IERi.IEj 位写“0”并且执行 WAIT 指令。此时，如果不等到先执行的写操作结束就向软件待机状态转移，就有可能因发生应该被禁止的中断而返回，因此必须在执行 WAIT 指令前确认已经写完 IERi.IEj 位。

10.7.2 使用 NMI 引脚中断时的 WAIT 指令的注意事项

不能在 NMISR.NMIST 标志为“1”的状态下执行 WAIT 指令。必须在执行 WAIT 指令前确认 NMISR.NMIST 标志为“0”。

11. 总线

11.1 概要

总线的规格和结构图分别如表 11.1 和图 11.1 所示。

表 11.1 总线的规格

| 总线的种类 | | 内容 |
|--------|----------|--|
| CPU 总线 | 指令总线 | <ul style="list-style-type: none"> • 连接 CPU（指令）。 • 连接内部存储器（内部 RAM 和内部 ROM）。 • 与系统时钟（ICLK）同步运行。 |
| | 操作数总线 | <ul style="list-style-type: none"> • 连接 CPU（操作数）。 • 连接内部存储器（内部 RAM 和内部 ROM）。 • 与系统时钟（ICLK）同步运行。 |
| 内部主总线 | 内部主总线 1 | <ul style="list-style-type: none"> • 连接 CPU。 • 与系统时钟（ICLK）同步运行。 |
| | 内部主总线 2 | <ul style="list-style-type: none"> • 连接 DMAC 和 DTC。 • 连接内部存储器（内部 RAM 和内部 ROM）。 • 与系统时钟（ICLK）同步运行。 |
| 内部外围总线 | 内部外围总线 1 | <ul style="list-style-type: none"> • 连接外围功能。 • 与系统时钟（ICLK）同步运行。 |
| | 内部外围总线 2 | <ul style="list-style-type: none"> • 连接外围功能、内部 ROM（P/E）/ 数据闪存。 • 与外围模块时钟（PCLK）同步运行。 |
| 外部总线 | | <ul style="list-style-type: none"> • 连接外部设备。 • 与外部总线时钟（BCLK）同步运行。 |

P/E: 编程 / 擦除

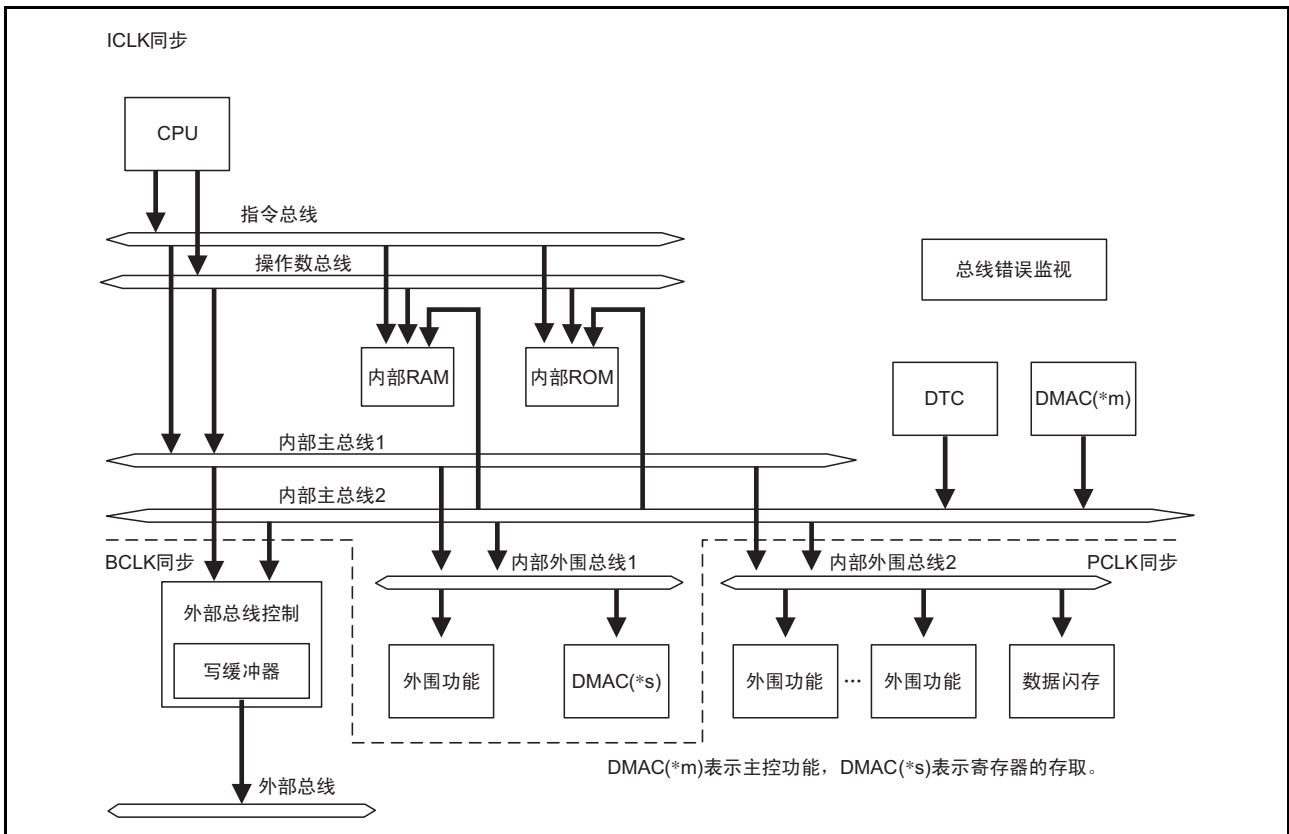


图 11.1 总线的结构图

11.2 总线说明

11.2.1 CPU 总线

CPU 总线有指令总线和操作数总线，连接内部主总线 1。指令总线用于 CPU 的取指令，操作数总线用于存取操作数。

指令总线和操作数总线连接内部 RAM 和内部 ROM，能不通过内部主总线 1 而被 CPU 直接存取。内部 ROM 为只读区域，能被 CPU 直接存取，但是编程和擦除必须通过内部外围总线进行。

对内部 RAM、内部 ROM 和内部主总线 1 的取指令和操作数的总线权请求进行仲裁，优先级为操作数 > 取指令。

如果指令总线、操作数总线 1 和内部主总线的请求是不同的从属模块，就能同时进行各自的总线存取。例如，内部 ROM 和内部 RAM、内部 ROM 和外部存取能并行运行。

11.2.2 内部主总线

内部主总线由 CPU 使用的总线（内部主总线 1）以及除 CPU 以外的总线主控（DMAC 和 DTC）使用的总线（内部主总线 2）构成。通过内部主总线 2 对 DMAC 和 DTC 的总线权请求进行仲裁。优先级为 DMAC > DTC，如表 11.2 所示。

对外部总线、外围总线和内部存储器的各从属模块的 2 条内部主总线的总线权请求进行仲裁。CPU 和除 CPU 以外的总线主控能同时对不同的从属模块进行总线存取。

2 条总线的优先级是内部主总线 1（CPU）的优先级低于内部主总线 2（除 CPU 以外的总线主控）的优先级。如果通过 CPU 执行 XCHG 指令，就在 XCHG 指令进行的总线存取结束前不接受 CPU 以外的总线存取，而且在读以及回写 DTC 的传送信息过程中也不接受 DTC 以外的总线存取。

表 11.2 总线主控的优先级

| 优先级 | 总线主控 |
|------------------|--------|
| 高 ↑ ↓ 低 | DMA 目标 |
| | DMA 源 |
| | DTC |
| | CPU |

11.2.3 内部外围总线

连接内部外围总线的外围功能如表 11.3 所示。

表 11.3 连接内部外围总线的外围功能

| 总线的种类 | 外围功能 |
|----------|---|
| 内部外围总线 1 | <ul style="list-style-type: none"> • DMAC • 中断控制器 |
| 内部外围总线 2 | <ul style="list-style-type: none"> • 内部外围总线 1 以外的外围功能 |

11.2.4 外部总线

外部总线的规格如表 11.4 所示。

表 11.4 外部总线的规格

| 项目 | 内容 |
|--------|--|
| 外部地址空间 | <ul style="list-style-type: none"> • 将外部地址空间分成 8 个区域（CS0 ~ CS7）进行管理。 • 各区域可输出片选。 • 各区域可选择 8 位总线空间或者 16 位总线空间。 • 各区域能设定字节序。 |
| 等待控制功能 | <ul style="list-style-type: none"> • 可插入恢复周期。 最多插入 15 个读恢复周期和 15 个写恢复周期。 • 周期等待功能：最多等待 31 个周期（最多等待 7 个页面存取周期）。 • 等待控制 能设定片选信号（CS0# ~ CS7#）的有效或者无效时序。 能设定读信号（RD#）和写信号（WR0#、WR1#、WR#）的有效时序。 能设定数据输出的开始时序和结束时序。 • 写存取模式：1 次写选通模式 / 字节选通模式 |
| 写缓冲器功能 | <ul style="list-style-type: none"> • 在将总线主控的写数据写到写缓冲器时，结束总线主控侧的写存取。 |
| 频率 | <ul style="list-style-type: none"> • 与外部总线时钟（BCLK）同步运行。 |

外部总线的输入 / 输出引脚如表 11.5 所示。

表 11.5 外部总线的输入 / 输出引脚

| 引脚名 | 输入 / 输出 | 功能 |
|----------------|---------|--|
| A23 ~ A0 (注 1) | 输出 | 地址的输出引脚 |
| BC0# (注 1) | 输出 | 选通信号 表示在 1 次写选通模式中正在存取外部地址空间 (BC0# 信号为 Low 电平) 并且低位字节 (D7 ~ D0) 有效。当设定 8 位总线空间时, 总是输出 Low 电平, 与写存取模式无关。 |
| BC1# | 输出 | 选通信号 表示在 1 次写选通模式中正在存取外部地址空间 (BC1# 信号为 Low 电平) 并且高位字节 (D15 ~ D8) 有效。当设定 8 位总线空间时, 不使用此引脚。 |
| D15 ~ D0 | 输入 / 输出 | 数据的输入 / 输出引脚 当设定 16 位总线空间时, D15 ~ D0 有效; 当设定 8 位总线空间时, D7 ~ D0 有效。 |
| CS0# | 输出 | 表示已选择区域 0 (CS0) 的选通信号 |
| CS1# | 输出 | 表示已选择区域 1 (CS1) 的选通信号 |
| CS2# | 输出 | 表示已选择区域 2 (CS2) 的选通信号 |
| CS3# | 输出 | 表示已选择区域 3 (CS3) 的选通信号 |
| CS4# | 输出 | 表示已选择区域 4 (CS4) 的选通信号 |
| CS5# | 输出 | 表示已选择区域 5 (CS5) 的选通信号 |
| CS6# | 输出 | 表示已选择区域 6 (CS6) 的选通信号 |
| CS7# | 输出 | 表示已选择区域 7 (CS7) 的选通信号 |
| RD# | 输出 | 表示正在读外部地址空间的选通信号 |
| WR0# | 输出 | 选通信号 表示在字节选通模式中正在写外部地址空间 (WR0# 信号为 Low 电平) 并且低位字节 (D7 ~ D0) 有效。在 1 次写选通模式中正在写外部地址空间。当设定 8 位总线空间时, 总是在写存取时输出 Low 电平, 与写存取模式无关。 |
| WR1# | 输出 | 选通信号 表示在字节选通模式中正在写外部地址空间 (WR1# 信号为 Low 电平) 并且高位字节 (D15 ~ D8) 有效。在 1 次写选通模式中, 输出 BC1# 信号。当设定 8 位总线空间时, 不使用此引脚。 |
| WR# | 输出 | 选通信号 表示在 1 次写选通模式中正在写外部地址空间。 |
| WAIT# | 输入 | 存取外部地址空间时的等待请求信号 (Low: 等待请求) |

注 1. A0 引脚与 BC0# 引脚兼用。在字节写模式中, 各区域的 A0 引脚有效; 在 1 次写选通模式中, 各区域的 BC0# 引脚有效。有关其他的兼用引脚, 请参照“14. I/O 端口”。

11.2.5 并行运行

能在各总线主控存取不同的从属模块时并行运行。例如，CPU 能同时进行 3 种运行，一边从内部 ROM 取指令，一边对内部 RAM 进行操作数存取，同时，DMAC 还能进行外围总线和外部总线之间的传送。

并行运行的例子如图 11.2 所示。在此例中，CPU 能使用指令总线和操作数总线同时存取内部 ROM 和内部 RAM，而且在 CPU 存取内部 ROM 和内部 RAM 过程中，DMAC 能使用内部主总线 2 同时存取外围总线或者外部总线。

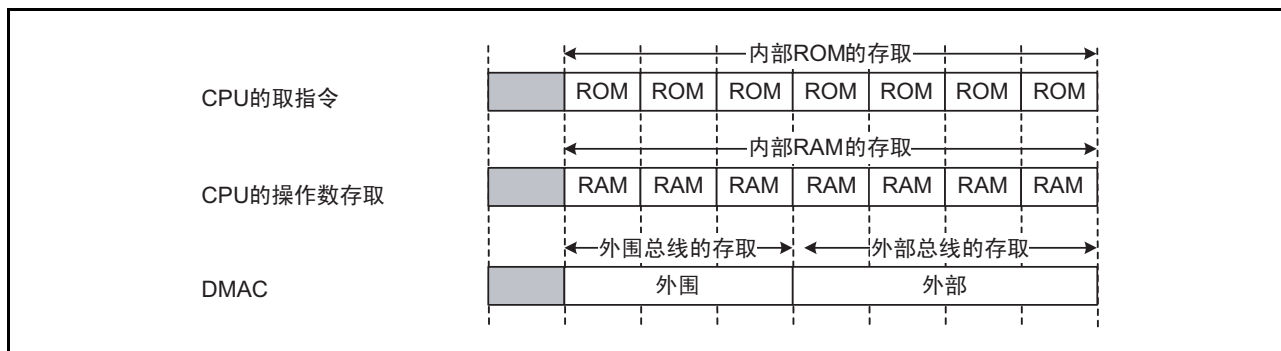


图 11.2 并行运行的例子

11.3 寄存器说明

外部总线控制部的寄存器一览表如表 11.6 所示。

表 11.6 外部总线控制部的寄存器一览表

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|---------------|----------|------------|------------|------|
| CS0 控制寄存器 | CS0CNT | 0021h | 0008 3802h | 16 |
| CS0 恢复周期设定寄存器 | CS0REC | 0000h | 0008 380Ah | 16 |
| CS1 控制寄存器 | CS1CNT | 0000h | 0008 3812h | 16 |
| CS1 恢复周期设定寄存器 | CS1REC | 0000h | 0008 381Ah | 16 |
| CS2 控制寄存器 | CS2CNT | 0000h | 0008 3822h | 16 |
| CS2 恢复周期设定寄存器 | CS2REC | 0000h | 0008 382Ah | 16 |
| CS3 控制寄存器 | CS3CNT | 0000h | 0008 3832h | 16 |
| CS3 恢复周期设定寄存器 | CS3REC | 0000h | 0008 383Ah | 16 |
| CS4 控制寄存器 | CS4CNT | 0000h | 0008 3842h | 16 |
| CS4 恢复周期设定寄存器 | CS4REC | 0000h | 0008 384Ah | 16 |
| CS5 控制寄存器 | CS5CNT | 0000h | 0008 3852h | 16 |
| CS5 恢复周期设定寄存器 | CS5REC | 0000h | 0008 385Ah | 16 |
| CS6 控制寄存器 | CS6CNT | 0000h | 0008 3862h | 16 |
| CS6 恢复周期设定寄存器 | CS6REC | 0000h | 0008 386Ah | 16 |
| CS7 控制寄存器 | CS7CNT | 0000h | 0008 3872h | 16 |
| CS7 恢复周期设定寄存器 | CS7REC | 0000h | 0008 387Ah | 16 |
| CS0 模式寄存器 | CS0MOD | 0000h | 0008 3002h | 16 |
| CS0 等待控制寄存器 1 | CS0WCNT1 | 0707 0707h | 0008 3004h | 32 |
| CS0 等待控制寄存器 2 | CS0WCNT2 | 0000 0007h | 0008 3008h | 32 |
| CS1 模式寄存器 | CS1MOD | 0000h | 0008 3012h | 16 |
| CS1 控制寄存器 1 | CS1WCNT1 | 0707 0707h | 0008 3014h | 32 |
| CS1 等待控制寄存器 2 | CS1WCNT2 | 0000 0007h | 0008 3018h | 32 |
| CS2 模式寄存器 | CS2MOD | 0000h | 0008 3022h | 16 |
| CS2 等待控制寄存器 1 | CS2WCNT1 | 0707 0707h | 0008 3024h | 32 |
| CS2 等待控制寄存器 2 | CS2WCNT2 | 0000 0007h | 0008 3028h | 32 |
| CS3 模式寄存器 | CS3MOD | 0000h | 0008 3032h | 16 |
| CS3 等待控制寄存器 1 | CS3WCNT1 | 0707 0707h | 0008 3034h | 32 |
| CS3 等待控制寄存器 2 | CS3WCNT2 | 0000 0007h | 0008 3038h | 32 |
| CS4 模式寄存器 | CS4MOD | 0000h | 0008 3042h | 16 |
| CS4 等待控制寄存器 1 | CS4WCNT1 | 0707 0707h | 0008 3044h | 32 |
| CS4 等待控制寄存器 2 | CS4WCNT2 | 0000 0007h | 0008 3048h | 32 |
| CS5 模式寄存器 | CS5MOD | 0000h | 0008 3052h | 16 |
| CS5 等待控制寄存器 1 | CS5WCNT1 | 0707 0707h | 0008 3054h | 32 |
| CS5 等待控制寄存器 2 | CS5WCNT2 | 0000 0007h | 0008 3058h | 32 |
| CS6 模式寄存器 | CS6MOD | 0000h | 0008 3062h | 16 |
| CS6 等待控制寄存器 1 | CS6WCNT1 | 0707 0707h | 0008 3064h | 32 |
| CS6 等待控制寄存器 2 | CS6WCNT2 | 0000 0007h | 0008 3068h | 32 |
| CS7 模式寄存器 | CS7MOD | 0000h | 0008 3072h | 16 |
| CS7 等待控制寄存器 1 | CS7WCNT1 | 0707 0707h | 0008 3074h | 32 |
| CS7 等待控制寄存器 2 | CS7WCNT2 | 0000 0007h | 0008 3078h | 32 |
| 总线错误源清除寄存器 | BERCLR | 00h | 0008 1300h | 8 |
| 总线错误监视允许寄存器 | BEREN | 00h | 0008 1304h | 8 |
| 总线错误中断允许寄存器 | BERIE | 00h | 0008 1306h | 8 |

11.3.1 CSi 控制寄存器 (CSiCNT) (i=0 ~ 7)

地址 0008 3802h (CS0CNT)

| | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|-------|----|----|------------|----|----|----|----|-------|
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | — | — | — | — | — | EMODE | — | — | BSIZE[1:0] | | — | — | — | EXENB |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

地址 0008 3812h~0008 3872h (CS1CNT ~ CS7CNT)

| | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|-------|----|----|------------|----|----|----|----|-------|
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | — | — | — | — | — | EMODE | — | — | BSIZE[1:0] | | — | — | — | EXENB |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|--------|------------|-----------|--|--------------|
| b0 | EXENB | 运行允许位 | 0: 禁止运行 1: 允许运行 | R/W (注 1) |
| b3-b1 | — | 保留位 | 读写值都为“0”。 | R/W |
| b5-b4 | BSIZE[1:0] | 外部总线宽度选择位 | b5 b4 00: 设定为 16 位总线空间 01: 不能设定 10: 设定为 8 位总线空间 11: 不能设定 | R/W (注 2) |
| b7-b6 | — | 保留位 | 读写值都为“0”。 | R/W |
| b8 | EMODE | 字节序模式指定位 | 0: 区域 i 的字节序模式和运行模式的字节序相同 (i=0 ~ 7) 1: 区域 i 的字节序模式和运行模式的字节序不同 (i=0 ~ 7) | R/W |
| b15-b9 | — | 保留位 | 读写值都为“0”。 | R/W |

注 1. CS0CNT.EXENB 位复位后的值为“1”，CSiCNT.EXENB 位 (i=1 ~ 7) 复位后的值为“0”。

注 2. CS0CNT.BSIZE[1:0] 位复位后的值为“10”。

CSiCNT 寄存器是允许外部地址空间各区域的运行并且设定数据总线宽度和字节序的寄存器。

EXENB 位 (运行允许位)

此位设定允许或者禁止各区域的运行。

复位后，只允许区域 0 (“1”) 的运行，禁止其他区域 (“0”) 的运行。

如果存取被设定为禁止运行的区域，就不发生外部总线存取。此时，如果将总线错误监视控制寄存器的非法地址存取检测允许位 (BEREN.IGAEN=1) 设定为允许检测，就发生非法地址错误。

BSIZE[1:0] 位 (外部总线宽度选择位)

这些位设定各区域的数据总线宽度。

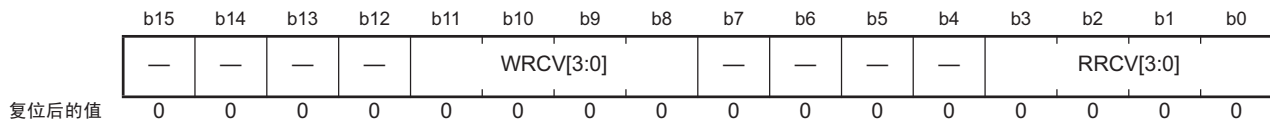
复位后，区域 0 的数据总线宽度为 8 位总线空间。

EMODE 位 (字节序模式指定位)

此位设定各区域的字节序。

11.3.2 CSi 恢复周期设定寄存器 (CSiREC) (i=0 ~ 7)

地址 0008 380Ah ~ 0008 387Ah



| 位 | 符号 | 位名 | 功能 | R/W |
|---------|-----------|----------|--|-----|
| b3-b0 | RRCV[3:0] | 读恢复周期设定位 | b3 b0 0 0 0 0: 不插入恢复周期 0 0 0 1: 插入 1 个恢复周期 0 0 1 0: 插入 2 个恢复周期 0 0 1 1: 插入 3 个恢复周期 0 1 0 0: 插入 4 个恢复周期 0 1 0 1: 插入 5 个恢复周期 0 1 1 0: 插入 6 个恢复周期 0 1 1 1: 插入 7 个恢复周期 1 0 0 0: 插入 8 个恢复周期 1 0 0 1: 插入 9 个恢复周期 1 0 1 0: 插入 10 个恢复周期 1 0 1 1: 插入 11 个恢复周期 1 1 0 0: 插入 12 个恢复周期 1 1 0 1: 插入 13 个恢复周期 1 1 1 0: 插入 14 个恢复周期 1 1 1 1: 插入 15 个恢复周期 | R/W |
| b7-b4 | — | 保留位 | 读写值都为“0”。 | R/W |
| b11-b8 | WRCV[3:0] | 写恢复周期设定位 | b11 b8 0 0 0 0: 不插入恢复周期 0 0 0 1: 插入 1 个恢复周期 0 0 1 0: 插入 2 个恢复周期 0 0 1 1: 插入 3 个恢复周期 0 1 0 0: 插入 4 个恢复周期 0 1 0 1: 插入 5 个恢复周期 0 1 1 0: 插入 6 个恢复周期 0 1 1 1: 插入 7 个恢复周期 1 0 0 0: 插入 8 个恢复周期 1 0 0 1: 插入 9 个恢复周期 1 0 1 0: 插入 10 个恢复周期 1 0 1 1: 插入 11 个恢复周期 1 1 0 0: 插入 12 个恢复周期 1 1 0 1: 插入 13 个恢复周期 1 1 1 0: 插入 14 个恢复周期 1 1 1 1: 插入 15 个恢复周期 | R/W |
| b15-b12 | — | 保留位 | 读写值都为“0”。 | R/W |

CSiREC 寄存器设定外部地址空间各区域的读写存取后的恢复周期数。

RRCV[3:0] 位（读恢复周期设定位）

这些位设定在对外部总线进行读存取后要插入的恢复周期数。

如果设定“0000b”以外的值，就在以下情况下插入 1 ~ 15 个恢复周期。

- 在对外部总线进行读存取后有外部总线的写存取时
（即使在相同区域内进行连续存取，也插入恢复周期。）
- 在对外部总线进行读存取后有不同区域的读存取时
（在相同区域内进行连续存取时，不插入恢复周期。）

WRCV[3:0] 位（写恢复周期设定位）

这些位设定在对外部总线进行写存取后要插入的恢复周期数。

如果设定“0000b”以外的值，就在以下情况下插入 1 ~ 15 个恢复周期。

- 在对外部总线进行写存取后有外部总线的读存取时
（即使在相同区域内进行连续存取，也插入恢复周期。）

在写存取后的写存取时不插入恢复周期。

表 11.7 恢复周期的插入

| 存取的种类 | 外部地址空间 | 恢复周期的插入 |
|----------|--------|-----------------------|
| 写存取后的读存取 | 相同区域 | 插入 WRCV[3:0] 位设定的周期数。 |
| | 不同区域 | 插入 WRCV[3:0] 位设定的周期数。 |
| 写存取后的写存取 | 相同区域 | 不插入恢复周期。 |
| | 不同区域 | 不插入恢复周期。 |
| 读存取后的写存取 | 相同区域 | 插入 RRCV[3:0] 位设定的周期数。 |
| | 不同区域 | 插入 RRCV[3:0] 位设定的周期数。 |
| 读存取后的读存取 | 相同区域 | 不插入恢复周期。 |
| | 不同区域 | 插入 RRCV[3:0] 位设定的周期数。 |

11.3.3 CSi 模式寄存器 (CSiMOD) (i=0 ~ 7)

地址 0008 3002h ~ 0008 3872h

| | | | | | | | | | | | | | | | | |
|-------|-------|-----|-----|-----|-----|-----|-------|-------|----|----|----|----|-------|----|----|-------|
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | PRMOD | — | — | — | — | — | PWENB | PRENB | — | — | — | — | EWENB | — | — | WRMOD |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|-------|------------|----------------------------|-----|
| b0 | WRMOD | 写存取模式选择位 | 0: 字节选通模式 1: 1次写选通模式 | R/W |
| b2-b1 | — | 保留位 | 读写值都为“0”。 | R/W |
| b3 | EWENB | 外部等待允许位 | 0: 禁止外部等待 1: 允许外部等待 | R/W |
| b7-b4 | — | 保留位 | 读写值都为“0”。 | R/W |
| b8 | PRENB | 页面读存取允许位 | 0: 禁止页面的读存取 1: 允许页面的读存取 | R/W |
| b9 | PWENB | 页面写存取允许位 | 0: 禁止页面的写存取 1: 允许页面的写存取 | R/W |
| b14-b10 | — | 保留位 | 读写值都为“0”。 | R/W |
| b15 | PRMOD | 页面读存取模式选择位 | 0: 正常存取兼容模式 1: 读连续有效模式 | R/W |

CSiMOD 寄存器是设定外部地址空间各区域的存取模式的寄存器。

WRMOD 位 (写存取模式选择位)

此位选择写存取时的运行模式。

如果将此位置“0”，就为字节选通模式。此时，通过与各字节位置对应的 WR0# 信号、WR1# 信号和 WR# 信号控制数据的写操作。

如果将此位置“1”，就为 1 次写选通模式。此时，通过与各字节位置对应的 BCn# 信号 (n=1,0)、WR0# 信号、WR1# 信号和 WR# 信号控制数据的写操作。

表 11.8 写存取模式的控制信号

| 写存取模式 | 数据写信号 | | | 字节控制信号 | |
|---------|-------|------|-----|--------|------|
| | WR0# | WR1# | WR# | BC0# | BC1# |
| 字节选通模式 | ○ | ○ | × | × | × |
| 1次写选通模式 | ○ | × | ○ | ○ | ○ |

○: 有效 ×: 无效

EWENB 位 (外部等待允许位)

此位设定允许或者禁止外部等待。

如果将此位置“1”，就允许外部等待，能通过 WAIT# 信号任意控制各周期的等待数。此时，在 WAIT# 信号为 Low 电平期间，插入等待周期。

如果将此位置“0”，WAIT# 信号就无效。

PRENB 位（页面读存取允许位）

此位设定允许或者禁止页面的读存取。

PWENB 位（页面写存取允许位）

此位设定允许或者禁止页面的写存取。

PRMOD 位（页面读存取模式选择位）

此位选择页面读存取时的运行模式。

如果将此位置“0”，就为正常存取兼容模式。此时，在每读1个数据时，RD#信号变为无效，并且插入RD有效等待。但是在RD有效等待为“0”并且不进行外部总线存取的最后传送时，RD#信号有效。

如果将此位置“1”，就为读连续有效模式。此时，插入RD有效等待，但是在此期间，RD#信号持续有效。

11.3.4 CSi 等待控制寄存器 1 (CSiWCNT1) (i=0 ~ 7)

地址 0008 3004h ~ 0008 3074h

| | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|--------------|-----|---------------|-----|-----|-----|-----|--------------|-----|-----|---------------|-----|-----|
| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| | — | — | — | CSRWAIT[4:0] | | | | — | — | — | CSWWAIT[4:0] | | | | | |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | — | — | — | CSPRWAIT[2:0] | | | — | — | — | — | — | CSPWWAIT[2:0] | | |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|-------------------|------------------|---|-----|
| b2-b0 | CSPWWAIT [2:0] | 页面写周期的等待选择位 (注1) | b2 b0 000: 不插入等待 001: 插入1个等待周期 010: 插入2个等待周期 011: 插入3个等待周期 100: 插入4个等待周期 101: 插入5个等待周期 110: 插入6个等待周期 111: 插入7个等待周期 | R/W |
| b7-b3 | — | 保留位 | 读写值都为“0”。 | R/W |
| b10-b8 | CSPRWAIT [2:0] | 页面读周期的等待选择位 (注2) | b10 b8 000: 不插入等待 001: 插入1个等待周期 010: 插入2个等待周期 011: 插入3个等待周期 100: 插入4个等待周期 101: 插入5个等待周期 110: 插入6个等待周期 111: 插入7个等待周期 | R/W |
| b15-b11 | — | 保留位 | 读写值都为“0”。 | R/W |

注1. CSPWWAIT[2:0] 位的设定值只在 CSiMOD.PWENB 位为“1”时有效。

注2. CSPRWAIT[2:0] 位的设定值只在 CSiMOD.PRENB 位为“1”时有效。

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|------------------|-------------|---|-----|
| b20-b16 | CSWWAIT [4:0] | 正常写周期的等待选择位 | b20 b16 00000: 不插入等待 00001: 插入1个等待周期 00010: 插入2个等待周期 00011: 插入3个等待周期 00100: 插入4个等待周期 00101: 插入5个等待周期 00110: 插入6个等待周期 00111: 插入7个等待周期 01000: 插入8个等待周期 01001: 插入9个等待周期 01010: 插入10个等待周期 01011: 插入11个等待周期 01100: 插入12个等待周期 01101: 插入13个等待周期 01110: 插入14个等待周期 01111: 插入15个等待周期 10000: 插入16个等待周期 10001: 插入17个等待周期 10010: 插入18个等待周期 10011: 插入19个等待周期 10100: 插入20个等待周期 10101: 插入21个等待周期 10110: 插入22个等待周期 10111: 插入23个等待周期 11000: 插入24个等待周期 11001: 插入25个等待周期 11010: 插入26个等待周期 11011: 插入27个等待周期 11100: 插入28个等待周期 11101: 插入29个等待周期 11110: 插入30个等待周期 11111: 插入31个等待周期 | R/W |
| b23-b21 | — | 保留位 | 读写值都为“0”。 | R/W |
| b28-b24 | CSRWAIT [4:0] | 正常读周期的等待选择位 | b28 b24 00000: 不插入等待 00001: 插入1个等待周期 00010: 插入2个等待周期 00011: 插入3个等待周期 00100: 插入4个等待周期 00101: 插入5个等待周期 00110: 插入6个等待周期 00111: 插入7个等待周期 01000: 插入8个等待周期 01001: 插入9个等待周期 01010: 插入10个等待周期 01011: 插入11个等待周期 01100: 插入12个等待周期 01101: 插入13个等待周期 01110: 插入14个等待周期 01111: 插入15个等待周期 10000: 插入16个等待周期 10001: 插入17个等待周期 10010: 插入18个等待周期 10011: 插入19个等待周期 10100: 插入20个等待周期 10101: 插入21个等待周期 10110: 插入22个等待周期 10111: 插入23个等待周期 11000: 插入24个等待周期 11001: 插入25个等待周期 11010: 插入26个等待周期 11011: 插入27个等待周期 11100: 插入28个等待周期 11101: 插入29个等待周期 11110: 插入30个等待周期 11111: 插入31个等待周期 | R/W |
| b31-b29 | — | 保留位 | 读写值都为“0”。 | R/W |

CSiWCNT1 寄存器是选择外部地址空间各区域的等待周期数的寄存器。

CSPWAIT[2:0] 位（页面写周期的等待选择位）

这些位设定在第 2 个页面写周期以后的存取中要插入的周期数。

当 CSiMOD.PWENB 位为“1”时，CSPWAIT[2:0] 位的设定有效。

注 1. 必须设定为 $1 \leq \text{WDON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSPWAIT}[2:0]$ 或者 $\text{CSON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSPWAIT}[2:0]$ 。

CSPRWAIT[2:0] 位（页面读周期的等待选择位）

这些位设定在第 2 个页面读周期以后的存取中要插入的周期数。

当 CSiMOD.PRENB 位为“1”时，CSPRWAIT[2:0] 位的设定有效。

注 1. 必须设定为 $\text{CSON}[2:0] \leq \text{RDON}[2:0] \leq \text{CSPRWAIT}[2:0]$ 。

CSWAIT[4:0] 位（正常写周期的等待选择位）

这些位设定在正常写周期和页面写周期的第 1 次存取中要插入的周期数。

注 1. 必须设定为 $1 \leq \text{WDON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSWAIT}[4:0]$ 或者 $\text{CSON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSWAIT}[4:0]$ 。

CSRWAIT[4:0] 位（正常读周期的等待选择位）

这些位设定在正常读周期和页面读周期的第 1 次存取中要插入的周期数。

注 1. 必须设定为 $\text{CSON}[2:0] \leq \text{RDON}[2:0] \leq \text{CSRWAIT}[4:0]$ 。

注 必须在“11.5.5.1 正常存取和页面存取时的限制事项”所示的限制范围内设定各位。

11.3.5 CSi 等待控制寄存器 2 (CSiWCNT2) (i=0 ~ 7)

地址 0008 3008h ~ 0008 3078h

| | | | | | | | | | | | | | | | |
|-------|-----------|-----|-----|-----|------------|-----|-----|-----|-------------|-----|-----|-----|-------------|-----|-----|
| b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| — | CSON[2:0] | | | — | WDON[2:0] | | | — | WRON[2:0] | | | — | RDON[2:0] | | |
| 复位后的值 | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | — | WDOFF[2:0] | | | — | CSWOFF[2:0] | | | — | CSROFF[2:0] | | |
| 复位后的值 | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|-------------|-----------------|---|-----|
| b2-b0 | CSROFF[2:0] | 读取时的 CS 延长周期选择位 | b2 b0 000: 不插入等待 001: 插入 1 个等待周期 010: 插入 2 个等待周期 011: 插入 3 个等待周期 100: 插入 4 个等待周期 101: 插入 5 个等待周期 110: 插入 6 个等待周期 111: 插入 7 个等待周期 | R/W |
| b3 | — | 保留位 | 读写值都为“0”。 | R/W |
| b6-b4 | CSWOFF[2:0] | 写入时的 CS 延长周期选择位 | b6 b4 000: 不插入等待 001: 插入 1 个等待周期 010: 插入 2 个等待周期 011: 插入 3 个等待周期 100: 插入 4 个等待周期 101: 插入 5 个等待周期 110: 插入 6 个等待周期 111: 插入 7 个等待周期 | R/W |
| b7 | — | 保留位 | 读写值都为“0”。 | R/W |
| b10-b8 | WDOFF[2:0] | 写数据的输出延长周期选择位 | b10 b8 000: 不插入等待 001: 插入 1 个等待周期 010: 插入 2 个等待周期 011: 插入 3 个等待周期 100: 插入 4 个等待周期 101: 插入 5 个等待周期 110: 插入 6 个等待周期 111: 插入 7 个等待周期 | R/W |
| b15-b11 | — | 保留位 | 读写值都为“0”。 | R/W |

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|-----------|-------------|--|-----|
| b18-b16 | RDON[2:0] | RD 有效等待选择位 | b18 b16 000: 不插入等待 001: 插入 1 个等待周期 010: 插入 2 个等待周期 011: 插入 3 个等待周期 100: 插入 4 个等待周期 101: 插入 5 个等待周期 110: 插入 6 个等待周期 111: 插入 7 个等待周期 | R/W |
| b19 | — | 保留位 | 读写值都为“0”。 | R/W |
| b22-b20 | WRON[2:0] | WR 有效等待选择位 | b22 b20 000: 不插入等待 001: 插入 1 个等待周期 010: 插入 2 个等待周期 011: 插入 3 个等待周期 100: 插入 4 个等待周期 101: 插入 5 个等待周期 110: 插入 6 个等待周期 111: 插入 7 个等待周期 | R/W |
| b23 | — | 保留位 | 读写值都为“0”。 | R/W |
| b26-b24 | WDON[2:0] | 写数据的输出等待选择位 | b26 b24 000: 不插入等待 001: 插入 1 个等待周期 010: 插入 2 个等待周期 011: 插入 3 个等待周期 100: 插入 4 个等待周期 101: 插入 5 个等待周期 110: 插入 6 个等待周期 111: 插入 7 个等待周期 | R/W |
| b27 | — | 保留位 | 读写值都为“0”。 | R/W |
| b30-b28 | CSON[2:0] | CS 有效等待选择位 | b30 b28 000: 不插入等待 001: 插入 1 个等待周期 010: 插入 2 个等待周期 011: 插入 3 个等待周期 100: 插入 4 个等待周期 101: 插入 5 个等待周期 110: 插入 6 个等待周期 111: 插入 7 个等待周期 | R/W |
| b31 | — | 保留位 | 读写值都为“0”。 | R/W |

CSiWCNT2 寄存器是选择外部地址空间各区域的等待周期数的寄存器。

CSROFF[2:0] 位（读取时的 CS 延长周期选择位）

这些位设定从读存取时的等待周期结束（RD# 信号无效）到 CSi# 信号（i=0 ~ 7）无效的周期数。

CSWOFF[2:0] 位（写入时的 CS 延长周期选择位）

这些位设定从写存取时的等待周期结束（WR0# 信号、WR1# 信号和 WR# 信号无效）到 CSi# 信号（i=0 ~ 7）无效的周期数。

注 1. 必须设定为 $1 \leq \text{WDOFF}[2:0] \leq \text{CSWOFF}[2:0]$ 。

WDOFF[2:0] 位（写数据的输出延长周期选择位）

这些位设定从写存取时的等待周期结束（WR0# 信号、WR1# 信号和 WR# 信号无效）到写数据输出结束的周期数。

注 1. 必须设定为 $1 \leq \text{WDOFF}[2:0] \leq \text{CSWOFF}[2:0]$ 。

RDON[2:0] 位（RD 有效等待选择位）

这些位设定在 RD# 信号有效前插入的等待周期数。

注 1. 在正常读存取时，必须设定为 $\text{CSON}[2:0] \leq \text{RDON}[2:0] \leq \text{CSRWAIT}[4:0]$ 。

在页面读存取时，必须设定为 $\text{CSON}[2:0] \leq \text{RDON}[2:0] \leq \text{CSPRWAIT}[2:0]$ 。

WRON[2:0] 位（WR 有效等待选择位）

这些位设定在 WR0# 信号、WR1# 信号或者 WR# 信号有效前插入的等待周期数。

注 1. 在正常写存取时，必须设定为 $1 \leq \text{WDON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSWAIT}[4:0]$ 或者 $\text{CSON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSWAIT}[4:0]$ 。

在页面写存取时，必须设定为 $1 \leq \text{WDON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSPWAIT}[2:0]$ 或者 $\text{CSON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSPWAIT}[2:0]$ 。

WDON[2:0] 位（写数据的输出等待选择位）

这些位设定在输出写数据前插入的等待周期数。

注 1. 在正常写存取时，必须设定为 $1 \leq \text{WDON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSWAIT}[4:0]$ 。

在页面写存取时，必须设定为 $1 \leq \text{WDON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSPWAIT}[2:0]$ 。

CSON[2:0] 位（CS 有效等待选择位）

这些位设定在 CSi# 信号（i=0 ~ 7）有效前插入的等待周期数。

注 1. 在正常读存取时，必须设定为 $\text{CSON}[2:0] \leq \text{RDON}[2:0] \leq \text{CSRWAIT}[4:0]$ 。

在页面读存取时，必须设定为 $\text{CSON}[2:0] \leq \text{RDON}[2:0] \leq \text{CSPRWAIT}[2:0]$ 。

在正常写存取时，必须设定为 $\text{CSON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSWAIT}[4:0]$ 。

在页面写存取时，必须设定为 $\text{CSON}[2:0] \leq \text{WRON}[2:0] \leq \text{CSPWAIT}[2:0]$ 。

注 必须在“11.5.5.1 正常存取和页面存取时的限制事项”所示的限制范围内设定各位。

11.3.6 总线错误源清除寄存器（BERCLR）

地址 0008 1300h

| | | | | | | | | |
|-------|----|----|----|----|----|----|----|------------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | — | — | — | — | — | STSC LR |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------------|----------|---------------------|----------------|
| b0 | STSC LR | 总线错误源清除位 | 0: 无效 1: 清除总线错误源 | R/(W) (注 1) |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

注 1. 只能写“1”。

STSCLR 位（总线错误源清除位）

如果给此位写“1”，就清除内部的总线错误源。在总线错误引起的中断处理程序内，必须通过给此位写“1”来清除总线保持的总线错误源。

写“0”无效，读取值为“0”。

11.3.7 总线错误监视允许寄存器 (BEREN)

地址 0008 1304h

| | | | | | | | |
|----|----|----|----|----|----|------|-------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | — | — | TOEN | IGAEN |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-----------------|----------------------------------|-----|
| b0 | IGAEN | 非法地址存取的检测允许位 | 0: 禁止检测非法地址的存取 1: 允许检测非法地址的存取 | R/W |
| b1 | TOEN | 超时检测允许位 (注1、注2) | 0: 禁止检测总线超时 1: 允许检测总线超时 | R/W |
| b7-b2 | — | 保留位 | 读写值都为“0”。 | R/W |

注1. 如果在设定为禁止检测 (TOEN 位 = 0) 后进行总线存取, 总线就有可能被冻结。

注2. 不能在检测超时错误时将 TOEN 位设定为禁止检测。

IGAEN 位 (非法地址的存取检测允许位)

此位设定允许或者禁止检测非法地址的存取。

TOEN 位 (超时检测允许位)

此位允许或者禁止检测总线超时。

11.3.8 总线错误中断允许寄存器 (BERIE)

地址 0008 1306h

| | | | | | | | |
|----|----|----|----|----|----|----|------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | — | — | — | CPEN |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|----------------|-----------------------------|-----|
| b0 | CPEN | CPU 总线错误的通知控制位 | 0: 不通知总线错误中断 1: 通知总线错误中断 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

CPEN 位 (CPU 总线错误的通知控制位)

此位在检测到总线错误时选择是否通知中断控制器。

11.4 字节序和数据对齐

CPU 和其他内部总线主控的数据长度有 8 位、16 位和 32 位。外部总线控制部具有数据定位功能，根据外部地址空间的总线规格（8 位或者 16 位的总线空间）、数据长度和字节序模式，控制在存取外部地址空间时是使用数据总线 D15 ~ D8 还是使用 D7 ~ D0。

11.4.1 16 位总线空间

如果通过 CSiCNT.BSIZE[1:0] 位选择 16 位总线空间，地址总线（A23 ~ A1）就作为以 16 位为单位的地址输出信号并且有效，地址 A0 总是“0”。

当选择字节选通模式（CSiMOD.WRMOD 位 =0）时，WRn#（n=0,1）引脚有效。不使用 BCn#（n=0,1）引脚。

当选择 1 次写选通模式（CSiMOD.WRMOD 位 =1）时，WR# 引脚有效，并且与数据长度无关，在进行写存取时从 WR# 引脚输出 Low 电平。通过 BCn#（n=0,1）引脚表示有效的字节位置。不使用 WRn#（n=0,1）引脚。

在存取 32 位数据时，有可能发生页面存取。发生页面存取时的例子如图 11.3 和图 11.4 中的 (p) 所示。在大端法和小端法中，芯片的外部数据和控制信号的有效位置不同。

| 数据长度 | 存取地址 | 存取次数 | 总线周期 | 数据量 | 地址 | WR1#/BC1# WR0#/BC0# | | RD# | |
|-------|------|------|------|-------|-------------|-------------------------|--|-----|--|
| | | | | | | 数据总线 D15 D8 D7 D0 | | | |
| 8bit | 4n | 1次 | 第1次 | 8bit | 4n | [7 0] | | | |
| | 4n+1 | 1次 | 第1次 | 8bit | 4n | [7 0] | | | |
| | 4n+2 | 1次 | 第1次 | 8bit | 4n+2 | [7 0] | | | |
| | 4n+3 | 1次 | 第1次 | 8bit | 4n+2 | [7 0] | | | |
| 16bit | 4n | 1次 | 第1次 | 16bit | 4n | [15 8 7 0] | | | |
| | 4n+1 | 2次 | 第1次 | 8bit | 4n | [7 0] | | | |
| | | | 第2次 | 8bit | 4n+2 | [15 8] | | | |
| | 4n+2 | 1次 | 第1次 | 16bit | 4n+2 | [15 8 7 0] | | | |
| | 4n+3 | 2次 | 第1次 | 8bit | 4n+2 | [7 0] | | | |
| | | | 第2次 | 8bit | 4n+4 | [15 8] | | | |
| 32bit | 4n | 2次 | 第1次 | 16bit | 4n | [15 8 7 0] | | | |
| | | | 第2次 | 16bit | 4n+2 (p) | [31 24 23 16] | | | |
| | 4n+1 | 3次 | 第1次 | 8bit | 4n | [7 0] | | | |
| | | | 第2次 | 16bit | 4n+2 | [23 16 15 8] | | | |
| | | | 第3次 | 8bit | 4n+4 | [31 24] | | | |
| | 4n+2 | 2次 | 第1次 | 16bit | 4n+2 | [15 8 7 0] | | | |
| | | | 第2次 | 16bit | 4n+4 | [31 24 23 16] | | | |
| | 4n+3 | 3次 | 第1次 | 8bit | 4n+2 | [7 0] | | | |
| | | | 第2次 | 16bit | 4n+4 | [23 16 15 8] | | | |
| 第3次 | | | 8bit | 4n+6 | [31 24] | | | | |

如果通过CSiMOD.PRENB位和CSiMOD.PWENB位允许页面存取，(p)表示的存取就为页面存取的对象。

图 11.3 16 位总线空间的数据定位（小端法）

| 数据长度 | 存取地址 | 存取次数 | 总线周期 | 数据量 | 地址 | WR1#/BC1# | WR0#/BC0# | |
|-------|------|------|------|-------|----------|-----------|-----------|----|
| | | | | | | RD# | | |
| | | | | | | 数据总线 | | |
| | | | | | | D15 | D8 D7 | D0 |
| 8bit | 4n | 1次 | 第1次 | 8bit | 4n | 7 | 0 | |
| | 4n+1 | 1次 | 第1次 | 8bit | 4n | 7 | 0 | |
| | 4n+2 | 1次 | 第1次 | 8bit | 4n+2 | 7 | 0 | |
| | 4n+3 | 1次 | 第1次 | 8bit | 4n+2 | 7 | 0 | |
| 16bit | 4n | 1次 | 第1次 | 16bit | 4n | 15 | 8 7 | 0 |
| | 4n+1 | 2次 | 第1次 | 8bit | 4n | 15 | 8 | |
| | | | 第2次 | 8bit | 4n+2 | 7 | 0 | |
| | 4n+2 | 1次 | 第1次 | 16bit | 4n+2 | 15 | 8 7 | 0 |
| | 4n+3 | 2次 | 第1次 | 8bit | 4n+2 | 15 | 8 | |
| | | | 第2次 | 8bit | 4n+4 | 7 | 0 | |
| 32bit | 4n | 2次 | 第1次 | 16bit | 4n | 31 | 24 23 | 16 |
| | | | 第2次 | 16bit | 4n+2 (p) | 15 | 8 7 | 0 |
| | 4n+1 | 3次 | 第1次 | 8bit | 4n | 31 | 24 | |
| | | | 第2次 | 16bit | 4n+2 | 23 | 16 15 | 8 |
| | | | 第3次 | 8bit | 4n+4 | 7 | 0 | |
| | 4n+2 | 2次 | 第1次 | 16bit | 4n+2 | 31 | 24 23 | 16 |
| | | | 第2次 | 16bit | 4n+4 | 15 | 8 7 | 0 |
| | 4n+3 | 3次 | 第1次 | 8bit | 4n+2 | 31 | 24 | |
| | | | 第2次 | 16bit | 4n+4 | 23 | 16 15 | 8 |
| 第3次 | | | 8bit | 4n+6 | 7 | 0 | | |

如果通过CSiMOD.PRENb位和CSiMOD.PWENb位允许页面存取，(p)表示的存取就为页面存取的对象。

图 11.4 16 位总线空间的数据定位（大端法）

11.4.2 8 位总线空间

如果通过 CSiCNT.BSIZE[1:0] 位选择 8 位总线空间，地址总线（A23 ~ A0）就作为以字节为单位的地址信号并且有效。

当选择字节选通模式时，WR0# 引脚有效；当选择 1 次写选通模式时，WR# 引脚有效。在进行写存取时，将 Low 电平输出到 WR0# 引脚和 WR# 引脚。不使用 BC0#、WR1# 和 BC1# 引脚。

在存取 16 位或者 32 位数据时，有可能发生页面存取。发生页面存取时的例子如图 11.5 和图 11.6 的 (p) 所示。

与字节序无关，芯片的外部数据和控制信号的有效位置不变。

| 数据长度 | 存取地址 | 存取次数 | 总线周期 | 数据量 | 地址 | WR1#/BC1# WR0#/BC0# | |
|-------|------|------|------|------|----------|-----------------------|--------------|
| | | | | | | RD# | |
| | | | | | | 数据总线 | |
| | | | | | | D15 | D8 D7 D0 |
| 8bit | 4n | 1次 | 第1次 | 8bit | 4n | 7 | 0 |
| | 4n+1 | 1次 | 第1次 | 8bit | 4n+1 | 7 | 0 |
| | 4n+2 | 1次 | 第1次 | 8bit | 4n+2 | 7 | 0 |
| | 4n+3 | 1次 | 第1次 | 8bit | 4n+3 | 7 | 0 |
| 16bit | 4n | 2次 | 第1次 | 8bit | 4n | 7 | 0 |
| | | | 第2次 | 8bit | 4n+1 (p) | 15 | 8 |
| | 4n+1 | 2次 | 第1次 | 8bit | 4n+1 | 7 | 0 |
| | | | 第2次 | 8bit | 4n+2 (p) | 15 | 8 |
| | 4n+2 | 2次 | 第1次 | 8bit | 4n+2 | 7 | 0 |
| | | | 第2次 | 8bit | 4n+3 (p) | 15 | 8 |
| | 4n+3 | 2次 | 第1次 | 8bit | 4n+3 | 7 | 0 |
| | | | 第2次 | 8bit | 4n+4 | 15 | 8 |
| 32bit | 4n | 4次 | 第1次 | 8bit | 4n | 7 | 0 |
| | | | 第2次 | 8bit | 4n+1 (p) | 15 | 8 |
| | | | 第3次 | 8bit | 4n+2 (p) | 23 | 16 |
| | | | 第4次 | 8bit | 4n+3 (p) | 31 | 24 |
| | 4n+1 | 4次 | 第1次 | 8bit | 4n+1 | 7 | 0 |
| | | | 第2次 | 8bit | 4n+2 (p) | 15 | 8 |
| | | | 第3次 | 8bit | 4n+3 (p) | 23 | 16 |
| | | | 第4次 | 8bit | 4n+4 | 31 | 24 |
| | 4n+2 | 4次 | 第1次 | 8bit | 4n+2 | 7 | 0 |
| | | | 第2次 | 8bit | 4n+3 (p) | 15 | 8 |
| | | | 第3次 | 8bit | 4n+4 | 23 | 16 |
| | | | 第4次 | 8bit | 4n+5 (p) | 31 | 24 |
| | 4n+3 | 4次 | 第1次 | 8bit | 4n+3 | 7 | 0 |
| | | | 第2次 | 8bit | 4n+4 | 15 | 8 |
| | | | 第3次 | 8bit | 4n+5 (p) | 23 | 16 |
| | | | 第4次 | 8bit | 4n+6 (p) | 31 | 24 |

如果通过CSiMOD.PRENB位和CSiMOD.PWENB位允许页面存取，(p)表示的存取就为页面存取的对象。

图 11.5 8 位总线空间的数据定位（小端法）

| 数据长度 | 存取地址 | 存取次数 | 总线周期 | 数据量 | 地址 | WR1#/BC1# WR0#/BC0# | |
|-------|------|------|------|------|----------|-----------------------|--------------|
| | | | | | | RD# | |
| | | | | | | 数据总线 | |
| | | | | | | D15 | D8 D7 D0 |
| 8bit | 4n | 1次 | 第1次 | 8bit | 4n | 7 | 0 |
| | 4n+1 | 1次 | 第1次 | 8bit | 4n+1 | 7 | 0 |
| | 4n+2 | 1次 | 第1次 | 8bit | 4n+2 | 7 | 0 |
| | 4n+3 | 1次 | 第1次 | 8bit | 4n+3 | 7 | 0 |
| 16bit | 4n | 2次 | 第1次 | 8bit | 4n | 15 | 8 |
| | | | 第2次 | 8bit | 4n+1 (p) | 7 | 0 |
| | 4n+1 | 2次 | 第1次 | 8bit | 4n+1 | 15 | 8 |
| | | | 第2次 | 8bit | 4n+2 (p) | 7 | 0 |
| | 4n+2 | 2次 | 第1次 | 8bit | 4n+2 | 15 | 8 |
| | | | 第2次 | 8bit | 4n+3 (p) | 7 | 0 |
| | 4n+3 | 2次 | 第1次 | 8bit | 4n+3 | 15 | 8 |
| | | | 第2次 | 8bit | 4n+4 | 7 | 0 |
| 32bit | 4n | 4次 | 第1次 | 8bit | 4n | 31 | 24 |
| | | | 第2次 | 8bit | 4n+1 (p) | 23 | 16 |
| | | | 第3次 | 8bit | 4n+2 (p) | 15 | 8 |
| | | | 第4次 | 8bit | 4n+3 (p) | 7 | 0 |
| | 4n+1 | 4次 | 第1次 | 8bit | 4n+1 | 31 | 24 |
| | | | 第2次 | 8bit | 4n+2 (p) | 23 | 16 |
| | | | 第3次 | 8bit | 4n+3 (p) | 15 | 8 |
| | | | 第4次 | 8bit | 4n+4 | 7 | 0 |
| | 4n+2 | 4次 | 第1次 | 8bit | 4n+2 | 31 | 24 |
| | | | 第2次 | 8bit | 4n+3 (p) | 23 | 16 |
| | | | 第3次 | 8bit | 4n+4 | 15 | 8 |
| | | | 第4次 | 8bit | 4n+5 (p) | 7 | 0 |
| | 4n+3 | 4次 | 第1次 | 8bit | 4n+3 | 31 | 24 |
| | | | 第2次 | 8bit | 4n+4 | 23 | 16 |
| | | | 第3次 | 8bit | 4n+5 (p) | 15 | 8 |
| | | | 第4次 | 8bit | 4n+6 (p) | 7 | 0 |

如果通过CSIMOD.PRENB位和CSIMOD.PWENB位允许页面存取，(p)表示的存取就为页面存取的对象。

图 11.6 8 位总线空间的数据定位（大端法）

11.5 运行说明

11.5.1 外部总线的存取时序

时序图中记载的各周期的说明如下所示：

1. **$T_{w1} \sim T_{wn}$** （正常读周期等待和正常写周期等待）

这是从外部总线存取开始到等待结束周期（后述）之间的周期期间，能选择0~31个周期。根据等待设定，在此期间内将CSn#、RD#、WRn#、WR#信号设定为Low电平有效。能通过CSi等待控制寄存器2（CSiWCNT2）的CS有效等待（CSON）选择位、RD有效等待（RDON）选择位、WR有效等待（WRON）选择位和写数据的输出等待（WDON）选择位，对有效时序进行控制。各等待周期数从外部总线存取的开始周期算起，能选择0~7个周期。但是可选择的周期数不超过读/写周期等待的周期数。

2. **T_{end}** （选通信号有效周期）

T_{end} 表示正常读/写周期等待期间或者页面读/写周期等待期间结束后的下一个周期。当正常读/写周期等待或者页面读/写周期等待的各选择位为“0”时，总线存取的开始周期为选通信号的有效周期。在选通信号有效周期的下一个周期，RD#、WRn#和WR#信号无效。在进行读存取时， T_{end} 为读数据的采样周期。在允许外部等待的情况下，在选通信号的有效周期对等待信号进行采样。当等待信号为Low电平时，延长总线周期，并且在等待信号变为High电平时，在下一个周期结束总线周期。（ T_{end} ）表示开始等待信号的采样周期。

在页面存取中的第1次选通信号有效周期时，除设定了写存取时的写数据输出延长周期（非“0”值）的情况（5.）以外，开始下一个周期的第2次以后的页面存取（6.）。如果RD有效等待选择位或者WR有效等待选择位的设定值不是“0”，RD#、WRn#和WR#信号就在下一个周期无效。如果设定值是“0”，各信号就继续有效，而且CSn#信号也继续有效。

3. **$T_{n1} \sim T_{nm}$** （CS延长周期）

在正常存取的情况下， $T_{n1} \sim T_{nm}$ 表示从选通信号有效周期（ T_{end} ）的下一个周期到CSn#信号无效的周期期间。在进行读存取时，能通过读CSi等待控制寄存器2（CSiWCNT2）时的CS延长周期选择位（CSROFF）控制无效时序；在进行写存取时，能通过写入时的CS延长周期选择位（CSWOFF）控制无效时序。

周期数从选通信号有效周期的下一个周期开始算起。

在页面存取的情况下， $T_{n1} \sim T_{nm}$ 表示从最后的选通信号有效周期到CSn#信号无效的周期期间。

在进行写存取时，通过写数据的输出延长周期选择位（WDOFF）延长地址和输出数据。

4. **T_h** （地址保持期间）

在CS延长周期结束后的下一个周期，保持前一次存取的地址值。但是，在总线主控的1次传送请求至少需要2次外部总线存取的情况下，只保持被分割的最后总线存取的地址值。对于最后总线存取以外的存取，在CSn#信号无效时地址被更新为下一次存取的值。在正常存取的情况下，在CSn#信号无效期间插入1个周期；在页面存取的情况下，在CSn#信号无效期间不插入周期（参照图11.10和图11.11）。

5. **$T_{dw1} \sim T_{dwn}$** （写数据的输出延长周期）

如果在进行写存取时写数据的输出延长周期选择位的设定值不是“0”，就从选通信号有效周期（ T_{end} ）的下一个周期插入写数据的输出延长周期。

在正常存取的情况下，在CS延长周期（3.）期间内插入周期。

在页面存取的情况下，在选通信号有效周期（ T_{end} ）和后续的页面存取之间以及CS延长周期（3.）的期间内插入周期。在此期间，地址和输出数据被延长并且WRn#信号和WR#信号无效。

6. $T_{pw1} \sim T_{pwn}$ (页面读周期等待和页面写周期等待)

对于页面存取的第2次以后的总线周期，使用页面读周期等待和页面写周期等待的值而不使用正常读周期等待和正常写周期等待的值。和第1次存取一样，WR有效等待的设定有效。RD有效的操作因页面读存取模式 (CSiMOD.PRMOD) 的设定而不同。

CDiMOD.PRMOD=0 : 和第1次一样，插入RD有效等待并且RD#信号无效。

CDiMOD.PRMOD=1 : 和正常存取兼容模式一样，插入RD有效等待，但是RD#信号在该期间继续有效。

7. $T_{r1} \sim T_{rn}$ (恢复周期)

能从总线周期结束开始插入恢复周期。能通过CSi恢复周期设定寄存器 (CSiREC) 的读恢复 (RRCV) 设定、写恢复周期 (WRCV) 设定，控制恢复周期数。各恢复周期数从地址保持期间的下一个周期开始算起，可选择0~15个周期。恢复周期的详细内容请参照“11.5.3 恢复周期的插入”。

11.5.1.1 正常存取

如果将CSiMOD.PRENB位置“0”(禁止页面读存取)、将CSiMOD.PWENB位置“0”(禁止页面写存取)，全部总线存取就进行正常存取。

即使将CSiMOD.PRENB位置“1”(允许页面读存取)、将CSiMOD.PWENB位置“1”(允许页面写存取)，不对应页面存取的总线存取也为正常存取。

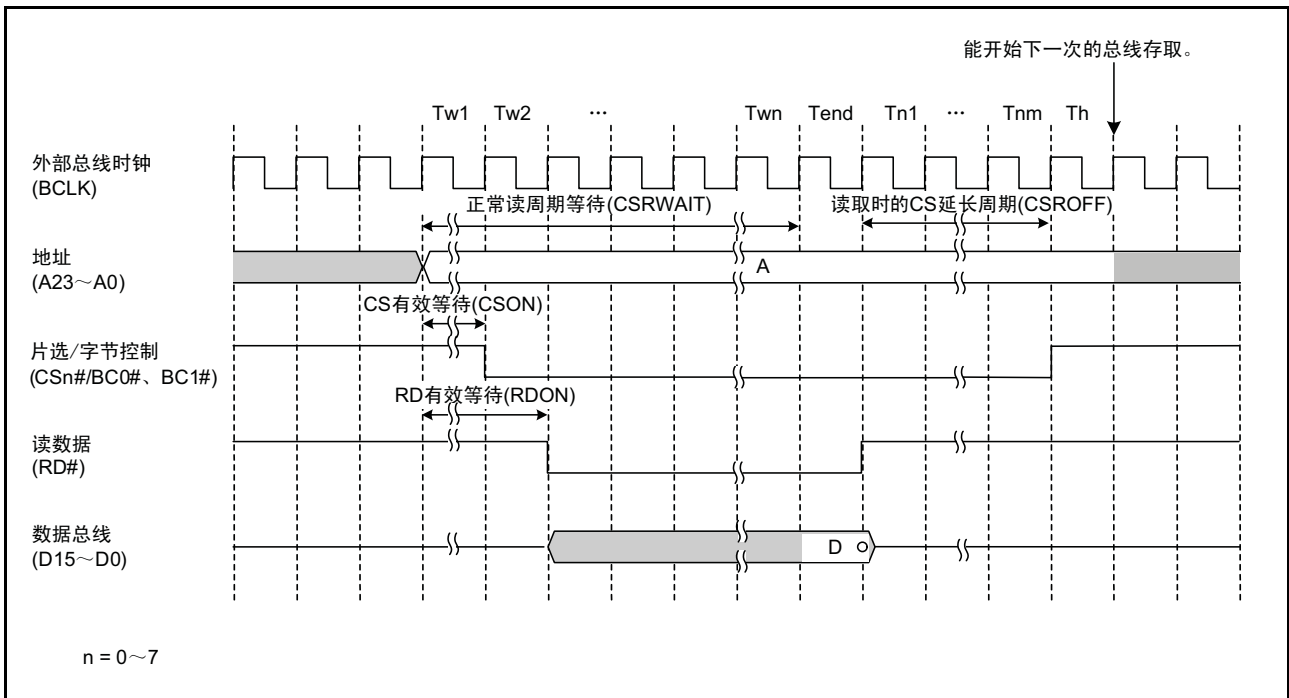


图 11.7 总线时序 (正常读)

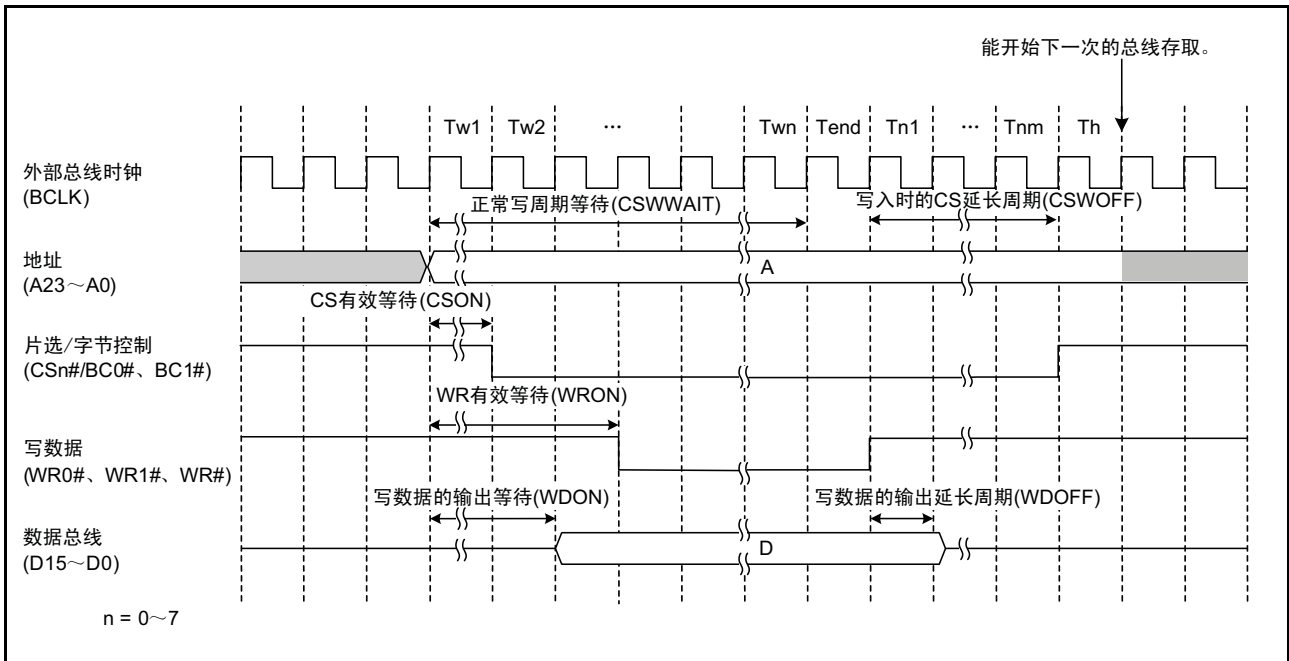


图 11.8 总线时序（正常写）

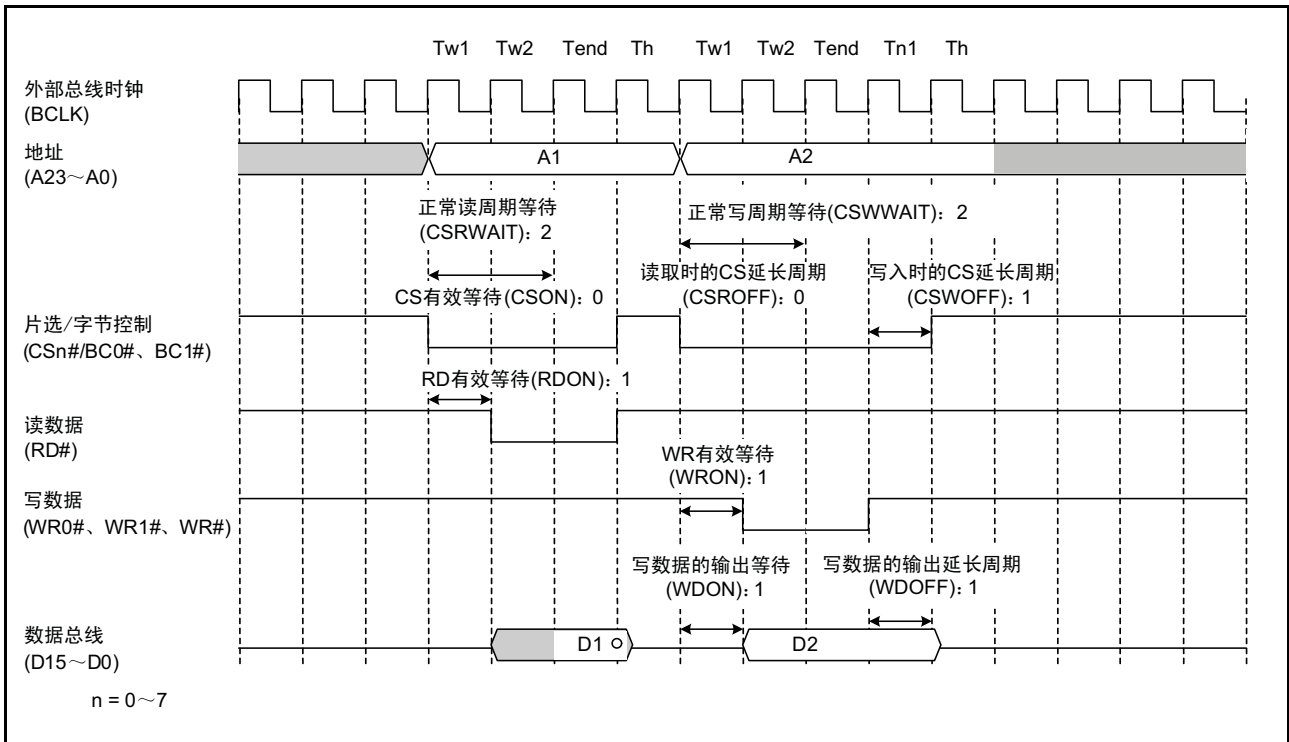


图 11.9 正常存取的运行例子（读写）

在总线主控的 1 次传送请求至少需要 2 次外部总线存取时，重复正常存取的操作（(1) ~ (4)）。1 次传送请求发生 2 次总线存取的运行例子如图 11.10 和图 11.11 所示，图中各等待控制寄存器的值是设定例子，必须根据要连接的器件规格设定寄存器的值。

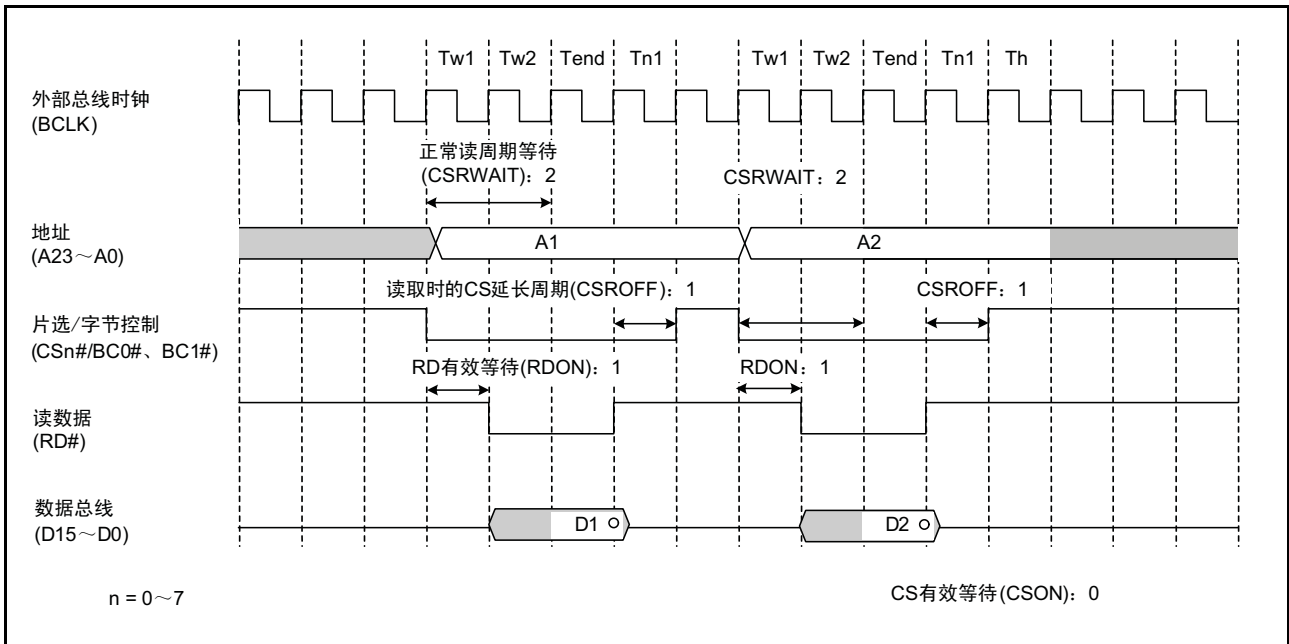


图 11.10 正常读存取的运行例子 (1次传送请求发生2次总线存取的情况)

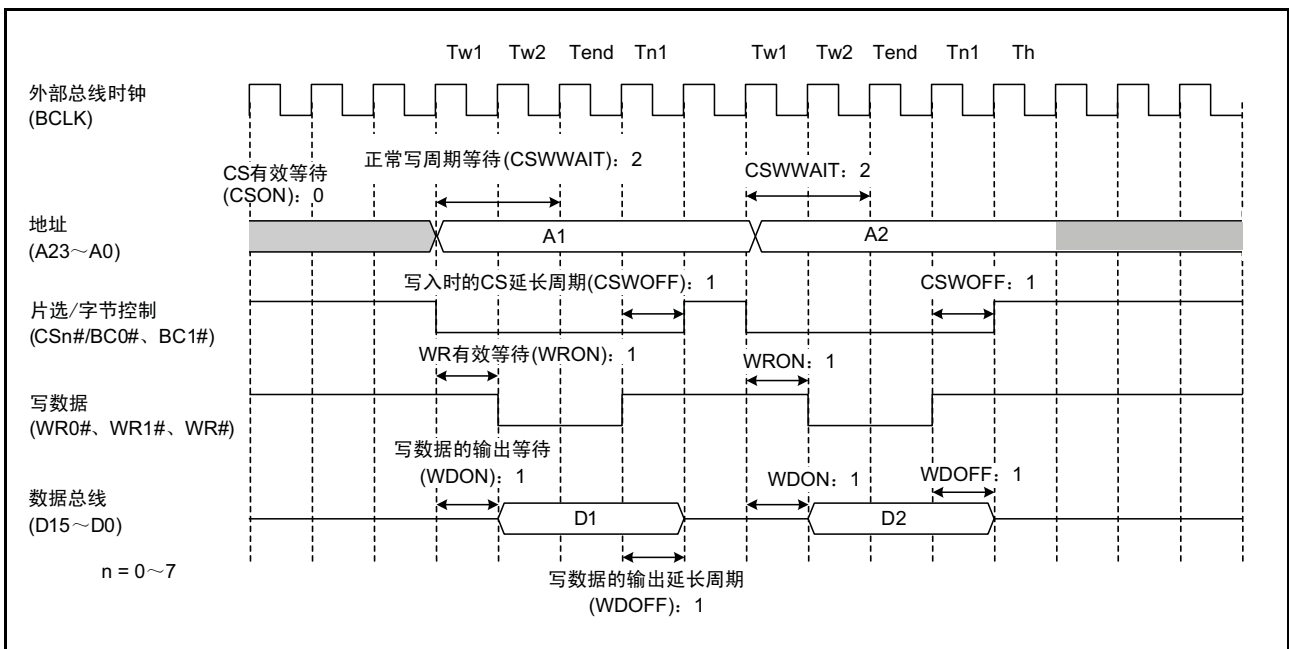


图 11.11 正常写存取的运行例子 (1次传送请求发生2次总线存取的情况)

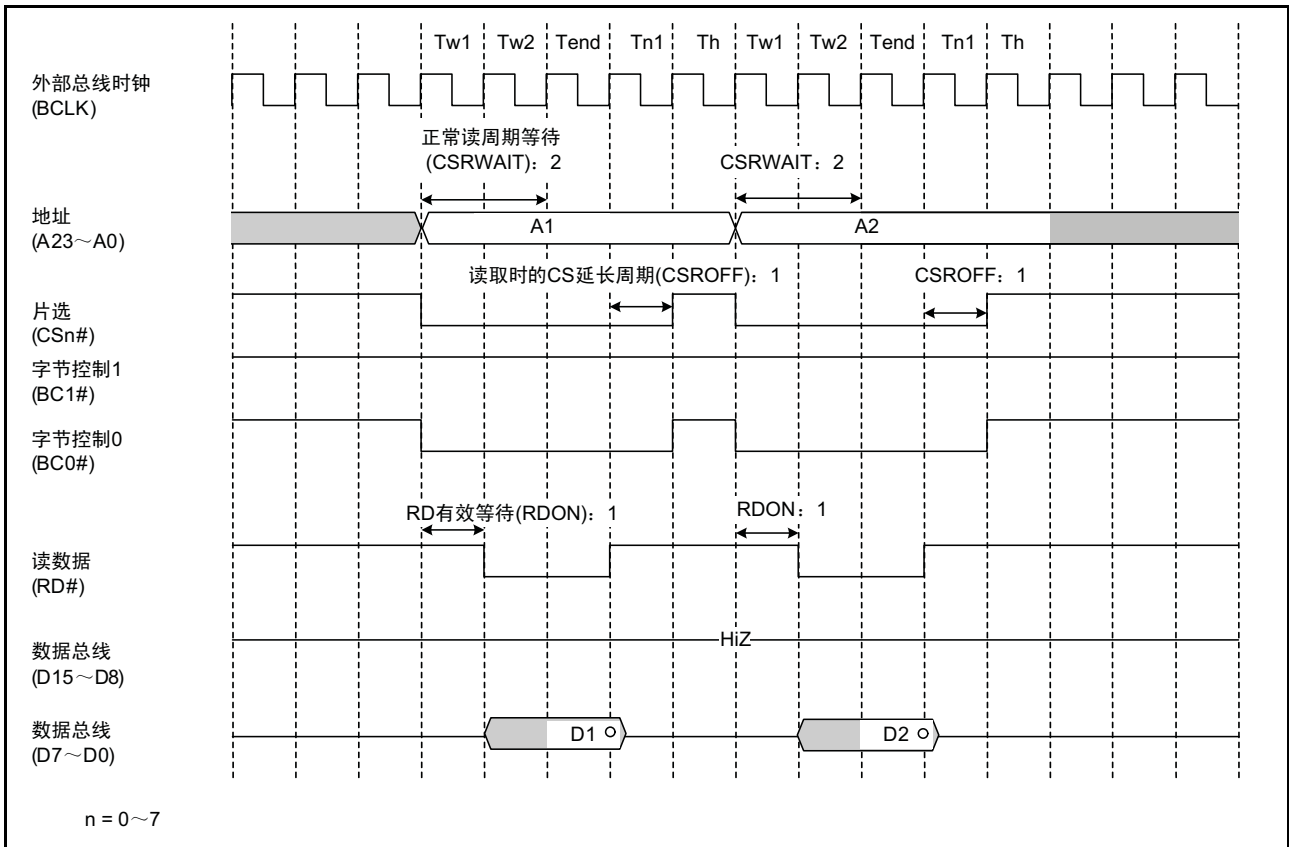


图 11.12 正常读存取的运行例子（对 16 位总线空间进行 8 位存取的情况）

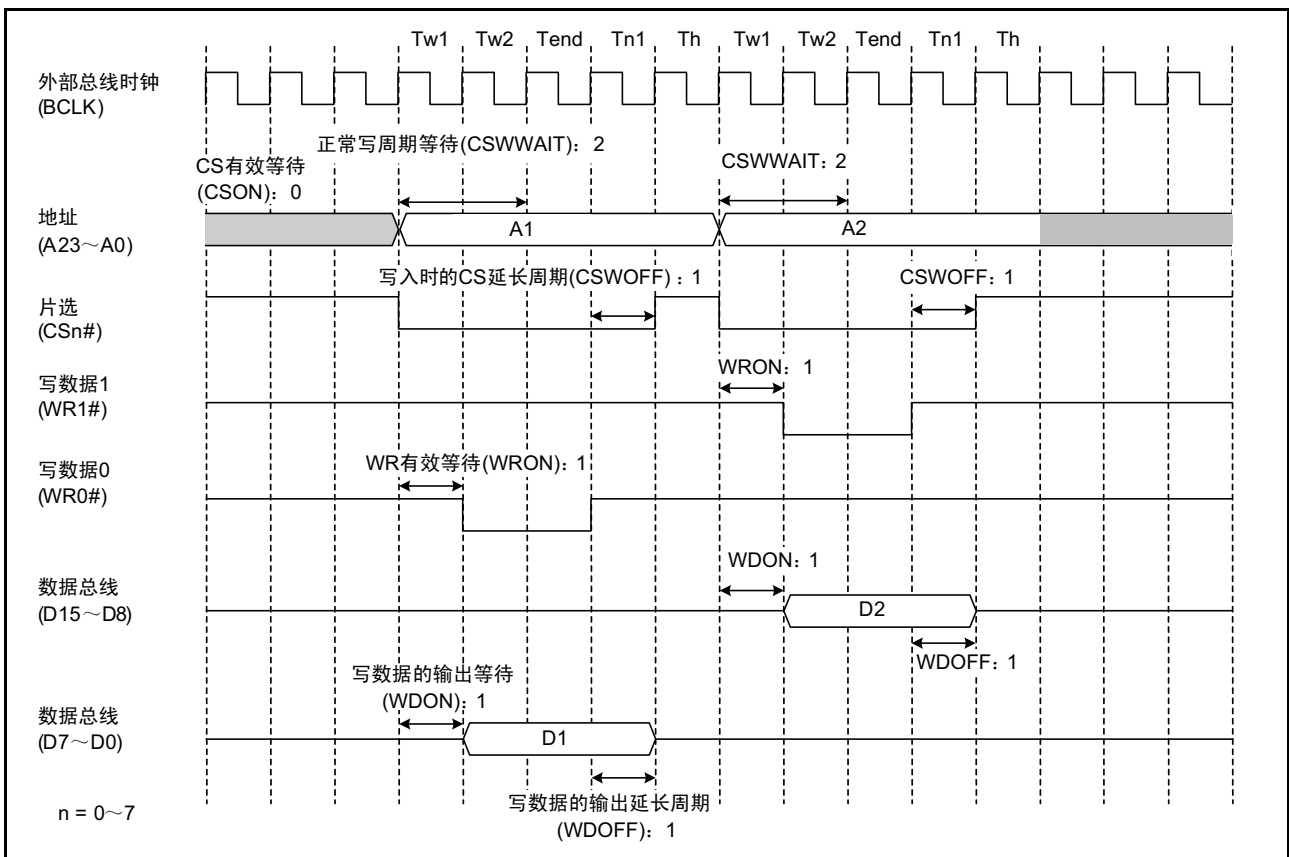


图 11.13 正常写存取的运行例子（在字节选通模式中，对 16 位总线空间进行 8 位存取的情况）

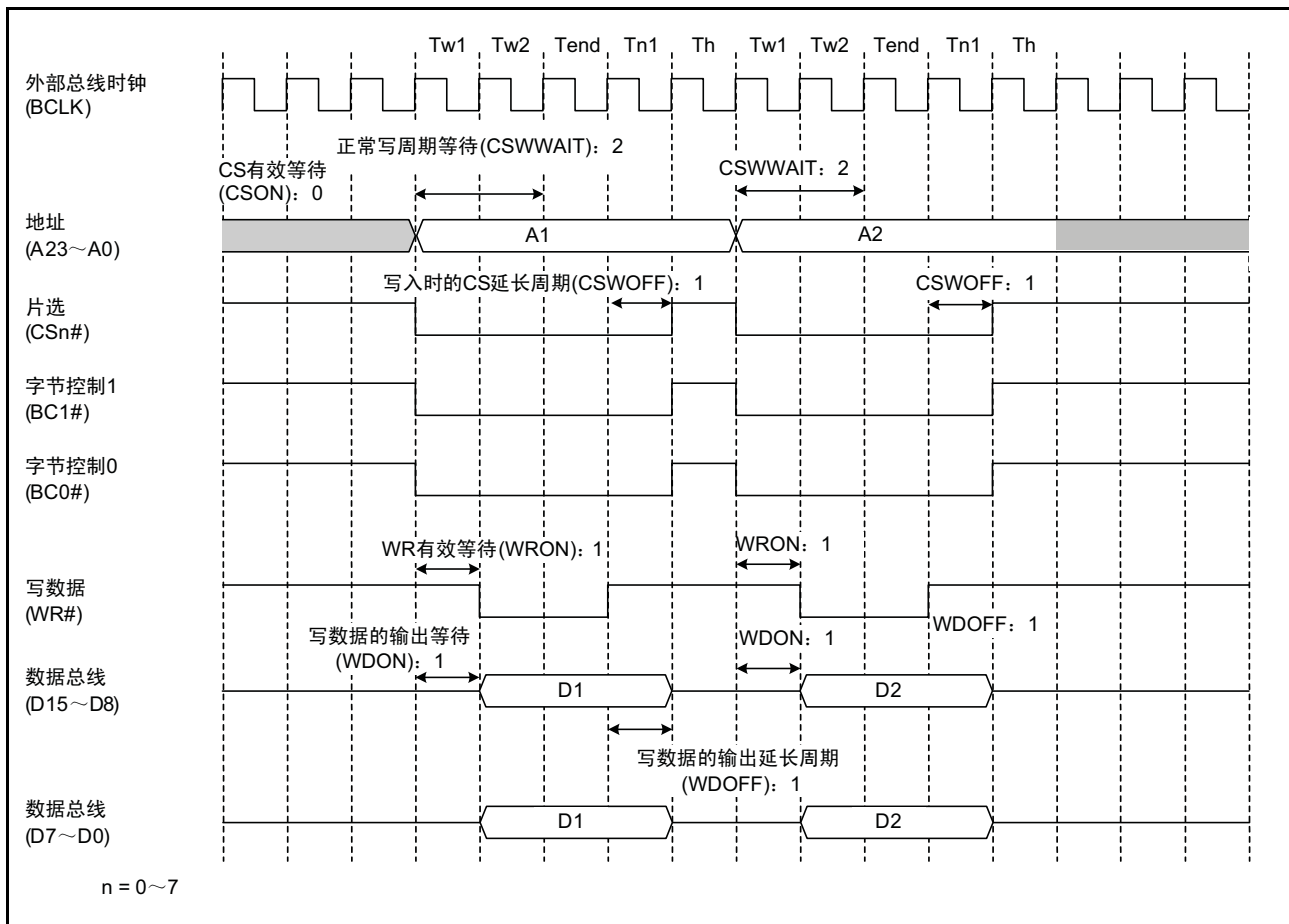


图 11.14 正常写存取的运行例子 (在 1 次写选通模式中, 对 16 位总线空间进行 16 位存取的情况)

11.5.1.2 页面存取

如果将 CSiMOD.PRENB 位置 “1”（允许页面读存取）、将 CSiMOD.PWENB 位置 “1”（允许页面写存取），对应页面存取的总线存取就为页面存取。在总线主控的 1 次传送请求至少需要 2 次外部总线存取时，为页面存取的对象。有关页面存取的发生条件，请参照图 11.5 ~ 图 11.8。

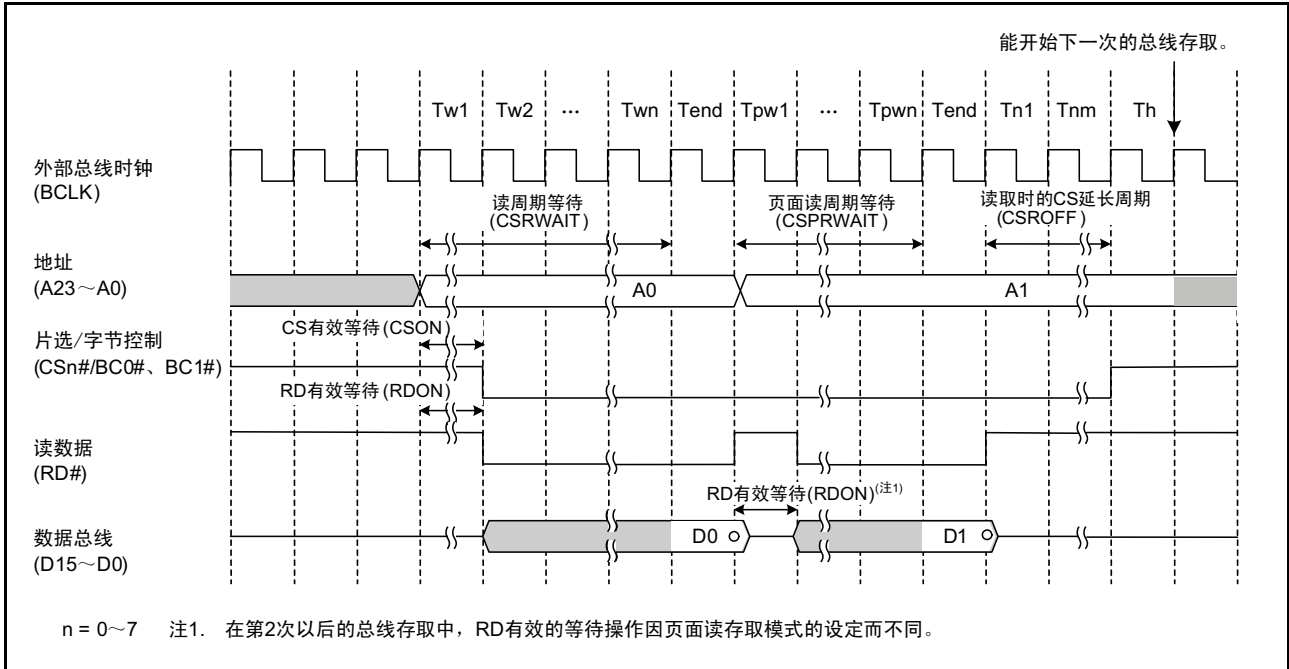


图 11.15 页面读存取时序

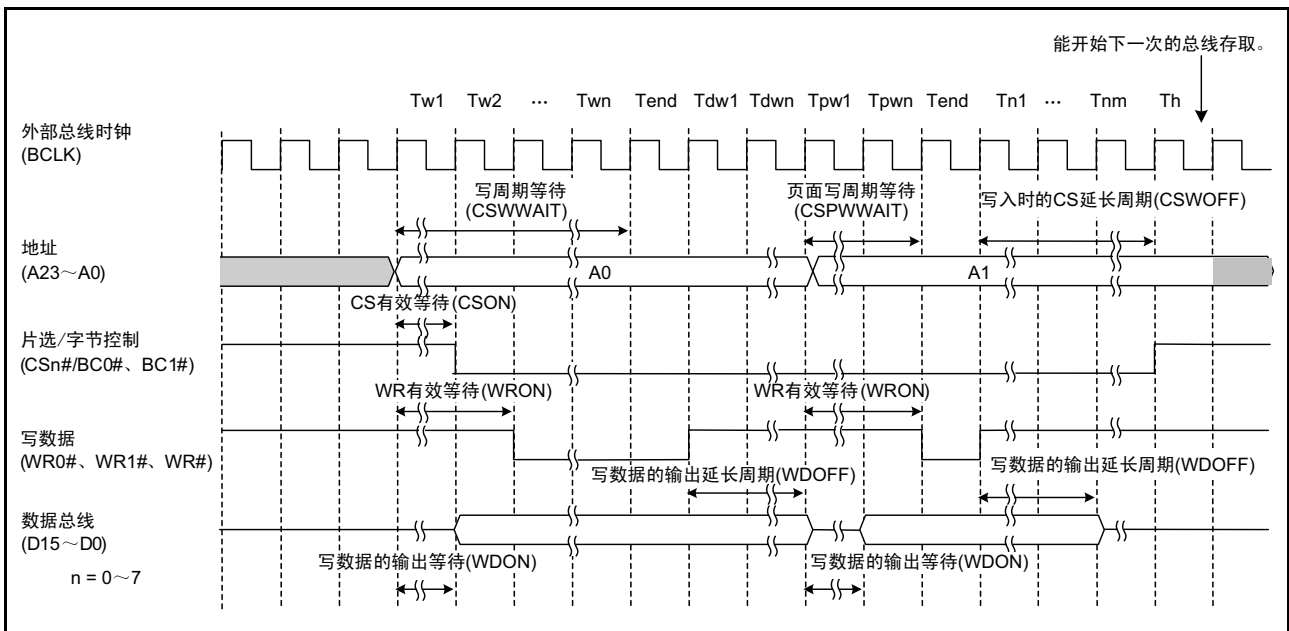


图 11.16 页面写存取时序

对 16 位总线空间进行 32 位存取时的运行例子如图 11.17 和图 11.18 所示，图中各等待控制寄存器的值为设定例子，必须根据要连接的器件规格设定寄存器的值。

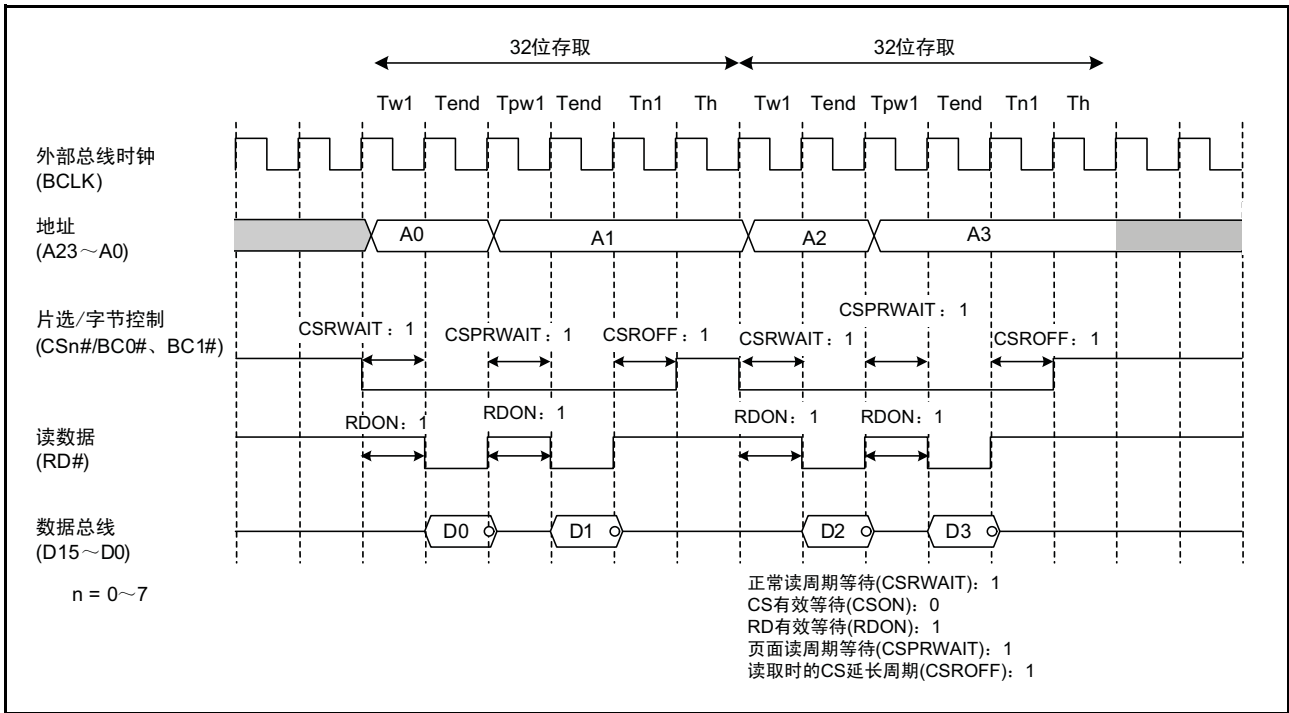


图 11.17 页面读存取的运行例子（对 16 位总线空间进行 32 位存取的情况）

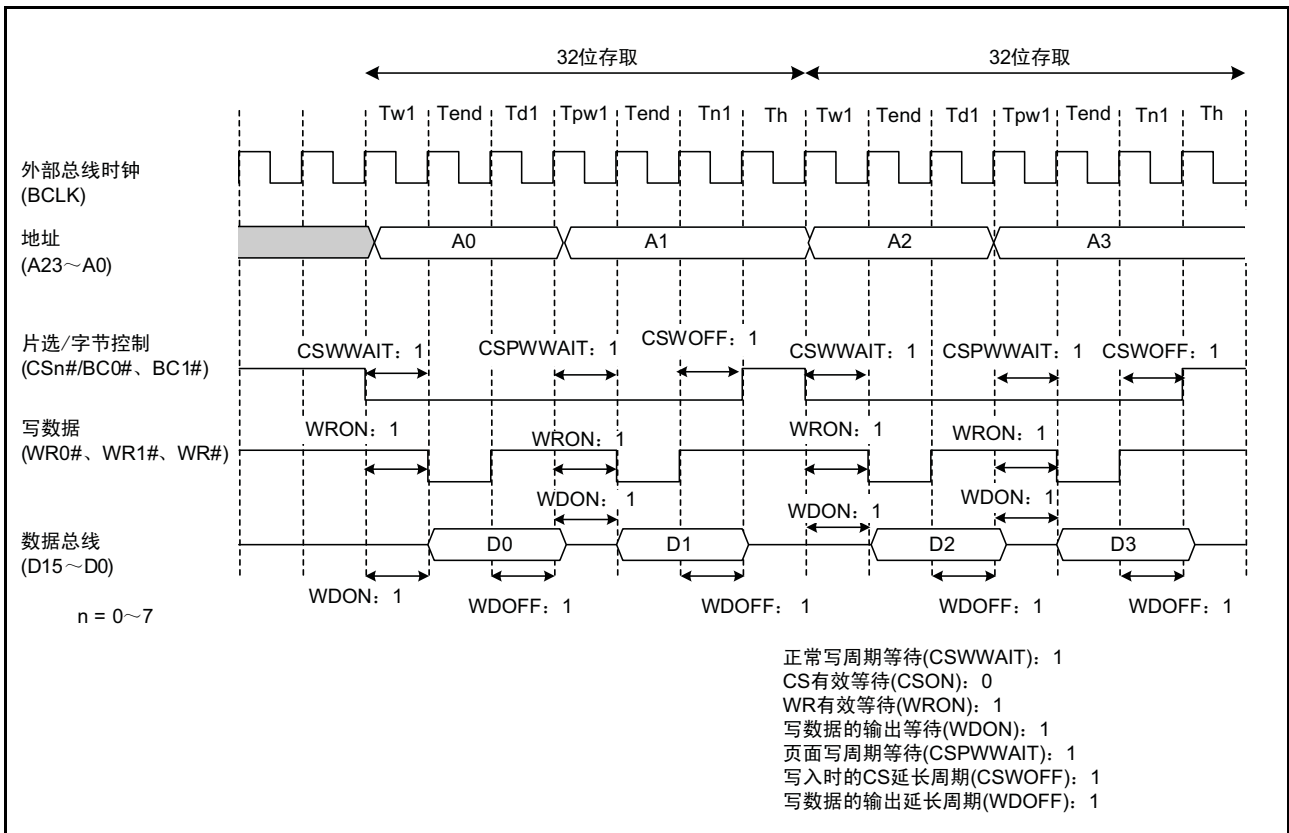


图 11.18 页面写存取的运行例子（对 16 位总线空间进行 32 位存取的情况）

11.5.2 外部等待功能

能通过 WAIT# 信号将等待周期延长超过 CSi 等待控制寄存器 1 (CSiWCNT1) 设定的正常存取周期等待 (CSRWAIT[4:0] 位和 CSWAIT[4:0] 位) 和页面存取周期等待 (CSPRWAIT[2:0] 位和 CSPWAIT[2:0] 位)。

如果将 CSiMOD.EWENB 位置 “1” (允许外部等待), 就在 WAIT# 信号为 Low 电平期间插入等待周期。如果 CSiMOD.EWENB 位为 “0” (禁止外部等待), WAIT# 信号就无效。

与 WAIT# 信号无关, 插入 CSiWCNT1 寄存器设定的各周期等待。

11.5.2.1 正常存取

从 CSiWCNT1 寄存器设定的周期等待数结束 (Tend) 开始对 WAIT# 信号进行采样。在 WAIT# 信号为 Low 电平期间, 延长总线周期, 并且 WAIT# 信号变为 High 电平后的下一个周期为等待周期的结束 (Tend)。

11.5.2.2 页面存取

最初的存取和正常存取相同。从 CSiWCNT1 寄存器设定的周期等待数结束 (Tend) 开始对 WAIT# 信号进行采样。在 WAIT# 信号为 Low 电平期间, 延长总线周期, 并且 WAIT# 信号变为 High 电平后的下一个周期为等待周期的结束 (Tend)。

第 2 次以后的读存取从页面存取等待周期结束 (Tend) 开始对 WAIT# 信号进行采样。在 WAIT# 信号为 Low 电平期间, 延长页面存取的等待周期, 并且 WAIT# 信号变为 High 后的下一个周期为等待周期的结束 (Tend)。

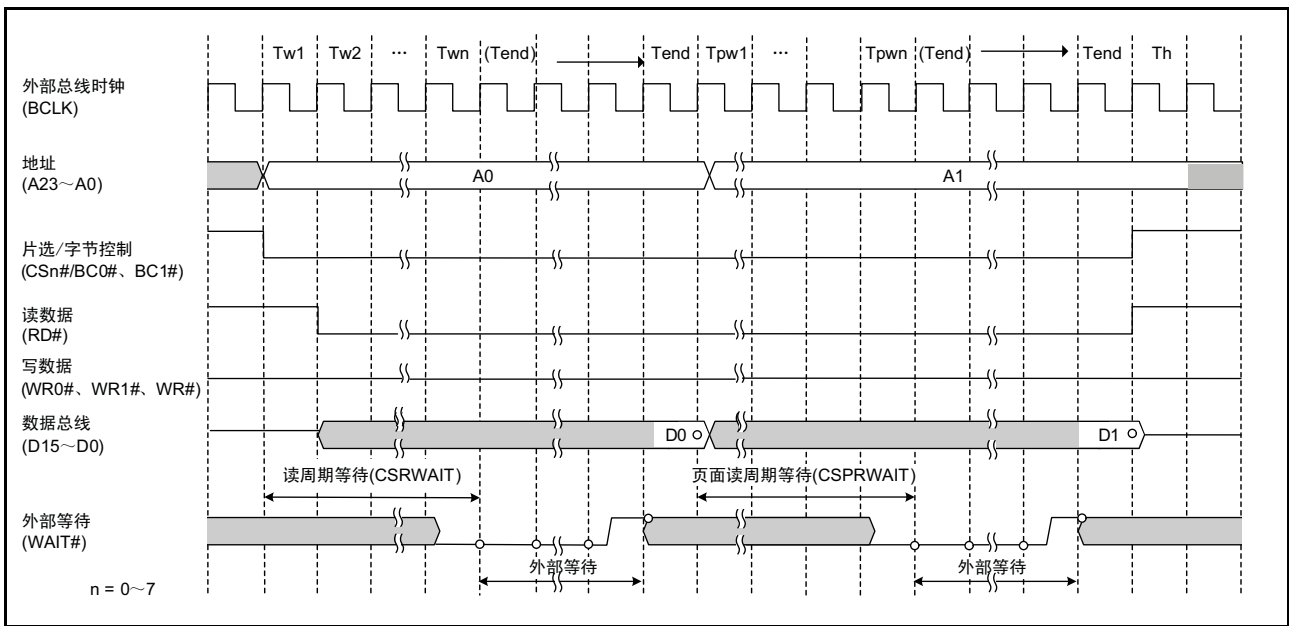


图 11.19 外部等待时序的例子 (对 16 位总线空间进行页面读存取)

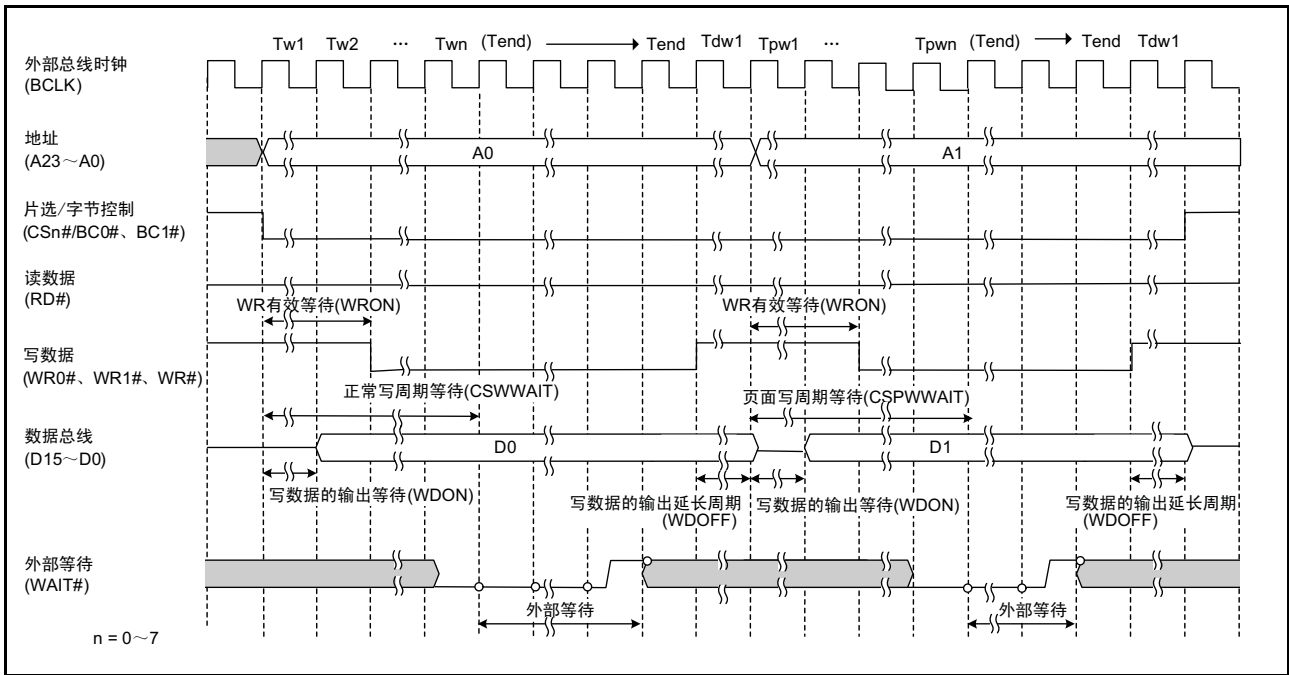


图 11.20 外部等待时序的例子（对 16 位总线空间进行页面写存取）

11.5.3 恢复周期的插入

能在连续的外部总线存取期间插入恢复周期。可插入恢复周期的条件如下所示：

- 在对外部总线进行读存取后，有外部总线的写存取时
 - 在对外部总线进行读存取后，有不同区域的读存取时
 - 在对外部总线进行写存取后，有外部总线的读存取时
- 在写存取后的写存取中，不插入恢复周期。

能在写周期后和读周期后分别设定要插入的恢复周期。通过上次总线周期的写存取区域的 CSiREC.WRCV[3:0] 位设定写周期后的恢复周期数，通过上次总线周期的读存取区域的 CSiREC.RRCV[3:0] 位设定读周期后的恢复周期数。例如，如果在 CS0 读存取后有 CS1 读存取，在此期间插入的恢复周期就为 CS0REC.RRCV[3:0] 位设定的周期数。

恢复周期的起点是上次总线周期的结束（地址保持期间的下一个周期）。在从结束开始设定的恢复周期期间插入 CSn# 信号的 High 电平期间。

在恢复周期结束后，最快在恢复周期结束的下一个周期，下一次总线存取的片选 CSn# 信号有效。即使在恢复周期期间产生下一个外部地址区的存取请求，也在等到恢复周期结束后才开始下一次外部总线的存取。

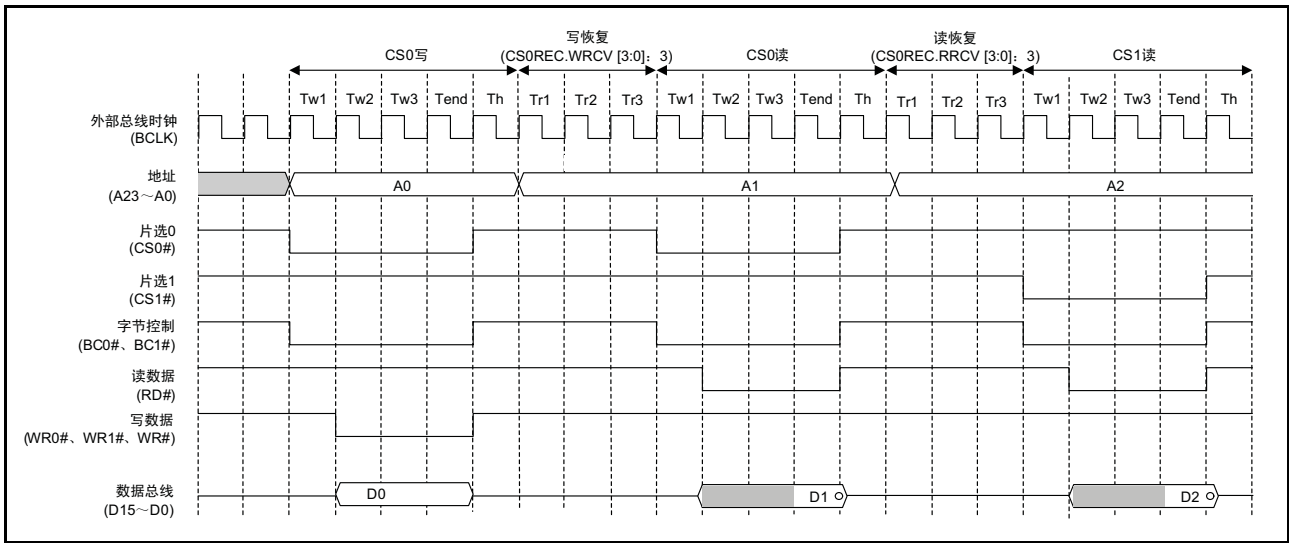


图 11.21 恢复周期的时序例子

11.5.4 写缓冲器功能

在写存取的情况下，因为通过将数据写到写缓冲器，所以能不等运行结束就释放内部主总线，开始下一次的总线存取。如果下一次总线存取为外部地址空间的存取或者外部总线控制部的寄存器的存取，就要等到上一次外部总线运行结束为止。

使用写缓冲器功能时的运行例子如图 11.22 所示。如果使用此功能并且在进行外部写操作以后有内部存取，就不等到外部写操作的结束而并行执行内部存取（内部存储器和外围模块的存取）。

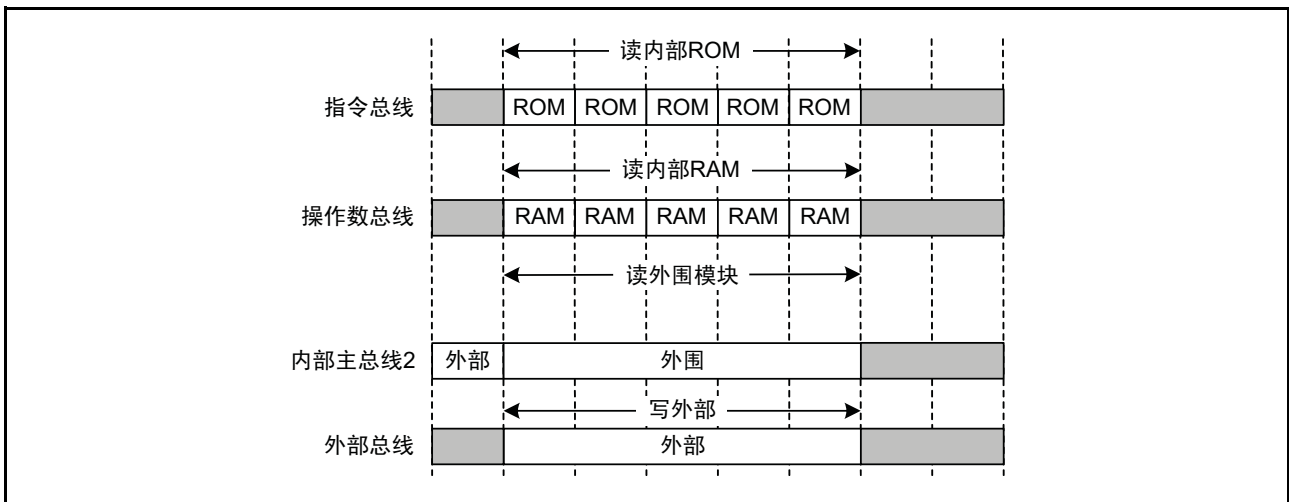


图 11.22 使用写缓冲器功能时的运行例子

11.5.5 限制事项

11.5.5.1 正常存取和页面存取时的限制事项

正常存取和页面存取时 CSi 等待控制寄存器 1 (CSiWCNT1) 和 CSi 等待控制寄存器 2 (CSiWCNT2) 的各位设定值的限制事项如表 11.9 所示。

即使 CSi 模式寄存器的页面读存取允许位有效 (CSiMOD.PRENB=1) 或者页面写存取允许位有效 (CSiMOD.PWENB=1)，在进行第 1 次页面存取或者进行不作为页面存取对象的存取时，也作为正常存取运行并且需要满足正常存取的限制条件。有关不作为页面存取对象时的详细内容，请参照“11.5.1 外部总线的存取时序”。

表 11.9 正常存取和页面存取时的限制事项

| 正常存取时的限制条件 | | 页面存取时的限制条件 | |
|--|--|--|---|
| 读 | 写 | 读 | 写 |
| CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON | CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON | CSON[2:0] ≤ CSPRWAIT RDON[2:0] ≤ CSPRWAIT CSON[2:0] ≤ RDON | CSON[2:0] ≤ CSPWWAIT WRON[2:0] ≤ CSPWWAIT WDON[2:0] ≤ CSPWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON |

11.5.5.2 禁止跨地址空间的多个区域的存取

因为禁止在 1 次存取中跨地址空间的多个区域的存取，所以无法保证运行。在地址空间的各区域边界，1 个字存取或者长字存取不能跨 2 个区域。

11.5.5.3 有关字符串操作指令的限制事项

- 外部空间有各区域的字节序转换功能（只限于数据），但是不能将字符串操作指令的操作对象数据分配到进行了不同于芯片字节序设定的区域。如果将字符串操作指令的操作对象数据分配到外部空间，就必须分配到进行了与芯片字节序相同设定的区域。
- 不能将字符串操作指令的操作对象数据分配到 I/O 寄存器。

11.5.5.4 设定寄存器时的注意事项

- 设定写数据的保持时间 (tWDH) 时的注意事项
为了确保写数据的保持时间，必须将 CSiWCNT2.WDOFF[2:0] 位（写数据的输出延长周期选择位）设定为 WDOFF[2:0] ≥ 1。必须注意：如果设定其他条件，就有可能无法确保写数据的保持时间。

11.6 总线错误监视部

总线错误监视部监视各区域的总线错误并且在发生总线错误时产生中断。

11.6.1 总线错误的种类

总线错误有非法地址存取和超时 2 种。

当存取非法区域时被检测为非法地址存取；当总线存取没有在 768 个周期以内结束时被检测为超时。

11.6.1.1 非法地址存取

当总线错误监视控制寄存器的非法地址存取检测允许位有效（BEREN.IGAEN=1）并且进行了以下存取时，发生非法地址存取。

- 对于外部区域，存取了被设定为禁止运行（CSiCNT.EXENB=0（i=0~7））的区域时
- 对于上述以外的区域，存取了非法地址区域时

非法地址存取错误的发生区域如表 11.10 所示。

11.6.1.2 超时

当总线错误监视控制寄存器的超时检测允许位有效（BEREN.TOEN=1）并且总线存取在 768 个周期以内不结束时，发生超时。此时，通过各从属模块的运行时钟对周期数进行计数。

- 对于外部区域，在开始总线存取后的 768 个周期以内总线存取不结束（WAIT#信号无效）时

注 1. RX610 群在上述以外的区域内不会发生超时。

11.6.2 发生总线错误时的运行

如果设定为向 CPU 通知发生总线错误（BERIE.CPEN=1），就能产生中断（BUSERR）。

11.6.3 总线错误的发生条件

地址空间的各区域发生的总线错误种类如表 11.10 所示。

表 11.10 发生的总线错误种类

| 地址 | 内容 | | 种类 | | | |
|-------------------------|--------------------|-----------------|--------|--------|--------|----|
| | | | 非法地址存取 | | 超时 | |
| | 有效 | 无效 | 有效 | 无效 | 有效 | 无效 |
| 0000 0000h ~ 0001 FFFFh | 内部 RAM | | — | | — | |
| 0002 0000h ~ 0007 FFFFh | 保留区 | | — | | — | |
| 0008 0000h ~ 0008 FFFFh | 外围 I/O 寄存器 | | — | | — | |
| 0009 0000h ~ 000F FFFFh | | | ○ | | — | |
| 0010 0000h ~ 0011 FFFFh | 数据闪存 | 保留区 | — | ○ | — | — |
| 0012 0000h ~ 007F 7FFFh | 保留区 | | ○ | | — | |
| 007F 8000h ~ 007F 9FFFh | FCU-RAM | | — | | — | |
| 007F A000h ~ 007F BFFFh | 保留区 | | ○ | | — | |
| 007F C000h ~ 007F C4FFh | 外围 I/O 寄存器 | | — | | — | |
| 007F C500h ~ 007F FBFFh | 保留区 | | ○ | | — | |
| 007F FC00h ~ 00FF FFFFh | 外围 I/O 寄存器 | | — | | — | |
| 0080 0000h ~ 00DF FFFFh | 保留区 | | — | | — | |
| 00E0 0000h ~ 00FF FFFFh | 内部 ROM (改写专用) | | — | | — | |
| 0100 0000h ~ 07FF FFFFh | 外部地址空间 (CS1 ~ CS7) | | ○ (注1) | | ○ (注2) | |
| 0800 0000h ~ 7FFF FFFFh | 保留区 | | ○ | | — | |
| 8000 0000h ~ FEFF FFFFh | 内部 ROM (读专用) | 保留区 | — | ○ (注1) | — | — |
| FF00 0000h ~ FFFF FFFFh | | 外部地址空间 (CS0) | — | | ○ (注2) | |

—: 不发生总线错误

○: 发生总线错误

注 1. 当将这些区域设定为禁止运行 (CSiCNT.EXENB=0 (i=0 ~ 7)) 时, 被检测为总线错误。

注 2. 当总线存取没有在 768 个周期以内结束 (WAIT# 信号无效) 时, 被检测为总线错误。

12. DMA 控制器 (DMAC)

RX610 群内置 4 个通道的 DMAC (Direct Memory Access Controller)。

DMAC 是不通过 CPU 进行数据传送的模块。

如果发生传送请求，DMAC 就将传送源地址的数据传送到传送目标地址。

12.1 概要

DMAC 的规格和框图分别如表 12.1 和图 12.1 所示。

表 12.1 DMAC 的规格

| 项目 | | 内容 | |
|-------------|----------|---|--|
| 通道数 | | 4 个通道 (DMAC _m (m=0 ~ 3)) | |
| 传送空间 | | 4G 字节 (00000000h ~ FFFFFFFFh 中除保留区以外的区域) | |
| 最大传送字节数 | | 64M 字节 | |
| DMA 启动源 | | 软件触发 外部引脚中断的触发输入 各外围功能的中断请求 | |
| 通道优先级 | | 通道 0 > 通道 1 > 通道 2 > 通道 3 (通道 0 最优先) | |
| 传送数据 | 1 个数据 | 位长: 8 位、16 位、32 位 | |
| | 1 个操作数 | 数据的个数: 1 个、2 个、4 个、8 个、16 个、32 个、64 个、128 个 | |
| 传送方式 | 操作数的传送方式 | 单个 | 对于 1 次 DMA 传送请求, 传送 1 个操作数。 在 1 个操作数传送结束后进行通道仲裁。 在 DMA 传送结束前, 每当 1 个操作数传送结束时, 需要 DMA 传送请求。 |
| | | 连续 | 对于 1 次 DMA 传送请求, 在 DMA 传送结束前逐个传送操作数。 在 1 个操作数传送结束后进行通道仲裁。 只在最初需要 DMA 传送请求。 |
| | 无停止传送方式 | 对于 1 次 DMA 传送请求, 在 DMA 传送结束前进行连续的传送。 在 DMA 传送结束前不进行通道仲裁。 只在最初需要 DMA 传送请求。 | |
| DMA 传送的开始条件 | | 一旦满足以下的全部条件, 就开始 DMA 传送: DMAC _m .DMCRE.DEN 位为 “1” (允许 DMA 传送)。 DMSCNT.DMST 位为 “1” (DMAC 运行)。 发生通道 m (DMAC _m) 的 DMA 传送请求并且通过通道仲裁取得执行权时 | |
| DMA 传送的结束条件 | | DMAC _m .DMCBC 寄存器的值变为 “0000000h” 时 | |
| 中断请求的产生时序 | | DMAC _m .DMCBC 寄存器的值变为 “0000000h” 时 | |
| 1 个数据的传送时间 | | 最短 3 个总线时钟 | |
| 选择功能 | | 重加载功能 在 DMA 传送结束时, 将传送源地址、传送目标地址和传送字节计数的重加载寄存器的值重新加载到当前寄存器。 | |

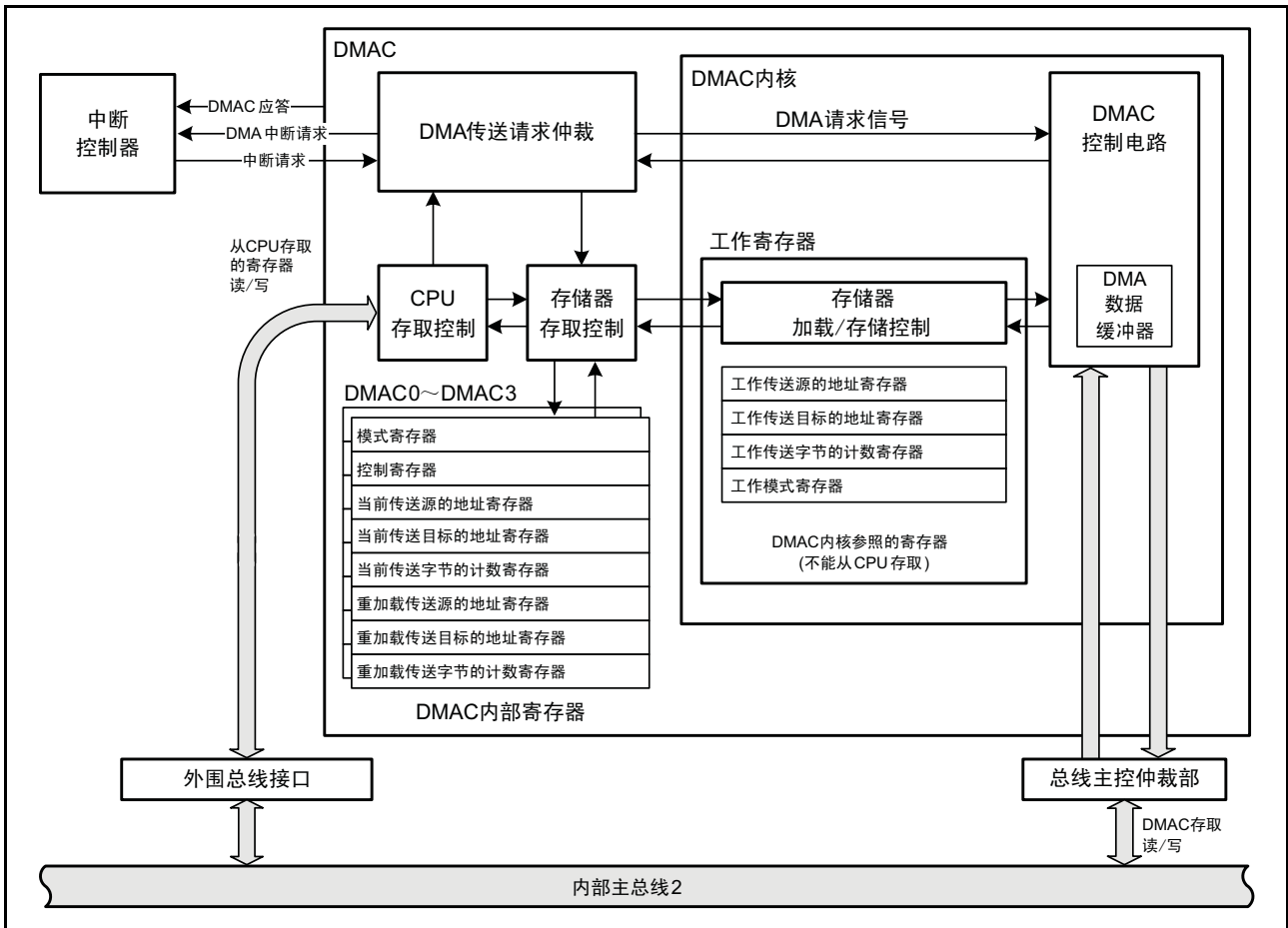


图 12.1 DMAC 的框图

12.2 寄存器说明

DMAC 的寄存器一览表如表 12.2 所示，DMAC0 ~ DMAC3 寄存器的功能相同。

表 12.2 DMAC 的寄存器一览表 (1 / 2)

| 通道 | 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|-------|-------------------|-------|------------|------------|------|
| DMAC0 | DMA 模式寄存器 | DMMOD | xxxx xxxxh | 0008 200Ch | 32 |
| | DMA 控制寄存器 A | DMCRA | 0000 0000h | 0008 2400h | 32 |
| | DMA 控制寄存器 B | DMCRB | 00h | 0008 2404h | 8 |
| | DMA 控制寄存器 C | DMCRC | 00h | 0008 2405h | 8 |
| | DMA 控制寄存器 D | DMCRD | 00h | 0008 2406h | 8 |
| | DMA 控制寄存器 E | DMCRE | 00h | 0008 2407h | 8 |
| | DMA 当前传送源的地址寄存器 | DMCSA | xxxx xxxxh | 0008 2000h | 32 |
| | DMA 当前传送目标的地址寄存器 | DMCDA | xxxx xxxxh | 0008 2004h | 32 |
| | DMA 当前传送字节的计数寄存器 | DMCBC | xxxx xxxxh | 0008 2008h | 32 |
| | DMA 重加载传送源的地址寄存器 | DMRSA | xxxx xxxxh | 0008 2200h | 32 |
| | DMA 重加载传送目标的地址寄存器 | DMRDA | xxxx xxxxh | 0008 2204h | 32 |
| | DMA 重加载传送字节的计数寄存器 | DMRBC | xxxx xxxxh | 0008 2208h | 32 |

表 12.2 DMAC 的寄存器一览表 (2/2)

| 通道 | 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|---------|-------------------|--------|------------|------------|------|
| DMAC1 | DMA 模式寄存器 | DMMOD | xxxx xxxxh | 0008 201Ch | 32 |
| | DMA 控制寄存器 A | DMCRA | 0000 0000h | 0008 2408h | 32 |
| | DMA 控制寄存器 B | DMCRB | 00h | 0008 240Ch | 8 |
| | DMA 控制寄存器 C | DMCRC | 00h | 0008 240Dh | 8 |
| | DMA 控制寄存器 D | DMCRD | 00h | 0008 240Eh | 8 |
| | DMA 控制寄存器 E | DMCRE | 00h | 0008 240Fh | 8 |
| | DMA 当前传送源的地址寄存器 | DMCSA | xxxx xxxxh | 0008 2010h | 32 |
| | DMA 当前传送目标的地址寄存器 | DMCDA | xxxx xxxxh | 0008 2014h | 32 |
| | DMA 当前传送字节的计数寄存器 | DMCBC | xxxx xxxxh | 0008 2018h | 32 |
| | DMA 重加载传送源的地址寄存器 | DMRSA | xxxx xxxxh | 0008 2210h | 32 |
| | DMA 重加载传送目标的地址寄存器 | DMRDA | xxxx xxxxh | 0008 2214h | 32 |
| | DMA 重加载传送字节的计数寄存器 | DMRBC | xxxx xxxxh | 0008 2218h | 32 |
| DMAC2 | DMA 模式寄存器 | DMMOD | xxxx xxxxh | 0008 202Ch | 32 |
| | DMA 控制寄存器 A | DMCRA | 0000 0000h | 0008 2410h | 32 |
| | DMA 控制寄存器 B | DMCRB | 00h | 0008 2414h | 8 |
| | DMA 控制寄存器 C | DMCRC | 00h | 0008 2415h | 8 |
| | DMA 控制寄存器 D | DMCRD | 00h | 0008 2416h | 8 |
| | DMA 控制寄存器 E | DMCRE | 00h | 0008 2417h | 8 |
| | DMA 当前传送源的地址寄存器 | DMCSA | xxxx xxxxh | 0008 2020h | 32 |
| | DMA 当前传送目标的地址寄存器 | DMCDA | xxxx xxxxh | 0008 2024h | 32 |
| | DMA 当前传送字节的计数寄存器 | DMCBC | xxxx xxxxh | 0008 2028h | 32 |
| | DMA 重加载传送源的地址寄存器 | DMRSA | xxxx xxxxh | 0008 2220h | 32 |
| | DMA 重加载传送目标的地址寄存器 | DMRDA | xxxx xxxxh | 0008 2224h | 32 |
| | DMA 重加载传送字节的计数寄存器 | DMRBC | xxxx xxxxh | 0008 2228h | 32 |
| DMAC3 | DMA 模式寄存器 | DMMOD | xxxx xxxxh | 0008 203Ch | 32 |
| | DMA 控制寄存器 A | DMCRA | 0000 0000h | 0008 2418h | 32 |
| | DMA 控制寄存器 B | DMCRB | 00h | 0008 241Ch | 8 |
| | DMA 控制寄存器 C | DMCRC | 00h | 0008 241Dh | 8 |
| | DMA 控制寄存器 D | DMCRD | 00h | 0008 241Eh | 8 |
| | DMA 控制寄存器 E | DMCRE | 00h | 0008 241Fh | 8 |
| | DMA 当前传送源的地址寄存器 | DMCSA | xxxx xxxxh | 0008 2030h | 32 |
| | DMA 当前传送目标的地址寄存器 | DMCDA | xxxx xxxxh | 0008 2034h | 32 |
| | DMA 当前传送字节的计数寄存器 | DMCBC | xxxx xxxxh | 0008 2038h | 32 |
| | DMA 重加载传送源的地址寄存器 | DMRSA | xxxx xxxxh | 0008 2230h | 32 |
| | DMA 重加载传送目标的地址寄存器 | DMRDA | xxxx xxxxh | 0008 2234h | 32 |
| | DMA 重加载传送字节的计数寄存器 | DMRBC | xxxx xxxxh | 0008 2238h | 32 |
| DMAC 通用 | DMA 中断控制寄存器 | DMICNT | 00h | 0008 250Bh | 8 |
| | DMA 启动控制寄存器 | DMSCNT | 00h | 0008 2502h | 8 |
| | DMA 仲裁状态寄存器 | DMASTS | 00h | 0008 251Bh | 8 |
| | DMA 传送结束的检测寄存器 | DMEDET | 00h | 0008 2517h | 8 |

x: 不定值

12.2.1 DMA 模式寄存器 (DMMOD)

地址 DMAC0.DMMOD 0008 200Ch、DMAC1.DMMOD 0008 201Ch
DMAC2.DMMOD 0008 202Ch、DMAC3.DMMOD 0008 203Ch

| | | | | | | | | | | | | | | | | |
|-------|-----|-----------|-----|-----|------------|-----------|-----|-----|-----|-----|-----|-----|-----|------------|-----|-----|
| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| | — | — | — | — | OPSEL[3:0] | | | | — | — | — | — | — | SZSEL[2:0] | | |
| 复位后的值 | 0 | 0 | 0 | 0 | x | x | x | x | 0 | 0 | 0 | 0 | 0 | x | x | x |
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | SMOD[2:0] | | | — | DMOD[2:0] | | | — | — | — | — | — | — | — | — |
| 复位后的值 | 0 | x | x | x | 0 | x | x | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

x: 不定值

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|------------|----------------|--|-----|
| b7-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b10-b8 | DMOD[2:0] | 传送目标地址的增加方向选择位 | b10 b8 0 0 0: 固定 0 0 1: 正方向 0 1 0: 负方向 0 1 1: 循环 上述以外: 不能设定 | R/W |
| b11 | — | 保留位 | 读写值都为“0”。 | R/W |
| b14-b12 | SMOD[2:0] | 传送源地址的增加方向选择位 | b14 b12 0 0 0: 固定 0 0 1: 正方向 0 1 0: 负方向 0 1 1: 循环 上述以外: 不能设定 | R/W |
| b15 | — | 保留位 | 读写值都为“0”。 | R/W |
| b18-b16 | SZSEL[2:0] | 传送数据的位长选择位 | b18 b16 0 0 0: 8 位 0 0 1: 16 位 0 1 0: 32 位 上述以外: 不能设定 | R/W |
| b23-b19 | — | 保留位 | 读写值都为“0”。 | R/W |
| b27-b24 | OPSEL[3:0] | 操作数的传送数据个数选择位 | b27 b24 0 0 0 0: 1 个数据 0 0 0 1: 2 个数据 0 0 1 0: 4 个数据 0 0 1 1: 8 个数据 0 1 0 0: 16 个数据 0 1 0 1: 32 个数据 0 1 1 0: 64 个数据 0 1 1 1: 128 个数据 上述以外: 不能设定 | R/W |
| b31-b28 | — | 保留位 | 读写值都为“0”。 | R/W |

DMMOD 寄存器是设定传送源或者传送目标地址的增加方向以及传送数据长度的寄存器。

在设定 DMMOD 寄存器时，必须在 DMAC 停止或者禁止 DMA 传送时而不能在数据传送过程中进行写操作。

必须对 DMMOD 寄存器进行 32 位的存取。

DMOD[2:0] 位 (传送目标地址的增加方向选择位)

SMOD[2:0] 位 (传送源地址的增加方向选择位)

这些位设定 DMA 传送过程中地址的增加方向。

在增加方向选择了循环时，如果地址进行正方向增加并且在 1 个操作数传送结束时，这些位的值就为开始 DMA 传送时的设定值。

如表 12.3 所示，地址的增加值取决于位长。

表 12.3 地址的增加方向和位长的增减值

| SZSEL[2:0] 位 | SMOD[2:0] 位、DMOD[2:0] 位 | | | |
|---------------|-------------------------|-----------------|-----------------|----------------|
| | “000b” (固定) | “001b” (正方向) | “010b” (反方向) | “011b” (循环) |
| “000b” (8 位) | ±0 | +1 | -1 | +1 |
| “001b” (16 位) | ±0 | +2 | -2 | +2 |
| “010b” (32 位) | ±0 | +4 | -4 | +4 |

SZSEL[2:0] 位 (传送数据的位长选择位)

这些位设定传送数据的位长。

OPSEL[3:0] 位 (操作数的传送数据个数选择位)

这些位设定 1 个操作数要传送的数据个数。

在使用操作数传送方式时，将 OPSEL[3:0] 位设定的个数的数据作为 1 个操作数进行连续的传送。

在使用无停止传送方式时，OPSEL[3:0] 位的设定无效，连续传送由 DMA 当前传送字节的计数寄存器 (DMACm.DMCBC) 设定的字节数的数据。

12.2.2 DMA 控制寄存器 A (DMCRA)

地址 DMAC0.DMCRA 0008 2400h、DMAC1.DMCRA 0008 2408h
DMAC2.DMCRA 0008 2410h、DMAC3.DMCRA 0008 2418h

| | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-------|-----------|-------|-----|-----|-----------|-----|-----|-----|-----|-----|
| | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
| | — | — | — | — | — | — | DSEL[1:0] | | — | — | — | — | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | — | — | — | BRL0D | SRL0D | DRLOD | — | — | DCTG[5:0] | | | | | |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|---------|-----------|-----------------|--|-----|
| b5-b0 | DCTG[5:0] | DMA 启动源选择位 | 设定 DMA 启动源 (参照表 12.4)。 | R/W |
| b7-b6 | — | 保留位 | 读写值都为“0”。 | R/W |
| b8 | DRLOD | 传送目标地址的重加载功能选择位 | 0: 不使用传送目标地址的重加载功能 1: 使用传送目标地址的重加载功能 | R/W |
| b9 | SRL0D | 传送源地址的重加载功能选择位 | 0: 不使用传送源地址的重加载功能 1: 使用传送源地址的重加载功能 | R/W |
| b10 | BRL0D | 传送字节计数的重加载功能选择位 | 0: 不使用传送字节计数的重加载功能 1: 使用传送字节计数的重加载功能 | R/W |
| b23-b11 | — | 保留位 | 读写值都为“0”。 | R/W |
| b25-b24 | DSEL[1:0] | 传送方式选择位 | b25 b24 0 0: 单个操作数传送 0 1: 连续操作数传送 1 0: 不能设定 1 1: 无停止传送 | R/W |
| b31-b26 | — | 保留位 | 读写值都为“0”。 | R/W |

DMCRA 寄存器是控制 DMAC 功能的寄存器。

DCTG[5:0] 位 (DMA 启动源选择位)

这些位设定 DMA 启动源。

在设定 DCTG[5:0] 位时, 必须在 DMAC 停止或者禁止 DMA 传送时而不能在数据传送过程中进行写操作。

如果设定 DCTG[5:0] 位, 就必须在将 DMACm.DMCRD.DREQ 位置“0”(无 DMA 传送请求)后设定为 DMAC 运行并且允许 DMA 传送。

DCTG[5:0] 位的设定如表 12.4 所示。

表 12.4 DCTG[5:0] 位的设定

| DCTG[5:0] 位 | DMA 启动源 | | | |
|-----------------|---|------|------|------|
| | DMA0 | DMA1 | DMA2 | DMA3 |
| 000000 | 软件触发 | | | |
| 000001 | CMI0 (比较匹配定时器单元 0 的 CMT0 比较匹配中断) | | | |
| 000010 | CMI1 (比较匹配定时器单元 0 的 CMT1 比较匹配中断) | | | |
| 000011 | CMI2 (比较匹配定时器单元 1 的 CMT2 比较匹配中断) | | | |
| 000100 | CMI3 (比较匹配定时器单元 1 的 CMT3 比较匹配中断) | | | |
| 000101 | IRQ0 (外部引脚中断) | | | |
| 000110 | IRQ1 (外部引脚中断) | | | |
| 000111 | IRQ2 (外部引脚中断) | | | |
| 001000 | IRQ3 (外部引脚中断) | | | |
| 001001 | ADI0 (A/D 转换器单元 0 的 ADC 中断) | | | |
| 001010 | ADI1 (A/D 转换器单元 1 的 ADC 中断) | | | |
| 001011 | ADI2 (A/D 转换器单元 2 的 ADC 中断) | | | |
| 001100 | ADI3 (A/D 转换器单元 3 的 ADC 中断) | | | |
| 001101 | TGI0A (16 位定时器脉冲单元的单元 0 的 TPU0 输入捕捉 / 比较匹配中断) | | | |
| 001110 | TGI1A (16 位定时器脉冲单元的单元 0 的 TPU1 输入捕捉 / 比较匹配中断) | | | |
| 001111 | TGI2A (16 位定时器脉冲单元的单元 0 的 TPU2 输入捕捉 / 比较匹配中断) | | | |
| 010000 | TGI3A (16 位定时器脉冲单元的单元 0 的 TPU3 输入捕捉 / 比较匹配中断) | | | |
| 010001 | TGI4A (16 位定时器脉冲单元的单元 0 的 TPU4 输入捕捉 / 比较匹配中断) | | | |
| 010010 | TGI5A (16 位定时器脉冲单元的单元 0 的 TPU5 输入捕捉 / 比较匹配中断) | | | |
| 010011 | TGI6A (16 位定时器脉冲单元的单元 1 的 TPU6 输入捕捉 / 比较匹配中断) | | | |
| 010100 | TGI7A (16 位定时器脉冲单元的单元 1 的 TPU7 输入捕捉 / 比较匹配中断) | | | |
| 010101 | TGI8A (16 位定时器脉冲单元的单元 1 的 TPU8 输入捕捉 / 比较匹配中断) | | | |
| 010110 | TGI9A (16 位定时器脉冲单元的单元 1 的 TPU9 输入捕捉 / 比较匹配中断) | | | |
| 010111 | TGI10A (16 位定时器脉冲单元的单元 1 的 TPU10 输入捕捉 / 比较匹配中断) | | | |
| 011000 | TGI11A (16 位定时器脉冲单元的单元 1 的 TPU11 输入捕捉 / 比较匹配中断) | | | |
| 011001 | RXI0 (串行通信接口 SCI0 的接收数据满中断) | | | |
| 011010 | TXI0 (串行通信接口 SCI0 的发送数据空中断) | | | |
| 011011 | RXI1 (串行通信接口 SCI1 的接收数据满中断) | | | |
| 011100 | TXI1 (串行通信接口 SCI1 的发送数据空中断) | | | |
| 011101 | RXI2 (串行通信接口 SCI2 的接收数据满中断) | | | |
| 011110 | TXI2 (串行通信接口 SCI2 的发送数据空中断) | | | |
| 011111 | RXI3 (串行通信接口 SCI3 的接收数据满中断) | | | |
| 100000 | TXI3 (串行通信接口 SCI3 的发送数据空中断) | | | |
| 100001 | RXI4 (串行通信接口 SCI4 的接收数据满中断) | | | |
| 100010 | TXI4 (串行通信接口 SCI4 的发送数据空中断) | | | |
| 100011 | RXI5 (串行通信接口 SCI5 的接收数据满中断) | | | |
| 100100 | TXI5 (串行通信接口 SCI5 的发送数据空中断) | | | |
| 100101 | RXI6 (串行通信接口 SCI6 的接收数据满中断) | | | |
| 100110 | TXI6 (串行通信接口 SCI6 的发送数据空中断) | | | |
| 100111 | ICRXI0 (I ² C 总线接口 RIIC0 的接收数据满中断) | | | |
| 101000 | ICTXI0 (I ² C 总线接口 RIIC0 的发送数据空中断) | | | |
| 101001 | ICRXI1 (I ² C 总线接口 RIIC1 的接收数据满中断) | | | |
| 101010 | ICTXI1 (I ² C 总线接口 RIIC1 的发送数据空中断) | | | |
| 101011 ~ 111111 | 什么也不分配, 并且不能选择。 | | | |

要向 DMAC 通知 DMA 传送请求时，需要将 DMA 启动源设定的中断请求置为有效，将中断控制器 (ICU) 的 ICU.IERi 寄存器 (i=02h ~ 1Fh) 的对应位置 “1” (允许中断请求) 并且通过 ICU.ISELRi 寄存器 (i= 中断向量号) 将中断传输目标设定为 DMAC。详细内容请参照以下所示的中断控制器和各外围模块的章节。

- “10. 中断控制器 (ICU)”
- “15. 16位定时器脉冲单元 (TPU)”
- “18. 比较匹配定时器 (CMT)”
- “20. 串行通信接口 (SCI)”
- “22. I²C 总线接口 (RIIC)”
- “23. A/D 转换器”

DRLOD 位 (传送目标地址的重加载功能选择位)

此位控制传送目标地址的重加载功能。

如果将 DRLOD 位置 “1”，就在 DMA 传送结束时将 DMA 重加载传送目标的地址寄存器 (DMACm.DMRDA) 的内容重新加载到 DMA 当前传送目标的地址寄存器 (DMACm.DMCDA)。

在不使用重加载功能时，必须将 DMACm.DMCRC.ECLR 位置 “1” (在 DMA 传送结束时，将 DEN 位置 “0”)，使 DMACm.DMCRE.DEN 变为 “0” (禁止 DMA 传送)。

SRLOD 位 (传送源地址的重加载功能选择位)

此位控制传送源地址的重加载功能。

如果将 SRLOD 位置 “1”，就在 DMA 传送结束时将 DMA 重加载传送源的地址寄存器 (DMACm.DMRSA) 的内容重新加载到 DMA 当前传送源的地址寄存器 (DMACm.DMCSA)。

在不使用重加载功能时，必须将 DMACm.DMCRC.ECLR 位置 “1”，使 DMACm.DMCRE.DEN 位变为 “0”。

BRLOD 位 (传送字节计数的重加载功能选择位)

此位控制传送字节计数的重加载功能。

如果将 BRLOD 位置 “1”，就在 DMA 传送结束时将 DMA 重加载传送字节的计数寄存器 (DMACm.DMRBC) 的内容重新加载到 DMA 当前传送字节的计数寄存器 (DMACm.DMCBC)。

在不使用重加载功能时，必须将 DMACm.DMCRC.ECLR 位置 “1”，使 DMACm.DMCRE.DEN 位变为 “0”。

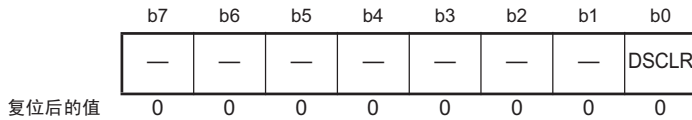
DSEL[1:0] 位 (传送方式选择位)

这些位设定传送方式。

在设定 DSEL[1:0] 位时，必须在 DMAC 停止或者禁止 DMA 传送时而不能在数据传送过程中进行写操作。

12.2.3 DMA 控制寄存器 B (DMCRB)

地址 DMAC0.DMCRB 0008 2404h、DMAC1.DMCRB 0008 240Ch
DMAC2.DMCRB 0008 2414h、DMAC3.DMCRB 0008 241Ch



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|----------------|---|-----|
| b0 | DSCLR | DMAC 内部状态的初始化位 | 如果给此位写“1”，就开始对 DMAC 内部状态进行初始化。写“0”无效，读取值为“0”。 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

DMCRB 寄存器是控制 DMA 传送的寄存器。

DSCLR 位 (DMAC 内部状态的初始化位)

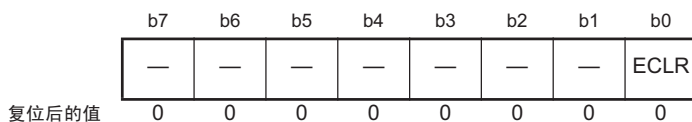
此位是对 DMAC 内部状态进行初始化的位。

如果在暂停 DMA 传送后将 DSCLR 位置“1”，就中止剩余的 DMA 传送并且对 DMAC 内部传送状态进行初始化。但是，此时不对各寄存器进行初始化。因为不保持被写的“1”，所以读取值总是“0”，写“0”无效。

在设定 DSCLR 位时，必须在 DMAC 停止或者禁止 DMA 传送时而不能在数据传送过程中进行写操作。

12.2.4 DMA 控制寄存器 C (DMCRC)

地址 DMAC0.DMCRC 0008 2405h、DMAC1.DMCRC 0008 240Dh
DMAC2.DMCRC 0008 2415h、DMAC3.DMCRC 0008 241Dh



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|-------------|---|-----|
| b0 | ECLR | DMA 传送允许清除位 | 0: 在 DMA 传送结束时，不将 DEN 位置“0”。 1: 在 DMA 传送结束时，将 DEN 位置“0”。 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

DMCRC 寄存器是控制 DMA 传送的寄存器。

ECLR 位 (DMA 传送允许清除位)

此位控制 DMA 传送结束时的 DMACm.DMCRE.DEN 位。

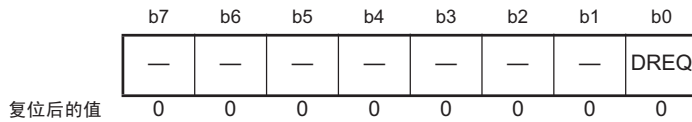
如果将 ECLR 位置“1”，DMACm.DMCRE.DEN 位就在 DMA 传送结束时变为“0”，此后不进行该通道的 DMA 传送。

在不使用重加载功能时，必须将 ECLR 位置“1”，使 DMACm.DMCRE.DEN 位变为“0”。

在设定 ECLR 位时，不能在数据传送过程中进行写操作。

12.2.5 DMA 控制寄存器 D (DMCRD)

地址 DMAC0.DMCRD 0008 2406h、DMAC1.DMCRD 0008 240Eh
DMAC2.DMCRD 0008 2416h、DMAC3.DMCRD 0008 241Eh



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|-----------|--------------------------------|-----|
| b0 | DREQ | DMA 传送请求位 | 0: 无 DMA 传送请求 1: 有 DMA 传送请求 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

DMCRD 寄存器是控制 DMA 传送的寄存器。

DREQ 位 (DMA 传送请求位)

此位是表示有无 DMA 传送请求的位。

即使在 DMAC 停止或者禁止 DMA 传送的状态下，DREQ 位也因 DMA 传送请求的有无而发生变化。

在将软件触发设定为 DMA 启动源的情况下，如果通过程序给 DREQ 位写“1”，就产生 DMA 传送请求。

如果 DMA 启动源不是软件触发，就不能通过程序给 DREQ 位写“1”。

在给 DREQ 位写“0”时，必须在 DMAC 停止或者禁止 DMA 传送时而不能在数据传送过程中进行写操作。

DREQ 位因 DMA 启动源而不同，如下所示：

1. DMA 启动源为软件触发的情况

[为“1”的条件]

- 通过程序写“1”时

[为“0”的条件]

- 通过程序写“0”时
- 在接受 DMA 传送请求后开始数据传送。

2. DMA 启动源不是软件触发的情况

[为“1”的条件]

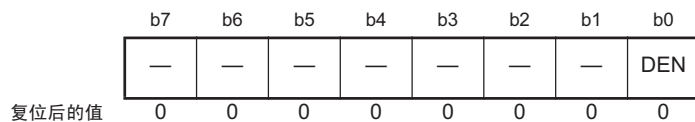
- 检测到 DMACm.DMCRA.DCTG[5:0] 位选择的 DMA 传送请求时

[为“0”的条件]

- 通过程序写“0”时
- 在接受 DMA 传送请求后开始数据传送时

12.2.6 DMA 控制寄存器 E (DMCRE)

地址 DMAC0.DMCRE 0008 2407h、DMAC1.DMCRE 0008 240Fh
DMAC2.DMCRE 0008 2417h、DMAC3.DMCRE 0008 241Fh



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-----|-----------|------------------------------|-----|
| b0 | DEN | DMA 传送允许位 | 0: 禁止 DMA 传送 1: 允许 DMA 传送 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

DMCRE 寄存器是控制 DMA 传送的寄存器。

DEN 位 (DMA 传送允许位)

此位是允许 DMA 传送的位。

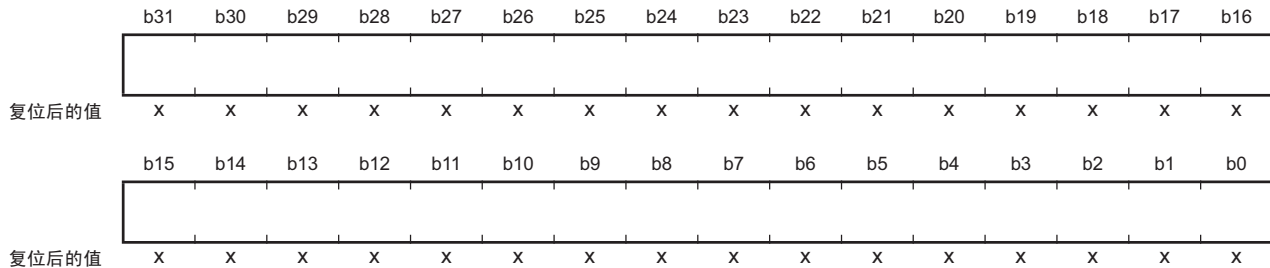
如果在 DMACm.DMCRC.ECLR 位为“1”时结束 DMA 传送，DEN 位就自动变为“0”。

如果在操作数传送方式的传送过程中将 DEN 位置“0”，该通道就在结束正在传送的 1 个操作数后暂停 DMA 传送。此后，如果将此位置“1”，就重新开始 DMA 传送。

在无停止传送过程中，即使 DEN 位变为“0”，也不暂停传送而一直传送到 DMA 传送结束为止。

12.2.7 DMA 当前传送源的地址寄存器 (DMCSA)

地址 DMAC0.DMCSA 0008 2000h, DMAC1.DMCSA 0008 2010h
DMAC2.DMCSA 0008 2020h, DMAC3.DMCSA 0008 2030h



x: 不定值

| 位 | 功能 | 设定范围 | R/W |
|--------|-------------|-------------------------------|-----|
| b31-b0 | 设定传送源的起始地址。 | 00000000h ~ FFFFFFFFh (4G 字节) | R/W |

DMCSA 寄存器是设定传送源起始地址的寄存器。

在设定 DMACm.DMCSA 寄存器时, 必须在 DMAC 停止或者禁止 DMA 传送时而不能在数据传送过程中进行写操作。

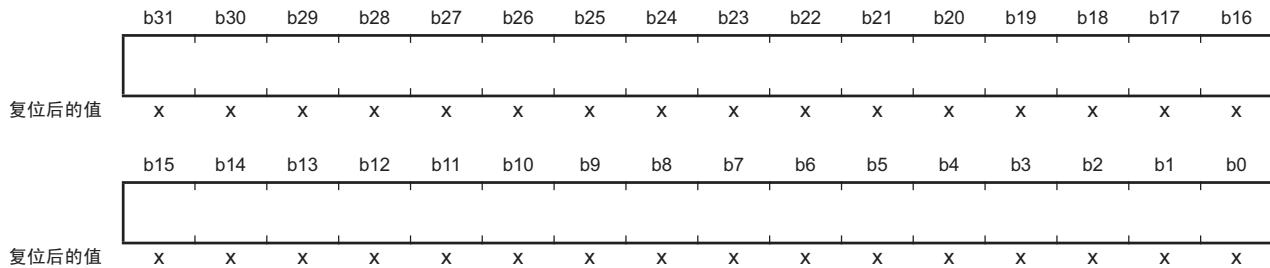
必须对 DMACm.DMCSA 寄存器进行 32 位的存取。

在给 DMACm.DMCSA 寄存器设定地址时, 必须在位长为 16 位时设定 2 的倍数, 在位长为 32 位时设定 4 的倍数, 使 b31 ~ b0 对应 A31 ~ A0。

在开始 DMA 传送时, 将 DMACm.DMCSA 寄存器的设定值传送到 DMAC 内核的工作寄存器, 并且在 1 个操作数传送结束或者 DMA 传送结束时恢复工作寄存器的值。如果 DMACm.DMMOD.SMOD[2:0] 位为 “011b” (循环), 就不恢复工作寄存器的值而保持 DMA 传送开始时的设定值。当 DMACm.DMCRA.SRL0D 位为 “1” (使用传送源地址的重加载功能) 时, 如果 DMA 传送结束, 就重新加载 DMACm.DMCSA 寄存器的值。

12.2.8 DMA 当前传送目标的地址寄存器 (DMCDA)

地址 DMAC0.DMCDA 0008 2004h、DMAC1.DMCDA 0008 2014h
DMAC2.DMCDA 0008 2024h、DMAC3.DMCDA 0008 2034h



x: 不定值

| 位 | 功能 | 设定范围 | R/W |
|--------|--------------|-------------------------------|-----|
| b31-b0 | 设定传送目标的起始地址。 | 00000000h ~ FFFFFFFFh (4G 字节) | R/W |

DMCDA 寄存器是设定传送目标起始地址的寄存器。

在设定 DMACm.DMCDA 寄存器时，必须在 DMAC 停止或者禁止 DMA 传送时而不能在数据传送过程中进行写操作。

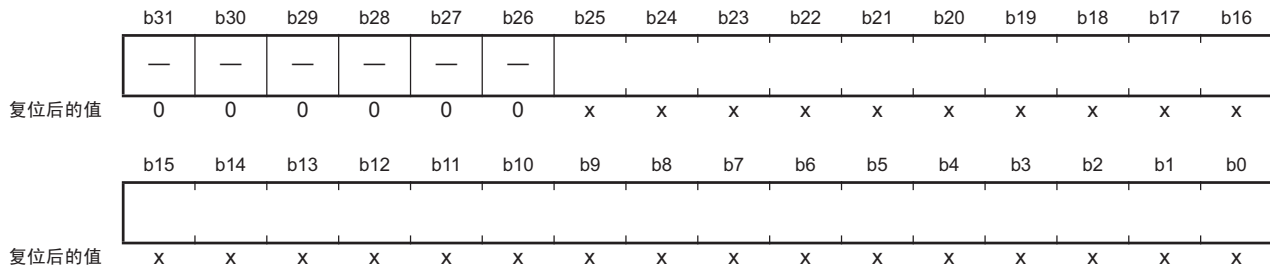
必须对 DMACm.DMCDA 寄存器进行 32 位的存取。

在给 DMACm.DMCDA 寄存器设定地址时，必须在位长为 16 位时设定 2 的倍数，在位长为 32 位时设定 4 的倍数，使 b31 ~ b0 对应 A31 ~ A0。

在开始 DMA 传送时，将 DMACm.DMCDA 寄存器的设定值传送到 DMAC 内核的工作寄存器，并且在 1 个操作数传送结束或者 DMA 传送结束时恢复工作寄存器的值。如果 DMACm.DMMOD.DMOD[2:0] 位为 “011b” (循环)，就不恢复工作寄存器的值而保持 DMA 传送开始时的设定值。当 DMACm.DMCRA.DRLOD 位为 “1” (使用传送目标地址的重加载功能) 时，如果 DMA 传送结束，就重新加载 DMACm.DMRDA 寄存器的值。

12.2.9 DMA 当前传送字节的计数寄存器 (DMCBC)

地址 DMAC0.DMCBC 0008 2008h、DMAC1.DMCBC 0008 2018h
DMAC2.DMCBC 0008 2028h、DMAC3.DMCBC 0008 2038h



x: 不定值

| 位 | 功能 | 设定范围 | R/W |
|---------|---------------|---------------------|-----|
| b25-b0 | 设定 DMA 传送字节数。 | 0000000h ~ 3FFFFFFh | R/W |
| b31-b26 | 保留位 | 读写值都为“0”。 | R/W |

DMCBC 寄存器是设定 DMA 传送字节数的寄存器。

在设定 DMACm.DMCBC 寄存器时，必须在 DMAC 停止或者禁止 DMA 传送时而不能在数据传送过程中进行写操作。

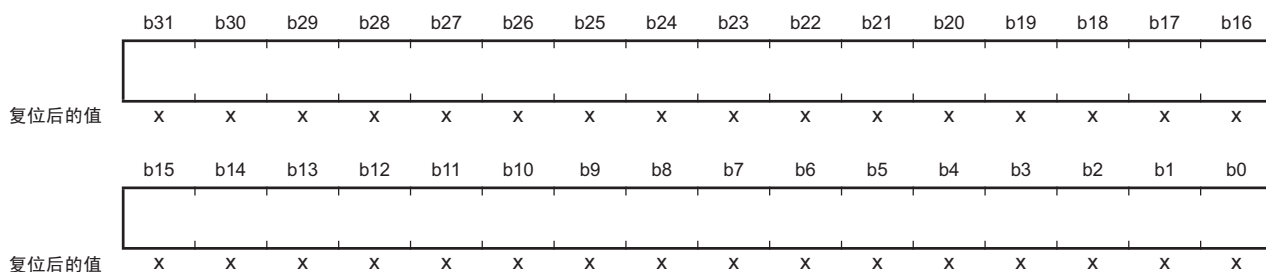
必须对 DMACm.DMCBC 寄存器进行 32 位的存取。

在给 DMACm.DMCBC 寄存器设定地址时，必须在位长为 16 位时设定 2 的倍数，在位长为 32 位时设定 4 的倍数。如果设定为“0000000h”，传送字节数就为 64M 字节。

在开始 DMA 传送时，将 DMACm.DMCBC 寄存器的设定值传送到 DMAC 内核的工作寄存器。每传送 1 个数据，工作寄存器的值就减少传送的字节数（当数据的位长为 8 位时减 1；当位长为 16 位时减 2；当位长为 32 位时减 4），如果减少到“0000000h”，DMA 传送就结束。在 1 个操作数传送结束或者 DMA 传送结束时恢复工作寄存器的值。如果 DMACm.DMCRA.BRLOD 位为“1”（使用传送字节计数的重加载功能），就在 DMA 传送结束时重新加载 DMACm.DMRBC 寄存器的值。

12.2.10 DMA 重加载传送源的地址寄存器 (DMRSA)

地址 DMAC0.DMRSA 0008 2200h、DMAC1.DMRSA 0008 2210h
DMAC2.DMRSA 0008 2220h、DMAC3.DMRSA 0008 2230h



x: 不定值

| 位 | 功能 | 设定范围 | R/W |
|--------|------------------------------|-------------------------------|-----|
| b31-b0 | 设定要重新加载到 DMACm.DMCSA 寄存器的地址。 | 00000000h ~ FFFFFFFFh (4G 字节) | R/W |

DMRSA 寄存器设定要重新加载到 DMCSA 寄存器的地址。

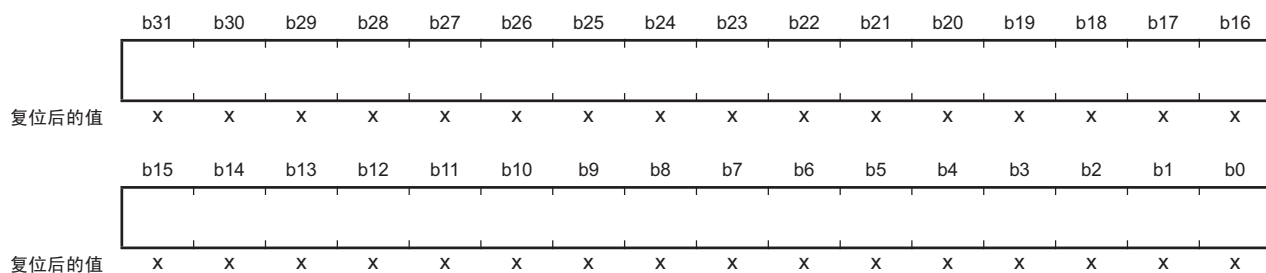
必须对 DMACm.DMRSA 寄存器进行 32 位的存取。

当 DMACm.DMCRA.SRL0D 位为“1”（使用传送源地址的重加载功能）时，在 DMA 传送结束时将 DMACm.DMRSA 寄存器的值重新加载到 DMACm.DMCSA 寄存器。

在给 DMACm.DMRSA 寄存器设定地址时，必须在位长为 16 位时设定 2 的倍数，在位长为 32 位时设定 4 的倍数，使 b31 ~ b0 对应 A31 ~ A0。

12.2.11 DMA 重加载传送目标的地址寄存器 (DMRDA)

地址 DMAC0.DMRDA 0008 2204h、DMAC1.DMRDA 0008 2214h
DMAC2.DMRDA 0008 2224h、DMAC3.DMRDA 0008 2234h



x: 不定值

| 位 | 功能 | 设定范围 | R/W |
|--------|------------------------------|-------------------------------|-----|
| b31-b0 | 设定要重新加载到 DMACm.DMCDA 寄存器的地址。 | 00000000h ~ FFFFFFFFh (4G 字节) | R/W |

DMRDA 寄存器设定要重新加载到 DMCDA 寄存器的地址。

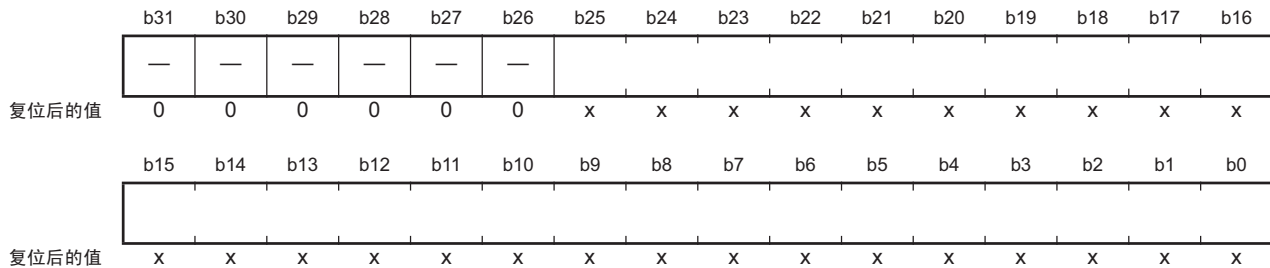
必须对 DMACm.DMRDA 寄存器进行 32 位的存取。

当 DMACm.DMCRA.DRL0D 位为“1”（使用传送目标地址的重加载功能）时，在 DMA 传送结束时将 DMACm.DMRDA 寄存器的值重新加载到 DMACm.DMCDA 寄存器。

在给 DMACm.DMRDA 寄存器设定地址时，必须在位长为 16 位时设定 2 的倍数，在位长为 32 位时设定 4 的倍数，使 b31 ~ b0 对应 A31 ~ A0。

12.2.12 DMA 重加载传送字节的计数寄存器 (DMRBC)

地址 DMAC0.DMRBC 0008 2208h、DMAC1.DMRBC 0008 2218h
DMAC2.DMRBC 0008 2228h、DMAC3.DMRBC 0008 2238h



x: 不定值

| 位 | 功能 | 设定范围 | R/W |
|---------|--------------------------------------|---------------------|-----|
| b25-b0 | 设定要重新加载到 DMACm.DMCBC 寄存器的 DMA 传送字节数。 | 0000000h ~ 3FFFFFFh | R/W |
| b31-b26 | 保留位 | 读写值都为“0”。 | R/W |

DMRBC 寄存器设定要重新加载到 DMCBC 寄存器的 DMA 传送字节数。

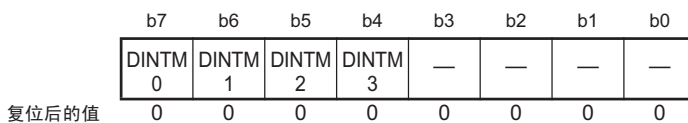
必须对 DMACm.DMRBC 寄存器进行 32 位的存取。

当 DMACm.DMCRA.BRLOD 位为“1”（使用传送字节计数的重加载功能）时，在 DMA 传送结束时将 DMACm.DMRBC 寄存器的值重新加载到 DMACm.DMCBC 寄存器。

在给 DMACm.DMRBC 寄存器设定地址时，必须在位长为 16 位时设定 2 的倍数，在位长为 32 位时设定 4 的倍数。如果设定为“0000000h”，传送字节数就为 64M 字节。

12.2.13 DMA 中断控制寄存器 (DMICNT)

地址 0008 250Bh



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|--------|------------|--------------------|-----|
| b3-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b4 | DINTM3 | DMA3 中断允许位 | 0: 禁止中断 1: 允许中断 | R/W |
| b5 | DINTM2 | DMA2 中断允许位 | | R/W |
| b6 | DINTM1 | DMA1 中断允许位 | | R/W |
| b7 | DINTM0 | DMA0 中断允许位 | | R/W |

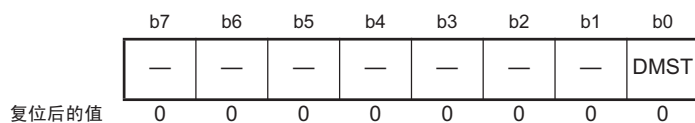
DMICNT 寄存器是允许各通道 DMA_m 中断请求 (DMTEND_m) (m=0 ~ 3) 的寄存器。

DINTM_m 位 (DMA_m 中断允许位) (m=0 ~ 3)

如果将 DINTM_m 位置“1”，就在通道 m 的 DMA 传送结束时产生 DMA_m 中断请求 (DMTEND_m)。如果将 DINTM_m 位置“0”，就不产生 DMA_m 中断请求 (DMTEND_m)。

12.2.14 DMA 启动控制寄存器 (DMSCNT)

地址 0008 2502h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|----------|--------------------------|-----|
| b0 | DMST | DMAC 启动位 | 0: DMAC 停止 1: DMAC 运行 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

DMSCNT 寄存器是启动 DMAC 的寄存器。

DMST 位 (DMAC 启动位)

如果将 DMST 位置“1”，就启动 DMAC。

如果在操作数传送方式的传送过程中将 DMST 位置“0”，全部通道就在结束正在传送 1 个操作数后暂停 DMA 传送。此后，如果将此位置“1”，就重新开始 DMA 传送。

在无停止传送过程中，即使 DMST 位变为“0”，也不暂停传送而一直传送到 DMA 传送结束为止。

在向 DMAC 的模块停止状态以及向全模块时钟停止模式、软件待机模式、深度软件待机模式转移时，必须将 DMST 位置“0” (DMAC 停止)。

有关向模块停止状态以及向全模块时钟停止模式、软件待机模式、深度软件待机模式的转移，请参照“8. 低功耗功能”。

12.2.15 DMA 仲裁状态寄存器 (DMASTS)

地址 0008 251Bh

| | | | | | | | | |
|-------|--------|--------|--------|--------|----|----|----|----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | DASTS0 | DASTS1 | DASTS2 | DASTS3 | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|--------|--------------|--|-----|
| b3-b0 | — | 保留位 | 读取值为“0”，写操作无效。 | R |
| b4 | DASTS3 | 通道 3 的仲裁状态标志 | 0: 不在进行数据传送 1: 正在进行数据传送 (正在进行操作数传送或者正在进行无停止传送) | R |
| b5 | DASTS2 | 通道 2 的仲裁状态标志 | | R |
| b6 | DASTS1 | 通道 1 的仲裁状态标志 | | R |
| b7 | DASTS0 | 通道 0 的仲裁状态标志 | | R |

DMASTS 寄存器是表示各通道数据传送状态的寄存器。

DASTSm 标志 (通道 m 的仲裁状态标志) (m=0 ~ 3)

如果开始通道 m 的数据传送 (1 个操作数传送或者无停止传送)，对应的 DASTSm 标志就变为“1”。如果数据传送结束，此标志就变为“0”。

[为“1”的条件]

- 在操作数传送方式的情况下，开始 1 个操作数的传送时
- 在无停止传送方式的情况下，开始 DMA 传送时

[为“0”的条件]

- 在 1 个操作数传送结束或者 DMA 传送结束时

12.2.16 DMA 传送结束的检测寄存器 (DMEDET)

地址 0008 2517h

| | | | | | | | |
|--------|--------|--------|--------|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| DEDET0 | DEDET1 | DEDET2 | DEDET3 | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|--------|---------------------|--|-----|
| b3-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b4 | DEDET3 | 通道 3 的 DMA 传送结束检测标志 | 读时 0: 未检测到 1: 检测到 | R/W |
| b5 | DEDET2 | 通道 2 的 DMA 传送结束检测标志 | | R/W |
| b6 | DEDET1 | 通道 1 的 DMA 传送结束检测标志 | | R/W |
| b7 | DEDET0 | 通道 0 的 DMA 传送结束检测标志 | | R/W |
| | | | 写时 0: 无效 1: 将 DEDETM 标志 (n=0 ~ 3) 置“0” | |

DMEDET 寄存器是表示各通道 DMA 传送结束的寄存器。

DEDETM 标志 (通道 m 的 DMA 传送结束检测标志) (m=0 ~ 3)

如果通道 m 的 DMA 传送结束, DEDETM 标志就变为“1”。此标志一旦变为“1”, 就不会自动变为“0”。要将 DEDETM 标志置“0”时, 必须通过程序写“1”。此时, 不保持所写的“1”。写“0”无效。

在使用 DMAm 中断 (DMTENDm) 时, 必须在中断处理程序内给发生中断请求的通道的 DEDETM 标志写“1”。

[为“1”的条件]

- DMA 传送结束时

[为“0”的条件]

- 给 DEDETM 位写“1”时

不能将 BSET 指令等位操作指令用于清除 DEDETM 位。要清除 DEDETM 位时, 必须使用 MOV 指令只将要清除的通道的位置“1”, 并且写 DMACm.DMEDET 寄存器。

12.3 运行说明

12.3.1 总线权的释放时序

在 1 个数据的读存取和写存取之间，DMAC 至少释放 1 个周期的总线权。在此期间，能进行其他总线主控 (CPU 和 DTC) 的存取。

即使在 DMAC 的存取过程中，只要不是 DMAC 的存取目标，也能进行 CPU 的存取。

详细内容请参照“11. 总线”。

DMAC 和其他总线主控的总线权例子如图 12.2 所示。

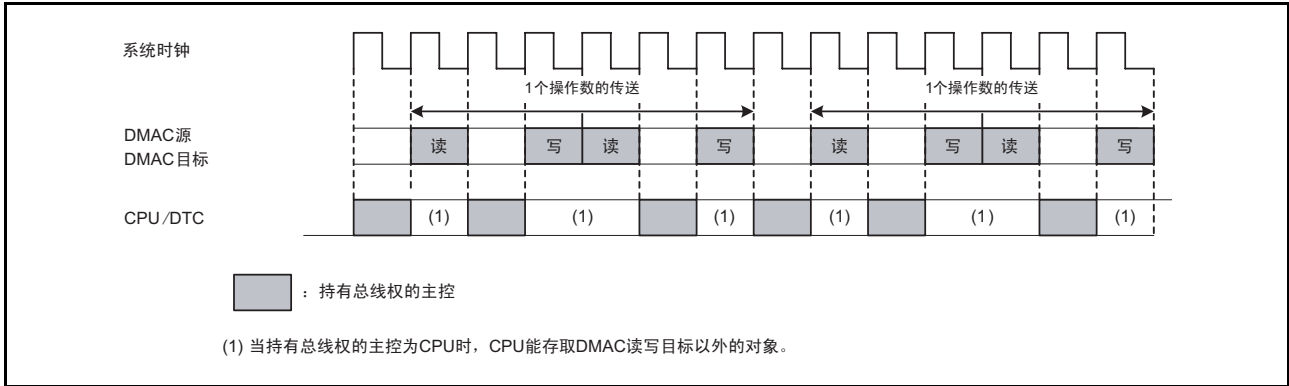


图 12.2 DMAC 和其他总线主控的总线权例子

12.3.2 传送方式

传送方式有操作数传送和无停止传送。

操作数传送方式有单个操作数传送和连续操作数传送。单个操作数传送是指 1 次 DMA 传送请求只传送 1 个操作数，连续操作数传送是指 1 次 DMA 传送请求逐个传送操作数，直到 DMA 传送结束为止。

无停止传送是指 1 次 DMA 传送请求连续传送数据，直到 DMA 传送结束为止。

与传送方式无关，1 次 DMA 传送是指传送 DMACm.DMCBC 寄存器设定的字节数的数据，如果 DMACm.DMCBC 寄存器变为“0000000h”，DMA 传送就结束。

传送方式如表 12.5 所示。

表 12.5 传送方式

| DMACm.DMCRA.DSEL[1:0] 位 | 传送方式 | 1 次 DMA 传送请求要传送的字节数 |
|-------------------------|---|------------------------|
| “00b” (单个操作数传送) | <ul style="list-style-type: none"> 在开始 DMA 传送时传送 1 个操作数的数据。 在 DMA 传送结束前，每产生 1 次 DMA 传送请求，就传送 1 个操作数。 在 1 个操作数传送结束时进行通道仲裁。 每当 1 个操作数传送结束时，需要 DMA 传送请求。 | 1 个操作数的数据个数 × 对应位长的字节数 |
| “01b” (连续操作数传送) | <ul style="list-style-type: none"> 在开始 DMA 传送时传送 1 个操作数的数据。 在 DMA 传送结束前，逐个传送操作数。 每当 1 个操作数传送结束时进行通道仲裁。 只在最初需要 DMA 传送请求。 | DMACm.DMCBC 寄存器设定的字节数 |
| “11b” (无停止传送) | <ul style="list-style-type: none"> 在开始 DMA 传送时连续传送数据。 在 DMA 传送结束前继续传送。 在 DMA 传送结束前不进行通道仲裁。 只在最初需要 DMA 传送请求。 | DMACm.DMCBC 寄存器设定的字节数 |

在操作数传送方式的情况下，如果在 1 个操作数传送结束时的通道仲裁中有优先级高的通道的 DMA 传送请求，就接受此传送请求。如果没有，就继续传送下一个操作数。但是，在单个操作数传送的情况下，如果没有 DMA 传送请求，就不传送下一个操作数。

在无停止传送方式的情况下，因为从 DMA 传送开始到 DMA 传送结束连续传送数据，所以，即使在 DMA 传送中有优先级高的通道的 DMA 传送请求，也不接受。

不同传送方式的传送例子如图 12.3 所示。

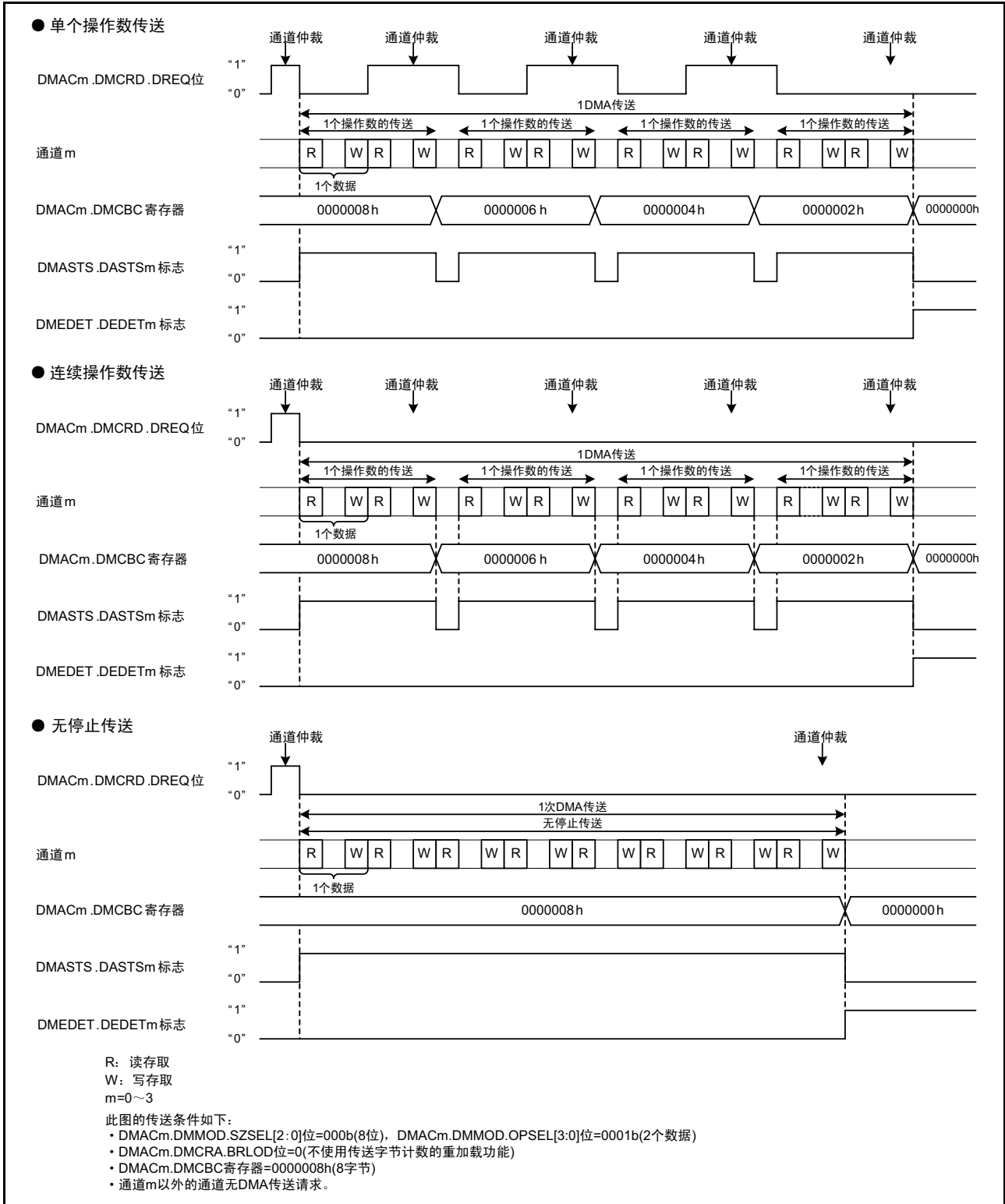


图 12.3 不同传送方式的传送例子

12.3.3 DMAC 的启动

寄存器的设定步骤如图 12.4 所示。



图 12.4 寄存器的设定步骤

12.3.4 DMA 传送的开始

如果在将 DMACm.DMCRE.DEN 位置 “1” (允许 DMA 传送) 后将 DMSCNT.DMST 位置 “1” (DMAC 运行), 就能进行通道 m (m=0 ~ 3) 的 DMA 传送。

如果产生 DMA 传送请求, 就进行通道仲裁, 并且在接受优先级高的通道的 DMA 传送请求后开始 DMA 传送。一旦接收 DMA 传送请求并且开始 DMA 传送, DMASTS.DASTSm 标志就变为 “1” (正在进行数据传送)。

12.3.5 DMA 传送的结束

如果 DMACm.DMCBC 寄存器变为 “0000000h”, 通道 m (m=0 ~ 3) 的 DMA 传送就结束并且进行以下的处理:

- DMACm.DMEDET.DEDETm 标志变为 “1” (检测到 DMA 传送结束)。
- 当 DMICNT.DINTMm 位为 “1” (允许中断) 时, 产生 DMAm 中断请求 (DMTENDm)。
- 当 DMACm.DMCRC.ECLR 位为 “1” (在 DMA 传送结束时, 将 DEN 位置 “0”) 时, DMACm.DMCRE.DEN 位变为 “0” (禁止 DMA 传送), 以后不进行通道 m 的 DMA 传送。
- 在使用重加载功能的情况下, 将重加载寄存器的值重新加载到当前寄存器。

12.3.6 DMA 传送的暂停、重新开始和中止

(1) DMA 传送的暂停

如果在操作数传送方式的 DMA 传送过程中, 将 DMSCNT.DMST 位置 “0” (DMAC 停止) 或者将 DMACm.DMCRE.DEN 位置 “0” (禁止 DMA 传送), 就能暂停 DMA 传送。

如果将 DMST 位置 “0”, 全部通道的 DMA 传送就为暂停对象; 如果将 DEN 位置 “0”, 对应通道的 DMA 传送就为暂停对象, 在结束正在执行的 1 个操作数传送后暂停。

如果在无停止传送模式的 DMA 传送过程中, 即使 DMST 位或者 DEN 位变为 “0”, 也不暂停 DMA 传送而一直传送到 DMA 传送结束为止。

在向 DMAC 的模块停止状态以及向全模块时钟停止模式、软件待机模式、深度软件待机模式转移时, 必须将 DMST 位置 “0” (DMAC 停止)。

(2) DMA 传送的重新开始

通过将 DMSCNT.DMST 位或者 DMACm.DMCRE.DEN 位置 “1” (允许 DMA 传送), 重新开始已暂停的通道的 DMA 传送。

如果在从 DMAC 的模块停止状态、全模块时钟停止模式或者软件待机模式返回后, 将 DMST 位置 “1” (DMAC 运行), 就重新开始已暂停的通道的 DMA 传送。

(3) DMA 传送的中止

如果在暂停各通道的状态下给 DMACm.DMCRB.DSCLR 位写 “1”, 就对 DMAC 的内部状态进行初始化并且中止 DMA 传送。但是, 此时只对 DMAC 内部电路的传送状态而不对各寄存器进行初始化。

12.3.7 DMA 启动源

能选择软件触发、中断控制器 (ICU) 设定的外部引脚中断或者外围功能中断作为 DMA 启动源。

12.3.7.1 软件触发

在选择了软件触发作为 DMA 启动源的情况下，如果通过程序给 DMACm.DMCRD.DREQ 位写“1”（有 DMA 传送请求），就产生 DMA 传送请求。与 DMA 传送状态无关，能将 DREQ 位置“1”。如果要将此位置“0”（无 DMA 传送请求），就必须在 DMAC 停止或者禁止 DMA 传送而不能在数据传送过程中进行写操作。

12.3.7.2 外部引脚中断和外围功能中断

在通过中断控制器 (ICU) 的中断请求允许寄存器 (ICU.IERi (i=02h ~ 1Fh)) 允许外部引脚中断或者外围功能中断，在通过中断请求目标设定位 (ICU.ISELRI.ISEL[1:0] (i= 中断向量号)) 将对应的中断请求目标设定为 DMAC 并且通过 DMACm.DMCRA.DCTG[5:0] 位选择 DMA 启动源的情况下，如果发生所选的中断请求，就产生 DMA 传送请求。

如果检测到 DMA 传送请求，DMACm.DMCRD.DREQ 位就变为“1”（有 DMA 传送请求），即使此后输入电平发生变化，DREQ 位也不变，将“1”保持到通过程序写“0”（无 DMA 传送请求）或者接受 DMA 传送请求为止。

即使在 DREQ 位为“1”时产生新的传送请求，也忽视该传送请求。

DMACm.DMCRD.DREQ 位的时序如图 12.5 所示。

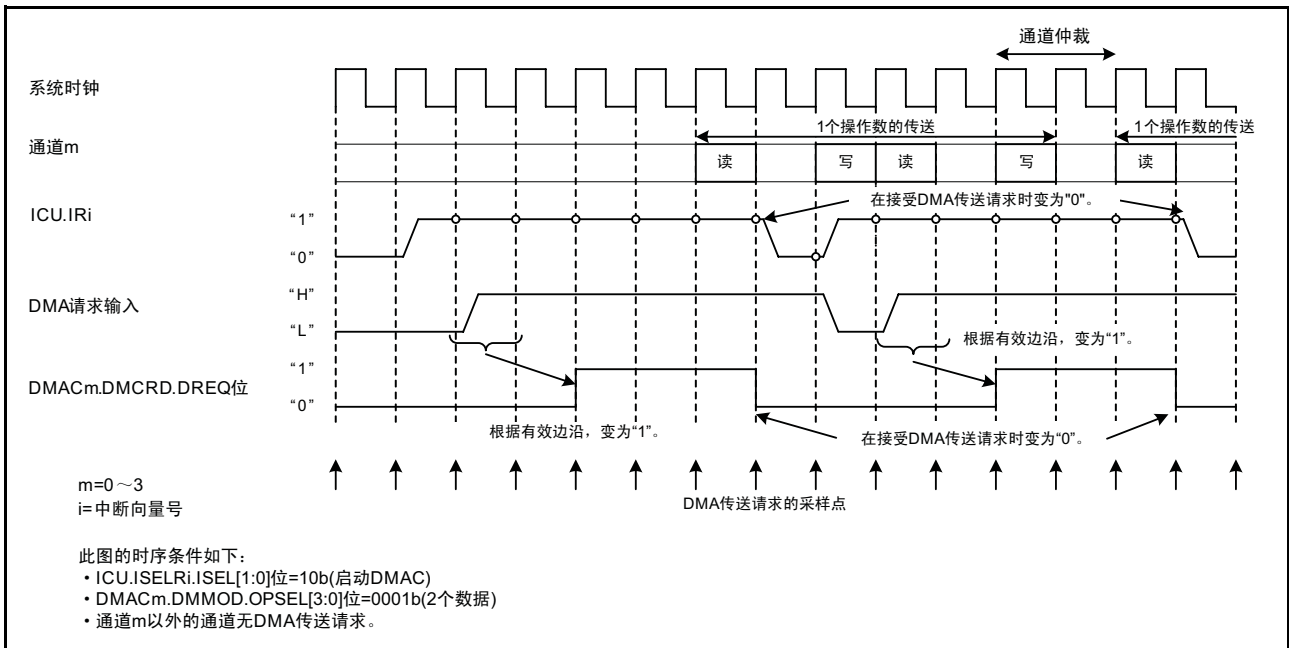


图 12.5 DMACm.DMCRD.DREQ 位的时序

12.3.8 通道仲裁

当有多个 DMA 传送请求时，DMAC 判断发生请求的通道的优先级。通道的优先级固定为通道 0 > 通道 1 > 通道 2 > 通道 3。

如果在数据传送过程中发生 DMA 传送请求，就在开始最后数据的写存取时开始通道仲裁。如果在数据传送过程中发生优先级高的通道的 DMA 传送请求，就在数据传送结束后开始该通道的传送。

12.3.9 重加载功能

重加载功能是指在 DMA 传送结束时将重加载寄存器 (DMACm.DMRSA、DMACm.DMRDA、DMACm.DMRBC) 的值重新加载到当前寄存器 (DMACm.DMCSA、DMACm.DMCDA、DMACm.DMCBC) 的功能, 能用于重新加载传送源地址、传送目标地址和传送字节计数。因为使用重加载功能能对分离的区域进行连续传送, 所以在同一个通道能将不同传送区域和字节数的多个传送块进行连续传送。通过在传送结束前给重加载寄存器写值, 能在不影响 DMA 传送过程中的当前寄存器的情况下准备下一个传送。

在使用重加载功能时, 必须给重加载寄存器和当前寄存器设定数据。必须在开始传送最后的数据 (作为 DMA 传送结束) 前设定重加载寄存器, 否则就有可能在 DMA 传送结束后来不及重新加载。

在不使用重加载功能时, 必须将 DMACm.DMCRC.ECLR 位置 “1” (在 DMA 传送结束时, 将 DEN 位置 “0”), 使 DMACm.DMCRE.DEN 位变为 “0”。

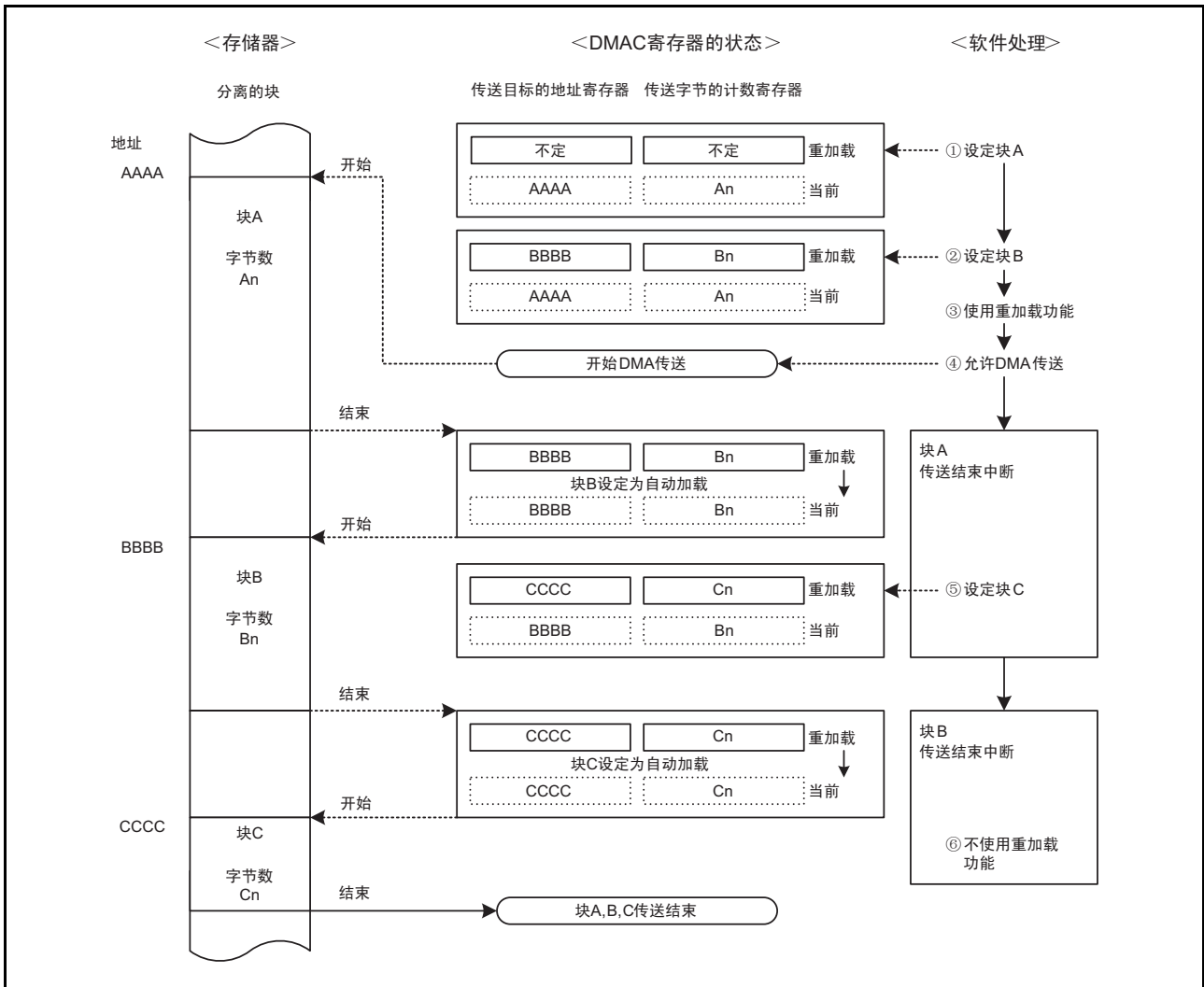


图 12.6 使用重加载功能时的传送例子

12.3.10 循环

在通过 DMACm.DMMOD.DMOD[2:0] 位或者 DMACm.DMMOD.SMOD[2:0] 位选择了循环的情况下，如果在数据传送过程中地址正方向增加并且 1 个操作数传送结束，就将开始 DMA 传送时的设定值恢复到地址寄存器。

循环的传送例子如图 12.7 所示。

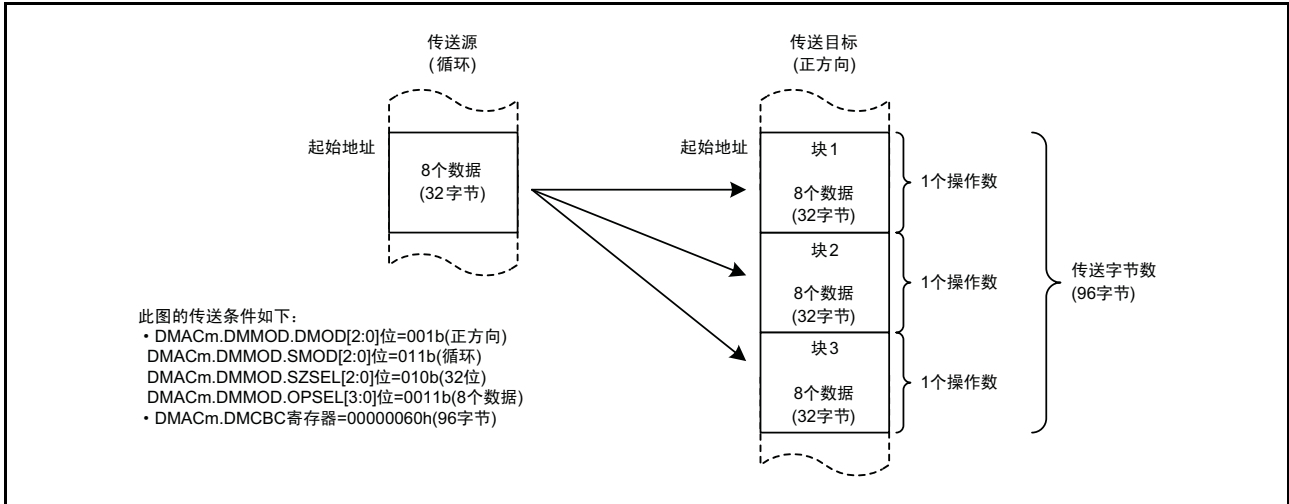


图 12.7 循环的传送例子

12.4 中断

如果在 DMICNT.DINTMm 位 (n=0~3) 为“1” (允许中断) 时通道 m 的 DMA 传送结束，就产生 DMAm 中断请求 (DMTENDm)。中断输出的概略逻辑图如图 12.8 所示。

在使用 DMAm 中断时，必须在中断处理程序内给发生中断请求的通道的 DMEDET.DEDETm 标志写“1”。如果在 DMEDET.DEDETm 标志为“1”时将 DMICNT.DINTMm 位从“0”变为“1”，就产生 DMAm 中断请求 (DMTENDm)，与 DMAC 的传送状态无关。

能通过中断控制器 (ICU) 的设定将 DMAm 中断设定为数据传送控制器 (DTC) 的启动源，详细内容请参照“10. 中断控制器 (ICU)”。

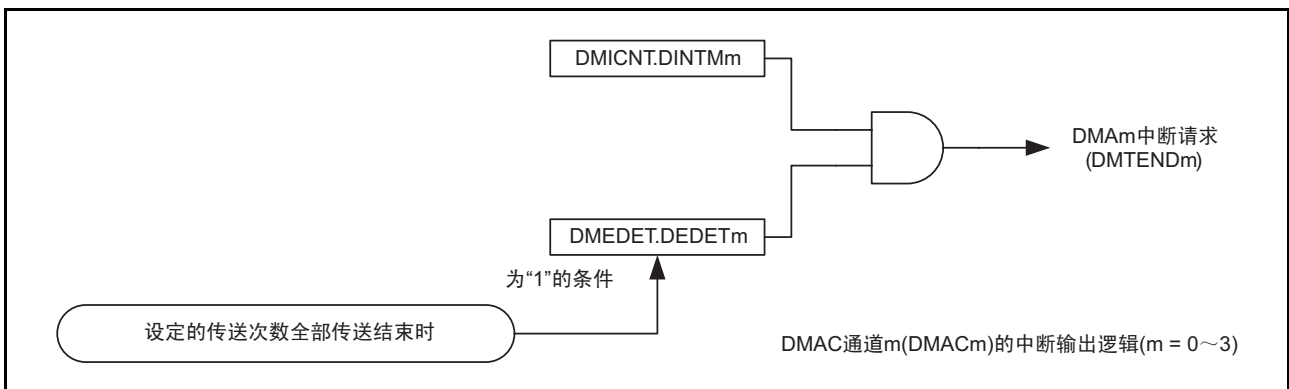


图 12.8 中断输出的概略逻辑图

12.5 低功耗功能

如果 DMAC 接受向模块停止状态以及向全模块时钟停止模式、软件待机模式、深度软件待机模式转移的请求，就暂停正在进行的 DMA 传送，向低功耗的各模式转移。

在向 DMAC 模块停止状态以及向全模块时钟停止模式、软件待机模式、深度软件待机模式转移时，必须将 DMSCNT.DMST 位置“0” (DMAC 停止)。

(1) 模块停止的情况

通过在给 DMSCNT.DMST 位写“0” (DMAC 停止) 后给 MSTPCRA.MSTPA28 位写“1” (转移到 DMAC 模块停止状态)，DMAC 的模块停止功能有效。如果在给 MSTPA28 位写“1”时正在进行 DMA 传送，就暂停 DMA 传送并且转移到模块停止状态。当 MSTPA28 位为“1”时，无论是否正在传送数据，都不能存取 DMAC 的内部寄存器。

通过给 MSTPA28 位写“0” (解除模块停止状态) 来解除 DMAC 的模块停止。如果给 DMAC 提供时钟，就能存取 DMAC 的内部寄存器。

(2) 全模块时钟停止模式的情况

在给 DMSCNT.DMST 位写“0” (DMAC 停止) 后给 MSTPCRA.ACSE 位写“1” (允许全模块时钟停止模式)，在给包括 MSTPCRA.MSTPA28 位 (转移到 DMAC 模块停止状态) 在内的 MSTPCRA 寄存器和 MSTPCRB 寄存器的全部位写“1”并且确认 MSTPCRA 寄存器和 MSTPCRB 寄存器的全部位已变为“1”后，如果执行 WAIT 指令，就向全模块时钟停止模式转移。如果在执行 WAIT 指令时正在进行 DMA 传送，就能在暂停 DMA 传送后转移到全模块时钟停止状态。

通过外部中断 (NMI、IRQ0 ~ IRQ15)、RES# 引脚的复位或者内部中断 (8 位定时器和看门狗定时器) 来解除全模块时钟停止模式。

(3) 软件待机模式和深度软件待机模式的情况

在给 DMSCNT.DMST 位写“0” (DMAC 停止) 后给 SBYCR.SSBY 位写“1” (在执行 WAIT 指令后转移到软件待机模式) 并且给 DPSBYCR.DPSBY 位写“0” (在执行 WAIT 指令后转移到软件待机模式)，然后通过执行 WAIT 指令转移到软件待机模式。

如果在执行 WAIT 指令时正在进行 DMA 传送，就在暂停 DMA 传送后转移到软件待机模式。

通过外部中断 (NMI、IRQ0 ~ IRQ15) 或者 RES# 引脚的复位来解除软件待机模式。

如果将 DPSBYCR.DPSBY 位置“1” (在执行 WAIT 指令后转移到深度软件待机模式)，就转移到深度软件待机模式。

12.6 使用时的注意事项

12.6.1 寄存器的设定

1. 必须在设定的通道的DMASTS.DASTSm标志 (m=0~3) 为“0” (不在进行数据传送) 并且DMACm.DMCRE.DEN位为“0” (禁止DMA传送) 或者DMSCNT.DMST位为“0” (DMAC停止) 的情况下, 设定以下的寄存器或者位:
DMACm.DMMOD、DMACm.DMCRB、DMACm.DMCRC、DMACm.DMCSA、DMACm.DMCDA、DMACm.DMCBC 寄存器
DMACm.DMCRA.DSEL[1:0]位和DMACm.DMCRA.DCTG[5:0]位
给DMACm.DMCRD.DREQ位写“0” (无论是否正在传送数据, 都能给DREQ位写“1”)。
2. 必须以32位存取以下的寄存器:
DMACm.DMMOD、DMACm.DMCSA、DMACm.DMCDA、DMACm.DMCBC、DMACm.DMRSA、DMACm.DMRDA 寄存器
3. 必须在DMASTS.DASTSm标志 (n=0~3) 为“0” (不在进行数据传送) 时写DMACm.DMCRC.ECLR位。在不使用重加载功能时, 必须通过将ECLR位置“1” (在DMA传送结束时, 将DEN位置“0”), 使DMACm.DMCRE.DEN位为“0”。
4. 如果设定了DMACm.DMCRA.DCTG[5:0]位, 就必须在将设定的通道的DMACm.DMCRD.DREQ位置“0”后, 将DMSCNT.DMST位置“1” (DMAC运行) 并且将DMACm.DMCRE.DEN位置“1” (允许DMA传送)。
5. DMACm.DMCRD.DREQ位与DMSCNT.DMST位和DMACm.DMCRE.DEN位的设定无关, 因DMA传送请求的有无而发生变化。如果DMA启动源不是软件触发, 就不能通过程序给DREQ位写“1” (有DMA传送请求)。
6. 必须给各寄存器设定地址和传送字节数 (根据位长进行对齐后的值)。根据位长进行的对齐和寄存器低2位的设定值如表12.6所示。
7. 不能将BSET指令等位操作指令用于清除DMEDET.DEDETM标志。要清除DEDETM标志时, 只能使用MOV指令给要清除的通道的标志写“1”。

表 12.6 根据位长进行的对齐和寄存器低 2 位的设定值

| DMACm.DMMOD.SZSEL[2:0] 位 | 对齐 | 地址寄存器 | | 字节计数寄存器 | |
|--------------------------|-------|-------|----|---------|----|
| | | b1 | b0 | b1 | b0 |
| “000b” (8 位) | 整数倍 | x | x | x | x |
| “001b” (16 位) | 2 的倍数 | x | 0 | x | 0 |
| “010b” (32 位) | 4 的倍数 | 0 | 0 | 0 | 0 |

x: “0” 或者 “1” 都可以。

12.6.2 使用外部设备的情况

在进行外部设备的 DMA 传送过程中, 在从开始写最后的数据到外部总线存取结束前, DMASTS.DASTSm 标志 (m=0~3) 有可能变为“0” (不在进行数据传送)。

13. 数据传送控制器 (DTC)

RX610 群内置数据传送控制器 (DTC)。
能通过中断请求启动 DTC 并且进行数据传送。

13.1 概要

DTC 的规格和框图分别如表 13.1 和图 13.1 所示。

表 13.1 DTC 的规格

| 项目 | 内容 |
|---------------|---|
| 传送模式 | <ul style="list-style-type: none"> • 正常传送模式 • 重复传送模式 • 块传送模式 |
| 任意个通道的传送 | <ul style="list-style-type: none"> • 对于 1 个启动源，能进行多个通道的数据传送（链传送）。 • 能设定为在数据传送后进行链传送。 |
| 短地址模式 / 全地址模式 | <ul style="list-style-type: none"> • 在短地址模式中，传送信息分配 3 个长字；在全地址模式中，传送信息分配 4 个长字。 • 在短地址模式中，能用 24 位指定传送源地址和传送目标地址，并且能直接指定 16M 字节的地址空间。 • 在全地址模式中，能用 32 位指定传送源地址和传送目标地址，并且能直接指定 4G 字节的地址空间。 |
| 数据传送单位 | <ul style="list-style-type: none"> • 字节、字、长字 有关指定奇数地址的字传送 / 长字传送或者指定地址 $4n+2$ 的长字传送的可执行区域，请参照“11. 总线”。 |
| 中断源 | <ul style="list-style-type: none"> • 在 1 次数据传送结束后向 CPU 请求中断。 • 在指定数据个数的数据传送结束后向 CPU 请求中断。 |

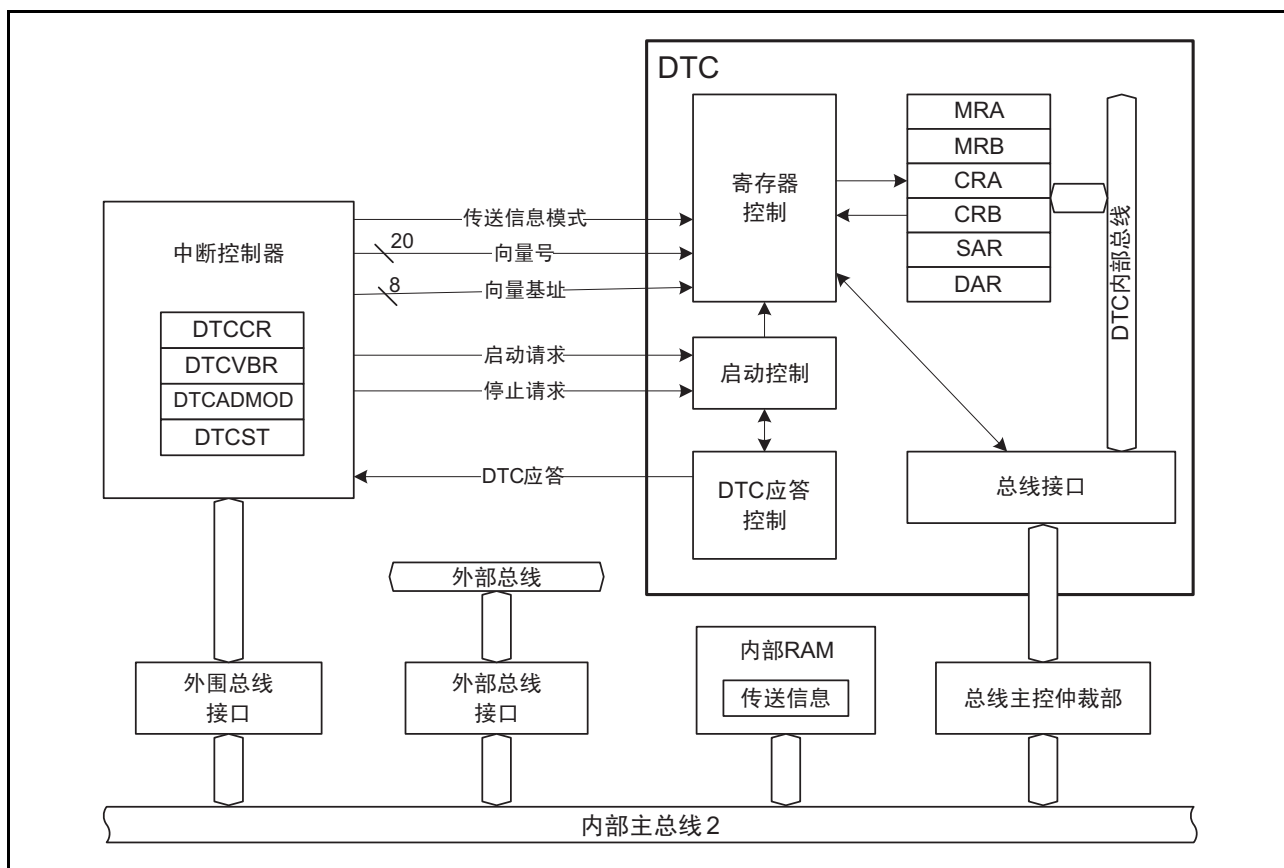


图 13.1 DTC 的框图

13.2 寄存器说明

DTC 的寄存器一览表如表 13.2 所示。

MRA、MRB、SAR、DAR、CRA、CRB 寄存器不能从 CPU 直接存取，而是作为传送信息被分配到数据区。

如果产生 DTC 启动请求，就根据各启动源决定的向量地址读取传送信息的起始地址，在将任意的传送信息传送到 DTC 后进行数据传送。一旦传送结束，这些寄存器的内容就被回写。

表 13.2 DTC 的寄存器一览表

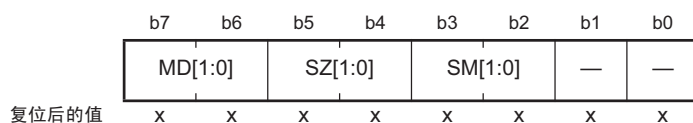
| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|---------------|----------|-----------|------------|------|
| DTC 模式寄存器 A | MRA | xxh | — | 8 位 |
| DTC 模式寄存器 B | MRB | xxh | — | 8 位 |
| DTC 源地址寄存器 | SAR | xxxxxxxxh | — | 32 位 |
| DTC 目标地址寄存器 | DAR | xxxxxxxxh | — | 32 位 |
| DTC 传送计数寄存器 A | CRA | xxxxh | — | 16 位 |
| DTC 传送计数寄存器 B | CRB | xxxxh | — | 16 位 |
| DTC 控制寄存器 | DTCCR | 00h | 0008 7400h | 8 位 |
| DTC 向量基址寄存器 | DTCVBR | 00000000h | 0008 7404h | 32 位 |
| DTC 地址模式寄存器 | DTCADM0D | 00h | 0008 7408h | 8 位 |
| DTC 模块启动寄存器 | DTCST | 00h | 0008 740Ch | 8 位 |

x: 不定值

注 1. 要启动 DTC 时，需要设定中断控制器 (ICU) 的 ISEL_{Ri}.ISEL[1:0] 位和 IER_i.IEN_j 位，详细内容请参照“10. 中断控制器 (ICU)”。

13.2.1 DTC 模式寄存器 A (MRA)

地址 (CPU不能直接存取)



x: 不定值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|-----------------|---|-----|
| b1-b0 | — | 保留位 | 读取值为不定值，只能写“0”。 | — |
| b3-b2 | SM[1:0] | SAR 传送源地址的寻址方式位 | b3 b2 0 0: SAR 寄存器为地址固定 (省略 SAR 寄存器的回写) 0 1: SAR 寄存器为地址固定 (省略 SAR 寄存器的回写) 1 0: 传送后 SAR 寄存器递增 (当 SZ[1:0] 位为“00b”时加 1；为 “01b”时加 2；为“10b”时加 4) 1 1: 传送后 SAR 寄存器递减 (当 SZ[1:0] 位为“00b”时减 1；为 “01b”时减 2；为“10b”时减 4) | — |
| b5-b4 | SZ[1:0] | DTC 数据传送长度位 | b5 b4 0 0: 字节传送 0 1: 字传送 1 0: 长字传送 1 1: 不能设定 | — |
| b7-b6 | MD[1:0] | DTC 模式位 | b7 b6 0 0: 正常传送模式 0 1: 重复传送模式 1 0: 块传送模式 1 1: 不能设定 | — |

MRA 寄存器是选择 DTC 运行模式的寄存器。

CPU 不能直接存取 MRA 寄存器。

SM[1:0] 位 (SAR 传送源地址的寻址方式位)

这些位指定数据传送后的 SAR 寄存器的运行。

SZ[1:0] 位 (DTC 数据传送长度位)

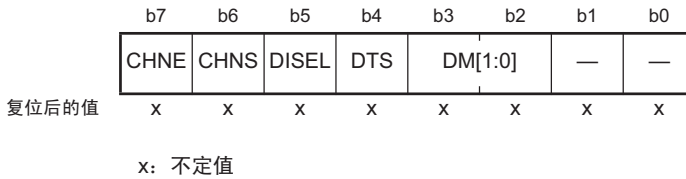
这些位指定传送数据的长度。

MD[1:0] 位 (DTC 模式位)

这些位指定 DTC 传送模式。

13.2.2 DTC 模式寄存器 B (MRB)

地址 (CPU不能直接存取)



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|------------------|---|-----|
| b1-b0 | — | 保留位 | 读取值为不定值，只能写“0”。 | — |
| b3-b2 | DM[1:0] | DAR 传送目标地址的寻址方式位 | b3 b2 00: DAR 寄存器为地址固定 (省略 DAR 寄存器的回写) 01: DAR 寄存器为地址固定 (省略 DAR 寄存器的回写) 10: 传送后 DAR 寄存器递增 (当 MAR.SZ[1:0] 位为“00b”时加 1; 为 “01b”时加 2; 为“10b”时加 4) 11: 传送后 DAR 寄存器递减 (当 MAR.SZ[1:0] 位为“00b”时减 1; 为 “01b”时减 2; 为“10b”时减 4) | — |
| b4 | DTS | DTC 传送模式选择位 | 0: 传送目标为重复区域或者块区域 1: 传送源为重复区域或者块区域 | — |
| b5 | DISEL | DTC 中断选择位 | 0: 在指定的数据传送结束时向 CPU 请求中断 1: 在每次 DTC 数据传送时都向 CPU 请求中断 | — |
| b6 | CHNS | DTC 链传送选择位 | 0: 连续进行链传送 1: 只在传送计数器为“0”时进行链传送 | — |
| b7 | CHNE | DTC 链传送允许位 | 0: 禁止链传送 1: 允许链传送 | — |

MRB 寄存器是选择 DTC 运行模式的寄存器。
CPU 不能直接存取 MRB 寄存器。

DM[1:0] 位 (DAR 传送目标地址的寻址方式位)
这些位指定数据传送后的 DAR 寄存器的运行。

DTS 位 (DTC 传送模式选择位)
在重复传送模式或者块传送模式中，此位指定是将传送源还是将传送目标作为重复区域或者块区域。

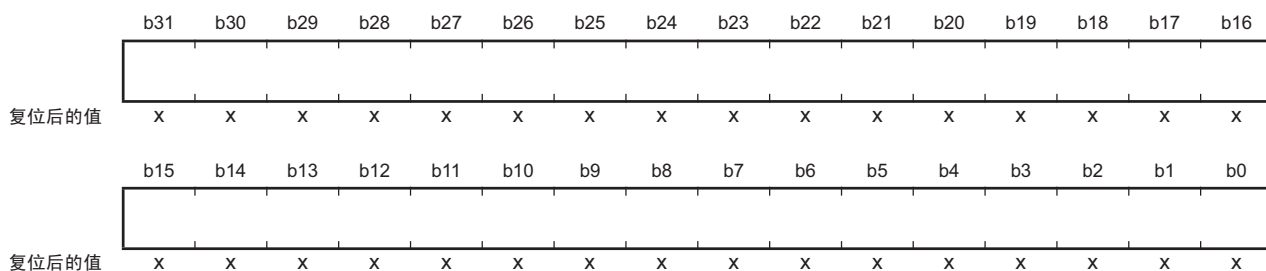
DISEL 位 (DTC 中断选择位)
此位指定是在每次 DTC 数据传送时还是只在数据传送结束时向 CPU 请求中断。

CHNS 位 (DTC 链传送选择位)
此位选择链传送的条件。
如果下一次传送是链传送，就不判断指定次数的传送结束，并且不清除启动源标志，也不向 CPU 请求中断。

CHNE 位 (DTC 链传送允许位)
此位指定链传送。
通过 CHNS 位选择链传送的条件。
有关链传送的详细内容，请参照“13.4.6 链传送”。

13.2.3 DTC 源地址寄存器 (SAR)

地址 (CPU不能直接存取)



x: 不定值

SAR 寄存器是设定传送源起始地址的寄存器。

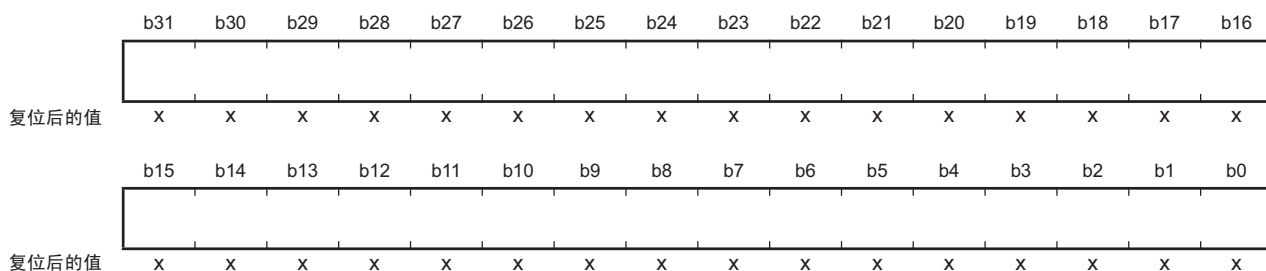
在全地址模式中 32 位有效。

在短地址模式中低 24 位有效，忽视高 8 位 (b31 ~ b24) 的设定，通过 b23 指定的值进行位扩展。

CPU 不能直接存取 SAR 寄存器。

13.2.4 DTC 目标地址寄存器 (DAR)

地址 (CPU不能直接存取)



x: 不定值

DAR 寄存器是设定传送目标起始地址的寄存器。

在全地址模式中 32 位有效。

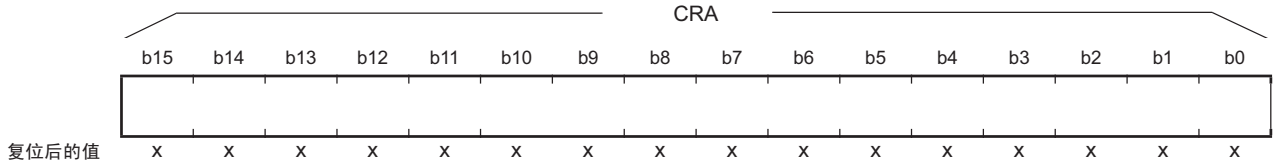
在短地址模式中低 24 位有效，忽视高 8 位 (b31 ~ b24) 的设定，通过 b23 指定的值进行位扩展。

CPU 不能直接存取 DAR 寄存器。

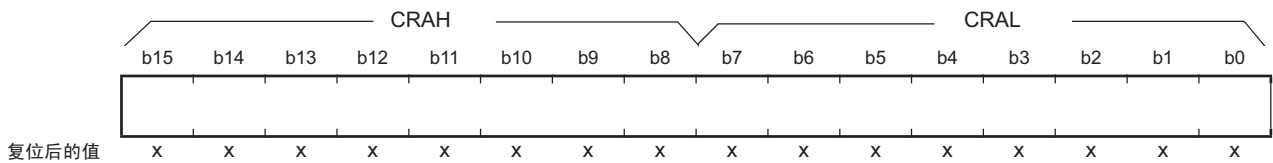
13.2.5 DTC 传送计数寄存器 A (CRA)

地址 (CPU不能直接存取)

• 正常传送模式



• 重复传送模式和块传送模式



x: 不定值

注1. 功能因传送模式而不同。

| 符号 | 寄存器名 | 功能 | R/W |
|------|----------------|---------|-----|
| CRAL | 传送计数器 A 的低位寄存器 | 设定传送次数。 | — |
| CRAH | 传送计数器 A 的高位寄存器 | | — |

注 1. 在重复传送模式和块传送模式中，必须给 CRAH 寄存器和 CRAL 寄存器设定相同的值。

CRA 寄存器是指定 DTC 传送次数的寄存器，其功能因传送模式而不同。

CPU 不能直接存取 CRA 寄存器。

(1) 正常传送模式 (MRA.MD[1:0] 位 =00b)

在正常传送模式中，CRA 寄存器用作 16 位传送计数器。

当设定值为“0001h”时，传送次数为 1 次；当设定值为“FFFFh”时，传送次数为 65535 次；当设定值为“0000h”时，传送次数为 65536 次。

每进行 1 次数据传送，CRA 寄存器就减 1。

(2) 重复传送模式 (MRA.MD[1:0] 位 =01b)

CRAH 寄存器保持传送次数，CRAL 寄存器用作 8 位传送计数器。

当设定值为“01h”时，传送次数为 1 次；当设定值为“FFh”时，传送次数为 255 次；当设定值为“00h”时，传送次数为 256 次。

每进行 1 次数据传送，CRAL 寄存器就减 1。当 CRAL 寄存器变为“00h”时，就传送 CRAH 寄存器的值。

(3) 块传送模式 (MRA.MD[1:0] 位 =10b)

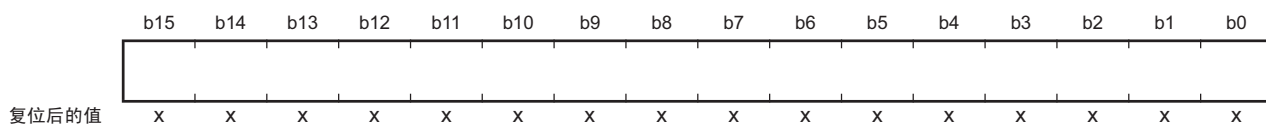
CRAH 寄存器保持块的大小，CRAL 寄存器用作 8 位块大小计数器。

当设定值为“01h”时，传送次数为 1 次；当设定值为“FFh”时，传送次数为 255 次；当设定值为“00h”时，传送次数为 256 次。

每进行 1 次数据传送，CRAL 寄存器就减 1。当 CRAL 寄存器变为“00h”时，就传送 CRAH 寄存器的值。

13.2.6 DTC 传送计数寄存器 B (CRB)

地址 (CPU不能直接存取)



x: 不定值

CRB 寄存器是指定块传送模式的块传送次数的寄存器。

当设定值为“0001h”时，传送次数为 1 次；当设定值为“FFFFh”时，传送次数为 65535 次；当设定值为“0000h”时，传送次数为 65536 次。每进行 1 次数据传送，CRB 寄存器就减 1。

在正常传送模式和重复传送模式中，不使用 CRB 寄存器并且忽视设定值。

CPU 不能直接存取 CRB 寄存器。

13.2.7 DTC 控制寄存器 (DTCCR)

地址 0008 7400h

| | | | | | | | |
|-------|----|----|-----|-------|----|----|-----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | RRS | RCHNE | — | — | ERR |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|------------------|------------------------------------|-----|
| b0 | ERR (注1) | 传送停止标志 | 0: 无 DTC 传送停止请求 1: 有 DTC 传送停止请求 | R |
| b2-b1 | — | 保留位 | 读写值都为“0”。 | R/W |
| b3 | RCHNE | DTC 重复传送后的链传送允许位 | 0: 禁止重复传送后的链传送 1: 允许重复传送后的链传送 | R/W |
| b4 | RRS | DTC 传送信息的跳读允许位 | 0: 不跳读传送信息 1: 在向量号相同时跳读传送信息 | R/W |
| b7-b5 | — | 保留位 | 读写值都为“0”。 | R/W |

注 1. 当 ERR 位为“1”时，不启动 DTC。要启动 DTC 时，必须进行以下的处理：

当传送停止源是非屏蔽中断时，必须清除该中断源；当传送停止源不是非屏蔽中断时，因为是由总线错误引起的中断，所以必须通过总线错误监视部的总线错误源清除寄存器 (BERCLR) 来清除该中断源。非屏蔽中断的详细内容请参照“10. 中断控制器 (ICU)”，总线错误的详细内容请参照“11. 总线”。

DTCCR 寄存器是控制 DTC 的寄存器。

ERR 标志 (传送停止标志)

此标志表示因总线错误或者非屏蔽中断而产生 DTC 传送停止请求。

DTC 一旦接受传送停止请求就停止传送。

[为“1”的条件]

- 在发生非屏蔽中断或者总线错误中断时

[为“0”的条件]

- 在非屏蔽中断处理结束并且清除了中断源，或者总线错误中断处理结束并且清除了总线错误源时

RCHNE 位 (DTC 重复传送后的链传送允许位)

在重复传送模式中，此位设定允许或者禁止在传送计数器为“0” (指定次数的传送结束) 时的链传送。

在重复传送模式中，如果传送计数器的 CRAL 寄存器变为“00h”，就将 CRAH 寄存器的设定值回写到 CRAL 寄存器，因此在传送计数器为“0”时不进行链传送。通过将 RCHNE 位置“1”，允许在回写传送计数器时的链传送。

RRS 位 (DTC 传送信息的跳读允许位)

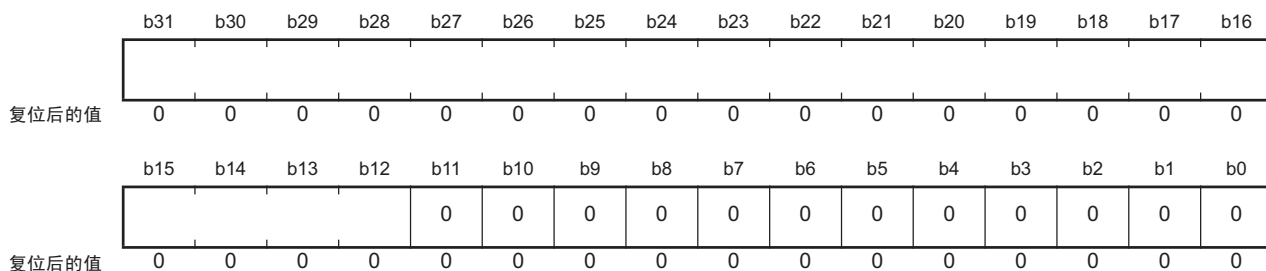
DTC 向量号随时和上次启动的向量号进行比较。

如果向量号相同并且 RRS 位为“1”，就不读传送信息而进行 DTC 的数据传送。如果上次的启动为链传送，就读传送信息，与 RRS 位的值无关。

在上次的传送为正常传送并且传送计数器 (CRA 寄存器) 为“0”时，或者在上次的传送为块传送并且传送计数器 (CRB 寄存器) 为“0”时，都读传送信息，与 RRS 位的值无关。

13.2.8 DTC 向量基址寄存器 (DTCVBR)

地址 0008 7404h



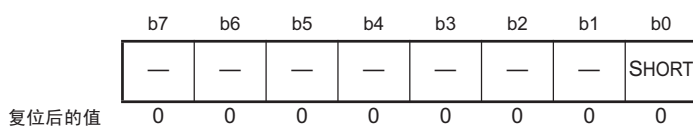
DTCVBR 寄存器设定在计算 DTC 向量表地址时的基址。

低 12 位 (b11-b0) 固定为“0”，写操作无效。

忽视高 4 位 (b31-b28) 的设定，通过 b27 指定的值进行位扩展。

13.2.9 DTC 地址模式寄存器 (DTCADM0D)

地址 0008 7408h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|----------|----------------------|-----|
| b0 | SHORT | 短地址模式设定位 | 0: 全地址模式 1: 短地址模式 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

DTCADM0D 寄存器是设定 DTC 能存取的地域的寄存器。

SHORT 位 (短地址模式设定位)

在全地址模式中，能存取 4G 字节空间 (00000000h ~ FFFFFFFFh)。

在短地址模式中，能存取 16M 字节空间 (00000000h ~ 007FFFFFFFh 和 FF800000h ~ FFFFFFFFh)。

13.2.10 DTC 模块启动寄存器 (DTCST)

地址 0008 740Ch

| | | | | | | | | |
|-------|----|----|----|----|----|----|----|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | — | — | — | — | — | DTCST |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-----------|----------------------------|-----|
| b0 | DTCST | DTC 模块启动位 | 0: DTC 模块停止 1: DTC 模块运行 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

DTCST 寄存器是设定 DTC 模块的运行或者停止的寄存器。

DTCST 位 (DTC 模块启动位)

为了使 DTC 能接受传送请求, 必须将 DTCST 位置“1”。如果将 DTCST 位置“0”, 就不能接受新的传送请求。

如果在运行过程中将此位改写为“0”, 已接受的传送请求就保持到处理结束为止。

在向模块停止状态以及向全模块时钟停止模式、软件待机模式、深度软件待机模式转移时, 必须将 DTCST 位置“0”。

有关向模块停止状态以及向全模块时钟停止模式、软件待机模式、深度软件待机模式的转移, 请参照“13.8.1 DTC 模块启动寄存器的设定”和“8. 低功耗功能”。

13.3 启动源

通过中断请求启动 DTC。如果将启动 DTC 的中断对应的中断控制器 (ICU) 的 ISEL_{Ri}.ISEL[1:0] 位 (i 为中断向量号) 置 “01b”，该中断就为 DTC 的启动源；如果将此位置 “00b”，该中断就为 CPU 的中断源。

在 1 次数据传送 (在链传送时，为连续传送的最后) 结束时，将作为启动源的中断标志或者对应的 ISEL_{Ri}.ISEL[1:0] 位清 “00b”。

13.3.1 传送信息的分配和 DTC 向量表

传送信息分配在数据区。传送信息的起始地址必须为地址 4n，如果不指定地址 4n，低 2 位就作为 “00b” 进行存取。

能通过短地址模式 (3 个长字) 或者全地址模式 (4 个长字) 分配传送信息。通过 DTCADM_{MOD}.SHORT 位设定短地址模式 (SHORT 位为 “1”) 或者全地址模式 (SHORT 位为 “0”)。

数据区的传送信息的分配如图 13.2 所示，DTC 按启动源从向量表读传送信息的起始地址并且从起始地址读传送信息。DTC 向量表和传送信息的对应如图 13.3 所示。

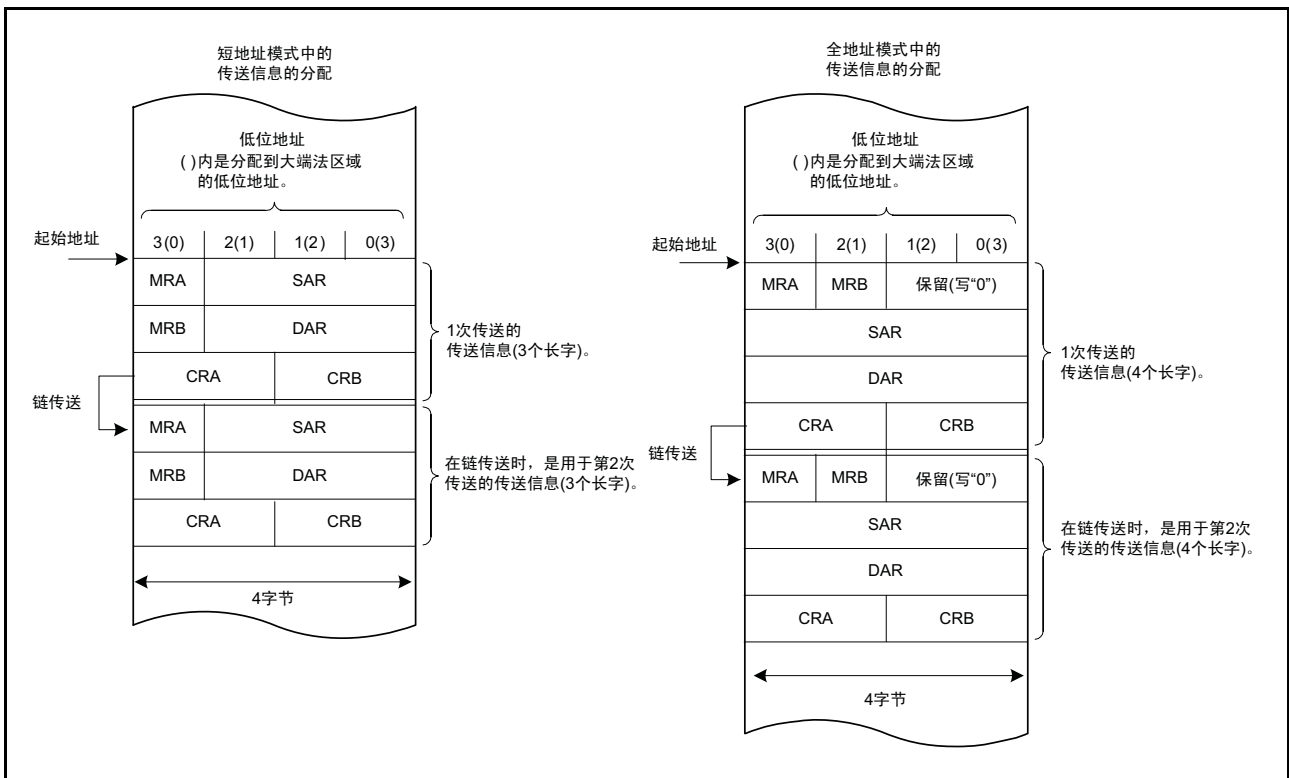


图 13.2 数据区的传送信息的分配

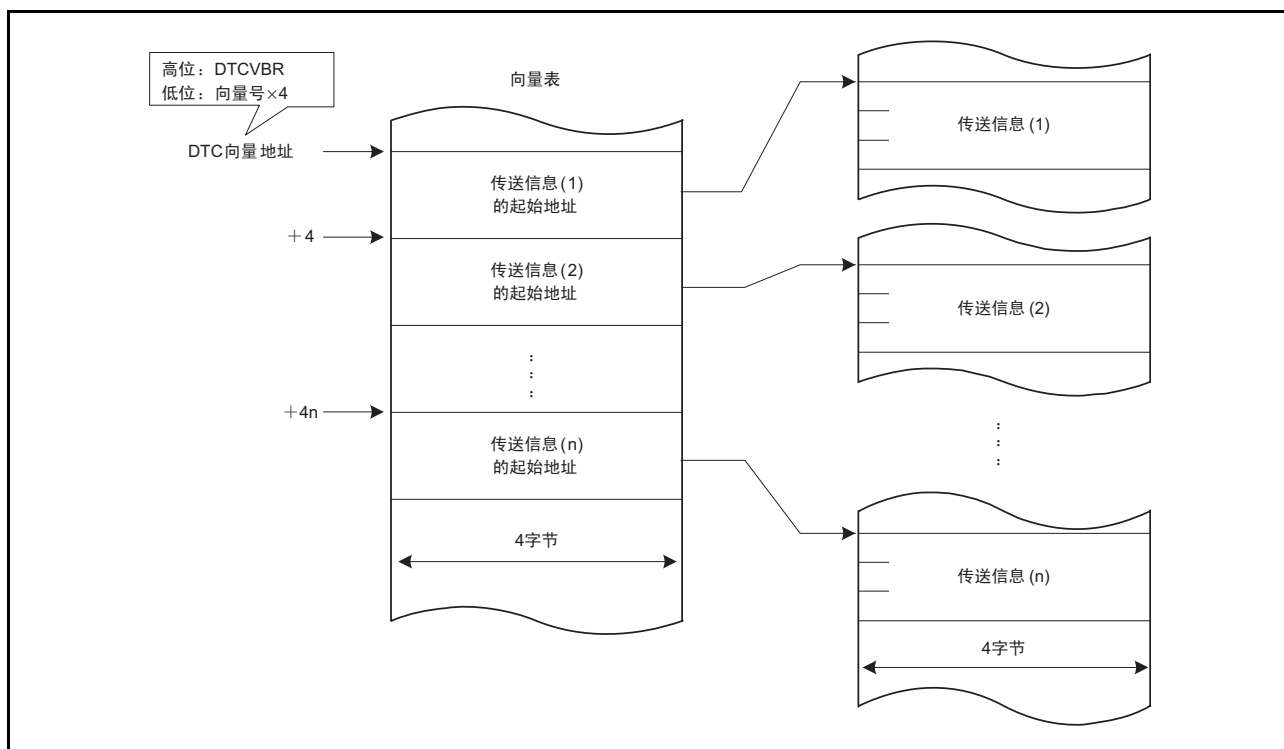


图 13.3 DTC 向量表和传送信息的对应

13.3.2 启动源和向量地址

DTC 启动源和向量地址的关系如表 13.3 所示。

表 13.3 中断源、DTC 向量地址和 ICU.ISELRi 寄存器的对应 (1/2)

| 启动请求发生源 | 启动源 | 向量号 | DTC 向量地址的偏移量 | ICU.ISELRi | 优先级 |
|----------|-------|-------|--------------|------------|--------|
| CMT 单元 0 | CMT0 | 28 | 0070h | ISELR028 | 高 ↑ |
| | CMT1 | 29 | 0074h | ISELR029 | |
| CMT 单元 1 | CMT2 | 30 | 0078h | ISELR030 | |
| | CMT3 | 31 | 007Ch | ISELR031 | |
| 外部引脚 | IRQ0 | 64 | 0100h | ISELR064 | |
| | IRQ1 | 65 | 0104h | ISELR065 | |
| | IRQ2 | 66 | 0108h | ISELR066 | |
| | IRQ3 | 67 | 010Ch | ISELR067 | |
| | IRQ4 | 68 | 0110h | ISELR068 | |
| | IRQ5 | 69 | 0114h | ISELR069 | |
| | IRQ6 | 70 | 0118h | ISELR070 | |
| | IRQ7 | 71 | 011Ch | ISELR071 | |
| | IRQ8 | 72 | 0120h | ISELR072 | |
| | IRQ9 | 73 | 0124h | ISELR073 | |
| | IRQ10 | 74 | 0128h | ISELR074 | |
| | IRQ11 | 75 | 012Ch | ISELR075 | |
| | IRQ12 | 76 | 0130h | ISELR076 | |
| | IRQ13 | 77 | 0134h | ISELR077 | |
| | IRQ14 | 78 | 0138h | ISELR078 | |
| IRQ15 | 79 | 013Ch | ISELR079 | | |
| AD0 | ADI0 | 98 | 0188h | ISELR098 | |
| AD1 | ADI1 | 99 | 018Ch | ISELR099 | |
| AD2 | ADI2 | 100 | 0190h | ISELR100 | |
| AD3 | ADI3 | 101 | 0194h | ISELR101 | |
| TPU0 | TGI0A | 104 | 01A0h | ISELR104 | |
| | TGI0B | 105 | 01A4h | ISELR105 | |
| | TGI0C | 106 | 01A8h | ISELR106 | |
| | TGI0D | 107 | 01ACh | ISELR107 | |
| TPU1 | TGI1A | 111 | 01BCh | ISELR111 | |
| | TGI1B | 112 | 01C0h | ISELR112 | |
| TPU2 | TGI2A | 117 | 01D4h | ISELR117 | |
| | TGI2B | 118 | 01D8h | ISELR118 | |
| TPU3 | TGI3A | 122 | 01E8h | ISELR122 | |
| | TGI3B | 123 | 01ECh | ISELR123 | |
| | TGI3C | 124 | 01F0h | ISELR124 | |
| | TGI3D | 125 | 01F4h | ISELR125 | |
| TPU4 | TGI4A | 127 | 01FCh | ISELR127 | |
| | TGI4B | 128 | 0200h | ISELR128 | |
| TPU5 | TGI5A | 133 | 0214h | ISELR133 | |
| | TGI5B | 134 | 0218h | ISELR134 | 低 ↓ |

13.4 运行说明

DTC 将传送信息保存到数据区。

一旦启动 DTC，就读与向量号对应的 DTC 向量。然后从 DTC 向量所示的传送信息的保存地址读传送信息，并且进行数据传送以及回写数据传送后的传送信息。能通过将传送信息保存到数据区，进行任意个通道的数据传送。

传送模式有正常传送模式、重复传送模式和块传送模式。

DTC 通过 SAR 寄存器指定传送源地址，通过 DAR 寄存器指定传送目标地址。在传送后，SAR 寄存器和 DAR 寄存器分别递增、递减或者为地址固定。

DTC 的传送模式如表 13.4 所示。

表 13.4 DTC 的传送模式

| 传送模式 | 1 次传送请求能传送的数据长度 | 存储器地址的增减 | 能指定的传送次数 |
|--------------|---|-----------------|-----------------|
| 正常传送模式 | 1 字节 / 字 / 长字 | 增减 1、2、4 或者地址固定 | 1 ~ 65536 次 |
| 重复传送模式 (注 1) | 1 字节 / 字 / 长字 | 增减 1、2、4 或者地址固定 | 1 ~ 256 次 (注 3) |
| 块传送模式 (注 2) | CRAH 寄存器指定的块大小 (1 ~ 256 字节 / 字 / 长字) | 增减 1、2、4 或者地址固定 | 1 ~ 65536 次 |

注 1. 将传送源或者传送目标设定为重复区域。

注 2. 将传送源或者传送目标设定为块区域。

注 3. 在指定次数的传送结束后，恢复初始状态并且继续（重复）运行。

能通过先将 MRB.CHNE 位置“1”，用 1 个启动源进行多次传送（链传送）。也能通过设定 MRB.CHNS 位，在传送计数器为“0”时进行链传送。

DTC 的运行流程图如图 13.4 所示，链传送的条件如表 13.5 所示（省略了第 2 次到第 3 次的传送组合以及第 3 次以后的传送组合）。

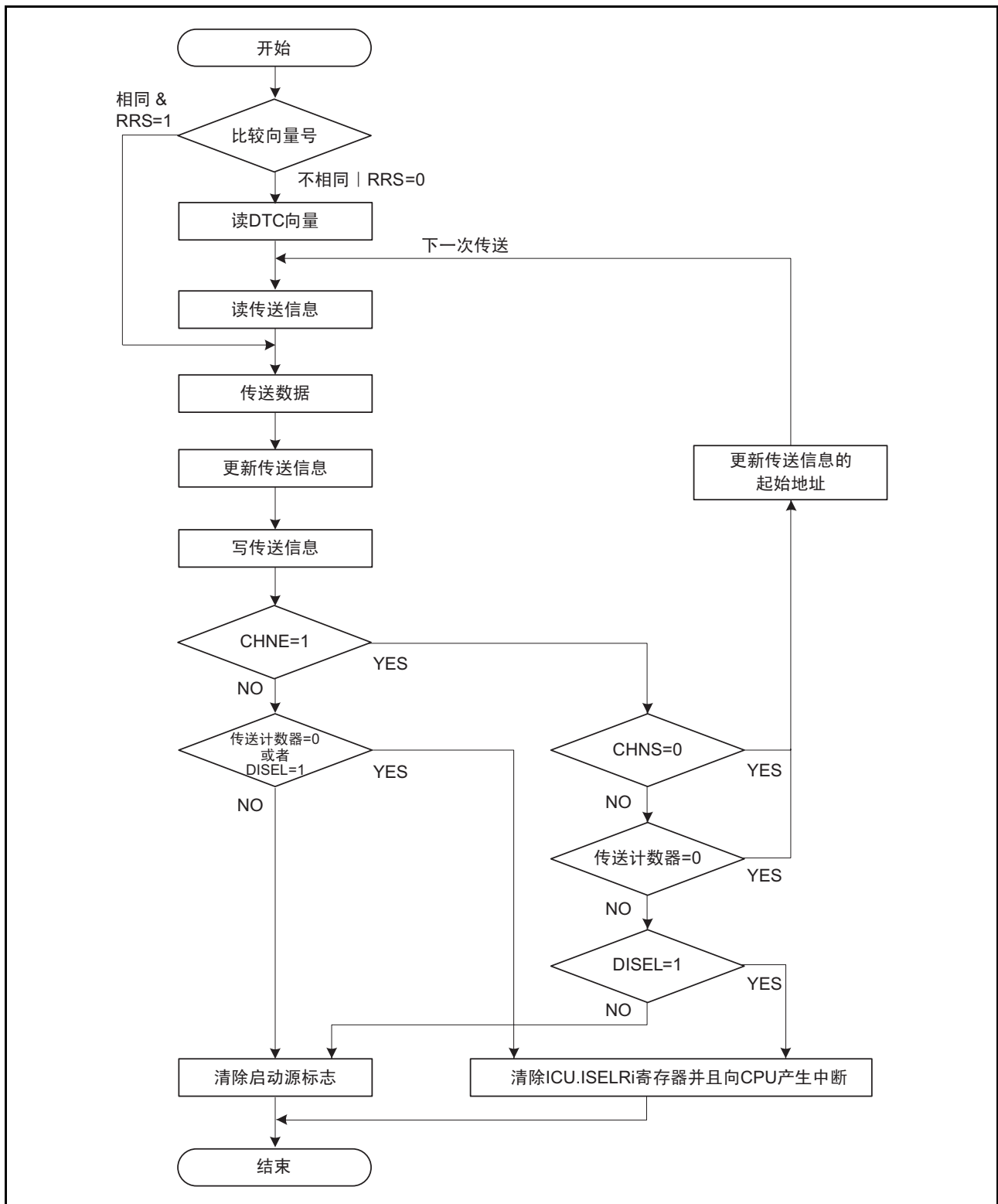


图 13.4 DTC 的运行流程图

表 13.5 链传送的条件

| 第 1 次传送 | | | | 第 2 次传送 | | | | DTC 传送 |
|---------|--------|---------|-------------|---------|--------|---------|-------------|---------------------------|
| CHNE 位 | CHNS 位 | DISEL 位 | 传送计数器 (注 1) | CHNE 位 | CHNS 位 | DISEL 位 | 传送计数器 (注 1) | |
| 0 | — | 0 | 非“0” | — | — | — | — | 到第 1 次传送为止结束。 |
| 0 | — | 0 | 0 (注 2) | — | — | — | — | 到第 1 次传送为止结束并且向 CPU 请求中断。 |
| 0 | — | 1 | — | — | — | — | — | |
| 1 | 0 | — | — | 0 | — | 0 | 非“0” | 到第 2 次传送为止结束。 |
| | | | | 0 | — | 0 | 0 (注 2) | 到第 2 次传送为止结束并且向 CPU 请求中断。 |
| | | | | 0 | — | 1 | — | |
| 1 | 1 | 0 | 非“0” | — | — | — | — | 到第 1 次传送为止结束。 |
| 1 | 1 | — | 0 (注 2) | 0 | — | 0 | 非“0” | 到第 2 次传送为止结束。 |
| | | | | 0 | — | 0 | 0 (注 2) | 到第 2 次传送为止结束并且向 CPU 请求中断。 |
| | | | | 0 | — | 1 | — | |
| 1 | 1 | 1 | 非“0” | — | — | — | — | 到第 1 次传送为止结束并且向 CPU 请求中断。 |

注 1. 正常传送模式: CRA 寄存器 重复传送模式: CRAL 寄存器 块传送模式: CRB 寄存器。

注 2. 在重复传送模式中, CRAL 寄存器的值被改写为 CRAH 寄存器的值时。

13.4.1 传送信息的跳读功能

能通过设定 DTCCR.RRS 位，跳读向量地址和传送信息。

在发生 DTC 启动请求时，这次启动的 DTC 向量号随时和上次启动的 DTC 向量号进行比较。如果比较结果相同并且 RRS 位为“1”，就不读向量地址和传送信息而进行 DTC 的数据传送。如果上次启动为链传送，就读向量地址和传送信息。如果上次传送为正常传送并且传送计数器 (CRA 寄存器) 变为“0”或者上次传送为块传送并且传送计数器 (CRB 寄存器) 变为“0”，就都读传送信息，与 RRS 位的值无关。跳读传送信息的例子如图 13.5 所示。

要更新向量表和传送信息时，必须先将 RRS 位置“0”并且在更新向量表和传送信息后设定 RRS 位。一旦将 RRS 位置“0”，保持的向量号就被取消，在下次启动时读被更新的向量表和传送信息。

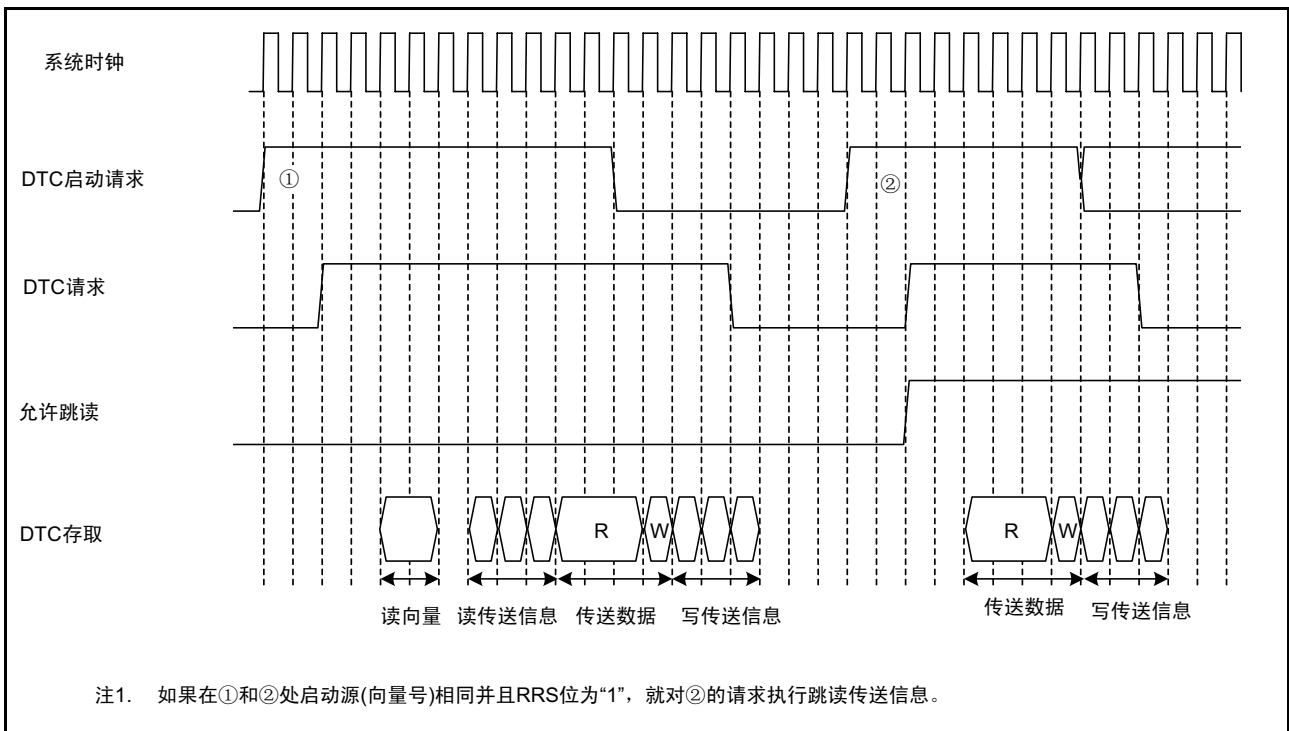


图 13.5 跳读传送信息的例子 (向量、传送信息和传送目标为内部 RAM，传送源为外围模块的情况)

13.4.2 传送信息的回写省略功能

如果将 MRA.SM[1:0] 位或者 MRB.DM[1:0] 位设定为“地址固定”，就不回写部分传送信息，此功能与短地址模式和全地址模式的设定无关。传送信息的回写省略条件以及省略回写的寄存器如表 13.6 所示。

与短地址模式和全地址模式的设定无关，回写 CRA 寄存器和 CRB 寄存器。在全地址模式中，省略 MRA 寄存器和 MRB 寄存器的回写。

表 13.6 传送信息的回写省略条件以及省略回写的寄存器

| MRA.SM[1:0] 位 | | MRB.DM[1:0] 位 | | SAR 寄存器 | DAR 寄存器 |
|---------------|----|---------------|----|---------|---------|
| b3 | b2 | b3 | b2 | | |
| 0 | 0 | 0 | 0 | 省略 | 省略 |
| 0 | 0 | 0 | 1 | | |
| 0 | 1 | 0 | 0 | | |
| 0 | 1 | 0 | 1 | | |
| 0 | 0 | 1 | 0 | 省略 | 回写 |
| 0 | 0 | 1 | 1 | | |
| 0 | 1 | 1 | 0 | | |
| 0 | 1 | 1 | 1 | | |
| 1 | 0 | 0 | 0 | 回写 | 省略 |
| 1 | 0 | 0 | 1 | | |
| 1 | 1 | 0 | 0 | | |
| 1 | 1 | 0 | 1 | | |
| 1 | 0 | 1 | 0 | 回写 | 回写 |
| 1 | 0 | 1 | 1 | | |
| 1 | 1 | 1 | 0 | | |
| 1 | 1 | 1 | 1 | | |

13.4.3 正常传送模式

用 1 个启动源进行 1 个字节、1 个字或者 1 个长字的传送，传送次数为 1 ~ 65536 次。

能将传送源地址和传送目标地址分别设定为递增、递减或者固定。一旦指定次数的传送结束，就能向 CPU 请求中断。

正常传送模式的寄存器功能和存储器映像分别如表 13.7 和图 13.6 所示。

表 13.7 正常传送模式的寄存器功能

| 寄存器 | 功能 | 写传送信息时被回写的值 |
|-----|---------|--------------------|
| SAR | 传送源地址 | 递增 / 递减 / 固定 (注 1) |
| DAR | 传送目标地址 | 递增 / 递减 / 固定 (注 1) |
| CRA | 传送计数器 A | CRA-1 |
| CRB | 传送计数器 B | 不更新 |

注 1. 在地址固定的情况下，省略回写。

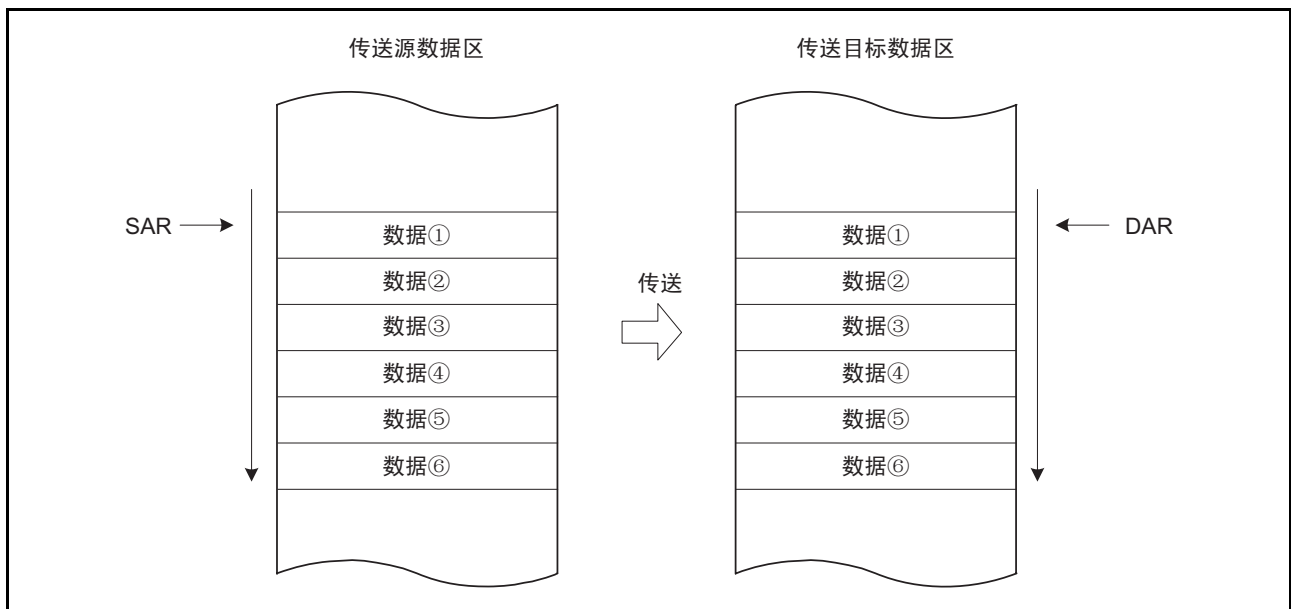


图 13.6 正常传送模式的存储器映像

13.4.4 重复传送模式

用 1 个启动源进行 1 个字节、1 个字或者 1 个长字的传送。

通过 MRB.DTS 位将传送源或者传送目标指定为重复区域，传送次数可指定为 1 ~ 256 次。一旦指定次数的传送结束，传送计数器和被指定为重复区域的地址寄存器就恢复初始状态并且进行重复传送。其他的地址寄存器继续递增、递减或者为地址固定。

在重复传送模式中，如果传送计数器的 CRAL 寄存器的值变为 “00h”，CRAL 寄存器的值就被更新为 CRAH 寄存器的设定值。因为传送计数器不变为 “00h”，所以在 MRB.DISEL 位为 “0”（在指定的数据传送结束时向 CPU 请求中断）时不向 CPU 请求中断。

重复传送模式的寄存器功能和存储器映像分别如表 13.8 和图 13.7 所示。

表 13.8 重复传送模式的寄存器功能

| 寄存器 | 功能 | 写传送信息时被回写的值 | |
|------|----------|--------------------|---|
| | | CRAL≠1 | CRAL=1 |
| SAR | 传送源地址 | 递增 / 递减 / 固定 (注 1) | (当 MRB.DTS 位 =0 时) 递增 / 递减 / 固定 (注 1) (当 MRB.DTS 位 =1 时) SAR 寄存器的初始值 |
| DAR | 传送目标地址 | 递增 / 递减 / 固定 (注 1) | (当 MRB.DTS 位 =0 时) DAR 寄存器的初始值 (当 MRB.DTS 位 =1 时) 递增 / 递减 / 固定 (注 1) |
| CRAH | 传送计数器的保持 | CRAH | CRAH |
| CRAL | 传送计数器 A | CRAL-1 | CRAH |
| CRB | 传送计数器 B | 不更新 | 不更新 |

注 1. 在地址固定的情况下，省略回写。

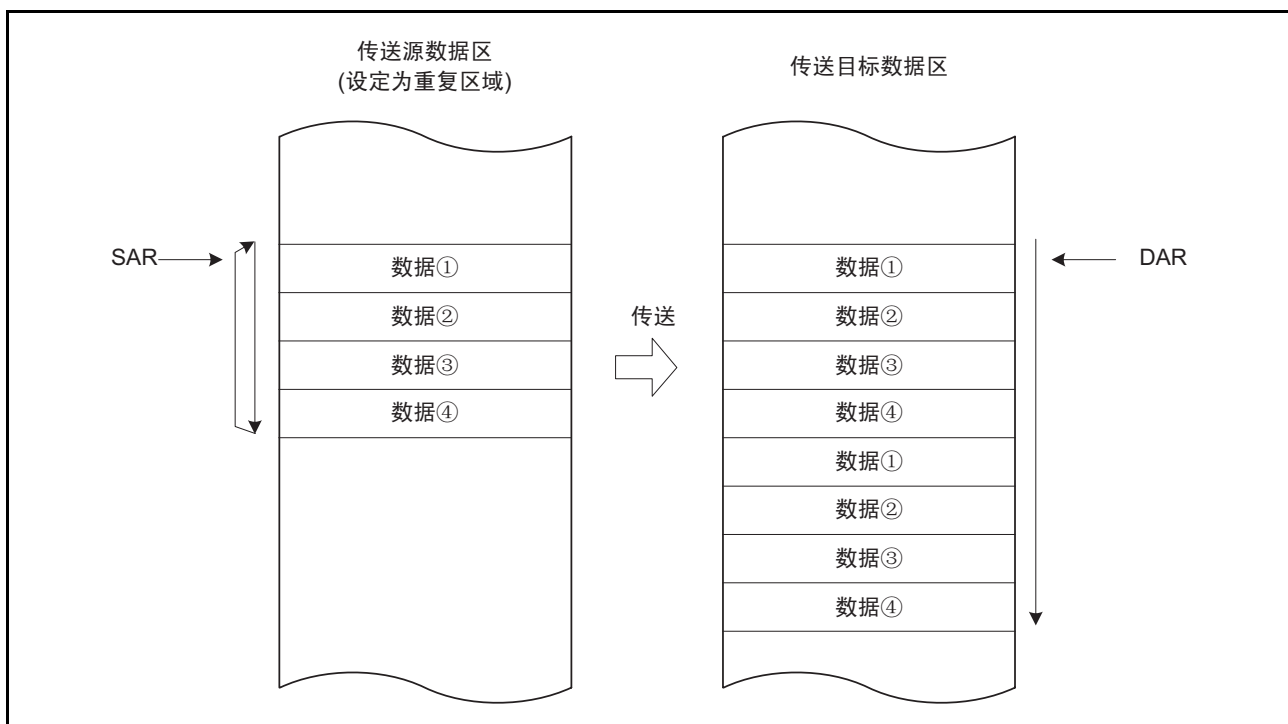


图 13.7 重复传送模式的存储器映像 (将传送源设定为重复区域的情况)

13.4.5 块传送模式

用 1 个启动源进行 1 个块的传送。

通过 MRB.DTS 位将传送源或者传送目标指定为块区域，块大小可指定为 1 ~ 256 字节（或者 1 ~ 256 个字、1 ~ 256 个长字）。

一旦指定的 1 个块的传送结束，块大小计数器的 CRAL 寄存器和被指定为块区域的地址寄存器（当 MRB.DTS 位为“1”时为 SAR 寄存器，当 MRB.DTS 位为“0”时为 DAR 寄存器）就恢复初始状态。其他的地址寄存器继续递增、递减或者为地址固定。

传送次数（块次数）可指定为 1 ~ 65536 次。一旦指定次数的块传送结束，就能向 CPU 请求中断。

块传送模式的寄存器功能和存储器映像分别如表 13.9 和图 13.8 所示。

表 13.9 块传送模式的寄存器功能

| 寄存器 | 功能 | 写传送信息时被回写的值 |
|------|----------|---|
| SAR | 传送源地址 | (当 MRB.DTS 位 =0 时) 递增 / 递减 / 固定 (注 1) (当 MRB.DTS 位 =1 时) SAR 寄存器的初始值 |
| DAR | 传送目标地址 | (当 MRB.DTS 位 =0 时) DAR 寄存器的初始值 (当 MRB.DTS 位 =1 时) 递增 / 递减 / 固定 (注 1) |
| CRAH | 块大小的保持 | CRAH |
| CRAL | 块大小计数器 | CRAH |
| CRB | 块传送次数计数器 | CRB-1 |

注 1. 在地址固定的情况下，省略回写。

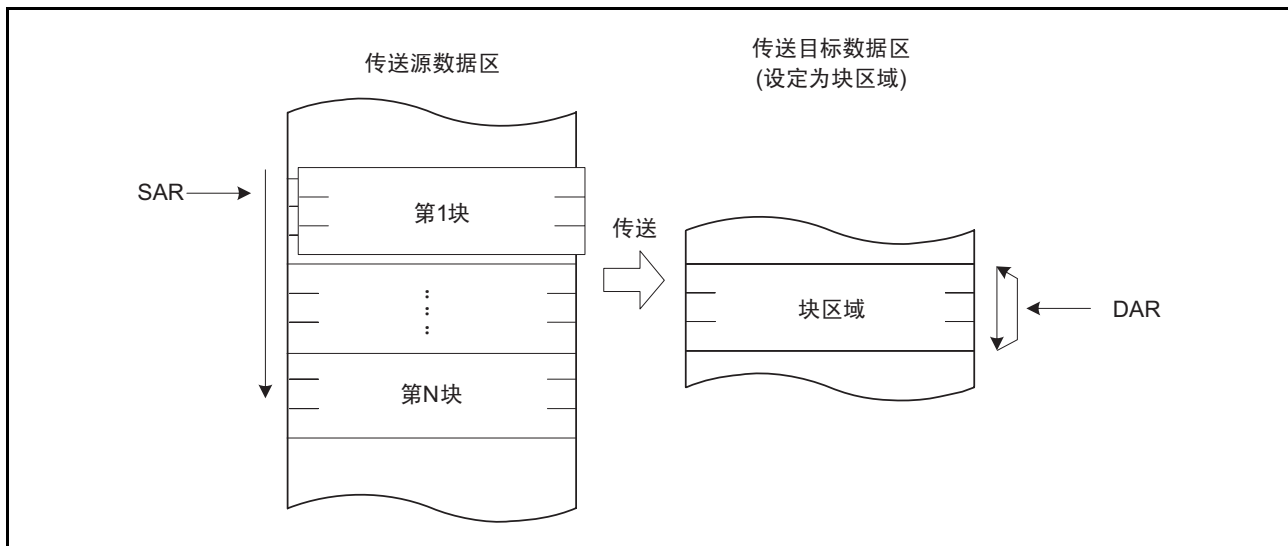


图 13.8 块传送模式的存储器映像（将传送目标指定为块区域的情况）

13.4.6 链传送

如果将 MRB.CHNE 位置 “1”，就能用 1 个启动源连续进行多个数据的传送。能分别设定要定义数据传送的 SAR、DAR、CRA、CRB、MRA、MRB 寄存器。链传送的运行如图 13.9 所示。

如果将 MRB.CHNE 位和 MRB.CHNS 位分别置 “1” 和 “0”，就不通知 CPU 中断，也不清除作为启动源的中断源标志，而根据下一个传送信息进行传送（链传送）。

如果将 MRB.CHNE 位和 MRB.CHNS 位都置 “1”，就只在传送计数器为 “0” 时不通知 CPU 中断，也不清除作为启动源的中断源标志，而根据下一个传送信息进行传送（链传送）。如果传送计数器不为 “0”，就通过设定 MRB.DISEL 位，通知 CPU 中断以及清除作为启动源的中断源标志。

如果进行链传送，就在根据 MRB.CHNE 位为 “0” 的传送信息进行的传送结束时，通知 CPU 中断或者清除作为启动源的中断源标志。

链传送的运行流程图的详细内容请参照“图 13.4 DTC 的运行流程图”和“表 13.5 链传送的条件”。

在重复传送模式中，如果将 DTCCR.RCHNE 位、MRB.CHNE 位和 MRB.CHNS 位分别置 “1”，就能在传送计数器为 “1” 的传送结束后进行链传送。

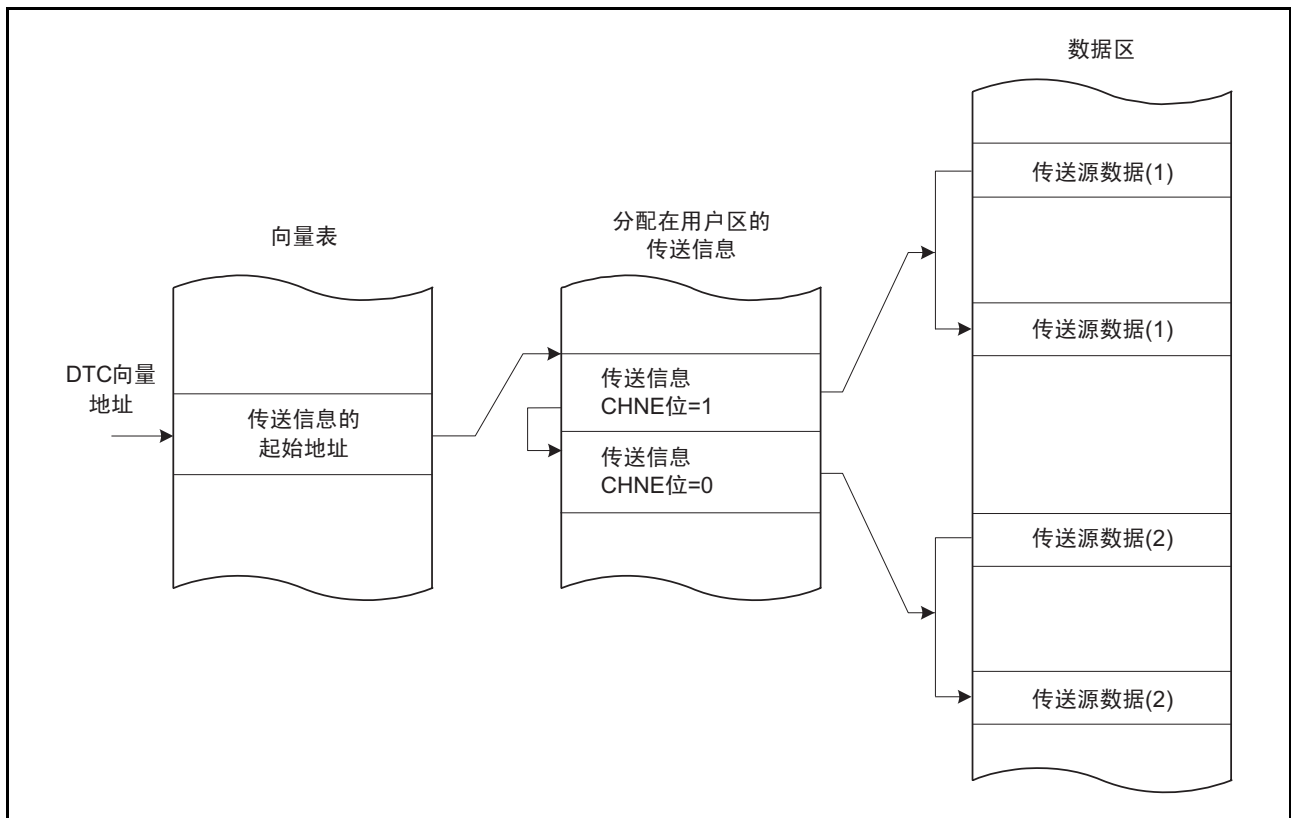


图 13.9 链传送的运行

13.4.7 运行时序

DTC 的运行时序例子如图 13.10 ~ 图 13.13 所示。

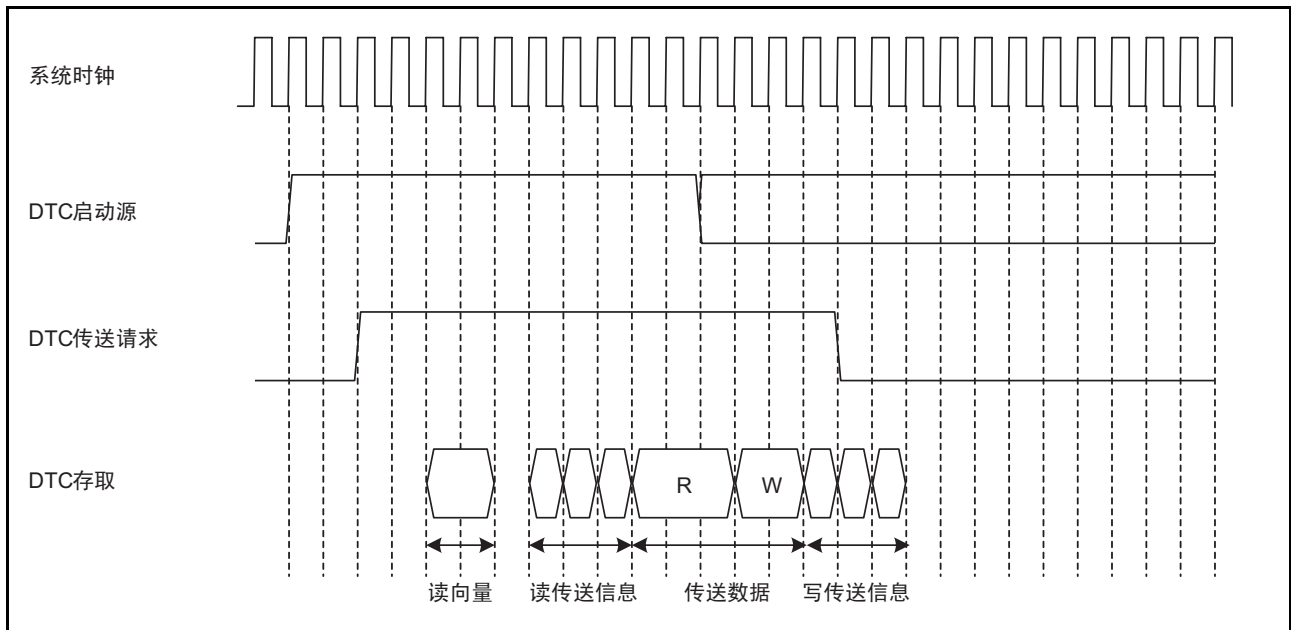


图 13.10 DTC 的运行时序例子 1 (短地址模式、正常传送模式、重复传送模式的情况)

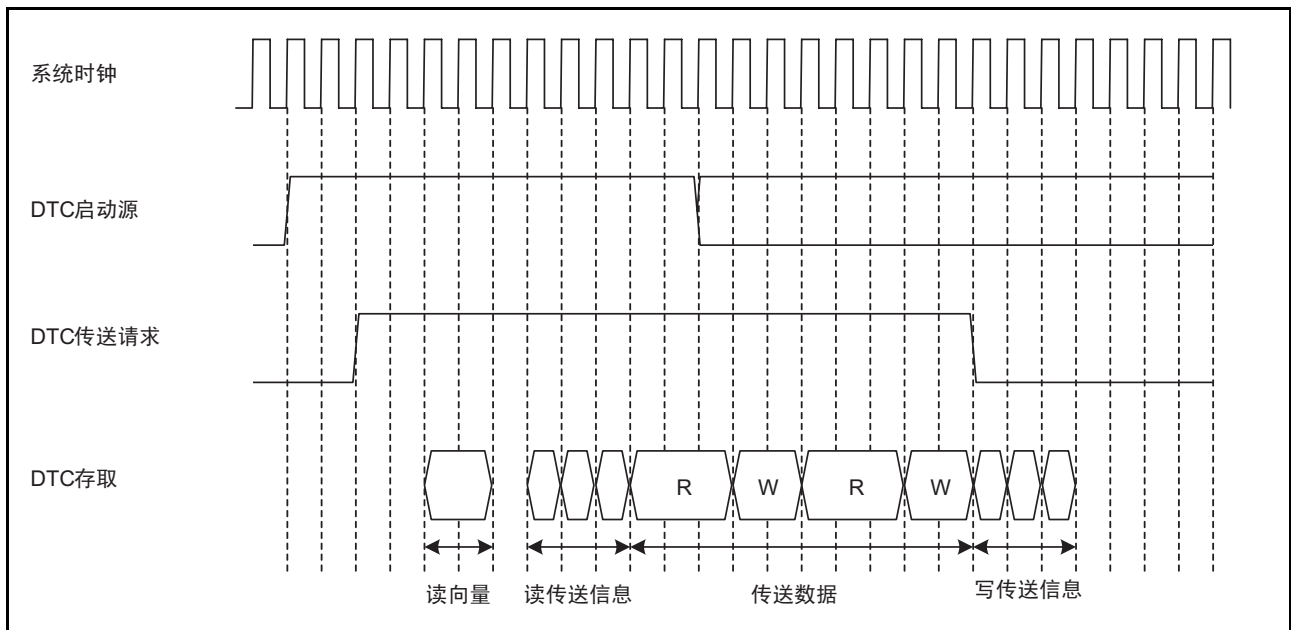


图 13.11 DTC 的运行时序例子 2 (短地址模式、块传送模式、块大小 = 2 的情况)

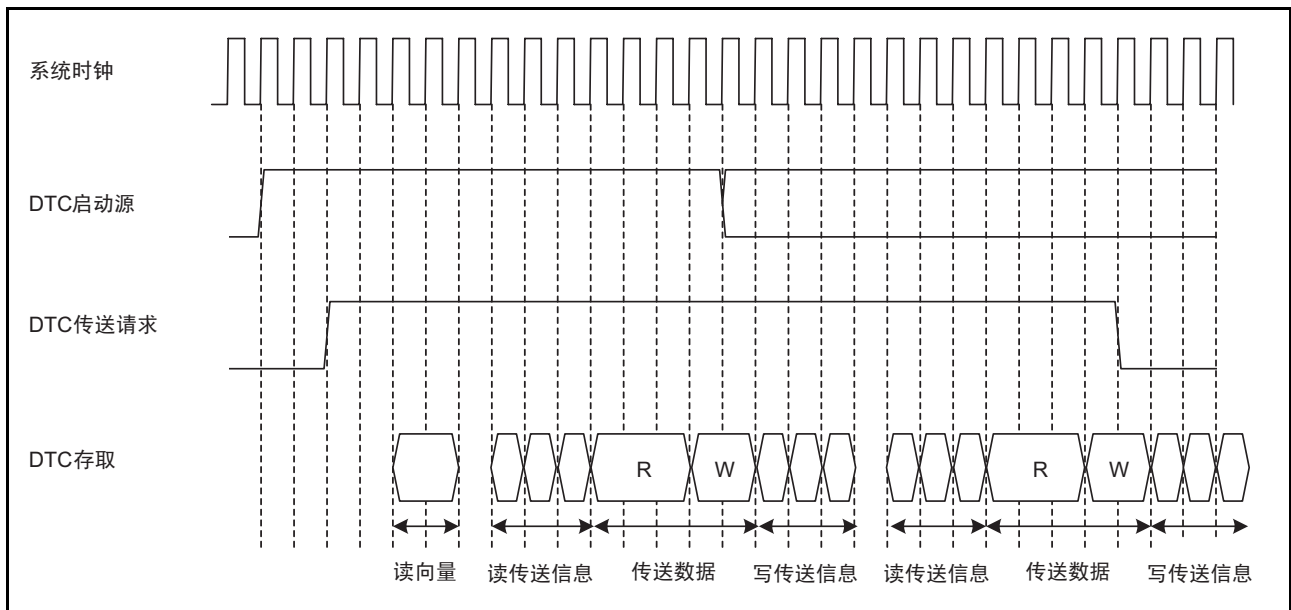


图 13.12 DTC 的运行时序例子 3 (短地址模式、链传送的情况)

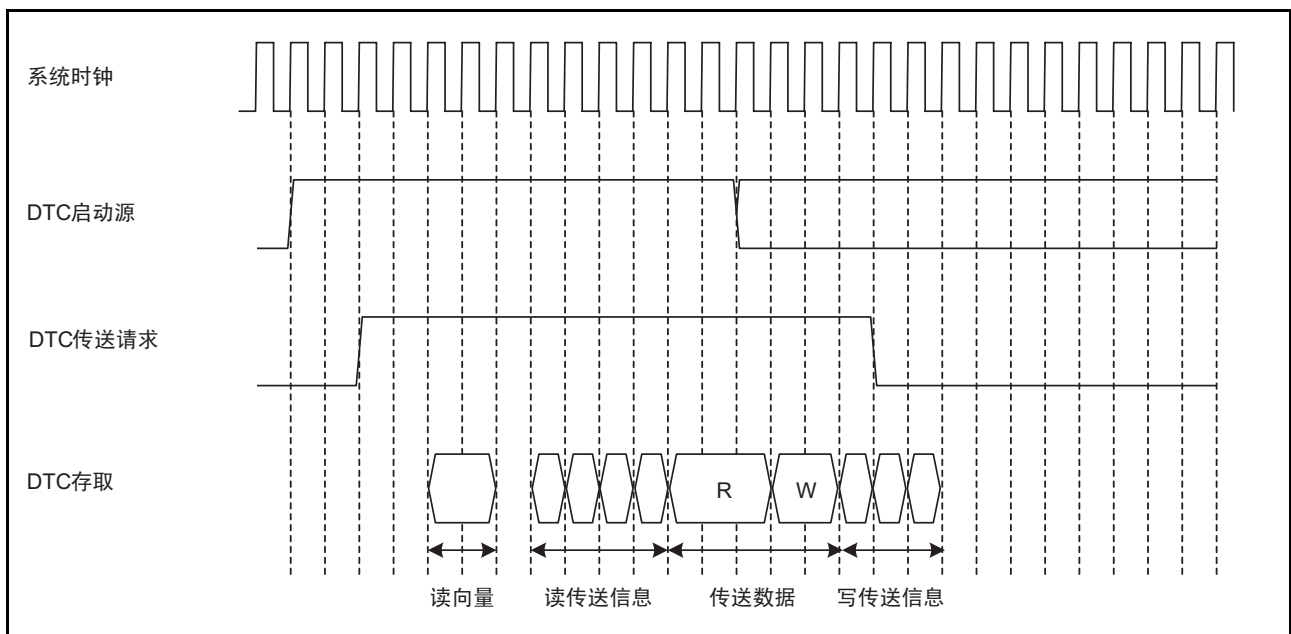


图 13.13 DTC 的运行时序例子 4 (全地址模式、正常传送模式、重复传送模式的情况)

13.4.8 DTC 的执行状态

DTC 的 1 次数据传送的执行状态如表 13.10 所示。

表 13.10 DTC 的执行状态

| 传送模式 | 读向量 | | 读传送信息 | | | 写传送信息 | | | 读数据 | 写数据 | 内部运行 | |
|------|-----|--------|--------|--------|--------|-----------|--------|--------|-----|-----|------|--------|
| 正常 | 1 | 0 (注1) | 4 (注2) | 3 (注3) | 0 (注1) | 3 (注2、注3) | 2 (注4) | 1 (注5) | 1 | 1 | 1 | 0 (注1) |
| 重复 | 1 | 0 (注1) | 4 (注2) | 3 (注3) | 0 (注1) | 3 (注2、注3) | 2 (注4) | 1 (注5) | 1 | 1 | 1 | 0 (注1) |
| 块 | 1 | 0 (注1) | 4 (注2) | 3 (注3) | 0 (注1) | 3 (注2、注3) | 2 (注4) | 1 (注5) | 1·P | 1·P | 1 | 0 (注1) |

P: 块大小 (CRAH 寄存器和 CRAL 寄存器的初始设定)

- 注 1. 这是省略传送信息的情况。
- 注 2. 这是全地址模式运行的情况。
- 注 3. 这是短地址模式运行的情况。
- 注 4. SAR 寄存器或者 DAR 寄存器为地址固定的情况。
- 注 5. SAR 寄存器和 DAR 寄存器为地址固定的情况。

13.4.9 DTC 的总线权释放时序

在读写传送信息过程中, DTC 不释放总线权。在其他时序中, 根据总线主控仲裁部决定的优先级进行总线仲裁。

13.5 DTC 的设定步骤

DTC 的设定步骤如图 13.14 所示。

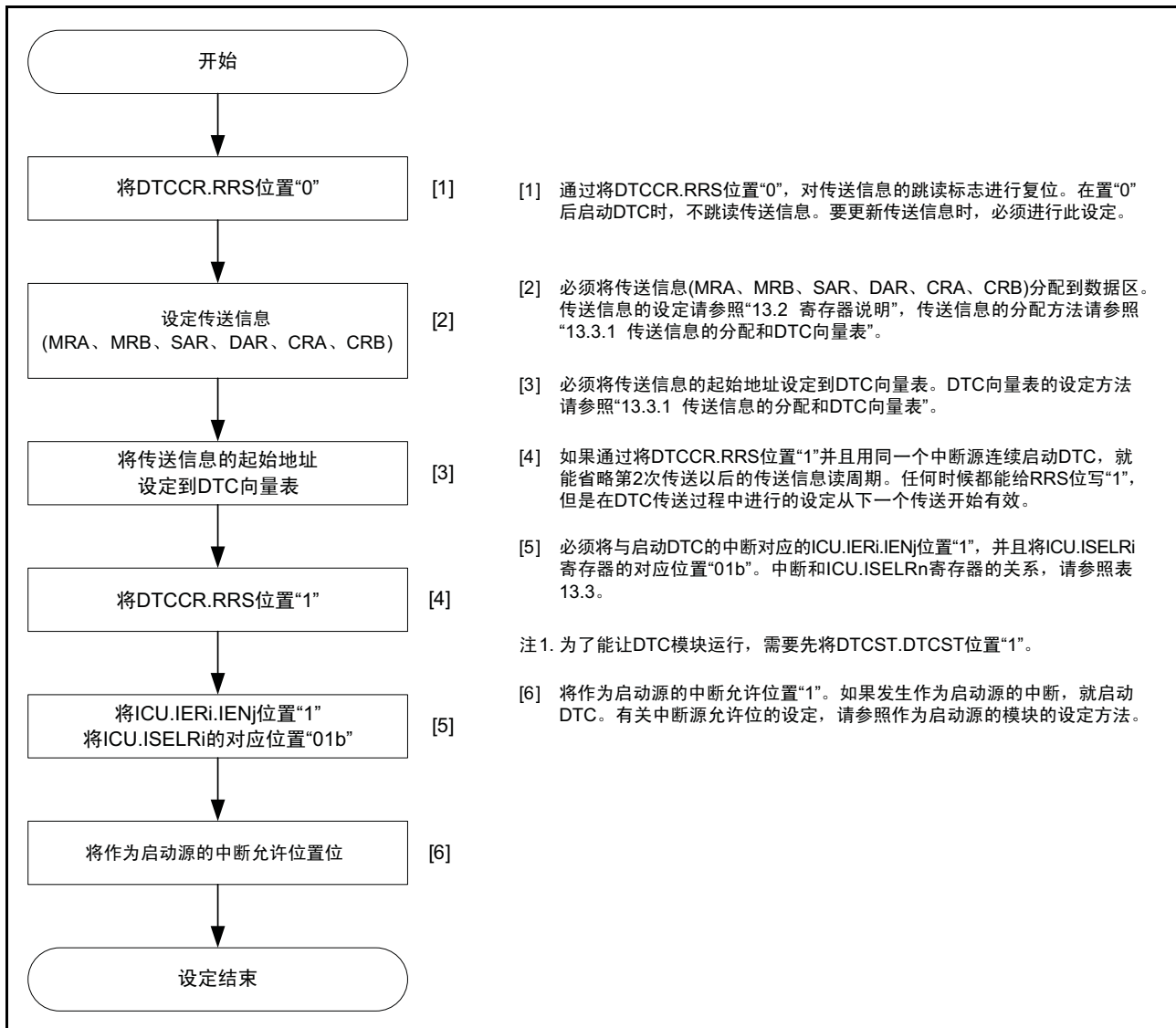


图 13.14 DTC 的设定步骤

13.6 DTC 的使用例子

13.6.1 正常传送

以 SCI 进行的 128 字节数据接收为例说明 DTC 的使用。

1. 给 MRA 寄存器设定传送源地址固定 (MRA.SM[1:0]位=00b)、传送目标地址递增 (MRB.DM[1:0]位=10b)、正常传送模式 (MRA.MD[1:0]位=00b) 和字节传送 (MRA.SZ[1:0]位=00b)。MRB.DTS 位可设定为任意值。MRB 寄存器通过 1 次中断进行 1 次数据传送 (MRB.CHNE 位=0, MRB.DISEL 位=0)。给 SAR 寄存器设定 SCIm.RDR 寄存器 (m=0~6) 的地址; 给 DAR 寄存器设定用于保存数据的 RAM 起始地址; 给 CRA 寄存器设定 128 (“0080h”)。CRB 寄存器可设定为任意值。
2. 将用于 RXI 中断的传送信息的起始地址设定到 DTC 向量表。
3. 将对应的 ICU.ISEL_{Ri} 寄存器置 “01b”, 将 ICU.IER_i.IEN_j 位和 DTCST.DTCST 位置 “1”。
4. 将 SCI 设定为规定的接收模式。将 SCIm.SCR.RIE 位置 “1” 并且允许接收结束 (RXI) 中断。如果在 SCI 接收过程中发生接收错误, 就不进行以后的接收, 因此必须设定为 CPU 能接受接收错误中断。
5. 在每次 SCI 的 1 字节数据接收结束时发生 RXI 中断, 并且启动 DTC。通过 DTC 将接收数据从 SCIm.RDR 寄存器传送到 RAM, 并且进行 DAR 寄存器的递增和 CRA 寄存器的递减。
6. 如果在 128 次的数据传送结束后 CRA 寄存器变为 “0”, 就向 CPU 请求 RXI 中断。必须通过中断处理程序进行结束处理。

13.6.2 链传送

以 PPG 进行的脉冲输出为例说明 DTC 的链传送。

能使用链传送进行脉冲输出数据的传送以及 PPG 输出触发周期的变更。设定为: 在链传送的前半部分将数据重复传送到 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器 (m=0,1), 在后半部分将数据正常传送到 TPUm.TGRA ~ TPUm.TGRD 寄存器 (m=0~11)。这是因为只能在链传送的后半部分 (MRB.CHNE 位为 “0” 的传送) 清除启动源以及在指定次数的传送结束时发生中断。

1. 对 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器的传送进行设定。给 MRA 寄存器设定传送源地址递增 (MRA.SM[1:0]位=10b)、传送目标地址固定 (MRB.DM[1:0]位=00b)、重复传送模式 (MRA.MD[1:0]位=01b)、字传送 (MRA.SZ[1:0]位=01b)。将传送源设定为重复区域 (MRB.DTS 位=1)。给 MRB 寄存器设定链传送 (MRB.CHNE 位=1, MRB.CHNS 位=0, MRB.DISEL 位=0)。给 SAR 寄存器设定数据表的起始地址, 给 DAR 寄存器设定 PPGm.NDRH 寄存器的地址, 给 CRAH 寄存器和 CRAL 寄存器设定数据表的大小。CRB 寄存器可设定为任意值。
2. 对 TPUm.TGRA 寄存器的传送进行设定。给 MRA 寄存器设定传送源地址递增 (MRA.SM[1:0]位=10b)、传送目标地址固定 (MRB.DM[1:0]位=00b)、正常传送模式 (MRA.MD[1:0]位=00b)、字传送 (MRA.SZ[1:0]位=01b)。给 SAR 寄存器设定数据表的起始地址, 给 DAR 寄存器设定 TPUm.TGRA 寄存器的地址, 给 CRA 寄存器设定数据表的大小。CRB 寄存器可设定为任意值。
3. 在用于传送到 PPGm 的传送信息后面连续配置用于传送到 TPU 的传送信息。
4. 将用于传送到 PPGm 的传送信息的起始地址设定到 DTC 向量表。
5. 将 TGIA 中断对应的 ICU.ISEL_{Ri} 寄存器置 “01b”, 将 ICU.IER_i.IEN_j 位和 DTCST.DTCST 位置 “1”。
6. 通过 TPUm.TIORH 寄存器和 TPUm.TIORL 寄存器将 TPUm.TGRA 寄存器设定为输出比较寄存器 (禁止输出), 通过 TPUm.TIER 寄存器允许 TGIA 中断。
7. 给 PPGm.PODRH 寄存器和 PPGm.PODRL 寄存器设定输出的初始值, 给 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器设定下一个输出值。将 Pm.DDR 寄存器 (m=0~9, A~E) 以及进行 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器的输出的对应位置 “1”。通过 PPGm.PCR 寄存器 (m=0,1) 选择作为输出触发的 TPU 比较匹配信号。
8. 将 TPU.TSTR_y.CST[5:0] 位 (y=A,B) 置 “1”, 开始 TPUm.TCNT 计数器的计数。
9. 在每次发生 TPUm.TGRA 寄存器的比较匹配时, 将下一个输出值传送到 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器并且将下一个输出触发周期的设定值传送到 TPUm.TGRA 寄存器。
10. 在指定次数的传送结束后 (用于 TPU 传送的 CRA 寄存器变为 “0”), 向 CPU 请求 TGIA 中断。必须通过中断处理程序进行结束处理。

13.6.3 计数器为“0”时的链传送

只能在计数器变为“0”时进行第2次数据的传送。能通过重新设定第1次数据的传送，至少进行256次的重复传送。

这是构成128K字节输入缓冲器的例子，假设输入缓冲器从低位地址“0000h”开始。计数器为“0”时的链传送如图13.15所示。

1. 作为第1次数据传送，设定输入数据的正常传送模式。传送源地址必须固定，CRA寄存器为“0000h”（65536次），MRB.CHNE位为“1”（允许链传送），MRB.CHNS位为“1”（只在传送计数器为“0”时进行链传送），MRB.DISEL位为“0”（在指定的数据传送结束时向CPU请求中断）。
2. 必须在其他区域（ROM等）准备第1次数据传送的传送目标地址的每65536次的起始地址高8位。例如，当输入缓冲器为“20000h”~“21FFFFh”时，准备“21h”和“20h”。
3. 作为第2次数据传送，设定为用于重新设定第1次数据传送的传送目标地址的重复传送模式（传送源为重复区域），传送目标为第1次传送信息区的DAR寄存器的高8位。此时，必须将MRB.CHNE位置“0”（禁止链传送），并且将MRB.DISEL位置“0”（在指定的数据传送结束时向请求CPU中断）。当上述输入缓冲器为“20000h”~“21FFFFh”时，必须将传送计数器设定为“2”。
4. 通过中断进行65536次的第1次数据的传送。如果第1次数据传送的传送计数器变为“0”，就启动第2次数据传送。将第1次数据传送的传送源地址高8位置“21h”，第1次数据传送的传送目标地址低16位的传送计数器变为“0000h”。
5. 通过中断继续进行在第1次数据传送指定的65536次的第1次数据传送。如果第1次数据传送的传送计数器变为“0”，就启动第2次数据传送。将第1次数据传送的传送源地址高8位置“20h”，第1次的数据传送的传送目标地址低16位的传送计数器变为“0000h”。
6. 无限重复上述的第4步骤和第5步骤。因为第2次数据传送为重复传送模式，所以不向CPU请求中断。

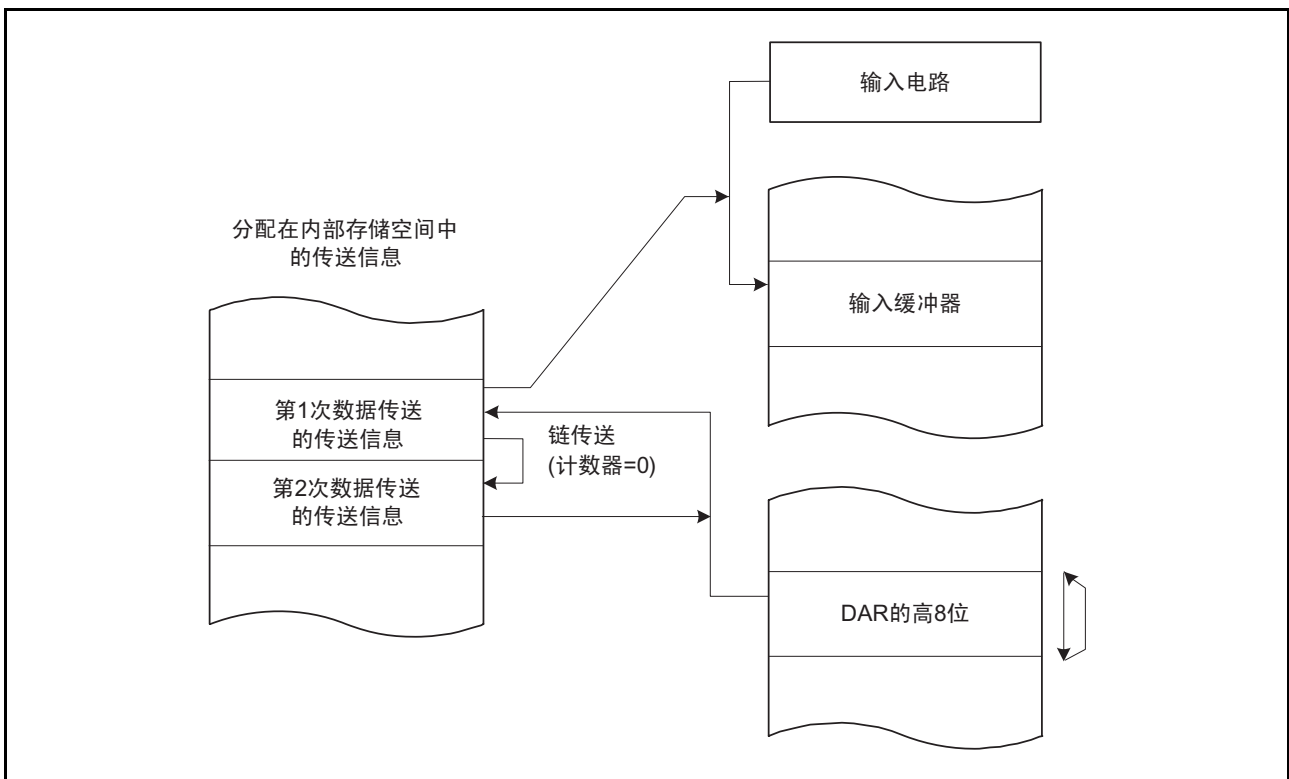


图 13.15 计数器为“0”时的链传送

13.7 中断源

在结束 DTC 指定次数的数据传送以及结束 MRB.DISEL 位为“1”（在每次 DTC 数据传送时向 CPU 请求中断）的数据传送时，通过启动 DTC 的中断源向 CPU 请求中断。这些向 CPU 请求的中断受 CPU 的屏蔽级和中断控制器的优先级的控制。

13.8 低功耗功能

13.8.1 DTC 模块启动寄存器的设定

在向模块停止状态以及向全模块时钟停止模式、软件待机模式、深度软件待机模式转移时，必须将 DTCST.DTCST 位置“0”（模块停止）。

(1) 模块停止功能

通过给 MSTPCRA.MSTPA27 位写“1”（向模块停止状态转移），使 DTC 的模块停止功能有效。如果在给 MSTPCRA.MSTPA27 位写 1 时 DTC 正在进行传送，就在 DTC 传送结束后转移到模块停止状态。

(2) 向全模块时钟停止模式的转移

在给 MSTPCRA.ACSE 位写“1”（允许全模块时钟停止模式）并且给包括 MSTPCRA.MSTPA27 位（DTC 模块停止设定位）在内的 MSTPCRA.MSTPAj 和 MSTPCRB.MSTPBj 的全部位写“1”后，通过执行 WAIT 指令转移到全模块时钟停止模式。如果在执行 WAIT 指令时 DTC 正在进行传送，就在 DTC 传送结束后转移到全模块时钟停止模式。

(3) 向软件待机模式或者深度软件待机模式的转移

在给 SBYCR.SSBY 位写“1”（在执行 WAIT 指令后转移到软件待机模式）后，通过执行 WAIT 指令转移到软件待机模式。如果在执行 WAIT 指令时 DTC 正在进行传送，就在 DTC 传送结束后转移到软件待机模式。另外，在转移到软件待机模式时，如果已将 DPSBYCR.DPSBY 位置“1”，就转移到深度软件待机模式。

13.9 使用时的注意事项

13.9.1 传送信息起始地址、传送源地址和传送目标地址

必须给向量表中指定的传送信息的起始地址指定地址 $4n$ ，否则地址的低 2 位就作为 “00b” 进行存取。

13.9.2 传送信息的分配

如图 13.16 所示，在将传送信息分配到存储器时，必须根据要分配区域的字节序进行分配。

例如，当用 16 位写 CRA 和 CRB 的设定数据时，在大端法的情况下，必须将 CRA 和 CRB 的设定数据分别写到低位地址 0 和低位地址 2；在小端法的情况下，必须将 CRA 和 CRB 的设定数据分别写到低位地址 2 和低位地址 0。当用 32 位写 CRA 和 CRB 的设定数据时，与字节序无关，必须在将 CRA 和 CRB 的设定数据分别分配到 32 位的 MSB 侧和 LSB 侧后写到低位地址 0。

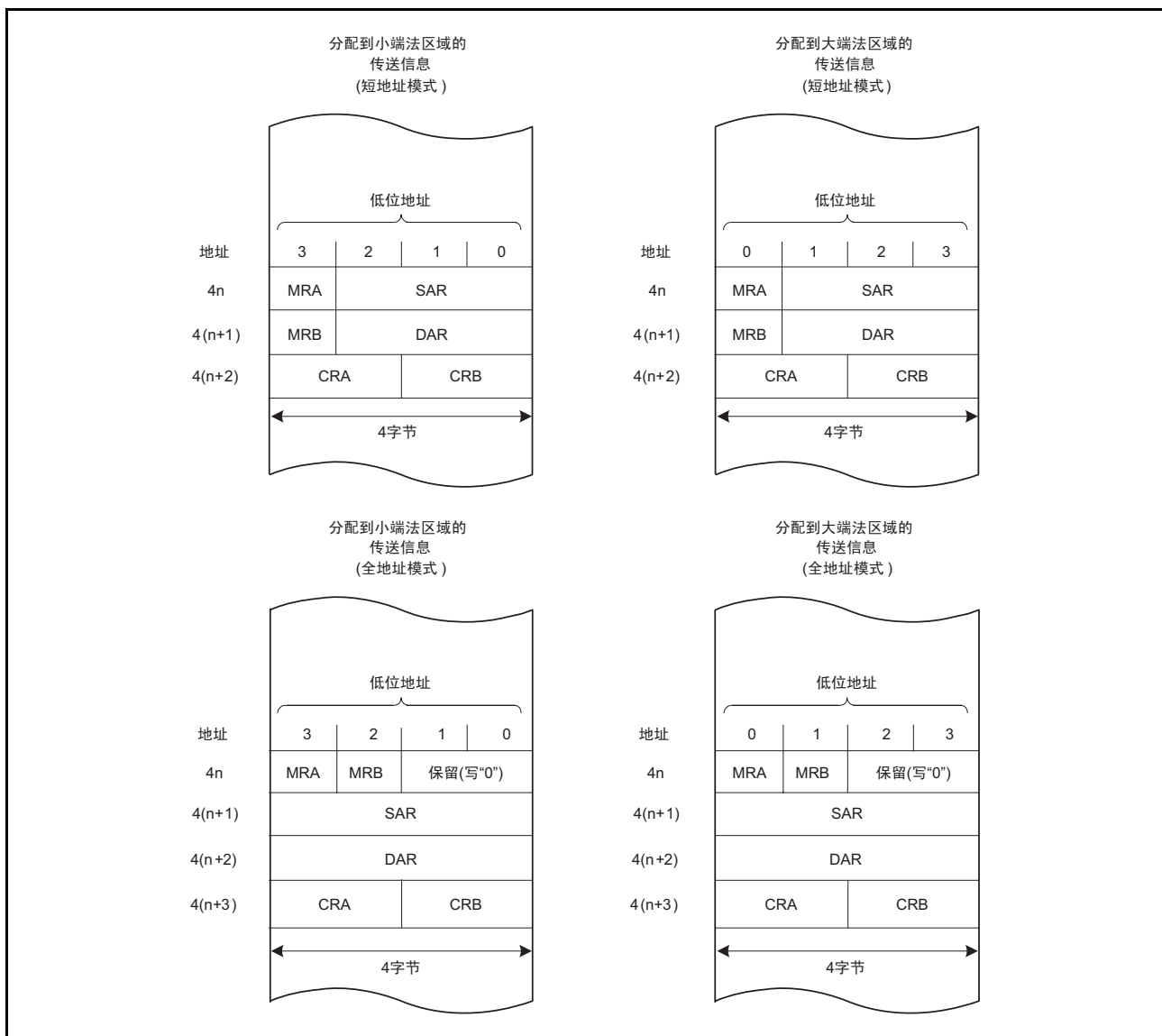


图 13.16 传送信息的分配

14. I/O 端口

RX610 群的 I/O 端口用作可编程输入 / 输出端口、外围功能的输入 / 输出、中断输入引脚或者总线控制引脚。

各端口与外围模块的输入 / 输出引脚和中断输入引脚兼用。在复位后立即变为输入端口，通过设定寄存器进行功能的转换。各端口的设定取决于 I/O 端口的寄存器和内部外围模块的寄存器的设定。

各端口有控制输入 / 输出的数据方向寄存器 (DDR)、保存输出数据的数据寄存器 (DR)、读引脚状态的端口寄存器 (PORT) 以及控制输入缓冲器有效或者无效的输入缓冲控制寄存器 (ICR)。

I/O 端口的结构因封装而不同。144 引脚 LQFP 由端口 0 ~ 9 和端口 A ~ E 共 15 个端口构成，有 117 个输入 / 输出引脚；176 引脚 LFBGA 由端口 0 ~ 9 和端口 A ~ H 共 18 个端口构成，有 140 个输入 / 输出引脚。

14.1 概要

I/O 端口的规格和端口功能一览表分别如表 14.1 和表 14.2 所示。

表 14.1 I/O 端口的规格

| 项目 | | 内容 |
|-----------|--------------|--|
| 输入 / 输出引脚 | 144 引脚 LQFP | 117 个 |
| | 176 引脚 LFBGA | 140 个 |
| 端口 | 144 引脚 LQFP | 15 个端口 (端口 0 ~ 9 和端口 A ~ E) |
| | 176 引脚 LFBGA | 18 个端口 (端口 0 ~ 9 和端口 A ~ H) |
| 内置输入上拉电阻 | | 端口 A,B,C,D,E |
| 漏极开路输出 | | 端口 2 和端口 C |
| 5V 容许 | | 端口 0 和端口 1 (P14、P15、P16、P17) |
| 施密特触发输入引脚 | | 全部端口输入、IRQ 输入、TPU 输入、TMR 输入、RIIC 输入、SCI 输入 |
| 其他 | | <ul style="list-style-type: none"> 能驱动 1 个 TTL 负载和 30pF 的电容负载。 在输出时驱动达林顿晶体管。 |

表 14.2 端口功能一览表 (1/4)

| 端口 | 概要 | 位 | 功能 | | | CMOS 输入引脚 | 施密特触发 输入引脚 | 输入上拉 电阻功能 | 漏极开路 输出功能 |
|------|---|---|---------------------|----------------------------|---------------|--------------|---------------|--------------|--------------|
| | | | 输入 / 输出 | 输入 | 输出 | | | | |
| 端口 0 | 与内部仿真器 输入、中断输 入、TMR 输 入 / 输出和 SCI 输入 / 输 出兼用的通用 输入 / 输出端 口 | 0 | P00 | TMRI2/IRQ8-A | TxD6 | — | 全部输入功能 | — | — |
| | | 1 | P01 | TMCI2/RxD6/ IRQ9-A | | | 全部输入功能 | | |
| | | 2 | P02/SCK6 | IRQ10-A/TRST# | TMO2 | | 全部输入功能 | | |
| | | 3 | P03/SCK4 | TMRI3/IRQ11-A/ TMS | | | 全部输入功能 | | |
| | | 4 | P04 | TMCI3/IRQ12-A/ TDI | TxD4 | | 全部输入功能 | | |
| | | 5 | P05 | RxD4/IRQ13-A/ TCK | TMO3 | | 全部输入功能 | | |
| 端口 1 | 与中断输入、 TPU 输入、 SCI 输入 / 输 出、RIIC 输入 / 输出和 A/D 转换器输入兼 用的通用输入 / 输出端口 | 0 | P10 | IRQ0-B | | — | 全部输入功能 | — | — |
| | | 1 | P11/SCK2 | IRQ1-B | | | 全部输入功能 | | |
| | | 2 | P12 | RxD2/IRQ2-B | | | 全部输入功能 | | |
| | | 3 | P13 | ADTRG0#/IRQ3-B | TxD2 | | 全部输入功能 | | |
| | | 4 | P14/SDA1 | TCLKA-B/IRQ4-B | | | 全部输入功能 | | |
| | | 5 | P15/SCK3/SCL1 | TCLKB-B/IRQ5-B | | | 全部输入功能 | | |
| | | 6 | P16/SDA0 | TCLKC-B/RxD3/ IRQ6-B | | | 全部输入功能 | | |
| | | 7 | P17/SCL0 | TCLKD-B/ ADTRG1#/IRQ7-B | TxD3 | | 全部输入功能 | | |
| 端口 2 | 与 TPU 输入 / 输出、PPG 输出、TMR 输入 / 输出和 SCI 输入 / 输 出兼用的通用 输入 / 输出端 口 | 0 | P20/TIOCB3 | TIOCA3/TMRI0 | PO0/TxD0 | — | 全部输入功能 | — | ○ |
| | | 1 | P21/TIOCA3 | TMCI0/RxD0 | PO1 | | 全部输入功能 | | |
| | | 2 | P22/TIOCC3/ SCK0 | | PO2/TMO0 | | 全部输入功能 | | |
| | | 3 | P23/TIOCD3 | TIOCC3 | PO3 | | 全部输入功能 | | |
| | | 4 | P24/TIOCB4 | TIOCA4/TMRI1 | PO4 | | 全部输入功能 | | |
| | | 5 | P25/TIOCA4 | TMCI1/RxD1 | PO5 | | 全部输入功能 | | |
| | | 6 | P26/TIOCA5 | | PO6/TMO1/TxD1 | | 全部输入功能 | | |
| | | 7 | P27/TIOCB5/ SCK1 | TIOCA5 | PO7 | | 全部输入功能 | | |
| 端口 3 | 与中断输入、 TPU 输入 / 输 出和 PPG 输 出兼用的通用 输入 / 输出端 口 | 0 | P30/TIOCA0 | IRQ0-A | PO8 | — | 全部输入功能 | — | — |
| | | 1 | P31/TIOCB0 | TIOCA0/IRQ1-A | PO9 | | 全部输入功能 | | |
| | | 2 | P32/TIOCC0 | TCLKA-A/IRQ2-A | PO10 | | 全部输入功能 | | |
| | | 3 | P33/TIOCD0 | TIOCC0/TCLKB-A/ IRQ3-A | PO11 | | 全部输入功能 | | |
| | | 4 | P34/TIOCA1 | IRQ4-A | PO12 | | 全部输入功能 | | |
| | | 5 | P35/TIOCB1 | TIOCA1/TCLKC-A | PO13 | | 全部输入功能 | | |
| | | 6 | P36/TIOCA2 | | PO14 | | 全部输入功能 | | |
| | | 7 | P37/TIOCB2 | TIOCA2/TCLKD-A | PO15 | | 全部输入功能 | | |
| 端口 4 | 与中断输入和 A/D 转换器输 入兼用的通用 输入 / 输出端 口 | 0 | P40 | AN0/IRQ8-B | | — | P40、IRQ8-B | — | — |
| | | 1 | P41 | AN1/IRQ9-B | | | P41、IRQ9-B | | |
| | | 2 | P42 | AN2/IRQ10-B | | | P42、IRQ10-B | | |
| | | 3 | P43 | AN3/IRQ11-B | | | P43、IRQ11-B | | |
| | | 4 | P44 | AN4/IRQ12-B | | | P44、IRQ12-B | | |
| | | 5 | P45 | AN5/IRQ13-B | | | P45、IRQ13-B | | |
| | | 6 | P46 | AN6/IRQ14-B | | | P46、IRQ14-B | | |
| | | 7 | P47 | AN7/IRQ15-B | | | P47、IRQ15-B | | |

表 14.2 端口功能一览表 (2/4)

| 端口 | 概要 | 位 | 功能 | | | CMOS 输入引脚 | 施密特触发 输入引脚 | 输入上拉 电阻功能 | 漏极开路 输出功能 |
|------|---|---|---------|---------|--|--------------|---------------|--------------|--------------|
| | | | 输入 / 输出 | 输入 | 输出 | | | | |
| 端口 5 | 与系统时钟输出、总线控制输入 / 输出和跟踪输入 / 输出兼用的通用输入 / 输出端口 | 0 | P50 | | WR0#/WR# | — | 全部输入功能 | — | — |
| | | 1 | P51 | | WR1#/BC1# | | 全部输入功能 | | |
| | | 2 | P52 | | RD# | | 全部输入功能 | | |
| | | 3 | | P53 | BCLK | | 全部输入功能 | | |
| | | 4 | P54 | | TRDATA0 | | 全部输入功能 | | |
| | | 5 | P55 | | TRDATA1 | | 全部输入功能 | | |
| | | 6 | P56 | | TRDATA2 | | 全部输入功能 | | |
| | | 7 | P57 | WAIT# | TRDATA3 | | 全部输入功能 | | |
| 端口 6 | 与中断输入、总线控制输出和 D/A 转换器输出兼用的通用输入 / 输出端口 | 0 | P60 | | CS0#/CS4#-A/ CS5#-B | — | 全部输入功能 | — | — |
| | | 1 | P61 | | CS1#/CS2#-B/ CS5#-A/CS6#-B/ CS7#-B | | 全部输入功能 | | |
| | | 2 | P62 | | CS2#-A/CS6#-A | | 全部输入功能 | | |
| | | 3 | P63 | | CS3#-A/CS7#-A | | 全部输入功能 | | |
| | | 4 | P64 | | CS4#-B | | 全部输入功能 | | |
| | | 5 | P65 | IRQ15-A | | | 全部输入功能 | | |
| | | 6 | P66 | | DA0 | | 全部输入功能 | | |
| | | 7 | P67 | | DA1 | | 全部输入功能 | | |
| 端口 7 | 与中断输入、总线控制输出和 A/D 转换器输入兼用的通用输入 / 输出端口 | 0 | P70 | ADTRG2# | CS3#-B | — | 全部输入功能 | — | — |
| | | 1 | P71 | | CS4#-C/CS5#-C/ CS6#-C/CS7#-C | | 全部输入功能 | | |
| | | 2 | P72 | | | | 全部输入功能 | | |
| | | 3 | P73 | | | | 全部输入功能 | | |
| | | 4 | P74 | ADTRG3# | | | 全部输入功能 | | |
| | | 5 | P75 | | | | 全部输入功能 | | |
| | | 6 | P76 | IRQ14-A | | | 全部输入功能 | | |
| | | 7 | P77 | | | | 全部输入功能 | | |
| 端口 8 | 与跟踪输出兼用的通用输入 / 输出端口 | 0 | P80 | | | — | 全部输入功能 | — | — |
| | | 1 | P81 | | TRSYNC | | 全部输入功能 | | |
| | | 2 | P82 | | TRCLK | | 全部输入功能 | | |
| | | 3 | P83 | | | | 全部输入功能 | | |
| | | 4 | P84 | | | | 全部输入功能 | | |
| | | 5 | P85 | | | | 全部输入功能 | | |
| | | 6 | P86 | | | | 全部输入功能 | | |
| 端口 9 | 与 A/D 转换器输入兼用的通用输入 / 输出端口 | 0 | P90 | AN8 | | — | P90 | — | — |
| | | 1 | P91 | AN9 | | | P91 | | |
| | | 2 | P92 | AN10 | | | P92 | | |
| | | 3 | P93 | AN11 | | | P93 | | |
| | | 4 | P94 | AN12 | | | P94 | | |
| | | 5 | P95 | AN13 | | | P95 | | |
| | | 6 | P96 | AN14 | | | P96 | | |
| | | 7 | P97 | AN15 | | | P97 | | |

表 14.2 端口功能一览表 (3/4)

| 端口 | 概要 | 位 | 功能 | | | CMOS 输入引脚 | 施密特触发 输入引脚 | 输入上拉 电阻功能 | 漏极开路 输出功能 |
|------|---|---|-------------|--------------|----------------------------|--------------|---------------|--------------|--------------|
| | | | 输入 / 输出 | 输入 | 输出 | | | | |
| 端口 A | 与地址输出、 TPU 输入 / 输出 和 PPG 输出 兼用的通用 输入 / 输出端 口 | 0 | PA0/TIOCA6 | | A0/PO16/BC0# | — | 全部输入功能 | ○ | — |
| | | 1 | PA1/TIOCB6 | TIOCA6 | A1/PO17 | | 全部输入功能 | | |
| | | 2 | PA2/TIOCC6 | TCLKE | A2/PO18 | | 全部输入功能 | | |
| | | 3 | PA3/TIOCD6 | TIOCC6/TCLKF | A3/PO19 | | 全部输入功能 | | |
| | | 4 | PA4/TIOCA7 | | A4/PO20 | | 全部输入功能 | | |
| | | 5 | PA5/TIOCB7 | TIOCA7/TCLKG | A5/PO21 | | 全部输入功能 | | |
| | | 6 | PA6/TIOCA8 | | A6/PO22 | | 全部输入功能 | | |
| | | 7 | PA7/TIOCB8 | TIOCA8/TCLKH | A7/PO23 | | 全部输入功能 | | |
| 端口 B | 与地址输出、 TPU 输入 / 输出 和 PPG 输出 兼用的通用 输入 / 输出端 口 | 0 | PB0/TIOCA9 | | A8/PO24 | — | 全部输入功能 | ○ | — |
| | | 1 | PB1/TIOCB9 | TIOCA9 | A9/PO25 | | 全部输入功能 | | |
| | | 2 | PB2/TIOCC9 | | A10/PO26 | | 全部输入功能 | | |
| | | 3 | PB3/TIOCD9 | TIOCC9 | A11/PO27 | | 全部输入功能 | | |
| | | 4 | PB4/TIOCA10 | | A12/PO28 | | 全部输入功能 | | |
| | | 5 | PB5/TIOCB10 | TIOCA10 | A13/PO29 | | 全部输入功能 | | |
| | | 6 | PB6/TIOCA11 | | A14/PO30 | | 全部输入功能 | | |
| | | 7 | PB7/TIOCB11 | TIOCA11 | A15/PO31 | | 全部输入功能 | | |
| 端口 C | 与地址输出、 总线控制输出 和 SCI 输入 / 输出兼用的通 用输入 / 输出 端口 | 0 | PC0 | | A16 | — | 全部输入功能 | ○ | ○ |
| | | 1 | PC1 | | A17 | | 全部输入功能 | | |
| | | 2 | PC2 | | A18 | | 全部输入功能 | | |
| | | 3 | PC3 | | A19 | | 全部输入功能 | | |
| | | 4 | PC4 | | A20 | | 全部输入功能 | | |
| | | 5 | PC5/SCK5 | | A21/CS5#-D | | 全部输入功能 | | |
| | | 6 | PC6 | RxD5 | A22/CS6#-D | | 全部输入功能 | | |
| | | 7 | PC7 | | A23/CS4#-D/ CS7#-D/TxD5 | | 全部输入功能 | | |
| 端口 D | 与双向数据总 线兼用的通用 输入 / 输出端 口 | 0 | PD0/D0 | | | D0 | PD0 | ○ | — |
| | | 1 | PD1/D1 | | | D1 | PD1 | | |
| | | 2 | PD2/D2 | | | D2 | PD2 | | |
| | | 3 | PD3/D3 | | | D3 | PD3 | | |
| | | 4 | PD4/D4 | | | D4 | PD4 | | |
| | | 5 | PD5/D5 | | | D5 | PD5 | | |
| | | 6 | PD6/D6 | | | D6 | PD6 | | |
| | | 7 | PD7/D7 | | | D7 | PD7 | | |
| 端口 E | 与双向数据总 线和中断输入 兼用的通用输 入 / 输出端口 | 0 | PE0/D8 | | | D8 | PE0 | ○ | — |
| | | 1 | PE1/D9 | | | D9 | PE1 | | |
| | | 2 | PE2/D10 | | | D10 | PE2 | | |
| | | 3 | PE3/D11 | | | D11 | PE3 | | |
| | | 4 | PE4/D12 | | | D12 | PE4 | | |
| | | 5 | PE5/D13 | IRQ5-A | | D13 | PE5、IRQ5-A | | |
| | | 6 | PE6/D14 | IRQ6-A | | D14 | PE6、IRQ6-A | | |
| | | 7 | PE7/D15 | IRQ7-A | | D15 | PE7、IRQ7-A | | |

表 14.2 端口功能一览表 (4/4)

| 端口 | 概要 | 位 | 功能 | | | CMOS 输入引脚 | 施密特触发 输入引脚 | 输入上拉 电阻功能 | 漏极开路 输出功能 |
|------|-------------|---|---------|----|----|--------------|---------------|--------------|--------------|
| | | | 输入 / 输出 | 输入 | 输出 | | | | |
| 端口 F | 通用输入 / 输出端口 | 0 | PF0 | | | — | 全部输入功能 | — | — |
| | | 1 | PF1 | | | | 全部输入功能 | | |
| | | 2 | PF2 | | | | 全部输入功能 | | |
| | | 3 | PF3 | | | | 全部输入功能 | | |
| | | 4 | PF4 | | | | 全部输入功能 | | |
| | | 5 | PF5 | | | | 全部输入功能 | | |
| | | 6 | PF6 | | | | 全部输入功能 | | |
| 端口 G | 通用输入 / 输出端口 | 0 | PG0 | | | — | 全部输入功能 | — | — |
| | | 1 | PG1 | | | | 全部输入功能 | | |
| | | 2 | PG2 | | | | 全部输入功能 | | |
| | | 3 | PG3 | | | | 全部输入功能 | | |
| | | 4 | PG4 | | | | 全部输入功能 | | |
| | | 5 | PG5 | | | | 全部输入功能 | | |
| | | 6 | PG6 | | | | 全部输入功能 | | |
| | | 7 | PG7 | | | | 全部输入功能 | | |
| 端口 H | 通用输入 / 输出端口 | 0 | PH0 | | | — | 全部输入功能 | — | — |
| | | 1 | PH1 | | | | 全部输入功能 | | |
| | | 2 | PH2 | | | | 全部输入功能 | | |
| | | 3 | PH3 | | | | 全部输入功能 | | |
| | | 4 | PH4 | | | | 全部输入功能 | | |
| | | 5 | PH5 | | | | 全部输入功能 | | |
| | | 6 | PH6 | | | | 全部输入功能 | | |
| | | 7 | PH7 | | | | 全部输入功能 | | |

14.2 寄存器说明

各端口的寄存器一览表如表 14.3 所示。

表 14.3 各端口的寄存器一览表 (1/3)

| 端口符号 | 寄存器名 | 寄存器符号 | 复位后的值 | 地址 | 存取长度 |
|------|-----------|-------|-------|------------|------|
| P0 | 数据方向寄存器 | DDR | 00h | 0008 C000h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C020h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C040h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C060h | 8 |
| P1 | 数据方向寄存器 | DDR | 00h | 0008 C001h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C021h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C041h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C061h | 8 |
| P2 | 数据方向寄存器 | DDR | 00h | 0008 C002h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C022h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C042h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C062h | 8 |
| | 漏极开路控制寄存器 | ODR | 00h | 0008 C082h | 8 |
| P3 | 数据方向寄存器 | DDR | 00h | 0008 C003h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C023h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C043h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C063h | 8 |
| P4 | 数据方向寄存器 | DDR | 00h | 0008 C004h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C024h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C044h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C064h | 8 |
| P5 | 数据方向寄存器 | DDR | 00h | 0008 C005h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C025h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C045h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C065h | 8 |
| P6 | 数据方向寄存器 | DDR | 00h | 0008 C006h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C026h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C046h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C066h | 8 |
| P7 | 数据方向寄存器 | DDR | 00h | 0008 C007h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C027h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C047h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C067h | 8 |
| P8 | 数据方向寄存器 | DDR | 00h | 0008 C008h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C028h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C048h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C068h | 8 |

表 14.3 各端口的寄存器一览表 (2/3)

| 端口符号 | 寄存器名 | 寄存器符号 | 复位后的值 | 地址 | 存取长度 |
|------|-----------|-------|-------|------------|------|
| P9 | 数据方向寄存器 | DDR | 00h | 0008 C009h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C029h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C049h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C069h | 8 |
| PA | 数据方向寄存器 | DDR | 00h | 0008 C00Ah | 8 |
| | 数据寄存器 | DR | 00h | 0008 C02Ah | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C04Ah | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C06Ah | 8 |
| | 上拉电阻控制寄存器 | PCR | 00h | 0008 C0CAh | 8 |
| PB | 数据方向寄存器 | DDR | 00h | 0008 C00Bh | 8 |
| | 数据寄存器 | DR | 00h | 0008 C02Bh | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C04Bh | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C06Bh | 8 |
| | 上拉电阻控制寄存器 | PCR | 00h | 0008 C0CBh | 8 |
| PC | 数据方向寄存器 | DDR | 00h | 0008 C00Ch | 8 |
| | 数据寄存器 | DR | 00h | 0008 C02Ch | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C04Ch | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C06Ch | 8 |
| | 漏极开路控制寄存器 | ODR | 00h | 0008 C08Ch | 8 |
| | 上拉电阻控制寄存器 | PCR | 00h | 0008 C0CCh | 8 |
| PD | 数据方向寄存器 | DDR | 00h | 0008 C00Dh | 8 |
| | 数据寄存器 | DR | 00h | 0008 C02Dh | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C04Dh | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C06Dh | 8 |
| | 上拉电阻控制寄存器 | PCR | 00h | 0008 C0CDh | 8 |
| PE | 数据方向寄存器 | DDR | 00h | 0008 C00Eh | 8 |
| | 数据寄存器 | DR | 00h | 0008 C02Eh | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C04Eh | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C06Eh | 8 |
| | 上拉电阻控制寄存器 | PCR | 00h | 0008 C0CEh | 8 |
| PF | 数据方向寄存器 | DDR | 00h | 0008 C00Fh | 8 |
| | 数据寄存器 | DR | 00h | 0008 C02Fh | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C04Fh | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C06Fh | 8 |
| PG | 数据方向寄存器 | DDR | 00h | 0008 C010h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C030h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C050h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C070h | 8 |

表 14.3 各端口的寄存器一览表 (3/3)

| 端口符号 | 寄存器名 | 寄存器符号 | 复位后的值 | 地址 | 存取长度 |
|------|-------------|-------|-------|------------|------|
| PH | 数据方向寄存器 | DDR | 00h | 0008 C011h | 8 |
| | 数据寄存器 | DR | 00h | 0008 C031h | 8 |
| | 端口寄存器 | PORT | 不定值 | 0008 C051h | 8 |
| | 输入缓冲控制寄存器 | ICR | 00h | 0008 C071h | 8 |
| 通用 | 端口功能控制寄存器 0 | PFCR0 | 00h | 0008 C100h | 8 |
| | 端口功能控制寄存器 1 | PFCR1 | 00h | 0008 C101h | 8 |
| | 端口功能控制寄存器 2 | PFCR2 | 00h | 0008 C102h | 8 |
| | 端口功能控制寄存器 3 | PFCR3 | 00h | 0008 C103h | 8 |
| | 端口功能控制寄存器 4 | PFCR4 | 00h | 0008 C104h | 8 |
| | 端口功能控制寄存器 5 | PFCR5 | 00h | 0008 C105h | 8 |
| | 端口功能控制寄存器 6 | PFCR6 | 00h | 0008 C106h | 8 |
| | 端口功能控制寄存器 7 | PFCR7 | 00h | 0008 C107h | 8 |
| | 端口功能控制寄存器 8 | PFCR8 | 00h | 0008 C108h | 8 |
| | 端口功能控制寄存器 9 | PFCR9 | 00h | 0008 C109h | 8 |

14.2.1 数据方向寄存器 (DDR)

地址 P0.DDR 0008 C000h、P1.DDR 0008 C001h、P2.DDR 0008 C002h、P3.DDR 0008 C003h、
P4.DDR 0008 C004h、P5.DDR0008 C005h、P6.DDR 0008 C006h、P7.DDR 0008 C007h、
P8.DDR 0008 C008h、P9.DDR 0008 C009h、PA.DDR 0008 C00Ah、PB.DDR 0008 C00Bh、
PC.DDR 0008 C00Ch、PD.DDR 0008 C00Dh、PE.DDR 0008 C00Eh、PF.DDR 0008 C00Fh、
PG.DDR 0008 C010h、PH.DDR 0008 C011h

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

复位后的值

注1. P0.DDR寄存器的低6位有效，高2位是保留位。
P8.DDR寄存器的低7位有效，高1位是保留位。
PF.DDR寄存器的低7位有效，高1位是保留位。
保留位的读写值都为“0”。

| 位 | 符号 | 位名 | 功能 | R/W |
|----|----|-----------------------------------|--------------------|-----|
| b0 | B0 | Pm0 输入 / 输出指定位 (m=0 ~ 9,A ~ H) | 0: 输入端口 1: 输出端口 | R/W |
| b1 | B1 | Pm1 输入 / 输出指定位 | | R/W |
| b2 | B2 | Pm2 输入 / 输出指定位 | | R/W |
| b3 | B3 | Pm3 输入 / 输出指定位 | | R/W |
| b4 | B4 | Pm4 输入 / 输出指定位 | | R/W |
| b5 | B5 | Pm5 输入 / 输出指定位 | | R/W |
| b6 | B6 | Pm6 输入 / 输出指定位 | | R/W |
| b7 | B7 | Pm7 输入 / 输出指定位 | | R/W |

DDR 寄存器是选择了通用输入 / 输出端口功能时的指定端口输入 / 输出的寄存器。

Pm.DDR 寄存器 (m=0 ~ 9,A ~ H) 的各位分别对应端口 m 的各引脚，并且能以 1 位为单位进行指定。

14.2.2 数据寄存器 (DR)

地址 P0.DR 0008 C020h、P1.DR 0008 C021h、P2.DR 0008 C022h、P3.DR 0008 C023h、
P4.DR 0008 C024h、P5.DR 0008 C025h、P6.DR 0008 C026h、P7.DR 0008 C027h、
P8.DR 0008 C028h、P9.DR 0008 C029h、PA.DR 0008 C02Ah、PB.DR 0008 C02Bh、
PC.DR 0008 C02Ch、PD.DR 0008 C02Dh、PE.DR 0008 C02Eh、PF.DR 0008 C02Fh、
PG.DR 0008 C030h、PH.DR 0008 C031h

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

复位后的值

- 注1. P0.DR寄存器的低6位有效，高2位是保留位。
P5.DR.B3位是保留位。
P8.DR寄存器的低7位有效，高1位是保留位。
PF.DR寄存器的低7位有效，高1位是保留位。
P5.DR.B3位以外的保留位的读写值都为“0”。
P5.DR.B3位是可读写位。

| 位 | 符号 | 位名 | 功能 | R/W |
|----|----|--------------------------------|----------|-----|
| b0 | B0 | Pm0 输出数据保存位 (m=0 ~ 9,A ~ H) | 保存输出的数据。 | R/W |
| b1 | B1 | Pm1 输出数据保存位 | | R/W |
| b2 | B2 | Pm2 输出数据保存位 | | R/W |
| b3 | B3 | Pm3 输出数据保存位 | | R/W |
| b4 | B4 | Pm4 输出数据保存位 | | R/W |
| b5 | B5 | Pm5 输出数据保存位 | | R/W |
| b6 | B6 | Pm6 输出数据保存位 | | R/W |
| b7 | B7 | Pm7 输出数据保存位 | | R/W |

DR 寄存器保存用作通用输出端口的引脚输出数据。

P53 的输出为 BCLK，即使给 P5.DR.B3 位设定值，也不影响引脚。

14.2.3 端口寄存器 (PORT)

地址 P0.PORT 0008 C040h、P1.PORT 0008 C041h、P2.PORT 0008 C042h、P3.PORT 0008 C043h、
P4.PORT 0008 C044h、P5.PORT 0008 C045h、P6.PORT 0008 C046h、P7.PORT 0008 C047h、
P8.PORT 0008 C048h、P9.PORT 0008 C049h、PA.PORT 0008 C04Ah、PB.PORT 0008 C04Bh、
PC.PORT 0008 C04Ch、PD.PORT 0008 C04Dh、PE.PORT 0008 C04Eh、PF.PORT 0008 C04Fh、
PG.PORT 0008 C050h、PH.PORT 0008 C051h

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

复位后的值 x x x x x x x x

注1. P0.PORT寄存器的低6位有效，高2位是保留位。
P8.PORT寄存器的低7位有效，高1位是保留位。
PF.PORT寄存器的低7位有效，高1位是保留位。
保留位的读写值都为“0”。

| 位 | 符号 | 位名 | 功能 | R/W |
|----|----|-----------------------|------------|-----|
| b0 | B0 | Pm0 位 (m=0 ~ 9,A ~ H) | 反映端口的引脚状态。 | R |
| b1 | B1 | Pm1 位 | | R |
| b2 | B2 | Pm2 位 | | R |
| b3 | B3 | Pm3 位 | | R |
| b4 | B4 | Pm4 位 | | R |
| b5 | B5 | Pm5 位 | | R |
| b6 | B6 | Pm6 位 | | R |
| b7 | B7 | Pm7 位 | | R |

PORT 寄存器是反映端口引脚状态的寄存器。

如果读 Pm.PORT 寄存器 (m=0 ~ 9,A ~ H)，Pm.DDR 寄存器为“1”（输出端口）的位就能读 Pm.DR 寄存器的值。Pm.DDR 寄存器为“0”（输入端口）的位读引脚的状态，与 Pm.ICR 寄存器的值无关，

14.2.4 输入缓冲控制寄存器 (ICR)

地址 P0.ICR 0008 C060h、P1.ICR 0008 C061h、P2.ICR 0008 C062h、P3.ICR 0008 C063h、
P4.ICR 0008 C064h、P5.ICR 0008 C065h、P6.ICR 0008 C066h、P7.ICR 0008 C067h、
P8.ICR 0008 C068h、P9.ICR 0008 C069h、PA.ICR 0008 C06Ah、PB.ICR 0008 C06Bh、
PC.ICR 0008 C06Ch、PD.ICR 0008 C06Dh、PE.ICR 0008 C06Eh、PF.ICR 0008 C06Fh、
PG.ICR 0008 C070h、PH.ICR 0008 C071h

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

复位后的值

注1. P0.ICR寄存器的低6位有效，高2位是保留位。
P8.ICR寄存器的低7位有效，高1位是保留位。
PF.ICR寄存器的低7位有效，高1位是保留位。
保留位的读写值都为“0”。

| 位 | 符号 | 位名 | 功能 | R/W |
|----|----|---------------------------------|--|-----|
| b0 | B0 | Pm0 输入缓冲器控制位 (n=0 ~ 9,A ~ H) | 0: 对应引脚的输入缓冲器无效，将输入信号固定为 High 电平。 | R/W |
| b1 | B1 | Pm1 输入缓冲器控制位 | 1: 对应引脚的输入缓冲器有效 | R/W |
| b2 | B2 | Pm2 输入缓冲器控制位 | 在用作外围模块的输入引脚时，必须将对应位置“1”。 必须将模拟输入 / 输出引脚和未用作输入引脚的对应位置“0”。 | R/W |
| b3 | B3 | Pm3 输入缓冲器控制位 | | R/W |
| b4 | B4 | Pm4 输入缓冲器控制位 | | R/W |
| b5 | B5 | Pm5 输入缓冲器控制位 | | R/W |
| b6 | B6 | Pm6 输入缓冲器控制位 | | R/W |
| b7 | B7 | Pm7 输入缓冲器控制位 | | R/W |

ICR 寄存器是控制端口输入缓冲器的寄存器。

Pm.ICR 寄存器 (m=0 ~ 9,A ~ H) 的各位分别对应端口 m 的各引脚，并且能以 1 位为单位进行指定。

如果读 Pm.PORT 寄存器，就能读引脚的状态，与 Pm.ICR 寄存器的值无关。此时，Pm.ICR 寄存器为“0”的位，不将输入信号传送到对应的外围模块。

在更改 Pm.ICR 寄存器的设定时，有可能因引脚的状态而在内部产生边沿，因此必须在不使用该输入引脚时更改 Pm.ICR 寄存器的设定。例如，在 IRQn (n=0 ~ 15) 输入时，必须在禁止该中断的状态下更改 Pm.ICR 寄存器的设定。先将中断控制器的 IRi.IR 标志 (i=64 ~ 79 (IRQ 的中断向量号)) 置“0”，然后允许该中断。如果在更改 Pm.ICR 寄存器的设定后产生边沿，就必须取消该边沿。

14.2.5 上拉电阻控制寄存器 (PCR)

地址 PA.PCR 0008 C0CAh、PB.PCR 0008 C0CBh、PC.PCR 0008 C0CCh、
PD.PCR 0008 C0CDh、PE.PCR 0008 C0CEh

| | | | | | | | |
|-------|----|----|----|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| 复位后的值 | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|----|----------------------------|----------------------------|-----|
| b0 | B0 | Pm0 输入上拉电阻控制位 (n=A ~ E) | 0: 输入上拉电阻无效 1: 输入上拉电阻有效 | R/W |
| b1 | B1 | Pm1 输入上拉电阻控制位 | | R/W |
| b2 | B2 | Pm2 输入上拉电阻控制位 | | R/W |
| b3 | B3 | Pm3 输入上拉电阻控制位 | | R/W |
| b4 | B4 | Pm4 输入上拉电阻控制位 | | R/W |
| b5 | B5 | Pm5 输入上拉电阻控制位 | | R/W |
| b6 | B6 | Pm6 输入上拉电阻控制位 | | R/W |
| b7 | B7 | Pm7 输入上拉电阻控制位 | | R/W |

PCR 寄存器是控制端口输入上拉电阻的有效或者无效的寄存器。

在引脚为输入的状态下，与 Pm.PCR 寄存器为“1”的位对应的引脚输入上拉电阻有效。输入上拉电阻的状态如表 14.4 所示。

表 14.4 输入上拉电阻的状态

| 端口 | 引脚状态 | 复位中 | 运行中 |
|------|-------------|-----|---------|
| 端口 A | 地址输出 | 无效 | |
| | 外围模块输出 | 无效 | |
| | 端口输出 | 无效 | |
| | 端口输入、外围模块输入 | 无效 | 有效 / 无效 |
| 端口 B | 地址输出 | 无效 | |
| | 外围模块输出 | 无效 | |
| | 端口输出 | 无效 | |
| | 端口输入、外围模块输入 | 无效 | 有效 / 无效 |
| 端口 C | 地址输出 | 无效 | |
| | 外围模块输出 | 无效 | |
| | 端口输出 | 无效 | |
| | 端口输入、外围模块输入 | 无效 | 有效 / 无效 |
| 端口 D | 数据输入 / 输出 | 无效 | |
| | 端口输出 | 无效 | |
| | 端口输入 | 无效 | 有效 / 无效 |
| 端口 E | 数据输入 / 输出 | 无效 | |
| | 端口输出 | 无效 | |
| | 端口输入、外围模块输入 | 无效 | 有效 / 无效 |

无效 : 输入上拉电阻 MOS 总是无效。

有效 / 无效 : 将 Pm.PCR.Bj 位 (m=A ~ E, j=0 ~ 7) 置“1”时有效, 否则无效。

14.2.6 漏极开路控制寄存器 (ODR)

地址 P2.ODR 0008 C082h、PC.ODR 0008 C08Ch

| | | | | | | | |
|-------|----|----|----|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|----|---------------------|------------------------------|-----|
| b0 | B0 | Pm0 输出形态指定位 (n=2,C) | 0: CMOS 输出 1: NMOS 漏极开路输出 | R/W |
| b1 | B1 | Pm1 输出形态指定位 | | R/W |
| b2 | B2 | Pm2 输出形态指定位 | | R/W |
| b3 | B3 | Pm3 输出形态指定位 | | R/W |
| b4 | B4 | Pm4 输出形态指定位 | | R/W |
| b5 | B5 | Pm5 输出形态指定位 | | R/W |
| b6 | B6 | Pm6 输出形态指定位 | | R/W |
| b7 | B7 | Pm7 输出形态指定位 | | R/W |

ODR 寄存器是选择引脚输出形态的寄存器。

14.2.7 端口功能控制寄存器 0 (PFCR0)

地址 0008 C100h

| | | | | | | | |
|-------|------|------|------|------|------|------|------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| CS7E | CS6E | CS5E | CS4E | CS3E | CS2E | CS1E | CS0E |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|------|---------|---|-----|
| b0 | CS0E | CS0 允许位 | 0: 用作 I/O 端口 1: 用作 CSn# 输出引脚 (n=0 ~ 7) | R/W |
| b1 | CS1E | CS1 允许位 | | R/W |
| b2 | CS2E | CS2 允许位 | | R/W |
| b3 | CS3E | CS3 允许位 | | R/W |
| b4 | CS4E | CS4 允许位 | | R/W |
| b5 | CS5E | CS5 允许位 | | R/W |
| b6 | CS6E | CS6 允许位 | | R/W |
| b7 | CS7E | CS7 允许位 | | R/W |

PFCR0 寄存器是选择允许或者禁止 CSn# 输出的寄存器。

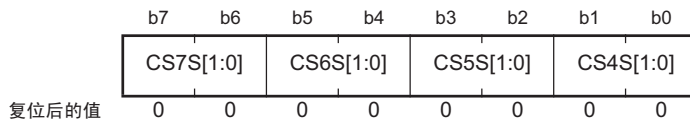
CSnE 位 (CSn 允许位) (n=0 ~ 7)

此位选择允许或者禁止对应的 CSn# 输出。

在输出 CSn 时, 必须将对应的 PFCR0.CSnE 位置 “1”。

14.2.8 端口功能控制寄存器 1 (PFCR1)

地址 0008 C101h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-----------|--------------|---|-----|
| b1-b0 | CS4S[1:0] | CS4# 输出引脚选择位 | b1 b0 0 0: 从 P60 输出 CS4#-A 0 1: 从 P64 输出 CS4#-B 1 0: 从 P71 输出 CS4#-C 1 1: 从 PC7 输出 CS4#-D | R/W |
| b3-b2 | CS5S[1:0] | CS5# 输出引脚选择位 | b3 b2 0 0: 从 P61 输出 CS5#-A 0 1: 从 P60 输出 CS5#-B 1 0: 从 P71 输出 CS5#-C 1 1: 从 PC5 输出 CS5#-D | R/W |
| b5-b4 | CS6S[1:0] | CS6# 输出引脚选择位 | b5 b4 0 0: 从 P62 输出 CS6#-A 0 1: 从 P61 输出 CS6#-B 1 0: 从 P71 输出 CS6#-C 1 1: 从 PC6 输出 CS6#-D | R/W |
| b7-b6 | CS7S[1:0] | CS7# 输出引脚选择位 | b7 b6 0 0: 从 P63 输出 CS7#-A 0 1: 从 P61 输出 CS7#-B 1 0: 从 P71 输出 CS7#-C 1 1: 从 PC7 输出 CS7#-D | R/W |

PFCR1 寄存器是选择 CSn# 输出引脚 (n=4 ~ 7) 的寄存器。

能通过 PFCR1 寄存器给 1 个引脚设定多个 CS 输出。如果进行这样的设定, 就输出所指定的全部 CS 信号。此时, 必须将与输出到相同引脚的 CS 对应的外部总线接口进行相同的设定。

CSnS[1:0] 位 (CSn# 输出引脚选择位) (n=4 ~ 7)

在允许 CSn# 输出时 (PFCR0.CSiE 位为 “1”), 这些位选择 CSn# 的输出引脚。

将 CS5 空间和 CS6 空间的 CSn# 信号输出到相同引脚时的时序如图 14.1 所示，CS 输出引脚选择寄存器和输出引脚的关系如表 14.5 所示。

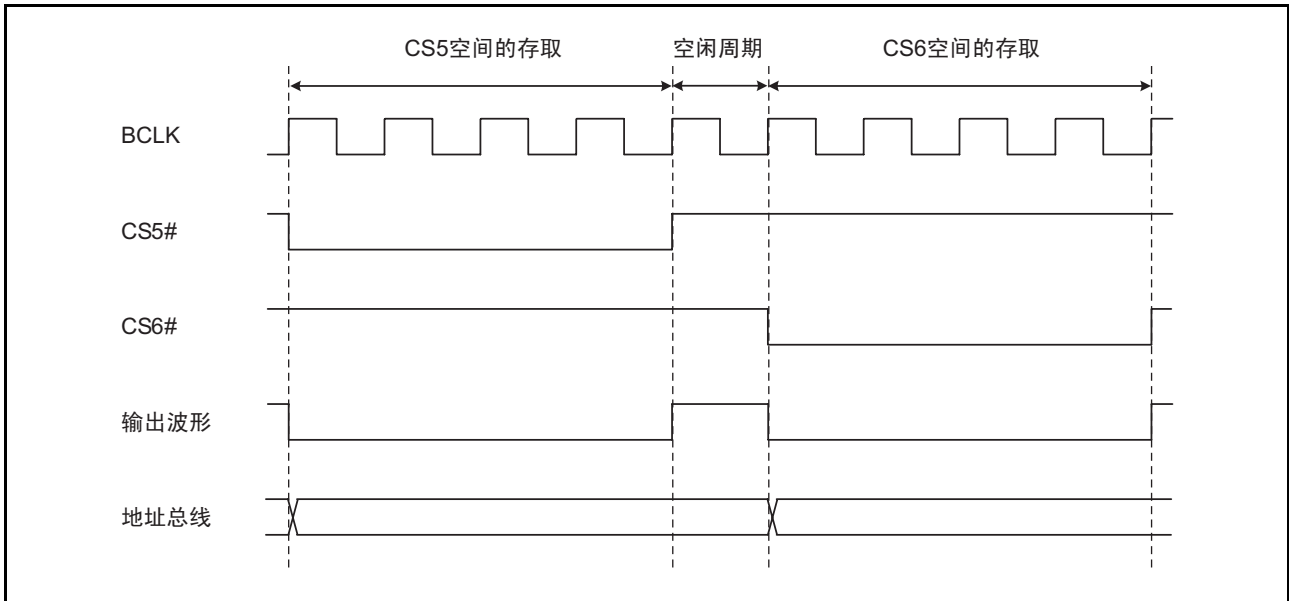


图 14.1 将 CSn# 输出到相同引脚时的时序

表 14.5 CSn# 输出引脚选择寄存器和输出引脚的关系

| 输出选择 | CS0# | CS1# | CS2# | CS3# | CS4# | CS5# | CS6# | CS7# |
|------|------|------|-------------|-------------|------------------|------------------|------------------|------------------|
| | — | — | PF2CR2.CS2S | PF2CR2.CS3S | PF1CR1.CS4S[1:0] | PF1CR1.CS5S[1:0] | PF1CR1.CS6S[1:0] | PF1CR1.CS7S[1:0] |
| P60 | CS0# | | | | CS4#-A | CS5#-B | | |
| P61 | | CS1# | CS2#-B | | | CS5#-A | CS6#-B | CS7#-B |
| P62 | | | CS2#-A | | | | CS6#-A | |
| P63 | | | | CS3#-A | | | | CS7#-A |
| P64 | | | | | CS4#-B | | | |
| P70 | | | | CS3#-B | | | | |
| P71 | | | | | CS4#-C | CS5#-C | CS6#-C | CS7#-C |
| PC5 | | | | | | CS5#-D | | |
| PC6 | | | | | | | CS6#-D | |
| PC7 | | | | | CS4#-D | | | CS7#-D |

14.2.9 端口功能控制寄存器 2 (PFCR2)

地址 0008 C102h

| | | | | | | | | |
|-------|------|------|----|----|----|----|----|----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | CS3S | CS2S | — | — | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|--------------|--|-----|
| b5-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b6 | CS2S | CS2# 输出引脚选择位 | 0: 从 P62 输出 CS2#-A 1: 从 P61 输出 CS2#-B | R/W |
| b7 | CS3S | CS3# 输出引脚选择位 | 0: 从 P63 输出 CS3#-A 1: 从 P70 输出 CS3#-B | R/W |

PFCR2 寄存器是选择 CSn# 输出引脚 (n=2,3) 的寄存器。

CSnS 位 (CSn# 输出引脚选择位) (n=2,3)

在允许 CSn# 输出时 (PFCR0.CSiE 位为“1”), 此位选择 CSn# 的输出引脚。

如果通过 CSn# 输出引脚选择位 (n=2,3) 给相同的引脚设定多个 CS# 输出, 就从该引脚输出多个 CS#。详细内容请参照“14.2.8 端口功能控制寄存器 1 (PFCR1)”。

14.2.10 端口功能控制寄存器 3 (PFCR3)

地址 0008 C103h

| | | | | | | | |
|------|------|------|------|------|------|------|------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| A23E | A22E | A21E | A20E | A19E | A18E | A17E | A16E |

复位后的值 0 0 0 0 0 0 0 0

| 位 | 符号 | 位名 | 功能 | R/W |
|----|------|------------|----------------------------|-----|
| b0 | A16E | 地址 A16 允许位 | 0: A16 输出无效 1: A16 输出有效 | R/W |
| b1 | A17E | 地址 A17 允许位 | 0: A17 输出无效 1: A17 输出有效 | R/W |
| b2 | A18E | 地址 A18 允许位 | 0: A18 输出无效 1: A18 输出有效 | R/W |
| b3 | A19E | 地址 A19 允许位 | 0: A19 输出无效 1: A19 输出有效 | R/W |
| b4 | A20E | 地址 A20 允许位 | 0: A20 输出无效 1: A20 输出有效 | R/W |
| b5 | A21E | 地址 A21 允许位 | 0: A21 输出无效 1: A21 输出有效 | R/W |
| b6 | A22E | 地址 A22 允许位 | 0: A22 输出无效 1: A22 输出有效 | R/W |
| b7 | A23E | 地址 A23 允许位 | 0: A23 输出无效 1: A23 输出有效 | R/W |

PFCR3 寄存器是选择允许或者禁止地址输出的寄存器。

AnE 位 (地址 An 允许位) (n=16 ~ 23)

此位选择允许或者禁止地址输出 (An)。

14.2.11 端口功能控制寄存器 4 (PFCR4)

地址 0008 C104h

| | | | | | | | | |
|-------|------|------|------|------|------|------|-----|-----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | A15E | A14E | A13E | A12E | A11E | A10E | A9E | A8E |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|------|------------|----------------------------|-----|
| b0 | A8E | 地址 A8 允许位 | 0: A8 输出无效 1: A8 输出有效 | R/W |
| b1 | A9E | 地址 A9 允许位 | 0: A9 输出无效 1: A9 输出有效 | R/W |
| b2 | A10E | 地址 A10 允许位 | 0: A10 输出无效 1: A10 输出有效 | R/W |
| b3 | A11E | 地址 A11 允许位 | 0: A11 输出无效 1: A11 输出有效 | R/W |
| b4 | A12E | 地址 A12 允许位 | 0: A12 输出无效 1: A12 输出有效 | R/W |
| b5 | A13E | 地址 A13 允许位 | 0: A13 输出无效 1: A13 输出有效 | R/W |
| b6 | A14E | 地址 A14 允许位 | 0: A14 输出无效 1: A14 输出有效 | R/W |
| b7 | A15E | 地址 A15 允许位 | 0: A15 输出无效 1: A15 输出有效 | R/W |

PFCR4 寄存器是选择允许或者禁止地址输出的寄存器。

AnE 位 (地址 An 允许位) (n=8 ~ 15)

此位选择允许或者禁止地址输出 (An)。

14.2.12 端口功能控制寄存器 5 (PFCR5)

地址 0008 C105h

| | | | | | | | |
|----|---------|----|-----|-------|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | WR1BC1E | — | DHE | TCLKS | — | — | — |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|-----------------|--|-----|
| b2-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b3 | TCLKS | TPU 外部时钟输入引脚选择位 | 0: 将 P32、P33、P35、P37 设定为外部时钟输入引脚 1: 将 P14 ~ P17 设定为外部时钟输入引脚 | R/W |
| b4 | DHE | 数据 D15 ~ D8 允许位 | 0: 将 PE7 ~ PE0 设定为 I/O 端口 1: 将 PE7 ~ PE0 设定为外部数据总线 D15 ~ D8 | R/W |
| b5 | — | 保留位 | 读写值都为“0”。 | R/W |
| b6 | WR1BC1E | WR1#/BC1# 输出允许位 | 0: 将 P51 设定为 I/O 端口 1: 将 P51 设定为 WR#1 或者 BC1# | R/W |
| b7 | — | 保留位 | 读写值都为“0”。 | R/W |

PFCR5 寄存器是选择 TPU 时钟输入引脚的寄存器。

TCLKS 位 (TPU 外部时钟输入引脚选择位)

此位选择 TPU 的外部时钟的输入引脚。

DHE 位 (数据 D15 ~ D8 允许位)

在内部 ROM 无效 / 有效扩展模式中，此位选择允许或者禁止数据输入 / 输出 (D15 ~ D8)。

注 1. 此设定必须符合 CSi 控制寄存器的外部总线宽度选择位 (CSiCNT.BSIZE[1:0] 位) 设定的外部总线宽度。如果在设定为外部 16 位总线的状态下将 DHE 位置“0”，就有可能阻碍端口 E 的运行。有关 CSiCNT.BSIZE[1:0] 位，请参照“11.3.1 CSi 控制寄存器 (CSiCNT) (i=0 ~ 7)”。

WR1BC1E 位 (WR1#/BC1# 输出允许位)

在内部 ROM 无效 / 有效扩展模式中，此位选择允许或者禁止 WR1#/BC1# 输出。

14.2.13 端口功能控制寄存器 6 (PFCR6)

地址 0008 C106h

| | | | | | | | | |
|-------|--------|--------|---------|---------|--------|--------|---------|---------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | TPUMS5 | TPUMS4 | TPUMS3A | TPUMS3B | TPUMS2 | TPUMS1 | TPUMS0A | TPUMS0B |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|---------|--------------------------|--|-----|
| b0 | TPUMS0B | TPU 输入 / 输出引脚的多功能选择 0B 位 | 0: 输出比较的输出和输入捕捉的输入为 P32。 1: 输入捕捉的输入为 P33, 输出比较的输出为 P32。 | R/W |
| b1 | TPUMS0A | TPU 输入 / 输出引脚的多功能选择 0A 位 | 0: 输出比较的输出和输入捕捉的输入为 P30。 1: 输入捕捉的输入为 P31, 输出比较的输出为 P30。 | R/W |
| b2 | TPUMS1 | TPU 输入 / 输出引脚的多功能选择 1 位 | 0: 输出比较的输出和输入捕捉的输入为 P34。 1: 输入捕捉输入为 P35, 输出比较的输出为 P34。 | R/W |
| b3 | TPUMS2 | TPU 输入 / 输出引脚的多功能选择 2 位 | 0: 输出比较的输出和输入捕捉的输入为 P36。 1: 输入捕捉的输入为 P37, 输出比较的输出为 P36。 | R/W |
| b4 | TPUMS3B | TPU 输入 / 输出引脚的多功能选择 3B 位 | 0: 输出比较的输出和输入捕捉的输入为 P22。 1: 输入捕捉的输入为 P23, 输出比较的输出为 P22。 | R/W |
| b5 | TPUMS3A | TPU 输入 / 输出引脚的多功能选择 3A 位 | 0: 输出比较的输出和输入捕捉的输入为 P21。 1: 输入捕捉的输入为 P20, 输出比较的输出为 P21。 | R/W |
| b6 | TPUMS4 | TPU 输入 / 输出引脚的多功能选择 4 位 | 0: 输出比较的输出和输入捕捉的输入为 P25。 1: 输入捕捉的输入为 P24, 输出比较的输出为 P25。 | R/W |
| b7 | TPUMS5 | TPU 输入 / 输出引脚的多功能选择 5 位 | 0: 输出比较的输出和输入捕捉的输入为 P26。 1: 输入捕捉的输入为 P27, 输出比较的输出为 P26。 | R/W |

PFCR6 寄存器是选择 TPU (单元 0) 输入 / 输出引脚多功能的寄存器。

能通过将 TPUMS0A 位、TPUMS1 位、TPUMS2 位、TPUMS3A 位、TPUMS4 位和 TPUMS5 位置“1”，将 TPUM 的 TGRA 和 TGRB 的输入捕捉的输入分配到相同的引脚，并且能通过将 TPUMS0B 位和 TPUMS3B 位置“1”，将 TPUM 的 TGRC 和 TGRD 的输入捕捉的输入分配到相同的引脚。

也能通过设定 TPU 的定时器模式寄存器 (TPUm.TMDR)，将 TGRA 和 TGRB、TGRC 和 TGRD 的输入捕捉的输入设定到相同的引脚。

PFCR6 寄存器和 TPUm.TMDR 寄存器的设定值与输入捕捉的输入和外部引脚的关系如表 14.6 所示。

TPUMS0B 位 (TPU 输入 / 输出引脚的多功能选择 0B 位)

此位选择 TIOCC0 的输入引脚。

TPUMS0A 位 (TPU 输入 / 输出引脚的多功能选择 0A 位)

此位选择 TIOCA0 的输入引脚。

TPUMS1 位 (TPU 输入 / 输出引脚的多功能选择 1 位)

此位选择 TIOCA1 的输入引脚。

TPUMS2 位 (TPU 输入 / 输出引脚的多功能选择 2 位)

此位选择 TIOCA2 的输入引脚。

TPUMS3B 位 (TPU 输入 / 输出引脚的多功能选择 3B 位)

此位选择 TIOCC3 的输入引脚。

TPUMS3A 位 (TPU 输入 / 输出引脚的多功能选择 3A 位)
此位选择 TIOCA3 的输入引脚。

TPUMS4 位 (TPU 输入 / 输出引脚的多功能选择 4 位)
此位选择 TIOCA4 的输入引脚。

TPUMS5 位 (TPU 输入 / 输出引脚的多功能选择 5 位)
此位选择 TIOCA5 的输入引脚。

表 14.6 PFCR6 寄存器和 TPUm 的定时器模式寄存器设定值与输入捕捉的输入和外部引脚的关系

| TPU0.TMDR.ICSELD | PFCR6.TPUMS0B | TPU0.TGRC | | TPU0.TGRD | |
|------------------|---------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCC0 | P32 | TIOCD0 | P33 |
| 0 | 1 | | P33 | TIOCD0 | P33 |
| 1 | 0 | | P32 | TIOCC0 | P32 |
| 1 | 1 | | P33 | TIOCC0 | P33 |

| TPU0.TMDR.ICSELB | PFCR6.TPUMS0A | TPU0.TGRA | | TPU0.TGRB | |
|------------------|---------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA0 | P30 | TIOCB0 | P31 |
| 0 | 1 | | P31 | TIOCB0 | P31 |
| 1 | 0 | | P30 | TIOCA0 | P30 |
| 1 | 1 | | P31 | TIOCA0 | P31 |

| TPU1.TMDR.ICSELB | PFCR6.TPUMS1 | TPU1.TGRA | | TPU1.TGRB | |
|------------------|--------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA1 | P34 | TIOCB1 | P35 |
| 0 | 1 | | P35 | TIOCB1 | P35 |
| 1 | 0 | | P34 | TIOCA1 | P34 |
| 1 | 1 | | P35 | TIOCA1 | P35 |

| TPU2.TMDR.ICSELB | PFCR6.TPUMS2 | TPU2.TGRA | | TPU2.TGRB | |
|------------------|--------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA2 | P36 | TIOCB2 | P37 |
| 0 | 1 | | P37 | TIOCB2 | P37 |
| 1 | 0 | | P36 | TIOCA2 | P36 |
| 1 | 1 | | P37 | TIOCA2 | P37 |

| TPU3.TMDR.ICSELD | PFCR6.TPUMS3B | TPU3.TGRC | | TPU3.TGRD | |
|------------------|---------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCC3 | P22 | TIOCD3 | P23 |
| 0 | 1 | | P23 | TIOCD3 | P23 |
| 1 | 0 | | P22 | TIOCC3 | P22 |
| 1 | 1 | | P23 | TIOCC3 | P23 |

| TPU3.TMDR.ICSELB | PFCR6.TPUMS3A | TPU3.TGRA | | TPU3.TGRB | |
|------------------|---------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA3 | P21 | TIOCB3 | P20 |
| 0 | 1 | | P20 | TIOCB3 | P20 |
| 1 | 0 | | P21 | TIOCA3 | P21 |
| 1 | 1 | | P20 | TIOCA3 | P20 |

| TPU4.TMDR.ICSELB | PFCR6.TPUMS4 | TPU4.TGRA | | TPU4.TGRB | |
|------------------|--------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA4 | P25 | TIOCB4 | P24 |
| 0 | 1 | | P24 | TIOCB4 | P24 |
| 1 | 0 | | P25 | TIOCA4 | P25 |
| 1 | 1 | | P24 | TIOCA4 | P24 |

| TPU5.TMDR.ICSELB | PFCR6.TPUMS5 | TPU5.TGRA | | TPU5.TGRB | |
|------------------|--------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA5 | P26 | TIOCB5 | P27 |
| 0 | 1 | | P27 | TIOCB5 | P27 |
| 1 | 0 | | P26 | TIOCA5 | P26 |
| 1 | 1 | | P27 | TIOCA5 | P27 |

14.2.14 端口功能控制寄存器 7 (PFCR7)

地址 0008 C107h

| | | | | | | | | |
|-------|-------------|-------------|-------------|-------------|------------|------------|-------------|-------------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | TPUM S11 | TPUM S10 | TPUM S9A | TPUM S9B | TPUM S8 | TPUM S7 | TPUM S6A | TPUM S6B |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|---------|--------------------------|--|-----|
| b0 | TPUMS6B | TPU 输入 / 输出引脚的多功能选择 6B 位 | 0: 输出比较的输出和输入捕捉的输入为 PA2。 1: 输入捕捉的输入为 PA3, 输出比较的输出为 PA2。 | R/W |
| b1 | TPUMS6A | TPU 输入 / 输出引脚的多功能选择 6A 位 | 0: 输出比较的输出和输入捕捉的输入为 PA0。 1: 输入捕捉的输入为 PA1, 输出比较的输出为 PA0。 | R/W |
| b2 | TPUMS7 | TPU 输入 / 输出引脚的多功能选择 7 位 | 0: 输出比较的输出和输入捕捉的输入为 PA4。 1: 输入捕捉的输入为 PA5, 输出比较的输出为 PA4。 | R/W |
| b3 | TPUMS8 | TPU 输入 / 输出引脚的多功能选择 8 位 | 0: 输出比较的输出和输入捕捉的输入为 PA6。 1: 输入捕捉的输入为 PA7, 输出比较的输出为 PA6。 | R/W |
| b4 | TPUMS9B | TPU 输入 / 输出引脚的多功能选择 9B 位 | 0: 输出比较的输出和输入捕捉的输入为 PB2。 1: 输入捕捉的输入为 PB3, 输出比较的输出为 PB2。 | R/W |
| b5 | TPUMS9A | TPU 输入 / 输出引脚的多功能选择 9A 位 | 0: 输出比较的输出和输入捕捉的输入为 PB0。 1: 输入捕捉的输入为 PB1, 输出比较的输出为 PB0。 | R/W |
| b6 | TPUMS10 | TPU 输入 / 输出引脚的多功能选择 10 位 | 0: 输出比较的输出和输入捕捉的输入为 PB4。 1: 输入捕捉的输入为 PB5, 输出比较的输出为 PB4。 | R/W |
| b7 | TPUMS11 | TPU 输入 / 输出引脚的多功能选择 11 位 | 0: 输出比较的输出和输入捕捉的输入为 PB6。 1: 输入捕捉的输入为 PB7, 输出比较的输出为 PB6。 | R/W |

PFCR7 寄存器是选择 TPU (单元 1) 输入 / 输出引脚多功能的寄存器。

能通过将 TPUMS6A 位、TPUMS7 位、TPUMS8 位、TPUMS9A 位、TPUMS10 位和 TPUMS11 位置“1”，将 TPUM 的 TGRA 和 TGRB 的输入捕捉的输入分配到相同的引脚，并且能通过将 TPUMS6B 位和 TPUMS9B 位置“1”，将 TPUM 的 TGRC 和 TGRD 的输入捕捉的输入分配到相同的引脚。

也能通过设定 TPU 的定时器模式寄存器 (TPUm.TMDR)，将 TGRA 和 TGRB、TGRC 和 TGRD 的输入捕捉的输入设定到相同的引脚。

PFCR7 寄存器和 TPUm.TMDR 寄存器的设定值与输入捕捉的输入和外部引脚的关系如表 14.7 所示。

TPUMS6B 位 (TPU 输入 / 输出引脚的多功能选择 6B 位)

此位选择 TIOCC6 的输入引脚。

TPUMS6A 位 (TPU 输入 / 输出引脚的多功能选择 6A 位)

此位选择 TIOCA6 的输入引脚。

TPUMS7 位 (TPU 输入 / 输出引脚的多功能选择 7 位)

此位选择 TIOCA7 的输入引脚。

TPUMS8 位 (TPU 输入 / 输出引脚的多功能选择 8 位)

此位选择 TIOCA8 的输入引脚。

TPUMS9B 位 (TPU 输入 / 输出引脚的多功能选择 9B 位)

此位选择 TIOCC9 的输入引脚。

TPUMS9A 位（TPU 输入 / 输出引脚的多功能选择 9A 位）
此位选择 TIOCA9 的输入引脚。

TPUMS10 位（TPU 输入 / 输出引脚的多功能选择 10 位）
此位选择 TIOCA10 的输入引脚。

TPUMS11 位（TPU 输入 / 输出引脚的多功能选择 11 位）
此位选择 TIOCA11 的输入引脚。

表 14.7 PFCR7 寄存器和 TPUm 的定时器模式寄存器设定值与输入捕捉的输入和外部引脚的关系

| TPU6.TMDR.ICSELD | PFCR7.TPUMS6B | TPU6.TGRC | | TPU6.TGRD | |
|------------------|---------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCC6 | PA2 | TIOCD6 | PA3 |
| 0 | 1 | | PA3 | TIOCD6 | PA3 |
| 1 | 0 | | PA2 | TIOCC6 | PA2 |
| 1 | 1 | | PA3 | TIOCC6 | PA3 |

| TPU6.TMDR.ICSELB | PFCR7.TPUMS6A | TPU6.TGRA | | TPU6.TGRB | |
|------------------|---------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA6 | PA0 | TIOCB6 | PA1 |
| 0 | 1 | | PA1 | TIOCB6 | PA1 |
| 1 | 0 | | PA0 | TIOCA6 | PA0 |
| 1 | 1 | | PA1 | TIOCA6 | PA1 |

| TPU7.TMDR.ICSELB | PFCR7.TPUMS7 | TPU7.TGRA | | TPU7.TGRB | |
|------------------|--------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA7 | PA4 | TIOCB7 | PA5 |
| 0 | 1 | | PA5 | TIOCB7 | PA5 |
| 1 | 0 | | PA4 | TIOCA7 | PA4 |
| 1 | 1 | | PA5 | TIOCA7 | PA5 |

| TPU8.TMDR.ICSELB | PFCR7.TPUMS8 | TPU8.TGRA | | TPU8.TGRB | |
|------------------|--------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA8 | PA6 | TIOCB8 | PA7 |
| 0 | 1 | | PA7 | TIOCB8 | PA7 |
| 1 | 0 | | PA6 | TIOCA8 | PA6 |
| 1 | 1 | | PA7 | TIOCA8 | PA7 |

| TPU9.TMDR.ICSELD | PFCR7.TPUMS9B | TPU9.TGRC | | TPU9.TGRD | |
|------------------|---------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCC9 | PB2 | TIOCD9 | PB3 |
| 0 | 1 | | PB3 | TIOCD9 | PB3 |
| 1 | 0 | | PB2 | TIOCC9 | PB2 |
| 1 | 1 | | PB3 | TIOCC9 | PB3 |

| TPU9.TMDR.ICSELB | PFCR7.TPUMS9A | TPU9.TGRA | | TPU9.TGRB | |
|------------------|---------------|-----------|------|-----------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA9 | PB0 | TIOCB9 | PB1 |
| 0 | 1 | | PB1 | TIOCB9 | PB1 |
| 1 | 0 | | PB0 | TIOCA9 | PB0 |
| 1 | 1 | | PB1 | TIOCA9 | PB1 |

| TPU10.TMDR.ICSELB | PFCR7.TPUMS10 | TPU10.TGRA | | TPU10.TGRB | |
|-------------------|---------------|------------|------|------------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA10 | PB4 | TIOCB10 | PB5 |
| 0 | 1 | | PB5 | TIOCB10 | PB5 |
| 1 | 0 | | PB4 | TIOCA10 | PB4 |
| 1 | 1 | | PB5 | TIOCA10 | PB5 |

| TPU11.TMDR.ICSELB | PFCR7.TPUMS11 | TPU11.TGRA | | TPU11.TGRB | |
|-------------------|---------------|------------|------|------------|------|
| | | 输入捕捉的输入 | 外部引脚 | 输入捕捉的输入 | 外部引脚 |
| 0 | 0 | TIOCA11 | PB6 | TIOCB11 | PB7 |
| 0 | 1 | | PB7 | TIOCB11 | PB7 |
| 1 | 0 | | PB6 | TIOCA11 | PB6 |
| 1 | 1 | | PB7 | TIOCA11 | PB7 |

14.2.15 端口功能控制寄存器 8 (PFCR8)

地址 0008 C108h

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|------|------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| ITS15 | ITS14 | ITS13 | ITS12 | ITS11 | ITS10 | ITS9 | ITS8 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|-------------|--|-----|
| b0 | ITS8 | IRQ8 引脚选择位 | 0: 将 P00 设定为 IRQ8-A 输入引脚 1: 将 P40 设定为 IRQ8-B 输入引脚 | R/W |
| b1 | ITS9 | IRQ9 引脚选择位 | 0: 将 P01 设定为 IRQ9-A 输入引脚 1: 将 P41 设定为 IRQ9-B 输入引脚 | R/W |
| b2 | ITS10 | IRQ10 引脚选择位 | 0: 将 P02 设定为 IRQ10-A 输入引脚 1: 将 P42 设定为 IRQ10-B 输入引脚 | R/W |
| b3 | ITS11 | IRQ11 引脚选择位 | 0: 将 P03 设定为 IRQ11-A 输入引脚 1: 将 P43 设定为 IRQ11-B 输入引脚 | R/W |
| b4 | ITS12 | IRQ12 引脚选择位 | 0: 将 P04 设定为 IRQ12-A 输入引脚 1: 将 P44 设定为 IRQ12-B 输入引脚 | R/W |
| b5 | ITS13 | IRQ13 引脚选择位 | 0: 将 P05 设定为 IRQ13-A 输入引脚 1: 将 P45 设定为 IRQ13-B 输入引脚 | R/W |
| b6 | ITS14 | IRQ14 引脚选择位 | 0: 将 P76 设定为 IRQ14-A 输入引脚 1: 将 P46 设定为 IRQ14-B 输入引脚 | R/W |
| b7 | ITS15 | IRQ15 引脚选择位 | 0: 将 P65 设定为 IRQ15-A 输入引脚 1: 将 P47 设定为 IRQ15-B 输入引脚 | R/W |

PFCR8 寄存器是选择 IRQ8 ~ IRQ15 输入引脚的寄存器。

ITS_n (IRQ_n 引脚选择位) (n=8 ~ 15)
此位选择 IRQ_n 的输入引脚。

14.2.16 端口功能控制寄存器 9 (PFCR9)

地址 0008 C109h

| | | | | | | | |
|------|------|------|------|------|------|------|------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| ITS7 | ITS6 | ITS5 | ITS4 | ITS3 | ITS2 | ITS1 | ITS0 |

复位后的值 0 0 0 0 0 0 0 0

| 位 | 符号 | 位名 | 功能 | R/W |
|----|------|------------|--|-----|
| b0 | ITS0 | IRQ0 引脚选择位 | 0: 将 P30 设定为 IRQ0-A 输入引脚 1: 将 P10 设定为 IRQ0-B 输入引脚 | R/W |
| b1 | ITS1 | IRQ1 引脚选择位 | 0: 将 P31 设定为 IRQ1-A 输入引脚 1: 将 P11 设定为 IRQ1-B 输入引脚 | R/W |
| b2 | ITS2 | IRQ2 引脚选择位 | 0: 将 P32 设定为 IRQ2-A 输入引脚 1: 将 P12 设定为 IRQ2-B 输入引脚 | R/W |
| b3 | ITS3 | IRQ3 引脚选择位 | 0: 将 P33 设定为 IRQ3-A 输入引脚 1: 将 P13 设定为 IRQ3-B 输入引脚 | R/W |
| b4 | ITS4 | IRQ4 引脚选择位 | 0: 将 P34 设定为 IRQ4-A 输入引脚 1: 将 P14 设定为 IRQ4-B 输入引脚 | R/W |
| b5 | ITS5 | IRQ5 引脚选择位 | 0: 将 PE5 设定为 IRQ5-A 输入引脚 1: 将 P15 设定为 IRQ5-B 输入引脚 | R/W |
| b6 | ITS6 | IRQ6 引脚选择位 | 0: 将 PE6 设定为 IRQ6-A 输入引脚 1: 将 P16 设定为 IRQ6-B 输入引脚 | R/W |
| b7 | ITS7 | IRQ7 引脚选择位 | 0: 将 PE7 设定为 IRQ7-A 输入引脚 1: 将 P17 设定为 IRQ7-B 输入引脚 | R/W |

PFCR9 寄存器是选择 IRQ0 ~ IRQ7 输入引脚的寄存器。

ITS_n (IRQ_n 引脚选择位) (n=0 ~ 7)此位选择 IRQ_n 的输入引脚。

14.3 端口的设定

在引脚名后面附加“_OE”记载各外围模块的引脚（例如：TIOCA4_OE）。这是表示将对象的功能输出设定为有效（“1”）或者其他（“0”）。

各端口输出信号的有效设定一览表如表 14.8 所示，对应的输出信号的详细内容请参照“各外围模块的寄存器说明”。

能通过端口功能控制寄存器 y（PFCRy），更改各外围模块引脚名末尾带有 A ~ D 的任意一个字符的引脚功能。用 () 记载了用作输入的引脚，表中的“x”表示“Don't care”。

通过将输入缓冲控制寄存器（Pm.ICR）的对应位置“1”，从引脚向各外围模块的输入功能变为有效（注 1）。要使用有效的输入功能时，需要通过各自的外围模块进行设定，设定方法请参照各外围模块的章节。

注 1. 在将输入功能分配到多个外部引脚时，不仅需要设定 Pm.ICR 寄存器，还要设定 PFCRy 寄存器并且选择输入引脚。

14.3.1 端口 0（P0）

(1) P00/(TMR12)/TxD6/(IRQ8-A)

通过 SCI 的寄存器和 P0.DDR.B0 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|-------------|---------|-----------|
| | | SCI | I/O 端口 |
| | | TxD6_OE | P0.DDR.B0 |
| SCI | TxD6 输出 | 1 | x |
| I/O 端口 | P00 输出 | 0 | 1 |
| | P00 输入（初始值） | 0 | 0 |

(2) P01/(TMC12)/(RxD6)/(IRQ9-A)

通过设定 P0.DDR.B1 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|-------------|-----------|--|
| | | I/O 端口 | |
| | | P0.DDR.B1 | |
| I/O 端口 | P01 输出 | 1 | |
| | P01 输入（初始值） | 0 | |

(3) P02/TMO2/SCK6/(IRQ10-A)/(TRST#)

通过 TMR、SCI 的寄存器和 P0.DDR.B2 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|-------------|---------|---------|-----------|
| | | TMR | SCI | I/O 端口 |
| | | TMO2_OE | SCK6_OE | P0.DDR.B2 |
| TMR | TMO2 输出 | 1 | x | x |
| SCI | SCK6 输出 | 0 | 1 | x |
| I/O 端口 | P02 输出 | 0 | 0 | 1 |
| | P02 输入（初始值） | 0 | 0 | 0 |

(4) P03/(TMRI3)/SCK4/(IRQ11-A)/(TMS)

通过 SCI 的寄存器和 P0.DDR.B3 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|---------|-----------|
| | | SCI | I/O 端口 |
| | | SCK4_OE | P0.DDR.B3 |
| SCI | SCK4 输出 | 1 | x |
| I/O 端口 | P03 输出 | 0 | 1 |
| | P03 输入 (初始值) | 0 | 0 |

(5) P04/(TMCI3)/TxD4/(IRQ12-A)/(TDI)

通过 SCI 的寄存器和 P0.DDR.B4 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|---------|-----------|
| | | SCI | I/O 端口 |
| | | TxD4_OE | P0.DDR.B4 |
| SCI | TxD4 输出 | 1 | x |
| I/O 端口 | P04 输出 | 0 | 1 |
| | P04 输入 (初始值) | 0 | 0 |

(6) P05/TMO3/(RxD4)/(IRQ13-A)/(TCK)

通过 TMR 的寄存器和 P0.DDR.B5 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|---------|-----------|
| | | TMR | I/O 端口 |
| | | TMO3_OE | P0.DDR.B5 |
| TMR | TMO3 输出 | 1 | x |
| I/O 端口 | P05 输出 | 0 | 1 |
| | P05 输入 (初始值) | 0 | 0 |

14.3.2 端口 1 (P1)

(1) P10/(IRQ0-B)

通过设定 P1.DDR.B0 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|-----------|--|
| | | I/O 端口 | |
| | | P1.DDR.B0 | |
| I/O 端口 | P10 输出 | 1 | |
| | P10 输入 (初始值) | 0 | |

(2) P11/SCK2/(IRQ1-B)

通过 SCI 的寄存器和 P1.DDR.B1 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|---------|-----------|
| | | SCI | I/O 端口 |
| | | SCK2_OE | P1.DDR.B1 |
| SCI | SCK2 输出 | 1 | x |
| I/O 端口 | P11 输出 | 0 | 1 |
| | P11 输入 (初始值) | 0 | 0 |

(3) P12/(RxD2)/(IRQ2-B)

通过设定 P1.DDR.B2 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|-----------|--|
| | | I/O 端口 | |
| | | P1.DDR.B2 | |
| I/O 端口 | P12 输出 | 1 | |
| | P12 输入 (初始值) | 0 | |

(4) P13/TxD2/(ADTRG0#)/(IRQ3-B)

通过 SCI 的寄存器和 P1.DDR.B3 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|---------|-----------|
| | | SCI | I/O 端口 |
| | | TxD2_OE | P1.DDR.B3 |
| SCI | TxD2 输出 | 1 | x |
| I/O 端口 | P13 输出 | 0 | 1 |
| | P13 输入 (初始值) | 0 | 0 |

(5) P14/(TCLKA-B)/SDA1/(IRQ4-B)

通过 RIIC 的寄存器和 P1.DDR.B4 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|---------|-----------|
| | | RIIC | I/O 端口 |
| | | SDA1_OE | P1.DDR.B4 |
| RIIC | SDA1 输入 / 输出 | 1 | x |
| I/O 端口 | P14 输出 | 0 | 1 |
| | P14 输入 (初始值) | 0 | 0 |

(6) P15/(TCLKB-B)/SCK3/SCL1/(IRQ5-B)

通过 SCI、RIIC 的寄存器和 P1.DDR.B5 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|---------|---------|-----------|
| | | SCI | RIIC | I/O 端口 |
| | | SCK3_OE | SCL1_OE | P1.DDR.B5 |
| SCI | SCK3 输出 | 1 | x | x |
| RIIC | SCL1 输入 / 输出 | 0 | 1 | x |
| I/O 端口 | P15 输出 | 0 | 0 | 1 |
| | P15 输入 (初始值) | 0 | 0 | 0 |

(7) P16/(TCLKC-B)/(RxD3)/SDA0/(IRQ6-B)

通过 RIIC 的寄存器和 P1.DDR.B6 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|---------|-----------|
| | | RIIC | I/O 端口 |
| | | SDA0_OE | P1.DDR.B6 |
| RIIC | SDA0 输入 / 输出 | 1 | x |
| I/O 端口 | P16 输出 | 0 | 1 |
| | P16 输入 (初始值) | 0 | 0 |

(8) P17/(TCLKD-B)/TxD3/SCL0/(ADTRG1#)/(IRQ7-B)

通过 SCI、RIIC 的寄存器和 P1.DDR.B7 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|---------|---------|-----------|
| | | SCI | RIIC | I/O 端口 |
| | | TxD3_OE | SCL0_OE | P1.DDR.B7 |
| SCI | TxD3 输出 | 1 | x | x |
| RIIC | SCL0 输入 / 输出 | 0 | 1 | x |
| I/O 端口 | P17 输出 | 0 | 0 | 1 |
| | P17 输入 (初始值) | 0 | 0 | 0 |

14.3.3 端口 2 (P2)

(1) P20/PO0/(TIOCA3)/TIOCB3/(TMRI0)/TxD0

通过 TPU、SCI、PPG 的寄存器和 P2.DDR.B0 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|-----------|---------|--------|-----------|
| | | TPU | SCI | PPG | I/O 端口 |
| | | TIOCB3_OE | TxD0_OE | PO0_OE | P2.DDR.B0 |
| TPU | TIOCB3 输出 | 1 | x | x | x |
| SCI | TxD0 输出 | 0 | 1 | x | x |
| PPG | PO0 输出 | 0 | 0 | 1 | x |
| I/O 端口 | P20 输出 | 0 | 0 | 0 | 1 |
| | P20 输入 (初始值) | 0 | 0 | 0 | 0 |

(2) P21/PO1/TIOCA3/(TMCI0)/(RxD0)

通过 TPU、PPG 的寄存器和 P2.DDR.B1 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|--------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCA3_OE | PO1_OE | P2.DDR.B1 |
| TPU | TIOCA3 输出 | 1 | x | x |
| PPG | PO1 输出 | 0 | 1 | x |
| I/O 端口 | P21 输出 | 0 | 0 | 1 |
| | P21 输入 (初始值) | 0 | 0 | 0 |

(3) P22/PO2/TIOCC3/TMO0/SCK0

通过 TPU、TMR、SCI、PPG 的寄存器和 P2.DDR.B2 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | | |
|--------|--------------|-----------|---------|---------|--------|-----------|
| | | TPU | TMR | SCI | PPG | I/O 端口 |
| | | TIOCC3_OE | TMO0_OE | SCK0_OE | PO2_OE | P2.DDR.B2 |
| TPU | TIOCC3 输出 | 1 | x | x | x | x |
| TMR | TMO0 输出 | 0 | 1 | x | x | x |
| SCI | SCK0 输出 | 0 | 0 | 1 | x | x |
| PPG | PO2 输出 | 0 | 0 | 0 | 1 | x |
| I/O 端口 | P22 输出 | 0 | 0 | 0 | 0 | 1 |
| | P22 输入 (初始值) | 0 | 0 | 0 | 0 | 0 |

(4) P23/PO3/(TIOCC3)/TIOCD3

通过 TPU、PPG 的寄存器和 P2.DDR.B3 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|--------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCD3_OE | PO3_OE | P2.DDR.B3 |
| TPU | TIOCD3 输出 | 1 | x | x |
| PPG | PO3 输出 | 0 | 1 | x |
| I/O 端口 | P23 输出 | 0 | 0 | 1 |
| | P23 输入 (初始值) | 0 | 0 | 0 |

(5) P24/PO4/(TIOCA4)/TIOCB4/(TMR11)

通过 TPU、PPG 的寄存器和 P2.DDR.B4 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|--------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCB4_OE | PO4_OE | P2.DDR.B4 |
| TPU | TIOCB4 输出 | 1 | x | x |
| PPG | PO4 输出 | 0 | 1 | x |
| I/O 端口 | P24 输出 | 0 | 0 | 1 |
| | P24 输入 (初始值) | 0 | 0 | 0 |

(6) P25/PO5/TIOCA4/(TMCI1)/(RxD1)

通过 TPU、PPG 的寄存器和 P2.DDR.B5 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|--------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCA4_OE | PO5_OE | P2.DDR.B5 |
| TPU | TIOCA4 输出 | 1 | x | x |
| PPG | PO5 输出 | 0 | 1 | x |
| I/O 端口 | P25 输出 | 0 | 0 | 1 |
| | P25 输入 (初始值) | 0 | 0 | 0 |

(7) P26/PO6/TIOCA5/TMO1/TxD1

通过 TPU、TMR、SCI、PPG 的寄存器和 P2.DDR.B6 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | | |
|--------|--------------|-----------|---------|---------|--------|-----------|
| | | TPU | TMR | SCI | PPG | I/O 端口 |
| | | TIOCA5_OE | TMO1_OE | TxD1_OE | PO6_OE | P2.DDR.B6 |
| TPU | TIOCA5 输出 | 1 | x | x | x | x |
| TMR | TMO1 输出 | 0 | 1 | x | x | x |
| SCI | TxD1 输出 | 0 | 0 | 1 | x | x |
| PPG | PO6 输出 | 0 | 0 | 0 | 1 | x |
| I/O 端口 | P26 输出 | 0 | 0 | 0 | 0 | 1 |
| | P26 输入 (初始值) | 0 | 0 | 0 | 0 | 0 |

(8) P27/PO7/(TIOCA5)/TIOCB5/SCK1

通过 TPU、SCI、PPG 的寄存器和 P2.DDR.B7 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|-----------|---------|--------|-----------|
| | | TPU | SCI | PPG | I/O 端口 |
| | | TIOCB5_OE | SCK1_OE | PO7_OE | P2.DDR.B7 |
| TPU | TIOCB5 输出 | 1 | x | x | x |
| SCI | SCK1 输出 | 0 | 1 | x | x |
| PPG | PO7 输出 | 0 | 0 | 1 | x |
| I/O 端口 | P27 输出 | 0 | 0 | 0 | 1 |
| | P27 输入 (初始值) | 0 | 0 | 0 | 0 |

14.3.4 端口 3 (P3)

(1) P30/PO8/TIOCA0/(IRQ0-A)

通过 TPU、PPG 的寄存器和 P3.DDR.B0 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|--------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCA0_OE | PO8_OE | P3.DDR.B0 |
| TPU | TIOCA0 输出 | 1 | x | x |
| PPG | PO8 输出 | 0 | 1 | x |
| I/O 端口 | P30 输出 | 0 | 0 | 1 |
| | P30 输入 (初始值) | 0 | 0 | 0 |

(2) P31/PO9/(TIOCA0)/TIOCB0/(IRQ1-A)

通过 TPU、PPG 的寄存器和 P3.DDR.B1 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|--------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCB0_OE | PO9_OE | P3.DDR.B1 |
| TPU | TIOCB0 输出 | 1 | x | x |
| PPG | PO9 输出 | 0 | 1 | x |
| I/O 端口 | P31 输出 | 0 | 0 | 1 |
| | P31 输入 (初始值) | 0 | 0 | 0 |

(3) P32/PO10/TIOCC0/(TCLKA-A)/(IRQ2-A)

通过 TPU、PPG 的寄存器和 P3.DDR.B2 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|---------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCC0_OE | PO10_OE | P3.DDR.B2 |
| TPU | TIOCC0 输出 | 1 | x | x |
| PPG | PO10 输出 | 0 | 1 | x |
| I/O 端口 | P32 输出 | 0 | 0 | 1 |
| | P32 输入 (初始值) | 0 | 0 | 0 |

(4) P33/PO11/(TIOCC0)/TIOCD0/(TCLKB-A)/(IRQ3-A)

通过 TPU、PPG 的寄存器和 P3.DDR.B3 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|---------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCD0_OE | PO11_OE | P3.DDR.B3 |
| TPU | TIOCD0 输出 | 1 | x | x |
| PPG | PO11 输出 | 0 | 1 | x |
| I/O 端口 | P33 输出 | 0 | 0 | 1 |
| | P33 输入 (初始值) | 0 | 0 | 0 |

(5) P34/PO12/TIOCA1/(IRQ4-A)

通过 TPU、PPG 的寄存器和 P3.DDR.B4 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|---------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCA1_OE | PO12_OE | P3.DDR.B4 |
| TPU | TIOCA1 输出 | 1 | x | x |
| PPG | PO12 输出 | 0 | 1 | x |
| I/O 端口 | P34 输出 | 0 | 0 | 1 |
| | P34 输入 (初始值) | 0 | 0 | 0 |

(6) P35/PO13/(TIOCA1)/TIOCB1/(TCLKC-A)

通过 TPU、PPG 的寄存器和 P3.DDR.B5 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|---------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCB1_OE | PO13_OE | P3.DDR.B5 |
| TPU | TIOCB1 输出 | 1 | x | x |
| PPG | PO13 输出 | 0 | 1 | x |
| I/O 端口 | P35 输出 | 0 | 0 | 1 |
| | P35 输入 (初始值) | 0 | 0 | 0 |

(7) P36/PO14/TIOCA2

通过 TPU、PPG 的寄存器和 P3.DDR.B6 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|---------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCA2_OE | PO14_OE | P3.DDR.B6 |
| TPU | TIOCA2 输出 | 1 | x | x |
| PPG | PO14 输出 | 0 | 1 | x |
| I/O 端口 | P36 输出 | 0 | 0 | 1 |
| | P36 输入 (初始值) | 0 | 0 | 0 |

(8) P37/PO15/(TIOCA2)/TIOCB2/(TCLKD-A)

通过 TPU、PPG 的寄存器和 P3.DDR.B7 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | |
|--------|--------------|-----------|---------|-----------|
| | | TPU | PPG | I/O 端口 |
| | | TIOCB2_OE | PO15_OE | P3.DDR.B7 |
| TPU | TIOCB2 输出 | 1 | x | x |
| PPG | PO15 输出 | 0 | 1 | x |
| I/O 端口 | P37 输出 | 0 | 0 | 1 |
| | P37 输入 (初始值) | 0 | 0 | 0 |

14.3.5 端口 4 (P4)

(1) P40/(AN0)/(IRQ8-B)

通过设定 P4.DDR.B0 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P4.DDR.B0 |
| I/O 端口 | P40 输出 | 1 |
| | P40 输入 (初始值) | 0 |

(2) P41/(AN1)/(IRQ9-B)

通过设定 P4.DDR.B1 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P4.DDR.B1 |
| I/O 端口 | P41 输出 | 1 |
| | P41 输入 (初始值) | 0 |

(3) P42/(AN2)/(IRQ10-B)

通过设定 P4.DDR.B2 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P4.DDR.B2 |
| I/O 端口 | P42 输出 | 1 |
| | P42 输入 (初始值) | 0 |

(4) P43/(AN3)/(IRQ11-B)

通过设定 P4.DDR.B3 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P4.DDR.B3 |
| I/O 端口 | P43 输出 | 1 |
| | P43 输入 (初始值) | 0 |

(5) P44/(AN4)/(IRQ12-B)

通过设定 P4.DDR.B4 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P4.DDR.B4 |
| I/O 端口 | P44 输出 | 1 |
| | P44 输入 (初始值) | 0 |

(6) P45/(AN5)/(IRQ13-B)

通过设定 P4.DDR.B5 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|-----------|--|
| | | I/O 端口 | |
| | | P4.DDR.B5 | |
| I/O 端口 | P45 输出 | 1 | |
| | P45 输入 (初始值) | 0 | |

(7) P46/(AN6)/(IRQ14-B)

通过设定 P4.DDR.B6 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|-----------|--|
| | | I/O 端口 | |
| | | P4.DDR.B6 | |
| I/O 端口 | P46 输出 | 1 | |
| | P46 输入 (初始值) | 0 | |

(8) P47/(AN7)/(IRQ15-B)

通过设定 P4.DDR.B7 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|-----------|--|
| | | I/O 端口 | |
| | | P4.DDR.B7 | |
| I/O 端口 | P47 输出 | 1 | |
| | P47 输入 (初始值) | 0 | |

14.3.6 端口 5 (P5)

(1) P50/WR0#/WR#

通过总线控制的寄存器和 P5.DDR.B0 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|------------------|----------------|-----------|
| | | 总线控制 | I/O 端口 |
| | | WR0#_OE/WR#_OE | P5.DDR.B0 |
| 总线控制 | WR0#/WR# 输出 (注1) | 1 | x |
| I/O 端口 | P50 输出 | 0 | 1 |
| | P50 输入 (初始值) | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(2) P51/WR1#/BC1#

通过端口功能控制寄存器 y (PFCRy) 和 P5.DDR.B1 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|-------------------|-----------------|-----------|
| | | 总线控制 | I/O 端口 |
| | | WR1#_OE/BC1#_OE | P5.DDR.B1 |
| 总线控制 | WR1#/BC1# 输出 (注1) | 1 | x |
| I/O 端口 | P51 输出 | 0 | 1 |
| | P51 输入 (初始值) | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(3) P52/RD#

通过运行模式以及系统控制寄存器 0 (SYSCR0) 的外部总线允许位 (EXBE) 和 P5.DDR.B2 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|-------|-----------|
| | | 总线控制 | I/O 端口 |
| | | RD_OE | P5.DDR.B2 |
| 总线控制 | RD# 输出 (注1) | 1 | x |
| I/O 端口 | P52 输出 | 0 | 1 |
| | P52 输入 (初始值) | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(4) P53/BCLK

通过时钟发生电路的寄存器和 P5.DDR.B3 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|---------------------|--|
| | | I/O 端口 | |
| | | P5.DDR.B3 (BCLK_OE) | |
| 时钟发生电路 | BCLK 输出 | 1 | |
| I/O 端口 | P53 输入 (初始值) | 0 | |

注 1. 要输出 BCLK 时, 必须先将 SCKCR.PSTOP1 位置 "1", 停止 BCLK, 然后通过 P5.DDR.B3 位将引脚设定为输出, 并且重新将 SCKCR.PSTOP1 位置 "0", 输出 BCLK。

(5) P54/TRDATA0

通过设定 P5.DDR.B4 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|-----------|--|
| | | I/O 端口 | |
| | | P5.DDR.B4 | |
| I/O 端口 | P54 输出 | 1 | |
| | P54 输入 (初始值) | 0 | |

(6) P55/TRDATA1

通过设定 P5.DDR.B5 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|-------------|-----------|--|
| | | I/O 端口 | |
| | | P5.DDR.B5 | |
| I/O 端口 | P55 输出 | 1 | |
| | P55 输入（初始值） | 0 | |

(7) P56/TRDATA2

通过设定 P5.DDR.B6 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|-------------|-----------|--|
| | | I/O 端口 | |
| | | P5.DDR.B6 | |
| I/O 端口 | P56 输出 | 1 | |
| | P56 输入（初始值） | 0 | |

(8) P57/(WAIT#)/TRDATA3

通过设定 P5.DDR.B7 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|-------------|-----------|--|
| | | I/O 端口 | |
| | | P5.DDR.B7 | |
| I/O 端口 | P57 输出 | 1 | |
| | P57 输入（初始值） | 0 | |

14.3.7 端口 6（P6）

(1) P60/CS0#/CS4#-A/CS5#-B

通过运行模式以及系统控制寄存器 0（SYSCR0）的外部总线允许位（EXBE）、总线控制的寄存器、端口功能控制寄存器 y（PFCRy）和 P6.DDR.B0 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|-------------|---------|-----------|-----------|-----------|
| | | 总线控制 | | | I/O 端口 |
| | | CS0#_OE | CS4#-A_OE | CS5#-B_OE | P6.DDR.B0 |
| 总线控制 | CS0# 输出 | 1 | x | x | x |
| | CS4#-A 输出 | x | 1 | x | x |
| | CS5#-B 输出 | x | x | 1 | x |
| I/O 端口 | P60 输出 | 0 | 0 | 0 | 1 |
| | P60 输入（初始值） | 0 | 0 | 0 | 0 |

(2) P61/CS1#/CS2#-B/CS5#-A/CS6#-B/CS7#-B

通过运行模式以及 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 P6.DDR.B1 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | | | | | |
|--------|----------------|---------|-----------|-----------|-----------|-----------|-----------|
| | | 总线控制 | | | | | I/O 端口 |
| | | CS1#_OE | CS2#-B_OE | CS5#-A_OE | CS6#-B_OE | CS7#-B_OE | P6.DDR.B1 |
| 总线控制 | CS1# 输出 (注1) | 1 | x | x | x | x | x |
| | CS2#-B 输出 (注1) | x | 1 | x | x | x | x |
| | CS5#-A 输出 (注1) | x | x | 1 | x | x | x |
| | CS6#-B 输出 (注1) | x | x | x | 1 | x | x |
| | CS7#-B 输出 (注1) | x | x | x | x | 1 | x |
| I/O 端口 | P61 输出 | 0 | 0 | 0 | 0 | 0 | 1 |
| | P61 输入 (初始值) | 0 | 0 | 0 | 0 | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(3) P62/CS2#-A/CS6#-A

通过 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 P6.DDR.B2 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | | |
|--------|----------------|-----------|-----------|-----------|
| | | 总线控制 | | I/O 端口 |
| | | CS2#-A_OE | CS6#-A_OE | P6.DDR.B2 |
| 总线控制 | CS2#-A 输出 (注1) | 1 | x | x |
| | CS6#-A 输出 (注1) | x | 1 | x |
| I/O 端口 | P62 输出 | 0 | 0 | 1 |
| | P62 输入 (初始值) | 0 | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(4) P63/CS3#-A/CS7#-A

通过 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 P6.DDR.B3 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | | |
|--------|----------------|-----------|-----------|-----------|
| | | 总线控制 | | I/O 端口 |
| | | CS3#-A_OE | CS7#-A_OE | P6.DDR.B3 |
| 总线控制 | CS3#-A 输出 (注1) | 1 | x | x |
| | CS7#-A 输出 (注1) | x | 1 | x |
| I/O 端口 | P63 输出 | 0 | 0 | 1 |
| | P63 输入 (初始值) | 0 | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(5) P64/CS4#-B

通过 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 P6.DDR.B4 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|----------------|-----------|-----------|
| | | 总线控制 | I/O 端口 |
| | | CS4#-B_OE | P6.DDR.B4 |
| 总线控制 | CS4#-B 输出 (注1) | 1 | x |
| I/O 端口 | P64 输出 | 0 | 1 |
| | P64 输入 (初始值) | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(6) P65/(IRQ15-A)

通过设定 P6.DDR.B5 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|-----------|--|
| | | I/O 端口 | |
| | | P6.DDR.B5 | |
| I/O 端口 | P65 输出 | 1 | |
| | P65 输入 (初始值) | 0 | |

(7) P66/DA0

通过 D/A 转换器的寄存器和 P6.DDR.B6 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|---------|--------------|---------|-----------|
| | | D/A 转换器 | I/O 端口 |
| | | DA0_OE | P6.DDR.B6 |
| D/A 转换器 | DA0 输出 | 1 | x |
| I/O 端口 | P66 输出 | 0 | 1 |
| | P66 输入 (初始值) | 0 | 0 |

(8) P67/DA1

通过 D/A 转换器的寄存器和 P6.DDR.B7 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|---------|--------------|---------|-----------|
| | | D/A 转换器 | I/O 端口 |
| | | DA1_OE | P6.DDR.B7 |
| D/A 转换器 | DA1 输出 | 1 | x |
| I/O 端口 | P67 输出 | 0 | 1 |
| | P67 输入 (初始值) | 0 | 0 |

14.3.8 端口 7 (P7)

(1) P70/CS3#-B/(ADTRG2#)

通过系统控制寄存器 0 (SYSCR0) 的外部总线允许位 (EXBE)、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 P7.DDR.B0 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | | |
|--------|----------------|-----------|-----------|--------|
| | | 总线控制 | | I/O 端口 |
| | | CS3#-B_OE | P7.DDR.B0 | |
| 总线控制 | CS3#-B 输出 (注1) | 1 | x | |
| I/O 端口 | P70 输出 | 0 | 1 | |
| | P70 输入 (初始值) | 0 | 0 | |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(2) P71/CS4#-C/CS5#-C/CS6#-C/CS7#-C

通过 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 P7.DDR.B1 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | | | | |
|--------|----------------|-----------|-----------|-----------|-----------|-----------|
| | | 总线控制 | | | | I/O 端口 |
| | | CS4#-C_OE | CS5#-C_OE | CS6#-C_OE | CS7#-C_OE | P7.DDR.B1 |
| 总线控制 | CS4#-C 输出 (注1) | 1 | x | x | x | x |
| | CS5#-C 输出 (注1) | x | 1 | x | x | x |
| | CS6#-C 输出 (注1) | x | x | 1 | x | x |
| | CS7#-C 输出 (注1) | x | x | x | 1 | x |
| I/O 端口 | P71 输出 | 0 | 0 | 0 | 0 | 1 |
| | P71 输入 (初始值) | 0 | 0 | 0 | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(3) P72

通过设定 P7.DDR.B2 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|-----------|--|
| | | I/O 端口 | |
| | | P7.DDR.B2 | |
| I/O 端口 | P72 输出 | 1 | |
| | P72 输入 (初始值) | 0 | |

(4) P73

通过设定 P7.DDR.B3 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|-----------|--|
| | | I/O 端口 | |
| | | P7.DDR.B3 | |
| I/O 端口 | P73 输出 | 1 | |
| | P73 输入 (初始值) | 0 | |

(5) P74/(ADTRG3#)

通过设定 P7.DDR.B4 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P7.DDR.B4 |
| I/O 端口 | P74 输出 | 1 |
| | P74 输入 (初始值) | 0 |

(6) P75

通过设定 P7.DDR.B5 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P7.DDR.B5 |
| I/O 端口 | P75 输出 | 1 |
| | P75 输入 (初始值) | 0 |

(7) P76/(IRQ14-A)

通过设定 P7.DDR.B6 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P7.DDR.B6 |
| I/O 端口 | P76 输出 | 1 |
| | P76 输入 (初始值) | 0 |

(8) P77

通过设定 P7.DDR.B7 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P7.DDR.B7 |
| I/O 端口 | P77 输出 | 1 |
| | P77 输入 (初始值) | 0 |

14.3.9 端口 8 (P8)

(1) P80

通过设定 P8.DDR.B0 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P8.DDR.B0 |
| I/O 端口 | P80 输出 | 1 |
| | P80 输入 (初始值) | 0 |

(2) P81/TRSYNC

通过设定 P8.DDR.B1 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P8.DDR.B1 |
| I/O 端口 | P81 输出 | 1 |
| | P81 输入 (初始值) | 0 |

(3) P82/TRCLK

通过设定 P8.DDR.B2 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P8.DDR.B2 |
| I/O 端口 | P82 输出 | 1 |
| | P82 输入 (初始值) | 0 |

(4) P83

通过设定 P8.DDR.B3 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P8.DDR.B3 |
| I/O 端口 | P83 输出 | 1 |
| | P83 输入 (初始值) | 0 |

(5) P84

通过设定 P8.DDR.B4 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P8.DDR.B4 |
| I/O 端口 | P84 输出 | 1 |
| | P84 输入 (初始值) | 0 |

(6) P85

通过设定 P8.DDR.B5 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P8.DDR.B5 |
| I/O 端口 | P85 输出 | 1 |
| | P85 输入 (初始值) | 0 |

(7) P86

通过设定 P8.DDR.B6 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P8.DDR.B6 |
| I/O 端口 | P86 输出 | 1 |
| | P86 输入 (初始值) | 0 |

14.3.10 端口 9 (P9)

(1) P90/(AN8)

通过设定 P9.DDR.B0 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P9.DDR.B0 |
| I/O 端口 | P90 输出 | 1 |
| | P90 输入 (初始值) | 0 |

(2) P91/(AN9)

通过设定 P9.DDR.B1 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P9.DDR.B1 |
| I/O 端口 | P91 输出 | 1 |
| | P91 输入 (初始值) | 0 |

(3) P92/(AN10)

通过设定 P9.DDR.B2 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P9.DDR.B2 |
| I/O 端口 | P92 输出 | 1 |
| | P92 输入 (初始值) | 0 |

(4) P93/(AN11)

通过设定 P9.DDR.B3 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P9.DDR.B3 |
| I/O 端口 | P93 输出 | 1 |
| | P93 输入 (初始值) | 0 |

(5) P94/(AN12)

通过设定 P9.DDR.B4 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P9.DDR.B4 |
| I/O 端口 | P94 输出 | 1 |
| | P94 输入 (初始值) | 0 |

(6) P95/(AN13)

通过设定 P9.DDR.B5 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P9.DDR.B5 |
| I/O 端口 | P95 输出 | 1 |
| | P95 输入 (初始值) | 0 |

(7) P96/(AN14)

通过设定 P9.DDR.B6 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P9.DDR.B6 |
| I/O 端口 | P96 输出 | 1 |
| | P96 输入 (初始值) | 0 |

(8) P97/(AN15)

通过设定 P9.DDR.B7 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | P9.DDR.B7 |
| I/O 端口 | P97 输出 | 1 |
| | P97 输入 (初始值) | 0 |

14.3.11 端口 A (PA)

(1) PA0/A0/BC0#/PO16/TIOCA6

通过 PPG、TPU 的寄存器、系统控制寄存器 0 (SYSCR0) 的外部总线允许位 (EXBE) 和 PA.DDR.B0 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | | | | |
|--------|--------------|-------|---------|-----------|---------|-----------|
| | | 总线控制 | | TPU | PPG | I/O 端口 |
| | | A0_OE | BC0#_OE | TIOCA6_OE | PO16_OE | PA.DDR.B0 |
| 总线控制 | 地址输出 (注1) | 1 | 0 | x | x | 1 |
| | 字节控制输出 (注1) | 0 | 1 | x | x | 1 |
| TPU | TIOCA6 输出 | 0 | 0 | 1 | x | x |
| PPG | PO16 输出 | 0 | 0 | 0 | 1 | x |
| I/O 端口 | PA0 输出 (注) | 0 | 0 | 0 | 0 | 1 |
| | PA0 输入 (初始值) | 0 | 0 | 0 | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时, 能通过 PA.DDR.B0=1 设定为地址输出。

(2) PA1/A1/PO17/(TIOCA6)/TIOCB6

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位和 PA.DDR.B1 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|-------|-----------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A1_OE | TIOCB6_OE | PO17_OE | PA.DDR.B1 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | 1 |
| TPU | TIOCB6 输出 | 0 | 1 | x | x |
| PPG | PO17 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PA1 输出 (注1) | 0 | 0 | 0 | 1 |
| | PA1 输入 (初始值) | 0 | 0 | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时, 能通过 PA.DDR.B1=1 设定为地址输出。

(3) PA2/A2/PO18/TIOCC6/(TCLKE)

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位和 PA.DDR.B2 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|-------|-----------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A2_OE | TIOCC6_OE | PO18_OE | PA.DDR.B2 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | 1 |
| TPU | TIOCC6 输出 | 0 | 1 | x | x |
| PPG | PO18 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PA2 输出 (注1) | 0 | 0 | 0 | 1 |
| | PA2 输入 (初始值) | 0 | 0 | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时, 能通过 PA.DDR.B2=1 设定为地址输出。

(4) PA3/A3/PO19/(TIOCC6)/TIOCD6/(TCLKF)

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位和 PA.DDR.B3 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|-------|-----------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A3_OE | TIOCD6_OE | PO19_OE | PA.DDR.B3 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | 1 |
| TPU | TIOCD6 输出 | 0 | 1 | x | x |
| PPG | PO19 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PA3 输出 (注1) | 0 | 0 | 0 | 1 |
| | PA3 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时, 能通过 PA.DDR.B3=1 设定为地址输出。

(5) PA4/A4/PO20/TIOCA7

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位和 PA.DDR.B4 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|-------|-----------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A4_OE | TIOCA7_OE | PO20_OE | PA.DDR.B4 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | 1 |
| TPU | TIOCA7 输出 | 0 | 1 | x | x |
| PPG | PO20 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PA4 输出 (注1) | 0 | 0 | 0 | 1 |
| | PA4 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时, 能通过 PA.DDR.B4=1 设定为地址输出。

(6) PA5/A5/PO21/(TIOCA7)/TIOCB7/(TCLKG)

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位和 PA.DDR.B5 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|-------|-----------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A5_OE | TIOCB7_OE | PO21_OE | PA.DDR.B5 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | 1 |
| TPU | TIOCB7 输出 | 0 | 1 | x | x |
| PPG | PO21 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PA5 输出 (注1) | 0 | 0 | 0 | 1 |
| | PA5 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时, 能通过 PA.DDR.B5=1 设定为地址输出。

(7) PA6/A6/PO22/TIOCA8

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位和 PA.DDR.B6 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|-------|-----------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A6_OE | TIOCA8_OE | PO22_OE | PA.DDR.B6 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | 1 |
| TPU | TIOCA8 输出 | 0 | 1 | x | x |
| PPG | PO22 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PA6 输出 (注1) | 0 | 0 | 0 | 1 |
| | PA6 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时, 能通过 PA.DDR.B6=1 设定为地址输出。

(8) PA7/A7/PO23/(TIOCA8)/TIOCB8/(TCLKH)

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位和 PA.DDR.B7 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|-------|-----------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A7_OE | TIOCB8_OE | PO23_OE | PA.DDR.B7 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | 1 |
| TPU | TIOCB8 输出 | 0 | 1 | x | x |
| PPG | PO23 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PA7 输出 (注1) | 0 | 0 | 0 | 1 |
| | PA7 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时, 能通过 PA.DDR.B7=1 设定为地址输出。

14.3.12 端口 B (PB)

(1) PB0/A8/PO24/TIOCA9

通过 PPG、TPU 的寄存器、系统控制寄存器 0 (SYSCR0) 的外部总线允许位 (EXBE)、端口功能控制寄存器 y (PFCRy) 和 PB.DDR.B0 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|-------|-----------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A8_OE | TIOCA9_OE | PO24_OE | PB.DDR.B0 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | x |
| TPU | TIOCA9 输出 | 0 | 1 | x | x |
| PPG | PO24 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PB0 输出 (注1) | 0 | 0 | 0 | 1 |
| | PB0 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(2) PB1/A9/PO25/(TIOCA9)/TIOCB9

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位、端口功能控制寄存器 y (PFCRy) 和 PB.DDR.B1 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|-------|-----------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A9_OE | TIOCB9_OE | PO25_OE | PB.DDR.B1 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | x |
| TPU | TIOCB9 输出 | 0 | 1 | x | x |
| PPG | PO25 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PB1 输出 (注1) | 0 | 0 | 0 | 1 |
| | PB1 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(3) PB2/A10/PO26/TIOCC9

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位、端口功能控制寄存器 y (PFCRy) 和 PB.DDR.B2 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|--------|-----------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A10_OE | TIOCC9_OE | PO26_OE | PB.DDR.B2 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | x |
| TPU | TIOCC9 输出 | 0 | 1 | x | x |
| PPG | PO26 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PB2 输出 (注1) | 0 | 0 | 0 | 1 |
| | PB2 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(4) PB3/A11/PO27/(TIOCC9)/TIOCD9

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位、端口功能控制寄存器 y (PFCRy) 和 PB.DDR.B3 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|--------|-----------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A11_OE | TIOCD9_OE | PO27_OE | PB.DDR.B3 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | x |
| TPU | TIOCD9 输出 | 0 | 1 | x | x |
| PPG | PO27 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PB3 输出 (注1) | 0 | 0 | 0 | 1 |
| | PB3 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(5) PB4/A12/PO28/TIOCA10

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位、端口功能控制寄存器 y (PFCRy) 和 PB.DDR.B4 的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|--------|------------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A12_OE | TIOCA10_OE | PO28_OE | PB.DDR.B4 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | x |
| TPU | TIOCA10 输出 | 0 | 1 | x | x |
| PPG | PO28 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PB4 输出 (注1) | 0 | 0 | 0 | 1 |
| | PB4 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(6) PB5/A13/PO29/(TIOCA10)/TIOCB10

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位、端口功能控制寄存器 y (PFCRy) 和 PB.DDR.B5 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|--------|------------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A13_OE | TIOCB10_OE | PO29_OE | PB.DDR.B5 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | x |
| TPU | TIOCB10 输出 | 0 | 1 | x | x |
| PPG | PO29 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PB5 输出 (注1) | 0 | 0 | 0 | 1 |
| | PB5 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(7) PB6/A14/PO30/TIOCA11

通过 PPG、TPU 的寄存器，SYSCR0.EXBE 位、端口功能控制寄存器 y (PFCRy) 和 PB.DDR.B6 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|--------|------------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A14_OE | TIOCA11_OE | PO30_OE | PB.DDR.B6 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | x |
| TPU | TIOCA11 输出 | 0 | 1 | x | x |
| PPG | PO30 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PB6 输出 (注1) | 0 | 0 | 0 | 1 |
| | PB6 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(8) PB7/A15/PO31/(TIOCA11)/TIOCB11

通过 PPG、TPU 的寄存器、SYSCR0.EXBE 位、端口功能控制寄存器 y (PFCRy) 和 PB.DDR.B7 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | | | |
|--------|--------------|--------|------------|---------|-----------|
| | | 总线控制 | TPU | PPG | I/O 端口 |
| | | A15_OE | TIOCB11_OE | PO31_OE | PB.DDR.B7 |
| 总线控制 | 地址输出 (注1) | 1 | x | x | x |
| TPU | TIOCB11 输出 | 0 | 1 | x | x |
| PPG | PO31 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PB7 输出 (注1) | 0 | 0 | 0 | 1 |
| | PB7 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

14.3.13 端口 C (PC)

(1) PC0/A16

通过系统控制寄存器 0 (SYSCR0) 的外部总线允许位 (EXBE)、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 PC.DDR.B0 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|--------|-----------|
| | | 总线控制 | I/O 端口 |
| | | A16_OE | PC.DDR.B0 |
| 总线控制 | A16 输出 (注1) | 1 | x |
| I/O 端口 | PC0 输出 | 0 | 1 |
| | PC0 输入 (初始值) | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(2) PC1/A17

通过 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 PC.DDR.B1 位的组合进行以下的转换：

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|--------|-----------|
| | | 总线控制 | I/O 端口 |
| | | A17_OE | PC.DDR.B1 |
| 总线控制 | A17 输出 (注1) | 1 | x |
| I/O 端口 | PC1 输出 | 0 | 1 |
| | PC1 输入 (初始值) | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(3) PC2/A18

通过 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 PC.DDR.B2 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|--------|-----------|
| | | 总线控制 | I/O 端口 |
| | | A18_OE | PC.DDR.B2 |
| 总线控制 | A18 输出 (注1) | 1 | x |
| I/O 端口 | PC2 输出 | 0 | 1 |
| | PC2 输入 (初始值) | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(4) PC3/A19

通过 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 PC.DDR.B3 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|--------|-----------|
| | | 总线控制 | I/O 端口 |
| | | A19_OE | PC.DDR.B3 |
| 总线控制 | A19 输出 (注1) | 1 | x |
| I/O 端口 | PC3 输出 | 0 | 1 |
| | PC3 输入 (初始值) | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(5) PC4/A20

通过 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 PC.DDR.B4 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|--------|-----------|
| | | 总线控制 | I/O 端口 |
| | | A20_OE | PC.DDR.B4 |
| 总线控制 | A20 输出 (注1) | 1 | x |
| I/O 端口 | PC4 输出 | 0 | 1 |
| | PC4 输入 (初始值) | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(6) PC5/A21/SCK5/CS5#-D

通过 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 PC.DDR.B5 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | | | |
|--------|----------------|--------|-----------|---------|-----------|
| | | 总线控制 | | SCI | I/O 端口 |
| | | A21_OE | CS5#-D_OE | SCK5_OE | PC.DDR.B5 |
| 总线控制 | A21 输出 (注1) | 1 | x | x | x |
| | CS5#-D 输出 (注1) | 0 | 1 | x | x |
| SCI | SCK5 输出 | 0 | 0 | 1 | x |
| I/O 端口 | PC5 输出 | 0 | 0 | 0 | 1 |
| | PC5 输入 (初始值) | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(7) PC6/A22/(RxD5)/CS6#-D

通过 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 PC.DDR.B6 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | | |
|--------|----------------|--------|-----------|-----------|
| | | 总线控制 | | I/O 端口 |
| | | A22_OE | CS6#-D_OE | PC.DDR.B6 |
| 总线控制 | A22 输出 (注1) | 1 | x | x |
| | CS6#-D 输出 (注1) | 0 | 1 | x |
| I/O 端口 | PC6 输出 | 0 | 0 | 1 |
| | PC6 输入 (初始值) | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

(8) PC7/A23/TxD5/CS4#-D/CS7#-D

通过 SYSCR0.EXBE 位、总线控制的寄存器、端口功能控制寄存器 y (PFCRy) 和 PC.DDR.B7 位的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | | | | |
|--------|----------------|--------|-----------|-----------|---------|-----------|
| | | 总线控制 | | | SCI | I/O 端口 |
| | | A23_OE | CS4#-D_OE | CS7#-D_OE | TxD5_OE | PC.DDR.B7 |
| 总线控制 | A23 输出 (注1) | 1 | x | x | x | x |
| | CS4#-D 输出 (注1) | 0 | 1 | x | x | x |
| | CS7#-D 输出 (注1) | 0 | 0 | 1 | x | x |
| SCI | TxD5 输出 | 0 | 0 | 0 | 1 | x |
| I/O 端口 | PC7 输出 | 0 | 0 | 0 | 0 | 1 |
| | PC7 输入 (初始值) | 0 | 0 | 0 | 0 | 0 |

注1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

14.3.14 端口 D (PD)

(1) PD0/D0、PD1/D1、PD2/D2、PD3/D3、PD4/D4、PD5/D5、PD6/D6、PD7/D7

通过系统控制寄存器 0 (SYSCR0) 的外部总线允许位 (EXBE) 和 PD.DDR.Bj 位 (j=0 ~ 7) 的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|-----------------|-----------|--|
| | | I/O 端口 | |
| | | PD.DDR.Bj | |
| 总线控制 | 数据输入 / 输出 (注 1) | x | |
| I/O 端口 | PDn 输出 | 1 | |
| | PDn 输入 (初始值) | 0 | |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

14.3.15 端口 E (PE)

(1) PE0/D8、PE1/D9、PE2/D10、PE3/D11、PE4/D12、PE5/D13/(IRQ5-A)、PE6/D14/(IRQ6-A)、PE7/D15/(IRQ7-A)

通过总线模式、SYSCR0.EXBE 位、端口功能控制寄存器 y (PFCRy) 和 PE.DDR.Bj 位 (j=0 ~ 7) 的组合进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|-----------------|----------------|-----------|
| | | 总线控制 | I/O 端口 |
| | | Dn_E(n=8 ~ 15) | PE.DDR.Bj |
| 总线控制 | 数据输入 / 输出 (注 1) | 1 | x |
| I/O 端口 | PEn 输出 | 0 | 1 |
| | PEn 输入 (初始值) | 0 | 0 |

注 1. 在内部 ROM 无效 / 有效扩展模式 (SYSCR0.EXBE=1) 时有效。

当 16 位总线模式为 “1” 时, 为 D15 ~ D8 输入/输出。

当 16 位总线模式为 “0” 时, 为通用输入/输出。

14.3.16 端口 F (PF)

(1) PF0

通过设定 PF.DDR.B0 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 | |
|--------|--------------|-----------|--|
| | | I/O 端口 | |
| | | PF.DDR.B0 | |
| I/O 端口 | PF0 输出 | 1 | |
| | PF0 输入 (初始值) | 0 | |

(2) PF1

通过设定 PF.DDR.B1 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PF.DDR.B1 |
| I/O 端口 | PF1 输出 | 1 |
| | PF1 输入 (初始值) | 0 |

(3) PF2

通过设定 PF.DDR.B2 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PF.DDR.B2 |
| I/O 端口 | PF2 输出 | 1 |
| | PF2 输入 (初始值) | 0 |

(4) PF3

通过设定 PF.DDR.B3 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PF.DDR.B3 |
| I/O 端口 | PF3 输出 | 1 |
| | PF3 输入 (初始值) | 0 |

(5) PF4

通过设定 PF.DDR.B4 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PF.DDR.B4 |
| I/O 端口 | PF4 输出 | 1 |
| | PF4 输入 (初始值) | 0 |

(6) PF5

通过设定 PF.DDR.B5 位进行以下的转换：

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PF.DDR.B5 |
| I/O 端口 | PF5 输出 | 1 |
| | PF5 输入 (初始值) | 0 |

(7) PF6

通过设定 PF.DDR.B6 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PF.DDR.B6 |
| I/O 端口 | PF6 输出 | 1 |
| | PF6 输入 (初始值) | 0 |

14.3.17 端口 G (PG)

(1) PG0

通过设定 PG.DDR.B0 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PG.DDR.B0 |
| I/O 端口 | PG0 输出 | 1 |
| | PG0 输入 (初始值) | 0 |

(2) PG1

通过设定 PG.DDR.B1 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PG.DDR.B1 |
| I/O 端口 | PG1 输出 | 1 |
| | PG1 输入 (初始值) | 0 |

(3) PG2

通过设定 PG.DDR.B2 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PG.DDR.B2 |
| I/O 端口 | PG2 输出 | 1 |
| | PG2 输入 (初始值) | 0 |

(4) PG3

通过设定 PG.DDR.B3 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PG.DDR.B3 |
| I/O 端口 | PG3 输出 | 1 |
| | PG3 输入 (初始值) | 0 |

(5) PG4

通过设定 PG.DDR.B4 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PG.DDR.B4 |
| I/O 端口 | PG4 输出 | 1 |
| | PG4 输入 (初始值) | 0 |

(6) PG5

通过设定 PG.DDR.B5 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PG.DDR.B5 |
| I/O 端口 | PG5 输出 | 1 |
| | PG5 输入 (初始值) | 0 |

(7) PG6

通过设定 PG.DDR.B6 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PG.DDR.B6 |
| I/O 端口 | PG6 输出 | 1 |
| | PG6 输入 (初始值) | 0 |

(8) PG7

通过设定 PG.DDR.B7 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PG.DDR.B7 |
| I/O 端口 | PG7 输出 | 1 |
| | PG7 输入 (初始值) | 0 |

14.3.18 端口 H (PH)

(1) PH0

通过设定 PH.DDR.B0 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PH.DDR.B0 |
| I/O 端口 | PH0 输出 | 1 |
| | PH0 输入 (初始值) | 0 |

(2) PH1

通过设定 PH.DDR.B1 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PH.DDR.B1 |
| I/O 端口 | PH1 输出 | 1 |
| | PH1 输入 (初始值) | 0 |

(3) PH2

通过设定 PH.DDR.B2 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PH.DDR.B2 |
| I/O 端口 | PH2 输出 | 1 |
| | PH2 输入 (初始值) | 0 |

(4) PH3

通过设定 PH.DDR.B3 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PH.DDR.B3 |
| I/O 端口 | PH3 输出 | 1 |
| | PH3 输入 (初始值) | 0 |

(5) PH4

通过设定 PH.DDR.B4 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PH.DDR.B4 |
| I/O 端口 | PH4 输出 | 1 |
| | PH4 输入 (初始值) | 0 |

(6) PH5

通过设定 PH.DDR.B5 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PH.DDR.B5 |
| I/O 端口 | PH5 输出 | 1 |
| | PH5 输入 (初始值) | 0 |

(7) PH6

通过设定 PH.DDR.B6 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PH.DDR.B6 |
| I/O 端口 | PH6 输出 | 1 |
| | PH6 输入 (初始值) | 0 |

(8) PH7

通过设定 PH.DDR.B7 位进行以下的转换:

| 模块名 | 引脚功能 | 设定 |
|--------|--------------|-----------|
| | | I/O 端口 |
| | | PH.DDR.B7 |
| I/O 端口 | PH7 输出 | 1 |
| | PH7 输入 (初始值) | 0 |

14.4 输出信号有效的设定一览表

将各端口的输出功能置为有效的设定一览表如表 14.8 所示。

表 14.8 各端口的输出信号有效的设定一览表 (1/6)

| 端口 | 对应的 外围模块 | 输出设定 信号名 | 输出信号名 | 信号选择 寄存器的设定 | 各内部模块的设定 | |
|----|-------------|-------------|---------|----------------|--|--|
| P0 | 0 | SCI6 | TxD6_OE | TxD6 | SCR.TE=1 | |
| | 1 | | — | — | — | |
| | 2 | TMR2 | TMO2_OE | TMO2 | | TCSR.OSA[1:0] = 01/10/11 或者 TCSR.OSB[1:0]=01/10/11 |
| | | SCI6 | SCK6_OE | SCK6 | | 当 SCMR.SMIF=1 时: SMR.GM=0, SCR.CKE[1:0]=01 或者 SMR.GM=1 SCR.TE=1 或者 SCR.RE=1 当 SCMR.SMIF=0 时: SMR.CM=0, SCR.CKE[1:0]=01 或者 SMR.CM=1, SCR.CKE[1]=0 SCR.TE=1 或者 SCR.RE=1 |
| | 3 | SCI4 | SCK4_OE | SCK4 | | 当 SCMR.SMIF=1 时: SMR.GM=0, SCR.CKE[1:0]=01 或者 SMR.GM=1 SCR.TE=1 或者 SCR.RE=1 当 SCMR.SMIF=0 时: SMR.CM=0, SCR.CKE[1:0]=01 或者 SMR.CM=1, SCR.CKE[1]=0、 SCR.TE=1 或者 SCR.RE=1 |
| | 4 | SCI4 | TxD4_OE | TxD4 | | SCR.TE=1 |
| 5 | TMR3 | TMO3_OE | TMO3 | | TCSR.OSA[1:0]=01/10/11 或者 TCSR.OSB[1:0]=01/10/11 | |
| P1 | 0 | | — | — | — | |
| | 1 | SCI2 | SCK2_OE | SCK2 | | 当 SCMR.SMIF=1 时: SMR.GM=0, SCR.CKE[1:0]=01 或者 SMR.GM=1 SCR.TE=1 或者 SCR.RE=1 当 SCMR.SMIF=0 时: SMR.CM=0, SCR.CKE[1:0]=01 或者 SMR.CM=1, SCR.CKE[1]=0 SCR.TE=1 或者 SCR.RE=1 |
| | 2 | | — | — | — | |
| | 3 | SCI2 | TxD2_OE | TxD2 | | SCR.TE=1 |
| | 4 | RIIC1 | SDA1_OE | SDA1 | | ICCR1.ICE=1 |
| | 5 | SCI3 | SCK3_OE | SCK3 | | 当 SCMR.SMIF=1 时: SMR.GM=0, SCR.CKE[1:0]=01 或者 SMR.GM=1 SCR.TE=1 或者 SCR.RE=1 当 SCMR.SMIF=0 时: SMR.CM=0, SCR.CKE[1:0]=01 或者 SMR.CM=1, SCR.CKE[1]=0 SCR.TE=1 或者 SCR.RE=1 |
| | | RIIC1 | SCL1_OE | SCL1 | | ICCR1.ICE=1 |
| | 6 | RIIC0 | SDA0_OE | SDA0 | | ICCR1.ICE=1 |
| 7 | SCI3 | TxD3_OE | TxD3 | | SCR.TE=1 | |
| | RIIC0 | SCL0_OE | SCL0 | | ICCR1.ICE=1 | |

表 14.8 各端口的输出信号有效的设定一览表 (2/6)

| 端口 | 对应的 外围模块 | 输出设定 信号名 | 输出信号名 | 信号选择 寄存器的设定 | 各内部模块的设定 | |
|------|-------------|-------------|-----------|----------------|---------------|---|
| P2 | 0 | TPU3 | TIOCB3_OE | TIOCB3 | | TIORH.IOB[3]=0, TIORH.IOB[1:0]=01/10/11 |
| | | SCI0 | TxD0_OE | TxD0 | | SCR.TE=1 |
| | | PPG0 | PO0_OE | PO0 | | NDERL.NDER0=1 |
| | 1 | TPU3 | TIOCA3_OE | TIOCA3 | | TIORH.IOA[3]=0, TIORH.IOA[1:0]=01/10/11 |
| | | PPG0 | PO1_OE | PO1 | | NDERL.NDER1=1 |
| | 2 | TPU3 | TIOCC3_OE | TIOCC3 | | TMDR.BFA=0, TIORL.IOC[3]=0, TIORL.IOC[1:0]=01/10/11 |
| | | TMR0 | TMO0_OE | TMO0 | | TCSR.OSA[1:0]=01/10/11 或者 TCSR.OSB[1:0]=01/10/11 |
| | | SCI0 | SCK0_OE | SCK0 | | 当 SCMR.SMIF=1 时: SMR.GM=0, SCR.CKE[1:0]=01 或者 SMR.GM=1 SCR.TE=1 或者 SCR.RE=1 当 SCMR.SMIF=0 时: SMR.CM=0, SCR.CKE[1:0]=01 或者 SMR.CM=1, SCR.CKE[1]=0 SCR.TE=1 或者 SCR.RE=1 |
| | | PPG0 | PO2_OE | PO2 | | NDERL.NDER2=1 |
| | 3 | TPU3 | TIOCD3_OE | TIOCD3 | | TMDR.BFB=0, TIORL.IOD[3]=0, TIORL.IOD[1:0]=01/10/11 |
| | | PPG0 | PO3_OE | PO3 | | NDERL.NDER3=1 |
| | 4 | TPU4 | TIOCB4_OE | TIOCB4 | | TIOR.IOB[3]=0, TIOR.IOB[1:0]=01/10/11 |
| | | PPG0 | PO4_OE | PO4 | | NDERL.NDER4=1 |
| | 5 | TPU4 | TIOCA4_OE | TIOCA4 | | TIOR.IOA[3]=0, TIOR.IOA[1:0]=01/10/11 |
| | | PPG0 | PO5_OE | PO5 | | NDERL.NDER5=1 |
| | 6 | TPU5 | TIOCA5_OE | TIOCA5 | | TIOR.IOA[3]=0, TIOR.IOA[1:0]=01/10/11 |
| | | TMR1 | TMO1_OE | TMO1 | | TCSR.OSA[1:0]=01/10/11 或者 TCSR.OSB[1:0]=01/10/11 |
| | | SCI1 | TxD1_OE | TxD1 | | SCR.TE=1 |
| | | PPG0 | PO6_OE | PO6 | | NDERL.NDER6=1 |
| | 7 | TPU5 | TIOCB5_OE | TIOCB5 | | TIOR.IOB[3]=0, TIOR.IOB[1:0]=01/10/11 |
| | | SCI1 | SCK1_OE | SCK1 | | 当 SCMR.SMIF=1 时: SMR.GM=0, SCR.CKE[1:0]=01 或者 SMR.GM=1 SCR.TE=1 或者 SCR.RE=1 当 SCMR.SMIF=0 时: SMR.CM=0, SCR.CKE[1:0]=01 或者 SMR.CM=1, SCR.CKE[1]=0 SCR.TE=1 或者 SCR.RE=1 |
| PPG0 | | PO7_OE | PO7 | | NDERL.NDER7=1 | |

表 14.8 各端口的输出信号有效的设定一览表 (3/6)

| 端口 | | 对应的外围模块 | 输出设定信号名 | 输出信号名 | 信号选择寄存器的设定 | 各内部模块的设定 |
|----|------|-----------|-----------|--------|---------------------------------------|--|
| P3 | 0 | TPU0 | TIOCA0_OE | TIOCA0 | | TIORH.IOA[3]=0, TIOH.IOA[1:0]=01/10/11 |
| | | PPG0 | PO8_OE | PO8 | | NDERH.NDER8=1 |
| | 1 | TPU0 | TIOCB0_OE | TIOCB0 | | TIORH.IOB[3]=0, TIORH.IOB[1:0]=01/10/11 |
| | | PPG0 | PO9_OE | PO9 | | NDERH.NDER9=1 |
| | 2 | TPU0 | TIOCC0_OE | TIOCC0 | | TMDR.BFA=0, TIORL.IOC[3]=0, TIORL.IOC[1:0]=01/10/11 |
| | | PPG0 | PO10_OE | PO10 | | NDERH.NDER10=1 |
| | 3 | TPU0 | TIOCD0_OE | TIOCD0 | | TMDR.BFB=0, TIORL.IOD[3]=0, TIORL.IOD[1:0]=01/10/11 |
| | | PPG0 | PO11_OE | PO11 | | NDERH.NDER11=1 |
| | 4 | TPU1 | TIOCA1_OE | TIOCA1 | | TIOR.IOA[3]=0, TIOR.IOA[1:0]=01/10/11 |
| | | PPG0 | PO12_OE | PO12 | | NDERH.NDER12=1 |
| | 5 | TPU1 | TIOCB1_OE | TIOCB1 | | TIOR.IOB[3]=0, TIOR.IOB[1:0]=01/10/11 |
| | | PPG0 | PO13_OE | PO13 | | NDERH.NDER13=1 |
| | 6 | TPU2 | TIOCA2_OE | TIOCA2 | | TIOR.IOA[3]=0, TIOR.IOA[1:0]=01/10/11 |
| | | PPG0 | PO14_OE | PO14 | | NDERH.NDER14=1 |
| 7 | TPU2 | TIOCB2_OE | TIOCB2 | | TIOR.IOB[3]=0, TIOR.IOB[1:0]=01/10/11 | |
| | PPG0 | PO15_OE | PO15 | | NDERH.NDER15=1 | |
| P5 | 0 | SYSC,BSC | WR0_OE | WR0 | | SYSCR0.EXBE=1, CSiMOD.WRMOD=0 |
| | | SYSC,BSC | WR_OE | WR | | SYSCR0.EXBE=1, CSiMOD.WRMOD=1 |
| | 1 | SYSC,BSC | WR1_OE | WR1 | | SYSCR0.EXBE=1, PFCR5.WR1BC1E=1, CSiMOD.WRMOD=0, CSiCNT.BSIZE[1:0]=00 |
| | | SYSC,BSC | BC1_OE | BC1 | | SYSCR0.EXBE=1, PFCR5.WR1BC1E=1, CSiMOD.WRMOD=1, CSiCNT.BSIZE[1:0]=00 |
| | 2 | SYSC | RD_OE | RD | | SYSCR0.EXBE=1 |
| | 3 | PORT | BCLK_OE | BCLK | | P5.DDR.B3=1 |
| | 4 | | — | — | — | — |
| 5 | | — | — | — | — | |
| P6 | 0 | SYSC | CS0_OE | CS0 | | SYSCR0.EXBE=1, PFCR0.CS0E=1, CS0CNT.EXENB=1 |
| | | SYSC | CS4A_OE | CS4 | PFCR1.CS4S[1:0]=00 | SYSCR0.EXBE=1, PFCR0.CS4E=1, CS4CNT.EXENB=1 |
| | | SYSC | CS5B_OE | CS5 | PFCR1.CS5S[1:0]=01 | SYSCR0.EXBE=1, PFCR0.CS5E=1, CS5CNT.EXENB=1 |
| | 1 | SYSC | CS1_OE | CS1 | | SYSCR0.EXBE=1, PFCR0.CS1E=1, CS1CNT.EXENB=1 |
| | | SYSC | CS2B_OE | CS2 | PFCR2.CS2S=1 | SYSCR0.EXBE=1, PFCR0.CS2E=1, CS2CNT.EXENB=1 |
| | | SYSC | CS5A_OE | CS5 | PFCR1.CS5S[1:0]=00 | SYSCR0.EXBE=1, PFCR0.CS5E=1, CS5CNT.EXENB=1 |
| | | SYSC | CS6B_OE | CS6 | PFCR1.CS6S[1:0]=01 | SYSCR0.EXBE=1, PFCR0.CS6E=1, CS6CNT.EXENB=1 |
| | | SYSC | CS7B_OE | CS7 | PFCR1.CS7S[1:0]=01 | SYSCR0.EXBE=1, PFCR0.CS7E=1, CS7CNT.EXENB=1 |
| | | SYSC | CS7A_OE | CS7 | PFCR1.CS7S[1:0]=00 | SYSCR0.EXBE=1, PFCR0.CS7E=1, CS7CNT.EXENB=1 |
| | 2 | SYSC | CS2A_OE | CS2 | PFCR2.CS2S=0 | SYSCR0.EXBE=1, PFCR0.CS2E=1, CS2CNT.EXENB=1 |
| | | SYSC | CS6A_OE | CS6 | PFCR1.CS6S[1:0]=00 | SYSCR0.EXBE=1, PFCR0.CS6E=1, CS6CNT.EXENB=1 |
| | 3 | SYSC | CS3A_OE | CS3 | PFCR2.CS3S=0 | SYSCR0.EXBE=1, PFCR0.CS3E=1, CS3CNT.EXENB=1 |
| | | SYSC | CS7A_OE | CS7 | PFCR1.CS7S[1:0]=00 | SYSCR0.EXBE=1, PFCR0.CS7E=1, CS7CNT.EXENB=1 |
| | 4 | SYSC | CS4B_OE | CS4 | PFCR1.CS4S[1:0]=01 | SYSCR0.EXBE=1, PFCR0.CS4E=1, CS4CNT.EXENB=1 |
| 5 | | — | — | — | — | |
| 6 | DAC | DA0_OE | DA0 | | DACR.DAOE0=1 | |
| 7 | DAC | DA1_OE | DA1 | | DACR.DAOE1=1 | |

表 14.8 各端口的输出信号有效的设定一览表 (4/6)

| 端口 | | 对应的外围模块 | 输出设定信号名 | 输出信号名 | 信号选择寄存器的设定 | 各内部模块的设定 |
|------|---|----------|-----------|--------|----------------------------|---|
| P7 | 0 | SYSC | CS3B_OE | CS3 | PFCR2.CS3S=1 | SYSCR0.EXBE=1, PFCR0.CS3E=1, CS3CNT.EXENB=1 |
| | 1 | SYSC | CS4C_OE | CS4 | PFCR1.CS4S[1:0]=10 | SYSCR0.EXBE=1, PFCR0.CS4E=1, CS4CNT.EXENB=1 |
| | | SYSC | CS5C_OE | CS5 | PFCR1.CS5S[1:0]=10 | SYSCR0.EXBE=1, PFCR0.CS5E=1, CS5CNT.EXENB=1 |
| | | SYSC | CS6C_OE | CS6 | PFCR1.CS6S[1:0]=10 | SYSCR0.EXBE=1, PFCR0.CS6E=1, CS6CNT.EXENB=1 |
| | | SYSC | CS7C_OE | CS7 | PFCR1.CS7S[1:0]=10 | SYSCR0.EXBE=1, PFCR0.CS7E=1, CS7CNT.EXENB=1 |
| | 2 | | — | — | — | — |
| | 3 | | — | — | — | — |
| | 4 | | — | — | — | — |
| 5 | | — | — | — | — | |
| PA | 0 | TPU6 | TIOCA6_OE | TIOCA6 | | TIORH.IOA[3]=0, TIORH.IOA[1:0]=01/10/11 |
| | | PPG1 | PO 16_OE | PO16 | | NDERL.NDER16=1 |
| | | SYSC,BSC | BC0_OE | BC0 | | SYSCR0.EXBE=1, CSiMOD.WRMOD=1, PA.DDR.B0=1 |
| | | SYSC,BSC | A0_OE | A0 | | SYSCR0.EXBE=1, CSiMOD.WRMOD=0, PA.DDR.B0=1 |
| | 1 | TPU6 | TIOCB6_OE | TIOCB6 | | TIORH.IOB[3]=0, TIORH.IOB[1:0]=01/10/11 |
| | | PPG1 | PO 17_OE | PO17 | | NDERL.NDER17=1 |
| | | SYSC | A1_OE | A1 | | SYSCR0.EXBE=1, PA.DDR.B1=1 |
| | 2 | TPU6 | TIOCC6_OE | TIOCC6 | | TMDR.BFA=0, TIORL.IOC[3]=0, TIORL.IOC[1:0]=01/10/11 |
| | | PPG1 | PO 18_OE | PO18 | | NDERL.NDER18=1 |
| | | SYSC | A2_OE | A2 | | SYSCR0.EXBE=1, PA.DDR.B2=1 |
| | 3 | TPU6 | TIOCD6_OE | TIOCD6 | | TMDR.BFB=0, TIORL.IOD[3]=0, TIORL.IOD[1:0]=01/10/11 |
| | | PPG1 | PO 19_OE | PO19 | | NDERL.NDER19=1 |
| | | SYSC | A3_OE | A3 | | SYSCR0.EXBE=1, PA.DDR.B3=1 |
| | 4 | TPU7 | TIOCA7_OE | TIOCA7 | | TIOR.IOA[3]=0, TIOR.IOA[1:0]=01/10/11 |
| | | PPG1 | PO 20_OE | PO20 | | NDERL.NDER20=1 |
| | | SYSC | A4_OE | A4 | | SYSCR0.EXBE=1, PA.DDR.B4=1 |
| | 5 | TPU7 | TIOCB7_OE | TIOCB7 | | TIOR.IOB[3]=0, TIOR.IOB[1:0]=01/10/11 |
| | | PPG1 | PO 21_OE | PO21 | | NDERL.NDER21=1 |
| | | SYSC | A5_OE | A5 | | SYSCR0.EXBE=1, PA.DDR.B5=1 |
| | 6 | TPU8 | TIOCA8_OE | TIOCA8 | | TIOR.IOA[3]=0, TIOR.IOA[1:0]=01/10/11 |
| | | PPG1 | PO 22_OE | PO22 | | NDERL.NDER22=1 |
| | | SYSC | A6_OE | A6 | | SYSCR0.EXBE=1, PA.DDR.B6=1 |
| | 7 | TPU8 | TIOCB8_OE | TIOCB8 | | TIOR.IOB[3]=0, TIOR.IOB[1:0]=01/10/11 |
| | | PPG1 | PO 23_OE | PO23 | | NDERL.NDER23=1 |
| SYSC | | A7_OE | A7 | | SYSCR0.EXBE=1, PA.DDR.B7=1 | |

表 14.8 各端口的输出信号有效的设定一览表 (5/6)

| 端口 | 对应的外围模块 | 输出设定信号名 | 输出信号名 | 信号选择寄存器的设定 | 各内部模块的设定 | |
|------|---------|------------|------------|--------------------|---|---|
| PB | 0 | TPU9 | TIOCA9_OE | TIOCA9 | | TIORH.IOA[3]=0, TIORH.IOA[1:0]=01/10/11 |
| | | PPG1 | PO24_OE | PO24 | | NDERH.NDER24=1 |
| | | SYSC | A8_OE | A8 | | SYSCR0.EXBE=1, PFCR4.A08E=1 |
| | 1 | TPU9 | TIOCB9_OE | TIOCB9 | | TIORH.IOB[3]=0, TIORH.IOB[1:0]=01/10/11 |
| | | PPG1 | PO25_OE | PO25 | | NDERH.NDER25=1 |
| | | SYSC | A9_OE | A9 | | SYSCR0.EXBE=1, PFCR4.A09E=1 |
| | 2 | TPU9 | TIOCC9_OE | TIOCC9 | | TMDR.BFA=0, TIORL.IOC[3]=0, TIORL.IOC[1:0]=01/10/11 |
| | | PPG1 | PO26_OE | PO26 | | NDERH.NDER26=1 |
| | | SYSC | A10_OE | A10 | | SYSCR0.EXBE=1, PFCR4.A10E=1 |
| | 3 | TPU9 | TIOCD9_OE | TIOCD9 | | TMDR.BFB=0, TIORL.IOD[3]=0, TIORL.IOD[1:0]=01/10/11 |
| | | PPG1 | PO27_OE | PO27 | | NDERH.NDER27=1 |
| | | SYSC | A11_OE | A11 | | SYSCR0.EXBE=1, PFCR4.A11E=1 |
| | 4 | TPU10 | TIOCA10_OE | TIOCA10 | | TIOR.IOA[3]=0, TIOR.IOA[1:0]=01/10/11 |
| | | PPG1 | PO28_OE | PO28 | | NDERH.NDER28=1 |
| | | SYSC | A12_OE | A12 | | SYSCR0.EXBE=1, PFCR4.A12E=1 |
| | 5 | TPU10 | TIOCB10_OE | TIOCB10 | | TIOR.IOB[3]=0, TIOR.IOB[1:0]=01/10/11 |
| | | PPG1 | PO29_OE | PO29 | | NDERH.NDER29=1 |
| | | SYSC | A13_OE | A13 | | SYSCR0.EXBE=1, PFCR4.A13E=1 |
| | 6 | TPU11 | TIOCA11_OE | TIOCA11 | | TIOR.IOA[3]=0, TIOR.IOA[1:0]=01/10/11 |
| | | PPG1 | PO30_OE | PO30 | | NDERH.NDER30=1 |
| | | SYSC | A14_OE | A14 | | SYSCR0.EXBE=1, PFCR4.A14E=1 |
| 7 | TPU11 | TIOCB11_OE | TIOCB11 | | TIOR.IOB[3]=0, TIOR.IOB[1:0]=01/10/11 | |
| | PPG1 | PO31_OE | PO31 | | NDERH.NDER31=1 | |
| | SYSC | A15_OE | A15 | | SYSCR0.EXBE=1, PFCR4.A15E=1 | |
| PC | 0 | SYSC | A16_OE | A16 | | SYSCR0.EXBE=1, PFCR3.A16E=1 |
| | 1 | SYSC | A17_OE | A17 | | SYSCR0.EXBE=1, PFCR3.A17E=1 |
| | 2 | SYSC | A18_OE | A18 | | SYSCR0.EXBE=1, PFCR3.A18E=1 |
| | 3 | SYSC | A19_OE | A19 | | SYSCR0.EXBE=1, PFCR3.A19E=1 |
| | 4 | SYSC | A20_OE | A20 | | SYSCR0.EXBE=1, PFCR3.A20E=1 |
| | 5 | SYSC | A21_OE | A21 | | SYSCR0.EXBE=1, PFCR3.A21E=1 |
| | | SCI5 | SCK5_OE | SCK5 | | 当 SCMR.SMIF=1 时: SMR.GM=0, SCR.CKE[1:0]=01 或者 SMR.GM=1 SCR.TE=1 或者 SCR.RE=1 当 SCMR.SMIF=0 时: SMR.CM=0, SCR.CKE[1:0]=01 或者 SMR.CM=1, SCR.CKE[1]=0 SCR.TE=1 或者 SCR.RE=1 |
| | | SYSC | CS5D_OE | CS5 | PFCR1.CS5S[1:0]=11 | SYSCR0.EXBE=1, PFCR0.CS5E=1, CS5CNT.EXENB=1 |
| | 6 | SYSC | A22_OE | A22 | | SYSCR0.EXBE=1, PFCR3.A22E=1, PC.DDR.B6=1 |
| | | SYSC | CS6D_OE | CS6 | PFCR1.CS6S[1:0]=11 | SYSCR0.EXBE=1, PFCR0.CS6E=1, CS6CNT.EXENB=1 |
| | 7 | SYSC | A23_OE | A23 | | SYSCR0.EXBE=1, PFCR3.A23E=1, PC.DDR.B7=1 |
| | | SCI5 | TxD5_OE | TxD5 | | SCR.TE=1 |
| | | SYSC | CS4D_OE | CS4 | PFCR1.CS4S[1:0]=11 | SYSCR0.EXBE=1, PFCR0.CS4E=1, CS4CNT.EXENB=1 |
| SYSC | | CS7D_OE | CS7 | PFCR1.CS7S[1:0]=11 | SYSCR0.EXBE=1, PFCR0.CS7E=1, CS7CNT.EXENB=1 | |

表 14.8 各端口的输出信号有效的设定一览表 (6/6)

| 端口 | 对应的 外围模块 | 输出设定 信号名 | 输出信号名 | 信号选择 寄存器的设定 | 各内部模块的设定 |
|----|-------------|-------------|-------|--|--|
| PD | 0 | SYSC | D0_E | D0 | SYSCR0.EXBE=1 |
| | 1 | SYSC | D1_E | D1 | SYSCR0.EXBE=1 |
| | 2 | SYSC | D2_E | D2 | SYSCR0.EXBE=1 |
| | 3 | SYSC | D3_E | D3 | SYSCR0.EXBE=1 |
| | 4 | SYSC | D4_E | D4 | SYSCR0.EXBE=1 |
| | 5 | SYSC | D5_E | D5 | SYSCR0.EXBE=1 |
| | 6 | SYSC | D6_E | D6 | SYSCR0.EXBE=1 |
| PE | 0 | SYSC,BSC | D8_E | D8 | SYSCR0.EXBE=1, CSICNT.BSIZE[1:0]=00, PFCR5.DHE=1 |
| | 1 | SYSC,BSC | D9_E | D9 | SYSCR0.EXBE=1, CSICNT.BSIZE[1:0]=00, PFCR5.DHE=1 |
| | 2 | SYSC,BSC | D10_E | D10 | SYSCR0.EXBE=1, CSICNT.BSIZE[1:0]=00, PFCR5.DHE=1 |
| | 3 | SYSC,BSC | D11_E | D11 | SYSCR0.EXBE=1, CSICNT.BSIZE[1:0]=00, PFCR5.DHE=1 |
| | 4 | SYSC,BSC | D12_E | D12 | SYSCR0.EXBE=1, CSICNT.BSIZE[1:0]=00, PFCR5.DHE=1 |
| | 5 | SYSC,BSC | D13_E | D13 | SYSCR0.EXBE=1, CSICNT.BSIZE[1:0]=00, PFCR5.DHE=1 |
| | 6 | SYSC,BSC | D14_E | D14 | SYSCR0.EXBE=1, CSICNT.BSIZE[1:0]=00, PFCR5.DHE=1 |
| 7 | SYSC,BSC | D15_E | D15 | SYSCR0.EXBE=1, CSICNT.BSIZE[1:0]=00, PFCR5.DHE=1 | |

表 14.9 各运行模式中的引脚功能

| 运行模式的选择 | 运行模式 | P5 | | | | P6 | | P7 | PA | PB | PC | | PD | PE | PF | PG | PH |
|---------------|---------------|-----|------|-----|-------|-------|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | | b7 | b3 | b2 | b1-b0 | b4-b1 | b0 | b1-b0 | b7-b0 | b7-b0 | b7-b5 | b4-b0 | b7-b0 | b7-b0 | b6-b0 | b7-b0 | b7-b0 |
| 通过模式引脚 的选择 | 引导模式 | P | P*/C | P | P | P | P | P | P | P | P | P | P | P | P | P | P |
| | 用户引导模式 | P | P*/C | P | P | P | P | P | P | P | P | P | P | P | P | P | P |
| | 单芯片模式 | P | P*/C | P | P | P | P | P | P | P | P | P | P | P | P | P | P |
| 寄存器的转移 | 单芯片模式 | P | P/C | P | P | P | P | P | P | P | P | P | P | P | P | P | P |
| | 内部 ROM 有效扩展模式 | P/C | P/C | P/C | P/C | P/C | P/C | P/C | A | P/A | P/A/C | P/A | P/D | P/D | P | P | P |
| | 内部 ROM 无效扩展模式 | P/C | P/C | P/C | P/C | P/C | P/C | P/C | A | P/A | P/A/C | P/A | P/D | P/D | P | P | P |

P: 输入 / 输出端口 A: 地址总线输出 D: 数据总线输出 C: 控制信号或者时钟输入 / 输出

各内部模块的设定 *: 刚复位后

14.5 未使用引脚的处理

未使用引脚的处理内容如表 14.10 所示。

表 14.10 未使用引脚的处理内容

| 引脚名 | 内部 ROM 无效扩展模式 (总线宽度 16 位) | 内部 ROM 无效扩展模式 (总线宽度 8 位) | 内部 ROM 有效扩展模式 | 单芯片模式 |
|---|--|---|---------------|--|
| EMLE | 通过电阻连接 VSS (下拉)。 | | | |
| MD1 ~ MD0 | (用作模式引脚) | | | |
| MDE | (用作模式引脚) | | | |
| NMI | 通过电阻连接 VCC (上拉)。 | | | |
| EXTAL | (用作时钟引脚) | | | |
| XTAL | 将引脚置为开路。 | | | |
| WDTOVF# | 将引脚置为开路。 | | | |
| 端口 0 端口 1 端口 2 端口 3 端口 4 P57 ~ P54 P67 ~ P61 端口 7 端口 8 端口 9 PC7 ~ PC5 | <ul style="list-style-type: none"> 每个引脚通过电阻连接 VCC (上拉) 或者 VSS (下拉)。 也能在 Pm.ICR 为初始值 (输入缓冲器无效) 的状态下将引脚置为开路 (注 1)。 | | | |
| P53 | <ul style="list-style-type: none"> 每个引脚通过电阻连接 VCC (上拉) 或者 VSS (下拉)。 也能在 Pm.ICR 为初始值 (输入缓冲器无效) 的状态下将引脚置为开路 (注 1)。 | | | <ul style="list-style-type: none"> 每个引脚通过电阻连接 VCC (上拉) 或者 VSS (下拉)。 也能在 RPm.IC 为初始值 (输入缓冲器无效) 的状态下将引脚置为开路 (注 1)。 |
| P52 | 因为 RD# 输出, 所以将引脚置为开路。 | | | |
| P51 | <ul style="list-style-type: none"> 每个引脚通过电阻连接 VCC (上拉) 或者 VSS (下拉)。 也能在 Pm.ICR 为初始值 (输入缓冲器无效) 的状态下将引脚置为开路 (注 1)。 | | | |
| P50 | 因为 WR0#/WR# 输出, 所以将引脚置为开路。 | | | |
| P60 端口 A 端口 B PC4 ~ PC0 | <ul style="list-style-type: none"> 每个引脚通过电阻连接 VCC (上拉) 或者 VSS (下拉)。 也能在 Pm.ICR 为初始值 (输入缓冲器无效) 的状态下将引脚置为开路 (注 1)。 | | | |
| 端口 D | (用作数据总线) | | | |
| 端口 E | (用作数据总线) | <ul style="list-style-type: none"> 在初始状态下为通用输入, 因此每个引脚通过电阻连接 VCC (上拉) 或者 VSS (下拉)。 也能在 Pm.ICR 为初始值 (输入缓冲器无效) 的状态下将引脚置为开路 (注 1)。 | | |
| 端口 F | <ul style="list-style-type: none"> 每个引脚通过电阻连接 VCC (上拉) 或者 VSS (下拉)。 也能在 Pm.ICR 为初始值 (输入缓冲器无效) 的状态下将引脚置为开路 (注 1)。 | | | |
| 端口 G | | | | |
| 端口 H | | | | |
| VREFH | 连接 AVCC。 | | | |

注 1. 不能改变 Pm.ICR 寄存器的初始值, 否则就可能产生穿透电流流过。

14.6 使用时的注意事项

14.6.1 输入缓冲控制寄存器 (Pm.ICR) 的设定

在更改 Pm.ICR 寄存器的设定时，有可能因引脚的状态而在内部产生边沿，导致意想不到的运行。因此，必须在将引脚固定为 High 电平的状态下或者在通过设定被分配到该引脚的外围模块将输入功能置为无效的状态下更改 Pm.ICR 寄存器的设定。

如果通过设定 Pm.ICR 寄存器将输入置为有效，对于分配了多个输入功能的引脚，就将引脚状态反映到全部的输入功能。对于未使用的输入功能，必须注意各外围模块的设定。

在将引脚用作输出引脚时，如果通过设定 Pm.ICR 寄存器将输入置为有效，输出数据就作为引脚状态被取入。必须通过设定 Pm.ICR 寄存器将用作输出的引脚输入置为无效。

14.6.2 端口功能控制寄存器 (PFCRm) 的设定

PFCRm 寄存器控制 I/O 端口。在设定各引脚的输入 / 输出时，必须在选择输入 / 输出目标后将输入 / 输出置为有效。

在更改为输入引脚时，如果更改前的引脚电平和更改后的引脚电平不同，就有可能在内部产生边沿，导致意想不到的运行。在更改为输入引脚时，必须按照以下步骤进行设定：

1. 通过设定与要更改的引脚功能对应的外围模块，将输入功能置为无效。
2. 通过设定 PFCRm 寄存器，选择输入引脚。
3. 通过设定与要更改的引脚功能对应的外围模块，将输入功能置为有效。

对于 1 个引脚功能，如果有更改输入 / 输出目标的引脚选择位以及将引脚功能置为有效的允许位，必须在通过引脚选择位设定引脚的输入 / 输出目标后，通过允许位将引脚功能置为有效。

14.6.3 使用 A/D 转换器输入时的端口设定

即使在端口 4 和端口 9 的引脚中只有 1 个引脚用作 A/D 转换器的输入，也必须将 P4.DDR.Bj 和 P9.DDR.Bj 置“0”，将不用作 A/D 转换器输入的端口 4 和端口 9 的引脚设定为输入引脚或者中断输入。

15. 16 位定时器脉冲单元 (TPU)

RX610 群内置由 6 个通道的 16 位定时器构成的 16 位定时器脉冲单元 (TPU) 2 个 (单元 0 和单元 1)，共计 12 个通道 (TPU0 ~ TPU11)。

15.1 概要

TPU 的规格如表 15.1 所示，TPU (单元 0) 和 TPU (单元 1) 的功能一览表分别如表 15.2 和表 15.3 所示。

TPU (单元 0) 和 TPU (单元 1) 的框图分别图 15.1 和图 15.2 所示。

表 15.1 TPU 的规格

| 项目 | 内容 |
|---------------------|--|
| 脉冲输入 / 输出 | 最多 16 个 |
| 计数时钟 | 各通道有 7 种或者 8 种。 |
| 能设定的运行 | <ul style="list-style-type: none"> • 通过比较匹配进行波形输出。 • 输入捕捉功能 • 计数器清除运行 • 同时写多个定时器的计数器 (TCNT)。 • 通过比较匹配 / 输入捕捉进行同时清除。 • 通过计数器的同步运行进行各寄存器的同步输入 / 输出。 • 通过和同步运行的组合进行最多 15 相的 PWM 输出。 • 级联运行。 |
| 通道 0 和通道 3 | 能设定缓冲运行。 |
| 通道 1、通道 2、通道 4、通道 5 | 能分别设定相位计数模式。 |
| 中断源 | 26 种 |
| 缓冲运行 | 寄存器数据的自动传送 |
| 触发生成 | 能生成可编程脉冲生成器 (PPG) 的输出触发。 |
| | 能生成 A/D 转换器的转换开始触发。 |
| 低功耗功能 | 各单元能设定为模块停止状态。 |

表 15.2 TPU (单元 0) 的功能一览表 (1/2)

| 项目 | TPU0 | TPU1 | TPU2 | TPU3 | TPU4 | TPU5 |
|-------------------------------|--|--|--|---|---|---|
| 计数时钟 | PCLK/1 PCLK/4 PCLK/16 PCLK/64 TCLKA TCLKB TCLKC TCLKD | PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKB | PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKB TCLKC | PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 PCLK/4096 TCLKA | PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKC | PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKC TCLKD |
| 定时器的通用寄存器 (TGRy) (y=A ~ D) | TGRA TGRB TGRC (注1) TGRD (注1) | TGRA TGRB | TGRA TGRB | TGRA TGRB TGRC (注1) TGRD (注1) | TGRA TGRB | TGRA TGRB |
| 输入 / 输出引脚 | TIOCA0 TIOCB0 TIOCC0 TIOCD0 | TIOCA1 TIOCB1 | TIOCA2 TIOCB2 | TIOCA3 TIOCB3 TIOCC3 TIOCD3 | TIOCA4 TIOCB4 | TIOCA5 TIOCB5 |
| 计数器清除功能 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 |
| 比较匹配 输出 | Low 电平 输出 | ○ | ○ | ○ | ○ | ○ |
| | High 电平 输出 | ○ | ○ | ○ | ○ | ○ |
| | 交替输出 | ○ | ○ | ○ | ○ | ○ |
| 输入捕捉功能 | ○ | ○ | ○ | ○ | ○ | ○ |
| 同步运行 | ○ | ○ | ○ | ○ | ○ | ○ |
| PWM 模式 | ○ | ○ | ○ | ○ | ○ | ○ |
| 相位计数模式 | — | ○ | ○ | — | ○ | ○ |
| 缓冲运行 | ○ | — | — | ○ | — | — |
| DTC 的启动 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 |

○: 能

—: 不能

注 1. 能将 TGRC 寄存器和 TGRD 寄存器设定为缓冲寄存器。

表 15.2 TPU (单元 0) 的功能一览表 (2/2)

| 项目 | TPU0 | TPU1 | TPU2 | TPU3 | TPU4 | TPU5 |
|---------------|--|--|--|--|--|--|
| DMAC 的启动 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 |
| A/D 转换开始触发 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 |
| | TGRA ~ TGRD 的比较匹配或者输入捕捉 | — | — | — | — | — |
| PPG 触发 | TGRA、TGRB 的比较匹配或者输入捕捉 | TGRA、TGRB 的比较匹配或者输入捕捉 | TGRA、TGRB 的比较匹配或者输入捕捉 | TGRA、TGRB 的比较匹配或者输入捕捉 | — | — |
| 中断源 | 5 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 0A • 比较匹配 / 输入捕捉 0B • 比较匹配 / 输入捕捉 0C • 比较匹配 / 输入捕捉 0D • 上溢 | 4 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 1A • 比较匹配 / 输入捕捉 1B • 上溢 • 下溢 | 4 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 2A • 比较匹配 / 输入捕捉 2B • 上溢 • 下溢 | 5 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 3A • 比较匹配 / 输入捕捉 3B • 比较匹配 / 输入捕捉 3C • 比较匹配 / 输入捕捉 3D • 上溢 | 4 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 4A • 比较匹配 / 输入捕捉 4B • 上溢 • 下溢 | 4 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 5A • 比较匹配 / 输入捕捉 5B • 上溢 • 下溢 |
| 模块停止的设定 (注 2) | MSTPCRA.MSTPA13 位 | | | | | |

注 2. 详细内容请参照“8. 低功耗功能”。

表 15.3 TPU (单元 1) 的功能一览表 (1/2)

| 项目 | TPU6 | TPU7 | TPU8 | TPU9 | TPU10 | TPU11 |
|-------------------------------|--|--|--|---|---|---|
| 计数时钟 | PCLK/1 PCLK/4 PCLK/16 PCLK/64 TCLKE TCLKF TCLKG TCLKH | PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKE TCLKF | PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKE TCLKF TCLKG | PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 PCLK/4096 TCLKE | PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKE TCLKG | PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKE TCLKG TCLKH |
| 定时器的通用寄存器 (TGRy) (y=A ~ D) | TGRA TGRB TGRC (注1) TGRD (注1) | TGRA TGRB | TGRA TGRB | TGRA TGRB TGRC (注1) TGRD (注1) | TGRA TGRB | TGRA TGRB |
| 输入 / 输出引脚 | TIOCA6 TIOCB6 TIOCC6 TIOCD6 | TIOCA7 TIOCB7 | TIOCA8 TIOCB8 | TIOCA9 TIOCB9 TIOCC9 TIOCD9 | TIOCA10 TIOCB10 | TIOCA11 TIOCB11 |
| 计数器清除功能 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 |
| 比较匹配 输出 | Low 电平 输出 | ○ | ○ | ○ | ○ | ○ |
| | High 电平 输出 | ○ | ○ | ○ | ○ | ○ |
| | 交替输出 | ○ | ○ | ○ | ○ | ○ |
| 输入捕捉功能 | ○ | ○ | ○ | ○ | ○ | ○ |
| 同步运行 | ○ | ○ | ○ | ○ | ○ | ○ |
| PWM 模式 | ○ | ○ | ○ | ○ | ○ | ○ |
| 相位计数模式 | — | ○ | ○ | — | ○ | ○ |
| 缓冲运行 | ○ | — | — | ○ | — | — |
| DTC 的启动 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 | TGRy 的比较 匹配或者输入 捕捉 |

○: 能

—: 不能

注 1. 能将 TGRC 寄存器和 TGRD 寄存器设定为缓冲寄存器。

表 15.3 TPU (单元 1) 的功能一览表 (2/2)

| 项目 | TPU6 | TPU7 | TPU8 | TPU9 | TPU10 | TPU11 |
|--------------|--|--|--|--|--|--|
| DMAC 的启动 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 |
| A/D 转换开始触发 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 | TGRA 的比较匹配或者输入捕捉 |
| PPG 触发 | TGRA、TGRB 的比较匹配 | TGRA、TGRB 的比较匹配 | TGRA、TGRB 的比较匹配 | TGRA、TGRB 的比较匹配 | — | — |
| 中断源 | 5 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 6A • 比较匹配 / 输入捕捉 6B • 比较匹配 / 输入捕捉 6C • 比较匹配 / 输入捕捉 6D • 上溢 | 4 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 7A • 比较匹配 / 输入捕捉 7B • 上溢 • 下溢 | 4 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 8A • 比较匹配 / 输入捕捉 8B • 上溢 • 下溢 | 5 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 9A • 比较匹配 / 输入捕捉 9B • 比较匹配 / 输入捕捉 9C • 比较匹配 / 输入捕捉 9D • 上溢 | 4 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 10A • 比较匹配 / 输入捕捉 10B • 上溢 • 下溢 | 4 个源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 11A • 比较匹配 / 输入捕捉 11B • 上溢 • 下溢 |
| 模块停止的设定 (注2) | MSTPCRA.MSTPA12 位 | | | | | |

注 2. 详细内容请参照“8. 低功耗功能”。

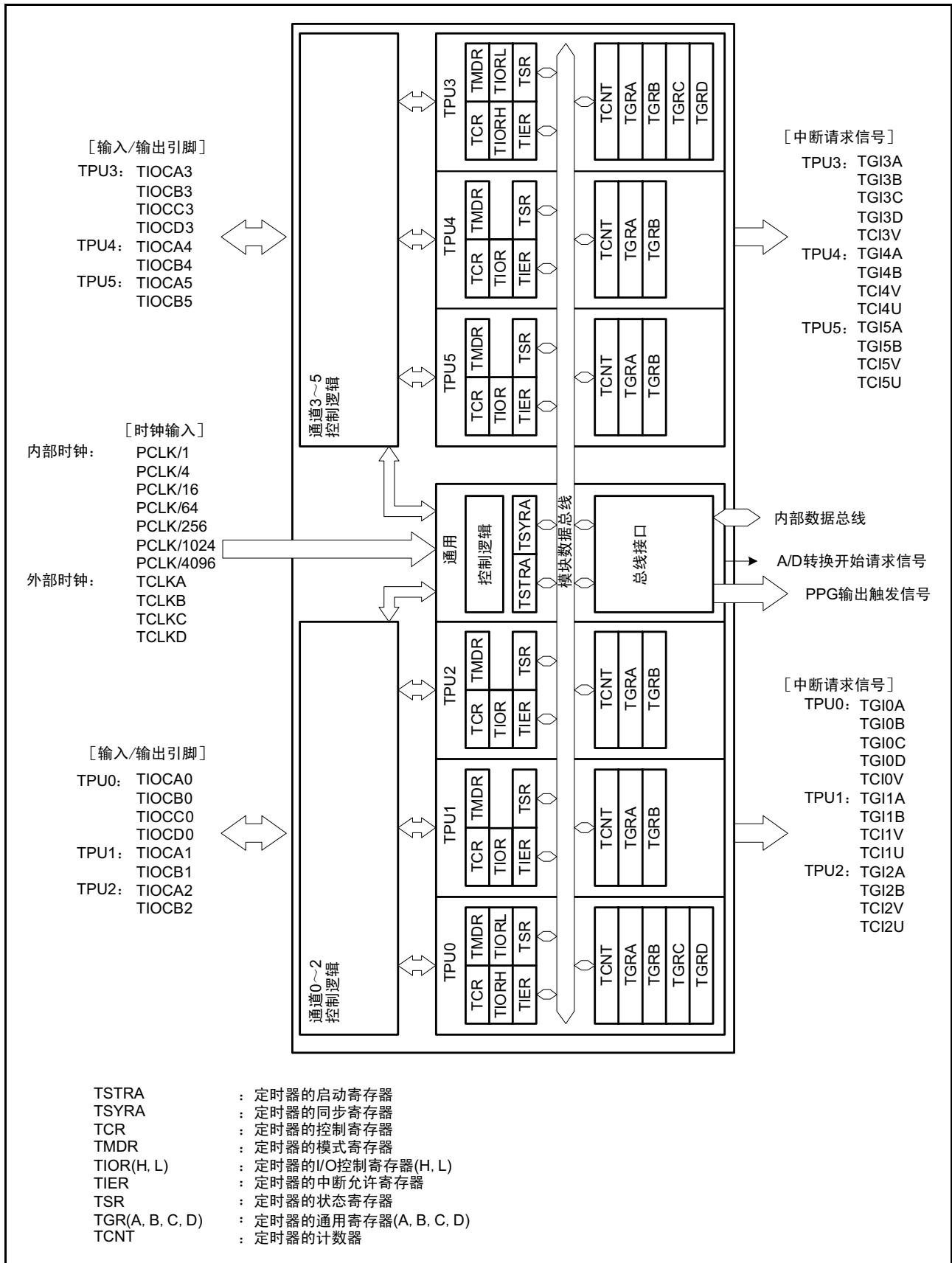


图 15.1 TPU (单元 0) 的框图

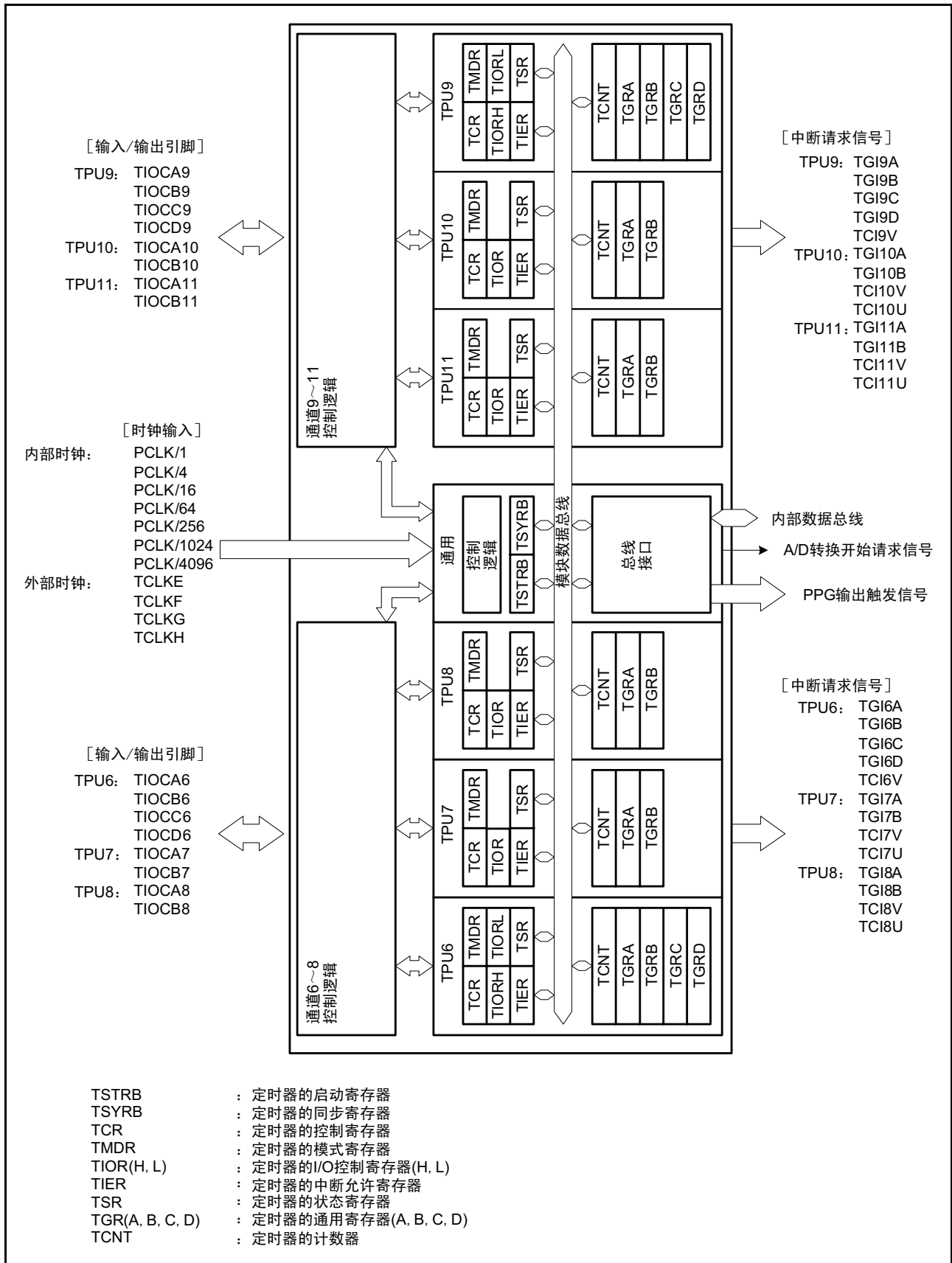


图 15.2 TPU (单元 1) 的框图

TPU 使用的输入 / 输出引脚如表 15.4 所示。

表 15.4 TPU 的输入 / 输出引脚

| 单元 | 通道 | 引脚名 | 输入 / 输出 | 功能 |
|------|-------|---------|---------|---|
| 单元 0 | 通用 | TCLKA | 输入 | 外部时钟 A 的输入引脚 (TPU1 和 TPU5 的相位计数模式 A 相输入) |
| | | TCLKB | 输入 | 外部时钟 B 的输入引脚 (TPU1 和 TPU5 的相位计数模式 B 相输入) |
| | | TCLKC | 输入 | 外部时钟 C 的输入引脚 (TPU2 和 TPU4 的相位计数模式 A 相输入) |
| | | TCLKD | 输入 | 外部时钟 D 的输入引脚 (TPU2 和 TPU4 的相位计数模式 B 相输入) |
| | TPU0 | TIOCA0 | 输入 / 输出 | TPU0.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB0 | 输入 / 输出 | TPU0.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCC0 | 输入 / 输出 | TPU0.TGRC 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCD0 | 输入 / 输出 | TPU0.TGRD 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | TPU1 | TIOCA1 | 输入 / 输出 | TPU1.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB1 | 输入 / 输出 | TPU1.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | TPU2 | TIOCA2 | 输入 / 输出 | TPU2.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB2 | 输入 / 输出 | TPU2.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | TPU3 | TIOCA3 | 输入 / 输出 | TPU3.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB3 | 输入 / 输出 | TPU3.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCC3 | 输入 / 输出 | TPU3.TGRC 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCD3 | 输入 / 输出 | TPU3.TGRD 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | TPU4 | TIOCA4 | 输入 / 输出 | TPU4.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB4 | 输入 / 输出 | TPU4.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | TPU5 | TIOCA5 | 输入 / 输出 | TPU5.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB5 | 输入 / 输出 | TPU5.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| 单元 1 | 通用 | TCLKE | 输入 | 外部时钟 E 的输入引脚 (TPU7 和 TPU11 的相位计数模式 A 相输入) |
| | | TCLKF | 输入 | 外部时钟 F 的输入引脚 (TPU7 和 TPU11 的相位计数模式 B 相输入) |
| | | TCLKG | 输入 | 外部时钟 G 的输入引脚 (TPU8 和 TPU10 的相位计数模式 A 相输入) |
| | | TCLKH | 输入 | 外部时钟 H 的输入引脚 (TPU8 和 TPU10 的相位计数模式 B 相输入) |
| | TPU6 | TIOCA6 | 输入 / 输出 | TPU6.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB6 | 输入 / 输出 | TPU6.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCC6 | 输入 / 输出 | TPU6.TGRC 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCD6 | 输入 / 输出 | TPU6.TGRD 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | TPU7 | TIOCA7 | 输入 / 输出 | TPU7.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB7 | 输入 / 输出 | TPU7.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | TPU8 | TIOCA8 | 输入 / 输出 | TPU8.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB8 | 输入 / 输出 | TPU8.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | TPU9 | TIOCA9 | 输入 / 输出 | TPU9.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB9 | 输入 / 输出 | TPU9.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCC9 | 输入 / 输出 | TPU9.TGRC 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCD9 | 输入 / 输出 | TPU9.TGRD 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | TPU10 | TIOCA10 | 输入 / 输出 | TPU10.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB10 | 输入 / 输出 | TPU10.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | TPU11 | TIOCA11 | 输入 / 输出 | TPU11.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |
| | | TIOCB11 | 输入 / 输出 | TPU11.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚 |

15.2 寄存器说明

TPU 的寄存器一览表如表 15.5 所示。

表 15.5 TPU 的寄存器一览表 (1/3)

| 单元 | 通道 | 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|------|------|------------------|-------|-------|------------|------|
| 单元 0 | TPU0 | 定时器的控制寄存器 | TCR | 00h | 0008 8110h | 8 |
| | | 定时器的模式寄存器 | TMDR | 00h | 0008 8111h | 8 |
| | | 定时器的 I/O 控制寄存器 H | TIORH | 00h | 0008 8112h | 8 |
| | | 定时器的 I/O 控制寄存器 L | TIORL | 00h | 0008 8113h | 8 |
| | | 定时器的中断允许寄存器 | TIER | 40h | 0008 8114h | 8 |
| | | 定时器的状态寄存器 | TSR | xxh | 0008 8115h | 8 |
| | | 定时器的计数器 | TCNT | 0000h | 0008 8116h | 16 |
| | | 定时器的通用寄存器 A | TGRA | FFFFh | 0008 8118h | 16 |
| | | 定时器的通用寄存器 B | TGRB | FFFFh | 0008 811Ah | 16 |
| | | 定时器的通用寄存器 C | TGRC | FFFFh | 0008 811Ch | 16 |
| | | 定时器的通用寄存器 D | TGRD | FFFFh | 0008 811Eh | 16 |
| | TPU1 | 定时器的控制寄存器 | TCR | 00h | 0008 8120h | 8 |
| | | 定时器的模式寄存器 | TMDR | 00h | 0008 8121h | 8 |
| | | 定时器的 I/O 控制寄存器 | TIOR | 00h | 0008 8122h | 8 |
| | | 定时器的中断允许寄存器 | TIER | 40h | 0008 8124h | 8 |
| | | 定时器的状态寄存器 | TSR | xxh | 0008 8125h | 8 |
| | | 定时器的计数器 | TCNT | 0000h | 0008 8126h | 16 |
| | | 定时器的通用寄存器 A | TGRA | FFFFh | 0008 8128h | 16 |
| | | 定时器的通用寄存器 B | TGRB | FFFFh | 0008 812Ah | 16 |
| | TPU2 | 定时器的控制寄存器 | TCR | 00h | 0008 8130h | 8 |
| | | 定时器的模式寄存器 | TMDR | 00h | 0008 8131h | 8 |
| | | 定时器的 I/O 控制寄存器 | TIOR | 00h | 0008 8132h | 8 |
| | | 定时器的中断允许寄存器 | TIER | 40h | 0008 8134h | 8 |
| | | 定时器的状态寄存器 | TSR | xxh | 0008 8135h | 8 |
| | | 定时器的计数器 | TCNT | 0000h | 0008 8136h | 16 |
| | | 定时器的通用寄存器 A | TGRA | FFFFh | 0008 8138h | 16 |
| | | 定时器的通用寄存器 B | TGRB | FFFFh | 0008 813Ah | 16 |
| | TPU3 | 定时器的控制寄存器 | TCR | 00h | 0008 8140h | 8 |
| | | 定时器的模式寄存器 | TMDR | 00h | 0008 8141h | 8 |
| | | 定时器的 I/O 控制寄存器 H | TIORH | 00h | 0008 8142h | 8 |
| | | 定时器的 I/O 控制寄存器 L | TIORL | 00h | 0008 8143h | 8 |
| | | 定时器的中断允许寄存器 | TIER | 40h | 0008 8144h | 8 |
| | | 定时器的状态寄存器 | TSR | xxh | 0008 8145h | 8 |
| | | 定时器的计数器 | TCNT | 0000h | 0008 8146h | 16 |
| | | 定时器的通用寄存器 A | TGRA | FFFFh | 0008 8148h | 16 |
| | | 定时器的通用寄存器 B | TGRB | FFFFh | 0008 814Ah | 16 |
| | | 定时器的通用寄存器 C | TGRC | FFFFh | 0008 814Ch | 16 |
| | | 定时器的通用寄存器 D | TGRD | FFFFh | 0008 814Eh | 16 |

表 15.5 TPU 的寄存器一览表 (2/3)

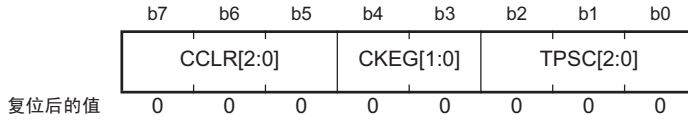
| 单元 | 通道 | 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 | |
|------------------|------|----------------|-----------|-------|------------|------------|---|
| 单元 0 | TPU4 | 定时器的控制寄存器 | TCR | 00h | 0008 8150h | 8 | |
| | | 定时器的模式寄存器 | TMDR | 00h | 0008 8151h | 8 | |
| | | 定时器的 I/O 控制寄存器 | TIOR | 00h | 0008 8152h | 8 | |
| | | 定时器的中断允许寄存器 | TIER | 40h | 0008 8154h | 8 | |
| | | 定时器的状态寄存器 | TSR | xxh | 0008 8155h | 8 | |
| | | 定时器的计数器 | TCNT | 0000h | 0008 8156h | 16 | |
| | | 定时器的通用寄存器 A | TGRA | FFFFh | 0008 8158h | 16 | |
| | | 定时器的通用寄存器 B | TGRB | FFFFh | 0008 815Ah | 16 | |
| | TPU5 | 定时器的控制寄存器 | TCR | 00h | 0008 8160h | 8 | |
| | | 定时器的模式寄存器 | TMDR | 00h | 0008 8161h | 8 | |
| | | 定时器的 I/O 控制寄存器 | TIOR | 00h | 0008 8162h | 8 | |
| | | 定时器的中断允许寄存器 | TIER | 40h | 0008 8164h | 8 | |
| | | 定时器的状态寄存器 | TSR | xxh | 0008 8165h | 8 | |
| | | 定时器的计数器 | TCNT | 0000h | 0008 8166h | 16 | |
| | | 定时器的通用寄存器 A | TGRA | FFFFh | 0008 8168h | 16 | |
| | | 定时器的通用寄存器 B | TGRB | FFFFh | 0008 816Ah | 16 | |
| | 通用 | 定时器的启动寄存器 | TSTRA | 00h | 0008 8100h | 8 | |
| | | 定时器的同步寄存器 | TSYRA | 00h | 0008 8101h | 8 | |
| | 单元 1 | TPU6 | 定时器的控制寄存器 | TCR | 00h | 0008 8180h | 8 |
| | | | 定时器的模式寄存器 | TMDR | 00h | 0008 8181h | 8 |
| 定时器的 I/O 控制寄存器 H | | | TIORH | 00h | 0008 8182h | 8 | |
| 定时器的 I/O 控制寄存器 L | | | TIORL | 00h | 0008 8183h | 8 | |
| 定时器的中断允许寄存器 | | | TIER | 40h | 0008 8184h | 8 | |
| 定时器的状态寄存器 | | | TSR | xxh | 0008 8185h | 8 | |
| 定时器的计数器 | | | TCNT | 0000h | 0008 8186h | 16 | |
| 定时器的通用寄存器 A | | | TGRA | FFFFh | 0008 8188h | 16 | |
| 定时器的通用寄存器 B | | | TGRB | FFFFh | 0008 818Ah | 16 | |
| 定时器的通用寄存器 C | | | TGRC | FFFFh | 0008 818Ch | 16 | |
| 定时器的通用寄存器 D | | | TGRD | FFFFh | 0008 818Eh | 16 | |
| TPU7 | | | 定时器的控制寄存器 | TCR | 00h | 0008 8190h | 8 |
| | | 定时器的模式寄存器 | TMDR | 00h | 0008 8191h | 8 | |
| | | 定时器的 I/O 控制寄存器 | TIOR | 00h | 0008 8192h | 8 | |
| | | 定时器的中断允许寄存器 | TIER | 40h | 0008 8194h | 8 | |
| | | 定时器的状态寄存器 | TSR | xxh | 0008 8195h | 8 | |
| | | 定时器的计数器 | TCNT | 0000h | 0008 8196h | 16 | |
| | | 定时器的通用寄存器 A | TGRA | FFFFh | 0008 8198h | 16 | |
| | | 定时器的通用寄存器 B | TGRB | FFFFh | 0008 819Ah | 16 | |

表 15.5 TPU 的寄存器一览表 (3/3)

| 单元 | 通道 | 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|-------------|----------------|------------------|-----------|------------|------------|------------|
| 单元 1 | TPU8 | 定时器的控制寄存器 | TCR | 00h | 0008 81A0h | 8 |
| | | 定时器的模式寄存器 | TMDR | 00h | 0008 81A1h | 8 |
| | | 定时器的 I/O 控制寄存器 | TIOR | 00h | 0008 81A2h | 8 |
| | | 定时器的中断允许寄存器 | TIER | 40h | 0008 81A4h | 8 |
| | | 定时器的状态寄存器 | TSR | xxh | 0008 81A5h | 8 |
| | | 定时器的计数器 | TCNT | 0000h | 0008 81A6h | 16 |
| | | 定时器的通用寄存器 A | TGRA | FFFFh | 0008 81A8h | 16 |
| | | 定时器的通用寄存器 B | TGRB | FFFFh | 0008 81AAh | 16 |
| | TPU9 | 定时器的控制寄存器 | TCR | 00h | 0008 81B0h | 8 |
| | | 定时器的模式寄存器 | TMDR | 00h | 0008 81B1h | 8 |
| | | 定时器的 I/O 控制寄存器 H | TIORH | 00h | 0008 81B2h | 8 |
| | | 定时器的 I/O 控制寄存器 L | TIORL | 00h | 0008 81B3h | 8 |
| | | 定时器的中断允许寄存器 | TIER | 40h | 0008 81B4h | 8 |
| | | 定时器的状态寄存器 | TSR | xxh | 0008 81B5h | 8 |
| | | 定时器的计数器 | TCNT | 0000h | 0008 81B6h | 16 |
| | | 定时器的通用寄存器 A | TGRA | FFFFh | 0008 81B8h | 16 |
| | | 定时器的通用寄存器 B | TGRB | FFFFh | 0008 81BAh | 16 |
| | | 定时器的通用寄存器 C | TGRC | FFFFh | 0008 81BCh | 16 |
| | | 定时器的通用寄存器 D | TGRD | FFFFh | 0008 81BEh | 16 |
| | | TPU10 | 定时器的控制寄存器 | TCR | 00h | 0008 81C0h |
| | 定时器的模式寄存器 | | TMDR | 00h | 0008 81C1h | 8 |
| | 定时器的 I/O 控制寄存器 | | TIOR | 00h | 0008 81C2h | 8 |
| | 定时器的中断允许寄存器 | | TIER | 40h | 0008 81C4h | 8 |
| | 定时器的状态寄存器 | | TSR | xxh | 0008 81C5h | 8 |
| | 定时器的计数器 | | TCNT | 0000h | 0008 81C6h | 16 |
| | 定时器的通用寄存器 A | | TGRA | FFFFh | 0008 81C8h | 16 |
| | 定时器的通用寄存器 B | | TGRB | FFFFh | 0008 81CAh | 16 |
| | TPU11 | 定时器的控制寄存器 | TCR | 00h | 0008 81D0h | 8 |
| | | 定时器的模式寄存器 | TMDR | 00h | 0008 81D1h | 8 |
| | | 定时器的 I/O 控制寄存器 | TIOR | 00h | 0008 81D2h | 8 |
| | | 定时器的中断允许寄存器 | TIER | 40h | 0008 81D4h | 8 |
| | | 定时器的状态寄存器 | TSR | xxh | 0008 81D5h | 8 |
| | | 定时器的计数器 | TCNT | 0000h | 0008 81D6h | 16 |
| 定时器的通用寄存器 A | | TGRA | FFFFh | 0008 81D8h | 16 | |
| 定时器的通用寄存器 B | | TGRB | FFFFh | 0008 81DAh | 16 | |
| 通用 | 定时器的启动寄存器 | TSTRB | 00h | 0008 8170h | 8 | |
| | 定时器的同步寄存器 | TSYRB | 00h | 0008 8171h | 8 | |

15.2.1 定时器的控制寄存器 (TCR)

地址 TPU0.TCR 0008 8110h、TPU1.TCR 0008 8120h、TPU2.TCR 0008 8130h
 TPU3.TCR 0008 8140h、TPU4.TCR 0008 8150h、TPU5.TCR 0008 8160h
 TPU6.TCR 0008 8180h、TPU7.TCR 0008 8190h、TPU8.TCR 0008 81CAh
 TPU9.TCR 0008 81B0h、TPU10.TCR 0008 81C0h、TPU11.TCR 0008 81D0h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|--------------------|------------|----------------------|-----|
| b2-b0 | TPSC[2:0] | 定时器的预分器选择位 | 请参照表 15.6 ~ 表 15.11。 | R/W |
| b4-b3 | CKEG[1:0] | 输入时钟边沿选择位 | 请参照表 15.2。 | R/W |
| b7-b5 | CCLR[2:0] (注 1) | 计数器清除源选择位 | 请参照表 15.3 和表 15.4。 | R/W |

注 1. 单元 0 的 TPU1.TCR、TPU2.TCR、TPU4.TCR、TPU5.TCR 寄存器以及单元 1 的 TPU7.TCR、TPU8.TCR、TPU10.TCR、TPU11.TCR 寄存器的 b7 为保留位，读写值都为“0”。

在 TPU 中，各通道有 1 个 TCR 寄存器，共计 12 个。

TPUm.TCR 寄存器控制各通道的 TPUm.TCNT 计数器。

必须在 TPUm.TCNT 计数器停止运行的状态下设定 TPUm.TCR 寄存器。

TPSC[2:0] 位 (定时器的预分器选择位)

这些位选择 TCNT 计数器的时钟，各通道能分别选择时钟源。

在选择外部时钟作为时钟源时，必须将该引脚的数据方向寄存器 (DDR) 的位置“0” (输入端口)，并且将输入缓冲控制寄存器 (ICR) 的位置“1” (对应的引脚输入缓冲器有效)。详细内容请参照“14. I/O 端口”。

CKEG[1:0] 位 (输入时钟边沿选择位)

这些位选择输入时钟的边沿。

如果在双边沿对内部时钟进行计数，输入时钟的周期就为 1/2 (例如：PCLK/4 的双边沿 = PCLK/2 的上升沿)。

内部时钟的边沿选择在输入时钟为 PCLK/4 或者慢于 PCLK/4 时有效。如果选择 PCLK/1 或者其他通道的上溢 / 下溢作为输入时钟，就忽略此设定。

CCLR[2:0] 位 (计数器清除源选择位)

这些位选择 TCNT 计数器的清除源。

表 15.6 TPSC[2:0] 位 (TPU0、TPU6)

| 通道 | TPSC[2:0] 位 | | | 功能 |
|-------------|-------------|----|----|--------------------------|
| | b2 | b1 | b0 | |
| TPU0 (单元 0) | 0 | 0 | 0 | 内部时钟: 通过 PCLK/1 进行计数。 |
| TPU6 (单元 1) | 0 | 0 | 1 | 内部时钟: 通过 PCLK/4 进行计数。 |
| | 0 | 1 | 0 | 内部时钟: 通过 PCLK/16 进行计数。 |
| | 0 | 1 | 1 | 内部时钟: 通过 PCLK/64 进行计数。 |
| | 1 | 0 | 0 | 外部时钟: 通过 TCLKA 引脚输入进行计数。 |
| | 1 | 0 | 1 | 外部时钟: 通过 TCLKB 引脚输入进行计数。 |
| | 1 | 1 | 0 | 外部时钟: 通过 TCLKC 引脚输入进行计数。 |
| | 1 | 1 | 1 | 外部时钟: 通过 TCLKD 引脚输入进行计数。 |

表 15.7 TPSC[2:0] 位 (TPU1、TPU7)

| 通道 | TPSC[2:0] 位 | | | 功能 |
|-------------|-------------|----|----|--|
| | b2 | b1 | b0 | |
| TPU1 (单元 0) | 0 | 0 | 0 | 内部时钟: 通过 PCLK/1 进行计数。 |
| TPU7 (单元 1) | 0 | 0 | 1 | 内部时钟: 通过 PCLK/4 进行计数。 |
| | 0 | 1 | 0 | 内部时钟: 通过 PCLK/16 进行计数。 |
| | 0 | 1 | 1 | 内部时钟: 通过 PCLK/64 进行计数。 |
| | 1 | 0 | 0 | 外部时钟: 通过 TCLKA 引脚输入进行计数。 |
| | 1 | 0 | 1 | 外部时钟: 通过 TCLKB 引脚输入进行计数。 |
| | 1 | 1 | 0 | 内部时钟: 通过 PCLK/256 进行计数。 |
| | 1 | 1 | 1 | <ul style="list-style-type: none"> • TPU1 (单元 0) 通过 TPU2.TCNT 计数器的上溢 / 下溢进行计数。 • TPU7 (单元 1) 通过 TPU8.TCNT 计数器的上溢 / 下溢进行计数。 |

注 1. 在 TPU1 或者 TPU7 为相位计数模式时, 此设定无效。

表 15.8 TPSC[2:0] 位 (TPU2、TPU8)

| 通道 | TPSC[2:0] 位 | | | 功能 |
|-------------|-------------|----|----|--------------------------|
| | b2 | b1 | b0 | |
| TPU2 (单元 0) | 0 | 0 | 0 | 内部时钟: 通过 PCLK/1 进行计数。 |
| TPU8 (单元 1) | 0 | 0 | 1 | 内部时钟: 通过 PCLK/4 进行计数。 |
| | 0 | 1 | 0 | 内部时钟: 通过 PCLK/16 进行计数。 |
| | 0 | 1 | 1 | 内部时钟: 通过 PCLK/64 进行计数。 |
| | 1 | 0 | 0 | 外部时钟: 通过 TCLKA 引脚输入进行计数。 |
| | 1 | 0 | 1 | 外部时钟: 通过 TCLKB 引脚输入进行计数。 |
| | 1 | 1 | 0 | 外部时钟: 通过 TCLKC 引脚输入进行计数。 |
| | 1 | 1 | 1 | 内部时钟: 通过 PCLK/1024 进行计数。 |

注 1. 在 TPU2 或者 TPU8 为相位计数模式时, 此设定无效。

表 15.9 TPSC[2:0] 位 (TPU3、TPU9)

| 通道 | TPSC[2:0] 位 | | | 功能 |
|-------------|-------------|----|----|--------------------------|
| | b2 | b1 | b0 | |
| TPU3 (单元 0) | 0 | 0 | 0 | 内部时钟: 通过 PCLK/1 进行计数。 |
| TPU9 (单元 1) | 0 | 0 | 1 | 内部时钟: 通过 PCLK/4 进行计数。 |
| | 0 | 1 | 0 | 内部时钟: 通过 PCLK/16 进行计数。 |
| | 0 | 1 | 1 | 内部时钟: 通过 PCLK/64 进行计数。 |
| | 1 | 0 | 0 | 外部时钟: 通过 TCLKA 引脚输入进行计数。 |
| | 1 | 0 | 1 | 内部时钟: 通过 PCLK/1024 进行计数。 |
| | 1 | 1 | 0 | 内部时钟: 通过 PCLK/256 进行计数。 |
| | 1 | 1 | 1 | 内部时钟: 通过 PCLK/4096 进行计数。 |

表 15.10 TPSC[2:0] 位 (TPU4、TPU10)

| 通道 | TPSC[2:0] 位 | | | 功能 |
|--------------|-------------|----|----|--|
| | b2 | b1 | b0 | |
| TPU4 (单元 0) | 0 | 0 | 0 | 内部时钟: 通过 PCLK/1 进行计数。 |
| TPU10 (单元 1) | 0 | 0 | 1 | 内部时钟: 通过 PCLK/4 进行计数。 |
| | 0 | 1 | 0 | 内部时钟: 通过 PCLK/16 进行计数。 |
| | 0 | 1 | 1 | 内部时钟: 通过 PCLK/64 进行计数。 |
| | 1 | 0 | 0 | 外部时钟: 通过 TCLKA 引脚输入进行计数。 |
| | 1 | 0 | 1 | 外部时钟: 通过 TCLKC 引脚输入进行计数。 |
| | 1 | 1 | 0 | 内部时钟: 通过 PCLK/1024 进行计数。 |
| | 1 | 1 | 1 | <ul style="list-style-type: none"> • TPU4 (单元 0) 通过 TPU5.TCNT 计数器的上溢 / 下溢进行计数。 • TPU10 (单元 1) 通过 TPU11.TCNT 计数器的上溢 / 下溢进行计数。 |

注 1. 在 TPU4 或者 TPU10 为相位计数模式时, 此设定无效。

表 15.11 TPSC[2:0] 位 (TPU5、TPU11)

| 通道 | TPSC[2:0] 位 | | | 功能 |
|--------------|-------------|----|----|--------------------------|
| | b2 | b1 | b0 | |
| TPU5 (单元 0) | 0 | 0 | 0 | 内部时钟: 通过 PCLK/1 进行计数。 |
| TPU11 (单元 1) | 0 | 0 | 1 | 内部时钟: 通过 PCLK/4 进行计数。 |
| | 0 | 1 | 0 | 内部时钟: 通过 PCLK/16 进行计数。 |
| | 0 | 1 | 1 | 内部时钟: 通过 PCLK/64 进行计数。 |
| | 1 | 0 | 0 | 外部时钟: 通过 TCLKA 引脚输入进行计数。 |
| | 1 | 0 | 1 | 外部时钟: 通过 TCLKC 引脚输入进行计数。 |
| | 1 | 1 | 0 | 内部时钟: 通过 PCLK/256 进行计数。 |
| | 1 | 1 | 1 | 外部时钟: 通过 TCLKD 引脚输入进行计数。 |

注 1. 在 TPU5 或者 TPU11 为相位计数模式时, 此设定无效。

表 15.12 CKEG[1:0] 位

| CKEG[1:0] 位 | | 输入时钟 | |
|-------------|----|-----------|-----------|
| b4 | b3 | 内部时钟 | 外部时钟 |
| 0 | 0 | 在下降沿进行计数。 | 在上升沿进行计数。 |
| 0 | 1 | 在上升沿进行计数。 | 在下降沿进行计数。 |
| 1 | 0 | 在双边沿进行计数。 | 在双边沿进行计数。 |
| 1 | 1 | 在双边沿进行计数。 | 在双边沿进行计数。 |

表 15.13 CCLR[2:0] 位 (TPU0、TPU3、TPU6、TPU9)

| 通道 | CCLR[2:0] 位 | | | 功能 |
|---------------------|-------------|----|----|--|
| | b7 | b6 | b5 | |
| (单元 0) TPU0、TPU3 | 0 | 0 | 0 | 禁止清除 TCNT 计数器。 |
| | 0 | 0 | 1 | 在发生 TGRA 寄存器的比较匹配 / 输入捕捉时, 清除 TCNT 计数器。 |
| (单元 1) TPU6、TPU9 | 0 | 1 | 0 | 在发生 TGRB 寄存器的比较匹配 / 输入捕捉时, 清除 TCNT 计数器。 |
| | 0 | 1 | 1 | 在清除正在进行同步清除 / 同步运行的其他通道的计数器时, 清除 TCNT 计数器 (注 2)。 |
| | 1 | 0 | 0 | 禁止清除 TCNT 计数器。 |
| | 1 | 0 | 1 | 在发生 TGRC 寄存器的比较匹配 / 输入捕捉时, 清除 TCNT 计数器 (注 1)。 |
| | 1 | 1 | 0 | 在发生 TGRD 寄存器的比较匹配 / 输入捕捉时, 清除 TCNT 计数器 (注 1)。 |
| | 1 | 1 | 1 | 在清除正在进行同步清除 / 同步运行的其他通道的计数器时, 清除 TCNT 计数器 (注 2)。 |

注 1. 在将 TGRC 寄存器或者 TGRD 寄存器用作缓冲寄存器时, 因为优先设定缓冲寄存器并且不发生比较匹配 / 输入捕捉, 所以不清除 TCNT 计数器。

注 2. 通过将 TSYRm.SYNCj 位 (m=A,B, j=0,3) 位置 “1”, 设定同步运行。

表 15.14 CCLR[2:0] 位 (TPU1、TPU2、TPU4、TPU5、TPU7、TPU8、TPU10、TPU11)

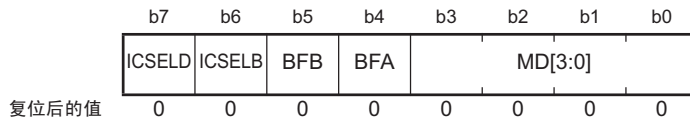
| 通道 | CCLR[2:0] 位 (注 1) | | | 功能 |
|--|-------------------|----|----|--|
| | b7 | b6 | b5 | |
| (单元 0) TPU1、TPU2 TPU4、TPU5 | 0 | 0 | 0 | 禁止清除 TCNT 计数器。 |
| | 0 | 0 | 1 | 在发生 TGRA 寄存器的比较匹配 / 输入捕捉时, 清除 TCNT 计数器。 |
| (单元 1) TPU7、TPU8 TPU10、 TPU11 | 0 | 1 | 0 | 在发生 TGRB 寄存器的比较匹配 / 输入捕捉时, 清除 TCNT 计数器。 |
| | 0 | 1 | 1 | 在清除正在进行同步清除 / 同步运行的其他通道的计数器时, 清除 TCNT 计数器 (注 2)。 |
| | 1 | 0 | 0 | 不能设定 |
| | 1 | 0 | 1 | 不能设定 |
| | 1 | 1 | 0 | 不能设定 |
| | 1 | 1 | 1 | 不能设定 |

注 1. 单元 0 的 TPU1.TCR、TPU2.TCR、TPU4.TCR、TPU5.TCR 寄存器以及单元 1 的 TPU7.TCR、TPU8.TCR、TPU10.TCR、TPU11.TCR 寄存器的 b7 为保留位, 读写值都为 “0”。

注 2. 通过将 TSYRm.SYNCj 位 (m=A,B, j=1,2,4,5) 位置 “1”, 设定同步运行。

15.2.2 定时器的模式寄存器 (TMDR)

地址 TPU0.TMDR 0008 8111h、TPU1.TMDR 0008 8121h、TPU2.TMDR 0008 8131h
 TPU3.TMDR 0008 8141h、TPU4.TMDR 0008 8151h、TPU5.TMDR 0008 8161h
 TPU6.TMDR 0008 8181h、TPU7.TMDR 0008 8191h、TPU8.TMDR 0008 81A1h
 TPU9.TMDR 0008 81B1h、TPU10.TMDR 0008 81C1h、TPU11.TMDR 0008 81D1h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------------|-----------------|--|-----|
| b3-b0 | MD[3:0] | 模式选择位 | b3 (注1) b0 0 0 0 0: 正常运行 0 0 0 1: 不能设定 0 0 1 0: PWM 模式 1 0 0 1 1: PWM 模式 2 0 1 0 0: 相位计数模式 1 (注2) 0 1 0 1: 相位计数模式 2 (注2) 0 1 1 0: 相位计数模式 3 (注2) 0 1 1 1: 相位计数模式 4 (注2) 上述以外: 不能设定 | R/W |
| b4 | BFA (注3) | 缓冲运行 A 位 | 0: TPUm.TGRA 寄存器进行正常运行 1: TPUm.TGRA 寄存器和 TPUm.TGRC 寄存器进行缓冲运行 (m=0,3,6,9) | R/W |
| b5 | BFB (注4) | 缓冲运行 B 位 | 0: TPUm.TGRB 寄存器进行正常运行 1: TPUm.TGRB 寄存器和 TPUm.TGRD 寄存器进行缓冲运行 (m=0,3,6,9) | R/W |
| b6 | ICSELB | TGRB 输入捕捉的输入选择位 | 0: 输入捕捉的输入源为 TIOCBn 引脚 1: 输入捕捉的输入源为 TIOCA n 引脚 (n=0 ~ 11) | R/W |
| b7 | ICSELD (注4) | TGRD 输入捕捉的输入选择位 | 0: 输入捕捉的输入源为 TIOCDn 引脚 1: 输入捕捉的输入源为 TIOCCn 引脚 (n=0,3,6,9) | R/W |

注 1. b3 为保留位，读写值都为“0”。

注 2. 不能将 TPU0、TPU3 (单元 0)、TPU6 和 TPU9 (单元 1) 设定为相位计数模式，必须将 b2 置“0”。

注 3. 没有 TGRC 寄存器的 TPU1、TPU2、TPU4、TPU5 (单元 0)、TPU7、TPU8、TPU10 和 TPU11 (单元 1) 的 b4 为保留位，读写值都为“0”。

注 4. 没有 TGRD 寄存器的 TPU1、TPU2、TPU4、TPU5 (单元 0)、TPU7、TPU8、TPU10 和 TPU11 (单元 1) 的 b5 和 b7 为保留位，读写值都为“0”。

在 TPU 中，各通道有 1 个 TMDR 寄存器，共计 12 个。

TPUm.TMDR 寄存器是设定各通道运行模式的寄存器。

必须在 TPUm.TCNT 计数器停止运行的状态下设定 TPUm.TMDR 寄存器。

MD[3:0] 位 (模式选择位)

这些位设定定时器的运行模式。

BFA 位 (缓冲运行 A 位)

此位选择是将 TPUm.TGRA 寄存器 (m=0,3,6,9) 进行正常运行还是将 TPUm.TGRA 寄存器和 TPUm.TGRC 寄存器 (m=0,3,6,9) 组合进行缓冲运行。

如果将 TGRC 寄存器用作缓冲寄存器, 就不会发生 TGRC 寄存器的输入捕捉 / 输出比较。

BFB 位 (缓冲运行 B 位)

此位选择是将 TPUm.TGRB 寄存器 (m=0,3,6,9) 进行正常运行还是将 TPUm.TGRB 寄存器和 TPUm.TGRD 寄存器 (m=0,3,6,9) 组合进行缓冲运行。

如果将 TGRD 寄存器用作缓冲寄存器, 就不会发生 TGRD 寄存器的输入捕捉 / 输出比较。

ICSELB 位 (TGRB 输入捕捉的输入选择位)

此位选择 TPUm.TGRB 寄存器 (m=0 ~ 11) 的输入捕捉的输入。能使用此功能并且通过 1 个 TIOCA_n 输入引脚测量输入脉冲的 High 电平宽度和周期。

ICSELD 位 (TGRD 输入捕捉的输入选择位)

此位选择 TPUm.TGRD 寄存器 (m=0,3,6,9) 的输入捕捉的输入。

能使用此功能并且通过 1 个 TIOCC_n 输入引脚测量输入脉冲的 High 电平宽度和周期。

15.2.3 定时器的 I/O 控制寄存器 (TIORH、TIORL、TIOR)

- 单元0 (TPU0.TIORH、TPU1.TIOR、TPU2.TIOR、TPU3.TIORH、TPU4.TIOR、TPU5.TIOR)
- 单元1 (TPU6.TIORH、TPU7.TIOR、TPU8.TIOR、TPU9.TIORH、TPU10.TIOR、TPU11.TIOR)

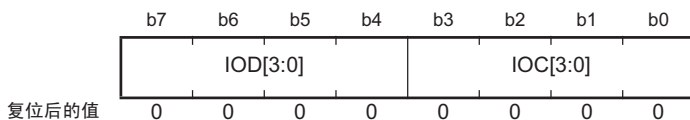
地址 TPU0.TIORH 0008 8112h、TPU1.TIOR 0008 8122h、TPU2.TIOR 0008 8132h
 TPU3.TIORH 0008 8142h、TPU4.TIOR 0008 8152h、TPU5.TIOR 0008 8162h
 TPU6.TIORH 0008 8182h、TPU7.TIOR 0008 8192h、TPU8.TIOR 0008 81A2h
 TPU9.TIORH 0008 81B2h、TPU10.TIOR 0008 81C2h、TPU11.TIOR 0008 81D2h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|--------------|-----------------------|-----|
| b3-b0 | IOA[3:0] | TGRA 寄存器的控制位 | 请参照表 15.15 ~ 表 15.20。 | R/W |
| b7-b4 | IOB[3:0] | TGRB 寄存器的控制位 | 请参照表 15.15 ~ 表 15.20。 | R/W |

- 单元0 (TPU0.TIORL、TPU3.TIORL)
- 单元1 (TPU6.TIORL、TPU9.TIORL)

地址 TPU0.TIORL 0008 8113h、TPU3.TIORL 0008 8143h
 TPU6.TIORL 0008 8183h、TPU9.TIORL 0008 81B3h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|--------------|----------------------|-----|
| b3-b0 | IOC[3:0] | TGRC 寄存器的控制位 | 请参照表 15.21 和表 15.22。 | R/W |
| b7-b4 | IOD[3:0] | TGRD 寄存器的控制位 | 请参照表 15.21 和表 15.22。 | R/W |

在 TPU 中，TPU0、TPU3、TPU6 和 TPU9 各有 1 个 TIORH 寄存器，共计 4 个；TPU0、TPU3、TPU6 和 TPU9 各有 1 个 TIORL 寄存器，共计 4 个；TPU1、TPU2、TPU4、TPU5、TPU7、TPU8、TPU10 和 TPU11 各有 1 个 TIOR 寄存器，共计 8 个。总共 16 个定时器的 I/O 控制寄存器。

TIORH、TIORL、TIOR 寄存器控制 TGRA ~ TGRD 寄存器。

必须注意：TIORH、TIORL、TIOR 寄存器受 TMDR 寄存器设定的影响。

在计数器停止计数运行的状态下 (TSTRm.CSTj 位 (m=A,B, j=0 ~ 5) 为 “0”)，TIORH、TIORL、TIOR 寄存器指定的初始输出有效。另外，在 PWM 模式 2 的情况下，指定 TCNT 计数器为 “0” 时的输出。

如果在将 TGRC 寄存器或者 TGRD 寄存器设定为缓冲运行，此设定就无效，用作缓冲寄存器。

在通过 TIORH、TIORL、TIOR 寄存器选择输入捕捉的功能时，必须将对应的引脚的数据方向寄存器 (DDR) 的位置 “0” (输入端口)，并且将输入缓冲控制寄存器 (ICR) 的位置 “1” (对应的引脚输入缓冲器有效)。详细内容请参照 “14. I/O 端口”。

IOA[3:0] 位 (TGRA 寄存器的控制位)

这些位选择 TPUm.TGRA 寄存器 (m=0 ~ 11) 的功能。

IOB[3:0] 位 (TGRB 寄存器的控制位)

这些位选择 TPUm.TGRB 寄存器 (m=0 ~ 11) 的功能。

IOC[3:0] 位 (TGRC 寄存器的控制位)

这些位选择 TPUm.TGRC 寄存器 (m=0,3,6,9) 的功能。

IOD[3:0] 位 (TGRD 寄存器的控制位)

这些位选择 TPU_m.TGRD 寄存器 (m=0,3,6,9) 的功能。

表 15.15 TPU0.TIORH、TPU6.TIORH

| IOA[3:0] 位 | | | | 说明 | |
|------------|----|----|----|--|--|
| b3 | b2 | b1 | b0 | TPU _m .TGRA 寄存器 (m=0,6) 的功能 | TIOCA _n 引脚 (n=0,6) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时 High 输出电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCA _n 引脚, 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCA _n 引脚, 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCA _n 引脚, 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> • TPU0 捕捉输入源为 TPU1 的计数时钟。 通过 TPU1.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注1)。 • TPU6 捕捉输入源为 TPU7 的计数时钟。 通过 TPU7.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注1)。 |

| IOB[3:0] 位 | | | | 说明 | |
|------------|----|----|----|--|--|
| b7 | b6 | b5 | b4 | TPU _m .TGRB 寄存器 (m=0,6) 的功能 | TIOCB _n 引脚 (n=0,6) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注2), 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注2), 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注2), 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> • TPU0 捕捉输入源为 TPU1 的计数时钟。 通过 TPU1.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注1)。 • TPU6 捕捉输入源为 TPU7 的计数时钟。 通过 TPU7.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注1)。 |

x: Don't care

注 1. 如果 TPU_m.TCR.TPSC[2:0] 位为 "000b" 并且 TPU_m.TCNT 计数器的计数时钟使用 PCLK/1, 此设定就无效, 不发生输入捕捉 (m=1,7)。

注 2. 通过 TPU_m.TMDR.ICSELB 位进行选择 (m=0,6)。

表 15.16 TPU1.TIOR、TPU7.TIOR

| IOA[3:0] 位 | | | | 说明 | |
|------------|----|----|----|---------------------------|--|
| b3 | b2 | b1 | b0 | TPUm.TGRA 寄存器 (m=1,7) 的功能 | TIOCA _n 引脚 (n=1,7) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCA _n 引脚, 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCA _n 引脚, 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCA _n 引脚, 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> • TPU1 捕捉输入源为 TPU0.TGRA 寄存器的比较匹配 / 输入捕捉。 通过发生 TPU0.TGRA 寄存器的比较匹配 / 输入捕捉进行输入捕捉。 • TPU7 捕捉输入源为 TPU6.TGRA 寄存器的比较匹配 / 输入捕捉。 通过发生 TPU6.TGRA 寄存器的比较匹配 / 输入捕捉进行输入捕捉。 |

| IOB[3:0] 位 | | | | 说明 | |
|------------|----|----|----|---------------------------|---|
| b7 | b6 | b5 | b4 | TPUm.TGRB 寄存器 (m=1,7) 的功能 | TIOCB _n 引脚 (n=1,7) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注1), 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注1), 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注1), 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> • TPU1 捕捉输入源为 TPU0.TGRC 寄存器比较匹配 / 输入捕捉。 通过发生 TPU0.TGRC 寄存器的比较匹配 / 输入捕捉进行输入捕捉。 • TPU7 捕捉输入源为 TPU6.TGRC 寄存器的比较匹配 / 输入捕捉。 通过发生 TPU6.TGRC 寄存器的比较匹配 / 输入捕捉进行输入捕捉。 |

x: Don't care

注 1. 通过 TPUm.TMDR.ICSELB 位进行选择 (m=1,7)。

表 15.17 TPU2.TIOR、TPU8.TIOR

| IOA[3:0] 位 | | | | 说明 | |
|------------|----|----|----|---------------------------|---|
| b3 | b2 | b1 | b0 | TPUm.TGRA 寄存器 (m=2,8) 的功能 | TIOCA _n 引脚 (n=2,8) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | x | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCA _n 引脚, 在上升沿进行输入捕捉。 |
| 1 | x | 0 | 1 | | 捕捉输入源为 TIOCA _n 引脚, 在下降沿进行输入捕捉。 |
| 1 | x | 1 | x | | 捕捉输入源为 TIOCA _n 引脚, 在双边沿进行输入捕捉。 |

| IOB[3:0] 位 | | | | 说明 | |
|------------|----|----|----|---------------------------|--|
| b7 | b6 | b5 | b4 | TPUm.TGRB 寄存器 (m=2,8) 的功能 | TIOCB _n 引脚 (n=2,8) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | x | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注1), 在上升沿进行输入捕捉。 |
| 1 | x | 0 | 1 | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注1), 在下降沿进行输入捕捉。 |
| 1 | x | 1 | x | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注1), 在双边沿进行输入捕捉。 |

x: Don't care

注 1. 通过 TPUm.TMDR.ICSELB 位进行选择 (m=2,8)。

表 15.18 TPU3.TIORH、TPU9.TIORH

| IOA[3:0] 位 | | | | 说明 | |
|------------|----|----|----|---------------------------|--|
| b3 | b2 | b1 | b0 | TPUm.TGRA 寄存器 (m=3,9) 的功能 | TIOCA _n 引脚 (n=3,9) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCA _n 引脚, 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCA _n 引脚, 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCA _n 引脚, 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> • TPU3 捕捉输入源为 TPU4 的计数时钟。 通过 TPU4.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注1) • TPU9 捕捉输入源为 TPU10 的计数时钟。 通过 TPU10.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注1) |

| IOB[3:0] 位 | | | | 说明 | |
|------------|----|----|----|---------------------------|--|
| b7 | b6 | b5 | b4 | TPUm.TGRB 寄存器 (m=3,9) 的功能 | TIOCB _n 引脚 (n=3,9) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注2), 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注2), 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注2), 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> • TPU3 捕捉输入源为 TPU4 的计数时钟。 通过 TPU4.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注1)。 • TPU9 捕捉输入源为 TPU10 的计数时钟。 通过 TPU10.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注1)。 |

x: Don't care

注 1. 如果 TPUm.TCR.TPSC[2:0] 位为 "000b" 并且 TPUm.TCNT 的计数时钟使用 PCLK/1, 此设定就无效, 不发生输入捕捉 (m=4,10)。

注 2. 通过 TPUm.TMDR.ICSELB 位进行选择 (m=3,9)。

表 15.19 TPU4.TIOR、TPU10.TIOR

| IOA[3:0] 位 | | | | 说明 | |
|------------|----|----|----|----------------------------|---|
| b3 | b2 | b1 | b0 | TPUm.TGRA 寄存器 (m=4,10) 的功能 | TIOCA _n 引脚 (n=4,10) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCA _n 引脚, 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCA _n 引脚, 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCA _n 引脚, 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> • TPU4 捕捉输入源为 TPU3.TGRA 寄存器的比较匹配 / 输入捕捉。 通过发生 TPU3.TGRA 寄存器的比较匹配 / 输入捕捉进行输入捕捉。 • TPU10 捕捉输入源为 TPU9.TGRA 寄存器的比较匹配 / 输入捕捉。 通过发生 TPU9.TGRA 寄存器的比较匹配 / 输入捕捉进行输入捕捉。 |

| IOB[3:0] 位 | | | | 说明 | |
|------------|----|----|----|----------------------------|---|
| b7 | b6 | b5 | b4 | TPUm.TGRB 寄存器 (m=4,10) 的功能 | TIOCB _n 引脚 (n=4,10) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注1), 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注1), 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注1), 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> • TPU4 捕捉输入源为 TPU3.TGRC 寄存器的比较匹配 / 输入捕捉。 通过发生 TPU3.TGRC 寄存器的比较匹配 / 输入捕捉进行输入捕捉。 • TPU10 捕捉输入源为 TPU9.TGRC 寄存器的比较匹配 / 输入捕捉。 通过发生 TPU9.TGRC 寄存器的比较匹配 / 输入捕捉进行输入捕捉。 |

x: Don't care

注 1. 通过 TPUm.TMDR.ICSELB 位进行选择 (m=4,10)。

表 15.20 TPU5.TIOR、TPU11.TIOR

| IOA[3:0] 位 | | | | 说明 | |
|------------|----|----|----|----------------------------|---|
| b3 | b2 | b1 | b0 | TPUm.TGRA 寄存器 (m=5,11) 的功能 | TIOCA _n 引脚 (n=5,11) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | x | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCA _n 引脚, 在上升沿进行输入捕捉。 |
| 1 | x | 0 | 1 | | 捕捉输入源为 TIOCA _n 引脚, 在下降沿进行输入捕捉。 |
| 1 | x | 1 | x | | 捕捉输入源为 TIOCA _n 引脚, 在双边沿进行输入捕捉。 |

| IOB[3:0] 位 | | | | 说明 | |
|------------|----|----|----|----------------------------|---|
| b7 | b6 | b5 | b4 | TPUm.TGRB 寄存器 (m=5,11) 的功能 | TIOCB _n 引脚 (n=5,11) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | x | 0 | 0 | 输入捕捉寄存器 | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注 1), 在上升沿进行输入捕捉。 |
| 1 | x | 0 | 1 | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注 1), 在下降沿进行输入捕捉。 |
| 1 | x | 1 | x | | 捕捉输入源为 TIOCB _n /TIOCA _n 引脚 (注 1), 在双边沿进行输入捕捉。 |

x: Don't care

注 1. 通过 TPUm.TMDR.ICSELB 位进行选择 (m=5,11)。

表 15.21 TPU0.TIORL、TPU6.TIORL

| IOC[3:0] 位 | | | | 说明 | |
|------------|----|----|----|---------------------------|--|
| b3 | b2 | b1 | b0 | TPUm.TGRC 寄存器 (m=0,6) 的功能 | TIOCCn 引脚 (n=0,6) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 (注1) | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 (注1) | 捕捉输入源为 TIOCCn 引脚, 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCCn 引脚, 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCCn 引脚, 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> • TPU0 捕捉输入源为 TPU1 的计数时钟。 通过 TPU1.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注3)。 • TPU6 捕捉输入源为 TPU7 的计数时钟。 通过 TPU7.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注3)。 |

| IOD[3:0] 位 | | | | 说明 | |
|------------|----|----|----|---------------------------|--|
| b7 | b6 | b5 | b4 | TPUm.TGRD 寄存器 (m=0,6) 的功能 | TIOCDn 引脚 (n=0,6) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 (注2) | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 (注2) | 捕捉输入源为 TIOCDn/TIOCCn 引脚 (注4), 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCDn/TIOCCn 引脚 (注4), 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCDn/TIOCCn 引脚 (注4), 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> • TPU0 捕捉输入源为 TPU1 的计数时钟。 通过 TPU1.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注3)。 • TPU6 捕捉输入源为 TPU7 的计数时钟。 通过 TPU7.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注3)。 |

x: Don't care

- 注 1. 如果将 TPUm.TMDR.BFA 位置 “1” (TPUm.TGRA 寄存器和 TPUm.TGRC 寄存器进行缓冲运行) 后将 TPUm.TGRC 寄存器用作缓冲寄存器, 此设定就无效, 不发生输入捕捉 / 输出比较 (m=0,6)。
- 注 2. 如果将 TPUm.TMDR.BFB 位置 “1” (TPUm.TGRB 寄存器和 TPUm.TGRD 寄存器进行缓冲运行) 后将 TPUm.TGRD 寄存器用作缓冲寄存器, 此设定就无效, 不发生输入捕捉 / 输出比较 (m=0,6)。
- 注 3. 如果 TPUm.TCR.TPSC[2:0] 位为 “000b” 并且 TPUm.TCNT 计数器的计数时钟使用 PCLK/1, 此设定就无效, 不发生输入捕捉 (m=1,7)。
- 注 4. 通过设定 TPUm.TMDR.ICSELD 位进行选择 (m=0,6)。

表 15.22 TPU3.TIORL、TPU9.TIORL

| IOC[3:0] 位 | | | | 说明 | |
|------------|----|----|----|---------------------------|--|
| b3 | b2 | b1 | b0 | TPUm.TGRC 寄存器 (m=3,9) 的功能 | TIOCCn 引脚 (n=3,9) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 (注1) | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 (注1) | 捕捉输入源为 TIOCCn 引脚, 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCCn 引脚, 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCCn 引脚, 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> TPU3 捕捉输入源为 TPU4 的计数时钟。 通过 TPU4.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注3)。 TPU9 捕捉输入源为 TPU10 的计数时钟。 通过 TPU10.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注3)。 |

| IOD[3:0] 位 | | | | 说明 | |
|------------|----|----|----|---------------------------|--|
| b7 | b6 | b5 | b4 | TPUm.TGRD 寄存器 (m=3,9) 的功能 | TIOCDn 引脚 (n=3,9) 的功能 |
| 0 | 0 | 0 | 0 | 输出比较寄存器 (注2) | 禁止输出。 |
| 0 | 0 | 0 | 1 | | 初始输出 Low 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 0 | 1 | 0 | | 初始输出 Low 电平, 在比较匹配时输出 High 电平。 |
| 0 | 0 | 1 | 1 | | 初始输出 Low 电平, 在比较匹配时进行交替输出。 |
| 0 | 1 | 0 | 0 | | 禁止输出。 |
| 0 | 1 | 0 | 1 | | 初始输出 High 电平, 在比较匹配时输出 Low 电平。 |
| 0 | 1 | 1 | 0 | | 初始输出 High 电平, 在比较匹配时输出 High 电平。 |
| 0 | 1 | 1 | 1 | | 初始输出 High 电平, 在比较匹配时进行交替输出。 |
| 1 | 0 | 0 | 0 | 输入捕捉寄存器 (注2) | 捕捉输入源为 TIOCDn/TIOCCn 引脚 (注4), 在上升沿进行输入捕捉。 |
| 1 | 0 | 0 | 1 | | 捕捉输入源为 TIOCDn/TIOCCn 引脚 (注4), 在下降沿进行输入捕捉。 |
| 1 | 0 | 1 | x | | 捕捉输入源为 TIOCDn/TIOCCn 引脚 (注4), 在双边沿进行输入捕捉。 |
| 1 | 1 | x | x | | <ul style="list-style-type: none"> TPU3 捕捉输入源为 TPU4 的计数时钟。 通过 TPU4.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注3)。 TPU9 捕捉输入源为 TPU10 的计数时钟。 通过 TPU10.TCNT 计数器的递增计数 / 递减计数进行输入捕捉 (注3)。 |

x: Don't care

- 注 1. 如果将 TPUm.TMDR.BFA 位置 “1” (TPUm.TGRA 寄存器和 TPUm.TGRC 寄存器进行缓冲运行) 后将 TPUm.TGRC 寄存器用作缓冲寄存器, 此设定就无效, 不发生输入捕捉 / 输出比较 (m=3,9)。
- 注 2. 如果将 TPUm.TMDR.BFB 位置 “1” (TPUm.TGRB 寄存器和 TPUm.TGRD 寄存器进行缓冲运行) 后将 TPUm.TGRD 寄存器用作缓冲寄存器, 此设定就无效, 不发生输入捕捉 / 输出比较 (m=3,9)。
- 注 3. 如果 TPUm.TCR.TPSC[2:0] 位为 “000b” 并且 TPUm.TCNT 的计数时钟使用 PCLK/1, 此设定就无效, 不发生输入捕捉 (m=4,10)。
- 注 4. 通过设定 TPUm.TMDR.ICSELD 位进行选择 (m=3,9)。

15.2.4 定时器的中断允许寄存器 (TIER)

地址 TPU0.TIER 0008 8114h、TPU1.TIER 0008 8124h、TPU2.TIER 0008 8134h
 TPU3.TIER 0008 8144h、TPU4.TIER 0008 8154h、TPU5.TIER 0008 8164h
 TPU6.TIER 0008 8184h、TPU7.TIER 0008 8194h、TPU8.TIER 0008 81A4h
 TPU9.TIER 0008 81B4h、TPU10.TIER 0008 81C4h、TPU11.TIER 0008 81D4h

| | | | | | | | |
|-------|----|-------|-------|-------|-------|-------|-------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TTGE | — | TCIEU | TCIEV | TGIED | TGIEC | TGIEB | TGIEA |
| 复位后的值 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|------------|----------------|---|-----|
| b0 | TGIEA | TGRA 中断允许位 | 0: 禁止中断 (TGImA) 1: 允许中断 (TGImA) (m=0 ~ 11) | R/W |
| b1 | TGIEB | TGRB 中断允许位 | 0: 禁止中断 (TGImB) 1: 允许中断 (TGImB) (m=0 ~ 11) | R/W |
| b2 | TGIEC (注1) | TGRC 中断允许位 | 0: 禁止中断 (TGImC) 1: 允许中断 (TGImC) (m=0,3,6,9) | R/W |
| b3 | TGIED (注1) | TGRD 中断允许位 | 0: 禁止中断 (TGImD) 1: 允许中断 (TGImD) (m=0,3,6,9) | R/W |
| b4 | TCIEV | 上溢中断允许位 | 0: 禁止中断 (TCImV) 1: 允许中断 (TCImV) (m=0 ~ 11) | R/W |
| b5 | TCIEU (注2) | 下溢中断允许位 | 0: 禁止中断 (TCImU) 1: 允许中断 (TCImU) (m=1,2,4,5,7,8,10,11) | R/W |
| b6 | — | 保留位 | 读写值都为“1”。 | R/W |
| b7 | TTGE | A/D 转换开始的请求允许位 | 0: 禁止 A/D 转换开始请求的发生 1: 允许 A/D 转换开始请求的发生 | R/W |

注 1. 单元 0 的 TPU1.TIER、TPU2.TIER、TPU4.TIER、TPU5.TIER 寄存器以及单元 1 的 TPU7.TIER、TPU8.TIER、TPU10.TIER、TPU11.TIER 寄存器的 b3 和 b2 为保留位，读写值都为“0”。

注 2. 单元 0 的 TPU0.TIER、TPU3.TIER 寄存器以及单元 1 的 TPU6.TIER、TPU9.TIER 寄存器的 b5 为保留位，读写值都为“0”。

在 TPU 中，各通道有 1 个 TIER 寄存器，共计 12 个。

TPUm.TIER 寄存器是控制允许或者禁止各通道中断的寄存器。

TGIEA 位 (TGRA 中断允许位)

此位允许或者禁止中断 (TGImA) (m=0 ~ 11)。

TGIEB 位 (TGRB 中断允许位)

此位允许或者禁止中断 (TGImB) (m=0 ~ 11)。

TGIEC 位 (TGRC 中断允许位)

此位允许或者禁止中断 (TGImC) (m=0,3,6,9)。

TGIED 位 (TGRD 中断允许位)

此位允许或者禁止中断 (TGImD) (m=0,3,6,9)。

TCIEV 位 (上溢中断允许位)

此位允许或者禁止中断 (TCImV) (m=0 ~ 11)。

TCIEU 位 (下溢中断允许位)

此位允许或者禁止中断 (TCImU) (m=1,2,4,5,7,8,10,11)。

TTGE 位 (A/D 转换开始的请求允许位)

此位允许或者禁止由 TPUm.TGRA 寄存器 (m=0 ~ 11) 的输入捕捉 / 比较匹配产生的 A/D 转换开始请求。

15.2.5 定时器的状态寄存器 (TSR)

地址 TPU0.TSR 0008 8115h、TPU1.TSR 0008 8125h、TPU2.TSR 0008 8135h
 TPU3.TSR 0008 8145h、TPU4.TSR 0008 8155h、TPU5.TSR 0008 8165h
 TPU6.TSR 0008 8185h、TPU7.TSR 0008 8195h、TPU8.TSR 0008 81A5h
 TPU9.TSR 0008 81B5h、TPU10.TSR 0008 81C5h、TPU11.TSR 0008 81D5h

| | | | | | | | |
|------|----|----|----|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TCFD | — | — | — | — | — | — | — |

复位后的值 1 1 x x x x x x

x: 不定值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-----------|--------|---|-----|
| b5-b0 | — | 保留位 | 读取值为不定值, 只能写“1”。 | R/W |
| b6 | — | 保留位 | 读写值都为“1”。 | R/W |
| b7 | TCFD (注1) | 计数方向标志 | 0: TPUm.TCNT 计数器进行递减计数 1: TPUm.TCNT 计数器进行递增计数 (n=1,2,4,5,7,8,10,11) | R |

注1. 单元 0 的 TPU0.TSR、TPU3.TSR 寄存器以及单元 1 的 TPU6.TSR、TPU9.TSR 寄存器的 b7 为保留位, 读写值都为“1”。

在 TPU 中, 各通道有 1 个 TSR 寄存器, 共计 12 个。

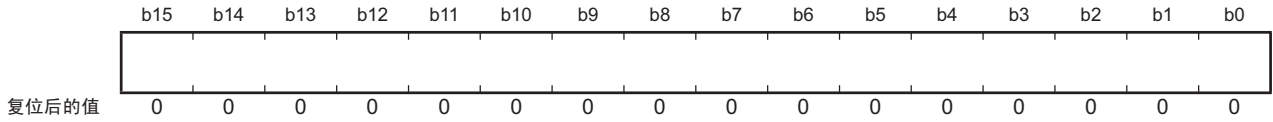
TPUm.TSR 寄存器表示 TPUm.TCNT 计数器的计数方向。

TCFD 标志 (计数方向标志)

这是表示 TPUm.TCNT 计数器 (m=1,2,4,5,7,8,10,11) 的计数方向的状态标志。

15.2.6 定时器的计数器 (TCNT)

地址 TPU0.TCNT 0008 8116h、TPU1.TCNT 0008 8126h、TPU2.TCNT 0008 8136h
 TPU3.TCNT 0008 8146h、TPU4.TCNT 0008 8156h、TPU5.TCNT 0008 8166h
 TPU6.TCNT 0008 8186h、TPU7.TCNT 0008 8196h、TPU8.TCNT 0008 81A6h
 TPU9.TCNT 0008 81B6h、TPU10.TCNT 0008 81C6h、TPU11.TCNT 0008 81D6h



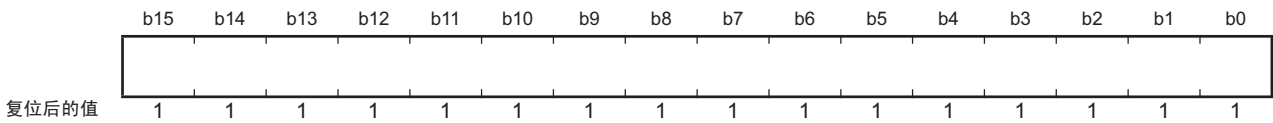
在 TPU 中，各通道有 1 个 TCNT 计数器，共计 12 个。TPUm.TCNT 计数器是能对内部时钟或者外部事件进行计数的 16 位计数器。

能以 16 位为单位进行读写。

在复位时为 “0000h”。

15.2.7 定时器的通用寄存器 A (TGRA) 定时器的通用寄存器 B (TGRB) 定时器的通用寄存器 C (TGRC) 定时器的通用寄存器 D (TGRD)

地址 TPU0.TGRA 0008 8118h、TPU0.TGRB 0008 811Ah、TPU0.TGRC 0008 811Ch、TPU0.TGRD 0008 811Eh
 TPU1.TGRA 0008 8128h、TPU1.TGRB 0008 812Ah
 TPU2.TGRA 0008 8138h、TPU2.TGRB 0008 813Ah
 TPU3.TGRA 0008 8148h、TPU3.TGRB 0008 814Ah、TPU3.TGRC 0008 814Ch、TPU3.TGRD 0008 814Eh
 TPU4.TGRA 0008 8158h、TPU4.TGRB 0008 815Ah
 TPU5.TGRA 0008 8168h、TPU5.TGRB 0008 816Ah
 TPU6.TGRA 0008 8188h、TPU6.TGRB 0008 818Ah、TPU6.TGRC 0008 818Ch、TPU6.TGRD 0008 818Eh
 TPU7.TGRA 0008 8198h、TPU7.TGRB 0008 819Ah
 TPU8.TGRA 0008 81A8h、TPU8.TGRB 0008 81AAh
 TPU9.TGRA 0008 81B8h、TPU9.TGRB 0008 81BAh、TPU9.TGRC 0008 81BCh、TPU9.TGRD 0008 81BEh
 TPU10.TGRA 0008 81C8h、TPU10.TGRB 0008 81CAh
 TPU11.TGRA 0008 81D8h、TPU11.TGRB 0008 81DAh



在 TPU 中，TPU0,3,6,9 各有 4 个定时器的通用寄存器，TPU1,2,4,5,7,8,10,11 各有 2 个定时器的通用寄存器，共计 32 个。

TPUm.TGRA (m=0 ~ 11)、TPUm.TGRB (m=0 ~ 11)、TPUm.TGRC (m=0,3,6,9)、TPUm.TGRD (m=0,3,6,9) 寄存器是 16 位输出比较 / 输入捕捉兼用的寄存器。

能以 16 位为单位进行读写。

能将 TPUm.TGRC 寄存器和 TPUm.TGRD 寄存器作为缓冲寄存器运行。在缓冲运行中，寄存器的组合为 TPUm.TGRA 寄存器和 TPUm.TGRC 寄存器、TPUm.TGRB 寄存器和 TPUm.TGRD 寄存器。

15.2.8 定时器的启动寄存器 (TSTRA、TSTRB)

地址 TSTRA 0008 8100h、TSTRB 0008 8170h

| | | | | | | | | |
|-------|----|----|------|------|------|------|------|------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | CST5 | CST4 | CST3 | CST2 | CST1 | CST0 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|-----------|--------------------------------------|-----------|
| b0 | CST0 | 计数器开始 0 位 | 0: TCNT 计数器停止计数运行 1: TCNT 计数器计数运行 | R/W |
| b1 | CST1 | 计数器开始 1 位 | | R/W |
| b2 | CST2 | 计数器开始 2 位 | | R/W |
| b3 | CST3 | 计数器开始 3 位 | | R/W |
| b4 | CST4 | 计数器开始 4 位 | | R/W |
| b5 | CST5 | 计数器开始 5 位 | | R/W |
| b7-b6 | — | 保留位 | | 读写值都为“0”。 |

TSTRA 寄存器是选择 TPU0 ~ TPU5 的 TCNT 计数器运行或者停止的寄存器。

TSTRB 寄存器是选择 TPU6 ~ TPU11 的 TCNT 计数器运行或者停止的寄存器。

在通过 TPUm.TMDR 寄存器设定运行模式或者通过 TPUm.TCR 寄存器设定 TPUm.TCNT 计数器的计数时钟时，必须先停止 TPUm.TCNT 计数器的计数器运行，再进行设定。

CSTj 位 (计数器开始位) (j=0 ~ 5)

此位选择 TCNT 计数器的运行或者停止。

在 CSTj 位为“1”并且对应的 TIOCyn 引脚 (y=A ~ D, n=0 ~ 11) 为输出状态时，如果将 CSTj 位置“0”，就停止计数运行并且保持对应的 TIOCyn 引脚的输出比较的输出电平。

如果在 CSTj 位为“0”时写 TIORH、TIORL、TIOR 寄存器，就将引脚的输出电平更新为已设定的初始输出值。

15.2.9 定时器的同步寄存器 (TSYRA、TSYRB)

地址 TSYRA 0008 8101h、TSYRB 0008 8171h

| | | | | | | | | |
|-------|----|----|-------|-------|-------|-------|-------|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | SYNC5 | SYNC4 | SYNC3 | SYNC2 | SYNC1 | SYNC0 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|------------|---|-----------|
| b0 | SYNC0 | 定时器的同步 0 位 | 0: TCNT 计数器进行独立运行 (TCNT 计数器的预置 / 清除与其他通道无关) 1: TCNT 计数器进行同步运行 (注 1) (能进行 TCNT 计数器的同步预置 / 同步清除) | R/W |
| b1 | SYNC1 | 定时器的同步 1 位 | | R/W |
| b2 | SYNC2 | 定时器的同步 2 位 | | R/W |
| b3 | SYNC3 | 定时器的同步 3 位 | | R/W |
| b4 | SYNC4 | 定时器的同步 4 位 | | R/W |
| b5 | SYNC5 | 定时器的同步 5 位 | | R/W |
| b7-b6 | — | 保留位 | | 读写值都为“0”。 |

注 1. 要设定同步运行时，至少需要将 2 个通道的 SYNC_j 位 (j=0 ~ 5) 置“1”；要设定同步清除时，除了 SYNC_j 位以外，还需要通过 TCR.CCLR[2:0] 位选择 TCNT 的清除源。

TSYRA 寄存器选择 TPU0 ~ TPU5 的 TCNT 计数器的独立运行或者同步运行。

TSYRB 寄存器选择 TPU6 ~ TPU11 的 TCNT 计数器的独立运行或者同步运行。

SYNC_j 位 (定时器的同步位) (j=0 ~ 5)

此位选择 TCNT 计数器是和其他通道的 TCNT 计数器进行同步运行还是进行独立运行。

如果选择同步运行，就能对多个 TCNT 计数器进行同步预置并且通过清除其他通道的计数器进行同步清除。

15.3 运行说明

15.3.1 概要

各通道有 TPUm.TCNT 计数器和 TPUm.TGRy 寄存器 (y=A ~ D)。

TCNT 计数器是 16 位递增计数器，能用作自由运行计数器、周期计数器或者事件计数器。

TGRy 寄存器能分别用作输入捕捉寄存器或者输出比较寄存器。

(1) 计数器的运行

如果将 TSTRA.CSTj 位 (j=0 ~ 5) 或者 TSTRB.CSTj 位 (j=0 ~ 5) 置“1”，对应通道的 TCNT 计数器就开始计数运行。

(a) 计数运行的设定步骤例子

计数器运行的设定步骤例子如图 15.3 所示。

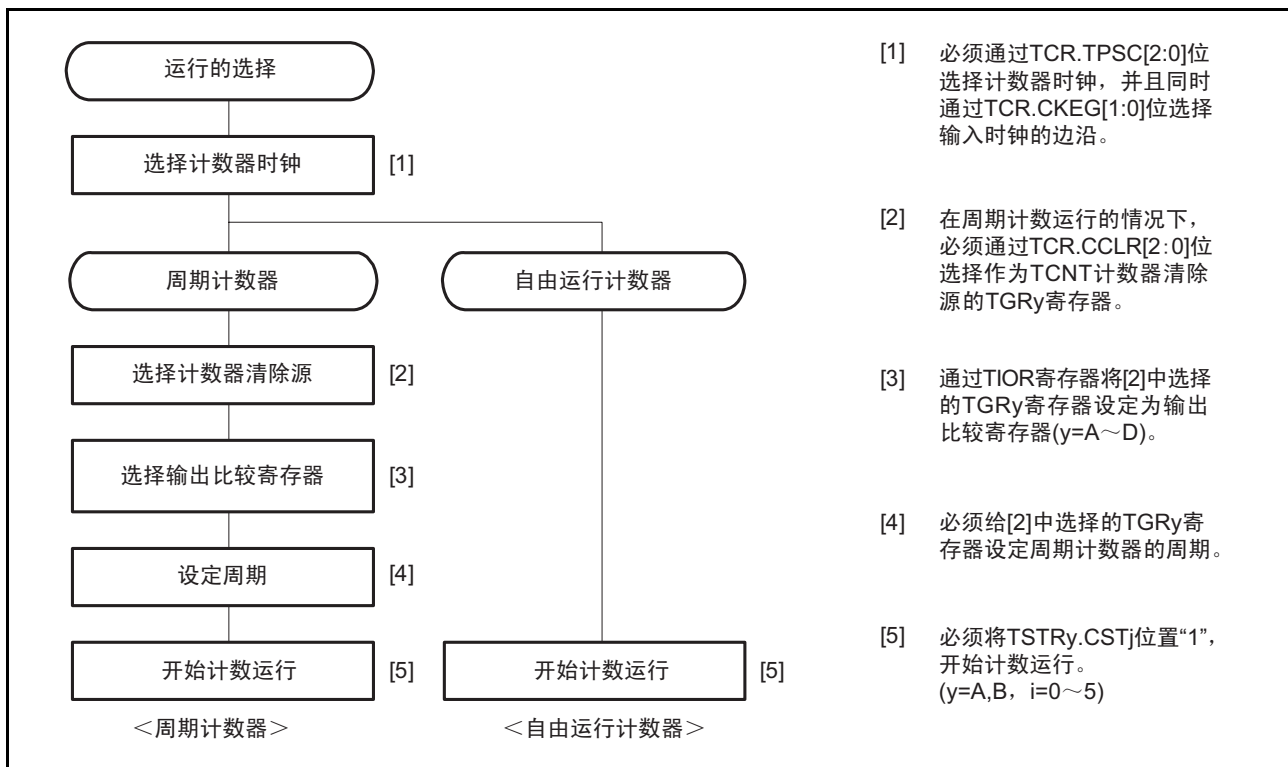


图 15.3 计数器运行的设定步骤例子

(b) 自由运行计数的运行和周期计数的运行

TPUm.TCNT 计数器在复位后立即被全部设定为自由运行计数器，如果将 TSTRA 寄存器或者 TSTRB 寄存器的对应位置“1”，就作为自由运行计数器开始递增计数。当 TCNT 计数器发生上溢 (“FFFFh” → “0000h”) 时，TPU 就请求中断。TCNT 计数器在发生上溢后从 “0000h” 开始继续递增计数。

自由运行计数器的运行如图 15.4 所示。

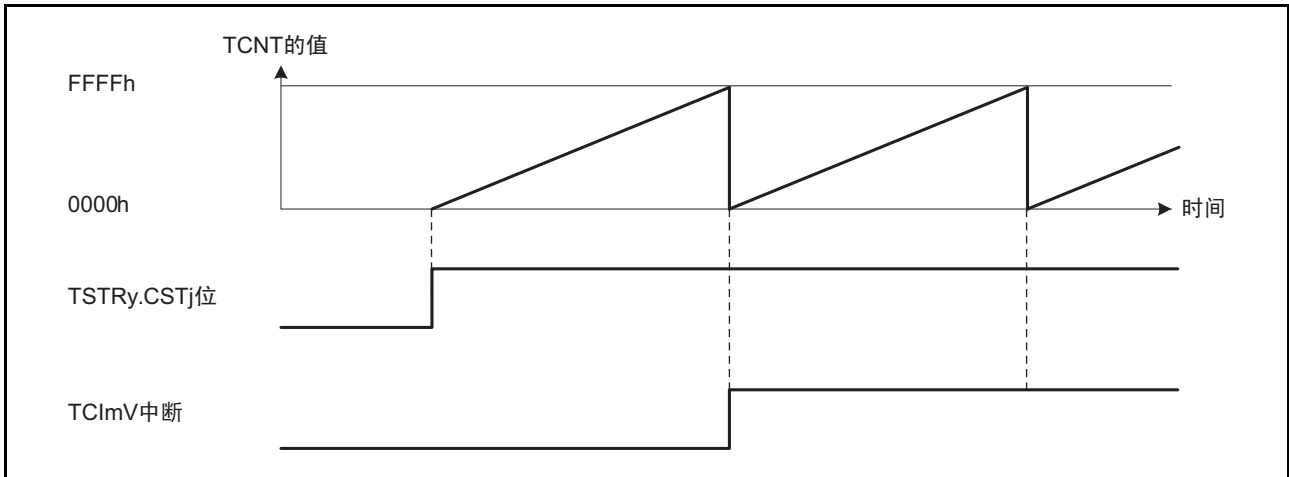


图 15.4 自由运行计数器的运行

当选择比较匹配为 TCNT 计数器的清除源时，对应通道的 TCNT 计数器就进行周期计数运行。将用于周期设定的 TPUm.TGRy 寄存器设定为输出比较寄存器，并且通过 TPUm.TCR.CCLR[2:0] 位选择在比较匹配时清除计数器。在设定后，如果将 TSTRA 寄存器或者 TSTRB 寄存器的对应位置“1”，就作为周期计数器开始递增计数。当计数值和 TGRy 寄存器的值相同时，TCNT 计数器就变为 “0000h”。

此时，TPU 请求中断。TCNT 计数器在比较匹配后从 “0000h” 开始继续递增计数。

周期计数器的运行如图 15.5 所示。

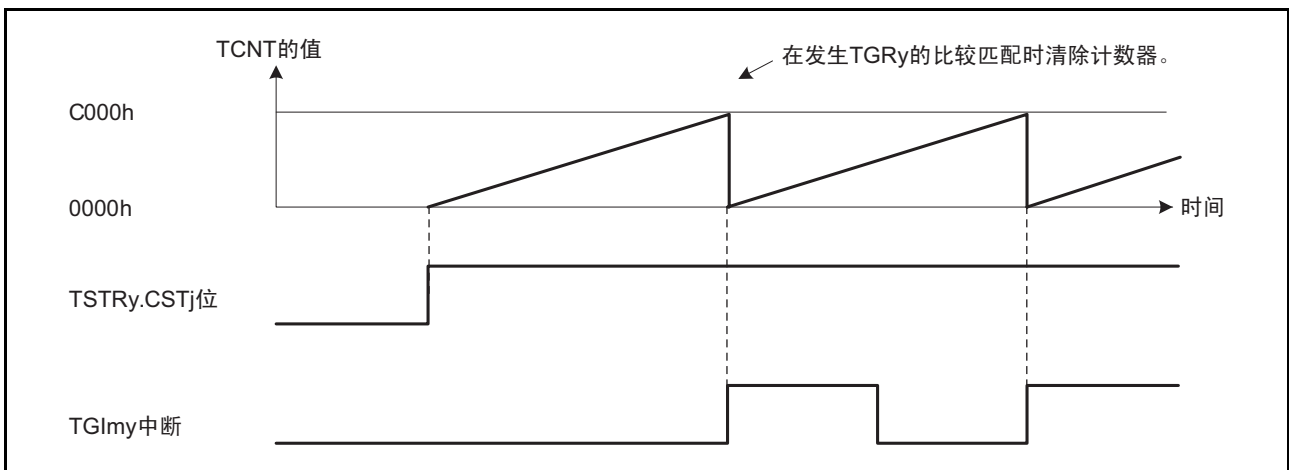


图 15.5 周期计数器的运行

(2) 通过比较匹配进行的波形输出功能

TPU 能通过比较匹配从对应的输出引脚输出 Low 电平或者 High 电平或者进行交替输出。

(a) 通过比较匹配进行波形输出运行的设定步骤例子

通过比较匹配进行波形输出运行的设定步骤例子如图 15.6 所示。

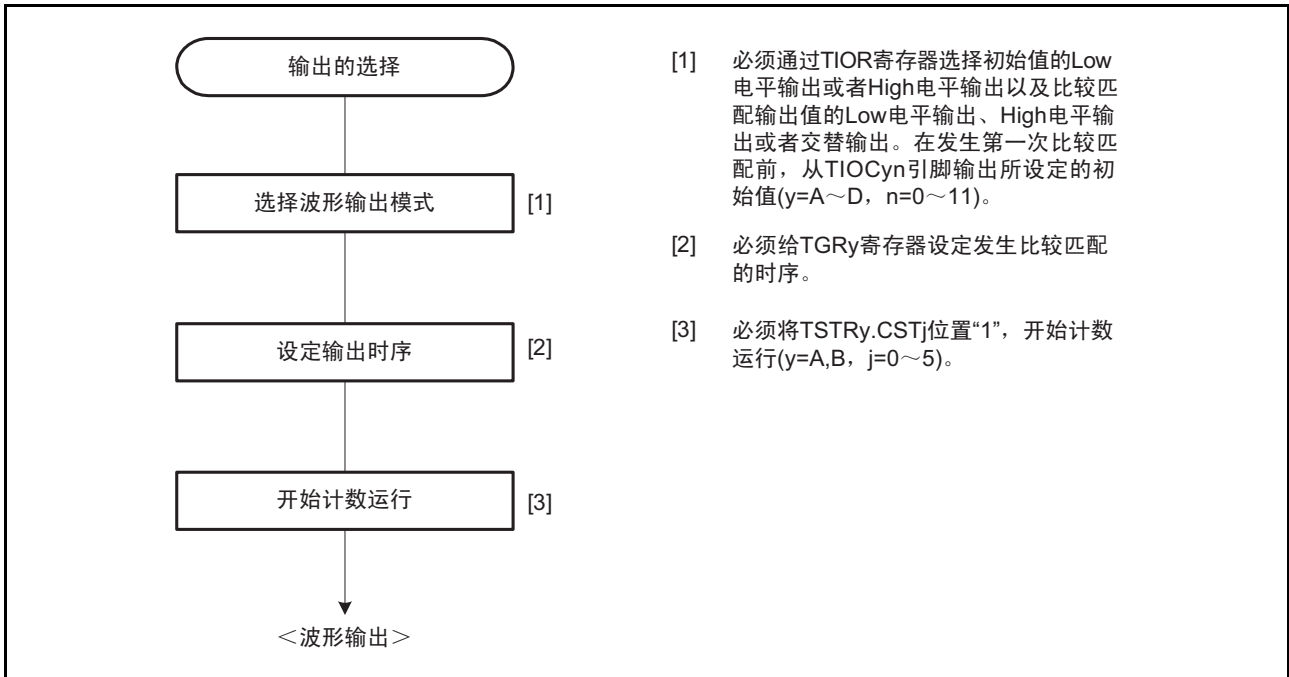


图 15.6 通过比较匹配进行波形输出的运行例子

(b) 波形输出的运行例子

输出 Low 电平 /High 电平的运行例子如图 15.7 所示。

在此例子中，设定 TPUm.TCNT 为自由运行的计数运行，在比较匹配 A 时输出 High 电平，在比较匹配 B 时输出 Low 电平。如果设定的电平和引脚的电平相同，引脚的电平就不变。

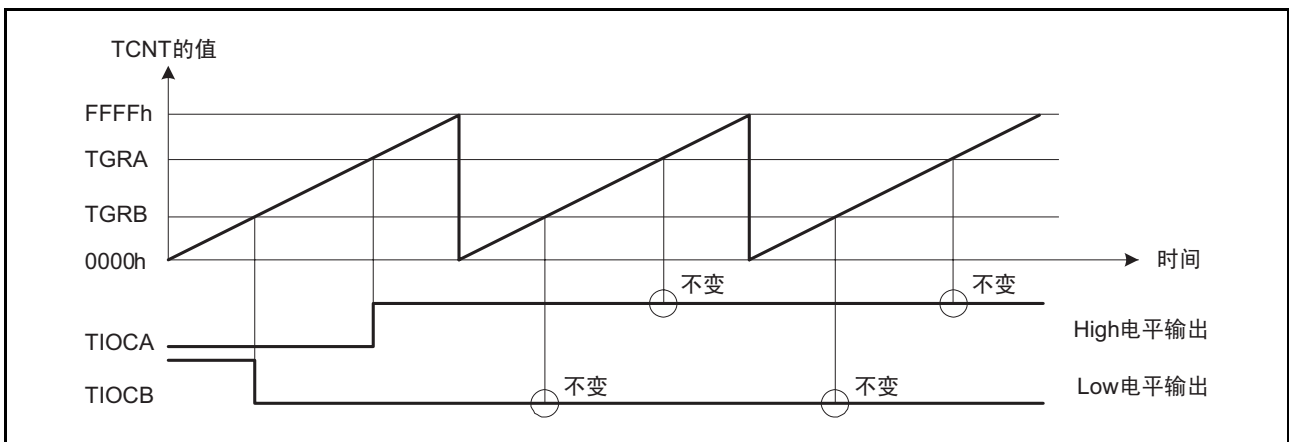


图 15.7 输出 Low 电平 /High 电平的运行例子

交替输出的运行例子如图 15.8 所示。

在此例子中，设定 TPUm.TCNT 计数器为周期计数运行（在比较匹配 B 时清除计数器），并且将比较匹配 A 和比较匹配 B 都设定为交替输出。

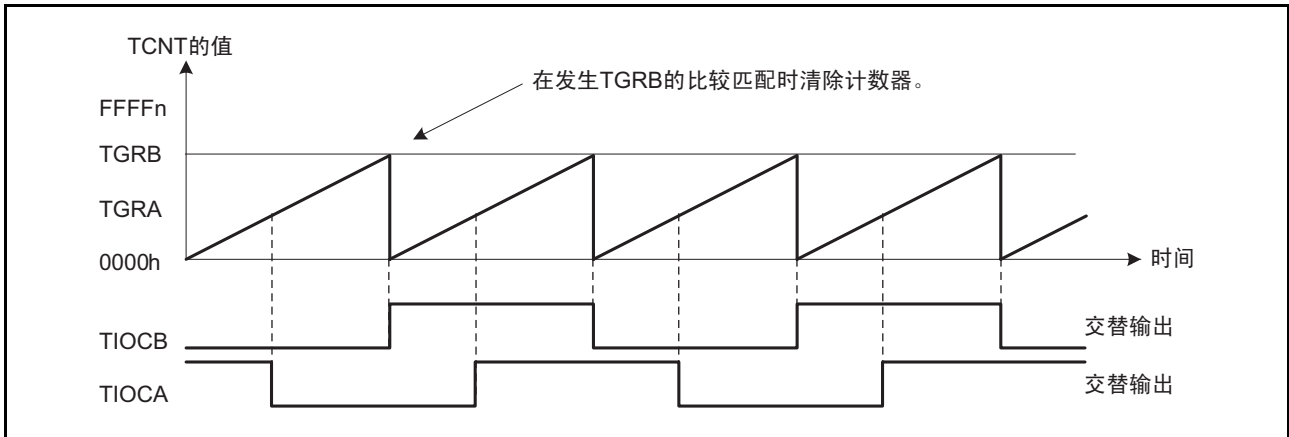


图 15.8 交替输出的运行例子

(3) 输入捕捉功能

能检测 TIOCy_n 引脚 (y=A ~ D, n=0 ~ 11) 的输入边沿，并且将 TPUm.TCNT 计数器的值传送到 TPUm.TGRy 寄存器。

能从上升沿、下降沿和双边沿中选择检测边沿，也能将 TPU0,1,3,4 (TPU6,7,9,10) 的计数器输入时钟或者比较匹配信号作为输入捕捉源。

注 1. 如果在 TPU0 和 TPU3 (TPU6 和 TPU9) 中将其他通道的计数器输入时钟作为输入捕捉的输入，就不能选择 PCLK/1 作为输入捕捉的输入的计数器输入时钟，否则就不发生输入捕捉。

(a) 输入捕捉运行的设定步骤例子

输入捕捉运行的设定步骤例子如图 15.9 所示。

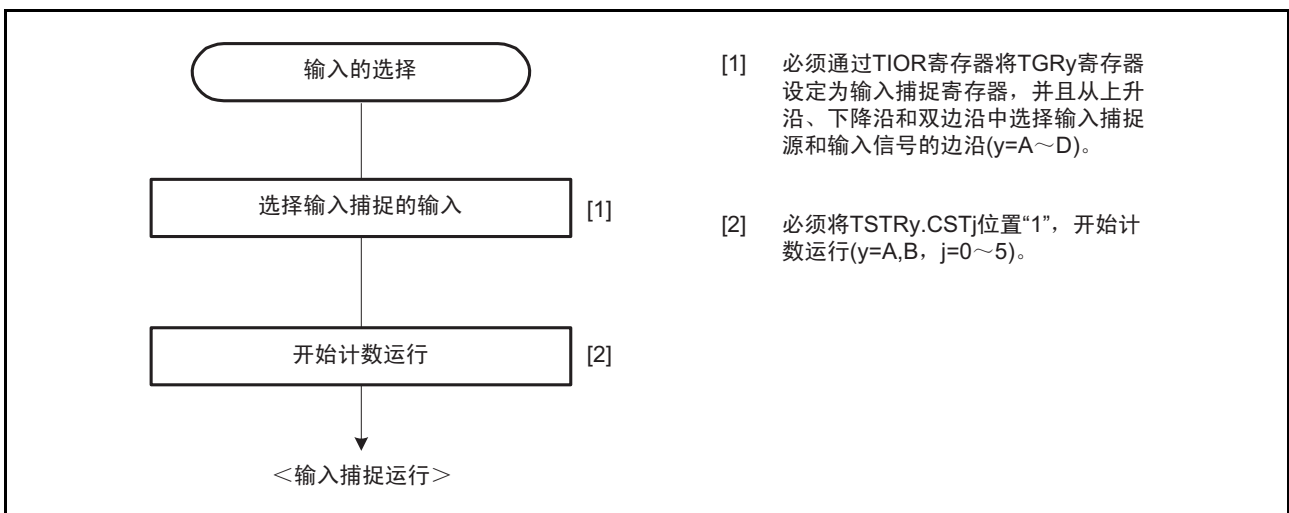


图 15.9 输入捕捉运行的设定例子

(b) 输入捕捉的运行例子

输入捕捉的运行例子如图 15.10 所示。

在此例子中，设定 TIOCA_n 引脚的输入捕捉的输入边沿为上升沿和下降沿的双边沿， TIOCB_n 引脚的输入捕捉的输入边沿为下降沿，并且在发生 TPU_m.TGRB 寄存器的输入捕捉时清除 TPU_m.TCNT 计数器。

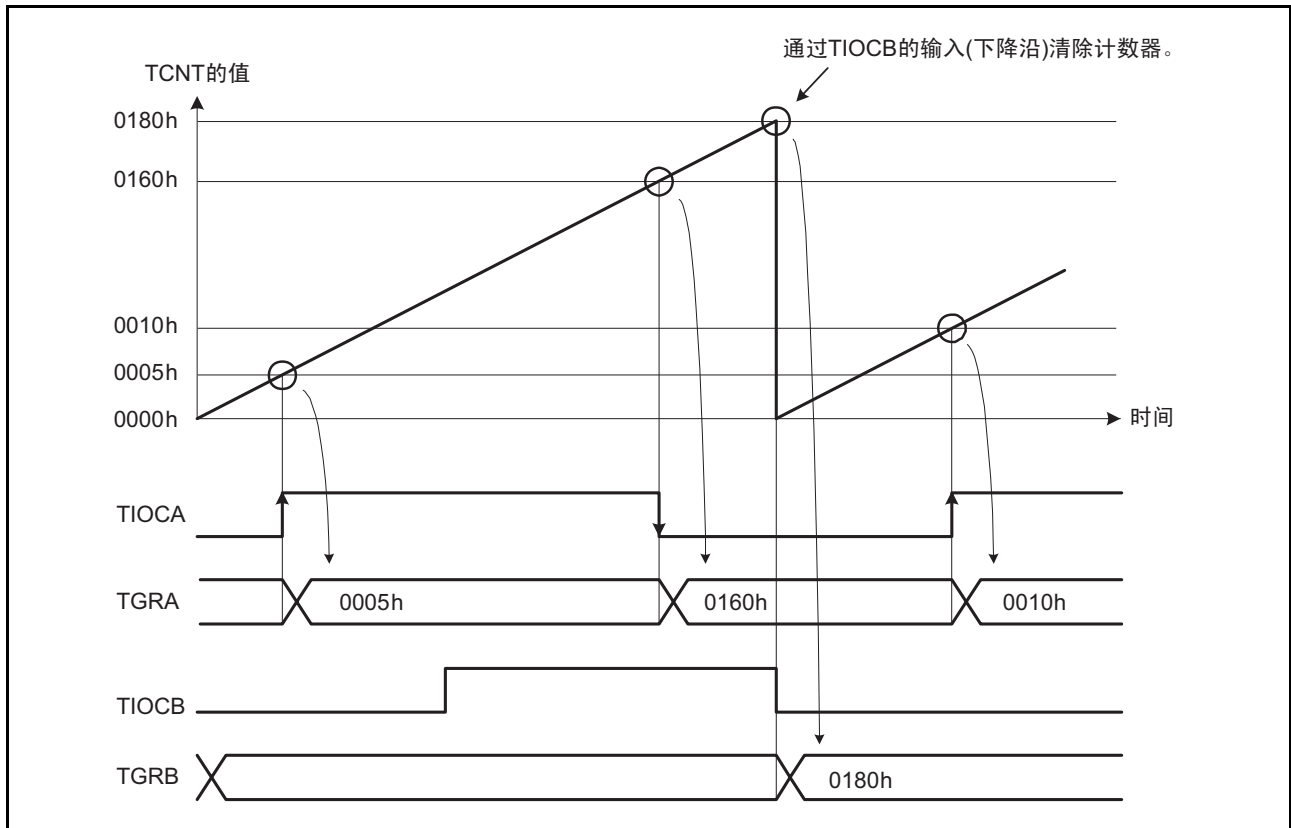


图 15.10 输入捕捉的运行例子

15.3.2 同步运行

使用同步运行，能同时改写多个 TPUm.TCNT 计数器的值（同步预置），并且能通过设定 TPUm.TCR 寄存器，同时清除多个 TCNT 计数器（同步清除）。

能通过同步运行对 1 个时基增加 TPUm.TGRy 寄存器。

TPU0 ~ TPU5（TPU6 ~ TPU11）都能设定为同步运行。

(1) 同步运行的设定步骤例子

同步运行的设定步骤例子如图 15.11 所示。

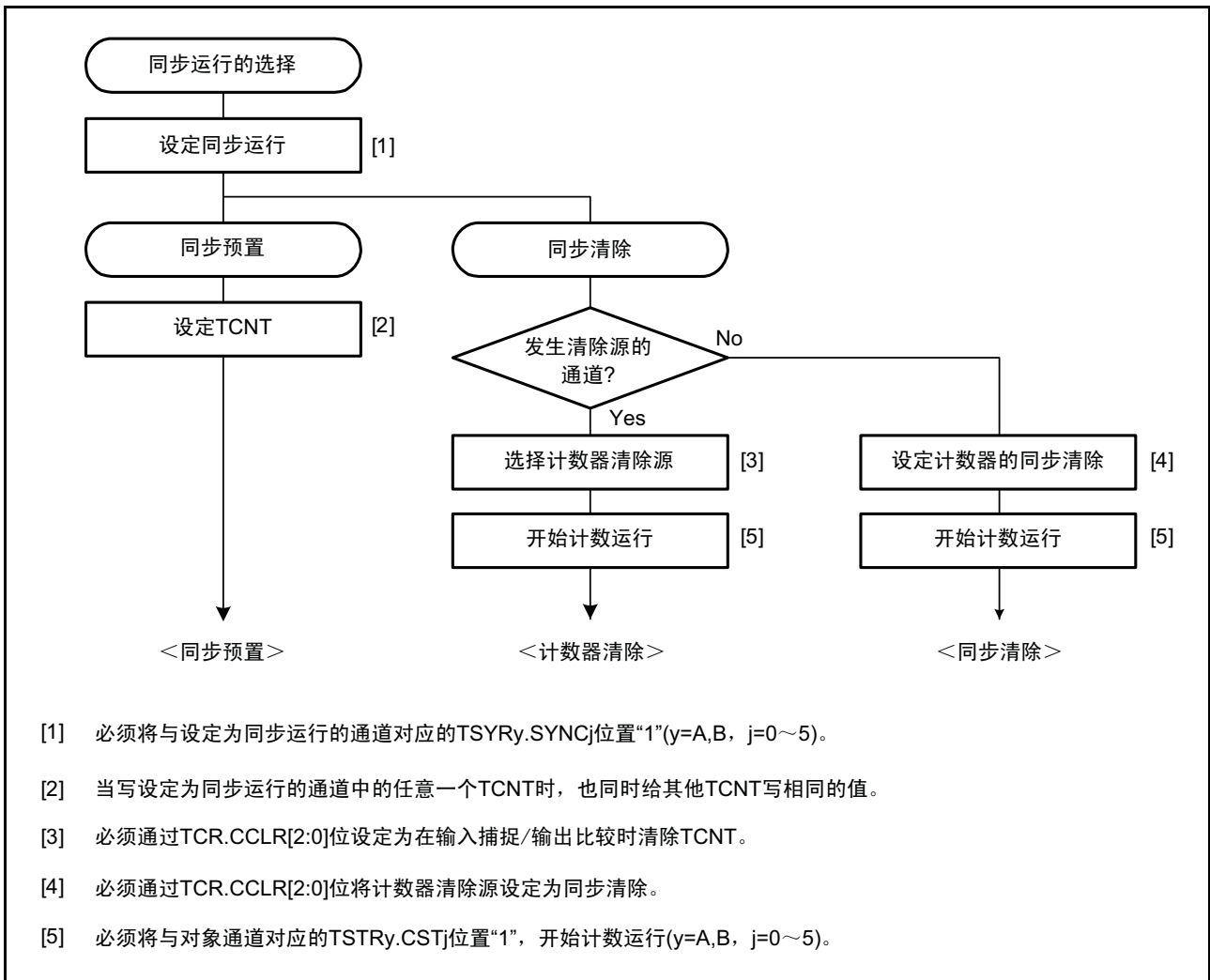


图 15.11 同步运行的设定步骤例子

(2) 同步运行的例子

同步运行的运行例子如图 15.12 所示。

在此例子中，设定 TPU0 ~ TPU2 为同步运行和 PWM 模式 1，TPU0 的计数器清除源为 TPU0.TGRB 寄存器的比较匹配，TPU1 和 TPU2 的计数器清除源为同步清除。

从 TIOCA0、TIOCA1、TIOCA2 引脚输出 3 相 PWM 波形。此时，对 TPU0 ~ TPU2 的 TPU_m.TCNT 进行同步预置并且在发生 TPU0.TGRB 寄存器的比较匹配时进行同步清除，TPU0.TGRB 寄存器设定的数据为 PWM 周期。

有关 PWM 模式请参照“15.3.5 PWM 模式”。

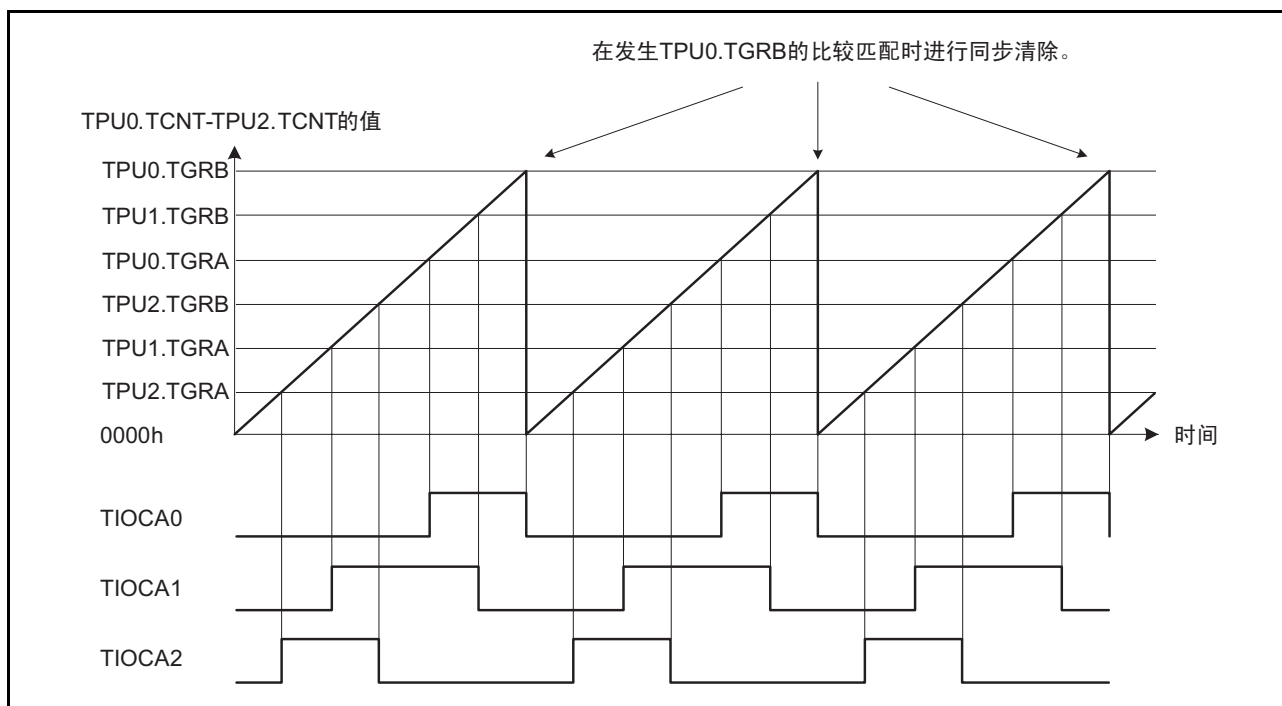


图 15.12 同步运行的例子

15.3.3 缓冲运行

缓冲运行是 TPU0 和 TPU3 (TPU6 和 TPU9) 具有的功能。能将 TPU_m.TGRC 寄存器和 TPU_m.TGRD 寄存器用作缓冲寄存器。

对于是将 TPU_m.TGRy 寄存器设定为输入捕捉寄存器还是设定为比较匹配寄存器，缓冲运行的内容不同。缓冲运行时的寄存器组合如表 15.23 所示。

表 15.23 寄存器的组合

| 单元 | 通道 | 定时器的通用寄存器 | 缓冲寄存器 |
|----|------|-----------|-----------|
| 0 | TPU0 | TPU0.TGRA | TPU0.TGRC |
| | | TPU0.TGRB | TPU0.TGRD |
| | TPU3 | TPU3.TGRA | TPU3.TGRC |
| | | TPU3.TGRB | TPU3.TGRD |
| 1 | TPU6 | TPU6.TGRA | TPU6.TGRC |
| | | TPU6.TGRB | TPU6.TGRD |
| | TPU9 | TPU9.TGRA | TPU9.TGRC |
| | | TPU9.TGRB | TPU9.TGRD |

- TPU_m.TGRy 寄存器为输出比较寄存器的情况
如果发生比较匹配，就将对应通道的缓冲寄存器的值传送到定时器的通用寄存器。
此运行如图 15.13 所示。

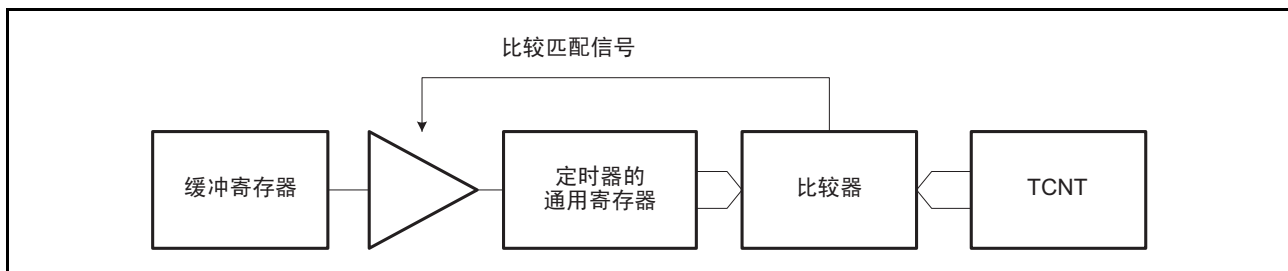


图 15.13 比较匹配的缓冲运行

- TPU_m.TGRy 寄存器为输入捕捉寄存器的情况
如果发生输入捕捉，就在将 TPU_m.TCNT 计数器的值传送到 TGRy 寄存器的同时，将以前保存在 TGRy 寄存器的值传送到缓冲寄存器。
此运行如图 15.14 所示。

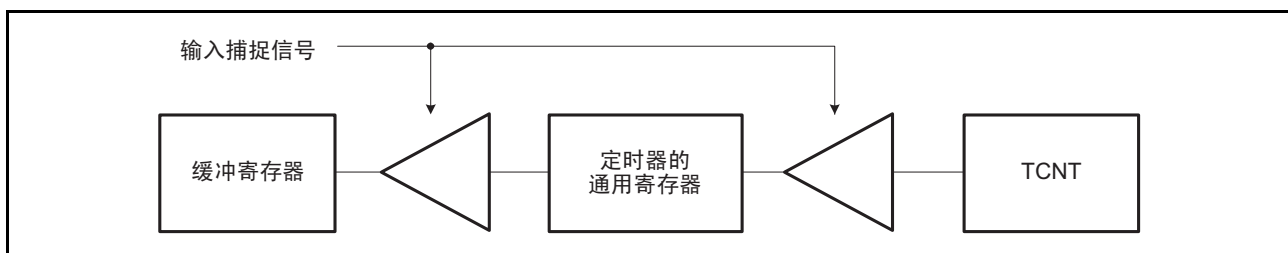


图 15.14 输入捕捉缓冲运行

(1) 缓冲运行的设定步骤例子

缓冲运行的设定步骤例子如图 15.15 所示。

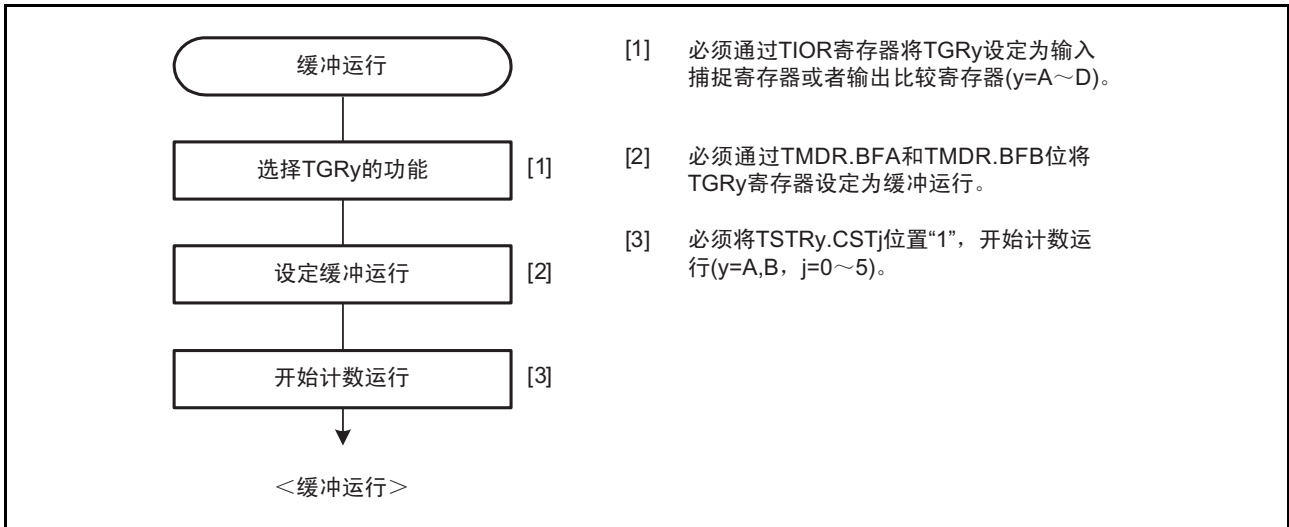


图 15.15 缓冲运行的设定步骤例子

(2) 缓冲运行的例子

(a) TPUm.TGRy 寄存器为输出比较寄存器的情况

设定 TPU0 为 PWM 模式 1 以及 TPU0.TGRA 寄存器和 TPU0.TGRC 寄存器为缓冲运行时的运行例子如图 15.16 所示。在此例子中，通过比较匹配 B 清除 TPU0.TCNT 计数器，在比较匹配 A 时输出 High 电平，在比较匹配 B 时输出 Low 电平。

因为设定了缓冲运行，所以当发生比较匹配 A 时，就在输出发生变化的同时将 TPU0.TGRC 寄存器的值传送到 TPU0.TGRA 寄存器。每当发生比较匹配 A 时，重复此操作。

有关 PWM 模式请参照“15.3.5 PWM 模式”。

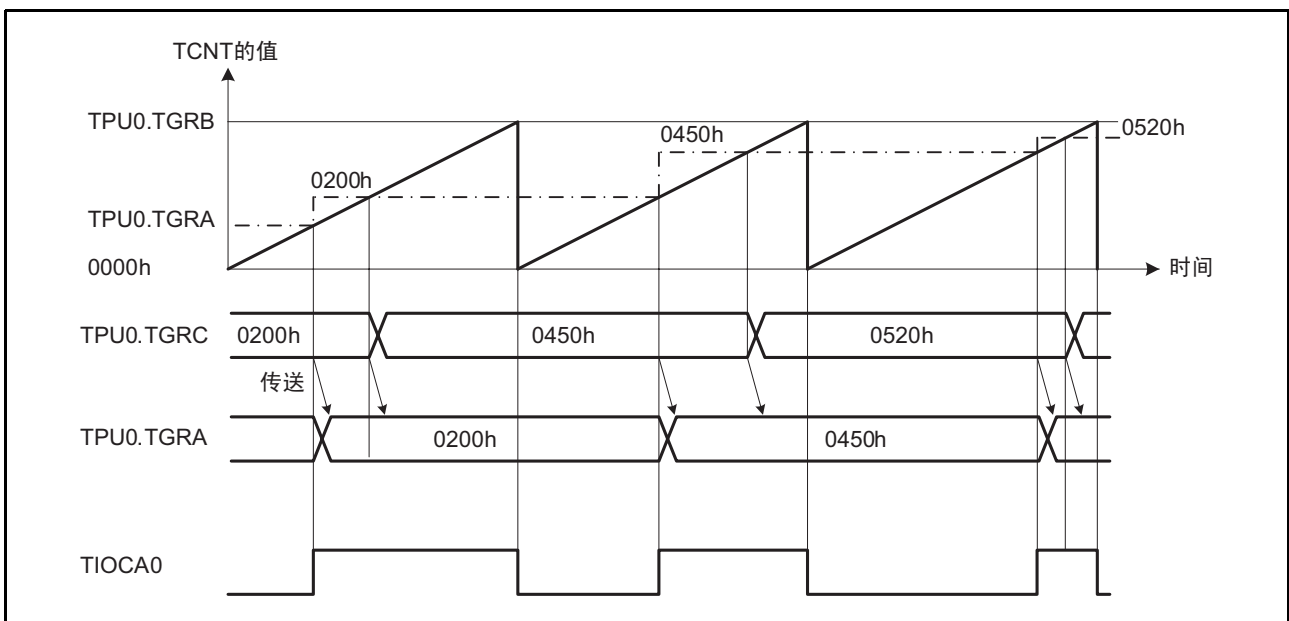


图 15.16 缓冲运行的例子 (1)

(b) TPUm.TGRy 寄存器为输入捕捉寄存器的情况

设定 TPUm.TGRA 寄存器为输入捕捉寄存器以及 TGRA 寄存器和 TPUm.TGRC 寄存器为缓冲运行时的运行例子如图 15.17 所示。

在 TGRA 寄存器的输入捕捉时清除 TPUm.TCNT 计数器，TIOCA_n 引脚的输入捕捉的输入边沿选择上升沿和下降沿的双边沿。

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

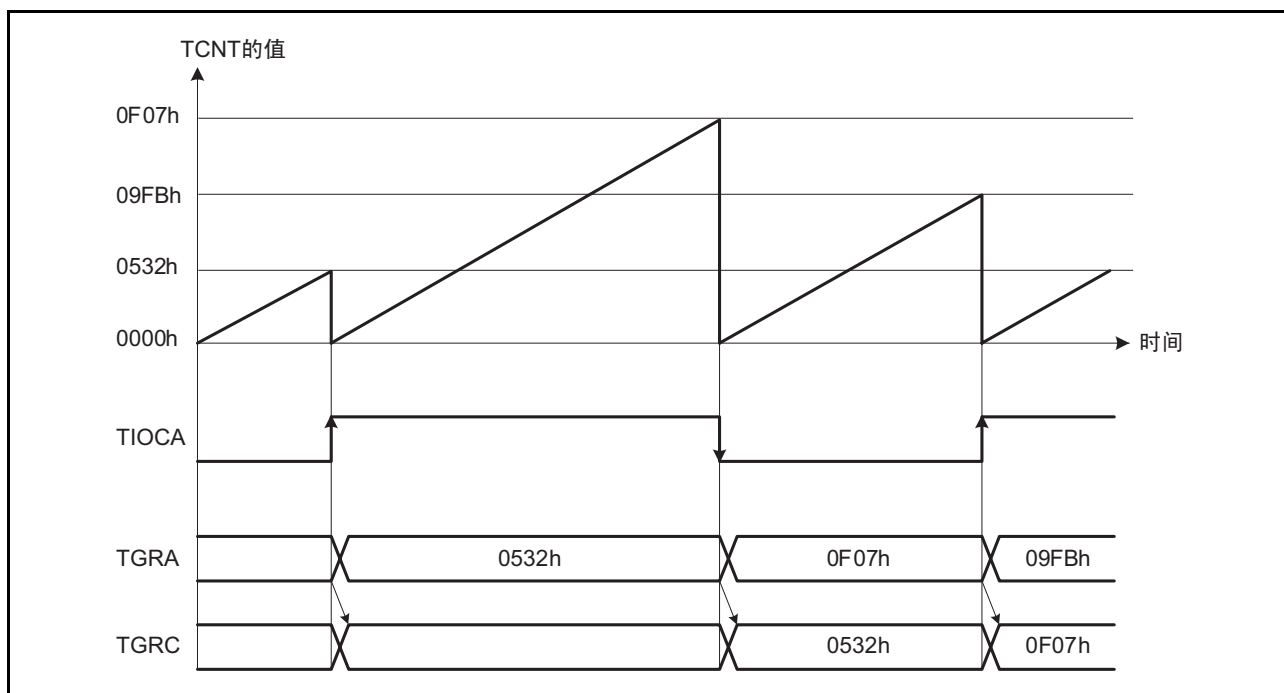


图 15.17 缓冲运行的例子 (2)

15.3.4 级联运行

级联运行是将 2 个通道的 16 位计数器连接为 32 位计数器的功能。

在单元 0 的情况下，通过 TPU1.TCR.TPSC[2:0] (TPU4.TCR.TPSC[2:0]) 位将 TPU1 (TPU4) 的计数器时钟设定为通过 TPU2.TCNT (TPU5.TCNT) 计数器的上溢 / 下溢进行计数。

在单元 1 的情况下，通过 TPU7.TCR.TPSC[2:0] (TPU10.TCR.TPSC[2:0]) 位将 TPU7 (TPU10) 的计数器时钟设定为通过 TPU8.TCNT (TPU11.TCNT) 计数器的上溢 / 下溢进行计数。

只在低 16 位的 TPU_m.TCNT 计数器为相位计数模式时发生下溢。

级联的组合如表 15.24 所示。

注 1. 如果将 TPU1 和 TPU4 (TPU7 和 TPU10) 设定为相位计数模式，计数器时钟的设定就无效，并且在相位计数模式中独立运行。

表 15.24 级联的组合

| 单元 | 组合 | 高 16 位 | 低 16 位 |
|----|---------------|------------|------------|
| 0 | TPU1 和 TPU2 | TPU1.TCNT | TPU2.TCNT |
| | TPU4 和 TPU5 | TPU4.TCNT | TPU5.TCNT |
| 1 | TPU7 和 TPU8 | TPU7.TCNT | TPU8.TCNT |
| | TPU10 和 TPU11 | TPU10.TCNT | TPU11.TCNT |

(1) 级联运行的设定步骤例子

级联运行的设定步骤例子如图 15.18 所示。

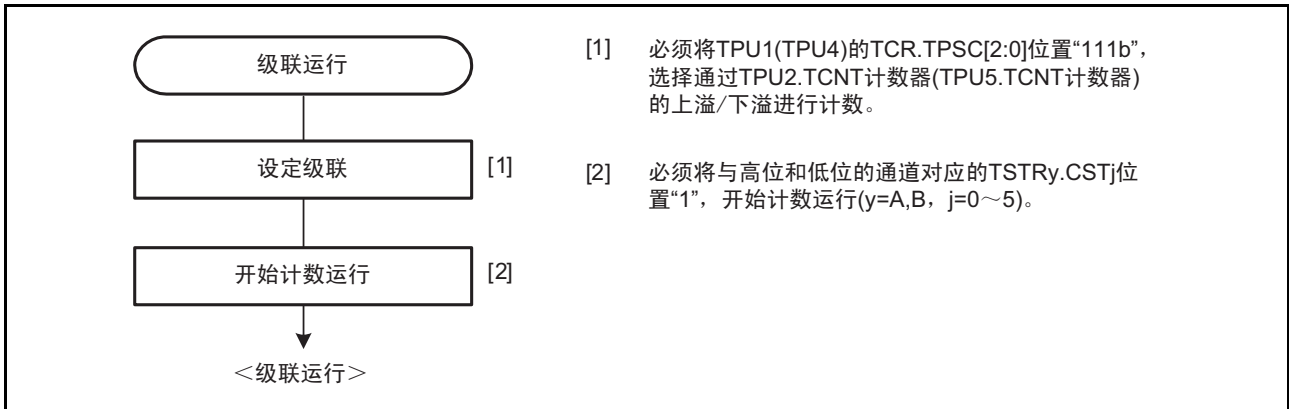


图 15.18 级联运行的设定步骤例子

(2) 级联的运行例子

级联的运行例子 (1) 如图 15.19 所示, 在此例中, TPU1.TCNT 计数器通过 TPU2.TCNT 计数器的上溢 / 下溢进行计数, 将 TPU1.TGRA 寄存器和 TPU2.TGRA 寄存器设定为输入捕捉寄存器, 选择 TIOCA1 引脚和 TIOCA2 引脚的上升沿。

通过给 TIOCA1 引脚和 TIOCA2 引脚同时输入上升沿, 将高 16 位和低 16 位的 32 位数据分别传送到 TPU1.TGRA 寄存器和 TPU2.TGRA 寄存器。

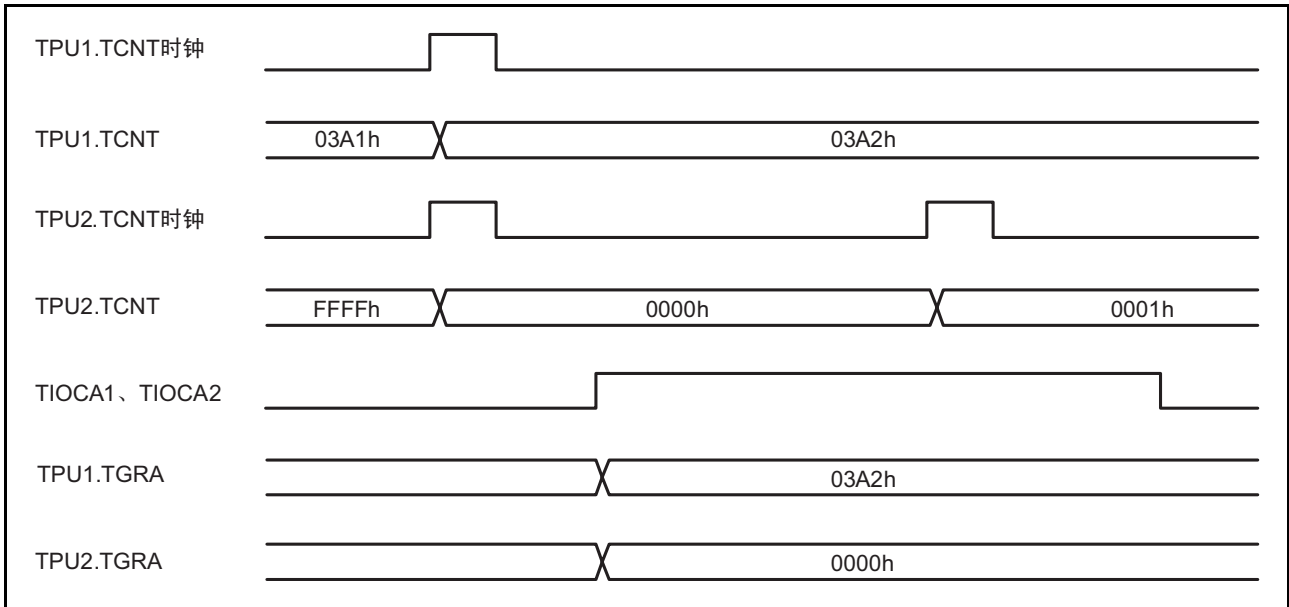


图 15.19 级联运行的例子 (1)

级联运行例子 (2) 如图 15.20 所示, 在此例中, TPU1.TCNT 计数器通过 TPU2.TCNT 计数器的上溢 / 下溢进行计数, 将 TPU2 设定为相位计数模式。

TPU1.TCNT 计数器通过 TPU2.TCNT 计数器的上溢进行递增计数, 并且通过 TPU2.TCNT 计数器的下溢进行递减计数。

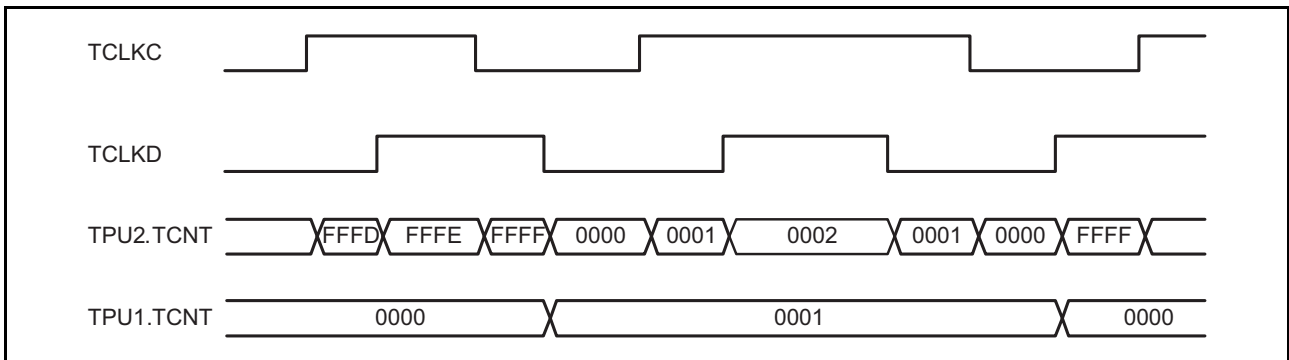


图 15.20 级联运行的例子 (2)

15.3.5 PWM 模式

PWM 模式是从输出引脚分别输出各种 PWM 波形的模式。能从 Low 电平输出、High 电平输出和交替输出中选择各 TPUm.TGRy 寄存器的比较匹配的输出来电。

能通过设定各 TGRy 寄存器，输出 0 ~ 100% 占空比的 PWM 波形。

能通过将 TGRy 寄存器的比较匹配作为计数器清除源，给该寄存器设定周期。全部通道能分别设定为 PWM 模式，也能进行同步运行。

PWM 模式有以下 2 种：

1. PWM 模式 1

TPUm.TGRA 寄存器和 TPUm.TGRB 寄存器、TPUm.TGRC 寄存器和 TPUm.TGRD 寄存器配对使用，从 TIOCA_n 引脚和 TIOCC_n 引脚输出 PWM 波形。通过比较匹配 A 和比较匹配 C 从 TIOCA_n 引脚和 TIOCC_n 引脚进行 TPUm.TIOR(H).IOA[3:0] 位和 TPUm.TIORL.IOC[3:0] 位指定的输出，并且通过比较匹配 B 和比较匹配 D 从 TIOCA_n 引脚和 TIOCC_n 引脚进行 TPUm.TIOR(H).IOB[3:0] 位和 TPUm.TIORL.IOD[3:0] 位指定的输出。初始输出值为 TGRA 寄存器和 TGRC 寄存器的设定值。在配对使用的 TGRy 寄存器的设定值相同的情况下，即使发生比较匹配，输出值也不变。

在 PWM 模式 1 中，能进行最多 8 相的 PWM 输出。

2. PWM 模式 2

将 1 个 TPUm.TGRy 寄存器用于周期寄存器，其他 TGRy 寄存器用于占空比寄存器，进行 PWM 波形输出。通过比较匹配进行 TPUm.TIORH、TPUm.TIORL、TPUm.TIOR 寄存器指定的输出。另外，在通过同步寄存器的比较匹配进行计数器的清除后，各引脚的输出值为 TIORH、TIORL、TIOR 寄存器设定的初始值。在周期寄存器和占空比寄存器的设定值相同的情况下，即使发生比较匹配，输出值也不变。

在 PWM 模式 2 中，能通过和同步运行的并用来进行最多 15 相的 PWM 波形输出。

PWM 的输出引脚和寄存器的对应如表 15.25 所示。

表 15.25 各 PWM 输出的寄存器和输出引脚

| 单元 | 通道 | 寄存器 | 输出引脚 | | |
|-------|------|------------|-----------|----------|--------|
| | | | PWM 模式 1 | PWM 模式 2 | |
| 0 | TPU0 | TPU0.TGRA | TIOCA0 | TIOCA0 | |
| | | TPU0.TGRB | | TIOCB0 | |
| | | TPU0.TGRC | TIOCC0 | TIOCC0 | |
| | | TPU0.TGRD | | TIOCD0 | |
| | TPU1 | TPU1.TGRA | TIOCA1 | TIOCA1 | |
| | | TPU1.TGRB | | TIOCB1 | |
| | TPU2 | TPU2.TGRA | TIOCA2 | TIOCA2 | |
| | | TPU2.TGRB | | TIOCB2 | |
| | TPU3 | TPU3.TGRA | TIOCA3 | TIOCA3 | |
| | | TPU3.TGRB | | TIOCB3 | |
| | | TPU3.TGRC | TIOCC3 | TIOCC3 | |
| | | TPU3.TGRD | | TIOCD3 | |
| | TPU4 | TPU4.TGRA | TIOCA4 | TIOCA4 | |
| | | TPU4.TGRB | | TIOCB4 | |
| | TPU5 | TPU5.TGRA | TIOCA5 | TIOCA5 | |
| | | TPU5.TGRB | | TIOCB5 | |
| | 1 | TPU6 | TPU6.TGRA | TIOCA6 | TIOCA6 |
| | | | TPU6.TGRB | | TIOCB6 |
| | | | TPU6.TGRC | TIOCC6 | TIOCC6 |
| | | | TPU6.TGRD | | TIOCD6 |
| TPU7 | | TPU7.TGRA | TIOCA7 | TIOCA7 | |
| | | TPU7.TGRB | | TIOCB7 | |
| TPU8 | | TPU8.TGRA | TIOCA8 | TIOCA8 | |
| | | TPU8.TGRB | | TIOCB8 | |
| TPU9 | | TPU9.TGRA | TIOCA9 | TIOCA9 | |
| | | TPU9.TGRB | | TIOCB9 | |
| | | TPU9.TGRC | TIOCC9 | TIOCC9 | |
| | | TPU9.TGRD | | TIOCD9 | |
| TPU10 | | TPU10.TGRA | TIOCA10 | TIOCA10 | |
| | | TPU10.TGRB | | TIOCB10 | |
| TPU11 | | TPU11.TGRA | TIOCA11 | TIOCA11 | |
| | | TPU11.TGRB | | TIOCB11 | |

注 1. 在 PWM 模式 2 中，不能对已设定周期的 TPU_m.TGR_y 寄存器进行 PWM 波形输出。

(1) PWM 模式的设定步骤例子

PWM 模式的设定步骤例子如图 15.21 所示。

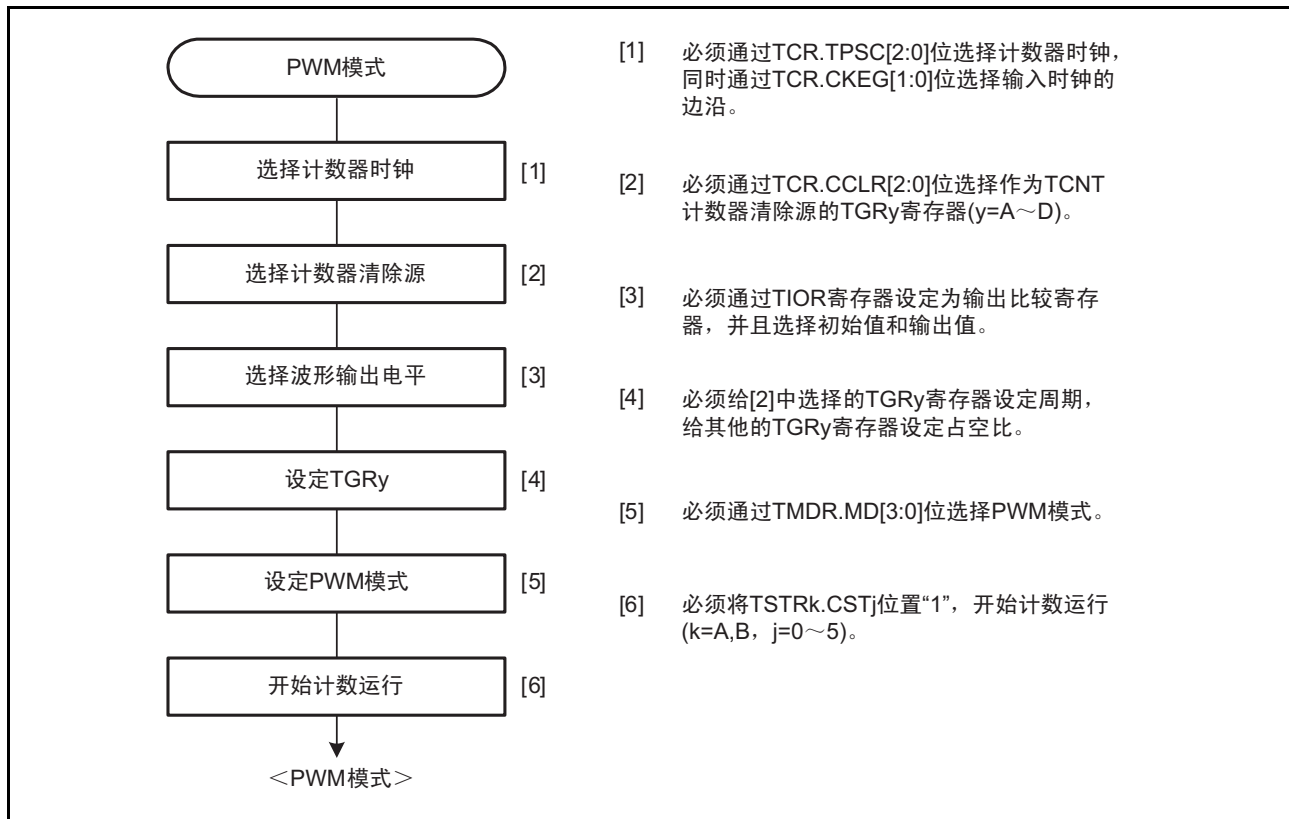


图 15.21 PWM 模式的设定步骤例子

(2) PWM 模式的运行例子

PWM 模式 1 的运行例子如图 15.22 所示。

在此例子中，将 TPUm.TGRA 寄存器的比较匹配作为 TPUm.TCNT 计数器的清除源，将 TGRA 寄存器的初始输出值和输出值置“0”，并且将 TPUm.TGRB 寄存器的输出值置“1”。

此时，TGRA 寄存器的设定值为周期，TGRB 寄存器的设定值为占空比。

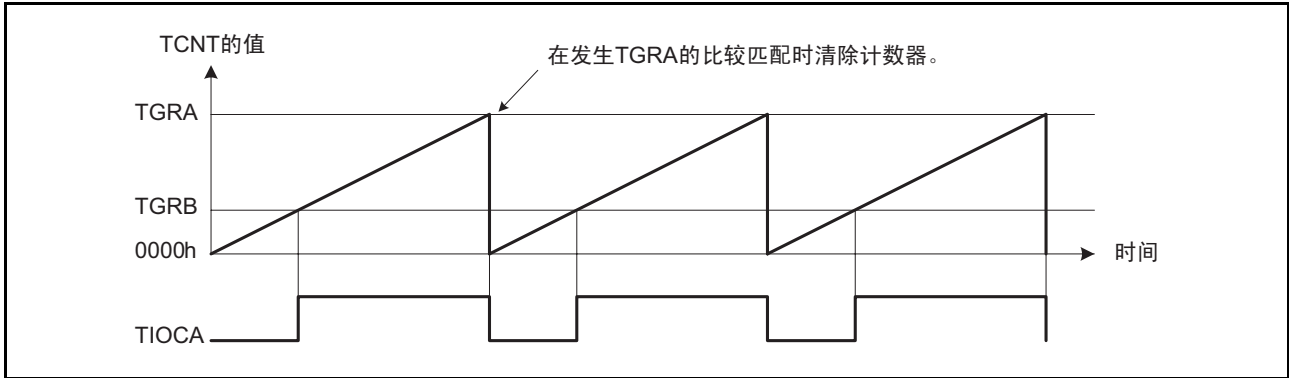


图 15.22 PWM 模式的运行例子 (1)

PWM 模式 2 的运行例子如图 15.23 所示。

在此例子中，TPU0 和 TPU1 同步运行，将 TPU1.TGRB 的比较匹配作为 TPUm.TCNT 计数器的清除源，将其他 TPUm.TGRy 寄存器 (TPU0.TGRA ~ TPU0.TGRD、TPU1.TGRA) 的初始输出值和输出值分别置“0”和置“1”，输出 5 相的 PWM 波形。

此时，TPU1.TGRB 寄存器的设定值为周期，其他 TGRy 寄存器的设定值为占空比。

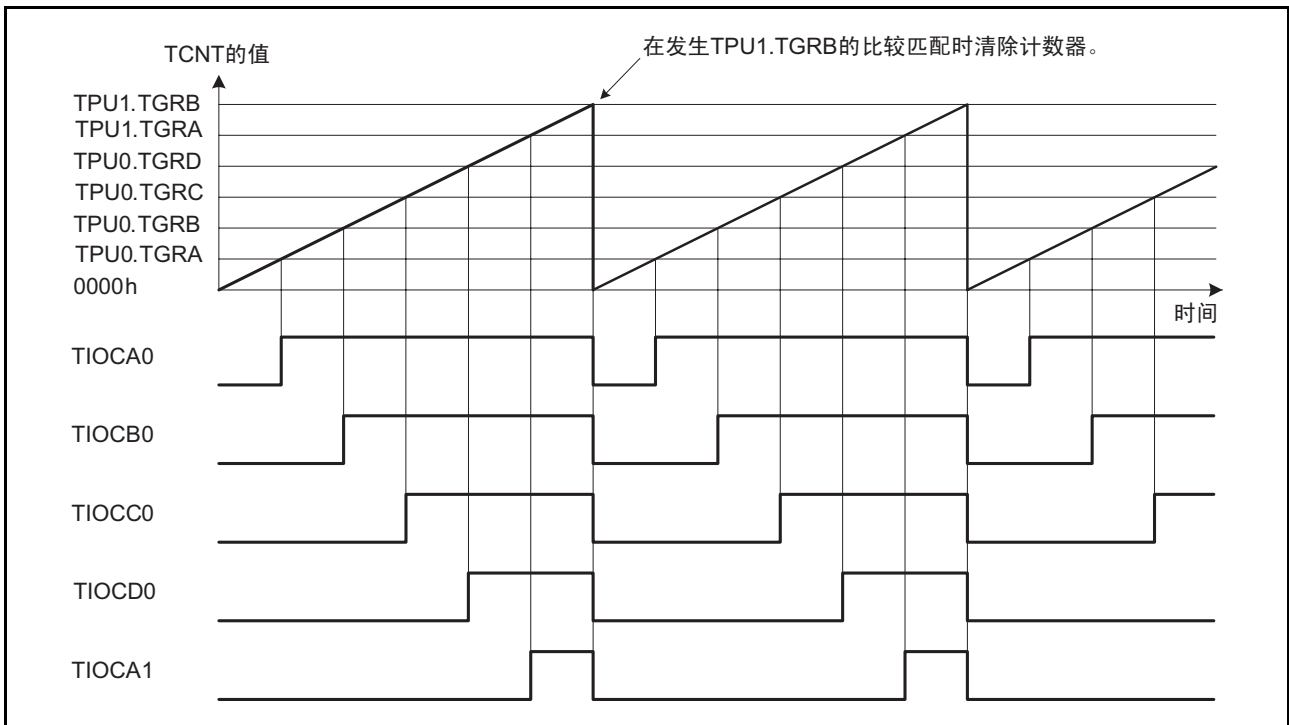


图 15.23 PWM 模式的运行例子 (2)

在 PWM 模式中，输出 0% 占空比和 100% 占空比的 PWM 波形的例子如图 15.24 所示。

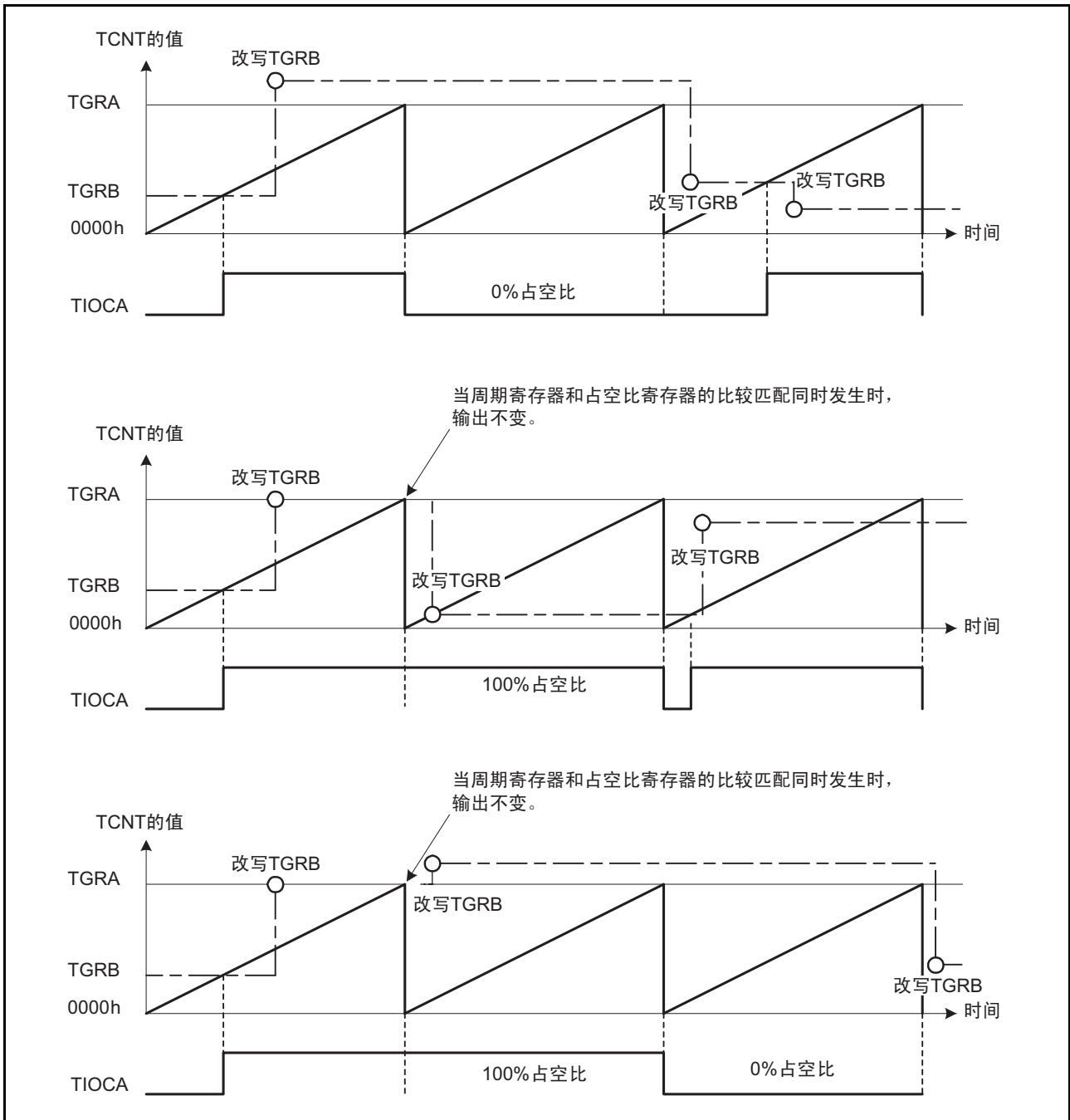


图 15.24 PWM 模式的运行例子 (3)

15.3.6 相位计数模式

在相位计数模式中，通过设定单元 0 中的通道 1,2,4,5（单元 1 中的通道 7,8,10,11），检测 2 个外部时钟输入的相位差，并且 TPUm.TCNT 计数器进行递增 / 递减计数。

如果设定为相位计数模式，就将外部时钟选择为计数器输入时钟，并且 TCNT 计数器作为递增 / 递减计数器运行，与 TPUm.TCR.TPSC[2:0] 位和 TPUm.TCR.CKEG[1:0] 位的设定无关。但是，因为 TPUm.TCR.CCLR[2:0] 位的低 2 位、TPUm.TIORH、TPUm.TIORL、TPUm.TIOR 寄存器、TPUm.TIER 寄存器、TPUm.TGRy 寄存器的功能有效，所以能使用输入捕捉 / 比较匹配功能和中断功能。

相位计数模式能用作 2 相编码器脉冲的输入。

如果 TCNT 计数器在进行递增计数时发生上溢，就产生 TCIV 中断请求；如果 TCNT 计数器在进行递减计数时发生下溢，就产生 TCIU 中断请求。TPUm.TSR.TCFD 标志是计数方向标志，能通过读 TCFD 标志，确认 TCNT 计数器是在进行递增计数还是在进行递减计数。

外部时钟引脚和通道的对应如表 15.26 所示。

表 15.26 相位计数模式的时钟输入引脚

| 单元 | 通道 | 外部时钟引脚 | |
|----|----------------------------|--------|-------|
| | | A 相 | B 相 |
| 0 | 将 TPU1 或者 TPU5 设定为相位计数模式时 | TCLKA | TCLKB |
| | 将 TPU2 或者 TPU4 设定为相位计数模式时 | TCLKC | TCLKD |
| 1 | 将 TPU7 或者 TPU11 设定为相位计数模式时 | TCLKE | TCLKF |
| | 将 TPU8 或者 TPU10 设定为相位计数模式时 | TCLKG | TCLKH |

(1) 相位计数模式的设定步骤例子

相位计数模式的设定步骤例子如图 15.25 所示。



图 15.25 相位计数模式的设定步骤例子

(2) 相位计数模式的运行例子

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

(a) 相位计数模式 1

相位计数模式 1 的运行例子如图 15.26 所示，TCNTn 的递增计数 / 递减计数条件如表 15.27 所示。

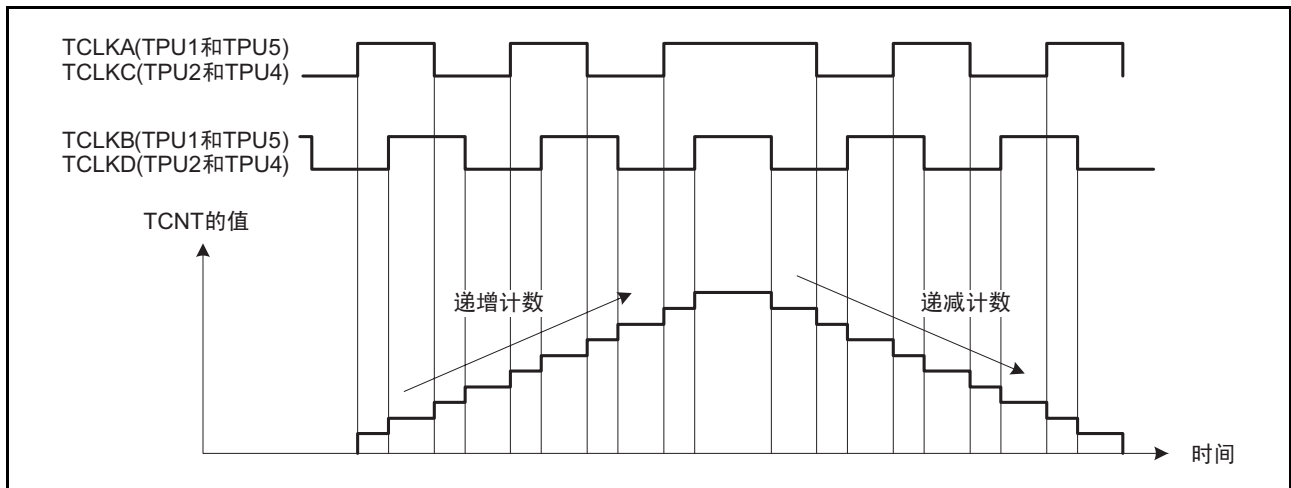


图 15.26 相位计数模式 1 的运行例子

表 15.27 相位计数模式 1 的递增计数 / 递减计数的条件

| TCLKA (TPU1 和 TPU5) TCLKC (TPU2 和 TPU4) | TCLKB (TPU1 和 TPU5) TCLKD (TPU2 和 TPU4) | 运行内容 |
|--|--|------|
| High 电平 | | 递增计数 |
| Low 电平 | | |
| | Low 电平 | |
| | High 电平 | |
| High 电平 | | 递减计数 |
| Low 电平 | | |
| | High 电平 | |
| | Low 电平 | |

【符号说明】

: 上升沿

: 下降沿

(b) 相位计数模式 2

相位计数模式 2 的运行例子如图 15.27 所示，TPUm.TCNT 的递增计数 / 递减计数的条件如表 15.28 所示。

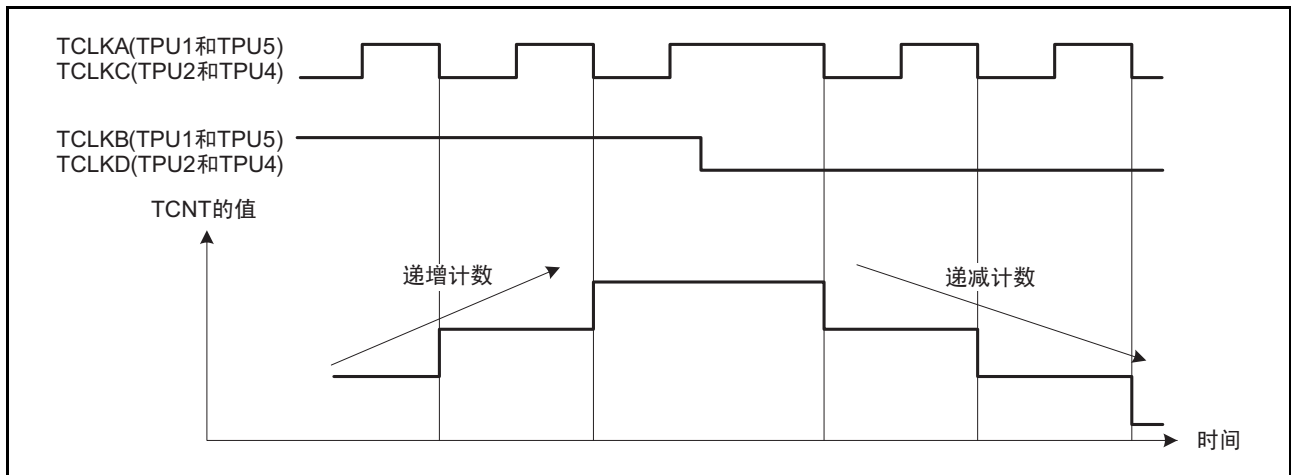


图 15.27 相位计数模式 2 的运行例子

表 15.28 相位计数模式 2 的递增计数 / 递减计数的条件

| TCLKA (TPU1 和 TPU5) TCLKC (TPU2 和 TPU4) | TCLKB (TPU1 和 TPU5) TCLKD (TPU2 和 TPU4) | 运行内容 |
|--|--|------------|
| High 电平 | | Don't care |
| Low 电平 | | Don't care |
| | Low 电平 | Don't care |
| | High 电平 | 递增计数 |
| High 电平 | | Don't care |
| Low 电平 | | Don't care |
| | High 电平 | Don't care |
| | Low 电平 | 递减计数 |

【符号说明】

: 上升沿

: 下降沿

(c) 相位计数模式 3

相位计数模式 3 的运行例子如图 15.28 所示，TPUm.TCNT 的递增计数 / 递减计数的条件如表 15.29 所示。

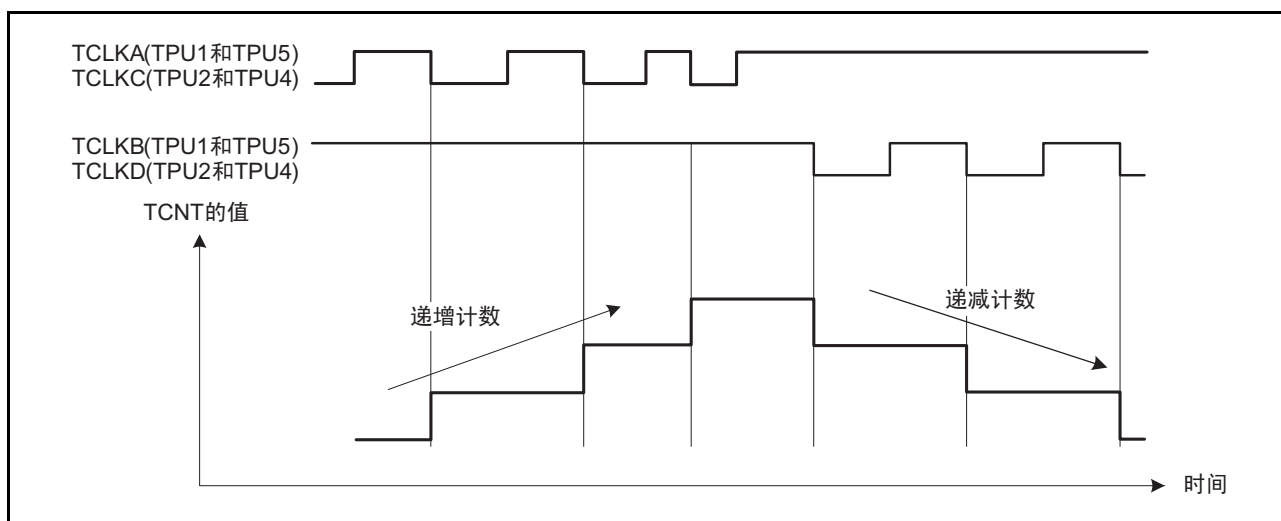


图 15.28 相位计数模式 3 的运行例子

表 15.29 相位计数模式 3 的递增计数 / 递减计数的条件

| TCLKA (TPU1 和 TPU5) TCLKC (TPU2 和 TPU4) | TCLKB (TPU1 和 TPU5) TCLKD (TPU2 和 TPU4) | 运行内容 |
|--|--|------------|
| High 电平 | ↑ | Don't care |
| Low 电平 | ↓ | Don't care |
| ↑ | Low 电平 | Don't care |
| ↓ | High 电平 | 递增计数 |
| High 电平 | ↓ | 递减计数 |
| Low 电平 | ↑ | Don't care |
| ↑ | High 电平 | Don't care |
| ↓ | Low 电平 | Don't care |

【符号说明】

↑ : 上升沿

↓ : 下降沿

(d) 相位计数模式 4

相位计数模式 4 的运行例子如图 15.29 所示，TPUm.TCNT 的递增计数 / 递减计数的条件如表 15.30 所示。

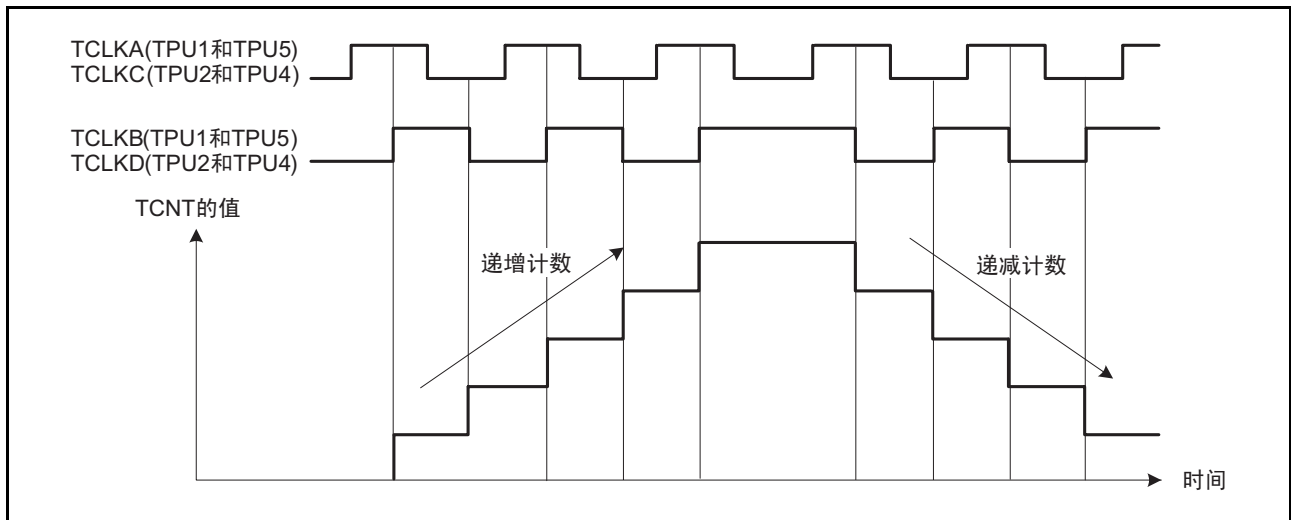


图 15.29 相位计数模式 4 的运行例子

表 15.30 相位计数模式 4 的递增计数 / 递减计数的条件

| TCLKA (TPU1 和 TPU5) TCLKC (TPU2 和 TPU4) | TCLKB (TPU1 和 TPU5) TCLKD (TPU2 和 TPU4) | 运行内容 |
|--|--|-------------|
| High 电平 | ↑ | 递增计数 |
| Low 电平 | ↓ | |
| ↑ | Low 电平 | Don' t care |
| ↓ | High 电平 | |
| High 电平 | ↓ | 递减计数 |
| Low 电平 | ↑ | |
| ↑ | High 电平 | Don' t care |
| ↓ | Low 电平 | |

【符号说明】

↑ : 上升沿

↓ : 下降沿

15.3.6.1 相位计数模式的应用例子

将 TPU1 设定为相位计数模式，并且与 TPU0 配合输入伺服马达 2 相编码器脉冲后检测位置或者速度的例子如图 15.30 所示。

将 TPU1 设定为相位计数模式 1，给 TCLKA 引脚和 TCLKB 引脚输入编码器脉冲的 A 相和 B 相。

TPU0 通过 TPU0.TGRC 寄存器的比较匹配清除 TPU0.TCNT 计数器，TPU0.TGRA 寄存器和 TPU0.TGRC 寄存器用于比较匹配功能，并且设定速度控制周期和位置控制周期。TPU0.TGRB 寄存器用于输入捕捉功能，使 TPU0.TGRB 寄存器和 TPU0.TGRD 寄存器进行缓冲运行。将 TPU0.TGRB 寄存器的输入捕捉源作为 TPU1 的计数器输入时钟，检测 2 相编码器的 4 倍频脉冲的脉宽。

将 TPU1 的 TPU1.TGRA 寄存器和 TPU1.TGRB 寄存器设定为输入捕捉功能，选择 TPU0 的 TPU0.TGRA 寄存器和 TPU0.TGRC 寄存器的比较匹配为输入捕捉源，保存各控制周期时的递增 / 递减计数器的值。

能用此方法检测正确的位置和速度。

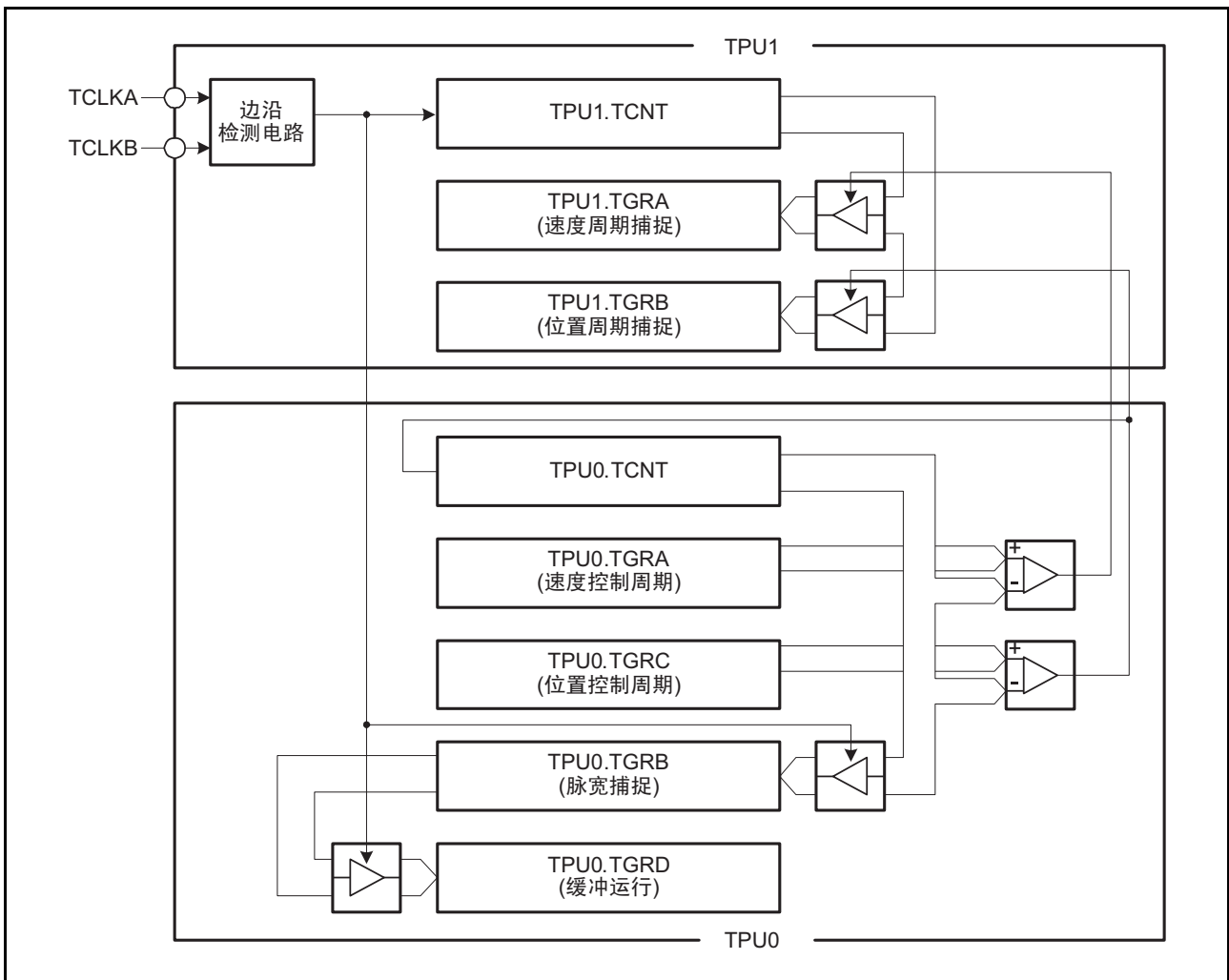


图 15.30 相位计数模式的应用例子

15.4 中断源

TPU 的中断源有 TPUm.TGRy 寄存器的输入捕捉 / 比较匹配、TPUm.TCNT 计数器的上溢和 TPUm.TCNT 计数器的下溢共 3 种。

能通过中断控制器更改通道之间的优先级，但是不能更改通道内的优先级。详细内容请参照“10. 中断控制器 (ICU)”。

TPU 的中断源一览表如表 15.31 所示。

表 15.31 TPU 中断一览表 (1/2)

| 单元 | 通道 | 名称 | 中断源 | DTC 的启动 | DMAC 的启动 |
|----|------|-------|------------------------|---------|----------|
| 0 | TPU0 | TGI0A | TPU0.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI0B | TPU0.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TGI0C | TPU0.TGRC 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TGI0D | TPU0.TGRD 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI0V | TPU0.TCNT 的上溢 | 不能 | 不能 |
| | TPU1 | TGI1A | TPU1.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI1B | TPU1.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI1V | TPU1.TCNT 的上溢 | 不能 | 不能 |
| | | TCI1U | TPU1.TCNT 的下溢 | 不能 | 不能 |
| | TPU2 | TGI2A | TPU2.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI2B | TPU2.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI2V | TPU2.TCNT 的上溢 | 不能 | 不能 |
| | | TCI2U | TPU2.TCNT 的下溢 | 不能 | 不能 |
| | TPU3 | TGI3A | TPU3.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI3B | TPU3.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TGI3C | TPU3.TGRC 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TGI3D | TPU3.TGRD 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI3V | TPU3.TCNT 的上溢 | 不能 | 不能 |
| | TPU4 | TGI4A | TPU4.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI4B | TPU4.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI4V | TPU4.TCNT 的上溢 | 不能 | 不能 |
| | | TCI4U | TPU4.TCNT 的下溢 | 不能 | 不能 |
| | TPU5 | TGI5A | TPU5.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI5B | TPU5.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI5V | TPU5.TCNT 的上溢 | 不能 | 不能 |
| | | TCI5U | TPU5.TCNT 的下溢 | 不能 | 不能 |

表 15.31 TPU 中断一览表 (2/2)

| 单元 | 通道 | 名称 | 中断源 | DTC 的启动 | DMAC 的启动 |
|----|-------|--------|-------------------------|------------|-------------|
| 1 | TPU6 | TGI6A | TPU6.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI6B | TPU6.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TGI6C | TPU6.TGRC 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TGI6D | TPU6.TGRD 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI6V | TPU6.TCNT 的上溢 | 不能 | 不能 |
| | TPU7 | TGI7A | TPU7.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI7B | TPU7.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI7V | TPU7.TCNT 的上溢 | 不能 | 不能 |
| | | TCI7U | TPU7.TCNT 的下溢 | 不能 | 不能 |
| | TPU8 | TGI8A | TPU8.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI8B | TPU8.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI8V | TPU8.TCNT 的上溢 | 不能 | 不能 |
| | | TCI8U | TPU8.TCNT 的下溢 | 不能 | 不能 |
| | TPU9 | TGI9A | TPU9.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI9B | TPU9.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TGI9C | TPU9.TGRC 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TGI9D | TPU9.TGRD 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI9V | TPU9.TCNT 的上溢 | 不能 | 不能 |
| | TPU10 | TGI10A | TPU10.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI10B | TPU10.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI10V | TPU10.TCNT 的上溢 | 不能 | 不能 |
| | | TCI10U | TPU10.TCNT 的下溢 | 不能 | 不能 |
| | TPU11 | TGI11A | TPU11.TGRA 的输入捕捉 / 比较匹配 | 能 | 能 |
| | | TGI11B | TPU11.TGRB 的输入捕捉 / 比较匹配 | 能 | 不能 |
| | | TCI11V | TPU11.TCNT 的上溢 | 不能 | 不能 |
| | | TCI11U | TPU11.TCNT 的下溢 | 不能 | 不能 |

注 1. 表示复位后的初始状态。能通过中断控制器更改通道之间的优先级。

(1) 输入捕捉 / 比较匹配中断

如果在各通道的 TPU_m.TGR_y 寄存器发生输入捕捉 / 比较匹配时 TPU_m.TIER.TGIE_y 位 (y=A,B,C,D) 为“1”，就请求中断。在 TPU 中，TPU0 和 TPU3 (TPU6 和 TPU9) 各有 4 个输入捕捉 / 比较匹配中断，TPU1,2,4,5 (TPU7,8,10,11) 各有 2 个输入捕捉 / 比较匹配中断，共计 32 个。

(2) 上溢中断

如果在各通道的 TPU_m.TCNT 计数器发生上溢时 TPU_m.TIER.TCIEV 位为“1”，就请求中断。在 TPU 中，各通道各有 1 个上溢中断，共计 12 个。

(3) 下溢中断

如果在各通道的 TPU_m.TCNT 计数器发生下溢时 TPU_m.TIER.TCIEU 位为“1”，就请求中断。在 TPU 中，TPU1,2,4,5 (TPU7,8,10,11) 各有 1 个下溢中断，共计 8 个。

15.5 DTC 的启动

能通过各通道的 TPUm.TGRy 寄存器的输入捕捉 / 比较匹配中断来启动 DTC。详细内容请参照“13. 数据传送控制器 (DTC)”。

在 TPU 中, 能将 TPU0 和 TPU3 (TPU6 和 TPU9) 的各 4 个以及 TPU1,2,4,5 (TPU7,8,10,11) 的各 2 个, 共计 32 个输入捕捉 / 比较匹配中断作为 DTC 的启动源。

15.6 DMAC 的启动

能通过各通道的 TPUm.TGRA 寄存器的输入捕捉 / 比较匹配中断来启动 DMAC。详细内容请参照“12. DMA 控制器 (DMAC)”。

在 TPU 中, 能将各通道的各 1 个共计 12 个 TPUm.TGRA 寄存器的输入捕捉 / 比较匹配中断作为 DMAC 的启动源。

15.7 A/D 转换器的启动

TPU 能通过各通道的 TPUm.TGRA 寄存器的输入捕捉 / 比较匹配来启动 A/D 转换器, 并且通过 TPU0 的 TGRA ~ TGRD 寄存器的输入捕捉 / 比较匹配来启动 A/D 转换器。

在发生各通道的 TPUm.TGRA 寄存器的输入捕捉 / 比较匹配时, 如果 TPUm.TIER.TTGE 位为“1”, TPU 就向 A/D 转换器请求开始 A/D 转换。当发生 TPU0 的 TGRA ~ TGRD 寄存器的输入捕捉 / 比较匹配时, TPU 就向对应的 A/D 转换器请求开始 A/D 转换。此时, 如果已在 A/D 转换器侧选择了 TPU 的转换开始触发, 就开始 A/D 转换。

有关对应的 A/D 转换器单元请参照“23. A/D 转换器”。

15.8 运行时序

15.8.1 输入 / 输出时序

(1) TPUm.TCNT 计数器的计数时序

内部时钟运行和外部时钟运行时的 TPUm.TCNT 计数器的计数时序分别如图 15.31 和图 15.32 所示。

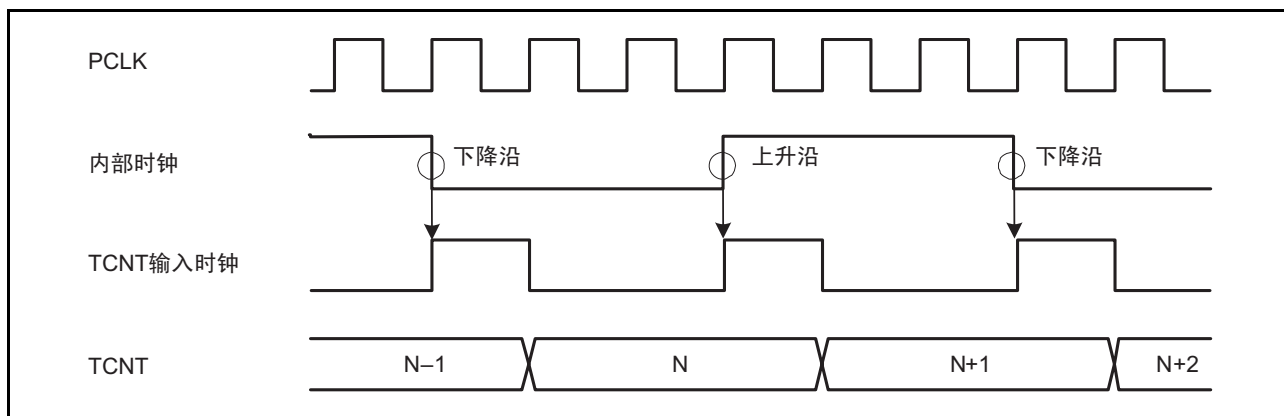


图 15.31 内部时钟运行时的计数时序

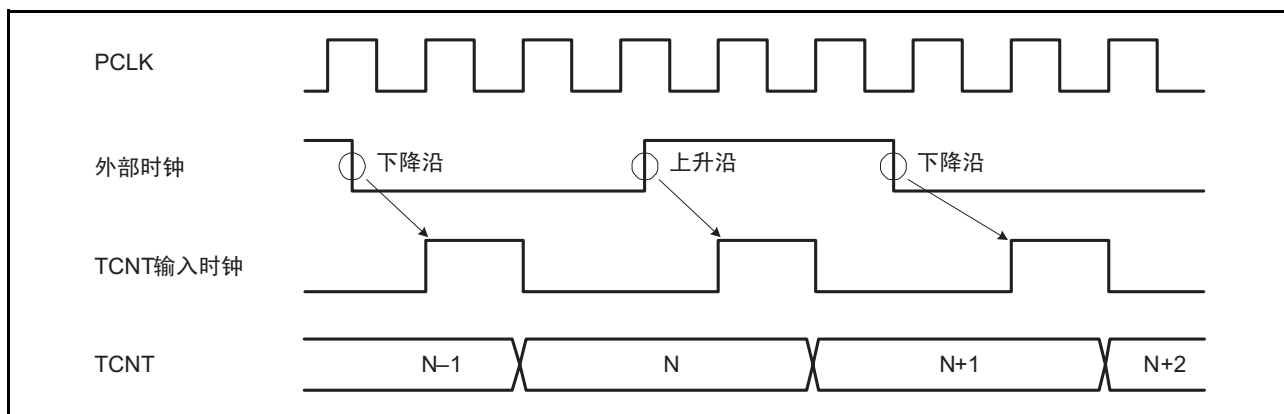


图 15.32 外部时钟运行时的计数时序

(2) 输出比较的输出时序

在 TPUm.TCNT 计数器和 TPUm.TGRy 寄存器的值相同的最后状态（在更新 TCNT 计数器相同后的计数值时）产生比较匹配信号。在产生比较匹配信号后，将 TPUm.TIORH、TPUm.TIORL、TPUm.TIOR 寄存器设定的输出值从输出比较的输出引脚（TIOCyn（y=A ~ D，n=0 ~ 11）引脚）输出。在从 TCNT 计数器和 TGRy 寄存器的值相同后到产生 TCNT 计数器的输入时钟前，不产生比较匹配信号。

输出比较的输出时序如图 15.33 所示。

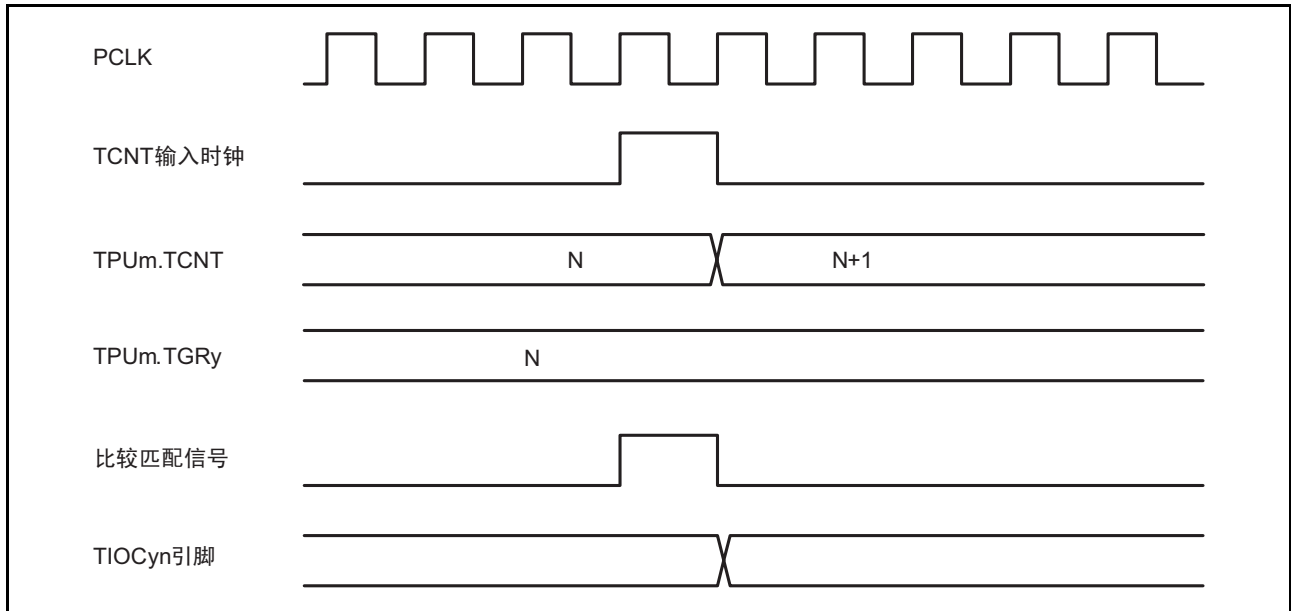


图 15.33 输出比较的输出时序

(3) 输入捕捉的输入时序

输入捕捉的输入时序如图 15.34 所示。

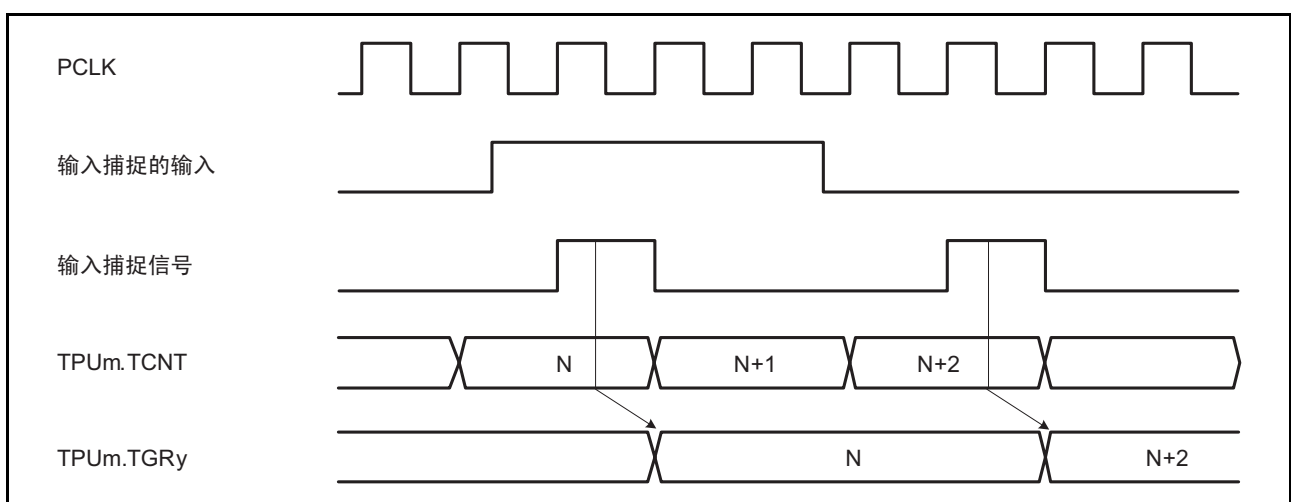


图 15.34 输入捕捉的输入时序

(4) 通过比较匹配 / 输入捕捉进行的计数器清除时序

指定通过比较匹配进行计数器清除时的时序如图 15.35 所示。

指定通过输入捕捉进行计数器清除时的时序如图 15.36 所示。

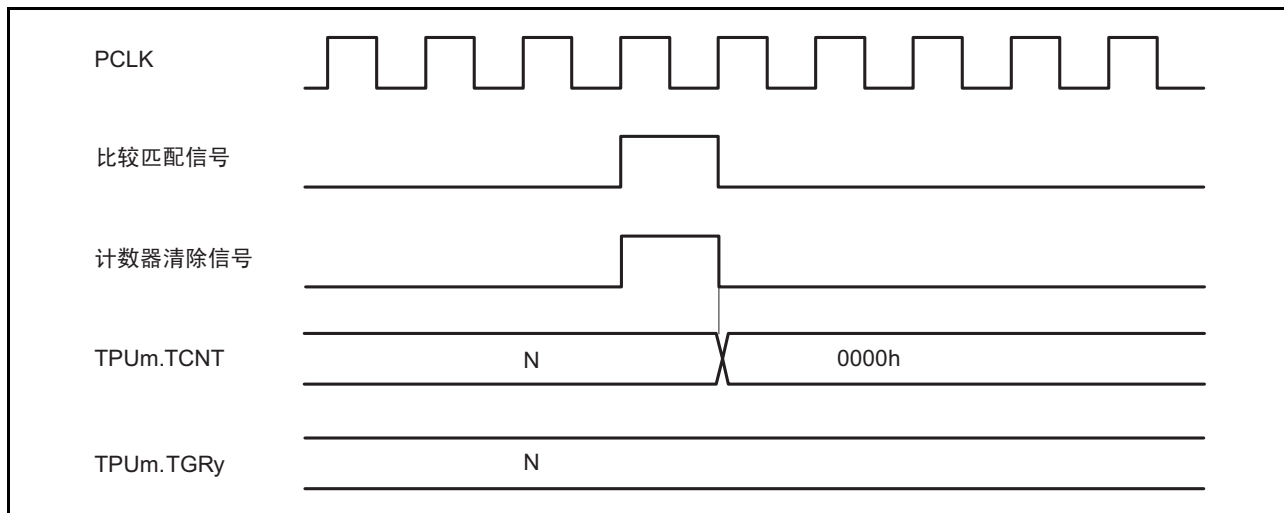


图 15.35 计数器清除的时序 (比较匹配)

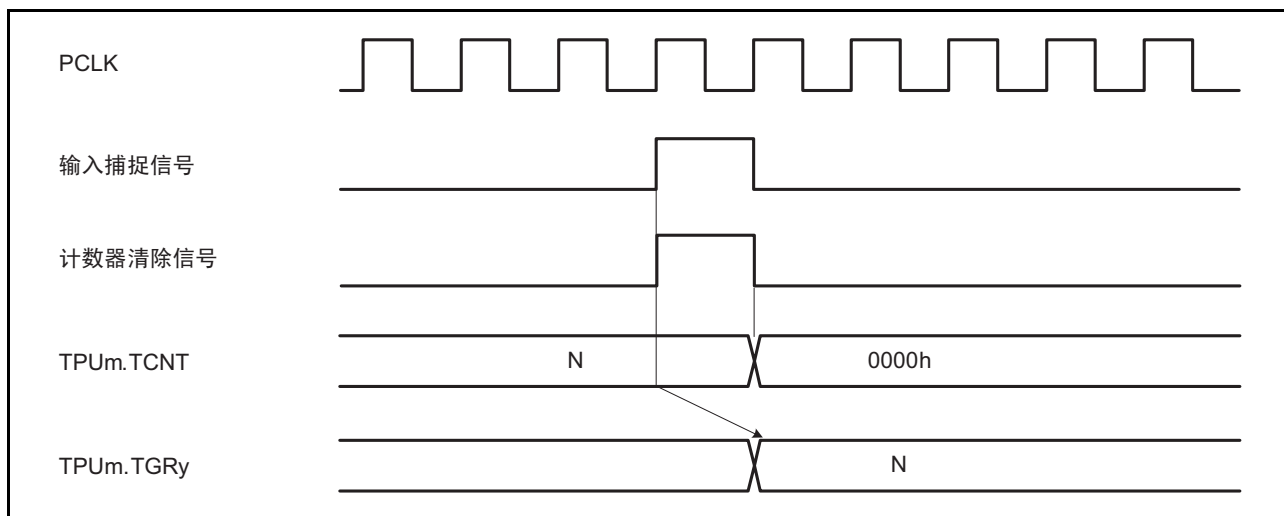


图 15.36 计数器清除的时序 (输入捕捉)

(5) 缓冲运行的时序

缓冲运行时的时序如图 15.37 和图 15.38 所示。

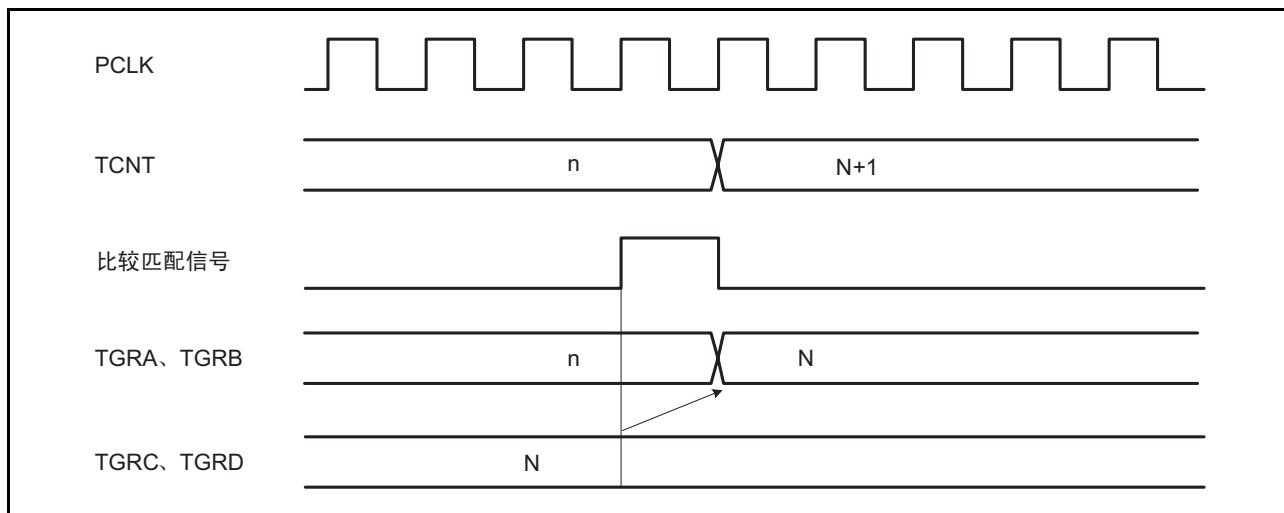


图 15.37 缓冲运行的时序 (比较匹配)

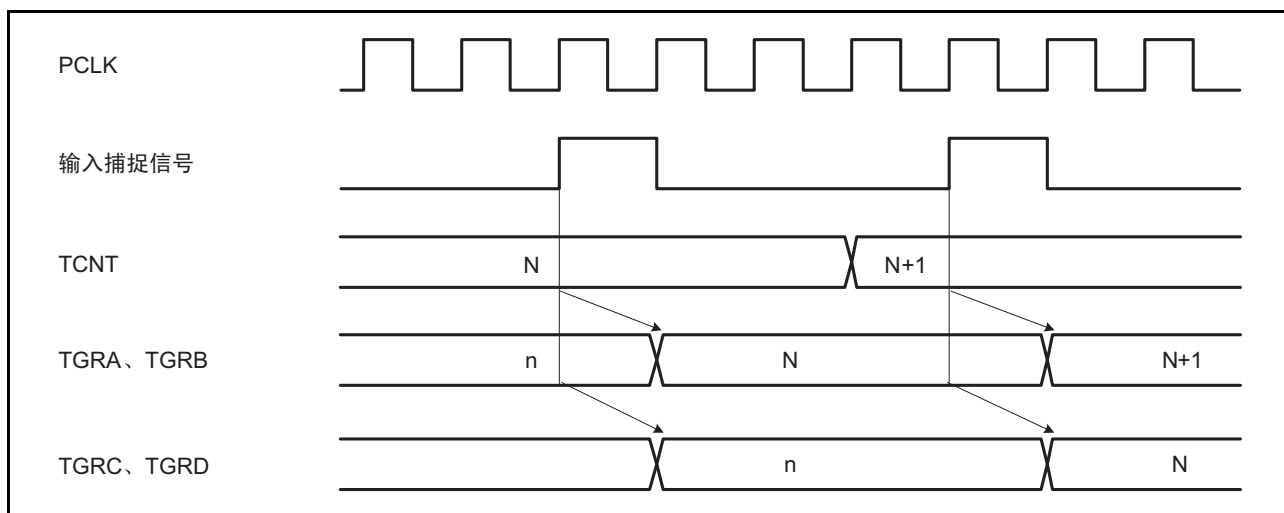


图 15.38 缓冲运行的时序 (输入捕捉)

15.8.2 中断信号的时序

(1) 比较匹配时的中断标志变为“1”的时序

由比较匹配产生的中断标志的时序如图 15.39 所示。

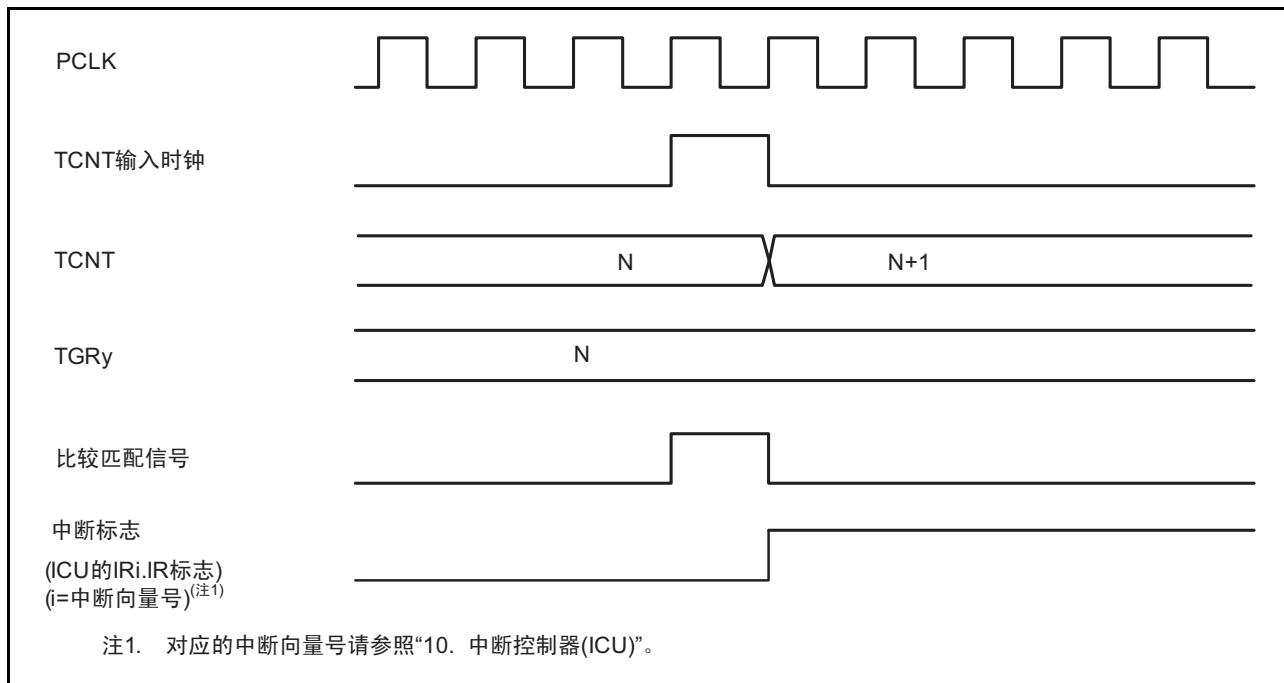


图 15.39 TGI_{my} 中断时序 (比较匹配)

(2) 输入捕捉时的中断标志变为“1”的时序

由输入捕捉产生的中断标志的时序如图 15.40 所示。

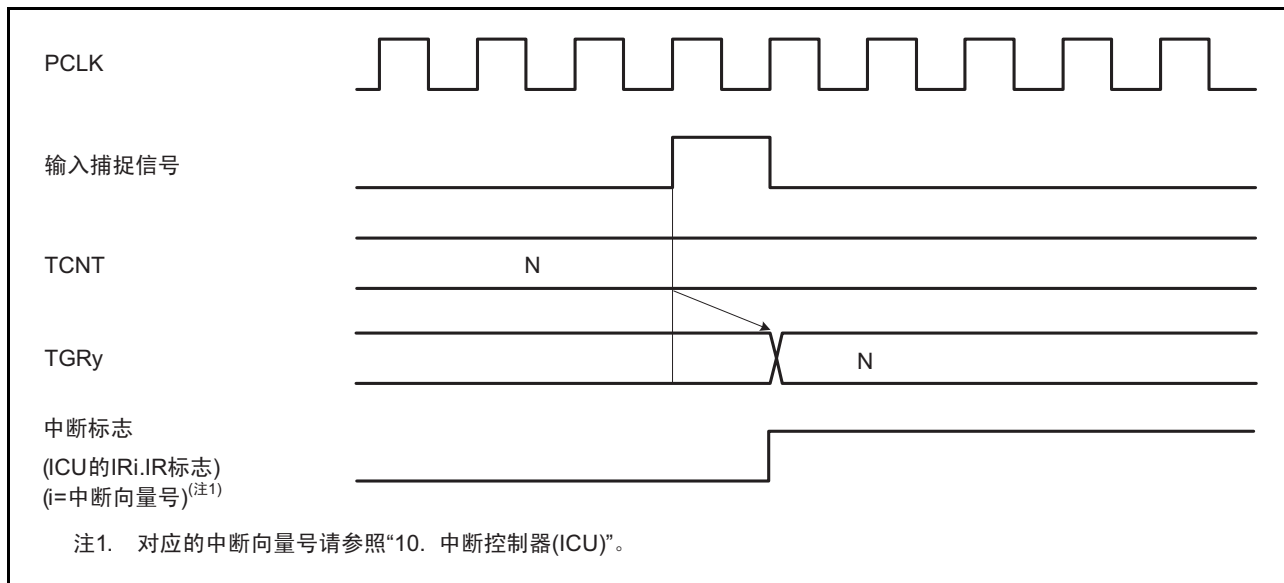


图 15.40 TGI_{my} 中断时序 (输入捕捉)

(3) TCImV 和 TCImU 中断标志变为“1”的时序

由上溢产生的 TCImV 中断请求信号的时序如图 15.41 所示。

由下溢产生的 TCImU 中断请求信号的时序如图 15.42 所示。

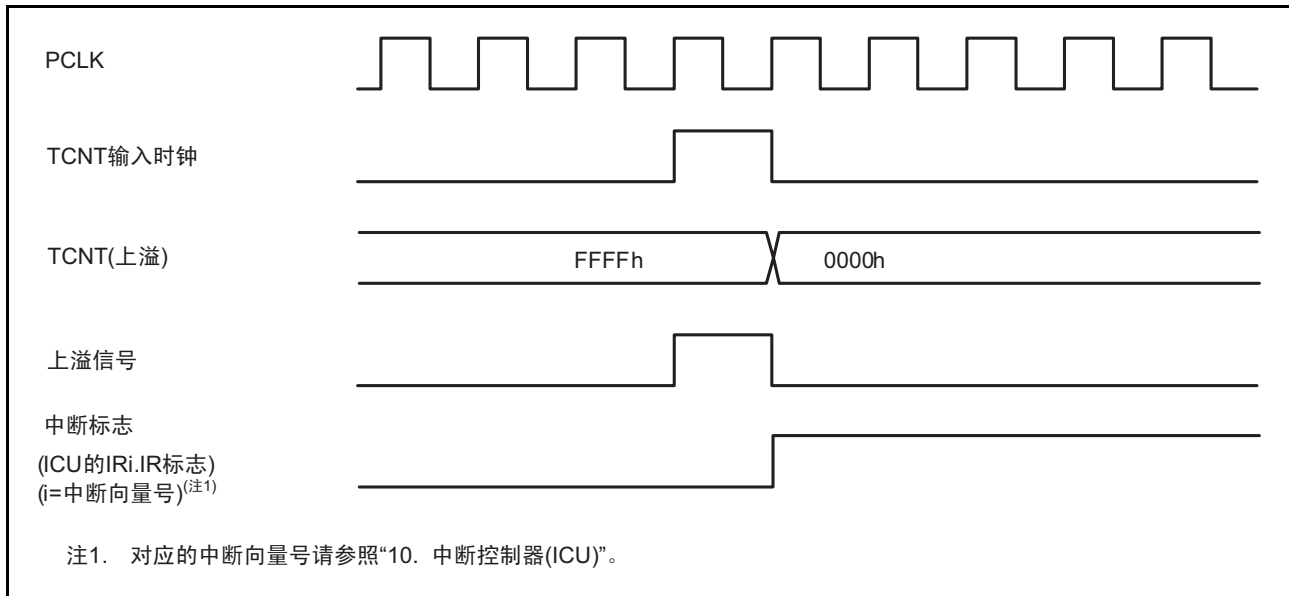


图 15.41 TCImV 中断的置位时序

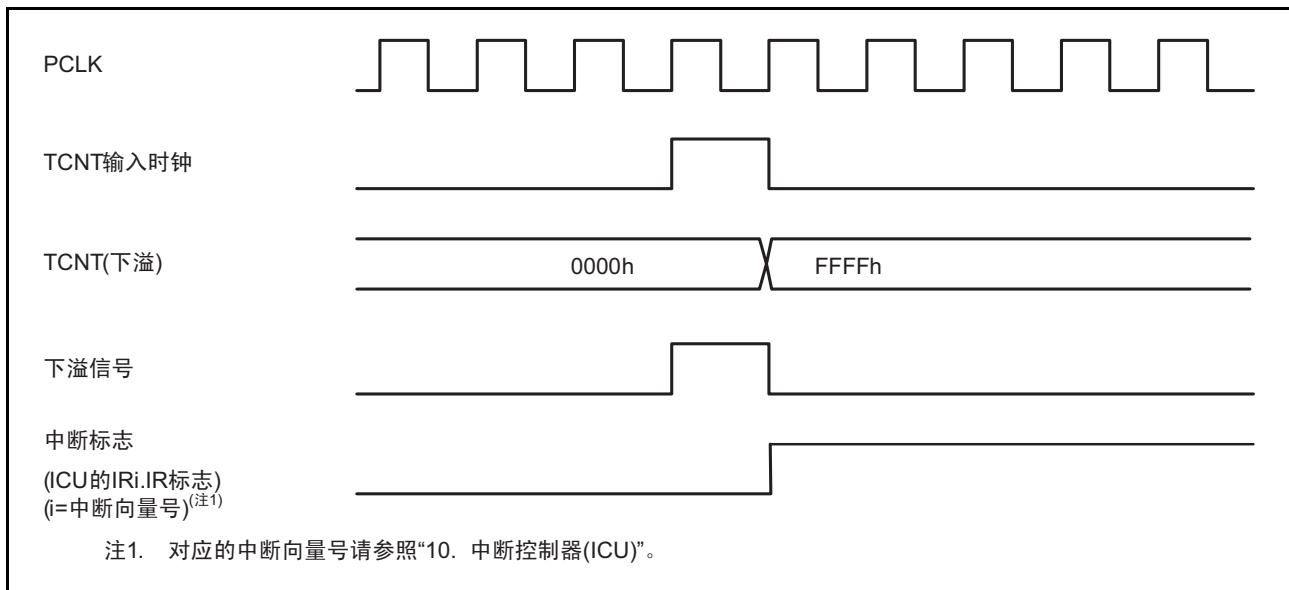


图 15.42 TCImU 中断的置位时序

15.9 使用时的注意事项

15.9.1 模块停止功能的设定

能通过模块停止控制寄存器设定允许或者禁止 TPU 的运行，初始值为停止 TPU 的运行。能通过解除模块停止状态，使寄存器变为可存取的状态，详细内容请参照“8. 低功耗功能”。

15.9.2 输入时钟的限制事项

必须注意：在单边沿的情况下，输入时钟的脉宽至少为 1.5 个状态；在双边沿的情况下，输入时钟的脉宽至少为 2.5 个状态。否则，就不能正常运行。

在相位计数模式中，2 个输入时钟的相位差和重叠都至少为 1.5 个状态，脉宽至少为 2.5 个状态。相位计数模式的输入时钟条件如图 15.43 所示。

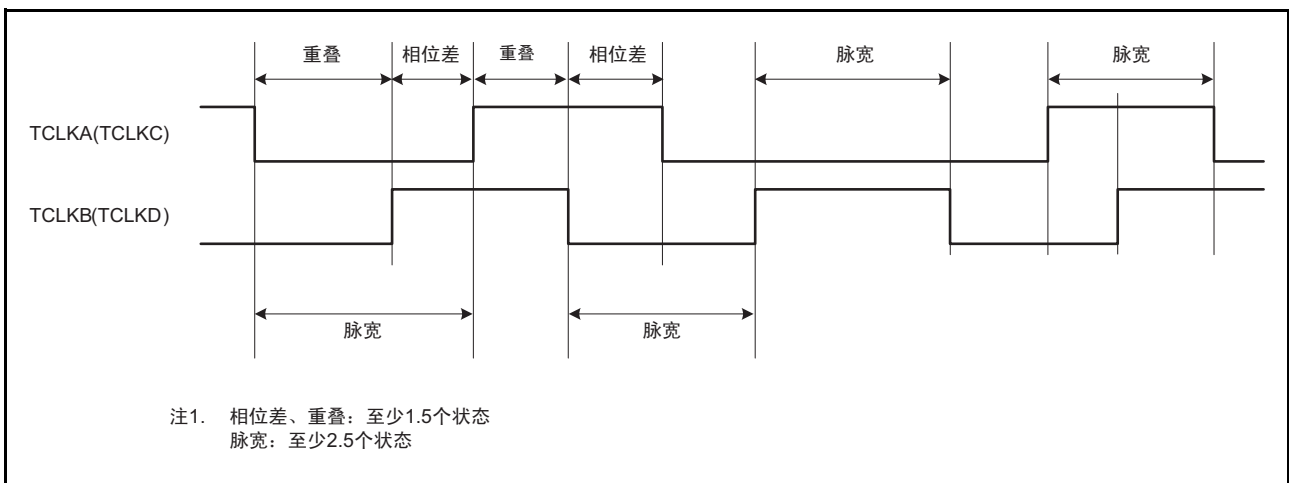


图 15.43 相位计数模式中的位相差、重叠和脉宽

15.9.3 设定周期时的注意事项

如果设定为通过比较匹配清除计数器，就在 TPUm.TCNT 计数器和 TPUm.TGRy 寄存器的值相同的最后状态（在更新 TCNT 计数器相同后的计数值时）清除 TPUm.TCNT 计数器。因此，实际的计数器的频率用以下的表达式表示：

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

- f: 计数器的频率
PCLK: 工作频率
N: TGRy 寄存器的设定值

15.9.4 TPUm.TCNT 计数器的写和清除的竞争

如果在 TCNT 计数器的写周期产生计数器清除信号，就不写 TCNT 计数器而优先清除 TCNT 计数器。此时序如图 15.44 所示。

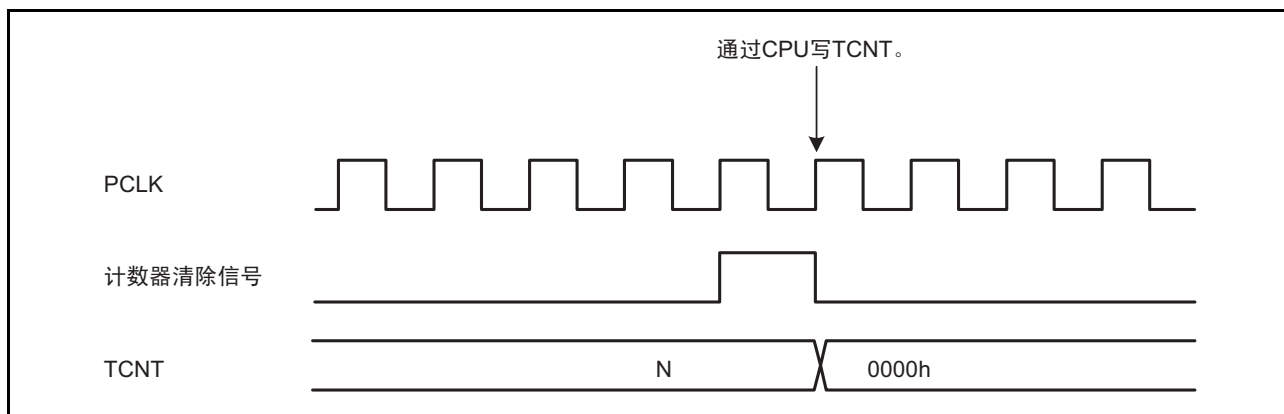


图 15.44 TPUm.TCNT 计数器的写和清除的竞争

15.9.5 TPUm.TCNT 计数器的写和递增计数的竞争

即使在 TCNT 计数器的写周期产生递增计数信号，也不进行递增计数而优先写 TCNT 计数器。此时序如图 15.45 所示。

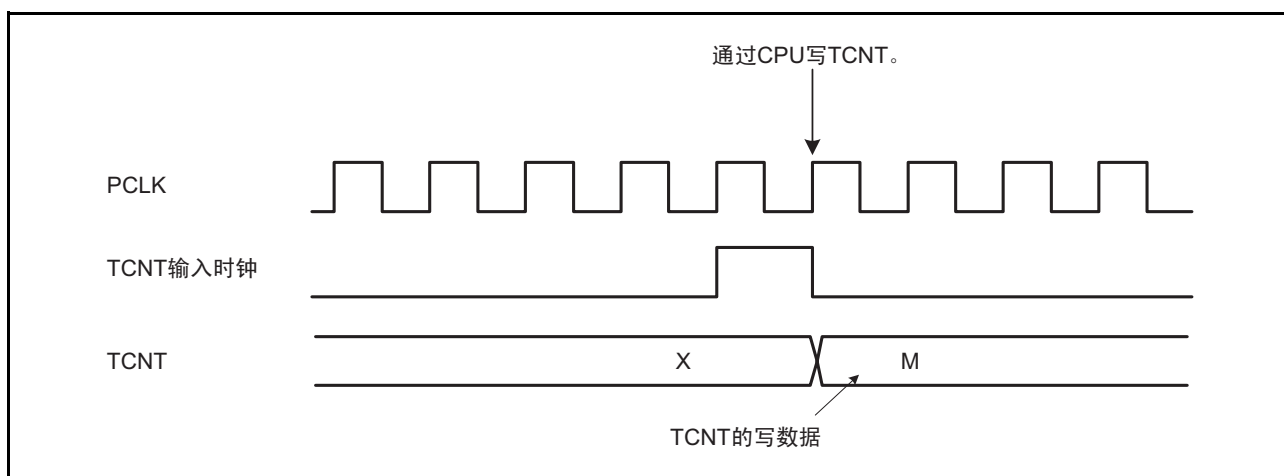


图 15.45 TPUm.TCNT 计数器的写和递增计数的竞争

15.9.6 TPUm.TGRy 寄存器的写和比较匹配的竞争

即使在 TGRy 寄存器的写周期产生比较匹配信号，也禁止比较匹配信号而优先写 TGRy 寄存器。即使写和上次相同的值，也不发生比较匹配。

此时序如图 15.46 所示。

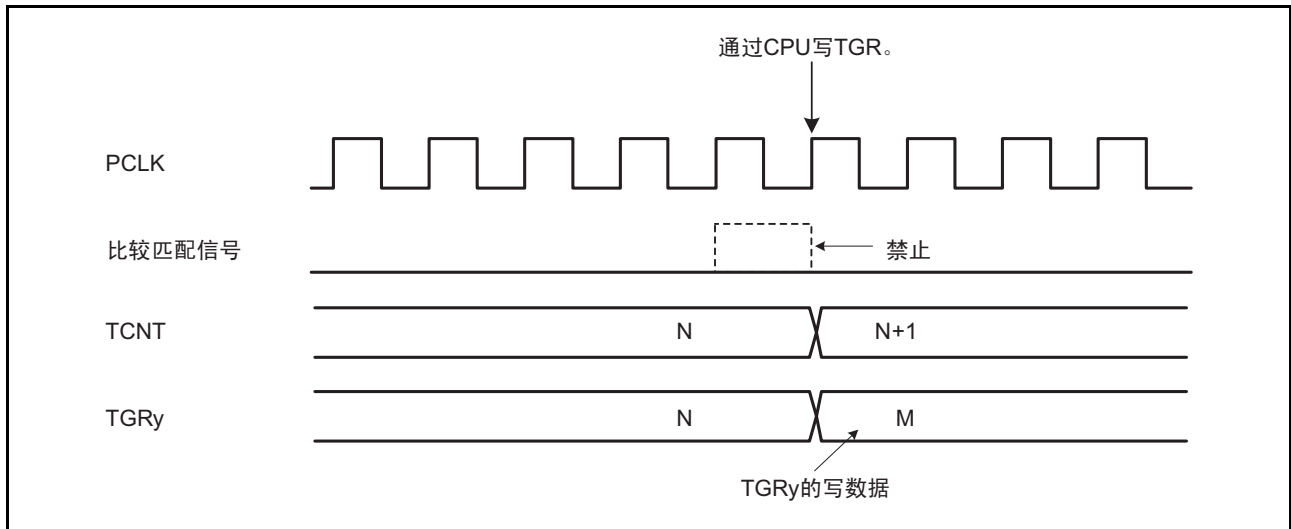


图 15.46 TPUm.TGRy 寄存器的写和比较匹配的竞争

15.9.7 缓冲寄存器的写和比较匹配的竞争

如果在 TPUm.TGRy 寄存器的写周期产生比较匹配信号，通过缓冲运行传送到 TGRy 寄存器的数据就为写之前的数据。

此时序如图 15.47 所示。

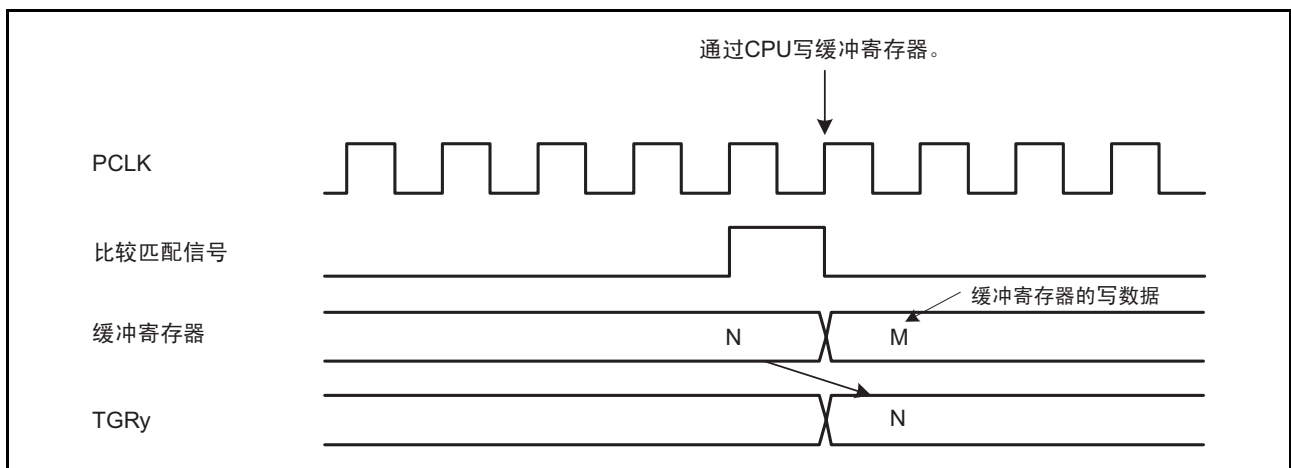


图 15.47 缓冲寄存器的写和比较匹配的竞争

15.9.8 TPUm.TGRy 寄存器的读和输入捕捉的竞争

如果在 TGRy 寄存器的读周期产生输入捕捉信号，读出的数据就为传送输入捕捉前的数据。此时序如图 15.48 所示。

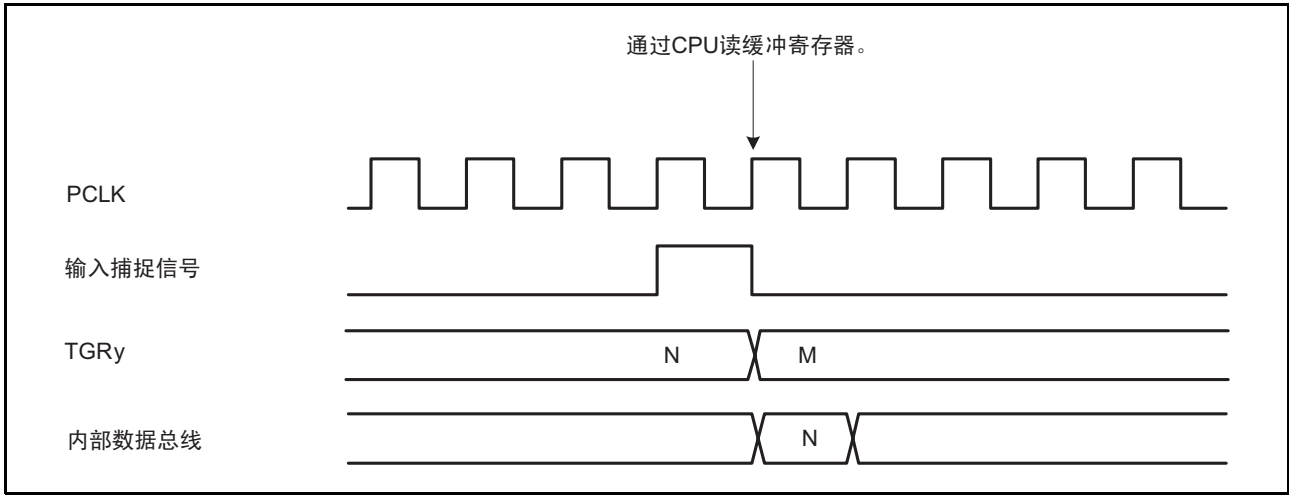


图 15.48 TPUm.TGRy 寄存器的读和输入捕捉的竞争

15.9.9 TPUm.TGRy 寄存器的写和输入捕捉的竞争

如果在 TGRy 寄存器的写周期产生输入捕捉信号，就不写 TGRy 寄存器而优先输入捕捉。此时序如图 15.49 所示。

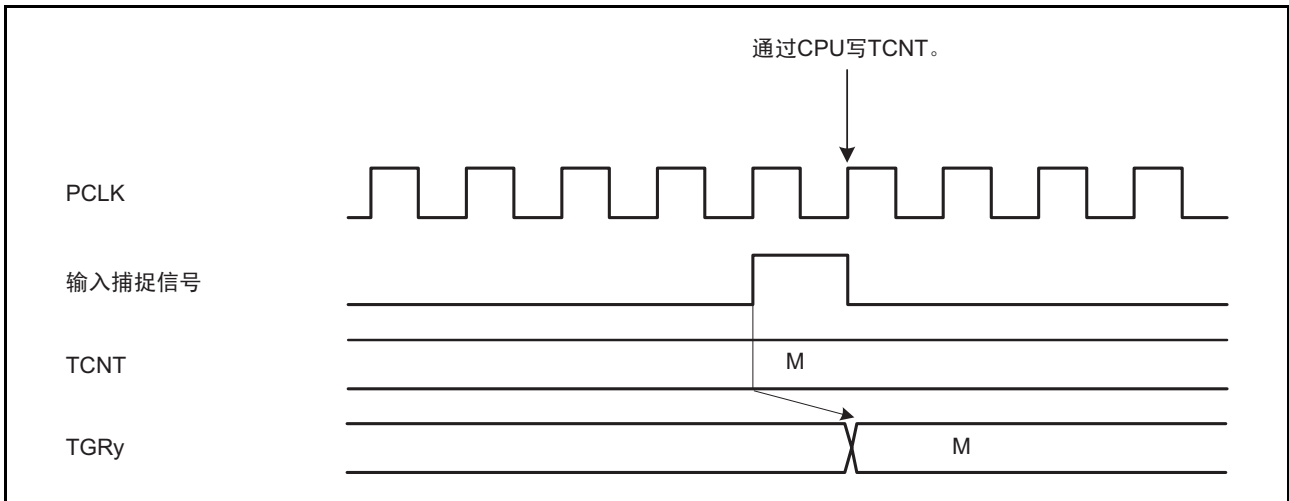


图 15.49 TPUm.TGRy 寄存器的写和输入捕捉的竞争

15.9.10 缓冲寄存器的写和输入捕捉的竞争

如果在缓冲寄存器的写周期产生输入捕捉信号，就不写缓冲寄存器而优先缓冲运行。此时序如图 15.50 所示。

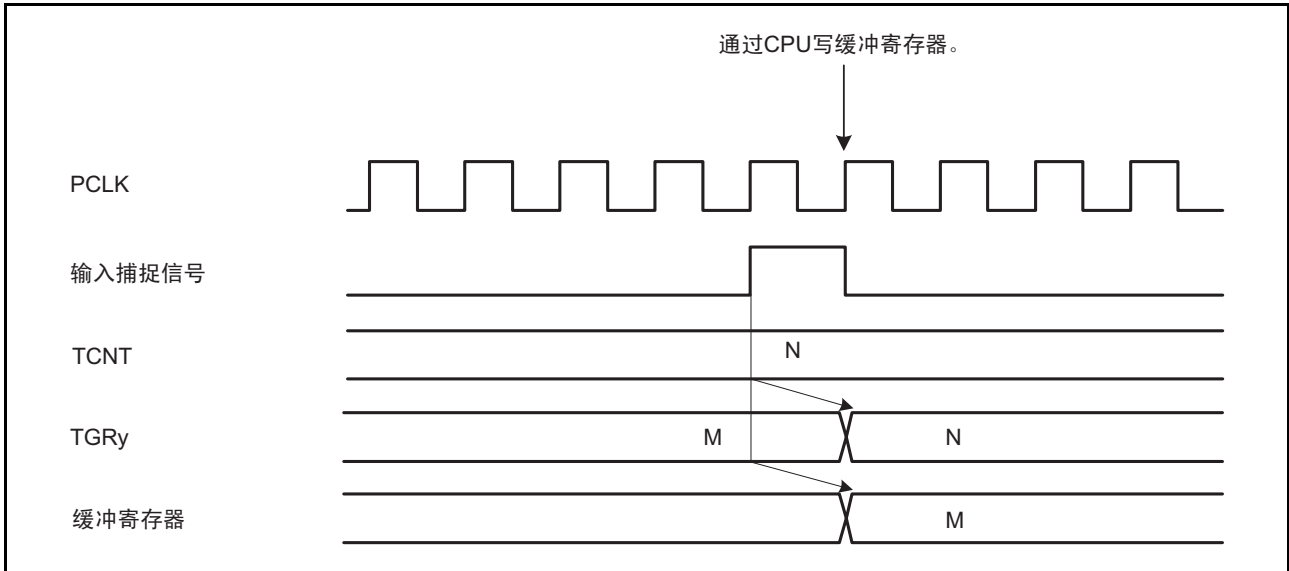


图 15.50 缓冲寄存器的写和输入捕捉的竞争

15.9.11 上溢 / 下溢和计数器清除的竞争

如果同时发生上溢 / 下溢和计数器清除，就在发生比较匹配中断的同时，清除 TPUm.TCNT 计数器，并且发生上溢中断。

将 TPUm.TGRy 寄存器的比较匹配作为清除源并且将 TGRy 寄存器置 “FFFFh” 时的运行时序如图 15.51 所示。

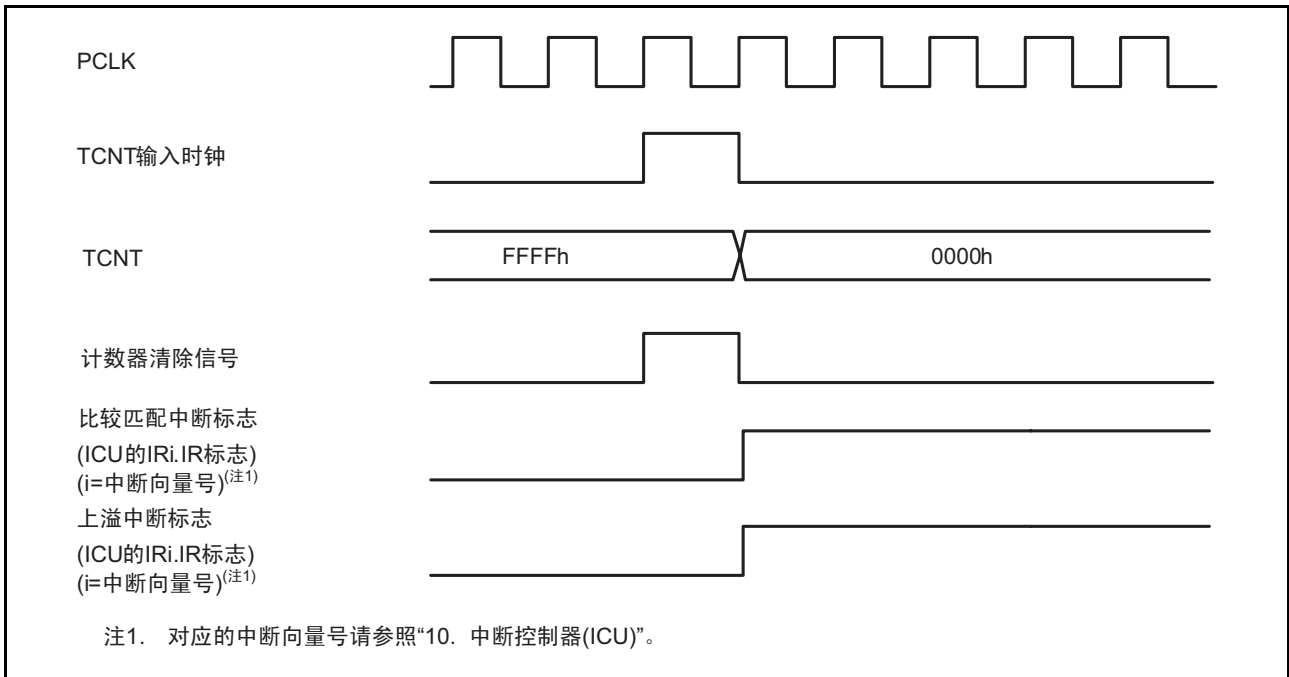


图 15.51 上溢和计数器清除的竞争

15.9.12 TPUm.TCNT 计数器的写和上溢 / 下溢的竞争

如果在 TCNT 计数器的写周期发生递增计数 / 递减计数，即使发生上溢 / 下溢，也优先写 TCNT 计数器。TCNT 计数器的写和上溢竞争时的运行时序如图 15.52 所示。

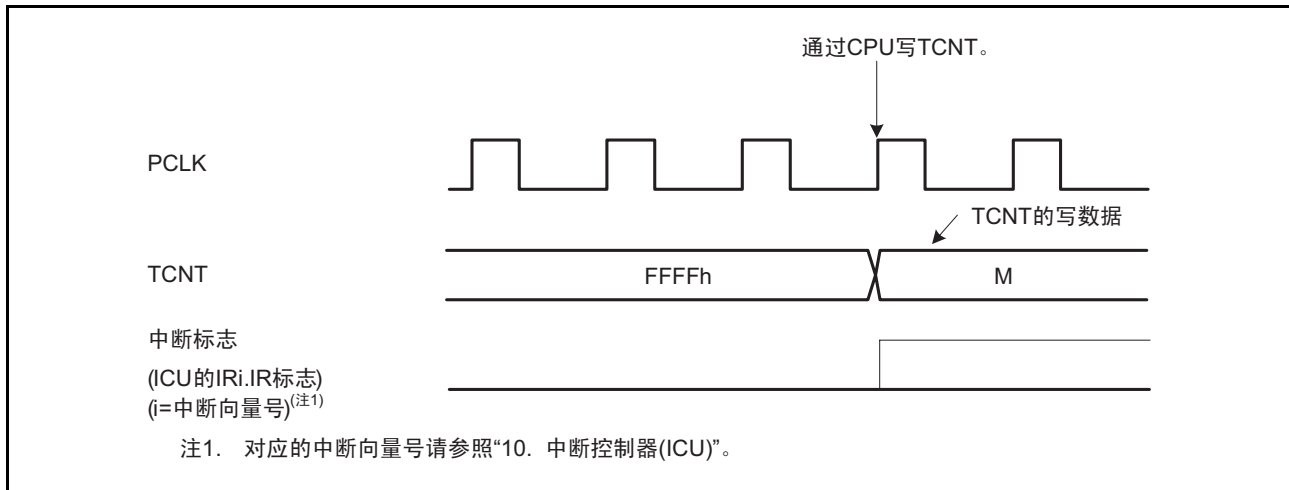


图 15.52 TPUm.TCNT 计数器的写和上溢的竞争

15.9.13 输入 / 输出引脚的兼用

在 RX610 群中，TCLKA-A 输入引脚和 TIOCC0 输入 / 输出引脚、TCLKB-A 输入引脚和 TIOCD0 输入 / 输出引脚、TCLKC-A 输入引脚和 TIOCB1 输入 / 输出引脚以及 TCLKD-A 输入引脚和 TIOCB2 输入 / 输出引脚兼用。要输入外部时钟时，不能从兼用的引脚输出比较匹配。

16. 可编程脉冲发生器 (PPG)

可编程脉冲发生器 (PPG) 以 16 位定时器脉冲单元 (TPU) 为时基输出脉冲。

RX610 群装载了 2 个单元的 PPG，每个单元都有 16 位的脉冲输出引脚，以 4 位为单位构成 1 个脉冲输出组。各脉冲输出组既能同时运行，也能各自独立运行。

16.1 概要

PPG 的规格和功能一览表分别如表 16.1 和表 16.2 所示。

PPG 的框图如图 16.1 和图 16.2 所示。

表 16.1 PPG 的规格

| 项目 | 内容 |
|---------|--|
| 输出位数 | 最多 32 位 |
| 脉冲输出 | <ul style="list-style-type: none"> 能输出 4 组 ×2 个单元。 能选择输出触发信号。 能进行非重叠运行。 能指定反相输出。 |
| 输出数据的传送 | 能和 DTC、DMAC 联动运行 (使用 TPU 的中断功能时)。 |
| 低功耗功能 | 各单元能设定为模块停止状态。 |

表 16.2 PPG 功能一览表

| 项目 | | PPG0 | PPG1 |
|---------------|--|-------------------|-------------------|
| PPG 输出触发 | TPU (单元 0) 的 通道 0 ~ 通道 3 (TPU0 ~ TPU3) | 比较匹配 | ○ |
| | | 输入捕捉 | ○ |
| | TPU (单元 1) 的 通道 6 ~ 通道 9 (TPU6 ~ TPU9) | 比较匹配 | — |
| | | 输入捕捉 | ○ |
| 非重叠运行 | | ○ | ○ |
| 输出数据的传送 | DTC | ○ | ○ |
| | DMAC | ○ | ○ |
| 反相输出的指定 | | ○ | ○ |
| 模块停止的设定 (注 1) | | MSTPCRA.MSTPA11 位 | MSTPCRA.MSTPA10 位 |

○: 能

—: 不能

注 1. 详细内容请参照“8. 低功耗功能”。

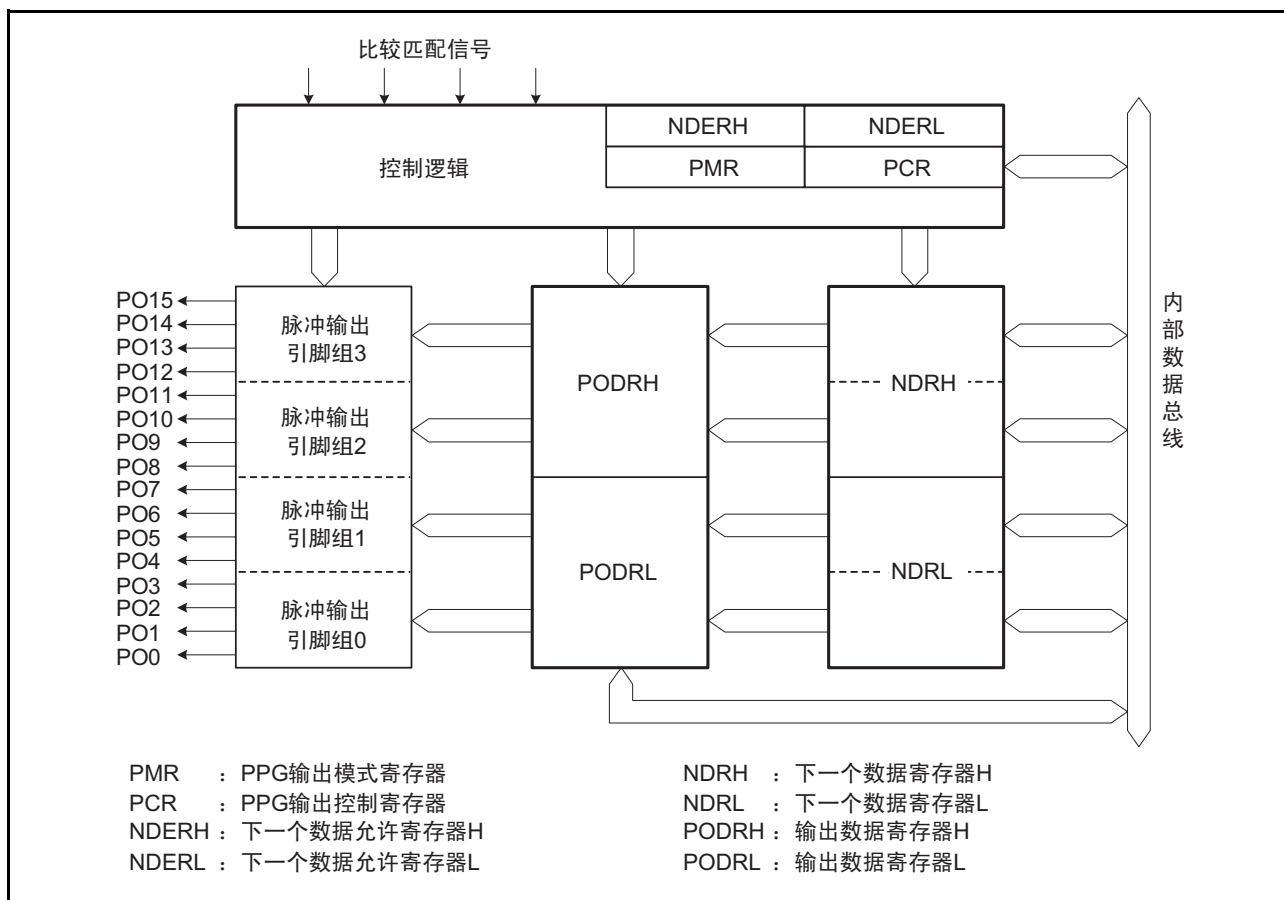


图 16.1 PPG (单元 0) 的框图

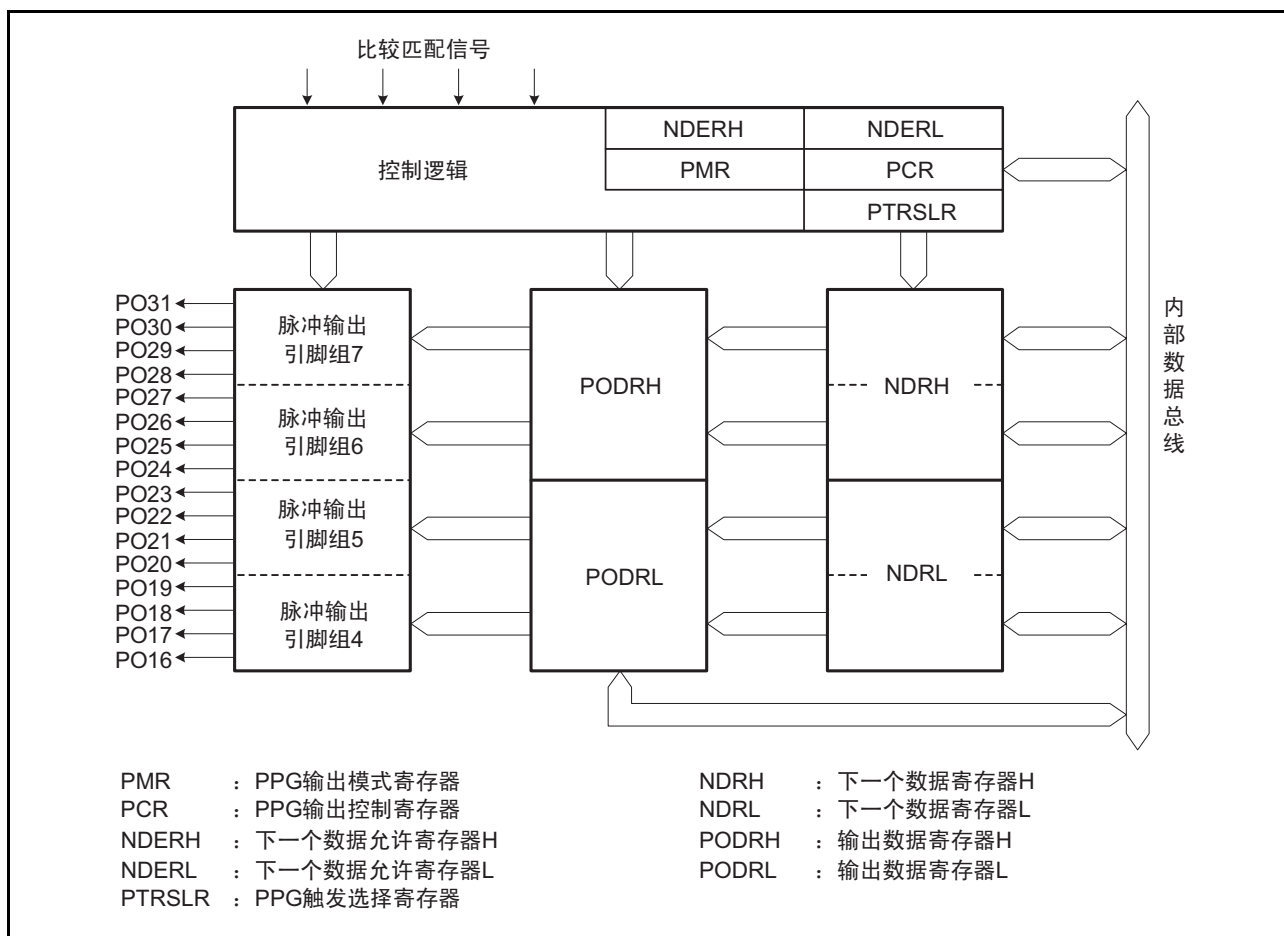


图 16.2 PPG (单元 1) 的框图

PPG 使用的输入 / 输出引脚如表 16.3 所示。

表 16.3 PPG 的输入 / 输出引脚

| 单元 | 引脚名 | 输入 / 输出 | 功能 |
|------|------|---------|---------------|
| PPG0 | PO0 | 输出 | 脉冲输出组 0 的脉冲输出 |
| | PO1 | 输出 | |
| | PO2 | 输出 | |
| | PO3 | 输出 | |
| | PO4 | 输出 | 脉冲输出组 1 的脉冲输出 |
| | PO5 | 输出 | |
| | PO6 | 输出 | |
| | PO7 | 输出 | |
| | PO8 | 输出 | 脉冲输出组 2 的脉冲输出 |
| | PO9 | 输出 | |
| | PO10 | 输出 | |
| | PO11 | 输出 | |
| | PO12 | 输出 | 脉冲输出组 3 的脉冲输出 |
| | PO13 | 输出 | |
| | PO14 | 输出 | |
| PO15 | 输出 | | |
| PPG1 | PO16 | 输出 | 脉冲输出组 4 的脉冲输出 |
| | PO17 | 输出 | |
| | PO18 | 输出 | |
| | PO19 | 输出 | |
| | PO20 | 输出 | 脉冲输出组 5 的脉冲输出 |
| | PO21 | 输出 | |
| | PO22 | 输出 | |
| | PO23 | 输出 | |
| | PO24 | 输出 | 脉冲输出组 6 的脉冲输出 |
| | PO25 | 输出 | |
| | PO26 | 输出 | |
| | PO27 | 输出 | |
| | PO28 | 输出 | 脉冲输出组 7 的脉冲输出 |
| | PO29 | 输出 | |
| | PO30 | 输出 | |
| PO31 | 输出 | | |

16.2 寄存器说明

PPG 的寄存器一览表如表 16.4 所示。

表 16.4 PPG 的寄存器一览表

| 单元 | 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|------------|--------------|--------|------------------|--------------------------|------|
| PPG0 | PPG 输出控制寄存器 | PCR | FFh | 0008 81E6h | 8 |
| | PPG 输出模式寄存器 | PMR | F0h | 0008 81E7h | 8 |
| | 下一个数据允许寄存器 H | NDERH | 00h | 0008 81E8h | 8 |
| | 下一个数据允许寄存器 L | NDERL | 00h | 0008 81E9h 0008 81E9h | 8 |
| | 输出数据寄存器 H | PODRH | 00h | 0008 81EAh 0008 81EAh | 8 |
| | 输出数据寄存器 L | PODRL | 00h | 0008 81EBh | 8 |
| | 下一个数据寄存器 H | NDRH | 00h | 0008 81ECh (注 1) | 8 |
| | 下一个数据寄存器 L | NDRL | 00h | 0008 81EDh (注 2) | 8 |
| | 下一个数据寄存器 H | NDRH | 00h | 0008 81EEh (注 1) | 8 |
| | 下一个数据寄存器 L | NDRL | 00h | 0008 81EFh (注 2) | 8 |
| PPG1 | PPG 触发选择寄存器 | PTRSLR | 01h | 0008 81F0h | 8 |
| | PPG 输出控制寄存器 | PCR | FFh | 0008 81F6h | 8 |
| | PPG 输出模式寄存器 | PMR | F0h | 0008 81F7h | 8 |
| | 下一个数据允许寄存器 H | NDERH | 00h | 0008 81F8h | 8 |
| | 下一个数据允许寄存器 L | NDERL | 00h | 0008 81F9h | 8 |
| | 输出数据寄存器 H | PODRH | 00h | 0008 81FAh | 8 |
| | 输出数据寄存器 L | PODRL | 00h | 0008 81FBh | 8 |
| | 下一个数据寄存器 H | NDRH | 00h | 0008 81FCh (注 3) | 8 |
| | 下一个数据寄存器 L | NDRL | 00h | 0008 81FDh (注 3) | 8 |
| | 下一个数据寄存器 H | NDRH | 00h | 0008 81FEh (注 4) | 8 |
| 下一个数据寄存器 L | NDRL | 00h | 0008 81FFh (注 4) | 8 | |

注 1. 通过设定 PPG0.PCR，当脉冲输出组 2 和脉冲输出组 3 的输出触发设定值相同时，PPG0.NDRH 的地址为 0008 81ECh；当输出触发设定值不同时，对应脉冲输出组 2 和脉冲输出组 3 的 PPG0.NDRH 的地址分别为 0008 81EEh 和 0008 81ECh。

注 2. 通过设定 PPG0.PCR，当脉冲输出组 0 和脉冲输出组 1 的输出触发设定值相同时，PPG0.NDRL 的地址为 0008 81EDh；当输出触发设定值不同时，对应脉冲输出组 0 和脉冲输出组 1 的 PPG0.NDRL 的地址分别为 0008 81EFh 和 0008 81EDh。

注 3. 通过设定 PPG1.PCR，当脉冲输出组 6 和脉冲输出组 7 的输出触发设定值相同时，PPG1.NDRH 的地址为 0008 81FCh；当输出触发设定值不同时，对应脉冲输出组和脉冲输出组 7 的 PPG1.NDRH 的地址分别为 0008 81FEh 和 0008 81FCh。

注 4. 通过设定 PPG1.PCR，当脉冲输出组 4 和脉冲输出组 5 的输出触发设定值相同时，PPG1.NDRL 的地址为 0008 81FDh；当输出触发设定值不同时，对应脉冲输出组 4 和脉冲输出组 5 的 PPG1.NDRL 的地址分别为 0008 81FFh 和 0008 81FDh。

16.2.1 PPG 触发选择寄存器 (PTRSLR)

地址 0008 81F0h

| | | | | | | | |
|----|----|----|----|----|----|----|-------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | — | — | — | PTRSL |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

复位后的值

• PPG1.PTRSLR

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-----------|--|-----|
| b0 | PTRSL | PPG 触发选择位 | 0: PPG1 的触发为 TPU0 ~ TPU3 1: PPG1 的触发为 TPU6 ~ TPU9 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

PPG1.PTRSLR 寄存器是选择触发输入的寄存器。

PTRSL 位 (PPG 触发选择位)

此位从 TPU0 ~ TPU3 或者 TPU6 ~ TPU9 中选择 PPG1 的触发输入。

如果将此位置“0”，PPG1 的触发输入就为 TPU0 ~ TPU3；如果置“1”，PPG1 的触发输入就为 TPU6 ~ TPU9。

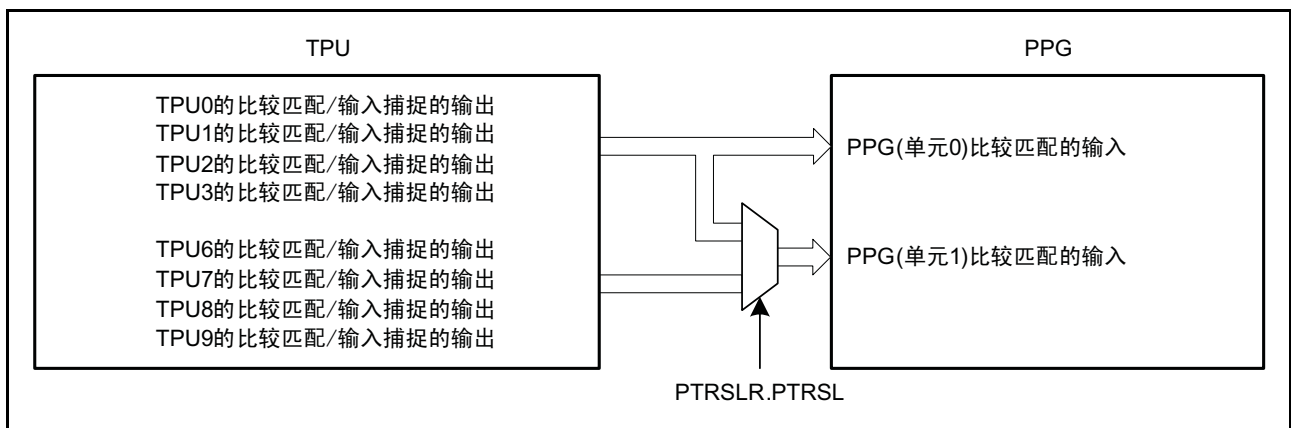


图 16.3 PPG 触发选择的框图

16.2.2 下一个数据允许寄存器 H (NDERH)、下一个数据允许寄存器 L (NDERL)

地址 0008 81E8h

• PPG0.NDERH

| | | | | | | | | |
|-------|--------|--------|--------|--------|--------|--------|-------|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | NDER15 | NDER14 | NDER13 | NDER12 | NDER11 | NDER10 | NDER9 | NDER8 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

地址 0008 81E9h

• PPG0.NDERL

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | NDER7 | NDER6 | NDER5 | NDER4 | NDER3 | NDER2 | NDER1 | NDER0 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

• PPG0.NDERH

| 位 | 符号 | 位名 | 功能 | R/W |
|----|--------|------------|------------------------|-----|
| b0 | NDER8 | 下一个数据传送允许位 | 0: 禁止数据传送 1: 允许数据传送 | R/W |
| b1 | NDER9 | 下一个数据传送允许位 | | R/W |
| b2 | NDER10 | 下一个数据传送允许位 | | R/W |
| b3 | NDER11 | 下一个数据传送允许位 | | R/W |
| b4 | NDER12 | 下一个数据传送允许位 | | R/W |
| b5 | NDER13 | 下一个数据传送允许位 | | R/W |
| b6 | NDER14 | 下一个数据传送允许位 | | R/W |
| b7 | NDER15 | 下一个数据传送允许位 | | R/W |

PPG0.NDERH 寄存器以位为单位允许或者禁止 PPG 产生的 PO8 ~ PO15 脉冲输出。

NDERj 位 (下一个数据传送允许位) (j=8 ~ 15)

如果将此位置“1”，就在产生由 PTRSLR 寄存器选择的触发时，将数据从 PPG0.NDRH 寄存器的对应位传送到 PPG0.PODRH 寄存器。但是，为“0”的位不进行从 PPG0.NDRH 寄存器到 PPG0.PODRH 寄存器的数据传送。

- PPG0.NDERL

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|------------|------------------------|-----|
| b0 | NDER0 | 下一个数据传送允许位 | 0: 禁止数据传送 1: 允许数据传送 | R/W |
| b1 | NDER1 | 下一个数据传送允许位 | | R/W |
| b2 | NDER2 | 下一个数据传送允许位 | | R/W |
| b3 | NDER3 | 下一个数据传送允许位 | | R/W |
| b4 | NDER4 | 下一个数据传送允许位 | | R/W |
| b5 | NDER5 | 下一个数据传送允许位 | | R/W |
| b6 | NDER6 | 下一个数据传送允许位 | | R/W |
| b7 | NDER7 | 下一个数据传送允许位 | | R/W |

PPG0.NDERL 寄存器以位为单位允许或者禁止 PPG 产生的 PO0 ~ PO7 脉冲输出。

NDERj 位 (下一个数据传送允许位) (j=0 ~ 7)

如果将此位置“1”，就在产生由 PTRSLR 寄存器选择的触发时，将数据从 PPG0.NDERL 寄存器的对应位传送到 PPG0.PODRL 寄存器。但是，为“0”的位不进行从 PPG0.NDERL 寄存器到 PPG0.PODRL 寄存器的数据传送。

地址 0008 81F8h

• PPG1.NDERH

| | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | NDER | NDER | NDER | NDER | NDER | NDER | NDER | NDER |
| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

地址 0008 81F9h

• PPG1.NDERL

| | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | NDER | NDER | NDER | NDER | NDER | NDER | NDER | NDER |
| | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

• PPG1.NDERH

| 位 | 符号 | 位名 | 功能 | R/W |
|----|--------|------------|------------------------|-----|
| b0 | NDER24 | 下一个数据传送允许位 | 0: 禁止数据传送 1: 允许数据传送 | R/W |
| b1 | NDER25 | 下一个数据传送允许位 | | R/W |
| b2 | NDER26 | 下一个数据传送允许位 | | R/W |
| b3 | NDER27 | 下一个数据传送允许位 | | R/W |
| b4 | NDER28 | 下一个数据传送允许位 | | R/W |
| b5 | NDER29 | 下一个数据传送允许位 | | R/W |
| b6 | NDER30 | 下一个数据传送允许位 | | R/W |
| b7 | NDER31 | 下一个数据传送允许位 | | R/W |

PPG1.NDERH 寄存器以位为单位允许或者禁止 PPG 产生的 PO24 ~ PO31 脉冲输出。

NDER_j 位 (下一个数据传送允许位) (j=24 ~ 31)

如果将此位置“1”，就在产生由 PTRSLR 寄存器选择的触发时，将数据从 PPG1.NDRH 寄存器的对应位传送到 PPG1.PODRH 寄存器。但是，为“0”的位不进行从 PPG1.NDRH 寄存器到 PPG1.PODRH 寄存器的数据传送。

• PPG1.NDERL

| 位 | 符号 | 位名 | 功能 | R/W |
|----|--------|------------|------------------------|-----|
| b0 | NDER16 | 下一个数据传送允许位 | 0: 禁止数据传送 1: 允许数据传送 | R/W |
| b1 | NDER17 | 下一个数据传送允许位 | | R/W |
| b2 | NDER18 | 下一个数据传送允许位 | | R/W |
| b3 | NDER19 | 下一个数据传送允许位 | | R/W |
| b4 | NDER20 | 下一个数据传送允许位 | | R/W |
| b5 | NDER21 | 下一个数据传送允许位 | | R/W |
| b6 | NDER22 | 下一个数据传送允许位 | | R/W |
| b7 | NDER23 | 下一个数据传送允许位 | | R/W |

PPG1.NDERL 寄存器以位为单位允许或者禁止 PPG 产生的 PO16 ~ PO23 脉冲输出。

NDER_j 位 (下一个数据传送允许位) (j=16 ~ 23)

如果将此位置“1”，就在产生由 PTRSLR 寄存器选择的触发时，将数据从 PPG1.NDRL 寄存器的对应位传送到 PPG1.PODRL 寄存器。但是，为“0”的位不进行 PPG1.NDRL 寄存器到 PPG1.PODRL 寄存器的数据从传送。

16.2.3 输出数据寄存器 H (PODRH)、输出数据寄存器 L (PODRL)

地址 0008 81EAh

• PPG0.PODRH

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|------|------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | POD15 | POD14 | POD13 | POD12 | POD11 | POD10 | POD9 | POD8 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

地址 0008 81EBh

• PPG0.PODRL

| | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | POD7 | POD6 | POD5 | POD4 | POD3 | POD2 | POD1 | POD0 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

• PPG0.PODRH

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|----------|---|-----|
| b0 | POD8 | 输出数据寄存器位 | 只限于通过 PPG0.NDERH 寄存器设定为允许数据传送的位。如果在 PPG 运行过程中产生输出触发, 就将 PPG0.NDRH 寄存器的值传送到此寄存器。在 PPG0.NDERH.NDERj (j=8 ~ 15) 的某位为“1”的期间, CPU 不能写此寄存器。能在 PPG0.NDERH 寄存器为“00h”的状态下设定脉冲的初始输出值。 | R/W |
| b1 | POD9 | 输出数据寄存器位 | | R/W |
| b2 | POD10 | 输出数据寄存器位 | | R/W |
| b3 | POD11 | 输出数据寄存器位 | | R/W |
| b4 | POD12 | 输出数据寄存器位 | | R/W |
| b5 | POD13 | 输出数据寄存器位 | | R/W |
| b6 | POD14 | 输出数据寄存器位 | | R/W |
| b7 | POD15 | 输出数据寄存器位 | | R/W |

PPG0.PODRH 寄存器保存脉冲输出值。对于由 PPG0.NDERH 寄存器设定为脉冲输出的位, 通过输出触发将 PPG0.NDRH 寄存器的值传送到此寄存器。

• PPG0.PODRL

| 位 | 符号 | 位名 | 功能 | R/W |
|----|------|----------|--|-----|
| b0 | POD0 | 输出数据寄存器位 | 只限于通过 PPG0.NDERL 寄存器设定为允许数据传送的位。如果在 PPG 运行过程中产生输出触发, 就将 PPG0.NDRL 寄存器的值传送到此寄存器。在 PPG0.NDERL.NDERj (j=0 ~ 7) 的某位为“1”的期间, CPU 不能写此寄存器。能在 PPG0.NDERL 寄存器为“00h”的状态下设定脉冲的初始输出值。 | R/W |
| b1 | POD1 | 输出数据寄存器位 | | R/W |
| b2 | POD2 | 输出数据寄存器位 | | R/W |
| b3 | POD3 | 输出数据寄存器位 | | R/W |
| b4 | POD4 | 输出数据寄存器位 | | R/W |
| b5 | POD5 | 输出数据寄存器位 | | R/W |
| b6 | POD6 | 输出数据寄存器位 | | R/W |
| b7 | POD7 | 输出数据寄存器位 | | R/W |

PPG0.PODRL 寄存器保存脉冲输出值。对于由 PPG0.NDERL 寄存器设定为脉冲输出的位, 通过输出触发将 PPG0.NDRL 寄存器的值传送到此寄存器。

地址 0008 81FAh

• PPG1.PODRH

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | POD31 | POD30 | POD29 | POD28 | POD27 | POD26 | POD25 | POD24 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

地址 0008 81FBh

• PPG1.PODRL

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | POD23 | POD22 | POD21 | POD20 | POD19 | POD18 | POD17 | POD16 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

• PPG1.PODRH

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|----------|--|-----|
| b0 | POD24 | 输出数据寄存器位 | 只限于通过 PPG1.NDERH 寄存器设定为允许数据传送的位。如果在 PPG 运行过程中产生输出触发, 就将 PPG1.NDRH 寄存器的值传送到此寄存器。在 PPG1.NDERH.NDERj (j=24 ~ 31) 的某位为“1”的期间, CPU 不能写此寄存器。能在 PPG1.NDERH 寄存器为“00h”的状态下设定脉冲的初始输出值。 | R/W |
| b1 | POD25 | 输出数据寄存器位 | | R/W |
| b2 | POD26 | 输出数据寄存器位 | | R/W |
| b3 | POD27 | 输出数据寄存器位 | | R/W |
| b4 | POD28 | 输出数据寄存器位 | | R/W |
| b5 | POD29 | 输出数据寄存器位 | | R/W |
| b6 | POD30 | 输出数据寄存器位 | | R/W |
| b7 | POD31 | 输出数据寄存器位 | | R/W |

PPG1.PODRH 寄存器保存脉冲输出值。对于由 PPG1.NDERH 寄存器设定为脉冲输出的位, 通过输出触发将 PPG1.NDRH 寄存器的值传送到此寄存器。

• PPG1.PODRL

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|----------|--|-----|
| b0 | POD16 | 输出数据寄存器位 | 只限于通过 PPG1.NDERL 寄存器设定为允许数据传送的位。如果在 PPG 运行过程中产生输出触发, 就将 PPG1.NDRL 寄存器的值传送到此寄存器。在 PPG1.NDERL.NDERj (j=16 ~ 23) 的某位为“1”的期间, CPU 不能写此寄存器。能在 PPG1.NDERL 寄存器为“00h”的状态下设定脉冲的初始输出值。 | R/W |
| b1 | POD17 | 输出数据寄存器位 | | R/W |
| b2 | POD18 | 输出数据寄存器位 | | R/W |
| b3 | POD19 | 输出数据寄存器位 | | R/W |
| b4 | POD20 | 输出数据寄存器位 | | R/W |
| b5 | POD21 | 输出数据寄存器位 | | R/W |
| b6 | POD22 | 输出数据寄存器位 | | R/W |
| b7 | POD23 | 输出数据寄存器位 | | R/W |

PPG1.PODRL 寄存器保存脉冲输出值。对于由 PPG1.NDERL 寄存器设定为脉冲输出的位, 通过输出触发将 PPG1.NDRL 寄存器的值传送到此寄存器。

16.2.4 下一个数据寄存器 H (NDRH)、下一个数据寄存器 L (NDRL)

地址 0008 81ECh、0008 81EEh

• PPG0.NDRH

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|------|------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | NDR15 | NDR14 | NDR13 | NDR12 | NDR11 | NDR10 | NDR9 | NDR8 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

地址 0008 81EDh、0008 81EFh

• PPG0.NDRL

| | | | | | | | | |
|-------|------|------|------|------|------|------|------|------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | NDR7 | NDR6 | NDR5 | NDR4 | NDR3 | NDR2 | NDR1 | NDR0 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

• PPG0.NDRH

PPG0.NDRH 寄存器保存脉冲输出的下一个数据。PPG0.NDRH 寄存器的地址因将脉冲输出组设定为相同的输出触发或者不同的输出触发而不同。

(1) 脉冲输出组 2 和脉冲输出组 3 的输出触发相同的情况

如果将脉冲输出组 2 和脉冲输出组 3 设定为相同的输出触发，就能将此 8 位全部映像到相同的地址，并且能同时存取此 8 位。

(0008 81ECh)

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|-----------|--|-----|
| b0 | NDR8 | 下一个数据寄存器位 | 通过由 PPG0.PCR 寄存器指定的输出触发，将此寄存器的值传送到 PPG0.PODRH 寄存器的对应位。 | R/W |
| b1 | NDR9 | 下一个数据寄存器位 | | R/W |
| b2 | NDR10 | 下一个数据寄存器位 | | R/W |
| b3 | NDR11 | 下一个数据寄存器位 | | R/W |
| b4 | NDR12 | 下一个数据寄存器位 | | R/W |
| b5 | NDR13 | 下一个数据寄存器位 | | R/W |
| b6 | NDR14 | 下一个数据寄存器位 | | R/W |
| b7 | NDR15 | 下一个数据寄存器位 | | R/W |

(2) 脉冲输出组 2 和脉冲输出组 3 的输出触发不同的情况

如果脉冲输出组 2 和脉冲输出组 3 选择不同的输出触发，就将高 4 位和低 4 位映像到不同的地址。

(脉冲输出组 3: 0008 81ECh)

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-----------|--|-----|
| b3-b0 | — | 保留位 | 读写值都为“1”。 | R/W |
| b4 | NDR12 | 下一个数据寄存器位 | 通过由 PPG0.PCR 寄存器指定的输出触发，将此寄存器的值传送到 PPG0.PODRH 寄存器的对应位。 | R/W |
| b5 | NDR13 | 下一个数据寄存器位 | | R/W |
| b6 | NDR14 | 下一个数据寄存器位 | | R/W |
| b7 | NDR15 | 下一个数据寄存器位 | | R/W |

(脉冲输出组2: 0008 81EEh)

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-----------|---|-----|
| b0 | NDR8 | 下一个数据寄存器位 | 通过由 PPG0.PCR 寄存器指定的输出触发, 将此寄存器的值传送到 PPG0.PODRH 寄存器的对应位。 | R/W |
| b1 | NDR9 | 下一个数据寄存器位 | | R/W |
| b2 | NDR10 | 下一个数据寄存器位 | | R/W |
| b3 | NDR11 | 下一个数据寄存器位 | | R/W |
| b7-b4 | — | 保留位 | 读写值都为“1”。 | R/W |

• PPG0.NDRL

PPG0.NDRL 寄存器保存脉冲输出的下一个数据。PPG0.NDRL 寄存器的地址因将脉冲输出组设定为相同的输出触发或者不同的输出触发而不同。

(1) 脉冲输出组0和脉冲输出组1的输出触发相同的情况

如果将脉冲输出组0和脉冲输出组1设定为相同的输出触发, 就能将此8位全部映像到相同的地址, 并且能同时存取此8位。

(0008 81EDh)

| 位 | 符号 | 位名 | 功能 | R/W |
|----|------|-----------|---|-----|
| b0 | NDR0 | 下一个数据寄存器位 | 通过由 PPG0.PCR 寄存器指定的输出触发, 将此寄存器的值传送到 PPG0.PODRL 寄存器的对应位。 | R/W |
| b1 | NDR1 | 下一个数据寄存器位 | | R/W |
| b2 | NDR2 | 下一个数据寄存器位 | | R/W |
| b3 | NDR3 | 下一个数据寄存器位 | | R/W |
| b4 | NDR4 | 下一个数据寄存器位 | | R/W |
| b5 | NDR5 | 下一个数据寄存器位 | | R/W |
| b6 | NDR6 | 下一个数据寄存器位 | | R/W |
| b7 | NDR7 | 下一个数据寄存器位 | | R/W |

(2) 脉冲输出组0和脉冲输出组1的输出触发不同的情况

如果脉冲输出组0和脉冲输出组1选择不同的输出触发, 就将高4位和低4位映像到不同的地址。

(脉冲输出组1: 0008 81EDh)

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|-----------|---|-----|
| b3-b0 | — | 保留位 | 读写值都为“1”。 | R/W |
| b4 | NDR4 | 下一个数据寄存器位 | 通过由 PPG0.PCR 寄存器指定的输出触发, 将此寄存器的值传送到 PPG0.PODRL 寄存器的对应位。 | R/W |
| b5 | NDR5 | 下一个数据寄存器位 | | R/W |
| b6 | NDR6 | 下一个数据寄存器位 | | R/W |
| b7 | NDR7 | 下一个数据寄存器位 | | R/W |

(脉冲输出组0: 0008 81EFh)

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|-----------|---|-----|
| b0 | NDR0 | 下一个数据寄存器位 | 通过由 PPG0.PCR 寄存器指定的输出触发, 将此寄存器的值传送到 PPG0.PODRL 寄存器的对应位。 | R/W |
| b1 | NDR1 | 下一个数据寄存器位 | | R/W |
| b2 | NDR2 | 下一个数据寄存器位 | | R/W |
| b3 | NDR3 | 下一个数据寄存器位 | | R/W |
| b7-b4 | — | 保留位 | 读写值都为“1”。 | R/W |

地址 0008 81FCh、0008 81FEh

• PPG1.NDRH

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | NDR31 | NDR30 | NDR29 | NDR28 | NDR27 | NDR26 | NDR25 | NDR24 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

地址 0008 81FDh、0008 81FFh

• PPG1.NDRL

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | NDR23 | NDR22 | NDR21 | NDR20 | NDR19 | NDR18 | NDR17 | NDR16 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

• PPG1.NDRH

PPG1.NDRH 寄存器保存脉冲输出的下一个数据。PPG1.NDRH 寄存器的地址因将脉冲输出组设定为相同的输出触发或者不同的输出触发的而不同。

(1) 脉冲输出组 6 和脉冲输出组 7 的输出触发相同的情况

如果将脉冲输出组 6 和脉冲输出组 7 设定为相同的输出触发，就能将此 8 位全部映像到相同的地址，并且能同时存取此 8 位。

(0008 81FCh)

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|-----------|--|-----|
| b0 | NDR24 | 下一个数据寄存器位 | 通过由 PPG1.PCR 寄存器指定的输出触发，将此寄存器的值传送到 PPG1.PODRH 寄存器的对应位。 | R/W |
| b1 | NDR25 | 下一个数据寄存器位 | | R/W |
| b2 | NDR26 | 下一个数据寄存器位 | | R/W |
| b3 | NDR27 | 下一个数据寄存器位 | | R/W |
| b4 | NDR28 | 下一个数据寄存器位 | | R/W |
| b5 | NDR29 | 下一个数据寄存器位 | | R/W |
| b6 | NDR30 | 下一个数据寄存器位 | | R/W |
| b7 | NDR31 | 下一个数据寄存器位 | | R/W |

(2) 脉冲输出组 6 和脉冲输出组 7 的输出触发不同的情况

如果脉冲输出组 6 和脉冲输出组 7 选择不同的输出触发，就将高 4 位和低 4 位映像到不同的地址。

(脉冲输出组 7: 0008 81FCh)

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-----------|--|-----|
| b3-b0 | — | 保留位 | 读写值都为“1”。 | R/W |
| b4 | NDR28 | 下一个数据寄存器位 | 通过由 PPG1.PCR 寄存器指定的输出触发，将此寄存器的值传送到 PPG1.PODRH 寄存器的对应位。 | R/W |
| b5 | NDR29 | 下一个数据寄存器位 | | R/W |
| b6 | NDR30 | 下一个数据寄存器位 | | R/W |
| b7 | NDR31 | 下一个数据寄存器位 | | R/W |

(脉冲输出组6: 0008 81FEh)

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-----------|---|-----|
| b0 | NDR24 | 下一个数据寄存器位 | 通过由 PPG1.PCR 寄存器指定的输出触发, 将此寄存器的值传送到 PPG1.PODRH 寄存器的对应位。 | R/W |
| b1 | NDR25 | 下一个数据寄存器位 | | R/W |
| b2 | NDR26 | 下一个数据寄存器位 | | R/W |
| b3 | NDR27 | 下一个数据寄存器位 | | R/W |
| b7-b4 | — | 保留位 | 读写值都为“1”。 | R/W |

• PPG1.NDRL

PPG1.NDRL 寄存器保存脉冲输出的下一个数据。PPG1.NDRL 寄存器的地址因将脉冲输出组设定为相同的输出触发或者不同的输出触发而不同。

(1) 脉冲输出组4和脉冲输出组5的输出触发相同的情况

如果将脉冲输出组4和脉冲输出组5设定为相同的输出触发, 就能将此8位全部映像到相同的地址, 并且能同时存取此8位。

(0008 81FDh)

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|-----------|---|-----|
| b0 | NDR16 | 下一个数据寄存器位 | 通过由 PPG1.PCR 寄存器指定的输出触发, 将此寄存器的值传送到 PPG1.PODRL 寄存器的对应位。 | R/W |
| b1 | NDR17 | 下一个数据寄存器位 | | R/W |
| b2 | NDR18 | 下一个数据寄存器位 | | R/W |
| b3 | NDR19 | 下一个数据寄存器位 | | R/W |
| b4 | NDR20 | 下一个数据寄存器位 | | R/W |
| b5 | NDR21 | 下一个数据寄存器位 | | R/W |
| b6 | NDR22 | 下一个数据寄存器位 | | R/W |
| b7 | NDR23 | 下一个数据寄存器位 | | R/W |

(2) 脉冲输出组4和脉冲输出组5的输出触发不同的情况

如果脉冲输出组4和脉冲输出组5选择不同的输出触发, 就将高4位和低4位映像到不同的地址。

(脉冲输出组5: 0008 81FDh)

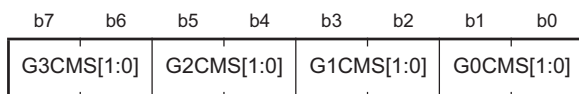
| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-----------|---|-----|
| b3-b0 | — | (保留位) | 读写值都为“1”。 | R/W |
| b4 | NDR20 | 下一个数据寄存器位 | 通过由 PPG1.PCR 寄存器指定的输出触发, 将此寄存器的值传送到 PPG1.PODRL 寄存器的对应位。 | R/W |
| b5 | NDR21 | 下一个数据寄存器位 | | R/W |
| b6 | NDR22 | 下一个数据寄存器位 | | R/W |
| b7 | NDR23 | 下一个数据寄存器位 | | R/W |

(脉冲输出组4: 0008 81FFh)

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-----------|---|-----|
| b0 | NDR16 | 下一个数据寄存器位 | 通过由 PPG1.PCR 寄存器指定的输出触发, 将此寄存器的值传送到 PPG1.PODRL 寄存器的对应位。 | R/W |
| b1 | NDR17 | 下一个数据寄存器位 | | R/W |
| b2 | NDR18 | 下一个数据寄存器位 | | R/W |
| b3 | NDR19 | 下一个数据寄存器位 | | R/W |
| b7-b4 | — | 保留位 | 读写值都为“1”。 | R/W |

16.2.5 PPG 输出控制寄存器 (PCR)

地址 PPG0.PCR 0008 81E6h、PPG1.PCR 0008 81F6h



复位后的值 1 1 1 1 1 1 1 1

• PPG0.PCR

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------------|--------------|---|-----|
| b1-b0 | G0CMS[1:0] | 组 0 的比较匹配选择位 | b1 b0 0 0: TPU0 的比较匹配 0 1: TPU1 的比较匹配 1 0: TPU2 的比较匹配 1 1: TPU3 的比较匹配 | R/W |
| b3-b2 | G1CMS[1:0] | 组 1 的比较匹配选择位 | b3 b2 0 0: TPU0 的比较匹配 0 1: TPU1 的比较匹配 1 0: TPU2 的比较匹配 1 1: TPU3 的比较匹配 | R/W |
| b5-b4 | G2CMS[1:0] | 组 2 的比较匹配选择位 | b5 b4 0 0: TPU0 的比较匹配 0 1: TPU1 的比较匹配 1 0: TPU2 的比较匹配 1 1: TPU3 的比较匹配 | R/W |
| b7-b6 | G3CMS[1:0] | 组 3 的比较匹配选择位 | b7 b6 0 0: TPU0 的比较匹配 0 1: TPU1 的比较匹配 1 0: TPU2 的比较匹配 1 1: TPU3 的比较匹配 | R/W |

• PPG1.PCR

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------------|--------------|--|-----|
| b1-b0 | G0CMS[1:0] | 组 4 的比较匹配选择位 | 当 PPG1.PTRSLR.PTRSL 位为 “0” 时 b1 b0 0 0: TPU0 的比较匹配 0 1: TPU1 的比较匹配 1 0: TPU2 的比较匹配 1 1: TPU3 的比较匹配 当 PPG1.PTRSLR.PTRSL 位为 “1” 时 b1 b0 0 0: TPU6 的比较匹配 0 1: TPU7 的比较匹配 1 0: TPU8 的比较匹配 1 1: TPU9 的比较匹配 | R/W |
| b3-b2 | G1CMS[1:0] | 组 5 的比较匹配选择位 | 当 PPG1.PTRSLR.PTRSL 位为 “0” 时 b3 b2 0 0: TPU0 的比较匹配 0 1: TPU1 的比较匹配 1 0: TPU2 的比较匹配 1 1: TPU3 的比较匹配 当 PPG1.PTRSLR.PTRSL 位为 “1” 时 b3 b2 0 0: TPU6 的比较匹配 0 1: TPU7 的比较匹配 1 0: TPU8 的比较匹配 1 1: TPU9 的比较匹配 | R/W |
| b5-b4 | G2CMS[1:0] | 组 6 的比较匹配选择位 | 当 PPG1.PTRSLR.PTRSL 位为 “0” 时 b5 b4 0 0: TPU0 的比较匹配 0 1: TPU1 的比较匹配 1 0: TPU2 的比较匹配 1 1: TPU3 的比较匹配 当 PPG1.PTRSLR.PTRSL 位为 “1” 时 b5 b4 0 0: TPU6 的比较匹配 0 1: TPU7 的比较匹配 1 0: TPU8 的比较匹配 1 1: TPU9 的比较匹配 | R/W |
| b7-b6 | G3CMS[1:0] | 组 7 的比较匹配选择位 | 当 PPG1.PTRSLR.PTRSL 位为 “0” 时 b7 b6 0 0: TPU0 的比较匹配 0 1: TPU1 的比较匹配 1 0: TPU2 的比较匹配 1 1: TPU3 的比较匹配 当 PPG1.PTRSLR.PTRSL 位为 “1” 时 b7 b6 0 0: TPU6 的比较匹配 0 1: TPU7 的比较匹配 1 0: TPU8 的比较匹配 1 1: TPU9 的比较匹配 | R/W |

PPGm.PCR 寄存器 (m=0,1) 是以组为单位选择脉冲输出触发信号的寄存器。

有关输出触发的选择, 请参照 “16.2.6 PPG 输出模式寄存器 (PMR)”。

GjCMS[1:0] 位 (组 k 的比较匹配选择位) (j=0 ~ 3, k=0 ~ 7)

这些位选择脉冲输出组 k 的输出触发。

16.2.6 PPG 输出模式寄存器 (PMR)

地址 PPG0.PMR 0008 81E7h、PPG1.PMR 0008 81F7h

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| G3INV | G2INV | G1INV | G0INV | G3NOV | G2NOV | G1NOV | G0NOV |

复位后的值 1 1 1 1 1 1 1 1

• PPG0.PMR

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|--------------|---|-----|
| b0 | G0NOV | 组 0 的非重叠位 | 0: 正常运行 (在产生所选的 TPU _m 的比较匹配 A 时更新输出值) 1: 非重叠运行 (在产生所选的 TPU _m 的比较匹配 A 和比较匹配 B 时更新输出值) (m=0 ~ 3) | R/W |
| b1 | G1NOV | 组 1 的非重叠位 | 0: 正常运行 (在产生所选的 TPU _m 的比较匹配 A 时更新输出值) 1: 非重叠运行 (在产生所选的 TPU _m 的比较匹配 A 和比较匹配 B 时更新输出值) (m=0 ~ 3) | R/W |
| b2 | G2NOV | 组 2 的非重叠位 | 0: 正常运行 (在产生所选的 TPU _m 的比较匹配 A 时更新输出值) 1: 非重叠运行 (在产生所选的 TPU _m 的比较匹配 A 和比较匹配 B 时更新输出值) (m=0 ~ 3) | R/W |
| b3 | G3NOV | 组 3 的非重叠位 | 0: 正常运行 (在产生所选的 TPU _m 的比较匹配 A 时更新输出值) 1: 非重叠运行 (在产生所选的 TPU _m 的比较匹配 A 和比较匹配 B 时更新输出值) (m=0 ~ 3) | R/W |
| b4 | G0INV | 组 0 的输出极性变更位 | 0: 反相输出 1: 直接输出 | R/W |
| b5 | G1INV | 组 1 的输出极性变更位 | 0: 反相输出 1: 直接输出 | R/W |
| b6 | G2INV | 组 2 的输出极性变更位 | 0: 反相输出 1: 直接输出 | R/W |
| b7 | G3INV | 组 3 的输出极性变更位 | 0: 反相输出 1: 直接输出 | R/W |

• PPG1.PMR

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|--------------|---|-----|
| b0 | G0NOV | 组 4 的非重叠位 | 0: 正常运行 (在产生所选的 TPUm 的比较匹配 A 时更新输出值) 1: 非重叠运行 (在产生所选的 TPUm 的比较匹配 A 和比较匹配 B 时更新输出值) (m=0 ~ 3,6 ~ 9) | R/W |
| b1 | G1NOV | 组 5 的非重叠位 | 0: 正常运行 (在产生所选的 TPUm 的比较匹配 A 时更新输出值) 1: 非重叠运行 (在产生所选的 TPUm 的比较匹配 A 和比较匹配 B 时更新输出值) (m=0 ~ 3,6 ~ 9) | R/W |
| b2 | G2NOV | 组 6 的非重叠位 | 0: 正常运行 (在产生所选的 TPUm 的比较匹配 A 时更新输出值) 1: 非重叠运行 (在产生所选的 TPUm 的比较匹配 A 和比较匹配 B 时更新输出值) (m=0 ~ 3,6 ~ 9) | R/W |
| b3 | G3NOV | 组 7 的非重叠位 | 0: 正常运行 (在产生所选的 TPUm 的比较匹配 A 时更新输出值) 1: 非重叠运行 (在产生所选的 TPUm 的比较匹配 A 和比较匹配 B 时更新输出值) (m=0 ~ 3,6 ~ 9) | R/W |
| b4 | G0INV | 组 4 的输出极性变更位 | 0: 反相输出 1: 直接输出 | R/W |
| b5 | G1INV | 组 5 的输出极性变更位 | 0: 反相输出 1: 直接输出 | R/W |
| b6 | G2INV | 组 6 的输出极性变更位 | 0: 反相输出 1: 直接输出 | R/W |
| b7 | G3INV | 组 7 的输出极性变更位 | 0: 反相输出 1: 直接输出 | R/W |

PPGm.PMR 寄存器 (m=0,1) 是以组为单位设定 PPG 脉冲输出模式的寄存器。

当设定为反相输出时, 如果 PPGm.PODRH 寄存器和 PPGm.PODRL 寄存器的值为“1”, 就将 Low 电平输出到引脚; 如果 PPGm.PODRH 寄存器和 PPGm.PODRL 寄存器的值为“0”, 就将 High 电平输出到引脚。当设定为非重叠运行时, PPG 就在产生作为输出触发的 TPU 的比较匹配 A 和比较匹配 B 时更新输出值。

详细内容请参照“16.3.4 脉冲输出非重叠运行”。

GjNOV 位 (组 k 的非重叠位) (j=0 ~ 3, k=0 ~ 7)

这些位选择是使脉冲输出组 k 进行正常运行还是进行非重叠运行。

GjINV 位 (组 k 的反相位) (j=0 ~ 3, k=0 ~ 7)

这些位选择是使脉冲输出组 k 进行直接输出还是进行反相输出。

16.3 运行说明

PPG 概要图如图 16.4 所示。

通过将 PPGm.NDERH 寄存器和 PPGm.NDERL 寄存器 (m=0,1) 的对应位分别置“1” (允许数据传送), 使 PPG 进入脉冲输出状态。

初始输出值取决于对应的 PPGm.PODRH 寄存器和 PPGm.PODRL 寄存器的初始设定值。此后, 如果产生由 PPGm.PCR 寄存器指定的比较匹配, 就将对应的 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器的值分别传送到 PPGm.PODRH 寄存器和 PPGm.PODRL 寄存器, 更新输出值。

通过在产生下一个比较匹配前将输出数据写到 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器, 就能在每次产生比较匹配时依次输出最多 16 位的数据。

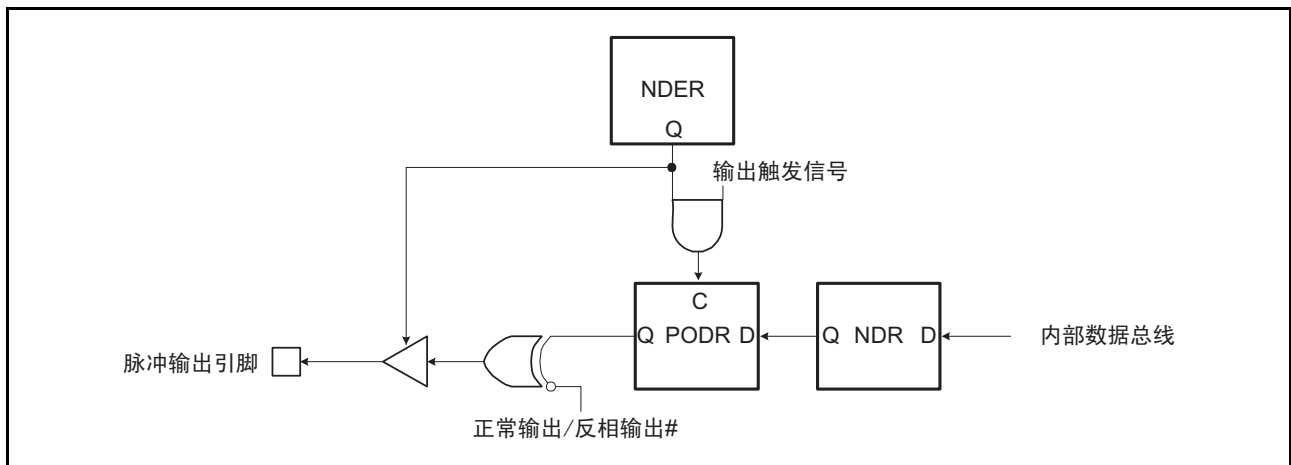


图 16.4 PPG 概要图

16.3.1 输出时序

如果在允许脉冲输出的状态下产生指定的比较匹配，就将 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器 ($m=0,1$) 的值分别传送到 PPGm.PODRH 寄存器和 PPGm.PODRL 寄存器并且进行输出。

此时序如图 16.5 所示，这是组 2 和组 3 通过比较匹配 A 进行正常输出的例子。

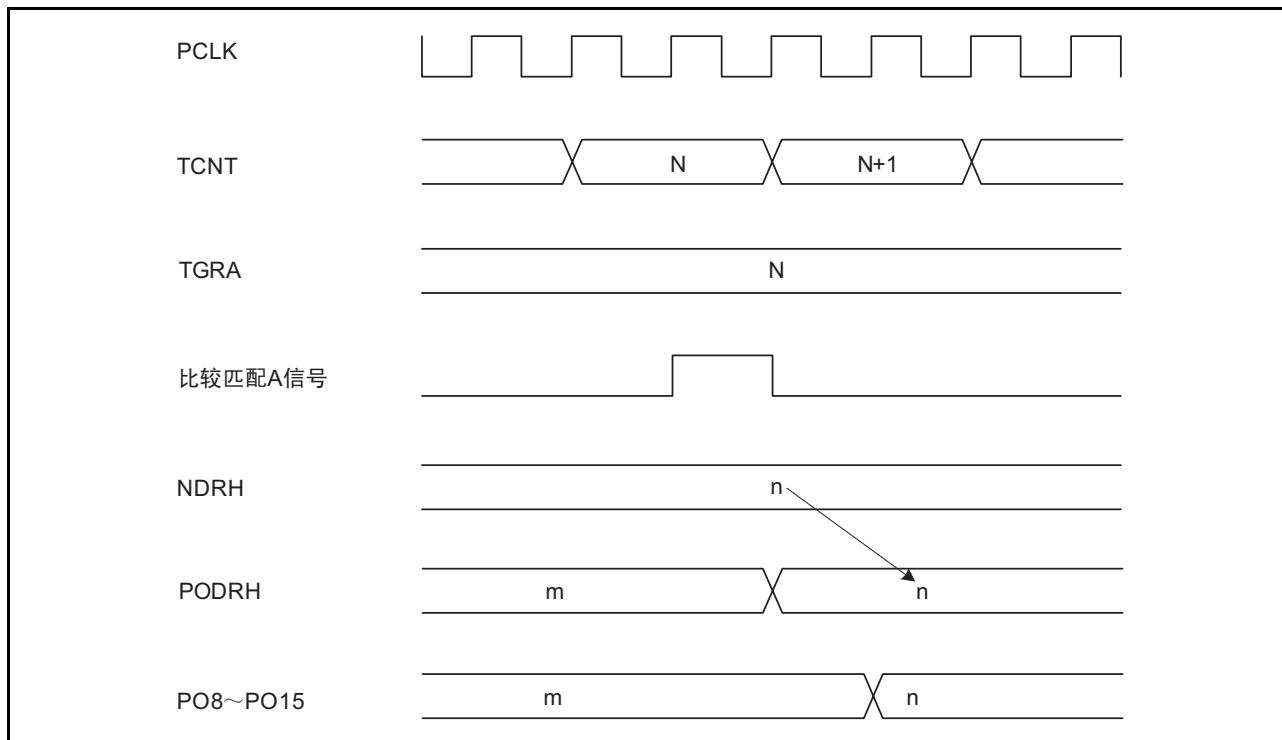


图 16.5 传送并且输出 NDR 寄存器的值的时序例子

16.3.2 脉冲输出正常运行的设定步骤例子

脉冲输出正常运行的设定步骤例子如图 16.6 和图 16.7 所示。

(1) PPG0 的设定

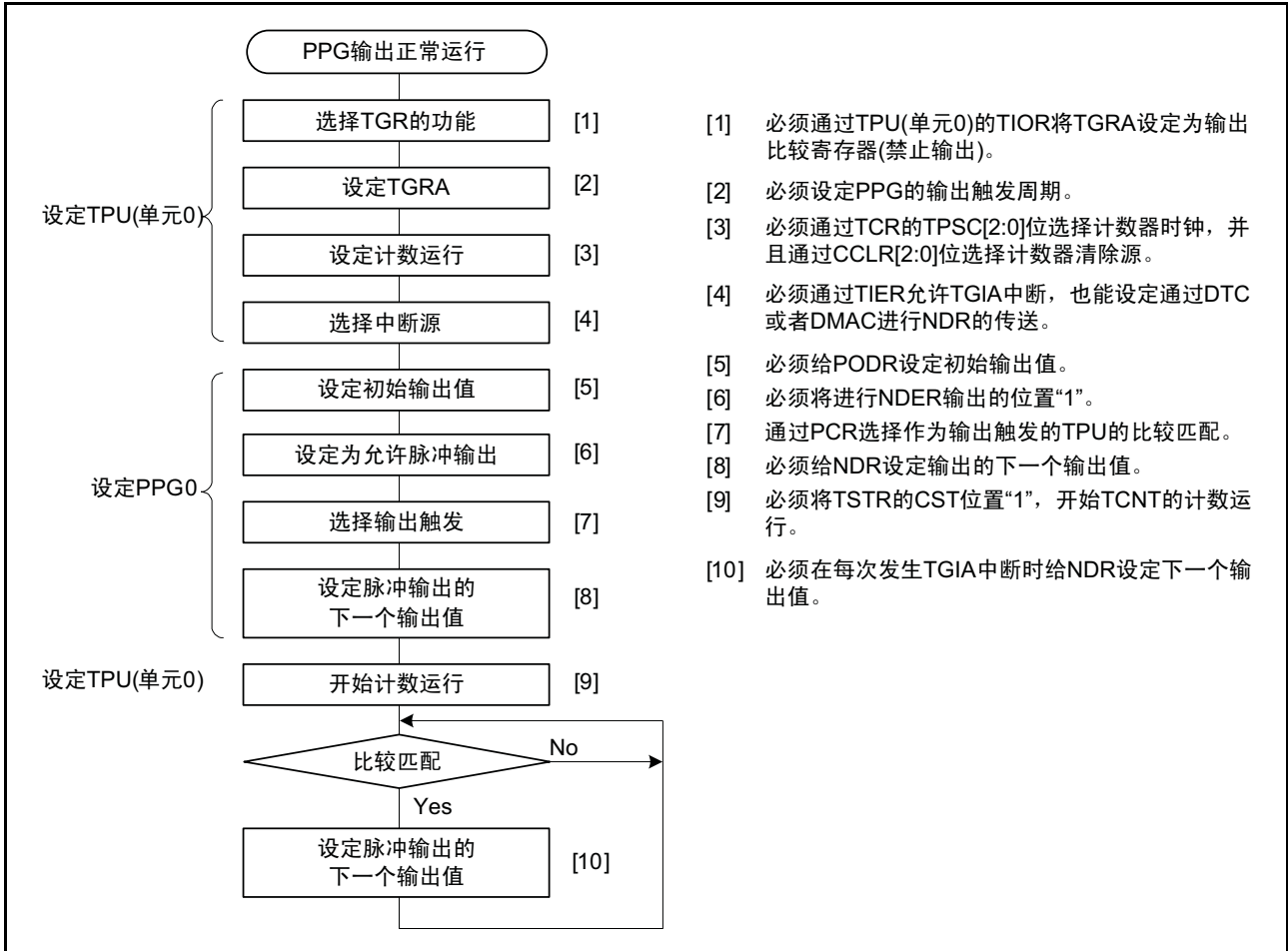


图 16.6 脉冲输出正常运行的设定步骤例子 (PPG0 的设定)

(2) PPG1 的设置

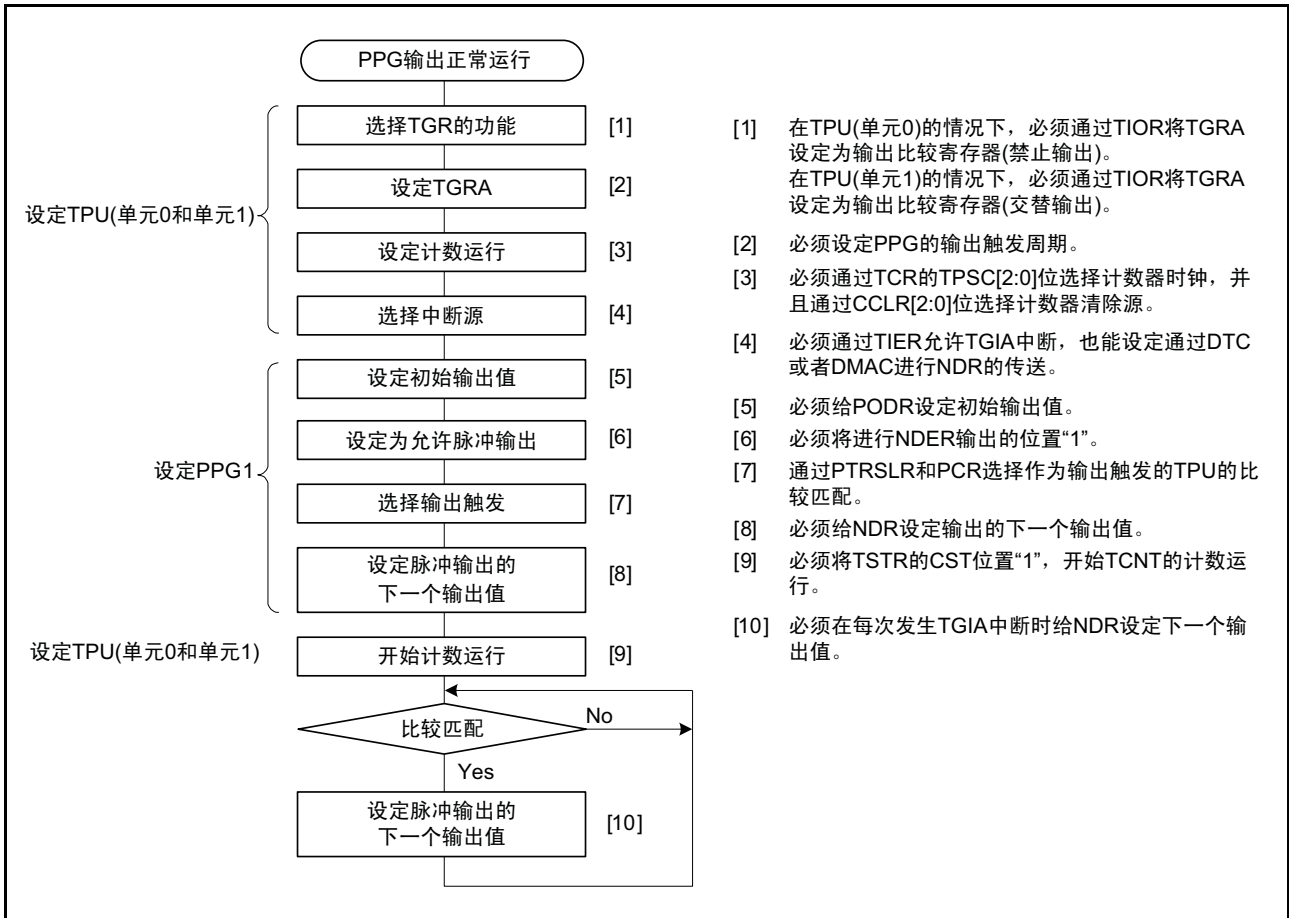


图 16.7 脉冲输出正常运行的设定步骤例子 (PPG1 的设置)

16.3.3 脉冲输出正常运行的例子 (5相脉冲输出例子)

使用脉冲输出, 以一定的周期输出 5 相脉冲的 PPG0 的例子如图 16.8 所示。

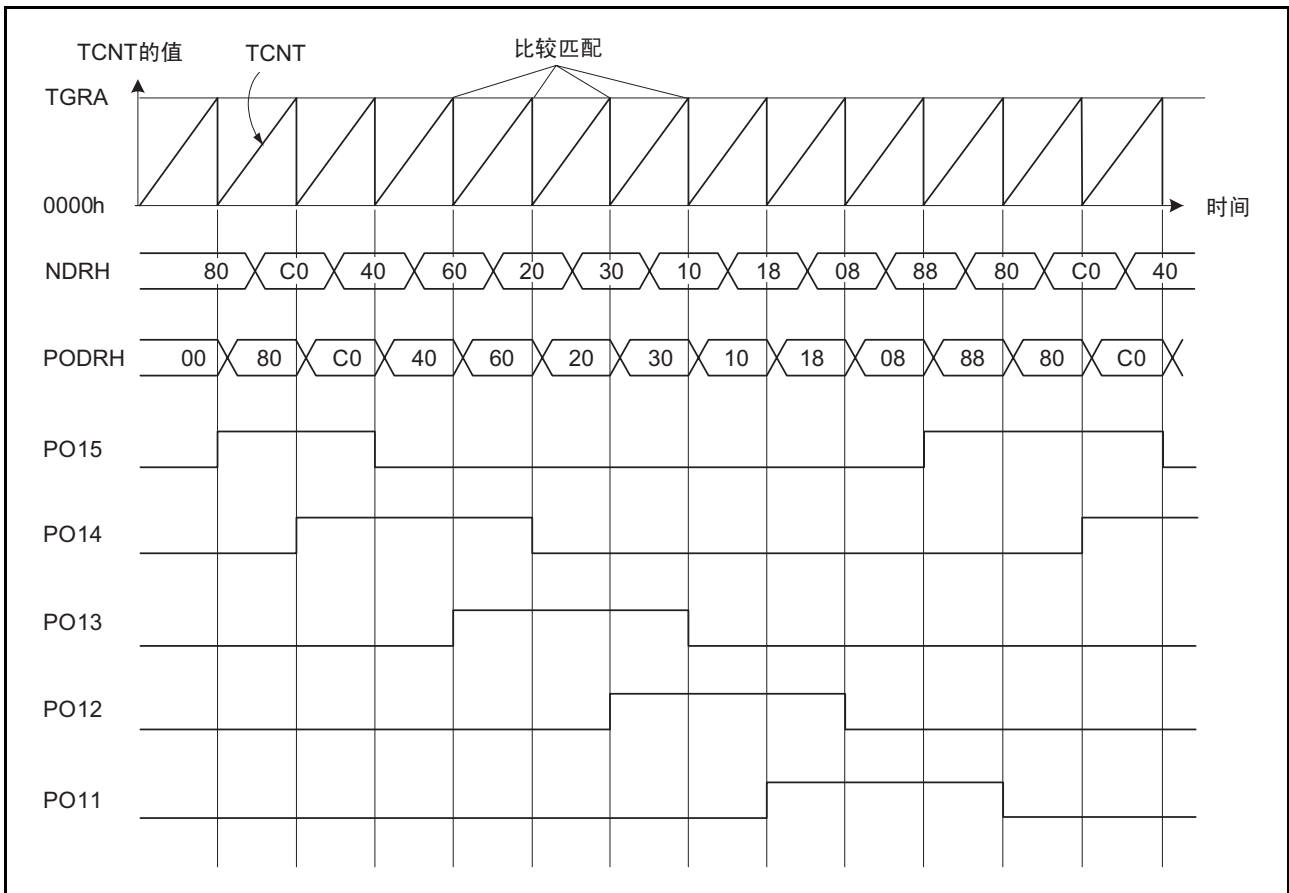


图 16.8 脉冲输出正常运行的例子 (5相脉冲输出例子)

1. 将作为输出触发的 TPU 的 TPUm.TGRA 寄存器 (m=0~3) 设定为输出比较寄存器。给 TGRA 寄存器设定周期并且选择通过比较匹配 A 进行计数器的清除。将 TPUm.TIER.TGIEA 位置 “1”，允许比较匹配/输入捕捉 A (TGImA) 中断。
2. 将 “F8h” 写到 PPG0.NDRH 寄存器，并且通过 PPG0.PCR.G3CMS[1:0] 位和 G2CMS[1:0] 位，将输出触发设定到上述 1. 中选择的 TPUm 的比较匹配。将输出数据 “80h” 写到 PPG0.NDRH 寄存器。
3. 一旦 TPU 对应的通道开始运行并且产生比较匹配 A，就将 PPG0.NDRH 寄存器的值传送到 PPG0.PODRH 寄存器并且进行输出。通过 TGImA 中断处理将下一个输出数据 “C0h” 写到 PPG0.NDRH 寄存器。
4. 通过 TGImA 中断依次写 “40h”、“60h”、“20h”、“30h”、“10h”、“18h”、“08h”、“88h”……，进行 5 相的 1-2 相脉冲输出。

如果设定为通过 TGImA 中断启动 DTC 或者 DMAC，就能在 CPU 没有负担的情况下进行脉冲输出。

16.3.4 脉冲输出非重叠运行

非重叠运行时的从 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器 (m=0,1) 到 PPGm.PODRH 寄存器和 PPGm.PODRL 寄存器的传送如下:

- 在产生比较匹配 A 时, 总是将 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器的值传送到 PPGm.PODRH 寄存器和 PPGm.PODRL 寄存器。
- 在产生比较匹配 B 时, 只在 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器的传送位为 “0” 时进行传送, 为 “1” 时不进行传送。

脉冲输出非重叠运行如图 16.9 所示。

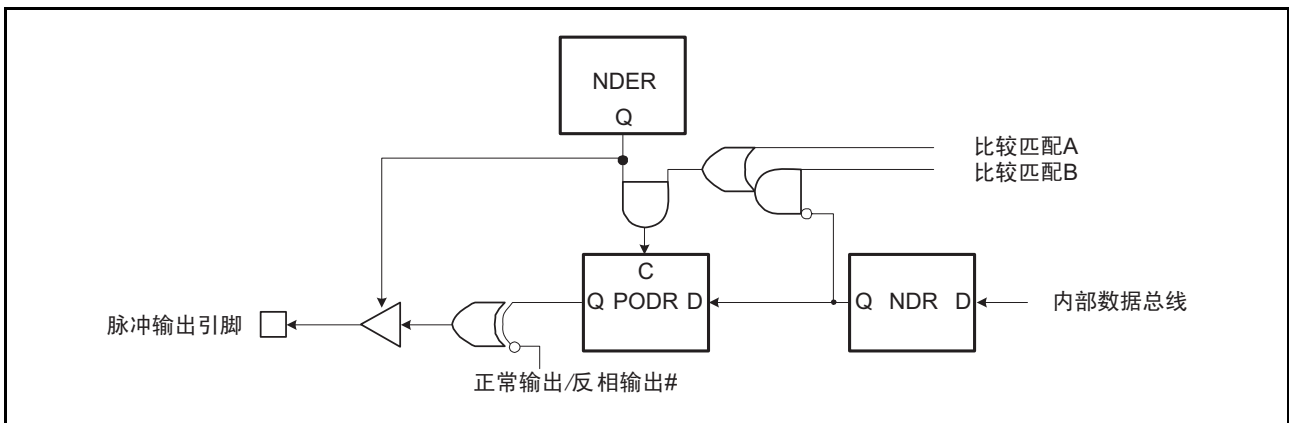


图 16.9 脉冲输出非重叠运行

因此, 能通过在产生比较匹配 A 前产生比较匹配 B, 在传送 “1” 的数据前传送 “0” 的数据。

此时, 在产生比较匹配 B 后到产生比较匹配 A 前的期间 (非重叠期间), 不能更改 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器的值。必须通过 TGIA 中断的中断处理程序将下一个数据写到 PPGm.NDRH 寄存器和 PPGm.NDRL 寄存器, 也能通过 TGIA 中断启动 DTC 或者 DMAC。但是, 必须在产生下一个比较匹配 B 前进行此写操作。

此时序如图 16.10 所示。

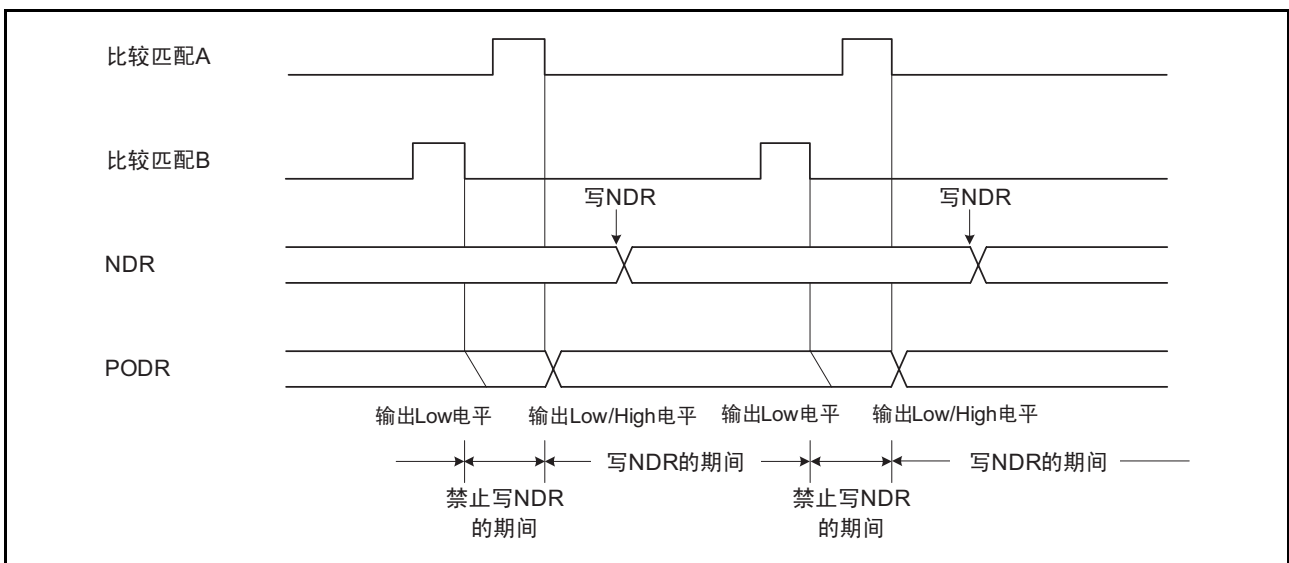


图 16.10 非重叠运行和 PPGm.NDRH、PPGm.NDRL 寄存器的写时序

16.3.5 脉冲输出非重叠运行的设定步骤例子

脉冲输出非重叠运行的设定步骤例子如图 16.11 和图 16.12 所示。

(1) PPG0 的设定

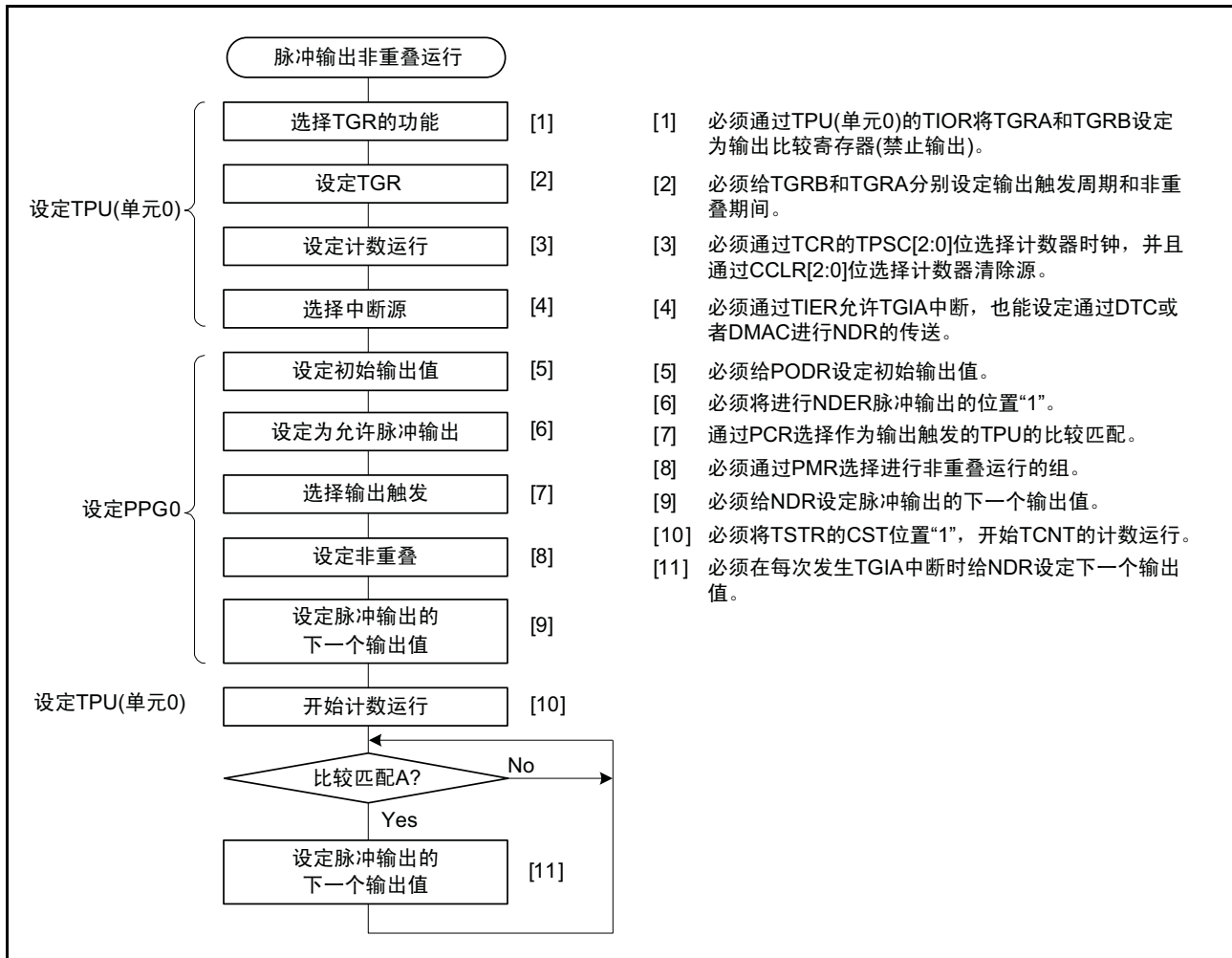


图 16.11 脉冲输出非重叠运行的设定步骤例子 (PPG0 的设定)

(2) PPG1 的设置

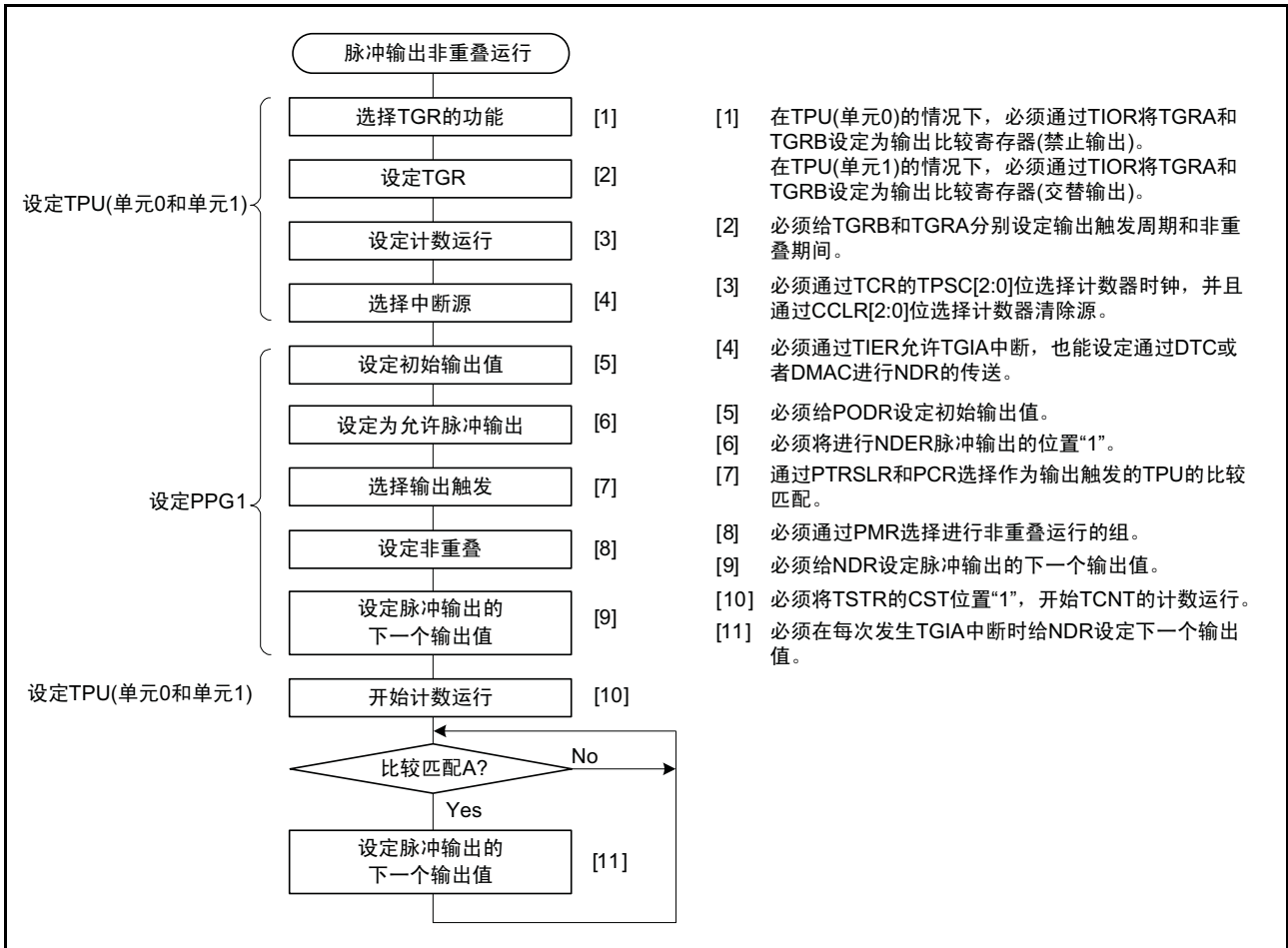


图 16.12 脉冲输出非重叠运行的设定步骤例子 (PPG1 的设置)

16.3.6 脉冲输出非重叠运行的例子 (4 相互补的非重叠输出例子)

使用脉冲输出, 输出 4 相互补非重叠脉冲的 PPG0 的例子如图 16.13 所示。

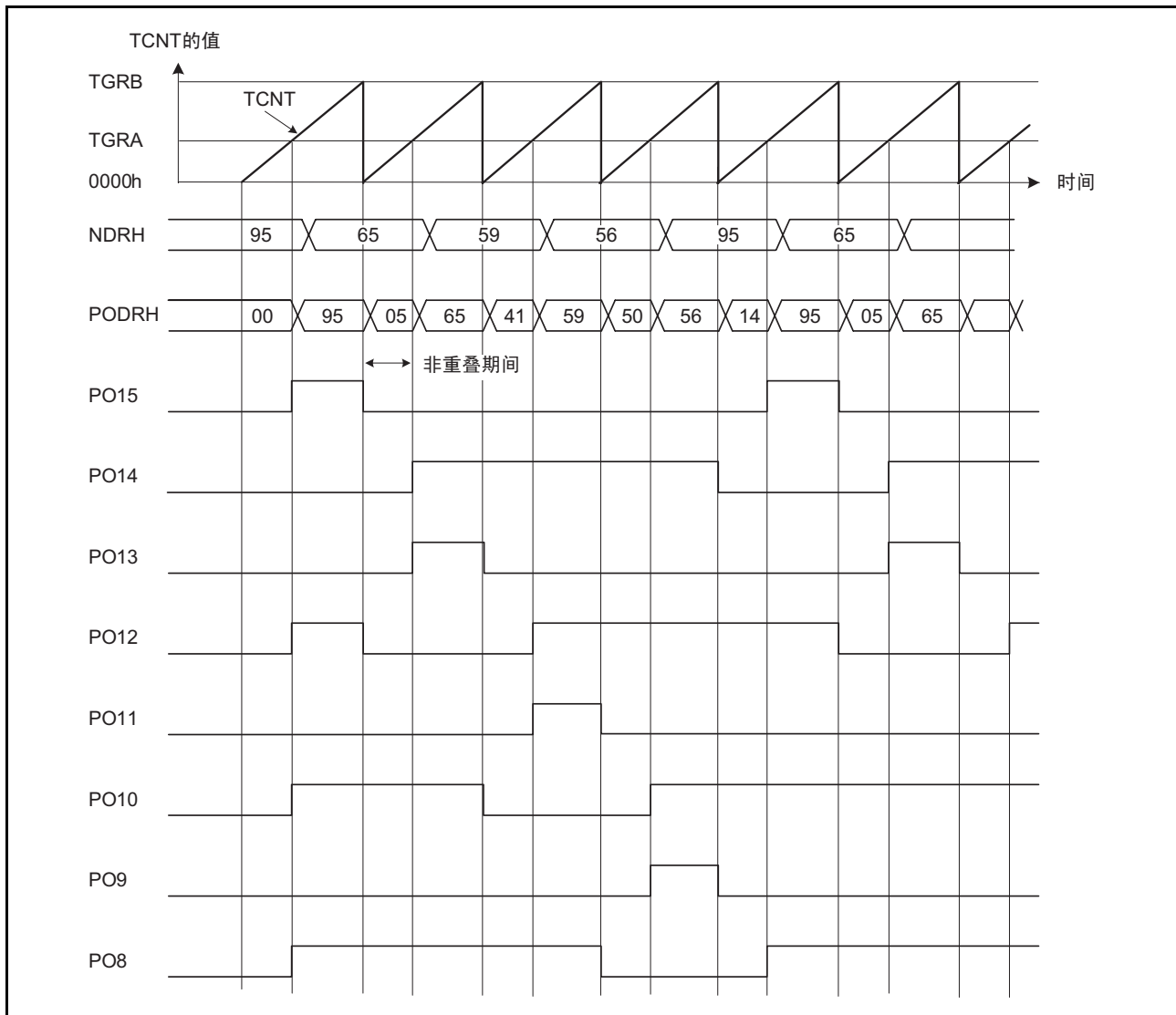


图 16.13 脉冲输出非重叠运行的例子 (4 相互补的非重叠输出)

1. 将作为输出触发的 TPU 的 TPU_m.TGRA 寄存器和 TPU_m.TGRB 寄存器 (m=0~3) 设定为输出比较寄存器。分别给 TGRB 寄存器和 TGRA 寄存器设定周期和非重叠期间, 并且选择通过比较匹配 B 进行计数器的清除。将 TPU_m.TIER.TGIEA 位置 “1”, 允许比较匹配/输入捕捉 A (TGImA) 中断。
将 “FFh” 写到 PPG0.NDRH, 并且通过 PPG0.PCR.G3CMS[1:0] 位和 G2CMS[1:0] 位, 将输出触发设定为上述 1. 中选择的 TPU_m 的比较匹配。
将 PPG0.PMR.G3NOV 位和 PPG0.PMR.G2NOV 位分别置 “1”, 设定为非重叠运行。
将输出数据 “95h” 写到 PPG0.NDRH。
3. 一旦 TPU 的该通道开始运行, 就分别通过 TGRB 寄存器和 TGRA 寄存器的比较匹配进行 High 电平输出 → Low 电平输出的转换以及 Low 电平输出 → High 电平输出的转换 (Low 电平输出 → High 电平输出的转换有可能会延迟, 延迟时间为 TGRA 寄存器的设定值)。
通过 TGImA 中断处理将下一个输出数据 “65h” 写到 PPG0.NDRH 寄存器。
4. 通过 TGImA 中断依次写 “59h”、“56h”、“95h” ……, 能产生 4 相互补的非重叠输出。
如果设定为通过 TGImA 中断启动 DTC 或者 DMAC, 就能在 CPU 没有负担的情况下进行脉冲输出。

16.3.7 脉冲的反相输出

如果将 PPG0.PMR.G3INV、PPG0.PMR.G2INV、PPG0.PMR.G1INV、PPG0.PMR.G0INV 位置“0”，就能将 PPG0.PODRH 寄存器和 PPG0.PODRL 寄存器的反相值输出到引脚。

使用图 16.13 的设定并且将 G3INV 位和 G2INV 位置“0”时的引脚输出例子如图 16.14 所示。

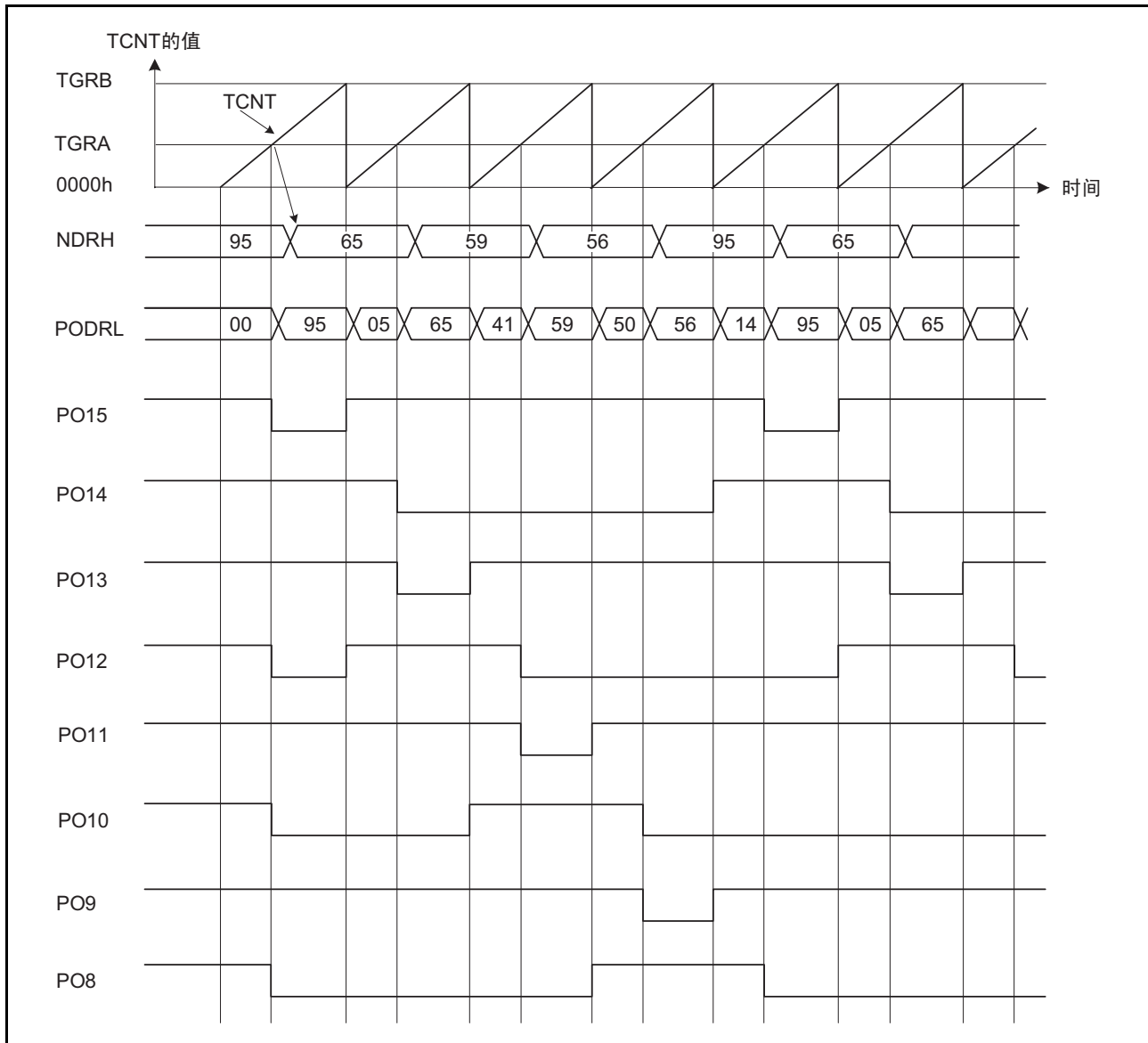


图 16.14 脉冲的反相输出例子

16.3.8 通过输入捕捉进行的脉冲输出

不仅能通过 TPU (单元 0) 的比较匹配, 而且也能通过输入捕捉进行 PPG0 的脉冲输出。在 PPG0.PCR 寄存器选择的 TPU (单元 0) 的 TPUm.TGRA 寄存器 (m=0 ~ 3) 用作输入捕捉寄存器时, 通过输入捕捉信号进行脉冲输出。

此时序如图 16.15 所示。

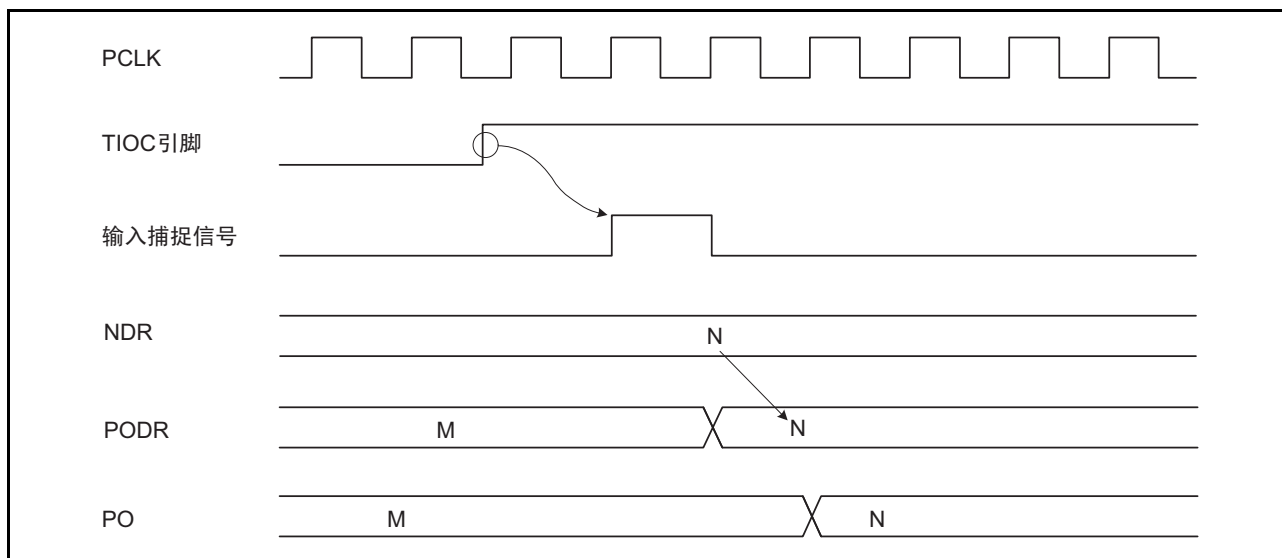


图 16.15 通过输入捕捉进行脉冲输出的例子

16.4 使用时的注意事项

16.4.1 模块停止功能的设定

能通过模块停止控制寄存器设定允许或者禁止 PPG 的运行, 初始值为停止 PPG 的运行。能通过解除模块停止状态, 使寄存器变为可存取的状态。详细内容请参照 “8. 低功耗功能”。

17. 8 位定时器 (TMR)

RX610 群内置以 8 位计数器为基础的 2 个通道的 8 位定时器 (TMR)，有 2 个单元 (单元 0 和单元 1)，共计 4 个通道。除了能对外部事件进行计数以外，还能通过和 2 个寄存器的比较匹配信号，作为多功能定时器应用于计数器复位、中断请求、任意占空比的脉冲输出等方面。

单元 0 和单元 1 的功能相同，能生成 SCI 的波特率时钟。

17.1 概要

TMR 的规格如表 17.1 所示。

单元 0 和单元 1 的框图分别如图 17.1 和图 17.2 所示。

表 17.1 TMR 的规格

| 项目 | 规格 |
|----------------|---|
| 计数时钟 | <ul style="list-style-type: none"> 内部时钟: PCLK、PCLK/2、PCLK/8、PCLK/32、PCLK/64、PCLK/1024、PCLK/8192 外部时钟 |
| 通道数 | (8 位 × 2 个通道) × 2 个单元 |
| 比较匹配 | <ul style="list-style-type: none"> 8 位模式 (比较匹配 A 和比较匹配 B) 16 位模式 (比较匹配 A 和比较匹配 B) |
| 计数器清除 | 可选择比较匹配 A、比较匹配 B 或者外部复位信号。 |
| 定时器输出 | 任意占空比的脉冲输出和 PWM 输出。 |
| 2 个通道的级联 | <ul style="list-style-type: none"> 16 位计数模式 将 TMR0 和 TMR1 分别作为高位和低位 (或者将 TMR2 和 TMR3 分别作为高位和低位) 的 16 位定时器。 比较匹配计数模式 TMR1 对 TMR0 的比较匹配进行计数 (或者 TMR3 对 TMR2 的比较匹配进行计数)。 |
| 中断源 | 比较匹配 A、比较匹配 B 或者上溢。 |
| DTC 的启动 | 能通过比较匹配 A 中断或者比较匹配 B 中断启动 DTC。 |
| A/D 转换器的转换开始触发 | TMR0 和 TMR2 的比较匹配 A (注 1)。 |
| SCI 的波特率时钟的生成 | 生成 SCI5 和 SCI6 的波特率时钟 (注 2)。 |
| 低功耗功能 | 各单元能设定为模块停止状态。 |

注 1. 详细内容请参照“23. A/D 转换器”。

注 2. 详细内容请参照“20. 串行通信接口 (SCI)”。

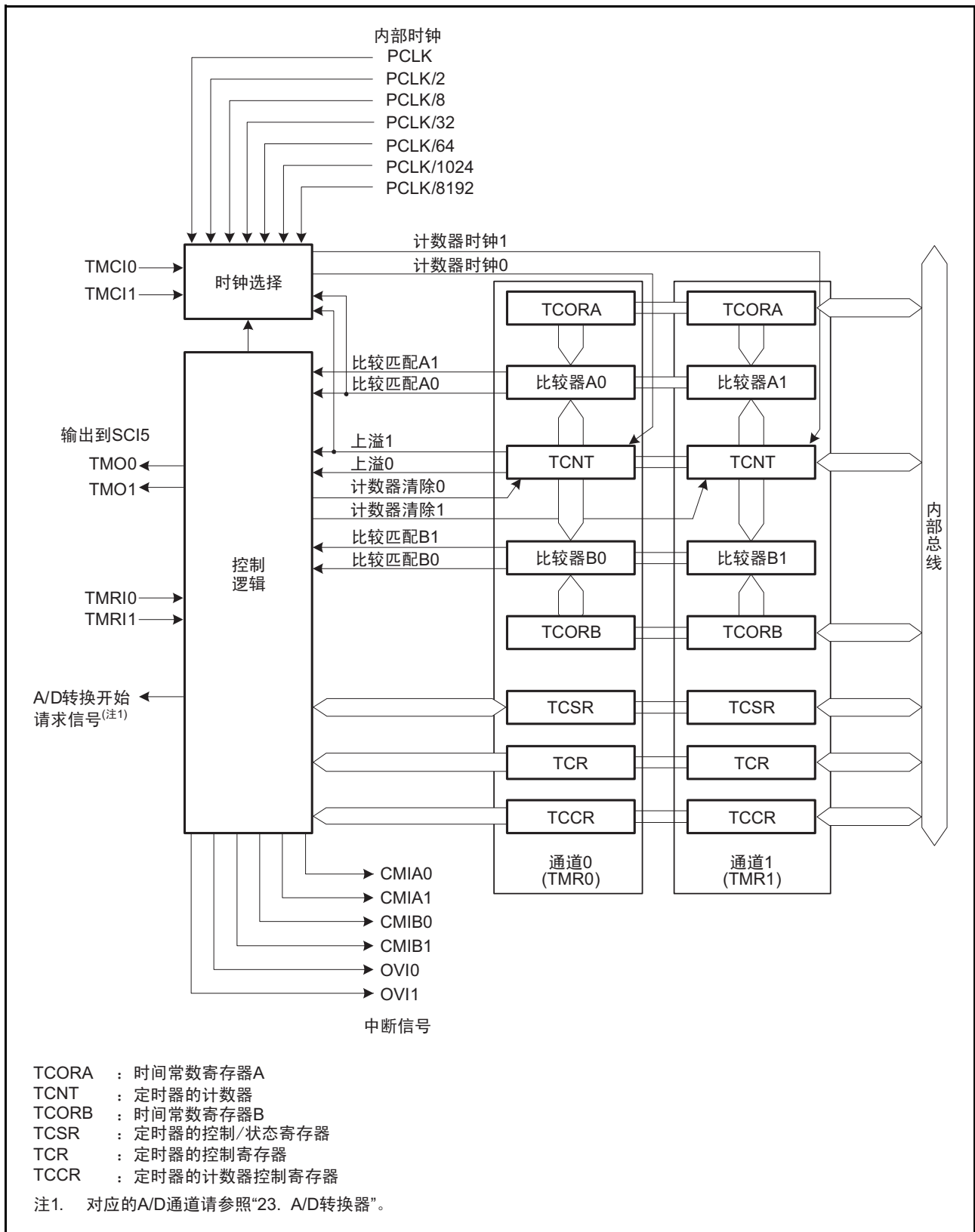


图 17.1 TMR (单元 0) 的框图

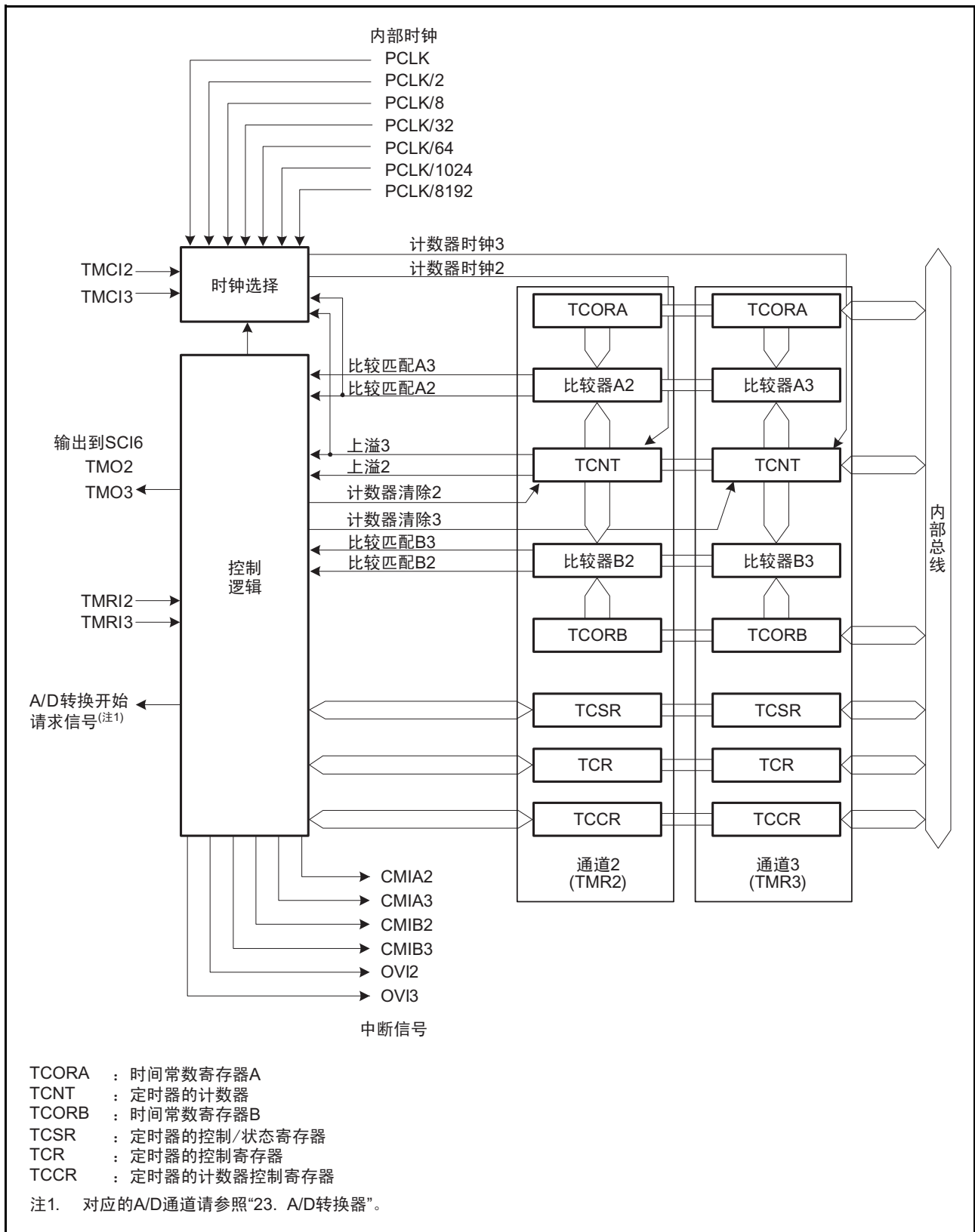


图 17.2 TMR (单元 1) 的框图

TMR 使用的输入 / 输出引脚如表 17.2 所示。

表 17.2 TMR 的输入 / 输出引脚

| 单元 | 通道 | 引脚名 | 输入 / 输出 | 功能 |
|----|------|-------|---------|------------|
| 0 | TMR0 | TMO0 | 输出 | 比较匹配的输出生 |
| | | TMCI0 | 输入 | 计数器外部时钟的输入 |
| | | TMRI0 | 输入 | 计数器外部复位的输入 |
| | TMR1 | TMO1 | 输出 | 比较匹配的输出生 |
| | | TMCI1 | 输入 | 计数器外部时钟的输入 |
| | | TMRI1 | 输入 | 计数器外部复位的输入 |
| 1 | TMR2 | TMO2 | 输出 | 比较匹配的输出生 |
| | | TMCI2 | 输入 | 计数器外部时钟的输入 |
| | | TMRI2 | 输入 | 计数器外部复位的输入 |
| | TMR3 | TMO3 | 输出 | 比较匹配的输出生 |
| | | TMCI3 | 输入 | 计数器外部时钟的输入 |
| | | TMRI3 | 输入 | 计数器外部复位的输入 |

17.2 寄存器说明

TMR 的寄存器一览表如表 17.3 所示。

表 17.3 TMR 的寄存器一览表

| 单元 | 通道 | 寄存器名 | 符号 | 复位后的值 | 地址 (注1) | 存取长度 |
|----|------|----------------|-------|-------|------------|--------------|
| 0 | TMR0 | 定时器的计数器 | TCNT | 00h | 0008 8208h | 8 或者 16 |
| | | 时间常数寄存器 A | TCORA | FFh | 0008 8204h | 8 或者 16 |
| | | 时间常数寄存器 B | TCORB | FFh | 0008 8206h | 8 或者 16 |
| | | 定时器的控制寄存器 | TCR | 00h | 0008 8200h | 8 |
| | | 定时器的计数器控制寄存器 | TCCR | 00h | 0008 820Ah | 8 或者 16 |
| | | 定时器的控制 / 状态寄存器 | TCSR | x0h | 0008 8202h | 8 |
| | TMR1 | 定时器的计数器 | TCNT | 00h | 0008 8209h | 8 或者 16 (注1) |
| | | 时间常数寄存器 A | TCORA | FFh | 0008 8205h | 8 或者 16 (注1) |
| | | 时间常数寄存器 B | TCORB | FFh | 0008 8207h | 8 或者 16 (注1) |
| | | 定时器的控制寄存器 | TCR | 00h | 0008 8201h | 8 |
| | | 定时器的计数器控制寄存器 | TCCR | 00h | 0008 820Bh | 8 或者 16 (注1) |
| | | 定时器的控制 / 状态寄存器 | TCSR | x0h | 0008 8203h | 8 |
| 1 | TMR2 | 定时器的计数器 | TCNT | 00h | 0008 8218h | 8 或者 16 |
| | | 时间常数寄存器 A | TCORA | FFh | 0008 8214h | 8 或者 16 |
| | | 时间常数寄存器 B | TCORB | FFh | 0008 8216h | 8 或者 16 |
| | | 定时器的控制寄存器 | TCR | 00h | 0008 8210h | 8 |
| | | 定时器的计数器控制寄存器 | TCCR | 00h | 0008 821Ah | 8 或者 16 |
| | | 定时器的控制 / 状态寄存器 | TCSR | x0h | 0008 8212h | 8 |
| | TMR3 | 定时器的计数器 | TCNT | 00h | 0008 8219h | 8 或者 16 (注1) |
| | | 时间常数寄存器 A | TCORA | FFh | 0008 8215h | 8 或者 16 (注1) |
| | | 时间常数寄存器 B | TCORB | FFh | 0008 8217h | 8 或者 16 (注1) |
| | | 定时器的控制寄存器 | TCR | 00h | 0008 8211h | 8 |
| | | 定时器的计数器控制寄存器 | TCCR | 00h | 0008 821Bh | 8 或者 16 (注1) |
| | | 定时器的控制 / 状态寄存器 | TCSR | x0h | 0008 8213h | 8 |

注 1. 不能对奇数地址进行 16 位存取。在对寄存器进行 16 位存取时, 必须对 TMR0 寄存器或者 TMR2 寄存器的地址进行存取。

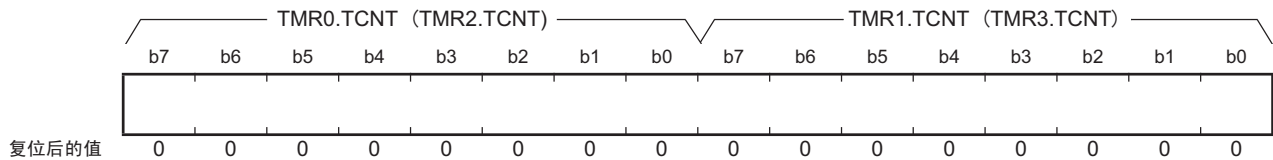
16 位存取的寄存器分配如表 17.4 所示。

表 17.4 16 位存取的寄存器分配

| 地址 | 高 8 位 | 低 8 位 |
|------------|------------|------------|
| 0008 8208h | TMR0.TCNT | TMR1.TCNT |
| 0008 8204h | TMR0.TCORA | TMR1.TCORA |
| 0008 8206h | TMR0.TCORB | TMR1.TCORB |
| 0008 820Ah | TMR0.TCCR | TMR1.TCCR |
| 0008 8218h | TMR2.TCNT | TMR3.TCNT |
| 0008 8214h | TMR2.TCORA | TMR3.TCORA |
| 0008 8216h | TMR2.TCORB | TMR3.TCORB |
| 0008 821Ah | TMR2.TCCR | TMR3.TCCR |

17.2.1 定时器的计数器 (TCNT)

地址 TMR0.TCNT 0008 8208h、TMR1.TCNT 0008 8209h
TMR2.TCNT 0008 8218h、TMR3.TCNT 0008 8219h



TCNT 计数器是 8 位可读写的递增计数器。

也能将 TMR0.TCNT 计数器和 TMR1.TCNT 计数器（或者 TMR2.TCNT 计数器和 TMR3.TCNT 计数器）作为 16 位计数器进行字存取。

能通过 TCCR.CSS[1:0] 和 TCCR.CKS[2:0] 位选择时钟。

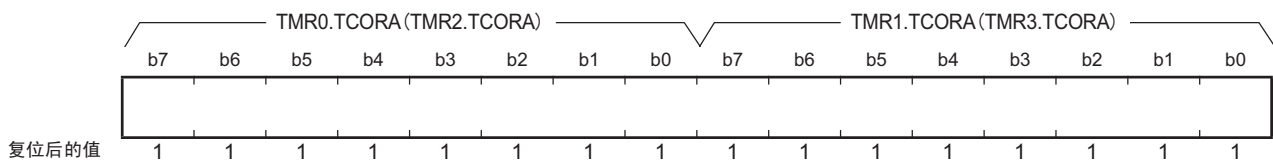
能通过外部复位输入信号、比较匹配 A 信号或者比较匹配 B 信号清除 TCNT 计数器，并且通过 TCR.CCLR[1:0] 位选择用哪个信号进行清除。

如果 TCNT 计数器发生上溢 (“FFh” → “00h”)，中断标志就变为 “1”。

对应的中断向量号请参照 “10. 中断控制器 (ICU)” 和 “表 17.6 TMR 的中断源”。

17.2.2 时间常数寄存器 A (TCORA)

地址 TMR0.TCORA 0008 8204h、TMR1.TCORA 0008 8205h
TMR2.TCORA 0008 8214h、TMR3.TCORA 0008 8215h



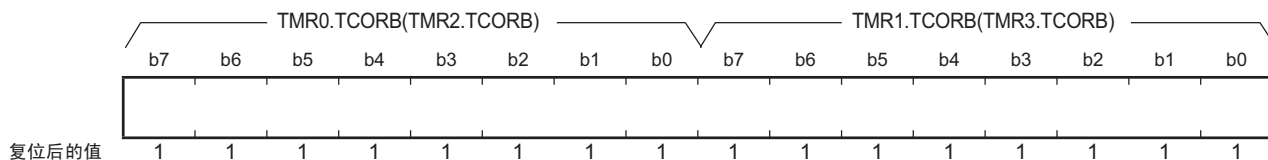
TCORA 寄存器是 8 位可读写寄存器。

也能将 TMR0.TCORA 寄存器和 TMR1.TCORA 寄存器（或者 TMR2.TCORA 寄存器和 TMR3.TCORA 寄存器）作为 16 位寄存器进行字存取。

TCORA 寄存器的值随时和 TCNT 计数器进行比较，如果两者的值相同，比较匹配 A 信号就变为 High 电平。但是，在写 TCORA 寄存器时不进行比较。还能通过 TCSR.OSA[1:0] 位的设定和此比较匹配 A 信号，控制 TMO_n 引脚的定时器输出。

17.2.3 时间常数寄存器 B (TCORB)

地址 TMR0.TCORB 0008 8206h、TMR1.TCORB 0008 8207h
TMR2.TCORB 0008 8216h、TMR3.TCORB 0008 8217h



TCORB 寄存器是 8 位可读写寄存器。

也能将 TMR0.TCORB 寄存器和 TMR1.TCORB 寄存器（或者 TMR2.TCORB 寄存器和 TMR3.TCORB 寄存器）作为 16 位寄存器进行字存取。

TCORB 寄存器的值随时和 TCNT 计数器进行比较。如果两者的值相同，比较匹配 B 信号就变为 High 电平。但是，在写 TCORB 寄存器时不进行比较。还能通过 TCSR.OSB[1:0] 位的设定和此比较匹配 B 信号，控制 TMO_n 引脚的定时器输出。

17.2.4 定时器的控制寄存器 (TCR)

地址 TMR0.TCR 0008 8200h、TMR1.TCR 0008 8201h
TMR2.TCR 0008 8210h、TMR3.TCR 0008 8211h

| | | | | | | | |
|-------|-------|------|-----------|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| CMIEB | CMIEA | OVIE | CCLR[1:0] | — | — | — | — |

复位后的值 0 0 0 0 0 0 0 0

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-----------|--------------|--|-----|
| b2-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b4-b3 | CCLR[1:0] | 计数器清除位 (注1) | b4 b3 00: 禁止清除 01: 通过比较匹配 A 进行清除 10: 通过比较匹配 B 进行清除 11: 通过外部复位输入进行清除 (通过 TCCR.TMRIS 位选择边沿或者电平) | R/W |
| b5 | OVIE | 定时器的上溢中断允许位 | 0: 禁止由上溢引起的中断请求 (OVIm) 1: 允许由上溢引起的中断请求 (OVIm) | R/W |
| b6 | CMIEA | 比较匹配中断允许 A 位 | 0: 禁止由比较匹配 A 引起的中断请求 (CMIAm) 1: 允许由比较匹配 A 引起的中断请求 (CMIAm) | R/W |
| b7 | CMIEB | 比较匹配中断允许 B 位 | 0: 禁止由比较匹配 B 引起的中断请求 (CMIBm) 1: 允许由比较匹配 B 引起的中断请求 (CMIBm) | R/W |

注 1. 在使用外部复位时, 必须将对应引脚的 DDR 寄存器的位和 ICR 寄存器的位分别置“0”和“1”, 详细内容请参考“14. I/O 端口”。

TCR 寄存器是指定 TCNT 计数器清除条件的寄存器。

CCLR[1:0] 位 (计数器清除位)

这些位指定 TCNT 计数器的清除条件。

OVIE 位 (定时器的上溢中断允许位)

此位选择允许或者禁止由 TCNT 计数器的上溢引起的中断请求 (OVIm)。

CMIEA 位 (比较匹配中断允许 A 位)

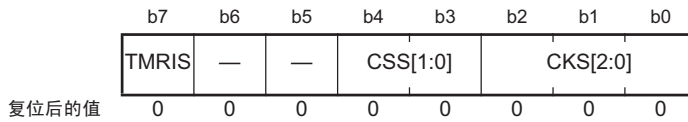
此位选择允许或者禁止在 TCORA 寄存器和 TCNT 计数器的值相同时输出的比较匹配 A 引起的中断请求 (CMIAm)。

CMIEB 位 (比较匹配中断允许 B 位)

此位选择允许或者禁止在 TCORB 寄存器和 TCNT 计数器的值相同时输出的比较匹配 B 引起的中断请求 (CMIBm)。

17.2.5 定时器的计数器控制寄存器 (TCCR)

地址 TMR0.TCCR 0008 820Ah、TMR1.TCCR 0008 820Bh
TMR2.TCCR 0008 821Ah、TMR3.TCCR 0008 821Bh



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|---------------|--|-----|
| b2-b0 | CKS[2:0] | 时钟选择位 (注1) | 请参照表 17.5。 | R/W |
| b4-b3 | CSS[1:0] | 时钟源选择位 | 请参照表 17.5。 | R/W |
| b6-b5 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | TMRIS | 定时器的复位检测条件选择位 | 0: 在外部复位的上升沿进行清除 1: 通过外部复位的 High 电平进行清除 | R/W |

注 1. 在使用外部时钟时, 必须将对应引脚的 DDR 寄存器的位和 ICR 寄存器的位分别置“0”和“1”, 详细内容请参照“14. I/O 端口”。

TCCR 寄存器是选择 TCNT 计数器的内部时钟和外部复位检测条件的寄存器。

CKS[2:0] 位 (时钟选择位)

CSS[1:0] 位 (时钟源选择位)

通过 CKS[2:0] 位和 CSS[1:0] 位选择时钟, 详细内容请参照表 17.5。

TMRIS 位 (定时器的复位检测条件选择位)

此位在 TCR.CCLR[1:0] 位为“11b” (通过外部复位输入进行清除) 时有效, 用于选择外部复位检测条件 (电平或者边沿)。

表 17.5 TCNT 计数器的输入时钟和计数条件

| 通道 | TCCR 寄存器 | | | | | 功能 | | |
|----------------|----------|----|----------|----|----|---|--------------------------|--------------------------|
| | CSS[1:0] | | CKS[2:0] | | | | | |
| | b4 | b3 | b2 | b1 | b0 | | | |
| TMR0 (TMR2) | 0 | 0 | — | 0 | 0 | 禁止时钟输入。 | | |
| | | | | | 1 | 在外部时钟的上升沿进行计数 (注1)。 | | |
| | | | | 1 | 0 | 在外部时钟的下降沿进行计数 (注1)。 | | |
| | | | | | 1 | 在外部时钟的双边沿进行计数 (注1)。 | | |
| | 0 | 1 | 0 | 0 | 0 | 内部时钟: 通过 PCLK 进行计数。 | | |
| | | | | | 1 | 内部时钟: 通过 PCLK/2 进行计数。 | | |
| | | | | | 1 | 0 | 内部时钟: 通过 PCLK/8 进行计数。 | |
| | | | | | | 1 | 内部时钟: 通过 PCLK/32 进行计数。 | |
| | | | | 1 | 0 | 0 | 内部时钟: 通过 PCLK/64 进行计数。 | |
| | | | | | | 1 | 内部时钟: 通过 PCLK/1024 进行计数。 | |
| | | | | | | 1 | 0 | 内部时钟: 通过 PCLK/8192 进行计数。 |
| | | | | | | | 1 | 禁止时钟输入。 |
| | 1 | 0 | — | — | — | 不能设定 | | |
| | 1 | 1 | — | — | — | 通过 TMR1.TCNT (TMR3.TCNT) 的上溢信号进行计数 (注2)。 | | |
| TMR1 (TMR3) | 0 | 0 | — | 0 | 0 | 禁止时钟输入。 | | |
| | | | | | 1 | 在外部时钟的上升沿进行计数 (注1)。 | | |
| | | | | 1 | 0 | 在外部时钟的下降沿进行计数 (注1)。 | | |
| | | | | | 1 | 在外部时钟的双边沿进行计数 (注1)。 | | |
| | 0 | 1 | 0 | 0 | 0 | 内部时钟: 通过 PCLK 进行计数。 | | |
| | | | | | 1 | 内部时钟: 通过 PCLK/2 进行计数。 | | |
| | | | | | 1 | 0 | 内部时钟: 通过 PCLK/8 进行计数。 | |
| | | | | | | 1 | 内部时钟: 通过 PCLK/32 进行计数。 | |
| | | | | 1 | 0 | 0 | 内部时钟: 通过 PCLK/64 进行计数。 | |
| | | | | | | 1 | 内部时钟: 通过 PCLK/1024 进行计数。 | |
| | | | | | | 1 | 0 | 内部时钟: 通过 PCLK/8192 进行计数。 |
| | | | | | | | 1 | 禁止时钟输入。 |
| | 1 | 0 | — | — | — | 不能设定 | | |
| | 1 | 1 | — | — | — | 通过 TMR0.TCNT (TMR2.TCNT) 的比较匹配 A 进行计数 (注2)。 | | |

注 1. 在使用外部时钟时, 必须将对应引脚的 DDR 寄存器的位和 ICR 寄存器的位分别置“0”和“1”, 详细内容请参照“14. I/O 端口”。

注 2. 如果将 TMR0 (或者 TMR2) 的时钟输入和 TMR1 (或者 TMR3) 的时钟输入分别用作 TMR1.TCNT (或者 TMR3.TCNT) 计数器的上溢信号和 TMR0.TCNT (或者 TMR2.TCNT) 计数器的比较匹配信号, 就不产生递增计数时钟。不能进行此设定。

17.2.6 定时器的控制 / 状态寄存器 (TCSR)

地址 TMR0.TCSR 0008 8202h、TMR2.TCSR 0008 8212h

| | | | | | | | |
|----|----|----|------|----------|----------|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | ADTE | OSB[1:0] | OSA[1:0] | | |
| x | x | x | 0 | 0 | 0 | 0 | 0 |

地址 TMR1.TCSR 0008 8203h、TMR3.TCSR 0008 8213h

| | | | | | | | |
|----|----|----|----|----------|----------|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | OSB[1:0] | OSA[1:0] | | |
| x | x | x | 1 | 0 | 0 | 0 | 0 |

x: 不定值

• TMR0.TCSR 寄存器和 TMR2.TCSR 寄存器

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|-----------------|---|-----|
| b1-b0 | OSA[1:0] | 输出选择位 A (注 1) | b1 b0 0 0: 不变 0 1: Low 电平输出 1 0: High 电平输出 1 1: 反相输出 (交替输出) | R/W |
| b3-b2 | OSB[1:0] | 输出选择位 B (注 1) | b3 b2 0 0: 不变 0 1: Low 电平输出 1 0: High 电平输出 1 1: 反相输出 (交替输出) | R/W |
| b4 | ADTE | A/D 触发允许位 (注 2) | 0: 禁止由比较匹配 A 引起的 A/D 转换开始请求 1: 允许由比较匹配 A 引起的 A/D 转换开始请求 | R/W |
| b7-b5 | — | 保留位 | 读取值为不定值, 只能写“1”。 | R/W |

注 1. 当 OSB[1:0] 位和 OSA[1:0] 位都为“0”时, 禁止定时器输出。在从复位后到产生最初的比较匹配的期间, 定时器输出为“0”。

注 2. 有关对应的 A/D 通道, 请参照“23. A/D 转换器”。

• TMR1.TCSR 寄存器和 TMR3.TCSR 寄存器

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|---------------|---|-----|
| b1-b0 | OSA[1:0] | 输出选择位 A (注 1) | b1 b0 0 0: 不变 0 1: Low 电平输出 1 0: High 电平输出 1 1: 反相输出 (交替输出) | R/W |
| b3-b2 | OSB[1:0] | 输出选择位 B (注 1) | b3 b2 0 0: 不变 0 1: Low 电平输出 1 0: High 电平输出 1 1: 反相输出 (交替输出) | R/W |
| b4 | — | 保留位 | 读写值都为“1”。 | R/W |
| b7-b5 | — | 保留位 | 读取值为不定值, 只能写“1”。 | R/W |

注 1. 当 OSB[1:0] 位和 OSA[1:0] 位都为“0”时, 禁止定时器输出。在从复位后到产生最初的比较匹配的期间, 定时器输出为“0”。

TCSR 寄存器控制比较匹配的输出。

OSA[1:0] 位 (输出选择位 A)

这些位选择 TCORA 寄存器和 TCNT 计数器的比较匹配 A 的 TMO_n 引脚的输出方法。

OSB[1:0] 位 (输出选择位 B)

这些位选择 TCORB 寄存器和 TCNT 计数器的比较匹配 B 的 TMO_n 引脚的输出方法。

ADTE 位 (A/D 触发允许位)

此位选择允许或者禁止由比较匹配 A 引起的 A/D 转换开始请求。

在 TMR1.TCSR 寄存器和 TMR3.TCSR 寄存器中, 此位为保留位。

17.3 运行说明

17.3.1 脉冲输出

任意占空比的脉冲输出例子如图 17.3 所示。

1. 为了通过 TCORA 寄存器的比较匹配清除 TCNT 计数器，将 TCR.CCLR[1:0] 位置 “01b” (通过比较匹配 A 进行清除)。
 2. 为了通过 TCORA 寄存器和 TCORB 寄存器的比较匹配分别进行 High 电平输出和 Low 电平输出，将 TCSR.OSA[1:0] 位置 “10b” (High 电平输出) 并且将 TCSR.OSB[1:0] 位置 “01b” (Low 电平输出)。
- 通过上述设定，能不通过软件而输出周期由 TCORA 寄存器、脉宽由 TCORB 寄存器决定的波形。
在从复位后到产生最初的比较匹配的期间，定时器为 Low 电平输出。

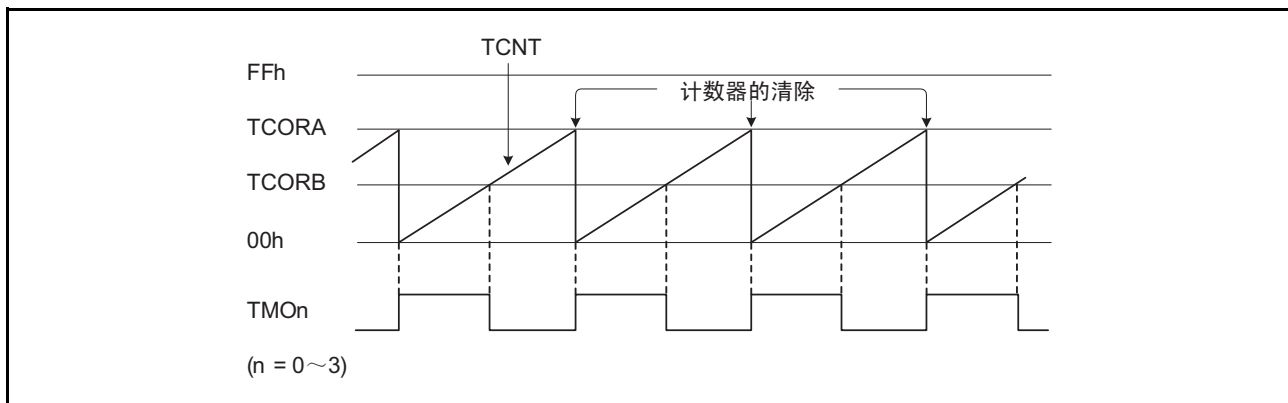


图 17.3 脉冲输出例子

17.3.2 复位输入

TMRIn 输入的任意延迟时间的脉冲输出例子如图 17.4 所示。

1. 为了通过 TMRIn 输入的 High 电平清除 TCNT 计数器，将 TCR.CCLR[1:0] 位置 “11b” (通过外部复位输入进行清除) 并且将 TCCR.TMRIS 位置 “1” (通过外部复位的 High 电平进行清除)。
2. 为了通过 TCORA 寄存器和 TCORB 寄存器的比较匹配分别进行 High 电平输出和 Low 电平输出，将 TCSR.OSA[1:0] 位置 “10b” (High 电平输出) 并且将 TCSR.OSB[1:0] 位置 “01b” (Low 电平输出)。

通过上述设定，能输出延迟 (从 TMRIn 输入开始的延迟) 由 TCORA 寄存器、脉宽由 (TCORB-TCORA) 决定的波形。

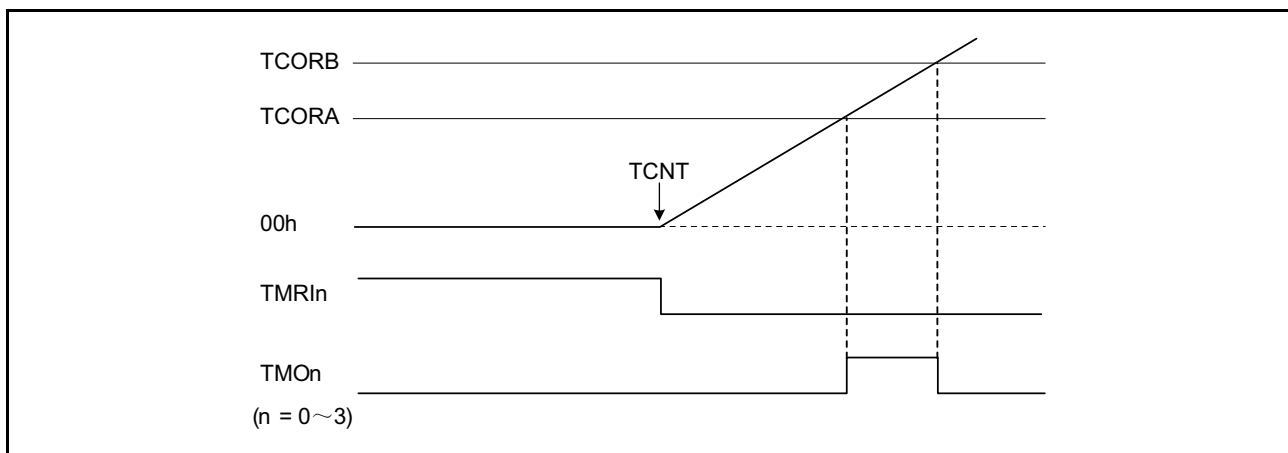


图 17.4 复位输入例子

17.4 运行时序

17.4.1 TCNT 计数器的计数时序

内部时钟和外部时钟运行时的 TCNT 计数器的计数时序分别如图 17.5 和图 17.6 所示。

必须注意：在单边沿的情况下，外部时钟的脉宽至少需要 1.5 个状态；在双边沿的情况下，外部时钟的脉宽至少需要 2.5 个状态。否则，就不能正常运行。

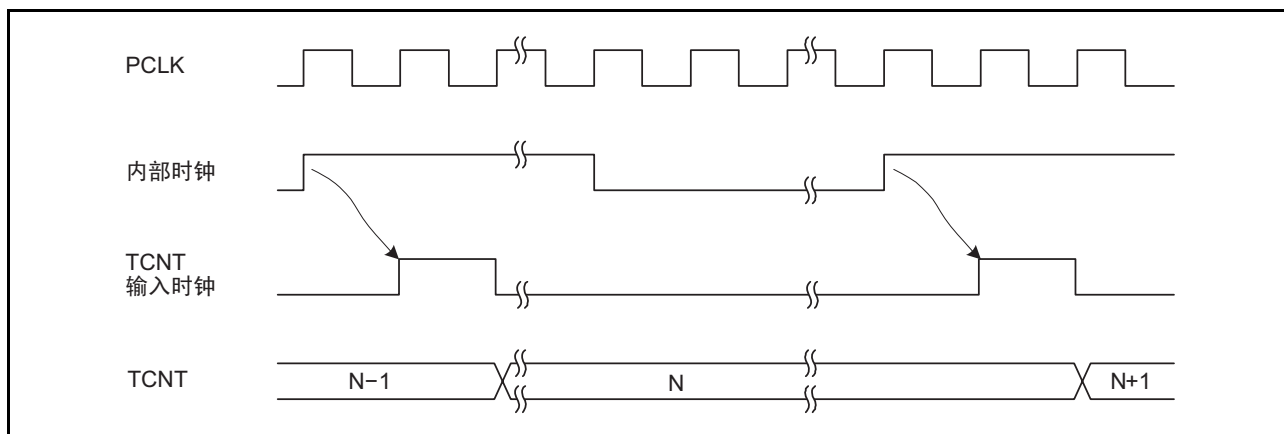


图 17.5 内部时钟运行时的计数时序

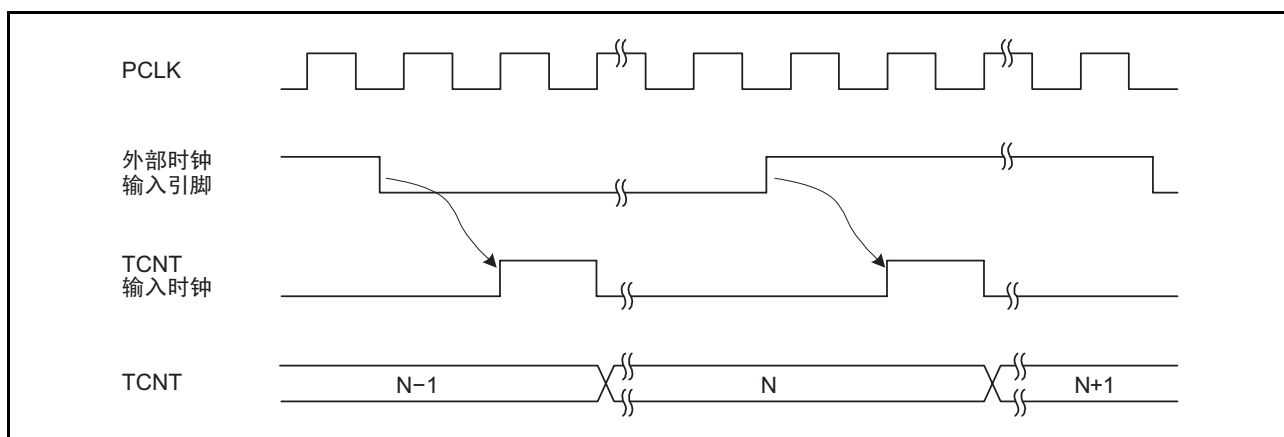


图 17.6 外部时钟运行时的计数时序

17.4.2 比较匹配时的中断标志变为“1”的时序

如果 TCORA 寄存器、TCORB 寄存器和 TCNT 计数器的值相同并且输出比较匹配信号，中断标志就变为“1”。

在 TCORA 寄存器、TCORB 寄存器和 TCNT 计数器的值相同的最后状态（在更新 TCNT 计数器相同后的计数值时）产生比较匹配信号。因此，在 TCNT 计数器和 TCORA 寄存器、TCORB 寄存器的值相同后到产生 TCNT 计数器的输入时钟前，不产生比较匹配信号。

中断标志变为“1”的时序如图 17.7 所示。

对应的中断向量号请参照“10. 中断控制器 (ICU)”和“表 17.6 TMR 的中断源”。

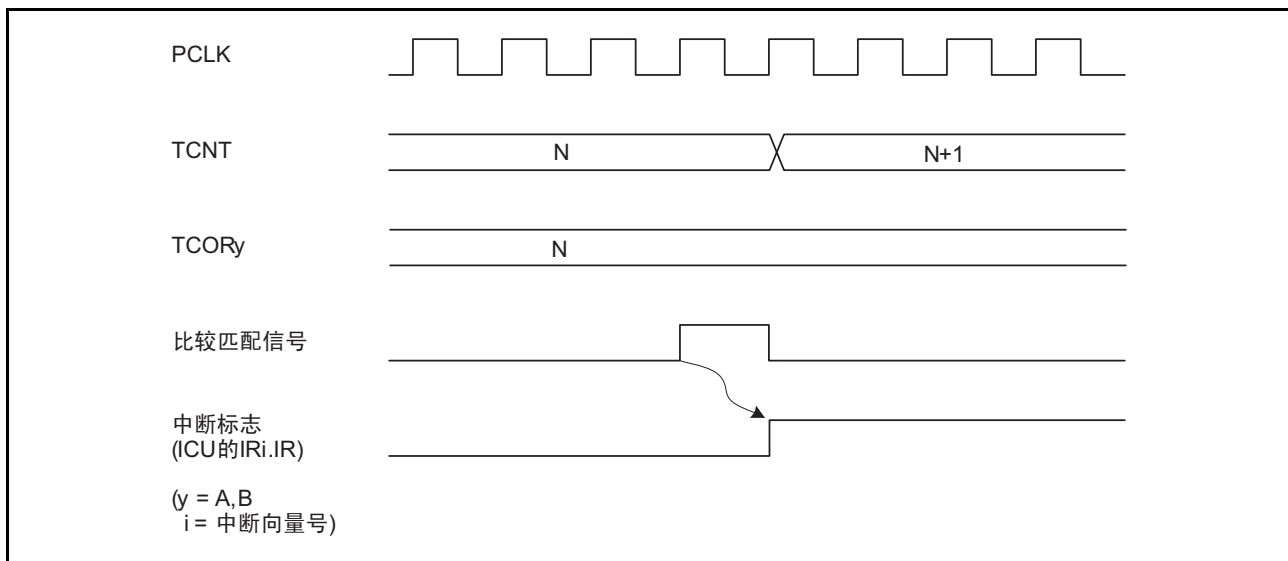


图 17.7 比较匹配时的中断标志变为“1”的时序

17.4.3 比较匹配时的定时器输出时序

在产生比较匹配信号时，将 TCSR.OSA[1:0] 位和 TCSR.OSB[1:0] 位设定的输出值输出到定时器的输出引脚。通过比较匹配 A 信号进行交替输出时的定时器输出时序如图 17.8 所示。

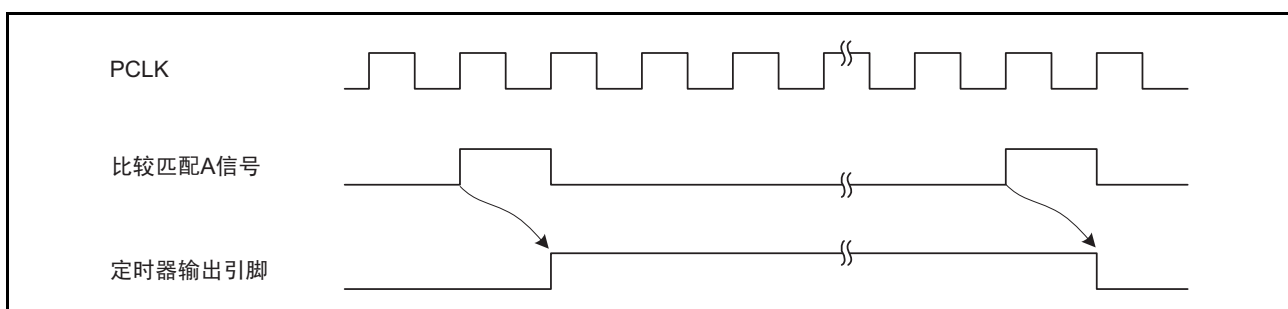


图 17.8 通过比较匹配 A 信号进行的定时器输出时序

17.4.4 通过比较匹配进行的计数器清除时序

根据 TCR.CCLR[1:0] 位的选择, 通过比较匹配 A 或者比较匹配 B 进行 TCNT 计数器的清除。
通过比较匹配进行的计数器清除时序如图 17.9 所示。

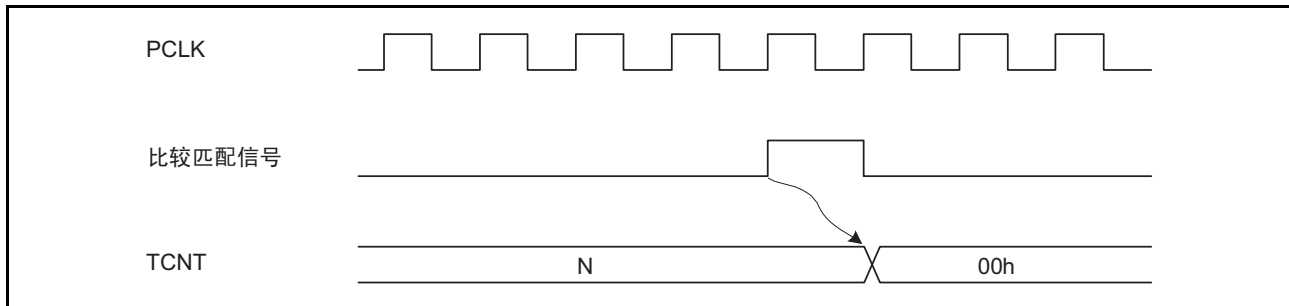


图 17.9 通过比较匹配进行的计数器清除时序

17.4.5 TCNT 计数器的外部复位时序

根据 TCRm.CCLR[1:0] 位的选择, 在外部复位输入的上升沿或者通过 High 电平清除 TCNT 计数器。从输入外部复位到清除 TCNT 计数器前, 至少需要 2 个状态。

通过外部复位输入进行的清除时序如图 17.10 和图 17.11 所示。

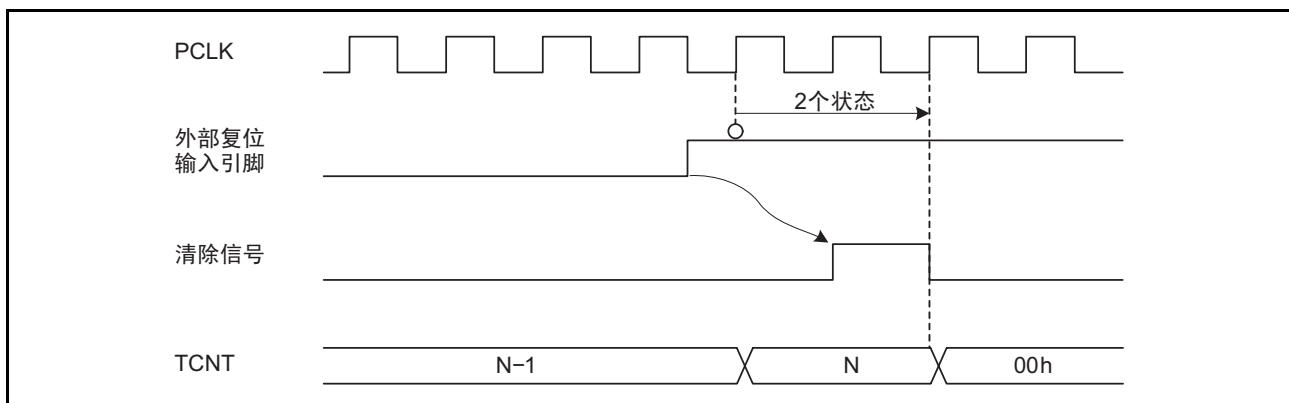


图 17.10 通过外部复位输入进行的清除时序 (上升沿)

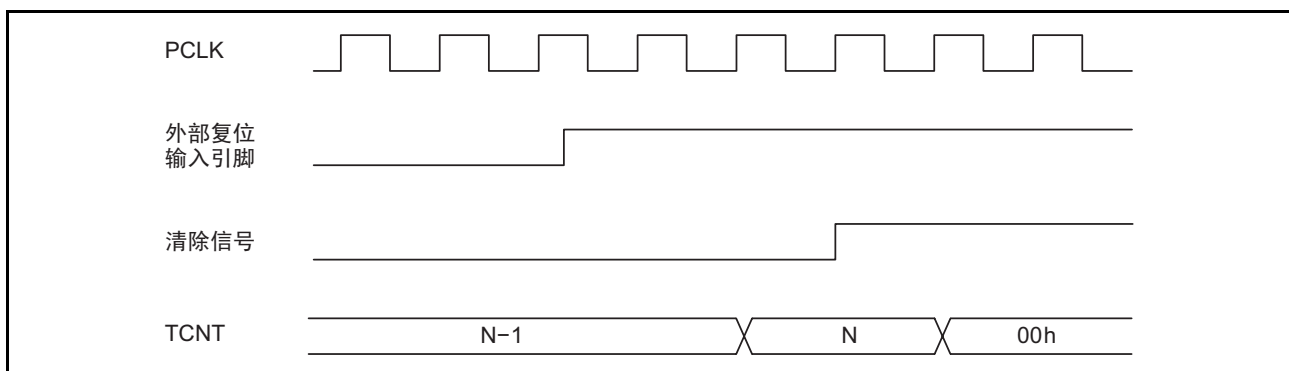


图 17.11 通过外部复位输入进行的清除时序 (High 电平)

17.4.6 通过上溢信号使中断标志变为“1”的时序

通过 TCNT 计数器发生上溢 (“FFh” → “00h”) 时输出的上溢信号使中断标志变为 “1”。

中断标志的置位时序如图 17.12 所示。

对应的中断向量号请参照 “10. 中断控制器 (ICU)” 和 “表 17.6 TMR 的中断源”。

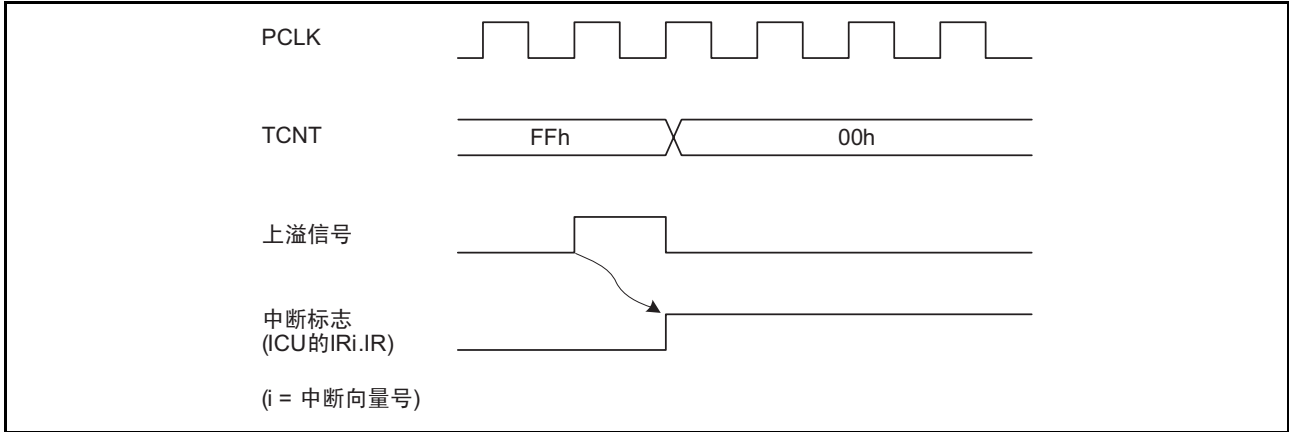


图 17.12 通过上溢信号使中断标志变为“1”的时序

17.5 级联时的运行

如果将 TMR0.TCCR 寄存器或者 TMR1.TCCR 寄存器的 CSS[1:0] 位置 “11b”，就将 2 个通道的 TMR 级联。此时，可设定为 16 位计数模式（用作 1 个 16 位定时器）或者比较匹配计数模式（通过 TMR1 对 TMR0 的比较匹配进行计数）。

在此只说明单元 0，单元 1 级联时的运行和单元 0 相同。

17.5.1 16 位计数模式

当 TMR0.TCCR.CSS[1:0] 位为 “11b” 时，作为 1 个通道的 16 位定时器（TMR0 和 TMR1 分别为高 8 位和低 8 位）运行。只有在此模式中才能对表 17.3 所示的存取长度栏中记载为 8 位或者 16 位的寄存器进行 16 位存取。

(1) 计数器清除的指定

- TMR0.TCR.CCLR[1:0] 位的设定对 16 位计数器有效。如果在通过 TMR0.TCR.CCLR[1:0] 位设定为通过比较匹配进行计数器清除时产生 16 位比较匹配，就清除 16 位计数器（TMR0.TCNT 计数器和 TMR1.TCNT 计数器）。如果设定为通过 TMR10 引脚进行计数器清除，也能清除 16 位计数器（TMR0.TCNT 计数器和 TMR1.TCNT 计数器）。
- TMR1.TCR.CCLR[1:0] 位的设定无效。

(2) 引脚输出

- 根据 16 位的比较匹配条件，通过 TMR0.TCSR.OSA[1:0] 位和 TMR0.TCSR.OSB[1:0] 位控制 TMO0 引脚的输出。
- 根据低 8 位的比较匹配条件，通过 TMR1.TCSR.OSA[1:0] 位和 TMR1.TCSR.OSB[1:0] 位控制 TMO1 引脚的输出。

17.5.2 比较匹配计数模式

当 TMR1.TCCR.CSS[1:0] 位为 “11b” 时，TMR1.TCNT 计数器对 TMR0 比较匹配 A 的发生次数进行计数。分别控制 TMR0 和 TMR1，并且根据各通道的设定，控制中断的发生、TMO_n (n=0,1) 引脚的输出和计数器的清除等。

17.6 中断源

17.6.1 中断源和 DTC 启动

TMRm 的中断源有 CMIAm、CMIBm 和 OVI_m 共 3 种，各中断源和优先级如表 17.6 所示。
能通过 CMIAm 中断或者 CMIBm 中断启动 DTC，而不能通过 TMRm 的中断源启动 DMAC。

表 17.6 TMR 的中断源

| 名称 | 中断源 | 中断状态标志 | DTC 的启动 | 优先级 |
|-------|------------------|----------|---------|--|
| CMIA0 | TMR0.TCORA 的比较匹配 | IR174.IR | 能 | 高  低 |
| CMIB0 | TMR0.TCORB 的比较匹配 | IR175.IR | 能 | |
| OVI0 | TMR0.TCNT 的上溢 | IR176.IR | 不能 | |
| CMIA1 | TMR1.TCORA 的比较匹配 | IR177.IR | 能 | |
| CMIB1 | TMR1.TCORB 的比较匹配 | IR178.IR | 能 | |
| OVI1 | TMR1.TCNT 的上溢 | IR179.IR | 不能 | |
| CMIA2 | TMR2.TCORA 的比较匹配 | IR180.IR | 能 | |
| CMIB2 | TMR2.TCORB 的比较匹配 | IR181.IR | 能 | |
| OVI2 | TMR2.TCNT 的上溢 | IR182.IR | 不能 | |
| CMIA3 | TMR3.TCORA 的比较匹配 | IR183.IR | 能 | |
| CMIB3 | TMR3.TCORB 的比较匹配 | IR184.IR | 能 | |
| OVI3 | TMR3.TCNT 的上溢 | IR185.IR | 不能 | |

有关中断状态标志的详细内容，请参照“10. 中断控制器 (ICU)”。

17.6.2 A/D 转换器的启动

能通过 TMR0 或者 TMR2 的比较匹配 A 启动 A/D 转换器 (注)。

如果在 TMRn.TCSR.ADTE 位为“1” (允许由比较匹配 A 引起的 A/D 转换开始请求) 的状态下产生比较匹配 A，就向 A/D 转换器请求开始 A/D 转换。此时，如果在 A/D 转换器侧选择了 8 位定时器的转换触发，就开始 A/D 转换。

有关对应的 A/D 转换器的单元，请参照“23. A/D 转换器”。

17.7 使用时的注意事项

17.7.1 模块停止功能的设定

能通过模块停止控制寄存器设定为禁止或者允许 TMR 的运行，初始值为禁止 TMR 的运行。能通过解除模块停止状态，使寄存器变为可存取的状态。详细内容请参照“8. 低功耗功能”。

17.7.2 设定周期时的注意事项

如果设定为通过比较匹配进行计数器清除，就在 TCNT 计数器和 TCORA 寄存器、TCORB 寄存器的值相同的最后状态（在更新 TCNT 计数器相同后的计数值时）清除 TCNT 计数器。因此，计数器的频率用以下的表达式表示（f: 计数器的频率， ϕ : 工作频率，N: TCORA 寄存器或者 TCORB 的设定值）。

$$f = \phi / (N + 1)$$

17.7.3 TCNT 计数器的写和计数器清除的竞争

如果在通过 CPU 写 TCNT 计数器的同时产生计数器清除信号，就不写计数器而优先清除计数器。此时序如图 17.13 所示，

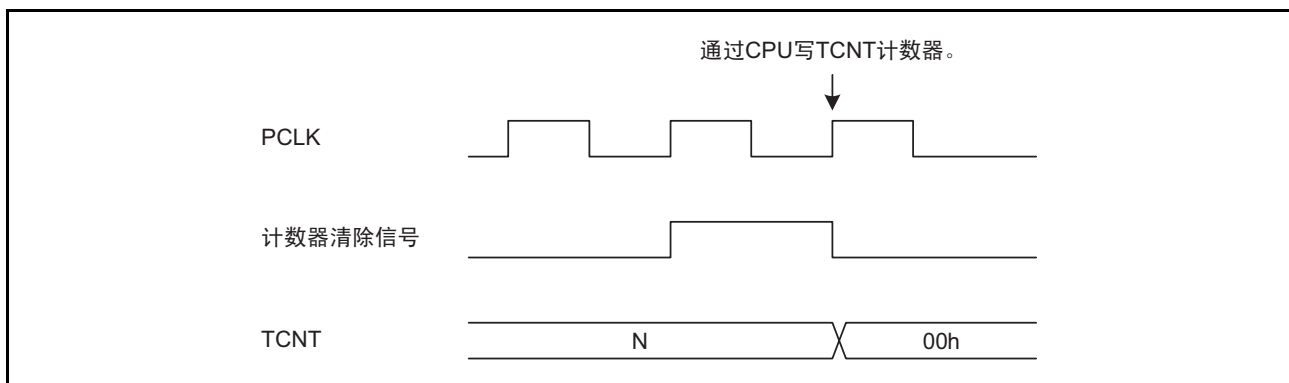


图 17.13 TCNT 计数器的写和计数器清除的竞争

17.7.4 TCNT 计数器的写和递增计数的竞争

即使在通过 CPU 写 TCNT 计数器的同时发生递增计数，也不进行递增计数而优先写 TCNT 计数器。此时序如图 17.14 所示。

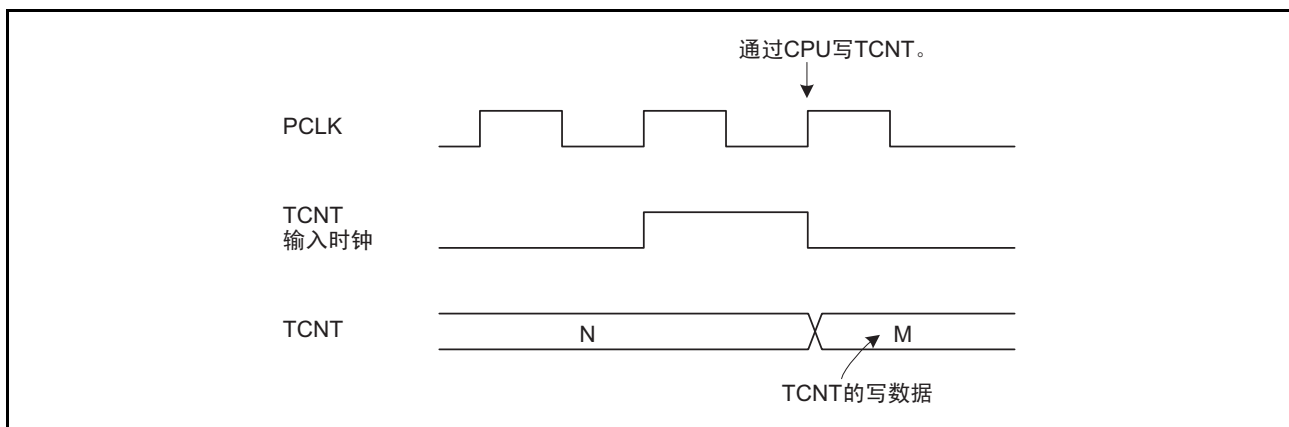


图 17.14 TCNT 计数器的写和递增计数的竞争

17.7.5 TCORA 寄存器、TCORB 寄存器的写和比较匹配的竞争

如图 17.15 所示，即使在通过 CPU 写 TCORA 寄存器或者 TCORB 寄存器的同时产生比较匹配信号，也优先写 TCORA 寄存器或者 TCORB 寄存器，而比较匹配信号不变为 High 电平。

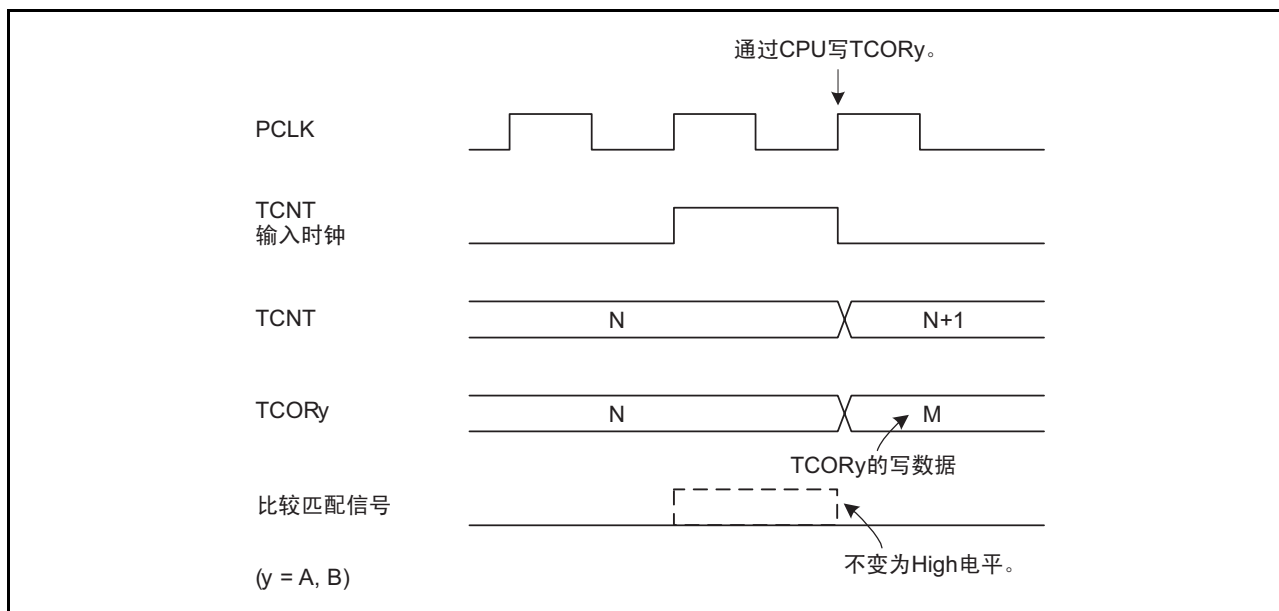


图 17.15 TCORA 寄存器、TCORB 寄存器的写和比较匹配的竞争

17.7.6 比较匹配 A 和比较匹配 B 的竞争

如果同时产生比较匹配 A 和比较匹配 B，就从比较匹配 A 和比较匹配 B 的输出设定中，进行如表 17.7 所示的优先级高的定时器输出。

表 17.7 定时器输出的优先级

| 输出的设定 | 优先级 |
|-----------|-------------|
| 交替输出 | 高 ↑ 低 |
| High 电平输出 | |
| Low 电平输出 | |
| 不变 | |

17.7.7 内部时钟的转换和 TCNT 计数器的运行

在进行内部时钟的转换时，TCNT 计数器有可能进行了递增计数。内部时钟的转换时序 (TCCR.CKS[2:0] 位的改写) 和 TCNT 计数器运行的关系如表 17.8 所示。

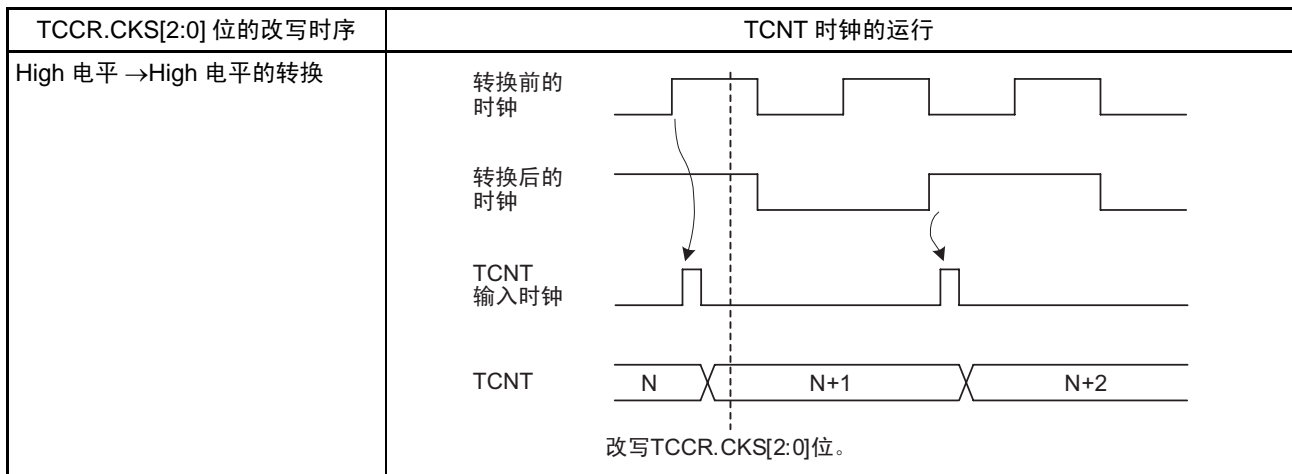
在内部时钟生成 TCNT 计数器的时钟的情况下，检测内部时钟的上升沿。因此，如表 17.8 的 No.2 所示，如果进行 Low 电平 → High 电平的时钟转换，就将转换时序视为边沿，产生 TCNT 计数器时钟并且 TCNT 计数器进行了递增计数。

在进行内部时钟和外部时钟的转换时，TCNT 计数器也有可能进行递增计数。

表 17.8 内部时钟的转换和 TCNT 计数器的运行 (1/2)

| TCCR.CKS[2:0] 位的改写时序 | TCNT 时钟的运行 |
|--------------------------|------------|
| Low 电平 → Low 电平的转换 (注1) | |
| Low 电平 → High 电平的转换 (注2) | |
| High 电平 → Low 电平的转换 (注4) | |

表 17.8 内部时钟的转换和 TCNT 计数器的运行 (2/2)



注 1. 包括 Low 电平 → 停止和停止 → Low 电平的转换。

注 2. 包括停止 → High 电平的转换。

注 3. 因为将转换时序视为边沿，所以产生了 TCNT 输入时钟，并且 TCNT 进行了递增计数。

注 4. 包括 High 电平 → 停止的转换。

17.7.8 级联时的时钟源设定

如果同时设定 16 位计数模式和比较匹配计数模式，因为不产生 TMR0.TCNT 计数器或者 TMR1.TCNT 计数器 (TMR2.TCNT 计数器和 TMR3.TCNT 计数器) 的输入时钟，所以计数器停止后不再运行。不能进行此设定。

18. 比较匹配定时器 (CMT)

RX610 群内置由 2 个通道的 16 位定时器构成的比较匹配定时器 (CMT)，有 2 个单元 (单元 0 和单元 1)，共计 4 个通道。CMT 有 16 位计数器，能按设定的各周期产生中断。

18.1 概要

CMT 的规格如表 18.1 所示。

CMT (单元 0) 的框图如图 18.1 所示。2 个通道的 CMT 构成 1 个单元，单元 0 和单元 1 的规格相同。

表 18.1 CMT 的规格

| 项目 | 功能 |
|-------|--|
| 计数时钟 | <ul style="list-style-type: none"> • 4 种内部时钟 各通道可分别从 PCLK/8、PCLK/32、PCLK/128、PCLK/512 中选择。 |
| 中断 | 能分别向各通道请求比较匹配中断。 |
| 低功耗功能 | 各单元能设定为模块停止状态。 |

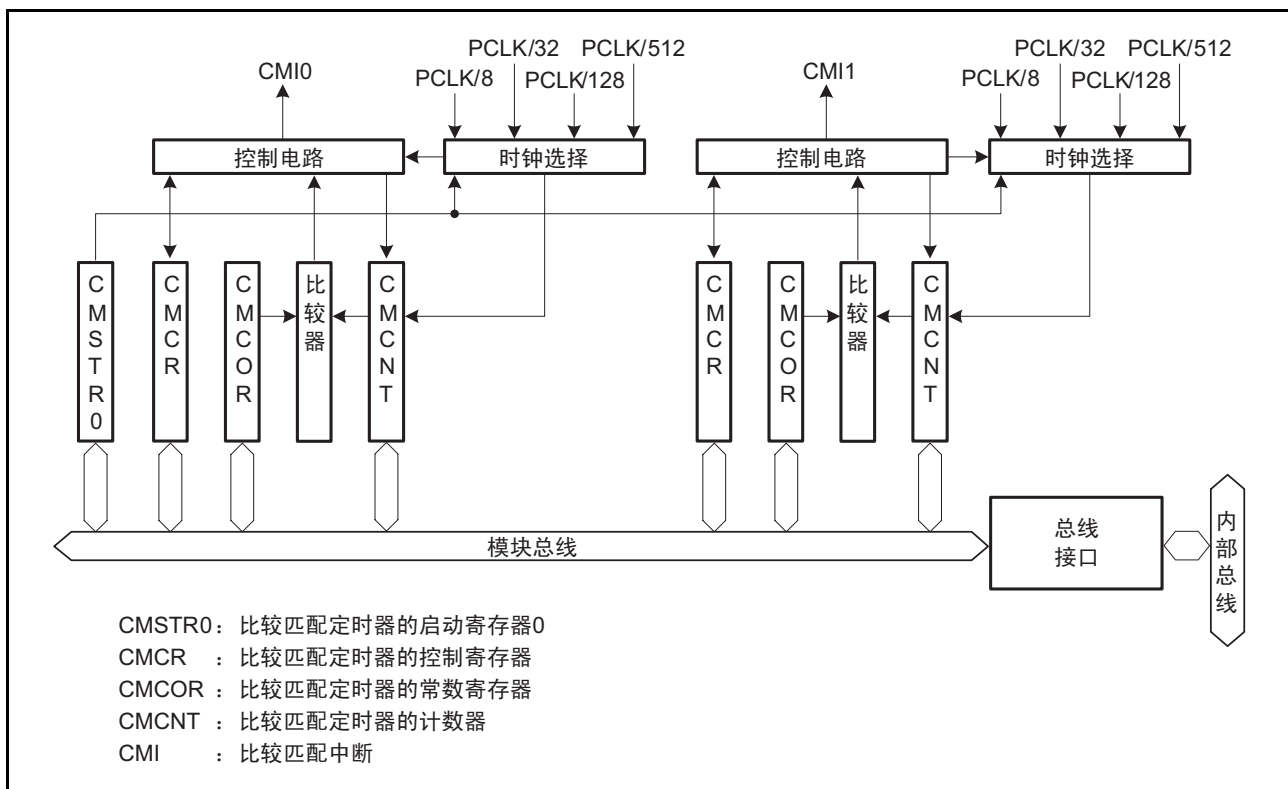


图 18.1 CMT (单元 0) 的框图

18.2 寄存器说明

CMT 的寄存器一览表如表 18.2 所示。

表 18.2 CMT 的寄存器一览表

| 单元 | 通道 | 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|---------------|------|-----------------|--------|------------|------------|------|
| 单元 0 | 通用 | 比较匹配定时器的启动寄存器 0 | CMSTR0 | 0000h | 0008 8000h | 16 |
| | CMT0 | 比较匹配定时器的控制寄存器 | CMCR | 00x0h | 0008 8002h | 16 |
| | | 比较匹配定时器的计数器 | CMCNT | 0000h | 0008 8004h | 16 |
| | | 比较匹配定时器的常数寄存器 | CMCOR | FFFFh | 0008 8006h | 16 |
| | CMT1 | 比较匹配定时器的控制寄存器 | CMCR | 00x0h | 0008 8008h | 16 |
| | | 比较匹配定时器的计数器 | CMCNT | 0000h | 0008 800Ah | 16 |
| 比较匹配定时器的常数寄存器 | | CMCOR | FFFFh | 0008 800Ch | 16 | |
| 单元 1 | 通用 | 比较匹配定时器的启动寄存器 1 | CMSTR1 | 0000h | 0008 8010h | 16 |
| | CMT2 | 比较匹配定时器的控制寄存器 | CMCR | 00x0h | 0008 8012h | 16 |
| | | 比较匹配定时器的计数器 | CMCNT | 0000h | 0008 8014h | 16 |
| | | 比较匹配定时器的常数寄存器 | CMCOR | FFFFh | 0008 8016h | 16 |
| | CMT3 | 比较匹配定时器的控制寄存器 | CMCR | 00x0h | 0008 8018h | 16 |
| | | 比较匹配定时器的计数器 | CMCNT | 0000h | 0008 801Ah | 16 |
| 比较匹配定时器的常数寄存器 | | CMCOR | FFFFh | 0008 801Ch | 16 | |

18.2.1 比较匹配定时器的启动寄存器 0 (CMSTR0)

地址 0008 8000h

| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|-------|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|------|------|
| 复位后的值 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | STR1 | STR0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|--------|------|----------|--|-----|
| b0 | STR0 | 计数开始 0 位 | 0: CMT0.CMCNT 计数器停止计数 1: CMT0.CMCNT 计数器开始计数 | R/W |
| b1 | STR1 | 计数开始 1 位 | 0: CMT1.CMCNT 计数器停止计数 1: CMT1.CMCNT 计数器开始计数 | R/W |
| b15-b2 | — | 保留位 | 读写值都为“0”。 | R/W |

CMSTR0 寄存器设定开始或者停止 CMT0.CMCNT 计数器和 CMT1.CMCNT 计数器的计数。

STR0 位 (计数开始 0 位)

此位选择开始或者停止 CMT0.CMCNT 计数器的计数。

STR1 位 (计数开始 1 位)

此位选择开始或者停止 CMT1.CMCNT 计数器的计数。

18.2.2 比较匹配定时器的启动寄存器 1 (CMSTR1)

地址 0008 8010h

| | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|------|------|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | — | — | — | — | — | — | — | — | — | — | STR3 | STR2 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|--------|------|----------|--|-----|
| b0 | STR2 | 计数开始 2 位 | 0: CMT2.CMCNT 计数器停止计数 1: CMT2.CMCNT 计数器开始计数 | R/W |
| b1 | STR3 | 计数开始 3 位 | 0: CMT3.CMCNT 计数器停止计数 1: CMT3.CMCNT 计数器开始计数 | R/W |
| b15-b2 | — | 保留位 | 读写值都为“0”。 | R/W |

CMSTR1 寄存器设定开始或者停止 CMT2.CMCNT 计数器和 CMT3.CMCNT 计数器的计数。

STR2 位 (计数开始 2 位)

此位选择开始或者停止 CMT2.CMCNT 计数器的计数。

STR3 位 (计数开始 3 位)

此位选择开始或者停止 CMT3.CMCNT 计数器的计数。

18.2.3 比较匹配定时器的控制寄存器 (CMCR)

地址 CMT0.CMCR 0008 8002h、CMT1.CMCR 0008 8008h、
CMT2.CMCR 0008 8012h、CMT3.CMCR 0008 8018h

| | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|----|----|----|------|----|----|----|----|----------|----|
| b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | — | — | — | — | — | — | CMIE | — | — | — | — | CKS[1:0] | |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

x: 不定值

| 位 | 符号 | 位名 | 功能 | R/W |
|--------|----------|-----------|--|-----|
| b1-b0 | CKS[1:0] | 时钟选择位 | b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512 | R/W |
| b5-b2 | — | 保留位 | 读写值都为“0”。 | R/W |
| b6 | CMIE | 比较匹配中断允许位 | 0: 禁止比较匹配中断 (CMIIm) 1: 允许比较匹配中断 (CMIIm) | R/W |
| b7 | — | 保留位 | 读取值为不定值, 只能写“1”。 | R/W |
| b15-b8 | — | 保留位 | 读写值都为“0”。 | R/W |

CMCR 寄存器是设定递增计数时钟的寄存器。

CKS[1:0] 位 (时钟选择位)

这些位从外围模块时钟 (PCLK) 分频后得到的 4 种内部时钟中选择 CMCNT 计数器的输入时钟。

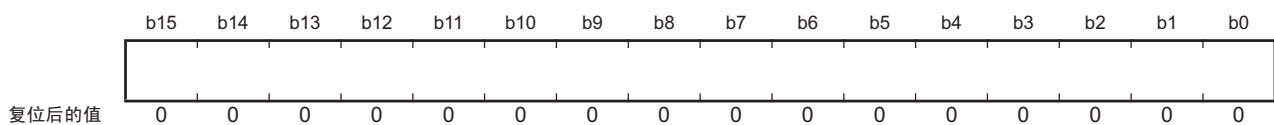
如果将 CMSTRy.STRj 位 (y=0,1, j=0 ~ 3) 置 “1”，对应的 CMCNT 计数器就通过 CKS[1:0] 位选择的时钟开始递增计数。

CMIE 位 (比较匹配中断允许位)

在 CMCNT 和 CMCOR 的值相同时，此位选择允许或者禁止比较匹配中断 (CMIm) (m=0 ~ 3) 的发生。

18.2.4 比较匹配定时器的计数器 (CMCNT)

地址 CMT0.CMCNT 0008 8004h、CMT1.CMCNT 0008 800Ah、
CMT2.CMCNT 0008 8014h、CMT3.CMCNT 0008 801Ah



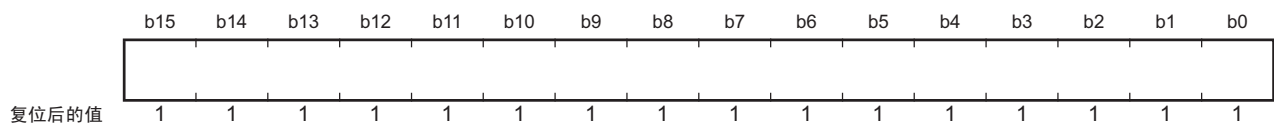
CMCNT 计数器是用于产生中断请求的可读写递增计数器。

如果通过 CMCR.CKS[1:0] 位选择内部时钟并且将 CMSTRy.STRj 位 (y=0,1, j=0 ~ 3) 置 “1”，CMCNT 计数器就通过该时钟开始递增计数。

如果 CMCNT 计数器的值和 CMCOR 寄存器的值相同，CMCNT 计数器就变为 “0000h”，并且产生比较匹配中断 (CMIn) (n=0 ~ 3)。

18.2.5 比较匹配定时器的常数寄存器 (CMCOR)

地址 CMT0.CMCOR 0008 8006h、CMT1.CMCOR 0008 800Ch、
CMT2.CMCOR 0008 8016h、CMT3.CMCOR 0008 801Ch



CMCOR 寄存器是设定和 CMCNT 计数器的比较匹配周期的寄存器。

18.3 运行说明

18.3.1 周期计数

如果通过 CMCY.CKS[1:0] 位选择内部时钟并且将 CMSTRy.STRj 位 (y=0,1, j=0 ~ 3) 置“1”，CMCNT 计数器就通过所选的时钟开始递增计数。

如果 CMCNT 计数器的值和 CMCOR 寄存器的值相同，CMCNT 计数器就变为“0000h”，并且产生比较匹配中断 (CMIm) (m=0 ~ 3)。CMCNT 计数器从“0000h”重新开始递增计数，CMCNT 计数器的计数如图 18.2 所示。

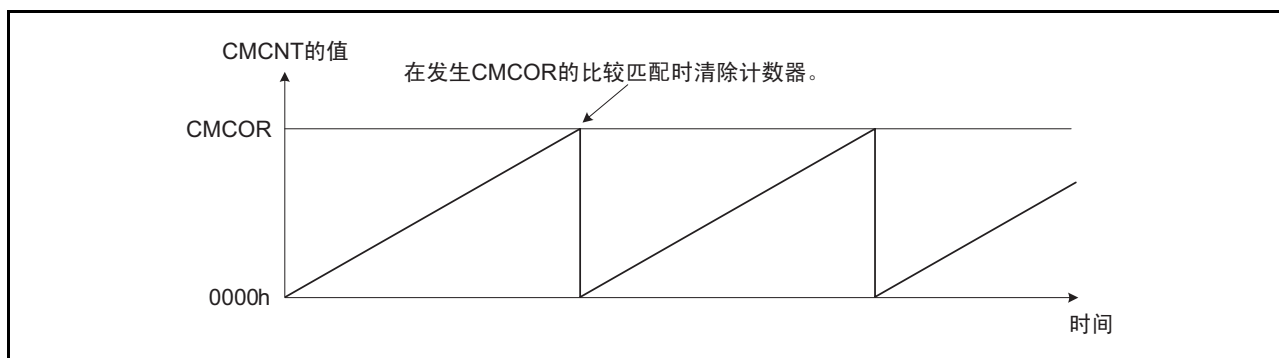


图 18.2 CMCNT 计数器的计数

18.3.2 CMCNT 计数器的计数时序

能通过 CMCY.CKS[1:0] 位选择外围模块时钟 (PCLK) 分频后的 4 种内部时钟 (PCLK/8、PCLK/32、PCLK/128、PCLK/512)，此时的 CMCNT 计数器的计数时序如图 18.3 所示。

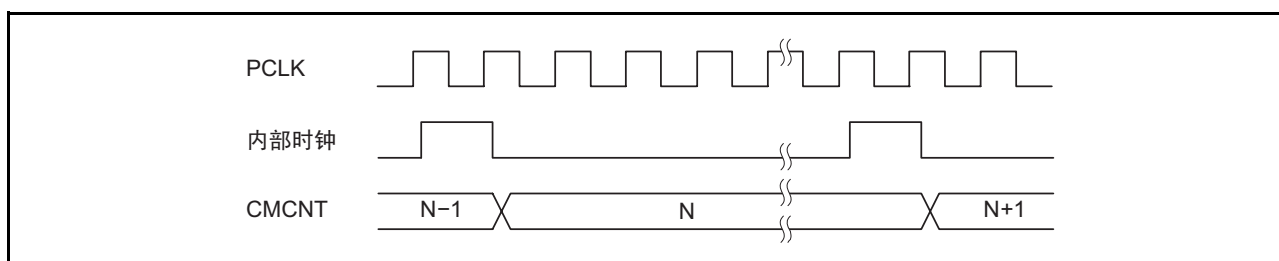


图 18.3 CMCNT 计数器的计数时序

18.4 中断

18.4.1 中断源

CMT 的各通道有比较匹配中断 (CMI_m) (m=0 ~ 3)，各中断分配有向量地址。如果发生比较匹配中断，就输出对应的中断请求。

在通过中断请求启动 CPU 中断时，能通过设定中断控制器更改通道之间的优先级，详细内容请参照“10. 中断控制器 (ICU)”。

表 18.3 CMT 的中断源

| 名称 | 中断源 | 中断状态标志 | DTC 的启动 | DMAC 的启动 |
|------|-------------------------------|----------|---------|----------|
| CMI0 | CMT0.CMCNT 和 CMT0.CMCOR 的比较匹配 | IR028.IR | 能 | 能 |
| CMI1 | CMT1.CMCNT 和 CMT1.CMCOR 的比较匹配 | IR029.IR | 能 | 能 |
| CMI2 | CMT2.CMCNT 和 CMT2.CMCOR 的比较匹配 | IR030.IR | 能 | 能 |
| CMI3 | CMT3.CMCNT 和 CMT3.CMCOR 的比较匹配 | IR031.IR | 能 | 能 |

18.4.2 比较匹配中断的发生时序

在 CMCNT 计数器和 CMCOR 寄存器的值相同时，发生比较匹配中断 (CMI_m) (m=0 ~ 3)。

在 CMCNT 计数器和 CMCOR 寄存器的值相同后的最后状态 (在更新 CMCNT 计数器相同后的计数值时) 产生比较匹配信号。因此，在从这两者的值相同后到产生 CMCNT 计数器的输入时钟前，不产生比较匹配信号。

中断标志变为“1”的时序如图 18.4 所示。

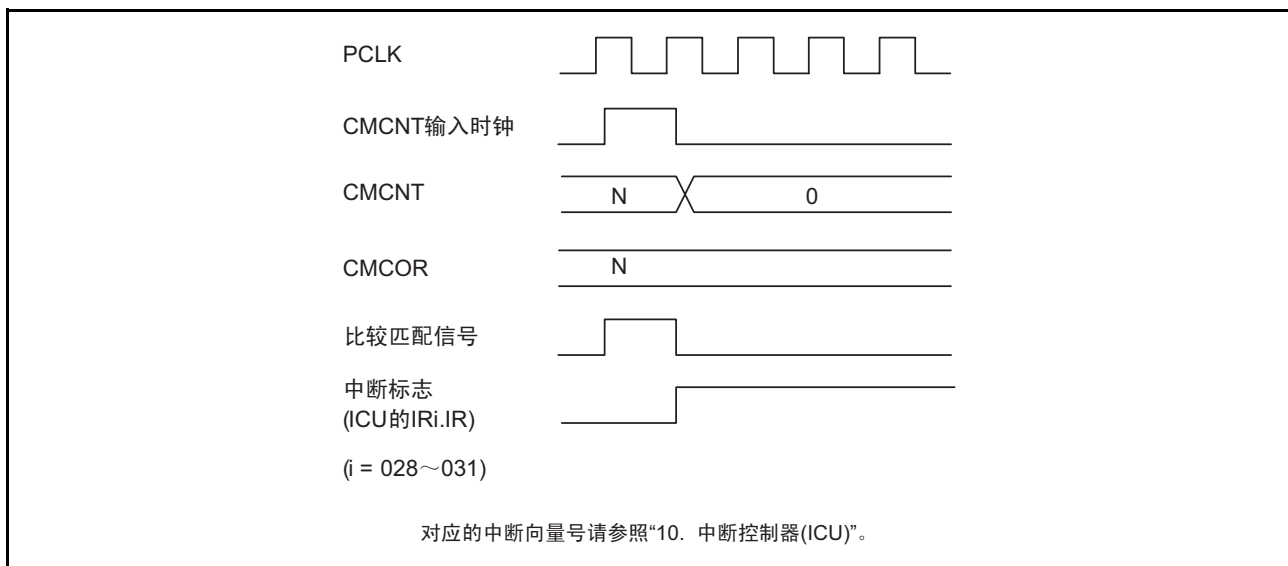


图 18.4 比较匹配中断标志变为“1”的时序

18.5 使用时的注意事项

18.5.1 模块停止功能的设定

能通过模块停止控制寄存器设定禁止或者允许 CMT 的计数，初始值为停止 CMT 的计数。能通过解除模块停止状态，使寄存器变为可存取的状态。详细内容请参照“8. 低功耗功能”。

18.5.2 CMCNT 计数器的写和比较匹配的竞争

如果在写 CMCNT 计数器的过程中产生比较匹配信号，就不写 CMCNT 计数器而优先清除 CMCNT 计数器。此时序如图 18.5 所示。

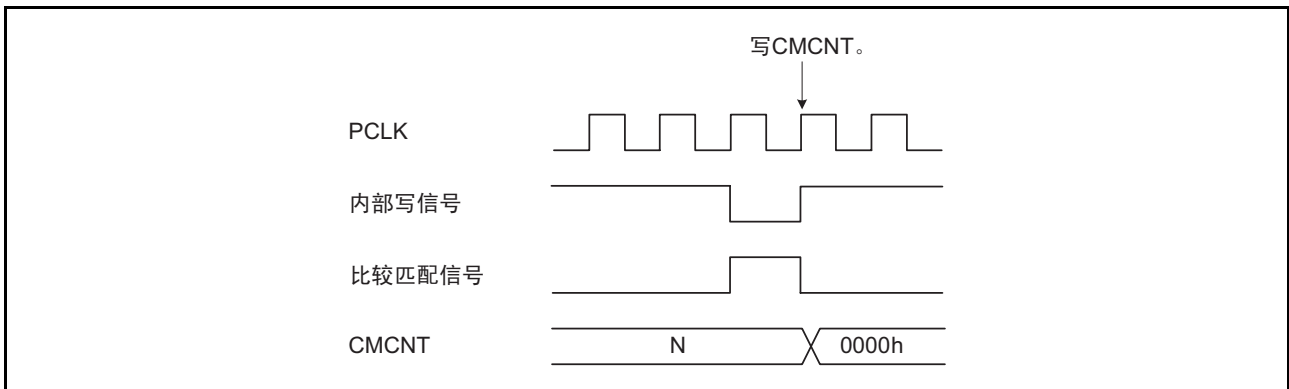


图 18.5 CMCNT 计数器的写和比较匹配的竞争

18.5.3 CMCNT 计数器的写和递增计数的竞争

即使在写 CMCNT 计数器的过程中发生递增计数，CMCNT 计数器也不进行递增计数而优先写 CMCNT 计数器。此时序如图 18.6 所示。

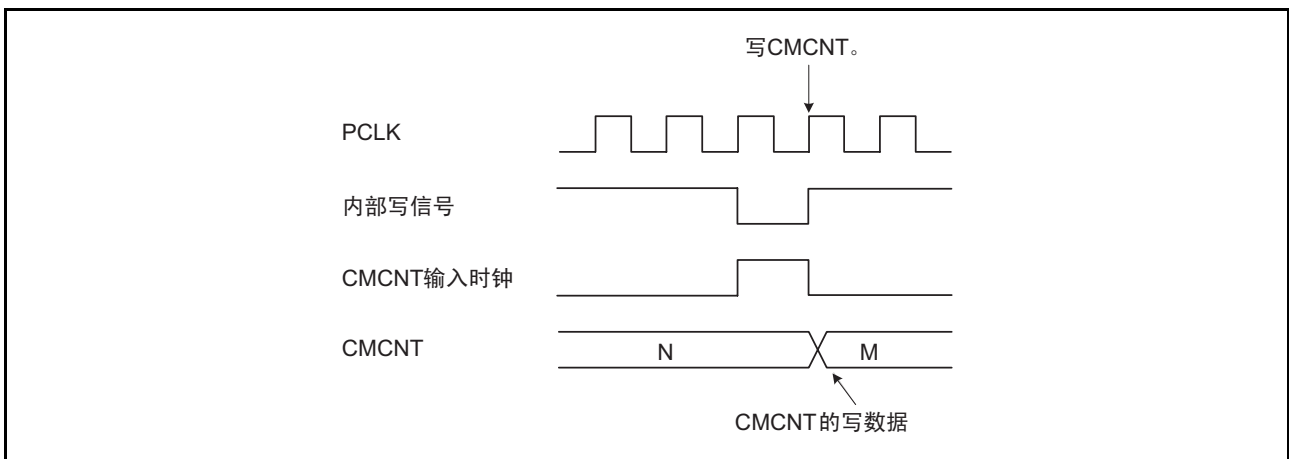


图 18.6 CMCNT 计数器的写和递增计数的竞争

19. 看门狗定时器 (WDT)

看门狗定时器 (WDT) 是 8 位定时器, 如果因系统失控等而不改写计数器的值导致发生上溢, 就将上溢信号 (WDTOVF#) 输出到外部, 同时能对 LSI 内部进行复位。

在不用作看门狗定时器时, 也能用作间隔定时器。在用作间隔定时器的情况下, 每当计数器发生上溢时, 就产生间隔定时器中断。

19.1 概要

WDT 的规格和框图分别如表 19.1 和图 19.1 所示。

表 19.1 WDT 的规格

| 项目 | 内容 |
|----------|---|
| 计数时钟 | PCLK/4、PCLK/64、PCLK/128、PCLK/512、PCLK/2048、PCLK/8192、PCLK/32768、PCLK/131072 |
| 通道数 | 8 位 × 1 个通道 |
| 计数器清除 | 写 TCNT。 |
| 运行模式 | 看门狗定时器模式和间隔定时器模式的转换 |
| 看门狗定时器模式 | 如果计数器发生上溢, 就将 WDTOVF# 信号输出到外部, 并且能选择是否同时对 LSI 内部进行复位。 |
| 间隔定时器模式 | 如果计数器发生上溢, 就产生间隔定时器中断 (WOVI)。 |

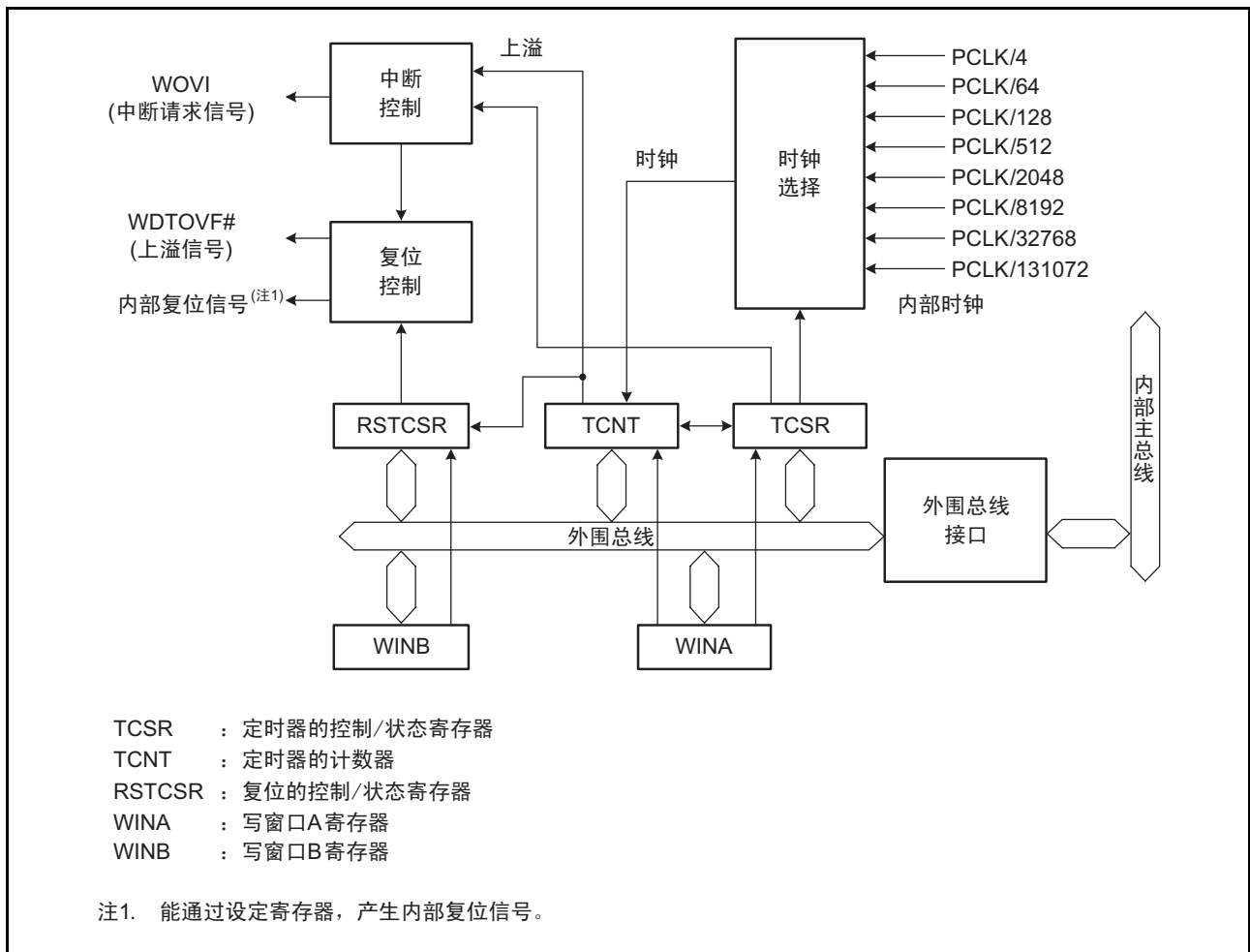


图 19.1 WDT 的框图

WDT 使用的输入 / 输出引脚如表 19.2 所示。

表 19.2 WDT 的输入 / 输出引脚

| 引脚名 | 输入 / 输出 | 功能 |
|---------|---------|---------------------|
| WDTOVF# | 输出 | 输出看门狗定时器模式的计数器上溢信号。 |

19.2 寄存器说明

WDT 的寄存器一览表如表 19.3 所示。

表 19.3 WDT 的寄存器一览表

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|----------------|--------|-------|------------------|------|
| 定时器的控制 / 状态寄存器 | TCSR | x8h | 0008 8028h (注 1) | 8 |
| 定时器的计数器 | TCNT | 00h | 0008 8029h (注 1) | 8 |
| 复位的控制 / 状态寄存器 | RSTCSR | 1Fh | 0008 802Bh (注 1) | 8 |
| 写窗口 A 寄存器 | WINA | — | 0008 8028h (注 2) | 16 |
| 写窗口 B 寄存器 | WINB | — | 0008 802Ah (注 2) | 16 |

注 1. 这是只读寄存器。

注 2. 这是只写寄存器。

19.2.1 定时器的计数器 (TCNT)

地址 0008 8029h



TCNT 计数器是对内部时钟进行计数的 8 位递增计数器。

如果将 TCSR.TME 位置 “0”，CNT 计数器就被初始化为 “00h”。

必须以 8 位为单位读 TCNT。

写 TCNT 寄存器时，必须以 16 位为单位写 WINA 寄存器。

详细内容请参照 “19.5.1 存取寄存器时的注意事项”。

19.2.2 定时器的控制 / 状态寄存器 (TCSR)

地址 0008 8028h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|----------|---|-----|
| b2-b0 | CKS[2:0] | 时钟选择位 | b2 b0 0 0 0: PCLK/4 (周期为 20.4μs) 0 0 1: PCLK/64 (周期为 326.4μs) 0 1 0: PCLK/128 (周期为 652.8μs) 0 1 1: PCLK/512 (周期为 2.6ms) 1 0 0: PCLK/2048 (周期为 10.4ms) 1 0 1: PCLK/8192 (周期为 41.8ms) 1 1 0: PCLK/32768 (周期为 167.1ms) 1 1 1: PCLK/131072 (周期为 668.5ms) 注 1. () 内表示 PCLK 为 50MHz 时的上溢周期。 | R/W |
| b4-b3 | — | 保留位 | 读写值都为“1”。 | R/W |
| b5 | TME | 定时器允许位 | 0: TCNT 计数器停止计数并且被初始化为“00h” 1: TCNT 计数器开始计数 | R/W |
| b6 | TMS | 定时器模式选择位 | 0: 间隔定时器模式 在 TCNT 计数器发生上溢时, 请求间隔定时器中断 (WOVI)。 1: 看门狗定时器模式 在 TCNT 计数器发生上溢时, 将 WDTOVF# 输出到外部。 | R/W |
| b7 | — | 保留位 | 读取值为不定值, 只能写“1”。 | R/W |

TCSR 寄存器是选择 TCNT 计数器的输入时钟和模式的寄存器。

必须以 8 位为单位读 TCSR。

写 TCSR 寄存器时, 必须以 16 位为单位写 WINA 寄存器。

详细内容请参照“19.5.1 存取寄存器时的注意事项”。

CKS[2:0] 位 (时钟选择位)

这些位选择 TCNT 计数器的输入时钟。

TME 位 (定时器允许位)

此位选择开始或者停止 TCNT 计数器的计数。

如果将此位置“1”, TCNT 计数器就开始计数; 如果置“0”, TCNT 计数器就停止计数并且被初始化为“00h”。

TMS 位 (定时器模式选择位)

此位选择是用作看门狗定时器还是用作间隔定时器。

19.2.3 复位的控制 / 状态寄存器 (RSTCSR)

地址 0008 802Bh

| | | | | | | | |
|------|------|----|----|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| WOVF | RSTE | — | — | — | — | — | — |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|------------|--|---------------|
| b4-b0 | — | 保留位 | 读写值都为“1”。 | R/W |
| b5 | — | 保留位 | 读写值都为“0”。 | R/W |
| b6 | RSTE | 复位允许位 | 0: 在看门狗定时器模式中, 即使 TCNT 计数器发生上溢, LSI 内部也不被复位 (WDT 的 TCNT 计数器和 TCSR 寄存器被复位)。 1: 在看门狗定时器模式中, 当 TCNT 计数器发生上溢时, LSI 内部被复位。 | R/W |
| b7 | WOVF | 看门狗定时器上溢标志 | 0: 在看门狗定时器模式中, TCNT 计数器不发生上溢。 1: 在看门狗定时器模式中, TCNT 计数器发生上溢。 | R(W) (注 1) |

注 1. 只能写“0”。

RSTCSR 寄存器控制因 TCNT 计数器的上溢而产生的内部复位信号以及选择内部复位信号的种类。

通过 RES# 引脚的复位信号和深度软件待机复位将 RSTCSR 寄存器初始化为“1Fh”，而不通过因看门狗定时器的上溢而产生的内部复位信号进行初始化。

必须以 8 位为单位读 RSTCSR 寄存器。

写 RSTCSR 寄存器时, 必须以 16 位为单位写 WINB 寄存器。

详细内容请“19.5.1 存取寄存器时的注意事项”。

RSTE 位 (复位允许位)

在看门狗定时器模式中, 此位选择是否通过 TCNT 计数器的上溢对 LSI 内部进行复位。

WOVF 标志 (看门狗定时器上溢标志)

在看门狗定时器模式中, 此标志表示 TCNT 计数器发生了上溢; 在间隔定时器模式中, 此标志不为“1”。

[为“1”的条件]

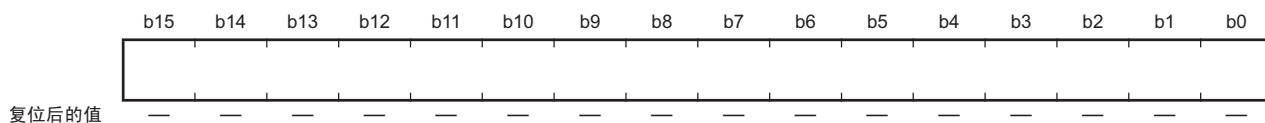
- 在看门狗定时器模式中, TCNT 计数器发生上溢 (“FFh” → “00h”) 时

[为“0”的条件]

- 读“1”后写“0”时

19.2.4 写窗口 A 寄存器 (WINA)

地址 0008 8028h



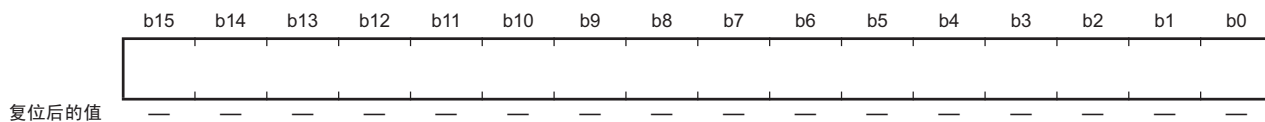
WINA 寄存器是改写 TCNT 计数器值和 TCSR 寄存器值的只写寄存器。

TCNT 计数器和 TCSR 寄存器的写法不同，详细内容请参照“19.5.1 存取寄存器时的注意事项”。

必须以 16 位为单位写 WINA 寄存器。

19.2.5 写窗口 B 寄存器 (WINB)

地址 0008 802Ah



WINB 寄存器是改写 RSTCSR 寄存器值的只写寄存器。

给 RSTCSR.WOVF 标志写“0”的方法不同于 RSTCSR.RSTE 位的写法，详细内容请参照“19.5.1 存取寄存器时的注意事项”。

必须以 16 位为单位写 WINB 寄存器。

19.3 运行说明

19.3.1 看门狗定时器模式

在用作看门狗定时器模式时，必须将 TCSR.TMS 位置“1”（看门狗定时器模式）并且将 TCSR.TME 位置“1”（TCNT 计数器开始计数）。

如果在用作看门狗定时器时因系统失控等而不改写 TCNT 计数器的值导致发生上溢，就输出 WDTOVF# 信号。在系统正常运行期间，TCNT 计数器不发生上溢。为了避免上溢的产生，必须在 TCNT 计数器发生上溢前改写 TCNT 计数器的值（通常写“00h”）。而且，能在看门狗定时器模式中对 LSI 内部进行复位。

如果将 RSTCSR.RSTE 位置“1”，就在 TCNT 计数器发生上溢时输出 WDTOVF# 信号，同时产生对 LSI 内部进行复位的信号。如果 RES# 引脚输入信号的复位和看门狗定时器上溢的复位同时发生，就优先进行 RES# 引脚的复位，并且 RSTCSR.WOVF 标志变为“0”。

当 RSTE 位为“1”时，在 PCLK 的 257 个状态期间输出 WDTOVF# 信号；当 RSTE 位为“0”时，在 PCLK 的 256 个状态期间输出内部复位信号。在 PCLK 的 1027 个状态期间输出内部复位信号。

因为在 RSTE 位为“1”时产生内部复位信号并且系统时钟的控制寄存器（SCKCR）被复位，所以 PCLK 输入时钟的倍率变为初始值。

因为在 RSTE 位为“0”时不产生内部复位信号并且保持 SCKCR 寄存器的设定，所以 PCLK 输入时钟的倍率不变。

如果在看门狗定时器模式中 TCNT 计数器发生上溢，WOVF 标志就变为“1”。

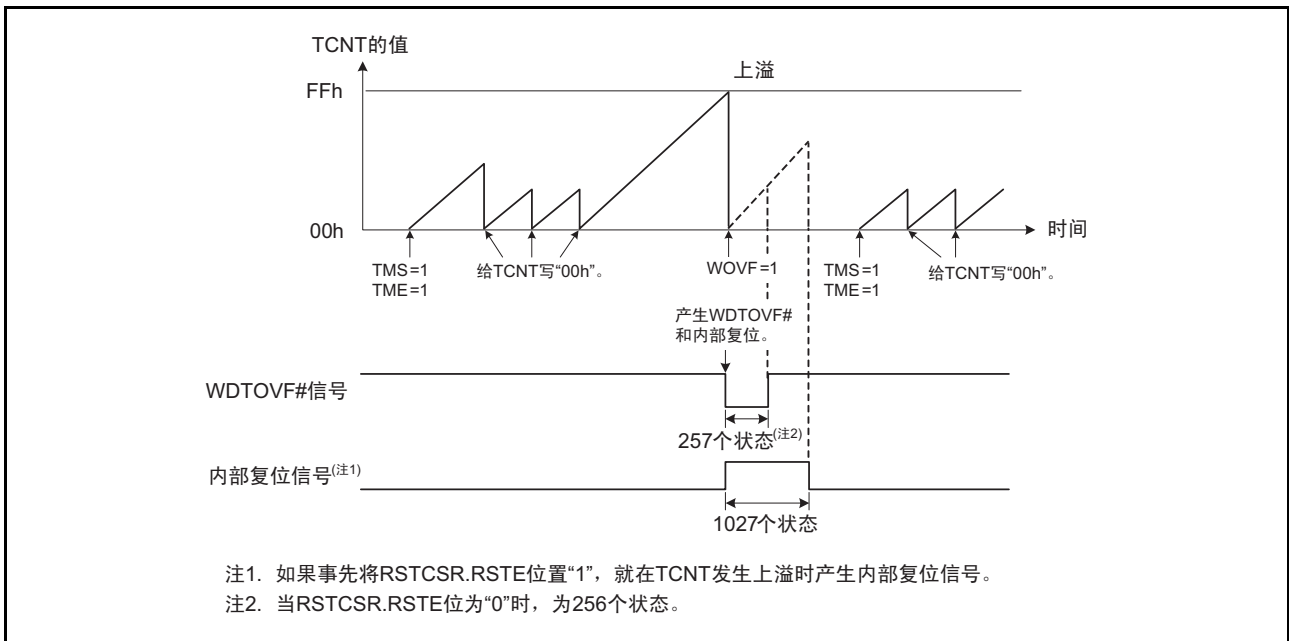


图 19.2 看门狗定时器模式的运行

19.3.2 间隔定时器模式

在用作间隔定时器时，必须将 TCSR.TMS 位置“0”（间隔定时器模式）并且将 TCSR.TME 位置“1”（TCNT 计数器开始计数）。

在用作间隔定时器的情况下，每当 TCNT 计数器发生上溢时，就产生间隔定时器中断（WOVI）。因此，能按固定的时间产生中断。

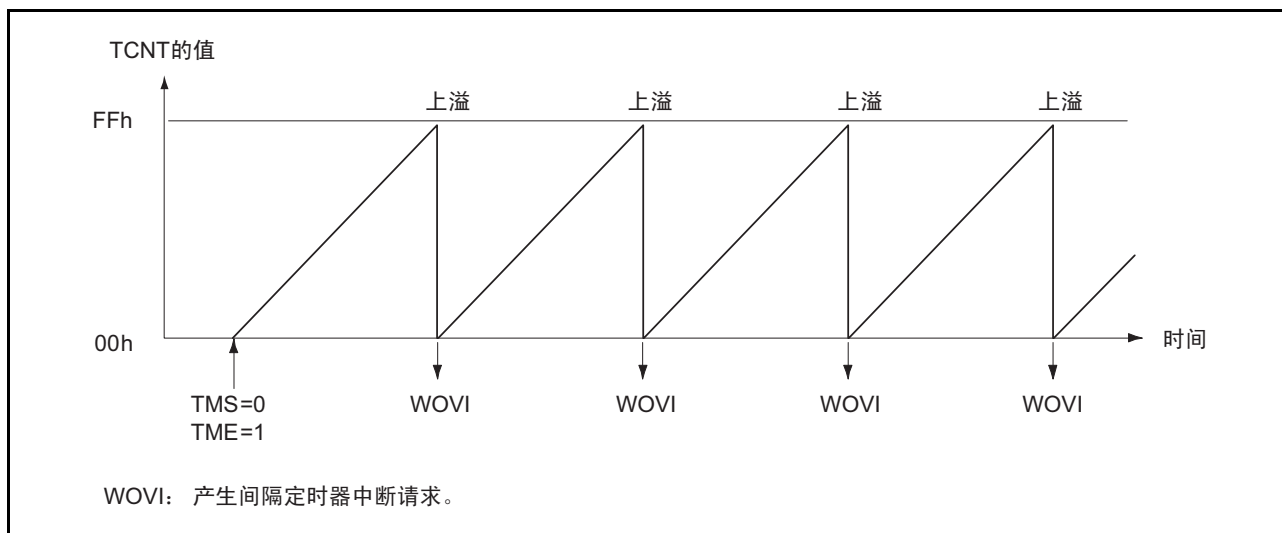


图 19.3 间隔定时器模式的运行

19.4 中断源

在间隔定时器模式中，通过 TCNT 计数器的上溢产生间隔定时器中断（WOVI），详细内容请参照“10. 中断控制器（ICU）”。

表 19.4 WDT 的中断源

| 名称 | 中断源 | 中断状态标志 | DTC 的启动 | DMAC 的启动 |
|------|----------|----------|---------|----------|
| WOVI | TCNT 的上溢 | IR096.IR | 不能 | 不能 |

19.5 使用时的注意事项

19.5.1 存取寄存器时的注意事项

为了 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器不被轻易改写，其写法不同于一般寄存器。

(1) 写 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器

在写 TCNT 计数器、TCSR 寄存器时，必须对写窗口 A 寄存器 (WINA) (0008 8028h) 使用字传送指令。TCNT 计数器和 TCSR 寄存器的写操作被分配在相同的地址，因此必须如图 19.4 进行设定。

在写 TCNT 计数器时，必须在将高位字节置“5Ah”并且将低位字节设定为 TCNT 计数器的写数据后进行传送。

在写 TCSR 寄存器时，必须在将高位字节置“A5h”并且将低位字节设定为 TCSR 寄存器的写数据后进行传送。

在写 RSTCSR 寄存器时，必须对写窗口 B 寄存器 (WINB) (0008 802Ah) 使用字传送指令。

给 RSTCSR.WOVF 标志写“0”的方法不同于 RSTCSR.RSTE 位的写法。

在给 WOVF 标志写“0”时，如图 19.4 所示，必须在将高位字节和低位字节分别置“5Ah”和“00h”后，以 16 位为单位写数据。此时，不影响 RSTE 位。

在写 RSTE 位时，如图 19.4 所示，必须在将高位字节置“5Ah”并且将低位字节设定为 RSTCSR 寄存器的写数据后，以 16 位为单位写数据。此时，不影响 WOVF 标志。

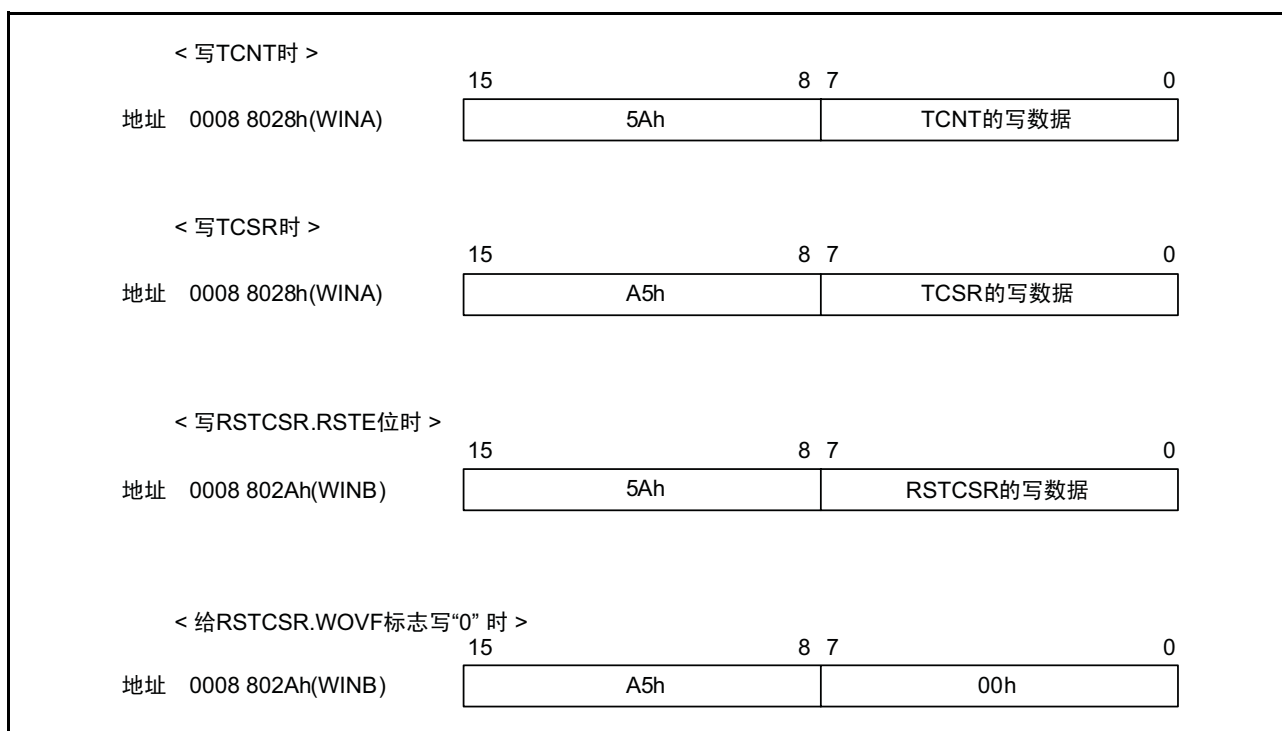


图 19.4 写 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器

(2) 读 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器

TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器的读法和一般寄存器相同。

TCSR 寄存器、TCNT 计数器和 RSTCSR 寄存器必须以 8 位为单位分别存取地址 0008 8028h、地址 0008 8029h 和地址 0008 802Bh。

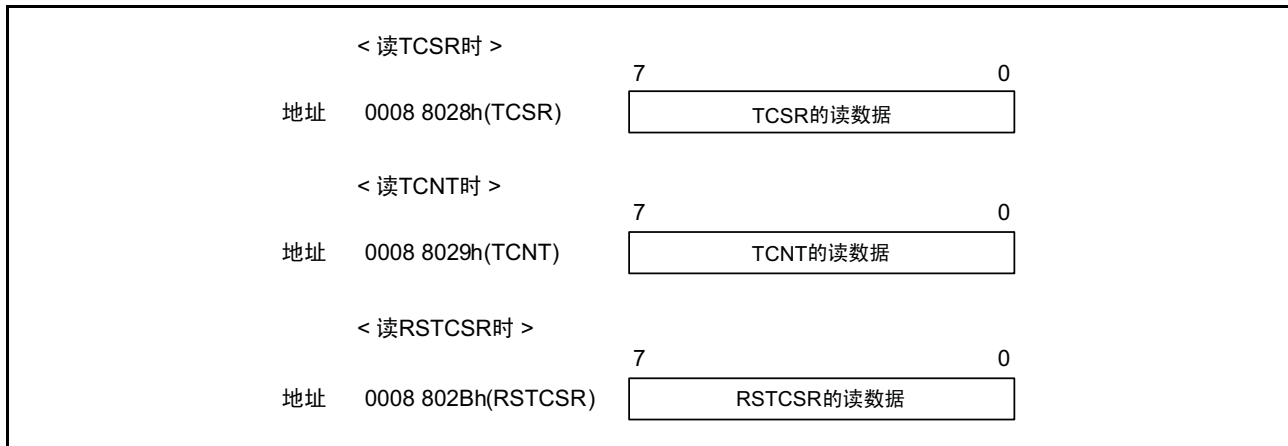


图 19.5 读 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器

19.5.2 定时器的计数器 (TCNT) 的写和递增计数的竞争

即使在写 TCNT 计数器的过程中输入递增计数的时钟，也不进行递增计数而优先写 TCNT 计数器，如图 19.6 所示。

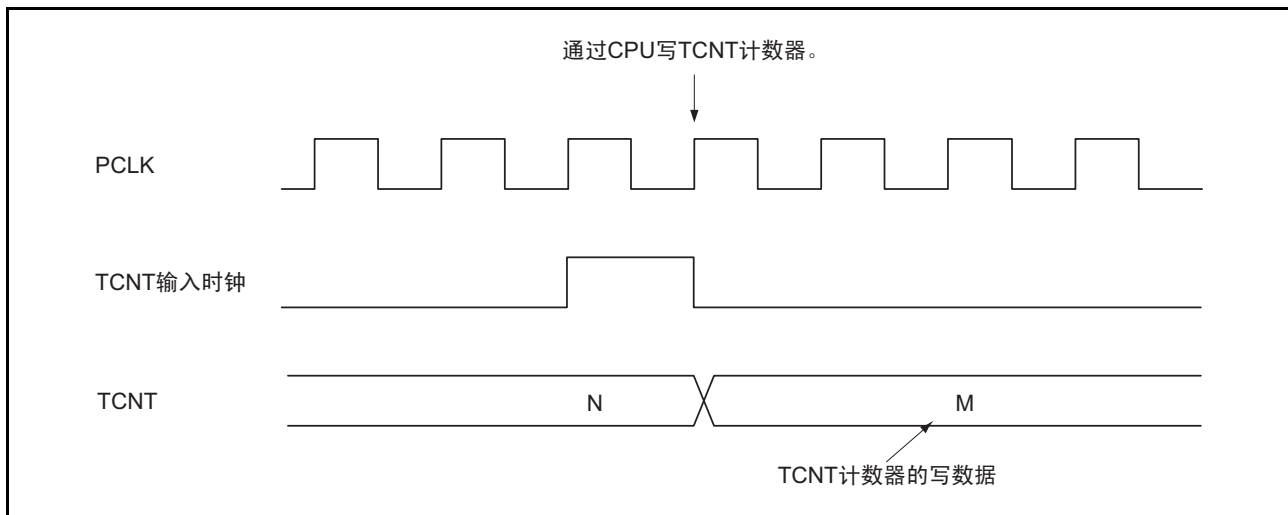


图 19.6 TCNT 计数器的写和递增计数的竞争

19.5.3 CKS[2:0] 位的改写

如果在看门狗定时器运行过程中改写 TCSR.CKS[2:0] 位，就可能无法正常进行递增计数。因此，必须在停止看门狗定时器（将 TCSR.TME 位置“0”）后改写 CKS[2:0] 位。

19.5.4 看门狗定时器模式和间隔定时器模式的转换

如果在看门狗定时器运行过程中进行看门狗定时器模式和间隔定时器模式的转换，就可能无法正常运行。因此，必须在停止看门狗定时器（将 TCSR.TME 位置“0”）后转换定时器模式。

19.5.5 看门狗定时器模式的内部复位

如果在看门狗定时器模式中将 RSTCSR.RSTE 位置“0”，即使 TCNT 计数器发生上溢，LSI 内部也不被复位，但是看门狗定时器的 TCNT 计数器和 TCSR 寄存器被复位。

在 WDTOVF# 信号输出 Low 电平期间，不能写 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器，也无法识别 RSTCSR.WOVF 标志的读操作。因此，要清除 WOVF 标志时，必须在 WDTOVF# 信号变为 High 电平后，在读 RSTCSR 寄存器后给 WOVF 标志写“0”。

19.5.6 WDTOVF# 信号引起的系统复位

如果将 WDTOVF# 信号输入到 RES# 引脚，就不能对 LSI 进行正常的初始化。不能将 WDTOVF# 信号输入到 RES# 引脚。在通过 WDTOVF# 信号对整个系统进行复位时，必须在如图 19.7 所示的电路上进行。

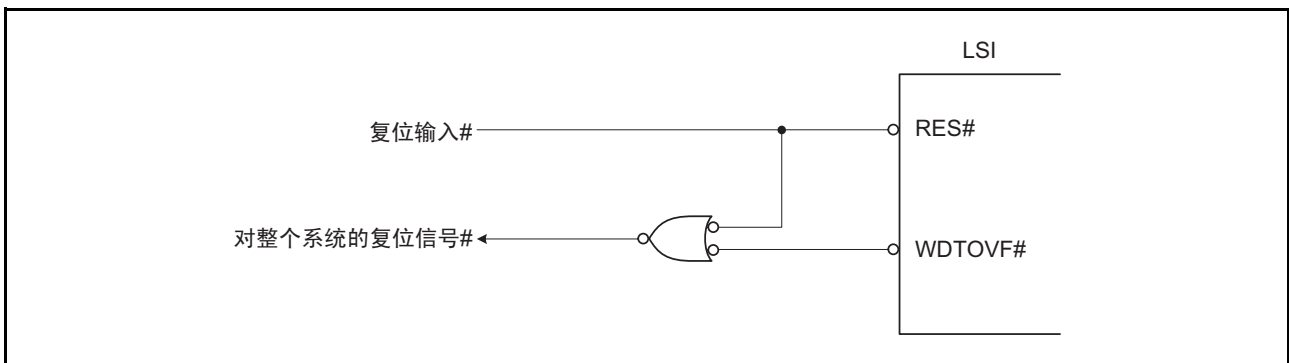


图 19.7 通过 WDTOVF# 信号进行系统复位的电路例子

19.5.7 向看门狗定时器模式和软件待机模式的转移

在用作看门狗定时器模式时，即使在待机控制寄存器的软件待机位（SBYCR.SSBY）置“1”（在执行 WAIT 指令后转移到软件待机模式）的状态下执行 WAIT 指令，也不转移到软件待机模式，而转移到睡眠模式或者全模块时钟停止模式。

如果要转移到软件待机模式转移，就必须在停止看门狗定时器（将 TCSR.TME 位置“0”）后执行 WAIT 指令。

在用作间隔定时器模式时，如果在 SSBY 位置“1”的状态下执行 WAIT 指令，就转移到软件待机模式。详细内容请参照“8. 低功耗功能”。

20. 串行通信接口 (SCI)

RX610 群具有 7 个独立通道的串行通信接口 (SCI: Serial Communication Interface)。

SCI 能进行异步串行通信和时钟同步串行通信。

在异步模式中, 能和 Universal Asynchronous Receiver/Transmitter (UART) 或者 Asynchronous Communication Interface Adapter (ACIA) 等标准异步通信 LSI 进行串行通信。

另外, 作为异步模式的扩展功能, 对应符合 ISO/IEC 7816-3 (Identification Card) 规格的智能卡 (IC 卡) 接口。

20.1 概要

SCI 的规格和各通道功能一览表分别如表 20.1 和表 20.2 所示。

SCI0 ~ SCI4 的框图如图 20.1 所示, SCI5 和 SCI6 的框图如图 20.2 所示。

表 20.1 SCI 的规格

| 项目 | | 内容 |
|-----------|----------|---|
| 串行通信方式 | | <ul style="list-style-type: none"> • 异步 • 时钟同步 • 智能卡接口 |
| 传送率 | | 能通过内部波特率发生器设定任意的位速率。 |
| 全双工通信 | | 发送部: 能通过双缓冲结构进行连续的发送。 接收部: 能通过双缓冲结构进行连续的接收。 |
| 输入 / 输出引脚 | | 参照表 20.3。 |
| 数据传送 | | 可选择 LSB first 或者 MSB first。 |
| 中断源 | | 发送结束、发送数据空、接收数据满、接收错误 |
| 低功耗功能 | | 各通道能设定为模块停止状态。 |
| 异步模式 | 数据长度 | 7 位或者 8 位 |
| | 发送停止位 | 1 位或者 2 位 |
| | 奇偶校验功能 | 偶校验、奇校验或者无奇偶校验 |
| | 接收错误检测功能 | 奇偶校验错误、溢出错误、帧错误 |
| | 中止的检测 | 在发生帧错误时, 能通过直接读 RxDn (n=0 ~ 6) 引脚的电平检测中止。 |
| | 时钟源 | 可选择内部时钟或者外部时钟。 能输入 TMR 的传送率时钟 (SCI5 和 SCI6)。 |
| 时钟同步模式 | 数据长度 | 8 位 |
| | 接收错误的检测 | 溢出错误 |
| 智能卡接口模式 | 错误处理 | 如果在接收时检测到奇偶校验错误, 就自动发送错误信号。 |
| | | 如果在发送时接收到错误信号, 就自动重新发送数据。 |
| | 数据类型 | 正向协议或者反向协议 |

表 20.2 SCI 的各通道功能一览表

| 项目 | SCI0 ~ SCI4 | SCI5 和 SCI6 |
|----------|-------------|-------------|
| 异步模式 | ○ | ○ |
| 时钟同步模式 | ○ | ○ |
| 智能卡接口模式 | ○ | ○ |
| TMR 时钟输入 | — | ○ |

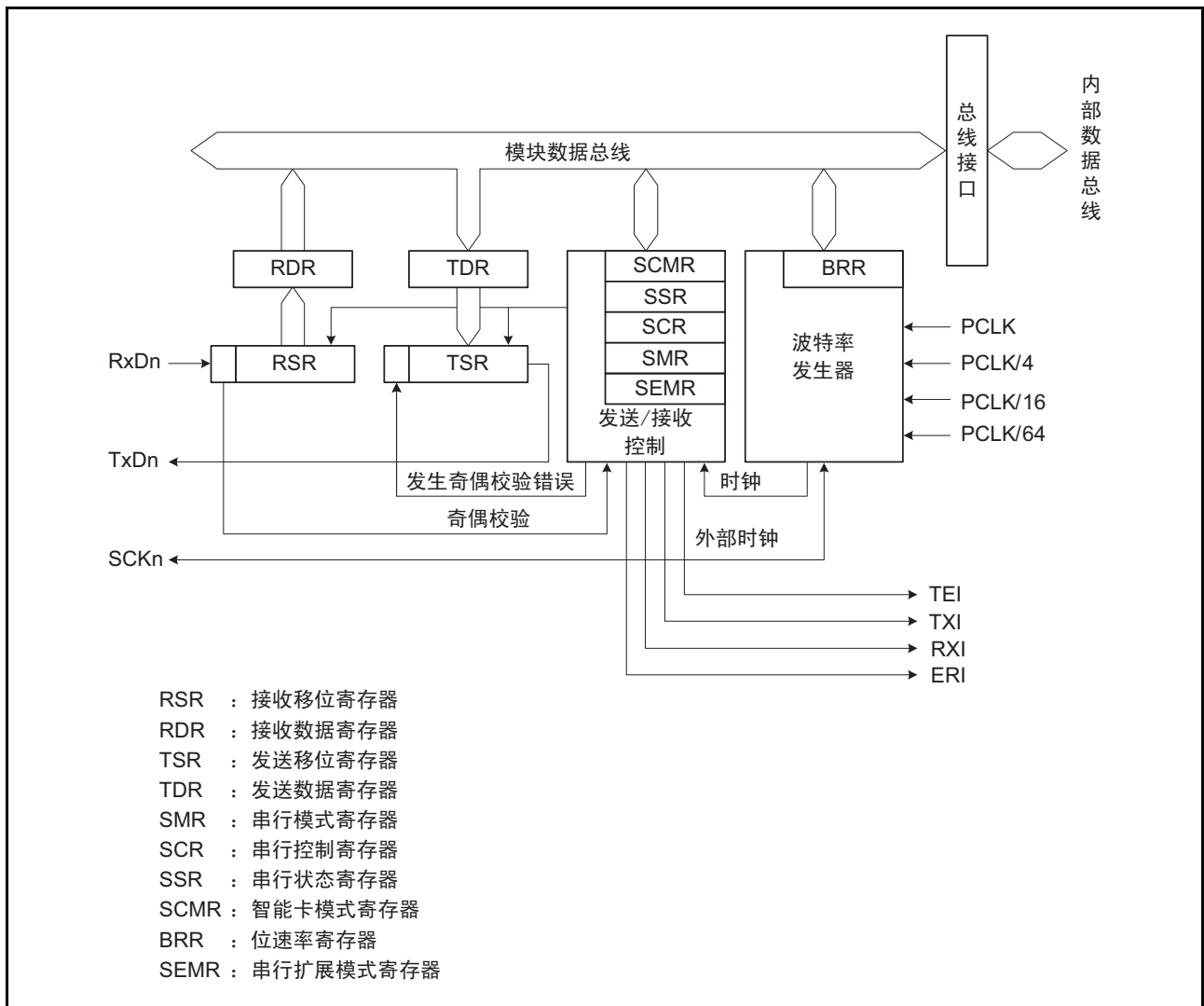


图 20.1 SCI0 ~ SCI4 的框图

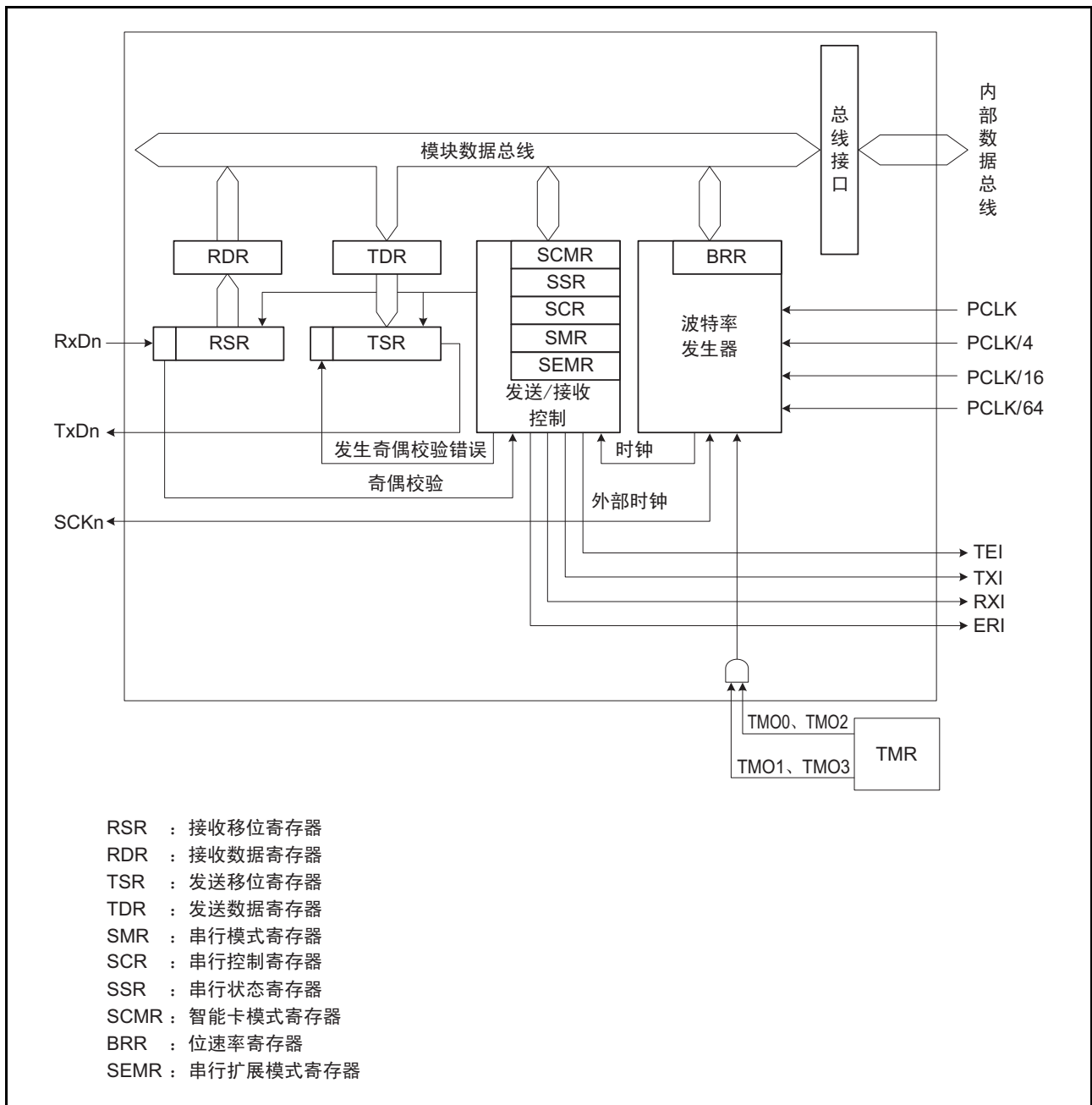


图 20.2 SCI5 和 SCI6 的框图

SCI 使用的输入 / 输出引脚如表 20.3 所示。

表 20.3 SCI 的输入 / 输出引脚

| 通道 | 引脚名 | 输入 / 输出 | 功能 |
|------|------|---------|-------------------|
| SCI0 | SCK0 | 输入 / 输出 | SCI0 的时钟输入 / 输出引脚 |
| | RxD0 | 输入 | SCI0 的接收数据输入引脚 |
| | TxD0 | 输出 | SCI0 的发送数据输出引脚 |
| SCI1 | SCK1 | 输入 / 输出 | SCI1 的时钟输入 / 输出引脚 |
| | RxD1 | 输入 | SCI1 的接收数据输入引脚 |
| | TxD1 | 输出 | SCI1 的发送数据输出引脚 |
| SCI2 | SCK2 | 输入 / 输出 | SCI2 的时钟输入 / 输出引脚 |
| | RxD2 | 输入 | SCI2 的接收数据输入引脚 |
| | TxD2 | 输出 | SCI2 的发送数据输出引脚 |
| SCI3 | SCK3 | 输入 / 输出 | SCI3 的时钟输入 / 输出引脚 |
| | RxD3 | 输入 | SCI3 的接收数据输入引脚 |
| | TxD3 | 输出 | SCI3 的发送数据输出引脚 |
| SCI4 | SCK4 | 输入 / 输出 | SCI4 的时钟输入 / 输出引脚 |
| | RxD4 | 输入 | SCI4 的接收数据输入引脚 |
| | TxD4 | 输出 | SCI4 的发送数据输出引脚 |
| SCI5 | SCK5 | 输入 / 输出 | SCI5 的时钟输入 / 输出引脚 |
| | RxD5 | 输入 | SCI5 的接收数据输入引脚 |
| | TxD5 | 输出 | SCI5 的发送数据输出引脚 |
| SCI6 | SCK6 | 输入 / 输出 | SCI6 的时钟输入 / 输出引脚 |
| | RxD6 | 输入 | SCI6 的接收数据输入引脚 |
| | TxD6 | 输出 | SCI6 的发送数据输出引脚 |

20.2 寄存器说明

SCI 的寄存器一览表如表 20.4 所示。

表 20.4 SCI 的寄存器一览表 (1/2)

| 通道 | 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 | |
|------|-----------|------|-------|------------|------|--|
| SCI0 | 串行模式寄存器 | SMR | 00h | 0008 8240h | 8 | |
| | 位速率寄存器 | BRR | FFh | 0008 8241h | 8 | |
| | 串行控制寄存器 | SCR | 0xh | 0008 8242h | 8 | |
| | 发送数据寄存器 | TDR | FFh | 0008 8243h | 8 | |
| | 串行状态寄存器 | SSR | x4h | 0008 8244h | 8 | |
| | 接收数据寄存器 | RDR | 00h | 0008 8245h | 8 | |
| | 智能卡模式寄存器 | SCMR | F2h | 0008 8246h | 8 | |
| | 串行扩展模式寄存器 | SEMR | 00h | 0008 8247h | 8 | |
| SCI1 | 串行模式寄存器 | SMR | 00h | 0008 8248h | 8 | |
| | 位速率寄存器 | BRR | FFh | 0008 8249h | 8 | |
| | 串行控制寄存器 | SCR | 0xh | 0008 824Ah | 8 | |
| | 发送数据寄存器 | TDR | FFh | 0008 824Bh | 8 | |
| | 串行状态寄存器 | SSR | x4h | 0008 824Ch | 8 | |
| | 接收数据寄存器 | RDR | 00h | 0008 824Dh | 8 | |
| | 智能卡模式寄存器 | SCMR | F2h | 0008 824Eh | 8 | |
| | 串行扩展模式寄存器 | SEMR | 00h | 0008 824Fh | 8 | |
| SCI2 | 串行模式寄存器 | SMR | 00h | 0008 8250h | 8 | |
| | 位速率寄存器 | BRR | FFh | 0008 8251h | 8 | |
| | 串行控制寄存器 | SCR | 0xh | 0008 8252h | 8 | |
| | 发送数据寄存器 | TDR | FFh | 0008 8253h | 8 | |
| | 串行状态寄存器 | SSR | x4h | 0008 8254h | 8 | |
| | 接收数据寄存器 | RDR | 00h | 0008 8255h | 8 | |
| | 智能卡模式寄存器 | SCMR | F2h | 0008 8256h | 8 | |
| | 串行扩展模式寄存器 | SEMR | 00h | 0008 8257h | 8 | |
| SCI3 | 串行模式寄存器 | SMR | 00h | 0008 8258h | 8 | |
| | 位速率寄存器 | BRR | FFh | 0008 8259h | 8 | |
| | 串行控制寄存器 | SCR | 0xh | 0008 825Ah | 8 | |
| | 发送数据寄存器 | TDR | FFh | 0008 825Bh | 8 | |
| | 串行状态寄存器 | SSR | x4h | 0008 825Ch | 8 | |
| | 接收数据寄存器 | RDR | 00h | 0008 825Dh | 8 | |
| | 智能卡模式寄存器 | SCMR | F2h | 0008 825Eh | 8 | |
| | 串行扩展模式寄存器 | SEMR | 00h | 0008 825Fh | 8 | |

表 20.4 SCI 的寄存器一览表 (2/2)

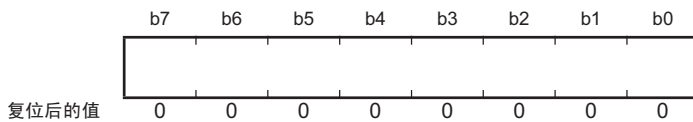
| 通道 | 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 | |
|------|-----------|------|-------|------------|------|--|
| SCI4 | 串行模式寄存器 | SMR | 00h | 0008 8260h | 8 | |
| | 位速率寄存器 | BRR | FFh | 0008 8261h | 8 | |
| | 串行控制寄存器 | SCR | 0xh | 0008 8262h | 8 | |
| | 发送数据寄存器 | TDR | FFh | 0008 8263h | 8 | |
| | 串行状态寄存器 | SSR | x4h | 0008 8264h | 8 | |
| | 接收数据寄存器 | RDR | 00h | 0008 8265h | 8 | |
| | 智能卡模式寄存器 | SCMR | F2h | 0008 8266h | 8 | |
| | 串行扩展模式寄存器 | SEMR | 00h | 0008 8267h | 8 | |
| SCI5 | 串行模式寄存器 | SMR | 00h | 0008 8268h | 8 | |
| | 位速率寄存器 | BRR | FFh | 0008 8269h | 8 | |
| | 串行控制寄存器 | SCR | 0xh | 0008 826Ah | 8 | |
| | 发送数据寄存器 | TDR | FFh | 0008 826Bh | 8 | |
| | 串行状态寄存器 | SSR | x4h | 0008 826Ch | 8 | |
| | 接收数据寄存器 | RDR | 00h | 0008 826Dh | 8 | |
| | 智能卡模式寄存器 | SCMR | F2h | 0008 826Eh | 8 | |
| | 串行扩展模式寄存器 | SEMR | 00h | 0008 826Fh | 8 | |
| SCI6 | 串行模式寄存器 | SMR | 00h | 0008 8270h | 8 | |
| | 位速率寄存器 | BRR | FFh | 0008 8271h | 8 | |
| | 串行控制寄存器 | SCR | 0xh | 0008 8272h | 8 | |
| | 发送数据寄存器 | TDR | FFh | 0008 8273h | 8 | |
| | 串行状态寄存器 | SSR | x4h | 0008 8274h | 8 | |
| | 接收数据寄存器 | RDR | 00h | 0008 8275h | 8 | |
| | 智能卡模式寄存器 | SCMR | F2h | 0008 8276h | 8 | |
| | 串行扩展模式寄存器 | SEMR | 00h | 0008 8277h | 8 | |

20.2.1 接收移位寄存器 (RSR)

RSR 寄存器是将 RxDn 引脚输入的串行数据转换为并行数据的接收移位寄存器。
如果接收 1 帧的数据，数据就自动被传送到 RDR 寄存器。
CPU 不能直接存取 RSR 寄存器。

20.2.2 接收数据寄存器 (RDR)

地址 SCI0.RDR 0008 8245h、SCI1.RDR 0008 824Dh、SCI2.RDR 0008 8255h、SCI3.RDR 0008 825Dh
SCI4.RDR 0008 8265h、SCI5.RDR 0008 826Dh、SCI6.RDR 0008 8275h



RDR 寄存器是保存接收数据的 8 位寄存器。

如果接收 1 帧的数据，就将接收数据从 RSR 寄存器传送到此寄存器并且 RSR 寄存器变为能接收下一个数据的状态。

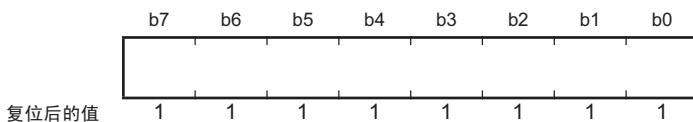
因为 RSR 寄存器和 RDR 寄存器是双缓冲结构，所以能连续接收。

在产生接收数据满中断 (RXI) 请求时，只能读 1 次 RDR 寄存器。必须注意：如果不从 RDR 读接收数据而接收下一帧的数据，就会产生溢出错误。

CPU 不能写 RDR 寄存器。

20.2.3 发送数据寄存器 (TDR)

地址 SCI0.TDR 0008 8243h、SCI1.TDR 0008 824Bh、SCI2.TDR 0008 8253h、SCI3.TDR 0008 825Bh
SCI4.TDR 0008 8263h、SCI5.TDR 0008 826Bh、SCI6.TDR 0008 8273h



TDR 寄存器是保存发送数据的 8 位寄存器。

如果检测到 TSR 寄存器为空，就将写在 TDR 寄存器的发送数据传送到 TSR 寄存器，开始发送。

因为 TDR 寄存器和 TSR 寄存器为双缓冲结构，所以能连续发送。如果在发送了 1 帧的数据时将下一个发送数据写到 TDR 寄存器，就将此数据传送到 TSR 寄存器，继续发送。

CPU 能随时读写 TDR 寄存器。在产生发送数据空中断 (TXI) 请求时，只能给 TDR 寄存器写 1 次发送数据。

20.2.4 发送移位寄存器 (TSR)

TSR 寄存器是发送串行数据的移位寄存器。

写在 TDR 寄存器的发送数据自动被传送到 TSR 寄存器，通过将数据发送到 TxDn 引脚进行串行数据的发送。

CPU 不能直接存取 TSR 寄存器。

20.2.5 串行模式寄存器 (SMR)

注 1. 在一般的串行通信接口模式和智能卡接口模式中, SMR 寄存器的部分位的功能不同。

(1) 串行通信接口模式 (SCMR.SMIF 位 =0)

地址 SCI0.SMR 0008 8240h、SCI1.SMR 0008 8248h、SCI2.SMR 0008 8250h、SCI3.SMR 0008 8258h
SCI4.SMR 0008 8260h、SCI5.SMR 0008 8268h、SCI6.SMR 0008 8270h

| | | | | | | | |
|----|-----|----|----|------|----|----------|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| CM | CHR | PE | PM | STOP | — | CKS[1:0] | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|---------|---|--------------|
| b1-b0 | CKS[1:0] | 时钟选择位 | b1 b0 0 0: PCLK 时钟 (n=0) (注 1) 0 1: PCLK/4 时钟 (n=1) (注 1) 1 0: PCLK/16 时钟 (n=2) (注 1) 1 1: PCLK/64 时钟 (n=3) (注 1) | R/W (注 4) |
| b2 | — | 保留位 | 读写值都为“0”。 | R/W (注 4) |
| b3 | STOP | 停止位长选择位 | (只在异步模式中有效) 0: 1 个停止位 1: 2 个停止位 | R/W (注 4) |
| b4 | PM | 奇偶校验模式位 | (只在异步模式中并且 PE 位为“1”时有效) 0: 偶校验 1: 奇校验 | R/W (注 4) |
| b5 | PE | 奇偶校验允许位 | (只在异步模式中有效) • 发送时 0: 无奇偶校验位 1: 附加奇偶校验位 • 接收时 0: 不进行奇偶校验 1: 进行奇偶校验 | R/W (注 4) |
| b6 | CHR | 字符长位 | (只在异步模式中有效) 0: 数据长度为 8 位 (注 2) 1: 数据长度为 7 位 (注 3) | R/W (注 4) |
| b7 | CM | 通信模式位 | 0: 异步通信模式 1: 时钟同步通信模式 | R/W (注 4) |

注 1. n 为设定值的 10 进制数, 表示“20.2.9 位速率寄存器 (BRR)”中的 n 的值。

注 2. 在时钟同步模式中, 与设定值无关, 数据长度为 8 位。

注 3. 固定为 LSB first, 在发送时不发送 TDR 寄存器的 MSB (b7)。

注 4. 只能在 SCR.TE 位和 SCR.RE 位都为“0”(禁止串行发送和串行接收)时写这些位。

SMR 寄存器选择通信格式以及内部波特率发生器的时钟源。

CKS[1:0] 位 (时钟选择位)

这些位选择内部波特率发生器的时钟源。

有关 CKS[1:0] 位的设定值和波特率的关系, 请参照 “20.2.9 位速率寄存器 (BRR)”。

STOP 位 (停止位长选择位)

此位选择发送数据的停止位长。

在接收时, 与此位的设定无关, 只检查停止位的第 1 位, 当第 2 位为 “0” 时, 将第 2 位视为下一个发送帧的起始位。

PM 位 (奇偶校验模式位)

此位选择发送和接收时的奇偶校验 (偶校验或者奇校验)。

PE 位 (奇偶校验允许位)

当 PE 位为 “1” 时, 在发送时附加奇偶校验位, 在接收时进行奇偶校验。

CHR 位 (字符长位)

此位选择发送 / 接收数据的数据长度。

在时钟同步模式中, 数据长度为 8 位。

CM 位 (通信模式位)

此位选择异步模式或者时钟同步模式。

(2) 智能卡接口模式 (SCMR.SMIF 位 =1)

地址 SCI0.SMR 0008 8240h、SCI1.SMR 0008 8248h、SCI2.SMR 0008 8250h、SCI3.SMR 0008 8258h
SCI4.SMR 0008 8260h、SCI5.SMR 0008 8268h、SCI6.SMR 0008 8270h

| | | | | | | | |
|-------|-----|----|----|----------|----------|----------|----------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| GM | BLK | PE | PM | BCP[1:0] | BCP[1:0] | CKS[1:0] | CKS[1:0] |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|---------|---|-------------|
| b1-b0 | CKS[1:0] | 时钟选择位 | b1 b0 0 0: PCLK 时钟 (n=0) (注1) 0 1: PCLK/4 时钟 (n=1) (注1) 1 0: PCLK/16 时钟 (n=2) (注1) 1 1: PCLK/64 时钟 (n=3) (注1) | R/W (注3) |
| b3-b2 | BCP[1:0] | 基本时钟脉冲位 | 通过和 SCMR.BCP2 位组合进行选择。 SCMR.BCP2 位和 SMR.BCP[1:0] 位的设定值 BCP2 b3 b2 0 0 0: 93 个时钟 (S=93) (注2) 0 0 1: 128 个时钟 (S=128) (注2) 0 1 0: 186 个时钟 (S=186) (注2) 0 1 1: 512 个时钟 (S=512) (注2) 1 0 0: 32 个时钟 (S=32) (注2) (初始值) 1 0 1: 64 个时钟 (S=64) (注2) 1 1 0: 372 个时钟 (S=372) (注2) 1 1 1: 256 个时钟 (S=256) (注2) | R/W (注3) |
| b4 | PM | 奇偶校验模式位 | (只在异步模式中并且 PE 位为 “1” 时有效) 0: 偶校验 1: 奇校验 | R/W (注3) |
| b5 | PE | 奇偶校验允许位 | (只在异步模式中有效) 当 PE 位为 “1” 时, 在发送时附加奇偶校验位, 在接收时进行奇偶校验。在智能卡接口模式中, 必须将 PE 位置 “1”。 | R/W (注3) |
| b6 | BLK | 块传送模式位 | 0: 正常模式 1: 块传送模式 | R/W (注3) |
| b7 | GM | GSM 模式位 | 0: 正常模式 1: GSM 模式 | R/W (注3) |

注1. n 为设定值的 10 进制数, 表示 “20.2.9 位速率寄存器 (BRR)” 中的 n 的值。

注2. S 表示 “20.2.9 位速率寄存器 (BRR)” 中的 S 的值。

注3. 只能在 SCR.TE 位和 SCR.RE 位都为 “0” (禁止串行发送和串行接收) 时写这些位。

SMR 寄存器选择通信格式以及内部波特率发生器的时钟源。

CKS[1:0] 位 (时钟选择位)

这些位选择内部波特率发生器的时钟源。

有关 CKS[1:0] 位的设定值和波特率的关系, 请参照 “20.2.9 位速率寄存器 (BRR)”。

BCP[1:0] 位 (基本时钟脉冲位)

这些位选择智能卡接口模式中 1 位传送时间的基本时钟数。

通过和 SCMR.BCP2 位组合进行选择。

详细内容请参照“20.5.4 接收数据的采样时序和接收容限”。

PM 位 (奇偶校验模式位)

此位选择发送和接收时的奇偶校验 (偶校验或者奇校验)。

有关此位在智能卡接口模式中的使用方法, 请参照“20.5.2 数据格式 (块传送模式除外)”。

PE 位 (奇偶校验允许位)

必须将 PE 位置“1”。

在发送时附加奇偶校验位, 在接收时进行奇偶校验。

BLK (块传送模式位)

如果将 BLK 位置“1”, 就以块传送模式运行。

有关块传送模式, 请参照“20.5.3 块传送模式”。

GM 位 (GSM 模式位)

如果将 GM 位置“1”, 就以 GSM 模式运行。

在 GSM 模式中, SSR.TEND 标志变为“1”的时序为开始发送后的 11.0etu (etu: Elementary Time Unit, 1 位传送时间), 并且追加时钟输出控制功能, 详细内容请参照“20.5.6 串行数据的发送 (块传送模式除外)”和“20.5.8 时钟的输出控制”。

20.2.6 串行控制寄存器 (SCR)

注 1. 在一般的串行通信接口模式和智能卡接口模式中, SCR 寄存器的部分位的功能不同。

(1) 串行通信接口模式 (SCMR.SMIF 位 =0)

地址 SCI0.SCR 0008 8242h、SCI1.SCR 0008 824Ah、SCI2.SCR 0008 8252h、SCI3.SCR 0008 825Ah
SCI4.SCR 0008 8262h、SCI5.SCR 0008 826Ah、SCI6.SCR 0008 8272h

| | | | | | | | |
|-----|-----|----|----|----|------|----------|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TIE | RIE | TE | RE | — | TEIE | CKE[1:0] | |

复位后的值

0 0 0 0 0 0 0 0

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|-------|--|--------------|
| b1-b0 | CKE[1:0] | 时钟允许位 | <ul style="list-style-type: none"> • SCI0 ~ SCI4 的情况 (异步模式) b1 b0 0 0: 内部波特率发生器 SCKn 引脚能用作输入 / 输出端口。 0 1: 内部波特率发生器 从 SCKn 引脚输出频率为位速率的时钟。 1 0: 外部时钟 必须从 SCKn 引脚输入频率为 16 倍位速率的时钟。 1 1: 外部时钟 必须从 SCKn 引脚输入频率为 16 倍位速率的时钟。 (时钟同步模式) b1 b0 0 0: 内部时钟 SCKn 引脚为时钟的输出引脚。 0 1: 内部时钟 SCKn 引脚为时钟的输出引脚。 1 0: 外部时钟 SCKn 引脚为时钟的输入引脚。 1 1: 外部时钟 SCKn 引脚为时钟的输入引脚。 | R/W (注 1) |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|-----------|---|--------------|
| b1-b0 | CKE[1:0] | 时钟允许位 | <ul style="list-style-type: none"> • SCI5 和 SCI6 的情况 (异步模式) b1 b0 0 0: 内部波特率发生器 SCKn 引脚能用作输入 / 输出端口。 0 1: 内部波特率发生器 从 SCKn 引脚输出频率为位速率的时钟。 1 0: 外部时钟或者 TMR 时钟 <ul style="list-style-type: none"> • 在使用外部时钟时, 必须从 SCKn 引脚输入频率为 16 倍位速率的时钟。 • 能使用 TMR 时钟。 1 1: 外部时钟或者 TMR 时钟 <ul style="list-style-type: none"> • 在使用外部时钟时, 必须从 SCKn 引脚输入频率为 16 倍位速率的时钟。 • 能使用 TMR 时钟。 (时钟同步模式) b1 b0 0 0: 内部时钟 SCKn 引脚为时钟的输出引脚。 0 1: 内部时钟 SCKn 引脚为时钟的输出引脚。 1 0: 外部时钟 SCKn 引脚为时钟的输入引脚。 1 1: 外部时钟 SCKn 引脚为时钟的输入引脚。 | R/W (注 1) |
| b2 | TEIE | 发送结束中断允许位 | 0: 禁止 TEI 中断 1: 允许 TEI 中断 | R/W |
| b3 | — | 保留位 | 读取值为不定值, 只能写 “0”。 | R/W |
| b4 | RE | 接收允许位 | 0: 禁止串行接收 1: 允许串行接收 | R/W (注 2) |
| b5 | TE | 发送允许位 | 0: 禁止串行发送 1: 允许串行发送 | R/W (注 2) |
| b6 | RIE | 接收中断允许位 | 0: 禁止 RXI 中断和 ERI 中断 1: 允许 RXI 中断和 ERI 中断 | R/W |
| b7 | TIE | 发送结束中断位 | 0: 禁止 TXI 中断 1: 允许 TXI 中断 | R/W |

注 1. 只能在 TE 位和 RE 位都为 “0” 时写这些位。

注 2. 只能在 TE 位和 RE 位都为 “0” 时写 “1”。一旦将 TE 位或者 RE 位置 “1”, 就只能在 TE 位和 RE 位都为 “0” 时写此位。

SCR 寄存器是控制发送 / 接收以及选择发送 / 接收时钟源的寄存器。

CKE[1:0] 位 (时钟允许位)

这些位选择时钟源和 SCKn 引脚的功能。

TEIE 位 (发送结束中断允许位)

此位允许或者禁止 TEI 中断。

要禁止 TEI 中断时, 将 TEIE 位置 “0”。

RE 位 (接收允许位)

此位允许或者禁止串行接收。

在将 RE 位置 “1” 后, 如果在异步模式和时钟同步模式中分别检测到起始位和同步时钟输入, 就开始串行接收。必须在将 RE 位置 “1” 前设定 SMR 寄存器, 决定接收格式。

即使通过将 RE 位置 “0” 来停止接收, SSR.ORER 标志、SSR.FER 标志和 SSR.PER 标志也不受影响而保持原来的状态。

TE 位 (发送允许位)

此位允许或者禁止串行发送。

如果将 TE 位置 “1”, 就通过给 TDR 寄存器写发送数据, 开始串行发送。必须在将 TE 位置 “1” 前设定 SMR 寄存器, 决定发送格式。

RIE 位 (接收中断允许位)

此位允许或者禁止 RXI 中断和 ERI 中断。

要禁止 RXI 中断时, 将 RIE 位置 “0”。

如果在从 SSR.ORER 标志、SSR.FER 标志和 SSR.PER 标志读 “1” 后将这些标志位置 “0”, 或者将 RIE 位置 “0”, ERI 中断请求信号就消失。

TIE 位 (发送中断允许位)

此位允许或者禁止通知 TXI 中断。

要禁止 TXI 中断时, 将 TIE 位置 “0”。

(2) 智能卡接口模式 (SCMR.SMIF=1)

地址 SCI0.SCR 0008 8242h、SCI1.SCR 0008 824Ah、SCI2.SCR 0008 8252h、SCI3.SCR 0008 825Ah
SCI4.SCR 0008 8262h、SCI5.SCR 0008 826Ah、SCI6.SCR 0008 8272h

| | | | | | | | |
|-----|-----|----|----|----|------|----------|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| TIE | RIE | TE | RE | — | TEIE | CKE[1:0] | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|-----------|--|--------------|
| b1-b0 | CKE[1:0] | 时钟允许位 | <ul style="list-style-type: none"> SMR.GM 位 =0 的情况 b1 b0 0 0: 禁止输出 (SCKn 引脚可用作输入 / 输出端口) 0 1: 时钟输出 1 0: 不能设定 1 1: 不能设定 <ul style="list-style-type: none"> SMR.GM 位 =1 的情况 0 0: 固定为 Low 电平输出 0 1: 时钟输出 1 0: 固定为 High 电平输出 1 1: 时钟输出 | R/W (注 1) |
| b2 | TEIE | 发送结束中断允许位 | 在智能卡接口模式中, 必须置 “0”。 | R/W |
| b3 | — | 保留位 | 读取值为不定值, 只能写 “0”。 | R/W |
| b4 | RE | 接收允许位 | 0: 禁止串行接收 1: 允许串行接收 | R/W (注 2) |
| b5 | TE | 发送允许位 | 0: 禁止串行发送 1: 允许串行发送 | R/W (注 2) |
| b6 | RIE | 接收中断允许位 | 0: 禁止 RXI 中断和 ERI 中断 1: 允许 RXI 中断和 ERI 中断 | R/W |
| b7 | TIE | 发送中断允许位 | 0: 禁止 TXI 中断 1: 允许 TXI 中断 | R/W |

注 1. 只能在 TE 位和 RE 位都为 “0” 时写这些位。

注 2. 只能在 TE 位和 RE 位都为 “0” 时写 “1”。一旦将 TE 位或者 RE 位置 “1”, 就只能在 TE 位和 RE 位都为 “0” 时写此位。

SCR 寄存器是控制发送 / 接收和中断以及选择发送 / 接收时钟源的寄存器。
有关各中断源, 请参照 “20.6 中断源”。

CKE[1:0] 位 (时钟允许位)

这些位控制 SCKn 引脚的时钟输出。

能在 GSM 模式中对时钟输出进行动态转换, 详细内容请参照 “20.5.8 时钟的输出控制”。

TEIE 位 (发送结束中断允许位)

在智能卡接口模式中, 必须将此位置 “0”。

RE 位 (接收允许位)

此位允许或者禁止串行接收。

在将 RE 位置 “1” 后, 如果检测到起始位, 就开始串行接收。必须在将 RE 位置 “1” 前设定 SMR 寄存器, 决定接收格式。

即使通过将 RE 位置 “0” 来停止接收, SSR.ORER 标志、SSR.FER 标志和 SSR.PER 标志也不受影响而保持原来的状态。

TE 位 (发送允许位)

此位允许或者禁止串行发送。

如果将 TE 位置 “1”, 就通过给 TDR 寄存器写发送数据, 开始串行发送。必须在将 TE 位置 “1” 前设定 SMR 寄存器, 决定发送格式。

RIE 位 (接收中断允许位)

此位允许或者禁止 RXI 中断和 ERI 中断。

要禁止 RXI 中断时, 将 RIE 位置 “0”。

如果在从 SSR.ORER 标志、SSR.FER 标志和 SSR.PER 标志读 “1” 后将这些标志位置 “0”, 或者将 RIE 位置 “0”, ERI 中断请求信号就消失。

TIE 位 (发送中断允许位)

此位允许或者禁止通知 TXI 中断。

要禁止 TXI 中断时, 将 TIE 位置 “0”。

20.2.7 串行状态寄存器 (SSR)

注 1. 在一般的串行通信接口模式和智能卡接口模式中, SSR 寄存器的部分位的功能不同。

(1) 串行通信接口模式 (SCMR.SMIF 位 =0)

地址 SCI0.SSR 0008 8244h、SCI1.SSR 0008 824Ch、SCI2.SSR 0008 8254h、SCI3.SSR 0008 825Ch
SCI4.SSR 0008 8264h、SCI5.SSR 0008 826Ch、SCI6.SSR 0008 8274h

| | | | | | | | | |
|-------|----|----|------|-----|-----|------|----|----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | ORER | FER | PER | TEND | — | — |
| 复位后的值 | x | x | 0 | 0 | 0 | 1 | 0 | 0 |

x: 不定值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|----------|-----------------------------|---------------|
| b1-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b2 | TEND | 发送结束标志 | 0: 正在发送字符 1: 字符发送结束 | R |
| b3 | PER | 奇偶校验错误标志 | 0: 未发生奇偶校验错误 1: 发生奇偶校验错误 | R(W) (注 1) |
| b4 | FER | 帧错误标志 | 0: 未发生帧错误 1: 发生帧错误 | R(W) (注 1) |
| b5 | ORER | 溢出错误标志 | 0: 未发生溢出错误 1: 发生溢出错误 | R(W) (注 1) |
| b7-b6 | — | 保留位 | 读取值为不定值, 只能写“1”。 | R/W |

注 1. 只能写“0”。

SSR 寄存器由 SCI 的状态标志构成。

TEND 标志 (发送结束标志)

此标志表示发送已经结束。

[为“1”的条件]

- SCR.TE 位为“0” (禁止串行发送) 时
- 在发送字符的最后一位时未更新 TDR 寄存器时

[为“0”的条件]

- 给 TDR 寄存器写发送数据时

在通过给 TDR 寄存器写发送数据将 TEND 标志置“0”时, 必须读 TEND 标志, 确认 TEND 标志已变为“0”。

PER 标志 (奇偶校验错误标志)

此标志表示在异步模式中接收的数据发生了奇偶校验错误。

[为“1”的条件]

- 在接收时检测到奇偶校验错误时
将发生奇偶校验错误时的接收数据传送到RDR寄存器，但是不产生RXI中断请求。在PER标志为“1”的状态下，不能继续进行以后的串行接收。在时钟同步模式中，也不能继续进行串行发送。

[为“0”的条件]

- 读“1”后写“0”（必须在写“0”后确认PER标志已变为“0”）时
即使将SCR.RE位置“0”（禁止串行接收），PER标志也不受影响而保持原来的状态。

FER 标志 (帧错误标志)

此标志表示在异步模式中接收的数据发生了帧错误。

[为“1”的条件]

- 停止位为“0”时
在2个停止位模式中，只判断第1个停止位是否为“1”而不检查第2个停止位。将发生帧错误的接收数据传送到RDR寄存器，但是不产生RXI中断请求。在FER标志为“1”的状态下，不能继续进行以后的串行接收。在时钟同步模式中，也不能继续进行串行发送。

[为“0”的条件]

- 读“1”后写“0”（必须在写“0”后确认FER标志已变为“0”）时
即使将SCR.RE位置“0”，FER标志也不受影响而保持原来的状态。

ORER 标志 (溢出错误标志)

此标志表示接收的数据发生了溢出错误。

[为“1”的条件]

- 不读RDR寄存器的接收数据就接收到下一个数据时
RDR寄存器保持发生溢出错误前的接收数据而放弃后面的接收数据。在ORER标志为“1”的状态下，不能继续进行以后的串行接收。在时钟同步模式中，也不能继续进行串行发送。

[为“0”的条件]

- 读“1”后写“0”（必须在写“0”后确认ORER标志已变为“0”）时
即使将SCR.RE位置“0”，ORER标志也不受影响而保持原来的状态。

(2) 智能卡接口模式 (SCMR.SMIF=1)

地址 SCI0.SSR 0008 8244h、SCI1.SSR 0008 824Ch、SCI2.SSR 0008 8254h、SCI3.SSR 0008 825Ch
SCI4.SSR 0008 8264h、SCI5.SSR 0008 826Ch、SCI6.SSR 0008 8274h

| | | | | | | | | |
|-------|----|----|------|-----|-----|------|----|----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | ORER | ERS | PER | TEND | — | — |
| 复位后的值 | x | x | 0 | 0 | 0 | 1 | 0 | 0 |

x: 不定值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|----------|--|--------------|
| b1-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b2 | TEND | 发送结束标志 | 0: 正在发送字符 1: 字符发送结束 | R |
| b3 | PER | 奇偶校验错误标志 | 0: 未发生奇偶校验错误 1: 发生奇偶校验错误 | R(W) (注1) |
| b4 | ERS | 错误信号状态标志 | 0: 无错误信号的 Low 电平响应 1: 有错误信号的 Low 电平响应 | R(W) (注1) |
| b5 | ORER | 溢出错误标志 | 0: 未发生溢出错误 1: 发生溢出错误 | R(W) (注1) |
| b7-b6 | — | 保留位 | 读取值为不定值, 只能写“1”。 | R/W |

注 1. 只能写“0”。

SSR 寄存器由 SCI 的状态标志构成。

TEND 标志 (发送结束标志)

在没有接收侧的错误信号响应并且能将下一个发送数据传送到 TDR 寄存器的情况下, 此标志变为“1”。

[为“1”的条件]

- SCR.TE 位为“0” (禁止串行发送) 并且 ERS 标志为“0”时
- 在发送 1 字节数据后的一定时间后, ERS 标志变为“0”并且未更新 TDR 寄存器时
根据寄存器的设定, 变为“1”的时序如下:
当 SMR.GM 位和 SMR.BLK 位都为“0”时, 在开始发送后的 12.5etu。
当 SMR.GM 位为“0”并且 SMR.BLK 位为“1”时, 在开始发送后的 11.5etu。
当 SMR.GM 位为“1”并且 SMR.BLK 位为“0”时, 在开始发送后的 11.0etu。
当 SMR.GM 位和 SMR.BLK 位都为“1”时, 在开始发送后的 11.0etu。

[为“0”的条件]

- 给 TDR 寄存器写发送数据时

PER 标志 (奇偶校验错误标志)

此标志表示在异步模式中接收的数据发生了奇偶校验错误。

[为“1”的条件]

- 在接收时检测到奇偶校验错误时
将发生奇偶校验错误时的接收数据传送到RDR寄存器，但是不产生RXI中断请求。在PER标志为“1”的状态下，不能继续进行以后的串行接收。在时钟同步模式中，也不能继续进行串行发送。

[为“0”的条件]

- 读“1”后写“0”（必须在写“0”后确认PER标志已变为“0”）时
即使将SCR.RE位置“0”（禁止串行接收），PER标志也不受影响而保持原来的状态。

ERS 标志 (错误信号状态标志)

[为“1”的条件]

- 对错误信号的Low电平进行采样时

[为“0”的条件]

- 读“1”后写“0”时

ORER 标志 (溢出错误标志)

此标志表示接收的数据发生了溢出错误。

[为“1”的条件]

- 不读RDR寄存器的接收数据就接收到下一个数据时
RDR寄存器保持发生溢出错误前的接收数据而放弃后面的接收数据。在ORER标志为“1”的状态下，不能继续进行以后的串行接收。

[为“0”的条件]

- 读“1”后写“0”（必须在写“0”后确认ORER标志已变为“0”）时
即使将SCR.RE位置“0”，ORER标志也不受影响而保持原来的状态。

20.2.8 智能卡模式寄存器 (SCMR)

地址 SCI0.SCMR 0008 8246h、SCI1.SCMR 0008 824Eh、SCI2.SCMR 0008 8256h、SCI3.SCMR 0008 825Eh
SCI4.SCMR 0008 8266h、SCI5.SCMR 0008 826Eh、SCI6.SCMR 0008 8276h

| | | | | | | | | |
|-------|------|----|----|----|------|------|----|------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | BCP2 | — | — | — | SDIR | SINV | — | SMIF |
| 复位后的值 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|------|------------|--|-------------|
| b0 | SMIF | 智能卡接口模式选择位 | 0: 串行通信接口模式 1: 智能卡接口模式 | R/W (注1) |
| b1 | — | 保留位 | 读写值都为“1”。 | R/W |
| b2 | SINV | 智能卡数据反相位 | 0: 发送 TDR 寄存器的值、将接收数据保存到 RDR 寄存器 1: 将 TDR 寄存器的值取反后发送、将接收数据取反后保存到 RDR 寄存器 | R/W (注1) |
| b3 | SDIR | 位序选择位 | 0: 以 LSB first 进行发送和接收 1: 以 MSB first 进行发送和接收 | R/W (注1) |
| b6-b4 | — | 保留位 | 读写值都为“1”。 | R/W |
| b7 | BCP2 | 基本时钟脉冲位 2 | 通过和 SMR.BCP[1:0] 位组合进行选择。 SCMR.BCP2 位和 SMR.BCP[1:0] 位的设定值 BCP2 BCP1 BCP0 0 0 0: 93 个时钟 (S=93) (注2) 0 0 1: 128 个时钟 (S=128) (注2) 0 1 0: 186 个时钟 (S=186) (注2) 0 1 1: 512 个时钟 (S=512) (注2) 1 0 0: 32 个时钟 (S=32) (注2) (初始值) 1 0 1: 64 个时钟 (S=64) (注2) 1 1 0: 372 个时钟 (S=372) (注2) 1 1 1: 256 个时钟 (S=256) (注2) | R/W (注1) |

注 1. 只能在 SCR.TE 位和 SCR.RE 位都为“0”（禁止串行发送和串行接收）时写此位。

注 2. S 表示“20.2.9 位速率寄存器 (BRR)”中的 S 的值。

SCMR 寄存器是选择智能卡接口模式及其格式的寄存器。

SMIF 位 (智能卡接口模式选择位)

在以智能卡接口模式运行时, 将此位置“1”。

在以异步模式或者时钟同步模式运行时, 将此位置“0”。

SINV 位 (智能卡数据反相位)

将发送 / 接收数据的逻辑电平取反。SINV 位不影响奇偶校验位的逻辑电平。如果要将奇偶校验位取反, 就必须将 SMR.PM 位取反。

SDIR 位 (位序选择位)

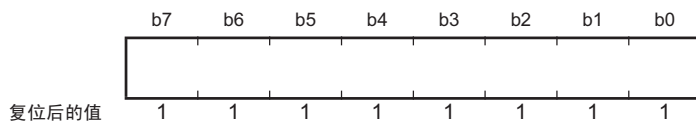
此位选择串行 / 并行转换的方向。

BCP2 位 (基本时钟脉冲位 2)

在智能卡接口模式中, 通过和 SMR.BCP[1:0] 位组合, 选择 1 位传送时间的基本时钟数。

20.2.9 位速率寄存器 (BRR)

地址 SCI0.BRR 0008 8241h、SCI1.BRR 0008 8249h、SCI2.BRR 0008 8251h、SCI3.BRR 0008 8259h
SCI4.BRR 0008 8261h、SCI5.BRR 0008 8269h、SCI6.BRR 0008 8271h



BRR 寄存器是调整位速率的 8 位寄存器。

因为 SCI 各通道的波特率发生器独立运行，所以能设定不同的位速率。在一般的异步模式、时钟同步模式和智能卡接口模式中，BRR 寄存器的设定值 N 和位速率 B 的关系如表 20.5 所示。

BRR 寄存器的初始值为“FFh”。

CPU 能随时读 BRR 寄存器，但是只能在 SCR.TE 位和 SCR.RE 位都为“0”时写此寄存器。

表 20.5 BRR 寄存器的设定值 N 和位速率 B 的关系

| 模式 | SEMR.ABCS位 | BRR寄存器的设定值 | 误差 |
|-------|------------|--|---|
| 异步 | 0 | $N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$ | 误差(%) = $\left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$ |
| | 1 | $N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$ | 误差(%) = $\left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$ |
| 时钟同步 | | $N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$ | |
| 智能卡接口 | | $N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$ | 误差(%) = $\left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$ |

B : 位速率 (Bjt/s)

N : 波特率发生器的 BRR 设定值 ($0 \leq N \leq 255$)

PCLK : 工作频率 (MHz)

n、S : 取决于下表中 SMR 的设定值。

| SMR 寄存器的设定值 | 时钟源 | n |
|-------------|------------|---|
| CKS[1:0] 位 | | |
| 0 0 | PCLK 时钟 | 0 |
| 0 1 | PCLK/4 时钟 | 1 |
| 1 0 | PCLK/16 时钟 | 2 |
| 1 1 | PCLK/64 时钟 | 3 |

| SCMR 寄存器的设定值 | SMR 寄存器的设定值 | 基本时钟 | S |
|--------------|-------------|---------|-----|
| BCP2 位 | BCP[1:0] 位 | | |
| 0 | 0 0 | 93 个时钟 | 93 |
| 0 | 0 1 | 128 个时钟 | 128 |
| 0 | 1 0 | 186 个时钟 | 186 |
| 0 | 1 1 | 512 个时钟 | 512 |
| 1 | 0 0 | 32 个时钟 | 32 |
| 1 | 0 1 | 64 个时钟 | 64 |
| 1 | 1 0 | 372 个时钟 | 372 |
| 1 | 1 1 | 256 个时钟 | 256 |

一般的异步模式中的 BRR 寄存器值 N 的设定例子如表 20.6 和表 20.7 所示，在各工作频率下能设定的最大位速率如表 20.8 所示；时钟同步模式和智能卡接口模式中的 BRR 寄存器值 N 的设定例子分别如表 20.11 和表 20.13 所示。在智能卡接口模式中，能选择 1 位传送时间的基本时钟数 S。详细内容请参照“20.5.4 接收数据的采样时序和接收容限”。输入外部时钟时的最大位速率如表 20.9、表 20.10 和表 20.12 所示。

在异步模式中，如果将串行扩展模式寄存器 (SEMR) 的异步基本时钟选择位 (ABCS) 置“1”，位速率就变为表 20.6 和表 20.7 所示的位速率的 2 倍。

表 20.6 对应位速率的 BRR 设定例子 (异步模式) (1)

| 位速率 (Bjt/s) | 工作频率 PCLK (MHz) | | | | | | | | | | | |
|----------------|-----------------|-----|--------|--------|-----|--------|----|-----|--------|----|-----|--------|
| | 8 | | | 9.8304 | | | 10 | | | 12 | | |
| | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) |
| 110 | 2 | 141 | 0.03 | 2 | 174 | -0.26 | 2 | 177 | -0.25 | 2 | 212 | 0.03 |
| 150 | 2 | 103 | 0.16 | 2 | 127 | 0.00 | 2 | 129 | 0.16 | 2 | 155 | 0.16 |
| 300 | 1 | 207 | 0.16 | 1 | 255 | 0.00 | 2 | 64 | 0.16 | 2 | 77 | 0.16 |
| 600 | 1 | 103 | 0.16 | 1 | 127 | 0.00 | 1 | 129 | 0.16 | 1 | 155 | 0.16 |
| 1200 | 0 | 207 | 0.16 | 0 | 255 | 0.00 | 1 | 64 | 0.16 | 1 | 77 | 0.16 |
| 2400 | 0 | 103 | 0.16 | 0 | 127 | 0.00 | 0 | 129 | 0.16 | 0 | 155 | 0.16 |
| 4800 | 0 | 51 | 0.16 | 0 | 63 | 0.00 | 0 | 64 | 0.16 | 0 | 77 | 0.16 |
| 9600 | 0 | 25 | 0.16 | 0 | 31 | 0.00 | 0 | 32 | -1.36 | 0 | 38 | 0.16 |
| 19200 | 0 | 12 | 0.16 | 0 | 15 | 0.00 | 0 | 15 | 1.73 | 0 | 19 | -2.34 |
| 31250 | 0 | 7 | 0.00 | 0 | 9 | -1.70 | 0 | 9 | 0.00 | 0 | 11 | 0.00 |
| 38400 | — | — | — | 0 | 7 | 0.00 | 0 | 7 | 1.73 | 0 | 9 | -2.34 |

| 位速率 (Bjt/s) | 工作频率 PCLK (MHz) | | | | | | | | |
|----------------|-----------------|-----|--------|----|-----|--------|----|-----|--------|
| | 12.288 | | | 14 | | | 16 | | |
| | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) |
| 110 | 2 | 217 | 0.08 | 2 | 248 | -0.17 | 3 | 70 | 0.03 |
| 150 | 2 | 159 | 0.00 | 2 | 181 | 0.16 | 2 | 207 | 0.16 |
| 300 | 2 | 79 | 0.00 | 2 | 90 | 0.16 | 2 | 103 | 0.16 |
| 600 | 1 | 159 | 0.00 | 1 | 181 | 0.16 | 1 | 207 | 0.16 |
| 1200 | 1 | 79 | 0.00 | 1 | 90 | 0.16 | 1 | 103 | 0.16 |
| 2400 | 0 | 159 | 0.00 | 0 | 181 | 0.16 | 0 | 207 | 0.16 |
| 4800 | 0 | 79 | 0.00 | 0 | 90 | 0.16 | 0 | 103 | 0.16 |
| 9600 | 0 | 39 | 0.00 | 0 | 45 | -0.93 | 0 | 51 | 0.16 |
| 19200 | 0 | 19 | 0.00 | 0 | 22 | -0.93 | 0 | 25 | 0.16 |
| 31250 | 0 | 11 | 2.40 | 0 | 13 | 0.00 | 0 | 15 | 0.00 |
| 38400 | 0 | 9 | 0.00 | — | — | — | 0 | 12 | 0.16 |

这是 SEMR.ABCS 位为 “0” 时的例子。

如果将 ABCS 位置 “1”，位速率就变为原来的 2 倍。

表 20.7 对应位速率的 BRR 设定例子 (异步模式) (2)

| 位速率 (Bjt/s) | 工作频率 PCLK (MHz) | | | | | | | | | | | |
|----------------|-----------------|-----|--------|----|-----|--------|---------|-----|--------|----|-----|--------|
| | 17.2032 | | | 18 | | | 19.6608 | | | 20 | | |
| | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) |
| 110 | 3 | 75 | 0.48 | 3 | 79 | -0.12 | 3 | 86 | 0.31 | 3 | 88 | -0.25 |
| 150 | 2 | 223 | 0.00 | 2 | 233 | 0.16 | 2 | 255 | 0.00 | 3 | 64 | 0.16 |
| 300 | 2 | 111 | 0.00 | 2 | 116 | 0.16 | 2 | 127 | 0.00 | 2 | 129 | 0.16 |
| 600 | 1 | 223 | 0.00 | 1 | 233 | 0.16 | 1 | 255 | 0.00 | 2 | 64 | 0.16 |
| 1200 | 1 | 111 | 0.00 | 1 | 116 | 0.16 | 1 | 127 | 0.00 | 1 | 129 | 0.16 |
| 2400 | 0 | 223 | 0.00 | 0 | 233 | 0.16 | 0 | 255 | 0.00 | 1 | 64 | 0.16 |
| 4800 | 0 | 111 | 0.00 | 0 | 116 | 0.16 | 0 | 127 | 0.00 | 0 | 129 | 0.16 |
| 9600 | 0 | 55 | 0.00 | 0 | 58 | -0.69 | 0 | 63 | 0.00 | 0 | 64 | 0.16 |
| 19200 | 0 | 27 | 0.00 | 0 | 28 | 1.02 | 0 | 31 | 0.00 | 0 | 32 | -1.36 |
| 31250 | 0 | 16 | 1.20 | 0 | 17 | 0.00 | 0 | 19 | -1.70 | 0 | 19 | 0.00 |
| 38400 | 0 | 13 | 0.00 | 0 | 14 | -2.34 | 0 | 15 | 0.00 | 0 | 15 | 1.73 |

| 位速率 (Bjt/s) | 工作频率 PCLK (MHz) | | | | | | | | | | | |
|----------------|-----------------|-----|--------|----|-----|--------|----|-----|--------|----|-----|--------|
| | 25 | | | 30 | | | 33 | | | 50 | | |
| | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) |
| 110 | 3 | 110 | -0.02 | 3 | 132 | 0.13 | 3 | 145 | 0.33 | 3 | 221 | -0.02 |
| 150 | 3 | 80 | 0.47 | 3 | 97 | -0.35 | 3 | 106 | 0.39 | 3 | 162 | -0.15 |
| 300 | 2 | 162 | -0.15 | 2 | 194 | 0.16 | 2 | 214 | -0.07 | 3 | 80 | 0.47 |
| 600 | 2 | 80 | 0.47 | 2 | 97 | -0.35 | 2 | 106 | 0.39 | 2 | 162 | -0.15 |
| 1200 | 1 | 162 | -0.15 | 1 | 194 | 0.16 | 1 | 214 | -0.07 | 2 | 80 | 0.47 |
| 2400 | 1 | 80 | 0.47 | 1 | 97 | -0.35 | 1 | 106 | 0.39 | 1 | 162 | -0.15 |
| 4800 | 0 | 162 | -0.15 | 0 | 194 | 0.16 | 0 | 214 | -0.07 | 1 | 80 | 0.47 |
| 9600 | 0 | 80 | 0.47 | 0 | 97 | -0.35 | 0 | 106 | 0.39 | 1 | 40 | -0.77 |
| 19200 | 0 | 40 | -0.76 | 0 | 48 | -0.35 | 0 | 53 | -0.54 | 0 | 80 | 0.47 |
| 31250 | 0 | 24 | 0.00 | 0 | 29 | 0 | 0 | 32 | 0 | 0 | 49 | 0.00 |
| 38400 | 0 | 19 | 1.73 | 0 | 23 | 1.73 | 0 | 26 | -0.54 | 0 | 40 | -0.77 |

这是 SEMR.ABCS 位为 “0” 时的例子。

如果将 ABCS 位置 “1”，位速率就变为原来的 2 倍。

表 20.8 各工作频率下的最大位速率 (异步模式)

| PCLK (MHz) | 最大位速率 (Bjt/s) | n | N | PCLK (MHz) | 最大位速率 (Bjt/s) | n | N |
|------------|---------------|---|---|------------|---------------|---|---|
| 8 | 250000 | 0 | 0 | 18 | 562500 | 0 | 0 |
| 9.8304 | 307200 | 0 | 0 | 19.6608 | 614400 | 0 | 0 |
| 10 | 312500 | 0 | 0 | 20 | 625000 | 0 | 0 |
| 12 | 375000 | 0 | 0 | 25 | 781250 | 0 | 0 |
| 12.288 | 384000 | 0 | 0 | 30 | 937500 | 0 | 0 |
| 14 | 437500 | 0 | 0 | 33 | 1031250 | 0 | 0 |
| 16 | 500000 | 0 | 0 | 50 | 1562500 | 0 | 0 |
| 17.2032 | 537600 | 0 | 0 | | | | |

如果将 SEMR.ABCS 位置 “1”，位速率就变为原来的 2 倍。

表 20.9 输入外部时钟时的最大位速率 (异步模式) (1)

| PCLK (MHz) | 外部输入时钟 (MHz) | 最大位速率 (Bjt/s) | PCLK (MHz) | 外部输入时钟 (MHz) | 最大位速率 (Bjt/s) |
|------------|--------------|---------------|------------|--------------|---------------|
| 8 | 2.0000 | 125000 | 18 | 4.5000 | 281250 |
| 9.8304 | 2.4576 | 153600 | 19.6608 | 4.9152 | 307200 |
| 10 | 2.5000 | 156250 | 20 | 5.0000 | 312500 |
| 12 | 3.0000 | 187500 | 25 | 6.2500 | 390625 |
| 12.288 | 3.0720 | 192000 | 30 | 7.5000 | 468750 |
| 14 | 3.5000 | 218750 | 33 | 8.2500 | 515625 |
| 16 | 4.0000 | 250000 | 50 | 12.5000 | 781250 |
| 17.2032 | 4.3008 | 268800 | | | |

这是 SEMR.ABCS 位为 “0” 时的例子。

表 20.10 输入外部时钟时的最大位速率 (异步模式) (2)

| PCLK (MHz) | 外部输入时钟 (MHz) | 最大位速率 (Bjt/s) | PCLK (MHz) | 外部输入时钟 (MHz) | 最大位速率 (Bjt/s) |
|------------|--------------|---------------|------------|--------------|---------------|
| 8 | 2.0000 | 250000 | 18 | 4.5000 | 562500 |
| 9.8304 | 2.4576 | 307200 | 19.6608 | 4.9152 | 614400 |
| 10 | 2.5000 | 312500 | 20 | 5.0000 | 625000 |
| 12 | 3.0000 | 375000 | 25 | 6.2500 | 781250 |
| 12.288 | 3.0720 | 384000 | 30 | 7.5000 | 937500 |
| 14 | 3.5000 | 437500 | 33 | 8.2500 | 1031250 |
| 16 | 4.0000 | 500000 | 50 | 12.5000 | 1562500 |
| 17.2032 | 4.3008 | 537600 | | | |

这是 SEMR.ABCS 位为 “1” 时的例子。

表 20.11 对应位速率的 BRR 设定例子 (时钟同步模式)

| 位速率 (Bjt/s) | 工作频率 PCLK (MHz) | | | | | | | | | | | | | | | |
|-------------|-----------------|-----|----|--------|----|-----|----|--------|----|-----|----|-----|----|-----|----|-----|
| | 8 | | 10 | | 16 | | 20 | | 25 | | 30 | | 33 | | 50 | |
| | n | N | n | N | n | N | n | N | n | N | n | N | n | N | n | N |
| 110 | | | | | | | | | | | | | | | | |
| 250 | 3 | 124 | — | — | 3 | 249 | | | | | | | | | | |
| 500 | 2 | 249 | — | — | 3 | 124 | — | — | | | 3 | 233 | | | | |
| 1k | 2 | 124 | — | — | 2 | 249 | — | — | 3 | 97 | 3 | 116 | 3 | 128 | 3 | 194 |
| 2.5k | 1 | 199 | 1 | 249 | 2 | 99 | 2 | 124 | 2 | 155 | 2 | 187 | 2 | 205 | 3 | 77 |
| 5k | 1 | 99 | 1 | 124 | 1 | 199 | 1 | 249 | 2 | 77 | 2 | 93 | 2 | 102 | 2 | 155 |
| 10k | 0 | 199 | 0 | 249 | 1 | 99 | 1 | 124 | 1 | 155 | 1 | 187 | 1 | 205 | 2 | 77 |
| 25k | 0 | 79 | 0 | 99 | 0 | 159 | 0 | 199 | 0 | 249 | 1 | 74 | 1 | 82 | 1 | 124 |
| 50k | 0 | 39 | 0 | 49 | 0 | 79 | 0 | 99 | 0 | 124 | 0 | 149 | 0 | 164 | 1 | 61 |
| 100k | 0 | 19 | 0 | 24 | 0 | 39 | 0 | 49 | 0 | 62 | 0 | 74 | 0 | 82 | 0 | 124 |
| 250k | 0 | 7 | 0 | 9 | 0 | 15 | 0 | 19 | 0 | 24 | 0 | 29 | 0 | 32 | 0 | 49 |
| 500k | 0 | 3 | 0 | 4 | 0 | 7 | 0 | 9 | — | — | 0 | 14 | — | — | 0 | 24 |
| 1M | 0 | 1 | | | 0 | 3 | 0 | 4 | — | — | — | — | — | — | — | — |
| 2.5M | | | 0 | 0 (注1) | | | 0 | 1 | — | — | 0 | 2 | — | — | 0 | 4 |
| 5M | | | | | | | 0 | 0 (注1) | — | — | — | — | — | — | — | — |

空栏: 不能设定。

— : 能设定, 但是会产生误差。

注 1. 不能进行连续的发送和接收。

表 20.12 输入外部时钟时的最大位速率 (时钟同步模式)

| PCLK (MHz) | 外部输入时钟 (MHz) | 最大位速率 (Bjt/s) | PCLK (MHz) | 外部输入时钟 (MHz) | 最大位速率 (Bjt/s) |
|------------|--------------|---------------|------------|--------------|---------------|
| 8 | 1.3333 | 1333333.3 | 20 | 3.3333 | 3333333.3 |
| 10 | 1.6667 | 1666666.7 | 25 | 4.1667 | 4166666.7 |
| 12 | 2.0000 | 2000000.0 | 30 | 5.0000 | 5000000.0 |
| 14 | 2.3333 | 2333333.3 | 33 | 5.5000 | 5500000.0 |
| 16 | 2.6667 | 2666666.7 | 50 | 8.3333 | 8333333.3 |
| 18 | 3.0000 | 3000000.0 | | | |

表 20.13 对应位速率的 BRR 设定例子 (在智能卡接口模式中, n 为 “0” 并且 S 为 “372” 的情况)

| 位速率 (Bjt/s) | 工作频率 PCLK (MHz) | | | | | | | | | | | |
|----------------|-----------------|---|--------|-------|---|--------|---------|---|--------|-------|---|--------|
| | 7.1424 | | | 10.00 | | | 10.7136 | | | 13.00 | | |
| | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) |
| 9600 | 0 | 0 | 0.00 | 0 | 1 | 30 | 0 | 1 | 25 | 0 | 1 | 8.99 |

| 位速率 (Bjt/s) | 工作频率 PCLK (MHz) | | | | | | | | | | | |
|----------------|-----------------|---|--------|-------|---|--------|-------|---|--------|-------|---|--------|
| | 14.2848 | | | 16.00 | | | 18.00 | | | 20.00 | | |
| | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) |
| 9600 | 0 | 1 | 0.00 | 0 | 1 | 12.01 | 0 | 2 | 15.99 | 0 | 2 | 6.66 |

| 位速率 (Bjt/s) | 工作频率 PCLK (MHz) | | | | | | | | | | | |
|----------------|-----------------|---|--------|-------|---|--------|-------|---|--------|-------|----|--------|
| | 25.00 | | | 30.00 | | | 33.00 | | | 50.00 | | |
| | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) | n | N | 误差 (%) |
| 9600 | 0 | 3 | 12.49 | 0 | 3 | 5.01 | 0 | 4 | 7.59 | 0 | 27 | 0.00 |

表 20.14 各工作频率下的最大位速率 (在智能卡接口模式中并且 S 为 “372” 的情况)

| PCLK (MHz) | 最大位速率 (Bjt/s) | n | N | PCLK (MHz) | 最大位速率 (Bjt/s) | n | N |
|---------------|------------------|---|---|---------------|------------------|---|---|
| 10.00 | 13441 | 0 | 0 | 20.00 | 26882 | 0 | 0 |
| 10.7136 | 14400 | 0 | 0 | 25.00 | 33602 | 0 | 0 |
| 13.00 | 17473 | 0 | 0 | 30.00 | 40323 | 0 | 0 |
| 16.00 | 21505 | 0 | 0 | 33.00 | 44355 | 0 | 0 |
| 18.00 | 24194 | 0 | 0 | 50.00 | 67205 | 0 | 0 |

20.2.10 串行扩展模式寄存器 (SEMR)

地址 SCI0.SEMR 0008 8247h、SCI1.SEMR 0008 824Fh、SCI2.SEMR 0008 8257h、SCI3.SEMR 0008 825Fh
SCI4.SEMR 0008 8267h、SCI5.SEMR 0008 826Fh、SCI6.SEMR 0008 8277h

| | | | | | | | |
|----|----|----|------|----|----|----|------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | ABCS | — | — | — | ACS0 |

复位后的值 0 0 0 0 0 0 0 0

| 位 | 符号 | 位名 | 功能 | R/W | | | | | | | | | |
|-------|------|-----------|--|--------------|-----|--------|------|------|-----------|------|------|-----------|--------------|
| b0 | ACS0 | 异步时钟源选择位 | (只在异步模式中有效) 0: 外部时钟的输入 (SCI0 ~ SCI6) 1: TMR 时钟的输入 (只对 SCI5 和 SCI6 有效) SCI 通道和比较匹配输出的对应如下所示: <table border="1" style="margin: 10px auto;"> <thead> <tr> <th>SCI</th> <th>TMR</th> <th>比较匹配输出</th> </tr> </thead> <tbody> <tr> <td>SCI5</td> <td>单元 0</td> <td>TMO0、TMO1</td> </tr> <tr> <td>SCI6</td> <td>单元 1</td> <td>TMO2、TMO3</td> </tr> </tbody> </table> | SCI | TMR | 比较匹配输出 | SCI5 | 单元 0 | TMO0、TMO1 | SCI6 | 单元 1 | TMO2、TMO3 | R/W (注 1) |
| SCI | TMR | 比较匹配输出 | | | | | | | | | | | |
| SCI5 | 单元 0 | TMO0、TMO1 | | | | | | | | | | | |
| SCI6 | 单元 1 | TMO2、TMO3 | | | | | | | | | | | |
| b3-b1 | — | 保留位 | 读写值都为“0”。 | R/W | | | | | | | | | |
| b4 | ABCS | 异步基本时钟选择位 | (只在异步模式中有效) 0: 16 个基本时钟周期的时间为 1 位时间的传送率 1: 8 个基本时钟周期的时间为 1 位时间的传送率 | R/W (注 1) | | | | | | | | | |
| b7-b5 | — | 保留位 | 读写值都为“0”。 | R/W | | | | | | | | | |

注 1. 只能在 SCR.TE 位和 SCR.RE 位都为“0” (禁止串行发送和串行接收) 时写此位。

SEMR 寄存器选择异步模式中的 1 位时间的时钟。

对于 SCI5 和 SCI6, 能将 TMR 单元 0 和单元 1 的 TMO_n (n=0 ~ 3) 输出设定为串行发送 / 接收的基本时钟。

选择 TMR_m (m=0 ~ 3) 的 TMO_n 输出时的设定例子如图 20.3 所示。

ACS0 位 (异步时钟源选择位)

此位选择异步模式中的时钟源。

ACS0 位在异步模式 (SMR.CM 位 = 0) 中并且输入外部时钟 (SCR.CKE[1:0] 位 = 10b, 11b) 时有效。能选择外部时钟输入或者内部 TMR 时钟输入。

在 SCI5 和 SCI6 以外时, 此位为保留位, 只能写“0”。

ABCS 位 (异步基本时钟选择位)

此位选择 1 位时间的基本时钟的脉冲数。

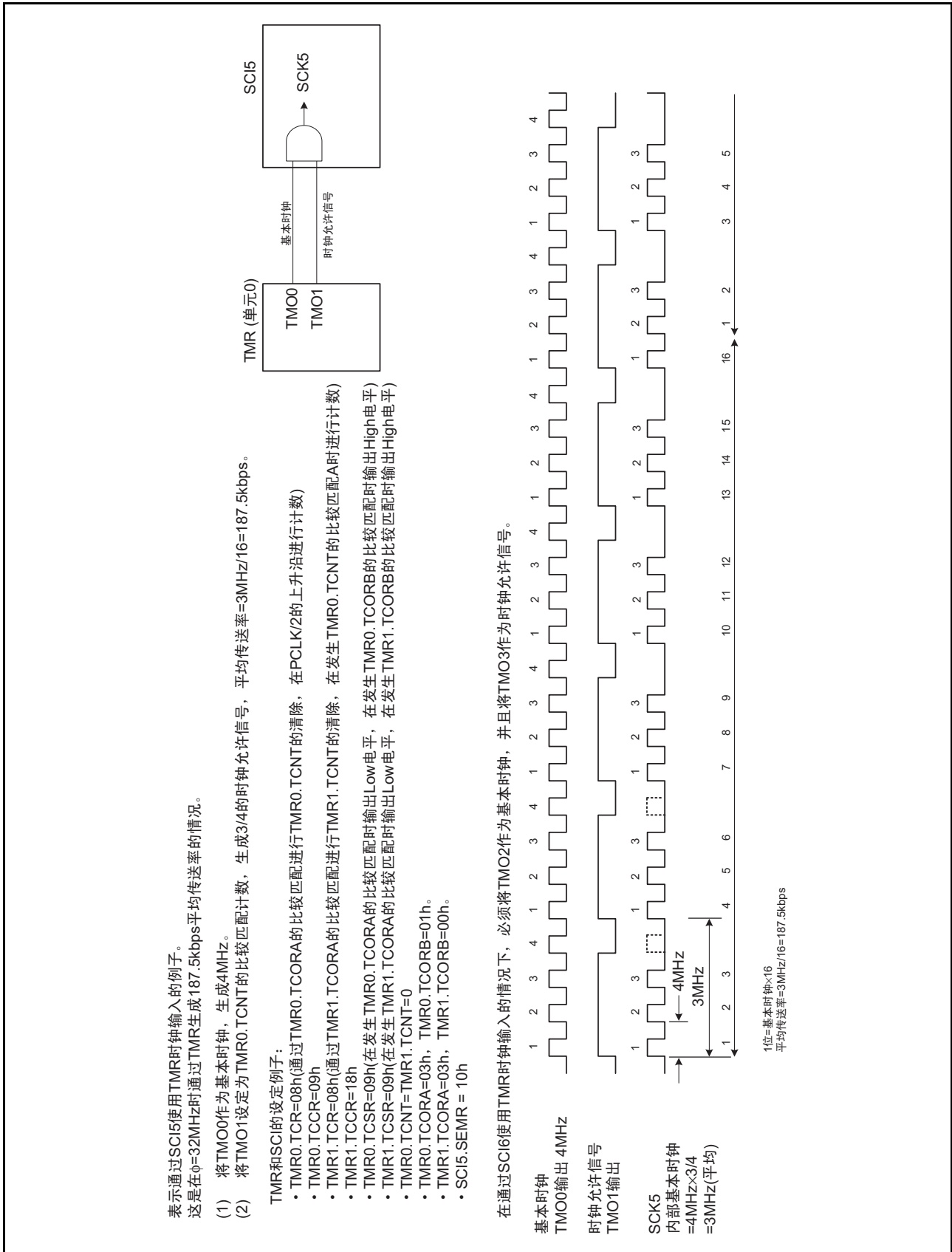


图 20.3 输入 TMR 时钟时的平均传送率的设定例子

20.3 异步模式的运行

异步串行通信的一般数据格式如图 20.4 所示。

按照起始位 (Low 电平)、发送 / 接收数据、奇偶校验位、停止位 (High 电平) 的顺序构成 1 帧。

在异步串行通信模式中, 通信线路通常保持标记状态 (High 电平)。

SCI 监视通信线路, 如果检测到 Low 电平, 就开始串行通信。

SCI 内部的发送部和接收部各自独立, 因此能进行全双工通信。因为发送部和接收部都为双缓冲结构, 所以能在发送和接收时读写数据, 也能进行连续的发送和接收。

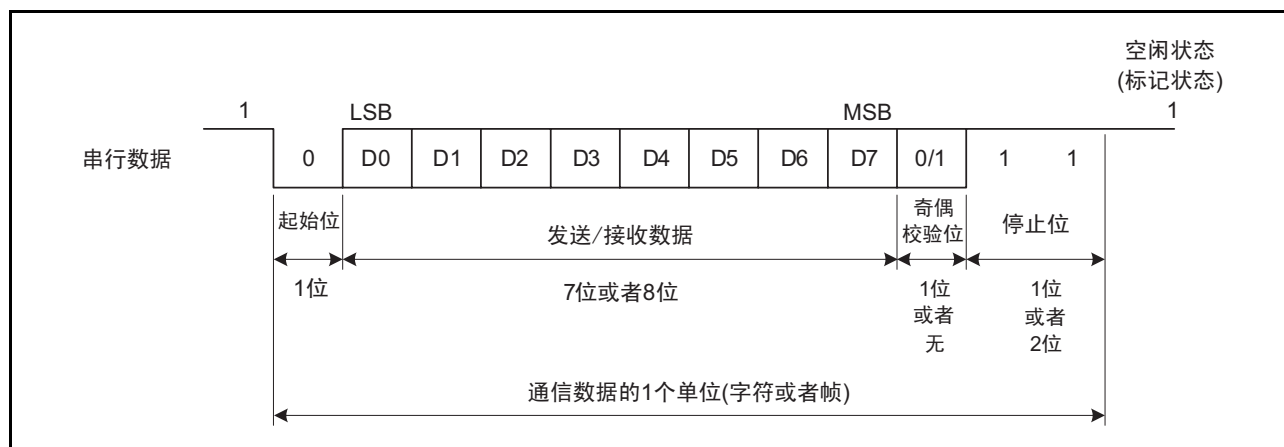


图 20.4 异步串行通信的数据格式 (8 位数据 / 有奇偶校验 / 2 个停止位的例子)

20.3.1 串行发送 / 接收格式

异步模式中能设定的串行发送 / 接收格式如表 20.15 所示。

串行发送 / 接收格式有 8 种，能通过设定 SMR 寄存器进行选择。

表 20.15 串行发送 / 接收格式 (异步模式)

| SMR 寄存器的设定 | | | 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 | S | 8位数据 | | | | | | | | STOP | | | | |
| 0 | 0 | 1 | S | 8位数据 | | | | | | | | STOP | STOP | | | |
| 0 | 1 | 0 | S | 8位数据 | | | | | | | | P | STOP | | | |
| 0 | 1 | 1 | S | 8位数据 | | | | | | | | P | STOP | STOP | | |
| 1 | 0 | 0 | S | 7位数据 | | | | | | | STOP | | | | | |
| 1 | 0 | 1 | S | 7位数据 | | | | | | | STOP | STOP | | | | |
| 1 | 1 | 0 | S | 7位数据 | | | | | | | P | STOP | | | | |
| 1 | 1 | 1 | S | 7位数据 | | | | | | | P | STOP | STOP | | | |

S : 起始位

STOP : 停止位

P : 奇偶校验位

20.3.2 异步模式的接收数据采样时序和接收容限

在异步模式中，SCI 通过频率为 16 倍位速率（注 1）的基本时钟运行。

在接收时，通过基本时钟对起始位的下降沿进行采样，并且与内部取得同步。如图 20.5 所示，通过在第 8 个时钟（注 1）的上升沿对接收数据进行采样，在各位的中央取数据。因此，能用表达式（1）表示异步模式的接收容限。

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100[\%] \quad \dots \text{表达式(1)}$$

M : 接收容限

N : 对应时钟的位速率比

（当 SEMR.ABCS 位为 “0” 时，N=16；当 ABCS 位为 “1” 时，N=8）

D : 时钟的占空比 (D=0.5 ~ 1.0)

L : 帧长 (L=9 ~ 12)

F : 时钟频率的偏差绝对值

假设表达式（1）中的 F（时钟频率的偏差绝对值）为 “0”，D（时钟的占空比）为 “0.5”，则

$$M = \{ 0.5 - 1 / (2 \times 16) \} \times 100 (\%) = 46.875\%$$

但是，此值毕竟是计算值，在进行系统设计时，必须留出 20 ~ 30% 的容限。

注 1. 这是 SEMR.ABCS 位为 “0” 时的例子。当 ABCS 位为 “1” 时，基本时钟的频率为位速率的 8 倍，在基本时钟的第 4 个上升沿对接收数据进行采样。

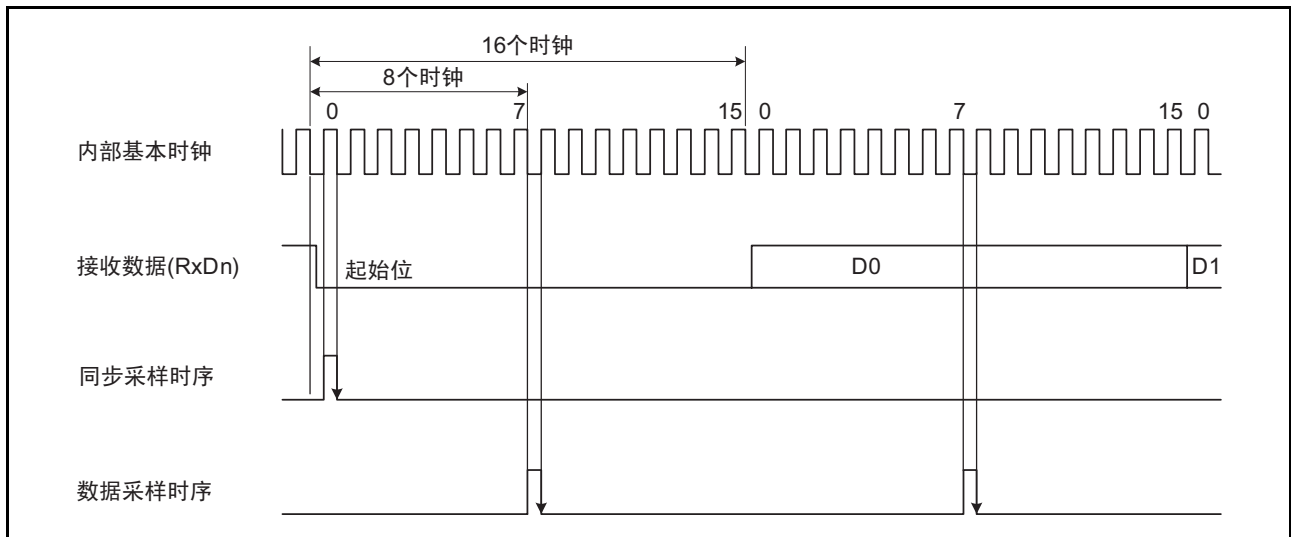


图 20.5 异步模式的接收数据采样时序

20.3.3 时钟

能通过设定 `SMR.CM` 位和 `SCR.CKE[1:0]` 位, 选择内部波特率发生器生成的内部时钟或者 `SCKn` 引脚输入的外部时钟作为 SCI 的发送 / 接收时钟。

在使用外部时钟的情况下, 必须将频率为 16 倍位速率 (`SEMR.ABCS` 位 =0) 或者频率为 8 倍位速率 (`SEMR.ABCS` 位 =1) 的时钟输入到 `SCKn` 引脚。如果选择外部时钟, 就能通过设定 `SCIm.SEMR.ACS0` 位 ($m=5,6$), 选择 `TMR0`、`TMR1` 的基本时钟。

在通过内部时钟运行时, 能从 `SCKn` 引脚输出时钟。此时, 输出时钟的频率和位速率相等, 发送时的相位如图 20.6 所示, 时钟在发送数据的中央上升。

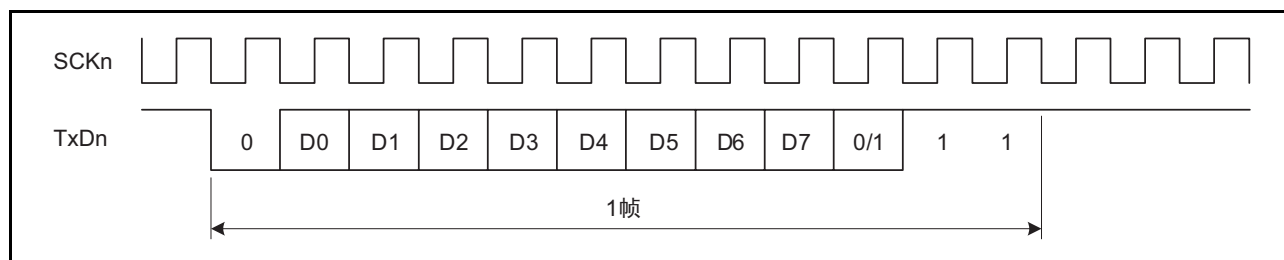


图 20.6 输出时钟和发送数据的相位关系 (异步模式)

20.3.4 SCI 的初始化 (异步模式)

必须在发送和接收数据前给 SCR 寄存器写初始值 “00h”，并且根据图 20.7 的流程图例子对 SCI 进行初始化。必须在对 SCR 寄存器进行初始化后更改运行模式或者通信格式。

在异步模式中使用外部时钟的情况下，必须提供时钟（包括初始化期间）。

必须注意：即使将 SCR.RE 位置 “0”，也不对 SSR.ORER 标志、SSR.FER 标志、SSR.PER 标志和 RDR 寄存器进行初始化。

必须注意：如果将 SCR.TE 位从 “1” 变为 “0” 或者从 “0” 变为 “1”，就在 SCR.TIE 位为 “1” 时产生 TXI 中断。

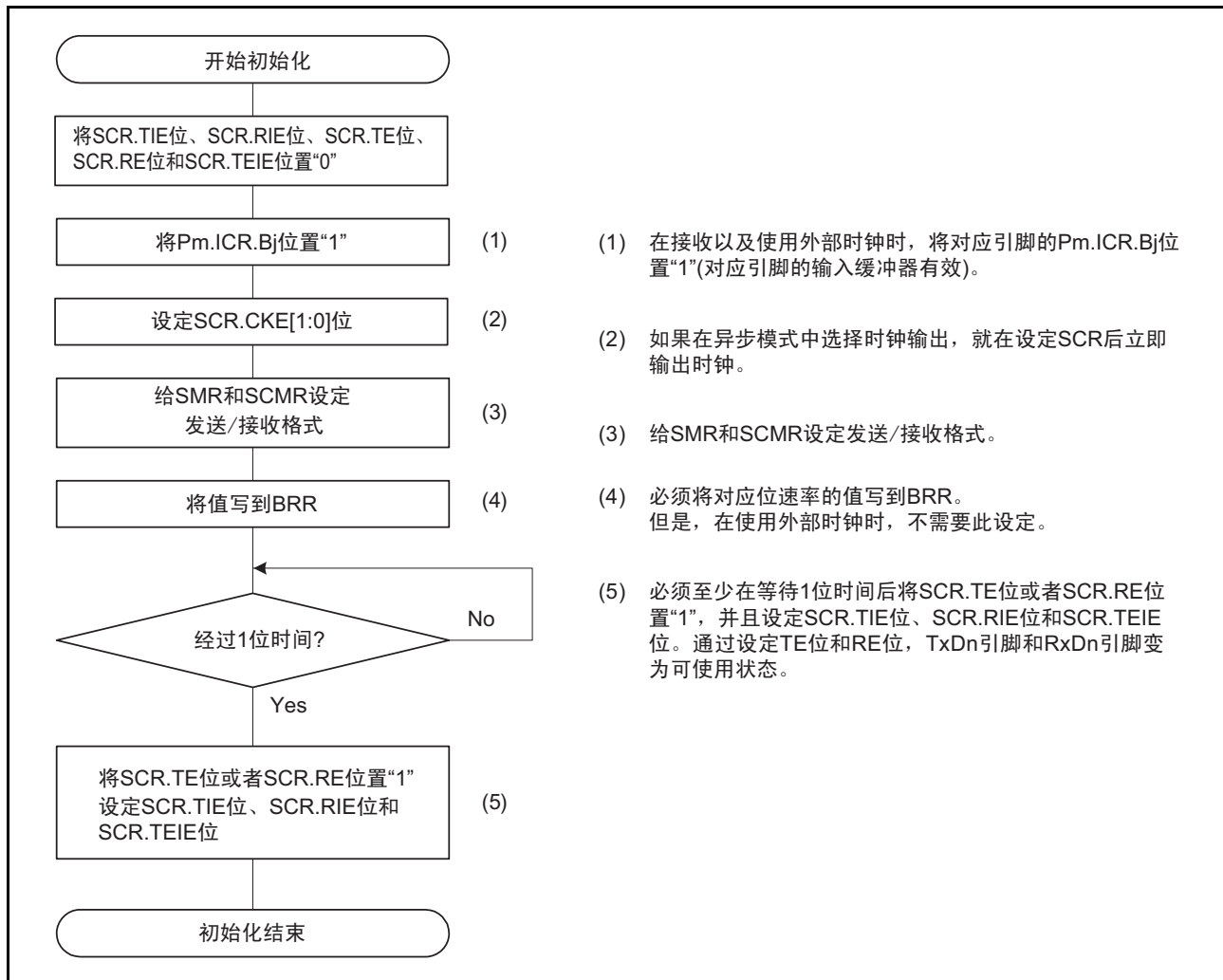


图 20.7 SCI 的初始化流程图例子 (异步模式)

20.3.5 串行数据的发送 (异步模式)

异步模式的串行发送例子如图 20.8 所示。

在发送串行数据时，SCI 的运行如下：

1. 如果通过TXI中断处理程序给TDR寄存器写数据，SCI就将数据从TDR寄存器传送到TSR寄存器。在开始发送时，通过将SCR.TIE位置“1”后将SCR.TE位置“1”，或者同时将这2个位置“1”，产生TXI中断请求。
2. 通过将数据从TDR寄存器传送到TSR寄存器，开始发送。此时，如果SCR.TIE位为“1”，就产生TXI中断请求。能通过TXI中断处理程序，在上次传送的数据发送结束前给TDR寄存器写下一个发送数据，进行连续的发送。
3. 从TxDn引脚依次发送起始位、发送数据、奇偶校验位（根据格式，有时没有此位）和停止位。
4. 在发送停止位时检查TDR寄存器的更新（写）。
5. 如果TDR寄存器已被更新，就将下一个发送数据从TDR寄存器传送到TSR寄存器，并且在发送停止位后开始下一帧的发送。
6. 如果TDR寄存器未被更新，就在将SSR.TEND标志置“1”并且发送停止位后，通过输出High电平进入标记状态。当SCR.TEIE位为“1”时，就在SSR.TEND标志变为“1”的同时产生TEI中断请求。

串行发送的流程图例子如图 20.9 所示。

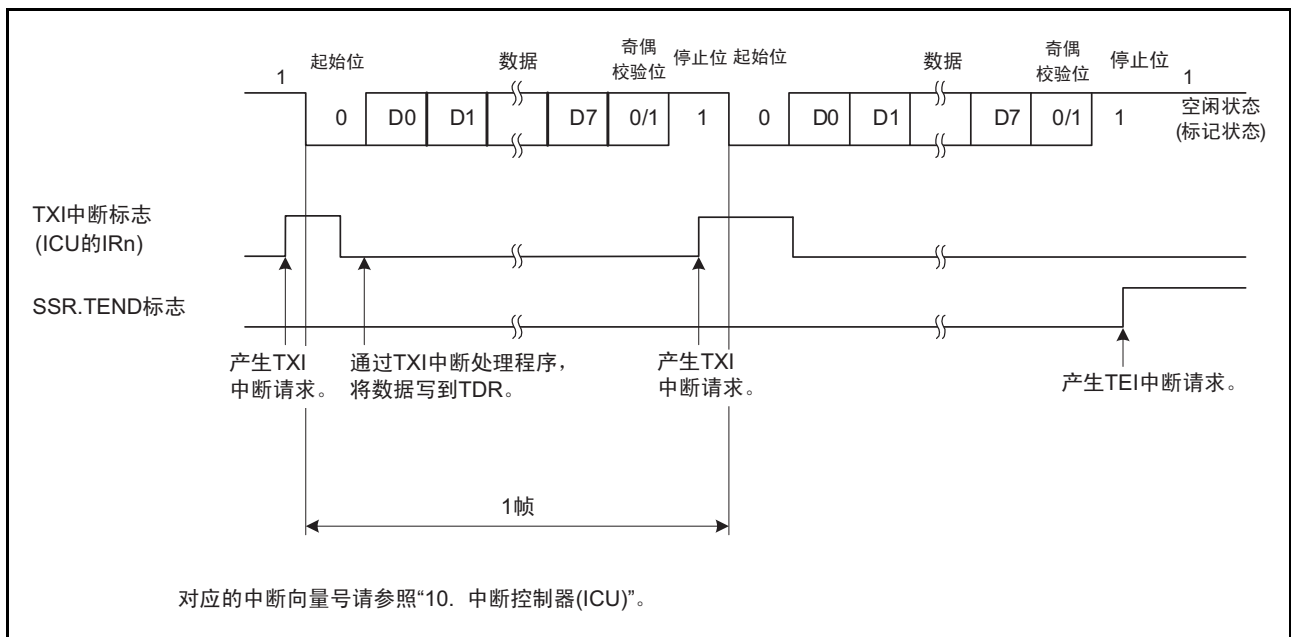


图 20.8 异步模式的串行发送例子 (8 位数据 / 有奇偶校验 / 1 个停止位的例子)

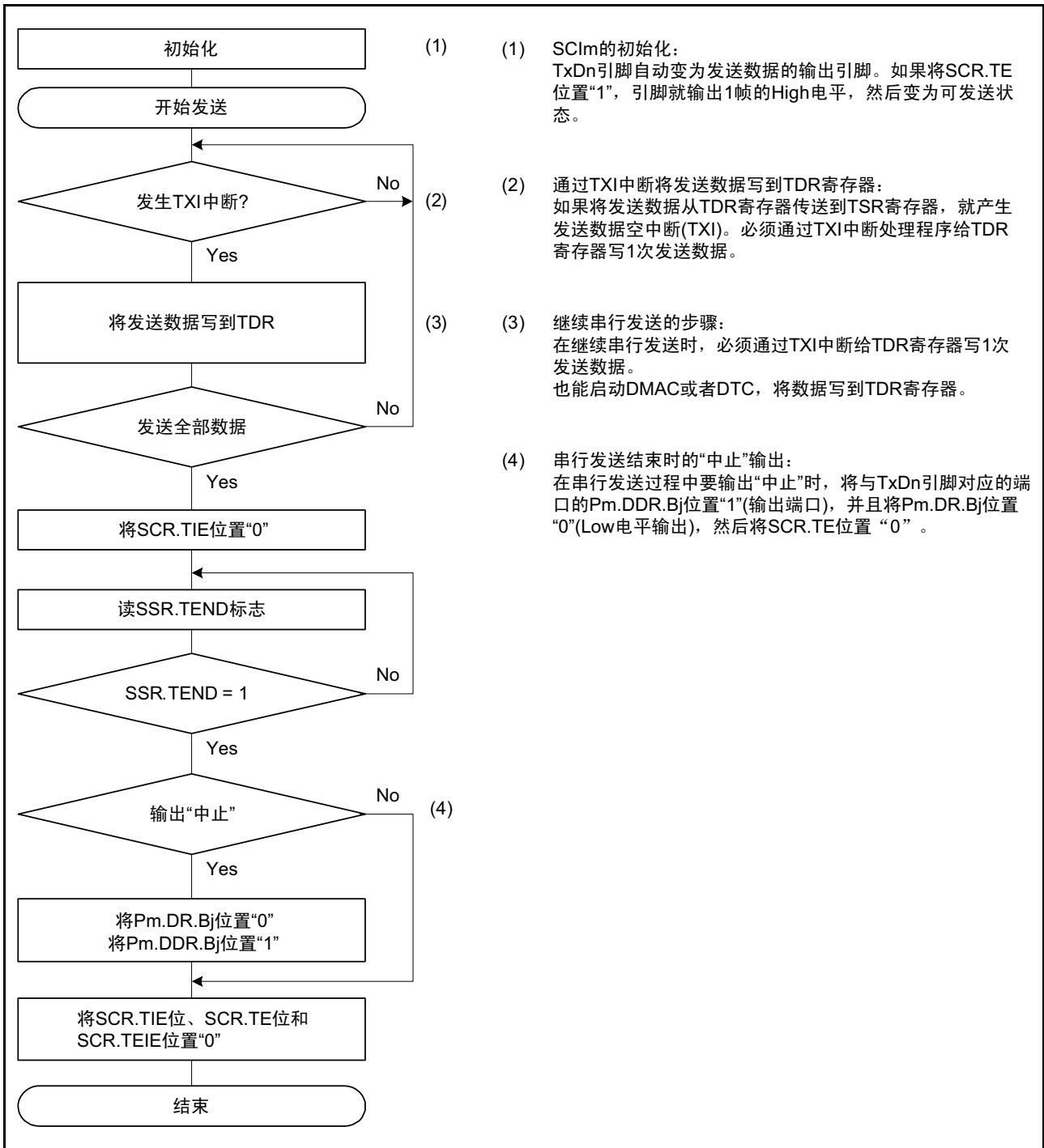


图 20.9 异步模式的串行发送的流程图例子

20.3.6 串行数据的接收 (异步模式)

异步模式的串行接收例子如图 20.10 所示。

在接收串行数据时，SCI 的运行如下：

1. 监视通信线路，如果检测到起始位，就与内部同步将接收数据取到 RSR 寄存器，并且检查奇偶校验位和停止位。
2. 如果发生溢出错误，SSR.ORER 标志就变为“1”。此时，如果 SCR.RIE 位为“1”，就产生 ERI 中断请求。不将接收数据传送到 RDR 寄存器。
3. 如果检测到奇偶校验错误，SSR.PER 标志就变为“1”，并且将接收数据传送到 RDR 寄存器。此时，如果 RIE 位为“1”，就产生 ERI 中断请求。
4. 如果检测到帧错误（停止位为“0”），SSR.FER 标志就变为“1”，并且将接收数据传送到 RDR 寄存器。此时，如果 RIE 位为“1”，就产生 ERI 中断请求。
5. 如果正常接收，就将接收数据传送到 RDR 寄存器。此时，如果 RIE 位为“1”，就产生 RXI 中断请求。能通过此 RXI 中断处理程序，在下一个数据接收结束前读已被传送到 RDR 寄存器的接收数据，进行连续接收。

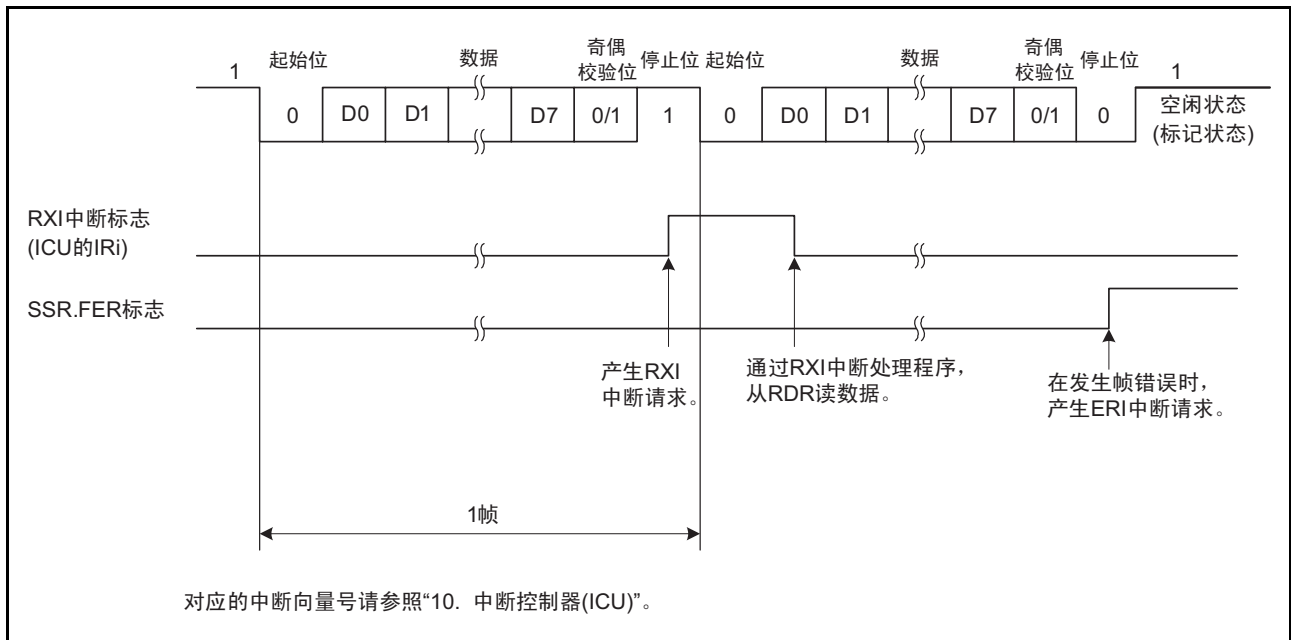


图 20.10 异步模式的串行接收例子 (8 位数据 / 有奇偶校验 / 1 个停止位的例子)

检测到接收错误时的 SSR 寄存器各状态标志的状态和接收数据的处理如表 20.16 所示。

一旦检测到接收错误，就产生 ERI 中断请求而不产生 RXI 中断请求。在接收错误标志为“1”的状态下不能进行以后的接收。因此，必须在继续接收前将 ORER 标志、FER 标志和 PER 标志置“0”。另外，在进行溢出错误处理时，必须读 RDR 寄存器。

串行接收的流程图例子如图 20.11 和图 20.12 所示。

表 20.16 SSR 寄存器状态标志的状态和接收数据的处理

| SSR 寄存器的状态标志 | | | 接收数据 | 接收错误的状态 |
|--------------|-----|-----|---------|---------------------|
| ORER | FER | PER | | |
| 1 | 0 | 0 | 消失 | 溢出错误 |
| 0 | 1 | 0 | 传送到 RDR | 帧错误 |
| 0 | 0 | 1 | 传送到 RDR | 奇偶校验错误 |
| 1 | 1 | 0 | 消失 | 溢出错误 + 帧错误 |
| 1 | 0 | 1 | 消失 | 溢出错误 + 奇偶校验错误 |
| 0 | 1 | 1 | 传送到 RDR | 帧错误 + 奇偶校验错误 |
| 1 | 1 | 1 | 消失 | 溢出错误 + 帧错误 + 奇偶校验错误 |

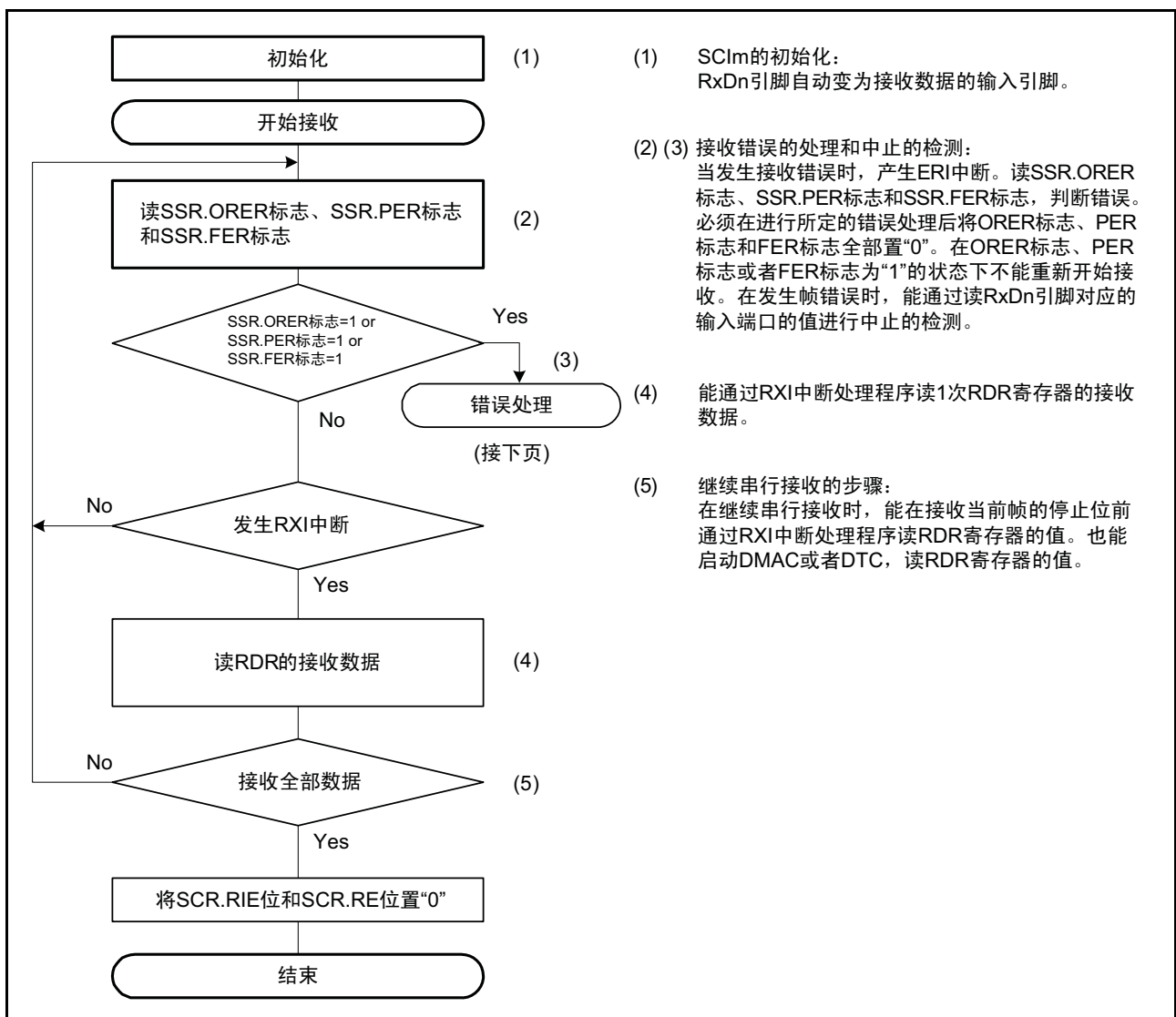


图 20.11 异步模式的串行接收的流程图例子 (1)

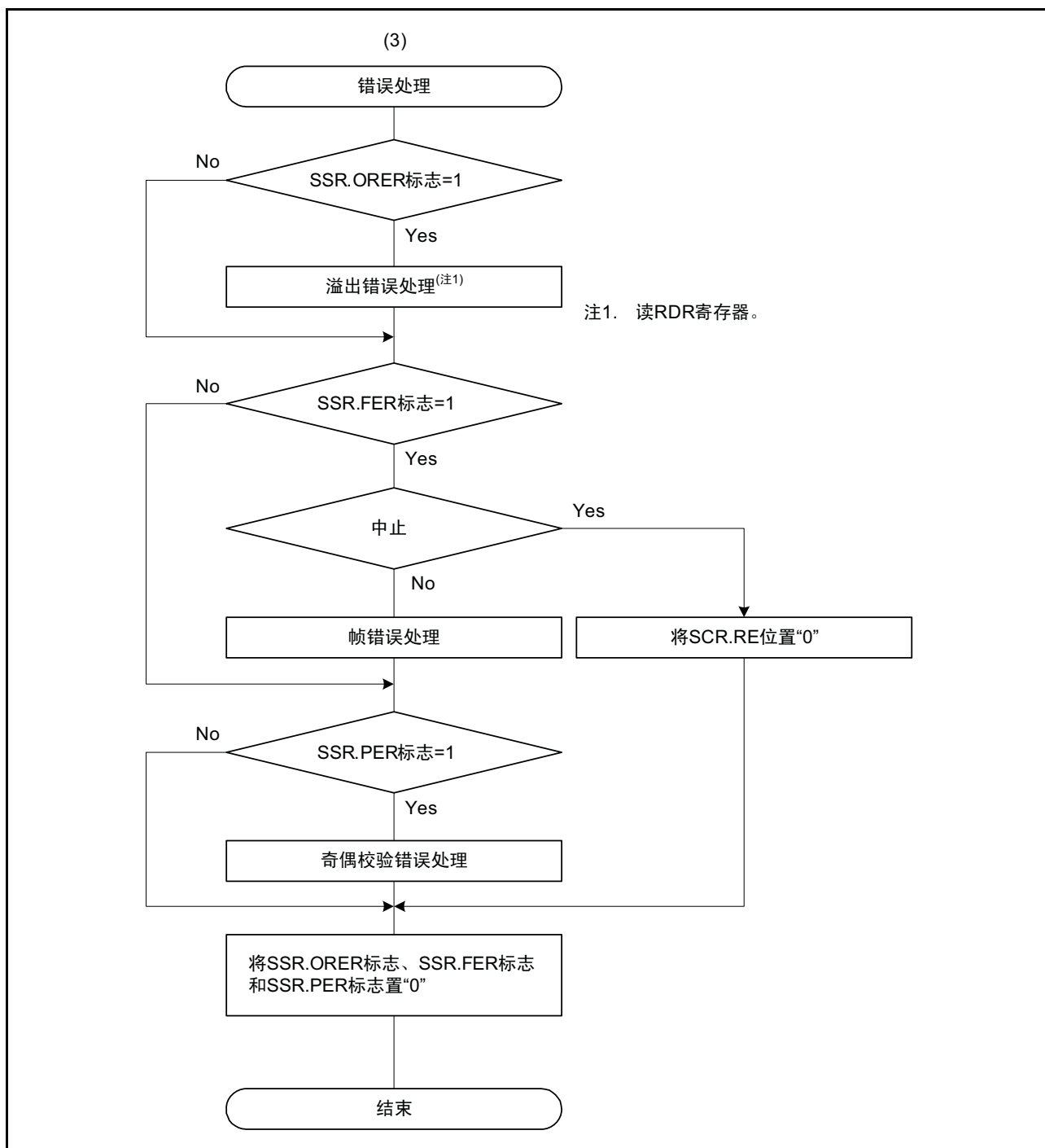


图 20.12 异步模式的串行接收的流程图例子 (2)

20.4 时钟同步模式的运行

时钟同步串行通信的数据格式如图 20.13 所示。

在时钟同步模式中，与时钟脉冲同步发送和接收数据。通信数据的 1 个字符由 8 位数据构成，在时钟同步模式中，不能附加奇偶校验位。

SCI 在发送数据时，从同步时钟的下降沿开始到下一个下降沿前输出数据。在接收数据时，与时钟的上升沿同步接收数据。输出 8 位数据后的通信线路保持最后 1 位的输出状态。

SCI 内部的发送部和接收部各自独立，因此能通过共享时钟进行全双工通信。因为发送部和接收部都为双缓冲结构，所以能在发送时写下一个发送数据，通过在接收时读前一个接收数据，进行连续接收和发送。

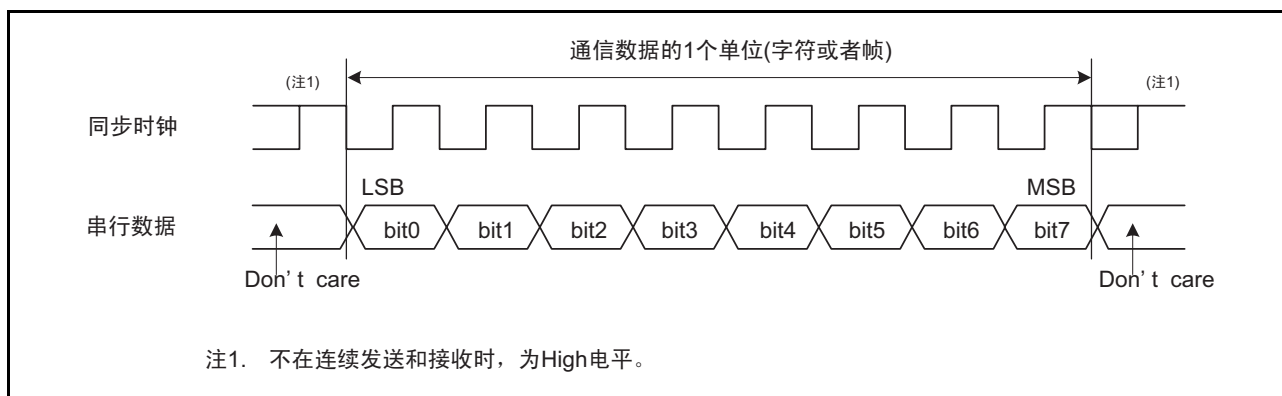


图 20.13 时钟同步串行通信的数据格式 (LSB first)

20.4.1 时钟

能通过设定 `SCR.CKE[1:0]` 位，选择内部波特率发生器生成的内部时钟或者 `SCKn` 引脚输入的外部同步时钟。

在通过内部时钟运行时，从 `SCKn` 引脚输出同步时钟。同步时钟在进行 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 **High** 电平。但是，在只进行接收时，在发生溢出错误或者在将 `SCR.RE` 位置“0”前输出同步时钟。

20.4.2 SCI 的初始化 (时钟同步模式)

必须在发送和接收数据前给 SCR 寄存器写初始值 “00h”，并且根据图 20.14 的流程图例子进行初始化。必须在对 SCR 寄存器进行初始化后更改运行模式或者通信格式。

必须注意：即使将 SCR.RE 位置 “0”，也不对 SSR.ORER 标志、SSR.FER 标志、SSR.PER 标志或者 RDR 寄存器进行初始化。

必须注意：如果将 SCR.TE 位从 “1” 变为 “0” 或者从 “0” 变为 “1”，就在 SCR.TIE 位为 “1” 时产生 TXI 中断。

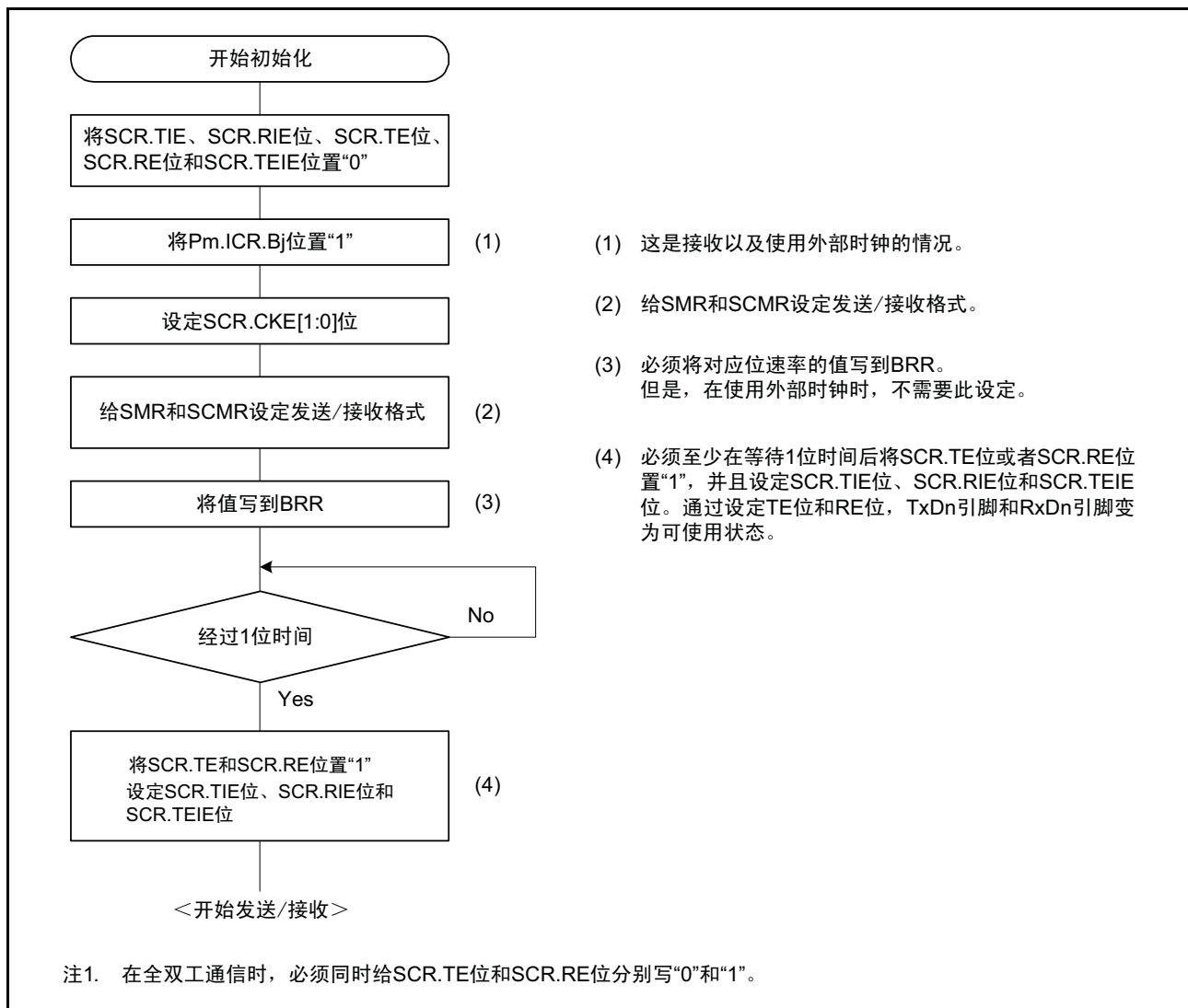


图 20.14 SCI 的初始化流程图例子 (时钟同步模式)

20.4.3 串行数据的发送 (时钟同步模式)

时钟同步模式的串行发送例子如图 20.15 所示。

在发送串行数据时，SCI 的运行如下：

1. 如果通过 TXI 中断处理程序给 TDR 寄存器写数据，SCI 就将数据从 TDR 寄存器传送到 TSR 寄存器。在开始发送时，通过将 SCR.TIE 位置“1”后将 SCR.TE 位置“1”，或者同时将这 2 个位置“1”，产生 TXI 中断请求。
2. 通过将数据从 TDR 寄存器传送到 TSR 寄存器，开始发送。此时，如果 TIE 位为“1”，就产生 TXI 中断请求。能通过此 TXI 中断处理程序，在上次传送的数据发送结束前给 TDR 寄存器写下一个发送数据，进行连续的发送。
3. 如果设定为时钟输出模式，就与输出时钟同步从 TxDn 引脚输出 8 位数据。如果设定为外部时钟，就与输入时钟同步从 TxDn 引脚输出 8 位数据。
4. 在发送最后 1 位数据时检查 TDR 寄存器的更新 (写)。
5. 如果 TDR 寄存器已被更新，就将数据从 TDR 寄存器传送到 TSR 寄存器，开始下一帧的发送。
6. 如果 TDR 寄存器未被更新，SSR.TEND 标志就变为“1”，并且保持最后 1 位的输出状态。此时，如果 SCR.TEIE 位为“1”，就产生 TEI 中断请求。SCKn 引脚被固定为 High 电平。

串行发送的流程图例子如图 20.16 所示。

不能在接收错误标志 (SSR.ORER、SSR.FER、SSR.PER) 为“1”的状态下开始发送，必须在开始发送前将接收错误标志置“0”。必须注意：如果只将 SCR.RE 位置“0”，接收错误标志也不变为“0”。

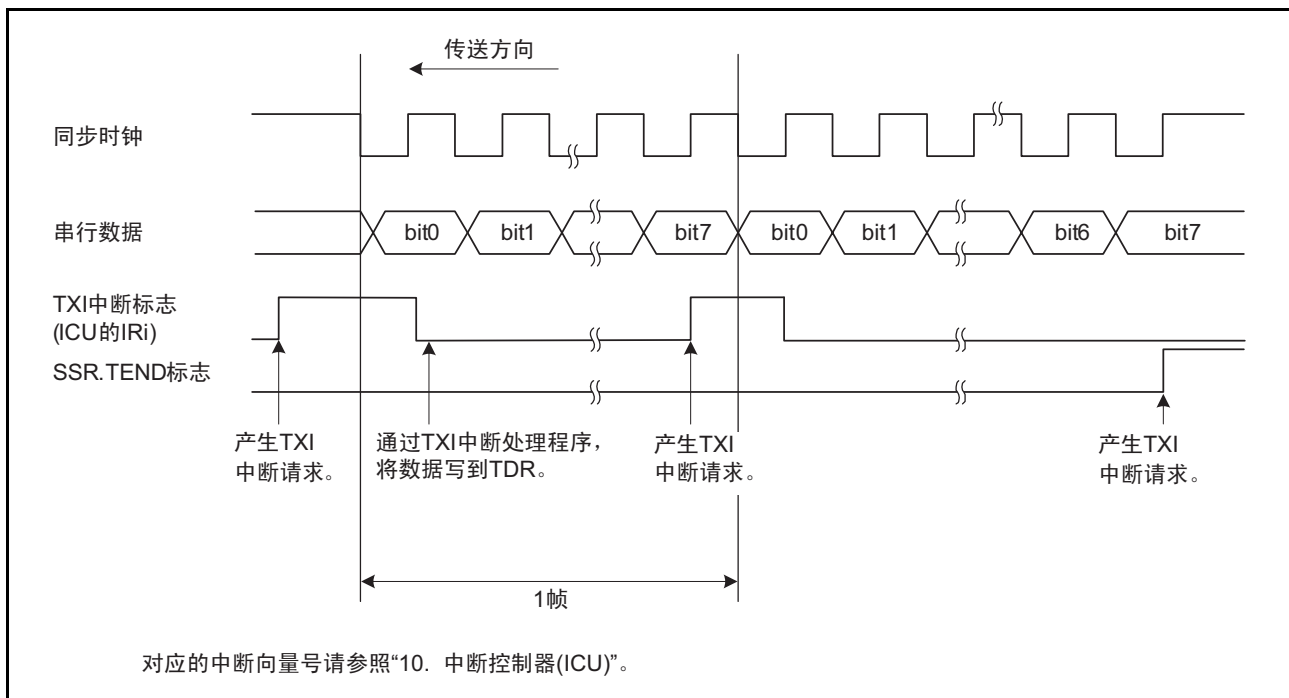


图 20.15 时钟同步模式的串行发送例子

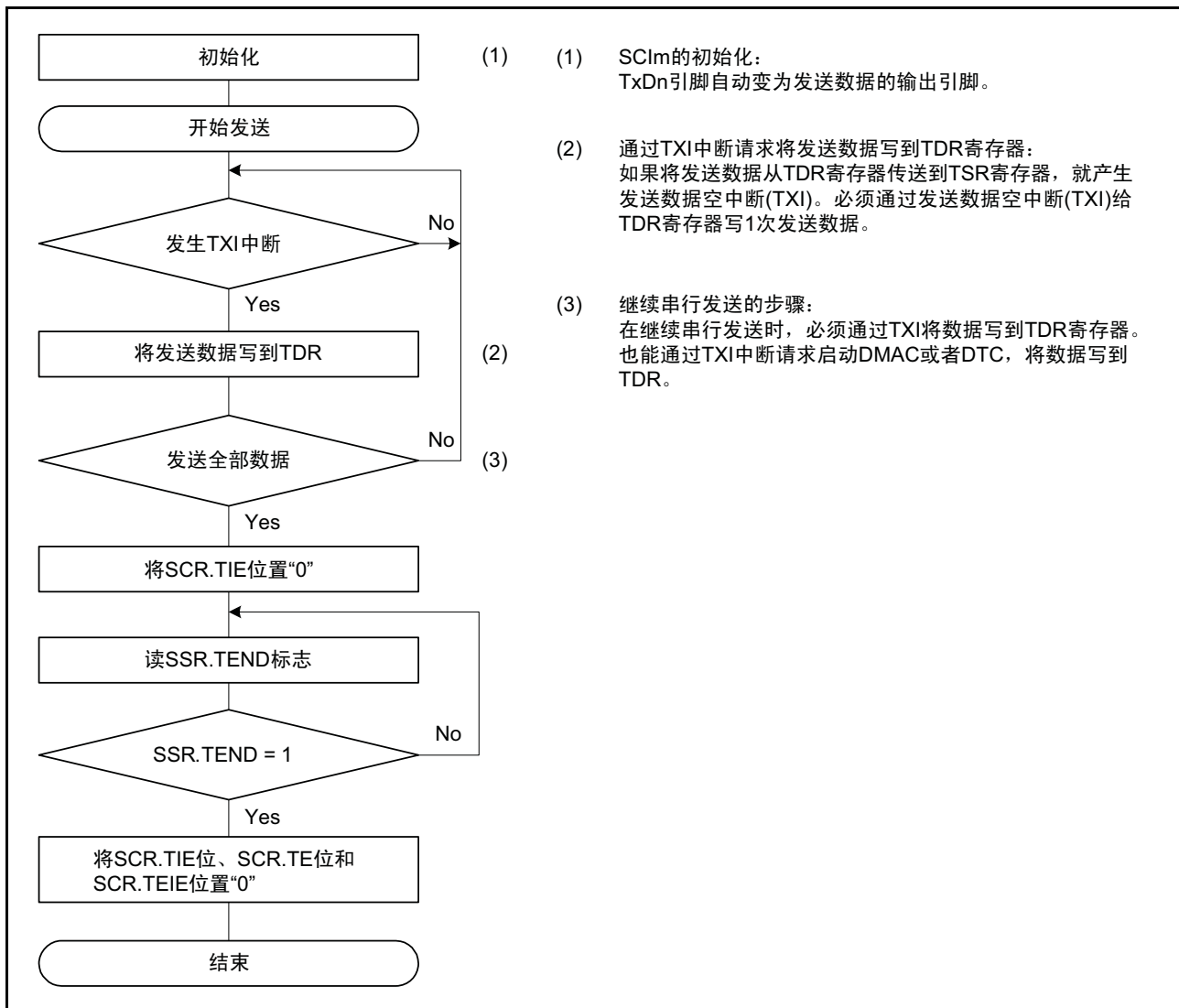


图 20.16 时钟同步模式的串行发送的流程图例子

20.4.4 串行数据的接收 (时钟同步模式)

时钟同步模式的串行接收例子如图 20.17 所示。

在接收串行数据时，SCI 的运行如下：

1. SCI与同步时钟的输入同步或者与输出同步将内部进行初始化后开始接收，并且将接收数据取到RDR寄存器。
2. 如果发生溢出错误，SSR.ORER标志就变为“1”。此时，如果SCR.RIE位为“1”，就产生ERI中断请求。不将接收数据传送到RDR寄存器。
3. 如果正常接收，就将接收数据传送到RDR寄存器。此时，如果RIE位为“1”，就产生RXI中断请求。能通过此RXI中断处理程序，在下一个数据接收结束前读已传送到RDR寄存器的接收数据，进行连续的接收。

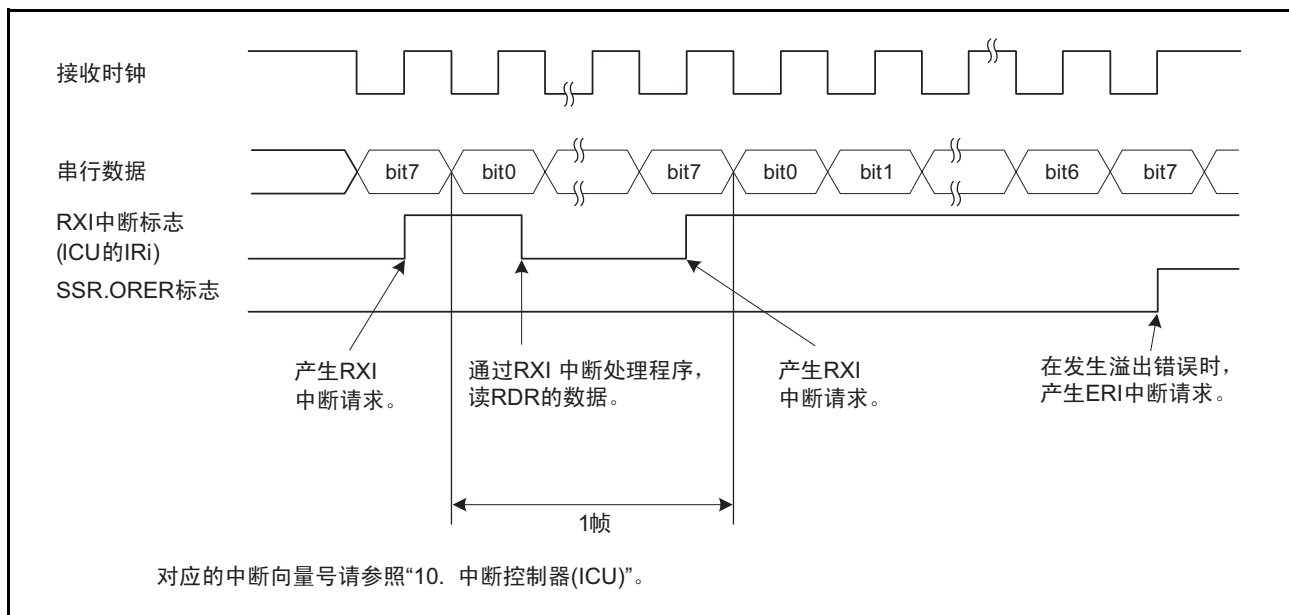


图 20.17 时钟同步模式的串行接收例子

在接收错误标志为“1”的状态下，不能进行以后的发送和接收。因此，必须在继续接收前将 SSR.ORER 标志、SSR.FER 标志和 SSR.PER 标志置“0”。另外，必须在进行溢出错误处理时读 RDR 寄存器。

串行接收的流程图例子如图 20.18 所示。

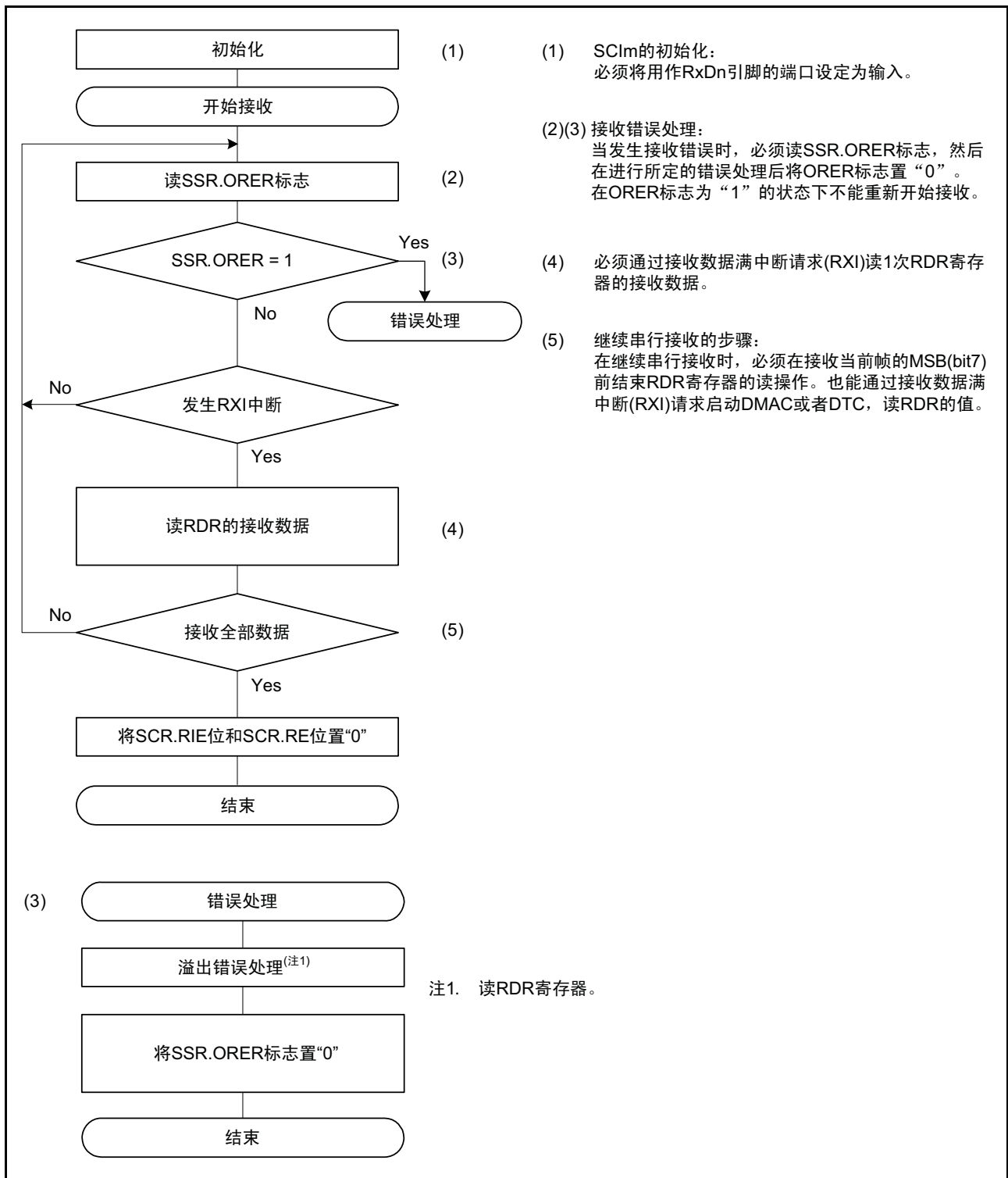


图 20.18 时钟同步模式的串行接收的流程图例子

20.4.5 串行数据的全双工通信 (时钟同步模式)

时钟同步模式的串行全双工通信的流程图例子如图 20.19 所示。

在对 SCI 进行初始化后, 必须按照以下步骤进行串行全双工通信。

要从发送转换为全双工通信时, 必须通过读已被置“1”的 SSR.TEND 标志, 确认 SCI 处于发送结束状态。然后, 必须在对 SCR 寄存器进行初始化后, 将 SCR.TIE 位、SCR.RIE 位、SCR.TE 位、SCR.RE 位和 SCR.TEIE 位同时置“1”。

要从接收转换为全双工通信时, 必须在确认 SCI 处于接收结束状态后, 先将 SCR.RIE 和 SCR.RE 位置“0”, 然后确认错误标志 (SSR.ORER、SSR.FER 和 SSR.PER) 为“0”, 最后将 SCR.TIE 位、SCR.RIE 位、SCR.TE 位、SCR.RE 位和 SCR.TEIE 位同时置“1”。

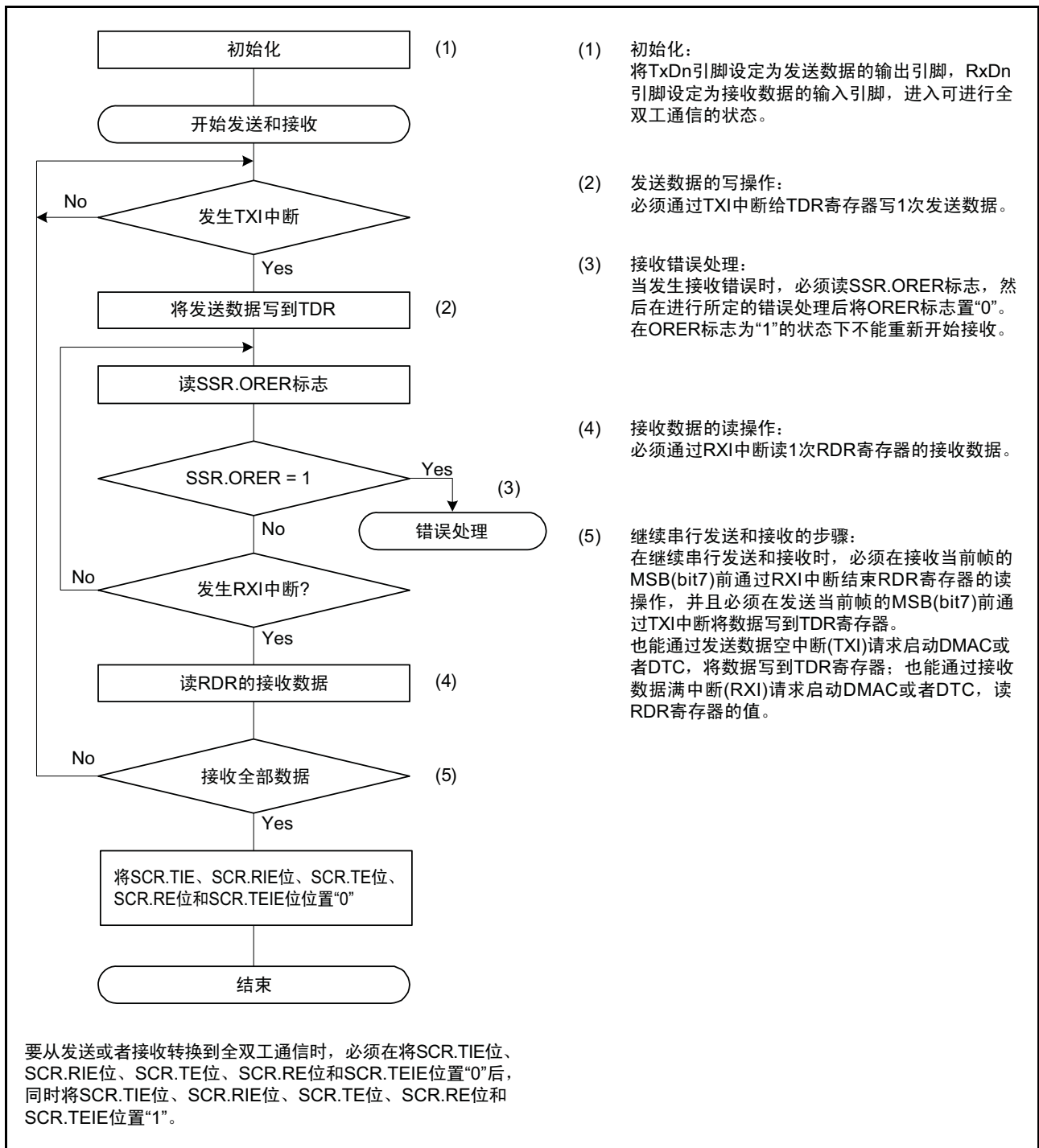


图 20.19 时钟同步模式的串行全双工通信的流程图例子

20.5 智能卡接口模式的运行

作为 SCI 的扩展功能，对应符合 ISO/IEC 7816-3 (Identification Card) 规格的智能卡 (IC 卡) 接口。通过寄存器转换为智能卡接口模式。

20.5.1 连接例子

智能卡 (IC 卡) 的连接例子如图 20.20 所示。

IC 卡是指通过 1 条数据传送线进行发送和接收，因此必须连接 TxDn 引脚和 RxDn 引脚并且通过电阻将数据传送线上拉到电源 VCC 侧。

如果在不连接 IC 卡的状态下将 SCR.TE 位和 SCR.RE 位都置“1”，就能进行自诊断（自己能接收自己发送的数据）。

如果给 IC 卡提供由 SCI 生成的时钟，就必须将 SCKn 引脚的输出信号输入到 IC 卡的 CLK 引脚。

能将 LSI 的输出端口用于复位信号的输出。

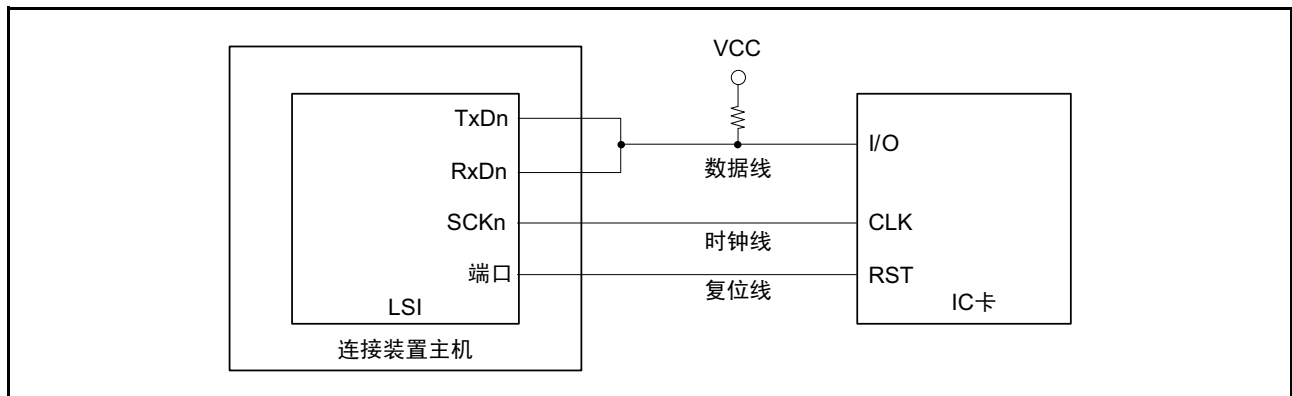


图 20.20 智能卡 (IC 卡) 的连接例子

20.5.2 数据格式 (块传送模式除外)

智能卡接口模式的发送 / 接收格式如图 20.21 所示。

- 在异步模式中，1 帧由 8 位数据和奇偶校验位构成。
- 在发送时，从奇偶校验位结束到下一帧的开始至少需要 2etu (Elementary Time Unit: 1 位传送时间) 的保护时间。
- 如果在接收时检测到奇偶校验错误，就在经过 10.5etu 后，从起始位输出 1etu 期间的错误信号 (Low 电平)。
- 如果在发送时对错误信号进行采样，就在至少经过 2etu 后，自动重新发送相同的数据。

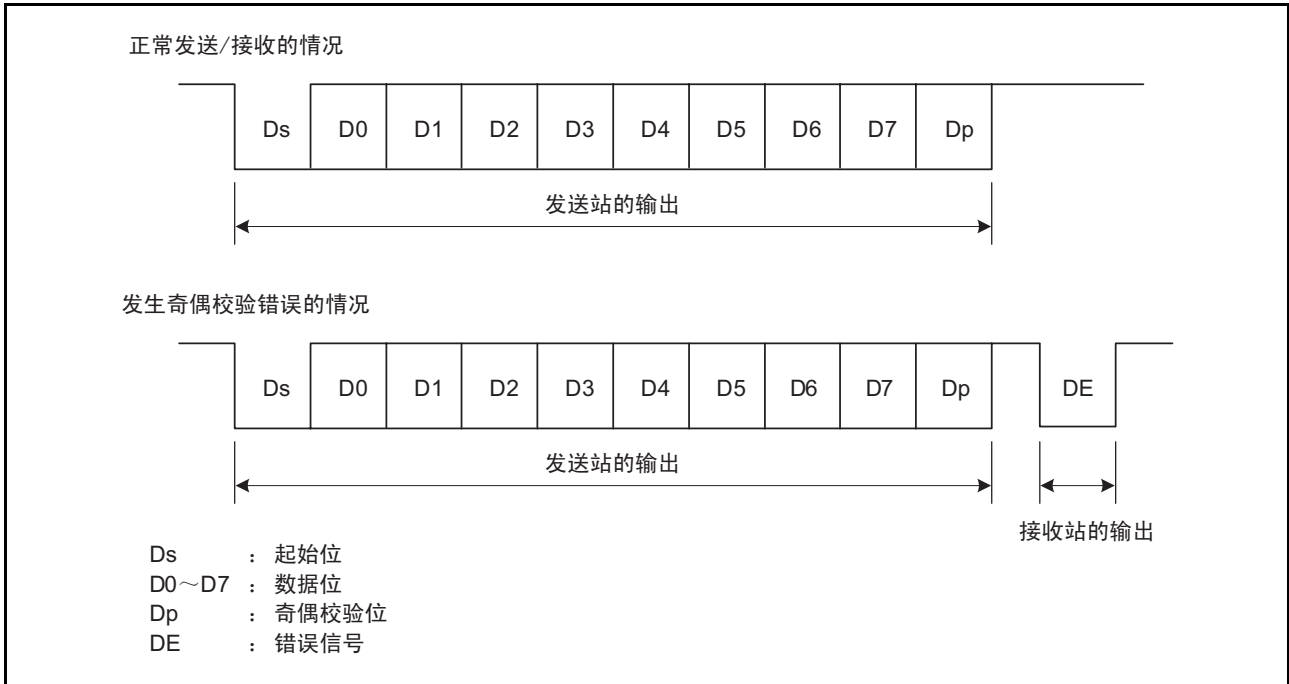


图 20.21 智能卡接口模式的数据格式

正向协议型和反向协议型这 2 种 IC 卡的发送和接收如下：

(1) 正向协议型

正向协议型如图 20.22 所示的开始字符例子，使逻辑 1 电平对应状态 Z，使逻辑 0 电平对应状态 A，以 LSB first 进行发送和接收。图 20.22 中的开始字符数据为“3Bh”。

对于正向协议型，必须将 SCMR.SDIR 位和 SCMR.SINV 位都置“0”。为了根据智能卡的规格进行偶校验，必须将 SMR.PM 位置“0”。

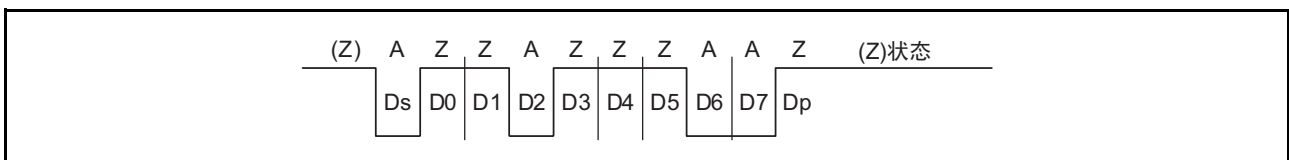


图 20.22 正向协议 (SCMR.SDIR 位 = 0, SCMR.SINV 位 = 0, SMR.PM 位 = 0)

(2) 反向协议型

反向协议型使逻辑 1 电平对应状态 A，使逻辑 0 电平对应状态 Z，以 MSB first 进行发送和接收。图 20.23 中的开始字符数据为“3Fh”。

对于反向协议型，必须将 SCMR.SDIR 位和 SCMR.SINV 位都置“1”。在根据智能卡的规格进行偶校验时，奇偶校验位为逻辑 0，对应状态 Z。在 LSI 中，SINV 位只将数据位 D7 ~ D0 取反。因此，在发送和接收时，必须将 SMR.PM 位置“1”，将奇偶校验位取反。

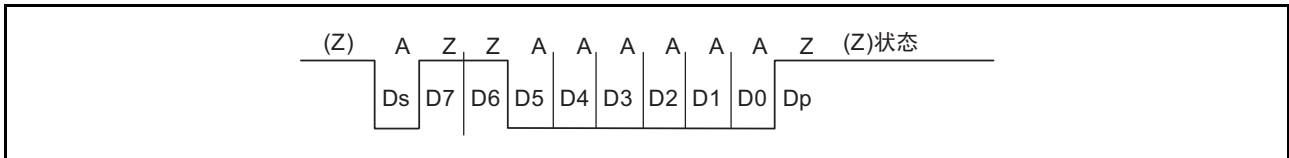


图 20.23 反向协议 (SCMR.SDIR 位=1, SCMR.SINV 位=1, SMR.PM 位=1)

20.5.3 块传送模式

块传送模式和一般的智能卡接口模式比较，有以下的不同点：

- 在接收时进行奇偶校验，即使检测到错误，也不输出错误信号。因为 SSR.PER 标志变为“1”，所以必须在接收下一帧的奇偶校验位前将此位置“0”。
- 在发送时，从奇偶校验位结束到下一帧的开始至少需要 1etu 的保护时间。
- 因为不重新进行发送，所以在开始发送后的 11.5etu 后，SSR.TEND 标志变为“1”。
- 和一般的智能卡接口模式一样，SSR.ERS 标志表示错误信号的状态。因为不发送和接收错误信号，所以 SSR.ERS 标志总是为“0”。

20.5.4 接收数据的采样时序和接收容限

能用于智能卡接口模式的发送 / 接收时钟只有内部波特率发生器生成的内部时钟。

在智能卡接口模式中，根据 SCMR.BCP2 位和 SMR.BCP[1:0] 位的设定，SCI 通过频率为 32 倍、64 倍、372 倍、256 倍、93 倍、128 倍、186 倍或者 512 倍（在一般的异步模式中固定为 16 倍）位速率的基本时钟运行。

在接收时，通过基本时钟对起始位的下降沿进行采样，并且与内部取得同步。如图 20.24 所示，通过在第 16 个、32 个、186 个、128 个、46 个、64 个、93 个、256 个时钟的上升沿分别对接收数据进行采样，在各位的中央取数据。能用以下的表达式表示此时的接收容限：

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100[\%]$$

M: 接收容限 (%)

N: 对于时钟的位速率比 (N=32,64,372,256)

D: 时钟占空比 (D=0 ~ 1.0)

L: 帧长 (L=10)

F: 时钟频率的偏差绝对值

假设在上述的表达式中，F=0，D=0.5，N=372，则接收容限如下所示：

$$M = \left(0.5 - 1 / (2 \times 372) \right) \times 100\% = 49.866\%$$

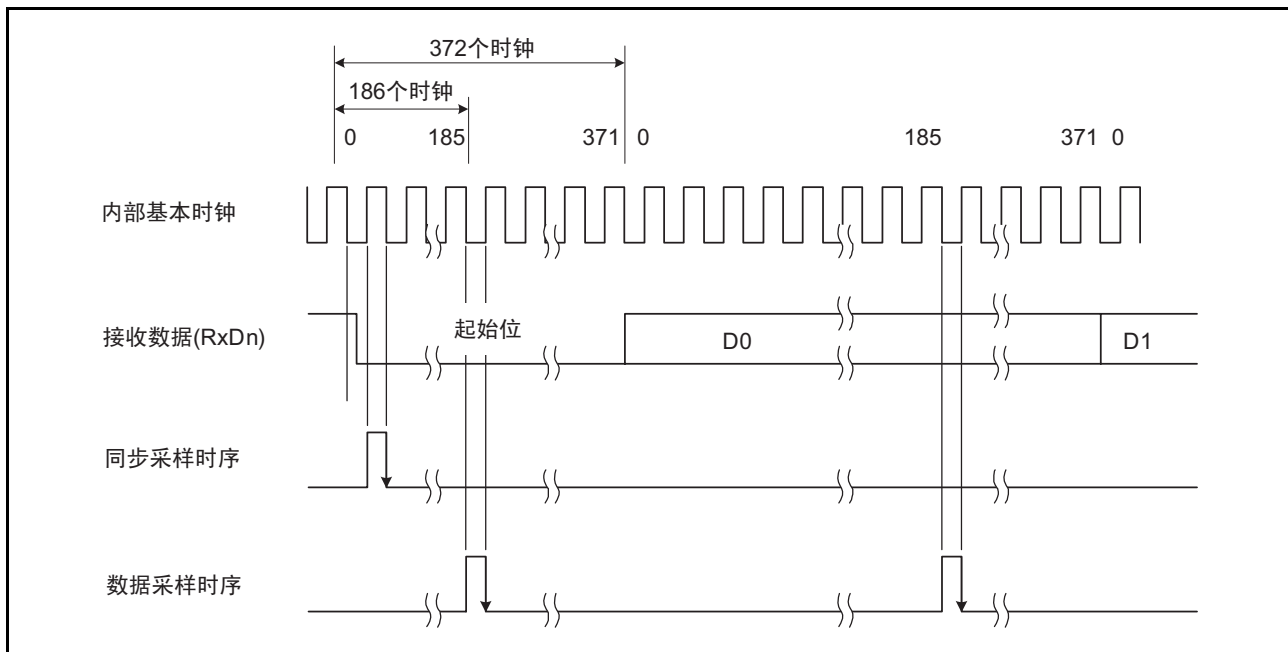


图 20.24 智能卡接口模式的接收数据的采样时序（使用 372 倍的时钟）

20.5.5 SCI 的初始化

在发送和接收数据前，必须按照以下的步骤对 SCI 进行初始化。在从发送模式转换为接收模式或者从接收模式转换为发送模式时，都需要进行初始化。

1. 给 SCR 寄存器写初始值“00h”。
2. 必须将对应引脚的 Pm.ICR.Bj 位 (m=0~9, A~E, j=0~7) 置“1”。
3. 必须将 SSR 寄存器的错误标志 (ORER、ERS、PER) 置“0”。
4. 必须设定 SMR.GM 位、SMR.BLK 位、SMR.PM 位、SMR.BCP[1:0] 位、SMR.CKS[1:0] 位和 SCMR.BCP2 位。此时，必须将 SMR.PE 位置“1”。
5. 必须设定 SCMR.SDIR 位、SCMR.SINV 位和 SCMR.SMIF 位，并且将 TxDn 引脚对应的 Pm.DDR.Bj 位置“0”。因此，TxDn 引脚和 RxDn 引脚都从端口转换为 SCI 的引脚，并且为高阻抗状态。
6. 给 BRR 寄存器设定与位速率对应的值。
7. 必须设定 SCR.CKE[1:0] 位。此时，必须将 SCR.TIE 位、SCR.RIE 位、SCR.TE 位、SCR.RE 位和 SCR.TEIE 位置“0”。
如果将 CKE0 位置“1”，就从 SCKn 引脚输出时钟。
8. 必须在至少等待 1 位时间后设定 SCR.TIE 位、SCR.RIE 位、SCR.TE 位和 SCR.RE 位。除了进行自诊断以外，必须将 TE 位和 RE 位同时置“1”。

要从接收模式转换为发送模式时，必须在确认接收结束后从初始化开始进行，并且将 TE 位置“1”、RE 位置“0”。能通过 RXI 中断请求、SSR.ORER 标志或者 SSR.PER 标志确认接收的结束。

要从发送模式转换为接收模式时，必须在确认发送结束后从初始化开始进行，并且将 TE 位置“0”、RE 位置“1”。能通过 SSR.TEND 标志确认发送的结束。

20.5.6 串行数据的发送 (块传送模式除外)

智能卡接口模式的串行发送包括错误信号的采样和重新发送处理, 因此和一般的串行通信接口模式的运行不同 (块传送模式除外)。发送时的重新传送如图 20.25 所示。

1. 如果在结束 1 帧的发送后采样到接收侧的错误信号, **SSR.ERS** 标志就变为 “1”。此时, 如果 **SCR.RIE** 位为 “1”, 就产生 **ERI** 中断请求。必须在对下一个奇偶校验位进行采样前将 **ERS** 标志置 “0”。
2. 在接收到错误信号的帧中, **SSR.TEND** 标志不变为 “1”。再次将数据从 **TDR** 寄存器传送到 **TSR** 寄存器, 自动进行重新发送。
3. 如果接收侧没有返回错误信号, **ERS** 标志就不变为 “1”。
4. 在判断出包括重新传送的 1 帧的发送已结束后, **SSR.TEND** 标志变为 “1”。此时, 如果 **SCR.TIE** 位为 “1”, 就产生 **TXI** 中断请求。通过给 **TDR** 寄存器写发送数据, 开始下一个数据的发送。

串行发送的流程图例子如图 20.27 所示。能通过 **TXI** 中断源启动 **DTC** 或者 **DMAC**, 自动进行上述的一系列处理。

在发送时, 如果在 **SCR.TIE** 位为 “1” 时 **SSR.TEND** 标志变为 “1”, 就产生 **TXI** 中断请求。如果预先将 **TXI** 中断设定为 **DTC** 或者 **DMAC** 的启动源, 就在通过 **TXI** 中断请求启动 **DTC** 或者 **DMAC** 后, 进行发送数据的传送。在通过 **DTC** 或者 **DMAC** 进行数据传送时, **TEND** 标志自动变为 “0”。

如果发生错误, **SCI** 就自动重新发送相同的数据。在此期间, **TEND** 标志保持 “0” 并且不启动 **DTC** 或者 **DMAC**。因此, 自动发送 **SCI** 以及 **DTC** 或者 **DMAC** 指定的字节数, 包括发生错误时的重新发送。但是, 在发生错误时 **ERS** 标志不自动变为 “0”, 因此必须预先将 **RIE** 位置 “1”, 使在发生错误时产生 **ERI** 中断请求, 并且将 **ERS** 标志置 “0”。

在使用 **DTC** 或者 **DMAC** 进行发送和接收时, 必须先设定 **DTC** 或者 **DMAC**, 在设定为允许状态后进行 **SCI** 的设定。

有关 **DTC** 或者 **DMAC** 的设定方法, 请参照 “12. DMA 控制器 (DMAC)” 和 “13. 数据传送控制器 (DTC)”。

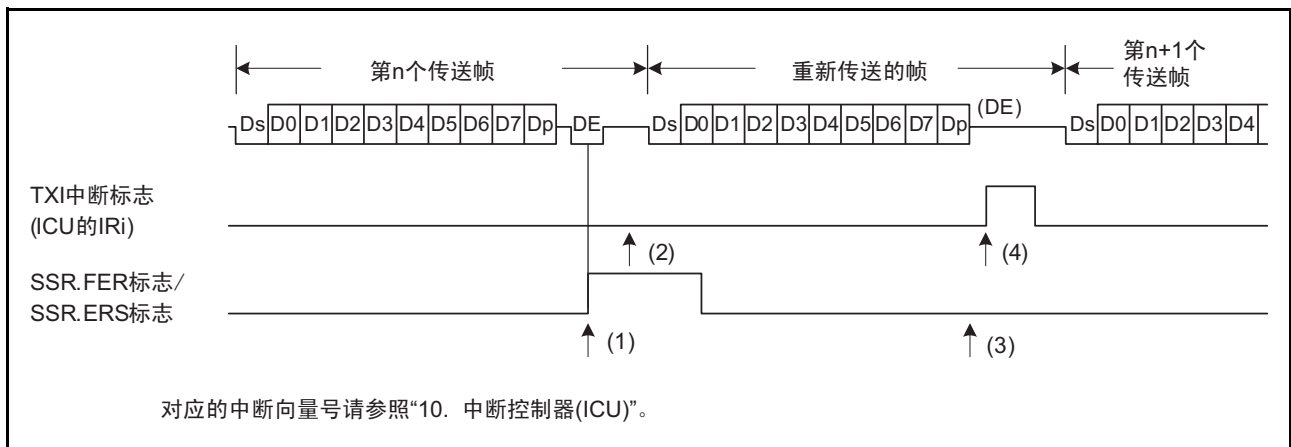


图 20.25 SCI 发送模式的重新传送 (发送时的重新传送)

SSR.TEND 标志变为 “1” 的时序取决于 **SMR.GM** 位的设定, **TEND** 标志的产生时序如图 20.26 所示。

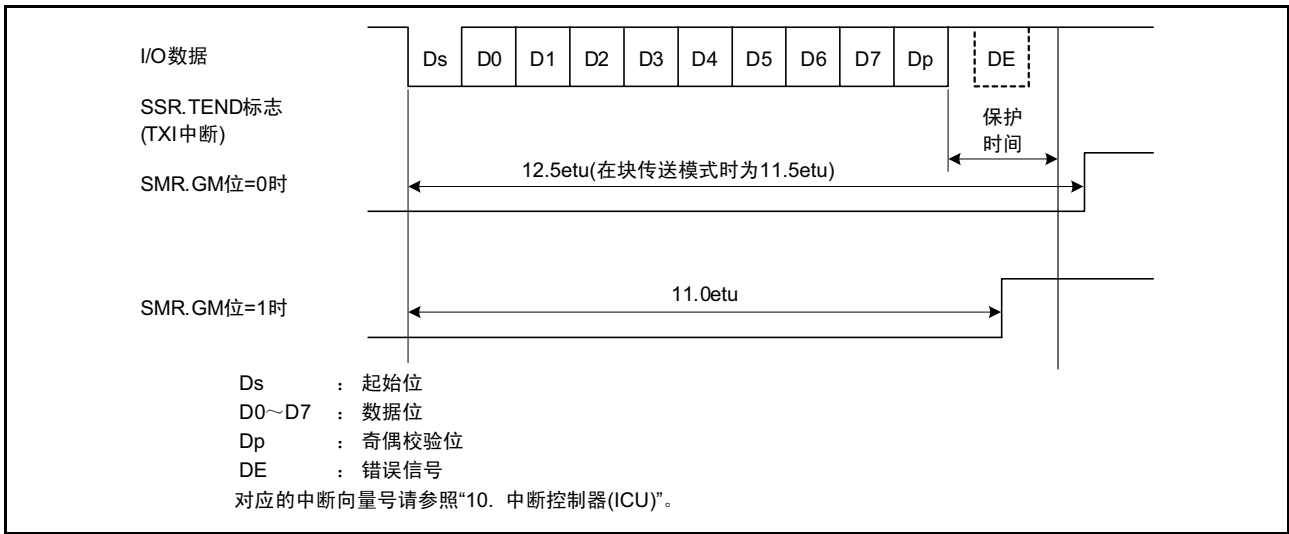


图 20.26 发送时的 SSR.TEND 标志的产生时序

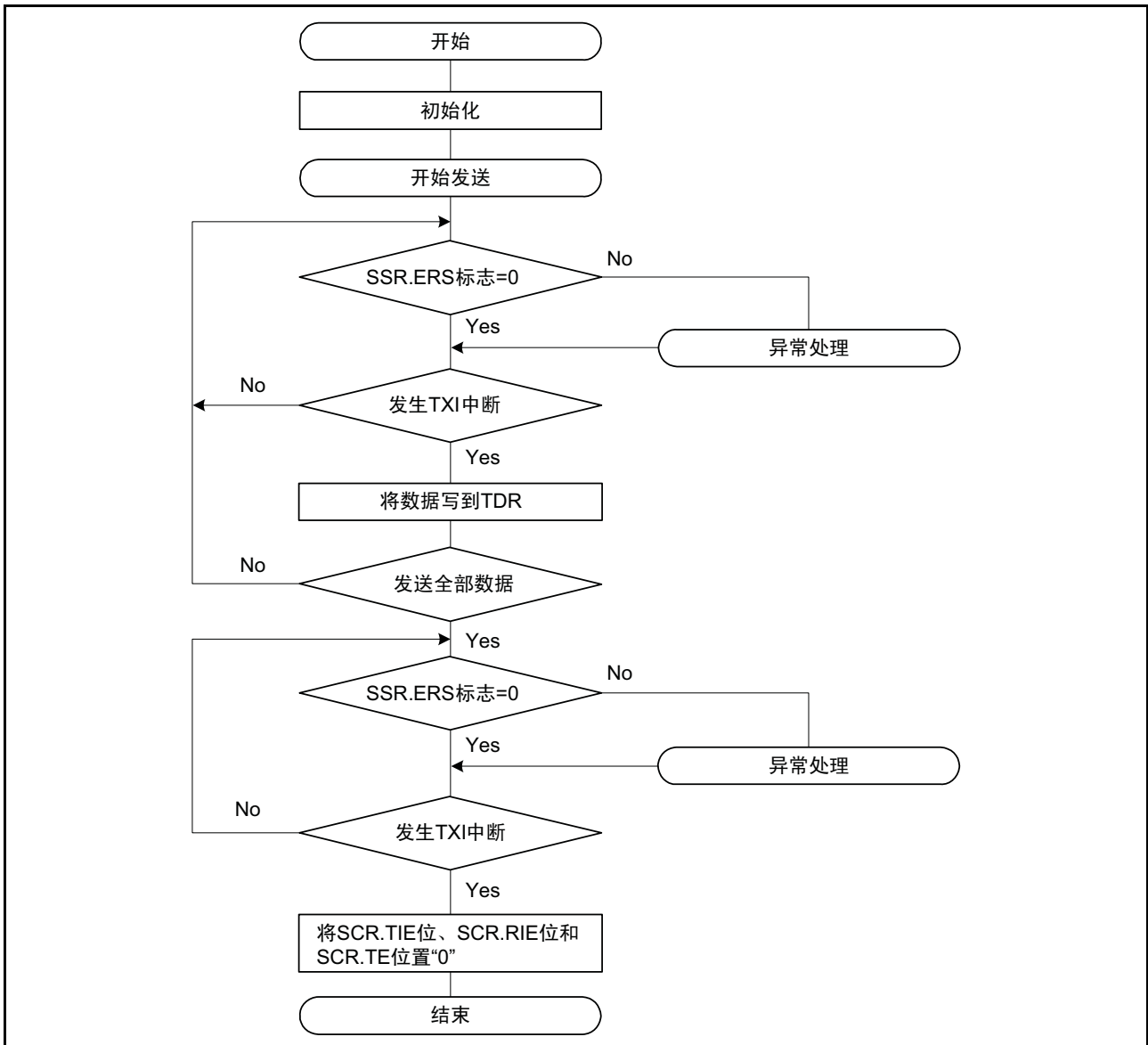


图 20.27 串行发送的流程图例子

20.5.7 串行接收 (块传送模式除外)

智能卡接口模式的串行接收和串行通信接口模式的处理步骤相同。接收模式的重新传送如图 20.28 所示。

1. 如果在接收数据时检测到奇偶校验错误, SSR.PER 标志就变为 “1”。此时, 如果 SCR.RIE 位为 “1”, 就产生 ERI 中断请求。必须在下一个奇偶校验位的采样时序前将 PER 标志置 “0”。
2. 对于检测到奇偶校验错误的帧, 不发生 RXI 中断。
3. 如果检测不到奇偶校验错误, SSR.PER 标志就不变为 “1”。
4. 如果在判断出接收正常结束后 RIE 位变为 “1”, 就产生 RXI 中断请求。

串行接收的流程图例子如图 20.29 所示。能通过 RXI 中断请求启动 DTC 或者 DMAC, 自动进行上述的一系列处理。

如果在接收时预先将 RIE 位置 “1”, 就产生 RXI 中断请求。如果预先将 RXI 中断设定为 DTC 或者 DMAC 的启动源, 就在通过 RXI 中断请求启动 DTC 或者 DMAC 后, 进行接收数据的传送。

如果在接收时发生错误, 并且 SSR.ORER 标志或者 SSR.PER 标志变为 “1”, 就产生接收错误中断 (ERI) 请求, 因此必须将错误标志置 “0”。因为在发生错误时不启动 DTC 或者 DMAC 而省略接收数据, 所以只传送由 DTC 或者 DMAC 设定的字节数的接收数据。

在接收时发生奇偶校验错误并且 PER 标志为 “1” 的情况下, 也将接收数据传送到 RDR 寄存器, 因此能读此数据。

注 1. 有关块传送模式, 请参照 “20.3 异步模式的运行”。

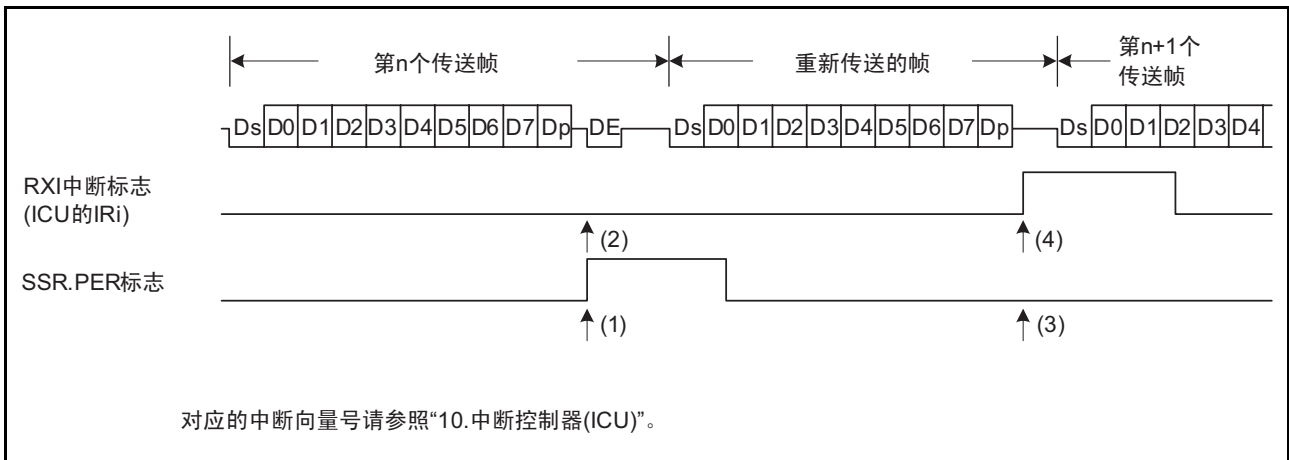


图 20.28 SCI 接收模式的重新传送 (接收时的重新传送)

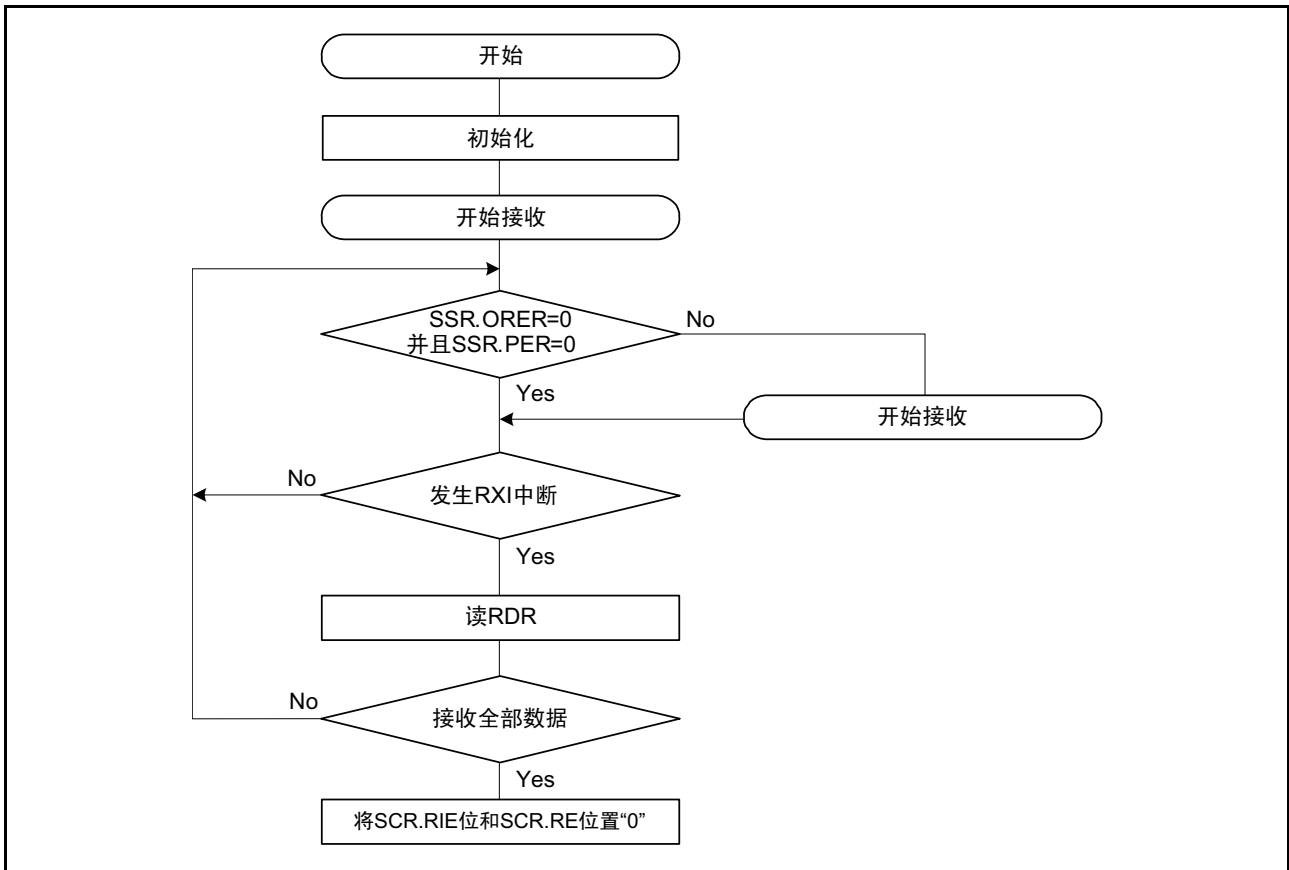


图 20.29 串行接收的流程图例子

20.5.8 时钟的输出控制

当 SMR.GM 位为 “1” 时，能通过 SCR.CKE[1:0] 位停止时钟输出。此时，能将时钟脉冲的最小宽度设定为指定的宽度。

时钟输出的停止时序如图 20.30 所示，这是将 GM 位和 CKE1 位分别置 “1” 和 “0” 并且控制 CKE0 位时的例子。

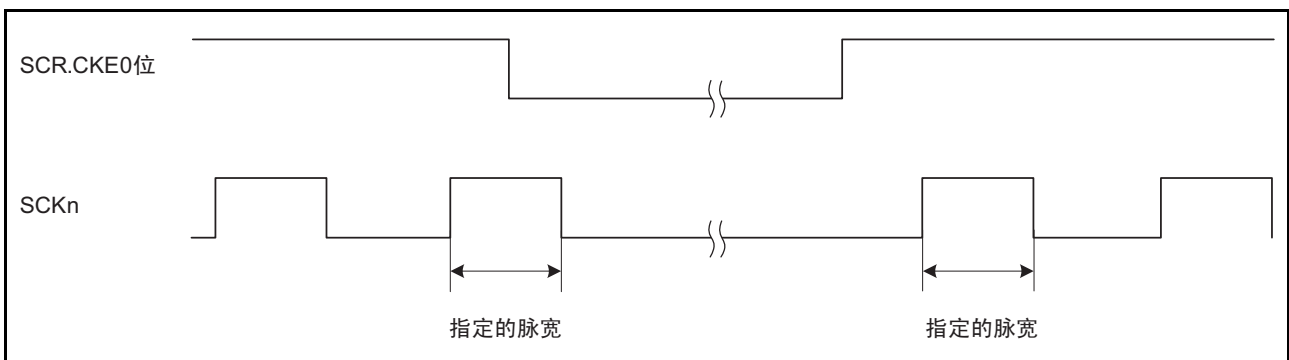


图 20.30 时钟输出的停止时序

在接通电源、向软件待机模式转移或者从软件待机模式返回时，为了确保时钟的占空比，必须按照以下的步骤进行处理：

(1) 电源接通时

为了从接通电源开始确保时钟的占空比的转换步骤如下所示：

1. 初始状态为端口输入的高阻抗状态。要固定电位时，必须使用上拉电阻或者下拉电阻。
2. 必须通过SCR.CKE1位将SCKn引脚固定为指定的输出。
3. 必须将SMR寄存器和SCMR寄存器置位并且转换为智能卡接口模式的运行，并且将SCR.CKE0位置“1”，开始时钟的输出。

(2) 模式转换时

(a) 从智能卡接口模式转移到软件待机模式时

1. 必须给SCKn引脚对应的数据寄存器 (Pm.DR) 和数据方向寄存器 (Pm.DDR) 设定软件待机模式时的输出固定状态的值。
2. 必须给SCR.TE位和SCR.RE位写“0”，停止发送和接收。
同时，必须给SCR.CKE1位设定软件待机时的输出固定状态的值。
3. 必须给SCR.CKE0位写“0”，停止时钟。
4. 必须等待1个串行时钟周期。在此期间，在输出指定的High电平宽度后，时钟输出以Low电平停止运行。
5. 必须转移到软件待机状态。

(b) 从软件待机模式返回到智能卡接口模式时

1. 必须解除软件待机状态。
2. 必须将SCR.CKE0位置“1”，以指定的频率重新开始时钟的输出。

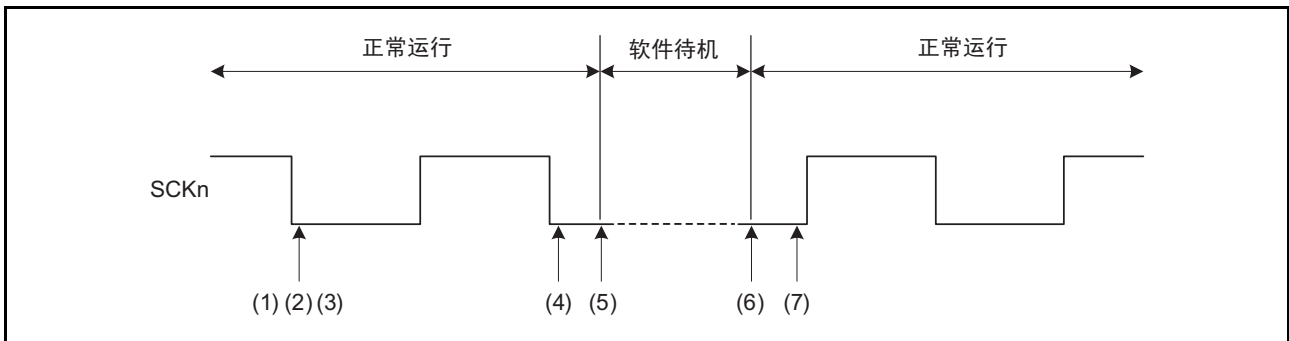


图 20.31 时钟的停止和重新启动的步骤

20.6 中断源

20.6.1 串行通信接口模式的中断

串行通信接口模式的中断源如表 20.17 所示。各中断源分配有不同的中断向量，能通过 SCR 寄存器的允许位分别允许各中断源。

如果将发送数据从 TDR 寄存器传送到 TSR 寄存器，就在 SCR.TIE 位为“1”时产生 TXI 中断请求。通过在将 SCR.TIE 位置“1”后将 SCR.TE 位置“1”，或者将 SCR.TIE 位和 SCR.TE 位同时置“1”，产生 TXI 中断请求。能通过 TXI 中断请求，在启动 DTC 或者 DMAC 后进行数据传送。

如果给 RDR 寄存器设定接收数据，就在 SCR.RIE 位为“1”时产生 RXI 中断请求。能通过 RXI 中断请求，在启动 DTC 或者 DMAC 后进行数据传送。

在 SSR.ORER 标志、SSR.FER 标志或者 SSR.PER 标志为“1”并且 SCR.RIE 位为“1”的情况下，产生 ERI 中断请求。此时，不产生 RXI 中断请求。

如果在发送数据的最后 1 位时未更新 TDR 寄存器，SSR.TEND 标志就变为“1”，并且在 SCR.TEIE 位为“1”时产生 TEI 中断请求。如果通过 TXI 中断处理给 TDR 寄存器写数据，SSR.TEND 标志就变为“0”并且 TEI 中断请求被取消。在通过给 TDR 寄存器写发送数据，将 SSR.TEND 标志置“0”时，必须读 SSR.TEND 标志，确认该标志已变为“0”。

通过在将 SCR.TIE 位置“1”后将 SCR.TE 位置“1”，或者将 SCR.TIE 位和 SCR.TE 位同时置“1”，产生 TXI 中断请求。即使在 SCR.TIE 位为“0”的状态下将 SCR.TE 位或者 SCR.TIE 位置“1”，也不产生 TXI 中断请求。因此，在发送最后的数据时，必须暂时禁止 TXI 中断，在进行发送结束中断处理后要重新开始数据发送时，不通过 SCR.TIE 位而通过 TXI 中断对应的 ICU.IERi.IENj 位控制中断的禁止和允许。

表 20.17 SCI 中断源

| 名称 | 中断源 | 中断标志 | DTC 的启动 | DMAC 的启动 | 优先级 |
|-----|-------|--------------|---------|----------|------------------|
| ERI | 接收错误 | ORER、FER、PER | 不能 | 不能 | 高 ↑ ↓ 低 |
| RXI | 接收数据满 | — | 能 | 能 | |
| TXI | 发送数据空 | — | 能 | 能 | |
| TEI | 发送结束 | TEND | 不能 | 不能 | |

20.6.2 智能卡接口模式的中断

智能卡接口模式的中断源如表 20.18 所示，不能使用发送结束中断 (TEI)。

表 20.18 SCI 中断源

| 名称 | 中断源 | 中断标志 | DTC 的启动 | DMAC 的启动 | 优先级 |
|-----|-------------|--------------|---------|----------|-----|
| ERI | 接收错误、错误信号检测 | ORER、PER、ERS | 不能 | 不能 | 高 |
| RXI | 接收数据满 | — | 能 | 能 | ▲ |
| TXI | 发送数据空 | TEND | 能 | 能 | 低 |

智能卡接口模式和一般的 SCI 相同，能通过 DTC 或者 DMAC 进行发送和接收。如果在发送时 SSR.TEND 标志变为“1”，就产生 TXI 中断请求。如果预先将 TXI 中断设定为 DTC 或者 DMAC 的启动源，就通过 TXI 中断请求，在启动 DTC 或者 DMAC 后传送发送数据。在通过 DTC 或者 DMAC 进行数据传送时，TEND 标志自动变为“0”。

如果发生错误，SCI 就自动重新发送相同的数据。在此期间，TEND 标志保持“0”，不启动 DTC 或者 DMAC。因此，SCI 和 DTC、DMAC 自动发送指定字节数的数据，包括发生错误时的重新发送。但是，在发生错误时，SSR.ERS 标志不自动变为“0”，因此必须先将 SCR.RIE 位置“1”，然后在发生错误时产生 ERI 中断请求，并且将 ERS 标志置“1”。

在通过 DTC 或者 DMAC 进行发送和接收时，必须先设定 DTC 或者 DMAC，在设定为允许状态后进行 SCI 的设定。DTC 或者 DMAC 的设定方法请参照“12. DMA 控制器 (DMAC)”和“13. 数据传送控制器 (DTC)”。

如果在接收时将接收数据设定到 RDR 寄存器，就产生 RXI 中断请求。如果预先将 RXI 中断设定为 DTC 或者 DMAC 的启动源，就通过 RXI 中断请求，在启动 DTC 或者 DMAC 后传送接收数据。如果发生错误，错误标志就变为“1”。因此，不启动 DTC 或者 DMAC 而向 CPU 请求 ERI 中断，所以必须将错误标志置“0”。

20.7 使用时的注意事项

20.7.1 模块停止功能的设定

能通过模块停止控制寄存器 B (MSTPCRB) 禁止或者允许 SCI 的运行，初始值为停止 SCI 的运行。能通过解除模块停止状态，使寄存器变为可存取的状态。详细内容请参照“8. 低功耗功能”。

20.7.2 有关中止的检测和处理

在检测到帧错误时，能通过直接读 RxDn 引脚的值检测中止。因为在中止时 RxDn 引脚的输入全部为“0”，所以 SSR.FER 标志变为“1”（发生帧错误），SSR.PER 标志也有可能变为“1”（发生奇偶校验错误）。SCI 在接收到中止后继续进行接收。因此必须注意：即使将 FER 标志置“0”（未发生帧错误），FER 标志也会再次变为“1”。

20.7.3 标记状态和中止的发送

当 SCR.TE 位为“0”（禁止串行发送）时，TxDn 引脚变为 I/O 端口。能通过此变化将 TxDn 引脚置为标记状态或者在发送数据时发送中止。

为了在 TE 位置“1”（允许串行发送）前将通信线路置为标记状态（“1”的状态），必须将 Pm.DDR.Bj 位和 Pm.DR.Bj 位置“1”。当 TE 位为“0”时，因为 TxDn 引脚变为 I/O 端口，所以输出“1”。

另外，要在发送数据时发送中止的情况下，必须在将 Pm.DDR.Bj 位置“1”并且将 Pm.DR.Bj 位置“0”后，将 TE 位置“0”。如果将 TE 位置“0”，就对发送部进行初始化，与当前的发送状态无关，TxDn 引脚变为 I/O 端口，并且从 TxDn 引脚输出“0”。

20.7.4 有关接收错误标志和发送 (只限于时钟同步模式)

在接收错误标志 (SSR.ORER、SSR.FER、SSR.PER) 为“1”的状态下, 即使给 TDR 寄存器写数据, 也不能开始发送。在开始发送时, 必须将接收错误标志置“0”。必须注意: 即使将 SCR.RE 位置“0” (禁止串行接收), 接收错误标志也不能变为“0”。

20.7.5 有关写 TDR

能随时给 TDR 寄存器写数据。但是, 如果在 TDR 寄存器留有发送数据的状态下给 TDR 寄存器写新数据, 保存在 TDR 寄存器的数据就可能没有被传送到 TSR 寄存器而丢失。因此, 必须通过 TXI 中断请求给 TDR 寄存器写发送数据。

20.7.6 时钟同步发送时的限制事项

在将外部时钟源用于同步时钟时, 必须通过 DMAC 或者 DTC 更新 TDR 寄存器并且通过 PCLK 时钟至少经过 5 个时钟后, 输入发送时钟。如果在更新 TDR 寄存器后的 4 个时钟以内输入发送时钟, 就可能发生误动作。

20.7.7 使用 DMAC 或者 DTC 时的限制事项

在通过 DMAC 或者 DTC 读 RDR 寄存器时, 必须将该 SCI 的接收结束中断 (RXI) 设定为启动源。

20.7.8 有关低功耗状态时的运行

(1) 发送

必须在停止运行 (SCR.TIE 位 =0, TE 位 =0, TEIE 位 =0) 的状态下设定为模块停止状态或者向软件待机模式转移。对 TSR、TDR、SSR 寄存器进行复位。模块停止状态以及软件待机模式时的输出引脚的状态取决于端口的设定, 在解除模块停止状态或者软件待机模式后, 为 High 电平输出。如果在发送时转移到这些模式, 发送就被中止。

如果在解除低功耗状态后不更改发送模式而进行发送, 就必须按照将 TE 位置“1”以及读 SSR 寄存器 → 写 TDR 寄存器的顺序开始发送。如果在更改发送模式后进行发送, 就必须在进行初始设定后开始发送。

发送时的模式转移的流程图例子如图 20.32 所示, 模式转移时的端口的引脚状态如图 20.33 和图 20.34 所示。

必须在停止运行 (TE 位 =0) 的状态下, 从通过 DTC 传送进行的发送模式设定为模块停止状态或者转移到软件待机模式。在解除这些模式后通过 DTC 进行发送时, 如果将 TE 位置“1”, 就产生 TXI 中断并且开始通过 DTC 进行的发送。

(2) 接收

必须在停止接收 (SCR.RE 位 =0) 的状态下设定为模块停止状态或者向软件待机模式转移。对 RSR、RDR、SSR 寄存器进行复位。如果在接收时进行转移, 正在接收的数据就无效。

如果在解除低功耗状态后不更改接收模式而进行接收, 就必须通过将 RE 位置“1”, 开始接收。如果在更改接收模式后进行接收, 就必须在进行初始设定后开始接收。

接收时的模式转移的流程图例子如图 20.35 所示。

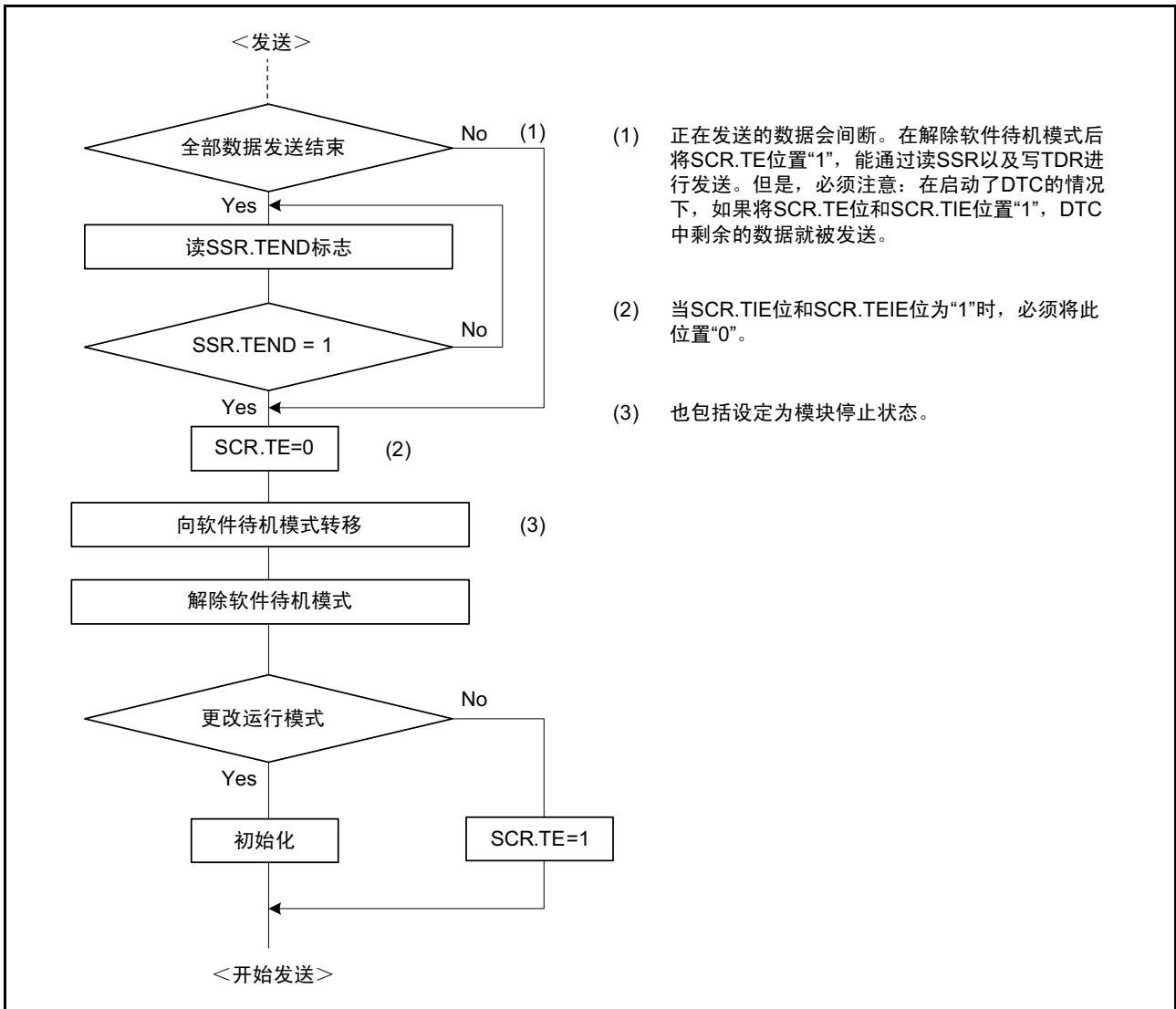


图 20.32 发送时向软件待机模式转移的流程图例子

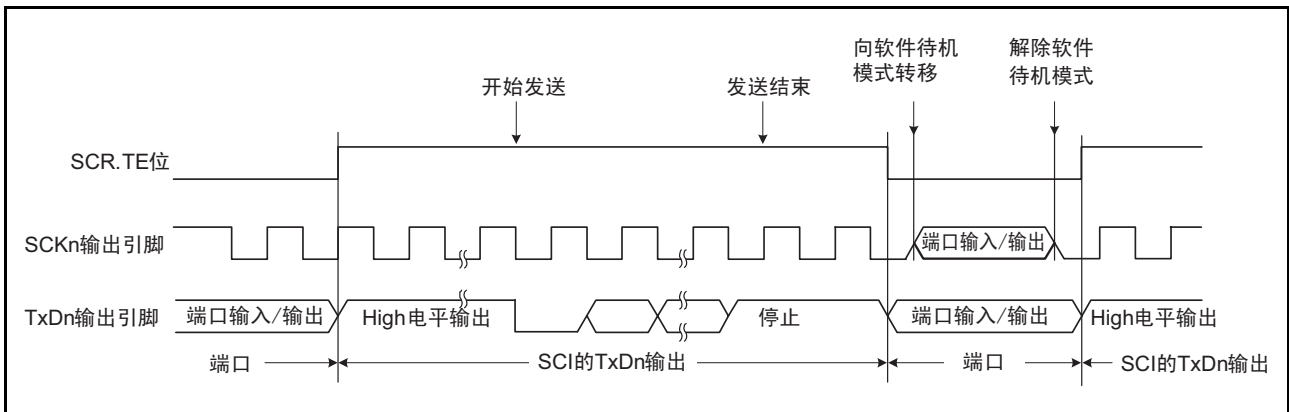


图 20.33 向软件待机模式转移时的端口引脚状态 (内部时钟、异步发送)

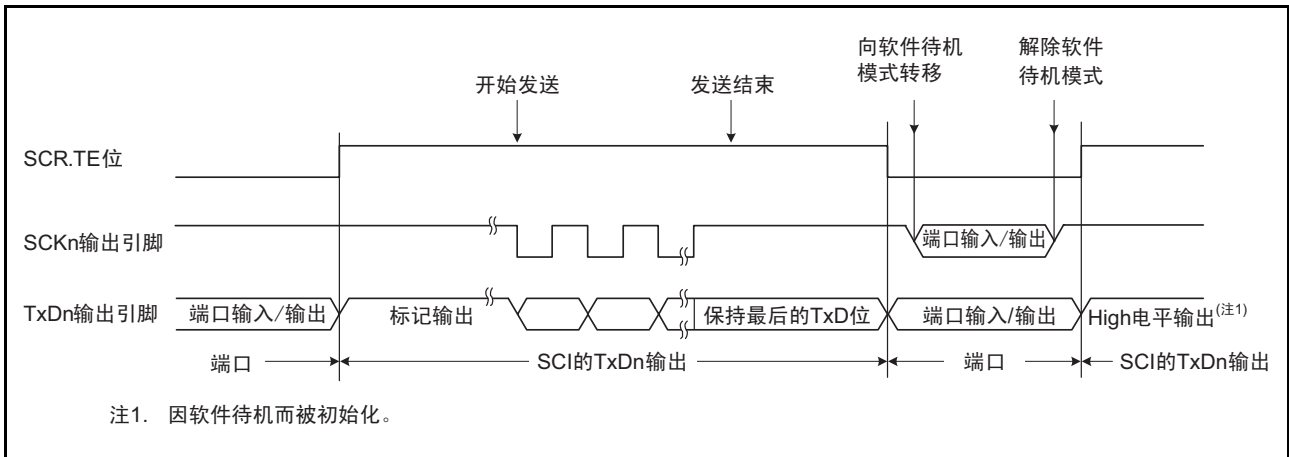


图 20.34 向软件待机模式转移时的端口引脚状态 (内部时钟、时钟同步发送)

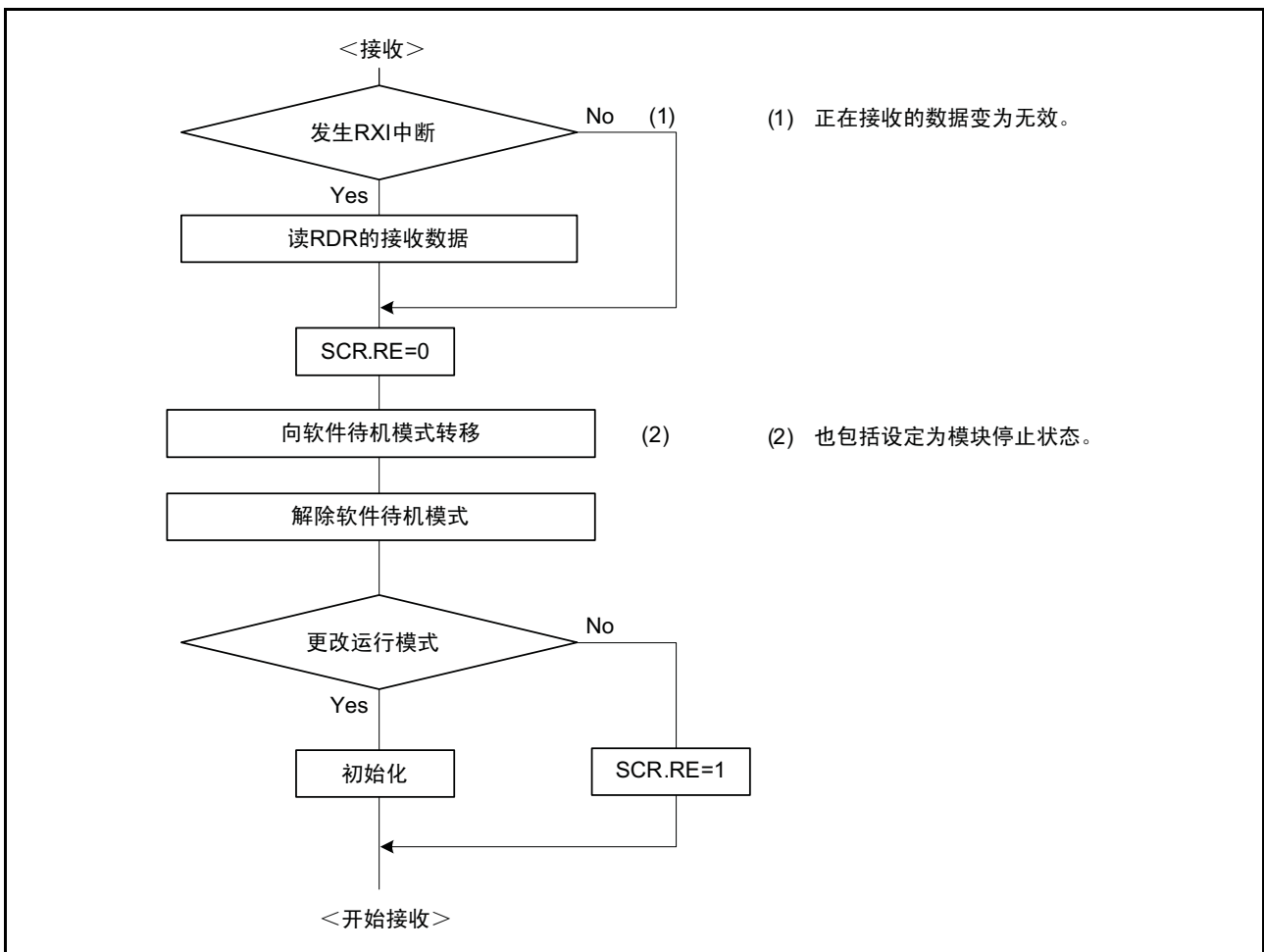


图 20.35 接收时向软件待机模式转移的流程图例子

20.7.9 时钟同步模式的外部时钟输入

在时钟同步模式中，外部时钟的 SCKn 输入必须至少为 2 个时钟的 High 电平脉冲期间和 Low 电平脉冲期间，或者 6 个时钟周期。

21. CRC 运算器 (CRC)

CRC (Cyclic Redundancy Check) 运算器生成数据块的 CRC 码。

21.1 概要

CRC 运算器的规格和框图分别如表 21.1 和图 21.1 所示。

表 21.1 CRC 运算器的规格

| 项目 | 内容 |
|-------------------|--|
| CRC 运算的对象数据 (注 1) | 对于 8n 位的数据, 生成 CRC 码 (n 为自然数)。 |
| 数据块的大小 | 8 位 |
| CRC 运算的处理方式 | 并行进行 8 位运算。 |
| CRC 生成多项式 | 可从 3 个多项式中选择: <ul style="list-style-type: none"> • 8 位 CRC X^8+X^2+X+1 • 16 位 CRC $X^{16}+X^{15}+X^2+1$ $X^{16}+X^{12}+X^5+1$ |
| CRC 运算的转换 | 能选择生成 LSB first 通信的 CRC 码或者生成 MSB first 通信的 CRC 码。 |
| 低功耗功能 | 能设定为模块停止状态。 |

注 1. 没有将运算对象数据分割成数据块的功能。必须以 8 位为单位进行写操作。

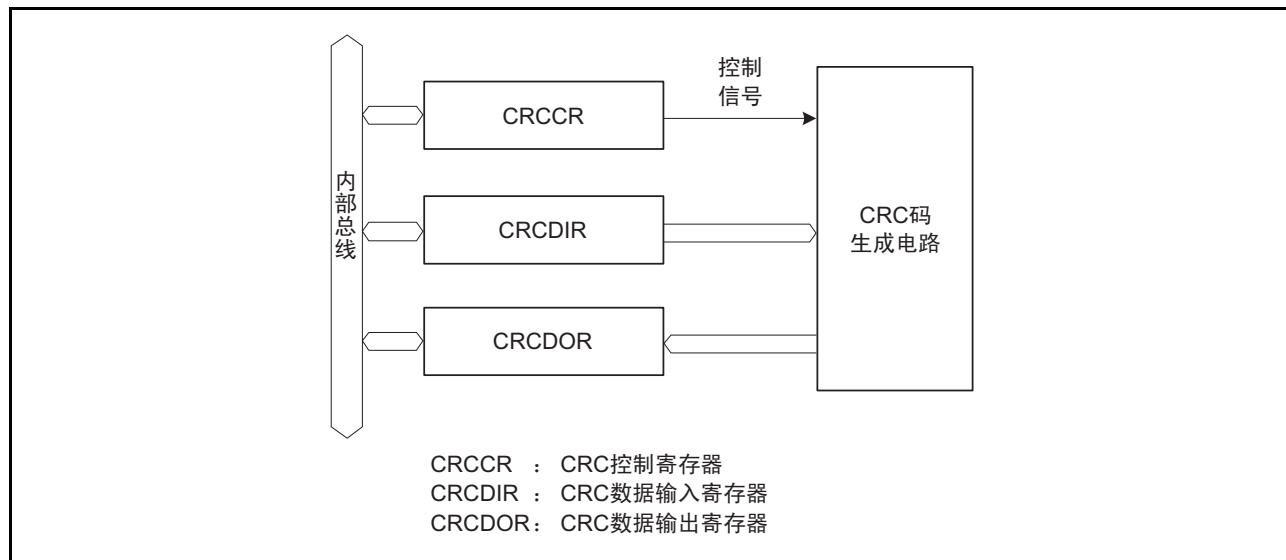


图 21.1 CRC 运算器的框图

21.2 寄存器说明

CRC 运算器的寄存器一览表如表 21.2 所示。

表 21.2 CRC 运算器的寄存器一览表

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|-------------|--------|-------|------------|------|
| CRC 控制寄存器 | CRCCR | 00h | 0008 8280h | 8 |
| CRC 数据输入寄存器 | CRCDIR | 00h | 0008 8281h | 8 |
| CRC 数据输出寄存器 | CRCDOR | 0000h | 0008 8282h | 16 |

21.2.1 CRC 控制寄存器 (CRCCR)

地址 0008 8280h

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|--------|----|----|----|----|-----|----------|----|
| DORCLR | — | — | — | — | LMS | GPS[1:0] | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|---------------|---|-----|
| b1-b0 | GPS[1:0] | CRC 生成多项式转换位 | $b1\ b0$ 0 0: 不运算 (注 1) 0 1: X^8+X^2+X+1 1 0: $X^{16}+X^{15}+X^2+1$ 1 1: $X^{16}+X^{12}+X^5+1$ | R/W |
| b2 | LMS | CRC 运算转换位 | 0: 进行 LSB first 通信时的 CRC 运算 在将 CRCDOR 寄存器的值 (CRC 码) 以字节为单 外分开发送时, 必须先发送低位字节 (b7 ~ b0)。 1: 进行 MSB first 通信时的 CRC 运算 在将 CRCDOR 寄存器的值 (CRC 码) 以字节为单 外分开发送时, 必须先发送高位字节 (b15 ~ b8)。 | R/W |
| b6-b3 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | DORCLR | CRCDOR 寄存器清除位 | 0: 不影响运行 1: 清除 CRCDOR 寄存器 读取值为“0”。 | W |

注 1. CRC 数据输出寄存器 (CRCDOR) 的值总是为“0000h”。

CRCCR 寄存器是选择 CRC 运算器的初始化、运算转换和生成多项式的寄存器。

GPS[1:0] 位 (CRC 生成多项式转换位)

这些位选择 CRC 码的生成多项式。

LMS 位 (CRC 运算转换位)

此位选择是生成 LSB first 通信的 CRC 码, 还是生成 MSB first 通信的 CRC 码。

DORCLR 位 (CRCDOR 寄存器清除位)

如果将 DORCLR 位置“1”, CRCDOR 寄存器就变为“0000h”。
读取值为“0”。

21.2.2 CRC 数据输入寄存器 (CRCDIR)

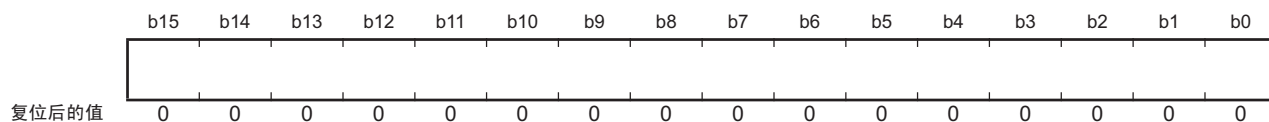
地址 0008 8281h



CRCDIR 寄存器是设定 CRC 运算对象数据块的 8 位可读写寄存器。

21.2.3 CRC 数据输出寄存器 (CRCDOR)

地址 0008 8282h



CRCDOR 寄存器是保存运算结果的 16 位可读写寄存器。

在一般情况下，如果在为了检查通信数据而在数据通信之后进行 CRC 码的运算时没有发生错误，CRCDOR 寄存器的值就为“0”。

如果使用 8 位 CRC (X^8+X^2+X+1 的多项式)，低位字节 (b7-b0) 就能得到有效的 CRC 码。高位字节 (b15-b8) 为“00h”。

21.3 CRC 运算器的运行说明

CRC 运算器生成用于 LSB first/MSB first 通信的 CRC 码。

假设 CRCCR.GPS[1:0] 位为 “11b”，使用 16 位 CRC ($X^{16}+X^{12}+X^5+1$ 的多项式)，将数据 “F0h” 生成 CRC 码的使用例子如下所示。

如果使用 8 位 CRC (X^8+X^2+X+1 的多项式)，CRCDOR 寄存器的低位字节就能得到有效的 CRC 码。

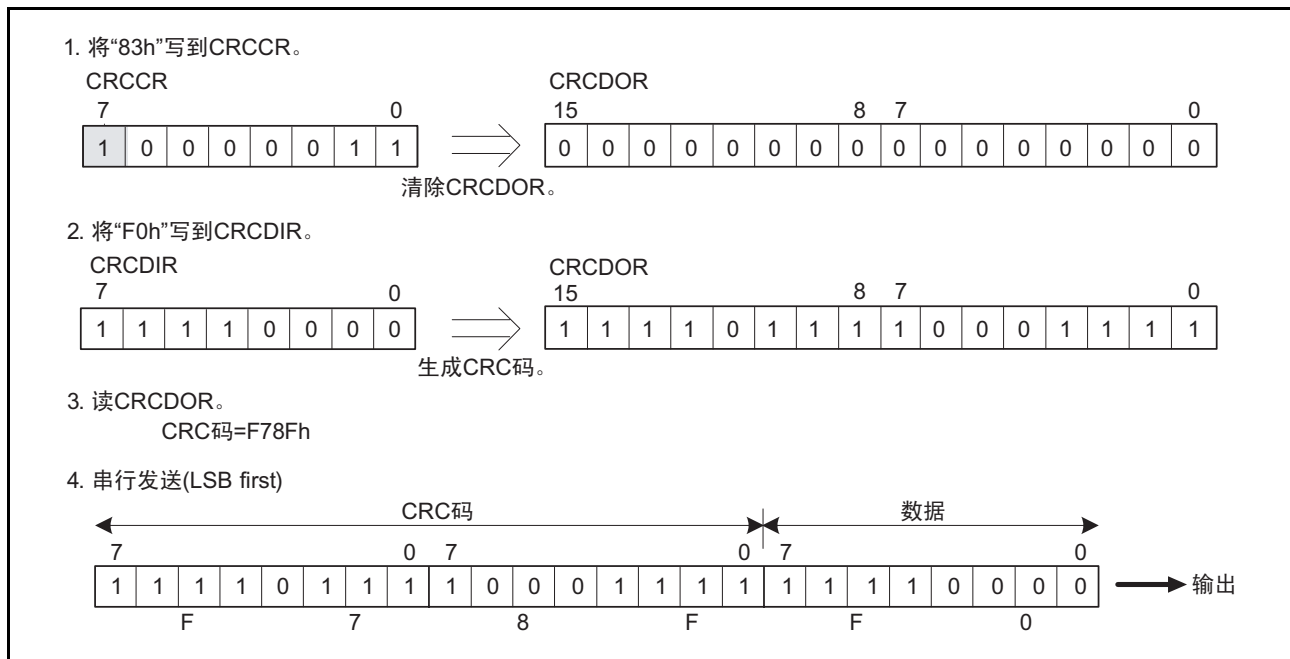


图 21.2 LSB first 的数据发送

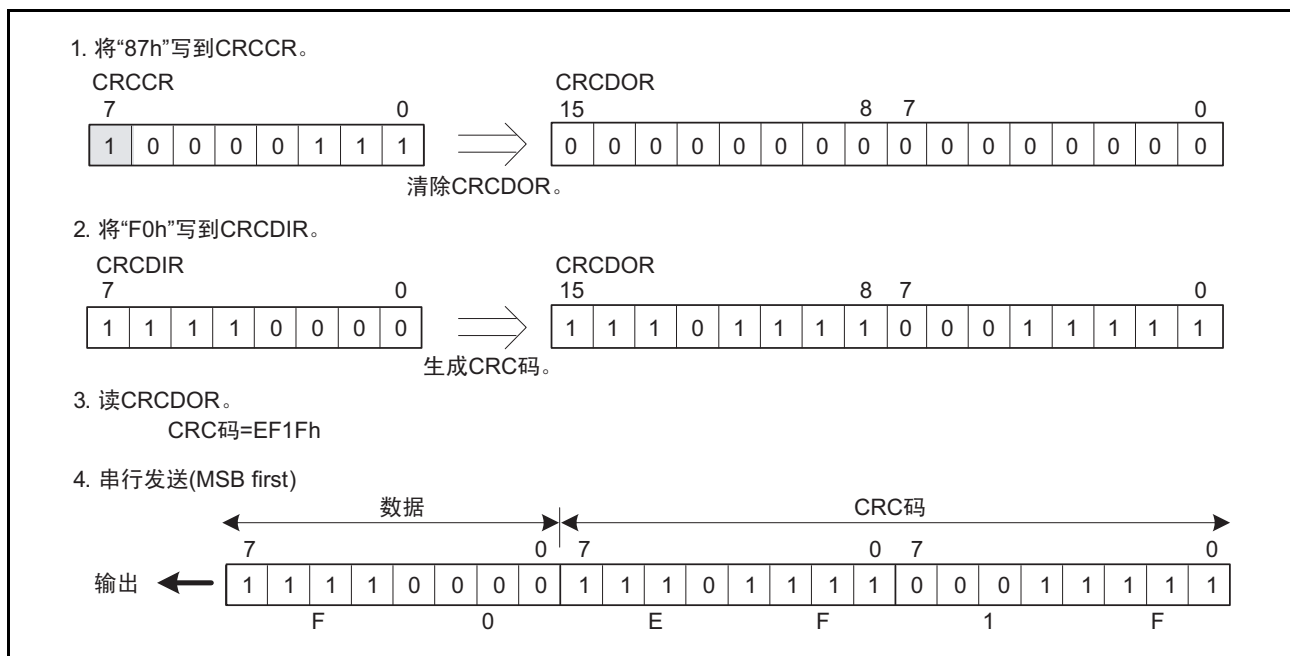


图 21.3 MSB first 的数据发送

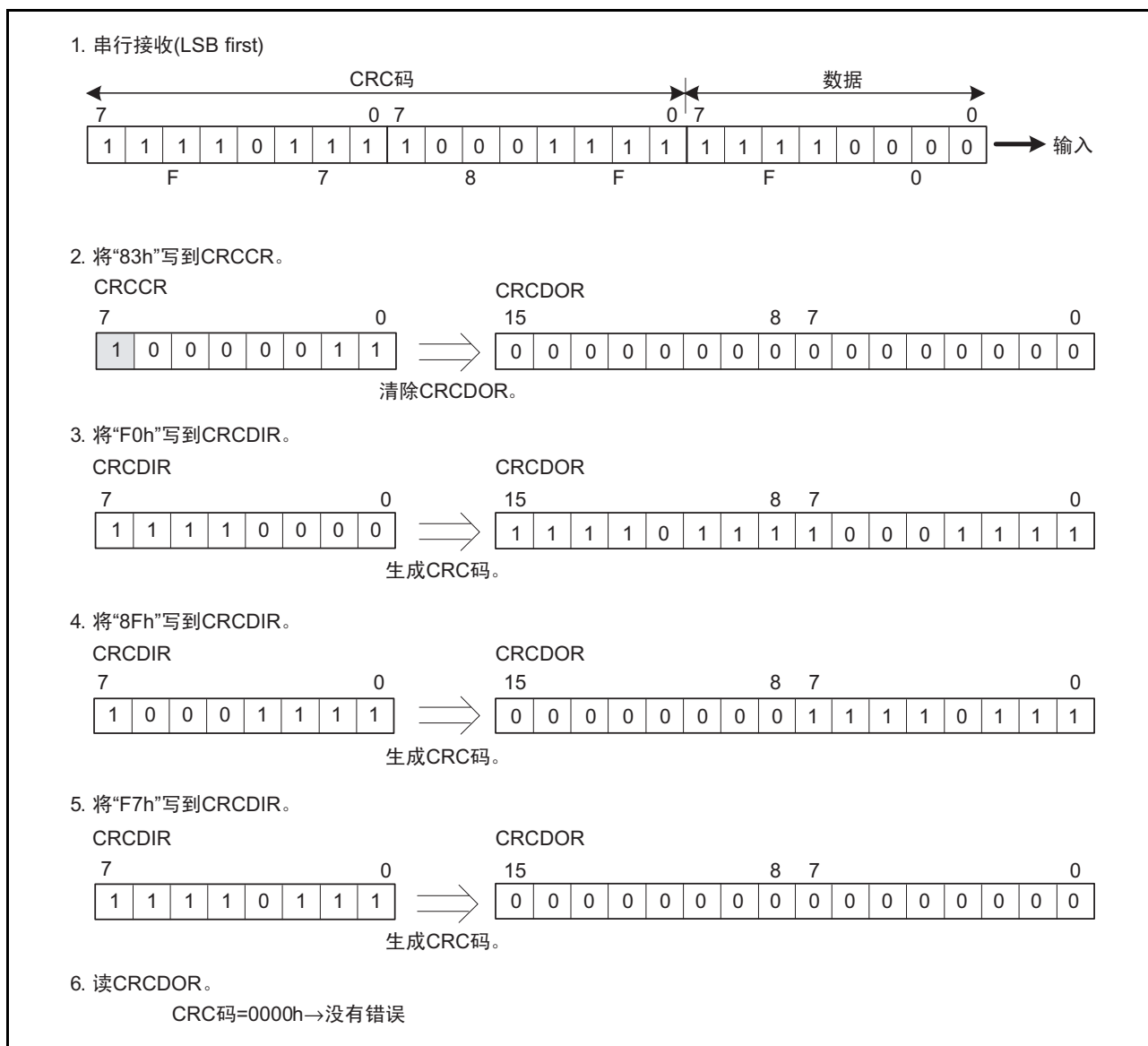


图 21.4 LSB first 的数据接收

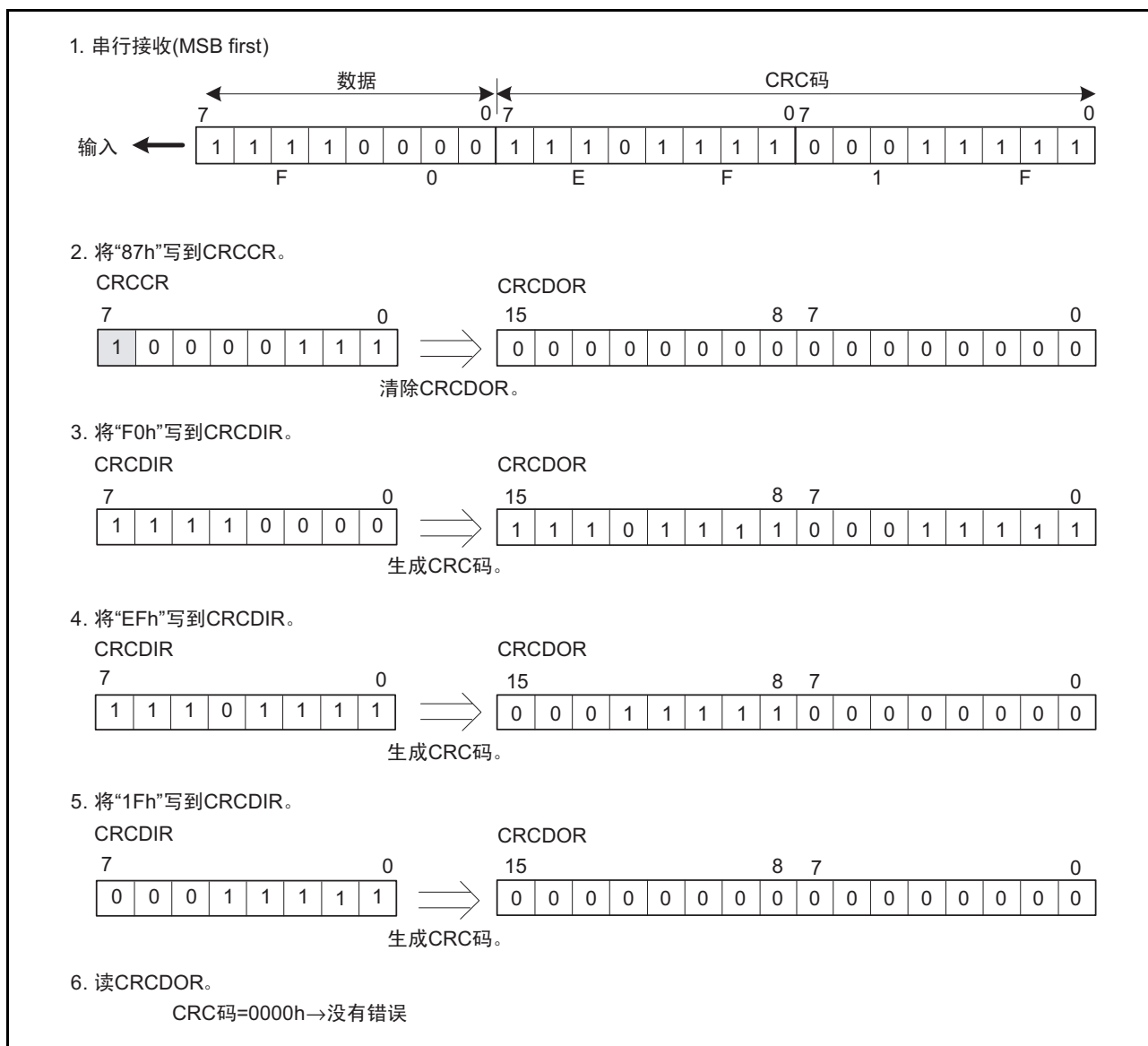


图 21.5 MSB first 的数据接收

21.4 使用时的注意事项

21.4.1 模块待机功能的设定

能通过模块待机控制寄存器 B (MSTPCRB) 禁止或者允许 CRC 运算器的运行, 初始值为停止 CRC 运算器的运行。能通过解除模块待机状态, 使寄存器变为可存取的状态。详细内容请参照 “8. 低功耗功能”。

21.5 传送时的注意事项

必须注意: LSB first 发送和 MSB first 发送时的 CRC 码的发送顺序不同。

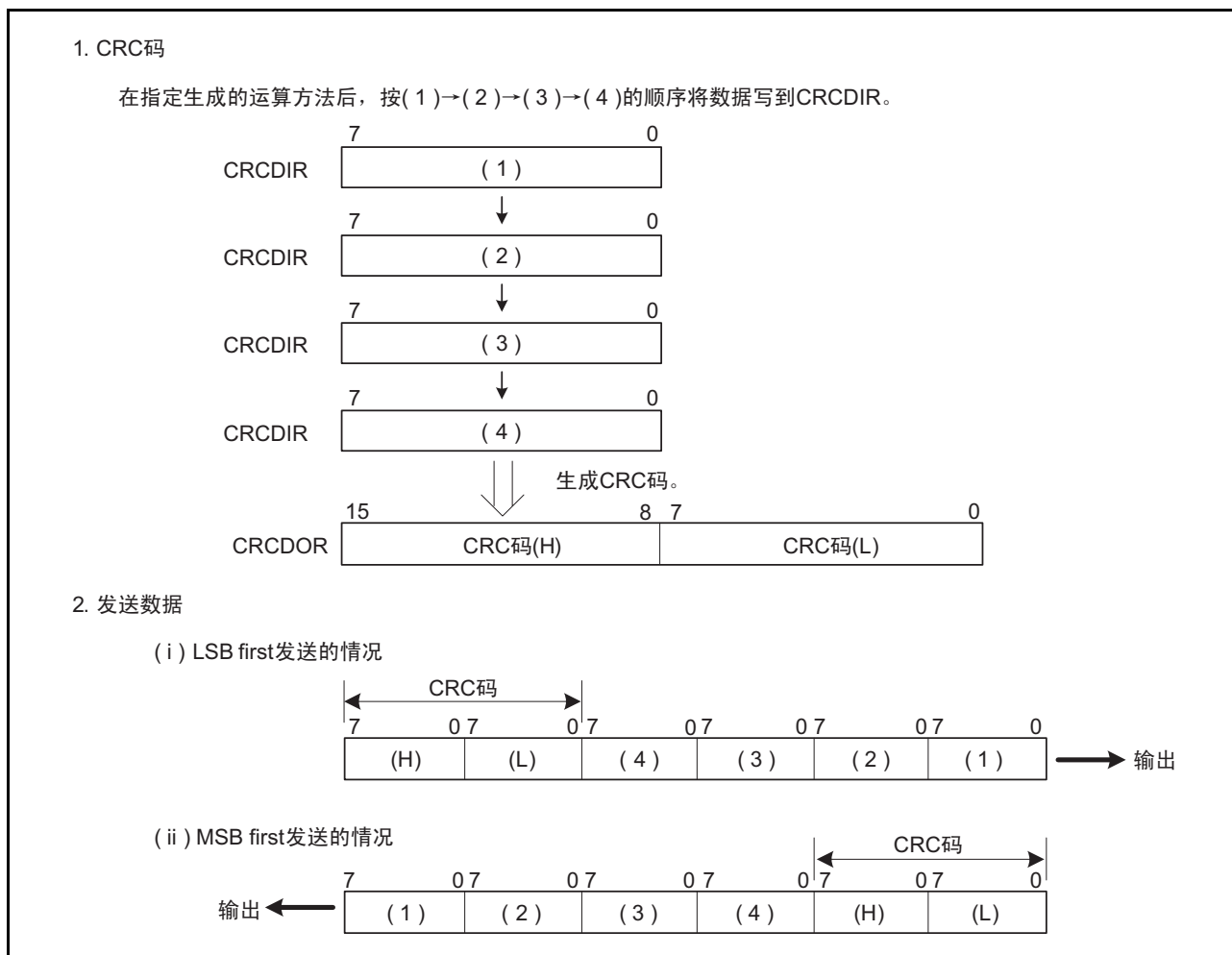


图 21.6 LSB first 和 MSB first 的发送数据

22. I²C 总线接口 (RIIC)

RX610 群内置 2 个通道的 I²C 总线接口 (RIIC)。

RIIC 以 NXP 公司提倡的 I²C 总线 (Inter-IC-Bus) 接口方式为基准, 装载了子集功能。

22.1 概要

RIIC 的规格和框图分别如表 22.1 和图 22.1 所示, 输入 / 输出引脚的外部电路连接例子 (I²C 总线的结构例子) 如图 22.2 所示, RIIC 使用的输入 / 输出引脚如表 22.2 所示。

表 22.1 RIIC 的规格

| 项目 | 内容 |
|------------|---|
| 通信格式 | <ul style="list-style-type: none"> • I²C 总线格式或者 SMBus 格式 • 能选择主控模式或者从属模式。 • 自动确保与所设的传送率对应的各种准备时间、保持时间和总线空闲时间。 |
| 传送率 | ~ 1Mbps |
| SCL 时钟 | 在 主控模式中, 能将 SCL 时钟的占空比设定在 4%~96% 的范围内。 |
| 条件的发行和检测 | 自动生成开始条件、重新开始条件和停止条件, 并且能检测到开始条件 (包括重新开始条件) 和停止条件。 |
| 从属地址 | <ul style="list-style-type: none"> • 能设定 3 组从属地址。 • 对应 7 位或者 10 位的地址格式 (能同时存在)。 • 能检测到全呼地址、设备 ID 地址和 SMBus 的主机地址。 |
| 应答 | <ul style="list-style-type: none"> • 在发送时自动加载应答位。 能在接收 NACK 时自动中止下一个发送数据的传送。 • 在接收时自动发送应答位。 <p>如果选择在第 8 个时钟和第 9 个时钟之间有等待, 就能通过软件控制与接收数据内容对应的应答位。</p> |
| 等待功能 | <ul style="list-style-type: none"> • 在接收时, 能通过保持 SCL 时钟的 Low 电平进行等待。 在第 8 个时钟和第 9 个时钟之间等待。 在第 9 个时钟和下次传送的第 1 个时钟之间等待 (WAIT 功能)。 |
| SDA 输出延迟功能 | 能延迟数据发送 (包括发送 ACK) 的输出时序。 |
| 仲裁 | <ul style="list-style-type: none"> • 对应多主控 在和其他主控发生 SCL 时钟冲突时, 能与 SCL 时钟同步运行。 在发生开始条件的发行竞争时, 如果 SDA 线上的信号状态不同, 就能检测到仲裁失败。 在 主控模式中, 能在发送数据不同时检测到仲裁失败。 • 如果在总线忙时发行开始条件, 就能检测到仲裁失败 (防止双重发行开始条件)。 • 在发送 NACK 时, 如果 SDA 线上的信号状态不同, 就能检测到仲裁失败。 • 在从属发送模式中, 能在数据不同时检测到仲裁失败。 |
| 超时检测功能 | 能通过内部超时检测功能检测 SCL 时钟的长时间停止。 |
| 噪声消除 | SCL 输入和 SDA 输入内置数字噪声滤波器, 噪声消除宽度为可编程调整。 |
| 中断源 | <ul style="list-style-type: none"> • 4 种 通信错误 / 事件的发生 (AL 检测、NACK 检测、超时检测、开始条件检测 (包括重新开始条件)、停止条件检测) 接收数据满 (包括从属地址匹配的情况) 发送数据空 (包括从属地址匹配的情况) 发送结束 |

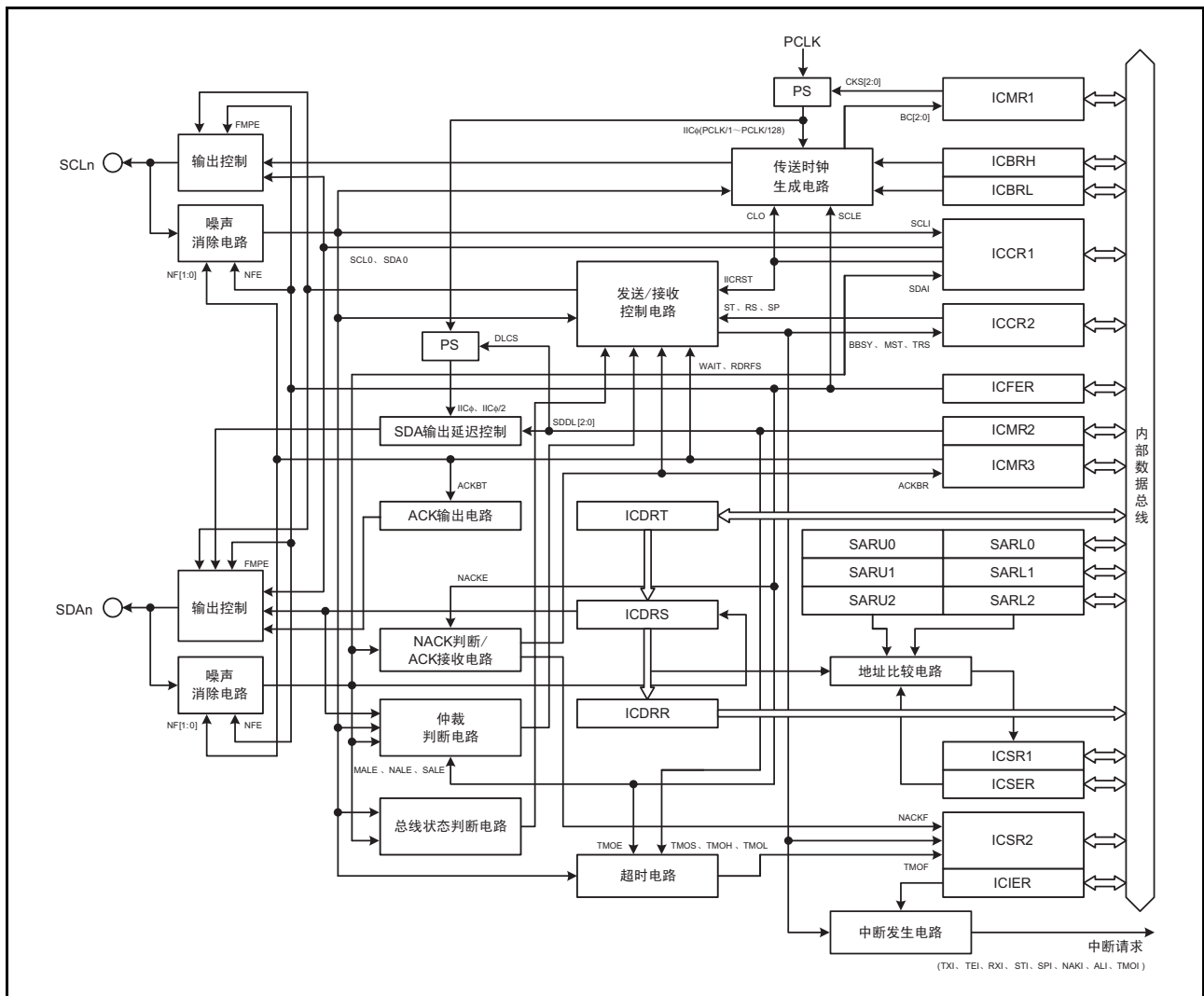


图 22.1 RIIC 的框图

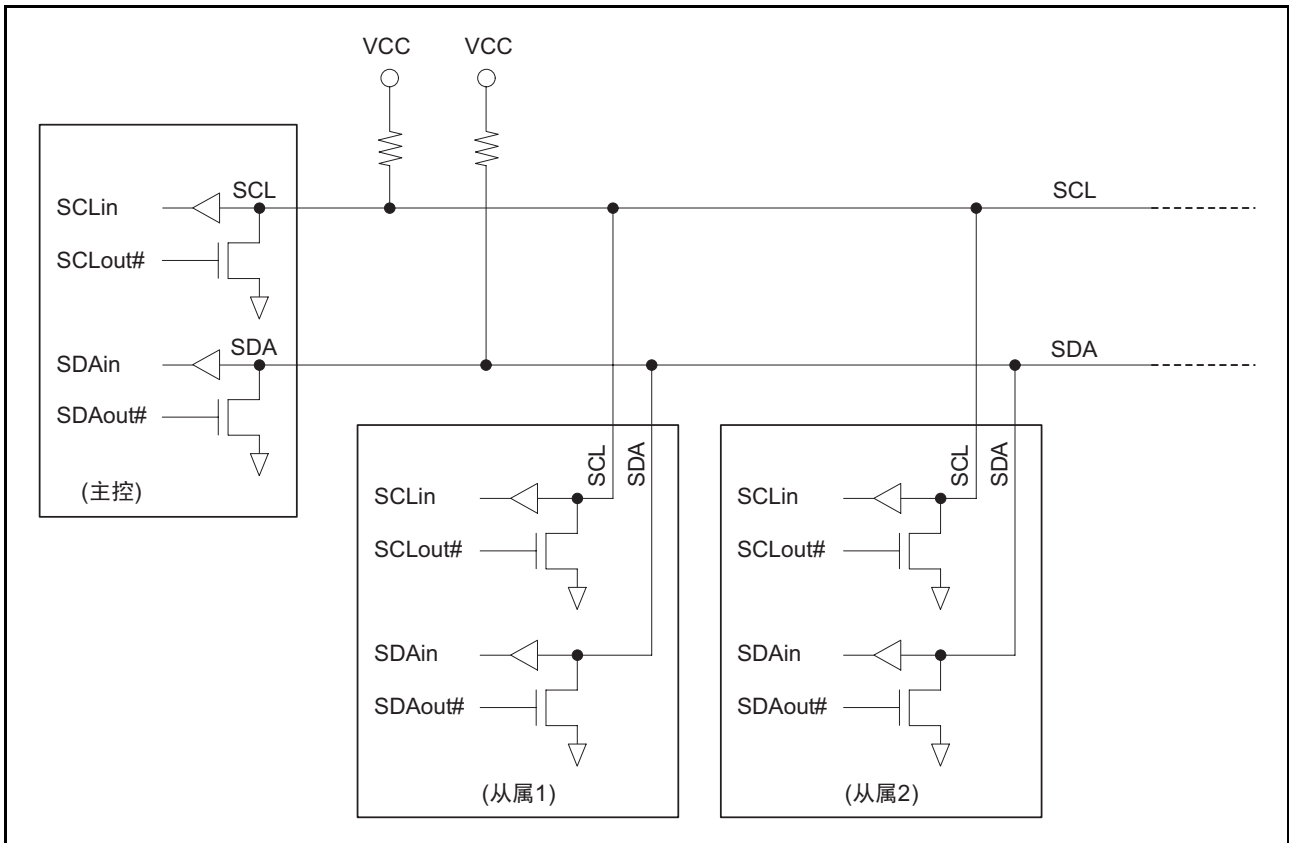


图 22.2 输入 / 输出引脚的外部电路连接例子 (I²C 总线的结构例子)

表 22.2 RIIC 的输入 / 输出引脚

| 通道 | 引脚名 | 输入 / 输出 | 功能 |
|-------|------|---------|------------------------|
| RIIC0 | SCL0 | 输入 / 输出 | RIIC0 侧的串行时钟的输入 / 输出引脚 |
| | SDA0 | 输入 / 输出 | RIIC0 侧的串行数据的输入 / 输出引脚 |
| RIIC1 | SCL1 | 输入 / 输出 | RIIC1 侧的串行时钟的输入 / 输出引脚 |
| | SDA1 | 输入 / 输出 | RIIC1 侧的串行数据的输入 / 输出引脚 |

22.2 寄存器说明

RIIC 的寄存器一览表如表 22.3 所示。

表 22.3 RIIC 的寄存器一览表

| 通道 | 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|-------|------------------------------|-------|-------|------------|------|
| RIIC0 | I ² C 总线控制寄存器 1 | ICCR1 | 1Fh | 0008 8300h | 8 |
| | I ² C 总线控制寄存器 2 | ICCR2 | 00h | 0008 8301h | 8 |
| | I ² C 总线模式寄存器 1 | ICMR1 | 08h | 0008 8302h | 8 |
| | I ² C 总线模式寄存器 2 | ICMR2 | 06h | 0008 8303h | 8 |
| | I ² C 总线模式寄存器 3 | ICMR3 | 00h | 0008 8304h | 8 |
| | I ² C 总线功能允许寄存器 | ICFER | 72h | 0008 8305h | 8 |
| | I ² C 总线状态允许寄存器 | ICSER | 09h | 0008 8306h | 8 |
| | I ² C 总线中断允许寄存器 | ICIER | 00h | 0008 8307h | 8 |
| | I ² C 总线状态寄存器 1 | ICSR1 | 00h | 0008 8308h | 8 |
| | I ² C 总线状态寄存器 2 | ICSR2 | 00h | 0008 8309h | 8 |
| | 从属地址寄存器 L0 | SARL0 | 00h | 0008 830Ah | 8 |
| | 从属地址寄存器 U0 | SARU0 | 00h | 0008 830Bh | 8 |
| | 从属地址寄存器 L1 | SARL1 | 00h | 0008 830Ch | 8 |
| | 从属地址寄存器 U1 | SARU1 | 00h | 0008 830Dh | 8 |
| | 从属地址寄存器 L2 | SARL2 | 00h | 0008 830Eh | 8 |
| | 从属地址寄存器 U2 | SARU2 | 00h | 0008 830Fh | 8 |
| | I ² C 总线位速率低电平寄存器 | ICBRL | FFh | 0008 8310h | 8 |
| | I ² C 总线位速率高电平寄存器 | ICBRH | FFh | 0008 8311h | 8 |
| | I ² C 总线发送数据寄存器 | ICDRT | FFh | 0008 8312h | 8 |
| | I ² C 总线接收数据寄存器 | ICDRR | 00h | 0008 8313h | 8 |
| | I ² C 总线移位寄存器 | ICDRS | — | — | 8 |
| RIIC1 | I ² C 总线控制寄存器 1 | ICCR1 | 1Fh | 0008 8320h | 8 |
| | I ² C 总线控制寄存器 2 | ICCR2 | 00h | 0008 8321h | 8 |
| | I ² C 总线模式寄存器 1 | ICMR1 | 08h | 0008 8322h | 8 |
| | I ² C 总线模式寄存器 2 | ICMR2 | 06h | 0008 8323h | 8 |
| | I ² C 总线模式寄存器 3 | ICMR3 | 00h | 0008 8324h | 8 |
| | I ² C 总线功能允许寄存器 | ICFER | 72h | 0008 8325h | 8 |
| | I ² C 总线状态允许寄存器 | ICSER | 09h | 0008 8326h | 8 |
| | I ² C 总线中断允许寄存器 | ICIER | 00h | 0008 8327h | 8 |
| | I ² C 总线状态寄存器 1 | ICSR1 | 00h | 0008 8328h | 8 |
| | I ² C 总线状态寄存器 2 | ICSR2 | 00h | 0008 8329h | 8 |
| | 从属地址寄存器 L0 | SARL0 | 00h | 0008 832Ah | 8 |
| | 从属地址寄存器 U0 | SARU0 | 00h | 0008 832Bh | 8 |
| | 从属地址寄存器 L1 | SARL1 | 00h | 0008 832Ch | 8 |
| | 从属地址寄存器 U1 | SARU1 | 00h | 0008 832Dh | 8 |
| | 从属地址寄存器 L2 | SARL2 | 00h | 0008 832Eh | 8 |
| | 从属地址寄存器 U2 | SARU2 | 00h | 0008 832Fh | 8 |
| | I ² C 总线位速率低电平寄存器 | ICBRL | FFh | 0008 8330h | 8 |
| | I ² C 总线位速率高电平寄存器 | ICBRH | FFh | 0008 8331h | 8 |
| | I ² C 总线发送数据寄存器 | ICDRT | FFh | 0008 8332h | 8 |
| | I ² C 总线接收数据寄存器 | ICDRR | 00h | 0008 8333h | 8 |
| | I ² C 总线移位寄存器 | ICDRS | — | — | 8 |

22.2.1 I²C 总线控制寄存器 1 (ICCR1)

地址 RIIC0.ICCR1 0008 8300h、RIIC1.ICCR1 0008 8320h

| | | | | | | | |
|-----|--------|-----|------|------|------|------|------|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| ICE | IICRST | CLO | SOWP | SCLO | SDAO | SCLI | SDAI |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|----|--------|-----------------------------|---|----------------------|
| b0 | SDAI | SDA 总线输入监视位 | 0: SDA _n 引脚输入为 Low 电平 1: SDA _n 引脚输入为 High 电平 | R |
| b1 | SCLI | SCL 总线输入监视位 | 0: SCL _n 引脚输入为 Low 电平 1: SCL _n 引脚输入为 High 电平 | R |
| b2 | SDAO | SDA 输出控制位 | <ul style="list-style-type: none"> 读时 0: SDA_n 引脚为 Low 电平输出 1: SDA_n 引脚为高阻抗 写时 0: 将 SDA_n 引脚改为 Low 电平 1: 将 SDA_n 引脚改为高阻抗 (通过外部上拉电阻输出 High 电平) | R/W (注 1、 注 2) |
| b3 | SCLO | SCL 输出控制位 | <ul style="list-style-type: none"> 读时 0: SCL_n 引脚为 Low 电平输出 1: SCL_n 引脚为高阻抗 写时 0: 将 SCL_n 引脚改为 Low 电平输出 1: 将 SCL_n 引脚改为高阻抗 (通过外部上拉电阻输出 High 电平) | R/W (注 1、 注 2) |
| b4 | SOWP | SCLO/SDAO 写保护位 | 0: 设定 SCLO 位和 SDAO 位的值 (读取值为“1”。) | R/W (注 2) |
| b5 | CLO | SCL 时钟的追加输出位 | 0: 不追加输出 SCL 时钟 (正常状态) 1: 追加输出 SCL 时钟 (在输出 1 个时钟后, 此位自动变为“0”。) | R/W |
| b6 | IICRST | I ² C 总线接口内部复位的位 | 0: 解除 RIIC 复位或者内部复位 1: RIIC 复位或者内部复位状态 (清除位计数器并且解除 SCL _n /SDA _n 的输出锁存) | R/W |
| b7 | ICE | I ² C 总线接口允许位 | 0: RIIC 的功能停止 (SCL _n 引脚 /SDA _n 引脚为端口功能) 1: RIIC 能进行传送 (SCL _n 引脚 /SDA _n 引脚为总线驱动功能) | R/W |

注 1. 不能在通信过程中写此位。如果在通信过程中更改此位的值, 就可能发生发送 / 接收异常或者 AL 错误。

注 2. 必须在将 SOWP 位置“0”的同时改写 SDAO 位和 SCLO 位。

ICCR1 寄存器控制 RIIC 的运行 / 停止、RIIC 内部状态的复位、SCL 时钟的追加输出、SCL_n 引脚 /SDA_n 引脚的操作以及 SCL_n 引脚 /SDA_n 引脚的监视等。

SDAI 位 (SDA 总线输入监视位)

此位表示 SDA_n 引脚的输入电平。

SCLI 位 (SCL 总线输入监视位)

此位表示 SCL_n 引脚的输入电平。

SDAO 位 (SDA 输出控制位)

此位控制 SDA_n 引脚的输出电平并且表示此引脚的输出状态。

SCLO 位 (SCL 输出控制位)

此位控制 SCL_n 引脚的输出电平并且表示此引脚的输出状态。

SOWP 位 (SCLO/SDAO 写保护位)

此位控制 SCLO 位和 SDAO 位的写操作。

CLO 位 (SCL 时钟的追加输出位)

此位具有以 1 个时钟为单位追加输出 SCL 时钟的功能，用于调试或者异常处理。

在一般情况下，必须将此位“0”。如果在正常的通信过程中使用此位，就可能引起通信错误。

有关此功能的详细内容，请参照“22.11.2 SCL 时钟追加输出功能”。

IICRST 位 (I²C 总线内部复位的位)

此位对 RIIC 的内部状态进行复位。

如果将 IICRST 位置“1”，就能进行 RIIC 复位或者内部复位。

RIIC 复位和内部复位取决于 IICRST 位和 ICE 位的组合。RIIC 复位的种类如表 22.4 所示。

RIIC 复位是指对包括 ICCR2.BBSY 标志在内的全部寄存器和内部状态进行复位；内部复位是指对位计数器 (ICMR1.BC[2:0] 位)、I²C 总线移位寄存器 (ICDRS)、I²C 总线状态寄存器 (ICSR1 和 ICSR2) 和内部状态进行复位。有关各寄存器的复位状况，请参照“22.14 复位状况”。

如果在运行时 (ICE 位为“1”的状态) 因通信故障等引起总线和 RIIC 发生意外停机的情况下将 IICRST 位置“1”，就能在不对端口的设定、RIIC 的各控制寄存器和设定寄存器进行初始化的情况下对 RIIC 的内部状态进行复位。

如果在 RIIC 输出 Low 电平的状态下发生意外停机，就能通过对内部状态进行复位，将 SCL_n 引脚 /SDA_n 引脚置为高阻抗，然后释放总线。

注 1. 在从属模式中和主控器件进行通信时，如果因总线发生意外停机而通过 IICRST 位进行内部复位，RIIC 就可能变为和主控器件的状态不同的状态（主要是双方的位计数器信息产生差异），因此原则上不在从属模式中进行内部复位而从主控器件进行恢复处理。在从属模式中将 SCL_n 线置为 Low 电平输出的状态下，如果 RIIC 发生意外停机而需要内部复位，就必须在内部复位后从主控器件发行重新开始条件，或者在发行停止条件后发行开始条件，重新开始通信。如果只单独对从属器件进行复位，并且在没有从主控器件发行开始条件或者重新开始条件的情况下重新开始通信，就有可能因双方运行状态的差异而导致不同步。

表 22.4 RIIC 复位的种类

| IICRST | ICE | 状态 | 内容 |
|--------|-----|---------|---|
| 1 | 0 | RIIC 复位 | 对 RIIC 的全部寄存器和内部状态进行复位。 |
| | 1 | 内部复位 | 对 ICMR1.BC[2:0] 位、ICSR1、ICSR2、ICDRS 寄存器和内部状态进行复位。 |

ICE 位 (I²C 总线接口允许位)

此位选择 RIIC 是能进行传送还是功能停止。

如果将 ICE 位置“0” (功能停止)，SCL_n 引脚 /SDA_n 引脚就为端口功能。如果在 ICE 位为“0”时将 IICRST 位置“1”，就进行 RIIC 复位；如果在 ICE 位为“1”时将 IICRST 位置“1”，就进行内部复位。

另外，为了防止意想不到的通信开始，必须在设定 RIIC 的寄存器时将 ICE 位置“0” (功能停止) 并且在设定完全部寄存器后将 ICE 位置“1” (能传送状态)。

注 1. RX610 群的引脚功能也分配了 I²C 总线引脚以外的其他功能。在用作 I²C 总线引脚 (SCL_n 引脚 /SDA_n 引脚) 时，必须禁止其他功能的输出。因为 I²C 总线引脚的 SCL_n 引脚 /SDA_n 引脚都为输入 / 输出引脚，因此必须将对应的 I/O 端口的 Pm.DDR 寄存器和 Pm.ICR 寄存器分别设定为“0” (输入) 和“1” (输入缓冲器有效)。

22.2.2 I²C 总线控制寄存器 2 (ICCR2)

地址 RIIC0.ICCR2 0008 8301h、RIIC1.ICCR2 0008 8321h

| | | | | | | | | |
|-------|------|-----|-----|----|----|----|----|----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | BBSY | MST | TRS | — | SP | RS | ST | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|------|-------------|---|-------------|
| b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b1 | ST | 开始条件发行请求位 | 0: 不请求发行开始条件 1: 请求发行开始条件 | R/W |
| b2 | RS | 重新开始条件发行请求位 | 0: 不请求发行重新开始条件 1: 请求发行重新开始条件 | R/W |
| b3 | SP | 停止条件发行请求位 | 0: 不请求发行停止条件 1: 请求发行停止条件 | R/W |
| b4 | — | 保留位 | 读写值都为“0”。 | R/W |
| b5 | TRS | 发送 / 接收模式位 | 0: 接收模式 1: 发送模式 | R/W (注1) |
| b6 | MST | 主控 / 从属模式位 | 0: 从属模式 1: 主控模式 | R/W (注1) |
| b7 | BBSY | 总线忙检测标志 | 0: I ² C 总线为释放状态 (总线空闲状态) 1: I ² C 总线为占有状态 (总线忙状态或者总线处于空闲期间) | R |

注 1. 当 ICMR1.MTWP 位为“1”时，能写 MST 位和 TRS 位。

ICCR2 寄存器是控制 I²C 总线接口的寄存器，表示 I²C 总线占有状态或者释放状态。

ST 位 (开始条件发行请求位)

此位请求向主控模式的转移以及开始条件的发行。

如果 ST 位为“1”，就请求发行开始条件，并且在 BBSY 标志为“0” (总线空闲) 时发行开始条件。有关发行开始条件的详细内容，请参照“22.10 开始条件、重新开始条件和停止条件的发行功能”。

[为“1”的条件]

- 写“1”时

[为“0”的条件]

- 写“0”时
- 结束开始条件的发行时
- ICSR2.AL 标志变为“1” (仲裁失败) 时
- 给 ICCR1.IICRST 位写“1”，并且进行 RIIC 复位或者内部复位时

注 1. 必须在 BBSY 标志为“0” (总线空闲) 时将 ST 位置“1” (请求发行开始条件)。

必须注意: 如果在 BBSY 标志为“1” (总线忙) 时将 ST 位置“1” (请求发行开始条件)，就作为开始条件的发行错误而产生仲裁失败。

RS 位 (重新开始条件发行请求位)

此位在主控模式中请求发行重新开始条件。

如果 RS 位为“1”，就请求发行重新开始条件，在 BBSY 标志为“1”（总线忙）并且 MST 位为“1”（主控模式）时，发行重新开始条件。

有关发行重新开始条件的详细内容，请参照“22.10 开始条件、重新开始条件和停止条件的发行功能”。

[为“1”的条件]

- 在 ICCR2.BBSY 标志为“1”的状态下写“1”时

[为“0”的条件]

- 写“0”时
- 结束重新开始条件的发行或者检测到开始条件时
- ICSR2.AL 标志变为“1”（仲裁失败）时
- 给 ICCR1.IICRST 位写“1”，进行 RIIC 复位或者内部复位时

注 1. 不能在发行停止条件过程中将 RS 位置“1”。

注 2. 必须注意：如果在主控模式以外的模式中给 RS 位写“1”（请求发行重新开始条件），就不能在该运行模式中发行重新开始条件而 RS 位保持“1”。如果在此状态下将运行模式转移到主控模式，就可能发行重新开始条件。

SP 位 (停止条件发行请求位)

此位在主控模式中请求发行停止条件。

如果 SP 位为“1”，就请求发行停止条件，在 BBSY 标志为“1”（总线忙）并且 MST 位为“1”（主控模式）时，发行停止条件。

有关发行停止条件的详细内容，请参照“22.10 开始条件、重新开始条件和停止条件的发行功能”。

[为“1”的条件]

- 在 ICCR2.BBSY 标志为“1”并且 ICCR2.MST 位为“1”的状态下写“1”时

[为“0”的条件]

- 写“0”时
- 结束停止条件的发行或者检测到停止条件时
- ICSR2.AL 标志变为“1”（仲裁失败）时
- 检测到开始条件和重新开始条件时
- 给 ICCR1.IICRST 位写“1”，并且进行 RIIC 复位或者内部复位时

注 1. 不能在 BBSY 标志为“0”（总线空闲）时写此位。

注 2. 不能在发行重新开始条件过程中将 SP 位置“1”。

TRS 位 (发送 / 接收模式位)

此位是表示发送模式或者接收模式的位。

当 TRS 位为“0”时，为接收模式；当 TRS 位为“1”时，为发送模式。通过和 MST 位的组合，表示 RIIC 的运行模式。

TRS 位通过开始条件的发行和检测以及 R/W# 位等变为“1”或者“0”，并且自动转移到发送模式或者接收模式。当 ICMR1.MTWP 位为“1”时，能写 TRS 位，但是一般不需要写。

[为“1”的条件]

- 根据开始条件的发行请求，正常地发行了开始条件（在 ST 位为“1”的状态下检测到开始条件）时
- 在主控模式中，附加到从属地址的 R/W# 位为“0”时
- 在从属模式中，接收的从属地址与 ICSER 寄存器的有效地址匹配，并且接收到为“1”的 R/W# 位时
- 在 ICMR1.MTWP 位为“1”的状态下写“1”时

[为“0”的条件]

- 检测到停止条件时
- ICSR2.AL标志变为“1”(仲裁失败)时
- 在主控模式中,附加到从属地址的R/W#位为“1”时
- 在从属模式中,接收的从属地址与ICSER寄存器的有效地址匹配,并且接收到为“0”的R/W#位(包括全呼地址)时
- 在从属模式中,检测到重新开始条件(在ICCR2.BBSY位“1”并且ICCR2.MST为“0”的状态下检测到开始条件)时
- 在ICMR1.MTWP位为“1”的状态下写“0”时
- 给ICCR1.IICRST位写“1”,进行RIIC复位或者内部复位时

MST位(主控/从属模式位)

此位是表示主控模式或者从属模式的位。

当MST位为“0”时,为从属模式;当MTS位为“1”时,为主控模式。通过和TRS位的组合,表示RIIC的运行模式。

通过开始条件的发行以及停止条件的发行和检测,MST位变为“1”或者“0”,并且自动转移到主控模式或者从属模式。当ICMR1.MTWP位为“1”时能写MST位,但是一般不需要写。

[为“1”的条件]

- 根据开始条件的发行请求,正常地发行了开始条件(在ST位为“1”的状态下检测到开始条件)时
- 在ICMR1.MTWP位为“1”的状态下写“1”时

[为“0”的条件]

- 检测到停止条件时
- ICSR2.AL标志变为“1”(仲裁失败)时
- 在ICMR1.MTWP位为“1”的状态下写“0”时
- 给ICCR1.IICRST位写“1”,进行RIIC复位或者内部复位时

BBSY标志(总线忙检测标志)

此标志表示I²C总线的占有(总线忙)状态或者释放状态(总线空闲)。

如果在SCLn线为High电平的状态下SDAn线从High电平变为Low电平,就认为发行了开始条件,此标志变为“1”。

如果在SCLn线为High电平的状态下SDAn线从Low电平变为High电平,就认为发行了停止条件,在没有检测到总线空闲时间(ICBRL寄存器的设定时间)的开始条件时,此标志变为“0”。

[为“1”的条件]

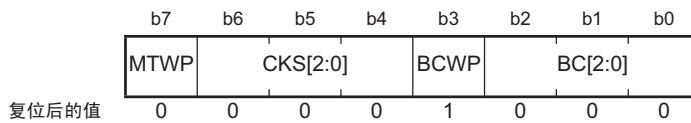
- 检测到开始条件时

[为“0”的条件]

- 在检测到停止条件后没有检测到总线空闲时间(ICBRL寄存器的设定时间)的开始条件时
- 在ICCR1.IICE位为“0”的状态下给ICCR1.IICRST位写“1”(RIIC复位)时

22.2.3 I²C 总线模式寄存器 1 (ICMR1)

地址 RIIC0.ICMR1 0008 8302h、RIIC1.ICMR1 0008 8322h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|--------------|--|--------------|
| b2-b0 | BC[2:0] | 位计数器 | b2 b0 0 0 0: 9 位 0 0 1: 2 位 0 1 0: 3 位 0 1 1: 4 位 1 0 0: 5 位 1 0 1: 6 位 1 1 0: 7 位 1 1 1: 8 位 | R/W (注 1) |
| b3 | BCWP | BC 写保护位 | 0: 允许设定 BC[2:0] 的值 (读取值为“1”) | R/W (注 1) |
| b6-b4 | CKS[2:0] | 内部基准时钟选择位 | b6 b4 0 0 0: PCLK/1 时钟 0 0 1: PCLK/2 时钟 0 1 0: PCLK/4 时钟 0 1 1: PCLK/8 时钟 1 0 0: PCLK/16 时钟 1 0 1: PCLK/32 时钟 1 1 0: PCLK/64 时钟 1 1 1: PCLK/128 时钟 | R/W |
| b7 | MTWP | MST/TRS 写保护位 | 0: 禁止写 ICCR2.MST 位和 ICCR2.TRS 位 1: 允许写 ICCR2.MST 位和 ICCR2.TRS 位 | R/W |

注 1. 必须在将 BCWP 位置“0”后, 使用 MOV 指令改写 BC[2:0] 位。

ICMR1 寄存器选择 RIIC 的内部基准时钟源, 取传送位数信息以及控制 ICCR2.MST 位和 ICCR2.TRS 位的写保护。

BC[2:0] 位 (位计数器)

这些位是在 SCLn 线的上升沿进行递减计数的计数器。如果读这些位, 就能得知剩下的传送位数。能读写这些位, 但是一般不需要存取。

写这些位时, 必须指定要传送的数据位数 +1 (数据附加 1 位应答位后传送), 并且在传送帧期间以及 SCLn 线为 Low 电平的状态下进行。

在结束包含应答的数据传送或者检测到开始条件 (包括重新开始条件) 时, BC[2:0] 位自动返回“000b”。

BCWP 位 (BC 写保护位)

此位控制 BC[2:0] 位的写操作。

CKS[2:0] 位 (内部基准时钟选择位)

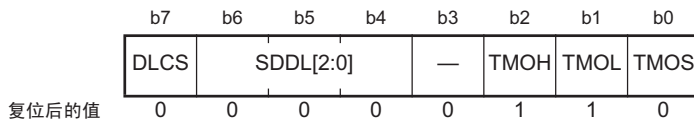
这些位选择 RIIC 的内部基准时钟源 (IIC ϕ)。

MTWP 位 (MST/TRS 写保护位)

此位控制 ICCR2.MST 位和 ICCR2.TRS 位的写操作。

22.2.4 I²C 总线模式寄存器 2 (ICMR2)

地址 RIIC0.ICMR2 0008 8303h、RIIC1.ICMR2 0008 8323h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-----------|----------------|---|-----|
| b0 | TMOS | 超时检测时间选择位 | 0: 选择长模式 1: 选择短模式 | R/W |
| b1 | TMOL | 超时 L 电平计数控制位 | 0: 在 SCLn 线为 Low 电平时, 禁止计数。 1: 在 SCLn 线为 Low 电平时, 允许计数。 | R/W |
| b2 | TMOH | 超时 H 电平计数控制位 | 0: 在 SCLn 线为 High 电平时, 禁止计数。 1: 在 SCLn 线为 High 电平时, 允许计数。 | R/W |
| b3 | — | 保留位 | 读写值都为“0”。 | R/W |
| b6-b4 | SDDL[2:0] | SDA 输出延迟计数器 | <ul style="list-style-type: none"> • 当 ICMR2.DLCS=0 (IICϕ) 时 <li style="margin-left: 20px;">b6 b4 0 0 0: 无输出延迟 0 0 1: 1 个 IICϕ 周期 0 1 0: 2 个 IICϕ 周期 0 1 1: 3 个 IICϕ 周期 1 0 0: 4 个 IICϕ 周期 1 0 1: 5 个 IICϕ 周期 1 1 0: 6 个 IICϕ 周期 1 1 1: 7 个 IICϕ 周期 • 当 ICMR2.DLCS=1 (IICϕ/2) 时 <li style="margin-left: 20px;">b6 b4 0 0 0: 无输出延迟 0 0 1: 1 ~ 2 个 IICϕ 周期 0 1 0: 3 ~ 4 个 IICϕ 周期 0 1 1: 5 ~ 6 个 IICϕ 周期 1 0 0: 7 ~ 8 个 IICϕ 周期 1 0 1: 9 ~ 10 个 IICϕ 周期 1 1 0: 11 ~ 12 个 IICϕ 周期 1 1 1: 13 ~ 14 个 IICϕ 周期 | R/W |
| b7 | DLCS | SDA 输出延迟时钟源选择位 | 0: 选择内部基准时钟 (IIC ϕ) 为 SDA 输出延迟计数器的时钟源 1: 选择内部基准时钟的 2 分频时钟 (IIC ϕ /2) 为 SDA 输出延迟计数器的时钟源 | R/W |

ICMR2 寄存器是具有超时检测功能和 SDA 输出延迟功能的寄存器。

TMOS 位 (超时检测时间选择位)

此位是在超时检测功能有效时 (ICFER.TMOE 位 =1) 选择超时检测时间的位。如果将此位置“0”, 就为长模式; 如果置“1”, 就为短模式。在长模式中, 用于检测超时的内部计数器用作 16 位计数器; 在短模式中, 此内部计数器用作 14 位计数器。在 SCLn 线为 TMOH 位或者 TMOL 位选择的状态时, 将内部基准时钟 (IIC ϕ) 作为计数源进行递增计数。

有关超时检测功能的详细内容, 请参照“22.11.1 超时检测功能”。

TMOL 位 (超时 L 电平计数控制位)

在超时检测功能有效时 (ICFER.TMOE 位 =1)，此位选择在 SCLn 线为 Low 电平期间允许或者禁止超时检测功能的内部计数器的递增计数。

TMOH 位 (超时 H 电平计数控制位)

在超时检测功能有效时 (ICFER.TMOE 位 =1)，此位选择在 SCLn 线为 High 电平期间允许或者禁止超时检测功能的内部计数器的递增计数。

SDDL[2:0] 位 (SDA 输出延迟计数器)

能通过 SDDL[2:0] 位的设定值使 SDA 延迟输出。SDA 输出延迟计数器通过由 DLCS 位选择的时钟源进行计数。另外，此功能的设定还适用于包括应答位发送在内的全部 SDA 输出。

有关此功能的详细内容，请参照“22.5 SDA 输出延迟功能”。

注 1. 必须注意：SDA 输出延迟的设定必须符合 I²C 总线规格 (数据有效时间 / 应答有效时间 (注 2) 内) 或者 SMBus 规格 (数据保持时间：至少为 300ns，SCL 时钟的 Low 电平宽度的数据准备时间：250ns 的范围内)，否则就有可能引起通信设备的通信故障，或者根据总线状态视如开始条件或者停止条件。

注 2. 数据有效时间 / 应答有效时间

3,450ns (~ 100kbps : 标准模式 [Sm])

900ns (~ 400kbp : 快速模式 [fm])

450ns (~ 1Mbps : 快速模式+[fm+])

DLCS 位 (SDA 输出延迟时钟源选择位)

此位选择内部基准时钟 (IIC ϕ) 或者内部基准时钟的 2 分频 (IIC ϕ /2) 时钟为 SDA 输出延迟时间的时钟源。

22.2.5 I²C 总线模式寄存器 3 (ICMR3)

地址 RIIC0.ICMR3 0008 8304h、RIIC1.ICMR3 0008 8324h

| | | | | | | | |
|-------|------|-------|-------|-------|-------|---------|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| SMBS | WAIT | RDRFS | ACKWP | ACKBT | ACKBR | NF[1:0] | |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|------------------------------|--|--------------|
| b1-b0 | NF[1:0] | 噪声滤波器的段数选择位 | b1 b0 00: 消除不超过 1 个 PCLK 的噪声 (滤波器为 1 段) 01: 消除不超过 2 个 PCLK 的噪声 (滤波器为 2 段) 10: 消除不超过 3 个 PCLK 的噪声 (滤波器为 3 段) 11: 消除不超过 4 个 PCLK 的噪声 (滤波器为 4 段) | R/W |
| b2 | ACKBR | 接收应答位 | 0: 应答位接收 “0” (ACK 接收) 1: 应答位接收 “1” (NACK 接收) | R |
| b3 | ACKBT | 发送应答位 | 0: 应答位发送 “0” (ACK 发送) 1: 应答位发送 “1” (NACK 发送) | R/W (注 1) |
| b4 | ACKWP | ACKBT 写保护位 | 0: 禁止写 ACKBT 位 1: 允许写 ACKBT 位 | W (注 1) |
| b5 | RDRFS | RDRF 标志的置位时序选择位 | 0: 在 SCL 时钟的第 9 个时钟上升时, 此位变为 “1”。 (在第 8 个时钟的下降沿, SCLn 线不保持 Low 电平) 1: 在 SCL 时钟的第 8 个时钟上升时, 此位变为 “1”。 (在第 8 个时钟的下降沿, SCLn 线保持 Low 电平) 通过写 ACKBT 位来解除保持的 Low 电平。 | R/W (注 2) |
| b6 | WAIT | WAIT 位 | 0: 无 WAIT (在第 9 个时钟和下次传送的第 1 个时钟之间不保持 Low 电平) 1: 有 WAIT (在第 9 个时钟和下次传送的第 1 个时钟之间保持 Low 电平) • 通过读 ICDRR 寄存器来解除保持的 Low 电平。 | R/W (注 2) |
| b7 | SMBS | SMBus/I ² C 总线选择位 | 0: 选择 I ² C 总线 1: 选择 SMBus | R/W |

注 1. 即使在写 ACKBT 位的同时将 ACKWP 位置 “1”, 也不能写 ACKBT 位。

注 2. WAIT 位和 RDRFS 位只在接收模式中有效, 在发送模式中无效。

ICMR3 寄存器控制应答发送 / 接收功能、RIIC 接收时的 RDRF 标志以及 WAIT 运行等。

NF[1:0] 位 (噪声滤波器的段数选择位)

这些位选择数字噪声滤波器的段数。

注 1. 必须注意: 用噪声滤波器消除的噪声宽度必须小于 SCLn 线的高/低电平宽度。

如果大于等于 (SCL 时钟的宽度: High 电平宽度和 Low 电平宽度中较短的一方) - {1.5 个内部基准时钟同步 (IIC ϕ) + 模拟噪声滤波器: 120ns (参考值)} 的值, RIIC 的噪声滤波器功能就可能将 SCL 时钟视为噪声而无法正常运行。

ACKBR 位 (接收应答位)

在发送模式中, 此位保存从接收设备收到的应答位的内容。

[为“1”的条件]

- 在ICCR2.TRS位为“1”的状态下应答位接收到“1”时

[为“0”的条件]

- 在ICCR2.TRS位为“1”的状态下应答位接收到“0”时
- 在ICCR1.ICE位为“0”的状态下给ICCR1.IICRST位写“1”(RIIC复位)时

ACKBT 位 (发送应答位)

在接收模式中, 此位设定应答时要发送的位。

[为“1”的条件]

- 在ACKWP位为“1”的状态下写“1”时

[为“0”的条件]

- 在ACKWP位为“1”的状态下写“0”时
- 检测到停止条件的发行 (在ICCR2.SP位为“1”的状态下检测到停止条件)时
- 在ICCR1.ICE位为“0”的状态下给ICCR1.IICRST位写“1”(RIIC复位)时

注 1. 必须在 ACKWP 位为“1”的状态下写 ACKBT 位。如果在 ACKWP 位为“0”的状态下写 ACKBT 位, 写操作就无效。

ACKWP 位 (ACKBT 写保护位)

此位控制 ACKBT 位的写操作。

RDRFS 位 (RDRF 标志的置位时序选择位)

此位选择接收模式中的 RDRF 标志的置位时序以及在 SCL 时钟的第 8 个时钟的下降沿是否保持 SCLn 线的 Low 电平。

当 RDRFS 位为“0”时, 在第 8 个时钟的下降沿 SCLn 线不保持 Low 电平, 在第 9 个时钟的上升沿将 RDRF 标志置“1”。

当 RDRFS 位为“1”时, 在第 8 个时钟的上升沿将 RDRF 标志置“1”, 在第 8 个时钟的下降沿 SCLn 线保持 Low 电平。通过写 ACKBT 位来解除此 SCLn 线保持的 Low 电平。

因为在进行此设定时, 在接收数据后到发送应答位前 SCLn 线自动保持 Low 电平, 所以能根据接收数据的内容发送 ACK (ACKBT 位为“0”)或者 NACK (ACKBT 位为“1”)。

WAIT 位 (WAIT 位)

在接收模式中, 如果每接收 1 字节数据, 就在读完接收数据缓冲器 (ICDRR 寄存器) 前, WAIT 位控制在 SCL 时钟的第 9 个时钟和下次传送的第 1 个时钟之间是否保持 Low 电平。

当 WAIT 位为“0”时, 在 SCL 时钟的第 9 个时钟和下次传送的第 1 个时钟之间不保持 Low 电平, 继续接收数据。在 RDRFS 位和 WAIT 位都为“0”时, 也能通过双缓冲器进行连续的接收。

当 WAIT 位为“1”时, 如果每接收 1 字节数据, 就从第 9 个时钟下降后到读 ICDRR 寄存器的值前, SCLn 线保持 Low 电平。因此能按字节接收数据。

注 1. 必须先读 ICDRR, 然后将 WAIT 位置“0”。

SMBS 位 (SMBus/I²C 总线选择位)

如果将 SMBS 位置“1”, 就选择 SMBus 并且 ICSEH.HOAE 位有效。

22.2.6 I²C 总线功能允许寄存器 (ICFER)

地址 RIIC0.ICFER 0008 8305h、RIIC1.ICFER 0008 8325h

| | | | | | | | | |
|-------|------|-----|-------|------|------|------|------|---|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | |
| FMPE | SCLE | NFE | NACKE | SALE | NALE | MALE | TMOE | |
| 复位后的值 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|------------------|---|-----|
| b0 | TMOE | 超时检测功能允许位 | 0: 超时检测功能无效 1: 超时检测功能有效 | R/W |
| b1 | MALE | 主控仲裁失败检测允许位 | 0: 禁止主控仲裁失败的检测 (将仲裁失败检测功能设定为无效, 在发生仲裁失败时不自动清除 ICCR2.MST 位和 ICCR2.TRS 位) 1: 允许主控仲裁失败的检测 (将仲裁失败检测功能设定为有效, 在发生仲裁失败时自动清除 ICCR2.MST 位和 ICCR2.TRS 位) | R/W |
| b2 | NALE | NACK 发送仲裁失败检测允许位 | 0: 禁止 NACK 发送仲裁失败的检测 1: 允许 NACK 发送仲裁失败的检测 | R/W |
| b3 | SALE | 从属仲裁失败检测允许位 | 0: 禁止从属仲裁失败的检测 1: 允许从属仲裁失败的检测 | R/W |
| b4 | NACKE | NACK 接收传送中止允许位 | 0: 在接收 NACK 时, 不中止传送 (禁止中止传送)。 1: 在接收 NACK 时, 中止传送 (允许中止传送)。 | R/W |
| b5 | NFE | 数字噪声滤波器电路允许位 | 0: 不使用数字噪声滤波器电路 1: 使用数字噪声滤波器电路 | R/W |
| b6 | SCLE | SCL 同步电路允许位 | 0: SCL 同步电路无效 1: SCL 同步电路有效 | R/W |
| b7 | FMPE | 快速模式 + 允许位 | 0: 不将 fm+ 的斜率控制电路用于 SCLn 引脚 / SDAn 引脚 1: 将 fm+ 的斜率控制电路用于 SCLn 引脚 / SDAn 引脚 | R/W |

ICFER 寄存器控制是否使用超时检测功能、仲裁失败、接收 NACK 时的接收、SCL 同步电路和数字噪声滤波器电路。

TMOE 位 (超时检测功能允许位)

此位选择超时检测功能的有效或者无效。

有关超时检测功能的详细内容, 请参照“22.11.1 超时检测功能”。

MALE 位 (主控仲裁失败检测允许位)

此位决定主控模式中仲裁失败检测功能的有效或者无效。一般必须将此位置“1”。

NALE 位 (NACK 发送仲裁失败检测允许位)

在接收模式中, 此位选择在发送 NACK 时检测到 ACK 的情况下 (总线上有相同地址的从属器件, 或者 2 个以上 (包括 2 个) 的主控器件同时选择了相同的从属器件并且各自接收的字节数不同等情况) 是否产生仲裁失败。

SALE 位 (从属仲裁失败检测允许位)

在从属发送模式中, 此位选择在总线上检测到的值和正在发送的值不同的情况下 (总线上有相同地址的从属器件, 或者因噪声的影响而产生和发送数据不同的数据等情况) 是否产生仲裁失败。

NACKE 位 (NACK 接收传送中止允许位)

在发送模式中, 此位选择在从属器件接收到 NACK 时是继续传送还是中止传送。一般必须将此位置 “1”。

如果在 NACKE 位为 “1” 时接收到 NACK, 就中止下一个传送。

当 NACKE 位为 “0” 时, 与接收应答的内容无关, 继续进行下一个传送。

有关 NACK 接收传送中止功能的详细内容, 请参照 “22.8.2 NACK 接收传送中止功能”。

NFE 位 (数字噪声滤波器电路允许位)

此位选择是否使用数字噪声滤波器电路。

SCLE 位 (SCL 同步电路允许位)

对于 SCL 输入时钟, 此位选择是否与 SCL 时钟同步。一般必须将此位置 “1”。

如果将 SCLE 位置 “0” (SCL 同步电路无效), 就不与时钟同步。在此设定下, 与 SCLn 线的状态无关, RIIC 输出 ICBRH 寄存器和 ICBRL 寄存器所设传送率的 SCL 时钟。因此必须注意: 在 I²C 总线的总线负载远远大于规格值时或者多主控模式中的 SCL 时钟输出发生重叠时, 有可能变为规格外的短时钟。在 SCL 同步电路无效的情况下, 也影响开始条件、重新开始条件、停止条件的发行以及 SCL 时钟追加输出的连续输出。

只在确认是否输出了所设传送率的时钟时才能将 SCLE 位置 “0”。

FMPE 位 (快速模式 + 允许位)

此位选择斜率控制电路是否使用快速模式 +[fm+] 的斜率控制电路。

如果将 FMPE 位置 “1”, 就选择符合 I²C 总线的快速模式 +[fm+] 的斜率控制规格 (tof) 的斜率控制电路; 如果置 “0”, 就选择符合 I²C 总线的标准模式 [Sm] 和快速模式 [fm] 的斜率控制规格 (tof) 的斜率控制电路。

如果使用 I²C 总线规格的 ~ 1Mbps (快速模式 +[fm+]) 的通信速度, 就必须将此位置 “1”。如果使用其他通信速度 (~ 100kbps[Sm] 或者 ~ 400kbps[fm]) 或者 SMBus (10kbps ~ 100kbps), 就必须将此位置 “0”。

22.2.7 I²C 总线状态允许寄存器 (ICSER)

地址 RIIC0.ICSER 0008 8306h、RIIC1.ICSER 0008 8326h

| | | | | | | | | |
|-------|------|----|------|----|------|-------|-------|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | HOAE | — | DIDE | — | GCAE | SAR2E | SAR1E | SAR0E |
| 复位后的值 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|---------------|--|-----|
| b0 | SAR0E | 从属地址寄存器 0 允许位 | 0: SARL0 和 SARU0 的设定值无效 1: SARL0 和 SARU0 的设定值有效 | R/W |
| b1 | SAR1E | 从属地址寄存器 1 允许位 | 0: SARL1 和 SARU1 的设定值无效 1: SARL1 和 SARU1 的设定值有效 | R/W |
| b2 | SAR2E | 从属地址寄存器 2 允许位 | 0: SARL2 和 SARU2 的设定值无效 1: SARL2 和 SARU2 的设定值有效 | R/W |
| b3 | GCAE | 全呼地址允许位 | 0: 全呼地址的检测无效 1: 全呼地址的检测有效 | R/W |
| b4 | — | 保留位 | 读写值都为“0”。 | R/W |
| b5 | DIDE | 设备 ID 地址检测允许位 | 0: 设备 ID 地址的检测无效 1: 设备 ID 地址的检测有效 | R/W |
| b6 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | HOAE | 主机地址允许位 | 0: 主机地址的检测无效 1: 主机地址的检测有效 | R/W |

ICSER 寄存器选择从属地址比较、全呼地址检测、设备 ID 地址检测、主机地址检测的有效或者无效。

SARyE 位 (从属地址寄存器 y 允许位) (y=0 ~ 2)

此位选择 SARLy 寄存器和 SARUy 寄存器设定的从属地址是否有效。

如果将 SARyE 位置“1”，SARLy 寄存器和 SARUy 寄存器的设定值就有效，和接收的从属地址进行比较。

如果将 SARyE 位置“0”，SARLy 寄存器和 SARUy 寄存器的设定值就无效，即使与接收的从属地址匹配，也忽视此设定值。

GCAE (全呼地址允许位)

在接收到全呼地址 (0000 000b+0[W]; All “0”) 时，此位选择是否忽视此地址。

在 GCAE 位为“1”时，如果接收的从属地址与全呼地址匹配，RIIC 就将此从属地址视为全呼地址进行接收，与 SARLy 寄存器和 SARUy 寄存器 (y=0 ~ 2) 设定的从属地址无关。

在 GCAE 位为“0”时，即使接收的从属地址与全呼地址匹配，也忽视此地址。

DIDE 位 (设备 ID 地址检测允许位)

在检测到开始条件或者重新开始条件后的第 1 帧接收到设备 ID 地址 (1111 100b) 时，此位选择是否将此地址视为设备 ID 地址。

在 DIDE 位为“1”时，如果接收的第 1 帧与设备 ID 地址匹配，RIIC 就认为已接收到设备 ID 地址，当后续的 R/W# 位为“0” [W] 时，将第 2 帧以后的帧视为从属地址，继续接收。

在 DIDE 位为“0”时，即使接收的第 1 帧与设备 ID 地址匹配，也忽视此帧，而将第 1 帧视为一般的从属地址。

有关设备 ID 地址检测的详细内容，请参照“22.7.3 设备 ID 地址检测功能”。

HOAE 位 (主机地址允许位)

在 ICMR3.SMBS 位为“1”时接收到主机地址 (0001 000b) 的情况下，此位选择是否忽视此地址。

在 ICMR3.SMBS 位为“1”并且 HOAE 位为“1”时，如果接收的从属地址与主机地址匹配，RIIC 就将此从属地址视为主机地址进行接收，与 SARLy 寄存器和 SARUy 寄存器 (y=0 ~ 2) 设定的从属地址无关。

在 ICMR3.SMBS 位或者 HOAE 位为“0”时，即使接收的从属地址与主机地址匹配，也忽视此地址。

22.2.8 I²C 总线中断允许寄存器 (ICIER)

地址 RIIC0.ICIER 0008 8307h、RIIC1.ICIER 0008 8327h

| | | | | | | | | |
|-------|-----|------|-----|-------|------|------|------|-------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | TIE | TEIE | RIE | NAKIE | SPIE | STIE | ALIE | TMOIE |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|--------------|--|-----|
| b0 | TMOIE | 超时中断允许位 | 0: 禁止超时中断 (TMOI) 1: 允许超时中断 (TMOI) | R/W |
| b1 | ALIE | 仲裁失败中断允许位 | 0: 禁止仲裁失败中断 (ALI) 1: 允许仲裁失败中断 (ALI) | R/W |
| b2 | STIE | 开始条件检测中断允许位 | 0: 禁止开始条件检测中断 (STI) 1: 允许开始条件检测中断 (STI) | R/W |
| b3 | SPIE | 停止条件检测中断允许位 | 0: 禁止停止条件检测中断 (SPI) 1: 允许停止条件检测中断 (SPI) | R/W |
| b4 | NAKIE | NACK 接收中断允许位 | 0: 禁止 NACK 接收中断 (NAKI) 1: 允许 NACK 接收中断 (NAKI) | R/W |
| b5 | RIE | 接收数据满中断允许位 | 0: 禁止接收数据满中断 (ICRXI) 1: 允许接收数据满中断 (ICRXI) | R/W |
| b6 | TEIE | 发送结束中断允许位 | 0: 禁止发送结束中断 (ICTEI) 1: 允许发送结束中断 (ICTEI) | R/W |
| b7 | TIE | 发送数据空中断允许位 | 0: 禁止发送数据空中断 (ICTXI) 1: 允许发送数据空中断 (ICTXI) | R/W |

ICIER 寄存器是选择是否使用各种中断源的寄存器。

TMOIE 位 (超时中断允许位)

当 ICSR2.TMOF 标志为“1”时，此位选择允许或者禁止超时中断 (TMOI)。能通过将 TMOF 标志或者 TMOIE 位置“0”来解除 TMOI 中断。

ALIE 位 (仲裁失败中断允许位)

当 ICSR2.AL 标志为“1”时，此位选择允许或者禁止仲裁失败中断 (ALI)。能通过将 AL 标志或者 ALIE 位置“0”来解除 ALI 中断。

STIE 位 (开始条件检测中断允许位)

当 ICSR2.START 标志为“1”时，此位选择允许或者禁止开始条件的检测中断 (STI)。能通过将 START 标志或者 STIE 位置“0”来解除 STI 中断。

SPIE 位 (停止条件检测中断允许位)

当 ICSR2.STOP 标志为“1”时，此位选择允许或者禁止停止条件的检测中断 (SPI)。能通过将 STOP 标志或者 SPIE 位置“0”来解除 SPI 中断。

NAKIE 位 (NACK 接收中断允许位)

当 ICSR2.NACKF 标志为“1”时，此位选择允许或者禁止 NACK 接收中断 (NAKI)。通过将 NACKF 标志或者 NAKIE 位置“0”来解除 NAKI 中断。

RIE 位 (接收数据满中断允许位)

当 ICSR2.RDRF 标志为“1”时，此位选择允许或者禁止接收数据满中断 (ICRXI)。

TEIE 位 (发送结束中断允许位)

当 ICSR2.TEND 标志为“1”时，此位选择允许或者禁止发送结束中断 (ICTEI)。能通过将 TEND 标志或者 TEIE 位置“0”来解除 ICTEI 中断。

TIE 位 (发送数据空中断允许位)

当 ICSR2.TDRE 标志为“1”时，此位选择允许或者禁止发送数据空中断 (ICTXI)。

22.2.9 I²C 总线状态寄存器 1 (ICSR1)

地址 RIIC0.ICSR1 0008 8308h、RIIC1.ICSR1 0008 8328h

| | | | | | | | | |
|-------|-----|----|-----|----|-----|------|------|------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | HOA | — | DID | — | GCA | AAS2 | AAS1 | AAS0 |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|------|---------------|--|---------------|
| b0 | AAS0 | 从属地址 0 的检测标志 | 0: 未检测到从属地址 0 1: 检测到从属地址 0 • 在 SARU0.FS 位为 “0” (选择 7 位地址格式) 的情况下, 接收的从属地址与 SARL0.SVA[7:1] 匹配时 • 在 SARU0.FS 位为 “1” (选择 10 位地址格式) 的情况下, 接收的从属地址与 1111 0b+SARU0.SVA[9:8] 匹配, 并且后续的地址与 SARL0 寄存器匹配时 (在 SARL0 匹配判断帧的 SCL 时钟的第 9 个时钟上升时, 此标志变为 “1”。) | R(W) (注 1) |
| b1 | AAS1 | 从属地址 1 的检测标志 | 0: 未检测到从属地址 1 1: 检测到从属地址 1 • 在 SARU1.FS 位为 “0” (选择 7 位地址格式) 的情况下, 接收的从属地址与 SARL1.SVA[7:1] 匹配时 • 在 SARU1.FS 位为 “1” (选择 10 位地址格式) 的情况下, 接收的从属地址与 1111 0b+SARU1.SVA[9:8] 匹配, 并且后续的地址与 SARL1 寄存器匹配时 (在 SARL1 匹配判断帧的 SCL 时钟的第 9 个时钟上升时, 此标志变为 “1”。) | R(W) (注 1) |
| b2 | AAS2 | 从属地址 2 的检测标志 | 0: 未检测到从属地址 2 1: 检测到从属地址 2 • 在 SARU2.FS 位为 “0” (选择 7 位地址格式) 的情况下, 接收的从属地址与 SARL2.SVA[7:1] 匹配时 • 在 SARU2.FS 位为 “1” (选择 10 位地址格式) 的情况下, 接收的从属地址与 1111 0b+SARU2.SVA[9:8] 匹配, 并且后续的地址与 SARL2 寄存器匹配时 (在 SARL2 匹配判断帧的 SCL 时钟的第 9 个时钟上升时, 此标志变为 “1”。) | R(W) (注 1) |
| b3 | GCA | 全呼地址的检测标志 | 0: 未检测到全呼地址 1: 检测到全呼地址 • 接收的从属地址与全呼地址 (All “0”) 匹配时 | R(W) (注 1) |
| b4 | — | 保留位 | 读写值都为 “0”。 | R/W |
| b5 | DID | 设备 ID 地址的检测标志 | 0: 未检测到设备 ID 地址 1: 检测到设备 ID 地址 • 紧接在开始条件后的第 1 帧与设备 ID 地址 (1111 100b) +0[W] 匹配时 | R(W) (注 1) |
| b6 | — | 保留位 | 读写值都为 “0”。 | R/W |
| b7 | HOA | 主机地址的检测标志 | 0: 未检测到主机地址 1: 检测到主机地址 • 接收的从属地址与主机地址 (0001 000b) 匹配时 | R(W) (注 1) |

注 1. 只能写 “0”。

ICSR1 寄存器是确认各种地址检测状态的寄存器。

AASy 标志 (从属地址 y 的检测标志) (y=0 ~ 2)

[为 “1” 的条件]

【选择 7 位地址格式时: SARUy.FS 位 =0】

- 在 ICSEr.SARyE 位为 “1” (从属地址 y 的检测有效) 的状态下接收的从属地址与 SARLy.SVA[7:1] 匹配时, 该帧的 SCL 时钟的第 9 个时钟上升

【选择 10 位地址格式时: SARUy.FS 位 =1】

- 在 ICSEr.SARyE 位为 “1” (从属地址 y 的检测有效) 的状态下接收的从属地址与 1111 0b+SARUy.SVA[9:8] 匹配, 并且后续的地址与 SARLy 寄存器匹配时, 该帧的 SCL 时钟的第 9 个时钟上升

[为 “0” 的条件]

- 读 “1” 后写 “0” 时
- 检测到停止条件时
- 给 ICCR1.IICRST 位写 “1”, 并且进行 RIIC 复位或者内部复位时

【选择 7 位地址格式时: SARUy.FS 位 =0】

- 在 ICSEr.SARyE 位为 “1” (从属地址 y 的检测有效) 的状态下接收的从属地址与 SARLy.SVA[7:1] 不匹配时, 该帧的 SCL 时钟的第 9 个时钟上升

【选择 10 位地址格式时: SARUy.FS 位 =1】

- 在 ICSEr.SARyE 位为 “1” (从属地址 y 的检测有效) 的状态下接收的从属地址与 1111 0b+SARUy.SVA[9:8] 不匹配时, 该帧的 SCL 时钟的第 9 个时钟上升
- 在 ICSEr.SARyE 位为 “1” (从属地址 y 的检测有效) 的状态下接收的从属地址与 1111 0b+SARUy.SVA[9:8] 匹配, 而后续的地址与 SARLy 寄存器不匹配时, 该帧的 SCL 时钟的第 9 个时钟上升

GCA 标志 (全呼地址的检测标志)

[为 “1” 的条件]

- 在 ICSEr.GCAE 位为 “1” (全呼地址的检测有效) 的状态下接收的从属地址与全呼地址 (0000 000b +0[W]) 匹配时, 该帧的 SCL 时钟的第 9 个时钟上升

[为 “0” 的条件]

- 读 “1” 后写 “0” 时
- 检测到停止条件时
- 在 ICSEr.GCAE 位为 “1” (全呼地址的检测有效) 的状态下接收的从属地址与全呼地址 (0000 000b +0[W]) 不匹配时, 该帧的 SCL 时钟的第 9 个时钟上升
- 给 ICCR1.IICRST 位写 “1”, 并且进行 RIIC 复位或者内部复位时

DID 标志 (设备 ID 地址的检测标志)

[为“1”的条件]

- 在 IC SER.DIDE 位为“1” (设备 ID 地址的检测有效) 的状态下检测到开始条件或者重新开始条件后的第 1 帧与设备 ID 地址 (1111 100b) +0[W] 匹配时, 该帧的 SCL 时钟的第 9 个时钟上升

[为“0”的条件]

- 读“1”后写“0”时
- 检测到停止条件时
- 在 IC SER.DIDE 位为“1” (设备 ID 地址的检测有效) 的状态下检测到开始条件或者重新开始条件检测后的第 1 帧与设备 ID 地址 (1111 100b) 不匹配时, 该帧的 SCL 时钟的第 9 个时钟上升
- 在 IC SER.DIDE 位为“1” (设备 ID 地址的检测有效) 的状态下检测到开始条件或者重新开始条件检测后的第 1 帧与设备的 ID 地址 (1111 100b) +0[W] 匹配, 而后续的第 2 帧与从属地址 0~2 全部不匹配时, 该帧的 SCL 时钟的第 9 个时钟上升
- 给 IC CR1.IICRST 位写“1”, 并且进行 RIIC 复位或者内部复位时

HOA 标志 (主机地址的检测标志)

[为“1”的条件]

- 在 IC SER.HOAE 位为“1” (主机地址的检测有效) 的状态下接收的从属地址与主机地址 (0001 000b) 匹配时, 该帧的 SCL 时钟的第 9 个时钟上升

[为“0”的条件]

- 读“1”后写“0”时
- 检测到停止条件时
- 给 IC MR3.SMBS 位或者 IC SER.HOAE 位写“0”时
- 在 IC SER.HOAE 位为“1” (主机地址的检测有效) 的状态下接收的从属地址与主机地址 (0001 000b) 不匹配时, 该帧的 SCL 时钟的第 9 个时钟上升
- 给 IC CR1.IICRST 位写“1”, 并且进行 RIIC 复位或者内部复位时

22.2.10 I²C 总线状态寄存器 2 (ICSR2)

地址 RIIC0.ICSR2 0008 8309h、RIIC1.ICSR2 0008 8329h

| | | | | | | | | |
|-------|------|------|------|-------|------|-------|----|------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | TDRE | TEND | RDRF | NACKF | STOP | START | AL | TMOF |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|-------|-----------|--|--------------|
| b0 | TMOF | 超时检测标志 | 0: 未检测到超时 1: 检测到超时 | R(W) (注1) |
| b1 | AL | 仲裁失败标志 | 0: 未发生仲裁失败 1: 发生仲裁失败 | R(W) (注1) |
| b2 | START | 开始条件检测标志 | 0: 未检测到开始条件 1: 检测到开始条件 | R(W) (注1) |
| b3 | STOP | 停止条件检测标志 | 0: 未检测到停止条件 1: 检测到停止条件 | R(W) (注1) |
| b4 | NACKF | NACK 检测标志 | 0: 未检测到 NACK 1: 检测到 NACK | R(W) (注1) |
| b5 | RDRF | 接收数据满标志 | 0: ICDRR 寄存器无接收数据 1: ICDRR 寄存器有接收数据 | R(W) (注1) |
| b6 | TEND | 发送结束标志 | 0: 正在发送数据 1: 数据发送结束 | R(W) (注1) |
| b7 | TDRE | 发送数据空标志 | 0: ICDRT 寄存器有发送数据 1: ICDRT 寄存器无发送数据 | R |

注 1. 只能写“0”。

ICSR2 寄存器是确认各种中断请求标志及其状态的寄存器。

TMOF 标志 (超时检测标志)

如果 SCLn 线的状态在一定期间内不发生变化, 就视为超时, 此标志变为“1”。

[为“1”的条件]

- 当 ICFER.TMOE 位为“1” (超时检测功能有效), 并且在指定为主控模式或者从属模式的状态下由 ICMR2.TMOH 位、ICMR2.TMOL 位、ICMR2.TMOS 位所选的条件的期间内, SCLn 线的状态没有变化时

[为“0”的条件]

- 读“1”后写“0”时
- 给 ICCR1.IICRST 位写“1”, 并且进行 RIIC 复位或者内部复位时

AL 标志 (仲裁失败标志)

在发行开始条件或者发送地址和数据时, 此标志表示因总线竞争等而失去了总线占有权 (仲裁失败)。

RIIC 在发送过程中监视 SDA_n 线的电平, 如果输出数据和 SDA_n 线的电平不同, 就将 AL 标志置“1”, 表示总线被其他设备占有。

另外, 通过设定, RIIC 也能在主动模式中检测发送 NACK 时的仲裁失败, 在从属模式中检测发送数据时的仲裁失败。

[为“1”的条件]

【主控仲裁失败的检测有效时：ICFER.MALE 位 =1】

- 在 主控发送模式的数据发送（包括从属地址的发送）过程中，在 ACK 期间以外的 SCL 时钟的上升沿，RIIC 自身发送的 SDA 信号和 SDA_n 线上的信号状态不同（内部 SDA 输出为 High 电平输出（=SDA_n 引脚为高阻抗状态）而检测到 SDA_n 线为 Low 电平）时
- 在 ICCR2.ST 位为“1”（请求发行开始条件）的状态下检测到开始条件时，RIIC 自身发送的 SDA 信号和 SDA_n 线上的信号状态不同时
- 在 ICCR2.BBSY 标志为“1”的状态下将 ICCR2.ST 位置“1”（请求发行开始条件）时

【NACK 仲裁失败的检测有效时：ICFER.NALE 位 =1】

- 在接收模式的 NACK 发送过程中，在 ACK 期间的 SCL 时钟的上升沿，RIIC 自身发送的 SDA 信号和 SDA_n 线上的信号状态不同时

【从属仲裁失败的检测有效时：ICFER.SALE 位 =1】

- 在从属发送模式的数据发送过程中，在 ACK 期间以外的 SCL 时钟上升沿，RIIC 自身发送的 SDA 信号和 SDA_n 线上的信号状态不同时

[为“0”的条件]

- 读“1”后写“0”时
- 给 ICCR1.IICRST 位写“1”，并且进行 RIIC 复位或者内部复位时

表 22.5 仲裁失败发生源和各仲裁失败允许功能的关系

| ICFER | | | ICSR2 | 错误内容 | 仲裁失败发生源 |
|-------|------|------|-------|-------------|--|
| MALE | NALE | SALE | AL | | |
| 1 | * | * | 1 | 开始条件发行错误 | 在 ICCR2.ST 位为“1”的状态下检测到开始条件时，RIIC 自身发送的 SDA 信号和 SDA _n 线上的信号状态不同。 在 ICCR2.BBSY 位为“1”的状态下将 ICCR2.ST 位置“1”。 |
| | | | 1 | 发送数据不同 | 在 主控发送模式中，发送数据（包括从属地址的发送）与总线状态不同。 |
| * | 1 | * | 1 | 发送的 NACK 不同 | 在 主控接收模式或者从属接收模式中，在发送 NACK 时检测到 ACK。 |
| * | * | 1 | 1 | 发送数据不同 | 在 从属发送模式中，发送数据与总线状态不同。 |

*: Don't care

START 标志（开始条件检测标志）

[为“1”的条件]

- 检测到开始条件（包括重新开始条件）时

[为“0”的条件]

- 读“1”后写“0”时
- 检测到停止条件时
- 给 ICCR1.IICRST 位写“1”，并且进行 RIIC 复位或者内部复位时

STOP 标志（停止条件检测标志）

[为“1”的条件]

- 检测到停止条件时

[为“0”的条件]

- 读“1”后写“0”时
- 给 ICCR1.IICRST 位写“1”，并且进行 RIIC 复位或者内部复位时

NACKF 标志 (NACK 检测标志)

[为“1”的条件]

- 在ICFER.NACK位为“1”（允许中止传送）的状态下，发送模式中没有来自接收设备的应答（接收到NACK）时

[为“0”的条件]

- 读“1”后写“0”时
- 给ICCR1.IICRST位写“1”，并且进行RIIC复位或者内部复位时

注1. 如果 NACKF 标志变为“1”，RIIC 就中止通信。即使在 NACKF 标志为“1”的状态下写 ICDRT 寄存器（在发送模式中）或者读 ICDRR 寄存器（在接收模式中），也不进行发送和接收。要重新开始通信时，必须将 NACKF 标志置“0”。

RDRF 标志 (接收数据满标志)

[为“1”的条件]

- 在将接收数据从ICDRS寄存器传送到ICDRR寄存器时，通过设定ICMR3.RDRFS位，在SCL时钟的第8个时钟或者第9个时钟的上升沿RDRF标志变为“1”
- 在检测到开始条件（包括重新开始条件）后，接收的从属地址匹配并且ICCR2.TRS位为“0”时

[为“0”的条件]

- 读“1”后写“0”时
- 读ICDRR寄存器时
- 给ICCR1.IICRST位写“1”，并且进行RIIC复位或者内部复位时

TEND 标志 (发送结束标志)

[为“1”的条件]

- 在TDRE标志为“1”的状态下，SCL时钟的第9个时钟上升

[为“0”的条件]

- 读“1”后写“0”时
- 给ICDRT寄存器写数据时
- 检测到停止条件时
- 给ICCR1.IICRST位写“1”，并且进行RIIC复位或者内部复位时

TDRE 标志 (发送数据空标志)

[为“1”的条件]

- 将数据从ICDRT寄存器传送到ICDRS寄存器并且ICDRT寄存器为空时
- ICCR2.TRS位变为“1”时
 - a. 在检测到开始条件（包括重新开始条件）后，ICCR2.MST位为“1”时
 - b. 从接收模式变为发送模式时
 - c. 在ICMR1.MTWP位为“1”的状态下写“1”时
- 接收的从属地址匹配并且TRS位为“1”时

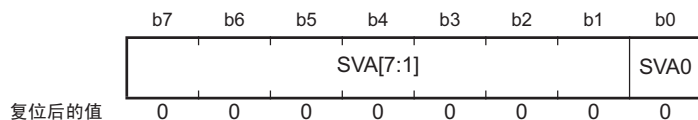
[为“0”的条件]

- 给ICDRT寄存器写数据时
- ICCR2.TRS位变为“0”时
 - a. 检测到停止条件时
 - b. 从发送模式变为接收模式时
 - c. 在ICMR1.MTWP位为“1”的状态下写“0”时
- 给ICCR1.IICRST位写“1”，并且进行RIIC复位或者内部复位时

注1. 如果在ICFER.NACK位为“1”的状态下NACKF标志变为“1”，RIIC就中止通信。此时，如果TDRE标志为“0”（已写下一个发送数据的状态），就在第9个时钟的上升沿将数据传送到ICDRS寄存器并且ICDRT寄存器变为空状态，但是TDRE标志不变为“1”。

22.2.11 从属地址寄存器 Lm (SARLy) (m=0 ~ 2)

地址 RIIC0.SARL0 0008 830Ah、RIIC1.SARL0 0008 832Ah
 RIIC0.SARL1 0008 830Ch、RIIC1.SARL1 0008 832Ch
 RIIC0.SARL2 0008 830Eh、RIIC1.SARL2 0008 832Eh



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|------------------|--|-----|
| b0 | SVA0 | 10 位地址的最低有效位 | 必须设定从属地址。 • 当 SARUy.FS 位为 “0” (选择 7 位地址格式) 时, SVA0 位无效。 • 当 SARUy.FS 位为 “1” (选择 10 位地址格式) 时, SVA0 位有效。SVA0 位和 SVA[7:1] 位合并为 10 位从属地址的低 8 位地址。 | R/W |
| b7-b1 | SVA[7:1] | 7 位地址 /10 位地址的低位 | 必须设定从属地址。 • 当 SARUy.FS 位为 “0” (选择 7 位地址格式) 时, SVA[7:1] 位为 7 位从属地址。 • 当 SARUy.FS 位 “1” (选择 10 位地址格式) 时, SVA[7:1] 位和 SVA0 位合并为 10 位从属地址的低 8 位地址。 | R/W |

SARLy 寄存器是设定从属地址 y (7 位地址或者 10 位地址的低 8 位) 的寄存器。

SVA0 位 (10 位地址的最低有效位)

在选择 10 位地址格式时 (SARUy.FS 位 =1), 作为 10 位地址的最低有效位, 此位和 SVA[7:1] 位合并, 用作 10 位地址的低 8 位。

当 IC SER.SARyE 位为 “1” (SARLy 寄存器和 SARUy 寄存器有效) 并且 SARUy.FS 位为 “1” 时, 设定值有效; 当 SARUy.FS 位或者 SARyE 位为 “0” 时, 忽视设定值。

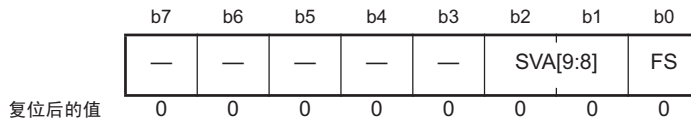
SVA[7:1] 位 (7 位地址 /10 位地址的低位)

在选择 7 位地址格式时 (SARUy.FS 位 =0), 用作 7 位地址; 在选择 10 位地址格式时 (SARUy.FS 位 =1), 此位和 SVA0 位合并, 用作 10 位地址的低 8 位。

当 IC SER.SARyE 位为 “0” 时, 忽视设定值。

22.2.12 从属地址寄存器 Uy (SARUy) (y=0 ~ 2)

地址 RIIC0.SARU0 0008 830Bh、RIIC1.SARU0 0008 832Bh
 RIIC0.SARU1 0008 830Dh、RIIC1.SARU1 0008 832Dh
 RIIC0.SARU2 0008 830Fh、RIIC1.SARU2 0008 832Fh



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|------------------|---|-----|
| b0 | FS | 7 位 /10 位地址格式选择位 | 0: 选择 7 位地址格式 1: 选择 10 位地址格式 | R/W |
| b2-b1 | SVA[9:8] | 10 位地址的高位 | 必须设定从属地址。 <ul style="list-style-type: none"> • 当 SARUy.FS 位为 “0” (7 位地址格式选择) 时, SVA[9:8] 位无效。 • 当 SARUy.FS 位为 “1” (10 位地址格式选择) 时, SVA[9:8] 位有效, 作为 10 位从属地址的高 2 位地址。 | R/W |
| b7-b3 | — | 保留位 | 读写值都为 “0”。 | R/W |

SARUy 寄存选择 7 位 /10 位地址格式以及设定 10 位从属地址的高位。

FS 位 (7 位 /10 位地址格式选择位)

此位选择从属地址 y (SARLy 寄存器和 SARUy 寄存器) 为 7 位地址或者 10 位地址。

当 IC SER.SARyE 位为 “1” (SARLy 寄存器和 SARUy 寄存器有效) 并且 SARUy.FS 位为 “0” 时, 选择从属地址 y 为 7 位地址格式, SARLy.SVA[7:1] 位的设定值有效, 忽视 SVA[9:8] 位和 SARLy.SVA0 位的设定值。

当 IC SER.SARyE 位为 “1” (SARLy 寄存器和 SARUy 寄存器有效) 并且 SARUy.FS 位为 “1” 时, 选择从属地址 m 为 10 位地址格式, SVA[9:8] 位和 SARLy 寄存器的设定值有效。

当 IC SER.SARyE 位为 “0” (SARLy 寄存器和 SARUy 寄存器无效) 时, SARUy.FS 位的设定值无效。

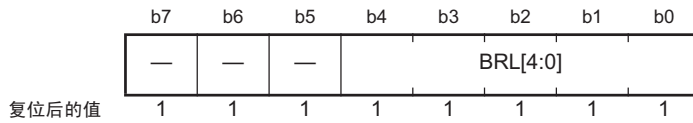
SVA[9:8] 位 (10 位地址的高位)

在选择 10 位地址格式时 (FS 位 =1), 用作 10 位地址的高 2 位地址。

当 IC SER.SARyE 位为 “1” (SARLy 寄存器和 SARUy 寄存器有效) 并且 SARUy.FS 位为 “1” 时, 设定值有效; 当 SARUy.FS 位或者 SARyE 位为 “0” 时, 忽视设定值。

22.2.13 I²C 总线位速率低电平寄存器 (ICBRL)

地址 RIIC0.ICBRL 0008 8310h、RIIC1.ICBRL 0008 8330h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|-----------------|------------------------|-----|
| b4-b0 | BRL[4:0] | 位速率 Low 电平宽度设定位 | 设定 SCL 时钟的 Low 电平宽度的值。 | R/W |
| b7-b5 | — | 保留位 | 读取值为“0”，只能写“1”。 | R/W |

ICBRL 寄存器是设定 SCL 时钟的 Low 电平宽度的 5 位寄存器。

在发生 SCL 自动保持 Low 电平时（参照“22.8 SCL 的 Low 电平自动保持功能”），ICBRL 寄存器用作数据准备时间的确保寄存器。因此，在 RIIC 只用于从属模式的情况下，设定值不能小于数据准备时间（注 1）的值。

通过 ICMR1.CKS[2:0] 位选择的内部基准时钟源（ $IIC\phi$ ），ICBRL 寄存器对 Low 电平宽度进行计数。

注 1. 数据准备时间 (t_{SU:DAT})

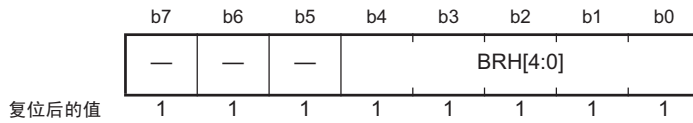
250ns（~ 100kbps：标准模式 [Sm]）

100ns（~ 400kbps：快速模式 [fm]）

50ns（~ 1Mbps：快速模式+[fm+]）

22.2.14 I²C 总线位速率高电平寄存器 (ICBRH)

地址 RIIC0.ICBRH 0008 8311h、RIIC1.ICBRH 0008 8331h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|------------------|-------------------------|-----|
| b4-b0 | BRH[4:0] | 位速率 High 电平宽度设定位 | 设定 SCL 时钟的 High 电平宽度的值。 | R/W |
| b7-b5 | — | 保留位 | 读写值都为“1”。 | R/W |

ICBRH 寄存器是设定 SCL 时钟的 High 电平宽度的 5 位寄存器，在主控模式中有效。在 RIIC 总是用于从属模式的情况下，不需要设定 High 电平宽度。

通过 ICMR1.CKS[2:0] 位选择的内部基准时钟源 (IIC ϕ)，ICBRH 寄存器对 High 电平宽度进行计数。

用以下表达式计算 I²C 传送率和 SCL 时钟的占空比。

$$\text{传送率} = (\text{PCLK} \times 10^6 \times \text{分频比 (注1)}) / (\text{SCLn 线的上升时间 [tr] (注2)} + \text{SCLn 的下降时间 [tf] (注2)} + (\text{ICBRH}+1) + (\text{ICBRL}+1))$$

$$\text{占空比} = \{ \text{SCLn 线的上升时间 [tr] (注2)} + (\text{ICBRH}+1) \} / \{ \text{SCLn 线的下降时间 [tf] (注2)} + (\text{ICBRL}+1) \}$$

注 1. 分频比为 ICMR1.CKS[2:0] 位的设定值 (1/1 ~ 1/128)。

注 2. SCLn 线的上升时间 [tr] 和下降时间 [tf] 取决于总线的总电容量 [Cb] 和上拉电阻 [Rp]，详细内容请参照 NXP 公司的 I²C 总线规格书。

ICBRH 寄存器和 ICBRL 寄存器值的设定例子如表 22.6 所示。

表 22.6 对应传送率的 ICBRH 寄存器和 ICBRL 寄存器的设定例子

| 传送率 (kbps) | 工作频率 PCLK (MHz) | | | | | | | | |
|---------------|-----------------|----------|----------|----------|----------|----------|----------|----------|----------|
| | 8 | | | 10 | | | 12.5 | | |
| | CKS[2:0] | ICBRH | ICBRL | CKS[2:0] | ICBRH | ICBRL | CKS[2:0] | ICBRH | ICBRL |
| 10 | 100b | 22 (F6h) | 25 (F9h) | 101b | 13 (EDh) | 15 (EFh) | 101b | 16 (F0h) | 20 (F4h) |
| 50 | 010b | 16 (F0h) | 19 (F3h) | 010b | 21 (F5h) | 24 (F8h) | 011b | 12 (ECh) | 15 (EFh) |
| 100 | 001b | 15 (EFh) | 18 (F2h) | 001b | 19 (F3h) | 23 (F7h) | 001b | 24 (F8h) | 29 (FDh) |
| 400 | 000b | 4 (E4h) | 10 (EAh) | 000b | 5 (E5h) | 12 (ECh) | 000b | 7 (E7h) | 16 (F0h) |
| 1000 | 000b | 2 (E2h) | 3 (E3h) | 000b | 2 (E2h) | 4 (E4h) | 000b | 3 (E3h) | 6 (E6h) |

| 传送率 (kbps) | 工作频率 PCLK (MHz) | | | | | | | | |
|---------------|-----------------|----------|----------|----------|----------|----------|----------|----------|----------|
| | 16 | | | 20 | | | 25 | | |
| | CKS[2:0] | ICBRH | ICBRL | CKS[2:0] | ICBRH | ICBRL | CKS[2:0] | ICBRH | ICBRL |
| 10 | 101b | 22 (F6h) | 25 (F9h) | 110b | 13 (EDh) | 15 (EFh) | 110b | 16 (F0h) | 20 (F4h) |
| 50 | 011b | 16 (F0h) | 19 (F3h) | 011b | 21 (F5h) | 24 (F8h) | 100b | 12 (ECh) | 15 (EFh) |
| 100 | 010b | 15 (EFh) | 18 (F2h) | 010b | 19 (F3h) | 23 (F7h) | 010b | 24 (F8h) | 29 (FDh) |
| 400 | 000b | 9 (E9h) | 20 (F4h) | 000b | 11 (EBh) | 25 (F9h) | 001b | 7 (E7h) | 16 (F0h) |
| 1000 | 000b | 4 (E4h) | 7 (E7h) | 000b | 5 (E5h) | 9 (E9h) | 000b | 6 (E6h) | 12 (ECh) |

| 传送率 (kbps) | 工作频率 PCLK (MHz) | | | | | | | | |
|---------------|-----------------|----------|----------|----------|----------|----------|----------|----------|----------|
| | 30 | | | 33 | | | 50 | | |
| | CKS[2:0] | ICBRH | ICBRL | CKS[2:0] | ICBRH | ICBRL | CKS[2:0] | ICBRH | ICBRL |
| 10 | 110b | 20 (F4h) | 24 (F8h) | 110b | 22 (F6h) | 26 (FAh) | 111b | 16 (F0h) | 20 (F4h) |
| 50 | 100b | 15 (EFh) | 18 (F2h) | 100b | 17 (F1h) | 20 (F4h) | 100b | 26 (FAh) | 31 (FFh) |
| 100 | 010b | 2 (E2h) | 3 (E3h) | 011b | 16 (F0h) | 19 (F3h) | 011b | 24 (F8h) | 29 (FDh) |
| 400 | 001b | 8 (E8h) | 19 (F3h) | 001b | 9 (E9h) | 21 (F5h) | 010b | 7 (E7h) | 16 (F0h) |
| 1000 | 000b | 7 (E7h) | 14 (EEh) | 000b | 8 (E8h) | 16 (F0h) | 000b | 12 (ECh) | 24 (F8h) |

注 1. 计算的设定例子是假设:

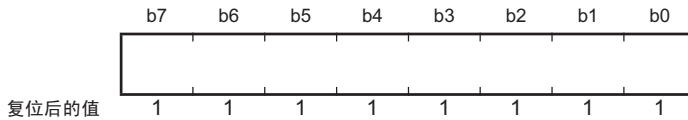
SCLn 线的上升时间 (tr) 不超过 ~100kbps, [Sm] 为 1000ns 或者 ~400kbps, [Fm] 为 300ns 或者 ~1Mbps, [Fm+] 为 120ns。

SCLn 线的下降时间 (tf) 不超过 ~400kbps, [Sm/Fm] 为 300ns 或者 ~1Mbps, [Fm+] 为 120ns。

有关 SCLn 线的上升时间 (tr) 和下降时间 (tf) 的规格值, 请参照 NXP 公司的 I²C 总线规格书。

22.2.15 I²C 总线发送数据寄存器 (ICDRT)

地址 RIIC0.ICDRT 0008 8312h、RIIC1.ICDRT 0008 8332h



ICDRT 寄存器是保存发送数据的 8 位寄存器。

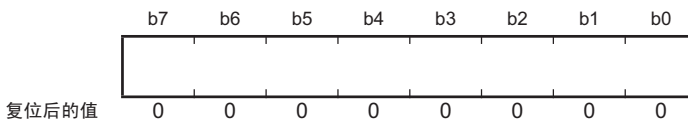
如果检测到 I²C 总线移位寄存器 (ICDRS) 为空, 就将写在 ICDRT 寄存器的发送数据传送到 ICDRS 寄存器, 在发送模式中开始数据发送。

ICDRT 寄存器和 ICDRS 寄存器为双缓冲结构, 如果在 ICDRS 寄存器的数据发送过程中将下一个要发送的数据写到 ICDRT 寄存器, 就能进行连续的发送。

能随时读写 ICDRT 寄存器。只能在发生发送数据空中断 (ICTXI) 请求时给 ICDRT 寄存器写 1 次发送数据。

22.2.16 I²C 总线接收数据寄存器 (ICDRR)

地址 RIIC0.ICDRR 0008 8313h、RIIC1.ICDRR 0008 8333h



ICDRR 寄存器是保存接收数据的 8 位寄存器。

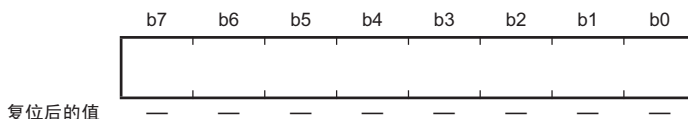
如果 1 字节数据接收结束, 就能将接收数据从 I²C 总线移位寄存器 (ICDRS) 传送到 ICDRR 寄存器, 进入能接收下一个数据的状态。

ICDRS 寄存器和 ICDRR 寄存器为双缓冲结构, 如果在 ICDRS 寄存器的数据接收过程中从 ICDRR 寄存器读接收数据, 就能进行连续的接收。

不能写 ICDRR 寄存器。只能在发生接收数据满中断 (ICRXI) 请求时读 1 次 ICDRR 寄存器。

如果不从 ICDRR 寄存器读接收数据 (ICSR2.RDRF 标志为“1”的状态) 而立即接收下一个数据, RIIC 就在 RDRF 标志变为“1”时的前一个 SCLn 时钟自动保持 Low 电平。

22.2.17 I²C 总线移位寄存器 (ICDRS)



ICDRS 寄存器是发送和接收数据的移位寄存器。

在发送时, 将发送数据从 ICDRT 寄存器传送到 ICDRS 寄存器, 从 SDA_n 引脚发送数据。在接收时, 一旦 1 字节数据接收结束, 就将数据从 ICDRS 寄存器传送到 ICDRR 寄存器。

不能直接存取 ICDRS 寄存器。

22.3 运行说明

22.3.1 通信数据的格式

I²C 总线格式由 8 位数据和 1 个应答位构成。接在开始条件或者重新开始条件后面的帧是地址帧，用于指定主控器件通信对象的从属器件。在指定新的从属器件或者发行停止条件前，指定的从属器件有效。

I²C 总线的格式及其总线时序分别如图 22.3 和图 22.4 所示。

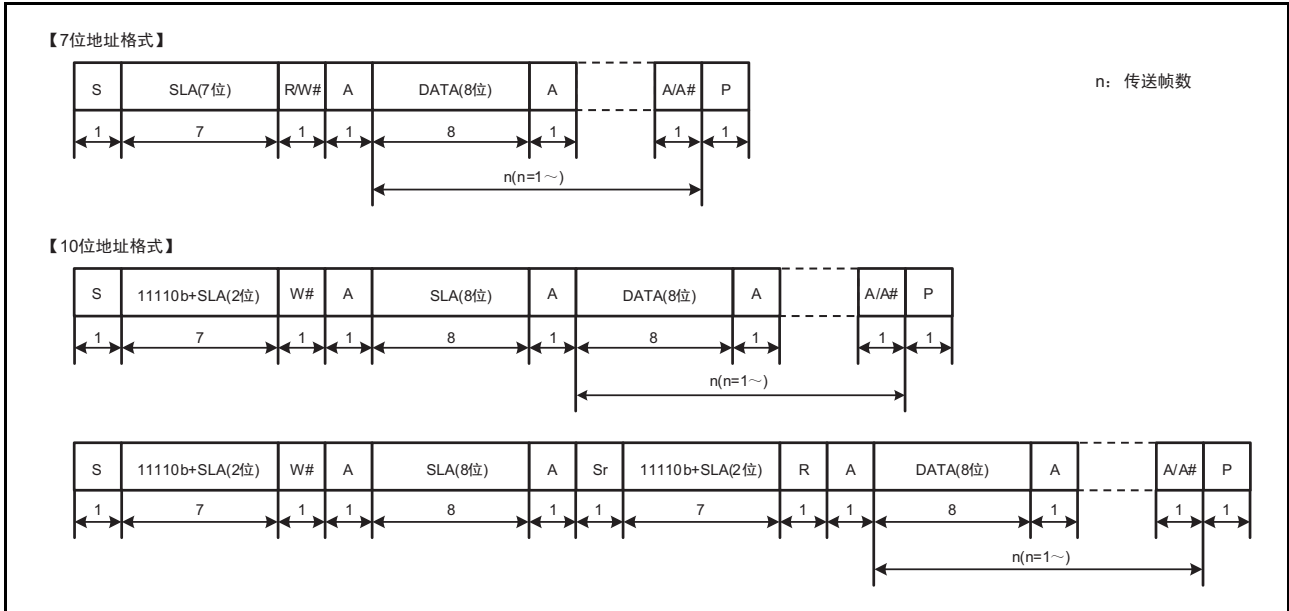


图 22.3 I²C 总线格式

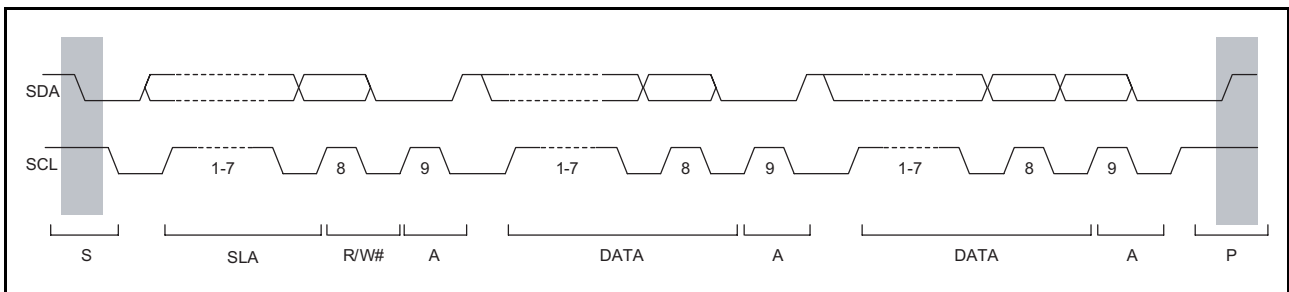


图 22.4 I²C 总线时序 (SLA 为 7 位)

【符号说明】

- S : 开始条件。主控器件在 SCLn 线为 High 电平的状态下将 SDA_n 线从 High 电平变为 Low 电平。
- SLA : 从属地址。主控器件选择从属器件。
- R/W# : 发送和接收的方向。当 R/W# 为 “1” 时，将数据从从属器件发送到主控器件；当 R/W# 为 “0” 时，将数据从主控器件发送到从属器件。
- A : 应答。接收器件将 SDA_n 线置为 Low 电平（在发送模式中，从属器件返回应答；在接收模式中，主控器件返回应答）。
- Sr : 重新开始条件。主控器件在 SCLn 线为 High 电平的状态下经过准备时间后将 SDA_n 线从 High 电平变为 Low 电平。
- DATA : 发送和接收的数据。
- P : 停止条件。主控器件在 SCLn 线为 High 电平的状态下将 SDA_n 线从 Low 电平变为 High 电平。

22.3.2 初始设定

在开始发送或者接收数据时，必须按照图 22.5 所示的步骤对 RIIC 进行初始化。

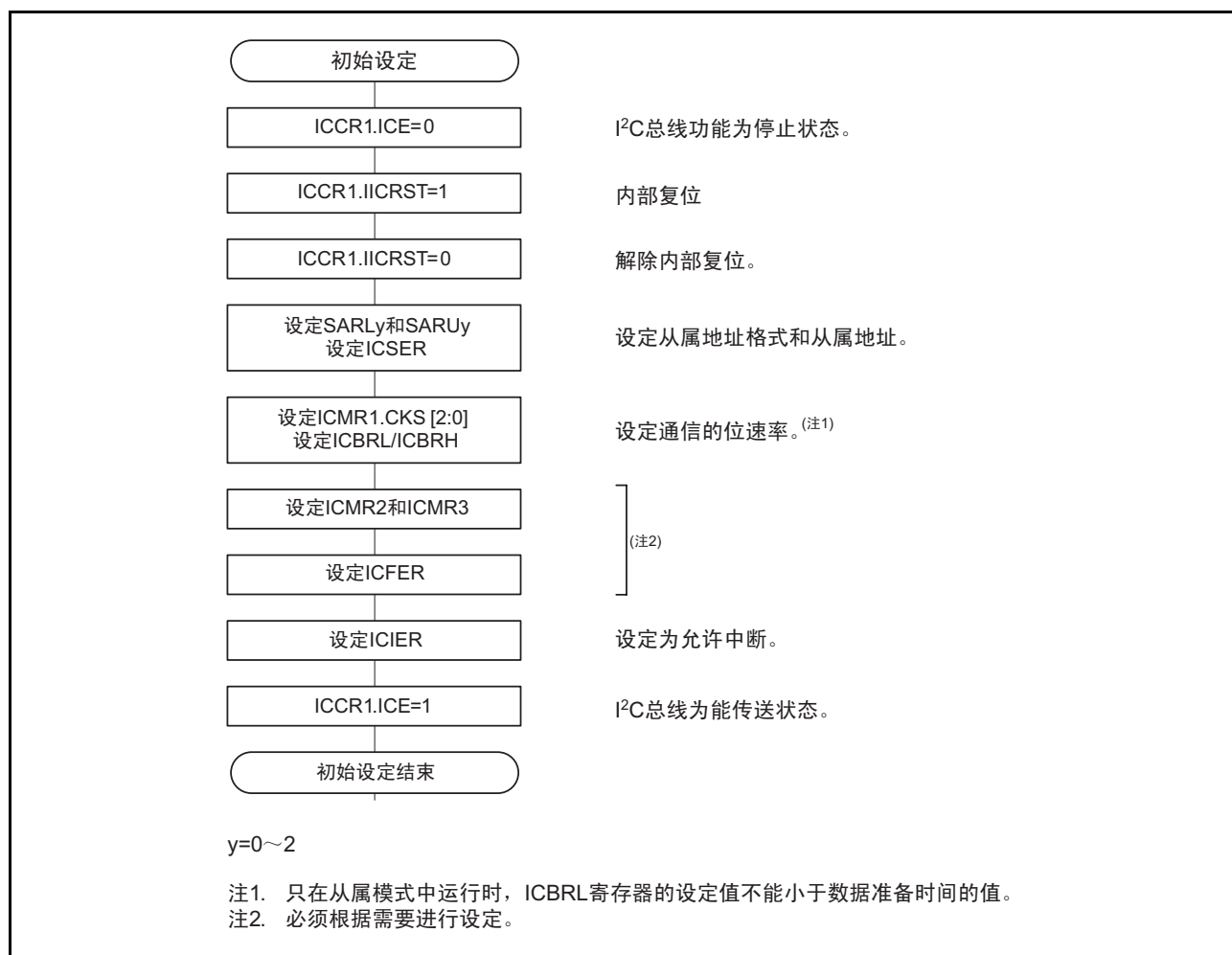


图 22.5 RIIC 的初始化流程图例子

22.3.3 主控发送

在主控发送模式中，主控器件的 RIIC 输出 SCL 时钟和发送数据，从属器件返回应答。主控发送模式的使用例子和运行时序分别如图 22.6 和图 22.7 ~ 图 22.9 所示。

主控发送模式的发送步骤和运行如下所示：

1. 在将 ICCR1.ICE 位置 “0”（功能停止状态）的状态下，通过将 ICCR1.IICRST 位置 “1”（内部复位）后又置 “0”（解除复位），对 ICSR1 寄存器的各标志和内部状态进行初始化。然后，设定 SARLy、SARUy、ICSER、ICMR1、ICBRH、ICBRL 寄存器（y=0~2）并且必须根据需要设定其他寄存器（有关 RIIC 的初始设定请参照图 22.5）。一旦设定完所需的寄存器，就必须将 ICE 位置 “1”（能传送状态）。在 RIIC 已初始化的情况下，不需要此步骤。
2. 读 ICCR2.BBSY 标志，在确认总线为释放状态后将 ICCR2.ST 位置 “1”（请求发行开始条件）。如果 RIIC 接受开始条件的发行请求，就发行开始条件。如果 RIIC 检测到开始条件，就自动将 BBSY 标志和 ICSR2.START 标志置 “1”，并且自动将 ST 位置 “0”。此时，如果在 ST 位为 “1” 的状态下 RIIC 自身发送的 SDA 信号与 SDA_n 线的信号状态相同，并且检测到开始条件，RIIC 就视为通过 ST 位正确地发行了开始条件，在将 ICCR2.MST 位和 ICCR2.TRS 位自动置 “1” 后变为主控发送模式。另外，ICSR2.TDRE 因 TRS 位为 “1” 而自动变为 “1”。
3. 必须在确认 ICSR2.TDRE 标志为 “1” 后将发送数据（从属地址和 R/W# 位）写到 ICDRT 寄存器。一旦将发送数据写到 ICDRT 寄存器，TDRE 标志就自动变为 “0”，在将数据从 ICDRT 寄存器传送到 ICDRS 寄存器后，TDRE 标志再次变为 “1”。一旦结束包含 R/W# 位的从属地址的发送，就根据被发送的 R/W# 位自动更改 TRS 位，并且选择发送模式或者接收模式。如果接收到为 “0” 的 R/W# 位，RIIC 就继续保持主控发送模式的状态。

此时，如果 ICSR2.NACKF 标志为 “1”，因为没有识别到从属器件或者发生了通信故障，所以必须发行停止条件。通过给 ICCR2.SP 位写 “1” 来发行停止条件。

在用 10 位地址格式进行发送时，必须先在第 1 次地址发送处理中给 ICDRT 寄存器写 1111 0b+ 从属地址的高 2 位+W，然后在第 2 次地址发送处理中给 ICDRT 寄存器写从属地址的低 8 位。

4. 必须在确认 ICSR2.TDRE 标志为 “1” 后将发送数据写到 ICDRT 寄存器。在准备好发送数据之前或者发行停止条件前，RIIC 自动将 SCL_n 线保持 Low 电平。
5. 在将要发送的全部字节写到 ICDRT 寄存器后，必须在等待 ICSR2.TEND 标志变为 “1” 后再给 ICCR2.SP 位写 “1”（请求发行停止条件）。如果 RIIC 接受停止条件的发行请求，就发行停止条件。
6. 如果 RIIC 检测到停止条件，ICCR2.MST 位和 ICCR2.TRS 位就自动变为 “00b” 并且转移到从属接收模式，而且 ICSR2.TDRE 标志和 ICSR2.TEND 标志因检测到停止条件而自动变为 “0”，ICSR2.STOP 标志变为 “1”。
7. 在确认 ICSR2.STOP 标志为 “1” 后，必须将 ICSR2.NACKF 标志和 ICSR2.STOP 标志置 “0”，以便进行下一轮的通信。

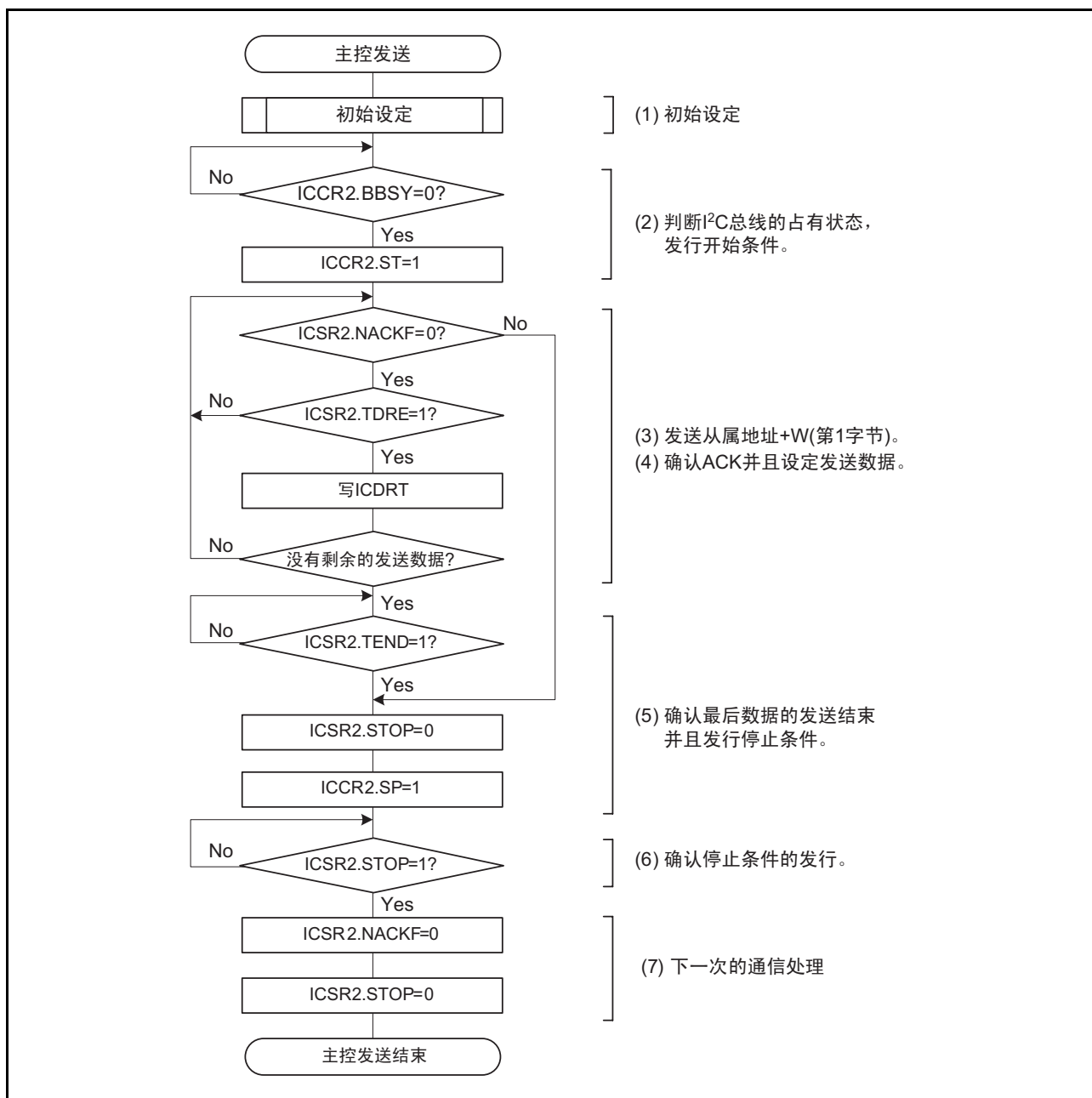


图 22.6 主控发送模式的流程图例子

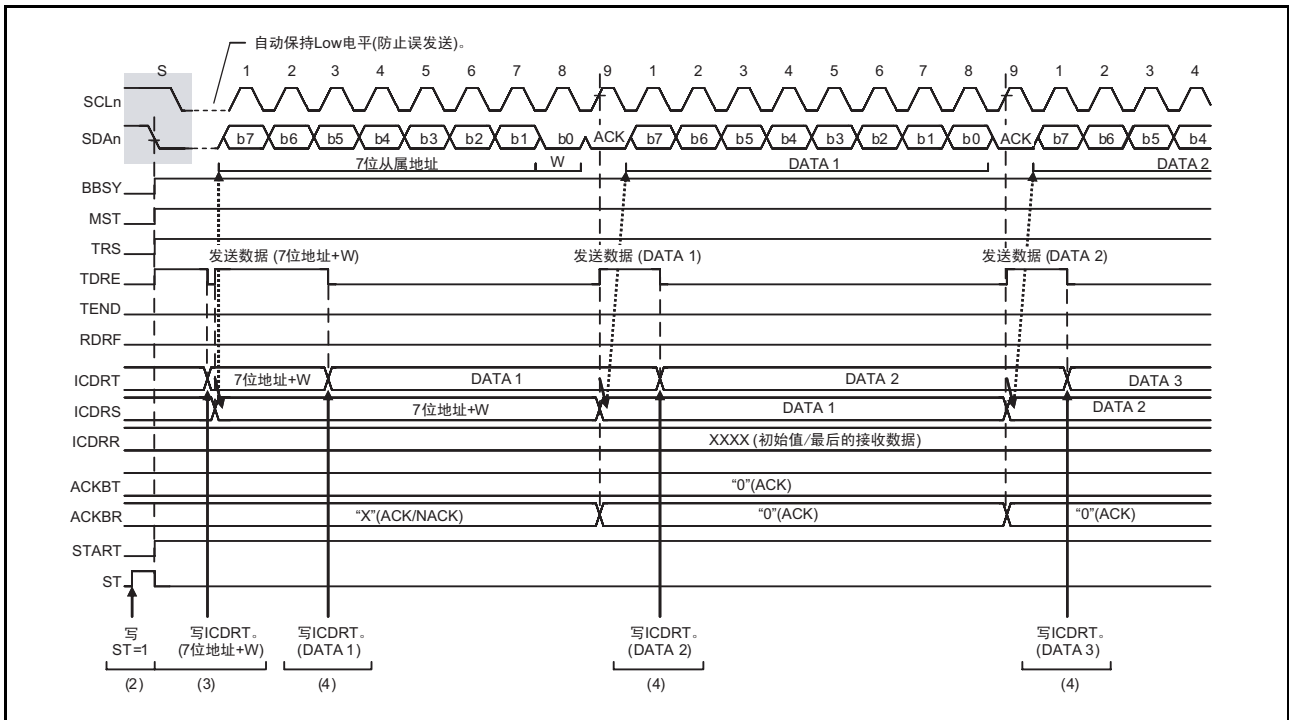


图 22.7 主控发送模式的运行时序 (1) (7 位地址格式的情况)

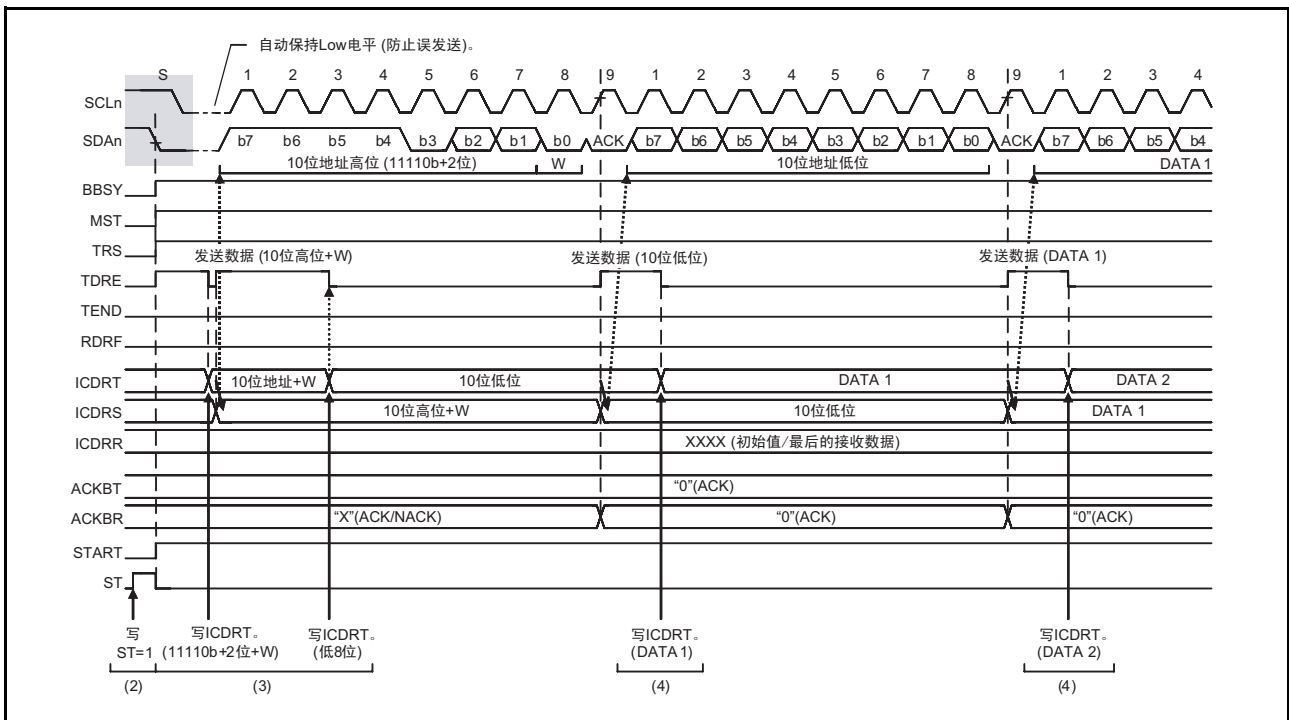


图 22.8 主控发送模式的运行时序 (2) (10 位地址格式的情况)

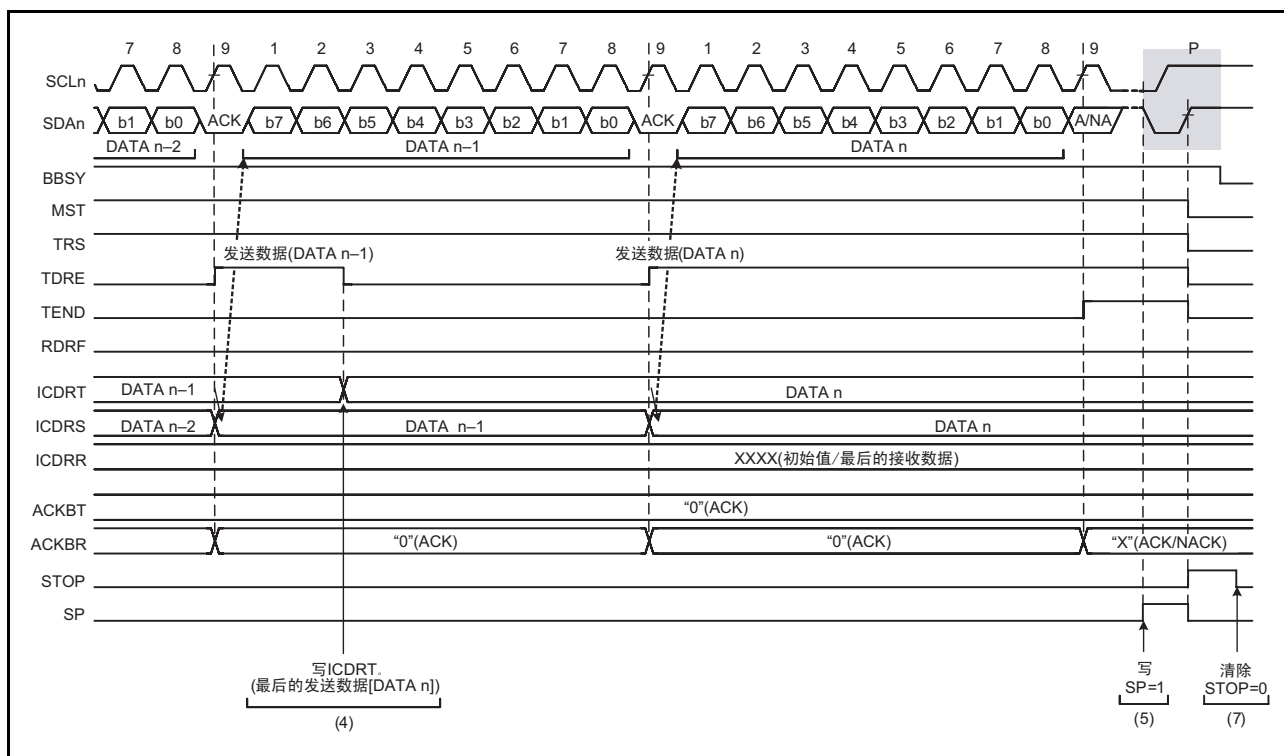


图 22.9 主控发送模式的运行时序 (3)

22.3.4 主控接收

在**主控接收模式**中，主控器件的**RIIC**输出**SCL**时钟，从属器件接收数据后返回应答。因为首先需要将从属地址发送到从属器件，所以必须先**在**主控发送模式中发送从属地址，然后**在**主控接收模式中接收数据。

主控接收模式的使用例子（7位地址格式）和运行时序分别如图22.10和图22.11～图22.13所示。

主控接收模式的接收步骤和运行如下所示：

1. 在将**ICCR1.ICE**位置“0”（功能停止状态）的状态下，通过将**ICCR1.IICRST**位置“1”（内部复位）后又置“0”（解除复位），对**ICSR1**寄存器的各标志和内部状态进行初始化。然后，设定**SARLy**、**SARUy**、**ICSER**、**ICMR1**、**ICBRH**、**ICBRL**寄存器（ $y=0\sim 2$ ）并且必须根据需要设定其他的寄存器（有关**RIIC**的初始设定请参照图22.5）。必须在设定完所需的寄存器后将**ICE**位置“1”（能传送状态）。在**RIIC**已初始化的情况下，不需要此步骤。
2. 读**ICCR2.BBSY**标志，在确认总线为释放状态后将**ICCR2.ST**位置“1”（请求发行开始条件）。如果**RIIC**接受开始条件的发行请求，就发行开始条件。如果**RIIC**检测到开始条件，就自动将**BBSY**标志和**ICSR2.START**标志置“1”，并且自动将**ST**位置“0”。此时，如果在**ST**位为“1”的状态下**RIIC**自身发送的**SDA**信号与**SDAn**线的状态相同，并且检测到开始条件，**RIIC**就视为通过**ST**位正常地发行了开始条件，在将**ICCR2.MST**位和**ICCR2.TRS**位自动置“1”后变为**主控发送模式**。**ICSR2.TDRE**标志因**TRS**位为“1”而自动变为“1”。
3. 必须在确认**ICSR2.TDRE**标志为“1”后将发送数据（从属地址和**R/W#**位）写到**ICDRT**寄存器。一旦将发送数据写到**ICDRT**寄存器，**TDRE**标志就自动变为“0”，在将数据从**ICDRT**寄存器传送到**ICDRS**寄存器后，**TDRE**标志再次变为“1”。一旦结束包含**R/W#**位的从属地址的发送，就根据被发送的**R/W#**位自动更改**ICCR2.TRS**位，并且选择发送模式或者接收模式。如果**RIIC**接收到为“1”的**R/W#**位，就在第9个时钟的上升沿将**TRS**位置“0”后转移到**主控接收模式**，此时**TDRE**标志变为“0”，**ICSR2.RDRF**标志自动变为“1”。
此时，如果**ICSR2.NACKF**标志为“1”，因为没有识别到从属器件或者发生了通信故障，所以必须发行停止条件。通过将**ICCR2.SP**位置“1”来发行停止条件。
在用10位地址格式进行**主控接收**时，先在**主控发送模式**中发送10位地址，然后发行重新开始条件。接着，通过发送1111 0b+从属地址的高2位+R，转移到**主控接收模式**。
4. 如果在确认**ICSR2.RDRF**标志为“1”后虚读**ICDRR**寄存器，**RIIC**就在输出**SCL**时钟后开始接收。
5. 结束1字节数据的接收，在**ICMR3.RDRFS**位设定的**SCL**时钟的第8个时钟或者第9个时钟的上升沿**ICSR2.RDRF**标志变为“1”。此时，如果读**ICDRR**寄存器，就能读到接收数据，同时**RDRF**标志自动变为“0”，并且将**ICMR3.ACKBT**位的设定值返回给**SCL**时钟的第9个时钟的应答位。
6. 在下一个接收数据为最后字节的情况下，必须在读**ICDRR**寄存器（最后字节-第1个字节）前将**ICMR3.WAIT**位置“1”（有**WAIT**）。因此，在接收下一个数据（最后字节）后的第9个时钟的下降沿将**SCLn**线固定为Low电平，处于能发行停止条件的状态。
在**ICMR3.RDRFS**位为“0”并且需要将“下一个数据接收后通信结束”通知从属器件时，必须将**ICMR3.ACKBT**位置“1”（**NACK**）。
7. 在读**ICDRR**寄存器（最后字节-第1个字节）后，必须先确认**ICSR2.RDRF**标志为“1”，然后给**ICCR2.SP**位写“1”（请求发行停止条件）并且读**ICDRR**寄存器（最后字节）。**RIIC**通过读**ICDRR**寄存器来解除**WAIT**状态，在结束第9个时钟的Low电平输出或者解除**SCL**线保持的Low电平后发行停止条件。
8. 如果**RIIC**检测到停止条件，**ICCR2.MST**位和**ICCR2.TRS**位就自动变为“00b”并且转移到**从属接收模式**，而且**ICSR2.STOP**标志因检测到停止条件而变为“1”。
9. 在确认**ICSR2.STOP**标志为“1”后，必须将**ICSR2.NACKF**和**ICSR2.STOP**标志置“0”，以便进行下一次的通信。

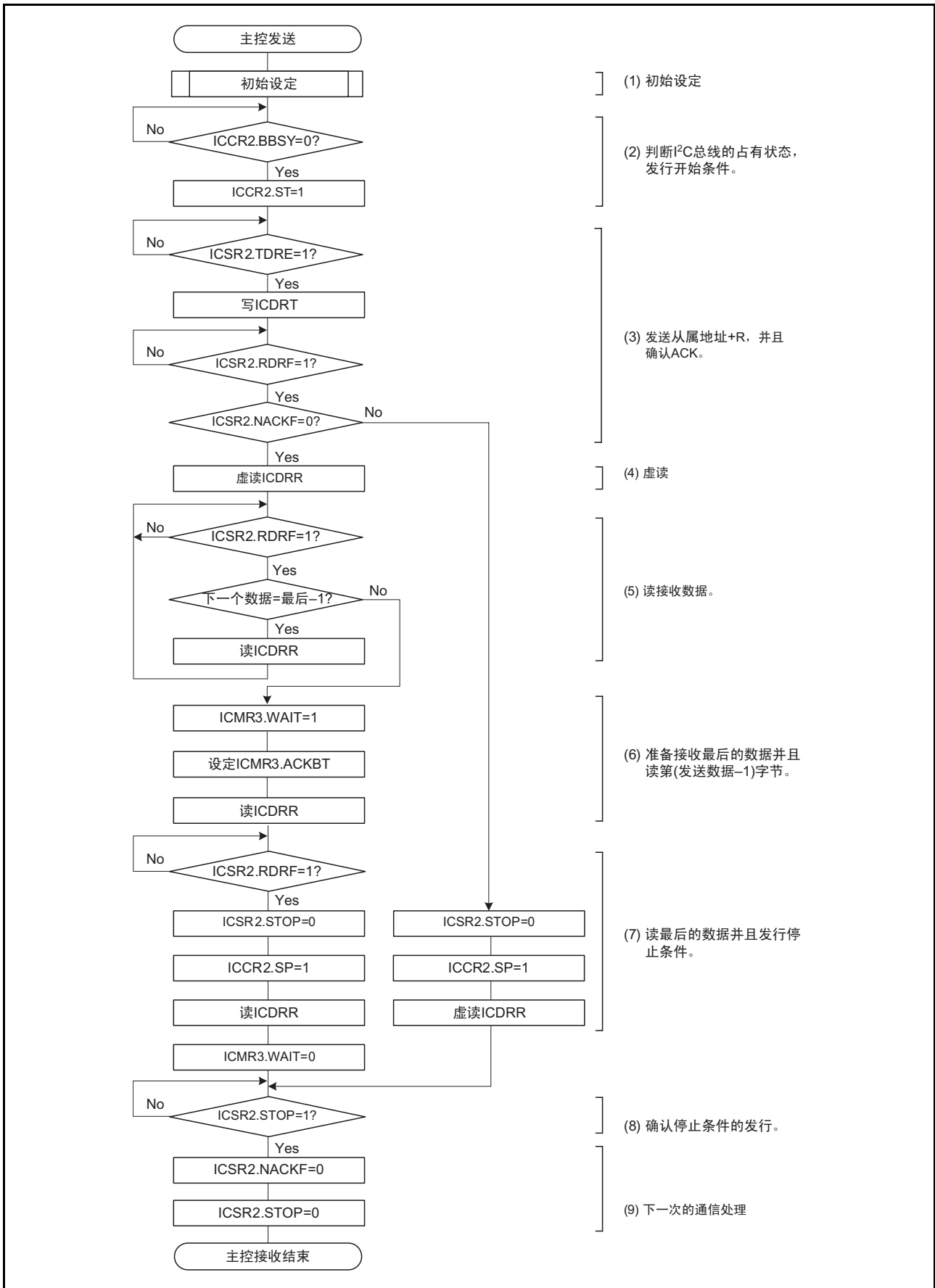


图 22.10 主控接收模式的流程图例子 (7 位地址格式的情况)

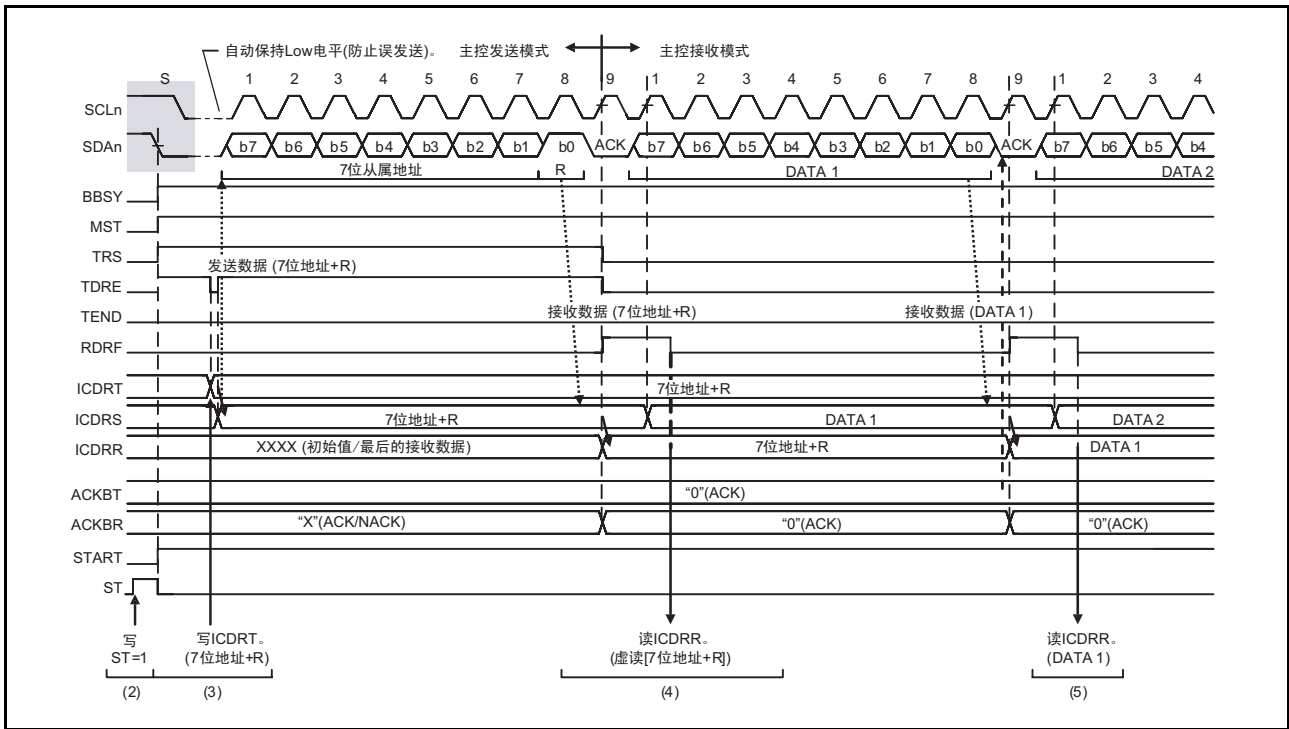


图 22.11 主控接收模式的运行时序 (1) (7 位地址格式, RDRFS=0 的情况)

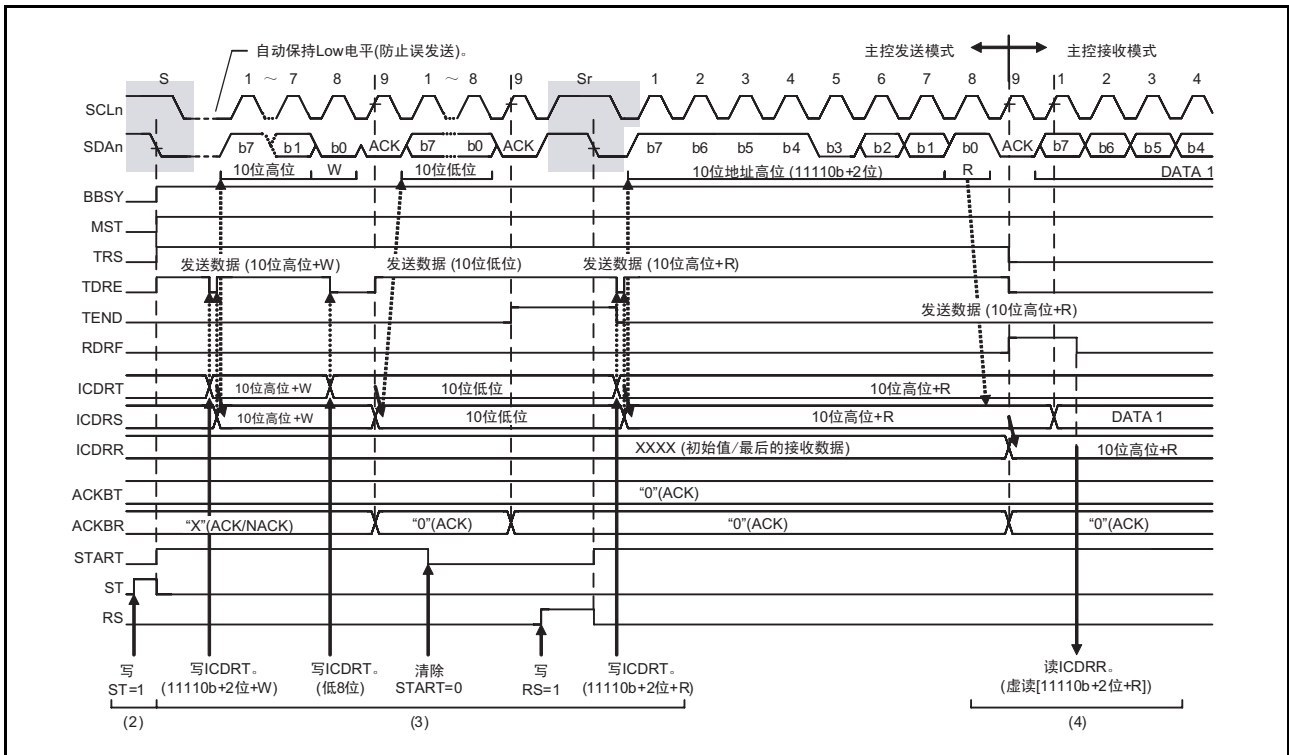


图 22.12 主控接收模式的运行时序 (2) (10 位地址格式, RDRFS=0 的情况)

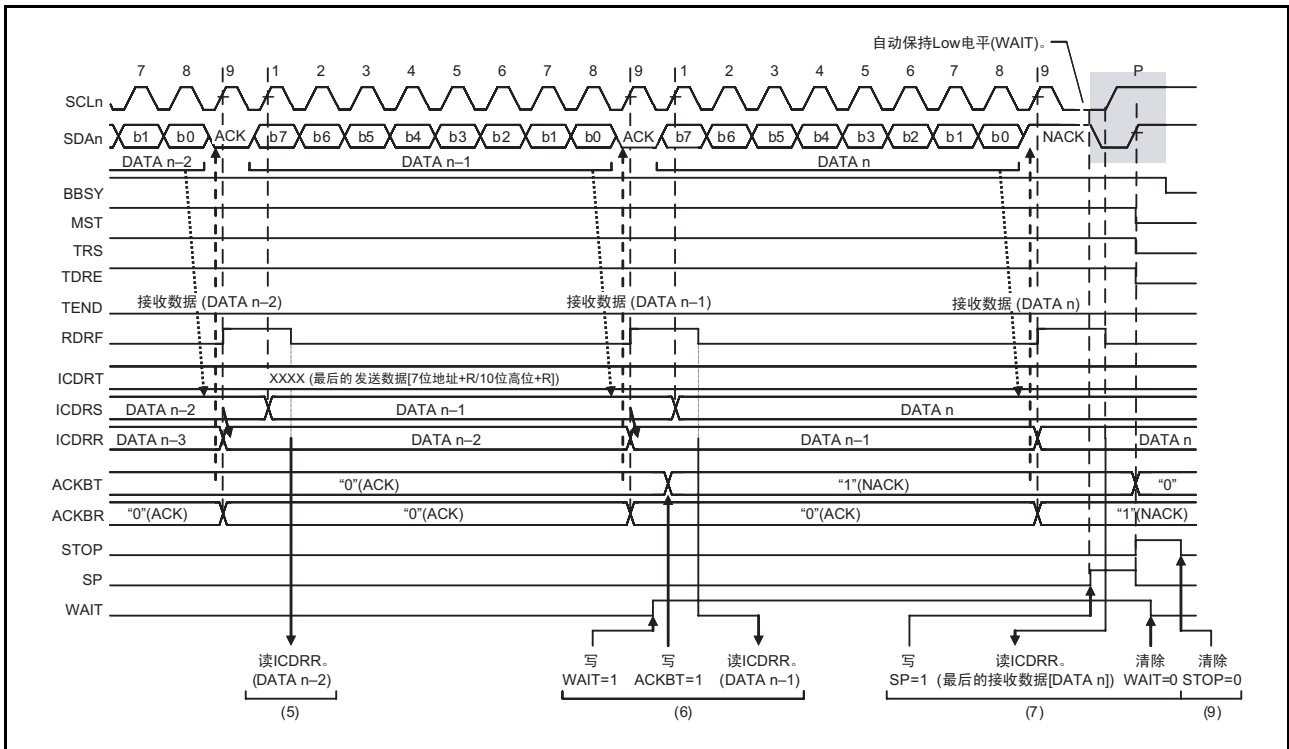


图 22.13 主控接收模式的运行时序 (3) (RDRFS=0 的情况)

22.3.5 从属发送

在从属发送模式中，主控器件输出 SCL 时钟，从属器件的 RIIC 发送数据，并且主控器件返回应答。
从属发送模式的使用例子和运行时序分别如图 22.14 和图 22.15 ~ 图 22.16 所示。

从属发送模式的发送步骤及其运行如下所示：

1. 必须按照图 22.5 所示的步骤对 RIIC 进行初始设定。在 RIIC 已初始化的情况下，不需要此步骤。从初始设定结束后到从属地址匹配前，RIIC 处于待机状态。
2. 如果从属地址匹配，RIIC 就在 SCL 时钟的第 9 个时钟上升沿将对应的 ICSR1.HOA 位、ICSR1.GCA 位、ICSR1.AASy 位 (y=0~2) 置“1”，并且将 ICMR3.ACKBT 位的设定值返回给 SCL 时钟的第 9 个时钟的应答位。此时，如果接收的 R/W# 位为“1”，就将 ICCR2.TRS 位和 ICSR2.TDRE 标志置“1”并且自动转换为从属发送模式。
3. 必须在确认 ICSR2.TDRE 标志为“1”后将发送数据写到 ICDRT 寄存器。此时，如果在 ICFER.NACKE 位为“1”的状态下主控器件没有应答（接收到 NACK），RIIC 就中止下一次的通信。
4. 在 ICSR2.NACKF 标志变为“1”或者将发送数据写到 ICDRT 寄存器后，必须在 ICSR2.TDRE 标志为“1”的状态下等到 ICSR2.TEND 标志变为“1”。当 ICSR2.NACKF 标志或者 TEND 标志为“1”时，RIIC 在第 9 个时钟下降后将 SCLn 线保持 Low 电平。
5. 当 ICSR2.NACKF 标志或者 ICSR2.TEND 标志为“1”时，必须通过虚读 ICDRR 寄存器来结束处理，从而释放 SCLn 线。
6. 如果 RIIC 检测到停止条件，就自动将 ICSR1.HOA 位、ICSR1.GCA 位、ICSR1.AASy 位 (y=0~2)、ICSR2.TDRE 位、ICSR2.TEND 标志和 ICCR2.TRS 位置“0”并且转移到从属接收模式。
7. 在确认 ICSR2.STOP 标志为“1”后，必须将 ICSR2.NACKF 标志和 ICSR2.STOP 标志置“0”，以便进行下一次的通信。

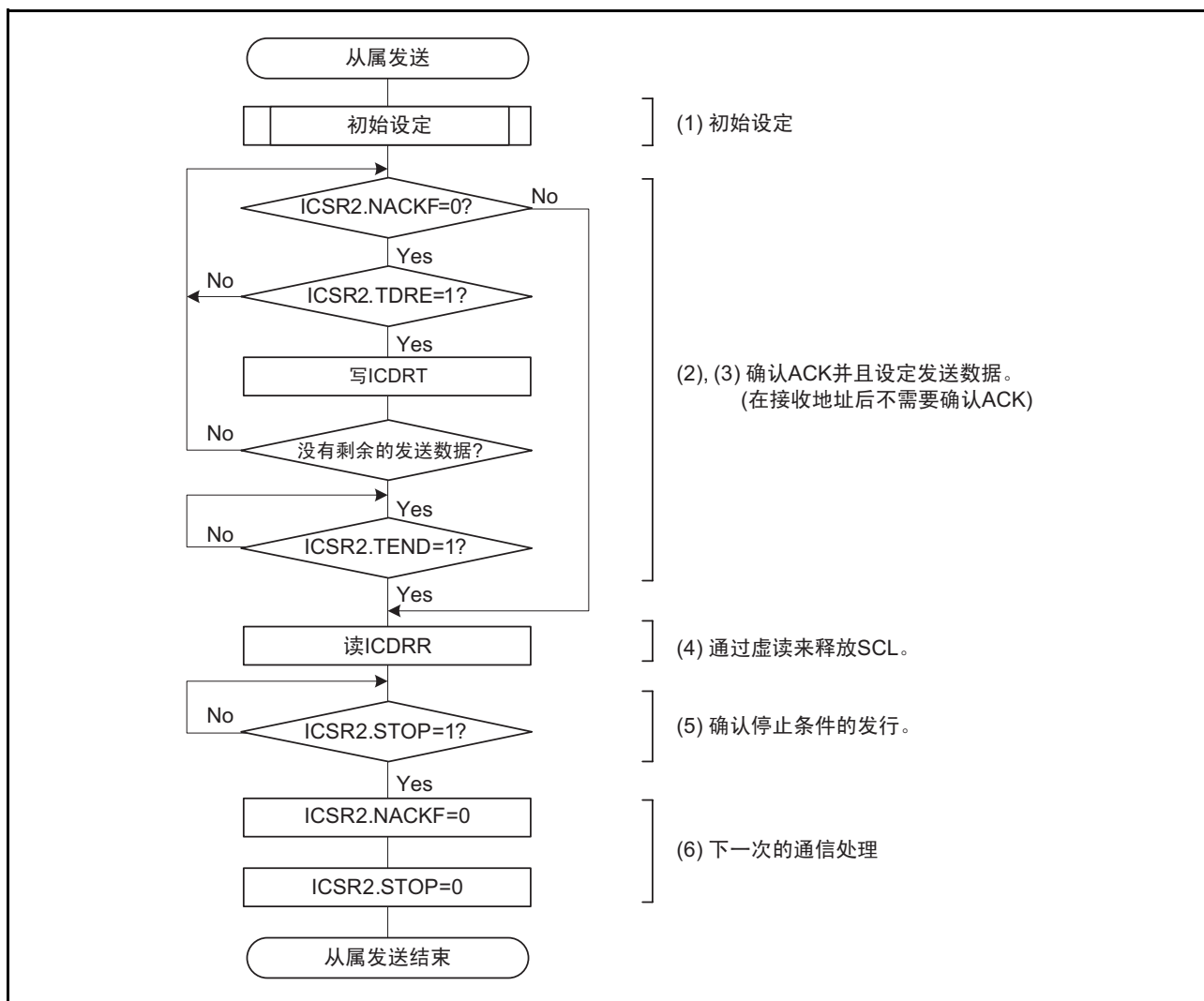


图 22.14 从属发送模式的流程图例子

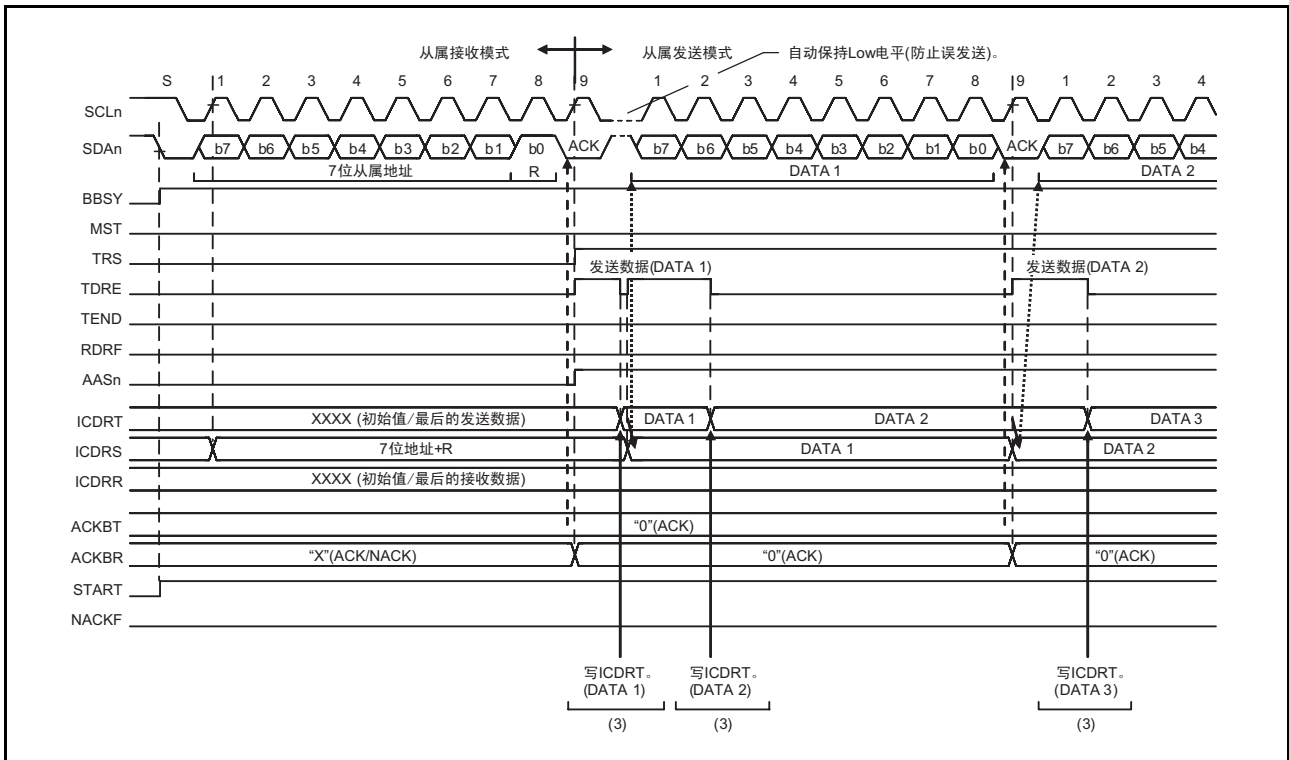


图 22.15 从属发送模式的运行时序 (1) (7 位地址格式的情况)

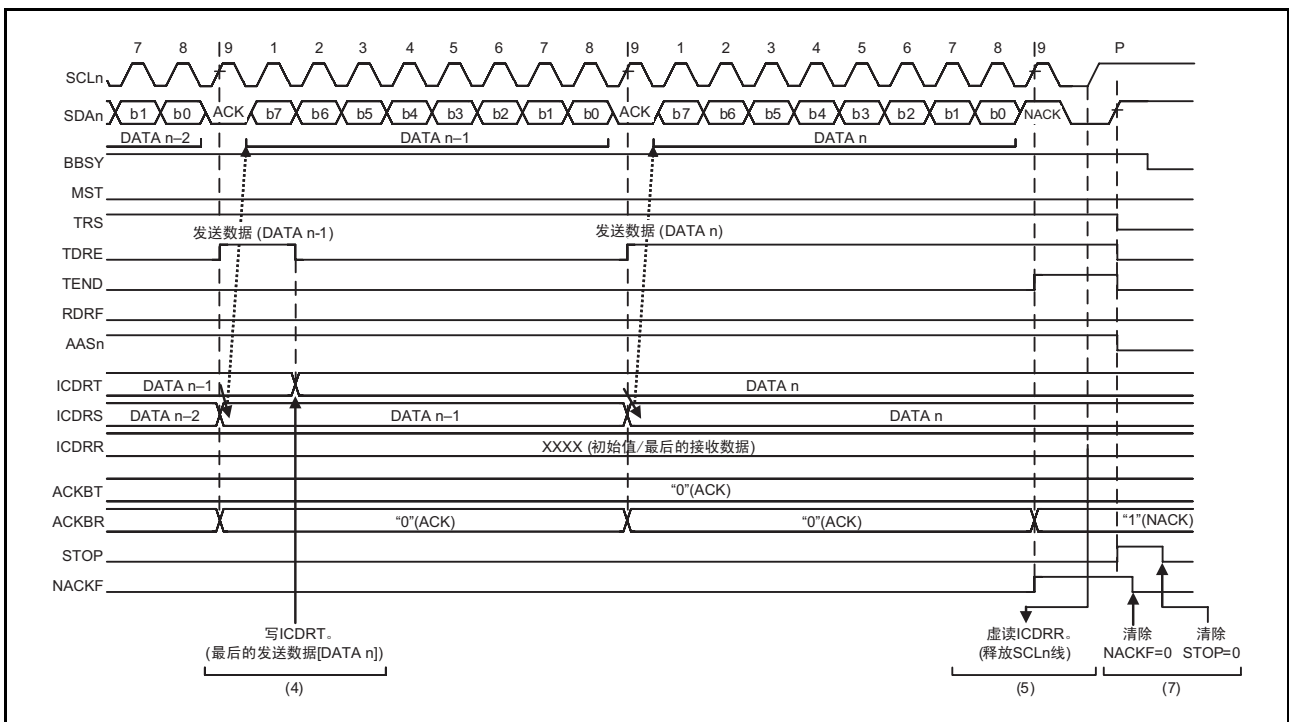


图 22.16 从属发送模式的运行时序 (2)

22.3.6 从属接收

在从属接收模式中，主控制器输出 SCL 时钟和发送数据，从属器件的 RIIC 返回应答。

从属接收模式的使用例子和运行时序分别如图 22.17 和图 22.18 ~ 图 22.19 所示。

从属接收模式的接收步骤和运行如下所示：

1. 必须按照图 22.5 所示的步骤对 RIIC 进行初始设定。在 RIIC 已初始化的情况下，不需要此步骤。在初始设定结束后到从属地址匹配前，RIIC 处于待机状态。
2. 如果从属地址匹配，RIIC 就在 SCL 时钟的第 9 个时钟的上升沿将对应的 ICSR1.HOA 位、ICSR1.GCA 位、ICSR1.AASy 位 (y=0~2) 置“1”并且将 ICMR3.ACKBT 位的设定值返回给 SCL 时钟的第 9 个时钟的应答位。此时，如果接收的 R/W# 位为“0”，就继续保持从属接收模式的状态并且将 ICSR2.RDRF 标志置“1”。
3. 在确认 ICSR2.STOP 标志为“0”并且 ICSR2.RDRF 标志为“1”后，第 1 次必须虚读 ICDRR 寄存器（虚读的接收数据在 7 位地址格式时为从属地址+R/W# 位，在 10 位地址格式时为低 8 位地址）。
4. 如果读 ICDRR 寄存器，RIIC 就自动将 ICSR2.RDRF 标志置“0”。如果在延迟了 ICDRR 寄存器的读操作并且 RDRF 标志为“1”的状态下接收下一个数据，RIIC 就在 RDRF 标志变为“1”时的前一个 SCL 时钟下降沿将 SCLn 线保持 Low 电平。通过读 ICDRR 寄存器来解除被保持的 Low 电平，RIIC 释放 SCLn 线。必须在 ICSR2.STOP 标志和 ICSR2.RDRF 标志都为“1”或者接收完全部数据时读 ICDRR 寄存器。
5. 如果 RIIC 检测到停止条件，就自动将 ICSR1.HOA 位、ICSR1.GCA 位和 ICSR1.AASy 位 (y=0~2) 置“0”。
6. 在确认 ICSR2.STOP 标志为“1”后，必须将 ICSR2.STOP 标志置“0”，以便进行下一次的通信。

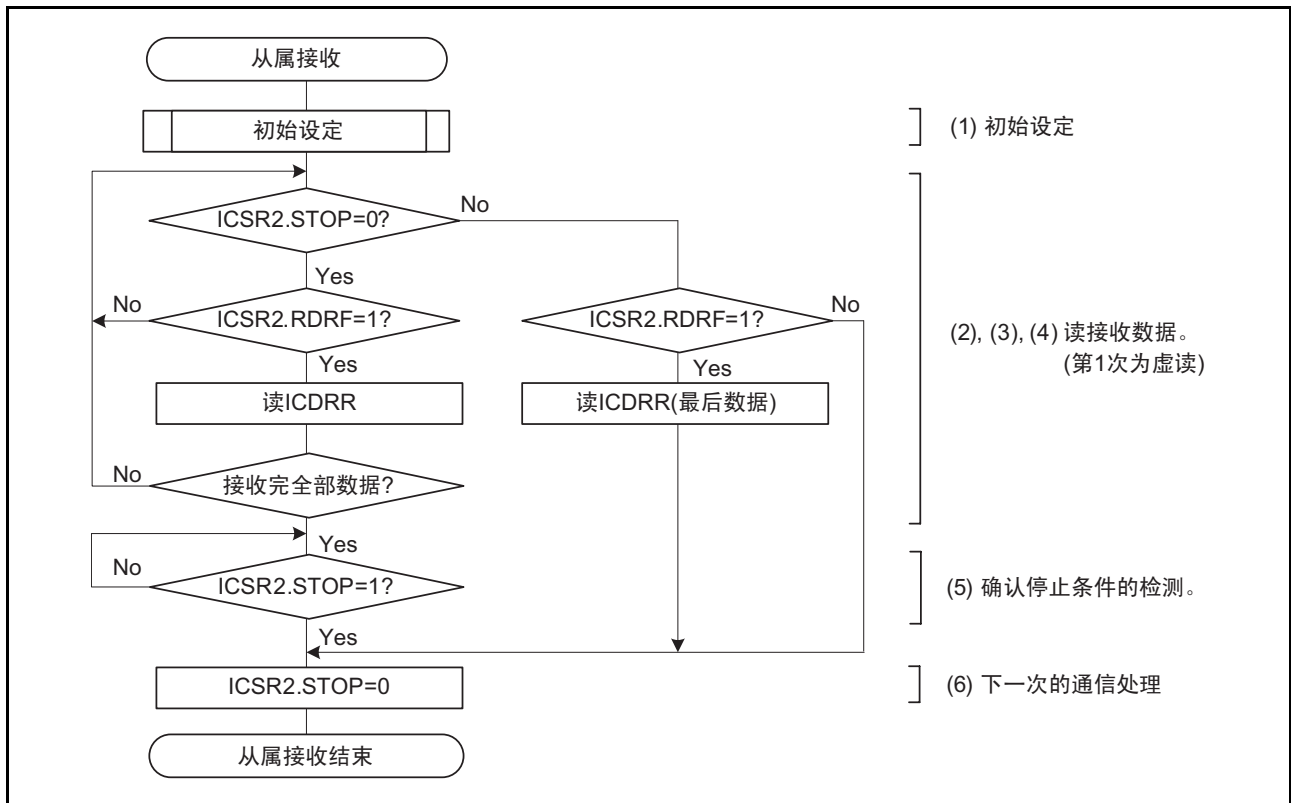


图 22.17 从属接收模式的流程图例子

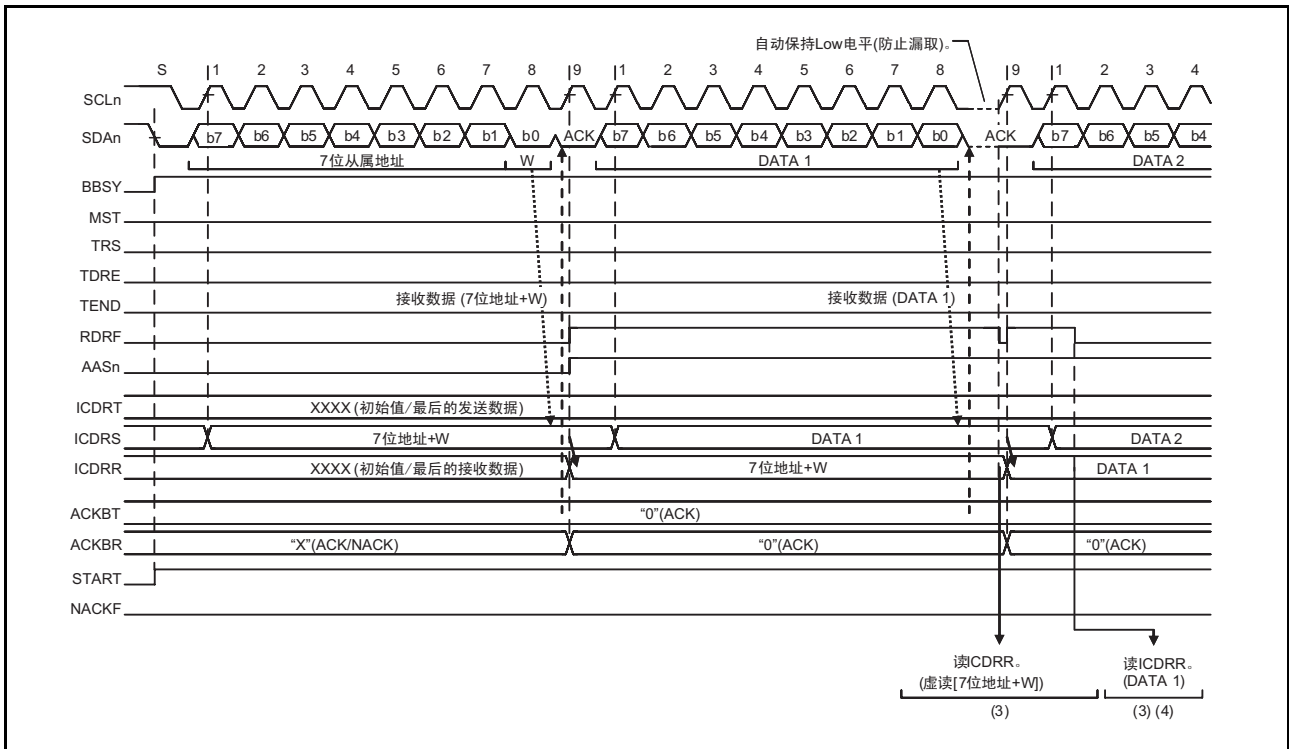


图 22.18 从属接收模式的运行时序 (1) (7 位地址格式, RDRFS=0 的情况)

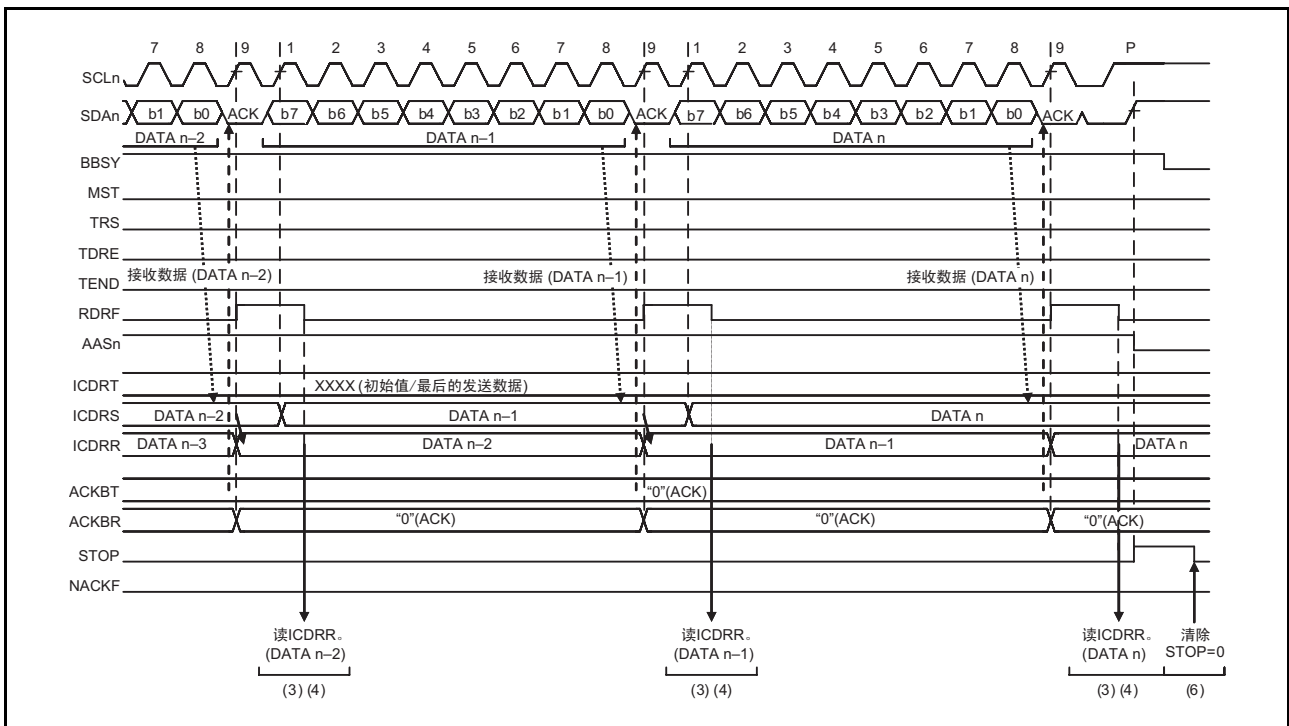


图 22.19 从属接收模式的运行时序 (2) (RDRFS=0 的情况)

22.4 SCL 同步电路

如果 RIIC 的 SCL 时钟生成电路检测到 SCLn 线的上升沿, 就开始对 ICBRH 寄存器设定的 High 电平宽度进行计数, 在结束 High 电平宽度的计数时将 SCLn 线驱动为 Low 电平。如果检测到 SCLn 线的下降沿, 就开始对 ICBRL 寄存器设定的 Low 电平宽度进行计数, 在结束 Low 电平宽度的计数时结束 SCLn 线的 Low 电平驱动, 然后释放 SCLn 线。通过此方法生成 SCL 时钟。

在多主控模式中使用 I²C 总线时, 有可能因和其他主控器件的竞争而使 SCL 时钟和 SCL 时钟发生冲突。如果 SCL 时钟发生冲突, 主控器件就需要与 SCL 时钟同步, 而且需要逐位进行 SCL 时钟的同步。RIIC 具有 SCL 同步电路功能, 在主导模式中监视 SCLn 线, 并且一边逐位取得同步, 一边生成 SCL 时钟。

RIIC 检测到 SCLn 线的上升沿并且在 ICBRH 寄存器设定的 High 电平进行计数的过程中, 如果因其他主控器件的 SCL 时钟输出而使 SCLn 线下降, RIIC 就在检测到 SCLn 线的下降沿时中止 High 电平宽度的递增计数, 并且在 SCLn 线被驱动为 Low 电平的同时开始对 ICBRL 寄存器设定的 Low 电平宽度进行递增计数, 在结束 Low 电平宽度的计数时结束 SCLn 线的 Low 电平驱动, 然后释放 SCLn 线。此时, 如果其他主控器件的 SCL 时钟的 Low 电平宽度大于 RIIC 设定的 Low 电平宽度, 就延长 SCL 时钟的 Low 电平宽度。当其他主控器件结束 Low 电平输出时, 释放 SCLn 线并且 SCL 时钟上升。因此, 在发生 SCL 时钟输出冲突时, SCL 时钟的 High 电平宽度与短时钟同步, Low 电平宽度与长时钟同步。此 SCL 同步在 ICFER.SCLE 位为 “1” 时有效。

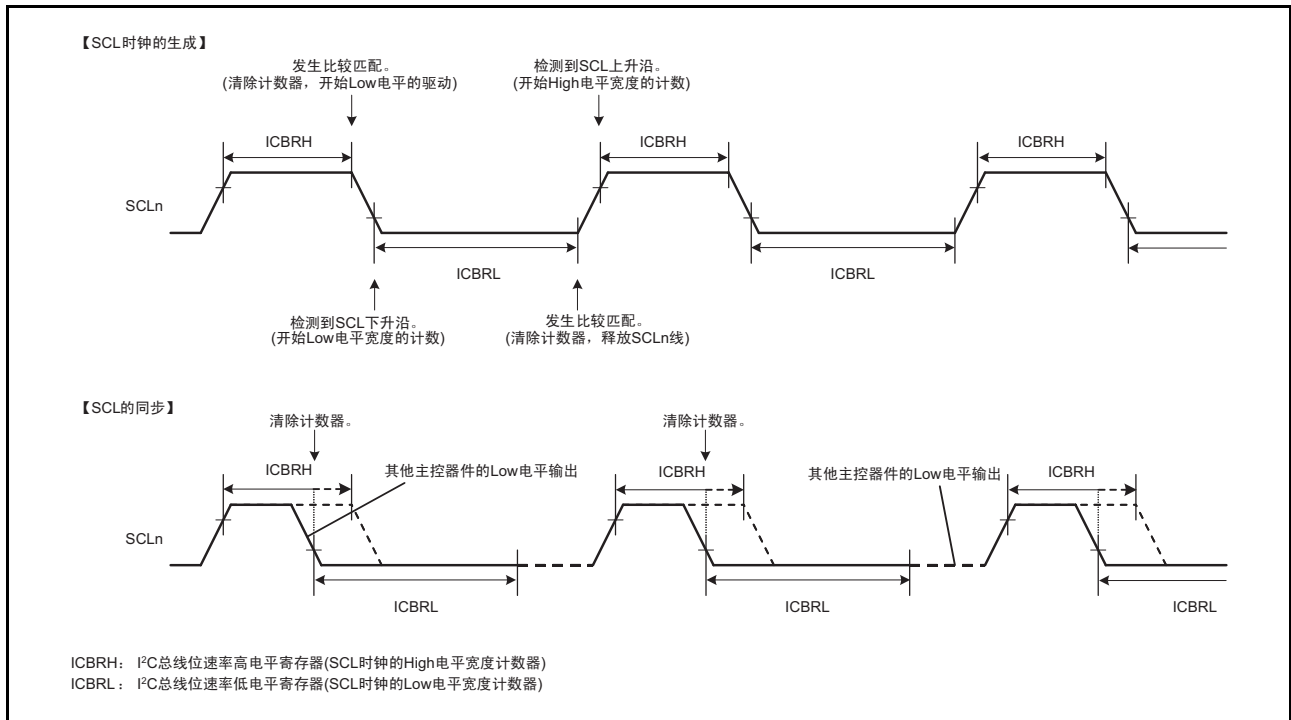


图 22.20 RIIC 的 SCL 时钟生成和 SCL 同步

22.5 SDA 输出延迟功能

RIIC 有 SDA 输出延迟功能。能通过 SDA 输出延迟功能，使全部的 SDA 输出时序（发行（开始 / 重新开始 / 停止）、数据输出和 ACK/NACK 输出）延迟。

SDA 输出延迟功能是从检测到 SCL 时钟的下降沿开始延迟 SDA 的输出，并且通过在 SCL 时钟的 Low 电平期间确保 SDA 的输出，防止通信设备的误动作，也能用于满足 SMBus 数据保持时间（300ns（min））的规格。

SDA 输出延迟功能在 ICMR2.SDDL[2:0] 位不为“000b”时有效，在 SDDL[2:0] 位为“000b”时无效。

在 SDA 输出延迟功能有效（SDDL[2:0] 位不为“000b”）时，SDA 输出延迟计数器将 ICMR2.DLCS 位选择的内部基准时钟（IIC ϕ ）或者内部基准时钟的 2 分频时钟（IIC ϕ /2）作为计数源进行 SDDL[2:0] 位设定周期的计数，RIIC 在结束延迟周期的计数时进行 SDA 输出（发行（开始 / 重新开始 / 停止）、数据输出和 ACK/NACK 输出）。

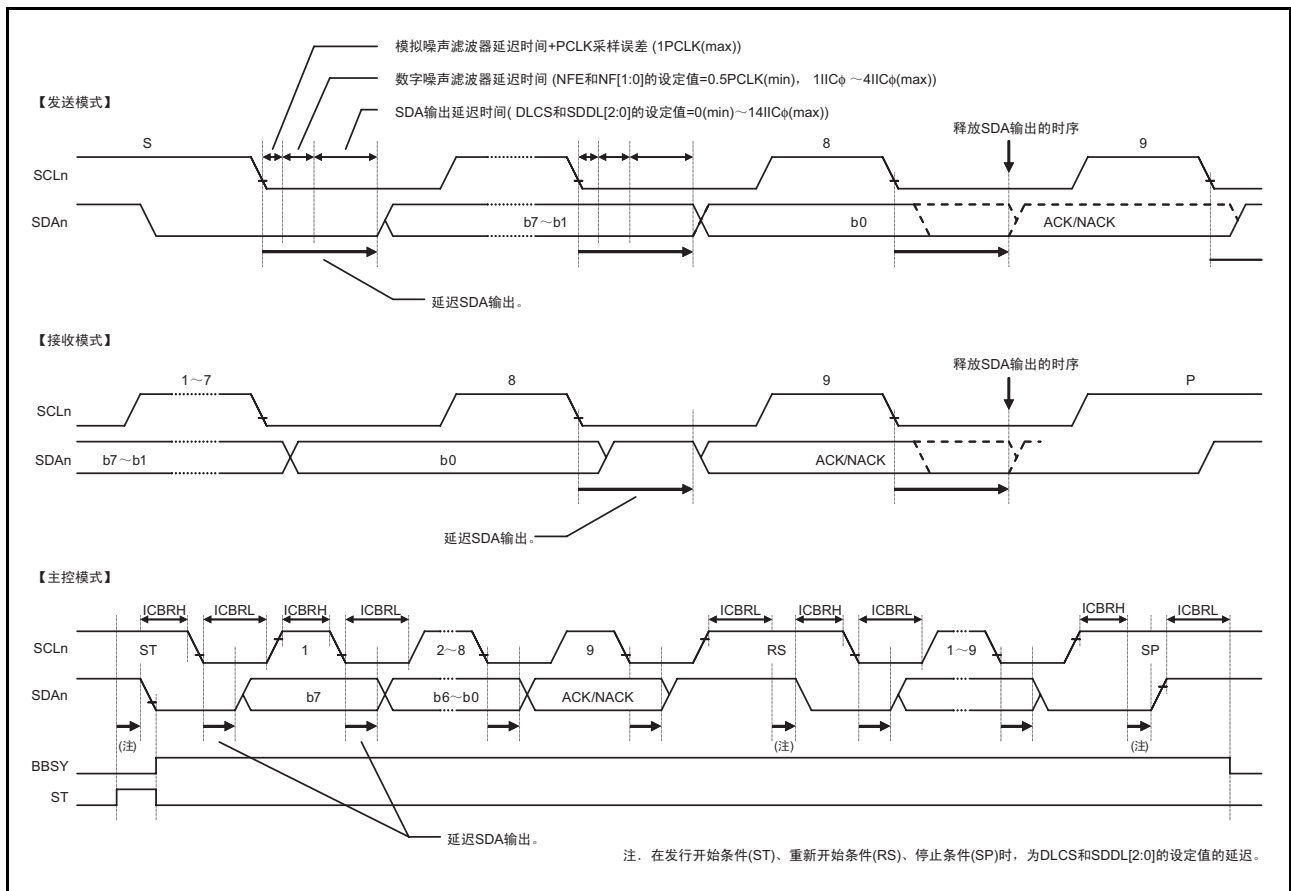


图 22.21 SDA 输出延迟时序

22.6 数字噪声滤波器电路

SCLn 引脚和 SDA_n 引脚的状态经由模拟噪声滤波器电路和数字噪声滤波器电路进入内部。数字噪声滤波器电路的框图如图 22.22 所示。

RIIC 的内部数字噪声滤波器电路由 4 段串联的触发电路和匹配检测电路构成。

通过 ICMR3.NF[1:0] 位选择数字噪声滤波器的有效段数，根据选择的有效段数，噪声消除能力为 1 ~ 4 个 PCLK 周期。

在 PCLK 的下降沿对 SCLn 引脚的输入信号（或者 SDA_n 引脚的输入信号）进行采样，如果 ICMR3.NF[1:0] 位设定的有效段数的触发电路输出全部匹配，就将该电平作为内部信号进行传输，否则就保持原来的值。

在内部运行时钟（PCLK）与通信速度的比小的情况下（如：PCLK 为 4MHz 时的 400kbps 的通信），根据数字噪声滤波器的特性，有可能在产生噪声时需要的信号也被消除。在此情况下，能禁止（ICFER.NFE 位 = 0）使用数字噪声滤波器电路而只使用模拟噪声滤波器电路。

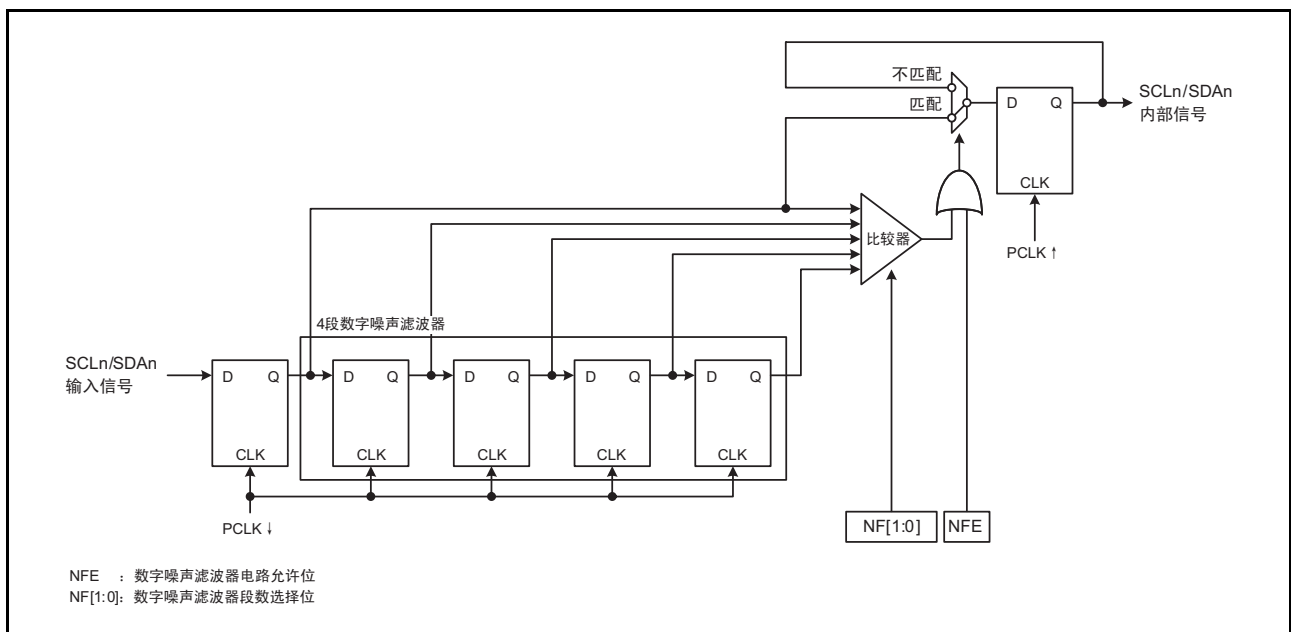


图 22.22 数字噪声滤波器电路的框图

22.7 地址匹配检测功能

RIIC 能设定全呼地址和主机地址以外的 3 种从属地址，从属地址能设定 7 位地址或者 10 位地址。

22.7.1 从属地址匹配检测功能

RIIC 能设定 3 种从属地址，有分别对应的从属地址检测功能。当 ICSER.SARyE 位 (y=0 ~ 2) 为 “1” 时，能检测到 SARUy/SARLy 寄存器 (y=0 ~ 2) 设定的从属地址。

如果 RIIC 检测到设定的从属地址匹配，就在 SCL 时钟的第 9 个时钟的下降沿将对应的 ICSR1.AASy 标志 (y=0 ~ 2) 置 “1”，然后根据后续的 R/W# 位将 ICSR2.RDRF 标志或者 ICSR2.TDRE 标志置 “1”。因此，能产生接收数据满中断 (ICRXI) 或者发送数据空中断 (ICTXI)，并且能通过 AASy 标志判断指定了哪个从属地址。

AASn 标志变为 “1” 的时序如图 22.23 ~ 图 22.25 所示。

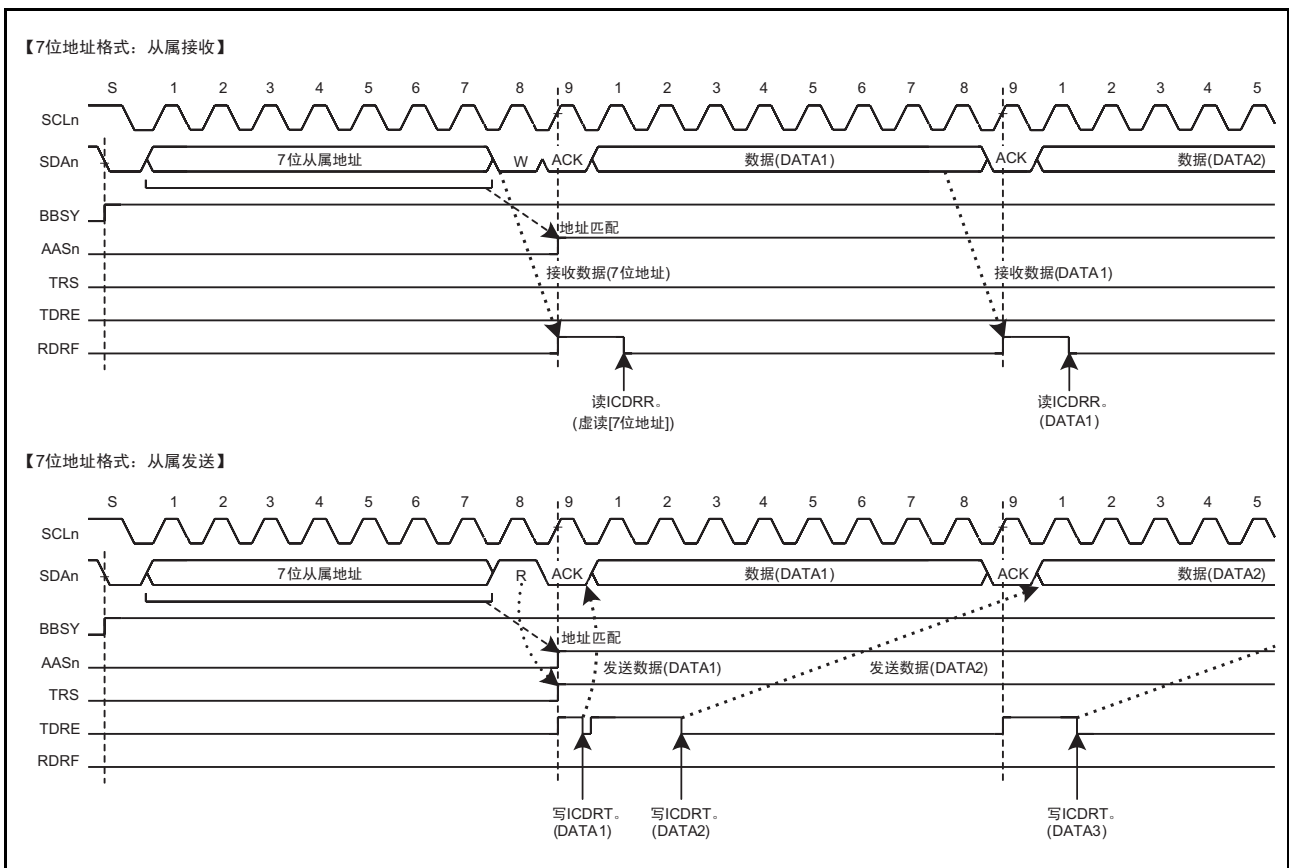


图 22.23 选择 7 位地址格式时的 AASy 标志变为 “1” 的时序

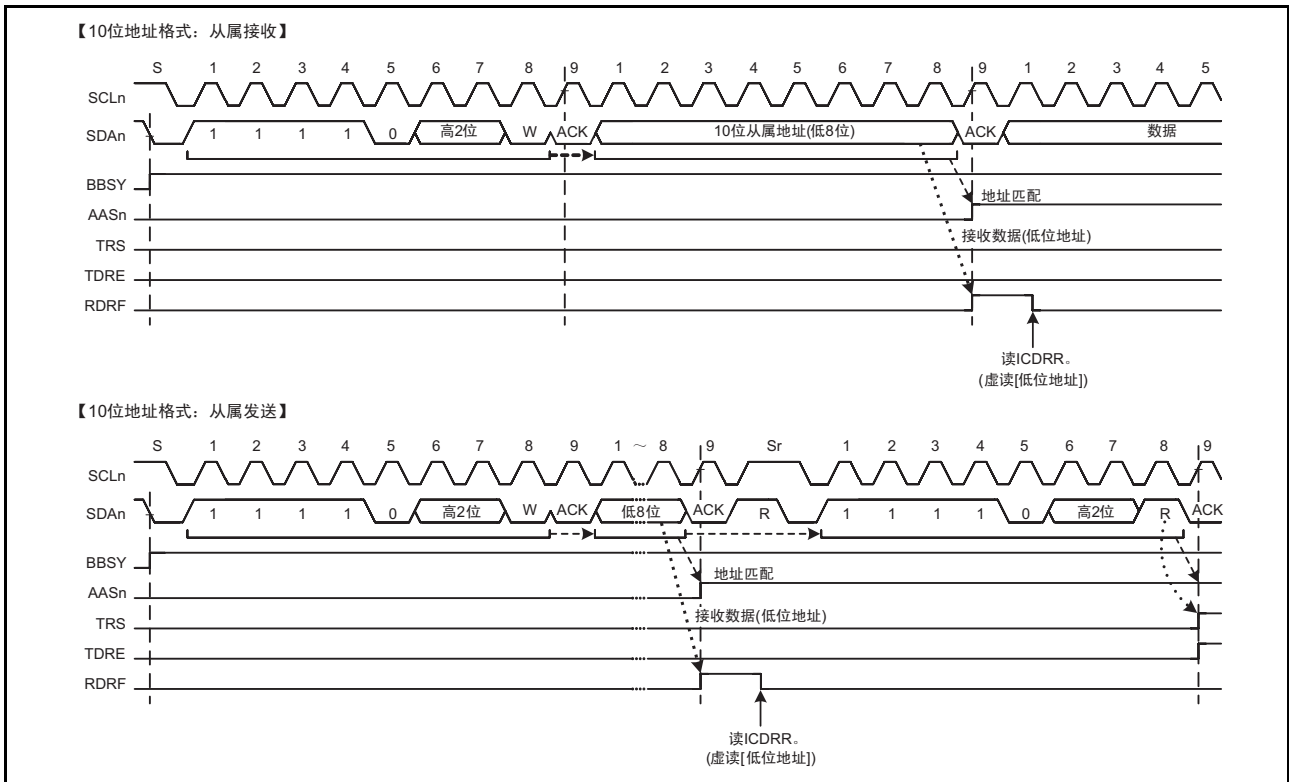


图 22.24 选择 10 位地址格式时的 AASy 标志变为“1”的时序

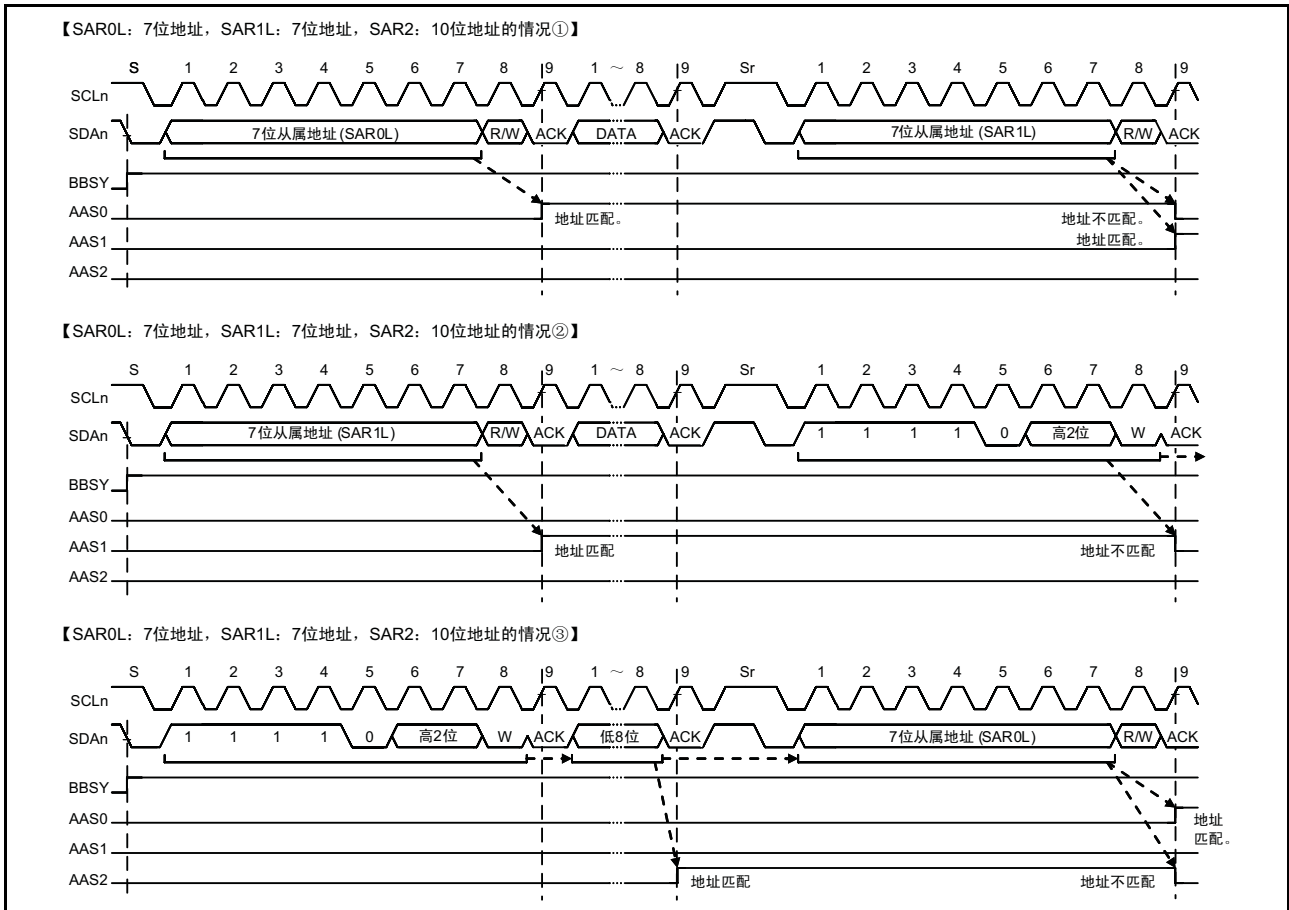


图 22.25 7 位 /10 位地址格式同时存在时的 AASy 标志变为“1” / “0”的时序

22.7.2 全呼地址检测功能

RIIC 具有全呼地址 (0000 000b+0[W]) 检测功能。当 ICSER.GCAE 位为 “1” 时，能检测全呼地址。

如果开始条件或者重新开始条件后的地址为 0000 000b+1[R] (开始字节)，RIIC 就将此地址视为 All “0” 的从属地址而不视为全呼地址。

如果 RIIC 检测到全呼地址，就在 SCL 时钟的第 9 个时钟的下降沿将 ICSR1.GCA 标志置 “1”，同时将 ICSR2.RDRF 标志置 “1”。因此，能产生接收数据满中断 (ICRXI)，并且能通过 GCA 标志判断是否发送了全呼地址。

全呼地址检测后的运行和普通的从属接收运行相同。

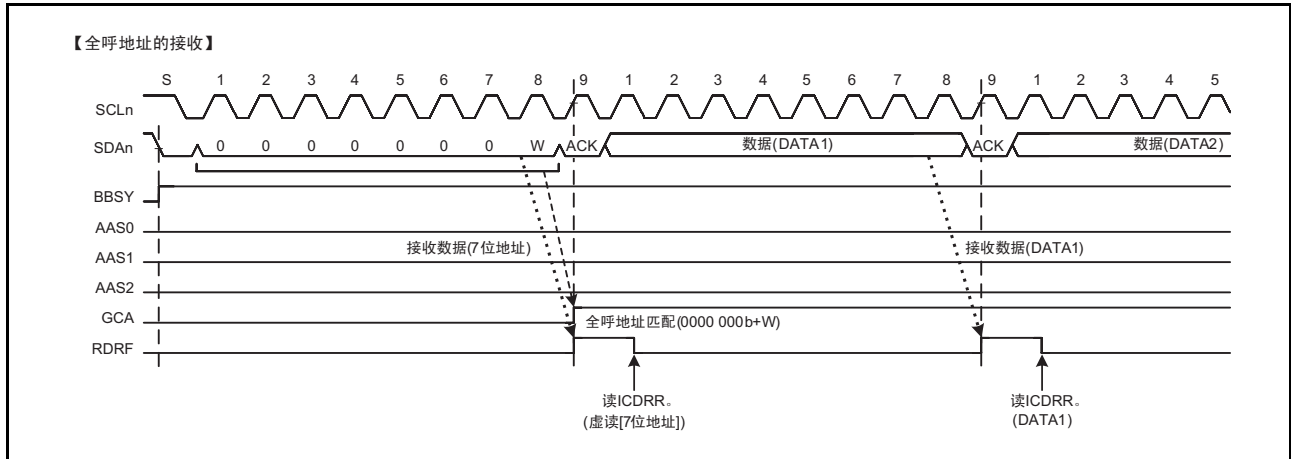


图 22.26 接收全呼地址时的 GCA 标志变为 “1” 的时序

22.7.3 设备 ID 地址检测功能

RIIC 具有以 I²C 总线 (Rev.03) 为基准的设备 ID 地址检测功能。在将 ICSER.DIDE 位置 “1” 的状态下，如果开始条件或者重新开始条件之后的第 1 个字节接收到 “1111 100b”，RIIC 就将此地址视为设备 ID 地址，当后续的 R/W# 位为 “0” 时，就在 SCL 时钟的第 9 个时钟的上升沿将 ICSR1.DID 标志置 “1”，然后将第 2 个字节以后的地址和自己的从属地址进行比较。如果第 2 个字节以后的地址与从属地址寄存器的值匹配，对应的 ICSR1.AASy 标志 (y=0 ~ 2) 就变为 “1”。

此后，如果开始条件或者重新开始条件之后的第 1 个字节再次与设备 ID 地址 (1111 100b) 匹配并且后续的 R/W# 位为 “1”，RIIC 就不比较第 2 个字节以后的地址，而将 ICSR2.TDRE 标志置 “1”。

设备 ID 地址检测功能，在与自己的从属地址不匹配时或者自己的从属地址匹配后的重新开始条件后面的地址与设备 ID 地址不匹配时，将 DID 标志置 “0”。在开始条件或者重新开始条件之后的第 1 个字节与设备 ID 地址 (1111 100b) 匹配并且 R/W# 位为 “0” 时，将 DID 标志置 “1”，并且将第 2 个字节以后的地址和从属地址进行比较。当 R/W# 位为 “1” 时，DID 标志继续保持原来的值，并且不比较第 2 个字节以后的从属地址。因此，通过在确认 TDRE 为 “1” 后检查 DID 标志，能确认接收到设备 ID。

在接收一连串的设备 ID 后，要发送给主机的设备 ID 字段所需的信息 (3 个字节：厂商 [12 位]+ 部件识别 [9 位]+ 版本 [3 位]) 和通常的发送数据一样，必须预先准备。有关设备 ID 字段所需信息的详细内容，请向 NXP 公司询问。

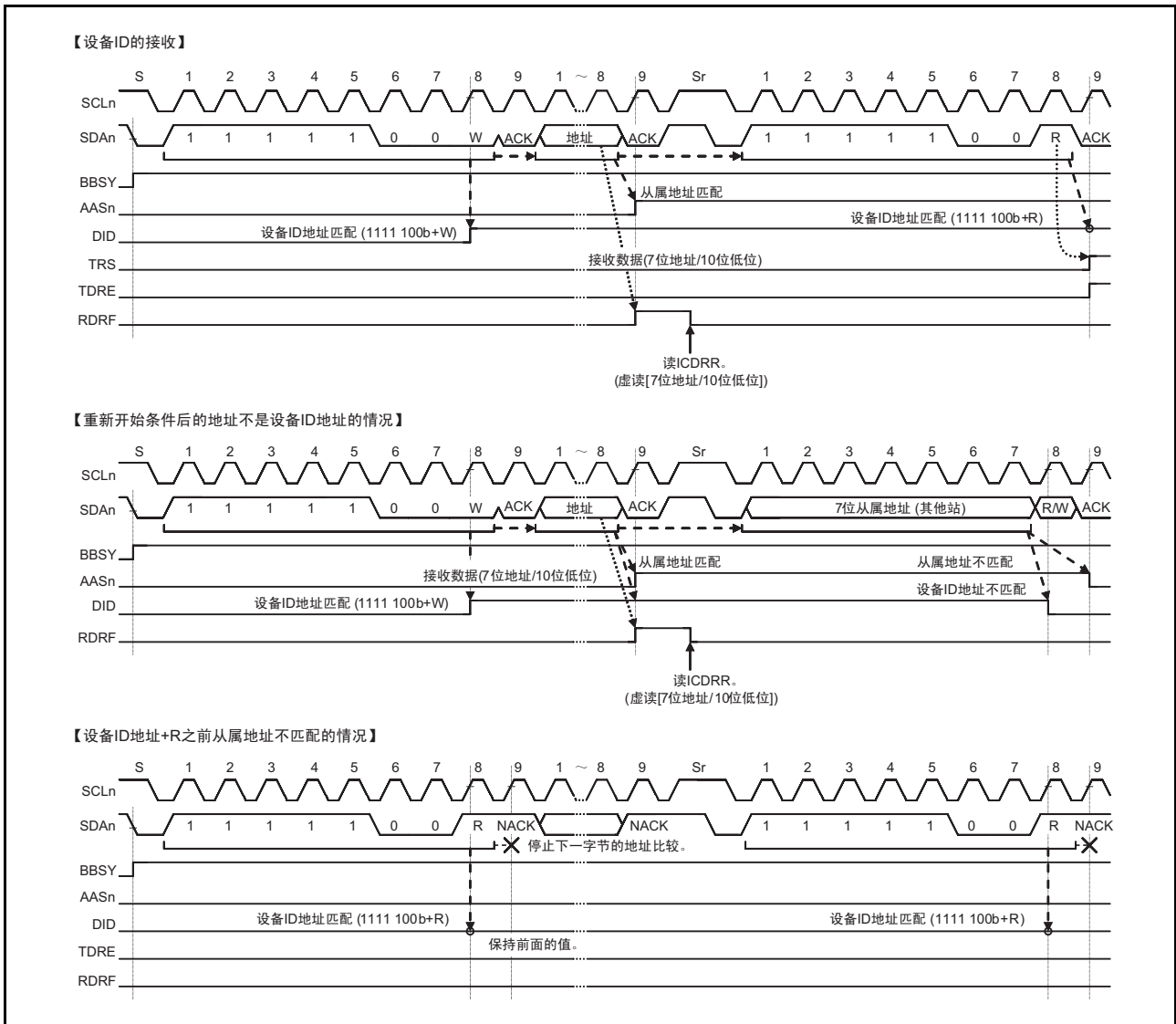


图 22.27 接收设备 ID 地址时的 AASn 标志和 DID 标志的置位 / 清除时序

22.7.4 主机地址检测功能

RIIC 具有 SMBus 运行时的主机地址检测功能。如果在 ICMR3.SMBS 位为 “1” 时将 IC SER.HOAE 位置 “1”，就能在从属接收模式 (ICCR2.MST 位和 ICCR2.TRS 位为 “00b”) 中检测主机地址 (0001 000b)。

如果 RIIC 检测到主机地址，就在 SCL 时钟的第 9 个时钟的下降沿将 ICSR1.HOA 标志置 “1”，并且在 Wr 位的情况下 (R/W# 位接收到 “0”)，将 ICSR2.TDRE 标志置 “1”。因此，能产生发送数据空中断 (ICTXI)，并且能通过 HOA 标志，确认从智能电池等发送了主机地址。

即使接在主机地址 (0001 000b) 后面的位是 Rd 位 (接收到为 “1” 的 R/W# 位)，也能检测主机地址。主机地址检测后的运行和普通的从属模式运行相同。

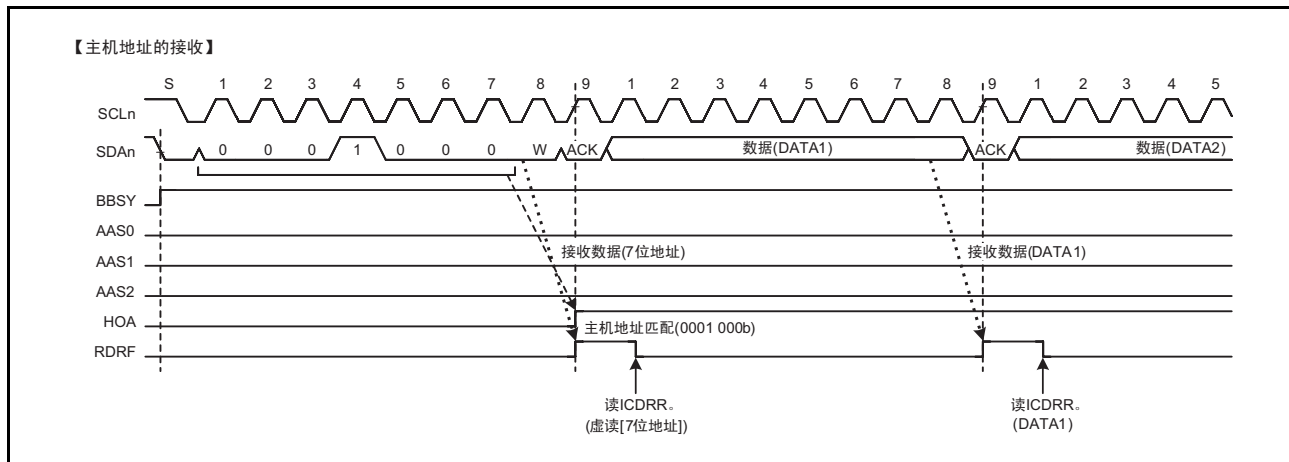


图 22.28 接收主机地址时的 HOA 标志变为 “1” 的时序

22.8 SCL 的 Low 电平自动保持功能

22.8.1 发送数据的误发送防止功能

在发送模式中 (ICCR2.TRS 位 = 1)，如果移位寄存器 (ICDRS 寄存器) 为空状态并且未写发送数据 (ICDRT 寄存器)，RIIC 就在以下所示的区间自动保持 SCLn 线的 Low 电平。Low 电平的保持延长了写发送数据前的 Low 电平区间，防止意想不到的数据误发送。

《主控发送模式》

- 发行开始条件或者重新开始条件后的 Low 电平区间
- 第 9 个时钟和下次传送的第 1 个时钟的 Low 电平区间

《从属发送模式》

- 第 9 个时钟和下次传送的第 1 个时钟的 Low 电平区间

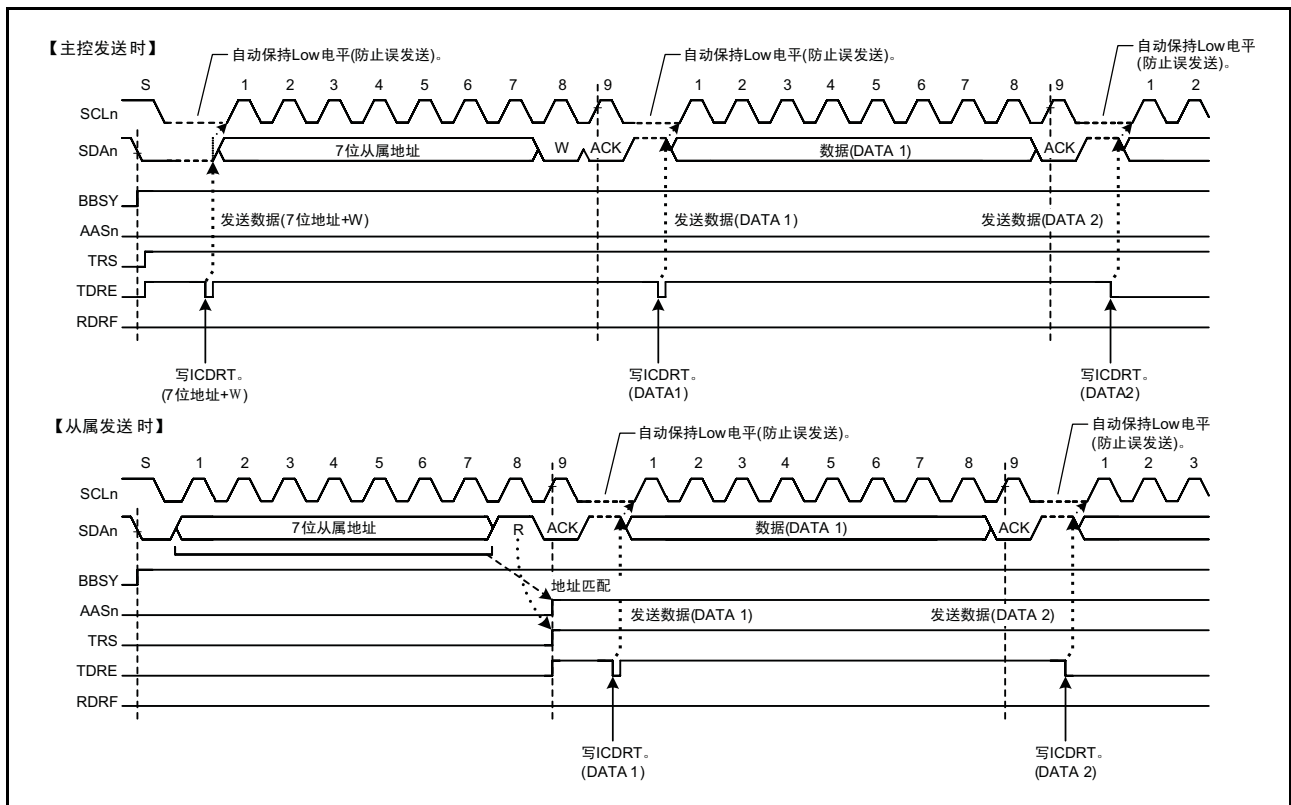


图 22.29 发送模式的 Low 电平自动保持运行

22.8.2 NACK 接收传送中止功能

RIIC 具有在发送模式中 (ICCR2.TRS 位 =1) 接收到 NACK 时中止传送的功能。此功能在 ICFER.NACKE 位为 “1” (允许中止传送) 时有效, 如果在接收 NACK 时已经写了下一个发送数据 (ICSR2.TDRE 标志为 “0” 的状态), 就在 SCL 时钟的第 9 个时钟下降时自动中止下一个数据的发送。因此, 能在下一个发送数据的 MSB 为 “0” 时防止 SDA_n 线被固定为 Low 电平输出。

如果通过 NACK 接收传送中止功能来中止传送 (ICSR2.NACKF 标志为 “1”), 就不进行以后的发送和接收。要重新开始传送时, 必须将 NACKF 标志置 “0”。在主控发送模式中, 必须将 NACKF 标志置 “0”, 然后在发行重新开始条件或者停止条件后发行开始条件, 重新开始传送。

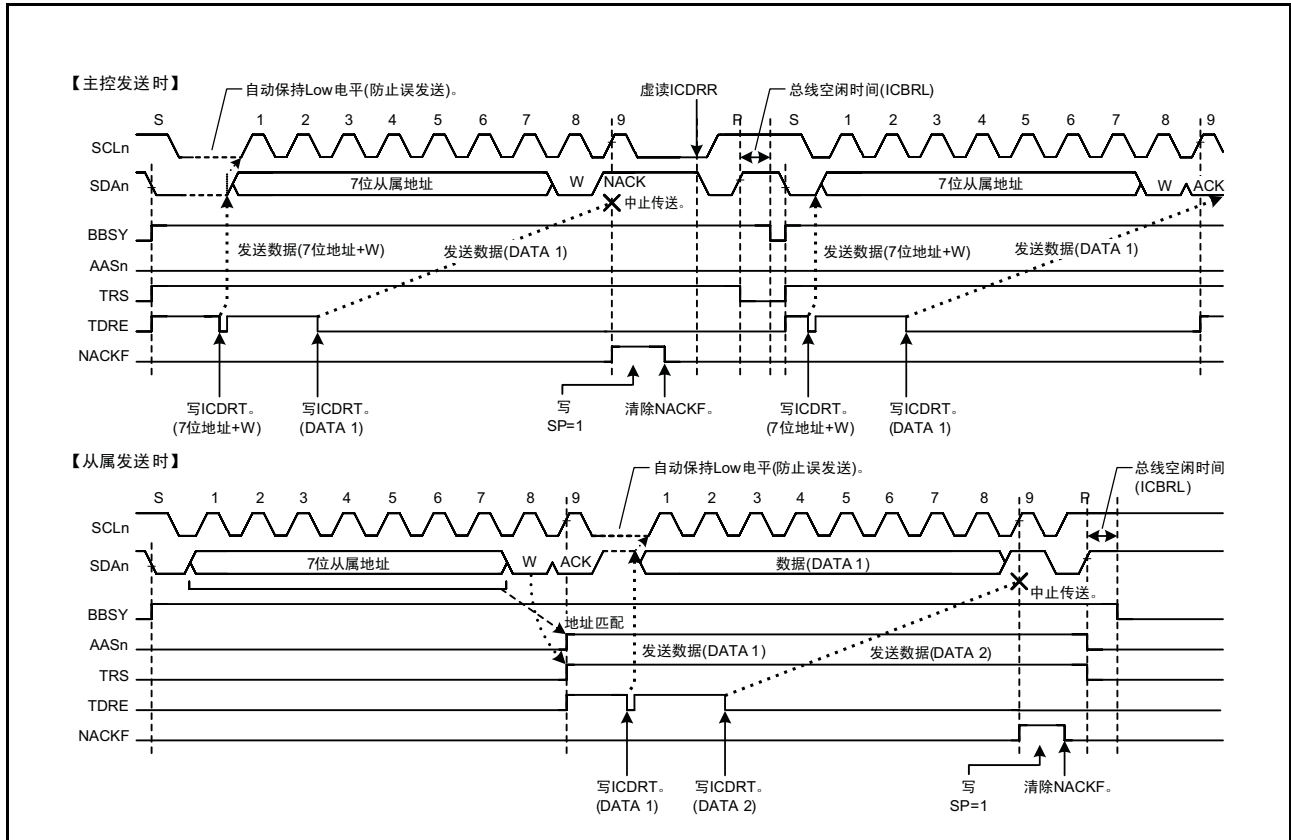


图 22.30 接收 NACK 时的传送中止 (NACKE=1)

22.8.3 接收数据漏取防止功能

如果在接收模式中 (ICCR2.TRS 位 =0) 并且接收数据满 (ICSR2.RDRF 标志 =1) 的状态下因至少推迟 1 个传送帧的读接收数据 (ICDRR 寄存器) 等而发生响应处理延迟, RIIC 就在开始下一个数据接收前自动保持 SCL_n 线的 Low 电平, 防止漏取接收数据。

在最后接收数据的读处理延迟期间, 即使在停止条件后指定了自己的从属地址, 通过自动保持 Low 电平防止漏取接收数据的功能仍然有效。因为在停止条件后自己的从属地址不匹配时不保持 Low 电平, 所以不妨碍其他通信。

RIIC 能通过 ICMR3.WAIT 位和 ICMR3.RDRFS 位的组合选择保持 Low 电平的区间。

(1) 通过 WAIT 位进行的 1 字节接收以及 Low 电平自动保持功能

如果将 ICMR3.WAIT 位置“1”，RIIC 就通过 WAIT 位的功能进行 1 字节的接收。当 ICMR3.RDRFS 位为“0”时，RIIC 就自动给从 SCL 时钟的第 8 个时钟的下降沿到第 9 个时钟的下降沿期间的应答位发送 ICMR3.ACKBT 位的内容，如果检测到第 9 个时钟的下降沿，就通过 WAIT 位的功能自动保持 SCLn 线的 Low 电平。通过读 ICDRR 寄存器来解除保持的 Low 电平。因此，能逐字节地进行接收。

在主控接收模式或者从属接收模式中，WAIT 位的功能从与自己的从属地址（包括全呼地址和主机地址）匹配后的接收帧开始有效。

(2) 通过 RDRFS 位进行的 1 字节接收（控制 ACK/NACK 发送）以及 Low 电平的自动保持功能

如果将 ICMR3.RDRFS 位置“1”，RIIC 就通过 RDRFS 位的功能进行 1 字节的接收。如果将 RDRFS 位置“1”，就将接收数据满标志（ICSR2.RDRF 标志）变为“1”的时序更改为 SCL 时钟的第 8 个时钟的上升。如果检测到第 8 个时钟的下降沿，就自动保持 SCLn 线的 Low 电平。通过写 ICMR3.ACKBT 位来解除保持的 Low 电平，而在读 ICDRR 寄存器时无法解除。因此，能逐字节地根据接收数据的内容进行 ACK/NACK 发送的接收运行。

在主控接收模式或者从属接收模式中，RDRFS 位的功能从与自己的从属地址（包括全呼地址和主机地址）匹配后的接收帧开始有效。

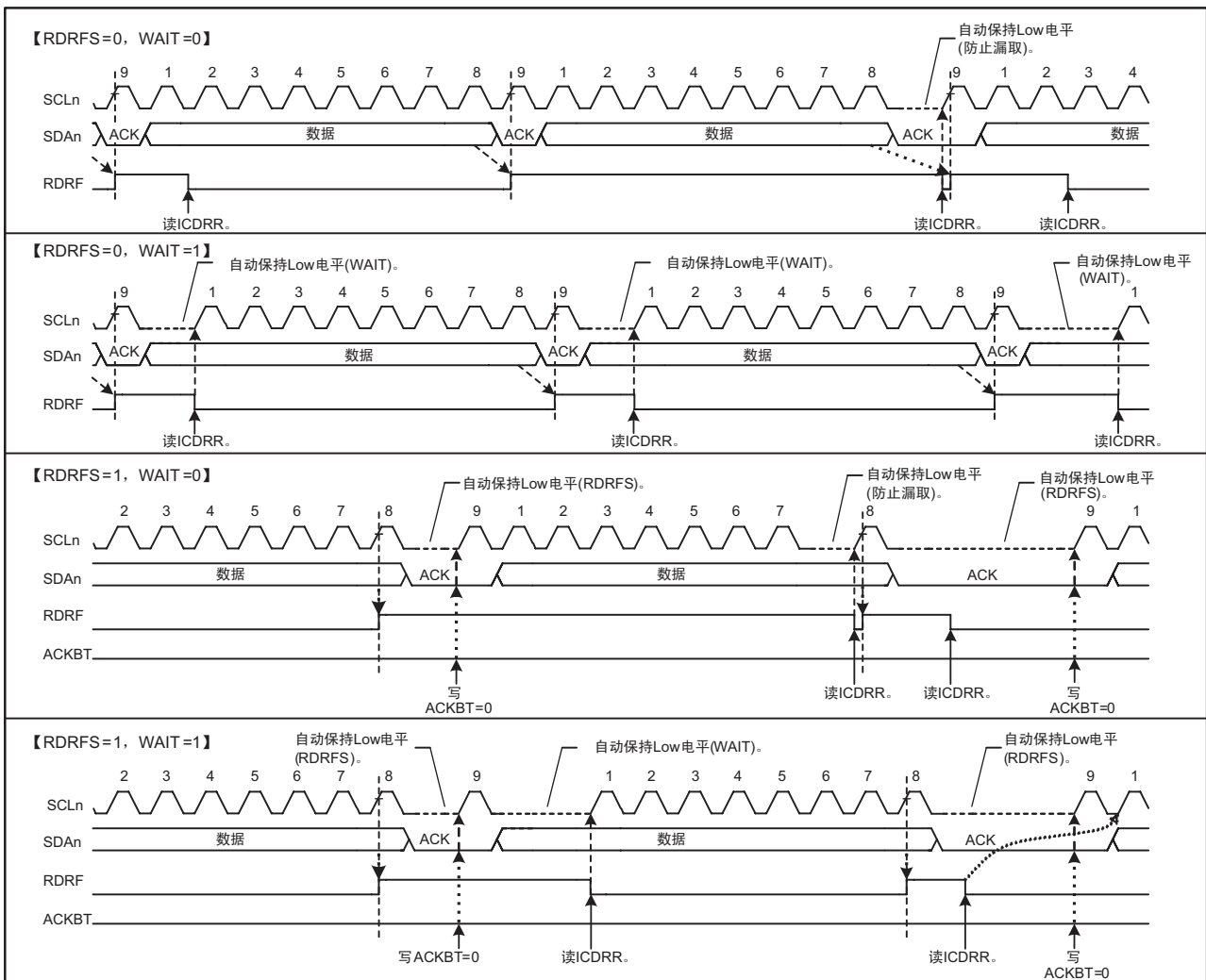


图 22.31 接收模式的 Low 电平自动保持（RDRFS 位和 WAIT 位）

22.9 仲裁失败检测功能

RIIC 除了 I²C 总线规格规定的普通仲裁失败检测功能以外，还具有防止双重发行开始条件功能、NACK 发送时的仲裁失败检测功能以及从属发送时的仲裁失败检测功能。

22.9.1 主控仲裁失败检测功能 (MALE 位)

在发行开始条件时，RIIC 将 SDA_n 线置为 Low 电平。但是，如果其他主控器件提前发行开始条件后将 SDA_n 线置为 Low 电平，就产生仲裁失败并且优先其他主控器件的通信。同样地，如果在 ICCR2.BBSY 标志为“1”（总线忙）时将 ICCR2.ST 位置“1”，就产生仲裁失败并且优先其他主控器件的通信。此时，不生成开始条件。

在正常地发行开始条件时，如果包括发送地址在内的发送数据（SDA 信号）与 SDA_n 线上的信号状态不同（自身发送的 SDA 输出为 High 电平输出（SDA_n 引脚为高阻抗状态）而检测到 SDA 线为 Low 电平时），就产生仲裁失败。

如果发生主控仲裁失败，RIIC 就立即转移到从属接收模式。此时，如果包括全呼地址在内的从属地址匹配，就继续从属模式的运行。

对于主控仲裁失败检测，在 ICFER.MALE 位为“1”（允许主控仲裁失败检测）的状态下以下条件成立时，检测到仲裁失败。

[主控仲裁失败条件]

- 在 ICCR2.BBSY 标志为“0”的状态下通过将 ICCR2.ST 位置“1”来发行开始条件时，SDA 信号与 SDA_n 线上的信号状态不同（开始条件发行错误）时
- 在 ICCR2.BBSY 标志为“1”的状态下将 ICCR2.ST 位置“1”（开始条件双重发行错误）时
- 在发送模式中（ICCR2.MST 位和 ICCR2.TRS 位为“11b”），应答除外的发送数据（SDA 信号）与 SDA_n 线上的信号状态不同时

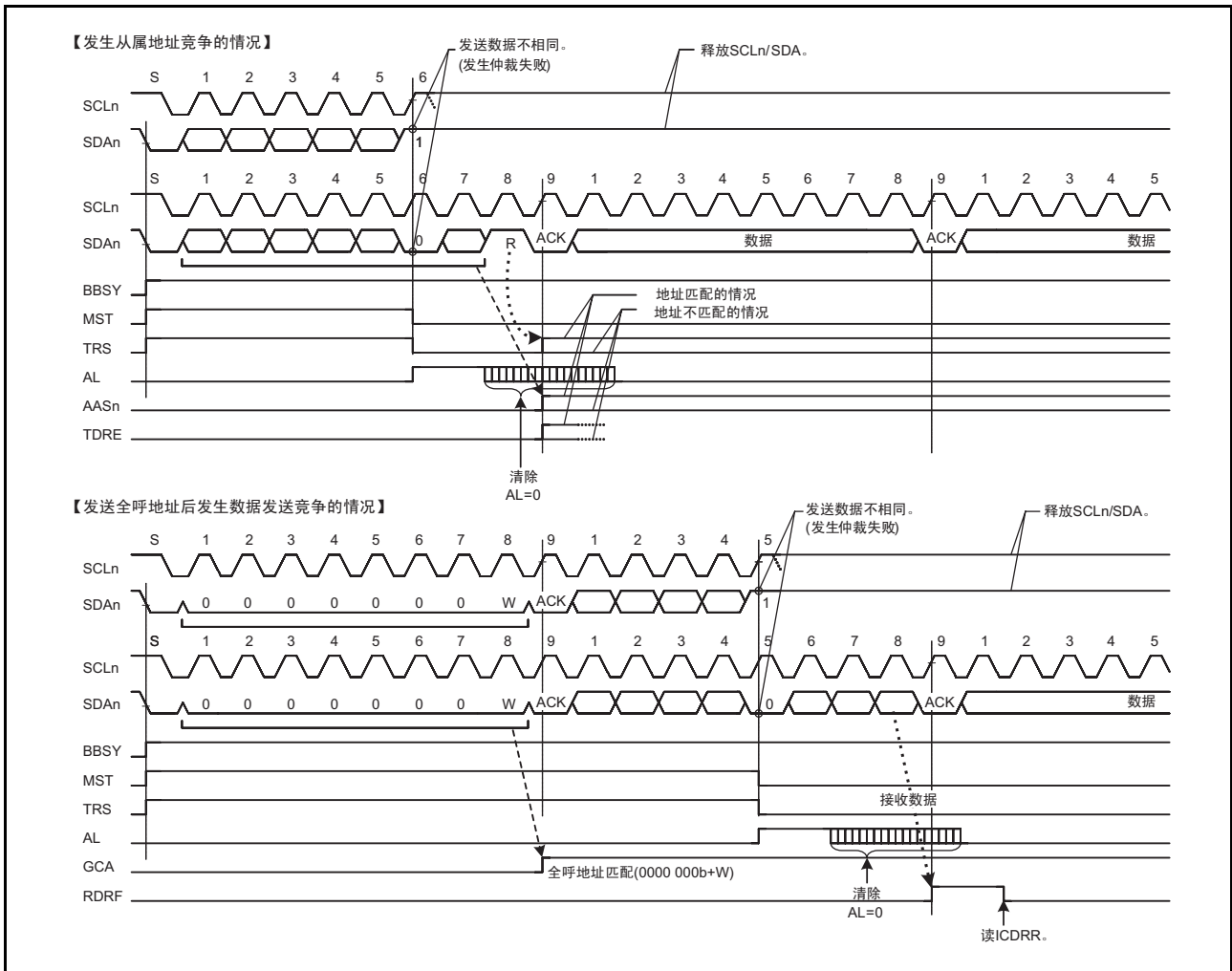


图 22.32 主控仲裁失败检测的运行例子 (MALE=1 的情况)

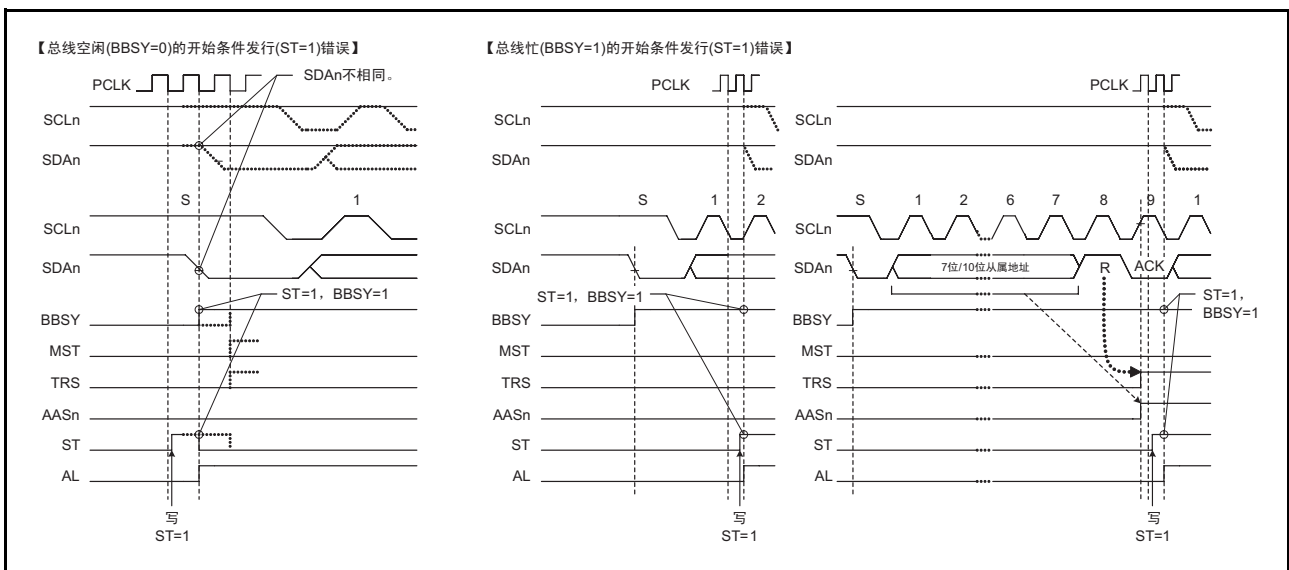


图 22.33 发行开始条件时的仲裁失败 (MALE=1 的情况)

22.9.2 NACK 发送仲裁失败检测功能 (NALE 位)

在接收模式中，如果在发送 NACK 时 RIIC 自身发送的 SDA 信号与 SDA 线上的信号状态不同（自身发送的 SDA 输出为 High 电平输出（SDAn 引脚为高阻抗状态）而检测到 SDA 线为 Low 电平时），就产生仲裁失败。主要在多主控系统中 2 个或者 2 个以上的主控器件同时从同一个从属器件接收数据时发生 NACK 发送和 ACK 发送的冲突。这种情况发生在 2 个或者 2 个以上的主控器件通过 1 个从属器件交换共同信息时。NACK 发送仲裁失败检测的运行例子如图 22.34 所示。

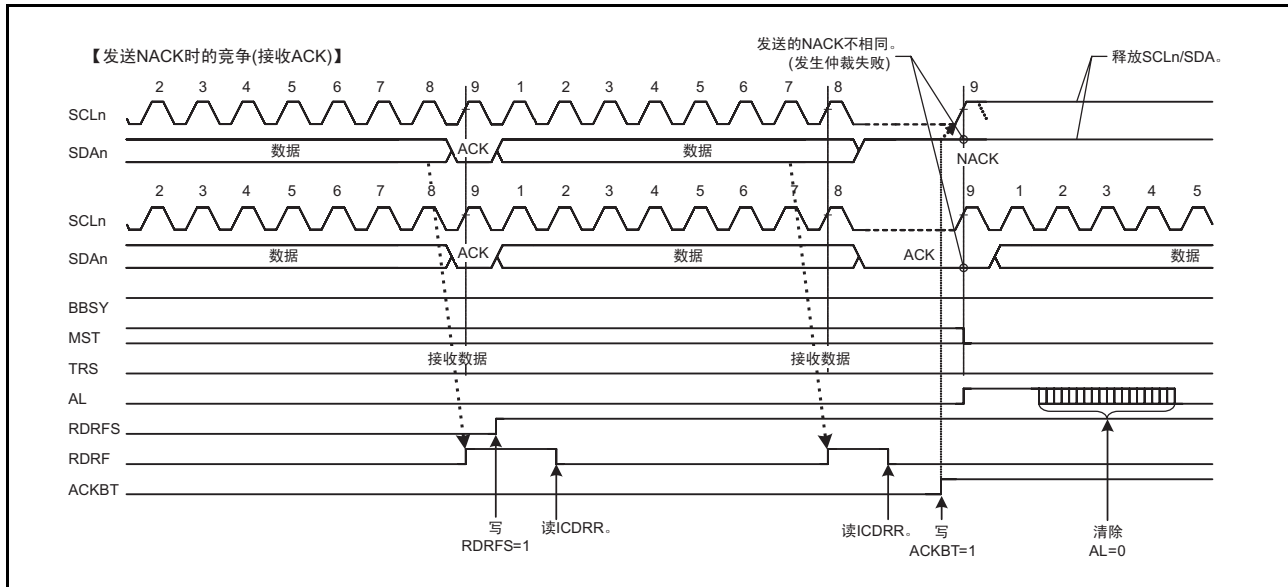


图 22.34 NACK 发送仲裁失败检测的运行例子 (NALE=1 的情况)

以 2 个主控器件（主控 A 和主控 B）和 1 个从属器件连接总线的情况为例进行说明。假设主控 A 从从属器件接收 2 字节的数据，主控 B 从从属器件接收 4 字节的数据。

在主控 A 和主控 B 同时存取从属器件的情况下，因为从属地址相同，所以主控 A 和主控 B 在存取从属器件时不发生仲裁失败，因此主控 A 和主控 B 都取得了总线权。然后，主控 A 在从从属器件接收完最后 2 字节数据时发送 NACK，主控 B 因为没有从属器件接收到需要的 4 字节而发送 ACK，因此主控 A 发送的 NACK 和主控 B 发送的 ACK 发生了冲突。一般情况下，如果发生这种状况，因为主控 A 在无法检测到主控 B 发送的 ACK 的状态下发行停止条件，所以与主控 B 的 SCL 时钟输出发生竞争并且阻碍通信。

RIIC 对于此类发送 NACK 时接收 ACK 的情况，能检测到和其他主控器件的竞争失败，产生仲裁失败。

如果发生 NACK 发送仲裁失败，RIIC 就立即解除从属匹配状态并且转移到从属接收模式。因此，能将停止条件的发行防范于未然并且防止总线的通信故障。

在 SMBus 的 ARP 命令处理中，能省去在与指定地址的 UDID（唯一设备标识）不匹配时的 NACK 发送后以及确定指定地址后的 Get UDID（通用）的 NACK 发送后的剩余处理（FFh 发送处理）。

对于 NACK 发送仲裁失败检测，在 ICFER.NALE 位为“1”（允许 NACK 发送仲裁失败检测）的状态下以下条件成立时，检测到仲裁失败。

[NACK 发送仲裁失败条件]

- 在发送 NACK 时（ICMR3.ACKBT 位=1），自身发送的 SDA 信号与 SDA_n 线上的信号状态不同（接收到 ACK）。

22.9.3 从属仲裁失败检测功能 (SALE 位)

如果在从属发送时发送数据 (自身发送的 SDA 信号) 与 SDA 线上的信号状态不同 (自身发送的 SDA 输出为 High 电平输出 (SDAn 引脚为高阻抗状态) 而检测到 SDA 线为 Low 电平), RIIC 就产生仲裁失败。此仲裁失败功能主要用于 SMBus 的 UDID (唯一设备标识符) 的发送。

如果发生从属仲裁失败, RIIC 就立即解除从属匹配状态并且转移到从属接收模式。

通过使用此功能, 能省去发送 SMBus 的 UDID 时的数据冲突检测和数据冲突后的剩余处理 (FFh 发送处理)。

对于从属仲裁失败检测, 在 ICFER.SALE 位为 “1” (允许从属仲裁失败检测) 的状态下以下条件成立时, 检测到仲裁失败。

[从属仲裁失败条件]

- 在从属发送模式中 (ICCR2.MST 位和 ICCR2.TRS 位为 “01b”), 应答除外的发送数据 (自身发送的 SDA 信号) 与 SDA 线上的信号状态不同。

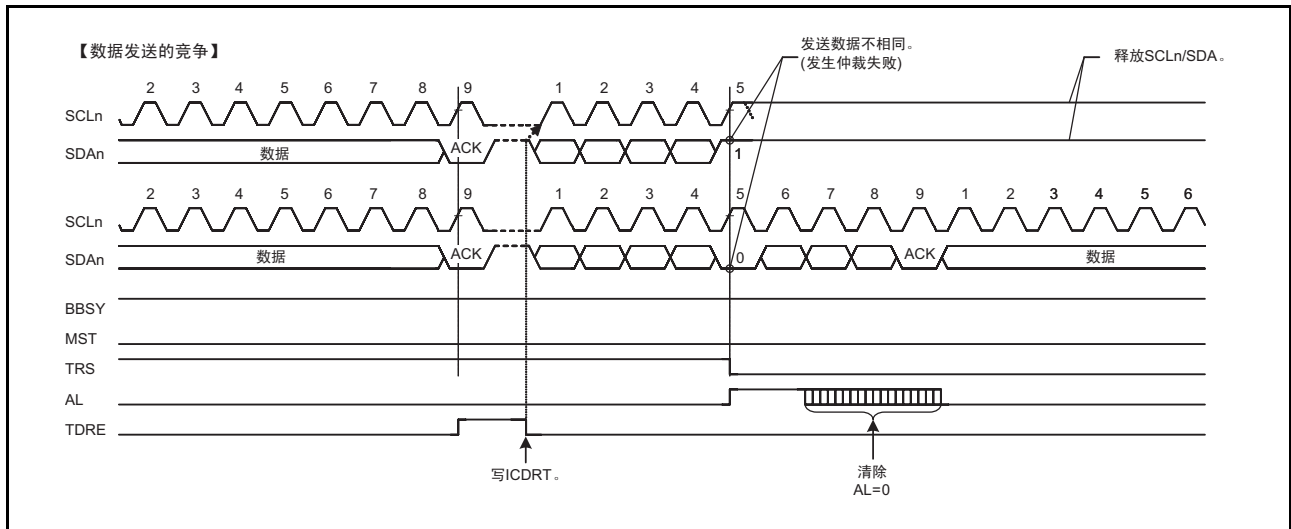


图 22.35 从属仲裁失败检测的运行例子 (SALE=1 的情况)

22.10 开始条件、重新开始条件和停止条件的发行功能

22.10.1 开始条件的发行

RIIC 通过 ICCR2.ST 位发行开始条件。

如果将 ST 位置 “1”，就请求发行开始条件，在 ICCR2.BBSY 标志为 “0”（总线空闲）的状态下发行开始条件。如果正常地发行开始条件，RIIC 就自动转移到主控发送模式。

开始条件的发行步骤如下：

[开始条件的发行]

1. 使SDAn线下降（从High电平变为Low电平）。
2. 确保ICBRH寄存器所设时间的开始条件的保持时间。
3. 使SCLn线下降（从High电平变为Low电平）。
4. 在检测到SCLn线的Low电平后，确保ICBRL寄存器所设时间的SCLn线的Low电平宽度。

22.10.2 重新开始条件的发行

RIIC 通过 ICCR2.RS 位发行重新开始条件。

如果将 RS 位置 “1”，就请求发行重新开始条件，RIIC 在 ICCR2.BBSY 标志为 “1”（总线忙）并且 ICCR2.MST 位为 “1”（主控模式）的状态下发行重新开始条件。

重新开始条件的发行步骤如下：

[重新开始条件的发行]

1. 释放SDAn线。
2. 确保ICBRL寄存器所设时间的SCLn线的Low电平宽度。
3. 释放SCLn线（从Low电平变为High电平）。
4. 在检测到SCLn线的High电平后，确保ICBRL寄存器所设时间的重新开始条件的准备时间。
5. 使SDAn线下降（从High电平变为Low电平）。
6. 确保ICBRH寄存器所设时间的重新开始条件的保持时间。
7. 使SCLn线下降（从High电平变为Low电平）。
8. 在检测到SCL线的Low电平后，确保ICBRL寄存器所设时间的SCLn线的Low电平宽度。

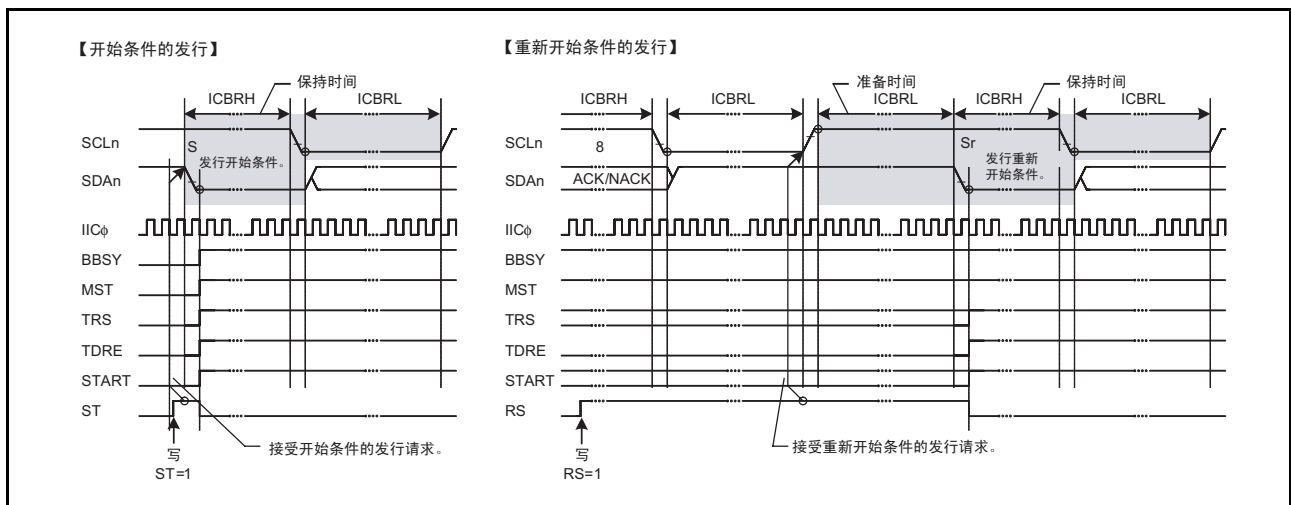


图 22.36 开始条件 / 重新开始条件的发行时序 (ST 位和 RS 位)

22.10.3 停止条件的发行

RIIC 通过 ICCR2.SP 位发行停止条件。

如果将 SP 位置“1”，就请求发行停止条件，RIIC 在 ICCR2.BBSY 标志为“1”（总线忙）并且 ICCR2.MST 位为“1”（主控模式）的状态下发行停止条件。

停止条件的发行步骤如下：

[停止条件的发行]

- 使 SDA_n 线下降（从 High 电平变为 Low 电平）
- 确保 ICBRL 寄存器所设时间的 SCL_n 线的 Low 电平宽度
- 释放 SCL_n 线（从 Low 电平变为 High 电平）
- 在检测到 SCL_n 线的 High 电平后，确保 ICBRH 寄存器所设时间的停止条件的准备时间
- 释放 SDA_n 线（从 Low 电平变为 High 电平）
- 确保 ICBRL 寄存器所设时间的总线空闲时间
- 清除 BBSY 标志（释放总线权）

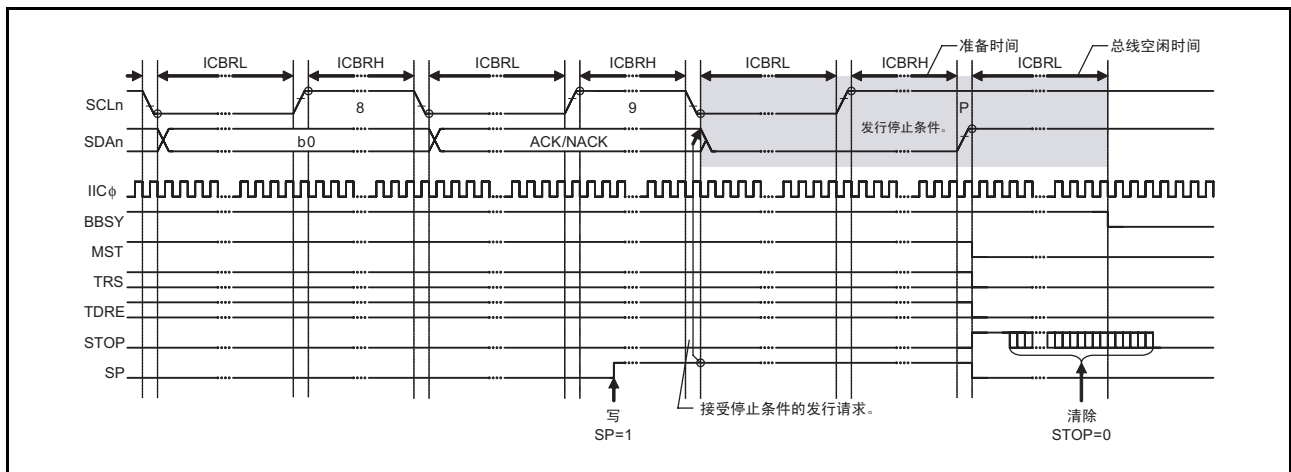


图 22.37 停止条件的发行时序 (SP 位)

22.11 总线意外停机

如果 I²C 总线主要因噪声等的影响而在主控制器件和从属器件之间失去同步，就有可能在 SCL_n 线和 SDA_n 线被固定的状态下发生总线意外停机。

对于总线意外停机的状态，RIIC 具有超时检测功能（能通过监视 SCL_n 线来检测总线意外停机状态）、SCL 时钟的追加输出功能（用于解除因失去同步引起的总线意外停机状态）和 RIIC/ 内部复位功能。

能通过 ICCR1.SCLO 位、ICCR1.SDAO 位、ICCR1.SCLI 位和 ICCR1.SDAI 位，确认是 RIIC 自身给 SCL_n 线 /SDA_n 线输出了 Low 电平还是通信器件输出了 Low 电平。

22.11.1 超时检测功能

RIIC 具有超时检测功能，检测 SCLn 线在一定时间后没有变化的状态。在总线忙的状态下，RIIC 能通过监视到 SCLn 线的 Low 电平或者 High 电平被固定了一定时间以上，并且检测到总线的异常状态。

超时检测功能监视 SCLn 线的状态，通过内部计数器对 Low 电平或者 High 电平的时间进行计数。如果 SCLn 线有变化（上升/下降），就对内部计数器进行复位，否则就继续进行计数。如果在 SCLn 线没有变化的状态下内部计数器发生上溢，RIIC 就能检测到超时并且通知总线的异常状态。

超时检测功能在 ICFER.TMOE 位为“1”时有效，在主控模式中总线忙时（ICCR2.BBSY 标志为“1”的状态），或者在从属模式中 BBSY 标志为“1”并且自己的从属地址匹配时（ICSR1 寄存器 ≠00h），能检测到 SCLn 线被固定为 Low 电平或者 High 电平的总线异常状态。

超时检测功能的内部计数器将 ICMR1.CKS[2:0] 位设定的内部基准时钟（IICφ）作为计数源，在选择长模式时（ICMR2.TMOS 位 =0），用作 16 位计数器；在选择短模式时（TMOS 位 =1），用作 14 位计数器。

对于内部计数器的计数，能通过设定 ICMR2.TMOH 位和 ICMR2.TMOL 位选择是在 SCLn 线的 Low 电平还是在 High 电平的状态下进行计数，或者在 Low 电平和 High 电平的状态下都进行计数器。如果将 ICMR2.TMOH 位和 ICMR2.TMOL 位都置“0”，就不进行内部计数。

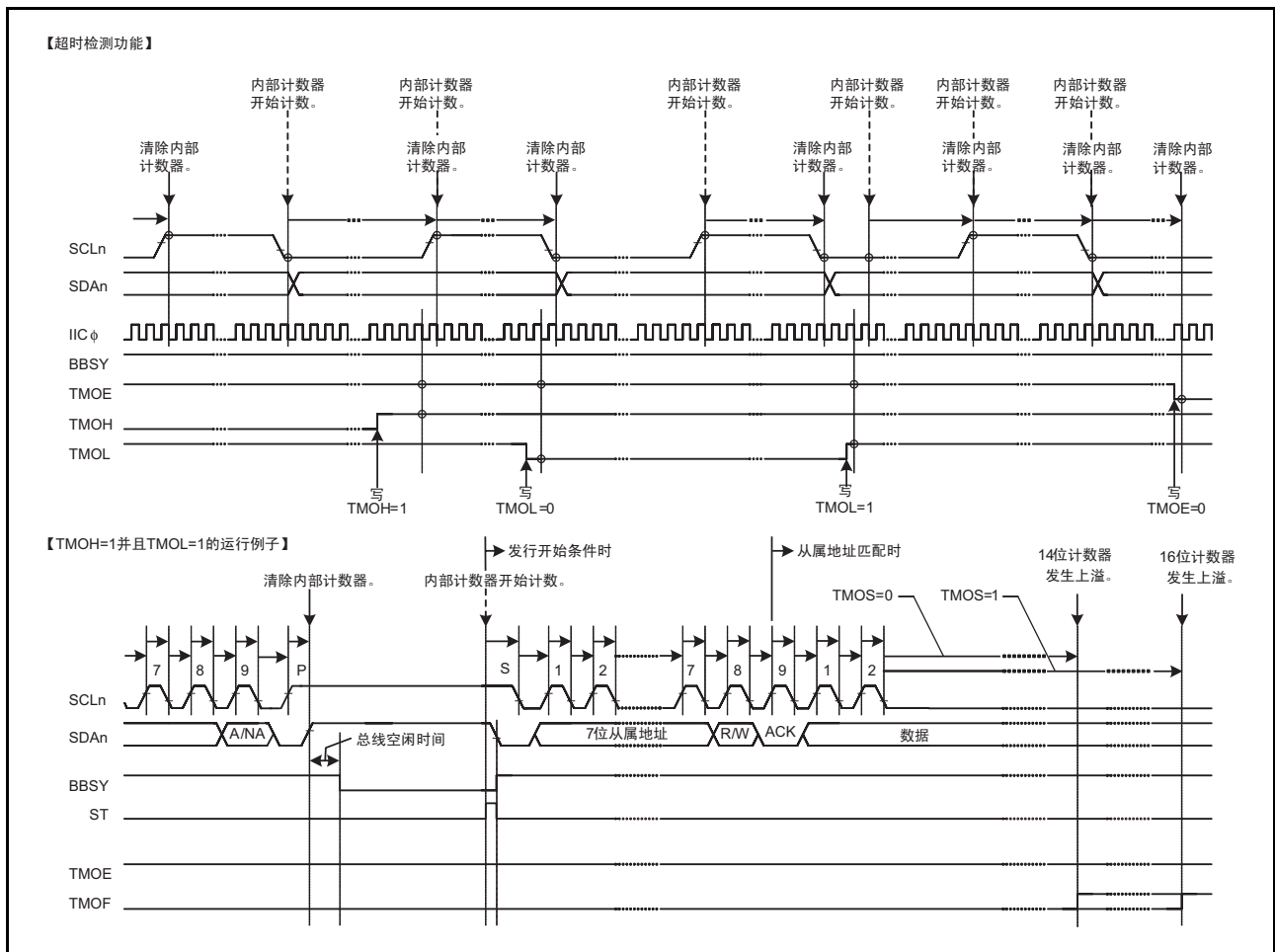


图 22.38 超时检测功能（TMOE 位、TMOS 位、TMOH 位和 TMOL 位）

22.11.2 SCL 时钟追加输出功能

RIIC 具有 SCL 时钟的追加输出功能，在主机模式中用于释放因与从属器件失去同步而使从属器件的 SDA_n 线被固定为 Low 电平的状态。

SCL 时钟追加输出功能是以 1 个时钟为单位追加输出 SCL 时钟的功能，主要用于在主机模式中从属器件在 SDA_n 线固定为 Low 电平的状态而无法发行停止条件时释放从属器件的 SDA_n 线的固定状态。一般情况下，不使用此功能。如果在正常的通信过程中使用此功能，就可能引起通信异常。

对于 SCL 时钟的追加输出，如果将 ICCR1.CLO 位置“1”，就将 ICMR1.CKS[2:0] 位、ICBRH 寄存器和 ICBRL 寄存器所设传送率的 SCL 时钟作为 1 个时钟的追加时钟进行输出。一旦输出完 1 个时钟周期的追加时钟，CLO 位就自动变为“0”。因此，能通过软件在确认 CLO 位为“0”后写“1”，连续输出追加时钟。

在主机模式中，因噪声等影响引起和从属器件失去同步，导致从属器件将 SDA_n 线固定为 Low 电平的状态而无法发行停止条件的总线异常。在这种情况下，RIIC 能通过使用 SCL 时钟追加输出功能逐个输出追加时钟，释放从属器件的 SDA_n 线 Low 电平的固定状态，恢复总线状态。能通过检查 ICCR1.SDAI 位，确认此从属器件的 SDA_n 线的释放。为了在确认从属器件的 SDA_n 线被释放后结束通信，必须重新发行停止条件。

在使用此功能时，必须将 ICFER.MALE 位置“0”（禁止主控仲裁失败检测）。必须注意：当 MALE 位为“1”（允许主控仲裁失败检测）时，在 ICCR1.SDAO 位的值与 SDA_n 线上的信号状态不同时会产生仲裁失败。

[ICCR1.CLO 位的输出条件]

- 在总线空闲状态（ICCR2.BBSY 标志=0）下或者在主机模式（ICCR2.MST 位或者 BBSY 标志为“1”的状态）中。
- 通信器件没有保持 SCL_n 线的 Low 电平状态。

SCL 时钟追加输出功能（CLO 位）如图 22.39 所示。

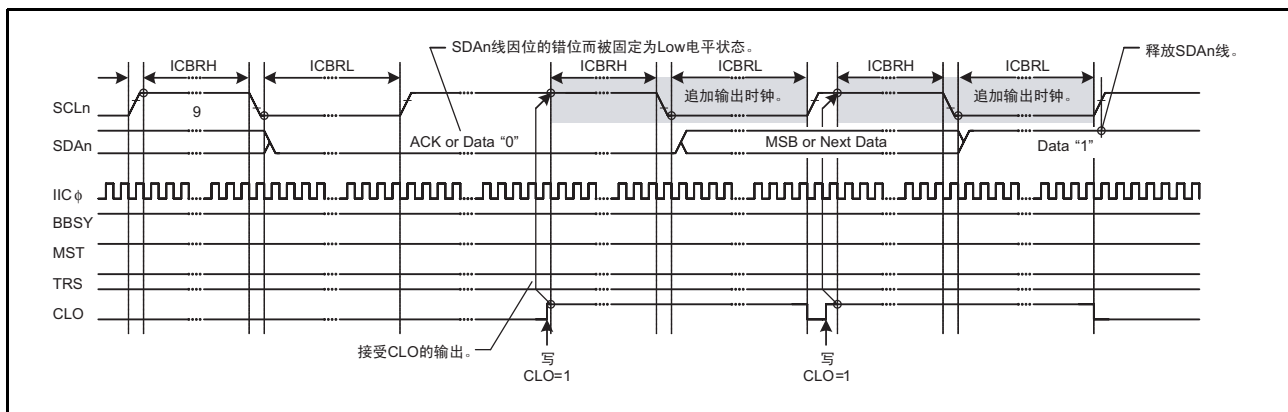


图 22.39 SCL 时钟追加输出功能（CLO 位）

22.11.3 RIIC/ 内部复位

RIIC 具有对 RIIC 模块进行复位的功能。有 2 种复位，1 种是对包括 ICCR2.BBSY 标志在内的全部寄存器进行初始化的 RIIC 复位，另一种是在保持各种设定值的状态下解除从属地址匹配状态以及对内部计数器进行初始化等的内部复位。

复位后，必须将 ICCR1.IICRST 位置“0”。

因为无论进行哪种复位都要解除 SCL_n 引脚 /SDA_n 引脚的输出状态而变为高阻抗状态，所以也能用于解除总线意外停机状态。

从属模式中的复位会引起与主控器件的不同步，因此尽量避免使用。必须注意：在 RIIC 复位（ICCR1.ICE 位和 ICCR1.IICRST 位为“01b”）过程中不能监视开始条件等的总线状态。

有关 RIIC/ 内部复位的详细内容，请参照“22.14 复位状况”。

22.12 SMBus 运行

RIIC 能进行以 SMBus (Ver.2.0) 为基准的通信。在进行 SMBus 通信时, 必须将 ICMR3.SMBS 位置 “1”。必须通过设定 ICMR1.CKS[2:0] 位、ICBRH 寄存器和 ICBRL 寄存器, 将传送率设定在 SMBus 规格的 10kbps ~ 100kbps 范围内, 通过设定 ICMR2.DLCS 位和 ICMR2.SDDL[2:0] 位的值, 使数据的保持时间符合 300ns (min) 的规格。在只将 RIIC 用作从属器件时, 不需要设定传送率。

对于 SMBus 设备的默认地址 (1100 001b), 必须使用从属地址寄存器 L0 ~ L2 (SARL0、SARL1、SARL2) 中的任意 1 个, 并且必须将对应的 SARUy.FS 位 (y=0 ~ 2) (7 位 /10 位地址格式选择位) 置 “0” (7 位地址格式)。

在发送 UDID (唯一设备标识符) 时, 必须将 ICFER.SALE 位置 “1”, 使从属仲裁失败检测功能有效。

22.12.1 SMBus 超时测量

(1) 从属器件的超时测量

SMBus 通信的从属器件需要测量以下所示的区间 (超时间隔: $T_{\text{LOW:SEXT}}$)。

- 开始条件到停止条件的区间

在通过从属器件进行超时测量时, 使用 RIIC 的开始条件检测中断 (STI) 和停止条件检测中断 (SPI) 并且通过 TPU 定时器或者 TMR 定时器, 测量从检测到开始条件到检测到停止条件的时间。此超时的测量时间必须在 SMBus 规格的时钟 Low 电平的累积时间 [从属器件] $T_{\text{LOW:SEXT}}$: 25ms (max) 以内。

如果通过 TPU 或者 TMR 测量的时间超过 SMBus 规格的时钟 Low 电平检测的超时 T_{TIMEOUT} : 25ms (min), 从属器件就需要释放总线。从属器件释放总线时, 必须给 ICCR1.IICRST 位写 “1”, 进行 RIIC 的内部复位。一旦进行内部复位, RIIC 就能中止 SCLn 引脚 /SDAn 引脚的总线驱动, 并且将引脚置为高阻抗状态, 因此能释放总线。

(2) 主控制件的超时测量

SMBus 通信的主控器件需要测量以下所示的区间 (超时间隔: $T_{\text{LOW:MEXT}}$)。

- 开始条件到应答位的区间
- 应答位到下一个应答位的区间
- 应答位到停止条件的区间

在通过主控器件进行超时测量时, 使用 RIIC 的开始条件检测中断 (STI)、停止条件检测中断 (SPI) 以及发送结束中断 (ICTEI) 或者接收数据满中断 (ICRXI), 通过 TPU 定时器或者 TMR 定时器测量各区间的时间。此超时的测量时间必须在 SMBus 规格的时钟 Low 电平的累积时间 [主控器件] $T_{\text{LOW:MEXT}}$: 10ms (max) 以内, 开始条件到停止条件的全部 $T_{\text{LOW:MEXT}}$ 的累加结果必须在 $T_{\text{LOW:SEXT}}$: 25ms (max) 以内。

在主控发送模式中 (主控发送器), 需要通过 ICSR2.TEND 标志监视 ACK 接收时序 (SMBCLK 的第 9 个时钟上升); 在主控接收模式中 (主控接收器), 需要通过 ICSR2.RDRF 标志监视 ACK 接收时序。因此, 在主控发送时, 进行 1 字节的发送; 在主控接收时, 必须在接收最后字节前将 ICMR3.RDRFS 位置 “0”。当 RDRFS 位为 “0” 时, RDRF 标志在 SMBCLK 的第 9 个时钟的上升沿变为 “1”。

如果通过 TPU 或者 TMR 测量的时间超过 SMBus 规格的时钟 Low 电平的累积时间 [主控器件] $T_{\text{LOW:MEXT}}$: 10ms (max), 或者各测量时间的累加结果超过 SMBus 规格的时钟 Low 电平检测的超时 T_{TIMEOUT} : 25ms (min), 主控器件就需要中止处理。在主控发送时, 必须立即中止发送 (写 ICDRT 寄存器)。通过发行停止条件中止主控器件的处理。

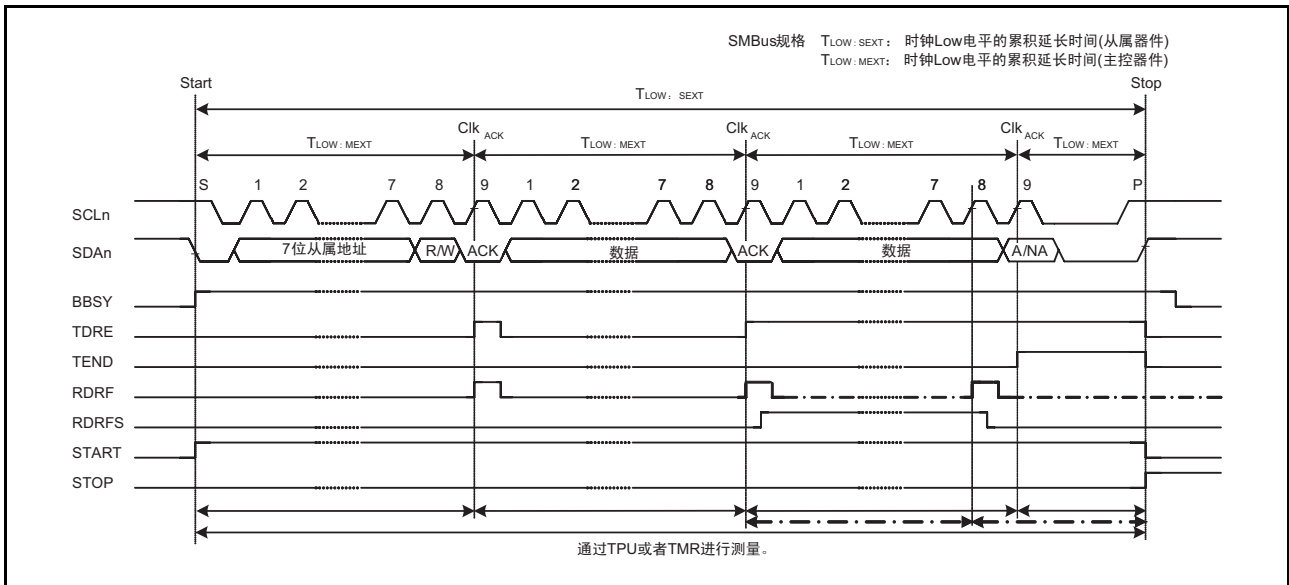


图 22.40 SMBus 超时测量

22.12.2 数据包错误码 (PEC)

RX610 群内置 CRC 运算器。在 RIIC 通信中，能利用 CRC 运算器，发送 SMBus 的数据包错误码 (PEC) 或者检查接收数据。有关 CRC 运算器的多项式，请参照“21. CRC 运算器 (CRC)”。

能通过给 CRC 运算器的 CRC 数据输入寄存器 (CRCDIR) 写全部的发送数据，生成主控发送 (主控发送器) 的 PEC 数据。

通过给 CRC 运算器的 CRCDIR 寄存器写全部的接收数据，得到 CRC 数据输出寄存器 (CRCDOR) 的值，然后将此值和接收的 PEC 数据进行比较，检查主控接收 (主控接收器) 的 REC 数据。

在检查 PEC 码的过程中接收最后的字节接收时，根据是否匹配发送 ACK 或者 NACK，就必须在接收最后字节的 SMBCLK 的第 8 个时钟上升前将 ICMR3.RDRFS 位置“1”，并且在第 8 个时钟的下降沿将 SCLn 线保持为 Low 电平。

22.12.3 SMBus 主机通知协议 /Notify ARP master

在 SMBus 通信中，从属器件能临时变为主控器件，将自己的自从属地址通知给 SMBus 主机 (或者 ARP 主控)，或者向 SMBus 主机请求自己的从属地址。

将 RX610 群用作 SMBus 主机 (或者 ARP 主控) 时，需要将主从属器件发送的主机地址 (0001 000b) 作为从属地址进行检测，RIIC 具有此主机地址的检测功能。如果将主机地址作为从属地址进行检测，就必须将 ICMR3.SMBS 位和 ICSER.HOAE 位置“1”。主机地址检测后的运行和普通的从属模式运行相同。

22.13 中断源

RIIC 的中断源有通信错误 / 事件的发生 (仲裁失败检测、NACK 检测、超时检测、开始条件检测、停止条件检测)、接收数据满、发送数据空、发送结束共 4 种。

各中断一览表如表 22.7 所示。能根据接收数据满和发送数据空的中断请求, 在启动 DTC 或者 DMAC 后进行数据传送。

表 22.7 中断源

| 名称 | 中断源 | 中断标志 | DTC 的启动 | DMAC 的启动 | 优先级 | 中断条件 |
|-----------------------|-------|-------|---------|----------|--------|--------------------|
| ICEEI 通信错误 / 事件的发生 | | AL | 不能 | 不能 | 高 ↑ | AL=1 并且 ALIE=1 |
| | | NACKF | | | | NACKF=1 并且 NAKIE=1 |
| | | TMOF | | | | TMOF=1 并且 TMOIE=1 |
| | | START | | | | START=1 并且 STIE=1 |
| | | STOP | | | | STOP=1 并且 SPIE=1 |
| ICRXI | 接收数据满 | — | 能 | 能 | 低 ↓ | RDRF=1 并且 RIE=1 |
| ICTXI | 发送数据空 | — | 能 | 能 | | TDRE=1 并且 TIE=1 |
| ICTEI | 发送结束 | TEND | 不能 | 不能 | | TEND=1 并且 TEIE=1 |

必须在中断处理中清除或者屏蔽各自的标志。

【中断处理时的注意事项】

1. CPU 执行写外围模块的指令和实际写模块的时序有延时。如果清除或者屏蔽中断标志, 就必须重新读中断标志, 并且在确认清除或者写屏蔽位结束后从中断处理返回。如果不确认已写完模块而从中断处理返回, 就可能再次产生相同的中断。
2. 因为 ICTXI 中断为边沿中断, 所以不需要清除。通过给 ICDRT 寄存器写发送数据或者检测到停止条件 (ICSR2.STOP 标志 =1), 作为 ICTXI 中断条件的 ICSR2.TDRE 标志自动变为 “0”。
3. 因为 ICRXI 中断为边沿中断, 所以不需要清除。通过读 ICDRR 寄存器, 作为 ICRXI 中断条件的 ICSR2.RDRF 标志自动变为 “0”。
4. 在使用 ICTEI 中断时, 必须在 ICTEI 中断处理中清除 ICSR2.TEND 标志。通过给 ICDRT 寄存器写发送数据或者检测到停止条件 (ICSR2.STOP 标志 =1), ICSR2.TEND 标志自动变为 “0”。

22.14 复位状况

RIIC 具有芯片复位、RIIC 复位和内部复位的复位功能，各种复位的范围和状况如表 22.8 所示。

表 22.8 复位状况

| | | 芯片复位 | RIIC 复位 (ICE 位 =0 并且 IICRST 位 =1) | 内部复位 (ICE 位 =1 并且 IICRST 位 =1) | 开始条件 / 重新开始 条件的检测 | 停止条件的检测 | |
|--|----------------|------|---|--------------------------------------|----------------------|---------|---------|
| ICCR1 | ICE、 IICRST | 复位 | 保持 | 保持 | 运行 (保持) | 运行 (保持) | |
| | SCLO、 SDAO | | 复位 | 复位 | | | |
| | 其他 | | | 保持 | | | |
| ICCR2 | BBSY | 复位 | 复位 | 运行 | 运行 | 运行 | |
| | ST | | | 复位 | 复位 | 运行 (保持) | |
| | 其他 | | | | | 复位 | |
| ICMR1 | BC[2:0] | 复位 | 复位 | 复位 | 复位 | 运行 (保持) | |
| | 其他 | | | 保持 | 运行 (保持) | | |
| ICMR2 | | 复位 | 复位 | 保持 | 运行 (保持) | 运行 (保持) | |
| ICMR3 | | 复位 | 复位 | 保持 | 运行 (保持) | 运行 (保持) | |
| ICFER | | 复位 | 复位 | 保持 | 运行 (保持) | 运行 (保持) | |
| ICSER | | 复位 | 复位 | 保持 | 运行 (保持) | 运行 (保持) | |
| ICIER | | 复位 | 复位 | 保持 | 运行 (保持) | 运行 (保持) | |
| ICSR1 | | 复位 | 复位 | 复位 | 运行 (保持) | 复位 | |
| ICSR2 | TDRE、 TEND | 复位 | 复位 | 复位 | 运行 (保持) | 复位 | |
| | START | | | | 运行 | | |
| | STOP | | | | 运行 (保持) | | 运行 |
| | 其他 | | | | | | 运行 (保持) |
| SARL0、SARL1、 SARL2 SARU0、SARU1、 SARU2 | | 复位 | 复位 | 保持 | 运行 (保持) | 运行 (保持) | |
| ICBRH、ICBRL | | 复位 | 复位 | 保持 | 运行 (保持) | 运行 (保持) | |
| ICDRT | | 复位 | 复位 | 保持 | 运行 (保持) | 运行 (保持) | |
| ICDRR | | 复位 | 复位 | 保持 | 运行 (保持) | 运行 (保持) | |
| ICDRS | | 复位 | 复位 | 复位 | 运行 (保持) | 运行 (保持) | |
| 超时检测功能 | | 复位 | 复位 | 运行 | 运行 | 运行 | |
| 总线空闲时间的测量 | | 复位 | 复位 | 运行 | 运行 | 运行 | |

22.15 使用时的注意事项

22.15.1 模块停止功能的设定

能通过模块停止控制寄存器 B (MSTPCRB) 转移到模块停止状态或者解除模块停止状态, 初始值为 RIIC 处于模块停止状态。能通过解除模块停止状态, 使 RIIC 的寄存器变为可存取的状态。

有关模块停止控制寄存器 B 的详细内容, 请参照“8. 低功耗功能”。

22.15.2 输入缓冲控制寄存器的设定

能通过输入缓冲控制寄存器 (Pm.ICR) 设定外围模块输入的有效或者无效, 初始值为 RIIC 输入无效状态。

因为 I²C 总线的 SCL 线和 SDA 线是双向线, 所以 RIIC 的 SCLn 引脚和 SDA_n 引脚为输入/输出引脚。必须设定与 RIIC 的 SCLn 引脚和 SDA_n 引脚对应的 P1.ICR 寄存器的输入缓冲控制位, 并且将 RIIC 的输入设定为有效。当对应引脚的输入处于无效状态时, 不能进行开始条件 (包括重新开始条件) 和停止条件的检测以及 SCL 时钟的计数。

有关输入缓冲控制寄存器的详细内容, 请参照“14. I/O 端口”。

22.15.3 发送应答位的写和输出时序

发送应答位 (ICMR3.ACKBT 位) 与 RDRF 标志的置位时序选择位 (ICMR3.RDRFS 位) 的设定值无关, 在 SCL 时钟的第 8 个时钟下降时输出 ACKBT 位的值。在下一个帧的第 8 个时钟下降时, 输出在 SCL 时钟的第 8 个时钟下降后给 ACKBT 位写的值。

22.15.4 主控发送时的停止条件发行请求和发送数据写时序的限制事项

在主控发送模式中, 如果在发送结束标志 (ICSR2.TEND 标志) 为“1”的状态下第 9 个时钟的 Low 电平输出的结束时序和停止条件的发行请求 (给 ICCR2.SP 位写“1”) 发生竞争, 一旦因外接的上拉电阻值使 SCL 线急速上升, 就可能与因发行停止条件使 SDA 线下降的关系而产生似如开始条件的波形。在这种情况下, 必须改变外接的上拉电阻值或者在等待第 9 个时钟的 Low 电平输出结束后请求发行停止条件。

同样地, 在主控发送模式中, 如果在发送结束标志 (ICSR2.TEND 标志) 为“1”的状态下第 9 个时钟的 Low 电平输出的结束时序和发送数据 (ICDRT) 的写操作发生竞争, 就可能因在 SDA 线上瞬间输出 Low 电平而产生似如停止条件的波形。在主控发送模式中, 必须在发送结束标志为“0”的状态下写发送数据, 如果发送结束标志为“1”, 就必须在等待第 9 个时钟的 Low 电平输出结束后写发送数据。

22.15.5 主控模式中接收到 NACK 时的重新开始通信的注意事项

如果在主控发送模式中接收到从属器件的 NACK (ICMR3.ACKBR=1), 就必须在通过发行停止条件结束通信后, 通过发行开始条件来重新开始通信。

RIIC 的发送运行行为缓冲运行, 如果不发行用于结束通信的停止条件而发行重新开始条件继续通信, 就可能输出未被发送的 ICDRT 的发送数据。如果在接收到 NACK 后要重新开始通信, 就必须在通过发行停止条件结束通信后, 通过发行开始条件来重新开始通信。

22.15.6 RDRF 标志的置位时序选择位 (RDRFS 位) 的注意事项

必须注意: 在从属模式中, 如果在 RDRF 标志的置位时序选择位 (RDRFS 位) 为“1” (在 SCL 时钟的第 8 个时钟上升时将接收数据满标志 (RDRF 标志) 置“1”) 的状态下从属地址匹配 1 次后, 不通过停止条件结束通信而通过重新开始条件再次指定 RIIC 的从属地址 (SAR0 ~ SAR2) 时, 就在地址帧的 SCL 的第 8 个时钟 RDRF 标志变为“1”后, 在 SCL 的第 9 个时钟 RDRF 标志再次变为“1”。

23. A/D 转换器

23.1 概要

RX610 群内置 4 个单元（单元 0 ~ 3）的逐次逼近方式的 10 位 A/D 转换器，各单元最多能选择 4 个通道的模拟输入。

A/D 转换器的运行模式有对 1 个通道的模拟输入只进行 1 次转换的单次模式，以及对最多 4 个通道的模拟输入依次进行连续转换的扫描模式。

A/D 转换器的规格以及各单元的功能比较分别如表 23.1 和表 23.2 所示，各单元的框图如图 23.1 ~ 图 23.4 所示。

表 23.1 A/D 转换器的规格

| 项目 | 规格 |
|------------|---|
| 单元数 | 4 个单元 |
| 输入通道 | 各单元有 4 个通道（共 16 个通道）。 |
| A/D 转换方式 | 逐次逼近方式 |
| 分辨率 | 10 位 |
| 转换时间 | 每个通道 1.0 μ s（外围模块时钟，以 PCLK=50MHz 进行工作时） |
| A/D 转换时钟 | 4 种：PCLK、PCLK/2、PCLK/4、PCLK/8 |
| 运行模式 | <ul style="list-style-type: none"> • 单次模式：对 1 个通道的模拟输入只进行 1 次转换。 • 扫描模式 连续扫描模式：重复转换最多 4 个通道的模拟输入。 1 个周期扫描模式：对最多 4 个通道的模拟输入只进行 1 个周期的转换。 |
| A/D 转换开始条件 | <ul style="list-style-type: none"> • 软件触发 • 16 位定时器脉冲单元（TPU）或者 8 位定时器（TMR）触发 • 外部触发 能通过各单元的 ADTRGn# 引脚启动 A/D 转换。 能通过 ADTRG0# 引脚同时启动单元 0 和单元 1 的 A/D 转换。 能通过 ADTRG2# 引脚同时启动单元 2 和单元 3 的 A/D 转换。 |
| 功能 | <ul style="list-style-type: none"> • 采样 & 保持功能 • 采样状态数的可变功能 |
| 中断源 | <ul style="list-style-type: none"> • 各单元在 A/D 转换结束时产生 ADC 中断请求（ADI）。 • 能通过 ADI 中断来启动数据传送控制器（DTC）和 DMA 控制器（DMAC）。 |
| 低功耗功能 | 各单元能设定为模块停止状态。 |

表 23.2 各单元的功能比较

| 项目 | | 单元 0 (AD0) | 单元 1 (AD1) | 单元 2 (AD2) | 单元 3 (AD3) | |
|-------------------------|-------|---------------------------|--------------------------|----------------------------|------------------------------|------|
| 模拟输入通道 | | AN0 AN1 AN2 AN3 | AN4 AN5 AN6 AN7 | AN8 AN9 AN10 AN11 | AN12 AN13 AN14 AN15 | |
| A/D 转换 开始条件 (注 1) | 软件 | 软件触发 | ○ | ○ | ○ | ○ |
| | 外部触发 | ADTRG0# | ○ | ○ | — | — |
| | | ADTRG1# | — | ○ | — | — |
| | | ADTRG2# | — | — | ○ | ○ |
| | | ADTRG3# | — | — | — | ○ |
| | 定时器触发 | TPU0 的比较匹配 / 输入捕捉 | TGRA | TGRB | TGRC | TGRD |
| | | TPU0 ~ TPU5 的比较匹配 / 输入捕捉 | TGRA | TGRA | TGRA | TGRA |
| | | TPU6 ~ TPU11 的比较匹配 / 输入捕捉 | TGRA | TGRA | TGRA | TGRA |
| | | TMR0 的比较匹配 | 比较匹配 A | 比较匹配 A | — | — |
| TMR2 的比较匹配 | | — | — | 比较匹配 A | 比较匹配 A | |
| 中断 | | ADI0 | ADI1 | ADI2 | ADI3 | |
| 模块停止功能的设定 (注 2) | | MSTPCRA. MSTPA23 位 | MSTPCRA. MSTPA22 位 | MSTPCRA. MSTPA21 位 | MSTPCRA. MSTPA20 位 | |

○：能

—：不能

注 1. 各单元能选择 A/D 转换开始条件。

注 2. 详细内容请参照“8. 低功耗功能”。

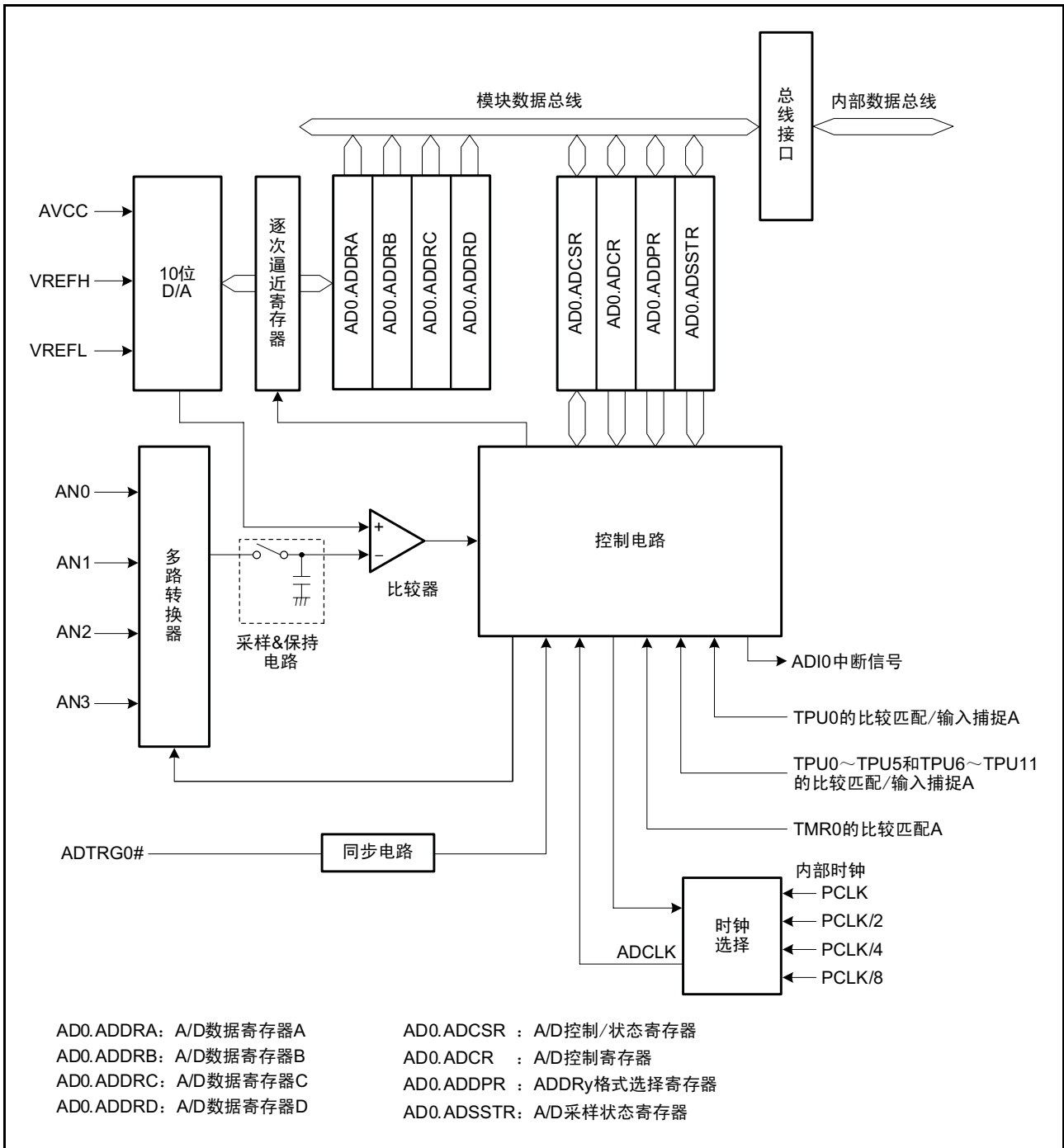


图 23.1 A/D 转换器（单元 0/AD0）的框图

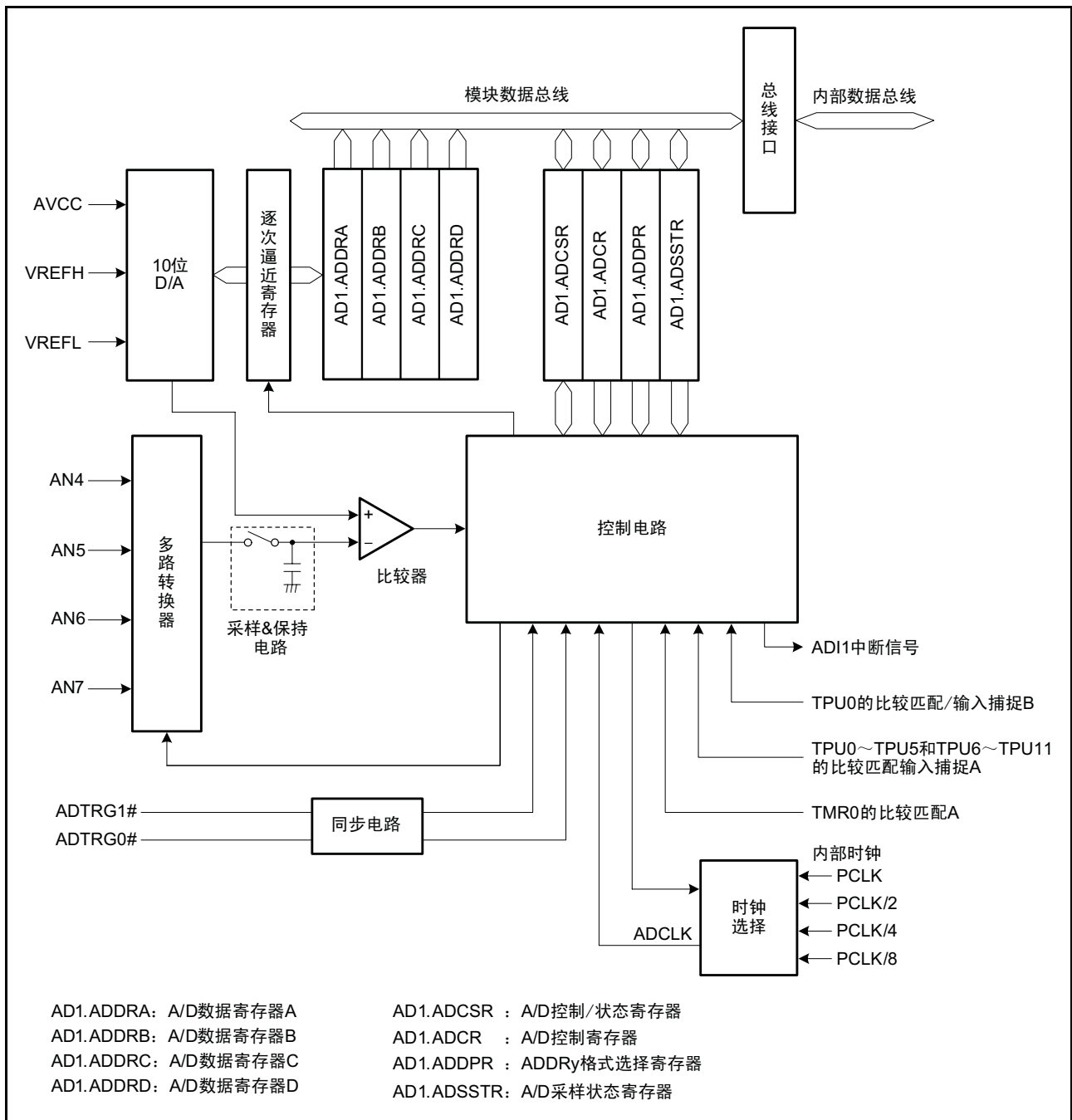


图 23.2 A/D 转换器（单元 1/AD1）的框图

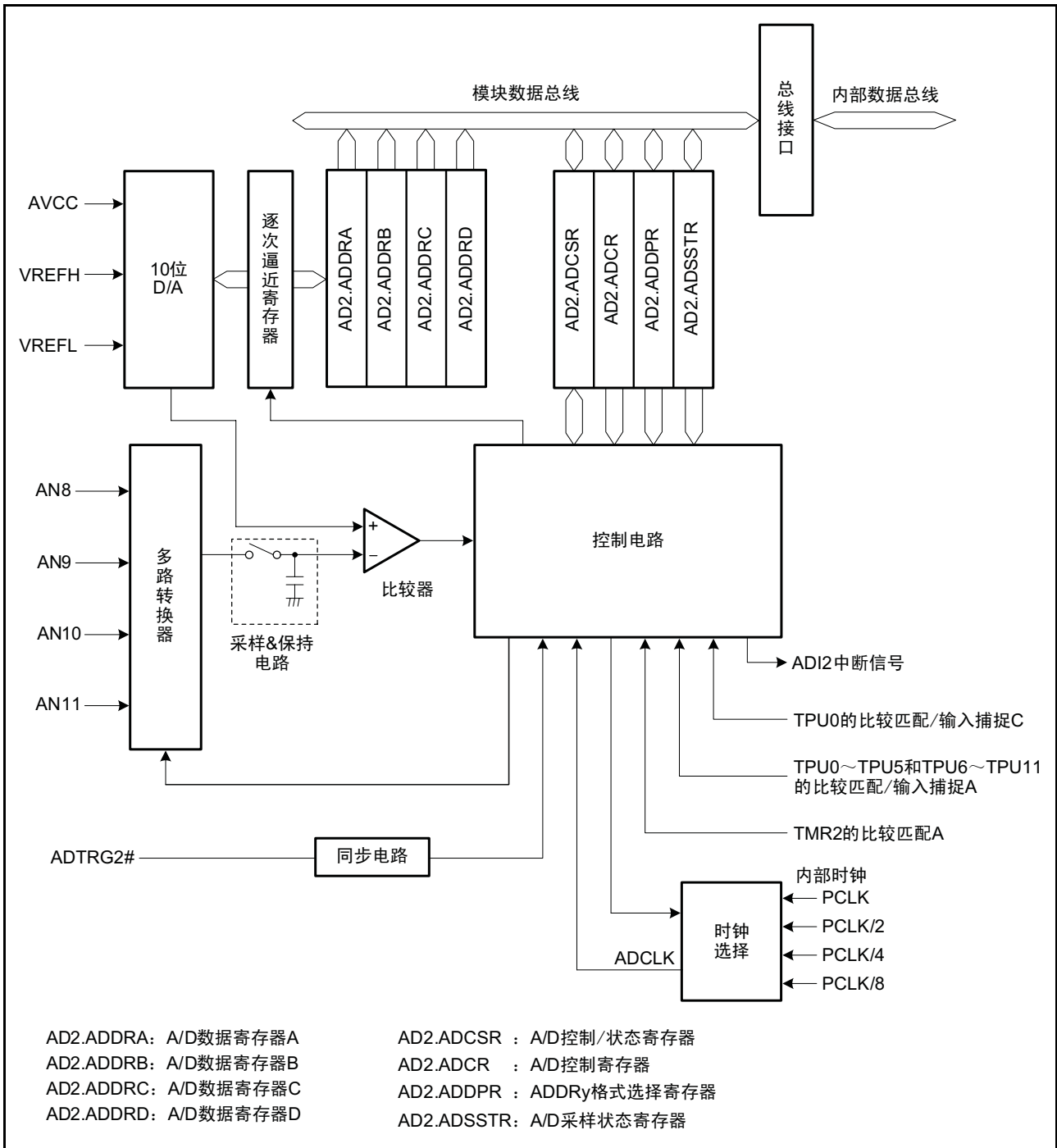


图 23.3 A/D 转换器（单元 2/AD2）的框图

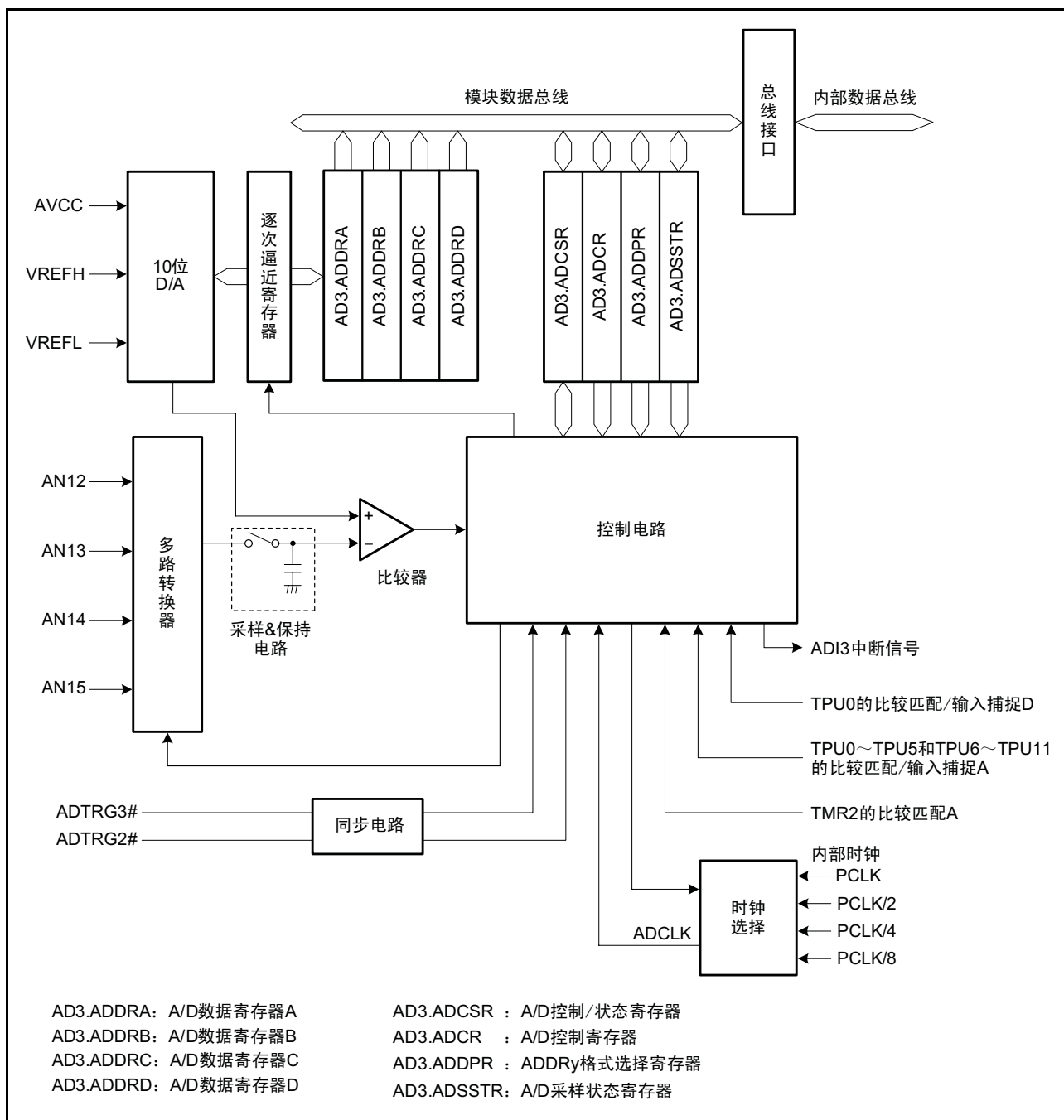


图 23.4 A/D 转换器（单元 3/AD3）的框图

A/D 转换器使用的输入引脚如表 23.3 所示。

表 23.3 A/D 转换器的输入引脚

| 单元 | 模块符号 | 引脚名 | 输入 | 功能 |
|----|------|-------------|----|------------------------------------|
| 0 | AD0 | AN0 ~ AN3 | 输入 | 模拟输入引脚 |
| | | ADTRG0# | 输入 | 用于开始 A/D 转换的外部触发输入引脚 |
| 1 | AD1 | AN4 ~ AN7 | 输入 | 模拟输入引脚 |
| | | ADTRG0# | 输入 | 用于开始 A/D 转换的外部触发输入引脚 |
| | | ADTRG1# | 输入 | 用于开始 A/D 转换的外部触发输入引脚 |
| 2 | AD2 | AN8 ~ AN11 | 输入 | 模拟输入引脚 |
| | | ADTRG2# | 输入 | 用于开始 A/D 转换的外部触发输入引脚 |
| 3 | AD3 | AN12 ~ AN15 | 输入 | 模拟输入引脚 |
| | | ADTRG2# | 输入 | 用于开始 A/D 转换的外部触发输入引脚 |
| | | ADTRG3# | 输入 | 用于开始 A/D 转换的外部触发输入引脚 |
| 通用 | | AVCC | 输入 | 模拟电路的电源引脚 |
| | | AVSS | 输入 | 模拟电路的接地引脚 |
| | | VREFH | 输入 | A/D 转换器的基准电源引脚 |
| | | VREFL | 输入 | A/D 转换器的基准接地引脚 必须连接模拟基准电源 (0V)。 |

23.2 寄存器说明

A/D 转换器的寄存器一览表如表 23.4 所示。

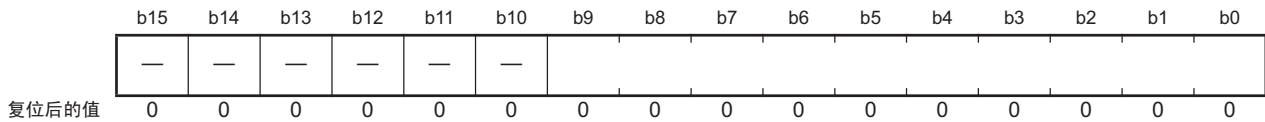
表 23.4 A/D 转换器的寄存器一览表

| 单元 | 模块符号 | 寄存器名 | 寄存器符号 | 复位后的值 | 地址 | 存取长度 |
|----|------|----------------|--------|-------|------------|------|
| 0 | AD0 | A/D 数据寄存器 A | ADDRA | 0000h | 0008 8040h | 16 |
| | | A/D 数据寄存器 B | ADDRB | 0000h | 0008 8042h | 16 |
| | | A/D 数据寄存器 C | ADDRC | 0000h | 0008 8044h | 16 |
| | | A/D 数据寄存器 D | ADDRD | 0000h | 0008 8046h | 16 |
| | | A/D 控制 / 状态寄存器 | ADCSR | x0h | 0008 8050h | 8 |
| | | A/D 控制寄存器 | ADCR | 00h | 0008 8051h | 8 |
| | | ADDRy 格式选择寄存器 | ADDPR | 00h | 0008 8052h | 8 |
| | | A/D 采样状态寄存器 | ADSSTR | 19h | 0008 8053h | 8 |
| 1 | AD1 | A/D 数据寄存器 A | ADDRA | 0000h | 0008 8060h | 16 |
| | | A/D 数据寄存器 B | ADDRB | 0000h | 0008 8062h | 16 |
| | | A/D 数据寄存器 C | ADDRC | 0000h | 0008 8064h | 16 |
| | | A/D 数据寄存器 D | ADDRD | 0000h | 0008 8066h | 16 |
| | | A/D 控制 / 状态寄存器 | ADCSR | x0h | 0008 8070h | 8 |
| | | A/D 控制寄存器 | ADCR | 00h | 0008 8071h | 8 |
| | | ADDRy 格式选择寄存器 | ADDPR | 00h | 0008 8072h | 8 |
| | | A/D 采样状态寄存器 | ADSSTR | 19h | 0008 8073h | 8 |
| 2 | AD2 | A/D 数据寄存器 A | ADDRA | 0000h | 0008 8080h | 16 |
| | | A/D 数据寄存器 B | ADDRB | 0000h | 0008 8082h | 16 |
| | | A/D 数据寄存器 C | ADDRC | 0000h | 0008 8084h | 16 |
| | | A/D 数据寄存器 D | ADDRD | 0000h | 0008 8086h | 16 |
| | | A/D 控制 / 状态寄存器 | ADCSR | x0h | 0008 8090h | 8 |
| | | A/D 控制寄存器 | ADCR | 00h | 0008 8091h | 8 |
| | | ADDRy 格式选择寄存器 | ADDPR | 00h | 0008 8092h | 8 |
| | | A/D 采样状态寄存器 | ADSSTR | 19h | 0008 8093h | 8 |
| 3 | AD3 | A/D 数据寄存器 A | ADDRA | 0000h | 0008 80A0h | 16 |
| | | A/D 数据寄存器 B | ADDRB | 0000h | 0008 80A2h | 16 |
| | | A/D 数据寄存器 C | ADDRC | 0000h | 0008 80A4h | 16 |
| | | A/D 数据寄存器 D | ADDRD | 0000h | 0008 80A6h | 16 |
| | | A/D 控制 / 状态寄存器 | ADCSR | x0h | 0008 80B0h | 8 |
| | | A/D 控制寄存器 | ADCR | 00h | 0008 80B1h | 8 |
| | | ADDRy 格式选择寄存器 | ADDPR | 00h | 0008 80B2h | 8 |
| | | A/D 采样状态寄存器 | ADSSTR | 19h | 0008 80B3h | 8 |

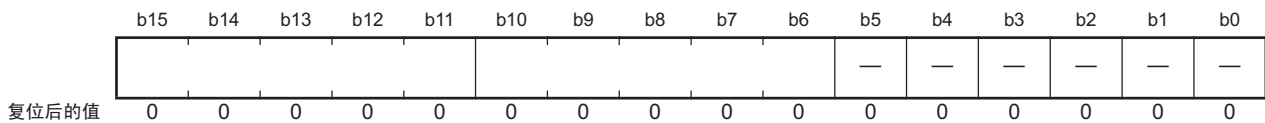
23.2.1 A/D 数据寄存器 y (ADDRy) (y=A ~ D)

地址 AD0.ADDRA 0008 8040h、AD0.ADDRB 0008 8042h、AD0.ADDRC 0008 8044h、AD0.ADDRD 0008 8046h
 AD1.ADDRA 0008 8060h、AD1.ADDRB 0008 8062h、AD1.ADDRC 0008 8064h、AD1.ADDRD 0008 8066h
 AD2.ADDRA 0008 8080h、AD2.ADDRB 0008 8082h、AD2.ADDRC 0008 8084h、AD2.ADDRD 0008 8086h
 AD3.ADDRA 0008 80A0h、AD3.ADDRB 0008 80A2h、AD3.ADDRC 0008 80A4h、AD3.ADDRD 0008 80A6h

- ADDPR.DPSEL位=0 (数据往LSB靠紧)



- ADDPR.DPSEL位=1 (数据往MSB靠紧)



ADDRy 寄存器是保存 A/D 转换结果的 16 位只读寄存器。

模拟输入通道和 ADDRy 寄存器的对应如表 23.5 所示。

能通过设定 ADDPR.DPSEL 位，更改 10 位数据的排列。

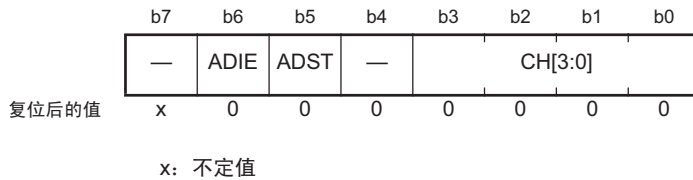
“—” 的位的读写值都为 “0”。

表 23.5 模拟输入通道和 ADDRy 寄存器的对应

| 模拟输入通道 | ADDRy 寄存器 |
|--------|-----------|
| AN0 | AD0.ADDRA |
| AN1 | AD0.ADDRB |
| AN2 | AD0.ADDRC |
| AN3 | AD0.ADDRD |
| AN4 | AD1.ADDRA |
| AN5 | AD1.ADDRB |
| AN6 | AD1.ADDRC |
| AN7 | AD1.ADDRD |
| AN8 | AD2.ADDRA |
| AN9 | AD2.ADDRB |
| AN10 | AD2.ADDRC |
| AN11 | AD2.ADDRD |
| AN12 | AD3.ADDRA |
| AN13 | AD3.ADDRB |
| AN14 | AD3.ADDRC |
| AN15 | AD3.ADDRD |

23.2.2 A/D 控制 / 状态寄存器 (ADCSR)

地址 AD0.ADCSR 0008 8050h、AD1.ADCSR 0008 8070h
AD2.ADCSR 0008 8090h、AD3.ADCSR 0008 80B0h



| 位 | 符号 | 位名 | 功能 | R/W | | |
|-------|---------|------------|--|---|--|-----|
| b3-b0 | CH[3:0] | 通道选择位 (注1) | 单元 | 单次模式 (ADCR.MODE[1:0] 位 为 “00b”) | 扫描模式 (ADCR.MODE[1:0] 位 为 “10b” 或者 “11b”) | R/W |
| | | | 单元 0 | b3 b0 0 0 0 0: AN0 0 0 0 1: AN1 0 0 1 0: AN2 0 0 1 1: AN3 上述以外: 不能设定 | b3 b0 0 0 0 0: AN0 0 0 0 1: AN0、AN1 0 0 1 0: AN0 ~ AN2 0 0 1 1: AN0 ~ AN3 上述以外: 不能设定 | |
| | | | 单元 1 | b3 b0 0 0 0 0: AN4 0 0 0 1: AN5 0 0 1 0: AN6 0 0 1 1: AN7 上述以外: 不能设定 | b3 b0 0 0 0 0: AN4 0 0 0 1: AN4、AN5 0 0 1 0: AN4 ~ AN6 0 0 1 1: AN4 ~ AN7 上述以外: 不能设定 | |
| | | | 单元 2 | b3 b0 0 0 0 0: AN8 0 0 0 1: AN9 0 0 1 0: AN10 0 0 1 1: AN11 上述以外: 不能设定 | b3 b0 0 0 0 0: AN8 0 0 0 1: AN8、AN9 0 0 1 0: AN8 ~ AN10 0 0 1 1: AN8 ~ AN11 上述以外: 不能设定 | |
| | | | 单元 3 | b3 b0 0 0 0 0: AN12 0 0 0 1: AN13 0 0 1 0: AN14 0 0 1 1: AN15 上述以外: 不能设定 | b3 b0 0 0 0 0: AN12 0 0 0 1: AN12、AN13 0 0 1 0: AN12 ~ AN14 0 0 1 1: AN12 ~ AN15 上述以外: 不能设定 | |
| b4 | — | 保留位 | 读写值都为 “0”。 | R/W | | |
| b5 | ADST | A/D 开始位 | 0: 停止 A/D 转换 1: 开始 A/D 转换 | R/W | | |
| b6 | ADIE | A/D 中断允许位 | 0: 禁止由 A/D 转换结束引起的 ADI 中断 1: 允许由 A/D 转换结束引起的 ADI 中断 | R/W | | |
| b7 | — | 保留位 | 读取值为不定值, 只能写 “1”。 | R/W | | |

注 1. 必须将用作模拟输入引脚的 Pm.DDR.Bj 位置 “0” (输入端口) 并且将 Pm.ICR.Bj 位 (m=4,9, j=0~7) 置 “0” (对应引脚的输入缓冲器无效, 输入信号固定为 High 电平)。详细内容请参照 “14. I/O 端口”。

ADCSR 寄存器是控制 A/D 转换的寄存器。

CH[3:0] 位（通道选择位）

这些位选择进行 A/D 转换的模拟输入通道。

- 单次模式（ADCR.MODE[1:0]位为“00b”）
选择1个进行A/D转换的模拟输入通道。
- 扫描模式（ADCR.MODE[1:0]位为“10b”或者“11b”）
最多选择4个进行A/D转换的模拟输入通道。

ADST 位（A/D 开始位）

此位控制 A/D 转换的开始或者停止。

必须在将 ADST 位置“1”前设定 A/D 转换时钟和运行模式。

[为“1”的条件]

- 通过软件写“1”时
- 检测到ADCR.TRGS[2:0]位所选的触发时

[为“0”的条件]

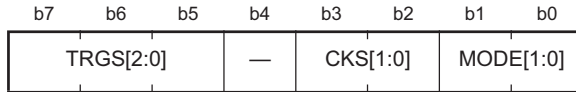
- 通过软件写“0”时
- 在单次模式中A/D转换结束时
- 在1个周期扫描模式中所选通道的A/D转换全部结束时

ADIE 位（A/D 中断允许位）

此位允许或者禁止由 A/D 转换结束引起的 ADI 中断。

23.2.3 A/D 控制寄存器 (ADCR)

地址 AD0.ADCR 0008 8051h、AD1.ADCR 0008 8071h
AD2.ADCR 0008 8091h、AD3.ADCR 0008 80B1h



复位后的值

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

| 位 | 符号 | 位名 | 功能 | R/W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------|--|-----------------------------|--|------|------|------|--|-----|----|----------------------------|-----|---|--------------|-----|---|----------------------------|-----|---|---------------------|-----|---|-----------------------------|-----|---|---------------------|-----|---|-----------------------------|-----|---|------|-----|---|------|------|--|----|----|------|-----|---|------|-----|---|----------------------------|-----|---|--------------|-----|---|------------------|-----|---|---------------------|-----|---|-----------------------------|-----|---|------|-----|---|------------------|-----|
| b1-b0 | MODE[1:0] | 运行模式选择位 | b1 b0 0 0: 单次模式 0 1: 不能设定 1 0: 连续扫描模式 1 1: 1 个周期扫描模式 | R/W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b3-b2 | CKS[1:0] | 时钟选择位 | b3 b2 0 0: PCLK/8 0 1: PCLK/4 1 0: PCLK/2 1 1: PCLK | R/W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b4 | — | 保留位 | 读写值都为“0”。 | R/W | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7-b5 | TRGS[2:0] | 触发选择位 | <table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">单元</th> <th style="width: 90%;">触发信号</th> </tr> </thead> <tbody> <tr> <td>单元 0</td> <td> <table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">b7</th> <th style="width: 10%;">b5</th> <th style="width: 80%;">触发信号</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>0</td> <td>软件触发</td> </tr> <tr> <td>0 0</td> <td>1</td> <td>TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>0 1</td> <td>0</td> <td>TMR0 的比较匹配 A</td> </tr> <tr> <td>0 1</td> <td>1</td> <td>ADTRG0# (注1) 的触发</td> </tr> <tr> <td>1 0</td> <td>0</td> <td>TPU0 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>1 0</td> <td>1</td> <td>TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>1 1</td> <td>0</td> <td>不能设定</td> </tr> <tr> <td>1 1</td> <td>1</td> <td>不能设定</td> </tr> </tbody> </table> </td> </tr> <tr> <td>单元 1</td> <td> <table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">b7</th> <th style="width: 10%;">b5</th> <th style="width: 80%;">触发信号</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>0</td> <td>软件触发</td> </tr> <tr> <td>0 0</td> <td>1</td> <td>TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>0 1</td> <td>0</td> <td>TMR0 的比较匹配 A</td> </tr> <tr> <td>0 1</td> <td>1</td> <td>ADTRG1# (注1) 的触发</td> </tr> <tr> <td>1 0</td> <td>0</td> <td>TPU0 的比较匹配 / 输入捕捉 B</td> </tr> <tr> <td>1 0</td> <td>1</td> <td>TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>1 1</td> <td>0</td> <td>不能设定</td> </tr> <tr> <td>1 1</td> <td>1</td> <td>ADTRG0# (注1) 的触发</td> </tr> </tbody> </table> </td> </tr> </tbody> </table> | 单元 | 触发信号 | 单元 0 | <table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">b7</th> <th style="width: 10%;">b5</th> <th style="width: 80%;">触发信号</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>0</td> <td>软件触发</td> </tr> <tr> <td>0 0</td> <td>1</td> <td>TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>0 1</td> <td>0</td> <td>TMR0 的比较匹配 A</td> </tr> <tr> <td>0 1</td> <td>1</td> <td>ADTRG0# (注1) 的触发</td> </tr> <tr> <td>1 0</td> <td>0</td> <td>TPU0 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>1 0</td> <td>1</td> <td>TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>1 1</td> <td>0</td> <td>不能设定</td> </tr> <tr> <td>1 1</td> <td>1</td> <td>不能设定</td> </tr> </tbody> </table> | b7 | b5 | 触发信号 | 0 0 | 0 | 软件触发 | 0 0 | 1 | TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A | 0 1 | 0 | TMR0 的比较匹配 A | 0 1 | 1 | ADTRG0# (注1) 的触发 | 1 0 | 0 | TPU0 的比较匹配 / 输入捕捉 A | 1 0 | 1 | TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A | 1 1 | 0 | 不能设定 | 1 1 | 1 | 不能设定 | 单元 1 | <table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">b7</th> <th style="width: 10%;">b5</th> <th style="width: 80%;">触发信号</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>0</td> <td>软件触发</td> </tr> <tr> <td>0 0</td> <td>1</td> <td>TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>0 1</td> <td>0</td> <td>TMR0 的比较匹配 A</td> </tr> <tr> <td>0 1</td> <td>1</td> <td>ADTRG1# (注1) 的触发</td> </tr> <tr> <td>1 0</td> <td>0</td> <td>TPU0 的比较匹配 / 输入捕捉 B</td> </tr> <tr> <td>1 0</td> <td>1</td> <td>TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>1 1</td> <td>0</td> <td>不能设定</td> </tr> <tr> <td>1 1</td> <td>1</td> <td>ADTRG0# (注1) 的触发</td> </tr> </tbody> </table> | b7 | b5 | 触发信号 | 0 0 | 0 | 软件触发 | 0 0 | 1 | TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A | 0 1 | 0 | TMR0 的比较匹配 A | 0 1 | 1 | ADTRG1# (注1) 的触发 | 1 0 | 0 | TPU0 的比较匹配 / 输入捕捉 B | 1 0 | 1 | TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A | 1 1 | 0 | 不能设定 | 1 1 | 1 | ADTRG0# (注1) 的触发 | R/W |
| 单元 | 触发信号 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 单元 0 | <table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">b7</th> <th style="width: 10%;">b5</th> <th style="width: 80%;">触发信号</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>0</td> <td>软件触发</td> </tr> <tr> <td>0 0</td> <td>1</td> <td>TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>0 1</td> <td>0</td> <td>TMR0 的比较匹配 A</td> </tr> <tr> <td>0 1</td> <td>1</td> <td>ADTRG0# (注1) 的触发</td> </tr> <tr> <td>1 0</td> <td>0</td> <td>TPU0 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>1 0</td> <td>1</td> <td>TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>1 1</td> <td>0</td> <td>不能设定</td> </tr> <tr> <td>1 1</td> <td>1</td> <td>不能设定</td> </tr> </tbody> </table> | b7 | b5 | 触发信号 | 0 0 | 0 | 软件触发 | 0 0 | 1 | TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A | 0 1 | 0 | TMR0 的比较匹配 A | 0 1 | 1 | ADTRG0# (注1) 的触发 | 1 0 | 0 | TPU0 的比较匹配 / 输入捕捉 A | 1 0 | 1 | TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A | 1 1 | 0 | 不能设定 | 1 1 | 1 | 不能设定 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | b5 | 触发信号 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 0 | 0 | 软件触发 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 0 | 1 | TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 1 | 0 | TMR0 的比较匹配 A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 1 | 1 | ADTRG0# (注1) 的触发 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 0 | 0 | TPU0 的比较匹配 / 输入捕捉 A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 0 | 1 | TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 1 | 0 | 不能设定 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 1 | 1 | 不能设定 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 单元 1 | <table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">b7</th> <th style="width: 10%;">b5</th> <th style="width: 80%;">触发信号</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>0</td> <td>软件触发</td> </tr> <tr> <td>0 0</td> <td>1</td> <td>TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>0 1</td> <td>0</td> <td>TMR0 的比较匹配 A</td> </tr> <tr> <td>0 1</td> <td>1</td> <td>ADTRG1# (注1) 的触发</td> </tr> <tr> <td>1 0</td> <td>0</td> <td>TPU0 的比较匹配 / 输入捕捉 B</td> </tr> <tr> <td>1 0</td> <td>1</td> <td>TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A</td> </tr> <tr> <td>1 1</td> <td>0</td> <td>不能设定</td> </tr> <tr> <td>1 1</td> <td>1</td> <td>ADTRG0# (注1) 的触发</td> </tr> </tbody> </table> | b7 | b5 | 触发信号 | 0 0 | 0 | 软件触发 | 0 0 | 1 | TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A | 0 1 | 0 | TMR0 的比较匹配 A | 0 1 | 1 | ADTRG1# (注1) 的触发 | 1 0 | 0 | TPU0 的比较匹配 / 输入捕捉 B | 1 0 | 1 | TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A | 1 1 | 0 | 不能设定 | 1 1 | 1 | ADTRG0# (注1) 的触发 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| b7 | b5 | 触发信号 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 0 | 0 | 软件触发 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 0 | 1 | TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 1 | 0 | TMR0 的比较匹配 A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 1 | 1 | ADTRG1# (注1) 的触发 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 0 | 0 | TPU0 的比较匹配 / 输入捕捉 B | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 0 | 1 | TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 1 | 0 | 不能设定 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 1 | 1 | ADTRG0# (注1) 的触发 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 位 | 符号 | 位名 | 功能 | | R/W |
|-------|-----------|-------|------|---|-----|
| b7-b5 | TRGS[2:0] | 触发选择位 | 单元 | 触发信号 | R/W |
| | | | 单元 2 | b7 b5 0 0 0: 软件触发 0 0 1: TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A 0 1 0: TMR2 的比较匹配 A 0 1 1: ADTRG2# (注1) 的触发 1 0 0: TPU0 的比较匹配 / 输入捕捉 C 1 0 1: TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A 1 1 0: 不能设定 1 1 1: 不能设定 | |
| | | | 单元 3 | b7 b5 0 0 0: 软件触发 0 0 1: TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A 0 1 0: TMR2 的比较匹配 A 0 1 1: ADTRG3# (注1) 的触发 1 0 0: TPU0 的比较匹配 / 输入捕捉 D 1 0 1: TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A 1 1 0: 不能设定 1 1 1: ADTRG2# (注1) 的触发 | |

注 1. 在将 A/D 转换的开始触发设定为 ADTRGn# (n=0 ~ 3) 时, 必须将 Pm.DDR.Bj 位置 “0” (输入端口) 并且将 Pm.ICR.Bj 位 (m=1,7, j=0,3,4,7) 置 “1” (对应引脚的输入缓冲器有效)。详细内容请参照 “14. I/O 端口”。

ADCR 寄存器设定 A/D 转换的开始触发、运行模式和 A/D 转换时钟。
必须在 ADCSR.ADST 位为 “0” 时设定 ADCR 寄存器。

MODE[1:0] 位 (运行模式选择位)

这些位选择 A/D 转换的运行模式。

CKS[1:0] 位 (时钟选择位)

这些位设定决定 A/D 转换时间的 A/D 转换时钟 (ADCLK) 的频率。

必须将 ADCLK 的频率设定为大于等于 4MHz。

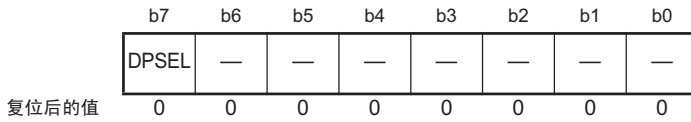
详细内容请参照 “23.3.3 输入采样和 A/D 转换时间”。

TRGS[2:0] 位 (触发选择位)

这些位选择 A/D 转换的开始触发。

23.2.4 ADDRy 格式选择寄存器 (ADDPR)

地址 AD0.ADDPR 0008 8052h、AD1.ADDPR 0008 8072h
AD2.ADDPR 0008 8092h、AD3.ADDPR 0008 80B2h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-------------|--------------------------------|-----|
| b6-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | DPSEL | ADDRy 格式选择位 | 0: 数据往 LSB 靠紧 1: 数据往 MSB 靠紧 | R/W |

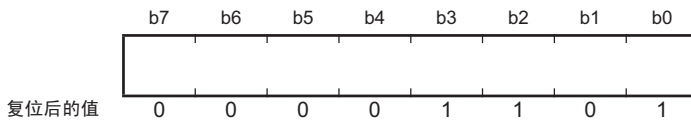
ADDPR 寄存器选择 A/D 数据寄存器的数据排列。

DPSEL 位 (ADDRy 格式选择位)

此位选择 A/D 数据寄存器的数据是往 LSB 靠紧还是往 MSB 靠紧。

23.2.5 A/D 采样状态寄存器 (ADSSTR)

地址 AD0.ADSSTR 0008 8053h、AD1.ADSSTR 0008 8073h
AD2.ADSSTR 0008 8093h、AD3.ADSSTR 0008 80B3h



ADSSTR 寄存器是设定模拟输入的采样时间的 8 位可读写寄存器。

能在模拟输入的信号源阻抗高并且采样时间不够的情况下或者在外围模块时钟 (PCLK) 慢的情况下调整采样时间。

设定值必须大于等于“02h”。

为了避免误动作, 必须在 A/D 转换停止 (ADCSR.ADST 位为“0”) 的状态下进行改写。

详细内容请参照“23.3.3 输入采样和 A/D 转换时间”。

23.3 运行说明

RX610 群内置 4 个单元的 A/D 转换器，各单元的功能相同。

以下说明各单元的运行。

A/D 转换器的运行模式有单次模式和扫描模式。

单次模式对指定的 1 个通道的模拟输入只进行 1 次转换。

扫描模式依次连续转换最多 4 个通道的模拟输入。扫描模式有重复进行 A/D 转换的连续扫描以及对指定的通道只进行 1 个周期 A/D 转换的单次扫描。

23.3.1 单次模式

单次模式对指定的 1 个通道的模拟输入只进行 1 次 A/D 转换，如下所示：

1. 如果通过软件、TPU、TMR 或者外部触发输入使 ADCSR.ADST 位变为 “1”（开始 A/D 转换），就开始所选通道的 A/D 转换。
2. 如果 A/D 转换结束，就将 A/D 转换结果保存到对应该通道的 A/D 数据寄存器（ADDRy）。
3. 如果在 A/D 转换结束后将 ADCSR.ADIE 位置 “1”（允许由 A/D 转换结束引起的 ADI 中断），就产生 ADI 中断请求。
4. 在 A/D 转换过程中，ADST 位保持 “1”，如果转换结束，此位就自动变为 “0”，A/D 转换器进入待机状态。
5. 如果在 A/D 转换过程中将 ADST 位置 “0”（停止 A/D 转换），就中止 A/D 转换，A/D 转换器进入待机状态。

选择 AN1 进行模拟输入时的运行例子如图 23.5 所示。

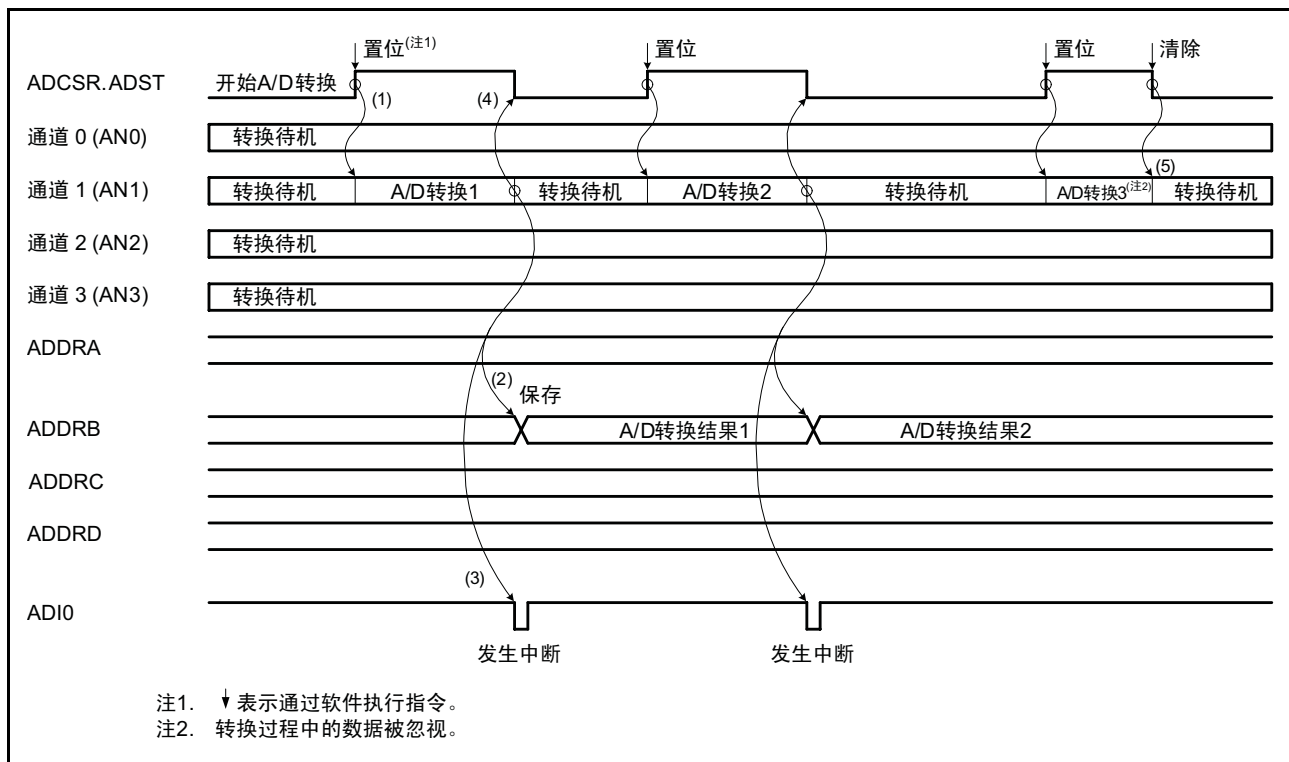


图 23.5 A/D 转换器的运行例子（单次模式）

23.3.2 扫描模式

扫描模式对最多 4 个通道的模拟输入依次连续进行 A/D 转换，如下所示。

扫描模式有重复进行 A/D 转换的连续扫描以及对指定的通道只进行 1 个周期 A/D 转换的单次扫描。

23.3.2.1 连续扫描模式

连续扫描模式对指定通道的模拟输入重复进行 A/D 转换，如下所示：

1. 如果通过软件、TPU、TMR 或者外部触发输入使 ADCSR.ADST 位变为 “1”（开始 A/D 转换），就从所选通道中通道号小的通道开始 A/D 转换。
2. 如果 A/D 转换结束，就将 A/D 转换结果保存到对应的 A/D 数据寄存器（ADDRy）。
3. 如果在所选通道的 A/D 转换全部结束后将 ADCSR.ADIE 位置 “1”（允许由 A/D 转换结束引起的 ADI 中断），就产生 ADI 中断请求。A/D 转换器就从通道号小的通道开始 A/D 转换。
4. ADST 位不自动变为 “0”，而在为 “1”（开始 A/D 转换）的期间重复 2.~3.。如果将 ADST 位置 “0”（停止 A/D 转换），就中止 A/D 转换，A/D 转换器进入待机状态。
5. 此后，如果将 ADST 位置 “1”（开始 A/D 转换），就重新从通道号小的通道开始 A/D 转换。

选择 AN0 ~ AN2 的 3 个通道进行模拟输入时的运行例子如图 23.6 所示。

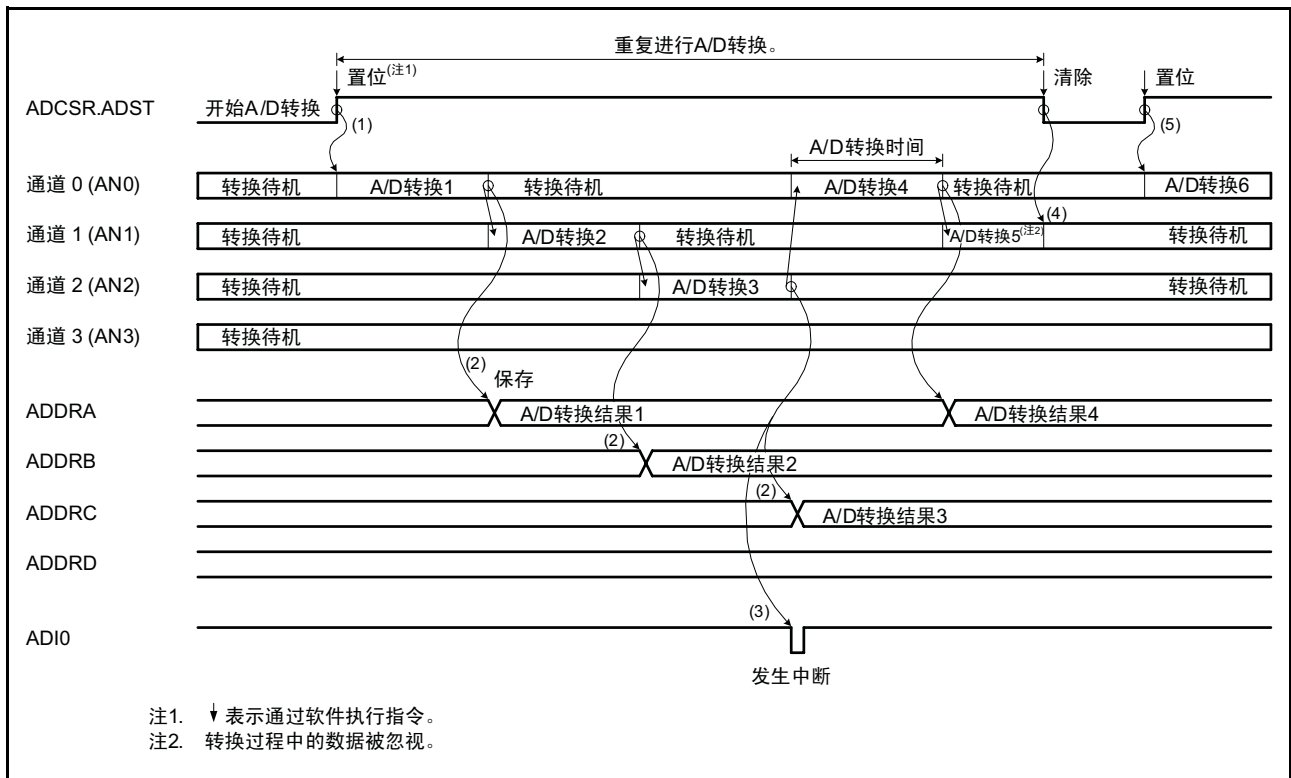


图 23.6 A/D 转换器的运行例子（连续扫描模式）

23.3.2.2 1 个周期扫描模式

1 个周期扫描模式对指定通道的模拟输入只进行 1 个周期的 A/D 转换，如下所示：

1. 通过软件、TPU、TMR 或者外部触发输入使 ADCSR.ADST 位变为 “1”（开始 A/D 转换），就从所选通道中通道号小的通道开始 A/D 转换。
2. 如果 A/D 转换结束，就将 A/D 转换结果保存到对应的 A/D 数据寄存器（ADDRy）。
3. 如果在所选通道的 A/D 转换全部结束后将 ADCSR.ADIE 位置 “1”（允许由 A/D 转换结束引起的 ADI 中断），就产生 ADI 中断请求。
4. 在 A/D 转换过程中，ADST 位保持为 “1”（开始 A/D 转换），如果所选通道的 A/D 转换全部结束，此位就自动变为 “0”，A/D 转换器进入待机状态。

选择 AN4 ~ AN6 的 3 个通道进行模拟输入时的运行例子如图 23.7 所示。

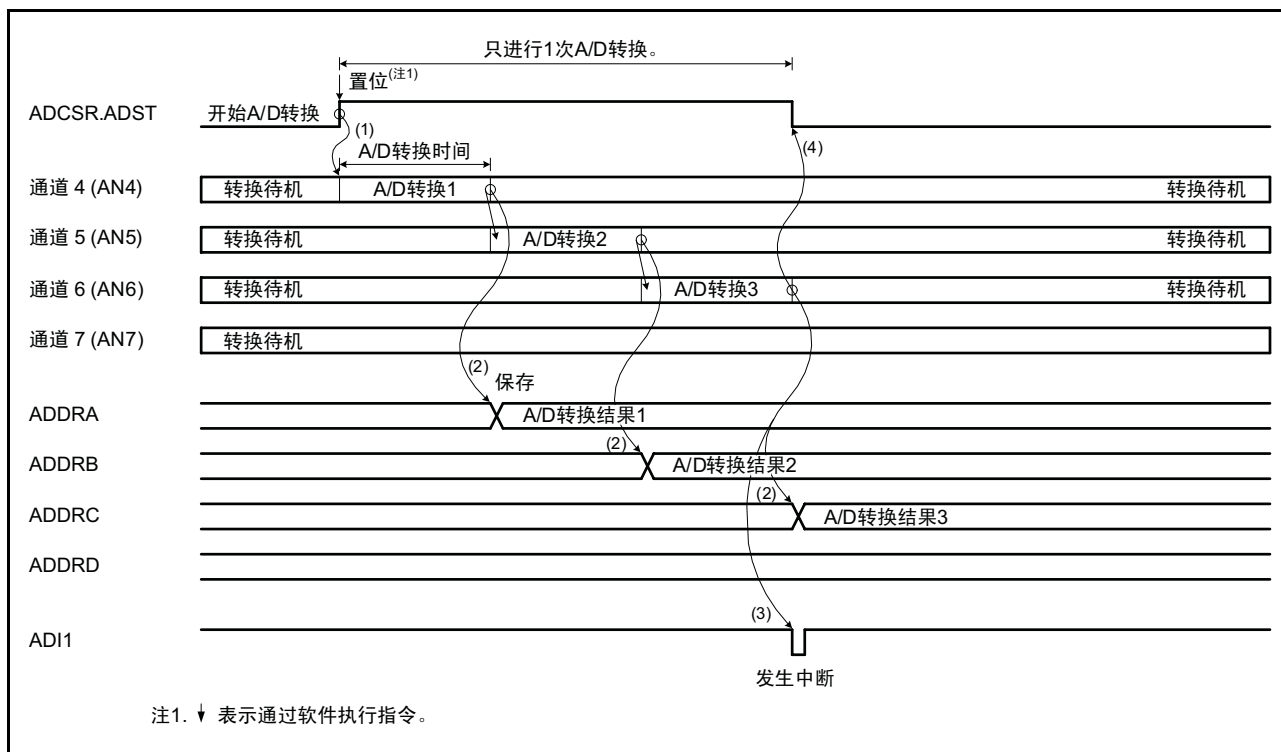


图 23.7 A/D 转换器的运行例子（1 个周期扫描模式）

23.3.3 输入采样和 A/D 转换时间

在通过软件、TPU、TMR 或者外部触发产生 A/D 转换开始条件后经过 A/D 转换开始延迟时间 (t_D)，A/D 转换器对模拟输入进行采样，然后开始 A/D 转换。

A/D 转换时序如图 23.8 所示。

A/D 转换开始条件产生后的 A/D 转换时间 (t_{CONV}) 为 t_D 、采样时间 (t_{SPL}) 和逐次转换时间 (t_{SAM}) 的总时间。此后的 A/D 转换时间 (t_{CONV}) 还包括 t_{SPL} 和 t_{SAM} 的时间。

采样时间 (t_{SPL}) 是给 A/D 转换器的采样 & 保持电路充电的时间。在模拟输入的信号源阻抗高并且采样时间不够的情况下或者在外围模块时钟 PCLK 慢的情况下，能通过 ADSSTR 寄存器调整采样时间。

逐次转换时间 (t_{SAM}) 总是固定为 25 个 ADCLK 周期。

ADSSTR 寄存器的设定例子以及 A/D 转换时间分别如表 23.6 和表 23.7 所示。

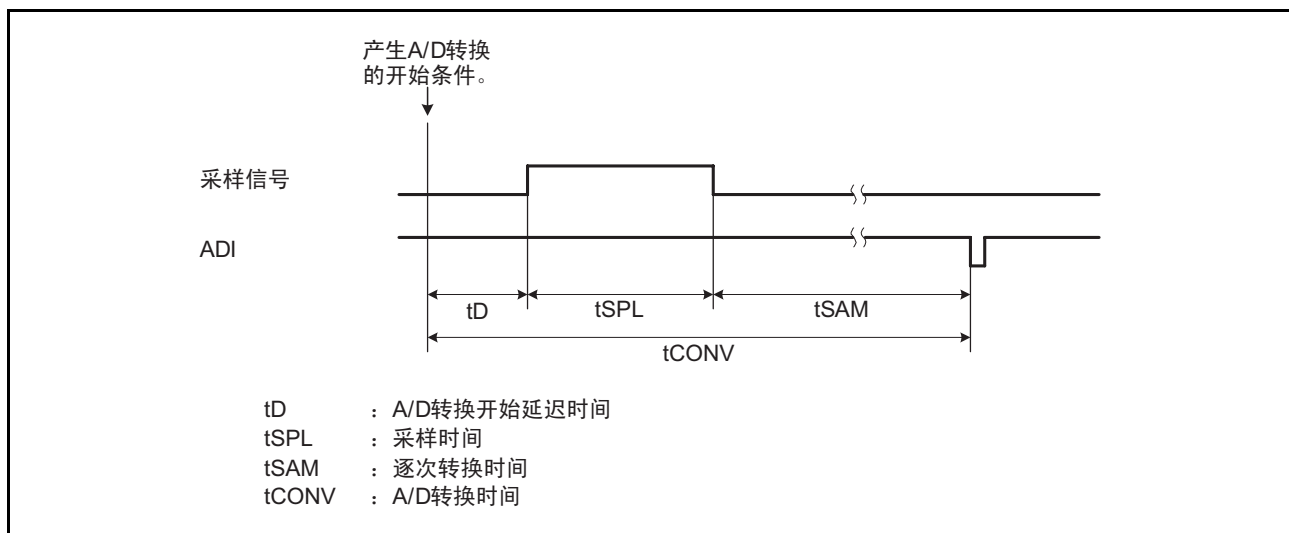


图 23.8 A/D 转换时序

表 23.6 ADSSTR 寄存器的设定例子

| 使用例子 | 设定范围 | 采样时间 (注 1) |
|---------------------------------------|---------------|--|
| 标准 (初始值) | “19h” | 0.5 μ s (当 PCLK=ADCLK=50MHz 时) |
| 在模拟输入的信号源阻抗高并且采样时间不够的情况下进行设定。 | “1Ah” ~ “FFh” | 例: “FFh” 5.1 μ s (当 PCLK=ADCLK=50MHz 时) |
| 在 ADCLK 小于 50MHz 以及采样时间比初始值短的情况下进行设定。 | “02h” ~ “18h” | 例: “14h” 0.5 μ s (当 PCLK=ADCLK=40MHz 时) |

注 1. 必须设定为采样时间 $\geq 0.5\mu$ s，用以下表达式计算采样时间：

$$\text{采样时间 } (\mu\text{s}) = \frac{\text{ADSSTR寄存器的设定值}}{\text{ADCLK (MHz)}}$$

表 23.7 A/D 转换时间

| 项目 | 符号 | 表达式 | |
|-------------------|-------|---|--|
| | | min | max |
| A/D 转换开始的延迟时间 (①) | tD | $\frac{3}{\text{PCLK (MHz)}}$ | $\frac{1}{\text{ADCLK (MHz)}} + \frac{4}{\text{PCLK (MHz)}}$ |
| 采样时间 (②) | tSPL | $\frac{\text{ADSSTR寄存器的设定值}}{\text{ADCLK (MHz)}}$ | |
| 逐次转换时间 (③) | tSAM | $\frac{25}{\text{ADCLK (MHz)}}$ | |
| A/D 转换时间 (注1) | tCONV | ① + ② + ③ | |
| A/D 转换时间 (注2) | tCONV | ② + ③ | |

注1. 单次模式或者扫描模式（第1次）的A/D转换时间

注2. 扫描模式（第2次以后）的A/D转换时间

A/D转换时间的计算例子如下所示：

当 PCLK=ADCLK=50MHz，ADSSTR=19h 并且为扫描模式（第2次）时

$$\begin{aligned}
 \text{A/D 转换时间 (tCONV)} &= \text{ADSSTR/ADCLK} + 25/\text{ADCLK} \\
 &= 25/50\text{MHz} + 25/50\text{MHz} \\
 &= 0.5\mu\text{s} + 0.5\mu\text{s} \\
 &= 1.0\mu\text{s}
 \end{aligned}$$

当 PCLK=ADCLK=40MHz，ADSSTR=14h 并且为扫描模式（第1次：min）时

$$\begin{aligned}
 \text{A/D 转换时间 (tCONV)} &= 3/\text{PCLK} + \text{ADSSTR/ADCLK} + 25/\text{ADCLK} \\
 &= 3/40\text{MHz} + 20/40\text{MHz} + 25/40\text{MHz} \\
 &= 0.075\mu\text{s} + 0.5\mu\text{s} + 0.625\mu\text{s} \\
 &= 1.2\mu\text{s}
 \end{aligned}$$

23.3.4 通过外部触发的启动

能通过外部触发的输入（ADTRG0# ~ ADTRG3#）开始各单元的 A/D 转换。

在单元 0 的情况下，如果将 AD0.ADCR.TRGS[2:0] 位置“011b”（ADTRG0# 的触发），就在 ADTRG0# 的下降沿 AD0.ADCSR.ADST 位变为“1”（开始 A/D 转换），开始 A/D 转换，此时序如图 23.9 所示。

能通过外部触发的输入同时开始 2 个单元的 A/D 转换（同时启动单元）。在单元 0 和单位 1 的情况下，如果将 AD0.ADCR.TRGS[2:0] 位置“011b”（ADTRG0# 的触发）并且将 AD1.ADCR.TRGS[2:0] 位置“111b”（ADTRG0# 的触发），就能在 ADTRG0# 的下降沿同时启动单元 0 和单元 1。同样地，能在 ADTRG2# 的下降沿同时启动单元 2 和单元 3。

必须注意：在使用外部触发时，如果外部触发的输入已经为 Low 电平，内部信号就可能产生下降沿并开始 A/D 转换。

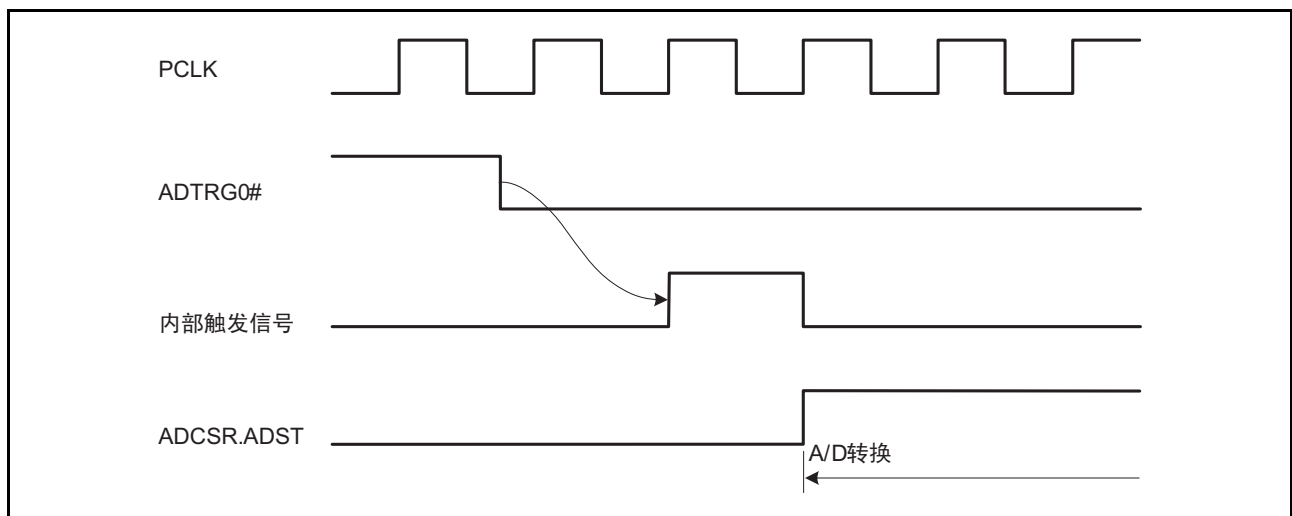


图 23.9 外部触发输入时序

23.3.5 通过 TPU0 的比较匹配 / 输入捕捉 A ~ D 的启动

能通过 TPU0 的比较匹配 / 输入捕捉 A ~ D 开始单元 0 ~ 3 的 A/D 转换。

TPU0 的比较匹配 / 输入捕捉 A ~ D 和各单元的关系如图 23.10 所示。

能通过 TPU0 的比较匹配 / 输入捕捉 A ~ D 开始最多 4 个单元的 A/D 转换。在单元 0 ~ 3 的情况下，如果将 AD0.ADCR.TRGS[2:0] 位置 “100b” (TPU0 的比较匹配 / 输入捕捉 A)，将 AD1.ADCR.TRGS[2:0] 位置 “100b” (TPU0 的比较匹配 / 输入捕捉 B)，将 AD2.ADCR.TRGS[2:0] 位置 “100b” (TPU0 的比较匹配 / 输入捕捉 C) 并且将 AD3.ADCR.TRGS[2:0] 位置 “100b” (TPU0 的比较匹配 / 输入捕捉 D)，就能通过 TPU0 的比较匹配 / 输入捕捉 A ~ D 开始单元 0 ~ 3 的 A/D 转换。

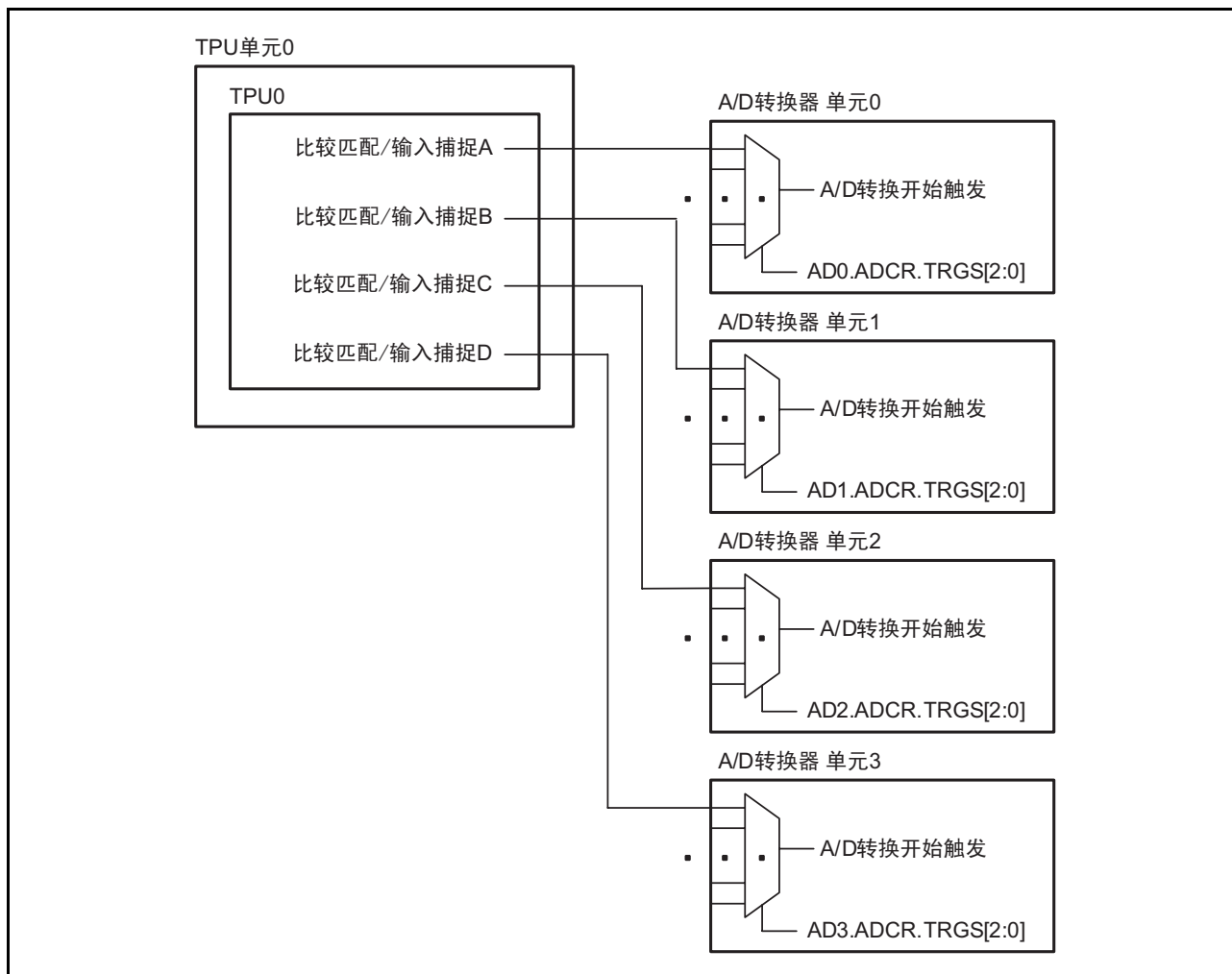


图 23.10 TPU0 的比较匹配 / 输入捕捉 A ~ D 和各单元的关系

23.3.6 通过 TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A 的启动

能通过 TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A 开始单元 0 ~ 3 的 A/D 转换。同样地，能通过 TPU6 ~ TPU11 的比较匹配 / 输入捕捉 A 开始单元 0 ~ 3 的 A/D 转换。

比较匹配 / 输入捕捉 A 和各单元的关系如图 23.11 所示。

能通过 TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A 同时开始最多 4 个单元的 A/D 转换。在通过 TPU0 和 TPU2 的比较匹配 / 输入捕捉 A 开始单元 0 ~ 3 的转换时，如果将 ADn.ADCR.TRGS[2:0] 位 (n=0 ~ 3) 置 “001b” (TPU0 ~ TPU5 的比较匹配 / 输入捕捉 A) 并且将 TPU0.TIER.TTGE 位和 TPU2.TIER.TTGE 位置 “1”，就能通过 TPU0 和 TPU2 的比较匹配 / 输入捕捉 A 同时开始单元 0 ~ 3 的 A/D 转换。

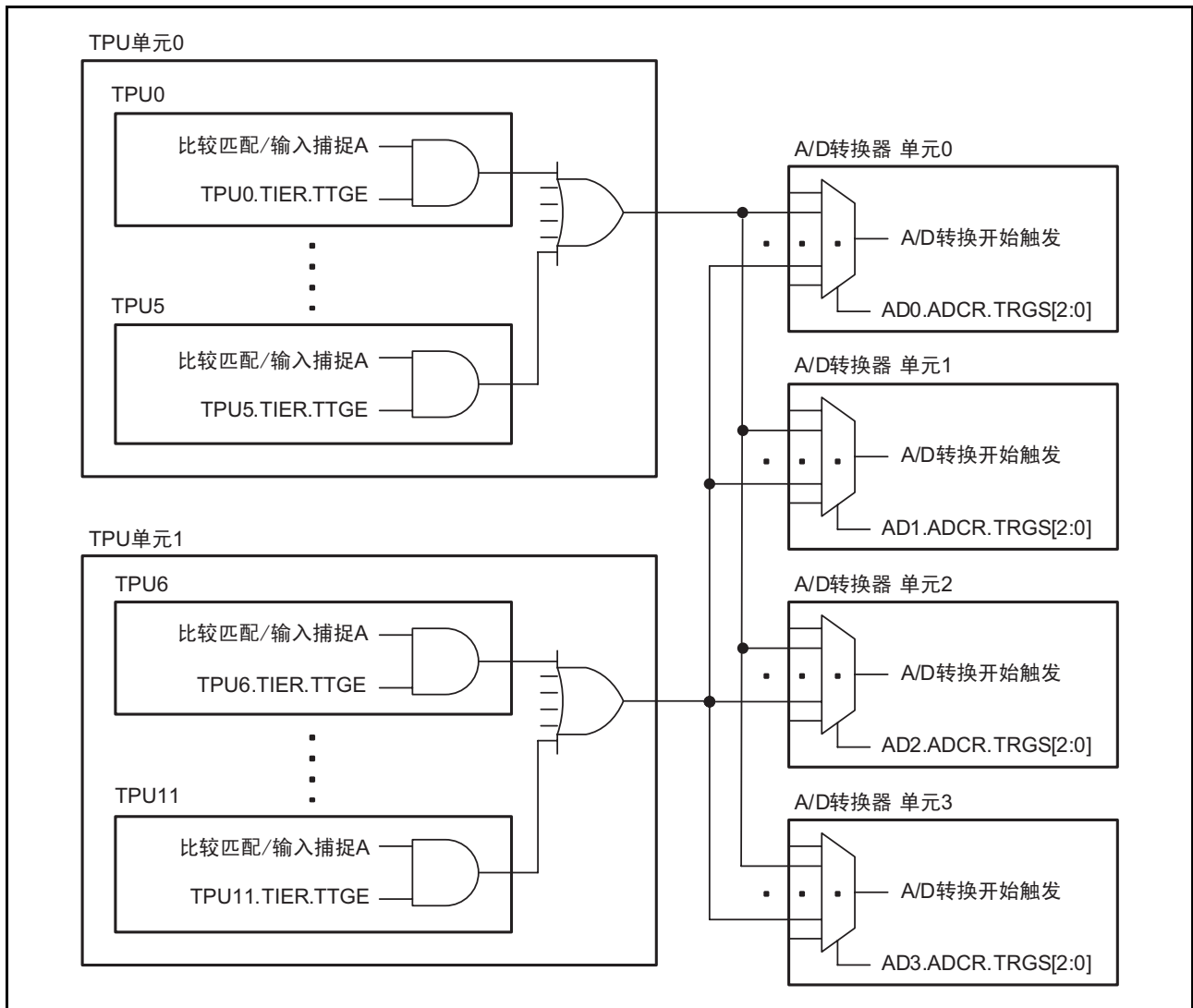


图 23.11 比较匹配 / 输入捕捉 A 和各单元的关系

23.3.7 通过 TMR 的比较匹配的启动

能通过 TMR0 的比较匹配 A 开始单元 0 和单元 1 的 A/D 转换。

同样地，能通过 TMR2 的比较匹配 A 开始单元 2 和单元 3 的 A/D 转换。

TMR0 和 TMR2 的比较匹配 A 和各单元的关系如图 23.12 所示。

能通过 TMR0 的比较匹配 A 同时开始最多 2 个单元的 A/D 转换。在单元 0 和单元 1 的情况下，如果将 ADn.ADCR.TRGS[2:0] 位 (n=0,1) 置“010b”(TMR0 的比较匹配 A) 并且将 TMR0.TCSR.ADTE 位置“1”，就能通过 TMR0 的比较匹配 A 同时开始单元 0 和单元 1 的 A/D 转换。

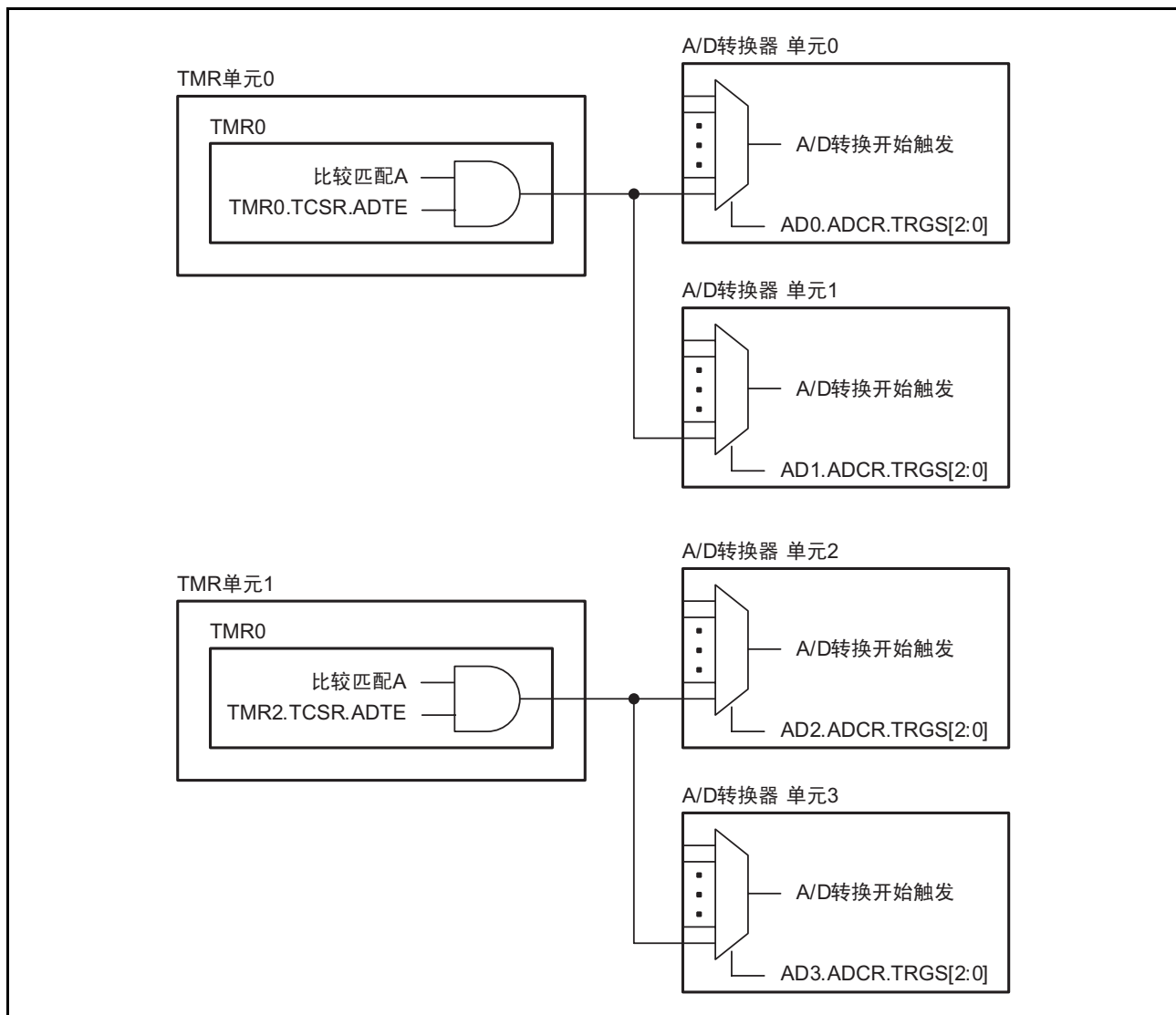


图 23.12 TMR0、TMR2 的比较匹配 A 和各单元的关系

23.4 中断源

当 ADCSR.ADIE 位为 “1”（允许由 A/D 转换结束引起的 ADI 中断）时，如果 A/D 转换结束，A/D 转换器就产生中断（ADI）。

各单元有 A/D 转换器的中断源，能使用这些中断启动数据传送控制器（DTC）和 DMA 控制器（DMAC）。如果通过 ADI 中断启动 DTC 或者 DMAC，并且读被转换的数据，就能不通过 CPU 进行连续的转换。

表 23.8 A/D 转换器的中断源

| 名称 | 中断源 | 中断状态标志 | DTC 的启动 | DMAC 的启动 |
|------|----------|--------------|---------|----------|
| ADI0 | A/D 转换结束 | ICU.IR98.IR | 能 | 能 |
| ADI1 | A/D 转换结束 | ICU.IR99.IR | 能 | 能 |
| ADI2 | A/D 转换结束 | ICU.IR100.IR | 能 | 能 |
| ADI3 | A/D 转换结束 | ICU.IR101.IR | 能 | 能 |

23.5 A/D 转换精度的定义

A/D 转换精度的定义如下：

- 分辨率
这是指 A/D 转换器的数字输出码数。
- 量化误差
量化误差是 A/D 转换器固有的偏差，为 1/2LSB（图 23.13）。
- 偏移误差
这是在数字输出从最小电压值 “000000000b（000h）” 变为 “000000001b（001h）” 时与模拟输入电压值的理想 A/D 转换特性的偏差（图 23.14）。
- 满刻度误差
这是在数字输出从 “111111110b（3FEh）” 变为 “111111111b（3FFh）” 时与模拟输入电压值的理想 A/D 转换特性的偏差（图 23.14）。
- 非线性误差
这是在零电压到满刻度电压之间的理想 A/D 转换特性的误差，但是不包含偏移误差、满刻度误差和量化误差（图 23.14）。
- 绝对精度
这是数字值和模拟输入值的偏差，包含偏移误差、满刻度误差、量化误差和非线性误差。

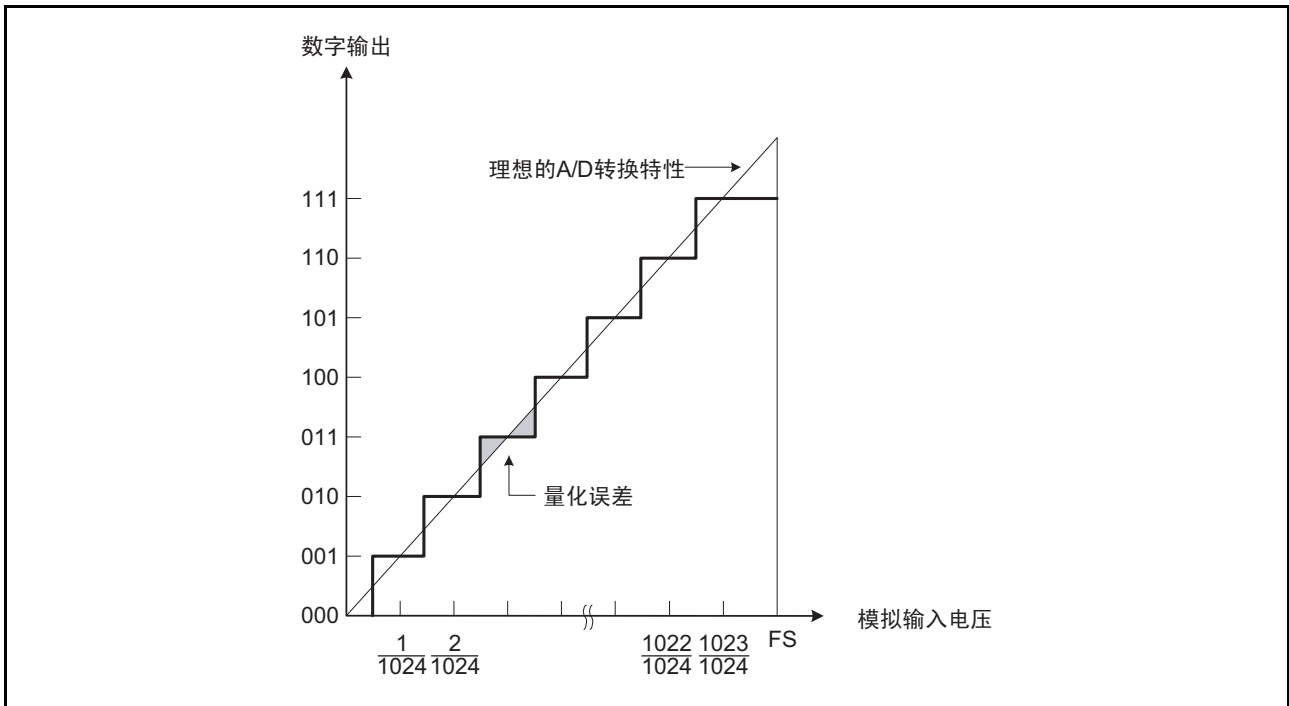


图 23.13 A/D 转换精度的定义 (1)

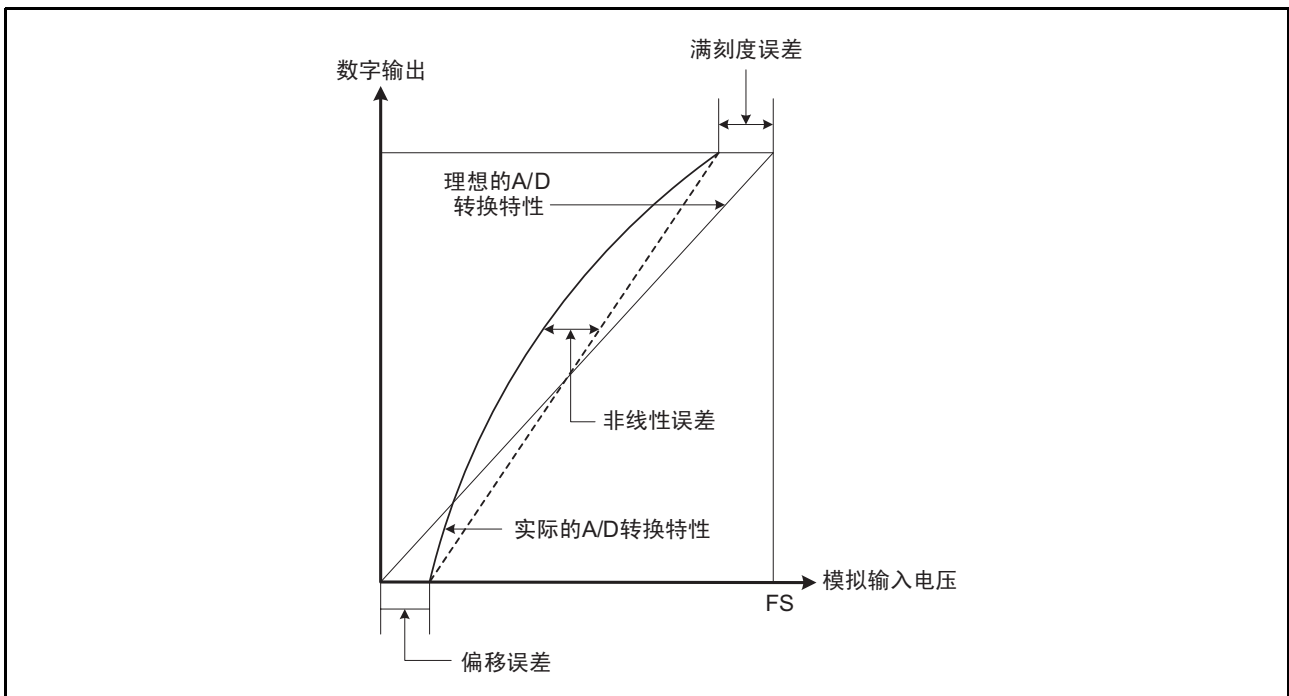


图 23.14 A/D 转换精度的定义 (2)

23.6 使用时的注意事项

23.6.1 模块停止功能的设定

能通过模块停止控制寄存器设定禁止或者允许各单元的 A/D 转换器的运行，初始值为停止 A/D 转换器的运行。能通过解除模块停止状态，使寄存器变为可存取的状态。详细内容请参照“8. 低功耗功能”。

23.6.2 停止 A/D 转换时的注意事项

在选择外部触发或者定时器为 A/D 转换开始条件时，为了停止 A/D 转换，必须在将 ADCR.TRGS[2:0] 位置“000b”并且选择软件触发为 A/D 转换开始条件后，将 ADCSR.ADST 位置“0”（停止 A/D 转换）。

23.6.3 重新开始 A/D 转换时的注意事项

如果通过将 ADCSR.ADST 位置“0”来停止 A/D 转换，就需要 1 个 ADCLK 周期的时间，停止 A/D 转换器模拟电路。

如果通过将 ADST 位置“0”后立即将 ADST 位置“1”来重新开始 A/D 转换，就在经过 1 个 ADCLK 周期的时间后，重新开始 A/D 转换。

23.6.4 向低功耗状态转移时的注意事项

在允许 A/D 转换的状态下，如果 LSI 向模块停止状态或者软件待机模式转移，模拟电源的电流就和 A/D 转换过程中的电流相同。如果需要在模块停止状态或者软件待机模式中减少模拟电源电流，就必须停止 A/D 转换。在停止 A/D 转换时，需要确保从 ADCSR.ADST 位被置“0”后到 A/D 转换器的模拟电路停止前的时间。为了确保此时间，必须按照以下的步骤进行设定。

必须在将 ADCR.TRGS[2:0] 位置“000b”（软件触发）并且将 ADCSR.ADST 位置“0”后，将 ADCR.CKS[1:0] 位置“11b”（PCLK）。此后，必须在确认 A/D 转换已经停止后转移到模块停止状态或者软件待机模式状态。

23.6.5 有关容许信号源阻抗

为了实现 $1.0\mu\text{s}$ 的高速转换，对于信号源阻抗不超过 $1.0\text{k}\Omega$ 的输入信号，LSI 的模拟输入能保证转换精度。在单次模式中进行转换并且外接大电容的情况下，因为输入负载实际上只有 $6.5\text{k}\Omega$ 的内部输入电阻，所以信号源阻抗可忽略不计。但是，由于形成低通滤波器，所以有可能无法跟踪急剧变化的模拟信号（例如变动率至少为 $5\text{mV}/\mu\text{s}$ ）（图 23.15）。在转换高速模拟信号或者在扫描模式中进行转换时，必须插入 1 个输出阻抗低的缓冲放大器。

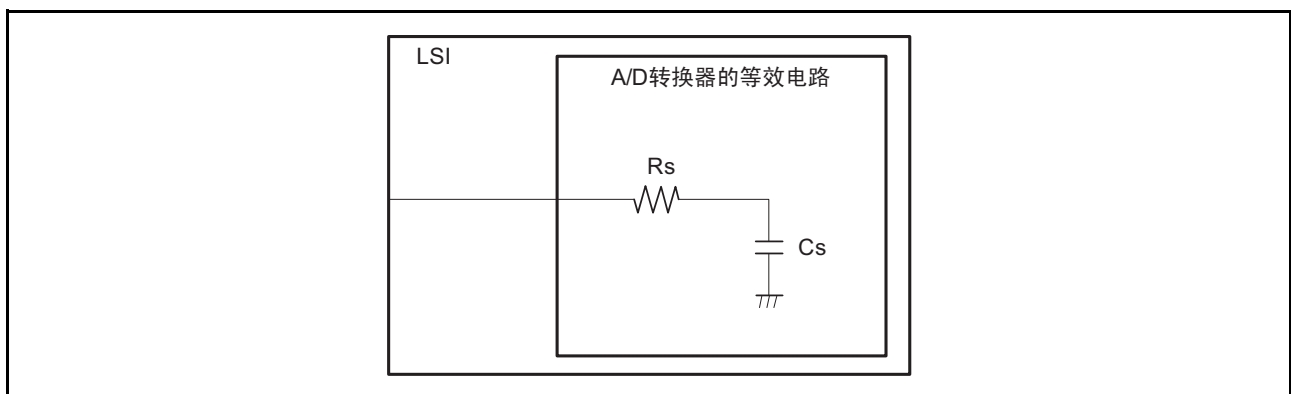


图 23.15 模拟输入引脚的内部等效电路

表 23.9 模拟引脚的规格

| 项目 | | min | max | 单位 |
|-----------|----|-----|-----|------------|
| 容许信号源阻抗 | | — | 1.0 | k Ω |
| 引脚的内部等效电路 | Rs | — | 6.5 | k Ω |
| | Cs | — | 6.0 | pF |

23.6.6 对绝对精度的影响

由于附加电容会导致与 GND 的耦合，如果 GND 有噪声就可能降低绝对精度，所以必须连接 VREFL 等电特性稳定的 GND。

必须注意：滤波器电路不要干扰安装电路板上的数字信号，也不要充当天线。

23.6.7 模拟电源引脚等的设定范围

- 模拟输入电压的设定范围
必须将模拟输入引脚 ANn 的外加电压设定在 $VREFL \leq VAN \leq VREFH$ 的范围内。
- 各电源引脚（AVCC-AVSS、VREFH-VREFL、VCC-VSS）的关系
AVCC、AVSS 和 VCC、VSS 的关系必须为 $AVCC=VCC$ 并且 $AVSS=VSS$ 。如图 23.16 所示，为了能形成最短的闭环，必须在各电源间连接 $0.1\mu\text{F}$ 的电容器，并且在供电处使 $AVCC=VCC$ 并且 $VREFL=AVSS=VSS$ 。在不使用 A/D 转换器时，必须使 $VREFH=AVCC=VCC$ 并且 $VREFL=AVSS=VSS$ 。
- VREFH 的设定范围
VREFH 引脚的基准电压的设定范围必须为 $VREFH \leq AVCC$ 。

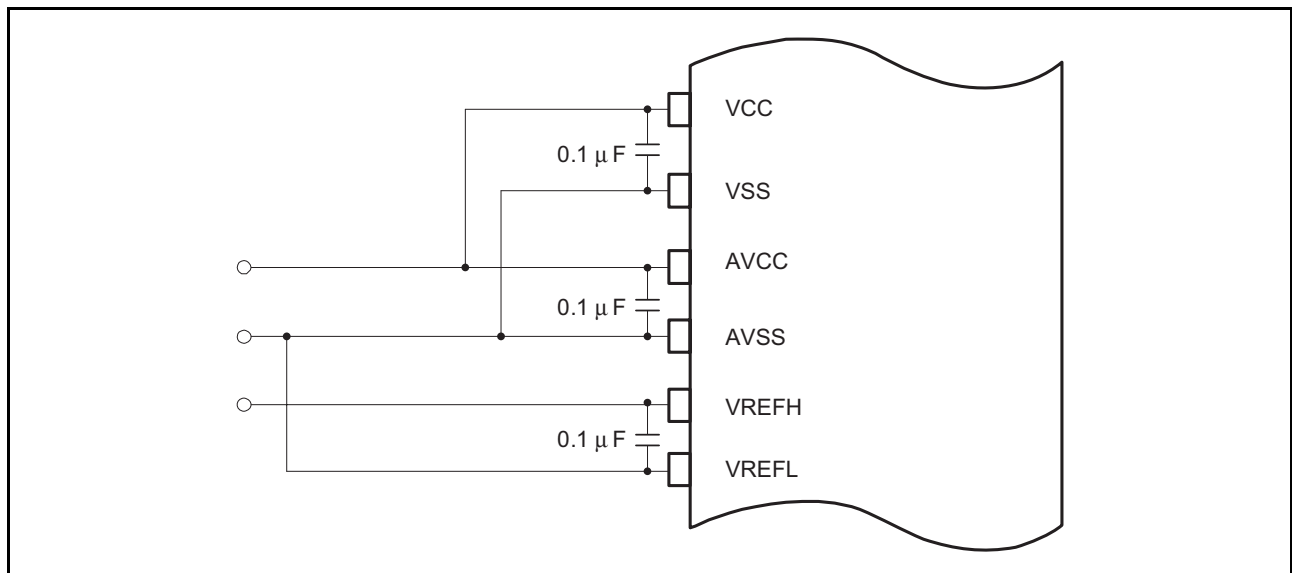


图 23.16 各电源引脚的连接例子

23.6.8 设计电路板时的注意事项

在设计电路板时，必须尽量将数字电路和模拟电路分开，而且不能使数字电路的信号线和模拟电路的信号线交叉或者靠近，否则会给模拟信号带来噪声并且给 A/D 转换值带来不良影响。必须通过模拟接地（VREFL）将模拟输入引脚（AN0 ~ AN15）、模拟基准电源（VREFH）和模拟电源电压（AVCC）与数字电路分开，而且模拟接地（VREFL）必须与电路板上稳定的接地（VSS）进行单点连接。

23.6.9 噪声对策的注意事项

如图 23.17 所示，为了防止过大电涌等异常电压对模拟输入引脚（AN0 ~ AN15）造成破坏，必须在 AVCC 和 AVSS 之间以及 VREFH 和 VREFL 之间连接电容器，并且以模拟输入引脚（AN0 ~ AN15）为基准连接保护电路。

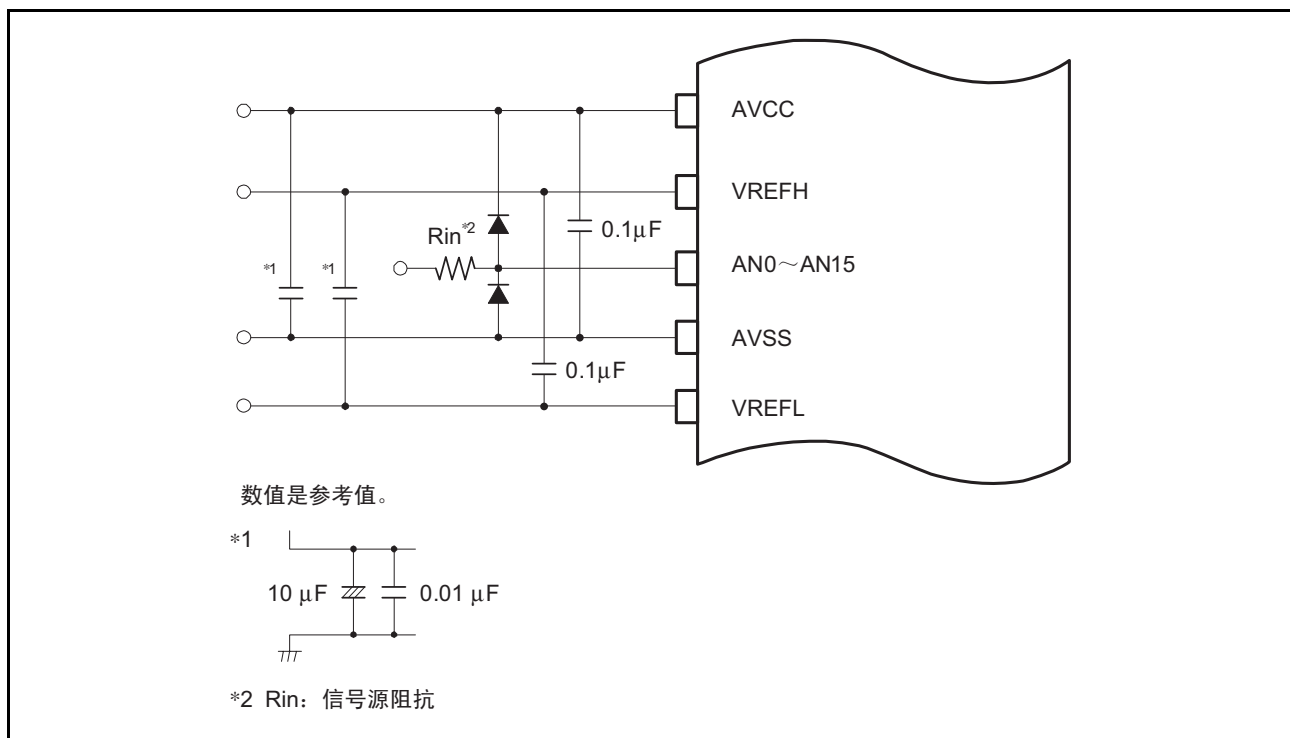


图 23.17 模拟输入的保护电路例子

23.6.10 高速转换的实现

如图 23.18 所示，为了实现高速转换，必须在模拟输入引脚（AN0 ~ AN15）和 VREFL 之间外接 0.1 μ F 的电容器。但是，对于 A/D 转换器的采样 & 保持电路的输入电容，为了隐藏信号源阻抗，需要在开始转换前给外接电容器积累足够的电荷。如果由于扫描等模拟输入引脚的电压电平发生变动而引起外接电容器的电荷变化，就不能实现高速转换。

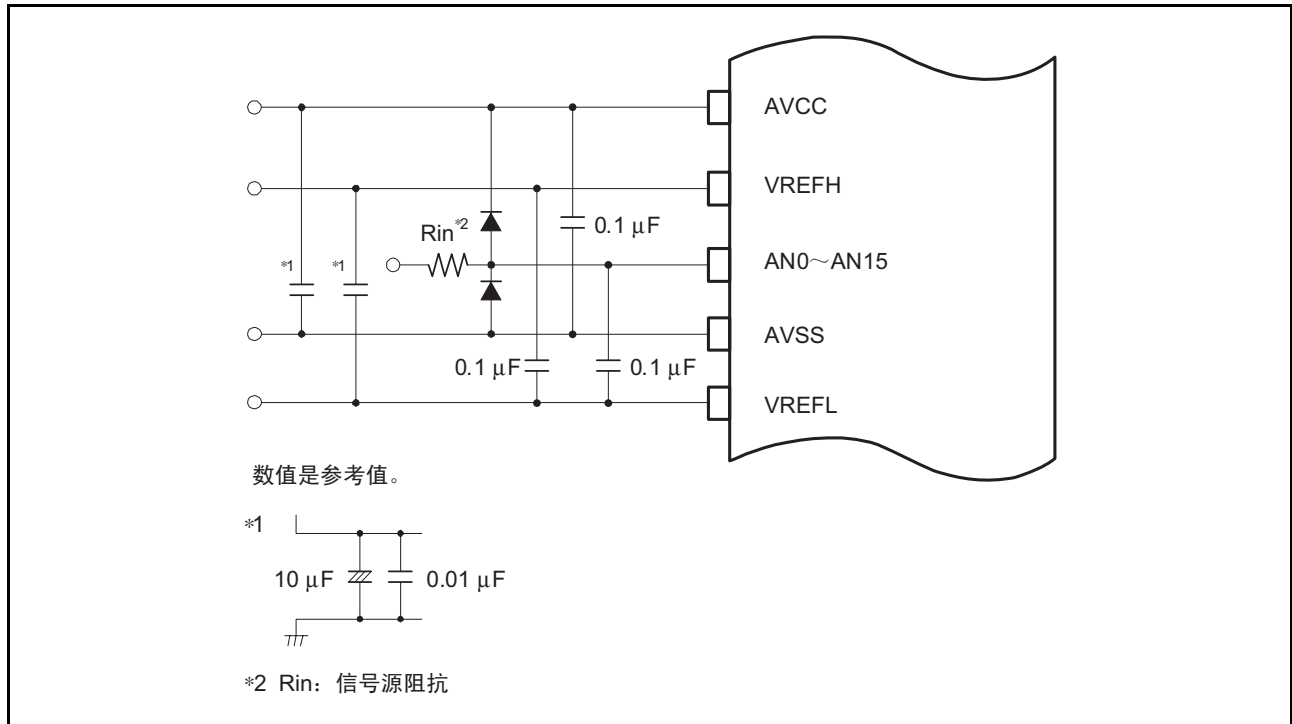


图 23.18 用于高速转换的外接电容器连接例子

24. D/A 转换器

24.1 概要

RX610 群内置 2 个通道的 10 位 D/A 转换器。

D/A 转换器的规格和框图分别如表 24.1 和图 24.1 所示。

表 24.1 D/A 转换器的规格

| 项目 | 内容 |
|-------|-------------|
| 分辨率 | 10 位 |
| 输出通道 | 2 个通道 |
| 低功耗功能 | 能设定为模块停止状态。 |

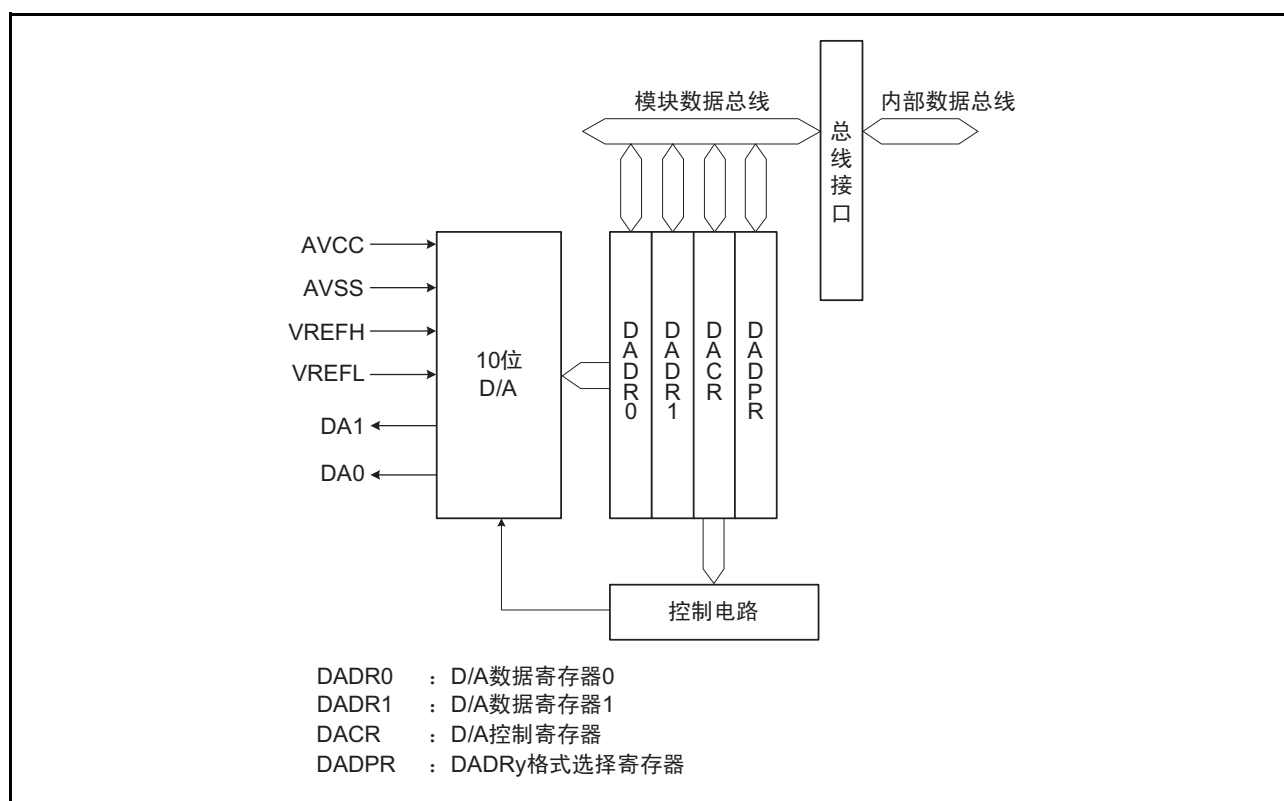


图 24.1 D/A 转换器的框图

D/A 转换器使用的输入 / 输出引脚如表 24.2 所示。

表 24.2 D/A 转换器的输入 / 输出引脚

| 引脚名 | 输入 / 输出 | 功能 |
|-------|---------|------------------------------------|
| AVCC | 输入 | 模拟电路的电源引脚 |
| AVSS | 输入 | 模拟电路的接地引脚 |
| VREFH | 输入 | D/A 转换器的基准电源引脚 |
| VREFL | 输入 | D/A 转换器的基准接地引脚 必须连接模拟基准电源 (0V)。 |
| DA0 | 输出 | 通道 0 的模拟输出 |
| DA1 | 输出 | 通道 1 的模拟输出 |

24.2 寄存器说明

D/A 转换器的寄存器一览表如表 24.3 所示。

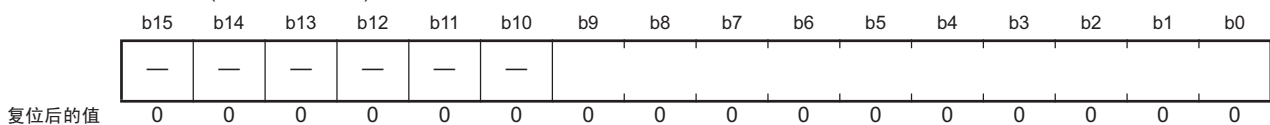
表 24.3 D/A 转换器的寄存器一览表

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|---------------|-------|-------|------------|------|
| D/A 数据寄存器 0 | DADR0 | 0000h | 0008 80C0h | 16 |
| D/A 数据寄存器 1 | DADR1 | 0000h | 0008 80C2h | 16 |
| D/A 控制寄存器 | DACR | 1Fh | 0008 80C4h | 8 |
| DADRy 格式选择寄存器 | DADPR | 00h | 0008 80C5h | 8 |

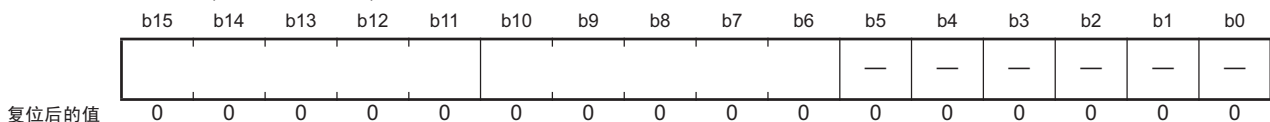
24.2.1 D/A 数据寄存器 y (DADRy) (y=0,1)

地址 DADR0 0008 80C0h、DADR1 0008 80C2h

- DADPR.DPSEL位=0(数据往LSB靠紧)



- DADPR.DPSEL位=1(数据往MSB靠紧)



DADRy 寄存器是保存 D/A 转换数据的 16 位可读写寄存器。如果允许模拟输出，就转换 DADRy 寄存器的值并且输出到模拟输出引脚。

能通过设定 DADPR.DPSEL 位，更改 10 位数据的排列。“—”的位的读写值都为“0”。

24.2.2 D/A 控制寄存器 (DACR)

地址 0008 80C4h

| | | | | | | | | |
|-------|-------|-------|-----|----|----|----|----|----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | DAOE1 | DAOE0 | DAE | — | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|--------------|--|-----|
| b4-b0 | — | 保留位 | 读写值都为“1”。 | |
| b5 | DAE (注1) | D/A 允许位 | 0: 分别控制通道 0 和通道 1 的 D/A 转换 1: 同时允许通道 0 和通道 1 的 D/A 转换 | R/W |
| b6 | DAOE0 | D/A 输出允许 0 位 | 0: 禁止通道 0 的模拟输出 (DA0) 1: 允许通道 0 的 D/A 转换 允许通道 0 的模拟输出 (DA0) (注2) | R/W |
| b7 | DAOE1 | D/A 输出允许 1 位 | 0: 禁止通道 1 的模拟输出 (DA1) 1: 允许通道 1 的 D/A 转换 允许通道 1 的模拟输出 (DA1) (注2) | R/W |

注 1. 通过和 DAOEj 位 (j=0,1) 的组合对 D/A 转换进行控制, 由 DAOEj 位控制转换结果的输出, 详细内容请参照表 24.4。

注 2. 必须将用作模拟输出引脚的 P6.DDR.Bj 位 (j=7,6) 和 P6.ICR.Bj 位 (j=7,6) 都置“0”, 详细内容请参照“14. I/O 端口”。

表 24.4 D/A 转换的控制

| b5 | b7 | b6 | 说明 |
|-----|-------|-------|--|
| DAE | DAOE1 | DAOE0 | |
| 0 | 0 | 0 | 禁止 D/A 转换。 |
| | | 1 | 允许通道 0 的 D/A 转换, 禁止通道 1 的 D/A 转换。 允许通道 0 的模拟输出 (DA0), 禁止通道 1 的模拟输出 (DA1)。 |
| | 1 | 0 | 禁止通道 0 的 D/A 转换, 允许通道 1 的 D/A 转换。 禁止通道 0 的模拟输出 (DA0), 允许通道 1 的模拟输出 (DA1)。 |
| | | 1 | 允许通道 0 和通道 1 的 D/A 转换。 允许通道 0 和通道 1 的模拟输出 (DA0 和 DA1)。 |
| 1 | 0 | 0 | 允许通道 0 和通道 1 的 D/A 转换。 禁止通道 0 和通道 1 的模拟输出 (DA0 和 DA1)。 |
| | | 1 | 允许通道 0 和通道 1 的 D/A 转换。 允许通道 0 的模拟输出 (DA0), 禁止通道 1 的模拟输出 (DA1)。 |
| | 1 | 0 | 允许通道 0 和通道 1 的 D/A 转换。 禁止通道 0 的模拟输出 (DA0), 允许通道 1 的模拟输出 (DA1)。 |
| | | 1 | 允许通道 0 和通道 1 的 D/A 转换。 允许通道 0 和通道 1 的模拟输出 (DA0 和 DA1)。 |

DACR 寄存器是控制 D/A 转换器运行的寄存器。

DAE 位 (D/A 允许位)

此位通过和 DAOEj 位 (j=0,1) 的组合对 D/A 转换进行控制。

当 DAE 位为“0”时, 分别控制通道 0 和通道 1 的 D/A 转换; 当 DAE 位为“1”时, 同时控制通道 0 和通道 1 的 D/A 转换, 由 DAOEi 位控制转换结果的输出。

DAOE0 位 (D/A 输出允许 0 位)

此位控制 D/A 转换和模拟输出。

DAOE1 位 (D/A 输出允许 1 位)

此位控制 D/A 转换和模拟输出。

24.2.3 DADRy 格式选择寄存器 (DADPR)

地址 0008 80C5h

| | | | | | | | | |
|-------|-------|----|----|----|----|----|----|----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | DPSEL | — | — | — | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|-------------|--|-----|
| b6-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | DPSEL | DADRy 格式选择位 | 0: D/A 数据寄存器的数据往 LSB 靠紧 1: D/A 数据寄存器的数据往 MSB 靠紧 | R/W |

DADPR 寄存器选择 D/A 数据寄存器的数据排列。

DPSEL 位 (DADRy 格式选择位)

此位选择 D/A 数据寄存器的数据是往 LSB 靠紧还是往 MSB 靠紧。

24.3 运行说明

2 个通道的 D/A 转换器能独立进行转换。如果将 DACR.DA0E_i 位 (i=0,1) 置 “1”，就允许 D/A 转换并且输出转换结果。

进行通道 0 的 D/A 转换时的运行例子如下所示，此时的运行时序如图 24.2 所示。

1. 给 DADR0 寄存器写转换数据。
2. 如果将 DACR.DA0E0 位置 “1”，就开始进行 D/A 转换。在经过 t_DCONV 时间后，从模拟输出引脚 DA0 输出转换结果。在改写 DADR0 寄存器或者将 DA0E0 位置 “0” 前，连续输出转换结果。用以下的表达式计算输出值：

$$\frac{\text{DADR0寄存器的值}}{1024} \times V_{\text{REFH}}$$

3. 如果改写 DADR0 寄存器，就立即开始转换。在经过 t_DCONV 时间后，输出转换结果。
4. 如果将 DA0E0 位置 “0”，就禁止模拟输出。

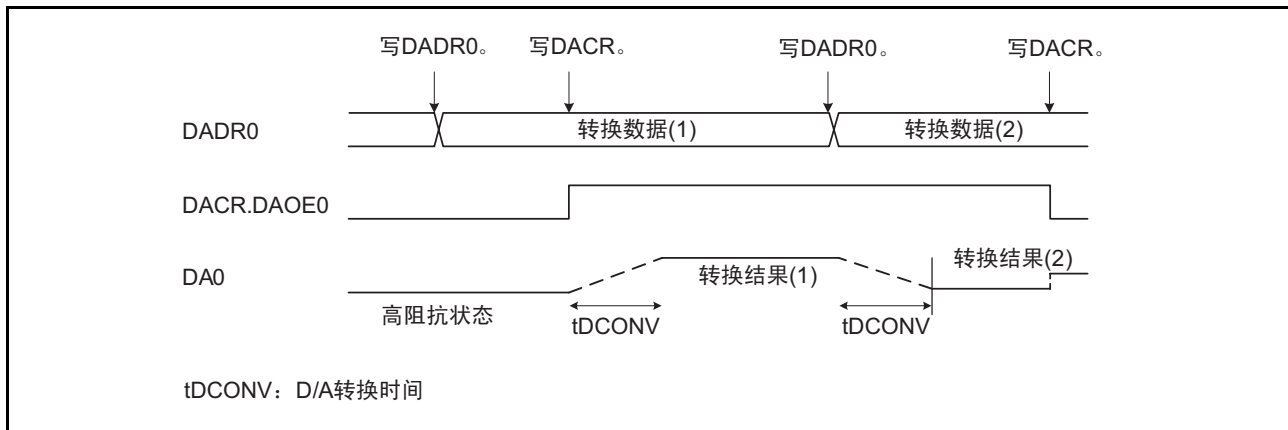


图 24.2 D/A 转换器的运行例子

24.4 使用时的注意事项

24.4.1 模块停止功能的设定

能通过模块停止控制寄存器设定禁止或者允许 D/A 转换器的运行，初始值为停止 D/A 转换器的运行。能通过解除模块停止状态，使寄存器变为可存取的状态。详细内容请参照 “8. 低功耗功能”。

24.4.2 模块停止时的 D/A 转换器的运行

如果在允许 D/A 转换的状态下 LSI 变为模块停止状态，就保持 D/A 输出，模拟电源电流和 D/A 转换过程中的电流相同。如果需要在模块停止时减少模拟电源电流，就必须在将 DACR.DA0E1 位、DACR.DA0E0 位和 DACR.DAE 位全部置 “0” 后禁止 D/A 输出。

24.4.3 软件待机模式时的 D/A 转换器的运行

如果在允许 D/A 转换的状态下 LSI 转移到软件待机模式，就保持 D/A 输出，模拟电源电流和 D/A 转换过程中的电流相同。如果需要在软件待机模式中减少模拟电源电流，就必须在将 DACR.DA0E1 位、DACR.DA0E0 位和 DACR.DAE 位全部置 “0” 后禁止 D/A 输出。

24.4.4 深度软件待机模式时的注意事项

如果在允许 D/A 转换的状态下 LSI 转移到深度软件待机模式，D/A 输出就变为高阻抗状态。

25. RAM

RX610 群内置高速静态 RAM。

25.1 概要

RAM 的规格如表 25.1 所示。

表 25.1 RAM 的规格

| 项目 | 内容 |
|--------|--|
| RAM 容量 | 128K 字节 (RAM0: 64K 字节, RAM1: 64K 字节) |
| RAM 地址 | RAM0: 0000 0000h ~ 0000 FFFFh RAM1: 0001 0000h ~ 0001 FFFFh |
| 存取 | <ul style="list-style-type: none"> 以 1 个周期进行读写。 可选择 RAM 的有效或者无效 (注 1)。 |
| 数据保存功能 | 在深度待机模式中, 能保持 RAM0 的数据。 |
| 低功耗功能 | RAM0 和 RAM1 能分别设定为模块停止状态。 |

注 1. 能通过 SYSCR1.RAME 位进行选择。有关 SYSCR1 寄存器, 请参照“3.2.4 系统控制寄存器 1 (SYSCR1)”。

25.2 运行说明

25.2.1 数据的保持

内部 RAM 的地址空间分为 RAM0 区和 RAM1 区, RAM0 区和 RAM1 区的不同点是在深度软件待机模式中是否能提供内部电源。

能通过设定 DPSBYCR.RAMCUTj 位 (j=0 ~ 2), 选择在深度软件待机模式中是否给 RAM0 提供内部电源。

能通过深度软件待机模式中给 RAM0 提供内部电源, 保持 RAM0 的数据。此时, 因为停止给 RAM1 提供内部电源, 所以不能保持 RAM1 的数据。

有关 DPSBYCR.RAMCUTj 位 (j=0 ~ 2) 的详细内容, 请参照“8. 低功耗功能”。

25.2.2 低功耗功能

通过设定模块停止控制寄存器 C (MSTPCRC) 停止给 RAM 提供时钟, 能降低功耗。

如果将 MSTPCRC.MSTPC0 位置“1”, 提供给 RAM0 的时钟就停止; 如果将 MSTPCRC.MSTPC1 位置“1”, 提供给 RAM1 的时钟就停止。

通过停止提供时钟, 使 RAM0 和 RAM1 各自变为模块停止状态。但是在复位后, RAM 运行。

一旦进入模块停止状态, 就无法存取 RAM。不能在存取 RAM 的过程中转移到模块停止状态。

有关 MSTPCRC 寄存器的详细内容, 请参照“8. 低功耗功能”。

26. ROM（保存代码的闪存）

RX610 群内置用于保存最多 2M 字节代码的闪存（ROM）以及用于保存 32K 字节数据的闪存（数据闪存）。本章说明用于保存代码的闪存，有关数据闪存请参照“27. 数据闪存（保存数据的闪存）”。

26.1 概要

ROM 的规格如表 26.1 所示，ROM 和数据闪存外围的框图如图 26.1 所示。

表 26.1 ROM 的规格

| 项目 | | 内容 |
|--------------------|--------|---|
| 2 种存储器 MAT | | <ul style="list-style-type: none"> • 用户 MAT : 2M 字节、1.5M 字节、1M 字节或者 768K 字节（注 1） • 用户引导 MAT : 16K 字节 |
| 高速读取能力 | | 能进行 1 个 ICLK 周期的高速读取。 |
| 编程 / 擦除方式 | | <ul style="list-style-type: none"> • 内置对 ROM/ 数据闪存进行改写的专用定序器（FCU）。 • 能通过给 FCU 发行命令，进行 ROM/ 数据闪存的编程或者擦除。 |
| BGO（后台操作）功能 | | <ul style="list-style-type: none"> • 在对 ROM 进行编程或者擦除的期间，CPU 能执行非 ROM 区和非数据闪存区的程序。 • 在对数据闪存进行编程或者擦除的期间，CPU 能执行 ROM 区的程序。 |
| 挂起 / 恢复功能 | | <ul style="list-style-type: none"> • 中止 ROM 的编程或者擦除，CPU 能执行 ROM 区的程序（挂起）。 • 在中止后，能重新开始对 ROM 进行编程或者擦除（恢复）。 |
| 编程 / 擦除单位 | | <ul style="list-style-type: none"> • 用户 MAT 和用户引导 MAT 的编程单位：256 字节 • 用户 MAT 的擦除单位：8K 字节（8 块）、64K 字节（9 块）、128K 字节（11 块） • 用户引导 MAT 的擦除单位：16K 字节 |
| 板上编程 （3 种） | 引导模式 | <ul style="list-style-type: none"> • 能使用 SCI 改写用户 MAT 和用户引导 MAT。 • 能自动调整主机和 RX610 之间的 SCI 通信的位速率。 |
| | 用户引导模式 | 能从用户引导 MAT 启动并且改写用户 MAT。 |
| | 用户编程 | 能通过程序改写用户 MAT。 |
| 板外编程 | | 能使用 PROM 编程器改写用户 MAT 和用户引导 MAT。 |
| 保护功能 | 软件保护功能 | 能通过 FENTRYR.FENTRY1 位（注 2）、FENTRYR.FENTRY0 位、FWEPROR.FLWE[1:0] 位和锁定位防止意料不到的改写。 |
| | 错误保护功能 | 如果在编程或者擦除过程中检测到异常，就禁止以后的编程或者擦除处理。 |
| 编程时间 / 擦除时间 / 改写次数 | | 请参照“29. 电特性”。 |

注 1. ROM 容量因产品种类而不同。

| 产品型号 | ROM 容量 | ROM 地址 |
|----------|---------|-------------------------|
| R5F56108 | 2M 字节 | FFE0 0000h ~ FFFF FFFFh |
| R5F56107 | 1.5M 字节 | FFE8 0000h ~ FFFF FFFFh |
| R5F56106 | 1M 字节 | FFF0 0000h ~ FFFF FFFFh |
| R5F56104 | 768K 字节 | FFF4 0000h ~ FFFF FFFFh |

注 2. ROM 容量小于等于 1M 字节的产品不能使用。

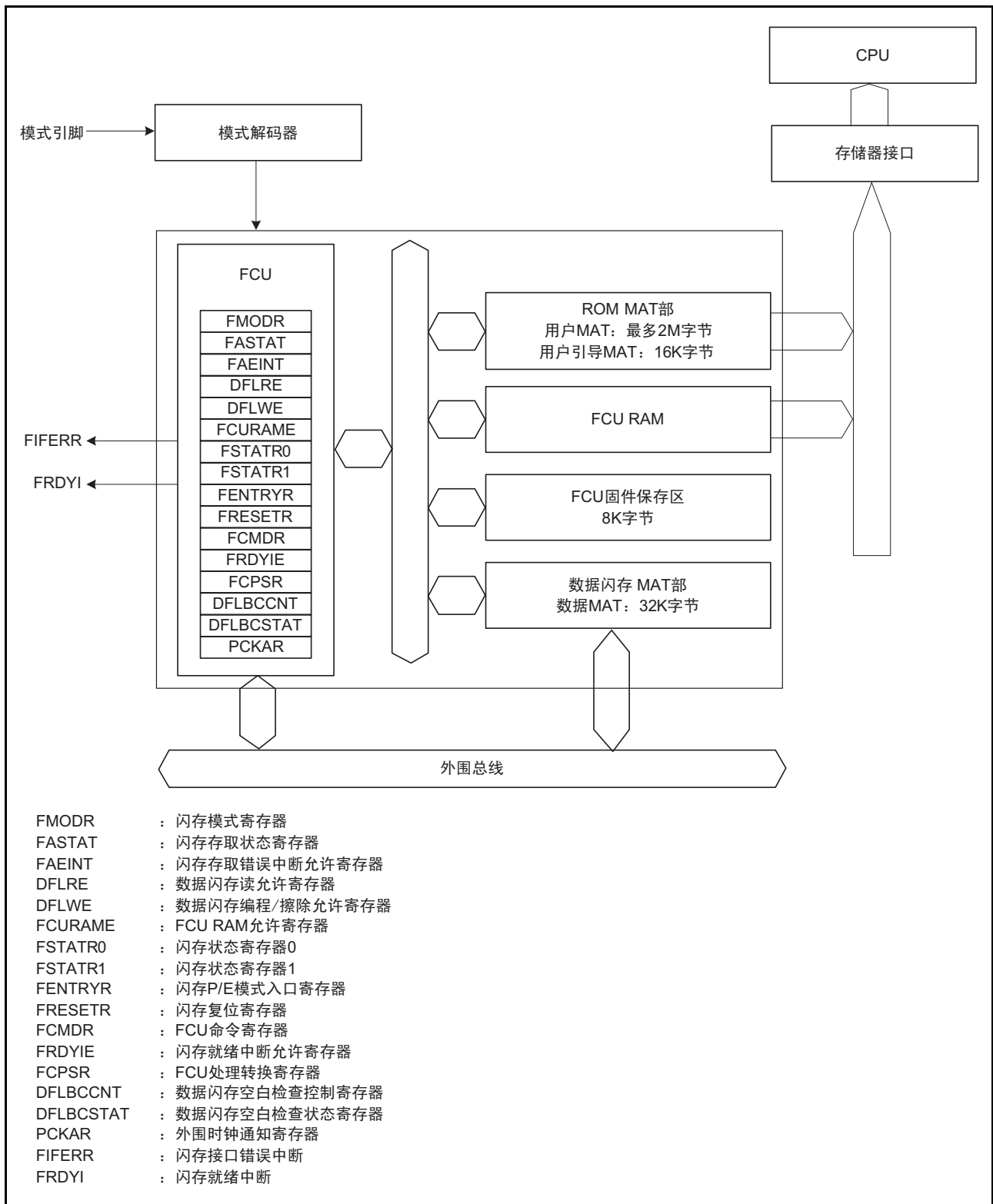


图 26.1 ROM 的框图

ROM 的相关输入 / 输出引脚如表 26.2 所示。

表 26.2 ROM 的相关输入 / 输出引脚

| 引脚名 | 输入 / 输出 | 功能 |
|----------|---------|----------------------------|
| P05/RxD4 | 输入 | 用于引导模式。SCI4 的接收数据（用于主机通信）。 |
| P04/TxD4 | 输出 | 用于引导模式。SCI4 的发送数据（用于主机通信）。 |

26.2 寄存器说明

ROM 的相关寄存器一览表如表 26.3 所示。部分寄存器也有数据闪存的相关位，本章只说明 ROM 的相关位的功能。有关数据闪存相关位功能的详细内容，请参照“27. 数据闪存（保存数据的闪存）”的“27.2 寄存器说明”。

通过复位对 ROM 的相关寄存器进行初始化。

表 26.3 ROM 的相关寄存器一览表

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|----------------|---------|-------|-------------|------|
| 闪存模式寄存器 | FMODR | 00h | 007F C402h | 8 |
| 闪存存取状态寄存器 | FASTAT | 00h | 007F C410h | 8 |
| 闪存存取错误中断允许寄存器 | FAEINT | 9Bh | 007F C411h | 8 |
| 闪存就绪中断允许寄存器 | FRDYIE | 00h | 007F C412h | 8 |
| FCU RAM 允许寄存器 | FCURAME | 0000h | 007F C454h | 16 |
| 闪存状态寄存器 0 | FSTATR0 | 80h | 007F FFB0h | 8 |
| 闪存状态寄存器 1 | FSTATR1 | 0xh | 007F FFB1h | 8 |
| 闪存 P/E 模式入口寄存器 | FENTRYR | 0000h | 007F FFB2h | 16 |
| 闪存保护寄存器 | FPROTR | 0000h | 007F FFB4h | 16 |
| 闪存复位寄存器 | FRESETR | 0000h | 007F FFB6h | 16 |
| FCU 命令寄存器 | FCMDR | FFFFh | 007F FFBAh | 16 |
| FCU 处理转换寄存器 | FCPSR | 0000h | 007F FFC8h | 16 |
| 闪存 P/E 状态寄存器 | FPESTAT | 0000h | 007F FFCCCh | 16 |
| 外围时钟通知寄存器 | PCKAR | 0000h | 007F FFE8h | 16 |
| 闪存编程 / 擦除保护寄存器 | FWEPROR | 02h | 0008 C289h | 8 |

26.2.1 闪存模式寄存器（FMODR）

地址 007F C402h

| | | | | | | | | |
|-------|----|----|----|-------|----|----|----|----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | — | FRDMD | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|------------|---|-----|
| b3-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b4 | FRDMD | FCU 读模式选择位 | 0: 存储区读方式 在 ROM 锁定位读模式中读 ROM 的锁定位时进行设定。 1: 寄存器读方式 在使用锁定位读 2 命令读 ROM 的锁定位时进行设定。 | R/W |
| b7-b5 | — | 保留位 | 读写值都为“0”。 | R/W |

FMODR 寄存器是指定锁定位的读法的寄存器。

在内部 ROM 无效模式中，FMODR 寄存器的读数据为“00h”，不能写。

通过复位对 FMODR 寄存器进行初始化。

FRDMD 位（FCU 读模式选择位）

此位指定锁定位的读法。

在使用数据闪存的空白检查命令时，需要设定为寄存器读方式。

详细内容请参照“27. 数据闪存（保存数据的闪存）”。

26.2.2 闪存存取状态寄存器（FASTAT）

地址 007F C410h

| | | | | | | | | |
|-------|-------|----|----|-------|-------|----|--------|--------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | ROMAE | — | — | CMDLK | DFLAE | — | DFLRPE | DFLWPE |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|--------|-------------------|-------------------------------------|---------------|
| b0 | DFLWPE | 数据闪存的编程 / 擦除保护违反位 | 参照“27. 数据闪存（保存数据的闪存）”。 | R/(W) (注1) |
| b1 | DFLRPE | 数据闪存的读保护违反位 | 参照“27. 数据闪存（保存数据的闪存）”。 | R/(W) (注1) |
| b2 | — | 保留位 | 读写值都为“0”。 | R/W |
| b3 | DFLAE | 数据闪存的存取违反位 | 参照“27. 数据闪存（保存数据的闪存）”。 | R/(W) (注1) |
| b4 | CMDLK | FCU 命令锁定位 | 0: FCU 不处于命令锁定状态 1: FCU 处于命令锁定状态 | R |
| b6-b5 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | ROMAE | ROM 存取违反位 | 0: 无 ROM 存取错误 1: 有 ROM 存取错误 | R/(W) (注1) |

注1. 为了将标志置“0”，只能在读“1”后写“0”。

FASTAT 寄存器是确认有无 ROM/ 数据闪存的存取违反的寄存器。

在内部 ROM 无效模式中，FASTAT 寄存器的读数据为“00h”，不能写。如果 FASTAT 寄存器的某位变为“1”，FCU 就进入命令锁定状态（参照“26.8.2 错误保护”）。为了解除命令锁定状态，需要在将 FASTAT 寄存器置“10h”后，给 FCU 发行状态寄存器清除命令。

通过复位对 FASTAT 寄存器进行初始化。

CMDLK 位（FCU 命令锁定位）

此位表示 FCU 处于命令锁定状态（参照“26.8.2 错误保护”）。

[为“1”的条件]

- FCU 检测到错误并且转移到命令锁定状态后

[为“0”的条件]

- 在 FASTAT 寄存器为“10h”的状态下给 FCU 发行状态寄存器清除命令后

ROMAE 位（ROM 存取违反位）

此位表示有无 ROM 的存取违反。

如果 ROMAE 位变为“1”，FSTATR0.ILGLERR 位就变为“1”，FCU 进入命令锁定状态。

[为“1”的条件]

- 在 FENTRYR.FENTRY1 位（注1）为“1”并且在 ROM P/E 正常模式的状态下，对 ROM 的编程/擦除地址（00E0 0000h～00EF FFFFh）发行读存取命令
- 在 FENTRYR.FENTRY0 位为“1”并且在 ROM P/E 正常模式的状态下，对 ROM 的编程/擦除地址（00F0 0000h～00FF FFFFh）发行读存取命令
- 在 FENTRY1 位为“0”的状态下，对 ROM 的编程/擦除地址（00E0 0000h～00EF FFFFh）发行存取命令
- 在 FENTRY0 位为“0”的状态下，对 ROM 的编程/擦除地址（00F0 0000h～00FF FFFFh）发行存取命令
- 在 FENTRYR 寄存器不为“0000h”的状态下，对 ROM 的读地址（FFE0 0000h～FFFF FFFFh）发行读存取命令

[为“0”的条件]

- 读“1”后写“0”时

注1. ROM 容量小于等于 1M 字节的产品不能使用。

26.2.3 闪存存取错误中断允许寄存器（FAEINT）

地址 007F C411h

| | | | | | | | | |
|-------|---------|----|----|---------|---------|----|----------|----------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | ROMAEIE | — | — | CMDLKIE | DFLAEIE | — | DFLRPEIE | DFLWPEIE |
| 复位后的值 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|-----------------------|---|-----|
| b0 | DFLWPEIE | 数据闪存的编程 / 擦除保护违反中断允许位 | 参照“27. 数据闪存（保存数据的闪存）”。 | R/W |
| b1 | DFLRPEIE | 数据闪存的读保护违反中断允许位 | 参照“27. 数据闪存（保存数据的闪存）”。 | R/W |
| b2 | — | 保留位 | 读写值都为“0”。 | R/W |
| b3 | DFLAEIE | 数据闪存的存取违反中断允许位 | 参照“27. 数据闪存（保存数据的闪存）”。 | R/W |
| b4 | CMDLKIE | FCU 命令锁定中断允许位 | 0: 在 FASTAT.CMDLK 位为“1”时，不产生 FIFERR 中断请求。 1: 在 FASTAT.CMDLK 位为“1”时，产生 FIFERR 中断请求。 | R/W |
| b6-b5 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | ROMAEIE | ROM 存取违反中断允许位 | 0: 在 FASTAT.ROMAE 位为“1”时，不产生 FIFERR 中断请求。 1: 在 FASTAT.ROMAE 位为“1”时，产生 FIFERR 中断请求。 | R/W |

FAEINT 寄存器设定允许或者禁止闪存接口错误中断（FIFERR）的输出。
在内部 ROM 无效模式中，FAEINT 寄存器的读数据为“00h”，不能写。
通过复位对 FAEINT 寄存器进行初始化。

CMDLKIE 位（FCU 命令锁定中断允许位）

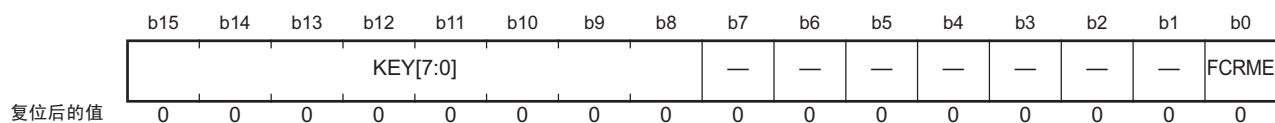
在发生 FCU 命令锁定并且 FASTAT.CMDLK 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的产生。

ROMAEIE 位（ROM 存取违反中断允许位）

在发生 ROM 存取违反并且 FASTAT.ROMAE 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的产生。

26.2.4 FCU RAM 允许寄存器（FCURAME）

地址 007F C454h



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|----------|---------------|--|---------------|
| b0 | FCRME | FCU 的 RAM 允许位 | 0: 禁止存取 FCU 的 RAM 1: 允许存取 FCU 的 RAM | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |
| b15-b8 | KEY[7:0] | 键码 | 控制能否改写 FCRME 位。 | R(/W) (注1) |

注 1. 不保持写数据。

FCURAME 寄存器允许或者禁止存取 FCU 的 RAM 区。

只在给高位字节写特定值时才能通过字存取写此寄存器，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，FCURAME 寄存器的读数据为“00h”，不能写。

通过复位对 FCURAME 寄存器进行初始化。

FCRME 位（FCU 的 RAM 允许位）

此位允许或者禁止存取 FCU 的 RAM。

只在 KEY[7:0] 位为“C4h”时才能通过字存取写 FCRME 位。要写 FCU 的 RAM 时，必须将 FENTRYR 寄存器置“0000h”，使 FCU 停止。

KEY[7:0] 位（键码）

这些位控制能否改写 FCRME 位。

不保持 KEY[7:0] 位的写数据。

26.2.5 闪存状态寄存器 0（FSTATR0）

地址 007F FFB0h

| | | | | | | | | |
|-------|------|-------------|------------|------------|------------|----|------------|------------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | FRDY | ILGLE RR | ERSE RR | PRGE RR | SUSR DY | — | ERSS PD | PRGS PD |
| 复位后的值 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|----|---------|---------|---|-----|
| b0 | PRGSPD | 编程挂起状态位 | 0: 下述以外的状态 1: 正在进行编程的中止处理或者正在编程挂起 | R |
| b1 | ERSSPD | 擦除挂起状态位 | 0: 下述以外的状态 1: 正在进行擦除的中止处理或者正在擦除挂起 | R |
| b2 | — | 保留位 | 读取值为“0”，写操作无效。 | R |
| b3 | SUSRDY | 挂起就绪位 | 0: 不能接受 P/E 挂起命令 1: 能接受 P/E 挂起命令 | R |
| b4 | PRGERR | 编程错误位 | 0: 编程处理正常结束 1: 在编程处理过程中发生错误 | R |
| b5 | ERSERR | 擦除错误位 | 0: 擦除处理正常结束 1: 在擦除处理过程中发生错误 | R |
| b6 | ILGLERR | 非法命令错误位 | 0: FCU 未检测到非法命令和 ROM/ 数据闪存的非法存取 1: FCU 检测到非法命令或者 ROM/ 数据闪存的非法存取 | R |
| b7 | FRDY | 闪存就绪位 | 0: 正在进行编程或者擦除处理，正在进行编程或者擦除的中止处理，正在进行锁定位读 2 命令处理，或者正在进行数据闪存的空白检查处理（参照“27. 数据闪存（保存数据的闪存）”）。 1: 未执行上述处理 | R |

FSTATR0 寄存器是确认 FCU 状态的寄存器。

在内部 ROM 无效模式中，FSTATR0 寄存器的读数据为“00h”。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FSTATR0 寄存器进行初始化。

PRGSPD 位（编程挂起状态位）

此位表示 FCU 正在进行编程的中止处理或者已经转移到编程挂起状态，详细内容请参照“26.7 挂起”。

[为“1”的条件]

- 开始进行编程的中止处理时

[为“0”的条件]

- 接受恢复命令时

ERSSPD 位（擦除挂起状态位）

此位表示 FCU 正在进行擦除的中止处理或者已经转移到擦除挂起状态，详细内容请参照“26.7 挂起”。

[为“1”的条件]

- 开始进行擦除的中止处理时

[为“0”的条件]

- 接受恢复命令时

SUSRDY 位（挂起就绪位）

此位表示 FCU 能否接受 P/E 挂起命令。

[为“1”的条件]

- 在开始进行编程或者擦除处理后，转移到能接受P/E挂起命令的状态时

[为“0”的条件]

- 接受P/E挂起命令时
- 在编程或者擦除处理过程中转移到命令锁定状态时

PRGERR 位（编程错误位）

此位表示通过 FCU 对 ROM/ 数据闪存进行编程处理的结果。

当 PRGERR 位为“1”时，FCU 进入命令锁定状态，详细内容请参照“26.8.2 错误保护”。

[为“1”的条件]

- 在编程过程中发生错误时
- 对由锁定位保护的区域发行编程命令时

[为“0”的条件]

- FCU 发行状态寄存器清除命令后

ERSERR 位（擦除错误位）

此位表示通过 FCU 对 ROM/ 数据闪存进行擦除处理的结果。

当 ERSERR 位为“1”时，FCU 进入命令锁定状态，详细内容请参照“26.8.2 错误保护”。

[为“1”的条件]

- 在擦除过程中发生错误时
- 对由锁定位保护的区域发行擦除命令时

[为“0”的条件]

- FCU 发行状态寄存器清除命令后

ILGLERR 位（非法命令错误位）

此位表示 FCU 检测到非法命令或者 ROM/ 数据闪存的非法存取。

当 ILGLERR 位为“1”时，FCU 进入命令锁定状态，详细内容请参照“26.8.2 错误保护”。

[为“1”的条件]

- FCU 检测到非法命令时
- FCU 检测到 ROM/ 数据闪存的非法存取时
(FASTAT.ROMAE 位、FASTAT.DFLAE 位、FASTAT.DFLRPE 位或者 FASTAT.DFLWPE 位为“1”)
- FENTRYR 寄存器的设定不正确时

[为“0”的条件]

- 在 FASTAT 寄存器为“10h”的状态下，FCU 发行状态寄存器清除命令后

FRDY 位（闪存就绪位）

此位是确认 FCU 处理状态的位。

26.2.6 闪存状态寄存器 1（FSTATR1）

地址 007F FFB1h

| | | | | | | | | |
|-------|------------|----|----|-------------|----|----|----|----|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | FCUE RR | — | — | FLOC KST | — | — | — | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | x | x |

x: 不定值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|---------|---------------------------------------|-----|
| b1-b0 | — | 保留位 | 读取值为不定值，写操作无效。 | R |
| b3-b2 | — | 保留位 | 读取值为“0”，写操作无效。 | R |
| b4 | FLOCKST | 锁定位的状态位 | 0: 保护状态 1: 非保护状态 | R |
| b6-b5 | — | 保留位 | 读取值为“0”，写操作无效。 | R |
| b7 | FCUERR | FCU 错误位 | 0: 在 FCU 处理中未发生错误 1: 在 FCU 处理中发生错误 | R |

FSTATR1 寄存器是确认 FCU 状态的寄存器。

在内部 ROM 无效模式中，FSTATR1 寄存器的读数据为“00h”。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FSTATR1 寄存器进行初始化。

FLOCKST 位（锁定位的状态位）

当使用锁定位读 2 命令时，此位反映锁定位的读数据。

在发行锁定位读 2 命令后 FSTATR0.FRDY 位变为“1”时，将有效数据保存到 FLOCKST 位。FLOCKST 位的值保持到下一个锁定位读 2 命令结束为止。

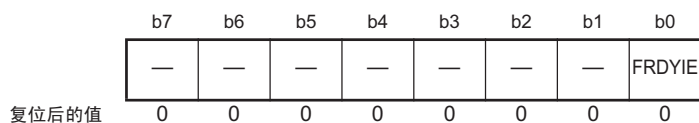
FCUERR 位（FCU 错误位）

此位表示在 FCU 内部处理中发生错误。

当 FCUERR 位为“1”时，必须将 FRESETR.FRESET 位置“1”，对 FCU 进行初始化。必须将 FCU 固件从 FCU 固件区重新复制到 FCU 的 RAM 区。

26.2.7 闪存就绪中断允许寄存器（FRDYIE）

地址 007F C412h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|--------|-----------|--|-----|
| b0 | FRDYIE | 闪存就绪中断允许位 | 0: 禁止 FRDYI 中断请求的产生 1: 允许 FRDYI 中断请求的产生 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

RDYIE 寄存器设定允许或者禁止闪存就绪中断（FRDYI）的输出。

在内部 ROM 无效模式中，FRDYIE 寄存器的读数据为“00h”，不能写。

通过复位对 FRDYIE 寄存器进行初始化。

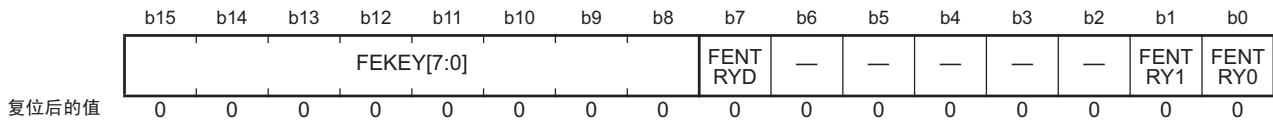
FRDYIE 位（闪存就绪中断允许位）

此位允许或者禁止在结束编程或者擦除处理时产生 FRDYI 中断请求。

如果在将 FRDYIE 位置“1”的状态下结束 FCU 命令的执行（FSTAT0.FRDY 位从“0”变为“1”），就产生闪存就绪中断请求（FRDYI）。

26.2.8 闪存 P/E 模式入口寄存器（FENTRYR）

地址 007F FFB2h



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|------------------|-----------------|---|---------------|
| b0 | FENTRY0 | ROM P/E 模式入口位 0 | 0: ROM 的 1M 字节以内（注 1）的区域为 ROM 读模式 1: ROM 的 1M 字节以内（注 1）的区域为 ROM P/E 模式 | R/W |
| b1 | FENTRY1 （注 2） | ROM P/E 模式入口位 1 | 0: ROM 的 1M 字节以上（注 1）的区域为 ROM 读模式 1: ROM 的 1M 字节以上（注 1）的区域为 ROM P/E 模式 | R/W |
| b6-b2 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | FENTRYD | 数据闪存的 P/E 模式入口位 | 参照“27. 数据闪存（保存数据的闪存）”。 | R/W |
| b15-b8 | FEKEY[7:0] | 键码 | 控制能否改写 FENTRYD 位、FENTRY1 位（注 2）和 FENTRY0 位。 | R(W) （注 3） |

注 1. ROM 的 1M 字节以内 读地址 : FFF0 0000h ~ FFFF FFFFh
编程/擦除地址 : 00F0 0000h ~ 00FF FFFFh
ROM 的 1M 字节以上 读地址 : FFE0 0000h ~ FFEF FFFFh
编程/擦除地址 : 00E0 0000h ~ 00EF FFFFh

注 2. ROM 容量小于等于 1M 字节的产品不能使用。

注 3. 不保持写数据。

FENTRYR 寄存器是将 ROM/ 数据闪存设定为 P/E 模式的寄存器。

为了将 ROM/ 数据闪存设定为 P/E 模式，使 FCU 能接受命令，需要将 FENTRYD 位、FENTRY1 位（注 1）或者 FENTRY0 位置“1”。但是，如果将这些位中的多个位置“1”，FSTATR0.ILGLERR 位就变为“1”，FCU 进入命令锁定状态。

只在给高位字节写特定值时才能通过字存取写此寄存器，而在其他情况下写此寄存器时，此寄存器就被初始化，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，FENTRYR 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FENTRYR 寄存器进行初始化。

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

FENTRY0 位（ROM P/E 模式入口位 0）

此位将 ROM 的 1M 字节（读地址：FFF0 0000h ~ FFFF FFFFh，编程 / 擦除地址：00F0 0000h ~ 00FF FFFFh）设定为 P/E 模式。

[写操作的有效条件（满足以下全部条件时）]

- 内部ROM为有效模式
- FSTATR0.FRDY位为“1”
- 通过字存取给FEKEY[7:0]位写“AAh”

[为“1”的条件]

- 在满足写操作的有效条件并且FENTRYR寄存器为“0000h”的状态下，给FENTRY0位写“1”时

[为“0”的条件]

- 通过字节存取进行写操作时
- 在FEKEY[7:0]位不为“AAh”的状态下，通过字存取进行写操作时
- 在满足写操作的有效条件的状态下给FENTRY0位写“0”时
- 在满足写操作的有效条件并且FENTRYR寄存器不为“0000h”的状态下，写FENTRYR寄存器时

FENTRY1 位（注1）（ROM P/E 模式入口位 1）

此位将 ROM 的 1M 字节（读地址：FFE0 0000h ~ FFEF FFFFh，编程 / 擦除地址：00E0 0000h ~ 00EF FFFFh）设定为 P/E 模式。

[写操作的有效条件（满足以下全部条件时）]

- 内部ROM为有效模式
- FSTATR0.FRDY位为“1”
- 通过字存取给FEKEY[7:0]位写“AAh”

[为“1”的条件]

- 在满足写操作的有效条件并且FENTRYR寄存器为“0000h”的状态下，给FENTRY1位（注1）写“1”时

[为“0”的条件]

- 通过字节存取进行写操作时
- 在FEKEY[7:0]位不为“AAh”的状态下，通过字存取进行写操作时
- 在满足写操作的有效条件的状态下给FENTRY1位（注1）写“0”时
- 在满足写操作的有效条件并且FENTRYR寄存器不为“0000h”的状态下，写FENTRYR寄存器时

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

FEKEY[7:0] 位（键码）

这些位控制能否改写 FENTRYD 位、FENTRY1 位（注1）和 FENTRY0 位。

不保持 FEKEY[7:0] 位的写数据。

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

表 26.4 各产品的 FENTRY1 位和 FENTRY0 位的对应表

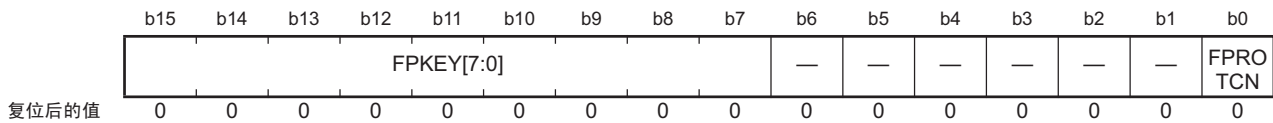
| 产品型号 | ROM 容量 | FENTRY1 位 (注 1) | FENTRY0 位 | FCU 模式 | 对象地址 |
|----------|---------|--------------------|-----------|------------|---|
| R5F56108 | 2M 字节 | 0 | 0 | ROM 读模式 | 读地址: FFE0 0000h ~ FFFF FFFFh |
| | | 0 | 1 | ROM P/E 模式 | 编程 / 擦除地址: 00F0 0000h ~ 00FF FFFFh |
| | | 1 | 0 | | 禁止存取地址: FFE0 0000h ~ FFFF FFFFh |
| | | 1 | 1 | 不能设定 | — |
| R5F56107 | 1.5M 字节 | 0 | 0 | ROM 读模式 | 读地址: FFE8 0000h ~ FFFF FFFFh |
| | | 0 | 1 | ROM P/E 模式 | 编程 / 擦除地址: 00F0 0000h ~ 00FF FFFFh |
| | | 1 | 0 | | 禁止存取地址: FFE8 0000h ~ FFFF FFFFh |
| | | 1 | 1 | 不能设定 | — |
| R5F56106 | 1M 字节 | 不能使用 (注 2) | 0 | ROM 读模式 | 读地址: FFF0 0000h ~ FFFF FFFFh |
| | | 不能使用 (注 2) | 1 | ROM P/E 模式 | 编程 / 擦除地址: 00F0 0000h ~ 00FF FFFFh 禁止存取地址: FFF0 0000h ~ FFFF FFFFh |
| R5F56104 | 768K 字节 | 不能使用 (注 2) | 0 | ROM 读模式 | 读地址: FFF4 0000h ~ FFFF FFFFh |
| | | 不能使用 (注 2) | 1 | ROM P/E 模式 | 编程 / 擦除地址: 00F4 0000h ~ 00FF FFFFh 禁止存取地址: FFF4 0000h ~ FFFF FFFFh |

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

注 2. 在 R5F56106 和 R5F56104 的情况下，此位为保留位，只能写“0”。

26.2.9 闪存保护寄存器（FPROTR）

地址 007F FFB4h



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|------------|----------|------------------------------------|----------------|
| b0 | FPROTCN | 锁定位保护取消位 | 0: 通过锁定位进行的保护有效 1: 通过锁定位进行的保护无效 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |
| b15-b8 | FPKEY[7:0] | 键码 | 控制能否改写 FPROTCN 位。 | R(/W) (注 1) |

注 1. 不保持写数据。

FPROTR 寄存器设定通过锁定位进行的编程 / 擦除保护功能的有效或者无效。

只在给高位字节写特定值时才能通过字存取写此寄存器，而在其他情况下写此寄存器时，此寄存器就被初始化，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，FPROTR 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FPROTR 寄存器进行初始化。

FPROTCN 位（锁定位保护取消位）

此位设定通过锁定位进行的编程 / 擦除保护的有效或者无效。

[为“1”的条件]

- 在 FENTRYR 寄存器的值不为“0000h”的状态下，通过字存取分别给 FPKEY[7:0]位和 FPROTCN 位写“55h”和“1”时

[为“0”的条件]

- 通过字节存取进行写操作时
- 在 FPKEY[7:0]位不为“55h”的状态下，通过字存取进行写操作时
- 通过字存取分别给 FPKEY[7:0]位和 FPROTCN 位写“55h”和“0”时
- FENTRYR 寄存器的值为“0000h”时

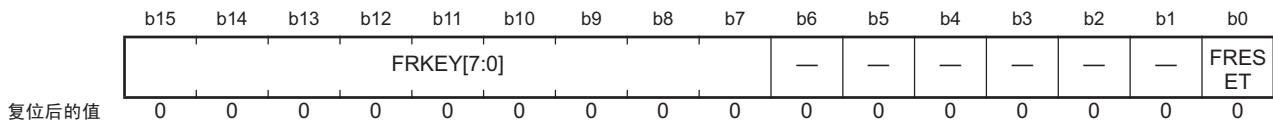
FPKEY[7:0] 位（键码）

此位控制能否改写 FPROTCN 位。

不保持 FPKEY[7:0] 位的写数据。

26.2.10 闪存复位寄存器（FRESETR）

地址 007F FFB6h



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|------------|--------|---------------------------------|--------------|
| b0 | FRESETR | 闪存复位的位 | 0: 不对 FCU 进行复位 1: 对 FCU 进行复位 | R/W |
| b7-b1 | — | 保留位 | 读写值都为“0”。 | R/W |
| b15-b8 | FRKEY[7:0] | 键码 | 控制能否改写 FRESETR 位。 | R(W) (注1) |

注 1. 不保持写数据。

FRESETR 寄存器是用于对 FCU 进行初始化的寄存器。

只在给高位字节写特定值时才能通过字存取写此寄存器，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，FRESETR 寄存器的读数据为“0000h”，不能写。

通过复位对 FRESETR 寄存器进行初始化。

FRESETR 位（闪存复位的位）

如果将 FRESETR 位置“1”，ROM/数据闪存的编程或者擦除就被强制结束并且 FCU 被初始化。

给编程或者擦除过程中的 ROM/数据闪存的存储器外加高电压。为了确保存储器外加电压的下降所需的时间，在对 FCU 进行初始化时，FRESETR 位为“1”的状态必须保持 tRESW2（参照“29. 电特性”）的时间。在 FRESETR 位保持“1”的期间，必须禁止读 ROM/数据闪存。因为在 FRESETR 位为“1”的状态下对 FENTRYR 寄存器进行初始化，所以不能使用 FCU 命令。

只在 FRKEY[7:0] 位为“CCh”时才能通过字存取写 FRESETR 位。

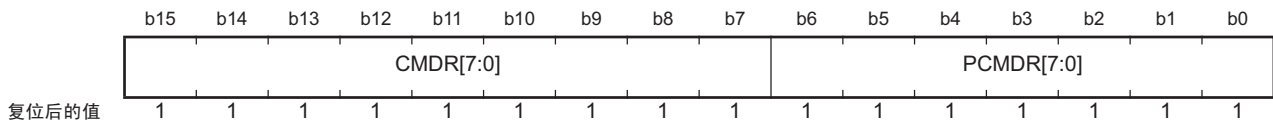
FRKEY[7:0] 位（键码）

这些位控制能否改写 FRESETR 位。

不保持 FRKEY[7:0] 位的写数据。

26.2.11 FCU 命令寄存器（FCMDR）

地址 007F FFBAh



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|------------|------|------------------|-----|
| b7-b0 | PCMDR[7:0] | 预置命令 | 保存 FCU 接受的前一个命令。 | R |
| b15-b8 | CMDR[7:0] | 命令 | 保存 FCU 接受的最新命令。 | R |

FCMDR 寄存器保存 FCU 接受的命令。

在内部 ROM 无效模式中，FCMDR 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FCMDR 进行初始化。

接受各命令后的 FCMDR 寄存器的状态如表 26.5 所示，空白检查的内容请参照数据闪存的“27.6 数据闪存的编程 / 擦除”。

表 26.5 接受各命令后的 FCMDR 寄存器状态

| 命令 | CMDR | PCMDR |
|------------------|------|-------|
| 正常模式转移 | FFh | 前一个命令 |
| 状态读模式转移 | 70h | 前一个命令 |
| 锁定位读模式转移（锁定位读 1） | 71h | 前一个命令 |
| 外围时钟通知命令 | E9h | 前一个命令 |
| 编程 | E8h | 前一个命令 |
| 块擦除 | D0h | 20h |
| P/E 挂起 | B0h | 前一个命令 |
| P/E 恢复 | D0h | 前一个命令 |
| 状态寄存器清除 | 50h | 前一个命令 |
| 锁定位读 2 的空白检查 | D0h | 71h |
| 锁定位编程 | D0h | 77h |

26.2.12 FCU 处理转换寄存器（FCPSR）

地址 007F FFC8h

| | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|---------|
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ESUSPMD |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|--------|---------|---------|------------------------|-----|
| b0 | ESUSPMD | 擦除挂起模式位 | 0: 挂起优先模式 1: 擦除优先模式 | R/W |
| b15-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

FCPSR 寄存器选择 FCU 擦除处理的挂起方法。

在内部 ROM 无效模式中，FCPSR 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FCPSR 寄存器进行初始化。

ESUSPMD 位（擦除挂起模式位）

在 FCU 进行 ROM/ 数据闪存的擦除处理过程中发行 P/E 挂起命令时，此位选择擦除中断处理模式，详细内容请参照“26.7 挂起”。

26.2.13 闪存 P/E 状态寄存器（FPESTAT）

地址 007F FFCCh

| | | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|----|--------------|----|----|----|----|----|----|----|---|
| | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | |
| | — | — | — | — | — | — | — | — | PEERRST[7:0] | | | | | | | — | — |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| 位 | 符号 | 位名 | 功能 | R/W |
|--------|--------------|-----------|--|-----|
| b7-b0 | PEERRST[7:0] | P/E 错误状态位 | 01h: 锁定位保护区的编程错误 02h: 由锁定位保护以外的原因引起的编程错误 11h: 锁定位保护区的擦除错误 12h: 由锁定位保护以外的原因引起的擦除错误 上述以外: 保留 | R |
| b15-b8 | — | 保留位 | 读取值为“0”，写操作无效。 | R |

FPESTAT 寄存器是表示 ROM/ 数据闪存的编程或者擦除处理结果的寄存器。

在内部 ROM 无效模式中，FPESTAT 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FPESTAT 寄存器进行初始化。

PEERRST[7:0] 位（P/E 错误状态位）

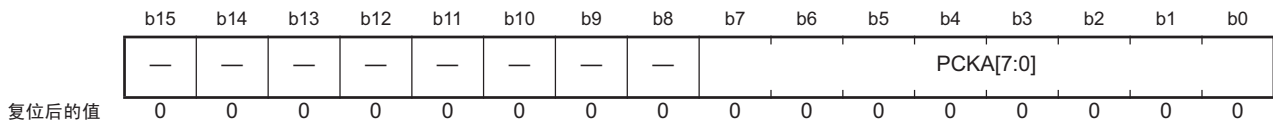
这些位表示在 ROM/ 数据闪存的编程或者擦除处理过程中发生错误时的错误原因。

只在 FSTATR0.ERSERR 位或者 FSTATR0.PRGERR 位为“1”的状态下，PEERRST[7:0] 位的值才有效。

当 ERSERR 位和 PRGERR 位为“0”时，PEERRST[7:0] 位保持过去发生的错误原因的值。

26.2.14 外围时钟通知寄存器（PCKAR）

地址 007F FFE8h



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|-----------|---------|--|-----|
| b7-b0 | PCKA[7:0] | 外围时钟通知位 | 在对 ROM/ 数据闪存进行编程或者擦除时，这些位设定外围时钟（PCLK）。 | R/W |
| b15-b8 | — | 保留位 | 读写值都为“0”。 | R/W |

在对 ROM/ 数据闪存进行编程或者擦除时，PCKAR 寄存器将外围时钟（PCLK）的频率设定信息通知定序器。此设定用于控制编程或者擦除的时间。

在内部 ROM 无效模式中，PCKAR 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 PCKAR 寄存器进行初始化。

PCKA[7:0] 位（外围时钟通知位）

在对 ROM/ 数据闪存进行编程或者擦除时，这些位设定外围时钟（PCLK）。

必须在编程或者擦除前给 PCKA[7:0] 位设定 PCLK 的频率，然后发行外围时钟通知命令。不能在对 ROM/ 数据闪存进行编程或者擦除过程中更改频率。

如下进行设定值的计算：

- 将以 MHz 为单位表现的工作频率转换为 2 进制数并且写到 PCKA[7:0] 位。
以外围时钟的工作频率为 35.9MHz 时的情况为具体例子，说明如下：
- 将 35.9 进行舍入。
- 将 36 转换为 2 进制数，给 PCKA[7:0] 位的高位设定“00h”、低位设定“24h”（0010 0100b）。

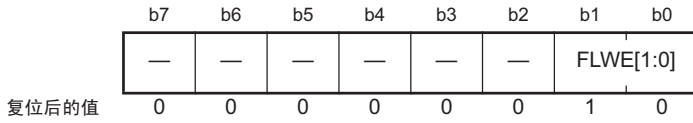
注 1. 如果将 PCKA[7:0] 位设定在 8MHz ~ 50MHz 的范围以外，就不能对 ROM/ 数据闪存发行改写命令。

注 2. 如果给 PCKA[7:0] 位设定和实际频率不同的频率，就可能破坏 ROM/ 数据闪存的数据。

注 3. 即使有效地利用 PCKA[7:0] 位，在一定程度上改写时间也取决于频率。

26.2.15 闪存编程 / 擦除保护寄存器（FWEPROR）

地址 0008 C289h



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-----------|------------|---|-----|
| b1-b0 | FLWE[1:0] | 闪存编程 / 擦除位 | b1 b0 0 0: 不能进行编程和擦除 0 1: 能进行编程和擦除 1 0: 不能进行编程和擦除（初始值） 1 1: 不能进行编程和擦除 | R/W |
| b7-b2 | — | 保留位 | 读写值都为“0”。 | R/W |

FWEPROR 寄存器是通过软件进行闪存编程 / 擦除保护的读写寄存器。
 在软件待机模式或者深度软件待机模式中，也对 FWEPROR 寄存器进行初始化。

FLWE[1:0] 位（闪存编程 / 擦除位）
 此位通过软件进行闪存编程 / 擦除的保护。

26.3 ROM 的存储器 MAT 结构

RX610 群的 ROM 由最多 2M 字节的用户 MAT 和 16K 字节的用户引导 MAT 构成，这些 MAT 的地址如图 26.2 所示。

必须注意：在进行读、编程或者擦除时，用户 MAT 的地址不同。

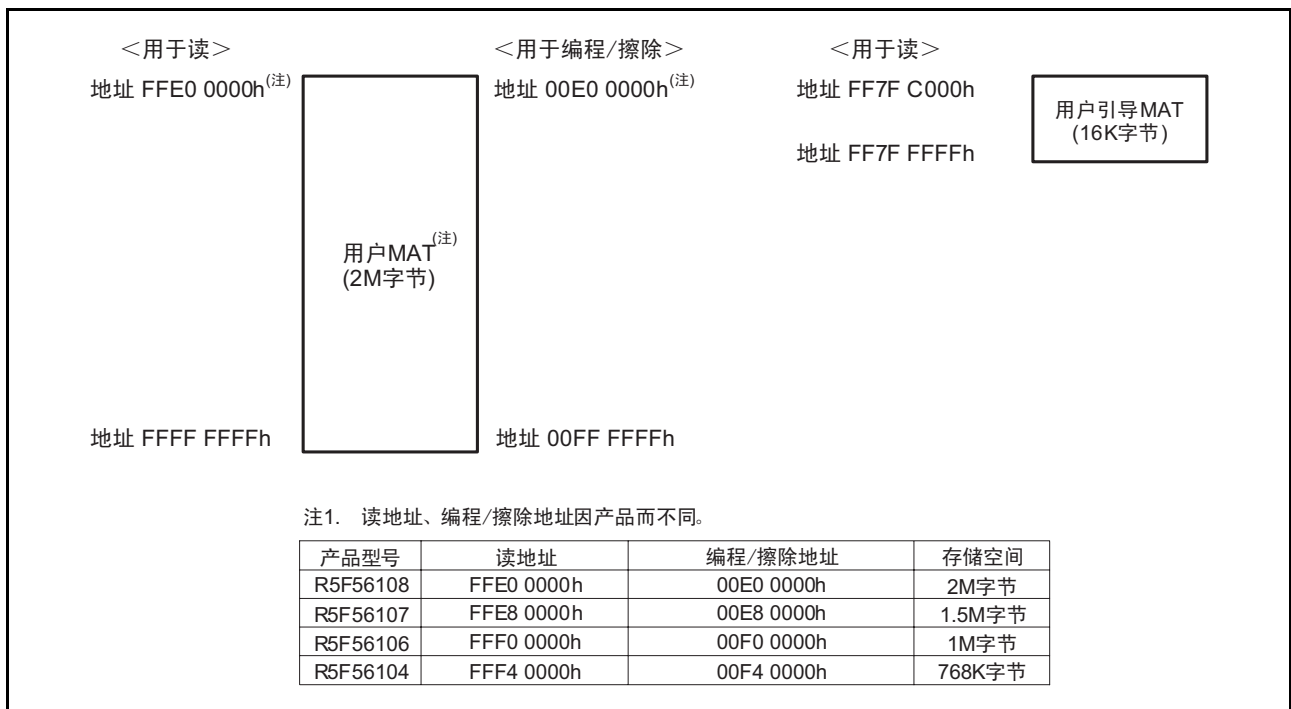


图 26.2 ROM 的存储器 MAT 结构

26.4 块结构

用户 MAT 的擦除块结构如图 26.3 所示。用户 MAT 被分割为 128K 字节（11 块）、64K 字节（9 块）和 8K 字节（8 块），以块为单位进行擦除。以低位地址为“00h”开始的 256 字节为单位进行编程。

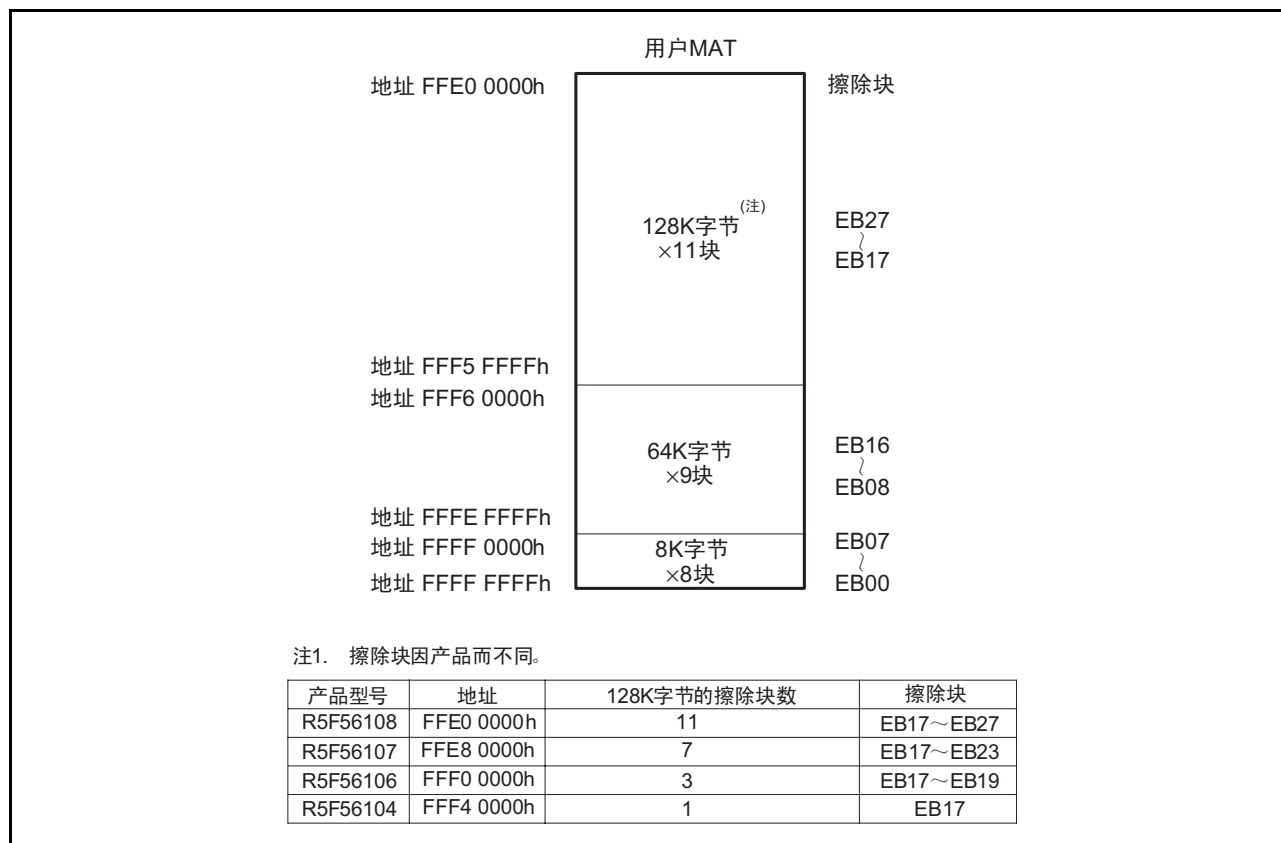


图 26.3 用户 MAT 的擦除块结构

26.5 ROM 相关的运行模式

RX610 群的运行模式转移图如图 26.4 所示。

一旦设定 MD1 引脚和 MD0 引脚并且进行复位解除，就进行如图 26.4 的转移。

有关 MD1 引脚和 MD0 引脚的设定值与 RX610 群的运行模式的关系，请参照“3. 运行模式”。

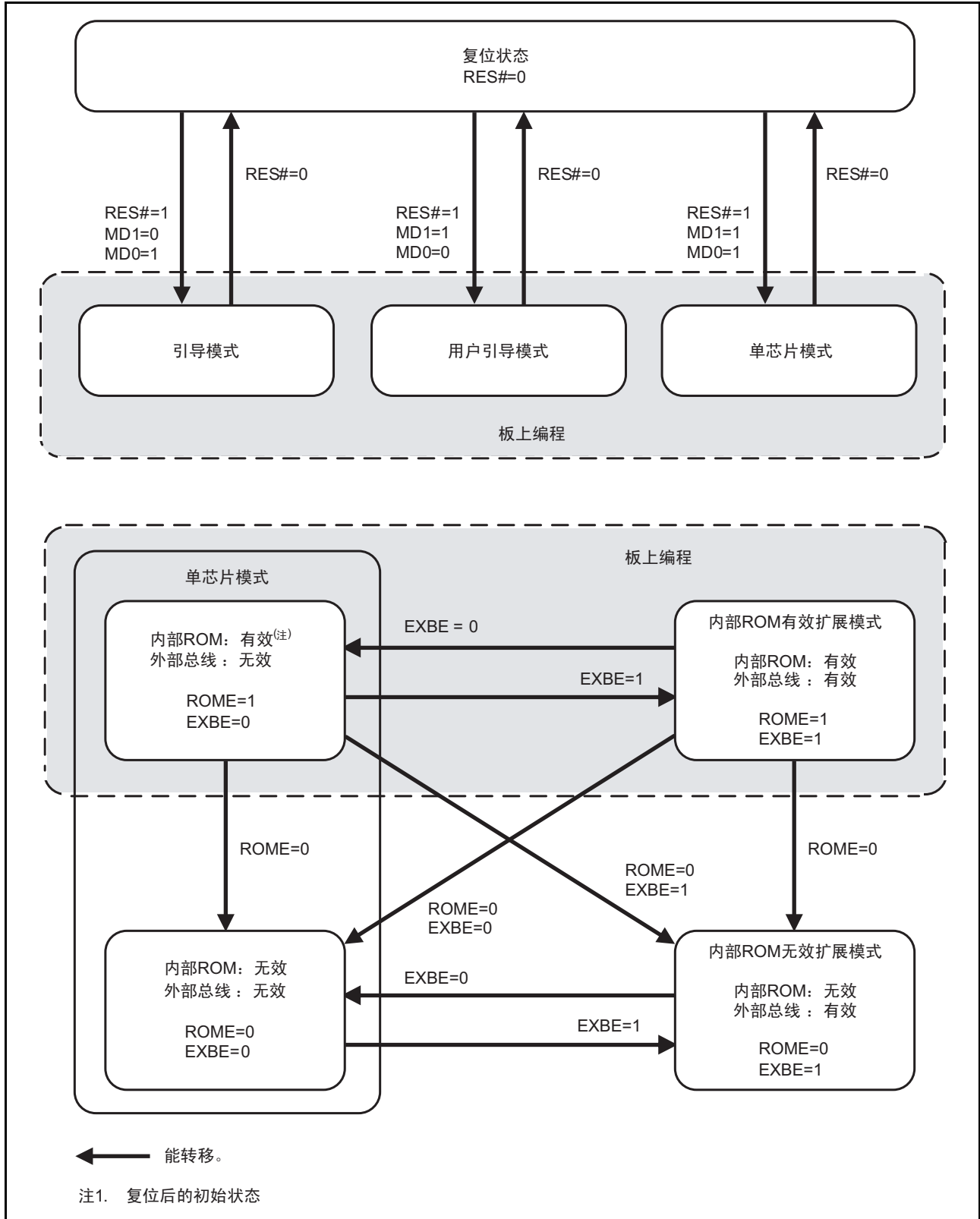


图 26.4 ROM 相关的运行模式转移图

在引导模式、用户引导模式、单芯片模式（内部 ROM 有效）或者内部 ROM 有效扩展模式中，能进行板上 ROM 的读、编程或者擦除。

在各模式中，能编程和擦除的 MAT、复位时的启动 MAT 等不同，各模式的不同点如表 26.6 所示。

表 26.6 各模式的不同点

| 项目 | 引导模式 | 用户引导模式 | 单芯片模式（内部 ROM 有效）/内部 ROM 有效扩展模式 |
|---------------|--------------------|----------|--------------------------------|
| 编程 / 擦除环境 | 板上编程 | | |
| 能编程 / 擦除的 MAT | 用户 MAT 用户引导 MAT | 用户 MAT | 用户 MAT |
| 块分割擦除 | ○（注 1） | ○ | ○ |
| 复位时的启动 MAT | 嵌入式程序存储 MAT（注 2） | 用户引导 MAT | 用户 MAT |

注 1. 有可能在启动时被全部擦除。此后能擦除特定块，详细内容请参照“26.10.2 ID 码保护”。

注 2. 用户不能使用。

- 只能在引导模式中对用户引导 MAT 进行编程和擦除。
- 在引导模式中，主机能经由 SCI 对用户 MAT、用户引导 MAT 或者数据 MAT 进行读和编程。
- 在用户引导模式中，能从用户引导 MAT 启动，并且通过任意的接口对用户 MAT 或者数据 MAT 进行读和编程。
- 在引导模式中，通过引导模式的嵌入式程序使用内部 RAM。因此，不保持内部 RAM 的数据。

26.6 ROM 的编程 / 擦除

通过给编程 / 擦除的专用定序器（FCU）发行命令（FCU 命令），对 ROM 进行编程和擦除。FCU 有 5 种模式，通过在模式转移后发行编程和擦除命令，进行编程和擦除。

以下说明 ROM 的编程和擦除所需的模式转移和命令体系。这些内容在引导模式、用户引导模式、单芯片模式（内部 ROM 有效模式）和内部 ROM 有效扩展模式中通用。

26.6.1 FCU 的模式

FCU 有 5 种模式，通过写 FENTRYR 寄存器以及 FCU 命令进行模式的转移，FCU 的模式转移图如图 26.5 所示。

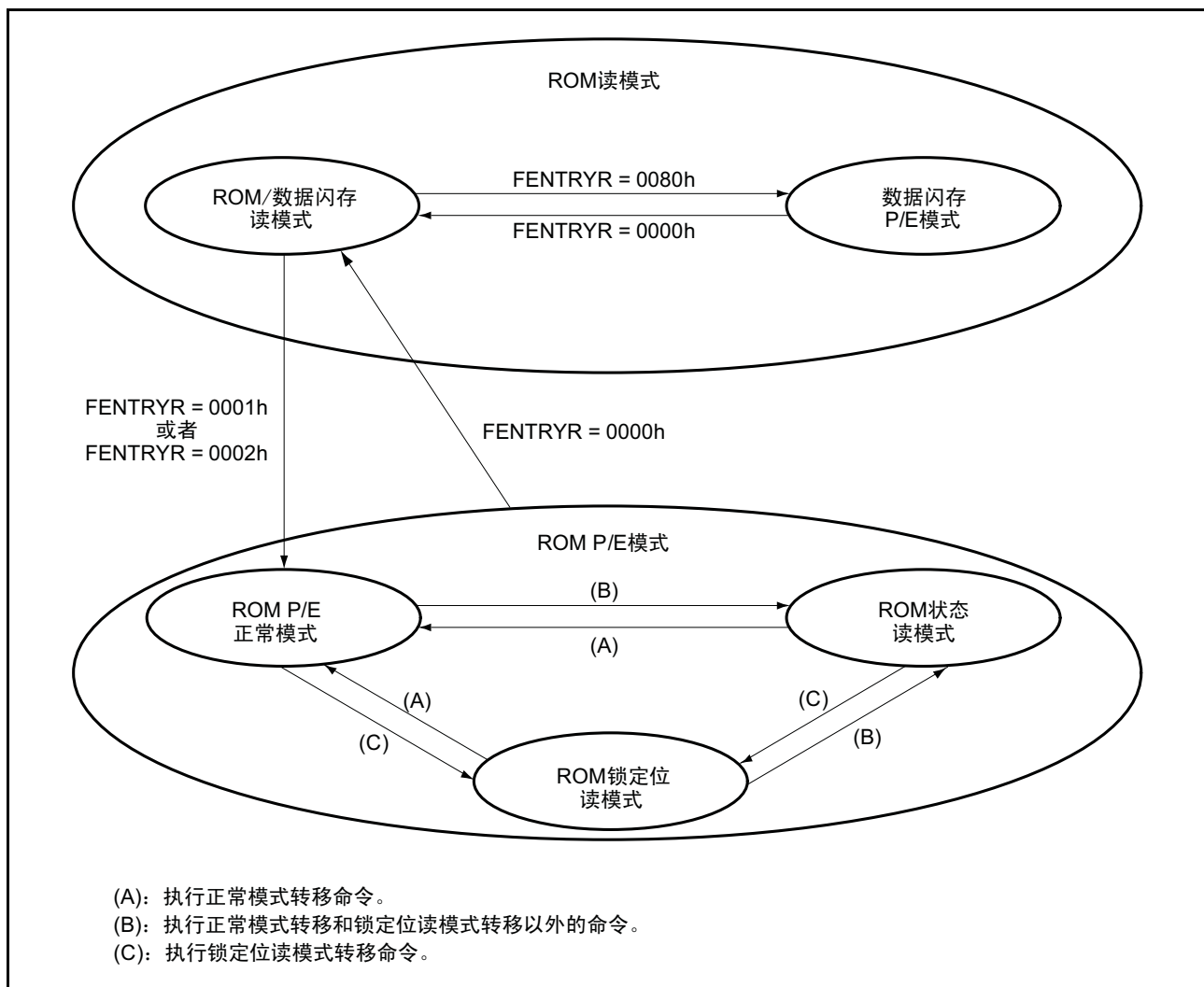


图 26.5 FCU 的模式转移图（ROM 相关）

26.6.1.1 ROM 读模式

ROM 读模式是高速读 ROM 的模式。如果对读地址进行读存取，就能进行 1 个 ICLK 周期的高速读取。ROM 读模式有 ROM/ 数据闪存读模式和数据闪存 P/E 模式。

(1) ROM/ 数据闪存读模式

ROM/ 数据闪存读模式是能读 ROM 和数据闪存的模式，不接受 FCU 命令。如果将 FENTRYR.FENTRY1 位（注）、FENTRYR.FENTRY0 位和 FENTRYR.FENTRYD 位都置“0”，就转移到此模式。

(2) 数据闪存 P/E 模式

数据闪存 P/E 模式是对数据闪存进行编程和擦除的模式，能高速读 ROM。此模式接受数据闪存的 FCU 命令，但是不接受 ROM 的 FCU 命令。如果将 FENTRYR.FENTRY1 位（注）、FENTRYR.FENTRY0 位置“0”并且将 FENTRYR.FENTRYD 位置“1”，就转移到此模式。

有关数据闪存 P/E 模式的详细内容，请参照“27. 数据闪存（保存数据的闪存）”的“27.6.1 FCU 的模式”。

26.6.1.2 ROM P/E 模式

ROM P/E 模式是对 ROM 进行编程和擦除的模式，不能高速读 ROM。如果对读地址进行读存取，就发生 ROM 存取违反，FCU 进入命令锁定状态（参照“26.8.2 错误保护”）。

ROM P/E 模式有 ROM P/E 正常模式、ROM 状态读模式和 ROM 锁定位读模式共 3 种。

(1) ROM P/E 正常模式

ROM P/E 正常模式是在对 ROM 进行编程或者擦除时最先转移的模式。如果在 ROM 读模式中将 FENTRYR.FENTRYD 位置“0”，并且将 FENTRYR.FENTRY1 位（注）或者 FENTRYR.FENTRY0 位置“1”，或者在 ROM P/E 模式中接受正常模式转移命令，就转移到此模式。能接受的命令如表 26.9 所示。

如果在 FENTRYR.FENTRY1 位（注1）和 FENTRYR.FENTRY0 位为“1”的状态下对编程 / 擦除地址进行读存取，就产生 ROM 的存取违反，FCU 进入命令锁定状态（参照“26.8.2 错误保护”）。

(2) ROM 状态读模式

ROM 状态读模式是能读 ROM 状态的模式。如果在 ROM P/E 模式中接受正常模式转移和锁定位读模式转移以外的命令，就转移到此模式。

FSTATR0.FRDY 位为“0”的状态以及发生错误后的命令锁定状态也是 ROM 状态读模式中的状态。能接受的命令如表 26.9 所示。

如果在 FENTRYR.FENTRY1 位（注1）和 FENTRYR.FENTRY0 位为“1”的状态下对相应的编程 / 擦除地址进行读存取，就能读 FSTATR0 寄存器的值。

(3) ROM 锁定位读模式

ROM 锁定位读模式是通过读 ROM 来读锁定位的模式。如果在 ROM P/E 模式中接受锁定位读模式转移命令，就转移到此模式。能接受的命令如表 26.9 所示。

如果在 FENTRYR.FENTRY1 位（注1）和 FENTRYR.FENTRY0 位为“1”的状态下对相应的编程 / 擦除地址进行读存取，读数据的全部位就为存取对象的擦除块的锁定位值。

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

26.6.2 FCU 命令一览表

FCU 命令有 FCU 模式转移命令和编程 / 擦除命令。ROM 能使用的 FCU 命令一览表如表 26.7 所示。

表 26.7 FCU 命令一览表（ROM 相关）

| 命令 | 功能 |
|------------------|--|
| P/E 正常模式转移 | 转移到正常模式（参照“26.6.3 FCU 的模式和命令的关系”）。 |
| 状态读模式转移 | 转移到状态读模式（参照“26.6.3 FCU 的模式和命令的关系”）。 |
| 锁定位读模式转移（锁定位读 1） | 转移到锁定位读模式（参照“26.6.3 FCU 的模式和命令的关系”）。 |
| 外围时钟通知 | 设定外围时钟的频率。 |
| 编程 | 对 ROM 进行编程（以 256 字节为单位）。 |
| 块擦除 | 擦除 ROM（以块为单位，同时擦除锁定位）。 |
| P/E 挂起 | 中止编程和擦除。 |
| P/E 恢复 | 重新开始编程和擦除。 |
| 状态寄存器清除 | 清除 FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位和 FSTATR0.PRGERR 位以及解除命令锁定状态。 |
| 锁定位读 2 / 空白检查 | 读指定的擦除块的锁定位（将锁定位反映到 FSTATR1.FLOCKST 位） / 数据闪存的空白检查。 |
| 锁定位编程 | 对指定的擦除块的锁定位进行编程。 |

锁定位读 2 命令兼用数据闪存的空白检查命令。如果对数据闪存发行锁定位读 2 命令，就进行数据闪存的空白检查（参照“27. 数据闪存（保存数据的闪存）”）。

通过写 ROM 的编程 / 擦除地址给 FCU 发行命令。FCU 命令的格式如表 26.8 所示。如果在 FCU 的特定条件下进行表 26.8 所示的写存取，FCU 就进行各命令对应的处理。

有关 FCU 的特定条件和命令的使用方法，请分别参照“26.6.3 FCU 的模式和命令的关系”和“26.6.4 FCU 命令的使用方法”。

表 26.8 FCU 命令的格式

| 命令 | 总线周期数 | 第 1 周期 | | 第 2 周期 | | 第 3 周期 | | 第 4 ~ 第 5 周期 | | 第 6 周期 | | 第 7 ~ 第 130 周期 | | 第 131 周期 | |
|------------------|-------|--------|-----|--------|-----|--------|------|--------------|------|--------|-----|----------------|-----|----------|-----|
| | | 地址 | 数据 | 地址 | 数据 | 地址 | 数据 | 地址 | 数据 | 地址 | 数据 | 地址 | 数据 | 地址 | 数据 |
| P/E 正常模式转移 | 1 | RA | FFh | — | — | — | — | — | — | — | — | — | — | — | — |
| 状态读模式转移 | 1 | RA | 70h | — | — | — | — | — | — | — | — | — | — | — | — |
| 锁定位读模式转移（锁定位读 1） | 1 | RA | 71h | — | — | — | — | — | — | — | — | — | — | — | — |
| 外围时钟通知 | 6 | RA | E9h | RA | 03h | RA | 0F0h | RA | 0F0h | RA | D0h | — | — | — | — |
| 编程 | 131 | RA | E8h | RA | 80h | WA | WDn | RA | WDn | RA | WDn | RA | WDn | RA | D0h |
| 块擦除 | 2 | RA | 20h | BA | D0h | — | — | — | — | — | — | — | — | — | — |
| P/E 挂起 | 1 | RA | B0h | — | — | — | — | — | — | — | — | — | — | — | — |
| P/E 恢复 | 1 | RA | D0h | — | — | — | — | — | — | — | — | — | — | — | — |
| 状态寄存器清除 | 1 | RA | 50h | — | — | — | — | — | — | — | — | — | — | — | — |
| 锁定位读 2 | 2 | RA | 71h | BA | D0h | — | — | — | — | — | — | — | — | — | — |
| 锁定位编程 | 2 | RA | 77h | BA | D0h | — | — | — | — | — | — | — | — | — | — |

地址列 EA: ROM 的编程 / 擦除地址

当 FENTRYR.FENTRY0 位为“1”时: 00F0 0000h ~ 00FF FFFFh 的任意地址

当 FENTRYR.FENTRY1 位（注 1）为“1”时: 00E0 0000h ~ 00EF FFFFh 的任意地址

WA: ROM 的编程目标地址

编程数据 256 字节的起始地址

BA: ROM 的擦除块地址

擦除对象块内的任意地址（由编程 / 擦除地址指定）

数据列 WDn: 编程数据的第 n 个字（n=1 ~ 128）

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

26.6.3 FCU 的模式和命令的关系

FCU 的各模式能接受的 FCU 命令是固定的，而且能接受的命令也因这些模式的 FCU 状态而不同。

需要在进行 FCU 的模式转移后确认 FCU 的状态，然后发行 FCU 命令。

FCU 的模式 / 状态和能接受的命令如表 26.9 所示。如果发行不能接受的命令，FCU 就进入命令锁定状态（参照“26.8.2 错误保护”）。

必须在转移到能接受的模式后，确认 FSTATR0.FRDY 位、FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位和 FSTATR1.FCUERR 位的值，然后发行 FCU 命令。能根据 FASTAT.CMDLK 位的值确认有无发生错误。FASTAT.CMDLK 位的值是 FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位和 FSTATR1.FCUERR 位的值的逻辑或。

表 26.9 FCU 的模式 / 状态和能接受的命令的关系（ROM P/E 模式）

| | P/E 正常模式 | | | 状态读模式 | | | | | | 锁定位读模式 | | | |
|------------------|----------|-------|------|------------|--------------|------------|-------|-------|--------|--------|-------|-------|------|
| | 编程挂起中 | 擦除挂起中 | 其他状态 | 编程 / 擦除处理中 | 编程 / 擦除中止处理中 | 锁定位读 2 处理中 | 编程挂起中 | 擦除挂起中 | 命令锁定状态 | 其他状态 | 编程挂起中 | 擦除挂起中 | 其他状态 |
| FSTATR0.FRDY 位 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0/1 | 1 | 1 | 1 | 1 |
| FSTATR0.SUSRDY 位 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| FSTATR0.ERSSPD 位 | 0 | 1 | 0 | 0 | 0/1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| FSTATR0.PRGSPD 位 | 1 | 0 | 0 | 0 | 0/1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| FASTAT.CMDLK 位 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 正常模式转移 | ○ | ○ | ○ | × | × | × | ○ | ○ | × | ○ | ○ | ○ | ○ |
| 状态读模式转移 | ○ | ○ | ○ | × | × | × | ○ | ○ | × | ○ | ○ | ○ | ○ |
| 锁定位读模式转移（锁定位读 1） | ○ | ○ | ○ | × | × | × | ○ | ○ | × | ○ | ○ | ○ | ○ |
| 外围时钟通知 | × | × | ○ | × | × | × | × | × | × | ○ | × | × | ○ |
| 编程 | × | △ | ○ | × | × | × | × | △ | × | ○ | × | △ | ○ |
| 块擦除 | × | × | ○ | × | × | × | × | × | × | ○ | × | × | ○ |
| P/E 挂起 | × | × | × | ○ | × | × | × | × | × | × | × | × | × |
| P/E 恢复 | ○ | ○ | × | × | × | × | ○ | ○ | × | × | ○ | ○ | × |
| 状态寄存器清除 | ○ | ○ | ○ | × | × | × | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 锁定位读 2 | ○ | ○ | ○ | × | × | × | ○ | ○ | × | ○ | ○ | ○ | ○ |
| 锁定位编程 | × | △ | ○ | × | × | × | × | △ | × | ○ | × | △ | ○ |

○：能接受。

△：只能接受对非擦除中止块进行的编程。

×：不能接受。

26.6.4 FCU 命令的使用方法

FCU 命令有转移 FCU 模式的命令、实际对 ROM 进行编程和擦除的命令、错误处理命令和挂起 / 恢复命令。以下说明各命令，有关能接受各命令的模式和状态，请参照“26.6.3 FCU 的模式和命令的关系”。

26.6.4.1 模式的转移

在此说明模式转移的相关命令，各模式转移的关系请参照图 26.5。

(1) ROM P/E 模式的转移方法

要执行 ROM 相关的 FCU 命令时，需要转移到 ROM P/E 模式。

为了转移到 ROM P/E 模式，将要进行编程或者擦除的 ROM 地址对应的 FENTRYR.FENTRY1 位（注 1）、FENTRYR.FENTRY0 位置“1”。

在进行编程或者擦除时，必须以字节为单位将“01h”写到 FWEPROR 寄存器，进入能进行编程和擦除的状态（参照“26.2.15 闪存编程 / 擦除保护寄存器（FWEPROR）”）。

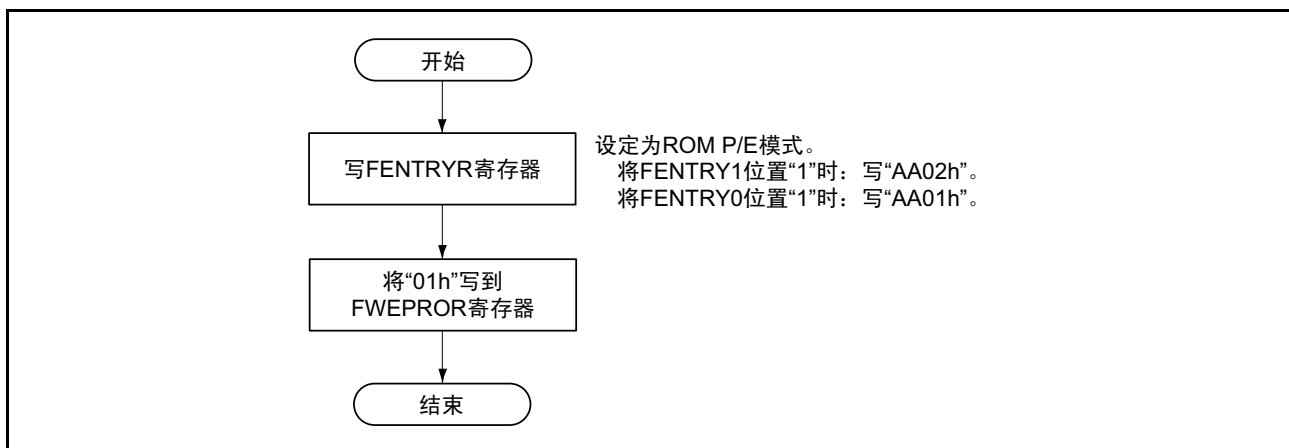


图 26.6 ROM P/E 模式的转移流程

(2) ROM 读模式的转移方法

要高速读 ROM 时，需要将 FENTRYR.FENTRY1 位（注 1）、FENTRYR.FENTRY0 位置“0”，将 FCU 设定为 ROM 读模式。

还需要通过字节存取将“02h”写到 FWEPROR 寄存器，进入不能编程和擦除的状态（参照“26.2.15 闪存编程 / 擦除保护寄存器（FWEPROR）”）。

必须在结束 FCU 的命令处理并且 FCU 未检测到错误的状态下从 ROM P/E 模式转移到 ROM 读模式。

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

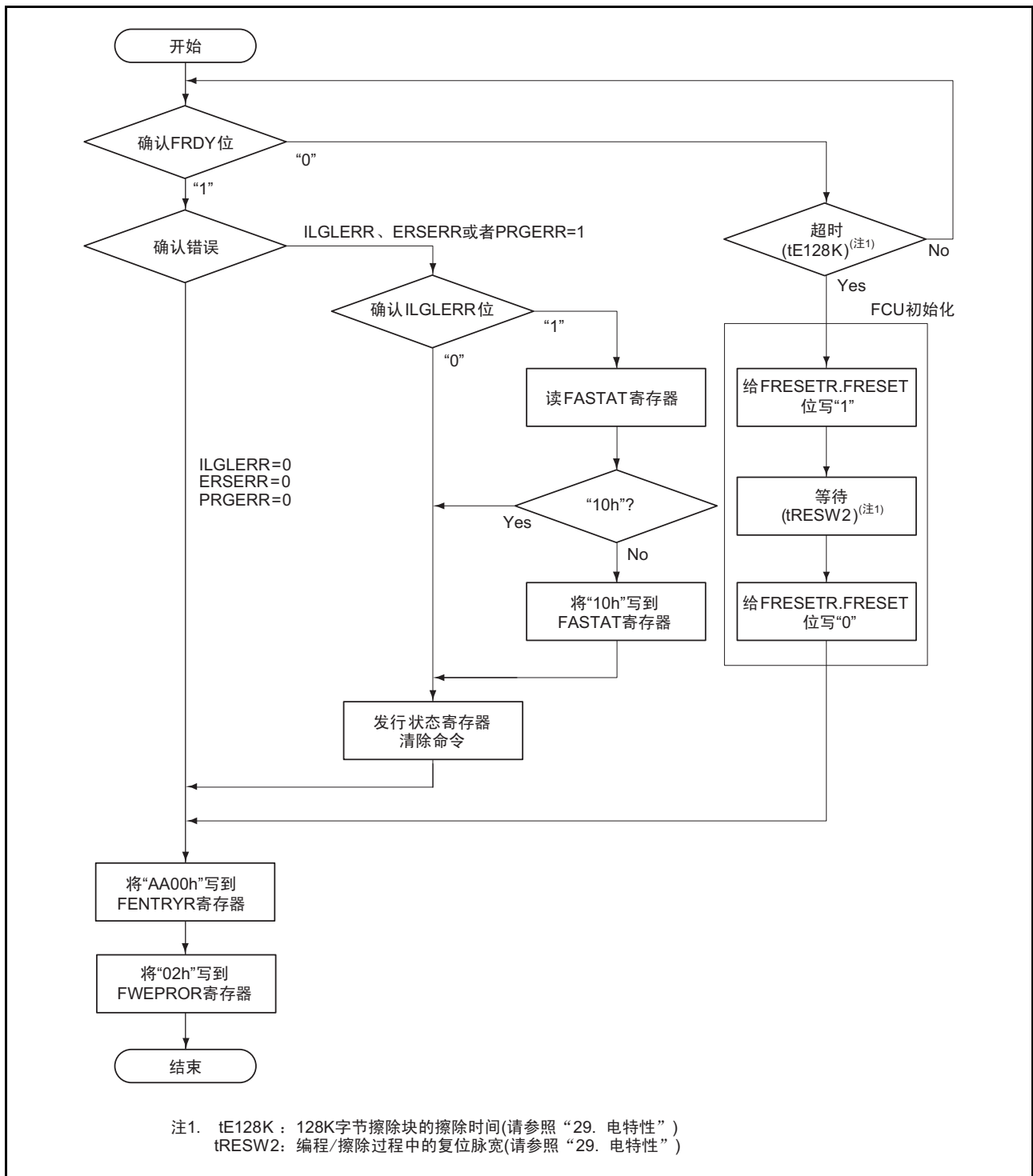


图 26.7 ROM 读模式的转移流程

(3) ROM P/E 正常模式的转移方法

向 ROM P/E 正常模式的转移方法有：在 ROM 读模式中设定 FENTRYR 寄存器的方法（参照“26.6.1 FCU 的模式”）以及在 ROM P/E 模式中发行正常模式转移命令（图 26.8）的方法。通过字节存取将“FFh”写到 ROM 编程/擦除地址，执行正常模式转移命令。

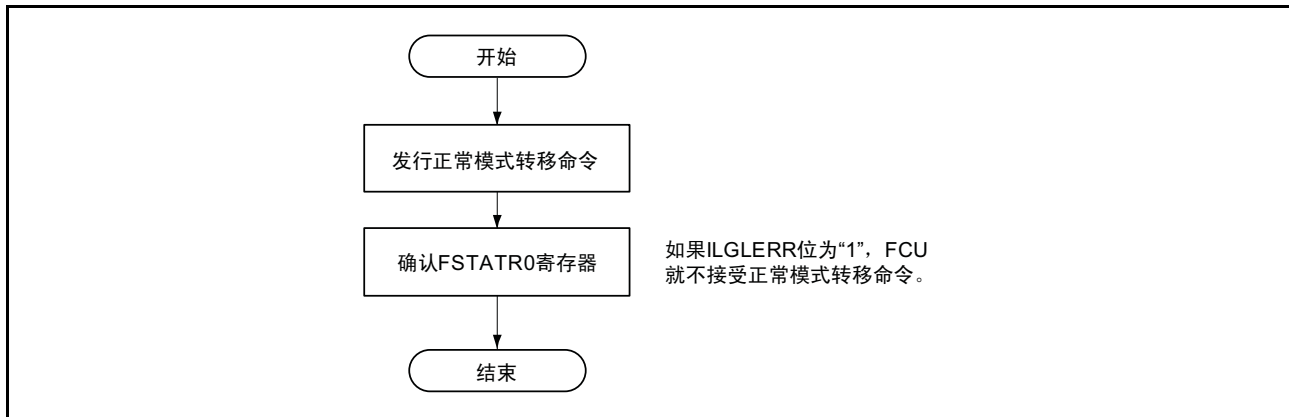


图 26.8 ROM P/E 正常模式的转移流程

(4) ROM 状态读模式的转移方法

如果发行正常模式转移和锁定位读模式转移以外的 FCU 命令，FCU 就转移到 ROM 状态读模式。也能通过发行状态读模式转移命令，转移到 ROM 状态读模式。FSTATR0 寄存器的确认例子如图 26.9 所示。在此例中，通过发行状态读模式转移命令，转移到 ROM 状态读模式，然后对 ROM 编程/擦除地址进行读存取，确认 FSTATR0 的内容。

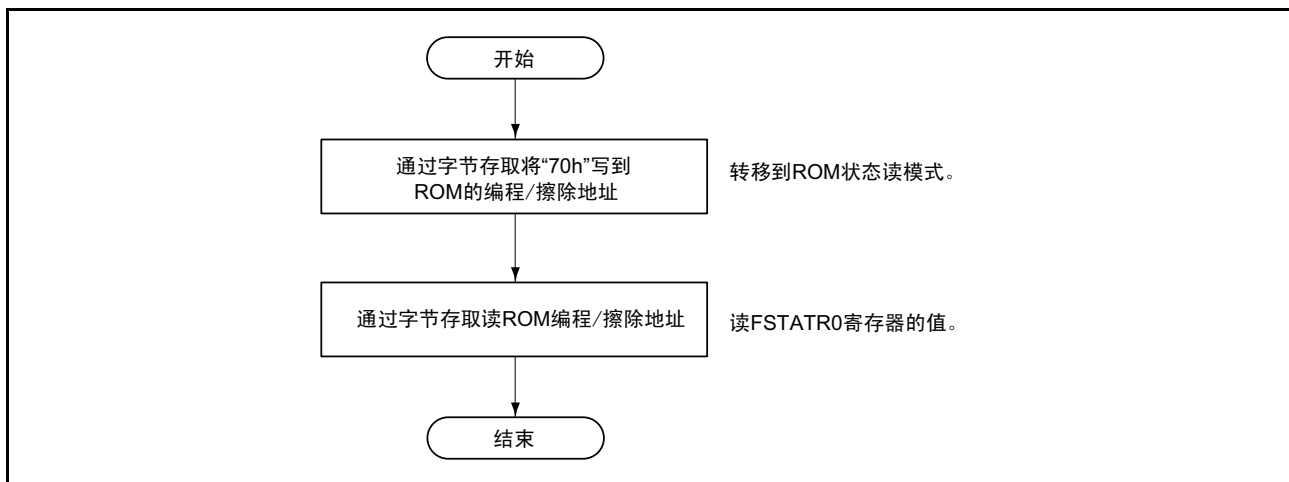


图 26.9 ROM 状态读模式的转移流程以及状态的确认方法

(5) ROM 锁定位读模式的转移方法

在 FMODR.FRDM 位为“0”（存储区读方式）时，通过发行锁定位读模式转移命令（锁定位读 1），转移到 ROM 锁定位读模式。如果在转移到 ROM 锁定位读模式后读 ROM 编程 / 擦除地址，就读与存取对象对应的擦除块的锁定位并且复制到读数据的全部位（图 26.10）。

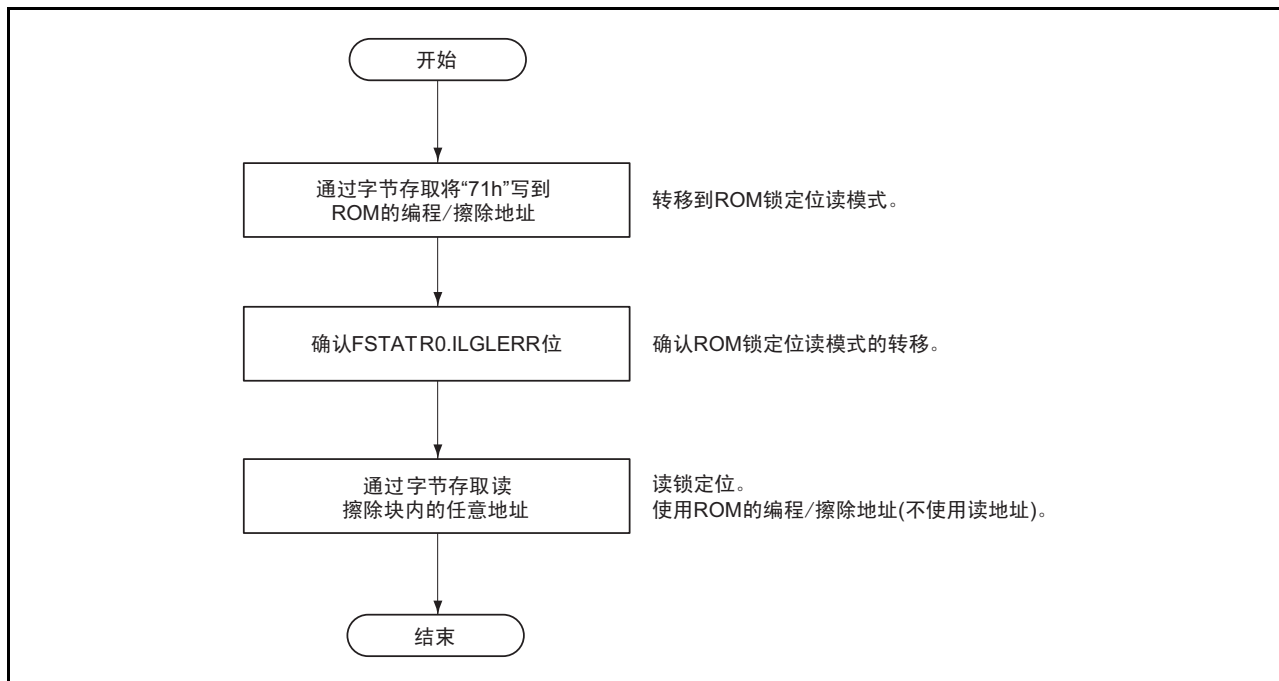


图 26.10 ROM 锁定位读模式的转移流程以及锁定位的读法

26.6.4.2 编程 / 擦除的步骤

在此说明 ROM 的编程 / 擦除流程。有关 FCU 命令的接受条件，请参照“26.6.3 FCU 的模式和命令的关系”。FCU 命令的概略流程如图 26.11 所示。

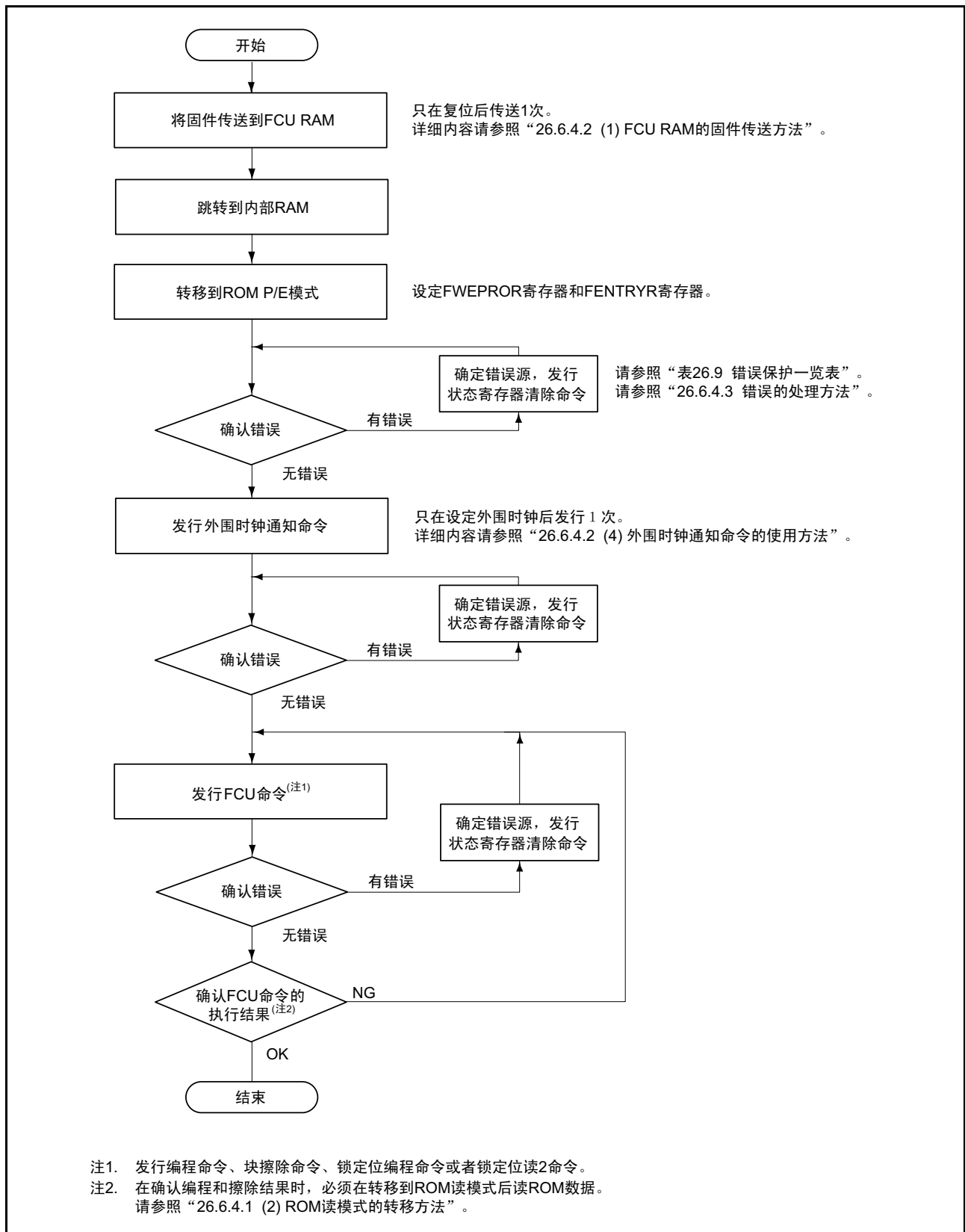


图 26.11 编程 / 擦除处理的概略流程

(1) FCU RAM 的固件传送方法

要使用 FCU 命令时，需要将 FCU 的固件保存到 FCU 的 RAM。因为在启动芯片时没有将 FCU 的固件保存到 FCU 的 RAM，所以需要将保存在 FCU 固件区的 FCU 固件复制到 FCU 的 RAM。当 FSTATR1.FCUEERR 位为“1”时，保存在 FCU RAM 的固件就可能被破坏，因此需要对 FCU 进行复位并且重新复制 FCU 固件。

FCU RAM 的固件传送流程如图 26.12 所示。在给 FCU 的 RAM 写数据时，必须将 FENTRYR 寄存器置“0000h”，使 FCU 停止。有关 DMAC 设定方法的详细内容，请参照“12. DMA 控制器（DMAC）”。

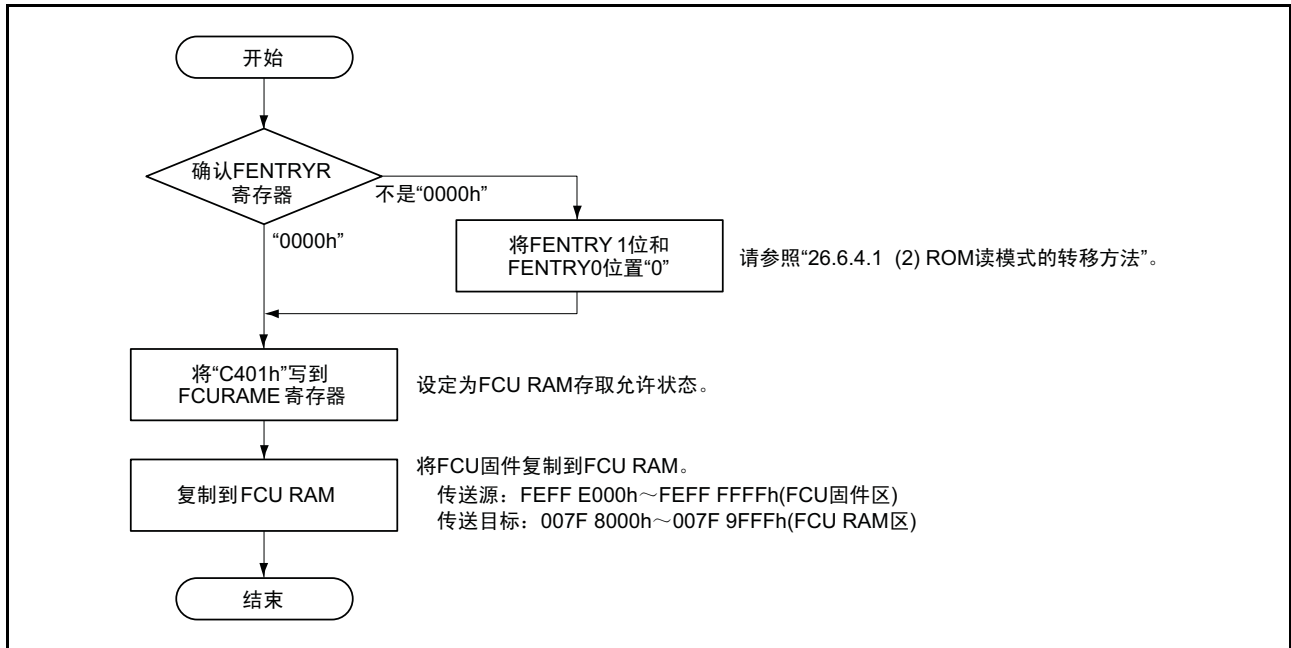


图 26.12 FCU RAM 的固件传送流程

(2) 向内部 RAM 的跳转

在对 ROM 进行编程或者擦除时，因为不对 ROM 取指令，所以需要转移到 ROM 以外的区域。必须在将所需的指令码复制到内部 RAM 后跳转到内部 RAM。

(3) ROM P/E 模式的转移

需要设定 FENTRYR.FENTRY1 位（注 1）、FENTRYR.FENTRY0 位、FWEPROR 寄存器，并且将 FCU 设定为 ROM P/E 模式。详细内容请参照“26.6.4.1 (3) ROM P/E 正常模式的转移方法”。

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

(4) 外围时钟通知命令的使用方法

需要给 PCKAR 寄存器设定在对 ROM 进行编程或者擦除前使用的外围时钟的频率。能设定的频率范围是 8 ~ 50MHz。如果频率不在此范围内，FCU 就检测到错误，进入命令锁定状态（参照“26.8.2 错误保护”）。

在设定 PCKAR 寄存器后使用外围时钟通知命令。在外围时钟通知命令的第 1 周期通过字节存取将“E9h”写到 ROM 的编程/擦除地址，在第 2 周期，通过字节存取将“03h”写到 ROM 的编程/擦除地址。在命令的第 3 周期~第 5 周期，通过字存取进行写操作。此时，起始地址必须使用调整为以 4 字节为边界的地址。在通过 3 次字存取将数据“0F0Fh”写到 ROM 的编程/擦除地址后，如果在第 6 周期通过字节存取将“D0h”写到 ROM 的编程/擦除地址，FCU 就开始进行外围时钟的频率设定处理。能通过 FSTATR0.FRDY 位确认设定的结束。

第 1 周期~第 6 周期中能指定的地址因 FENTRYR.FENTRY1 位（注 1）、FENTRYR.FENTRY0 位的设定而不同，必须指定与 FENTRYR.FENTRY1 位（注 1）、FENTRYR.FENTRY0 位对应的地址。如果因 FENTRYR.FENTRY1 位（注 1）、FENTRYR.FENTRY0 位和指定地址的错误组合而发行命令，FCU 就检测到错误，进入命令锁定状态（参照“26.8.2 错误保护”）。

如果在复位后不更改正在使用的外围时钟的设定，此设定就只需执行 1 次而且对后续的 FCU 命令有效。

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

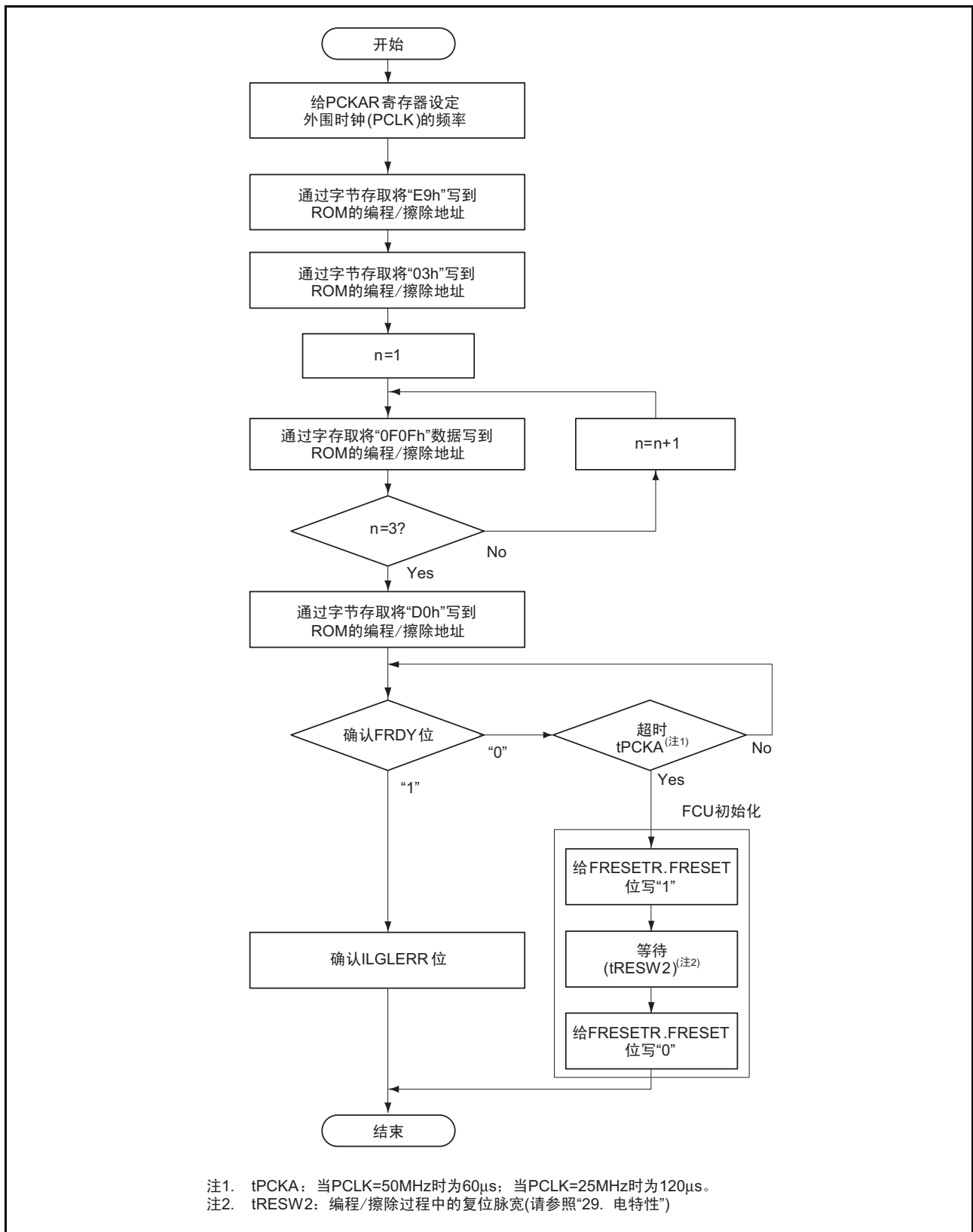


图 26.13 外围时钟通知命令的使用方法

(5) 编程方法

使用编程命令对 ROM 进行数据编程。

在编程命令的第 1 周期，通过字节存取将“E8h”写到 ROM 的编程 / 擦除地址，在第 2 周期，通过字节存取将“80h”写到 ROM 的编程 / 擦除地址。在第 3 周期的存取中，必须通过字存取将编程数据写到编程对象区的起始地址。此时，起始地址必须使用调整为以 256 字节为边界的地址。在第 4 周期~第 130 周期，必须通过 127 次字存取将编程数据写到 ROM 的编程 / 擦除地址。如果在第 131 周期通过字节存取将“D0h”写到 ROM 的编程 / 擦除地址，FCU 就开始对 ROM 进行编程处理。能通过 FSTATR0.FRDY 位确认编程的结束。

第 1 周期~第 131 周期中能指定的地址因 FENTRYR.FENTRY1 位（注 1）、FENTRYR.FENTRY0 位的设定而不同，必须指定与 FENTRYR.FENTRY1 位（注 1）、FENTRYR.FENTRY0 位对应的地址。如果因 FENTRYR.FENTRY1 位（注 1）、FENTRYR.FENTRY0 位和指定地址的错误组合而发行命令，FCU 就检测到错误，进入命令锁定状态（参照“26.8.2 错误保护”）。

如果在第 3 周期~第 130 周期中存取的区域包括不需要编程的地址，就必须将该地址的编程数据置“FFFFh”。如果要在将通过锁定位进行的保护设定为无效后进行编程，就必须将 FPROTR.FPROTCN 位置“1”。

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

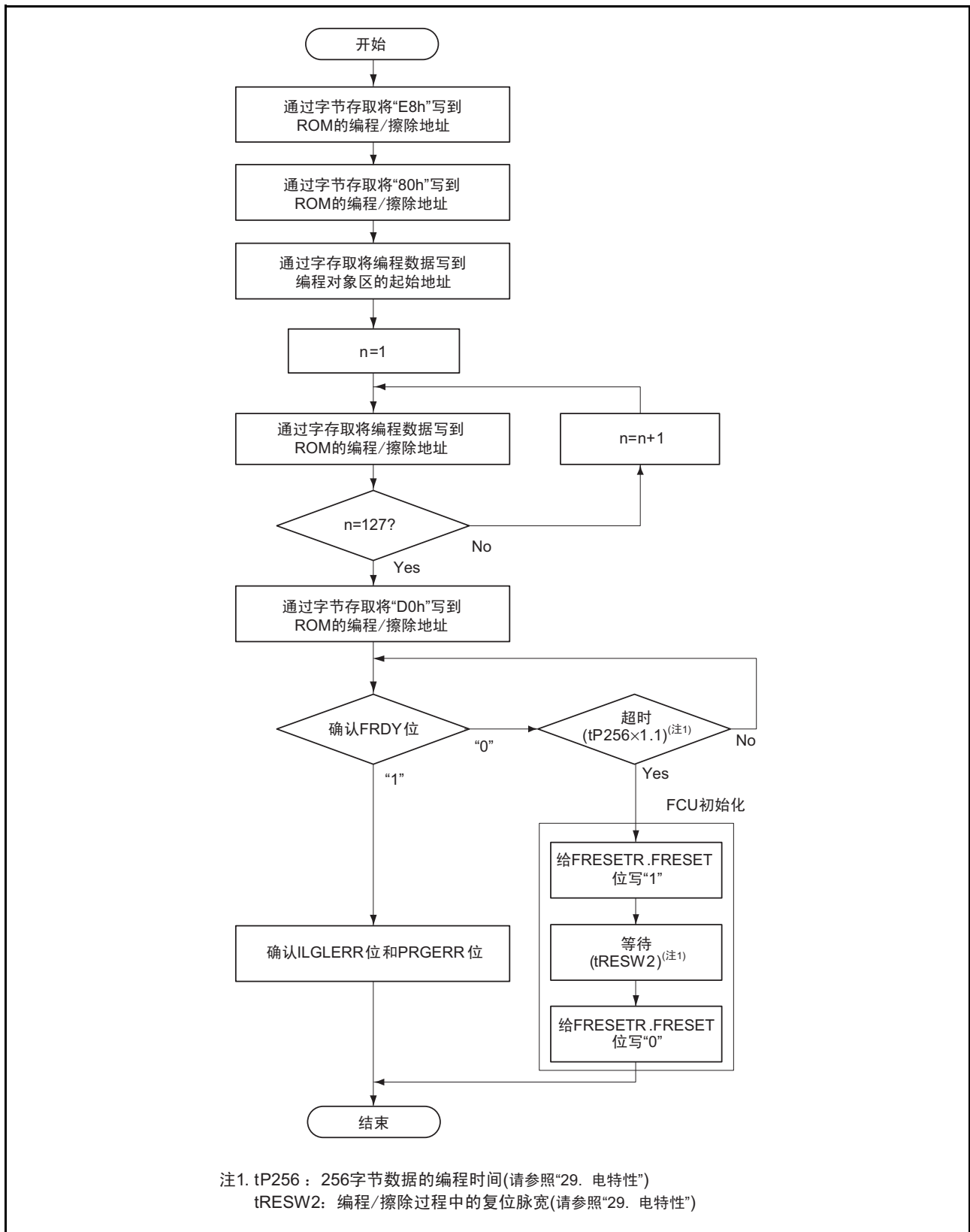


图 26.14 ROM 的编程方法

(6) 擦除方法

使用块擦除命令来擦除 ROM。

在块擦除命令的第 1 周期，通过字节存取将 “20h” 写到 ROM 的编程 / 擦除地址。如果在第 2 周期通过字节存取将 “D0h” 写到擦除对象块内的任意地址，FCU 就开始对 ROM 进行擦除处理。能通过 FSTATR0.FRDY 位确认擦除的结束。

如果要在将通过锁定位进行的保护设定为无效后进行擦除，就必须将 FPROTR.FPROTCN 位置 “1”。

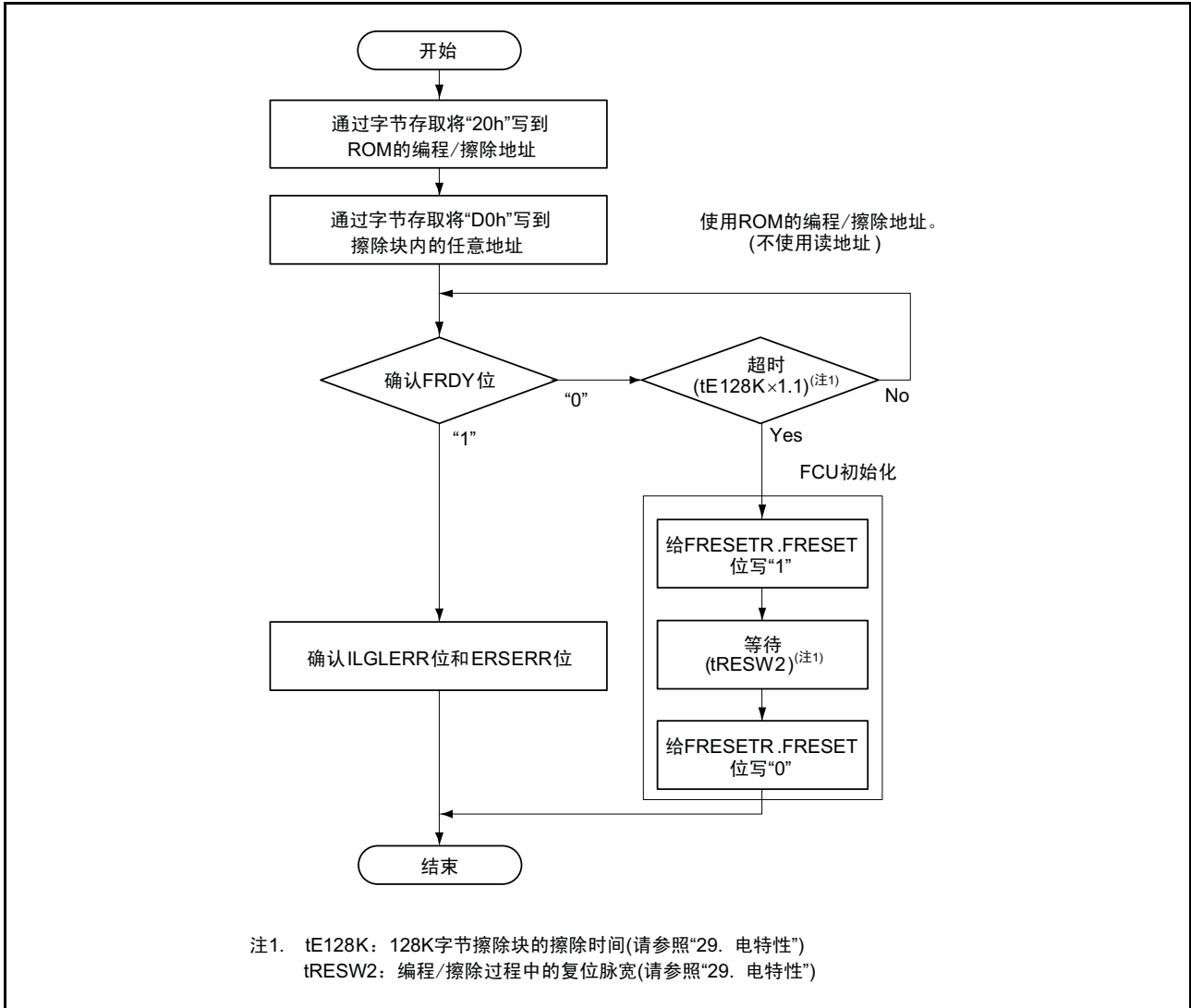


图 26.15 ROM 的擦除方法

(7) 锁定位的编程 / 擦除方法

用户 MAT 的各擦除块内置锁定位。使用锁定位编程命令对锁定位进行编程。在锁定位编程命令的第 1 周期，通过字节存取将“77h”写到 ROM 的编程 / 擦除地址。如果在第 2 周期通过字节存取将“D0h”写到要进行锁定位编程的擦除块内的任意地址，FCU 就开始进行锁定位的编程处理。能通过 FSTATR0.FRDY 位确认编程的结束。

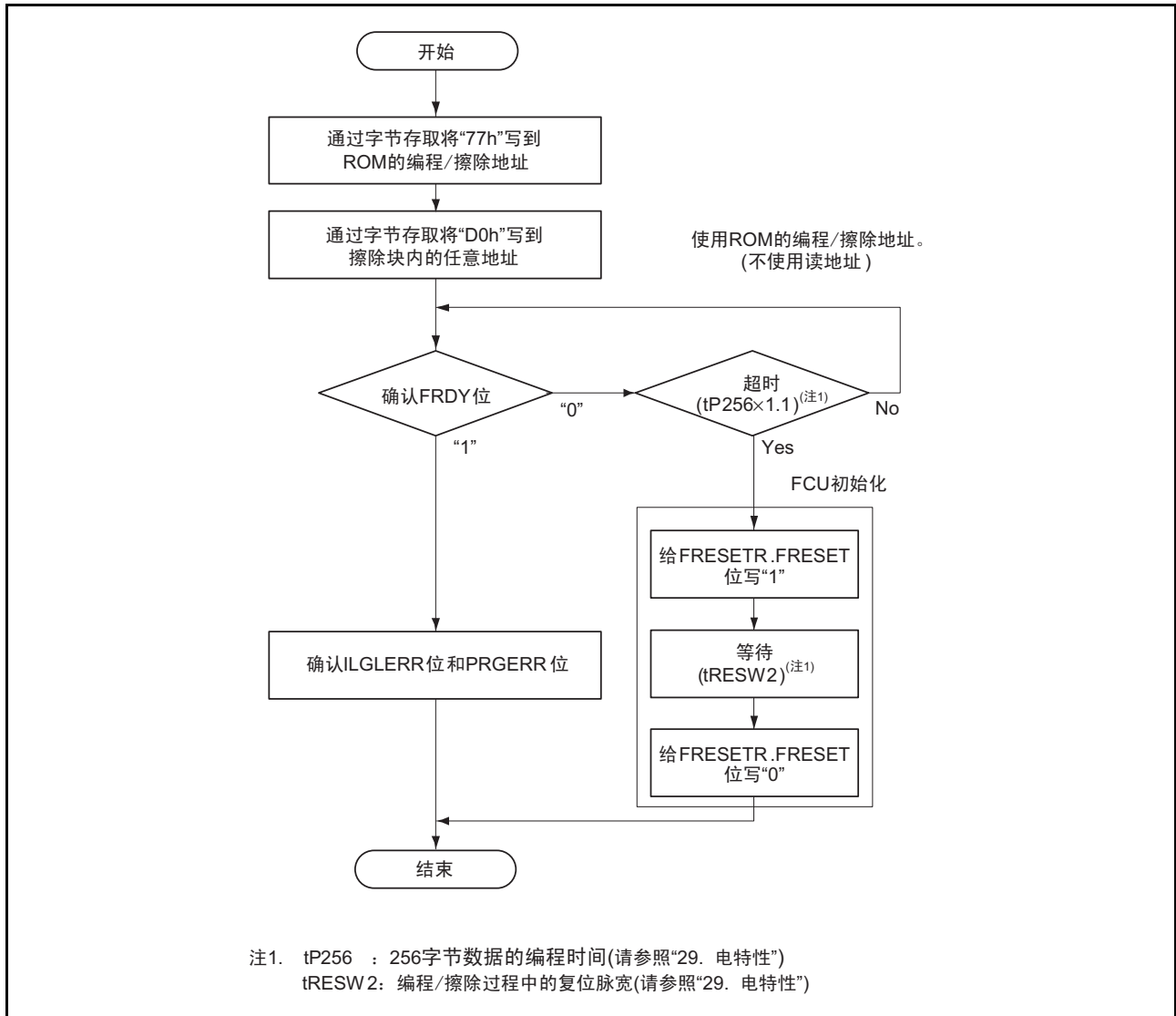


图 26.16 锁定位的编程设定方法

使用块擦除命令来擦除锁定位。

在 FPROTR.FPROTCN 位为“0”的状态下，不能擦除锁定位为“0”的擦除块。要擦除锁定位时，必须在将 FPROTCN 位置“1”的状态下发行块擦除命令。如果使用块擦除命令，擦除块内的全部数据就被擦除，而不能只擦除锁定位。

(8) 锁定位的读法

锁定位的读法有存储区读方式和寄存器读方式。

寄存器读方式（FMODR.FRDM位为“1”）使用锁定位读2命令，对要读锁定位的擦除块的编程/擦除地址发行锁定位读2命令。如果在锁定位读2命令的第1周期，通过字节存取写“71h”，在第2周期，通过字节存取写“D0h”，就将对应的擦除块的锁定位复制到FSTATR1.FLOCKST位。

通过转移到锁定位读模式并且读ROM的编程/擦除地址来执行存储区读方式（FMODR.FRDM位为“0”）。有关详细内容请参照“26.6.4.1 (5) ROM 锁定位读模式的转移方法”。

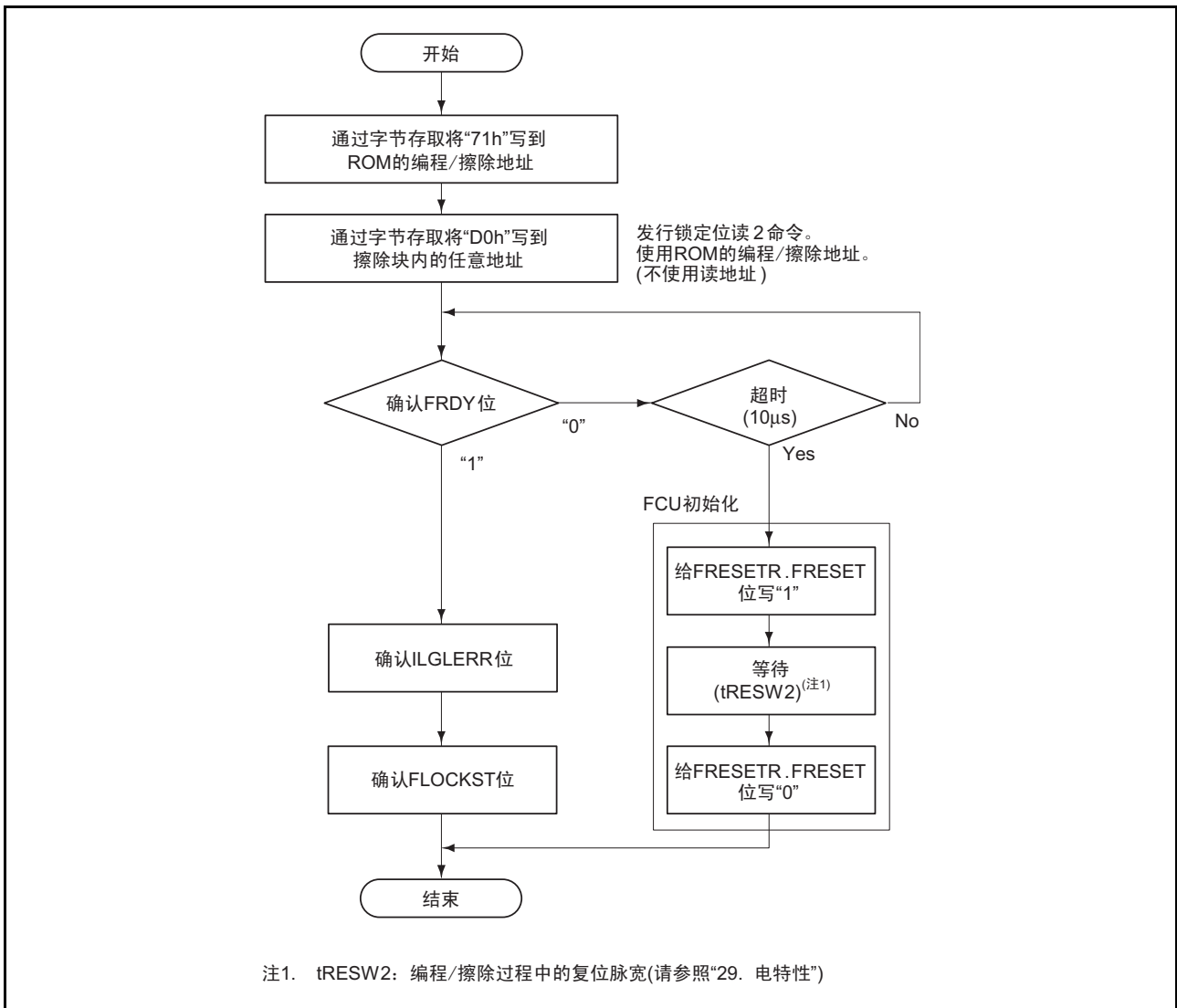


图 26.17 通过寄存器读方式读锁定位的方法

26.6.4.3 错误的处理方法

以下说明发生错误时的处理方法，各种错误的内容请参照“26.8 保护”。

(1) 闪存状态寄存器 0（FSTATR0）的确认方法

FSTATR0 寄存器的确认方法有直接读 FSTATR0 寄存器的方法和在 ROM 状态读模式中读 ROM 的编程 / 擦除地址的方法。

在 ROM 状态读模式中的读法请参照“26.6.4.1 (4) ROM 状态读模式的转移方法”。

(2) 闪存状态寄存器 0（FSTATR0）的清除方法

使用状态寄存器清除命令将 FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位和 FSTATR0.PRGERR 位置“0”。

当 FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位或者 FSTATR0.PRGERR 位为“1”时，FCU 进入命令锁定状态，不接受状态寄存器清除命令以外的 FCU 命令。当 ILGLERR 位为“1”时，也必须确认 FASTAT.ROMAE 位、FASTAT.DFLAE 位、FASTAT.DFLRPE 位和 DFLWPE 位的值。即使不将这些位置“0”而发行状态寄存器清除命令，ILGLERR 位也不变为“0”。

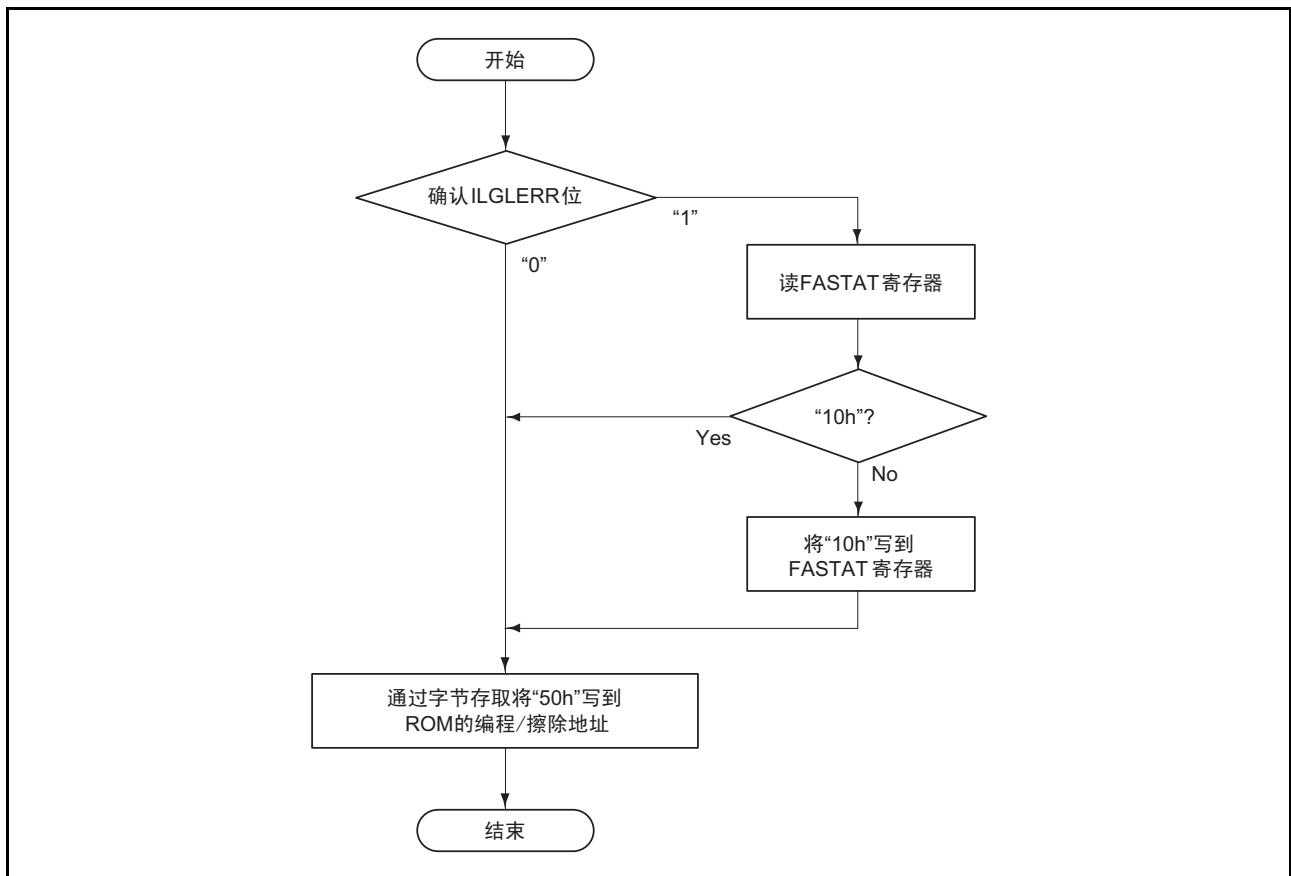


图 26.18 FSTATR0 寄存器的清除方法

(3) FCU 的初始化方法

在发行 FCU 命令后，如果因超时而使 FSTATR0.FRDY 位不变为“1”，就需要通过 FRESETR 寄存器对 FCU 进行初始化。在 FSTATR1.FCUERR 位为“1”时，也需要通过 FRESETR 寄存器对 FCU 进行初始化。无论在何种情况下，FRESETR.FRESETR 位为“1”的状态都必须保持 tRESW2 的时间（参照“29. 电特性”）。在 FRESETR 位保持“1”的期间，禁止读 ROM/ 数据闪存。因为在 FRESETR 位为“1”的状态下 FENTRYR 寄存器被初始化，所以不能使用 FCU 命令，必须重新进行图 26.11 的处理。

26.6.4.4 挂起 / 恢复

(1) 编程 / 擦除的挂起方法

使用 P/E 挂起命令来中止 ROM 的编程或者擦除。

要发行 P/E 挂起命令时，必须预先通过检查 FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位和 FSTATR1.FCUERR 位为“0”，确认正常进行了编程或者擦除处理。要确认能接受挂起命令时，也必须确认 FSTATR0.SUSRDY 位为“1”。在发行 P/E 挂起命令后，必须读 FSTATR0 寄存器和 FSTATR1 寄存器，确认没有发生错误。

如果在编程或者擦除处理过程中发生异常，ILGLERR 位、PRGERR 位、ERSERR 位和 FCUERR 位中至少有 1 位变为“1”。在确认 SUSRDY 位为“1”后到接受 P/E 挂起命令的期间，如果编程或者擦除处理已经结束，发行的 P/E 挂起命令就被检测为非法命令，因此 ILGLERR 位变为“1”。

如果在接受 P/E 挂起命令的同时编程 / 擦除处理结束，就不产生错误，也不转移到挂起状态（FSTATR0.FRDIY 位为“1”并且 FSTATR0.ERSSPD 位和 FSTATR0.PRGSPPD 位为“0”）。如果在接受 P/E 挂起命令并且编程 / 擦除的中止处理正常结束，FCU 就转移到挂起状态，FRDIY 位变为“1”并且 ERSSPD 位或者 PRGSPPD 位变为“1”。在发行 P/E 挂起命令后，必须通过检查 ERSSPD 位或者 PRGSPPD 位为“1”，确认转移到挂起状态，然后决定后续的流程。即使未转移到挂起状态，如果在后续的流程中发行 P/E 恢复命令，也产生非法命令错误，FCU 转移到命令锁定状态（参照“26.8.2 错误保护”）。

如果转移到擦除挂起状态，就能对擦除对象以外的块进行编程。在编程或者擦除挂起状态下，能通过清除 FENTRYR 寄存器转移到 ROM 读模式。

有关接受 P/E 挂起命令时的 FCU 运行内容，请参照“26.7 挂起”。

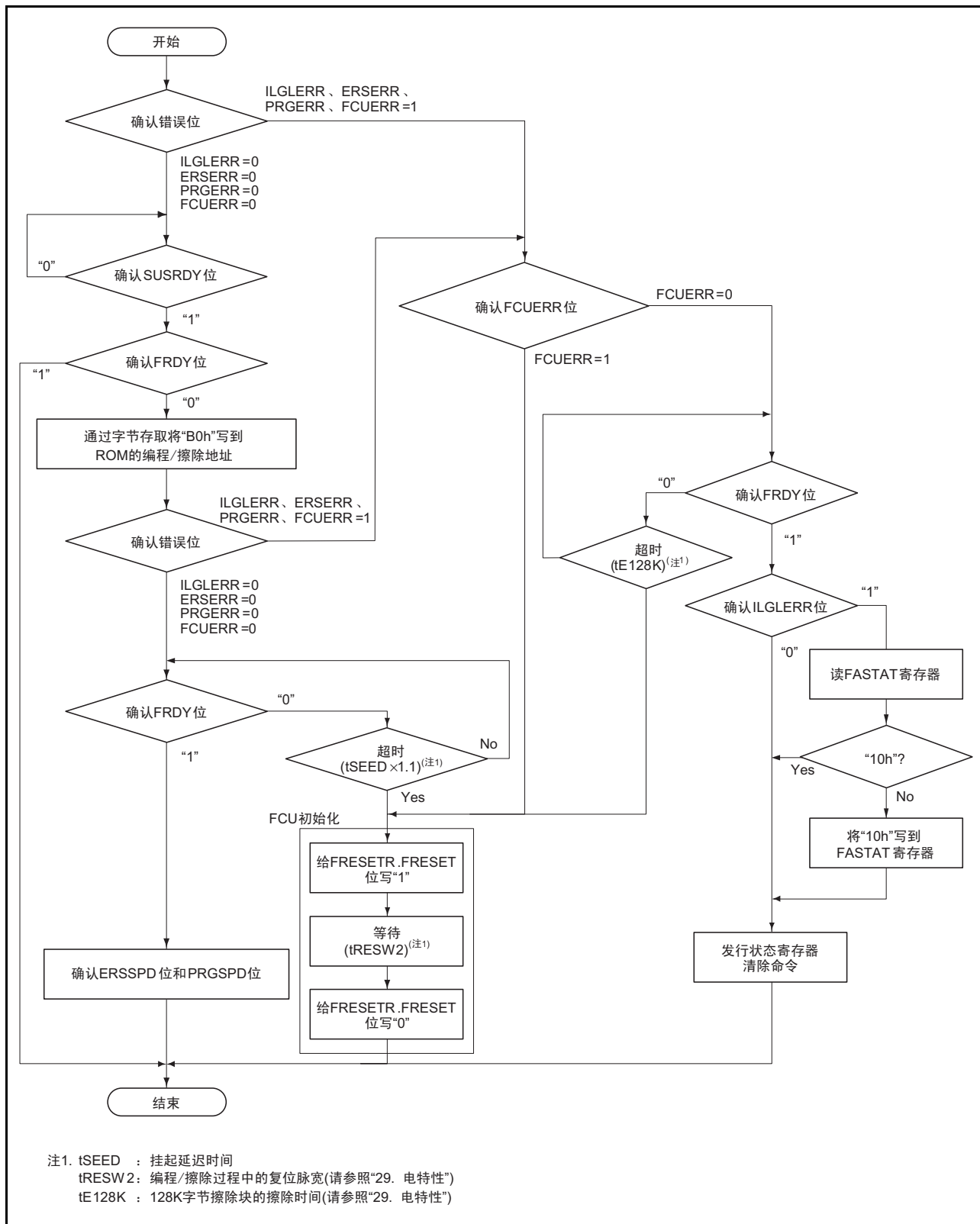


图 26.19 编程 / 擦除的挂起方法

(2) 编程 / 擦除的恢复方法

使用 P/E 恢复命令重新开始挂起的编程或者擦除处理。如果在挂起过程中更改了 FENTRYR 寄存器的设定，就必须在发行 P/E 恢复命令前将发行 P/E 挂起命令前的值重新设定到 FENTRYR 寄存器。

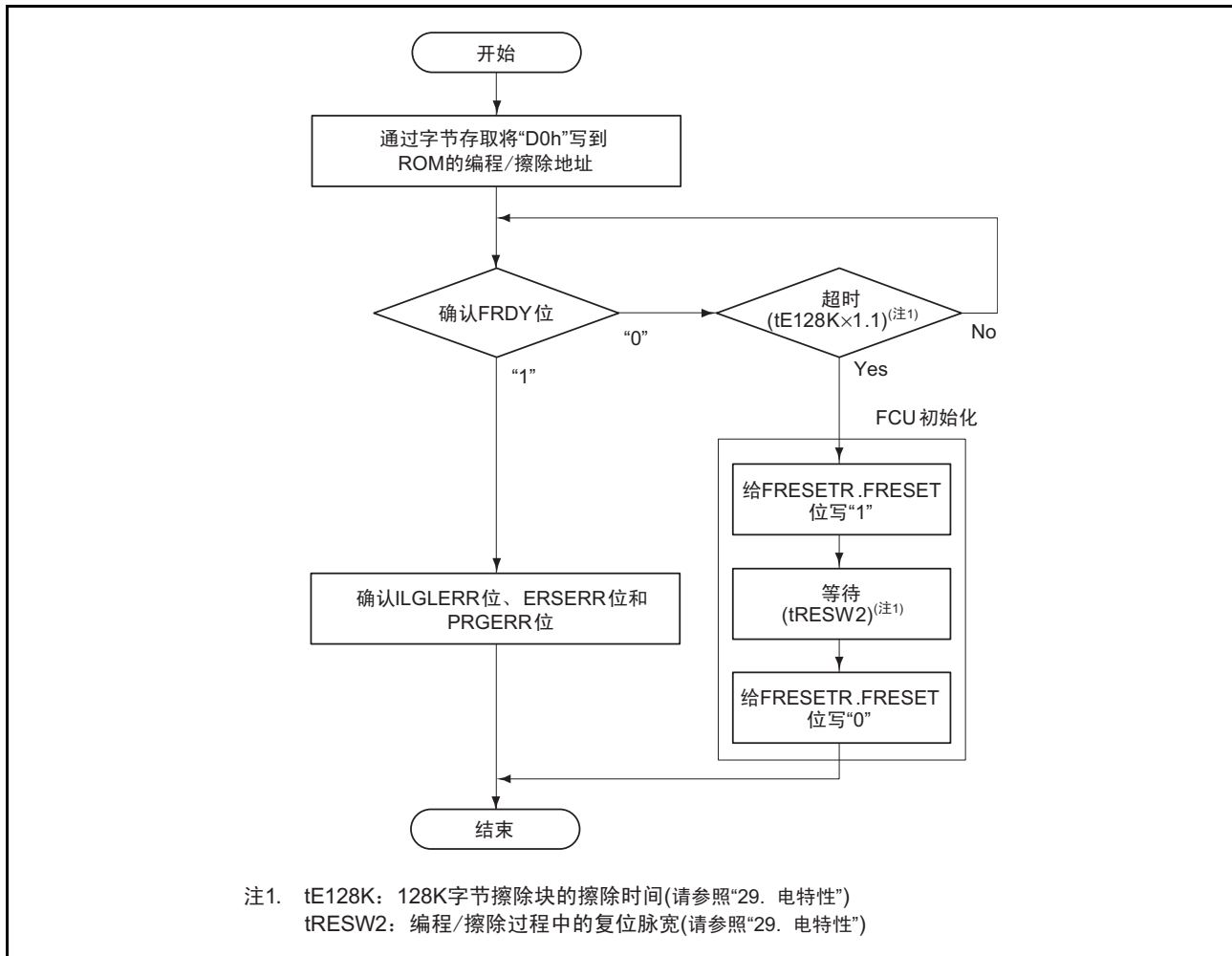


图 26.20 编程 / 擦除的恢复方法

26.7 挂起

不能在编程或者擦除处理过程中读 ROM。通过发行 P/E 挂起命令来中止 ROM 的编程或者擦除处理，变为能读 ROM 的状态。P/E 挂起命令有 1 种编程模式和 2 种擦除模式（挂起优先模式和擦除优先模式），还有用于重新开始被中止的编程或者擦除处理的 P/E 恢复命令。

26.7.1 编程时的挂起

如果在对 ROM 进行编程或者擦除过程中发行 P/E 挂起命令，FCU 就中止编程处理。编程处理的中止如图 26.21 所示。

如果 FCU 接受编程命令，就将 FSTATR0.FRDY 位置“0”，开始编程处理。如果在开始编程处理后 FCU 转移到能接受 P/E 挂起命令的状态，FSTATR0.SUSRDY 位就变为“1”。如果发行 P/E 挂起命令，FCU 就在接受挂起命令后将 SUSRDY 位置“0”。如果 FCU 在外加编程脉冲过程中接受 P/E 挂起命令，FCU 就继续外加脉冲。如果超过规定的脉冲外加时间，FCU 就结束脉冲的外加，并且在开始进行编程的中止处理后将 FSTATR0.PRGSPPD 位置“1”。一旦中止处理结束，FCU 就将 FRDY 位置“1”，转移到编程挂起状态。如果 FCU 在编程挂起状态下接受 P/E 恢复命令，就将 FRDY 位和 PRGSPPD 位置“0”，重新开始编程处理。

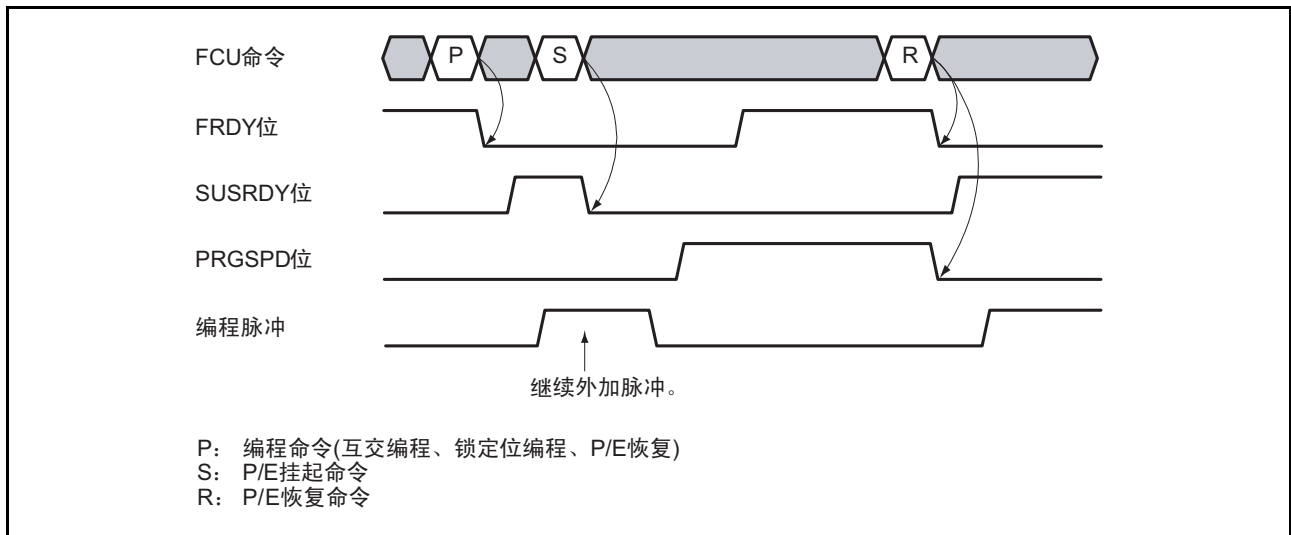


图 26.21 编程处理的中止

26.7.2 擦除时的挂起（挂起优先模式）

擦除挂起模式为挂起优先模式（FCPSR.ESUSPMD 位为“0”）时的擦除处理的中止如图 26.22 所示。

如果 FCU 接受擦除命令，就将 FSTAT0.FRDY 位置“0”，开始进行擦除处理。如果在开始擦除处理后 FCU 转移到能接受 P/E 挂起命令的状态，FSTAT0.SUSRDY 位就变为“1”。如果发行 P/E 挂起命令，FCU 就在接受挂起命令后将 SUSRDY 位置“0”。如果在擦除处理过程中接受挂起命令，即使在外加擦除脉冲过程中，FCU 也开始中止处理，然后将 FSTAT0.ERSSPD 位置“1”。一旦中断处理结束，FCU 就将 FRDY 位置“1”，转移到擦除挂起状态。如果 FCU 在擦除挂起状态下接受 P/E 恢复命令，就将 FRDY 位和 ERSSPD 位置“0”，重新开始擦除处理。

擦除挂起模式的设定影响擦除脉冲的控制方式。在挂起优先模式中，如果 FCU 在外加过去没有被中止的擦除脉冲 A 过程中接受 P/E 挂起命令，就在中止外加擦除脉冲 A 后转移到擦除挂起状态。通过 P/E 恢复命令重新开始擦除并且重新外加擦除脉冲 A，如果在此期间 FCU 接受 P/E 挂起命令，FCU 就继续外加擦除脉冲 A。如果超过规定的脉冲外加时间，FCU 就在结束擦除脉冲的外加后转移到擦除挂起状态。接着，如果在 FCU 接受 P/E 恢复命令并且开始外加新的擦除脉冲 B 后，FCU 再次接受 P/E 挂起命令，就中止擦除脉冲 B 的外加。在挂起优先模式中，因为以 1 个脉冲中止 1 次擦除脉冲的外加并且优先挂起处理，所以能缩短挂起的延迟时间。

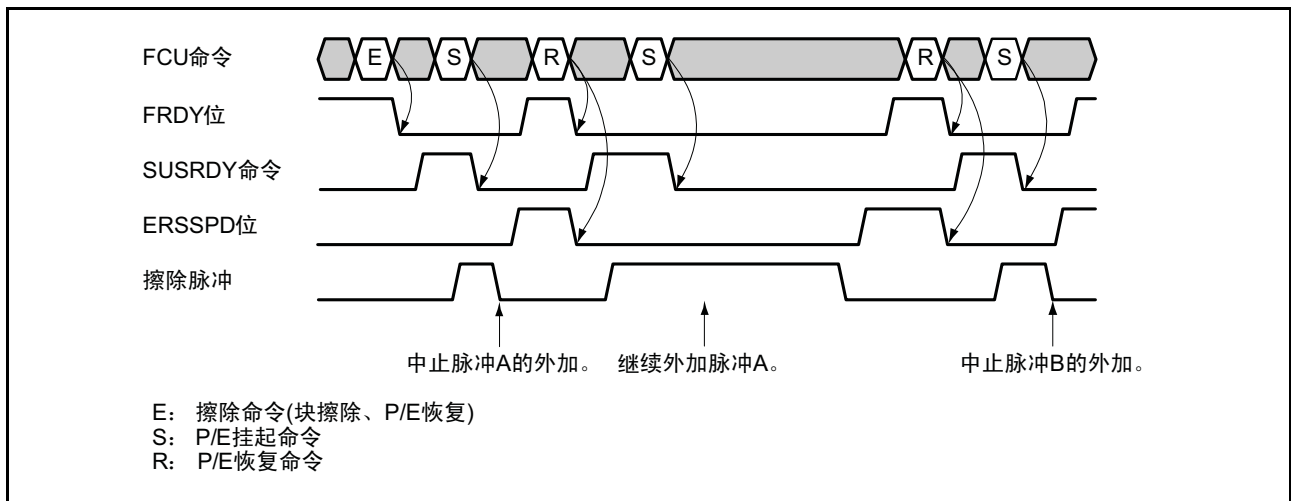


图 26.22 擦除处理的中止（挂起优先模式）

26.7.3 擦除时的挂起（擦除优先模式）

擦除优先模式（FCPSR.ESUSPMD 位为“1”）时的擦除处理的中止如图 26.23 所示。擦除优先模式的擦除脉冲控制方式和编程中止处理的编程脉冲控制方式相同。

如果 FCU 在外加擦除脉冲过程中接受 P/E 挂起命令，就继续外加擦除脉冲。在此模式中，因为在发行 P/E 恢复命令时不会再次外加擦除脉冲，所以与挂起优先模式相比，能缩短整个擦除处理所需的时间。

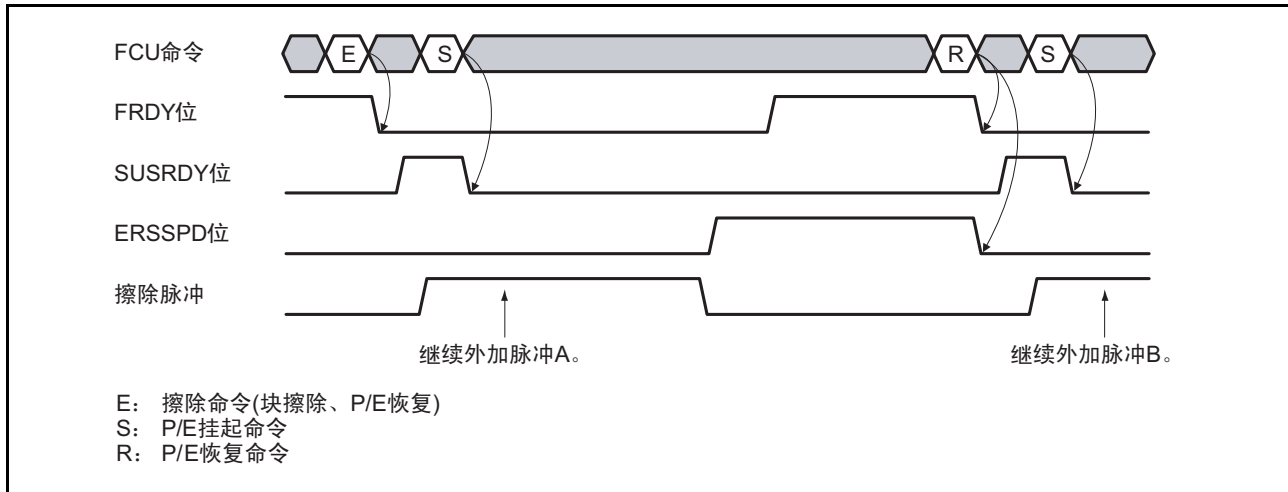


图 26.23 擦除处理的中止（擦除优先模式）

26.8 保护

ROM 的编程 / 擦除的保护有软件保护和错误保护。

26.8.1 软件保护

软件保护是通过设定控制寄存器和用户 MAT 的锁定位来禁止对 ROM 进行编程 / 擦除的状态。如果违反软件保护，发行 ROM 的编程 / 擦除命令，FCU 就检测到错误，进入命令锁定状态。

(1) 通过 FWEPROR 寄存器进行的保护

如果不将 FWEPROR.FLWE[1:0] 位置“01b”，在任何模式中都不能进行改写。

(2) 通过 FENTRYR 寄存器进行的保护

当 FENTRYR.FENTRY1 位（注 1）、FENTRYR.FENTRY0 位为“0”时，FCU 为 ROM 读模式。因为在 ROM 读模式中不接受 FCU 命令，所以 ROM 为禁止编程和擦除的状态。如果在 ROM 读模式中发行 FCU 命令，FCU 就检测到非法命令错误，进入命令锁定状态（参照“26.8.2 错误保护”）。

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

(3) 通过锁定位进行的保护

用户 MAT 的各擦除块内置锁定位。当 FPROTR.FPROTCN 位为“0”时，锁定位为“0”的擦除块为禁止编程和擦除的状态。要对锁定位为“0”的擦除块进行编程或者擦除时，必须将 FPROTCN 位置“1”。如果违反通过锁定位进行的保护，发行 ROM 的编程 / 擦除命令，FCU 就检测到编程 / 擦除错误，进入命令锁定状态（参照“26.8.2 错误保护”）。

26.8.2 错误保护

错误保护是因误发行 FCU 命令以及发生禁止的存取而导致 FCU 在检测到误动作后禁止接受 FCU 命令的状态（命令锁定状态）。

一旦 FCU 进入命令锁定状态（FASTAT.CMDLK 位为“1”），就将状态位（FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位、FSTATR1.FCUERR 位和 FASTST.ROMAE 位）中的某位或者多个位置“1”，禁止对 ROM 进行编程和擦除。要解除命令锁定状态时，需要在 FASTAT 寄存器为“10h”的状态下发行状态寄存器清除命令。

当 FAEINT.CMDLKIE 位为“1”时，如果 FCU 进入命令锁定状态（FASTAT.CMDLK 位为“1”），就产生闪存接口错误（FIFERR）中断；当 FAEINT.ROMAEIE 位为“1”时，即使 FASTAT.ROMAE 位为“1”，也产生 FIFERR 中断。

ROM 相关的错误保护内容和错误检测时的状态位的值（FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位、FSTATR1.FCUERR 位和 FASTAT.ROMAE 位）的关系如表 26.10 所示。如果在编程或者擦除处理过程中发行挂起以外的命令，就转移到命令锁定状态，但是 FCU 继续进行编程或者擦除处理。在此状态下无法通过发行 P/E 挂起命令来中止编程或者擦除。如果在命令锁定状态下发行命令，ILGLERR 位就变为“1”。

表 26.10 错误保护一览表（ROM 专用 +ROM/ 数据闪存通用）

| 分类 | 内容 | ILGLERR | ERSERR | PRGERR | FCUERR | ROMAE | CMDLK |
|---------------|--|---------|--------|--------|--------|-------|-------|
| FENTRYR 的设置错误 | 将 FENTRYR.FENTRYD 位、FENTRYR.FENTRY1 位（注1）、FENTRYR.FENTRY0 位中的多个位置“1”。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 挂起和恢复时的 FENTRYR 寄存器的设定不同。 | 1 | 0 | 0 | 0 | 0 | 1 |
| 非法命令错误 | 在 FCU 命令的第 1 周期指定未定义代码。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 在多个周期的 FCU 命令的最后周期指定“D0h”以外的数据。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 给 PCKAR 寄存器设定 8 ~ 50MHz 频率以外的外围时钟。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 在编程或者擦除处理过程中发行挂起以外的命令。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 在编程和擦除以外的处理过程中发行挂起命令。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 在挂起状态下发行挂起命令。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 在挂起以外的状态下发行恢复命令。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 在编程挂起状态下发行编程 / 擦除（编程、锁定编程、块擦除）命令。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 在擦除挂起状态下发行块擦除命令。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 在擦除挂起状态下发行擦除挂起对象区的编程命令 / 锁定编程命令。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 在编程命令的第 2 个周期指定“80h”以外的数据。 | 1 | 0 | 0 | 0 | 0 | 1 |
| 在命令锁定状态下发行命令。 | 1 | 0/1 | 0/1 | 0/1 | 0/1 | 1 | |
| 擦除错误 | 在擦除处理过程中发生错误。 | 0 | 1 | 0 | 0 | 0 | 1 |
| | 在 FPROTR.FPROTCN 位为“0”时，对锁定位为“0”的擦除块发行块擦除命令。 | 0 | 1 | 0 | 0 | 0 | 1 |
| 编程错误 | 在编程处理过程中发生错误。 | 0 | 0 | 1 | 0 | 0 | 1 |
| | 在 FPROTR.FPROTCN 位为“0”时，对锁定位为“0”的擦除块发行编程命令或者锁定编程命令。 | 0 | 0 | 1 | 0 | 0 | 1 |
| FCU 错误 | 在 FCU 内部处理过程中发生错误。 | 0 | 0 | 0 | 1 | 0 | 1 |
| ROM 存取违反 | 在 FENTRYR.FENTRY1 位（注1）为“1”时并且在 ROM P/E 正常模式中，对 00E0 0000h ~ 00EF FFFFh 发行读存取命令。 | 1 | 0 | 0 | 0 | 1 | 1 |
| | 在 FENTRYR.FENTRY0 位为“1”时并且在 ROM P/E 正常模式中，对 00F0 0000h ~ 00FF FFFFh 发行读存取命令。 | 1 | 0 | 0 | 0 | 1 | 1 |
| | 在 FENTRYR.FENTRY1 位（注1）为“0”时，对 00E0 0000h ~ 00EF FFFFh 发行存取命令。 | 1 | 0 | 0 | 0 | 1 | 1 |
| | 在 FENTRYR.FENTRY0 位为“0”时，对 00F0 0000h ~ 00FF FFFFh 发行存取命令。 | 1 | 0 | 0 | 0 | 1 | 1 |
| | 在 FENTRYR 寄存器不为“0000h”的状态下，对 FFE0 0000h ~ FFFF FFFFh 发行读存取命令。 | 1 | 0 | 0 | 0 | 1 | 1 |

注 1. ROM 容量小于等于 1M 字节的产品不能使用。

26.9 用户引导模式

如果通过 MD1 引脚和 MD0 引脚设定为用户引导模式并且进行复位解除，就转移到用户引导模式。此时的复位向量为用户引导 MAT 的地址 FF7F FFFCh。其他的向量请参照标准的向量表（参照“10. 中断控制器（ICU）”）。

在用户引导模式中，能建立使用任意接口的编程程序，并且能通过发行 FCU 命令对用户 MAT 或者数据 MAT 进行编程或者擦除。必须在引导模式中对用户引导 MAT 进行编程。

26.10 引导模式

26.10.1 系统结构

在引导模式中，能从主机发送控制命令和编程数据，对用户 MAT、用户引导 MAT 或者数据 MAT 进行编程或者擦除。为了进行主机和 RX610 之间的通信，将内部 SCI 用于异步模式。需要准备用于将控制命令发送到主机的工具和编程数据。

如果通过引导模式启动 RX610，就执行嵌入式程序存储 MAT 中的程序，此程序在自动调整 SCI 的位速率以及接受主机的控制命令后，控制编程和擦除。

引导模式的系统结构如图 26.24 所示。

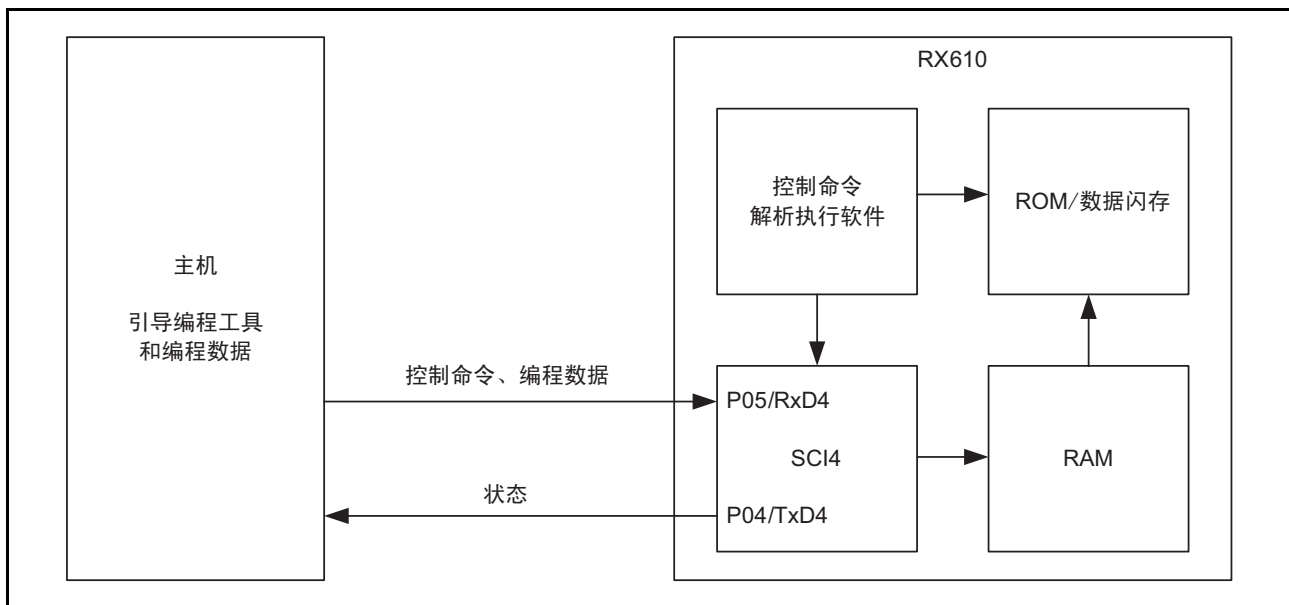


图 26.24 引导模式的系统结构

26.10.2 ID 码保护

这是禁止从主机读、编程和擦除的功能。

使用写在 ROM 中的控制码和 ID 码，判断 ID 码保护的有效或者无效以及 ID 码保护。在 ID 码保护有效的情况下，只在从主机送来的代码和 ROM 中的控制码及 ID 码相同时，允许读、编程和擦除。

ROM 中的控制码和 ID 码是 32 位长的 4 字数据，控制码和 ID 码的结构如图 26.25 所示。必须以 32 位为单位设定 ID 码。

| | 31 | 24 | 23 | 16 | 15 | 8 | 7 | 0 |
|------------|-------|----|-------|----|-------|---|-------|---|
| FFFF FFA0h | 控制码 | | ID码1 | | ID码2 | | ID码3 | |
| FFFF FFA4h | ID码4 | | ID码5 | | ID码6 | | ID码7 | |
| FFFF FFA8h | ID码8 | | ID码9 | | ID码10 | | ID码11 | |
| FFFF FFACH | ID码12 | | ID码13 | | ID码14 | | ID码15 | |

图 26.25 ROM 中的控制码和 ID 码的结构

(1) 控制码

控制码决定 ID 码保护的有效或者无效以及主机的验证方法，控制码和验证方法如表 26.11 所示。

表 26.11 ID 码保护的规格

| 控制码 | ID 码 | 保护状态 | 连接 SCI 时的运行 |
|------|---|------------------|---|
| 45h | 任意 | 保护有效 (验证方法 1) | ID 码相同：结束 ID 码保护并且转移到主机命令等待状态。 ID 码不同：再次转移到 ID 码等待状态。但是，如果连续 3 次 ID 码不同，就进行全部擦除。 |
| 52h | 50h、72h、6Fh、74h、65h、 63h、74h、FFh、……、FFh 除外 | 保护有效 (验证方法 2) | ID 码相同：结束 ID 码保护并且转移到主机命令等待状态。 ID 码不同：再次转移到 ID 码等待状态。 |
| | 50h、72h、6Fh、74h、65h、 63h、74h、FFh、……、FFh | 保护有效 (验证方法 3) | 总是判断为 ID 码不同。 |
| 上述以外 | — | 保护无效 | 擦除全部的块。 |

(2) ID 码

ID 码能设定为任意的值。如果控制码为“52h”并且 ID 码从 ID 码 1 开始按顺序设定“50h”、“72h”、“6Fh”、“74h”、“65h”、“63h”、“74h”、“FFh”、……、“FFh”，就不判断 ID 码是否相同而总是视为不同，禁止从主机读、编程和擦除。

(3) 设定 ID 码的编程例子

控制码为“45h”，ID 码从 ID 码 1 开始按顺序设定“01h”、“02h”、“03h”、“04h”、“05h”、“06h”、“07h”、“08h”、“0Ah”、“0Bh”、“0Ch”、“0Dh”、“0Eh”、“0Fh”时的编程例子：

```
.SECTION ID_CODE, CODE
.ORG 0FFFFFFA0h
.LWORD 45010203h
.LWORD 04050607h
.LWORD 08090A0Bh
.LWORD 0C0D0E0Fh
```

26.10.3 引导模式的状态转移

引导模式的状态转移图如图 26.26 所示。

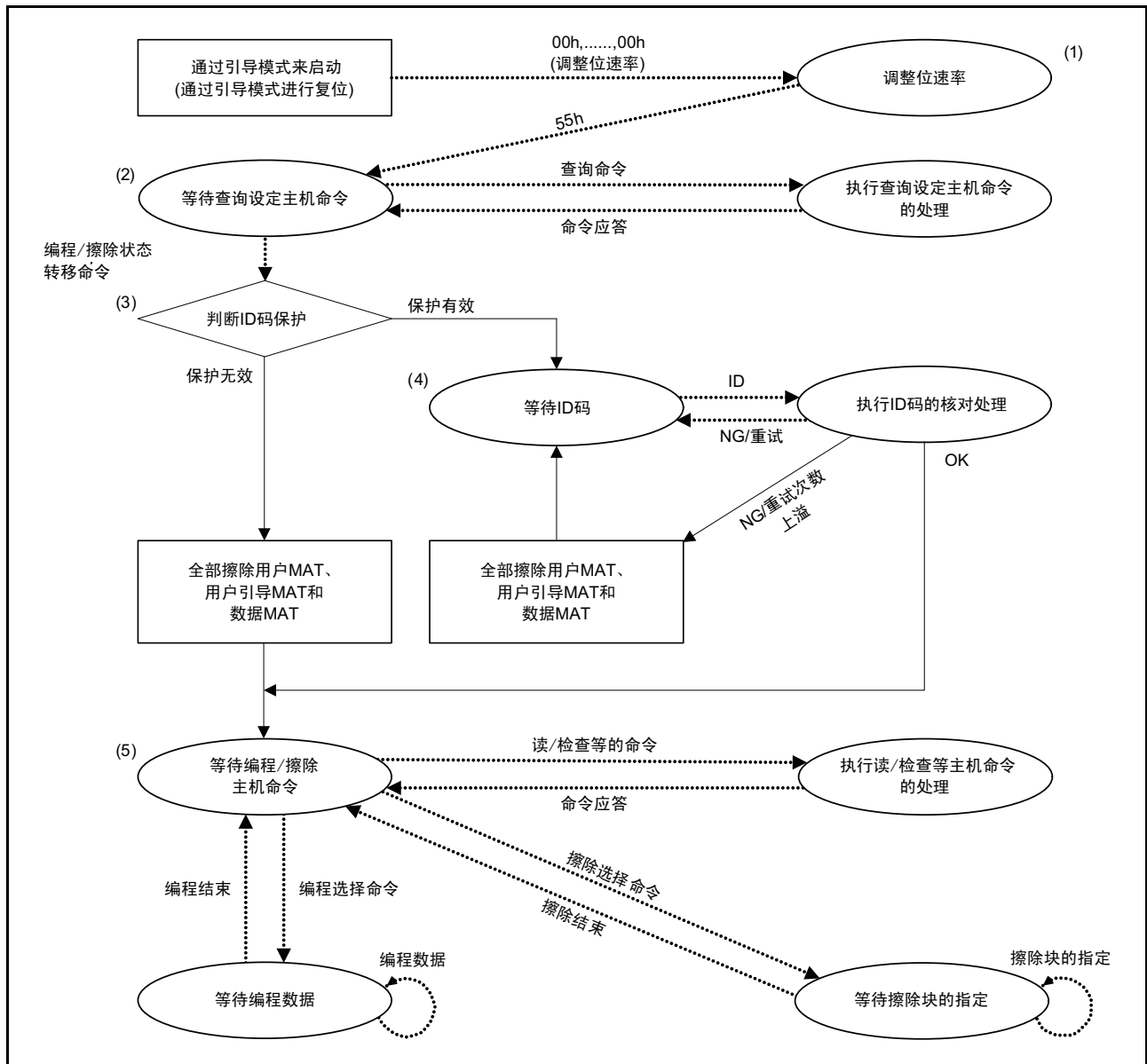


图 26.26 引导模式的状态转移图

(1) 位速率的调整

如果通过引导模式启动 RX610，就自动调整主机和 SCI 的位速率。如果位速率的自动调整结束，就将“00h”从 RX610 发送到主机。此后，如果 RX610 正常接受从主机送来的“55h”，就转移到查询设定主机命令等待状态。有关位速率调整的详细内容，请参照“26.10.4 位速率的自动调整”。

(2) 查询设定主机命令的等待

这是查询 MAT 大小、MAT 结构、MAT 起始地址和支持状况等以及选择器件、时钟模式和位速率的状态。如果从主机发行编程 / 擦除状态转移命令，就转移到判断 ID 码保护的有效或者无效的状态。有关查询设定主机命令的详细内容，请参照“26.10.5 查询设定主机命令的等待状态”。

(3) ID 码保护的判断

判断 ID 码保护的有效或者无效。通过写在 ROM 中的控制码和 ID 码，判断 ID 码保护的有效或者无效。当 ID 码保护有效时，转移到 ID 码等待状态；当 ID 码保护无效时，全部擦除用户 MAT、用户引导 MAT 和数据 MAT，并且转移到编程 / 擦除主机命令等待状态。有关控制码和 ID 码的详细内容，请参照“26.10.2 ID 码保护”。

(4) ID 码的等待

等待从主机送来的控制码和 ID 码。将主机送来的控制码、ID 码和 ROM 中的代码进行比较，如果相同，就转移到编程 / 擦除主机命令等待状态；如果不同，就返回到 ID 码等待状态。如果连续 3 次不同并且保护状态为验证方法 1，就进行全部擦除并且再次返回到 ID 码等待状态。要解除此不同的状态时，需要进行复位。有关控制码和 ID 码的详细内容，请参照“26.10.2 ID 码保护”。

(5) 编程 / 擦除主机命令的等待

这是根据主机的命令进行编程或者擦除的状态。根据 RX610 接收的命令，转移到编程数据等待状态、擦除块指定等待状态、读 / 检查等命令处理执行状态。

如果 RX610 接收编程选择命令，就转移到编程数据等待状态。主机在发送编程选择命令后，必须接着发送编程起始地址和编程数据。如果将编程起始地址设定为 FFFF FFFFh，就在编程结束后，从编程数据等待状态转移到编程 / 擦除命令等待状态。

如果 RX610 接收擦除选择命令，就转移到擦除块指定等待状态。主机在发送擦除选择命令后，必须接着发送擦除块号。如果将擦除块号设定为“FFh”，就在擦除结束后，从擦除块指定等待状态转移到编程 / 擦除命令等待状态。因为在通过引导模式启动后到转移到编程 / 擦除主机命令状态的期间，用户 MAT、用户引导 MAT 和数据 MAT 被全部擦除，所以在引导模式中除了不进行复位而要擦除新编程的数据的情况以外，不需要进行擦除。

除了编程 / 擦除命令以外，还有用户 MAT 和用户引导 MAT 的校验和命令、空白检查（擦除检查）命令、存储器读命令、状态信息取命令等主机命令。

26.10.4 位速率的自动调整

如果通过引导模式启动 RX610，就测量从主机连续发送的异步 SCI 通信数据“00h”的 Low 电平期间。在测量 Low 电平期间时，主机的 SCI 发送 / 接收格式必须为 8 位数据、1 个停止位、无奇偶校验，位速率必须设定为 9,600bps 或者 19,200bps。从测量的 Low 电平期间计算主机的 SCI 的位速率，如果位速率调整结束，RX610 就将“00h”发送到主机。如果主机正常接收到“00h”，就必须将“55h”从主机发送到 RX610。否则，就必须通过引导模式重新启动 RX610，并且重新自动调整位速率。如果 RX610 正常接收“55h”，就发送“E6h”。否则，就发送“FFh”。



图 26.27 自动调整位速率时的 SCI 发送 / 接收格式



图 26.28 主机和 RX610 之间的通信顺序

位速率的自动调整取决于主机的 SCI 的位速率和 RX610 的外围时钟的频率，因为有可能无法进行正常的调整，所以必须在表 26.12 所示的条件下进行 SCI 的通信。

表 26.12 可自动调整位速率的条件

| 主机的 SCI 的位速率 | EXTAL 的频率范围 |
|--------------|-------------|
| 9,600bps | 8 ~ 14MHz |
| 19,200bps | 8 ~ 14MHz |

26.10.5 查询设定主机命令的等待状态

在查询设定主机命令等待状态下能使用的主机命令一览表如表 26.13 所示。在编程 / 擦除主机命令等待状态下，也能使用嵌入式程序状态查询命令。其他命令只能在查询设定主机命令等待状态下使用。

表 26.13 查询设定主机命令

| 主机命令名 | 功能 |
|---------------|----------------------------------|
| 支持器件查询 | 查询器件码和嵌入式程序的产品型号。 |
| 器件选择 | 选择器件码。 |
| 时钟模式查询 | 查询时钟模式数和各自的值。 |
| 时钟模式选择 | 通知已选择的时钟模式。 |
| 倍增比查询 | 查询时钟的种类、倍增比 / 分频比的种类以及倍增比 / 分频比。 |
| 工作频率查询 | 查询时钟的种类、最大工作频率和最小工作频率。 |
| 用户引导 MAT 信息查询 | 查询用户引导 MAT 的个数以及起始地址和结束地址。 |
| 用户 MAT 信息查询 | 查询用户 MAT 的个数以及起始地址和结束地址。 |
| 擦除块信息查询 | 查询块数以及起始地址和结束地址。 |
| 编程长度查询 | 查询编程时的数据长度。 |
| 新位速率选择 | 更改主机和 RX610 之间的 SCI 通信的位速率。 |
| 编程 / 擦除状态转移 | 转移到 ID 码保护判断状态。 |
| 嵌入式程序状态查询 | 查询处理状态。 |

如果主机发送未定义的命令，RX610 就发送命令错误应答。命令错误应答的内容如下所示。在命令中保存主机发送的命令的起始字节。

| | | |
|------|-----|----|
| 错误应答 | 80h | 命令 |
|------|-----|----|

在查询设定主机命令等待状态下，必须参考查询命令的应答，按照器件的选择 → 时钟模式的选择 → 新位速率的选择的顺序从主机发送选择命令，设定 RX610。在发行时钟模式选择命令前，不能使用支持器件的查询和时钟模式的查询以外的查询命令。如果搞错命令的发送顺序，RX610 就发送命令错误应答。查询设定主机命令等待状态下的主机命令使用例子图 26.33 所示。

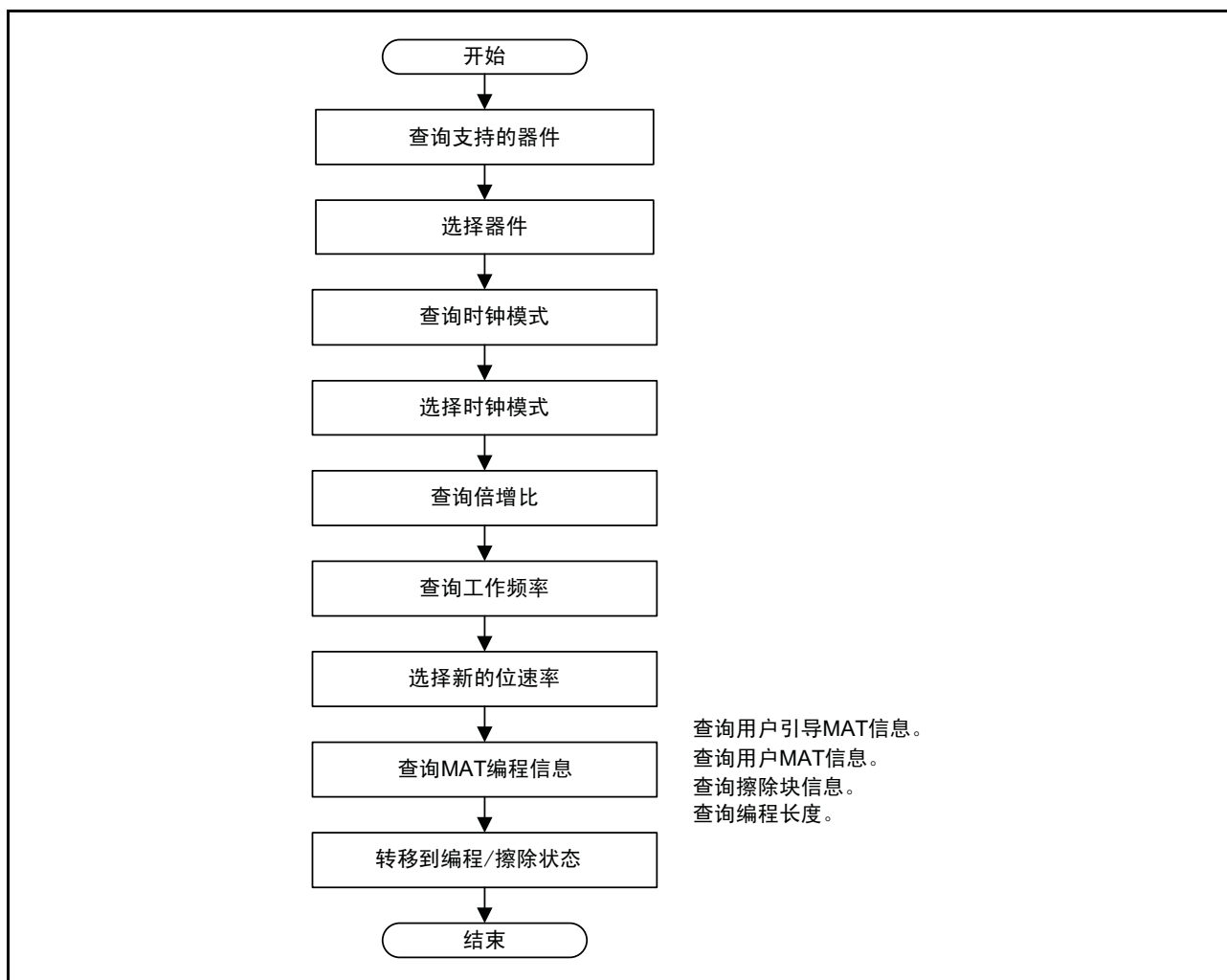


图 26.29 用户 MAT/ 用户引导 MAT 的查询设定主机命令的使用例子

以下说明各主机命令的详细内容。说明内容中的“命令”是从主机发送到 RX610 的命令，“应答”是从 RX610 发送到主机的应答。“校验和”是指在合计 RX610 群发送的各字节时，为了得到“00h”而算出的字节数据。

(1) 支持器件查询

如果主机发送支持器件查询命令，RX610 就通过引导模式的嵌入式程序发送能支持的器件信息。如果在主机选择器件后发送支持器件查询命令，RX610 就只发送所选的器件信息。

命令

| |
|-----|
| 20h |
|-----|

| | | | | |
|----|-----|-----|-----|------|
| 应答 | 30h | 长度 | 器件数 | |
| | 字符数 | 器件码 | | 产品名称 |
| | 字符数 | 器件码 | | 产品名称 |
| | ... | ... | | ... |
| | 字符数 | 器件码 | | 产品名称 |
| | SUM | | | |

| | | |
|--------|------------|---------------------------|
| 【符号说明】 | 长度（1 字节） | : 器件数、字符数、器件码和产品名称的数据总字节数 |
| | 器件数（1 字节） | : 引导模式的嵌入式程序支持的品种数 |
| | 字符数（1 字节） | : 器件码和产品名称的字符数 |
| | 器件码（4 字节） | : 芯片产品名称的 ASCII 码 |
| | 产品名称（n 字节） | : 支持器件名称的 ASCII 码 |
| | SUM（1 字节） | : 校验和 |

(2) 器件选择

如果主机发送器件选择命令，RX610 就检查所指定的器件是否为能支持的器件。如果是能支持的器件，RX610 就将支持器件更改为指定的器件并且发送应答（06h）。如果不是能支持的器件或者送来的命令为非法命令，RX610 就发送错误应答（90h）。

即使在支持器件查询结果的器件数为“01h”时，也必须通过器件选择命令设定查询结果的器件码的值。

命令

| | | | |
|-----|----|-----|-----|
| 10h | 长度 | 器件码 | SUM |
|-----|----|-----|-----|

应答

| |
|-----|
| 06h |
|-----|

错误应答

| | |
|-----|----|
| 90h | 错误 |
|-----|----|

| | | |
|--------|-----------|--------------------------------------|
| 【符号说明】 | 长度（1 字节） | : 器件码的字符数（是固定值，为“4”。） |
| | 器件码（4 字节） | : 芯片产品名称的 ASCII 码（和支持器件查询命令的应答相同的代码） |
| | SUM（1 字节） | : 校验和 |
| | 错误（1 字节） | : 错误码 |
| | | 11h: 校验和错误（命令为非法命令） |
| | | 21h: 器件码不同 |

(3) 时钟模式查询

如果主机发送时钟模式查询命令，RX610 就发送可选择的时钟模式。如果在主机选择时钟模式后发送时钟模式查询命令，RX610 就只发送所选的时钟模式的信息。

| | | | | |
|----|-----|----|-----|----|
| 命令 | 21h | | | |
| 应答 | 31h | 长度 | | |
| | 模式 | 模式 | ... | 模式 |
| | SUM | | | |

| | | |
|--------|-----------|--------------------------|
| 【符号说明】 | 长度（1 字节） | : 模式数、模式的数据总字节数 |
| | 模式（1 字节） | : 可选择的时钟模式（例：01h 时钟模式 1） |
| | SUM（1 字节） | : 校验和 |

(4) 时钟模式选择

如果主机发送时钟模式选择命令，RX610 就检查所指定的时钟模式是否为能支持的模式。如果是能支持的模式，RX610 就将时钟模式更改为指定的模式并且发送应答（06h）。如果不是能支持的模式或者送来的命令为非法命令，RX610 就发送错误应答（91h）。

必须在发送器件选择命令后发送时钟模式选择命令。即使在时钟模式查询结果的时钟模式数为“00h”或者“01h”时，也必须通过时钟模式选择命令设定查询结果的模式的值。

| | | | | |
|------|-----|----|----|-----|
| 命令 | 11h | 长度 | 模式 | SUM |
| 应答 | 06h | | | |
| 错误应答 | 91h | 错误 | | |

| | | |
|--------|-----------|---------------------------|
| 【符号说明】 | 长度（1 字节） | : 模式的字符数（是固定值，为“1”。） |
| | 模式（1 字节） | : 时钟模式（和时钟模式查询命令的应答相同的模式） |
| | SUM（1 字节） | : 校验和 |
| | 错误（1 字节） | : 错误码 |
| | | 11h: 校验和错误（命令为非法命令） |
| | | 22h: 时钟模式不同 |

(5) 倍增比查询

如果主机发送倍增比查询命令，RX610 就发送时钟的种类、倍增比 / 分频比的种类以及倍增比 / 分频比的信息。

| | | | | |
|----|--------|-----|-----|-----|
| 命令 | 22h | | | |
| 应答 | 32h | 长度 | 时钟数 | |
| | 倍增比的种类 | 倍增比 | 倍增比 | ... |
| | 倍增比的种类 | 倍增比 | 倍增比 | ... |
| | ... | ... | ... | ... |
| | 倍增比的种类 | 倍增比 | 倍增比 | ... |
| | SUM | | | |

- 【符号说明】
- 长度（1 字节）：时钟数、倍增比的种类、倍增比的数据总字节数
 - 时钟数（1 字节）：时钟的种类（例：02h 系统时钟和外围时钟）
 - 倍增比的种类（1 字节）：可选择的倍增比 / 分频比的种类
（例：04h 系统时钟有 1 倍增、2 倍增、4 倍增、8 倍增共 4 种）
 - 倍增比（1 字节）：倍增比（例：04h=4 4 倍增）← 用正数指定
分频比（例：FEh=-2 2 分频）← 用负数指定
 - SUM（1 字节）：校验和

(6) 工作频率查询

如果主机发送工作频率查询命令，RX610 就发送各时钟的工作频率的最小值和最大值的消息。

| | | |
|----|------|------|
| 命令 | 23h | |
| 应答 | 33h | 长度 |
| | 最小频率 | 最大频率 |
| | 最小频率 | 最大频率 |
| | ... | ... |
| | 最小频率 | 最大频率 |
| | SUM | |

- 【符号说明】
- 长度（1 字节）：时钟数、最小频率、最大频率的数据总字节数
 - 时钟数（1 字节）：时钟的种类（例：02h 系统时钟和外围时钟）
 - 最小频率（2 字节）：工作频率的最小值（例：07D0h 20.00MHz）
这是将频率（MHz）的小数点后的 2 位的值扩大 100 倍的值。
 - 最大频率（2 字节）：工作频率的最大值
格式和最小频率相同。
 - SUM（1 字节）：校验和

(7) 用户引导 MAT 信息查询

如果主机查询用户引导 MAT 信息，RX610 就发送用户引导 MAT 的区域数和地址的信息。

| | | | |
|----|---------|----|-----|
| 命令 | 24h | | |
| 应答 | 34h | 长度 | 区域数 |
| | 区域的起始地址 | | |
| | 区域的结束地址 | | |
| | 区域的起始地址 | | |
| | 区域的结束地址 | | |
| | ... | | |
| | 区域的起始地址 | | |
| | 区域的结束地址 | | |
| | SUM | | |

- 【符号说明】
- 长度（1 字节）：区域数、区域的起始地址、区域的结束地址的数据总字节数
 - 区域数（1 字节）：用户引导 MAT 的区域数（连续区域为 1 个区域）
 - 区域的起始地址（4 字节）：用户引导 MAT 区域的起始地址
 - 区域的结束地址（4 字节）：用户引导 MAT 区域的结束地址
 - SUM（1 字节）：校验和

(8) 用户 MAT 信息查询

如果主机查询用户 MAT 信息，RX610 就发送用户 MAT 的区域数和地址的信息。

| | | | |
|----|---------|----|-----|
| 命令 | 25h | | |
| 应答 | 35h | 长度 | 区域数 |
| | 区域的起始地址 | | |
| | 区域的结束地址 | | |
| | 区域的起始地址 | | |
| | 区域的结束地址 | | |
| | ... | | |
| | 区域的起始地址 | | |
| | 区域的结束地址 | | |
| | SUM | | |

- 【符号说明】
- 长度（1 字节）：区域数、区域的起始地址、区域的结束地址的数据总字节数
 - 区域数（1 字节）：用户 MAT 的区域数（连续区域为 1 个区域）
 - 区域的起始地址（4 字节）：用户 MAT 区域的起始地址
 - 区域的结束地址（4 字节）：用户 MAT 区域的结束地址
 - SUM（1 字节）：校验和

(9) 擦除块信息查询

如果主机查询擦除块信息，RX610 就发送用户 MAT 的擦除块数和地址的信息。

| | | | |
|----|--------|----|----|
| 命令 | 26h | | |
| 应答 | 36h | 长度 | 块数 |
| | 块的起始地址 | | |
| | 块的结束地址 | | |
| | 块的起始地址 | | |
| | 块的结束地址 | | |
| | ... | | |
| | 块的起始地址 | | |
| | 块的结束地址 | | |
| | SUM | | |

- 【符号说明】
- 长度（1 字节）：块数、块的起始地址、块的结束地址的数据总字节数
 - 块数（1 字节）：用户 MAT 的擦除块数
 - 区域的起始地址（4 字节）：擦除块的起始地址
 - 区域的结束地址（4 字节）：擦除块的结束地址
 - SUM（1 字节）：校验和

(10) 编程长度查询

如果主机查询编程长度，RX610 就发送编程长度的信息。

| | | | |
|----|-----|----|------|
| 命令 | 27h | | |
| 应答 | 37h | 长度 | 编程长度 |
| | SUM | | |

- 【符号说明】
- 长度（1 字节）：编程长度的字符数（是固定值，为“2”。）
 - 编程长度（2 字节）：编程单位（以字节数为单位）
 - SUM（1 字节）：校验和

(11) 新位速率的选择

如果主机发送新位速率选择命令，RX610 就检查能否将内部 SCI 设定为指定的新位速率。如果能设定新为位速率，RX610 就发送应答（06h）并且将 SCI 设定为新位速率。如果不能设定为新位速率或者送来的命令为非法命令，RX610 就发送错误应答（BFh）。如果主机接收应答（06h），就用发送新位速率选择命令时的位速率进行 1 位时间的等待，并且将主机的位速率更改为新位速率。然后，主机通过新位速率发送确认数据（06h），RX610 发送确认数据的应答（06h）。

必须在发送时钟模式选择命令后发送新位速率选择命令。



图 26.30 新位速率的选择顺序

| | | | | | |
|------|-----|-------|-------|--|------|
| 命令 | 3Fh | 长度 | 位速率 | | 输入频率 |
| | 时钟数 | 倍增比 1 | 倍增比 2 | | |
| | SUM | | | | |
| 应答 | 06h | | | | |
| 错误应答 | BFh | 错误 | | | |
| 确认 | 06h | | | | |
| 应答 | 06h | | | | |

- 【符号说明】**
- 长度（1 字节）：位速率、输入频率、时钟数、倍增比的数据总字节数
 - 位速率（2 字节）：新位速率（例：00C0h 19200bps）
将位速率值设定为 1/100 的值。
 - 输入频率（2 字节）：RX610 的输入频率（例：04E2h 12.50MHz）
这是将输入频率的小数点后的 2 位的值扩大 100 倍的值。
 - 时钟数（1 字节）：时钟的种类（例：02h 系统时钟和外围时钟）
 - 倍增比 1（1 字节）：输入频率对应的系统时钟（ICLK）的倍增比 / 分频比
倍增比（例：04h=4 4 倍增）← 用正数指定
分频比（例：FEh=-2 2 分频）← 用负数指定
 - 倍增比 2（1 字节）：输入频率对应的外围时钟（PCLK）的倍增比 / 分频比
和倍增比 1 相同的格式
 - SUM（1 字节）：校验和
 - 错误：错误码
11h: 校验和错误
24h: 位速率不能选择错误
25h: 输入频率错误
26h: 倍增比错误
27h: 工作频率错误

- 位速率不能选择错误

如果因RX610的SCI的误差不足4%而无法通过新位速率选择命令设定所指定的位速率，就产生位速率不能选择错误。假设通过新位速率选择命令指定的位速率为B、输入频率为fEX、倍增比2为MPφ、SCI的位速率寄存器（BRR）的设定值为N以及串行模式寄存器（SMR）的CK5[1:0]位的设定值为n，则用以下表达式求位速率的误差。

$$\text{误差(\%)} = \frac{f_{EX} \times M_{P\phi} \times 10^6}{(N+1) \times B \times 32 \times 2^{2n-1}} - 1$$

- 输入频率错误

如果通过新位速率选择命令指定的输入频率不在与通过时钟模式选择命令指定的时钟模式对应的输入频率的最小值和最大值的范围内，就产生输入频率错误。

- 倍增比错误

如果通过新位速率选择命令指定的倍增比不是与通过时钟模式选择命令指定的时钟模式对应的倍增比，就产生倍增比错误。必须使用倍增比查询命令确认能选择的倍增比。

- 工作频率错误

如果在通过新位速率选择命令指定的工作频率下RX610无法运行，就产生工作频率错误。RX610通过新位速率选择命令指定的输入频率和倍增比进行工作频率的计算，并且检查计算结果是否在各时钟工作频率的最小值和最大值的范围内。必须使用工作频率查询命令确认各时钟工作频率的最小值和最大值。

(12) 编程 / 擦除状态转移

如果主机发送编程 / 擦除状态转移命令，RX610就通过写在ROM中的控制码和ID码判断ID码保护的有效或者无效。当ID码保护有效时，发送应答（16h）并且转移到ID码等待状态；当ID码保护无效时，全部擦除用户MAT、用户引导MAT和数据MAT。在全部擦除后，RX610发送应答（06h）并且转移到编程 / 擦除主机命令等待状态。如果发生错误而没有全部擦除，RX610就发送错误应答（C0h→51h）。

不能在选择器件、时钟模式和新位速率前发行编程 / 擦除状态转移命令。

| | |
|------|------------|
| 命令 | 40h |
| 应答 | ACK |
| 错误应答 | C0h 51h |

【符号说明】 ACK（1字节） : ACK码
 06h: ID码保护无效
 16h: ID码保护有效

(13) 嵌入式程序状态查询

如果主机发送嵌入式程序状态查询命令，RX610 就发送当前的状态。能在查询设定主机命令等待状态和编程 / 擦除主机命令等待状态下使用嵌入式程序状态查询命令。

| | | | | |
|----|-----|----|----|----|
| 命令 | 4Fh | | | |
| 应答 | 5Fh | 长度 | 状态 | 错误 |

- 【符号说明】 长度（1 字节）：状态、错误的数据总字节数（是固定值，为“2”。）
 状态（1 字节）：RX610 的状态（参照表 26.14）
 错误（1 字节）：RX610 的错误发生状况（参照表 26.15）

表 26.14 状态的内容

| 代码 | 内容 |
|-----|---------------------------------|
| 11h | 等待器件的选择。 |
| 12h | 等待时钟模式的选择。 |
| 13h | 等待位速率的选择。 |
| 1Fh | 等待向编程 / 擦除主机命令等待状态的转移（位速率选择结束）。 |
| 31h | 正在擦除用户 MAT 和用户引导 MAT。 |
| 3Fh | 等待编程 / 擦除主机命令。 |
| 4Fh | 等待编程数据的接收。 |
| 5Fh | 等待擦除块的指定。 |

表 26.15 错误的内容

| 代码 | 内容 |
|-----|-----------|
| 00h | 无错误 |
| 11h | 校验和错误 |
| 21h | 器件码不同错误 |
| 22h | 时钟模式不同错误 |
| 24h | 位速率不能选择错误 |
| 25h | 输入频率错误 |
| 26h | 倍增比错误 |
| 27h | 工作频率错误 |
| 29h | 块号错误 |
| 2Ah | 地址错误 |
| 2Bh | 数据长度错误 |
| 51h | 擦除错误 |
| 52h | 未擦除错误 |
| 53h | 编程错误 |
| 54h | 选择处理错误 |
| 80h | 命令错误 |
| FFh | 位速率调整确认错误 |

26.10.6 ID 码等待状态

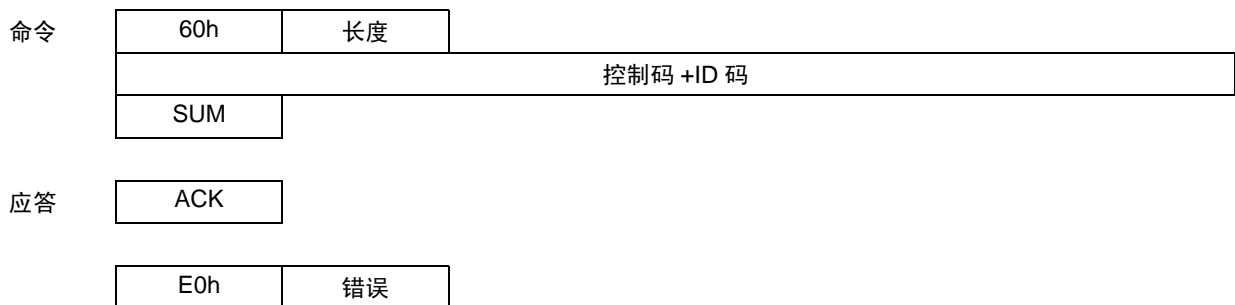
在 ID 码等待状态下能使用的主机命令一览表如表 26.16 所示。

表 26.16 ID 码检查主机命令

| 主机命令名 | 功能 |
|--------|-------------|
| ID 码检查 | 进行 ID 码的检查。 |

如果主机发送未定义命令，RX610 就发送命令错误应答。有关命令错误的内容，请参照“26.10.5 查询设定主机命令的等待状态”。

(1) ID 码检查



- 【符号说明】
- 长度（1 字节）：ID 码的字节数（是固定值，为“16”。）
 - ID 码（16 字节）：控制码（1 字节）+ ID 码（15 字节）
 - SUM（1 字节）：校验和
 - ACK（1 字节）：ACK 码
 - 错误（1 字节）：错误码
 - 26h: 对编程 / 擦除状态转移的应答
 - 11h: 校验和错误
 - 61h: ID 码不同
 - 63h: ID 码不同 [擦除错误]
 - 因 ID 码不同而发生擦除错误。

26.10.7 编程 / 擦除主机命令等待状态

在编程 / 擦除主机命令等待状态下能使用的主机命令一览表如表 26.17 所示。

表 26.17 编程 / 擦除主机命令

| 主机命令名 | 功能 |
|---------------|-----------------|
| 用户引导 MAT 编程选择 | 选择用户引导 MAT 的编程。 |
| 用户 MAT 编程选择 | 选择用户 MAT 的编程。 |
| 256 字节编程 | 256 字节的编程 |
| 擦除选择 | 选择擦除。 |
| 块擦除 | 擦除块数据。 |
| 存储器读 | 读存储器。 |
| 用户引导 MAT 校验和 | 用户引导 MAT 的校验和 |
| 用户 MAT 校验和 | 用户 MAT 的校验和 |
| 用户引导 MAT 空白检查 | 用户引导 MAT 的空白检查 |
| 用户 MAT 空白检查 | 用户 MAT 的空白检查 |
| 读锁定位状态 | 读锁定位。 |
| 锁定位编程 | 对锁定位进行编程。 |
| 锁定位有效 | 设定为锁定位保护有效。 |
| 锁定位无效 | 设定为锁定位保护无效。 |
| 嵌入式程序状态查询 | 查询 RX610 的状态。 |

如果主机发送未定义命令，RX610 就发送命令错误应答。有关命令错误的内容，请参照“26.10.5 查询设定主机命令的等待状态”

在进行 ROM 的编程时，主机在发送编程选择命令（选择用户 MAT 编程 / 用户引导 MAT 编程）后发送 256 字节编程命令。如果主机发送编程选择命令，RX610 就进入编程数据等待状态（参照“26.10.3 引导模式的状态转移”）。如果在编程数据等待状态下主机发送 256 字节编程命令，RX610 就对 ROM 进行数据编程。如果主机在将编程目标地址设定为 FFFF FFFFh 后发送 256 字节编程命令，RX610 就判断为编程结束，转移到编程 / 擦除主机命令等待状态。

在擦除 ROM 时，主机在发送擦除选择命令后发送块擦除命令。如果主机发送擦除选择命令，RX610 就进入擦除块指定等待状态（参照“26.10.3 引导模式的状态转移”）。如果在擦除块指定等待状态下主机发送块擦除命令，RX610 就对 ROM 进行块擦除。如果主机在将块号设定为“FFh”后发送块擦除命令，RX610 就判断为擦除结束，转移到编程 / 擦除主机命令等待状态。

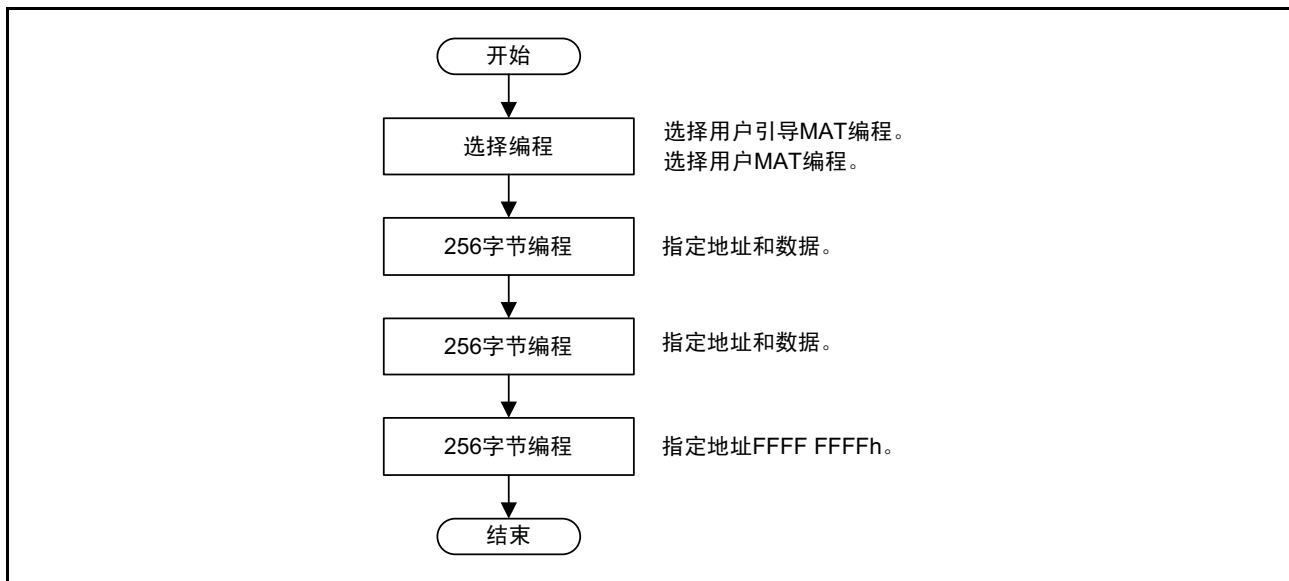


图 26.31 引导模式的 ROM 编程方法

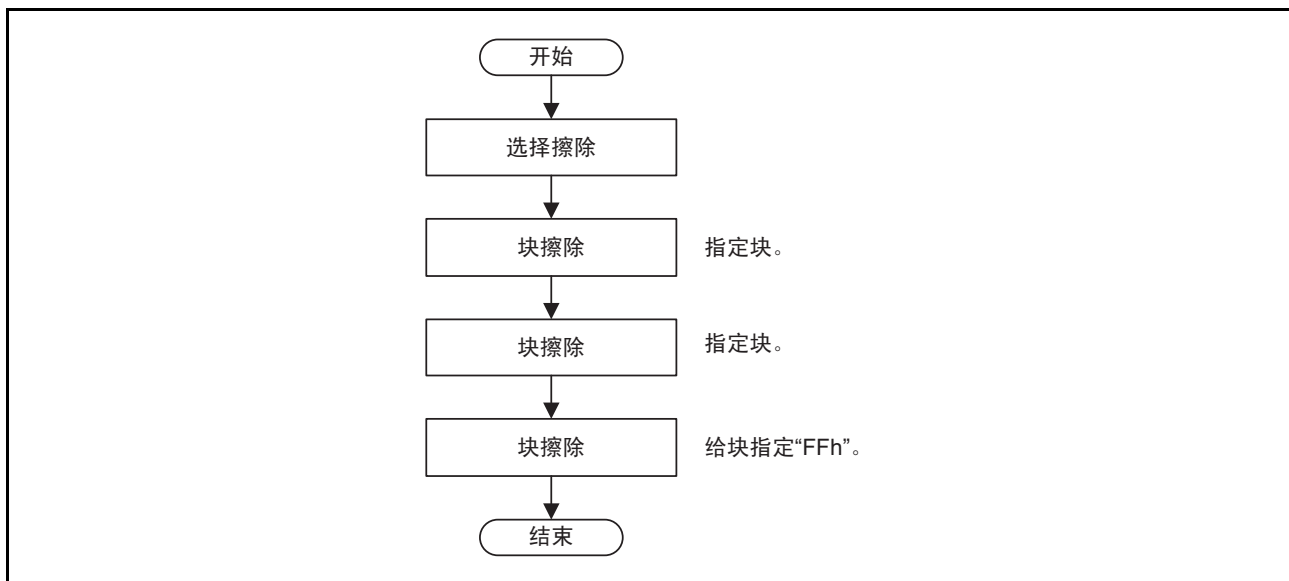


图 26.32 引导模式的 ROM 擦除方法

以下说明各主机命令的详细内容。说明内容中的“命令”是从主机发送到RX610的命令，“应答”是从RX610发送到主机的应答。“校验和”是指在合计发送的各字节时，为了得到“00h”而算出的字节数据。

(1) 用户引导 MAT 编程选择

如果主机发送用户引导 MAT 编程选择命令，RX610 就选择用户引导 MAT 编程程序，进入编程数据等待状态。

命令

| |
|-----|
| 42h |
|-----|

应答

| |
|-----|
| 06h |
|-----|

(2) 用户 MAT 编程选择

如果主机发送用户 MAT 编程选择命令，RX610 就选择用户 MAT 编程程序，进入编程数据等待状态。

命令

| |
|-----|
| 43h |
|-----|

应答

| |
|-----|
| 06h |
|-----|

(3) 256 字节编程

如果主机发送 256 字节编程命令，RX610 就进行 ROM 的编程。如果 ROM 的编程正常结束，RX610 就发送应答（06h）。如果在编程处理过程中发生错误，RX610 就发送错误应答（D0h）。

| | | | | |
|------|-----|------|-----|----|
| 命令 | 50h | 编程地址 | | |
| | 数据 | 数据 | ... | 数据 |
| | SUM | | | |
| 应答 | 06h | | | |
| 错误应答 | D0h | 错误 | | |

| | | |
|--------|------------|--|
| 【符号说明】 | 编程地址（4 字节） | : 编程目标地址 在进行编程时调整为以 256 字节为边界的地址。 在指定编程结束时发送“FFFF FFFFh”。 |
| | 数据（256 字节） | : 编程数据 给不需要编程的字节指定“FFh”。 在指定编程结束时不需要发送数据。 (按照编程地址 →SUM 的顺序进行发送) |
| | SUM（1 字节） | : 校验和 |
| | 错误（1 字节） | : 错误码 11h: 校验和错误 2Ah: 地址错误（地址不在指定的 MAT 内） 53h: 发生编程错误，不能进行编程。 |

(4) 擦除选择

如果主机发送擦除选择命令，RX610 就选择擦除程序，进入擦除块指定等待状态。

| | |
|----|-----|
| 命令 | 48h |
|----|-----|

| | |
|----|-----|
| 应答 | 06h |
|----|-----|

(5) 块擦除

如果主机发送块擦除命令，RX610 就擦除 ROM。如果 ROM 擦除正常结束，RX610 就发送应答（06h）。如果在擦除处理过程中发生错误，RX610 就发送错误应答（D8h）。

| | | | | |
|----|-----|----|---|-----|
| 命令 | 58h | 长度 | 块 | SUM |
|----|-----|----|---|-----|

| | |
|----|-----|
| 应答 | 06h |
|----|-----|

| | | |
|------|-----|----|
| 错误应答 | D8h | 错误 |
|------|-----|----|

【符号说明】

| | |
|-----------|--|
| 长度（1 字节） | : 块的数据字节数（是固定值，为“1”。） |
| 块（1 字节） | : 要擦除的擦除块号 在指定擦除的结束时发送“FFh”。 |
| SUM（1 字节） | : 校验和 |
| 错误（1 字节） | : 错误码 11h: 校验和错误 29h: 块号错误（块号不正确） 51h: 发生擦除错误，不能进行擦除。 |

(6) 存储器读

如果主机发送存储器读命令，RX610 就读 ROM。如果读操作正常结束，RX610 就发送由存储器读命令指定地址的数据。如果没有执行读操作，RX610 就发送错误应答（D2h）。

| | | | | | |
|------|-----|-----|-----|-------|--|
| 命令 | 52h | 长度 | 区域 | 读起始地址 | |
| | 读长度 | | | SUM | |
| 应答 | 52h | 读长度 | | | |
| | 数据 | 数据 | ... | 数据 | |
| | SUM | | | | |
| 错误应答 | D2h | 错误 | | | |

| | | | |
|--------|-------------|---|--|
| 【符号说明】 | 长度（1 字节） | : | 区域、读地址、读长度的数据总字节数 |
| | 区域（1 字节） | : | 读对象的 MAT 00h: 用户引导 MAT 01h: 用户 MAT |
| | 读起始地址（4 字节） | : | 读对象区的起始地址 |
| | 读长度（4 字节） | : | 读取数据的长度（以字节为单位） |
| | SUM（1 字节） | : | 校验和 |
| | 数据（1 字节） | : | 从 ROM 读取的数据 |
| | 错误（1 字节） | : | 错误码 11h: 校验和错误 2Ah: 地址错误 • 在区域的选择中指定“00h”和“01h”以外的值。 • 读起始地址不在指定的 MAT 区内。 2Bh: 长度错误 • 在读长度的选择中指定“00h”。 • 读长度超过 MAT 的长度。 • 由读起始地址和读长度计算出的地址不在 MAT 区内。 |

(7) 用户引导 MAT 校验和

如果主机发送用户引导 MAT 校验和命令，RX610 就发送以字节为单位将用户引导 MAT 的数据相加后的结果（校验和）。

| | | | | |
|----|-----|----|----------|-----|
| 命令 | 4Ah | | | |
| 应答 | 5Ah | 长度 | MAT 的校验和 | SUM |

| | | | |
|--------|----------------|---|--------------------------|
| 【符号说明】 | 长度（1 字节） | : | MAT 的校验和的字节数（是固定值，为“4”。） |
| | MAT 的校验和（4 字节） | : | 用户引导 MAT 的校验和结果 |
| | SUM（1 字节） | : | 校验和（应答数据的校验和） |

(8) 用户 MAT 校验和

如果主机发送用户 MAT 校验和命令，RX610 就发送以字节为单位将用户 MAT 的数据相加后的结果（校验和）。

| | | | | |
|----|-----|----|----------|-----|
| 命令 | 4Bh | | | |
| 应答 | 5Bh | 长度 | MAT 的校验和 | SUM |

- 【符号说明】 长度（1 字节）：MAT 的校验和的字节数（是固定值，为“4”。）
 MAT 的校验和（4 字节）：用户 MAT 的校验和结果
 用户 MAT 也包含用于验证调试功能的键码。
 必须注意：相加的结果包含键码值。
 SUM（1 字节）：校验和（应答数据的校验和）

(9) 用户引导 MAT 空白检查

如果主机发送用户引导 MAT 空白检查命令，RX610 就检查用户引导 MAT 是否全部为擦除状态。如果用户引导 MAT 全部为擦除状态，RX610 就发送应答（06h）。如果用户引导 MAT 有未擦除区域，RX610 就发送错误应答（CCh→52h）。

| | | |
|------|-----|-----|
| 命令 | 4Ch | |
| 应答 | 06h | |
| 错误应答 | CCh | 52h |

(10) 用户 MAT 空白检查

如果主机发送用户 MAT 空白检查命令，RX610 就检查用户 MAT 是否全部为擦除状态。如果用户 MAT 全部为擦除状态，RX610 就发送应答（06h）。如果用户 MAT 有未擦除区域，RX610 就发送错误应答（CDh→52h）。

| | | |
|------|-----|-----|
| 命令 | 4Dh | |
| 应答 | 06h | |
| 错误应答 | CDh | 52h |

(11) 读锁定位状态

如果主机发送读锁定位状态命令，RX610 就读锁定位。如果读操作正常结束，RX610 就发送由读锁定位状态命令指定的地址的数据。如果没有执行读操作，RX610 就发送错误应答（F1h）。

| | | | | | | | |
|------|-----|----|----|------|------|-------|-----|
| 命令 | 71h | 长度 | 区域 | 中位地址 | 高位地址 | 最高位地址 | SUM |
| 应答 | 状态 | | | | | | |
| 错误应答 | F1h | 错误 | | | | | |

- 【符号说明】
- 长度（1 字节）：区域、中位地址、高位地址、最高位地址的数据总字节数（是固定值，为“4”。）
 - 区域（1 字节）：读对象的 MAT
01h: 用户 MAT
 - 中位地址（1 字节）：指定块的结束地址的中位地址（8～15 位）
 - 高位地址（1 字节）：指定块的结束地址的高位地址（16～23 位）
 - 最高有效位地址（1 字节）：指定块的结束地址的最高位地址（24～31 位）
 - SUM（1 字节）：校验和
 - 状态（1 字节）：当 bit6 为“0”时，为锁定状态。
当 bit6 为“1”时，为非锁定状态。
 - 错误（1 字节）：错误码
11h: 校验和错误
2Ah: 地址错误（地址不在指定的 MAT 内）

(12) 锁定位编程

如果主机发送锁定位编程命令，RX610 就对锁定位进行编程，将指定块设定为锁定状态。如果被正常锁定，RX610 就发送应答（06h）。否则，RX610 就发送错误应答（F7h）。

| | | | | | | | |
|------|-----|----|----|------|------|-------|-----|
| 命令 | 77h | 长度 | 区域 | 中位地址 | 高位地址 | 最高位地址 | SUM |
| 应答 | 06h | | | | | | |
| 错误应答 | F7h | 错误 | | | | | |

- 【符号说明】
- 长度（1 字节）：区域、中位地址、高位地址、最高位地址的数据总字节数（是固定值，为“4”。）
 - 区域（1 字节）：锁定对象的 MAT
01h: 用户 MAT
 - 中位地址（1 字节）：指定块的结束地址的中位地址（8～15 位）
 - 高位地址（1 字节）：指定块的结束地址的高位地址（16～23 位）
 - 最高有效位地址（1 字节）：指定块的结束地址的最高位地址（24～31 位）
 - SUM（1 字节）：校验和
 - 错误（1 字节）：错误码
11h: 校验和错误
2Ah: 地址错误（地址不在指定的 MAT 内）
53h: 发生编程错误，不能设定为锁定状态。

(13) 锁定位有效

如果主机发送锁定位有效命令，RX610 就将锁定位置为有效。

命令

| |
|-----|
| 7Ah |
|-----|

应答

| |
|-----|
| 06h |
|-----|

(14) 锁定位无效

如果主机发送锁定位无效命令，RX610 就将锁定位置为无效。

命令

| |
|-----|
| 75h |
|-----|

应答

| |
|-----|
| 06h |
|-----|

(15) 嵌入式程序状态查询

请参照“26.10.5 查询设定主机命令的等待状态”。

26.11 on-chip 调试器的 ID 码保护

这是禁止连接 on-chip 调试器的功能。在连接 on-chip 调试器时，使用写在 ROM 中的控制码和 ID 码，决定 on-chip 调试器 ID 码保护的有效或者无效以及判断 on-chip 调试器的 ID 码保护。当 ID 码保护有效时，就将从 on-chip 调试器送来的代码和 ROM 中的控制码、ID 码进行比较。如果相同，就允许连接 on-chip 调试器；否则就不能连接 on-chip 调试器。但是，如果控制码为“52h”并且 ID 码从 ID 码 1 开始按照顺序设定“50h”、“72h”、“6Fh”、“74h”、“65h”、“63h”、“74h”、“FFh”、……、“FFh”，就不判断 ID 码而总是视为不同，禁止连接 on-chip 调试器。当控制码和 ID 码都为“FFh”时，就不判断 ID 码而视为相同，允许连接 on-chip 调试器。闪存的 ID 码的结构和图 26.25 相同。

表 26.18 on-chip 调试器的 ID 码保护的规格

| 控制码 | ID 码 | 保护状态 | 连接 on-chip 调试器时的运行 |
|------|--|------|--|
| FFh | FFh、……、FFh（全部为“FFh”） | 保护无效 | 不判断控制码和 ID 码，总是视为和 ID 码相同，允许连接 on-chip 调试器。 |
| 52h | 50h、72h、6Fh、74h、65h、63h、74h、FFh、……、FFh | 保护有效 | 不判断控制码和 ID 码，总是视为和 ID 码不同，禁止连接 on-chip 调试器。 |
| 上述以外 | 上述以外 | 保护有效 | ID 码相同：结束 on-chip 调试器的验证，允许连接 on-chip 调试器。 ID 码不同：再次转移到 ID 码等待状态。 |

26.12 ROM 码保护

ROM 码保护是在使用 PROM 编程器时禁止读和改写闪存的功能。闪存的 ROM 码是 32 位长的数据。ROM 码的结构如图 26.33 所示，必须以 32 位为单位设定 ROM 码。

要解除 ROM 码保护时，必须在引导模式或者用户编程模式中擦除包括 ROM 码的用户 MAT 的 EB00 块。

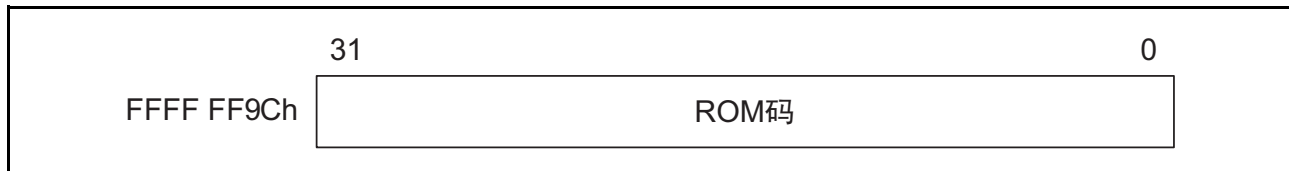


图 26.33 ROM 码的结构

表 26.19 ROM 码保护的规格

| ROM 码 | 保护状态 | 连接 PROM 编程器时的运行 |
|------------|--------------------------|--------------------------|
| 0000 0000h | ROM 码保护有效 (ROM 码保护 1) | 禁止读和改写用户 MAT 和用户引导 MAT。 |
| 0000 0001h | ROM 码保护有效 (ROM 码保护 2) | 禁止读用户 MAT 和用户引导 MAT。 |
| 上述以外 | ROM 码保护无效 | 允许读或者改写用户 MAT 和用户引导 MAT。 |

26.13 使用时的注意事项

(1) 编程 / 擦除挂起的对象区

编程 / 擦除挂起中的区域的保存数据为不定值。必须注意：不能执行编程 / 擦除挂起对象区的指令以及读数据，以免因读到不确定的数据而发生误动作。

(2) 编程 / 擦除挂起引起的中止

如果通过编程 / 擦除挂起命令中止编程 / 擦除处理，就必须通过恢复命令结束处理。

(3) 追加编程的禁止

不能对同一区域进行 2 次或者 2 次以上的编程。要改写已编程的区域时，必须擦除该区域。

(4) 编程 / 擦除过程中的复位

不能在编程和擦除过程中通过 RES# 引脚进行复位，否则有可能对闪存产生永久性的破坏。如果错误地进行了复位输入，就必须在至少等待 100 μ s 的复位输入期间后解除复位。

如果在编程和擦除过程中通过 FRESETR.FRESET 位对 FCU 进行复位或者因 WDT 上溢引起内部复位，就必须保持 tRESW2（参照“29. 电特性”）时间的复位状态。不能在复位期间读 ROM。

(5) 编程 / 擦除过程中的 NMI 中断的禁止

如果在编程 / 擦除过程中发生 NMI 中断，就从 ROM 取向量，读到不确定的数据。因此，不能在对 ROM 进行编程或者擦除过程中发生 NMI 中断。

(6) 编程 / 擦除过程中的中断向量的分配

如果在编程 / 擦除过程中发生中断，就有可能从 ROM 取向量。为了避免从 ROM 取向量，可以通过 CPU 的中断表寄存器（INTB）将取中断向量目标表设定在 ROM 以外的区域。

(7) 编程 / 擦除过程中的禁止事项

在编程 / 擦除过程中给闪存的内部外加高电压。为了防止损坏器件，不能进行以下的操作：

- 转移到软件待机模式和深度软件待机模式。
- 切断RX610的电源。
- 更新FWEPROR.FLWE[1:0]位的值。
- 通过设定SYSCR0.ROME位更改运行模式。

27. 数据闪存（保存数据的闪存）

RX610 群内置用于保存最多 2M 字节代码的闪存（ROM）以及用于保存 32K 字节数据的闪存（数据闪存）。本章说明数据闪存，有关 ROM 请参照“26. ROM（保存代码的闪存）”。

27.1 概要

数据闪存的规格表 27.1 所示，ROM 和数据闪存外围的框图如图 27.1 所示。

表 27.1 数据闪存的规格

| 项目 | | 内容 |
|--------------------|--------|--|
| 存储空间 | | 32K 字节 |
| 经由外围总线的读操作 | | 在进行字存取或者字节存取时，以 3 个 PCLK 周期进行读操作。 |
| 编程 / 擦除方式 | | <ul style="list-style-type: none"> 内置对 ROM/ 数据闪存进行改写的专用定序器（FCU）。 能通过给 FCU 发行命令，进行 ROM/ 数据闪存的编程或者擦除。 |
| BGO（后台操作）功能 | | <ul style="list-style-type: none"> 在对 ROM 进行编程或者擦除的期间，CPU 能执行分配在 ROM 区 / 数据闪存以外区域的程序。 在对数据闪存进行编程或者擦除的期间，CPU 能执行分配在 ROM 区的程序。 |
| 挂起 / 恢复功能 | | <ul style="list-style-type: none"> 中止 ROM 的编程或者擦除，CPU 能执行 ROM 区的程序（挂起）。 在中止后，能重新开始对 ROM 进行编程或者擦除（恢复）。 |
| 编程 / 擦除单位 | | <ul style="list-style-type: none"> 数据 MAT 的编程单位：8 字节或者 128 字节 数据 MAT 的擦除单位：8K 字节（4 块） |
| 空白检查功能 | | <ul style="list-style-type: none"> 能执行用于确认数据闪存擦除状态的空白检查命令。 能进行空白检查的区域为 8 字节或者 8K 字节。 |
| 板上编程 (3 种) | 引导模式 | <ul style="list-style-type: none"> 能使用 SCI 改写数据 MAT。 能自动调整主机和 RX610 之间的 SCI 通信的位速率。 |
| | 用户引导模式 | 能从用户引导 MAT 启动并且改写数据 MAT。 |
| | 用户编程模式 | 能通过程序改写数据 MAT。 |
| 保护功能 | 软件保护功能 | 能通过 FENTRYR.FENTRYD 位、FWEPROR.FLWE[1:0] 位、DFLRE 寄存器和 DFLWE 寄存器防止意料不到的改写。 |
| | 错误保护功能 | 如果在编程或者擦除过程中检测到异常，就禁止以后的编程或者擦除处理。 |
| 编程时间 / 擦除时间 / 改写次数 | | 请参照“29. 电特性”。 |

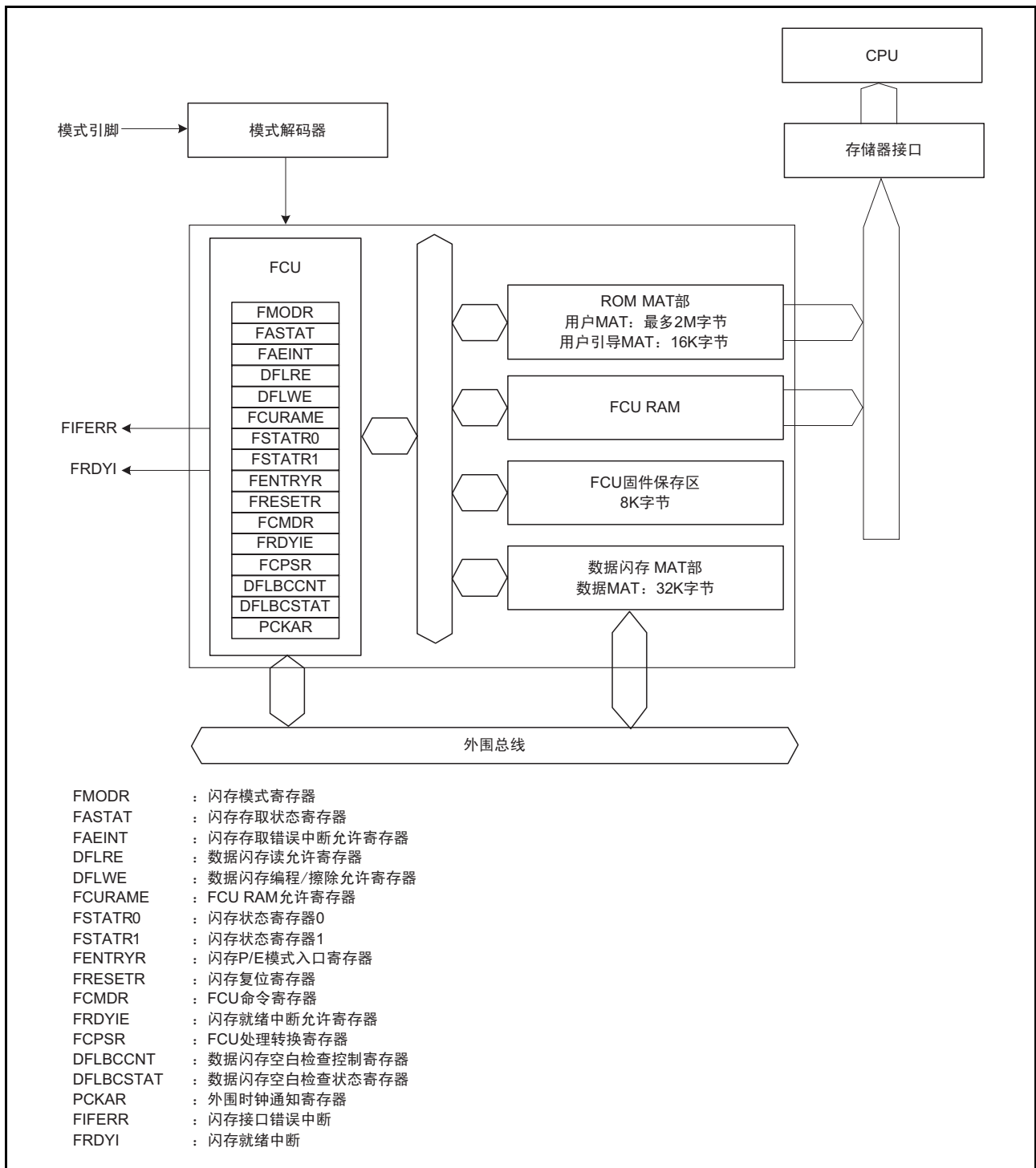


图 27.1 数据闪存的框图

数据闪存的相关输入 / 输出引脚如表 27.2 所示。

表 27.2 数据闪存的相关输入 / 输出引脚

| 引脚名 | 输入 / 输出 | 功能 |
|----------|---------|----------------------------|
| P05/RxD4 | 输入 | 用于引导模式。SCI4 的接收数据（用于主机通信）。 |
| P04/TxD4 | 输出 | 用于引导模式。SCI4 的发送数据（用于主机通信）。 |

27.2 寄存器说明

数据闪存的相关寄存器一览表如表 27.3 所示。部分寄存器也有 ROM 的相关位，本章只说明数据闪存相关位的功能。有关由 ROM/ 数据闪存的通用位构成的寄存器（FRDYIE、FCURAME、FSTATR0、FSTATR1、FRESETR、FCMDR、FCPSR、PCKAR、FWEPROR）以及 ROM 专用位功能的详细内容，请参照“26. ROM（保存代码的闪存）”的“26.2 寄存器说明”。

通过复位对数据闪存的相关寄存器进行初始化。

表 27.3 数据闪存的相关寄存器一览表

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|------------------|-----------|-------|------------|------|
| 闪存模式寄存器 | FMODR | 00h | 007F C402h | 8 |
| 闪存存取状态寄存器 | FASTAT | 00h | 007F C410h | 8 |
| 闪存存取错误中断允许寄存器 | FAEINT | 9Bh | 007F C411h | 8 |
| 闪存就绪中断允许寄存器 | FRDYIE | 00h | 007F C412h | 8 |
| 数据闪存读允许寄存器 | DFLRE | 0000h | 007F C440h | 16 |
| 数据闪存编程 / 擦除允许寄存器 | DFLWE | 0000h | 007F C450h | 16 |
| FCU RAM 允许寄存器 | FCURAME | 0000h | 007F C454h | 16 |
| 闪存状态寄存器 0 | FSTATR0 | 80h | 007F FFB0h | 8 |
| 闪存状态寄存器 1 | FSTATR1 | 00h | 007F FFB1h | 8 |
| 闪存 P/E 模式入口寄存器 | FENTRYR | 0000h | 007F FFB2h | 16 |
| 闪存复位寄存器 | FRESETR | 0000h | 007F FFB6h | 16 |
| FCU 命令寄存器 | FCMDR | FFFFh | 007F FFBAh | 16 |
| FCU 处理转换寄存器 | FCPSR | 0000h | 007F FFC8h | 16 |
| 数据闪存空白检查控制寄存器 | DFLBCCNT | 0000h | 007F FFCAh | 16 |
| 数据闪存空白检查状态寄存器 | DFLBCSTAT | 0000h | 007F FFCEh | 16 |
| 外围时钟通知寄存器 | PCKAR | 0000h | 007F FFE8h | 16 |
| 闪存编程 / 擦除保护寄存器 | FWEPROR | 02h | 0008 C289h | 8 |

27.2.1 闪存模式寄存器（FMODR）

地址 007F C402h

| | | | | | | | |
|----|----|----|-------|----|----|----|----|
| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| — | — | — | FRDMD | — | — | — | — |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

复位后的值

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|-------|------------|--|-----|
| b3-b0 | — | 保留位 | 读写值都为“0”。 | R/W |
| b4 | FRDMD | FCU 读模式选择位 | 0: 存储区读方式 在转移到数据闪存锁定位读模式时进行设定。 数据闪存没有锁定位，如果在转移到锁定位读模式后读数据闪存区，就会读到不确定的数据。 1: 寄存器读方式 在使用空白检查命令时进行设定。 | R/W |
| b7-b5 | — | 保留位 | 读写值都为“0”。 | R/W |

FMODR 寄存器是指定锁定位的读法的寄存器。在使用空白检查命令时，必须将 FRDMD 位置“1”。在内部 ROM 无效模式中，FMODR 寄存器的读数据为“00h”，不能写。通过复位对 FMODR 寄存器进行初始化。

FRDMD 位（FCU 读模式选择位）

此位选择是进行数据闪存锁定位读模式的转移处理还是进行空白检查处理。

在读 ROM 的锁定位时，使用 FRDMD 位选择锁定位的读法（参照“26. ROM（保存代码的闪存）”）。

27.2.2 闪存存取状态寄存器（FASTAT）

地址 007F C410h

| | | | | | | | | |
|-------|-------|----|----|-------|-------|----|--------|--------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | ROMAE | — | — | CMDLK | DFLAE | — | DFLRPE | DFLWPE |
| 复位后的值 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|--------|-------------------|---|--------------|
| b0 | DFLWPE | 数据闪存的编程 / 擦除保护违反位 | 0: 不发行数据闪存的编程 / 擦除命令（违反 DFLWE 寄存器的设定） 1: 发行数据闪存的编程 / 擦除命令（违反 DFLWE 寄存器的设定） | R(W) (注1) |
| b1 | DFLRPE | 数据闪存的读保护违反位 | 0: 不读数据闪存（违反 DFLRE 寄存器的设定） 1: 读数据闪存（违反 DFLRE 寄存器的设定） | R(W) (注1) |
| b2 | — | 保留位 | 读写值都为“0”。 | R/W |
| b3 | DFLAE | 数据闪存的存取违反位 | 0: 不违反数据闪存的存取 1: 违反数据闪存的存取 | R(W) (注1) |
| b4 | CMDLK | FCU 命令锁定位 | 0: FCU 不处于命令锁定状态 1: FCU 处于命令锁定状态 | R |
| b6-b5 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | ROMAE | ROM 存取违反位 | 请参照“26. ROM（保存代码的闪存）”。 | R(W) (注1) |

注1. 为了将标志置“0”，只能在读“1”后写“0”。

FASTAT 寄存器是确认有无 ROM/ 数据闪存的存取违反的寄存器。

在内部 ROM 无效模式中，FASTAT 寄存器的读数据为“00h”，不能写。如果 FASTAT 寄存器的某位变为“1”，FCU 就进入命令锁定状态（参照“27.7.2 错误保护”）。为了解除命令锁定状态，需要在将 FASTAT 寄存器置“10h”后，给 FCU 发行状态寄存器清除命令。

通过复位对 FASTAT 寄存器进行初始化。

DFLWPE 位（数据闪存的编程 / 擦除保护违反位）

此位表示有无违反由 DFLWE 寄存器设定的编程 / 擦除保护。

[为“1”的条件]

- 对通过 DFLWE 寄存器设定为禁止编程或者擦除的数据闪存区发行编程/擦除命令

[为“0”的条件]

- 读“1”后写“0”时

DFLRPE 位（数据闪存的读保护违反位）

此位表示有无违反由 DFLRE 寄存器设定的读保护。

[为“1”的条件]

- 对通过 DFLRE 寄存器设定为禁止读的数据闪存区发行读存取命令

[为“0”的条件]

- 读“1”后写“0”时

DFLAE 位（数据闪存的存取违反位）

此位表示有无数据闪存的存取违反。

如果 DFLAE 位变为“1”，FSTATR0.ILGLERR 位就变为“1”，FCU 进入命令锁定状态。

有关 FSTATR0 寄存器，请参照“26.2.5 闪存状态寄存器 0（FSTATR0）”。

[为“1”的条件]

- 在 FENTRYR.FENTRYD 位为“1”时并且在数据闪存 P/E 正常模式中，对数据闪存区发行读存取命令
- 在 FENTRYD 位为“0”的状态下，对数据闪存区发行写存取命令
- 在 FENTRYR.FENTRY1 位或者 FENTRYR.FENTRY0 位为“1”的状态下，对数据闪存区发行存取命令

[为“0”的条件]

- 读“1”后写“0”时

CMDLK 位（FCU 命令锁定位）

此位表示 FCU 处于命令锁定状态（参照“27.7.2 错误保护”）。

[为“1”的条件]

- FCU 检测到错误并且转移到命令锁定状态后

[为“0”的条件]

- FCU 发行状态寄存器清除命令后

27.2.3 闪存存取错误中断允许寄存器（FAEINT）

地址 007F C411h

| | | | | | | | | |
|-------|-------------|----|----|-------------|-------------|----|--------------|--------------|
| | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
| | ROMA EIE | — | — | CMDL KIE | DFLAE IE | — | DFLR PEIE | DFLW PEIE |
| 复位后的值 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |

| 位 | 符号 | 位名 | 功能 | R/W |
|-------|----------|-----------------------|---|-----|
| b0 | DFLWPEIE | 数据闪存的编程 / 擦除保护违反中断允许位 | 0: 在 FASTAT.DFLWPE 位为“1”时, 不发生 FIFERR 中断请求。 1: 在 FASTAT.DFLWPE 位为“1”时, 发生 FIFERR 中断请求。 | R/W |
| b1 | DFLRPEIE | 数据闪存的读保护违反中断允许位 | 0: 在 FASTAT.DFLRPE 位为“1”时, 不发生 FIFERR 中断请求。 1: 在 FASTAT.DFLRPE 位为“1”时, 发生 FIFERR 中断请求。 | R/W |
| b2 | — | 保留位 | 读写值都为“0”。 | R/W |
| b3 | DFLAEIE | 数据闪存的存取违反中断允许位 | 0: 在 FASTAT.DFLAE 位为“1”时, 不发生 FIFERR 中断请求。 1: 在 FASTAT.DFLAE 位为“1”时, 发生 FIFERR 中断请求。 | R/W |
| b4 | CMDLKIE | FCU 命令锁定中断允许位 | 0: 在 FASTAT.CMDLK 位为“1”时, 不发生 FIFERR 中断请求。 1: 在 FASTAT.CMDLK 位为“1”时, 发生 FIFERR 中断请求。 | R/W |
| b6-b5 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | ROMAEIE | ROM 存取违反中断允许位 | 请参照“26. ROM（保存代码的闪存）”。 | R/W |

FAEINT 寄存器设定允许或者禁止闪存接口错误中断（FIFERR）的输出。

在内部 ROM 无效模式中，FAEINT 寄存器的读数据为“00h”，不能写。

通过复位对 FAEINT 寄存器进行初始化。

DFLWPEIE 位（数据闪存的编程 / 擦除保护违反中断允许位）

在发生数据闪存的编程 / 擦除保护违反并且 FASTAT.DFLWPE 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的发生。

DFLRPEIE 位（数据闪存的读保护违反中断允许位）

在发生数据闪存的读保护违反并且 FASTAT.DFLRPE 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的发生。

DFLAEIE 位（数据闪存的存取违反中断允许位）

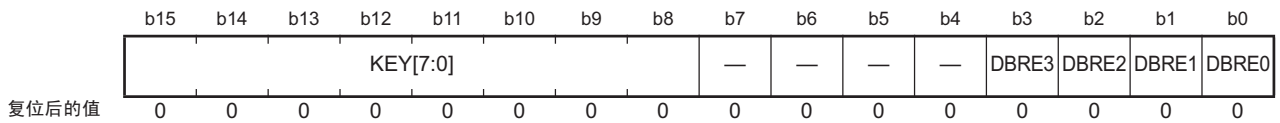
在发生数据闪存的存取违反并且 FASTAT.DFLAE 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的发生。

CMDLKIE 位（FCU 命令锁定中断允许位）

在发生 FCU 命令锁定并且 FASTAT.CMDLK 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的发生。

27.2.4 数据闪存读允许寄存器（DFLRE）

地址 007F C440h



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|----------|------------|---------------------------|---------------|
| b0 | DBRE0 | DB0 块的读允许位 | 0: 禁止读 1: 允许读 | R/W |
| b1 | DBRE1 | DB1 块的读允许位 | | R/W |
| b2 | DBRE2 | DB2 块的读允许位 | | R/W |
| b3 | DBRE3 | DB3 块的读允许位 | | R/W |
| b7-b4 | — | 保留位 | 读写值都为“0”。 | R/W |
| b15-b8 | KEY[7:0] | 键码 | 控制能否改写 DBREj 位 (j=0 ~ 3)。 | R(W) (注 1) |

注 1. 不保持写数据。

DFLRE 寄存器允许或者禁止读数据 MAT 的 DB0 ~ DB3 块（参照图 27.3）。

只在通过字存取给高位字节写特定值时才能写此寄存器，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，DFLRE 寄存器的读数据为“0000h”，不能写。

通过复位对 DFLRE 寄存器进行初始化。

DBREj 位（DBj 块的读允许位）（j=0 ~ 3）

此位设定允许或者禁止读数据 MAT 的 DB0 ~ DB3 块。

DBREj 位用于控制 DBj 块的读。

只在 KEY[7:0] 位为“2Dh”时才能通过字存取写 DBREj 位。

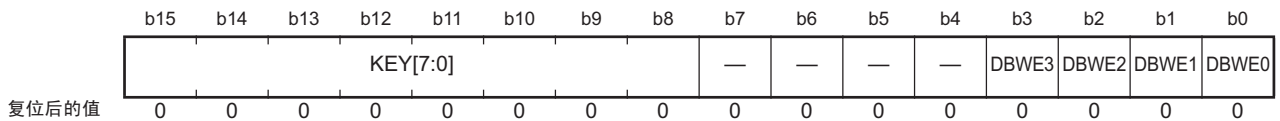
KEY[7:0] 位（键码）

这些位控制能否改写 DBREj 位。

不保持 KEY[7:0] 位的写数据。

27.2.5 数据闪存编程 / 擦除允许寄存器（DFLWE）

地址 007F C450h



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|----------|------------------|---------------------------|--------------|
| b0 | DBWE0 | DB0 块的编程 / 擦除允许位 | 0: 禁止编程和擦除 1: 允许编程和擦除 | R/W |
| b1 | DBWE1 | DB1 块的编程 / 擦除允许位 | | R/W |
| b2 | DBWE2 | DB2 块的编程 / 擦除允许位 | | R/W |
| b3 | DBWE3 | DB3 块的编程 / 擦除允许位 | | R/W |
| b7-b4 | — | 保留位 | 读写值都为“0”。 | R/W |
| b15-b8 | KEY[7:0] | 键码 | 控制能否改写 DBWEj 位 (j=0 ~ 3)。 | R(W) (注1) |

注 1. 不保持写数据。

DFLWE 寄存器允许或者禁止对数据 MAT 的 DB0 ~ DB3 块（参照图 27.3）进行编程或者擦除。

只在通过字存取给高位字节写特定值时才能写此寄存器，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，DFLWE 寄存器的读数据为“0000h”，不能写。

通过复位对 DFLWE 寄存器进行初始化。

DBWEj 位（DBj 块的编程 / 擦除允许位）（j=0 ~ 3）

此位设定允许或者禁止对数据 MAT 的 DB0 ~ DB3 块进行编程或者擦除。

DBWEj 位用于控制 DBj 块的编程和擦除。

只在 KEY[7:0] 位为“1Eh”时才能通过字存取写 DBWEj 位。

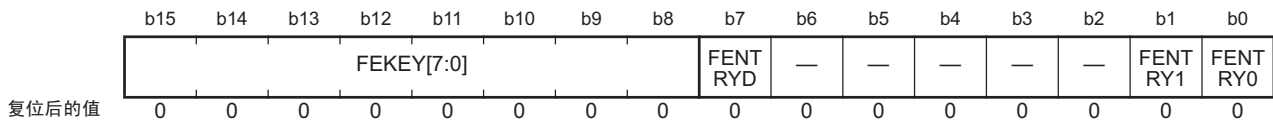
KEY[7:0] 位（键码）

这些位控制能否改写 DBWEj 位。

不保持 KEY[7:0] 位的写数据。

27.2.6 闪存 P/E 模式入口寄存器（FENTRYR）

地址 007F FFB2h



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|------------|-----------------|--|---------------|
| b0 | FENTRY0 | ROM P/E 模式入口位 0 | 请参照“26. ROM（保存代码的闪存）”。 | R/W |
| b1 | FENTRY1 | ROM P/E 模式入口位 1 | 请参照“26. ROM（保存代码的闪存）”。 | R/W |
| b6-b2 | — | 保留位 | 读写值都为“0”。 | R/W |
| b7 | FENTRYD | 数据闪存的 P/E 模式入口位 | 0: 数据闪存为读模式 1: 数据闪存为 P/E 模式 | R/W |
| b15-b8 | FEKEY[7:0] | 键码 | 控制能否改写 FENTRYD 位、FENTRY1 位和 FENTRY0 位。 | R(W) (注 1) |

注 1. 不保持写数据。

FENTRYR 寄存器是将 ROM/ 数据闪存设定为 P/E 模式的寄存器。

为了将 ROM/ 数据闪存设定为 P/E 模式，使 FCU 能接收命令，需要将 FENTRYD 位、FENTRY1 位或者 FENTRY0 位置“1”。但是，如果将多个位置“1”，FSTAT0.ILGLERR 位就变为“1”，FCU 进入命令锁定状态。

只在通过字存取给高位字节写特定值时才能写此寄存器，而在其他情况下写此寄存器时，此寄存器就被初始化。不保持高位字节的写数据。

在内部 ROM 无效模式中，FENTRYR 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESETR 位置“1”，对 FENTRYR 寄存器进行初始化。

有关 FSTAT0 寄存器，请参照“26.2.5 闪存状态寄存器 0（FSTAT0）”。

有关 FRESETR 寄存器，请参照“26.2.10 闪存复位寄存器（FRESETR）”。

FENTRYD 位（数据闪存 P/E 模式入口位）

此位将数据闪存设定为 P/E 模式。

[写操作的有效条件（满足以下全部条件时）]

- 内部 ROM 为有效模式
- FSTAT0.FRDI 位为“1”
- 通过字存取给 FEKEY[7:0] 位写“AAh”

[为“1”的条件]

- 在满足写操作的有效条件并且 FENTRYR 寄存器为“0000h”的状态下，给 FENTRYD 位写“1”时

[为“0”的条件]

- 通过字节存取进行写操作时
- 在 FEKEY[7:0] 位不为“AAh”的状态下，通过字存取进行写操作时
- 在满足写操作的有效条件的状态下，给 FENTRYD 位写“0”时
- 在满足写操作的有效条件并且 FENTRYR 寄存器不为“0000h”的状态下，写 FENTRYR 寄存器时

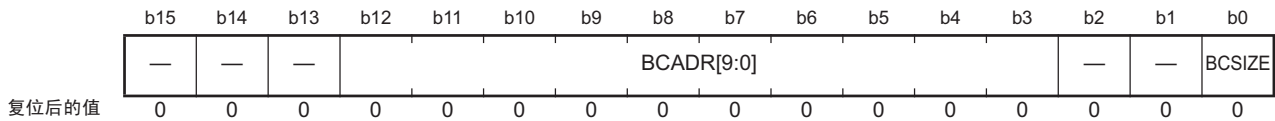
FEKEY[7:0] 位（键码）

这些位控制能否改写 FENTRYD 位、FENTRY1 位和 FENTRY0 位。

不保持 FEKEY[7:0] 位的写数据。

27.2.7 数据闪存空白检查控制寄存器（DFLBCCNT）

地址 007F FFCAh



| 位 | 符号 | 位名 | 功能 | R/W |
|---------|------------|-----------|---------------------------------------|-----|
| b0 | BCSIZE | 空白检查大小设定位 | 0: 空白检查对象区为 8 字节 1: 空白检查对象区为 8K 字节 | R/W |
| b2-b1 | — | 保留位 | 读写值都为“0”。 | R/W |
| b12-b3 | BCADR[9:0] | 空白检查地址设定位 | 设定检查对象区的地址。 | R/W |
| b15-b13 | — | 保留位 | 读写值都为“0”。 | R/W |

DFLBCCNT 寄存器指定空白检查命令的检查对象区的地址和大小。

在内部 ROM 无效模式中，DFLBCCNT 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 DFLBCCNT 寄存器进行初始化。

有关 FRESETR 寄存器，请参照“26.2.10 闪存复位寄存器（FRESETR）”。

BCSIZE 位（空白检查大小设定位）

此位设定空白检查命令的检查对象区的大小。

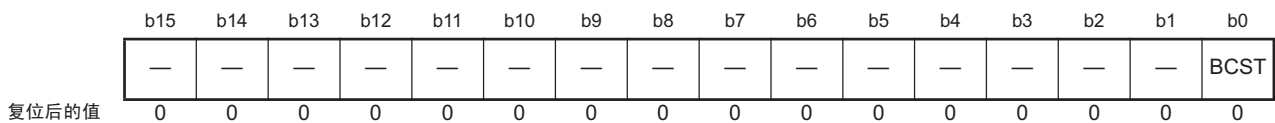
BCADR[9:0] 位（空白检查地址设定位）

在空白检查命令的检查对象区大小为 8 字节（BCSIZE 位为“0”）时，此位设定检查对象区的地址。

当 BCSIZE 位为“0”时，DFLBCCNT 寄存器的设定值（将 BCADR 位的设定值向 MSB 侧移 3 位后的值）以及发行空白检查命令时指定的擦除块起始地址相加后的值为检查对象区的起始地址。

27.2.8 数据闪存空白检查状态寄存器（DFLBCSTAT）

地址 007F FFCEh



| 位 | 符号 | 位名 | 功能 | R/W |
|--------|------|---------|--|-----|
| b0 | BCST | 空白检查状态位 | 0: 空白检查对象区为擦除状态（空白） 1: 空白检查对象区为写“0”或者“1”的状态 | R |
| b15-b1 | — | 保留位 | 读写值都为“0”。 | R/W |

DFLBCSTAT 寄存器保存空白检查命令的处理结果。

在内部 ROM 无效模式中，DFLBCSTAT 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 DFLBCSTAT 寄存器进行初始化。

有关 FRESETR 寄存器，请参照“26.2.10 闪存复位寄存器（FRESETR）”。

BCST 位（空白检查状态位）

此位表示空白检查的结果。

27.3 数据闪存的存储器 MAT 结构

RX610 群的数据闪存由 32K 字节的数据 MAT 构成，数据 MAT 的地址如图 27.2 所示。在读、编程和擦除时，数据 MAT 的地址相同。

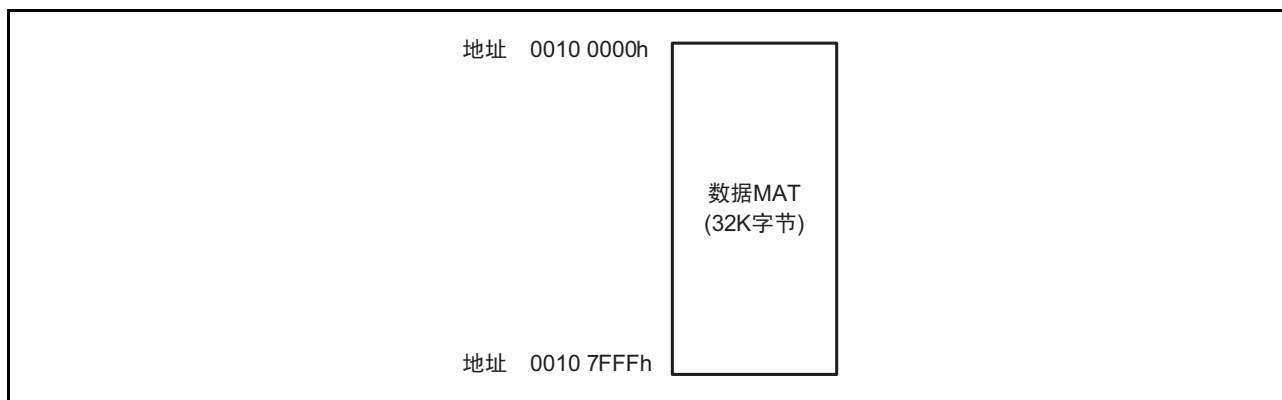


图 27.2 数据闪存的数据 MAT 结构

27.4 块结构

数据 MAT 的擦除块结构如图 27.3 所示。数据 MAT 被分割为 8K 字节（4 块），以块为单位进行擦除，以 8 字节或者 128 字节为单位进行编程。以低位地址的 3 位全部为“0”的 8 字节为单位进行 8 字节编程；以低位地址为“00h”或者“80h”开始的 128 字节为单位进行 128 字节编程。

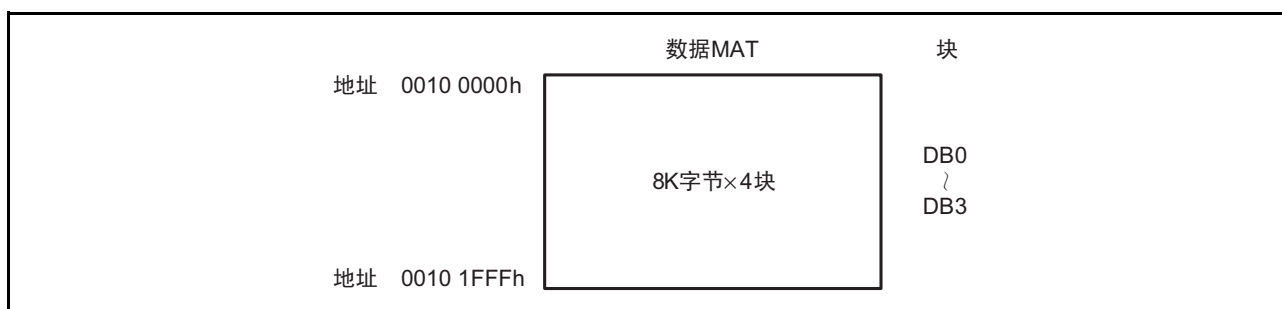


图 27.3 数据 MAT 的块分割

27.5 数据闪存相关的运行模式

有关运行模式的转移，请参照“26.5 ROM 相关的运行模式”。

在引导模式、用户引导模式、单芯片模式（内部 ROM 有效）和内部 ROM 有效扩展模式中，能进行板上数据 MAT 的读、编程或者擦除。

各模式的不同点如表 27.4 所示。

表 27.4 各模式的不同点

| 项目 | 引导模式 | 用户引导模式 | 单芯片模式（内部 ROM 有效）/ 内部 ROM 有效扩展模式 |
|-------------|------------------|----------|---------------------------------|
| 编程 / 擦除环境 | 板上编程 | | |
| 能编程和擦除的 MAT | 数据 MAT | 数据 MAT | 数据 MAT |
| 块分割擦除 | ○（注 1） | ○ | ○ |
| 复位时的启动 MAT | 嵌入式程序存储 MAT（注 2） | 用户引导 MAT | 用户 MAT |

注 1. 有可能在启动时被全部擦除。此后能擦除特定块，详细内容请参照“26.11 on-chip 调试器的 ID 码保护”。

注 2. 用户不能使用。

- 在引导模式中，主机能经由 SCI 对数据 MAT 进行读和编程。
- 在用户引导模式中，能从用户引导 MAT 启动，并且通过任意的接口对用户 MAT/数据 MAT 进行读和编程。
- 在引导模式中，通过引导模式的嵌入式程序使用内部 RAM。因此，不保持内部 RAM 的数据。

27.6 数据闪存的编程 / 擦除

通过给编程 / 擦除的专用定序器（FCU）发行命令（FCU 命令），对数据闪存进行编程和擦除。FCU 有 5 种模式，通过在模式转移后发行编程和擦除命令，进行编程和擦除。

以下说明数据闪存的编程和擦除所需的模式转移和命令体系。这些内容在引导模式、用户引导模式、单芯片模式（内部 ROM 有效模式）和内部 ROM 有效扩展模式中通用。

27.6.1 FCU 的模式

FCU 有 5 种模式，通过写 FENTRYR 寄存器以及 FCU 命令进行模式的转移，FCU 的模式转移图如图 27.4 所示。

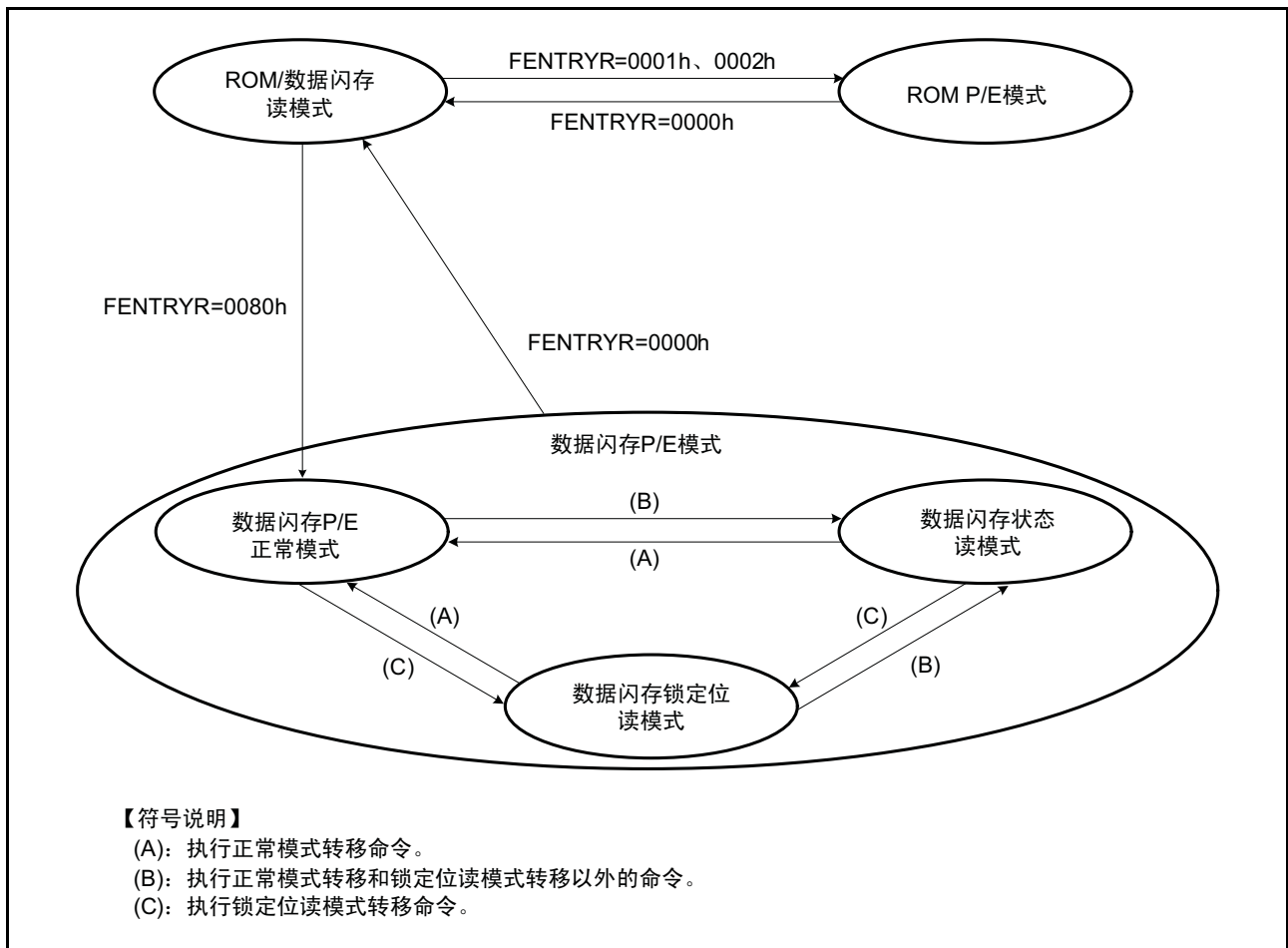


图 27.4 FCU 的模式转移图（数据闪存相关）

27.6.1.1 ROM P/E 模式

ROM P/E 模式是对 ROM 进行编程和擦除的模式。

有关 ROM P/E 模式的详细内容，请参照“26.6.1.2 ROM P/E 模式”。

27.6.1.2 ROM/ 数据闪存读模式

ROM/ 数据闪存读模式是能读 ROM 和数据闪存的模式，不接受 FCU 命令。如果将 FENTRYR.FENTRYD 位置“0”并且将 FENTRYR.FENTRY1 位和 FENTRYR.FENTRY0 位置“00b”，就转移到此模式。

27.6.1.3 数据闪存 P/E 模式

数据闪存 P/E 模式是对数据闪存进行编程和擦除的模式，不能读数据闪存。

数据闪存 P/E 模式有数据闪存 P/E 正常模式、数据闪存状态读模式、数据闪存锁定位读模式共 3 种。

(1) 数据闪存 P/E 正常模式

数据闪存 P/E 正常模式是在对数据闪存进行编程或者擦除时最先转移的模式。如果在 ROM/ 数据闪存读模式中将 FENTRYR.FENTRYD 位置 “1” 并且将 FENTRYR.FENTRY1 位和 FENTRYR.FENTRY0 位置 “00b”，或者在数据闪存 P/E 模式中接受正常模式转移命令，就转移到此模式。能接受的命令如表 27.7 所示。

如果读数据闪存区，就产生数据闪存的存取违反，FCU 进入命令锁定状态。能高速读 ROM。

(2) 数据闪存状态读模式

数据闪存状态读模式是能读数据闪存状态的模式。

如果在数据闪存 P/E 模式中接受正常模式转移和锁定位读模式转移以外的命令，就转移到此模式。FSTATR0.FRDY 位为 “0” 的状态以及发生错误后的命令锁定状态也是数据闪存状态读模式中的状态。能接受的命令如表 27.7 所示。

如果读数据闪存区，就读 FSTATR0 寄存器的值。能高速读 ROM。

(3) 数据闪存锁定位读模式

数据闪存锁定位读模式是能读数据闪存锁定位的模式。但是，因为数据闪存没有锁定位，所以不能读锁定位。

如果在数据闪存 P/E 模式中接受锁定位读模式转移命令，就转移到数据闪存锁定位读模式。能接受的命令如表 27.7 所示。

数据闪存没有锁定位，如果读数据闪存区，读数据就为不定值，但是不发生数据闪存的存取违反。能高速读 ROM。

27.6.2 FCU 命令一览表

FCU 命令有 FCU 模式转移命令和编程 / 擦除命令。对数据闪存进行编程或者擦除时能使用的 FCU 命令一览表如表 27.5 所示。

表 27.5 FCU 命令一览表（数据闪存相关）

| 命令 | 功能 |
|------------------|---|
| 正常模式转移 | 转移到正常模式（参照“27.6.3 FCU 的模式和命令的关系”）。 |
| 状态读模式转移 | 转移到状态读模式（参照“27.6.3 FCU 的模式和命令的关系”）。 |
| 锁定位读模式转移（锁定位读 1） | 转移到锁定位读模式（参照“27.6.3 FCU 的模式和命令的关系”）。 |
| 外围时钟设定 | 设定外围时钟的频率。 |
| 编程 | 对数据闪存进行编程（以 8 字节或者 128 字节为单位）。 |
| 块擦除 | 擦除数据闪存（以块为单位）。 |
| P/E 挂起 | 中止编程和擦除。 |
| P/E 恢复 | 重新开始编程和擦除。 |
| 状态寄存器清除 | 清除 FSTATR0.IGLERR 位、FSTATR0.ERSERR 位和 FSTATR0.PRGERR 位以及解除命令锁定状态。 |
| 锁定位读 2/ 空白检查 | 确认指定的区域是否为擦除状态（空白）。 |

ROM 也使用空白检查命令以外的 FCU 命令。锁定位读 2 命令兼用数据闪存的空白检查命令。如果对 ROM 发行空白检查命令，就读 ROM 的锁定位。

通过写数据闪存区给 FCU 发行命令。

编程命令和空白检查命令的格式如表 27.6 所示。有关编程命令和空白检查命令以外的 FCU 命令格式，请参照“26. ROM（保存代码的闪存）”的“26.6.2 FCU 命令一览表”。

如果在特定条件下进行如表 27.6 所示的写存取，FCU 就进行各命令对应的处理。有关 FCU 的特定条件和命令的使用方法，请分别参照“27.6.3 FCU 的模式和命令的关系”和“27.6.4 FCU 命令的使用方法”。

表 27.6 FCU 命令的格式（数据闪存专用命令）

| 命令 | 总线周期数 | 第 1 周期 | | 第 2 周期 | | 第 3 周期 | | 第 4 ~ 第 N+2 周期 | | 第 N+3 周期 | |
|------------------------|-------|--------|-----|--------|-----|--------|-----|----------------|-----|----------|-----|
| | | 地址 | 数据 | 地址 | 数据 | 地址 | 数据 | 地址 | 数据 | 地址 | 数据 |
| 编程 (8 字节编程: N=4) | 7 | EA | E8h | EA | 04h | WA | WDn | EA | WDn | EA | D0h |
| 编程 (128 字节编程: N=64) | 67 | EA | E8h | EA | 40h | WA | WDn | EA | WDn | EA | D0h |
| 空白检查 | 2 | EA | 71h | BA | D0h | — | — | — | — | — | — |

地址列 EA: 数据闪存区的地址
0010 0000h ~ 0010 7FFFh 的任意地址
WA: 8 字节或者 128 字节的起始地址
BA: 数据闪存的擦除块地址
擦除对象块内的任意地址
数据列 WDn: 编程数据的第 n 个字 (n=1 ~ N)

27.6.3 FCU 的模式和命令的关系

FCU 的各模式能接受的 FCU 命令是固定的，而且能接受的命令也因这些模式的 FCU 状态而不同。

需要在进行 FCU 的模式转移后确认 FCU 的状态，然后发行 FCU 命令。

FCU 的模式 / 状态和能接受的命令如表 27.7 所示。如果发行不能接受的命令，FCU 就进入命令锁定状态（参照“27.7.2 错误保护”）。必须在转移到能接受的模式后，确认 FSTATR0.FRDY 位、FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位和 FSTATR1.FCUERR 位的值，然后发行 FCU 命令。能根据 FSTAT.CMDLK 位的值确认有无发生错误，FSTAT.CMDLK 位的值是 FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位和 FSTATR1.FCUERR 位的值的逻辑或。

表 27.7 FCU 的模式 / 状态和能接受的命令的关系（数据闪存 P/E 模式）

| | P/E 正常模式 | | | 状态读模式 | | | | | | | 锁定位读模式 | | |
|------------------|----------|-------|------|------------|--------------|---------|-------|-------|--------|------|--------|-------|------|
| | 编程挂起中 | 擦除挂起中 | 其他状态 | 编程 / 擦除处理中 | 编程 / 擦除中止处理中 | 空白检查处理中 | 编程挂起中 | 擦除挂起中 | 命令锁定状态 | 其他状态 | 编程挂起中 | 擦除挂起中 | 其他状态 |
| FSTATR0.FRDY 位 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0/1 | 1 | 1 | 1 | 1 |
| FSTATR0.SUSRDY 位 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| FSTATR0.ERSSPD 位 | 0 | 1 | 0 | 0 | 0/1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| FSTATR0.PRGSPD 位 | 1 | 0 | 0 | 0 | 0/1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| FASTAT.CMDLK 位 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 正常模式转移 | ○ | ○ | ○ | × | × | × | ○ | ○ | × | ○ | ○ | ○ | ○ |
| 状态读模式转移 | ○ | ○ | ○ | × | × | × | ○ | ○ | × | ○ | ○ | ○ | ○ |
| 锁定位读模式转移（锁定位读 1） | ○ | ○ | ○ | × | × | × | ○ | ○ | × | ○ | ○ | ○ | ○ |
| 外围时钟通知 | × | × | ○ | × | × | × | × | × | × | ○ | × | × | ○ |
| 编程 | × | △ | ○ | × | × | × | × | △ | × | ○ | × | △ | ○ |
| 块擦除 | × | × | ○ | × | × | × | × | × | × | ○ | × | × | ○ |
| P/E 挂起 | × | × | × | ○ | × | × | × | × | × | × | × | × | × |
| P/E 恢复 | ○ | ○ | × | × | × | × | ○ | ○ | × | × | ○ | ○ | × |
| 状态寄存器清除 | ○ | ○ | ○ | × | × | × | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 空白检查 | ○ | ○ | ○ | × | × | × | ○ | ○ | × | ○ | ○ | ○ | ○ |

○：能接受。

△：只能接受对非擦除中止块进行的编程。

×：不能接受。

27.6.4 FCU 命令的使用方法

FCU 命令的使用方法有：使用编程命令和块擦除命令对数据闪存进行编程和擦除的方法以及使用空白检查命令确认数据闪存的擦除状态的方法。有关 FCU RAM 的固件传送方法以及其他 FCU 命令的使用方法，请参照“26. ROM（保存代码的闪存）”的“26.6.4 FCU 命令的使用方法”。

(1) 外围时钟通知命令的使用方法

通知外围时钟的频率，详细内容请参照“26. ROM（保存代码的闪存）”的“26.6.4 FCU 命令的使用方法”。必须在将 FENTRYR.FENTRYD 位置“1”后设定数据闪存区内的地址。

(2) 编程方法

使用编程命令对数据闪存进行数据编程。

在编程命令的第 1 周期，通过字节存取将“E8h”写到数据闪存区的地址，在第 2 周期，通过字节存取将编程字数（N）^{（注 1）} 写到数据闪存区的地址。在命令的第 3～第 N+2 周期，必须通过字存取写编程数据。在第 3 周期的存取中，必须将编程数据写到编程对象区的起始地址。在进行 8 字节编程时，必须将起始地址调整为以 8 字节为边界的地址，在进行 128 字节编程时，必须将起始地址调整为以 128 字节为边界的地址。如果在通过 N 次字存取将编程数据写到数据闪存区的地址后，在第 N+3 周期通过字节存取将“D0h”写到数据闪存区的地址，FCU 就开始对数据闪存进行编程处理。能通过 FSTATR0.FRDI 位确认编程的结束。

如果在第 3 周期～第 N+2 周期中存取的区域包括不需要编程的地址，就必须将该地址的编程数据置“FFFFh”。如果要在将通过 DFLWE 寄存器控制的编程/擦除保护功能设定为无效后进行编程，就必须将编程对象块的编程/擦除允许位置“1”。

数据闪存的编程方法如图 27.5 所示。

注 1. 当进行 8 字节编程时，N 为“04h”；当进行 128 字节编程时，N 为“40h”。

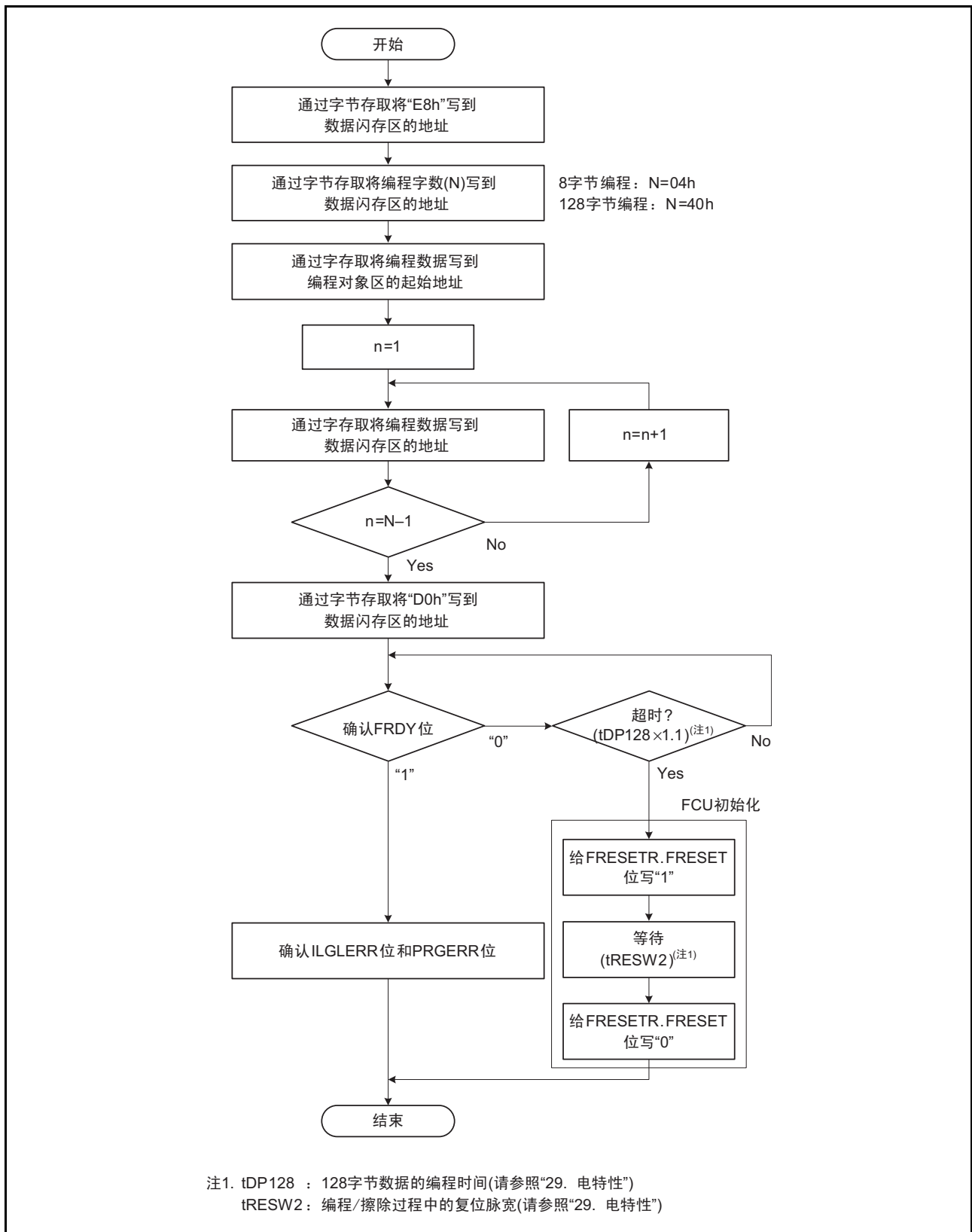


图 27.5 数据闪存的编程

(3) 擦除方法

使用块擦除命令来擦除数据闪存。使用块擦除命令的擦除方法和 ROM 的擦除方法相同（参照“26. ROM（保存代码的闪存）”）。

必须注意：数据闪存具有 DFLWE 寄存器控制的编程 / 擦除保护功能。如果要在将通过 DFLWE 寄存器控制的保护功能设定为无效后进行擦除，就必须将擦除对象块的编程 / 擦除允许位置“1”。

(4) 空白检查

即使通过 CPU 读擦除状态的数据闪存，读取值也为不定值，因此需要使用空白检查命令确认擦除状态。在使用空白检查命令时，必须预先将 FMODR.FRDM 位置“1”，使空白检查命令处于可使用的状态，然后给 DFLBCCNT 寄存器设定检查对象区的大小和地址。当 DFLBCCNT.BCSIZE 位为“1”时，能在空白检查命令的第 2 周期对指定的整个擦除块（8K 字节）进行空白检查；当 BCSIZE 位为“0”时，能在空白检查命令的第 2 周期对指定的擦除块的起始地址和 DFLBCCNT 寄存器值相加后的地址开始的 8 字节区域进行空白检查。在空白检查命令的第 1 周期，通过字节存取将“71h”写到数据闪存区的地址。在命令的第 2 周期，如果通过字节存取将“D0h”写到包括空白检查对象区的擦除块内的任意地址，FCU 就开始进行数据闪存空白检查处理。能通过 FSTAT0.FRDY 位确认空白检查的结束。如果在结束空白检查后确认 DFLBCSTAT.BCST 位的值，就能确认检查对象区是为擦除状态还是为写“0”或者“1”的状态。

数据闪存空白检查方法如图 27.6 所示。

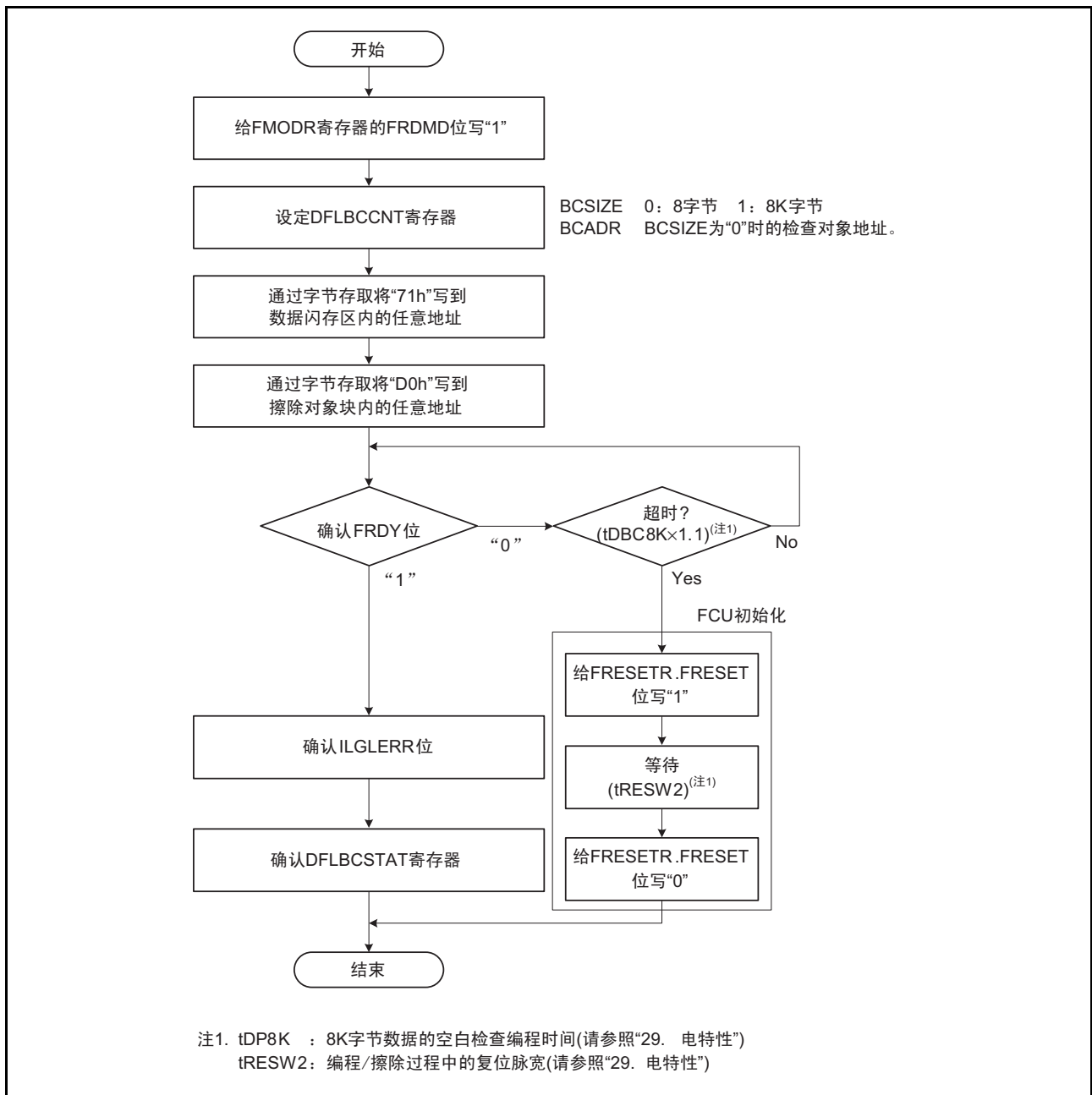


图 27.6 数据闪存的空白检查

27.7 保护

数据闪存的编程 / 擦除的保护有软件保护和错误保护。

27.7.1 软件保护

软件保护是通过设定控制寄存器禁止对数据闪存进行编程、擦除和读的状态。如果违反软件保护而对数据闪存发行编程 / 擦除命令或者读数据闪存，FCU 就检测到错误，进入命令锁定状态。

(1) 通过 FWEPROR 寄存器进行的保护

如果不将 FWEPROR.FLWE[1:0] 位置 “01b”，就不能在任何模式中进行改写。

(2) 通过 FENTRYR 寄存器进行的保护

当 FENTRYR.FENTRYD 位为 “0” 时，FCU 进入 ROM/ 数据闪存读模式。因为在 ROM/ 数据闪存读模式中不接受 FCU 命令，所以数据闪存为禁止编程和擦除的状态。如果在 ROM/ 数据闪存读模式中对数据闪存发行 FCU 命令，FCU 就检测到非法命令错误，进入命令锁定状态（参照 “27.7.2 错误保护”）。

(3) 通过 DFLWE 寄存器进行的保护

当 DFLWE.DBWE_j (j=0 ~ 3) 位为 “0” 时，数据 MAT 的 DB_j 块为禁止编程和擦除的状态。如果在 DBWE_j 位为 “0” 的状态下对 DB_j 块进行编程或者擦除，FCU 就检测到写保护违反，进入命令锁定状态（参照 “27.7.2 错误保护”）。

(4) 通过 DFLRE 寄存器进行的保护

当 DFLRE.DBRE_j (j=0 ~ 3) 位为 “0” 时，数据 MAT 的 DB_j 块为禁止读的状态。如果在 DBRE_i 位为 “0” 的状态下读 DB_j 块，FCU 就检测到读保护违反，进入命令锁定状态（参照 “27.7.2 错误保护”）。

27.7.2 错误保护

错误保护是因误发行 FCU 命令以及发生禁止的存取而导致 FCU 在检测到误动作后禁止接受 FCU 命令的状态（命令锁定状态）。如果 FCU 进入命令锁定状态（FASTAT.CMDLK 位为 “1”），就将状态位（FSTAT0.ILGLERR 位、FSTAT0.ERSERR 位、FSTAT0.PRGERR 位、FSTAT1.FCUERR 位、FASTST.DFLAE 位、FASTST.DFLRPE 位和 FASTST.DFLWPE 位）中的某位或者多个位置 “1”，禁止对数据闪存进行编程和擦除。要解除命令锁定状态时，需要在 FASTAT 寄存器为 “10h” 的状态下发行状态寄存器清除命令。

当 FAEINT.CMDLKIE 位为 “1” 时，如果 FCU 进入命令锁定状态（FASTAT.CMDLK 位为 “1”），就产生闪存接口错误（FIFERR）中断。当 FAEINT 寄存器的数据闪存相关的中断允许位（DFLAEIE 位、DFLRPEIE 位和 DFLWPEIE 位）为 “1” 时，即使 FASTAT 寄存器的对应位（DFLAE 位、DFLRPE 位和 DFLWPE 位）变为 “1”，也产生 FIFERR 中断。

数据闪存相关的错误保护内容和错误检测后的状态位的值（FSTAT0.ILGLERR 位、FSTAT0.ERSERR 位、FSTAT0.PRGERR 位、FASTAT.DFLAE 位、FASTAT.DFLRPE 位和 FASTAT.DFLWPE 位）的关系如表 27.8 所示。有关 ROM/ 数据闪存通用的错误保护内容（FENTRYR 设定错误、大部分非法命令、擦除错误、编程错误和 FCU 错误），请参照 “26. ROM（保存代码的闪存）” 的 “26.8.1 软件保护”。

如果在编程或者擦除处理过程中发行挂起以外的命令，就转移到命令锁定状态，但是 FCU 继续进行编程或者擦除处理。在此状态下无法通过发行 P/E 挂起命令来中止编程或者擦除。如果在命令锁定状态下发行命令，ILGLERR 位就变为 “1”。

表 27.8 错误保护一览表（数据闪存专用）

| 分类 | 内容 | ILGLRR | ERSERR | PRGERR | DFLAE | DFLRPE | CMDLK |
|------------|---|--------|--------|--------|-------|--------|-------|
| 非法命令 | 在编程命令的第 2 周期指定“04h”和“40h”以外的数据。 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 在 FENTRYR.FENTRYD 位为“1”的状态下，对数据闪存区发行锁定位编程命令。 | 1 | 0 | 0 | 0 | 0 | 1 |
| 数据闪存的存取违反 | 在 FENTRYD 位为“1”时并且在数据闪存 P/E 正常模式中，对数据闪存区发行读存取命令。 | 1 | 0 | 0 | 1 | 0 | 1 |
| | 在 FENTRYD 位为“0”的状态下，对数据闪存区发行写存取命令。 | 1 | 0 | 0 | 1 | 0 | 1 |
| | 在 FENTRYR.FENTRY1 位或者 FENTRYR.FENTRY0 位为“1”的状态下，对数据闪存区发行存取命令。 | 1 | 0 | 0 | 1 | 0 | 1 |
| 数据闪存的读保护违反 | 对通过 DFLRE 寄存器设定为禁止读的数据闪存区发行读存取命令。 | 1 | 0 | 0 | 0 | 1 | 1 |
| 数据闪存的写保护违反 | 对通过 DFLWE 寄存器设定为禁止编程和擦除的数据闪存区发行编程 / 块擦除命令。 | 1 | 0 | 0 | 0 | 0 | 1 |

27.8 引导模式

在引导模式中，能从主机发送控制命令和编程数据，并且对数据 MAT 进行编程或者擦除。有关引导模式的系统结构和使用方法的详细内容，请参照“26.10 引导模式”。以下说明数据闪存固有的命令。

27.8.1 查询设定主机命令

数据闪存固有的查询设定主机命令一览表如表 27.9 所示。在“26.10.5 查询设定主机命令的等待状态”和“图 26.29 用户 MAT/ 用户引导 MAT 的查询设定主机命令的使用例子”所示的流程中，“查询 MAT 编程信息”使用数据 MAT 有无查询命令和数据 MAT 信息查询命令。

表 27.9 查询设定主机命令（数据闪存固有）

| 主机命令名 | 功能 |
|-------------|-------------------------|
| 数据 MAT 有无查询 | 查询是否有数据 MAT。 |
| 数据 MAT 信息查询 | 查询数据 MAT 的个数、起始地址和结束地址。 |

以下说明各命令的详细内容。说明内容中的“命令”是从主机发送到 RX610 的命令，“应答”是从 RX610 发送到主机的应答。“校验和”是指在合计 RX610 发送的各字节时，为了得到“00h”而算出的字节数据。

(1) 数据 MAT 有无查询

如果主机发送数据 MAT 有无查询命令，RX610 就发送表示有数据 MAT 的信息。

| | | | | |
|----|-----|----|--------|-----|
| 命令 | 2Ah | | | |
| 应答 | 3Ah | 长度 | 有无 MAT | SUM |

- 【符号说明】 长度（1 字节）：有无 MAT 的字符数（是固定值，为“1”。）
 MAT 有无（1 字节）：有无数据 MAT（是固定值，为“21h”。）
 21h：有数据 MAT
 SUM（1 字节）：校验和

(2) 数据 MAT 信息查询

如果主机发送数据 MAT 信息查询命令，RX610 就发送数据 MAT 的区域数和地址的信息。

| | | | |
|----|---------|----|-----|
| 命令 | 2Bh | | |
| 应答 | 3Bh | 长度 | 区域数 |
| | 区域的起始地址 | | |
| | 区域的结束地址 | | |
| | 区域的起始地址 | | |
| | 区域的结束地址 | | |
| | ... | | |
| | 区域的起始地址 | | |
| | 区域的结束地址 | | |
| | SUM | | |

- 【符号说明】 长度（1 字节）：区域数、区域的起始地址、区域的结束地址的数据总字节数
 区域数（1 字节）：数据 MAT 的区域数（连续区域为 1 个区域）
 区域的起始地址（4 字节）：数据 MAT 区域的起始地址
 区域的最后地址（4 字节）：数据 MAT 区域的结束地址
 SUM（1 字节）：校验和

数据 MAT 的块结构信息包括在擦除块信息查询命令（参照“26.10.5 查询设定主机命令的等待状态”）的应答中。

27.8.2 编程 / 擦除主机命令

数据闪存固有的编程 / 擦除主机命令一览表如表 27.10 所示。数据闪存固有的主机命令只有数据 MAT 校验和命令以及数据 MAT 空白检查命令，而编程命令、擦除命令、读命令和 ROM 通用。

在对数据 MAT 进行编程时，在发行用户 MAT 编程选择命令后，用 256 字节编程命令给编程地址指定数据 MAT 的地址。在擦除数据 MAT 时，在发行擦除选择命令后，用块擦除命令指定数据 MAT 的擦除块。数据 MAT 的擦除块信息包括在擦除块查询命令的应答中。在读数据 MAT 时，用存储器读命令选择用户 MAT 并且给读对象地址指定数据 MAT 的地址。

有关用户 MAT 编程选择命令、用户引导 MAT 编程选择命令、256 字节编程命令、擦除选择命令、块擦除命令和存储器读命令的详细内容，请参照“26.10.7 编程 / 擦除主机命令等待状态”。有关擦除块信息查询命令的详细内容，请参照“26.10.5 查询设定主机命令的等待状态”。

表 27.10 编程 / 擦除主机命令（数据闪存固有）

| 主机命令名 | 功能 |
|-------------|--------------|
| 数据 MAT 校验和 | 数据 MAT 的校验和 |
| 数据 MAT 空白检查 | 数据 MAT 的空白检查 |

以下说明各命令的详细内容。说明内容中的“命令”是从主机发送到 RX610 的命令，“应答”是从 RX610 发送到主机的应答。“校验和”是指在合计 RX610 发送的各字节时，为了得到“00h”而算出的字节数据。

(1) 数据 MAT 校验和

如果主机发送数据 MAT 校验和命令，RX610 就发送以字节为单位将数据 MAT 的数据相加后的结果（校验和）。

命令

| |
|-----|
| 61h |
|-----|

应答

| | | | |
|-----|----|---------|-----|
| 71h | 长度 | MAT 校验和 | SUM |
|-----|----|---------|-----|

【符号说明】 长度（1 字节）：MAT 校验和的字节数（是固定值，为“4”。）
 MAT 校验和（4 字节）：数据 MAT 的校验和结果
 SUM（4 字节）：校验和（应答数据的校验和）

(2) 数据 MAT 空白检查

如果主机发送数据 MAT 空白检查命令，RX610 就检查数据 MAT 是否为全部擦除状态。如果数据 MAT 为全部擦除状态，RX610 就发送应答（06h）。如果数据 MAT 有未擦除的区域，RX610 就发送错误应答（E2h → 52h）。

命令

| |
|-----|
| 62h |
|-----|

应答

| |
|-----|
| 06h |
|-----|

错误应答

| | |
|-----|-----|
| E2h | 52h |
|-----|-----|

27.9 使用时的注意事项

(1) 启动复位后的数据 MAT 保护状态

因为 DFLRE 寄存器和 DFLWE 寄存器的初始值为“0000h”，所以启动复位后的数据 MAT 为禁止读、编程和擦除的状态。如果需要读数据 MAT，就必须在设定 DFLRE 寄存器后存取数据 MAT。如果需要对数据 MAT 进行编程或者擦除，就必须在设定 DFLWE 寄存器后发行编程 / 擦除的 FCU 命令。如果不设定寄存器而进行读、编程或者擦除，FCU 就检测到错误，进入命令锁定状态。

(2) 其他注意事项

其他注意事项和 ROM 相同，请参照“26. ROM（保存代码的闪存）”的“26.13 使用时的注意事项”。

28. 边界扫描

RX610 群内置边界扫描功能，只有 176 引脚 LFBGA 才装载边界扫描功能。

边界扫描是和 JTAG（Joint Test Action Group, IEEE Std.1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture）同等的串行输入 / 输出接口。

28.1 概要

边界扫描的规格如表 28.1 所示。

边界扫描的框图如图 28.1 所示。

表 28.1 边界扫描的规格

| 项目 | 内容 |
|-------------|--|
| 边界扫描有效 / 无效 | 在将 EMLE 引脚置为 Low 电平并且将 BSCANP 引脚置为 High 电平时，边界扫描功能有效。 |
| 边界扫描的专用引脚 | 在边界扫描功能有效时，P02、P03、P04、P05、WDTOVF# 为 JTAG 专用引脚。 |
| 6 种测试模式 | <ul style="list-style-type: none"> • BYPASS 模式 • EXTEST 模式 • SAMPLE/PRELOAD 模式 • CLAMP 模式 • HIGHZ 模式 • IDCODE 模式 |

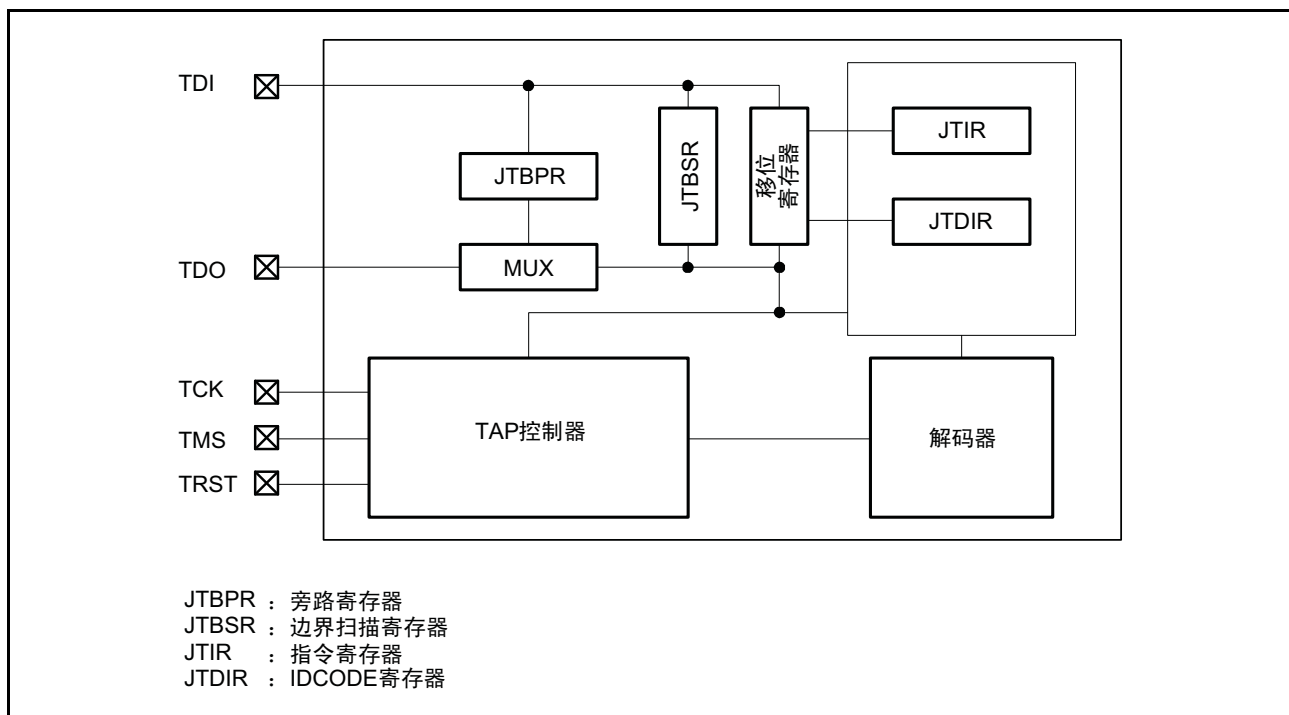


图 28.1 JTAG 的框图

边界扫描使用的输入 / 输出引脚如表 28.2 所示。

表 28.2 JTAG 输入 / 输出引脚

| 引脚名 | 输入 / 输出 | 功 能 |
|------|---------|--|
| TCK | 输入 | 测试时钟的输入 边界扫描的时钟信号 在使用边界扫描功能时，必须输入占空比为 50% 的时钟。 |
| TMS | 输入 | 测试模式的选择 |
| TDI | 输入 | 测试数据的输入 |
| TDO | 输出 | 测试数据的输出 |
| TRST | 输入 | 测试复位的输入 |

28.2 寄存器说明

边界扫描的寄存器一览表如表 28.3 所示。

表 28.3 边界扫描的寄存器一览表

| 寄存器名 | 符号 | 复位后的值 | 地址 | 存取长度 |
|------------|-------|------------|----|------|
| 指令寄存器 | JTIR | 4h | — | 4 |
| 旁路寄存器 | JTBPR | 不定值 | — | 1 |
| 边界扫描寄存器 | JTBSR | 不定值 | — | — |
| IDCODE 寄存器 | JTIDR | 0809 9447h | — | 32 |

能通过串行传送将指令从 TDI 引脚输入到 JTIR 寄存器。

JTBPR 寄存器是 1 位寄存器，在 BYPASS 模式中将 TDI 引脚和 TDO 引脚连接到此寄存器。

JTBSR 寄存器是表 28.6 结构的寄存器，在移入测试数据时，在 TDI 引脚和 TDO 引脚之间连接此寄存器。

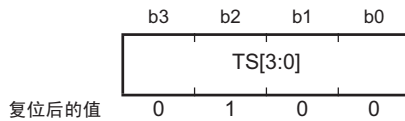
CPU 不能存取全部的寄存器。

各寄存器的串行传送如表 28.4 所示。

表 28.4 寄存器的串行传送

| 寄存器 | 串行输入 | 串行输出 |
|-------|------|------|
| JTIR | 能 | 能 |
| JTBPR | 能 | 能 |
| JTBSR | 能 | 能 |
| JTIDR | 能 | 能 |

28.2.1 指令寄存器 (JTIR)



| 位 | 符号 | 位名 | 功能 | R/W |
|-------|---------|--------|-----------------|-----|
| b3-b0 | TS[3:0] | 测试位的置位 | 命令结构如表 28.5 所示。 | — |

表 28.5 命令结构

| TS3 | TS2 | TS1 | TS0 | 指令 |
|------|-----|-----|-----|----------------|
| 0 | 0 | 0 | 0 | EXTEST |
| 0 | 0 | 0 | 1 | SAMPLE/PRELOAD |
| 0 | 1 | 0 | 0 | IDCODE (初始值) |
| 0 | 1 | 1 | 0 | CLAMP |
| 0 | 1 | 1 | 1 | HIGHZ |
| 1 | 1 | 1 | 1 | BYPASS |
| 上述以外 | | | | 保留 |

JTIR 寄存器是 4 位寄存器。

能通过 TDI 引脚的串行输入将边界扫描的指令传送到 JTIR 寄存器。

JTIR 寄存器在 TRST 引脚为 Low 电平时或者在 TAP 控制器为 Test-Logic-Reset 的状态下被初始化。

28.2.2 旁路寄存器 (JTBPR)

JTBPR 寄存器是 1 位寄存器。

如果设定为 BYPASS 模式，就将 TDI 引脚和 TDO 引脚连接到 JTBPR 寄存器。

CPU 不能读写 JTBPR 寄存器。

28.2.3 边界扫描寄存器 (JTBSR)

JTBSR 寄存器是为了控制 LSI 的输入 / 输出引脚而被配置在 PAD 上的移位寄存器。

能使用 EXTEST、SAMPLE/PRELOAD、CLAMP 和 HIGHZ 命令进行边界扫描测试。

LSI 的引脚和边界扫描寄存器的对应如表 28.6 所示。

复位后的值为不定值。

表 28.6 边界扫描寄存器 (1 / 10)

| from TDI | | | |
|--------------|-----|---------|-----|
| 176 引脚 LFBGA | 引脚名 | 输入 / 输出 | 位名 |
| B1 | P67 | 输入 | 411 |
| | | 输出允许 | 410 |
| | | 输出 | 409 |
| C2 | P66 | 输入 | 408 |
| | | 输出允许 | 407 |
| | | 输出 | 406 |
| D2 | P01 | 输入 | 405 |
| | | 输出允许 | 404 |
| | | 输出 | 403 |
| E4 | P00 | 输入 | 402 |
| | | 输出允许 | 401 |
| | | 输出 | 400 |
| D1 | P65 | 输入 | 399 |
| | | 输出允许 | 398 |
| | | 输出 | 397 |
| F4 | MDE | 输入 | 396 |
| F1 | MD1 | 输入 | 395 |
| F2 | MD0 | 输入 | 394 |
| G4 | P86 | 输入 | 393 |
| | | 输出允许 | 392 |
| | | 输出 | 391 |
| G3 | P85 | 输入 | 390 |
| | | 输出允许 | 389 |
| | | 输出 | 388 |
| H2 | NMI | 输入 | 387 |
| J4 | P34 | 输入 | 386 |
| | | 输出允许 | 385 |
| | | 输出 | 384 |
| J3 | PF6 | 输入 | 383 |
| | | 输出允许 | 382 |
| | | 输出 | 381 |
| J1 | PF5 | 输入 | 380 |
| | | 输出允许 | 379 |
| | | 输出 | 378 |
| J2 | PF4 | 输入 | 377 |
| | | 输出允许 | 376 |
| | | 输出 | 375 |
| K4 | P33 | 输入 | 374 |
| | | 输出允许 | 373 |
| | | 输出 | 372 |

表 28.6 边界扫描寄存器 (2 / 10)

| from TDI | | | |
|--------------|-----|---------|-----|
| 176 引脚 LFBGA | 引脚名 | 输入 / 输出 | 位名 |
| K3 | P32 | 输入 | 371 |
| | | 输出允许 | 370 |
| | | 输出 | 369 |
| K1 | P31 | 输入 | 368 |
| | | 输出允许 | 367 |
| | | 输出 | 366 |
| K2 | P30 | 输入 | 365 |
| | | 输出允许 | 364 |
| | | 输出 | 363 |
| L3 | PF3 | 输入 | 362 |
| | | 输出允许 | 361 |
| | | 输出 | 360 |
| L1 | PF2 | 输入 | 359 |
| | | 输出允许 | 358 |
| | | 输出 | 357 |
| L2 | PF1 | 输入 | 356 |
| | | 输出允许 | 355 |
| | | 输出 | 354 |
| L4 | PF0 | 输入 | 353 |
| | | 输出允许 | 352 |
| | | 输出 | 351 |
| M1 | P27 | 输入 | 350 |
| | | 输出允许 | 349 |
| | | 输出 | 348 |
| M2 | P26 | 输入 | 347 |
| | | 输出允许 | 346 |
| | | 输出 | 345 |
| N1 | P25 | 输入 | 344 |
| | | 输出允许 | 343 |
| | | 输出 | 342 |
| N2 | P24 | 输入 | 341 |
| | | 输出允许 | 340 |
| | | 输出 | 339 |
| P1 | P23 | 输入 | 338 |
| | | 输出允许 | 337 |
| | | 输出 | 336 |
| P2 | P22 | 输入 | 335 |
| | | 输出允许 | 334 |
| | | 输出 | 333 |
| R1 | P21 | 输入 | 332 |
| | | 输出允许 | 331 |
| | | 输出 | 330 |

表 28.6 边界扫描寄存器 (3 / 10)

| from TDI | | | |
|--------------|-----|---------|-----|
| 176 引脚 LFBGA | 引脚名 | 输入 / 输出 | 位名 |
| N3 | P20 | 输入 | 329 |
| | | 输出允许 | 328 |
| | | 输出 | 327 |
| R2 | P17 | 输入 | 326 |
| | | 输出允许 | 325 |
| | | 输出 | 324 |
| N4 | P16 | 输入 | 323 |
| | | 输出允许 | 322 |
| | | 输出 | 321 |
| P4 | P15 | 输入 | 320 |
| | | 输出允许 | 319 |
| | | 输出 | 318 |
| M5 | P14 | 输入 | 317 |
| | | 输出允许 | 316 |
| | | 输出 | 315 |
| R4 | P13 | 输入 | 314 |
| | | 输出允许 | 313 |
| | | 输出 | 312 |
| N5 | P12 | 输入 | 311 |
| | | 输出允许 | 310 |
| | | 输出 | 309 |
| P5 | P11 | 输入 | 308 |
| | | 输出允许 | 307 |
| | | 输出 | 306 |
| R5 | P10 | 输入 | 305 |
| | | 输出允许 | 304 |
| | | 输出 | 303 |
| M6 | P37 | 输入 | 302 |
| | | 输出允许 | 301 |
| | | 输出 | 300 |
| N6 | P36 | 输入 | 299 |
| | | 输出允许 | 298 |
| | | 输出 | 297 |
| R6 | P35 | 输入 | 296 |
| | | 输出允许 | 295 |
| | | 输出 | 294 |
| P6 | P84 | 输入 | 293 |
| | | 输出允许 | 292 |
| | | 输出 | 291 |
| M7 | P57 | 输入 | 290 |
| | | 输出允许 | 289 |
| | | 输出 | 288 |

表 28.6 边界扫描寄存器 (4 / 10)

| from TDI | | | |
|--------------|-----|---------|-----|
| 176 引脚 LFBGA | 引脚名 | 输入 / 输出 | 位名 |
| N7 | P56 | 输入 | 287 |
| | | 输出允许 | 286 |
| | | 输出 | 285 |
| R7 | P55 | 输入 | 284 |
| | | 输出允许 | 283 |
| | | 输出 | 282 |
| P7 | P54 | 输入 | 281 |
| | | 输出允许 | 280 |
| | | 输出 | 279 |
| M8 | P83 | 输入 | 278 |
| | | 输出允许 | 277 |
| | | 输出 | 276 |
| R8 | P82 | 输入 | 275 |
| | | 输出允许 | 274 |
| | | 输出 | 273 |
| M9 | P81 | 输入 | 272 |
| | | 输出允许 | 271 |
| | | 输出 | 270 |
| N9 | P80 | 输入 | 269 |
| | | 输出允许 | 268 |
| | | 输出 | 267 |
| R9 | P53 | 输入 | 266 |
| | | 输出允许 | 265 |
| | | 输出 | 264 |
| P9 | P52 | 输入 | 263 |
| | | 输出允许 | 262 |
| | | 输出 | 261 |
| M10 | P51 | 输入 | 260 |
| | | 输出允许 | 259 |
| | | 输出 | 258 |
| N10 | P50 | 输入 | 257 |
| | | 输出允许 | 256 |
| | | 输出 | 255 |
| R10 | PH7 | 输入 | 254 |
| | | 输出允许 | 253 |
| | | 输出 | 252 |
| P10 | PH6 | 输入 | 251 |
| | | 输出允许 | 250 |
| | | 输出 | 249 |
| R11 | PH5 | 输入 | 248 |
| | | 输出允许 | 247 |
| | | 输出 | 246 |

表 28.6 边界扫描寄存器 (5 / 10)

| from TDI | | | |
|--------------|-----|---------|-----|
| 176 引脚 LFBGA | 引脚名 | 输入 / 输出 | 位名 |
| M11 | PH4 | 输入 | 245 |
| | | 输出允许 | 244 |
| | | 输出 | 243 |
| R12 | PH3 | 输入 | 242 |
| | | 输出允许 | 241 |
| | | 输出 | 240 |
| P12 | P77 | 输入 | 239 |
| | | 输出允许 | 238 |
| | | 输出 | 237 |
| N12 | P76 | 输入 | 236 |
| | | 输出允许 | 235 |
| | | 输出 | 234 |
| R13 | P75 | 输入 | 233 |
| | | 输出允许 | 232 |
| | | 输出 | 231 |
| M12 | PC7 | 输入 | 230 |
| | | 输出允许 | 229 |
| | | 输出 | 228 |
| P13 | PC6 | 输入 | 227 |
| | | 输出允许 | 226 |
| | | 输出 | 225 |
| R14 | PC5 | 输入 | 224 |
| | | 输出允许 | 223 |
| | | 输出 | 222 |
| P14 | PC4 | 输入 | 221 |
| | | 输出允许 | 220 |
| | | 输出 | 219 |
| R15 | PC3 | 输入 | 218 |
| | | 输出允许 | 217 |
| | | 输出 | 216 |
| N13 | PH2 | 输入 | 215 |
| | | 输出允许 | 214 |
| | | 输出 | 213 |
| N14 | PC2 | 输入 | 212 |
| | | 输出允许 | 211 |
| | | 输出 | 210 |
| N15 | PC1 | 输入 | 209 |
| | | 输出允许 | 208 |
| | | 输出 | 207 |
| M14 | PC0 | 输入 | 206 |
| | | 输出允许 | 205 |
| | | 输出 | 204 |

表 28.6 边界扫描寄存器 (6 / 10)

| from TDI | | | |
|--------------|-----|---------|-----|
| 176 引脚 LFBGA | 引脚名 | 输入 / 输出 | 位名 |
| L12 | PB7 | 输入 | 203 |
| | | 输出允许 | 202 |
| | | 输出 | 201 |
| M15 | PB6 | 输入 | 200 |
| | | 输出允许 | 199 |
| | | 输出 | 198 |
| L13 | PB5 | 输入 | 197 |
| | | 输出允许 | 196 |
| | | 输出 | 195 |
| L14 | PB4 | 输入 | 194 |
| | | 输出允许 | 193 |
| | | 输出 | 192 |
| L15 | PB3 | 输入 | 191 |
| | | 输出允许 | 190 |
| | | 输出 | 189 |
| K12 | PB2 | 输入 | 188 |
| | | 输出允许 | 187 |
| | | 输出 | 186 |
| K13 | PB1 | 输入 | 185 |
| | | 输出允许 | 184 |
| | | 输出 | 183 |
| K15 | P74 | 输入 | 182 |
| | | 输出允许 | 181 |
| | | 输出 | 180 |
| K14 | P73 | 输入 | 179 |
| | | 输出允许 | 178 |
| | | 输出 | 177 |
| J12 | P72 | 输入 | 176 |
| | | 输出允许 | 175 |
| | | 输出 | 174 |
| J13 | P71 | 输入 | 173 |
| | | 输出允许 | 172 |
| | | 输出 | 171 |
| J15 | P70 | 输入 | 170 |
| | | 输出允许 | 169 |
| | | 输出 | 168 |
| H12 | PB0 | 输入 | 167 |
| | | 输出允许 | 166 |
| | | 输出 | 165 |
| H15 | PH1 | 输入 | 164 |
| | | 输出允许 | 163 |
| | | 输出 | 162 |

表 28.6 边界扫描寄存器 (7 / 10)

| from TDI | | | |
|--------------|-----|---------|-----|
| 176 引脚 LFBGA | 引脚名 | 输入 / 输出 | 位名 |
| H14 | PH0 | 输入 | 161 |
| | | 输出允许 | 160 |
| | | 输出 | 159 |
| G12 | PA7 | 输入 | 158 |
| | | 输出允许 | 157 |
| | | 输出 | 156 |
| G13 | PA6 | 输入 | 155 |
| | | 输出允许 | 154 |
| | | 输出 | 153 |
| G15 | PA5 | 输入 | 152 |
| | | 输出允许 | 151 |
| | | 输出 | 150 |
| G14 | PA4 | 输入 | 149 |
| | | 输出允许 | 148 |
| | | 输出 | 147 |
| F12 | PA3 | 输入 | 146 |
| | | 输出允许 | 145 |
| | | 输出 | 144 |
| F13 | PA2 | 输入 | 143 |
| | | 输出允许 | 142 |
| | | 输出 | 141 |
| F15 | PA1 | 输入 | 140 |
| | | 输出允许 | 139 |
| | | 输出 | 138 |
| F14 | PA0 | 输入 | 137 |
| | | 输出允许 | 136 |
| | | 输出 | 135 |
| E13 | PG7 | 输入 | 134 |
| | | 输出允许 | 133 |
| | | 输出 | 132 |
| E14 | PG6 | 输入 | 131 |
| | | 输出允许 | 130 |
| | | 输出 | 129 |
| D15 | PG5 | 输入 | 128 |
| | | 输出允许 | 127 |
| | | 输出 | 126 |
| D14 | PE7 | 输入 | 125 |
| | | 输出允许 | 124 |
| | | 输出 | 123 |
| D13 | PE6 | 输入 | 122 |
| | | 输出允许 | 121 |
| | | 输出 | 120 |

表 28.6 边界扫描寄存器 (8 / 10)

| from TDI | | | |
|--------------|-----|---------|-----|
| 176 引脚 LFBGA | 引脚名 | 输入 / 输出 | 位名 |
| C15 | PE5 | 输入 | 119 |
| | | 输出允许 | 118 |
| | | 输出 | 117 |
| D12 | PE4 | 输入 | 116 |
| | | 输出允许 | 115 |
| | | 输出 | 114 |
| C14 | PE3 | 输入 | 113 |
| | | 输出允许 | 112 |
| | | 输出 | 111 |
| B15 | PE2 | 输入 | 110 |
| | | 输出允许 | 109 |
| | | 输出 | 108 |
| B14 | PE1 | 输入 | 107 |
| | | 输出允许 | 106 |
| | | 输出 | 105 |
| A15 | PE0 | 输入 | 104 |
| | | 输出允许 | 103 |
| | | 输出 | 102 |
| C13 | PD7 | 输入 | 101 |
| | | 输出允许 | 100 |
| | | 输出 | 99 |
| A14 | PD6 | 输入 | 98 |
| | | 输出允许 | 97 |
| | | 输出 | 96 |
| B13 | PD5 | 输入 | 95 |
| | | 输出允许 | 94 |
| | | 输出 | 93 |
| A13 | PD4 | 输入 | 92 |
| | | 输出允许 | 91 |
| | | 输出 | 90 |
| D11 | P64 | 输入 | 89 |
| | | 输出允许 | 88 |
| | | 输出 | 87 |
| A12 | P63 | 输入 | 86 |
| | | 输出允许 | 85 |
| | | 输出 | 84 |
| C11 | P62 | 输入 | 83 |
| | | 输出允许 | 82 |
| | | 输出 | 81 |
| B11 | P61 | 输入 | 80 |
| | | 输出允许 | 79 |
| | | 输出 | 78 |

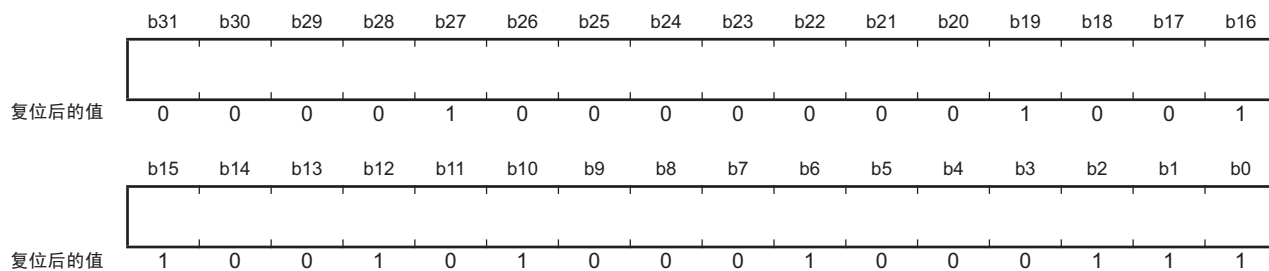
表 28.6 边界扫描寄存器 (9 / 10)

| from TDI | | | |
|--------------|-----|---------|----|
| 176 引脚 LFBGA | 引脚名 | 输入 / 输出 | 位名 |
| A11 | P60 | 输入 | 77 |
| | | 输出允许 | 76 |
| | | 输出 | 75 |
| D10 | PD3 | 输入 | 74 |
| | | 输出允许 | 73 |
| | | 输出 | 72 |
| C10 | PD2 | 输入 | 71 |
| | | 输出允许 | 70 |
| | | 输出 | 69 |
| A10 | PD1 | 输入 | 68 |
| | | 输出允许 | 67 |
| | | 输出 | 66 |
| B10 | PD0 | 输入 | 65 |
| | | 输出允许 | 64 |
| | | 输出 | 63 |
| D9 | PG4 | 输入 | 62 |
| | | 输出允许 | 61 |
| | | 输出 | 60 |
| C9 | PG3 | 输入 | 59 |
| | | 输出允许 | 58 |
| | | 输出 | 57 |
| A9 | PG2 | 输入 | 56 |
| | | 输出允许 | 55 |
| | | 输出 | 54 |
| B9 | PG1 | 输入 | 53 |
| | | 输出允许 | 52 |
| | | 输出 | 51 |
| D8 | PG0 | 输入 | 50 |
| | | 输出允许 | 49 |
| | | 输出 | 48 |
| A8 | P97 | 输入 | 47 |
| | | 输出允许 | 46 |
| | | 输出 | 45 |
| B8 | P96 | 输入 | 44 |
| | | 输出允许 | 43 |
| | | 输出 | 42 |
| D7 | P95 | 输入 | 41 |
| | | 输出允许 | 40 |
| | | 输出 | 39 |
| C7 | P94 | 输入 | 38 |
| | | 输出允许 | 37 |
| | | 输出 | 36 |

表 28.6 边界扫描寄存器 (10 / 10)

| from TDI | | | |
|--------------|-----|---------|----|
| 176 引脚 LFBGA | 引脚名 | 输入 / 输出 | 位名 |
| A7 | P93 | 输入 | 35 |
| | | 输出允许 | 34 |
| | | 输出 | 33 |
| B7 | P92 | 输入 | 32 |
| | | 输出允许 | 31 |
| | | 输出 | 30 |
| D6 | P91 | 输入 | 29 |
| | | 输出允许 | 28 |
| | | 输出 | 27 |
| A6 | P90 | 输入 | 26 |
| | | 输出允许 | 25 |
| | | 输出 | 24 |
| C5 | P47 | 输入 | 23 |
| | | 输出允许 | 22 |
| | | 输出 | 21 |
| A5 | P46 | 输入 | 20 |
| | | 输出允许 | 19 |
| | | 输出 | 18 |
| B5 | P45 | 输入 | 17 |
| | | 输出允许 | 16 |
| | | 输出 | 15 |
| D5 | P44 | 输入 | 14 |
| | | 输出允许 | 13 |
| | | 输出 | 12 |
| A4 | P43 | 输入 | 11 |
| | | 输出允许 | 10 |
| | | 输出 | 9 |
| B4 | P42 | 输入 | 8 |
| | | 输出允许 | 7 |
| | | 输出 | 6 |
| C4 | P41 | 输入 | 5 |
| | | 输出允许 | 4 |
| | | 输出 | 3 |
| D4 | P40 | 输入 | 2 |
| | | 输出允许 | 1 |
| | | 输出 | 0 |

28.2.4 IDCODE 寄存器 (JTID)



| 位 | 功能 | R/W |
|--------|----------------------|-----|
| b31-b0 | 表示设备 IDCODE 的固定值的寄存器 | — |

JTID 寄存器是 32 位寄存器。

在执行 IDCODE 指令时，从 TDO 引脚输出 JTID 寄存器的数据。

28.3 运行说明

边界扫描功能在将 EMLE 引脚置为 Low 电平并且将 BSCANP 引脚置为 High 电平时有效。

28.3.1 TAP 控制器

TAP 控制器的状态转移图如图 28.2 所示。

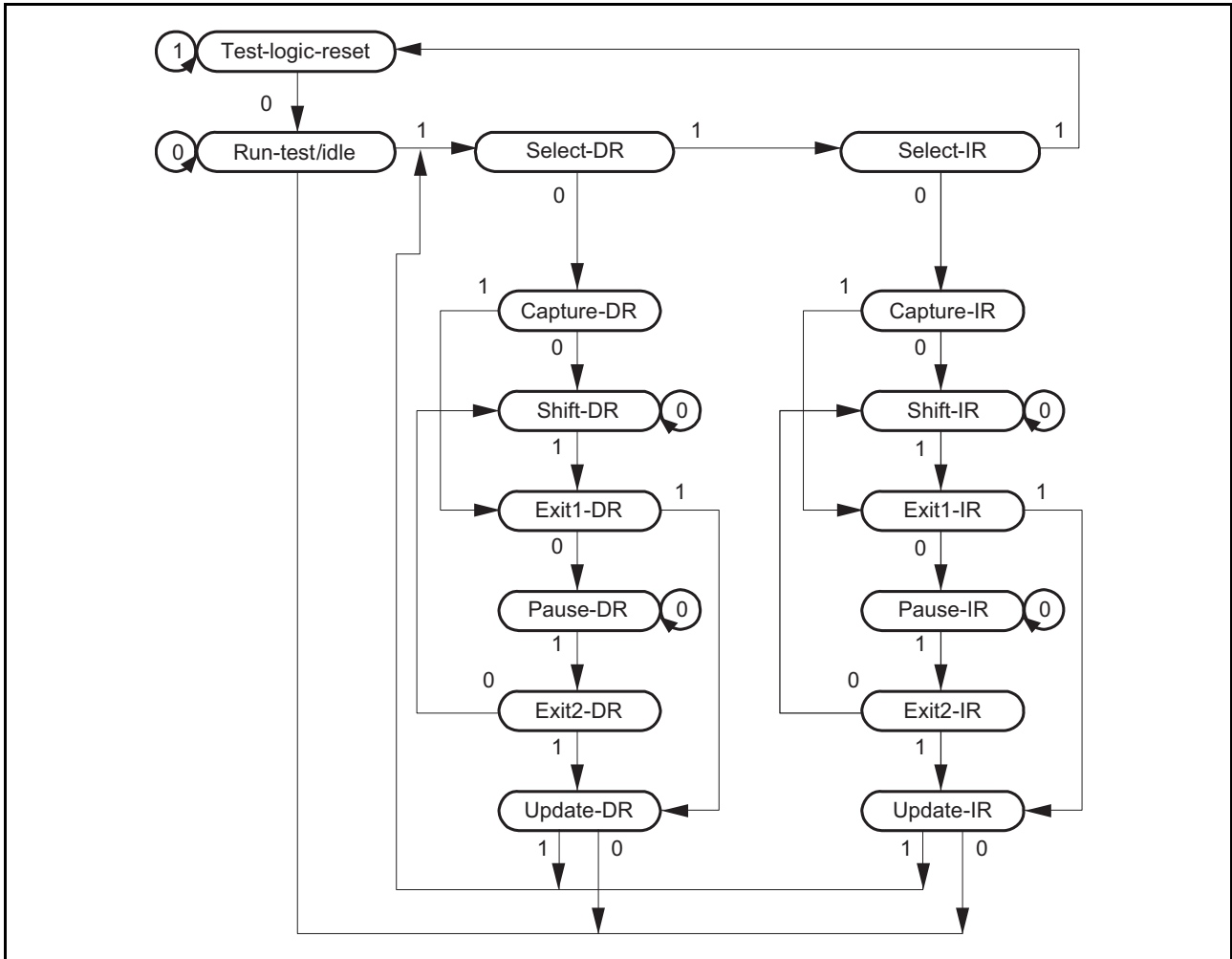


图 28.2 TAP 控制器

28.3.2 命令一览

(1) BYPASS[指令码: 1111b]

BYPASS 指令是使 JTBPR 寄存器运行的指令，缩短移位路径，使印刷电路板上的其他 LSI 串行数据的传送速度高速化。在执行此指令时，测试电路对系统电路没有任何影响。

在 TDI 引脚和 TDO 引脚之间连接 JTBPR 寄存器。在 Shift-DR 运行中变为旁路运行。在 Shift-DR 的第 1 个时钟，TDO 为 Low 电平；在以后的 Shift-DR 时钟，从 TDO 输出 TDI。

(2) EXTEST[指令码: 0000b]

EXTEST 指令是在将 LSI 安装到印刷电路板时测试外部电路的指令。在执行此指令时，输出引脚用于将测试数据（已用 SAMPLE/PRELOAD 指令进行了设定）从 JTBSR 寄存器输出到印刷电路板，输入引脚用于将测试结果从印刷电路板取到 JTBSR 寄存器。

(3) SAMPLE/PRELOAD[指令码: 0001b]

SAMPLE/PRELOAD 指令是将值从 LSI 的内部电路输入到 JTBSR 寄存器并且从扫描路径输出或者将数据加载到扫描路径的指令。在执行此指令时，将 LSI 的输入引脚的信号传输到内部电路，并且将内部电路的值从输出引脚输出到外部。在执行此指令时，LSI 的系统电路不受任何影响。

在 SAMPLE 执行过程中，将从输入引脚传送到内部电路的值或者从内部电路传送到输出引脚的值的快照取到 JTBSR 寄存器，并且从扫描路径中读数据。取快照与 Capture-DR 状态的 TCK 的上升沿同步，而不妨碍 LSI 的正常运行。

在 PRELOAD 执行过程中，在执行 EXTEST 指令之前先将初始值从扫描路径设定到 JTBSR 寄存器的并行输出锁存器。如果不执行 PRELOAD 而执行 EXTEST 指令，就在最初的扫描顺序结束（传送到输出锁存器）前从输出引脚输出不定值（EXTEST 指令总是将并行输出锁存器的内容输出到输出引脚）。

(4) IDCODE[指令码: 0100b]

如果选择 IDCODE 指令，就在 TAP 控制器的 Shift-DR 状态下 TDO 引脚从 LSB 开始输出 JTIDR 寄存器的值。在执行此指令时，测试电路对系统电路没有任何影响。在 TAP 控制器的 Test-Logic-Reset 状态下，指令寄存器被初始化为 IDCODE 指令。

(5) CLAMP[指令码: 0110b]

如果选择 CLAMP 指令，输出引脚就输出预先通过 SAMPLE/PRELOAD 指令设定的 JTBSR 寄存器的值。在选择了 CLAMP 指令的期间，JTBSR 寄存器的状态与 TAP 控制器的状态无关，保持以前的状态。

此指令将旁路寄存器连接在 TDI 引脚和 TDO 引脚之间，和选择 BYPASS 模式时的运行一样。

(6) HIGHZ[指令码: 0111b]

如果选择 HIGHZ 指令，全部的输出引脚就变为高阻抗状态。在选择了 HIGHZ 指令的期间，JTBSR 寄存器的状态与 TAP 控制器的状态无关，保持以前的状态。

此指令将旁路寄存器连接在 TDI 引脚和 TDO 引脚之间，和选择 BYPASS 指令时的运行一样。

28.4 使用时的注意事项

1. 如图28.3所示，在进行串行传送时，从LSB侧开始输入/输出数据。

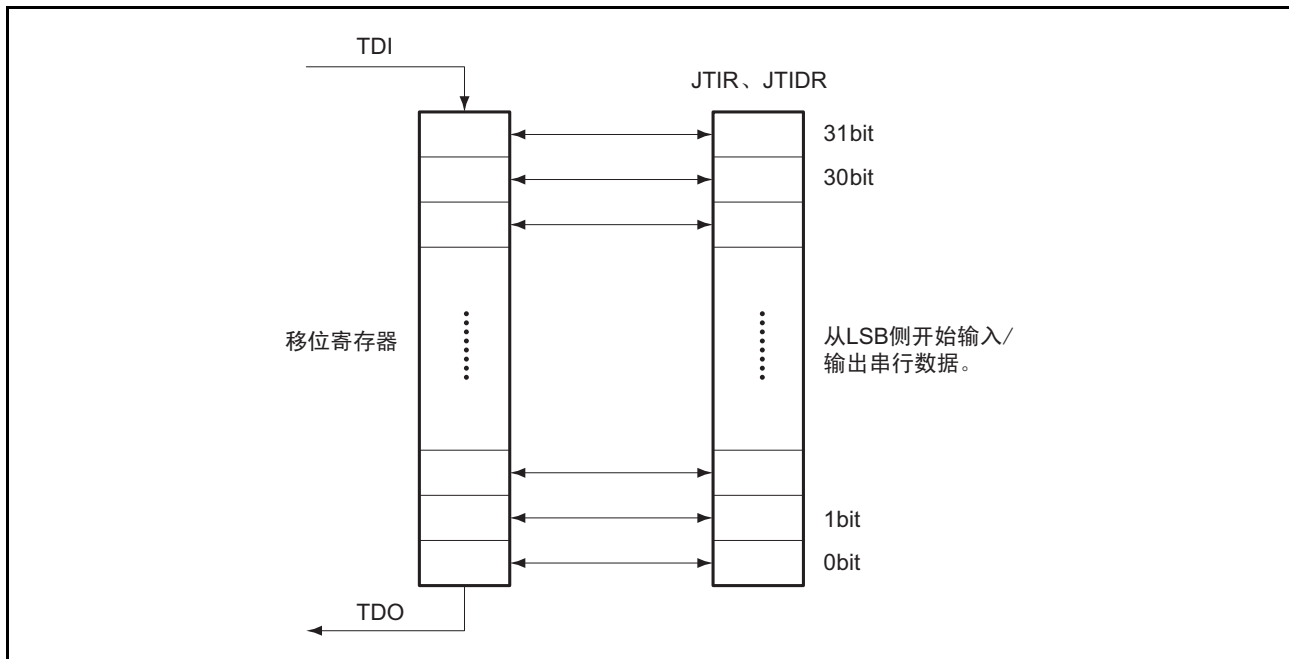


图 28.3 串行数据的输入 / 输出

2. 必须通过上拉电阻将边界扫描的引脚（TCK、TDI、TMS、TRST#）进行上拉。
3. 电源引脚（VCC、VCL、VSS、AVCC、AVSS、VREFH、VREFL、PLLCC、PLLSS）不是边界扫描对象。
4. 时钟引脚（EXTAL、XTAL）不是边界扫描对象。
5. 复位引脚（RES#）不是边界扫描对象。
6. 边界扫描的引脚（TCK、TMS、TRST#、TDI、TDO）不是边界扫描对象。
7. 在以下的情况下，不能使用边界扫描功能。
 - 复位状态
 - 软件待机模式、深度软件待机模式
8. 如果在将有漏极开路功能的引脚置为漏极开路功能有效的情况下使用，就通过边界扫描功能将输出的扫描寄存器和输出寄存器设定为1。如果执行EXTTEST指令、CLAMP指令或者SAMPLE/PRELOAD指令，引脚就不变为高阻抗而变为High电平输出。
9. P14～P17的引脚结构如图28.4（1）所示。如果将P14～P17用作IIC引脚（SDA0、SDA1、SCL0、SCL1），就在使用边界扫描功能时需要注意与漏极开路输出的冲突和寄生电流。
10. P40～P47和P90～P97的引脚结构如图28.4（2）所示。如果将P40～P47、P90～P97用作AD输入（AN0～AN7、AN8～AN15），就在使用边界扫描功能时需要注意与AD输入的冲突和寄生电流。
11. P66和P67的引脚结构如图28.4（3）所示。如果将P66和P67用作DA输出（DA0、DA1），就在使用边界扫描功能时需要注意与DA输出的冲突和寄生电流。

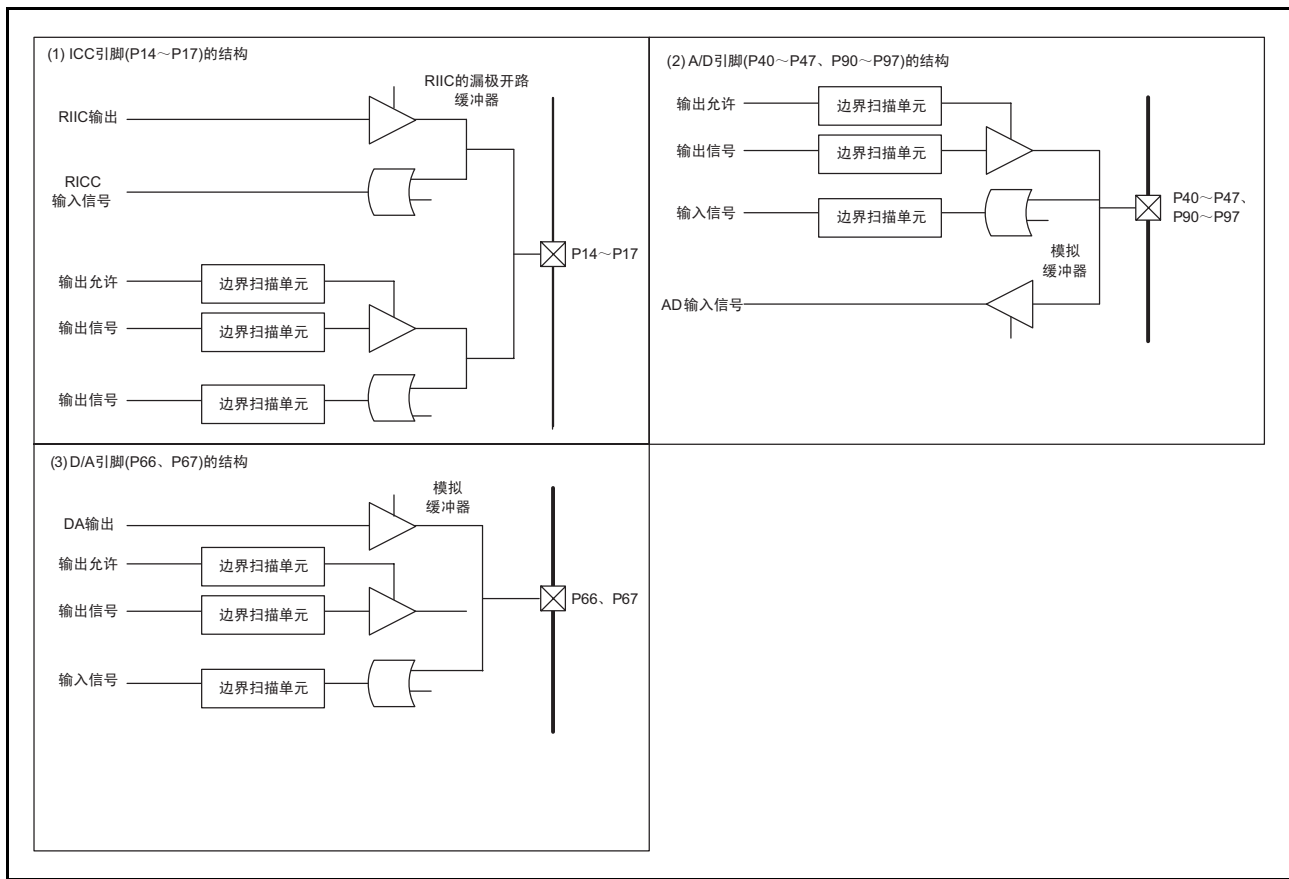


图 28.4 各引脚的结构

29. 电特性

29.1 绝对最大额定值

表 29.1 绝对最大额定值

| 项目 | 符号 | 额定值 | 单位 |
|-------------------------------|-------------------------|-----------------------|----|
| 电源电压 | V_{CC} 、 $PLL V_{CC}$ | -0.3 ~ +4.6 | V |
| 输入电压 (端口 0 和端口 14 ~ 17 除外) | V_{in} | -0.3 ~ $V_{CC}+0.3$ | V |
| 输入电压 (端口 0 和端口 14 ~ 17 (注 1)) | V_{in} | -0.3 ~ +6.5 | V |
| 基准电源电压 | V_{REFH} | -0.3 ~ $V_{CC}+0.3$ | V |
| 模拟电源电压 | AV_{CC} (注 2) | -0.3 ~ +4.6 | V |
| 模拟输入电压 | V_{AN} | -0.3 ~ $V_{CC}+0.3$ | V |
| 工作温度 | T_{opr} | 普通规格产品: -20 ~ +85 | °C |
| | | 大温度范围的规格产品: -40 ~ +85 | |
| 保存温度 | T_{stg} | -55 ~ +125 | °C |

【使用时的注意事项】如果在使用 LSI 时超过绝对最大额定值，就有可能造成 LSI 的永久性破坏。

注 1. 端口 0 和端口 14 ~ 17 对应 5V 容许。

注 2. AV_{CC} 必须连接 V_{CC} 。在不使用 A/D 转换器和 D/A 转换器时，不能将 AV_{CC} 、 AV_{SS} 、 V_{REFH} 、 V_{REFL} 引脚置为开路。 AV_{CC} 引脚和 V_{REFH} 引脚必须连接 V_{CC} ， AV_{SS} 引脚和 V_{REFL} 引脚连接 V_{SS} 。

29.2 DC 特性

表 29.2 DC 特性 (1)

条件: $V_{CC}=PLL V_{CC}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL V_{SS}=V_{REFL}=0V$
 $T_a=-20 \sim +85^{\circ}C$ (普通规格产品), $T_a=-40 \sim +85^{\circ}C$ (大温度范围的规格产品)

| 项目 | 符号 | min | typ | max | 单位 | 测量条件 | | |
|-----------------------------------|---|--------------|----------------------|-----|-----------------------------------|---------|--|------------------|
| 施密特触发 输入电压 | IRQ 输入引脚 (注1)、 TPU 输入引脚 (注1)、 TMR 输入引脚 (注1)、 SCI 输入引脚 (注1)、 ADTRG# 输入引脚 (注1)、 RES#、NMI | V_{IH} | $V_{CC} \times 0.8$ | — | $V_{CC} + 0.3$ | V | | |
| | | V_{IL} | -0.3 | — | $V_{CC} \times 0.2$ | | | |
| | | ΔV_T | $V_{CC} \times 0.06$ | — | — | | | |
| | RIIC 输入引脚 | V_{IH} | $V_{CC} \times 0.7$ | — | 5.8 | | | |
| | | V_{IL} | -0.3 | — | $V_{CC} \times 0.3$ | | | |
| | | ΔV_T | $V_{CC} \times 0.05$ | — | — | | | |
| | 端口 0 和端口 14 ~ 17 (注2) | V_{IH} | $V_{CC} \times 0.8$ | — | 5.8 | | | |
| | | V_{IL} | -0.3 | — | $V_{CC} \times 0.2$ | | | |
| | 端口 10 ~ 13、 端口 2 ~ E (144 引脚 LQFP)、 端口 2 ~ H (176 引脚 LFBGA)、 其他输入引脚 | V_{IH} | $V_{CC} \times 0.8$ | — | $V_{CC} + 0.3$ | | | |
| | | V_{IL} | -0.3 | — | $V_{CC} \times 0.2$ | | | |
| 输入 High 电平电压 (施密特触发输入引脚 除外) | MD 引脚、EMLE | V_{IH} | $V_{CC} \times 0.9$ | — | $V_{CC} + 0.3$ | V | | |
| | EXTAL | | $V_{CC} \times 0.8$ | — | $V_{CC} + 0.3$ | | | |
| | D0 ~ D15 | | $V_{CC} \times 0.7$ | — | $V_{CC} + 0.3$ | | | |
| 输入 Low 电平电压 (施密特触发输入引脚 除外) | MD 引脚、EMLE | V_{IL} | -0.3 | — | $V_{CC} \times 0.1$ | V | | |
| | EXTAL | | -0.3 | — | $V_{CC} \times 0.2$ | | | |
| | D0 ~ D15 | | -0.3 | — | $V_{CC} \times 0.3$ | | | |
| 输出 High 电平电压 | 全部输出引脚 | V_{OH} | $V_{CC} - 0.5$ | — | — | V | $I_{OH} = -1mA$ | |
| 输出 Low 电平电压 | 全部输出引脚 (RIIC 引脚除外) | V_{OL} | — | — | 0.5 | V | $I_{OL} = 1.0mA$ | |
| | | | RIIC 引脚 | — | — | | 0.4 | $I_{OL} = 3.0mA$ |
| | | | — | — | 0.6 | | $I_{OL} = 6.0mA$ | |
| | RIIC 引脚 (只限于 P14 和 P15 的 通道 1) | | — | — | 0.4 | | $I_{OL} = 15mA$ (ICFER.FMPE=1) | |
| — | — | 0.4 | — | — | $I_{OL} = 20mA$ (ICFER.FMPE=1) | | | |
| 输入泄漏电流 | RES#、MD 引脚、EMLE、NMI | $ I_{in} $ | — | — | 1.0 | μA | $V_{in} = 0V, V_{CC}$ | |
| 三态泄漏电流 (OFF 状态) | 端口 10 ~ 13、 端口 2 ~ E (144 引脚 LQFP)、 端口 2 ~ H (176 引脚 LFBGA) | $ I_{TSI} $ | — | — | 1.0 | μA | $V_{in} = 0V, V_{CC}$ | |
| | | | — | — | 5.0 | | | |
| | 端口 0 和端口 14 ~ 17 | | — | — | 5.0 | | | |
| 输入上拉电阻电流 | 端口 A ~ E | $-I_p$ | 10 | — | 300 | μA | $V_{CC} = 3.0 \sim 3.6V,$ $V_{in} = 0V$ | |
| 输入电容 | 全部输入引脚 (端口 0 和端口 14 ~ 17 除外) | C_{in} | — | — | 15 | pF | $V_{in} = 0V,$ $f = 1MHz,$ $T_a = 25^{\circ}C$ | |
| | 端口 0 和端口 14 ~ 17 | | — | — | 30 | | | |

表 29.2 DC 特性 (2)

条件: $V_{CC}=PLL V_{CC}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL V_{SS}=V_{REFL}=0V$
 $T_a=-20 \sim +85^\circ C$ (普通规格产品), $T_a=-40 \sim +85^\circ C$ (大温度范围的规格产品)

| 项目 | | | 符号 | min | typ | max | 单位 | 测量条件 |
|--------------------|-------------------------------|--------------------|---------------|--------|------|-----|---------|---|
| 消耗电流 (注3) | 运行时 | 最大运行 (注4) | I_{CC} (注5) | — | — | 100 | mA | ICLK=100MHz PCLK=50MHz BCLK=25MHz |
| | | 正常运行 (注6) | | — | 35 | — | | |
| | | 因 BGO 运行 (注7) 而增加。 | | — | 15 | — | | |
| | 睡眠时 | — | | 18 | 52 | | | |
| | 全模块时钟停止时 (注8) | — | | 14 | 28 | | | |
| | 待机时 | 软件待机时 | | — | 0.08 | 3.0 | μA | |
| | | 深度软件待机时 | | RAM 保持 | — | 15 | | |
| RAM 电源停止 | | | — | 0.9 | 26 | | | |
| 模拟电源 电流 | A/D 转换中 (每个单元) | | AI_{CC} | — | 0.8 | 1.2 | mA | |
| | D/A 转换中 (每个单元) | | | — | 0.3 | 1.0 | μA | |
| | 在 A/D 转换或者 D/A 转换待机时 (全部单元) | | | — | 0.3 | 1.0 | | |
| 基准电源 电流 | A/D 转换中 (每个单元) | | | — | 0.06 | 0.1 | mA | |
| | D/A 转换中 (每个通道) | | | — | 0.4 | 0.6 | | |
| | 在 A/D 转换待机或者 D/A 转换待机时 (全部单元) | | | — | 0.3 | 1.0 | μA | |
| RAM 待机电压 | | | V_{RAM} | 2.5 | — | — | V | |
| V_{CC} 开始电压 (注9) | | | $V_{CCSTART}$ | — | — | 0.8 | V | |
| V_{CC} 上升斜率 (注9) | | | SV_{CC} | — | — | 20 | ms/V | |

注 1. 不与对应 5V 容许的端口 0 和端口 14 ~ 17 兼用。

注 2. 包括兼用引脚, 但是端口 14 ~ 17 的 RIIC 输入引脚除外。

注 3. 消耗电流值是指全部的输出引脚为无负载状态而全部的输入引脚为 $V_{IH}=V_{CC}$ 和 $V_{IL}=0V$ 并且内部上拉电阻为 OFF 状态时的值。

注 4. 外围功能为提供时钟的状态, BGO 运行除外。

注 5. 根据以下表达式, I_{CC} 取决于 $f(\text{ICLK})$ 。(ICLK:PCLK:BCLK=8:4:2)

$$I_{CC \text{ max}}=0.89 \times f + 11 \quad (\text{最大运行时})$$

$$I_{CC \text{ typ}}=0.30 \times f + 5 \quad (\text{正常运行时})$$

$$I_{CC \text{ max}}=0.41 \times f + 11 \quad (\text{睡眠时})$$

注 6. 外围功能为停止提供时钟的状态, BGO 运行除外。

注 7. 这是在程序执行过程中对 ROM 或者用于保存数据的闪存进行数据编程或者擦除时的增加。

注 8. 这是参考值。

注 9. 在接通电源时, 适用于 RES# 引脚为 Low 电平的条件。

表 29.3 输出容许电流值

条件: $V_{CC}=PLL_{V_{CC}}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL_{V_{SS}}=V_{REFL}=0V$
 $T_a=-20 \sim +85^{\circ}C$ (普通规格产品), $T_a=-40 \sim +85^{\circ}C$ (大温度范围的规格产品)

| 项目 | | 符号 | min | typ | max | 单位 |
|----------------------------|-----------|------------------|-----|-----|-----|----|
| 输出 Low 电平的容许电流 (每个引脚的平均值) | 全部输出引脚 | I_{OL} | — | — | 2.0 | mA |
| 输出 Low 电平的容许电流 (每个引脚的峰值) | 全部输出引脚 | I_{OL} | — | — | 4.0 | mA |
| 输出 Low 电平的容许电流 (总和) | 全部输出引脚的总和 | ΣI_{OL} | — | — | 80 | mA |
| 输出 High 电平的容许电流 (每个引脚的平均值) | 全部输出引脚 | $-I_{OH}$ | — | — | 2.0 | mA |
| 输出 High 电平的容许电流 (每个引脚的峰值) | 全部输出引脚 | $-I_{OH}$ | — | — | 4.0 | mA |
| 输出 High 电平的容许电流 (总和) | 全部输出引脚的总和 | $\Sigma -I_{OH}$ | — | — | 80 | mA |

【使用时的注意事项】为了确保 LSI 的可靠性，输出电流值不能超过表 29.3 中的值。

29.3 AC 特性

表 29.4 工作频率

条件: $V_{CC}=PLL_{V_{CC}}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL_{V_{SS}}=V_{REFL}=0V$
 $T_a=-20 \sim +85^\circ C$ (普通规格产品), $T_a=-40 \sim +85^\circ C$ (大温度范围的规格产品)

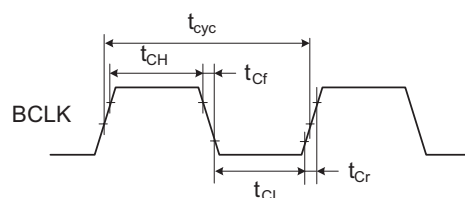
| 项目 | | 符号 | min | typ | max | 单位 |
|------|---------------|----|-----|-----|-----|-----|
| 工作频率 | 系统时钟 (ICLK) | f | 8 | — | 100 | MHz |
| | 外围模块时钟 (PCLK) | | 8 | — | 50 | |
| | 外部总线时钟 (BCLK) | | 8 | — | 25 | |

29.3.1 时钟时序

表 29.5 时钟时序

条件: $V_{CC}=PLL_{V_{CC}}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL_{V_{SS}}=V_{REFL}=0V$
 $ICLK=8 \sim 100MHz$, $BCLK=8 \sim 25MHz$, $PCLK=8 \sim 50MHz$
 $T_a=-20 \sim +85^\circ C$ (普通规格产品), $T_a=-40 \sim +85^\circ C$ (大温度范围的规格产品)

| 项目 | 符号 | min | max | 单位 | 测量条件 |
|-------------------|------------|-------|-----|----|--------|
| 时钟周期时间 | t_{cyc} | 40 | 125 | ns | 图 29.1 |
| 时钟的 High 电平脉宽 | t_{CH} | 15 | — | ns | |
| 时钟的 Low 电平脉宽 | t_{CL} | 15 | — | ns | |
| 时钟上升时间 | t_{Cr} | — | 5 | ns | |
| 时钟下降时间 | t_{Cf} | — | 5 | ns | |
| 复位振荡稳定时间 (晶体) | t_{OSC1} | 10 | — | ms | 图 29.4 |
| 软件待机振荡稳定时间 (晶体) | t_{OSC2} | 10 | — | ms | 图 29.2 |
| 深度软件待机振荡稳定时间 (晶体) | t_{OSC3} | 10 | — | ms | 图 29.3 |
| 外部时钟输出的延迟稳定时间 | t_{DEXT} | 1 | — | ms | 图 29.4 |
| 外部时钟输入的 Low 电平脉宽 | t_{EXL} | 30.71 | — | ns | 图 29.5 |
| 外部时钟输入的 High 电平脉宽 | t_{EXH} | 30.71 | — | ns | |
| 外部时钟上升时间 | t_{EXr} | — | 5 | ns | |
| 外部时钟下降时间 | t_{EXf} | — | 5 | ns | |



测量条件: $V_{OH} = V_{CC} \times 0.7$, $V_{OL} = V_{CC} \times 0.3$, $I_{OH} = -1.0mA$, $I_{OL} = 1.0mA$, $C = 30pF$

图 29.1 外部总线时钟时序

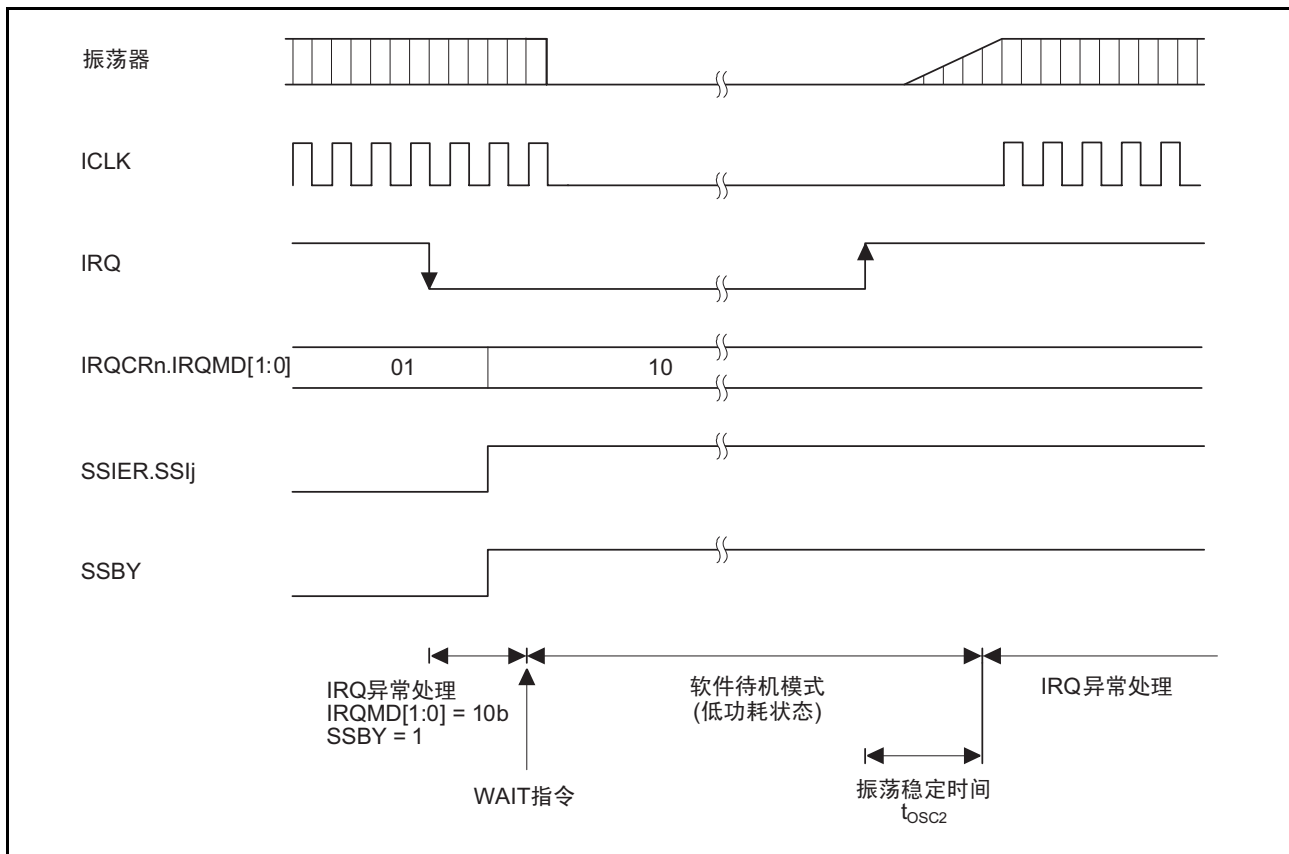


图 29.2 软件待机振荡稳定时间的时序

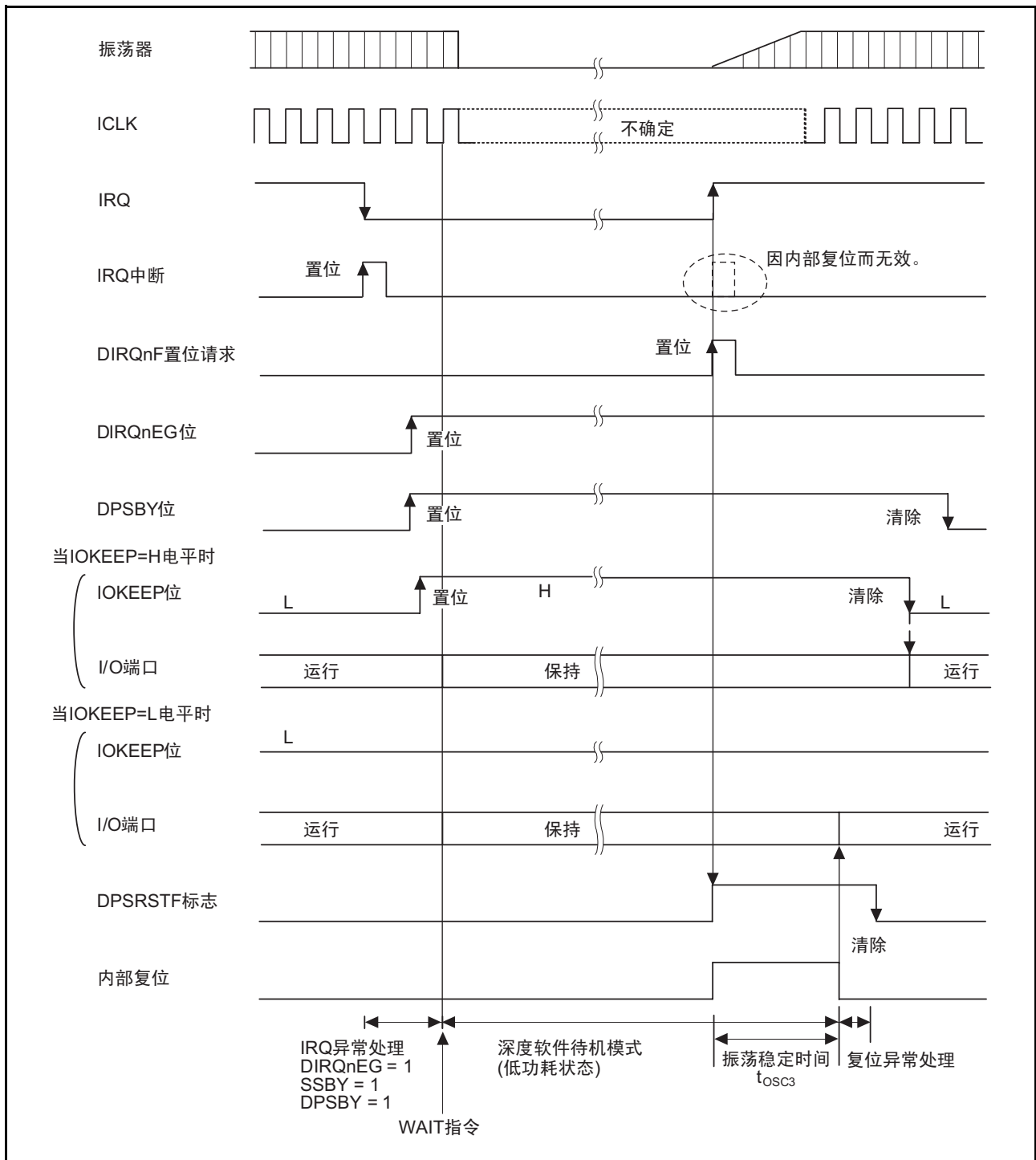


图 29.3 深度软件待机振荡稳定时间的时序

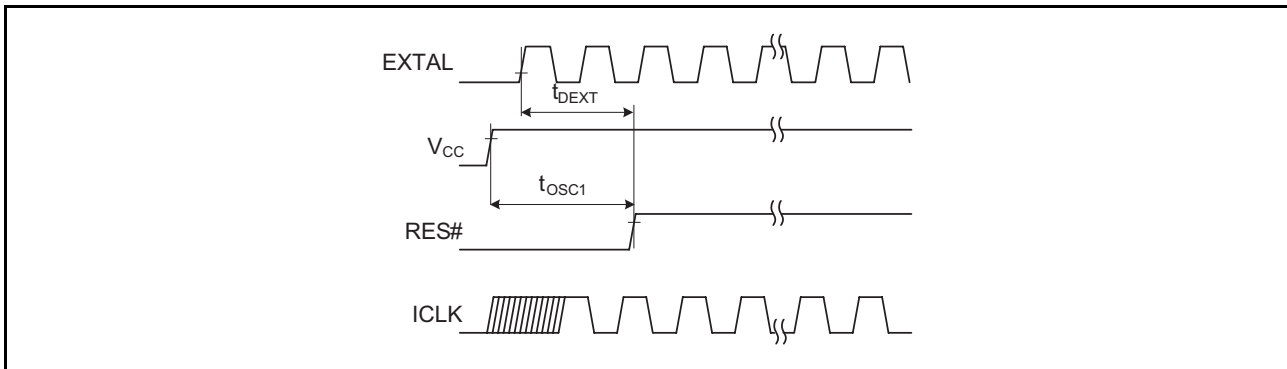


图 29.4 振荡稳定时间的时序

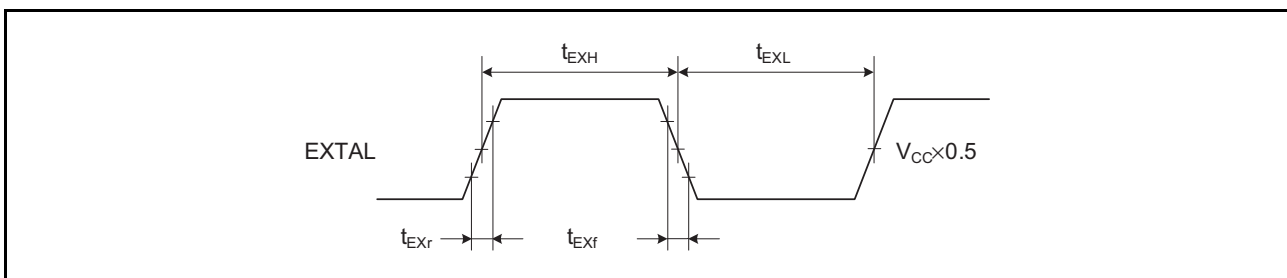


图 29.5 外部输入时钟的时序

29.3.2 控制信号的时序

表 29.6 控制信号的时序

条件: $V_{CC}=PLL_{V_{CC}}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL_{V_{SS}}=V_{REFL}=0V$
 $ICLK=8 \sim 100MHz$, $BCLK=8 \sim 25MHz$
 $T_a=-20 \sim +85^{\circ}C$ (普通规格产品), $T_a=-40 \sim +85^{\circ}C$ (大温度范围的规格产品)

| 项目 | 符号 | min | max | 单位 | 测量条件 |
|-------------------------------|------------------|-----|-----|-----------|--------|
| RES# 脉宽 (ROM 和数据闪存的编程 / 擦除除外) | t_{RESW} (注1) | 20 | — | t_{cyc} | 图 29.6 |
| | | 1.5 | — | μs | |
| 内部复位时间 (ROM 和数据闪存的编程 / 擦除) | t_{RESW2} (注2) | 35 | — | μs | |
| NMI 脉宽 | t_{NMIW} | 200 | — | ns | 图 29.7 |
| IRQ 脉宽 | t_{IRQW} | 200 | — | ns | 图 29.8 |

注 1. 需要同时满足时间和周期数的规定。

注 2. 本项目是对 FCU 复位和 WDT 复位的规定。

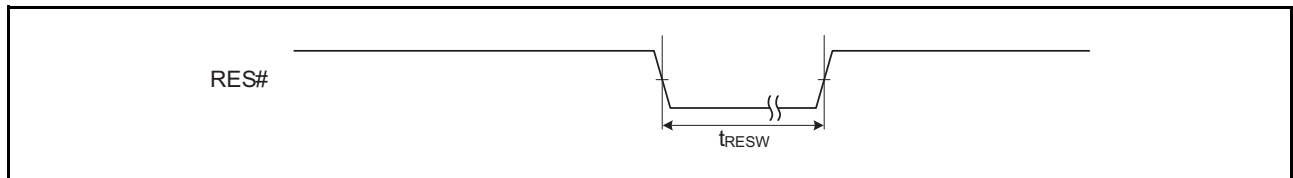


图 29.6 复位输入的时序

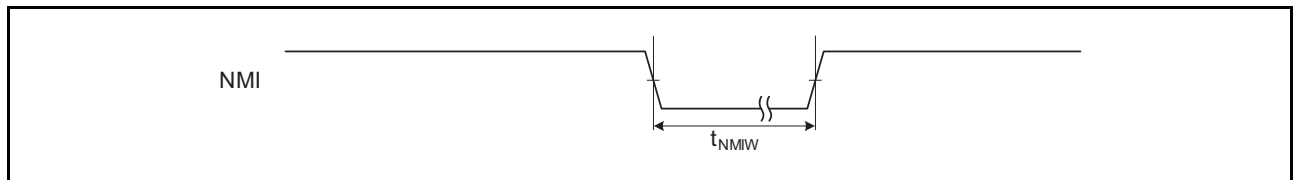


图 29.7 NMI 中断输入的时序

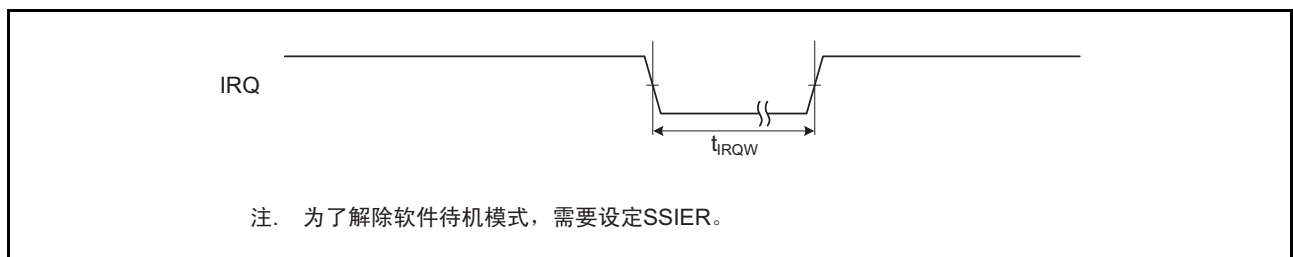


图 29.8 IRQ 中断输入的时序

29.3.3 总线时序

表 29.7 总线时序

条件: $V_{CC}=PLL V_{CC}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL V_{SS}=V_{REFL}=0V$, $BCLK=8 \sim 25MHz$
 $T_a=-20 \sim +85^{\circ}C$ (普通规格产品), $T_a=-40 \sim +85^{\circ}C$ (大温度范围的规格产品)
 输出负载条件: $V_{OH}=V_{CC} \times 0.5$, $V_{OL}=V_{CC} \times 0.5$, $I_{OH}=-1.0mA$, $I_{OL}=1.0mA$, $C=30pF$

| 项目 | 符号 | min | max | 单位 | 测量条件 |
|------------|-----------|----------------------------|-----|----|---------------------|
| 地址的延迟时间 | t_{AD} | — | 30 | ns | 图 29.9 ~ 图 29.12 |
| 字节控制的延迟时间 | t_{BCD} | — | 30 | ns | |
| CS# 延迟时间 | t_{CSD} | — | 30 | ns | |
| RD# 延迟时间 | t_{RSD} | — | 20 | ns | |
| RD# 准备时间 | t_{RSS} | $0.5 \times (1/BCLK) - 20$ | — | ns | |
| 读数据的准备时间 | t_{RDS} | 15 | — | ns | |
| 读数据的保持时间 | t_{RDH} | 0 | — | ns | |
| WR# 延迟时间 | t_{WRD} | — | 20 | ns | |
| WR# 准备时间 | t_{WRS} | $0.5 \times (1/BCLK) - 20$ | — | ns | |
| 写数据的延迟时间 | t_{WDD} | — | 35 | ns | |
| 写数据的保持时间 | t_{WDH} | 0 | — | ns | |
| WAIT# 准备时间 | t_{WTS} | 15 | — | ns | 图 29.13 |
| WAIT# 保持时间 | t_{WTH} | 0 | — | ns | |

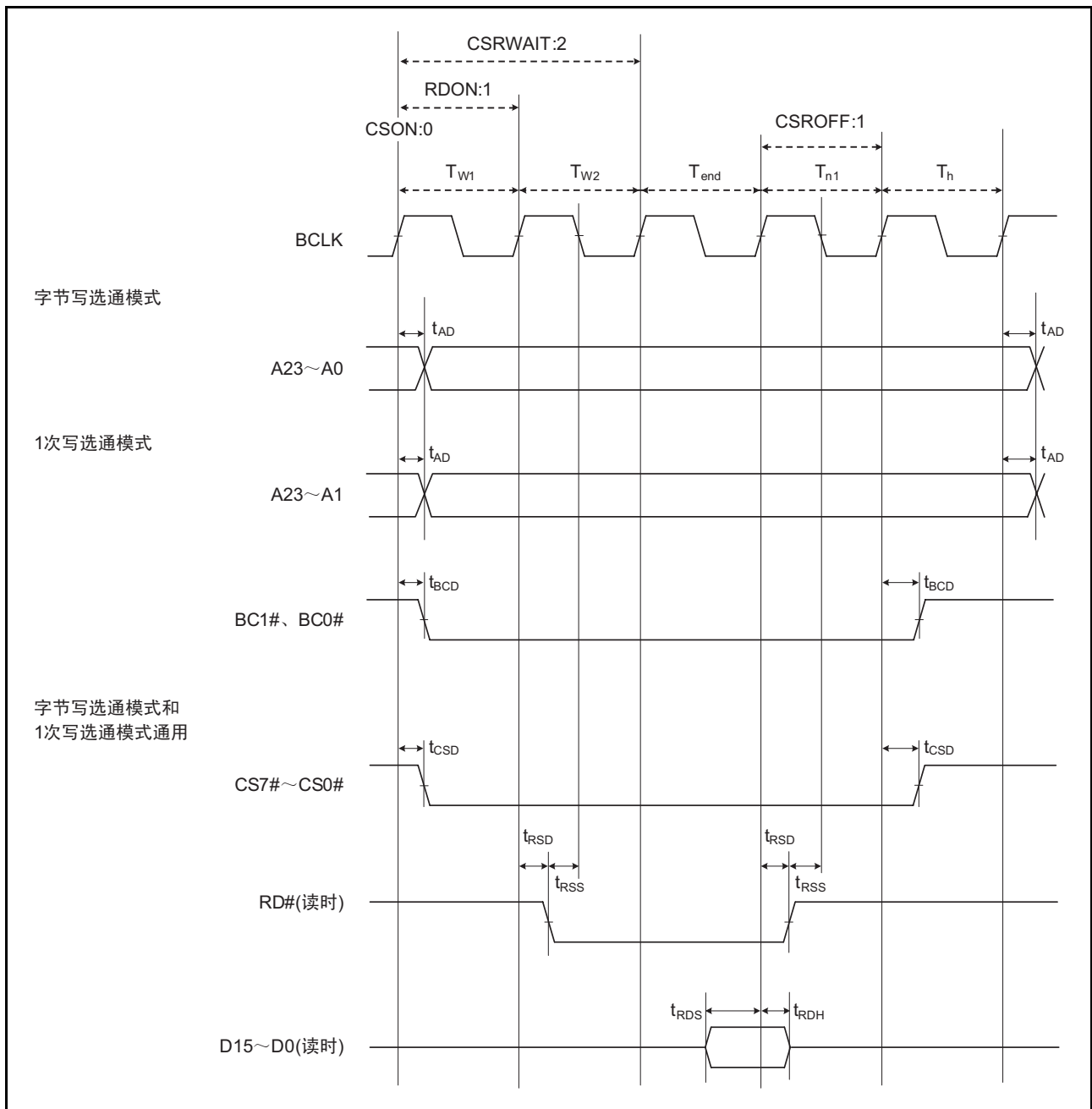


图 29.9 外部总线时序 / 正常读周期 (总线时钟同步)

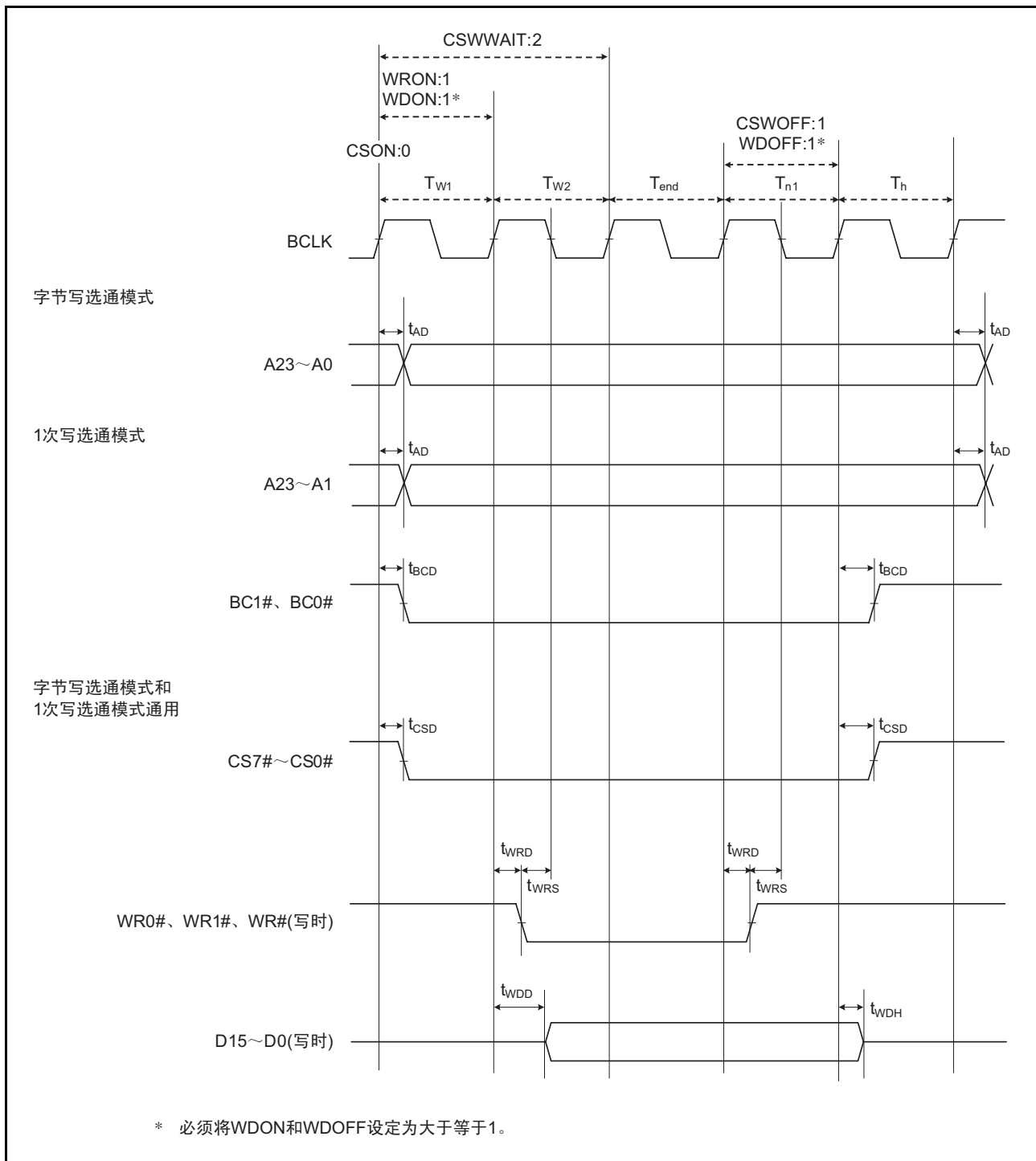


图 29.10 外部总线时序 / 正常写周期 (总线时钟同步)

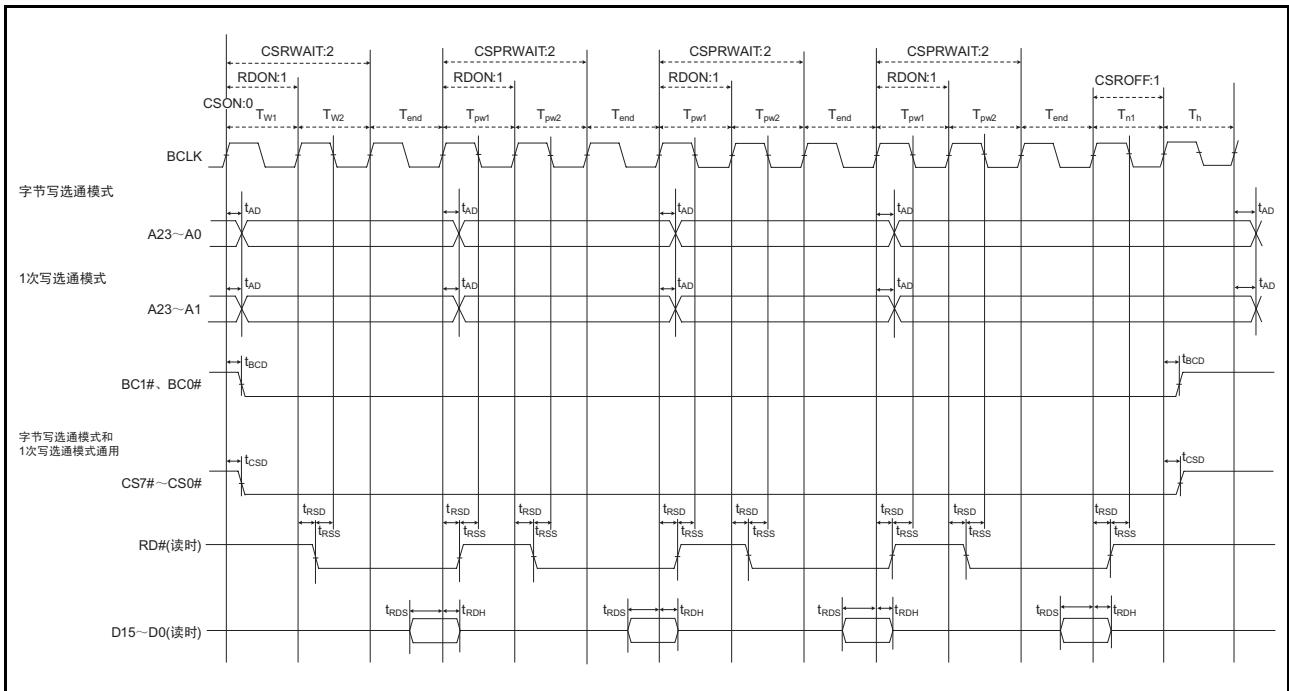


图 29.11 外部总线时序 / 页面读周期 (总线时钟同步)

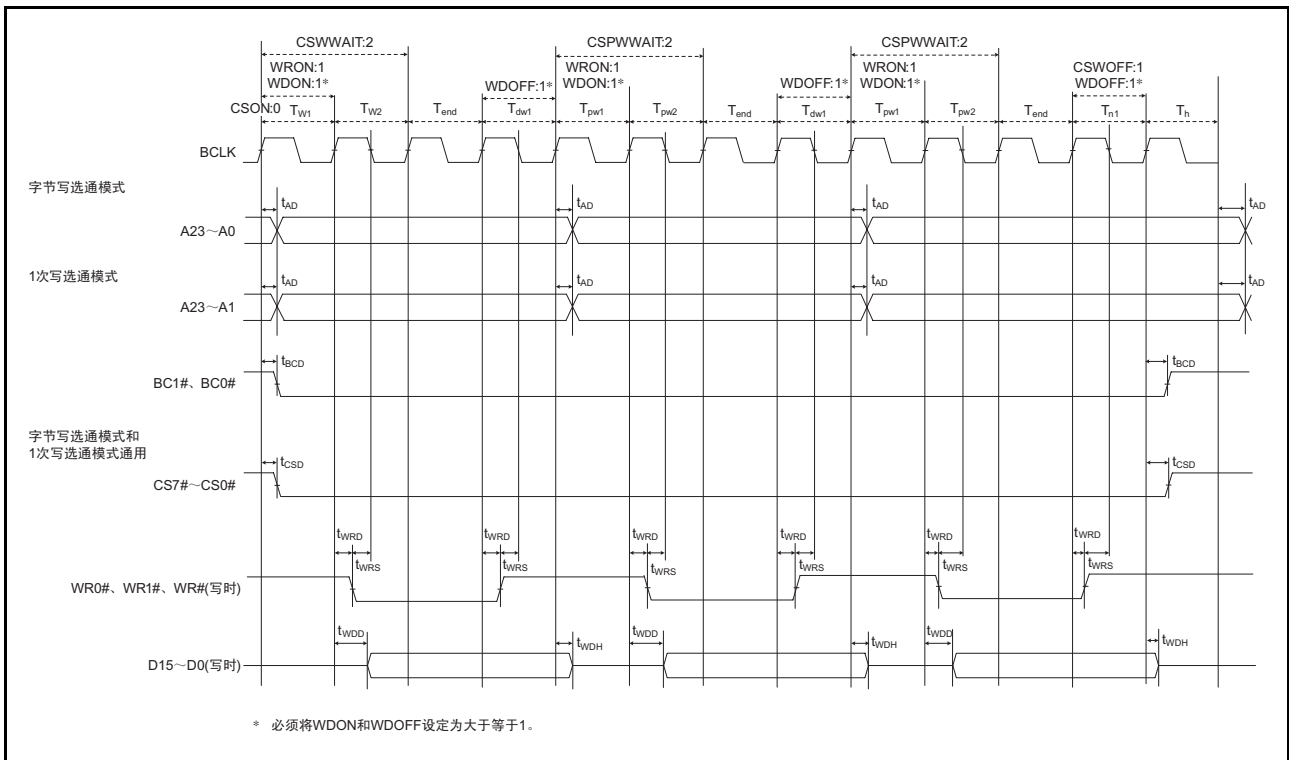


图 29.12 外部总线时序 / 页面写周期 (总线时钟同步)

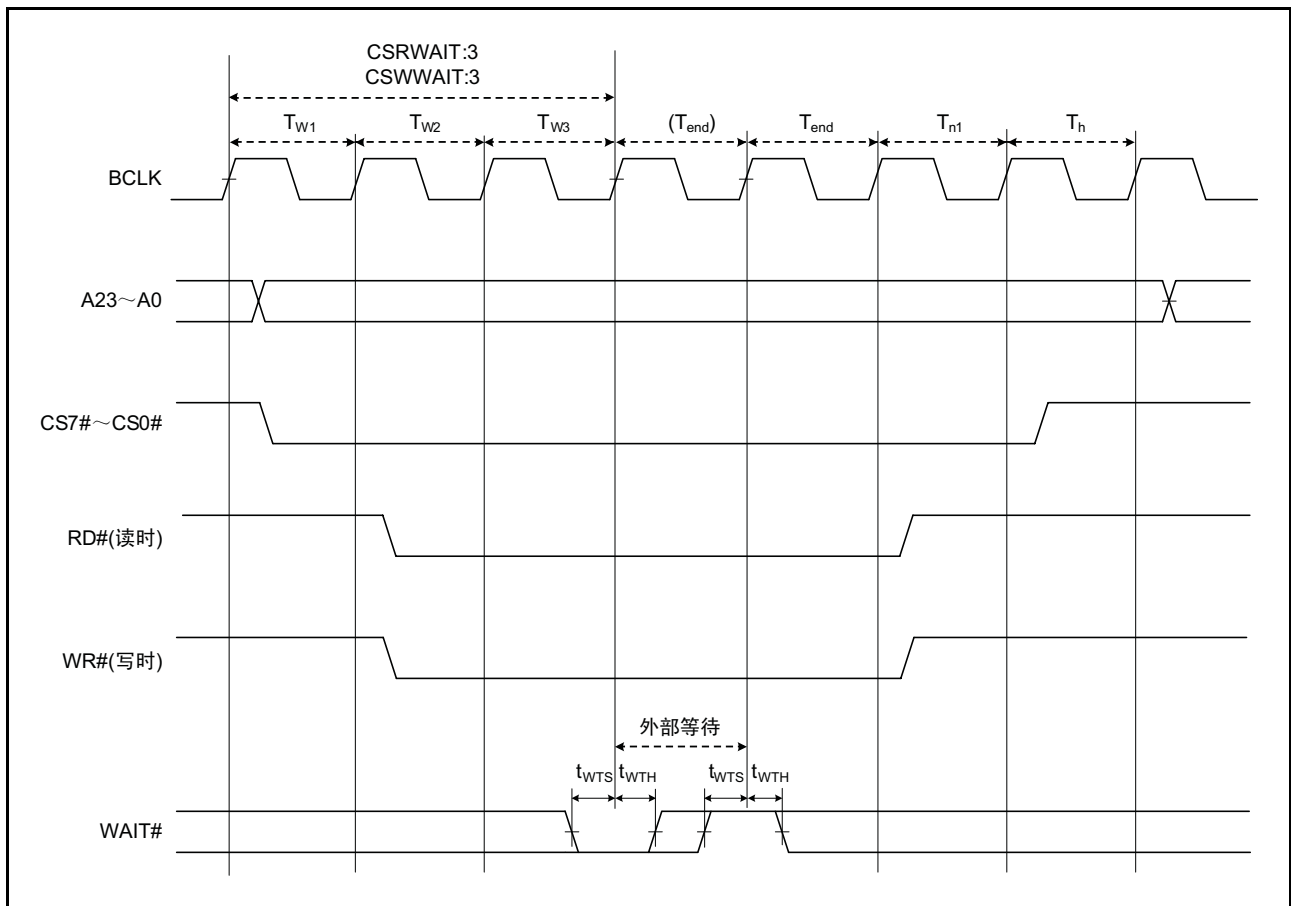


图 29.13 外部总线时序 / 外部等待控制

29.3.4 内部外围模块的时序

表 29.8 内部外围模块的时序 (1)

条件: $V_{CC}=PLL_{V_{CC}}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL_{V_{SS}}=V_{REFL}=0V$, $PCLK=8 \sim 50MHz$
 $T_a=-20 \sim +85^\circ C$ (普通规格产品), $T_a=-40 \sim +85^\circ C$ (大温度范围的规格产品)
 输出负载条件: $V_{OH}=V_{CC} \times 0.5$, $V_{OL}=V_{CC} \times 0.5$, $I_{OH}=-1.0mA$, $I_{OL}=1.0mA$, $C=30pF$

| 项目 | | 符号 | min | max | 单位 | 测量条件 | |
|------------------|------------------|-------------|-----------------------|-----------------------|----|---------|---------|
| I/O 端口 | 输出数据的延迟时间 | t_{PWD} | — | 40 | ns | 图 29.14 | |
| | 输入数据的准备时间 | t_{PRS} | 25 | — | ns | | |
| | 输入数据的保持时间 | t_{PRH} | 25 | — | ns | | |
| TPU | 定时器输出的延迟时间 | t_{TOCD} | — | 40 | ns | 图 29.15 | |
| | 定时器输入的准备时间 | t_{TICS} | 25 | — | ns | | |
| | 定时器时钟输入的准备时间 | t_{TCKS} | 25 | — | ns | 图 29.16 | |
| | 定时器时钟脉宽 | 单边沿的指定 | t_{TCKWH} | $1.5 \times (1/PCLK)$ | — | | ns |
| 双边沿的指定 | | t_{TCKWL} | $2.5 \times (1/PCLK)$ | — | ns | | |
| PPG | 脉冲输出的延迟时间 | t_{POD} | — | 40 | ns | 图 29.17 | |
| 8 位定时器 | 定时器输出延迟时间 | t_{TMOD} | — | 40 | ns | 图 29.18 | |
| | 定时器复位输入的准备时间 | t_{TMRS} | 25 | — | ns | 图 29.19 | |
| | 定时器时钟输入的准备时间 | t_{TMCS} | 25 | — | ns | 图 29.20 | |
| | 定时器时钟脉宽 | 单边沿的指定 | t_{TMCWH} | $1.5 \times (1/PCLK)$ | — | | ns |
| 双边沿的指定 | | t_{TMCWL} | $2.5 \times (1/PCLK)$ | — | ns | | |
| WDT | 上溢输出的延迟时间 | t_{WOVD} | — | 40 | ns | 图 29.21 | |
| SCI | 输入时钟周期 | 异步 | t_{Scyc} | $4 \times (1/PCLK)$ | — | ns | 图 29.22 |
| | | 时钟同步 | | $6 \times (1/PCLK)$ | — | | |
| | 输入时钟脉宽 | t_{SCKW} | $0.4 \times t_{Scyc}$ | $0.6 \times t_{Scyc}$ | ns | | |
| | 输入时钟的上升时间 | t_{SCKr} | — | 20 | ns | | |
| | 输入时钟的下降时间 | t_{SCKf} | — | 20 | ns | | |
| | 输出时钟周期 | 异步 | t_{Scyc} | $4 \times (1/PCLK)$ | — | ns | |
| | | 时钟同步 | | $6 \times (1/PCLK)$ | — | | |
| | 输出时钟脉宽 | t_{SCKW} | $0.4 \times t_{Scyc}$ | $0.6 \times t_{Scyc}$ | ns | | |
| | 输出时钟的上升时间 | t_{SCKr} | — | 20 | ns | | |
| | 输出时钟的下降时间 | t_{SCKf} | — | 20 | ns | | |
| | 发送数据的延迟时间 | t_{TXD} | — | 40 | ns | 图 29.23 | |
| | 接收数据的准备时间 (时钟同步) | t_{RXS} | 40 | — | ns | | |
| 接收数据的保持时间 (时钟同步) | t_{RXH} | 40 | — | ns | | | |
| A/D 转换器 | 触发输入的准备时间 | t_{TRGS} | 25 | — | ns | 图 29.24 | |

表 29.8 内部外围模块的时序 (2)

条件: $V_{CC}=PLL_{V_{CC}}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL_{V_{SS}}=V_{REFL}=0V$, $PCLK=8 \sim 50MHz$
 $T_a=-20 \sim +85^\circ C$ (普通规格产品), $T_a=-40 \sim +85^\circ C$ (大温度范围的规格产品)

| 项目 | | 符号 | min (注1、注2) | max | 单位 | 测量条件 |
|---|-------------------------|------------|--------------------------------|---------------------|----|---------|
| RIIC (Standard-mode) ICFER.FMPE=0 | SCL 输入的周期时间 | t_{SCL} | $8(10) \times (1/PCLK) + 1300$ | — | ns | 图 29.25 |
| | SCL 输入的 High 电平脉宽 | t_{SCLH} | $3(5) \times (1/PCLK) + 300$ | — | ns | |
| | SCL 输入的 Low 电平脉宽 | t_{SCLL} | $5 \times (1/PCLK) + 1000$ | — | ns | |
| | SCL 输入和 SDA 输入的上升时间 | t_{Sr} | — | 1000 | ns | |
| | SCL 输入和 SDA 输入的下降时间 | t_{Sf} | — | 300 | ns | |
| | SCL 输入和 SDA 输入尖峰脉冲的消除时间 | t_{SP} | 0 | $4 \times (1/PCLK)$ | ns | |
| | SDA 输入的总线空闲时间 | t_{BUF} | $5 \times (1/PCLK) + 1000$ | — | ns | |
| | 开始条件输入的保持时间 | t_{STAH} | $3(5) \times (1/PCLK) + 300$ | — | ns | |
| | 重新开始条件输入的准备时间 | t_{STAS} | $5 \times (1/PCLK) + 1000$ | — | ns | |
| | 停止条件输入的准备时间 | t_{STOS} | $3(5) \times (1/PCLK) + 300$ | — | ns | |
| | 数据输入的准备时间 | t_{SDAS} | 250 | — | ns | |
| | 数据输入的保持时间 | t_{SDAH} | 0 | — | ns | |
| | SCL 和 SDA 的电容性负载 | C_b | — | 400 | pF | |
| RIIC (Fast-mode) ICFER.FMPE=0 | SCL 输入的周期时间 | t_{SCL} | $8(10) \times (1/PCLK) + 600$ | — | ns | |
| | SCL 输入的 High 电平脉宽 | t_{SCLH} | $3(5) \times (1/PCLK) + 300$ | — | ns | |
| | SCL 输入的 Low 电平脉宽 | t_{SCLL} | $5 \times (1/PCLK) + 300$ | — | ns | |
| | SCL 输入和 SDA 输入的上升时间 | t_{Sr} | $20 + 0.1C_b$ | 300 | ns | |
| | SCL 输入和 SDA 输入的下降时间 | t_{Sf} | $20 + 0.1C_b$ | 300 | ns | |
| | SCL 输入和 SDA 输入尖峰脉冲的消除时间 | t_{SP} | 0 | $4 \times (1/PCLK)$ | ns | |
| | SDA 输入的总线空闲时间 | t_{BUF} | $5 \times (1/PCLK) + 300$ | — | ns | |
| | 开始条件输入的保持时间 | t_{STAH} | $3(5) \times (1/PCLK) + 300$ | — | ns | |
| | 重新开始条件输入的准备时间 | t_{STAS} | $5 \times (1/PCLK) + 300$ | — | ns | |
| | 停止条件输入的准备时间 | t_{STOS} | $3(5) \times (1/PCLK) + 300$ | — | ns | |
| | 数据输入的准备时间 | t_{SDAS} | 100 | — | ns | |
| | 数据输入的保持时间 | t_{SDAH} | 0 | — | ns | |
| | SCL 和 SDA 的电容性负载 | C_b | — | 400 | pF | |
| RIIC (Fast-mode+) ICFER.FMPE=1 | SCL 输入的周期时间 | t_{SCL} | $8(10) \times (1/PCLK) + 240$ | — | ns | |
| | SCL 输入的 High 电平脉宽 | t_{SCLH} | $3(5) \times (1/PCLK) + 120$ | — | ns | |
| | SCL 输入的 Low 电平脉宽 | t_{SCLL} | $5 \times (1/PCLK) + 120$ | — | ns | |
| | SCL 输入和 SDA 输入的上升时间 | t_{Sr} | — | 120 | ns | |
| | SCL 输入和 SDA 输入的下降时间 | t_{Sf} | — | 120 | ns | |
| | SCL 和 SDA 输入尖峰脉冲的擦除时间 | t_{SP} | 0 | $4 \times (1/PCLK)$ | ns | |
| | SDA 输入的总线空闲时间 | t_{BUF} | $5 \times (1/PCLK) + 120$ | — | ns | |
| | 开始条件输入的保持时间 | t_{STAH} | $3(5) \times (1/PCLK) + 120$ | — | ns | |
| | 重新开始条件输入的准备时间 | t_{STAS} | $5 \times (1/PCLK) + 120$ | — | ns | |
| | 停止条件输入的准备时间 | t_{STOS} | $3(5) \times (1/PCLK) + 120$ | — | ns | |
| | 数据输入的准备时间 | t_{SDAS} | 50 | — | ns | |
| | 数据输入的保持时间 | t_{SDAH} | 0 | — | ns | |
| | SCL 和 SDA 的电容性负载 | C_b | — | 550 | pF | |

注 1. () 内的数值表示在 ICFER.NFE 为 “1” 并且将数字滤波器置为有效的状态下 ICMR3.NF[1:0] 为 “11b” 的情况。

注 2. C_b 是总线的电容总和。

表 29.8 内部外围模块的时序 (3)

条件: $V_{CC}=PLL_{VCC}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL_{VSS}=V_{REFL}=0V$,
 $T_a=-20 \sim +85^{\circ}C$ (普通规格产品), $T_a=-40 \sim +85^{\circ}C$ (大温度范围的规格产品)

| 项目 | 符号 | min (注1、注2) | max | 单位 | 测量条件 | |
|------------------------|-------------------|--------------|-----|------|---------|---------|
| 边界扫描 (176 引脚 LFBGA) | TCK 时钟的周期时间 | t_{TCKcyc} | 100 | — | ns | 图 29.26 |
| | TCK 时钟的 High 电平脉宽 | t_{TCKH} | 45 | — | ns | |
| | TCK 时钟的 Low 电平脉宽 | t_{TCKL} | 45 | — | ns | |
| | TCK 时钟的上升时间 | t_{TCKr} | — | 5 | ns | |
| | TCK 时钟的下降时间 | t_{TCKf} | — | 5 | ns | |
| TRST# 脉宽 | t_{TRSTW} | 20 | — | Tcyc | 图 29.27 | |
| TMS 准备时间 | t_{TMSS} | 20 | — | ns | 图 29.28 | |
| TMS 保持时间 | t_{TMSh} | 20 | — | ns | | |
| TDI 准备时间 | t_{TDis} | 20 | — | ns | | |
| TDI 保持时间 | t_{TDIH} | 20 | — | ns | | |
| TDO 数据的延迟时间 | t_{TDOD} | — | 40 | ns | | |

注 1. () 内的数值表示在 ICFER.NFE 为 “1” 并且将数字滤波器置为有效的状态下 ICMR3.NF[1:0] 为 “11b” 的情况。

注 2. C_b 是总线的电容总和。

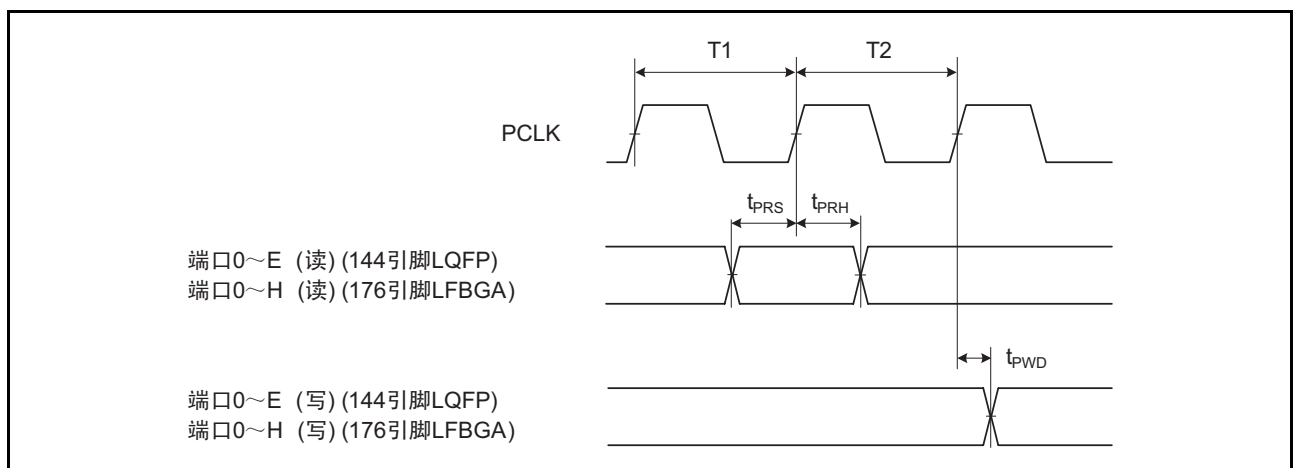


图 29.14 I/O 端口输入 / 输出的时序

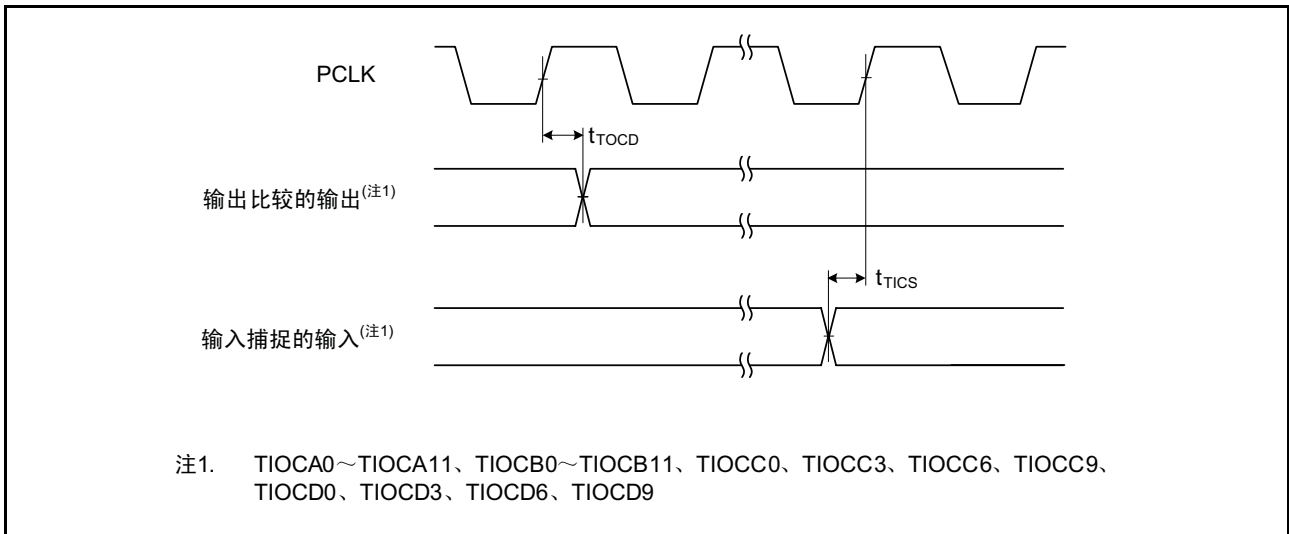


图 29.15 TPU 输入 / 输出的时序

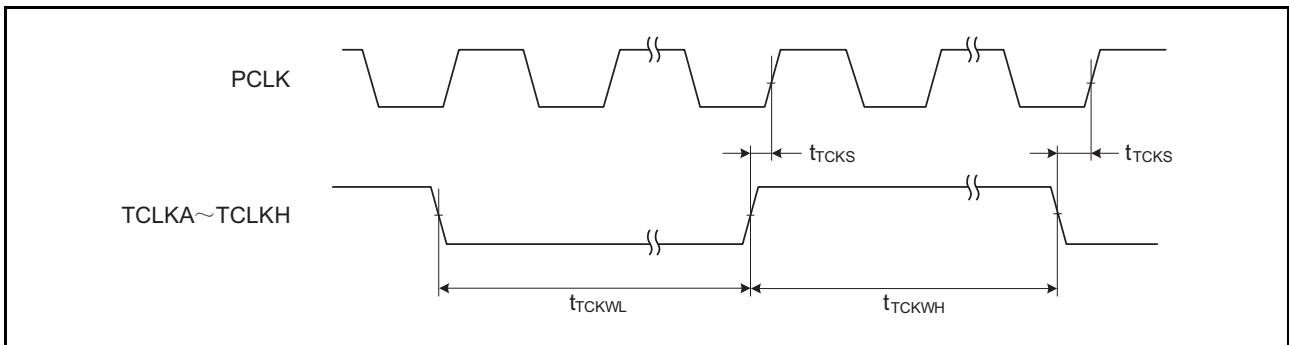


图 29.16 TPU 时钟输入的时序

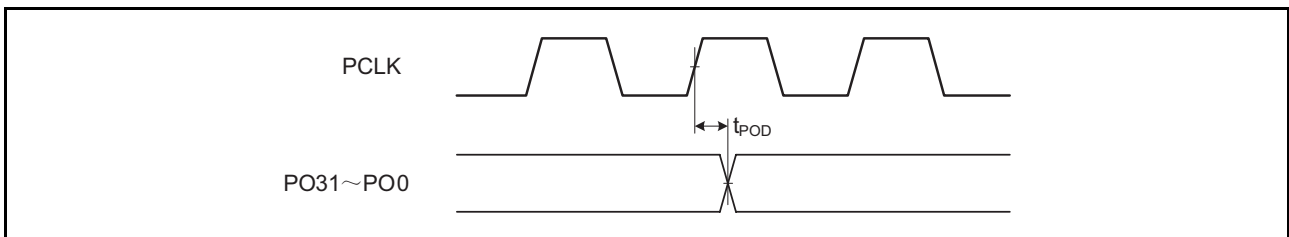


图 29.17 PPG 输出的时序

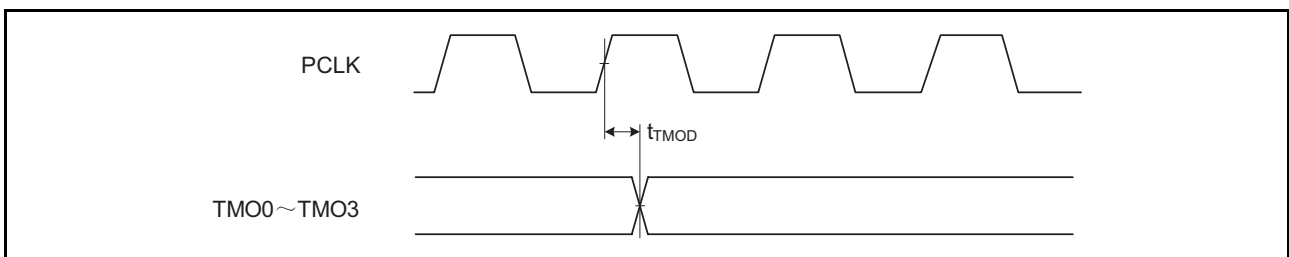


图 29.18 8 位定时器输出的时序

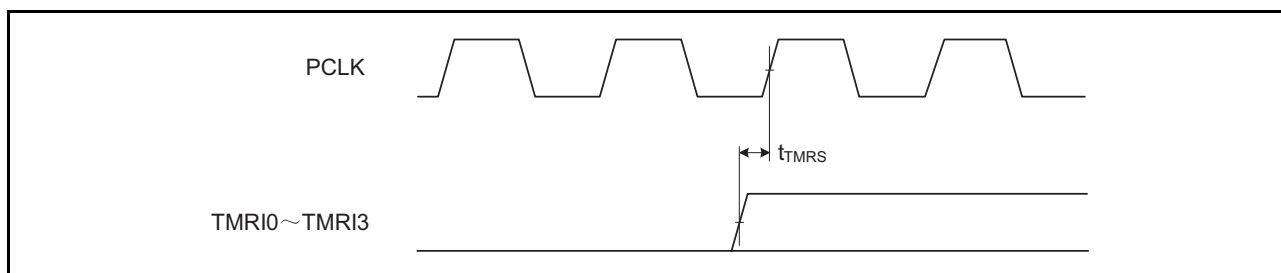


图 29.19 8 位定时器复位输入的时序

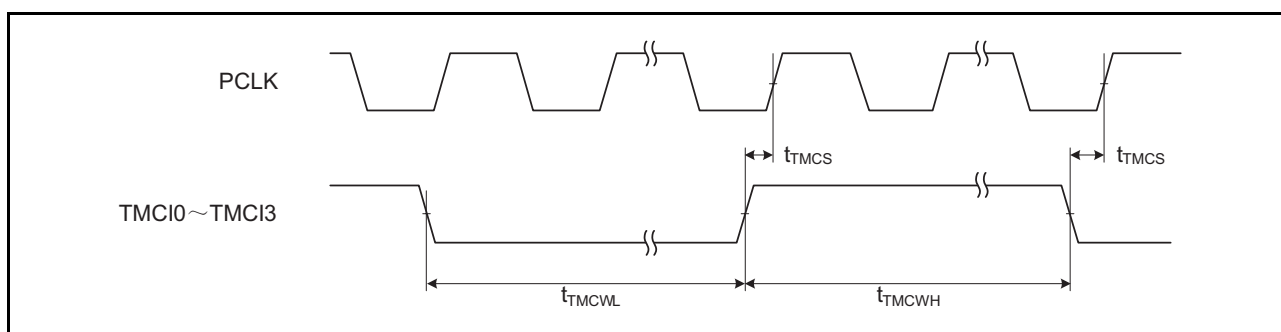


图 29.20 8 位定时器时钟输入的时序

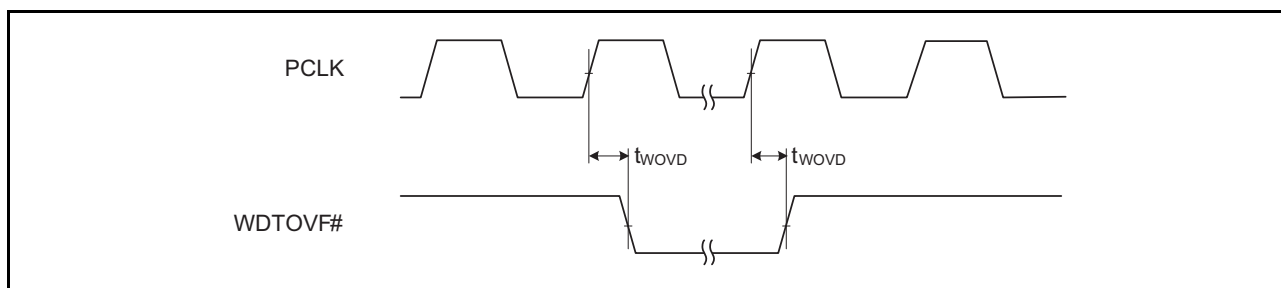


图 29.21 WDT 输出的时序

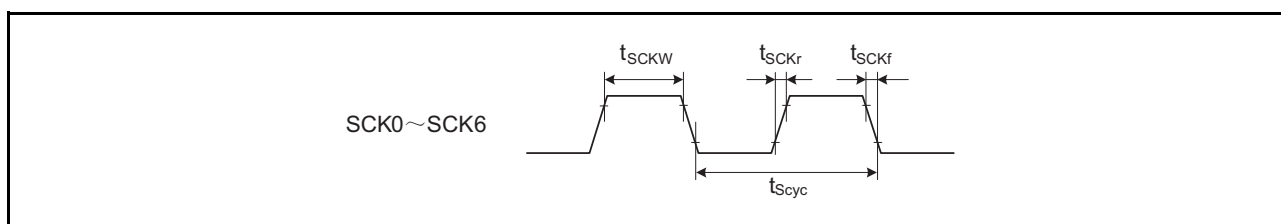


图 29.22 SCK 时钟输入的时序

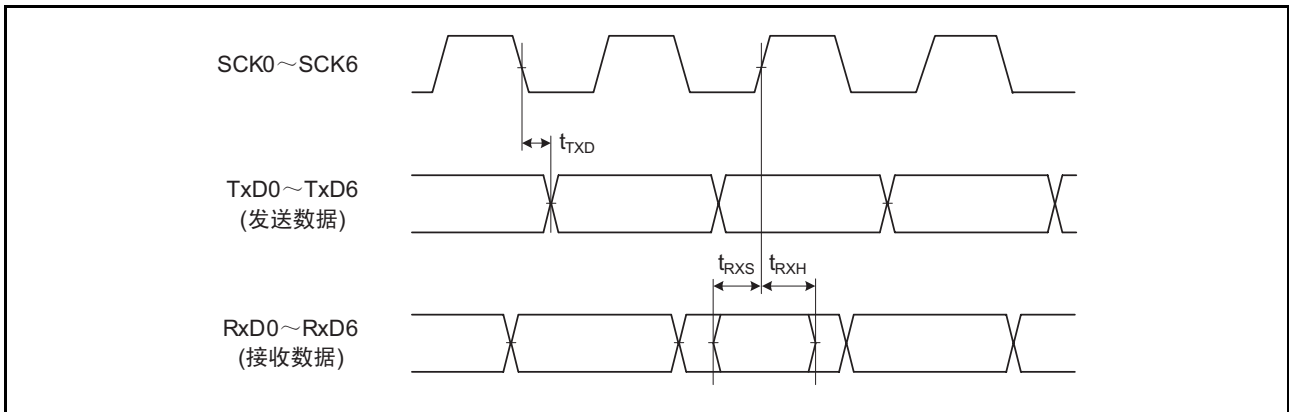


图 29.23 SCI 输入 / 输出的时序 / 时钟同步模式

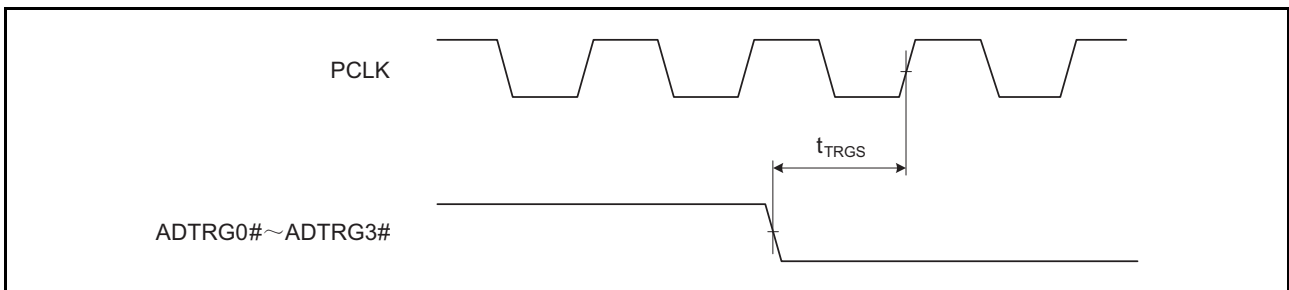


图 29.24 A/D 转换器外部触发输入的时序

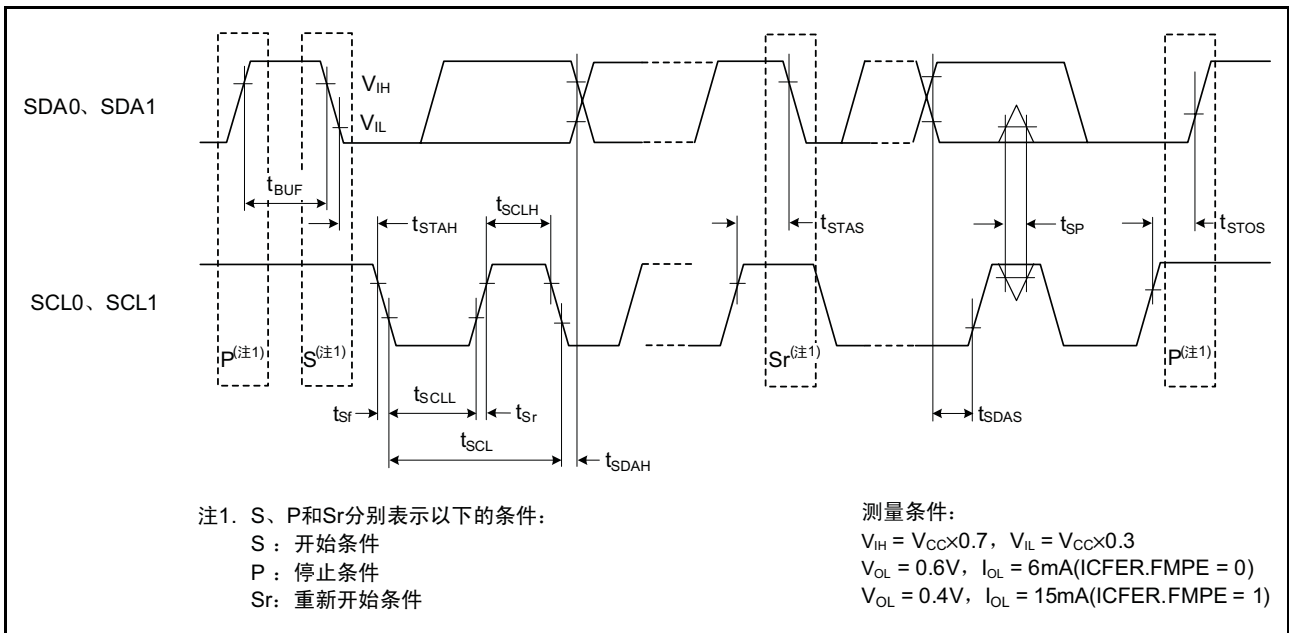


图 29.25 I²C 总线接口输入 / 输出的时序

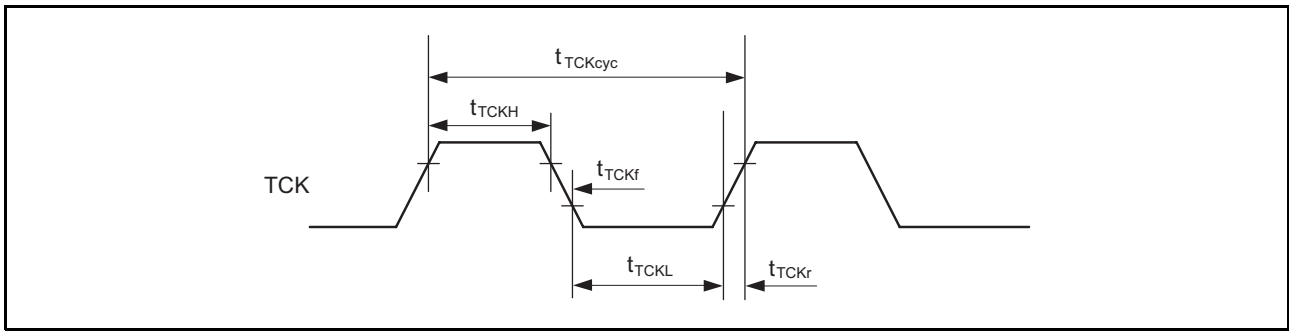


图 29.26 边界扫描 TCK 的时序

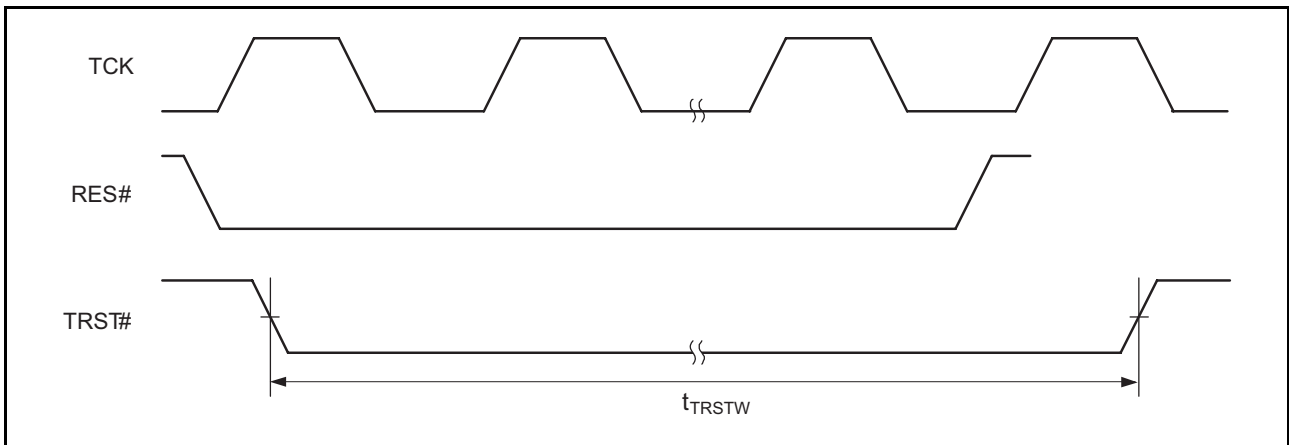


图 29.27 边界扫描 TRST# 的时序

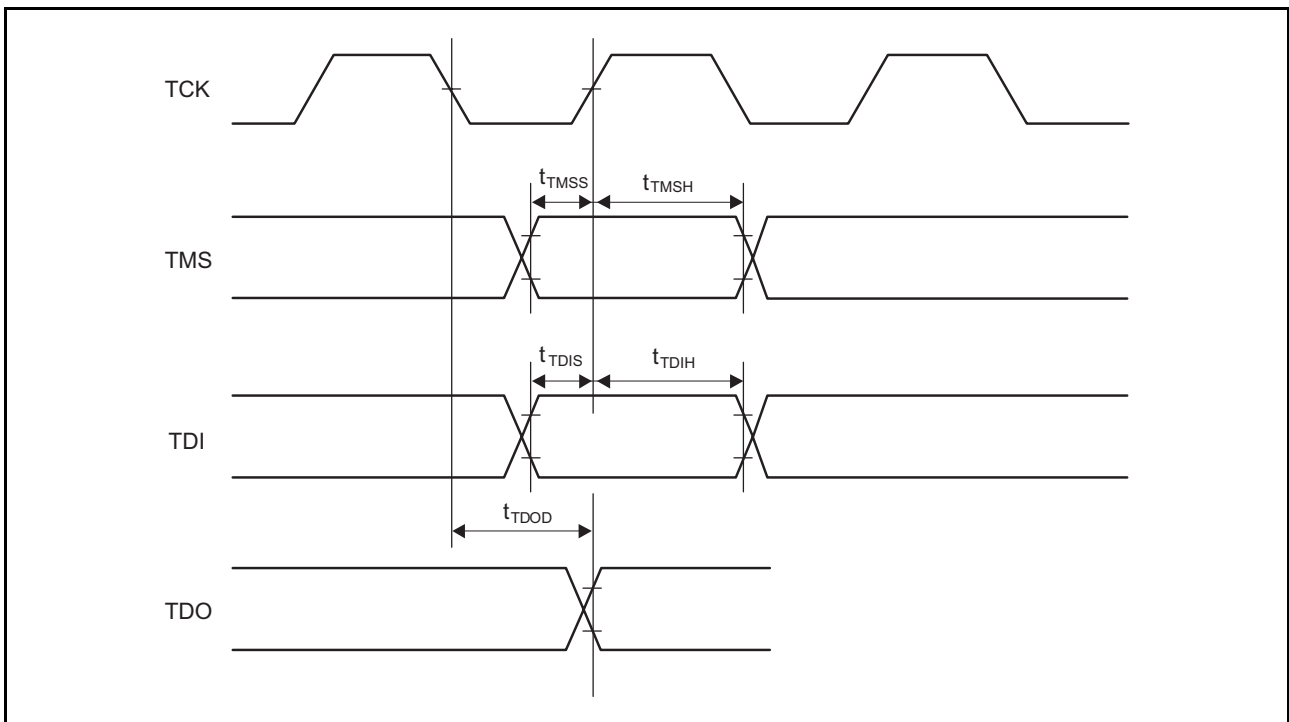


图 29.28 边界扫描输入 / 输出的时序

29.4 A/D 转换特性

表 29.9 A/D 转换特性

条件: $V_{CC}=PLL_{VCC}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL_{VSS}=V_{REFL}=0V$, $PCLK=8 \sim 50MHz$
 $ADCLK=4 \sim 50MHz$, $T_a=-20+85^{\circ}C$ (普通规格产品), $T_a=-40 \sim +85^{\circ}C$ (大温度范围的规格产品)

| 项目 | | | min | typ | max | 单位 | 测量条件 |
|-------------------------------------|------------------------|------------------------------|-------------------|-----------|-----------|---------|-----------------|
| 分辨率 | | | 10 | 10 | 10 | 位 | |
| 转换时间 (注1) (ADCLK= 50MHz 时) | 有外接电容器 0.1 μ F。 | 充分地给电容器 充完电时 (注2) | 0.8 (0.3) (注3) | — | — | μ s | 采样的 15 个 状态 |
| | 没有外接电容器。 | 容许信号源阻抗 max=1.0k Ω | 1.0 (0.5) (注3) | — | — | | 采样的 25 个 状态 |
| | | 容许信号源阻抗 max=5.0k Ω | 2.6 (2.1) (注3) | — | — | | 采样的 105 个 状态 |
| 模拟输入的电容 | | | — | — | 6.0 | pF | |
| 积分非线性误差 (INL) | | | — | ± 1.5 | ± 3.0 | LSB | |
| 偏移误差 | | | — | ± 1.5 | ± 3.0 | LSB | |
| 满刻度误差 | | | — | ± 1.5 | ± 3.0 | LSB | |
| 量化误差 | | | — | ± 0.5 | — | LSB | |
| 绝对精度 | | | — | ± 1.5 | ± 3.0 | LSB | |
| 微分非线性误差 (DNL) | | | — | ± 0.5 | ± 1.0 | LSB | |

注 1. 转换时间是采样时间和比较时间的总和, 各项的测量条件表示采样的状态数。

注 2. 不对应扫描。

注 3. () 表示采样时间。

29.5 D/A 转换特性

表 29.10 D/A 转换特性

条件: $V_{CC}=PLL_{VCC}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL_{VSS}=V_{REFL}=0V$, $PCLK=8 \sim 50MHz$
 $T_a=-20 \sim +85^{\circ}C$ (普通规格产品), $T_a=-40 \sim +85^{\circ}C$ (大温度范围的规格产品)

| 项 目 | min | typ | max | 单位 | 测量条件 |
|---------|-----|-----------|-----------|------------|-------------------|
| 分辨率 | 10 | 10 | 10 | 位 | |
| 转换时间 | — | — | 3 | μ s | 负载电容 20pF |
| 绝对精度 | — | ± 2.0 | ± 4.0 | LSB | 负载电阻 2M Ω |
| | — | — | ± 3.0 | LSB | 负载电阻 4M Ω |
| | — | — | ± 2.0 | LSB | 负载电阻 10M Ω |
| RO 输出电阻 | — | 3.6 | — | k Ω | |

29.6 ROM（保存代码的闪存）特性

表 29.11 ROM（保存代码的闪存）特性

条件: $V_{CC}=PLL_{VCC}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL_{VSS}=V_{REFL}=0V$ 编程 / 擦除时的工作范围: $T_a=-20 \sim +85^{\circ}C$ (普通规格产品), $T_a=-40 \sim +85^{\circ}C$ (大温度范围的规格产品)

| 项目 | 符号 | min | typ | max | 单位 | 测量条件 | |
|-------------------------------|-------------|--------------|-----|-----|---------|---------------------------|--------------------------------------|
| 编程时间 | 256B | t_{P256} | — | 2 | 12 | ms | PCLK=50MHz 当 $N_{PEC} \leq 100$ 时 |
| | 8KB | t_{P8K} | — | 45 | 100 | ms | |
| | 256B | t_{P256} | — | 2.4 | 14.4 | ms | PCLK=50MHz 当 $N_{PEC} > 100$ 时 |
| | 8KB | t_{P8K} | — | 54 | 120 | ms | |
| 擦除时间 | 8KB | t_{E8K} | — | 50 | 120 | ms | PCLK=50MHz 当 $N_{PEC} \leq 100$ 时 |
| | 64KB | t_{E64K} | — | 400 | 875 | ms | |
| | 128KB | t_{E128K} | — | 800 | 1750 | ms | |
| | 8KB | t_{E8K} | — | 60 | 144 | ms | PCLK=50MHz 当 $N_{PEC} > 100$ 时 |
| | 64KB | t_{E64K} | — | 480 | 1050 | ms | |
| | 128KB | t_{E128K} | — | 960 | 2100 | ms | |
| 再编程 / 擦除周期 (注1) | N_{PEC} | 1000 (注2) | — | — | — | 次 | |
| 编程时的挂起延迟时间 | t_{SPD} | — | — | 120 | μs | 图 29.29 当 PCLK=50MHz 时 | |
| 擦除时的第 1 次挂起延迟时间 (在挂起优先模式中) | t_{SESD1} | — | — | 120 | μs | | |
| 擦除时的第 2 次挂起延迟时间 (在挂起优先模式中) | t_{SESD2} | — | — | 1.7 | ms | | |
| 擦除时的挂起延迟时间 (在擦除优先模式中) | t_{SEED} | — | — | 1.7 | ms | | |
| 数据的保持时间 (注3) | t_{DRP} | 10 | — | — | 年 | | |

注 1. 再编程 / 擦除周期的定义:

再编程 / 擦除周期是指每块的擦除次数。

如果再编程 / 擦除周期为 n ($n=1000$), 就能逐块擦除 n 次。

例如, 对于 8KB 的块, 如果将 256B 的数据分 32 次写到不同的地址后擦除该块, 再编程 / 擦除次数就计为 1。但是, 对于 1 次的擦除, 不能对相同的地址进行多次编程 (禁止重写)。

注 2. 这是保证改写后的全部特性的 min 次数 (保证范围是 1 ~ min 值)。

注 3. 这是在包括 min 值的规格范围内进行改写时的特性。

29.7 数据闪存（保存数据的闪存）特性

表 29.12 数据闪存（保存数据的闪存）特性

条件: $V_{CC}=PLL_{V_{CC}}=AV_{CC}=3.0 \sim 3.6V$, $V_{REFH}=3.0V \sim AV_{CC}$, $V_{SS}=PLL_{V_{SS}}=V_{REFL}=0V$ 编程 / 擦除时的工作范围: $T_a=-20 \sim +85^{\circ}C$ (普通规格产品), $T_a=-40 \sim +85^{\circ}C$ (大温度范围的规格产品)

| 项目 | | 符号 | min | typ | max | 单位 | 测量条件 |
|-------------------------------|------|--------------|----------------|-----|-----|---------|---------------------------|
| 编程时间 | 8B | t_{DP8} | — | 0.4 | 2 | ms | 当 PCLK=50MHz 时 |
| | 128B | t_{DP128} | — | 1 | 5 | ms | |
| 擦除时间 | 8KB | t_{DE8K} | — | 300 | 900 | ms | 当 PCLK=50MHz 时 |
| 空白检查时间 | 8B | t_{DBC8} | — | — | 30 | μs | 当 PCLK=50MHz 时 |
| | 8KB | t_{DBC8K} | — | — | 2.5 | ms | |
| 再编程 / 擦除周期 (注 1) | | N_{DPEC} | 30000 (注 2) | — | — | 次 | |
| 编程时的挂起延迟时间 | | t_{DSPD} | — | — | 120 | μs | 图 29.29 当 PCLK=50MHz 时 |
| 擦除时的第 1 次挂起延迟时间 (在挂起优先模式中) | | t_{DSESD1} | — | — | 120 | μs | |
| 擦除时的第 2 次挂起延迟时间 (在挂起优先模式中) | | t_{DSESD2} | — | — | 1.7 | ms | |
| 擦除时的挂起延迟时间 (在擦除优先模式中) | | t_{DSEED} | — | — | 1.7 | ms | |
| 数据的保持时间 (注 3) | | t_{DDRP} | 10 | — | — | 年 | |

注 1. 再编程 / 擦除周期的定义:

再编程 / 擦除周期是指每块的擦除次数。

如果再编程 / 擦除周期为 n 次 ($n=30000$), 就能逐块擦除 n 次。

例如, 对于 8KB 的块, 如果将 128B 的数据分 64 次写到不同的地址后擦除该块, 再编程 / 擦除次数就计为 1。但是, 对于 1 次的擦除, 不能对相同的地址进行多次编程 (禁止重写)。

注 2. 这是保证改写后的全部特性的 min 次数。(保证范围是 1 ~ min 值)

注 3. 这是在包括 min 值的规格范围内进行改写时的特性。

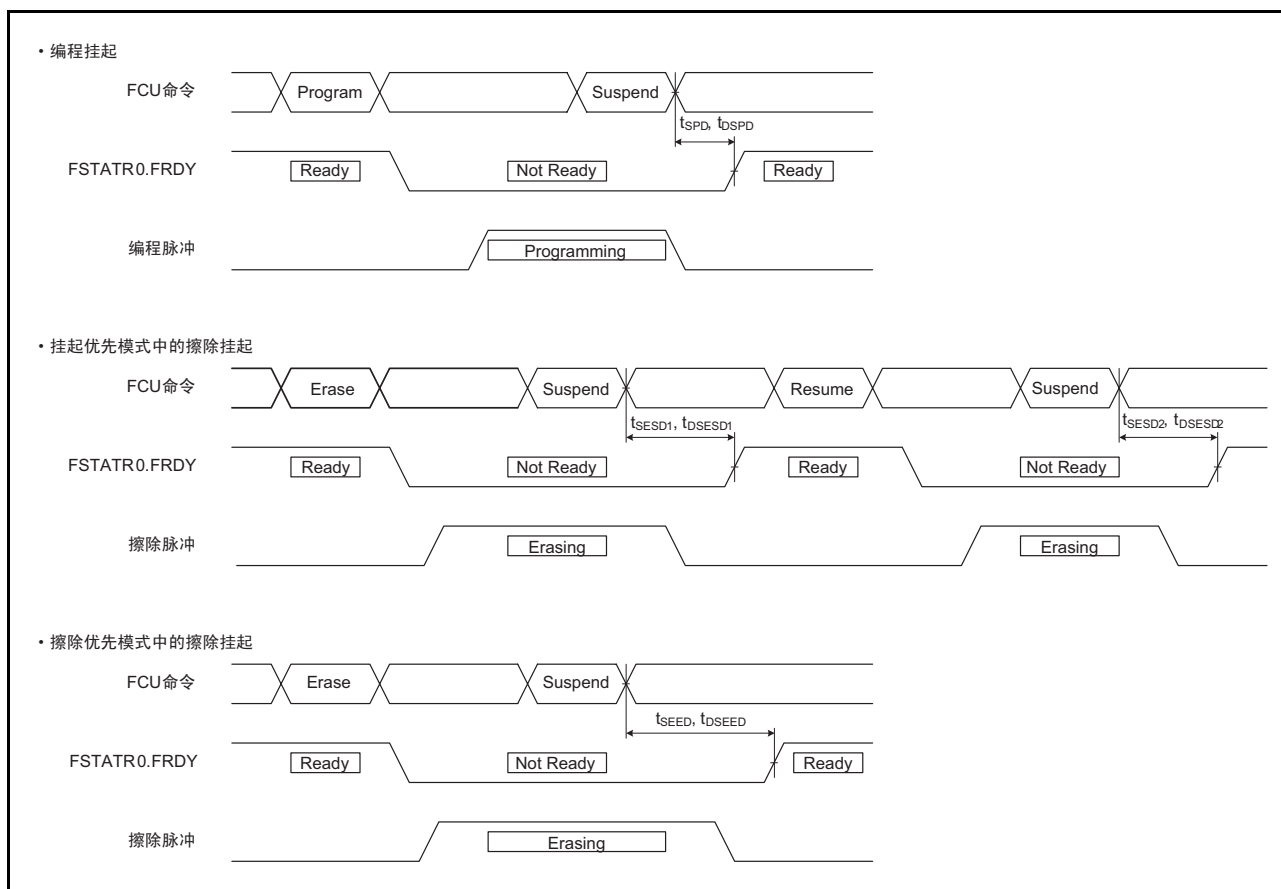


图 29.29 ROM 和数据闪存的编程 / 擦除挂起时序

附录

附录 1. 各运行模式中的端口状态

附表 1.1 运行模式中的端口状态 (1/3)

| 端口名 引脚名 | 寄存器设定的运行模式 | 复位 | 软件待机模式 | | 深度软件待机模式 IOKEEP=1/0 | 解除深度软件待机模式后 (返回到启运模式) | |
|--|-----------------------------------|-----|--|--|------------------------|--------------------------|----------|
| | | | OPE=1 | OPE=0 | | IOKEEP=1* | IOKEEP=0 |
| 端口 0 | 全部模式 | HiZ | Keep-O* ¹ | | Keep | Keep | HiZ |
| 端口 1 | 全部模式 | HiZ | Keep-O* ¹ | | Keep | Keep | HiZ |
| 端口 2 | 全部模式 | HiZ | Keep-O | | Keep | Keep | HiZ |
| P30 ~ P33 | 全部模式 | HiZ | Keep-O* ¹ | | Keep-O* ² | Keep | HiZ |
| P34/PO12/ TIOCA1/IRQ4-A | 全部模式 | HiZ | Keep-O* ¹ | | Keep | Keep | HiZ |
| P35 ~ P37 | 全部模式 | HiZ | Keep-O | | Keep | Keep | HiZ |
| 端口 4 | 全部模式 | HiZ | Keep-O* ¹ | | Keep | Keep | HiZ |
| P50/WR0#/WR# | 单芯片模式 (EXBE=0) | HiZ | Keep-O | | Keep | Keep | HiZ |
| | 内部 ROM 有效 / 无效扩展模式 (EXBE=1) | | [输出 WR0#/ WR# 时] H | [输出 WR0#/ WR# 时] HiZ | | | |
| P51/WR1#/ BC1# | 单芯片模式 (EXBE=0) | HiZ | Keep-O | | Keep | Keep | HiZ |
| | 内部 ROM 有效 / 无效扩展模式 (EXBE=1) | | [输出 WR1#/ BC1# 时] H | [输出 WR1#/ BC1# 时] HiZ | | | |
| | [上述以外] Keep-O | | | | | | |
| P52/RD# | 单芯片模式 (EXBE=0) | HiZ | Keep-O | | Keep | Keep | HiZ |
| | 内部 ROM 有效 / 无效扩展模式 (EXBE=1) | | [输出 RD# 时] H | [输出 RD# 时] HiZ | | | |
| P53/BCLK | 全部模式 | HiZ | [输出时钟时] H [上述以外] HiZ | | Keep | Keep | HiZ |
| P54/TRDATA0 | 全部模式 | HiZ | Keep-O | | Keep | Keep | HiZ |
| P55/TRDATA1 | 全部模式 | HiZ | Keep-O | | Keep | Keep | HiZ |
| P56/TRDATA2 | 全部模式 | HiZ | Keep-O | | Keep | Keep | HiZ |
| P57/TRDATA3/ WAIT# | 全部模式 | HiZ | Keep-O | | Keep | Keep | HiZ |
| P60/CS0#/ CS4#-A/CS5#-B | 全部模式 | HiZ | [输出 CS 时] H [上述以外] Keep-O | [输出 CS 时] HiZ [上述以外] Keep-O | Keep | Keep | HiZ |
| P61/CS1#/ CS2#-B/CS5#-A/ CS6#-B/CS7#-B | 全部模式 | HiZ | [输出 CS 时] H [上述以外] Keep-O | [输出 CS 时] HiZ [上述以外] Keep-O | Keep | Keep | HiZ |

附表 1.1 运行模式中的端口状态 (2/3)

| 端口名 引脚名 | 寄存器设定的运行模式 | 复位 | 软件待机模式 | | 深度软件待机模式 IOKEEP=1/0 | 解除深度软件待机模式后 (返回到启运模式) | |
|---|-----------------------------------|-----|--|--------------------------------------|---|--------------------------|----------|
| | | | OPE=1 | OPE=0 | | IOKEEP=1* | IOKEEP=0 |
| P62/CS2#-A/ CS6#-A | 全部模式 | HiZ | [输出 CS 时] H [上述以外] Keep-O | [输出 CS 时] HiZ [上述以外] Keep-O | Keep | Keep | HiZ |
| P63/CS3#-A/ CS7#-A | 全部模式 | HiZ | [输出 CS 时] H [上述以外] Keep-O | [输出 CS 时] HiZ [上述以外] Keep-O | Keep | Keep | HiZ |
| P64/CS4#-B | 全部模式 | HiZ | [输出 CS 时] H [上述以外] Keep-O | [输出 CS 时] HiZ [上述以外] Keep-O | Keep | Keep | HiZ |
| P65/IRQ15-A | 全部模式 | HiZ | Keep-O*1 | | Keep | Keep | HiZ |
| P66/DA0 | 全部模式 | HiZ | [DAOE0=1] 保持 DA 输出 [DAOE0=0] Keep-O | | [DAOE0=1] HiZ [DAOE0=0] Keep | Keep | HiZ |
| P67/DA1 | 全部模式 | HiZ | [DAOE1=1] 保持 DA 输出 [DAOE1=0] Keep-O | | [DAOE1=1] HiZ [DAOE1=0] Keep-O | Keep | HiZ |
| P70/CS3#-B/ ADTRG2# | 全部模式 | HiZ | [输出 CS 时] H [上述以外] Keep-O | [输出 CS 时] HiZ [上述以外] Keep-O | Keep | Keep | HiZ |
| P71/CS4#-C/ CS5#-C/CS6#-C/ CS7#-C | 全部模式 | HiZ | [输出 CS 时] H [上述以外] Keep-O | [输出 CS 时] HiZ [上述以外] Keep-O | Keep | Keep | HiZ |
| P72 ~ P75 | 全部模式 | HiZ | Keep-O | | Keep | Keep | HiZ |
| P76/IRQ14-A | 全部模式 | HiZ | Keep-O*1 | | Keep | Keep | HiZ |
| P77 | 全部模式 | HiZ | Keep-O | | Keep | Keep | HiZ |
| 端口 8 | 全部模式 | HiZ | Keep-O | | Keep | Keep | HiZ |
| 端口 9 | 全部模式 | HiZ | Keep-O | | Keep | Keep | HiZ |
| 端口 A | 单芯片模式 (EXBE=0) | HiZ | Keep-O | | Keep | Keep | HiZ |
| | 内部 ROM 有效 / 无效扩展模式 (EXBE=1) | | [输出地址时] 保持地址输出 [上述以外] HiZ | [输出地址时] HiZ [上述以外] HiZ | | | |
| 端口 B | 单芯片模式 (EXBE=0) | HiZ | Keep-O | | Keep | Keep | HiZ |
| | 内部 ROM 有效扩展模式 | | [输出地址时] 保持地址输出 [上述以外] Keep-O | [输出地址时] HiZ [上述以外] Keep-O | | | |

附表 1.1 运行模式中的端口状态 (3/3)

| 端口名 引脚名 | 寄存器设定的运行模式 | | 复位 | 软件待机模式 | | 深度软件待机模式 IOKEEP=1/0 | 解除深度软件待机模式后 (返回到启动模式) | |
|------------|-----------------------------------|------------------|------------------|---|--|------------------------|--------------------------|----------|
| | | | | OPE=1 | OPE=0 | | IOKEEP=1* | IOKEEP=0 |
| PC0 ~ PC4 | 单芯片模式 (EXBE=0) | | HiZ | Keep-O | | Keep | Keep | HiZ |
| | 内部 ROM 有效 / 无效扩展模式 (EXBE=1) | | | [输出地址时] 保持地址输出 [上述以外] Keep-O | [输出地址时] HiZ [上述以外] Keep-O | | | |
| PC5 ~ PC7 | 单芯片模式 (EXBE=0) | | HiZ | Keep-O | | Keep | Keep | HiZ |
| | 内部 ROM 有效 / 无效扩展模式 (EXBE=1) | | | [输出地址时] 保持地址输出 [输出 CS 时] H [上述以外] Keep-O | [输出地址时] HiZ [输出 CS 时] HiZ [上述以外] Keep-O | | | |
| 端口 D | 单芯片模式 (EXBE=0) | | HiZ | Keep-O | | Keep | Keep | HiZ |
| | 内部 ROM 有效 / 无效扩展模式 (EXBE=1) | | | HiZ | | | | |
| PE0 ~ PE4 | 单芯片模式 (EXBE=0) | | HiZ | Keep-O | | Keep | Keep | HiZ |
| | 内部 ROM 有效 / 无效扩展模式 (EXBE=1) | 总线宽 度 8 位 | | Keep-O | | | | |
| | | 总线宽 度 16 位 | | HiZ | | | | |
| PE5 ~ PE7 | 单芯片模式 (EXBE=0) | | HiZ | Keep-O* ¹ | | Keep | Keep | HiZ |
| | 内部 ROM 有效 / 无效扩展模式 (EXBE=1) | 总线宽 度 8 位 | | Keep-O* ¹ | | | | |
| | | 总线宽 度 16 位 | | HiZ | | | | |
| 端口 F | 全部模式 | | HiZ | Keep-O | | Keep | Keep | HiZ |
| 端口 G | 全部模式 | | HiZ | Keep-O | | Keep | Keep | HiZ |
| 端口 H | 全部模式 | | HiZ | Keep-O | | Keep | Keep | HiZ |
| WDTOVF | 全部模式 | | 输出 WDTOV F | H | | H | H | |

H : High 电平

L : Low 电平

Keep-O : 当用作输出引脚时, 保持以前的值; 当用作输入引脚时, 为高阻抗。

*1: 当用作外部中断引脚时, 能在设定为软件待机解除请求的情况下进行输入。

*2: 能在设定为深度软件待机解除请求的情况下进行输入。

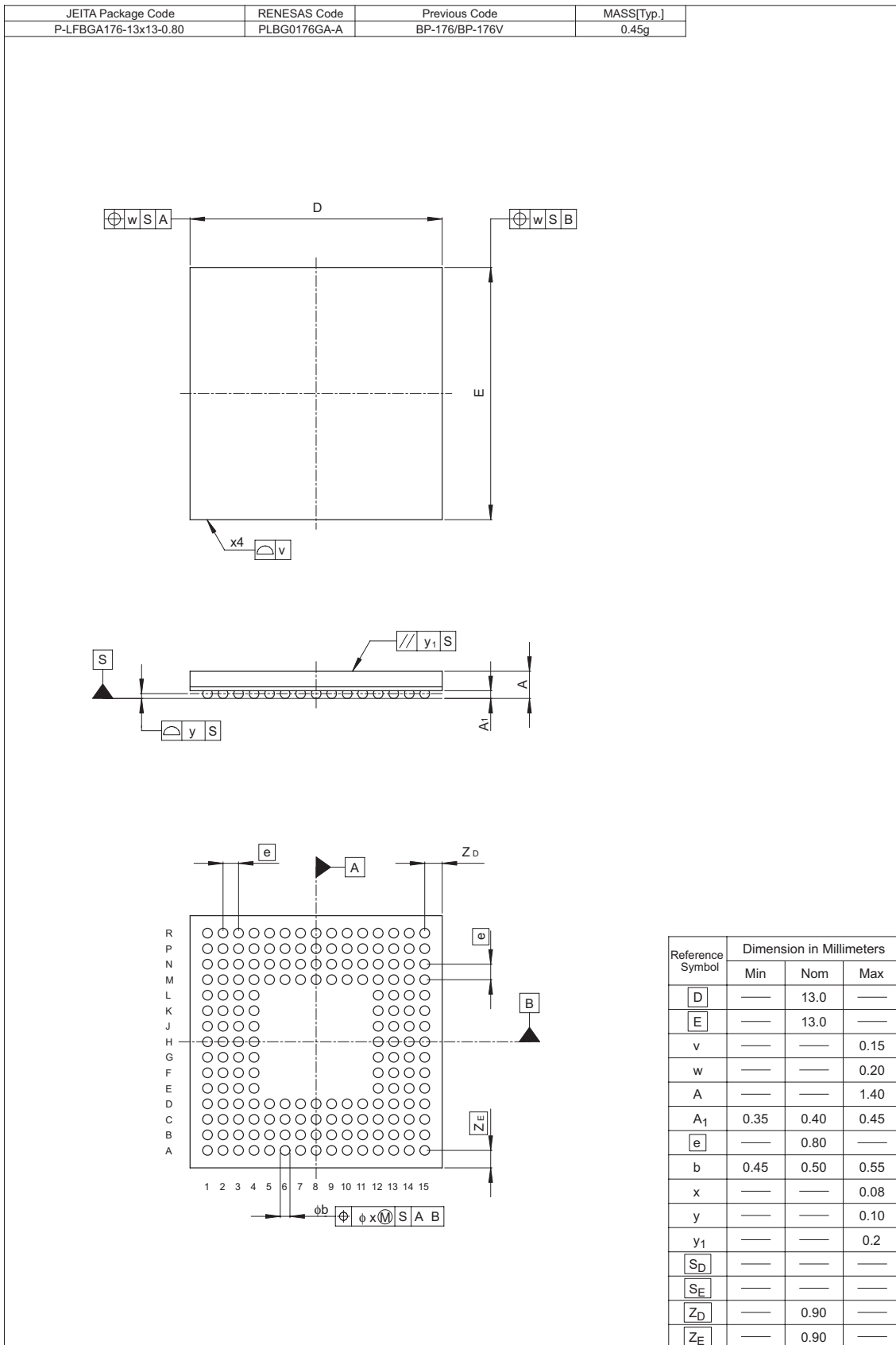
Keep : 保持软件待机中的引脚状态。

HiZ : 高阻抗

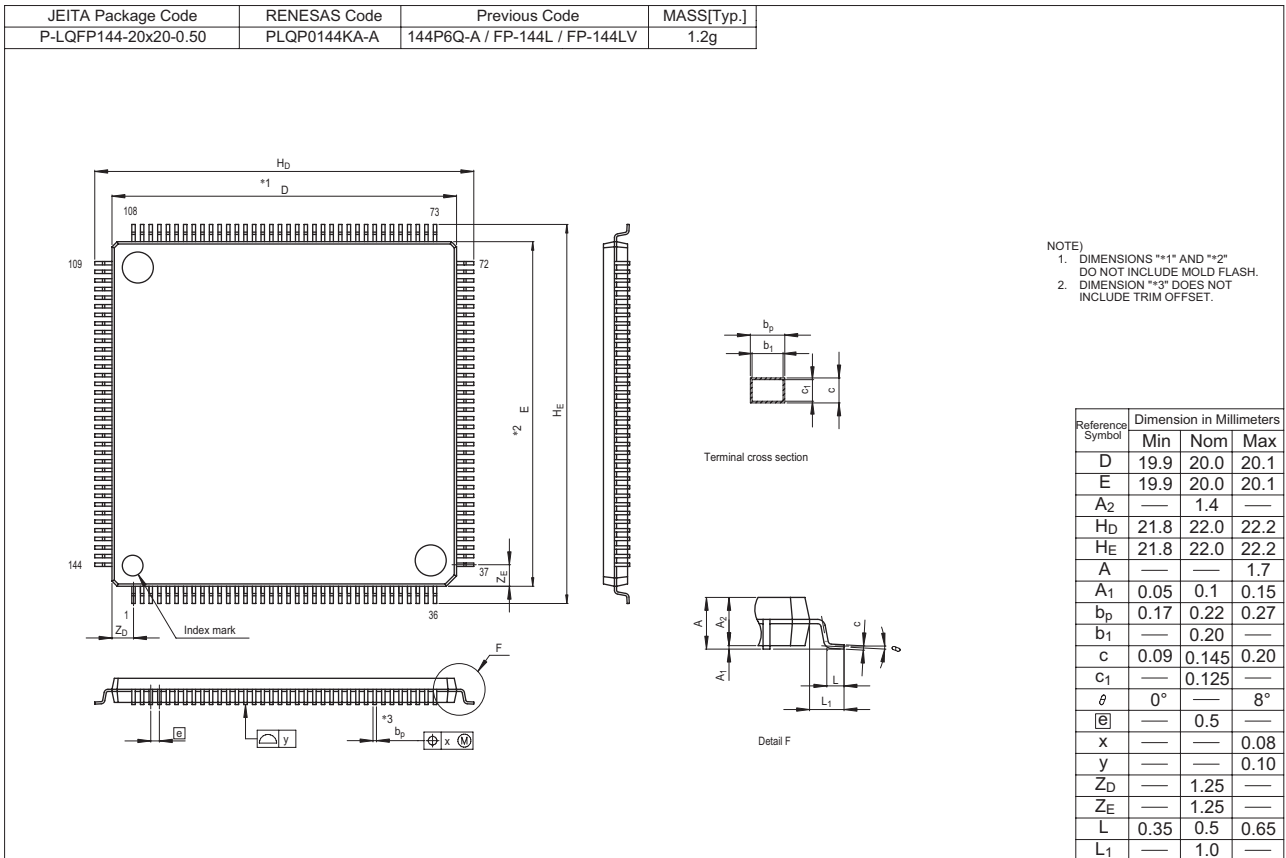
IOKEEP=1* : 在将 DPSBYCR.IOKEEP 位置 “0” 前, 保持 I/O 端口的状态。

附录 2. 封装尺寸图

有关封装尺寸图的最新版和安装的信息，刊登在瑞萨电子的主页的“封装”中。



附图 2.1 176 引脚 LFBGA (PLBG0176GA-A) 封装尺寸图



附图 2.2 144 引脚 LQFP (PLQP0144KA-A)

| | |
|------|------------------|
| 修订记录 | RX610 群 用户手册 硬件篇 |
|------|------------------|

| Rev. | 发行日 | 修订内容 | |
|------|------------|------|------|
| | | 页 | 修订处 |
| 1.00 | 2010.09.15 | — | 初版发行 |

RX610 群
用户手册 硬件篇

Publication Date: Rev.1.00 Sep 15, 2010

Published by: Renesas Electronics Corporation

**SALES OFFICES**

Renesas Electronics Corporation

<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics 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: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RX610群



瑞萨电子株式会社

R01UH0032CJ0100