

RX63T 群

用户手册 硬件篇

瑞萨 32 位单片机
RX 族 / RX600 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. 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.
2. 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.
3. 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.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
"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 etc.
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). 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 for which it is not intended. 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 which the product is not intended by Renesas Electronics.
6. 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.
7. 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 systems manufactured by you.
8. 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.
9. 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. You should not use Renesas Electronics products or 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. When exporting the Renesas Electronics 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.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
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.

注意事项

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，瑞萨电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，瑞萨电子已尽量做到合理注意，但是，瑞萨电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，瑞萨电子不承担任何责任。
3. 对于因使用本文档中的瑞萨电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，瑞萨电子不承担任何责任。本文档所记载的内容不应视为对瑞萨电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 用户不得更改、修改、复制或或以其他方式部分或全部地非法使用瑞萨电子的任何产品。对于用户或第三方因上述更改、修改、复制或以其他方式非法使用瑞萨电子产品的行为而遭受的任何损失，瑞萨电子不承担任何责任。
5. 瑞萨电子产品根据其质量等级分为两个等级：“标准等级”和“高质量等级”。每种瑞萨电子产品的推荐用途均取决于产品的质量等级，如下所示：
标准等级： 计算机、办公设备、通讯设备、测试和测量设备、视听设备、家用电器、机械工具、个人电子设备以及工业机器人等。
高质量等级： 运输设备（汽车、火车、轮船等）、交通控制系统、防灾系统、预防犯罪系统以及安全设备等。
瑞萨电子产品无意用于且未被授权用于可能对人类生命造成直接威胁的产品或系统及可能造成人身伤害的产品或系统（人工生命维持装置或系统、植埋于体内的装置等）中，或者可能造成重大财产损失的产品或系统（核反应堆控制系统、军用设备等）中。在将每种瑞萨电子产品用于某种特定应用之前，用户应先确认其质量等级。不得将瑞萨电子产品用于超出其设计用途之外的任何应用。对于用户或第三方因将瑞萨电子产品用于其设计用途之外而遭受的任何损害或损失，瑞萨电子不承担任何责任。
6. 使用本文档中记载的瑞萨电子产品时，应在瑞萨电子指定的范围内，特别是在最大额定值、电源工作电压范围、移动电源电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用瑞萨电子产品而产生的故障或损失，瑞萨电子不承担任何责任。
7. 虽然瑞萨电子一直致力于提高瑞萨电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，瑞萨电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当瑞萨电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。由于难于对微机软件单独进行评估，所以请用户自行对最终产品或系统进行安全评估。
8. 关于环境保护方面的详细内容，例如每种瑞萨电子产品的环境兼容性等，请与瑞萨电子的营业部门联系。使用瑞萨电子产品时，请遵守对管制物质的使用或含量进行管理的所有相应法律法规（包括但不限于《欧盟 RoHS 指令》）。对于因用户未遵守相应法律法规而导致的损害或损失，瑞萨电子不承担任何责任。
9. 不可将瑞萨电子产品和技术用于或者嵌入日本国内或海外相应的法律法规所禁止生产、使用及销售的任何产品或系统中。也不可将本文档中记载的瑞萨电子产品或技术用于与军事应用或者军事用途有关的任何目的（如大规模杀伤性武器的开发等）。在将本文档中记载的瑞萨电子产品或技术进行出口时，应当遵守相应的出口管制法律法规，并按照上述法律法规所规定的程序进行。
10. 向第三方分销或处分产品或以其他方式将产品置于第三方控制之下的瑞萨电子产品买方或分销商，有责任事先向上述第三方通知本文档规定的内容和条件；对于用户或第三方因非法使用瑞萨电子产品而遭受的任何损失，瑞萨电子不承担任何责任。
11. 在事先未得到瑞萨电子书面认可的情况下，不得以任何形式部分或全部转载或复制本文档。
12. 如果对本文档所记载的信息或瑞萨电子产品有任何疑问，或者用户有任何其他疑问，请向瑞萨电子的营业部门咨询。
(注 1) 瑞萨电子：在本文档中指瑞萨电子株式会社及其控股子公司。
(注 2) 瑞萨电子产品：指瑞萨电子开发或生产的任何产品。

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。

RX63T 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
Short Sheet	硬件的概要	—	—
数据表	硬件的概要和电特性	—	—
用户手册 硬件篇	硬件规格（引脚排列、存储器映像、外围功能规格、电特性、时序）和运行的说明	RX63T 群 用户手册 硬件篇	本硬件手册
用户手册 软件篇	CPU 指令集的说明	RX 族 用户手册 软件篇	R01US0028CJ0100
应用手册	应用例子参考程序等	—	—
RENESAS TECHNICAL UPDATE	产品规格、文档等的快速公告	—	—

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
b0 0 位 (注 2)	0: (注 3) 1: 不能设定。	R/W (注 1)
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: (注 3) 上述以外: 不能设定。	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。
本用户手册仅为参考译文，对应的日文版和英文版具有正式效力。

目 录

特点	1
1. 概要	2
1.1 规格概要	2
1.2 产品一览表	7
1.3 框图	8
1.4 引脚功能	9
1.5 引脚排列图	12
2. CPU	18
2.1 特点	18
2.2 CPU 寄存器组	19
2.2.1 通用寄存器 (R0 ~ R15)	20
2.2.2 控制寄存器	20
2.2.3 DSP 功能指令的相关寄存器	25
2.3 处理器模式	26
2.3.1 管理模式	26
2.3.2 用户模式	26
2.3.3 特权指令	26
2.3.4 处理器模式之间的转移	26
2.4 数据类型	27
2.4.1 整数	27
2.4.2 浮点数	27
2.4.3 位	28
2.4.4 字符串	28
2.5 字节序	29
2.5.1 字节序的设定	29
2.5.2 I/O 寄存器的存取	32
2.5.3 存取 I/O 寄存器时的注意事项	33
2.5.4 数据排列	33
2.6 向量表	34
2.6.1 固定向量表	34
2.6.2 可向量表	34
2.7 指令操作	35
2.7.1 RMPA 指令和字符串操作指令的数据预取	35
2.8 流水线	36
2.8.1 概要	36
2.8.2 指令和流水线处理	37
2.8.3 指令处理时间的计算方法	44
2.8.4 中断响应周期数	44
3. 运行模式	45
3.1 运行模式的种类和选择	45
3.2 寄存器说明	46
3.2.1 模式监视寄存器 (MDMONR)	46
3.2.2 系统控制寄存器 0 (SYSCR0)	46
3.2.3 系统控制寄存器 1 (SYSCR1)	47
3.3 运行模式的说明	47
3.3.1 单芯片模式	47

3.3.2	引导模式	47
3.4	运行模式的转移	48
3.4.1	通过模式引脚进行运行模式的转移	48
3.4.2	通过设定寄存器进行运行模式的转移	48
4.	地址空间	49
4.1	地址空间	49
5.	I/O 寄存器	50
5.1	I/O 寄存器地址一览表（地址顺序）	52
6.	复位	69
6.1	概要	69
6.2	寄存器说明	71
6.2.1	复位状态寄存器 0（RSTSR0）	71
6.2.2	复位状态寄存器 1（RSTSR1）	73
6.2.3	复位状态寄存器 2（RSTSR2）	74
6.2.4	软件复位寄存器（SWRR）	75
6.3	运行说明	76
6.3.1	引脚复位	76
6.3.2	上电复位和电压监视 0 复位	76
6.3.3	电压监视 1 复位和电压监视 2 复位	77
6.3.4	深度软件待机复位	78
6.3.5	独立看门狗定时器复位	78
6.3.6	看门狗定时器复位	78
6.3.7	软件复位	78
6.3.8	冷启动 / 热启动判断功能	79
6.3.9	复位发生源的判断	80
7.	选项设定存储器	81
7.1	概要	81
7.2	寄存器说明	82
7.2.1	选项功能选择寄存器 0（OFS0）	82
7.2.2	选项功能选择寄存器 1（OFS1）	86
7.2.3	字节序选择寄存器 S（MDES）	87
7.3	使用时的注意事项	87
7.3.1	选项设定存储器的设定例子	87
8.	电压检测电路（LVDA）	88
8.1	概要	88
8.2	寄存器说明	91
8.2.1	电压监视 1 电路的控制寄存器 1（LVD1CR1）	91
8.2.2	电压监视 1 电路的状态寄存器（LVD1SR）	91
8.2.3	电压监视 2 电路的控制寄存器 1（LVD2CR1）	92
8.2.4	电压监视 2 电路的状态寄存器（LVD2SR）	92
8.2.5	电压监视电路的控制寄存器（LVCMPCR）	93
8.2.6	电压检测电平选择寄存器（LVDLVLR）	93
8.2.7	电压监视 1 电路的控制寄存器 0（LVD1CR0）	94
8.2.8	电压监视 2 电路的控制寄存器 0（LVD2CR0）	96
8.3	VCC 输入电压的监视	98
8.3.1	Vdet0 的监视	98
8.3.2	Vdet1 的监视	98

8.3.3	Vdet2 的监视	98
8.4	电压监视 0 复位	99
8.5	电压监视 1 中断和电压监视 1 复位	100
8.6	电压监视 2 中断和电压监视 2 复位	102
9.	时钟发生电路	104
9.1	概要	104
9.2	寄存器说明	106
9.2.1	系统时钟控制寄存器 (SCKCR)	106
9.2.2	系统时钟控制寄存器 3 (SCKCR3)	108
9.2.3	PLL 控制寄存器 (PLLCR)	109
9.2.4	PLL 控制寄存器 2 (PLLCR2)	110
9.2.5	主时钟振荡器的控制寄存器 (MOSCCR)	111
9.2.6	低速内部振荡器的控制寄存器 (LOCOCR)	112
9.2.7	IWDT 专用内部振荡器的控制寄存器 (ILOCOCR)	113
9.2.8	振荡停止检测控制寄存器 (OSTDCR)	114
9.2.9	振荡停止检测状态寄存器 (OSTDSR)	115
9.2.10	主时钟振荡器的强制振荡控制寄存器 (MOFCR)	115
9.3	主时钟振荡器	116
9.3.1	连接谐振器的方法	116
9.3.2	输入外部时钟的方法	117
9.3.3	有关外部时钟输入的注意事项	117
9.4	振荡停止检测功能	118
9.4.1	振荡停止检测和检测后的运行	118
9.4.2	振荡停止检测中断	119
9.5	PLL 电路	119
9.6	内部时钟	120
9.6.1	系统时钟	120
9.6.2	定时器模块时钟	120
9.6.3	外围模块时钟	120
9.6.4	S12AD 时钟	120
9.6.5	FlashIF 时钟	120
9.6.6	CAC 时钟	121
9.6.7	IWDT 专用低速时钟	121
9.6.8	JTAG 时钟	121
9.7	连接谐振器时的引脚设定	121
9.8	使用时的注意事项	122
9.8.1	有关时钟发生电路的注意事项	122
9.8.2	有关谐振器的注意事项	122
9.8.3	设计电路板时的注意事项	122
10.	时钟频率精度测量电路 (CAC)	123
10.1	概要	123
10.2	寄存器说明	124
10.2.1	CAC 控制寄存器 0 (CACR0)	124
10.2.2	CAC 控制寄存器 1 (CACR1)	125
10.2.3	CAC 控制寄存器 2 (CACR2)	126
10.2.4	CAC 中断控制寄存器 (CAICR)	127
10.2.5	CAC 状态寄存器 (CASTR)	128
10.2.6	CAC 上限值设定寄存器 (CAULVR)	129
10.2.7	CAC 下限值设定寄存器 (CALLVR)	129
10.2.8	CAC 计数器的缓冲寄存器 (CACNTBR)	129

10.3	运行说明	130
10.3.1	以 CACREF 引脚输入为基准测量时钟频率	130
10.3.2	以其他时钟源为基准测量时钟频率	131
10.3.3	CACREF 引脚的数字滤波器功能	132
10.4	中断请求	132
10.5	使用时的注意事项	132
10.5.1	模块停止功能的设定	132
11.	低功耗功能	133
11.1	概要	133
11.2	寄存器说明	136
11.2.1	待机控制寄存器 (SBYCR)	136
11.2.2	模块停止控制寄存器 A (MSTPCRA)	137
11.2.3	模块停止控制寄存器 B (MSTPCRB)	138
11.2.4	模块停止控制寄存器 C (MSTPCRC)	139
11.2.5	主时钟振荡器的等待控制寄存器 (MOSCWTCR)	140
11.2.6	PLL 等待控制寄存器 (PLLWTCR)	141
11.2.7	深度待机控制寄存器 (DPSBYCR)	142
11.2.8	深度待机中断允许寄存器 0 (DPSIER0)	143
11.2.9	深度待机中断允许寄存器 2 (DPSIER2)	144
11.2.10	深度待机中断标志寄存器 0 (DPSIFR0)	145
11.2.11	深度待机中断标志寄存器 2 (DPSIFR2)	146
11.2.12	深度待机中断边沿寄存器 0 (DPSIEGR0)	147
11.2.13	深度待机中断边沿寄存器 2 (DPSIEGR2)	148
11.2.14	深度待机备用寄存器 (DPSBK _{Ry}) (y=0 ~ 31)	148
11.3	通过转换时钟来降低功耗	149
11.4	模块停止功能	149
11.5	低功耗状态	150
11.5.1	睡眠模式	150
11.5.2	全模块时钟停止模式	151
11.5.3	软件待机模式	152
11.5.4	深度软件待机模式	154
11.6	使用时的注意事项	158
11.6.1	I/O 端口状态	158
11.6.2	DMAC 和 DTC 的模块停止	158
11.6.3	内部外围模块的中断	158
11.6.4	MSTPCRA、MSTPCRB 和 MSTPCRC 寄存器的写操作	158
11.6.5	通过 DIRQnE 位 (n=0 ~ 5) 控制输入缓冲器	158
11.6.6	WAIT 指令的执行时序	158
11.6.7	睡眠模式中通过 DMAC 和 DTC 进行的寄存器改写	158
11.6.8	从软件待机模式返回时的注意事项 (可参照 63N)	158
12.	寄存器写保护功能	159
12.1	寄存器说明	160
12.1.1	保护寄存器 (PRCR)	160
13.	异常处理	161
13.1	异常事件	161
13.1.1	未定义指令异常	162
13.1.2	特权指令异常	162
13.1.3	存取异常	162
13.1.4	浮点异常	162

13.1.5	复位	162
13.1.6	非屏蔽中断	162
13.1.7	中断	162
13.1.8	无条件陷阱	162
13.2	异常处理步骤	163
13.3	异常事件的接受	165
13.3.1	接受时序和被保存的 PC 值	165
13.3.2	向量和 PC、PSW 的保存位置	165
13.4	接受异常 / 从异常返回时的硬件处理	166
13.5	硬件预处理	167
13.5.1	未定义指令异常	167
13.5.2	特权指令异常	167
13.5.3	存取异常	167
13.5.4	浮点异常	167
13.5.5	复位	167
13.5.6	非屏蔽中断	168
13.5.7	中断	168
13.5.8	无条件陷阱	168
13.6	从异常处理程序的返回	169
13.7	异常事件的优先级	169
14.	中断控制器 (ICUb)	170
14.1	概要	170
14.2	寄存器说明	172
14.2.1	中断请求寄存器 n (IRn) (n= 中断向量号)	172
14.2.2	中断请求允许寄存器 m (IERm) (m=02h ~ 1Fh)	173
14.2.3	中断源优先级寄存器 n (IPRn) (n=000 ~ 250)	174
14.2.4	高速中断设定寄存器 (FIR)	175
14.2.5	软件中断启动寄存器 (SWINTR)	176
14.2.6	DTC 启动允许寄存器 n (DTCERn) (n= 中断向量号)	176
14.2.7	DMAC 启动请求选择寄存器 m (DMRSRm) (m=DMAC 通道号)	177
14.2.8	IRQ 控制寄存器 i (IRQCRi) (i=0 ~ 5)	177
14.2.9	IRQ 引脚数字滤波器的允许寄存器 0 (IRQFLTE0)	178
14.2.10	IRQ 引脚数字滤波器的设定寄存器 0 (IRQFLTC0)	179
14.2.11	非屏蔽中断状态寄存器 (NMISR)	180
14.2.12	非屏蔽中断允许寄存器 (NMIER)	182
14.2.13	非屏蔽中断状态清除寄存器 (NMICLR)	184
14.2.14	NMI 引脚中断控制寄存器 (NMICR)	185
14.2.15	NMI 引脚数字滤波器的允许寄存器 (NMIFLTE)	185
14.2.16	NMI 引脚数字滤波器的设定寄存器 (NMIFLTC)	186
14.2.17	组 12 中断源寄存器 (GRP12)	187
14.2.18	组 12 中断允许寄存器 (GEN12)	188
14.3	向量表	189
14.3.1	中断向量表	189
14.3.2	高速中断的向量表地址	195
14.3.3	非屏蔽中断的向量表地址	195
14.4	外围模块中断请求组的功能	196
14.4.1	中断请求组	196
14.5	中断的运行说明	197
14.5.1	中断检测	197
14.5.2	中断请求的允许 / 禁止	201
14.5.3	中断请求目标的选择	202

14.5.4	优先级的判断	204
14.5.5	高速中断	204
14.5.6	数字滤波器	204
14.5.7	外部引脚中断	205
14.6	非屏蔽中断的运行说明	205
14.7	从低功耗状态的返回	206
14.7.1	从睡眠模式的返回	206
14.7.2	从全模块时钟停止模式的返回	206
14.7.3	从软件待机模式的返回	206
14.8	使用时的注意事项	207
14.8.1	使用非屏蔽中断时的 WAIT 指令的注意事项	207
15.	总线	208
15.1	概要	208
15.2	总线说明	210
15.2.1	CPU 总线	210
15.2.2	存储器总线	210
15.2.3	内部主总线	210
15.2.4	内部外围总线	211
15.2.5	写缓冲器功能（内部外围总线）	212
15.2.6	并行运行	212
15.2.7	限制事项	212
15.3	寄存器说明	213
15.3.1	总线错误状态清除寄存器（BERCLR）	213
15.3.2	总线错误监视允许寄存器（BEREN）	213
15.3.3	总线错误状态寄存器 1（BERSR1）	214
15.3.4	总线错误状态寄存器 2（BERSR2）	214
15.3.5	总线优先级控制寄存器（BUSPRI）	215
15.4	总线错误监视部	217
15.4.1	总线错误的种类	217
15.4.2	发生总线错误时的运行	217
15.4.3	总线错误的发生条件	218
16.	存储器保护单元（MPU）	219
16.1	概要	219
16.1.1	存取控制的种类	221
16.1.2	存取控制区域	221
16.1.3	后台区域	221
16.1.4	区域的重叠	221
16.1.5	跨区域的指令和数据	221
16.2	寄存器说明	222
16.2.1	区域 n 起始页号寄存器（RSPAGEn）（n=0 ~ 7）	222
16.2.2	区域 n 结束页号寄存器（REPAGEn）（n=0 ~ 7）	223
16.2.3	存储器保护功能有效化寄存器（MPEN）	224
16.2.4	后台存取控制寄存器（MPBAC）	225
16.2.5	存储器保护错误状态清除寄存器（MPECLR）	226
16.2.6	存储器保护错误状态寄存器（MPESTS）	227
16.2.7	数据存储器保护错误地址寄存器（MPDEA）	228
16.2.8	区域查找地址寄存器（MPSA）	228
16.2.9	区域查找操作寄存器（MPOPS）	229
16.2.10	区域无效化操作寄存器（MPOPI）	229
16.2.11	指令命中区域寄存器（MHITI）	230

16.2.12	数据命中区域寄存器 (MHITD)	232
16.3	功能	234
16.3.1	存储器保护功能	234
16.3.2	区域查找功能	234
16.3.3	存储器保护单元相关寄存器的保护	234
16.3.4	存储器保护功能的存取判断流程	235
16.4	存储器保护功能的使用步骤	237
16.4.1	存取控制信息的设定	237
16.4.2	存储器保护功能的有效化	237
16.4.3	向用户模式的转移	237
16.4.4	发生存储器保护错误时的处理	237
17.	DMA 控制器 (DMACA)	239
17.1	概要	239
17.2	寄存器说明	241
17.2.1	DMA 传送源地址寄存器 (DMSAR)	241
17.2.2	DMA 传送目标地址寄存器 (DMDAR)	241
17.2.3	DMA 传送计数寄存器 (DMCRA)	242
17.2.4	DMA 块传送计数寄存器 (DMCRB)	243
17.2.5	DMA 传送模式寄存器 (DMTMD)	244
17.2.6	DMA 中断设定寄存器 (DMINT)	245
17.2.7	DMA 地址模式寄存器 (DMAMD)	247
17.2.8	DMA 偏移寄存器 (DMOFR)	250
17.2.9	DMA 传送允许寄存器 (DMCNT)	250
17.2.10	DMA 软件启动寄存器 (DMREQ)	251
17.2.11	DMA 状态寄存器 (DMSTS)	252
17.2.12	DMA 启动源标志控制寄存器 (DMCSL)	253
17.2.13	DMA 模块启动寄存器 (DMAST)	254
17.3	运行说明	255
17.3.1	传送模式	255
17.3.2	扩展重复区域功能	258
17.3.3	带偏移的地址更新功能	259
17.3.4	启动源	263
17.3.5	运行时序	263
17.3.6	DMAC 的执行周期	264
17.3.7	DMAC 的启动	265
17.3.8	DMA 传送的开始	266
17.3.9	DMA 传送过程中的寄存器	266
17.3.10	通道的优先级	267
17.4	DMA 传送的结束	268
17.4.1	通过设定的总数据传送结束传送	268
17.4.2	通过重复大小结束中断结束传送	268
17.4.3	通过扩展重复区域上溢中断结束传送	268
17.5	中断	269
17.6	低功耗功能	271
17.7	使用时的注意事项	272
17.7.1	外围模块的 DMA 传送情况	272
17.7.2	有关 DMA 运行过程中的寄存器存取	272
17.7.3	有关保留区的 DMA 传送	272
17.7.4	因设定 DMA 启动源标志控制寄存器 (DMCSL) 而产生的各传送结束中断请求	272
17.7.5	中断控制器的 DMAC 启动请求寄存器 (ICU.DMRSRm) 的设定	272
17.7.6	保留或者重新开始 DMA 启动的方法	272

18. 数据传送控制器 (DTCa)	273
18.1 概要	273
18.2 寄存器说明	275
18.2.1 DTC 模式寄存器 A (MRA)	275
18.2.2 DTC 模式寄存器 B (MRB)	276
18.2.3 DTC 传送源寄存器 (SAR)	277
18.2.4 DTC 传送目标寄存器 (DAR)	277
18.2.5 DTC 传送计数寄存器 A (CRA)	278
18.2.6 DTC 传送计数寄存器 B (CRB)	279
18.2.7 DTC 控制寄存器 (DTCCR)	279
18.2.8 DTC 向量基址寄存器 (DTCVBR)	280
18.2.9 DTC 地址模式寄存器 (DTCADM0D)	280
18.2.10 DTC 模块启动寄存器 (DTCST)	281
18.2.11 DTC 状态寄存器 (DTCSTS)	282
18.3 启动源	283
18.3.1 传送信息的分配和 DTC 向量表	283
18.4 运行说明	284
18.4.1 传送信息的跳读功能	286
18.4.2 传送信息的回写省略功能	287
18.4.3 正常传送模式	288
18.4.4 重复传送模式	289
18.4.5 块传送模式	290
18.4.6 链传送	291
18.4.7 运行时序	292
18.4.8 DTC 的执行周期	295
18.4.9 DTC 的总线权释放时序	295
18.5 DTC 的设定步骤	296
18.6 DTC 的使用例子	297
18.6.1 正常传送	297
18.6.2 计数器为“0”时的链传送	298
18.7 中断源	299
18.8 低功耗功能	299
18.9 使用时的注意事项	300
18.9.1 传送信息起始地址 / 传送源地址 / 传送目标地址	300
18.9.2 传送信息的分配	300
18.9.3 中断控制器的 DTC 启动允许寄存器 (ICU.DTCERn) 的设定	300
19. I/O 端口	301
19.1 概要	301
19.2 输入 / 输出端口的结构	302
19.3 寄存器说明	303
19.3.1 端口方向寄存器 (PDR)	303
19.3.2 端口输出数据寄存器 (PODR)	304
19.3.3 端口输入数据寄存器 (PIDR)	305
19.3.4 端口模式寄存器 (PMR)	306
19.3.5 漏极开路控制寄存器 0 (ODR0)	307
19.3.6 漏极开路控制寄存器 1 (ODR1)	307
19.4 未使用引脚的处理	308

20. 多功能引脚控制器 (MPC)	309
20.1 概要	309
20.2 寄存器说明	312
20.2.1 写保护寄存器 (PWPR)	312
20.2.2 P0n 引脚功能控制寄存器 (P0nPFS) (n=0、1)	313
20.2.3 P1n 引脚功能控制寄存器 (P1nPFS) (n=0、1)	314
20.2.4 P2n 引脚功能控制寄存器 (P2nPFS) (n=2 ~ 4)	315
20.2.5 P3n 引脚功能控制寄存器 (P3nPFS) (n=0 ~ 3)	316
20.2.6 P4n 引脚功能控制寄存器 (P4nPFS) (n=0 ~ 7)	317
20.2.7 P7n 引脚功能控制寄存器 (P7nPFS) (n=0 ~ 6)	317
20.2.8 P9n 引脚功能控制寄存器 (P9nPFS) (n=1 ~ 4)	318
20.2.9 PAn 引脚功能控制寄存器 (PAnPFS) (n=2 ~ 5)	319
20.2.10 PBn 引脚功能控制寄存器 (PBnPFS) (n=0 ~ 7)	320
20.2.11 PDn 引脚功能控制寄存器 (PDnPFS) (n=3 ~ 7)	322
20.2.12 PE2 引脚功能控制寄存器 (PE2PFS)	322
20.3 使用时的注意事项	323
20.3.1 引脚输入 / 输出功能的设定步骤	323
20.3.2 设定 MPC 寄存器时的注意事项	323
20.3.3 使用模拟功能时的注意事项	323
21. 多功能定时器脉冲单元 3 (MTU3)	324
21.1 概要	324
21.2 寄存器说明	330
21.2.1 定时器的控制寄存器 (TCR)	330
21.2.2 定时器的模式寄存器 1 (TMDR1)	334
21.2.3 定时器的模式寄存器 2 (TMDR2A、TMDR2B)	336
21.2.4 定时器的 I/O 控制寄存器 (TIOR)	336
21.2.5 定时器的比较匹配清除寄存器 (TCNTCMPCLR)	351
21.2.6 定时器的中断允许寄存器 (TIER)	352
21.2.7 定时器的状态寄存器 (TSR)	355
21.2.8 定时器的缓冲运行传送模式寄存器 (TBTM)	359
21.2.9 定时器的输入捕捉控制寄存器 (TICCR)	360
21.2.10 定时器的同步清除寄存器 (TSYCR)	361
21.2.11 定时器的计数器 (TCNT)	361
21.2.12 定时器的通用寄存器 (TGR)	362
21.2.13 定时器的启动寄存器 (TSTR)	363
21.2.14 定时器的同步寄存器 (TSYR)	365
21.2.15 定时器的计数器同步启动寄存器 (TCSYSTR)	367
21.2.16 定时器的读写允许寄存器 (TRWERA、TRWERB)	369
21.2.17 定时器的输出主控允许寄存器 (TOER)	370
21.2.18 定时器的输出控制寄存器 1 (TOCR1A、TOCR1B)	372
21.2.19 定时器的输出控制寄存器 2 (TOCR2A、TOCR2B)	374
21.2.20 定时器的输出电平缓冲寄存器 (TOLBRA、TOLBRB)	376
21.2.21 定时器的门控寄存器 A (TGCRA)	377
21.2.22 定时器的副计数器 (TCNTSA、TCNTSB)	378
21.2.23 定时器的周期数据寄存器 (TCDRA、TCDRB)	378
21.2.24 定时器的周期缓冲寄存器 (TCBRA、TCBRB)	379
21.2.25 定时器的死区时间数据寄存器 (TDDRA、TDDR B)	379
21.2.26 定时器的死区时间允许寄存器 (TDERA、TDERB)	379
21.2.27 定时器的缓冲传送设定寄存器 (TBTERA、TBTERB)	380
21.2.28 定时器的波形控制寄存器 (TWCRA、TWCRB)	381
21.2.29 定时器的 A/D 转换开始请求控制寄存器 (TADCR)	383

21.2.30	定时器的 A/D 转换开始请求周期设定寄存器 (TADCORA、TADCORB)	387
21.2.31	定时器的 A/D 转换开始请求周期设定缓冲寄存器 (TADCOBRA、TADCOBRB)	387
21.2.32	定时器的中断减少模式寄存器 (TITMRA、TITMRB)	388
21.2.33	定时器的中断减少设定寄存器 1 (TITCR1A、TITCR1B)	389
21.2.34	定时器的中断减少次数计数器 1 (TITCNT1A、TITCNT1B)	391
21.2.35	定时器的中断减少设定寄存器 2 (TITCR2A、TITCR2B)	393
21.2.36	定时器的中断减少次数计数器 2 (TITCNT2A、TITCNT2B)	395
21.2.37	和总线主控的接口	396
21.3	运行说明	397
21.3.1	基本运行	397
21.3.2	同步运行	401
21.3.3	缓冲运行	402
21.3.4	级联运行	406
21.3.5	PWM 模式	410
21.3.6	相位计数模式	415
21.3.7	复位同步 PWM 模式	421
21.3.8	互补 PWM 模式	424
21.3.9	A/D 转换开始请求的延迟功能	459
21.3.10	MTU0 ~ 4 - MTU6、7 的同步运行	464
21.3.11	外部脉宽的测量功能	466
21.3.12	死区时间的补偿功能	467
21.3.13	在互补 PWM 的“波峰 / 波谷”进行的 TCNT 捕捉运行	468
21.4	中断源	469
21.4.1	中断源和优先级	469
21.4.2	DTC 的启动	471
21.4.3	A/D 转换器的启动	471
21.5	运行时序	473
21.5.1	输入 / 输出时序	473
21.5.2	中断信号的时序	478
21.6	使用时的注意事项	484
21.6.1	模块时钟停止模式的设定	484
21.6.2	输入时钟的限制事项	484
21.6.3	设定周期时的注意事项	484
21.6.4	TCNT 的写和清除的竞争	485
21.6.5	TCNT 的写和递增计数的竞争	485
21.6.6	TGR 的写和比较匹配的竞争	486
21.6.7	缓冲寄存器的写和比较匹配的竞争	486
21.6.8	缓冲寄存器的写和 TCNT 清除的竞争	487
21.6.9	TGR 寄存器的读和输入捕捉的竞争	487
21.6.10	TGR 寄存器的写和输入捕捉的竞争	488
21.6.11	缓冲寄存器的写和输入捕捉的竞争	489
21.6.12	级联中的 MTU2.TCNT 的写和上溢 / 下溢的竞争	490
21.6.13	互补 PWM 模式停止时的计数器值	491
21.6.14	互补 PWM 模式中的缓冲运行的设定	491
21.6.15	复位同步 PWM 模式的缓冲运行和比较匹配标志	492
21.6.16	复位同步 PWM 模式的上溢标志	493
21.6.17	上溢 / 下溢和计数器清除的竞争	493
21.6.18	TCNT 的写和上溢 / 下溢的竞争	494
21.6.19	从正常运行或者 PWM 模式 1 转移到复位同步 PWM 模式时的注意事项	494
21.6.20	互补 PWM 模式、复位同步 PWM 模式的输出电平	494
21.6.21	级联中的 MTU1.TCNT 和 MTU2.TCNT 的同时输入捕捉	495
21.6.22	中断减少功能 2	495

21.7	MTU 输出引脚的初始化方法	496
21.7.1	运行模式	496
21.7.2	因运行过程中的异常而重新设定时的运行	496
21.7.3	因运行过程中的异常而对引脚进行初始化的步骤以及模式转移的概要	497
22.	端口输出允许 3 (POE3)	519
22.1	概要	519
22.2	寄存器说明	522
22.2.1	输入电平控制 / 状态寄存器 1 (ICSR1)	522
22.2.2	输出电平控制 / 状态寄存器 1 (OCSR1)	523
22.2.3	有效电平寄存器 1 (ALR1)	524
22.2.4	输入电平控制 / 状态寄存器 3 (ICSR3)	526
22.2.5	输入电平控制 / 状态寄存器 4 (ICSR4)	527
22.2.6	输入电平控制 / 状态寄存器 5 (ICSR5)	528
22.2.7	软件端口输出允许寄存器 (SPOER)	529
22.2.8	端口输出允许控制寄存器 1 (POECR1)	531
22.2.9	端口输出允许控制寄存器 2 (POECR2)	532
22.2.10	端口输出允许控制寄存器 3 (POECR3)	533
22.2.11	端口输出允许控制寄存器 4 (POECR4)	534
22.2.12	端口输出允许控制寄存器 5 (POECR5)	535
22.2.13	端口输出允许控制寄存器 6 (POECR6)	536
22.2.14	输入电平控制 / 状态寄存器 6 (ICSR6)	538
22.3	运行说明	539
22.3.1	输入电平的检测	543
22.3.2	输出电平的比较	544
22.3.3	通过寄存器进行的高阻抗控制	544
22.3.4	通过振荡停止检测进行的高阻抗控制	544
22.3.5	通过比较器检测进行的高阻抗控制	544
22.3.6	高阻抗控制条件的追加功能	544
22.3.7	高阻抗状态的解除	545
22.4	中断	545
22.5	使用时的注意事项	545
23.	通用 PWM 定时器 (GPT)	546
23.1	概要	546
23.2	寄存器说明	550
23.2.1	通用 PWM 定时器的软件启动寄存器 (GTSTR)	550
23.2.2	通用 PWM 定时器的硬件源启动控制寄存器 (GTHSCR)	551
23.2.3	通用 PWM 定时器的硬件源清除控制寄存器 (GTHCCR)	552
23.2.4	通用 PWM 定时器的硬件启动源选择寄存器 (GTHSSR)	553
23.2.5	通用 PWM 定时器的硬件停止 / 清除源选择寄存器 (GTHPSR)	555
23.2.6	通用 PWM 定时器的写保护寄存器 (GTWP)	557
23.2.7	通用 PWM 定时器的同步寄存器 (GTSYNC)	558
23.2.8	通用 PWM 定时器的外部触发输入中断寄存器 (GTETINT)	559
23.2.9	通用 PWM 定时器的缓冲运行禁止寄存器 (GTBDR)	560
23.2.10	通用 PWM 定时器启动的写保护寄存器 (GTSWP)	561
23.2.11	LOCO 计数控制寄存器 (LCCR)	562
23.2.12	LOCO 计数状态寄存器 (LCST)	564
23.2.13	LOCO 计数值寄存器 (LCNT)	565
23.2.14	LOCO 计数结果平均寄存器 (LCNTA)	565
23.2.15	LOCO 计数结果寄存器 n (LCNTn) (n=00 ~ 15)	565
23.2.16	LOCO 计数上限 / 下限容许偏差值寄存器 (LCNTDU、LCNTDL)	566

23.2.17	通用 PWM 定时器的 I/O 控制寄存器 (GTIOR)	566
23.2.18	通用 PWM 定时器的中断输出设定寄存器 (GTINTAD)	570
23.2.19	通用 PWM 定时器的控制寄存器 (GTCR)	572
23.2.20	通用 PWM 定时器的缓冲允许寄存器 (GTBER)	573
23.2.21	通用 PWM 定时器的计数方向寄存器 (GTUDC)	575
23.2.22	通用 PWM 定时器的中断、A/D 转换开始请求减少设定寄存器 (GTITC)	576
23.2.23	通用 PWM 定时器的状态寄存器 (GTST)	578
23.2.24	通用 PWM 定时器的计数器 (GTCNT)	581
23.2.25	通用 PWM 定时器的比较捕捉寄存器 m (GTCCRm) (m=A ~ F)	582
23.2.26	通用 PWM 定时器的周期设定寄存器 (GTPR)	582
23.2.27	通用 PWM 定时器的周期设定缓冲寄存器 (GTPBR)	582
23.2.28	通用 PWM 定时器的周期设定双缓冲寄存器 (GTPDBR)	583
23.2.29	A/D 转换开始请求时序寄存器 m (GTADTRm) (m=A、B)	583
23.2.30	A/D 转换开始请求时序缓冲寄存器 m (GTADTBRm) (m=A、B)	583
23.2.31	A/D 转换开始请求时序双缓冲寄存器 m (GTADTDBRm) (m=A、B)	583
23.2.32	通用 PWM 定时器的输出无效控制寄存器 (GTONCR)	584
23.2.33	通用 PWM 定时器的死区时间控制寄存器 (GTDTCR)	586
23.2.34	通用 PWM 定时器的死区时间值寄存器 m (GTDVm) (m=U、D)	587
23.2.35	通用 PWM 定时器的死区时间缓冲寄存器 m (GTDBm) (m=U、D)	587
23.2.36	通用 PWM 定时器的输出保护功能状态寄存器 (GTSOS)	588
23.2.37	通用 PWM 定时器的输出保护功能暂时解除寄存器 (GTSOTR)	588
23.3	运行说明	589
23.3.1	基本运行	589
23.3.2	缓冲运行	598
23.3.3	PWM 输出运行模式	609
23.3.4	死区时间自动设定功能	621
23.3.5	计数方向转换功能	625
23.3.6	硬件启动 / 停止、清除运行	626
23.3.7	同步运行	631
23.3.8	PWM 输出的运行例子	639
23.4	中断源	645
23.4.1	中断源和优先级	645
23.4.2	DTC 的启动	648
23.4.3	中断、A/D 转换开始请求的减少功能	648
23.5	A/D 转换开始请求	652
23.6	IWDTCLK 计数功能	654
23.7	保护功能	657
23.7.1	计数器的写保护	657
23.7.2	缓冲运行的抑制	657
23.7.3	GTIOC 引脚输出的无效控制	658
23.7.4	GTIOC 引脚输出的输出保护功能	659
23.7.5	通过 POE 功能进行 GTIOC 引脚输出的高阻抗控制	663
23.8	输出引脚的初始化方法	664
23.8.1	复位后的引脚设定	664
23.8.2	因运行中的异常等引起的引脚初始化	664
23.9	使用时的注意事项	665
23.9.1	模块停止功能的设定	665
23.9.2	比较匹配运行时的 GTCCRn 寄存器的设定 (n=A、B、C、D、E、F)	665
23.9.3	安全停止定时器的方法	666
23.9.4	使用 IWDTCLK 计数功能时的低功耗功能的设定	666

24.	比较匹配定时器 (CMT)	667
24.1	概要	667
24.2	寄存器说明	668
24.2.1	比较匹配定时器的启动寄存器 0 (CMSTR0)	668
24.2.2	比较匹配定时器的启动寄存器 1 (CMSTR1)	668
24.2.3	比较匹配定时器的控制寄存器 (CMCR)	669
24.2.4	比较匹配定时器的计数器 (CMCNT)	670
24.2.5	比较匹配定时器的常数寄存器 (CMCOR)	670
24.3	运行说明	671
24.3.1	周期计数	671
24.3.2	CMCNT 计数器的计数时序	671
24.4	中断	672
24.4.1	中断源	672
24.4.2	比较匹配中断的发生时序	672
24.5	使用时的注意事项	673
24.5.1	模块停止功能的设定	673
24.5.2	CMCNT 计数器的写和比较匹配的竞争	673
24.5.3	CMCNT 计数器的写和递增计数的竞争	673
25.	看门狗定时器 (WDTA)	674
25.1	概要	674
25.2	寄存器说明	676
25.2.1	WDT 刷新寄存器 (WDTRR)	676
25.2.2	WDT 控制寄存器 (WDTCR)	677
25.2.3	WDT 状态寄存器 (WDTSR)	680
25.2.4	WDT 复位控制寄存器 (WDTRCR)	681
25.2.5	选项功能选择寄存器 0 (OFS0)	681
25.3	运行说明	681
25.3.1	各计数开始条件下的运行	681
25.3.2	WDTCR 寄存器和 WDTRCR 寄存器的写控制	684
25.3.3	刷新	684
25.3.4	状态标志	685
25.3.5	复位输出	686
25.3.6	中断源	686
25.3.7	递减计数器值的读操作	686
25.3.8	选项功能选择寄存器 0 (OFS0) 和 WDT 寄存器的对应	687
26.	独立看门狗定时器 (IWDTa)	688
26.1	概要	688
26.2	寄存器说明	690
26.2.1	IWDT 刷新寄存器 (IWDTRR)	690
26.2.2	IWDT 控制寄存器 (IWDTCR)	691
26.2.3	IWDT 状态寄存器 (IWDTSR)	694
26.2.4	IWDT 复位控制寄存器 (IWDTRCR)	695
26.2.5	IWDT 计数停止控制寄存器 (IWDTCSTPR)	695
26.2.6	选项功能选择寄存器 0 (OFS0)	695
26.3	运行说明	696
26.3.1	各计数开始条件下的运行	696
26.3.2	IWDTCR 寄存器、IWDTRCR 寄存器和 IWDTCSTPR 寄存器的写控制	699
26.3.3	刷新	699
26.3.4	状态标志	701
26.3.5	复位输出	701

26.3.6	中断源	702
26.3.7	递减计数器值的读操作	702
26.3.8	选项功能选择寄存器 0 (OFS0) 和 IWDT 寄存器的对应	703
26.4	使用时的注意事项	703
26.4.1	刷新	703
27.	串行通信接口 (SC1c、SC1d)	704
27.1	概要	704
27.2	寄存器说明	709
27.2.1	接收移位寄存器 (RSR)	709
27.2.2	接收数据寄存器 (RDR)	709
27.2.3	发送数据寄存器 (TDR)	709
27.2.4	发送移位寄存器 (TSR)	709
27.2.5	串行模式寄存器 (SMR)	710
27.2.6	串行控制寄存器 (SCR)	714
27.2.7	串行状态寄存器 (SSR)	718
27.2.8	智能卡模式寄存器 (SCMR)	722
27.2.9	位速率寄存器 (BRR)	723
27.2.10	串行扩展模式寄存器 (SEMR)	731
27.2.11	噪声滤波器的设定寄存器 (SNFR)	732
27.2.12	I ² C 模式寄存器 1 (SIMR1)	733
27.2.13	I ² C 模式寄存器 2 (SIMR2)	734
27.2.14	I ² C 模式寄存器 3 (SIMR3)	735
27.2.15	I ² C 状态寄存器 (SISR)	737
27.2.16	SPI 模式寄存器 (SPMR)	738
27.2.17	扩展串行模式有效寄存器 (ESMER)	740
27.2.18	控制寄存器 0 (CR0)	740
27.2.19	控制寄存器 1 (CR1)	741
27.2.20	控制寄存器 2 (CR2)	742
27.2.21	控制寄存器 3 (CR3)	743
27.2.22	端口控制寄存器 (PCR)	743
27.2.23	中断控制寄存器 (ICR)	744
27.2.24	状态寄存器 (STR)	745
27.2.25	状态清除寄存器 (STCR)	746
27.2.26	Control Field 0 数据寄存器 (CF0DR)	746
27.2.27	Control Field 0 比较允许寄存器 (CF0CR)	747
27.2.28	Control Field 0 接收数据寄存器 (CF0RR)	747
27.2.29	主 Control Field 1 数据寄存器 (PCF1DR)	747
27.2.30	次 Control Field 1 数据寄存器 (SCF1DR)	748
27.2.31	Control Field 1 比较允许寄存器 (CF1CR)	748
27.2.32	Control Field 1 接收数据寄存器 (CF1RR)	748
27.2.33	定时器的控制寄存器 (TCR)	749
27.2.34	定时器的模式寄存器 (TMR)	749
27.2.35	定时器的预分频寄存器 (TPRE)	750
27.2.36	定时器的计数器 (TCNT)	750
27.3	异步模式的运行	751
27.3.1	串行发送 / 接收格式	752
27.3.2	异步模式的接收数据采样时序和接收容限	753
27.3.3	时钟	754
27.3.4	CTS 和 RTS 功能	754
27.3.5	SCI 的初始化 (异步模式)	755
27.3.6	串行数据的发送 (异步模式)	756

27.3.7	串行数据的接收（异步模式）	758
27.4	多处理器通信功能	762
27.4.1	多处理器串行数据的发送	763
27.4.2	多处理器串行数据的接收	764
27.5	时钟同步模式的运行	767
27.5.1	时钟	767
27.5.2	CTS 和 RTS 功能	767
27.5.3	SCI 的初始化（时钟同步模式）	768
27.5.4	串行数据的发送（时钟同步模式）	769
27.5.5	串行数据的接收（时钟同步模式）	770
27.5.6	串行数据的同时发送和接收（时钟同步模式）	773
27.6	智能卡接口模式的运行	774
27.6.1	连接例子	774
27.6.2	数据格式（块传送模式除外）	775
27.6.3	块传送模式	776
27.6.4	接收数据的采样时序和接收容限	776
27.6.5	SCI 的初始化（智能卡接口模式）	777
27.6.6	串行数据的发送（块传送模式除外）	778
27.6.7	串行接收（块传送模式除外）	781
27.6.8	时钟的输出控制	783
27.7	简易 I ² C 模式的运行	784
27.7.1	开始条件、重新开始条件和停止条件的生成	785
27.7.2	时钟同步	786
27.7.3	SSDA 输出延迟	787
27.7.4	SCI 的初始化（简易 I ² C 模式）	788
27.7.5	主控发送（简易 I ² C 模式）	789
27.7.6	主控接收（简易 I ² C 模式）	791
27.8	简易 SPI 模式的运行	793
27.8.1	主控模式、从属模式和各引脚的状态	794
27.8.2	主控模式中的 SS 功能	794
27.8.3	从属模式中的 SS 功能	794
27.8.4	时钟和发送 / 接收数据的关系	794
27.8.5	SCI 的初始化（简易 SPI 模式）	795
27.8.6	串行数据的发送和接收（简易 SPI 模式）	795
27.9	扩展串行模式控制部的运行说明	796
27.9.1	串行通信协议	796
27.9.2	Start Frame 发送	797
27.9.3	Start Frame 接收	800
27.9.4	总线冲突检测功能	806
27.9.5	RXDX12 引脚输入的数字滤波器功能	807
27.9.6	位速率测量功能	808
27.9.7	RXDX12 接收数据的采样时序选择功能	809
27.9.8	定时器	810
27.10	噪声消除功能	812
27.11	中断源	813
27.11.1	TXI 中断和 RXI 中断的缓冲运行	813
27.11.2	串行通信接口模式和简易 SPI 模式的中断	813
27.11.3	智能卡接口模式的中断	814
27.11.4	简易 I ² C 模式的中断	815
27.11.5	扩展串行模式控制部的中断请求	815
27.12	使用时的注意事项	816
27.12.1	模块停止功能的设定	816

27.12.2	有关中止的检测和处理	816
27.12.3	标记状态和中止的发送	816
27.12.4	有关接收错误标志和发送（只限于时钟同步模式）	816
27.12.5	有关 TDR 寄存器的写操作	816
27.12.6	时钟同步发送时的限制事项	816
27.12.7	使用 DMAC 或者 DTC 时的限制事项	816
27.12.8	有关开始通信的注意事项	817
27.12.9	有关低功耗状态时的运行	817
27.12.10	时钟同步模式的外部时钟输入	819
27.12.11	简易 SPI 模式的限制事项	820
27.12.12	扩展串行模式控制部的使用限制事项 1	820
27.12.13	扩展串行模式控制部的使用限制事项 2	821
28.	I²C 总线接口 (RIIC)	822
28.1	概要	822
28.2	寄存器说明	825
28.2.1	I ² C 总线控制寄存器 1 (ICCR1)	825
28.2.2	I ² C 总线控制寄存器 2 (ICCR2)	827
28.2.3	I ² C 总线模式寄存器 1 (ICMR1)	830
28.2.4	I ² C 总线模式寄存器 2 (ICMR2)	831
28.2.5	I ² C 总线模式寄存器 3 (ICMR3)	833
28.2.6	I ² C 总线功能允许寄存器 (ICFER)	835
28.2.7	I ² C 总线状态允许寄存器 (ICSER)	837
28.2.8	I ² C 总线中断允许寄存器 (ICIER)	838
28.2.9	I ² C 总线状态寄存器 1 (ICSR1)	840
28.2.10	I ² C 总线状态寄存器 2 (ICSR2)	843
28.2.11	从属地址寄存器 Ly (SARLy) (y=0 ~ 2)	846
28.2.12	从属地址寄存器 Uy (SARUy) (y=0 ~ 2)	847
28.2.13	I ² C 总线位速率低电平寄存器 (ICBRL)	848
28.2.14	I ² C 总线位速率高电平寄存器 (ICBRH)	848
28.2.15	I ² C 总线发送数据寄存器 (ICDRT)	850
28.2.16	I ² C 总线接收数据寄存器 (ICDRR)	850
28.2.17	I ² C 总线移位寄存器 (ICDRS)	850
28.2.18	超时内部计数器 (TMOCNT)	851
28.3	运行说明	852
28.3.1	通信数据的格式	852
28.3.2	初始设定	853
28.3.3	主控发送	854
28.3.4	主控接收	858
28.3.5	从属发送	861
28.3.6	从属接收	864
28.4	SCL 同步电路	866
28.5	SDA 输出延迟功能	867
28.6	数字噪声滤波器电路	868
28.7	地址匹配检测功能	869
28.7.1	从属地址匹配检测功能	869
28.7.2	全呼地址检测功能	871
28.7.3	设备 ID 地址检测功能	872
28.7.4	主机地址检测功能	873
28.8	SCL 的 Low 电平自动保持功能	874
28.8.1	发送数据的误发送防止功能	874
28.8.2	NACK 接收传送中止功能	875

28.8.3	接收数据漏取防止功能	875
28.9	仲裁失败检测功能	877
28.9.1	主控仲裁失败检测功能 (MALE 位)	877
28.9.2	NACK 发送仲裁失败检测功能 (NALE 位)	879
28.9.3	从属仲裁失败检测功能 (SALE 位)	880
28.10	开始条件、重新开始条件和停止条件的发行功能	881
28.10.1	开始条件的发行	881
28.10.2	重新开始条件的发行	881
28.10.3	停止条件的发行	882
28.11	总线意外停机	883
28.11.1	超时检测功能	883
28.11.2	SCL 时钟追加输出功能	884
28.11.3	RIIC/ 内部复位	884
28.12	SMBus 运行	885
28.12.1	SMBus 超时测量	885
28.12.2	数据包错误码 (PEC)	886
28.12.3	SMBus 主机通知协议 /Notify ARP master	886
28.13	中断源	887
28.13.1	ICTXI 中断和 ICRXI 中断的缓冲运行	887
28.14	复位状况	888
28.15	使用时的注意事项	889
28.15.1	模块停止功能的设定	889
28.15.2	有关开始通信的注意事项	889
29.	串行外围接口 (RSPI)	890
29.1	概要	890
29.2	寄存器说明	893
29.2.1	RSPI 控制寄存器 (SPCR)	893
29.2.2	RSPI 从属选择极性寄存器 (SSLP)	894
29.2.3	RSPI 引脚控制寄存器 (SPPCR)	895
29.2.4	RSPI 状态寄存器 (SPSR)	896
29.2.5	RSPI 数据寄存器 (SPDR)	898
29.2.6	RSPI 顺序控制寄存器 (SPSCR)	901
29.2.7	RSPI 顺序状态寄存器 (SPSSR)	902
29.2.8	RSPI 位速率寄存器 (SPBR)	903
29.2.9	RSPI 数据控制寄存器 (SPDCR)	904
29.2.10	RSPI 时钟延迟寄存器 (SPCKD)	905
29.2.11	RSPI 从属选择无效延迟寄存器 (SSLND)	906
29.2.12	RSPI 下次存取延迟寄存器 (SPND)	907
29.2.13	RSPI 控制寄存器 2 (SPCR2)	908
29.2.14	RSPI 命令寄存器 0 ~ 7 (SPCMD0 ~ SPCMD7)	909
29.3	运行说明	912
29.3.1	RSPI 运行概要	912
29.3.2	RSPI 引脚的控制	913
29.3.3	RSPI 系统结构例子	914
29.3.4	数据格式	920
29.3.5	传送格式	929
29.3.6	通信运行模式	931
29.3.7	发送缓冲器空中断 / 接收缓冲器满中断	933
29.3.8	错误检测	934
29.3.9	RSPI 的初始化	937
29.3.10	SPI 运行	938

29.3.11	时钟同步运行	952
29.3.12	主控模式的运行	952
29.3.13	从属模式的运行	956
29.3.14	环回模式	958
29.3.15	奇偶校验位功能的自诊断	959
29.3.16	中断源	960
29.4	使用时的注意事项	960
29.4.1	模块停止功能的设定	960
29.4.2	低功耗功能的注意事项	960
29.4.3	有关开始通信的注意事项	960
30.	CRC 运算器 (CRC)	961
30.1	概要	961
30.2	寄存器说明	962
30.2.1	CRC 控制寄存器 (CRCCR)	962
30.2.2	CRC 数据输入寄存器 (CRCDIR)	962
30.2.3	CRC 数据输出寄存器 (CRCDOR)	963
30.3	CRC 运算器的运行说明	964
30.4	使用时的注意事项	967
30.4.1	模块停止功能的设定	967
30.5	传送时的注意事项	967
31.	12 位 A/D 转换器 (S12ADB)	968
31.1	概要	968
31.2	寄存器说明	973
31.2.1	A/D 数据寄存器 y (ADDRy) (y=0 ~ 7)、A/D 数据双重化寄存器 (ADDBLDR)、 A/D 数据双重化寄存器 A (ADDBLDRA)、A/D 数据双重化寄存器 B (ADDBLDRB)	973
31.2.2	A/D 自诊断数据寄存器 (ADRD)	976
31.2.3	A/D 控制寄存器 (ADCSR)	977
31.2.4	A/D 通道选择寄存器 A (ADANSA)	980
31.2.5	A/D 通道选择寄存器 B (ADANSB)	980
31.2.6	A/D 转换值加法运算模式选择寄存器 (ADADS)	981
31.2.7	A/D 转换值加法运算次数选择寄存器 (ADADC)	982
31.2.8	A/D 控制扩展寄存器 (ADCER)	983
31.2.9	A/D 开始触发选择寄存器 (ADSTRGR)	985
31.2.10	A/D 采样状态寄存器 n (ADSSTRn) (n=0 ~ 7)	988
31.2.11	采样 & 保持电路控制寄存器 (ADSHCR)	989
31.2.12	A/D 分组扫描优先控制寄存器 (ADGSPCR)	990
31.2.13	比较器运行模式选择寄存器 0 (ADCMPMD0)	992
31.2.14	比较器运行模式选择寄存器 1 (ADCMPMD1)	993
31.2.15	比较器的滤波器模式寄存器 0 (ADCMPNR0)	994
31.2.16	比较器检测标志寄存器 (ADCMPFR)	995
31.2.17	比较器中断选择寄存器 (ADCMPSEL)	996
31.3	运行说明	997
31.3.1	扫描的运行说明	997
31.3.2	1 个周期扫描模式	997
31.3.3	连续扫描模式	1007
31.3.4	分组扫描模式	1011
31.3.5	模拟输入的采样和扫描转换时间	1021
31.3.6	寄存器的自动清除功能的使用例子	1023
31.3.7	A/D 转换值加法运算功能	1023
31.3.8	模拟引脚放电功能	1023

31.3.9	通过异步触发开始的 A/D 转换	1023
31.3.10	通过外围模块的同步触发开始的 A/D 转换	1024
31.3.11	窗口比较器	1024
31.4	中断源和 DTC、DMAC 传送请求	1026
31.4.1	扫描结束时的中断请求	1026
31.4.2	比较器检测时的中断请求	1026
31.5	A/D 转换精度的定义	1027
31.6	使用时的注意事项	1028
31.6.1	读数据寄存器的注意事项	1028
31.6.2	停止 A/D 转换时的注意事项	1028
31.6.3	开始和强制停止 A/D 转换时的运行时序	1028
31.6.4	扫描结束中断处理的注意事项	1029
31.6.5	模块停止功能的设定	1029
31.6.6	向低功耗状态转移时的注意事项	1029
31.6.7	有关容许信号源阻抗	1029
31.6.8	对绝对精度的影响	1030
31.6.9	模拟电源引脚以外的设定范围	1030
31.6.10	设计电路板时的注意事项	1030
31.6.11	噪声对策的注意事项	1031
32.	数据运算电路 (DOC)	1032
32.1	概要	1032
32.2	寄存器说明	1033
32.2.1	DOC 控制寄存器 (DOCR)	1033
32.2.2	DOC 数据输入寄存器 (DODIR)	1034
32.2.3	DOC 数据设定寄存器 (DODSR)	1034
32.3	运算说明	1035
32.3.1	数据比较模式	1035
32.3.2	数据加法运算模式	1035
32.3.3	数据减法运算模式	1036
32.4	中断请求	1037
32.5	使用时的注意事项	1037
32.5.1	模块停止功能的设定	1037
33.	RAM	1038
33.1	概要	1038
33.2	运行说明	1038
33.2.1	低功耗功能	1038
34.	ROM (保存代码的闪存)	1039
34.1	概要	1039
34.2	寄存器说明	1041
34.2.1	闪存编程/擦除保护寄存器 (FWEPROR)	1041
34.2.2	闪存模式寄存器 (FMODR)	1042
34.2.3	闪存存取状态寄存器 (FASTAT)	1043
34.2.4	闪存存取错误中断允许寄存器 (FAEINT)	1045
34.2.5	闪存状态寄存器 0 (FSTATR0)	1046
34.2.6	闪存状态寄存器 1 (FSTATR1)	1048
34.2.7	闪存就绪中断允许寄存器 (FRDYIE)	1049
34.2.8	闪存 P/E 模式入口寄存器 (FENTRYR)	1050
34.2.9	闪存保护寄存器 (FPROTR)	1051
34.2.10	闪存复位寄存器 (FRESETR)	1052

34.2.11	FCU 命令寄存器 (FCMDR)	1053
34.2.12	FCU 处理转换寄存器 (FCPSR)	1054
34.2.13	闪存 P/E 状态寄存器 (FPESTAT)	1054
34.2.14	外围时钟通知寄存器 (PCKAR)	1055
34.3	ROM 的存储区结构	1056
34.4	块结构	1056
34.5	ROM 相关的运行模式	1057
34.6	ROM 的编程 / 擦除	1059
34.6.1	FCU 的模式	1059
34.6.2	FCU 命令一览表	1061
34.6.3	FCU 的模式和命令的关系	1063
34.6.4	FCU 命令的使用方法	1064
34.7	挂起	1079
34.7.1	编程时的挂起	1079
34.7.2	编程和擦除时的挂起 (挂起优先模式)	1080
34.7.3	擦除时的挂起 (擦除优先模式)	1081
34.8	保护	1082
34.8.1	软件保护	1082
34.8.2	命令锁定状态	1082
34.9	引导模式	1084
34.9.1	系统结构	1084
34.9.2	引导模式的状态转移	1085
34.9.3	位速率的自动调整	1087
34.9.4	ID 码保护 (引导模式)	1088
34.9.5	查询 / 设定命令的等待	1089
34.9.6	ID 码等待状态	1099
34.9.7	编程 / 擦除主机命令等待状态	1100
34.10	on-chip 调试器的 ID 码保护	1107
34.11	使用时的注意事项	1108
35.	E2 数据闪存 (保存数据的闪存)	1110
35.1	概要	1110
35.2	寄存器说明	1112
35.2.1	闪存模式寄存器 (FMDR)	1112
35.2.2	闪存存取状态寄存器 (FASTAT)	1113
35.2.3	闪存存取错误中断允许寄存器 (FAEINT)	1115
35.2.4	E2 数据闪存读允许寄存器 0 (DFLRE0)	1116
35.2.5	E2 数据闪存编程 / 擦除允许寄存器 0 (DFLWE0)	1117
35.2.6	闪存 P/E 模式入口寄存器 (FENTRYR)	1118
35.2.7	E2 数据闪存空白检查控制寄存器 (DFLBCCNT)	1119
35.2.8	E2 数据闪存空白检查状态寄存器 (DFLBCSTAT)	1119
35.3	E2 数据闪存的存储区结构	1120
35.4	块结构	1120
35.5	E2 数据闪存相关的运行模式	1121
35.6	E2 数据闪存的编程 / 擦除	1122
35.6.1	FCU 的模式	1122
35.6.2	FCU 命令一览表	1124
35.6.3	FCU 的模式和命令的关系	1125
35.6.4	FCU 命令的使用方法	1126
35.7	保护	1130
35.7.1	软件保护	1130
35.7.2	错误保护	1130

35.8	引导模式	1132
35.8.1	查询设定命令	1132
35.8.2	编程 / 擦除命令	1133
35.9	使用时的注意事项	1134
36.	电特性	1135
36.1	绝对最大额定值	1135
36.2	DC 特性	1136
36.3	AC 特性	1138
36.3.1	时钟时序	1138
36.3.2	复位时序	1140
36.3.3	从低功耗状态的恢复时序	1141
36.3.4	控制信号时序	1142
36.3.5	内部外围模块时序	1143
36.4	A/D 转换特性	1152
36.5	上电复位电路和电压检测电路的特性	1153
36.6	振荡停止检测时序	1155
36.7	ROM（保存代码的闪存）特性	1156
36.8	E2 数据闪存特性	1157
附录	1159
附录 1.	各运行模式中的端口状态	1159
附录 2.	封装尺寸图	1160

RX63T 群

瑞萨 32 位单片机

R01UH0238CJ0100

Rev.1.00

2012.07.31

100MHz、32 位 RX MCU、内部 FPU、165DMIPS、
2 位 ADC（3 个 S/H 电路、双数据寄存器、比较器）、
100MHz PWM（三相互补 2ch+ 单相互补 4ch 或者三相互补 3ch+ 单相互补 1ch）

特点

■ 内置 32 位 RX CPU 内核

- 最大工作频率：100MHz
165 DMIPS 的性能（以 100MHz 运行时）
- 32 位单精度浮点（以 IEEE754 为标准）
- 2 种乘加运算器（存储器之间、寄存器之间）
- 32 位乘法器（最快用 1 个时钟执行）
- 除法器（最快用 2 个时钟执行）
- 高速中断
- 5 段流水线的 CISC 哈佛体系结构
- 可变长度指令格式：大幅度地缩短编码
- 对应存储器保护单元（MPU）
- JTAG 和 FINE（双线式）两种调试接口

■ 低功耗功能

- 以 2.7V ~ 3.6V 运行的单电源
- 4 种低功耗模式

■ 内部主闪存（无等待）

- 以 100MHz 运行、读周期为 10ns（无等待）
- 64K/48K/32K 字节的容量
- 板上以及板外的用户编程

■ 内部数据闪存

- 8K 字节（能擦除 100K 次）
- 通过 Back Ground Operation（BGO）进行编程/擦除

■ 内部 SRAM（无等待）

- 8K 字节的 SRAM
- 用于操作数和指令
- 能在深度待机模式中进行备用

■ DMA

- DMA：内置 4 个通道
- DTC

■ 复位和电源电压控制

- 内置上电复位（POR）
- 能设定低电压检测功能（LVD）

■ 时钟功能

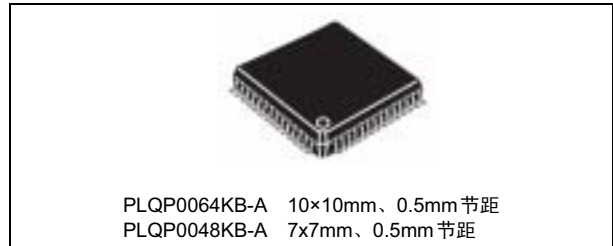
- 外部晶体振荡器和内部 PLL 支持 4MHz ~ 16MHz
- 装载内部 125kHz LOCO
- 用于 IWDT 的 125kHz LOCO 时钟
- 内置时钟频率精度测量电路（CAC）

■ 内置独立看门狗定时器

- 125kHz LOCO 时钟运行

■ 内置符合 IEC60730 标准的功能

- 振荡停止检测、频率测量功能、CRC、IWDT、A/D 自诊断等



■ 内置最多 5 个通信功能

- 支持多种能的 SCI（最多 3ch）。
可从异步模式、时钟同步模式、智能卡接口模式、简易 SPI 模式、简易 I²C 模式和扩展串行模式中选择
- I²C 总线接口 最大 400k bps 的传送（1ch）
- 装载了能进行高速通信的 RSPI（1ch）

■ 最多 16 个 16 位扩展定时器功能

- 16 位 MTU3：以 100MHz 运行、输入捕捉、输出比较、2ch 输出三相互补 PWM 波形、不对 CPU 外加负荷的互补 PWM、相位计数模式（8ch）
- 16 位 GPT：以 100MHz 运行、输入捕捉、输出比较、4ch 输出单相互补 PWM 波形或者 1ch 输出三相互补 PWM 波形 + 1ch 输出单相互补 PWM 波形、不对 CPU 外加负荷的互补 PWM、比较器联动（计数运行、PWM 无效控制）、振荡频率异常检测功能（支持 IEC60730）（4ch）
- 16 位 CMT（4ch）

■ 内置以 1MHz 运行的 12 位 A/D 转换器 8ch

- 内置采样 & 保持电路（12 位 × 最多 3ch）
- 自诊断功能（支持 IEC60730）
- ADC：3 个采样 & 保持电路、双数据寄存器、比较器（3ch）

■ 能保护重要的寄存器不被改写的寄存器写保护功能

■ 内置最多 48 个 GPIO。

- 5V 容许、漏极开路、输入上拉

■ 工作环境温度

- -40°C ~ +85°C

1. 概要

1.1 规格概要

规格概要和各封装的功能比较一览表分别如表 1.1 和表 1.2 所示。

表 1.1 的规格概要中记载了最大规格，外围模块的通道数因封装引脚数和 ROM 容量而不同，详细内容请参照“表 1.2 各封装的功能比较一览表”。

表 1.1 规格概要 (1/4)

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 基个指令：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 装载存储器保护单元（MPU）
	FPU	<ul style="list-style-type: none"> 单精度浮点数（32 位） 符合 IEEE754 规格的数据类型和浮点异常
存储器	ROM	<ul style="list-style-type: none"> 容量：64K/48K/32K 字节 100MHz、无等待存取 板上编程：2 种
	RAM	<ul style="list-style-type: none"> 容量：8K 字节 100MHz、无等待存取
	E2 数据闪存	<ul style="list-style-type: none"> 容量：8K 字节 编程 / 擦除次数：100000 次
MCU 运行模式		单芯片模式
时钟	时钟发生电路	<ul style="list-style-type: none"> 主时钟振荡器、低速内部振荡器、PLL 频率合成器、IWDG 专用内部振荡器 主时钟振荡停止检测：有 能分别设定系统时钟（ICLK）、外围模块时钟（PCLKA）、外围模块时钟（PCLKB）、FlashIF 时钟（FCLK）和 S12AD 时钟（PCLKD）。 CPU 和总线主控等系统设备与 ICLK 同步：100MHz max 多功能定时器脉冲单元 3 和通用 PWM 定时器与 PCLKA 同步：100MHz max 外围模块与 PCLKB 同步：50MHz max Flash IF 与 FCLK 同步：50MHz max 12 位 A/D 转换器与 PCLKD 同步：50MHz max
	时钟频率精度测量电路（CAC）	能监视主时钟振荡器、PLL 电路以及 IWDG 专用低速时钟振荡器的输出时钟频率的异常。
复位		引脚复位、上电复位、电压监视复位、独立看门狗定时器复位、看门狗定时器复位、深度软件待机复位、软件复位
电压检测电路		如果 VCC 经过电压检测电平（Vdet），就发生内部复位或者内部中断。

表 1.1 规格概要 (2/4)

分类	模块 / 功能	说明
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 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 (ICUb)	<ul style="list-style-type: none"> • 外围功能中断: 106 个中断源 • 外部中断: 6 个中断源 (IRQ0 ~ IRQ5 引脚) • 软件中断: 1 个中断源 • 非屏蔽中断: 6 个中断源 • 能设定 16 个中断优先级。
DMA	DMA 控制器 (DMACA)	<ul style="list-style-type: none"> • 4 个通道 • 传送模式: 正常传送模式、重复传送模式、块传送模式 • 启动源: 软件触发、外部中断、外围功能中断
	数据传送控制器 (DTCa)	<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: 设定软件中断启动寄存器、外部中断、外围功能中断
I/O 端口	可编程输入 / 输出端口	<ul style="list-style-type: none"> • 64 引脚 LQFP 输入 / 输出: 39 个 输入: 9 个 漏极开路输出: 2 个 5V 容许: 38 个 • 48 引脚 LQFP 输入 / 输出: 25 个 输入: 7 漏极开路输出: 2 个 5V 容许: 24 个
定时器	多功能定时器脉冲单元 3 (MTU3)	<ul style="list-style-type: none"> • 16 位 × 8 个通道 • 最多能输入 / 输出 16 个脉冲以及输入 3 个脉冲 • 各通道可从 10 种计数时钟 (PCLKA/1、PCLKA/4、PCLKA/16、PCLKA/64、PCLKA/256、PCLKA/1024、MTCLKA、MTCLKB、MTCLKC、MTCLKD) 中选择 8 种 (通道 1 可选择 7 种、通道 5 可选择 4 种、通道 6/ 通道 7 可选择 6 种)。 • 24 个输出比较寄存器兼输入捕捉寄存器 • 计数器清除运行 (能通过比较匹配 / 输入捕捉同时清除) • 同时写多个定时器的计数器 (TCNT) • 通过计数器的同步运行进行各寄存器的同步输入 / 输出 • 缓冲运行 • 级联运行 • 38 种中断源 • 自动传送寄存器数据 • 脉冲输出模式 交替 / PWM / 互补 PWM / 复位同步 PWM • 互补 PWM 输出模式 输出三相用于控制逆变器的非重叠波形 自动设定死区时间 能任意设定 PWM 的占空比 (0 ~ 100%) A/D 转换请求延迟功能 波峰 / 波谷中断减少功能 双缓冲功能 • 复位同步 PWM 模式 输出三相任意占空比的正相 / 反相 PWM 波形 • 相位计数模式 • 用于补偿死区时间的计数器功能 • 能生成 A/D 转换器的转换开始触发 • A/D 转换开始减少功能

表 1.1 规格概要 (3/4)

分类	模块 / 功能	说明
定时器	端口输出允许 3 (POE3)	<ul style="list-style-type: none"> 控制 MTU3/GPT 波形输出引脚的高阻抗 通过 POE0、POE8、POE10 和 POE11 这 4 个输入引脚启动 通过输出短路检测 (检测到 PWM 输出同时变为有效电平) 启动 通过比较器检测 / 振荡停止检测 / 软件启动 能通过可编程对输出控制对象引脚进行追加控制
	通用 PWM 定时器 (GPT)	<ul style="list-style-type: none"> 16 位 × 4 个通道 各计数器能选择递增计数或者递减计数 (锯齿波)、递增 / 递减计数 (三角波) 各通道可从 4 种计数时钟 (PCLKA/1、PCLKA/4、PCLKA/8、PCLKA/16) 中选择。 每个通道有 2 个输入 / 输出引脚 每个通道有 2 个用于输出比较 / 输入捕捉的寄存器 各通道分别有对应 2 个输出比较 / 输入捕捉寄存器的 4 个缓冲寄存器。在不进行缓冲运行时, 能作为比较寄存器运行。 能生成在输出比较运行时分别在波峰 / 波谷进行缓冲运行并且左右不对称的 PWM 波形。 给各通道装载用于帧周期的寄存器 (能通过上溢 / 下溢中断) 能同步运行各计数器 同步运行模式 (同时或者以任意的时序移动 (支持相移)) 能在 PWM 运行时生成死区时间 组合 3 个计数器, 能生成带死区时间的三相 PWM 波形。 能通过外部 / 内部触发开始、清除或者停止计数。 内部触发源有内部比较器输出、软件和比较匹配。 能通过主时钟的计数时钟来测量被分频的 IWDTC 专用低速时钟的边沿 (振荡异常检测)。
	比较匹配定时器 (CMT)	<ul style="list-style-type: none"> (16 位 × 2 个通道) × 2 个单元 可选择 4 种时钟 (PCLK/8、PCLK/32、PCLK/128、PCLK/512)。
	看门狗定时器 (WDTA)	<ul style="list-style-type: none"> 14 位 × 1 个通道 可选择 6 种计数时钟 (PCLK/4、PCLK/64、PCLK/128、PCLK/512、PCLK/2048、PCLK/8192)。
	独立看门狗定时器 (IWDTCa)	<ul style="list-style-type: none"> 14 位 × 1 个通道 计数时钟: IWDTC 专用内部振荡器 专用时钟 /1、专用时钟 /16、专用时钟 /32、专用时钟 /64、专用时钟 /128、专用时钟 /256
通信功能	串行通信接口 (SC1c、SC1d)	<ul style="list-style-type: none"> 3 个通道: (SC1c: 2 个通道 + SC1d: 1 个通道) SC1c <ul style="list-style-type: none"> 串行通信方式: 异步 / 时钟同步 / 智能卡接口 多处理器功能 能通过内部波特率发生器选择任意位速率 可选择 LSB first 或者 MSB first 支持简易 I²C 支持简易 SPI SC1d (给 SC1c 附加以下功能) <ul style="list-style-type: none"> 支持由起始帧和信息帧构成的串行通信协议 支持 LIN 格式
	I ² C 总线接口 (RIIC)	<ul style="list-style-type: none"> 1 个通道 通信格式 I²C 总线格式 / SMBus 格式 支持多主控 最大传送率: 400kbps

表 1.1 规格概要 (4/4)

分类	模块 / 功能	说明
通信功能	串行外围接口 (RSPI)	<ul style="list-style-type: none"> 1 个通道 RSPI 传送功能 能使用 MOSI (Master Out Slave In)、MISO (Master In Slave Out)、SSL (Slave Select)、RSPCK (RSPI Clock) 信号, 通过 SPI 运行 (4 线式) 或者时钟同步运行 (3 线式) 进行串行通信。 能在主控模式或者从属模式中进行串行通信。 数据格式 可选择 MSB first 或者 LSB first。 能将传送位长度改为 8 ~ 16 位、20 位、24 位或者 32 位。 发送 / 接收缓冲器为 128 位。 1 次发送和接收最多传送 4 帧 (1 帧最多为 32 位)。 缓冲器结构 发送 / 接收缓冲器为双缓冲结构。
12 位 A/D 转换器 (S12ADB)		<ul style="list-style-type: none"> 12 位 (8 个通道 × 1 个单元) 分辨率: 12 位 转换时间 每个通道 1.0μs (当 S12ADB 时钟 PCLKD (A/D 转换时钟 ADCLK) = 50MHz) 运行模式 扫描模式 (1 个周期扫描模式 / 连续扫描模式 / 分组扫描模式) 组 A 优先控制 (只限分组扫描模式) 采样 & 保持功能 装载了单元共同的采样 & 保持电路 还装载了专用的采样 & 保持电路 (3 个通道) 自诊断功能 能在内部生成 3 种用于自诊断功能的模拟输入电压 (VREFL0、VREFH0×1/2、VREFH0) 双触发模式 (A/D 转换数据双重化功能) 3 种 A/D 转换开始方法 软件触发、定时器 (MTU3、GPT) 的触发、外部触发 窗口比较器功能 (3 个通道)
CRC 运算器 (CRC)		<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 通信的 CRC 码, 还是生成 MSB first 通信的 CRC 码。
数据运算电路 (DOC)		<ul style="list-style-type: none"> 能对 16 位单位的任意数据进行比较、加法运算或者减法运算。
工作频率		100MHz max
电源电压		VCC=2.7 ~ 3.6V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0
工作环境温度		-40 ~ +85°C (大温度范围的规格产品)
封装		64 引脚 LQFP (PLQP0064KB-A) 48 引脚 LQFP (PLQP0048KB-A)
内部调试系统		<ul style="list-style-type: none"> E1 仿真器 (JTAG 接口和 FINE 接口) E20 仿真器 (JTAG 接口)

表 1.2 各封装的功能比较一览表

功能 封装		RX63T 群	
		64 引脚	48 引脚
DMA	DMA 控制器	ch0 ~ 3	
	数据传送控制器	有	
定时器	多功能定时器脉冲单元 3	ch0 ~ 7	
	端口输出允许 3	有 (4 个 POE 引脚)	
	通用 PWM 定时器	ch0 ~ 3	
	比较匹配定时器	ch0 ~ 3	
	看门狗定时器	有	
	独立看门狗定时器	有	
通信功能	串行通信接口 (SC1c)	ch0、ch1	
	串行通信接口 (SC1d)	ch12	
	I ² C 总线接口	ch0	
	串行外围接口	ch0	
12 位 A/D 转换器		AN000 ~ AN007	AN000 ~ AN004、AN007
时钟频率精度测量电路		有	
CRC 运算器		有	
数据运算电路		有	

1.2 产品一览表

产品一览表如表 1.3 所示，产品型号、存储容量和封装如图 1.1 所示。

表 1.3 产品一览表

群	产品型号	封装	内部 ROM 容量	内部 RAM 容量	E2 数据闪存	工作频率 (max)
RX63T	R5F563T6EDFM	PLQP0064KB-A	64K 字节	8K 字节	8K 字节	100MHz
	R5F563T5EDFM	PLQP0064KB-A	48K 字节	8K 字节	8K 字节	100MHz
	R5F563T4EDFM	PLQP0064KB-A	32K 字节	8K 字节	8K 字节	100MHz
	R5F563T6EDFL	PLQP0048KB-A	64K 字节	8K 字节	8K 字节	100MHz
	R5F563T5EDFL	PLQP0048KB-A	48K 字节	8K 字节	8K 字节	100MHz
	R5F563T4EDFL	PLQP0048KB-A	32K 字节	8K 字节	8K 字节	100MHz

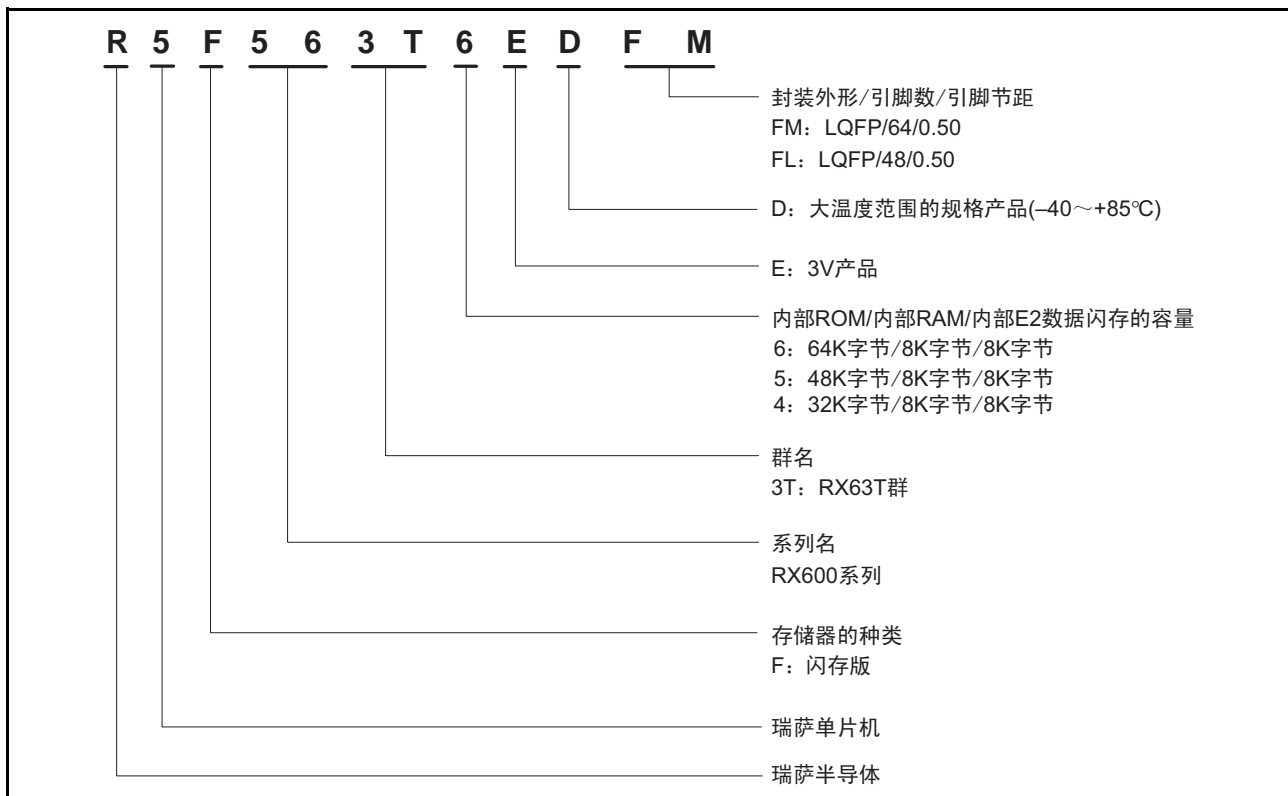


图 1.1 产品型号、存储容量和封装

1.3 框图

框图如图 1.2 所示。

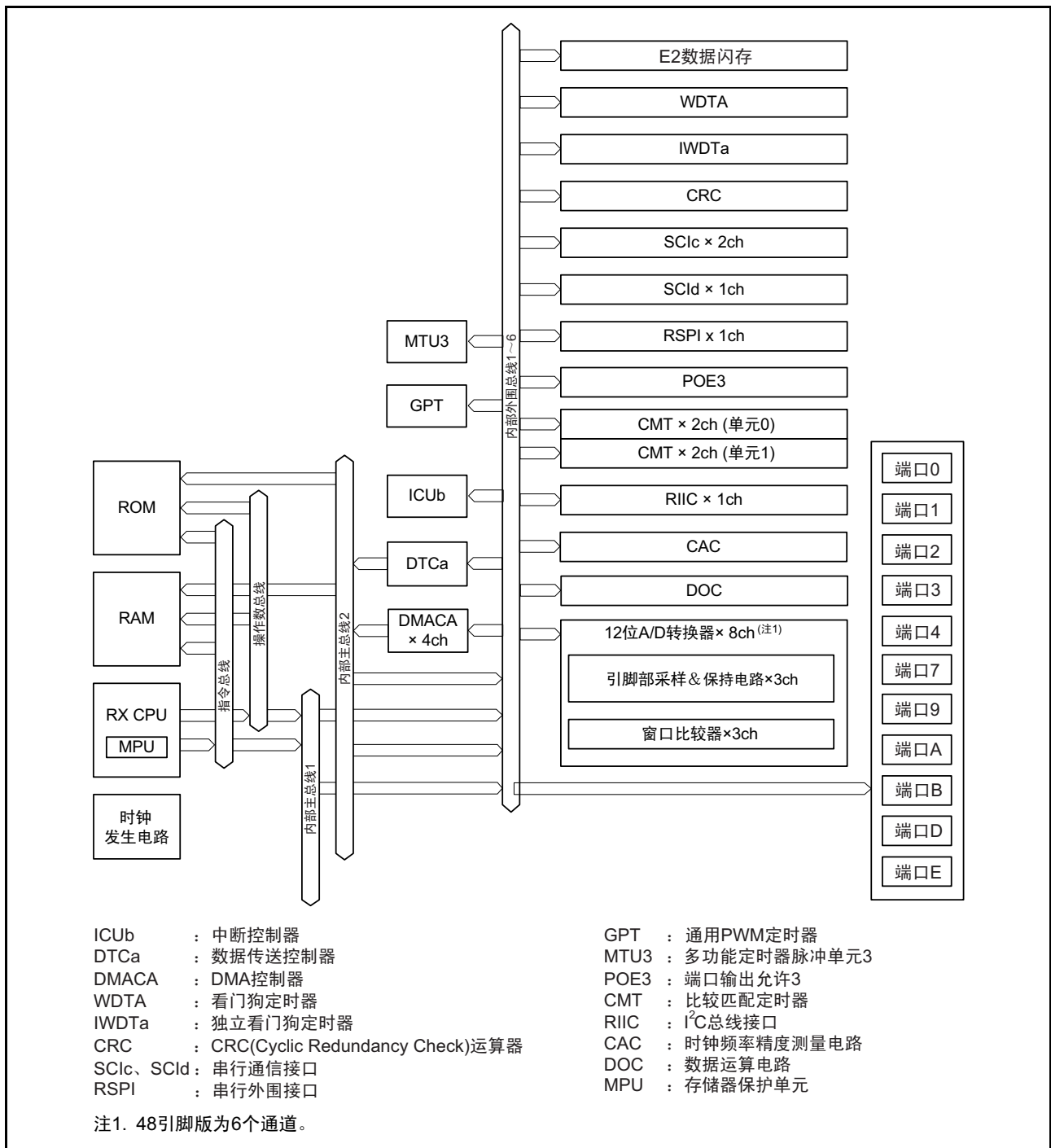


图 1.2 框图

1.4 引脚功能

引脚功能一览表如表 1.4 所示。

表 1.4 引脚功能一览表 (1/3)

分类	引脚名	输入 / 输出	功能
电源	VCC	—	电源引脚。必须连接系统电源。必须通过 0.1 μ F 的电容器连接 VSS，并且将电容器配置在引脚附近。
	VCL	—	必须通过 0.1 μ F 的电容器连接 VSS，并且将电容器配置在引脚附近。
	VSS	—	接地引脚 必须连接系统电源 (0V)。
时钟	XTAL	输出	晶体谐振器连接引脚
	EXTAL	输入	EXTAL 引脚也能输出外部时钟。
时钟频率精度测量	CACREF	输入	测量时钟频率精度的触发输入引脚
运行模式控制	MD	输入	设定运行模式。不能在运行过程中更改此引脚。
系统控制	RES#	输入	复位引脚 如果此引脚为 Low 电平，就进入复位状态。
	EMLE	输入	内部仿真器允许引脚 在使用内部仿真器时，必须置为 High 电平；在不使用内部仿真器时，必须置为 Low 电平。
内部仿真器	FINEC	输入	用于 FINE 接口的时钟引脚
	FINED	输入 / 输出	FINE 接口引脚
	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，此引脚就变为内部仿真器专用引脚。
	TDI	输入	
	TCK	输入	
	TDO	输出	
中断	NMI	输入	非屏蔽中断请求引脚
	IRQ0 ~ IRQ5	输入	中断请求引脚
多功能定时器 脉冲单元 3	MTIOC0A、MTIOC0B MTIOC0C、MTIOC0D	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 输出引脚
	MTIOC1A、MTIOC1B	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 输出引脚
	MTIOC2A、MTIOC2B	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 输出引脚
	MTIOC3A、MTIOC3B MTIOC3C、MTIOC3D	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 输出引脚
	MTIOC4A、MTIOC4B MTIOC4C、MTIOC4D	输入 / 输出	TGRA4 ~ TGRD4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚
	MTIC5U、MTIC5V MTIC5W	输入	TGRU5、TGRV5 和 TGRW5 的输入捕捉的输入引脚 / 死区时间补偿功能的输入引脚
	MTIOC6A、MTIOC6B MTIOC6C、MTIOC6D	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 输出引脚
	MTIOC7A、MTIOC7B MTIOC7C、MTIOC7D	输入 / 输出	TGRA7 ~ TGRD7 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚
	MTCLKA、MTCLKB MTCLKC、MTCLKD	输入	输入外部时钟
	端口输出允许 3	POE0#、POE8# POE10#、POE11#	输入

表 1.4 引脚功能一览表 (2/3)

分类	引脚名	输入 / 输出	功能	
通用 PWM 定时器	GTIOC0A、GTIOC0B	输入 / 输出	GPT0.GTGRA 和 GPT0.GTGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚	
	GTIOC1A、GTIOC1B	输入 / 输出	GPT1.GTGRA 和 GPT1.GTGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚	
	GTIOC2A、GTIOC2B	输入 / 输出	GPT2.GTGRA 和 GPT2.GTGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚	
	GTIOC3A、GTIOC3B	输入 / 输出	GPT3.GTGRA 和 GPT3.GTGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚	
	GTETRG	输入	用于 GPT 的外部触发输入引脚	
串行通信接口 (SC1c)	• 异步模式 / 时钟同步模式			
	SCK0、SCK1	输入 / 输出	时钟的输入 / 输出引脚	
	RXD0、RXD1	输入	接收数据的输入引脚	
	TXD0、TXD1	输出	发送数据的输出引脚	
	CTS0#、CTS1#	输入	用于控制发送 / 接收开始的输入引脚	
	RTS0#、RTS1#	输出	用于控制发送 / 接收开始的输出引脚	
	• 简易 I ² C 模式			
	SSCL0、SSCL1	输入 / 输出	I ² C 时钟的输入 / 输出引脚	
	SSDA0、SSDA1	输入 / 输出	I ² C 数据的输入 / 输出引脚	
	• 简易 SPI 模式			
	SCK0、SCK1	输入 / 输出	时钟的输入 / 输出引脚	
	SMISO0、SMISO1	输入 / 输出	从属发送数据的输入 / 输出引脚	
	SMOSI0、SMOSI1	输入 / 输出	主控发送数据的输入 / 输出引脚	
	SS0#、SS1#	输入	片选输入引脚	
串行通信接口 (SC1d)	• 异步模式 / 时钟同步模式			
	SCK12	输入 / 输出	时钟的输入 / 输出引脚	
	RXD12	输入	接收数据的输入引脚	
	TXD12	输出	发送数据的输出引脚	
	CTS12#	输入	用于控制发送 / 接收开始的输入引脚	
	RTS12#	输出	用于控制发送 / 接收开始的输出引脚	
	• 简易 I ² C 模式			
	SSCL12	输入 / 输出	I ² C 时钟的输入 / 输出引脚	
	SSDA12	输入 / 输出	I ² C 数据的输入 / 输出引脚	
	• 简易 SPI 模式			
	SCK12	输入 / 输出	时钟的输入 / 输出引脚	
	SMISO12	输入 / 输出	从属发送数据的输入 / 输出引脚	
	SMOSI12	输入 / 输出	主控发送数据的输入 / 输出引脚	
	SS12#	输入	片选输入引脚	
	• 扩展串行模式			
	RDX12	输入	接收数据的输入引脚	
	TXDX12	输出	发送数据的输出引脚	
	SIOX12	输入 / 输出	发送 / 接收数据的输入 / 输出引脚	
	I ² C 总线接口	SCL	输入 / 输出	时钟的输入 / 输出引脚 能通过 N 沟道漏极开路直接驱动总线。
		SDA	输入 / 输出	数据的输入 / 输出引脚 能通过 N 沟道漏极开路直接驱动总线。

表 1.4 引脚功能一览表 (3/3)

分类	引脚名	输入 / 输出	功能
串行外围接口	RSPCKA	输入 / 输出	时钟的输入 / 输出引脚
	MOSIA	输入 / 输出	主控发送数据的输入 / 输出引脚
	MISOA	输入 / 输出	从属发送数据的输入 / 输出引脚
	SSLA0	输入 / 输出	从属选择的输入 / 输出引脚
	SSLA1 ~ SSLA3	输出	从属选择的输出引脚
12 位 A/D 转换器	AN000 ~ AN007	输入	A/D 转换器的模拟输入引脚 48 引脚版中无 AN005 引脚和 AN006 引脚。
	ADTRG0#	输入	用于开始 A/D 转换的外部触发输入引脚
	CVREFH	输入	比较器的高电平基准电压输入引脚
	CVREFL	输入	比较器的低电平基准电压输入引脚
模拟电源	AVCC0	—	12 位 A/D 转换器的模拟电源引脚 在不使用 12 位 A/D 转换器时，必须连接 VCC。
	AVSS0	—	12 位 A/D 转换器的模拟接地引脚 在不使用 12 位 A/D 转换器时，必须连接 VSS。
	VREFH0	—	12 位 A/D 转换器的基准电源引脚 在不使用 12 位 A/D 转换器时，必须连接 VCC。
	VREFL0	—	12 位 A/D 转换器的基准接地引脚 在不使用 12 位 A/D 转换器时，必须连接 VSS。
I/O 端口	P00、P01	输入 / 输出	2 位输入 / 输出引脚 48 引脚版中无此引脚。
	P10、P11	输入 / 输出	2 位输入 / 输出引脚 48 引脚版中无此引脚。
	P22 ~ P24	输入 / 输出	3 位输入 / 输出引脚
	P30 ~ P33	输入 / 输出	4 位输入 / 输出引脚 48 引脚版中无 P31 ~ P33 引脚。
	P40 ~ P47	输入	8 位输入引脚 48 引脚版中无 P45、P46 引脚。
	P70 ~ P76	输入 / 输出	7 位输入 / 输出引脚
	P91 ~ P94	输入 / 输出	4 位输入 / 输出引脚 48 引脚版中无此引脚。
	PA2 ~ PA5	输入 / 输出	4 位输入 / 输出引脚 48 引脚版中无 PA4、PA5 引脚。
	PB0 ~ PB7	输入 / 输出	8 位输入 / 输出引脚 48 引脚版中无 PB7 引脚。
	PD3 ~ PD7	输入 / 输出	5 位输入 / 输出引脚
	PE2	输入	1 位输入引脚

1.5 引脚排列图

引脚排列图如图 1.3、图 1.4 所示，各功能引脚一览表如表 1.5、表 1.6 所示。

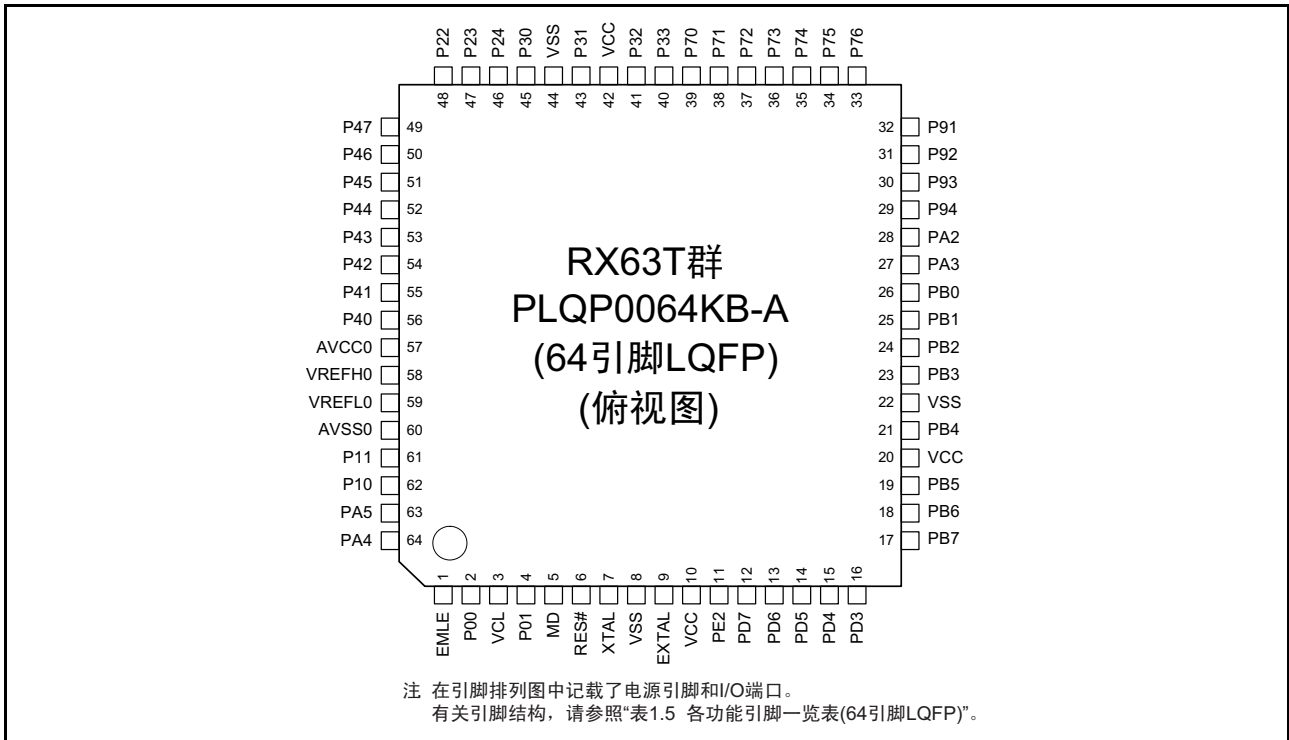


图 1.3 引脚排列图 (64 引脚 LQFP)

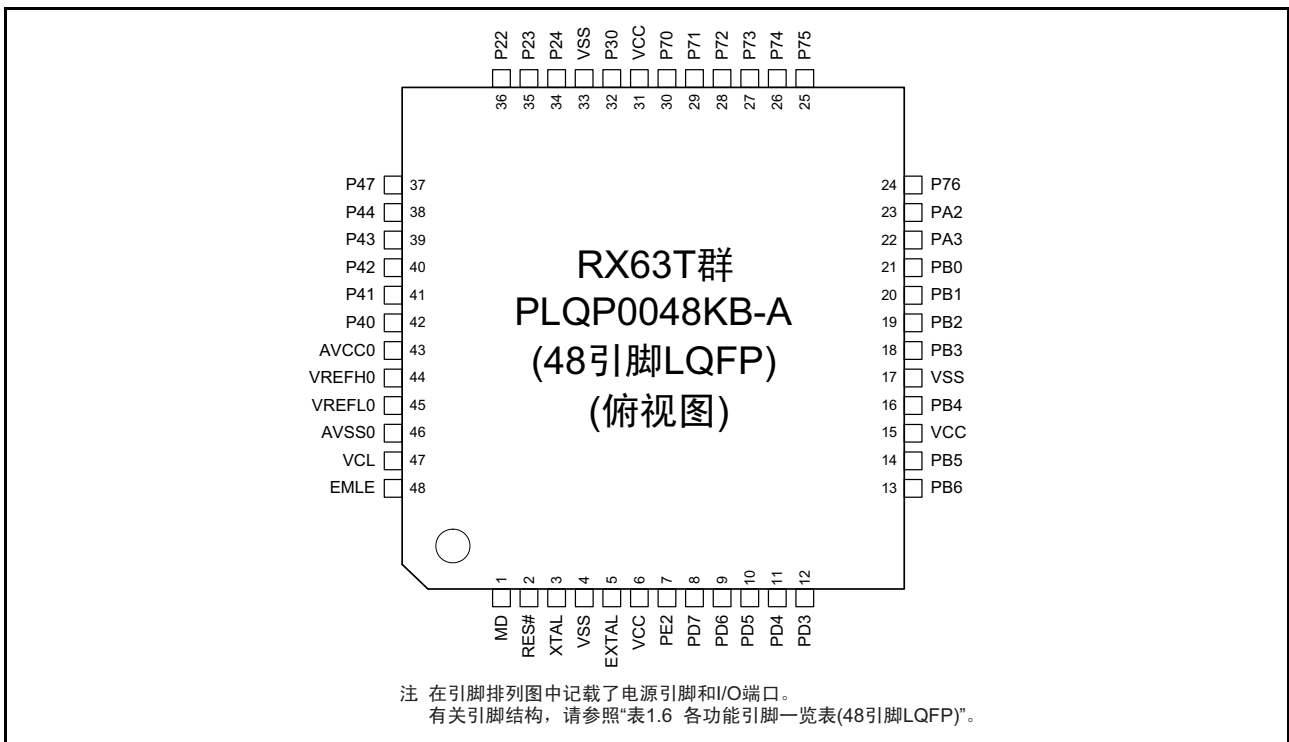


图 1.4 引脚排列图 (48 引脚 LQFP)

表 1.5 各功能引脚一览表 (64 引脚 LQFP) (1/3)

引脚序号 64 引脚 LQFP	电源时钟 系统控制	I/O 端口	POE3	定时器 (MTU3、GPT、CAC)	通信		中断	S12ADB
					(SCIc、SCId)	(RSPI、RIIC)		
1	EMLE							
2		P00		GTIOC3A	CTS0# RTS0# SS0#		IRQ2-DS	
3	VCL							
4		P01		GTIOC3B CACREF			IRQ4-DS	
5	MD FINED							
6	RES#							
7	XTAL							
8	VSS							
9	EXTAL							
10	VCC							
11		PE2	POE10#				NMI	
12	TRST#	PD7		GTIOC0A	CTS0# RTS0# SS0#			
13	TMS	PD6		GTIOC0B				
14	TDI	PD5		GTIOC1A	RXD1 SMISO1 SSCL1			
15	TCK FINEC	PD4		GTIOC1B	SCK1			
16	TDO	PD3		GTIOC2A	TXD1 SMOSI1 SSDA1			
17		PB7		GTIOC2B	SCK12			
18		PB6		GTIOC2B	RXD12 SMISO12 SSCL12 RXDX12			
19		PB5	POE11#		TXD12 SMOSI12 SSDA12 TXDX12 SIOX12		IRQ0	
20	VCC							
21		PB4	POE8#	GTETRGR	CTS12# RTS12# SS12#		IRQ3-DS	
22	VSS							
23		PB3		MTIOC0A MTCLKA CACREF	SCK0			

表 1.5 各功能引脚一览表 (64 引脚 LQFP) (2/3)

引脚序号 64 引脚 LQFP	电源时钟 系统控制	I/O 端口	POE3	定时器 (MTU3、GPT、CAC)	通信		中断	S12ADB
					(SCIc、SCId)	(RSPI、RIIC)		
24		PB2		MTIOC0B MTCLKB	TXD0 SMOSI0 SSDA0	SDA		
25		PB1		MTIOC0C	RXD0 SMISO0 SSCL0	SCL		
26		PB0		MTIOC0D		MOSIA		
27		PA3		MTIOC2A		SSLA0		
28		PA2		MTIOC2B		SSLA1		
29		P94			TXD1 SMOSI1 SSDA1			
30		P93			RXD1 SMISO1 SSCL1		IRQ1	
31		P92			SCK1			
32		P91			CTS1# RTS1# SS1#			
33		P76		MTIOC4D GTIOC2B				
34		P75		MTIOC4C GTIOC1B				
35		P74		MTIOC3D GTIOC0B				
36		P73		MTIOC4B GTIOC2A				
37		P72		MTIOC4A GTIOC1A				
38		P71		MTIOC3B GTIOC0A				
39		P70	POE0#		CTS1# RTS1# SS1#		IRQ5-DS	
40		P33		MTIOC3A		SSLA3		
41		P32		MTIOC3C		SSLA2		
42	VCC							
43		P31		MTIOC0A		SSLA1		
44	VSS							
45		P30		MTIOC0B MTCLKD	TXD0 SMOSI0 SSDA0	SSLA0		
46		P24		MTIC5U MTCLKC	RXD0 SMISO0 SSCL0	RSPCKA		

表 1.5 各功能引脚一览表 (64 引脚 LQFP) (3/3)

引脚序号 64 引脚 LQFP	电源时钟 系统控制	I/O 端口	POE3	定时器 (MTU3、GPT、CAC)	通信		中断	S12ADB
					(SCIc、SCId)	(RSPI、RIIC)		
47		P23		MTIC5V MTCLKB CACREF	SCK0	MOSIA		
48		P22		MTIC5W MTCLKA	CTS0# RTS0# SS0#	MISOA		
49		P47						AN007 CVREFH
50		P46						AN006
51		P45						AN005
52		P44						AN004
53		P43						AN003 CVREFL
54		P42						AN002
55		P41						AN001
56		P40						AN000
57	AVCC0							
58	VREFH0							
59	VREFL0							
60	AVSS0							
61		P11		MTCLKC			IRQ1-DS	
62		P10		MTCLKD			IRQ0-DS	
63		PA5		MTIOC1A		MISOA		
64		PA4		MTIOC1B		RSPCKA		ADTRG0#

表 1.6 各功能引脚一览表 (48 引脚 LQFP) (1/2)

引脚序号 48 引脚 LQFP	电源时钟 系统控制	I/O 端口	POE3	定时器 (MTU3、GPT、CAC)	通信		中断	S12ADB
					(SClC、SCId)	(RSPI、RIIC)		
1	MD FINED							
2	RES#							
3	XTAL							
4	VSS							
5	EXTAL							
6	VCC							
7		PE2	POE10#				NMI	
8	TRST#	PD7		GTIOC0A	CTS0# RTS0# SS0#			
9	TMS	PD6		GTIOC0B				
10	TDI	PD5		GTIOC1A	RXD1 SMISO1 SSCL1			
11	TCK FINEC	PD4		GTIOC1B	SCK1			
12	TDO	PD3		GTIOC2A	TXD1 SMOS11 SSDA1			
13		PB6		GTIOC2B	RXD12 SMISO12 SSCL12 RXDX12			
14		PB5	POE11#		TXD12 SMOS112 SSDA12 TXDX12 SIOX12		IRQ0	
15	VCC							
16		PB4	POE8#	GTETRG	CTS12# RTS12# SS12#		IRQ3-DS	
17	VSS							
18		PB3		MTIOC0A MTCLKA CACREF	SCK0			
19		PB2		MTIOC0B MTCLKB	TXD0 SMOS10 SSDA0	SDA		
20		PB1		MTIOC0C	RXD0 SMISO0 SSCL0	SCL		
21		PB0		MTIOC0D	SCK12	MOSIA		
22		PA3		MTIOC2A		SSLA0		
23		PA2		MTIOC2B		SSLA1		
24		P76		MTIOC4D GTIOC2B				

表 1.6 各功能引脚一览表（48 引脚 LQFP）(2/2)

引脚序号 48 引脚 LQFP	电源时钟 系统控制	I/O 端口	POE3	定时器 (MTU3、GPT、CAC)	通信		中断	S12ADB
					(SC1c、SC1d)	(RSPI、RIIC)		
25		P75		MTIOC4C GTIOC1B				
26		P74		MTIOC3D GTIOC0B				
27		P73		MTIOC4B GTIOC2A				
28		P72		MTIOC4A GTIOC1A				
29		P71		MTIOC3B GTIOC0A				
30		P70	POE0#		CTS1# RTS1# SS1#		IRQ5-DS	
31	VCC							
32		P30		MTIOC0B MTCLKD	TXD0 SMOSI0 SSDA0	SSLA0		
33	VSS							
34		P24		MTIC5U MTCLKC	RXD0 SMISO0 SSCL0	RSPCKA		
35		P23		MTIC5V MTCLKB	DCK0	MOSIA		
36		P22		MTIC5W MTCLKA	CTS0# RTS0# SS0#	MISOA		
37		P47						AN007 CVREFH
38		P44						AN004
39		P43						AN003 CVREFL
40		P42						AN002
41		P41						AN001
42		P40						AN000
43	AVCC0							
44	VREFH0							
45	VREFL0							
46	AVSS0							
47	VCL							
48	EMLE							

2. CPU

RX63T 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。如果因存储器的存取引起流水线的延长，就可能先执行后面的运算。RX CPU 采用“Out-of-Order-Completion”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“Out-of-Order-Completion”
- 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
- 存储器保护单元
- 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 个用于 DSP 功能指令的累加器。

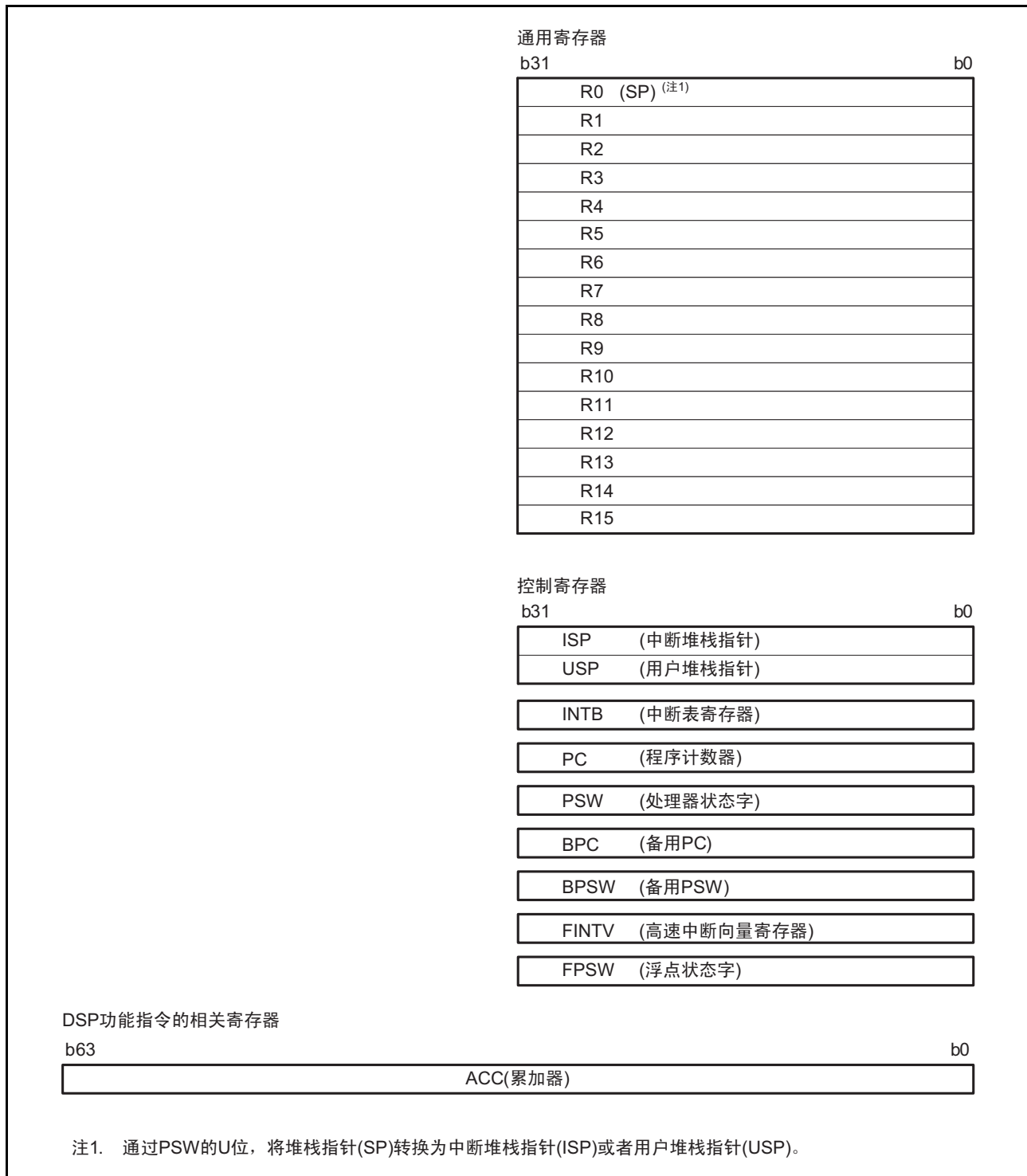


图 2.1 CPU 寄存器组

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。

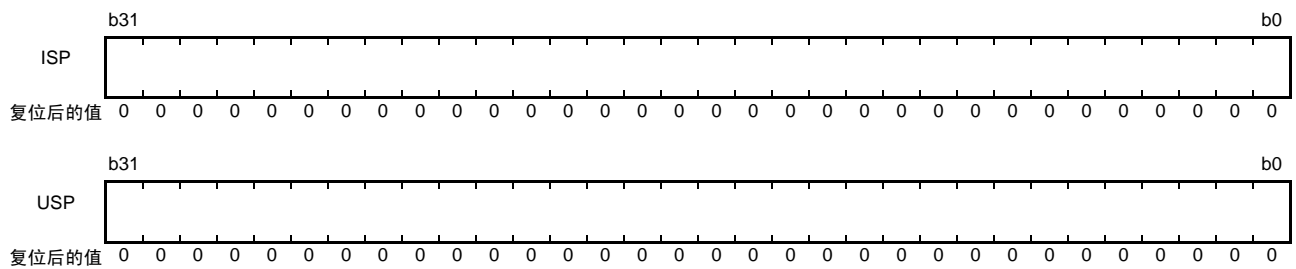
通用寄存器 R0 除了具有通用寄存器的功能以外，还有堆栈指针 (SP) 的功能。通过处理器状态字 (PSW) 的堆栈指针指定位 (U)，将 SP 转换为中断堆栈指针 (ISP) 或者用户堆栈指针 (USP)。

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)

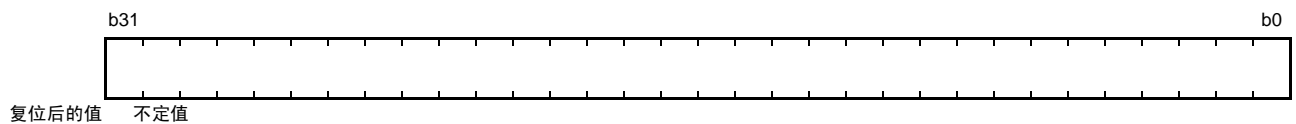
2.2.2.1 中断堆栈指针 (ISP) / 用户堆栈指针 (USP)



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)。

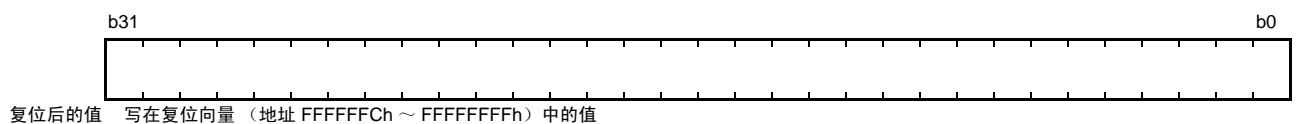
如果给 ISP 和 USP 设定 4 的倍数，带有堆栈操作的指令和中断响应顺序的周期数就会变短。

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)



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
b27-b24	IPL[3:0] (注1)	处理器中断优先级	b27 b24 0000: 0级 (最低) 0001: 1级 0010: 2级 0011: 3级 0100: 4级 0101: 5级 0110: 6级 0111: 7级 1000: 8级 1001: 9级 1010: 10级 1011: 11级 1100: 12级 1101: 13级 1110: 14级 1111: 15级 (最高)	R/W
b31-b28	—	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[3:0] 位、PM 位、U 位和 I 位的写操作。另外, 当用 MVTIPL 指令写 IPL[3:0] 位时, 发生特权指令异常。

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管理模式时, 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。

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”。

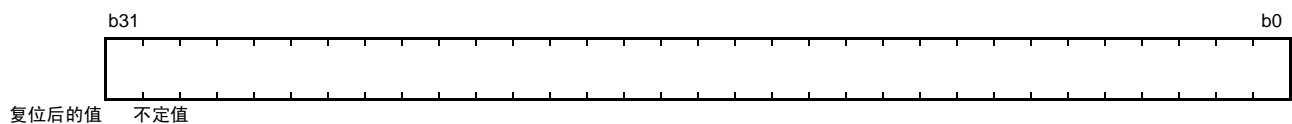
PM 位（处理器模式设定位）

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[3:0] 位（处理器中断优先级）

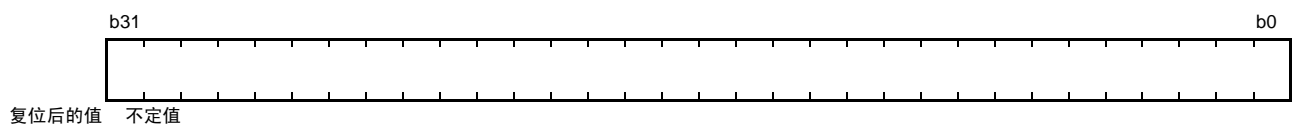
IPL[3:0] 位指定 0 级（最低）～ 15 级（最高）的 16 个处理器中断优先级。如果发生请求的中断的优先级高于处理器中断优先级，就允许该中断。在将 IPL[3:0] 位设定为“15 级”（Fh）时，禁止全部的中断。如果发生非屏蔽中断，IPL[3:0] 位就变为“15 级”（Fh）。如果发生中断，这些位就为接受中断的优先级。

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。

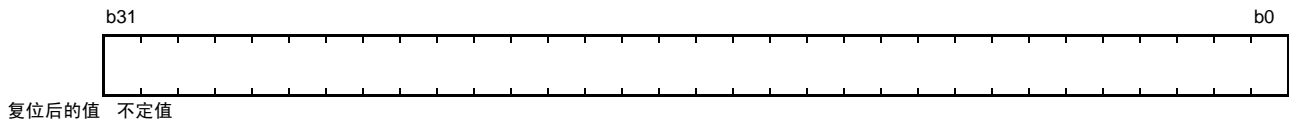
2.2.2.6 备用 PSW（BPSW）



备用 PSW（BPSW）是为实现中断响应高速化而设计的寄存器。

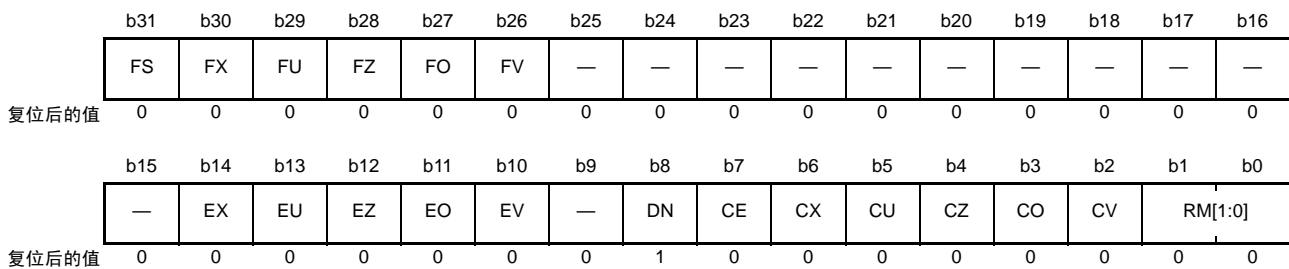
如果发生高速中断，就将处理器状态字（PSW）的内容保存到 BPSW。BPSW 的位配置与 PSW 对应。

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)



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 标志是累积型标志 ($j=X、U、Z、O、V$)。

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）

如果发生 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）和非安装处理，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 MVTC 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）

在因执行浮点运算指令而发生了 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”。

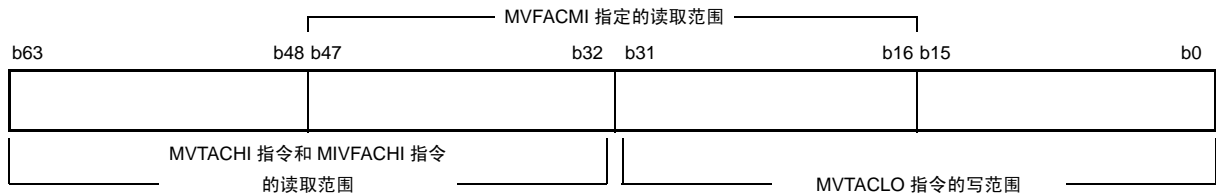
- 当 E_j 为“1”（允许异常处理）时，此标志不变。
- 如果该标志变为“1”，就在通过软件置“0”前一直保持“1”（累积标志）。

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。

2.2.3 DSP 功能指令的相关寄存器

2.2.3.1 累加器（ACC）



复位后的值 不定值

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[3:0]、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、MVTIPL、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.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.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

通过在将压栈的 PSW.PM 位置“1”后执行 RTE 指令，或者在将保存到备用 PSW（BPSW）中的 PSW.PM 位置“1”后执行 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 的补码表现。

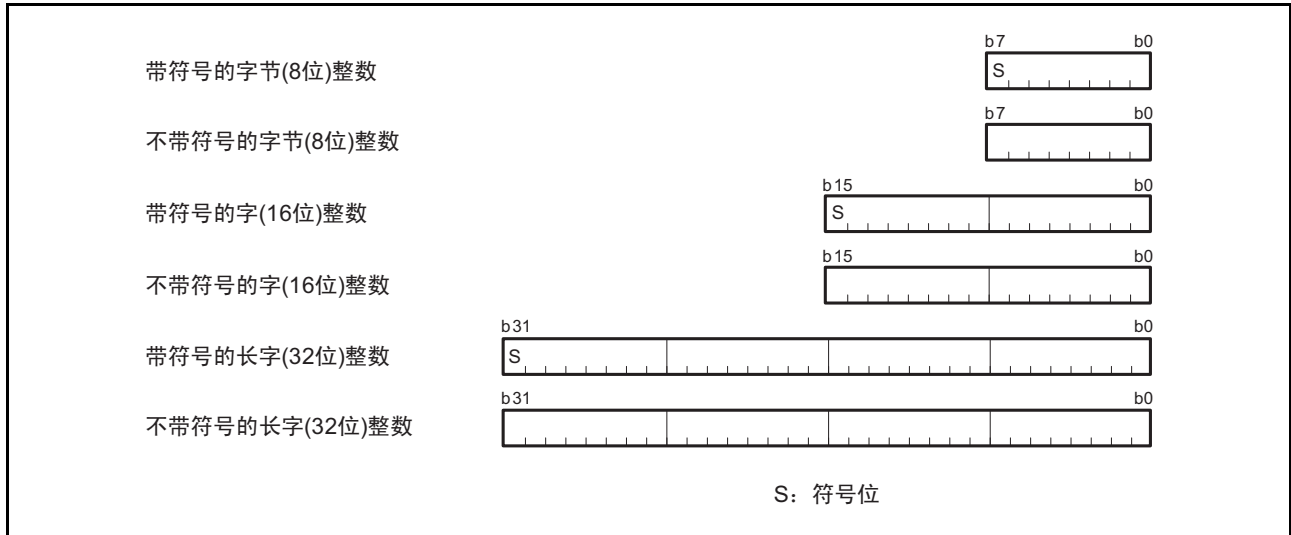


图 2.2 整数

2.4.2 浮点数

浮点数符合 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 和 ROUND 共 8 种指令。

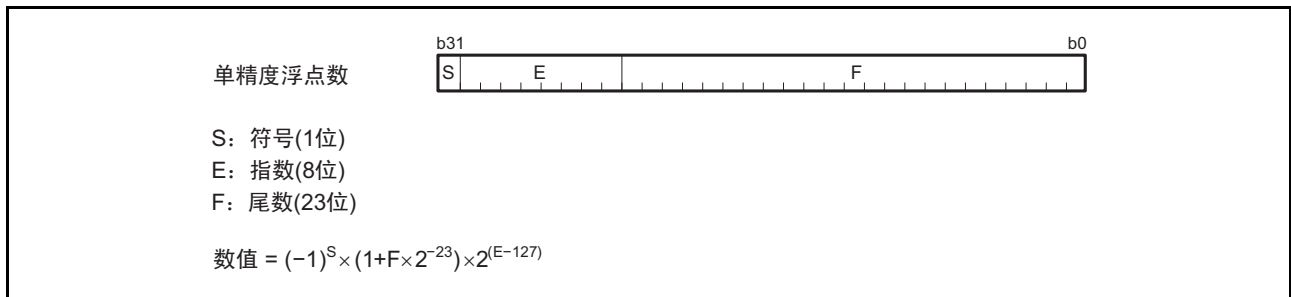


图 2.3 浮点数

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

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。

2.4.3 位

位用于位操作指令 **BCLR**、**BMCnd**、**BNOT**、**BSET** 和 **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。

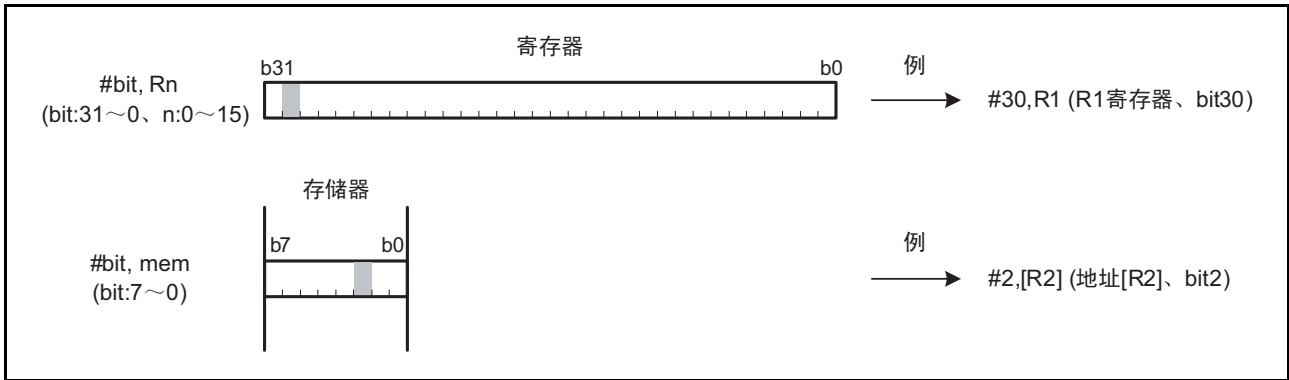


图 2.4 位

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** 和 **SWHILE** 共 7 种指令。

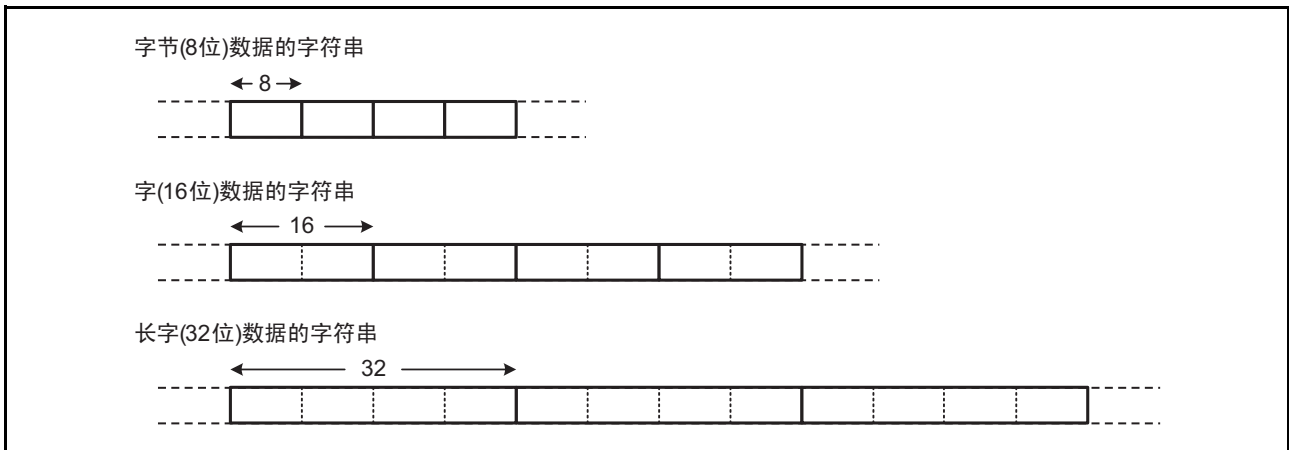


图 2.5 字符串

2.5 字节序

RX CPU 的指令固定为小端法。

数据排列能选择小端法或者大端法。

2.5.1 字节序的设定

在 RX63T 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。

有关字节序的设定，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	传送 LL	—	—	—	—
地址 1	传送 LH	传送 LL	—	—	—
地址 2	传送 HL	传送 LH	传送 LL	—	—
地址 3	传送 HH	传送 HL	传送 LH	传送 LL	—
地址 4	—	传送 HH	传送 HL	传送 LH	传送 LL
地址 5	—	—	传送 HH	传送 HL	传送 LH
地址 6	—	—	—	传送 HH	传送 HL
地址 7	—	—	—	—	传送 HH

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	传送 HH	—	—	—	—
地址 1	传送 HL	传送 HH	—	—	—
地址 2	传送 LH	传送 HL	传送 HH	—	—
地址 3	传送 LL	传送 LH	传送 HL	传送 HH	—
地址 4	—	传送 LL	传送 LH	传送 HL	传送 HH
地址 5	—	—	传送 LL	传送 LH	传送 HL
地址 6	—	—	—	传送 LL	传送 LH
地址 7	—	—	—	—	传送 LL

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	传送 LL	—	—	—	—	—	—
地址 1	传送 LH	传送 LL	—	—	—	—	—
地址 2	—	传送 LH	传送 LL	—	—	—	—
地址 3	—	—	传送 LH	传送 LL	—	—	—
地址 4	—	—	—	传送 LH	传送 LL	—	—
地址 5	—	—	—	—	传送 LH	传送 LL	—
地址 6	—	—	—	—	—	传送 LH	传送 LL
地址 7	—	—	—	—	—	—	传送 LH

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	传送 LH	—	—	—	—	—	—
地址 1	传送 LL	传送 LH	—	—	—	—	—
地址 2	—	传送 LL	传送 LH	—	—	—	—
地址 3	—	—	传送 LL	传送 LH	—	—	—
地址 4	—	—	—	传送 LL	传送 LH	—	—
地址 5	—	—	—	—	传送 LL	传送 LH	—
地址 6	—	—	—	—	—	传送 LL	传送 LH
地址 7	—	—	—	—	—	—	传送 LL

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	传送 LL	—	—	—
地址 1	—	传送 LL	—	—
地址 2	—	—	传送 LL	—
地址 3	—	—	—	传送 LL

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	传送 LL	—	—	—
地址 1	—	传送 LL	—	—
地址 2	—	—	传送 LL	—
地址 3	—	—	—	传送 LL

2.5.2 I/O 寄存器的存取

I/O 寄存器分配在固定地址，与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

必须按照以下规则存取 I/O 寄存器：

- 在存取8位总线宽度的I/O寄存器时，必须使用长度说明符（.size）为“.B”的指令或者长度扩展说明符（.memex）为“.B”或者“.UB”的指令。
- 在存取16位总线宽度的I/O寄存器时，必须使用长度说明符（.size）为“.W”的指令或者长度扩展说明符（.memex）为“.W”或者“.UW”的指令。
- 在存取32位总线宽度的I/O寄存器时，必须使用长度说明符（.size）为“.L”的指令或者长度扩展说明符（.memex）为“.L”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。

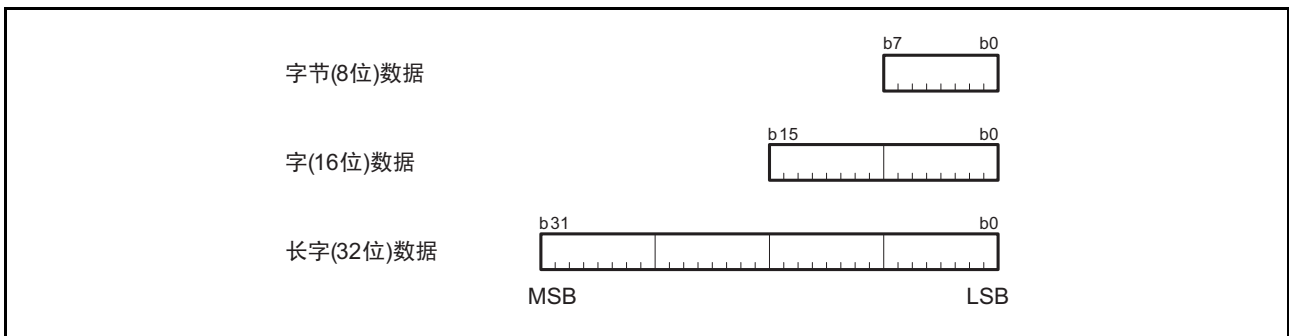


图 2.6 寄存器的数据排列

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。

数据类型	地址	数据映像 (小端法)	数据映像 (大端法)																																																																								
1位数据	地址L	<table border="1"> <tr> <td>b7</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>b0</td> </tr> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td> </tr> </table>	b7								b0	7	6	5	4	3	2	1	0		<table border="1"> <tr> <td>b7</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>b0</td> </tr> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td> </tr> </table>	b7								b0	7	6	5	4	3	2	1	0																																					
b7								b0																																																																			
7	6	5	4	3	2	1	0																																																																				
b7								b0																																																																			
7	6	5	4	3	2	1	0																																																																				
字节数据	地址L	<table border="1"> <tr> <td>MSB</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td> </tr> </table>	MSB								LSB	<table border="1"> <tr> <td>MSB</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td> </tr> </table>	MSB								LSB																																																						
MSB								LSB																																																																			
MSB								LSB																																																																			
字数据	地址M 地址M+1	<table border="1"> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td> </tr> <tr> <td>MSB</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td> </tr> </table>									LSB	MSB								LSB	<table border="1"> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td> </tr> <tr> <td>MSB</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td> </tr> </table>									LSB	MSB								LSB																																				
								LSB																																																																			
MSB								LSB																																																																			
								LSB																																																																			
MSB								LSB																																																																			
长字数据	地址N 地址N+1 地址N+2 地址N+3	<table border="1"> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td> </tr> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>MSB</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td> </tr> </table>									LSB																			MSB								LSB	<table border="1"> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td> </tr> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>MSB</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB</td> </tr> </table>									LSB																			MSB								LSB
								LSB																																																																			
MSB								LSB																																																																			
								LSB																																																																			
MSB								LSB																																																																			

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、存取异常、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)	
FFFFFFD0h	特权指令异常	
FFFFFFD4h	存取异常	
FFFFFFD8h	(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）。无条件陷阱发生源的 INT 指令分配在与操作数指定的序号（0 ~ 255）对应的向量，而 BRK 指令分配在序号 0 的向量。另外，中断源在分配在各产品规定的向量号（0 ~ 255）。有关中断的向量号，请参照“14.3.1 中断向量表”。

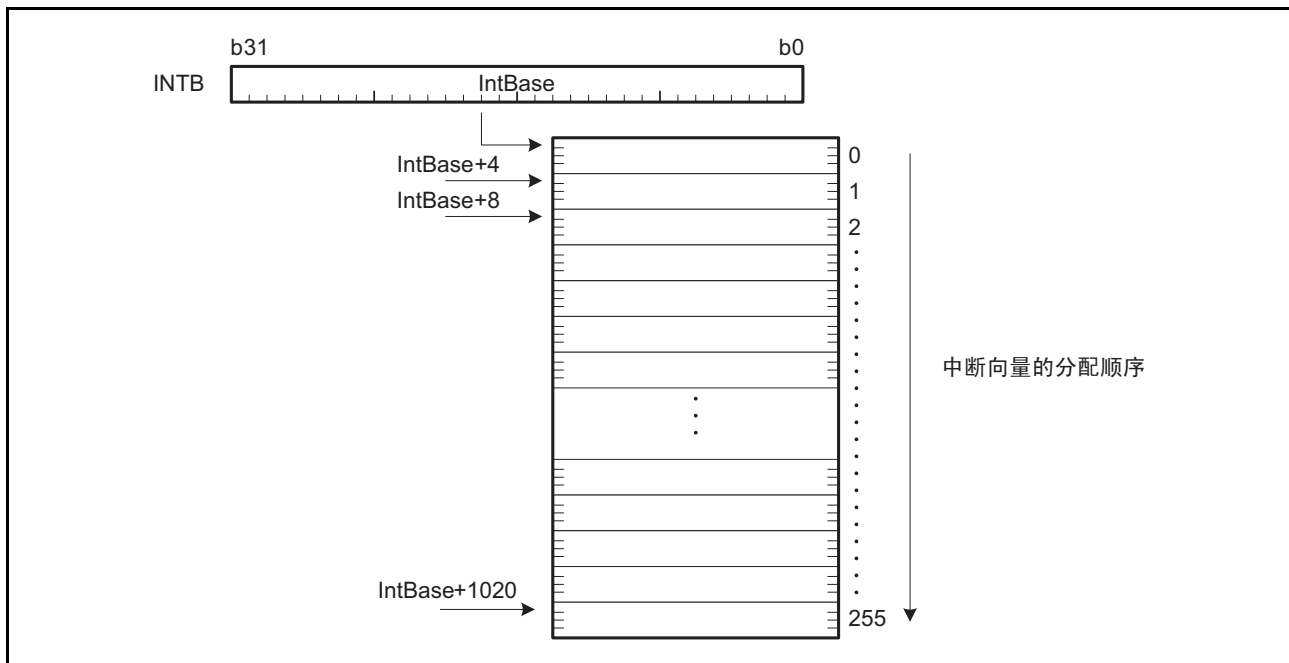


图 2.9 可变向量表

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，SSTR 指令以外的字符串操作指令（SCMPU、SMOVB、SMOVF、SMOVU、SUNTIL、SWHILE）和 RMPA 指令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、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。

装入时：一旦总线接受读请求，就进入 M2 阶段。如果接受请求和装入数据同时到达（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）。这是等待装入数据到达的阶段。一旦装入数据到达，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）。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。

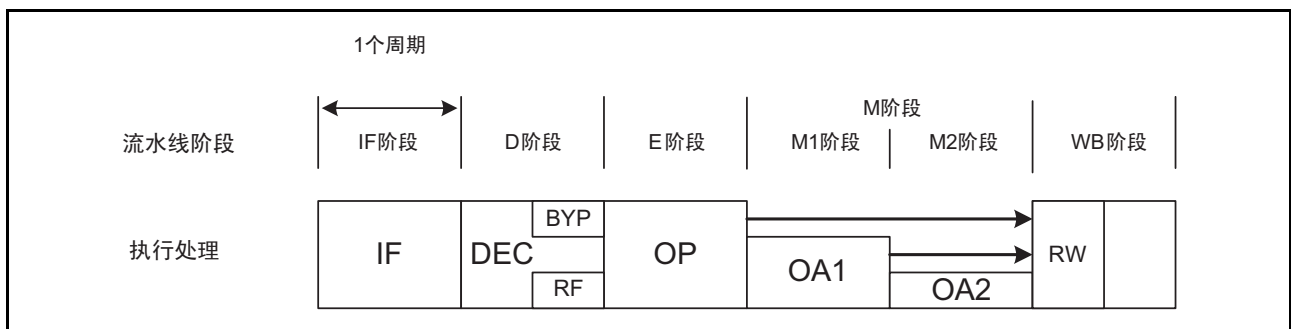


图 2.10 流水线结构及其运行

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

flag: 位、标志

Rs、Rs2、Rd、Rd2、Ri、Rb: 通用寄存器

CR: 控制寄存器

dsp: 位移量

pcdsp: 位移量

2.8.2.1 转换为单个微操作的指令和流水线处理

转换为单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 2.13 转换为单个微操作的指令

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) DIV、DIVU、EMUL、EMULU、 RMPA 和 SATR 除外	<ul style="list-style-type: none"> {ABS, NEG, NOT} "Rd"/"Rs, Rd" {ADC, MAX, MIN, ROTL, ROTR, XOR} "#IMM, Rd"/"Rs, Rd" ADD "#IMM, Rd"/"Rs, Rd"/"#IMM, Rs, Rd"/"Rs, Rs2, Rd" {AND, MUL, OR, SUB} "#IMM, Rd"/"Rs, Rd"/"Rs, Rs2, Rd" {CMP, TST} "#IMM, Rs"/"Rs, Rs2" NOP {ROLC, RORC, SAT} "Rd" SBB "Rs, Rd" {SHAR, SHLL, SHLR} "#IMM, Rd"/"Rs, Rd"/"#IMM, Rs, 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)	<ul style="list-style-type: none"> MOV "#IMM, Rd"/"Rs, Rd" {MOVU, REVL, REVW} "Rs, Rd" SCCnd "Rd" {STNZ, STZ} "#IMM, Rd" 	图 2.11	1
传送指令 (装入)	<ul style="list-style-type: none"> {MOV, MOVU} "[Rs], Rd"/"dsp[Rs], Rd"/"[Rs+], Rd" "/"[-Rs], Rd"/"[Ri, Rb], Rd" POP "Rd" 	图 2.12	吞吐量: 1 等待时间: 2 (注2)
传送指令 (存储)	<ul style="list-style-type: none"> MOV "Rs, [Rd]"/"Rs, dsp[Rd]"/"Rs, [Rd+]" "/"Rs, [-Rd]"/"Rs, [Ri, Rb]"/"#IMM, dsp[Rd]"/"#IMM, [Rd]" PUSH "Rs" PUSHC "CR" SCCnd "[Rd]"/"dsp[Rd]" 	图 2.13	1
位操作指令 (寄存器)	<ul style="list-style-type: none"> {BCL R, BNOT, BSET} "#IMM, Rd"/"Rs, Rd" BMCnd "#IMM, Rd" BTST "#IMM, Rs"/"Rs, Rs2" 	图 2.11	1
转移指令	<ul style="list-style-type: none"> 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, Rs"/"Rs, Rs2"	图 2.11	1
系统操作指令	<ul style="list-style-type: none"> CLRPSW, SETPSW "flag" MVTC "#IMM, CR"/"Rs, CR" MVFC "CR, Rd" MVTIPL "#IMM" 	—	1
DSP 功能指令	<ul style="list-style-type: none"> {MACHI, MACLO, MULHI, MULLO} "Rs, Rs2" {MVFACHI, MVFACMI} "Rd" {MVTACHI, MVTACLO} "Rs" RACW "#IMM" 	图 2.11	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”。

转换为基本的单个微操作的指令操作如图 2.11 ~ 图 2.13 所示。

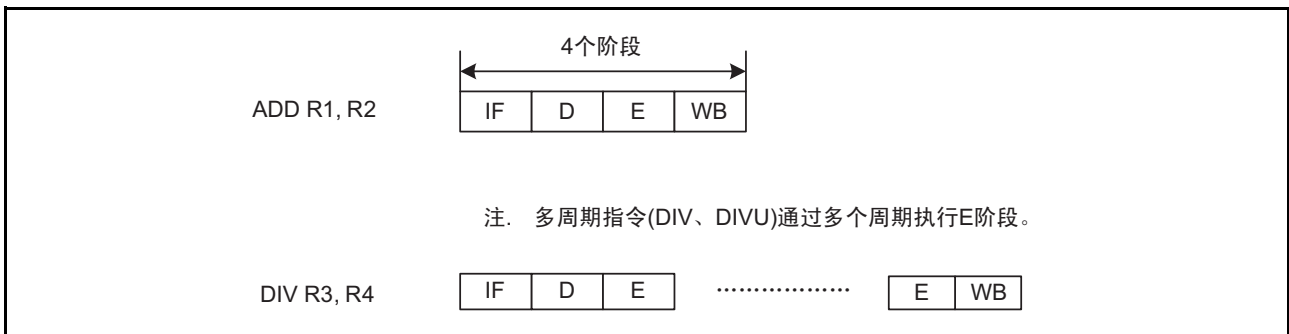


图 2.11 寄存器之间、立即数 - 寄存器的运算

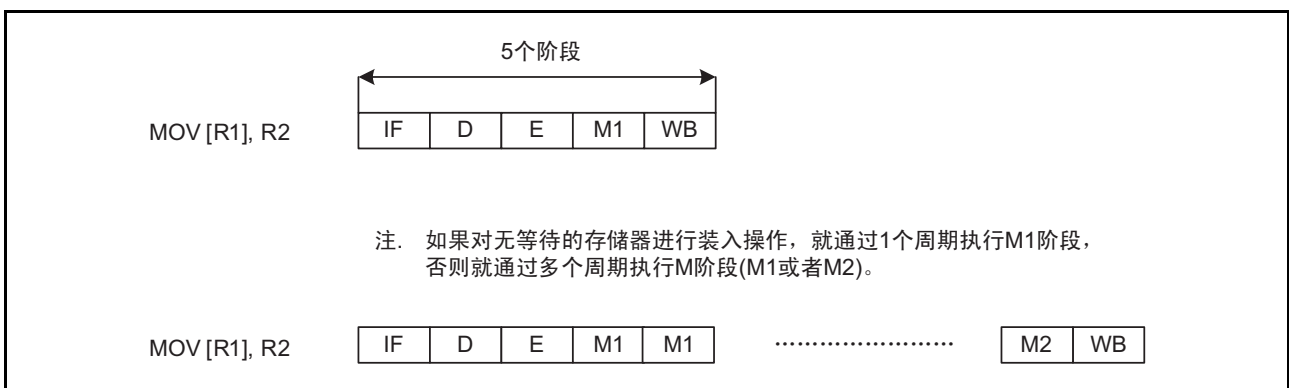


图 2.12 装入操作

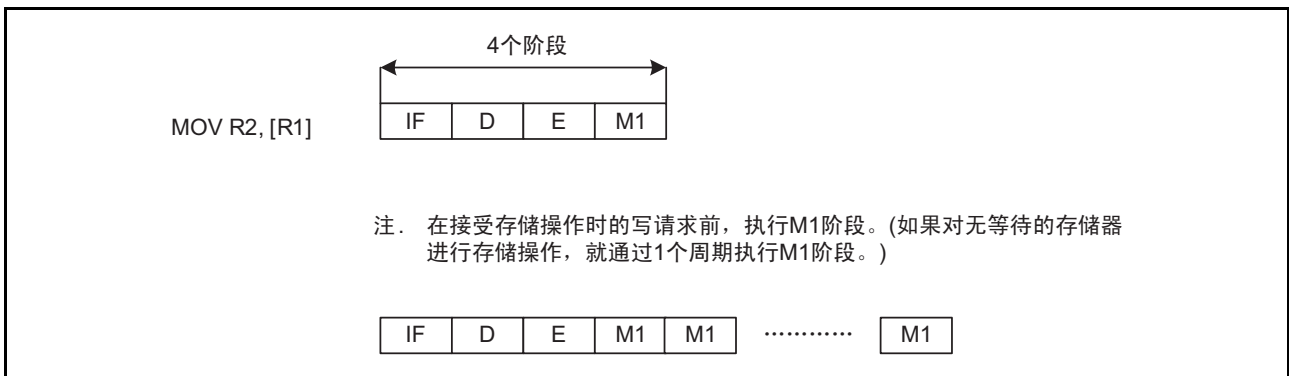


图 2.13 存储操作

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 (存储器源操作数)	<ul style="list-style-type: none"> {ADC, ADD, AND, MAX, MIN, MUL, OR, SBB, SUB, XOR} “[Rs], Rd”/“dsp[Rs], Rd” {CMP, TST} “[Rs], Rs2”/“dsp[Rs], Rs2” 	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 32bit×32bit→64bit) (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 (乘法 32bit×32bit→64bit) (存储器源操作数)	• {EMUL, EMULU} “[Rs], Rd”/“dsp[Rs], Rd”	—	4
算术 / 逻辑运算指令 (乘加运算)	• RMPA.B	—	6+7×floor(n/4)+4×(n%4) n 为处理字节数 (注 1)
	• RMPA.W	—	6+5×floor(n/2)+4×(n%2) n 为处理字数 (注 1)
	• RMPA.L	—	6+4n n 为处理长字数 (注 1)
算术 / 逻辑运算指令 (RMPA 指令的 64 位带符号的饱和处理)	• SATR	—	3
传送指令 (存储器之间的传送)	<ul style="list-style-type: none"> MOV “[Rs], [Rd]”/“dsp[Rs], [Rd]”/“[Rs], dsp[Rd]”/“dsp[Rs], dsp[Rd]” PUSH “[Rs]”/“dsp[Rs]” 	图 2.15	3
位操作指令 (存储器源操作数)	<ul style="list-style-type: none"> {BCLR, BNOT, BSET} “#IMM, [Rd]”/“#IMM, dsp[Rd]”/“Rs, [Rd]”/“Rs, dsp[Rd]” BMCnd “#IMM, [Rd]”/“#IMM, dsp[Rd]” BTST “#IMM, [Rs]”/“#IMM, dsp[Rs]”/“Rs, [Rs2]”/“Rs, dsp[Rs2]” 	图 2.15	3
传送指令 (装入指令)	• 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, ITOF} “Rs, Rd”	—	2
浮点运算指令 (存储器源操作数)	• {FADD, FSUB} “[Rs], Rd”/“dsp[Rs], Rd”	—	6
	• FCMP “[Rs], Rs2”/“dsp[Rs], Rs2”	—	3
	• FMUL “[Rs], Rd”/“dsp[Rs], Rd”	—	5
	• FDIV “[Rs], Rd”/“dsp[Rs], Rd”	—	18
	• {FTOI, ROUND, ITOF} “[Rs], Rd” /“dsp[Rs], Rd”	—	4
系统操作指令	• RTE	—	6
	• RTFI	—	3

?: 条件运算符

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 指令转换为多个装入操作。MOV 指令的装入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。

转换为基本的多个微操作的指令操作如图 2.14 ~ 图 2.21 所示。

注. mop: 微操作; stall: 流水线延迟

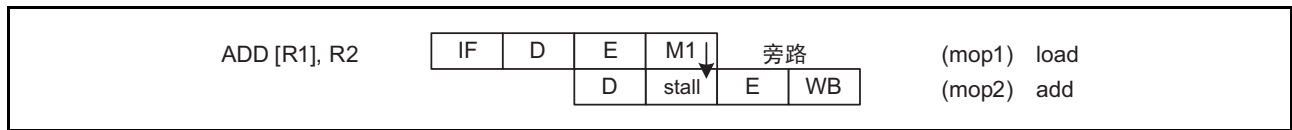


图 2.14 算术 / 逻辑运算指令 (存储器源操作数)

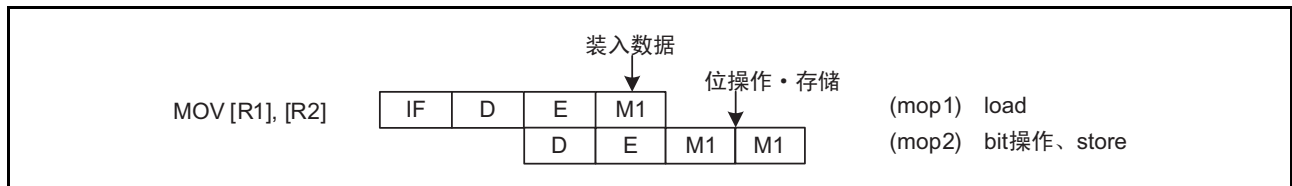


图 2.15 MOV 指令 (存储器之间的传送) 和位操作指令 (存储器源操作数)

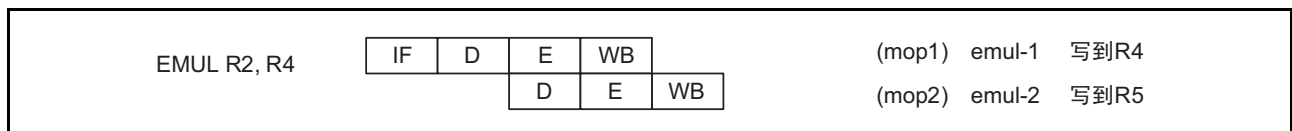


图 2.16 EMUL 指令和 EMULU 指令 (寄存器之间、寄存器 - 立即数)

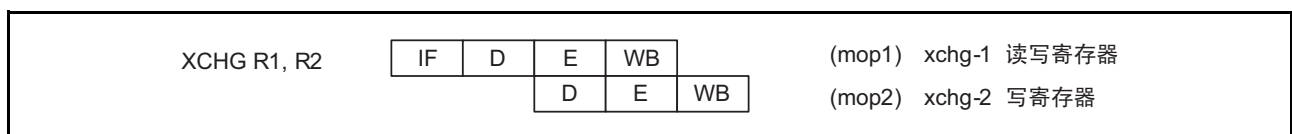


图 2.17 XCHG 指令 (寄存器)

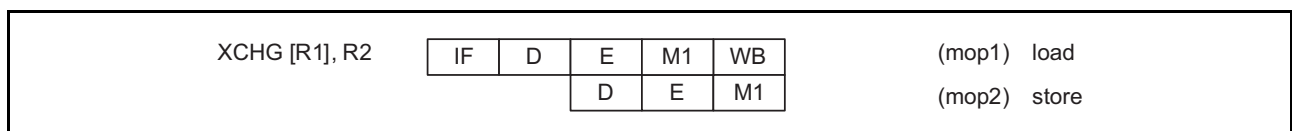


图 2.18 XCHG 指令 (存储器源操作数)

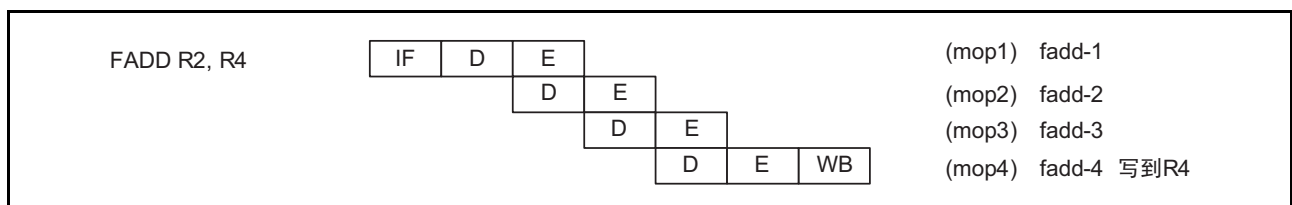


图 2.19 浮点运算指令 (寄存器之间、立即数 - 寄存器)

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。

注. mop: 微操作; stall: 流水线延迟

(1) 流水线处理混乱的情况

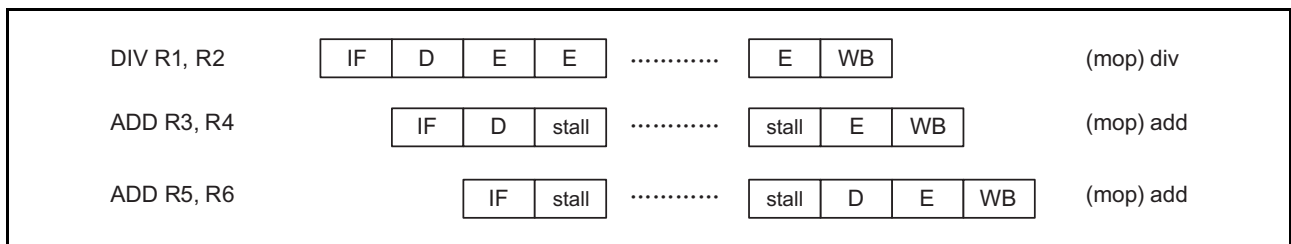


图 2.20 在执行 E 阶段过程中执行需要多个周期的指令的情况

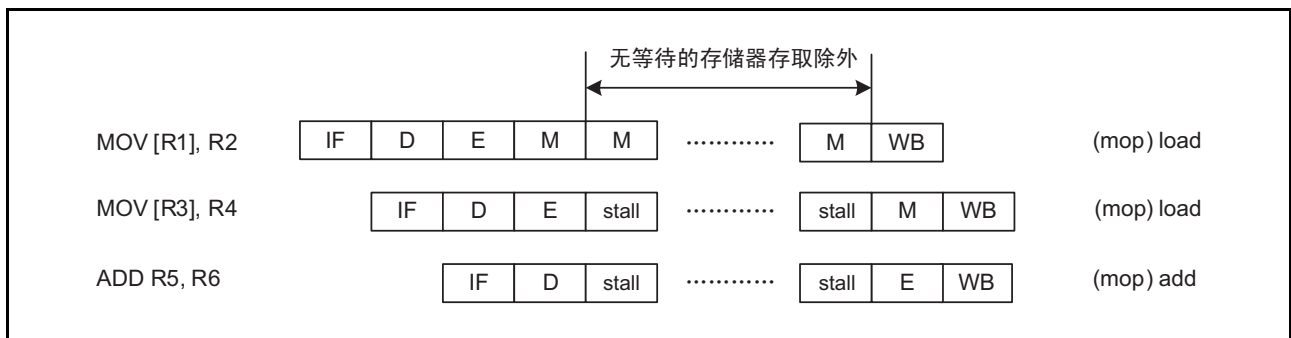


图 2.21 操作数存取没有在 1 个周期中结束的情况

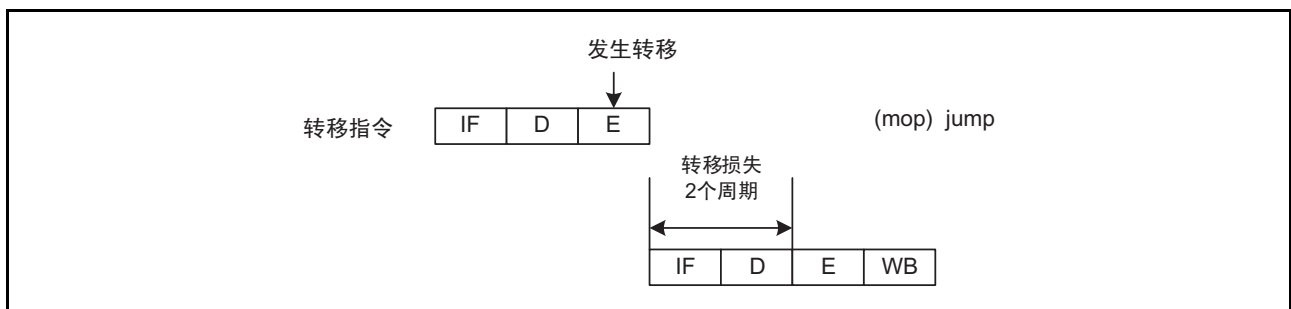


图 2.22 转移（在无条件的转移或者条件转移中条件成立的情况）

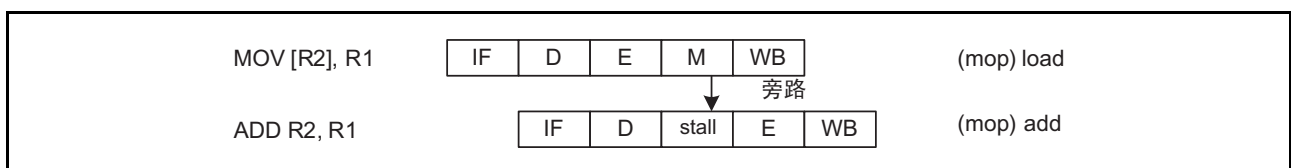


图 2.23 后续指令使用从存储器读取的操作数的情况

(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。

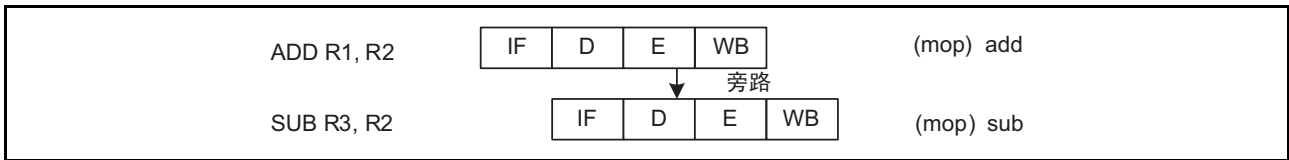


图 2.24 旁路

(b) 存储器的装入和运算的 WB 阶段重叠的情况

因为装入数据和运算结果能同时写到寄存器，所以即使存储器的装入和运算的 WB 阶段重叠，流水线处理也不会混乱。

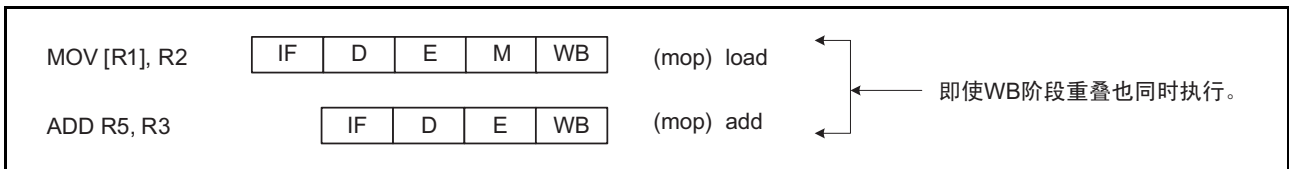


图 2.25 存储器的装入和运算的 WB 阶段重叠的情况

(c) 在结束存储器的装入前后续指令写相同寄存器的情况

在结束存储器的装入前，即使后续指令写相同的寄存器，因为取消了存储器装入的 WB 阶段，所以流水线处理不会混乱。

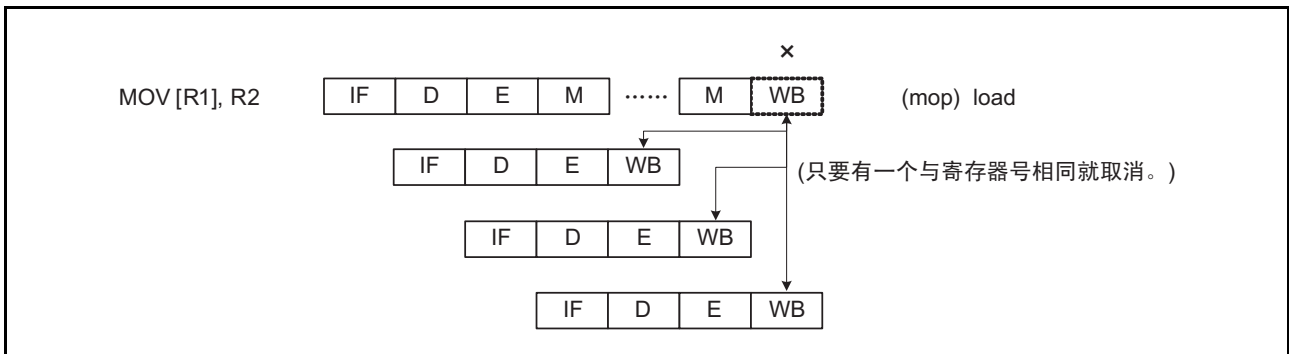


图 2.26 在结束存储器的装入前后续指令写相同寄存器的情况

(d) 后续指令不参照存储器装入的数据的情况

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

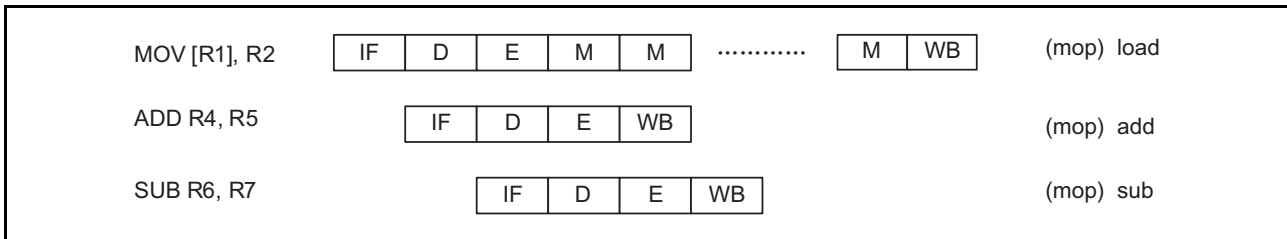


图 2.27 后续指令不参照存储器装入的数据的情况

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装入的结果，存储器装入的指令周期数就为“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。

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 的内部 RAM。 (将高速中断保存到控制寄存器。) 读向量。 转移到异常处理程序。	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。RX63T 群装载了能无等待存取的 ROM 和 RAM。通过将程序（包括向量）分配到 ROM，将堆栈区分配到 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 转换为单个微操作的指令”和“表 2.14 转换为多个微操作的指令”。

中断接受时序取决于流水线状态。有关中断的接受时序，请参照“13.3.1 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

有通过复位解除时的引脚电平选择的运行模式，也有在复位解除后通过软件选择的运行模式。

复位解除时的模式设定引脚（MD）的电平和当时所选运行模式的关系如表 3.1 所示。各运行模式的详细内容，请参照“3.3 运行模式的说明”。无论在何种模式中启动，都在内部 ROM（ROM、E2 数据闪存）有效的状态下开始运行。

表 3.1 通过模式设定引脚选择的运行模式

模式设定引脚	运行模式	SYSCR0 寄存器的初始状态
MD (注 1)		ROME 位
High	单芯片模式	1 (内部 ROM 有效)
Low	引导模式	

注 1. 不能在 MCU 运行中使 MD 引脚发生变化。

能通过系统控制寄存器 0（SYSCR0）设定的运行模式一览如表 3.2 所示。各运行模式的相信内容，请参照“3.3 运行模式的说明”。

表 3.2 通过寄存器选择的运行模式

SYSCR0 寄存器	运行模式
ROME 位	
0 (内部 ROM 无效) (注 1)	单芯片模式
1 (内部 ROM 有效)	

注 1. 如果将 ROME 位置“0”，就不能返回到“1”。

在单芯片模式中能选择字节序。通过表 3.3 中所示寄存器的字节序选择位（MDE[2:0]）设定各运行模式的字节序。设定值请参照表 3.4。

表 3.3 字节序的设定

运行模式	设定字节序
单芯片模式	通过选项设定存储器的字节序选择寄存器 S（MDES）选择。

表 3.4 字节序的选择

MDE[2:0] 位的设定值	选择的字节序
000b	大端法
111b	小端法

3.2 寄存器说明

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MD
复位后的值	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0/1 (注1)

注 1. 反映复位解除时的 MD 引脚的电平。

位	符号	位名	功能	R/W
b0	MD	MD 引脚状态标志	0: MD 引脚为 Low 电平 1: MD 引脚为 High 电平	R
b7-b1	—	保留位	读取值为“0”。	R
b8	—	保留位	读取值为不定值。	R
b15-b9	—	保留位	读取值为“0”。	R

3.2.2 系统控制寄存器 0 (SYSCR0)

地址 0008 0006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]							—	—	—	—	—	—	—	—	ROME
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	ROME	内部 ROM 有效位	0: 内部 ROM 无效 1: 内部 ROM 有效	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W
b15-b8	KEY[7:0]	SYSCR0 键码	控制是否改写 SYSCR0 寄存器。 如果改写 SYSCR0 寄存器，就必须以 16 位为单位，给高 8 位写“5Ah”、低 8 位置任意的值。	R/W (注1)

注 1. 不保持写数据。

ROME 位 (内部 ROM 有效位)

此位选择内部 ROM (ROM、E2 数据闪存) 的有效或者无效。

一旦将此位置“0”，就不能返回到“1”。

在执行内部 ROM 中的程序时，不能给此位写“0”。另外，在给 ROME 位写“0”后，必须在确认 ROME 位变为“0”后再进行以后的处理。

3.2.3 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
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

复位后的值

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

RAME 位 (RAM 有效位)

此位选择内部 RAM 的有效或者无效。

在存取 RAM 时，不能将此位置“0”。另外，在将 RAME 位从“0”改写为“1”后，必须在确认 RAME 位已被改写为“1”后进行 RAM 存取。

3.3 运行模式的说明

3.3.1 单芯片模式

单芯片模式是能将全部 I/O 端口用作通用输入 / 输出端口、外围功能输入 / 输出引脚或者中断输入引脚的模式。

如果在将 MD 引脚设定为 High 电平后解除复位，单芯片模式就启动。启动时的内部 ROM 有效 (SYSCR0.ROME 位 =1)。能通过软件将内部 ROM 设定为无效 (SYSCR0.ROME 位 =0)，但是，不能返回到内部 ROM 有效 (SYSCR0.ROME 位 =1)。

3.3.2 引导模式

引导模式是让保存在 MCU 内部的专用区域中的内部闪存改写程序 (引导程序) 运行的模式。

能使用异步串行接口 (SCI1)，从 MCU 外部改写内部闪存 (ROM、E2 数据闪存)。详细内容请参照“34. ROM (保存代码的闪存)”和“35. E2 数据闪存 (保存数据的闪存)”。

如果在将 MD 引脚设定为 Low 电平后解除复位，引导模式就启动。

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 MD 引脚进行运行模式的转移，如图 3.1 所示。

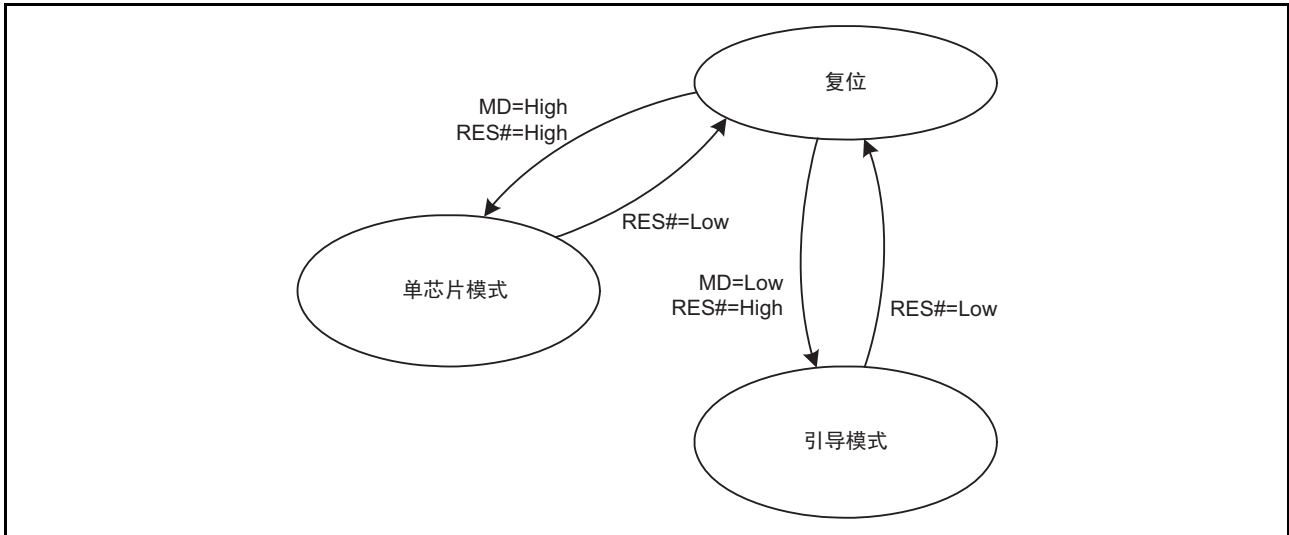


图 3.1 模式设定引脚的电平和运行模式

3.4.2 通过设定寄存器进行运行模式的转移

通过设定 SYSCR0.ROME 位进行运行模式的转移，如图 3.2 所示。

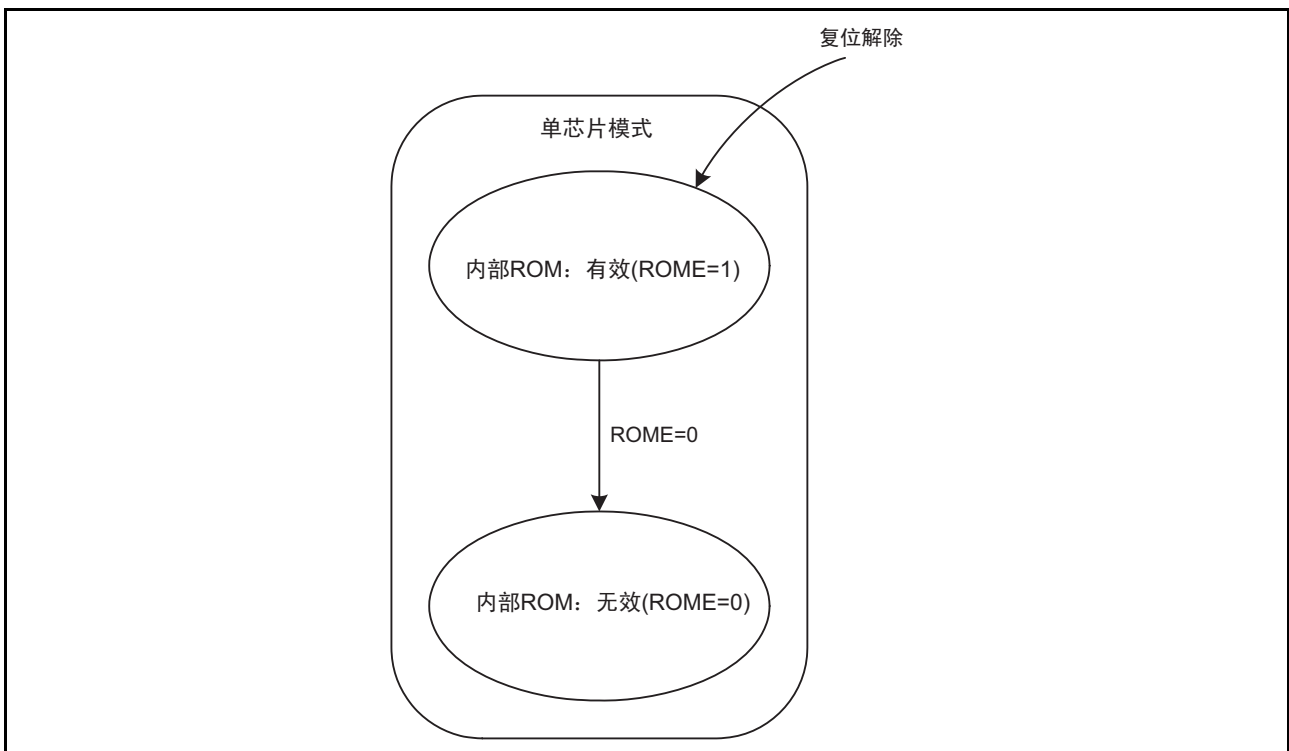


图 3.2 SYSCR0.ROME 位的设定和运行模式

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。

各运行模式的存储器映像如图 4.1 所示，能存取的区域因运行模式和各控制位的状态而不同。

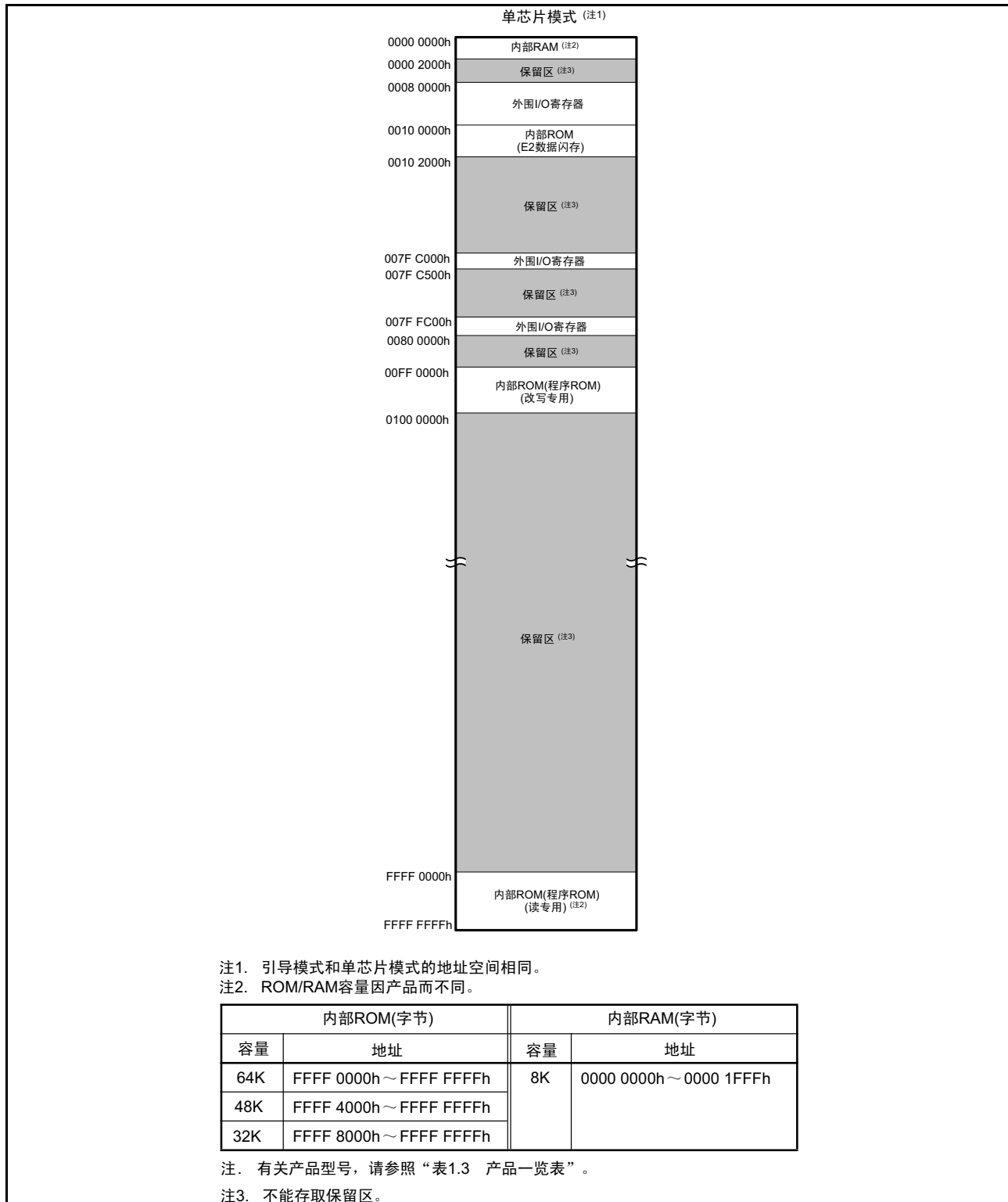


图 4.1 各运行模式的存储器映像

5. I/O 寄存器

I/O 寄存器一览表中汇总了有关内部寄存器地址的信息，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按顺序记载。
- 按照模块符号进行分类。
- 存取周期数表示指定的基准时钟的周期数。
- 在内部 I/O 寄存器的区域中，寄存器一览表中没有记载的地址区为保留区。因为无法保证存取这些寄存器时的运行及其后续运行，所以禁止存取保留区。

(2) 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。

需要注意：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]

- 要在通过清除中断请求允许位（ICU.IERn.IENj 位）来禁止中断请求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接着执行 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。

- (a) 写 I/O 寄存器
- (b) 将所写的 I/O 寄存器的值读到通用寄存器
- (c) 使用读取值进行运算
- (d) 执行后续指令

[指令例子]

- I/O 寄存器为字节的情况
MOV.L #SFR_ADDR, R1
MOV.B #SFR_DATA, [R1]
CMP [R1].UB, R1
;; 以后的处理
- I/O 寄存器为字的情况
MOV.L #SFR_ADDR, R1
MOV.W #SFR_DATA, [R1]
CMP [R1].W, R1
;; 以后的处理
- I/O 寄存器为长字的情况
MOV.L #SFR_ADDR, R1
MOV.L #SFR_DATA, [R1]
CMP [R1].L, R1
;; 以后的处理

另外，要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。

(3) I/O 寄存器的存取周期数

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”。
用以下表达式表示存取 I/O 寄存器时的存取周期数。(注 1)

$$\begin{aligned} \text{I/O 寄存器的存取周期数} = & \text{内部主总线 1 的总线周期数} + \\ & \text{分频时钟的同步周期数} + \\ & \text{内部外围总线 1} \sim \text{6 的总线周期数} \end{aligned}$$

内部外围总线 1 ~ 6 的总线周期数因存取的目标寄存器而不同。

在存取内部外围总线 2 ~ 6 连接的外围功能时，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 FCLK) 的频率比以及总线存取的时序而不同。

在外围功能部中，当频率关系为 $\text{ICLK} \geq \text{PCLK}$ (或者 FCLK) 时，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 FCLK) 周期，因此在表 5.1 中记载为 1 个 PCLK (或者 FCLK)。

另外，当频率关系为 $\text{ICLK} < \text{PCLK}$ (或者 FCLK) 时，从外围功能结束的下一个 ICLK 周期开始下一次的总线存取，因此以 ICLK 为单位记载。

注 1. 这是在和不同总线主控 (DMAC、DTC) 的总线存取不发生竞争的情况下，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 (1/17)

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		模块名	参照页数	
						ICLK ≥ PCLK 时	ICLK < PCLK 时			
0008 0000h	SYSTEM	模式监视寄存器	MDMONR	16	16	3 个 ICLK		运行模式	46	
0008 0006h	SYSTEM	系统控制寄存器 0	SYSCR0	16	16	3 个 ICLK			46	
0008 0008h	SYSTEM	系统控制寄存器 1	SYSCR1	16	16	3 个 ICLK			47	
0008 000Ch	SYSTEM	待机控制寄存器	SBYCR	16	16	3 个 ICLK		低功耗功能	136	
0008 0010h	SYSTEM	模块停止控制寄存器 A	MSTPCRA	32	32	3 个 ICLK			137	
0008 0014h	SYSTEM	模块停止控制寄存器 B	MSTPCRB	32	32	3 个 ICLK			138	
0008 0018h	SYSTEM	模块停止控制寄存器 C	MSTPCRC	32	32	3 个 ICLK			139	
0008 0020h	SYSTEM	系统时钟控制寄存器	SCKCR	32	32	3 个 ICLK			时钟发生电路	106
0008 0026h	SYSTEM	系统时钟控制寄存器 3	SCKCR3	16	16	3 个 ICLK				108
0008 0028h	SYSTEM	PLL 控制寄存器	PLLCR	16	16	3 个 ICLK		109		
0008 002Ah	SYSTEM	PLL 控制寄存器 2	PLLCR2	8	8	3 个 ICLK		110		
0008 0032h	SYSTEM	主时钟振荡器的控制寄存器	MOSCCR	8	8	3 个 ICLK		111		
0008 0034h	SYSTEM	低速内部振荡器的控制寄存器	LOCOCR	8	8	3 个 ICLK		112		
0008 0035h	SYSTEM	IWDT 专用内部振荡器的控制寄存器	ILOCOCR	8	8	3 个 ICLK		113		
0008 0040h	SYSTEM	振荡停止检测控制寄存器	OSTDCR	8	8	3 个 ICLK		114		
0008 0041h	SYSTEM	振荡停止检测状态寄存器	OSTDSR	8	8	3 个 ICLK		115		
0008 00A2h	SYSTEM	主时钟振荡器的等待控制寄存器	MOSCWTCR	8	8	3 个 ICLK		低功耗功能		140
0008 00A6h	SYSTEM	PLL 等待控制寄存器	PLLWTCR	8	8	3 个 ICLK				141
0008 00C0h	SYSTEM	复位状态寄存器 2	RSTSR2	8	8	3 个 ICLK		复位		74
0008 00C2h	SYSTEM	软件复位寄存器	SWRR	16	16	3 个 ICLK				75
0008 00E0h	SYSTEM	电压监视 1 电路的控制寄存器 1	LVD1CR1	8	8	3 个 ICLK		LVDA		91
0008 00E1h	SYSTEM	电压监视 1 电路的状态寄存器	LVD1SR	8	8	3 个 ICLK			91	
0008 00E2h	SYSTEM	电压监视 2 电路的控制寄存器 1	LVD2CR1	8	8	3 个 ICLK			92	
0008 00E3h	SYSTEM	电压监视 2 电路的状态寄存器	LVD2SR	8	8	3 个 ICLK			92	
0008 03FEh	SYSTEM	保护寄存器	PRCR	16	16	3 个 ICLK		寄存器的写保护功能	160	
0008 1300h	BSC	总线错误状态清除寄存器	BERCLR	8	8	2 个 ICLK			总线	213
0008 1304h	BSC	总线错误监视允许寄存器	BEREN	8	8	2 个 ICLK		213		
0008 1308h	BSC	总线错误状态寄存器 1	BERSR1	8	8	2 个 ICLK		214		
0008 130Ah	BSC	总线错误状态寄存器 2	BERSR2	16	16	2 个 ICLK		214		
0008 1310h	BSC	总线优先级控制寄存器	BUSPRI	16	16	2 个 ICLK		215		
0008 2000h	DMAC0	DMA 传送源地址寄存器	DMSAR	32	32	2 个 ICLK		DMACA		241
0008 2004h	DMAC0	DMA 传送目标地址寄存器	DMDAR	32	32	2 个 ICLK			241	
0008 2008h	DMAC0	DMA 传送计数寄存器	DMCRA	32	32	2 个 ICLK			242	
0008 200Ch	DMAC0	DMA 块传送计数寄存器	DMCRB	16	16	2 个 ICLK			243	
0008 2010h	DMAC0	DMA 传送模式寄存器	DMTMD	16	16	2 个 ICLK			244	
0008 2013h	DMAC0	DMA 中断设定寄存器	DMINT	8	8	2 个 ICLK			245	
0008 2014h	DMAC0	DMA 地址模式寄存器	DMAMD	16	16	2 个 ICLK			247	
0008 2018h	DMAC0	DMA 偏移寄存器	DMOFR	32	32	2 个 ICLK			250	
0008 201Ch	DMAC0	DMA 传送允许寄存器	DMCNT	8	8	2 个 ICLK			250	
0008 201Dh	DMAC0	DMA 软件启动寄存器	DMREQ	8	8	2 个 ICLK			251	
0008 201Eh	DMAC0	DMA 状态寄存器	DMSTS	8	8	2 个 ICLK			252	
0008 201Fh	DMAC0	DMA 启动源标志控制寄存器	DMCSL	8	8	2 个 ICLK			253	
0008 2040h	DMAC1	DMA 传送源地址寄存器	DMSAR	32	32	2 个 ICLK			241	
0008 2044h	DMAC1	DMA 传送目标地址寄存器	DMDAR	32	32	2 个 ICLK			241	
0008 2048h	DMAC1	DMA 传送计数寄存器	DMCRA	32	32	2 个 ICLK			242	
0008 204Ch	DMAC1	DMA 块传送计数寄存器	DMCRB	16	16	2 个 ICLK			243	
0008 2050h	DMAC1	DMA 传送模式寄存器	DMTMD	16	16	2 个 ICLK			244	
0008 2053h	DMAC1	DMA 中断设定寄存器	DMINT	8	8	2 个 ICLK			245	
0008 2054h	DMAC1	DMA 地址模式寄存器	DMAMD	16	16	2 个 ICLK			247	
0008 205Ch	DMAC1	DMA 传送允许寄存器	DMCNT	8	8	2 个 ICLK			250	
0008 205Dh	DMAC1	DMA 软件启动寄存器	DMREQ	8	8	2 个 ICLK			251	
0008 205Eh	DMAC1	DMA 状态寄存器	DMSTS	8	8	2 个 ICLK			252	
0008 205Fh	DMAC1	DMA 启动源标志控制寄存器	DMCSL	8	8	2 个 ICLK			253	
0008 2080h	DMAC2	DMA 传送源地址寄存器	DMSAR	32	32	2 个 ICLK			241	
0008 2084h	DMAC2	DMA 传送目标地址寄存器	DMDAR	32	32	2 个 ICLK			241	
0008 2088h	DMAC2	DMA 传送计数寄存器	DMCRA	32	32	2 个 ICLK			242	
0008 208Ch	DMAC2	DMA 块传送计数寄存器	DMCRB	16	16	2 个 ICLK			243	
0008 2090h	DMAC2	DMA 传送模式寄存器	DMTMD	16	16	2 个 ICLK			244	

表 5.1 I/O 寄存器地址一览表 (2/17)

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		模块名	参照页数
						ICLK \geq PCLK 时	ICLK < PCLK 时		
0008 2093h	DMAC2	DMA 中断设定寄存器	DMINT	8	8	2 个 ICLK		DMACA	245
0008 2094h	DMAC2	DMA 地址模式寄存器	DMAMD	16	16	2 个 ICLK			247
0008 209Ch	DMAC2	DMA 传送允许寄存器	DMCNT	8	8	2 个 ICLK			250
0008 209Dh	DMAC2	DMA 软件启动寄存器	DMREQ	8	8	2 个 ICLK			251
0008 209Eh	DMAC2	DMA 状态寄存器	DMSTS	8	8	2 个 ICLK			252
0008 209Fh	DMAC2	DMA 启动源标志控制寄存器	DMCSL	8	8	2 个 ICLK			253
0008 20C0h	DMAC3	DMA 传送源地址寄存器	DMSAR	32	32	2 个 ICLK			241
0008 20C4h	DMAC3	DMA 传送目标地址寄存器	DMDAR	32	32	2 个 ICLK			241
0008 20C8h	DMAC3	DMA 传送计数寄存器	DMCRA	32	32	2 个 ICLK			242
0008 20CCh	DMAC3	DMA 块传送计数寄存器	DMCRB	16	16	2 个 ICLK			243
0008 20D0h	DMAC3	DMA 传送模式寄存器	DMTMD	16	16	2 个 ICLK			244
0008 20D3h	DMAC3	DMA 中断设定寄存器	DMINT	8	8	2 个 ICLK			245
0008 20D4h	DMAC3	DMA 地址模式寄存器	DMAMD	16	16	2 个 ICLK			247
0008 20DCh	DMAC3	DMA 传送允许寄存器	DMCNT	8	8	2 个 ICLK			250
0008 20DDh	DMAC3	DMA 软件启动寄存器	DMREQ	8	8	2 个 ICLK			251
0008 20DEh	DMAC3	DMA 状态寄存器	DMSTS	8	8	2 个 ICLK			252
0008 20DFh	DMAC3	DMA 启动源标志控制寄存器	DMCSL	8	8	2 个 ICLK		253	
0008 2200h	DMAC	DMA 模块启动寄存器	DMAST	8	8	2 个 ICLK		254	
0008 2400h	DTC	DTC 控制寄存器	DTCCR	8	8	2 个 ICLK		DTCa	279
0008 2404h	DTC	DTC 向量基址寄存器	DTCVBR	32	32	2 个 ICLK			280
0008 2408h	DTC	DTC 地址模式寄存器	DTCADMOD	8	8	2 个 ICLK			280
0008 240Ch	DTC	DTC 模块启动寄存器	DTCST	8	8	2 个 ICLK			281
0008 240Eh	DTC	DTC 状态寄存器	DTCSTS	16	16	2 个 ICLK			282
0008 6400h	MPU	区域 0 起始页号寄存器	RSPAGE0	32	32	1 个 ICLK		MPU	222
0008 6404h	MPU	区域 0 结束页号寄存器	REPAGE0	32	32	1 个 ICLK			223
0008 6408h	MPU	区域 1 起始页号寄存器	RSPAGE1	32	32	1 个 ICLK			222
0008 640Ch	MPU	区域 1 结束页号寄存器	REPAGE1	32	32	1 个 ICLK			223
0008 6410h	MPU	区域 2 起始页号寄存器	RSPAGE2	32	32	1 个 ICLK			222
0008 6414h	MPU	区域 2 结束页号寄存器	REPAGE2	32	32	1 个 ICLK			223
0008 6418h	MPU	区域 3 起始页号寄存器	RSPAGE3	32	32	1 个 ICLK			222
0008 641Ch	MPU	区域 3 结束页号寄存器	REPAGE3	32	32	1 个 ICLK			223
0008 6420h	MPU	区域 4 起始页号寄存器	RSPAGE4	32	32	1 个 ICLK			222
0008 6424h	MPU	区域 4 结束页号寄存器	REPAGE4	32	32	1 个 ICLK			223
0008 6428h	MPU	区域 5 起始页号寄存器	RSPAGE5	32	32	1 个 ICLK			222
0008 642Ch	MPU	区域 5 结束页号寄存器	REPAGE5	32	32	1 个 ICLK			223
0008 6430h	MPU	区域 6 起始页号寄存器	RSPAGE6	32	32	1 个 ICLK			222
0008 6434h	MPU	区域 6 结束页号寄存器	REPAGE6	32	32	1 个 ICLK			223
0008 6438h	MPU	区域 7 起始页号寄存器	RSPAGE7	32	32	1 个 ICLK			222
0008 643Ch	MPU	区域 7 结束页号寄存器	REPAGE7	32	32	1 个 ICLK			223
0008 6500h	MPU	存储器保护功能有效化寄存器	MPEN	32	32	1 个 ICLK			224
0008 6504h	MPU	后台存取控制寄存器	MPBAC	32	32	1 个 ICLK			225
0008 6508h	MPU	存储器保护错误状态清除寄存器	MPECLR	32	32	1 个 ICLK			226
0008 650Ch	MPU	存储器保护错误状态寄存器	MPESTS	32	32	1 个 ICLK			227
0008 6514h	MPU	数据存储器保护错误地址寄存器	MPDEA	32	32	1 个 ICLK			228
0008 6520h	MPU	区域搜索地址寄存器	MPSA	32	32	1 个 ICLK			228
0008 6524h	MPU	区域搜索操作寄存器	MPOPS	16	16	1 个 ICLK			229
0008 6526h	MPU	区域无效化操作寄存器	MPOPI	16	16	1 个 ICLK			229
0008 6528h	MPU	指令命中区域寄存器	MHITI	32	32	1 个 ICLK		230	
0008 652Ch	MPU	数据命中区域寄存器	MHITD	32	32	1 个 ICLK		232	
0008 7010h	ICU	中断请求寄存器 016	IR016	8	8	2 个 ICLK		ICUb	172
0008 7015h	ICU	中断请求寄存器 021	IR021	8	8	2 个 ICLK			172
0008 7017h	ICU	中断请求寄存器 023	IR023	8	8	2 个 ICLK			172
0008 701Bh	ICU	中断请求寄存器 027	IR027	8	8	2 个 ICLK			172
0008 701Ch	ICU	中断请求寄存器 028	IR028	8	8	2 个 ICLK			172
0008 701Dh	ICU	中断请求寄存器 029	IR029	8	8	2 个 ICLK			172
0008 701Eh	ICU	中断请求寄存器 030	IR030	8	8	2 个 ICLK			172
0008 701Fh	ICU	中断请求寄存器 031	IR031	8	8	2 个 ICLK			172
0008 7024h	ICU	中断请求寄存器 036	IR036	8	8	2 个 ICLK			172
0008 7025h	ICU	中断请求寄存器 037	IR037	8	8	2 个 ICLK			172
0008 7026h	ICU	中断请求寄存器 038	IR038	8	8	2 个 ICLK			172
0008 7027h	ICU	中断请求寄存器 039	IR039	8	8	2 个 ICLK			172
0008 7028h	ICU	中断请求寄存器 040	IR040	8	8	2 个 ICLK			172

表 5.1 I/O 寄存器地址一览表 (3/17)

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		模块名	参照页数
						ICLK \geq PCLK 时	ICLK < PCLK 时		
0008 7029h	ICU	中断请求寄存器 041	IR041	8	8	2 个 ICLK		ICUb	172
0008 7039h	ICU	中断请求寄存器 057	IR057	8	8	2 个 ICLK			172
0008 7040h	ICU	中断请求寄存器 064	IR064	8	8	2 个 ICLK			172
0008 7041h	ICU	中断请求寄存器 065	IR065	8	8	2 个 ICLK			172
0008 7042h	ICU	中断请求寄存器 066	IR066	8	8	2 个 ICLK			172
0008 7043h	ICU	中断请求寄存器 067	IR067	8	8	2 个 ICLK			172
0008 7044h	ICU	中断请求寄存器 068	IR068	8	8	2 个 ICLK			172
0008 7045h	ICU	中断请求寄存器 069	IR069	8	8	2 个 ICLK			172
0008 7066h	ICU	中断请求寄存器 102	IR102	8	8	2 个 ICLK			172
0008 7067h	ICU	中断请求寄存器 103	IR103	8	8	2 个 ICLK			172
0008 7072h	ICU	中断请求寄存器 114	IR114	8	8	2 个 ICLK			172
0008 707Ah	ICU	中断请求寄存器 122	IR122	8	8	2 个 ICLK			172
0008 707Bh	ICU	中断请求寄存器 123	IR123	8	8	2 个 ICLK			172
0008 707Ch	ICU	中断请求寄存器 124	IR124	8	8	2 个 ICLK			172
0008 707Dh	ICU	中断请求寄存器 125	IR125	8	8	2 个 ICLK			172
0008 707Eh	ICU	中断请求寄存器 126	IR126	8	8	2 个 ICLK			172
0008 707Fh	ICU	中断请求寄存器 127	IR127	8	8	2 个 ICLK			172
0008 7080h	ICU	中断请求寄存器 128	IR128	8	8	2 个 ICLK			172
0008 7081h	ICU	中断请求寄存器 129	IR129	8	8	2 个 ICLK			172
0008 7082h	ICU	中断请求寄存器 130	IR130	8	8	2 个 ICLK		172	
0008 7083h	ICU	中断请求寄存器 131	IR131	8	8	2 个 ICLK		172	
0008 7084h	ICU	中断请求寄存器 132	IR132	8	8	2 个 ICLK		172	
0008 7085h	ICU	中断请求寄存器 133	IR133	8	8	2 个 ICLK		172	
0008 7086h	ICU	中断请求寄存器 134	IR134	8	8	2 个 ICLK		172	
0008 7087h	ICU	中断请求寄存器 135	IR135	8	8	2 个 ICLK		172	
0008 7088h	ICU	中断请求寄存器 136	IR136	8	8	2 个 ICLK		172	
0008 7089h	ICU	中断请求寄存器 137	IR137	8	8	2 个 ICLK		172	
0008 708Ah	ICU	中断请求寄存器 138	IR138	8	8	2 个 ICLK		172	
0008 708Bh	ICU	中断请求寄存器 139	IR139	8	8	2 个 ICLK		172	
0008 708Ch	ICU	中断请求寄存器 140	IR140	8	8	2 个 ICLK		172	
0008 708Dh	ICU	中断请求寄存器 141	IR141	8	8	2 个 ICLK		172	
0008 708Eh	ICU	中断请求寄存器 142	IR142	8	8	2 个 ICLK		172	
0008 708Fh	ICU	中断请求寄存器 143	IR143	8	8	2 个 ICLK		172	
0008 7090h	ICU	中断请求寄存器 144	IR144	8	8	2 个 ICLK		172	
0008 7091h	ICU	中断请求寄存器 145	IR145	8	8	2 个 ICLK		172	
0008 7092h	ICU	中断请求寄存器 146	IR146	8	8	2 个 ICLK		172	
0008 7093h	ICU	中断请求寄存器 147	IR147	8	8	2 个 ICLK		172	
0008 7094h	ICU	中断请求寄存器 148	IR148	8	8	2 个 ICLK		172	
0008 7095h	ICU	中断请求寄存器 149	IR149	8	8	2 个 ICLK		172	
0008 7096h	ICU	中断请求寄存器 150	IR150	8	8	2 个 ICLK		172	
0008 7097h	ICU	中断请求寄存器 151	IR151	8	8	2 个 ICLK		172	
0008 7098h	ICU	中断请求寄存器 152	IR152	8	8	2 个 ICLK		172	
0008 7099h	ICU	中断请求寄存器 153	IR153	8	8	2 个 ICLK		172	
0008 709Ah	ICU	中断请求寄存器 154	IR154	8	8	2 个 ICLK		172	
0008 709Bh	ICU	中断请求寄存器 155	IR155	8	8	2 个 ICLK		172	
0008 709Ch	ICU	中断请求寄存器 156	IR156	8	8	2 个 ICLK		172	
0008 709Dh	ICU	中断请求寄存器 157	IR157	8	8	2 个 ICLK		172	
0008 709Eh	ICU	中断请求寄存器 158	IR158	8	8	2 个 ICLK		172	
0008 70A1h	ICU	中断请求寄存器 161	IR161	8	8	2 个 ICLK		172	
0008 70A2h	ICU	中断请求寄存器 162	IR162	8	8	2 个 ICLK		172	
0008 70A3h	ICU	中断请求寄存器 163	IR163	8	8	2 个 ICLK		172	
0008 70A4h	ICU	中断请求寄存器 164	IR164	8	8	2 个 ICLK		172	
0008 70A5h	ICU	中断请求寄存器 165	IR165	8	8	2 个 ICLK		172	
0008 70A6h	ICU	中断请求寄存器 166	IR166	8	8	2 个 ICLK		172	
0008 70A8h	ICU	中断请求寄存器 168	IR168	8	8	2 个 ICLK		172	
0008 70A9h	ICU	中断请求寄存器 169	IR169	8	8	2 个 ICLK		172	
0008 70ABh	ICU	中断请求寄存器 171	IR171	8	8	2 个 ICLK		172	
0008 70ACh	ICU	中断请求寄存器 172	IR172	8	8	2 个 ICLK		172	
0008 70ADh	ICU	中断请求寄存器 173	IR173	8	8	2 个 ICLK		172	
0008 70C2h	ICU	中断请求寄存器 194	IR194	8	8	2 个 ICLK		172	
0008 70C3h	ICU	中断请求寄存器 195	IR195	8	8	2 个 ICLK		172	
0008 70C4h	ICU	中断请求寄存器 196	IR196	8	8	2 个 ICLK		172	

表 5.1 I/O 寄存器地址一览表 (4/17)

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		模块名	参照页数
						ICLK \geq PCLK 时	ICLK $<$ PCLK 时		
0008 70C5h	ICU	中断请求寄存器 197	IR197	8	8	2 个 ICLK		ICUb	172
0008 70C6h	ICU	中断请求寄存器 198	IR198	8	8	2 个 ICLK			172
0008 70C7h	ICU	中断请求寄存器 199	IR199	8	8	2 个 ICLK			172
0008 70C8h	ICU	中断请求寄存器 200	IR200	8	8	2 个 ICLK			172
0008 70C9h	ICU	中断请求寄存器 201	IR201	8	8	2 个 ICLK			172
0008 70D6h	ICU	中断请求寄存器 214	IR214	8	8	2 个 ICLK			172
0008 70D7h	ICU	中断请求寄存器 215	IR215	8	8	2 个 ICLK			172
0008 70D8h	ICU	中断请求寄存器 216	IR216	8	8	2 个 ICLK			172
0008 70D9h	ICU	中断请求寄存器 217	IR217	8	8	2 个 ICLK			172
0008 70DAh	ICU	中断请求寄存器 218	IR218	8	8	2 个 ICLK			172
0008 70DBh	ICU	中断请求寄存器 219	IR219	8	8	2 个 ICLK			172
0008 70E2h	ICU	中断请求寄存器 226	IR226	8	8	2 个 ICLK			172
0008 70E3h	ICU	中断请求寄存器 227	IR227	8	8	2 个 ICLK			172
0008 70E4h	ICU	中断请求寄存器 228	IR228	8	8	2 个 ICLK			172
0008 70E5h	ICU	中断请求寄存器 229	IR229	8	8	2 个 ICLK			172
0008 70E6h	ICU	中断请求寄存器 230	IR230	8	8	2 个 ICLK			172
0008 70E7h	ICU	中断请求寄存器 231	IR231	8	8	2 个 ICLK			172
0008 70E8h	ICU	中断请求寄存器 232	IR232	8	8	2 个 ICLK			172
0008 70E9h	ICU	中断请求寄存器 233	IR233	8	8	2 个 ICLK			172
0008 70EAh	ICU	中断请求寄存器 234	IR234	8	8	2 个 ICLK			172
0008 70EBh	ICU	中断请求寄存器 235	IR235	8	8	2 个 ICLK			172
0008 70ECh	ICU	中断请求寄存器 236	IR236	8	8	2 个 ICLK			172
0008 70EEh	ICU	中断请求寄存器 238	IR238	8	8	2 个 ICLK			172
0008 70EFh	ICU	中断请求寄存器 239	IR239	8	8	2 个 ICLK			172
0008 70F0h	ICU	中断请求寄存器 240	IR240	8	8	2 个 ICLK			172
0008 70F1h	ICU	中断请求寄存器 241	IR241	8	8	2 个 ICLK			172
0008 70F2h	ICU	中断请求寄存器 242	IR242	8	8	2 个 ICLK			172
0008 70F4h	ICU	中断请求寄存器 244	IR244	8	8	2 个 ICLK			172
0008 70F5h	ICU	中断请求寄存器 245	IR245	8	8	2 个 ICLK			172
0008 70F6h	ICU	中断请求寄存器 246	IR246	8	8	2 个 ICLK			172
0008 70F7h	ICU	中断请求寄存器 247	IR247	8	8	2 个 ICLK			172
0008 70F8h	ICU	中断请求寄存器 248	IR248	8	8	2 个 ICLK			172
0008 70FAh	ICU	中断请求寄存器 250	IR250	8	8	2 个 ICLK			172
0008 70FBh	ICU	中断请求寄存器 251	IR251	8	8	2 个 ICLK			172
0008 70FCh	ICU	中断请求寄存器 252	IR252	8	8	2 个 ICLK			172
0008 711Bh	ICU	DTC 启动允许寄存器 027	DT CER027	8	8	2 个 ICLK			176
0008 711Ch	ICU	DTC 启动允许寄存器 028	DT CER028	8	8	2 个 ICLK			176
0008 711Dh	ICU	DTC 启动允许寄存器 029	DT CER029	8	8	2 个 ICLK			176
0008 711Eh	ICU	DTC 启动允许寄存器 030	DT CER030	8	8	2 个 ICLK			176
0008 711Fh	ICU	DTC 启动允许寄存器 031	DT CER031	8	8	2 个 ICLK			176
0008 7127h	ICU	DTC 启动允许寄存器 039	DT CER039	8	8	2 个 ICLK		176	
0008 7128h	ICU	DTC 启动允许寄存器 040	DT CER040	8	8	2 个 ICLK		176	
0008 7140h	ICU	DTC 启动允许寄存器 064	DT CER064	8	8	2 个 ICLK		176	
0008 7141h	ICU	DTC 启动允许寄存器 065	DT CER065	8	8	2 个 ICLK		176	
0008 7142h	ICU	DTC 启动允许寄存器 066	DT CER066	8	8	2 个 ICLK		176	
0008 7143h	ICU	DTC 启动允许寄存器 067	DT CER067	8	8	2 个 ICLK		176	
0008 7144h	ICU	DTC 启动允许寄存器 068	DT CER068	8	8	2 个 ICLK		176	
0008 7145h	ICU	DTC 启动允许寄存器 069	DT CER069	8	8	2 个 ICLK		176	
0008 7166h	ICU	DTC 启动允许寄存器 102	DT CER102	8	8	2 个 ICLK		176	
0008 7167h	ICU	DTC 启动允许寄存器 103	DT CER103	8	8	2 个 ICLK		176	
0008 717Eh	ICU	DTC 启动允许寄存器 126	DT CER126	8	8	2 个 ICLK		176	
0008 717Fh	ICU	DTC 启动允许寄存器 127	DT CER127	8	8	2 个 ICLK		176	
0008 7180h	ICU	DTC 启动允许寄存器 128	DT CER128	8	8	2 个 ICLK		176	
0008 7181h	ICU	DTC 启动允许寄存器 129	DT CER129	8	8	2 个 ICLK		176	
0008 7185h	ICU	DTC 启动允许寄存器 133	DT CER133	8	8	2 个 ICLK		176	
0008 7186h	ICU	DTC 启动允许寄存器 134	DT CER134	8	8	2 个 ICLK		176	
0008 7189h	ICU	DTC 启动允许寄存器 137	DT CER137	8	8	2 个 ICLK		176	
0008 718Ah	ICU	DTC 启动允许寄存器 138	DT CER138	8	8	2 个 ICLK		176	
0008 718Dh	ICU	DTC 启动允许寄存器 141	DT CER141	8	8	2 个 ICLK		176	
0008 718Eh	ICU	DTC 启动允许寄存器 142	DT CER142	8	8	2 个 ICLK		176	
0008 718Fh	ICU	DTC 启动允许寄存器 143	DT CER143	8	8	2 个 ICLK		176	
0008 7190h	ICU	DTC 启动允许寄存器 144	DT CER144	8	8	2 个 ICLK		176	

表 5.1 I/O 寄存器地址一览表 (5/17)

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		模块名	参照页数
						ICLK \geq PCLK 时	ICLK < PCLK 时		
0008 7192h	ICU	DTC 启动允许寄存器 146	DTCER146	8	8	2 个 ICLK		ICUb	176
0008 7193h	ICU	DTC 启动允许寄存器 147	DTCER147	8	8	2 个 ICLK			176
0008 7194h	ICU	DTC 启动允许寄存器 148	DTCER148	8	8	2 个 ICLK			176
0008 7195h	ICU	DTC 启动允许寄存器 149	DTCER149	8	8	2 个 ICLK			176
0008 7196h	ICU	DTC 启动允许寄存器 150	DTCER150	8	8	2 个 ICLK			176
0008 7197h	ICU	DTC 启动允许寄存器 151	DTCER151	8	8	2 个 ICLK			176
0008 7198h	ICU	DTC 启动允许寄存器 152	DTCER152	8	8	2 个 ICLK			176
0008 7199h	ICU	DTC 启动允许寄存器 153	DTCER153	8	8	2 个 ICLK			176
0008 719Ah	ICU	DTC 启动允许寄存器 154	DTCER154	8	8	2 个 ICLK			176
0008 719Bh	ICU	DTC 启动允许寄存器 155	DTCER155	8	8	2 个 ICLK			176
0008 719Ch	ICU	DTC 启动允许寄存器 156	DTCER156	8	8	2 个 ICLK			176
0008 719Dh	ICU	DTC 启动允许寄存器 157	DTCER157	8	8	2 个 ICLK			176
0008 71A1h	ICU	DTC 启动允许寄存器 161	DTCER161	8	8	2 个 ICLK			176
0008 71A2h	ICU	DTC 启动允许寄存器 162	DTCER162	8	8	2 个 ICLK			176
0008 71A3h	ICU	DTC 启动允许寄存器 163	DTCER163	8	8	2 个 ICLK			176
0008 71A4h	ICU	DTC 启动允许寄存器 164	DTCER164	8	8	2 个 ICLK			176
0008 71A5h	ICU	DTC 启动允许寄存器 165	DTCER165	8	8	2 个 ICLK			176
0008 71ABh	ICU	DTC 启动允许寄存器 171	DTCER171	8	8	2 个 ICLK			176
0008 71ACh	ICU	DTC 启动允许寄存器 172	DTCER172	8	8	2 个 ICLK			176
0008 71ADh	ICU	DTC 启动允许寄存器 173	DTCER173	8	8	2 个 ICLK			176
0008 71C3h	ICU	DTC 启动允许寄存器 195	DTCER195	8	8	2 个 ICLK			176
0008 71C4h	ICU	DTC 启动允许寄存器 196	DTCER196	8	8	2 个 ICLK			176
0008 71C6h	ICU	DTC 启动允许寄存器 198	DTCER198	8	8	2 个 ICLK			176
0008 71C7h	ICU	DTC 启动允许寄存器 199	DTCER199	8	8	2 个 ICLK			176
0008 71C8h	ICU	DTC 启动允许寄存器 200	DTCER200	8	8	2 个 ICLK			176
0008 71C9h	ICU	DTC 启动允许寄存器 201	DTCER201	8	8	2 个 ICLK			176
0008 71D6h	ICU	DTC 启动允许寄存器 214	DTCER214	8	8	2 个 ICLK			176
0008 71D7h	ICU	DTC 启动允许寄存器 215	DTCER215	8	8	2 个 ICLK			176
0008 71D9h	ICU	DTC 启动允许寄存器 217	DTCER217	8	8	2 个 ICLK			176
0008 71DAh	ICU	DTC 启动允许寄存器 218	DTCER218	8	8	2 个 ICLK			176
0008 71E2h	ICU	DTC 启动允许寄存器 226	DTCER226	8	8	2 个 ICLK			226
0008 71E3h	ICU	DTC 启动允许寄存器 227	DTCER227	8	8	2 个 ICLK			226
0008 71E4h	ICU	DTC 启动允许寄存器 228	DTCER228	8	8	2 个 ICLK			176
0008 71E5h	ICU	DTC 启动允许寄存器 229	DTCER229	8	8	2 个 ICLK			176
0008 71E6h	ICU	DTC 启动允许寄存器 230	DTCER230	8	8	2 个 ICLK			176
0008 71E7h	ICU	DTC 启动允许寄存器 231	DTCER231	8	8	2 个 ICLK			176
0008 71E8h	ICU	DTC 启动允许寄存器 232	DTCER232	8	8	2 个 ICLK			176
0008 71E9h	ICU	DTC 启动允许寄存器 233	DTCER233	8	8	2 个 ICLK			176
0008 71EAh	ICU	DTC 启动允许寄存器 234	DTCER234	8	8	2 个 ICLK			176
0008 71EBh	ICU	DTC 启动允许寄存器 235	DTCER235	8	8	2 个 ICLK			176
0008 71ECh	ICU	DTC 启动允许寄存器 236	DTCER236	8	8	2 个 ICLK			176
0008 71EEh	ICU	DTC 启动允许寄存器 238	DTCER238	8	8	2 个 ICLK			176
0008 71EFh	ICU	DTC 启动允许寄存器 239	DTCER239	8	8	2 个 ICLK			176
0008 71F0h	ICU	DTC 启动允许寄存器 240	DTCER240	8	8	2 个 ICLK			176
0008 71F1h	ICU	DTC 启动允许寄存器 241	DTCER241	8	8	2 个 ICLK			176
0008 71F2h	ICU	DTC 启动允许寄存器 242	DTCER242	8	8	2 个 ICLK			176
0008 71F4h	ICU	DTC 启动允许寄存器 244	DTCER244	8	8	2 个 ICLK			176
0008 71F5h	ICU	DTC 启动允许寄存器 245	DTCER245	8	8	2 个 ICLK		176	
0008 71F6h	ICU	DTC 启动允许寄存器 246	DTCER246	8	8	2 个 ICLK		176	
0008 71F7h	ICU	DTC 启动允许寄存器 247	DTCER247	8	8	2 个 ICLK		176	
0008 71F8h	ICU	DTC 启动允许寄存器 248	DTCER248	8	8	2 个 ICLK		176	
0008 71FAh	ICU	DTC 启动允许寄存器 250	DTCER250	8	8	2 个 ICLK		176	
0008 71FBh	ICU	DTC 启动允许寄存器 251	DTCER251	8	8	2 个 ICLK		176	
0008 7202h	ICU	中断请求允许寄存器 02	IER02	8	8	2 个 ICLK		173	
0008 7203h	ICU	中断请求允许寄存器 03	IER03	8	8	2 个 ICLK		173	
0008 7204h	ICU	中断请求允许寄存器 04	IER04	8	8	2 个 ICLK		173	
0008 7205h	ICU	中断请求允许寄存器 05	IER05	8	8	2 个 ICLK		173	
0008 7207h	ICU	中断请求允许寄存器 07	IER07	8	8	2 个 ICLK		173	
0008 7208h	ICU	中断请求允许寄存器 08	IER08	8	8	2 个 ICLK		173	
0008 720Ch	ICU	中断请求允许寄存器 0C	IER0C	8	8	2 个 ICLK		173	
0008 720Eh	ICU	中断请求允许寄存器 0E	IER0E	8	8	2 个 ICLK		173	
0008 720Fh	ICU	中断请求允许寄存器 0F	IER0F	8	8	2 个 ICLK		173	

表 5.1 I/O 寄存器地址一览表 (6/17)

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		模块名	参照页数
						ICLK \geq PCLK 时	ICLK < PCLK 时		
0008 7210h	ICU	中断请求允许寄存器 10	IER10	8	8	2 个 ICLK		ICUb	173
0008 7211h	ICU	中断请求允许寄存器 11	IER11	8	8	2 个 ICLK			173
0008 7212h	ICU	中断请求允许寄存器 12	IER12	8	8	2 个 ICLK			173
0008 7213h	ICU	中断请求允许寄存器 13	IER13	8	8	2 个 ICLK			173
0008 7214h	ICU	中断请求允许寄存器 14	IER14	8	8	2 个 ICLK			173
0008 7215h	ICU	中断请求允许寄存器 15	IER15	8	8	2 个 ICLK			173
0008 7218h	ICU	中断请求允许寄存器 18	IER18	8	8	2 个 ICLK			173
0008 7219h	ICU	中断请求允许寄存器 19	IER19	8	8	2 个 ICLK			173
0008 721Ah	ICU	中断请求允许寄存器 1A	IER1A	8	8	2 个 ICLK			173
0008 721Bh	ICU	中断请求允许寄存器 1B	IER1B	8	8	2 个 ICLK			173
0008 721Ch	ICU	中断请求允许寄存器 1C	IER1C	8	8	2 个 ICLK			173
0008 721Dh	ICU	中断请求允许寄存器 1D	IER1D	8	8	2 个 ICLK			173
0008 721Eh	ICU	中断请求允许寄存器 1E	IER1E	8	8	2 个 ICLK			173
0008 721Fh	ICU	中断请求允许寄存器 1F	IER1F	8	8	2 个 ICLK			173
0008 72E0h	ICU	软件中断启动寄存器	SWINTR	8	8	2 个 ICLK			176
0008 72F0h	ICU	高速中断设定寄存器	FIR	16	16	2 个 ICLK			175
0008 7300h	ICU	中断源优先级寄存器 000	IPR000	8	8	2 个 ICLK			174
0008 7301h	ICU	中断源优先级寄存器 001	IPR001	8	8	2 个 ICLK			174
0008 7302h	ICU	中断源优先级寄存器 002	IPR002	8	8	2 个 ICLK			174
0008 7303h	ICU	中断源优先级寄存器 003	IPR003	8	8	2 个 ICLK			174
0008 7304h	ICU	中断源优先级寄存器 004	IPR004	8	8	2 个 ICLK		174	
0008 7305h	ICU	中断源优先级寄存器 000	IPR005	8	8	2 个 ICLK		174	
0008 7306h	ICU	中断源优先级寄存器 006	IPR006	8	8	2 个 ICLK		174	
0008 7307h	ICU	中断源优先级寄存器 007	IPR007	8	8	2 个 ICLK		174	
0008 7324h	ICU	中断源优先级寄存器 036	IPR036	8	8	2 个 ICLK		174	
0008 7327h	ICU	中断源优先级寄存器 039	IPR039	8	8	2 个 ICLK		174	
0008 7328h	ICU	中断源优先级寄存器 040	IPR040	8	8	2 个 ICLK		174	
0008 7329h	ICU	中断源优先级寄存器 041	IPR041	8	8	2 个 ICLK		174	
0008 7339h	ICU	中断源优先级寄存器 057	IPR057	8	8	2 个 ICLK		174	
0008 7340h	ICU	中断源优先级寄存器 064	IPR064	8	8	2 个 ICLK		174	
0008 7341h	ICU	中断源优先级寄存器 065	IPR065	8	8	2 个 ICLK		174	
0008 7342h	ICU	中断源优先级寄存器 066	IPR066	8	8	2 个 ICLK		174	
0008 7343h	ICU	中断源优先级寄存器 067	IPR067	8	8	2 个 ICLK		174	
0008 7344h	ICU	中断源优先级寄存器 068	IPR068	8	8	2 个 ICLK		174	
0008 7345h	ICU	中断源优先级寄存器 069	IPR069	8	8	2 个 ICLK		174	
0008 7366h	ICU	中断源优先级寄存器 102	IPR102	8	8	2 个 ICLK		174	
0008 7367h	ICU	中断源优先级寄存器 103	IPR103	8	8	2 个 ICLK		174	
0008 7372h	ICU	中断源优先级寄存器 114	IPR114	8	8	2 个 ICLK		174	
0008 737Ah	ICU	中断源优先级寄存器 122	IPR122	8	8	2 个 ICLK		174	
0008 737Eh	ICU	中断源优先级寄存器 126	IPR126	8	8	2 个 ICLK		174	
0008 7382h	ICU	中断源优先级寄存器 130	IPR130	8	8	2 个 ICLK		174	
0008 7385h	ICU	中断源优先级寄存器 133	IPR133	8	8	2 个 ICLK		174	
0008 7387h	ICU	中断源优先级寄存器 135	IPR135	8	8	2 个 ICLK		174	
0008 7389h	ICU	中断源优先级寄存器 137	IPR137	8	8	2 个 ICLK		174	
0008 738Bh	ICU	中断源优先级寄存器 139	IPR139	8	8	2 个 ICLK		174	
0008 738Dh	ICU	中断源优先级寄存器 141	IPR141	8	8	2 个 ICLK		174	
0008 7391h	ICU	中断源优先级寄存器 145	IPR145	8	8	2 个 ICLK		174	
0008 7392h	ICU	中断源优先级寄存器 146	IPR146	8	8	2 个 ICLK		174	
0008 7396h	ICU	中断源优先级寄存器 150	IPR150	8	8	2 个 ICLK		174	
0008 7397h	ICU	中断源优先级寄存器 151	IPR151	8	8	2 个 ICLK		174	
0008 739Ah	ICU	中断源优先级寄存器 154	IPR154	8	8	2 个 ICLK		174	
0008 739Eh	ICU	中断源优先级寄存器 158	IPR158	8	8	2 个 ICLK		174	
0008 73A1h	ICU	中断源优先级寄存器 161	IPR161	8	8	2 个 ICLK		174	
0008 73A3h	ICU	中断源优先级寄存器 163	IPR163	8	8	2 个 ICLK		174	
0008 73A5h	ICU	中断源优先级寄存器 165	IPR165	8	8	2 个 ICLK		174	
0008 73A6h	ICU	中断源优先级寄存器 166	IPR166	8	8	2 个 ICLK		174	
0008 73ABh	ICU	中断源优先级寄存器 171	IPR171	8	8	2 个 ICLK		174	
0008 73ACh	ICU	中断源优先级寄存器 172	IPR172	8	8	2 个 ICLK		174	
0008 73ADh	ICU	中断源优先级寄存器 173	IPR173	8	8	2 个 ICLK		174	
0008 73C2h	ICU	中断源优先级寄存器 194	IPR194	8	8	2 个 ICLK		174	
0008 73C6h	ICU	中断源优先级寄存器 198	IPR198	8	8	2 个 ICLK		174	
0008 73C7h	ICU	中断源优先级寄存器 199	IPR199	8	8	2 个 ICLK		174	

表 5.1 I/O 寄存器地址一览表 (7/17)

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		模块名	参照页数	
						ICLK ≥ PCLK 时	ICLK < PCLK 时			
0008 73C8h	ICU	中断源优先级寄存器 200	IPR200	8	8	2 个 ICLK		ICUb	174	
0008 73C9h	ICU	中断源优先级寄存器 201	IPR201	8	8	2 个 ICLK			174	
0008 73D6h	ICU	中断源优先级寄存器 214	IPR214	8	8	2 个 ICLK			174	
0008 73D9h	ICU	中断源优先级寄存器 217	IPR217	8	8	2 个 ICLK			174	
0008 73E2h	ICU	中断源优先级寄存器 226	IPR226	8	8	2 个 ICLK			174	
0008 73E5h	ICU	中断源优先级寄存器 229	IPR229	8	8	2 个 ICLK			174	
0008 73E8h	ICU	中断源优先级寄存器 232	IPR232	8	8	2 个 ICLK			174	
0008 73EBh	ICU	中断源优先级寄存器 235	IPR235	8	8	2 个 ICLK			174	
0008 73EEh	ICU	中断源优先级寄存器 238	IPR238	8	8	2 个 ICLK			174	
0008 73F1h	ICU	中断源优先级寄存器 241	IPR241	8	8	2 个 ICLK			174	
0008 73F4h	ICU	中断源优先级寄存器 244	IPR244	8	8	2 个 ICLK			174	
0008 73F7h	ICU	中断源优先级寄存器 247	IPR247	8	8	2 个 ICLK			174	
0008 73FAh	ICU	中断源优先级寄存器 250	IPR250	8	8	2 个 ICLK			174	
0008 7400h	ICU	DMAC 启动请求选择寄存器 0	DMRSR0	8	8	2 个 ICLK			177	
0008 7404h	ICU	DMAC 启动请求选择寄存器 1	DMRSR1	8	8	2 个 ICLK			177	
0008 7408h	ICU	DMAC 启动请求选择寄存器 2	DMRSR2	8	8	2 个 ICLK			177	
0008 740Ch	ICU	DMAC 启动请求选择寄存器 3	DMRSR3	8	8	2 个 ICLK			177	
0008 7500h	ICU	IRQ 控制寄存器 0	IRQCR0	8	8	2 个 ICLK			177	
0008 7501h	ICU	IRQ 控制寄存器 1	IRQCR1	8	8	2 个 ICLK			177	
0008 7502h	ICU	IRQ 控制寄存器 2	IRQCR2	8	8	2 个 ICLK			177	
0008 7503h	ICU	IRQ 控制寄存器 3	IRQCR3	8	8	2 个 ICLK		177		
0008 7504h	ICU	IRQ 控制寄存器 4	IRQCR4	8	8	2 个 ICLK		177		
0008 7505h	ICU	IRQ 控制寄存器 5	IRQCR5	8	8	2 个 ICLK		177		
0008 7510h	ICU	IRQ 引脚数字滤波器的允许寄存器 0	IRQFLTE0	8	8	2 个 ICLK		178		
0008 7514h	ICU	IRQ 引脚数字滤波器的设定寄存器 0	IRQFLTC0	16	16	2 个 ICLK		179		
0008 7580h	ICU	非屏蔽中断状态寄存器	NMISR	8	8	2 个 ICLK		180		
0008 7581h	ICU	非屏蔽中断允许寄存器	NMIER	8	8	2 个 ICLK		182		
0008 7582h	ICU	非屏蔽中断状态清除寄存器	NMICLR	8	8	2 个 ICLK		184		
0008 7583h	ICU	NMI 引脚中断控制寄存器	NMICR	8	8	2 个 ICLK		185		
0008 7590h	ICU	NMI 引脚数字滤波器的允许寄存器	NMIFLTE	8	8	2 个 ICLK		185		
0008 7594h	ICU	NMI 引脚数字滤波器的设定寄存器	NMIFLTC	8	8	2 个 ICLK		186		
0008 8000h	CMT	比较匹配定时器的启动寄存器 0	CMSTR0	16	16	2 ~ 3 个 PCLKB	2 个 ICLK	CMT	668	
0008 8002h	CMT0	比较匹配定时器的控制寄存器	CMCR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		669	
0008 8004h	CMT0	比较匹配定时器的计数器	CMCNT	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		670	
0008 8006h	CMT0	比较匹配定时器的常数寄存器	CMCOR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		670	
0008 8008h	CMT1	比较匹配定时器的控制寄存器	CMCR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		669	
0008 800Ah	CMT1	比较匹配定时器的计数器	CMCNT	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		670	
0008 800Ch	CMT1	比较匹配定时器的常数寄存器	CMCOR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		670	
0008 8010h	CMT	比较匹配定时器的启动寄存器 1	CMSTR1	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		668	
0008 8012h	CMT2	比较匹配定时器的控制寄存器	CMCR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		669	
0008 8014h	CMT2	比较匹配定时器的计数器	CMCNT	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		670	
0008 8016h	CMT2	比较匹配定时器的常数寄存器	CMCOR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		670	
0008 8018h	CMT3	比较匹配定时器的控制寄存器	CMCR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		669	
0008 801Ah	CMT3	比较匹配定时器的计数器	CMCNT	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		670	
0008 801Ch	CMT3	比较匹配定时器的常数寄存器	CMCOR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		670	
0008 8020h	WDT	WDT 刷新寄存器	WDTRR	8	8	2 ~ 3 个 PCLKB	2 个 ICLK		WDTA	676
0008 8022h	WDT	WDT 控制寄存器	WDTCR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK			677
0008 8024h	WDT	WDT 状态寄存器	WDTSR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		680	
0008 8026h	WDT	WDT 复位控制寄存器	WDTRCR	8	8	2 ~ 3 个 PCLKB	2 个 ICLK		681	
0008 8030h	IWDT	IWDT 刷新寄存器	IWDTRR	8	8	2 ~ 3 个 PCLKB	2 个 ICLK		IWDTa	690
0008 8032h	IWDT	IWDT 控制寄存器	IWDTCR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK			691
0008 8034h	IWDT	IWDT 状态寄存器	IWDTSR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK	694		
0008 8036h	IWDT	IWDT 复位控制寄存器	IWDTRCR	8	8	2 ~ 3 个 PCLKB	2 个 ICLK	695		
0008 8038h	IWDT	IWDT 计数停止控制寄存器	IWDTCSTPR	8	8	2 ~ 3 个 PCLKB	2 个 ICLK	695		
0008 8280h	CRC	CRC 控制寄存器	CRCCR	8	8	2 ~ 3 个 PCLKB	2 个 ICLK	CRC	962	
0008 8281h	CRC	CRC 数据输入寄存器	CRCDIR	8	8	2 ~ 3 个 PCLKB	2 个 ICLK		962	
0008 8282h	CRC	CRC 数据输出寄存器	CRCDOR	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		963	
0008 8300h	RIIC0	I ² C 总线控制寄存器 1	ICCR1	8	8	2 ~ 3 个 PCLKB	2 个 ICLK	RIIC	825	
0008 8301h	RIIC0	I ² C 总线控制寄存器 2	ICCR2	8	8	2 ~ 3 个 PCLKB	2 个 ICLK		827	
0008 8302h	RIIC0	I ² C 总线模式寄存器 1	ICMR1	8	8	2 ~ 3 个 PCLKB	2 个 ICLK		830	
0008 8303h	RIIC0	I ² C 总线模式寄存器 2	ICMR2	8	8	2 ~ 3 个 PCLKB	2 个 ICLK		831	
0008 8304h	RIIC0	I ² C 总线模式寄存器 3	ICMR3	8	8	2 ~ 3 个 PCLKB	2 个 ICLK	833		

表 5.1 I/O 寄存器地址一览表 (8/17)

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		模块名	参照页数	
						ICLK ≥ PCLK 时	ICLK < PCLK 时			
0008 8305h	RIIC0	I ² C 总线功能允许寄存器	ICFER	8	8	2~3 个 PCLKB	2 个 ICLK	RIIC	835	
0008 8306h	RIIC0	I ² C 总线状态允许寄存器	ICSER	8	8	2~3 个 PCLKB	2 个 ICLK		837	
0008 8307h	RIIC0	I ² C 总线中断允许寄存器	ICIER	8	8	2~3 个 PCLKB	2 个 ICLK		838	
0008 8308h	RIIC0	I ² C 总线状态寄存器 1	ICSR1	8	8	2~3 个 PCLKB	2 个 ICLK		840	
0008 8309h	RIIC0	I ² C 总线状态寄存器 2	ICSR2	8	8	2~3 个 PCLKB	2 个 ICLK		843	
0008 830Ah	RIIC0	从属地址寄存器 L0	SARL0	8	8	2~3 个 PCLKB	2 个 ICLK		846	
0008 830Ah	RIIC0	超时内部计数器 L	TMCNTL	16	16	2~3 个 PCLKB	2 个 ICLK		851	
0008 830Bh	RIIC0	从属地址寄存器 U0	SARU0	8	8	2~3 个 PCLKB	2 个 ICLK		847	
0008 830Bh	RIIC0	超时内部寄存器 U	TMCNTU	16	16	2~3 个 PCLKB	2 个 ICLK		851	
0008 830Ch	RIIC0	从属地址寄存器 L1	SARL1	8	8	2~3 个 PCLKB	2 个 ICLK		846	
0008 830Dh	RIIC0	从属地址寄存器 U1	SARU1	8	8	2~3 个 PCLKB	2 个 ICLK		847	
0008 830Eh	RIIC0	从属地址寄存器 L2	SARL2	8	8	2~3 个 PCLKB	2 个 ICLK		846	
0008 830Fh	RIIC0	从属地址寄存器 U2	SARU2	8	8	2~3 个 PCLKB	2 个 ICLK		847	
0008 8310h	RIIC0	I ² C 总线位速率低电平寄存器	ICBRL	8	8	2~3 个 PCLKB	2 个 ICLK		848	
0008 8311h	RIIC0	I ² C 总线位速率高电平寄存器	ICBRH	8	8	2~3 个 PCLKB	2 个 ICLK		848	
0008 8312h	RIIC0	I ² C 总线发送数据寄存器	ICDRT	8	8	2~3 个 PCLKB	2 个 ICLK		850	
0008 8313h	RIIC0	I ² C 总线接收数据寄存器	ICDRR	8	8	2~3 个 PCLKB	2 个 ICLK		850	
0008 8380h	RSPI0	RSPI 控制寄存器	SPCR	8	8	2~3 个 PCLKB	2 个 ICLK		RSPI	893
0008 8381h	RSPI0	RSPI 从属选择极性寄存器	SSLP	8	8	2~3 个 PCLKB	2 个 ICLK			894
0008 8382h	RSPI0	RSPI 引脚控制寄存器	SPPCR	8	8	2~3 个 PCLKB	2 个 ICLK			895
0008 8383h	RSPI0	RSPI 状态寄存器	SPSR	8	8	2~3 个 PCLKB	2 个 ICLK			896
0008 8384h	RSPI0	RSPI 数据寄存器	SPDR	32	16、32	2~3 个 PCLKB	2 个 ICLK			898
0008 8388h	RSPI0	RSPI 顺序控制寄存器	SPSCR	8	8	2~3 个 PCLKB	2 个 ICLK			901
0008 8389h	RSPI0	RSPI 顺序状态寄存器	SPSSR	8	8	2~3 个 PCLKB	2 个 ICLK	902		
0008 838Ah	RSPI0	RSPI 位速率寄存器	SPBR	8	8	2~3 个 PCLKB	2 个 ICLK	903		
0008 838Bh	RSPI0	RSPI 数据控制寄存器	SPDCR	8	8	2~3 个 PCLKB	2 个 ICLK	904		
0008 838Ch	RSPI0	RSPI 时钟延迟寄存器	SPCKD	8	8	2~3 个 PCLKB	2 个 ICLK	905		
0008 838Dh	RSPI0	RSPI 从属选择无效延迟寄存器	SSLND	8	8	2~3 个 PCLKB	2 个 ICLK	906		
0008 838Eh	RSPI0	RSPI 下次存取延迟寄存器	SPND	8	8	2~3 个 PCLKB	2 个 ICLK	907		
0008 838Fh	RSPI0	RSPI 控制寄存器 2	SPCR2	8	8	2~3 个 PCLKB	2 个 ICLK	908		
0008 8390h	RSPI0	RSPI 命令寄存器 0	SPCMD0	16	16	2~3 个 PCLKB	2 个 ICLK	909		
0008 8392h	RSPI0	RSPI 命令寄存器 1	SPCMD1	16	16	2~3 个 PCLKB	2 个 ICLK	909		
0008 8394h	RSPI0	RSPI 命令寄存器 2	SPCMD2	16	16	2~3 个 PCLKB	2 个 ICLK	909		
0008 8396h	RSPI0	RSPI 命令寄存器 3	SPCMD3	16	16	2~3 个 PCLKB	2 个 ICLK	909		
0008 8398h	RSPI0	RSPI 命令寄存器 4	SPCMD4	16	16	2~3 个 PCLKB	2 个 ICLK	909		
0008 839Ah	RSPI0	RSPI 命令寄存器 5	SPCMD5	16	16	2~3 个 PCLKB	2 个 ICLK	909		
0008 839Ch	RSPI0	RSPI 命令寄存器 6	SPCMD6	16	16	2~3 个 PCLKB	2 个 ICLK	909		
0008 839Eh	RSPI0	RSPI 命令寄存器 7	SPCMD7	16	16	2~3 个 PCLKB	2 个 ICLK	909		
0008 9000h	S12ADB	A/D 控制寄存器	ADCSR	16	16	2~3 个 PCLKB	2 个 ICLK	S12ADB		977
0008 9004h	S12ADB	A/D 通道选择寄存器 A	ADANSA	16	16	2~3 个 PCLKB	2 个 ICLK			980
0008 9008h	S12ADB	A/D 转换值加法运算模式选择寄存器	ADADS	16	16	2~3 个 PCLKB	2 个 ICLK		981	
0008 900Ch	S12ADB	A/D 转换值加法运算次数选择寄存器	ADADC	8	8	2~3 个 PCLKB	2 个 ICLK		982	
0008 900Eh	S12ADB	A/D 控制扩展寄存器	ADCER	16	16	2~3 个 PCLKB	2 个 ICLK		983	
0008 9010h	S12ADB	A/D 开始触发选择寄存器	ADSTRGR	8	8	2~3 个 PCLKB	2 个 ICLK		985	
0008 9014h	S12ADB	A/D 通道选择寄存器 B	ADANSB	16	16	2~3 个 PCLKB	2 个 ICLK		980	
0008 9018h	S12ADB	A/D 数据双重化寄存器	ADDBLDR	16	16	2~3 个 PCLKB	2 个 ICLK		973	
0008 901Eh	S12ADB	A/D 自诊断寄存器	ADRD	16	16	2~3 个 PCLKB	2 个 ICLK		976	
0008 9020h	S12ADB	A/D 数据寄存器 0	ADDR0	16	16	2~3 个 PCLKB	2 个 ICLK		973	
0008 9022h	S12ADB	A/D 数据寄存器 1	ADDR1	16	16	2~3 个 PCLKB	2 个 ICLK		973	
0008 9024h	S12ADB	A/D 数据寄存器 2	ADDR2	16	16	2~3 个 PCLKB	2 个 ICLK		973	
0008 9026h	S12ADB	A/D 数据寄存器 3	ADDR3	16	16	2~3 个 PCLKB	2 个 ICLK		973	
0008 9028h	S12ADB	A/D 数据寄存器 4	ADDR4	16	16	2~3 个 PCLKB	2 个 ICLK		973	
0008 902Ah	S12ADB	A/D 数据寄存器 5	ADDR5	16	16	2~3 个 PCLKB	2 个 ICLK		973	
0008 902Ch	S12ADB	A/D 数据寄存器 6	ADDR6	16	16	2~3 个 PCLKB	2 个 ICLK		973	
0008 902Eh	S12ADB	A/D 数据寄存器 7	ADDR7	16	16	2~3 个 PCLKB	2 个 ICLK		973	
0008 9060h	S12ADB	A/D 采样状态寄存器 0	ADSSTR0	8	8	2~3 个 PCLKB	2 个 ICLK		988	
0008 9066h	S12ADB	采样 & 保持电路控制寄存器	ADSHCR	16	16	2~3 个 PCLKB	2 个 ICLK		989	
0008 9073h	S12ADB	A/D 采样状态寄存器 1	ADSSTR1	8	8	2~3 个 PCLKB	2 个 ICLK		988	
0008 9074h	S12ADB	A/D 采样状态寄存器 2	ADSSTR2	8	8	2~3 个 PCLKB	2 个 ICLK		988	
0008 9075h	S12ADB	A/D 采样状态寄存器 3	ADSSTR3	8	8	2~3 个 PCLKB	2 个 ICLK		988	
0008 9076h	S12ADB	A/D 采样状态寄存器 4	ADSSTR4	8	8	2~3 个 PCLKB	2 个 ICLK		988	
0008 9077h	S12ADB	A/D 采样状态寄存器 5	ADSSTR5	8	8	2~3 个 PCLKB	2 个 ICLK		988	

表 5.1 I/O 寄存器地址一览表 (9/17)

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		模块名	参照页数	
						ICLK ≥ PCLK 时	ICLK < PCLK 时			
0008 9078h	S12ADB	A/D 采样状态寄存器 6	ADSSTR6	8	8	2~3 个 PCLKB	2 个 ICLK	S12ADB	988	
0008 9079h	S12ADB	A/D 采样状态寄存器 7	ADSSTR7	8	8	2~3 个 PCLKB	2 个 ICLK		988	
0008 9080h	S12ADB	A/D 分组扫描优先控制寄存器	ADGSPCR	16	16	2~3 个 PCLKB	2 个 ICLK		990	
0008 9084h	S12ADB	A/D 数据双重化寄存器 A	ADDBLDRA	16	16	2~3 个 PCLKB	2 个 ICLK		973	
0008 9086h	S12ADB	A/D 数据双重化寄存器 B	ADDBLDRB	16	16	2~3 个 PCLKB	2 个 ICLK		973	
0008 90E0h	S12ADB	比较器运行模式选择寄存器 0	ADCMPMD0	16	16	2~3 个 PCLKB	2 个 ICLK		992	
0008 90E2h	S12ADB	比较器运行模式选择寄存器 1	ADCMPMD1	16	16	2~3 个 PCLKB	2 个 ICLK		993	
0008 90E4h	S12ADB	比较器的滤波器模式寄存器 0	ADCMPNR0	16	16	2~3 个 PCLKB	2 个 ICLK		994	
0008 90E8h	S12ADB	比较器检测标志寄存器	ADCMPFR	8	8	2~3 个 PCLKB	2 个 ICLK		995	
0008 90EAh	S12ADB	比较器中断选择寄存器	ADCMPSEL	16	16	2~3 个 PCLKB	2 个 ICLK		996	
0008 A000h	SCI0	串行模式寄存器	SMR	8	8	2~3 个 PCLKB	2 个 ICLK	SCIC、SCID	710	
0008 A001h	SCI0	位速率寄存器	BRR	8	8	2~3 个 PCLKB	2 个 ICLK		723	
0008 A002h	SCI0	串行控制寄存器	SCR	8	8	2~3 个 PCLKB	2 个 ICLK		714	
0008 A003h	SCI0	发送数据寄存器	TDR	8	8	2~3 个 PCLKB	2 个 ICLK		709	
0008 A004h	SCI0	串行状态寄存器	SSR	8	8	2~3 个 PCLKB	2 个 ICLK		718	
0008 A005h	SCI0	接收数据寄存器	RDR	8	8	2~3 个 PCLKB	2 个 ICLK		709	
0008 A006h	SCI0	智能卡模式寄存器	SCMR	8	8	2~3 个 PCLKB	2 个 ICLK		722	
0008 A007h	SCI0	串行扩展模式寄存器	SEMR	8	8	2~3 个 PCLKB	2 个 ICLK		731	
0008 A008h	SCI0	噪声滤波器的设定寄存器	SNFR	8	8	2~3 个 PCLKB	2 个 ICLK		732	
0008 A009h	SCI0	I ² C 模式寄存器 1	SIMR1	8	8	2~3 个 PCLKB	2 个 ICLK		733	
0008 A00Ah	SCI0	I ² C 模式寄存器 2	SIMR2	8	8	2~3 个 PCLKB	2 个 ICLK		734	
0008 A00Bh	SCI0	I ² C 模式寄存器 3	SIMR3	8	8	2~3 个 PCLKB	2 个 ICLK		735	
0008 A00Ch	SCI0	I ² C 状态寄存器	SISR	8	8	2~3 个 PCLKB	2 个 ICLK		737	
0008 A00Dh	SCI0	SPI 模式寄存器	SPMR	8	8	2~3 个 PCLKB	2 个 ICLK		738	
0008 A020h	SCI1	串行模式寄存器	SMR	8	8	2~3 个 PCLKB	2 个 ICLK		710	
0008 A021h	SCI1	位速率寄存器	BRR	8	8	2~3 个 PCLKB	2 个 ICLK		723	
0008 A022h	SCI1	串行控制寄存器	SCR	8	8	2~3 个 PCLKB	2 个 ICLK		714	
0008 A023h	SCI1	发送数据寄存器	TDR	8	8	2~3 个 PCLKB	2 个 ICLK		709	
0008 A024h	SCI1	串行状态寄存器	SSR	8	8	2~3 个 PCLKB	2 个 ICLK		718	
0008 A025h	SCI1	接收数据寄存器	RDR	8	8	2~3 个 PCLKB	2 个 ICLK		709	
0008 A026h	SCI1	智能卡模式寄存器	SCMR	8	8	2~3 个 PCLKB	2 个 ICLK		722	
0008 A027h	SCI1	串行扩展模式寄存器	SEMR	8	8	2~3 个 PCLKB	2 个 ICLK		731	
0008 A028h	SCI1	噪声滤波器的设定寄存器	SNFR	8	8	2~3 个 PCLKB	2 个 ICLK		732	
0008 A029h	SCI1	I ² C 模式寄存器 1	SIMR1	8	8	2~3 个 PCLKB	2 个 ICLK		733	
0008 A02Ah	SCI1	I ² C 模式寄存器 2	SIMR2	8	8	2~3 个 PCLKB	2 个 ICLK		734	
0008 A02Bh	SCI1	I ² C 模式寄存器 3	SIMR3	8	8	2~3 个 PCLKB	2 个 ICLK		735	
0008 A02Ch	SCI1	I ² C 状态寄存器	SISR	8	8	2~3 个 PCLKB	2 个 ICLK		737	
0008 A02Dh	SCI1	SPI 模式寄存器	SPMR	8	8	2~3 个 PCLKB	2 个 ICLK		738	
0008 B000h	CAC	CAC 控制寄存器 0	CACR0	8	8	2~3 个 PCLKB	2 个 ICLK		CAC	124
0008 B001h	CAC	CAC 控制寄存器 1	CACR1	8	8	2~3 个 PCLKB	2 个 ICLK			125
0008 B002h	CAC	CAC 控制寄存器 2	CACR2	8	8	2~3 个 PCLKB	2 个 ICLK			126
0008 B003h	CAC	CAC 中断控制寄存器	CAICR	8	8	2~3 个 PCLKB	2 个 ICLK			127
0008 B004h	CAC	CAC 状态寄存器	CASTR	8	8	2~3 个 PCLKB	2 个 ICLK	128		
0008 B006h	CAC	CAC 上限值设定寄存器	CAULVR	16	16	2~3 个 PCLKB	2 个 ICLK	129		
0008 0008h	CAC	CAC 下限值设定寄存器	CALLVR	16	16	2~3 个 PCLKB	2 个 ICLK	129		
0008 000Ah	CAC	CAC 计数器缓冲寄存器	CACNTBR	16	16	2~3 个 PCLKB	2 个 ICLK	129		
0008 B080h	DOC	DOC 控制寄存器	DOCR	8	8	2~3 个 PCLKB	2 个 ICLK	DOC		1033
0008 B082h	DOC	DOC 数据输入寄存器	DODIR	16	16	2~3 个 PCLKB	2 个 ICLK		1034	
0008 B084h	DOC	DOC 数据设定寄存器	DODSR	16	16	2~3 个 PCLKB	2 个 ICLK		1034	
0008 B300h	SCI12	串行模式寄存器	SMR	8	8	2~3 个 PCLKB	2 个 ICLK	SCIC、SCID	710	
0008 B301h	SCI12	位速率寄存器	BRR	8	8	2~3 个 PCLKB	2 个 ICLK		723	
0008 B302h	SCI12	串行控制寄存器	SCR	8	8	2~3 个 PCLKB	2 个 ICLK		714	
0008 B303h	SCI12	发送数据寄存器	TDR	8	8	2~3 个 PCLKB	2 个 ICLK		709	
0008 B304h	SCI12	串行状态寄存器	SSR	8	8	2~3 个 PCLKB	2 个 ICLK		718	
0008 B305h	SCI12	接收数据寄存器	RDR	8	8	2~3 个 PCLKB	2 个 ICLK		709	
0008 B306h	SCI12	智能卡模式寄存器	SCMR	8	8	2~3 个 PCLKB	2 个 ICLK		722	
0008 B307h	SCI12	串行扩展模式寄存器	SEMR	8	8	2~3 个 PCLKB	2 个 ICLK		731	
0008 B308h	SCI12	噪声滤波器的设定寄存器	SNFR	8	8	2~3 个 PCLKB	2 个 ICLK		732	
0008 B309h	SCI12	I ² C 模式寄存器 1	SIMR1	8	8	2~3 个 PCLKB	2 个 ICLK		733	
0008 B30Ah	SCI12	I ² C 模式寄存器 2	SIMR2	8	8	2~3 个 PCLKB	2 个 ICLK		734	
0008 B30Bh	SCI12	I ² C 模式寄存器 3	SIMR3	8	8	2~3 个 PCLKB	2 个 ICLK		735	
0008 B30Ch	SCI12	I ² C 状态寄存器	SIS	8	8	2~3 个 PCLKB	2 个 ICLK		737	

表 5.1 I/O 寄存器地址一览表 (10/17)

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		模块名	参照页数	
						ICLK ≥ PCLK 时	ICLK < PCLK 时			
0008 B30Dh	SCI12	SPI 模式寄存器	SPMR	8	8	2~3 个 PCLKB	2 个 ICLK	SCIC、SCID	738	
0008 B320h	SCI12	扩展串行模式有效寄存器	ESMER	8	8	2~3 个 PCLKB	2 个 ICLK		740	
0008 B321h	SCI12	控制寄存器 0	CR0	8	8	2~3 个 PCLKB	2 个 ICLK		740	
0008 B322h	SCI12	控制寄存器 1	CR1	8	8	2~3 个 PCLKB	2 个 ICLK		741	
0008 B323h	SCI12	控制寄存器 2	CR2	8	8	2~3 个 PCLKB	2 个 ICLK		742	
0008 B324h	SCI12	控制寄存器 3	CR3	8	8	2~3 个 PCLKB	2 个 ICLK		743	
0008 B325h	SCI12	端口控制寄存器	PCR	8	8	2~3 个 PCLKB	2 个 ICLK		743	
0008 B326h	SCI12	中断控制寄存器	ICR	8	8	2~3 个 PCLKB	2 个 ICLK		744	
0008 B327h	SCI12	状态寄存器	STR	8	8	2~3 个 PCLKB	2 个 ICLK		745	
0008 B328h	SCI12	状态清除寄存器	STCR	8	8	2~3 个 PCLKB	2 个 ICLK		746	
0008 B329h	SCI12	Control Field 0 数据寄存器	CFODR	8	8	2~3 个 PCLKB	2 个 ICLK		746	
0008 B32Ah	SCI12	Control Field 0 比较允许寄存器	CFOCR	8	8	2~3 个 PCLKB	2 个 ICLK		747	
0008 B32Bh	SCI12	Control Field 0 接收数据寄存器	CFORR	8	8	2~3 个 PCLKB	2 个 ICLK		747	
0008 B32Ch	SCI12	主 Control Field 1 数据寄存器	PCF1DR	8	8	2~3 个 PCLKB	2 个 ICLK		747	
0008 B32Dh	SCI12	次 Control Field 1 数据寄存器	SCF1DR	8	8	2~3 个 PCLKB	2 个 ICLK		748	
0008 B32Eh	SCI12	Control Field 1 比较允许寄存器	CF1CR	8	8	2~3 个 PCLKB	2 个 ICLK		748	
0008 B32Fh	SCI12	Control Field 1 接收数据寄存器	CF1RR	8	8	2~3 个 PCLKB	2 个 ICLK		748	
0008 B330h	SCI12	定时器的控制寄存器	TCR	8	8	2~3 个 PCLKB	2 个 ICLK		749	
0008 B331h	SCI12	定时器的模式寄存器	TMR	8	8	2~3 个 PCLKB	2 个 ICLK		749	
0008 B332h	SCI12	定时器的预分频寄存器	TPRE	8	8	2~3 个 PCLKB	2 个 ICLK		750	
0008 B333h	SCI12	定时器的计数寄存器	TCNT	8	8	2~3 个 PCLKB	2 个 ICLK		750	
0008 C000h	PORT0	端口方向寄存器	PDR	8	8	2~3 个 PCLKB	2 个 ICLK		I/O 端口	303
0008 C001h	PORT1	端口方向寄存器	PDR	8	8	2~3 个 PCLKB	2 个 ICLK			303
0008 C002h	PORT2	端口方向寄存器	PDR	8	8	2~3 个 PCLKB	2 个 ICLK			303
0008 C003h	PORT3	端口方向寄存器	PDR	8	8	2~3 个 PCLKB	2 个 ICLK			303
0008 C007h	PORT7	端口方向寄存器	PDR	8	8	2~3 个 PCLKB	2 个 ICLK	303		
0008 C009h	PORT9	端口方向寄存器	PDR	8	8	2~3 个 PCLKB	2 个 ICLK	303		
0008 C00Ah	PORTA	端口方向寄存器	PDR	8	8	2~3 个 PCLKB	2 个 ICLK	303		
0008 C00Bh	PORTB	端口方向寄存器	PDR	8	8	2~3 个 PCLKB	2 个 ICLK	303		
0008 C00Dh	PORTD	端口方向寄存器	PDR	8	8	2~3 个 PCLKB	2 个 ICLK	303		
0008 C020h	PORT0	端口输出数据寄存器	PODR	8	8	2~3 个 PCLKB	2 个 ICLK	304		
0008 C021h	PORT1	端口输出数据寄存器	PODR	8	8	2~3 个 PCLKB	2 个 ICLK	304		
0008 C022h	PORT2	端口输出数据寄存器	PODR	8	8	2~3 个 PCLKB	2 个 ICLK	304		
0008 C023h	PORT3	端口输出数据寄存器	PODR	8	8	2~3 个 PCLKB	2 个 ICLK	304		
0008 C027h	PORT7	端口输出数据寄存器	PODR	8	8	2~3 个 PCLKB	2 个 ICLK	304		
0008 C029h	PORT9	端口输出数据寄存器	PODR	8	8	2~3 个 PCLKB	2 个 ICLK	304		
0008 C02Ah	PORTA	端口输出数据寄存器	PODR	8	8	2~3 个 PCLKB	2 个 ICLK	304		
0008 C02Bh	PORTB	端口输出数据寄存器	PODR	8	8	2~3 个 PCLKB	2 个 ICLK	304		
0008 C02Dh	PORTD	端口输出数据寄存器	PODR	8	8	2~3 个 PCLKB	2 个 ICLK	304		
0008 C040h	PORT0	端口输入数据寄存器	PIDR	8	8	2~3 个 PCLKB	2 个 ICLK	305		
0008 C041h	PORT1	端口输入数据寄存器	PIDR	8	8	2~3 个 PCLKB	2 个 ICLK	305		
0008 C042h	PORT2	端口输入数据寄存器	PIDR	8	8	2~3 个 PCLKB	2 个 ICLK	305		
0008 C043h	PORT3	端口输入数据寄存器	PIDR	8	8	2~3 个 PCLKB	2 个 ICLK	305		
0008 C044h	PORT4	端口输入数据寄存器	PIDR	8	8	2~3 个 PCLKB	2 个 ICLK	305		
0008 C047h	PORT7	端口输入数据寄存器	PIDR	8	8	2~3 个 PCLKB	2 个 ICLK	305		
0008 C049h	PORT9	端口输入数据寄存器	PIDR	8	8	2~3 个 PCLKB	2 个 ICLK	305		
0008 C04Ah	PORTA	端口输入数据寄存器	PIDR	8	8	2~3 个 PCLKB	2 个 ICLK	305		
0008 C04Bh	PORTB	端口输入数据寄存器	PIDR	8	8	2~3 个 PCLKB	2 个 ICLK	305		
0008 C04Dh	PORTD	端口输入数据寄存器	PIDR	8	8	2~3 个 PCLKB	2 个 ICLK	305		
0008 C04Eh	PORTE	端口输入数据寄存器	PIDR	8	8	2~3 个 PCLKB	2 个 ICLK	305		
0008 C060h	PORT0	端口模式寄存器	PMR	8	8	2~3 个 PCLKB	2 个 ICLK	306		
0008 C061h	PORT1	端口模式寄存器	PMR	8	8	2~3 个 PCLKB	2 个 ICLK	306		
0008 C062h	PORT2	端口模式寄存器	PMR	8	8	2~3 个 PCLKB	2 个 ICLK	306		
0008 C063h	PORT3	端口模式寄存器	PMR	8	8	2~3 个 PCLKB	2 个 ICLK	306		
0008 C067h	PORT7	端口模式寄存器	PMR	8	8	2~3 个 PCLKB	2 个 ICLK	306		
0008 C069h	PORT9	端口模式寄存器	PMR	8	8	2~3 个 PCLKB	2 个 ICLK	306		
0008 C06Ah	PORTA	端口模式寄存器	PMR	8	8	2~3 个 PCLKB	2 个 ICLK	306		
0008 C06Bh	PORTB	端口模式寄存器	PMR	8	8	2~3 个 PCLKB	2 个 ICLK	306		
0008 C06Dh	PORTD	端口模式寄存器	PMR	8	8	2~3 个 PCLKB	2 个 ICLK	306		
0008 C06Eh	PORTE	端口模式寄存器	PMR	8	8	2~3 个 PCLKB	2 个 ICLK	306		
0008 C085h	PORT2	漏极开路控制寄存器 1	ODR1	8	8	2~3 个 PCLKB	2 个 ICLK	307		
0008 C086h	PORT3	漏极开路控制寄存器 0	ODR0	8	8	2~3 个 PCLKB	2 个 ICLK	307		

表 5.1 I/O 寄存器地址一览表 (11/17)

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		模块名	参照页数
						ICLK ≥ PCLK 时	ICLK < PCLK 时		
0008 C092h	PORT9	漏极开路控制寄存器 0	ODR0	8	8	2~3 个 PCLKB	2 个 ICLK	I/O 端口	307
0008 C093h	PORT9	漏极开路控制寄存器 1	ODR1	8	8	2~3 个 PCLKB	2 个 ICLK		307
0008 C096h	PORTB	漏极开路控制寄存器 0	ODR0	8	8	2~3 个 PCLKB	2 个 ICLK		307
0008 C097h	PORTB	漏极开路控制寄存器 1	ODR1	8	8	2~3 个 PCLKB	2 个 ICLK		307
0008 C09Ah	PORTD	漏极开路控制寄存器 0	ODR0	8	8	2~3 个 PCLKB	2 个 ICLK		307
0008 C09Bh	PORTD	漏极开路控制寄存器 1	ODR1	8	8	2~3 个 PCLKB	2 个 ICLK		307
0008 C11Fh	MPC	写保护寄存器	PWPR	8	8	2~3 个 PCLKB	2 个 ICLK	MPC	312
0008 C140h	MPC	P00 引脚功能控制寄存器	P00PFS	8	8	2~3 个 PCLKB	2 个 ICLK		313
0008 C141h	MPC	P01 引脚功能控制寄存器	P01PFS	8	8	2~3 个 PCLKB	2 个 ICLK		313
0008 C148h	MPC	P10 引脚功能控制寄存器	P10PFS	8	8	2~3 个 PCLKB	2 个 ICLK		314
0008 C149h	MPC	P11 引脚功能控制寄存器	P11PFS	8	8	2~3 个 PCLKB	2 个 ICLK		314
0008 C152h	MPC	P22 引脚功能控制寄存器	P22PFS	8	8	2~3 个 PCLKB	2 个 ICLK		315
0008 C153h	MPC	P23 引脚功能控制寄存器	P23PFS	8	8	2~3 个 PCLKB	2 个 ICLK		315
0008 C154h	MPC	P24 引脚功能控制寄存器	P24PFS	8	8	2~3 个 PCLKB	2 个 ICLK		315
0008 C158h	MPC	P30 引脚功能控制寄存器	P30PFS	8	8	2~3 个 PCLKB	2 个 ICLK		316
0008 C159h	MPC	P31 引脚功能控制寄存器	P31PFS	8	8	2~3 个 PCLKB	2 个 ICLK		316
0008 C15Ah	MPC	P32 引脚功能控制寄存器	P32PFS	8	8	2~3 个 PCLKB	2 个 ICLK		316
0008 C15Bh	MPC	P33 引脚功能控制寄存器	P33PFS	8	8	2~3 个 PCLKB	2 个 ICLK		316
0008 C160h	MPC	P40 引脚功能控制寄存器	P40PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C161h	MPC	P41 引脚功能控制寄存器	P41PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C162h	MPC	P42 引脚功能控制寄存器	P42PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C163h	MPC	P43 引脚功能控制寄存器	P43PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C164h	MPC	P44 引脚功能控制寄存器	P44PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C165h	MPC	P45 引脚功能控制寄存器	P45PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C166h	MPC	P46 引脚功能控制寄存器	P46PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C167h	MPC	P47 引脚功能控制寄存器	P47PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C178h	MPC	P70 引脚功能控制寄存器	P70PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C179h	MPC	P71 引脚功能控制寄存器	P71PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C17Ah	MPC	P72 引脚功能控制寄存器	P72PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C17Bh	MPC	P73 引脚功能控制寄存器	P73PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C17Ch	MPC	P74 引脚功能控制寄存器	P74PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C17Dh	MPC	P75 引脚功能控制寄存器	P75PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C17Eh	MPC	P76 引脚功能控制寄存器	P76PFS	8	8	2~3 个 PCLKB	2 个 ICLK		317
0008 C189h	MPC	P91 引脚功能控制寄存器	P91PFS	8	8	2~3 个 PCLKB	2 个 ICLK		318
0008 C18Ah	MPC	P92 引脚功能控制寄存器	P92PFS	8	8	2~3 个 PCLKB	2 个 ICLK		318
0008 C18Bh	MPC	P93 引脚功能控制寄存器	P93PFS	8	8	2~3 个 PCLKB	2 个 ICLK		318
0008 C18Ch	MPC	P94 引脚功能控制寄存器	P94PFS	8	8	2~3 个 PCLKB	2 个 ICLK		318
0008 C192h	MPC	PA2 引脚功能控制寄存器	PA2PFS	8	8	2~3 个 PCLKB	2 个 ICLK		319
0008 C193h	MPC	PA3 引脚功能控制寄存器	PA3PFS	8	8	2~3 个 PCLKB	2 个 ICLK		319
0008 C194h	MPC	PA4 引脚功能控制寄存器	PA4PFS	8	8	2~3 个 PCLKB	2 个 ICLK		319
0008 C195h	MPC	PA5 引脚功能控制寄存器	PA5PFS	8	8	2~3 个 PCLKB	2 个 ICLK		319
0008 C198h	MPC	PB0 引脚功能控制寄存器	PB0PFS	8	8	2~3 个 PCLKB	2 个 ICLK		320
0008 C199h	MPC	PB1 引脚功能控制寄存器	PB1PFS	8	8	2~3 个 PCLKB	2 个 ICLK		320
0008 C19Ah	MPC	PB2 引脚功能控制寄存器	PB2PFS	8	8	2~3 个 PCLKB	2 个 ICLK		320
0008 C19Bh	MPC	PB3 引脚功能控制寄存器	PB3PFS	8	8	2~3 个 PCLKB	2 个 ICLK		320
0008 C19Ch	MPC	PB4 引脚功能控制寄存器	PB4PFS	8	8	2~3 个 PCLKB	2 个 ICLK		320
0008 C19Dh	MPC	PB5 引脚功能控制寄存器	PB5PFS	8	8	2~3 个 PCLKB	2 个 ICLK		320
0008 C19Eh	MPC	PB6 引脚功能控制寄存器	PB6PFS	8	8	2~3 个 PCLKB	2 个 ICLK		320
0008 C19Fh	MPC	PB7 引脚功能控制寄存器	PB7PFS	8	8	2~3 个 PCLKB	2 个 ICLK		320
0008 C1ABh	MPC	PD3 引脚功能控制寄存器	PD3PFS	8	8	2~3 个 PCLKB	2 个 ICLK		322
0008 C1ACh	MPC	PD4 引脚功能控制寄存器	PD4PFS	8	8	2~3 个 PCLKB	2 个 ICLK		322
0008 C1ADh	MPC	PD5 引脚功能控制寄存器	PD5PFS	8	8	2~3 个 PCLKB	2 个 ICLK		322
0008 C1AEh	MPC	PD6 引脚功能控制寄存器	PD6PFS	8	8	2~3 个 PCLKB	2 个 ICLK		322
0008 C1AFh	MPC	PD7 引脚功能控制寄存器	PD7PFS	8	8	2~3 个 PCLKB	2 个 ICLK		322
0008 C1B2h	MPC	PE2 引脚功能控制寄存器	PE2PFS	8	8	2~3 个 PCLKB	2 个 ICLK		322
0008 C280h	SYSTEM	深度待机控制寄存器	DPSBYCR	8	8	4~5 个 PCLKB	2~3 个 ICLK	低功耗功能	142
0008 C282h	SYSTEM	深度待机中允许寄存器 0	DPSIER0	8	8	4~5 个 PCLKB	2~3 个 ICLK		143
0008 C284h	SYSTEM	深度待机中允许寄存器 2	DPSIER2	8	8	4~5 个 PCLKB	2~3 个 ICLK		144
0008 C286h	SYSTEM	深度待机中标志寄存器 0	DPSIFR0	8	8	4~5 个 PCLKB	2~3 个 ICLK		145
0008 C288h	SYSTEM	深度待机中标志寄存器 2	DPSIFR2	8	8	4~5 个 PCLKB	2~3 个 ICLK		146
0008 C28Ah	SYSTEM	深度待机中断边沿寄存器 0	DPSIEGR0	8	8	4~5 个 PCLKB	2~3 个 ICLK		147
0008 C28Ch	SYSTEM	深度待机中断边沿寄存器 2	DPSIEGR2	8	8	4~5 个 PCLKB	2~3 个 ICLK		148

表 5.1 I/O 寄存器地址一览表 (12/17)

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		模块名	参照页数
						ICLK ≥ PCLK 时	ICLK < PCLK 时		
0008 C290h	SYSTEM	复位状态寄存器 0	RSTSR0	8	8	4 ~ 5 个 PCLKB	2 ~ 3 个 ICLK	复位	71
0008 C291h	SYSTEM	复位状态寄存器 1	RSTSR1	8	8	4 ~ 5 个 PCLKB	2 ~ 3 个 ICLK		73
0008 C293h	SYSTEM	主时钟振荡器的强制振荡控制寄存器	MOFCR	8	8	4 ~ 5 个 PCLKB	2 ~ 3 个 ICLK	时钟发生电路	115
0008 C296h	FLASH	闪存编程 / 擦除保护寄存器	FWEPROR	8	8	4 ~ 5 个 PCLKB	2 ~ 3 个 ICLK	ROM	1041
0008 C297h	SYSTEM	电压监视电路的控制寄存器	LVCMPCR	8	8	4 ~ 5 个 PCLKB	2 ~ 3 个 ICLK	LVDA	93
0008 C298h	SYSTEM	电压检测电平选择寄存器	LVDLVL	8	8	4 ~ 5 个 PCLKB	2 ~ 3 个 ICLK		93
0008 C29Ah	SYSTEM	电压监视 1 电路的控制寄存器 0	LVD1CR0	8	8	4 ~ 5 个 PCLKB	2 ~ 3 个 ICLK		94
0008 C29Bh	SYSTEM	电压监视 2 电路的控制寄存器 0	LVD2CR0	8	8	4 ~ 5 个 PCLKB	2 ~ 3 个 ICLK		96
0008 C2A0h ~ 0008 C2Bfh	SYSTEM	深度待机备用寄存器 0 ~ 31	DPSBKR0 ~ 31	8	8	4 ~ 5 个 PCLKB	2 ~ 3 个 ICLK	低功耗功能	148
0008 C330h	ICU	组 12 中断源寄存器	GRP12	32	32	1 ~ 2 个 PCLKB	2 个 ICLK	ICUb	187
0008 C370h	ICU	组 12 中断允许寄存器	GEN12	32	32	1 ~ 2 个 PCLKB	2 个 ICLK		188
0008 C4C0h	POE	输入电平控制 / 状态寄存器 1	ICSR1	16	8、16	2 ~ 3 个 PCLKB	2 个 ICLK	POE3	522
0008 C4C2h	POE	输出电平控制 / 状态寄存器 1	OCSR1	16	8、16	2 ~ 3 个 PCLKB	2 个 ICLK		523
0008 C4C8h	POE	输入电平控制 / 状态寄存器 3	ICSR3	16	8、16	2 ~ 3 个 PCLKB	2 个 ICLK		526
0008 C4CAh	POE	软件端口输出允许寄存器	SPOER	8	8	2 ~ 3 个 PCLKB	2 个 ICLK		529
0008 C4CBh	POE	端口输出允许控制寄存器 1	POECR1	8	8	2 ~ 3 个 PCLKB	2 个 ICLK		531
0008 C4CCh	POE	端口输出允许控制寄存器 2	POECR2	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		532
0008 C4CEh	POE	端口输出允许控制寄存器 3	POECR3	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		533
0008 C4D0h	POE	端口输出允许控制寄存器 4	POECR4	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		534
0008 C4D2h	POE	端口输出允许控制寄存器 5	POECR5	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		535
0008 C4D4h	POE	端口输出允许控制寄存器 6	POECR6	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		536
0008 C4D6h	POE	输入电平控制 / 状态寄存器 4	ICSR4	16	8、16	2 ~ 3 个 PCLKB	2 个 ICLK		527
0008 C4D8h	POE	输入电平控制 / 状态寄存器 5	ICSR5	16	8、16	2 ~ 3 个 PCLKB	2 个 ICLK		528
0008 C4DAh	POE	有效电平寄存器 1	ALR1	16	8、16	2 ~ 3 个 PCLKB	2 个 ICLK		524
0008 C4DCh	POE	输入电平控制 / 状态寄存器 6	ICSR6	16	16	2 ~ 3 个 PCLKB	2 个 ICLK		538
000C 1200h	MTU3	定时器的控制寄存器	TCR	8	8、16、32	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK	MTU3	330
000C 1201h	MTU4	定时器的控制寄存器	TCR	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		330
000C 1202h	MTU3	定时器的模式寄存器 1	TMDR1	8	8、16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		334
000C 1203h	MTU4	定时器的模式寄存器 1	TMDR1	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		334
000C 1204h	MTU3	定时器的 I/O 控制寄存器 H	TIORH	8	8、16、32	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		336
000C 1205h	MTU3	定时器的 I/O 控制寄存器 L	TIORL	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		337
000C 1206h	MTU4	定时器的 I/O 控制寄存器 H	TIORH	8	8、16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		336
000C 1207h	MTU4	定时器的 I/O 控制寄存器 L	TIORL	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		337
000C 1208h	MTU3	定时器的中断允许寄存器	TIER	8	8、16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		352
000C 1209h	MTU4	定时器的中断允许寄存器	TIER	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		352
000C 120Ah	MTU	定时器的输出主控允许寄存器 A	TOERA	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		370
000C 120Dh	MTU	定时器的门控寄存器 A	TGCR A	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		377
000C 120Eh	MTU	定时器的输出控制寄存器 1A	TOCR1A	8	8、16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		372
000C 120Fh	MTU	定时器的输出控制寄存器 2A	TOCR2A	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		374
000C 1210h	MTU3	定时器的计数器	TCNT	16	16、32	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		361
000C 1212h	MTU4	定时器的计数器	TCNT	16	16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		361
000C 1214h	MTU	定时器的周期数据寄存器 A	TCDRA	16	16、32	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		378
000C 1216h	MTU	定时器的死区时间数据寄存器	TDDRA	16	16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		379
000C 1218h	MTU3	定时器的通用寄存器 A	TGRA	16	16、32	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		362
000C 121Ah	MTU3	定时器的通用寄存器 B	TGRB	16	16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		362
000C 121Ch	MTU4	定时器的通用寄存器 A	TGRA	16	16、32	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		362
000C 121Eh	MTU4	定时器的通用寄存器 B	TGRB	16	16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		362
000C 1220h	MTU	定时器的副计数器 A	TCNTSA	16	16、32	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		378
000C 1222h	MTU	定时器的周期缓冲寄存器 A	TCBRA	16	16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		379
000C 1224h	MTU3	定时器的通用寄存器 C	TGRC	16	16、32	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		362
000C 1226h	MTU3	定时器的通用寄存器 D	TGRD	16	16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		362
000C 1228h	MTU4	定时器的通用寄存器 C	TGRC	16	16、32	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		362
000C 122Ah	MTU4	定时器的通用寄存器 D	TGRD	16	16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		362
000C 122Ch	MTU3	定时器的状态寄存器	TSR	8	8、16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		355
000C 122Dh	MTU4	定时器的状态寄存器	TSR	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		355
000C 1230h	MTU	定时器的中断减少设定寄存器 1A	TITCR1A	8	8、16	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		389
000C 1231h	MTU	定时器的中断减少次数计数器 1A	TITCNT1A	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		391
000C 1232h	MTU	定时器的缓冲传送设定寄存器 A	TBTERA	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		380
000C 1234h	MTU	定时器的死区时间允许寄存器 A	TDERA	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		379
000C 1236h	MTU	定时器的输出电平缓冲寄存器 A	TOLBRA	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		376

表 5.1 I/O 寄存器地址一览表 (13/17)

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		模块名	参照页数
						ICLK ≥ PCLK 时	ICLK < PCLK 时		
000C 1238h	MTU3	定时器的缓冲运行传送模式寄存器	TBTM	8	8、16	4~5 个 PCLKA	2~3 个 ICLK	MTU3	359
000C 1239h	MTU4	定时器的缓冲运行传送模式寄存器	TBTM	8	8	4~5 个 PCLKA	2~3 个 ICLK		359
000C 123Ah	MTU	定时器的中断减少模式寄存器 A	TITMRA	8	8	4~5 个 PCLKA	2~3 个 ICLK		388
000C 123Bh	MTU	定时器的中断减少设定寄存器 2A	TITCR2A	8	8	4~5 个 PCLKA	2~3 个 ICLK		393
000C 123Ch	MTU	定时器的中断减少次数计数器 2A	TITCNT2A	8	8	4~5 个 PCLKA	2~3 个 ICLK		395
000C 1240h	MTU4	定时器的 A/D 转换开始请求控制寄存器	TADCR	16	16	4~5 个 PCLKA	2~3 个 ICLK		383
000C 1244h	MTU4	定时器的 A/D 转换开始请求周期设定寄存器 A	TADCORA	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		387
000C 1246h	MTU4	定时器的 A/D 转换开始请求周期设定寄存器 B	TADCORB	16	16	4~5 个 PCLKA	2~3 个 ICLK		387
000C 1248h	MTU4	定时器的 A/D 转换开始请求周期设定缓冲寄存器 A	TADCOBRA	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		387
000C 124Ah	MTU4	定时器的 A/D 转换开始请求周期设定缓冲寄存器 B	TADCOBRB	16	16	4~5 个 PCLKA	2~3 个 ICLK		387
000C 1260h	MTU	定时器的波形控制寄存器	TWCA	8	8	4~5 个 PCLKA	2~3 个 ICLK		381
000C 1270h	MTU (注 1)	定时器的模式寄存器 2A	TMDR2A	8	8	4~5 个 PCLKA	2~3 个 ICLK		336
000C 1272h	MTU3	定时器的通用寄存器 E	TGRE	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1274h	MTU4	定时器的通用寄存器 E	TGRE	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1276h	MTU4	定时器的通用寄存器 F	TGRF	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1280h	MTU	定时器的启动寄存器 A	TSTRA	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		363
000C 1281h	MTU	定时器的同步寄存器 A	TSYRA	8	8	4~5 个 PCLKA	2~3 个 ICLK		365
000C 1282h	MTU	计数器的同步启动寄存器	TCSYSTR	8	8	4~5 个 PCLKA	2~3 个 ICLK		367
000C 1284h	MTU	定时器的读写允许寄存器 A	TRWERA	8	8	4~5 个 PCLKA	2~3 个 ICLK		369
000C 1300h	MTU0	定时器的控制寄存器	TCR	8	8、16、32	4~5 个 PCLKA	2~3 个 ICLK		330
000C 1301h	MTU0	定时器的模式寄存器 1	TMDR1	8	8	4~5 个 PCLKA	2~3 个 ICLK		334
000C 1302h	MTU0	定时器的 I/O 控制寄存器 H	TIORH	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		336
000C 1303h	MTU0	定时器的 I/O 控制寄存器 L	TIORL	8	8	4~5 个 PCLKA	2~3 个 ICLK		337
000C 1304h	MTU0	定时器的中断允许寄存器	TIER	8	8、16、32	4~5 个 PCLKA	2~3 个 ICLK		352
000C 1305h	MTU0	定时器的状态寄存器	TSR	8	8	4~5 个 PCLKA	2~3 个 ICLK		355
000C 1306h	MTU0	定时器的计数器	TCNT	16	16	4~5 个 PCLKA	2~3 个 ICLK		361
000C 1308h	MTU0	定时器的通用寄存器 A	TGRA	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		362
000C 130Ah	MTU0	定时器的通用寄存器 B	TGRB	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 130Ch	MTU0	定时器的通用寄存器 C	TGRC	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		362
000C 130Eh	MTU0	定时器的通用寄存器 D	TGRD	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1320h	MTU0	定时器的通用寄存器 E	TGRE	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1322h	MTU0	定时器的通用寄存器 F	TGRF	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1324h	MTU0	定时器的中断允许寄存器 2	TIER2	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		352
000C 1325h	MTU0	定时器的状态寄存器 2	TSR2	8	8	4~5 个 PCLKA	2~3 个 ICLK		355
000C 1326h	MTU0	定时器的缓冲器运行传送模式寄存器	TBTM	8	8	4~5 个 PCLKA	2~3 个 ICLK		359
000C 1380h	MTU1	定时器的控制寄存器	TCR	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		330
000C 1381h	MTU1	定时器的模式寄存器 1	TMDR1	8	8	4~5 个 PCLKA	2~3 个 ICLK		334
000C 1382h	MTU1	定时器的 I/O 控制寄存器	TIOR	8	8	4~5 个 PCLKA	2~3 个 ICLK		336
000C 1384h	MTU1	定时器的中断允许寄存器	TIER	8	8、16、32	4~5 个 PCLKA	2~3 个 ICLK		352
000C 1385h	MTU1	定时器的状态寄存器	TSR	8	8	4~5 个 PCLKA	2~3 个 ICLK		355
000C 1386h	MTU1	定时器的计数器	TCNT	16	16	4~5 个 PCLKA	2~3 个 ICLK		361
000C 1388h	MTU1	定时器的通用寄存器 A	TGRA	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		362
000C 138Ah	MTU1	定时器的通用寄存器 B	TGRB	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1390h	MTU1	定时器的输入捕捉控制寄存器	TICCR	8	8	4~5 个 PCLKA	2~3 个 ICLK		360
000C 1400h	MTU2	定时器的控制寄存器	TCR	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		330
000C 1401h	MTU2	定时器的模式寄存器 1	TMDR1	8	8	4~5 个 PCLKA	2~3 个 ICLK		334
000C 1402h	MTU2	定时器的 I/O 控制寄存器	TIOR	8	8	4~5 个 PCLKA	2~3 个 ICLK		336
000C 1404h	MTU2	定时器的中断允许寄存器	TIER	8	8、16、32	4~5 个 PCLKA	2~3 个 ICLK		352
000C 1405h	MTU2	定时器的状态寄存器	TSR	8	8	4~5 个 PCLKA	2~3 个 ICLK		355
000C 1406h	MTU2	定时器的计数器	TCNT	16	16	4~5 个 PCLKA	2~3 个 ICLK		361
000C 1408h	MTU2	定时器的通用寄存器 A	TGRA	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		362
000C 140Ah	MTU2	定时器的通用寄存器 B	TGRB	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A00h	MTU6	定时器的控制寄存器	TCR	8	8、16、32	4~5 个 PCLKA	2~3 个 ICLK		330
000C 1A01h	MTU7	定时器的控制寄存器	TCR	8	8	4~5 个 PCLKA	2~3 个 ICLK		330
000C 1A02h	MTU6	定时器的模式寄存器 1	TMDR1	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		334
000C 1A03h	MTU7	定时器的模式寄存器 1	TMDR1	8	8	4~5 个 PCLKA	2~3 个 ICLK		334
000C 1A04h	MTU6	定时器的 I/O 控制寄存器 H	TIORH	8	8、16、32	4~5 个 PCLKA	2~3 个 ICLK		336
000C 1A05h	MTU6	定时器的 I/O 控制寄存器 L	TIORL	8	8	4~5 个 PCLKA	2~3 个 ICLK		337
000C 1A06h	MTU7	定时器的 I/O 控制寄存器 H	TIORH	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		336
000C 1A07h	MTU7	定时器的 I/O 控制寄存器 L	TIORL	8	8	4~5 个 PCLKA	2~3 个 ICLK		337

表 5.1 I/O 寄存器地址一览表 (14/17)

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		模块名	参照页数
						ICLK ≥ PCLK 时	ICLK < PCLK 时		
000C 1A08h	MTU6	定时器的中断允许寄存器	TIER	8	8、16	4~5 个 PCLKA	2~3 个 ICLK	MTU3	352
000C 1A09h	MTU7	定时器的中断允许寄存器	TIER	8	8	4~5 个 PCLKA	2~3 个 ICLK		352
000C 1A0Ah	MTU	定时器的输出主控允许寄存器 B	TOERB	8	8	4~5 个 PCLKA	2~3 个 ICLK		370
000C 1A0Eh	MTU	定时器的输出控制寄存器 1B	TOCR1B	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		372
000C 1A0Fh	MTU	定时器的输出控制寄存器 2B	TOCR2B	8	8	4~5 个 PCLKA	2~3 个 ICLK		374
000C 1A10h	MTU6	定时器的计数器	TCNT	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		361
000C 1A12h	MTU7	定时器的计数器	TCNT	16	16	4~5 个 PCLKA	2~3 个 ICLK		361
000C 1A14h	MTU	定时器的周期数据寄存器 B	TCDRB	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		378
000C 1A16h	MTU	定时器的死区时间数据寄存器 B	TDDRB	16	16	4~5 个 PCLKA	2~3 个 ICLK		379
000C 1A18h	MTU6	定时器的通用寄存器 A	TGRA	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A1Ah	MTU6	定时器的通用寄存器 B	TGRB	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A1Ch	MTU7	定时器的通用寄存器 A	TGRA	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A1Eh	MTU7	定时器的通用寄存器 B	TGRB	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A20h	MTU	定时器的副计数器 B	TCNTSB	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		378
000C 1A22h	MTU	定时器的周期缓冲寄存器 B	TCBRB	16	16	4~5 个 PCLKA	2~3 个 ICLK		379
000C 1A24h	MTU6	定时器的通用寄存器 C	TGRC	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A26h	MTU6	定时器的通用寄存器 D	TGRD	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A28h	MTU7	定时器的通用寄存器 C	TGRC	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A2Ah	MTU7	定时器的通用寄存器 D	TGRD	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A2Ch	MTU6	定时器的状态寄存器	TSR	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		355
000C 1A2Dh	MTU7	定时器的状态寄存器	TSR	8	8	4~5 个 PCLKA	2~3 个 ICLK		355
000C 1A30h	MTU	定时器的中断减少设定寄存器 1B	TITCR1B	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		389
000C 1A31h	MTU	定时器的中断减少次数计数器 1B	TITCNT1B	8	8	4~5 个 PCLKA	2~3 个 ICLK		391
000C 1A32h	MTU	定时器的缓冲器传送设定寄存器 B	TBTERB	8	8	4~5 个 PCLKA	2~3 个 ICLK		380
000C 1A34h	MTU	定时器的死区时间允许寄存器 B	TDERB	8	8	4~5 个 PCLKA	2~3 个 ICLK		379
000C 1A36h	MTU	定时器的输出电平缓冲寄存器 B	TOLBRB	8	8	4~5 个 PCLKA	2~3 个 ICLK		376
000C 1A38h	MTU6	定时器的缓冲运行传送模式寄存器	TBTM	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		359
000C 1A39h	MTU7	定时器的缓冲运行传送模式寄存器	TBTM	8	8	4~5 个 PCLKA	2~3 个 ICLK		359
000C 1A3Ah	MTUB	定时器的中断减少模式寄存器 B	TITMRB	8	8	4~5 个 PCLKA	2~3 个 ICLK		388
000C 1A3Bh	MTU	定时器的中断减少设定寄存器 2B	TITCR2B	8	8	4~5 个 PCLKA	2~3 个 ICLK		393
000C 1A3Ch	MTU	定时器的中断减少次数计数器 2B	TITCNT2B	8	8	4~5 个 PCLKA	2~3 个 ICLK		395
000C 1A40h	MTU7	定时器的 A/D 转换开始请求控制寄存器	TADCR	16	16	4~5 个 PCLKA	2~3 个 ICLK		383
000C 1A44h	MTU7	定时器的 A/D 转换开始请求周期设定寄存器 A	TADCORA	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		387
000C 1A46h	MTU7	定时器的 A/D 转换开始请求周期设定寄存器 B	TADCORB	16	16	4~5 个 PCLKA	2~3 个 ICLK		387
000C 1A48h	MTU7	定时器的 A/D 转换开始请求周期设定缓冲寄存器 A	TADCOBRA	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		387
000C 1A4Ah	MTU7	定时器的 A/D 转换开始请求周期设定缓冲寄存器 B	TADCOBRB	16	16	4~5 个 PCLKA	2~3 个 ICLK		387
000C 1A50h	MTU6	定时器的同步清除寄存器	TSYCR	8	8	4~5 个 PCLKA	2~3 个 ICLK		361
000C 1A60h	MTU	定时器的波形控制寄存器 B	TWCRB	8	8	4~5 个 PCLKA	2~3 个 ICLK		381
000C 1A70h	MTU	定时器的模式寄存器 2B	TMDR2B	8	8	4~5 个 PCLKA	2~3 个 ICLK		336
000C 1A72h	MTU6	定时器的通用寄存器 E	TGRE	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A74h	MTU7	定时器的通用寄存器 E	TGRE	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A76h	MTU7	定时器的通用寄存器 F	TGRF	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1A80h	MTU	定时器的启动寄存器 B	TSTRB	8	8、16	4~5 个 PCLKA	2~3 个 ICLK		363
000C 1A81h	MTU	定时器的同步寄存器 B	TSYRB	8	8	4~5 个 PCLKA	2~3 个 ICLK		365
000C 1A84h	MTU	定时器的读写允许寄存器 B	TRWERB	8	8	4~5 个 PCLKA	2~3 个 ICLK		369
000C 1C80h	MTU5	定时器的计数器 U	TCNTU	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		361
000C 1C82h	MTU5	定时器的通用寄存器 U	TGRU	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1C84h	MTU5	定时器的控制寄存器 U	TCRU	8	8	4~5 个 PCLKA	2~3 个 ICLK		330
000C 1C86h	MTU5	定时器的 I/O 控制寄存器 U	TIORU	8	8	4~5 个 PCLKA	2~3 个 ICLK		337
000C 1C90h	MTU5	定时器的计数器 V	TCNTV	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		361
000C 1C92h	MTU5	定时器的通用寄存器 V	TGRV	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1C94h	MTU5	定时器的控制寄存器 V	TCRV	8	8	4~5 个 PCLKA	2~3 个 ICLK		330
000C 1C96h	MTU5	定时器的 I/O 控制寄存器 V	TIORV	8	8	4~5 个 PCLKA	2~3 个 ICLK		337
000C 1CA0h	MTU5	定时器的计数器 W	TCNTW	16	16、32	4~5 个 PCLKA	2~3 个 ICLK		361
000C 1CA2h	MTU5	定时器的通用寄存器 W	TGRW	16	16	4~5 个 PCLKA	2~3 个 ICLK		362
000C 1CA4h	MTU5	定时器的控制寄存器 W	TCRW	8	8	4~5 个 PCLKA	2~3 个 ICLK		330
000C 1CA6h	MTU5	定时器的 I/O 控制寄存器 W	TIORW	8	8	4~5 个 PCLKA	2~3 个 ICLK		337
000C 1CB0h	MTU5	定时器的状态寄存器	TSR	8	8	4~5 个 PCLKA	2~3 个 ICLK		355
000C 1CB2h	MTU5	定时器的中断允许寄存器	TIER	8	8	4~5 个 PCLKA	2~3 个 ICLK		352
000C 1CB4h	MTU5	定时器的启动寄存器	TSTR	8	8	4~5 个 PCLKA	2~3 个 ICLK		363

表 5.1 I/O 寄存器地址一览表 (15/17)

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		模块名	参照页数
						ICLK ≥ PCLK 时	ICLK < PCLK 时		
000C 1CB6h	MTU5	定时器的比较匹配清除寄存器	TCNTCMPCLR	8	8	4 ~ 5 个 PCLKA	2 ~ 3 个 ICLK	MTU3	351
000C 2000h	GPT	通用 PWM 定时器的软件启动寄存器	GTSTR	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK	GPT	550
000C 2004h	GPT	通用 PWM 定时器的硬件源启动控制寄存器	GTHSCR	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		551
000C 2006h	GPT	通用 PWM 定时器的硬件源清除控制寄存器	GTHCCR	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		552
000C 2008h	GPT	通用 PWM 定时器的硬件启动源选择寄存器	GTHSSR	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		553
000C 200Ah	GPT	通用 PWM 定时器的硬件停止 / 清除源选择寄存器	GTHPSR	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		555
000C 200Ch	GPT	通用 PWM 定时器的写保护寄存器	GTWP	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		557
000C 200Eh	GPT	通用 PWM 定时器的同步寄存器	GTSYNC	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		558
000C 2010h	GPT	通用 PWM 定时器的外部触发输入中断寄存器	GTETINT	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		559
000C 2014h	GPT	通用 PWM 定时器的缓冲运行禁止寄存器	GTBDR	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		560
000C 2018h	GPT	通用 PWM 定时器启动写保护寄存器	GTSWP	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		561
000C 2080h	GPT	LOCO 计数控制寄存器	LCCR	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		562
000C 2082h	GPT	LOCO 计数状态寄存器	LCST	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		564
000C 2084h	GPT	LOCO 计数值寄存器	LCNT	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 2086h	GPT	LOCO 计数结果平均寄存器	LCNTA	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 2088h	GPT	LOCO 计数结果寄存器 0	LCNT00	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 208Ah	GPT	LOCO 计数结果寄存器 1	LCNT01	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 208Ch	GPT	LOCO 计数结果寄存器 2	LCNT02	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 208Eh	GPT	LOCO 计数结果寄存器 3	LCNT03	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 2090h	GPT	LOCO 计数结果寄存器 4	LCNT04	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 2092h	GPT	LOCO 计数结果寄存器 5	LCNT05	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 2094h	GPT	LOCO 计数结果寄存器 6	LCNT06	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 2096h	GPT	LOCO 计数结果寄存器 7	LCNT07	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 2098h	GPT	LOCO 计数结果寄存器 8	LCNT08	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 209Ah	GPT	LOCO 计数结果寄存器 9	LCNT09	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 209Ch	GPT	LOCO 计数结果寄存器 10	LCNT10	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 209Eh	GPT	LOCO 计数结果寄存器 11	LCNT11	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 20A0h	GPT	LOCO 计数结果寄存器 12	LCNT12	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 20A2h	GPT	LOCO 计数结果寄存器 13	LCNT13	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 20A4h	GPT	LOCO 计数结果寄存器 14	LCNT14	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 20A6h	GPT	LOCO 计数结果寄存器 15	LCNT15	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		565
000C 20A8h	GPT	LOCO 计数上限容许偏差值寄存器	LCNTDU	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		566
000C 20AAh	GPT	LOCO 计数下限容许偏差值寄存器	LCNTDL	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		566
000C 2100h	GPT0	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		566
000C 2102h	GPT0	通用 PWM 定时器的中断输出设定寄存器	GTINTAD	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		570
000C 2104h	GPT0	通用 PWM 定时器的控制寄存器	GTCR	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		572
000C 2106h	GPT0	通用 PWM 定时器的缓冲允许寄存器	GTBER	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		573
000C 2108h	GPT0	通用 PWM 定时器的计数方向寄存器	GTUDC	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		575
000C 210Ah	GPT0	通用 PWM 定时器的中断、A/D 转换开始请求减少设定寄存器	GTITC	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		576
000C 210Ch	GPT0	通用 PWM 定时器的状态寄存器	GTST	16	8、16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		578
000C 210Eh	GPT0	通用 PWM 定时器的计数器	GTGNT	16	16	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		581
000C 2110h	GPT0	通用 PWM 定时器的比较捕捉寄存器 A	GTCCRA	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 ICLK		582
000C 2112h	GPT0	通用 PWM 定时器的比较捕捉寄存器 B	GTCCRB	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 ICLK		582
000C 2114h	GPT0	通用 PWM 定时器的比较捕捉寄存器 C	GTCCRC	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		582
000C 2116h	GPT0	通用 PWM 定时器的比较捕捉寄存器 D	GTCCRD	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		582
000C 2118h	GPT0	通用 PWM 定时器的比较捕捉寄存器 E	GTCCRE	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		582
000C 211Ah	GPT0	通用 PWM 定时器的比较捕捉寄存器 F	GTCCRF	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		582
000C 211Ch	GPT0	通用 PWM 定时器的周期设定寄存器	GTPR	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		582
000C 211Eh	GPT0	通用 PWM 定时器的周期设定缓冲寄存器	GTPBR	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		582
000C 2120h	GPT0	通用 PWM 定时器的周期设定双缓冲寄存器	GTPDBR	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		583
000C 2124h	GPT0	A/D 转换开始请求时序寄存器 A	GTADTRA	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		583
000C 2126h	GPT0	A/D 转换开始请求时序缓冲寄存器 A	GTADTBRA	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		583
000C 2128h	GPT0	A/D 转换开始请求时序双缓冲寄存器 A	GTADTDBRA	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		583
000C 212Ch	GPT0	A/D 转换开始请求时序寄存器 B	GTADTRB	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		583
000C 212Eh	GPT0	A/D 转换开始请求时序缓冲寄存器 B	GTADTBRB	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		583
000C 2130h	GPT0	A/D 转换开始请求时序双缓冲寄存器 B	GTADTDBRB	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		583
000C 2134h	GPT0	通用 PWM 定时器的输出无效控制寄存器	GTONCR	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		584
000C 2136h	GPT0	通用 PWM 定时器的死区时间控制寄存器	GTDTCR	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		586
000C 2138h	GPT0	通用 PWM 定时器的死区时间值寄存器	GTDVU	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		587
000C 213Ah	GPT0	通用 PWM 定时器的死区时间值寄存器	GTDVD	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		587
000C 213Ch	GPT0	通用 PWM 定时器的死区时间缓冲寄存器	GTDBU	16	16、32	2 ~ 5 个 PCLKA	2 ~ 3 个 ICLK		587

表 5.1 I/O 寄存器地址一览表 (16/17)

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		模块名	参照页数
						ICLK ≥ PCLK 时	ICLK < PCLK 时		
000C 213Eh	GPT0	通用 PWM 定时器的死区时间缓冲寄存器	GTDBD	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	GPT	587
000C 2140h	GPT0	通用 PWM 定时器的输出保护功能状态寄存器	GTSOS	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		588
000C 2142h	GPT0	通用 PWM 定时器的输出保护功能暂时解除寄存器	GTSOTR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		588
000C 2180h	GPT1	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		566
000C 2182h	GPT1	通用 PWM 定时器的中断输出设定寄存器	GTINTAD	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		570
000C 2184h	GPT1	通用 PWM 定时器的控制寄存器	GTCR	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		572
000C 2186h	GPT1	通用 PWM 定时器的缓冲允许寄存器	GTBER	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		573
000C 2188h	GPT1	通用 PWM 定时器的计数方向寄存器	GTUDC	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		575
000C 218Ah	GPT1	通用 PWM 定时器的中断、A/D 转换开始请求减少设定寄存器	GTITC	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		576
000C 218Ch	GPT1	通用 PWM 定时器的状态寄存器	GTST	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		578
000C 218Eh	GPT1	通用 PWM 定时器的计数器	GT CNT	16	16	2~5 个 PCLKA	2~3 个 ICLK		581
000C 2190h	GPT1	通用 PWM 定时器的比较捕捉寄存器 A	GTCCRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2192h	GPT1	通用 PWM 定时器的比较捕捉寄存器 B	GTCCRB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2194h	GPT1	通用 PWM 定时器的比较捕捉寄存器 C	GTCCRC	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2196h	GPT1	通用 PWM 定时器的比较捕捉寄存器 D	GTCCRD	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2198h	GPT1	通用 PWM 定时器的比较捕捉寄存器 E	GTCCRE	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 219Ah	GPT1	通用 PWM 定时器的比较捕捉寄存器 F	GTCCRF	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 219Ch	GPT1	通用 PWM 定时器的周期设定寄存器	GTPR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 219Eh	GPT1	通用 PWM 定时器的周期设定缓冲寄存器	GTPBR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 21A0h	GPT1	通用 PWM 定时器的周期设定双缓冲寄存器	GTPDBR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 21A4h	GPT1	A/D 转换开始请求时序寄存器 A	GTADTRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 21A6h	GPT1	A/D 转换开始请求时序缓冲寄存器 A	GTADTBRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 21A8h	GPT1	A/D 转换开始请求时序双缓冲寄存器 A	GTADTDBRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 21ACh	GPT1	A/D 转换开始请求时序寄存器 B	GTADTRB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 21AEh	GPT1	A/D 转换开始请求时序缓冲寄存器 B	GTADTB RB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 21B0h	GPT1	A/D 转换开始请求时序双缓冲寄存器 B	GTADTDB RB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 21B4h	GPT1	通用 PWM 定时器的输出无效控制寄存器	GT ONCR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		584
000C 21B6h	GPT1	通用 PWM 定时器的死区时间控制寄存器	GT DTCR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		586
000C 21B8h	GPT1	通用 PWM 定时器的死区时间值寄存器	GT DVU	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		587
000C 21BAh	GPT1	通用 PWM 定时器的死区时间值寄存器	GT DVU	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		587
000C 21BCh	GPT1	通用 PWM 定时器的死区时间缓冲寄存器	GT DBU	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		587
000C 21BEh	GPT1	通用 PWM 定时器的死区时间缓冲寄存器	GT DBD	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		587
000C 21C0h	GPT1	通用 PWM 定时器的输出保护功能状态寄存器	GTSOS	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		588
000C 21C2h	GPT1	通用 PWM 定时器的输出保护功能暂时解除寄存器	GTSOTR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		588
000C 2200h	GPT2	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		566
000C 2202h	GPT2	通用 PWM 定时器的中断输出设定寄存器	GTINTAD	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		570
000C 2204h	GPT2	通用 PWM 定时器的控制寄存器	GTCR	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		572
000C 2206h	GPT2	通用 PWM 定时器的缓冲允许寄存器	GTBER	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		573
000C 2208h	GPT2	通用 PWM 定时器的计数方向寄存器	GTUDC	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		575
000C 220Ah	GPT2	通用 PWM 定时器的中断、A/D 转换开始请求减少设定寄存器	GTITC	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		576
000C 220Ch	GPT2	通用 PWM 定时器的状态寄存器	GTST	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		578
000C 220Eh	GPT2	通用 PWM 定时器的计数器	GT CNT	16	16	2~5 个 PCLKA	2~3 个 ICLK		581
000C 2210h	GPT2	通用 PWM 定时器的比较捕捉寄存器 A	GTCCRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2212h	GPT2	通用 PWM 定时器的比较捕捉寄存器 B	GTCCRB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2214h	GPT2	通用 PWM 定时器的比较捕捉寄存器 C	GTCCRC	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2216h	GPT2	通用 PWM 定时器的比较捕捉寄存器 D	GTCCRD	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2218h	GPT2	通用 PWM 定时器的比较捕捉寄存器 E	GTCCRE	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 221Ah	GPT2	通用 PWM 定时器的比较捕捉寄存器 F	GTCCRF	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 221Ch	GPT2	通用 PWM 定时器的周期设定寄存器	GTPR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 221Eh	GPT2	通用 PWM 定时器的周期设定缓冲寄存器	GTPBR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2220h	GPT2	通用 PWM 定时器的周期设定双缓冲寄存器	GTPDBR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 2224h	GPT2	A/D 转换开始请求时序寄存器 A	GTADTRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 2226h	GPT2	A/D 转换开始请求时序缓冲寄存器 A	GTADTBRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 2228h	GPT2	A/D 转换开始请求时序双缓冲寄存器 A	GTADTDBRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 222Ch	GPT2	A/D 转换开始请求时序寄存器 B	GTADTRB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 222Eh	GPT2	A/D 转换开始请求时序缓冲寄存器 B	GTADTB RB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 2230h	GPT2	A/D 转换开始请求时序双缓冲寄存器 B	GTADTDB RB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		583
000C 2234h	GPT2	通用 PWM 定时器的输出无效控制寄存器	GT ONCR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		584
000C 2236h	GPT2	通用 PWM 定时器的死区时间控制寄存器	GT DTCR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		586
000C 2238h	GPT2	通用 PWM 定时器的死区时间值寄存器	GT DVU	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		587

表 5.1 I/O 寄存器地址一览表 (17/17)

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		模块名	参照页数
						ICLK ≥ PCLK 时	ICLK < PCLK 时		
000C 223Ah	GPT2	通用 PWM 定时器的死区时间值寄存器	GTDVD	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	GPT	587
000C 223Ch	GPT2	通用 PWM 定时器的死区时间缓冲寄存器	GTDBU	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		587
000C 223Eh	GPT2	通用 PWM 定时器的死区时间缓冲寄存器	GTDBD	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		587
000C 2240h	GPT2	通用 PWM 定时器的输出保护功能状态寄存器	GTSOS	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		588
000C 2242h	GPT2	通用 PWM 定时器的输出保护功能暂时解除寄存器	GTSOTR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		588
000C 2280h	GPT3	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		566
000C 2282h	GPT3	通用 PWM 定时器的中断输出设定寄存器	GTINTAD	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		570
000C 2284h	GPT3	通用 PWM 定时器的控制寄存器	GTCR	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		572
000C 2286h	GPT3	通用 PWM 定时器的缓冲允许寄存器	GTBER	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		573
000C 2288h	GPT3	通用 PWM 定时器的计数方向寄存器	GTUDC	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		575
000C 228Ah	GPT3	通用 PWM 定时器的中断、A/D 转换开始请求减少设定寄存器	GTITC	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		576
000C 228Ch	GPT3	通用 PWM 定时器的状态寄存器	GTST	16	8、16、32	2~5 个 PCLKA	2~3 个 ICLK		578
000C 228Eh	GPT3	通用 PWM 定时器的计数器	GTCNT	16	16	2~5 个 PCLKA	2~3 个 ICLK		581
000C 2290h	GPT3	通用 PWM 定时器的比较捕捉寄存器 A	GTCCRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2292h	GPT3	通用 PWM 定时器的比较捕捉寄存器 B	GTCCRB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2294h	GPT3	通用 PWM 定时器的比较捕捉寄存器 C	GTCCRC	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2296h	GPT3	通用 PWM 定时器的比较捕捉寄存器 D	GTCCRD	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 2298h	GPT3	通用 PWM 定时器的比较捕捉寄存器 E	GTCCRE	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 229Ah	GPT3	通用 PWM 定时器的比较捕捉寄存器 F	GTCCRF	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 229Ch	GPT3	通用 PWM 定时器的周期设定寄存器	GTPR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK		582
000C 229Eh	GPT3	通用 PWM 定时器的周期设定缓冲寄存器	GTPBR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	582	
000C 22A0h	GPT3	通用 PWM 定时器的周期设定双缓冲寄存器	GTPDBR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	583	
000C 22A4h	GPT3	A/D 转换开始请求时序寄存器 A	GTADTRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	583	
000C 22A6h	GPT3	A/D 转换开始请求时序缓冲寄存器 A	GTADTBRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	583	
000C 22A8h	GPT3	A/D 转换开始请求时序双缓冲寄存器 A	GTADTBRA	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	583	
000C 22ACh	GPT3	A/D 转换开始请求时序寄存器 B	GTADTRB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	583	
000C 22AEh	GPT3	A/D 转换开始请求时序缓冲寄存器 B	GTADTB RB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	583	
000C 22B0h	GPT3	A/D 转换开始请求时序双缓冲寄存器 B	GTADTB RB	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	583	
000C 22B4h	GPT3	通用 PWM 定时器的输出无效控制寄存器	GTONCR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	584	
000C 22B6h	GPT3	通用 PWM 定时器的死区时间控制寄存器	GTDTCR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	586	
000C 22B8h	GPT3	通用 PWM 定时器的死区时间值寄存器	GTDVU	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	587	
000C 22BAh	GPT3	通用 PWM 定时器的死区时间值寄存器	GTDVD	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	587	
000C 22BCh	GPT3	通用 PWM 定时器的死区时间缓冲寄存器	GTDBU	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	587	
000C 22BEh	GPT3	通用 PWM 定时器的死区时间缓冲寄存器	GTDBD	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	587	
000C 22C0h	GPT3	通用 PWM 定时器的输出保护功能状态寄存器	GTSOS	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	588	
000C 22C2h	GPT3	通用 PWM 定时器的输出保护功能暂时解除寄存器	GTSOTR	16	16、32	2~5 个 PCLKA	2~3 个 ICLK	588	
007F C402h	FLASH	闪存模式寄存器	FMODR	8	8	2~3 个 FCLK	2~3 个 ICLK	ROM/ E2 数据闪存	1042、 1112
007F C410h	FLASH	闪存存取状态寄存器	FASTAT	8	8	2~3 个 FCLK	2~3 个 ICLK		1043、 1113
007F C411h	FLASH	闪存存取错误中断允许寄存器	FAEINT	8	8	2~3 个 FCLK	2~3 个 ICLK		1045、 1115
007F C412h	FLASH	闪存就绪中断允许寄存器	FRDYIE	8	8	2~3 个 FCLK	2~3 个 ICLK	ROM	1049
007F C440h	FLASH	E2 数据闪存读允许寄存器 0	DFLRE0	16	16	2~3 个 FCLK	2~3 个 ICLK	E2 数据闪存	1116
007F C450h	FLASH	E2 数据闪存编程 / 擦除允许寄存器 0	DFLWE0	16	16	2~3 个 FCLK	2~3 个 ICLK		1117
007F FFB0h	FLASH	闪存状态寄存器 0	FSTATR0	8	8	2~3 个 FCLK	2~3 个 ICLK	ROM	1046
007F FFB1h	FLASH	闪存状态寄存器 1	FSTATR1	8	8	2~3 个 FCLK	2~3 个 ICLK		1048
007F FFB2h	FLASH	闪存 P/E 模式入口寄存器	FENTRYR	16	16	2~3 个 FCLK	2~3 个 ICLK	E2 数据闪存	1050、 1118
007F FFB4h	FLASH	闪存保护寄存器	FPROTR	16	16	2~3 个 FCLK	2~3 个 ICLK	ROM	1051
007F FFB6h	FLASH	闪存复位寄存器	FRESETR	16	16	2~3 个 FCLK	2~3 个 ICLK		1052
007F FFBAh	FLASH	FCU 命令寄存器	FCMDR	16	16	2~3 个 FCLK	2~3 个 ICLK		1053
007F FFC8h	FLASH	FCU 处理切换寄存器	FCPSR	16	16	2~3 个 FCLK	2~3 个 ICLK	1054	
007F FFCAh	FLASH	数据闪存空白检查控制寄存器	DFLBCNT	16	16	2~3 个 FCLK	2~3 个 ICLK	E2 数据闪存	1119
007F FFCCh	FLASH	闪存 P/E 状态寄存器	FPSTAT	16	16	2~3 个 FCLK	2~3 个 ICLK		ROM
007F FFCEh	FLASH	数据闪存空白检查控制状态寄存器	DFLBCSTAT	16	16	2~3 个 FCLK	2~3 个 ICLK	E2 数据闪存	1119
007F FFE8h	FLASH	外围时钟通知寄存器	PCKAR	16	16	2~3 个 FCLK	2~3 个 ICLK	ROM	1055

注 1. MTUA 表示 MTU3 和 MTU4 的通用寄存器。

注 2. MTU 表示 MTU0 ~ MTU7 的通用寄存器。

注 3. MTUB 表示 MTU6 和 MTU7 的通用寄存器

6. 复位

6.1 概要

复位有引脚复位、上电复位、电压监视 0 复位、电压监视 1 复位、电压监视 2 复位、深度软件待机复位、独立看门狗定时器复位、看门狗定时器复位和软件复位。

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。
上电复位	VCC 上升（监视电压：VPOR）（注 1）。
电压监视 0 复位	VCC 下降（监视电压：Vdet0）（注 1）。
电压监视 1 复位	VCC 下降（监视电压：Vdet1）（注 1）。
电压监视 2 复位	VCC 下降（监视电压：Vdet2）（注 1）。
深度软件待机复位	通过中断解除深度软件待机模式。
独立看门狗定时器复位	独立看门狗定时器发生下溢或者刷新错误。
看门狗定时器复位	看门狗定时器发生下溢或者刷新错误。
软件复位	设定寄存器。

注 1. 有关监视电压（VPOR、Vdet0、Vdet1、Vdet2），请参照“8. 电压检测电路（LVDA）”和“36. 电特性”。

通过复位对内部状态进行初始化，引脚为初始状态。
各种复位的初始化对象如表 6.2 所示。

表 6.2 各种复位的初始化对象

复位对象	复位源								
	引脚复位	上电复位	电压 监视 0 复位	独立看门 狗定时器 复位	看门狗 定时器 复位	电压 监视 1 复位	电压 监视 2 复位	深度软 件待机 复位	软件复位
上电复位检测标志 (RSTSR0.PORF)	○	—	—	—	—	—	—	—	—
冷启动 / 热启动的判断标志 (RSTSR1.CWSF)	—	○	—	—	—	—	—	—	—
电压监视 0 复位检测标志 (RSTSR0.LVD0RF)	○	○	—	—	—	—	—	—	—
独立看门狗定时器复位检测标志 (RSTSR2.IWDTRF)	○	○	○	—	—	—	—	○	—
独立看门狗定时器的寄存器 (IWDTRR、IWDTCR、IWDTSR、 IWDTRCR、IWDTCSTPR、 ILOCOCR)	○	○	○	—	—	—	—	○	—
看门狗定时器复位检测标志 (RSTSR2.WDTRF)	○	○	○	○	—	—	—	○	—
看门狗定时器的寄存器 (WDTRR、WDTCSR、WDTSR、 WDTRCR)	○	○	○	○	—	—	—	○	—
电压监视 1 复位检测标志 (RSTSR0.LVD1RF)	○	○	○	○	○	—	—	—	—
电压监视功能 1 的寄存器 (LVD1CR0、LVCMPCR.LVD1E、 LVDLVL.R.LVD1LVL)	○	○	○	○	○	—	—	—	—
(LVD1CR1、LVD1SR)	○	○	○	○	○	—	—	○	—
电压监视功能 2 的寄存器 (RSTSR0.LVD2RF、LVD2CR0、 LVCMPCR.LVD2E、 LVDLVL.R.LVD2LVL)	○	○	○	○	○	○	—	—	—
(LVD2CR1、LVD2SR)	○	○	○	○	○	○	—	○	—
深度软件待机复位检测标志 (RSTSR0.DPSRSTF)	○	○	○	○	○	○	○	—	—
软件复位检测标志 (RSTSR2.SWRF)	○	○	○	○	○	○	○	○	—
主时钟振荡器的相关寄存器 (MOFCR)	○	○	○	○	○	○	○	—	○
引脚状态	○	○	○	○	○	○	○	—	○
低功耗功能的相关寄存器 (DPSBYCR、DPSIER0-3、 DPSIFR0-3、DPSIEGR0-3)	○	○	○	○	○	○	○	—	○
上述以外的寄存器、CPU 和内部状态	○	○	○	○	○	○	○	○	○

○：初始化。—：不变

如果解除复位，就开始进行复位异常处理。有关复位异常处理，请参照“13. 异常处理”。
与复位有关的输入 / 输出引脚如表 6.3 所示。

表 6.3 与复位有关的输入 / 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

6.2.1 复位状态寄存器 0 (RSTSR0)

地址 0008 C290h

b7	b6	b5	b4	b3	b2	b1	b0
DPSRS TF	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF

复位后的值 0(注1) 0 0 0 0(注1) 0(注1) 0(注1) 0(注1)

注 1. 复位值因复位源而不同。

位	符号	位名	功能	R/W
b0	PORF	上电复位检测标志	0: 未检测到上电复位 1: 检测到上电复位	R(W) (注1)
b1	LVD0RF	电压监视 0 复位检测标志	0: 未检测到电压监视 0 复位 1: 检测到电压监视 0 复位	R(W) (注1)
b2	LVD1RF	电压监视 1 复位检测标志	0: 未检测到电压监视 1 复位 1: 检测到电压监视 1 复位	R(W) (注1)
b3	LVD2RF	电压监视 2 复位检测标志	0: 未检测到电压监视 2 复位 1: 检测到电压监视 2 复位	R(W) (注1)
b6-b4	—	保留位	读写值都为“0”。	R/W
b7	DPSRSTF	深度软件待机复位标志	0: 未发生由中断引起的深度软件待机模式的解除请求 1: 发生由中断引起的深度软件待机模式的解除请求	R(W) (注1)

注 1. 为了清除标志，只能写“0”。

PORF 标志（上电复位检测标志）

表示发生上电复位。

[为“1”的条件]

- 当发生上电复位时

[为“0”的条件]

- 当进行表6.2所示的复位时
- 在读“1”后写“0”时

LVD0RF 标志（电压监视 0 复位检测标志）

表示检测到 VCC 电压低于等于 Vdet0 电平。

[为“1”的条件]

- 当检测到 Vdet0 电平的 VCC 电压时

[为“0”的条件]

- 当进行表6.2所示的复位时
- 在读“1”后写“0”时

LVD1RF 标志（电压监视 1 复位检测标志）

表示检测到 VCC 电压低于等于 Vdet1 电平。

[为“1”的条件]

- 当检测到 Vdet1 电平的 VCC 电压时

[为“0”的条件]

- 当进行表 6.2 所示的复位时
- 在读“1”后写“0”时

LVD2RF 标志（电压监视 2 复位检测标志）

表示检测到 VCC 电压低于等于 Vdet2 电平。

[为“1”的条件]

- 当检测到 Vdet2 电平的 VCC 电压时

[为“0”的条件]

- 当进行表 6.2 所示的复位时
- 在读“1”后写“0”时

DPSRSTF 标志（深度软件待机复位标志）

表示在通过中断解除深度软件待机模式的同时发生内部复位（深度软件待机复位）。

[为“1”的条件]

- 当通过中断解除了深度软件待机模式时
详细内容请参照“11. 低功耗功能”。

[为“0”的条件]

- 当进行表 6.2 所示的复位时
- 在读“1”后写“0”时

6.2.2 复位状态寄存器 1 (RSTSR1)

地址 0008 C291h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CWSF
复位后的值	0	0	0	0	0	0	0	0/1 (注1)

注 1. 复位后的值因复位源而不同。

位	符号	位名	功能	R/W
b0	CWSF	冷启动 / 热启动的判断标志	0: 冷启动 1: 热启动	R(W) (注1)
b7-b1	—	保留位	读写值都为“0”。	R/W

注 1. 为了将标志置位，只能写“1”。

RSTSR1 寄存器判断是接通电源时的复位处理（冷启动）还是运行过程中输入复位信号时的复位处理（热启动）。

CWSF 标志（冷启动 / 热启动的判断标志）

表示冷启动或者热启动。

通过上电复位而不通过 RES# 引脚的复位信号对 CWSF 标志进行初始化。

[为“1”的条件]

- 当通过程序写“1”时。即使写“0”，值也不变。

[为“0”的条件]

- 当进行表 6.2 所示的复位时

6.2.3 复位状态寄存器 2 (RSTSR2)

地址 0008 00C0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SWRF	WDTRF	WDTRF
复位后的值	0	0	0	0	0	0(注1)	0(注1)	0(注1)

注 1. 复位后的值因复位源而不同。

位	符号	位名	功能	R/W
b0	IWDTRF	独立看门狗定时器复位检测标志	0: 未检测到独立看门狗定时器复位 1: 检测到独立看门狗定时器复位	R(W) (注 1)
b1	WDTRF	看门狗定时器复位检测标志	0: 未检测到看门狗定时器复位 1: 检测到看门狗定时器复位	R(W) (注 1)
b2	SWRF	软件复位检测标志	0: 未检测到软件复位 1: 检测到软件复位	R(W) (注 1)
b7-b3	—	保留位	读写值都为“0”。	R/W

注 1. 为了清除标志，只能写“0”。

IWDTRF 标志 (独立看门狗定时器复位检测标志)

表示发生独立看门狗定时器复位。

[为“1”的条件]

- 当发生独立看门狗定时器复位时

[为“0”的条件]

- 当进行表6.2所示的复位时
- 在读“1”后写“0”时

WDTRF 标志 (看门狗定时器复位检测标志)

表示发生看门狗定时器复位。

[为“1”的条件]

- 当发生看门狗定时器复位时

[为“0”的条件]

- 当进行表6.2所示的复位时
- 在读“1”后写“0”时

SWRF 标志 (软件复位检测标志)

表示发生软件复位。

[为“1”的条件]

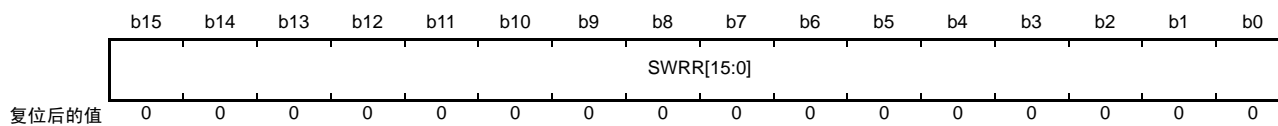
- 当发生软件复位时

[为“0”的条件]

- 当进行表6.2所示的复位时
- 在读“1”后写“0”时

6.2.4 软件复位寄存器 (SWRR)

地址 0008 00C2h



位	符号	位名	功能	R/W
b15-b0	SWRR[15:0]	软件复位的位	如果写“A501h”，就对 LSI 进行复位。读取值为“0000h”。	R/W

6.3 运行说明

6.3.1 引脚复位

这是 RES# 引脚引起的复位。

如果 RES# 引脚变为 Low 电平，就停止正在执行的全部处理，进入复位状态。

为了确保通过 RES# 引脚对该 LSI 进行复位，必须在接通电源时根据规定的电源稳定时间，使 RES# 引脚保持 Low 电平；在运行时根据规定的复位脉宽，使 RES# 引脚保持 Low 电平。

在将 RES# 引脚从 Low 电平置为 High 电平后，经过 RES# 解除后待机时间 (tRESWT)，解除内部复位，并且 CPU 开始进行复位异常处理。

详细内容请参照“36. 电特性”。

6.3.2 上电复位和电压监视 0 复位

上电复位是上电复位电路引起的内部复位。

如果在将 RES# 引脚置为 High 电平的状态下接通电源，就产生上电复位。如果 VCC 高于 VPOR，就在经过一定的时间（上电复位时间）后解除内部复位，CPU 开始进行复位异常处理。上电复位时间是使外部电源和 LSI 稳定的时间。

如果发生上电复位，RSTSR0.PORF 标志就变为“1”。通过引脚复位对 PORF 标志进行初始化。

电压监视 0 复位是电压监视电路引起的内部复位。

在选项功能选择寄存器 1 (OFS1) 的电压检测 0 电平选择位 (LVDAS) 为“0”（复位后，电压监视 0 复位有效）的状态下，如果 VCC 低于等于 Vdet0，RSTSR0.LVD0RF 标志就变为“1”，并且电压检测电路产生电压监视 0 复位。在使用电压监视 0 复位时，必须将 OFS1.LVDAS 位置“0”。如果 VCC 高于 Vdet0，就在经过 LVD0 复位时间 (tLVD0) 后解除内部复位，CPU 开始进行复位异常处理。

上电复位和电压监视 0 复位的运行例子如图 6.1 所示。

有关电压监视 0 复位的详细内容，请参照“8. 电压检测电路 (LVDA)”。

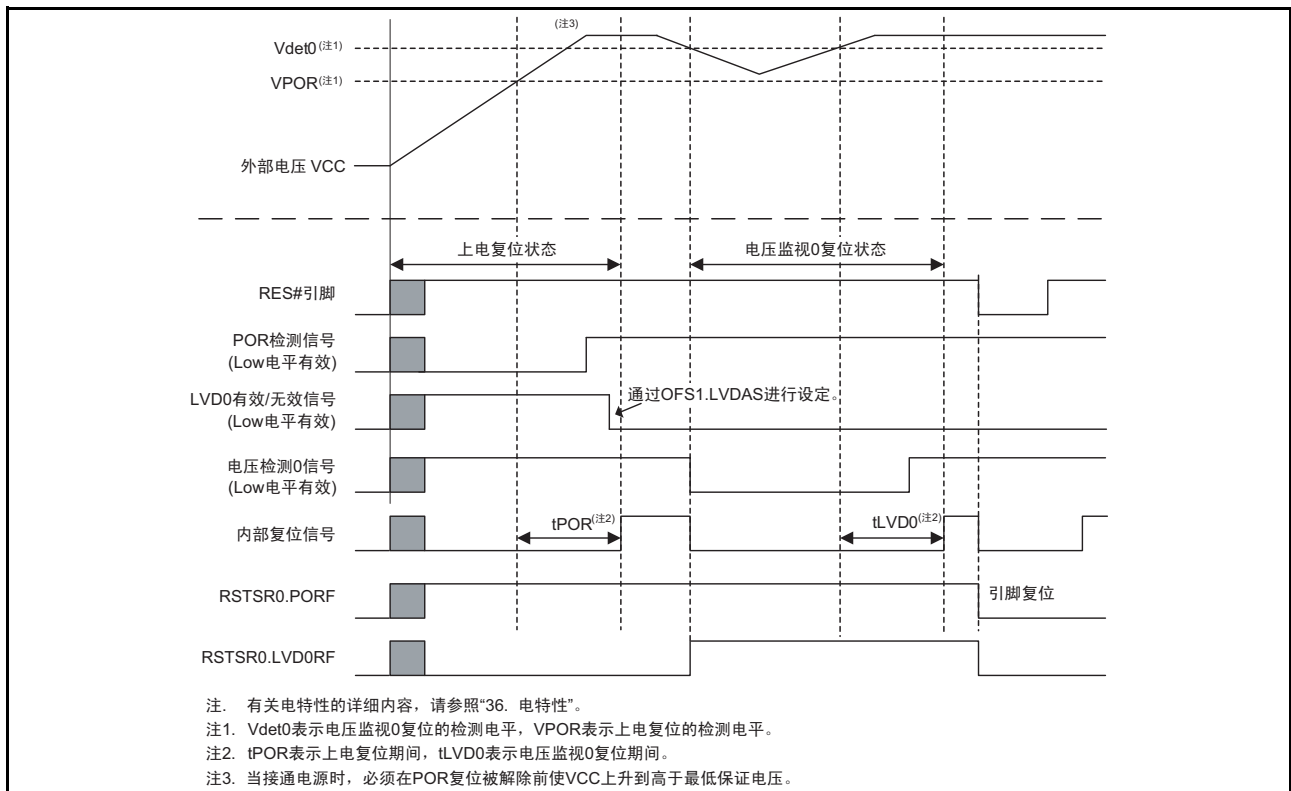


图 6.1 上电复位和电压监视 0 复位的运行例子

6.3.3 电压监视 1 复位和电压监视 2 复位

这是电压监视电路引起的内部复位。

在电压监视 1 电路的控制寄存器 0 (LVD1CR0) 的电压监视 1 中断 / 复位允许位 (LVD1RIE) 为“1” (电压检测电路引起的复位或者中断有效) 并且电压监视 1 电路模式选择位 (LVD1RI) 为“1” (在检测到低电压时产生复位) 的状态下, 如果 VCC 低于等于 V_{det1} , RSTSR0.LVD1RF 标志就变为“1”, 并且电压检测电路产生电压监视 1 复位。

同样, 在电压监视 2 的电路控制寄存器 0 (LVD2CR0) 的电压监视 2 中断 / 复位允许位 (LVD2RIE) 为“1” (电压检测电路引起的复位或者中断有效) 并且电压监视 2 电路模式选择位 (LVD2RI) 为“1” (在检测到低电压时产生复位) 的状态下, 如果 VCC 低于等于 V_{det2} , RSTSR0.LVD2RF 标志就变为“1”, 并且电压检测电路产生电压监视 2 复位。

能通过 LVD1CR0 寄存器的电压监视 1 复位无效选择位 (LVD1RN) 选择电压监视 1 复位的解除时序。当 LVD1CR0.LVD1RN 位为“0”时, VCC 低于等于 V_{det1} 。此后, 如果 VCC 高于 V_{det1} 并且经过 LVD1 复位时间 (t_{LVD1}), 就解除内部复位, CPU 开始进行复位异常处理。当 LVD1CR0.LVD1RN 位为“1”时, 在 VCC 低于等于 V_{det1} 并且经过 LVD1 复位时间 (t_{LVD1}) 后解除内部复位, CPU 开始进行复位异常处理。

同样, 能通过设定 LVD2CR0 寄存器的电压监视 2 复位无效选择位 (LVD2RN), 选择电压监视 2 复位的解除时序。

能通过设定电压检测电平选择寄存器 (LVDLVLR), 更改 V_{det1} 和 V_{det2} 的电压检测电平。

电压监视 1 复位和电压监视 2 复位的运行例子如图 6.2 所示。

有关电压监视 1 复位和电压监视 2 复位的详细内容, 请参照“8. 电压检测电路 (LVDA)”。

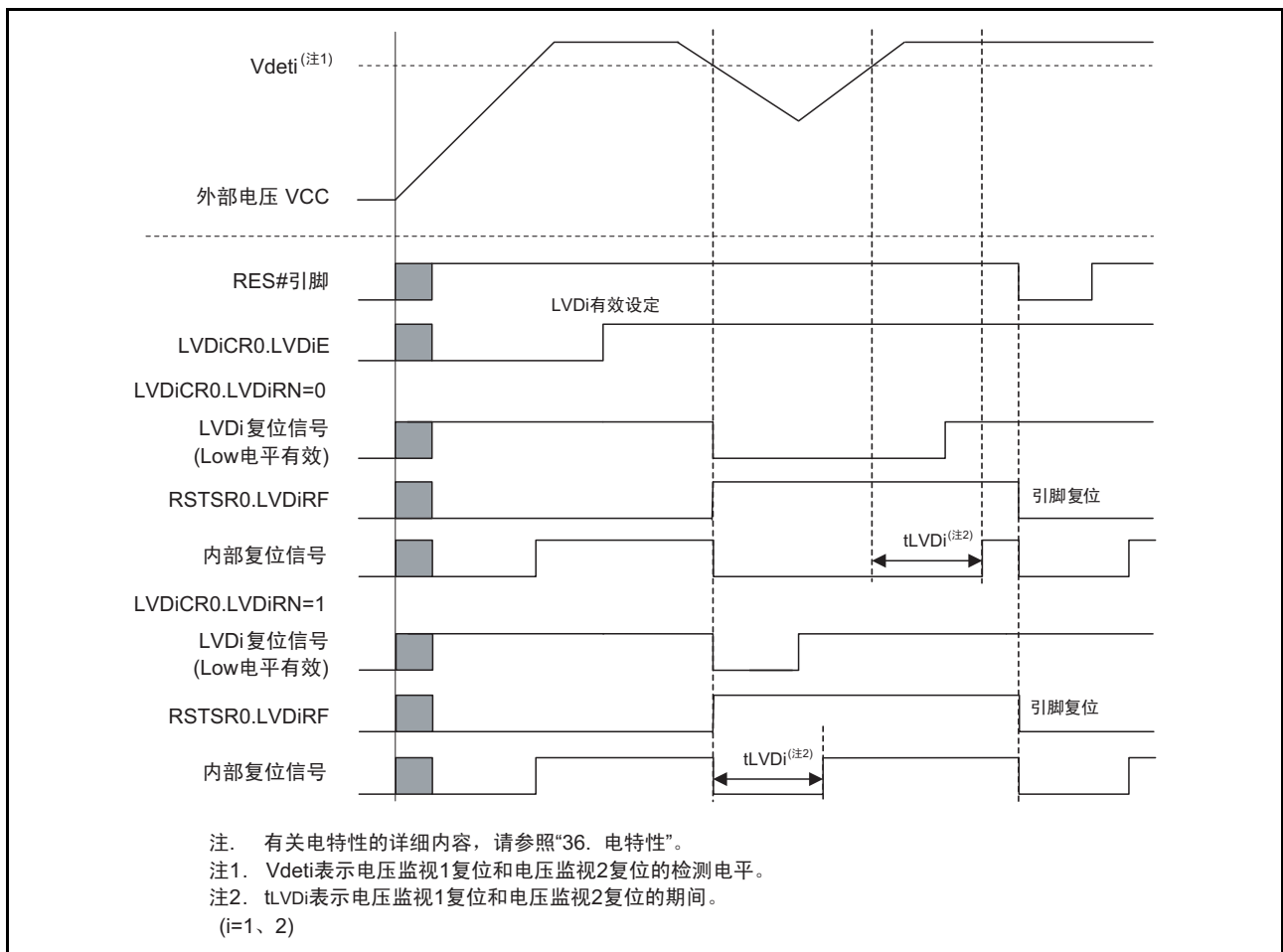


图 6.2 电压监视 1 复位和电压监视 2 复位的运行例子

6.3.4 深度软件待机复位

这是在通过中断解除深度软件待机模式时产生的内部复位。

如果发生深度软件待机模式的解除源，就产生深度软件待机复位。在解除深度软件待机模式并且经过返回时间（ t_{DSBY} ）后，在解除深度软件待机复位的同时也解除深度软件待机模式。

在解除深度软件待机模式并且经过待机时间（ t_{DSBYWT} ）后解除内部复位，CPU 开始进行复位异常处理。有关深度软件待机复位的详细内容，请参照“11. 低功耗功能”。

6.3.5 独立看门狗定时器复位

这是独立看门狗定时器引起的内部复位。

能通过设定 IWDTRCR 复位控制寄存器（IWDTRCR）或者选项功能选择寄存器 0（OFS0），选择是否从独立看门狗定时器输出独立看门狗定时器复位。

如果选择输出独立看门狗定时器复位，就在独立看门狗定时器发生下溢或者不在刷新允许期间进行写操作时产生独立看门狗定时器复位。在产生独立看门狗定时器复位并且经过内部复位时间（ t_{RESW2} ）后解除内部复位，CPU 开始进行复位异常处理。

有关独立看门狗定时器复位的详细内容，请参照“26. 独立看门狗定时器（IWDtA）”。

6.3.6 看门狗定时器复位

这是看门狗定时器引起的内部复位。

能通过设定 WDT 复位控制寄存器（WDTRCR）或者选项功能选择寄存器 0（OFS0），选择是否从看门狗定时器输出看门狗定时器复位。

如果选择输出看门狗定时器复位，就在看门狗定时器发生下溢或者不在刷新允许期间进行写操作时产生看门狗定时器复位。在产生看门狗定时器复位并且经过内部复位时间（ t_{RESW2} ）后解除内部复位，CPU 开始进行复位异常处理。

有关看门狗定时器复位的详细内容，请参照“25. 看门狗定时器（WDtA）”。

6.3.7 软件复位

这是软件复位电路引起的内部复位。

如果给 SWRR 寄存器写“A501h”，就产生软件复位。在产生软件复位并且经过内部复位时间（ t_{RESW2} ）后解除内部复位，CPU 开始进行复位异常处理。

6.3.8 冷启动 / 热启动判断功能

能通过 RSTSR1.CWSF 标志判断是接通电源时的复位处理（冷启动）还是运行过程中输入复位信号时的复位处理（热启动）。

如果发生上电复位，RSTSR1.CWSF 标志就变为“0”（冷启动）；如果进行其他复位，此标志就不变为“0”。如果通过程序给 RSTSR1.CWSF 标志写“1”，此标志就变为“1”；如果写“0”，值就不变。

冷启动 / 热启动判断功能的运行例子如图 6.3 所示。

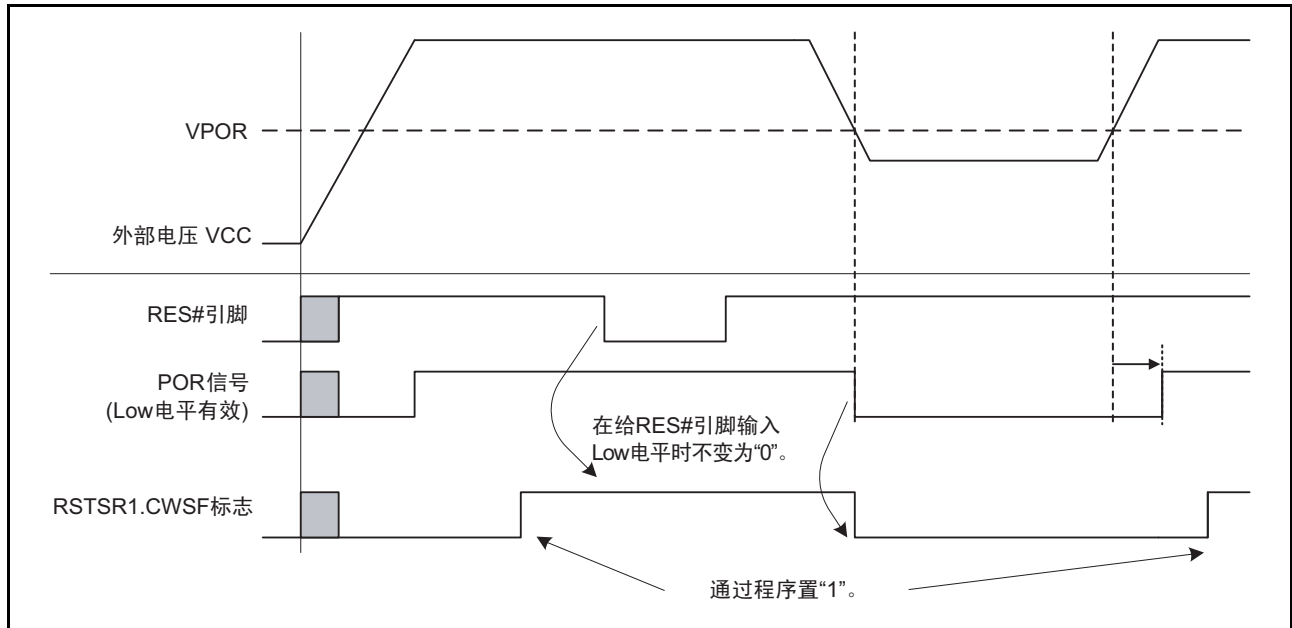


图 6.3 冷启动 / 热启动判断功能的运行例子

6.3.9 复位发生源的判断

能通过读 RSTSR0 寄存器和 RSTSR2 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 6.4。

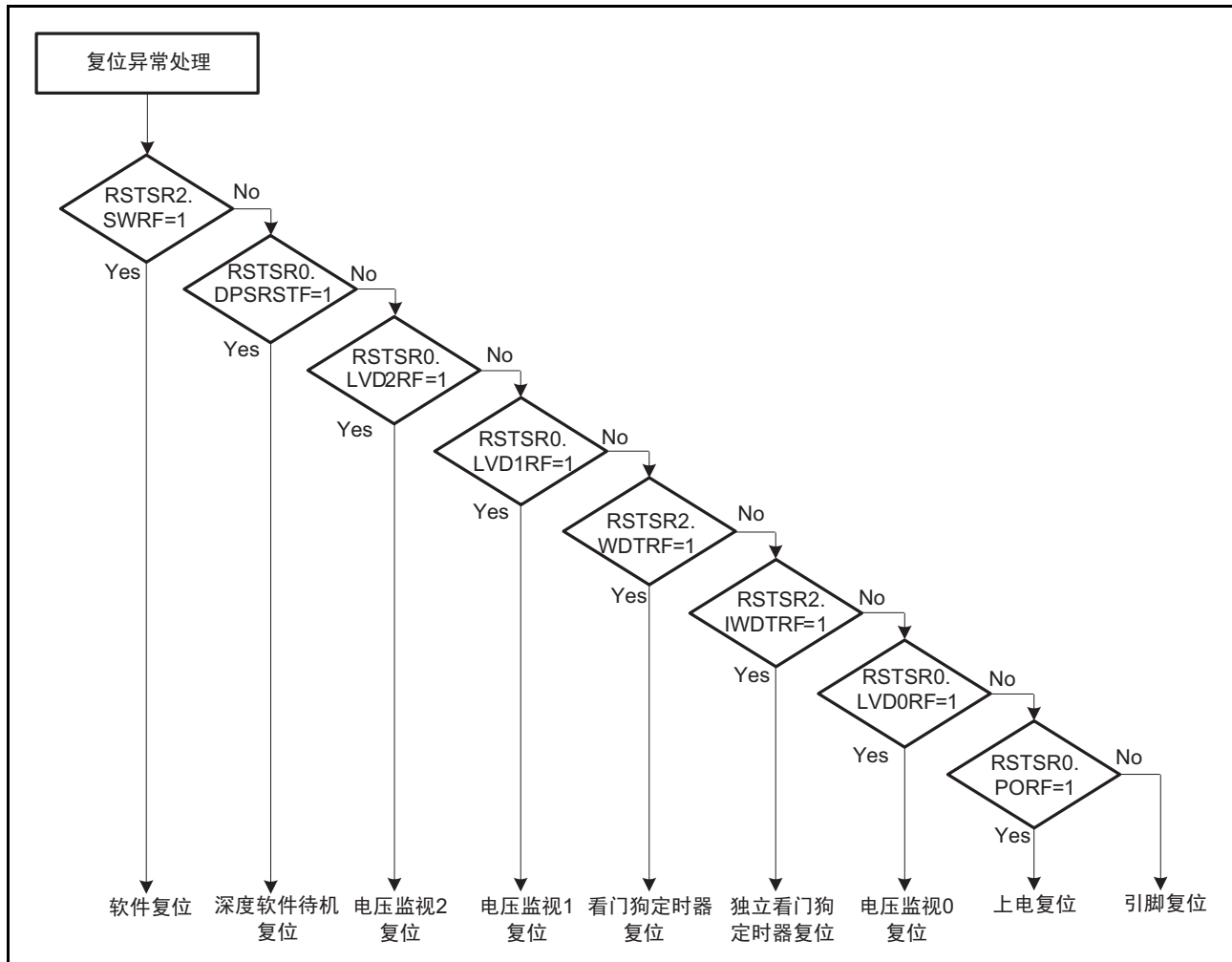


图 6.4 复位发生源的判断流程例子

7. 选项设定存储器

7.1 概要

选项设定存储器在内部 ROM 内，有选择复位后的单片机状态的寄存器。

选项设定存储区如图 7.1 所示。

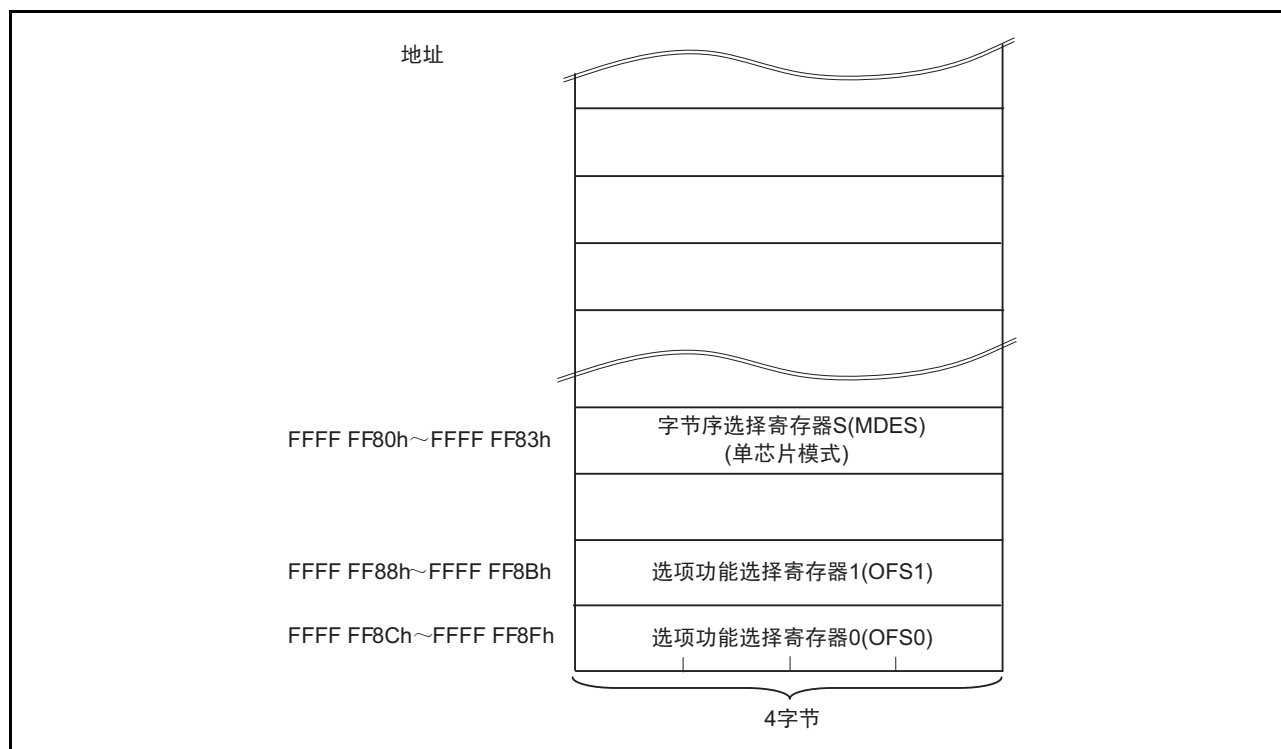


图 7.1 选项设定存储区

7.2 寄存器说明

7.2.1 选项功能选择寄存器 0 (OFS0)

地址 FFFF FF8Ch

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	WDTRS TIRQS	WDTRPSS[1:0]	WDTRPES[1:0]	WDTCKS[3:0]			WDTTOPS[1:0]	WDTST RT	—				
复位后的值 用户的设定值 (注1)															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	IWDTSL CSTP	—	IWDRS TIRQS	IWDRPSS[1:0]	IWDRPES[1:0]	IWDTCKS[3:0]			IWDTTOPS[1:0]	IWDTST RT	—				
复位后的值 用户的设定值 (注1)															

注 1. 空白产品为“FFFF FFFFh”，在用户编程后为编程后的值。

位	符号	位名	功能	R/W
b0	—	保留位	读取值为编程后的值。在编程时，必须置“1”。	R
b1	IWDTSTRT	IWDT 启动模式选择位	0: 复位后，IWDT 在自动启动模式中自动启动。 1: 复位后，IWDT 为停止状态。	R
b3-b2	IWDTTOPS[1:0]	IWDT 超时期间选择位	b3 b2 0 0: 1024 个周期 (03FFh) 0 1: 4096 个周期 (0FFFh) 1 0: 8192 个周期 (1FFFh) 1 1: 16384 个周期 (3FFFh)	R
b7-b4	IWDTCKS[3:0] (注1)	IWDT 时钟分频比选择位	b7 b4 0 0 0 0: 1 分频 (周期为 131ms) 0 0 1 0: 16 分频 (周期为 2.10s) 0 0 1 1: 32 分频 (周期为 4.19s) 0 1 0 0: 64 分频 (周期为 8.39s) 1 1 1 1: 128 分频 (周期为 16.8s) 0 1 0 1: 256 分频 (周期为 33.6s) 上述以外: 不能设定	R
b9-b8	IWDRPES[1:0]	IWDT 窗口结束位置选择位	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (没有窗口结束位置的设定)	R
b11-b10	IWDRPSS[1:0]	IWDT 窗口开始位置选择位	b11 b10 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (没有窗口开始位置的设定)	R
b12	IWDRSTIRQS	IWDT 复位中断请求选择位	0: 允许非屏蔽中断请求 1: 允许复位	R
b13	—	保留位	读取值为编程后的值。在编程时，必须置“1”。	R
b14	IWDTSLCSTP	IWDT 睡眠模式计数停止控制位	0: 计数停止无效 1: 在向睡眠模式、软件待机模式、深度软件待机模式和全模块时钟停止模式转移时，计数停止有效。	R
b16-b15	—	保留位	读取值为编程后的值。在编程时，必须置“1”。	R
b17	WDTSTRT	WDT 启动模式选择位	0: 复位后，WDT 在自动启动模式中自动启动。 1: 复位后，WDT 为停止状态。	R

位	符号	位名	功能	R/W
b19-b18	WDTTOPS[1:0]	WDT 超时期间选择位	b19 b18 0 0: 1024 个周期 (03FFh) 0 1: 4096 个周期 (0FFFh) 1 0: 8192 个周期 (1FFFh) 1 1: 16384 个周期 (3FFFh)	R
b23-b20	WDTCKS[3:0]	WDT 时钟分频比选择位	b23 b20 0 0 0 1: PCLK/4 0 1 0 0: PCLK/64 1 1 1 1: PCLK/128 0 1 1 0: PCLK/512 0 1 1 1: PCLK/2048 1 0 0 0: PCLK/8192 上述以外: 不能设定	R
b25-b24	WDRPES[1:0]	WDT 窗口结束位置选择位	b25 b24 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (没有窗口结束位置的设定)	R
b27-b26	WDRPSS[1:0]	WDT 窗口开始位置选择位	b27 b26 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (没有窗口开始位置的设定)	R
b28	WDRSTIRQS	WDT 复位中断请求选择位	0: 允许非屏蔽中断请求 1: 允许复位	R
b31-29	—	保留位	读取值为编程后的值。在编程时, 必须置“1”。	R

OFS0 寄存器在内部 ROM 内, 必须和程序一起写, 但是在写后不能对 OFS0 寄存器进行追加写。如果擦除包括 OFS0 寄存器的块, OFS0 寄存器的值就变为“FFFF FFFFh”。在引导模式中, OFS0 寄存器的设定无效, 值为“FFFF FFFFh”。

IWDTSTRT 位 (IWDT 启动模式选择位)

能选择复位后的 IWDT 启动模式 (停止状态或者在自动启动模式中的启动)。在自动启动模式中启动时, IWDT 的设定为 OFS0 寄存器的设定有效。

IWDTTOPS[1:0] 位 (IWDT 超时期间选择位)

以 IWDTCKS[3:0] 位设定的分频时钟为 1 个周期, 递减计数器发生下溢为止的超时期间可选择 1024 个周期、4096 个周期、8192 个周期或者 16384 个周期。

从刷新后到发生下溢为止的时间 (IWDT 专用低速时钟数) 取决于 IWDTCKS[3:0] 位和 IWDTTOPS[1:0] 位的组合。

详细内容请参照“26. 独立看门狗定时器 (IWDTa) ”。

IWDTCK[3:0] 位 (IWDT 时钟分频比选择位)

对 IWDT 专用低速时钟进行分频的预分频器的分频比可选择 1 分频、16 分频、32 分频、64 分频、128 分频或者 256 分频。能通过和 IWDTTOPS[1:0] 位的组合，将 IWDT 的计数期间设定在 IWDT 专用低速时钟的 1024 ~ 4194304 个时钟之间。

详细内容请参照“26. 独立看门狗定时器 (IWDTa) ”。

IWDRPES[1:0] 位 (IWDT 窗口结束位置选择位)

递减计数器的窗口结束位置可选择计数期间的 75%、50%、25% 或者 0%。窗口结束位置必须选择小于窗口开始位置的值 (窗口开始位置 > 窗口结束位置)。如果将窗口结束位置设定为大于窗口开始位置的值，就只有窗口开始位置的设定有效。

IWDRPSS[1:0] 位和 IWDRPES[1:0] 位设定的窗口开始 / 结束位置的计数器值因 IWDTTOPS[1:0] 位的设定而不同。

详细内容请参照“26. 独立看门狗定时器 (IWDTa) ”。

IWDRPSS[1:0] 位 (IWDT 窗口开始位置选择位)

递减计数器的窗口开始位置可选择计数期间 (开始计数时为 100%，发生下溢时为 0%) 的 100%、75%、50% 或者 25%。从窗口开始位置到窗口结束位置的期间为刷新允许期间，其他期间为刷新禁止期间。

详细内容请参照“26. 独立看门狗定时器 (IWDTa) ”。

IWDRSTIRQS 位 (IWDT 复位中断请求选择位)

设定递减计数器发生下溢或者刷新错误时的运行。能选择独立看门狗定时器复位或者非屏蔽中断请求。

详细内容请参照“26. 独立看门狗定时器 (IWDTa) ”。

IWDTSLCSTP 位 (IWDT 睡眠模式计数停止控制位)

选择向睡眠模式、软件待机模式、深度软件待机模式和全模块时钟停止模式转移时的计数停止。

详细内容请参照“26. 独立看门狗定时器 (IWDTa) ”。

WDTSTRT 位 (WDT 启动模式选择位)

能选择复位后的 WDT 启动模式 (停止状态或者在自动启动模式中的启动)。在自动启动模式中启动时，WDT 的设定为 OFS0 寄存器的设定有效。

WDTTOPS[1:0] 位 (WDT 超时期间选择位)

以 WDTCK[3:0] 位设定的分频时钟为 1 个周期，递减计数器发生下溢为止的超时期间可选择 1024 个周期、4096 个周期、8192 个周期或者 16384 个周期。

从刷新后到发生下溢为止的时间 (PCLK) 取决于 WDTCK[3:0] 和 WDTTOPS[1:0] 位的组合。

详细内容请参照“25. 看门狗定时器 (WDTA) ”。

WDTCK[3:0] 位 (WDT 时钟分频比选择位)

对 PCLK 进行分频的预分频器的分频比可选择 4 分频、64 分频、128 分频、512 分频、2048 分频或者 8192 分频。能通过和 WDTTOPS[1:0] 位的组合, 将 WDT 的计数期间设定在 PCLK 的 4096 ~ 134217728 个时钟之间。

详细内容请参照“25. 看门狗定时器 (WDTA)”。

WDTRPES[1:0] 位 (WDT 窗口结束位置选择位)

递减计数器的窗口结束位置可选择计数期间的 75%、50%、25% 或者 0%。窗口结束位置必须选择小于窗口开始位置的值 (窗口开始位置 > 窗口结束位置)。如果将窗口结束位置设定为大于窗口开始位置的值, 就只有窗口开始位置的设定有效。

WDTRPSS[1:0] 位和 WDTRPES[1:0] 位设定的窗口开始 / 结束位置的计数器值因 WDTTOPS[1:0] 位的设定而不同。

详细内容请参照“25. 看门狗定时器 (WDTA)”。

WDTRPSS[1:0] 位 (WDT 窗口开始位置选择位)

递减计数器的窗口开始位置可选择计数期间 (开始计数时为 100%, 发生下溢时为 0%) 的 100%、75%、50% 或者 25%。从窗口开始位置到窗口结束位置的期间为刷新允许期间, 其他期间为刷新禁止期间。

详细内容请参照“25. 看门狗定时器 (WDTA)”。

WDTIRSTIRQS 位 (WDT 复位中断请求选择位)

设定递减计数器发生下溢或者刷新错误时的运行。能选择看门狗定时器复位或者非屏蔽中断请求。

详细内容请参照“25. 看门狗定时器 (WDTA)”。

7.2.2 选项功能选择寄存器 1 (OFS1)

地址 FFFF FF88h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
复位后的值															
用户的设定值 (注1)															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	—	—	—	LVDAS	—	—
复位后的值															
用户的设定值 (注1)															

注 1. 空白产品为“FFFF FFFFh”，在用户编程后为编程后的值。

位	符号	位名	功能	R/W
b1-b0	—	保留位	读取值为编程后的值。在编程时，必须置“1”。	R
b2	LVDAS	电压检测 0 电路启动位	0: 复位后，电压监视 0 复位有效。 1: 复位后，电压监视 0 复位无效。	R
b31-b3	—	保留位	读取值为编程后的值。在编程时，必须置“1”。	R

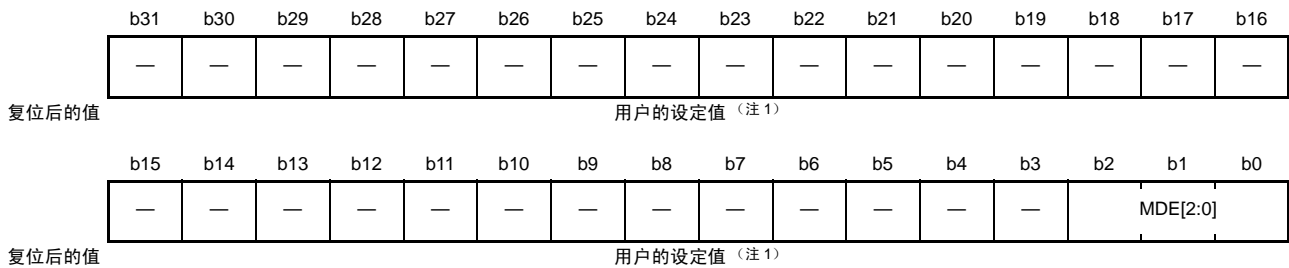
OFS1 寄存器在内部 ROM 内，必须和程序一起写，但是在写后不能对 OFS1 寄存器进行追加写。
如果擦除包括 OFS1 寄存器的块，OFS1 寄存器的值就变为“FFFF FFFFh”。

LVDAS 位 (电压检测 0 电路启动位)

在复位后，选择将电压监视 0 复位置为有效或者无效。

7.2.3 字节序选择寄存器 S (MDES)

地址 MDES (单芯片模式时): FFFF FF80h



注 1. 空白产品为“FFFF FFFFh”，在用户编程后为编程后的值。

位	符号	位名	功能	R/W
b2-b0	MDE[2:0]	字节序选择位	b2 b0 0 0 0: 大端法 1 1 1: 小端法 上述以外: 不能设定	R
b31-b3	—	保留位	读取值为编程后的值。在编程时, 必须置“1”。	R

MDES 寄存器是选择 CPU 字节序的寄存器。

MDES 寄存器在内部 ROM 内, 必须和程序一起写, 但是在写后不能对 MDES 寄存器进行追加写。

如果擦除包括 MDES 寄存器的块, MDES 寄存器的值就变为“FFFF FFFFh”。

MDE[2:0] 位 (字节序选择位)

选择小端法或者大端法。

7.3 使用时的注意事项

7.3.1 选项设定存储器的设定例子

因为选项设定存储器在内部 ROM 内, 所以不能通过执行指令进行改写, 必须在编写程序时写适当的值。设定例子如下所示:

- 给 OFS0 寄存器设定“ffff ff8h”的情况


```
.org      0fff ff8ch
.lword    0fff ffff8h
```

注. 程序的格式因编译器而不同, 必须通过编译程序的使用手册进行确认。

8. 电压检测电路 (LVDA)

电压检测电路是能通过程序监视 VCC 引脚的输入电压的电路。

8.1 概要

电压检测 0 能通过选项功能选择寄存器 1 (OFS1) 选择复位后的电压监视 0 复位的有效或者无效。

电压检测 1 和电压检测 2 能通过电压检测电平选择寄存器 (LVDLVLR) 设定检测电压。

能使用电压监视 0 复位、电压监视 1 复位 / 中断和电压监视 2 复位 / 中断。

电压检测电路的规格如表 8.1 所示, 电压检测电路的框图、电压监视 1 中断 / 复位发生电路的框图和电压监视 2 中断 / 复位发生电路的框图分别如图 8.1、图 8.2 和图 8.3 所示。

表 8.1 电压检测电路的规格

项目		电压监视 0	电压监视 1	电压监视 2
VCC 监视	监视电压	Vdet0	Vdet1	Vdet2
	检测对象	在下降过程中是否经过 Vdet0。	在上升或者下降过程中是否经过 Vdet1。	在上升或者下降过程中是否经过 Vdet2。
	检测电压	固定为高电平。	通过 LVDLVLR.LVD1LVL[3:0] 位进行指定。	通过 LVDLVLR.LVD2LVL[3:0] 位进行指定。
	监视标志	无	LVD1SR.LVD1MON 标志: 监视是高于还是低于 Vdet1。 LVD1SR.LVD1DET 标志: 检测是否经过 Vdet1。	LVD2SR.LVD2MON 标志: 监视是高于还是低于 Vdet2。 LVD2SR.LVD2DET 标志: 检测是否经过 Vdet2。
电压检测时的处理	复位	电压监视 0 复位 当 Vdet0 > VCC 时, 复位: 从 VCC > Vdet0 开始经过一定的时间后 CPU 重新开始运行。	电压监视 1 复位 当 Vdet1 > VCC 时, 复位: 可选择从 VCC > Vdet1 开始经过一定的时间后 CPU 重新开始运行或者从 Vdet1 > VCC 开始经过一定的时间后 CPU 重新开始运行。	电压监视 2 复位 当 Vdet2 > VCC 时, 复位: 可选择从 VCC > Vdet2 开始经过一定的时间后 CPU 重新开始运行或者从 Vdet2 > VCC 开始经过一定的时间后 CPU 重新开始运行。
	中断	无	电压监视 1 中断 非屏蔽中断	电压监视 2 中断 非屏蔽中断
			在 Vdet1 > VCC 和 VCC > Vdet1 时, 都产生中断请求、在 Vdet1 > VCC 时产生中断请求、或者在 VCC > Vdet1 时产生中断请求。	在 Vdet2 > VCC 和 VCC > Vdet2 时, 都产生中断请求、在 Vdet2 > VCC 时产生中断请求、或者在 VCC > Vdet2 时产生中断请求。
数字滤波器	有效 / 无效的选择	无数字滤波器功能	有	有
	采样时间	—	LOCO 的 n 分频 × 2 (n: 1、2、4、8)	LOCO 的 n 分频 × 2 (n: 1、2、4、8)

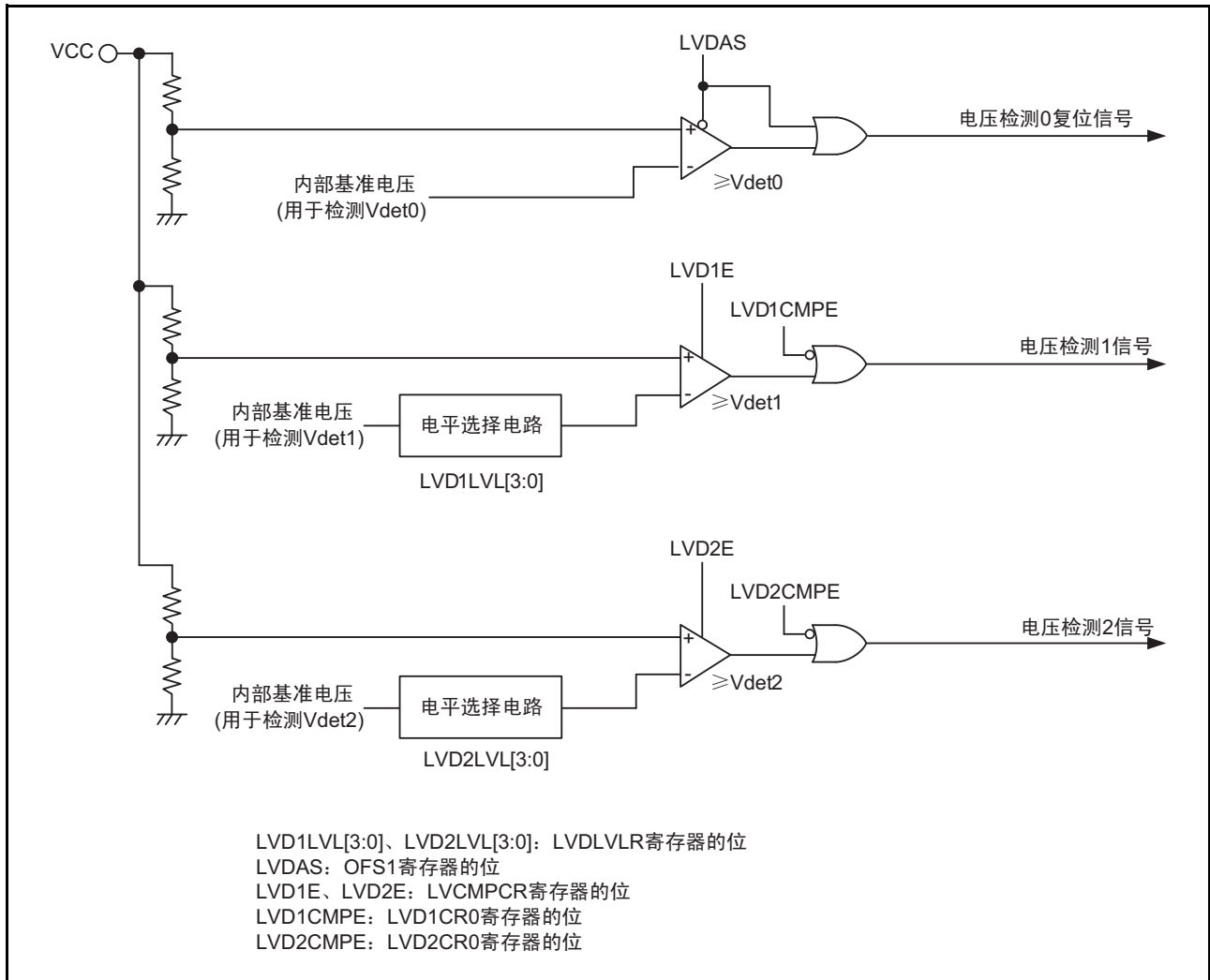


图 8.1 电压检测电路的框图

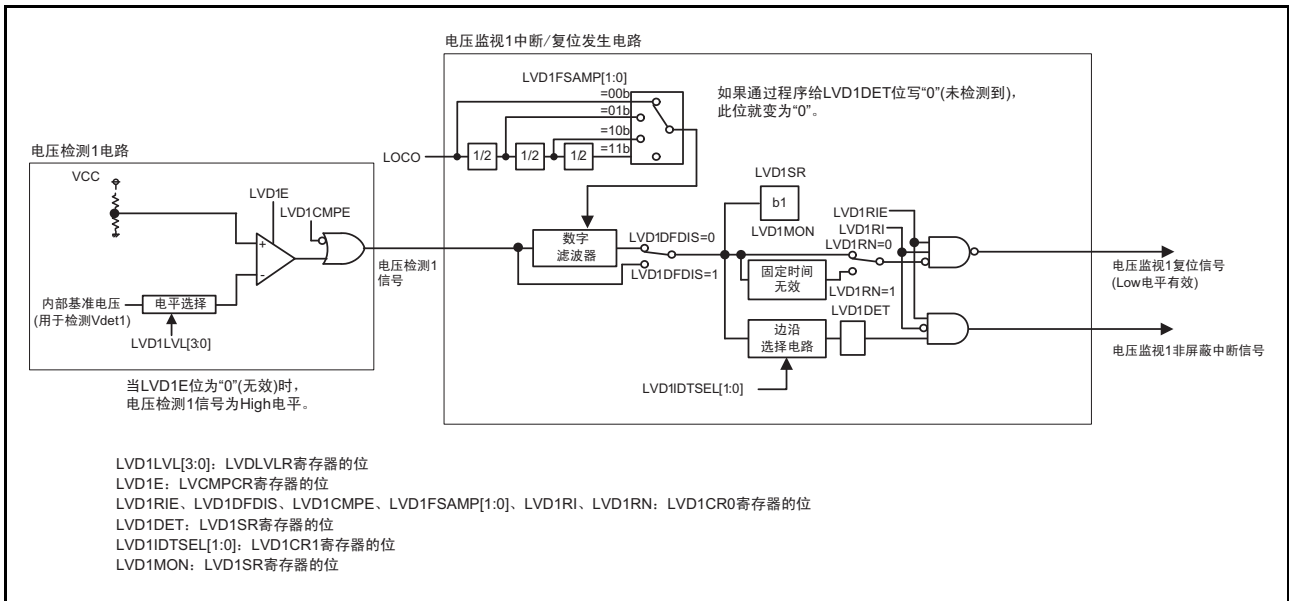


图 8.2 电压监视 1 中断 / 复位发生电路的框图

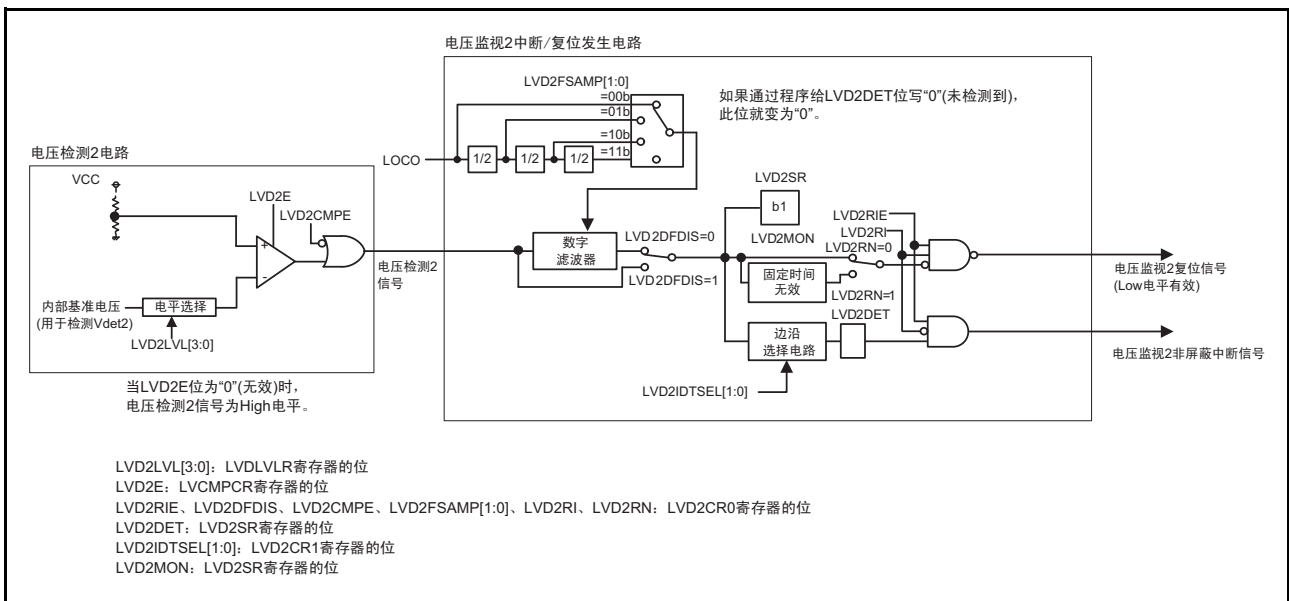


图 8.3 电压监视 2 中断 / 复位发生电路的框图

8.2 寄存器说明

8.2.1 电压监视 1 电路的控制寄存器 1 (LVD1CR1)

地址 0008 00E0h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	LVD1IDTSEL[1:0]	
0	0	0	0	0	0	0	1

复位后的值

位	符号	位名	功能	R/W
b1-b0	LVD1IDTSEL[1:0]	电压监视 1 中断发生条件选择位	b1 b0 0 0: 当检测到 $VCC \geq V_{det1}$ (上升) 时 0 1: 当检测到 $VCC < V_{det1}$ (下降) 时 1 0: 当检测到下降和上升时 1 1: 不能设定	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

8.2.2 电压监视 1 电路的状态寄存器 (LVD1SR)

地址 0008 00E1h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	LVD1MON	LVD1DET
0	0	0	0	0	0	1	0

复位后的值

位	符号	位名	功能	R/W
b0	LVD1DET	电压监视 1 电压变化检测标志	0: 未检测到 1: 检测到经过 V_{det1}	R(W) (注 1)
b1	LVD1MON	电压监视 1 信号监视标志	0: $VCC < V_{det1}$ 1: $VCC \geq V_{det1}$ 或者 LVD1MON 无效	R
b7-b2	—	保留位	读写值都为“0”。	R/W

注 1. 只能写“0”。从写“0”后被反映为 LVD1DET 位的读取值为止, 需要 2 个系统时钟周期。

LVD1DET 标志 (电压监视 1 电压变化检测标志)

当 LVCMPCR.LVD1E 位为“1” (电压检测 1 电路有效) 并且 LVD1CR0.LVD1CMPE 位为“1” (允许输出电压监视 1 电路的比较结果) 时, LVD1DET 标志有效。

如果改写 LVD1CR0 寄存器和 LVD1CR1 寄存器, LVD1DET 标志就可能变为“1”。

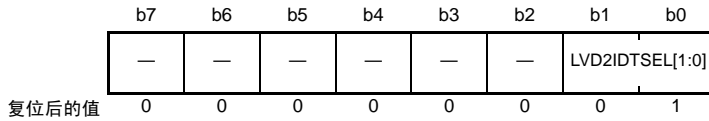
要将 LVD1DET 标志置“0”时, 必须先将 LVD1CR0.LVD1RIE 位置“0” (禁止)。在将 LVD1RIE 置“0”后, 如果要再次将 LVD1CR0.LVD1RIE 位置“1” (允许), 就必须至少经过 2 个 PCLKB 周期后进行设定。能通过读由 PCLKB 定义存取周期数的 I/O 寄存器, 至少确保 2 个 PCLKB 周期的等待时间。

LVD1MON 标志 (电压监视 1 信号监视标志)

当 LVCMPCR.LVD1E 位为“1” (电压检测 1 电路有效) 并且 LVD1CR0.LVD1CMPE 位为“1” (允许输出电压监视 1 电路的比较结果) 时, LVD1MON 标志有效。

8.2.3 电压监视 2 电路的控制寄存器 1 (LVD2CR1)

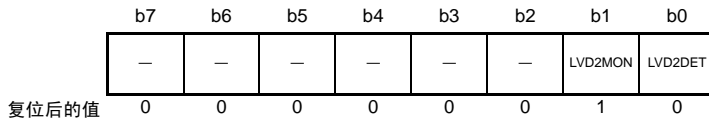
地址 0008 00E2h



位	符号	位名	功能	R/W
b1-b0	LVD2IDTSEL[1:0]	电压监视 2 中断发生条件选择位	b1 b0 0 0: 当检测到 $VCC \geq V_{det2}$ (上升) 时 0 1: 当检测到 $VCC < V_{det2}$ (下降) 时 1 0: 当检测到下降和上升时 1 1: 不能设定	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

8.2.4 电压监视 2 电路的状态寄存器 (LVD2SR)

地址 0008 00E3h



位	符号	位名	功能	R/W
b0	LVD2DET	电压监视 2 电压变化检测标志	0: 未检测到 1: 检测到经过 V_{det2}	R(W) (注 1)
b1	LVD2MON	电压监视 2 信号监视标志	0: $VCC < V_{det2}$ 1: $VCC \geq V_{det2}$ 或者 LVD2MON 无效	R
b7-b2	—	保留位	读写值都为“0”。	R/W

注 1. 只能写“0”。从写“0”后到被反映为 LVD2DET 位的读取值为止, 需要 2 个系统时钟周期。

LVD2DET 标志 (电压监视 2 电压变化检测标志)

当 LVCMPCR.LVD2E 位为“1” (电压检测 2 电路有效) 并且 LVD2CR0.LVD2CMPE 位为“1” (允许输出电压监视 2 电路的比较结果) 时, LVD2DET 标志有效。

如果改写 LVD2CR0 寄存器和 LVD2CR1 寄存器, LVD2DET 标志就可能变为“1”。

要将 LVD2DET 标志置“0”时, 必须先将 LVD2CR0.LVD2RIE 位置“0” (禁止)。如果要再次将 LVD2CR0.LVD2RIE 位置“1” (允许), 就必须至少经过 2 个 PCLKB 周期后进行设定。能通过读由 PCLKB 定义存取周期数的 I/O 寄存器, 至少确保 2 个 PCLKB 周期的等待时间。

LVD2MON 标志 (电压监视 2 信号监视标志)

当 LVCMPCR.LVD2E 位为“1” (电压检测 2 电路有效) 并且 LVD2CR0.LVD2CMPE 位为“1” (允许输出电压监视 2 电路的比较结果) 时, LVD2MON 标志有效。

8.2.5 电压监视电路的控制寄存器 (LVCMPCR)

地址 0008 C297h

b7	b6	b5	b4	b3	b2	b1	b0
—	LVD2E	LVD1E	—	—	—	—	—
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b4-b0	—	保留位	读写值都为“0”。	R/W
b5	LVD1E	电压检测 1 允许位	0: 电压检测 1 电路无效 1: 电压检测 1 电路有效	R/W
b6	LVD2E	电压检测 2 允许位	0: 电压检测 2 电路无效 1: 电压检测 2 电路有效	R/W
b7	—	保留位	读写值都为“0”。	R/W

LVD1E 位 (电压检测 1 允许位)

在使用电压检测 1 的中断 / 复位或者 LVD1SR.LVD1MON 位时, 必须将 LVD1E 位置“1”。在将 LVD1E 位从“0”置为“1”并且经过 td(E-A) 后, 电压检测 1 电路运行。

LVD2E 位 (电压检测 2 允许位)

在使用电压检测 2 的中断 / 复位或者 LVD2SR.LVD2MON 位时, 必须将 LVD2E 位置“1”。在将 LVD2E 位从“0”置为“1”并且经过 td(E-A) 后, 电压检测 2 电路运行。

8.2.6 电压检测电平选择寄存器 (LVDLVLR)

地址 0008 C298h

b7	b6	b5	b4	b3	b2	b1	b0
LVD2LVL[3:0]				LVD1LVL[3:0]			
1	x	x	x	1	x	x	x

复位后的值

x: 不定值

位	符号	位名	功能	R/W
b3-b0	LVD1LVL[3:0]	电压检测 1 电平选择位 (电压下降时的标准电压)	b3 b2 b1 b0 1 0 1 0: 2.95V 在写时, 不能进行上述以外的设定。	R/W
b7-b4	LVD2LVL[3:0]	电压检测 2 电平选择位 (电压下降时的标准电压)	b7 b6 b5 b4 1 0 1 0: 2.95V 在写时, 不能进行上述以外的设定。	R/W

只能在 LVCMPCR.LVD1E 位和 LVCMPCR.LVD2E 位都为“0” (电压检测 n 电路无效) (n=1、2) 时更改 LVDLVLR 寄存器。

8.2.7 电压监视 1 电路的控制寄存器 0 (LVD1CR0)

地址 0008 C29Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD1RN	LVD1RI	LVD1FSAMP[1:0]	—	LVD1CMPE	LVD1DFDIS	LVD1RIE	
复位后的值	1	0	0 0	x	0	1	0	

x: 不定值

位	符号	位名	功能	R/W
b0	LVD1RIE	电压监视 1 中断 / 复位允许位	0: 禁止 1: 允许	R/W
b1	LVD1DFDIS	电压监视 1 数字滤波器的无效模式选择位	0: 数字滤波器有效 1: 数字滤波器无效	R/W
b2	LVD1CMPE	电压监视 1 电路的比较结果输出允许位	0: 禁止输出电压监视 1 电路的比较结果 1: 允许输出电压监视 1 电路的比较结果	R/W
b3	—	保留位	读取值为不定值, 只能写“0”。	R/W
b5-b4	LVD1FSAMP[1:0]	采样时钟选择位	b5 b4 0 0: LOCO 的 1 分频 0 1: LOCO 的 2 分频 1 0: LOCO 的 4 分频 1 1: LOCO 的 8 分频	R/W
b6	LVD1RI	电压监视 1 电路模式选择位	0: 在经过 Vdet1 时产生电压监视 1 中断 1: 在下降过程中经过 Vdet1 时产生电压监视 1 复位	R/W
b7	LVD1RN	电压监视 1 复位无效选择位	0: 从检测到 VCC > Vdet1 开始经过一定的时间 (tLVD1) 后无效 1: 从 LVD1 复位有效开始经过一定的时间 (tLVD1) 后无效	R/W

LVD1RIE 位 (电压监视 1 中断 / 复位允许位)

在 LVCMPCR.LVD1E 位为“1” (电压检测 1 电路有效) 并且 LVD1CMPE 位为“1” (允许输出电压监视 1 电路的比较结果) 时 LVD1RIE 位变为有效。

不能在闪存的编程 / 擦除过程中, 产生电压监视 1 复位以及电压监视 2 中断。

LVD1DFDIS 位 (电压监视 1 数字滤波器的无效模式选择位)

要将 LVD1DFDIS 位置“0” (数字滤波器电路有效) 时, 必须将 LOCOCR.LCSTP 位置“0” (LOCO 运行)。

在软件待机模式或者深度软件待机模式中使用电压监视 1 电路时, 必须将 LVD1DFDIS 位置“1” (数字滤波器电路无效)。

LVD1FSAMP[1:0] 位 (采样时钟选择位)

只能在 LVD1DFDIS 位为“1” (数字滤波器电路无效) 时改写 LVD1FSAMP[1:0] 位。当 LVD1DFDIS 位为“0” (数字滤波器电路有效) 时, 不能改写 LVD1FSAMP[1:0] 位。

LVD1RI 位 (电压监视 1 电路模式选择位)

当 LVD1RI 位为“1” (选择电压监视 1 复位) 或者 LVD2CR0.LVD2RI 位为“1” (选择电压监视 2 复位) 时, 不能转移到深度软件待机模式, 而转移到软件待机模式。要转移到深度软件待机模式时, 必须将 LVD1RI 位置“0” (选择电压监视 1 中断) 并且将 LVD2CR0.LVD2RI 位置“0” (选择电压监视 2 中断)。

LVD1RN 位 (电压监视 1 复位无效选择位)

要将 LVD1RN 位置“1” (从 LVD1 复位有效开始经过一定的时间后无效) 时, 必须将 LOCOCR.LCSTP 位置“0” (LOCO 运行)。要转移到软件待机模式或者深度软件待机模式时, 只能将 LVD1RN 位置“0” (从检测到 $VCC > V_{det1}$ 开始经过一定的时间后无效), 而不能将 LVD1RN 位置“1” (从 LVD1 复位有效开始经过一定的时间后无效)。

8.2.8 电压监视 2 电路的控制寄存器 0 (LVD2CR0)

地址 0008 C29Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2RN	LVD2RI	LVD2FSAMP[1:0]	—	LVD2CMPE	LVD2DFDIS	LVD2RIE	
复位后的值	1	0	0 0	x	0	1	0	

x: 不定值

位	符号	位名	功能	R/W
b0	LVD2RIE	电压监视 2 中断 / 复位允许位	0: 禁止 1: 允许	R/W
b1	LVD2DFDIS	电压监视 2 数字滤波器的无效模式选择位	0: 数字滤波器有效 1: 数字滤波器无效	R/W
b2	LVD2CMPE	电压监视 2 电路的比较结果输出允许位	0: 禁止输出电压监视 2 电路的比较结果 1: 允许输出电压监视 2 电路的比较结果	R/W
b3	—	保留位	读取值为不定值, 只能写“0”。	R/W
b5-b4	LVD2FSAMP[1:0]	采样时钟选择位	b5 b4 0 0: LOCO 的 1 分频 0 1: LOCO 的 2 分频 1 0: LOCO 的 4 分频 1 1: LOCO 的 8 分频	R/W
b6	LVD2RI	电压监视 2 电路模式选择位	0: 在经过 Vdet2 时产生电压监视 2 中断 1: 在下降过程中经过 Vdet2 时产生电压监视 2 复位	R/W
b7	LVD2RN	电压监视 2 复位无效选择位	0: 从检测到 VCC > Vdet2 开始经过一定的时间 (tLVD2) 后无效 1: 从 LVD2 复位有效开始经过一定的时间 (tLVD2) 后无效	R/W

LVD2RIE 位 (电压监视 2 中断 / 复位允许位)

在 LVCMPCR.LVD2E 位为“1” (电压检测 2 电路有效) 并且 LVD2CMPE 位为“1” (允许输出电压监视 2 电路的比较结果) 时 LVD2RIE 位变为有效。

不能在闪存的编程 / 擦除过程中, 产生电压监视 2 复位以及电压监视 2 中断。

LVD2DFDIS 位 (电压监视 2 数字滤波器的无效模式选择位)

要将 LVD2DFDIS 位置“0” (数字滤波器电路有效) 时, 必须将 LOCOCR.LCSTP 位置“0” (LOCO 运行)。

在软件待机模式或者深度软件待机模式中使用电压监视 2 电路时, 必须将 LVD2DFDIS 位置“1” (数字滤波器电路无效)。

LVD2FSAMP[1:0] 位 (采样时钟选择位)

只能在 LVD2DFDIS 位为“1” (数字滤波器电路无效) 时改写 LVD2FSAMP[1:0] 位。当 LVD2DFDIS 位为“0” (数字滤波器电路有效) 时, 不能改写 LVD2FSAMP[1:0] 位。

LVD2RI 位 (电压监视 2 电路模式选择位)

当 LVD2RI 位为“1” (选择电压监视 2 复位) 或者 LVD1CR0.LVD1RI 位为“1” (选择电压监视 1 复位) 时, 不能转移到深度软件待机模式, 而转移到软件待机模式。要转移到深度软件待机模式时, 必须将 LVD2RI 位置“0” (选择电压监视 2 中断) 并且将 LVD1CR0.LVD1RI 位置“0” (选择电压监视 1 中断)。

LVD2RN 位 (电压监视 2 复位无效选择位)

要将 LVD2RN 位置“1” (从 LVD2 复位有效开始经过一定的时间后无效) 时, 必须将 LOCOCR.LCSTP 位置“0” (LOCO 运行)。要转移到软件待机模式或者深度软件待机模式时, 只能将 LVD2RN 位置“0” (从检测到 $VCC > V_{det2}$ 开始经过一定的时间后无效), 而不能将 LVD2RN 位置“1” (从 LVD2 复位有效开始经过一定的时间后无效)。

8.3 VCC 输入电压的监视

8.3.1 Vdet0 的监视

不能监视 Vdet0。

8.3.2 Vdet1 的监视

Vdet1 的监视设定步骤如表 8.2 所示。在进行设定后，能通过 LVD1SR.LVD1MON 标志监视电压监视 1 的比较结果。

表 8.2 Vdet1 的监视设定步骤

步骤	使用数字滤波器的情况	不使用数字滤波器的情况
1	通过 LVDLVLRL.LVD1LVL[3:0] 位设定检测电压。	
2	通过 LVD1CR0.LVD1FSAMP[1:0] 位选择数字滤波器的采样时钟。	将 LVD1CR0.LVD1DFDIS 位置“1”（数字滤波器无效）。
3	将 LVD1CR0.LVD1CMPE 位置“1”（允许输出电压监视 1 电路的比较结果）。	
4	至少等待 1 个 LOCO 周期。	—
5	将 LVD1CR0.LVD1DFDIS 位置“0”（数字滤波器有效）。	—
6	至少等待 $2n+3$ 个 LOCO 周期。（ $n=1、2、4、8$ ：数字滤波器的采样时钟 =LOCO 的 n 分频）	—（无等待时间）
7	将 LVCMPCLR.LVD1E 位置“1”（电压检测 1 电路有效）。	

8.3.3 Vdet2 的监视

Vdet2 的监视设定步骤如表 8.3 所示。在进行设定后，能通过 LVD2SR.LVD2MON 标志监视电压监视 2 的比较结果。

表 8.3 Vdet2 的监视的设定步骤

步骤	使用数字滤波器的情况	不使用数字滤波器的情况
1	通过 LVDLVLRL.LVD2LVL[3:0] 位设定检测电压。	
2	通过 LVD2CR0.LVD2FSAMP[1:0] 位选择数字滤波器的采样时钟。	将 LVD2CR0.LVD2DFDIS 位置“1”（数字滤波器无效）。
3	将 LVD2CR0.LVD2CMPE 位置“1”（允许输出电压监视 2 电路的比较结果）。	
4	至少等待 1 个 LOCO 周期。	—
5	将 LVD2CR0.LVD2DFDIS 位置“0”（数字滤波器有效）。	—
6	至少等待 $2n+3$ 个 LOCO 周期。（ $n=1、2、4、8$ ：数字滤波器的采样时钟 =LOCO 的 n 分频）	—（无等待时间）
7	将 LVCMPCLR.LVD2E 位置“1”（电压检测 2 电路有效）。	

8.4 电压监视 0 复位

在使用电压监视 0 复位时，必须将电压检测 0 的电路启动位 (OFS1.LVDAS) 置“0” (复位后，电压监视 0 复位有效)。

电压监视 0 复位的运行例子如图 8.4 所示。

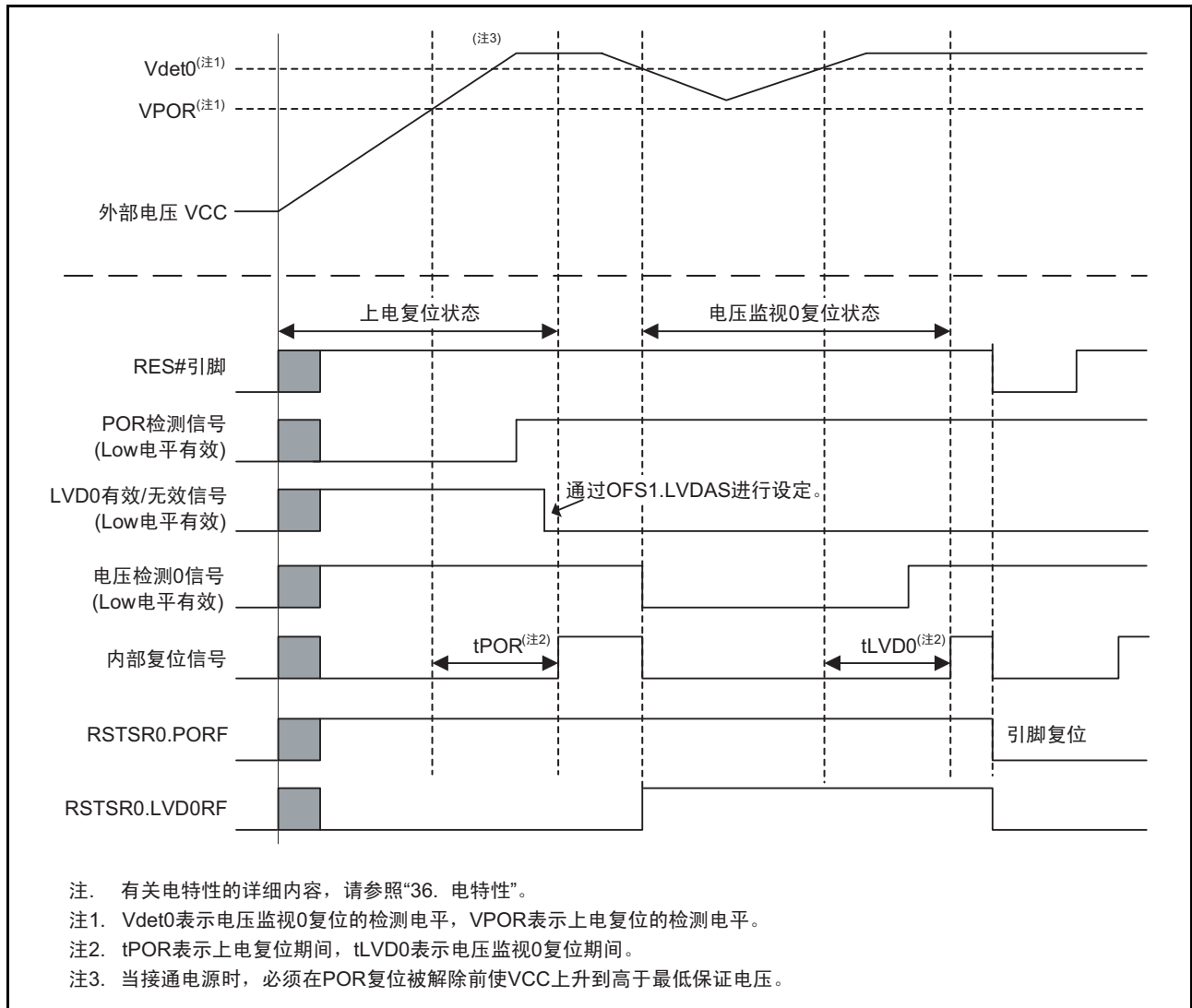


图 8.4 电压监视 0 复位的运行例子

8.5 电压监视 1 中断和电压监视 1 复位

电压监视 1 中断和电压监视 1 复位的相关位的运行设定步骤如表 8.4 所示，电压监视 1 中断和电压监视 1 复位的相关位的停止设定步骤如表 8.5 所示，电压监视 1 中断的运行例子如图 8.5 所示。有关电压监视 1 复位的运行例子，请参照“6. 复位”的图 6.2。

在软件待机模式或者深度软件待机模式中使用电压监视 1 电路时，必须将 LVD1CR0.LVD1DFDIS 位置“1”（数字滤波器无效）。

表 8.4 电压监视 1 中断和电压监视 1 复位的相关位的运行设定步骤

步骤	使用数字滤波器的情况		不使用数字滤波器的情况	
	电压监视 1 中断	电压监视 1 复位	电压监视 1 中断	电压监视 1 复位
1 (注 1)	通过 LVDLVL.R.LVD1LVL[3:0] 位设定检测电压。			
2 (注 2)	通过 LVD1CR0.LVD1FSAMP[1:0] 位选择数字滤波器的采样时钟。		将 LVD1CR0.LVD1DFDIS 位置“1”（数字滤波器无效）。	
3 (注 1、注 2)	将 LVD1CR0.LVD1RI 位置“0”（电压监视 1 中断）。	<ul style="list-style-type: none"> 将 LVD1CR0.LVD1RI 位置“1”（电压监视 1 复位）。 通过 LVD1CR0.LVD1RN 位选择复位无效的种类。 	将 LVD1CR0.LVD1RI 位置“0”（电压监视 1 中断）。	<ul style="list-style-type: none"> 将 LVD1CR0.LVD1RI 位置“1”（电压监视 1 复位）。 通过 LVD1CR0.LVD1RN 位选择复位无效的种类。
4	通过 LVD1CR1.LVD1IDTSEL [1:0] 位选择中断请求时序。	—	通过 LVD1CR1.LVD1IDTSEL [1:0] 位选择中断请求时序。	—
5	将 LVD1CR0.LVD1CMPE 位置“1”（允许输出电压监视 1 电路的比较结果）。			
6 (注 2)	至少等待 1 个 LOCO 周期。		—	
7	将 LVD1CR0.LVD1DFDIS 位置“0”（数字滤波器有效）。		—	
8	至少等待 $2n+3$ 个 LOCO 周期（ $n=1、2、4、8$ ：数字滤波器的采样时钟 = LOCO 的 n 分频）		—（没有等待时间）	
9	将 LVD1SR.LVD1DET 标志位置“0”。	—	将 LVD1SR.LVD1DET 标志位置“0”。	—
10	将 LVD1CR0.LVD1RIE 位置“1”（允许电压监视 1 中断 / 复位）。			
11 (注 1)	将 LVCMP.R.LVD1E 位置“1”（电压检测 1 电路有效）。			

注 1. 在以下的任何一种情况下，不需要步骤 1、步骤 3 和步骤 11：

- 在通过设定为电压监视 1 中断（LVD1CR0.LVD1RI=0）而运行时
 - 在停止后只更改 LVD1CR0.LVD1FSAMP[1:0] 位、LVD1CR0.LVD1DFDIS 位或者 LVD1CR1.LVD1IDTSEL[1:0] 位的设定而重新运行时
 - 在停止后不更改电压检测电路的相关设定而重新运行时
- 对于通过设定为电压监视 1 复位（LVD1CR0.LVD1RI=1）而运行时的更改，必须通过步骤 1 ~ 11 进行设定。

注 2. 可以用 1 条指令同时执行步骤 2 和步骤 3。

表 8.5 电压监视 1 中断和电压监视 1 复位的相关位的停止设定步骤

步骤	电压监视 1 中断和电压监视 1 复位的相关位的停止设定
1 (注 1)	将 LVCMP.R.LVD1E 位置“0”（电压检测 1 电路无效）。
2 (注 1)	至少等待 1 个 LOCO 周期。
3	将 LVD1CR0.LVD1RIE 位置“0”（禁止电压监视 1 中断 / 复位）。
4	将 LVD1CR0.LVD1CMPE 位置“0”（禁止输出电压监视 1 电路的比较结果）。
5	更改除 LVCMP.R.LVD1E 位、LVD1CR0.LVD1CMPE 和 LVD1CR0.LVD1RIE 位以外的电压检测电路相关寄存器的设定。

注 1. 在以下的任何一种情况下，不需要步骤 1 和步骤 2：

- 在通过设定为电压监视 1 中断（LVD1CR0.LVD1RI=0）而运行时
 - 在停止后只更改 LVD1CR0.LVD1FSAMP[1:0] 位、LVD1CR0.LVD1DFDIS 位或者 LVD1CR1.LVD1IDTSEL[1:0] 位的设定而重新运行时
 - 在停止后不更改电压检测电路的相关设定而重新运行时
- 对于通过设定为电压监视 1 复位（LVD1CR0.LVD1RI=1）而运行时的更改，必须通过步骤 1 ~ 5 进行设定。

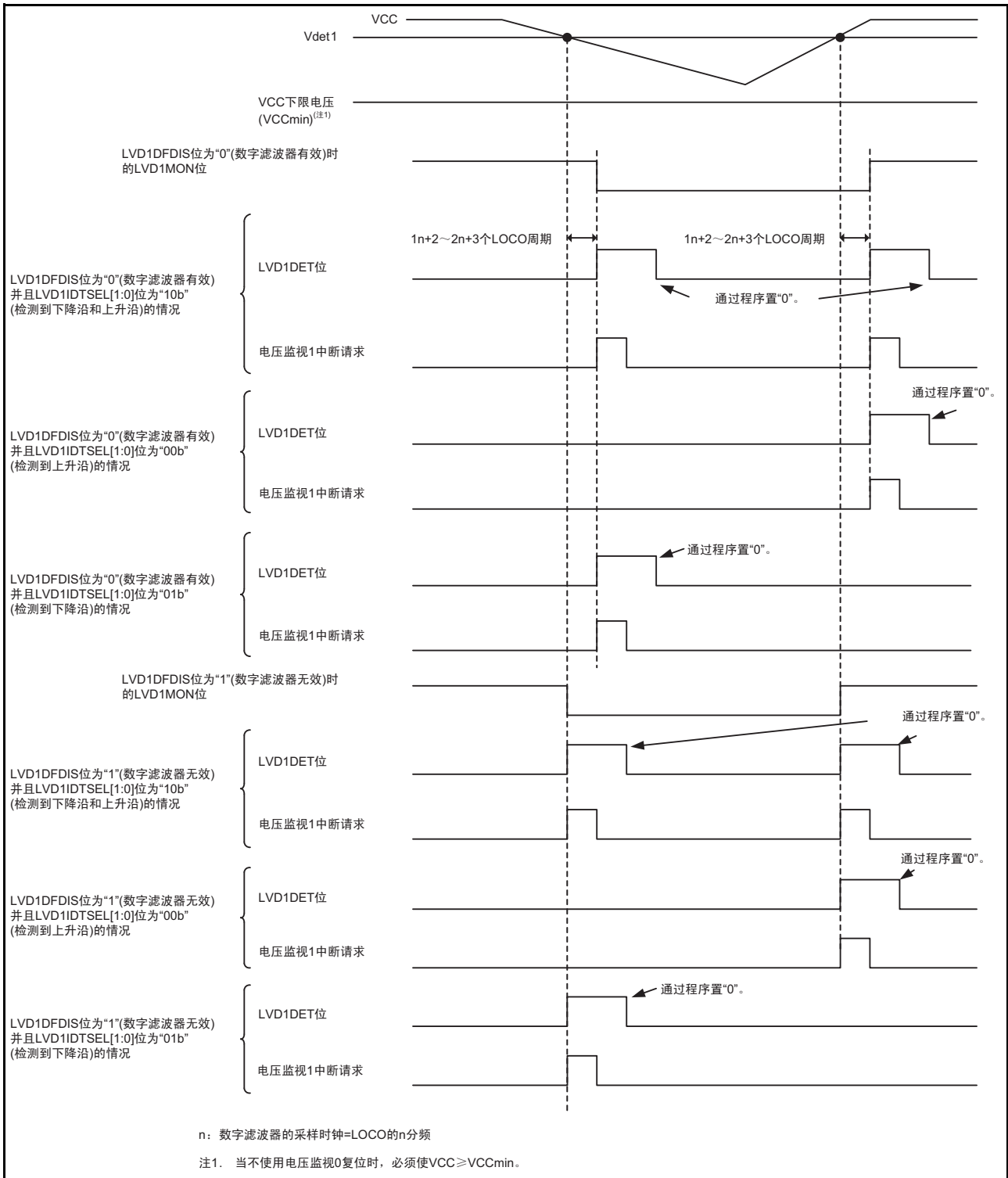


图 8.5 电压监视 1 中断的运行例子

8.6 电压监视 2 中断和电压监视 2 复位

电压监视 2 中断和电压监视 2 复位的相关位的运行设定步骤如表 8.6 所示，电压监视 2 中断和电压监视 2 复位的相关位的停止设定步骤如表 8.7 所示，电压监视 2 中断的运行例子如图 8.6 所示。有关电压监视 2 复位的运行例子，请参照“6. 复位”的图 6.2。

在软件待机模式或者深度软件待机模式中使用电压监视 2 电路时，必须将 LVD2CR0.LVD2DFDIS 位置“1”（数字滤波器无效）。

表 8.6 电压监视 2 中断和电压监视 2 复位的相关位的运行设定步骤

步骤	使用数字滤波器的情况		不使用数字滤波器的情况	
	电压监视 2 中断	电压监视 2 复位	电压监视 2 中断	电压监视 2 复位
1 (注 1)	通过 LVDLVLR.LVD2LVL[3:0] 位设定检测电压。			
2 (注 2)	通过 LVD2CR0.LVD2FSAMP[1:0] 位选择数字滤波器的采样时钟。		将 LVD2CR0.LVD2DFDIS 位置“1”（数字滤波器无效）。	
3 (注 1、注 2)	将 LVD2CR0.LVD2RI 位置“0”（电压监视 2 中断）。	<ul style="list-style-type: none"> 将 LVD2CR0.LVD2RI 位置“1”（电压监视 2 复位）。 通过 LVD2CR0.LVD2RN 位选择复位无效的种类。 	将 LVD2CR0.LVD2RI 位置“0”（电压监视 2 中断）。	<ul style="list-style-type: none"> 将 LVD2CR0.LVD2RI 位置“1”（电压监视 2 复位）。 通过 LVD2CR0.LVD2RN 位选择复位无效的种类。
4	通过 LVD2CR1.LVD2IDTSEL[1:0] 位选择中断请求时序。	—	通过 LVD2CR1.LVD2IDTSEL[1:0] 位选择中断请求时序。	—
5	将 LVD2CR0.LVD2CMPE 位置“1”（允许输出电压监视 2 电路的比较结果）。			
6	至少等待 1 个 LOCO 周期。		—	
7	将 LVD2CR0.LVD2DFDIS 位置“0”（数字滤波器有效）。		—	
8	至少等待 $2n+3$ 个 LOCO 周期（ $n=1、2、4、8$ ：数字滤波器的采样时钟 = LOCO 的 n 分频）		—（没有等待时间）	
9	将 LVD2SR.LVD2DET 标志位置“0”。	—	将 LVD2SR.LVD2DET 标志位置“0”。	—
10	将 LVD2CR0.LVD2RIE 位置“1”（允许电压监视 2 中断 / 复位）。			
11 (注 1)	将 LVCMPER.LVD2E 位置“1”（电压检测 2 电路有效）。			

注 1. 在以下的任何一种情况下，不需要步骤 1、步骤 3 和步骤 11：
 • 在通过设定为电压监视 2 中断（LVD2CR0.LVD2RI=0）而运行时
 • 在停止后只更改 LVD2CR0.LVD2FSAMP[1:0] 位、LVD2CR0.LVD2DFDIS 位或者 LVD2CR1.LVD2IDTSEL[1:0] 位的设定而重新运行时
 • 在停止后不更改电压检测电路的相关设定而重新运行时
 对于通过设定为电压监视 2 复位（LVD2CR0.LVD2RI=1）而运行时的更改，必须通过步骤 1～11 进行设定。

注 2. 可以用 1 条指令同时执行步骤 2 和步骤 3。

表 8.7 电压监视 2 中断和电压监视 2 复位的相关位的停止设定步骤

步骤	电压监视 2 中断和电压监视 2 复位的相关位的停止设定
1 (注 1)	将 LVCMPER.LVD2E 位置“0”（电压检测 2 电路无效）。
2 (注 1)	至少等待 1 个 LOCO 周期。
3	将 LVD2CR0.LVD2RIE 位置“0”（禁止电压监视 2 中断 / 复位）。
4	将 LVD2CR0.LVD2CMPE 位置“0”（禁止输出电压监视 2 电路的比较结果）。
5	更改除 LVCMPER.LVD2E 位、LVD2CR0.LVD2CMPE 位和 LVD2CR0.LVD2RIE 位以外的电压检测电路相关寄存器的设定。

注 1. 在以下的任何一种情况下，不需要步骤 1 和步骤 2：
 • 在通过设定为电压监视 2 中断（LVD2CR0.LVD2RI=0）而运行时
 • 在停止后只更改 LVD2CR0.LVD2FSAMP[1:0] 位、LVD2CR0.LVD2DFDIS 位或者 LVD2CR1.LVD2IDTSEL[1:0] 位的设定而重新运行时
 • 在停止后不更改电压检测电路的相关设定而重新运行时
 对于通过设定为电压监视 2 复位（LVD2CR0.LVD2RI=1）而运行时的更改，必须通过步骤 1～5 进行设定。

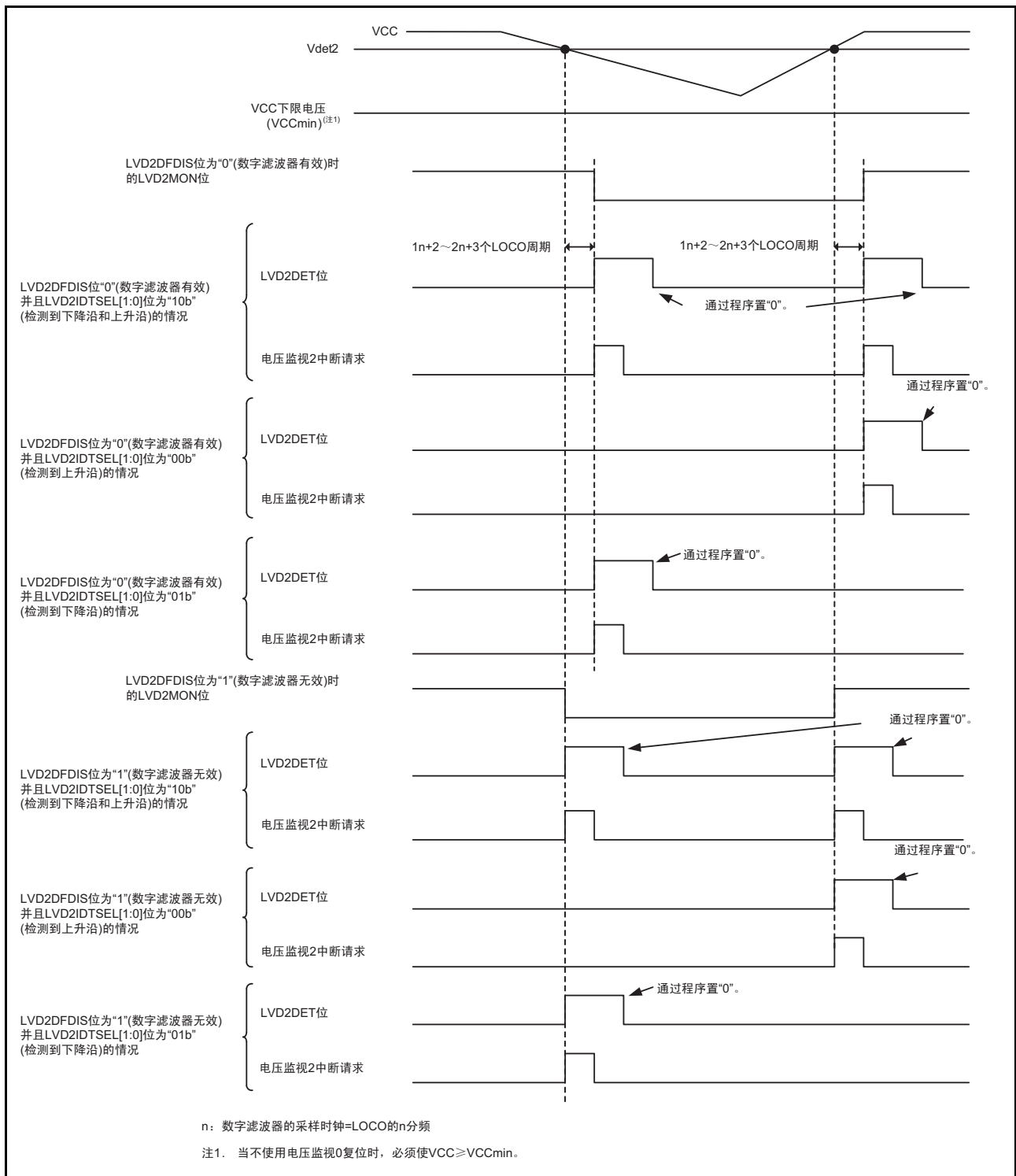


图 8.6 电压监视 2 中断的运行例子

9. 时钟发生电路

9.1 概要

RX63T 群内置时钟发生电路。

时钟发生电路的规格和框图分别如表 9.1 和图 9.1 所示。

表 9.1 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、DMAC、DTC、内部 ROM 和内部 RAM 提供的系统时钟 (ICLK)。 生成给 MTU3 和 GPT 提供的定时器模块时钟 (PCLKA)。(注 1) 生成给外围模块提供的外围模块时钟 (PCLKB)。 生成给 S12AD 提供的 S12AD 时钟 (PCLKD)。(注 1) 生成给 FlashIF 提供的 FlashIF 时钟 (FCLK)。 生成给 CAC 提供的 CAC 时钟 (CACCLK)。 生成给 IWDTC 提供的 IWDTC 专用低速时钟 (IWDTCCLK)。 生成给 JTAG 提供的 JTAG 时钟 (JTAGTCK)。
工作频率	<ul style="list-style-type: none"> ICLK: 100MHz(max) PCLKA: 100MHz(max) PCLKB: 50MHz(max) PCLKD: 50MHz(max) FCLK: 4MHz ~ 50MHz (ROM、E2 数据闪存 P/E 时) 50MHz(max) (E2 数据闪存读时) CACCLK: 与各振荡器的时钟相同 IWDTCCLK: 125kHz JTAGTCK: 10MHz(max)
主时钟振荡器	<ul style="list-style-type: none"> 谐振器频率: 4MHz ~ 16MHz 外部时钟的输入频率: 20MHz(max) 能连接的谐振器或者附加电路: 陶瓷谐振器、晶体谐振器 连接引脚: EXTAL、XTAL 振荡停止检测功能: 这是在检测到主时钟停止振荡时转换到 LOCO 并且将 MTU 引脚和 GPT 引脚置为高阻抗的功能。
PLL 电路	<ul style="list-style-type: none"> 输入时钟源: 主时钟 输入分频比: 可选择 1 分频、2 分频或者 4 分频。 输入频率: 4MHz ~ 16MHz 倍增比: 可选择递增 8 倍、10 倍、12 倍、16 倍、20 倍、24 倍、25 倍或者 50 倍。 VCO 振荡频率: 104MHz ~ 200MHz
低速内部振荡器 (LOCO)	振荡频率: 125kHz
IWDTC 专用内部振荡器	振荡频率: 125kHz
JTAG 外部时钟输入 (TCK)	输入时钟频率: 10MHz(max)

注 1. 在 RX63T 群中 PCLKB=PCLK。

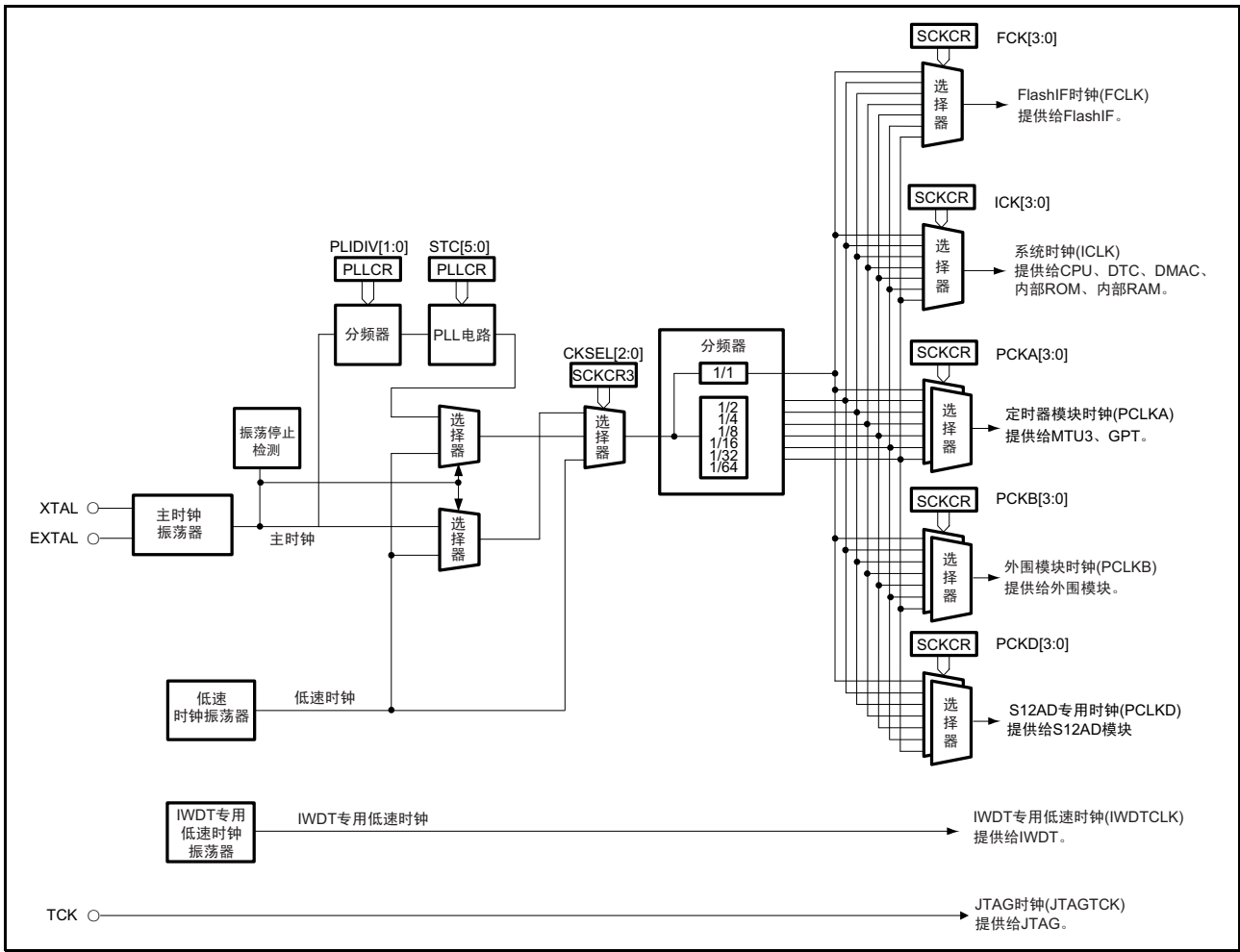


图 9.1 时钟发生电路的框图

时钟发生电路的输入 / 输出引脚如表 9.2 所示。

表 9.2 时钟发生电路的输入 / 输出引脚

引脚名	输入 / 输出	功能
XTAL	输出	连接谐振器的引脚
EXTAL	输入	EXTAL 引脚也能输入外部时钟，详细内容请参照“9.3.2 输入外部时钟的方法”。
TCK	输入	JTAG 的时钟输入引脚

9.2 寄存器说明

9.2.1 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
FCK[3:0]				ICK[3:0]				—	—	—	—	BCK[3:0]			
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
PCKA[3:0]				PCKB[3:0]				PCKC[3:0]				PCKD[3:0]			
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
b3-b0	PCKD[3:0]	S12AD 时钟 (PCLKD) 选择位 (注1、注4、注7)	b3 b0 0000: 1分频 0001: 2分频 0010: 4分频 0011: 8分频 0100: 16分频 0101: 32分频 0110: 64分频 上述以外: 不能设定	R/W
b7-b4	PCKC[3:0]	保留位	必须设定为与 ICK[3:0] 位和 PCKB[3:0] 位的设定值中频率大的值相同的值。	R/W
b11-b8	PCKB[3:0]	外围模块时钟 B (PCLKB) 选择位 (注1、注2、注4、注5、注7)	b11 b8 0000: 1分频 0001: 2分频 0010: 4分频 0011: 8分频 0100: 16分频 0101: 32分频 0110: 64分频 上述以外: 不能设定	R/W
b15-b12	PCKA[3:0]	定时器模块时钟 (PCLKA) 选择位 (注3~注5、注7)	b15 b12 0000: 1分频 0001: 2分频 0010: 4分频 0011: 8分频 0100: 16分频 0101: 32分频 0110: 64分频 上述以外: 不能设定	R/W
b19-b16	BCK[3:0]	保留位	必须设定为与 ICK[3:0] 位和 PCKB[3:0] 位的设定值中频率大的值相同的值。	R/W
b23-b20	—	保留位	读写值都为“0”。	R/W

位	符号	位名	功能	R/W
b27-b24	ICK[3:0]	系统时钟 (ICLK) 选择位 (注2、注3、注5~注7)	b27 b24 0000: 1分频 0001: 2分频 0010: 4分频 0011: 8分频 0100: 16分频 0101: 32分频 0110: 64分频 上述以外: 不能设定	R/W
b31-b28	FCK[3:0]	FlashIF 时钟 (FCLK) 选择位 (注5~注7)	b31 b28 0000: 1分频 0001: 2分频 0010: 4分频 0011: 8分频 0100: 16分频 0101: 32分频 0110: 64分频 上述以外: 不能设定	R/W

注1. 必须设定为 $PCLKB:PCLKD=N:1$ (N为整数) 的频率关系。

注2. 必须设定为 $ICLK:PCLKB=N:1or1:N$ (N为整数) 的频率关系。

注3. 必须设定为 $ICLK:PCLKA=N:1or1:N$ (N为整数) 的频率关系。

注4. 必须设定为 $PCLKA:PCLKB=N:1$ (N为整数) 的频率关系。

注5. 选择 PLL 时, 禁止设定为 1 分频。

注6. 必须设定为 $ICLK:FCLK=N:1or1:N$ (N为整数) 的频率关系。

注7. 在 $SCKCR3.CKSEL[2:0]$ 位为“010b”(选择主时钟振荡器)时, 不能设定为 1 分频和 2 分频。

在以下情况下禁止改写 SCKCR 寄存器:

- 当闪存 P/E 模式入口寄存器的 ROM P/E 模式入口位 0 (FENTRYR.FENTRY0) 为“1”(ROM P/E 模式) 时
- 在从发行用于转移到睡眠模式的 WAIT 指令开始到从睡眠模式返回到正常运行模式为止的期间

9.2.2 系统时钟控制寄存器 3 (SCKCR3)

地址 0008 0026h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	CKSEL[2:0]		—	—	—	—	—	—	—	—	—
0	0	0	0	0	0	0	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
b10-b8	CKSEL[2:0]	时钟源选择位	b10 b8 0 0 0: 选择 LOCO 0 1 0: 选择主时钟振荡器 1 0 0: 选择 PLL 电路 上述以外: 不能设定	R/W
b15-b11	—	保留位	读写值都为“0”。	R/W

在以下情况下禁止改写 SCKCR3 寄存器。

- 当闪存 P/E 模式入口寄存器的 ROM P/E 模式入口位 0 (FENTRYR.FENTRY0) 为“1” (ROM P/E 模式) 时
- 在从发行用于转移到睡眠模式的 WAIT 指令开始到从睡眠模式返回到正常运行模式为止的期间

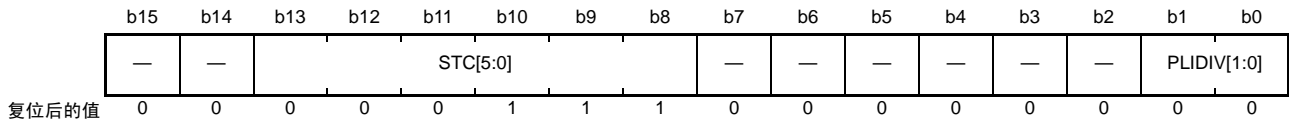
CKSEL[2:0] 位 (时钟源选择位)

从低速内部振荡器 (LOCO)、主时钟振荡器和 PLL 电路中选择系统时钟 (ICLK)、定时器模块时钟 (PCLKA)、外围模块时钟 (PCLKB)、S12AD 时钟 (PCLKD)、FlashIF 时钟 (FCLK) 的时钟源。

禁止转换到停止的时钟源。

9.2.3 PLL 控制寄存器 (PLLCR)

地址 0008 0028h



位	符号	位名	功能	R/W
b1-b0	PLIDIV[1:0]	PLL 输入分频比选择位	b1 b0 0 0: 1 分频 0 1: 2 分频 1 0: 4 分频 1 1: 不能设定	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W
b13-b8	STC[5:0]	频率倍增率设定位	b13 b8 0 0 0 1 1 1: x8 0 0 1 0 0 1: x10 0 0 1 0 1 1: x12 0 0 1 1 1 1: x16 0 1 0 0 1 1: x20 0 1 0 1 1 1: x24 0 1 1 0 0 0: x25 1 1 0 0 0 1: x50 上述以外: 不能设定	R/W
b15-b14	—	保留位	读写值都为“0”。	R/W

当 PLLCR2.PLEN 位为“0”（PLL 运行）时，禁止写 PLLCR 寄存器。

PLIDIV[1:0] 位（PLL 输入分频比选择位）

这些位选择 PLL 时钟源的输入分频比。

必须将 PLIDIV[1:0] 位设定在 PLL 输入频率（4 ~ 16MHz）的范围内。

STC[5:0] 位（频率倍增率设定位）

这些位设定 PLL 的频率倍增率。

必须将 STC[5:0] 位设定在 PLL 的 VCO 振荡频率（104 ~ 200MHz）的范围内。

9.2.4 PLL 控制寄存器 2 (PLLCR2)

地址 0008 002Ah

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	PLLEN
0	0	0	0	0	0	0	1

复位后的值

位	符号	位名	功能	R/W
b0	PLLEN	PLL 停止控制位	0: PLL 运行 1: PLL 停止	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

必须在设定“11. 低功耗功能”的 PLL 等待控制寄存器后设定 PLLCR2 寄存器。

PLLEN 位 (PLL 停止控制位)

此位控制 PLL 的运行或者停止。

必须在通过 PLLEN 位将 PLL 设定为运行并且经过 PLL 时钟振荡稳定待机时间 (t_{PLLWT1} 或者 t_{PLLWT2}) 后, 开始使用 PLL 时钟。

从设定为运行后到振荡稳定为止, PLL 需要一定的时间; 从设定为停止后到振荡停止为止, PLL 也需要一定的时间。因此必须注意: 有关运行的开始和停止, 有以下限制:

- 要在将 PLL 设定为停止后再次设定为运行时, 停止期间必须至少为 5 个 PLL 时钟周期的时间。
- 必须在 PLL 时钟振荡稳定的状态下进行 PLL 的停止设定。
- 与是否选择作为系统时钟无关, 在将 PLL 设定为运行后转移到软件待机模式或者深度软件待机模式时, 必须在振荡稳定的状态下执行 WAIT 指令。
- 要在将 PLL 设定为停止后转移到软件待机模式或者深度软件待机模式时, 必须在将 PLL 设定为停止并且至少等待 2 个 PLL 时钟周期后执行 WAIT 指令。

当通过系统时钟控制寄存器 3 的时钟源选择位 (SCKCR3.CKSEL[2:0]) 选择了 PLL 时, 禁止给 PLLEN 位写“1” (PLL 停止运行)。

9.2.5 主时钟振荡器的控制寄存器（MOSCCR）

地址 0008 0032h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	MOSTP
0	0	0	0	0	0	0	1

复位后的值

位	符号	位名	功能	R/W
b0	MOSTP	主时钟振荡器停止位	0: 主时钟振荡器振荡 1: 主时钟振荡器停止振荡	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

必须在设定“11. 低功耗功能”的主时钟振荡器等待控制寄存器后设定 MOSCCR 寄存器。

MOSTP 位（主时钟振荡器停止位）

此位控制主时钟振荡器的振荡或者停止。

通过 MOSTP 位和主时钟振荡器的强制振荡控制寄存器的强制振荡位（MOFCR.MOFXIN）控制主时钟振荡器的振荡或者停止。能通过将 MOSTP 位设定为振荡器振荡或者将 MOFXIN 位设定为强制振荡，使主时钟振荡器振荡。当将 MOFXIN 位设定为强制振荡时，即使在深度软件待机模式中也继续振荡。

要改写 MOSTP 位或者 MOFCR.MOFXIN 位时，必须在写后通过读该位来确认已被改写，然后执行后续指令（请参照“5. I/O 寄存器”的“(2) 写 I/O 寄存器时的注意事项”）。

在通过连接晶体谐振器来提供主时钟的情况下，必须在通过 MOSTP 位或者 MOFCR.MOFXIN 位将主时钟振荡器设定为振荡并且经过主时钟振荡稳定待机时间（晶体）（tMAINOSCWT）后，开始使用主时钟。

在通过输入外部时钟来提供主时钟的情况下，必须在通过 MOSTP 位或者 MOFCR.MOFXIN 位将主时钟振荡器设定为振荡并且经过 EXTAL 外部时钟输入待机时间（tEXWT）后，开始使用主时钟。

从设定为振荡后到振荡稳定为止，主时钟振荡器需要一定的时间；从设定为停止后到振荡停止为止，主时钟振荡器也需要一定的时间。因此必须注意：有关振荡的开始和停止，有以下限制：

- 要在将主时钟振荡器设定为停止后再次设定为振荡时，停止期间必须至少为 5 个主时钟周期的时间。
- 必须在主时钟振荡器振荡稳定的状态下进行主时钟振荡器的停止设定。
- 与是否选择作为系统时钟无关，要在将主时钟振荡器设定为振荡后转移到软件待机模式或者深度软件待机模式时，必须在主时钟振荡器振荡稳定的状态下执行 WAIT 指令。
- 要在将主时钟振荡器设定为停止后转移到软件待机模式或者深度软件待机模式时，必须在将主时钟设定为停止并且至少等待 2 个主时钟周期后执行 WAIT 指令。

当通过系统时钟控制寄存器 3 的时钟源选择位（SCKCR3.CKSEL[2:0]）选择了主时钟振荡器或者通过 SCKCR3.CKSEL[2:0] 位选择了 PLL 时，禁止给 MOSTP 位写“1”（主时钟振荡器停止振荡）。

9.2.6 低速内部振荡器的控制寄存器 (LOCOCR)

地址 0008 0034h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	LCSTP
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	LCSTP	LOCO 停止位	0: LOCO 振荡 1: LOCO 停止振荡	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

LCSTP 位 (LOCO 停止位)

此位控制 LOCO 的振荡或者停止。

在通过 LCSTP 位将 LOCO 从停止改为振荡后使用 LOCO 时钟的情况下，必须在经过 LOCO 时钟振荡稳定待机时间 (tLOCOWT) 后，开始使用 LOCO 时钟。

从设定为振荡后到振荡稳定为止，LOCO 需要一定的时间；从设定为停止后到振荡停止为止，LOCO 也需要一定的时间。因此必须注意：有关振荡的开始和停止，有以下的限制：

- 要在将 LOCO 设定为停止后再次设定为振荡时，停止期间必须至少为 5 个 LOCO 时钟周期的时间。
- 必须在 LOCO 振荡稳定的状态下进行 LOCO 的停止设定。
- 与是否选择作为系统时钟无关，要在将 LOCO 设定为振荡后转移到软件待机模式或者深度软件待机模式时，必须在 LOCO 振荡稳定的状态下执行 WAIT 指令。
- 要在将 LOCO 设定为停止后转移到软件待机模式或者深度软件待机模式时，必须在将 LOCO 设定为停止并且至少等待 3 个 LOCO 时钟周期后执行 WAIT 指令。

当通过系统时钟控制寄存器 3 的时钟源选择位 (SCKCR3.CKSEL[2:0]) 选择了 LOCO 时，禁止给 LCSTP 位写“1” (LOCO 停止振荡)。

当通过振荡停止检测控制寄存器的振荡停止检测功能允许位 (OSTDCR.OSTDE) 将振荡停止检测功能设定为有效时，禁止给 LCSTP 位写“1” (LOCO 停止振荡)。

9.2.7 IWDT 专用内部振荡器的控制寄存器 (ILOCOCR)

地址 0008 0035h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	ILCSTP
0	0	0	0	0	0	0	1

复位后的值

位	符号	位名	功能	R/W
b0	ILCSTP	IWDT 专用内部振荡器停止位	0: IWDT 专用内部振荡器振荡 1: IWDT 专用内部振荡器停止振荡	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

当选项功能选择寄存器 0 的 IWDT 启动模式选择位 (OFS0.IWDTSTRT) 为“0” (IWDT 振荡) 时, ILOCOCR 寄存器的设定无效; 当 OFS0.IWDTSTRT 位为“1” (IWDT 停止振荡) 时, ILOCOCR 寄存器的设定有效。在 ILOCOCR 寄存器有效并且 ILCSTP 位变为“0” (IWDT 专用内部振荡器振荡) 后, 不能将 ILCSTP 位置“1” (IWDT 专用内部振荡器停止振荡)。

ILCSTP 位 (IWDT 专用内部振荡器停止位)

此位控制 IWDT 专用内部振荡器的振荡或者停止。

如果通过 ILCSTP 位将 IWDT 专用内部振荡器从停止改为振荡, 就在经过相当于 LOCO 振荡稳定待机时间 (t_{LOCOWT}) 的一定时间后开始给 LSI 内部提供时钟。如果要使用 IWDT 专用低速时钟, 就必须在经过此待机时间后开始使用。

要在将 IWDT 专用内部振荡器设定为振荡后转移到软件待机模式或者深度软件待机模式时, 必须在振荡稳定的状态下执行 WAIT 指令。

9.2.8 振荡停止检测控制寄存器 (OSTDCR)

地址 0008 0040h

b7	b6	b5	b4	b3	b2	b1	b0
OSTDE	—	—	—	—	—	—	OSTDIE
复位后的值	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	OSTDIE	振荡停止检测中断允许位	0: 禁止振荡停止检测中断并且没有将振荡停止检测通知 POE3 1: 允许振荡停止检测中断并且将振荡停止检测通知 POE3	R/W
b6-b1	—	保留位	读写值都为“0”。	R/W
b7	OSTDE	振荡停止检测功能允许位	0: 振荡停止检测功能无效 1: 振荡停止检测功能有效	R/W

OSTDIE 位 (振荡停止检测中断允许位)

必须在将 OSTDIE 位置“0”后清除振荡停止检测状态寄存器的振荡停止检测标志 (OSTDSR.OSTDF)。此后, 如果要再次将 OSTDIE 位置“1”, 就必须至少等待 2 个 PCLKB 周期。能通过读由 PCLKB 定义存取周期数的 I/O 寄存器, 至少确保 2 个 PCLKB 周期的等待时间。

OSTDE 位 (振荡停止检测功能允许位)

此位设定振荡停止检测功能的有效或者无效。

如果将 OSTDE 位置“1” (振荡停止检测功能有效), LOCO 停止位 (LOCOCR.LCSTP) 就变为“0”并且 LOCO 振荡。在振荡停止检测功能有效期间, 不能停止 LOCO 的振荡。即使给 LOCOCR.LCSTP 位写“1” (LOCO 停止振荡), 写操作也无效。

当振荡停止检测状态寄存器的振荡停止检测标志 (OSTDSR.OSTDF) 为“1” (检测到主时钟振荡停止振荡) 时, 即使给 OSTDE 位写“0”, 写操作也无效。

当 OSTDE 位为“1”时, 不能转移到软件待机模式和深度软件待机模式。如果要转移到软件待机模式和深度软件待机模式, 就必须在将 OSTDE 位置“0”后执行 WAIT 指令。

9.2.9 振荡停止检测状态寄存器 (OSTDSR)

地址 0008 0041h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	OSTDF
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	OSTDF	振荡停止检测标志	0: 未检测到主时钟停止振荡 1: 检测到主时钟停止振荡	R(W) (注1)
b7-b1	—	保留位	读取值为“0”，写操作无效	R

注 1. 只能写“0”。

OSTDF 标志 (振荡停止检测标志)

这是表示主时钟状态的标志。当 OSTDF 标志为“1”时，表示检测到主时钟停止振荡。

在检测到主时钟停止振荡后，即使主时钟重新开始振荡，OSTDF 标志也不变为“0”。通过在读“1”后写“0”，使 OSTDF 标志变为“0”。到 OSTDF 为“0”被反映为读取值为止，至少需要等待 3 个 ICLK 周期。如果在主时钟停止振荡的状态下将 OSTDF 标志置“0”，OSTDF 标志就在变为“0”后再次变为“1”。

当通过系统时钟控制寄存器 3 的时钟源选择位 (SCKCR3.CKSEL[2:0]) 选择了主时钟振荡器 (“010b”) 或者 PLL (“100b”) 时，不能将 OSTDF 标志置“0”。必须在将时钟源转换到不是主时钟振荡器和 PLL 后将 OSTDF 标志置“0”。

[为“1”的条件]

- 在 OSTDCR.OSTDE 位为“1” (振荡停止检测功能有效) 的状态下主时钟停止振荡时

[为“0”的条件]

- 在 SCKCR3.CKSEL[2:0] 位不是“010b”和“100b”的情况下读“1”后写“0”时

9.2.10 主时钟振荡器的强制振荡控制寄存器 (MOFCR)

地址 0008 C293h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	MOFXIN
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	MOFXIN	主时钟振荡器强制振荡位	0: 不通过此位进行控制 1: 强制主时钟振荡器振荡	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

MOFXIN 位 (主时钟振荡器强制振荡位)

此位控制主时钟振荡器的强制振荡。

要改写 MOSCCR.MOSTP 位或者 MOFXIN 位时，必须在写后通过读该位来确认已被改写，然后执行后续的指令 (请参照“5. I/O 寄存器”的“(2) 写 I/O 寄存器时的注意事项”)。

9.3 主时钟振荡器

给主时钟振荡器提供时钟的方法有连接谐振器和输入外部时钟的方法。

9.3.1 连接谐振器的方法

连接谐振器时的连接例子如图 9.2 所示。

必须根据需要插入阻尼电阻 (R_d)。电阻值因谐振器和振荡驱动能力而不同，因此必须设定为谐振器厂商的推荐值。如果谐振器厂商有外加反馈电阻 (R_f) 的指示，就必须根据该指示，在 EXTAL 和 XTAL 之间插入 R_f 。

在连接谐振器提供时钟的情况下，连接的谐振器必须在表 9.1 的主时钟振荡器的谐振器频率的范围内。

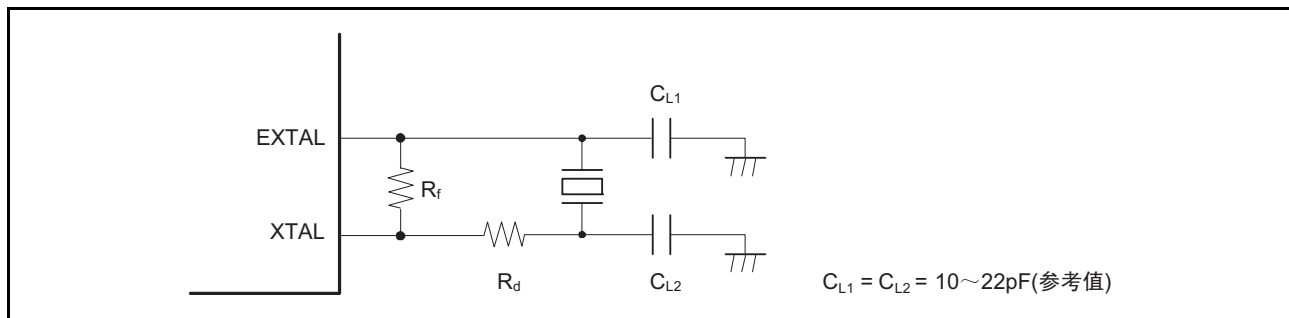


图 9.2 晶体谐振器的连接例子

表 9.3 阻尼电阻 (参考值)

频率 (MHz)	4	8	12	16
R_d (Ω)	380	200	0	0

晶体谐振器的等效电路如图 9.3 所示，必须使用表 9.4 所示特性的晶体谐振器。

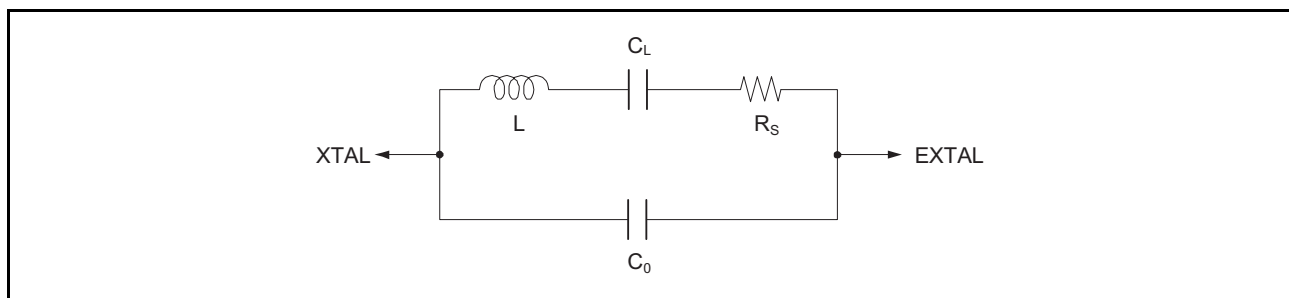


图 9.3 晶体谐振器的等效电路

表 9.4 晶体谐振器的特性 (参考值)

频率 (MHz)	4	8	12	16
R_S max (Ω)	160	80	60	40
C_0 max (pF)	7			

9.3.2 输入外部时钟的方法

输入外部时钟的连接例子如图 9.4 所示。在将 XTAL 引脚置为开路时，寄生电容必须小于等于 5pF。

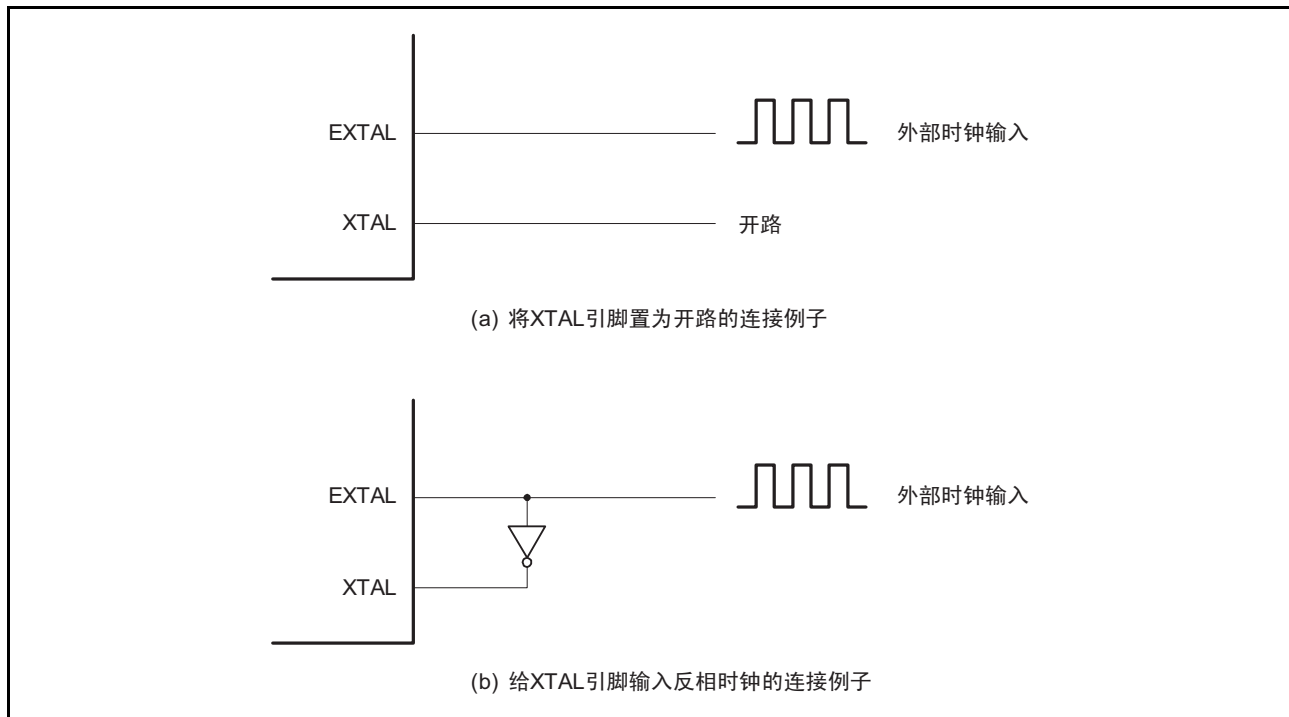


图 9.4 外部时钟的连接例子

9.3.3 有关外部时钟输入的注意事项

只能在主时钟振荡器停止振荡时更改外部时钟的输入频率。在主时钟振荡器停止位 (MOSCCR.MOSTP) 已被置“0” (主时钟振荡器振荡) 或者主时钟振荡器强制振荡位 (MOFCR.MOFXIN) 已被置“1” (强制主时钟振荡器振荡) 的期间, 不能更改外部时钟的输入频率。

9.4 振荡停止检测功能

9.4.1 振荡停止检测和检测后的运行

振荡停止检测功能检测主时钟振荡器的停止，并且作为系统时钟的时钟源，代替主时钟和 PLL 时钟提供低速内部振荡器输出的 LOCO 时钟。

能在检测到振荡停止时产生振荡停止检测中断请求，也能在检测到振荡停止时将 MTU3 的输出强制置为高阻抗。详细内容请参照“21. 多功能定时器脉冲单元 3 (MTU3)”和“22. 端口输出允许 3 (POE3)”。

如果输入时钟因主时钟振荡器发生异常等而在一定期间内变为“0”或者“1”（参照“36.6 振荡停止检测时序”），RX63T 群就检测到主时钟的停止振荡。

一旦检测到振荡停止，时钟源选择位 (SCKCR3.CKSEL[2:0]) 选择的主时钟和 PLL 时钟就通过各自前段的选择器转换到 LOCO 时钟。因此，如果在选择主时钟或者 PLL 时钟作为系统时钟的时钟源的状态下检测到振荡停止，CKSEL[2:0] 位的设定值就不变而将系统时钟的时钟源转换到 LOCO 时钟。

振荡停止检测标志 (OSTDSR.OSTDF) 控制主时钟和 LOCO 时钟的转换以及 PLL 时钟和 LOCO 时钟的转换。如果将 OSTDF 标志置“1”，就转换到 LOCO 时钟；如果将 OSTDF 标志置“0”，就恢复主时钟或者 PLL 时钟。但是，在通过 CKSEL[2:0] 位选择了主时钟或者 PLL 时钟的情况下，不能将 OSTDF 标志置“0”。如果在检测到振荡停止后将时钟源恢复为主时钟或者 PLL 时钟，就必须在将 CKSEL[2:0] 位的设定更改为不是主时钟和 PLL 时钟后将 OSTDF 标志置“0”。此后，必须确认 OSTDF 标志不是“1”，并且在经过规定的振荡稳定时间后将 CKSEL[2:0] 位的设定更改为主时钟或者 PLL 时钟。

在解除复位后，主时钟振荡器停止振荡，振荡停止检测功能无效。要将振荡停止检测功能置为有效时，必须使主时钟振荡器振荡，并且在经过规定的振荡稳定时间后写振荡停止检测功能允许位 (OSTDCR.OSTDE)。

因为振荡停止检测功能是防止由外部原因导致主时钟停止的功能，所以要通过软件使主时钟振荡器停止振荡或者转移到软件待机模式和深度软件待机模式时，必须预先将振荡停止检测功能置为无效。

通过振荡停止检测而转换到 LOCO 时钟的时钟是作为系统时钟源的主时钟、PLL 时钟、CAC 主时钟 (CACMCLK) 以及 CAC 外围模块时钟 (CACPLK)。

LOCO 时钟运行时的系统时钟 (ICLK) 的频率取决于 LOCO 振荡频率和系统时钟选择位 (SCKCR.ICK[3:0]) 的分频比的设定。

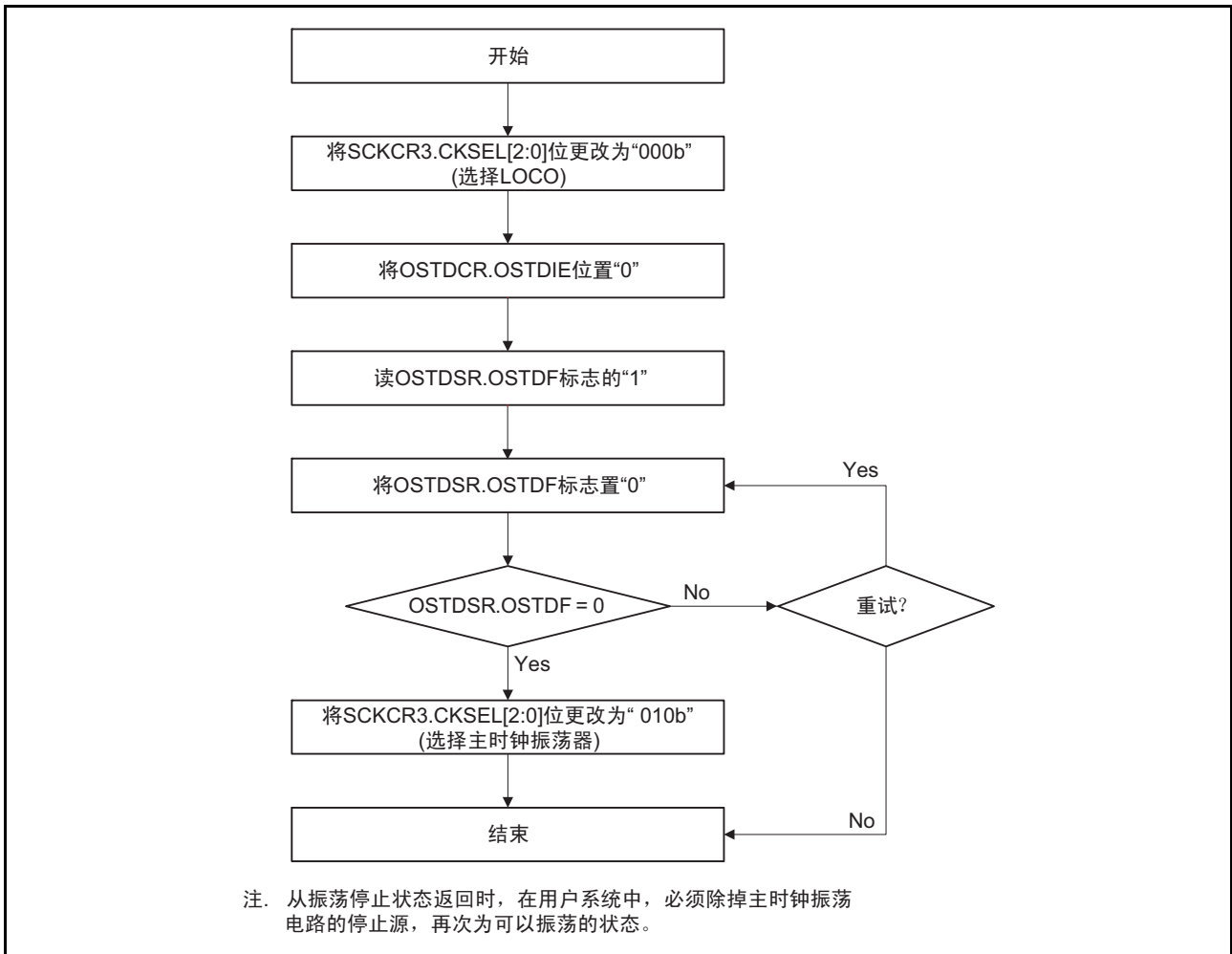


图 9.5 从振荡停止检测恢复的流程图例子

9.4.2 振荡停止检测中断

当振荡停止检测中断允许位 (OSTDCR.OSTDIE) 为 “1” (允许振荡停止检测中断) 时, 如果振荡停止检测标志 (OSTDSR.OSTDF) 变为 “1”, 就产生振荡停止检测中断 (OSTDI) 请求。此时, 将主时钟振荡器停止振荡通知到端口输出允许 3 (POE3)。在 POE3 接受振荡停止的通知后将输入电平控制 / 状态寄存器 3 的 OSTST 高阻抗标志 (ICSR3.OSTSTF) 置 “1”。必须注意: 在检测到振荡停止后的 10 个 PCLK 周期期间, 不能写 ICSR3.OSTSTF 标志。必须在将振荡停止检测中断允许位 (OSTDCR.OSTDIE) 置 “0” 后清除 OSTDSR.OSTDF 标志。此后, 要再次将 OSTDCR.OSTDIE 位置 “1” 时, 必须在至少等待 2 个 PCLKB 周期后进行设定。能通过读由 PCLKB 定义存取周期数的 I/O 寄存器, 至少确保 2 个 PCLKB 周期的等待时间。

振荡停止检测中断是非屏蔽中断。因为在解除复位后的初始状态下为 “禁止非屏蔽中断”, 所以在使用振荡停止检测中断时, 必须通过软件将非屏蔽中断置为有效。详细内容请参照 “14. 中断控制器 (ICUb)”。

9.5 PLL 电路

PLL 电路具有递增振荡器频率的功能。

9.6 内部时钟

作为内部时钟的时钟源，有主时钟、LOCO 时钟、PLL 时钟、IWDT 专用低速时钟、JTAG 外部时钟，由这些时钟生成如下所示的内部时钟。

1. CPU、DMAC、DTC、内部ROM和内部RAM的运行时钟：系统时钟（ICLK）
2. 定时器模块的运行时钟：定时器模块时钟（PCLKA）
3. 外围模块的运行时钟：外围模块时钟（PCLKB）
4. S12AD的运行时钟：S12AD时钟（PCLKD）
5. FlashIF的运行时钟：FlashIF时钟（FCLK）
6. CAC模块的运行时钟：CAC时钟（CACCLK）
7. IWDT模块的运行时钟：IWDT专用低速时钟（IWDTCLK）
8. JTAG模块的运行时钟：JTAG时钟（JTAGTCK）

通过选择分频比的 SCKCR.FCK[3:0] 位、SCKCR.ICK[3:0] 位、SCKCR.PCKA[3:0] 位、SCKCR.PCKB[3:0] 位、SCKCR.PCKD[3:0] 位、选择时钟源的 SCKCR3.CKSEL[2:0] 位以及选择 PLL 电路频率的 PLLCR.STC[5:0] 位和 PLLCR.PLIDIV[1:0] 位的组合，设定内部时钟的频率。在改写各位后，以更改后的频率运行。

9.6.1 系统时钟

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

通过 SCKCR.ICK[3:0] 位、SCKCR3.CKSEL[2:0] 位、PLLCR.STC[5:0] 位和 PLLCR.PLIDIV[1:0] 位设定 ICLK 的频率。

9.6.2 定时器模块时钟

定时器模块时钟（PCLKA）是定时器模块的运行时钟。

通过 SCKCR.PCKA[3:0] 位、SCKCR3.CKSEL[2:0] 位、PLLCR.STC[5:0] 位和 PLLCR.PLIDIV 位设定 PCLKA 的频率。

9.6.3 外围模块时钟

外围模块时钟（PCLKB）是外围模块的运行时钟。

通过 SCKCR.PCKB[3:0] 位、SCKCR3.CKSEL[2:0] 位、PLLCR.STC[5:0] 位和 PLLCR.PLIDIV[1:0] 位设定 PCLKB 的频率。

9.6.4 S12AD 时钟

S12AD 时钟（PCLKD）是 S12AD 的运行时钟。

通过 SCKCR.PCKD[3:0] 位、SCKCR3.CKSEL[2:0] 位、PLLCR.STC[5:0] 位和 PLLCR.PLIDIV[1:0] 位设定 PCLKD 的频率。

9.6.5 FlashIF 时钟

FlashIF 时钟（FCLK）是 FlashIF 的运行时钟。用于 ROM、E2 数据闪存的编程 / 擦除以及 E2 数据闪存的读操作。

通过 SCKCR.FCK[3:0] 位、SCKCR3.CKSEL[2:0] 位、PLLCR.STC[5:0] 位和 PLLCR.PLIDIV[1:0] 位设定 FCLK 的频率。

9.6.6 CAC 时钟

CAC 外围模块时钟 (CACCLK) 是 CAC 模块的运行时钟。

CACCLK 是通过主时钟振荡器、PLL 电路、低速时钟振荡器或者 IWDTC 专用低速时钟振荡器生成的时钟。

9.6.7 IWDTC 专用低速时钟

IWDTC 专用低速时钟 (IWDTCCLK) 是 IWDTC 模块的运行时钟。

IWDTCCLK 是 IWDTC 专用内部振荡器通过内部振荡生成的时钟。

9.6.8 JTAG 时钟

JTAG 时钟 (JTAGTCK) 是 JTAG 的运行时钟。

JTAGTCK 是 JTAG 外部时钟 (TCK) 生成的时钟。

9.7 连接谐振器时的引脚设定

(1) 主时钟

必须将主时钟振荡器停止位 (MOSCCR.MOSTP 位) 置“0” (振荡) 或者将主时钟振荡器强制振荡位 (MOFCR.MOFXIN) 置“1” (强制振荡)。

9.8 使用时的注意事项

9.8.1 有关时钟发生电路的注意事项

1. 通过 SCKCR 寄存器选择提供给各模块的系统时钟 (ICLK)、定时器模块时钟 (PCLKA)、外围模块时钟 (PCLKB)、S12AD 时钟 (PCLKD) 和 FlashIF 时钟 (FCLK) 的频率。各频率的设定注意事项如下所示：
各频率必须设定在电特性的 AC 时序的时钟周期时间 t_{cy} 的运行保证范围内。
必须将频率设定在表 9.1 的频率范围内。
外围模块基本上以 PCLKB 为基准运行。因此必须注意：在更改频率的前后定时器和 SCI 等的运行速度会发生变化。
2. 为了在更改时钟频率后确切地执行下一个处理，必须在频率更改的写操作后读同一个寄存器，然后执行下一个处理。

9.8.2 有关谐振器的注意事项

因为谐振器的各种特性与用户的电路板设计密接相关，所以请用户参照本章介绍的谐振器的连接例子，在给予充分的评估后使用。谐振器的电路常数因谐振器和安装电路的寄生电容而不同，因此必须和谐振器厂商仔细商谈后决定。外加在振荡引脚的电压不能超过最大额定值。

9.8.3 设计电路板时的注意事项

在使用谐振器时，必须尽量将谐振器和电容器排列在谐振器连接引脚的附近。如图 9.6 所示，在振荡电路附近不能通过信号线，否则就可能因电感而不能正常振荡。

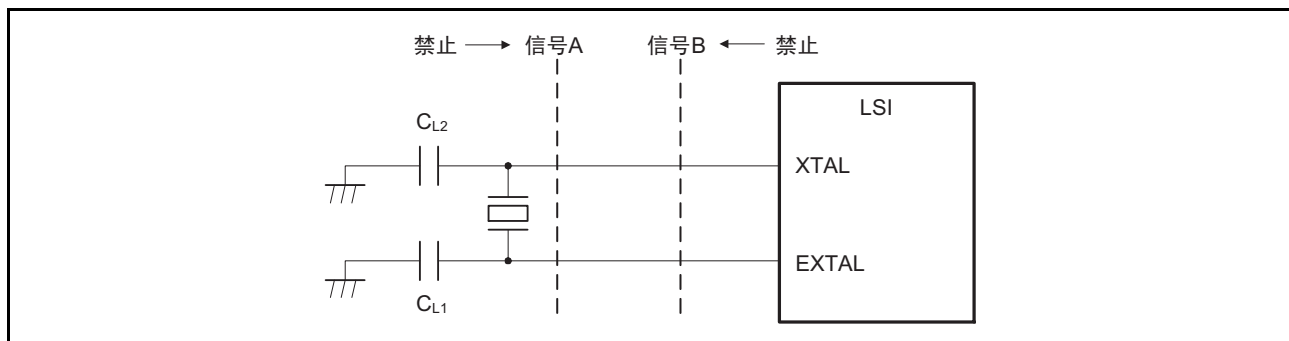


图 9.6 有关振荡电路部的电路板设计注意事项（主时钟振荡器的情况）

10. 时钟频率精度测量电路 (CAC)

时钟频率精度测量电路 (CAC) 根据从 LSI 外部输入的基准信号和其他时钟源来监视时钟频率，并且在超出设定范围时发生中断。

10.1 概要

CAC 的规格和框图分别如表 10.1 和图 10.1 所示。

表 10.1 CAC 的规格

项目	内容
时钟频率测量	能测量以下时钟的频率 • 主时钟振荡器的输出时钟 (主时钟) • IWDTC 专用低速时钟 (IWDTCCLK) • 外围模块时钟 (PCLKB)
选择功能	数字滤波器功能
中断源	• 测量结束中断 • 频率错误中断 • 上溢
低功耗功能	能设定为模块停止状态

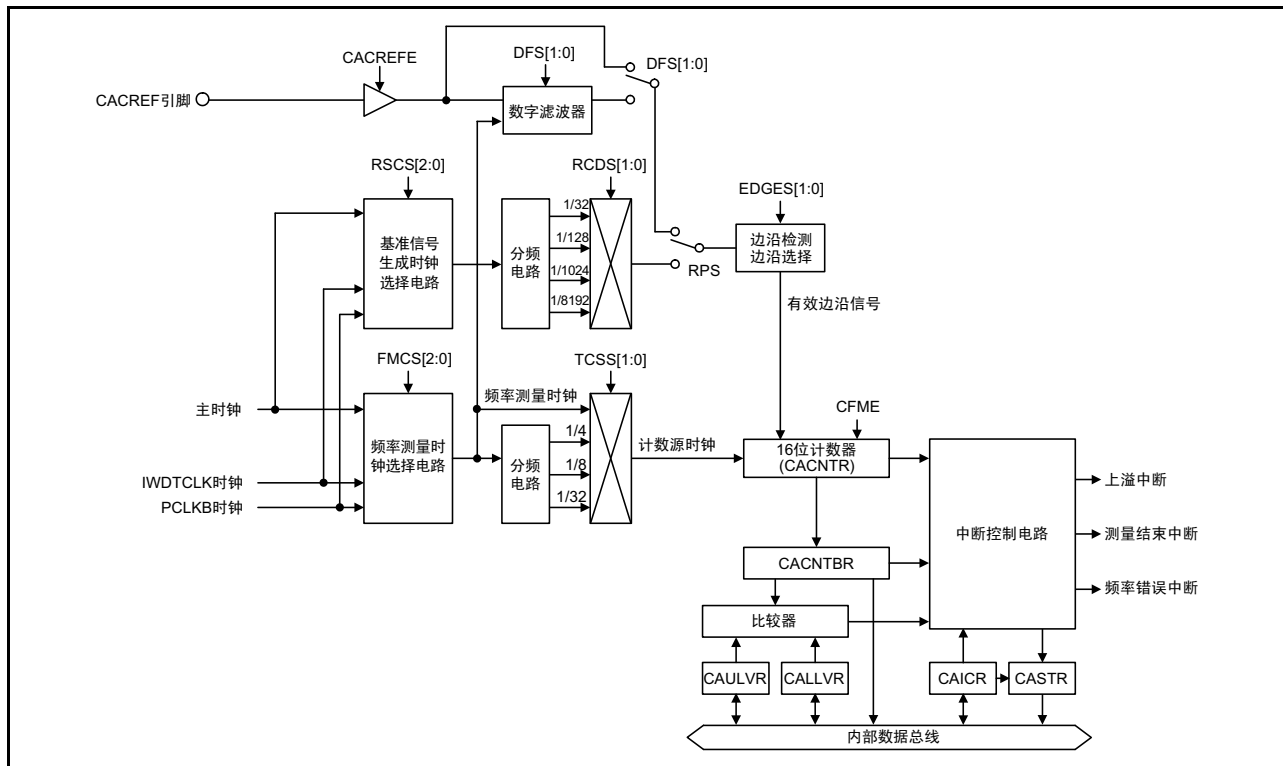


图 10.1 CAC 的框图

CAC 的输入 / 输出引脚如表 10.2 所示。

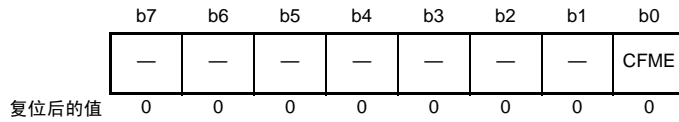
表 10.2 CAC 的输入 / 输出引脚

引脚名	输入 / 输出	功能
CACREF	输入	时钟频率精度测量电路的输入引脚

10.2 寄存器说明

10.2.1 CAC 控制寄存器 0 (CACR0)

地址 0008 B000h



位	符号	位名	功能	R/W
b0	CFME	时钟频率测量允许位	0: 时钟频率测量无效 1: 时钟频率测量有效	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

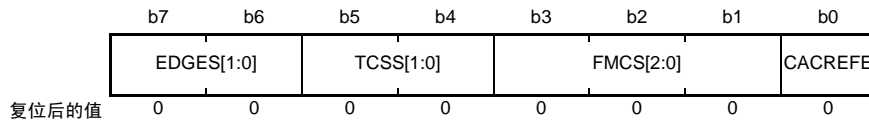
CFME 位 (时钟频率测量允许位)

当 CFME 位为“1”时，时钟频率测量有效。

在改写 CFME 位后，其结果反映到内部电路需要一些时间。如果在前一个值还未反映到内部电路的状态下改写 CFME 位，就忽视此改写。能通过读此位确认改写是否已被反映。

10.2.2 CAC 控制寄存器 1 (CACR1)

地址 0008 B001h



位	符号	位名	功能	R/W																											
b0	CACREFE	CACREF 引脚输入允许位	0: CACREF 引脚输入无效 1: CACREF 引脚输入有效	R/W																											
b3-b1	FMCS[2:0]	频率测量时钟选择位	<table style="width: 100%; border: none;"> <tr> <td style="width: 10%; text-align: right;">b3</td> <td style="width: 10%; text-align: left;">b1</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0: 主时钟振荡器的输出时钟</td> </tr> <tr> <td>0</td> <td>0</td> <td>1: 不能设定</td> </tr> <tr> <td>0</td> <td>1</td> <td>0: 不能设定</td> </tr> <tr> <td>0</td> <td>1</td> <td>1: 不能设定</td> </tr> <tr> <td>1</td> <td>0</td> <td>0: IWDG 专用低速时钟</td> </tr> <tr> <td>1</td> <td>0</td> <td>1: 外围模块时钟 (PCLKB)</td> </tr> <tr> <td>1</td> <td>1</td> <td>0: 不能设定</td> </tr> <tr> <td>1</td> <td>1</td> <td>1: 不能设定</td> </tr> </table>	b3	b1		0	0	0: 主时钟振荡器的输出时钟	0	0	1: 不能设定	0	1	0: 不能设定	0	1	1: 不能设定	1	0	0: IWDG 专用低速时钟	1	0	1: 外围模块时钟 (PCLKB)	1	1	0: 不能设定	1	1	1: 不能设定	R/W
b3	b1																														
0	0	0: 主时钟振荡器的输出时钟																													
0	0	1: 不能设定																													
0	1	0: 不能设定																													
0	1	1: 不能设定																													
1	0	0: IWDG 专用低速时钟																													
1	0	1: 外围模块时钟 (PCLKB)																													
1	1	0: 不能设定																													
1	1	1: 不能设定																													
b5-b4	TCSS[1:0]	定时器计数时钟源选择位	<table style="width: 100%; border: none;"> <tr> <td style="width: 10%; text-align: right;">b5</td> <td style="width: 10%; text-align: left;">b4</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0: 无分频时钟</td> </tr> <tr> <td>0</td> <td>1</td> <td>4 分频时钟</td> </tr> <tr> <td>1</td> <td>0</td> <td>8 分频时钟</td> </tr> <tr> <td>1</td> <td>1</td> <td>32 分频时钟</td> </tr> </table>	b5	b4		0	0	0: 无分频时钟	0	1	4 分频时钟	1	0	8 分频时钟	1	1	32 分频时钟	R/W												
b5	b4																														
0	0	0: 无分频时钟																													
0	1	4 分频时钟																													
1	0	8 分频时钟																													
1	1	32 分频时钟																													
b7-b6	EDGES[1:0]	有效边沿选择位	<table style="width: 100%; border: none;"> <tr> <td style="width: 10%; text-align: right;">b7</td> <td style="width: 10%; text-align: left;">b6</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0: 上升沿</td> </tr> <tr> <td>0</td> <td>1</td> <td>1: 下降沿</td> </tr> <tr> <td>1</td> <td>0</td> <td>0: 双边沿 (上升沿 / 下降沿)</td> </tr> <tr> <td>1</td> <td>1</td> <td>1: 不能设定</td> </tr> </table>	b7	b6		0	0	0: 上升沿	0	1	1: 下降沿	1	0	0: 双边沿 (上升沿 / 下降沿)	1	1	1: 不能设定	R/W												
b7	b6																														
0	0	0: 上升沿																													
0	1	1: 下降沿																													
1	0	0: 双边沿 (上升沿 / 下降沿)																													
1	1	1: 不能设定																													

CACREFE 位 (CACREF 引脚输入允许位)

当 CACREFE 位为“1”时，CACREF 引脚的输入有效。

FMCS[2:0] 位 (频率测量时钟选择位)

能通过设定 FMCS[2:0] 位选择要测量频率的时钟。

TCSS[1:0] 位 (定时器计数时钟源选择位)

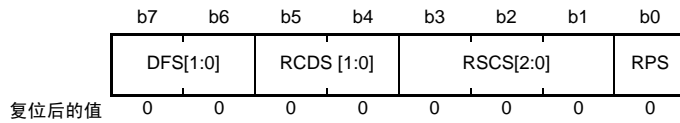
能通过设定 TCSS[1:0] 位选择时钟频率精度测量电路的计数时钟源。

EDGES[1:0] 位 (有效边沿选择位)

能通过设定 EDGES[1:0] 位选择基准信号的有效边沿。

10.2.3 CAC 控制寄存器 2 (CACR2)

地址 0008 B002h



位	符号	位名	功能	R/W
b0	RPS	基准信号选择位	0: CACREF 引脚输入 1: 内部生成信号	R/W
b3-b1	RSCS[2:0]	基准信号生成时钟选择位	b3 b1 0 0 0: 主时钟振荡器的输出时钟 0 0 1: 不能设定 0 1 0: 不能设定 0 1 1: 不能设定 1 0 0: IWDT 专用低速时钟 1 0 1: 外围模块时钟 (PCLKB) 1 1 0: 不能设定 1 1 1: 不能设定	R/W
b5-b4	RCDS[1:0]	基准信号生成时钟的分频比选择位	b5 b4 0 0: 32 分频时钟 0 1: 128 分频时钟 1 0: 1024 分频时钟 1 1: 8192 分频时钟	R/W
b7-b6	DFS[1:0]	数字滤波器功能选择位	b7 b6 0 0: 数字滤波器功能无效 0 1: 频率测量时钟 1 0: 频率测量时钟的 4 分频时钟 1 1: 频率测量时钟的 16 分频时钟	R/W

RPS 位 (基准信号选择位)

能通过设定 RPS 位选择是将 CACREF 引脚输入用作基准信号还是将内部生成信号用作基准信号。

RSCS[2:0] 位 (基准信号生成时钟选择位)

能通过设定 RSCS[2:0] 位选择生成基准信号的时钟源。

RCDS[1:0] 位 (基准信号生成时钟的分频比选择位)

能通过设定 RCDS[1:0] 位选择基准信号生成时钟的分频比。

DFS[1:0] 位 (数字滤波器功能选择位)

能通过设定 DFS[1:0] 位选择数字滤波器的有效或者无效以及采样时钟。

10.2.4 CAC 中断控制寄存器 (CAICR)

地址 0008 B003h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	FERRIE	频率错误中断允许位	0: 频率错误中断无效 1: 频率错误中断有效	R/W
b1	MENDIE	测量结束中断允许位	0: 测量结束中断无效 1: 测量结束中断有效	R/W
b2	OVFIE	上溢中断允许位	0: 上溢中断无效 1: 上溢中断有效	R/W
b3	—	保留位	读写值都为“0”。	R/W
b4	FERRFCL	FERRF 标志清除位	0: 不影响运行 1: 清除 FERRF 标志 读取值为“0”。	R/W
b5	MENDFCL	MENDF 标志清除位	0: 不影响运行 1: 清除 MENDF 标志 读取值为“0”。	R/W
b6	OVFFCL	OVFF 标志清除位	0: 不影响运行 1: 清除 OVFF 标志 读取值为“0”。	R/W
b7	—	保留位	读写值都为“0”。	R/W

FERRIE 位 (频率错误中断允许位)

当 FERRIE 位为“1”时，允许频率错误中断。

MENDIE 位 (测量结束中断允许位)

当 MENDIE 位为“1”时，允许测量结束中断。

OVFIE 位 (上溢中断允许位)

当 OVFIE 位为“1”时，允许上溢中断。

FERRFCL 位 (FERRF 标志清除位)

如果将 FERRFCL 位置“1”，就清除 FERRF 标志。

MENDFCL 位 (MENDF 标志清除位)

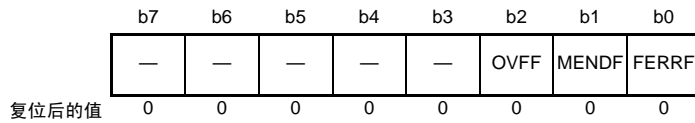
如果将 MENDFCL 位置“1”，就清除 MENDF 标志。

OVFFCL 位 (OVFF 标志清除位)

如果将 OVFFCL 位置“1”，就清除 OVFF 标志。

10.2.5 CAC 状态寄存器 (CASTR)

地址 0008 B004h



位	符号	位名	功能	R/W
b0	FERRF	频率错误标志	0: 时钟频率在设定值范围 1: 时钟频率不在设定值范围 (频率错误)	R
b1	MENDF	测量结束标志	0: 正在测量 1: 测量结束	R
b2	OVFF	上溢标志	0: 计数器没有发生上溢 1: 计数器发生上溢	R
b7-b3	—	保留位	读写值都为“0”。	R/W

FERRF 标志 (频率错误标志)

当时钟频率不在设定值范围 (频率错误) 时 FERRF 标志变为“1”。

[为“1”的条件]

- 当时钟频率不在设定值范围时

[为“0”的条件]

- 给 FERRFCL 位写“1”时

MENDF 标志 (测量结束标志)

在测量结束时 MENDF 标志变为“1”。

[为“1”的条件]

- 在测量结束时

[为“0”的条件]

- 给 MENDFCL 位写“1”时

OVFF 标志 (上溢标志)

在计数器发生上溢时 OVFF 标志变为“1”。

[为“1”的条件]

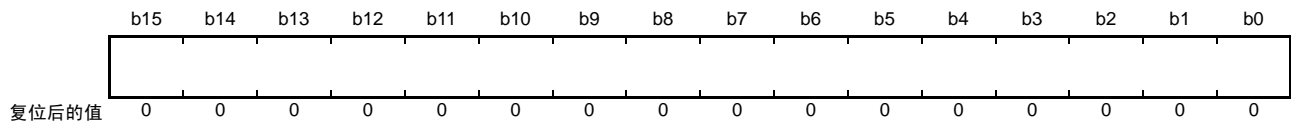
- 在计数器发生上溢时

[为“0”的条件]

- 给 OVFFCL 位写“1”时

10.2.6 CAC 上限值设定寄存器 (CAULVR)

地址 0008 B006h



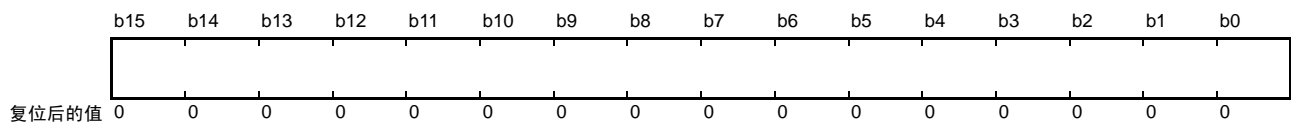
CAULVR 寄存器是保存频率上限值的 16 位可读写寄存器。

在 CACR0.CFME 位为“0”时设定此次寄存器。

由于数字滤波器、边沿检测电路、以及 CACREF 引脚输入的相位差，可能会使保存到 CACNTBR 寄存器的计数器值发生偏差，所以必须设定有宽裕的值。

10.2.7 CAC 下限值设定寄存器 (CALLVR)

地址 0008 B008h



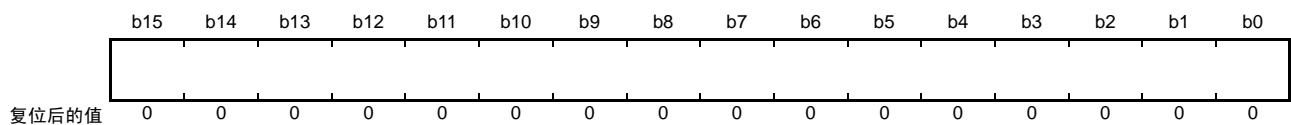
CALLVR 寄存器是保存频率下限值的 16 位可读写寄存器。

在 CACR0.CFME 位为“0”时设定此次寄存器。

由于数字滤波器、边沿检测电路、以及 CACREF 引脚输入的相位差，可能会使保存到 CACNTBR 寄存器的计数器值发生偏差，所以必须设定有宽裕的值。

10.2.8 CAC 计数器的缓冲寄存器 (CACNTBR)

地址 0008 B00Ah



CACNTBR 寄存器是 16 位只读专用寄存器，保存输入基准信号有效边沿时的计数器值。

10.3 运行说明

10.3.1 以 CACREF 引脚输入为基准测量时钟频率

以 CACREF 引脚输入为基准的时钟频率精度测量电路的运行例子如图 10.2 所示。
在测量时钟频率时，时钟频率精度测量电路如下运行。

1. 在将 CACR2.RPS 位置“0”、CACR1.CACREFE 位置“1”的状态下给 CACR0.CFME 位写“1”时，以 CACREF 引脚输入为基准的时钟频率测量有效。
2. 在给 CFME 位写“1”后，如果从 CACREFE 引脚输入 CACR1.EDGES[1:0] 位选择的有效边沿，就开始定时器的递增计数。
3. 在输入下一个有效边沿时，将寄存器值保存到 CACNTBR 寄存器，并且与 CAULVR 寄存器和 CALLVR 寄存器进行比较。当 CACNTBR 寄存器 \leq CAULVR 寄存器并且 CACNTBR 寄存器 \geq CALLVR 寄存器时，时钟频率为正常值，因此只有 CASTR.MENDF 标志被置“1”。另外，如果给 CAICR.MENDIE 位设定“1”，就发生测量结束中断。
4. 在输入下一个有效边沿时，将寄存器值保存到 CACNTBR 寄存器，并且与 CAULVR 寄存器和 CALLVR 寄存器进行比较。当 CACNTBR 寄存器 $>$ CAULVR 寄存器时，时钟频率就为异常值，因此 CASTR.FERRF 标志被置“1”。另外，如果给 CAICR.FERRIE 位设定“1”，就发生频率错误中断，而且 CASTR.MENDF 标志也被置“1”。如果给 CAICR.MENDIE 位设定“1”，就发生测量结束中断。
5. 在输入下一个有效边沿时，将寄存器值保存到 CACNTBR 寄存器，并且与 CAULVR 寄存器和 CALLVR 寄存器进行比较。当 CACNTBR 寄存器 $<$ CALLVR 寄存器时，时钟频率为异常值，因此 CASTR.FERRF 标志被置“1”。另外，如果给 CAICR.FERRIE 位设定“1”，就发生频率错误中断，而且 CASTR.MENDF 标志也被置“1”。如果给 CAICR.MENDIE 位设定“1”，就发生测量结束中断。
6. 在 CACR0.CFME 位为“1”的期间，每当输入有效边沿时，就将计数器值保存到 CACNTBR 寄存器，并且与 CAULVR 寄存器和 CALLVR 寄存器进行比较。如果给 CACR0.CFME 位写“0”，就清除计数器并且停止递增计数。

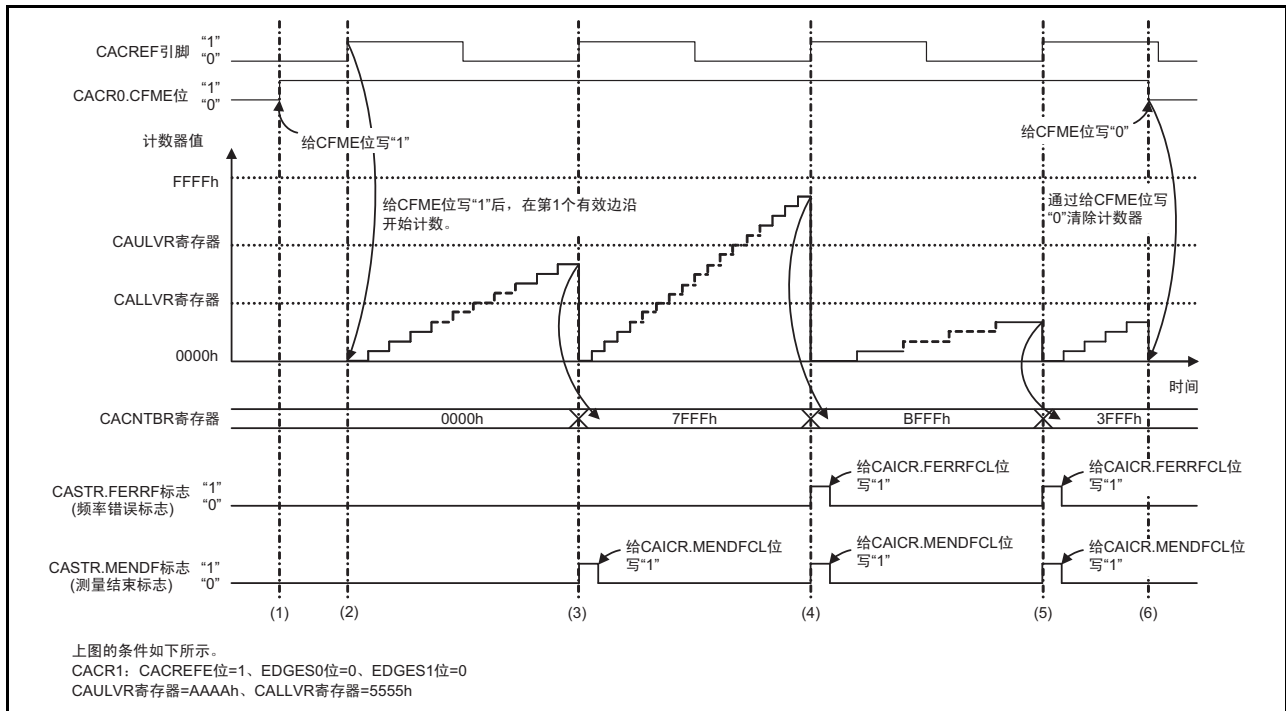


图 10.2 以 CACREF 引脚为基准的时钟频率精度测量电路的运行例子

10.3.2 以其他时钟源为基准测量时钟频率

以其他时钟源为基准的时钟频率精度测量电路的运行例子如图 10.3 所示。
在测量时钟频率时，时钟频率精度测量电路如下运行。

1. 在将 CACR2.RPS 位置“1”状态下给 CACR0.CFME 位写“1”时，以其他时钟源为基准的时钟频率测量有效。
2. 在给 CFME 位写“1”后，如果根据 CACR2.RSCS[2:0] 位选择的时钟源输入 CACR1.EDGES[1:0] 位选择的有效边沿，就开始定时器的递增计数。
3. 在输入下一个有效边沿时，将寄存器值保存到 CACNTBR 寄存器，并且与 CAULVR 寄存器和 CALLVR 寄存器进行比较。当 CACNTBR 寄存器 \leq CAULVR 寄存器并且 CACNTBR 寄存器 \geq CALLVR 寄存器时，时钟频率为正常值，因此只有 CASTR.MENDF 标志被置“1”。另外，如果给 CAICR.MENDIE 位设定“1”，就发生测量结束中断。
4. 在输入下一个有效边沿时，将寄存器值保存到 CACNTBR 寄存器，并且与 CAULVR 寄存器和 CALLVR 寄存器进行比较。当 CACNTBR 寄存器 $>$ CAULVR 寄存器时，时钟频率为异常值，因此 CASTR.FERRF 标志被置“1”。另外，如果给 CAICR.FERRIE 位设定“1”，就发生频率错误中断，而且 CASTR.MENDF 标志也被置“1”。如果给 CAICR.MENDIE 位设定“1”，就发生测量结束中断。
5. 在输入下一个有效边沿时，将寄存器值保存到 CACNTBR 寄存器，并且与 CAULVR 寄存器和 CALLVR 寄存器进行比较。当 CACNTBR 寄存器 $<$ CALLVR 寄存器时，时钟频率为异常值，因此 CASTR.FERRF 标志被置“1”。另外，如果给 CAICR.FERRIE 位设定“1”，就发生频率错误中断，而且 CASTR.MENDF 标志也被置“1”。如果给 CAICR.MENDIE 位设定“1”，就发生测量结束中断。
6. 在 CACR0.CFME 位为“1”的期间，每当输入有效边沿时，就将计数器值保存到 CACNTBR 寄存器，并且与 CAULVR 寄存器和 CALLVR 寄存器进行比较。如果给 CACR0.CFME 位写“0”，就清除计数器并且停止递增计数。

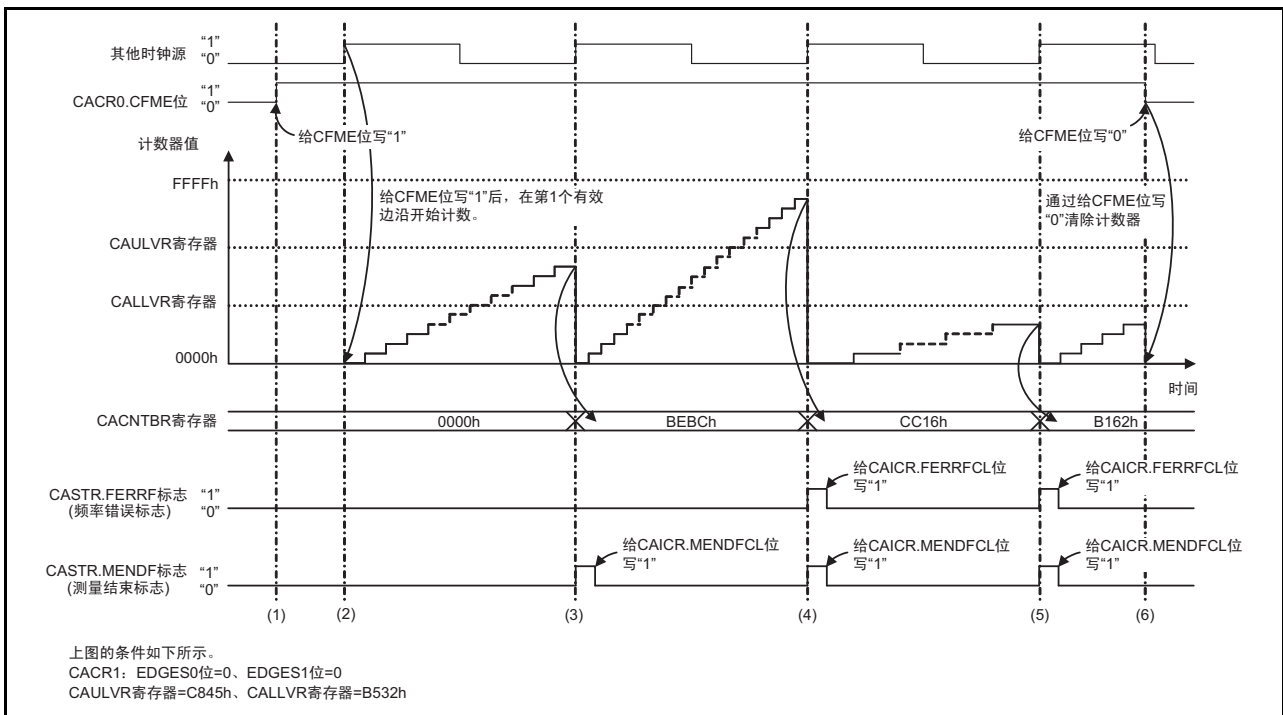


图 10.3 以其他时钟源为基准的时钟频率精度测量电路的运行例子

10.3.3 CACREF 引脚的数字滤波器功能

CACREF 引脚具有数字滤波器功能。数字滤波器功能在跟据设定的采样周期进行 3 次采样的引脚的电平相同时，将此相同的电平传达到内部，直到重新进行 3 次采样的引脚的电平相同为止。

数字滤波器功能可以设定数字滤波器功能的有效或者无效以及采样时钟。

由于数字滤波器和 CACREF 引脚输入信号的相位差，保存到 CACNTBR 寄存器的计数值有可能产生最大 1 个采样时钟周期的误差。

在选择分频时钟作为计数源时钟的情况下，能通过以下计算式表达计数值误差。

计数值误差 = (1 个计数源时钟周期) / (1 个采样时钟周期)

10.4 中断请求

时钟频率精度测量电路生成的中断请求有 3 种，分别是频率错误中断、测量结束中断以及上溢中断。如果产生各中断源，各状态标志就被置“1”。时钟频率精度测量电路的中断请求如表 10.3 所示。

表 10.3 时钟频率精度测量电路的中断请求

中断请求	中断允许位	状态标志	中断源
频率错误中断	CAICR.FERRIE	CASTR.FERRF	将 CACNTBR 寄存器与 CAULVR 寄存器和 CALLVR 寄存器比较，当比较结果为 CACNTBR 寄存器 > CAULVR 寄存器或者 CACNTBR 寄存器 < CALLVR 寄存器时
测量结束中断	CAICR.MENDIE	CASTR.MENDF	当输入基准信号的有效边沿时 但是，如果给 CACR0.CFME 位写“1”，就不能在第 1 个有效边沿发生测量结束中断。
上溢中断	CAICR.OVFIE	CASTR.OVFF	当计数器发生上溢时

10.5 使用时的注意事项

10.5.1 模块停止功能的设定

能通过模块停止控制寄存器 C (MSTPCRC) 设定禁止或者允许时钟频率精度测量电路的运行。初始值为停止时钟频率精度测量电路的运行。能通过解除模块停止状态存取寄存器。详细内容请参照“11. 低功耗功能”。

11. 低功耗功能

11.1 概要

作为低功耗功能，RX63T 群有通过转换时钟来降低功耗的功能、模块停止功能以及向低功耗状态转移的功能。

低功耗功能的规格如表 11.1 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 11.2 所示。

复位后，在正常的程序运行中，除 DMAC、DTC 和内部 RAM 以外的模块处于停止状态。

表 11.1 低功耗功能的规格

项目	内容
通过转换时钟来降低功耗的功能	能给系统时钟（ICLK）、定时器模块时钟（PCLKA）、外围模块时钟（PCLKB）、S12AD 时钟（PCLKD）和闪存接口时钟（FCLK）分别设定分频比（注 1）。
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	<ul style="list-style-type: none"> 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	<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

注 1. 详细内容请参照“9. 时钟发生电路”。

表 11.2 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
复位以外的解除方法	中断	中断 (注1)	中断 (注2)	中断 (注3)
解除后的状态 (注4)	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)
主时钟振荡器	能运行	能运行	能运行 (注5)	能运行 (注5)
低速内部振荡器	能运行	能运行	停止	停止
IWDT 专用内部振荡器	能运行 (注6)	能运行 (注6)	能运行 (注6)	停止 (不定值) (注6)
PLL	能运行	能运行	停止	停止
CPU	停止 (保持)	停止 (保持)	停止 (保持)	停止 (不定值)
内部 RAM0 (0000 0000h ~ 0000 1FFFh)	能运行 (保持)	停止 (保持)	停止 (保持)	停止 (不定值)
闪存	运行	停止 (保持)	停止 (保持)	停止 (保持)
看门狗定时器 (WDT)	停止 (保持)	停止 (保持)	停止 (保持)	停止 (不定值)
独立看门狗定时器 (IWDT)	能运行 (注6)	能运行 (注6)	能运行 (注6)	停止 (不定值) (注6)
电压检测电路 (LVD)	能运行	能运行	能运行	能运行 (注7)
上电复位电路	运行	运行	运行	运行
外围模块	能运行	停止 (保持)	停止 (保持)	停止 (不定值)
I/O 端口	运行	保持	保持	保持

“能运行”表示能通过设定控制寄存器来控制运行或者停止。

“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. 这是外部引脚中断 (NMI、IRQ0 ~ IRQ5) 和外围功能中断 (IWDT、电压监视 1、电压监视 2、振荡停止检测)。

注 2. 这是外部引脚中断 (NMI、IRQ0 ~ IRQ5) 和外围功能中断 (IWDT、电压监视 1、电压监视 2)。

注 3. 这是作为外部引脚中断发生源的部分引脚 (NMI、IRQ0-DS ~ IRQ5-DS) 和外围功能中断 (电压监视 1、电压监视 2)。但是, 这些中断源只在深度待机中断允许寄存器 i (DPSIERi) (i=0、2) 的对应位为“1”时有效。给引脚名附加 -DS 的引脚, 能作为深度软件待机模式的解除引脚使用。

注 4. 通过 RES# 引脚、上电复位、电压监视复位、独立看门狗定时器复位进行的解除除外。当通过 RES# 引脚、上电复位、电压监视复位、独立看门狗定时器复位进行解除时, 转移到复位状态。

注 5. 能通过设定主时钟振荡器的强制振荡控制寄存器的主时钟振荡器强制振荡位 (MOFCR.MOFOXIN), 选择运行或者停止。

注 6. 在 IWDT 自动启动模式中, 能通过设定选项功能选择寄存器 0 的 IWDT 睡眠模式计数停止控制位 (OFS0.IWDTSLCSTP), 选择运行或者停止。当 OFS0.IWDTSLCSTP 位为“0” (在转移到低功耗模式时继续进行计数) 时, 不转移到深度软件待机模式而转移到软件待机模式。如果不在 IWDT 自动启动模式中, 就能通过设定 IWDT 计数停止控制寄存器的睡眠模式计数停止控制位 (IWDTCSLTPR.SLCSTP), 选择运行或者停止。当 IWDTCSLTPR.SLCSTP 位为“0” (在低功耗模式转移时继续进行计数) 时, 不转移到深度软件待机模式而转移到软件待机模式。

注 7. 当电压监视 1 电路控制寄存器 0 的电压监视 1 电路模式选择位 (LVD1CR0.LVD1RI) 为“1”或者电压监视 2 电路控制寄存器 0 的电压监视 2 电路模式选择位 (LVD2CR0.LVD2RI) 为“1”时, 不转移到深度软件待机模式而转移到软件待机模式。

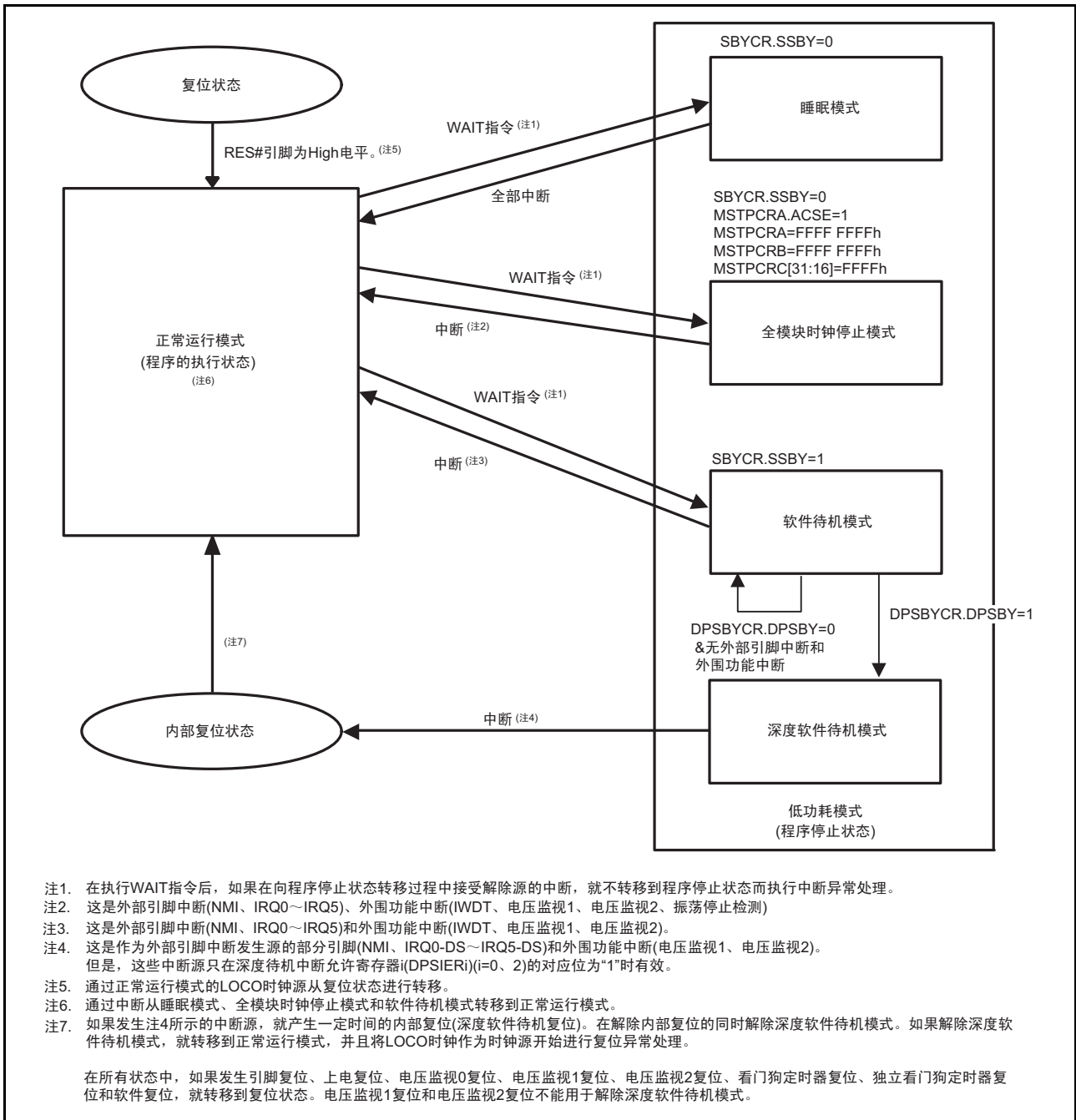


图 11.1 模式转移

11.2 寄存器说明

11.2.1 待机控制寄存器 (SBYCR)

地址 0008 000Ch

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

位	符号	位名	功能	R/W
b13-b0	—	保留位	读写值都为“0”。	R/W
b14	—	保留位	读写值都为“1”。	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

SSBY 位 (软件待机位)

此位设定 WAIT 指令执行后的转移目标。

如果在 SSBY 位为“1”的状态下执行 WAIT 指令, 就转移到软件待机模式。

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

当振荡停止检测控制寄存器的振荡停止检测功能允许位 (OSTDCR.OSTDE) 为“1”时, SSBY 位的设定值无效。当 SSBY 位为“1”时, 也在执行 WAIT 指令后转移到睡眠模式或者全模块时钟停止模式。

11.2.2 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	—	MSTPA29	MSTPA28	MSTPA27	—	—	MSTPA24	—	—	—	—	—	—	MSTPA17	—
复位后的值	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MSTPA15	MSTPA14	—	—	—	—	MSTPA9	—	MSTPA7	—	—	—	—	—	—	—
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
b6-b0	—	保留位	读写值都为“1”。	R/W
b7	MSTPA7	通用 PWM 定时器的模块停止设定位	对象模块：GPT 单元 0 (GPT0 ~ GPT3) 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b8	—	保留位	读写值都为“1”。	R/W
b9	MSTPA9	多功能定时器脉冲单元 3 的模块停止设定位	对象模块：MTU3 (MTU0 ~ MTU7) 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b13-b10	—	保留位	读写值都为“1”。	R/W
b14	MSTPA14	比较匹配定时器 (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 (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
b16	—	保留位	读写值都为“1”。	R/W
b17	MSTPA17	12 位 A/D 转换器的模块停止设定位	对象模块：S12AD 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b23-b18	—	保留位	读写值都为“1”。	R/W
b24	MSTPA24	12 位 A/D 转换器控制部的模块停止设定位	对象模块：S12AD 控制部 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b26-b25	—	保留位	读写值都为“1”。	R/W
b27	MSTPA27	模块停止 A27 设定位	读操作和写操作都有效。如果要转移到全模块停止模式，就必须预先给此位写“1”。	R/W
b28	MSTPA28	DMA 控制器 / 数据传送控制器的模块停止设定位	对象模块：DMAC/DTC 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b29	MSTPA29	模块停止 A29 设定位	读操作和写操作都有效。如果要转移到全模块停止模式，就必须预先给此位写“1”。	R/W
b30	—	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

ACSE 位 (全模块时钟停止模式允许位)

通过 ACSE 位设定允许或者禁止转移到全模块时钟停止模式。在将 ACSE 位置“1”并且 SBYCR.SSBY 位、MSTPCRA、MSTPCRB、MSTPCRC 寄存器满足规定条件的状态下，如果 CPU 执行 WAIT 指令，就转移到全模块时钟停止模式。详细内容请参照“11.5.2 全模块时钟停止模式”。

当 SBYCR.SSBY 位和 MSTPCRA.ACSE 位都为“0”时，在执行 WAIT 指令后转移到睡眠模式。

11.2.3 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
	MSTPB 31	MSTPB 30	—	—	—	—	—	—	MSTPB 23	—	MSTPB 21	—	—	—	MSTPB 17	—
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
	—	—	—	—	—	—	—	—	—	MSTPB6	—	MSTPB4	—	—	—	—
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	MSTPB4	串行通信接口 SCId 的模块停止设定位	对象模块: SCId (SCI12) 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b5	—	保留位	读写值都为“1”。	R/W
b6	MSTPB6	数据运算电路的模块停止设定位	对象模块: 数据运算电路 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b16-b7	—	保留位	读写值都为“1”。	R/W
b17	MSTPB17	串行外围接口 0 的模块停止设定位	对象模块: RSPI0 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b20-b18	—	保留位	读写值都为“1”。	R/W
b21	MSTPB21	I ² C 总线接口 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
b29-b24	—	保留位	读写值都为“1”。	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

11.2.4 模块停止控制寄存器 C (MSTPCRC)

地址 0008 0018h

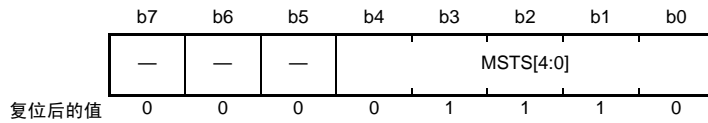
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC19	—	—	—
复位后的值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC0
复位后的值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	MSTPC0	RAM0 模块停止设定位 (注1)	对象模块: RAM0 (0000 0000h ~ 0000 1FFF) 0: 内部 RAM0 运行 1: 内部 RAM0 停止	R/W
b15-b1	—	保留位	读写值都为“0”。	R/W
b18-b16	—	保留位	读写值都为“1”。	R/W
b19	MSTPC19	时钟频率精度测量电路的模块停止设定位	对象模块: CAC 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b31-b20	—	保留位	读写值都为“1”。	R/W

注 1. 不能在存取内部 RAM 的过程中将 MSTPC0 位置“1”，而且不能在 MSTPC0 位为“1”的状态下存取内部 RAM。

11.2.5 主时钟振荡器的等待控制寄存器 (MOSCWTCR)

地址 0008 00A2h



位	符号	位名	功能	R/W
b4-b0	MSTS[4:0]	主时钟振荡器的等待时间设定位	b4 b0 00000: 待机时间 =2 个周期 00001: 待机时间 =4 个周期 00010: 待机时间 =8 个周期 00011: 待机时间 =16 个周期 00100: 待机时间 =32 个周期 00101: 待机时间 =64 个周期 00110: 待机时间 =512 个周期 00111: 待机时间 =1024 个周期 01000: 待机时间 =2048 个周期 01001: 待机时间 =4096 个周期 01010: 待机时间 =16384 个周期 01011: 待机时间 =32768 个周期 01100: 待机时间 =65536 个周期 01101: 待机时间 =131072 个周期 01110: 待机时间 =262144 个周期 01111: 待机时间 =524288 个周期 上述以外: 不能设定	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W

MOSCWTCR 寄存器是选择主时钟振荡器的振荡稳定等待时间的寄存器。

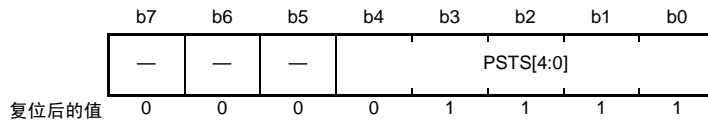
在对主时钟进行 MOSCWTCR 寄存器所设周期数的计数后, 开始给 LSI 内部提供主时钟。

必须将 MSTS[4:0] 位至少设定为主时钟振荡稳定时间 (tMAINOSC) 的待机时间。例如, 当使用的谐振器的振荡频率为 10MHz (周期为 100ns) 时, 如果将 MSTS[4:0] 位置“01101b”, 待机时间就为 100ns×131072 个周期≈ 13.12ms。在将主时钟用于外部输入时, 不需要待机时间。

只能在 MOSCCR.MOSTP 位为“1”时改写 MOSCWTCR 寄存器, 否则就不能进行改写。

11.2.6 PLL 等待控制寄存器 (PLLWTCR)

地址 0008 00A6h



位	符号	位名	功能	R/W
b4-b0	PSTS[4:0]	PLL 等待时间设定位	b4 b0 00000: 待机时间 =16 个周期 00001: 待机时间 =32 个周期 00010: 待机时间 =64 个周期 00011: 待机时间 =512 个周期 00100: 待机时间 =1024 个周期 00101: 待机时间 =2048 个周期 00110: 待机时间 =4096 个周期 00111: 待机时间 =16384 个周期 01000: 待机时间 =32768 个周期 01001: 待机时间 =65536 个周期 01010: 待机时间 =131072 个周期 01011: 待机时间 =262144 个周期 01100: 待机时间 =524288 个周期 01101: 待机时间 =1048576 个周期 01110: 待机时间 =2097152 个周期 01111: 待机时间 =4194304 个周期 上述以外: 不能设定	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W

PLLWTCR 寄存器是选择 PLL 振荡稳定等待时间的寄存器。

在对 PLL 时钟进行 PLLWTCR 寄存器所设周期数的计数后, 开始给 LSI 内部提供 PLL 时钟。

在将晶体谐振器用作谐振器并且不等待主时钟振荡稳定就将 PLL 设定为运行的情况下, 必须将 PSTS[4:0] 位至少设定为 PLL 振荡稳定时间 (不等主时钟振荡稳定就将 PLL 设定为 ON 的情况) (t_{PLL2}) 的待机时间。例如, 当 PLL 振荡频率为 200MHz (周期为 5ns) 时, 如果将 PSTS[4:0] 位置“01110b”, 待机时间就为 $5ns \times 2097152$ 个周期 $\approx 10.48ms$ 。

在将晶体谐振器用作谐振器并且在等待主时钟振荡稳定后将 PLL 设定为运行以及将主时钟用于外部输入的情况下, 因为不需要等待基准时钟的振荡稳定时间, 所以必须将 PSTS[4:0] 位至少设定为 PLL 振荡稳定时间 (在主时钟振荡稳定后将 PLL 设定为 ON 的情况) (t_{PLL1}) 的待机时间。

只能在 PLLCR2.PLEN 位为“1” (PLL 停止) 时改写 PLLWTCR 寄存器, 否则就不能进行改写。

11.2.7 深度待机控制寄存器 (DPSBYCR)

地址 0008 C280h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSBY	IOKEEP	—	—	—	—	—	—
复位后的地址	0	0	0	0	0	0	0	1

位	符号	位名	功能	R/W
b0	—	保留位	读写值都为“1”。	R/W
b5-b1	—	保留位	读写值都为“0”。	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: 在解除深度软件待机模式后仍然保持 I/O 端口的状态。此后, 如果给 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或者全模块时钟停止模式 0 1: 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 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

深度软件待机模式解除源的内部复位信号不对 DPSBYCR 寄存器进行初始化, 详细内容请参照“表 6.2 各种复位的初始化对象”。

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中相同的状态。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)

此位控制向深度软件待机模式的转移。

如果在 SBYCR.SSBY 位为“1”并且 DPSBY 位为“1”的状态下执行 WAIT 指令, 就经过软件待机模式转移到深度软件待机模式。

如果通过作为外部引脚中断发生源的部分引脚 (NMI、IRQ0-DS ~ IRQ5-DS) 或者外围功能中断 (电压监视 1、电压监视 2) 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 IWDT 为自动启动模式并且 OFS0.IWDTSLCSTP 位为“0” (继续进行计数) 时或者在 IWDT 为寄存器启动模式并且 IWDTTCSTPR.SLCSTP 位为“0”时, DPSBY 位的设定值无效。此时, 即使 SBYCR.SSBY 位为“1”并且 DPSBY 位为“1”, 也在执行 WAIT 指令后转移到软件待机模式。

如果在电压监视 1 电路模式中选择电压监视 1 复位 (LVD1CR0.LVD1RI=1) 或者在电压监视 2 电路模式中选择电压监视 2 复位 (LVD2CR0.LVD2RI=1), DPSBY 位的设定值就无效。此时, 即使 SBYCR.SSBY 位为“1”并且 DPSBY 位为“1”, 也在执行 WAIT 指令后转移到软件待机模式。

11.2.8 深度待机中断允许寄存器 0 (DPSIER0)

地址 0008 C282h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	DIRQ0E	IRQ0-DS 引脚允许位	0: 禁止通过 IRQ0-DS 引脚解除深度软件待机模式 1: 允许通过 IRQ0-DS 引脚解除深度软件待机模式	R/W
b1	DIRQ1E	IRQ1-DS 引脚允许位	0: 禁止通过 IRQ1-DS 引脚解除深度软件待机模式 1: 允许通过 IRQ1-DS 引脚解除深度软件待机模式	R/W
b2	DIRQ2E	IRQ2-DS 引脚允许位	0: 禁止通过 IRQ2-DS 引脚解除深度软件待机模式 1: 允许通过 IRQ2-DS 引脚解除深度软件待机模式	R/W
b3	DIRQ3E	IRQ3-DS 引脚允许位	0: 禁止通过 IRQ3-DS 引脚解除深度软件待机模式 1: 允许通过 IRQ3-DS 引脚解除深度软件待机模式	R/W
b4	DIRQ4E	IRQ4-DS 引脚允许位	0: 禁止通过 IRQ4-DS 引脚解除深度软件待机模式 1: 允许通过 IRQ4-DS 引脚解除深度软件待机模式	R/W
b5	DIRQ5E	IRQ5-DS 引脚允许位	0: 禁止通过 IRQ5-DS 引脚解除深度软件待机模式 1: 允许通过 IRQ5-DS 引脚解除深度软件待机模式	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W

深度软件待机模式解除源的内部复位信号不对 DPSIER0 寄存器进行初始化，详细内容请参照“表 6.2 各种复位的初始化对象”。

如果更改 DPSIER0 寄存器的设定，就可能根据引脚状态在内部产生边沿，使 DPSIFR0 寄存器变为“1”。必须在向深度软件待机模式转移前将 DPSIFR0 寄存器置“0”。

即使 DPSIER0 寄存器为“0”，也可能在向深度软件待机模式转移时，根据引脚状态在内部产生上升沿，使 DPSIFR0 寄存器变为“1”。因为在 DPSIEGR0 寄存器为“0”时不检测上升沿，所以 DPSIFR0 寄存器不变为“1”。

11.2.9 深度待机中断允许寄存器 2 (DPSIER2)

地址 0008 C284h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	DNMIE	—	—	DLVD2IE	DLVD1IE
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	DLVD1IE	LVD1 深度待机解除信号允许位	0: 禁止通过电压监视 1 信号解除深度软件待机模式 1: 允许通过电压监视 1 信号解除深度软件待机模式	R/W
b1	DLVD2IE	LVD2 深度待机解除信号允许位	0: 禁止通过电压监视 2 信号解除深度软件待机模式 1: 允许通过电压监视 2 信号解除深度软件待机模式	R/W
b3-b2	—	保留位	读写值都为“0”。	R/W
b4	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)
b7-b5	—	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。

深度软件待机模式解除源的内部复位信号不对 DPSIER2 寄存器进行初始化，详细内容请参照“表 6.2 各种复位的初始化对象”。

如果更改 DPSIER2 寄存器的设定，就可能根据引脚状态在内部产生边沿，使 DPSIFR2 寄存器变为“1”。必须在向深度软件待机模式转移前将 DPSIFR2 寄存器置“0”。

即使 DPSIER2 寄存器为“0”，也可能在向深度软件待机模式转移时，根据引脚状态在内部产生上升沿，使 DPSIFR2 寄存器变为“1”。因为在 DPSIEGR2 寄存器为“0”时不检测上升沿，所以 DPSIFR2 寄存器不变为“1”。

11.2.10 深度待机中断标志寄存器 0 (DPSIFR0)

地址 0008 C286h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	DIRQ5F	DIRQ4F	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	DIRQ0F	IRQ0-DS 引脚深度待机解除标志	0: 未发生由 IRQ0-DS 引脚引起的解除请求 1: 发生由 IRQ0-DS 引脚引起的解除请求	R(W) (注1)
b1	DIRQ1F	IRQ1-DS 引脚深度待机解除标志	0: 未发生由 IRQ1-DS 引脚引起的解除请求 1: 发生由 IRQ1-DS 引脚引起的解除请求	R(W) (注1)
b2	DIRQ2F	IRQ2-DS 引脚深度待机解除标志	0: 未发生由 IRQ2-DS 引脚引起的解除请求 1: 发生由 IRQ2-DS 引脚引起的解除请求	R(W) (注1)
b3	DIRQ3F	IRQ3-DS 引脚深度待机解除标志	0: 未发生由 IRQ3-DS 引脚引起的解除请求 1: 发生由 IRQ3-DS 引脚引起的解除请求	R(W) (注1)
b4	DIRQ4F	IRQ4-DS 引脚深度待机解除标志	0: 未发生由 IRQ4-DS 引脚引起的解除请求 1: 发生由 IRQ4-DS 引脚引起的解除请求	R(W) (注1)
b5	DIRQ5F	IRQ5-DS 引脚深度待机解除标志	0: 未发生由 IRQ5-DS 引脚引起的解除请求 1: 发生由 IRQ5-DS 引脚引起的解除请求	R(W) (注1)
b7-b6	—	保留位	读写值都为“0”。	R/W

注 1. 只能写“0”。

如果发生 DPSIEGR0 寄存器设定的解除请求，DPSIFR0 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，DPSIFR0 寄存器也可能变为“1”。也有可能因更改 DPSIER0 寄存器的设定而使 DPSIFR0 寄存器变为“1”。因此必须在将 DPSIFR0 寄存器置“00h”后转移到深度软件待机模式。

要在更改 DPSIER0 寄存器的设定后将 DPSIFR0 寄存器置“00h”时，必须在至少经过 6 个 PCLKB 周期并且读 DPSIFR0 寄存器后，给 DPSIFR0 寄存器写“0”。例如，能通过读 DPSIER0 寄存器，至少确保 6 个 PCLKB 周期。

深度软件待机模式解除源的内部复位信号不对 DPSIFR0 寄存器进行初始化，详细内容请参照“表 6.2 各种复位的初始化对象”。

DIRQnF (IRQn 深度待机解除标志) (n=0 ~ 5)

此标志表示发生由 IRQn-DS 引脚引起的解除请求。

[为“1”的条件]

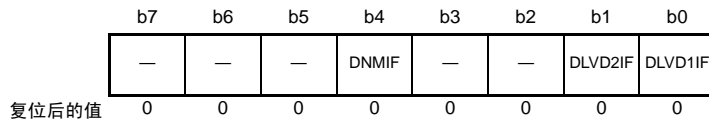
- 当发生由 DPSIEGR0 寄存器所选 IRQn-DS 引脚引起的解除请求时

[为“0”的条件]

- 在读“1”后写“0”时

11.2.11 深度待机中断标志寄存器 2 (DPSIFR2)

地址 0008 C288h



位	符号	位名	功能	R/W
b0	DLVD1IF	LVD1 深度待机解除标志	0: 未发生由电压监视 1 信号引起的解除请求 1: 发生由电压监视 1 信号引起的解除请求	R(W) (注 1)
b1	DLVD2IF	LVD2 深度待机解除标志	0: 未发生由电压监视 2 信号引起的解除请求 1: 发生由电压监视 2 信号引起的解除请求	R(W) (注 1)
b3-b2	—	保留位	读写值都为“0”。	R/W
b4	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)
b7-b5	—	保留位	读写值都为“0”。	R/W

注 1. 只能写“0”。

如果发生 DPSIEGR2 寄存器设定的解除请求，DPSIFR2 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，DPSIFR2 寄存器也可能变为“1”。也有可能因更改 DPSIER2 寄存器的设定而使 DPSIFR2 寄存器变为“1”。因此必须在将 DPSIFR2 寄存器置“00h”后转移到深度软件待机模式。

要在更改 DPSIER2 寄存器的设定后将 DPSIFR2 寄存器置“00h”时，必须在至少经过 6 个 PCLKB 周期并且读 DPSIFR2 寄存器后，给 DPSIFR2 寄存器写“0”。例如，能通过读 DPSIER2 寄存器，至少确保 6 个 PCLKB 周期。

深度软件待机模式解除源的内部复位信号不对 DPSIFR2 寄存器进行初始化，详细内容请参照“表 6.2 各种复位的初始化对象”。

DLVDmIF (LVDm 深度待机解除标志) (m=1、2)

此标志表示发生由电压监视 m 信号引起的解除请求。

[为“1”的条件]

- 当发生由 DPSIEGR2 寄存器所选电压监视 m 信号引起的解除请求时

[为“0”的条件]

- 在读“1”后写“0”时

DNMIF (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]

- 当发生由 DPSIEGR2 寄存器所设 NMI 引脚引起的解除请求时

[为“0”的条件]

- 在读“1”后写“0”时

11.2.12 深度待机中断边沿寄存器 0 (DPSIEGR0)

地址 0008 C28Ah

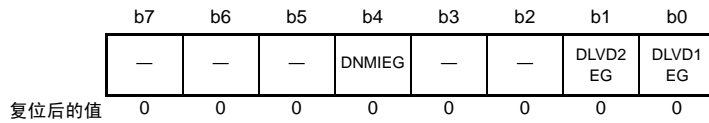
	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DIRQ5 EG	DIRQ4 EG	DIRQ3 EG	DIRQ2 EG	DIRQ1 EG	DIRQ0 EG
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	DIRQ0EG	IRQ0-DS 引脚边沿选择位	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-DS 引脚边沿选择位	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-DS 引脚边沿选择位	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-DS 引脚边沿选择位	0: 在下降沿产生解除请求 1: 在上升沿产生解除请求	R/W
b4	DIRQ4EG	IRQ4-DS 引脚边沿选择位	0: 在下降沿产生解除请求 1: 在上升沿产生解除请求	R/W
b5	DIRQ5EG	IRQ5-DS 引脚边沿选择位	0: 在下降沿产生解除请求 1: 在上升沿产生解除请求	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W

深度软件待机模式解除源的内部复位信号不对 DPSIEGR0 寄存器进行初始化，详细内容请参照“表 6.2 各种复位的初始化对象”。

11.2.13 深度待机中断边沿寄存器 2 (DPSIEGR2)

地址 0008 C28Ch

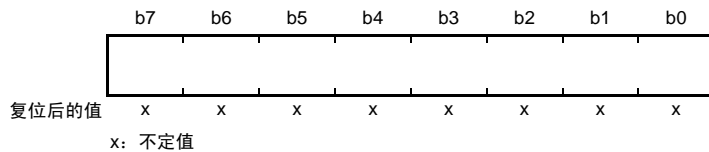


位	符号	位名	功能	R/W
b0	DLVD1EG	LVD1 边沿选择位	0: 在检测到 $VCC < V_{det1}$ (下降) 时产生解除请求 1: 在检测到 $VCC \geq V_{det1}$ (上升) 时产生解除请求	R/W
b1	DLVD2EG	LVD2 边沿选择位	0: 在检测到 $VCC < V_{det2}$ (下降) 时产生解除请求 1: 在检测到 $VCC \geq V_{det2}$ (上升) 时产生解除请求	R/W
b3-b2	—	保留位	读写值都为“0”。	R/W
b4	DNMIEG	NMI 边沿选择位	0: 在下降沿产生解除请求 1: 在上升沿产生解除请求	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W

深度软件待机模式解除源的内部复位信号不对 DPSIEGR2 寄存器进行初始化，详细内容请参照“表 6.2 各种复位的初始化对象”。

11.2.14 深度待机备用寄存器 (DPSBKRY) (y=0 ~ 31)

地址 0008 C2A0h ~ 0008 C2BFh



DPSBKRY 寄存器是用于在深度软件待机模式中保存数据的 32 字节可读写寄存器。不对 DPSBKRY 寄存器进行初始化。在刚接通电源后，寄存器的值为不定值。

11.3 通过转换时钟来降低功耗

如果设定 SCKCR.FCK[3:0] 位、SCKCR.ICK[3:0] 位、SCKCR.PCKA[3:0] 位、SCKCR.PCKB[3:0] 位和 SCKCR.PCKD[3:0] 位，就改变时钟频率。CPU、DMAC、DTC、内部 ROM 和内部 RAM 以 ICK[3:0] 位设定的运行时钟运行。

MTU3 和 GPT 以 PCKA[3:0] 位设定的运行时钟运行。

外围模块以 PCKB[3:0] 位设定的运行时钟运行。

S12AD 以 PCKD[3:0] 位设定的运行时钟运行。

闪存接口以 FCK[3:0] 位设定的运行时钟运行。

详细内容请参照“9. 时钟发生电路”。

11.4 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 MSTPmi 位 (m=A ~ C、i=31 ~ 0) 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 MSTPmi 位置“0”，解除模块停止状态并且在总线周期结束时模块重新开始运行。在模块停止状态下，保持模块的内部状态。

在解除复位后，除 DMAC、DTC 和内部 RAM 以外的全部模块都处于模块停止状态。

不能读写被设定为模块停止状态的模块的寄存器。但是必须注意：如果在设定模块停止后立即进行写操作，也许能写。

11.5 低功耗状态

11.5.1 睡眠模式

11.5.1.1 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，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。CPU 以外的外围功能不停止运行。

如果在使用 WDT 时转移到睡眠模式，WDT 就停止计数。

在自动启动模式中使用 IWDT 的情况下，如果在 OFS0.IWDTSLCSTP 位为“1”时转移到睡眠模式，IWDT 就停止计数。同样，在寄存器启动模式中使用 IWDT 的情况下，如果在 IWDCSTPR.SLCSTP 位为“1”时转移到睡眠模式，IWDT 就停止计数。

在自动启动模式中使用 IWDT 的情况下，即使在 OFS0.IWDTSLCSTP 位为“0”（在转移到低功耗模式时，IWDT 继续进行计数）时转移到睡眠模式，IWDT 也继续进行计数。同样，在寄存器启动模式中使用 IWDT 的情况下，即使在 IWDCSTPR.SLCSTP 位为“0”时转移到睡眠模式，IWDT 也继续进行计数。

要使用睡眠模式时，必须在进行以下设定后执行 WAIT 指令：

1. 将 CPU 的 PSW.I 位（注1）置“0”。
2. 将用于从睡眠模式返回的中断请求目标设定为 CPU。
3. 将用于从睡眠模式返回的中断优先级（注2）设定为高于 CPU 的 PSW.IPL[3:0] 位（注1）的优先级。
4. 将用于从睡眠模式返回的中断的 IERm.IENj 位（注2）置“1”。
5. 读最后写的 I/O 寄存器并且确认写入值已被反映。
6. 执行 WAIT 指令（通过执行 WAIT 指令，CPU 的 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“14. 中断控制器（ICUb）”。

11.5.1.2 睡眠模式的解除

通过全部中断、RES# 引脚复位、上电复位、电压监视复位或者 IWDT 下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，就解除睡眠模式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[3:0] 位（注2）的优先级），就不解除睡眠模式。
- 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。
- 通过上电复位进行的解除
通过上电复位来解除睡眠模式。
- 通过电压监视复位进行的解除
通过电压检测电路的电压监视复位来解除睡眠模式。
- 通过独立看门狗定时器复位进行的解除
通过 IWDT 下溢的内部复位来解除睡眠模式。但是在睡眠模式中，因为在 IWDT 停止计数的条件（OFS0.IWDTSTRT=0 并且 OFS0.IWDTSLCSTP=1，或者 OFS0.IWDTSTRT=1 并且 IWDCSTPR.SLCSTP=1）下 IWDT 停止运行，所以不能通过独立看门狗定时器复位进行解除。

注 1. 详细内容请参照“14. 中断控制器（ICUb）”。

注 2. 详细内容请参照“2. CPU”。

11.5.2 全模块时钟停止模式

11.5.2.1 向全模块时钟停止模式的转移

在将 MSTPCRA.ACSE 位置“1”并且将由 MSTPCRA、MSTPCRB、MSTPCRC 寄存器控制的模块置为模块停止状态（MSTPCRA=FFFF FFFFh（注1）、MSTPCRB=FFFF FFFFh、MSTPCRC[31:16]=FFFFh）的情况下，如果在将 SBYCR.SSBY 位置“0”的状态下执行 WAIT 指令，总线控制器、I/O 端口以及除 IWDT、上电复位电路、电压检测电路以外的全部模块就在总线周期结束时停止运行，转移到全模块时钟停止模式（注2）。

如果在使用 WDT 时转移到全模块时钟停止模式，WDT 就停止计数。

在自动启动模式中使用 IWDT 的情况下，如果在 OFS0.IWDTSLCSTP 位为“1”时转移到全模块时钟停止模式，IWDT 就停止计数。同样，在寄存器启动模式中使用 IWDT 的情况下，如果在 IWDTCSSTPR.SLCSTP 位为“1”时转移到全模块时钟停止模式，IWDT 就停止计数。

在自动启动模式中使用 IWDT 的情况下，即使在 OFS0.IWDTSLCSTP 位为“0”（在转移到低功耗模式时，IWDT 继续进行计数）时转移到全模块时钟停止模式，IWDT 也继续进行计数。同样，在寄存器启动模式中使用 IWDT 的情况下，即使在 IWDTCSSTPR.SLCSTP 位为“0”时转移到全模块时钟停止模式，IWDT 也继续进行计数。

要使用全模块时钟停止模式时，必须在进行以下设定后执行 WAIT 指令：

1. 将 CPU 的 PSW.I 位（注2）置“0”。
2. 将用于从全模块时钟停止模式返回的中断传达目标设定为 CPU。
3. 将用于从全模块时钟停止模式返回的中断优先级（注3）设定为高于 CPU 的 PSW.IPL[3:0] 位（注2）的优先级。
4. 将用于从全模块时钟停止模式返回的中断的 IERm.IENj 位（注2）置“1”。
5. 读最后写的 I/O 寄存器并且确认写入值已被反映。
6. 执行 WAIT 指令（通过执行 WAIT 指令，CPU 的 PSW.I 位（注2）自动变为“1”）。

注 1. 有可能因 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 位置“1”前，将 DMAC 的 DMAST.DMST 位和 DTC 的 DTCST.DTCST 位置“0”，并且在未启动 DTC 和 DMAC 的状态下进行转移。

注 2. 详细内容请参照“2. CPU”。

注 3. 详细内容请参照“14. 中断控制器（ICUb）”。

11.5.2.2 全模块时钟停止模式的解除

通过外部引脚中断（NMI、IRQ0～IRQ5）、外围功能中断（IWDT、电压监视 1、电压监视 2、振荡停止检测）、RES# 引脚复位、上电复位、电压监视复位或者独立看门狗定时器复位来解除全模块时钟停止模式，在经过异常处理后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1）小于等于 CPU 的 PSW.IPL[3:0] 位（注2）的优先级）或者被设定为 DTC 和 DMAC 的启动源，就不解除全模块时钟停止模式。

注 1. 详细内容请参照“14. 中断控制器（ICUb）”。

注 2. 详细内容请参照“2. CPU”。

11.5.3 软件待机模式

11.5.3.1 向软件待机模式的转移

如果在将 SBYCR.SSBY 位置“1”并且将 DPSBYCR.DPSBY 位置“0”的状态下执行 WAIT 指令，就转移到软件待机模式（注1）。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、内部外围功能状态和 I/O 端口状态。因为在软件待机模式中振荡器停止振荡，所以功耗明显降低。

必须在执行 WAIT 指令前将 DMAC 的 DMAST.DMST 位和 DTC 的 DTCST.DTCST 位置“0”。

如果在使用 WDT 时转移到软件待机模式，因为振荡器停止振荡，所以 WDT 就停止计数。

在自动启动模式中使用 IWDT 的情况下，如果在 OFS0.IWDTSLCSTP 位为“1”时转移到软件待机模式，IWDT 就停止计数。同样，在寄存器启动模式中使用 IWDT 的情况下，如果在 IWDTCSTPR.SLCSTP 位为“1”时转移到软件待机模式，IWDT 就停止计数。

在自动启动模式中使用 IWDT 的情况下，即使在 OFS0.IWDTSLCSTP 位为“0”（在转移到低功耗模式时，IWDT 继续进行计数）时转移到软件待机模式，IWDT 也继续进行计数。同样，在寄存器启动模式中使用 IWDT 的情况下，即使在 IWDTCSTPR.SLCSTP 位为“0”时转移到软件待机模式，IWDT 也继续进行计数。

当振荡停止检测功能有效（OSTDCR.OSTDE=1）时，不能转移到软件待机模式。如果要转移到软件待机模式，就必须在将振荡停止检测功能设定为无效（OSTDCR.OSTDE=0）后执行 WAIT 指令。

要使用软件待机模式时，必须在进行以下设定后执行 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传达目标设定为 CPU。
3. 将用于从软件待机模式返回的中断优先级（注2）设定为高于 CPU 的 PSW.IPL[3:0] 位（注1）的优先级。
4. 将用于从软件待机模式返回的中断的 IERm.IENj 位（注2）置“1”。
5. 读最后写的 I/O 寄存器并且确认写入值已被反映。
6. 执行 WAIT 指令（通过执行 WAIT 指令，CPU 的 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“14. 中断控制器（ICUb）”。

11.5.3.2 软件待机模式的解除

通过外部引脚中断（NMI、IRQ0～IRQ5）、外围功能中断（电压监视 1、电压监视 2）、RES# 引脚复位、上电复位、电压监视复位或者独立看门狗定时器复位来解除软件待机模式。通过中断解除软件待机模式时，转移到软件待机模式前运行的各振荡器就重新开始运行。在等待这些全部振荡器振荡稳定后，从软件待机模式返回。

- 通过中断进行的解除

如果发生 NMI、IRQ0～IRQ5、IWDT、电压监视 1 和电监视 2 的中断请求，转移到软件待机模式前运行的各振荡器就重新开始运行。此后，经过 MOSCWTCR.MSTS[4:0] 位、SOSCWTCR.SSTS[4:0] 位和 PLLWTCR.PSTS[4:0] 位设定的各振荡器的振荡稳定等待时间后，解除软件待机模式，开始中断异常处理。

- 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，并且在时钟开始振荡的同时给 LSI 提供时钟。此时，必须将 RES# 引脚的 Low 电平保持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。

- 通过上电复位进行的解除

如果因电源电压的降低而产生上电复位，就解除软件待机模式。

- 通过电压监视复位进行的解除

如果因电源电压的降低而产生电压监视复位，就解除软件待机模式。

- 通过独立看门狗定时器复位进行的解除

通过 IWDT 下溢的内部复位来解除软件待机模式。

但是在软件待机模式中，因为在独立看门狗定时器停止计数的条件（OFS0.IWDTSTRT=0 并且 OFS0.IWDTSLCSTP=1，或者 OFS0.IWDTSTRT=1 并且 IWDTCSTPR.SLCSTP=1）下独立看门狗定时器停止运行，所以不能通过独立看门狗定时器复位进行解除。

11.5.3.3 软件待机模式的应用例子

在 IRQn 引脚的下降沿转移到软件待机模式并且在 IRQn 引脚的上升沿解除软件待机模式的例子如图 11.2 所示。

此例中，在 ICU 的 IRQCRi.IRQMD[1:0] (i=0 ~ 5) 位为 “01b” (下降沿) 的状态下接受 IRQn (n=0 ~ 5) 中断后，将 IRQCRi.IRQMD[1:0] (i=0 ~ 5) 位置 “10b” (上升沿) 并且将 SBYCR.SSBY 位置 “1”，然后执行 WAIT 指令，转移到软件待机模式。此后，在 IRQn 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 “14. 中断控制器 (ICUb) ”。

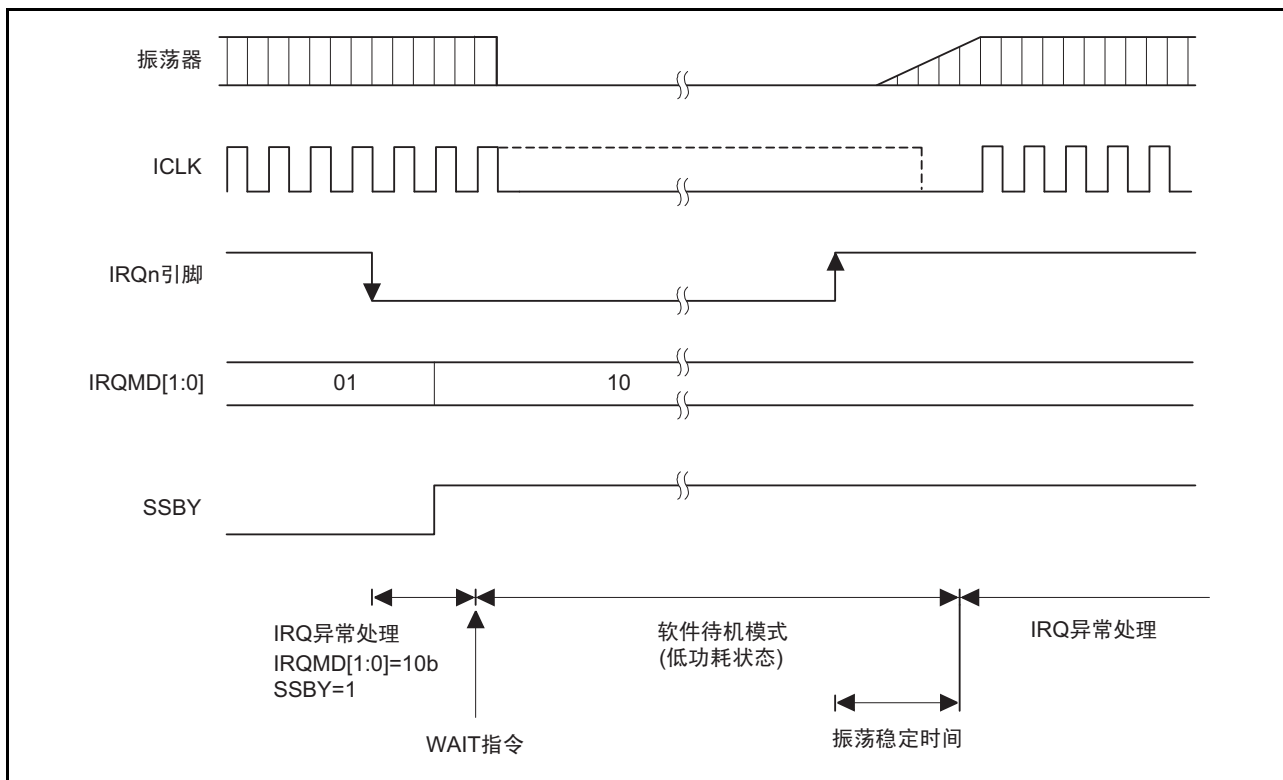


图 11.2 软件待机模式的应用例子

11.5.4 深度软件待机模式

11.5.4.1 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。

因为在深度软件待机模式中 CPU、内部外围功能、内部 RAM 和振荡器的全部功能停止运行并且停止给它们提供内部电源，所以功耗明显降低。此时，CPU、内部外围功能的寄存器内容都为不定值。

如果在使用 WDT 时转移到深度软件待机模式，振荡器就停止振荡，停止给 WDT 供电并且停止计数。

在自动启动模式中使用 IWDT 的情况下，如果在 OFS0.IWDTSLCSTP 位为“1”时转移到深度软件待机模式，就停止给 IWDT 专用低速时钟和 IWDT 供电并且停止计数。同样，在寄存器启动模式中使用 IWDT 的情况下，如果在 IWDTCSSTPR.SLCSTP 位为“1”时转移到深度软件待机模式，就停止给 IWDT 专用低速时钟和 IWDT 供电并且停止计数。

在自动启动模式中使用 IWDT 的情况下，当 OFS0.IWDTSLCSTP 位为“0”（在转移到低功耗模式时，IWDT 继续进行计数）时不转移到深度软件待机模式而转移到软件待机模式，并且继续进行计数。同样，在寄存器启动模式中使用 IWDT 的情况下，当 IWDTCSSTPR.SLCSTP 位为“0”时不转移到深度软件待机模式而转移到软件待机模式，并且继续进行计数。

如果在电压检测电路中选择电压监视 1 复位功能（LVD1CR0.LVD1RI=1）或者电压监视 2 复位功能（LVD2CR0.LVD2RI=1），就不能转移到深度软件待机模式而转移到软件待机模式。

端口保持软件待机模式中的状态。

注 1. 在执行 WAIT 指令前，需要满足转移到软件待机模式时的 DTC、DMAC、IWDT 的相关条件。详细内容请参照“11.5.3 软件待机模式”。

11.5.4.2 深度软件待机模式的解除

通过作为外部引脚中断发生源的部分引脚（NMI、IRQ0-DS ~ IRQ5-DS）、外围功能中断（电压监视 1、电压监视 2）、RES# 引脚复位、上电复位或者电压监视 0 复位来解除深度软件待机模式。

1. 通过外部中断引脚和内部中断信号进行的解除

通过 DPSIER_n（n=0、2）寄存器和 DPSIFR_n（n=0、2）寄存器控制深度软件待机模式的解除。如果发生能解除深度软件待机模式的中断请求，DPSIFR_n（n=0、2）寄存器的对应标志就变为“1”。此时，在 DPSIER_n（n=0、2）寄存器允许解除源的情况下解除深度软件待机模式。能通过 DPSIEGR_n（n=0、2）选择上升沿或者下降沿。可选择边沿的中断为 NMI、IRQ0-DS ~ IRQ5-DS、电压监视 1 和电压监视 2。

如果发生深度软件待机模式的解除请求，就提供内部电源，LOCO 时钟开始振荡，并且对整个 LSI 全体进行内部复位（深度软件待机复位）。此后，给整个 LSI 提供稳定的 LOCO 时钟并且解除深度软件待机复位。如果在解除深度软件待机复位的同时解除深度软件待机模式，就开始复位异常处理。

如果通过外部中断引脚和内部中断信号解除深度软件待机模式，RSTSR0.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，就解除深度软件待机模式。此时，必须根据“36. 电特性”，将 RES# 引脚保持为 Low 电平。如果将 RES# 引脚置 High 电平，CPU 就开始复位异常处理。

3. 通过上电复位进行的解除

如果因电源电压的降低而发生上电复位，就解除深度软件待机模式。

4. 通过电压监视 0 复位进行的解除

如果因电源电压的降低而发生电压监视 0 复位，就解除深度软件待机模式。

11.5.4.3 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保持软件待机模式中的 I/O 端口状态。

- DPSBYCR.IOKEEP 位为“0”的情况
通过解除深度软件待机模式的内部复位，使 I/O 端口变为初始状态。
- DPSBYCR.IOKEEP 位为“1”的情况
通过解除深度软件待机模式的内部复位对 LSI 内部进行初始化，I/O 端口与 LSI 的内部状态无关，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，也保持软件待机模式中的 I/O 端口状态。此后，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。

解除深度软件待机模式的内部复位不对 DPSBYCR.IOKEEP 位进行初始化。

11.5.4.4 深度软件待机模式的应用例子

在 IRQn-DS 引脚的下降沿转移到深度软件待机模式并且在 IRQn-DS 引脚的上升沿解除深度软件待机模式的运行例子如图 11.3 所示。

此例中，在 ICU 的 IRQCRi.IRQMD[1:0] 位为“01b”（下降沿）的状态下接受 IRQn 中断后，将 DPSIEGRy.DIRQnEG 位（y=0、n=0 ~ 5）置“1”（上升沿）并且将 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 IRQn-DS 引脚的上升沿解除深度软件待机模式。

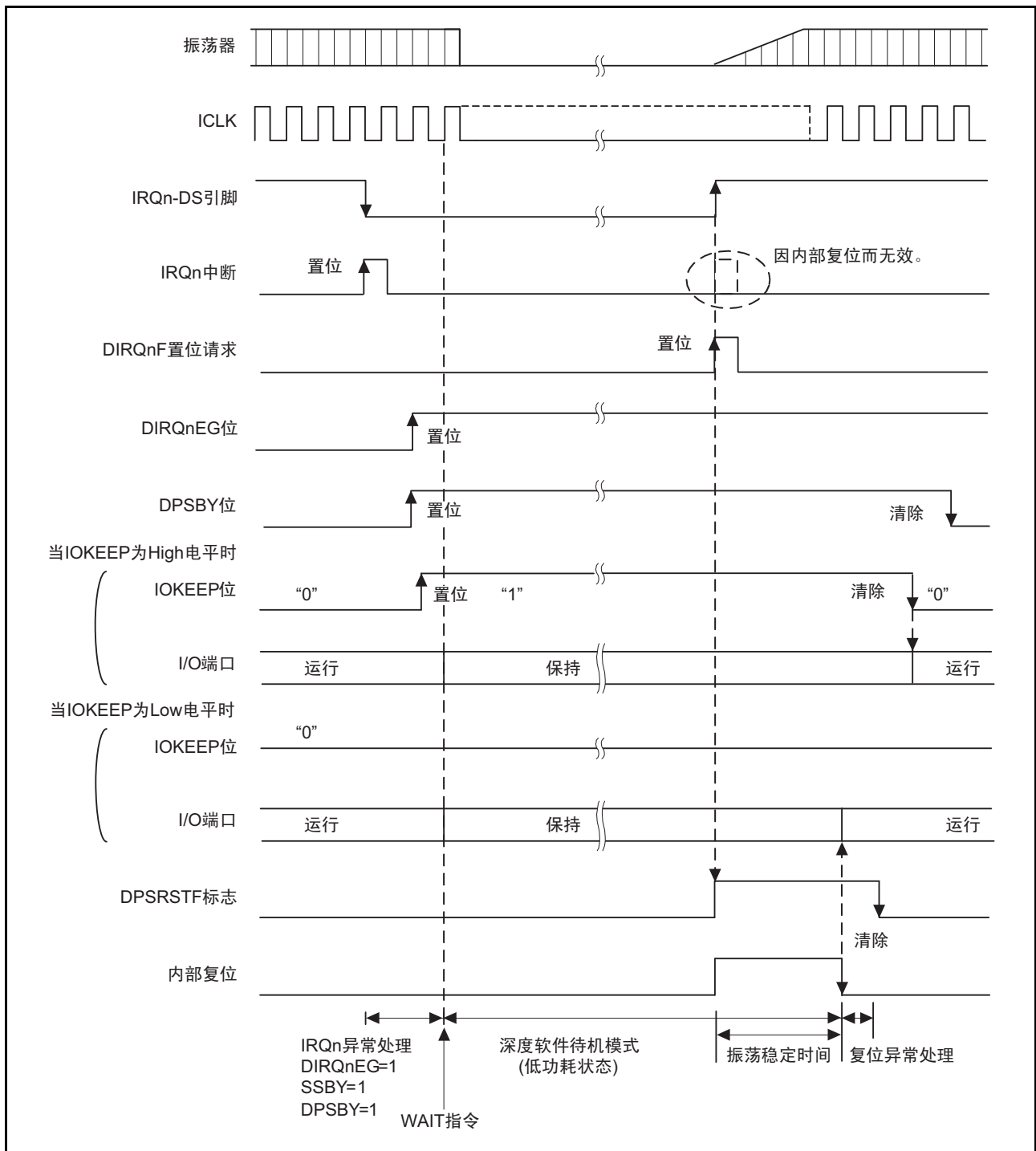


图 11.3 深度软件待机模式的应用例子

11.5.4.5 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 11.4 所示。

此例中，在进行复位异常处理后，根据复位功能的 RSTSR0.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。

当通过 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。

当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”。

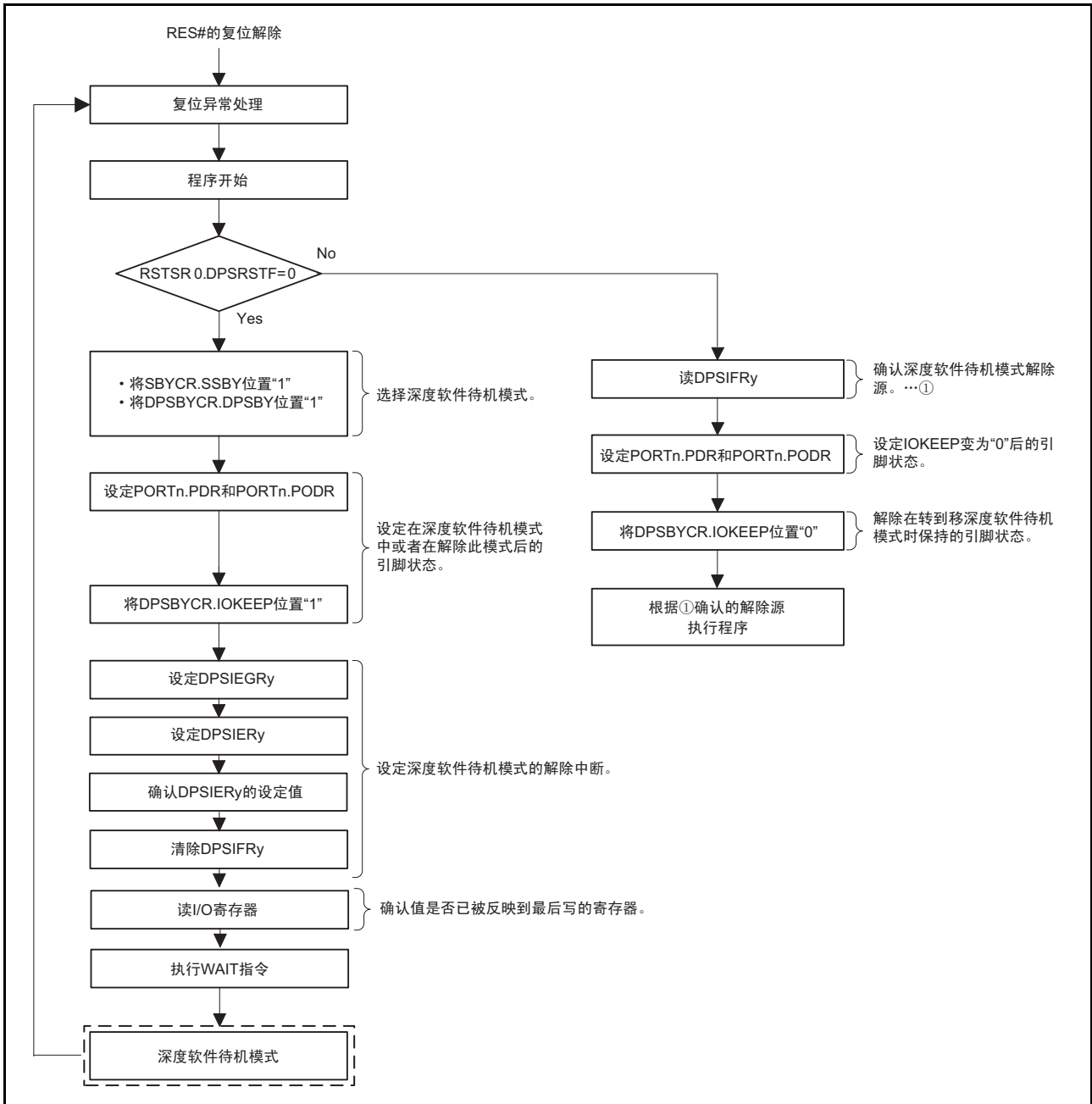


图 11.4 深度软件待机模式的流程图例子

11.6 使用时的注意事项

11.6.1 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。如果输入 High 电平，就无法降低输出电流的消耗电流。

11.6.2 DMAC 和 DTC 的模块停止

必须在将 MSTPCRA.MSTPA28 位置“1”前将 DMAC 的 DMAST.DMST 位和 DTC 的 DTCST.DTCST 位置“0”，并且将 DTC 和 DMAC 设定为未启动的状态。

详细内容请参照“17. DMA 控制器 (DMACA)”和“18. 数据传送控制器 (DTCa)”。

11.6.3 内部外围模块的中断

此中断不能在模块停止状态下运行。如果在发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，然后设定为模块停止状态。

11.6.4 MSTPCRA、MSTPCRB 和 MSTPCRC 寄存器的写操作

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

11.6.5 通过 DIRQnE 位 (n=0 ~ 5) 控制输入缓冲器

能通过将 DPSIERy.DIRQnE (y=0、n=0 ~ 5) 位置“1”，将 IRQ0-DS ~ IRQ5-DS 引脚的输入缓冲器置为有效。因此必须注意：这些引脚的输入被传送到 DPSIFRy.DIRQnF (y=0、n=0 ~ 5) 位而不被传送到中断控制器、外围模块和 I/O 端口。

11.6.6 WAIT 指令的执行时序

如果不等到先执行的寄存器的写操作结束而执行 WAIT 指令，就可能在通过写寄存器进行的设定更改被反映前执行 WAIT 指令，导致意想不到的运行。因此必须在确认已写完最后的寄存器后执行 WAIT 指令。

11.6.7 睡眠模式中通过 DMAC 和 DTC 进行的寄存器改写

在睡眠模式中，WDT 停止运行。不能在睡眠模式中通过 DMAC 和 DTC 改写 WDT 的相关寄存器。

在睡眠模式中，通过设定 OFS0.IWDTSLCSTP 位和 IWDTCSLTPR.SLCSTP 位来停止 IWDT 的运行。此时，不能在睡眠模式中通过 DMAC 和 DTC 改写 IWDT 的相关寄存器。

11.6.8 从软件待机模式返回时的注意事项

在没有将 IRQ0 ~ IRQ5 中断设定为软件待机模式的接触源时，由于软件待机模式中的对应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，根据引脚的状态，中断状态标志 (ICU.IRi.IR) 有可能因转移到软件待机模式而变为“1”。

在转移到软件待机模式时，通过 IERi.IENj 位禁止不作为解除源的 IRQ0 ~ IRQ5 中断后，必须执行 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。

12. 寄存器写保护功能

寄存器写保护功能是在程序失控时保护重要的寄存器不被改写的功能。通过保护寄存器（PRCR）设定要保护的寄存器。

PRCR 寄存器和被保护的寄存器的对应如表 12.1 所示。

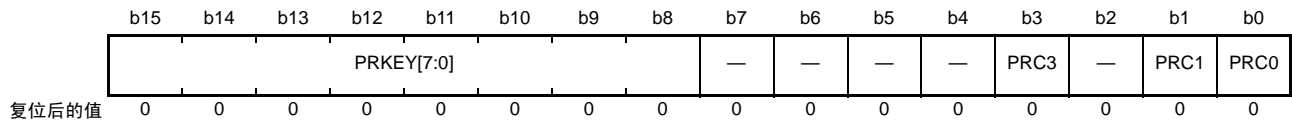
表 12.1 PRCR 寄存器和被保护的寄存器的对应

PRCR 寄存器	被保护的寄存器
PRC0 位	<ul style="list-style-type: none"> • 时钟发生电路的相关寄存器 SCKCR、SCKCR3、PLLCR、PLLCR2、MOSCCR、LOCOCR、ILOCOCR、OSTDCR、OSTDSR
PRC1 位	<ul style="list-style-type: none"> • 运行模式的相关寄存器 SYSCR0、SYSCR1 • 低功耗功能的相关寄存器 SBYCR、MSTPCRA、MSTPCRB、MSTPCRC、MOSCWTCR、PLLWTCR、DPSBYCR、DPSIER0、DPSIER2、DPSIFR0、DPSIFR2、DPSIEGR0、DPSIEGR2 • 时钟发生电路的相关寄存器 MOFCR • 软件复位寄存器 SWRR
PRC3 位	<ul style="list-style-type: none"> • LVD 的相关寄存器 LVCMPCR、LVDLVLRL、LVD1CR0、LVD1CR1、LVD1SR、LVD2CR0、LVD2CR1、LVD2SR

12.1 寄存器说明

12.1.1 保护寄存器 (PRCR)

地址 0008 03FEh



位	符号	位名	功能	R/W
b0	PRC0	保护位 0	允许写时钟发生电路的相关寄存器。 0: 禁止写 1: 允许写	R/W
b1	PRC1	保护位 1	允许写运行模式、低功耗功能和软件复位的相关寄存器。 0: 禁止写 1: 允许写	R/W
b2	—	保留位	读写值都为“0”。	R/W
b3	PRC3	保护位 3	允许写 LVD 的相关寄存器。 0: 禁止写 1: 允许写	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W
b15-b8	PRKEY[7:0]	PRC 键码位	控制是否能改写 PRCR 寄存器。 在改写 PRCR 寄存器时，以 16 位为单位给高位 8 位写“A5h”，给低位 8 位写任意的值。	R/(W) (注1)

注 1. 不保持写数据。

PRCi 位 (保护位 i) (i=0、1、3)

此位选择允许或者禁止写被保护的寄存器。

当 PRCi 位为“1”时，能写被保护的寄存器；当 PRCi 位为“0”时，不能写被保护的寄存器。

13. 异常处理

13.1 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，异常事件的种类如图 13.1 所示。

如果发生异常，处理器模式就转移到管理模式。

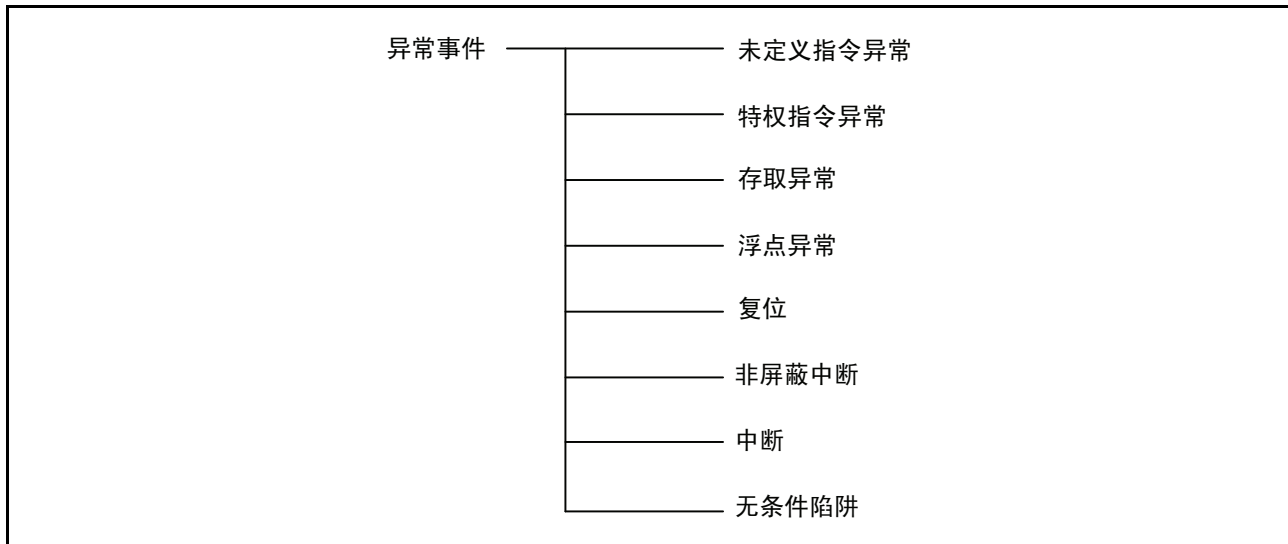


图 13.1 异常事件的种类

13.1.1 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。

13.1.2 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。

13.1.3 存取异常

在检测到因 CPU 进行存储器存取而产生的错误时发生存取异常。在存储器保护单元检测到指令存储器保护错误时发生指令存取异常；在存储器保护单元检测到数据存储器保护错误时发生操作数存取异常。

13.1.4 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。

13.1.5 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。

13.1.6 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。

13.1.7 中断

在给 CPU 输入中断信号时发生中断。能将中断中的 1 个中断源分配为高速中断，高速中断的硬件预处理和硬件后处理比一般中断快，并且高速中断的优先级为 15（最高）。

当 PSW 的 I 位为“0”时，禁止接受中断。

13.1.8 无条件陷阱

如果执行 INT 指令和 BRK 指令，就产生无条件陷阱。

13.2 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 13.2 所示。

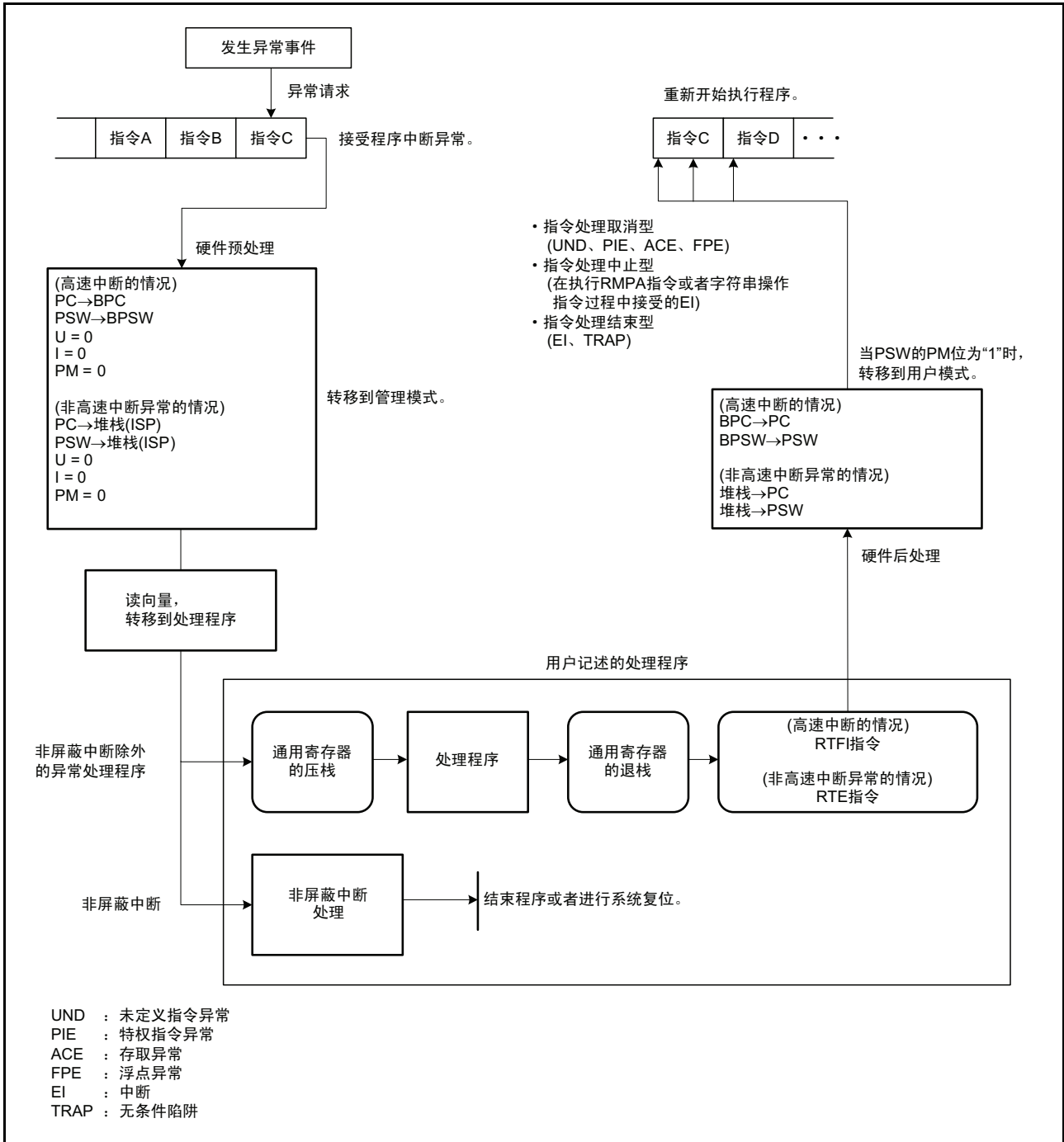


图 13.2 异常处理步骤的概要

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 保存到堆栈区。

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，从堆栈区恢复 PC 和 PSW 的值。

13.3 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。

13.3.1 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 13.1 所示。

表 13.1 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
存取异常		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

13.3.2 向量和 PC、PSW 的保存位置

各异常事件的向量、程序计数器（PC）和处理器状态字（PSW）的保存位置如表 13.2 所示。

表 13.2 向量和 PC、PSW 的保存位置

异常事件		向量	PC 和 PSW 的保存位置
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
存取异常		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
	非高速中断	可向量量表（INTB）	堆栈
无条件陷阱		可向量量表（INTB）	堆栈

13.4 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

（非高速中断异常的情况）

PSW→堆栈区

注. 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

（非高速中断异常的情况）

PC→堆栈区

(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

（非高速中断异常的情况）

堆栈区 →PSW

(b) PC 的恢复

（高速中断的情况）

BPC→PC

（非高速中断异常的情况）

堆栈区 →PC

13.5 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。

13.5.1 未定义指令异常

1. 将处理器状态字（PSW）的内容保存到堆栈区（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）的内容保存到堆栈区（ISP）。
4. 从地址FFFFFFDCh取向量。
5. 将取到的向量设定到PC后转移到异常处理程序。

13.5.2 特权指令异常

1. 将处理器状态字（PSW）的内容保存到堆栈区（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）的内容保存到堆栈区（ISP）。
4. 从地址FFFFFFD0h取向量。
5. 将取到的向量设定到PC后转移到异常处理程序。

13.5.3 存取异常

1. 将处理器状态字（PSW）的内容保存到堆栈区（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）的内容保存到堆栈区（ISP）。
4. 从地址FFFFFFD4h取向量。
5. 将取到的向量设定到PC后转移到异常处理程序。

13.5.4 浮点异常

1. 将处理器状态字（PSW）的内容保存到堆栈区（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）的内容保存到堆栈区（ISP）。
4. 从地址FFFFFFE4h取向量。
5. 将取到的向量设定到PC后转移到异常处理程序。

13.5.5 复位

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

13.5.6 非屏蔽中断

1. 将处理器状态字 (PSW) 的内容保存到堆栈区 (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) 内容保存到堆栈区 (ISP); 而在其他状态下, 将下一条指令的PC内容保存到堆栈区 (ISP)。
4. 将PSW的处理器中断优先级 (IPL[3:0]) 置“Fh”。
5. 从地址FFFFFFF8h取向量。
6. 将取到的向量设定到PC后转移到异常处理程序。

13.5.7 中断

1. 将处理器状态字 (PSW) 的内容保存到堆栈区 (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) 内容保存到堆栈区 (ISP); 而在其他状态下, 将下一条指令的PC内容保存到堆栈区 (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[3:0]) 设定已接受中断的中断优先级。
5. 从可变向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。

13.5.8 无条件陷阱

1. 将处理器状态字 (PSW) 的内容保存到堆栈区 (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) 内容保存到堆栈区 (ISP)。
4. 在使用INT指令时, 从可变向量表取对应INT指令号的向量。
在使用BRK指令时, 从可变向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。

13.6 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 13.3 所示的指令，就恢复异常处理程序前保存到堆栈区或者控制寄存器（BPC、BPSW）的程序计数器（PC）和处理器状态字（PSW）的内容。


表 13.3 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
存取异常		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
	非高速中断	RTE
无条件陷阱		RTE

13.7 异常事件的优先级

异常事件的优先级如表 13.4 所示。如果同时发生多个异常，就先接受优先级高的事件。

表 13.4 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 存取异常
	5 未定义指令异常 特权指令异常
	6 无条件陷阱
	7 操作数存取异常
	8 浮点异常

14. 中断控制器 (ICUb)

14.1 概要

中断控制器接受外围模块中断和外部引脚中断，向 CPU 请求中断以及启动 DTC 和 DMAC。
中断控制器的规格和框图分别如表 14.1 和图 14.1 所示。

表 14.1 中断控制器的规格

项目		内容
中断	外围功能中断	<ul style="list-style-type: none"> • 外围模块的中断 • 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是固定的。 • 组中断功能：将多个中断分配给 1 个中断向量。 电平检测中断组数：1 组（组 12）
	外部引脚中断	<ul style="list-style-type: none"> • IRQ0 ~ IRQ5 引脚的中断 • 中断源数：6 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。 • 数字滤波器功能：有
	软件中断	<ul style="list-style-type: none"> • 通过写寄存器来产生中断。 • 中断源数：1 个
	中断优先级	通过寄存器设定优先级。
	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（注 1）。
非屏蔽中断	NMI 引脚中断	<ul style="list-style-type: none"> • NMI 引脚的中断 • 中断检测：下降沿、上升沿 • 数字滤波器功能：有
	振荡停止检测中断	检测到振荡停止时的中断
	WDT 下溢 / 刷新错误	递减计数器发生下溢时，或者发生刷新错误时的中断
	IWDT 下溢 / 刷新错误	递减计数器发生下溢时，或者发生刷新错误时的中断
	电压监视 1 中断	电压检测电路 1（LVD1）的电压监视中断
	电压监视 2 中断	电压检测电路 2（LVD2）的电压监视中断
从低功耗状态的返回	<ul style="list-style-type: none"> • 睡眠模式：通过非屏蔽中断或者全部中断源返回。 • 全模块时钟停止模式：通过非屏蔽中断、IRQ0 ~ IRQ5 中断返回。 • 软件待机模式：通过非屏蔽中断、IRQ0 ~ IRQ5 中断返回。 	

注 1. 有关 DTC 和 DMAC 的启动源，请参照“表 14.3 中断向量表”。

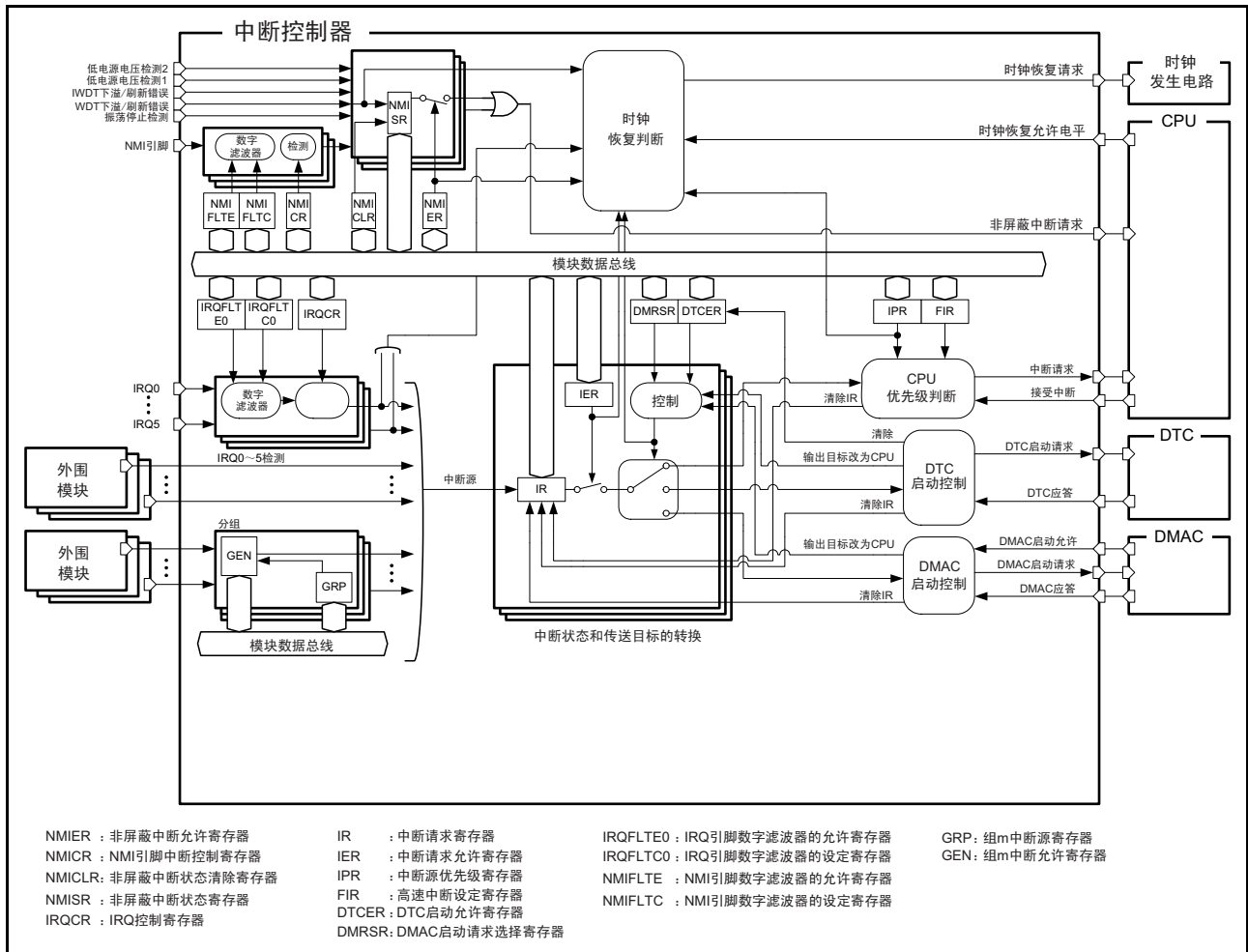


图 14.1 中断控制器的框图

中断控制器使用的输入 / 输出引脚如表 14.2 所示。

表 14.2 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
IRQ0 ~ IRQ5	输入	外部中断请求引脚

14.2 寄存器说明

14.2.1 中断请求寄存器 n (IRn) (n= 中断向量号)

地址 0008 7010h ~ 0008 70FCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IR
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	IR	中断状态标志	0: 没有中断请求 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. 在边沿检测源的情况下，只能写“0”而不能写“1”。
在电平检测源的情况下，不能写此位。

各中断源都有 IRn 寄存器，n 对应中断向量号。
中断源和中断向量号的对应请参照“表 14.3 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。如果发生中断请求，此标志就变为“1”。为了检测中断请求，需要通过外围模块的中断允许位允许中断请求的输出。

中断请求的检测方法有边沿检测和电平检测。对于外围模块中断，给各中断源规定了边沿检测或者电平检测。对于 IRQi 引脚中断，能通过设定 IRQCRi.IRQMD[1:0] 位 (i=0 ~ 5)，选择边沿检测或者电平检测。有关各中断源的检测方法，请参照“表 14.3 中断向量表”。

通过组 m 中断源寄存器 (GRPm) (m= 组号) 检测被分组的中断请求。GRPm 寄存器检测到的中断请求作为电平中断请求，通过该组对应的 IR 标志进行检测。有关中断分组功能的详细内容，请参照“14.4 外围模块中断请求组的功能”。

被分组的中断请求请参照“表 14.4 组 12 的中断请求”。

(1) 边沿检测的情况

[为“1”的条件]

- 如果发生外围模块中断请求或者 IRQi 引脚中断请求，此标志就变为“1”。有关各外围模块中断请求的发生，请参照各外围模块的章节。

[为“0”的条件]

- 如果中断请求目标接受中断请求，此标志就变为“0”。
- 如果给 IR 标志写“0”，此标志就变为“0”。但是，在将中断请求目标设定为 DTC 或者 DMAC 时，禁止给 IR 标志写“0”。

(2) 电平检测的情况

[为“1”的条件]

- 在发生外围模块中断请求或者 IRQi 引脚中断请求的期间，此标志为“1”。有关各外围模块中断请求的发生，请参照各外围模块的章节。
- 在被分组的中断请求的情况下，当组 m 中断允许寄存器的中断请求允许位 j (GENm.ENj) (m= 组号、j= 位号) 为“1”并且组 m 中断源寄存器的中断状态标志 (GRPm.ISj) 为“1”时，此标志为“1”。

[为“0”的条件]

- 如果清除中断请求的输出源，此标志就变为“0”（即使中断请求目标接受中断请求，此标志也不变为“0”）。有关各外围模块中断请求的清除，请参照各外围模块的章节。
- 在被分组的 interrupt 请求的情况下，如果 GENm.ENj 位为“0”或者 GRPm.ISj 标志为“0”，此标志就变为“0”。

在通过电平检测使用 IRQi 引脚的情况下，要取消中断时，必须将 IRQi 引脚置为 High 电平。
在电平检测的情况下，禁止给 IR 标志写“0”和“1”。

14.2.2 中断请求允许寄存器 m (IERm) (m=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
b0	IEN0	中断请求允许位 0	0: 禁止中断请求 1: 允许中断请求	R/W
b1	IEN1	中断请求允许位 1		R/W
b2	IEN2	中断请求允许位 2		R/W
b3	IEN3	中断请求允许位 3		R/W
b4	IEN4	中断请求允许位 4		R/W
b5	IEN5	中断请求允许位 5		R/W
b6	IEN6	中断请求允许位 6		R/W
b7	IEN7	中断请求允许位 7		R/W

注. 被保留的向量号的对应位的读写值都为“0”。

IENj 位 (中断请求允许位 j) (j = 7 ~ 0)

当 IENj 位为“1”时，将中断请求输出到中断请求目标。

当 IENj 位为“0”时，不将中断请求输出到中断请求目标。

IRn.IR 标志不受 IENj 位的影响。即使 IENj 位为“0”，在“14.2.1 中断请求寄存器 n (IRn) (n= 中断向量号)”所示的条件下，IR 标志也发生变化。

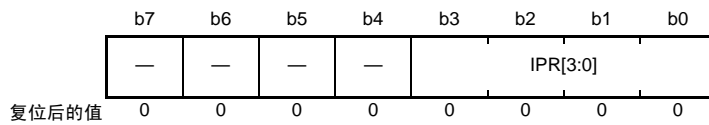
各中断源 (向量号) 都有 IERm.IENj 位。

中断源和 IERm.IENj 位的对应请参照“表 14.3 中断向量表”。

有关选择中断请求目标时的 IERm.IENj 位的设定步骤，请参照“14.5.3 中断请求目标的选择”。

14.2.3 中断源优先级寄存器 n (IPRn) (n=000 ~ 250)

地址 0008 7300h ~ 0008 73FAh



位	符号	位名	功能	R/W
b3-b0	IPR[3:0]	中断优先级设定位	b3 b0 0000: 0级 (禁止中断) (注1) 0001: 1级 0010: 2级 0011: 3级 0100: 4级 0101: 5级 0110: 6级 0111: 7级 1000: 8级 1001: 9级 1010: 10级 1011: 11级 1100: 12级 1101: 13级 1110: 14级 1111: 15级 (最高)	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

注 1. 在设定为高速中断的情况下, 即使为 0 级, 也能发行中断。

中断源和 IPRn 寄存器的对应请参照“表 14.3 中断向量表”。

IPR[3:0] 位 (中断优先级设定位)

这些位选择对应的中断源优先级。

IPR[3:0] 位选择的优先级只在判断向 CPU 请求的中断优先级时被参照, 不影响 DTC 和 DMAC 的启动请求。

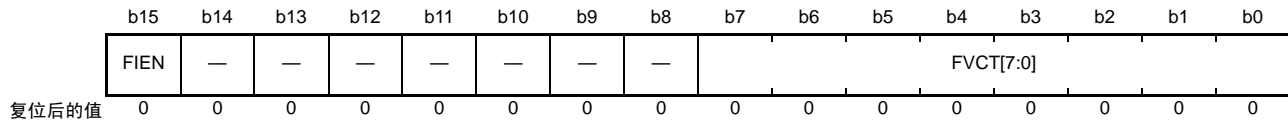
CPU 只接受优先级高于 PSW.IPL[3:0] 位所示优先级的中断请求, 进行中断处理。

如果同时发生多个中断请求, 就通过 IPR[3:0] 位的设定值进行优先级的比较。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源。

必须在禁止中断请求 (IERm.IENj 位 =0) 的状态下写这些位。

14.2.4 高速中断设定寄存器 (FIR)

地址 0008 72F0h



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

通过设定 FIR 寄存器来实现的高速化功能只对向 CPU 请求的中断有效，不影响 DTC 和 DMAC 的启动请求。必须在禁止中断请求 (IERm.IENj 位 =0) 的状态下写此寄存器。

FVCT[7:0] 位 (高速中断向量设定位)

这些位指定要使用高速中断功能的中断向量号。

FIEN 位 (高速中断允许位)

此位是允许高速中断的位。

如果将 FIEN 位置“1”，FVCT[7:0] 位所设向量号的中断就变为高速中断。

当 FIEN 位为“1”时，如果中断请求目标为 CPU 并且发生 FVCT[7:0] 位所指向量号的中断请求，就作为高速中断向 CPU 输出中断请求，与 IPRn 寄存器的设定无关。有关将高速中断用于从软件待机模式的返回，请参照“14.7.3 从软件待机模式的返回”。

对于通过 IERm.IENj 位 (m=02h ~ 1Fh、j=7 ~ 0) 禁止中断请求的中断源，不将中断请求输出到 CPU。能设定的向量号请参照“表 14.3 中断向量表”。

FVCT[7:0] 位不能指定被保留的向量号。

高速中断的详细内容请参照“13. 异常处理”和“14.5.5 高速中断”。

14.2.5 软件中断启动寄存器 (SWINTR)

地址 0008 72E0h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	SWINT
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	SWINT	软件中断启动位	读取值为“0”。通过写“1”，发行软件中断请求。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”。

SWINT 位 (软件中断启动位)

如果给 SWINT 位写“1”，中断请求寄存器 027 (IR027) 就变为“1”。

如果在将 DTC 启动允许寄存器 027 (DTCER027) 置“0”后给 SWINT 位写“1”，就向 CPU 请求中断。

如果在将 DTC 启动允许寄存器 027 (DTCER027) 置“1”后给 SWINT 位写“1”，就发行 DTC 启动请求。

14.2.6 DTC 启动允许寄存器 n (DTCERn) (n= 中断向量号)

地址 0008 711Bh ~ 0008 71FBh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	DTCE
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	DTCE	DTC 启动允许位	0: 禁止启动 DTC 1: 允许启动 DTC	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

对于和 DMAC 启动请求选择的中断源相同的中断源，禁止设定为允许启动 DTC。有关与中断源的对应，请参照“表 14.3 中断向量表”。

DTCE 位 (DTC 启动允许位)

如果将 DTCE 位置“1”，就选择对应的中断源作为 DTC 启动源。

[为“1”的条件]

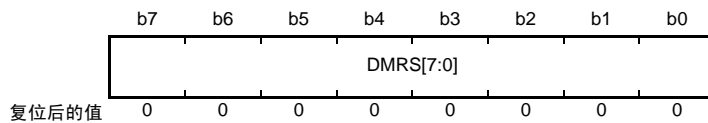
- 当给 DTCE 位写“1”时

[为“0”的条件]

- 当指定次数的数据传送结束时 (在链传送的情况下，当最后链传送的指定次数的数据传送结束时)
- 当给 DTCE 位写“0”时

14.2.7 DMAC 启动请求选择寄存器 m (DMRSRm) (m=DMAC 通道号)

地址 DMRSR0 0008 7400h、DMRSR1 0008 7404h、DMRSR2 0008 7408h、DMRSR3 0008 740Ch



位	符号	位名	功能	R/W
b7-b0	DMRS[7:0]	DMAC 启动源选择位	设定 DMAC 的启动请求向量号。	R/W

禁止给多个 DMRSRm 寄存器设定相同的中断源。对于和 DMRSRm 寄存器设定的中断源相同的中断源，禁止设定为允许启动 DTC。如果违反这些禁止事项，就不保证运行。

DMRS[7:0] 位 (DMAC 启动源选择位)

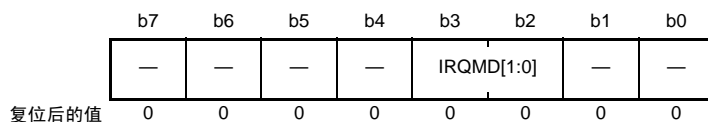
用 8 位指定要启动 DMAC 的中断源的向量号。不能设定未分配为 DMAC 的启动源的向量号。

中断源的向量号请参照“表 14.3 中断向量表”。

必须在 DMA 传送允许寄存器的 DMA 传送允许位 (DMACm.DMCNT.DTE) 为“0”的状态下写 DMRSRm 寄存器。

14.2.8 IRQ 控制寄存器 i (IRQCRi) (i=0 ~ 5)

地址 0008 7500h ~ 0008 7505h



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

必须在对应的中断请求允许位为禁止中断请求 (IERm.IENj 位为“0”) 的状态下更改此寄存器的设定。在更改寄存器后，必须清除 IR 标志，然后将中断请求允许位设定为允许。要更改为 Low 电平时，不需要清除 IR 标志。

IRQMD[1:0] 位 (IRQ 检测设定位)

这些位设定外部引脚中断源 (IRQ0 ~ IRQ5) 的检测方法。

外部引脚中断的检测设定步骤请参照“14.5.7 外部引脚中断”。

14.2.9 IRQ 引脚数字滤波器的允许寄存器 0 (IRQFLTE0)

地址 0008 7510h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	FLTEN5	FLTEN4	FLTEN3	FLTEN2	FLTEN1	FLTEN0
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	FLTEN0	IRQ0 数字滤波器允许位	0: 数字滤波器无效 1: 数字滤波器有效	R/W
b1	FLTEN1	IRQ1 数字滤波器允许位		R/W
b2	FLTEN2	IRQ2 数字滤波器允许位		R/W
b3	FLTEN3	IRQ3 数字滤波器允许位		R/W
b4	FLTEN4	IRQ4 数字滤波器允许位		R/W
b5	FLTEN5	IRQ5 数字滤波器允许位		R/W
b7-b6	—	保留位		读写值都为“0”。

FLTEN_i 位 (IRQ_i 数字滤波器允许位) (i = 0 ~ 5)

此位允许使用外部引脚中断源 (IRQ0 ~ IRQ7) 的数字滤波器。

当 FLTEN_i 位为“1”时，数字滤波器有效；当 FLTEN_i 位为“0”时，数字滤波器功能无效。

在 IRQFLTC0.FCLKSEL_i[1:0] 位设定的每个采样时钟对 IRQ_i 引脚电平进行采样，当电平 3 次相同时，更改数字滤波器的输出电平。

数字滤波器的详细内容请参照“14.5.6 数字滤波器”。

14.2.10 IRQ 引脚数字滤波器的设定寄存器 0 (IRQFLTC0)

地址 0008 7514h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	FCLKSEL5[1:0]	FCLKSEL4[1:0]	FCLKSEL3[1:0]	FCLKSEL2[1:0]	FCLKSEL1[1:0]	FCLKSEL0[1:0]						
0	0	0	0	0	0	0	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	FCLKSEL0[1:0]	IRQ0 数字滤波器的采样时钟设定位	0 0: PCLK	R/W
b3-b2	FCLKSEL1[1:0]	IRQ1 数字滤波器的采样时钟设定位	0 1: PCLK/8	R/W
b5-b4	FCLKSEL2[1:0]	IRQ2 数字滤波器的采样时钟设定位	1 0: PCLK/32	R/W
b7-b6	FCLKSEL3[1:0]	IRQ3 数字滤波器的采样时钟设定位	1 1: PCLK/64	R/W
b9-b8	FCLKSEL4[1:0]	IRQ4 数字滤波器的采样时钟设定位		R/W
b11-b10	FCLKSEL5[1:0]	IRQ5 数字滤波器的采样时钟设定位		R/W
b15-b12	—	保留位	读写值都为“0”。	R/W

FCLKSELi[1:0] 位 (IRQi 数字滤波器的采样时钟设定位) (i = 0 ~ 5)

这些位选择外部引脚中断请求引脚 (IRQ0 ~ IRQ5) 的数字滤波器的采样时钟。

从 PCLK (每个时钟)、PCLK/8 (每 8 个时钟采样 1 次)、PCLK/32 (每 32 个时钟采样 1 次) 或者 PCLK/64 (每 64 个时钟采样 1 次) 中选择采样时钟。

数字滤波器的详细内容请参照“14.5.6 数字滤波器”。

14.2.11 非屏蔽中断状态寄存器 (NMISR)

地址 0008 7580h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	LVD2ST	LVD1ST	IWDTST	WDTST	OSTST	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: 没有 NMI 引脚中断请求 1: 有 NMI 引脚中断请求	R
b1	OSTST	振荡停止检测中断状态标志	0: 没有振荡停止检测中断请求 1: 有振荡停止检测中断请求	R
b2	WDTST	WDT 下溢 / 刷新错误状态标志	0: 没有 WDT 下溢 / 刷新错误中断请求 1: 有 WDT 下溢 / 刷新错误中断请求	R
b3	IWDTST	IWDT 下溢 / 刷新错误状态标志	0: 没有 IWDT 下溢 / 刷新错误中断请求 1: 有 IWDT 下溢 / 刷新错误中断请求	R
b4	LVD1ST	电压监视 1 中断状态标志	0: 没有电压监视 1 中断请求 1: 有电压监视 1 中断请求	R
b5	LVD2ST	电压监视 2 中断状态标志	0: 没有电压监视 2 中断请求 1: 有电压监视 2 中断请求	R
b7-b6	—	保留位	读取值为“0”，写操作无效。	R

NMISR 寄存器是监视非屏蔽中断源状态的寄存器。忽视对 NMISR 寄存器的写操作。

非屏蔽中断允许寄存器 (NMIER) 的设定不影响这些状态标志。

必须在非屏蔽中断处理程序结束前读 NMISR 寄存器并且确认其他非屏蔽中断的发生状况。必须在确认 NMISR 寄存器的全部位都是“0”后结束处理程序。

NMIST 标志 (NMI 状态标志)

此标志表示 NMI 引脚中断请求。

只能读 NMIST 标志，通过 NMICLR.NMICLR 位清除此标志。

[为“1”的条件]

- 当将 NMICR.NMIMD 位设定的边沿输入到 NMI 引脚时

[为“0”的条件]

- 当给 NMICLR.NMICLR 位写“1”时

OSTST 标志 (振荡停止检测中断状态标志)

此标志表示振荡停止检测中断请求。

只能读 OSTST 标志，通过 NMICLR.OSTCLR 位清除此标志。

[为“1”的条件]

- 当发生振荡停止检测中断时

[为“0”的条件]

- 当给 NMICLR.OSTCLR 位写“1”时

WDTST 标志 (WDT 下溢 / 刷新错误状态标志)

此标志表示 WDT 下溢 / 刷新错误中断请求。

只能读 WDTST 标志，通过 NMICLR.WDTCLR 位清除此标志。

[为“1”的条件]

- 当发生 WDT 下溢 / 刷新错误中断时

[为“0”的条件]

- 当给 NMICLR.WDTCLR 位写“1”时

IWDTST 标志 (IWDT 下溢 / 刷新错误状态标志)

此标志表示 IWDT 下溢 / 刷新错误中断请求。

只能读 IWDTST 标志，通过 NMICLR.IWDTCLR 位清除此标志。

[为“1”的条件]

- 在发生源为允许发生中断的情况下产生 IWDT 下溢 / 刷新错误中断时

[为“0”的条件]

- 当给 NMICLR.IWDTCLR 位写“1”时

LVD1ST 标志 (电压监视 1 中断状态标志)

此标志表示电压监视 1 中断请求。

只能读 LVD1ST 标志，通过 NMICLR.LVD1CLR 位清除此标志。

[为“1”的条件]

- 在发生源为允许发生中断的情况下产生电压监视 1 中断时

[为“0”的条件]

- 当给 NMICLR.LVD1CLR 位写“1”时

LVD2ST 标志 (电压监视 2 中断状态标志)

此标志表示电压监视 2 中断请求。

只能读 LVD2ST 标志，通过 NMICLR.LVD2CLR 位清除此标志。

[为“1”的条件]

- 在发生源为允许发生中断的情况下产生电压监视 2 中断时

[为“0”的条件]

- 当给 NMICLR.LVD2CLR 位写“1”时

14.2.12 非屏蔽中断允许寄存器 (NMIER)

地址 0008 7581h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	LVD2EN	LVD1EN	IWDTEN	WDTEN	OSTEN	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 引脚中断允许位	0: 禁止 NMI 引脚中断 1: 允许 NMI 引脚中断	R/(W) (注 1)
b1	OSTEN	振荡停止检测中断允许位	0: 禁止振荡停止检测中断 1: 允许振荡停止检测中断	R/(W) (注 1)
b2	WDTEN	WDT 下溢 / 刷新错误允许位	0: 禁止 WDT 下溢 / 刷新错误中断 1: 允许 WDT 下溢 / 刷新错误中断	R/(W) (注 1)
b3	IWDTEN	IWDT 下溢 / 刷新错误允许位	0: 禁止 IWDT 下溢 / 刷新错误中断 1: 允许 IWDT 下溢 / 刷新错误中断	R/(W) (注 1)
b4	LVD1EN	电压监视 1 中断允许位	0: 禁止电压监视 1 中断 1: 允许电压监视 1 中断	R/(W) (注 1)
b5	LVD2EN	电压监视 2 中断允许位	0: 禁止电压监视 2 中断 1: 允许电压监视 2 中断	R/(W) (注 1)
b7-b6	—	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。

NMIEN 位 (NMI 引脚中断允许位)

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”。

OSTEN 位 (振荡停止检测中断允许位)

此位是允许使用振荡停止检测中断的位。
只能写 1 次“1”，以后的写操作无效。
不能写“0”。

WDTEN 位 (WDT 下溢 / 刷新错误允许位)

此位是允许使用 WDT 下溢 / 刷新错误中断的位。
只能写 1 次“1”，以后的写操作无效。
不能写“0”。

IWDTEN 位 (IWDT 下溢 / 刷新错误允许位)

此位是允许使用 IWDT 下溢 / 刷新错误中断的位。
只能写 1 次“1”，以后的写操作无效。
不能写“0”。

LVD1EN 位 (电压监视 1 中断允许位)

此位是允许使用电压监视 1 中断的位。
只能写 1 次“1”，以后的写操作无效。
不能写“0”。

LVD2EN 位 (电压监视 2 中断允许位)

此位是允许使用电压监视 2 中断的位。
只能写 1 次“1”，以后的写操作无效。
不能写“0”。

14.2.13 非屏蔽中断状态清除寄存器 (NMICLR)

地址 0008 7582h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2C LR	LVD1C LR	IWDTC LR	WDTC LR	OSTCLR	NMICLR
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	NMICLR	NMI 清除位	读取值为“0”。通过写“1”，清除 NMISR.NMIST 标志。写“0”无效。	R/(W) (注1)
b1	OSTCLR	OST 清除位	读取值为“0”。通过写“1”，清除 NMISR.OSTST 标志。写“0”无效。	R/(W) (注1)
b2	WDTCLR	WDT 清除位	读取值为“0”。通过写“1”，清除 NMISR.WDTST 标志。写“0”无效。	R/(W) (注1)
b3	IWDTCLR	IWDT 清除位	读取值为“0”。通过写“1”，清除 NMISR.IWDTST 标志。写“0”无效。	R/(W) (注1)
b4	LVD1CLR	LVD1 清除位	读取值为“0”。通过写“1”，清除 NMISR.LVD1ST 标志。写“0”无效。	R/(W) (注1)
b5	LVD2CLR	LVD2 清除位	读取值为“0”。通过写“1”，清除 NMISR.LVD2ST 标志。写“0”无效。	R/(W) (注1)
b7-b6	—	保留位	读写值都为“0”。	R/W

注 1. 只能写“1”。

NMICLR 位 (NMI 清除位)

如果写“1”，NMISR.NMIST 标志就变为“0”。读取值为“0”。

OSTCLR 位 (OST 清除位)

如果写“1”，NMISR.OSTST 标志就变为“0”。读取值为“0”。

WDTCLR 位 (WDT 清除位)

如果写“1”，NMISR.WDTST 标志就变为“0”。读取值为“0”。

IWDTCLR 位 (IWDT 清除位)

如果写“1”，NMISR.IWDTST 标志就变为“0”。读取值为“0”。

LVD1CLR 位 (LVD1 清除位)

如果写“1”，NMISR.LVD1ST 标志就变为“0”。读取值为“0”。

LVD2CLR 位 (LVD2 清除位)

如果写“1”，NMISR.LVD2ST 标志就变为“0”。读取值为“0”。

14.2.14 NMI 引脚中断控制寄存器 (NMICR)

地址 0008 7583h

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

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

14.2.15 NMI 引脚数字滤波器的允许寄存器 (NMIFLTE)

地址 0008 7590h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	NFLTEN
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	NFLTEN	NMI 数字滤波器允许位	0: 数字滤波器无效 1: 数字滤波器有效	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

NFLTEN 位 (NMI 数字滤波器允许位)

此位允许使用 NMI 引脚中断的数字滤波器。

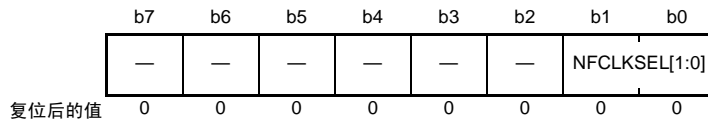
当 NFLTEN 位为“1”时，数字滤波器有效；当 NFLTEN 位为“0”时，数字滤波器功能无效。

在 NMIFLTC.NFCLKSEL[1:0] 位设定的每个采样时钟对 NMI 引脚电平进行采样，当电平 3 次相同时，更改数字滤波器的输出电平。

数字滤波器的详细内容请参照“14.5.6 数字滤波器”。

14.2.16 NMI 引脚数字滤波器的设定寄存器 (NMIFLTC)

地址 0008 7594h



位	符号	位名	功能	R/W
b1-b0	NFCLKSEL[1:0]	NMI 数字滤波器的采样时钟设定位	b1 b0 0 0: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/64	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

NFCLKSEL[1:0] 位 (NMI 数字滤波器的采样时钟设定位)

这些位选择 NMI 引脚中断的数字滤波器的采样时钟。

从 PCLK (每个时钟)、PCLK/8 (每 8 个时钟采样 1 次)、PCLK/32 (每 32 个时钟采样 1 次) 或者 PCLK/64 (每 64 个时钟采样 1 次) 中选择采样时钟。

数字滤波器的详细内容请参照“14.5.6 数字滤波器”。

14.2.17 组 12 中断源寄存器 (GRP12)

地址 GRP12 0008 C330h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
—	—	—	—	—	—	—	—	—	—	IS5	IS4	—	—	IS1	IS0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	IS0	中断状态标志 0	0: 没有中断请求	R
b1	IS1	中断状态标志 1	1: 有中断请求	R
b3-b2	—	保留位	读取值为“0”，写操作无效。	R
b4	IS4	中断状态标志 4	0: 没有中断请求	R
b5	IS5	中断状态标志 5	1: 有中断请求	R
b31-b6	—	保留位	读取值为“0”，写操作无效。	R

分组的详细内容请参照“14.4.1 中断请求组”。有关分配到各组的 interrupt 请求与 GRP12 寄存器内的位号的对应，请参照“表 14.4 组 12 的 interrupt 请求”，被分组的 interrupt 源与 interrupt 向量号的对应请参照“表 14.3 中断向量表”。

IS_j 标志 (中断状态标志) (j= 位号)

这是分配到组 12 第 j 位的外围模块 interrupt 请求的状态标志。如果发生 interrupt 请求并且通过组 12 中断允许寄存器的 interrupt 请求允许位 j (GEN12.EN_j) (j= 位号) 允许 interrupt，此标志就变为“1”。

当组内的任意一个 interrupt 源的 GEN12.EN_j 位和 IS_j 标志都为“1”时，该组对应的 interrupt 请求寄存器 n 的 interrupt 状态标志 (IR_n.IR) (n= interrupt 向量号) 变为“1”。

电平检测的 interrupt 请求被分在组 12。

[为“1”的条件]

- 在 GEN12.EN_j 位为“1”并且外围模块发生 interrupt 请求的期间，对应 interrupt 请求的 IS_j 标志为“1”。

[为“0”的条件]

- 如果清除 interrupt 请求的输出源，IS_j 标志就变为“0”。
- 当 GEN12.EN_j 位为“0”时，IS_j 标志为“0”。

不能写 IS_j 标志。

14.2.18 组 12 中断允许寄存器 (GEN12)

地址 GEN12 0008 C370h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
—	—	—	—	—	—	—	—	—	—	EN5	EN4	—	—	EN1	EN0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	EN0	中断请求允许位 0	0: 禁止中断请求	R/W
b1	EN1	中断请求允许位 1	1: 允许中断请求	R/W
b3-b2	—	保留位	读写值都为“0”。	R/W
b4	EN4	中断请求允许位 4	0: 禁止中断请求	R/W
b5	EN5	中断请求允许位 5	1: 允许中断请求	R/W
b31-b6	—	保留位	读写值都为“0”。	R/W

分组的详细内容请参照“14.4.1 中断请求组”。

EN_j 位 (中断请求允许位 j) (j= 位号)

当 EN_j 位为“1”时，能检测对应的 GRP12.IS_j 标志 (j= 位号) 的中断请求。如果检测到中断，GRP12.IS_j 标志就变为“1”。

当 EN_j 位为“0”时，不能检测中断请求。

即使给 EN_j 位写“0”，对应的 GRP12.IS_j 标志变为“0”。

14.3 向量表

通过中断控制器检测的异常事件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。

14.3.1 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，并且给 INTB 寄存器设定 4 的倍数。

中断向量表如表 14.3 所示，表 14.3 的各项目的内容如下：

项目	内容
中断请求发生源	表示中断请求发生源的名称。
名称	表示中断名称。
向量号	表示向量号。
向量地址的偏移	表示向量基址偏移值。
中断的检测方法	用“边沿”和“电平”表示中断的检测方法。
CPU 中断	用“○”表示 CPU 中断源。
DTC 启动	用“○”表示 DTC 启动源。
DMAC 启动	用“○”表示 DMAC 启动源。
sstb 返回	用“○”表示从软件待机模式的返回源。
sacs 返回	用“○”表示从全模块时钟停止模式的返回源。
IER	表示对应向量号的 IER 寄存器和位名。
IPR	表示对应中断源的 IPR 寄存器。
DTCER	表示对应 DTC 启动源的 DTCER 寄存器。

表 14.3 中断向量表 (1/6)

中断请求 发生源	名称	向量号 (注1)	向量地址的 偏移	中断的 检测方法	CPU 中断	DTC 启动	DMAC 启动	ssfb 返回	sacs 返回	IER	IPR	DTCER
—	保留	0	0000h	—	X	X	X	X	X	—	—	—
—	保留	1	0004h	—	X	X	X	X	X	—	—	—
—	保留	2	0008h	—	X	X	X	X	X	—	—	—
—	保留	3	000Ch	—	X	X	X	X	X	—	—	—
—	保留	4	0010h	—	X	X	X	X	X	—	—	—
—	保留	5	0014h	—	X	X	X	X	X	—	—	—
—	保留	6	0018h	—	X	X	X	X	X	—	—	—
—	保留	7	001Ch	—	X	X	X	X	X	—	—	—
—	保留	8	0020h	—	X	X	X	X	X	—	—	—
—	保留	9	0024h	—	X	X	X	X	X	—	—	—
—	保留	10	0028h	—	X	X	X	X	X	—	—	—
—	保留	11	002Ch	—	X	X	X	X	X	—	—	—
—	保留	12	0030h	—	X	X	X	X	X	—	—	—
—	保留	13	0034h	—	X	X	X	X	X	—	—	—
—	保留	14	0038h	—	X	X	X	X	X	—	—	—
—	保留	15	003Ch	—	X	X	X	X	X	—	—	—
BSC	BUSERR	16	0040h	电平	○	X	X	X	X	IER02.IEN0	IPR000	—
—	保留	17	0044h	—	X	X	X	X	X	—	—	—
—	保留	18	0048h	—	X	X	X	X	X	—	—	—
—	保留	19	004Ch	—	X	X	X	X	X	—	—	—
—	保留	20	0050h	—	X	X	X	X	X	—	—	—
FCU	FIFERR	21	0054h	电平	○	X	X	X	X	IER02.IEN5	IPR001	—
—	保留	22	0058h	—	X	X	X	X	X	—	—	—
FCU	FRDYI	23	005Ch	边沿	○	X	X	X	X	IER02.IEN7	IPR002	—
—	保留	24	0060h	—	X	X	X	X	X	—	—	—
—	保留	25	0064h	—	X	X	X	X	X	—	—	—
—	保留	26	0068h	—	X	X	X	X	X	—	—	—
ICU	SWINT	27	006Ch	边沿	○	○	X	X	X	IER03.IEN3	IPR003	DTCER027
CMT0	CMIO	28	0070h	边沿	○	○	○	X	X	IER03.IEN4	IPR004	DTCER028
CMT1	CMI1	29	0074h	边沿	○	○	○	X	X	IER03.IEN5	IPR005	DTCER029
CMT2	CMI2	30	0078h	边沿	○	○	○	X	X	IER03.IEN6	IPR006	DTCER030
CMT3	CMI3	31	007Ch	边沿	○	○	○	X	X	IER03.IEN7	IPR007	DTCER031
—	保留	32	0080h	—	X	X	X	X	X	—	—	—
—	保留	33	0084h	—	X	X	X	X	X	—	—	—
—	保留	34	0088h	—	X	X	X	X	X	—	—	—
—	保留	35	008Ch	—	X	X	X	X	X	—	—	—
CAC	FERRF	36	0090h	—	○	X	X	X	X	IER04.IEN4	IPR036	—
	MENDF	37	0094h	—	○	X	X	X	X	IER04.IEN5		—
	OVFF	38	0098h	—	○	X	X	X	X	IER04.IEN6		—
RSPIO	SPRIO	39	009Ch	边沿	○	○	○	X	X	IER04.IEN7	IPR039	DTCER039
	SPTIO	40	00A0h	边沿	○	○	○	X	X	IER05.IEN0	IPR040	DTCER040
	SPIIO	41	00A4h	电平	○	X	X	X	X	IER05.IEN1	IPR041	—
—	保留	42	00A8h	—	X	X	X	X	X	—	—	—
—	保留	43	00ACh	—	X	X	X	X	X	—	—	—
—	保留	44	00B0h	—	X	X	X	X	X	—	—	—
—	保留	45	00B4h	—	X	X	X	X	X	—	—	—
—	保留	46	00B8h	—	X	X	X	X	X	—	—	—

表 14.3 中断向量表 (2/6)

中断请求 发生源	名称	向量号 (注1)	向量地址的 偏移	中断的 检测方法	CPU 中断	DTC 启动	DMAC 启动	sstb 返回	sacs 返回	IER	IPR	DTCER
—	保留	47	00BCh	—	X	X	X	X	X	—	—	—
—	保留	48	00C0h	—	X	X	X	X	X	—	—	—
—	保留	49	00C4h	—	X	X	X	X	X	—	—	—
—	保留	50	00C8h	—	X	X	X	X	X	—	—	—
—	保留	51	00CCh	—	X	X	X	X	X	—	—	—
—	保留	52	00D0h	—	X	X	X	X	X	—	—	—
—	保留	53	00D4h	—	X	X	X	X	X	—	—	—
—	保留	54	00D8h	—	X	X	X	X	X	—	—	—
—	保留	55	00DCh	—	X	X	X	X	X	—	—	—
—	保留	56	00E0h	—	X	X	X	X	X	—	—	—
DOC	DOPCF	57	00E4h	电平	○	X	X	X	X	IER07.IEN1	IPR057	—
—	保留	58	00E8h	—	X	X	X	X	X	—	—	—
—	保留	59	00ECh	—	X	X	X	X	X	—	—	—
—	保留	60	00F0h	—	X	X	X	X	X	—	—	—
—	保留	61	00F4h	—	X	X	X	X	X	—	—	—
—	保留	62	00F8h	—	X	X	X	X	X	—	—	—
—	保留	63	00FCh	—	X	X	X	X	X	—	—	—
外部端子	IRQ0	64	0100h	边沿 / 电平	○	○	○	○	○	IER08.IEN0	IPR064	DTCER064
	IRQ1	65	0104h	边沿 / 电平	○	○	○	○	○	IER08.IEN1	IPR065	DTCER065
	IRQ2	66	0108h	边沿 / 电平	○	○	○	○	○	IER08.IEN2	IPR066	DTCER066
	IRQ3	67	010Ch	边沿 / 电平	○	○	○	○	○	IER08.IEN3	IPR067	DTCER067
	IRQ4	68	0110h	边沿 / 电平	○	○	○	○	○	IER08.IEN4	IPR068	DTCER068
	IRQ5	69	0114h	边沿 / 电平	○	○	○	○	○	IER08.IEN5	IPR069	DTCER069
—	保留	70	0118h	—	X	X	X	X	X	—	—	—
—	保留	71	011Ch	—	X	X	X	X	X	—	—	—
—	保留	72	0120h	—	X	X	X	X	X	—	—	—
—	保留	73	0124h	—	X	X	X	X	X	—	—	—
—	保留	74	0128h	—	X	X	X	X	X	—	—	—
—	保留	75	012Ch	—	X	X	X	X	X	—	—	—
—	保留	76	0130h	—	X	X	X	X	X	—	—	—
—	保留	77	0134h	—	X	X	X	X	X	—	—	—
—	保留	78	0138h	—	X	X	X	X	X	—	—	—
—	保留	79	013Ch	—	X	X	X	X	X	—	—	—
—	保留	80	0140h	—	X	X	X	X	X	—	—	—
—	保留	81	0144h	—	X	X	X	X	X	—	—	—
—	保留	82	0148h	—	X	X	X	X	X	—	—	—
—	保留	83	014Ch	—	X	X	X	X	X	—	—	—
—	保留	84	0150h	—	X	X	X	X	X	—	—	—
—	保留	85	0154h	—	X	X	X	X	X	—	—	—
—	保留	86	0158h	—	X	X	X	X	X	—	—	—
—	保留	87	015Ch	—	X	X	X	X	X	—	—	—
—	保留	88	0160h	—	X	X	X	X	X	—	—	—
—	保留	89	0164h	—	X	X	X	X	X	—	—	—
—	保留	90	0168h	—	X	X	X	X	X	—	—	—
—	保留	91	016Ch	—	X	X	X	X	X	—	—	—
—	保留	92	0170h	—	X	X	X	X	X	—	—	—
—	保留	93	0174h	—	X	X	X	X	X	—	—	—

表 14.3 中断向量表 (3/6)

中断请求 发生源	名称	向量号 (注1)	向量地址的 偏移	中断的 检测方法	CPU 中断	DTC 启动	DMAC 启动	sslb 返回	sacs 返回	IER	IPR	DTCER
—	保留	94	0178h	—	X	X	X	X	X	—	—	—
—	保留	95	017Ch	—	X	X	X	X	X	—	—	—
—	保留	96	0180h	—	X	X	X	X	X	—	—	—
—	保留	97	0184h	—	X	X	X	X	X	—	—	—
—	保留	98	0188h	—	X	X	X	X	X	—	—	—
—	保留	99	018Ch	—	X	X	X	X	X	—	—	—
—	保留	100	0190h	—	X	X	X	X	X	—	—	—
—	保留	101	0194h	—	X	X	X	X	X	—	—	—
S12AD	S12ADI	102	0198h	边沿	○	○	○	X	X	IER0C.IEN6	IPR102	DTCER102
	S12GBADI	103	019Ch	边沿	○	○	○	X	X	IER0C.IEN7	IPR103	DTCER103
—	保留	104	01A0h	—	X	X	X	X	X	—	—	—
—	保留	105	01A4h	—	X	X	X	X	X	—	—	—
—	保留	106	01A8h	—	X	X	X	X	X	—	—	—
—	保留	107	01ACh	—	X	X	X	X	X	—	—	—
—	保留	108	01B0h	—	X	X	X	X	X	—	—	—
—	保留	109	01B4h	—	X	X	X	X	X	—	—	—
—	保留	110	01B8h	—	X	X	X	X	X	—	—	—
—	保留	111	01BCh	—	X	X	X	X	X	—	—	—
—	保留	112	01C0h	—	X	X	X	X	X	—	—	—
—	保留	113	01C4h	—	X	X	X	X	X	—	—	—
ICU	GROUP12	114	01C8h	电平	○	X	X	X	X	IER0E.IEN2	IPR114	—
—	保留	115	01CCh	—	X	X	X	X	X	—	—	—
—	保留	116	01D0h	—	X	X	X	X	X	—	—	—
—	保留	117	01D4h	—	X	X	X	X	X	—	—	—
—	保留	118	01D8h	—	X	X	X	X	X	—	—	—
—	保留	119	01DCh	—	X	X	X	X	X	—	—	—
—	保留	120	01E0h	—	X	X	X	X	X	—	—	—
—	保留	121	01E4h	—	X	X	X	X	X	—	—	—
SCI12	SCIX0	122	01E8h	电平	○	X	X	X	X	IER0F.IEN2	IPR122	—
	SCIX1	123	01ECh	电平	○	X	X	X	X	IER0F.IEN3		—
	SCIX2	124	01F0h	电平	○	X	X	X	X	IER0F.IEN4		—
	SCIX3	125	01F4h	电平	○	X	X	X	X	IER0F.IEN5		—
MTU0	TGIA0	126	01F8h	边沿	○	○	○	X	X	IER0F.IEN6	IPR126	DTCER126
	TGIB0	127	01FCh	边沿	○	○	○	X	X	IER0F.IEN7		DTCER127
	TGIC0	128	0200h	边沿	○	○	○	X	X	IER10.IEN0		DTCER128
	TGID0	129	0204h	边沿	○	○	○	X	X	IER10.IEN1		DTCER129
	TCIV0	130	0208h	边沿	○	X	X	X	X	IER10.IEN2	IPR130	—
	TGIE0	131	020Ch	边沿	○	X	X	X	X	IER10.IEN3		—
	TGIF0	132	0210h	边沿	○	X	X	X	X	IER10.IEN4		—
MTU1	TGIA1	133	0214h	边沿	○	○	○	X	X	IER10.IEN5	IPR133	DTCER133
	TGIB1	134	0218h	边沿	○	○	○	X	X	IER10.IEN6		DTCER134
	TCIV1	135	021Ch	边沿	○	X	X	X	X	IER10.IEN7	IPR135	—
	TCIU1	136	0220h	边沿	○	X	X	X	X	IER11.IEN0		—
MTU2	TGIA2	137	0224h	边沿	○	○	○	X	X	IER11.IEN1	IPR137	DTCER137
	TGIB2	138	0228h	边沿	○	○	○	X	X	IER11.IEN2		DTCER138
	TCIV2	139	022Ch	边沿	○	X	X	X	X	IER11.IEN3	IPR139	—
	TCIU2	140	0230h	边沿	○	X	X	X	X	IER11.IEN4		—

表 14.3 中断向量表 (4/6)

中断请求 发生源	名称	向量号 (注1)	向量地址的 偏移	中断的 检测方法	CPU 中断	DTC 启动	DMAC 启动	sslb 返回	sacs 返回	IER	IPR	DTCER
MTU3	TGIA3	141	0234h	边沿	○	○	○	X	X	IER11.IEN5	IPR141	DTCER141
	TGIB3	142	0238h	边沿	○	○	○	X	X	IER11.IEN6		DTCER142
	TGIC3	143	023Ch	边沿	○	○	○	X	X	IER11.IEN7		DTCER143
	TGID3	144	0240h	边沿	○	○	○	X	X	IER12.IEN0		DTCER144
	TCIV3	145	0244h	边沿	○	X	X	X	X	IER12.IEN1	IPR145	—
MTU4	TGIA4	146	0248h	边沿	○	○	○	X	X	IER12.IEN2	IPR146	DTCER146
	TGIB4	147	024Ch	边沿	○	○	○	X	X	IER12.IEN3		DTCER147
	TGIC4	148	0250h	边沿	○	○	○	X	X	IER12.IEN4		DTCER148
	TGID4	149	0254h	边沿	○	○	○	X	X	IER12.IEN5		DTCER149
	TCIV4	150	0258h	边沿	○	○	○	X	X	IER12.IEN6	IPR150	DTCER150
MTU5	TGIU5	151	025Ch	边沿	○	○	○	X	X	IER12.IEN7	IPR151	DTCER151
	TGIV5	152	0260h	边沿	○	○	○	X	X	IER13.IEN0		DTCER152
	TGIW5	153	0264h	边沿	○	○	○	X	X	IER13.IEN1		DTCER153
MTU6	TGIA6	154	0268h	边沿	○	○	○	X	X	IER13.IEN2	IPR154	DTCER154
	TGIB6	155	026Ch	边沿	○	○	○	X	X	IER13.IEN3		DTCER155
	TGIC6	156	0270h	边沿	○	○	○	X	X	IER13.IEN4		DTCER156
	TGID6	157	0274h	边沿	○	○	○	X	X	IER13.IEN5		DTCER157
	TCIV6	158	0278h	边沿	○	X	X	X	X	IER13.IEN6	IPR158	—
—	保留	159	01CCh	—	X	X	X	X	X	—	—	—
—	保留	160	01D0h	—	X	X	X	X	X	—	—	—
MTU7	TGIA7	161	0284h	边沿	○	○	○	X	X	IER14.IEN1	IPR161	DTCER161
	TGIB7	162	0288h	边沿	○	○	○	X	X	IER14.IEN2		DTCER162
	TGIC7	163	028Ch	边沿	○	○	○	X	X	IER14.IEN3	IPR163	DTCER163
	TGID7	164	0290h	边沿	○	○	○	X	X	IER14.IEN4		DTCER164
	TCIV7	165	0294h	边沿	○	○	○	X	X	IER14.IEN5	IPR165	DTCER165
POE	OEI1	166	0298h	电平	○	X	X	X	X	IER14.IEN6	IPR166	—
	予約	167	029Ch	—	X	X	X	X	X	—		—
	OEI3	168	02A0h	电平	○	X	X	X	X	IER15.IEN0		—
	OEI4	169	02A4h	电平	○	X	X	X	X	IER15.IEN1		—
—	保留	170	02A8h	—	X	X	X	X	X	—	—	—
比较器	CMP0	171	02ACh	边沿	○	○	○	X	X	IER15.IEN3	IPR171	DTCER171
	CMP1	172	02B0h	边沿	○	○	○	X	X	IER15.IEN4	IPR172	DTCER172
	CMP2	173	02B4h	边沿	○	○	○	X	X	IER15.IEN5	IPR173	DTCER173
—	保留	174	02B8h	—	X	X	X	X	X	—	—	—
—	保留	175	02BCh	—	X	X	X	X	X	—	—	—
—	保留	176	02C0h	—	X	X	X	X	X	—	—	—
—	保留	177	02C4h	—	X	X	X	X	X	—	—	—
—	保留	178	02C8h	—	X	X	X	X	X	—	—	—
—	保留	179	02CCh	—	X	X	X	X	X	—	—	—
—	保留	180	02D0h	—	X	X	X	X	X	—	—	—
—	保留	181	02D4h	—	X	X	X	X	X	—	—	—
—	保留	182	02D8h	—	X	X	X	X	X	—	—	—
—	保留	183	02DCh	—	X	X	X	X	X	—	—	—
—	保留	184	02E0h	—	X	X	X	X	X	—	—	—
—	保留	185	02E4h	—	X	X	X	X	X	—	—	—
—	保留	186	02E8h	—	X	X	X	X	X	—	—	—
—	保留	187	02ECh	—	X	X	X	X	X	—	—	—

表 14.3 中断向量表 (5/6)

中断请求发生源	名称	向量号 (注1)	向量地址的 偏移	中断的 检测方法	CPU 中断	DTC 启动	DMAC 启动	sstb 返回	sacs 返回	IER	IPR	DTCER
—	保留	188	02F0h	—	X	X	X	X	X	—	—	—
—	保留	189	02F4h	—	X	X	X	X	X	—	—	—
—	保留	190	02F8h	—	X	X	X	X	X	—	—	—
—	保留	191	02FCh	—	X	X	X	X	X	—	—	—
—	保留	192	0300h	—	X	X	X	X	X	—	—	—
—	保留	193	0304h	—	X	X	X	X	X	—	—	—
RIIC0	EI10	194	0308h	电平	○	X	X	X	X	IER18.IEN2	IPR194	—
	RX10	195	030Ch	边沿	○	○	○	X	X	IER18.IEN3		DTCER195
	TX10	196	0310h	边沿	○	○	○	X	X	IER18.IEN4		DTCER196
	TE10	197	0314h	电平	○	X	X	X	X	IER18.IEN5		—
DMAC	DMAC0I	198	0318h	边沿	○	○	X	X	X	IER18.IEN6	IPR198	DTCER198
	DMAC1I	199	031Ch	边沿	○	○	X	X	X	IER18.IEN7	IPR199	DTCER199
	DMAC2I	200	0320h	边沿	○	○	X	X	X	IER19.IEN0	IPR200	DTCER200
	DMAC3I	201	0324h	边沿	○	○	X	X	X	IER19.IEN1	IPR201	DTCER201
—	保留	202	0328h	—	X	X	X	X	X	—	—	—
—	保留	203	032Ch	—	X	X	X	X	X	—	—	—
—	保留	204	0330h	—	X	X	X	X	X	—	—	—
—	保留	205	0334h	—	X	X	X	X	X	—	—	—
—	保留	206	0338h	—	X	X	X	X	X	—	—	—
—	保留	207	033Ch	—	X	X	X	X	X	—	—	—
—	保留	208	0340h	—	X	X	X	X	X	—	—	—
—	保留	209	0344h	—	X	X	X	X	X	—	—	—
—	保留	210	0348h	—	X	X	X	X	X	—	—	—
—	保留	211	034Ch	—	X	X	X	X	X	—	—	—
—	保留	212	0350h	—	X	X	X	X	X	—	—	—
—	保留	213	0354h	—	X	X	X	X	X	—	—	—
SCI0	RX10	214	0358h	边沿	○	○	○	X	X	IER1A.IEN6	IPR214	DTCER214
	TX10	215	035Ch	边沿	○	○	○	X	X	IER1A.IEN7		DTCER215
	TE10	216	0360h	电平	○	X	X	X	X	IER1B.IEN0		—
SCI1	RX11	217	0364h	边沿	○	○	○	X	X	IER1B.IEN1	IPR217	DTCER217
	TX11	218	0368h	边沿	○	○	○	X	X	IER1B.IEN2		DTCER218
	TE11	219	036Ch	电平	○	X	X	X	X	IER1B.IEN3		—
—	保留	220	0370h	—	X	X	X	X	X	—	—	—
—	保留	221	0374h	—	X	X	X	X	X	—	—	—
—	保留	222	0378h	—	X	X	X	X	X	—	—	—
—	保留	223	037Ch	—	X	X	X	X	X	—	—	—
—	保留	224	0380h	—	X	X	X	X	X	—	—	—
—	保留	225	0384h	—	X	X	X	X	X	—	—	—
GPT0	GTCIA0	226	0388h	边沿	○	○	○	X	X	IER1C.IEN2	IPR226	DTCER226
	GTCIB0	227	038Ch	边沿	○	○	○	X	X	IER1C.IEN3		DTCER227
	GTCIC0	228	0390h	边沿	○	○	○	X	X	IER1C.IEN4		DTCER228
	GTCIE0	229	0394h	边沿	○	○	○	X	X	IER1C.IEN5	IPR229	DTCER229
	GTCIV0	230	0398h	边沿	○	○	○	X	X	IER1C.IEN6		DTCER230
	LOCOI	231	039Ch	边沿	○	○	○	X	X	IER1C.IEN7		DTCER231

表 14.3 中断向量表 (6/6)

中断请求发生源	名称	向量号 (注1)	向量地址的 偏移	中断的 检测方法	CPU 中断	DTC 启动	DMAC 启动	sslb 返回	sacs 返回	IER	IPR	DT CER
GPT1	GTCIA1	232	03A0h	边沿	○	○	○	X	X	IER1D.IEN0	IPR232	DT CER232
	GTCIB1	233	03A4h	边沿	○	○	○	X	X	IER1D.IEN1		DT CER233
	GTCIC1	234	03A8h	边沿	○	○	○	X	X	IER1D.IEN2		DT CER234
	GTCIE1	235	03ACh	边沿	○	○	○	X	X	IER1D.IEN3	IPR235	DT CER235
	GTCIV1	236	03B0h	边沿	○	○	○	X	X	IER1D.IEN4		DT CER236
—	保留	237	03B4h	—	X	X	X	X	X	—	—	—
GPT2	GTCIA2	238	03B8h	边沿	○	○	○	X	X	IER1D.IEN6	IPR238	DT CER238
	GTCIB2	239	03BCh	边沿	○	○	○	X	X	IER1D.IEN7		DT CER239
	GTCIC2	240	03C0h	边沿	○	○	○	X	X	IER1E.IEN0		DT CER240
	GTCIE2	241	03C4h	边沿	○	○	○	X	X	IER1E.IEN1	IPR241	DT CER241
	GTCIV2	242	03C8h	边沿	○	○	○	X	X	IER1E.IEN2		DT CER242
—	保留	243	03CCh	—	X	X	X	X	X	—	—	—
GPT3	GTCIA3	244	03D0h	边沿	○	○	○	X	X	IER1E.IEN4	IPR244	DT CER244
	GTCIB3	245	03D4h	边沿	○	○	○	X	X	IER1E.IEN5		DT CER245
	GTCIC3	246	03D8h	边沿	○	○	○	X	X	IER1E.IEN6		DT CER246
	GTCIE3	247	03DCh	边沿	○	○	○	X	X	IER1E.IEN7	IPR247	DT CER247
	GTCIV3	248	03E0h	边沿	○	○	○	X	X	IER1F.IEN0		DT CER248
—	保留	249	03E4h	—	X	X	X	X	X	—	—	—
SCI12	RXI12	250	03E8h	边沿	○	○	○	X	X	IER1F.IEN2	IPR250	DT CER250
	TXI12	251	03ECh	边沿	○	○	○	X	X	IER1F.IEN3		DT CER251
	TEI12	252	03F0h	电平	○	X	X	X	X	IER1F.IEN4		—
—	保留	253	03F4h	—	X	X	X	X	X	—	—	—
—	保留	254	03F8h	—	X	X	X	X	X	—	—	—
—	保留	255	03FCh	—	X	X	X	X	X	—	—	—

注. 此表表示最大规格的中断向量。各产品的中断向量支持“表 1.2”所示的功能。详细内容请参考“表 1.2 各封装的功能比较一览表”。

注 1. 向量号越小，优先级越高。

14.3.2 高速中断的向量表地址

被设定为高速中断的中断向量表地址由 CPU 的高速中断向量寄存器 (FINTV) 指定。

14.3.3 非屏蔽中断的向量表地址

非屏蔽中断的向量表地址为“FFFF FFF8h”。

14.4 外围模块中断请求组的功能

14.4.1 中断请求组

最多将 4 个外围模块的中断信号进行分组，作为 1 个中断请求进行处理。

电平检测的中断请求被分在组 12。

如果在 GEN12.ENj 位 (j= 位号) 为“1”时检测到中断请求，GRP12.ISj 标志就变为“1”。

当组 12 内的任意一个 GRP12.ISj 标志为“1”时，该组对应的 IRn.IR 标志 (n= 中断向量号) 为“1”。

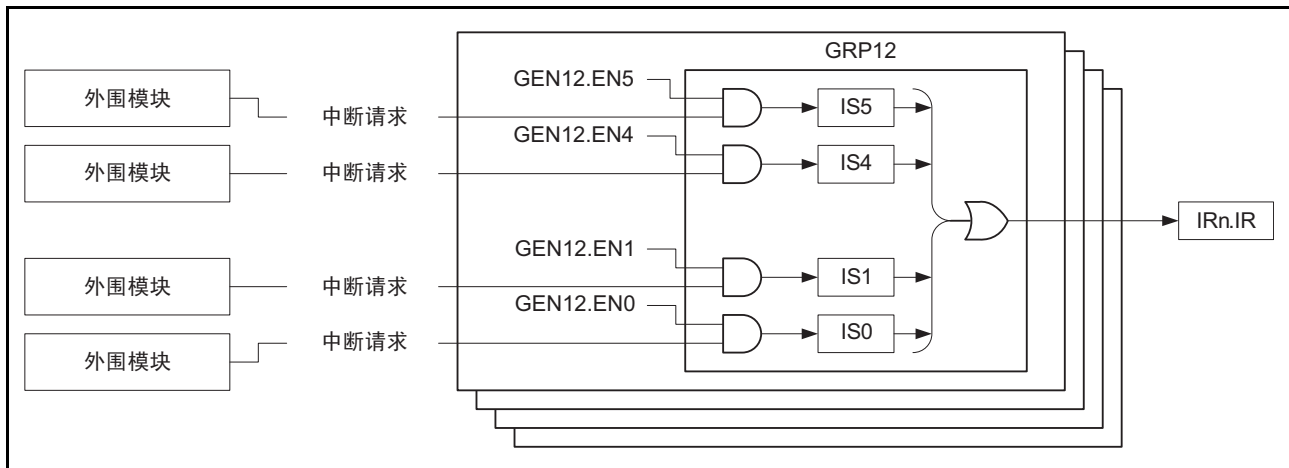


图 14.2 中断请求组的功能

(1) 电平检测的中断请求

电平检测的中断请求被分在组 12。

如果在 GEN12.ENj 位 (j= 位号) 为“1”时检测到中断请求，GRP12.ISj 标志就变为“1”。如果中断请求位为“0”，GRP12.ISj 标志也变为“0”。

当 GEN12.ENj 位为“0”时，不检测中断请求，GRP12.ISj 标志为“0”。

运行例子请参照“14.5.1.3 电平检测组中断和中断状态标志”。

(2) 各组的 interrupt 请求一览表

被分组的中断请求如“表 14.4 组 12 的中断请求”所示。

表 14.4 组 12 的中断请求

组	中断请求发生源	名称	GENm.ENj 位	GRPm.ISj 标志	向量号 (IRn.IR)
组 12	SCI0	ERI0 (SCI0 接收错误)	GEN12.EN0	GRP12.IS0	114
	SCI1	ERI1 (SCI1 接收错误)	GEN12.EN1	GRP12.IS1	
	SCI12	ERI12 (SCI12 接收错误)	GEN12.EN12	GRP12.IS12	
	RSPI0	SPEI0 (错误中断)	GEN12.EN13	GRP12.IS13	

14.5 中断的运行说明

中断控制器进行以下处理：

- 中断检测
- 中断的允许或者禁止控制
- 中断请求目标（CPU中断、DTC的启动、DMAC的启动）的选择
- 中断优先级的判断

14.5.1 中断检测

中断请求的检测方法有电平检测和边沿检测两种。

对于 IRQi 引脚 (i=0 ~ 5) 的外部中断请求，能通过设定 IRQCRi.IRQMD[1:0] 位，选择边沿检测或者电平检测。

对于外围模块的中断请求，给各中断源规定了边沿检测或者电平检测。

对应各中断源的检测方法请参照“表 14.3 中断向量表”。

中断请求分为两组，分别是边沿检测组和电平检测组，并且通过 GRP12 寄存器来保持。边沿检测组和电平检测组都将 IRn.IR 标志用作电平检测中断。中断分组功能的详细内容请参照“14.4 外围模块中断请求组的功能”，被分组的中断请求请参照“表 14.4 组 12 的中断请求”。

14.5.1.1 边沿检测的中断状态标志

对外围功能中断和外部引脚中断进行边沿检测时的 IRn.IR 标志变化如图 14.3 所示。

在发生中断请求时的中断信号的变化位置，IRn.IR 标志变为“1”。当中断请求目标为 CPU 时，如果接受中断，IRn.IR 标志就自动变为“0”。当中断请求目标为 DMAC 或者 DTC 时，IRn.IR 标志的清除时序因 DMAC/DTC 的传送设定和传送次数而不同。详细内容请参照“表 14.5 启动 DMAC 或者 DTC 时的运行”。不需要通过软件清除 IRn.IR 标志。

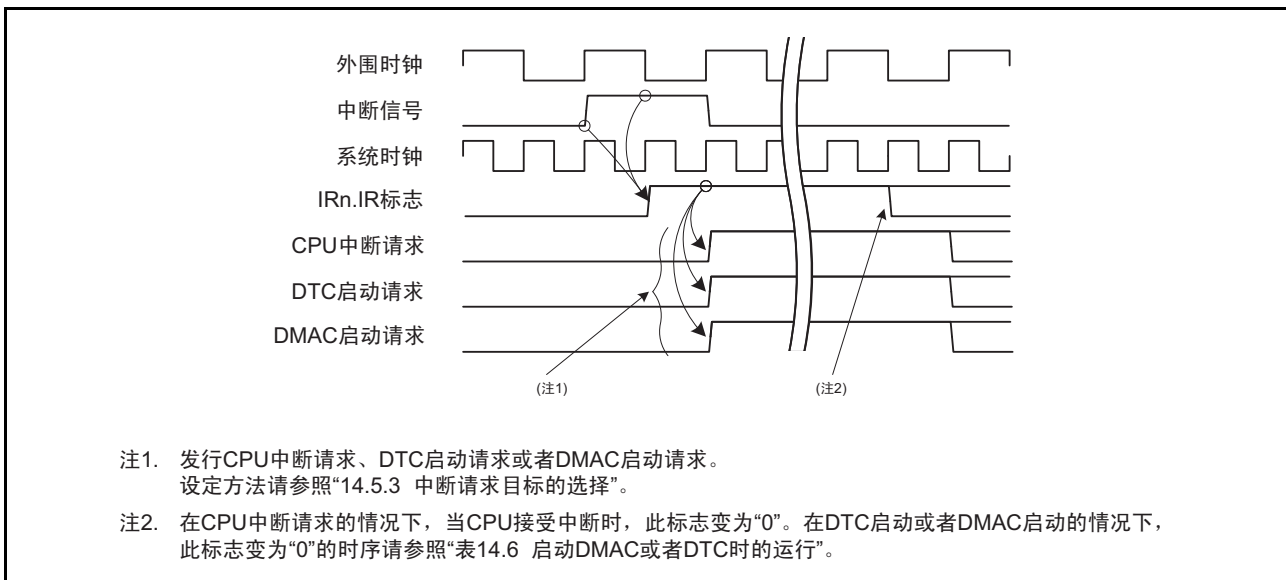


图 14.3 边沿检测时的 IRn.IR 标志变化

图 14.4 ~ 图 14.7 的中断信号是中断控制器的信号。中断向量号 64 ~ 69 的中断时序和其他中断不同。在中断向量号 64 ~ 69 的 IRQ 引脚中断的情况下，从 IRQ 引脚输入增加内部延迟 +2 个 PCLK 的延迟；在中断向量号 80 ~ 95 的中断的情况下，增加 2 个 PCLK 的延迟。

如果在每个周期产生中断信号，就不能检测后续的中断。对于连续的中断请求，必须至少空出系统时钟和外围时钟中频率慢的 2 个时钟周期。

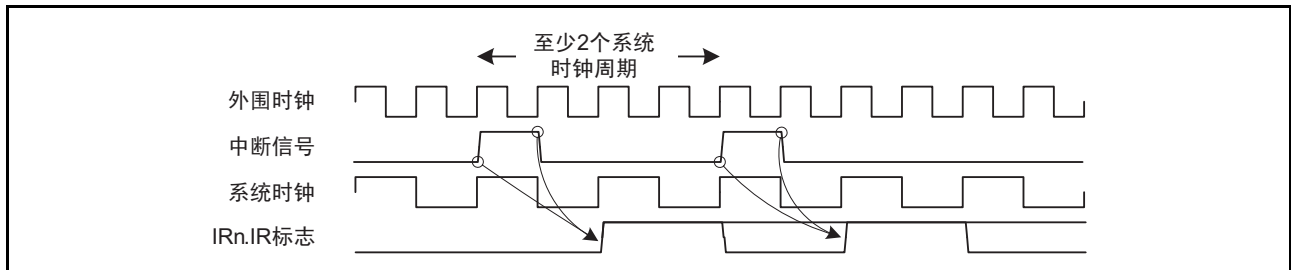


图 14.4 连续中断请求的发行间隔（系统时钟频率 < 外围时钟频率的情况）

在发生中断请求并且 IRn.IR 标志为“1”的状态下，忽视再次产生的中断请求（注 1）。

IRn.IR 标志再次被置位的时序如图 14.5 所示。

注 1. 但是，在 SCI、RSPI、RIIC 的各发送中断或者接收中断的情况下，保持在 IRn.IR 标志为“1”的状态下产生的中断请求，在 IRn.IR 标志变为“0”后，因被保持的请求而 IRn.IR 标志再次变为“1”。详细内容请参照“27. 串行通信接口（SC1c、SC1d）”、“28. I²C 总线接口（RIIC）”和“29. 串行外围接口（RSPI）”中的各中断的说明。

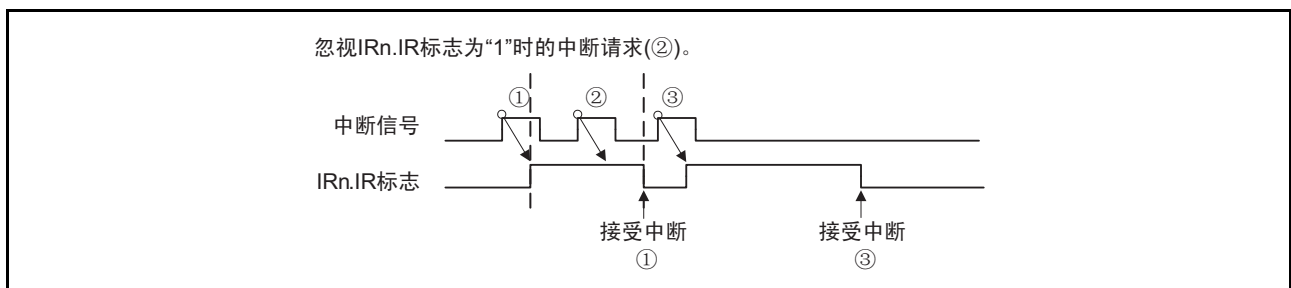


图 14.5 IRn.IR 标志再次被置位的时序

即使在 IRn.IR 标志变为“1”后禁止中断（通过外围模块的中断允许位禁止中断请求的输出），IRn.IR 标志也不受影响而保持“1”的状态。禁止中断时的 IRn.IR 标志变化如图 14.6 所示。

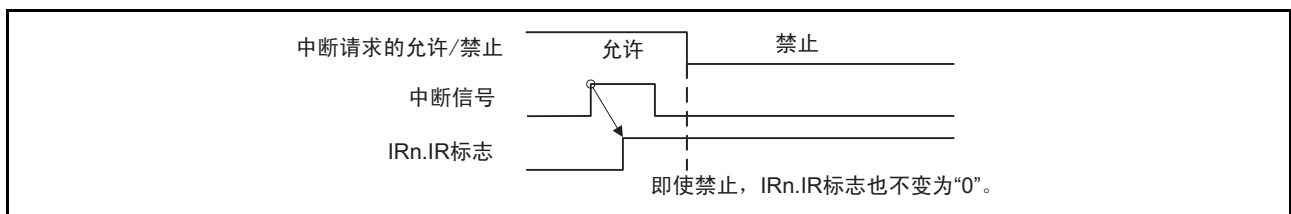


图 14.6 中断请求的禁止和 IRn.IR 标志的关系

14.5.1.2 电平检测的中断状态标志

对外围功能中断和外部引脚中断进行电平检测时的 IRn.IR 标志变化如图 14.7 所示。

在中断信号有效期间，IRn.IR 标志保持“1”。要将 IRn.IR 标志置“0”时，必须将中断发生源的中断请求标志置“0”。必须在确认中断请求发生源的中断请求标志和 IRn.IR 标志都变为“0”后结束中断处理程序。

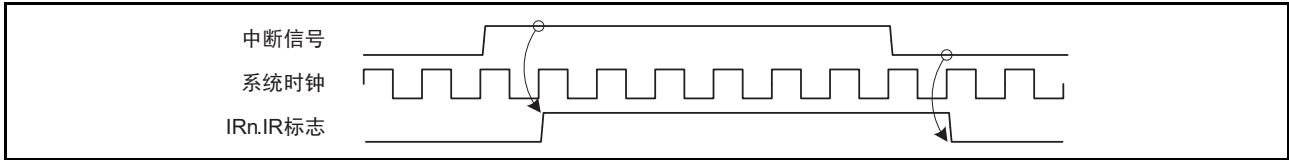


图 14.7 电平检测时的 IRn.IR 标志变化

电平检测的中断处理步骤如图 14.8 所示。

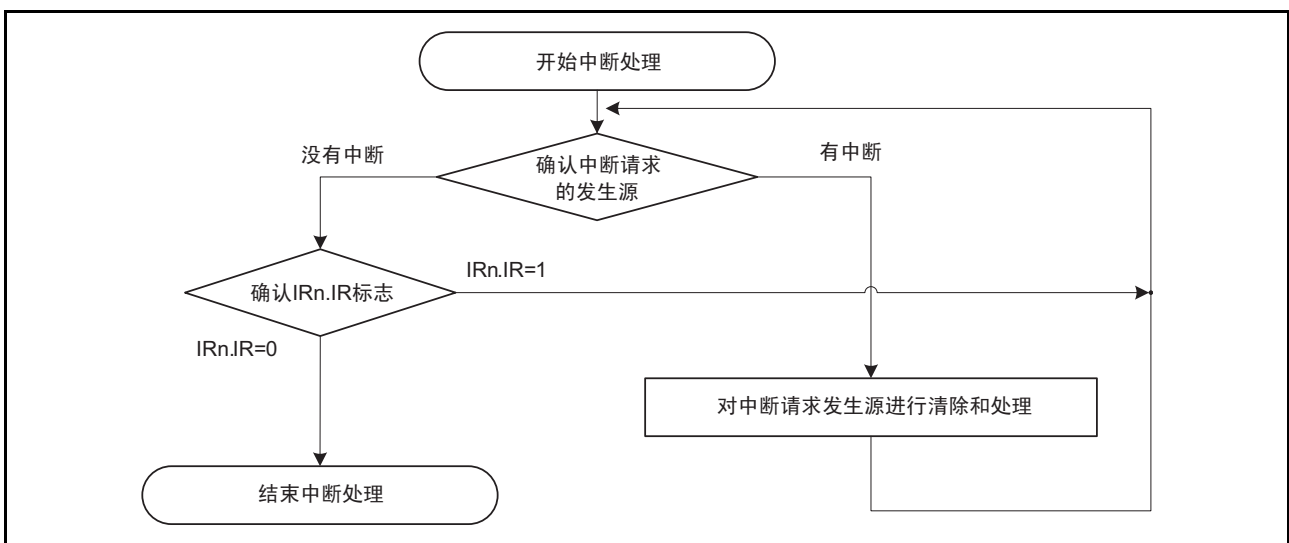


图 14.8 电平检测的中断处理步骤

14.5.1.3 电平检测组中断和中断状态标志

电平检测的中断请求被分在组 12。该组对应的 IRn.IR 标志用作电平检测中断。

电平检测中断请求的检测运行例子如图 14.9 所示，发生多个电平检测中断请求（被分配到 1 组）时的运行例子如图 14.10 所示。

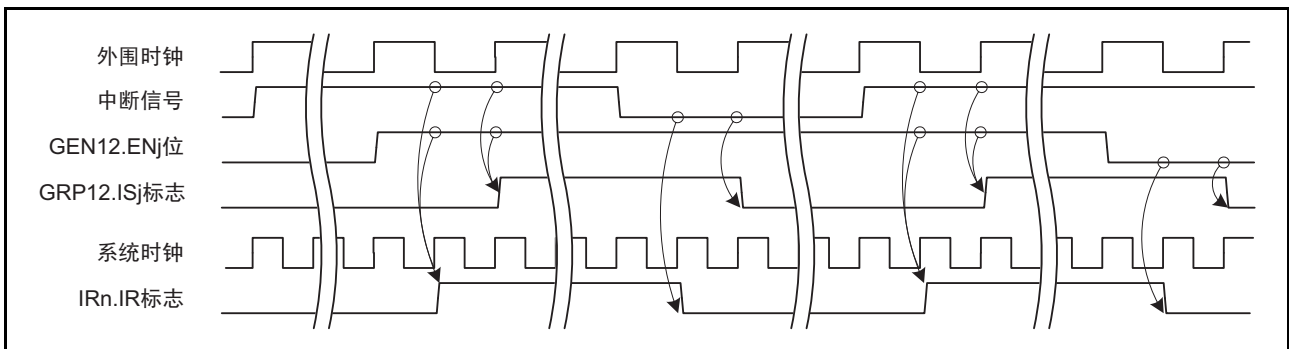


图 14.9 电平检测组中断的中断请求运行例子

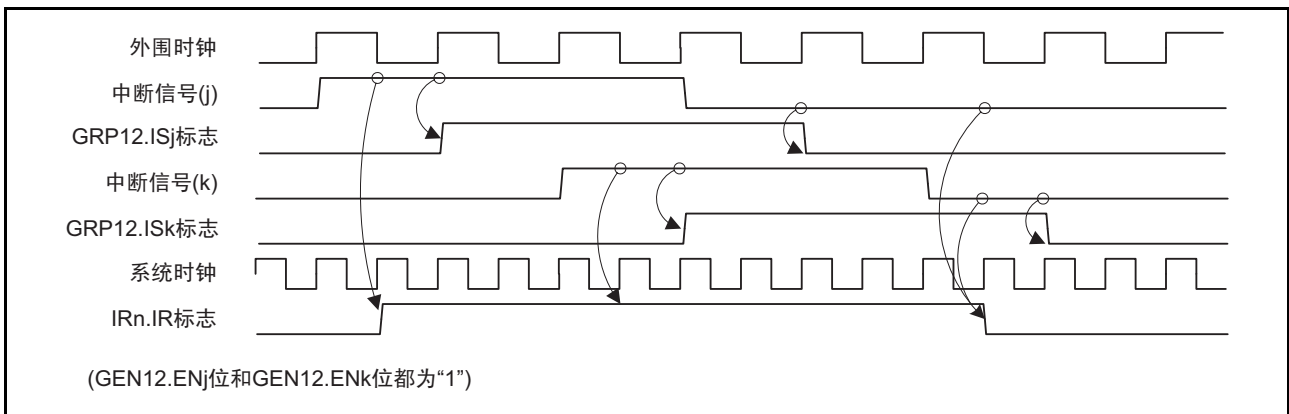


图 14.10 发生多个电平检测中断请求（被分配到 1 组）时的运行例子

必须按照图 14.11 所示的步骤进行电平检测组的中断处理。

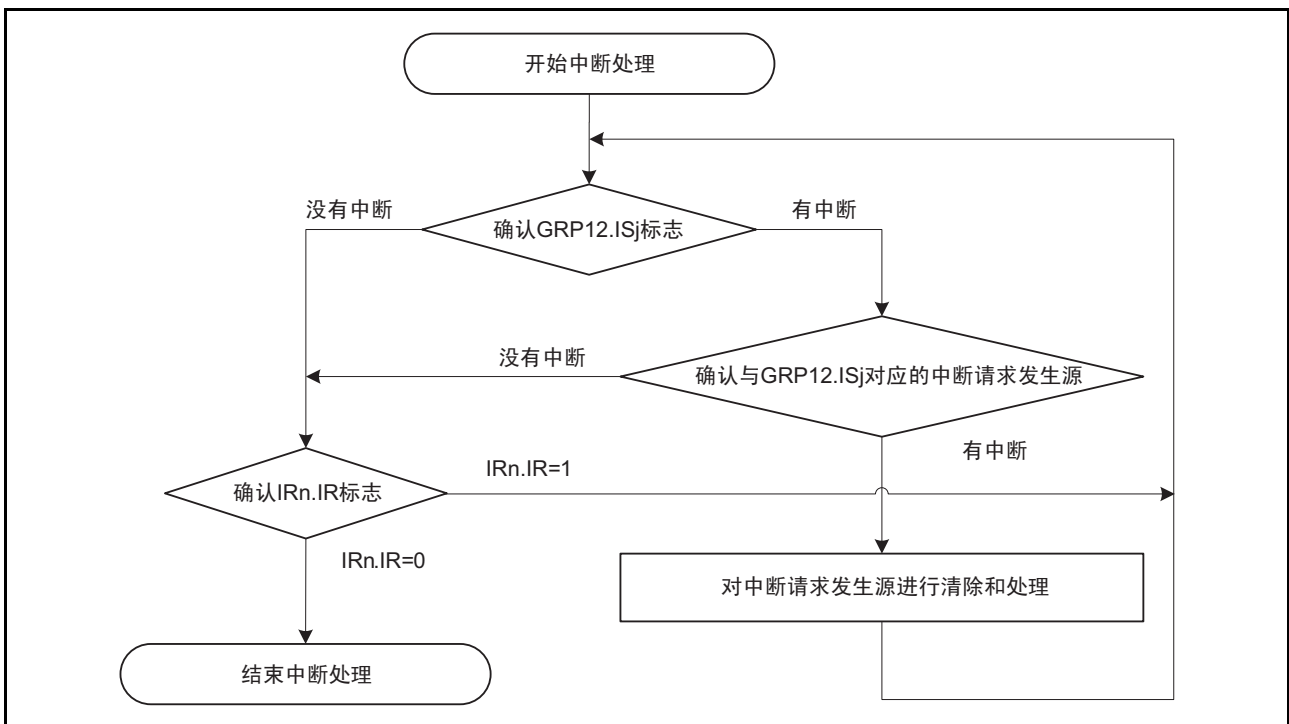


图 14.11 电平检测组的中断处理步骤

14.5.2 中断请求的允许 / 禁止

为了允许中断请求，需要进行以下设定：

1. 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中断请求的输出。
2. 通过 GEN12.ENj 位允许被分组的中断请求。
3. 通过 IERm.IENj 位允许中断请求。

如果在中断发生源发生被允许中断输出的中断请求，对应的 IRn.IR 标志就变为“1”。

如果发生被分组的中断请求，就在对应的 GRP12.ISj 标志变为“1”的同时，该组对应的 IRn.IR 标志也变为“1”。

通过 IERm.IENj 位允许中断请求，将 IRn.IR 标志为“1”的中断请求输出到中断请求目标。通过 IERm.IENj 位禁止中断请求，保留 IRn.IR 标志为“1”的中断请求。

IRn.IR 标志不受 IERm.IENj 位的影响。

禁止中断请求的步骤如下：

1. 将 IERm.IENj 位设定为禁止中断请求。
2. 在被分组的中断请求的情况下，将 GEN12.ENj 位设定为禁止。
3. 将外围模块的中断输出允许位设定为禁止，并且通过读已写的寄存器来确认写操作的结束。
4. 根据需要，确认 IRn.IR 标志或者将 IRn.IR 标志置“0”（注1）。
在被分组的中断请求的情况下，确认 GRP12.ISj 标志或者将 GRP12.ISj 标志置“0”。

注 1. 如果将 SCI、RSPI、RIIC 的各发送中断或者接收中断从允许状态改为禁止状态，就必须按照上述步骤将 IRn.IR 标志置“0”。详细内容请参照“27. 串行通信接口 (SClC、SClD)”、“28. I²C 总线接口 (RIIC)”和“29. 串行外围接口 (RSPI)”中的各中断的说明。

14.5.3 中断请求目标的选择

给各中断源规定了能设定的中断请求目标，能设定的请求目标如“表 14.3 中断向量表”所示。不能选择表 14.3 中没有记载“○”的中断请求目标。

要通过 IRQ 引脚将 DTC/DMAC 设定为中断请求目标时，必须将 IRQCri.IRQMD[1:0] 位设定为边沿检测。中断请求目标的设定方法如下所示。

(1) DMAC 的启动

必须在 IERm.IENj 位为“0”时对各中断源进行以下设定：

1. 给 DMAC 的各通道准备的 DMAC 启动请求选择寄存器 (DMRSRm) 指定对应的中断源向量号 (注1)。
2. 将 DMAC 对应通道的启动源 (DMACm.DMTMD.DCTG[1:0]) 设定为“01b” (检测中断模块)。
3. 将 DMAC 对应通道的 DMAC 启动允许位 (DMACm.DMCNT.DTE) 置“1”。

必须在上述状态下将 IERm.IENj 位置“1”。

还必须将 DMAC 运行允许位 (DMAST.DMST) 置“1”。各中断源的设定和 DMAC 运行允许位的设定不分先后顺序。

DMAC 的设定步骤请参照“17. DMA 控制器 (DMACA)”的“17.3.7 DMAC 的启动”。

(2) DTC 的启动

必须在 IERm.IENj 位为“0”时对各中断源进行以下设定：

1. 将对应中断源的 DTC 启动允许寄存器的 DTC 传送允许位 (DTCERn.DTCE) 置“1” (注1)。

必须在上述状态下将 IERm.IENj 位置“1”。

还必须将 DTC 模块启动位 (DTCST.DTCST) 置“1”。各中断源的设定和 DTC 模块启动位的设定不分先后顺序。

DTC 的设定步骤请参照“18. 数据传送控制器 (DTCa)”的“18.5 DTC 的设定步骤”。

注 1. 不能给 DTC 启动允许 (DTCERn.DTCE) 和 DMAC 启动请求选择 (DMRSRm) 设定相同的中断源，也不能给多个 DMRSRm 设定相同的中断源。

(3) CPU 中断请求

对于中断请求目标既不是 DMAC 也不是 DTC 的中断源，中断请求目标为 CPU。

必须在未设定上述的 DMAC 启动和 DTC 启动的状态下将 IERm.IENj 位置“1”。

将 DMAC 和 DTC 设定为中断请求目标时的运行如表 14.5 所示。

表 14.5 启动 DMAC 或者 DTC 时的运行

中断请求目标	DISEL	剩余的 传送次数	1 次请求的运行	IR (注 1)	传送后的中断请求目标
DMAC	1	≠0	DMA 传送 → CPU 中断	在接受 CPU 中断时清除。	DMAC
		=0	DMA 传送 → CPU 中断	在接受 CPU 中断时清除。	清除 DMACm.DMCNT.DTE 位并且转换到 CPU。
	0	≠0	DMA 传送	在开始 DMAC 传送时清除。	DMAC
		=0	DMA 传送 (注 2)	在开始 DMAC 传送时清除 (注 2)。	清除 DMACm.DMCNT.DTE 位并且转换到 CPU。
DTC (注 3)	1	≠0	DTC 传送 → CPU 中断	在接受 CPU 中断时清除。	DTC
		=0	DTC 传送 → CPU 中断	在接受 CPU 中断时清除。	清除 DTCER.DTCE 位并且转换到 CPU。
	0	≠0	DTC 传送	在读取 DTC 传送信息后开始 DTC 数据传送时清除。	DTC
		=0	DTC 传送 → CPU 中断 (注 2)	在接受 CPU 中断时清除 (注 2)。	清除 DTCER.DTCE 位并且转换到 CPU。

通过 DMACm.DMCSL.DISEL 位设定 DMAC 的 DISEL，通过 DTC.MRB.DISEL 位设定 DTC 的 DISEL。

注 1. 忽视在 IRn.IR 标志为“1”时再次产生的中断请求 (DTC/DMAC 的启动请求)。

注 2. DISEL 为“0”并且剩余的传送次数为“0”时的运行因 DTC 和 DMAC 而不同。

注 3. 在链传送的情况下，DTC 的传送继续到最后的链传送为止。最后的链传送时的 CPU 中断的有无、IRn.IR 标志的清除、传送后的中断请求目标等，取决于最后的链传送的 DISEL 和剩余的传送次数。有关链传送，请参照“18. 数据传输控制器 (DTCa)”的“表 18.3 链传送的条件”。

要更改中断请求目标时，必须在 IERm.IENj 位为“0”时进行。

要在进行“(1) DMAC 的启动”的设定后传送未结束的状态 (未清除 DMACm.DMCNT.DTE 位的状态) 下更改中断请求目标或者将 DMA 启动源更改为其他中断源时，必须按照以下步骤进行更改：

1. 将要取消的中断源以及作为新启动对象的中断源的 IERm.IENj 位置“0”。
2. 确认 DMAC 的传送状况。如果是在传送过程中，就等待传送结束。
3. 进行“(1) DMAC 的启动”的设定。

要在进行“(2) DTC 的启动”的设定后传送未结束的状态 (未清除 DTCERn.DTCE 位的状态) 下更改中断请求目标或者更改 DTC 的传送设定内容时，就必须按照以下步骤进行更改：

1. 将要取消的中断源以及作为新启动对象的中断源的 IERm.IENj 位置“0”。
2. 确认 DTC 的传送状况。如果是在传送过程中，就等待传送结束。
3. 设定“(2) DTC 的启动”。

14.5.4 优先级的判断

中断控制器判断各中断请求目标的优先级。各中断请求目标的优先级判断方法如下。

(1) 中断请求目标为 CPU 时的优先级判断

被设定为高速中断的中断源的优先级最高，其次优先中断优先级设定位 (IPRn.IPR[3:0]) 中值大的中断源。如果 IPRn.IPR[3:0] 位的值相同并且有多个优先级的中断源，就优先向量号小的中断源。

(2) 中断请求目标为 DTC 时的优先级判断

不受 IPRn.IPR[3:0] 位的影响，优先向量号小的中断源。

(3) 中断请求目标为 DMAC 时的优先级判断

不受 IPRn.IPR[3:0] 位的影响。有关 DMAC 通道的优先级，请参照“17. DMA 控制器 (DMACA)”。

14.5.5 高速中断

高速中断是高速地进行CPU中断处理的功能，只对CPU的中断请求有效，不影响DTC和DMAC的启动请求。

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 位设定为允许高速中断的状态下，如果发生对应的中断源，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 IPRn.IPR[3:0] 位的设定无关，是优先级最高的中断源。

有关高速中断的详细内容，请参照“13. 异常处理”。

14.5.6 数字滤波器

外部中断请求引脚 IRQi (i=0 ~ 5) 和 NMI 引脚中断具有数字滤波器功能。

数字滤波器通过滤波器的采样时钟 (PCLK) 对输入信号进行采样，消除采样周期不足 3 次的脉冲。

当使用 IRQi 引脚的数字滤波器时，必须通过 IRQFLTC0.FCLKSELi[1:0] 位 (i=0 ~ 5) 设定采样频率 (PCLK、PCLK/8、PCLK/32、PCLK/64) 并且将 IRQFLTE0.FLTENi 位置“1” (数字滤波器有效)。

当使用 NMI 引脚中断的数字滤波器时，必须通过 NMIFLTC.NFCLKSEL[1:0] 位设定采样频率 (PCLK、PCLK/8、PCLK/32、PCLK/64) 并且将 NMIFLTE.NFLTEN 位置“1” (数字滤波器有效)。数字滤波器的运行例子如图 14.12 所示。

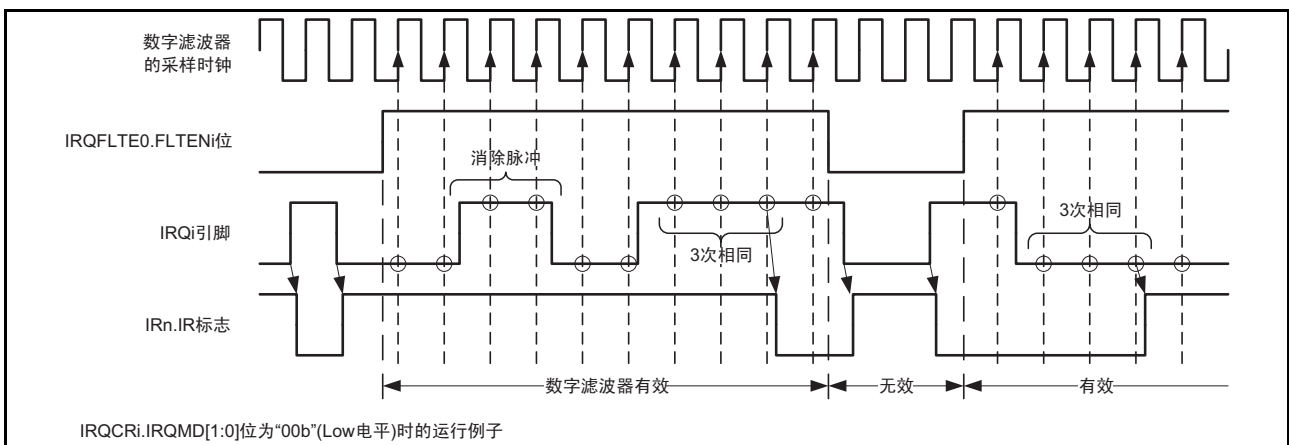


图 14.12 数字滤波器的运行例子

在向软件待机模式转移时，必须将 IRQFLTE0.FLTENi 位和 NMIFLTE.NFLTEN 位都置“0” (数字滤波器无效)。要在从软件待机模式返回后再次使用数字滤波器时，必须将 IRQFLTE0.FLTENi 位或者 NMIFLTE.NFLTEN 位置“1” (数字滤波器有效)。

14.5.7 外部引脚中断

使用外部引脚中断的步骤如下：

1. 将 IERm.IENj 位置“0”（禁止中断请求）。
2. 将 IRQFLTE0.FLTENi 位（i=0~5）置“0”（数字滤波器无效）。
3. 通过 IRQFLTC0.FCLKSELi[1:0] 位设定数字滤波器的采样时钟。
4. 设定 I/O 端口并且进行确认。
5. 通过 IRQCRi.IRQMD[1:0] 位设定检测方法。
6. 将 IRn.IR 标志置“0”（边沿检测的情况）。
7. 将 IRQFLTE0.FLTENi 位置“1”（数字滤波器有效）。
8. 当启动 DMAC 时，设定 DMRSRm.DMRS[7:0] 位；当启动 DTC 时，设定 DTCERn.DTCE 位（在不设定任何位的情况下，为 CPU 中断）。
9. 将 IERm.IENj 位置“1”（允许中断请求）。

14.6 非屏蔽中断的运行说明

非屏蔽中断有 NMI 引脚中断、振荡停止检测中断、WDT 下溢 / 刷新错误中断、IWDT 下溢 / 刷新错误中断、电压监视 1 中断和电压监视 2 中断。非屏蔽中断只限于向 CPU 请求的中断，不能启动 DTC 和 DMAC。在包括高速中断在内的全部中断中，非屏蔽中断是最优先的中断。

非屏蔽中断请求与 CPU 的 PSW.I 位（中断允许位）和 PSW.IPL[3:0] 位（处理器中断优先级）的状态无关，总是被接受。能通过非屏蔽中断状态寄存器（NMISR）确认非屏蔽中断的有无。

必须通过非屏蔽中断处理程序确认 NMISR 寄存器的全部位都是“0”。

初始状态为“禁止非屏蔽中断”。在使用非屏蔽中断的系统中，必须在程序处理的起始位置按照以下步骤进行设定。

非屏蔽中断的使用步骤：

1. 设定堆栈指针（SP）。
2. 在使用 NMI 引脚时，将 NMIFLTE.NFLTEN 位置“0”（数字滤波器无效）。
3. 在使用 NMI 引脚时，通过 NMIFLTC.NFCLKSEL[1:0] 位设定数字滤波器的采样时钟。
4. 在使用 NMI 引脚时，通过 NMICR.NMIMD 位设定 NMI 引脚的检测方法。
5. 在使用 NMI 引脚时，在给 NMICLR.NMICLR 位写“1”后将 NMISR.NMIST 标志置“0”。
6. 在使用 NMI 引脚时，将 NMIFLTE.NFLTEN 位置“1”（数字滤波器有效）。
7. 通过将非屏蔽中断允许寄存器（NMIER）允许的中断对应位置“1”，允许使用非屏蔽中断。

如果给 NMIER 寄存器写“1”，就忽视以后 NMIER 寄存器的写操作。不能禁止非屏蔽中断，只能通过复位禁止非屏蔽中断。

非屏蔽中断的处理流程请参照“13. 异常处理”。

通过给 NMICLR.NMICLR 位写“1”，NMI 状态标志（NMISR.NMIST）变为“0”。

通过给 NMICLR.OSTCLR 位写“1”，振荡停止检测中断状态标志（NMISR.OSTST）变为“0”。

通过给 NMICLR.WDTCLR 位写“1”，WDT 下溢 / 刷新错误状态标志（NMISR.WDTST）变为“0”。

通过给 NMICLR.IWDTCLR 位写“1”，IWDT 下溢 / 刷新错误状态标志（NMISR.IWDTST）变为“0”。

通过给 NMICLR.LVD1CLR 位写“1”，电压监视 1 中断状态标志（NMISR.LVD1ST）变为“0”。

通过给 NMICLR.LVD2CLR 位写“1”，电压监视 2 中断状态标志（NMISR.LVD2ST）变为“0”。

14.7 从低功耗状态的返回

能用作从睡眠模式、全模块时钟停止模式、软件待机模式返回中断的中断源如“表 14.3 中断向量表”所示。详细内容请参照“11. 低功耗功能”，各低功耗模式中返回对象中断的设定方法如下所示。

14.7.1 从睡眠模式的返回

能通过非屏蔽中断 (WDT 下溢 / 刷新错误除外) 和全部中断源的中断进行返回, 返回条件如下:

- 中断
 1. 中断请求目标为 CPU。
 2. 已经通过 IERm.IENj 位允许对应的中断请求。
 3. 高于 CPU.PSW.IPL[3:0] 位的中断优先级。
 4. 在被分组的中断请求的情况下, 已经通过 GEN12.ENj 位允许对应的中断请求。
- 非屏蔽中断

已经通过 NMIER 寄存器允许对应的中断请求。

14.7.2 从全模块时钟停止模式的返回

能通过非屏蔽中断 (WDT 下溢 / 刷新错误除外) 以及能从全模块时钟停止模式返回的中断进行返回, 返回条件如下:

- 中断
 1. 是能从全模块时钟停止模式返回的中断源。
 2. 中断请求目标为 CPU。
 3. 已经通过 IERm.IENj 位允许对应的中断请求。
 4. 高于 CPU.PSW.IPL[3:0] 位的中断优先级。
- 非屏蔽中断

已经通过 NMIER 寄存器允许对应的中断请求。

14.7.3 从软件待机模式的返回

能通过非屏蔽中断 (WDT 下溢 / 刷新错误除外) 以及能从软件待机模式返回的中断进行返回, 返回条件如下:

- 中断
 1. 是能从软件待机模式返回的中断源。
 2. 中断请求目标为 CPU。
 3. 已经通过 IERm.IENj 位允许对应的中断请求。
 4. 高于 CPU.PSW.IPL[3:0] 位的中断优先级。
(对于被设定为高速中断的中断源, 除了高速中断设定寄存器 (FIR), 还必须将对应的中断优先级 (IPRn) 设定为高于 CPU.PSW.IPL。)

对于不满足上述条件的 IRQ 引脚, 在软件待机模式中检测不到时钟停止期间内发生的中断源。

- 非屏蔽中断

已经通过 NMIER 寄存器允许对应的中断请求。
必须按照以下步骤进行软件待机模式的转移或者解除:

 1. 必须在向软件待机模式转移前将返回对象的中断源的数字滤波器置为无效 (将 IRQFLTEn.FLTENi 位和 NMIFLTE.NFLTEN 位都置“0”)。
 2. 要在从软件待机模式返回后再次使用数字滤波器时, 必须将数字滤波器置为有效 (将 IRQFLTEn.FLTENi 位和 NMIFLTE.NFLTEN 位都置“1”)。

14.8 使用时的注意事项

14.8.1 使用非屏蔽中断时的 WAIT 指令的注意事项

必须在确认 NMISR 寄存器的全部状态标志都为“0”后发行 WAIT 指令。

15. 总线

15.1 概要

总线的规格和各种总线的地址对应表分别如表 15.1 和表 15.2 所示，总线结构图如图 15.1 所示。

表 15.1 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（RAM、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（RAM、ROM）。 • 与系统时钟（ICLK）同步运行。
存储器总线	存储器总线 1	<ul style="list-style-type: none"> • 连接 RAM。
	存储器总线 2	<ul style="list-style-type: none"> • 连接 ROM。
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 DTC 和 DMAC。 • 连接内部存储器（RAM、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（DTC、DMAC、中断控制器、总线错误监视部）。 • 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（内部外围总线 1、3、4、5 以外的外围功能）。 • 与外围模块时钟（PCLKB）同步运行。
	内部外围总线 3	保留区
	内部外围总线 4	<ul style="list-style-type: none"> • 连接外围功能（MTU3、GPT）。 • 与外围模块时钟（PCLKA）同步运行。
	内部外围总线 5	保留区
	内部外围总线 6	<ul style="list-style-type: none"> • 连接 ROM（P/E）和 E2 数据闪存。 • 与 FlashIF 时钟（FCLK）同步运行。

P/E: 编程 / 擦除

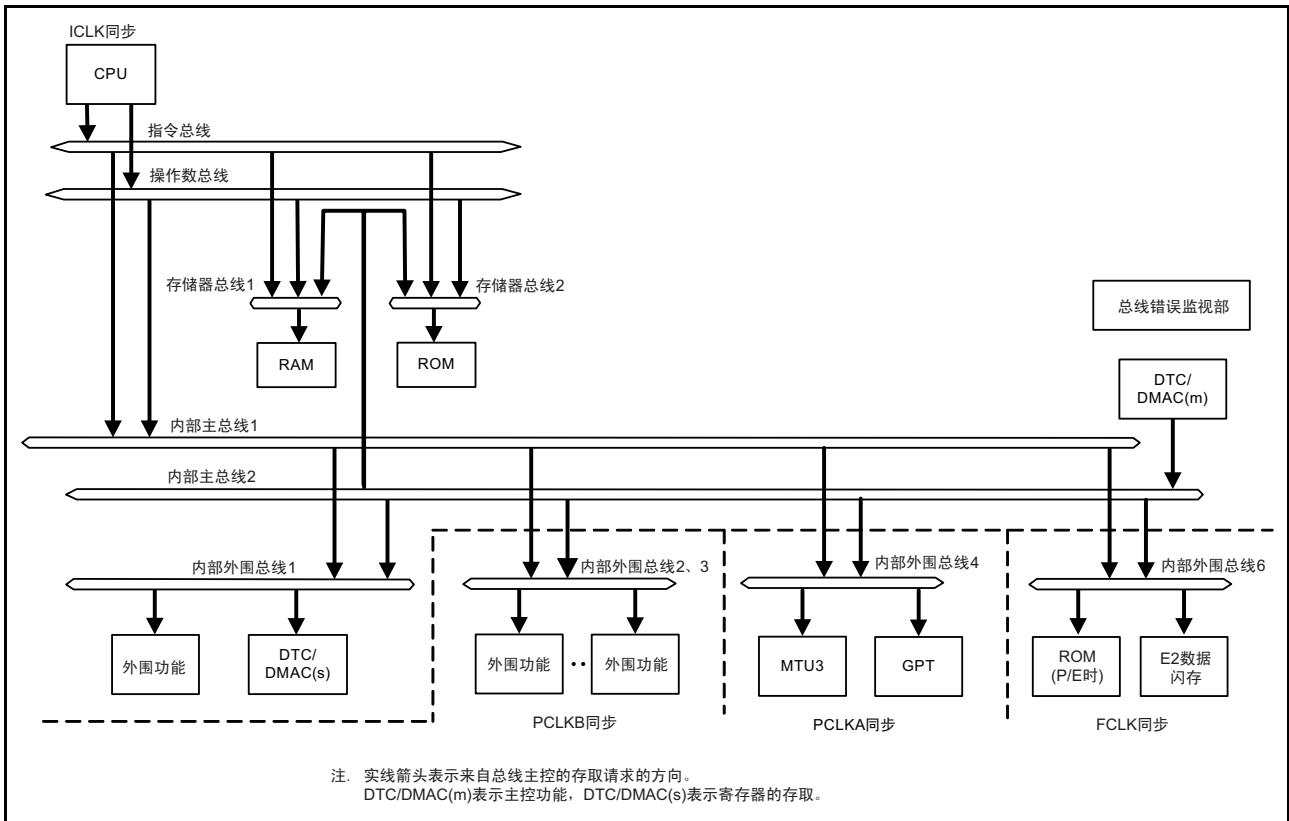


图 15.1 总线结构图

表 15.2 各种总线的地址对应表

地址	总线	内容
0000 0000h ~ 0000 1FFFh	存储器总线 1	RAM
0000 2000h ~ 0007 FFFFh		保留区
0008 0000h ~ 0008 7FFFh	内部外围总线 1	外围 I/O 寄存器
0008 8000h ~ 0009 FFFFh	内部外围总线 2	
000A 0000h ~ 000B FFFFh	保留区	
000C 0000h ~ 000D FFFFh	内部外围总线 4	
000E 0000h ~ 000F FFFFh	保留区	
0010 0000h ~ 00FF FFFFh	内部外围总线 6	E2 数据闪存、 ROM (P/E 时)
0100 0000h ~ FFFE FFFFh	保留区	保留区
FFFF 0000h ~ FFFF FFFFh	存储器总线 2	ROM

15.2 总线说明

15.2.1 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 CPU 的操作数存取。

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 而由 CPU 直接存取。内部 ROM 为只读区域，能由 CPU 直接存取，但是必须通过内部外围总线进行编程和擦除。

内部主总线 1 对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。

如果取指令和操作数存取的请求是针对不同的总线（存储器总线 1、存储器总线 2、内部主总线 1），就能同时进行各自的总线存取。例如，内部 ROM 和内部 RAM 能并行运行。

15.2.2 存储器总线

存储器总线有存储器总线 1 和存储器总线 2，存储器总线 1 连接内部 RAM，存储器总线 2 连接内部 ROM。存储器总线 1 和存储器总线 2 对 CPU 总线（取指令和操作数）和内部主总线 2 的总线权请求进行仲裁。

能分别通过总线优先级控制寄存器的存储器总线 1（内部 RAM）的优先级控制位（BUSPRI.BPRA[1:0]）和存储器总线 2（内部 ROM）的优先级控制位（BUSPRI.BPRO[1:0]）设定 2 条总线的优先级。在优先级固定的情况下，2 条总线的优先级为内部主总线 2 > CPU 总线（操作数 > 取指令）；在优先级交替的情况下，内部主总线 2 和 CPU 总线中已接受总线请求的一方的优先级变低。

15.2.3 内部主总线

内部主总线由 CPU 使用的总线（内部主总线 1）以及其他总线主控（DTC、DMAC）使用的总线（内部主总线 2）构成。

内部主总线 1 对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。

内部主总线 2 对 DTC 和 DMAC 的总线权请求进行仲裁，优先级为 DMAC > DTC，如表 15.3 所示。

对于 DTC 和 DMAC，只有接受了启动请求的一方才能请求总线权。DTC 和 DMAC 的启动请求优先级为 DMAC0 > DMAC1 > DMAC2 > DMAC3 > DTC，与 BUSPRI 寄存器的设定无关。

如果 CPU 和其他总线主控的请求是针对不同的总线（内部存储器、内部外围总线 1、2、4、6），就能同时进行各自的总线存取。

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，与总线优先级控制寄存器（BUSPRI）的设定无关，并且在读以及回写 DTC 的传送信息过程中也不接受 DTC 以外的总线存取。

表 15.3 总线主控优先级

优先级	总线主控
高	DMAC
↑	DTC
低	CPU

15.2.4 内部外围总线

连接内部外围总线的外围功能如表 15.4 所示。

表 15.4 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	DTC、DMAC、中断控制器、总线错误监视部
内部外围总线 2	内部外围总线 1、3、4、5 以外的外围功能
内部外围总线 3	保留区
内部外围总线 4	MTU3、GPT
内部外围总线 5	保留区
内部外围总线 6	ROM(P/E 时)/E2 数据闪存

内部外围总线 1、2、4、6 分别对 CPU（内部主总线 1）和其他总线主控（内部主总线 2）的总线权请求进行仲裁。

能通过总线优先级控制寄存器（BUSPRI）设定 2 条总线的优先级，能通过内部外围总线 1 的优先级控制位（BUSPRI.BPIB[1:0]）、内部外围总线 2 的优先级控制位（BUSPRI.BPGB[1:0]）以及内部外围总线 4 的优先级控制位（BUSPRI.BPFB[1:0]）设定各总线的优先级。在优先级固定的情况下，优先级为内部主总线 2 > 内部主总线 1；在优先级交替的情况下，内部主总线 1 和内部主总线 2 中已接受总线请求的一方的优先级变低。

必须注意：如果更改 BUSPRI 寄存器的设定，接受请求的顺序就可能发生变化（参照图 15.2）。

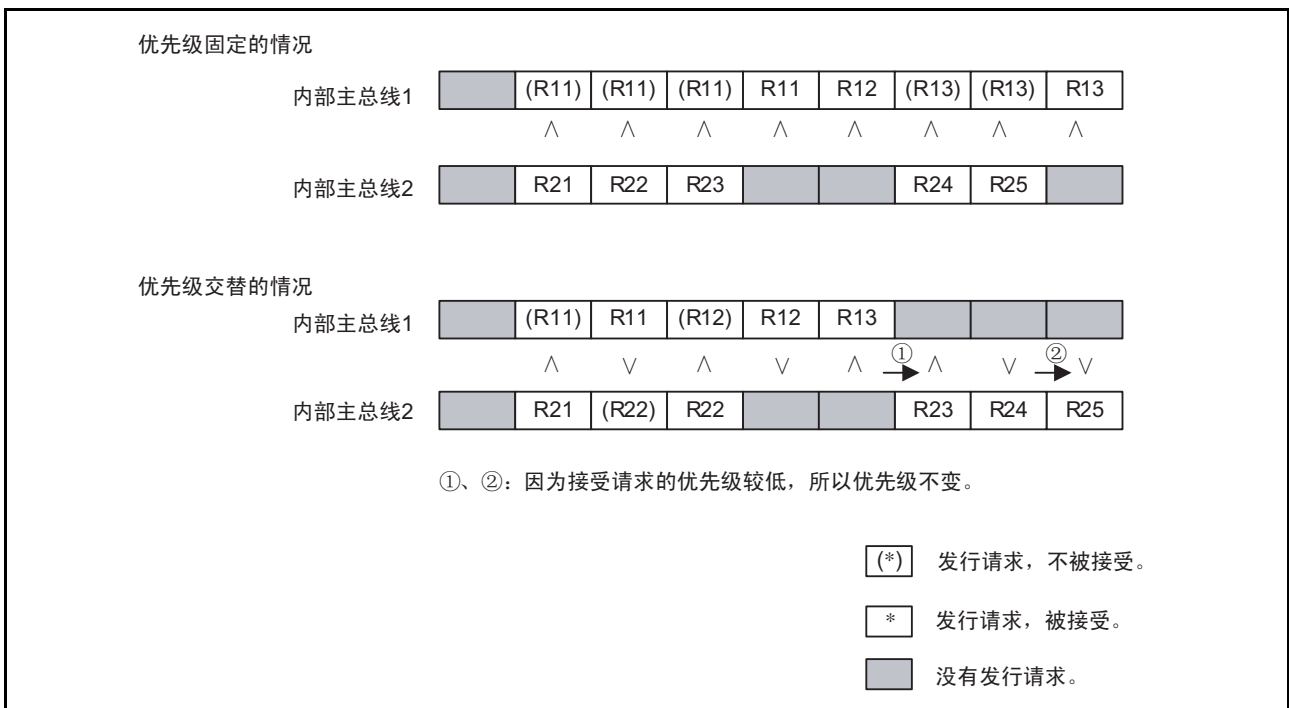


图 15.2 内部外围总线优先级

15.2.5 写缓冲器功能（内部外围总线）

内部外围总线具有写缓冲器功能，在进行写存取时，能不等运行结束就接受后面的存取。但是，在从相同的总线主控进行存取时，如果写存取后面的存取是针对不同的内部外围总线，后面的存取就要等到写存取结束。必须注意：如果在 CPU 对内部外围总线进行写存取后读内部存储器，因为不等运行结束就接受后面的存取，所以存取的顺序有可能发生变化（参照图 15.3）。

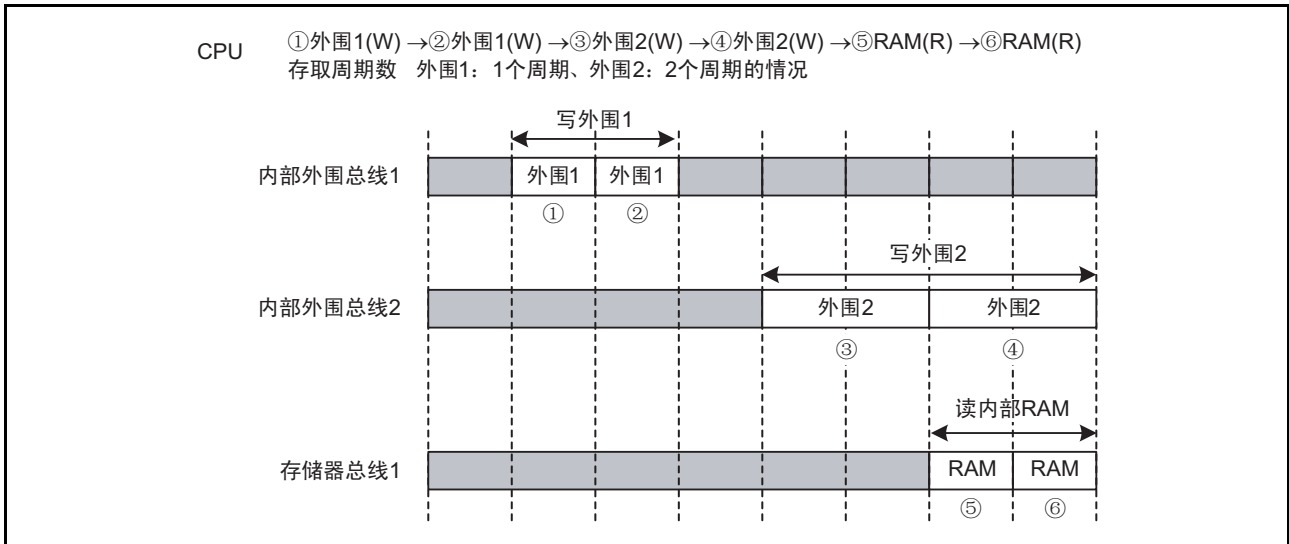


图 15.3 写缓冲器功能

15.2.6 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，在 CPU 从内部 ROM 取指令以及从内部 RAM 存取操作数的过程中，DMAC 能进行外围总线和外围总线之间的传送。并行运行的例子如图 15.4 所示。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 的过程中，DMAC 能使用内部主总线 2 同时存取外围总线。

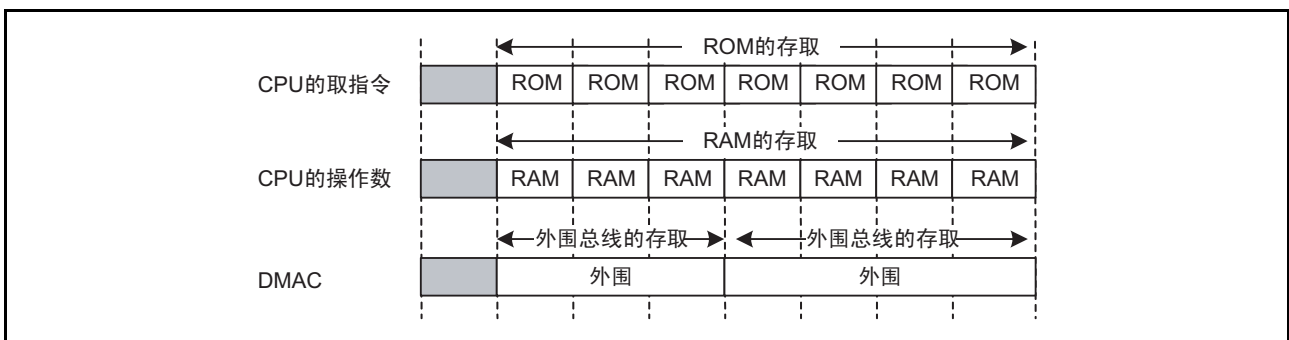


图 15.4 并行运行的例子

15.2.7 限制事项

- (a) 禁止将 RMPA 指令和字符串操作指令的操作对象数据分配到 I/O 寄存器，否则不保证运行。

15.3 寄存器说明

15.3.1 总线错误状态清除寄存器 (BERCLR)

地址 0008 1300h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	STSCLR
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	STSCLR	状态清除位	0: 无效 1: 清除总线错误状态寄存器	(W) (注1)
b7-b1	—	保留位	读写值都为“0”。	R/W

注 1. 只能写“1”，写“0”无效。

STSCLR 位 (状态清除位)

如果给此位写“1”，就清除总线错误状态寄存器 1 (BERSR1) 和总线错误状态寄存器 2 (BERSR2)。
写“0”无效，读取值为“0”。

15.3.2 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

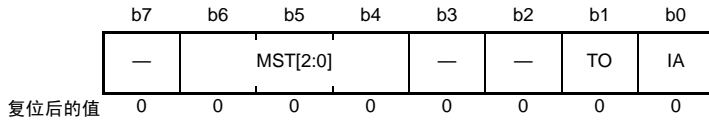
位	符号	位名	功能	R/W
b0	IGAEN	非法地址存取检测允许位	0: 禁止检测非法地址的存取 1: 允许检测非法地址的存取	R/W
b1	TOEN	超时检测允许位 (注1、注2)	0: 禁止检测总线超时 1: 允许检测总线超时	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

注 1. 如果在设定为禁止检测 (TOEN 位 =0) 后存取总线，总线就可能冻结。

注 2. 不能在检测超时错误的过程中将 TOEN 位置“0” (禁止检测)。

15.3.3 总线错误状态寄存器 1 (BERSR1)

地址 0008 1308h



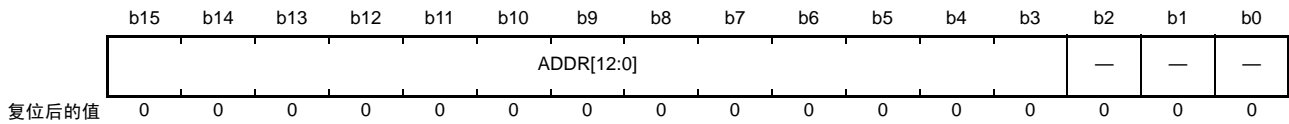
位	符号	位名	功能	R/W
b0	IA	非法地址存取位	0: 未发生非法地址的存取 1: 发生非法地址的存取	R
b1	TO	超时位	0: 未发生超时 1: 发生超时	R
b3-b2	—	保留位	读取值为“0”，写操作无效。	R
b6-b4	MST[2:0]	总线主控代码位	b6 b4 0 0 0: CPU 0 0 1: 保留 0 1 0: 保留 0 1 1: DTC/DMAC 1 0 0: 保留 1 0 1: 保留 1 1 0: 保留 1 1 1: 保留	R
b7	—	保留位	读取值为“0”，写操作无效。	R

MST[2:0] 位 (总线主控代码位)

这些位表示发生总线错误存取的总线主控。

15.3.4 总线错误状态寄存器 2 (BERSR2)

地址 0008 130Ah



位	符号	位名	功能	R/W
b2-b0	—	保留位	读取值为“0”，写操作无效。	R
b15-b3	ADDR[12:0]	总线错误发生地址位	发生总线错误存取的地址高 13 位 (以 512K 字节为单位)	R

15.3.5 总线优先级控制寄存器 (BUSPRI)

地址 0008 1310h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	BPFB[1:0]	BPHB[1:0]	BPGB[1:0]	BPIB[1:0]	BPRO[1:0]	BPRA[1:0]						
0	0	0	0	0	0	0	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	BPRA[1:0]	存储器总线 1 (内部 RAM) 的优先级控制位	b1 b0 0 0: 优先级固定 0 1: 优先级交替 1 0: 不能设定 1 1: 不能设定	R(W) (注 1)
b3-b2	BPRO[1:0]	存储器总线 2 (内部 ROM) 的优先级控制位	b3 b2 0 0: 优先级固定 0 1: 优先级交替 1 0: 不能设定 1 1: 不能设定	R(W) (注 1)
b5-b4	BPIB[1:0]	内部外围总线 1 的优先级控制位	b5 b4 0 0: 优先级固定 0 1: 优先级交替 1 0: 不能设定 1 1: 不能设定	R(W) (注 1)
b7-b6	BPGB[1:0]	内部外围总线 2 的优先级控制位	b7 b6 0 0: 优先级固定 0 1: 优先级交替 1 0: 不能设定 1 1: 不能设定	R(W) (注 1)
b9-b8	BPHB[1:0]	内部外围总线 4 的优先级控制位	b9 b8 0 0: 优先级固定 0 1: 优先级交替 1 0: 不能设定 1 1: 不能设定	R(W) (注 1)
b11-b10	BPFB[1:0]	内部外围总线 6 的优先级控制位	b11 b10 0 0: 优先级固定 0 1: 优先级交替 1 0: 不能设定 1 1: 不能设定	R(W) (注 1)
b15-b12	—	保留位	读写值都为“0”。	R/W

注 1. 只能在 DTC 和 DMAC 停止的状态下写 1 次此寄存器。如果写 2 次或者 2 次以上, 就不保证运行。

BPRA[1:0] 位 (存储器总线 1 (内部 RAM) 的优先级控制位)

这些位设定存储器总线 1 (内部 RAM) 的优先级。

在优先级固定的情况下, 优先级为内部主总线 2 > CPU 总线。

在优先级交替的情况下, 内部主总线 2 和 CPU 总线中已接受总线请求的一方的优先级变低。

BPRO[1:0] 位（存储器总线 2（内部 ROM）的优先级控制位）

这些位设定存储器总线 2（内部 ROM）的优先级。

在优先级固定的情况下，优先级为内部主总线 2 > CPU 总线。

在优先级交替的情况下，内部主总线 1 和 CPU 总线中已接受总线请求的一方的优先级变低。

BPIB[1:0] 位（内部外围总线 1 的优先级控制位）

这些位设定内部外围总线 1 的优先级。

在优先级固定的情况下，优先级为内部主总线 2 > 内部主总线 1。

在优先级交替的情况下，内部主总线 1 和内部主总线 2 中已接受总线请求的一方的优先级变低。

BPGB[1:0] 位（内部外围总线 2 的优先级控制位）

这些位设定内部外围总线 2 的优先级。

在优先级固定的情况下，优先级为内部主总线 2 > 内部主总线 1。

在优先级交替的情况下，内部主总线 1 和内部主总线 2 中已接受总线请求的一方的优先级变低。

BPHB[1:0] 位（内部外围总线 4 的优先级控制位）

这些位设定内部外围总线 4 的优先级。

在优先级固定的情况下，优先级为内部主总线 2 > 内部主总线 1。

在优先级交替的情况下，内部主总线 1 和内部主总线 2 中已接受总线请求的一方的优先级变低。

BPFB[1:0] 位（内部外围总线 6 的优先级控制位）

这些位设定内部外围总线 6 的优先级。

在优先级固定的情况下，优先级为内部主总线 2 > 内部主总线 1。

在优先级交替的情况下，内部主总线 1 和内部主总线 2 中已接受总线请求的一方的优先级变低。

15.4 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通知总线主控。

15.4.1 总线错误的种类

总线错误有非法地址存取和超时两种。

当存取非法地址时被检测为非法区域的存取；当总线存取在 768 个周期以内不结束时被检测为超时。

15.4.1.1 非法地址的存取

当总线错误监视允许寄存器的非法地址存取检测允许位被设定为有效（BEREN.IGAEN=1）并且发生以下存取时，发生非法地址存取。

- 当存取了非法地址区域时
哪个区域发生非法地址存取错误如表 15.5 所示。

15.4.1.2 超时

在总线错误监视允许寄存器的超时检测允许位被设定为有效（BEREN.TOEN=1）并且总线存取在 768 个周期以内不结束时，发生超时。

- 内部外围总线 2：在开始总线存取后，总线存取在外围模块时钟（PCLKB）的 768 个周期以内不结束时如果发生超时，就在 PCLKB 的 256 个周期内不接受总线主控的存取。
- 内部外围总线 6：在开始总线存取后，总线存取在 FlashIF 时钟（FCLK）的 768 个周期以内不结束时如果发生超时，就在 FCLK 的 256 个周期内不接受总线主控的存取。

15.4.2 发生总线错误时的运行

一旦发生总线错误，就将总线错误通知 CPU。当发生总线错误时，不保证运行。

- 向 CPU 通知总线错误的发生
发生中断。能通过 ICU.IERn 寄存器控制是否产生中断。

15.4.3 总线错误的发生条件

地址空间各区域发生的总线错误的种类如表 15.5 所示。

在没有发生总线错误的状态（清除总线错误状态寄存器 n（BERSRn）（n=1、2）的状态）下，如果检测到非法地址存取错误或者超时，就将当时的状态保存到 BERSRn 寄存器。一旦发生总线错误，即使以后还发生总线错误，如果没有清除 BERSRn，就不保存当时的状态。

当 2 个或者 2 个以上的总线主控同时发生总线错误时，只保存 1 个总线主控的信息。在发生总线错误后，状态保持到 BERSRn 寄存器被清除为止。

表 15.5 发生的总线错误种类

地址	内容		种类			
			非法地址存取		超时	
	内部 ROM 模式		内部 ROM 模式		内部 ROM 模式	
	有效	无效	有效	无效	有效	无效
0000 0000h ~ 0007 FFFFh	存储器总线 1		—		—	
0008 0000h ~ 0008 7FFFh	内部外围总线 1		—		—	
0008 8000h ~ 0009 FFFFh	内部外围总线 2		△		—	
000A 0000h ~ 000B FFFFh	保留区		△		—	
000C 0000h ~ 000D FFFFh	内部外围总线 4		△		—	
000E 0000h ~ 000F FFFFh	保留区		—		—	
0010 0000h ~ 00FF FFFFh	内部外围总线 6	保留区	△	○	—	—
0100 0000h ~ FFFE FFFFh	保留区		—		—	
FFFF 0000h ~ FFFF FFFFh	存储器总线 2	保留区	—	—	—	—

—：不发生总线错误。

△：总线错误不确定。

○：发生总线错误。

注． 装载的内部 RAM、数据闪存和内部 ROM 的容量因产品而不同。有关各产品的规格，请参照“33. RAM”、“34. ROM（保存代码的闪存）”和“35. E2 数据闪存（保存数据的闪存）”。

16. 存储器保护单元 (MPU)

16.1 概要

存储器保护单元内置于 RX CPU，对 CPU 存取的全部地址空间 (0000 0000h ~ FFFF FFFFh) 进行地址校验。

MPU 最多能设定 8 个区域，根据各区域的存取控制信息来允许存取。如果检测到设定区域外的存取，默认值就为发生存储器保护错误。

各区域的存取控制信息对应“允许读”、“允许写”和“允许执行”，在 CPU 的处理器模式为用户模式时有效，而在处理器模式为管理模式时不进行存储器保护。

存储器保护单元的规格和框图分别如表 16.1 和图 16.1 所示。

表 16.1 存储器保护的规格

规格	内容
存储器保护对象区域和处理器模式	0000 0000h ~ FFFF FFFFh (用户模式) 在处理器模式为管理模式时不进行存储器保护。
区域数	8 个
页大小 (最小保护单位)	16 字节
各区域的地址指定	通过起始页号和结束页号进行设定。
各区域的有效设定	通过区域 n 结束页号寄存器 (REPAGEn) 的有效位 (V) 设定各区域的有效或者无效 (n=0 ~ 7)。
各区域的存取控制信息	指令执行: 允许执行 操作数存取: 允许读、允许写
存储器保护的开始	在将存储器保护功能设定为有效后，通过转移到用户模式，开始监视存取。
存储器保护错误的处理	发生存取异常。
存储器保护错误的发生地址	指令执行地址: 将 PC 压栈。 操作数存取地址: 保存到数据存储器保护错误地址寄存器 (MPDEA)。
存储器保护错误源的判断	将错误源保存到存储器保护错误状态寄存器 (MPESTS)。
后台区域的设定	能给后台区域 (全部地址空间) 设定存取控制信息。
区域重叠的处理	如果给某地址重叠设定区域并且各区域的存取控制信息不同，就优先执行被允许的操作。

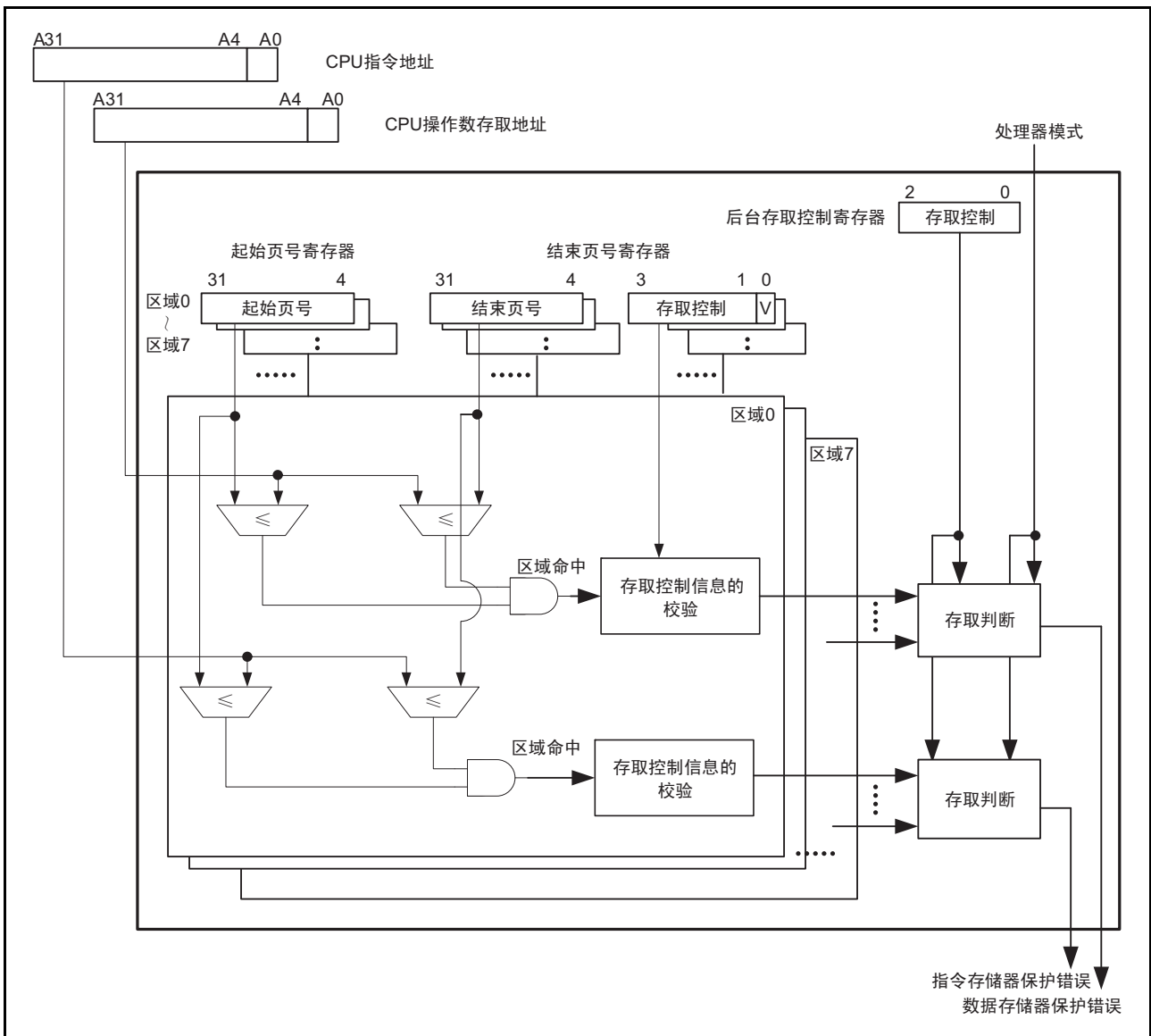


图 16.1 存储器保护单元的框图

16.1.1 存取控制的种类

存取控制有“允许执行指令”、“允许读操作数”和“允许写操作数”共3种。只对用户模式中的程序进行存取控制的违反检测，而对管理模式中的程序不进行存取控制的违反检测。

16.1.2 存取控制区域

最多能定义8个存取控制区域。通过区域n起始页号寄存器 (RSPAGEn) 和区域n结束页号寄存器 (REPAGEn) 设定各存取控制区域的范围 (n=0 ~ 7)。

页是将地址空间按16字节进行分割而成的存取控制的最小单位。地址 [31:0] 的高28位 ([31:4]) 对应页号。通过 REPAGEn 寄存器指定各区域的存取控制信息以及是否将该区域设定为有效。

16.1.3 后台区域

后台区域为全部地址空间 (0000 0000h ~ FFFF FFFFh)，通过后台存取控制寄存器 (MPBAC) 设定后台区域的存取控制信息。后台区域的存取控制信息不同于8个存取控制区域的信息，在存储器保护功能有效 (MPEN.MPEN 位为“1”) 时有效。

16.1.4 区域的重叠

多个区域重叠时的存取控制信息为重叠区域 (包括后台区域) 存取控制位的逻辑或，优先执行设定为允许的操作。

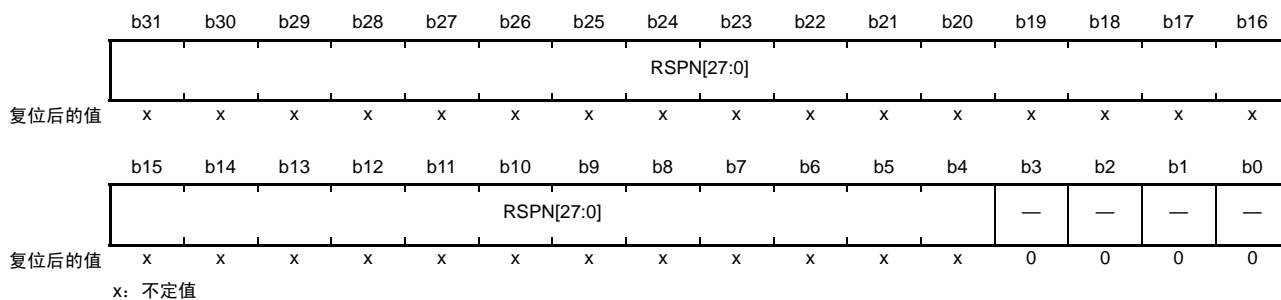
16.1.5 跨区域的指令和数据

对于跨不同存取控制设定的区域而分配的指令和数据，存储器保护错误的检测为不确定，因此不能跨不同存取控制设定的区域分配指令和数据。

16.2 寄存器说明

16.2.1 区域 n 起始页号寄存器 (RSPAGEn) (n=0 ~ 7)

地址 RSPAGE0 0008 6400h、RSPAGE1 0008 6408h、RSPAGE2 0008 6410h、RSPAGE3 0008 6418h
RSPAGE4 0008 6420h、RSPAGE5 0008 6428h、RSPAGE6 0008 6430h、RSPAGE7 0008 6438h



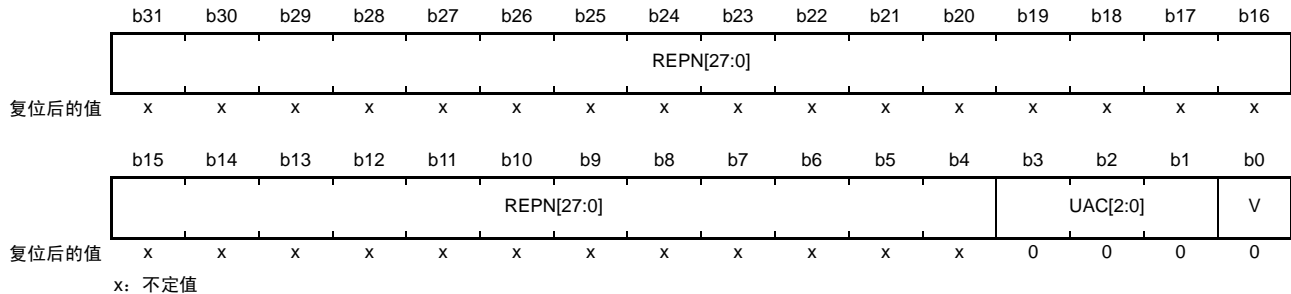
位	符号	位名	说明	R/W
b3-b0	—	保留位	读写值都为“0”。	R/W
b31-b4	RSPN[27:0]	区域起始页号位	用于区域判断的区域起始页号信息	R/W

RSPN[27:0] 位 (区域起始页号位)

这些位设定区域起始页号。

16.2.2 区域 n 结束页号寄存器 (REPAGEn) (n=0 ~ 7)

地址 REPAGE0 0008 6404h、REPAGE1 0008 640Ch、REPAGE2 0008 6414h、REPAGE3 0008 641Ch
REPAGE4 0008 6424h、REPAGE5 0008 642Ch、REPAGE6 0008 6434h、REPAGE7 0008 643Ch



位	符号	位名	说明	R/W
b0	V	有效位	0: 区域设定无效 1: 区域设定有效	R/W
b3-b1	UAC[2:0]	用户模式的存取控制位	b3 0: 禁止读 1: 允许读 b2 0: 禁止写 1: 允许写 b1 0: 禁止执行 1: 允许执行	R/W
b31-b4	REPN[27:0]	区域结束页号	用于区域判断的区域结束页号信息	R/W

V 位 (有效位)

此位选择将对应的区域设定为有效或者无效。

如果通过区域无效化寄存器 (MPOPI) 将全部存取控制区域设定为无效 (无效化), V 位就变为“0”。

UAC[2:0] 位 (用户模式的存取控制位)

这些位设定用户模式中的存取控制。

REPN[27:0] 位 (区域结束页号位)

这些位设定区域结束页号, 设定值必须大于或者等于对应区域的起始页号。区域结束页号也是存储器保护对象区域。

16.2.3 存储器保护功能有效化寄存器 (MPEN)

地址 0008 6500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MPEN
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	说明	R/W
b0	MPEN	存储器保护功能有效化位	1: 存储器保护功能有效 0: 存储器保护功能无效	R/W
b31-b1	—	保留位	读写值都为“0”。	R/W

MPEN 位 (存储器保护功能有效化位)

此位选择将存储器保护功能设定为有效或者无效。

在给 MPEN 位写“1”后，通过执行转移到用户模式的转移指令 (RTE、RTFI)，开始由 CPU 的存储器保护进行的地址校验。

16.2.4 后台存取控制寄存器 (MPBAC)

地址 0008 6504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
	—	—	—	—	—	—	—	—	—	—	—	—	UBAC[2:0]			—
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	说明	R/W
b0	—	保留位	读写值都为“0”。	R/W
b3-b1	UBAC[2:0]	用户模式的后台存取控制位	b3 0: 禁止读 1: 允许读 b2 0: 禁止写 1: 允许写 b1 0: 禁止执行 1: 允许执行	R/W
b31-b4	—	保留位	读写值都为“0”。	R/W

UBAC[2:0] 位 (用户模式的后台存取控制位)

这些位设定用户模式中的后台存取控制。

16.2.5 存储器保护错误状态清除寄存器 (MPECLR)

地址 0008 6508h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLR
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	说明	R/W
b0	CLR	错误状态清除位	【读时】 0: 读 (固定) 【写时】 0: 不进行任何操作 1: 将 MPESTS.DRW 位、 MPESTS.DA 位和 MPESTS.IA 位置 "0"。	R/W
b31-b1	—	保留位	读写值都为 "0"。	R/W

CLR 位 (错误状态清除位)

将存储器保护错误状态寄存器 (MPESTS) 的数据读写位 (DRW)、数据存储器保护错误发生位 (DA) 和指令存储器保护错误发生位 (IA) 置 "0"。

16.2.6 存储器保护错误状态寄存器 (MPESTS)

地址 0008 650Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	DRW	DA	IA
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	说明	R/W
b0	IA	指令存储器保护错误发生位	0: 未发生指令存储器保护错误 1: 发生指令存储器保护错误	R
b1	DA	数据存储器保护错误发生位	0: 未发生数据存储器保护错误 1: 发生数据存储器保护错误	R
b2	DRW	数据读写位	0: 读数据 1: 写数据	R
b31-b3	—	保留位	读写值都为“0”。	R/W

IA 位 (指令存储器保护错误发生位)

此位表示因执行指令而发生存储器保护错误的状态。

只有通过将存储器保护错误状态清除寄存器 (MPECLR) 的错误状态清除位 (CLR) 置“1”，IA 位才变为“0”。

DA 位 (数据存储器保护错误发生位)

此位表示因存取操作数而发生存储器保护错误的状态。

只有通过将存储器保护错误状态清除寄存器 (MPECLR) 的错误状态清除位 (CLR) 置“1”，DA 位才变为“0”。

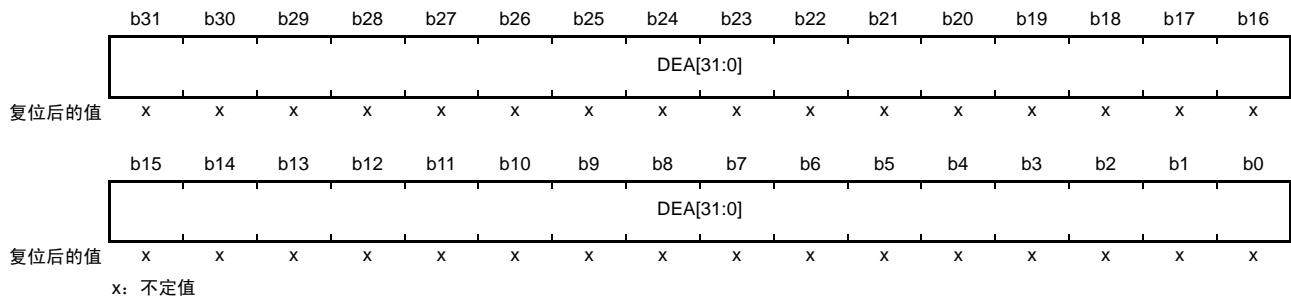
DRW 位 (数据读写位)

此位表示因存取操作数而发生存储器保护错误的存取读写属性，只在 DA 位为“1”时有效。

通过将存储器保护错误状态清除寄存器 (MPECLR) 的错误状态清除位 (CLR) 置“1”，DRW 位变为“0”。

16.2.7 数据存储器保护错误地址寄存器 (MPDEA)

地址 0008 6514h



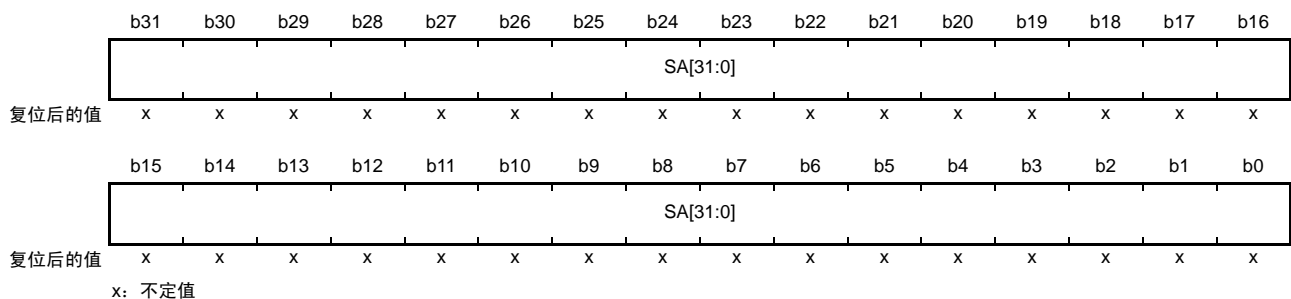
位	符号	位名	说明	R/W
b31-b0	DEA[31:0]	数据存储器保护错误地址位	数据存储器保护错误地址	R

DEA[31:0] 位 (数据存储器保护错误地址位)

这些位保持因存取操作数而发生存储器保护错误的地址。

16.2.8 区域搜索地址寄存器 (MPSA)

地址 0008 6520h



位	符号	位名	说明	R/W
b31-b0	SA[31:0]	区域搜索地址位	用于区域搜索的地址	R/W

SA[31:0] 位 (区域搜索地址位)

这些位通过区域搜索来设定和区域 n 起始页号寄存器 (RSPAGEn) 的区域起始地址、区域 n 结束页号寄存器 (REPAGE n) 的区域结束地址进行比较的地址。

16.2.9 区域搜索操作寄存器 (MPOPS)

地址 0008 6524h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	S
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	说明	R/W
b0	S	区域搜索操作位	【读时】 0: 读 (固定) 【写时】 0: 不进行任何操作 1: 进行区域搜索	R/W
b15-b1	—	保留位	读写值都为“0”。	R/W

S 位 (区域搜索操作位)

存储器保护单元通过将 S 位置“1”进行区域搜索。将区域搜索地址寄存器 (MPSA) 指定的地址和各区域的地址信息进行比较, 搜索命中的区域。

将搜索结果保存到数据命中区域寄存器 (MHITD) 的数据命中区域位 (HITD[7:0])。另外, 将命中区域的存取控制位的逻辑或保存到用户模式中的数据命中区域存取控制位 (UHACD[2:0])。

16.2.10 区域无效化操作寄存器 (MPOPI)

地址 0008 6526h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INV
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	说明	R/W
b0	INV	区域无效化操作启动位	【读】 0: 读 (固定) 【写】 0: 不进行任何操作 1: 使全部存取控制区域无效 (无效化)	R/W
b15-b1	—	保留位	读写值都为“0”。	R/W

INV 位 (区域无效化操作启动位)

通过将 INV 位置“1”, 将全部区域 n 结束页号寄存器 (REPAGEn) 的有效位 (V) 置“0”。在将 REPAGEn.V 位置“0”后, 后台区域的存取控制以外的设定无效。

16.2.11 指令命中区域寄存器 (MHITI)

地址 0008 6528h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
—	—	—	—	—	—	—	—	HITI[7:0]							—	—
复位后的值	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	
—	—	—	—	—	—	—	—	—	—	—	—	UHACI[2:0]			—	
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	说明	R/W
b0	—	保留位	读写值都为“0”。	R/W
b3-b1	UHACI[2:0]	用户模式的指令命中区域存取控制位	b3 0: 禁止读 1: 允许读 b2 0: 禁止写 1: 允许写 b1 0: 禁止执行 1: 允许执行	R
b15-b4	—	保留位	读写值都为“0”。	R/W
b23-b16	HITI[7:0]	指令命中区域位	指令存储器保护错误发生位 (MPESTS.IA) 为“1”的情况 当 [b23:b16] 位为“0000 0000b”时, 表示在后台区域发生指令存储器保护错误。 上述以外 b23 0: 区域 7 未发生指令存储器保护错误 1: 区域 7 发生指令存储器保护错误 b22 0: 区域 6 未发生指令存储器保护错误 1: 区域 6 发生指令存储器保护错误 b21 0: 区域 5 未发生指令存储器保护错误 1: 区域 5 发生指令存储器保护错误 b20 0: 区域 4 未发生指令存储器保护错误 1: 区域 4 发生指令存储器保护错误 b19 0: 区域 3 未发生指令存储器保护错误 1: 区域 3 发生指令存储器保护错误 b18 0: 区域 2 未发生指令存储器保护错误 1: 区域 2 发生指令存储器保护错误 b17 0: 区域 1 未发生指令存储器保护错误 1: 区域 1 发生指令存储器保护错误 b16 0: 区域 0 未发生指令存储器保护错误 1: 区域 0 发生指令存储器保护错误	R
b31-b24	—	保留位	读写值都为“0”。	R/W

UHACI[2:0] 位 (用户模式的指令命中区域存取控制位)

UHACI[2:0] 位保持发生指令存储器保护错误的区域的用户模式存取控制位 (REPAGEn.UAC[2:0])。如果重叠区域发生错误, 这些位就保持对应区域 (包括后台区域) 的用户模式存取控制位的逻辑或。

HITI[7:0] 位 (指令命中区域位)

HITI[7:0] 位表示发生指令存储器保护错误的区域。如果在后台区域发生指令存储器保护错误, 就将 HITI[7:0] 位置 “0000 0000b”。

16.2.12 数据命中区域寄存器 (MHITD)

地址 0008 652Ch

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16		
—	—	—	—	—	—	—	—	HITD[7:0]								—	—
复位后的值	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		
—	—	—	—	—	—	—	—	—	—	—	—	UHACD[2:0]			—		
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	说明	R/W
b0	—	保留位	读写值都为“0”。	R/W
b3-b1	UHACD[2:0]	用户模式的数据命中区域存取控制位	b3 0: 禁止读 1: 允许读 b2 0: 禁止写 1: 允许写 b1 0: 禁止执行 1: 允许执行	R
b15-b4	—	保留位	读写值都为“0”。	R/W
b23-b16	HITD[7:0]	数据命中区域位	数据存储器保护错误发生位 (DA) 为“1”的情况 当 [b23:b16] 位为“0000 0000b”时, 表示在后台区域发生数据存储器保护错误。 上述以外 b23 0: 区域 7 未发生数据存储器保护错误并且未搜索到命中区域 1: 区域 7 发生数据存储器保护错误或者搜索到命中区域 b22 0: 区域 6 未发生数据存储器保护错误并且未搜索到命中区域 1: 区域 6 发生数据存储器保护错误或者搜索到命中区域 b21 0: 区域 5 未发生数据存储器保护错误并且未搜索到命中区域 1: 区域 5 发生数据存储器保护错误或者搜索到命中区域 b20 0: 区域 4 未发生数据存储器保护错误并且未搜索到命中区域 1: 区域 4 发生数据存储器保护错误或者搜索到命中区域 b19 0: 区域 3 未发生数据存储器保护错误并且未搜索到命中区域 1: 区域 3 发生数据存储器保护错误或者搜索到命中区域 b18 0: 区域 2 未发生数据存储器保护错误并且未搜索到命中区域 1: 区域 2 发生数据存储器保护错误或者搜索到命中区域 b17 0: 区域 1 未发生数据存储器保护错误并且未搜索到命中区域 1: 区域 1 发生数据存储器保护错误或者搜索到命中区域 b16 0: 区域 0 未发生数据存储器保护错误并且未搜索到命中区域 1: 区域 0 发生数据存储器保护错误或者搜索到命中区域	R
b31-b24	—	保留位	读写值都为“0”。	R/W

UHACD[2:0] 位 (用户模式的数据命中区域存取控制位)

UHACD[2:0] 位保持发生数据存储器保护错误的区域或者在区域搜索中命中的区域的用户模式存取控制位 (REPAGEn.UAC[2:0])。

如果重叠区域发生错误或者在区域搜索中命中, 这些位就保持对应区域 (包括后台区域) 的用户模式存取控制位的逻辑或。

HITD[7:0] 位 (数据命中区域位)

HITD[7:0] 位表示发生数据存储器保护错误的区域或者在区域搜索中命中的区域。如果在后台区域发生数据存储器保护错误, 就将 HITD[7:0] 位置 “0000 0000b”。

注. 在用户模式中, 如果在存取存储器保护单元的寄存器后发生数据存储器保护错误, MHITD 寄存器的值就变为 “0000 0000h”。

16.3 功能

16.3.1 存储器保护功能

存储器保护功能是根据存取控制区域和后台区域设定的存取控制信息来监视用户模式中的程序是否进行了违反存取控制信息的存取。如果检测到违反的存取控制（存储器保护错误），存储器保护单元就将该信息通知 CPU，CPU 开始存取异常处理。

通过将存储器保护功能有效化寄存器（MPEN）的存储器保护功能有效化位（MPEN）置“1”，存储器保护功能变为有效。

如果检测到违反的指令执行，就发生指令存储器保护错误；如果检测到违反的操作数存取（读写），就发生数据存储器保护错误。在发生数据存储器保护错误时，不执行引起违反的存取控制的操作数存取。

16.3.2 区域搜索功能

区域搜索功能是调查某个特定地址命中 8 个存取控制区域中的哪一个区域以及该地址的存取控制信息（允许执行、允许读、允许写）是如何设定的。

通过将区域搜索操作寄存器（MPOPS）的区域搜索操作位（S）置“1”，将区域搜索地址寄存器（MPSA）指定的地址和各区域的地址进行比较。执行区域搜索后的数据命中区域寄存器（MHITD）表示命中区域和各区域存取控制信息的逻辑或。

16.3.3 存储器保护单元相关寄存器的保护

不能通过 CPU 的操作数存取以外的方法对存储器保护单元的相关寄存器进行存取。并且，只能在管理模式中对存储器保护单元的相关寄存器进行存取。如果在用户模式中通过 CPU 的操作数存取对存储器保护单元的相关寄存器进行存取，就发生数据存储器保护错误，与存储器保护功能是否有效无关。

16.3.4 存储器保护功能的存取判断流程

数据存取判断流程和指令存取判断流程分别如图 16.2 和图 16.3 所示。

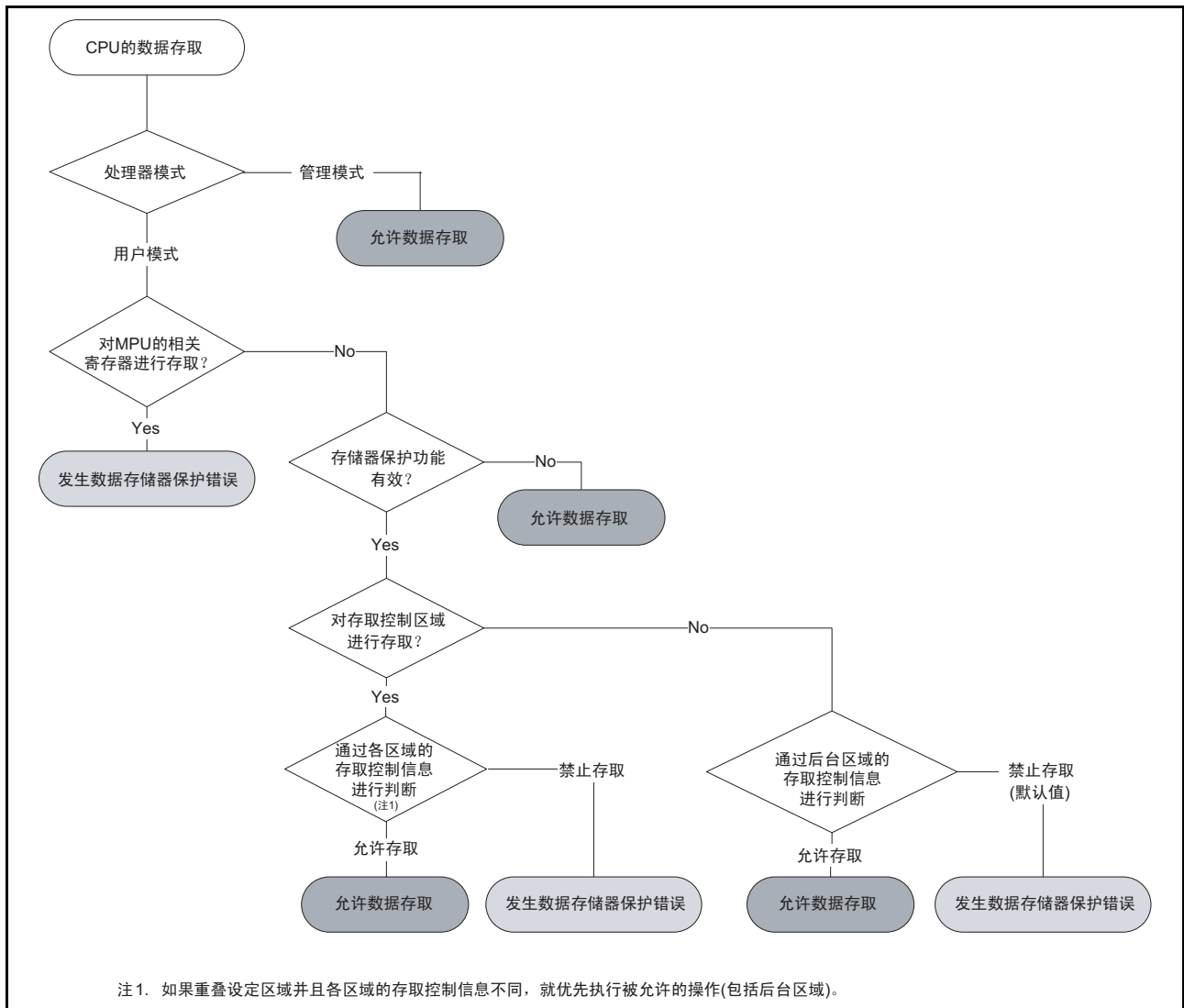


图 16.2 数据存取判断流程

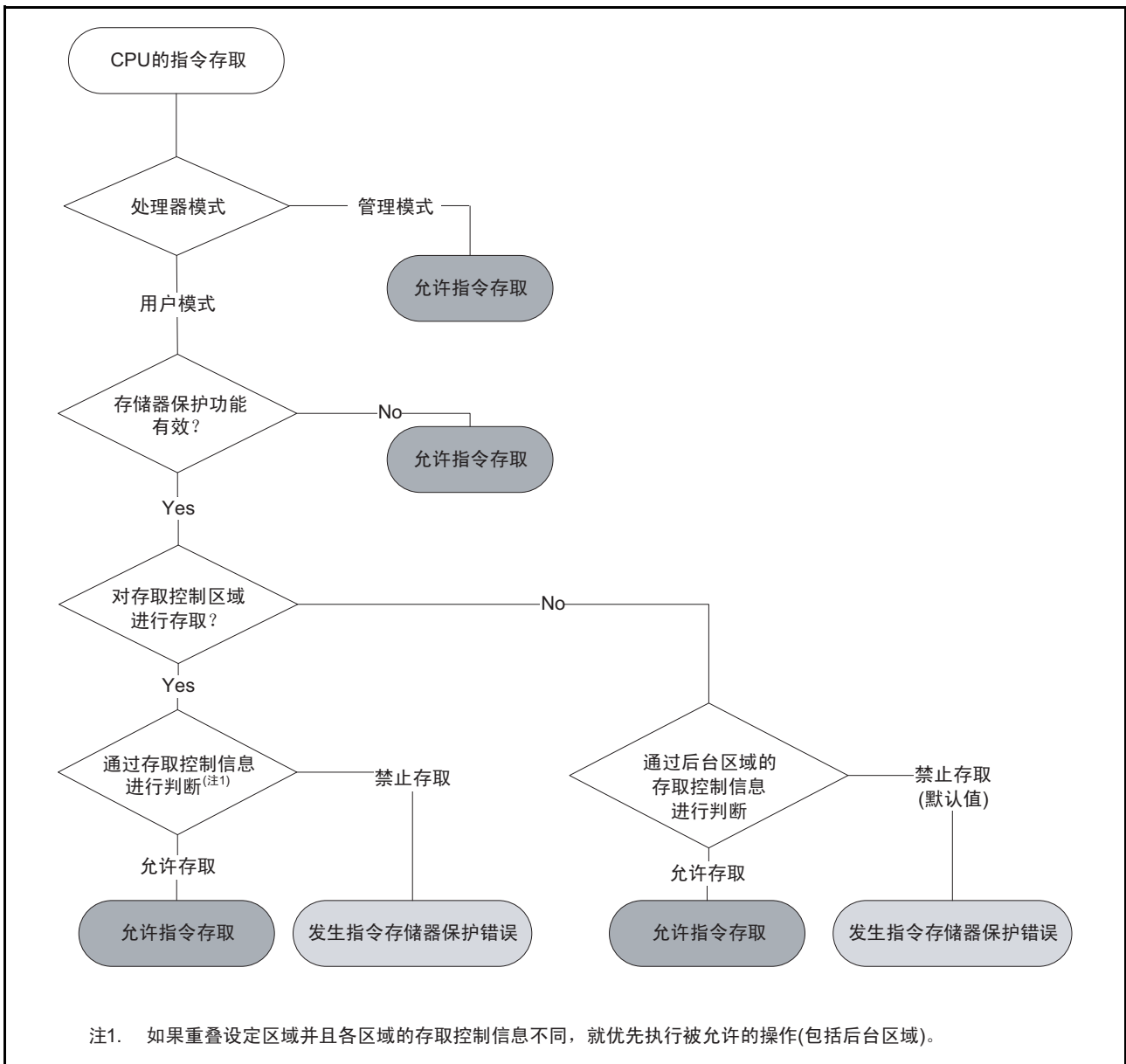


图 16.3 指令存取判断流程

16.4 存储器保护功能的使用步骤

16.4.1 存取控制信息的设定

在管理模式中设定各区域的存取控制信息。

通过区域 n 起始页号寄存器 (RSPAGEn) 和区域 n 结束页号寄存器 (REPAGEn)，最多设定 8 个存取控制区域 ($n=0 \sim 7$)。

通过后台存取控制寄存器 (MPBAC) 设定后台存取控制区域。

16.4.2 存储器保护功能的有效化

在管理模式中将存储器保护功能有效化寄存器 (MPEN) 的存储器保护功能有效化位 (MPEN) 置“1”。

16.4.3 向用户模式的转移

在改写存储器保护单元相关寄存器的设定后，必须在转移到用户模式前先读最后写的存储器保护单元相关寄存器并且确认值已被设定，然后转移到用户模式。

按照以下的任意方法从管理模式转移到用户模式：

- 在将压栈的处理器状态字 (PSW) 的处理器模式设定位 (PM) 置“1” (设定为用户模式) 后执行 RTE 指令。
- 在将备份 PSW (BPSW) 的 PM 位置“1”后执行 RTFI 指令。

注. 通过 MVTC 指令和 POPC 指令进行 PSW.PM 位的改写无效，而必须通过 RTE 指令或者 RTFI 指令更改 PSW.PM 位的值。

通过转移到用户模式，存储器保护单元开始对 CPU 的指令执行地址和操作数存取地址进行校验。

16.4.4 发生存储器保护错误时的处理

如果检测到违反的存取控制信息 (存储器保护错误)，CPU 就开始存取异常处理。存取异常处理中的 CPU 运行的详细内容请参照“13. 异常处理”。

在异常处理程序中，确认存储器保护错误状态寄存器 (MPESTS) 的指令存储器保护错误发生位 (IA) 和数据存储器保护错误发生位 (DA)，判断是指令存储器保护错误还是数据存储器保护错误。

在确认后，通过将存储器保护错误状态清除寄存器 (MPECLR) 的错误状态清除位 (CLR) 置“1”，清除 MPESTS 寄存器。

(1) 发生数据存储器保护错误时的情况

通过 CPU 的存取异常处理，将发生存储器保护错误的指令地址压栈，将发生存储器保护错误的操作数存取地址保存到数据存储器保护错误地址寄存器 (MPDEA) 以及将发生存储器保护错误的区域信息保存到数据命中区域寄存器 (MHITD)。

- **存取有效区域0~7却违反存取控制的情况**
与发生错误的区域号对应的数据命中区域位 (MHITD.HITD[7:0]) 变为“1”，并且发生错误的区域存取控制信息的逻辑或被设定到用户模式的数据命中区域存取控制位 (MHITD.UHACD[2:0])。
- **存取有效区域0~7以外的区域并且违反后台区域的存取控制的情况**
数据命中区域位 (MHITD.HITD[7:0]) 变为“0000 0000b”，并且后台区域的存取控制信息被设定到用户模式的数据命中区域存取控制位 (MHITD.UHACD[2:0])。

能通过参照这些信息进行特定错误原因等的处理。

(2) 发生指令存储器保护错误的情况

通过 CPU 的存取异常处理，将发生存储器保护错误的指令地址压栈以及将发生存储器保护错误的区域信息保存到指令命中区域寄存器 (MHITI)。

- **存取有效区域0~7却违反存取控制的情况**
与发生错误的区域号对应的指令命中区域位 (MHITI.HITI[7:0]) 变为“1”，并且发生错误的区域存取控制信息的逻辑或被设定到用户模式的指令命中区域存取控制位 (MHITI.UHACI[2:0])。
- **存取有效区域0~7以外的区域并且违反后台区域的存取控制的情况**
指令命中区域位 (MHITI.HITI[7:0]) 变为“0000 0000b”，并且后台区域的存取控制被设定到用户模式的指令命中区域存取控制位 (MHITI.UHACI[2:0])。

能通过参照这些信息进行特定错误原因等的处理。

17. DMA 控制器 (DMACA)

RX63T 群内置 4 个通道的 DMAC (Direct Memory Access Controller)。

DMAC 不通过 CPU 进行数据传送。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。

17.1 概要

DMAC 的规格和框图分别如表 17.1 和图 17.1 所示。

表 17.1 DMAC 的规格

项目		内容			
通道数		4 个通道 (DMAC _m (m=0 ~ 3))			
传送空间		512M 字节 (00000000h ~ 0FFFFFFFh 和 F0000000h ~ FFFFFFFFh 中除保留区以外的区域)			
最大传送数据量		1M 数据 (块传送模式的最大总传送量: 1024 个数据 × 1024 块)			
DMA 启动源		<ul style="list-style-type: none"> 各通道可选择启动源。 软件触发 外围模块中断请求 / 外部中断输入引脚的触发输入 (注 1) 			
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			
	块大小	数据量: 1 ~ 1024 个数据			
传送模式	正常传送模式	<ul style="list-style-type: none"> 1 次 DMA 传送请求进行 1 个数据的传送。 能不指定总数据传送量 (自由运行模式)。 			
	重复传送模式	<ul style="list-style-type: none"> 1 次 DMA 传送请求进行 1 个数据的传送。 如果传送由传送源或者传送目标设定的重复大小的数据, 就返回到开始传送时的地址。 重复大小最多能设定 1024 次。 			
	块传送模式	<ul style="list-style-type: none"> 1 次 DMA 传送请求进行 1 块数据的传送。 块大小最多能设定 1024 个数据。 			
选择功能	扩展重复区域功能	<ul style="list-style-type: none"> 能通过固定传送地址寄存器的高位的值, 重复设定特定范围的地址。 能分别给传送源和传送目标设定 2 字节到 128M 字节的扩展重复区域。 			
	中断请求	<table border="1"> <tr> <td>传送结束中断</td> <td>当传送计数器设定的数据量传送结束时, 发生此中断。</td> </tr> <tr> <td>传送退出结束中断</td> <td>当重复大小的数据传送结束或者扩展重复区域发生上溢时, 发生此中断。</td> </tr> </table>	传送结束中断	当传送计数器设定的数据量传送结束时, 发生此中断。	传送退出结束中断
传送结束中断	当传送计数器设定的数据量传送结束时, 发生此中断。				
传送退出结束中断	当重复大小的数据传送结束或者扩展重复区域发生上溢时, 发生此中断。				
低功耗功能		能设定为模块停止状态。			

注 1. DMAC 的启动源请参照“14. 中断控制器 (ICUb)”的“表 14.3 中断向量表”。

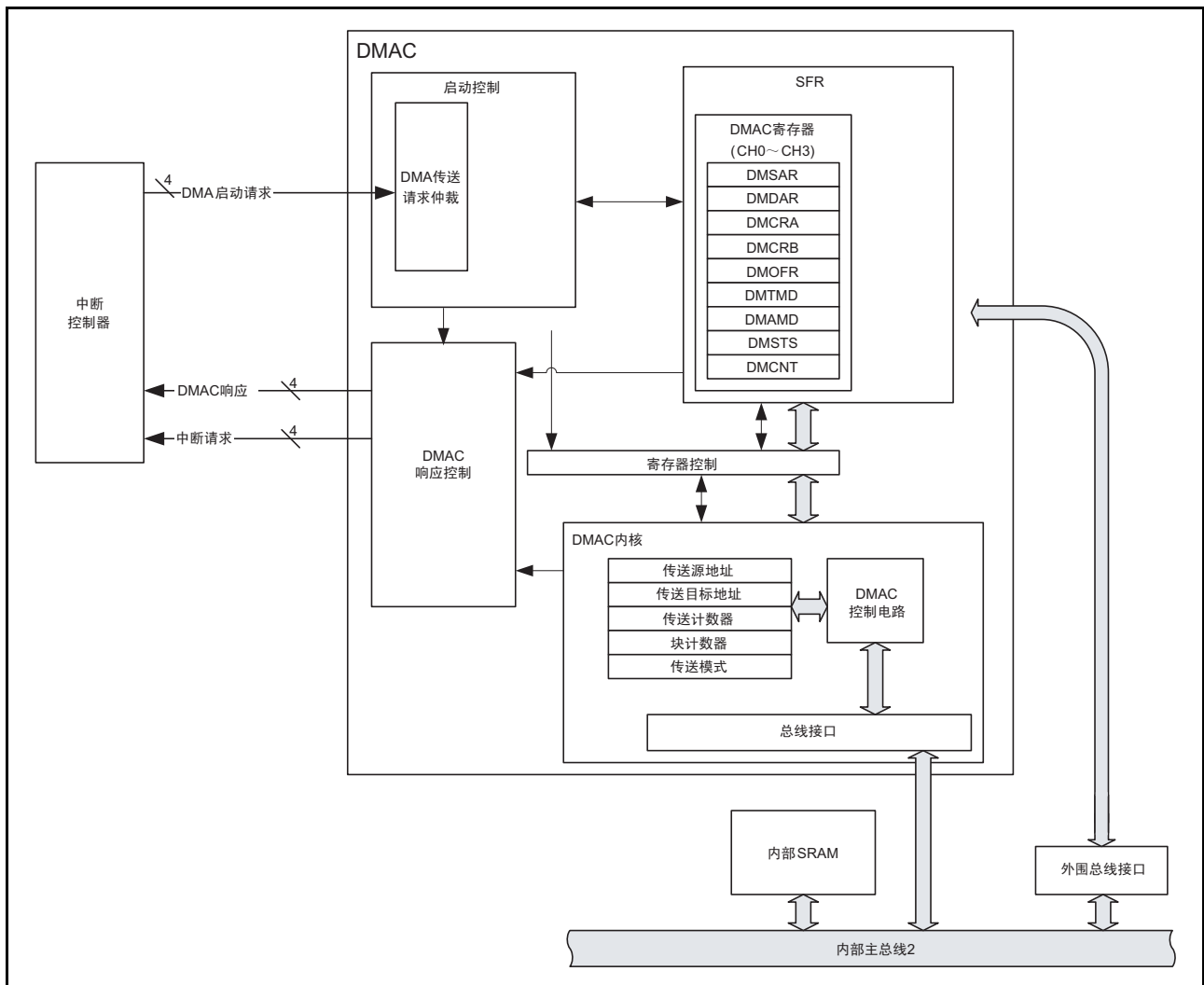
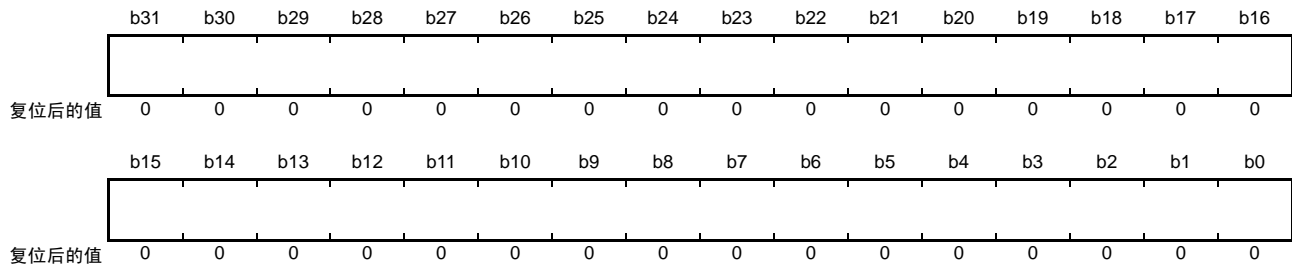


图 17.1 DMAC 的框图

17.2 寄存器说明

17.2.1 DMA 传送源地址寄存器 (DMSAR)

地址 DMAC0.DMSAR 0008 2000h、DMAC1.DMSAR 0008 2040h、DMAC2.DMSAR 0008 2080h、DMAC3.DMSAR 0008 20C0h



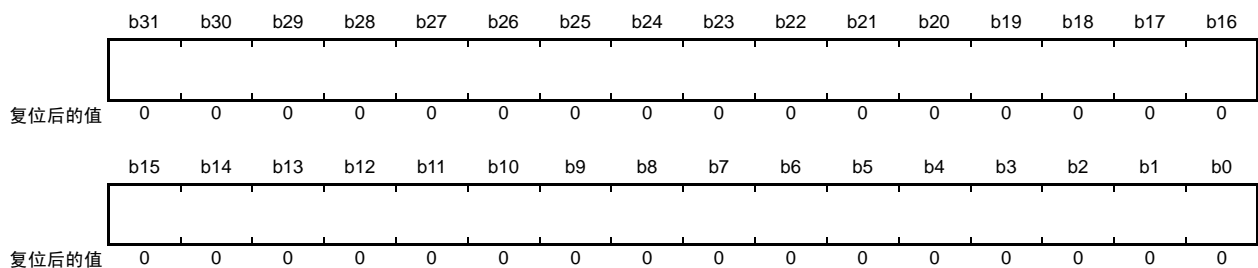
位	功能	设定范围	R/W
b31-b0	设定传送源的起始地址。	00000000h ~ 0FFFFFFFh (256M 字节) F0000000h ~ FFFFFFFFh (256M 字节)	R/W

在设定 DMSAR 寄存器时, 必须在禁止启动 DMAC (DMAST.DMST 位 =0) 或者禁止 DMA 传送 (DMCNT.DTE 位 =0) 时进行写操作。

bit31 ~ bit29 的设定值无效, 将 bit28 的值扩展到 bit31 ~ bit29。如果读 DMSAR 寄存器, 就会读到位扩展后的值。

17.2.2 DMA 传送目标地址寄存器 (DMDAR)

地址 DMAC0.DMDAR 0008 2004h、DMAC1.DMDAR 0008 2044h
DMAC2.DMDAR 0008 2084h、DMAC3.DMDAR 0008 20C4h



位	功能	设定范围	R/W
b31-b0	设定传送目标的起始地址。	00000000h ~ 0FFFFFFFh (256M 字节) F0000000h ~ FFFFFFFFh (256M 字节)	R/W

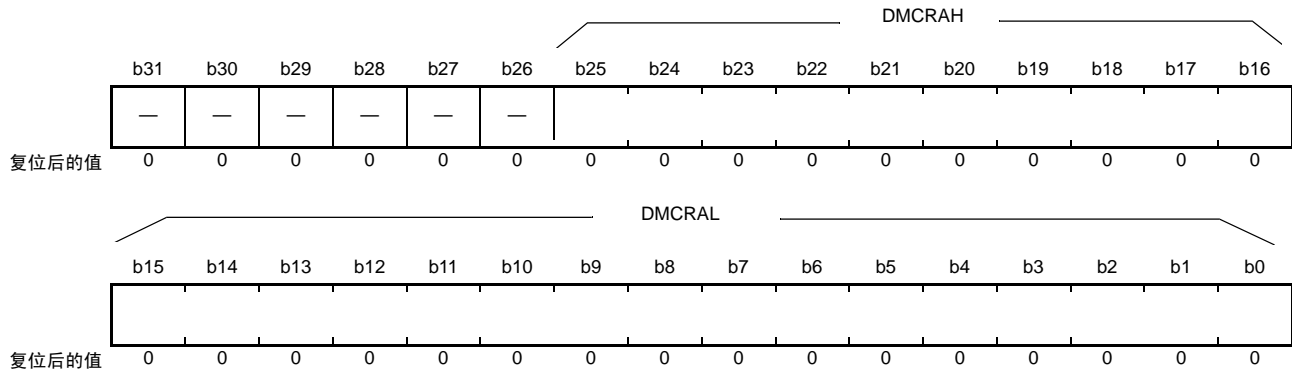
在设定 DMDAR 寄存器时, 必须在禁止启动 DMAC (DMAST.DMST 位 =0) 或者禁止 DMA 传送 (DMCNT.DTE 位 =0) 时进行写操作。

bit31 ~ bit29 的设定值无效, 将 bit28 的值扩展到 bit31 ~ bit29。如果读 DMDAR 寄存器, 就会读到位扩展后的值。

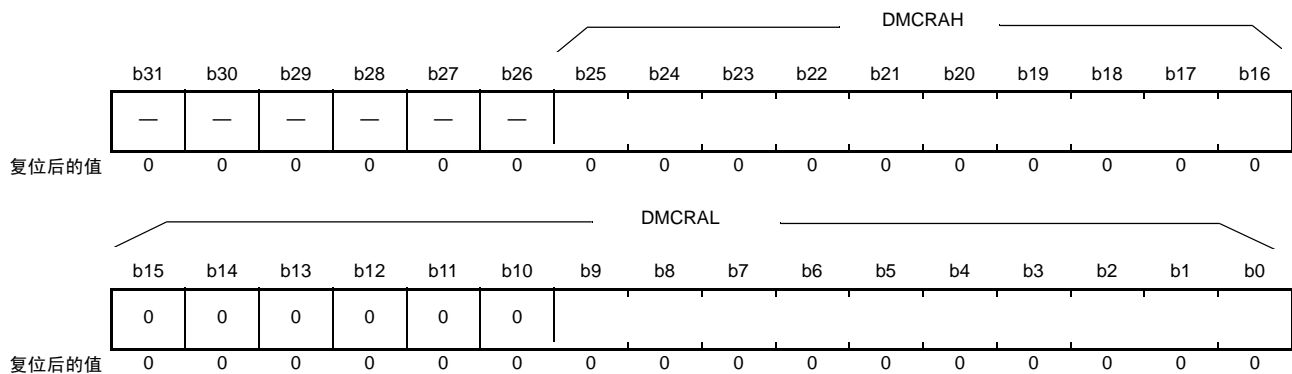
17.2.3 DMA 传送计数寄存器 (DMCRA)

地址 DMAC0.DMCRA 0008 2008h、DMAC1.DMCRA 0008 2048h、DMAC2.DMCRA 0008 2088h、DMAC3.DMCRA 0008 20C8h

- 正常传送模式



- 重复传送模式、块传送模式



符号	位名	功能	R/W
DMCRAL	传送计数的低位	设定传送次数。	R/W
DMCRAH	传送计数的高位		R/W

注. 在重复传送模式和块传送模式中，必须给 DMCRAH 寄存器和 DMCRAL 寄存器设定相同的值。

功能因传送模式而不同。

(1) 正常传送模式 (DMACm.DMTMD.MD[1:0] 位 =00b)

DMCRAL 寄存器用作 16 位传送计数器。

当设定值是“0001h”时，传送次数为 1 次；当设定值是“FFFFh”时，传送次数为 65535 次。每进行 1 次数数据传送，DMCRAL 寄存器就减 1。

当设定值是“0000h”时，不指定传送次数，在传送计数器停止计数后进行数据传送（自由运行模式）。

在正常传送模式中不使用 DMCRAH 寄存器，必须给 DMCRAH 寄存器写“0000h”。

(2) 重复传送模式 (DMACm.DMTMD.MD[1:0] 位 =01b)

DMCRAH 寄存器保持重复大小, DMCRAL 寄存器用作 10 位传送计数器。

当设定值是“001h”时, 传送次数为 1 次; 当设定值是“3FFh”时, 传送次数为 1023 次; 当设定是“000h”时, 传送次数为 1024 次。在重复传送模式中, DMCRAH 寄存器和 DMCRAL 寄存器的设定范围都是“000h ~ 3FFh”(1 次~ 1024 次)。

DMCRAL 寄存器的 bit15 ~ bit10 的设定值无效, 必须给 DMCRAL 寄存器的 bit15 ~ bit10 写“0”。

每进行 1 次数据传送, DMCRAL 寄存器就减 1。当 DMCRAL 寄存器变为“000h”时, 传送 DMCRAH 寄存器的值。

(3) 块传送模式 (DMACm.DMTMD.MD[1:0] 位 =10b)

DMCRAH 寄存器保持块大小, DMCRAL 寄存器用作 10 位块大小计数器。

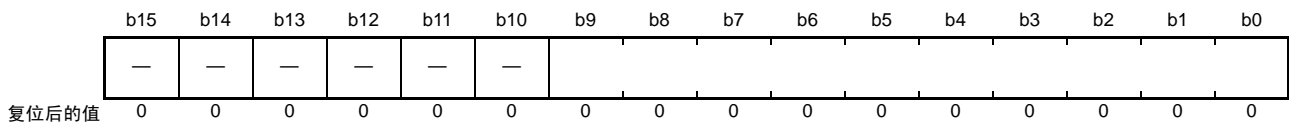
当设定值是“001h”时, 块大小为 1; 当设定值是“3FFh”时, 块大小为 1023; 当设定是“000h”时, 块大小为 1024 次。在块传送模式中, DMCRAH 寄存器和 DMCRAL 寄存器的设定范围都是“000h ~ 3FFh”。

DMCRAL 寄存器的 bit15 ~ bit10 的设定值无效, 必须给 DMCRAL 寄存器的 bit15 ~ bit10 写“0”。

每进行 1 次数据传送, DMCRAL 寄存器就减 1。当 DMCRAL 寄存器变为“000h”时, 传送 DMCRAH 寄存器的值。

17.2.4 DMA 块传送计数寄存器 (DMCRB)

地址 DMAC0.DMCRB 0008 200Ch, DMAC1.DMCRB 0008 204Ch, DMAC2.DMCRB 0008 208Ch, DMAC3.DMCRB 0008 20CCh



位	功能	设定范围	R/W
b9-b0	设定块传送次数和重复传送次数。	001h ~ 3FFh (1 ~ 1023 次) 000h (1024 次)	R/W
b15-b10	保留位	读写值都为“0”。	R/W

DMCRB 寄存器指定块传送模式的块传送次数以及指定重复传送模式的重复传送次数。

当设定值是“001h”时, 传送次数为 1 次; 当设定值是“3FFh”时, 传送次数为 1023 次; 当设定值是“000h”时, 传送次数为 1024 次。

在重复传送模式中传送 1 个重复大小的最后数据时, DMCRB 寄存器减 1。

在块传送模式中传送 1 块大小的最后数据时, DMCRB 寄存器减 1。

在设定为正常传送模式时, 不使用 DMCRB 寄存器, 设定值无效。

17.2.5 DMA 传送模式寄存器 (DMTMD)

地址 DMAC0.DMTMD 0008 2010h、DMAC1.DMTMD 0008 2050h、DMAC2.DMTMD 0008 2090h、DMAC3.DMTMD 0008 20D0h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MD[1:0]		DTS[1:0]		—	—	SZ[1:0]		—	—	—	—	—	—	DCTG[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	DCTG[1:0]	传送请求选择位	b1 b0 0 0: 软件 0 1: 外围模块中断和外部中断输入引脚的中断 (注1) 1 0: 不能设定 1 1: 不能设定	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W
b9-b8	SZ[1:0]	数据传送长度位	b9 b8 0 0: 8 位传送 0 1: 16 位传送 1 0: 32 位传送 1 1: 不能设定	R/W
b11-b10	—	保留位	读写值都为“0”。	R/W
b13-b12	DTS[1:0]	重复区域选择位	b13 b12 0 0: 传送目标为重复区域或者块区域 0 1: 传送源为重复区域或者块区域 1 0: 不设定重复区域和块区域 1 1: 不能设定	R/W
b15-b14	MD[1:0]	传送模式设定位	b15 b14 0 0: 正常传送 0 1: 重复传送 1 0: 块传送 1 1: 不能设定	R/W

注 1. 通过 ICU.DMRSRm 寄存器设定 DMAC 的启动源, 详细内容请参照“14. 中断控制器 (ICU)”的“表 14.3 中断向量表”。

DTS[1:0] 位 (重复区域选择位)

在重复传送模式或者块传送模式中, 这些位能选择传送源或者传送目标为重复区域; 在正常传送模式中, 这些位的设定值无效。

17.2.6 DMA 中断设定寄存器 (DMINT)

地址 DMAC0.DMINT 0008 2013h、DMAC1.DMINT 0008 2053h、DMAC2.DMINT 0008 2093h、DMAC3.DMINT 0008 20D3h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	DARIE	传送目标地址的扩展重复区域上溢中断允许位	0: 禁止传送目标地址的扩展重复区域上溢中断 1: 允许传送目标地址的扩展重复区域上溢中断	R/W
b1	SARIE	传送源地址的扩展重复区域上溢中断允许位	0: 禁止传送源地址的扩展重复区域上溢中断 1: 允许传送源地址的扩展重复区域上溢中断	R/W
b2	RPTIE	重复大小结束中断允许位	0: 禁止重复大小结束中断 1: 允许重复大小结束中断	R/W
b3	ESIE	传送退出结束中断允许位	0: 禁止退出中断 1: 允许退出中断	R/W
b4	DTIE	传送结束中断允许位	0: 禁止传送结束中断 1: 允许传送结束中断	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W

DARIE 位 (传送目标地址的扩展重复区域上溢中断允许位)

如果在将 DARIE 位置“1”时传送目标地址的扩展重复区域发生上溢, 就将 DMCNT.DTE 位清“0”, 同时将 DMSTS.ESIF 标志置“1”, 表示发生了传送目标地址的扩展重复区域上溢中断请求。

当和块传送模式并用时, 在 1 块数据传送结束后产生中断请求。如果通过中断将传送结束通道的 DMACm.DMCNT.DTE 位置“1”, 就能从传送结束状态重新开始传送。

如果没有将扩展重复区域设定为传送目标地址, DARIE 位的设定值就无效。

SARIE 位 (传送源地址的扩展重复区域上溢中断允许位)

如果在将 SARIE 位置“1”时传送源地址的扩展重复区域发生上溢, 就将 DMCNT.DTE 位清“0”, 同时将 DMSTS.ESIF 标志置“1”, 表示发生了传送源地址的扩展重复区域上溢中断请求。

当和块传送模式并用时, 在 1 块数据传送结束后产生中断请求。如果通过中断将传送结束通道的 DMACm.DMCNT.DTE 位置“1”, 就能从传送结束状态重新开始传送。

如果没有将扩展重复区域设定为传送源地址, SARIE 位的设定值就无效。

RPTIE 位 (重复大小结束中断允许位)

在重复传送模式中, 如果将 RPTIE 位置“1”, 就在 1 个重复大小的传送结束后将 DMCNT.DTE 位清“0”, 同时将 DMSTS.ESIF 标志置“1”, 表示发生了重复大小结束中断请求。即使 DMTMD.DTS[1:0] 位为“10b” (不指定重复区域或者块区域), 也能产生重复大小结束中断请求。

在块传送模式中, 如果将 DMINT.RPTIE 位置“1”, 同样也在 1 块数据传送结束后将 DMCNT.DTE 位清“0”, 同时将 DMSTS.ESIF 标志变为“1”, 表示发生了重复大小结束中断请求。即使 DMTMD.DTS[1:0] 位为“10b” (不指定重复区域或者块区域), 也能产生重复大小结束中断请求。

ESIE 位 (传送退出结束中断允许位)

此位允许或者禁止 DMA 传送过程中发生的退出中断请求 (重复大小结束中断和扩展重复区域上溢中断)。

如果将 ESIE 位置 “1”，就在 DMSTS.ESIF 标志被置 “1” 时产生传送退出结束中断。如果将 ESIE 位或者 DMSTS.ESIF 标志清 “0”，就解除传送退出结束中断。

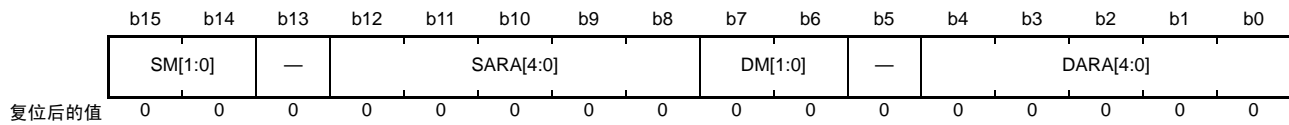
DTIE 位 (传送结束中断允许位)

此位允许或者禁止在指定次数的数据传送结束时发生的传送结束中断请求。

如果将 DTIE 位置 “1”，就在 DMSTS.DTIF 标志被置 “1” 时产生传送结束中断。如果将 DTIE 位或者 DMSTS.DTIF 标志清 “0”，就解除传送结束中断。

17.2.7 DMA 地址模式寄存器 (DMAMD)

地址 DMAC0.DMAMD 0008 2014h、DMAC1.DMAMD 0008 2054h、DMAC2.DMAMD 0008 2094h、DMAC3.DMAMD 0008 20D4h



位	符号	位名	功能	R/W
b4-b0	DARA[4:0]	传送目标地址的扩展重复区域设定位	能将扩展重复区域设定为传送目标地址，设定值的详细内容请参照表 17.2。	R/W
b5	—	保留位	读写值都为“0”。	R/W
b7-b6	DM[1:0]	传送目标地址的更新模式设定位	b7 b6 0 0: 地址固定 0 1: 偏移量增加 (注1) 1 0: 递增 1 1: 递减	R/W
b12-b8	SARA[4:0]	传送源地址的扩展重复区域设定位	能将扩展重复区域设定为传送源地址，设定值的详细内容请参照表 17.2。	R/W
b13	—	保留位	读写值都为“0”。	R/W
b15-b14	SM[1:0]	传送源地址的更新模式设定位	b15 b14 0 0: 地址固定 0 1: 偏移量增加 (注1) 1 0: 递增 1 1: 递减	R/W

注 1. 只有 DMAC0 能设定偏移量增加。

DARA[4:0] 位 (传送目标地址的扩展重复区域设定位)

这些位能将扩展重复区域设定为传送目标地址。通过将指定的低位地址作为地址更新的对象并且总是给剩余的高位取固定值，实现扩展重复区域功能。能设定 2 字节~ 128M 字节的扩展重复区域，设定间隔以 2 的乘方字节为单位。

由地址的增减引起的从扩展重复区域上溢的低位地址会发生以下变化：当地址增加时，为扩展重复区域的起始地址；当地址减小时，为扩展重复区域的结束地址。

如果将重复区域或者块区域设定为传送目标，就不能设定传送目标地址的扩展重复区域。如果在进行重复传送或者块传送时将 DMACm.DMTMD.DTS[1:0] 位置“00b”（传送目标为重复区域或者块区域），就必须给 DARA[4:0] 位写“0000b”。

如果 DMINT.DARIE 位为“1”，就能在扩展重复区域发生上溢时产生中断。扩展重复区域的设定和范围如表 17.2 所示。

DM[1:0] 位 (传送目标地址的更新模式设定位)

这些位设定传送目标地址的更新模式。

如果选择递增，就在 DMTMD.SZ[1:0] 位为“00b”时加 1；在 DMTMD.SZ[1:0] 位为“01b”时加 2；在 DMTMD.SZ[1:0] 位为“10b”时加 4。

如果选择递减，就在 DMTMD.SZ[1:0] 位为“00b”时减 1；在 DMTMD.SZ[1:0] 位为“01b”时减 2；在 DMTMD.SZ[1:0] 位为“10b”时减 4。

如果选择偏移量增加，就加上 DMAC0.DMOFR 寄存器的设定值。只有 DMAC0 能设定偏移量增加。

SARA[4:0] 位 (传送源地址的扩展重复区域设定位)

这些位能将扩展重复区域设定为传送源地址。通过将指定的低位地址作为地址更新的对象并且总是给剩余的高位取固定值, 实现扩展重复区域功能。能设定 2 字节~ 128M 字节的扩展重复区域, 设定间隔以 2 的乘方字节为单位。

由地址的增减引起的从扩展重复区域上溢的低位地址会发生以下变化: 当地址增加时, 为扩展重复区域的起始地址; 当地址减小时, 为扩展重复区域的结束地址。

如果将重复区域或者块区域设定为传送源, 就不能设定传送源地址的扩展重复区域。如果在进行重复传送或者块传送时将 DMACm.DMTMD.DTS[1:0] 位置“01b”(传送源为重复区域或者块区域), 就必须给 SARA[4:0] 位写“00000b”。

如果 DMINT.SARIE 位为“1”, 就能在扩展重复区域发生上溢时产生中断。扩展重复区域的设定和范围如表 17.2 所示。

SM 位 (传送源地址的更新模式设定位)

此位设定传送源地址的更新模式。

如果选择递增, 就在 DMTMD.SZ[1:0] 位为“00b”时加 1; 在 DMTMD.SZ[1:0] 位为“01b”时加 2; 在 DMTMD.SZ[1:0] 位为“10b”时加 4。

如果选择递减, 就在 DMTMD.SZ[1:0] 位为“00b”时减 1; 在 DMTMD.SZ[1:0] 位为“01b”时减 2; 在 DMTMD.SZ[1:0] 位为“10b”时减 4。

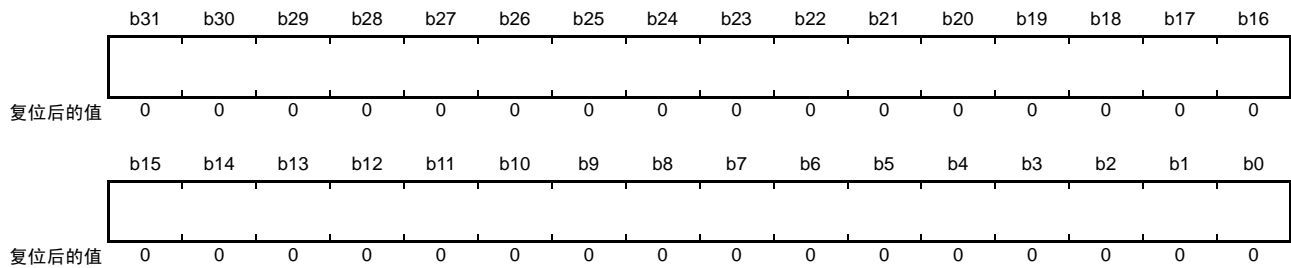
如果选择偏移量增加, 就将 DMAC0.DMOFR 寄存器的设定值相加。只有 DMAC0 能设定偏移量增加。

表 17.2 扩展重复区域的设定和范围

SARA[4:0]/DARA[4:0] 的值	扩展重复区域的范围
00000b	不设定扩展重复区域。
00001b	将该地址的低 1 位 (2 字节) 设定为扩展重复区域。
00010b	将该地址的低 2 位 (4 字节) 设定为扩展重复区域。
00011b	将该地址的低 3 位 (8 字节) 设定为扩展重复区域。
00100b	将该地址的低 4 位 (16 字节) 设定为扩展重复区域。
00101b	将该地址的低 5 位 (32 字节) 设定为扩展重复区域。
00110b	将该地址的低 6 位 (64 字节) 设定为扩展重复区域。
00111b	将该地址的低 7 位 (128 字节) 设定为扩展重复区域。
01000b	将该地址的低 8 位 (256 字节) 设定为扩展重复区域。
01001b	将该地址的低 9 位 (512 字节) 设定为扩展重复区域。
01010b	将该地址的低 10 位 (1K 字节) 设定为扩展重复区域。
01011b	将该地址的低 11 位 (2K 字节) 设定为扩展重复区域。
01100b	将该地址的低 12 位 (4K 字节) 设定为扩展重复区域。
01101b	将该地址的低 13 位 (8K 字节) 设定为扩展重复区域。
01110b	将该地址的低 14 位 (16K 字节) 设定为扩展重复区域。
01111b	将该地址的低 15 位 (32K 字节) 设定为扩展重复区域。
10000b	将该地址的低 16 位 (64K 字节) 设定为扩展重复区域。
10001b	将该地址的低 17 位 (128K 字节) 设定为扩展重复区域。
10010b	将该地址的低 18 位 (256K 字节) 设定为扩展重复区域。
10011b	将该地址的低 19 位 (512K 字节) 设定为扩展重复区域。
10100b	将该地址的低 20 位 (1M 字节) 设定为扩展重复区域。
10101b	将该地址的低 21 位 (2M 字节) 设定为扩展重复区域。
10110b	将该地址的低 22 位 (4M 字节) 设定为扩展重复区域。
10111b	将该地址的低 23 位 (8M 字节) 设定为扩展重复区域。
11000b	将该地址的低 24 位 (16M 字节) 设定为扩展重复区域。
11001b	将该地址的低 25 位 (32M 字节) 设定为扩展重复区域。
11010b	将该地址的低 26 位 (64M 字节) 设定为扩展重复区域。
11011b	将该地址的低 27 位 (128M 字节) 设定为扩展重复区域。
11100b ~ 11111b	不能设定。

17.2.8 DMA 偏移寄存器 (DMOFR)

地址 DMAC0.DMOFR 0008 2018h



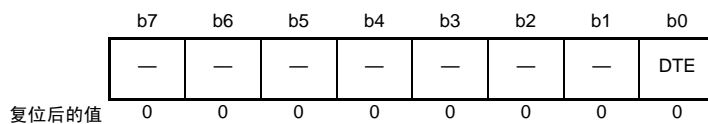
位	功能	设定范围	R/W
b31-b0	设定传送源或者传送目标的地址更新模式为偏移量增加时的偏移值。	00000000h ~ 00FFFFFFh (0 字节~ (16M-1) 字节) FF000000h ~ FFFFFFFFh (-16M 字节~-1 字节)	R/W

在设定 DMOFR 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。

bit31 ~ bit25 的设定值无效，将 bit24 的值扩展到 bit31 ~ bit25。如果读 DMOFR 寄存器，就会读到位扩展后的值。

17.2.9 DMA 传送允许寄存器 (DMCNT)

地址 DMAC0.DMCNT 0008 201Ch、DMAC1.DMCNT 0008 205Ch、DMAC2.DMCNT 0008 209Ch、DMAC3.DMCNT 0008 20DCh



位	符号	位名	功能	R/W
b0	DTE	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

DTE 位 (DMA 传送允许位)

能在 DMAST.DMST 位为“1”（允许启动 DMAC）并且 DTE 位为“1”（允许 DMA 传送）时开始对应通道的 DMA 传送。

[为“1”的条件]

- 当写“1”时

[为“0”的条件]

- 当写“0”时
- 当结束所设定的总传送数据量的传送时
- 当通过重复大小结束中断停止 DMA 传送时
- 当通过扩展重复区域上溢中断停止 DMA 传送时

17.2.10 DMA 软件启动寄存器 (DMREQ)

地址 DMAC0.DMREQ 0008 201Dh、DMAC1.DMREQ 0008 205Dh、DMAC2.DMREQ 0008 209Dh、DMAC3.DMREQ 0008 20DDh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	CLRS	—	—	—	SWREQ
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	SWREQ	DMA 软件启动位	0: 没有 DMA 传送请求 1: 有 DMA 传送请求	R/W
b3-b1	—	保留位	读写值都为“0”。	R/W
b4	CLRS	DMA 软件启动位的自动清除选择位	0: 在软件启动后清除 SWREQ 位 1: 在软件启动后不清除 SWREQ 位	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W

SWREQ 位 (DMA 软件启动位)

如果给 SWREQ 位写“1”，就产生 DMA 的传送请求。一旦开始该请求的传送，就在 CLRS 位为“0”时将 SWREQ 位清“0”，而在 CLRS 位为“1”时不将 SWREQ 位清“0”。此时，能在传送结束后再次产生 DMA 传送请求。

但是，SWREQ 位的值只在 DMTMD.DCTG[1:0] 位被置“00b” (DMA 启动源为软件) 时有效，能通过软件启动 DMA 传送。

如果 DMTMD.DCTG[1:0] 位不是“00b”，SWREQ 位的设定值就无效。

如果在 CLRS 位为“0”时要进行软件启动，就必须在确认 SWREQ 位是“0”后给 SWREQ 位写“1”。

[为“1”的条件]

- 当写“1”时

[为“0”的条件]

- 在 CLRS 位为“0” (在软件启动后清除 SWREQ 位) 的情况下接受软件请求并且开始数据传送时
- 当写“0”时

CLRS 位 (DMA 软件启动位的自动清除选择位)

当通过给 SWREQ 位写“1”开始 DMA 传送请求的传送时，此位设定是否将 SWREQ 位清“0”。当 CLRS 位为“0”时，一旦开始传送，就将 SWREQ 位清“0”；当 CLRS 位为“1”时，不将 SWREQ 位清“0”。此时，能在传送结束后再次产生 DMA 传送请求。

17.2.11 DMA 状态寄存器 (DMSTS)

地址 DMAC0.DMSTS 0008 201Eh、DMAC1.DMSTS 0008 205Eh、DMAC2.DMSTS 0008 209Eh、DMAC3.DMSTS 0008 20DEh

b7	b6	b5	b4	b3	b2	b1	b0
ACT	—	—	DTIF	—	—	—	ESIF
复位后的值	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	ESIF	传送退出中断标志	0: 未发生传送退出中断 1: 发生传送退出中断	R/W (注1)
b3-b1	—	保留位	读取值为“0”，写操作无效。	R
b4	DTIF	传送结束中断标志	0: 没有传送结束中断 1: 有传送结束中断	R/W (注1)
b6-b5	—	保留位	读取值为“0”，写操作无效。	R
b7	ACT	DMA 有效标志	0: DMAC 停止 1: DMAC 正在运行	R

注 1. 只能写“0”。

ESIF 标志 (传送退出中断标志)

这是表示发生了传送退出中断的标志。

[为“1”的条件]

- DMINT.RPTIE 位被置“1”并且在重复传送模式中 1 个重复大小的传送结束后
- DMINT.RPTIE 位被置“1”并且在块传送模式中 1 块数据传送结束后
- 在 DMINT.SARIE 位被置“1”而 DMAMD.SARA[4:0] 位被置的值不是“00000b” (将传送源地址指定为扩展重复区域) 的情况下传送源地址的扩展重复区域发生上溢时
- 在 DMINT.DARIE 位被置“1”而 DMAMD.DARA[4:0] 位被置的值不是“00000b” (将传送目标地址指定为扩展重复区域) 的情况下传送目标地址的扩展重复区域发生上溢时

[为“0”的条件]

- 当写“0”时
- 当给 DMCNT.DTE 位写“1”时

DTIF 标志 (传送结束中断标志)

这是表示发生了传送结束中断的标志。

[为“1”的条件]

- 在正常传送模式中指定次数的传送结束时 (DMCRAL 寄存器变为“0”并且传送结束时)
- 在重复传送模式中指定重复次数的传送结束时 (DMCRB 寄存器变为“0”并且传送结束时)
- 在块传送模式中指定块数的传送结束时 (DMCRB 寄存器变为“0”并且传送结束时)

[为“0”的条件]

- 当写“0”时
- 当给 DMCNT.DTE 位写“1”时

ACT 标志 (DMA 有效标志)

这是表示 DMAC 正在运行或者停止的标志。

[为“1”的条件]

- 当 DMAC 开始传送时

[为“0”的条件]

- 当 1 次传送请求的传送全部结束时

17.2.12 DMA 启动源标志控制寄存器 (DMCSL)

地址 DMAC0.DMCSL 0008 201Fh、DMAC1.DMCSL 0008 205Fh、DMAC2.DMCSL 0008 209Fh、DMAC3.DMCSL 0008 20DFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DISEL
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	DISEL	中断选择位	0: 在传送结束时, 将启动源的中断标志清“0”。 1: 在传送结束时, 通过启动源的中断标志向 CPU 请求中断。	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

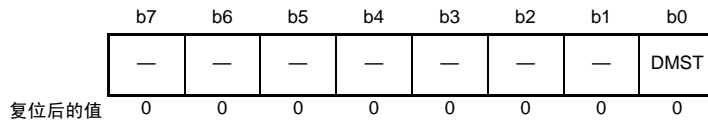
DISEL 位 (中断选择位)

在 DMAC 传送开始时, 此位选择是将启动源的中断标志清“0”还是通过中断标志向 CPU 请求中断。

在将 DMTMD.DCTG[1:0] 位置“00b” (通过软件启动) 时, DISEL 位的设定值无效。

17.2.13 DMA 模块启动寄存器 (DMAST)

地址 0008 2200h



位	符号	位名	功能	R/W
b0	DMST	DMAC 运行允许位	0: 禁止启动 DMAC 1: 允许启动 DMAC	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

DMST 位 (DMAC 运行允许位)

当 DMST 位为“1”时，允许启动 DMAC 的全部通道。

如果在给多个通道的 DMACm.DMCNT.DTE 位写“1”（允许 DMA 传送）后，将 DMST 位置“1”（允许启动 DMAC），就能将多个通道置为可同时接受传送请求的状态。

如果在 DMAC 运行过程中将 DMST 位置“0”，就在正在执行的 1 个传送请求的数据传送结束后暂停 DMA 运行。在此状态下，能通过再次将 DMST 位置“1”继续进行 DMA 传送。

[为“1”的条件]

- 当写“1”时

[为“0”的条件]

- 当写“0”时

17.3 运行说明

17.3.1 传送模式

(1) 正常传送模式

在正常传送模式中，1 次传送请求进行 1 个数据的传送。能通过设定 DMACm.DMCRAL 寄存器，设定最多 65535 次的数据传送次数。如果将 DMACm.DMCRAL 寄存器设定为“0000h”，就不需要指定传送次数，并且在传送计数器停止计数后进行数据传送（自由运行模式）。在正常传送模式中，DMACm.DMCRB 寄存器的设定无效。除自由运行模式以外，能在指定传送次数的传送结束后产生传送结束中断请求。

正常传送模式的寄存器的更新值以及正常传送模式的运行分别如表 17.3 和图 17.2 所示。

表 17.3 正常传送模式的寄存器的更新值

寄存器	功能	1 次传送请求的传送结束后的更新值
DMACm.DMSAR	传送源地址	递增 / 递减 / 固定 / 偏移量增加 (注 1)
DMACm.DMDAR	传送目标地址	递增 / 递减 / 固定 / 偏移量增加 (注 1)
DMACm.DMCRAL	传送计数	减 1 / 不更新 (自由运行模式)
DMACm.DMCRAH	—	不更新 (不用于正常传送模式)
DMACm.DMCRB	—	不更新 (不用于正常传送模式)

注 1. 只有 DMAC0 能指定偏移量增加。

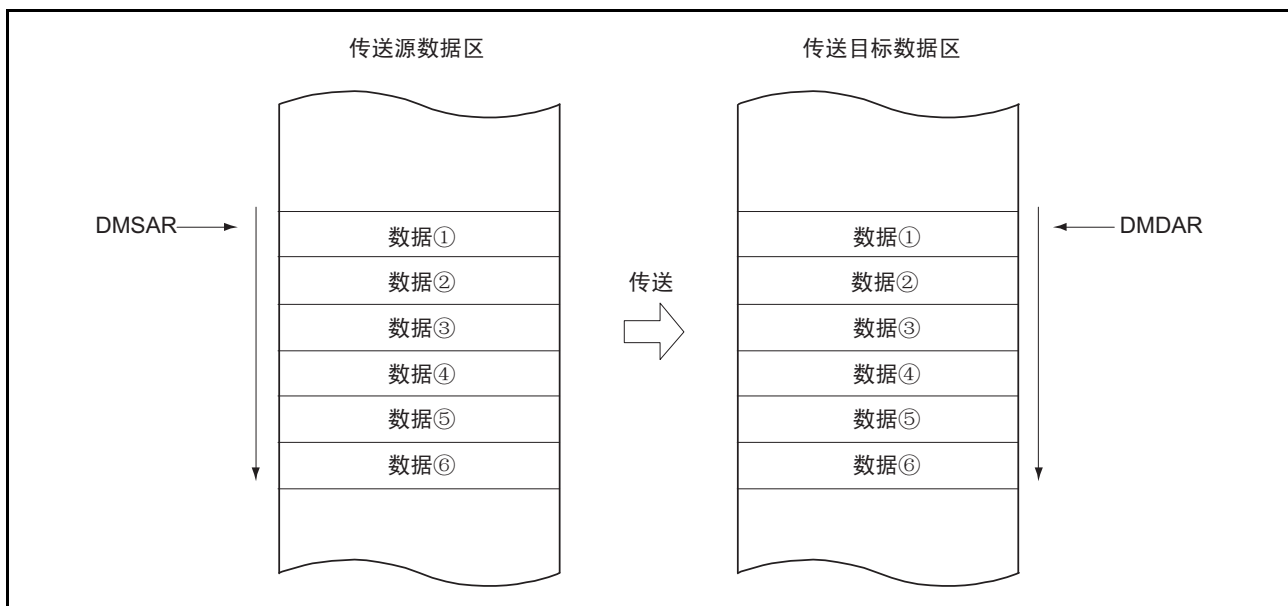


图 17.2 正常传送模式的运行

(2) 重复传送模式

在重复传送模式中，1 次传送请求进行 1 个数据的传送。

能通过 DMACm.DMCRA 寄存器设定最多 1K 数据的重复大小。

还能通过 DMACm.DMCRB 寄存器设定最多 1K 次的指定重复次数，指定最多 1K 数据 × 1K 重复次数 = 1M 数据的总数据传送量。

能将传送源或者传送目标指定为重复区域。一旦重复大小的数据传送结束，被指定为重复区域的地址寄存器 (DMACm.DMSAR 或者 DMACm.DMDAR) 就返回到初始地址。在重复传送模式中，能在重复大小的数据传送结束后停止 DMA 传送并且产生重复大小结束中断请求。如果通过重复大小结束中断处理给 DMACm.DMCNT.DTE 位写“1”，就能重新开始 DMA 传送。

能在指定重复次数的传送结束后产生传送结束中断请求。

重复传送模式的寄存器的更新值以及重复传送模式的运行分别如表 17.4 和图 17.3 所示。

表 17.4 重复传送模式的寄存器的更新值

寄存器	功能	1 次传送请求的传送结束后的更新值	
		DMACm.DMCRAL 寄存器不为“1”	DMACm.DMCRAL 寄存器为“1” (重复大小的最后数据的传送)
DMACm.DMSAR	传送源地址	递增 / 递减 / 固定 / 偏移量增加 (注 1)	<ul style="list-style-type: none"> DMACm.DMTMD.DTS[1:0]=00b 递增 / 递减 / 固定 / 偏移量增加 (注 1) DMACm.DMTMD.DTS[1:0]=01b DMACm.DMSAR 的初始值 DMACm.DMTMD.DTS[1:0]=10b 递增 / 递减 / 固定 / 偏移量增加 (注 1)
DMACm.DMDAR	传送目标地址	递增 / 递减 / 固定 / 偏移量增加 (注 1)	<ul style="list-style-type: none"> DMACm.DMTMD.DTS[1:0]=00b DMACm.DMDAR 的初始值 DMACm.DMTMD.DTS[1:0]=01b 递增 / 递减 / 固定 / 偏移量增加 (注 1) DMACm.DMTMD.DTS[1:0]=10b 递增 / 递减 / 固定 / 偏移量增加 (注 1)
DMACm.DMCRAH	重复大小	保持	保持
DMACm.DMCRAL	传送计数	减 1	DMACm.DMCRAH
DMACm.DMCRB	重复次数的计数	保持	减 1

注 1. 只有 DMAC0 能指定偏移量增加。

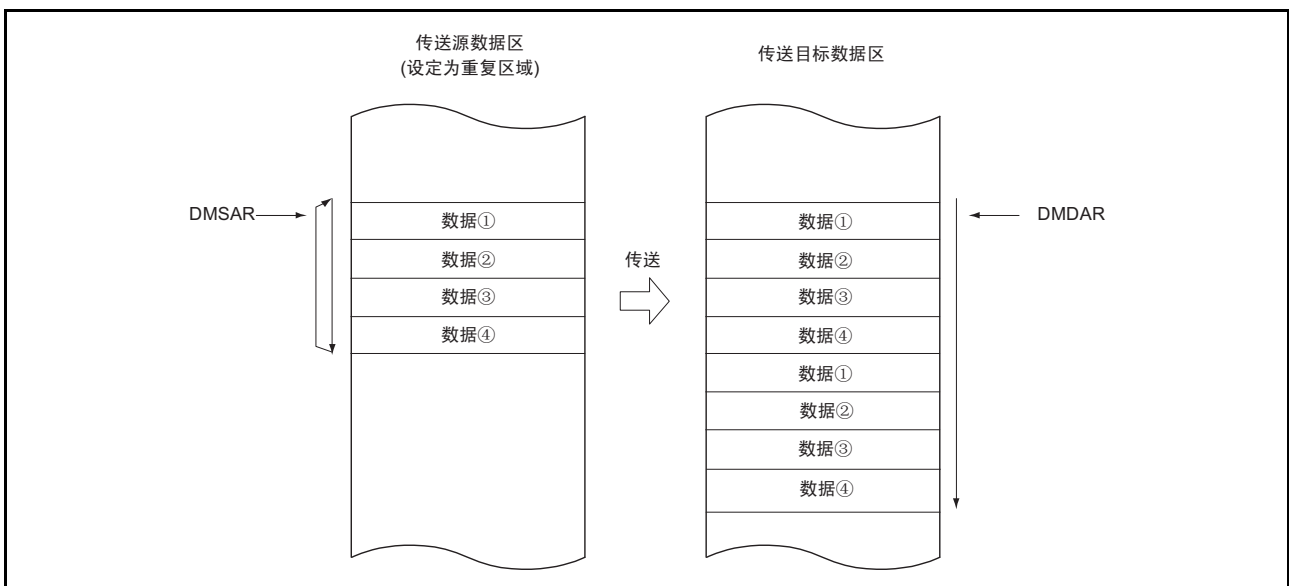


图 17.3 重复传送模式的运行

(3) 块传送模式

在块传送模式中，1 次传送请求进行 1 块数据的传送。

能通过 DMACm.DMCRA 寄存器设定最多 1K 个数据的块大小。

还能通过 DMACm.DMCRB 寄存器设定最多 1K 次的指定块数，指定最多 1K 数据 × 1K 块 = 1M 数据的总数据传送量。能将传送源或者传送目标指定为块区域。一旦 1 块数据传送结束，被指定为块区域的地址寄存器 (DMACm.DMSAR 或者 DMACm.DMDAR) 就返回到初始地址。在块传送模式中，能在 1 块数据传送结束后停止 DMA 传送并且产生重复大小结束中断请求。如果通过重复大小结束中断处理给 DMACm.DMCNT.DTE 位写“1”，就能重新开始 DMA 传送。

能在指定块数的传送结束后产生传送结束中断请求。

块传送模式的寄存器的更新值以及块传送模式的运行分别如表 17.5 和图 17.4 所示。

表 17.5 块传送模式的寄存器的更新值

寄存器	功能	1 次传送请求的 1 块传送结束后的更新值
DMACm.DMSAR	传送源地址	<ul style="list-style-type: none"> DMACm.DMTMD.DTS[1:0]=00b 递增 / 递减 / 固定 / 偏移量增加 (注 1) DMACm.DMTMD.DTS[1:0]=01b DMACm.DMSAR 的初始值 DMACm.DMTMD.DTS[1:0]=10b 递增 / 递减 / 固定 / 偏移量增加 (注 1)
DMACm.DMDAR	传送目标地址	<ul style="list-style-type: none"> DMACm.DMTMD.DTS[1:0]=00b DMACm.DMDAR 的初始值 DMACm.DMTMD.DTS[1:0]=01b 递增 / 递减 / 固定 / 偏移量增加 (注 1) DMACm.DMTMD.DTS[1:0]=10b 递增 / 递减 / 固定 / 偏移量增加 (注 1)
DMACm.DMCRAH	块大小	保持
DMACm.DMCRAL	传送计数	DMACm.DMCRAH
DMACm.DMCRB	块计数	减 1

注 1. 只有 DMAC0 能指定偏移量增加。

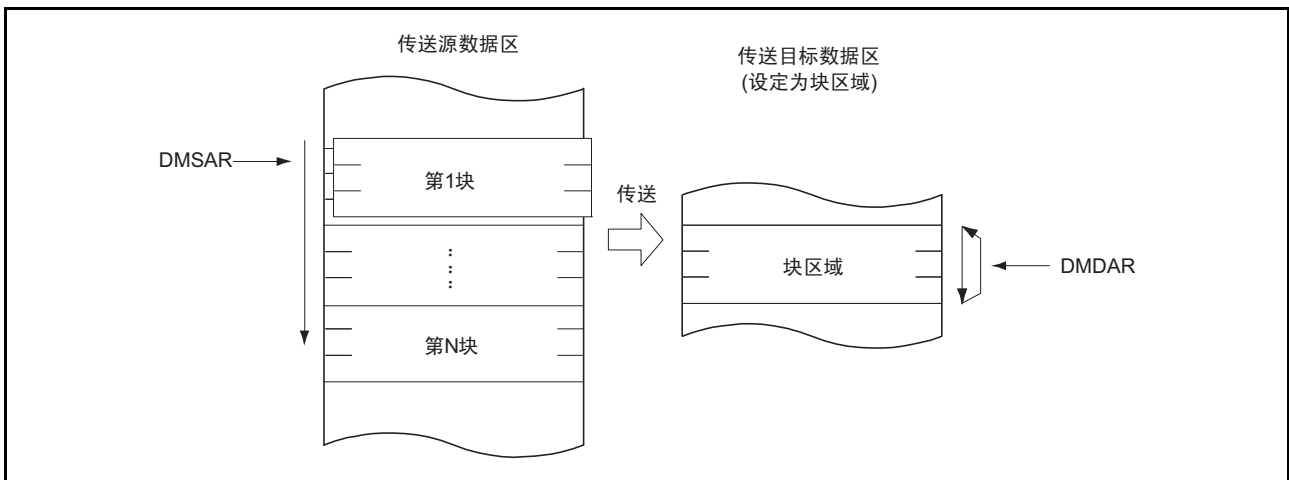


图 17.4 块传送模式的运行

17.3.2 扩展重复区域功能

DMAC 具有给传送源地址或者传送目标地址设定扩展重复区域的功能。如果设定扩展重复区域，地址寄存器就重复被指定为扩展重复区域范围的地址值。

能给 DMACm.DMSAR 寄存器（传送源地址寄存器）和 DMACm.DMDAR 寄存器（传送目标地址寄存器）独立设定扩展重复区域功能。

通过 DMACm.DMAMD.SARA[4:0] 位设定传送源地址的扩展重复区域，通过 DMACm.DMAMD.DARA[4:0] 位设定传送目标地址的扩展重复区域。能独立设定各扩展重复区域的大小，但是不能将被指定为重复区域或者块区域（传送源或者传送目标）的区域指定为扩展重复区域。

如果地址寄存器的值为扩展重复区域的结束地址并且扩展重复区域发生上溢，就暂停 DMA 传送，并且能产生扩展重复区域上溢中断请求。如果将 DMACm.DMINT.SARIE 位置“1”，就在传送源地址的扩展重复区域发生上溢时 DMACm.DMSTS.ESIF 标志变为“1”并且通过将 DMACm.DMCNT.DTE 位置“0”来结束 DMA 传送。此时，如果 DMACm.DMINT.ESIE 位为“1”，就产生扩展重复区域上溢中断请求。

如果将 DMACm.DMINT.DARIE 位置“1”，就以传送目标地址寄存器为对象。

如果通过扩展重复区域上溢中断处理将 DMACm.DMCNT.DTE 位置“1”，就能重新开始 DMA 传送。

扩展重复区域功能的例子如图 17.5 所示。

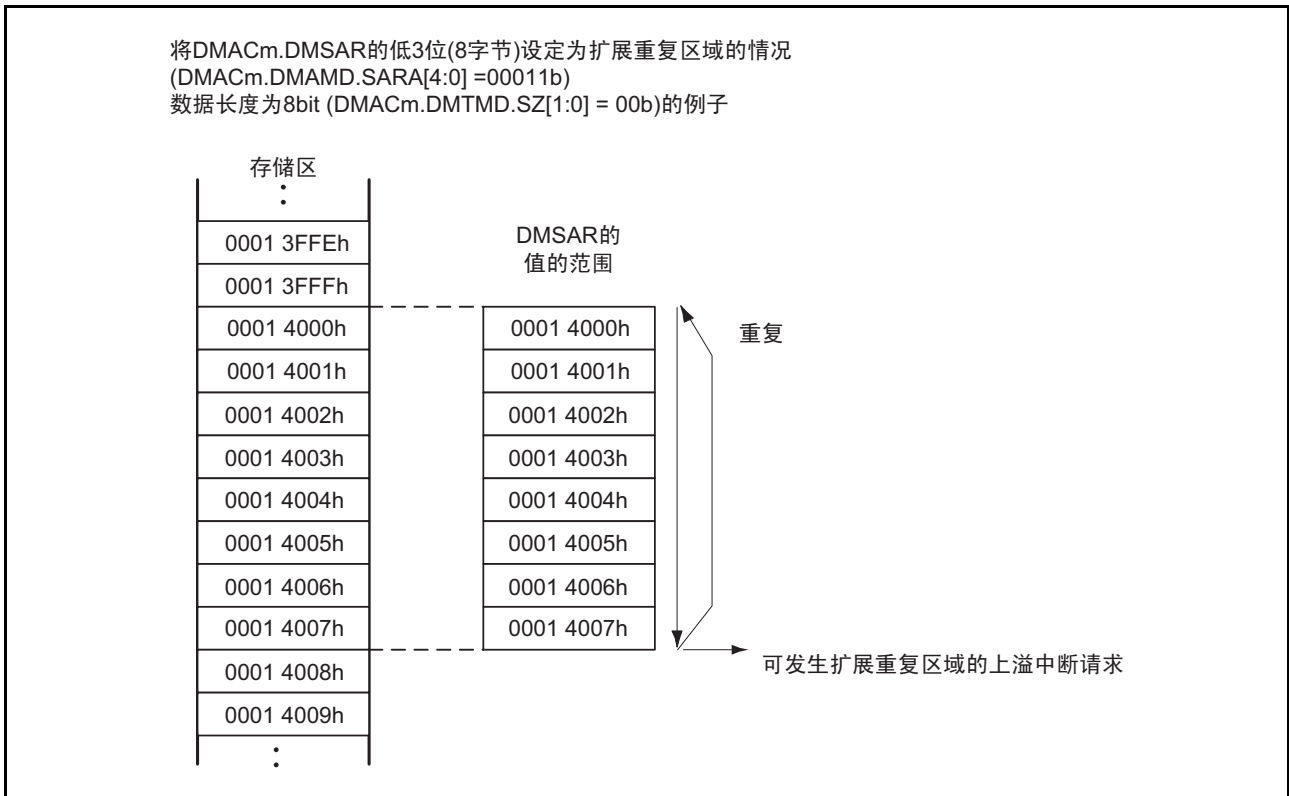


图 17.5 扩展重复区域功能的例子

在块传送模式和扩展重复区域上溢中断并用时，需要注意以下事项。

如果要在扩展重复区域发生上溢时结束传送，就需要将块大小设定为 2 的乘方，或者设定地址寄存器的值，使块大小的边界和扩展重复区域范围的边界相同。如果在传送 1 块数据的过程中扩展重复区域发生上溢，就将扩展重复区域上溢中断请求保留到 1 块数据传送结束，传送发生溢出。

块传送模式和扩展重复区域功能的并用例子如图 17.6 所示。

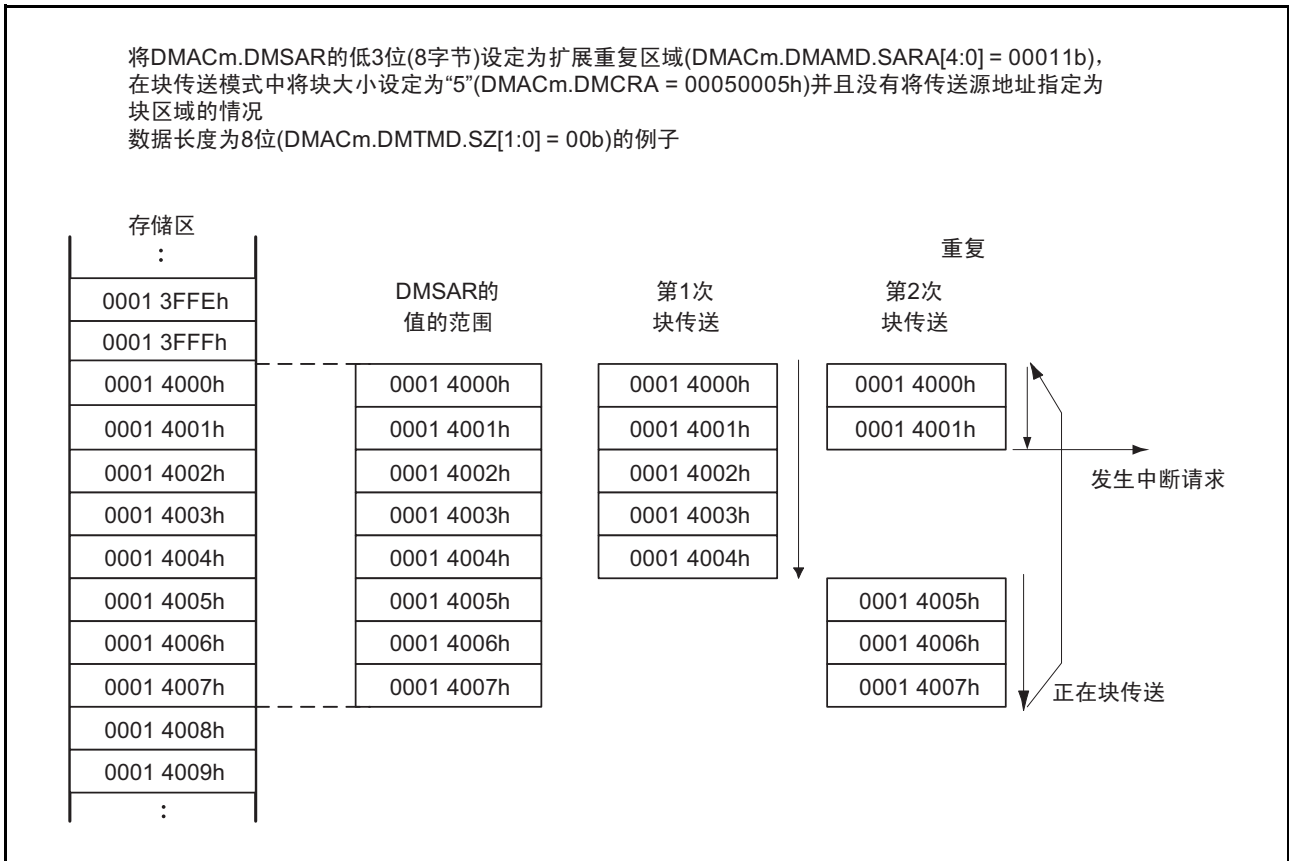


图 17.6 块传送模式和扩展重复区域功能的并用例子

17.3.3 带偏移的地址更新功能

传送源地址和传送目标地址的更新方法有固定、递增、递减和偏移量增加。偏移量增加是指在每进行 1 个数据的传送时加上 DMA 偏移寄存器 (DMAC0.DMOFR) 的设定值。通过此功能,能跳过中途地址进行数据传送。

如果用 2 的补数给 DMAC0.DMOFR 设定负值,也能通过偏移实现减法运算。

只有通道 DMAC0 才能使用带偏移的地址更新功能。

各地址更新模式的地址更新方法如表 17.6 所示。

表 17.6 各地址更新模式的地址更新方法

地址更新模式	DMACm.DMAMD.SM[1:0] DMACm.DMAMD.DM[1:0] 地址更新模式的设定值	更新方法 (各 DMACm.DMTMD.SZ[1:0] 设定值的更新方法)		
		SZ[1:0]=00b	SZ[1:0]=01b	SZ[1:0]=10b
地址固定	00b	固定		
偏移量增加	01b	+DMACm.DMOFR (注 1)		
递增	10b	+1	+2	+4
递减	11b	-1	-2	-4

注 1. 必须用 2 的补数给偏移寄存器设定负值。用以下表达式求 2 的补数:
负偏移值的 2 的补数表示 = \sim (偏移值) + 1 (\sim : 位取反)

(1) 使用偏移量增加的基本传送

偏移的地址更新功能的运行例子如图 17.7 所示。

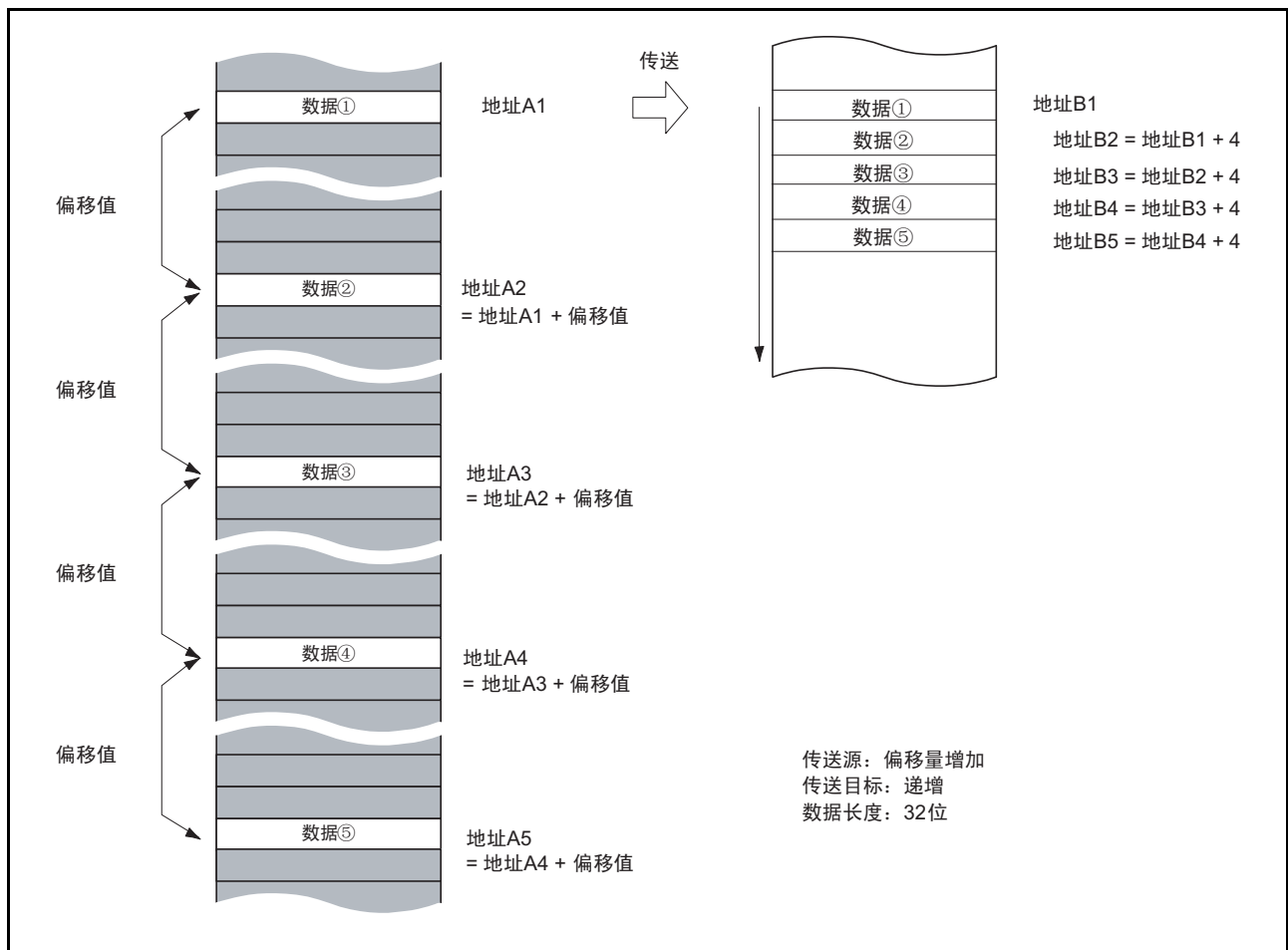


图 17.7 偏移的地址更新功能的运行例子

在图 17.7 中，将传送数据长度设定为“32 位”，传送源地址的更新模式设定为“偏移量增加”，传送目标地址的更新模式设定为“递增”。在进行第 2 次以后的传送源地址更新时，从上次传送时的地址加上偏移值后的地址读数据，然后将一定间隔的读数据写到传送目标的连续区域。

(2) 使用偏移量增加的 XY 转换例子

通过重复传送模式和偏移量增加的组合进行 XY 转换时的运行例子如图 17.8 所示。
 设定方法如下：

- DMAC0.DMAMD 寄存器：传送源地址更新模式设定：偏移量增加
- DMAC0.DMAMD 寄存器：传送目标地址更新模式设定：递增
- DMAC0.DMTMD 寄存器：传送数据大小位：32位传送
- DMAC0.DMTMD 寄存器：传送模式设定位：重复传送
- DMAC0.DMTMD 寄存器：重复区域选择位：传送源侧为重复区域
- DMAC0.DMOFR 寄存器：偏移地址：10h
- DMAC0.DMCRA 寄存器：重复大小：4h
- DMAC0.DMINT 寄存器：设定为重复大小结束中断

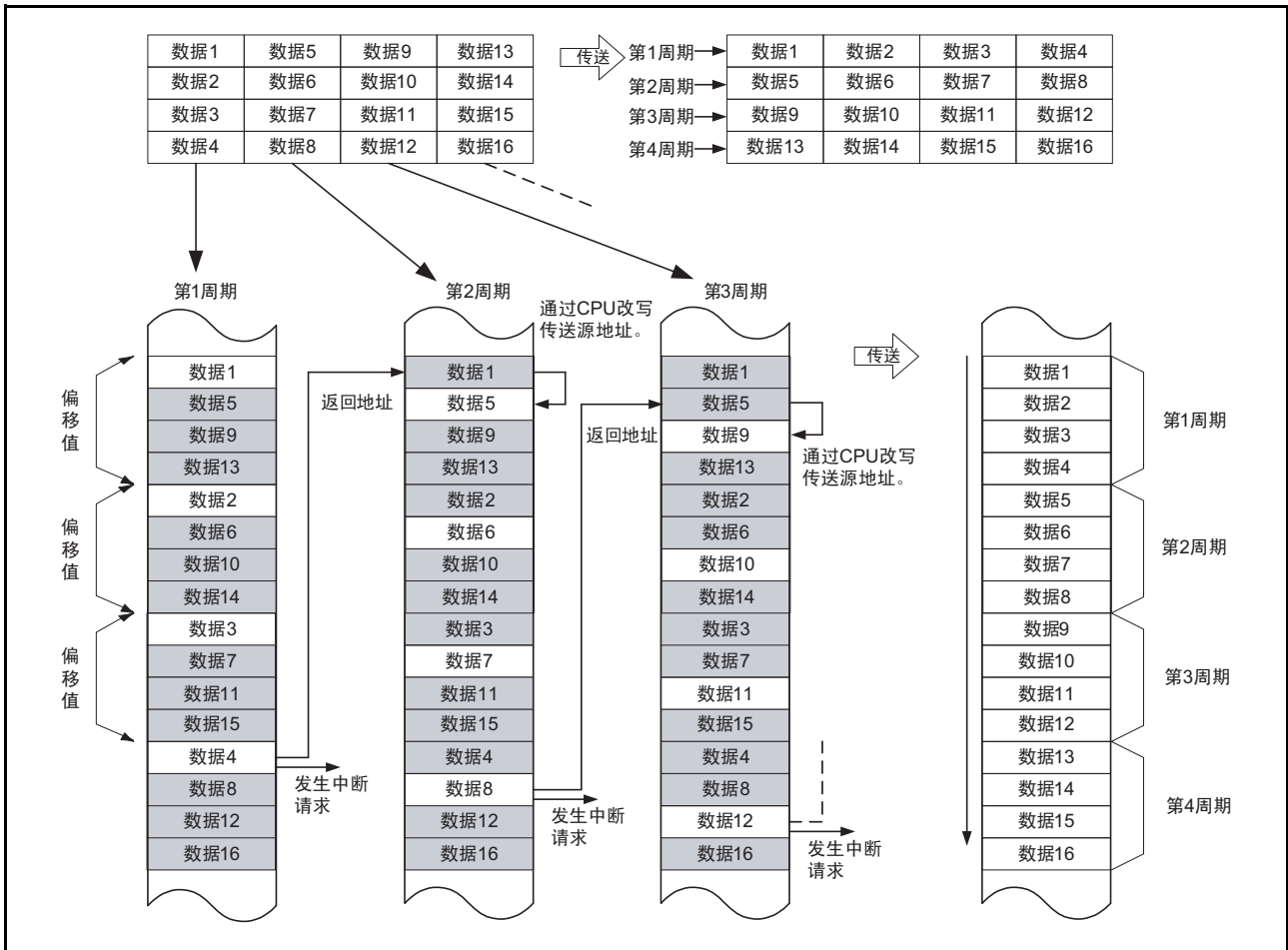


图 17.8 使用重复传送模式和偏移量增加进行 XY 转换时的运行

一旦传送开始，传送源就将地址加上偏移值进行数据传送，传送数据按照传送顺序连续排列在传送目标中。如果传送“数据4”为止的数据，就结束重复大小数据的传送，DMAC 将传送源地址返回到开始传送时的地址（传送源“数据1”的地址），同时产生重复大小结束中断请求。通过此中断请求一旦中止传送。必须通过中断进行以下处理：

- DMAC0.DMSAR 寄存器：将 DMA 传送源地址改写为“数据5”的地址。
 （在上述例子中，改写为“数据1”的地址+4的值）。
- DMAC0.DMCNT 寄存器：给 DTE 位写“1”。

从中止 DMA 传送的状态继续开始 DMA 传送。此后，如果重复同样的处理，就能在将传送源数据进行 XY 转换后传送到传送目标。

XY 转换的处理流程如图 17.9 所示。

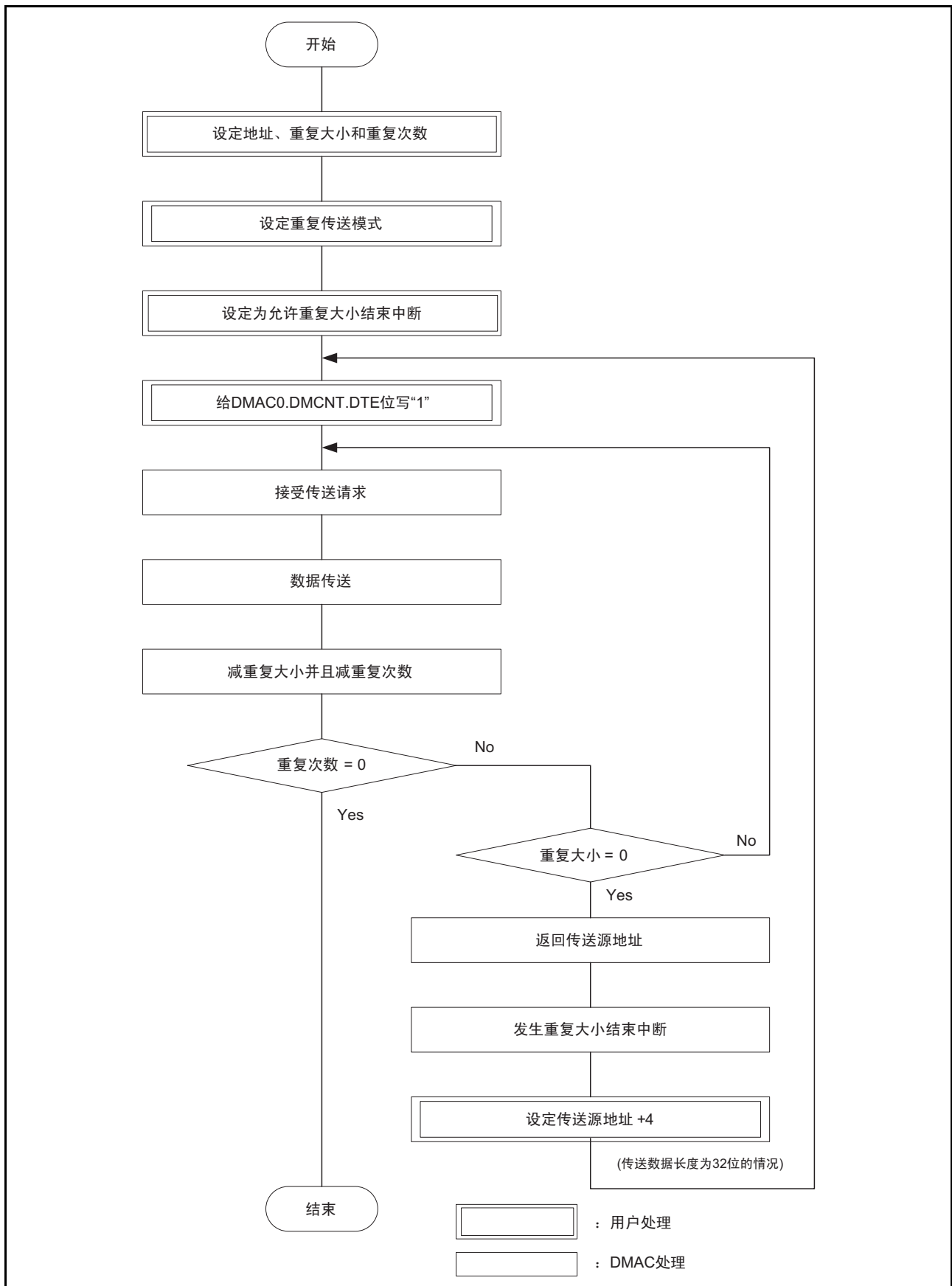


图 17.9 使用重复传送模式和偏移量增加进行 XY 转换的流程

17.3.4 启动源

DMAC 的启动源有软件、外围模块中断请求和外部中断请求，能通过 DMACm.DMTMD.DCTG[1:0] 位选择这些启动源。

(1) 通过软件进行的启动

如果将 DMACm.DMTMD.DCTG[1:0] 位置“00b”，就能通过软件进行启动。

要通过软件启动 DMA 传送时，必须在将 DMACm.DMTMD.DCTG[1:0] 位置“00b”后给 DMACm.DMCNT.DTE 位写“1”（允许 DMA 传送），并且将 DMAST.DMST 位置“1”（允许启动 DMAC）。此后，如果给 DMACm.DMREQ.SWREQ 位写“1”（有 DMA 传送请求），就开始 DMA 运行。

如果在 DMACm.DMREQ.CLR 位为“0”的状态下进行软件启动，就在开始 DMA 传送请求的传送时 DMACm.DMREQ.SWREQ 位变为“0”；如果在 DMACm.DMREQ.CLR 位为“1”的状态下进行软件启动，即使开始传送，DMACm.DMREQ.SWREQ 位也不变为“0”。在该请求的传送结束后再次发生 DMA 传送请求。

(2) 通过外围模块中断请求或者外部中断请求进行的启动

能将外围模块中断请求或者外部中断请求指定为 DMA 传送启动源。通过中断控制器 (ICU) 的 ICU.DMRSRm 寄存器 (m=0 ~ 3) 选择启动源，各通道能独立设定。

当通过外围模块中断请求或者外部中断请求启动 DMA 时，必须在将 DMACm.DMTMD.DCTG[1:0] 位置“01b”（外围模块中断和外部中断引脚的中断）后将 DMACm.DMCNT.DTE 位置“1”（允许 DMA 传送），并且将 DMAST.DMST 位置“1”（允许启动 DMAC）。此后，如果发生中断请求，就开始 DMA 运行。

DMAC 启动源的中断源一览表，请参照“14. 中断控制器 (ICUb)”的“表 14.3 中断向量表”。

17.3.5 运行时序

DMAC 的运行时序例子如图 17.10 和图 17.11 所示。

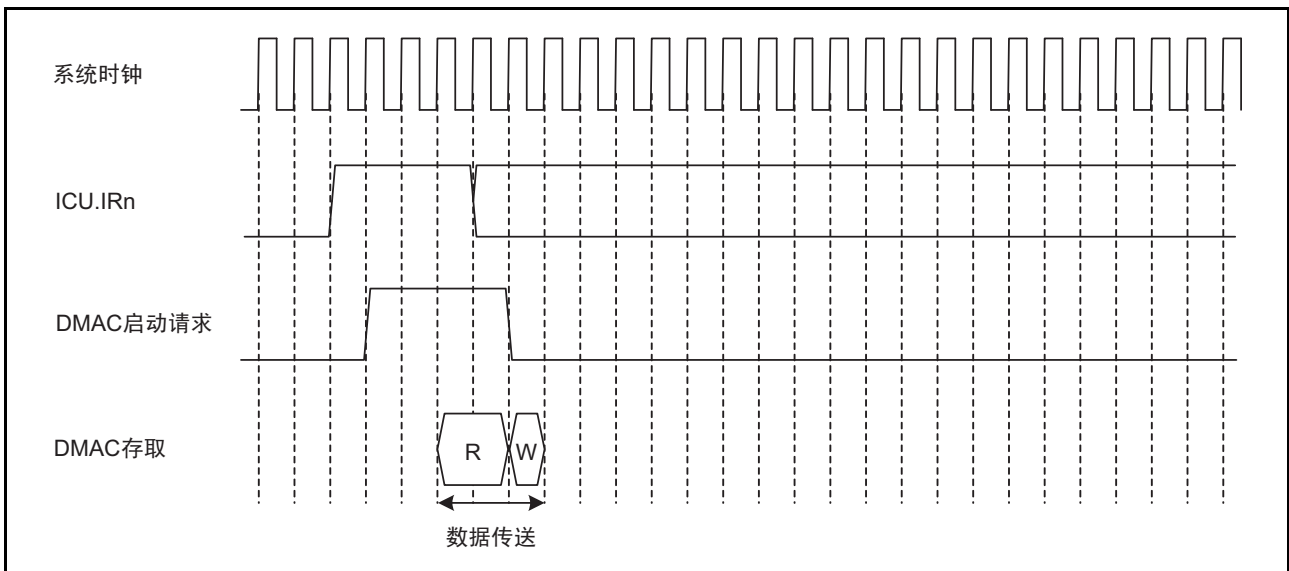


图 17.10 DMAC 的运行时序例子 (1) (在正常传送模式和重复传送模式中，通过外围模块或者外部中断输入引脚的中断启动 DMA 的情况)

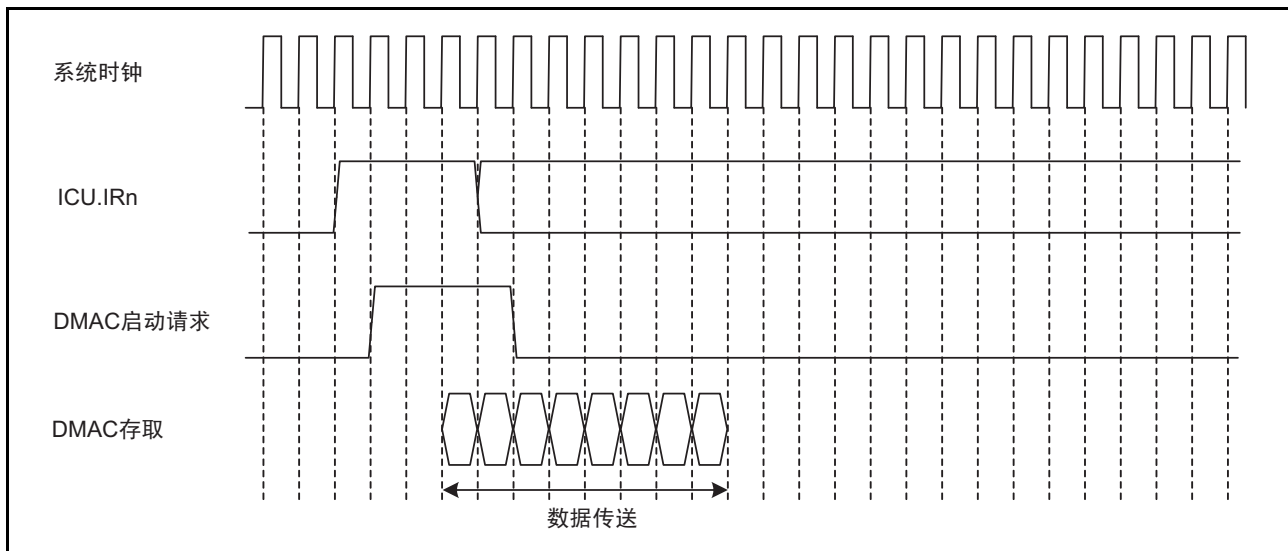


图 17.11 DMAC 的运行时序例子 (2) (在块传送模式中, 块大小为“4”, 通过外围模块或者外部中断输入引脚的中断启动 DMA 的情况)

17.3.6 DMAC 的执行周期

DMAC 的 1 次数据传送的执行状态如表 17.7 所示。

表 17.7 DMAC 的执行周期

传送模式	数据传送 (读)	数据传送 (写)
正常	$Cr+1$	Cw
重复	$Cr+1$	Cw
块 (注 1)	$P \times Cr$	$P \times Cw$

注 1. 这是块大小大于等于“2”的情况。当块大小为“1”时, 为正常传送的周期数。

P: 块大小 (DMCRAH 寄存器的设定值)

Cr: 数据的读目标存取周期

Cw: 数据的写目标存取周期

Cr 和 Cw 因存取目标而不同。各存取目标的周期数请参照“33. RAM”、“34. ROM (保存代码的闪存)”、“5. I/O 寄存器”。

数据传送 (读) 的“+1”的单位是系统时钟 (ICLK)。

运行例子请参照“17.3.5 运行时序”

17.3.7 DMAC 的启动

寄存器的设定步骤如图 17.12 所示。

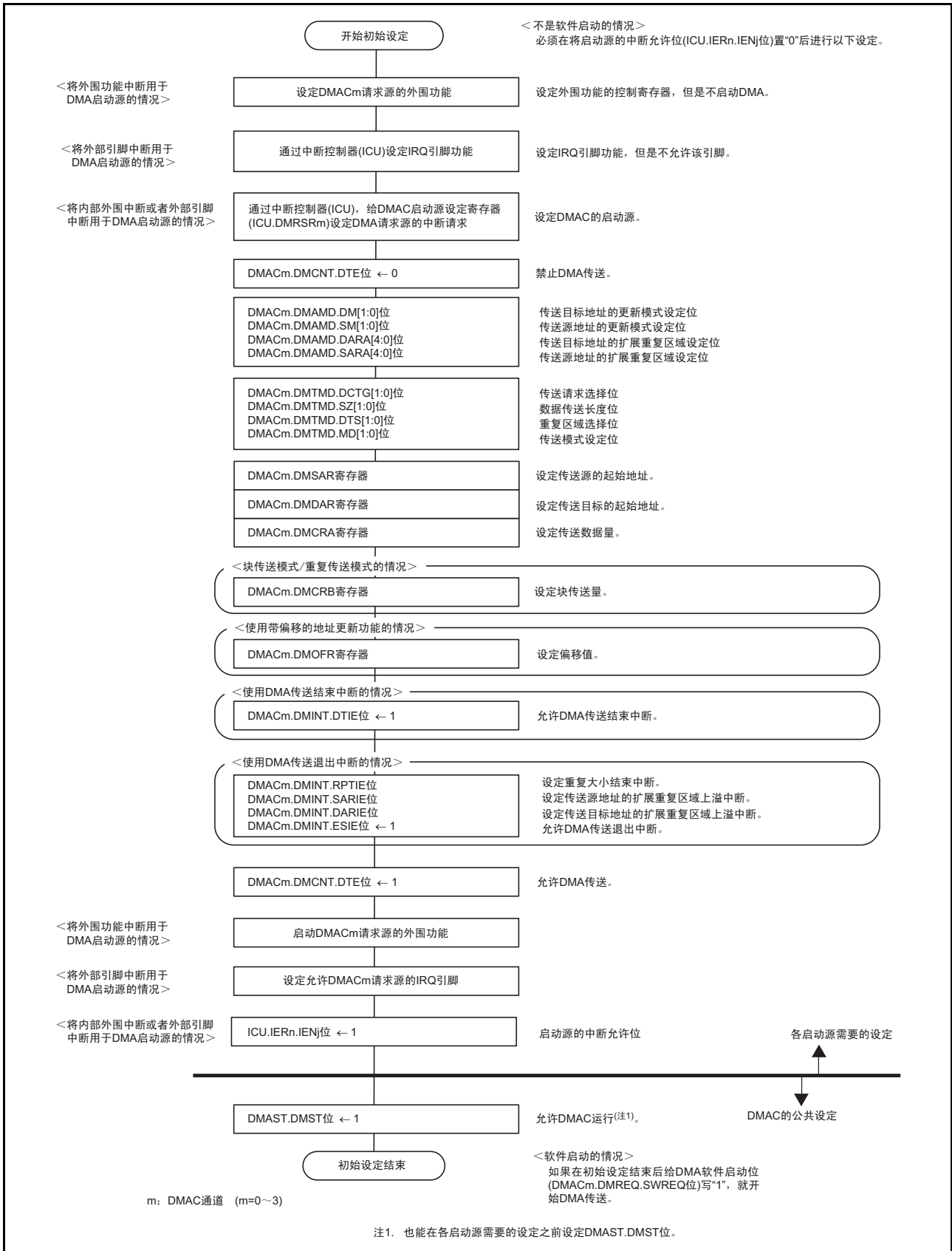


图 17.12 寄存器的设定步骤

17.3.8 DMA 传送的开始

如果在将 DMACm.DMCNT.DTE 位置“1”（允许 DMA 传送）后将 DMAST.DMST 位置“1”（允许启动 DMAC），就能进行通道 m（m=0～3）的 DMA 传送。

在 DTC 传送过程中，不接受其他 DMAC 通道的新启动请求。在先开始的传送结束时接受优先级最高的通道的 DMA 传送请求，开始 DMA 传送。一旦开始 DMA 传送，DMACm.DMSTS.ACT 位就变为“1”（DMAC 正在运行）。

17.3.9 DMA 传送过程中的寄存器

通过 DMA 传送处理更新 DMAC 寄存器的值，被更新的值因各种设定和传送状态而不同，被更新的寄存器为 DMACm.DMSAR、DMACm.DMDAR、DMACm.DMCRA、DMACm.DMCRB、DMACm.DMCNT 和 DMACm.DMSTS。

(1) DMA 传送源地址寄存器 (DMACm.DMSAR)

如果进行 1 次传送请求的数据传送，就通过下次请求将 DMSAR 寄存器更新为要存取的地址。各模式的寄存器更新请参照表 17.3～表 17.5。

(2) DMA 传送目标地址寄存器 (DMACm.DMDAR)

如果进行 1 次传送请求的数据传送，就在下次请求时将 DMDAR 寄存器更新为要存取的地址。各模式的寄存器更新请参照表 17.3～表 17.5。

(3) DMA 传送计数寄存器 (DMACm.DMCRA)

如果进行 1 次传送请求的数据传送，就更新计数值。各模式的更新值不同，各模式的寄存器更新请参照表 17.3～表 17.5。

(4) DMA 块传送计数寄存器 (DMACm.DMCRB)

如果进行 1 次传送请求的数据传送，就更新计数值。各模式的更新值不同，各模式的寄存器更新请参照表 17.3～表 17.5。

(5) DMA 传送允许位 (DMACm.DMCNT.DTE)

DMACm.DMCNT.DTE 位通过寄存器的写操作控制允许或者禁止数据传送。根据 DMA 传送状态，如果以下任意一个条件成立，DMACm.DMCNT.DTE 位就自动变为“0”。

- 设定的总传送数据量的传送结束时
- 发生重复大小结束中断请求并且传送结束时
- 发生扩展重复区域上溢中断请求并且传送结束时

禁止写 DMACm.DMCNT.DTE 位为“1”的通道的寄存器（DMACm.DMCNT 寄存器除外），必须在将 DMACm.DMCNT.DTE 位置“0”的状态下更改各寄存器的设定。

(6) DMA 有效标志 (DMACm.DMSTS.ACT 位)

此标志表示 DMACm 停止或者正在运行。当 DMAC 开始传送时，DMACm.DMSTS.ACT 位为“1”；当结束 1 次传送请求的数据传送时，此标志变为“0”。

即使在 DMA 传送过程中通过给 DMACm.DMCNT.DTE 位写“0”来停止 DMA 传送，DMACm.DMSTS.ACT 位也将“1”的状态保持到 DMA 传送结束为止。

(7) 传送结束中断标志 (DMACm.DMSTS.DTIF 标志)

当通过 DMA 传送结束总传送长度的传送时, 将 DMACm.DMSTS.DTIF 标志置“1”。

如果 DMACm.DMSTS.DTIF 标志为“1”并且 DMACm.DMINT.DTIE 位为“1”, 就产生传送结束中断请求。

在 DMA 传送的总线周期结束后 DMACm.DMSTS.ACT 位变为“0”并且结束传送时, DMACm.DMSTS.DTIF 标志变为“1”。

如果在中断处理过程中给 DMACm.DMCNT.DTE 位写“1”, DMACm.DMSTS.DTIF 标志就自动变为“0”。

(8) 传送退出中断标志 (DMACm.DMSTS.ESIF)

当发生重复大小结束中断请求或者扩展重复区域上溢中断请求时, 将 DMACm.DMSTS.ESIF 标志置“1”。如果 DMACm.DMSTS.ESIF 标志为“1”并且 DMACm.DMINT.ESIE 位为“1”, 就产生传送退出中断请求。

在作为中断请求源的 DMA 传送总线周期结束后 DMACm.DMSTS.ACT 标志变为“0”并且结束传送时, DMACm.DMSTS.ESIF 标志变为“1”。

如果在中断处理过程中给 DMACm.DMCNT.DTE 位写“1”, DMACm.DMSTS.ESIF 标志就自动变为“0”。

要将 DMAC 的中断请求通知 CPU 或者 DTC 时, 需要设定中断控制寄存器, 详细内容请参照“14. 中断控制器 (ICUb)”。

17.3.10 通道的优先级

当有多个 DMA 传送请求时, DMAC 判断有 DMA 传送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, 就在最后的数据传送结束后进行通道仲裁, 开始优先级高的通道的传送。

17.4 DMA 传送的结束

DMA 传送的结束因传送结束条件而不同。一旦 DMA 传送结束, DMACm.DMCNT.DTE 位和 DMACm.DMSTS.ACT 标志就从“1”变为“0”, 表示 DMA 传送已经结束。

17.4.1 通过设定的总数据传送结束传送

(1) 正常传送模式 (DMACm.DMTMD.MD[1:0]=00b)

当 DMACm.DMCRAL 寄存器的值从“1”变为“0”时, 结束对应通道的 DMA 传送并且将 DMACm.DMCNT.DTE 位清“0”, 同时将 DMACm.DMSTS.DTIF 标志置“1”。此时, 如果 DMACm.DMINT.DTIE 位为“1”, 就向 CPU 或者 DTC 请求传送结束中断。

(2) 重复传送模式 (DMACm.DMTMD.MD[1:0]=01b)

当 DMACm.DMCRB 寄存器的值从“1”变为“0”时, 结束对应通道的 DMA 传送并且将 DMACm.DMCNT.DTE 位清“0”, 同时将 DMACm.DMSTS.DTIF 标志置“1”。此时, 如果 DMACm.DMINT.DTIE 位为“1”, 就向 CPU 或者 DTC 请求中断。

(3) 块传送模式 (DMACm.DMTMD.MD[1:0]=10b)

当 DMACm.DMCRB 寄存器的值从“1”变为“0”时, 结束对应通道的 DMA 传送并且将 DMACm.DMCNT.DTE 位清“0”, 同时将 DMACm.DMSTS.DTIF 标志置“1”。此时, 如果 DMACm.DMINT.DTIE 位为“1”, 就向 CPU 或者 DTC 请求中断。

要将 DMAC 的中断请求通知 CPU 或者 DTC 时, 需要设定中断控制寄存器, 详细内容请参照“14. 中断控制器 (ICUb) ”。

17.4.2 通过重复大小结束中断结束传送

在重复传送模式中, 当 DMACm.DMINT.RPTIE 位为“1”时, 在 1 个重复大小的传送结束后发生重复大小结束中断请求。通过中断请求结束 DMA 传送并且将 DMACm.DMCNT.DTE 位清“0”, 同时将 DMACm.DMSTS.ESIF 标志置“1”。此时, 如果 DMACm.DMINT.ESIE 位为“1”, 就向 CPU 或者 DTC 请求中断。如果在此状态下将 DMACm.DMCNT.DTE 位置“1”, 就能重新开始传送。

即使在块传送模式中, 也能产生重复大小结束中断请求。在块传送模式中, 在 1 块数据传送结束后同样发生重复大小结束中断请求。

要将 DMAC 的中断请求通知 CPU 或者 DTC 时, 需要设定中断控制寄存器, 详细内容请参照“14. 中断控制器 (ICUb) ”。

17.4.3 通过扩展重复区域上溢中断结束传送

在指定扩展重复区域并且 DMACm.DMINT.SARIE 位或者 DMACm.DMINT.DARIE 位为“1”时, 如果地址的扩展重复区域发生上溢, 就产生扩展重复区域上溢中断请求。通过中断请求结束 DMA 传送并且将 DMACm.DMCNT.DTE 位清“0”, 同时将 DMACm.DMSTS.ESIF 标志置“1”。此时, 如果 DMACm.DMINT.ESIE 位为“1”, 就向 CPU 或者 DTC 请求中断。

即使在读周期中发生扩展重复区域上溢中断请求, 也执行后续的写周期处理。

在块传送模式中, 即使在 1 块数据传送过程中发生扩展重复区域上溢中断请求, 也进行 1 块数据的传送。在块大小的边界, 通过扩展重复区域上溢中断结束传送。

要将 DMAC 的中断请求通知 CPU 或者 DTC 时, 需要设定中断控制寄存器, 详细内容请参照“14. 中断控制器 (ICUb) ”。

17.5 中断

DMAC 能按各通道，在 1 次请求的传送结束后将中断请求输出到 CPU 或者 DTC。如果传送目标为内部外围总线，就在完成写缓冲器的写操作时而不是在完成实际传送目标的写操作时发行中断请求。

中断源、标志和允许位的关系如表 17.8 所示，中断输出的概略逻辑图如图 17.13 所示。通过 DMAC 中断处理重新开始或者中止 DMA 传送的步骤如图 17.14 所示。

表 17.8 中断源、标志和允许位的关系

中断源	允许位	状态标志	请求输出的允许
传送结束	—	DMACm.DMSTS.DTIF	DMACm.DMINT.DTIE
退出传送结束	重复大小结束	DMACm.DMINT.RPTIE	DMACm.DMINT.ESIE
	传送源地址的扩展重复区域发生上溢	DMACm.DMINT.SARIE	
	传送目标地址的扩展重复区域发生上溢	DMACm.DMINT.DARIE	

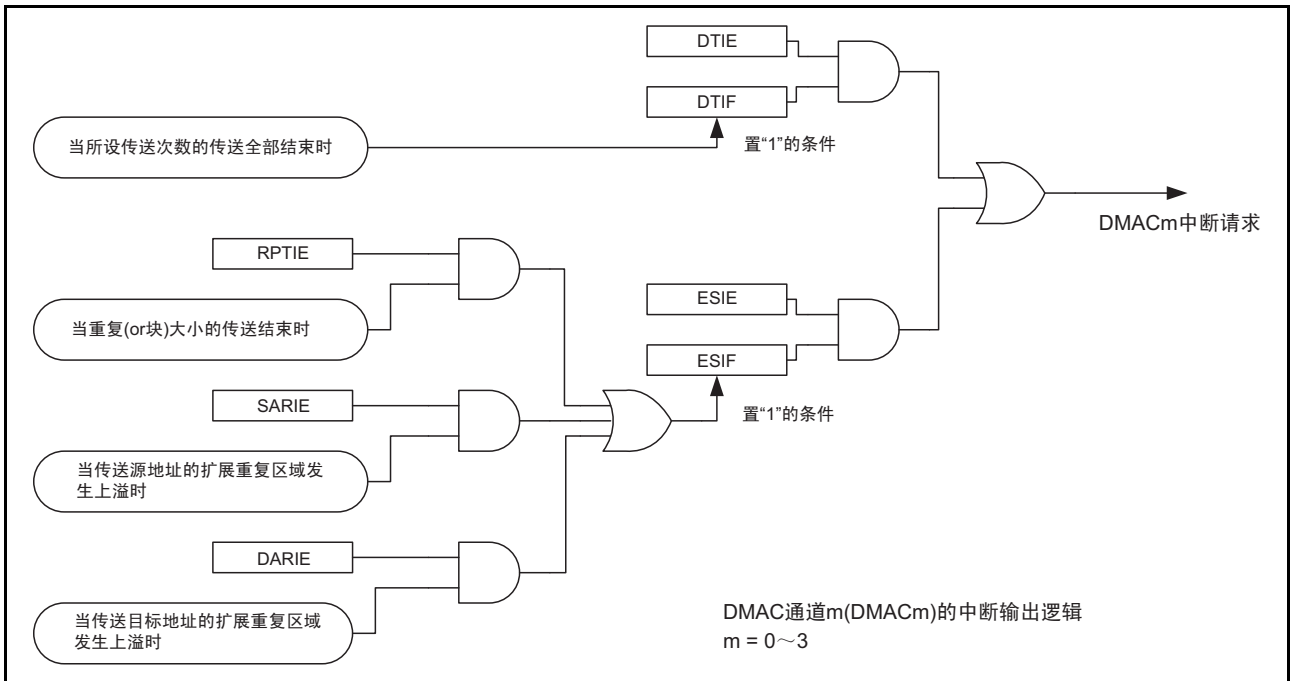


图 17.13 中断输出的概略逻辑图

在结束或者中止 DMA 传送时以及在继续传送时，通过中断处理程序解除中断后重新开始 DMA 传送的方法不同。

(1) 结束或者中止 DMA 传送的情况

在传送结束中断的情况下，必须给 DMACm.DMSTS.DTIF 标志写“0”；在重复大小中断和扩展重复区域上溢中断的情况下，必须给 DMACm.DMSTS.ESIF 标志写“0”。清除中断请求，DMACm 保持停止状态。此后，要进行新的 DMA 传送时，必须给所需寄存器写设定值，并且给 DMACm.DMCNT.DTE 位写“1”（允许 DMA 传送）。

(2) 继续进行 DMA 传送的情况

必须给 DMACm.DMCNT.DTE 位写“1”。自动将 DMACm.DMSTS.ESIF 标志清“0”（清除中断源）并且重新开始传送。

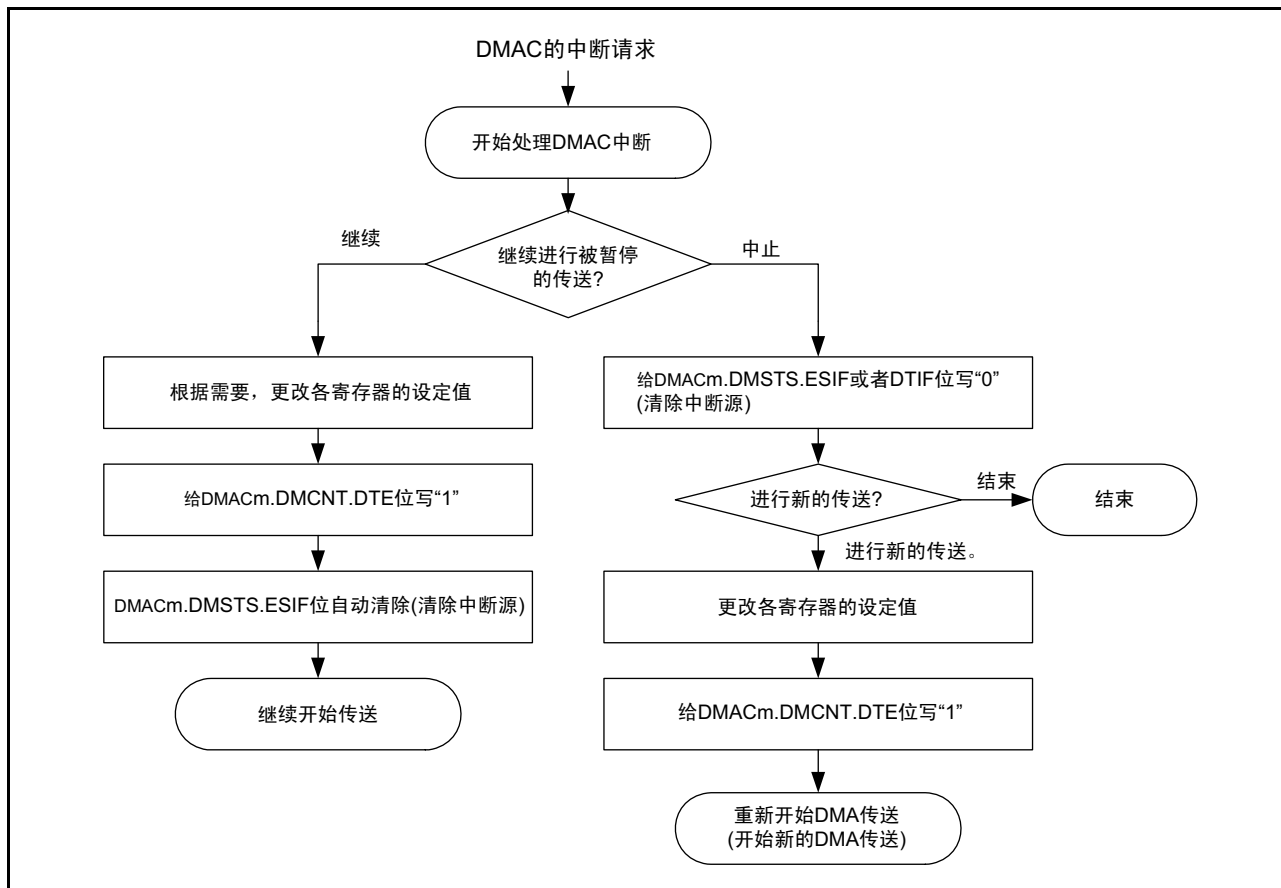


图 17.14 通过 DMAC 中断处理重新开始或者中止 DMA 传送的步骤

17.6 低功耗功能

在使用模块停止功能以及向全模块时钟停止模式、软件待机模式或者深度软件待机模式转移时，必须在给 DMAST.DMST 位写“0”（DMAC 模块停止）后分别进行以下的处理。

(1) 模块停止功能

通过给 MSTPCRA.MSTPA28 位写“1”（转移到模块停止状态），使 DMAC 的模块停止功能有效。如果在给 MSTPCRA.MSTPA28 位写“1”时正在进行 DMA 传送，就在 DMA 传送结束后转移到模块停止状态。

不能在 MSTPCRA.MSTPA28 位为“1”时存取 DMAC 的寄存器。

通过给 MSTPCRA.MSTPA28 位写“0”（解除模块停止状态），解除 DMAC 的模块停止状态。

(2) 全模块时钟停止模式

必须按照“11. 低功耗功能”的“11.5.2.1 向全模块时钟停止模式的转移”的步骤进行设定。

如果在执行 WAIT 指令时正在进行 DMA 传送，就在 DMA 传送结束后转移到全模块时钟停止模式。

在从全模块时钟停止模式返回后，通过给 MSTPCRA.MSTPA28 位写“0”，解除 DMAC 的模块停止状态。

(3) 软件待机模式和深度软件待机模式

必须按照“11. 低功耗功能”的“11.5.3.1 向软件待机模式的转移”或者“11.5.4.1 向深度软件待机模式的转移”的步骤进行设定。

如果在执行 WAIT 指令时正在进行 DMA 传送，就在 DMA 传送结束后转移到软件待机模式或者深度软件待机模式。

(4) 使用低功耗功能时的注意事项

有关 WAIT 指令和寄存器的设定步骤，请参照“11. 低功耗功能”的“11.6.6 WAIT 指令的执行时序”。

要在从低功耗模式返回后进行 DMA 传送时，必须再次给 DMAST.DMST 位写“1”。

当全模块时钟停止模式期间或者软件待机模式期间发生的请求不是 DMAC 启动请求而是 CPU 中断请求时，必须按照“14. 中断控制器 (ICUb)”的“14.5.3 中断请求目标的选择”的设定方法，在将中断请求源更改为 CPU 后执行 WAIT 指令。

17.7 使用时的注意事项

17.7.1 外围模块的 DMA 传送情况

在进行外围模块的 DMA 传送过程中，在从开始写最后数据到外围总线存取结束前，DMACm.DMSTS.ACT 标志有可能变为“0”（DMAC 停止）。

17.7.2 有关 DMA 运行过程中的寄存器存取

在 DMACm.DMSTS.ACT 标志为“1”（DMAC 正在运行）或者 DMACm.DMCNT.DTE 位为“1”（允许 DMA 传送）的状态下，不能存取相同通道的设定寄存器（DMSAR、DMDAR、DMCRA、DMCRB、DMTMD、DMINT、DMAMD、DMOFR、DMCSL）。

17.7.3 有关保留区的 DMA 传送

因为禁止保留区的 DMA 传送，所以无法保证存取保留区时的传送结果。保留区的详细内容请参照“4. 地址空间”。

17.7.4 因设定 DMA 启动源标志控制寄存器（DMCSL）而产生的各传送结束中断请求

如果将 DMACm.DMCSL.DISEL 位置“1”，就能在每结束 1 次 DMA 启动请求的传送时向 CPU 请求中断。此时发生的中断不同于 DMAC 输出的传送结束中断和退出结束中断，如果在 DMA 传送结束时不将作为 DMAC 启动源的中断标志清“0”，就通过将中断请求目标更改为 CPU，向 CPU 请求中断。在接受 CPU 中断时清除中断标志。

有关中断标志和中断请求目标的变更，请参照“14. 中断控制器（ICUb）”，DMACm.DMCSL.DISEL 位的设定请参照“17.2.12 DMA 启动源标志控制寄存器（DMCSL）”。

17.7.5 中断控制器的 DMAC 启动请求寄存器（ICU.DMRSRm）的设定

必须在 DMA 传送允许位（DMACm.DMCNT.DTE 位）为“0”（禁止 DMA 传送）时设定 DMAC 启动请求寄存器（ICU.DMRSRm）。不能将和 ICU.DMRSRm 寄存器设定的向量号相同的向量号对应的 DTC 启动允许寄存器（ICU.DTCERn）置“1”。ICU.DTCERn 寄存器和 ICU.DMRSRm 寄存器的详细内容请参照“14. 中断控制器（ICUb）”。

17.7.6 保留或者重新开始 DMA 启动的方法

要保留 DMA 的启动请求时，必须给启动源的中断允许位（ICU.IERn.IENj 位）写“0”；要重新开始 DMA 时，必须在进行“17.3.7 DMAC 的启动”的设定后给 ICU.IERn.IENj 位写“1”。

18. 数据传送控制器 (DTCa)

RX63T 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。

18.1 概要

DTC 的规格和框图分别如表 18.1 和图 18.1 所示。

表 18.1 DTC 的规格

项目	内容
传送模式	<ul style="list-style-type: none"> • 正常传送模式 1 次启动进行 1 个数据的传送。 • 重复传送模式 1 次启动进行 1 个数据的传送。 如果传送重复大小的数据, 就返回到开始传送的地址。 重复大小最多能设定 256 个数据。 • 块传送模式 1 次启动进行 1 块数据的传送。 块大小最多能设定 256 个数据。
传送通道	<ul style="list-style-type: none"> • 能进行对应中断源的通道传送 (通过 ICU 的 DTC 启动请求进行传送)。 • 能对 1 个启动源进行多个的数据传送 (链传送)。 • 能选择是在计数器为 "0" 时进行链传送还是每次都进行链传送。
传送空间	<ul style="list-style-type: none"> • 在短地址模式中为 16M 字节。 (0000 0000h ~ 007F FFFFh 和 FF80 0000h ~ FFFF FFFFh 中保留区除外的区域) • 在全地址模式中为 4G 字节。 (0000 0000h ~ FFFF FFFFh 中保留区除外的区域)
数据传送单位	<ul style="list-style-type: none"> • 1 个数据的位长: 8 位、16 位、32 位 • 块大小的数据量: 1 ~ 256 个数据
CPU 中断请求	<ul style="list-style-type: none"> • 能通过启动 DTC 的中断向 CPU 请求中断。 • 在 1 次数据传送结束后向 CPU 请求中断。 • 在指定数据量的数据传送结束后向 CPU 请求中断。
跳读	能指定传送信息的跳读。
回写的省略	当传送源地址被固定或者传送目标被固定时省略回写。
低功耗功能	能设定为模块停止状态。

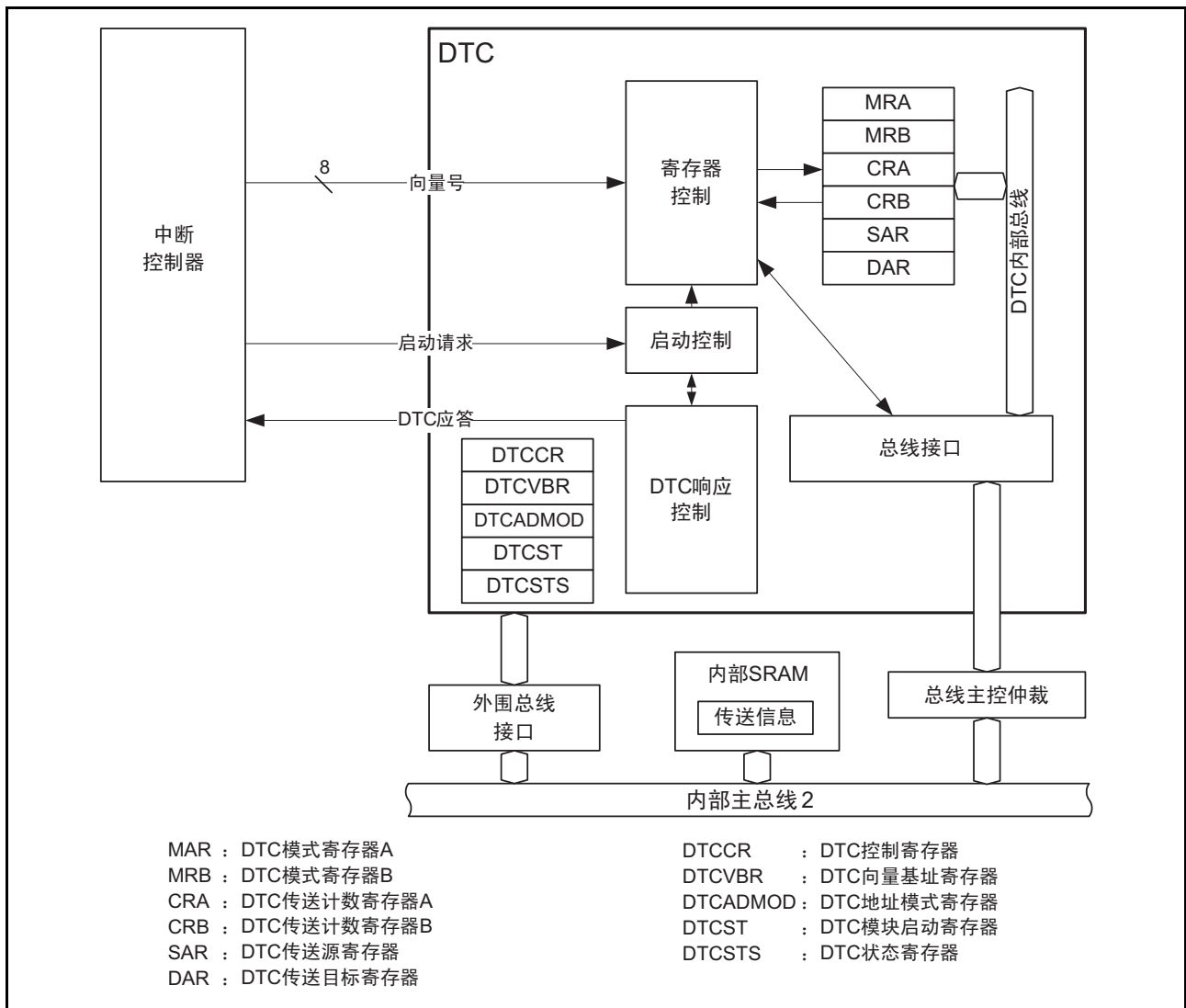


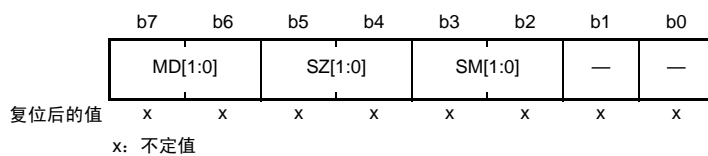
图 18.1 DTC 的框图

18.2 寄存器说明

MRA、MRB、SAR、DAR、CRA、CRB 寄存器是 DTC 的内部寄存器，不能从 CPU 直接存取，这些内部寄存器的设定值作为传送信息被分配到内部 RAM 区。如果发生 DTC 启动请求，就从内部 RAM 区读传送信息并且将读到的传送信息设定到内部寄存器。在进行数据传送后，DTC 的内部寄存器的内容作为传送信息被回写到内部 RAM 区。

18.2.1 DTC 模式寄存器 A (MRA)

地址 (CPU 不能直接存取)

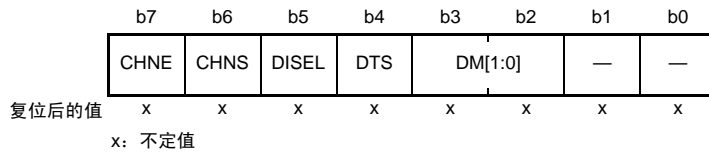


位	符号	位名	功能	R/W
b1-b0	—	保留位	读取值为不定值，只能写“0”。	—
b3-b2	SM[1:0]	传送源地址的寻址方式位	b3 b2 0 0: SAR 寄存器为地址固定 (省略 SAR 寄存器的回写) 0 1: SAR 寄存器为地址固定 (省略 SAR 寄存器的回写) 1 0: 传送后 SAR 寄存器递增 (当 SZ[1:0] 位为“00b”时加 1；为“01b”时加 2；为“10b”时加 4) 1 1: 传送后 SAR 寄存器递减 (当 SZ[1:0] 位为“00b”时减 1；为“01b”时减 2；为“10b”时减 4)	—
b5-b4	SZ[1:0]	DTC 数据传送长度位	b5 b4 0 0: 8 位 (字节) 传送 0 1: 16 位 (字) 传送 1 0: 32 位 (长字) 传送 1 1: 不能设定	—
b7-b6	MD[1:0]	DTC 传送模式选择位	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	—

CPU 不能直接存取 MRA 寄存器。

18.2.2 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]	传送目标地址的寻址方式位	b3 b2 0 0: DAR 寄存器为地址固定 (省略 DAR 寄存器的回写) 0 1: DAR 寄存器为地址固定 (省略 DAR 寄存器的回写) 1 0: 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) 1 1: 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: 只在传送计数器从“0”变为“1”或者从“1”变为 CRAH 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	—

CPU 不能直接存取 MRB 寄存器。

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。

CHNS 位 (DTC 链传送选择位)

此位选择链传送的条件。

当 CHNE 位为“0”时, 忽视 CHNS 位的设定。有关链传送选择条件的详细内容, 请参照“表 18.3 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, 并且不清除中断状态标志, 也不向 CPU 请求中断。

CHNE 位 (DTC 链传送允许位)

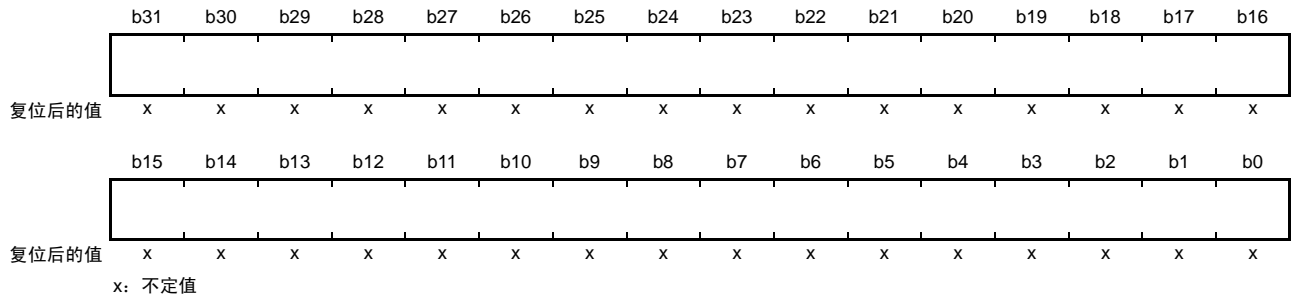
此位指定链传送。

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“18.4.6 链传送”。

18.2.3 DTC 传送源寄存器 (SAR)

地址 (CPU 不能直接存取)



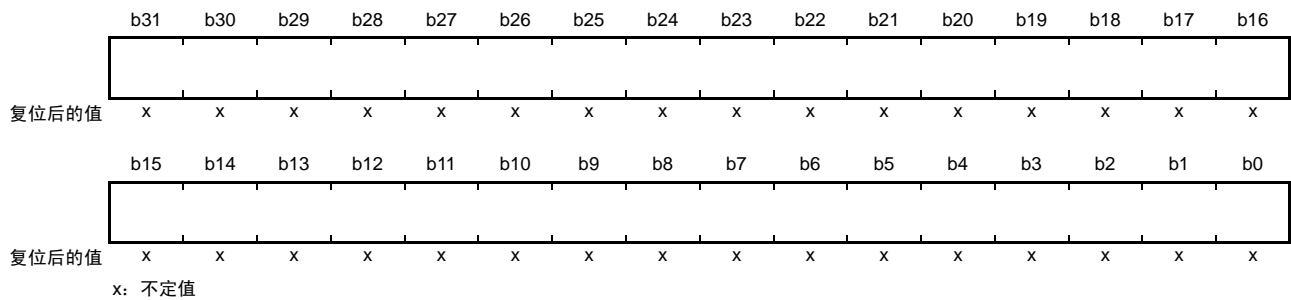
在全地址模式中 32 位有效。

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。

18.2.4 DTC 传送目标寄存器 (DAR)

地址 (CPU)



在全地址模式中 32 位有效。

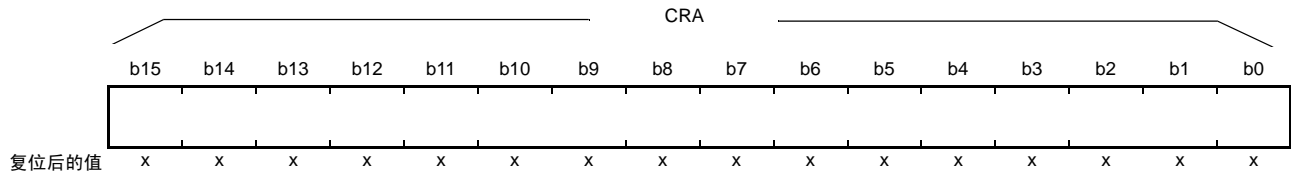
在短地址模式中低 24 位有效，忽视高 8 位 (b31 ~ b24) 的设定，通过 b23 指定的值进行位扩展。

CPU 不能直接存取 DAR 寄存器。

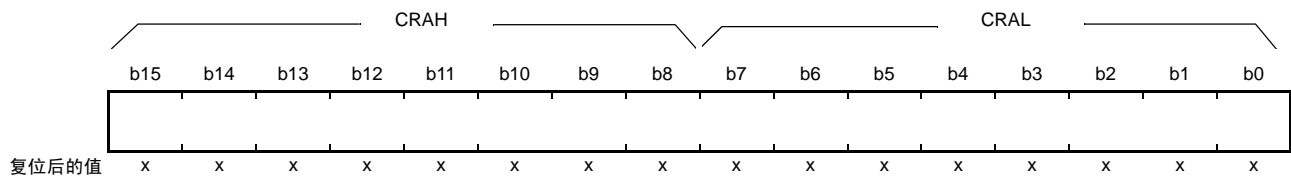
18.2.5 DTC 传送计数寄存器 A (CRA)

地址 (CPU 不能直接存取)

• 正常传送模式



• 重复传送模式、块传送模式



x: 不定值

注: 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: 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。

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)

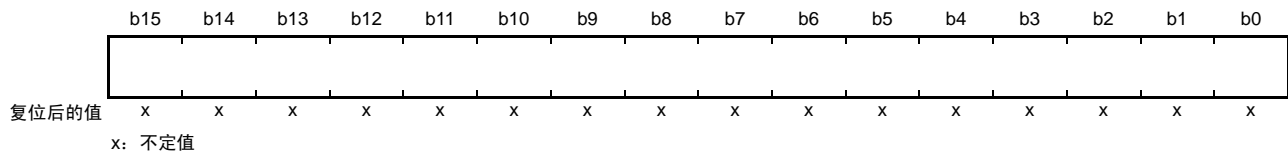
CRAH 寄存器保持块的大小, CRAL 寄存器用作 8 位块大小计数器。

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, 传送 CRAH 寄存器的值。

18.2.6 DTC 传送计数器 B (CRB)

地址 (CPU 不能直接存取)



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。

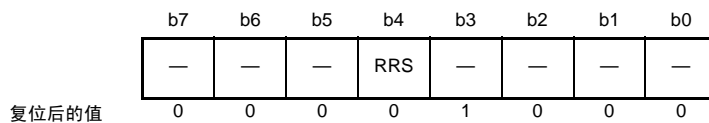
当进行 1 块大小的最后数据传送时，CRB 寄存器减 1。

在设定为正常传送模式和重复传送模式时，不使用 CRB 寄存器，忽视设定值。

CPU 不能直接存取 CRB 寄存器。

18.2.7 DTC 控制寄存器 (DTCCR)

地址 0008 2400h



位	符号	位名	功能	R/W
b2-b0	—	保留位	读写值都为“0”。	R/W
b3	—	保留位	读写值都为“1”。	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

RRS 位 (DTC 传送信息的跳读允许位)

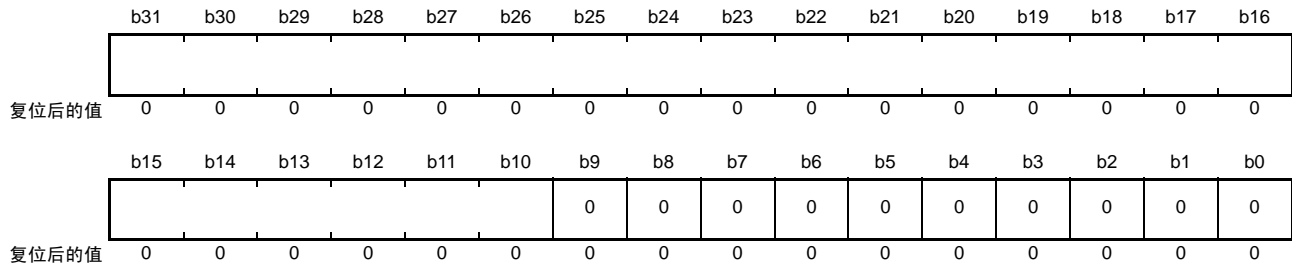
DTC 向量号随时和上次启动的向量号进行比较。

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。

18.2.8 DTC 向量基址寄存器 (DTCVBR)

地址 0008 2404h



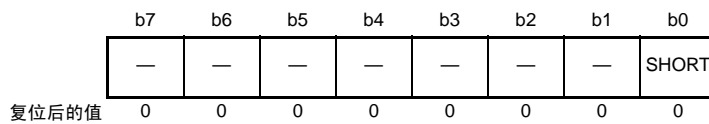
位	位名	功能	R/W
b9-b0	DTC 向量基址 (低 10 位)	读写值都为“0”。	R
b31-b10	DTC 向量基址 (高 22 位)	忽视高 4 位 (b31 ~ b28) 的写操作, 通过 b27 指定的值进行位扩展。	R/W

DTCVBR 寄存器设定在计算 DTC 向量表地址时的基址。

能在 0000 0000h ~ 07FF FC00h 和 F800 0000h ~ FFFF FC00h 的范围内, 以 1K 字节为单位进行设定。

18.2.9 DTC 地址模式寄存器 (DTCADMOD)

地址 0008 2408h



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

DTCADMOD 寄存器是设定 DTC 能存取的地域的寄存器。

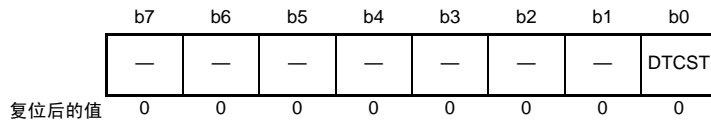
SHORT 位 (短地址模式设定位)

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 ~ 007FFFFFFh 和 FF800000h ~ FFFFFFFFh)。

18.2.10 DTC 模块启动寄存器 (DTCST)

地址 0008 240Ch



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 位 (DTC 模块启动位)

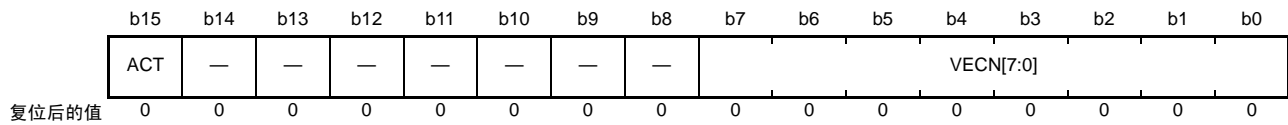
为了使 DTC 能接受传送请求, 必须将 DTCST 位置“1”。如果将 DTCST 位置“0”, 就不能接受新的传送请求。如果在运行过程中将此位改写为“0”, 已接受的传送请求就保持到处理结束为止。

在使用模块停止功能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”。

有关使用模块停止功能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“18.8 低功耗功能”和“11. 低功耗功能”。

18.2.11 DTC 状态寄存器 (DTCSTS)

地址 0008 240Eh



位	符号	位名	功能	R/W
b7-b0	VECN[7:0]	DTC 激活向量号监视位	在 DTC 传送过程中，用向量号表示该传送的启动源。只在 DTC 传送过程中 (ACT 标志为“1”)，此值为有效值。	R
b14-b8	—	保留位	读取值为“0”，写操作无效。	R
b15	ACT	DTC 有激活志	0: 不在进行 DTC 传送 1: 正在进行 DTC 传送	R

VECN[7:0] 位 (DTC 激活向量号监视位)

在 DTC 传送过程中，用向量号表示该传送的启动源。

如果在读 DTCSTS 寄存器时 ACT 标志为“1” (正在进行 DTC 传送)，读到的 VECN[7:0] 就为有效值；如果在读 DTCSTS 寄存器时 ACT 标志为“0” (不在进行 DTC 传送)，读到的 VECN[7:0] 就为无效值。

DTC 启动源和向量地址的关系请参照“14. 中断控制器 (ICUb)”的“表 14.3 中断向量表”。

ACT 标志 (DTC 激活标志)

表示 DTC 的传送状态。

[为“1”的条件]

- 当通过传送请求进行了 DTC 启动时

[为“0”的条件]

- 当对于 1 次传送请求 DTC 运行结束时

18.3 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 DTCERn.DTCE 位 (n 为中断向量号) 置“1”，该中断就为 DTC 的启动源。

DTC 启动源和向量地址的关系请参照“14. 中断控制器 (ICUb)”的“表 14.3 中断向量表”。软件启动请参照“14. 中断控制器 (ICUb)”的“14.2.5 软件中断启动寄存器 (SWINTR)”。

一旦接受启动请求，DTC 就在该 1 次请求的传送结束前不接受新的启动请求，与优先级无关。如果在 DMAC/DTC 传送过程中产生多个启动源，就在该传送结束时接受最高优先级的请求。如果在 DTC 模块启动位 (DTCST.DTCST) 为“0”的状态下产生多个启动请求，就在将 DTCST.DTCST 位置“1”时接受最高优先级的请求。

在 1 次数据传送 (在链传送时，为连续的最后传送) 时，DTC 进行以下的运行：

- 在指定的总传送量的传送结束时，在数据传送后将 DTCERn.DTCE 位置“0”，然后向 CPU 请求中断。
- 当 MRB.DISEL 位为“1”时，在数据传送后向 CPU 请求中断。
- 如果是上述以外的情况，就在开始数据传送时将作为启动源的中断状态标志置“0”。

18.3.1 传送信息的分配和 DTC 向量表

DTC 按启动源从向量表读传送信息的起始地址，并且从该起始地址读传送信息。

必须分配向量表的基址 (起始地址)，使低 10 位变为“0”。必须给 DTC 向量基址寄存器 (DTCVBR) 设定 DTC 向量表的基址。

传送信息分配在内部 RAM 区。对于向量表的基址，向量号 n 的传送信息 (n) 的起始地址必须为地址 +4n。

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.SHORT 位设定短地址模式 (SHORT 位 =1) 或者全地址模式 (SHORT 位 =0)。

DTC 向量表和传送信息的对应如图 18.2 所示。

内部 RAM 区内传送信息的分配如图 18.3 所示。低位地址因分配区域的字节序而不同，详细内容请参照“18.9.2 传送信息的分配”。

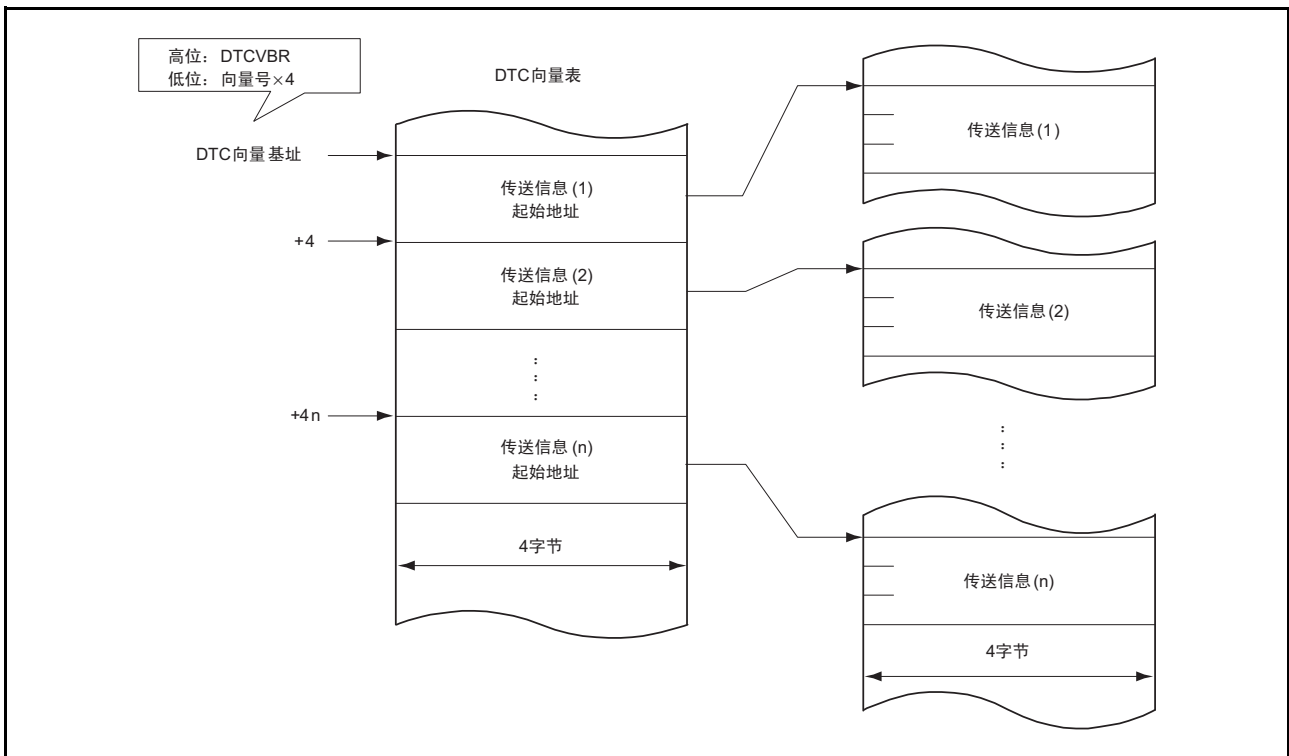


图 18.2 DTC 向量表和传送信息的对应

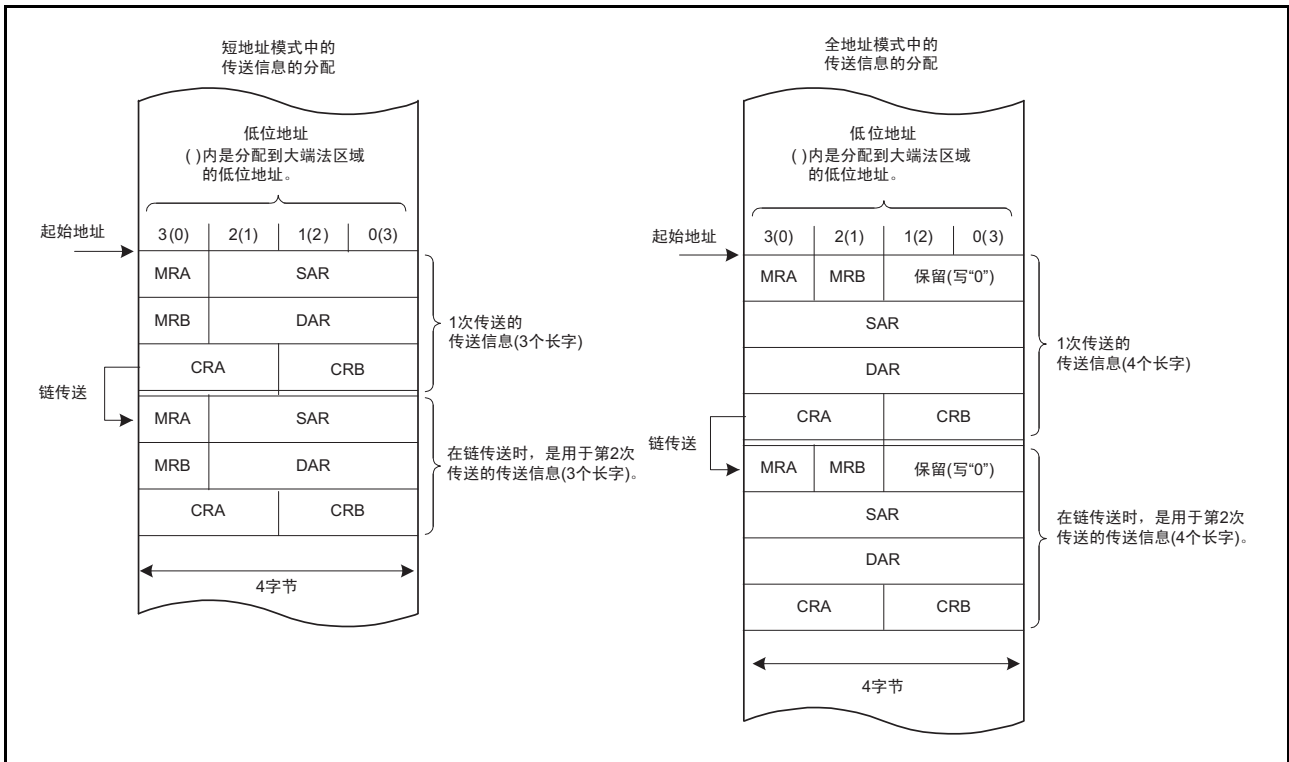


图 18.3 内部 RAM 区内的传送信息的分配

18.4 运行说明

DTC 以传送信息为准进行数据传送。为了使 DTC 运行，需要预先将传送信息保存到 RAM 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内部 RAM 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 18.2 所示。

表 18.2 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，在指定的数据传送结束时进行链传送的设定。

DTC 的运行流程图如图 18.4 所示，链传送的条件如表 18.3 所示。

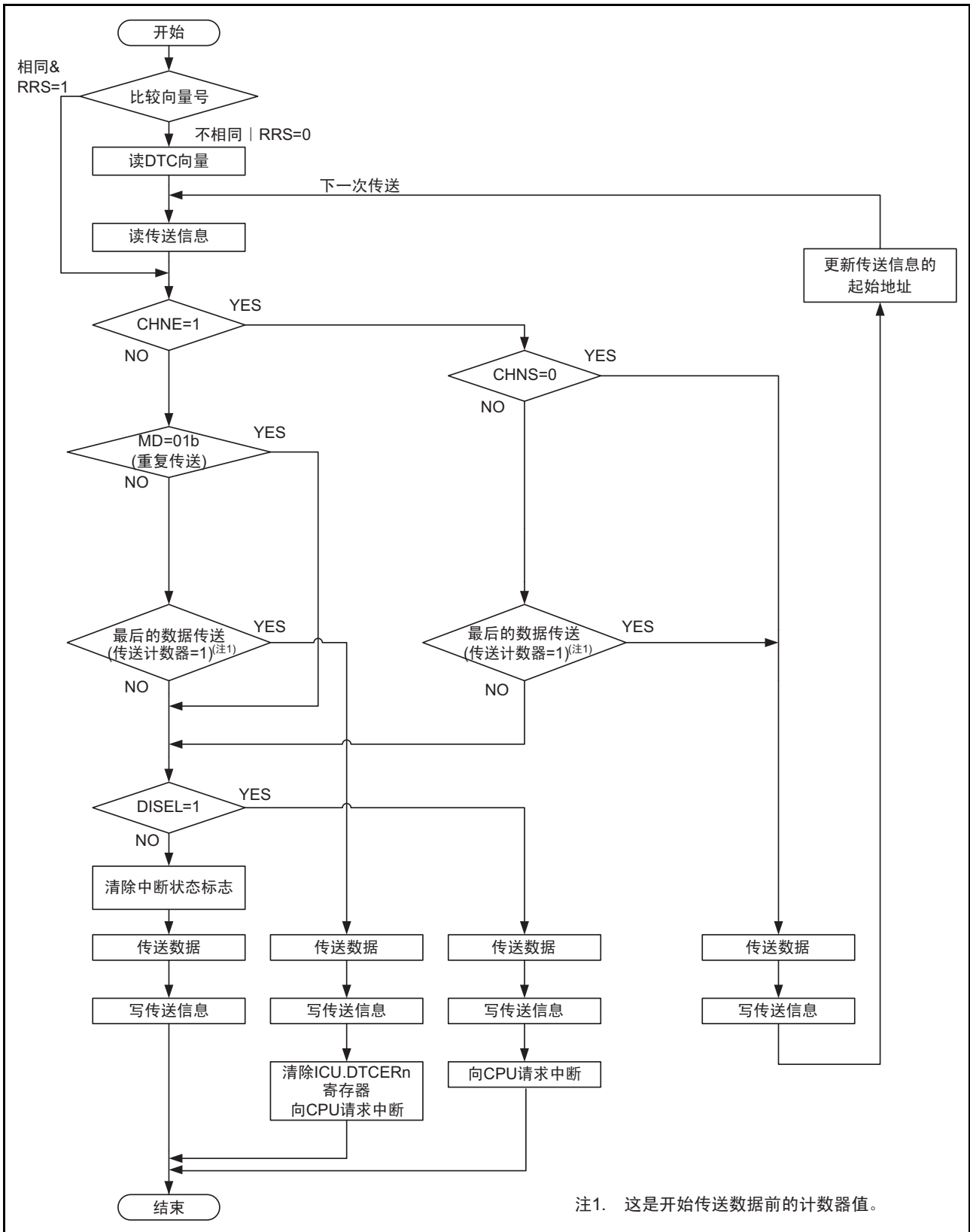


图 18.4 DTC 的运行流程图

表 18.3 链传送的条件

第 1 次传送				第 2 次传送 (注 3)				DTC 传送
CHNE 位	CHNS 位	DISEL 位	传送计数器 (注 1、注 2)	CHNE 位	CHNS 位	DISEL 位	传送计数器 (注 1、注 2)	
0	—	0	(1→0) 以外	—	—	—	—	到第 1 次传送为止结束。
0	—	0	(1→0)	—	—	—	—	到第 1 次传送为止结束并且 向 CPU 请求中断。
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	(1→0) 以外	到第 2 次传送为止结束。
				0	—	0	(1→0)	到第 2 次传送为止结束并且 向 CPU 请求中断。
				0	—	1	—	
1	1	0	(1→*) 以外	—	—	—	—	到第 1 次传送为止结束。
1	1	—	(1→*)	0	—	0	(1→0) 以外	到第 2 次传送为止结束。
				0	—	0	(1→0)	到第 2 次传送为止结束并且 向 CPU 请求中断。
				0	—	1	—	
1	1	1	(1→*) 以外	—	—	—	—	到第 1 次传送为止结束并且 向 CPU 请求中断。

注 1. 传送计数器因各传送模式而不同，各传送模式中的传送计数器如下：

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. 在正常传送模式或者块传送模式中，传送结束时的计数器变化为 (1→0)；在重复传送模式中，最后传送时的计数器变化为 (1→CRAH)。表中的 (1→*) 表示两种情况的变化。

注 3. 在第 2 次传送或者第 2 次以后的传送时能选择链传送，但是在第 2 次传送时省略 CHNE 位为“1”的组合。

18.4.1 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”，就读传送信息，与 RRS 位的值无关。跳读传送信息的运行例子如图 18.13 所示。

要更新 DTC 向量表和传送信息时，必须先将 RRS 位置“0”并且在更新 DTC 向量表和传送信息后设定 RRS 位。一旦将 RRS 位置“0”，就取消被保持的向量号，在下次启动时能读被更新的 DTC 向量表和传送信息。

18.4.2 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 18.4 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 18.4 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		

18.4.3 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 18.5 和图 18.5 所示。

表 18.5 正常传送模式的寄存器功能

寄存器	功能	回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。

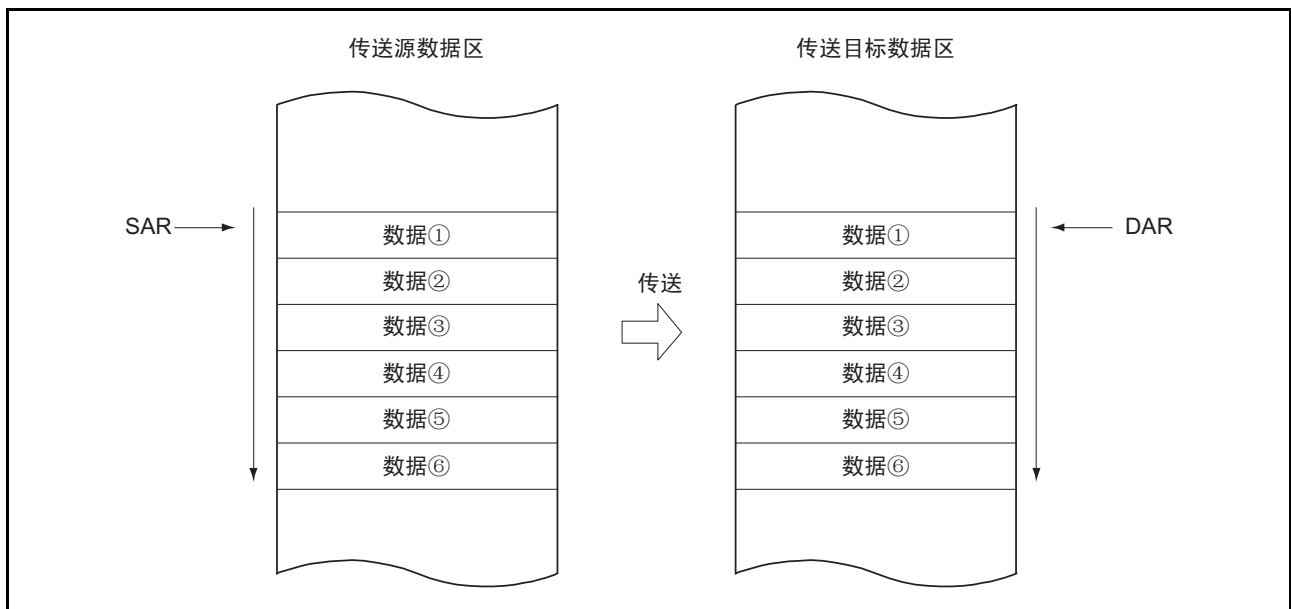


图 18.5 正常传送模式的存储器映像

18.4.4 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 18.6 和图 18.6 所示。

表 18.6 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 不为“1”	CRAL 为“1”
SAR	传送源地址	递增 / 递减 / 固定 (注 1)	(当 MRB.DTS 位为“0”时) 递增 / 递减 / 固定 (注 1) (当 MRB.DTS 位为“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 位为“0”时) DAR 寄存器的初始值 (当 MRB.DTS 位为“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。

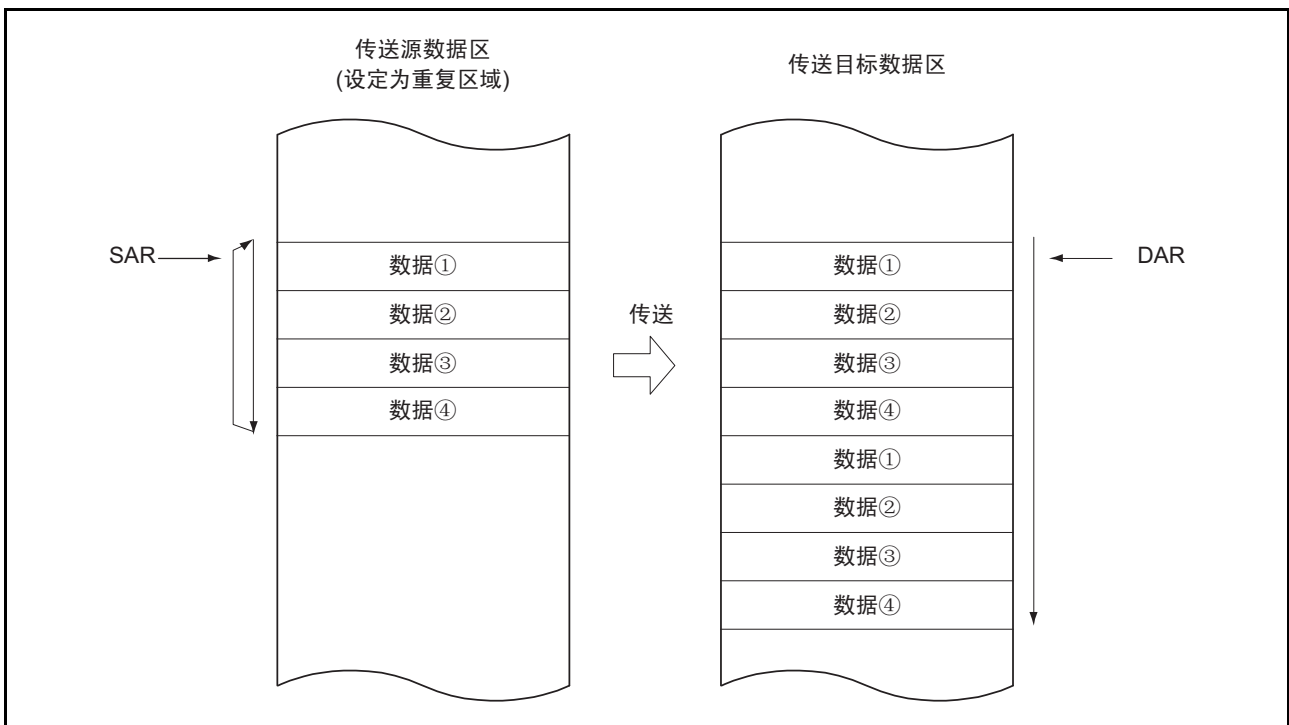


图 18.6 重复传送模式的存储器映像 (将传送源设定为重复区域的情况)

18.4.5 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（块数）能指定为 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 18.7 和图 18.7 所示。

表 18.7 块传送模式的寄存器功能

寄存器	功能	回写传送信息时被回写的值
SAR	传送源地址	（当 MRB.DTS 位为“0”时） 递增 / 递减 / 固定（注 1） （当 MRB.DTS 位为“1”时） SAR 寄存器的初始值
DAR	传送目标地址	（当 MRB.DTS 位为“0”时） DAR 寄存器的初始值 （当 MRB.DTS 位为“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。

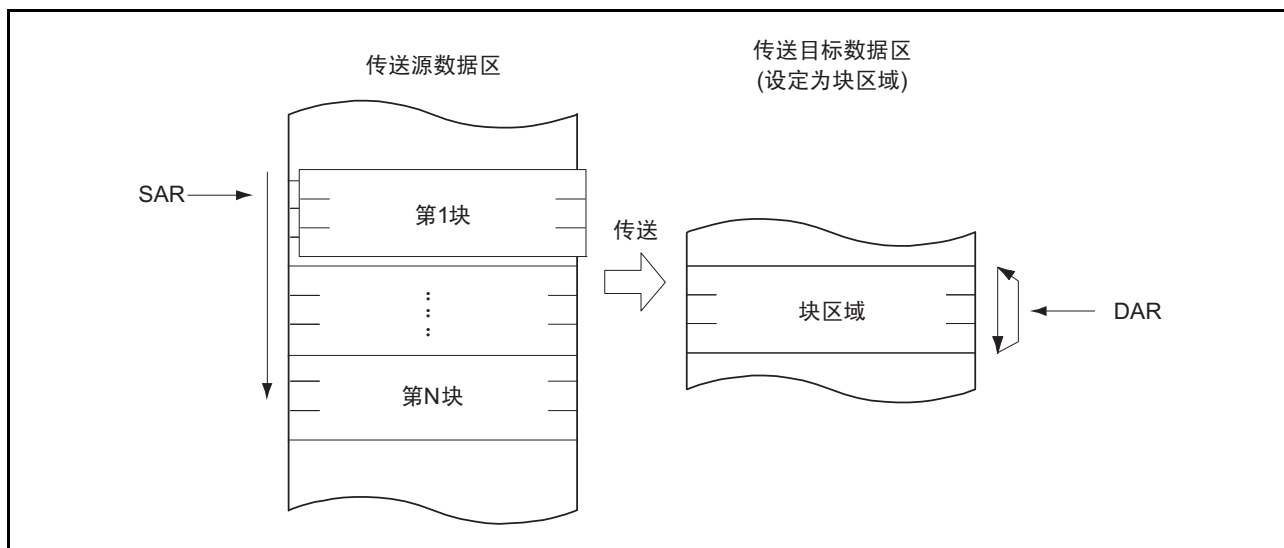


图 18.7 块传送模式的存储器映像（将传送目标指定为块区域的情况）

18.4.6 链传送

如果将 MRB.CHNE 位置“1”，就能进行链传送。链传送用 1 个启动源连续进行多个数据传送。

如果将 MRB.CHNE 位置“1”并且将 MRB.CHNS 位置“0”，就在指定的传送次数结束或者 MRB.DISEL 位为“1”（每次进行 DTC 数据传送时向 CPU 请求中断）时不向 CPU 请求中断，并且也不影响作为启动源的中断状态标志。

能分别设定要定义数据传送的 SAR、DAR、CRA、CRB、MRA 和 MRB 寄存器。链传送的运行如图 18.8 所示。

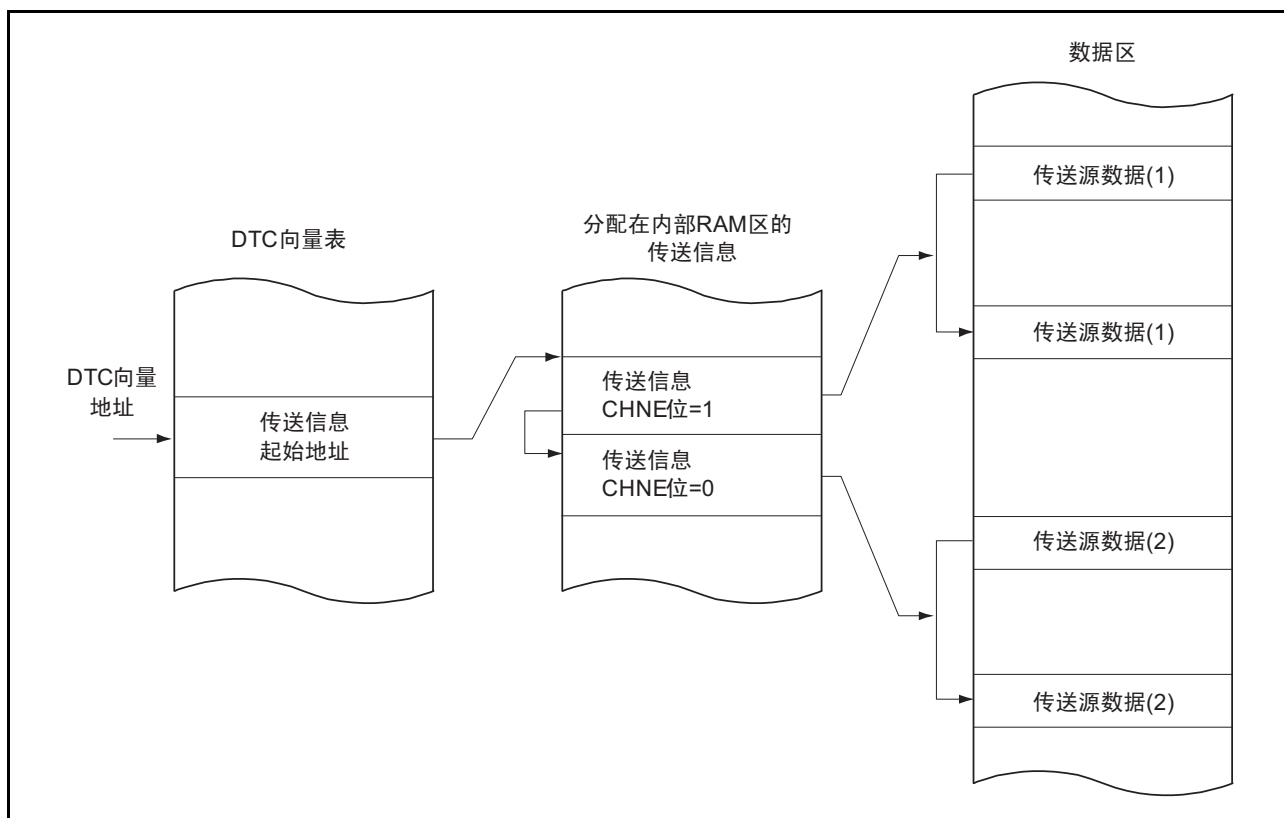


图 18.8 链传送的运行

如果将 MRB.CHNE 位和 MRB.CHNS 位置“1”，就只在指定的数据传送结束时进行链传送。即使在重复传送模式中，也在指定的数据传送结束时进行链传送。

有关链传送条件的详细内容，请参照“表 18.3 链传送的条件”。

18.4.7 运行时序

DTC 的运行时序例子如图 18.9 ~ 图 18.13 所示。

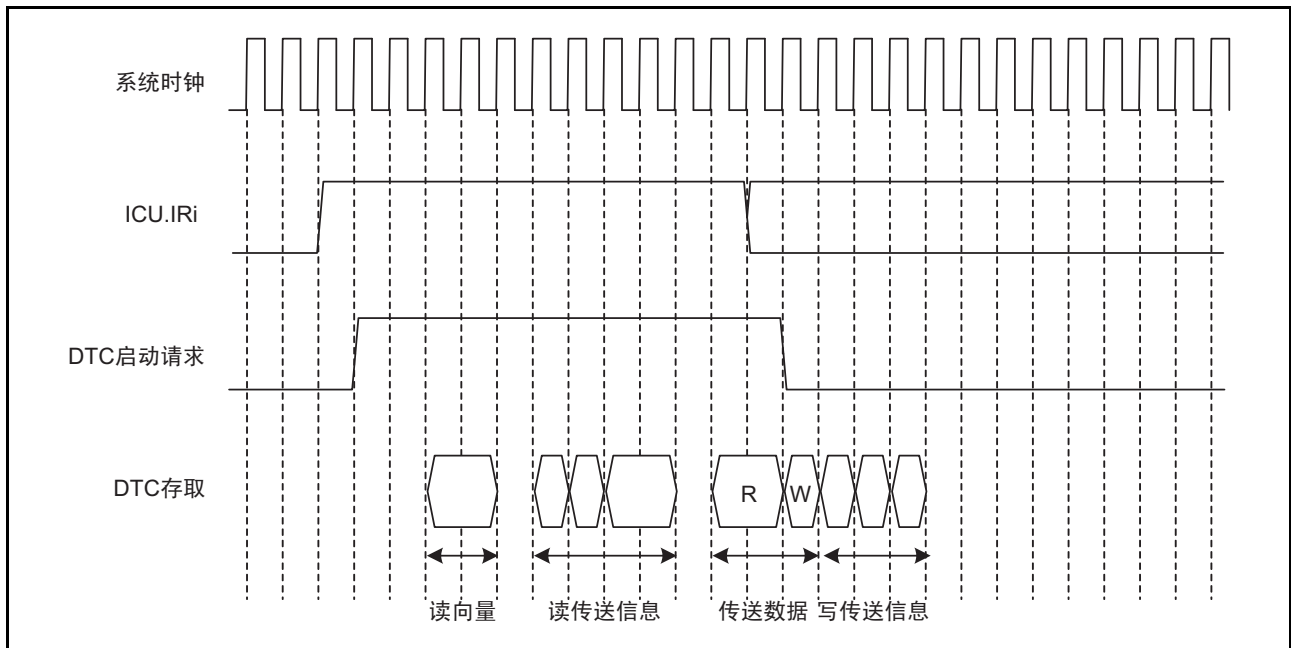


图 18.9 DTC 的运行时序例子 (1) (短地址模式、正常传送模式、重复传送模式的情况)

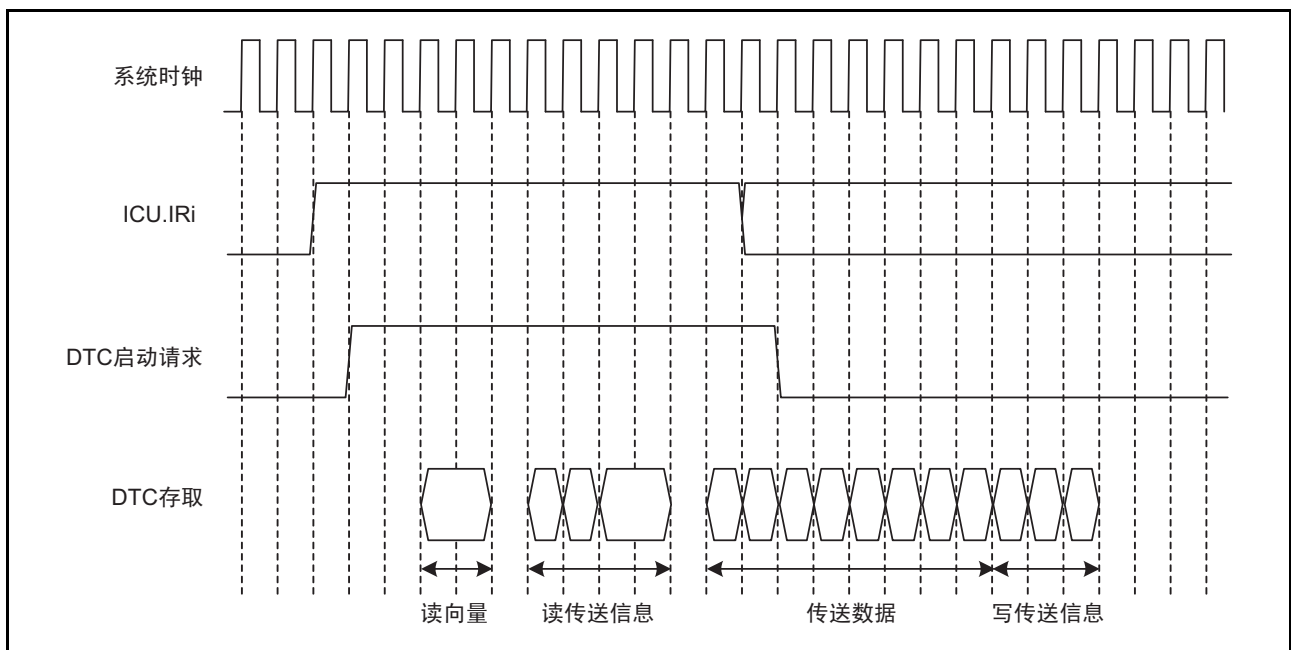


图 18.10 DTC 的运行时序例子 (2) (短地址模式、块传送模式、块大小为“4”的情况)

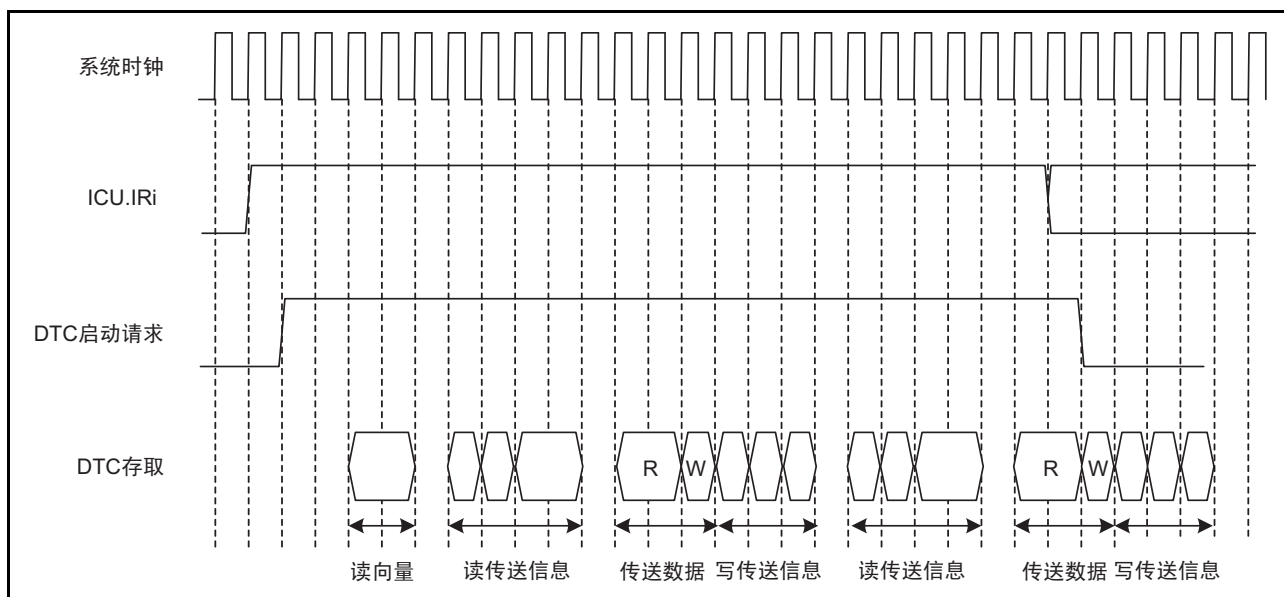


图 18.11 DTC 的运行时序例子 (3) (短地址模式、链传送的情况)

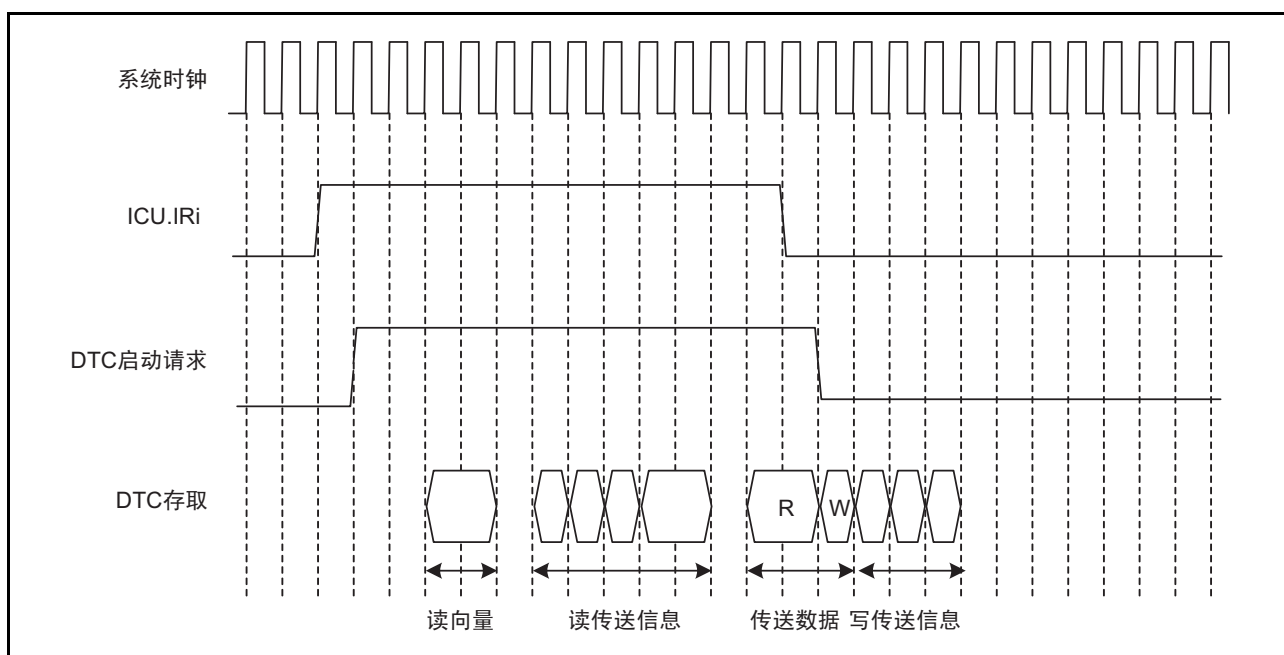


图 18.12 DTC 的运行时序例子 (4) (全地址模式、正常传送模式、重复传送模式的情况)

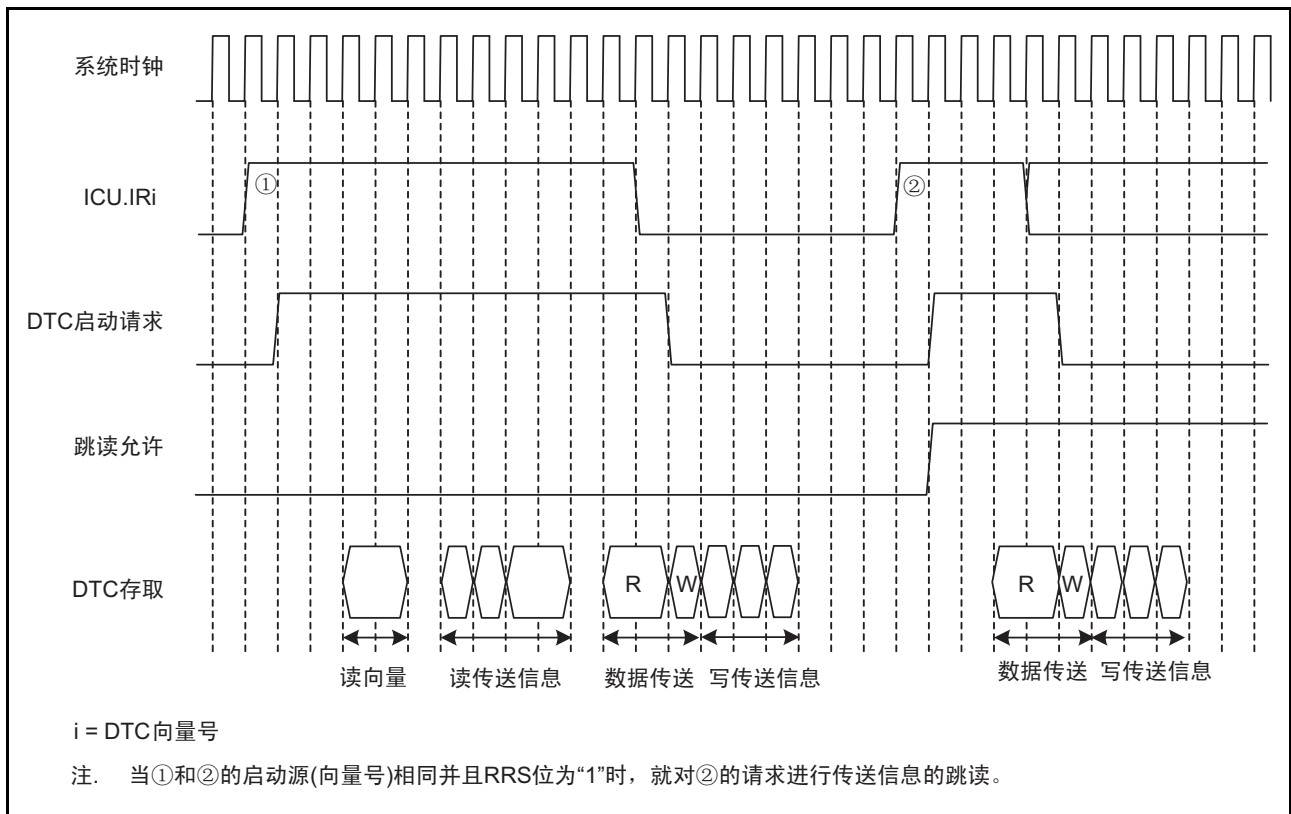


图 18.13 跳读传送信息的运行例子 (向量、传送信息、传送目标为内部 RAM, 传送源为外围模块的情况)

18.4.8 DTC 的执行周期

DTC 的 1 次数据传送的执行周期如表 18.8 所示。
各处理状态的执行顺序请参照“18.4.7 运行时序”。

表 18.8 DTC 的执行周期

传送模式	读向量		读传送信息			写传送信息			数据传送		内部运行	
									读	写		
正常	Cv+1	0 (注 1)	4×Ci+1 (注 2)	3×Ci+1 (注 3)	0 (注 1)	3×Ci (注 4)	2×Ci (注 5)	Ci (注 6)	Cr+1	Cw	2	0 (注 1)
重复									Cr+1	Cw		
块 (注 7)									P×Cr	P×Cw		

- 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 寄存器和 DAR 寄存器都不为地址固定的情况。
 注 5. SAR 寄存器或者 DAR 寄存器为地址固定的情况。
 注 6. SAR 寄存器和 DAR 寄存器为地址固定的情况。
 注 7. 这是块大小大于等于“2”的情况。当块大小为“1”时，为正常传送的周期数。

P: 块大小 (CRAH 寄存器和 CRAL 寄存器的设定值)

Cv: 向量传送信息保存目标的存取周期

Ci: 传送信息保存目标的存取周期

Cr: 数据读目标的存取周期

Cw: 数据写目标的存取周期

(读向量、读传送信息、读数据传送的“+1”、内部运行的“2”的单位都是系统时钟 (ICLK)。)

(Cv、Ci、Cr 和 Cw 因存取目标而不同。各存取目标的周期数请参照“33. RAM”、“34. ROM (保存代码的闪存)”、“5. I/O 寄存器”。)

18.4.9 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。有关总线仲裁请参照“15. 总线”。

18.5 DTC 的设定步骤

必须在使用 DTC 前设定 DTC 向量基址寄存器 (DTCVBR)。

启动 DTC 所需的设定步骤如图 18.14 所示。

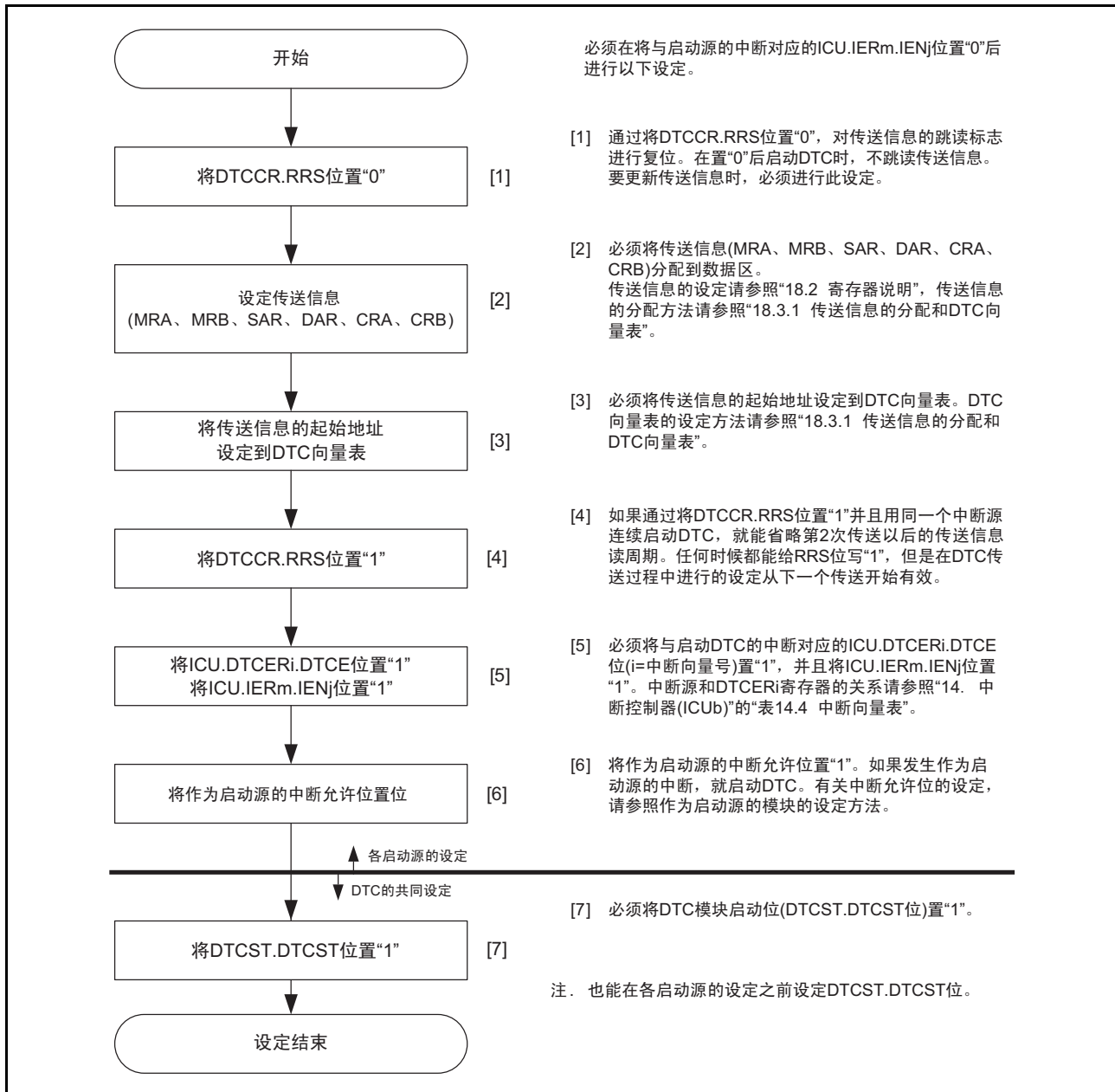


图 18.14 DTC 的设定步骤

18.6 DTC 的使用例子

18.6.1 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) 传送信息的设定

给 MRA 寄存器设定传送源地址固定 (MRA.SM[1:0] 位 =00b)、正常传送模式 (MRA.MD[1:0] 位 =00b) 和字节大小 (MRA.SZ[1:0] 位 =00b)；给 MRB 寄存器设定传送目标地址递增 (MRB.DM[1:0] 位 =10b)、1 次中断进行 1 次数据传送 (MRB.CHNE 位 =0 并且 MRB.DISEL 位 =0)，能给 MRB.DTS 位设定任意值；给 SAR 寄存器设定 SCIm.RDR 寄存器 (m=0、1、12) 的地址，给 DAR 寄存器设定用于保存数据的内部 RAM 起始地址，并且给 CRA 寄存器设定 128 (“0080h”)。CRB 寄存器可以是任意值。

(2) DTC 向量表的设定

将用于 RXI 中断的传送信息起始地址设定到 DTC 向量表。

(3) ICU 的设定和 DTC 模块的启动

将对应的 ICU.DTCERi.DTCE 位置 “1”，将 ICU.IERi.IENj 位和 DTCST.DTCST 位置 “1”。

(4) SCI 的设定

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) DTC 传送

在每次 SCI 的 1 字节数据接收结束时发生 RXI 中断，启动 DTC。通过 DTC 将接收数据从 SCIm.RDR 寄存器传送到内部 RAM，并且进行 DAR 寄存器的递增和 CRA 寄存器的递减。

(6) 中断处理

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。

18.6.2 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18.15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（内部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CPU请求中断）。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. 无限重复上述的步骤4和步骤5。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。

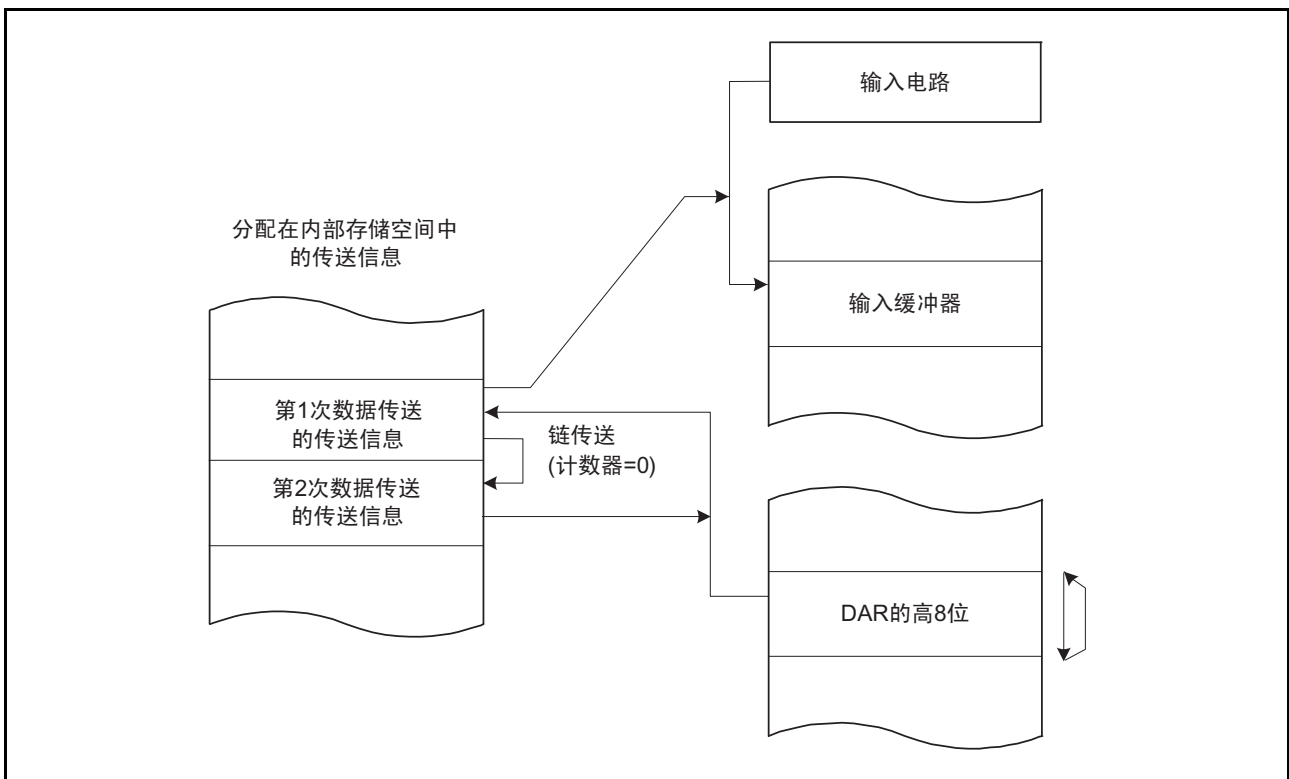


图 18.15 计数器为“0”时的链传送

18.7 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 PSW.I 位（中断允许位）、PSW.IPL[3:0] 位（处理器中断优先级）和中断控制器优先级的控制。

18.8 低功耗功能

在使用模块停止功能以及向全模块时钟停止模式、软件待机模式或者深度软件待机模式转移时，必须在给 DTCST.DTCST 位写“0”（DTC 模块停止）后分别进行以下的设定。

(1) 模块停止功能

通过给 MSTPCRA.MSTPA28 位写“1”（转移到模块停止状态），使 DTC 的模块停止功能有效。如果在给 MSTPCRA.MSTPA28 位写 1 时正在进行 DTC 传送，就在 DTC 传送结束后转移到模块停止状态。

不能在 MSTPCRA.MSTPA28 位为“1”时存取 DTC 的寄存器。

通过给 MSTPCRA.MSTPA28 位写“0”（解除模块停止状态），解除 DTC 模块停止状态。

(2) 全模块时钟停止模式

必须按照“11. 低功耗功能”的“11.5.2.1 向全模块时钟停止模式的转移”的步骤进行设定。

如果在执行 WAIT 指令时正在进行 DTC 传送，就在 DTC 传送结束后转移到全模块时钟停止模式。

在从全模块时钟停止模式返回后，通过给 MSTPCRA.MSTPA28 位写“0”，解除 DTC 的模块停止状态。

(3) 软件待机模式和深度软件待机模式

必须按照“11. 低功耗功能”的“11.5.3.1 向软件待机模式的转移”或者“11.5.4.1 向深度软件待机模式的转移”的步骤进行设定。

如果在执行 WAIT 指令时正在进行 DTC 传送，就在 DTC 传送结束后转移到软件待机模式或者深度软件待机模式。

(4) 低功耗功能的注意事项

有关 WAIT 指令和寄存器的设定步骤，请参照“11. 低功耗功能”的“11.6.6 WAIT 指令的执行时序”。

要在从低功耗模式返回后进行 DTC 传送时，必须再次给 DTCST.DTCST 位写“1”。

当全模块时钟停止模式期间或者软件待机模式期间发生的请求不是 DTC 启动请求而是 CPU 中断请求时，必须按照“14. 中断控制器 (ICUb)”的“14.5.3 中断请求目标的选择”的设定方法，在将中断请求源更改为 CPU 后执行 WAIT 指令。

18.9 使用时的注意事项

18.9.1 传送信息起始地址 / 传送源地址 / 传送目标地址

必须给向量表中指定的传送信息起始地址指定地址 $4n$ ，否则将地址的低 2 位作为“00b”进行存取。

18.9.2 传送信息的分配

如图 18.16 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 CRB 和 CRA 的设定数据分别写到低位地址 0 和低位地址 2。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 的设定数据分配到 32 位的 MSB 侧并且将 CRB 的设定数据分配到 LSB 侧后写到低位地址 0。

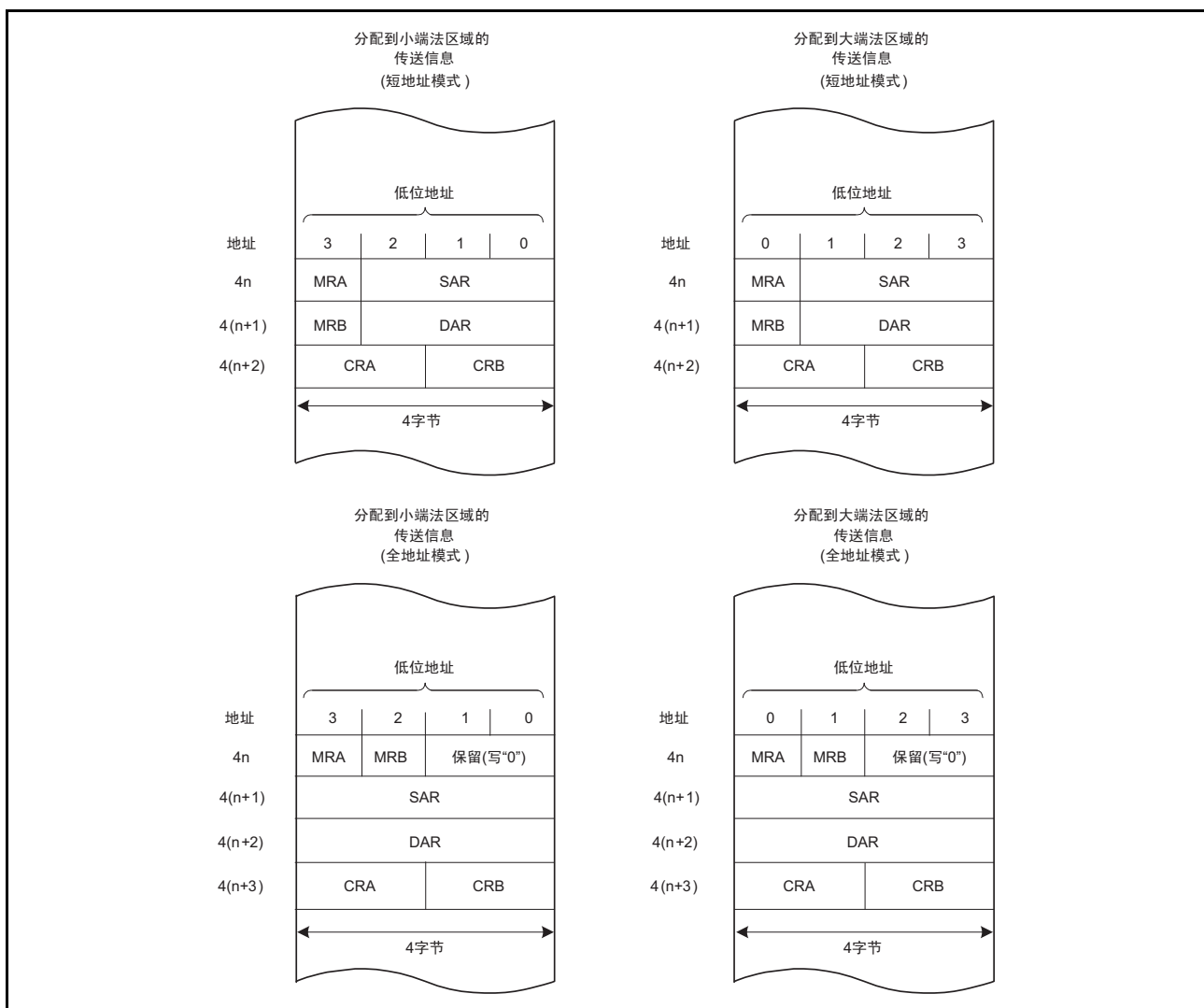


图 18.16 传送信息的分配

18.9.3 中断控制器的 DTC 启动允许寄存器 (ICU.DTCERn) 的设定

不能在将与通过 ICU.DTCERn 寄存器置“1”（允许 DTC 传送）的中断向量号相同的中断向量号设定给 DMAC 启动请求选择寄存器 (ICU.DMRSRn (n=DMAC 通道号)) 后启动 DMAC。有关 ICU.DTCERn 寄存器和 ICU.DMRSRn 寄存器的详细内容，请参照“14. 中断控制器 (ICUb)”。

19. I/O 端口

19.1 概要

RX63T 群的 I/O 端口用作可编程输入 / 输出端口、外围功能的输入 / 输出或者中断输入引脚。

各端口兼用外围模块的输入 / 输出引脚和中断输入引脚。在复位后立即变为输入端口，通过设定寄存器进行功能的切换。各端口的设定取决于 I/O 端口的寄存器设定和内部外围模块的寄存器设定。

各端口有指定输入 / 输出的端口方向寄存器（PDR）、保存输出数据的端口输出数据寄存器（PODR）、反映引脚状态的端口输入数据寄存器（PIDR）以及指定功能引脚的端口模式寄存器（PMR）。有关 PMR 寄存器的详细内容，请参照“20. 多功能引脚控制器（MPC）”。

I/O 端口的结构因封装而不同，I/O 端口的规格如表 19.1 所示，I/O 端口的功能一览如表 19.1 和表 19.2 所示。

表 19.1 I/O 端口的规格

端口符号	封装		封装	
	64 引脚	个数	48 引脚	个数
PORT0	P00、P01	2	无	0
PORT1	P10、P11	2	无	0
PORT2	P22 ~ P24	3	P22 ~ P24	3
PORT3	P30 ~ P33	4	P30	1
PORT4	P40 ~ P47	8	P40 ~ P44、P47	6
PORT7	P70 ~ P76	7	P70 ~ P76	7
PORT9	P91 ~ P94	4	无	0
PORTA	PA2 ~ PA5	4	PA2、PA3	2
PORTB	PB0 ~ PB7	8	PB0 ~ PB6	7
PORTD	PD3 ~ PD7	5	PD3 ~ PD7	5
PORTE	PE2	1	PE2	1
	端口总数	48	端口总数	32

表 19.2 I/O 端口的功能

端口符号	端口	输入上拉功能	漏极开路输出功能	5V 容许
PORT0	P00、P01	—	—	○
PORT1	P10、P11	—	—	○
PORT2	P22、P23	—	—	○
	P24	—	○	○
PORT3	P30	—	○	○
	P31 ~ P34	—	—	○
PORT4	P40 ~ P47	—	—	—
PORT7	P70 ~ P76	—	—	○
PORT9	P91、P92	—	—	○
	P93、P94	—	○	○
PORTA	PA2 ~ PA5	—	—	○
	PA4、PA5	—	—	○
PORTB	PB0、PB3、PB4、PB7	—	—	○
	PB1、PB2、PB5、PB6	—	○	○
PORTD	PD3、PD5	—	○	○
	PD4、PD6、PD7	—	—	○
PORTE	PE2	—	—	—

输入上拉功能、漏极开路输出功能和 5V 容许的设定也对通用输入 / 输出端口和引脚复用的其他信号有效。

19.2 输入 / 输出端口的结构

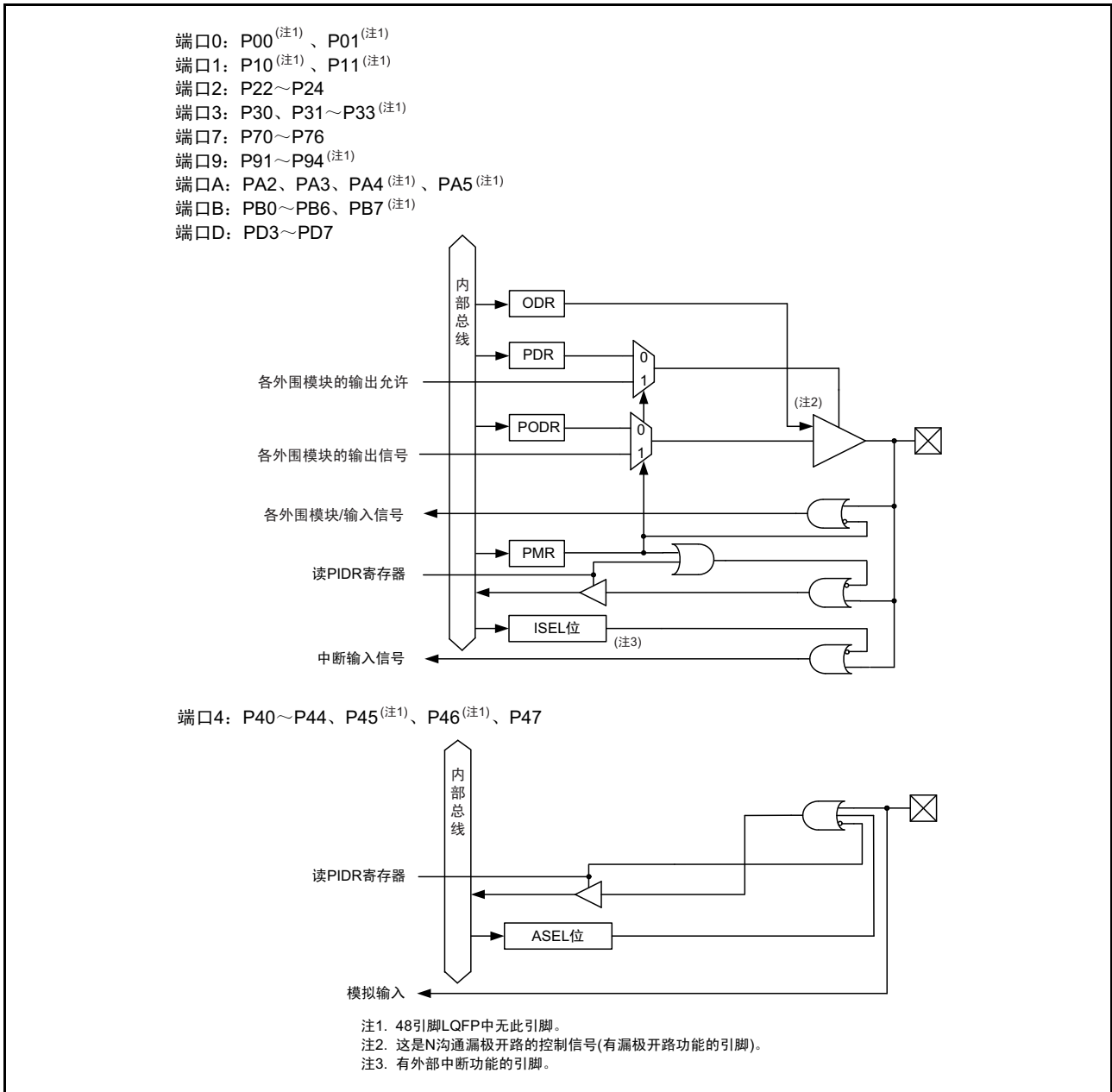


图 19.1 输入 / 输出端口的结构 (1)

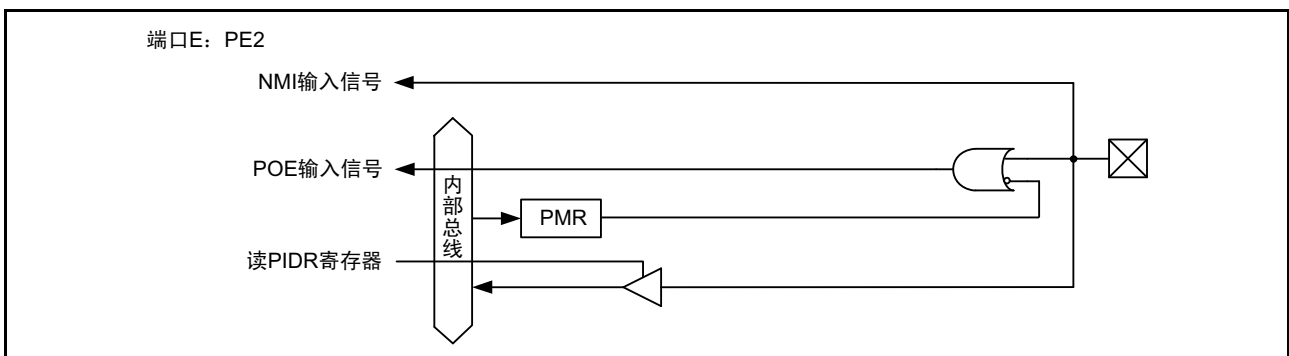


图 19.2 输入 / 输出端口的结构 (2)

19.3 寄存器说明

19.3.1 端口方向寄存器 (PDR)

地址 PORT0.PDR 0008 C000h、PORT1.PDR 0008 C001h、PORT2.PDR 0008 C002h、PORT3.PDR 0008 C003h、PORT7.PDR 0008 C007h、PORT9.PDR 0008 C009h、PORTA.PDR 0008 C00Ah、PORTB.PDR 0008 C00Bh、PORTD.PDR 0008 C00Dh

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 方向控制位	0: 输入 (用作输入端口) 1: 输出 (用作输出端口)	R/W
b1	B1	Pm1 方向控制位		R/W
b2	B2	Pm2 方向控制位		R/W
b3	B3	Pm3 方向控制位		R/W
b4	B4	Pm4 方向控制位		R/W
b5	B5	Pm5 方向控制位		R/W
b6	B6	Pm6 方向控制位		R/W
b7	B7	Pm7 方向控制位		R/W

m=0 ~ 3、7、9、A、B、D

PDR 寄存器是在选择了通用输入 / 输出端口功能时指定端口输入 / 输出的寄存器。

PORTm.PDR 寄存器的各位分别对应端口 m 的 1 个引脚，并且以 1 位为单位进行指定。

不存在的引脚的位为保留位，保留位的读写值都为“0”。

19.3.2 端口输出数据寄存器 (PODR)

地址 PORT0.PODR 0008 C020h、PORT1.PODR 0008 C021h、PORT2.PODR 0008 C022h、PORT3.PODR 0008 C023h、
PORT7.PODR 0008 C027h、PORT9.PODR 0008 C029h、PORTA.PODR 0008 C02Ah、PORTB.PODR 0008 C02Bh、
PORTD.PODR 0008 C02Dh

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

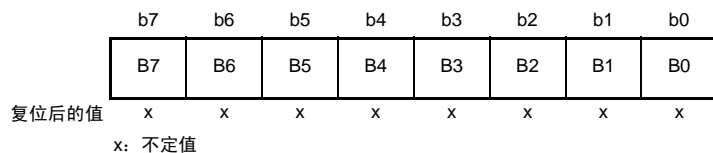
位	符号	位名	功能	R/W
b0	B0	Pm0 输出数据保存位	0: 输出 Low 电平 1: 输出 High 电平	R/W
b1	B1	Pm1 输出数据保存位		R/W
b2	B2	Pm2 输出数据保存位		R/W
b3	B3	Pm3 输出数据保存位		R/W
b4	B4	Pm4 输出数据保存位		R/W
b5	B5	Pm5 输出数据保存位		R/W
b6	B6	Pm6 输出数据保存位		R/W
b7	B7	Pm7 输出数据保存位		R/W

m=0 ~ 3、7、9、A、B、D

PODR 寄存器保存用作通用输出端口的引脚输出数据。
不存在的引脚的位为保留位，保留位的读写值都为“0”。

19.3.3 端口输入数据寄存器 (PIDR)

地址 PORT0.PIDR 0008 C040h、PORT1.PIDR 0008 C041h、PORT2.PIDR 0008 C042h、PORT3.PIDR 0008 C043h、PORT4.PIDR 0008 C044h、PORT7.PIDR 0008 C047h、PORT9.PIDR 0008 C049h、PORTA.PIDR 0008 C04Ah、PORTB.PIDR 0008 C04Bh、PORTD.PIDR 0008 C04Dh、PORTE.PIDR 0008 C04Eh



位	符号	位名	功能	R/W
b0	B0	Pm0 位	0: 输入 Low 电平 1: 输入 High 电平	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

m=0 ~ 4、7、9、A、B、D、E

PIDR 寄存器是反映端口引脚状态的寄存器。

如果读 PORTm.PIDR 寄存器，就能读到引脚状态，与 PORTm.PDR 寄存器和 PORTm.PMR 的值无关。

PE2 引脚能读 NMI 引脚的状态。

不存在的引脚的位为保留位，保留位的读取值为不定值，写操作无效。

19.3.4 端口模式寄存器 (PMR)

地址 PORT0.PMR 0008 C060h、PORT1.PMR 0008 C061h、PORT2.PMR 0008 C062h、PORT3.PMR 0008 C063h、
PORT7.PMR 0008 C067h、PORT9.PMR 0008 C069h、PORTA.PMR 0008 C06Ah、PORTB.PMR 0008 C06Bh、
PORTD.PMR 0008 C06Dh、PORTE.PMR 0008 C06Eh

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 引脚模式控制位	0: 用作通用输入 / 输出端口 1: 用作外围功能	R/W
b1	B1	Pm1 引脚模式控制位		R/W
b2	B2	Pm2 引脚模式控制位		R/W
b3	B3	Pm3 引脚模式控制位		R/W
b4	B4	Pm4 引脚模式控制位		R/W
b5	B5	Pm5 引脚模式控制位		R/W
b6	B6	Pm6 引脚模式控制位		R/W
b7	B7	Pm7 引脚模式控制位		R/W

m=0 ~ 3、7、9、A、B、D、E

PORTm.PMR 寄存器的各位分别对应端口 m 的 1 个引脚，并且能以 1 位为单位进行指定。
不存在的引脚的位为保留位，保留位的读写值都为“0”。

19.3.5 漏极开路控制寄存器 0 (ODR0)

地址 PORT3.ODR0 0008 C086h、PORT9.ODR0 0008 C092h、PORTB.ODR0 0008 C096h、PORTD.ODR0 0008 C09Ah

b7	b6	b5	b4	b3	b2	b1	b0
—	B6	—	B4	—	B2	—	B0
复位后的值	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	B0	P30 输出形态指定位	0: CMOS 输出 1: N 沟道漏极开路 (b1、b3、b5、b7: 保留位)	R/W
b1	—	保留位		R/W
b2	B2	PB1 输出形态指定位		R/W
b3	—	保留位		R/W
b4	B4	PB2 输出形态指定位		R/W
b5	—	保留位		R/W
b6	B6	Pm3 输出形态指定位 (m=9、D)		R/W
b7	—	保留位		R/W

ODR0 寄存器的奇数位 (b1、b3、b5、b7) 为保留位。
不存在的引脚的位为保留位，保留位的读写值都为“0”。

19.3.6 漏极开路控制寄存器 1 (ODR1)

地址 PORT2.ODR1 0008 C085h、PORT9.ODR1 0008 C093h、PORTB.ODR1 0008 C097h、PORTD.ODR1 0008 C09Bh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	B4	—	B2	—	B0
复位后的值	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	B0	Pm4 输出形态指定位 (m=2、9)	0: CMOS 输出 1: N 沟道漏极开路 (b1、b3、b5 ~ b7: 保留位)	R/W
b1	—	保留位		R/W
b2	B2	Pm5 输出形态指定位 (m=B、D)		R/W
b3	—	保留位		R/W
b4	B4	PB6 输出形态指定位		R/W
b7-b5	—	保留位		R/W

ODR1 寄存器的 b1、b3、b5 ~ b7 为保留位。
不存在的引脚的位为保留位，保留位的读写值都为“0”。

19.4 未使用引脚的处理

未使用引脚的处理内容如表 19.3 所示。

表 19.3 未使用引脚的处理内容

引脚名	处理内容
EMLE	通过电阻连接 VSS (下拉)。
MD	(用作模式引脚)
RES#	通过电阻连接 VCC (上拉)。
PE2/NMI	通过电阻连接 VCC (上拉)。
EXTAL	通过电阻连接 VSS (下拉)。
XTAL	将引脚置为开路。
端口 0 ~ 3、7、9、A、B、D	每个引脚通过电阻连接 VCC (上拉) 或者 VSS (下拉)。
端口 4	每个引脚通过电阻连接 AVCC0 (上拉) 或者 AVSS0 (下拉)。
VREFH0	连接 AVCC0。
VREFL0	连接 AVSS0。

20. 多功能引脚控制器（MPC）

20.1 概要

多功能引脚控制器（MPC）具有从多个端口中选择并且分配外围功能输入/输出和中断输入信号的功能。

多路复用引脚的分配引脚一览表如表 20.1 所示。表中用“○”和“—”表示不同封装的引脚的有无。禁止通过多个引脚将同一功能设定为有效。

表 20.1 多路复用引脚的分配引脚一览表 (1/3)

模块 / 功能	通道	引脚功能	分配端口	封装	
				64 引脚	48 引脚
中断	IRQ0	IRQ0-DS（输入）	P10	○	—
		IRQ0（输入）	PB5	○	○
	IRQ1	IRQ1-DS（输入）	P11	○	—
		IRQ1（输入）	P93	○	—
	IRQ2	IRQ2-DS（输入）	P00	○	—
	IRQ3	IRQ3-DS（输入）	PB4	○	○
	IRQ4	IRQ4-DS（输入）	P01	○	—
IRQ5	IRQ5-DS（输入）	P70	○	○	
多功能定时器单元 3	MTU0	MTIOC0A（输入 / 输出）	P31	○	—
			PB3	○	○
		MTIOC0B（输入 / 输出）	P30	○	○
			PB2	○	○
	MTIOC0C（输入 / 输出）	PB1	○	○	
	MTIOC0D（输入 / 输出）	PB0	○	○	
	MTU1	MTIOC1A（输入 / 输出）	PA5	○	—
		MTIOC1B（输入 / 输出）	PA4	○	—
	MTU2	MTIOC2A（输入 / 输出）	PA3	○	○
		MTIOC2B（输入 / 输出）	PA2	○	○
	MTU3	MTIOC3A（输入 / 输出）	P33	○	—
		MTIOC3B（输入 / 输出）	P71	○	○
		MTIOC3C（输入 / 输出）	P32	○	—
		MTIOC3D（输入 / 输出）	P74	○	○
	MTU4	MTIOC4A（输入 / 输出）	P72	○	○
		MTIOC4B（输入 / 输出）	P73	○	○
		MTIOC4C（输入 / 输出）	P75	○	○
		MTIOC4D（输入 / 输出）	P76	○	○
	MTU5	MTIC5U（输入）	P24	○	○
		MTIC5V（输入）	P23	○	○
		MTIC5W（输入）	P22	○	○
	MTU6	MTIOC6A（输入 / 输出）	P33	○	—
		MTIOC6B（输入 / 输出）	P71	○	○
		MTIOC6C（输入 / 输出）	P32	○	—
		MTIOC6D（输入 / 输出）	P74	○	○
	MTU7	MTIOC7A（输入 / 输出）	P72	○	○
		MTIOC7B（输入 / 输出）	P73	○	○
		MTIOC7C（输入 / 输出）	P75	○	○
MTIOC7D（输入 / 输出）		P76	○	○	

表 20.1 多路复用引脚的分配引脚一览表 (2/3)

模块 / 功能	通道	引脚功能	分配端口	封装	
				64 引脚	48 引脚
多功能定时器单元 3	MTU	MTCLKA (输入)	P22	○	○
			PB3	○	○
		MTCLKB (输入)	P23	○	○
			PB2	○	○
		MTCLKC (输入)	P11	○	—
			P24	○	○
		MTCLKD (输入)	P10	○	—
P30	○		○		
端口输出允许 3	POE0	POE0# (输入)	P70	○	○
	POE8	POE8# (输入)	PB4	○	○
	POE10	POE10# (输入)	PE2	○	○
	POE11	POE11# (输入)	PB5	○	○
通用 PWM 定时器	GPT0	GTIOC0A (输入 / 输出)	P71	○	○
			PD7	○	○
		GTIOC0B (输入 / 输出)	P74	○	○
			PD6	○	○
	GPT1	GTIOC1A (输入 / 输出)	P72	○	○
			PD5	○	○
		GTIOC1B (输入 / 输出)	P75	○	○
	PD4		○	○	
	GPT2	GTIOC2A (输入 / 输出)	P73	○	○
			PD3	○	○
		GTIOC2B (输入 / 输出)	P76	○	○
			PB7	○	—
	PB6	○	○		
	GPT3	GTIOC3A (输入 / 输出)	P00	○	—
GTIOC3B (输入 / 输出)		P01	○	—	
GPT	GTETRG	PB4	○	○	
串行通信接口	SCI0	RXD0 (输入) / SMISO0 (输入 / 输出) / SSCL0 (输入 / 输出)	PB1	○	○
			P24	○	○
		TXD0 (输出) / SMOSI0 (输入 / 输出) / SSDA0 (输入 / 输出)	PB2	○	○
			P30	○	○
		SCK0 (输入 / 输出)	PB3	○	○
			P23	○	○
		CTS0# (输入) / RTS0# (输出) / SS0# (输入)	PD7	○	○
			P22	○	○
	P00	○	—		
	SCI1	RXD1 (输入) / SMISO1 (输入 / 输出) / SSCL1 (输入 / 输出)	PD5	○	○
			P93	○	—
		TXD1 (输出) / SMOSI1 (输入 / 输出) / SSDA1 (输入 / 输出)	PD3	○	○
			P94	○	—
		SCK1 (输入 / 输出)	PD4	○	○
P92			○	—	
CTS1# (输入) / RTS1# (输出) / SS1# (输入)	P70	○	○		
P91	○	—			

表 20.1 多路复用引脚的分配引脚一览表 (3/3)

模块 / 功能	通道	引脚功能	分配端口	封装	
				64 引脚	48 引脚
串行通信接口	SCI12	RXD12 (输入) / SMISO12 (输入 / 输出) / SSCL12 (输入 / 输出) / RXDX12 (输入)	PB6	○	○
		TXD12 (输出) / SMOSI12 (输入 / 输出) / SSDA12 (输入 / 输出) / TXDX12 (输出) / SIOX12 (输入 / 输出)	PB5	○	○
		SCK12 (输入 / 输出)	PB7	○	—
		CTS12# (输入) / RTS12# (输出) / SS12# (输入)	PB4	○	○
I ² C 总线接口		SCL (输入 / 输出)	PB1	○	○
		SDA (输入 / 输出)	PB2	○	○
串行外围接口		RSPCKA (输入 / 输出)	P24	○	○
			PA4	○	—
		MOSIA (输入 / 输出)	P23	○	○
			PB0	○	○
		MISOA (输入 / 输出)	P22	○	○
			PA5	○	—
		SSLA0 (输入 / 输出)	P30	○	○
			PA3	○	○
		SSLA1 (输出)	P31	○	—
			PA2	○	○
SSLA2 (输出)	P32	○	—		
SSLA3 (输出)	P33	○	—		
12 位 A/D 转换器		AN000 (输入)	P40	○	○
		AN001 (输入)	P41	○	○
		AN002 (输入)	P42	○	○
		AN003 (输入) / CVREFL (输入)	P43	○	○
		AN004 (输入)	P44	○	○
		AN005 (输入)	P45	○	—
		AN006 (输入)	P46	○	—
		AN007 (输入) / CVREFH (输入)	P47	○	○
		ADTRG0# (输入)	PA4	○	—
时钟频率精度测量电路	CACREF	P01	○	—	
		P23	○	○	
		PB3	○	○	

20.2 寄存器说明

根据不同的封装，保留没有引脚的寄存器和位。要给对应的位写值时，必须写复位后的值。

20.2.1 写保护寄存器 (PWPR)

地址 0008 C11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	B0WI	PFSWE	—	—	—	—	—	—
复位后的值	1	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	PFSWE	PFS 寄存器的写允许位	0: 禁止写 PFS 寄存器 1: 允许写 PFS 寄存器	R/W
b7	B0WI	PFSWE 位的写禁止位	0: 允许写 PFSWE 位 1: 禁止写 PFSWE 位	R/W

PFSWE 位 (PFS 寄存器的写允许位)

只有在 PFSWE 位为“1”时，允许写 PmnPFS 寄存器。

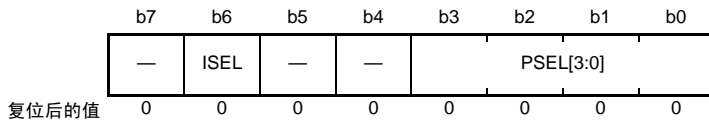
要将 PFSWE 位置“1”时，必须在给 B0WI 位写“0”后将 PFSWE 位置“1”。

B0WI 位 (PFSWE 位的写禁止位)

只有在 B0WI 位为“0”时，允许写 PFSWE 位。

20.2.2 P0n 引脚功能控制寄存器 (P0nPFS) (n=0、1)

地址 P00PFS 0008 C140h、P01PFS 0008 C141h



位	符号	位名	功能	R/W
b3-b0	PSEL[3:0]	引脚功能选择位	选择外围功能。有关各引脚功能，请参照表 20.2。	R/W
b5-b4	—	保留位	读写值都为“0”。	R/W
b6	ISEL	中断输入功能选择位	0: 不用作 IRQn 输入引脚 1: 用作 IRQn 输入引脚	R/W
b7	ASEL	模拟输入功能选择位	0: 不用作模拟引脚 1: 用作模拟引脚	R/W

Pmi 引脚功能控制寄存器 (PmiPFS) 选择引脚功能。通过 PSEL[3:0] 位设定分配给引脚的外围功能。

在用作 IRQ 输入引脚时设定 ISEL 位，也能和外围功能组合使用。但是，禁止通过 2 个或者 2 个以上的引脚来允许同一序号的 IRQn (外部引脚中断)。

PmiPFS 寄存器受写保护寄存器 (PWPR) 的保护。要改写时，必须先解除保护。

没有 IRQn 功能的引脚的 ISEL 位为保留位，没有模拟输入 / 输出功能的引脚的 ASEL 位为保留位。

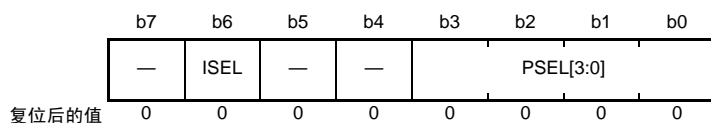
表 20.2 64 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚	
	P00	P01
0000b (初始值)	Hi-Z	
0101b	—	CACREF
0110b	GTIOC3A	GTIOC3B
1010b	GTS0# RTS0# SS0#	—

—: 不能设定。

20.2.3 P1n 引脚功能控制寄存器 (P1nPFS) (n=0、1)

地址 P10PFS 0008 C148h、P11PFS 0008 C149h



位	符号	位名	功能	R/W
b3-b0	PSEL[3:0]	引脚功能选择位	选择外围功能。有关各引脚功能，请参照表 20.3。	R/W
b5-b4	—	保留位	读写值都为“0”。	R/W
b6	ISEL	中断输入功能选择位	0: 不用作 IRQn 输入引脚 1: 用作 IRQn 输入引脚	R/W
b7	—	保留位	读写值都为“0”。	R/W

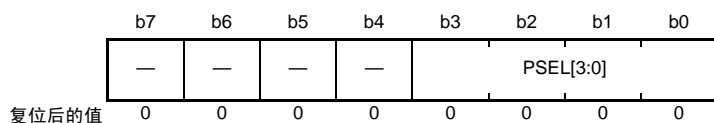
表 20.3 64 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚	
	P10	P11
0000b (初始值)	Hi-Z	
0001b	MDCLKD	MTCLKC

—: 不能设定。

20.2.4 P2n 引脚功能控制寄存器 (P2nPFS) (n=2 ~ 4)

地址 P22PFS 0008 C152h、P23PFS 0008 C153h、P24PFS 0008 C154h



位	符号	位名	功能	R/W
b3-b0	PSEL[3:0]	引脚功能选择位	选择外围功能。有关各引脚功能，请参照表 20.4。	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

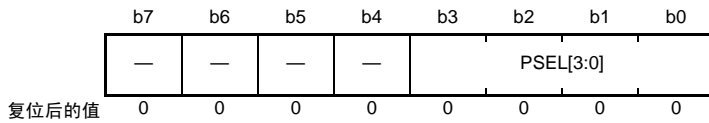
表 20.4 64 引脚 LQFP 和 48 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚		
	P22	P23	P24
0000b (初始值)	Hi-Z		
0001b	MTIC5W	MTIC5V	MTIC5U
0010b	MTCLKA	MTCLKB	MTCLKC
0101b	—	CACREF	—
1010b	CTS0# RTS0# SS0#	SCK0	RXD0 SMISO SSCLO
1101b	MISOA	MOSIA	RSPCKA

—: 不能设定。

20.2.5 P3n 引脚功能控制寄存器（P3nPFS）（n=0 ~ 3）

地址 P30PFS 0008 C158h、P31PFS 0008 C159h、P32PFS 0008 C15Ah、P33PFS 0008 C15Bh



位	符号	位名	功能	R/W
b3-b0	PSEL[3:0]	引脚功能选择位	选择外围功能。有关各引脚功能，请参照表 20.5、表 20.6。	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

表 20.5 64 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚			
	P30	P31	P32	P33
0000b（初始值）	Hi-Z			
0001b	MTIOC0B	MTIOC0A	MTIOC3C	MTIOC3A
0010b	MTCLKD	—	MTIOC6C	MTIOC6A
1010b	TXD0 SMOSI0 SSDA0	—	—	—
1101b	SSLA0	SSLA1	SSLA2	SSLA3

—：不能设定。

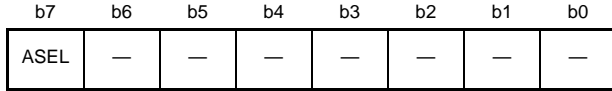
表 20.6 48 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚			
	P30	—	—	—
0000b（初始值）	Hi-Z			
0001b	MTIOC0B	—	—	—
0010b	MTCLKD	—	—	—
1010b	TXD0 SMOSI0 SSDA0	—	—	—
1101b	SSLA0	—	—	—

—：不能设定。

20.2.6 P4n 引脚功能控制寄存器 (P4nPFS) (n=0 ~ 7)

地址 P40PFS 0008 C160h、P41PFS 0008 C161h、P42PFS 0008 C162h、P43PFS 0008 C163h、P44PFS 0008 C164h、P45PFS 0008 C165h、P46PFS 0008 C166h、P47PFS 0008 C167h



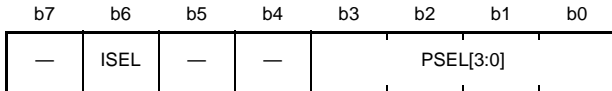
复位后的值 0 0 0 0 0 0 0 0

位	符号	位名	功能	R/W
b6-b0	—	保留位	读写值都为“0”。	R/W
b7	ASEL	模拟输入功能选择位	0: 不用作模拟引脚 1: 用作模拟引脚	R/W

在将引脚用作模拟引脚时设定 ASEL 位。此时，不能读引脚状态。

20.2.7 P7n 引脚功能控制寄存器 (P7nPFS) (n=0 ~ 6)

地址 P70PFS 0008 C178h、P71PFS 0008 C179h、P72PFS 0008 C17Ah、P73PFS 0008 C17Bh、P74PFS 0008 C17Ch、P75PFS 0008 C17Dh、P76PFS 0008 C17Eh



复位后的值 0 0 0 0 0 0 0 0

位	符号	位名	功能	R/W
b3-b0	PSEL[3:0]	引脚功能选择位	选择外围功能。有关各引脚功能，请参照表 20.7。	R/W
b5-b4	—	保留位	读写值都为“0”。	R/W
b6	ISEL	中断输入功能选择位	0: 不用作 IRQn 输入引脚 1: 用作 IRQn 输入引脚	R/W
b7	—	保留位	读写值都为“0”。	R/W

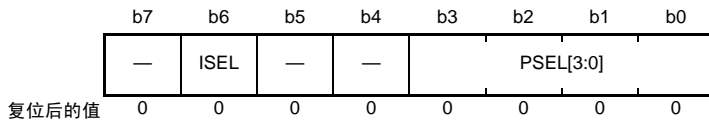
表 20.7 64 引脚 LQFP 和 48 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚						
	P70	P71	P72	P73	P74	P75	P76
0000b (初始值)	Hi-Z						
0001b	—	MTIOC3B	MTIOC4A	MTIOC4B	MTIOC3D	MTIOC4C	MTIOC4D
0010b	—	MTIOC6B	MTIOC7A	MTIOC7B	MTIOC6D	MTIOC7C	MTIOC7D
0110b	—	GTIOC0A	GTIOC1A	GTIOC2A	GTIOC0B	GTIOC1B	GTIOC2B
0111b	POE0#	—	—	—	—	—	—
1010b	CTS1# RTS1# SS1#	—	—	—	—	—	—

—: 不能设定。

20.2.8 P9n 引脚功能控制寄存器 (P9nPFS) (n=1 ~ 4)

地址 P91PFS 0008 C189h、P92PFS 0008 C18Ah、P93PFS 0008 C18Bh、P94PFS 0008 C18Ch



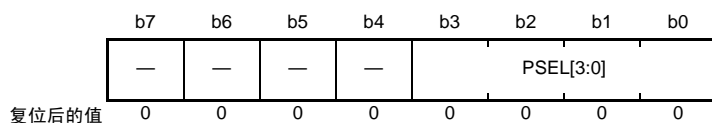
位	符号	位名	功能	R/W
b3-b0	PSEL[3:0]	引脚功能选择位	选择外围功能。有关各引脚功能，请参照表 20.8。	R/W
b5-b4	—	保留位	读写值都为“0”。	R/W
b6	ISEL	中断输入功能选择位	0: 不用作 IRQn 输入引脚 1: 用作 IRQn 输入引脚	R/W
b7	ASEL	模拟输入功能选择位	0: 不用作模拟引脚 1: 用作模拟引脚	R/W

表 20.8 64 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚			
	P91	P92	P93	P94
0000b (初始值)	Hi-Z			
1010b	CTS1# RTS1# SS1#	SCK1	RXD1 SMISO1 SSCL1	TXD1 SMOS1 SSDA1

20.2.9 PAn 引脚功能控制寄存器 (PAnPFS) (n=2 ~ 5)

地址 PA2PFS 0008 C192h、PA3PFS 0008 C193h、PA4PFS 0008 C194h、PA5PFS 0008 C195h



位	符号	位名	功能	R/W
b3-b0	PSEL[3:0]	引脚功能选择位	选择外围功能。有关各引脚功能，请参照表 20.9、表 20.10。	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

表 20.9 64 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚			
	PA2	PA3	PA4	PA5
0000b (初始值)	Hi-Z			
0001b	MTIOC2B	MTIOC2A	MTIOC1B	MTIOC1A
1001b	—	—	ADTRG0#	—
1101b	SSLA1	SSLA0	RSPCKA	MISOA

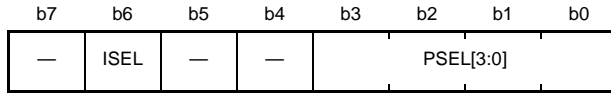
—: 不能设定。

表 20.10 48 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚	
	PA2	PA3
0000b (初始值)	Hi-Z	
0001b	MTIOC2B	MTIOC2A
1101b	SSLA1	SSLA0

20.2.10 PBn 引脚功能控制寄存器 (PBnPFS) (n=0 ~ 7)

地址 PB0PFS 0008 C198h、PB1PFS 0008 C199h、PB2PFS 0008 C19Ah、PB3PFS 0008 C19Bh、PB4PFS 0008 C19Ch、
PB5PFS 0008 C19Dh、PB6PFS 0008 C19Eh、PB7PFS 0008 C19Fh



复位后的值 0 0 0 0 0 0 0 0

位	符号	位名	功能	R/W
b3-b0	PSEL[3:0]	引脚功能选择位	选择外围功能。有关各引脚功能，请参照表 20.11、表 20.12。	R/W
b5-b4	—	保留位	读写值都为“0”。	R/W
b6	ISEL	中断输入功能选择位	0: 不用作 IRQn 输入引脚 1: 用作 IRQn 输入引脚	R/W
b7	—	保留位	读写值都为“0”。	R/W

表 20.11 64 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚							
	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7
0000b (初始值)	Hi-Z							
0001b	MTIOC0D	MTIOC0C	MTIOC0B	MTIOC0A	—	—	—	—
0010b	—	—	MTCLKB	MTCLKA	—	—	—	—
0101b	—	—	—	CACREF	—	—	—	—
0110b	—	—	—	—	GTETRG	—	GTIOC2B	GTIOC2B
0111b	—	—	—	—	POE8#	POE11#	—	—
1010b	—	RXD0 SMISO0 SSCL0	TXD0 SMOSI0 SSDA0	SCK0	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	SCK12
1100b	—	—	—	—	CTS12# RTS12# SS12#	—	—	—
1101b	MOSIA	—	—	—	—	—	—	—
1111b	—	SCL	SDA	—	—	—	—	—

—: 不能设定。

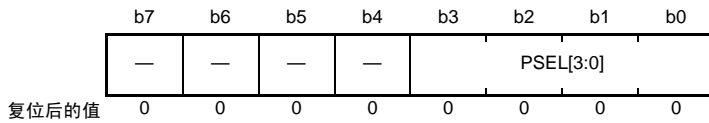
表 20.12 48 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚							
	PB0	PB1	PB2	PB3	PB4	PB5	PB6	—
0000b (初始值)	Hi-Z							
0001b	MTIOC0D	MTIOC0C	MTIOC0B	MTIOC0A	—	—	—	—
0010b	—	—	MTCLKB	MTCLKA	—	—	—	—
0101b	—	—	—	CACREF	—	—	—	—
0110b	—	—	—	—	GTETRG	—	GTIOC2B	—
0111b	—	—	—	—	POE8#	POE11#	—	—
1010b	—	RXD0 SMISO0 SSCL0	TXD0 SMOSIO SSDA0	SCK0	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	SCK12
1100b	—	—	—	—	CTS12# RTS12# SS12#	—	—	—
1101b	MOSIA	—	—	—	—	—	—	—
1111b	—	SCL	SDA	—	—	—	—	—

—: 不能设定。

20.2.11 PDn 引脚功能控制寄存器 (PDnPFS) (n=3 ~ 7)

地址 PD3PFS 0008 C1ABh、PD4PFS 0008 C1ACh、PD5PFS 0008 C1ADh、PD6PFS 0008 C1AEh、PD7PFS 0008 C1AFh



位	符号	位名	功能	R/W
b3-b0	PSEL[3:0]	引脚功能选择位	选择外围功能。有关各引脚功能, 请参照表 20.13。	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

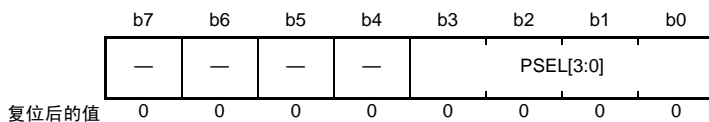
表 20.13 64 引脚 LQFP、48 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚				
	PD3	PD4	PD5	PD6	PD7
0000b (初始值)	Hi-Z				
0110b	GTIOC2A	GTIOC1B	GTIOC1A	GTIOC0B	GTIOC0A
1010b	TXD1 SMOSI1 SSDA1	SCK1	RXD1 SMISO1 SSCL1	—	CTS0# RTS0# SS0#

—: 不能设定。

20.2.12 PE2 引脚功能控制寄存器 (PE2PFS)

地址 PE2PFS 0008 C1B2h



位	符号	位名	功能	R/W
b3-b0	PSEL[3:0]	引脚功能选择位	选择外围功能。有关各引脚功能, 请参照表 20.14。	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

表 20.14 64 引脚 LQFP 和 48 引脚 LQFP 的引脚输入 / 输出功能寄存器的设定

PSEL[3:0] 位的 设定值	引脚
	PE2
0000b (初始值)	Hi-Z
0111b	POE10#

如果将 NMIER.NMIEN 位置“1”, 就优先 NMI 处理。

20.3 使用时的注意事项

20.3.1 引脚输入 / 输出功能的设定步骤

必须按照以下步骤设定引脚输入 / 输出功能。

1. 通过将端口模式寄存器 (PMR) 置“0”，设定为通用输入功能。
2. 给外围功能模块的对应引脚设定分配的输入/输出信号。
3. 通过将写保护寄存器 (PWPR) 的PWPR.B0WI位置“0”后将PWPR.PFSWE位置“1”，使Pmn引脚功能控制寄存器 (PmnPFS) 的写操作有效 (m=0~4、7、9、A、B、D、n=0~7)。
4. 通过PmnPFS.PSEL[3:0]位设定引脚输入/输出功能。
5. 通过将PWPR.PFSWE位置“0”后将PWPR.B0WI位置“1”，禁止写PmnPFS寄存器。
6. 根据需要，将PMR寄存器置“1”，切换到所选择的引脚输入/输出功能。

20.3.2 设定 MPC 寄存器时的注意事项

1. 必须在对应引脚的PMR寄存器为“0”的状态下设定Pmn引脚功能控制寄存器 (PmnPFS)。如果在对应引脚的PMR寄存器为“1”的状态下设定PmnPFS寄存器，就可能在输入功能的情况下输入意外的边沿，可能在输出功能的情况下输出意外的脉冲。
2. 必须设定PmnPFS寄存器能设定的功能。如果设定没有指定的功能，就不保证运行。
3. 不能设定为通过MPC给多个引脚分配相同的功能。

表 20.15 寄存器的设定

项目	PMR.Bn	PDR.Bn	PmnPFS			注意事项
			ASEL	ISEL	PSEL[3:0]	
解除复位后	0	0	0	0	00000b	在解除复位后，用作通用输入端口。
通用输入端口	0	0	0	0/1	x	当与中断输入并用时，必须将PmnPFS.ISEL位置“1”。
通用输出端口	0	1	x	0	x	
外围功能	1	x	x	0/1	外围功能 (参照表 20.2 ~ 表 20.14)	当与中断输入并用时，必须将PmnPFS.ISEL位置“1”。
中断输入	0	0	x	1	x	
NMI	x	x	x	x (注1)	x	不需要设定寄存器。
模拟输入	0	0	1	x (注1)	x	为了将输出缓冲器置为OFF，必须设定为通用输入端口。
EXTAL/XTAL	0	0	x	x	x	为了将输出缓冲器置为OFF，必须设定为通用输入端口。

x: 不需要设定。

0/1: 如果将PmnPFS.ISEL位置“0”，就不用作IRQ引脚。

如果将PmnPFS.ISEL位置“1”，就用作IRQ引脚 (IRQ被多路复用的情况)

注 1. 即使将PmnPFS.ISEL位置“1”，也不用作IRQn输入引脚。

- 注.
- 能在P4nPFS.ASEL位为“0”时读引脚状态。
 - 必须在PMR.Bn位为“0”的状态下更改PmnPFS.PSEL[3:0]位。

20.3.3 使用模拟功能时的注意事项

在使用模拟功能时，必须将Pmn引脚功能控制寄存器的引脚功能选择位 (PmnPFS.ASEL[1:0]) 置“1”。

21. 多功能定时器脉冲单元 3 (MTU3)

21.1 概要

RX63T 群内置由 8 个通道的 16 位定时器构成的多功能定时器脉冲单元 3 (MTU3)。

MTU 的规格如表 21.1 所示, MTU 的功能一览表如表 21.2 所示, MTU 的框图如图 21.1 和图 21.2 所示。

表 21.1 MTU 的规格

项目	内容
脉冲输入 / 输出	最多 24 个 (最多能同时使用 16 个) (注 1)
脉冲输入	3 个
计数时钟	各通道有 6 ~ 8 种 (通道 5 有 4 种)。
工作频率	8 ~ 100MHz
能设定的运行	<p>【MTU0 ~ 4、6、7】</p> <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)。 通过比较匹配或者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 8 相的 PWM 输出。(注 1) <p>【MTU0、3、4、6、7】</p> <ul style="list-style-type: none"> 能设定缓冲运行。 <p>【MTU3、4、6、7】</p> <ul style="list-style-type: none"> 能通过 MTU3/4 以及 MTU6/7 的联动运行输出互补 PWM 或者复位 PWM 的正负 3 相共 6 相波形。 能在互补 PWM 模式中的定时器计数器的波峰 / 波谷或者写缓冲寄存器 (MTU4.TGRD、MTU7.TGRD) 时, 将数据从缓冲寄存器传送到暂存器。 能在互补 PWM 模式中设定双缓冲功能。 <p>【MTU3、4】</p> <ul style="list-style-type: none"> 能通过 MTU0 的联动运行, 设定使用互补 PWM 或者复位 PWM 的 AC 同步马达 (无刷 DC 马达) 驱动模式, 并且选择 2 种 (斩波、电平) 波形的输出。 <p>【MTU1、2】</p> <ul style="list-style-type: none"> 能分别设定相位计数模式。 能级联运行。 <p>【MTU5】</p> <ul style="list-style-type: none"> 能用作补偿死区时间的计数器。
中断减少功能	能在互补 PWM 模式中, 减少计数器波峰 / 波谷的中断以及 A/D 转换器的转换开始触发。
中断源	38 种
缓冲运行	寄存器数据的自动传送 (从缓冲寄存器传送到定时器的寄存器)
触发生成	<p>能生成 A/D 转换器的转换开始触发。</p> <p>能通过 A/D 转换开始请求的延迟功能, 在任意的时序开始 A/D 转换。而且能与 PWM 输出同步运行。</p>
低功耗功能	能设定为模块停止状态。

注 1. MTIOC3A 和 MTIOC6A、MTIOC3B 和 MTIOC6B、MTIOC3C 和 MTIOC6C 以及 MTIOC3D 和 MTIOC6D 分别复用相同引脚, 因此能通过 MPC 选择其中一个使用。同样, MTIOC4A 和 MTIOC7A、MTIOC4B 和 MTIOC7B、MTIOC4C 和 MTIOC7C 以及 MTIOC4D 和 MTIOC7D 也分别复用相同引脚, 因此也能通过 MPC 选择其中一个使用。

表 21.2 MTU 的功能一览表 (1/2)

项目	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5	MTU6	MTU7
计数时钟	PCLKA/1 PCLKA/4 PCLKA/16 PCLKA/64 MTCLKA MTCLKB MTCLKC MTCLKD	PCLKA/1 PCLKA/4 PCLKA/16 PCLKA/64 PCLKA/256 MTCLKA MTCLKB	PCLKA/1 PCLKA/4 PCLKA/16 PCLKA/64 PCLKA/1024 MTCLKA MTCLKB MTCLKC	PCLKA/1 PCLKA/4 PCLKA/16 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/4 PCLKA/16 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/4 PCLKA/16 PCLKA/64	PCLKA/1 PCLKA/4 PCLKA/16 PCLKA/64 PCLKA/256 PCLKA/1024	PCLKA/1 PCLKA/4 PCLKA/16 PCLKA/64 PCLKA/256 PCLKA/1024
通用寄存器 (TGR)	TGRA TGRB TGRE	TGRA TGRB	TGRA TGRB	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW	TGRA TGRB	TGRA TGRB
通用寄存器 / 缓冲寄存器	TGRC TGRD TGRF	—	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF
输入 / 输出引脚	MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	输入引脚 MTIC5U MTIC5V MTIC5W	MTIOC6A MTIOC6B MTIOC6C MTIOC6D	MTIOC7A MTIOC7B MTIOC7C MTIOC7D
计数器清除功能	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉
比较匹配 输出	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 1	○	○	○	○	○	—	○	○
PWM 模式 2	○	○	○	—	—	—	—	—
互补 PWM 模式	—	—	—	○	○	—	○	○
复位同步 PWM	—	—	—	○	○	—	○	○
AC 同步马达驱动模式	○	—	—	○	○	—	—	—
相位计数模式	—	○	○	—	—	—	—	—
缓冲运行	○	—	—	○	○	—	○	○
用于补偿死区时间的 计数器功能	—	—	—	—	—	○	—	—
中断源	7 种中断源 • 比较匹配 / 输入捕捉 A • 比较匹配 / 输入捕捉 B • 比较匹配 / 输入捕捉 C • 比较匹配 / 输入捕捉 D • 比较匹配 E • 比较匹配 F • 上溢	4 种中断源 • 比较匹配 / 输入捕捉 A • 比较匹配 / 输入捕捉 B • 上溢 • 下溢	4 种中断源 • 比较匹配 / 输入捕捉 A • 比较匹配 / 输入捕捉 B • 上溢 • 下溢	5 种中断源 • 比较匹配 / 输入捕捉 A • 比较匹配 / 输入捕捉 B • 比较匹配 / 输入捕捉 C • 比较匹配 / 输入捕捉 D • 上溢	5 种中断源 • 比较匹配 / 输入捕捉 A • 比较匹配 / 输入捕捉 B • 比较匹配 / 输入捕捉 C • 比较匹配 / 输入捕捉 D • 上溢、下溢 (仅互补 PWM 模 式)	3 种中断源 • 比较匹配 / 输入捕捉 U • 比较匹配 / 输入捕捉 V • 比较匹配 / 输入捕捉 W	5 种中断源 • 比较匹配 / 输入捕捉 A • 比较匹配 / 输入捕捉 B • 比较匹配 / 输入捕捉 C • 比较匹配 / 输入捕捉 D • 上溢	5 种中断源 • 比较匹配 / 输入捕捉 A • 比较匹配 / 输入捕捉 B • 比较匹配 / 输入捕捉 C • 比较匹配 / 输入捕捉 D • 上溢、下溢 (仅互补 PWM 模 式)
DTC 的启动	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉、上溢、 下溢 (仅互补 PWM 模式)	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉	TGR 的比较 匹配或者输入 捕捉、上溢、 下溢 (仅互补 PWM 模式)

表 21.2 MTU 的功能一览表 (2/2)

项目	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5	MTU6	MTU7
A/D 转换开始触发	TGRA 的比较匹配或者输入捕捉、TGRE 的比较匹配	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、互补 PWM 模式中的 TCNT 下溢 (波谷)	—	TGRA 的比较匹配或者输入捕捉	TGRA 的比较匹配或者输入捕捉、互补 PWM 模式中的 TCNT 下溢 (波谷)
A/D 转换开始请求的延迟功能	—	—	—	—	<ul style="list-style-type: none"> 在 TADCORA 和 TCNT 匹配时请求开始 A/D 转换 在 TADCORB 和 TCNT 匹配时请求开始 A/D 转换 	—	—	<ul style="list-style-type: none"> 在 TADCORA 和 TCNT 匹配时请求开始 A/D 转换 在 TADCORB 和 TCNT 匹配时请求开始 A/D 转换
中断减少功能 1	—	—	—	<ul style="list-style-type: none"> 减少 TGRA 的比较匹配中断 	<ul style="list-style-type: none"> 减少 TCIV 中断 	—	<ul style="list-style-type: none"> 减少 TGRA 的比较匹配中断 	<ul style="list-style-type: none"> 减少 TCIV 中断
中断减少功能 2	—	—	—	—	<ul style="list-style-type: none"> 减少 TADCORA 和 TCNT、以及 TADCORB 和 TCNT 的比较次数 	—	—	<ul style="list-style-type: none"> 减少 TADCORA 和 TCNT、以及 TADCORB 和 TCNT 的比较次数
模块停止	MSTPCRA.MSTPA9 (注 1)							

○: 能、—: 不能

注 1. 有关模块停止的详细内容, 请参照“11. 低功耗功能”。

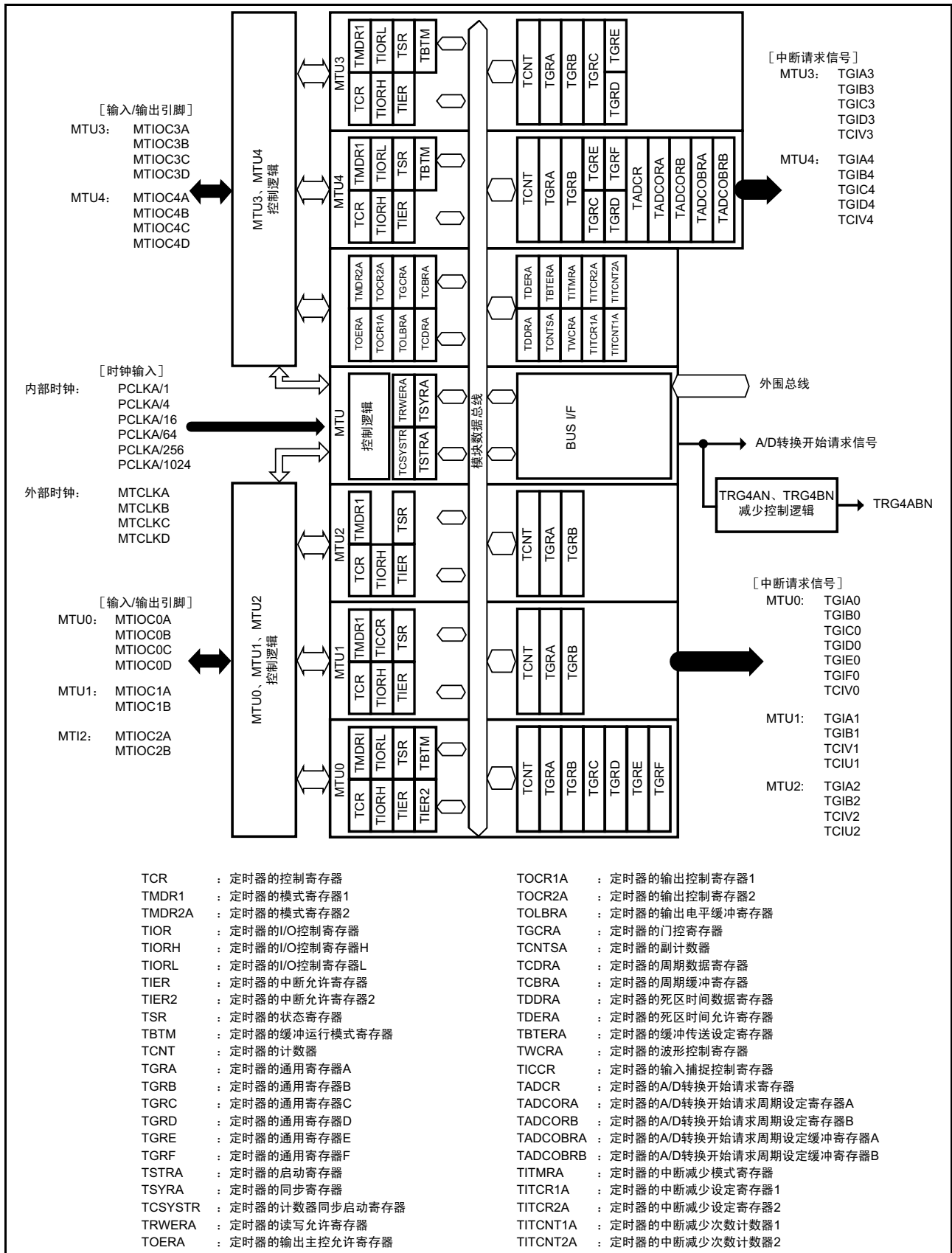


图 21.1 MTU 的框图 (MTU0 ~ 4)

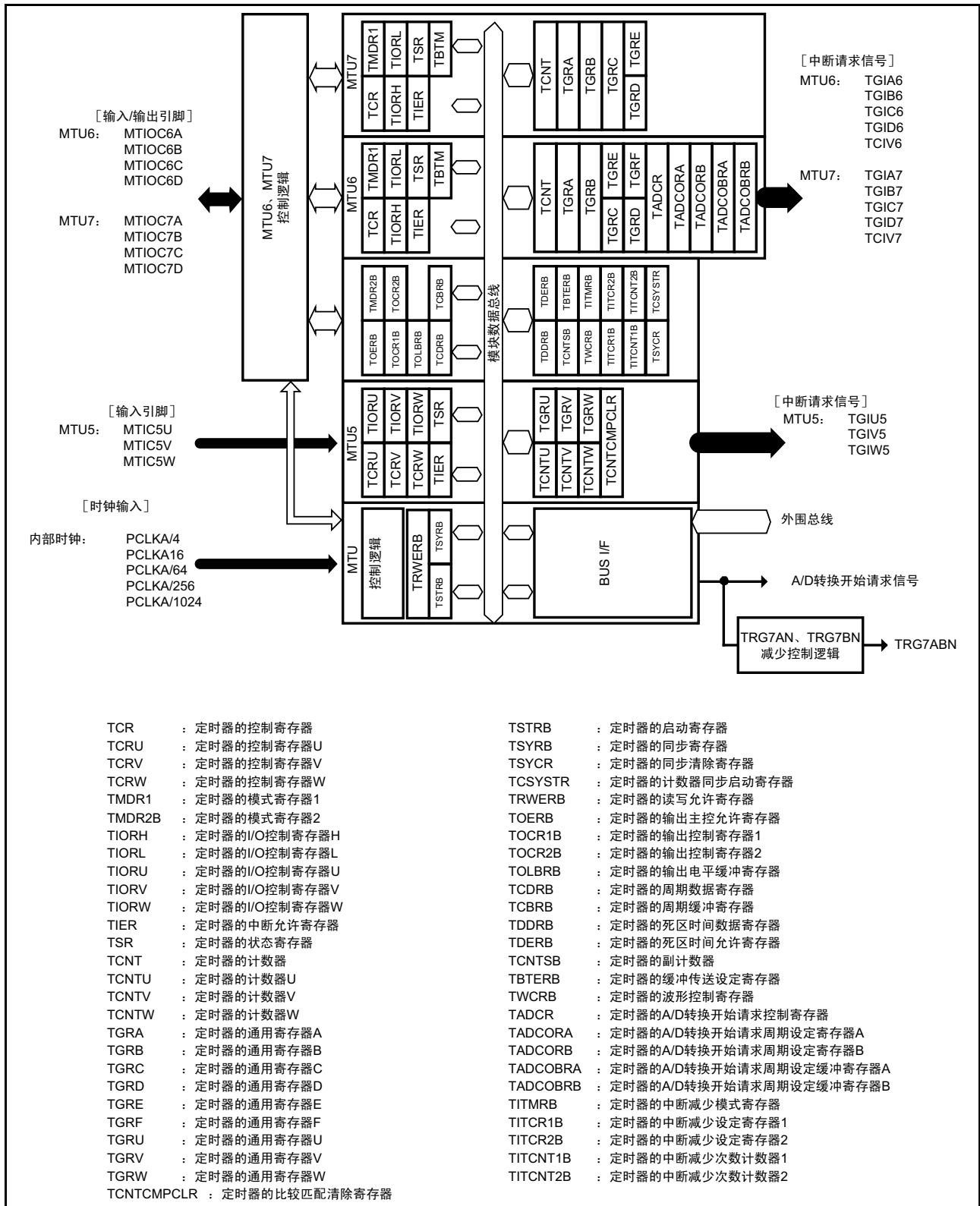


图 21.2 MTU 的框图 (MTU5 ~ 7)

MTU 使用的输入 / 输出引脚如表 21.3 所示。

表 21.3 MTU 的输入 / 输出引脚

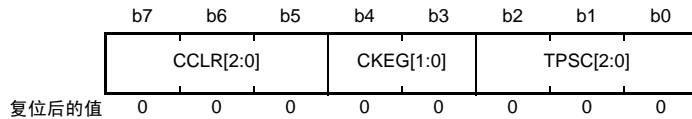
通道	引脚名	输入 / 输出	功能
MTU	MTCLKA	输入	外部时钟 A 的输入引脚 (MTU1 的相位计数模式的 A 相输入)
	MTCLKB	输入	外部时钟 B 的输入引脚 (MTU1 的相位计数模式的 B 相输入)
	MTCLKC	输入	外部时钟 C 的输入引脚 (MTU2 的相位计数模式的 A 相输入)
	MTCLKD	输入	外部时钟 D 的输入引脚 (MTU2 的相位计数模式的 B 相输入)
MTU0	MTIOC0A	输入 / 输出	MTU0.TGRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	MTIOC0B	输入 / 输出	MTU0.TGRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	MTIOC0C	输入 / 输出	MTU0.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
	MTIOC0D	输入 / 输出	MTU0.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
MTU1	MTIOC1A	输入 / 输出	MTU1.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
	MTIOC1B	输入 / 输出	MTU1.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
MTU2	MTIOC2A	输入 / 输出	MTU2.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
	MTIOC2B	输入 / 输出	MTU2.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
MTU3	MTIOC3A	输入 / 输出	MTU3.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
	MTIOC3B	输入 / 输出	MTU3.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
	MTIOC3C	输入 / 输出	MTU3.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
	MTIOC3D	输入 / 输出	MTU3.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
MTU4	MTIOC4A	输入 / 输出	MTU4.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
	MTIOC4B	输入 / 输出	MTU4.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
	MTIOC4C	输入 / 输出	MTU4.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
	MTIOC4D	输入 / 输出	MTU4.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
MTU5	MTIC5U	输入	MTU5.TGRU 的输入捕捉的输入引脚 / 外部脉冲的输入引脚
	MTIC5V	输入	MTU5.TGRV 的输入捕捉的输入引脚 / 外部脉冲的输入引脚
	MTIC5W	输入	MTU5.TGRW 的输入捕捉的输入引脚 / 外部脉冲的输入引脚
MTU6	MTIOC6A	输入 / 输出	MTU6.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
	MTIOC6B	输入 / 输出	MTU6.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
	MTIOC6C	输入 / 输出	MTU6.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
	MTIOC6D	输入 / 输出	MTU6.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
MTU7	MTIOC7A	输入 / 输出	MTU7.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
	MTIOC7B	输入 / 输出	MTU7.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
	MTIOC7C	输入 / 输出	MTU7.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
	MTIOC7D	输入 / 输出	MTU7.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

21.2 寄存器说明

21.2.1 定时器的控制寄存器 (TCR)

- MTU0、MTU1、MTU2、MTU3、MTU4、MTU6、MTU7

地址 MTU0.TCR 000C 1300h、MTU1.TCR 000C 1380h、MTU2.TCR 000C 1400h、MTU3.TCR 000C 1200h、MTU4.TCR 000C 1201h、MTU6.TCR 000C 1A00h、MTU7.TCR 000C 1A01h



位	符号	位名	功能	R/W
b2-b0	TPSC[2:0]	定时器的预分频器选择位	请参照表 21.6 ~ 表 21.9。	R/W
b4-b3	CKEG[1:0]	时钟边沿选择位	b4 b3 0 0: 在上升沿进行计数 0 1: 在下降沿进行计数 1 x: 在双边沿进行计数	R/W
b7-b5	CCLR[2:0]	计数器清除源选择位	请参照表 21.4 和表 21.5。	R/W

x: Don't care

TCR 寄存器控制各通道的 TCNT。MTU0 ~ 4、6、7 各有 1 个 TCR，MTU5 有 3 个 TCRU/V/W，共计 10 个。必须在 TCNT 停止计数的状态下设定 TCR 寄存器。

TPSC[2:0] 位 (定时器的预分频器选择位)

这些位选择 TCNT 计数器的时钟。各通道能独立选择时钟源，详细内容请参照表 21.6 ~ 表 21.9。

CKEG[1:0] 位 (时钟边沿选择位)

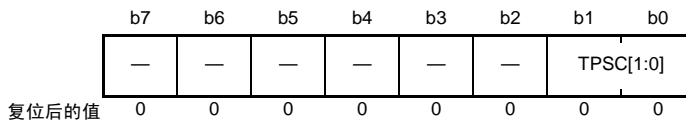
这些位选择输入时钟的边沿。如果在双边沿对内部时钟进行计数，输入时钟的周期就为 1/2 (例如: PCLKA/4 的双边沿 = PCLKA/2 的上升沿)。在 MTU1 和 MTU2 中使用相位计数模式时，忽视此设定而优先设定相位计数模式。内部时钟的边沿选择在输入时钟为 PCLKA/4 或者慢于 PCLKA/4 时有效。如果选择 PCLKA/1 或者其他通道的上溢 / 下溢作为输入时钟，虽然能写值，但是在运行时为初始值。

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，详细内容请参照表 21.4 和表 21.5。

- MTU5

地址 MTU5.TCRU 000C 1C84h、MTU5.TCRV 000C 1C94h、MTU5.TCRW 000C 1CA4h



位	符号	位名	功能	R/W
b1-b0	TPSC[1:0]	定时器的预分频器选择位	请参照表 21.10。	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

TPSC[1:0] 位 (定时器的预分配器选择位)

此位选择 TCNT 计数器的时钟。详细内容请参照表 21.10。

表 21.4 CCLR[2:0] (MTU0、3、4、6、7)

通道	bit7	bit6	bit5	说明
	CCLR2	CCLR1	CCLR0	
MTU0	0	0	0	禁止清除 TCNT。
MTU3	0	0	1	在发生 TGRA 的比较匹配 / 输入捕捉时清除 TCNT。
MTU4	0	1	0	在发生 TGRB 的比较匹配 / 输入捕捉时清除 TCNT。
MTU6	0	1	1	在清除正在进行同步清除或者同步运行的其他通道的计数器时清除 TCNT (注 1)。
MTU7	1	0	0	禁止清除 TCNT。
	1	0	1	在发生 TGRC 的比较匹配 / 输入捕捉时清除 TCNT (注 2)。
	1	1	0	在发生 TGRD 的比较匹配 / 输入捕捉时清除 TCNT (注 2)。
	1	1	1	在清除正在进行同步清除或者同步运行的其他通道的计数器时清除 TCNT (注 1)。

注 1. 通过将 TSYRA.SYNC 位和 TSYRB.SYNC 位置“1”，设定为同步运行。

注 2. 在将 TGRC 或者 TGRD 用作缓冲寄存器时，因为优先设定缓冲寄存器而且不发生比较匹配 / 输入捕捉，所以不清除 TCNT。

表 21.5 CCLR[2:0] (MTU1、2)

通道	bit7	bit6	bit5	说明
	保留位 (注2)	CCLR1	CCLR0	
MTU1	0	0	0	禁止清除 TCNT。
MTU2	0	0	1	在发生 TGRA 的比较匹配 / 输入捕捉时清除 TCNT。
	0	1	0	在发生 TGRB 的比较匹配 / 输入捕捉时清除 TCNT。
	0	1	1	在清除正在进行同步清除或者同步运行的其他通道的计数器时清除 TCNT (注 1)。

注 1. 通过将 TSYRA.SYNC 位和 TSYRB.SYNC 位置“1”，设定为同步运行。

注 2. 在 MTU1 和 MTU2 中，bit7 为保留位，读取值为“0”，写操作无效。

表 21.6 TPSC[2:0] (MTU0)

通道	bit2	bit1	bit0	说明
	TPSC2	TPSC1	TPSC0	
MTU0	0	0	0	内部时钟: 通过 PCLKA/1 进行计数
	0	0	1	内部时钟: 通过 PCLKA/4 进行计数
	0	1	0	内部时钟: 通过 PCLKA/16 进行计数
	0	1	1	内部时钟: 通过 PCLKA/64 进行计数
	1	0	0	外部时钟: 通过 MTCLKA 引脚输入进行计数
	1	0	1	外部时钟: 通过 MTCLKB 引脚输入进行计数
	1	1	0	外部时钟: 通过 MTCLKC 引脚输入进行计数
	1	1	1	外部时钟: 通过 MTCLKD 引脚输入进行计数

表 21.7 TPSC[2:0] (MTU1)

通道	bit2	bit1	bit0	说明
	TPSC2	TPSC1	TPSC0	
MTU1	0	0	0	内部时钟: 通过 PCLKA/1 进行计数
	0	0	1	内部时钟: 通过 PCLKA/4 进行计数
	0	1	0	内部时钟: 通过 PCLKA/16 进行计数
	0	1	1	内部时钟: 通过 PCLKA/64 进行计数
	1	0	0	外部时钟: 通过 MTCLKA 引脚输入进行计数
	1	0	1	外部时钟: 通过 MTCLKB 引脚输入进行计数
	1	1	0	内部时钟: 通过 PCLKA/256 进行计数
	1	1	1	通过 MTU2.TCNT 的上溢 / 下溢进行计数。

注. 在 MTU1 为相位计数模式时, 此设定无效。

表 21.8 TPSC[2:0] (MTU2)

通道	bit2	bit1	bit0	说明
	TPSC2	TPSC1	TPSC0	
MTU2	0	0	0	内部时钟: 通过 PCLKA/1 进行计数
	0	0	1	内部时钟: 通过 PCLKA/4 进行计数
	0	1	0	内部时钟: 通过 PCLKA/16 进行计数
	0	1	1	内部时钟: 通过 PCLKA/64 进行计数
	1	0	0	外部时钟: 通过 MTCLKA 引脚输入进行计数
	1	0	1	外部时钟: 通过 MTCLKB 引脚输入进行计数
	1	1	0	外部时钟: 通过 MTCLKC 引脚输入进行计数
	1	1	1	内部时钟: 通过 PCLKA/1024 进行计数

注. 在 MTU2 为相位计数模式时, 此设定无效。

表 21.9 TPSC[2:0] (MTU3、4、6、7)

通道	bit2	bit1	bit0	说明
	TPSC2	TPSC1	TPSC0	
MTU3	0	0	0	内部时钟: 通过 PCLKA/1 进行计数
MTU4	0	0	1	内部时钟: 通过 PCLKA/4 进行计数
MTU6	0	1	0	内部时钟: 通过 PCLKA/16 进行计数
MTU7	0	1	1	内部时钟: 通过 PCLKA/64 进行计数
	1	0	0	内部时钟: 通过 PCLKA/256 进行计数
	1	0	1	内部时钟: 通过 PCLKA/1024 进行计数
	1	1	0	外部时钟: 通过 MTCLKA 引脚输入进行计数 (注 1)
	1	1	1	外部时钟: 通过 MTCLKB 引脚输入进行计数 (注 1)

注 1 在 MTU6 和 MTU7 中不能设定。

表 21.10 TPSC[1:0] (MTU5)

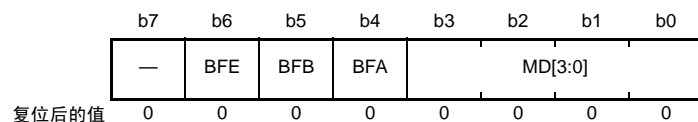
通道	bit1	bit0	说明
	TPSC1	TPSC0	
MTU5	0	0	内部时钟: 通过 PCLKA/1 进行计数
	0	1	内部时钟: 通过 PCLKA/4 进行计数
	1	0	内部时钟: 通过 PCLKA/16 进行计数
	1	1	内部时钟: 通过 PCLKA/64 进行计数

注. 在 MTU5 中, bit7 ~ 2 为保留位, 读写值都为“0”。

21.2.2 定时器的模式寄存器 1 (TMDR1)

• MTU0.TMDR1

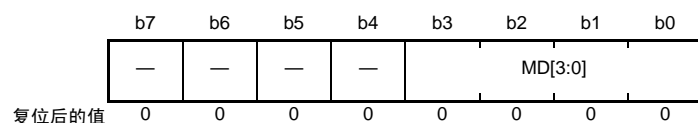
地址 MTU0.TMDR1 000C 1301h



复位后的值

• MTU1.TMDR1、MRU2.TMDR1

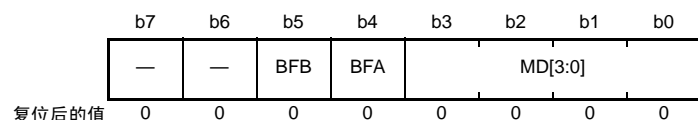
地址 MTU1.TMDR1 000C 1381h、MTU2.TMDR1 000C 1401h



复位后的值

• MTU3.TMDR1、MTU4.TMDR1、MTU6.TMDR1、MTU7.TMDR1

地址 MTU3.TMDR1 000C 1202h、MTU4.TMDR1 000C 1203h、MTU6.TMDR1 000C 1A02h、MTU7.TMDR1 000C 1A03h



复位后的值

位	符号	位名	功能	R/W
b3-b0	MD[3:0]	模式选择位	设定定时器的运行模式，请参照表 21.11。	R/W
b4	BFA	缓冲运行 A 位	0: TGRA 寄存器和 TGRC 寄存器进行正常运行 1: TGRA 寄存器和 TGRC 寄存器进行缓冲运行	R/W
b5	BFB	缓冲运行 B 位	0: TGRB 寄存器和 TGRD 寄存器进行正常运行 1: TGRB 寄存器和 TGRD 寄存器进行缓冲运行	R/W
b6	BFE	缓冲运行 E 位	0: MTU0.TGRE 寄存器和 MTU0.TGRF 寄存器进行正常运行 1: MTU0.TGRE 寄存器和 MTU0.TGRF 寄存器进行缓冲运行	R/W
b7	—	保留位	读写值都为“0”。	R/W

TMDR1 寄存器设定各通道的运行模式。MTU0 ~ 4、6、7 各有 1 个 TMDR1，共计 7 个。必须在 TCNT 停止计数的状态下设定 TMDR1 寄存器。

表 21.11 通过 MD[3:0] 位设定的运行模式

bit3	bit2	bit1	bit0	说明
MD3	MD2	MD1	MD0	
0	0	0	0	正常运行
0	0	0	1	禁止设定。
0	0	1	0	PWM 模式 1
0	0	1	1	PWM 模式 2 (注 1)
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)
1	0	0	0	复位同步 PWM 模式 (注 3)
1	0	0	1	不能设定。
1	0	1	x	不能设定。
1	1	0	0	不能设定。
1	1	0	1	互补 PWM 模式 1 (在波峰进行传送) (注 3)
1	1	1	0	互补 PWM 模式 2 (在波谷进行传送) (注 3)
1	1	1	1	互补 PWM 模式 3 (在波峰和波谷进行传送) (注 3)

x: Don't care

注 1. MTU3、4、6、7 不能设定为 PWM 模式 2。

注 2. MTU0、3、4、6、7 不能设定为相位计数模式。

注 3. 只有 MTU3 和 MTU6 能设定为复位同步 PWM 模式或者互补 PWM 模式。

如果将 MTU3 和 MTU6 设定为复位同步 PWM 模式或者互补 PWM 模式，MTU4 和 MTU7 的设定就无效并且自动服从 MTU3 和 MTU6 的设定。必须给 MTU4 和 MTU7 设定初始值 (正常运行)。

MTU0、1、2 不能设定为复位同步 PWM 模式或者互补 PWM 模式。

BFA 位 (缓冲运行 A 位)

此位设定是使 TGRA 寄存器进行正常运行还是组合 TGRA 寄存器和 TGRC 寄存器进行缓冲运行。如果将 TGRC 寄存器用作缓冲寄存器，就不发生 TGRC 寄存器的输入捕捉 / 输出比较。

另外，在复位同步 PWM 模式和互补 PWM 模式中的 MTU3 和 MTU4 (MTU6 和 MTU7) 的缓冲运行根据 MTU3.TMDR1 (MTU6.TMDR1) 的 BFA 位的设定进行运行。必须将 MTU4.TMDR1 (MTU7.TMDR1) 的 BFA 位置“0”。

在没有 TGRC 寄存器的 MTU1 和 MTU2 中，此位为保留位，读写值都为“0”。

BFB 位 (缓冲运行 B 位)

此位设定是使 TGRB 寄存器进行正常运行还是组合 TGRB 寄存器和 TGRD 寄存器进行缓冲运行。如果将 TGRD 寄存器用作缓冲寄存器，就不发生 TGRD 寄存器的输入捕捉 / 输出比较。

另外，在复位同步 PWM 模式和互补 PWM 模式中的 MTU3 和 MTU4 (MTU6 和 MTU7) 的缓冲运行根据 MTU3.TMDR1 (MTU6.TMDR1) 的 BFB 位的设定进行运行。必须将 MTU4.TMDR1 (MTU7.TMDR1) 的 BFB 位置“0”。

在没有 TGRD 寄存器的 MTU1 和 MTU2 中，此位为保留位，读写值都为“0”。

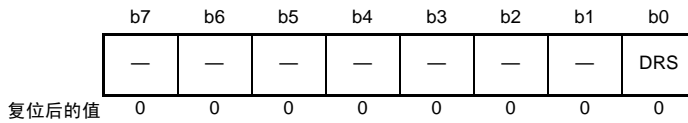
BFE 位 (缓冲运行 E 位)

此位选择是否使 MTU0.TGRE 和 MTU0.TGRF 进行正常运行或者缓冲运行。

在 MTU1 ~ 4、6、7 中，此位为保留位，读写值都为“0”。

21.2.3 定时器的模式寄存器 2 (TMDR2A、TMDR2B)

地址 MTU.TMDR2A 000C 1270h、MTU.TMDR2B 000C 1A70h



位	符号	位名	功能	R/W
b0	DRS	双缓冲选择位	0: 双缓冲功能无效 1: 双缓冲功能有效	R/W
b7	—	保留位	读写值都为“0”。	R/W

TMDR2 寄存器设定互补 PWM 模式 3 (在波峰、波谷传送) 中的双缓冲功能。MTU3 (TMDR2A)、MTU6 (TMDR2B) 各有一个 TMDR2, 共计 2 个。必须在 TCNT 停止计数的状态下设定 TMDR2A 寄存器和 TMDR2B 寄存器。

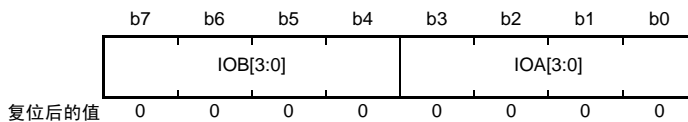
DRS 位 (双缓冲选择位)

此位选择互补 PWM 模式中的双缓冲功能的有效 / 无效。

21.2.4 定时器的 I/O 控制寄存器 (TIOR)

- MTU0.TIORH、MTU1.TIOR、MTU2.TIOR、MTU3.TIORH、MTU4.TIORH、MTU6.TIORH、MTU7.TIORH

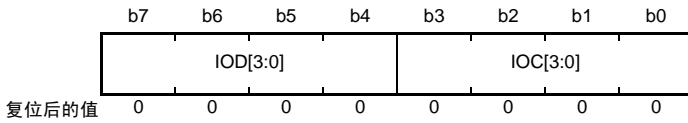
地址 MTU0.TIORH 000C 1302h、MTU1.TIOR 000C 1382h、MTU2.TIOR 000C 1402h、MTU3.TIORH 000C 1204h、MTU4.TIORH 000C 1206h、MTU6.TIORH 000C 1A04h、MTU7.TIORH 000C 1A06h



位	符号	位名	功能	R/W
b3-b0	IOA[3:0]	I/O 控制 A 位	请参照下表: MTU0.TIORH: 表 21.24、MTU1.TIOR: 表 21.26 MTU2.TIOR: 表 21.27、MTU3.TIORH: 表 21.28 MTU4.TIORH: 表 21.30、MTU6.TIORH: 表 21.32 MTU7.TIORH: 表 21.34	R/W
b7-b4	IOB[3:0]	I/O 控制 B 位	请参照下表: MTU0.TIORH: 表 21.12、MTU1.TIOR: 表 21.14、 MTU2.TIOR: 表 21.15、MTU3.TIORH: 表 21.16、 MTU4.TIORH: 表 21.18、MTU6.TIORH: 表 21.20、 MTU7.TIORH: 表 21.22	R/W

- MTU0.TIORL、MTU3.TIORL、MTU4.TIORL、MTU6.TIORL、MTU7.TIORL

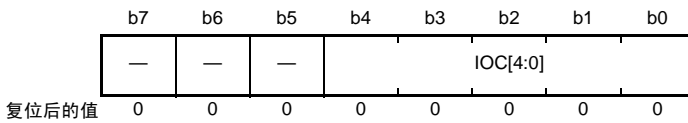
地址 MTU0.TIORL 000C 1303h、MTU3.TIORL 000C 1205h、MTU4.TIORL 000C 1207h、MTU6.TIORL 000C 1A05h、MTU7.TIORL 000C 1A07h



位	符号	位名	功能	R/W
b3-b0	IOC[3:0]	I/O 控制 C 位	请参照下表： MTU0.TIORL：表 21.25、MTU3.TIORL：表 21.29、 MTU4.TIORL：表 21.31、MTU6.TIORL：表 21.33、 MTU7.TIORL：表 21.35	R/W
b7-b4	IOD[3:0]	I/O 控制 D 位	请参照下表： MTU0.TIORL：表 21.13、MTU3.TIORL：表 21.17、 MTU4.TIORL：表 21.19、MTU6.TIORL：表 21.21、 MTU7.TIORL：表 21.23	R/W

- MTU5.TIORU、MTU5.TIORV、MTU5.TIORW

地址 MTU5.TIORU 000C 1C86h、MTU5.TIORV 000C 1C96h、MTU5.TIORW 000C 1CA6h



位	符号	位名	功能	R/W
b4-b0	IOC[4:0]	I/O 控制 C 位	请参照下表： MTU5.TIORU、MTU5.TIORV、MTU5.TIORW： 表 21.36	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W

TIOR 寄存器控制 TGR 寄存器。MTU0、3、4、6、7 各有 2 个 TIOR，MTU1、2 各有 1 个 TIOR，MTU5 有 3 个 MTU5.TIORU/V/W，共计 15 个。

必须注意：TMDR1 寄存器的设定影响 TIOR 寄存器。

在计数器停止计数（将 TSTRA.CST 位和 TSYRB.CST 置“0”）的状态下，TIOR 寄存器指定的初始输出有效。另外，在 PWM 模式 2 的情况下，指定计数器变为“0”时的输出。

如果将 TGRC 寄存器或者 TGRD 寄存器设定为缓冲运行，此设定就无效并且用作缓冲寄存器。

表 21.12 TIORH (MTU0)

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	MTU0.TGRB 的功能	MTIOC0B 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		捕捉输入源为 MTU1/ 计数时钟。 在 MTU1.TCNT 的递增计数或者递减计数时进行输入捕捉。

x: Don't care

表 21.13 TIORL (MTU0)

bit7	bit6	bit5	bit4	说明	
IOD3	IOD2	IOD1	IOD0	MTU0.TGRD 的功能	MTIOC0D 引脚的功能
0	0	0	0	输出比较寄存器 (注 1)	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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)	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		捕捉输入源为 MTU1/ 计数时钟。 在 MTU1.TCNT 的递增计数或者递减计数时进行输入捕捉。

x: Don't care n=0、6

注 1. 如果将 MTU0.TMDR1.BFB 位置“1”, 将 MTU0.TGRD 寄存器用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 21.14 TIOR (MTU1)

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	MTU1.TGRB 的功能	MTIOC1B 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		在发生 MTU0.TGRC 的比较匹配 / 输入捕捉时进行输入捕捉。

x: Don't care

表 21.15 TIOR (MTU2)

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	MTU2.TGRB 的功能	MTIOC2B 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

表 21.16 TIORH (MTU3)

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	MTU3.TGRB 的功能	MTIOC3B 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

表 21.17 TIORL (MTU3)

bit7	bit6	bit5	bit4	说明	
IOD3	IOD2	IOD1	IOD0	MTU3.TGRD 的功能	MTIOC3D 引脚的功能
0	0	0	0	输出比较寄存器 (注 1)	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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 (注 1)	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

注 1. 如果将 MTU3.TMDR1.BFB 位置“1”, 将 MTU3.TGRD 寄存器用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 21.18 TIORH (MTU4)

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	MTU4.TGRB 的功能	MTIOC4B 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

表 21.19 TIORL (MTU4)

bit7	bit6	bit5	bit4	说明	
IOD3	IOD2	IOD1	IOD0	MTU4.TGRD 的功能	MTIOC4D 引脚的功能
0	0	0	0	输出比较寄存器 (注 1)	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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 (注 1)	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

注 1. 如果将 MTU4.TMDR1.BFB 位置“1”, 将 MTU4.TGRD 寄存器用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 21.20 TIORH (MTU6)

bit3	bit2	bit1	bit0	说明	
IOB3	IOB2	IOB1	IOB0	MTU6.TGRB 的功能	MTIOC6B 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

表 21.21 TIORL (MTU6)

bit3	bit2	bit1	bit0	说明	
IOD3	IOD2	IOD1	IOD0	MTU6.TGRD 的功能	MTIOC6D 引脚的功能
0	0	0	0	输出比较寄存器 (注 1)	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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 (注 1)	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

注 1. 如果将 MTU6.TMDR1.BFB 位置“1”, 将 MTU6.TGRD 寄存器用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 21.22 TIORH (MTU7)

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	MTU7.TGRB 的功能	MTIOC7B 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

表 21.23 TIORL (MTU7)

bit7	bit6	bit5	bit4	说明	
IOD3	IOD2	IOD1	IOD0	MTU7.TGRD 的功能	MTIOC7D 引脚的功能
0	0	0	0	输出比较寄存器 (注 1)	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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 (注 1)	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

注 1. 如果将 MTU7.TMDR1.BFB 位置“1”, 将 MTU7.TGRD 寄存器用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 21.24 TIORH (MTU0)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	MTU0.TGRA 的功能	MTIOC0A 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		捕捉输入源为 MTU1/ 计数时钟。 在 MTU1.TCNT 的递增计数或者递减计数时进行输入捕捉。

x: Don't care

表 21.25 TIORL (MTU0)

bit3	bit2	bit1	bit0	说明	
IOC3	IOC2	IOC1	IOC0	MTU0.TGRC 的功能	MTIOC0C 引脚的功能
0	0	0	0	输出比较寄存器 (注 1)	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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)	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		捕捉输入源为 MTU1/ 计数时钟。 在 MTU1.TCNT 的递增计数或者递减计数时进行输入捕捉。

x: Don't care

注 1. 如果将 MTU0.TMDR1.BFA 位置“1”, 将 MTU0.TGRC 寄存器用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 21.26 TIOR (MTU1)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	MTU1.TGRA 的功能	MTIOC1A 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		在发生 MTU0.TGRA 的比较匹配 / 输入捕捉时进行输入捕捉。

x: Don't care

表 21.27 TIOR (MTU2)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	MTU2.TGRA 的功能	MTIOC2A 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

表 21.28 TIORH (MTU3)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	MTU3.TGRA 的功能	MTIOC3A 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

表 21.29 TIORL (MTU3)

bit3	bit2	bit1	bit0	说明	
IOC3	IOC2	IOC1	IOC0	MTU3.TGRC 的引脚	MTIOC3C 引脚的功能
0	0	0	0	输出比较寄存器 (注 1)	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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 (注 1)	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

注 1. 如果将 MTU3.TMDR1.BFA 位置“1”, 将 MTU3.TGRC 寄存器用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 21.30 TIORH (MTU4)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	MTU4.TGRA 的功能	MTIOC4A 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

表 21.31 TIORL (MTU4)

bit3	bit2	bit1	bit0	说明	
IOC3	IOC2	IOC1	IOC0	MTU4.TGRC 的功能	MTIOC4C 引脚的功能
0	0	0	0	输出比较寄存器 (注 1)	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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 (注 1)	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

注 1. 如果将 MTU4.TMDR1.BFA 位置“1”, 将 MTU4.TGRC 寄存器用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 21.32 TIORH (MTU6)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	MTU6.TGRA 的功能	MTIOC6A 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

表 21.33 TIORL (MTU6)

bit3	bit2	bit1	bit0	说明	
IOC3	IOC2	IOC1	IOC0	MTU6.TGRC 的功能	MTIOC6C 引脚的功能
0	0	0	0	输出比较寄存器 (注 1)	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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 (注 1)	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

注 1. 如果将 MTU6.TMDR1.BFA 位置“1”, 将 MTU6.TGRC 寄存器用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 21.34 TIORH (MTU7)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	MTU7.TGRA 的功能	MTIOC7A 引脚的功能
0	0	0	0	输出比较寄存器	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

表 21.35 TIORL (MTU7)

bit3	bit2	bit1	bit0	说明	
IOC3	IOC2	IOC1	IOC0	MTU7.TGRC 的功能	MTIOC7C 引脚的功能
0	0	0	0	输出比较寄存器 (注 1)	禁止输出。
0	0	0	1		初始输出为 Low 电平, 在比较匹配时输出 Low 电平。
0	0	1	0		初始输出为 Low 电平, 在比较匹配时输出 High 电平。
0	0	1	1		初始输出为 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		初始输出为 High 电平, 在比较匹配时输出 Low 电平。
0	1	1	0		初始输出为 High 电平, 在比较匹配时输出 High 电平。
0	1	1	1		初始输出为 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 (注 1)	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

x: Don't care

注 1. 如果将 MTU7.TMDR1.BFA 位置“1”, 将 MTU7.TGRC 寄存器用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 21.36 TIORU、TIO RV、TIO RW (MTU5)

bit4	bit3	bit2	bit1	bit0	说明	
IOC4	IOC3	IOC2	IOC1	IOC0	MTU5.TGRU、MTU5.TGRV、 MTU5.TGRW 的功能	MTIC5U、MTIC5V、MTIC5W 引脚的功能
0	0	0	0	0	比较匹配寄存器	比较匹配
0	0	0	0	1		不能设定。
0	0	0	1	x		不能设定。
0	0	1	x	x		不能设定。
0	1	x	x	x		不能设定。
1	0	0	0	0	输入捕捉寄存器	不能设定。
1	0	0	0	1		在上升沿进行输入捕捉。
1	0	0	1	0		在下降沿进行输入捕捉。
1	0	0	1	1		在双边沿进行输入捕捉。
1	0	1	x	x		不能设定。
1	1	0	0	0		不能设定。
1	1	0	0	1		用于测量外部输入信号的 Low 电平脉宽。 在互补 PWM 模式的波谷进行捕捉。
1	1	0	1	0		用于测量外部输入信号的 Low 电平脉宽。 在互补 PWM 模式的波峰进行捕捉。
1	1	0	1	1		用于测量外部输入的信号 Low 电平脉宽。 在互补 PWM 模式的波峰和波谷进行捕捉。
1	1	1	0	0		不能设定。
1	1	1	0	1		用于测量外部输入信号的 High 电平脉宽。 在互补 PWM 模式的波谷进行捕捉。
1	1	1	1	0		用于测量外部输入信号的 High 电平脉宽。 在互补 PWM 模式的波峰进行捕捉。
1	1	1	1	1		用于测量外部输入信号的 High 电平脉宽。 在互补 PWM 模式的波峰和波谷进行捕捉。

x: Don't care

21.2.5 定时器的比较匹配清除寄存器 (TCNTCMPCLR)

地址 MTU5.TCNTCMPCLR 000C 1CB6h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	CMPCLR5W	TCNT 比较清除 5W 位	0: 禁止通过 MTU5.TCNTW 和 MTU5.TGRW 的比较匹配 / 输入捕捉将 MTU5.TCNTW 清 “0000h” 1: 允许通过 MTU5.TCNTW 和 MTU5.TGRW 的比较匹配 / 输入捕捉将 MTU5.TCNTW 清 “0000h”	R/W
b1	CMPCLR5V	TCNT 比较清除 5V 位	0: 禁止通过 MTU5.TCNTV 和 MTU5.TGRV 的比较匹配 / 输入捕捉将 MTU5.TCNTV 清 “0000h” 1: 允许通过 MTU5.TCNTV 和 MTU5.TGRV 的比较匹配 / 输入捕捉将 MTU5.TCNTV 清 “0000h”	R/W
b2	CMPCLR5U	TCNT 比较清除 5U 位	0: 禁止通过 MTU5.TCNTU 和 MTU5.TGRU 的比较匹配 / 输入捕捉将 MTU5.TCNTU 清 “0000h” 1: 允许通过 MTU5.TCNTU 和 MTU5.TGRU 的比较匹配 / 输入捕捉将 MTU5.TCNTU 清 “0000h”	R/W
b7-b3	—	保留位	读写值都为 “0”。	R/W

TCNTCMPCLR 寄存器设定 MTU5.TCNTU、MTU5.TCNTV 和 MTU5.TCNTW 的清除请求。MTU5 有 1 个 TCNTCMPCLR 寄存器。

CMPCLR5n 位 (TCNT 比较清除 5n 位) (n=U、V、W)

此位允许或者禁止通过 MTU5.TCNTn 和 MTU5.TGRn 的比较匹配 / 输入捕捉清除 MTU5.TCNTn 的请求。

21.2.6 定时器的中断允许寄存器 (TIER)

- TIER (MTU1、MTU2)

地址 MTU1.TIER 000C 1384h、MTU2.TIER 000C 1404h

b7	b6	b5	b4	b3	b2	b1	b0
TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
复位后的值	0	0	0	0	0	0	0

- TIER (MTU0、MTU3、MTU6)

地址 MTU0.TIER 000C 1304h、MTU3.TIER 000C 1208h、MTU6.TIER 000C 1A08h

b7	b6	b5	b4	b3	b2	b1	b0
TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
复位后的值	0	0	0	0	0	0	0

- TIER (MTU4、MTU7)

地址 MTU4.TIER 000C 1209h、MTU7.TIER 000C 1A09h

b7	b6	b5	b4	b3	b2	b1	b0
TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
复位后的值	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	TGIEA	TGR 中断允许 A 位	0: 禁止中断请求 (TGIA) 1: 允许中断请求 (TGIA)	R/W
b1	TGIEB	TGR 中断允许 B 位	0: 禁止中断请求 (TGIB) 1: 允许中断请求 (TGIB)	R/W
b2	TGIEC	TGR 中断允许 C 位	0: 禁止中断请求 (TGIC) 1: 允许中断请求 (TGIC)	R/W
b3	TGIED	TGR 中断允许 D 位	0: 禁止中断请求 (TGID) 1: 允许中断请求 (TGID)	R/W
b4	TCIEV	上溢中断允许位	0: 禁止中断请求 (TCIV) 1: 允许中断请求 (TCIV)	R/W
b5	TCIEU	下溢中断允许位	0: 禁止中断请求 (TCIU) 1: 允许中断请求 (TCIU)	R/W
b6	TTGE2	A/D 转换开始请求允许 2 位	0: 禁止因 MTUn.TCNT 的下溢 (波谷) 而产生的 A/D 转换开始请求 1: 允许因 MTUn.TCNT 的下溢 (波谷) 而产生的 A/D 转换开始请求	R/W
b7	TTGE	A/D 转换开始请求允许位	0: 禁止产生 A/D 转换开始请求 1: 允许产生 A/D 转换开始请求	R/W

n=4、7

TIER 寄存器允许或者禁止各通道的中断请求。在 MTU0 中有 2 个 TIER，MTU1 ~ 7 各有 1 个 TIER，共计 9 个。

TGIEA 位、TGIEB 位 (TGR 中断允许 A 位、B 位)

在 TSR.TGF_n 标志变为“1”时，这些位允许或者禁止因 TSR.TGF_n 标志而产生的中断请求 (TGIn) (n=A、B)。

TGIEC 位、TGIED 位 (TGR 中断允许 C 位、D 位)

在 MTU0、3、4、6、7 中的 TSR.TGF_n 标志变为“1”时，这些位允许或者禁止因 TSR.TGF_n 标志而产生的中断请求 (TGIn) (n=C、D)。

在 MTU1 和 MTU2 中，这些位为保留位，读写值都为“0”。

TCIEV 位 (上溢中断允许位)

在 TSR.TCFV 标志变为“1”时，此位允许或者禁止因 TSR.TCFV 标志而产生的中断请求 (TCIV)。

TCIEU 位 (下溢中断允许位)

在 MTU1 和 MTU2 中的 TSR.TCFU 标志变为“1”时，此位允许或者禁止因 TSR.TCFU 而产生的中断请求 (TCIU)。

在 MTU0、3、4、6、7 中，此位为保留位，读写值都为“0”。

TTGE2 位 (A/D 转换开始请求允许 2 位)

在互补 PWM 模式中，此位允许或者禁止因 MTU_n.TCNT 的下溢 (波谷) 而产生的 A/D 转换开始请求 (n=4、7)。

在 MTU0 ~ 3、6 中，此位为保留位，读写值都为“0”。

TTGE 位 (A/D 转换开始请求允许位)

此位允许或者禁止因 TGRA 寄存器的输入捕捉 / 比较匹配而产生的 A/D 转换开始请求。

- TIER2 (MTU0)

地址 MTU0.TIER2 000C 1324h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE2	—	—	—	—	—	TGIEF	TGIEE
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	TGIEE	TGR 中断允许 E 位	0: 禁止中断请求 (TGIE) 1: 允许中断请求 (TGIE)	R/W
b1	TGIEF	TGR 中断允许 F 位	0: 禁止中断请求 (TGIF) 1: 允许中断请求 (TGIF)	R/W
b6-b2	—	保留位	读写值都为“0”。	R/W
b7	TTGE2	A/D 转换开始请求 2 位	0: 禁止因 MTU0.TCNT 和 MTU0.TGRE 的比较匹配而产生的 A/D 转换开始请求 1: 允许因 MTU0.TCNT 和 MTU0.TGRE 的比较匹配而产生的 A/D 转换开始请求	R/W

TGIEE 位、TGIEF 位 (TGR 中断允许 E 位、F 位)

此位允许或者禁止因 MTU0.TCNT 和 MTU0.TGR_n 的比较匹配而产生的中断请求 (n=E、F)。

TTGE2 位 (A/D 转换开始请求允许 2 位)

此位允许或者禁止因 MTU0.TCNT 和 MTU0.TGRE 的比较匹配而产生的 A/D 转换开始请求。

- TIER (MTU5)

地址 MTU5.TIER 000C 1CB2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	TGIE5W	TGR 中断允许 5W 位	0: 禁止 TGIW5 中断请求 1: 允许 TGIW5 中断请求	R/W
b1	TGIE5V	TGR 中断允许 5V 位	0: 禁止 TGI5V 中断请求 1: 允许 TGI5V 中断请求	R/W
b2	TGIE5U	TGR 中断允许 5U 位	0: 禁止 TGIU5 中断请求 1: 允许 TGIU5 中断请求	R/W
b7-b3	—	(保留位)	读写值都为“0”。	R/W

TGIE5_n 位 (TGR 中断允许 5_n 位)

在 MTU5.TSR.CMF_{n5} 标志变为“1”时, 这些位允许或者禁止因 MTU5.TSR.CMF_{n5} 标志而产生的中断请求 (TGIn5) (n=U、V、W)。

21.2.7 定时器的状态寄存器 (TSR)

• TSR (MTU0)

地址 MTU0.TSR 000C 1305h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
复位后的值	1	1	0	0	0	0	0	0

• TSR (MTU1、MTU2)

地址 MTU1.TSR 000C 1385h、MTU2.TSR 000C 1405h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
复位后的值	1	1	0	0	0	0	0	0

• TSR (MTU3、MTU4、MTU6、MTU7)

地址 MTU3.TSR 000C 122Ch、MTU4.TSR 000C 122Dh、MTU6.TSR 000C 1A2Ch、MTU7.TSR 000C 1A2Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
复位后的值	1	1	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	TGFA	输入捕捉 / 输出比较标志 A	0: 未发生 TGRA 寄存器的输入捕捉或者比较匹配 1: 发生 TGRA 寄存器的输入捕捉或者比较匹配	R/(W) (注 1)
b1	TGFB	输入捕捉 / 输出比较标志 B	0: 未发生 TGRB 寄存器的输入捕捉或者比较匹配 1: 发生 TGRB 寄存器的输入捕捉或者比较匹配	R/(W) (注 1)
b2	TGFC	输入捕捉 / 输出比较标志 C	0: 未发生 TGRC 寄存器的输入捕捉或者比较匹配 1: 发生 TGRC 寄存器的输入捕捉或者比较匹配	R/(W) (注 1)
b3	TGFD	输入捕捉 / 输出比较标志 D	0: 未发生 TGRD 寄存器的输入捕捉或者比较匹配 1: 发生 TGRD 寄存器的输入捕捉或者比较匹配	R/(W) (注 1)
b4	TCFV	上溢标志	0: 未发生 TCNT 的上溢 1: 发生 TCNT 的上溢	R/(W) (注 1)
b5	TCFU	下溢标志	0: 在 MTU1 和 MTU2 为相位计数模式时, 未发生 TCNT 的下溢 1: 在 MTU1 和 MTU2 为相位计数模式时, 发生 TCNT 的下溢	R/(W) (注 1)
b6	—	保留位	都写值都为“1”。	R/W
b7	TCFD	计数方向标志	0: TCNT 进行递减计数 1: TCNT 进行递增计数	R

注 1 为了将标志置“0”，只能在读“1”后写“0”。

TSR 寄存器表示各通道的状态。MTU0 有 2 个 TSR，MTU1 ~ 7 各有 1 个 TSR，共计 9 个。

TGFA 标志 (输入捕捉 / 输出比较标志 A)

这是表示发生 TGRA 寄存器的输入捕捉或者比较匹配的状态标志。只能写用于将标志置“0”的“0”。

[为“1”的条件]

- 在 TGRA 寄存器用作输出比较寄存器的情况下，TCNT=TGRA 时
- 在 TRGA 寄存器用作输入捕捉寄存器的情况下，通过输入捕捉信号将 TCNT 的值传送到 TGRA 寄存器时

[为“0”的条件]

- 在 TGFA=1 的状态下读 TGFA 标志后，给 TGFA 标志写“0”时

TGFB 标志 (输入捕捉 / 输出比较标志 B)

这是表示发生 TGRB 寄存器的输入捕捉或者比较匹配的状态标志。只能写用于将标志置“0”的“0”。

[为“1”的条件]

- 在 TGRB 寄存器用作输出比较寄存器的情况下，TCNT=TGRB 时
- 在 TRGB 寄存器用作输入捕捉寄存器的情况下，通过输入捕捉信号将 TCNT 的值传送到 TGRB 寄存器时

[为“0”的条件]

- 在 TGFB=1 的状态下读 TGFB 标志后，给 TGFB 标志写“0”时

TGFC 标志 (输入捕捉 / 输出比较标志 C)

这是表示发生 MTU0、3、4、6、7 的 TGRC 寄存器的输入捕捉或者比较匹配的状态标志。只能写用于将标志置“0”的“0”。

[为“1”的条件]

- 在 TGRC 寄存器用作输出比较寄存器的情况下，TCNT=TGRC 时
- 在 TRGC 寄存器用作输入捕捉寄存器的情况下，通过输入捕捉信号将 TCNT 的值传送到 TGRC 寄存器时

[为“0”的条件]

- 在 TGFC=1 的状态下读 TGFC 标志后，给 TGFC 标志写“0”时

在 MTU1 和 MTU2 中，此标志为保留位。读写值都为“0”。

TGFD 标志 (输入捕捉 / 输出比较标志 D)

这是表示发生 MTU0、3、4、6、7 的 TGRD 寄存器的输入捕捉或者比较匹配的状态标志。只能写用于将标志置“0”的“0”。

[为“1”的条件]

- 在 TGRD 寄存器用作输出比较寄存器的情况下，TCNT=TGRD 时
- 在 TRGD 寄存器用作输入捕捉寄存器的情况下，通过输入捕捉信号将 TCNT 的值传送到 TGRD 寄存器时

[为“0”的条件]

- 在 TGFD=1 的状态下读 TGFD 标志后，给 TGFD 标志写“0”时

在 MTU1 和 MTU2 中，此标志为保留位。读写值都为“0”。

TCFV 标志 (上溢标志)

这是表示发生 TCNT 的上溢的状态标志。只能写用于将标志置“0”的“0”。

[为“1”的条件]

- 在 TCNT 的值发生上溢时 (FFFFh → 0000h)
在 MTU4 和 MTU7 为互补 PWM 模式时, 如果 MTU4.TCNT、MTU7.TCNT 的值发生下溢 (0001h → 0000h), TCFV 标志也变为“1”。

[为“0”的条件]

- 在 TCFV=1 的状态下读 TCFV 标志后, 给 TCFV 标志写“0”时

TCFU 标志 (下溢标志)

这是表示在 MTU1 和 MTU2 为相位计数模式时发生 TCNT 的下溢的状态标志。只能写用于将标志置“0”的“0”。

[为“1”的条件]

- 在 TCNT 的值发生下溢时 (0000h → FFFFh)

[为“0”的条件]

- 在 TCFU=1 的状态下读 TCFU 标志后, 给 TCFU 标志写“0”时
在 MTU0、3、4、6、7 中, 此标志为保留位。读写值都为“0”。

TCFD 标志 (计数方向标志)

这是表示 MTU1 ~ 4、6、7 的计数方向的状态标志。

在 MTU0 中, 此标志为保留位。读写值都为“1”。

- TSR2 (MTU0)**

地址 MTU0.TSR2 000C 1325h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	TGFF	TGFE

复位后的值 1 1 0 0 0 0 0 0

位	符号	位名	功能	R/W
b0	TGFE	比较匹配标志 E	0: 未发生 MTU0.TCNT 和 MTU0.TGRE 的比较匹配 1: 发生 MTU0.TCNT 和 MTU0.TGRE 的比较匹配	R/(W) (注 1)
b1	TGFF	比较匹配标志 F	0: 未发生 MTU0.TCNT 和 MTU0.TGRF 的比较匹配 1: 发生 MTU0.TCNT 和 MTU0.TGRF 的比较匹配	R/(W) (注 1)
b5-b2	—	保留位	读写值都为“0”。	R/W
b7-b6	—	保留位	读写值都为“1”。	R/W

注 1. 为了将标志置“0”, 只能在读“1”后写“0”。

TGFE 标志

这是表示发生 MTU0.TCNT 和 MTU0.TGRE 的比较匹配的状态标志。

[为“1”的条件]

- 在 TRGE 寄存器用作比较匹配寄存器的情况下，MTU0.TCNT=MTU0.TGRE 时

[为“0”的条件]

- 在 TGFE=1 的状态下读 TGFE 标志后，给 TGFE 标志写“0”时

TGFF 标志

这是表示发生 MTU0.TCNT 和 MTU0.TGRF 的比较匹配的状态标志。

[为“1”的条件]

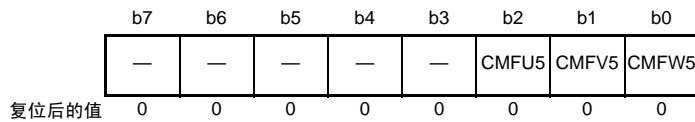
- 在 TRGF 寄存器用作比较匹配寄存器的情况下，MTU0.TCNT=MTU0.TGRF 时

[为“0”的条件]

- 在 TGFF=1 的状态下读 TGFF 标志后，给 TGFF 标志写“0”时

• **TSR (MTU5)**

地址 MTU5.TSR 000C 1CB0h



位	符号	位名	功能	R/W
b0	CMFW5	比较匹配 / 输入捕捉标志 W5	0: 未发生 MTU5.TGRW 的输入捕捉或者比较匹配 1: 发生 MTU5.TGRW 的输入捕捉或者比较匹配 (注 1)	R/(W)
b1	CMFV5	比较匹配 / 输入捕捉标志 V5	0: 未发生 MTU5.TGRV 的输入捕捉或者比较匹配 1: 发生 MTU5.TGRV 的输入捕捉或者比较匹配 (注 1)	R/(W)
b2	CMFU5	比较匹配 / 输入捕捉标志 U5	0: 未发生 MTU5.TGRU 的输入捕捉或者比较匹配 1: 发生 MTU5.TGRU 的输入捕捉或者比较匹配 (注 1)	R/(W)
b7-b3	—	保留位	读写值都为“0”。	R/W

注 1. 为了将标志置“0”，只能在读“1”后写“0”。

CMFn5 标志 (比较匹配 / 输入捕捉标志 n5) (n=U、V、W)

这是表示发生 MTU5.TGRn 寄存器的输入捕捉或者比较匹配的状态标志。为了将标志置“0”，只能写“0”。

[为“1”的条件]

- 在 MTU5.TGRn 寄存器用作比较匹配寄存器的情况下，MTU5.TCNTn=MTU5.TGRn 时
- 在 MTU5.TGRn 寄存器用作输入捕捉寄存器的情况下，通过输入捕捉信号将 MTU5.TCNTn 的值传送到 MTU5.TGRn 时
- 在 MTU5.TGRn 寄存器用作外部输入信号的脉宽测量的情况下，将 MTU5.TCNTn 的值传送到 MTU5.TGRn 时 (注 1)

[为“0”的条件]

- 在 CMFn5=1 的状态下读 CMFn5 标志后，给 CMFn5 标志写“0”时

注 1. 通过 MTU5.TIORU、MTU5.TIORV、MTU5.TIORW 寄存器的 IOC[4:0]位设定传送时序。

21.2.8 定时器的缓冲运行传送模式寄存器 (TBTM)

- MTU0.TBTM

地址 MTU0.TBTM 000C 1326h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TTSE	TTSB	TTSA
复位后的值	0	0	0	0	0	0	0	0

- MTU3.TBTM、MTU4.TBTM、MTU6.TBTM、MTU7.TBTM

地址 MTU3.TBTM 000C 1238h、MTU4.TBTM 000C 1239h、MTU6.TBTM 000C 1A38h、MTU7.TBTM 000C 1A39h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TTSB	TTSA
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	TTSA	时序选择 A 位	0: 在发生各通道的比较匹配 A 时从 TGRC 传送到 TGRA 1: 在清除各通道的 TCNT 时从 TGRC 传送到 TGRA	R/W
b1	TTSB	时序选择 B 位	0: 在发生各通道的比较匹配 B 时从 TGRD 传送到 TGRB 1: 在清除各通道的 TCNT 时从 TGRD 传送到 TGRB	R/W
b2	TTSE	时序选择 E 位	0: 在发生 MTU0 的比较匹配 E 时从 MTU0.TGRF 传送到 MTU0.TGRE 1: 在清除 MTU0.TCNT 时从 MTU0.TGRF 传送到 MTU0.TGRE	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

TBTM 寄存器设定在 PWM 模式中从缓冲寄存器到定时器的通用寄存器的传送时序。MTU0、3、4、6、7 各有 1 个 TBTM，共计 5 个。

TTSA 位 (时序选择 A 位)

此位设定在各通道进行缓冲运行时从 TGRC 到 TGRA 的传送时序。在用于其他模式 (非 PWM 模式) 的通道中，不能将 TTSA 位置“1”。

TTSB 位 (时序选择 B 位)

此位设定在各通道进行缓冲运行时从 TGRD 到 TGRB 的传送时序。在用于其他模式 (非 PWM 模式) 的通道中，不能将 TTSB 位置“1”。

TTSE 位 (时序选择 E 位)

此位设定在进行缓冲运行时从 MTU0.TGRF 到 MTU0.TGRE 的传送时序。在 MTU3、4、6、7 中，此位为保留位，读写值都为“0”。在用于其他模式 (非 PWM 模式) 的通道中，不能将 TTSE 位置“1”。

21.2.9 定时器的输入捕捉控制寄存器 (TICCR)

地址 MTU1.TICCR 000C 1390h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	I2BE	I2AE	I1BE	I1AE
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	I1AE	输入捕捉允许位	0: 不将 MTIOC1A 引脚追加到 MTU2.TGRA 的输入捕捉条件 1: 将 MTIOC1A 引脚追加到 MTU2.TGRA 的输入捕捉条件	R/W
b1	I1BE	输入捕捉允许位	0: 不将 MTIOC1B 引脚追加到 MTU2.TGRB 的输入捕捉条件 1: 将 MTIOC1B 引脚追加到 MTU2.TGRB 的输入捕捉条件	R/W
b2	I2AE	输入捕捉允许位	0: 不将 MTIOC2A 引脚追加到 MTU1.TGRA 的输入捕捉条件 1: 将 MTIOC2A 引脚追加到 MTU1.TGRA 的输入捕捉条件	R/W
b3	I2BE	输入捕捉允许位	0: 不将 MTIOC2B 引脚追加到 MTU1.TGRB 的输入捕捉条件 1: 将 MTIOC2B 引脚追加到 MTU1.TGRB 的输入捕捉条件	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

TICCR 寄存器控制 MTU1.TCNT 和 MTU2.TCNT 级联时的输入捕捉条件。MTU1 有 1 个 TICCR。

I1AE 位 (输入捕捉允许位)

此位选择是否将 MTIOC1A 引脚追加到 MTU2.TGRA 的输入捕捉条件。

I1BE 位 (输入捕捉允许位)

此位选择是否将 MTIOC1B 引脚追加到 MTU2.TGRB 的输入捕捉条件。

I2AE 位 (输入捕捉允许位)

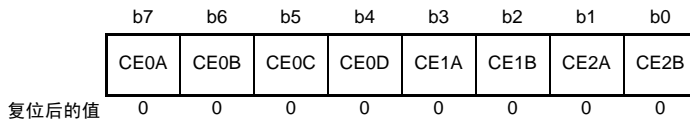
此位选择是否将 MTIOC2A 引脚追加到 MTU1.TGRA 的输入捕捉条件。

I2BE 位 (输入捕捉允许位)

此位选择是否将 MTIOC2B 引脚追加到 MTU1.TGRB 的输入捕捉条件。

21.2.10 定时器的同步清除寄存器 (TSYCR)

地址 MTU6.TSYCR 000C 1A50h



位	符号	位名	功能	R/W
b0	CE2B	清除允许 2B 位	0: 禁止在 MTU2.TSR 的 TGFB 标志置位时清除 1: 允许在 MTU2.TSR 的 TGFB 标志置位时清除	R/W
b1	CE2A	清除允许 2A 位	0: 禁止在 MTU2.TSR 的 TGFA 标志置位时清除 1: 允许在 MTU2.TSR 的 TGFA 标志置位时清除	R/W
b2	CE1B	清除允许 1B 位	0: 禁止在 MTU1.TSR 的 TGFB 标志置位时清除 1: 允许在 MTU1.TSR 的 TGFB 标志置位时清除	R/W
b3	CE1A	清除允许 1A 位	0: 禁止在 MTU1.TSR 的 TGFA 标志置位时清除 1: 允许在 MTU1.TSR 的 TGFA 标志置位时清除	R/W
b4	CE0D	清除允许 0D 位	0: 禁止在 MTU0.TSR 的 TGFD 标志置位时清除 1: 允许在 MTU0.TSR 的 TGFD 标志置位时清除	R/W
b5	CE0C	清除允许 0C 位	0: 禁止在 MTU0.TSR 的 TGFC 标志置位时清除 1: 允许在 MTU0.TSR 的 TGFC 标志置位时清除	R/W
b6	CE0B	清除允许 0B 位	0: 禁止在 MTU0.TSR 的 TGFB 标志置位时清除 1: 允许在 MTU0.TSR 的 TGFB 标志置位时清除	R/W
b7	CE0A	清除允许 0A 位	0: 禁止在 MTU0.TSR 的 TGFA 标志置位时清除 1: 允许在 MTU0.TSR 的 TGFA 标志置位时清除	R/W

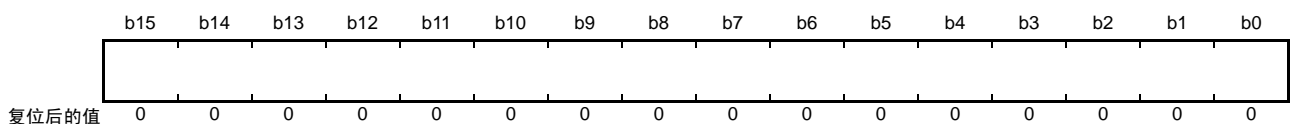
TSYCR 寄存器设定 MTU 的 MTU6.TCNT 和 MTU7.TCNT 的同步清除条件。MTU6 有一个 TSYCR。

CE_nm 位 (清除允许 nm 位) (n=0、1、2, m=A、B、C、D)

此位设定允许或者禁止在 MTU_n.TSR 的 TGF_m 标志置位时清除。

21.2.11 定时器的计数器 (TCNT)

地址 MTU0.TCNT 000C 1306h、MTU1.TCNT 000C 1386h、MTU2.TCNT 000C 1406h、MTU3.TCNT 000C 1210h、MTU4.TCNT 000C 1212h、MTU5.TCNTU 000C 1C80h、MTU5.TCNTV 000C 1C90h、MTU5.TCNTW 000C 1CA0h、MTU6.TCNT 000C 1A10h、MTU7.TCNT 000C 1A12h



注：禁止以 8 位为单位存取 TCNT。必须以 16 位为单位进行存取。

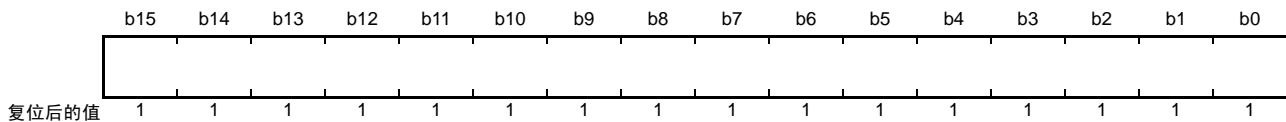
TCNT 寄存器是 16 位可读写计数器。MTU0 ~ 4、6、7 各有 1 个 TCNT，MTU5 有 3 个 MTU5.TCNTU/V/W，共计 10 个。

在复位时，TCNT 被初始化为“0000h”。

禁止以 8 位为单位存取 TCNT。必须以 16 位为单位进行存取。

21.2.12 定时器的通用寄存器 (TGR)

地址 MTU0.TGRA 000C 1308h、MTU0.TGRB 000C 130Ah、MTU0.TGRC 000C 130Ch、MTU0.TGRD 000C 130Eh、MTU0.TGRE 000C 1320h、MTU0.TGRF 000C 1322h、MTU1.TGRA 000C 1388h、MTU1.TGRB 000C 138Ah、MTU2.TGRA 000C 1408h、MTU2.TGRB 000C 140Ah、MTU3.TGRA 000C 1218h、MTU3.TGRB 000C 121Ah、MTU3.TGRC 000C 1224h、MTU3.TGRD 000C 1226h、MTU3.TGRE 000C 1272h、MTU4.TGRA 000C 121Ch、MTU4.TGRB 000C 121Eh、MTU4.TGRC 000C 1228h、MTU4.TGRD 000C 122Ah、MTU4.TGRE 000C 1274h、MTU4.TGRF 000C 1276h、MTU5.TGRU 000C 1C82h、MTU5.TGRV 000C 1C92h、TU5.TGRW 000C 1CA2h、MTU6.TGRA 000C 1A18h、MTU6.TGRB 000C 1A1Ah、MTU6.TGRC 000C 1A24h、MTU6.TGRD 000C 1A26h、MTU6.TGRE 000C 1A72h、MTU7.TGRA 000C 1A1Ch、MTU7.TGRB 000C 1A1Eh、MTU7.TGRC 000C 1A28h、MTU7.TGRD 000C 1A2Ah、MTU7.TGRE 000C 1A74h、MTU7.TGRF 000C 1A76h



注. 禁止以 8 位为单位存取 TGR。必须以 16 位为单位进行存取。TGR 的初始值为“FFFFh”。

TGR 寄存器是 16 位可读写寄存器。MTU0 有 6 个 TGR，MTU1 和 MTU2 各有 2 个 TGR，MTU3 和 MTU6 各有 5 个 TGR，MTU4 和 MTU7 各有 6 个 TGR，MTU5 有 3 个 TGR，共计 35 个。

TGRA、TGRB、TGRC 和 TGRD 寄存器是输出比较 / 输入捕捉兼用的寄存器。能将 MTU0、3、4、6、7 的 TGRC 和 TGRD 寄存器设定为缓冲寄存器。TGR 寄存器和缓冲寄存器的组合为 TGRA-TGRC、TGRB-TGRD。

当 MTU0.TGRE 和 MTU0.TGRF 用作比较寄存器，并且 MTU0.TCNT 和 MTU0.TGRE 匹配时，能产生 A/D 转换开始请求。能将 TGRF 寄存器设定为缓冲寄存器。TGR 寄存器和缓冲寄存器的组合为 TGRE-TGRF。

MTU5.TGRU、MTU5.TGRV 和 MTU5.TGRW 是比较匹配 / 输入捕捉 / 外部脉宽测量兼用的寄存器。

21.2.13 定时器的启动寄存器 (TSTR)

- TSTRA (MTU0、MTU1、MTU2、MTU3、MTU4)

地址 MTU.TSTRA 000C 1280h

	b7	b6	b5	b4	b3	b2	b1	b0
	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	计数开始 0 位	0: MTU0.TCNT 停止计数 1: MTU0.TCNT 计数	R/W
b1	CST1	计数开始 1 位	0: MTU1.TCNT 停止计数 1: MTU1.TCNT 计数	R/W
b2	CST2	计数开始 2 位	0: MTU2.TCNT 停止计数 1: MTU2.TCNT 计数	R/W
b5-b3	—	保留位	读写值都为“0”。	R/W
b6	CST3	计数开始 3 位	0: MTU3.TCNT 停止计数 1: MTU3.TCNT 计数	R/W
b7	CST4	计数开始 4 位	0: MTU4.TCNT 停止计数 1: MTU4.TCNT 计数	R/W

注. 将 TCSYSTR 寄存器置“1”时，TSTRA 寄存器的对应位就自动变为“1”。

TSTRA 寄存器选择 MTU0 ~ MTU4 的 TCNT 计数或者停止计数。

TSTRB 寄存器选择 MTU6 和 MTU7 的 TCNT 计数或者停止计数。

TSTR 寄存器选择 MTU5 的 TCNT 计数或者停止计数。

只有在 TCNT 停止计数后，才能给 TMDR1 寄存器设定运行模式或者给 TCR 寄存器设定 TCNT 的计数时钟。

CSTn 位 (计数开始 n 位) (n=0、1、2、3、4)

这些位选择各通道的 TCNT 计数或者停止计数。

MTIOC 引脚在输出状态下运行时，如果给 CSTn 位写“0”，计数器就停止计数，但是保持 MTIOC 引脚的输出比较的输出电平。如果在 CSTn 位为“0”的状态下写 TIOR 寄存器，就将引脚的输出电平更新为所设定的初始输出值。

- TSTRB (MTU6、MTU7)

地址 MTU.TSTRB 000C 1A80h

	b7	b6	b5	b4	b3	b2	b1	b0
	CST7	CST6	—	—	—	—	—	—
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	CST6	计数开始 6 位	0: MTU6.TCNT 停止计数 1: MTU6.TCNT 计数	R/W
b7	CST7	计数开始 7 位	0: MTU7.TCNT 停止计数 1: MTU7.TCNT 计数	R/W

注. 将 TCSYSTR 寄存器置“1”时，TSTRB 寄存器的对应位就自动变为“1”。

CSTn 位 (计数开始 n 位) (n=6、7)

这些位选择各通道的 TCNT 计数或者停止计数。

MTIOC 引脚在输出状态下运行时，如果给 CSTn 位写“0”，计数器就停止计数，但是保持 MTIOC 引脚的输出比较的输出电平。如果在 CSTn 位为“0”的状态下写 TIOR 寄存器，就将引脚的输出电平更新为所设定的初始输出值。

- TSTR (MTU5)

地址 MTU5.TSTR 000C 1CB4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSTU5	CSTV5	CSTW5
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	CSTW5	计数开始 W5 位	0: MTU5.TCNTW 停止计数 1: MTU5.TCNTW 计数	R/W
b1	CSTV5	计数开始 V5 位	0: MTU5.TCNTV 停止计数 1: MTU5.TCNTV 计数	R/W
b2	CSTU5	计数开始 U5 位	0: MTU5.TCNTU 停止计数 1: MTU5.TCNTU 计数	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

CSTn5 位 (计数开始 n5 位) (n=U、V、W)

这些位选择各 TCNT 计数或者停止计数。

21.2.14 定时器的同步寄存器 (TSYR)

- TSYRA (MTU0、MTU1、MTU2、MTU3、MTU4)

地址 MTU.TSYRA 000C 1281h

	b7	b6	b5	b4	b3	b2	b1	b0
	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: MTU0.TCNT 独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: MTU0.TCNT 进行同步运行 (能进行 TCNT 的同步预置 / 同步清除)	R/W
b1	SYNC1	定时器的同步 1 位	0: MTU1.TCNT 独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: MTU1.TCNT 进行同步运行 (能进行 TCNT 的同步预置 / 同步清除)	R/W
b2	SYNC2	定时器的同步 2 位	0: MTU2.TCNT 独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: MTU2.TCNT 进行同步运行 (能进行 TCNT 的同步预置 / 同步清除)	R/W
b5-b3	—	保留位	读写值都为“0”。	R/W
b6	SYNC3	定时器的同步 3 位	0: MTU3.TCNT 独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: MTU3.TCNT 进行同步运行 (能进行 TCNT 的同步预置 / 同步清除)	R/W
b7	SYNC4	定时器的同步 4 位	0: MTU4.TCNT 独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: MTU4.TCNT 进行同步运行 (能进行 TCNT 的同步预置 / 同步清除)	R/W

TSYRA 寄存器选择 MTU0 ~ 4 的 TCNT 进行独立运行或者同步运行。

TSYRB 寄存器选择 MTU6 和 MTU7 的 TCNT 进行独立运行或者同步运行。

对应位为“1”的通道进行同步运行。

SYNCn 位 (定时器的同步 n 位) (n=0、1、2、3、4)

这些位选择 TCNT 是与其他通道是独立运行还是同步运行。

如果选择同步运行, 就能对多个 TCNT 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。

要设定同步运行时, 需要至少将 2 个通道的 SYNC 位置“1”; 要设定同步清除时, 除了 SYNC 位以外, 还需要通过 TCR.CCLR[2:0] 位设定 TCNT 的清除源。

- TSYRB (MTU6、MTU7)

地址 MTU.TSYRB 000C 1A81h

	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC7	SYNC6	—	—	—	—	—	—
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	SYNC6	定时器的同步 6 位	0: MTU6.TCNT 独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: MTU6.TCNT 进行同步运行 (能进行 TCNT 的同步预置 / 同步清除)	R/W
b7	SYNC7	定时器的同步 7 位	0: MTU7.TCNT 独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: MTU7.TCNT 进行同步运行 (能进行 TCNT 的同步预置 / 同步清除)	R/W

SYNCn 位 (定时器的同步 n 位) (n=6、7)

这些位选择 TCNT 是与其他通道是独立运行还是同步运行。

如果选择同步运行, 就能对多个 TCNT 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。

要设定同步运行时, 需要至少将 2 个通道的 SYNC 位置“1”; 要设定同步清除时, 除了 SYNC 位以外, 还需要通过 TCR.CCLR[2:0] 位设定 TCNT 的清除源。

21.2.15 定时器的计数器同步启动寄存器 (TCSYSTR)

地址 MTU.TCSYSTR 000C 1282h

b7	b6	b5	b4	b3	b2	b1	b0
SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH6	SCH7
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	SCH7	同步启动 7 位	0: 不同步启动 MTU7.TCNT 1: 同步启动 MTU7.TCNT	R/(W) (注 1)
b1	SCH6	同步启动 6 位	0: 不同步启动 MTU6.TCNT 1: 同步启动 MTU6.TCNT	R/(W) (注 1)
b2	—	保留位	读写值都为“0”。	R
b3	SCH4	同步启动 4 位	0: 不同步启动 MTU4.TCNT 1: 同步启动 MTU4.TCNT	R/(W) (注 1)
b4	SCH3	同步启动 3 位	0: 不同步启动 MTU3.TCNT 1: 同步启动 MTU3.TCNT	R/(W) (注 1)
b5	SCH2	同步启动 2 位	0: 不同步启动 MTU2.TCNT 1: 同步启动 MTU2.TCNT	R/(W) (注 1)
b6	SCH1	同步启动 1 位	0: 不同步启动 MTU1.TCNT 1: 同步启动 MTU1.TCNT	R/(W) (注 1)
b7	SCH0	同步启动 0 位	0: 不同步启动 MTU0.TCNT 1: 同步启动 MTU0.TCNT	R/(W) (注 1)

注 1 为了将标志置“1”，只能写“1”。TCSYSTR 寄存器在写“1”后自动变为“0”。

TCSYSTR 寄存器进行计数器的同步启动。

SCH7 位 (同步启动 7 位)

此位控制 MTU7.TCNT 的同步启动。

[为“0”的条件]

- 在 SCH7 位为“1”的状态下将 TSTRA.CST7 位置“1”时

SCH6 位 (同步启动 6 位)

此位控制 MTU6.TCNT 的同步启动。

[为“0”的条件]

- 在 SCH6 位为“1”的状态下将 TSTRA.CST6 位置“1”时

SCH4 位 (同步启动 4 位)

此位控制 MTU4.TCNT 的同步启动。

[为“0”的条件]

- 在 SCH4 位为“1”的状态下将 TSTRA.CST4 位置“1”时

SCH3 位 (同步启动 3 位)

此位控制 MTU3.TCNT 的同步启动。

[为 “0” 的条件]

- 在 SCH3 位为 “1” 的状态下将 TSTRA.CST3 位置 “1” 时

SCH2 位 (同步启动 2 位)

此位控制 MTU2.TCNT 的同步启动。

[为 “0” 的条件]

- 在 SCH2 位为 “1” 的状态下将 TSTRA.CST2 位置 “1” 时

SCH1 位 (同步启动 1 位)

此位控制 MTU1.TCNT 的同步启动。

[为 “0” 的条件]

- 在 SCH1 位为 “1” 的状态下将 TSTRA.CST1 位置 “1” 时

SCH0 位 (同步启动 0 位)

此位控制 MTU0.TCNT 的同步启动。

[为 “0” 的条件]

- 在 SCH0 位为 “1” 的状态下将 TSTRA.CST0 位置 “1” 时

21.2.16 定时器的读写允许寄存器 (TRWERA、TRWERB)

地址 MTU.TRWERA 000C 1284h、MTU.TRWERB 000C 1A84h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	RWE
0	0	0	0	0	0	0	1

复位后的值

位	符号	位名	功能	R/W
b0	RWE	读写允许位	0: 禁止读写寄存器 1: 允许读写寄存器	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

TRWERA 寄存器设定允许或者禁止存取 MTU3 和 MTU4 的误写防止对象寄存器 / 计数器。
TRWERB 寄存器设定允许或者禁止存取 MTU6 和 MTU7 的误写防止对象寄存器 / 计数器。

RWE 位 (读写允许位)

此位设定允许或者禁止读写误写防止寄存器。

[为“0”的条件]

- 在 RWE 为“1”的状态下读 RWE 后，给 RWE 位写“0”时
- 误写防止的对象寄存器和对象计数器 (TRWERA)
MTUn.TCR、MTUn.TMDR1、MTUn.TIORH、MTUn.TIORL、MTUn.TIER、MTUn.TGRA、MTUn.TGRB、MTU.TOERA、MTU.TOCR1A、MTU.TOCR2A、MTU.TGCRA、MTU.TCDRA、MTU.TDDRA 和 MTUn.TCNT，共计 22 个寄存器 (n=3、4)。
- 误写防止的对象寄存器和对象计数器 (TRWERB)
MTUn.TCR、MTUn.TMDR1、MTUn.TIORH、MTUn.TIORL、MTUn.TIER、MTUn.TGRA、MTUn.TGRB、MTU.TOERB、MTU.TOCR1B、MTU.TOCR2B、MTU.TCDRB、MTU.TDDRB 和 MTUn.TCNT，共计 21 个寄存器 (n=6、7)。

21.2.17 定时器的输出主控允许寄存器 (TOER)

• TOERA

地址 MTU.TOERA 000C 120Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
复位后的值	1	1	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	OE3B	主控允许 MTIOC3B 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b1	OE4A	主控允许 MTIOC4A 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b2	OE4B	主控允许 MTIOC4B 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b3	OE3D	主控允许 MTIOC3D 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b4	OE4C	主控允许 MTIOC4C 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b5	OE4D	主控允许 MTIOC4D 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b7-b6	—	保留位	读写值都为“1”。	R/W

TOERA 寄存器允许或者禁止输出引脚的 MTIOC4D、MTIOC4C、MTIOC3D、MTIOC4B、MTIOC4A、MTIOC3B 的输出设定。

TOERB 寄存器允许或者禁止输出引脚的 MTIOC7D、MTIOC7C、MTIOC6D、MTIOC7B、MTIOC7A、MTIOC6B 的输出设定。

如果不设定 TOER 寄存器的各位，就无法正确地输出这些引脚。在 MTU3、4、6、7 中，必须在设定 MTU3、4、6、7 的 TIOR 寄存器前给 TOER 寄存器设定值。

必须在将 MTU.TSTRA 寄存器的 CST3 位和 CST4 位置“0”后才能设定 MTU.TOERA 寄存器。

必须在将 MTU.TSTRB 寄存器的 CST0 位和 CST1 位置“0”后才能设定 MTU.TOERB 寄存器（参照图 21.36 和图 21.39）。

OEnm 位（主控允许 MTIOCnm 位）（n=3、4，m=A～D）

这些位允许或者禁止 MTU 的 MTIOCnm 引脚输出。

- TOERB

地址 MTU.TOERB 000C 1A0Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OE7D	OE7C	OE6D	OE7B	OE7A	OE6B
复位后的值	1	1	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	OE6B	主控允许 MTIOC6B 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b1	OE7A	主控允许 MTIOC7A 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b2	OE7B	主控允许 MTIOC7B 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b3	OE36D	主控允许 MTIOC6D 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b4	OE7C	主控允许 MTIOC7C 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b5	OE7D	主控允许 MTIOC7D 位	0: 禁止 MTU 输出 1: 允许 MTU 输出	R/W
b7-b6	—	保留位	读写值都为“1”。	R/W

OEnm 位 (主控允许 MTIOCnm 位) (n=6、7, m=A ~ D)

这些位允许或者禁止 MTU 的 MTIOCnm 引脚输出。

21.2.18 定时器的输出控制寄存器 1 (TOCR1A、TOCR1B)

地址 MTU.TOCR1A 000C 120Eh、MTU.TOCR1B 000C 1A0Eh

b7	b6	b5	b4	b3	b2	b1	b0
—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
0	0	0	0	0 (注4)	0	0	0

位	符号	位名	功能	R/W
b0	OLSP	输出电平选择 P 位 (注1、注3)	请参照表 21.37。	R/W
b1	OLSN	输出电平选择 N 位 (注1、注3)	请参照表 21.38。	R/W
b2	TOCS	TOC 选择位	0: TOCR1j 的设定有效 (j=A、B) 1: TOCR2j 的设定有效	R/W
b3	TOCL	TOC 寄存器的写禁止位 (注2)	0: 允许写 TOCS 位、OLSN 位和 OLSP 位 1: 禁止写 TOCS 位、OLSN 位和 OLSP 位	R/W
b5-b4	—	保留位	读写值都为“0”。	R/W
b6	PSYE	PWM 同步输出允许位	0: 禁止交替输出 1: 允许交替输出	R/W
b7	—	保留位	读写值都为“0”。	R/W

注 1. 通过将 TOCR1j.TOCS 位置“0”，使此设定变为有效。

注 2. 能通过将 TOCR1j.TOCL 位置“1”，防止 CPU 失控时的误写。

注 3. 在不生成死区时间时，反相输出总是为正相的相反电平。此时，只有 OLSP 位有效。

注 4. 复位后，只能写 1 次“1”并且不能在写“1”后写“0”。

TOCR1A 寄存器和 TOCR1B 寄存器允许或者禁止与互补 PWM 模式 / 复位同步 PWM 模式的 PWM 周期同步进行交替输出以及对 PWM 输出的输出电平进行反相控制。

OLSP 位 (输出电平选择 P 位)

在复位同步 PWM 模式 / 互补 PWM 模式中，此位选择正相的输出电平。

OLSN 位 (输出电平选择 N 位)

在复位同步 PWM 模式 / 互补 PWM 模式中，此位选择反相的输出电平。

TOCS 位 (TOC 选择位)

此位选择 TOCR1j 寄存器的设定有效还是 TOCR2j 寄存器 (j=A、B) 的设定有效。TOCR1j 寄存器和 TOCR2j 寄存器 (j=A、B) 用于设定互补 PWM 模式 / 复位同步 PWM 模式的输出电平。

TOCL 位 (TOC 寄存器的写禁止位)

此位设定允许或者禁止写 TOCR1j 寄存器 (j=A、B) 的 TOCS 位、OLSN 位和 OLSP 位。

PSYE 位 (PWM 同步输出允许位)

此位设定允许或者禁止与 PWM 周期同步进行交替输出。

表 21.37 输出电平的选择功能

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

表 21.38 输出电平的选择功能

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

注. 在开始计数并且经过死区时间后，反相波形的初始输出值变为有效电平。

OLSN 为“1”并且 OLSP 为“1”时的互补 PWM 模式的输出例子（1 相）如图 21.3 所示。

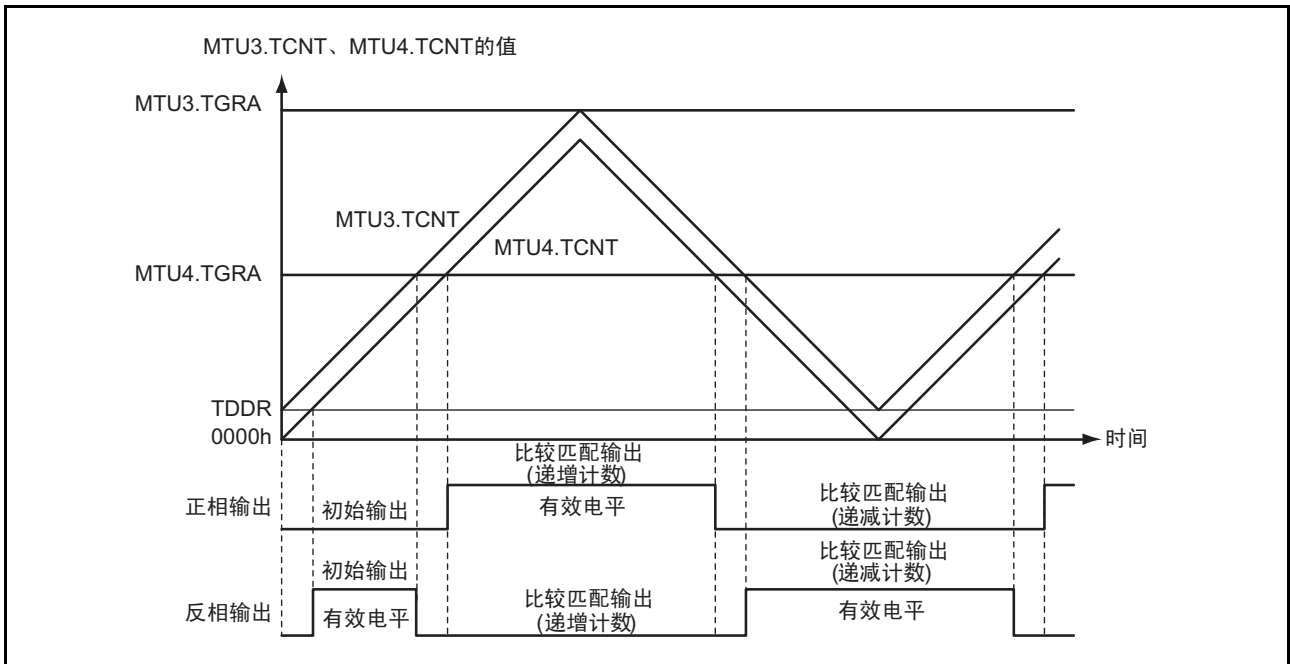
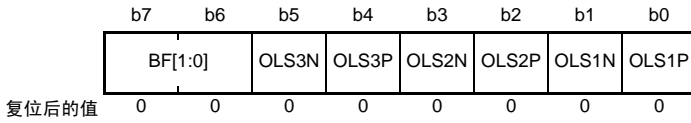


图 21.3 互补 PWM 模式的输出电平例子

21.2.19 定时器的输出控制寄存器 2 (TOCR2A、TOCR2B)

地址 MTU.TOCR2A 000C 120Fh、MTU.TOCR2B 000C 1A0Fh



位	符号	位名	功能	R/W
b0	OLS1P	输出电平选择 1P 位 (注 1、注 2)	选择在复位同步 PWM 模式 / 互补 PWM 模式中的 MTIOC3B/MTIOC6B 的输出电平, 请参照表 21.39。	R/W
b1	OLS1N	输出电平选择 1N 位 (注 1、注 2)	选择在复位同步 PWM 模式 / 互补 PWM 模式中的 MTIOC3D/MTIOC6D 的输出电平, 请参照表 21.40。	R/W
b2	OLS2P	输出电平选择 2P 位 (注 1、注 2)	选择在复位同步 PWM 模式 / 互补 PWM 模式中的 MTIOC4A/MTIOC7A 的输出电平, 请参照表 21.41。	R/W
b3	OLS2N	输出电平选择 2N 位 (注 1、注 2)	选择在复位同步 PWM 模式 / 互补 PWM 模式中的 MTIOC4C/MTIOC7C 的输出电平, 请参照表 21.42。	R/W
b4	OLS3P	输出电平选择 3P 位 (注 1、注 2)	选择在复位同步 PWM 模式 / 互补 PWM 模式中的 MTIOC4B/MTIOC7B 的输出电平, 请参照表 21.43。	R/W
b5	OLS3N	输出电平选择 3N 位 (注 1、注 2)	选择在复位同步 PWM 模式 / 互补 PWM 模式中的 MTIOC4D/MTIOC7D 的输出电平, 请参照表 21.44。	R/W
b7-b6	BF[1:0]	TOLBR 缓冲传送时序选择位	选择从 TOLBRj 到 TOCR2j 的缓冲传送时序, 详细内容请参照表 21.45。	R/W

j=A、B

注 1. 通过将 TOCR1j.TOCS 位置“1”, 使此设定变为有效。

注 2. 在不生成死区时间时, 反相输出总是为正相的相反电平。此时, 只有 OLSiP 位有效。(i=1、2、3)

TOCR2A 寄存器和 TOCR2B 寄存器对互补 PWM 模式 / 复位同步 PWM 模式中的 PWM 输出的输出电平进行反相控制。

表 21.39 MTIOCmB 输出电平的选择功能

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

m=3、6

表 21.40 MTIOCmD 输出电平的选择功能

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

m=3、6

注. 在开始计数并且经过死区时间后, 反相波形的初始输出值变为有效电平。

表 21.41 MTIOc mA 输出电平的选择功能

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

m=4、7

表 21.42 MTIOc mC 输出电平的选择功能

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

m=4、7

注. 在开始计数并且经过死区时间后, 反相波形的初始输出值变为有效电平。

表 21.43 MTIOc mB 输出电平的选择功能

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

m=4、7

表 21.44 MTIOc mD 输出电平的选择功能

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

m=4、7

注. 在开始计数并且经过死区时间后, 反相波形的初始输出值变为有效电平。

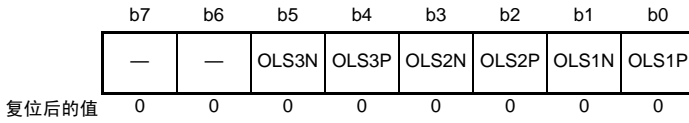
表 21.45 TOCR2j.BF[1:0] 位的设定

bit7	bit6	说明	
BF1	BF0	互补 PWM 模式	复位 PWM 模式
0	0	不从缓冲寄存器 (TOLBRj) 传送到 TOCR2j。	不从缓冲寄存器 (TOLBRj) 传送到 TOCR2j。
0	1	在 MTUn.TCNT 的波峰从缓冲寄存器 (TOLBRj) 传送到 TOCR2j。	在清除 MTUm.TCNT 计数器和 MTUn.TCNT 计数器时从缓冲寄存器 (TOLBRj) 传送到 TOCR2j。
1	0	在 MTUn.TCNT 的波谷从缓冲寄存器 (TOLBRj) 传送到 TOCR2j。	不能设定。
1	1	在 MTUn.TCNT 的波峰和波谷从缓冲寄存器 (TOLBRj) 传送到 TOCR2j。	不能设定。

n=4、7, m=3、6, j=A、B

21.2.20 定时器的输出电平缓冲寄存器 (TOLBRA、TOLBRB)

地址 MTU.TOLBRA 000C 1236h、MTU.TOLBRB 000C 1A36h



位	符号	位名	功能	R/W
b0	OLS1P	输出电平选择 1P 位	必须给 TOCR2j 的 OLS1P 位设定缓冲传送的值。	R/W
b1	OLS1N	输出电平选择 1N 位	必须给 TOCR2j 的 OLS1N 位设定缓冲传送的值。	R/W
b2	OLS2P	输出电平选择 2P 位	必须给 TOCR2j 的 OLS2P 位设定缓冲传送的值。	R/W
b3	OLS2N	输出电平选择 2N 位	必须给 TOCR2j 的 OLS2N 位设定缓冲传送的值。	R/W
b4	OLS3P	输出电平选择 3P 位	必须给 TOCR2j 的 OLS3P 位设定缓冲传送的值。	R/W
b5	OLS3N	输出电平选择 3N 位	必须给 TOCR2j 的 OLS3N 位设定缓冲传送的值。	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W

j=A、B

TOLBRA 寄存器和 TOLBRB 寄存器是 TOCR2A 寄存器和 TOCR2B 寄存器的缓冲寄存器，设定互补 PWM 模式 / 复位同步 PWM 模式的 PWM 输出电平。

在缓冲运行中设定 PWM 输出电平时的设定步骤例子如图 21.4 所示。

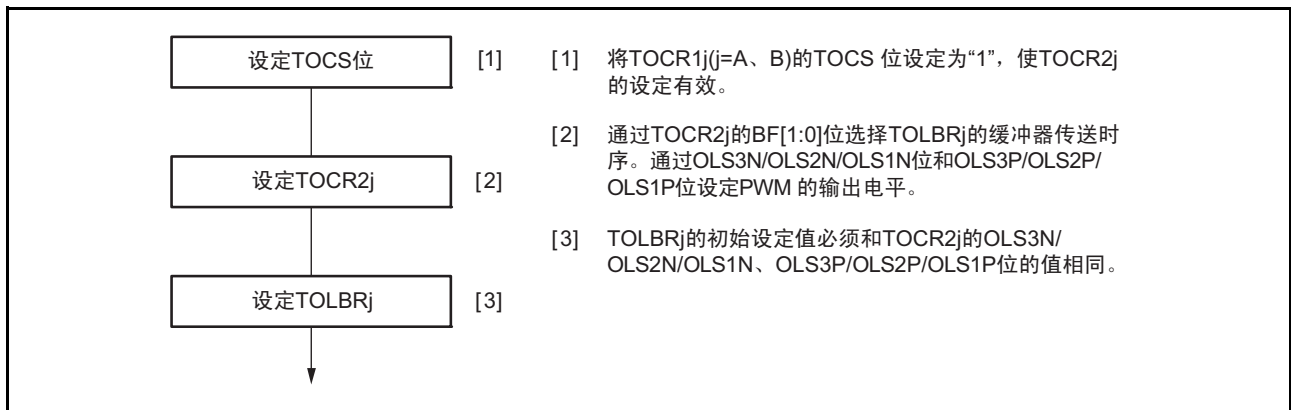


图 21.4 在缓冲运行中设定 PWM 输出电平时的设定步骤例子

21.2.21 定时器的门控寄存器 A (TGCRA)

地址 MTU.TGCRA 000C 120Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BDC	N	P	FB	WF	VF	UF
复位后的值	1	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	UF	输出相转换位	设定正相 / 反相输出相的 ON/OFF。这些位的设定只在 TGCRA.FB 位为“1”时有效。此时，b0 ~ b2 的设定取代外部输入，请参照表 21.46。	R/W
b1	VF			R/W
b2	WF			R/W
b3	FB	外部反馈信号允许位	0: 通过外部输入（输入源为 MTU0 的 TGRA、TGRB、TGRC 的输入捕捉信号）进行输出转换 1: 通过软件进行输出转换（TGCRA 的 UF、VF 和 WF 的设定值）	R/W
b4	P	正相输出（P）控制位	0: 电平输出 1: 复位同步 PWM/ 互补 PWM 输出	R/W
b5	N	反相输出（N）控制位	0: 电平输出 1: 复位同步 PWM/ 互补 PWM 输出	R/W
b6	BDC	无刷 DC 马达位	0: 正常输出 1: 此寄存器的功能有效	R/W
b7	—	保留位	读写值都为“1”。	R/W

TGCRA 寄存器在复位同步 PWM 模式 / 互补 PWM 模式中对控制无刷 DC 马达所需的波形输出进行控制。除了互补 PWM 模式 / 复位同步 PWM 模式以外，此寄存器的设定无效。

UF 位、VF 位、WF 位（输出相转换位）

这些位设定正相 / 反相输出相的 ON/OFF。

FB 位（外部反馈信号允许位）

此位选择是通过 MTU0 的 TGRA、TGRB 和 TGRC 寄存器的输入捕捉信号自动进行正反相输出的转换，还是通过给 TGCRA 寄存器的 bit2 ~ 0 写“0”或者“1”进行正反相输出的转换。

P 位（正相输出（P）控制位）

在进行正相引脚（MTIOC3B 引脚、MTIOC4A 引脚、MTIOC4B 引脚）输出时，此位选择是进行电平输出还是进行复位同步 PWM/ 互补 PWM 输出。

N 位（反相输出（N）控制位）

在进行反相引脚（MTIOC3D 引脚、MTIOC4C 引脚、MTIOC4D 引脚）输出时，此位选择是进行电平输出还是进行复位同步 PWM/ 互补 PWM 输出。

BDC 位（无刷 DC 马达位）

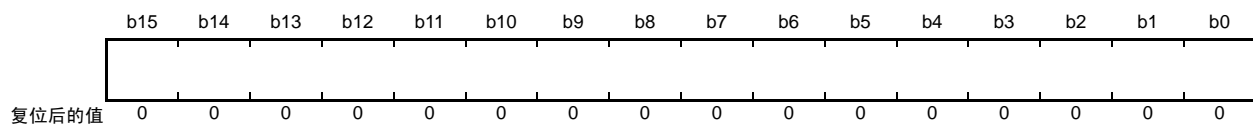
此位选择 TGCRA 寄存器功能的有效或者无效。

表 21.46 输出电平的选择功能

bit2	bit1	bit0	功能					
			MTIOC3B	MTIOC4A	MTIOC4B	MTIOC3D	MTIOC4C	MTIOC4D
WF	VF	UF	U 相	V 相	W 相	U 相	V 相	W 相
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
0	0	1	ON	OFF	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	ON	OFF	OFF
0	1	1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
1	0	1	ON	OFF	OFF	OFF	ON	OFF
1	1	0	OFF	OFF	ON	ON	OFF	OFF
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF

21.2.22 定时器的副计数器 (TCNTSA、TCNTSB)

地址 MTU.TCNTSA 000C 1220h、MTU.TCNTSB 000C 1A20h



注. 禁止以 8 位为单位存取 TCNTSA 寄存器和 TCNTSB 寄存器。必须以 16 位为单位进行存取。

TCNTSA 寄存器和 TCNTSB 寄存器是只用于互补 PWM 模式的 16 位只读计数器。复位后，TCNTSA 寄存器和 TCNTSB 寄存器的值为“0000h”。

21.2.23 定时器的周期数据寄存器 (TCDRA、TCDRB)

地址 MTU.TCDRA 000C 1214h、MTU.TCDRB 000C 1A14h

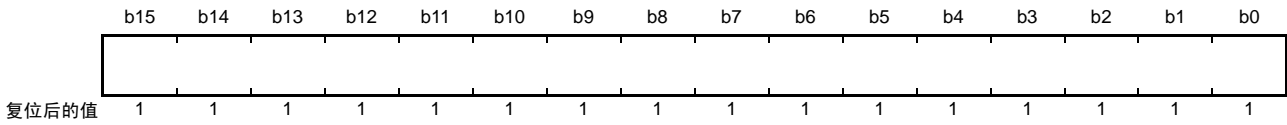


注. 禁止以 8 位为单位存取 TCDRA 寄存器和 TCDRB 寄存器。必须以 16 位为单位进行存取。

TCDRA 寄存器和 TCDRB 寄存器是只用于互补 PWM 模式的 16 位寄存器，必须给 TCDRA 和 TCDRB 寄存器设定 1/2 个 PWM 载波周期的值。复位后，TCDRA 寄存器和 TCDRB 寄存器的值为“FFFFh”。

21.2.24 定时器的周期缓冲寄存器 (TCBRA、TCBRB)

地址 MTU.TCBRA 000C 1222h、MTU.TCBRB 000C 1A22h

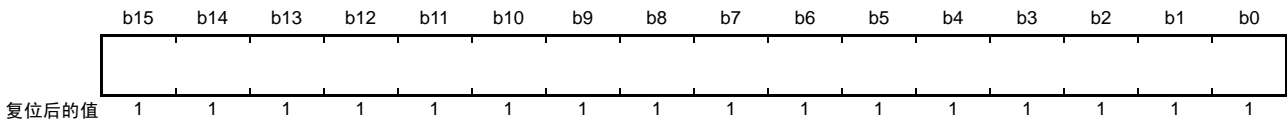


注. 禁止以 8 位为单位存取 TCBRA 寄存器和 TCBRB 寄存器。必须以 16 位为单位进行存取。

TCBRA 寄存器和 TCBRB 寄存器是只用于互补 PWM 模式的 16 位可读写寄存器，用作 TCDRA 寄存器和 TCDRB 寄存器的缓冲寄存器。在 TMDR1 寄存器设定的传送时序将 TCBRA 寄存器和 TCBRB 寄存器的值传送到 TCDRA 寄存器和 TCDRB 寄存器。复位后，TCBRA 寄存器和 TCBRB 寄存器的值为“FFFFh”。

21.2.25 定时器的死区时间数据寄存器 (TDDRA、TDDRb)

地址 MTU.TDDRA 000C 1216h、MTU.TDDRb 000C 1A16h

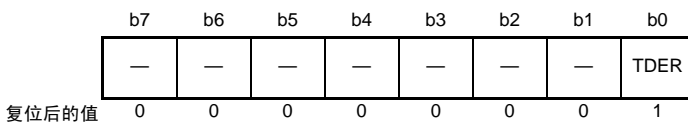


注. 禁止以 8 位为单位存取 TDDRA 寄存器和 TDDRb 寄存器。必须以 16 位为单位进行存取。

TDDRA 寄存器和 TDDRb 寄存器是只用于互补 PWM 模式的 16 可读写位寄存器，在互补 PWM 模式中设定 MTU3.TCNT (MTU6.TCNT) 计数器和 MTU3.TCNT (MTU6.TCNT) 计数器的偏移值。在互补 PWM 模式中清除 MTU3.TCNT (MTU6.TCNT) 计数器和 MTU3.TCNT (MTU6.TCNT) 计数器后重新开始计数时，将 TDDRA (TDDRb) 寄存器的值加载到 MTU3.TCNT (MTU6.TCNT) 计数器并且开始计数。复位后，TDDRA 寄存器和 TDDRb 寄存器的值为“FFFFh”。

21.2.26 定时器的死区时间允许寄存器 (TDERA、TDERB)

地址 MTU3.TDERA 000C 1234h、MTU6.TDERB 000C 1A34h



位	符号	位名	功能	R/W
b0	TDER	死区时间允许位	0: 不生成死区时间 1: 生成死区时间 (注 1)	R/(W)
b7-b1	—	保留位	读写值都为“0”。	R/W

注 1. 必须设定为 TDRRA ≥ 1、TDDRb ≥ 1。

TDERA 寄存器和 TDERB 寄存器控制互补 PWM 模式中的死区时间的生成。MTU3 有 1 个 TDERA，MTU6 有 1 个 TDERB。必须在 TCNT 停止计数的状态下进行 TDERA 寄存器和 TDERB 寄存器的设定。

TDER 位 (死区时间允许位)

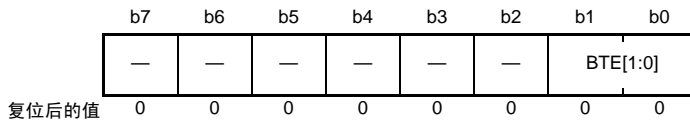
此位设定是否生成死区时间。

[为“0”的条件]

- 在 TDER 为“1”的状态下读 TDER 后，给 TDER 写“0”时

21.2.27 定时器的缓冲传送设定寄存器 (TBTERA、TBTERB)

地址 MTU.TBTERA 000C 1232h、MTU.TBTERB 000C 1A32h



位	符号	位名	功能	R/W
b1-b0	BTE[1:0]	缓冲传送抑制和中断减少联动设定位	此位设定是否抑制用于互补 PWM 模式的缓冲寄存器 (注1) 到暂存器的传送以及是否联动中断减少功能 1, 详细内容请参照表 21.47。	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

注 1. 对象缓冲寄存器 (TBTERA):

MTU3.TGRC、MTU3.TGRD、MTU4.TGRC、MTU4.TGRD、MTU.TCBRA

对象缓冲寄存器 (TBTERB):

MTU6.TGRC、MTU6.TGRD、MTU7.TGRC、MTU7.TGRD、MTU.TCBBR

TBTERA 寄存器和 TBTERB 寄存器设定是否抑制用于互补 PWM 模式的缓冲寄存器到暂存器的传送以及是否联动中断减少功能 1。

表 21.47 TBTER.BTE[1:0] 位的设定

bit1	bit0	说明
BTE1	BTE0	
0	0	不抑制缓冲寄存器到暂存器的传送 (注1), 也不联动中断减少功能 1。
0	1	抑制缓冲寄存器到暂存器的传送。
1	0	缓冲寄存器到暂存器的传送联动中断减少功能 1 (注2)。
1	1	不能设定。

注 1. 根据 TMDR1.MD[3:0] 位的设定进行传送, 详细内容请参照“21.3.8 互补 PWM 模式”。

注 2. 在禁止中断减少功能 1 时 (将定时器的中断减少设定寄存器 (TITCR1A (TITCR1B) 的 T3AEN 位和 T4VEN 位 (T6AEN 位和 T7VEN 位) 置“0”, 或者将 TITCR1A (TITCR1B) 的减少次数设定位 (T3ACOR 和 T4VCOR (T6ACOR 和 T7VCOR)) 置“0”时), 必须设定为缓冲传送不联动中断减少功能 1 (将定时器的缓冲传送寄存器 (TBTERA (TBTERB)) 的 BTE1 位置“0”)。在禁止中断减少功能 1 时, 如果设定为缓冲传送联动中断减少功能 1, 就不进行缓冲传送。

21.2.28 定时器的波形控制寄存器 (TWCRA、TWCRB)

地址 MTU.TWCRA 000C 1260h MTU.TWCRB 000C 1A60h

b7	b6	b5	b4	b3	b2	b1	b0
CCE	—	—	—	—	—	SCC	WRE
复位后的值 0 (注2)	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	WRE	波形保持允许位	0: 输出 TOCR1A、TOCR2A (TOCR1B、TOCR2B) 寄存器设定的初始输出值 1: 抑制初始输出	R/(W)
b1	SCC (注1)	同步清除控制位	(只 TWCRB 寄存器有效) 0: 通过 MTU0、1、2—MTU6、7 同步清除功能, 使 MTU6.TCNT、MTU7.TCNT 的清除有效 0: 通过 MTU0、1、2—MTU6、7 同步清除功能, 使 MTU6.TCNT、MTU7.TCNT 的清除无效	R/(W)
b6-b2	—	保留位	读写值都为“0”。	R/W
b7	CCE (注2)	比较匹配清除允许位	0: 不通过 MTU3.TGRA (MTU6.TGRA) 的比较匹配清除计数器 1: 通过 MTU3.TGRA (MTU6.TGRA) 的比较匹配清除计数器	R/(W)

注 1. 只 TWCRB 寄存器有效。在 TWCRA 中为保留位。

注 2. 除了互补 PWM 模式 1 以外, 不能写“1”。

TWCRA 寄存器和 TWCRB 寄存器控制在互补 PWM 模式中发生 MTU3.TCNT 和 MTU4.TCNT (MTU6.TCNT 和 MTU7.TCNT) 的同步计数器清除时的输出波形, 以及设定是否通过 MTU3.TGRA (MTU6.TGRA) 的比较匹配清除计数器。

必须在 TCNT 停止计数的状态下设定 TWCRA 寄存器和 TWCRB 寄存器的 CCE 位和 WRE 位。

WRE 位 (波形保持允许位)

此位选择在互补 PWM 模式中发生同步计数器清除时的输出波形。

只有在互补 PWM 模式的波谷 Tb 区间发生同步清除时, 才能通过此功能抑制初始输出。如果在其他区间发生同步清除, 就输出 TOCR1A 和 TOCR2A (TOCR1B 和 TOCR2B) 寄存器设定的初始值, 与 WRE 位的设定无关。如果在 MTU3.TCNT 和 MTU4.TCNT (MTU6.TCNT 和 MTU7.TCNT) 开始计数后的波谷 Tb 区间发生同步清除, 也输出 TOCR1A 和 TOCR2A (TOCR1B 和 TOCR2B) 寄存器设定的初始值。

有关互补 PWM 模式的波谷 Tb 区间, 请参照图 21.41。

[为“1”的条件]

- 在 TWCR.WRE 位为“0”的状态下读 TWCR.WRE 位后, 给 TWCR.WRE 位写“1”时

SCC 位 (同步清除控制位) (只限 TWCRB 寄存器)

此位设定在互补 PWM 模式中发生 MTU0、1、2—MTU6、7 计数器的同步清除时，是否清除 MTU6.TCNT 和 MTU7.TCNT。

使用此功能时，必须将 MTU6 和 MTU7 设定为互补 PWM 模式。另外，如果在计数器的运行过程中改写 SCC 位，必须使 CCE 位和 WRE 位的值不发生变化。

只有在波谷 Tb 区间以外的区间发生同步清除时，才能通过设定 SCC 位使 MTU 的同步清除无效。如果在 MTU6.TCNT 和 MTU7.TCNT 开始计数后的波谷 Tb 区间发生同步清除，就清除 MTU6.TCNT 和 MTU7.TCNT。

有关互补 PWM 模式的波谷 Tb 区间，请参照图 21.41。

[为“1”的条件]

- 在 SCC 位为“0”的状态下读 SCC 位后，给 SCC 位写“1”时

在 TWCRA 寄存器中为保留位。读写值都为“0”。

CCE 位 (比较匹配清除允许位)

此位设定在互补 PWM 模式中是否通过 MTU3.TGRA (MTU6.TGRA) 的比较匹配清除计数器。

[为“1”的条件]

- 在 CCE 位为“0”的状态下读 CCE 位后，给 CCE 位写“1”时

21.2.29 定时器的 A/D 转换开始请求控制寄存器 (TADCR)

• TADCR (MTU4)

地址 MTU4.TADCR 000C 1240h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BF[1:0]	—	—	—	—	—	—	—	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
复位后的值	0	0	0	0	0	0	0	0	0 (注4)	0	0 (注4)	0 (注4)	0 (注4)	0 (注4)	0 (注4)

位	符号	位名	功能	R/W
b0	ITB4VE (注4)	TCIV4 中断减少联动允许位	0: A/D 转换开始请求 (TRG4BN) 不联动 TCIV4 中断减少功能 1 1: A/D 转换开始请求 (TRG4BN) 联动 TCIV4 中断减少功能 1	R/W
b1	ITB3AE (注4)	TGIA3 中断减少联动允许位	0: A/D 转换开始请求 (TRG4BN) 不联动 TGI3A 中断减少功能 1 1: A/D 转换开始请求 (TRG4BN) 联动 TGI3A 中断减少功能 1	R/W
b2	ITA4VE (注4)	TCIV4 中断减少联动允许位	0: A/D 转换开始请求 (TRG4AN) 不联动 TCIV4 中断减少功能 1 1: A/D 转换开始请求 (TRG4AN) 联动 TCIV4 中断减少功能 1	R/W
b3	ITA3AE (注4)	TGIA3 中断减少联动允许位	0: A/D 转换开始请求 (TRG4AN) 不联动 TGI3A 中断减少功能 1 1: A/D 转换开始请求 (TRG4AN) 联动 TGI3A 中断减少功能 1	R/W
b4	DT4BE (注4)	递减计数 TRG4BN 允许位	0: 禁止在 MTU4.TCNT 进行递减计数时产生 A/D 转换开始请求 (TRG4BN) 1: 允许在 MTU4.TCNT 进行递减计数时产生 A/D 转换开始请求 (TRG4BN)	R/W
b5	UT4BE	递增计数 TRG4BN 允许位	0: 禁止在 MTU4.TCNT 进行递增计数时产生 A/D 转换开始请求 (TRG4BN) 1: 允许在 MTU4.TCNT 进行递增计数时产生 A/D 转换开始请求 (TRG4BN)	R/W
b6	DT4AE (注4)	递减计数 TRG4AN 允许位	0: 禁止在 MTU4.TCNT 进行递减计数时产生 A/D 转换开始请求 (TRG4AN) 1: 允许在 MTU4.TCNT 进行递减计数时产生 A/D 转换开始请求 (TRG4AN)	R/W
b7	UT4AE	递增计数 TRG4AN 允许位	0: 禁止在 MTU4.TCNT 进行递增计数时产生 A/D 转换开始请求 (TRG4AN) 1: 允许在 MTU4.TCNT 进行递增计数时产生 A/D 转换开始请求 (TRG4AN)	R/W
b13-b8	—	保留位	读写值都为“0”。	R/W
b15-b14	BF[1:0]	MTU4.TADCOBRA/B 传送时序选择位	选择从 MTU4.TADCOBRA 和 MTU4.TADCOBRB 到 MTU4.TADCORA 和 MTU4.TADCORB 的传送时序。详细内容请参照表 21.48。	R/W

- 注 1. 禁止以 8 位为单位存取 MTU4.TADCR 寄存器，必须以 16 位为单位进行存取。
- 注 2. 在禁止减少中断的情况下，将 TITCR1A.T3AEN 位和 TITCR1A.T4VEN 位置“0”，或者将 TITCR1A.T3ACOR 和 TITCR1A.T4VCOR 置“0”时，必须设定为不联动中断减少功能 1（MTU4.TADCR.ITA3AE、MTU4.TADCR.ITA4VE、MTU4.TADCR.ITB3AE 和 MTU4.TADCR.ITB4VE 位置“0”）。
- 注 3. 在禁止减少中断的情况下，如果设定为联动中断减少功能 1，就不进行 A/D 转换的开始请求。
- 注 4. 除了互补 PWM 模式以外，不能写“1”。

TADCR 寄存器设定允许或者禁止 A/D 转换开始请求，以及设定 A/D 转换开始请求是否联动中断减少功能 1。在 MTU 中，MTU4 和 MTU7 各有 1 个 TADCR。

表 21.48 通过 TADCR.BF[1:0] 位进行传送时序的设定 (MTU4)

bit15	bit14	说明
BF1	BF0	
0	0	不从周期设定缓冲寄存器 (MTU4.TADCOBRA、MTU4.TADCOBRB) 传送到周期设定寄存器 (MTU4.TADCORA、MTU4.TADCORB)。
0	1	在 MTU4.TCNT 的波峰从周期设定缓冲寄存器传送到周期设定寄存器 (注 1)。
1	0	在 MTU4.TCNT 的波谷从周期设定缓冲寄存器传送到周期设定寄存器 (注 2)。
1	1	在 MTU4.TCNT 的波峰和波谷从周期设定缓冲寄存器传送到周期设定寄存器 (注 2)。

- 注 1. 在互补 PWM 模式中，在 MTU4.TCNT 的波峰和写 MTU4.TGRD 寄存器时；在复位同步 PWM 模式中，在发生 MTU3.TCNT 和 MTU3.TGRA 寄存器的比较匹配时；在 PWM 模式 1 和正常运行模式中，在发生 MTU4.TCNT 和 MTU4.TGRA 寄存器的比较匹配时，从周期设定缓冲寄存器 (MTU4.TADCOBRA、MTU4.TADCOBRB) 传送到周期设定寄存器 (MTU4.TADCORA、MTU4.TADCORB)。
- 注 2. 除了互补 PWM 模式以外，禁止此设定。

- TADCR (MTU7)

地址 000C 1A40h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BF[1:0]	—	—	—	—	—	—	—	UT7AE	DT7AE	UT7BE	DT7BE	ITA6AE	TA7VE	TB6AE	TB7VE
复位后的值	0	0	0	0	0	0	0	0	0 (注4)	0	0 (注4)	0 (注4)	0 (注4)	0 (注4)	0 (注4)

位	符号	位名	功能	R/W
b0	ITB7VE (注4)	TCIV7 中断减少联动允许位	0: A/D 转换开始请求 (TRG7BN) 不联动 TCI7V 中断减少功能 1 1: A/D 转换开始请求 (TRG7BN) 联动 TCI7V 中断减少功能 1	R/W
b1	ITB6AE (注4)	TGIA6 中断减少联动允许位	0: A/D 转换开始请求 (TRG7BN) 不联动 TGI6A 中断减少功能 1 1: A/D 转换开始请求 (TRG7BN) 联动 TGI6A 中断减少功能 1	R/W
b2	ITA7VE (注4)	TCIV7 中断减少联动允许位	0: A/D 转换开始请求 (TRG7AN) 不联动 TCI7V 中断减少功能 1 1: A/D 转换开始请求 (TRG7AN) 联动 TCI7V 中断减少功能 1	R/W
b3	ITA6AE (注4)	TGIA6 中断减少联动允许位	0: A/D 转换开始请求 (TRG7AN) 不联动 TGI6A 中断减少功能 1 1: A/D 转换开始请求 (TRG7AN) 联动 TGI6A 中断减少功能 1	R/W
b4	DT7BE (注4)	递减计数 TRG7BN 允许位	0: 禁止在 MTU7.TCNT 进行递减计数时产生 A/D 转换开始请求 (TRG7BN) 1: 允许在 MTU7.TCNT 进行递减计数时产生 A/D 转换开始请求 (TRG7BN)	R/W
b5	UT7BE	递增计数 TRG7BN 允许位	0: 禁止在 MTU7.TCNT 进行递增计数时产生 A/D 转换开始请求 (TRG7BN) 1: 允许在 MTU7.TCNT 进行递增计数时产生 A/D 转换开始请求 (TRG7BN)	R/W
b6	DT7AE (注4)	递减计数 TRG7AN 允许位	0: 禁止在 MTU7.TCNT 进行递减计数时产生 A/D 转换开始请求 (TRG7AN) 1: 允许在 MTU7.TCNT 进行递减计数时产生 A/D 转换开始请求 (TRG7AN)	R/W
b7	UT7AE	递增计数 TRG7AN 允许位	0: 禁止在 MTU7.TCNT 进行递增计数时产生 A/D 转换开始请求 (TRG7AN) 1: 允许在 MTU7.TCNT 进行递增计数时产生 A/D 转换开始请求 (TRG7AN)	R/W
b13-b8	—	保留位	读写值都为“0”。	R/W
b15-b14	BF[1:0]	MTU7.TADCOBRA/B 传送时序选择位	选择从 MTU7.TADCOBRA 和 MTU7.TADCOBRB 到 MTU7.TADCORA 和 MTU7.TADCORB 的传送时序。详细内容请参照表 21.49。	R/W

- 注 1. 禁止以 8 位为单位存取 MTU7.TADCR 寄存器，必须以 16 位为单位进行存取。
- 注 2. 在禁止减少中断的情况下，将 TITCR1B.T6AEN 位和 TITCR1B.T7VEN 位置“0”，或者将 TITCR1B.T6ACOR 位和 TITCR1B.T7VCOR 位置“0”时，必须设定为不联动中断减少功能 1 (MTU7.TADCR.ITA6AE、MTU7.TADCR.ITA7VE、MTU7.TADCR.ITB6AE 和 MTU7.TADCR.ITB7VE 位置“0”)。
- 注 3. 在禁止减少中断的情况下，如果设定为联动中断减少功能 1，就不进行 A/D 转换的开始请求。
- 注 4. 除了互补 PWM 模式以外，不能设定为“1”。

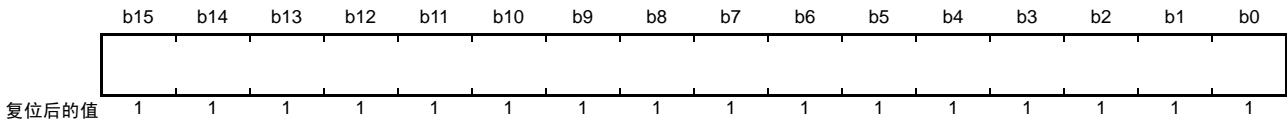
表 21.49 通过 TADCR.BF[1:0] 位进行传送时序的设定 (MTU7)

bit15	bit14	说明
BF1	BF0	
0	0	不从周期设定缓冲寄存器 (MTU7.TADCOBRA、MTU7.TADCOBRB) 传送到周期设定寄存器 (MTU7.TADCORA、MTU7.TADCORB)。
0	1	在 MTU7.TCNT 的波峰从周期设定缓冲寄存器传送到周期设定寄存器 (注 1)。
1	0	在 MTU7.TCNT 的波谷从周期设定缓冲寄存器传送到周期设定寄存器 (注 2)。
1	1	在 MTU7.TCNT 的波峰和波谷从周期设定缓冲寄存器传送到周期设定寄存器 (注 2)。

- 注 1. 在互补 PWM 模式中，在 MTU7.TCNT 的波峰和写 MTU7.TGRD 寄存器时；在复位同步 PWM 模式中，在发生 MTU6.TCNT 和 MTU6.TGRA 寄存器的比较匹配时；在 PWM 模式 1 和正常运行模式中，在发生 MTU7.TCNT 和 MTU7.TGRA 寄存器的比较匹配时，从周期设定缓冲寄存器 (MTU7.TADCOBRA、MTU7.TADCOBRB) 传送到周期设定寄存器 (MTU7.TADCORA、MTU7.TADCORB)。
- 注 2. 除了互补 PWM 模式以外，禁止此设定。

21.2.30 定时器的 A/D 转换开始请求周期设定寄存器 (TADCORA、TADCORB)

地址 MTU4.TADCORA 000C 1244h、MTU4.TADCORB 000C 1246h、MTU7.TADCORA 000C 1A44h、MTU7.TADCORB 000C 1A46h



- 注 1. 禁止以 8 位为单位存取 MTUn.TADCORA 和 MTUn.TADCORB (n=4、7)。必须以 16 位为单位进行存取。
- 注 2. 使用中断减少功能 1 联动 A/D 转换开始请求延迟功能 (详细内容请参照 "21.3.9(4) 联动中断减少功能 1 的 A/D 转换开始请求延迟功能") 时, 必须将此寄存器的值设定为: 0002h ~ TCDRA 的设定值 -2 (MTU4) 或者 0002h ~ TCDRB 的设定值 -2 (MTU7)
- 注 3. 在使用中断减少功能 2, 并且 MTUn.TADCORA 寄存器值与 MTUn.TADCORB 寄存器值之间的间隔小时, 就无法正确的对减少次数进行计数, 因此就有可能无法在期待的时序时产生 A/D 转换开始请求。
必须在以下的设定条件下使用。

(1) 当使用中断减少功能 2 并且中断减少次数=0 时

- MTUn.TADCORA 寄存器值与 MTUn.TADCORB 寄存器值之间的间隔 ≥ 4
- MTUn.TADCORA 的比较间隔 ≥ 4 个 PCLKA (将 MTUn.TADCORA 寄存器的更新值设定为: 更新前的值 + 大于等于 4 的值或者更新前的值 - 小于等于 4 的值)
- MTUn.TADCORB 的比较间隔 ≥ 4 个 PCLKA (将 MTUn.TADCORB 寄存器的更新值设定为: 更新前的值 + 大于等于 4 的值或者更新前的值 - 小于等于 4 的值)

(2) 当使用中断减少功能 2 并且中断减少次数 ≥ 1 时

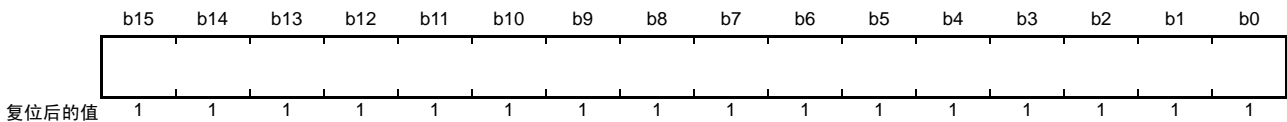
- MTUn.TADCORA 寄存器值与 MTUn.TADCORB 寄存器值之间的间隔 ≥ 2
- MTUn.TADCORB 的比较间隔 ≥ 2 个 PCLKA (将 MTUn.TADCORB 寄存器的更新值设定为: 更新前的值 + 大于等于 2 的值或者更新前的值 - 小于等于 2 的值)

TADCORA 寄存器和 TADCORB 寄存器是 16 位可读写寄存器。当与 MTUn.TCNT (n=4、7) 匹配时, 产生对应的 A/D 转换开始请求。

复位后, TADCORA 寄存器和 TADCORB 寄存器的值为 "FFFFh"。

21.2.31 定时器的 A/D 转换开始请求周期设定缓冲寄存器 (TADCOBRA、TADCOBRB)

地址 MTU4.TADCORA 000C 1248h、MTU4.TADCORB 000C 124Ah、MTU7.TADCORA 000C 1A48h、MTU7.TADCORB 000C 1A4Ah



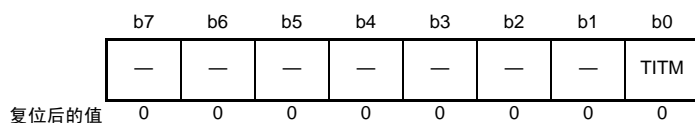
- 注. 禁止以 8 位为单位存取 TADCOBRA 寄存器和 TADCOBRB 寄存器。必须以 16 位为单位进行存取。

TADCOBRA 寄存器和 TADCOBRB 寄存器是 16 位可读写寄存器, 用作 TADCORA 寄存器和 TADCORB 寄存器的缓冲寄存器。在波峰或者波谷从 TADCORA 寄存器、TADCOBRB 寄存器传送到 TADCORA 寄存器、TADCORB 寄存器。

复位后, TADCOBRA 寄存器和 TADCOBRB 寄存器的值为 "FFFFh"。

21.2.32 定时器的中断减少模式寄存器 (TITMRA、TITMRB)

地址 MTU.TITMRA 000C 123Ah、MTU.TITMRB 000C 1A3Ah



位	符号	位名	功能	R/W
b0	TITM	中断减少功能选择位	能选择 2 种中断减少功能。 详细内容请参照表 21.50.	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

TITMRA 寄存器和 TITMRB 寄存器能选择 2 种中断减少功能。

表 21.50 通过 TITM 位设定的中断减少功能

bit0	说明
TITM	
0	中断减少功能 1 (注1)
1	中断减少功能 2 (注2)

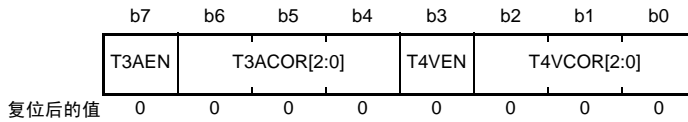
注 1. 通过设定 TITCR1A 寄存器和 TITCR1B 寄存器使中断减少功能 1 有效。

注 2. 通过设定 TITCR2A 寄存器和 TITCR2B 寄存器使中断减少功能 2 有效。

21.2.33 定时器的中断减少设定寄存器 1 (TITCR1A、TITCR1B)

• TITCR1A

地址 MTU.TITCR1A 000C 1230h



位	符号	位名	功能	R/W
b2-b0	T4VCOR[2:0]	TCIV4 中断减少次数设定位	将 TCIV4 的中断减少次数设定为 0 ~ 7 次, 详细内容请参照表 21.51。	R/W
b3	T4VEN	T4VEN 位	0: 禁止 TCIV4 中断的减少 1: 允许 TCIV4 中断的减少	R/W
b6-b4	T3ACOR[2:0]	TGIA3 中断减少次数设定位	将 TGIA3 的中断减少次数设定为 0 ~ 7 次 (注 1), 详细内容请参照表 21.52。	R/W
b7	T3AEN	T3AEN 位	0: 禁止 TGIA3 中断的减少 1: 允许 TGIA3 中断的减少	R/W

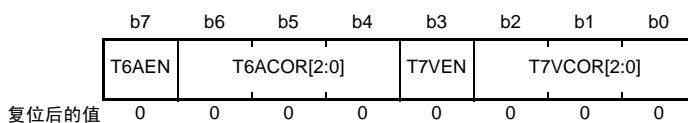
注 1. 如果将中断减少次数设定为“0”, 就不减少中断。

另外, 必须在更改中断减少次数前, 通过将 TITCR1A.T3AEN 位和 TITCR1A.T4VEN 位置“0”, 清除减少次数计数器 (TITCNT1A)。

TITCR1A 寄存器和 TITCR1B 寄存器禁止或者允许中断减少, 以及控制中断减少次数的设定。此设定只在将 TITMRA 寄存器和 TITMRB 寄存器置“0”时有效。如果将 TITMRA 寄存器和 TITMRB 寄存器置“1”, 此寄存器的值就被清除。

• TITCR1B

地址 MTU.TITCR1B 000C 1A30h



位	符号	位名	功能	R/W
b2-b0	T7VCOR[2:0]	TCIV7 中断减少次数设定位	将 TCIV7 的中断减少次数设定为 0 ~ 7 次, 详细内容请参照表 21.53。	R/W
b3	T7VEN	T7VEN 位	0: 禁止 TCIV7 中断的减少 1: 允许 TCIV7 中断的减少	R/W
b6-b4	T6ACOR[2:0]	TGIA6 中断减少次数设定位	将 TGIA6 的中断减少次数设定为 0 ~ 7 次 (注 1), 详细内容请参照表 21.54。	R/W
b7	T6AEN	T6AEN 位	0: 禁止 TGIA6 中断的减少 1: 允许 TGIA6 中断的减少	R/W

注 1. 如果将中断减少次数设定为“0”, 就不减少中断。

另外, 必须在更改中断减少次数前, 通过将 TITCR1B.T6AEN 位和 TITCR1B.T7VEN 位置“0”, 清除减少次数计数器 (TITCNT1B)。

表 21.51 通过 T4VCOR[2:0] 位设定的中断减少次数

bit2	bit1	bit0	说明
T4VCOR2	T4VCOR1	T4VCOR0	
0	0	0	不减少 TCIV4 的中断。
0	0	1	将 TCIV4 的中断减少次数设定为 1 次。
0	1	0	将 TCIV4 的中断减少次数设定为 2 次。
0	1	1	将 TCIV4 的中断减少次数设定为 3 次。
1	0	0	将 TCIV4 的中断减少次数设定为 4 次。
1	0	1	将 TCIV4 的中断减少次数设定为 5 次。
1	1	0	将 TCIV4 的中断减少次数设定为 6 次。
1	1	1	将 TCIV4 的中断减少次数设定为 7 次。

表 21.52 通过 T3ACOR[2:0] 位设定的中断减少次数

bit6	bit5	bit4	说明
T3ACOR2	T3ACOR1	T3ACOR0	
0	0	0	不减少 TGIA3 的中断。
0	0	1	将 TGIA3 的中断减少次数设定为 1 次。
0	1	0	将 TGIA3 的中断减少次数设定为 2 次。
0	1	1	将 TGIA3 的中断减少次数设定为 3 次。
1	0	0	将 TGIA3 的中断减少次数设定为 4 次。
1	0	1	将 TGIA3 的中断减少次数设定为 5 次。
1	1	0	将 TGIA3 的中断减少次数设定为 6 次。
1	1	1	将 TGIA3 的中断减少次数设定为 7 次。

表 21.53 通过 T7VCOR[2:0] 位设定的中断减少次数

bit2	bit1	bit0	说明
T7VCOR2	T7VCOR1	T7VCOR0	
0	0	0	不减少 TCIV7 的中断。
0	0	1	将 TCIV7 的中断减少次数设定为 1 次。
0	1	0	将 TCIV7 的中断减少次数设定为 2 次。
0	1	1	将 TCIV7 的中断减少次数设定为 3 次。
1	0	0	将 TCIV7 的中断减少次数设定为 4 次。
1	0	1	将 TCIV7 的中断减少次数设定为 5 次。
1	1	0	将 TCIV7 的中断减少次数设定为 6 次。
1	1	1	将 TCIV7 的中断减少次数设定为 7 次。

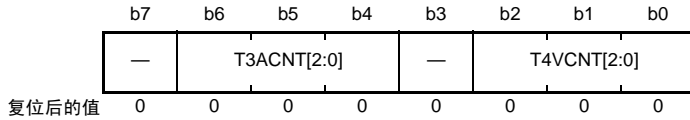
表 21.54 通过 T6ACOR[2:0] 位设定的中断减少次数

bit6	bit15	bit4	说明
T6ACOR2	T6ACOR1	T6ACOR0	
0	0	0	不减少 TGIA6 的中断。
0	0	1	将 TGIA6 的中断减少次数设定为 1 次。
0	1	0	将 TGIA6 的中断减少次数设定为 2 次。
0	1	1	将 TGIA6 的中断减少次数设定为 3 次。
1	0	0	将 TGIA6 的中断减少次数设定为 4 次。
1	0	1	将 TGIA6 的中断减少次数设定为 5 次。
1	1	0	将 TGIA6 的中断减少次数设定为 6 次。
1	1	1	将 TGIA6 的中断减少次数设定为 7 次。

21.2.34 定时器的中断减少次数计数器 1 (TITCNT1A、TITCNT1B)

• TITCNT1A

地址 MTU.TITCNT1A 000C 1231h



位	符号	位名	功能	R/W
b2-b0	T4VCNT[2:0]	TCIV4 中断计数器位	如果 TITCR1A 的 T4VEN 位为“1”，就在产生 TCIV4 中断源时递增 1。	R
b3	—	保留位	读取值为“0”，写操作无效。	R
b6-b4	T3ACNT[2:0]	TGIA3 中断计数器位	如果 TITCR1A 的 T3AEN 位为“1”，就在产生 TGIA3 中断源时递增 1。	R
b7	—	保留位	读取值为“0”，写操作无效。	R

注. 要清除 TITCNT1A 寄存器的值时，必须将 TITCR1A.T3AEN 位和 TITCR1A.T4VEN 位置“0”。

TITCNT1A 寄存器和 TITCNT1B 寄存器是 8 位可读计数器。在 MTU3.TCNT 和 MTU4.TCNT (MTU6.TCNT 和 MTU7.TCNT) 停止计数后，TITCNT1A 寄存器和 TITCNT1B 寄存器仍保持原来的值。

T4VCNT[2:0] 位 (TCIV4 中断计数器位)

如果 TITCR1A.T4VEN 位为“1”，就在产生 TCIV4 中断源时递增 1。

[为“0”的条件]

- 在 TITMRA.TITM 位为“1”时
- 在 TITCR1A.T4VEN 位为“0”时
- 在 TITCR1A.T4VCOR[2:0] 位为“000b”时
- 在 TITCR1A.T4VCOR[2:0] 位和 TITCNT1A.T4VCNT[2:0] 位匹配时

T3ACNT[2:0] 位 (TGIA3 中断计数器位)

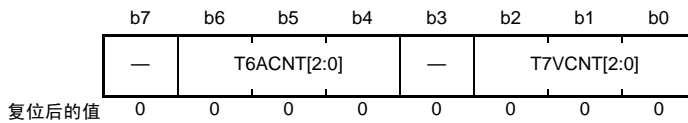
如果 TITCR1A.T3AEN 位为“1”，就在产生 TGIA3 中断源时递增 1。

[为“0”的条件]

- 在 TITMRA.TITM 位为“1”时
- 在 TITCR1A.T3AEN 位为“0”时
- 在 TITCR1A.T3ACOR[2:0] 位为“000b”时
- 在 TITCR1A.T3ACOR[2:0] 位和 TITCNT1A.T3ACNT[2:0] 位匹配时

- TITCNT1B

地址 MTU.TITCNT1B 000C 1A31h



位	符号	位名	功能	R/W
b2-b0	T7VCNT[2:0]	TCIV7 中断计数器位	如果 TITCR1B 的 T7VEN 位为“1”，就在产生 TCIV7 中断源时递增 1。	R
b3	—	保留位	读取值为“0”，写操作无效。	R
b6-b4	T6ACNT[2:0]	TGIA6 中断计数器位	如果 TITCR1B 的 T6AEN 位为“1”，就在产生 TGIA6 中断源时递增 1。	R
b7	—	保留位	读取值为“0”，写操作无效。	R

注. 要清除 TITCNT1B 寄存器的值时，必须将 TITCR1B.T6AEN 位和 TITCR1B.T7VEN 位置“0”。

T7VCNT[2:0] 位 (TCIV7 中断计数器位)

如果 TITCR1B.T7VEN 位为“1”，就在产生 TCIV7 中断源时递增 1。

[为“0”的条件]

- 在 TITMRB.TITM 位为“1”时
- 在 TITCR1B.T7VEN 位为“0”时
- 在 TITCR1B.T7VCOR[2:0] 位为“000b”时
- 在 TITCR1B.T7VCOR[2:0] 位和 TITCNT1B.T7VCNT[2:0] 位匹配时

T6ACNT[2:0] 位 (TGIA6 中断计数器位)

如果 TITCR1B.T6AEN 位为“1”，就在产生 TGIA6 中断源时递增 1。

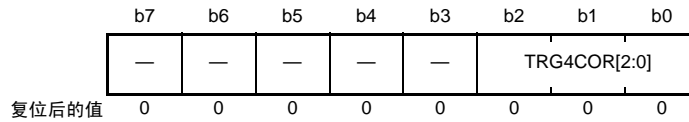
[为“0”的条件]

- 在 TITMRB.TITM 位为“1”时
- 在 TITCR1B.T6AEN 位为“0”时
- 在 TITCR1B.T6ACOR[2:0] 位为“000b”时
- 在 TITCR1B.T6ACOR[2:0] 位和 TITCNT1B.T6ACNT[2:0] 位匹配时

21.2.35 定时器的中断减少设定寄存器 2 (TITCR2A、TITCR2B)

• TITCR2A

地址 MTU.TITCR2A 000C 123Bh



位	符号	位名	功能	R/W
b2-b0	TRG4COR[2:0]	TRG4AN/TRG4BN 中断减少次数设定位	将 TRG4AN/TRG4BN 的中断减少次数设定为 0 ~ 7 次。详细内容请参照表 21.55。	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

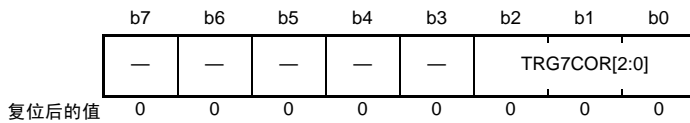
TITCR2A 寄存器和 TITCR2B 寄存器设定 TRG4AN 和 TRG4BN (TRG7AN 和 TRG7BN) 的中断减少次数。此设定只在将 TITMRA 寄存器和 TITMRB 寄存器置“1”时有效。

表 21.55 通过 TRG4COR[2:0] 位设定的中断减少次数

bit2	bit1	bit0	说明
TRG4COR2	TRG4COR1	TRG4COR0	
0	0	0	不减少 TRG4AN 和 TRG4BN 的中断。
0	0	1	将 TRG4AN 和 TRG4BN 的中断减少次数设定为 1 次。
0	1	0	将 TRG4AN 和 TRG4BN 的中断减少次数设定为 2 次。
0	1	1	将 TRG4AN 和 TRG4BN 的中断减少次数设定为 3 次。
1	0	0	将 TRG4AN 和 TRG4BN 的中断减少次数设定为 4 次。
1	0	1	将 TRG4AN 和 TRG4BN 的中断减少次数设定为 5 次。
1	1	0	将 TRG4AN 和 TRG4BN 的中断减少次数设定为 6 次。
1	1	1	将 TRG4AN 和 TRG4BN 的中断减少次数设定为 7 次。

- TITCR2B

地址 MTU.TITCR2B 000C 1A3Bh



位	符号	位名	功能	R/W
b2-b0	TRG7COR[2:0]	TRG7AN/TRG7BN 中断减少次数 设定位	将 TRG7AN/TRG7BN 的中断减少次数设定为 0 ~ 7 次。详细内容请参照表 21.56。	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

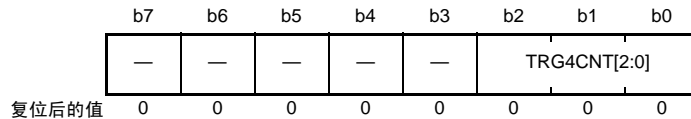
表 21.56 通过 TRG7COR[2:0] 位设定的中断减少次数

bit2	bit1	bit0	说明
TRG7COR2	TRG7COR1	TRG7COR0	
0	0	0	不减少 TRG7AN 和 TRG7BN 的中断。
0	0	1	将 TRG7AN 和 TRG7BN 的中断减少次数设定为 1 次。
0	1	0	将 TRG7AN 和 TRG7BN 的中断减少次数设定为 2 次。
0	1	1	将 TRG7AN 和 TRG7BN 的中断减少次数设定为 3 次。
1	0	0	将 TRG7AN 和 TRG7BN 的中断减少次数设定为 4 次。
1	0	1	将 TRG7AN 和 TRG7BN 的中断减少次数设定为 5 次。
1	1	0	将 TRG7AN 和 TRG7BN 的中断减少次数设定为 6 次。
1	1	1	将 TRG7AN 和 TRG7BN 的中断减少次数设定为 7 次。

21.2.36 定时器的中断减少次数计数器 2 (TITCNT2A、TITCNT2B)

• TITCNT2A

地址 MTU.TITCNT2A 000C 123Ch



位	符号	位名	功能	R/W
b2-b0	TRG4CNT[2:0]	TRG4AN/TRG4BN 中断计数器位	每当发生 TRG4AN、TRG4BN 的中断时，从 TRG4COR[2:0] 位设定的值开始递减，直到计数器的值变为“0”。在发生重加载时，TRG4AN 和 TRG4BN 的中断有效。	R
b7-b3	—	保留位	读取值为“0”，写操作无效。	R

每当发生 TRG4AN、TRG4BN (TITCNT2A) 的中断和 TRG7AN、TRG7BN (TITCNT2B) 的中断时，TITCNT2A 寄存器和 TITCNT2B 寄存器就从 TRG4COR[2:0] 位、TRG7COR[2:0] 位设定的值开始递减，直到计数器的值变为“0”。在发生重加载时，TRG4AN、TRG4BN 的中断和 TRG7AN、TRG7BN 的中断有效。

TRG4CNT[2:0] 位 (TRG4AN/TRG4BN 中断计数器位)

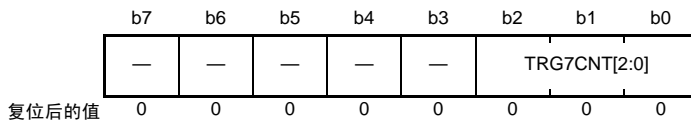
每当发生 TRG4AN 中断和 TRG4BN 中断时，从 TRG4COR[2:0] 位设定的值开始递减，直到计数器的值为“0”。在发生重加载时，TRG4AN 的中断和 TRG4BN 的中断有效。

[为“0”的条件]

- 在 TITMRA.TITM 位为“1”时
- 在 TITCR2A.TRG4COR[2:0] 位为“000b”时
- 在 TITCR2A.TRG4COR[2:0] 位与 TRG4AN 和 TRG4BN 的发生次数相同时

- TITCNT2B

地址 MTU.TITCNT2B 000C 1A3Ch



位	符号	位名	功能	R/W
b2-b0	TRG7CNT[2:0]	TRG7AN/TRG7BN 中断计数器位	每当发生 TRG7AN、TRG7BN 的中断时，从 TRG7COR[2:0] 位设定的值开始递减，直到计数器的值变为“0”。在发生重加载时，TRG7AN 和 TRG7BN 的中断有效。	R
b7-b3	—	保留位	读取值为“0”，写操作无效。	R

TRG7CNT[2:0] 位 (TRG7AN/TRG7BN 中断计数器位)

每当发生 TRG7AN 的中断和 TRG7BN 的中断时，从 TRG7COR[2:0] 位设定的值开始递减，直到计数器的值为“0”。在发生重加载时，TRG7AN 的中断和 TRG7BN 的中断有效。

[为“0”的条件]

- 在 TITMRB.TITM 位为“1”时
- 在 TITCR2B.TRG7COR[2:0] 位为“000b”时
- 在 TITCR2B.TRG7COR[2:0] 位设定的值与 TRG7AN 和 TRG7BN 的中断发生次数相同时

21.2.37 和总线主控的接口

定时器的计数器 (TCNT)、通用寄存器 (TGR)、定时器的副计数器 (TCNTSA、TCNTSB)、定时器的周期缓冲寄存器 (TCBRA、TCBRB)、定时器的死区时间数据寄存器 (TDDRA、TDDRB)、定时器的周期数据寄存器 (TCDRA、TCDRB)、定时器的 A/D 转换开始请求控制寄存器 (MTU4.TADCR、MTU7.TADCR)、定时器的 A/D 转换开始请求周期设定寄存器 (MTU4.TADCORA、MTU4.TADCORB、MTU7.TADCORA、MTU7.TADCORB) 和定时器的 A/D 转换开始请求周期设定缓冲寄存器 (MTU4.TADCOBRA、MTU4.TADCOBRB、MTU7.TADCOBRA、MTU7.TADCOBRB) 都是 16 位寄存器。因为和总线主控之间的数据总线宽度为 16 位，所以能以 16 位为单位进行读写，而不能以 8 位为单位进行读写。必须以 16 位为单位进行存取。

上述以外的寄存器都是 8 位寄存器。因为和 CPU 之间的数据总线宽度为 16 位，所以能以 16 位为单位进行读写。另外，也能以 8 位为单位进行读写。

21.3 运行说明

21.3.1 基本运行

各通道有 TCNT 寄存器和 TGR 寄存器。TCNT 进行递增计数，能进行自由运行、周期计数器运行或者外部事件计数运行。

TGR 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 寄存器的 CST0 ~ CST4 位、TSTRB 寄存器的 CST6 位和 CST7 位、以及 MTU5.TSTR 寄存器的 CSTU5 位、CSTV5 位、CSTW5 位置“1”，对应通道的 TCNT 就开始计数。能进行自由运行计数器和周期计数器的运行等。

(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 21.5 所示。

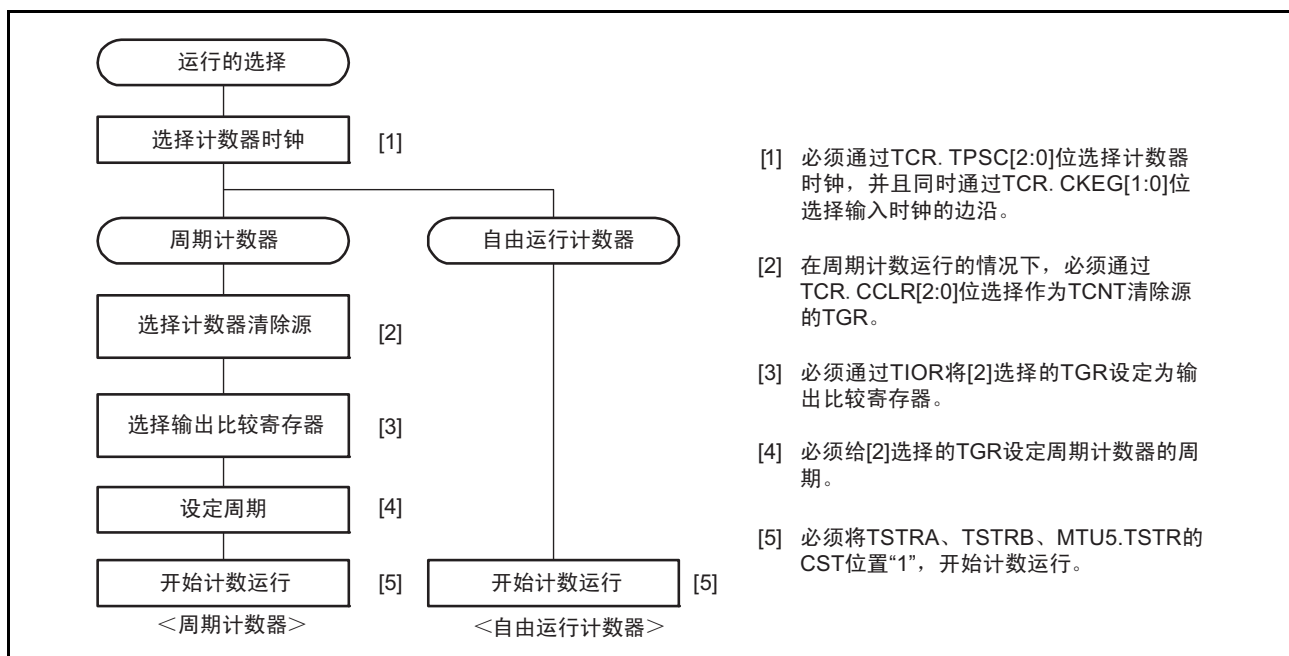


图 21.5 计数器运行的设定步骤例子

(b) 自由运行计数器的运行和周期计数器的运行

MTU 的 TCNT 在复位后立即被全部设定为自由运行计数器。如果将 TSTRA、TSTRB、MTU5.TSTR 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。如果 TCNT 发生上溢 (FFFFh→0000h)，TSR.TCFV 标志就变为“1”。此时，如果对应的 TIER.TCIEV 位为“1”，MTU 就请求中断。TCNT 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 21.6 所示。

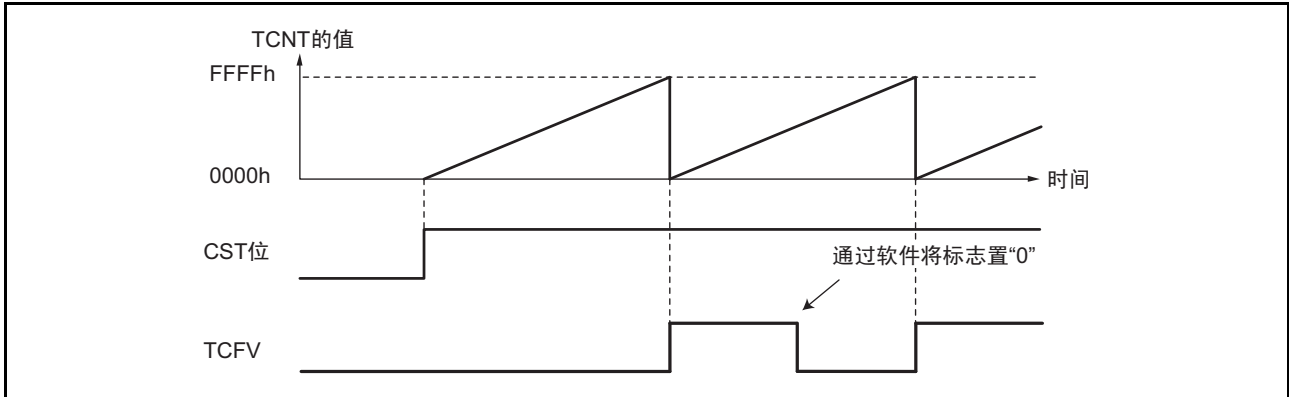


图 21.6 自由运行计数器的运行

当选择比较匹配作为 TCNT 的清除源时，对应通道的 TCNT 进行周期计数。将用于设定周期的 TGR 设定为输出比较寄存器，并且由 TCR.CCLR[2:0] 位选择通过比较匹配进行计数器清除。如果在设定后将 TSTRA、TSTRB 和 MTU5.TSTR 寄存器的对应位置“1”，就作为周期计数器开始递增计数。如果计数值和 TGR 的值相同，TSR.TGF 标志就变为“1”，TCNT 就变为“0000h”。

此时，如果对应的 TIER.TGIE 位为“1”，MTU 就请求中断。TCNT 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 21.7 所示。

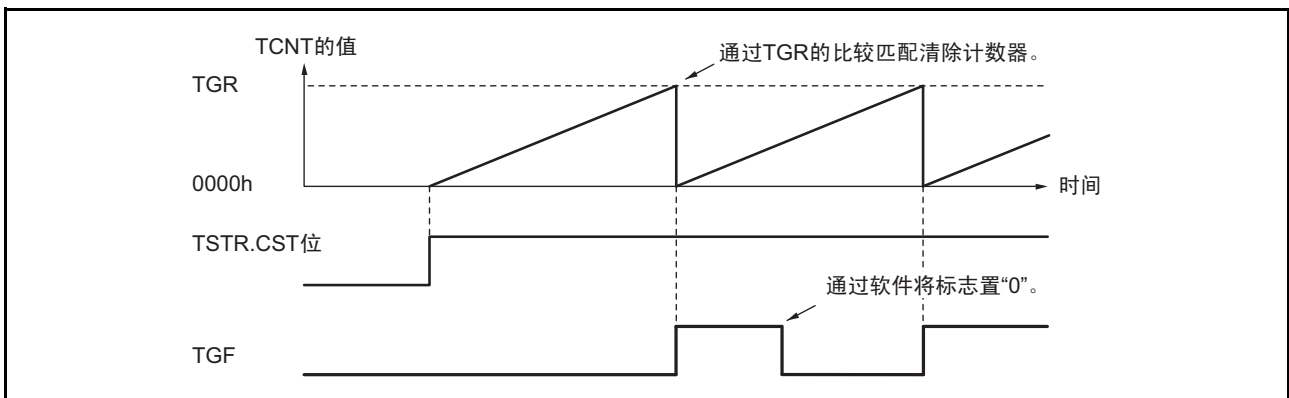


图 21.7 周期计数器的运行

(2) 通过比较匹配进行波形输出的功能

MTU 能通过比较匹配从对应的输出引脚输出 Low 电平、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 21.8 所示。

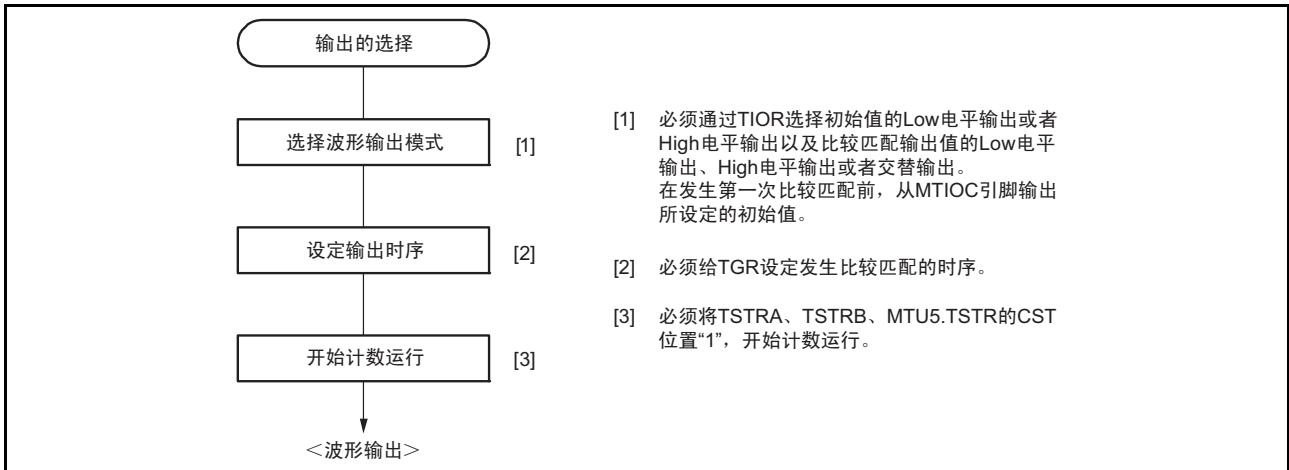


图 21.8 通过比较匹配进行波形输出的运行例子

(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 21.9 所示。

在此例子中，假设 TCNT 进行自由运行计数，并且通过比较匹配 A 输出 High 电平，通过比较匹配 B 输出 Low 电平的例子。如果设定的电平和引脚的电平相同，引脚的电平就不变。

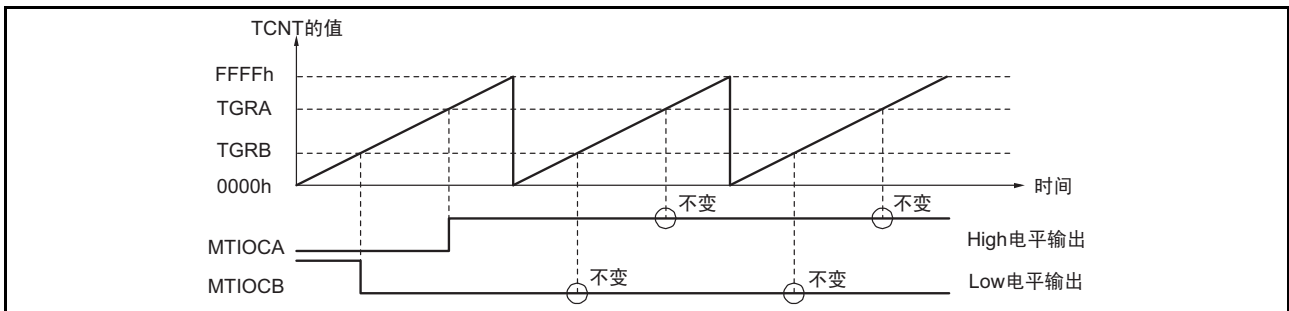


图 21.9 输出 Low 电平 /High 电平的运行例子

交替输出的运行例子如图 21.10 所示。

在此例子中，假设 TCNT 进行周期计数（通过比较匹配 B 清除计数器），并且比较匹配 A 和比较匹配 B 都进行交替输出。

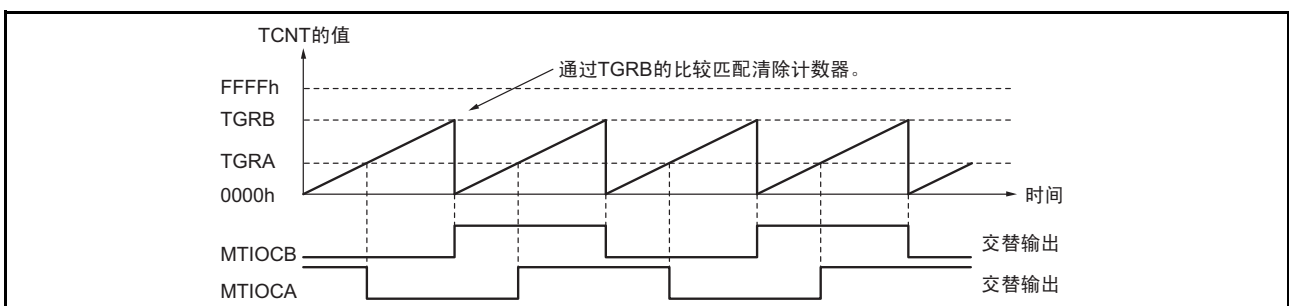


图 21.10 交替输出的运行例子

(3) 输入捕捉功能

能在检测到 MTIOC 引脚的输入边沿后将 TCNT 的值传送到 TGR 寄存器。

检测边沿可选择上升沿、下降沿或者双边沿。另外，MTU0 和 MTU1 也能将其他通道的计数器输入时钟或者比较匹配信号作为输入捕捉源。

注. 如果 MTU0 和 MTU1 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 PCLKA/1 作为输入捕捉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 21.11 所示。

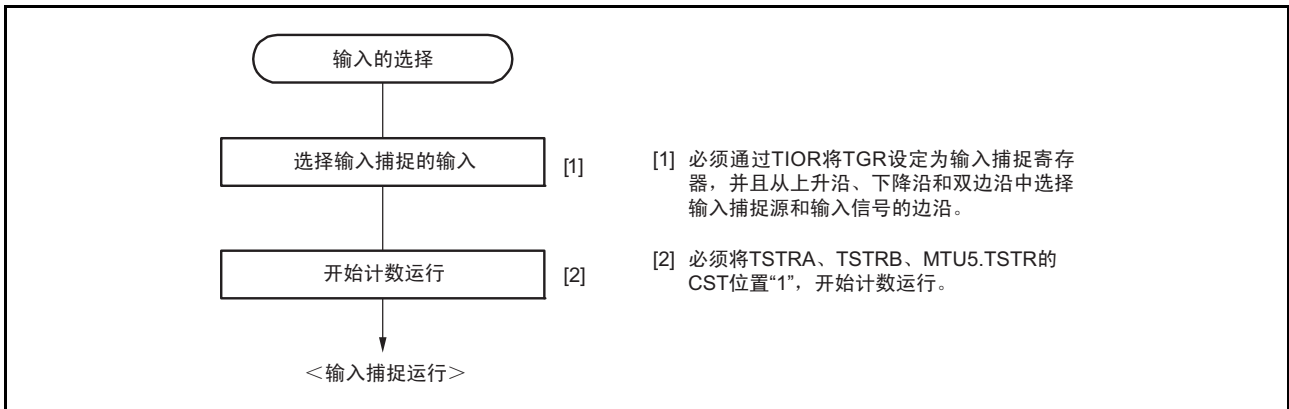


图 21.11 输入捕捉运行的设定例子

(b) 输入捕捉的运行例子

输入捕捉的运行例子如图 21.12 所示。

在此例子中，假设选择上升沿 / 下降沿的双边沿作为 MTIOCA 引脚的输入捕捉的输入边沿，选择下降沿作为 MTIOCB 引脚的输入捕捉的输入边沿，并且在发生 TGRB 的输入捕捉时清除 TCNT 计数器。

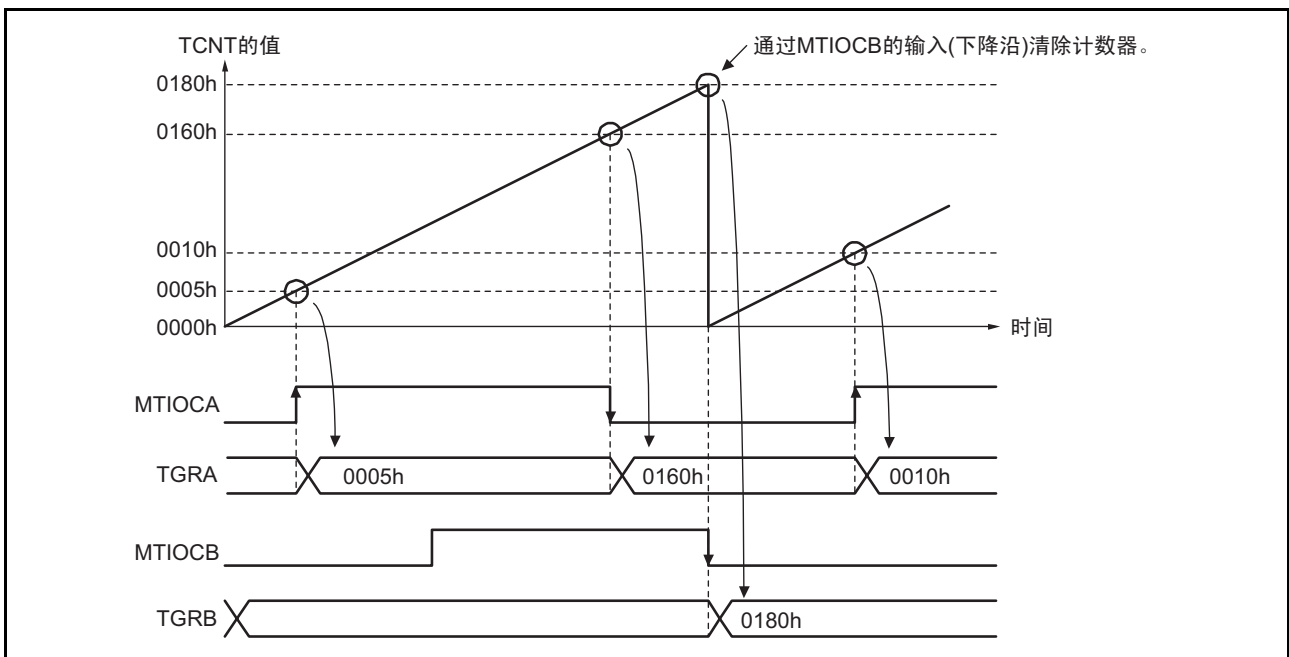


图 21.12 输入捕捉的运行例子

21.3.2 同步运行

能使用同步运行同时改写多个 TCNT 的值（同步预置），还能通过设定 TCR 寄存器同时将多个 TCNT 置“0”（同步清除）。

能通过同步运行对 1 个时基增加要运行的 TGR 寄存器个数。

MTU0 ~ 4、6、7 都能设定为同步运行。

MTU5 不能设定为同步运行。

(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 21.13 所示。

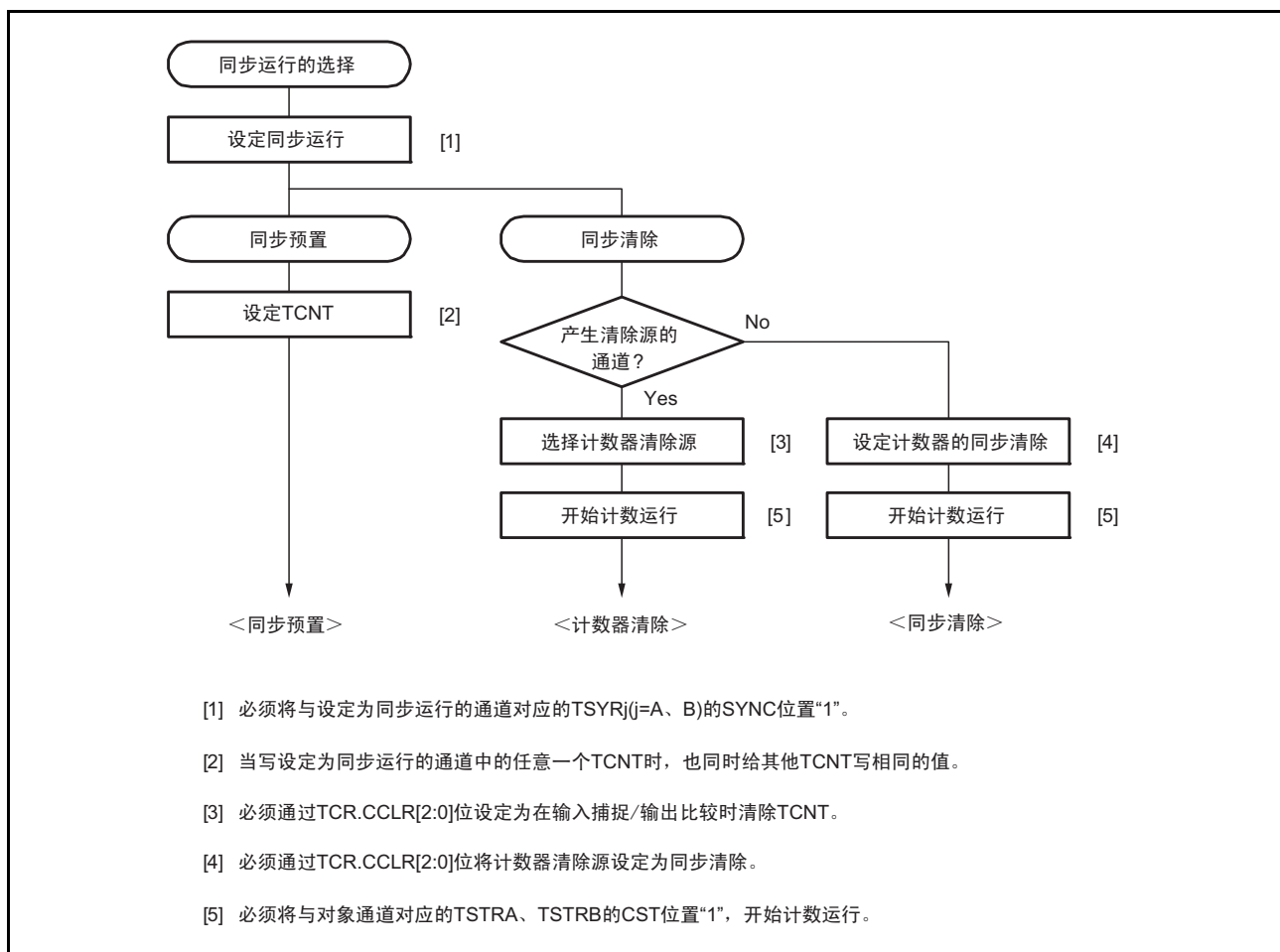


图 21.13 同步运行的设定步骤例子

(2) 同步运行的例子

同步运行的例子如图 21.14 所示。

在此例子中，假设将 MTU0 ~ 2 设定为同步运行和 PWM 模式 1，将 MTU0 的计数器清除源设定为 MTU0.TGRB 的比较匹配，将 MTU1 和 MTU2 的计数器清除源设定为同步清除。

从 MTIOC0A、MTIOC1A、MTIOC2A 引脚输出 3 相 PWM 波形。此时，MTU0 ~ 2 的 TCNT 进行同步预置并且通过 MTU0.TGRB 的比较匹配进行同步清除，MTU0.TGRB 的设定数据为 PWM 周期。

有关 PWM 模式，请参照“21.3.5 PWM 模式”。

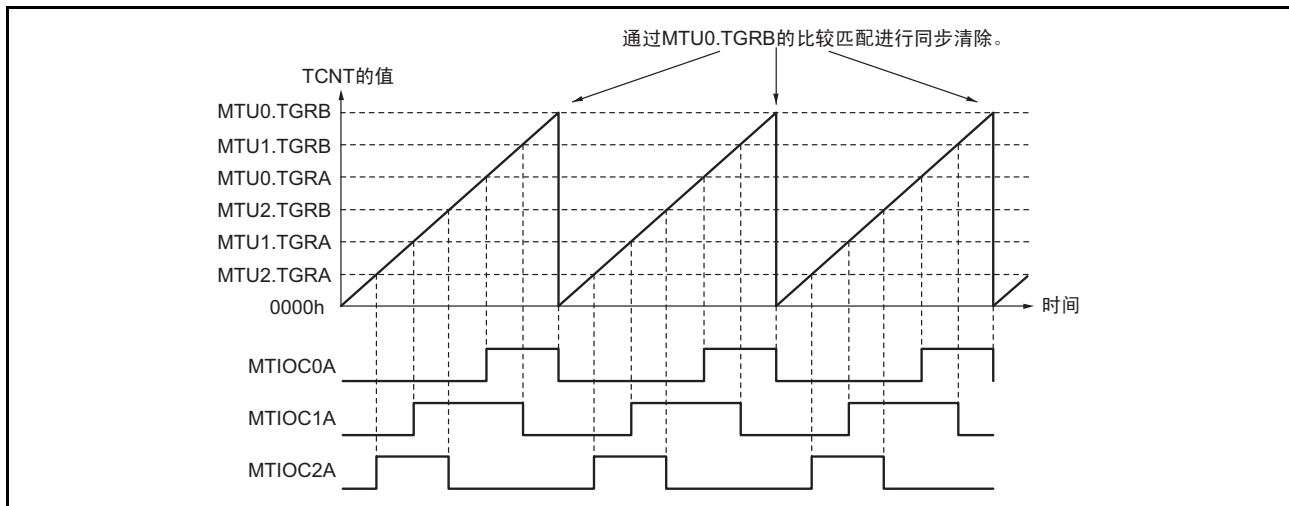


图 21.14 同步运行的例子

21.3.3 缓冲运行

缓冲运行是 MTU0、3、4、6、7 具有的功能，能将 TGRC 寄存器和 TGRD 寄存器用作缓冲寄存器，在 MTU0 还能将 TGRF 寄存器用作缓冲寄存器。

将 TGR 寄存器设定为输入捕捉寄存器和比较匹配寄存器时的缓冲运行内容不同。

注： MTU0.TGRE 不能被设定为输入捕捉寄存器，而只能用作比较匹配寄存器。

缓冲运行时的寄存器组合如表 21.57 所示。

表 21.57 寄存器的组合

通道	定时器的通用寄存器	缓冲寄存器
MTU0	TGRA	TGRC
	TGRB	TGRD
	TGRE	TGRF
MTU3	TGRA	TGRC
	TGRB	TGRD
MTU4	TGRA	TGRC
	TGRB	TGRD
MTU6	TGRA	TGRC
	TGRB	TGRD
MTU7	TGRA	TGRC
	TGRB	TGRD

- TGR 为输出比较寄存器的情况
如果发生比较匹配，就将对应通道的缓冲寄存器的值传送到定时器的通用寄存器。
此运行如图 21.15 所示。

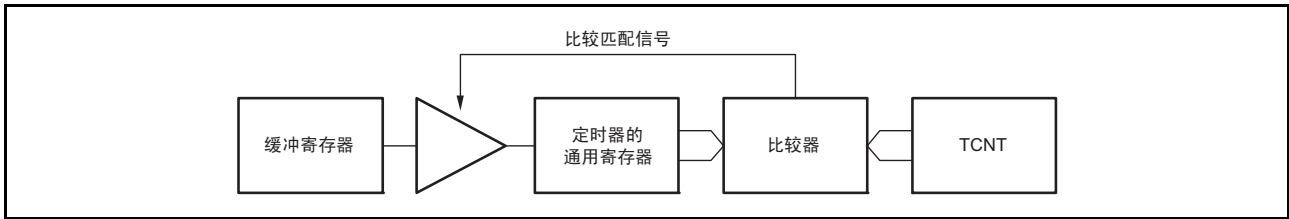


图 21.15 比较匹配的缓冲运行

- TGR 为输入捕捉寄存器的情况
如果发生输入捕捉，就在将 TCNT 的值传送到 TGR 寄存器的同时，将以前保存的 TGR 寄存器的值传送到缓冲寄存器。
此运行如图 21.16 所示。

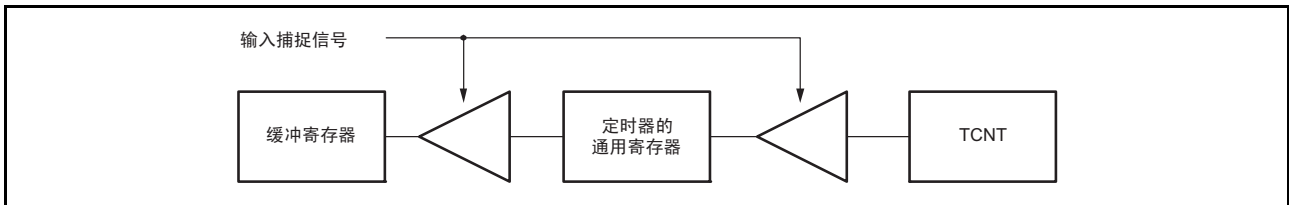


图 21.16 输入捕捉的缓冲运行

(1) 缓冲运行的设定步骤例子

缓冲运行的设定步骤例子如图 21.17 所示。

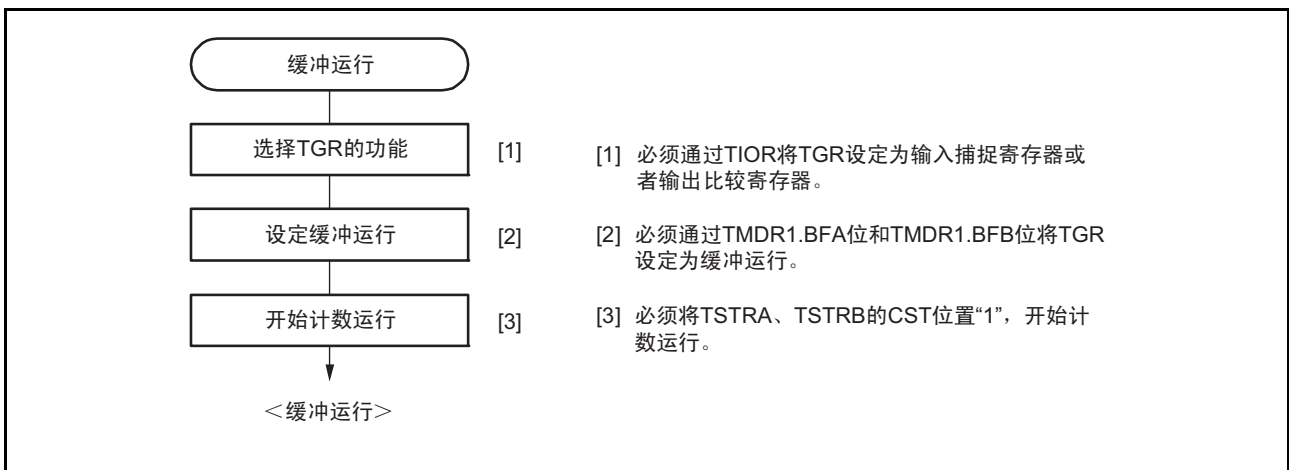


图 21.17 缓冲运行的设定步骤例子

(2) 缓冲运行的例子

(a) TGR 为输出比较寄存器的情况

将 MTU0 设定为 PWM 模式 1 并且将 TGRA 和 TGRC 设定为缓冲运行时的运行例子如图 21.18 所示。在此例子中，假设通过比较匹配 B 清除 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，将 TBTM.TTSA 位置“0”。

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

有关 PWM 模式，请参照“21.3.5 PWM 模式”。

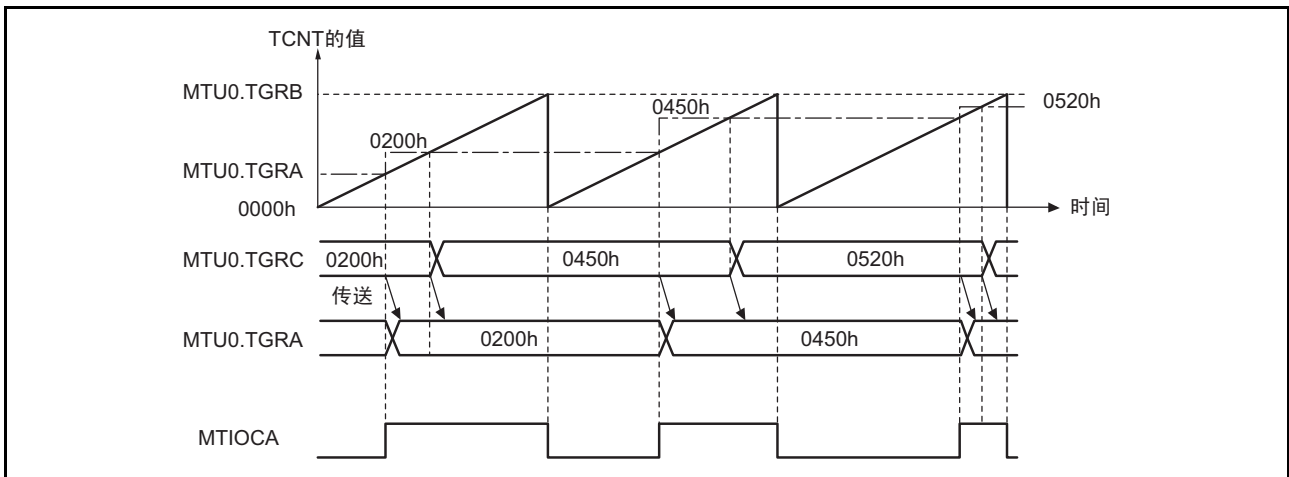


图 21.18 缓冲运行的例子 (1)

(b) TGR 为输入捕捉寄存器的情况

将 TGRA 设定为输入捕捉寄存器并且将 TGRA 和 TGRC 设定为缓冲运行时的运行例子如图 21.19 所示。

在 TGRA 的输入捕捉时清除 TCNT 计数器，选择上升沿 / 下降沿的双边沿作为 MPIOCA 引脚的输入捕捉的输入边沿。

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

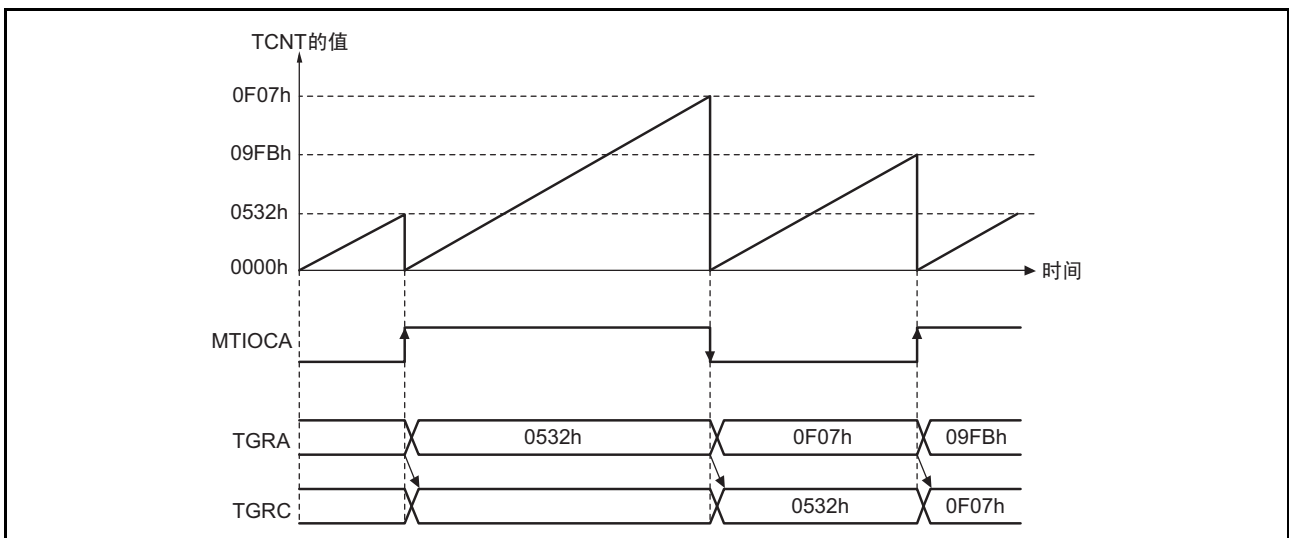


图 21.19 缓冲运行的例子 (2)

(3) 从缓冲运行时的缓冲寄存器到定时器的通用寄存器的传送时序选择

能通过设定缓冲运行传送模式寄存器 (MTUn.TBTM) (n=0、3、4、6、7)，选择 MTU0 为 PWM 模式 1 和 PWM 模式 2 时，以及 MTU3、4、6、7 为 PWM 模式 1 时的从缓冲寄存器到定时器的通用寄存器的传送时序。能选择在发生比较匹配时 (复位后的值) 或者在清除 TCNT 时进行缓冲传送。此时清除 TCNT 是指当以下的任意一个条件成立时。

- 在 TCNT 发生上溢时 (FFFFh→0000h)
- 在计数器运行过程中给 TCNT 写“0000h”时
- 通过 TCR.CCLR[2:0] 位设定的清除源使 TCNT 变为“0000h”时

注. 必须在 TCNT 停止计数的状态下设定 TBTM 寄存器。

将 MTU0 设定为 PWM 模式 1 并且将 MTU0.TGRA 和 MTU0.TGRC 设定为缓冲运行时的运行例子如图 21.20 所示。通过比较匹配 B 清除 MTU0.TCNT，通过比较匹配 A 输出 High 电平，通过比较匹配 B 输出 Low 电平，将 MTU0.TBTM.TTSA 位置“1”。

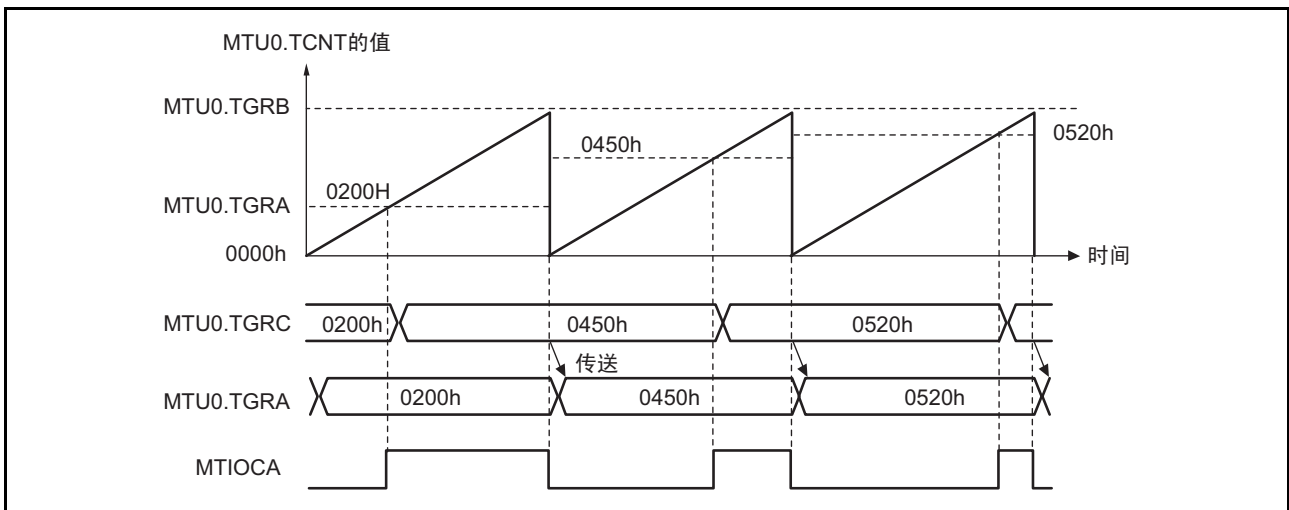


图 21.20 选择在清除 MTU0.TCNT 时进行从 MTU0.TGRC 到 MTU0.TGRA 的缓冲传送的运行例子

21.3.4 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。

通过 TCR 的 TPSC[2:0] 位将 MTU1 的计数器时钟设定为通过 MTU2.TCNT 的上溢 / 下溢进行计数，实现级联运行的功能。

只在低 16 位的 TCNT 为相位计数模式时发生下溢。

级联的组合如表 21.58 所示。

注. 如果将 MTU1 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 21.58 级联的组合

组合	高 16 位	低 16 位
MTU1 和 MTU2	MTU1.TCNT	MTU2.TCNT

在进行级联运行时，如果 MTU1.TCNT 和 MTU2.TCNT 同时进行输入捕捉，就能通过输入捕捉控制寄存器 (TICCR) 进行设定，将输入引脚追加到输入捕捉条件。对取得原有输入引脚输入电平和追加输入引脚输入电平的逻辑“或”的信号执行输入捕捉条件的边沿检测。因此，当其中任意一个为 High 电平时，即使另外一个发生变化，也不能执行边沿检测。详细内容请参照“(4) 级联运行的例子 (c)”。有关级联时的输入捕捉，请参照“21.6.21 级联中的 MTU1.TCNT 和 MTU2.TCNT 的同时输入捕捉”。

TICCR 的设定值和输入捕捉的输入引脚的对应如表 21.59 所示。

表 21.59 TICCR 寄存器的设定值和输入捕捉的输入引脚的对应

对象输入捕捉	TICCR 的设定值	输入捕捉的输入引脚
MTU1.TCNT 到 MTU1.TGRA 的输入捕捉	I2AE 位 =0 (复位后的值)	MTIOC1A
	I2AE 位 =1	MTIOC1A、MTIOC2A
MTU1.TCNT 到 MTU1.TGRB 的输入捕捉	I2BE 位 =0 (复位后的值)	MTIOC1B
	I2BE 位 =1	MTIOC1B、MTIOC2B
MTU2.TCNT 到 MTU2.TGRA 的输入捕捉	I1AE 位 =0 (复位后的值)	MTIOC2A
	I1AE 位 =1	MTIOC2A、MTIOC1A
MTU2.TCNT 到 MTU2.TGRB 的输入捕捉	I1BE 位 =0 (复位后的值)	MTIOC2B
	I1BE 位 =1	MTIOC2B、MTIOC1B

(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 21.21 所示。



图 21.21 级联运行的设定步骤例子

(2) 级联运行的例子 (a)

MTU1.TCNT 通过 MTU2.TCNT 的上溢/下溢进行计数并且将 MTU2 设定为相位计数模式时的运行如图 21.22 所示。

MTU1.TCNT 在 MTU2.TCNT 的上溢时进行递增计数，在 MTU2.TCNT 的下溢时进行递减计数。

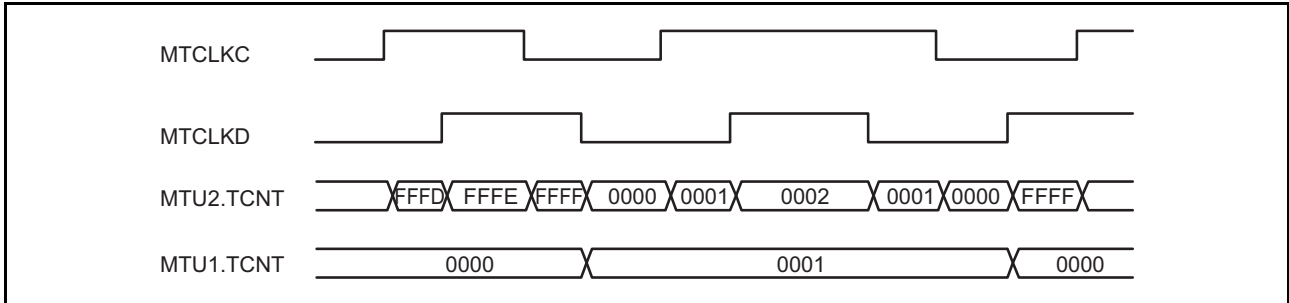


图 21.22 级联运行的例子 (a)

(3) 级联运行的例子 (b)

在将 MTU1.TCNT 和 MTU2.TCNT 进行级联，并且通过将 TICCR.I2AE 位置“1”使 MTIOC2A 引脚追加到 MTU1.TGRA 输入捕捉条件时的运行如图 21.23 所示。在此例子中，将 MTU1.TIOR.IOA[3:0] 位设定为在 (MTIOC1A 的) 上升沿进行输入捕捉，将 MTU2.TIOR.IOA[3:0] 位设定为在 (MTIOC2A 的) 上升沿进行输入捕捉。

此时，MTIOC1A 和 MTIOC2A 的上升沿被设定为 MTU1.TGRA 的输入捕捉条件，MTIOC2A 的上升沿被设定为 MTU2.TGRA 的输入捕捉条件。

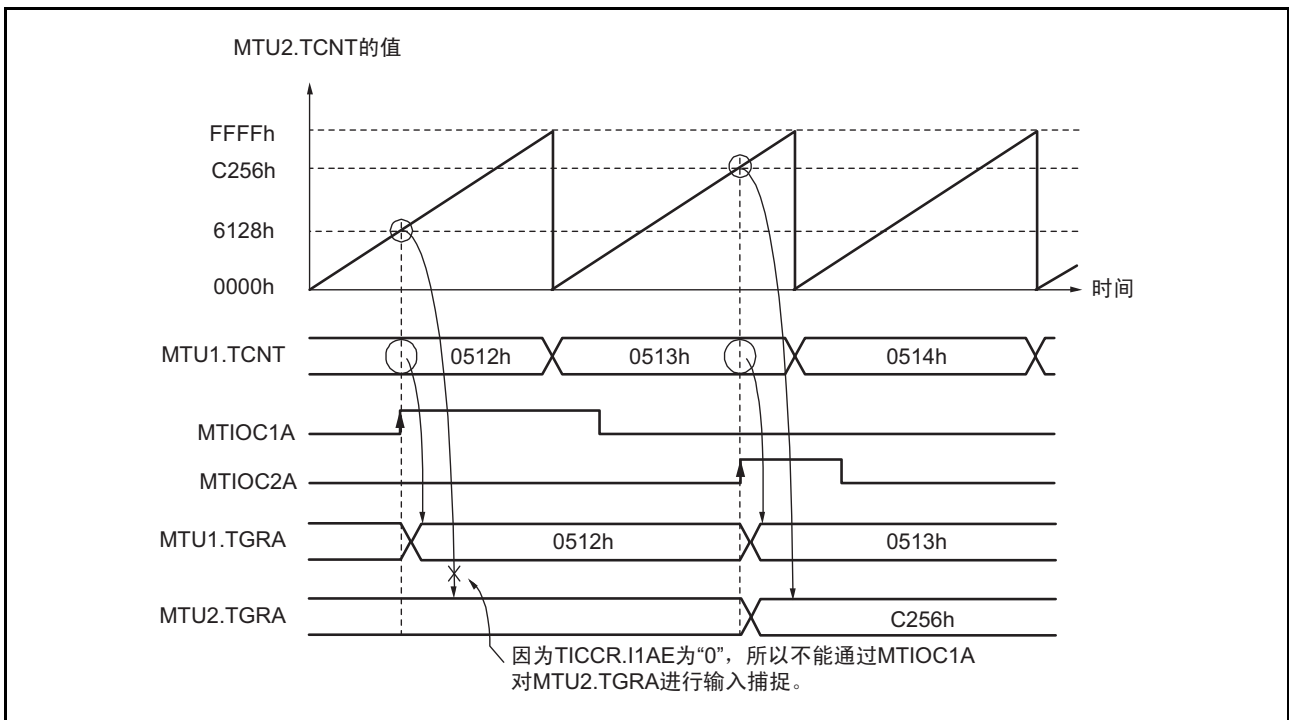


图 21.23 级联运行的例子 (b)

(4) 级联运行的例子 (c)

将 MTU1.TCNT 和 MTU2.TCNT 进行级联，并且通过将 TICCR.I2AE 位和 TICCR.I1AE 位置“1”使 MTIOC2A 引脚追加到 MTU1.TGRA 的输入捕捉条件以及使 MTIOC1A 引脚追加到 MTU2.TGRA 的输入捕捉条件时的运行如图 21.24 所示。在此例子中，将 MTU1.TIOR 和 MTU2.TIOR 的 IOA[3:0] 位都设定为在双边沿进行输入捕捉。此时，MTIOC1A 输入和 MTIOC2A 输入的 OR 为 MTU1.TGRA 和 MTU2.TGRA 的输入捕捉条件。

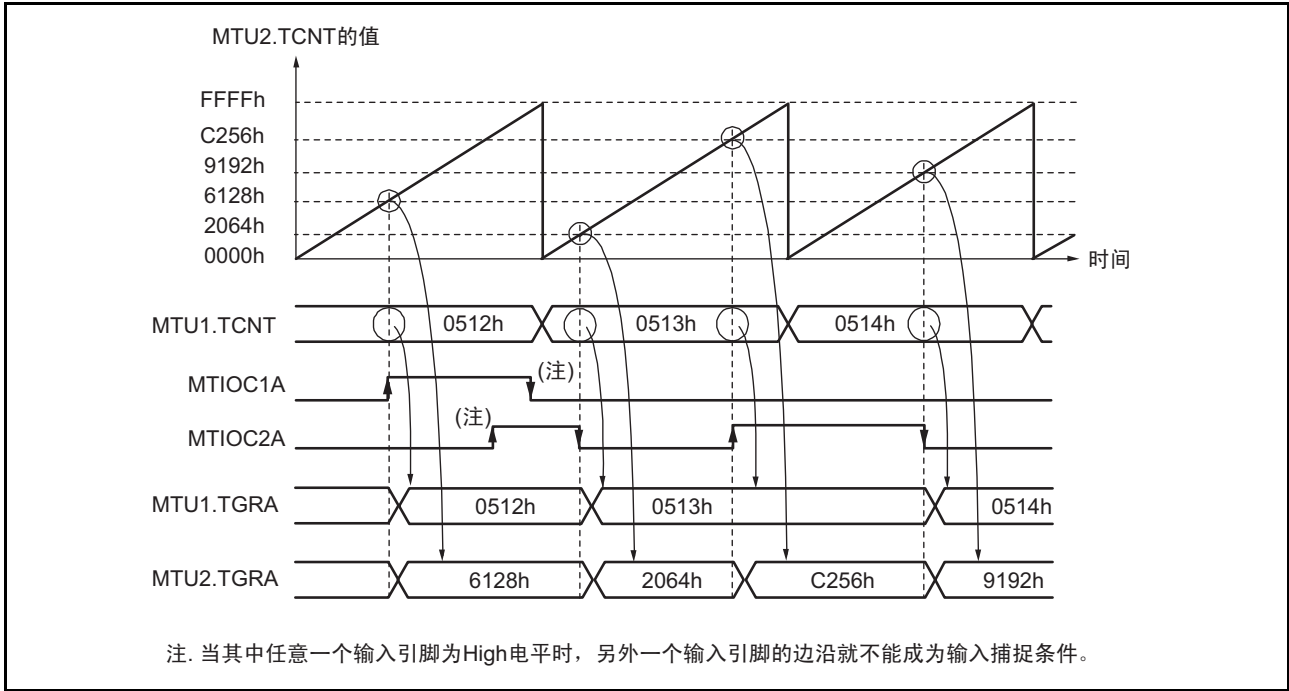


图 21.24 级联运行的例子 (c)

(5) 级联运行的例子 (d)

在将 MTU1.TCNT 和 MTU2.TCNT 进行级联，并且通过将 TICCR.I2AE 位置“1”使 MTIOC2A 引脚追加到 MTU1.TGRA 的输入捕捉条件时的运行如图 21.25 所示。在此例子中，将 MTU1.TIOR 的 IOA[3:0] 位设定为在发生 MTU0.TGRA 的比较匹配 / 输入捕捉时进行输入捕捉，将 MTU2.TIOR 的 IOA[3:0] 位设定为在 (MTIOC2A 的) 上升沿进行输入捕捉。

此时，因为将 MTU1.TIOR 设定为在发生 MTU0.TGRA 的比较匹配 / 输入捕捉时进行输入捕捉，所以即使将 TICCR.I2AE 位置“1”，MTIOC2A 的边沿也不会成为 MTU1.TGRA 的输入捕捉条件。

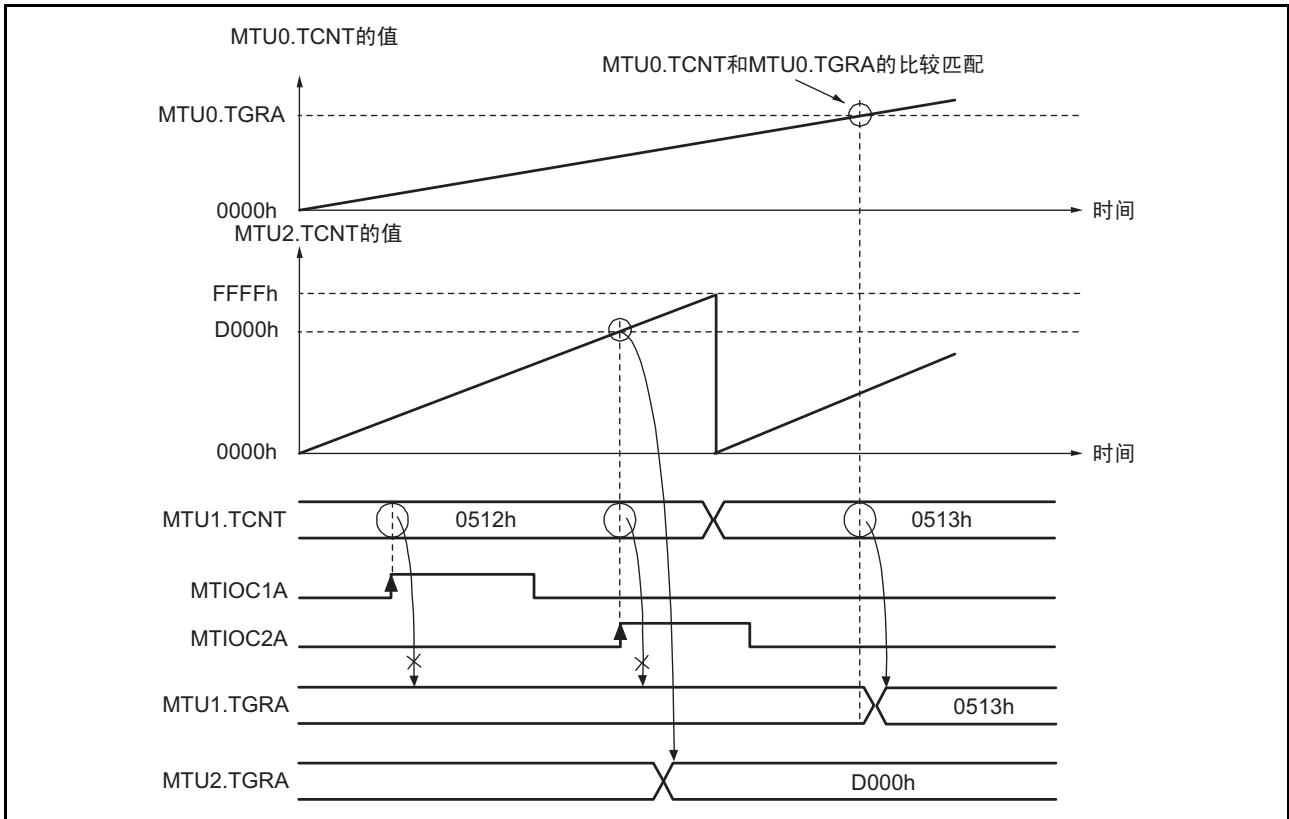


图 21.25 级联运行的例子 (d)

21.3.5 PWM 模式

PWM 模式是从输出引脚分别输出 PWM 波形的模式。各 TGR 寄存器的比较匹配的输出现平可选择 Low 电平输出、High 电平输出或者交替输出。

能通过设定各 TGR 寄存器，输出 0 ~ 100% 占空比的 PWM 波形。

能通过将 TGR 寄存器的比较匹配作为计数器的清除源，给该寄存器设定周期。能将全部通道分别设定为 PWM 模式，也能进行同步运行。

PWM 模式有以下 2 种模式：

(a) PWM 模式 1

TGRA 寄存器和 TGRB 寄存器、TGRC 寄存器和 TGRD 寄存器配对使用，从 MTIOCA 引脚和 MTIOCC 引脚输出 PWM 波形。通过比较匹配 A 和比较匹配 C 从 MTIOCA 引脚和 MTIOCC 引脚进行 TIOR.IOA[3:0] 位和 TIOR.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 MTIOCA 引脚和 MTIOCC 引脚进行 TIOR.IOB[3:0] 位和 TIOR.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。当配对使用的 TGR 寄存器的设定值相同时，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。

注. MTIOC3A 和 MTIOC6A、MTIOC3C 和 MTIOC6C、MTIOC4A 和 MTIOC7A 以及 MTIOC4C 和 MTIOC7C 分别复用相同引脚，因此能通过 MPC 选择其中一个使用。

(b) PWM 模式 2

将 1 个 TGR 寄存器用作周期寄存器而将其他 TGR 寄存器用作占空比寄存器，进行 PWM 波形输出。通过比较匹配进行 TIOR 寄存器指定的输出，并且通过周期寄存器的比较匹配进行计数器清除，各引脚的输出值为 TIOR 寄存器设定的初始值。如果周期寄存器和占空比寄存器的设定值相同，即使发生比较匹配，输出值也不变。

在 PWM 模式 2 中，能通过和同步运行的并用，进行最多 8 相的 PWM 波形输出。

PWM 的输出引脚和寄存器的对应如表 21.60 所示。

表 21.60 各 PWM 输出的寄存器和输出引脚

通道	寄存器	输出引脚	
		PWM 模式 1	PWM 模式 2
MTU0	TGRA	MTIOC0A	MTIOC0A
	TGRB		MTIOC0B
	TGRC	MTIOC0C	MTIOC0C
	TGRD		MTIOC0D
MTU1	TGRA	MTIOC1A	MTIOC1A
	TGRB		MTIOC1B
MTU2	TGRA	MTIOC2A	MTIOC2A
	TGRB		MTIOC2B
MTU3	TGRA	MTIOC3A	不能设定。
	TGRB		
	TGRC	MTIOC3C	
	TGRD		
MTU4	TGRA	MTIOC4A	
	TGRB		
	TGRC	MTIOC4C	
	TGRD		
MTU6	TGRA	MTIOC6A	
	TGRB		
	TGRC	MTIOC6C	
	TGRD		
MTU7	TGRA	MTIOC7A	
	TGRB		
	TGRC	MTIOC7C	
	TGRD		

注. 在 PWM 模式 2 中, 不能输出已设定周期的 TGR 的 PWM 波形。

(1) PWM 模式的设定步骤例子

PWM 模式的设定步骤例子如图 21.26 所示。

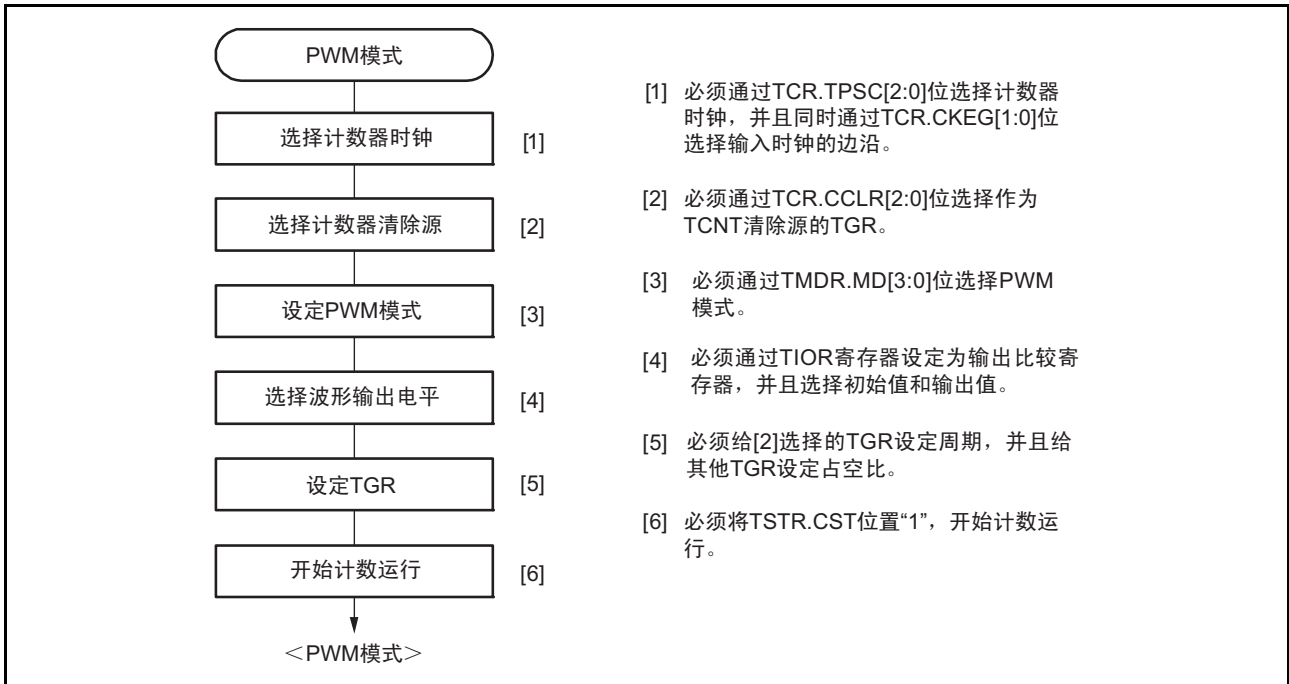


图 21.26 PWM 模式的设定步骤例子

(2) PWM 模式的运行例子

PWM 模式 1 的运行例子如图 21.27 所示。

在此例子中，假设将 TGRA 寄存器的比较匹配设定为 TCNT 的清除源，将 TGRA 寄存器的初始输出值和输出值设定为 Low 电平，将 TGRB 寄存器的输出值设定为“1”。

此时，TGRA 寄存器的设定值为周期，TGRB 的设定值为占空比。

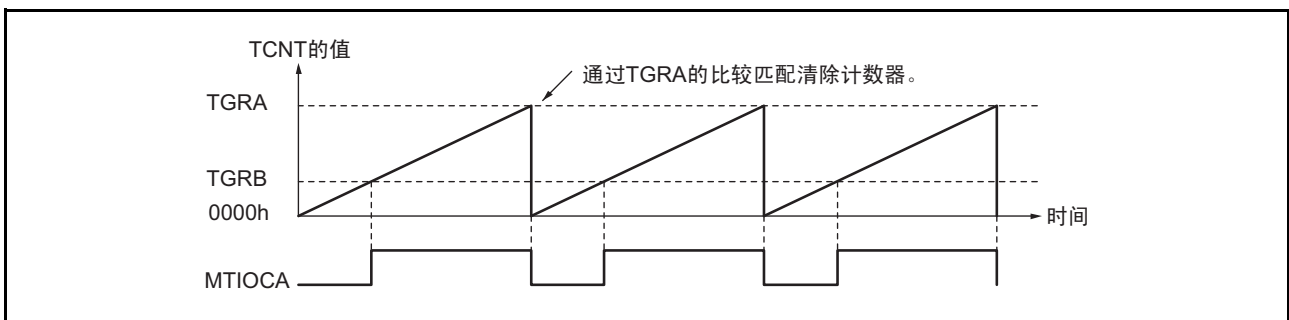


图 21.27 PWM 模式 1 的运行例子

PWM 模式 2 的运行例子如图 21.28 所示。

在此例子中，假设使 MTU0 和 MTU1 同步运行，将 MTU1.TGRB 寄存器的比较匹配设定为 TCNT 的清除源，将其他 TGR 寄存器 (MTU0.TGRA ~ MTU0.TGRD、MTU1.TGRA) 的初始输出值设定为“0”，将输出值设定为“1”，输出 5 相 PWM 波形。

此时，MTU1.TGRB 的设定值为周期，其他 TGR 寄存器 (MTU0.TGRA ~ MTU0.TGRD、MTU1.TGRA) 的设定值为各输出的占空比。

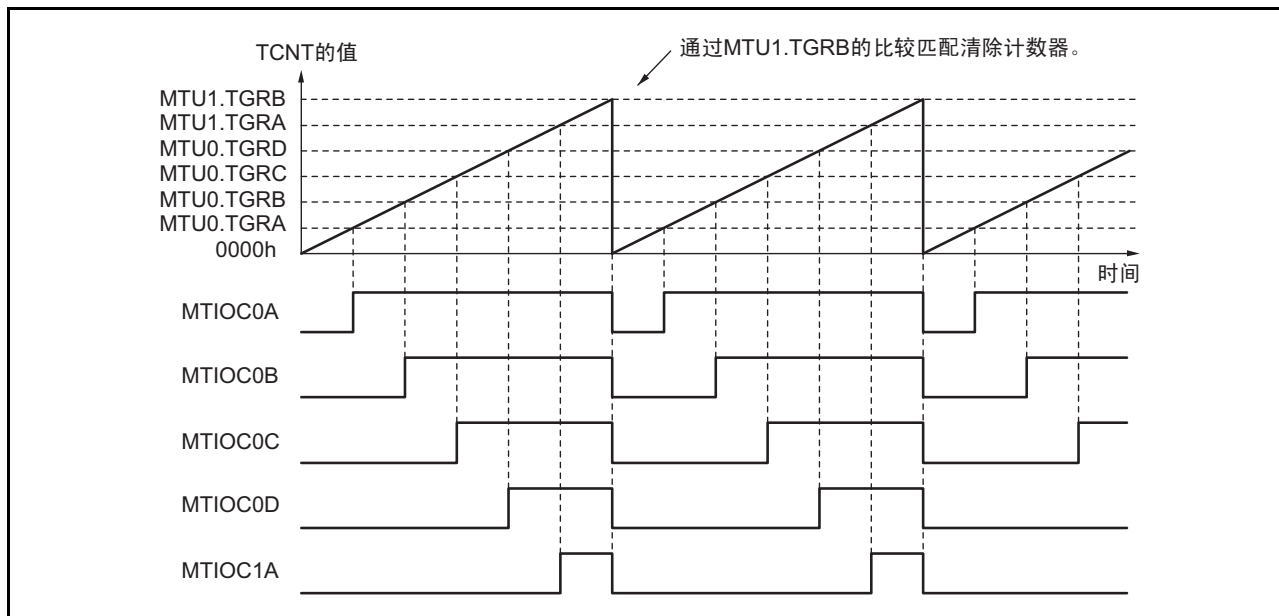


图 21.28 PWM 模式 2 的运行例子

在 PWM 模式中，占空比为 0% 和 100% 的 PWM 波形输出例子如图 21.29 所示。

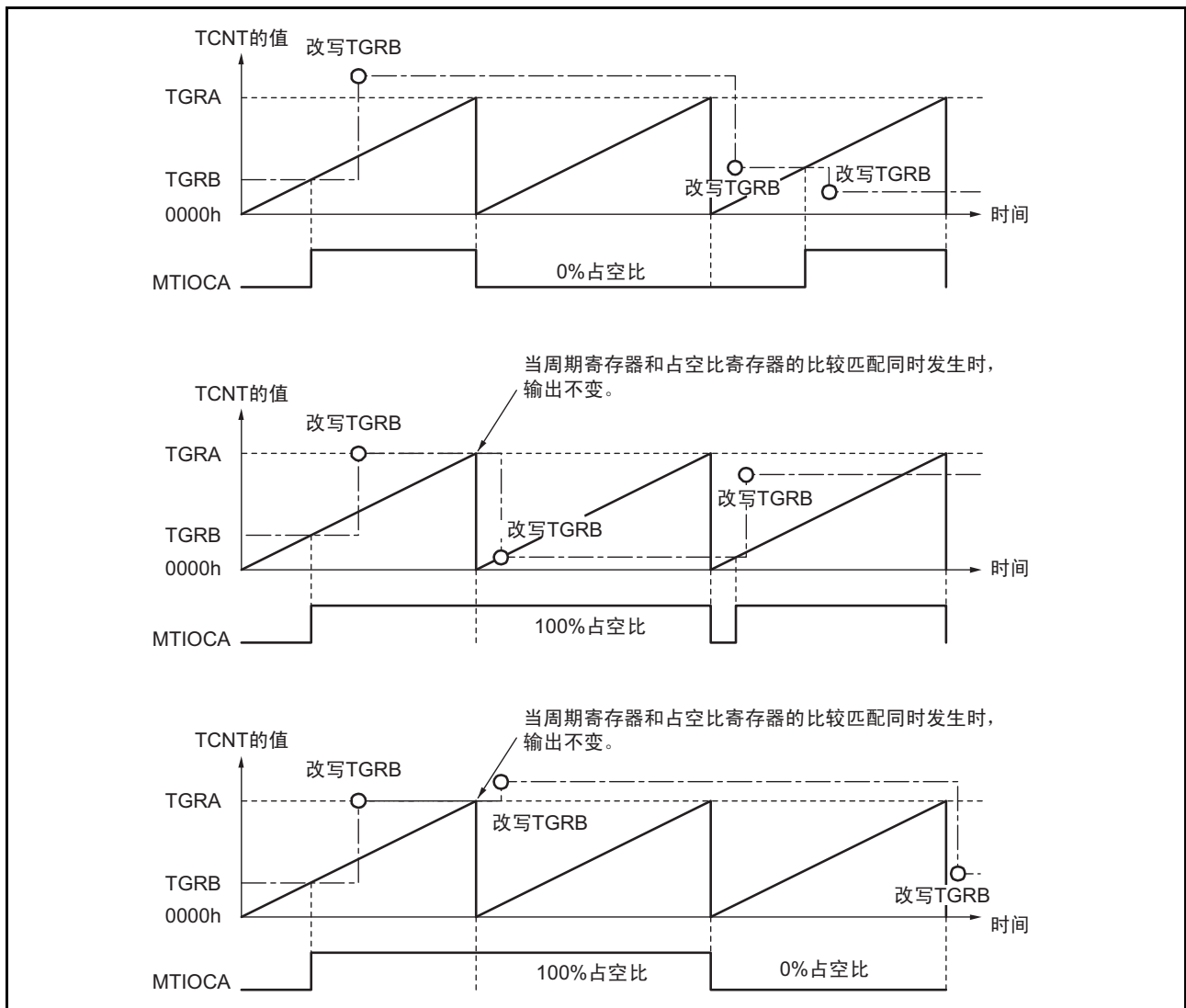


图 21.29 PWM 模式的运行例子 (0% 占空比和 100% 占空比的 PWM 波形输出例子)

21.3.6 相位计数模式

在相位计数模式中，通过设定 MTU1 和 MTU2，检测 2 个外部时钟输入的相位差，并且 TCNT 进行递增 / 递减计数。

如果设定为相位计数模式，就选择外部时钟作为计数器输入时钟，并且 TCNT 作为递增 / 递减计数器运行，与 TCR.TPSC[2:0] 位和 CKEG[1:0] 位的设定无关。但是，因为 TCR.CCLR[1:0] 位、TIOR 寄存器、TIER 寄存器、TGR 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。

如果在 TCNT 进行递增计数时发生上溢，TSR 寄存器的 TCFV 标志就被置位。如果在 TCNT 进行递减计数时发生下溢，TSR 寄存器的 TCFU 标志就被置位。

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

外部时钟引脚和通道的对应如表 21.61 所示。

表 21.61 相位计数模式的时钟输入引脚

通道	外部时钟引脚	
	A 相	B 相
将 MTU1 设定为相位计数模式的情况	MTCLKA	MTCLKB
将 MTU2 设定为相位计数模式的情况	MTCLKC	MTCLKD

(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 21.30 所示。

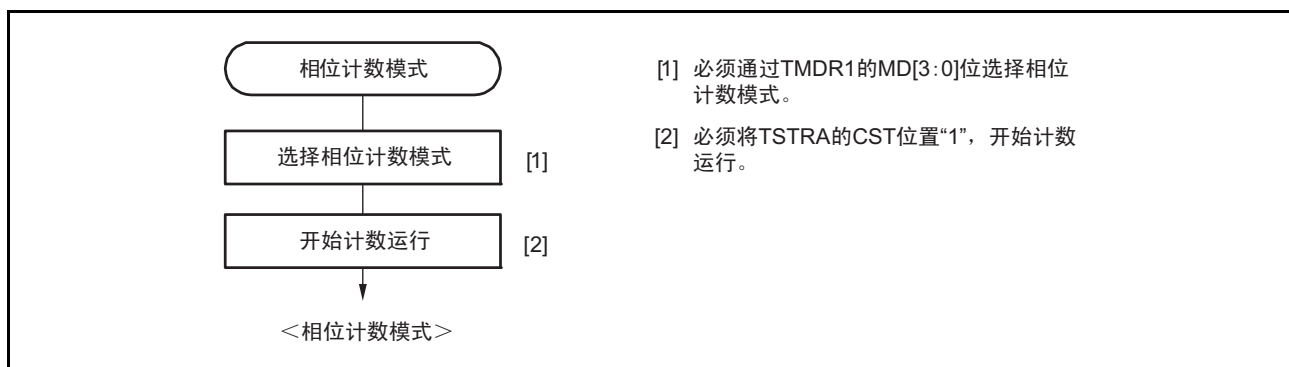


图 21.30 相位计数模式的设定步骤例子

(2) 相位计数模式的运行例子

在相位计数模式中，TCNT 通过 2 个外部时钟的相位差进行递增计数或者递减计数。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 21.31 所示，TCNT 的递增 / 递减计数条件如表 21.62 所示。

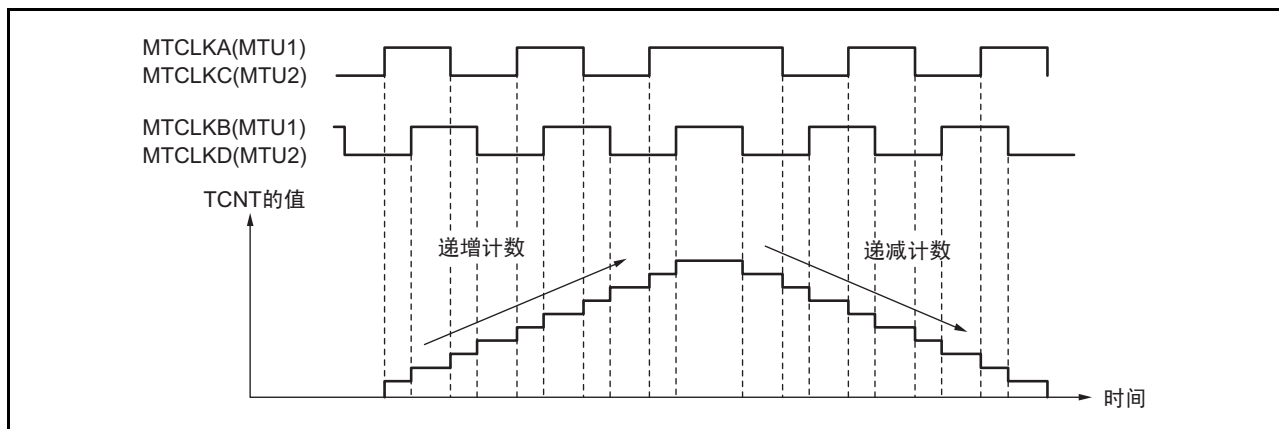


图 21.31 相位计数模式 1 的运行例子

表 21.62 相位计数模式 1 的递增 / 递减计数条件

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	运行内容
High 电平	上升沿	递增计数
Low 电平	下降沿	
上升沿	Low 电平	
下降沿	High 电平	
High 电平	下降沿	递减计数
Low 电平	上升沿	
上升沿	High 电平	
下降沿	Low 电平	

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 21.32 所示，TCNT 的递增 / 递减计数条件如表 21.63 所示。

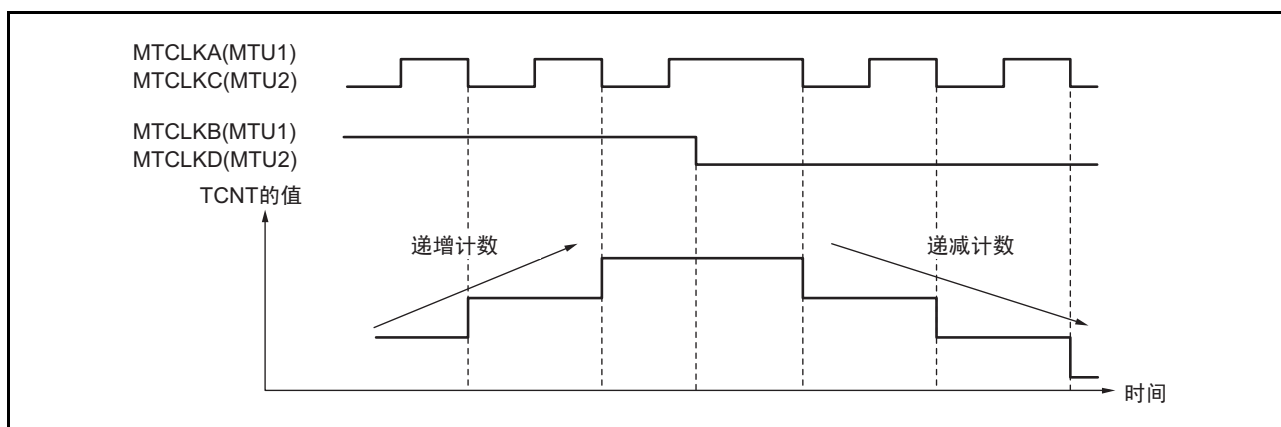


图 21.32 相位计数模式 2 的运行例子

表 21.63 相位计数模式 2 的递增 / 递减计数条件

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	运行内容
High 电平	↑	不计数 (Don't care)
Low 电平	↓	
↑	Low 电平	
↓	High 电平	递增计数
High 电平	↑	不计数 (Don't care)
Low 电平	↓	
↑	High 电平	
↓	Low 电平	递减计数

↑ : 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 21.33 所示，TCNT 的递增 / 递减计数条件如表 21.64 所示。

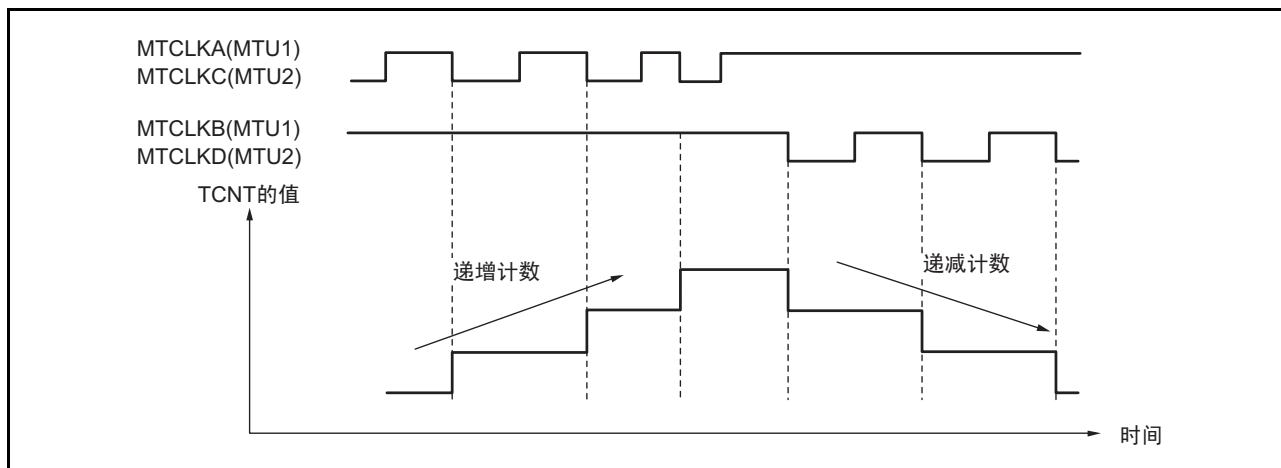


图 21.33 相位计数模式 3 的运行例子

表 21.64 相位计数模式 3 的递增 / 递减计数条件

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	运行内容
High 电平	上升沿	不计数 (Don't care)
Low 电平	下降沿	
上升沿	Low 电平	递增计数
下降沿	High 电平	
High 电平	下降沿	递减计数
Low 电平	上升沿	不计数 (Don't care)
上升沿	High 电平	
下降沿	Low 电平	

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 21.34 所示，TCNT 的递增 / 递减计数条件如表 21.65 所示。

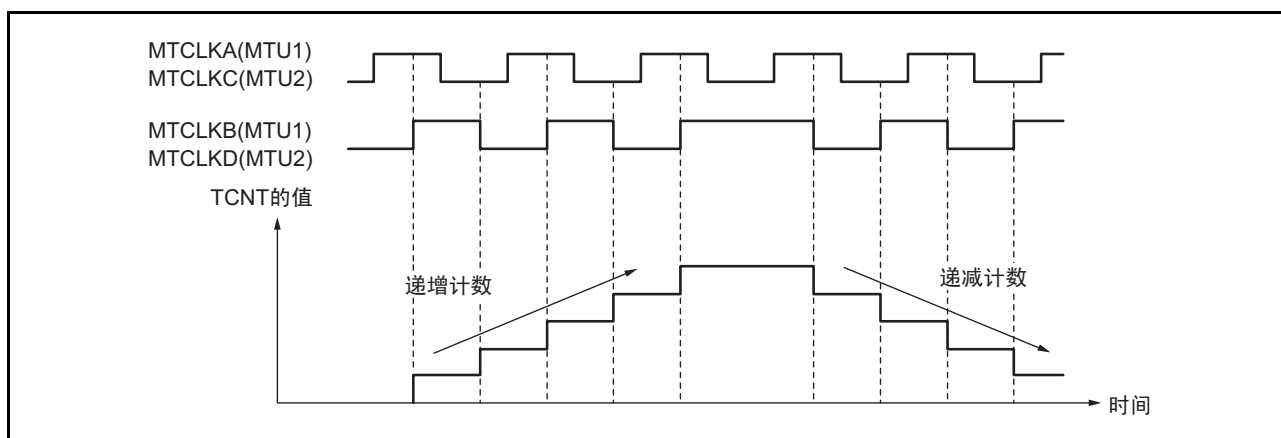


图 21.34 相位计数模式 4 的运行例子

表 21.65 相位计数模式 4 的递增 / 递减计数条件

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	运行内容
High 电平	↑	递增计数
Low 电平	↓	
↑	Low 电平	不计数 (Don't care)
↓	High 电平	
High 电平	↓	递减计数
Low 电平	↑	
↑	High 电平	不计数 (Don't care)
↓	Low 电平	

↑ : 上升沿

↓ : 下降沿

(3) 相位计数模式的应用例子

将 MTU1 设定为相位计数模式，并且在与 MTU0 组合输入伺服马达 2 相编码器脉冲后检测位置或者速度的例子如图 21.35 所示。

将 MTU1 设定为相位计数模式 1，给 MTCLKA 和 MTCLKB 输入编码器脉冲的 A 相和 B 相。

MTU0 通过 MTU0.TGRC 的比较匹配进行 TCNT 计数器清除，MTU0.TGRA 和 MTU0.TGRC 用于比较匹配功能，设定速度控制周期和位置控制周期。MTU0.TGRB 用于输入捕捉功能，使 MTU0.TGRB 和 MTU0.TGRD 进行缓冲运行。将 MTU1 的计数器输入时钟作为 MTU0.TGRB 的输入捕捉源，检测 2 相编码器的 4 倍频脉冲的脉宽。

将 MTU1 的 MTU1.TGRA 和 MTU1.TGRB 设定为输入捕捉功能，选择 MTU0 的 MTU0.TGRA 和 MTU0.TGRC 的比较匹配作为输入捕捉源，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。

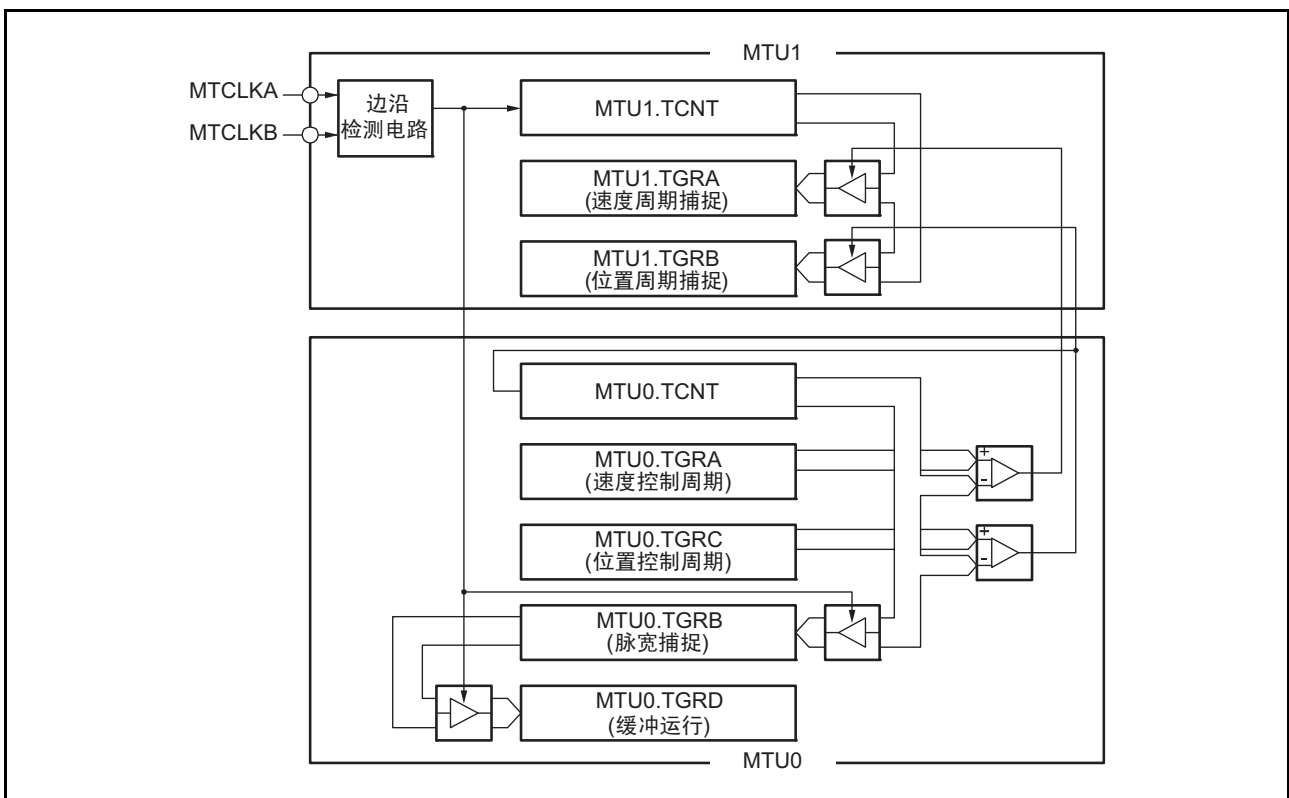


图 21.35 相位计数模式的应用例子

21.3.7 复位同步 PWM 模式

在复位同步 PWM 模式中，通过组合 MTU3 和 MTU4 以及 MTU6 和 MTU7，将一方的波形变化点作为共同关系的 PWM 波形进行（正相和反相）各 3 相共计 6 相输出。

如果设定为复位同步 PWM 模式，MTIOC3B、MTIOC3D、MTIOC4A、MTIOC4C、MTIOC4B、MTIOC4D、MTIOC6B、MTIOC6D、MTIOC7A、MTIOC7C、MTIOC7B 和 MTIOC7D 引脚就为 PWM 输出引脚，定时器的计数器 3 和计数器 6 (MTU3.TCNT、MTU6.TCNT) 用作递增计数器。

使用的 PWM 的输出引脚如表 21.66、使用的寄存器的设定如表 21.67 所示。

表 21.66 复位同步 PWM 模式中的输出引脚

通道	输出引脚	说明
MTU3	MTIOC3B	PWM 的输出引脚 1
	MTIOC3D	PWM 的输出引脚 1' (PWM 输出 1 的反相波形)
MTU4	MTIOC4A	PWM 的输出引脚 2
	MTIOC4C	PWM 的输出引脚 2' (PWM 输出 2 的反相波形)
	MTIOC4B	PWM 的输出引脚 3
	MTIOC4D	PWM 的输出引脚 3' (PWM 输出 3 的反相波形)
MTU6	MTIOC6B	PWM 的输出引脚 4
	MTIOC6D	PWM 的输出引脚 4' (PWM 输出 4 的反相波形)
MTU7	MTIOC7A	PWM 的输出引脚 5
	MTIOC7C	PWM 的输出引脚 5' (PWM 输出 5 的反相波形)
	MTIOC7B	PWM 的输出引脚 6
	MTIOC7D	PWM 的输出引脚 6' (PWM 输出 6 的反相波形)

表 21.67 复位同步 PWM 模式中的寄存器设定

寄存器	设定内容
MTU3.TCNT	初始设定“0000h”。
MTU4.TCNT	初始设定“0000h”。
MTU3.TGRA	设定 MTU3.TCNT 的计数周期。
MTU3.TGRB	设定从 MTIOC3B 引脚和 MTIOC3D 引脚输出的 PWM 波形变化点。
MTU4.TGRA	设定从 MTIOC4A 引脚和 MTIOC4C 引脚输出的 PWM 波形变化点。
MTU4.TGRB	设定从 MTIOC4B 引脚和 MTIOC4D 引脚输出的 PWM 波形变化点。
MTU6.TCNT	初始设定“0000h”。
MTU7.TCNT	初始设定“0000h”。
MTU6.TGRA	设定 MTU6.TCNT 的计数周期。
MTU6.TGRB	设定从 MTIOC6B 引脚和 MTIOC6D 引脚输出的 PWM 波形变化点。
MTU7.TGRA	设定从 MTIOC7A 引脚和 MTIOC7C 引脚输出的 PWM 波形变化点。
MTU7.TGRB	设定从 MTIOC7B 引脚和 MTIOC7D 引脚输出的 PWM 波形变化点。

(1) 复位同步 PWM 模式的设定步骤例子

复位同步 PWM 模式的设定步骤例子如图 21.36 所示。

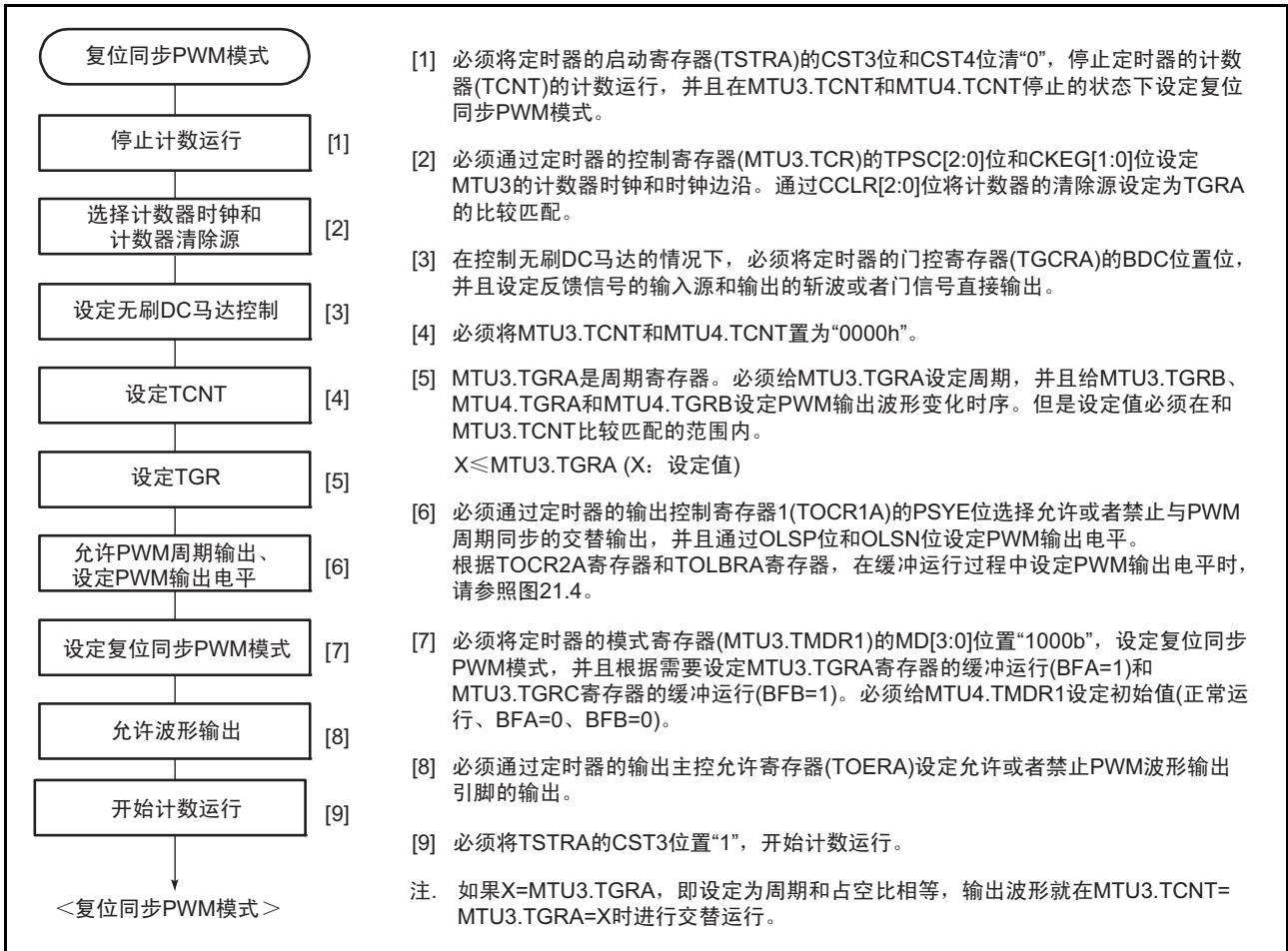


图 21.36 复位同步 PWM 模式的设定步骤例子

(2) 复位同步 PWM 模式的运行例子

复位同步 PWM 模式的运行例子如图 21.37 所示。

在复位同步 PWM 模式中，MTU3.TCNT 和 MTU4.TCNT (MTU6.TCNT 和 MTU7.TCNT) 作为递增计数器运行。如果 MTU3.TCNT (MTU6.TCNT) 和 MTU3.TGRA (MTU6.TGRA) 比较匹配，就清除计数器，并且从“0000h”重新开始递增计数。每当发生各 MTU3.TGRB (MTU6.TGRB)、MTU4.TGRA (MTU7.TGRA) 和 MTU4.TGRB (MTU7.TGRB) 的比较匹配和计数器清除时，PWM 输出引脚进行交替输出。

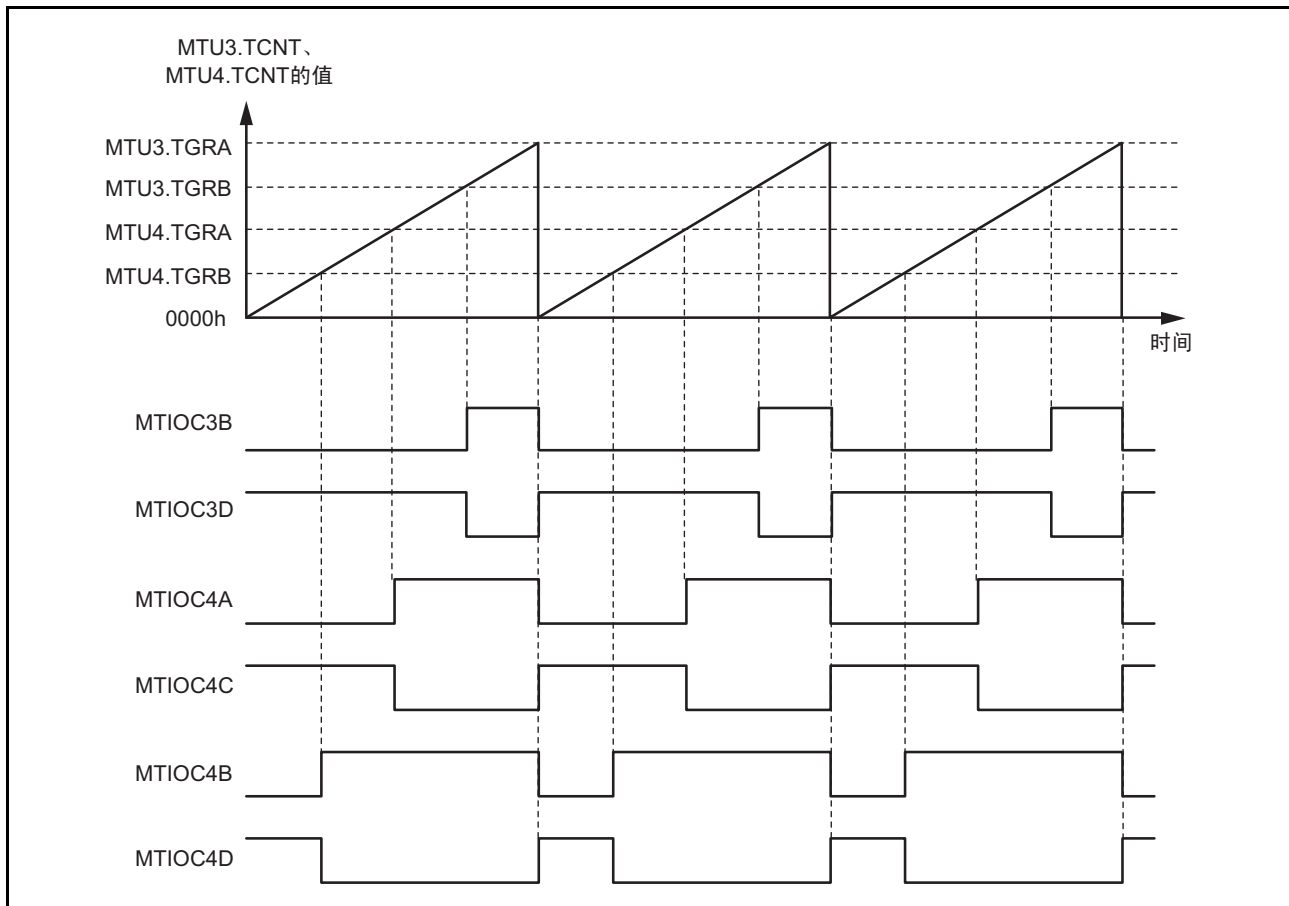


图 21.37 复位同步 PWM 模式的运行例子 (MTU3 和 MTU4、TOCR1A 的 OLSN=1、OLSP=1 的情况)

21.3.8 互补 PWM 模式

在互补 PWM 模式中，通过组合 MTU3 和 MTU4 以及 MTU6 和 MTU7，对正相和反相为非重叠关系的 PWM 波形进行各 3 相共计 6 相输出，也能设定为没有非重叠时间。

如果设定为互补 PWM 模式，MTIOC3B、MTIOC3D、MTIOC4A、MTIOC4B、MTIOC4C、MTIOC4D、MTIOC6B、MTIOC6D、MTIOC7A、MTIOC7B、MTIOC7C 和 MTIOC7D 引脚就为 PWM 输出引脚，也能将 MTIOC3A 引脚和 MTIOC6A 引脚设定为与 PWM 周期同步的交替输出。

MTU3.TCNT、MTU4.TCNT、MTU6.TCNT 和 MTU7.TCNT 用作递增 / 递减计数器。

使用的 PWM 的输出引脚如表 21.68、使用的寄存器的设定如表 21.69 所示。

作为端口功能，支持通过外部信号直接关闭 PWM 输出的功能。

表 21.68 互补 PWM 模式中的输出引脚

通道	输出引脚	说明
MTU3	MTIOC3A	与 PWM 周期同步的交替输出 (或者输入 / 输出端口)
	MTIOC3B	PWM 输出引脚 1
	MTIOC3C	输入 / 输出端口 (注 1)
	MTIOC3D	PWM 输出引脚 1' (与 PWM 输出 1 有非重叠关系的反相波形, 也能设定为没有非重叠时间)
MTU4	MTIOC4A	PWM 输出引脚 2
	MTIOC4C	PWM 输出引脚 2' (与 PWM 输出 2 有非重叠关系的反相波形, 也能设定为没有非重叠时间)
	MTIOC4B	PWM 输出引脚 3
	MTIOC4D	PWM 输出引脚 3' (与 PWM 输出 3 有非重叠关系的反相波形, 也能设定为没有非重叠时间)
MTU6	MTIOC6A	与 PWM 周期同步的交替输出 (或者输入 / 输出端口)
	MTIOC6B	PWM 输出引脚 4
	MTIOC6C	输入 / 输出端口 (注 1)
	MTIOC6D	PWM 输出引脚 4' (与 PWM 输出 4 有非重叠关系的反相波形, 也能设定为没有非重叠时间)
MTU7	MTIOC7A	PWM 输出引脚 5
	MTIOC7C	PWM 输出引脚 5' (与 PWM 输出 5 有非重叠关系的反相波形, 也能设定为没有非重叠时间)
	MTIOC7B	PWM 输出引脚 6
	MTIOC7D	PWM 输出引脚 6' (与 PWM 输出 6 有非重叠关系的反相波形, 也能设定为没有非重叠时间)

注 1. 在互补 PWM 模式中，不能将 MTIOC3C 引脚和 MTIOC6C 引脚设定为定时器的输入 / 输出引脚。

表 21.69 互补 PWM 模式中的寄存器设定 (1/2)

通道	计数器 / 寄存器	说明	由 CPU 进行读写
MTU3	TCNT	从死区时间寄存器的设定值开始递增计数	能通过设定 TRWERA 寄存器进行屏蔽 (注1)
	TGRA	设定 MTU3.TCNT 的上限值 (1/2 个载波周期 + 死区时间)	能通过设定 TRWERA 寄存器进行屏蔽 (注1)
	TGRB	PWM 输出 1 的比较寄存器	能通过设定 TRWERA 寄存器进行屏蔽 (注1)
	TGRC	MTU3.TGRA 的缓冲寄存器	随时能读写
	TGRD	PWM 输出 1/MTU3.TGRB 的缓冲寄存器	随时能读写
	TGRE	MTU3.TGRB 的缓冲寄存器 B (使用双缓冲功能时)	随时能读写
MTU4	TCNT	初始设定 "0000h" 并且开始递增计数	能通过设定 TRWERA 寄存器进行屏蔽 (注1)
	TGRA	PWM 输出 2 的比较寄存器	能通过设定 TRWERA 寄存器进行屏蔽 (注1)
	TGRB	PWM 输出 3 的比较寄存器	能通过设定 TRWERA 寄存器进行屏蔽 (注1)
	TGRC	PWM 输出 2/MTU4.TGRA 的缓冲寄存器	随时能读写
	TGRD	PWM 输出 3/MTU4.TGRB 的缓冲寄存器	随时能读写
	TGRE	MTU4.TGRA 的缓冲寄存器 B (使用双缓冲功能时)	随时能读写
	TGRF	MTU4.TGRB 的缓冲寄存器 B (使用双缓冲功能时)	随时能读写
MTU6	TCNT	从死区时间寄存器的设定值开始递增计数	能通过设定 TRWERB 寄存器进行屏蔽 (注2)
	TGRA	设定 MTU6.TCNT 的上限值 (1/2 个载波周期 + 死区时间)	能通过设定 TRWERB 寄存器进行屏蔽 (注2)
	TGRB	PWM 输出 4 的比较寄存器	能通过设定 TRWERB 寄存器进行屏蔽 (注2)
	TGRC	MTU6.TGRA 的缓冲寄存器	随时能读写
	TGRD	PWM 输出 4/MTU6.TGRB 的缓冲寄存器	随时能读写
	TGRE	MTU6.TGRB 的缓冲寄存器 B (使用双缓冲功能时)	随时能读写
MTU7	TCNT	初始设定 "0000h" 并且开始递增计数	能通过设定 TRWERB 寄存器进行屏蔽 (注2)
	TGRA	PWM 输出 5 的比较寄存器	能通过设定 TRWERB 寄存器进行屏蔽 (注2)
	TGRB	PWM 输出 6 的比较寄存器	能通过设定 TRWERB 寄存器进行屏蔽 (注2)
	TGRC	PWM 输出 5/MTU7.TGRA 的缓冲寄存器	随时能读写
	TGRD	PWM 输出 6/MTU7.TGRB 的缓冲寄存器	随时能读写
	TGRE	MTU7.TGRA 的缓冲寄存器 B (使用双缓冲功能时)	随时能读写
	TGRF	MTU7.TGRB 的缓冲寄存器 B (使用双缓冲功能时)	随时能读写

注 1. 能通过设定 TRWERA 寄存器 (定时器的读写允许寄存器 A), 允许或者禁止存取。

注 2. 能通过设定 TRWERB 寄存器 (定时器的读写允许寄存器 B), 允许或者禁止存取。

表 21.70 互补 PWM 模式中的寄存器设定 (2/2)

通道	计数器 / 寄存器	说明	由 CPU 进行读写
定时器的死区时间寄存器 A (TDDRA)		设定 MTU4.TCNT 和 MTU3.TCNT 的偏移值 (死区时间)	能通过设定 TRWERA 寄存器进行屏蔽 (注 1)
定时器的死区时间寄存器 B (TDDR B)		设定 MTU7.TCNT 和 MTU6.TCNT 的偏移值 (死区时间)	能通过设定 TRWERB 寄存器进行屏蔽 (注 2)
定时器的周期数据寄存器 A (TCDRA)		设定 MTU4.TCNT 的上限值 (1/2 个载波周期)	能通过设定 TRWERA 寄存器进行屏蔽 (注 1)
定时器的周期数据寄存器 B (TCDRB)		设定 MTU7.TCNT 的上限值 (1/2 个载波周期)	能通过设定 TRWERB 寄存器进行屏蔽 (注 2)
定时器的周期缓冲寄存器 A (TCBRA)		TCDRA 的缓冲寄存器	随时能读写
定时器的周期缓冲寄存器 B (TCBRB)		TCDRB 的缓冲寄存器	随时能读写
副计数器 A (TCNTSA)		用于生成死区时间的副计数器 A	只能读
副计数器 B (TCNTSB)		用于生成死区时间的副计数器 B	只能读
暂存器 1A (TEMP1A)		PWM 输出 1/MTU3.TGRB 的暂存器 A	随时能读写
暂存器 1B (TEMP1B)		PWM 输出 1/MTU3.TGRB 的暂存器 B (使用双缓冲功能时)	随时能读写
暂存器 2A (TEMP2A)		PWM 输出 2/MTU4.TGRA 的暂存器 A	随时能读写
暂存器 2B (TEMP2B)		PWM 输出 2/MTU4.TGRA 的暂存器 B (使用双缓冲功能时)	随时能读写
暂存器 3A (TEMP3A)		PWM 输出 3/MTU4.TGRB 的暂存器 A	随时能读写
暂存器 3B (TEMP3B)		PWM 输出 3/MTU4.TGRB 的暂存器 B (使用双缓冲功能时)	随时能读写
暂存器 4A (TEMP4A)		PWM 输出 4/MTU6.TGRB 的暂存器 A	随时能读写
暂存器 4B (TEMP4B)		PWM 输出 4/MTU6.TGRB 的暂存器 B (使用双缓冲功能时)	随时能读写
暂存器 5A (TEMP5A)		PWM 输出 5/MTU7.TGRA 的暂存器 A	随时能读写
暂存器 5B (TEMP5B)		PWM 输出 5/MTU7.TGRB 的暂存器 B (使用双缓冲功能时)	随时能读写
暂存器 6A (TEMP6A)		PWM 输出 6/MTU3.TGRB 的暂存器 A	随时能读写
暂存器 6B (TEMP6B)		PWM 输出 6/MTU3.TGRB 的暂存器 B (使用双缓冲功能时)	随时能读写

注 1. 能通过设定 TRWERA 寄存器 (定时器的读写允许寄存器 A), 允许或者禁止存取。

注 2. 能通过设定 TRWERB 寄存器 (定时器的读写允许寄存器 B), 允许或者禁止存取。

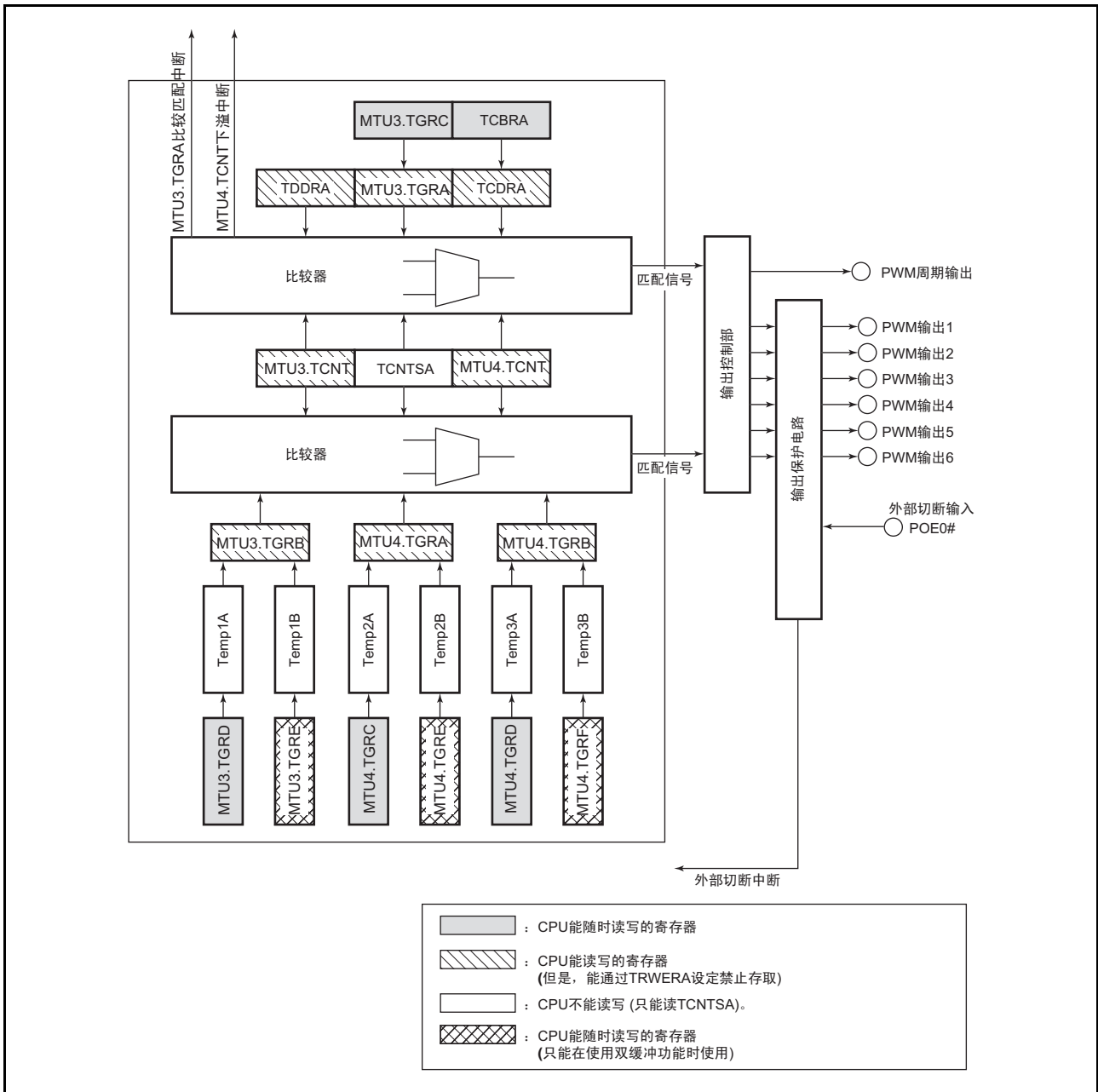


图 21.38 互补 PWM 模式中的 MTU3 和 MTU4 的框图

(1) 互补 PWM 模式的设定步骤例子

互补 PWM 模式的设定步骤例子如图 21.39 所示。



图 21.39 互补 PWM 模式的设定步骤例子

(2) 互补 PWM 模式的运行概要

在互补 PWM 模式中能进行 6 相的 PWM 输出。互补 PWM 模式的计数器运行 (MTU3 和 MTU4) 如图 21.40、互补 PWM 模式的运行例子如图 21.41 所示。

(a) 计数器的运行

在互补 PWM 模式中, MTU3.TCNT、MTU4.TCNT 和 TCNTSA (MTU6.TCNT、MTU7.TCNT 和 TCNTSB) 这 3 个计数器进行递增计数或者递减计数。

在设定为互补 PWM 模式并且 TSTRA (TSTRB) 的 CST 位为“0”时, 自动将 TDDRA (TDDRB) 的设定值作为初始值, 设定给 MTU3.TCNT (MTU6.TCNT)。如果将 CST 位置“1”, MTU3.TCNT (MTU6.TCNT) 就进行递增计数, 直到 MTU3.TGRA (MTU6.TGRA) 的设定值为止。一旦与 MTU3.TGRA (MTU6.TGRA) 的值相同, 就转换为递减计数。以后, 一旦 MTU4.TCNT (MTU7.TCNT) 的值变为“0000h”, MTU3.TCNT (MTU6.TCNT) 就再次转换为递增计数, 重复此运行。

将 MTU4.TCNT (MTU7.TCNT) 的初始值置为“0000h”。如果将 CST 位置“1”, 就与 MTU3.TCNT (MTU6.TCNT) 同步运行并且进行递增计数。一旦 MTU3.TCNT (MTU6.TCNT) 的值与 MTU3.TGRA (MTU6.TGRA) 的值相同, MTU4.TCNT (MTU7.TCNT) 就转换为递减计数。此后, 一旦变为“0000h”, MTU4.TCNT (MTU7.TCNT) 就再次转换为递增计数, 重复此运行。

TCNTSA (TCNTSB) 是只读计数器, 不需要设定初始值。当 MTU3.TCNT 和 MTU4.TCNT (MTU6.TCNT 和 MTU7.TCNT) 进行递增计数时, 一旦 MTU3.TCNT (MTU6.TCNT) 的值与 TCDRA (TCDRB) 的值相同, 就开始递减计数。以后, 一旦 MTU3.TCNT (MTU6.TCNT) 的值与 MTU3.TGRA (MTU6.TGRA) 的值相同, 就再次转换为递增计数。另外, 一旦 MTU4.TCNT (MTU7.TCNT) 的值与 TDDRA (TDDRB) 的值相同, TCNTSA (TCNTSB) 的值就被设定为 MTU3.TGRA (MTU6.TGRA) 的值, 并且停止计数。

当 MTU3.TCNT 和 MTU4.TCNT (MTU6.TCNT 和 MTU7.TCNT) 进行递减计数时, 一旦 MTU4.TCNT (MTU7.TCNT) 的值与 TDDRA (TDDRB) 的值相同, 就开始递增计数。以后, 一旦 MTU4.TCNT (MTU7.TCNT) 的值变为“0000h”, 就再次转换为递减计数。另外, 一旦 MTU3.TCNT (MTU6.TCNT) 的值与 TCDRA (TCDRB) 的值相同, TCNTSA (TCNTSB) 的值就被清为“0000h”, 并且停止计数。

TCNTSA (TCNTSB) 只在计数运行期间与已设定 PWM 占空比的比较寄存器和暂存器进行比较。

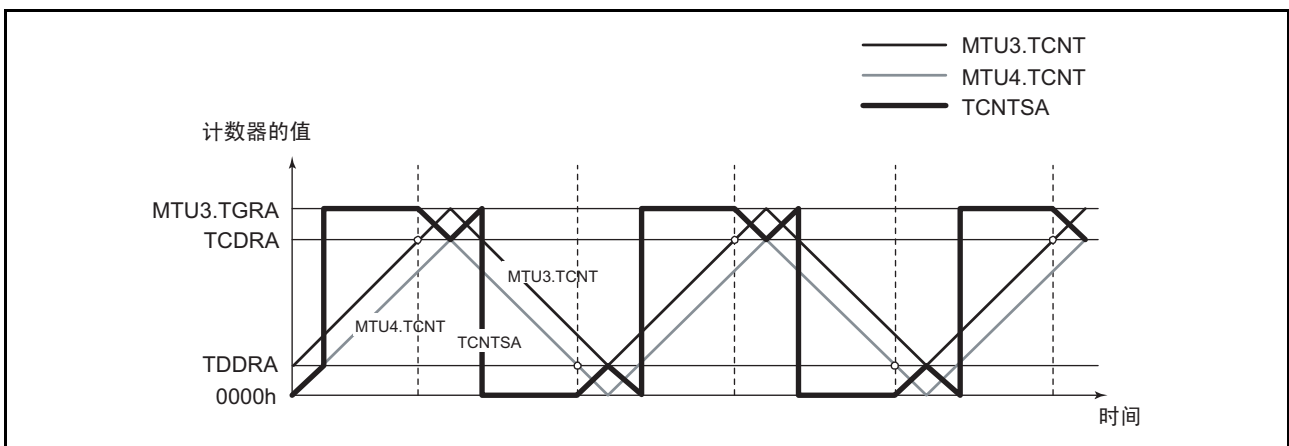


图 21.40 互补 PWM 模式的计数器运行

(b) 寄存器的运行

在互补 PWM 模式中，使用比较寄存器、缓冲寄存器和暂存器等 9 个寄存器。互补 PWM 模式的运行例子 (MTU3 和 MTU4) 如图 21.41 所示。

为了进行 PWM 输出，MTU3.TGRB、MTU4.TGRA、MTU4.TGRB (MTU6.TGRB、MTU7.TGRA、MTU7.TGRB) 寄存器总是和计数器进行比较。如果这些寄存器的值和计数器的值相同，就输出定时器的输出控制寄存器 (TOCR1A 和 TOCR1B) 的 OLSN 位和 OLSP 位的设定值。

这些比较寄存器的缓冲寄存器为 MTU3.TGRD、MTU4.TGRC、MTU4.TGRD (MTU6.TGRD、MTU7.TGRC、MTU7.TGRD)。

在使用双缓冲功能时，也使用缓冲寄存器 B 的 MTU3.TGRE、MTU4.TGRE、MTU4.TGRF (MTU6.TGRE、MTU7.TGRE、MTU7.TGRF) 寄存器。运行的详细内容请参照“21.3.8(2)(s) 互补 PWM 模式的双缓冲功能”。

在缓冲寄存器和比较寄存器之间有暂存器，但是 CPU 不能存取暂存器。

要更改比较寄存器的数据时，必须给对应的缓冲寄存器写要更改的数据。随时能读写缓冲寄存器。

在 Ta 区间总是将写到缓冲寄存器的数据传送到暂存器，而在 Tb 区间不传送到暂存器。在 Tb 区间结束后，将在此区间写到缓冲寄存器的数据传送到暂存器。

如果 Tb 区间结束的 TCNTSA (TCNTSB) 在进行递增计数时，与 MTU3.TGRA (MTU6.TGRA) 的值相同，或者在进行递减计数时为“0000h”，就将传送到暂存器的值传送到比较寄存器。能通过定时器的模式寄存器 1 (TMDR1) 的 MD[3:0] 位选择从暂存器到比较寄存器的传送时序。选择在波谷更改模式的例子如图 21.41 所示。

在不向暂存器传送数据的 Tb (图 21.41 为 Tb1) 区间中，暂存器具有和比较寄存器相同的功能，和计数器进行比较。因此，在此区间中，1 相输出有 2 个比较匹配寄存器，比较寄存器保存变更前的数据，暂存器保存要变更的新数据。而且，在此区间中，MTU3.TCNT、MTU4.TCNT 和 TCNTSA (MTU6.TCNT、MTU7.TCNT 和 TCNTSB) 这 3 个计数器与比较寄存器、暂存器进行比较，并且控制 PWM 输出。

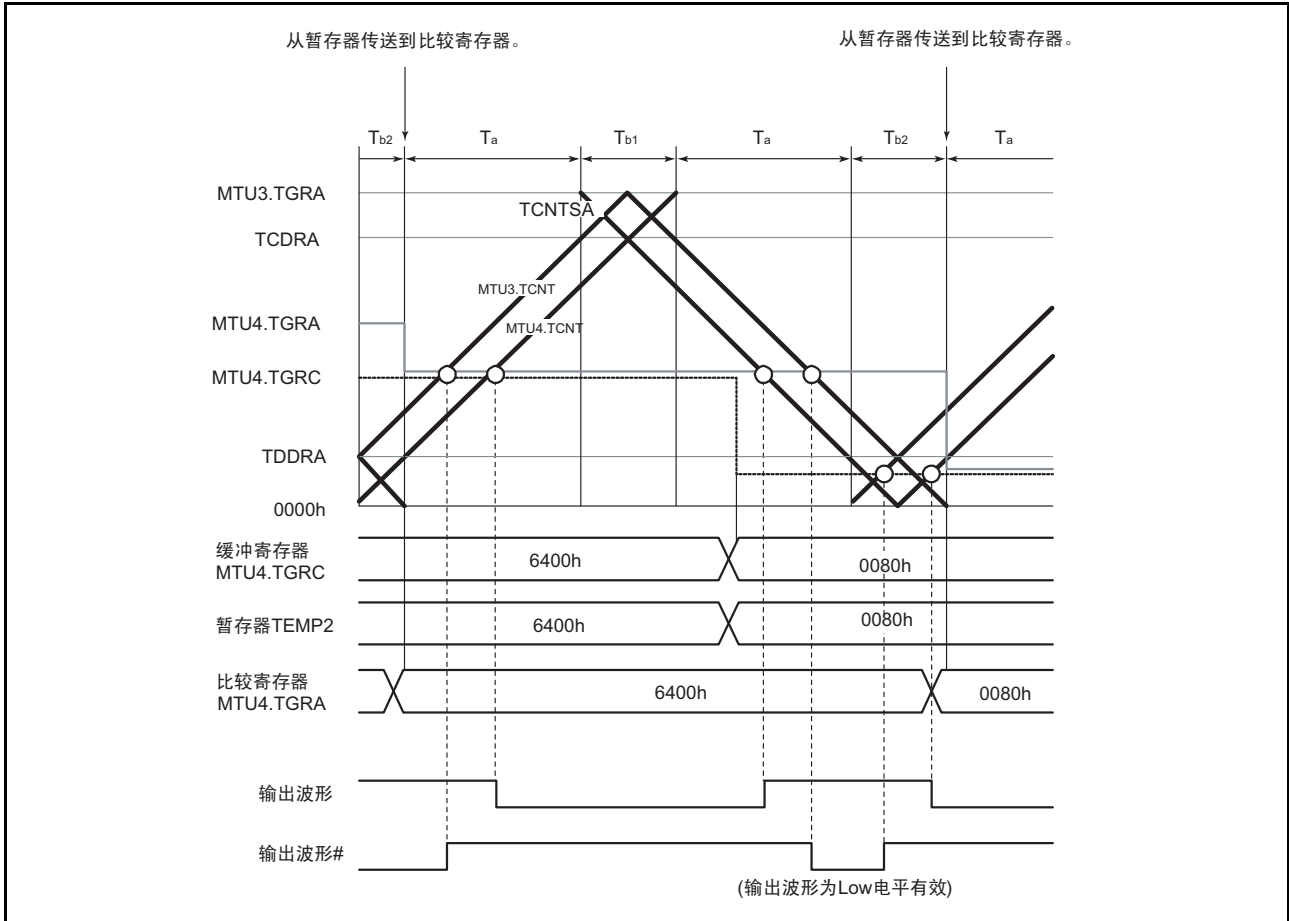


图 21.41 互补 PWM 模式的运行例子 (MTU3 和 MTU4)

(c) 初始设定

在互补 PWM 模式中，有 6 个需要初始设定的寄存器，还有 1 个用于设定有无生成死区时间的寄存器（只能在不生成死区时间的情况下进行设定）。

在通过定时器的模式寄存器 1 (TMDR1) 的 MD[3:0] 位设定为互补 PWM 模式前，必须设定以下寄存器的初始值。

MTU3.TGRC (MTU6.TGRC) 用作 MTU3.TGRA (MTU6.TGRA) 的缓冲寄存器，设定 1/2 个 PWM 载波周期 + 死区时间 Td。定时器的周期缓冲寄存器 (TCBRA 和 TCBRB) 用作定时器的周期数据寄存器 (TCDRA 和 TCDRB) 的缓冲寄存器，设定 1/2 个 PWM 载波周期，并且给定时器的死区时间数据寄存器 (TDDRA 和 TDDRB) 设定死区时间 Td。

如果不生成死区时间，就将定时器的死区时间允许寄存器 (TDERA 和 TDERB) 的 TDER 位置“0”，给 MTU3.TGRC、MTU3.TGRA (MTU6.TGRC、MTU6.TGRA) 设定 1/2 个 PWM 载波周期 +1，并且将 TDDRA (TDDRB) 置“1”。

给缓冲寄存器 A 的 3 个寄存器 (MTU3.TGRD、MTU4.TGRC、MTU4.TGRD (MTU6.TGRD、MTU7.TGRC、MTU7.TGRD)) 分别设定 PWM 占空比的初始值。

只能在使用双缓冲功能时给缓冲寄存器 B 的 3 个寄存器 (MTU3.TGRE、MTU4.TGRE、MTU4.TGRF (MTU6.TGRE、MTU7.TGRE、MTU7.TGRF)) 分别设定 PWM 占空比的初始值 -1。

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

必须在设定互补 PWM 模式前将 MTU4.TCNT (MTU7.TCNT) 设定为“0000h”。

表 21.71 需要初始设定的寄存器和计数器

寄存器 / 计数器	设定值
MTU3.TGRC MTU6.TGRC	1/2 个 PWM 载波周期 + 死区时间 Td (如果通过 TDERA/B 设定为不生成死区时间，就为 1/2 个 PWM 载波周期 +1)
TDDRA、TDDRB	死区时间 Td (如果通过 TDERA/B 设定为不生成死区时间，就为“1”)
TCBRA、TCBRB	1/2 个 PWM 载波周期
MTU3.TGRD、MTU4.TGRC、 MTU4.TGRD MTU6.TGRD、MTU7.TGRC、 MTU7.TGRD	各相 PWM 占空比的初始值
MTU3.TGRE、MTU4.TGRE、 MTU4.TGRE MTU6.TGRE、MTU7.TGRE MTU7.TGRF	各相 PWM 占空比的初始值 - 1 (只在使用双缓冲功能时)
MTU4.TCNT、MTU7.TCNT	0000h

注. MTU3.TGRC (MTU6.TGRC) 的设定值必须为给 TCBRA (TCBRB) 设定的 1/2 个 PWM 载波周期的值和给 TDDRA (TDDRB) 设定的死区时间 Td 值的和。但是，如果通过 TDERA (TDERB) 寄存器设定为不生成死区时间，就为 1/2 个 PWM 载波周期 +1。

(d) PWM 输出电平的设定

在互补 PWM 模式中，通过定时器的输出控制寄存器 1 (TOCR1A、TOCR1B) 的 OLSN 位和 OLSP 位或者定时器的输出控制寄存器 2 (TOCR2A、TOCR2B) 的 OLS1P ~ OLS3P 位和 OLS1N ~ OLS3N 位，设定 PWM 脉冲的输出电平。

能给 6 相输出中的 3 相正相和 3 相反相分别设定输出电平。

必须在解除互补 PWM 模式的状态下设定或者更改输出电平。

(e) 死区时间的设定

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

将非重叠时间设定到定时器的死区时间数据寄存器 (TDDRA、TDDRB)。TDDRA (TDDRB) 的设定值为 MTU3.TCNT (MTU6.TCNT) 的计数器启动值，生成 MTU3.TCNT (MTU6.TCNT) 和 MTU4.TCNT (MTU7.TCNT) 的非重叠关系。必须在解除互补 PWM 模式的状态下更改 TDDRA (TDDRB) 的内容。

(f) 不生成死区时间的设定

通过将定时器的死区时间允许寄存器 (TDERA、TDERB) 的 TDER 位置“0”，设定为不生成死区时间。只在 TDER 位为“1”的状态下读 TDER 位后给 TDER 位写“0”时，TDERA (TDERB) 能置为“0”。

给 MTU3.TGRA、MTU3.TGRC (MTU6.TGRA、MTU6.TGRC) 设定 1/2 个 PWM 载波周期 +1，并且将定时器的死区时间数据寄存器 (TDDRA、TDDRB) 置“1”。

如果设定为不生成死区时间，就能输出无死区时间的 PWM 波形。不生成死区时间的运行例子 (MTU3 和 MTU4) 如图 21.42 所示。

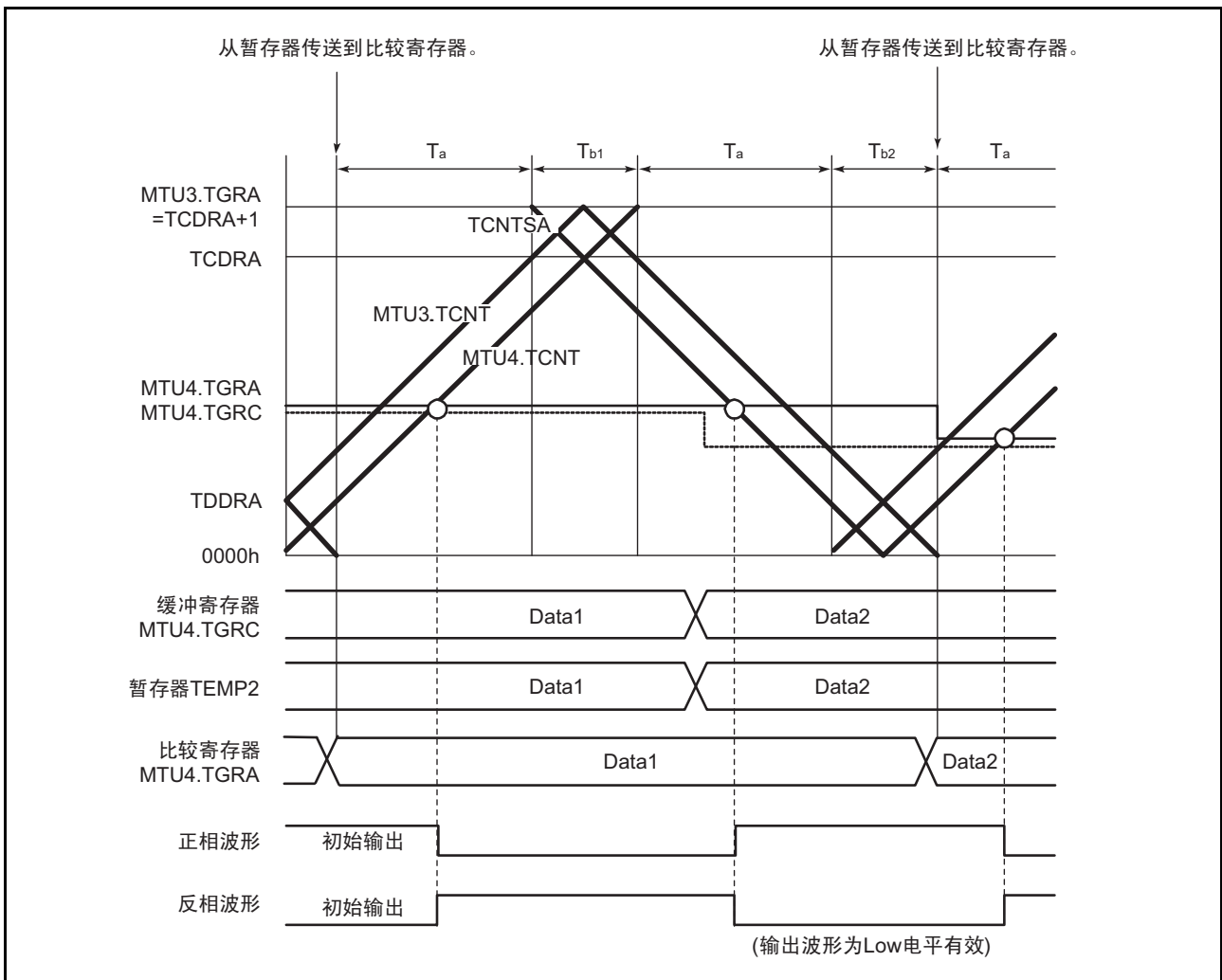


图 21.42 不生成死区时间的运行例子 (MTU3 和 MTU4)

(g) PWM 周期的设定

在互补 PWM 模式中，将 PWM 脉冲周期设定到用于设定 MTU3.TCNT (MTU6.TCNT) 的上限值的 MTU3.TGRA (MTU6.TGRA) 寄存器和用于设定 MTU4.TCNT (MTU7.TCNT) 的上限值的 TCDRA (TCDRB) 寄存器。这 2 个寄存器的关系必须如下设定。

生成死区时间：

MTU3.TGRA (MTU6.TGRA) 的设定值 = TCDRA (TCDRB) 的设定值 + TDDRA (TDDRB) 的设定值

不生成死区时间：MTU3.TGRA (MTU6.TGRA) 的设定值 = TCDRA (TCDRB) 的设定值 + 1

必须通过给缓冲寄存器的 MTU3.TGRC 和 TCBRA (MTU6.TGRC 和 TCBRB) 设定值，设定 MTU3.TGRA 和 TCDRA (MTU6.TGRA 和 TCDRB)。通过定时器的模式寄存器 1 (TMDR1) 的 MD[3:0] 位选择的传送时序，将 MTU3.TGRC 和 TCBRA (MTU6.TGRC 和 TCBRB) 的设定值同时传送到 MTU.TGRA 和 TCDRA (MTU6.TGRA 和 TCDRB)。

如果在波峰更新数据，就从下一个周期反映更改的 PWM 周期；如果在波谷更新数据，就从该周期反映被更改的 PWM 周期。在波峰更改 PWM 周期时的运行例子如图 21.43 所示。

有关各缓冲寄存器数据的更新方法，请参照下面“(h) 寄存器数据的更新”。

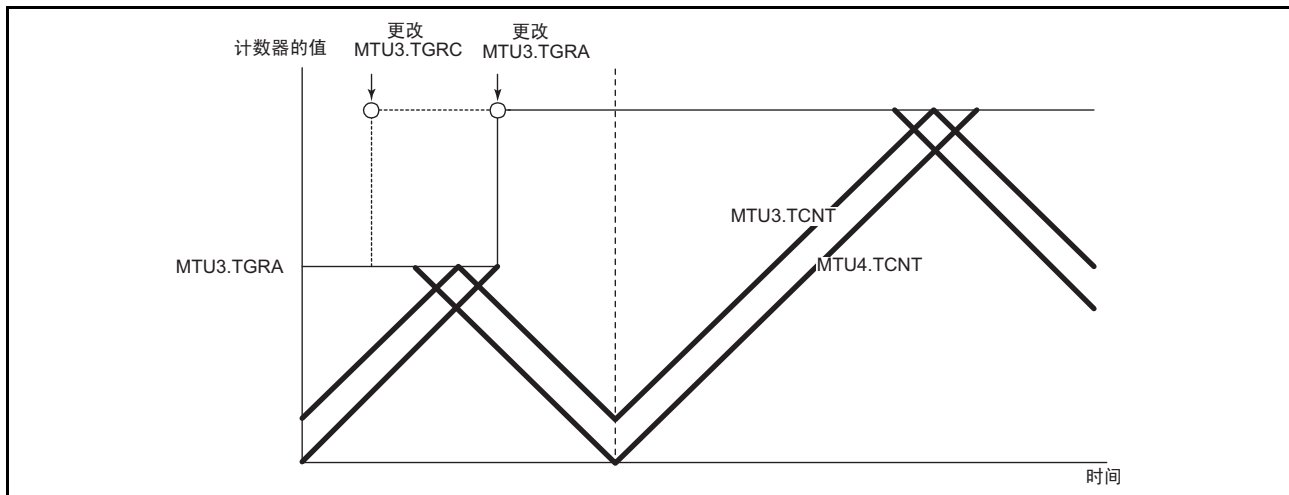


图 21.43 PWM 周期的更改例子 (MTU3 和 MTU4)

(h) 寄存器数据的更新

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

在这些寄存器和缓冲寄存器之间有各自的暂存器。如果在副计数器 TCNTSA (TCNTSB) 不进行计数的期间更新缓冲寄存器的数据，也会改写暂存器的值。在 TCNTSA (TCNTSB) 计数过程中，不进行缓冲寄存器到暂存器的传送，而在 TCNTSA (TCNTSB) 停止计数后传送缓冲寄存器的写入值。

通过定时器的模式寄存器 1 (TMDR1) 的 MD[3:0] 位设定的数据更新时序，将暂存器的值传送到比较寄存器。互补 PWM 模式的数据更新例子 (MTU3 和 MTU4) 如图 21.44 所示，此图是在计数器的波峰和波谷更新数据的模式例子。

在改写缓冲寄存器的数据时，最后必须写 MTU4.TGRD (MTU7.TGRD)。在写 MTU4.TGRD (MTU7.TGRD) 后，5 个寄存器的数据同时从缓冲寄存器传送到暂存器。

即使不全部更新 5 个寄存器或者不更新 MTU4.TGRD (MTU7.TGRD) 的数据，也必须在写要更新的寄存器的数据后写 MTU4.TGRD (MTU7.TGRD)。此时，写到 MTU4.TGRD (MTU7.TGRD) 的数据必须和写之前的数据相同。

有关使用双缓冲功能时的数据更新，请参照“21.3.8(2)(s) 互补 PWM 模式的双缓冲功能”。

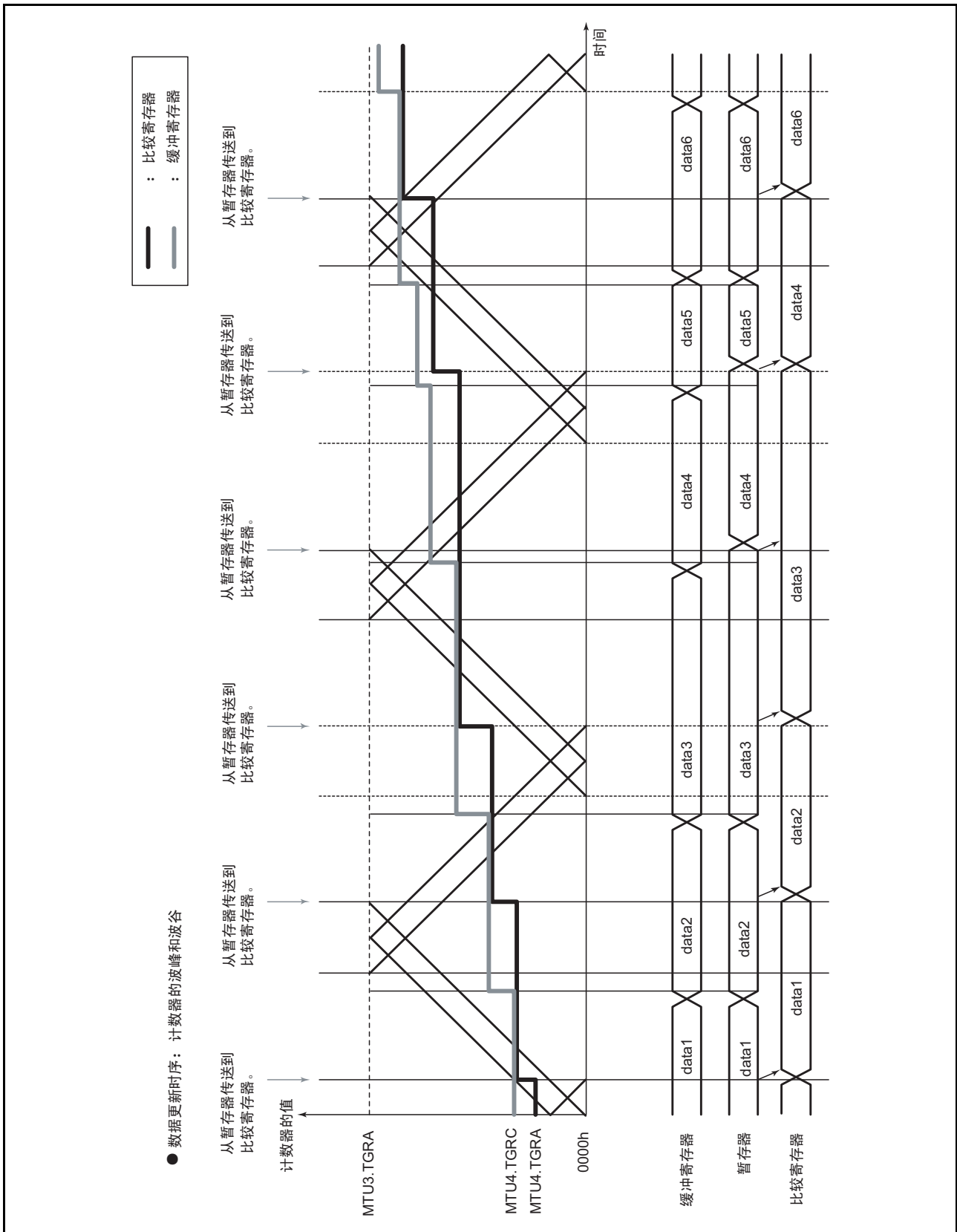


图 21.44 互补 PWM 模式的数据更新例子 (MTU3 和 MTU4)

(i) 互补 PWM 模式的初始输出

在互补 PWM 模式中，通过设定定时器的输出控制寄存器 1 (TOCR1A、TOCR1B) 的 OLSN 位和 OLSP 位或者定时器的输出控制寄存器 2 (TOCR2A、TOCR2B) 的 OLS1N ~ OLS3N 位和 OLS1P ~ OLS3P 位，决定初始输出。

此初始输出为 PWM 脉冲的无效电平，从通过定时器的模式寄存器 1 (TMDR1) 设定互补 PWM 模式开始到 MTU4.TCNT (MTU7.TCNT) 大于死区时间寄存器 (TDDRA、TDDRb) 设定的值前，输出此初始输出。互补 PWM 模式的初始输出例子如图 21.45 所示。

PWM 占空比的初始值小于 TDDRA (TDDRb) 的值时的波形例子 (MTU3 和 MTU4) 如图 21.46 所示。

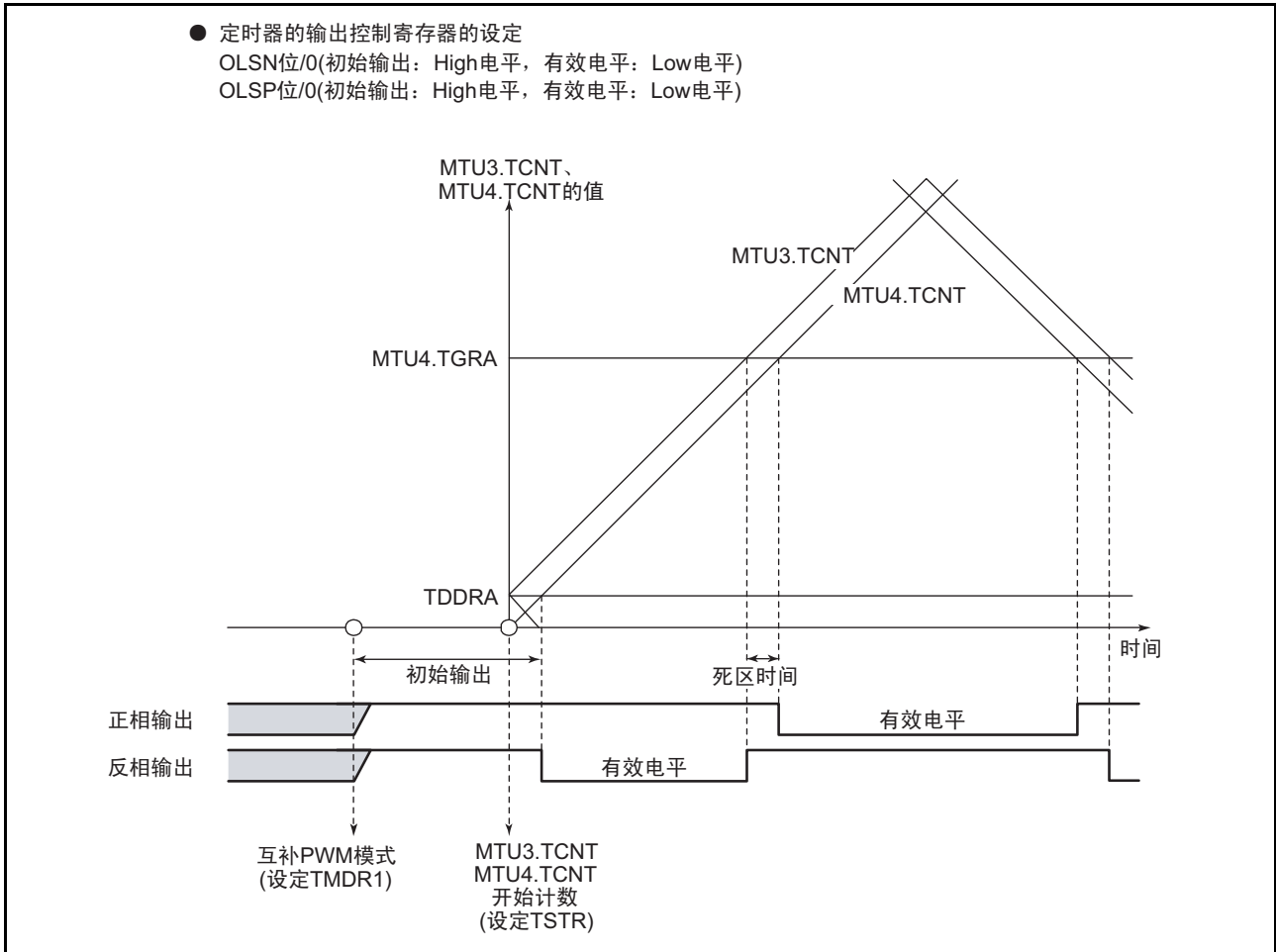


图 21.45 互补 PWM 模式的初始输出例子 (MTU3 和 MTU4) (1)

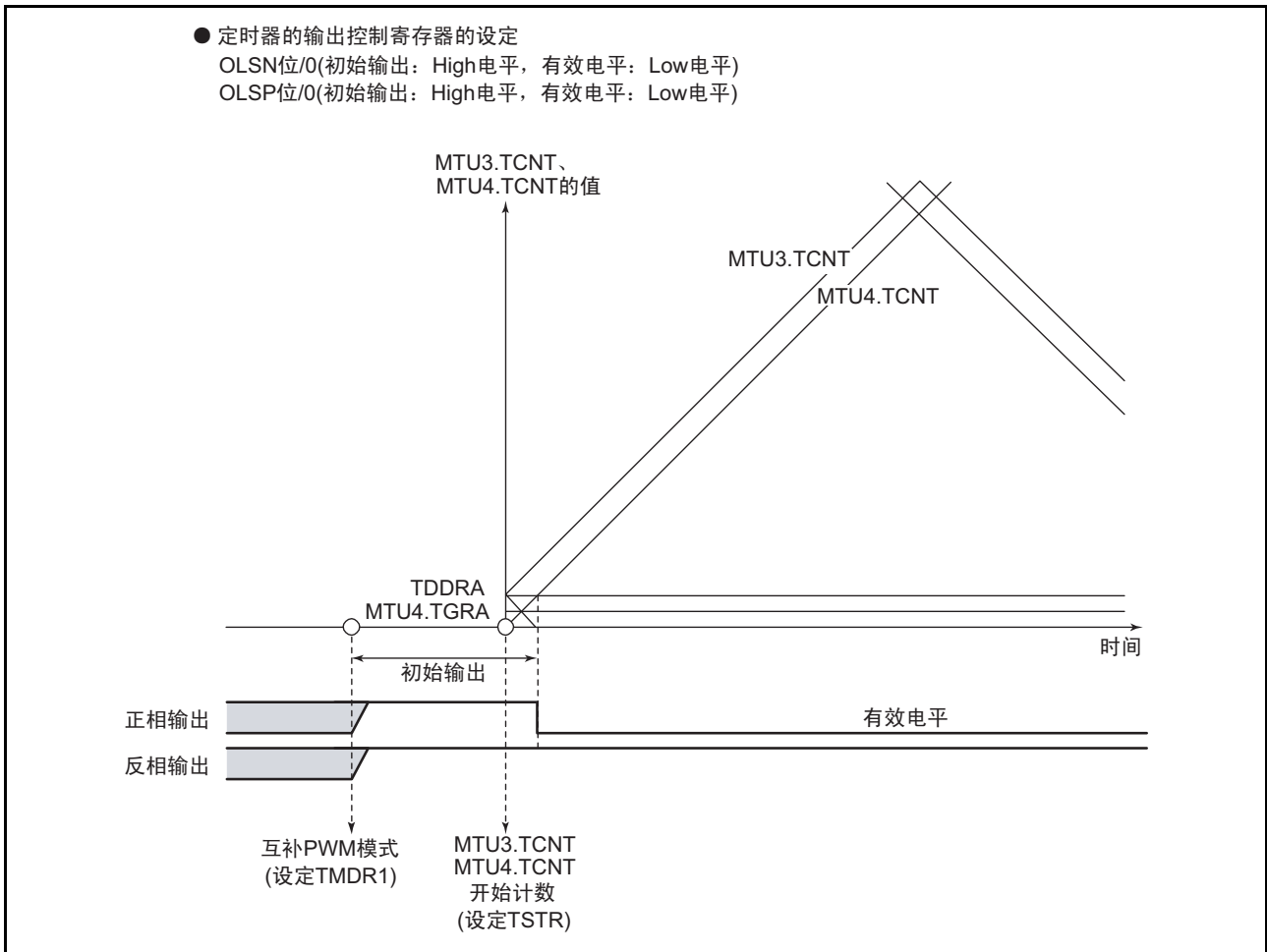


图 21.46 互补 PWM 模式的初始输出例子 (MTU3 和 MTU4) (2)

(j) 互补 PWM 模式的 PWM 输出生成方法

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

在发生计数器和比较寄存器的比较匹配时，通过输出定时器的输出控制寄存器选择的输出电平，生成 PWM 波形。在 TCNTSA (TCNTSB) 进行计数的期间，因为产生 0 ~ 100% 的连续 PWM 脉冲，所以同时比较比较寄存器的值和暂存器的值。此时，ON/OFF 比较匹配的发生时序会有前有后，为了确保死区时间并且使正相 / 反相的 ON 时间不重叠，必须优先使各相 OFF 的比较匹配。互补 PWM 模式的波形生成例子 (MTU3 和 MTU4) 如图 21.47 ~ 图 21.49 所示。

通过和 MTU3.TCNT 计数器的比较匹配，生成正相 / 反相的 OFF 时序；通过和 MTU4.TCNT 计数器 (比 MTU3.TCNT 计数器迟死区时间) 的比较匹配，生成 ON 时序。在 T1 期间，最优先使反相 OFF 的 a 的比较匹配，忽视比 a 先发生的比较匹配。在 T2 期间，最优先使正相 OFF 的 c 的比较匹配，忽视比 c 先发生的比较匹配。

如图 21.47 所示，通常按照 a→b→c→d (或者 c→d→a'→b') 的顺序发生比较匹配。

当不按 a→b→c→d 的顺序发生比较匹配时，因为反相的 OFF 时间短于 2 倍的死区时间，所以表示正相不为 ON；当不按 c→d→a'→b' 的顺序发生比较匹配时，因为正相的 OFF 时间短于 2 倍的死区时间，所以表示反相不为 ON。

如图 21.48 所示，如果在 a 的比较匹配之后先发生 c 的比较匹配，就忽视 b 的比较匹配，而通过 d 的比较匹配使反相 ON。这是因为比 b 的比较匹配 (正相 ON 时序) 先发生正相 OFF 的 c 的比较匹配而优先使正相 OFF (因为正相从 OFF 变为 OFF，所以波形不变)。

同样地，在图 21.49 所示的例子中，比 c 的比较匹配先发生和暂存器的新数据比较匹配的 a'，但是在发生使正相 OFF 的 c 前忽视其他比较匹配，因此不使反相 ON。

如此，在互补 PWM 模式中，优先 OFF 时序的比较匹配，即使 ON 时序的比较匹配比 OFF 先发生，也被忽视。

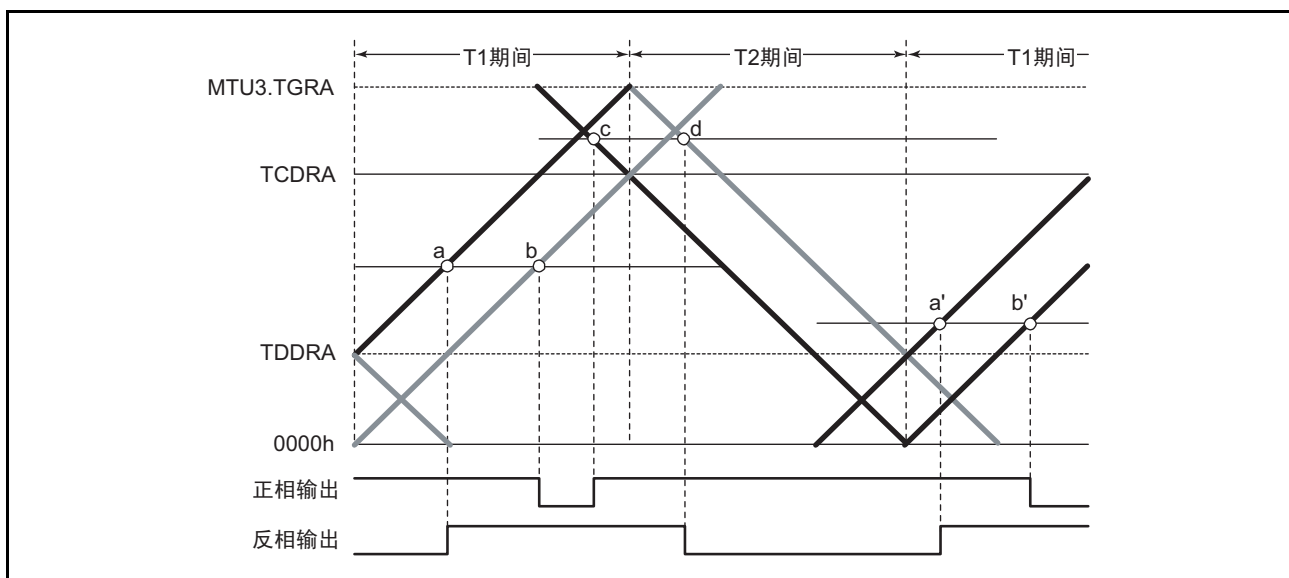


图 21.47 互补 PWM 模式的波形输出例子 (MTU3 和 MTU4) (1)

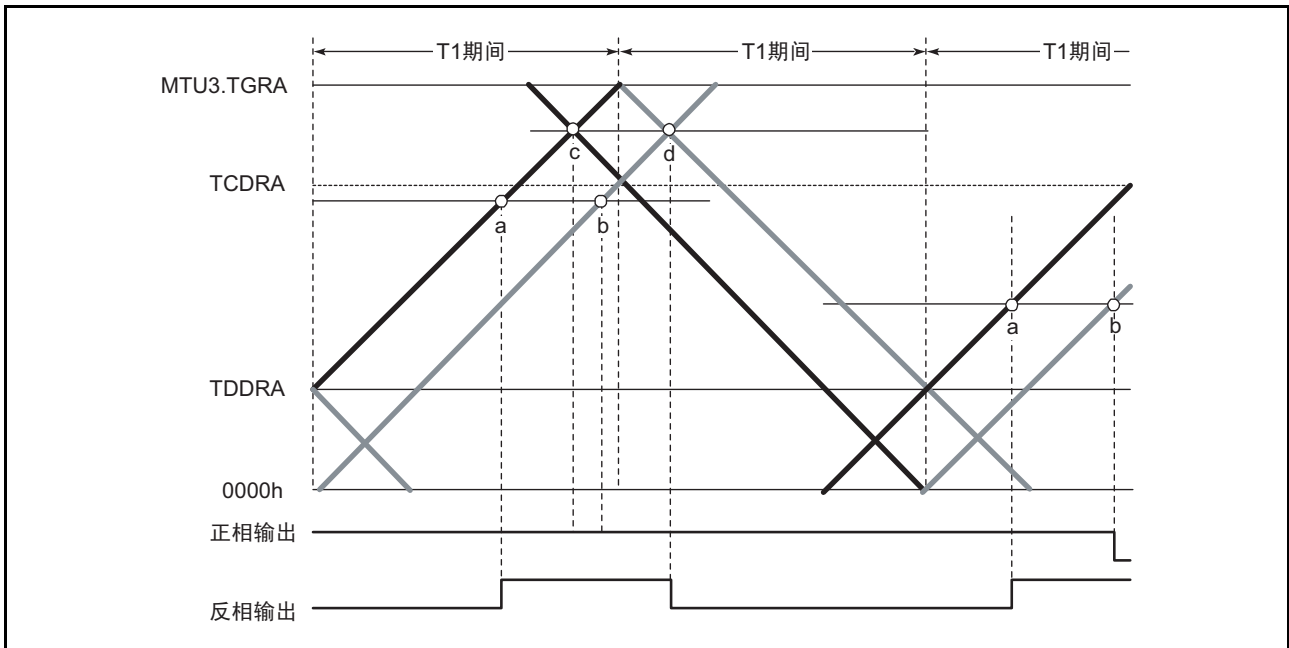


图 21.48 互补 PWM 模式的波形输出例子 (MTU3 和 MTU4) (2)

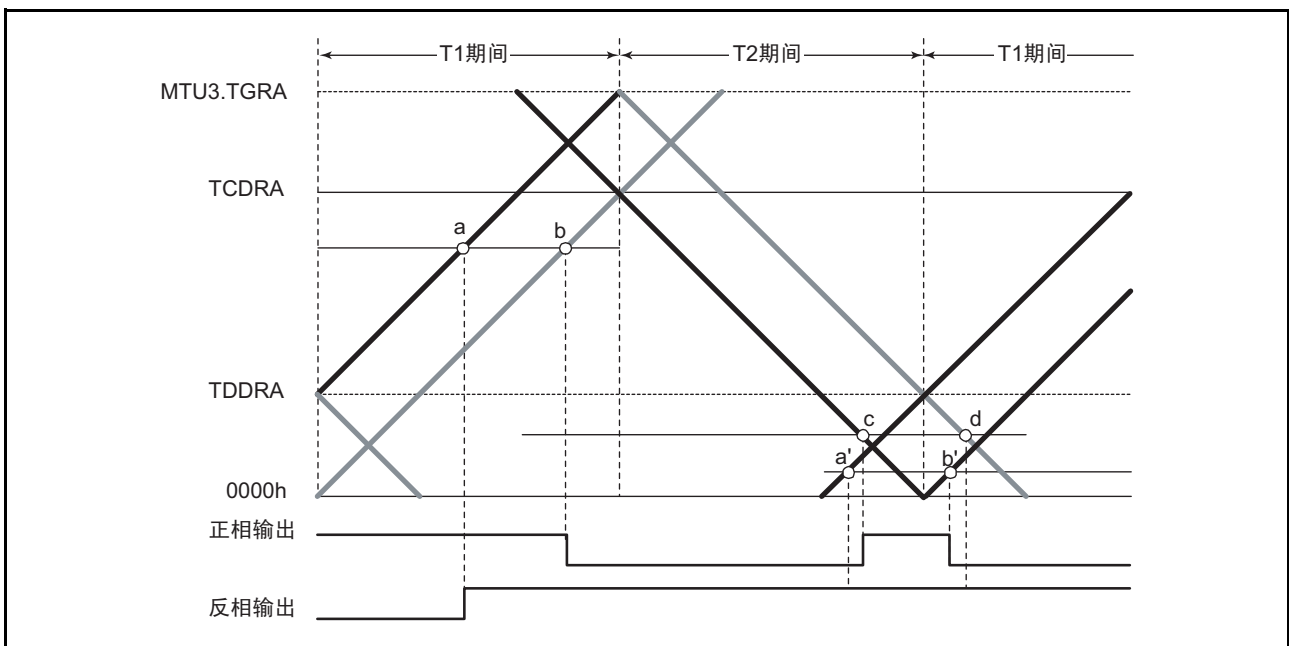


图 21.49 互补 PWM 模式的波形输出例子 (MTU3 和 MTU4) (3)

(k) 互补 PWM 模式的占空比 0%、100% 输出

在互补 PWM 模式中，能任意输出 0%、100% 的占空比，输出例子 (MTU3 和 MTU4) 如图 21.50 ~ 图 21.54 所示。

如果将比较寄存器的值设定为“0000h”，就输出 100% 的占空比。此时的波形是正相为 100%ON 状态的波形。如果将比较寄存器的值设定为和 MTU3.TGRA (MTU6.TGRA) 相同的值，就输出 0% 的占空比。此时的波形是正相为 100%OFF 状态的波形。

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

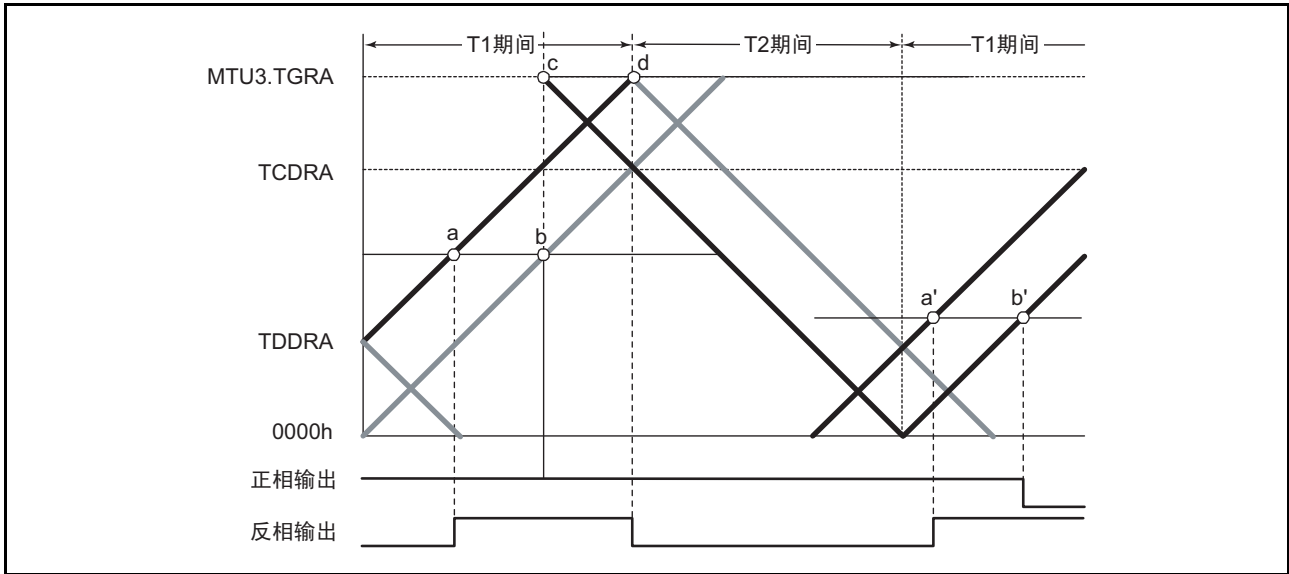


图 21.50 互补 PWM 模式的 0%、100% 波形输出例子 (MTU3 和 MTU4) (1)

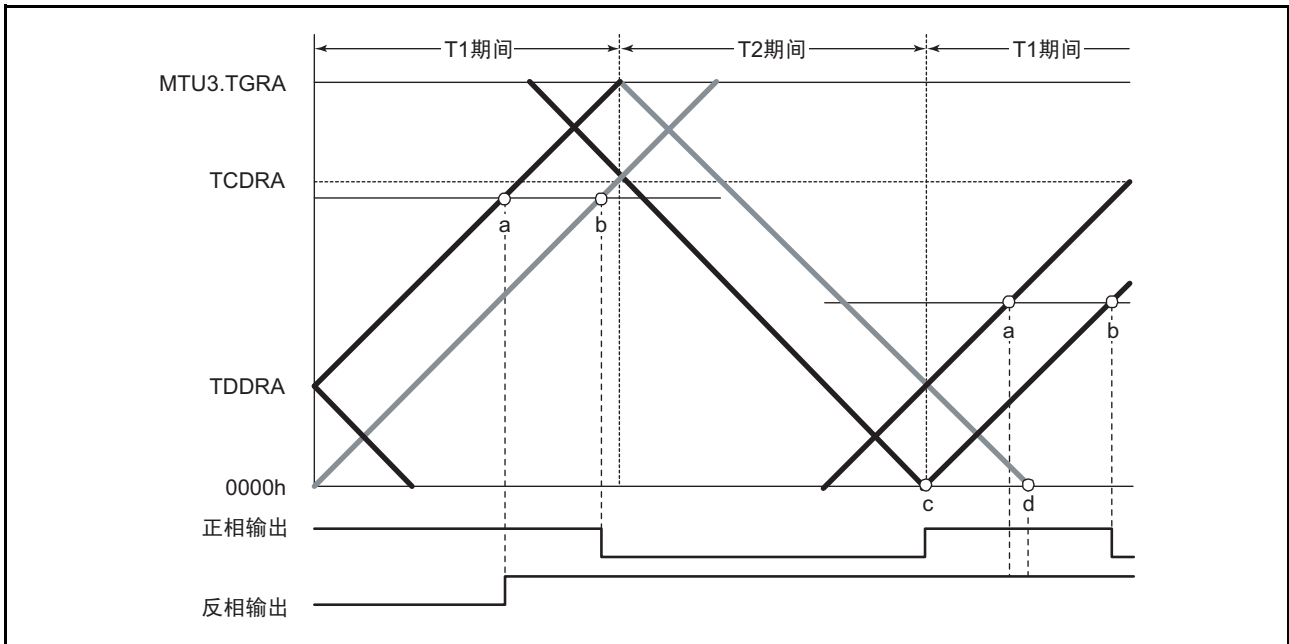


图 21.51 互补 PWM 模式的 0%、100% 波形输出例子 (MTU3 和 MTU4) (2)

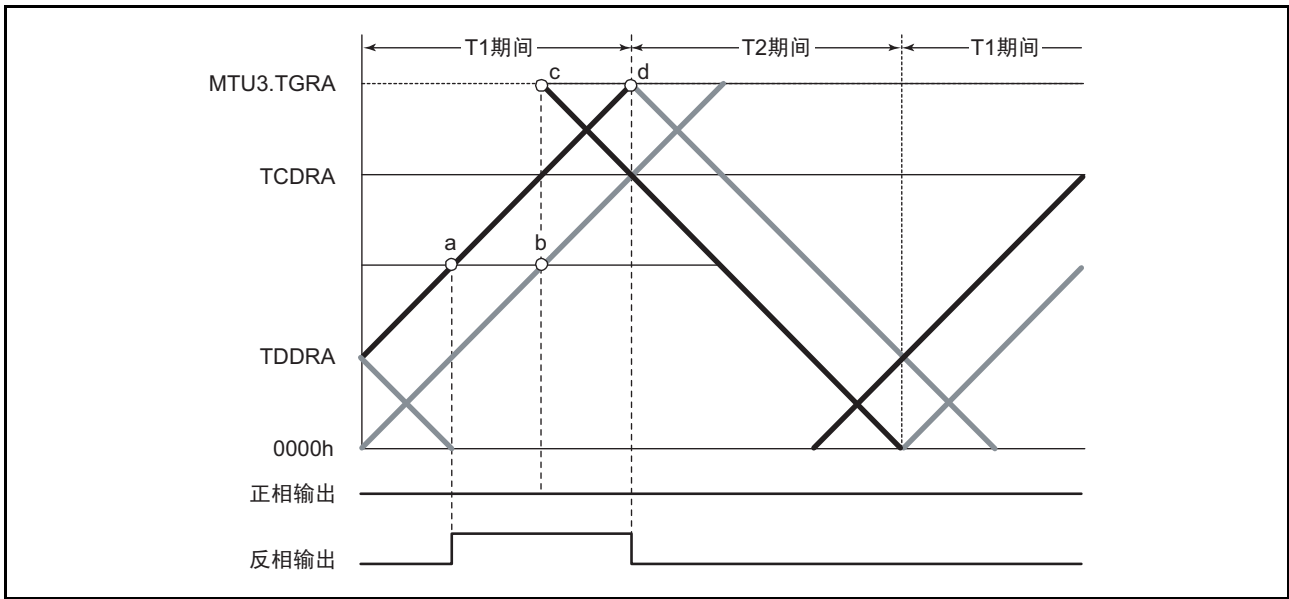


图 21.52 互补 PWM 模式的 0%、100% 波形输出例子 (MTU3 和 MTU4) (3)

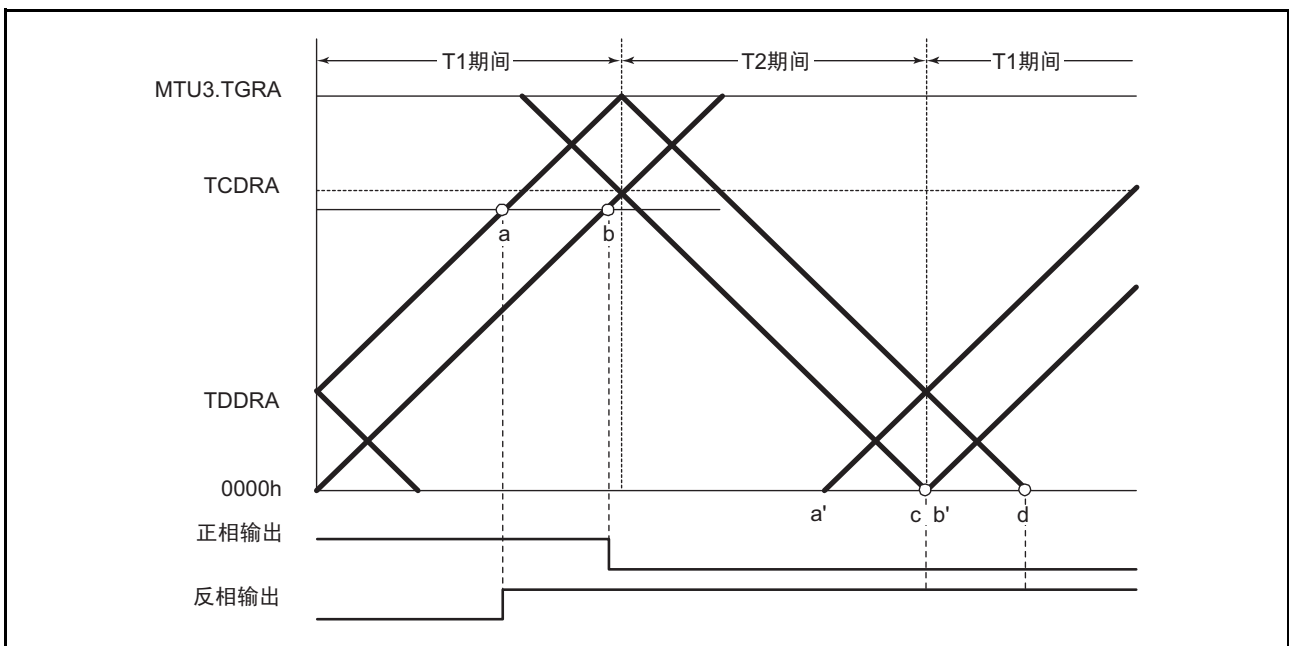


图 21.53 互补 PWM 模式的 0%、100% 波形输出例子 (MTU3 和 MTU4) (4)

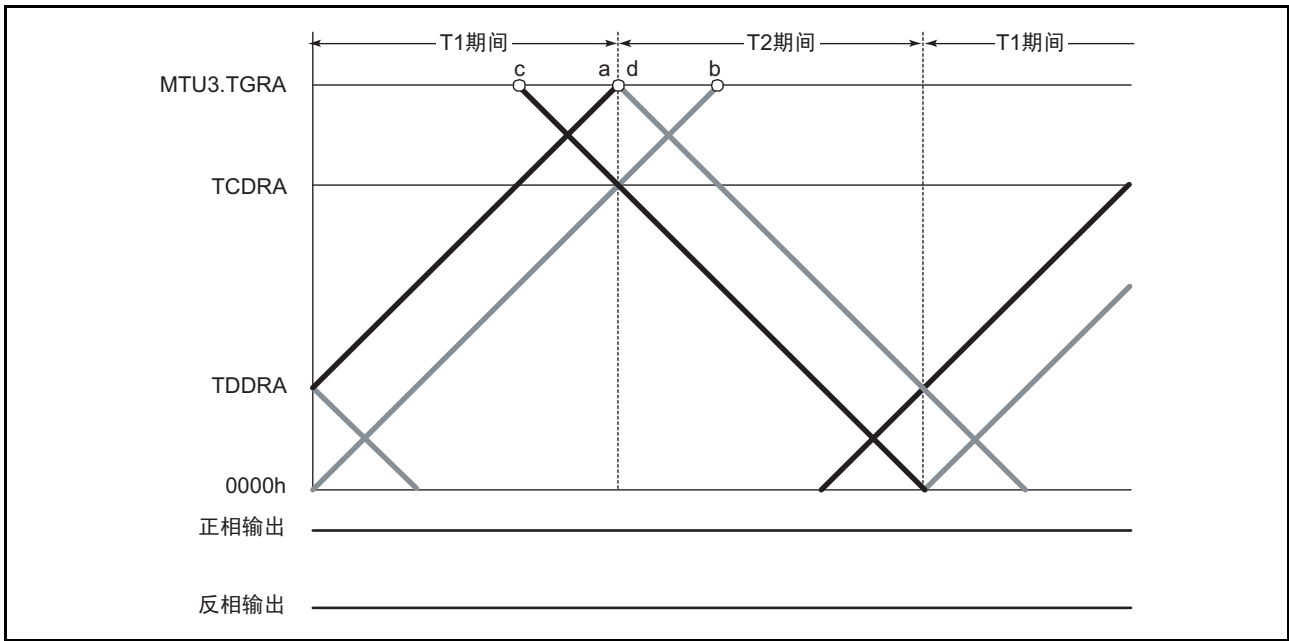


图 21.54 互补 PWM 模式的 0%、100% 波形输出例子 (MTU3 和 MTU4) (5)

(I) 与 PWM 周期同步的交替输出

在互补 PWM 模式中，能通过将定时器的输出控制寄存器 1 (TOCR1A、TOCR1B) 的 PSYE 位置“1”，进行与 PWM 载波周期同步的交替输出。交替输出的波形例子 (MTU3 和 MTU4) 如图 21.55 所示。

通过 MTU3.TCNT 和 MTU3.TGRA (MTU6.TCNT 和 MTU6.TGRA) 的比较匹配以及 MTU4.TCNT (MTU7.TCNT) 和 0000h 的比较匹配进行交替输出。

此交替输出的输出引脚为 MTIOC3A (MTIOC6A) 引脚，初始输出为 High 电平输出。

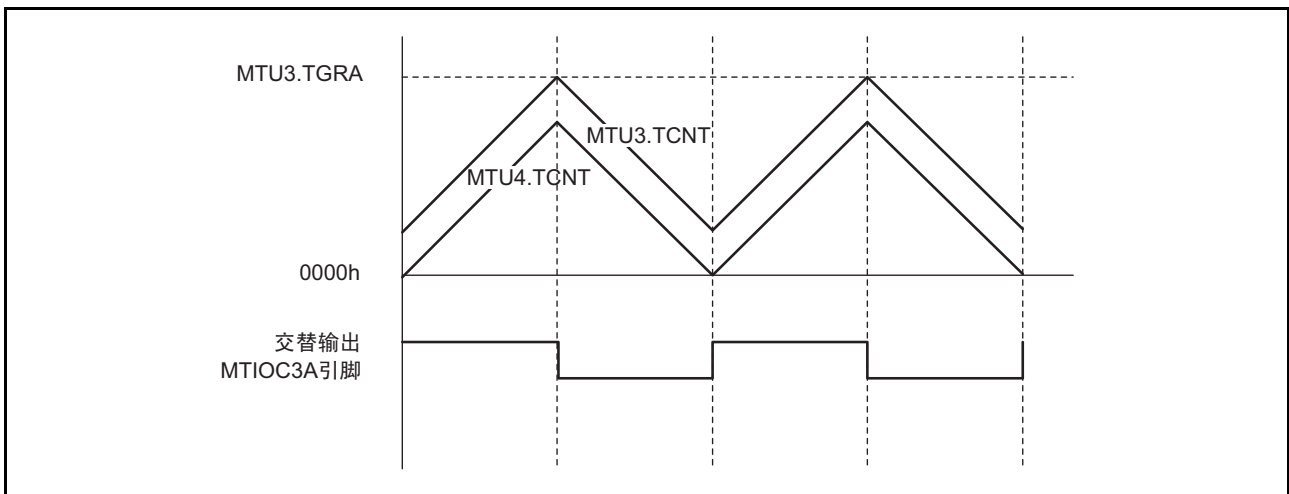


图 21.55 与 PWM 输出同步的交替输出波形例子 (MTU3 和 MTU4)

(m) 通过其他通道进行的计数器清除

在互补 PWM 模式中，当通过定时器的同步寄存器 (TSYRA、TSYRB) 设定为与其他通道的同步模式，并且通过定时器的控制寄存器 (TCR) 的 CCLR[2:0] 位选择同步清除时，能由其他通道进行 MTU3.TCNT、MTU4.TCNT 和 TCNTSA (MTU6.TCNT、MTU7.TCNT 和 TCNTSB) 的清除。

运行例子 (MTU3 和 MTU4) 如图 21.56 所示。

使用此功能，能通过外部信号进行计数器清除和重新开始。

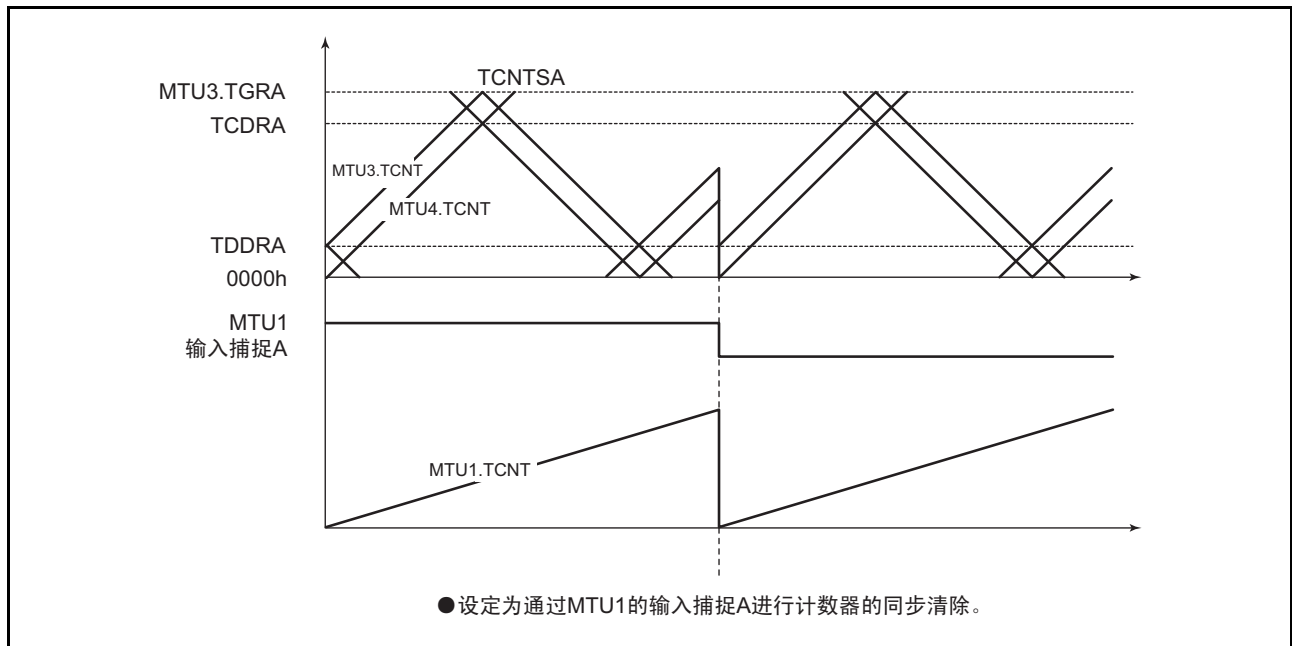


图 21.56 与其他通道同步的计数器清除 (MTU3 和 MTU4)

(n) 互补 PWM 模式中同步计数器清除时的输出波形控制

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

只有在如图 21.57 的⑩、⑪ 的波谷 Tb 区间进行同步清除时，才能通过将 WRE 位置“1”来抑制初始输出。如果在其他时序中发生同步清除，就输出 TOCR1A (TOCR1B) 寄存器的 OLS 位设定的初始值。即使在波谷的 Tb 区间，如果在图 21.57 的①所示的计数器开始计数后的初始输出期间发生同步清除，也不抑制初始输出。

能在 MTU3、MTU4 - MTU6、MTU7 中使用此功能。在 MTU3 和 MTU4 中，MTU3、MTU4 - MTU6、MTU7 的计数器清除源为 MTU0 ~ 2 的同步清除；在 MTU6 和 MTU7 中，MTU3、MTU4 - MTU6、MTU7 的计数器清除源为 MTU0 ~ 2 的标志的置位 (比较匹配 / 输入捕捉)。

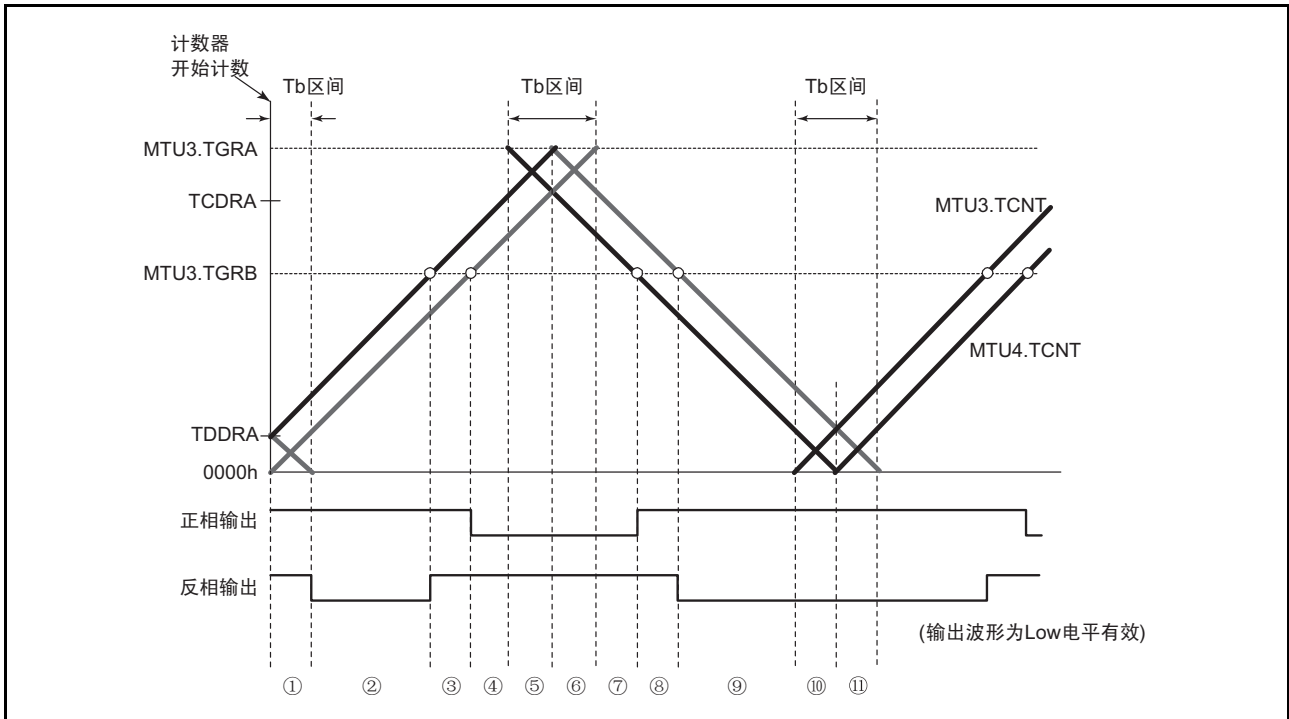


图 21.57 同步计数器清除的时序 (MTU3 和 MTU4)

- 互补 PWM 模式中同步计数器清除时的输出波形控制设定步骤例子
互补 PWM 模式中同步计数器清除时的控制输出波形设定步骤例子 (MTU3 和 MTU4) 如图 21.58 所示。

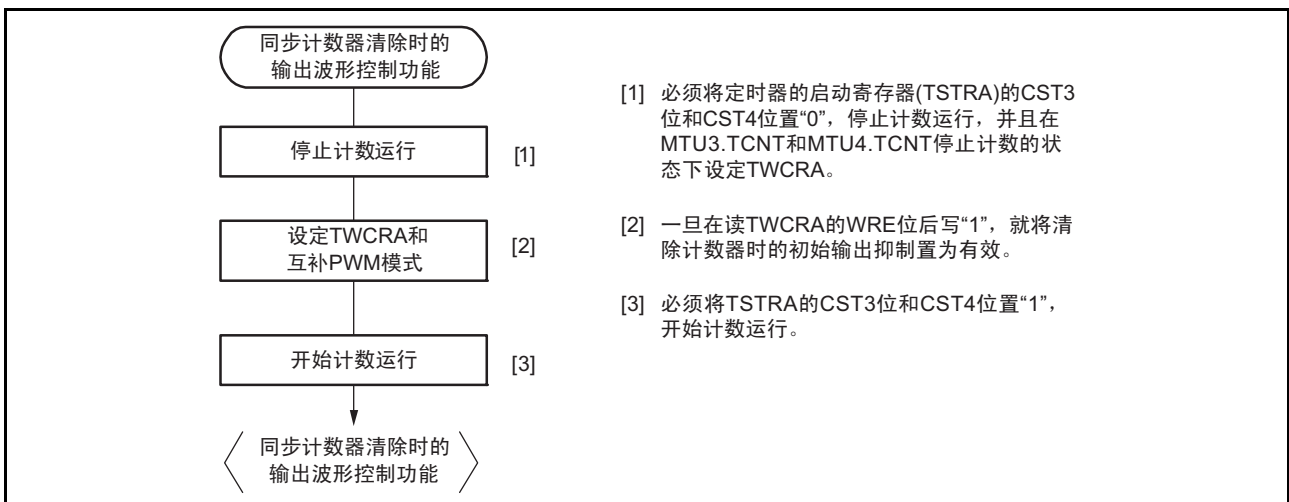


图 21.58 互补 PWM 模式中同步计数器清除时的输出波形控制设定步骤例子 (MTU3 和 MTU4)

• 互补PWM模式中同步计数器清除时的输出波形控制运行例子

在将 TWCRA.WRE 位置“1”的状态下使 MTU3 和 MTU4 进行互补 PWM 运行并且进行同步计数器清除时的运行例子如图 21.59 ~ 图 21.62 所示。在此，图 21.59 ~ 图 21.62 的同步计数器清除时序分别是图 21.57 的 ③、⑥、⑧、⑩ 所示的时序。

此例相当于，在将 TWCRA.SCC 位置“0”、WRE 位置“1”的状态下使 MTU6 和 MTU7 进行互补 PWM 运行并且进行同步计数器清除时的情况。

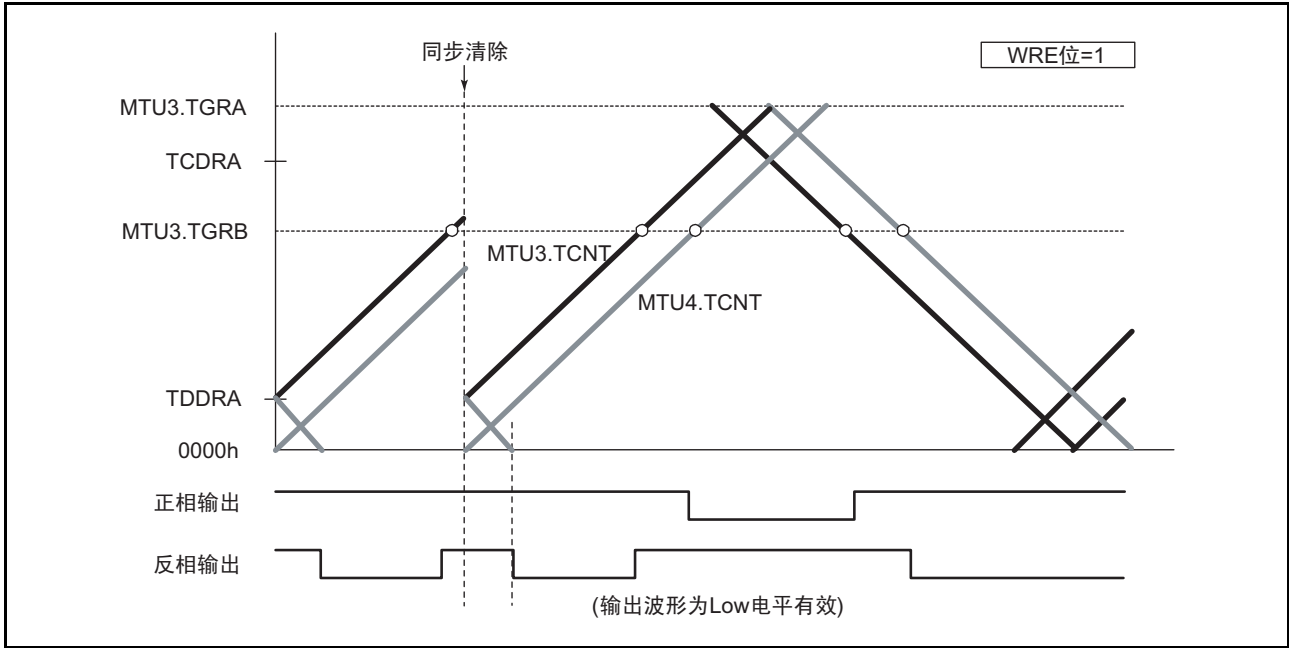


图 21.59 在递增计数过程中的死区时间发生同步清除的情况
(图 21.57 的时序③、TWCRA 寄存器的 WRE 位 =1)

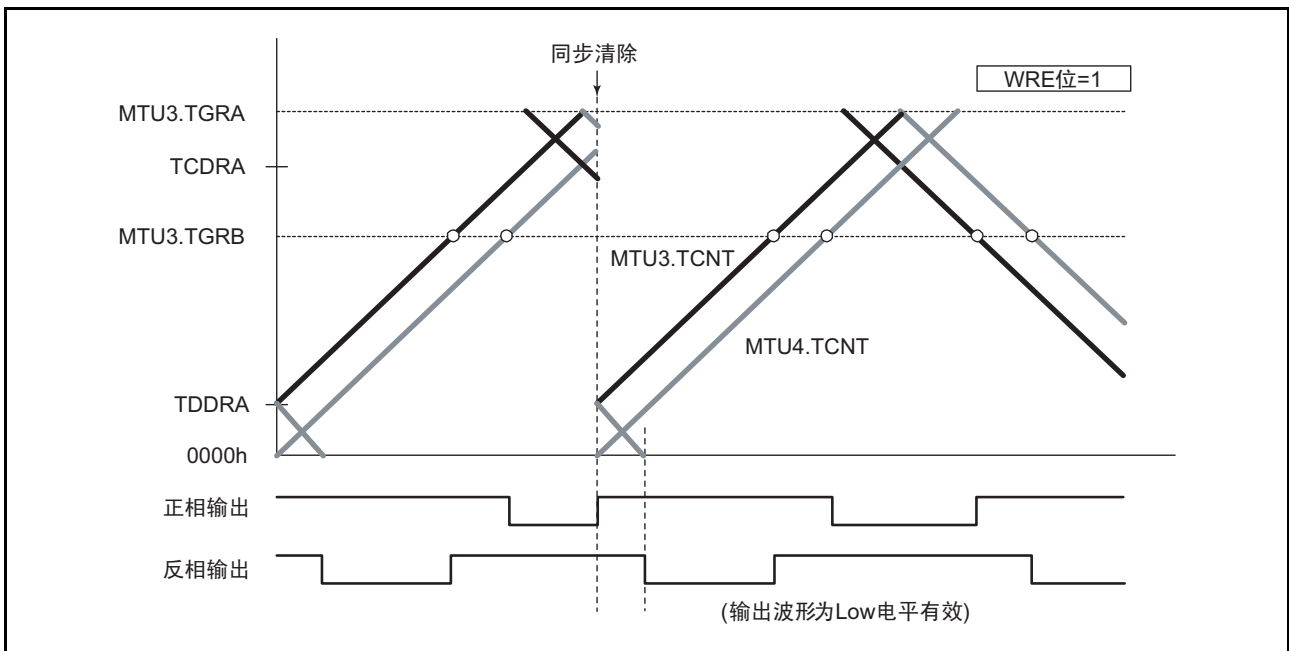


图 21.60 在波峰 Tb 区间发生同步清除的情况
(图 21.57 的时序⑥、TWCRA 寄存器的 WRE 位 =1)

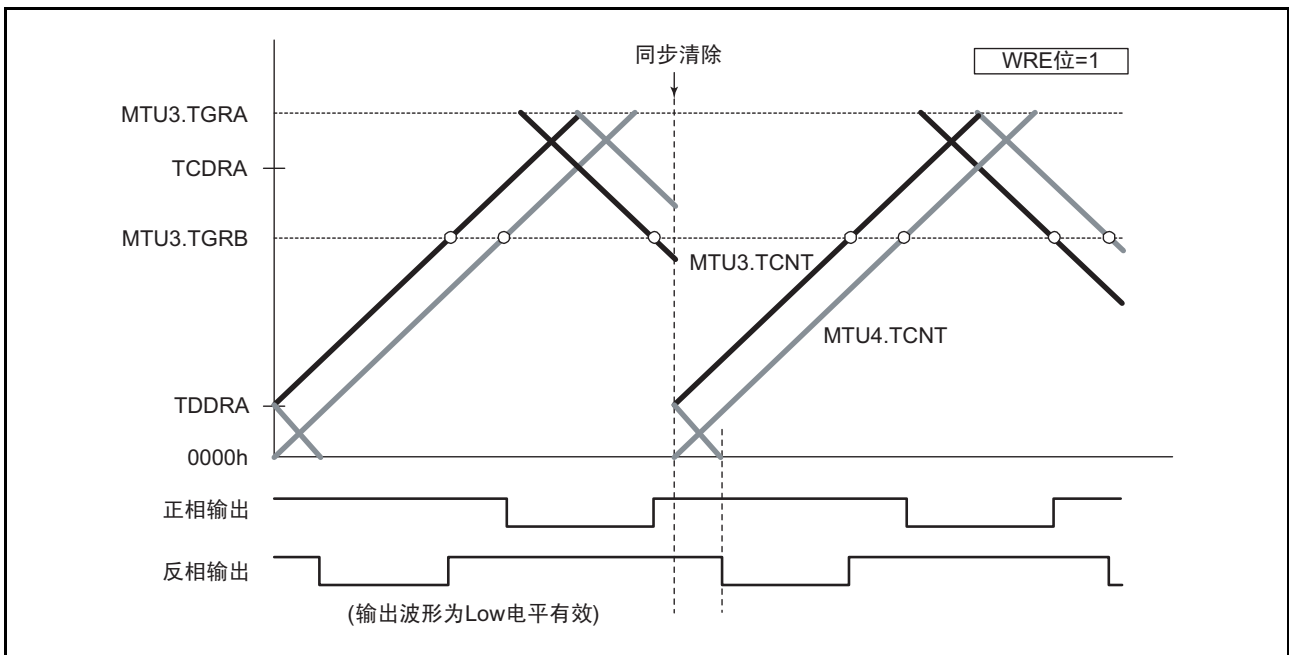


图 21.61 在递减计数过程中的死区时间发生同步清除的情况
(图 21.57 的时序⑧、TWCRA 寄存器的 WRE 位 =1)

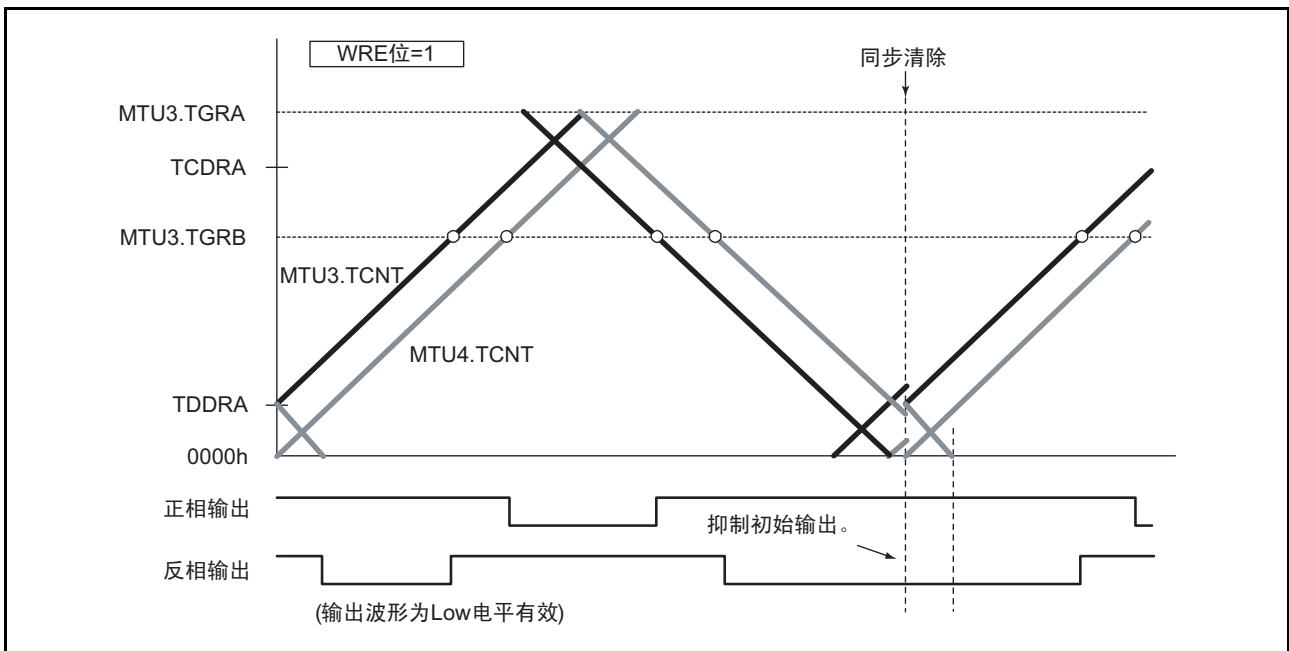


图 21.62 在波谷 Tb 区间发生同步清除的情况
(图 21.57 的时序⑩、TWCRA 寄存器的 WRE 位 =1)

(o) MTU0、1、2 – MTU6、7 的同步计数器清除的抑制功能

在 MTU6 和 MTU7 中，能通过将 TWCRCB.SCC 位置“1”，抑制 MTU0、1、2 的同步清除。

在图 21.63 所示的区间，能通过设定 SCC 位抑制 MTU0、1、2 的同步清除。

在使用此功能时，必须将 MTU6 和 MTU7 设定为互补 PWM 模式。

有关 MTU0、1、2 的同步清除的详细内容，请参照“21.3.10(2) 使用标志置位源的 MTU6 和 MTU7 的计数器清除 (MTU6 和 MTU7 的同步计数器清除)”。

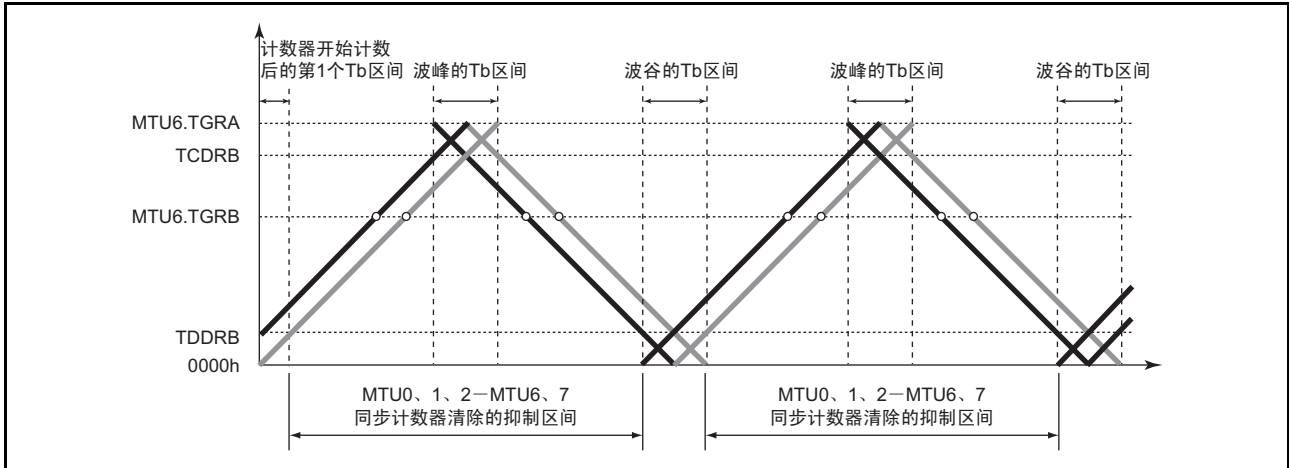


图 21.63 通过将 TWCRCB.SCC 位置位产生 MTU0、1、2 – MTU6、7 的同步清除抑制区间

• MTU0、1、2 – MTU6、7 的同步计数器清除抑制功能的设定步骤例子

MTU0、1、2 – MTU6、7 的同步计数器清除抑制功能的设定步骤例子如图 21.64 所示。

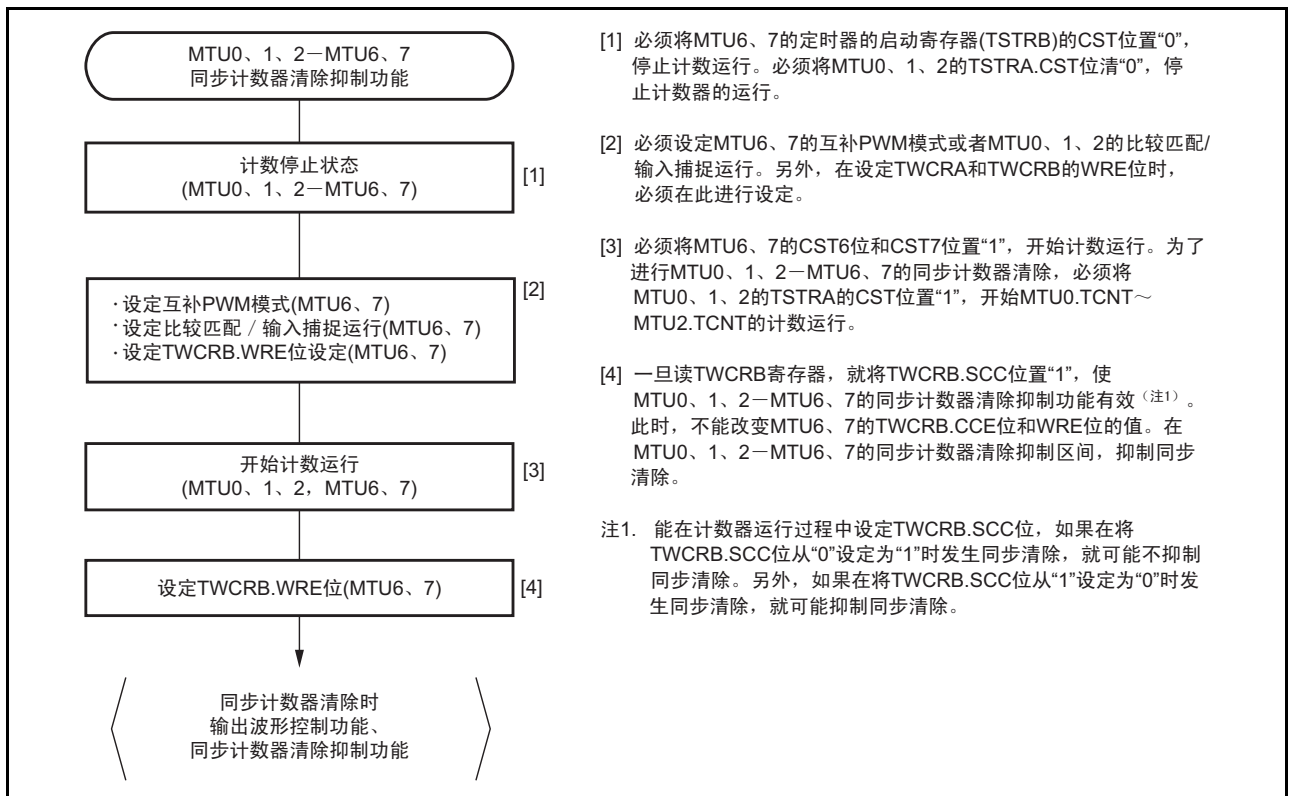


图 21.64 MTU0、1、2 – MTU6、7 的同步计数器清除抑制功能的设定步骤例子

- MTU0、1、2—MTU6、7 的同步计数器清除抑制功能的运行例子

通过将 MTU6 和 MTU7 的 TWCRB.SCC 位置“1”使 MTU6 和 MTU7 进行互补 PWM 运行并且将 MTU0、1、2—MTU6、7 的同步计数器清除抑制功能设定为有效时的运行例子如图 21.65 ~ 图 21.68 所示。在此，图 21.65 ~ 图 21.68 的同步计数器清除时序分别是图 21.57 的③、⑥、⑧、⑪所示的时序。在此例中，将 MTU6 和 MTU7 的 TWCRB.WRE 位置“1”。

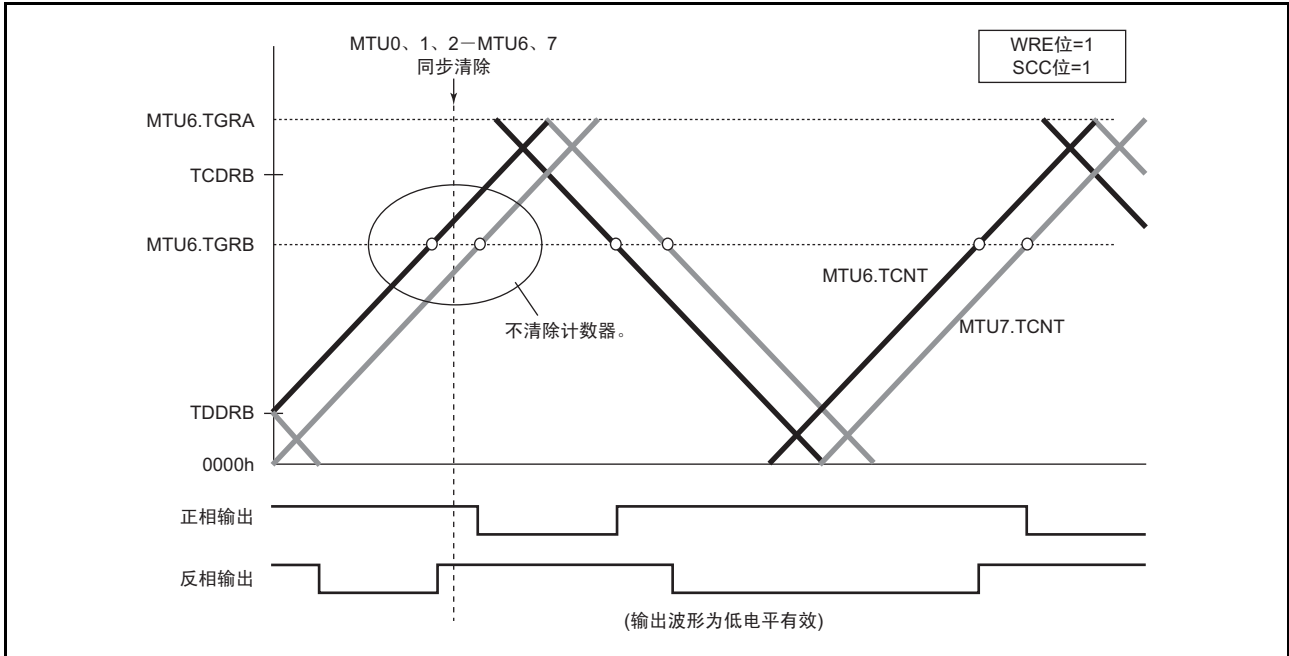


图 21.65 在递增计数过程中的死区时间发生同步清除的情况
(图 21.57 的时序③、MTU6 和 MTU7 的 TWCRB 寄存器的 WRE 位 =1、SCC 位 =1)

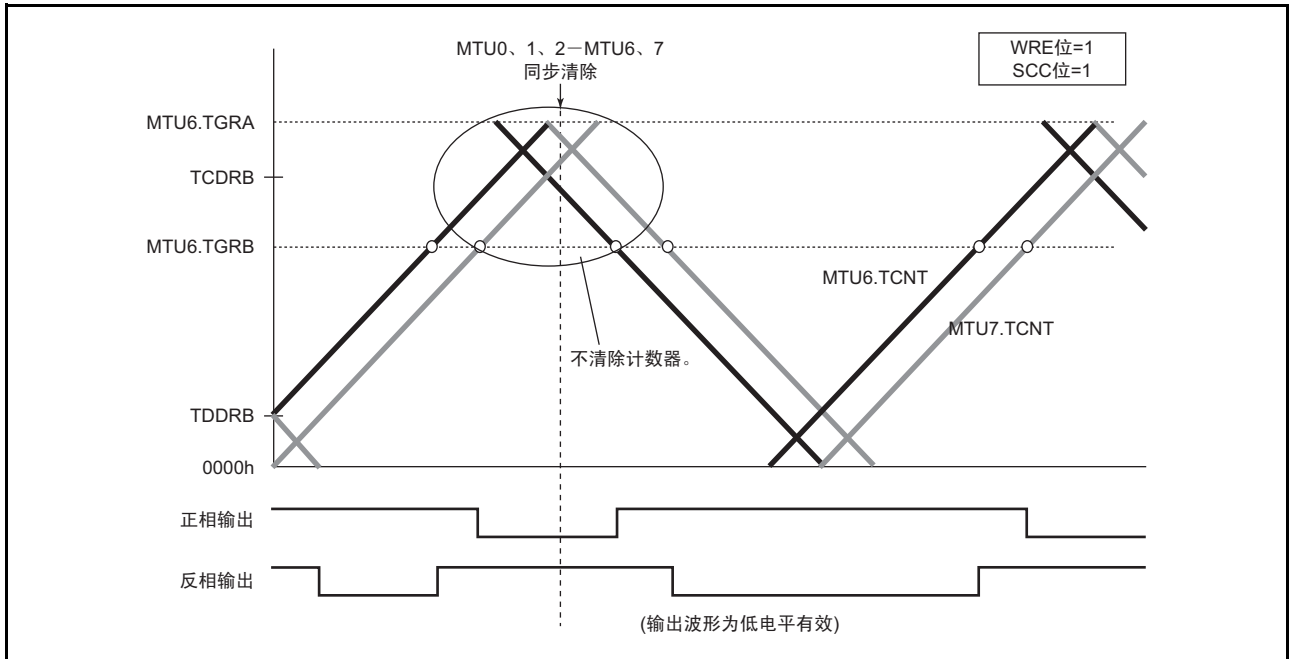


图 21.66 在波峰 Tb 区间发生同步清除的情况
(图 21.57 的时序⑥、MTU6 和 MTU7 的 TWCRB 寄存器的 WRE 位 =1、SCC 位 =1)

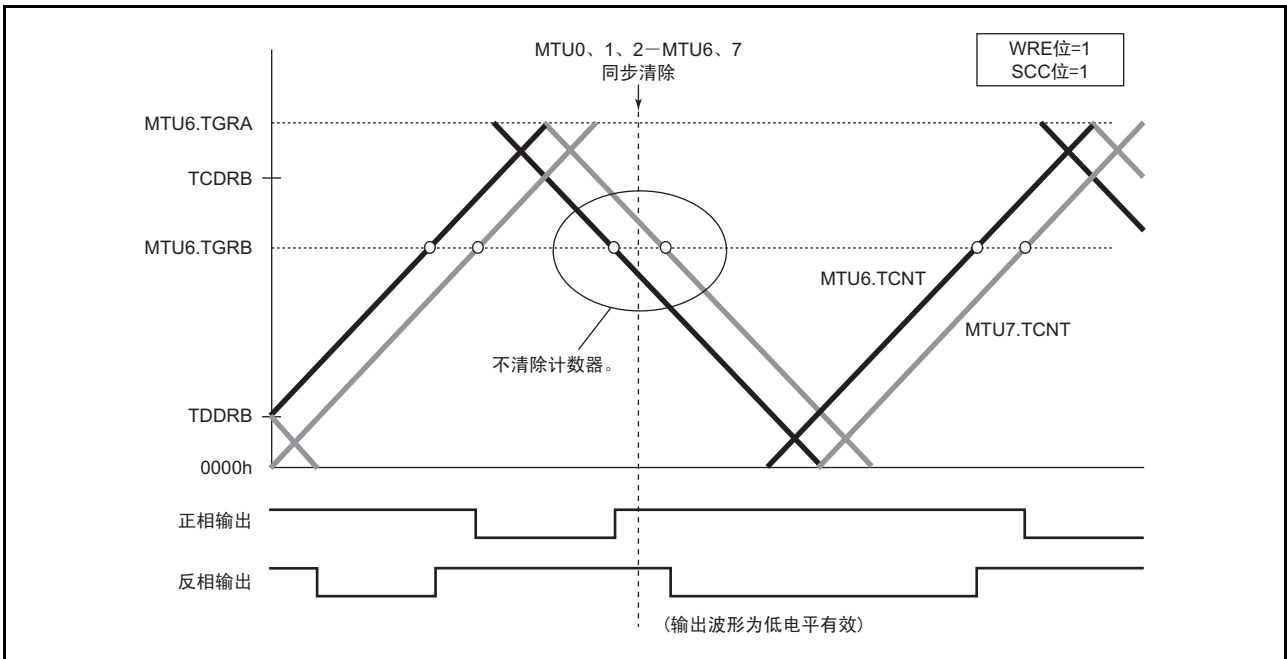


图 21.67 在递减计数过程中的死区时间发生同步清除的情况
 (图 21.57 的时序⑧、MTU6 和 MTU7 的 TWCRB 寄存器的 WRE 位 =1、SCC 位 =1)

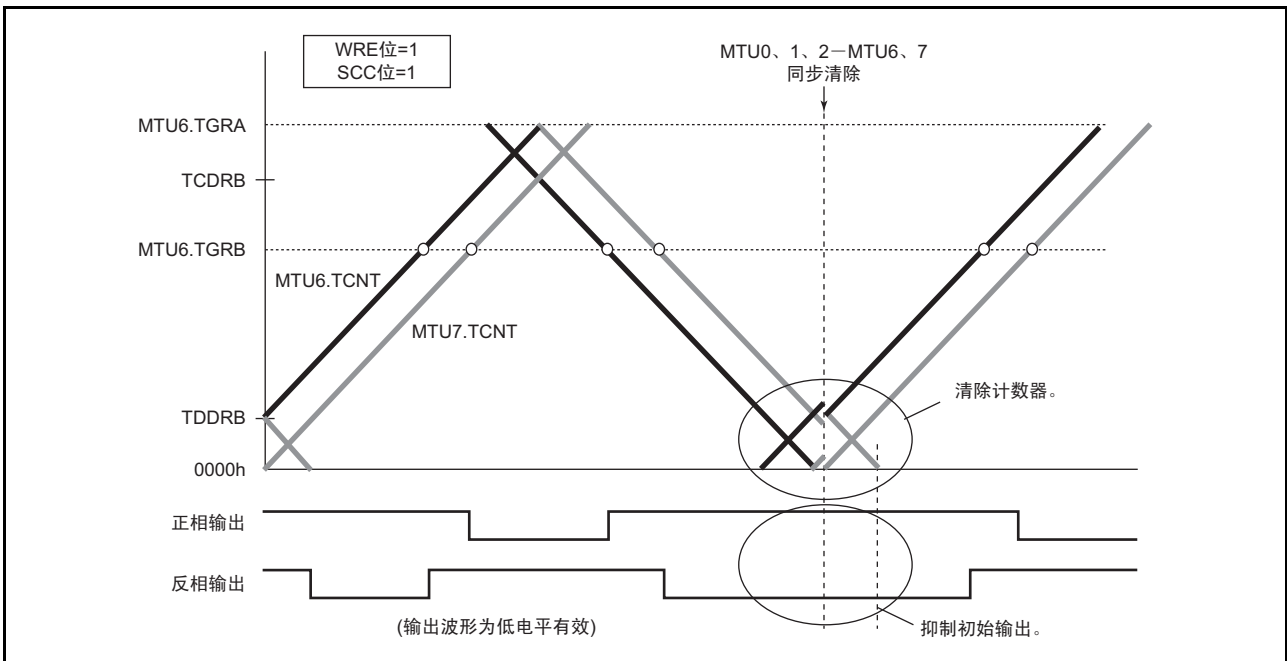


图 21.68 在波谷 Tb 区间发生同步清除的情况
 (图 21.57 的时序⑨、MTU6 和 MTU7 的 TWCRB 寄存器的 WRE 位 =1、SCC 位 =1)

(p) 通过 MTU3.TGRA (MTU6.TGRA) 的比较匹配进行的计数器清除

在互补 PWM 模式中，能通过设定定时器的波形控制寄存器 (TWCRA、TWCRB) 的 CCE 位，在 MTU3.TGRA (MTU6.TGRA) 的比较匹配时清除 MTU3.TCNT (MTU6.TCNT)、MTU4.TCNT (MTU7.TCNT) 和 TCNTSA (TCNTSB)。运行例子如图 21.69 所示。

- 注 1. 只能在互补 PWM 模式 1 (在波峰进行传送) 中使用。
- 注 2. 不能设定为与其他通道同步清除功能 (不能将定时器的同步寄存器 (TSYRA、TSYRB) 的 SYNC0 ~ SYNC4 位和 SYNC6 ~ SYNC7 置“1”，将定时器的同步清除寄存器 (TSYCRA、TSYCRB) 的 CE0A/B/C/D、CE1A/B/C/D 位置“1”)。
- 注 3. 不能将 PWM 占空比设定为“0000h”。
- 注 4. 不能将定时器的输出控制寄存器 1 (TOCR1A、TOCR1B) 的 PSYE 位置“1”。

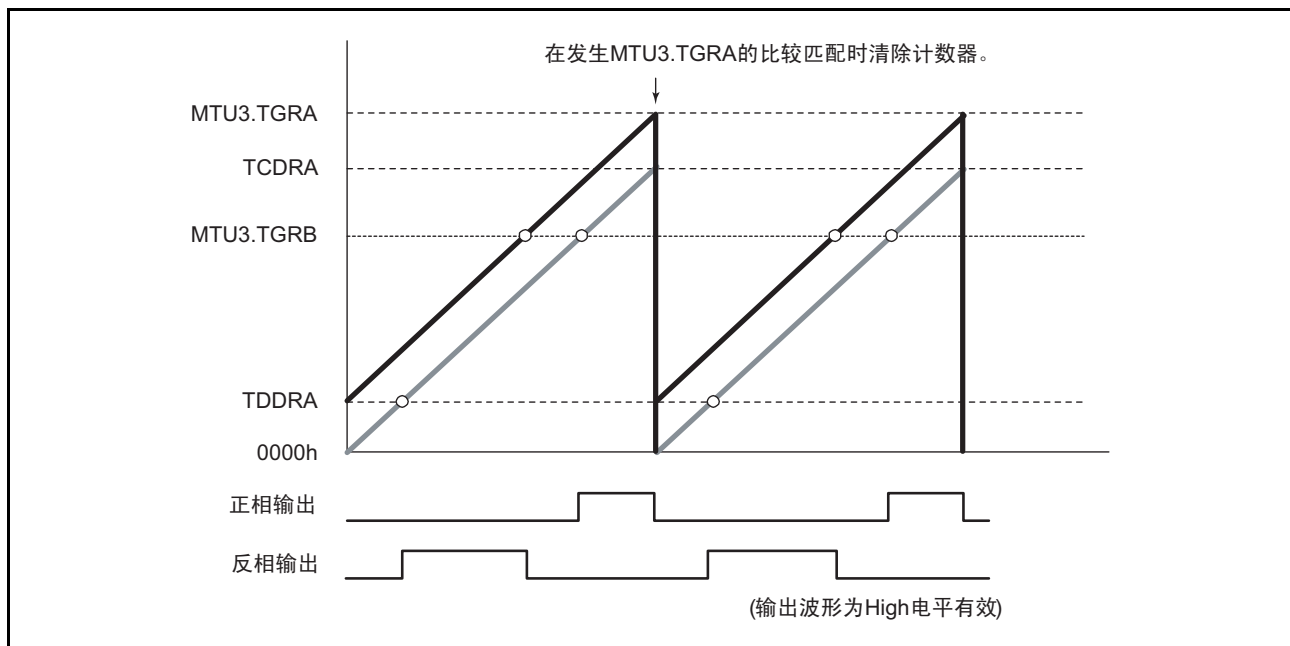


图 21.69 在 MTU3.TGRA 的比较匹配时清除计数器的运行例子

(q) AC 同步马达 (无刷 DC 马达) 的驱动波形输出例子

在互补 PWM 模式中, 能使用定时器的门控寄存器 (TGCRA), 简单地控制无刷 DC 马达。使用 TGCRA 寄存器的无刷 DC 马达的驱动波形例子如图 21.70 ~ 图 21.73 所示。

在通过使用霍尔元件等检测到的外部信号进行 3 相无刷 DC 马达的输出相转换时, 必须将 TGCRA.FB 位置“0”。此时, 将表示磁极位置的外部信号输入到 MTU0 的定时器输入引脚 MTIOC0A、MTIOC0B、MTIOC0C (必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 进行设定)。如果在 MTIOC0A、MTIOC0B、MTIOC0C 这 3 个引脚产生边沿, 输出的 ON/OFF 就自动进行转换。

在 TGCRA.FB 位为“1”的情况下, 如果将 TGCRA 的 UF 位、VF 位和 WF 位置“0”或者“1”, 输出的 ON/OFF 就进行转换。

从互补 PWM 模式的 6 相输出引脚输出驱动波形。

对于此 6 相输出, 能通过将 TGCRA 寄存器的 N 位或者 P 位置“1”, 在 ON 输出时使用互补 PWM 模式的输出, 进行斩波输出。如果 N 位或者 P 位为“0”, 就为电平输出。

6 相输出的有效电平 (ON 输出时的电平) 与 N 位和 P 位的设定无关, 能通过定时器的输出控制寄存器 1 (TOCR1A) 的 OLSN 位和 OLSP 位进行设定。

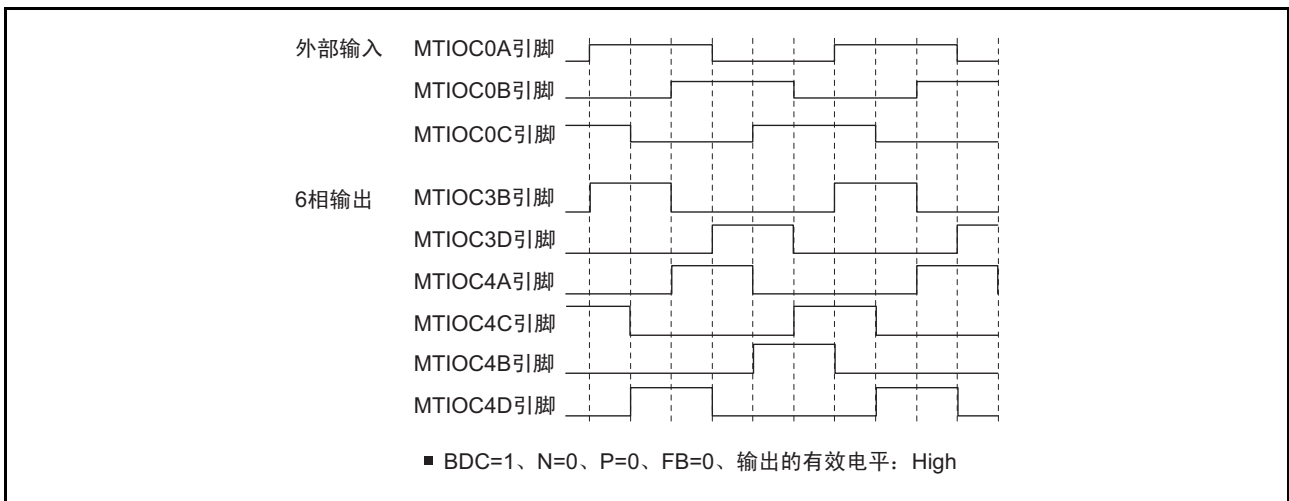


图 21.70 通过外部输入进行输出相转换的运行例子 (1)

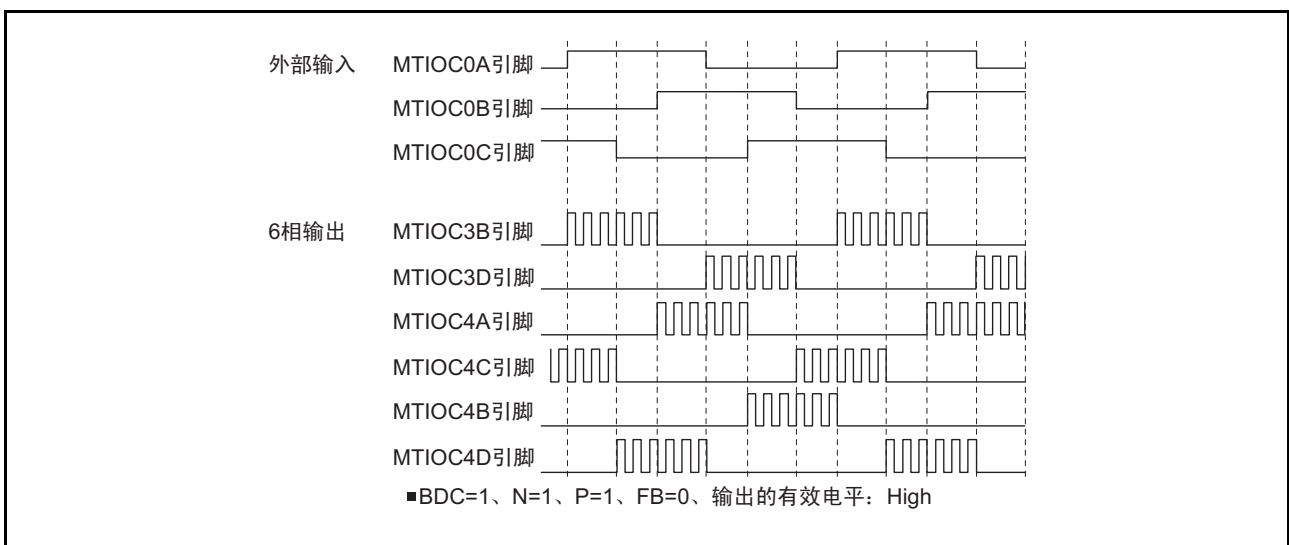


图 21.71 通过外部输入进行输出相转换的运行例子 (2)

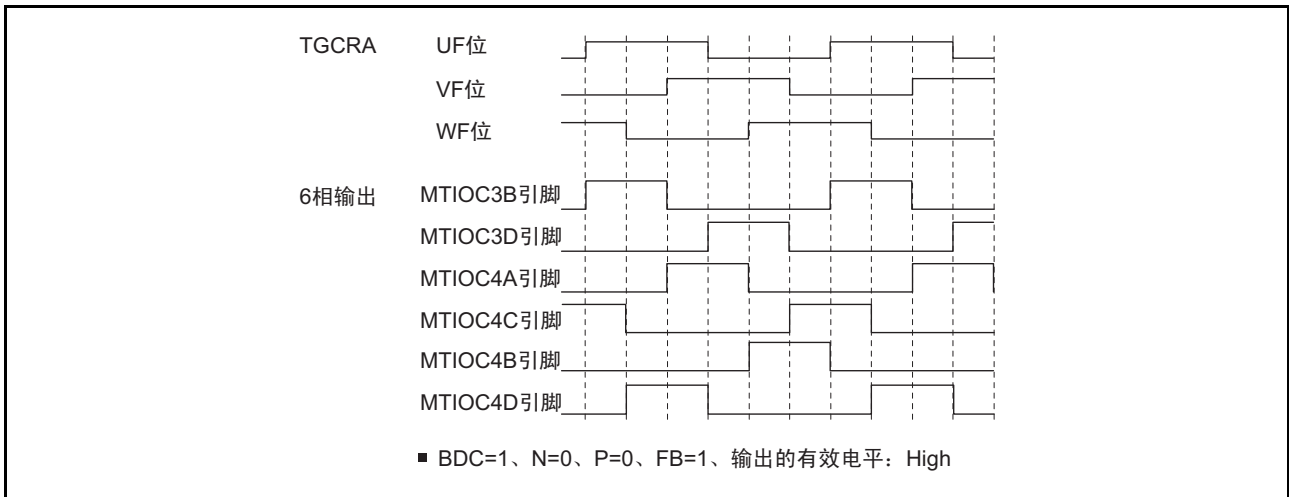


图 21.72 通过设定 UF 位、VF 位和 WF 位进行输出相转换的运行例子 (1)

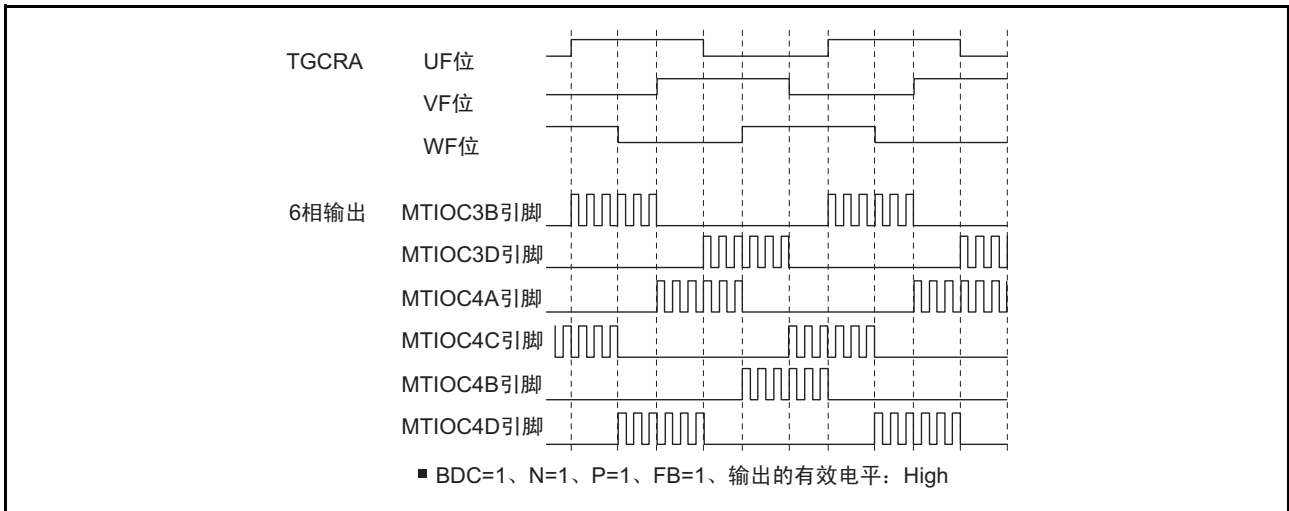


图 21.73 通过设定 UF 位、VF 位和 WF 位进行输出相转换的运行例子 (2)

(r) A/D 转换开始请求的设定

在互补 PWM 模式中，能使用 MTU3.TGRA (MTU6.TGRA) 的比较匹配、MTU4.TCNT (MTU7.TCNT) 的下溢 (波谷)、以及 MTU3 和 MTU4 (MTU6 和 MTU7) 以外通道的比较匹配，请求开始 A/D 转换。

如果使用 MTU3.TGRA (MTU6.TGRA) 的比较匹配设定开始请求，就能在 MTU3.TCNT (MTU6.TCNT) 的波峰开始 A/D 转换。

能通过将定时器的中断允许寄存器 (TIER) 的 TTGE 位置“1”，设定 A/D 转换的开始请求，并且能通过将 MTU4.TIER (MTU7.TIER) 的 TTGE2 位置“1”，设定 MTU4.TCNT (MTU7.TCNT) 下溢 (波谷) 的 A/D 转换开始请求。

(s) 互补 PWM 模式的双缓冲功能

在互补 PWM 模式 3 (在波峰和波谷进行传送) 中, 能通过将定时器的模式寄存器 2 (TMDR2A、TMDR2B) 的 DRS 位置 “1”, 将更改 PWM 时的 PWM 输出的最小分辨率设定为 $\pm 2 \sim \pm 1$ 。

必须在设定缓冲寄存器 A (MTU3.TGRD、MTU4.TGRC、MTU4.TGRD) 的同时, 设定缓冲寄存器 B (MTU3.TGRE、MTU4.TGRE、MTU4.TGRF), 还必须给缓冲寄存器 B 设定缓冲寄存器 A 的值或者缓冲寄存器 A 的值 -1。设定步骤的详细内容请参照 “21.3.8(1) 互补 PWM 模式的设定步骤例子”。

注 如果给缓冲寄存器 B 设定缓冲寄存器 A 的值, PWM 输出就左右对称; 如果给缓冲寄存器 B 设定缓冲寄存器 A 的值 -1, PWM 输出就不对称。

双缓冲功能的运行例子如图 21.74 所示。

各寄存器的数据传送方式如下:

- 在写 MTU4.TGRD (缓冲器 A) 时将数据从 MTU4.TGRD (缓冲器 A) 传送到 Temp3A (暂存器 A) 以及从 MTU4.TGRF (缓冲器 B) 传送到 Temp3B (暂存器 B)。
- 在①的时序将数据从 Temp3A (暂存器 A) 传送到 MTU4.TGRB (比较寄存器)。
- 在②的时序将数据从 Temp3B (暂存器 B) 传送到 MTU4.TGRB (比较寄存器)。

在波峰区间 (波峰的 Tb 区间), 比较寄存器和暂存器 A 有效; 在波谷区间 (波谷的 Tb 区间), 比较寄存器和暂存器 B 有效。

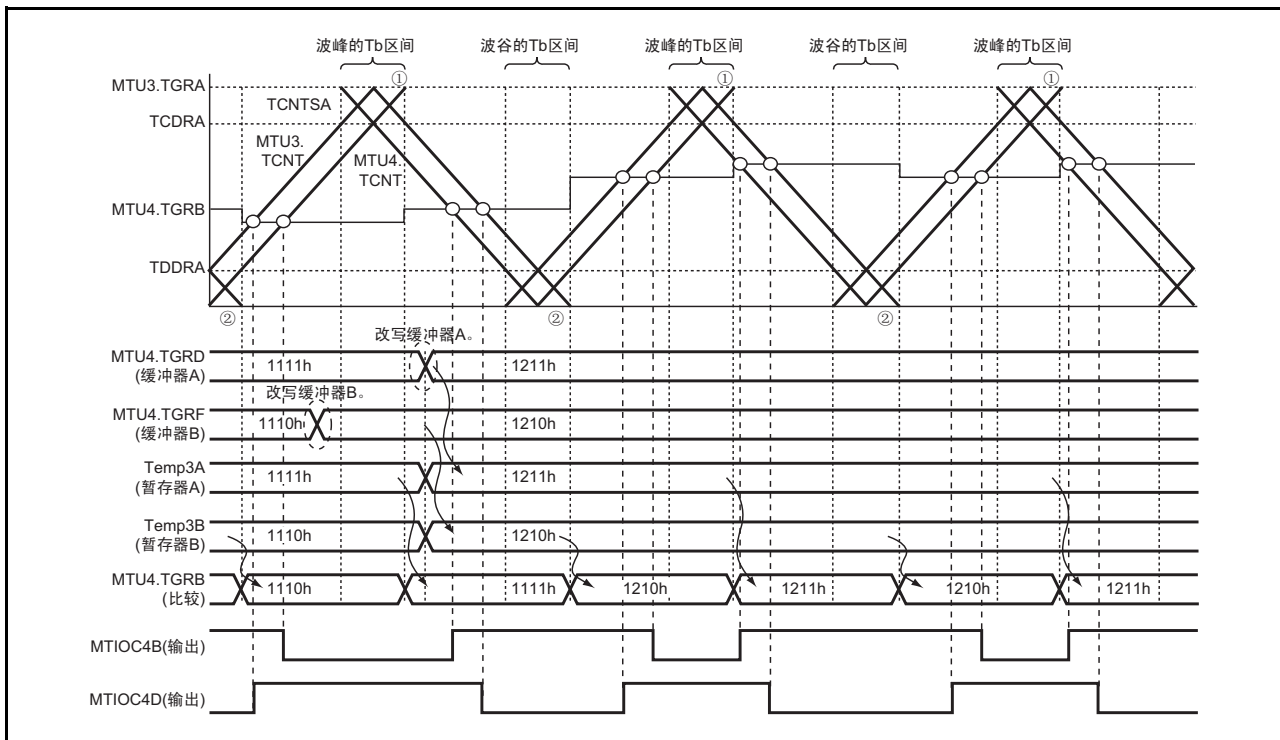


图 21.74 双缓冲功能的运行例子

缓冲器的写入值小于和大于 TDDRA 的值的情况分别如图 21.75 和图 21.76 所示。

在波峰区间，通过和比较寄存器或者暂存器 A 的比较匹配进行输出控制；在波谷区间，通过和比较寄存器或者暂存器 B 的比较匹配进行输出控制。

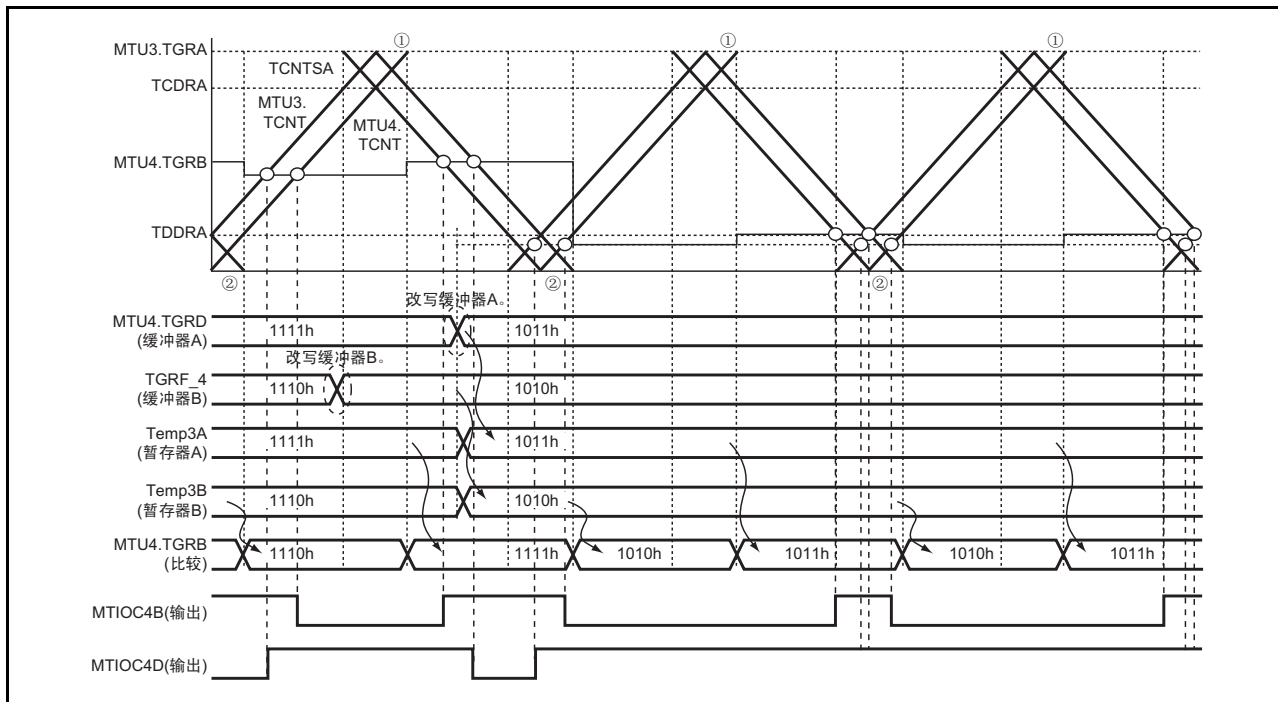


图 21.75 双缓冲功能的运行例子 (缓冲器的写入值小于 TDDRA 值的情况)

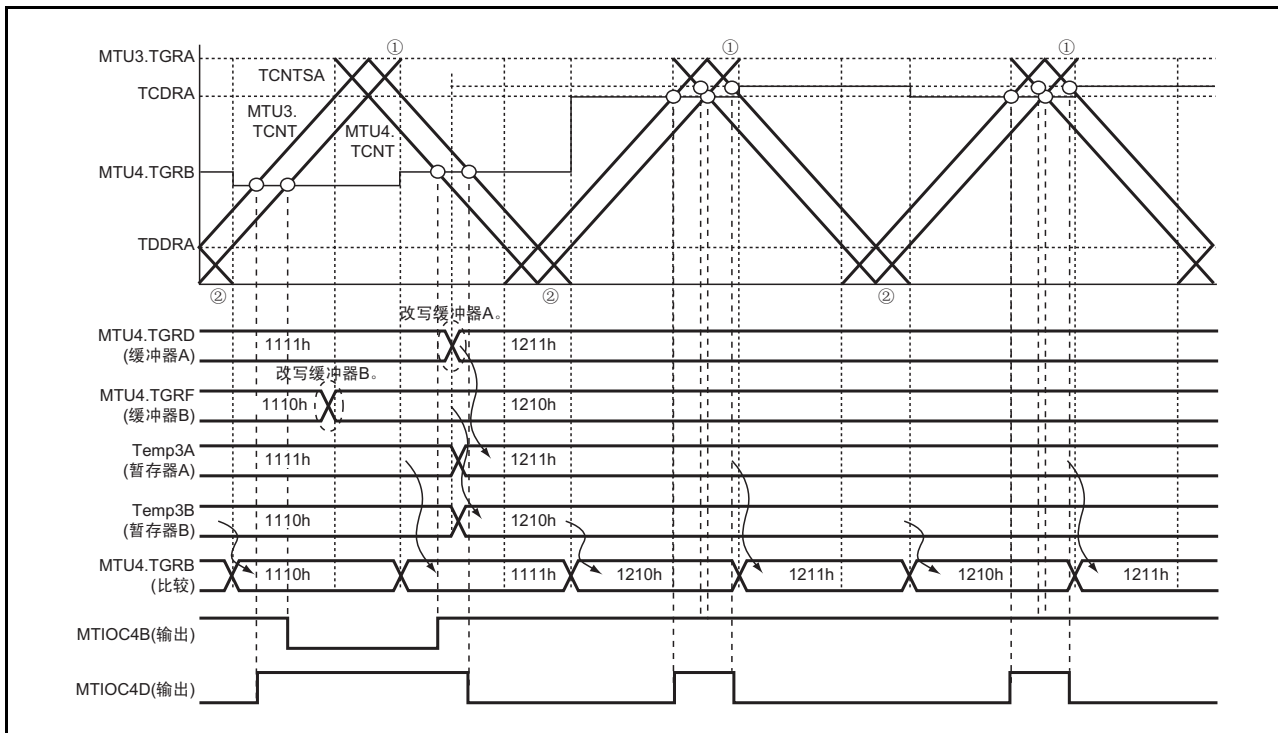


图 21.76 双缓冲功能的运行例子 (缓冲器的写入值大于 TCDRA 值的情况)

(3) 互补 PWM 模式的中断减少功能 1

能通过设定定时器的中断减少设定寄存器 1 (TITCR1A、TITCR1B)，使 MTU3 和 MTU4 (MTU6 和 MTU7) 的 TGIA3 (TGIA6) (波峰的中断) 以及 TCIV4 (TCIV7) (波谷的中断) 最多减少 7 次中断。

能通过设定定时器的缓冲传送寄存器 (TBTERA、TBTERB)，联动缓冲寄存器到暂存器 / 比较寄存器的数据传送来减少中断。有关和缓冲寄存器的联动，请参照“(c) 联动中断减少功能 1 的缓冲传送控制”。

能通过设定定时器的 A/D 转换请求控制寄存器 (MTU4.TADCR、MTU7.TADCR)，联动 A/D 转换开始请求延迟功能的 A/D 转换开始请求来减少中断。有关和 A/D 转换开始请求延迟功能的联动，请参照“21.3.9 A/D 转换开始请求的延迟功能”。

必须通过将定时器的中断减少模式寄存器 (TITMRA、TITMRB) 的 TITM 位置“0”，选择中断减少功能 1，并且通过设定 MTU3.TIER 和 MTU4.TIER (MTU6.TIER 和 MTU7.TIER) 寄存器，禁止 TGIA3 (TGIA6) 和 TCIV4 (TCIV7) 中断请求，在不发生由比较匹配产生的 TITCR1A (TITCR1B) 寄存器的 TGFA3 (TGFA6) 和 TCFV4 (TCFV7) 标志置位的状态下设定定时器的中断减少设定寄存器 1 (TITCR1A、TITCR1B)。在更改减少次数前，必须将 T3AEN (T6AEN) 和 T4VEN (T7VEN) 位置“0”，清除减少计数器。

(a) 中断减少功能 1 的设定步骤例子

中断减少功能 1 的设定步骤例子和中断减少次数的可变更期间分别如图 21.77 和图 21.78 所示。

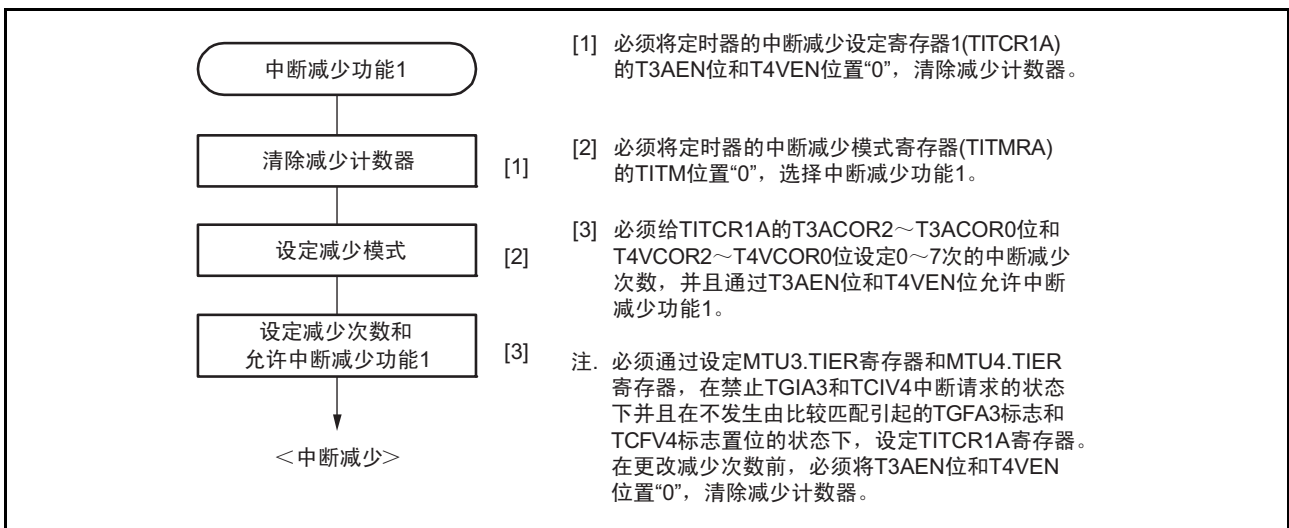


图 21.77 中断减少功能 1 的设定步骤例子

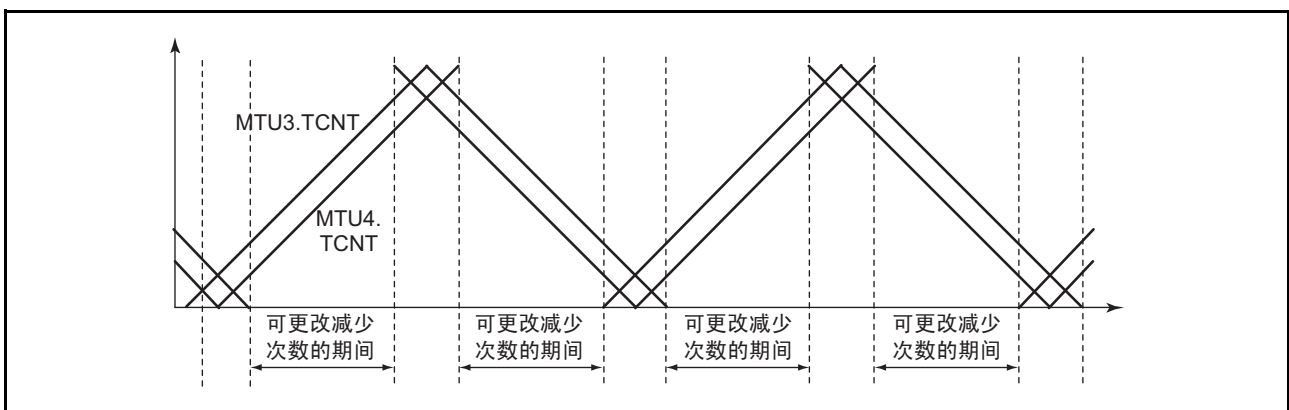


图 21.78 中断减少次数的可更改期间

(b) 中断减少功能 1 的运行例子

通过定时器的中断减少设定寄存器 1 (TITCR1A、TITCR1B) 的 T3ACOR (T6ACOR) 位将中断减少次数设定为 3 次, 将 T3AEN (T6AEN) 位置“1”时的 TGIA3 (TGIA6) 中断减少的运行例子如图 21.79 所示。

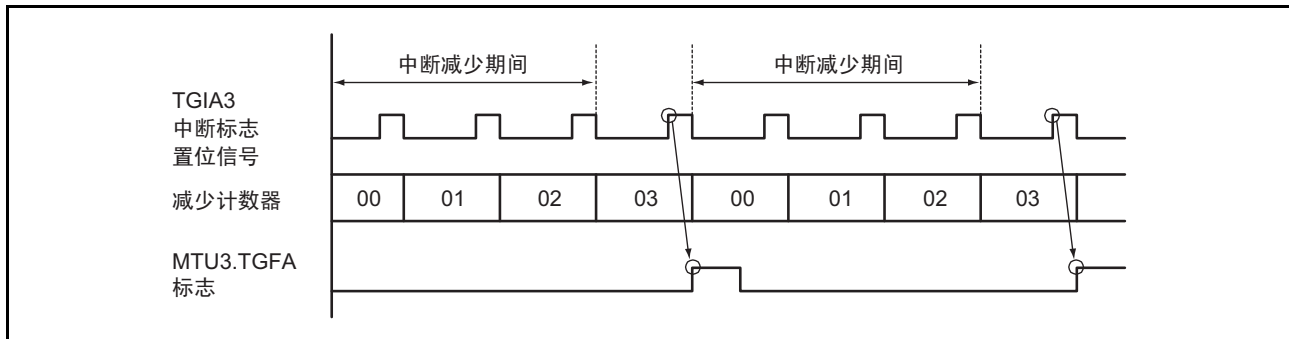


图 21.79 中断减少功能 1 的运行例子

(c) 联动中断减少功能 1 的缓冲传送控制

能通过设定定时器的缓冲传送设定寄存器 (TBTERA、TBTERB) 的 BTE[1:0] 位, 选择在互补 PWM 模式中是否进行缓冲寄存器到暂存器的缓冲传送以及是否联动中断减少功能 1。

设定为抑制缓冲传送 (BTE[1:0]=01b) 时的运行例子如图 21.80 所示。在此设定期间中, 不进行缓冲寄存器到暂存器的传送。

设定为缓冲传送联动中断减少功能 1 (BTE[1:0]=10b) 时的运行例子如图 21.81 所示。此时, 除了缓冲传送允许期间以外, 不进行缓冲寄存器到暂存器的传送。

如果将定时器的中断减少设定寄存器 1 (TITCR1A、TITCR1B) 的 T3AEN (T6AEN) 位置“1”, 或者将 T4VEN (T7VEN) 位置“1”, 或者将 TITCR1A (TITCR1B) 寄存器的 T3AEN 和 T4VEN (T6AEN 和 T7VEN) 位置“1”, 各缓冲传送允许期间就不同。TITCR1A (TITCR1B) 的 T3AEN、T4VEN (T6AEN、T7VEN) 位的设定和缓冲传送允许期间的关系如图 21.82 所示。

注. 此功能必须和中断减少功能 1 配合使用。

在禁止中断减少 (将定时器的中断减少设定寄存器 1 (TITCR1A、TITCR1B) 的 T3AEN 位和 T4VEN (T6AEN 和 T7VEN) 位置“0”, 或者将 TITCR1A (TITCR1B) 的减少次数设定位 (T3ACOR 和 T4VCOR (T6ACOR 和 T7VCOR)) 置“0”) 时, 必须设定为缓冲传送不联动中断减少功能 1 (将 TBTERA、TBTERB 的 BTE1 位置“0”)。在禁止中断减少时, 如果设定为缓冲传送联动中断减少功能 1, 就不进行缓冲传送。

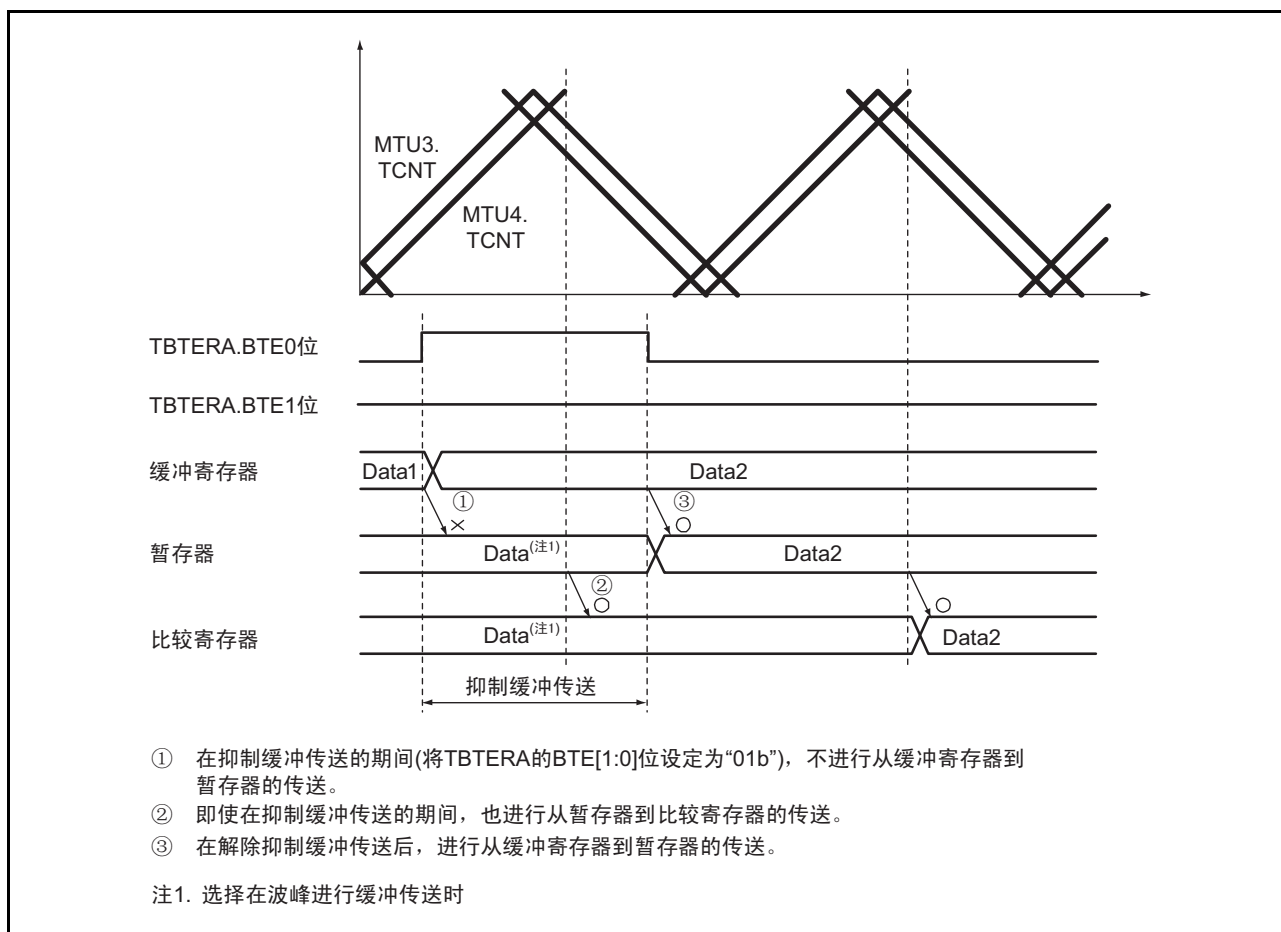


图 21.80 设定为抑制缓冲传送 (BTE[1:0]=01b) 时的运行例子

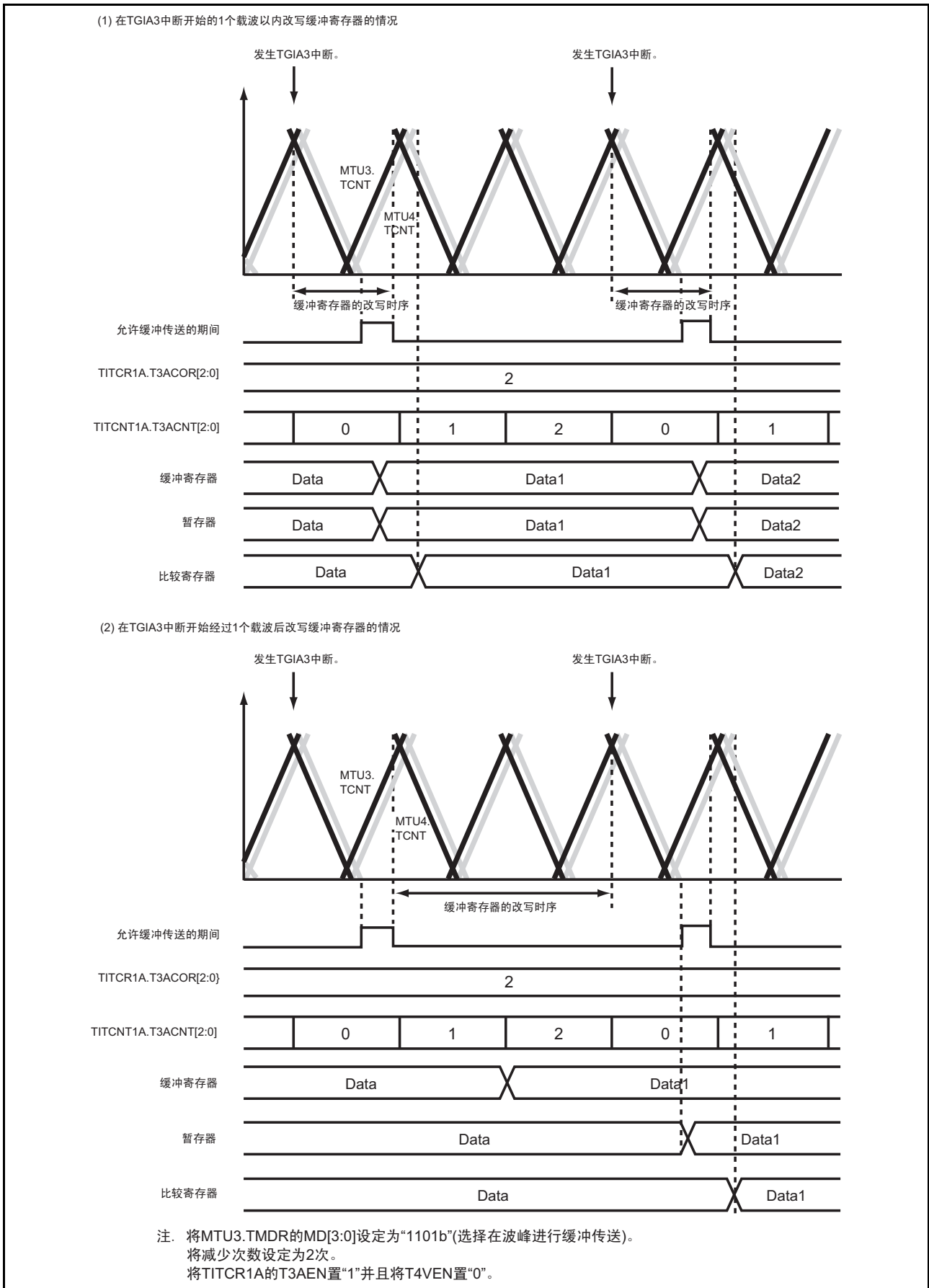


图 21.81 设定为缓冲传送联动中断减少功能 1 (BTE[1:0]=0) 时的运行例子

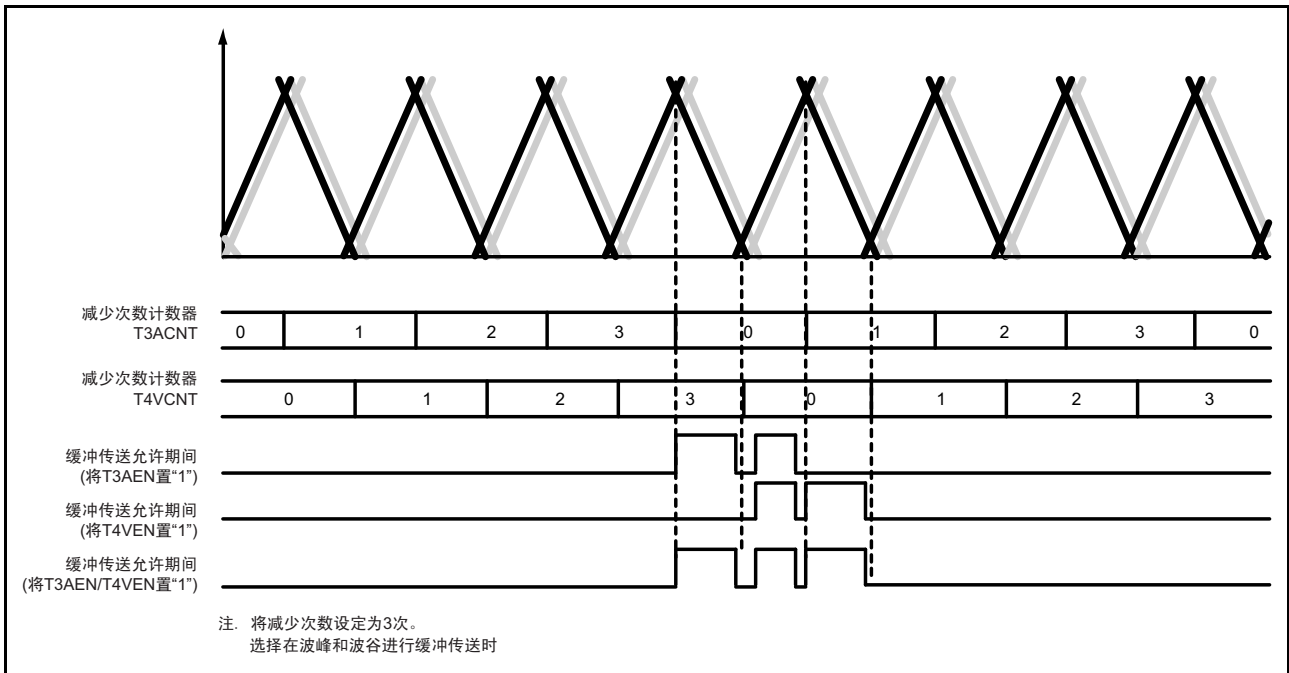


图 21.82 TITCR1A 寄存器的 T3AEN 位、T4VEN 位的设定和缓冲传送允许期间的关系

(4) 互补 PWM 模式的输出保护功能

互补 PWM 模式的输出具有以下保护功能。

(a) 寄存器和计数器的误写防止功能

在互补 PWM 模式中使用的寄存器和计数器当中，除了随时能改写的缓冲寄存器以外，能通过设定定时器的读写允许寄存器 (TRWERA、TRWERB) 的 RWE 位，选择允许或者禁止 CPU 存取模式寄存器、控制寄存器、比较寄存器和计数器。对象寄存器为 MTU3、4、6、7 的部分寄存器，适用于以下寄存器：

- MTU3.TCR、MTU4.TCR、MTU3.TMDR1、MTU4.TMDR1、MTU3.TIORH、MTU4.TIORH、
 - MTU3.TIORL、MTU4.TIORL、MTU3.TIER、MTU4.TIER、MTU3.TCNT、MTU4.TCNT、MTU3.TGRA、
 - MTU4.TGRA、MTU3.TGRB、MTU4.TGRB、TOERA、TOCR1A、TOCR2A、TGCRA、TCDRA、
 - TDDRA、MTU6.TCR、MTU7.TCR、MTU6.TMDR1、MTU7.TMDR1、MTU6.TIORH、MTU7.TIORH、
 - MTU6.TIORL、MTU7.TIORL、MTU6.TIER、MTU7.TIER、MTU6.TCNT、MTU7.TCNT、MTU6.TGRA、
 - MTU7.TGRA、MTU6.TGRB、MTU7.TGRB、TOERB、TOCR1B、TOCR2B、TCDRB、TDDRB
- 共计 43 个寄存器

通过此功能，能设定为禁止 CPU 存取模式寄存器、控制寄存器和计数器，以防止因 CPU 失控而产生的误写。在禁止存取的状态下读对象寄存器时，读取值为不定值，写操作无效。

(b) 通过外部信号进行的 PWM 输出停止功能

能通过输入指定的外部信号，使 6 相 PWM 输出引脚自动变为高阻抗状态。
详细内容请参照“22. 端口输出允许 3 (POE3)”。

(c) 振荡停止时的 PWM 输出停止功能

在检测到输入到 RX63T 的时钟停止后，6 相 PWM 输出引脚自动变为高阻抗状态。但是，一旦时钟重新开始振荡，就无法保证此引脚状态。

详细内容请参照“9.4 振荡停止检测功能”。

21.3.9 A/D 转换开始请求的延迟功能

能通过设定 MTU4 或者 MTU7 的定时器的 A/D 转换开始请求控制寄存器 (MTU4.TADCR、MTU7.TADCR)、定时器的 A/D 转换开始请求周期设定寄存器 (MTU4.TADCORA、MTU4.TADCORB、MTU7.TADCORA、MTU7.TADCORB) 或者定时器的 A/D 转换开始请求周期设定缓冲寄存器 (MTU4.TADCOBRA、MTU4.TADCOBRB、MTU7.TADCOBRA、MTU7.TADCOBRB)，进行 A/D 转换的开始请求。

A/D 转换开始请求的延迟功能是将 MTU4.TCNT 和 MTU4.TADCORA、MTU4.TADCORB (MTU7.TCNT 和 MTU7.TADCORA、MTU7.TADCORB) 进行比较，如果 MTU4.TCNT 和 MTU4.TADCORA、MTU4.TADCORB (MTU7.TCNT 和 MTU7.TADCORA、MTU7.TADCORB) 相同，就进行各自的 A/D 转换开始请求 (TRG4AN、TRG4BN (TRG7AN、TRG7BN))。

能通过设定 MTU4.TADCR 的 ITA3AE、ITA4VE、ITB3AE、ITB4VE (MTU7.TADCR 的 ITA6AE、ITA7VE、ITB6AE、ITB7VE) 位，联动中断减少功能 1 减少 A/D 转换的开始请求 (TRG4AN、TRG4BN (TRG7AN、TRG7BN))。

(1) A/D 转换开始请求延迟功能的设定步骤例子

A/D 转换开始请求延迟功能的设定步骤例子如图 21.83 所示。

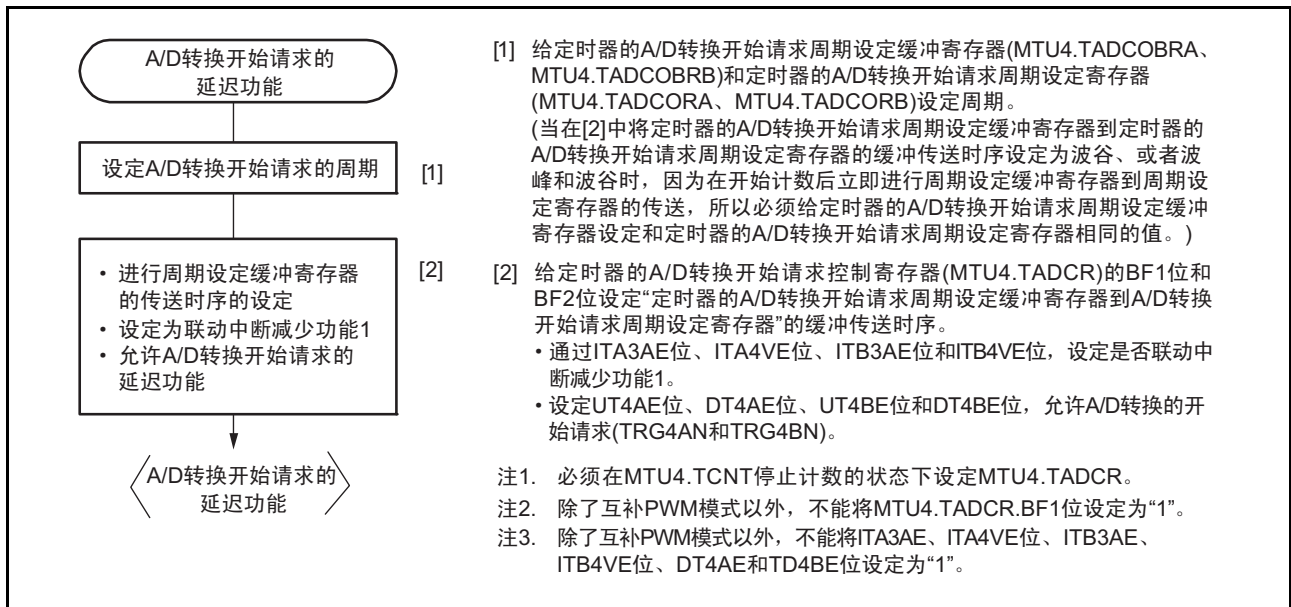


图 21.83 A/D 转换开始请求延迟功能的设定步骤例子

(2) A/D 转换开始请求延迟功能的基本运行例子

将缓冲传送时序设定为 MTU4.TCNT (MTU7.TCNT) 的波谷并且在 MTU4.TCNT (MTU7.TCNT) 进行递减计数时输出 A/D 转换开始请求信号 (TRG4AN (TRG7AN)) 时的 A/D 转换开始请求信号 (TRG4AN (TRG7AN)) 的基本运行例子如图 21.84 所示。

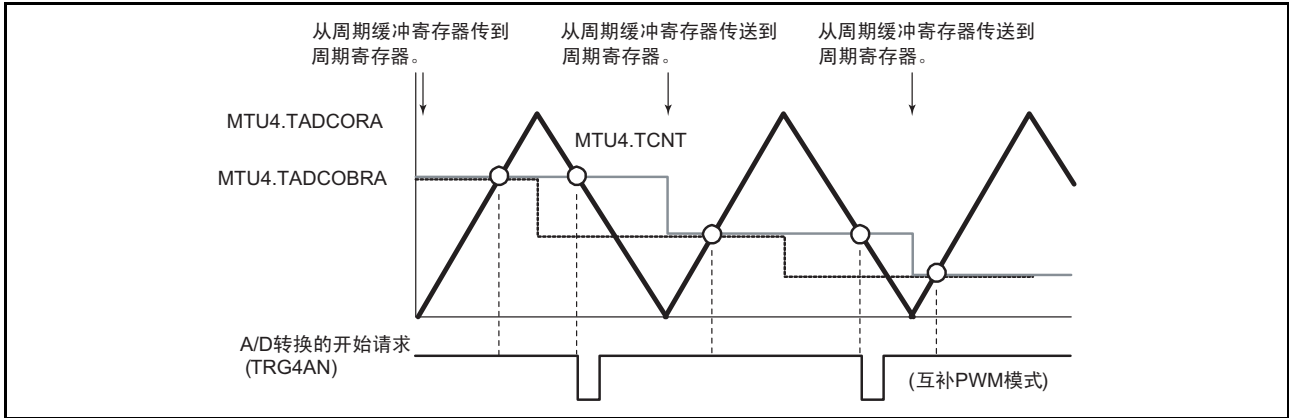


图 21.84 A/D 转换开始请求信号 (TRG4AN) 的基本运行例子

(3) 缓冲传送

通过将数据写到定时器的 A/D 转换开始请求周期设定缓冲寄存器 (MTU4.TADCOBRA、MTU4.TADCOBRB、MTU7.TADCOBRA、MTU7.TADCOBRB)，更新定时器的 A/D 转换开始请求周期设定寄存器 (MTU4.TADCORA、MTU4.TADCORB、MTU7.TADCORA、MTU7.TADCORB) 的数据。能通过设定定时器的 A/D 转换开始请求控制寄存器 (MTU4.TADCR、MTU7.TADCR) 的 BF[1:0] 位，选择从定时器的 A/D 转换开始请求周期设定缓冲寄存器到定时器的 A/D 转换开始请求周期设定寄存器的传送时序。

在互补 PWM 模式中改写定时器的通用寄存器 D (MTU4.TGRD、MTU7.TGRD) 时，也将数据从定时器的 A/D 转换开始请求周期设定缓冲寄存器传送到定时器的 A/D 转换开始请求周期设定寄存器。

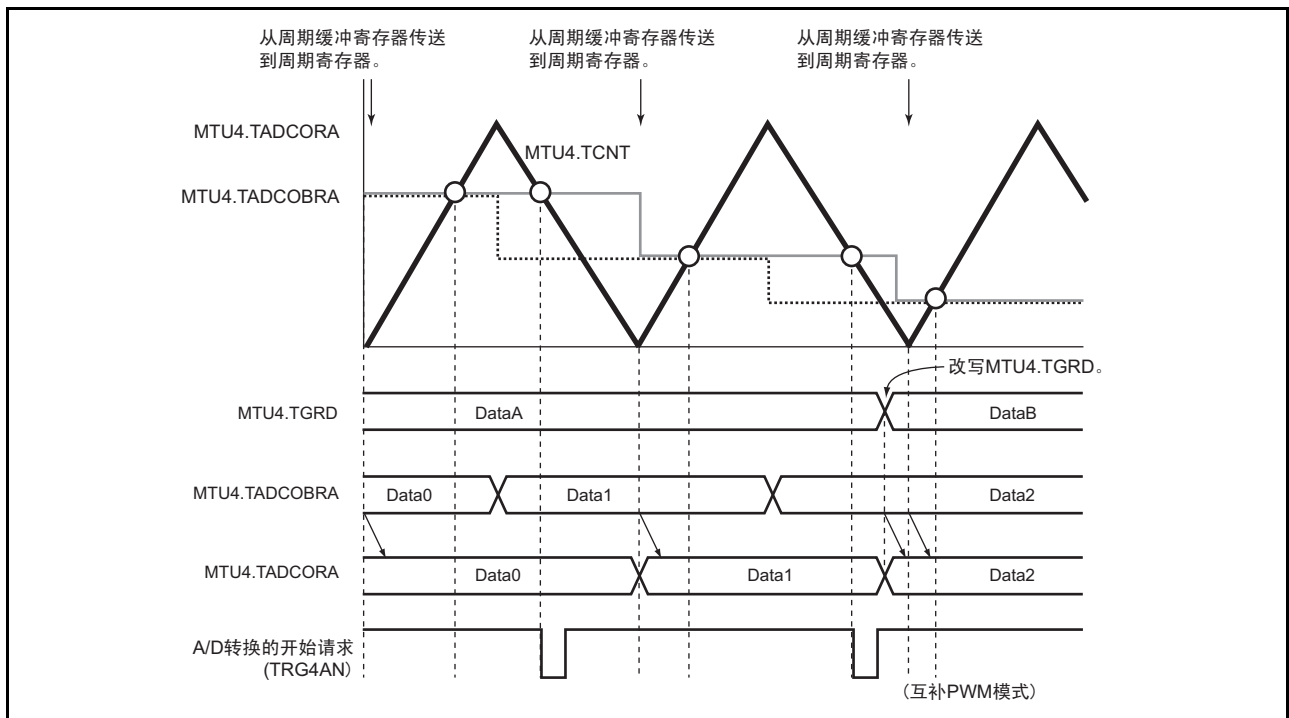


图 21.85 A/D 转换开始请求信号 (TRG4AN) 和缓冲传送的运行例子

(4) 联动中断减少功能 1 的 A/D 转换开始请求延迟功能

能通过设定定时器的 A/D 转换开始请求控制寄存器 (MTU4.TADCR、MTU7.TADCR) 的 ITA3AE、ITA4VE、ITB3AE、ITB4VE (ITA6AE、ITA7VE、ITB6AE、ITB7VE) 位, 联动中断减少功能 1 进行 A/D 转换的开始请求 (TRG4AN、TRG4BN (TRG7AN、TRG7BN))。在 MTU4.TCINT (MTU7.TCINT) 进行递增计数和递减计数时允许 TRG4AN (TRG7AN) 输出并且联动中断减少功能 1 时的 A/D 转换开始请求信号 (TRG4AN (TRG7AN)) 的运行例子如图 21.86 所示。

在 MTU4.TCINT (MTU7.TCINT) 进行递增计数时允许 TRG4AN (TRG7AN) 输出并且联动中断减少功能 1 时的 A/D 转换开始请求信号 (TRG4AN (TRG7AN)) 的运行例子如图 21.87 所示。

注. 此功能必须和中断减少功能 1 配合使用。

在禁止中断减少 (将定时器的中断减少设定寄存器 (TITCR1A (TITCR1B)) 的 T3AEN 位和 T4VEN 位 (T6AEN 位和 T7VEN 位) 置“0”, 或者将 TITCR1A (TITCR1B) 寄存器的减少次数设定位 (T3ACOR 和 T4VCOR (T6ACOR 和 T7VCOR)) 置“0”) 时, 必须设定为不联动中断减少功能 1 (将 MTU4.TADCR (MTU7.TADCR) 寄存器的 ITA3AE、ITA4VE、ITB3AE、ITB4VE (ITA6AE、ITA7VE、ITB6AE、ITB7VE) 位置“0”)。

另外, 在使用此功能时, 必须给 MTU4.TADCORA、MTU4.TADCORB (MTU7.TADCORA、MTU7.TADCORB) 设定 0002h ~ TCDRA 的设定值-2 (TCDRB 的设定值-2) 的值。

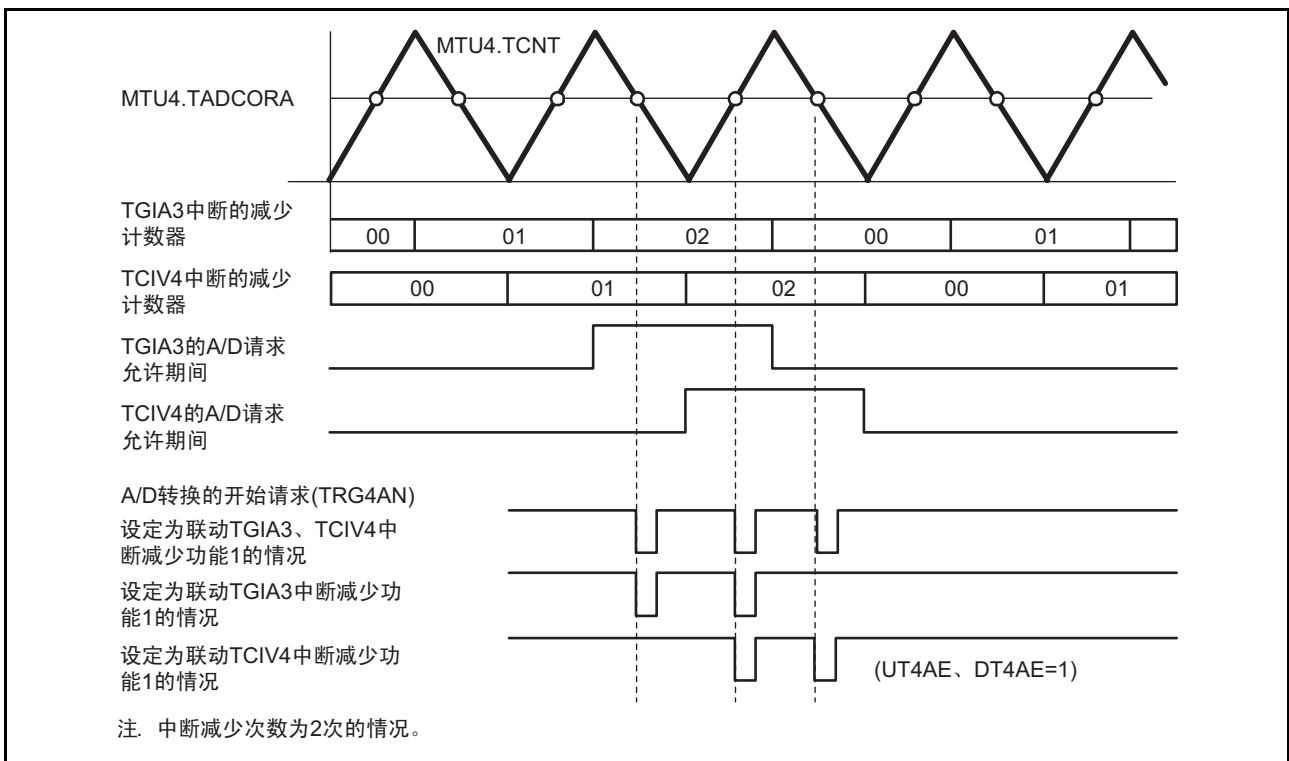


图 21.86 联动中断减少功能 1 时的 A/D 转换开始请求信号 (TRG4AN) 的运行例子 (UT4AE、DT4AE=1)

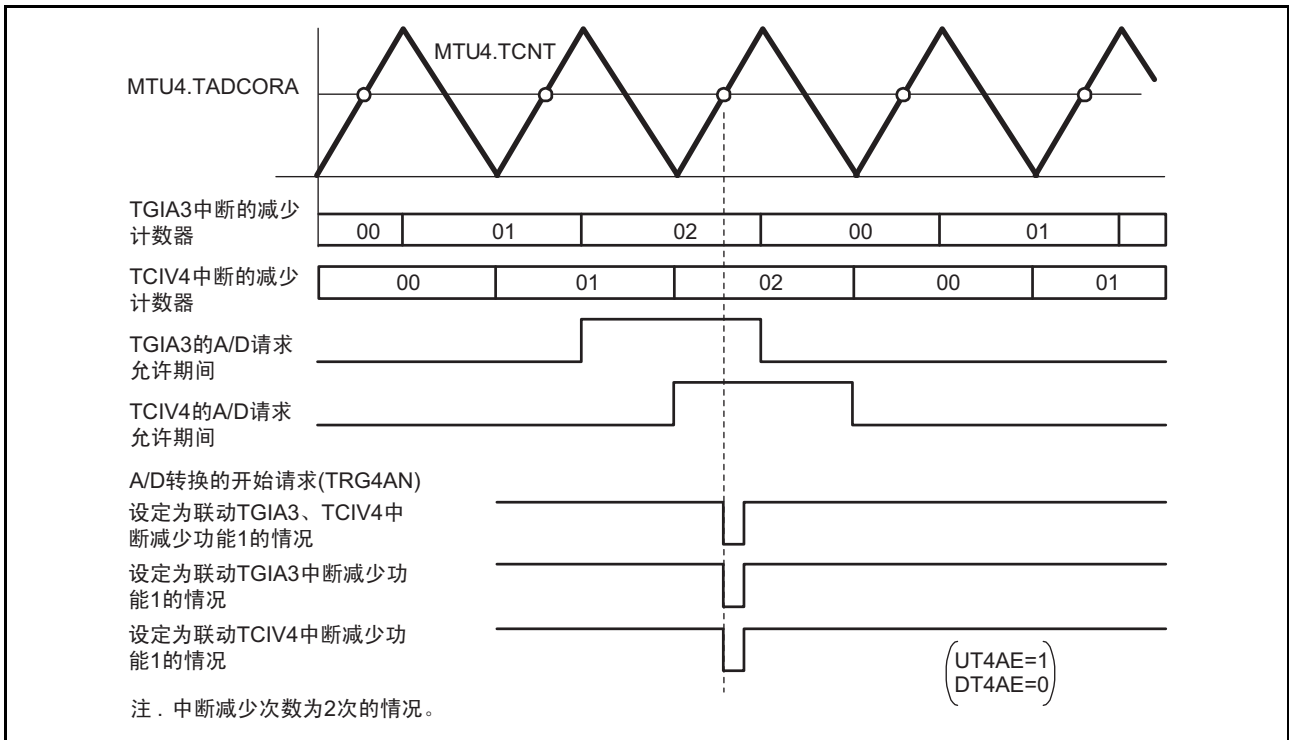


图 21.87 联动中断减少功能 1 时的 A/D 转换开始请求信号 (TRG4AN) 的运行例子 (UT4AE=1、DT4AE=0)

(5) 联动中断减少功能 2 的 A/D 转换开始请求延迟功能

在每次发生既存的 A/D 转换开始触发 (TGR4AN 和 TRG4BN (TGR7AN 和 TRG7BN)) 时, 将定时器的中断减少模式寄存器 (TITMRA、TITMRB) 的 TITM 位置“1”, 从定时器的中断减少设定寄存器 2 (TITCR2A (TITCR2B)) 的 TRG4COR[2:0] (TRG7COR[2:0]) 位的设定值 (0 ~ 7) 开始进行递减计数, 如果计数器的值变为“0”并且发生重加载, TRG4AN 和 TRG4BN (TRG7AN 和 TRG7BN) 的中断就有效, 输出 AD 转换开始请求信号 (TRG4ABN (TRG7ABN))。

此功能只在使用 AD 转换开始请求延迟功能时有效。

(a) 中断减少功能 2 的设定步骤例子

中断减少功能 2 的设定步骤例子如图 21.88 所示。

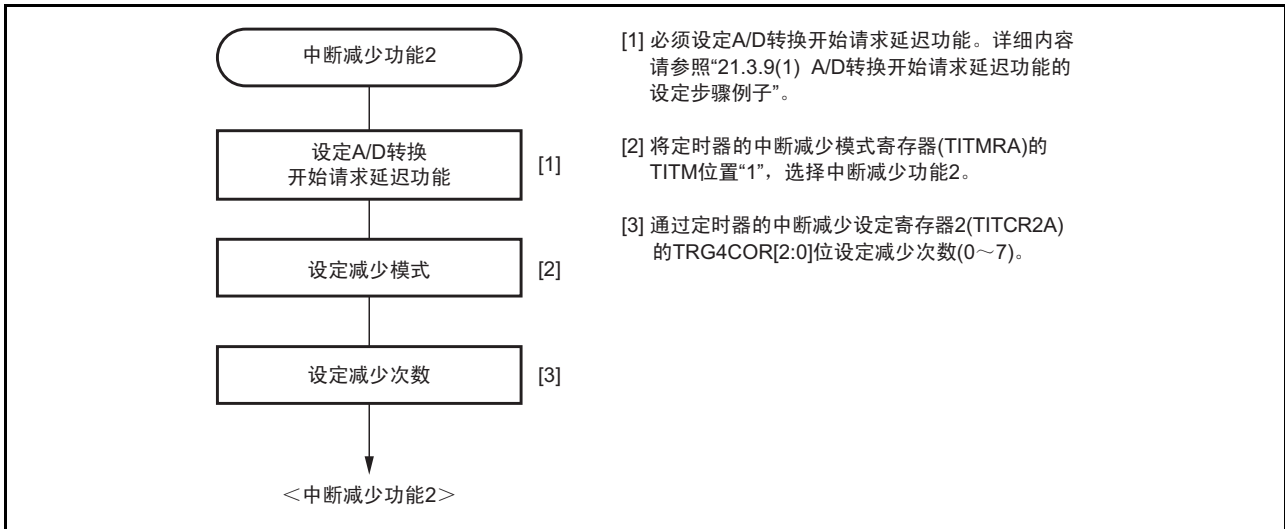


图 21.88 中断减少功能 2 的设定步骤例子

(b) 中断减少功能 2 的运行例子

中断减少功能 2 的运行例子如图 21.89 所示。

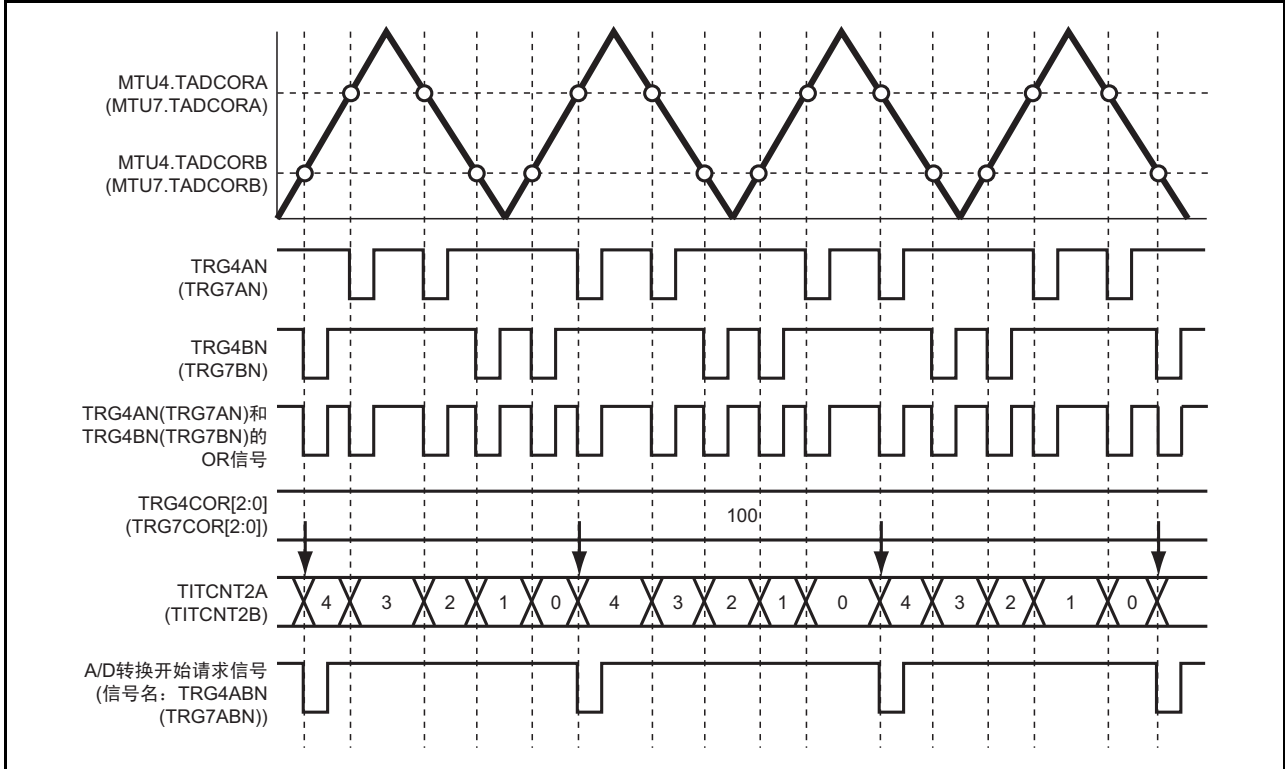


图 21.89 中断减少功能 2 的运行例子 (减少次数为 4 次的情况)

21.3.10 MTU0 ~ 4 – MTU6、7 的同步运行

(1) MTU0 ~ 4 – MTU6、7 的计数器同步启动

能通过设定 TCSYSTR 寄存器，同步启动 MTU0 ~ 4 和 MTU6、7 的计数器。

(a) MTU0 ~ 4 – MTU6、7 的计数器同步启动的设定步骤例子

计数器同步启动的设定步骤例子如图 21.90 所示。

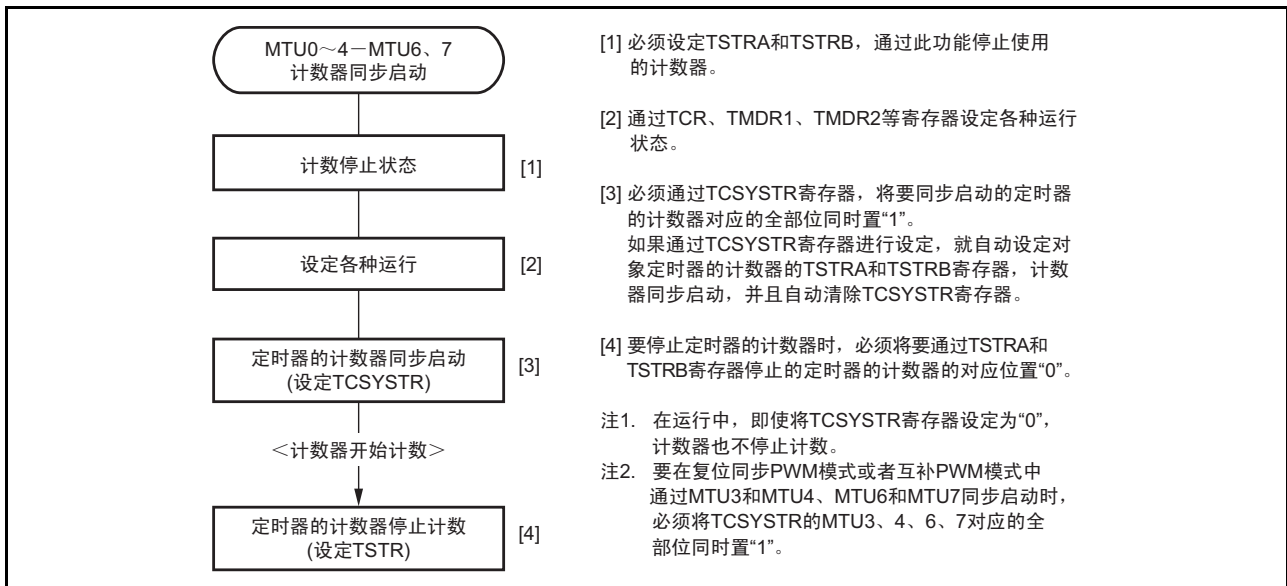


图 21.90 计数器同步启动的设定步骤例子

(b) 计数器同步启动的运行例子

MTU0 ~ 4 和 MTU6、7 的计数器同步启动的运行例子如图 21.95 所示。

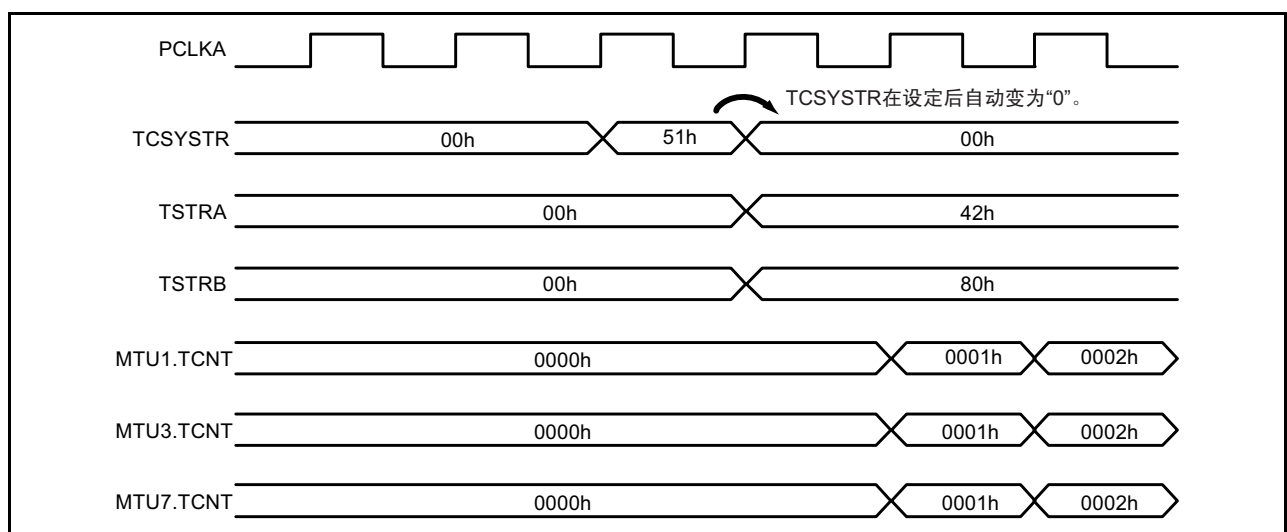


图 21.91 计数器同步启动的运行例子

(2) 使用标志置位源的 MTU6 和 MTU7 的计数器清除 (MTU6 和 MTU7 的同步计数器清除)

MTU6 和 MTU7 能通过设定 TSYCR 寄存器, 使用 MTU0.TSR ~ MTU2.TSR 的标志置位源进行计数器清除。

(a) 使用标志置位源的 MTU6 和 MTU7 的计数器清除的设定步骤例子

使用标志置位源的 MTU6 和 MTU7 的计数器清除的设定步骤例子如图 21.92 所示。

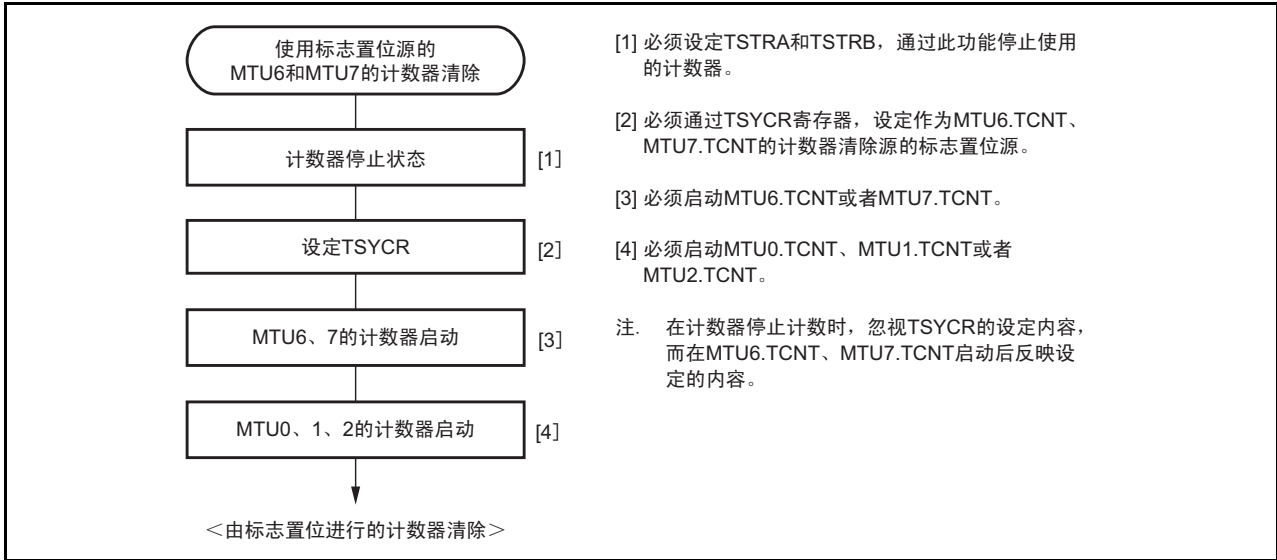


图 21.92 使用标志置位源的 MTU6 和 MTU7 的计数器清除的设定步骤例子

(b) 使用标志置位源的 MTU6 和 MTU7 的计数器清除的运行例子

使用标志置位源的 MTU6 和 MTU7 的计数器清除的运行例子如图 21.93 和图 21.94 所示。

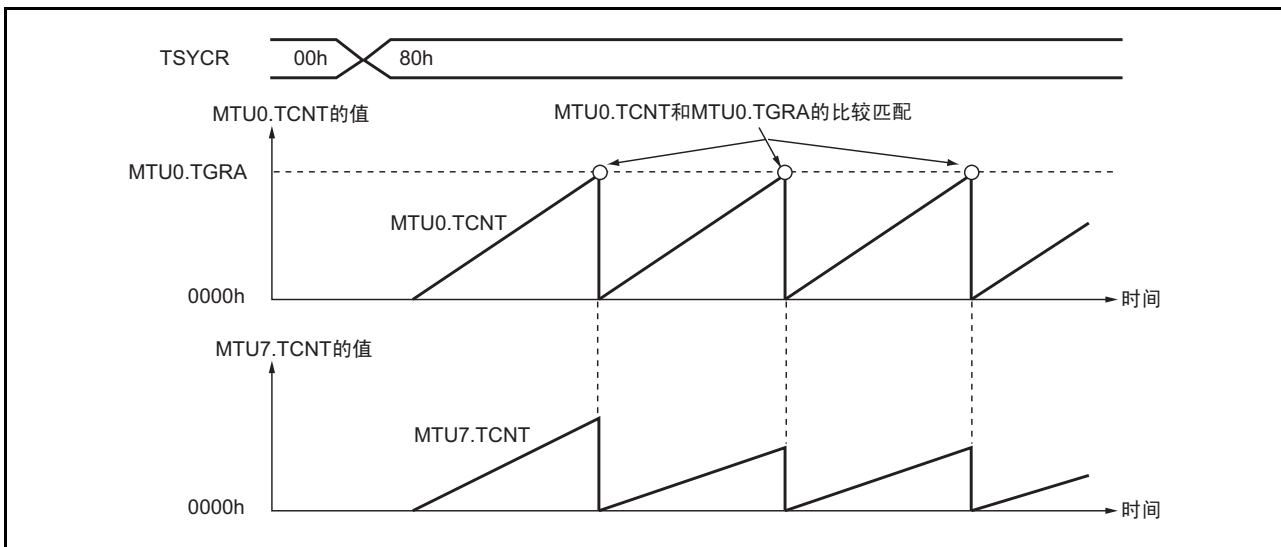


图 21.93 使用标志置位源的 MTU6 和 MTU7 的计数器清除的运行例子 (1)

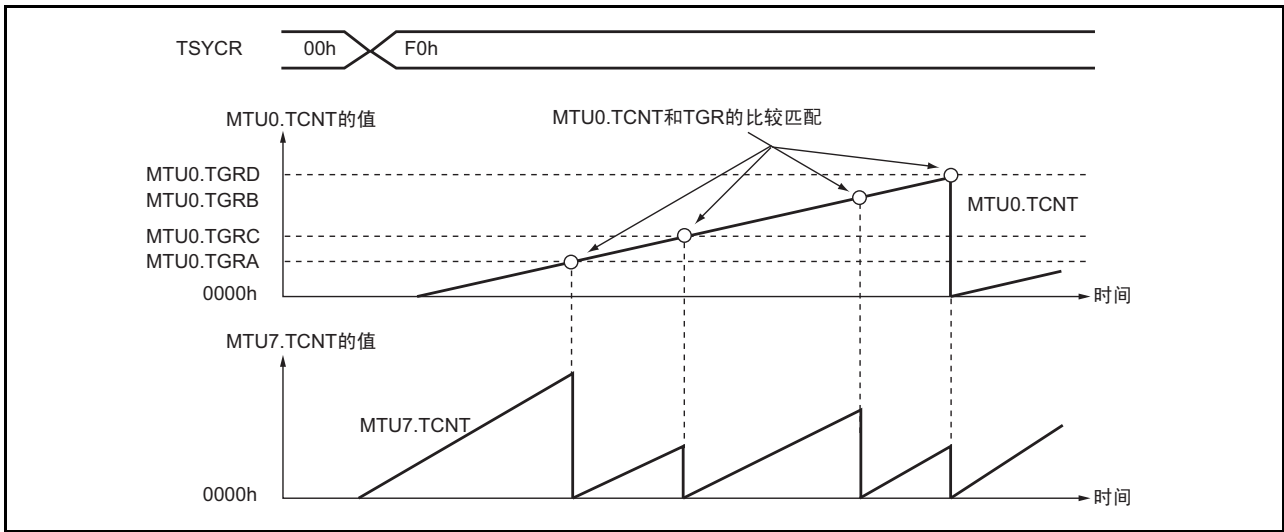


图 21.94 使用标志置位源的 MTU6 和 MTU7 的计数器清除的运行例子 (2)

21.3.11 外部脉宽的测量功能

MTU5 最多能测量 3 个外部脉宽。

(1) 测量外部脉宽的设定步骤例子

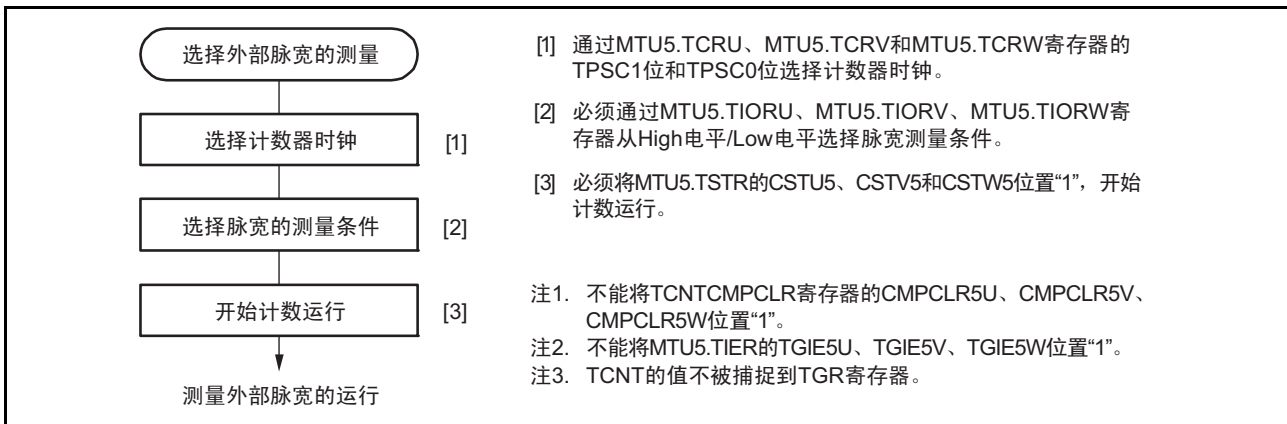


图 21.95 测量外部脉宽的设定步骤例子

(2) 测量外部脉宽的运行例子

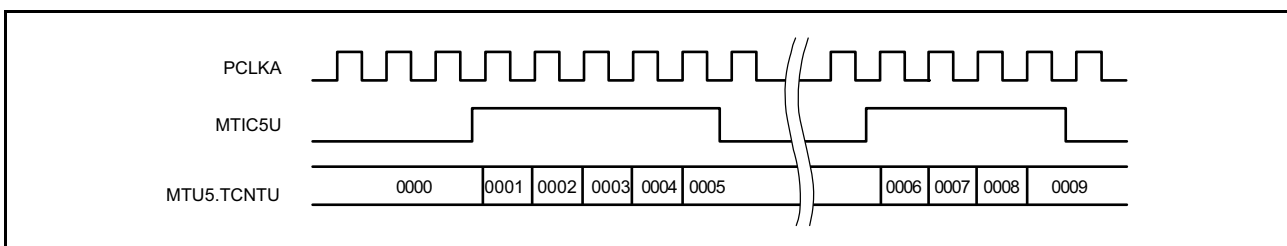


图 21.96 测量外部脉宽的运行例子 (测量 High 电平的脉宽)

21.3.12 死区时间的补偿功能

能通过测量输出波形的延迟并且将此延迟反映到占空比，将外部脉宽测量功能用作互补 PWM 运行时的 PWM 输出波形的死区时间补偿功能。

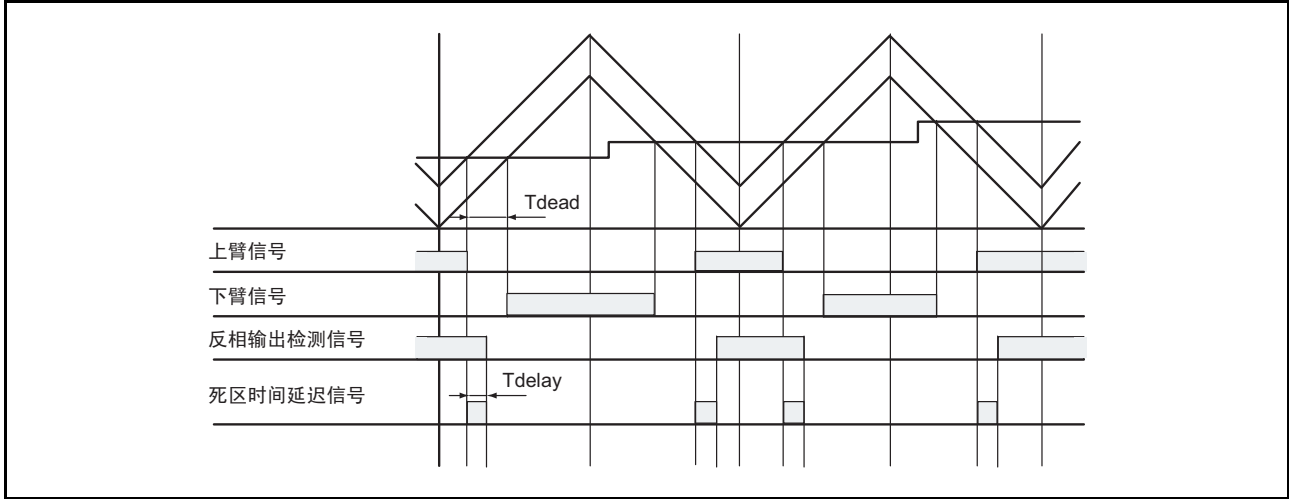


图 21.97 互补 PWM 运行时的死区时间延迟

(1) 死区时间补偿功能的设定步骤例子

使用 MTU5 的 3 个计数器的死区时间补偿功能的设定步骤例子如图 21.98 所示。

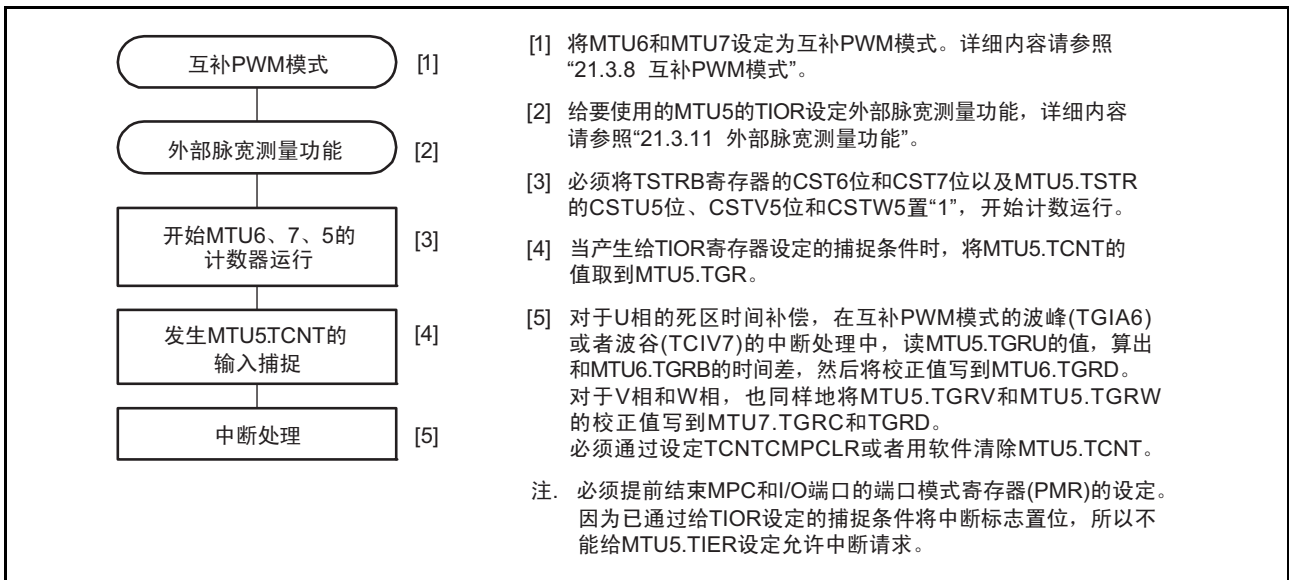


图 21.98 死区时间补偿功能的设定步骤例子

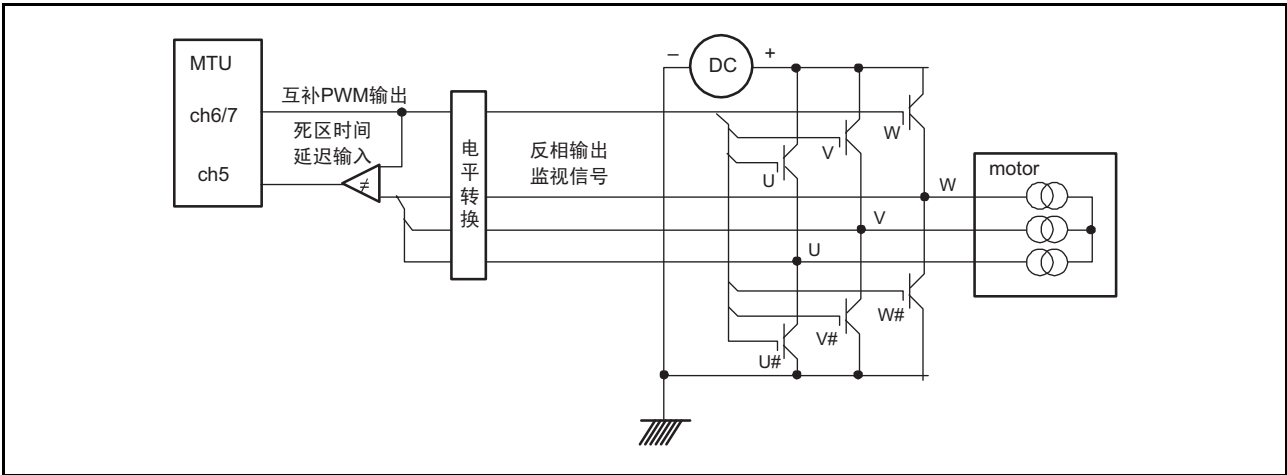


图 21.99 马达控制电路的结构例子

21.3.13 在互补 PWM 的“波峰 / 波谷”进行的 TCNT 捕捉运行

在互补 PWM 运行时，在“波峰、波谷、波峰和波谷”将 TCNT 的值保存到 TGR，通过 TIOR 选择要保存到 TGR 的时序转换。

TCNT 用作自由运行计数器（不被清除）并且在设定的“波峰、波谷”对 TGR 进行捕捉的运行例子如图 21.100 所示。

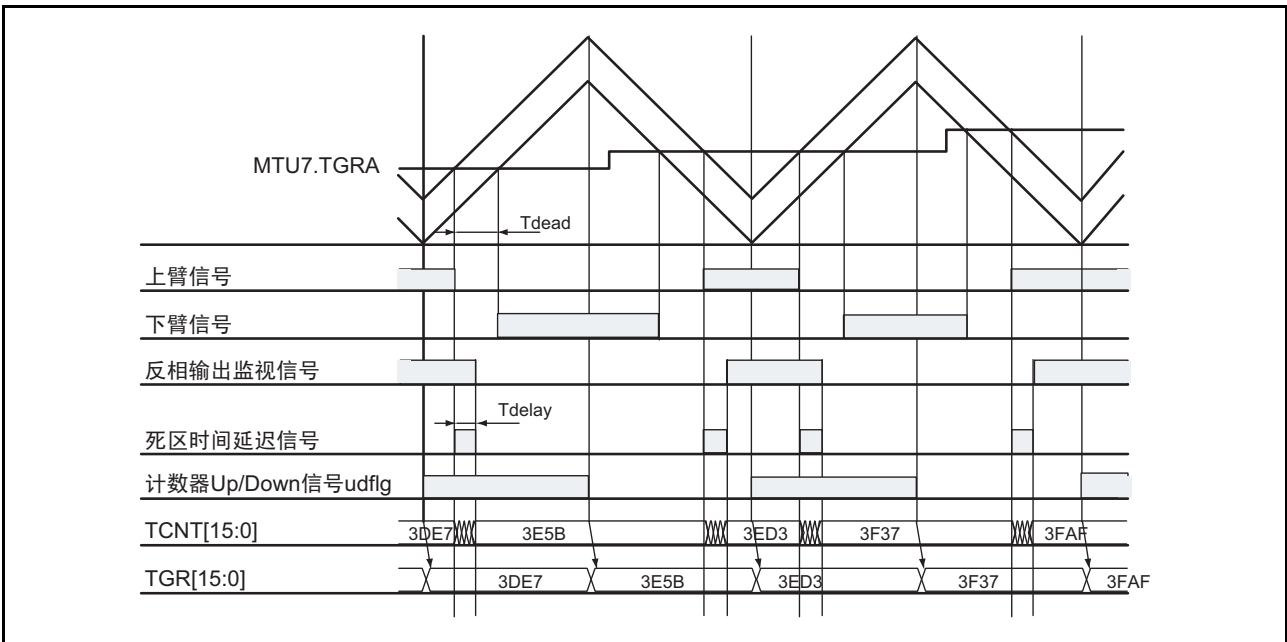


图 21.100 在互补 PWM 的“波峰 / 波谷”进行的 TCNT 捕捉运行

21.4 中断源

21.4.1 中断源和优先级

MTU 的中断源有 TGR 的输入捕捉 / 比较匹配、TCNT 的上溢和下溢共 3 种。因为各中断源有各自专用的状态标志和允许 / 禁止位，所以能独立允许或者禁止中断请求信号的产生。

如果产生中断源，TSR 寄存器的对应状态标志就变为“1”。此时，如果 TIER 寄存器的对应允许 / 禁止位为“1”，就请求中断。但是，在该状态标志为“1”的状态下忽视中断请求，因此为了能够再次发生中断，必须将该状态标志置“0”。

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“14. 中断控制器 (ICUb)”。MTU 的中断源一览表如表 21.72 所示。

表 21.72 MTU 中断源

通道	名称	中断源	DTC/DMAC 的 启动	优先级		
MTU0	TGIA0	MTU0.TGRA 的输入捕捉 / 比较匹配	能	高 ↑		
	TGIB0	MTU0.TGRB 的输入捕捉 / 比较匹配	能			
	TGIC0	MTU0.TGRC 的输入捕捉 / 比较匹配	能			
	TGID0	MTU0.TGRD 的输入捕捉 / 比较匹配	能			
	TCIV0	MTU0.TCNT 的上溢	不能			
	TGIE0	MTU0.TGRE 的比较匹配	不能			
	TGIF0	MTU0.TGRF 的比较匹配	不能			
MTU1	TGIA1	MTU1.TGRA 的输入捕捉 / 比较匹配	能	↑		
	TGIB1	MTU1.TGRB 的输入捕捉 / 比较匹配	能			
	TCIV1	MTU1.TCNT 的上溢	不能			
	TCIU1	MTU1.TCNT 的下溢	不能			
MTU2	TGIA2	MTU2.TGRA 的输入捕捉 / 比较匹配	能		↑	
	TGIB2	MTU2.TGRB 的输入捕捉 / 比较匹配	能			
	TCIV2	MTU2.TCNT 的上溢	不能			
	TCIU2	MTU2.TCNT 的下溢	不能			
MTU3	TGIA3	MTU3.TGRA 的输入捕捉 / 比较匹配	能			↑
	TGIB3	MTU3.TGRB 的输入捕捉 / 比较匹配	能			
	TGIC3	MTU3.TGRC 的输入捕捉 / 比较匹配	能			
	TGID3	MTU3.TGRD 的输入捕捉 / 比较匹配	能			
	TCIV3	MTU3.TCNT 的上溢	不能			
MTU4	TGIA4	MTU4.TGRA 的输入捕捉 / 比较匹配	能	↑		
	TGIB4	MTU4.TGRB 的输入捕捉 / 比较匹配	能			
	TGIC4	MTU4.TGRC 的输入捕捉 / 比较匹配	能			
	TGID4	MTU4.TGRD 的输入捕捉 / 比较匹配	能			
	TCIV4	MTU4.TCNT 的上溢 / 下溢 (只在互补 PWM 模式中)	能			
MTU5	TGIU5	MTU5.TGRU 的输入捕捉 / 比较匹配	能		↑	
	TGIV5	MTU5.TGRV 的输入捕捉 / 比较匹配	能			
	TGIW5	MTU5.TGRW 的输入捕捉 / 比较匹配	能			
MTU6	TGIA6	MTU6.TGRA 的输入捕捉 / 比较匹配	能			↑
	TGIB6	MTU6.TGRB 的输入捕捉 / 比较匹配	能			
	TGIC6	MTU6.TGRC 的输入捕捉 / 比较匹配	能			
	TGID6	MTU6.TGRD 的输入捕捉 / 比较匹配	能			
	TCIV6	MTU6.TCNT 的上溢	不能			
MTU7	TGIA7	MTU7.TGRA 的输入捕捉 / 比较匹配	能	低		
	TGIB7	MTU7.TGRB 的输入捕捉 / 比较匹配	能			
	TGIC7	MTU7.TGRC 的输入捕捉 / 比较匹配	能			
	TGID7	MTU7.TGRD 的输入捕捉 / 比较匹配	能			
	TCIV7	MTU7.TCNT 的上溢 / 下溢 (只在互补 PWM 模式中)	能			

注. 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

在各通道的 TGR 发生输入捕捉 / 比较匹配时将 TSR.TGF 标志置“1”的情况下，如果 TIER.TGIE 位为“1”，就请求中断。通过将 TGF 标志清“0”，解除中断请求。在 MTU 中，MTU0 有 6 个，MTU3、4、6、7 各有 4 个，MTU1 和 MTU2 各有 2 个，MTU5 有 3 个，共计 29 个输入捕捉 / 比较匹配中断。在输入捕捉时 MTU0.TGFE 标志和 MTU0.TGFF 标志不能变为“1”。

(2) 上溢中断

在各通道的 TCNT 发生上溢时将 TSR.TCFV 标志置“1”的情况下，如果 TIER.TCIEV 位为“1”，就请求中断。通过将 TCFV 标志清“0”，解除中断请求。在 MTU 中，各通道各有 1 个，共计 7 个上溢中断。

另外，在互补 PWM 中运行时，如果 MTU4.TCNT、MTU7.TCNT 发生下溢，TCFV 标志就变为“1”。

(3) 下溢中断

在各通道的 TCNT 发生下溢时将 TSR.TCFU 标志置“1”的情况下，如果 TIER.TCIEU 位为“1”，就请求中断。通过将 TCFU 标志清“0”，解除中断请求。在 MTU 中，MTU1 和 MTU2 各有 1 个，共计 2 个下溢中断。

21.4.2 DTC 的启动

(1) DTC 的启动

能通过各通道 TGR 的输入捕捉 / 比较匹配中断或者 MTU4 和 MTU7 的上溢中断来启动 DTC，详细内容请参照“18. 数据传送控制器 (DTCa)”。

在 MTU 中，MTU0、3、6 各有 4 个，MTU1 和 MTU2 各有 2 个，MTU4 和 MTU7 各有 5 个，MTU5 有 3 个，共计 29 个输入捕捉 / 比较匹配中断和上溢中断作为 DTC 的启动源。

但是，与中断处理相同，在对应状态标志为“1”的状态下忽视 DTC 请求，因此为了能够再次启动 DTC，必须将对应状态标志置“0”。

21.4.3 A/D 转换器的启动

在 MTU 中，能通过以下 3 种方法启动 A/D 转换器。

各中断源和 A/D 转换开始请求的对应如表 21.73 所示。

(1) 通过 TGRA 的输入捕捉 / 比较匹配和互补 PWM 模式的 MTU4.TCNT (MTU7.TCNT) 波谷启动 A/D 转换器

能通过各通道 TGRA 的输入捕捉 / 比较匹配来启动 A/D 转换器。另外，如果在将 MTU4.TIER (MTU7.TIER) 的 TTGE2 位置“1”的状态下进行互补 PWM 运行，就能在 MTU4.TCNT (MTU7.TCNT) 的波谷 (MTU4.TCNT (MTU7.TCNT) = 0000h) 启动 A/D 转换器。

在以下所示的条件下，对 A/D 转换器产生 A/D 转换开始请求 TRGAnN (n=MTU0 ~ 4、6、7)。

- 在发生各通道的 TGRA 输入捕捉 / 比较匹配的情况下 TSR.TGFA 标志和 TIER.TTGE 位为“1”时
- 在 MTU4.TIER (MTU7.TIER) 的 TTGE2 位为“1”的状态下进行互补 PWM 运行，并且 MTU4.TCNT (MTU7.TCNT) 变为波谷时 (MTU4.TCNT (MTU7.TCNT) = 0000h)

在上述条件下，如果已在 A/D 转换器侧选择 MTU 的转换开始触发 TRGAnN，就开始进行 A/D 转换。

(2) 通过 MTU0.TCNT 和 MTU0.TGRE 的比较匹配启动 A/D 转换器

能通过 MTU0.TCNT 和 MTU0.TGRE 的比较匹配，产生 A/D 转换开始请求 TRG0N 并且启动 A/D 转换器。

在发生 MTU0.TCNT 和 MTU0.TGRE 的比较匹配时将 MTU0.TSR2 的 TGFE 标志置“1”的状态下，如果 MTU0.TIER2 的 TTGE2 位为“1”，就对 A/D 转换器产生 A/D 转换开始请求 TRG0N。此时，如果已在 A/D 转换器侧选择 MTU 的转换开始触发 TRG0N，就开始进行 A/D 转换。

(3) 通过 A/D 转换开始请求的延迟功能启动 A/D 转换器

如果给 A/D 转换开始请求控制寄存器 (MTU4.TADCR (MTU7.TADCR)) 的 UT4AE、DT4AE、UT4BE、DT4BE (UT7AE、DT7AE、UT7BE、DT7BE) 位写“1”，就能在 MTU4.TADCORA、MTU4.TADCORB 和 MTU4.TCNT (MTU7.TADCORA、MTU7.TADCORB 和 MTU7.TCNT) 相同时产生 TRG4AN、TRG4BN (TRG7AN、TRG7BN)，并且启动 A/D 转换器。另外，能通过产生 TRG4AN (TRG7AN) 或者 TRG4BN (TRG7BN) 输出 TRG4ABN (TRG7ABN)，启动 A/D 转换器。详细内容请参照“21.3.9 A/D 转换开始请求的延迟功能”。

如果产生 TRG4AN (TRG7AN) 并且已在 A/D 转换器侧选择 MTU 的转换开始触发 TRG4AN (TRG7AN)，或者产生 TRG4BN (TRG7BN) 并且已在 A/D 转换器侧选择 MTU 的转换开始触发 TRG4ABN (TRG7ABN) 时，就开始进行 A/D 转换。

表 21.73 各中断源和 A/D 转换开始请求的对应

对象	AD 启动源	A/D 转换开始请求
MTU0.TGRA 和 MTU0.TCNT	输入捕捉 / 比较匹配	TRGA0N
MTU1.TGRA 和 MTU1.TCNT		TRGA1N
MTU2.TGRA 和 MTU2.TCNT		TRGA2N
MTU3.TGRA 和 MTU3.TCNT		TRGA3N
MTU4.TGRA 和 MTU4.TCNT (注 1)		TRGA4N
MTU4.TCNT	互补 PWM 模式中的 MTU4.TCNT 的波谷	
MTU6.TGRA 和 MTU6.TCNT	输入捕捉 / 比较匹配	TRGA6N
MTU7.TGRA 和 MTU7.TCNT (注 1)		TRGA7N
MTU7.TCNT		互补 PWM 模式中的 MTU7.TCNT 的波谷
MTU0.TGRE 和 MTU0.TCNT	比较匹配	TRG0N
MTU4.TADCORA 和 MTU4.TCNT	比较匹配	TRG4AN
MTU4.TADCORB 和 MTU4.TCNT		TRG4BN
MTU7.TADCORA 和 MTU7.TCNT		TRG7AN
MTU7.TADCORB 和 MTU7.TCNT		TRG7BN
MTU4.TADCORA 和 MTU4.TCNT、 MTU4.TADCORB 和 MTU4.TCNT	比较匹配 (中断减少功能 2)	TRG4ABN
MTU7.TADCORA 和 MTU7.TCNT、 MTU7.TADCORB 和 MTU7.TCNT		TRG7ABN

注 1. 在互补 PWM 模式中，为了生成 PWM 波形，MTU4.TGRA (MTU7.TGRA) 不仅对 MTU4.TCNT (MTU7.TCNT) 而且对 MTU3.TCNT (MTU6.TCNT) 和 TCNTSA (TCNTSB) 进行比较匹配的检测。因此，在发生和 MTU3.TCNT (MTU6.TCNT) 或者 TCNTSA (TCNTSB) 的比较匹配时，也产生 TRGA4N (TRGA7N)。

如果使 MTU3 和 MTU4 (MTU6 和 MTU7) 在互补 PWM 模式中运行并且产生 A/D 转换开始请求，就必须使用由 MTU4.TCNT (MTU7.TCNT) 和 MTU4.TADCORA/B (MTU7.TADCORA/B) 的比较匹配而产生的 A/D 转换开始请求。

21.5 运行时序

21.5.1 输入 / 输出时序

(1) TCNT 的计数时序

内部时钟运行时的 TCNT 计数时序如图 21.101 和图 21.102 所示，外部时钟运行（正常模式）时的 TCNT 计数时序如图 21.103，外部时钟运行（相位计数模式）的 TCNT 计数时序如图 21.104 所示。

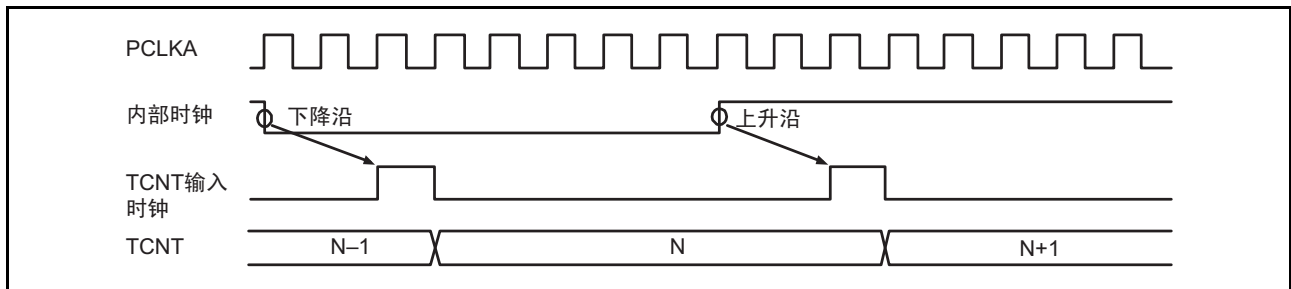


图 21.101 内部时钟运行时的计数时序 (MTU0 ~ 4、6、7)

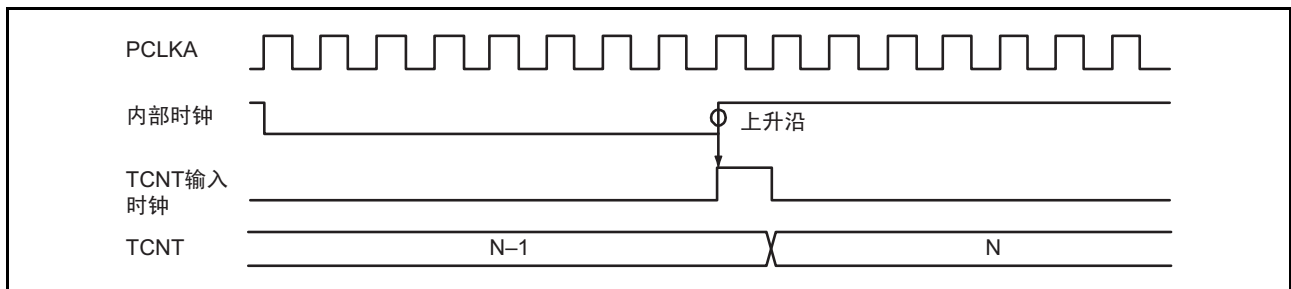


图 21.102 内部时钟运行时的计数时序 (MTU5)

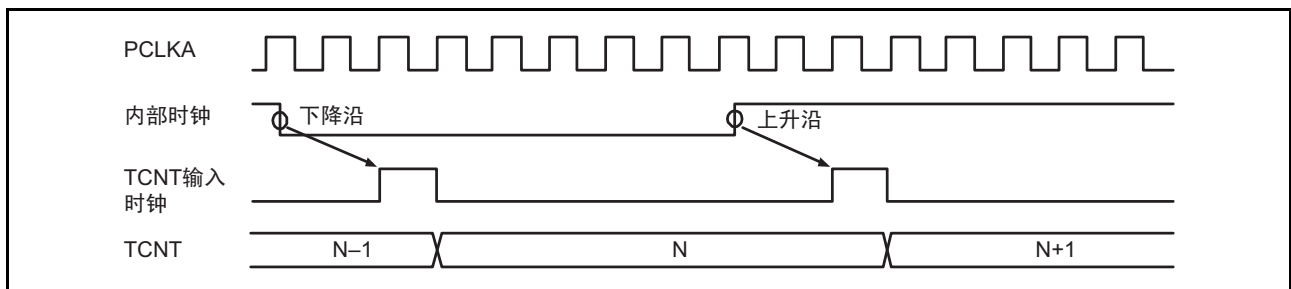


图 21.103 外部时钟运行时的计数时序 (MTU0 ~ 4)

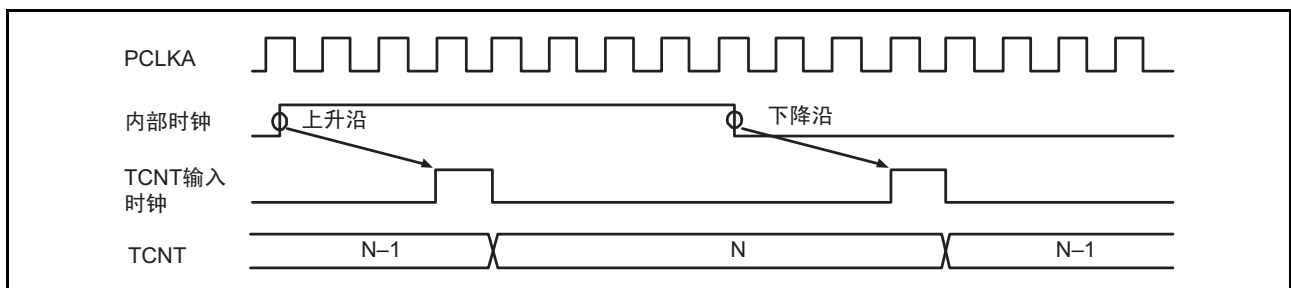


图 21.104 外部时钟运行时的计数时序 (相位计数模式)

(2) 输出比较的输出时序

在 TCNT 和 TGR 的值相同的最后状态（在更新 TCNT 相同后的计数值时）产生比较匹配信号。当产生比较匹配信号时，将 TIOR 设定的输出值输出到输出比较的输出引脚（MTIOC 引脚）。在 TCNT 和 TGR 的值相同后到产生 TCNT 输入时钟前，不产生比较匹配信号。

输出比较的输出时序（正常模式、PWM 模式）如图 21.105、输出比较的输出时序（互补 PWM 模式、复位同步 PWM 模式）如图 21.106 所示。

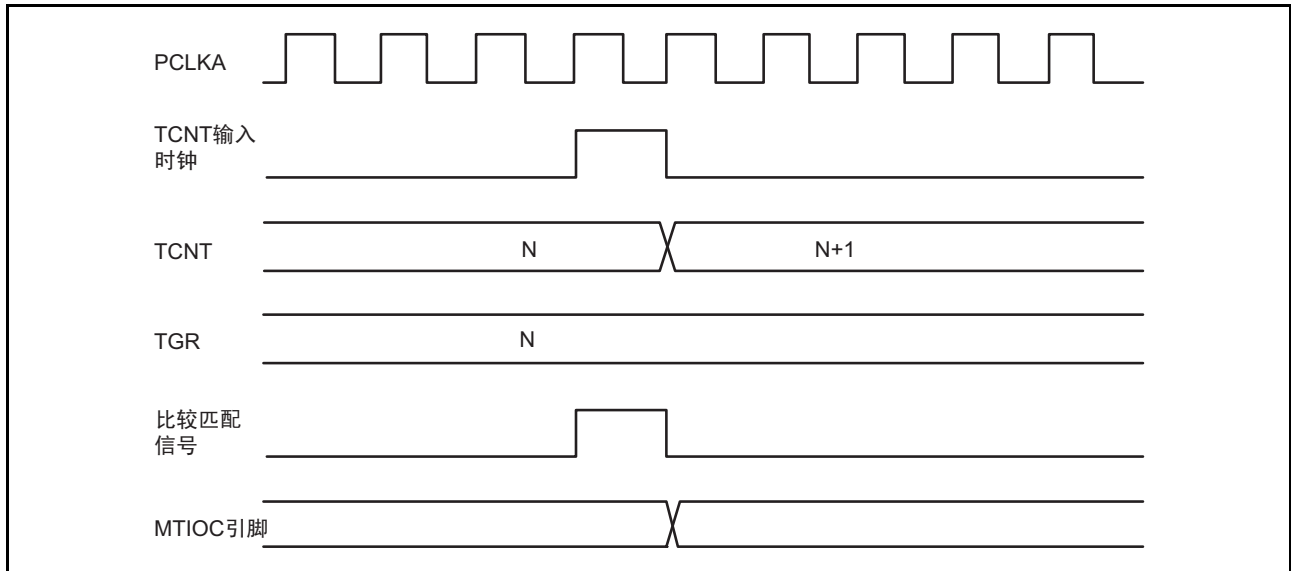


图 21.105 输出比较的输出时序（正常模式、PWM 模式）

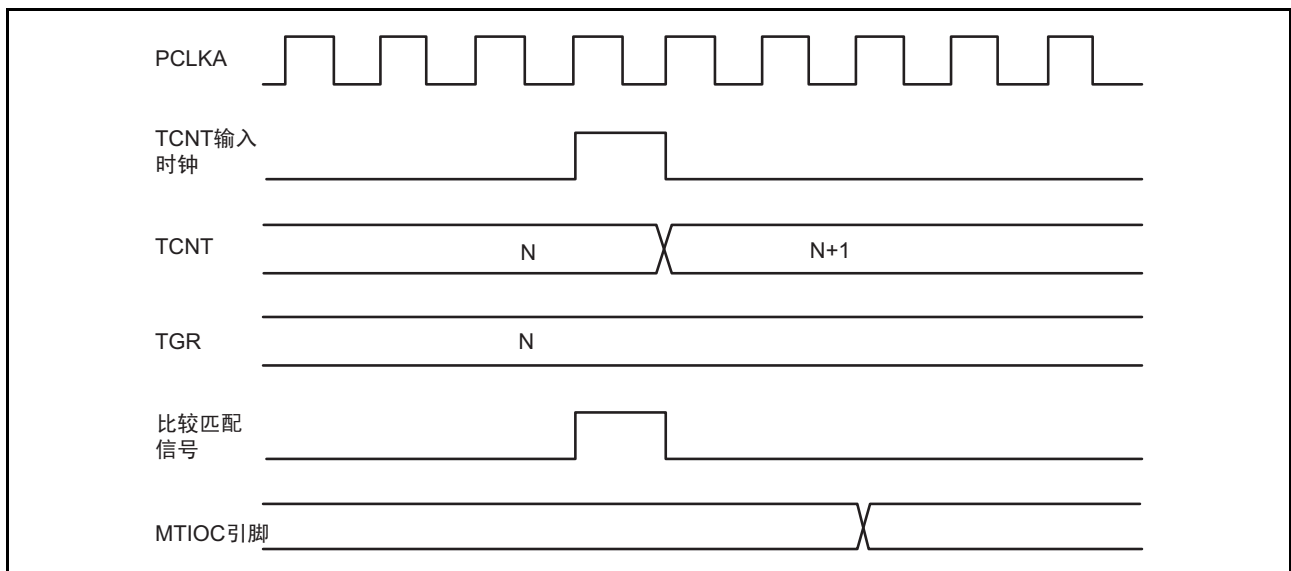


图 21.106 输出比较的输出时序（互补 PWM 模式、复位同步 PWM 模式）

(3) 输入捕捉信号的时序

输入捕捉的时序如图 21.107 所示。

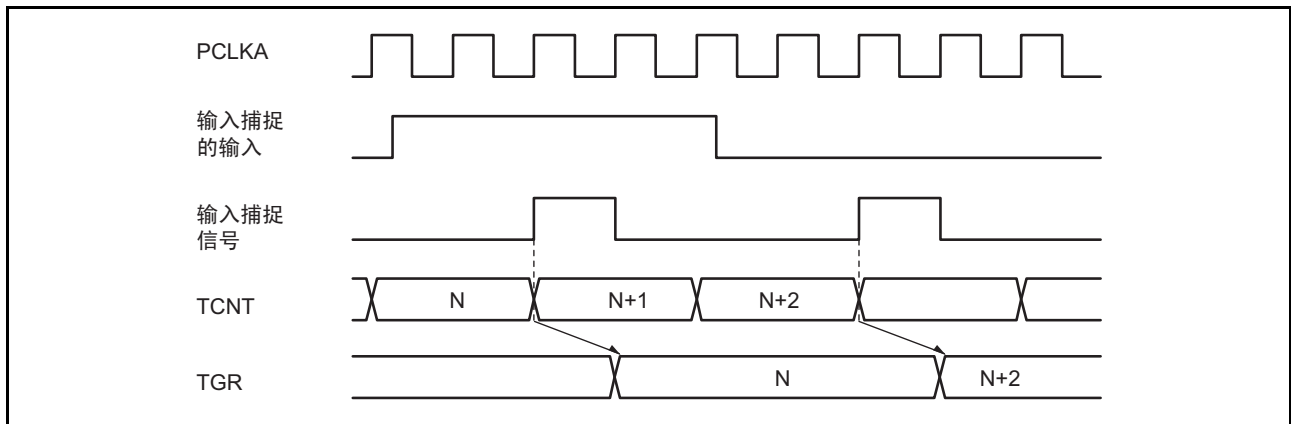


图 21.107 输入捕捉的输入信号时序

(4) 通过比较匹配 / 输入捕捉进行的计数器清除时序

指定通过发生比较匹配进行计数器清除时的时序如图 21.108 和图 21.109 所示，指定通过发生输入捕捉进行计数器清除时的时序如图 21.110 所示。

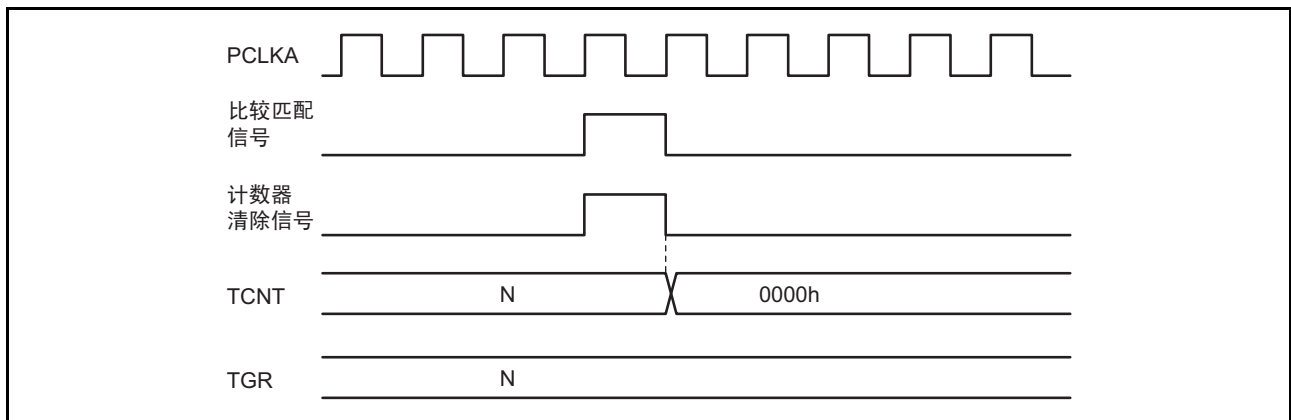


图 21.108 计数器清除时序 (比较匹配) (MTU0 ~ 4、6、7)

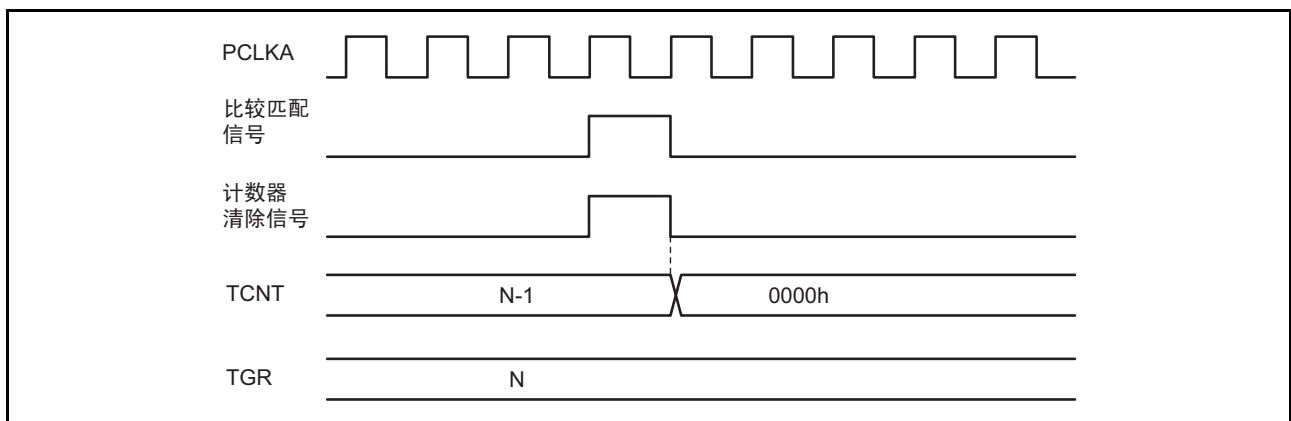


图 21.109 计数器清除时序 (比较匹配) (MTU5)

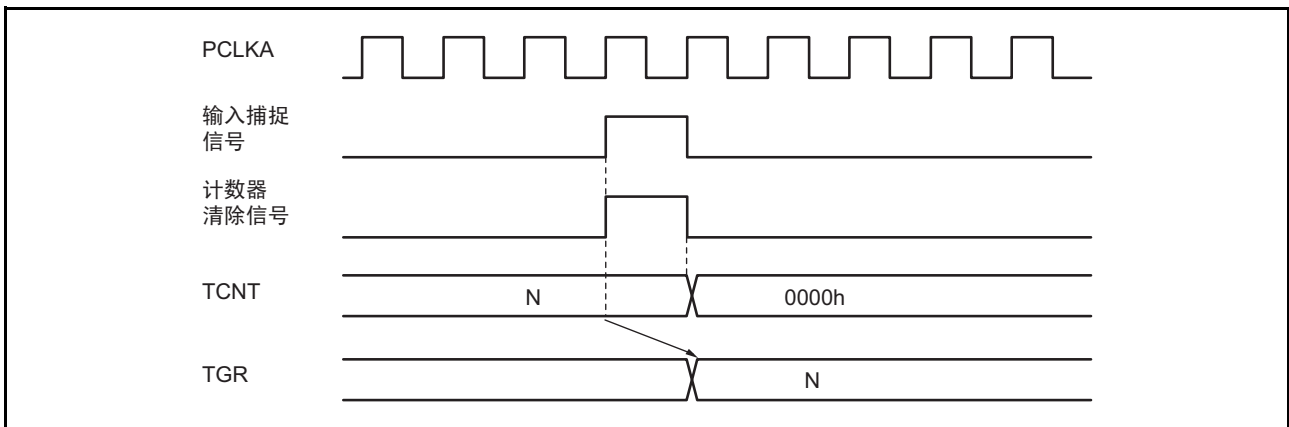


图 21.110 计数器清除时序 (输入捕捉) (MTU0 ~ 7)

(5) 缓冲运行的时序

缓冲运行的时序如图 21.111 ~ 图 21.113 所示。

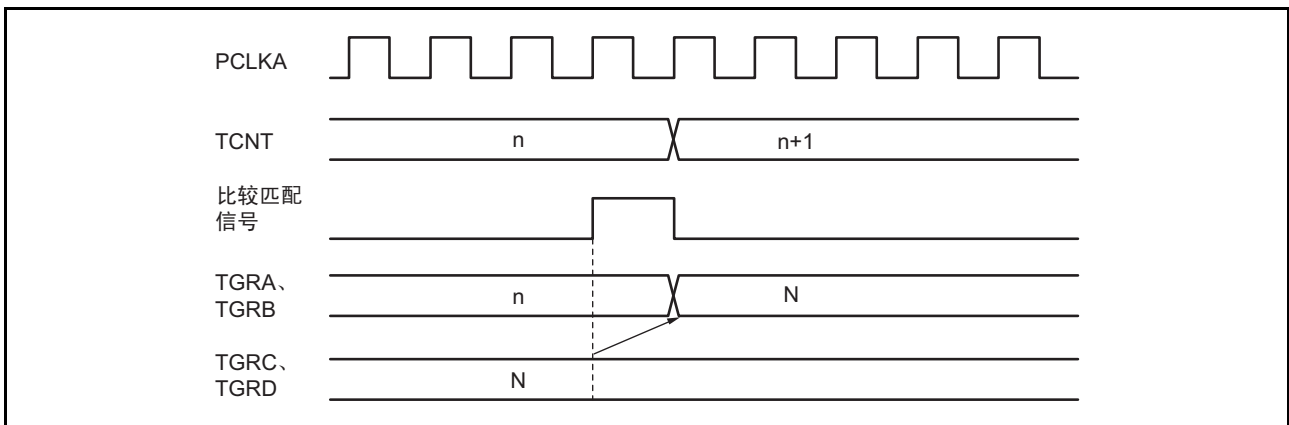


图 21.111 缓冲运行的时序 (比较匹配)

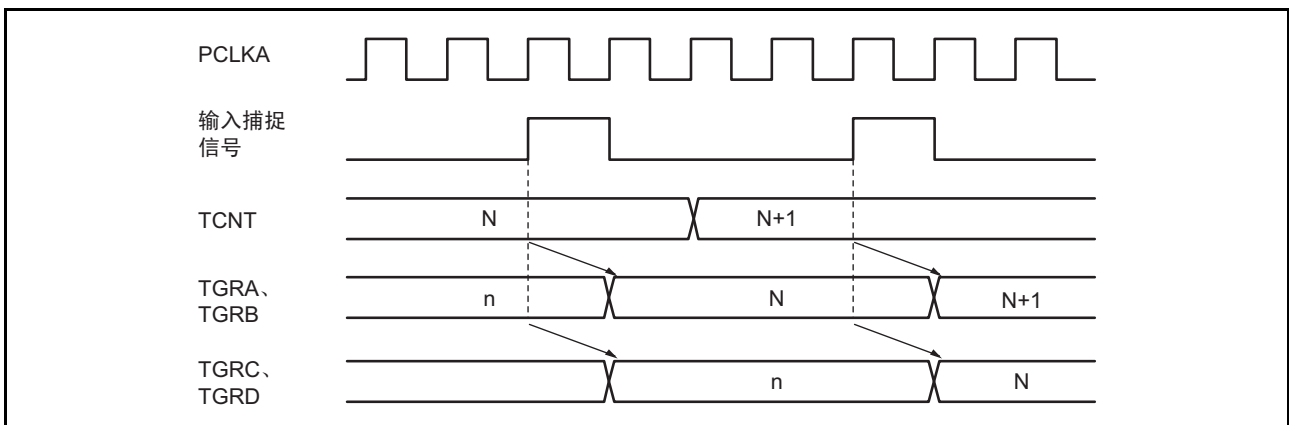


图 21.112 缓冲运行的时序 (输入捕捉)

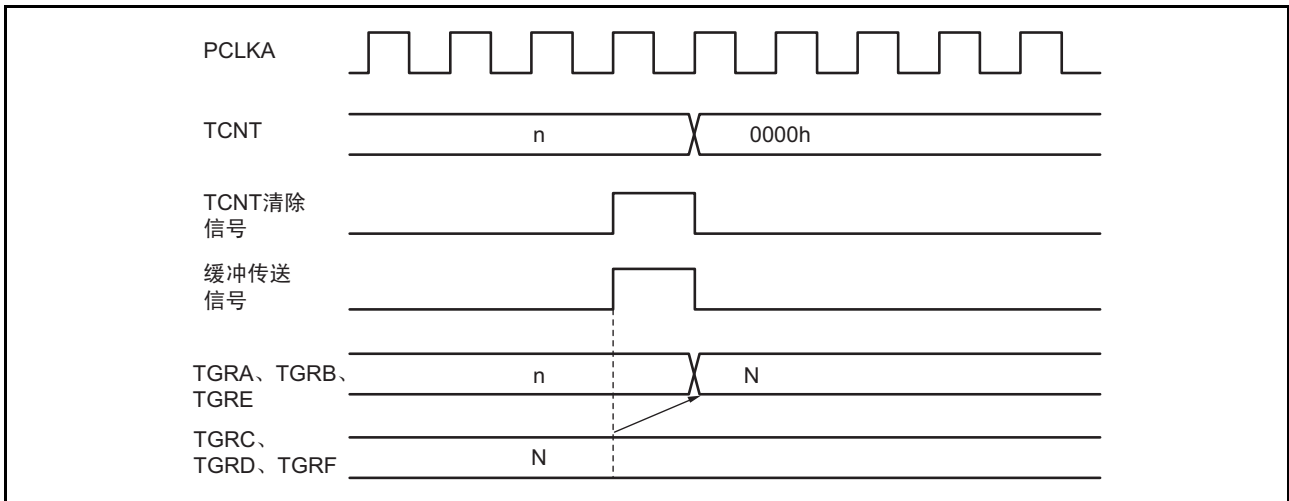


图 21.113 缓冲运行的时序 (清除 TCNT 的情况)

(6) 缓冲传送的时序 (互补 PWM 模式)

互补 PWM 模式的缓冲传送时序如图 21.114 ~ 图 21.116 所示。

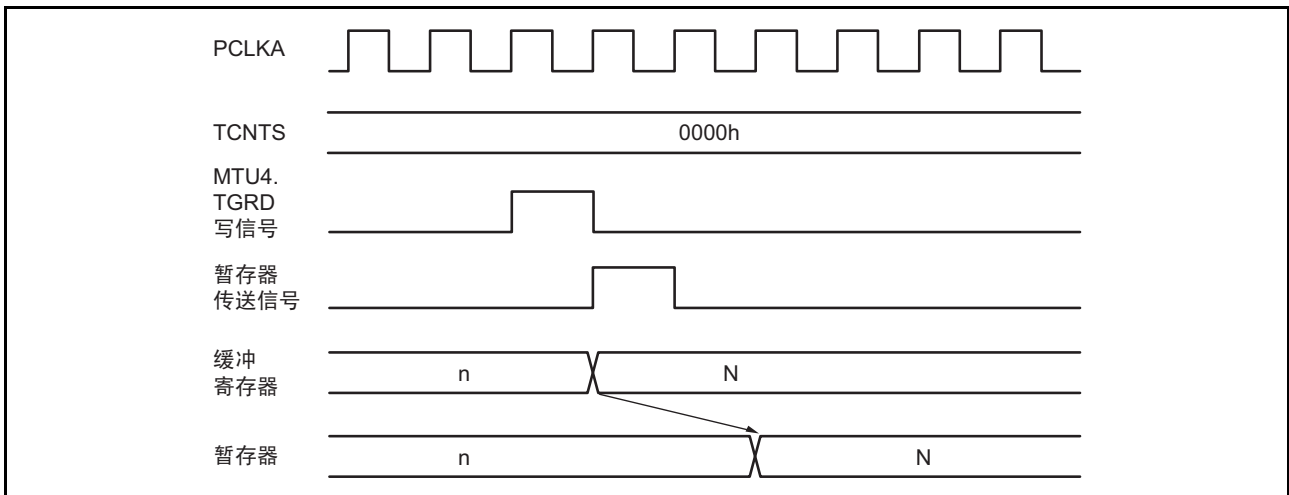


图 21.114 缓冲寄存器到暂存器的传送时序 (TCNTSA 停止)

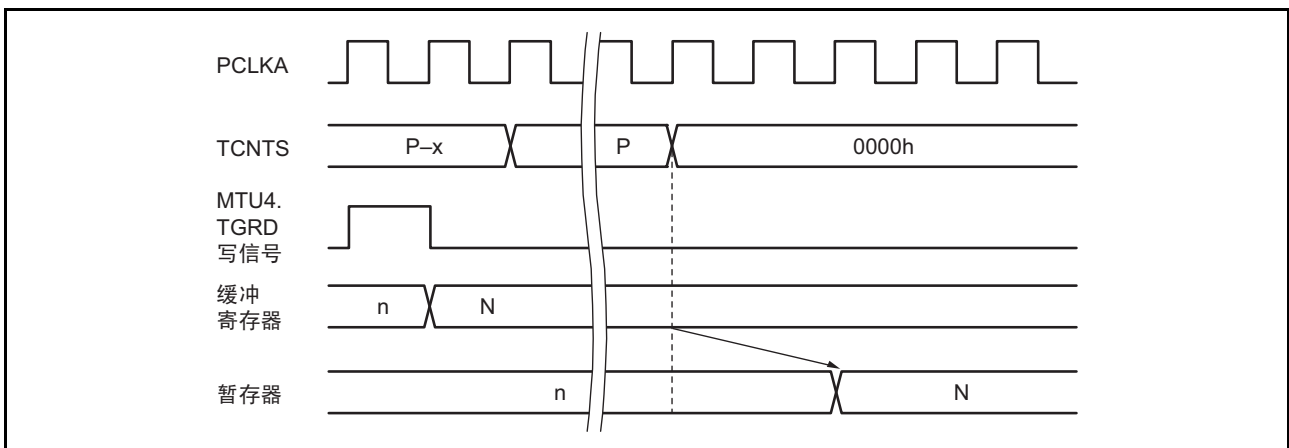


图 21.115 缓冲寄存器到暂存器的传送时序 (TCNTSA 正在运行)

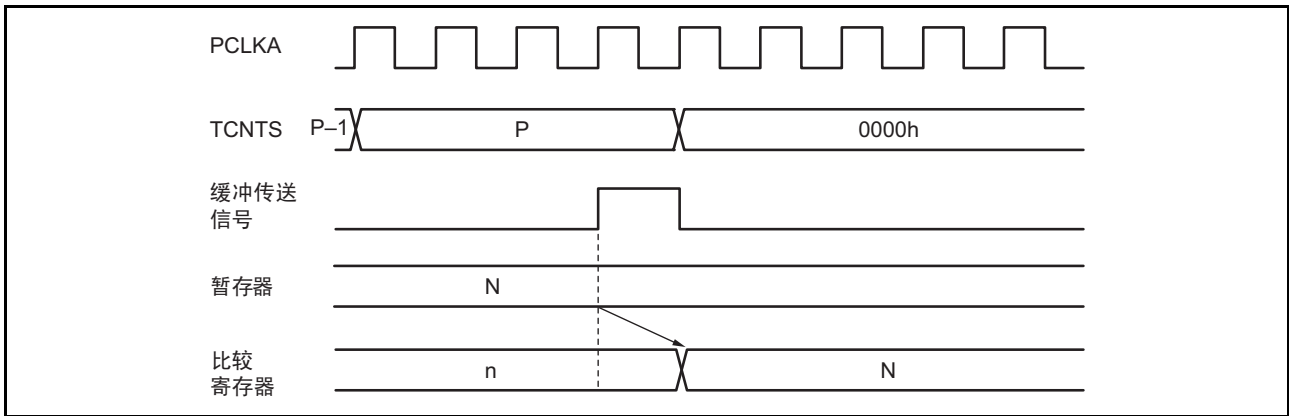


图 21.116 暂存器到比较寄存器的传送时序

21.5.2 中断信号的时序

(1) 比较匹配时的 TGF 标志的置位时序

由发生比较匹配产生的 TSR 寄存器的 TGF 标志的置位时序和 TGI 中断请求信号的时序分别如图 21.117 和图 21.118 所示。

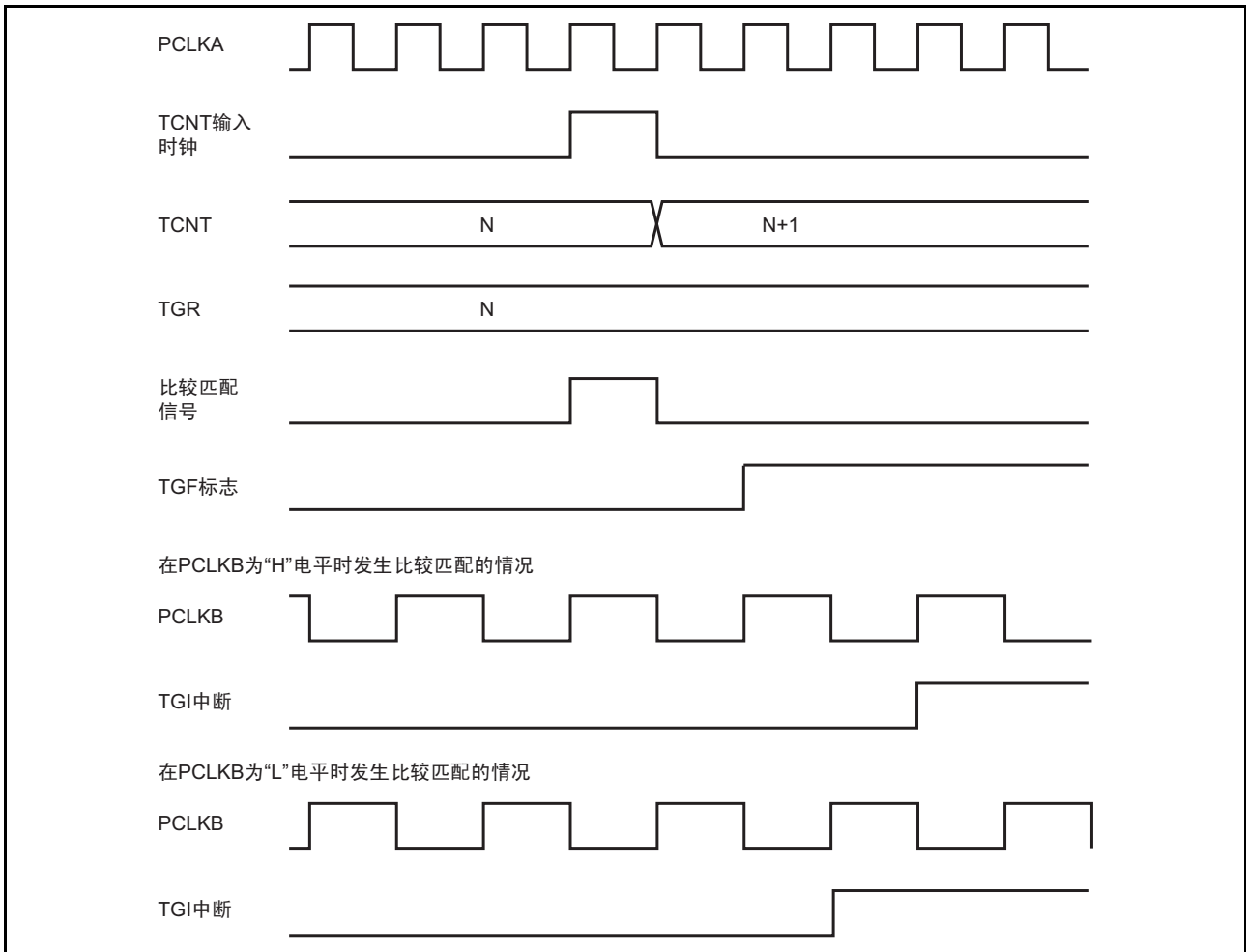


图 21.117 TGI 中断的时序 (比较匹配) (MTU0 ~ 4、6、7)

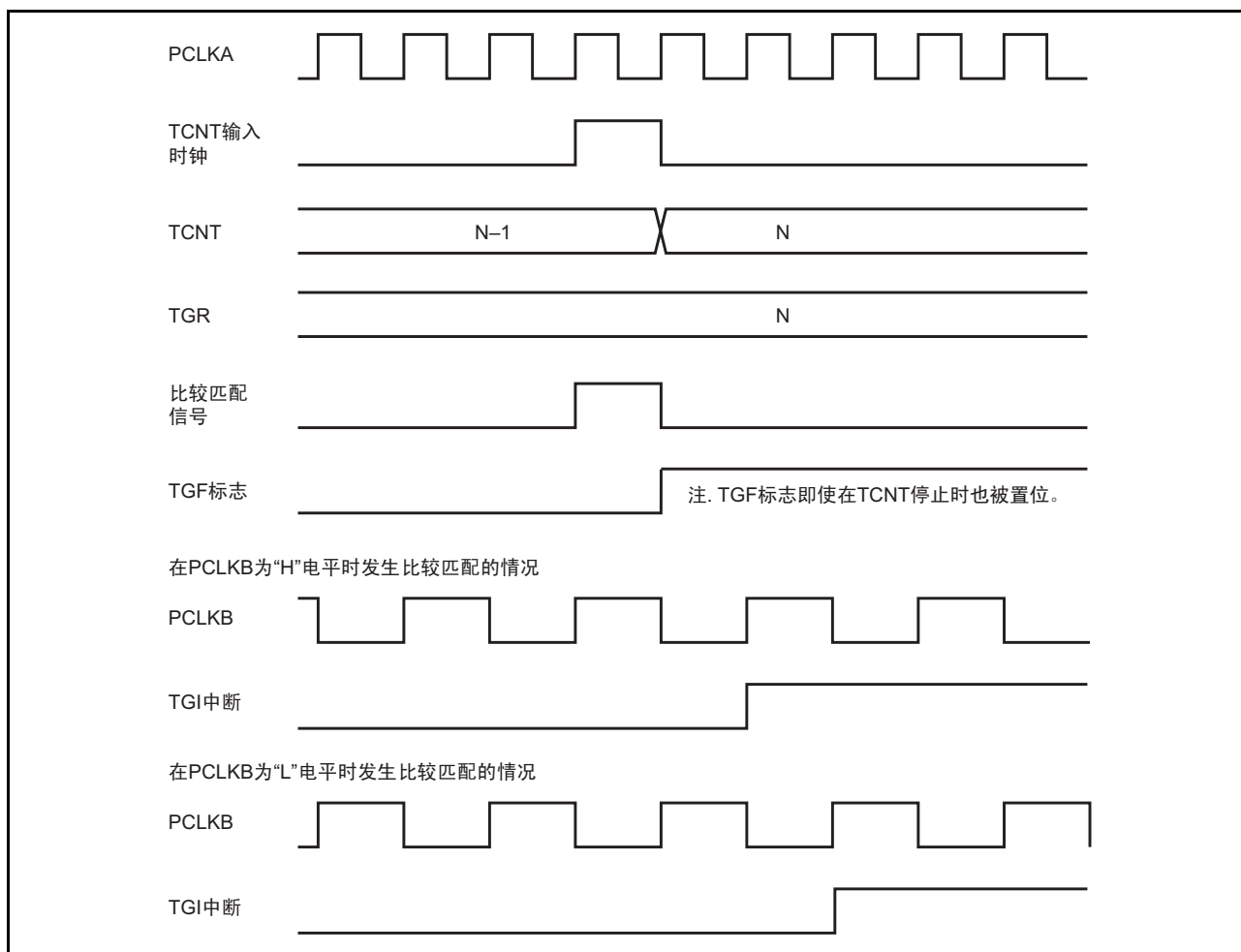


图 21.118 TGI 中断的时序 (比较匹配) (MTU5)

(2) 输入捕捉时的 TGF 标志的置位时序

由发生输入捕捉产生的 TSR 寄存器的 TGF 标志的置位时序和 TGI 中断请求信号的时序分别如图 21.119 和图 21.120 所示。

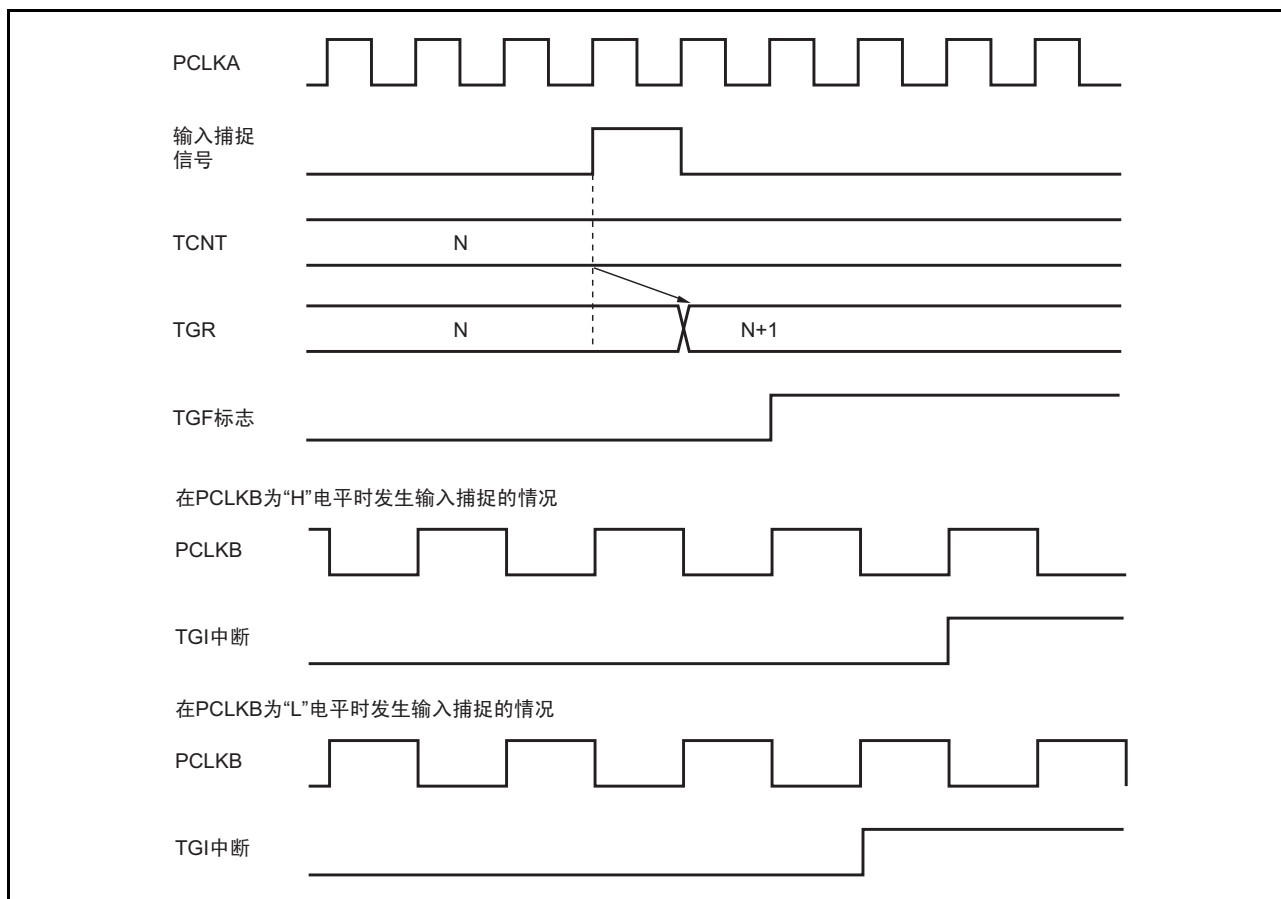


图 21.119 TGI 中断的时序 (输入捕捉) (MTU0 ~ 4、6、7)

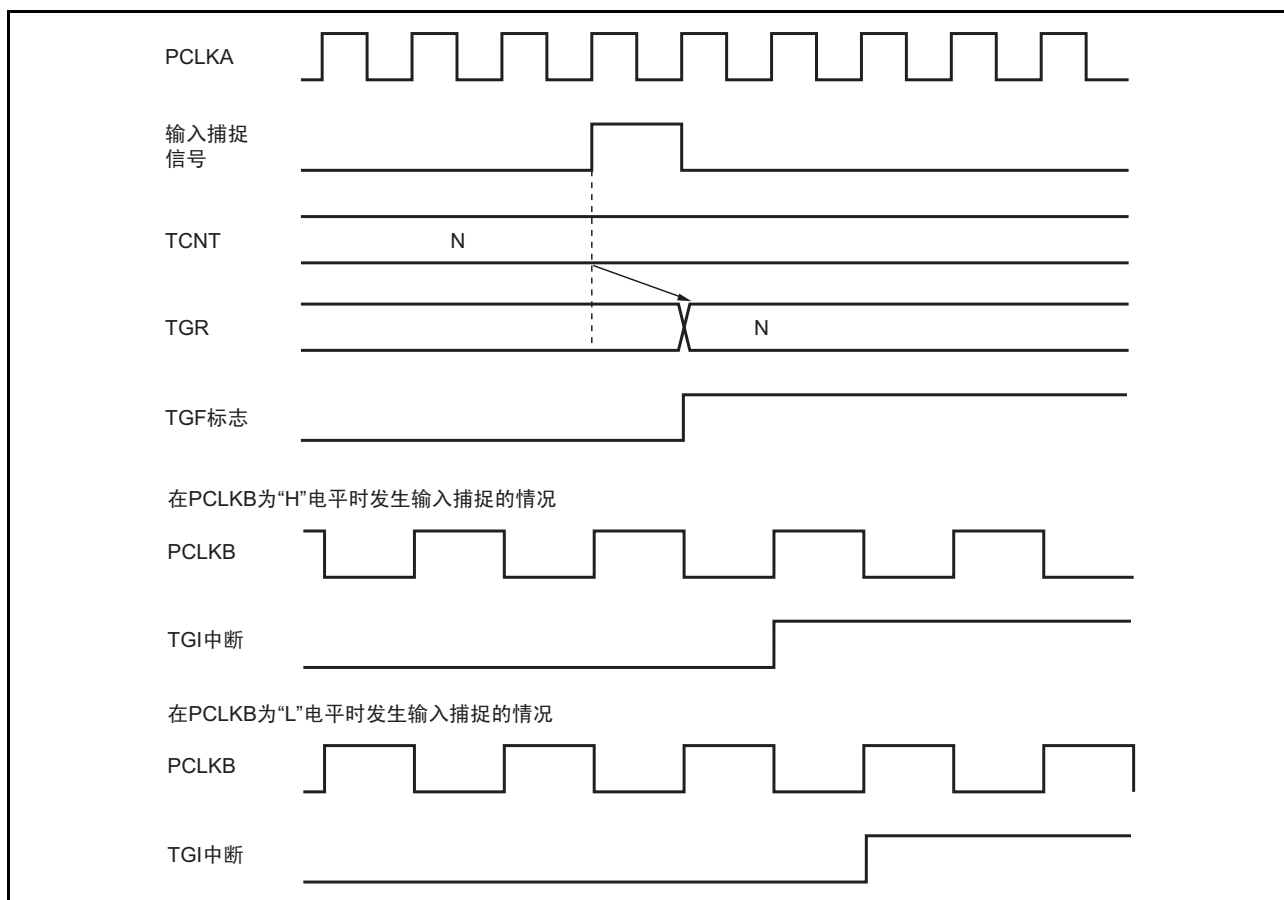


图 21.120 TGI 中断的时序 (输入捕捉) (MTU5)

(3) TCFV/TCFU 标志的置位时序

由发生上溢产生的 TSR 寄存器的 TCFV 标志的置位时序和 TCIV 中断请求信号的时序如图 21.121 所示。
由发生下溢产生的 TSR 寄存器的 TCFU 标志的置位时序和 TCIU 中断请求信号的时序如图 21.122 所示。

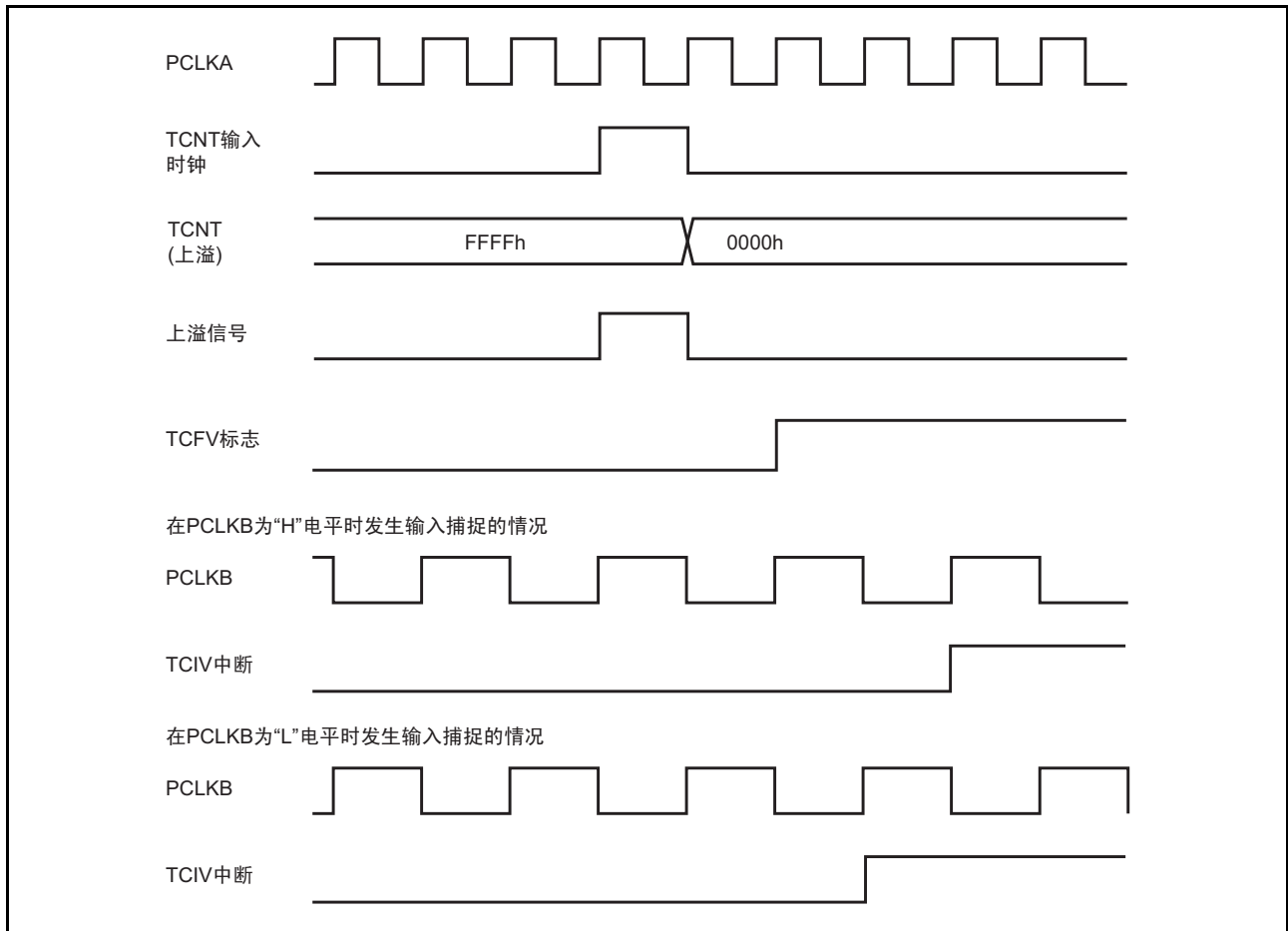


图 21.121 TCIV 中断的置位时序

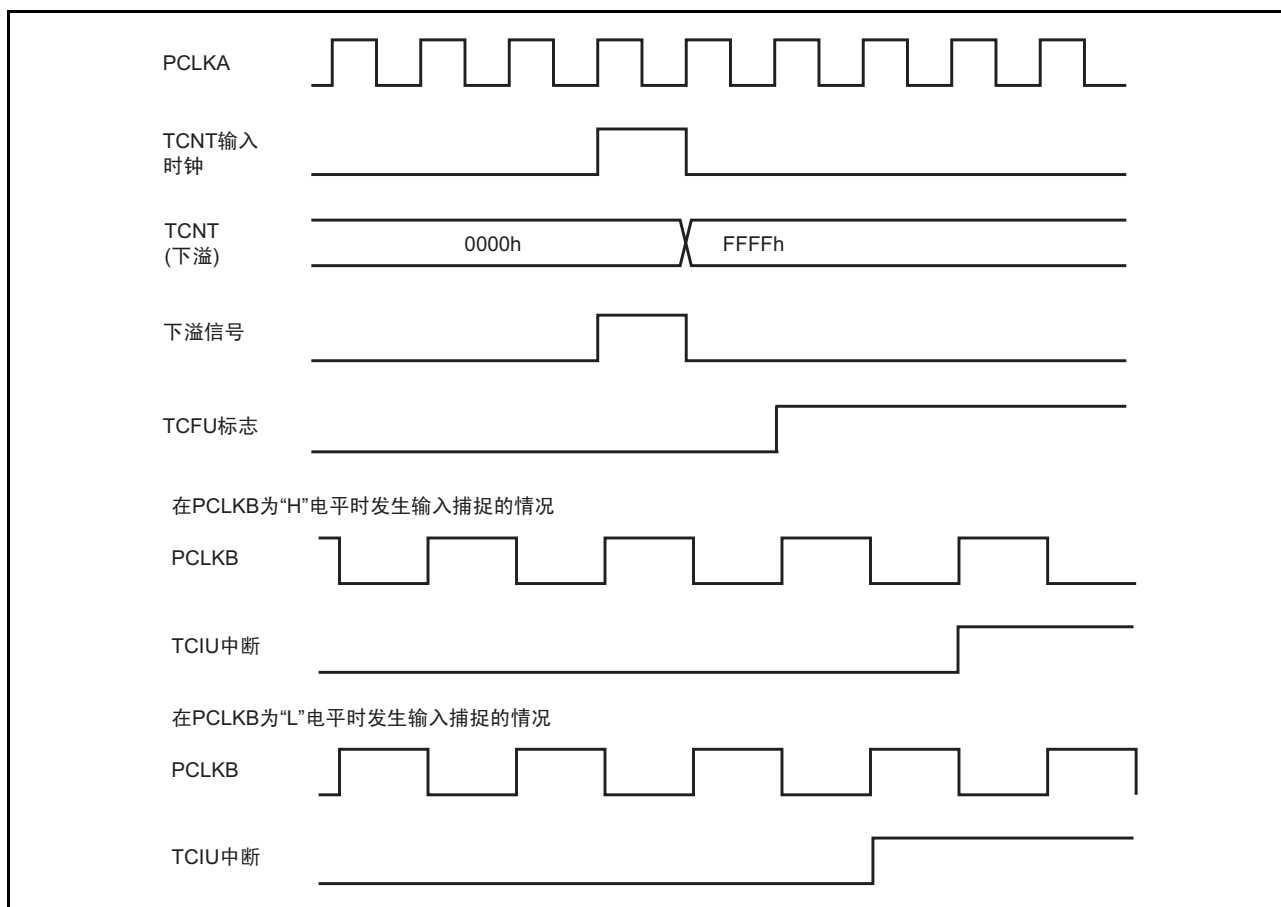


图 21.122 TCIU 中断的置位时序

(4) 状态标志的清除时序

如果在 CPU 读“1”的状态后写“0”，状态标志就变为“0”。通过 CPU 进行的状态标志的清除时序如图 21.123 所示。

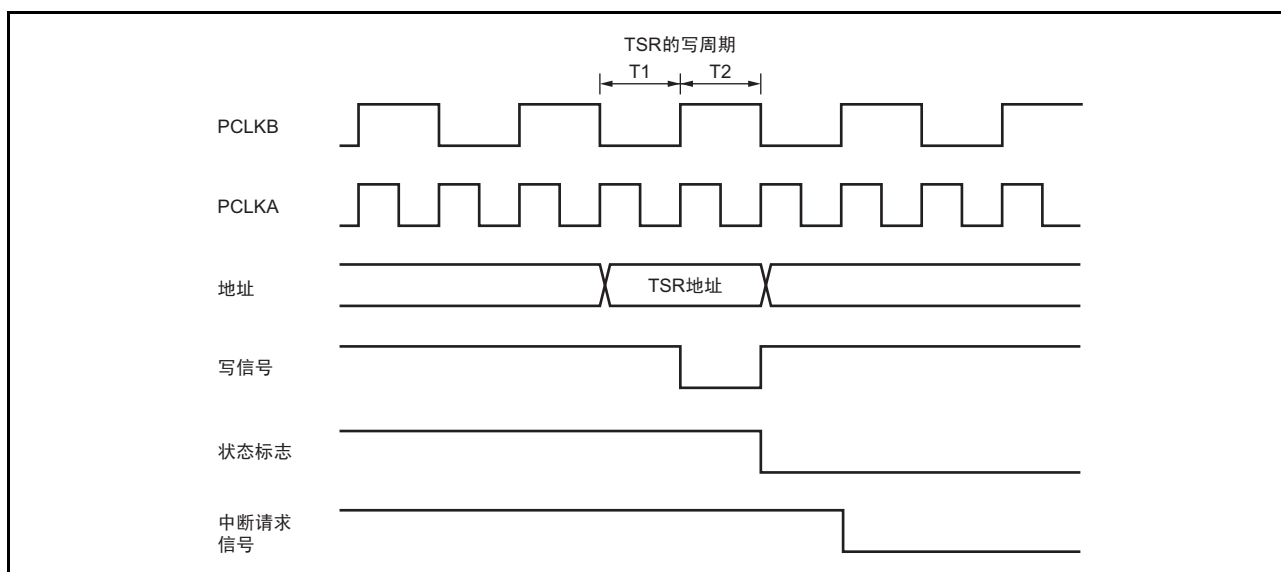


图 21.123 通过 CPU 进行的状态标志的清除时序 (MTU0 ~ 7)

21.6 使用时的注意事项

21.6.1 模块时钟停止模式的设定

能通过模块停止控制寄存器设定允许或者禁止 MTU 的运行，初始值为停止 MTU 的运行。能通过解除模块时钟停止模式，使寄存器变为可存取的状态，详细内容请参照“11. 低功耗功能”。

21.6.2 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 PCLKA 时钟；在双边沿的情况下，输入时钟的脉宽至少为 2.5 个 PCLKA 时钟。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 PCLKA 时钟，脉宽至少为 2.5 个 PCLKA 时钟。相位计数模式的输入时钟条件如图 21.124 所示。

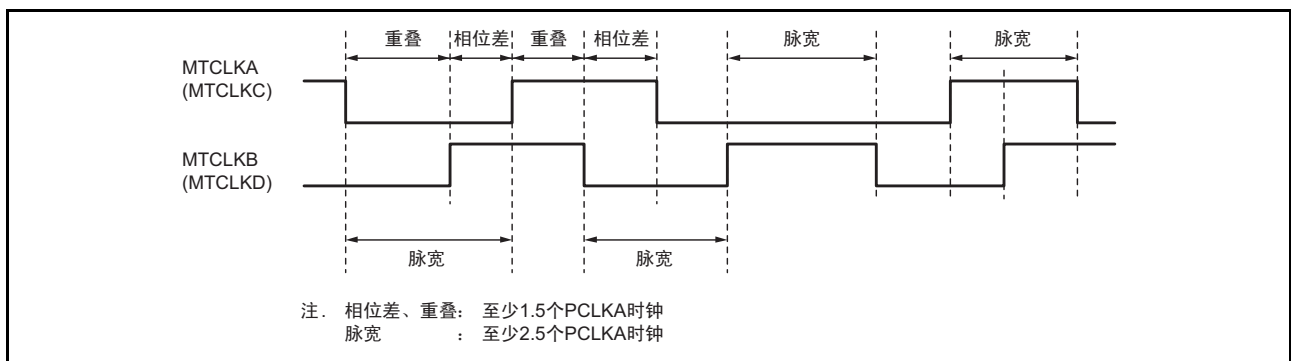


图 21.124 相位计数模式的相位差、重叠和脉宽

21.6.3 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 TCNT 和 TGR 的值相同的最后状态（在更新 TCNT 相同后的计数值时）清除 TCNT 计数器。因此，实际的计数器频率用以下表达式表示：

- MTU0~4、6、7的情况

$$f = \frac{\text{CNTCLK}}{(N+1)}$$

- MTU5的情况

$$f = \frac{\text{CNTCLK}}{N}$$

f : 计数器的频率

CNTCLK : 通过 TCR 的 TPSC[2:0] 位设定的计数器时钟频率

N : TGR 的设定值

21.6.4 TCNT 的写和清除的竞争

如果在 TCNT 的写周期中的 T2 状态产生计数器清除信号，就不写 TCNT 而优先清除 TCNT。此时序如图 21.125 所示。

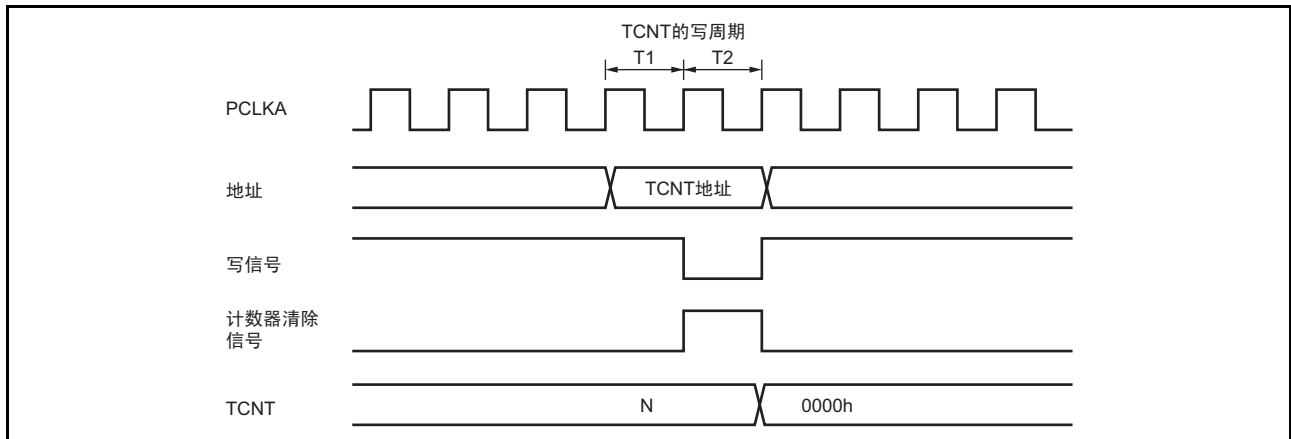


图 21.125 TCNT 的写和计数器清除的竞争

21.6.5 TCNT 的写和递增计数的竞争

即使在 TCNT 的写周期中的 T2 状态发生递增计数，也不进行递增计数而优先写 TCNT。此时序如图 21.126 所示。

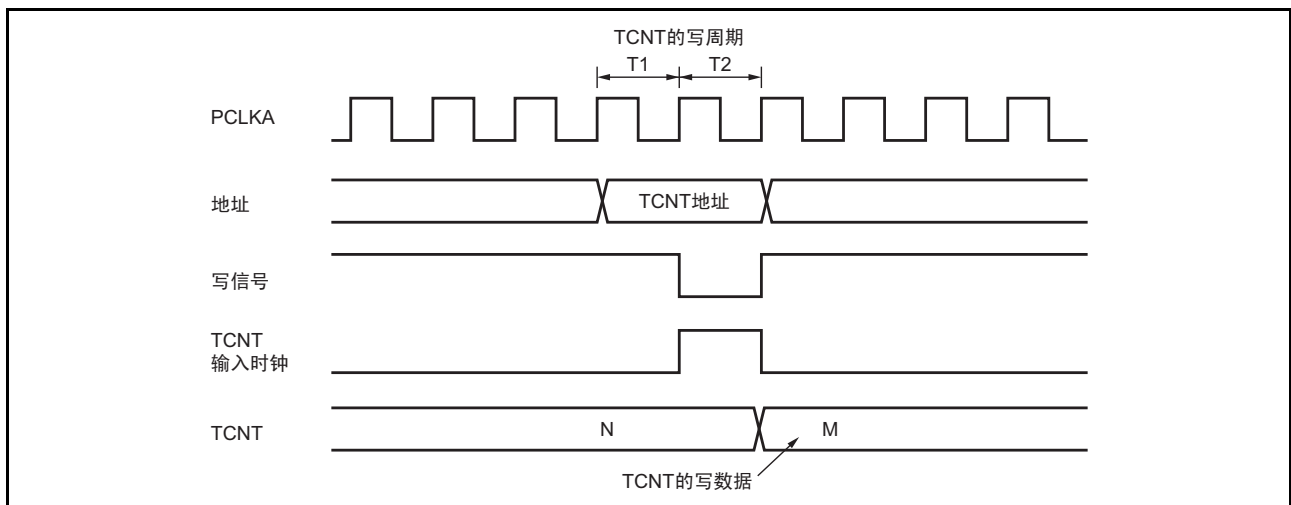


图 21.126 TCNT 的写和递增计数的竞争

21.6.6 TGR 的写和比较匹配的竞争

如果在 TGR 的写周期中的 T2 状态发生比较匹配，就写 TGR 并且也产生比较匹配信号。此时序如图 21.127 所示。

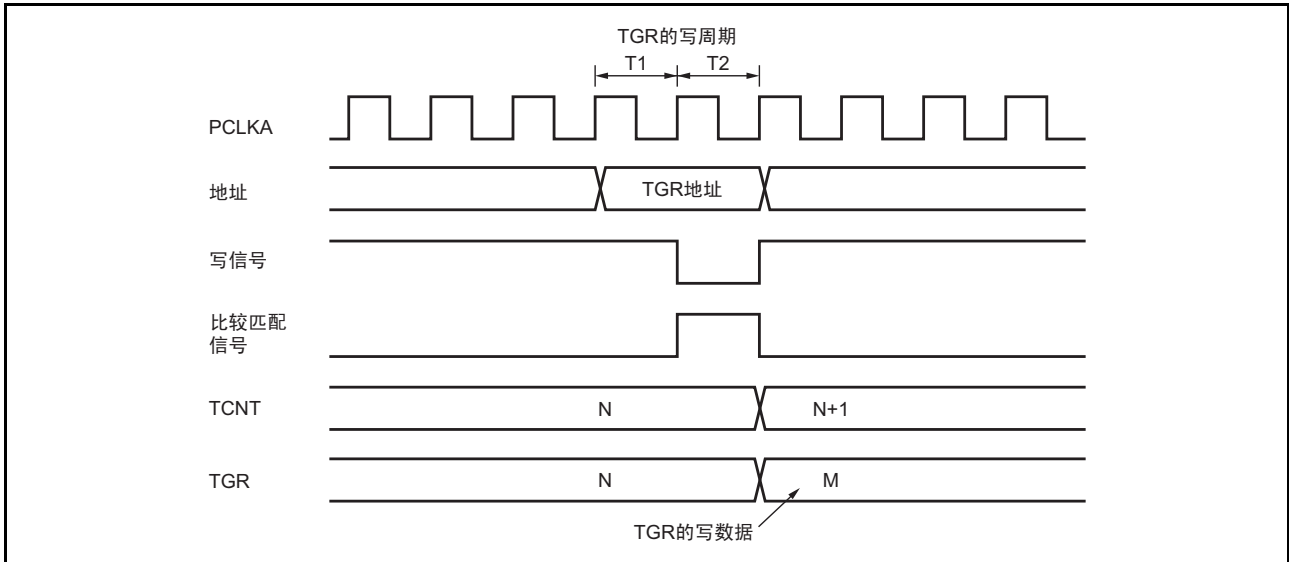


图 21.127 TGR 的写和比较匹配的竞争

21.6.7 缓冲寄存器的写和比较匹配的竞争

如果在 TGR 的写周期中的 T2 状态发生比较匹配，通过缓冲运行传送到 TGR 的数据就为写之前的数据。此时序如图 21.128 所示。

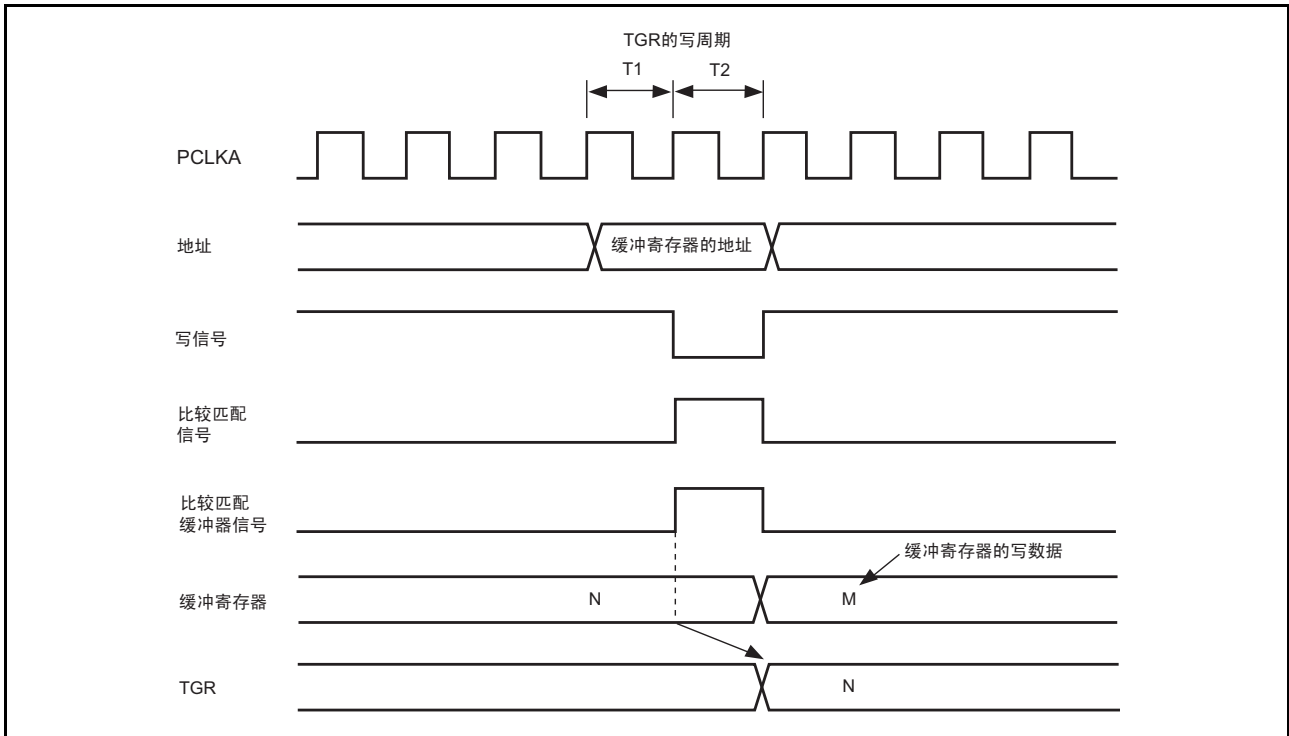


图 21.128 缓冲寄存器的写和比较匹配的竞争

21.6.8 缓冲寄存器的写和 TCNT 清除的竞争

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

此时序如图 21.129 所示。

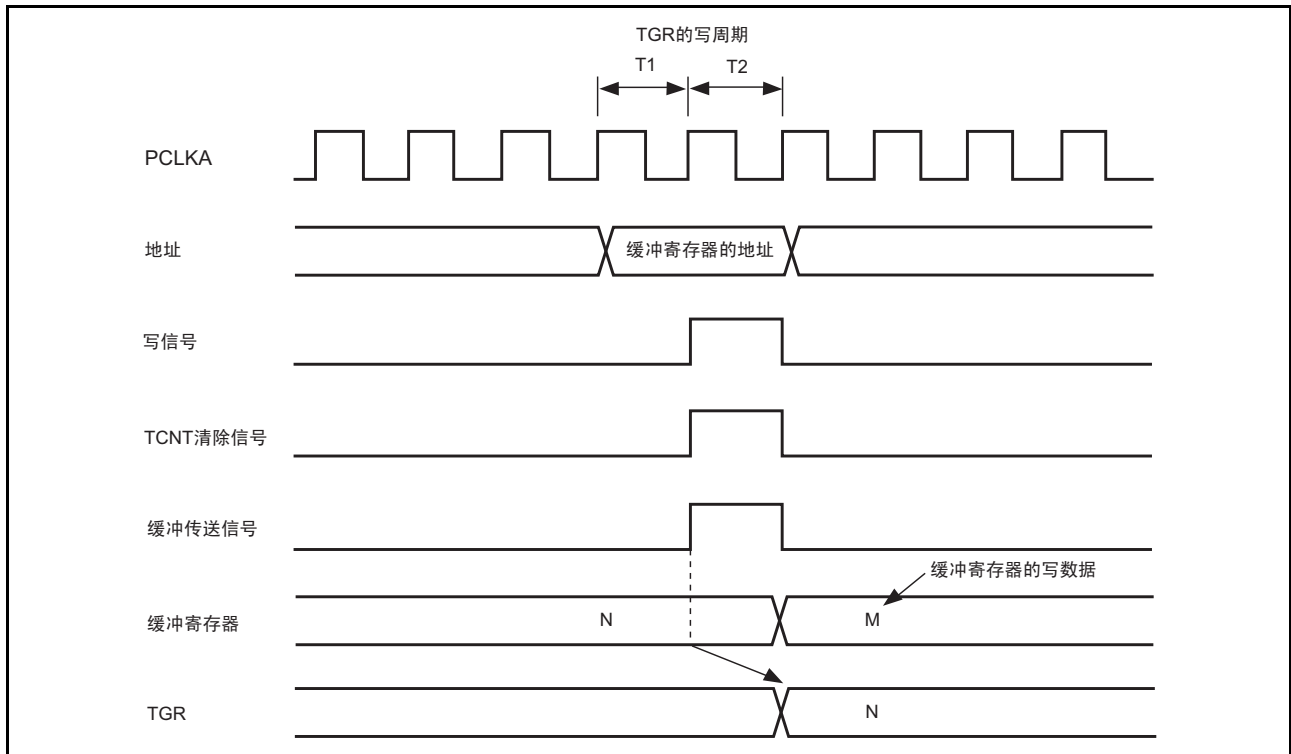


图 21.129 缓冲寄存器的写和 TCNT 清除的竞争

21.6.9 TGR 寄存器的读和输入捕捉的竞争

如果在 TGR 寄存器的读周期中的 T1 状态产生输入捕捉信号, 读出的数据就在 MTU0 ~ 4、6、7 时为输入捕捉传送前的数据而在 MTU5 时为输入捕捉传送后的数据。

此时序如图 21.130 和图 21.131 所示。

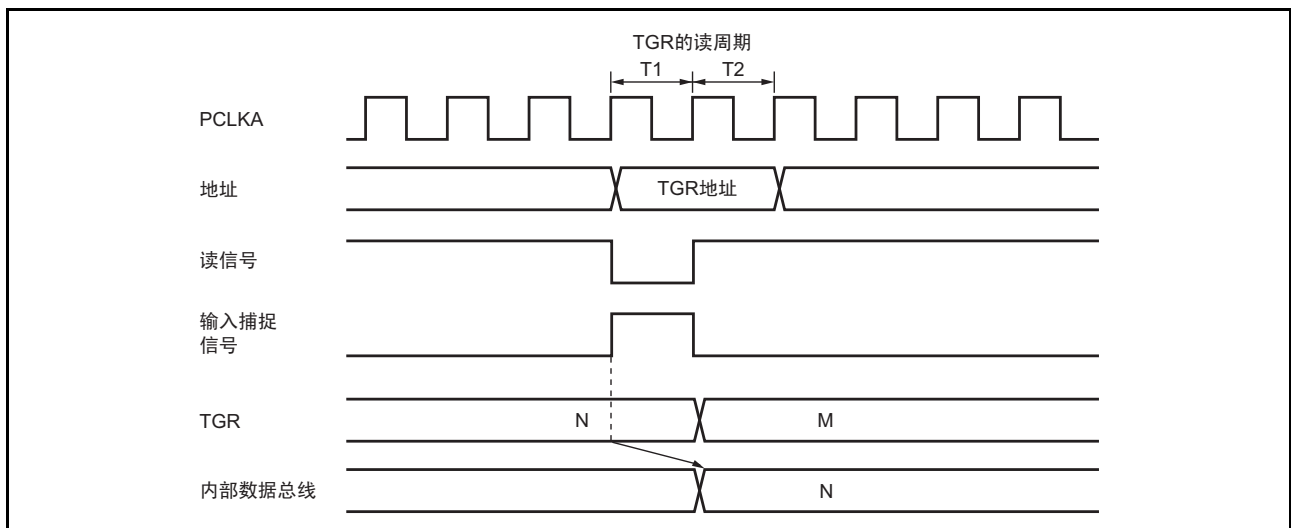


图 21.130 TGR 寄存器的读和输入捕捉的竞争 (MTU0 ~ 4、6、7)

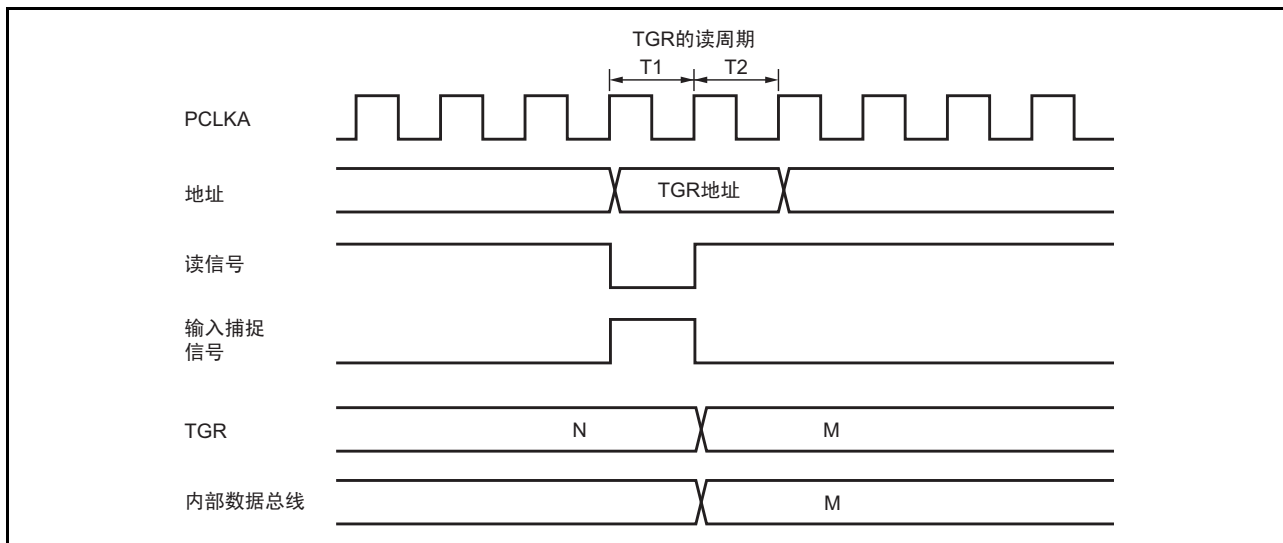


图 21.131 TGR 寄存器的读和输入捕捉的竞争 (MTU5)

21.6.10 TGR 寄存器的写和输入捕捉的竞争

如果在 TGR 寄存器的写周期中的 T2 状态产生输入捕捉信号，就在 MTU0 ~ 4、6、7 时不写 TGR 寄存器而优先输入捕捉，但是在 MTU5 时写 TGR 寄存器并且也产生输入捕捉信号。

此时序如图 21.132 和图 21.133 所示。

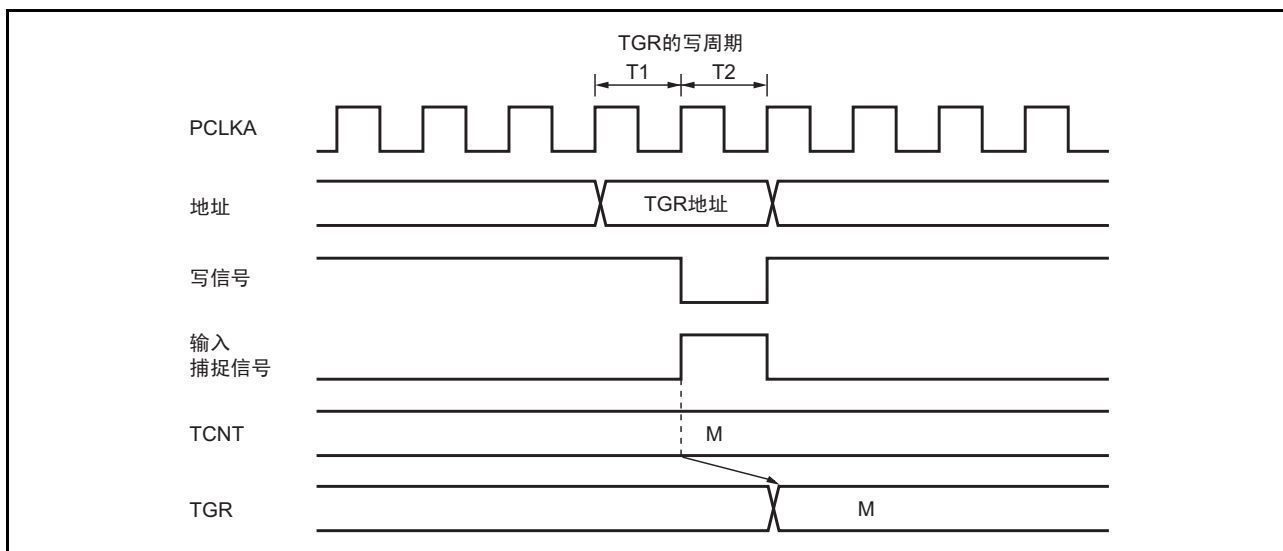


图 21.132 TGR 寄存器的写和输入捕捉的竞争 (MTU0 ~ 4、6、7)

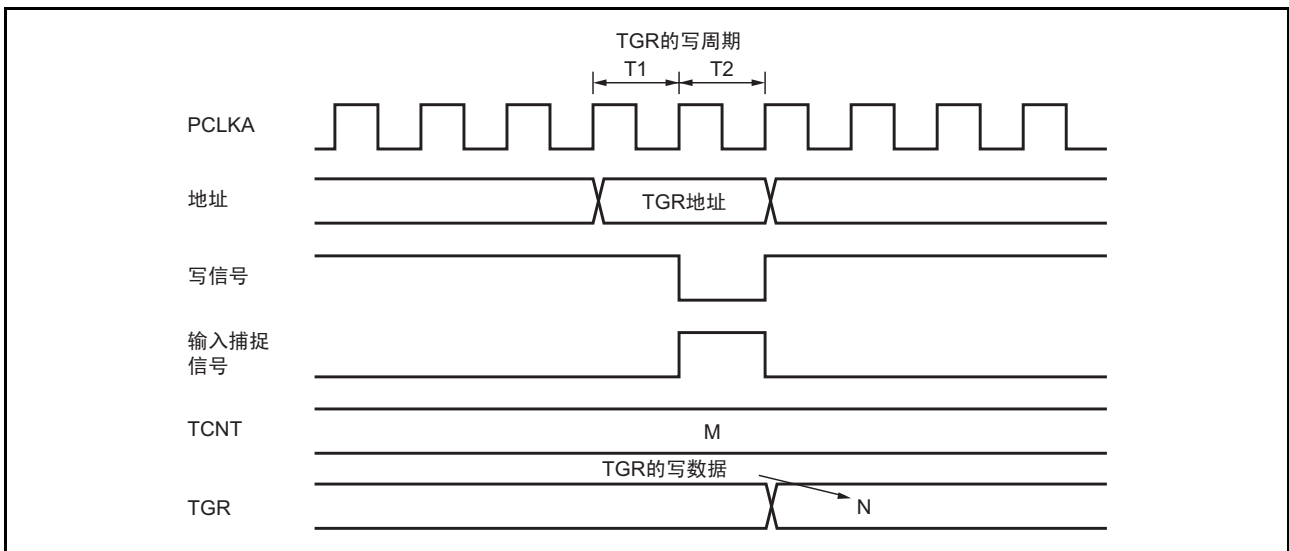


图 21.133 TGR 寄存器的写和输入捕捉的竞争 (MTU5)

21.6.11 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中的 T2 状态产生输入捕捉信号，就不写缓冲寄存器而优先缓冲运行。此时序如图 21.134 所示。

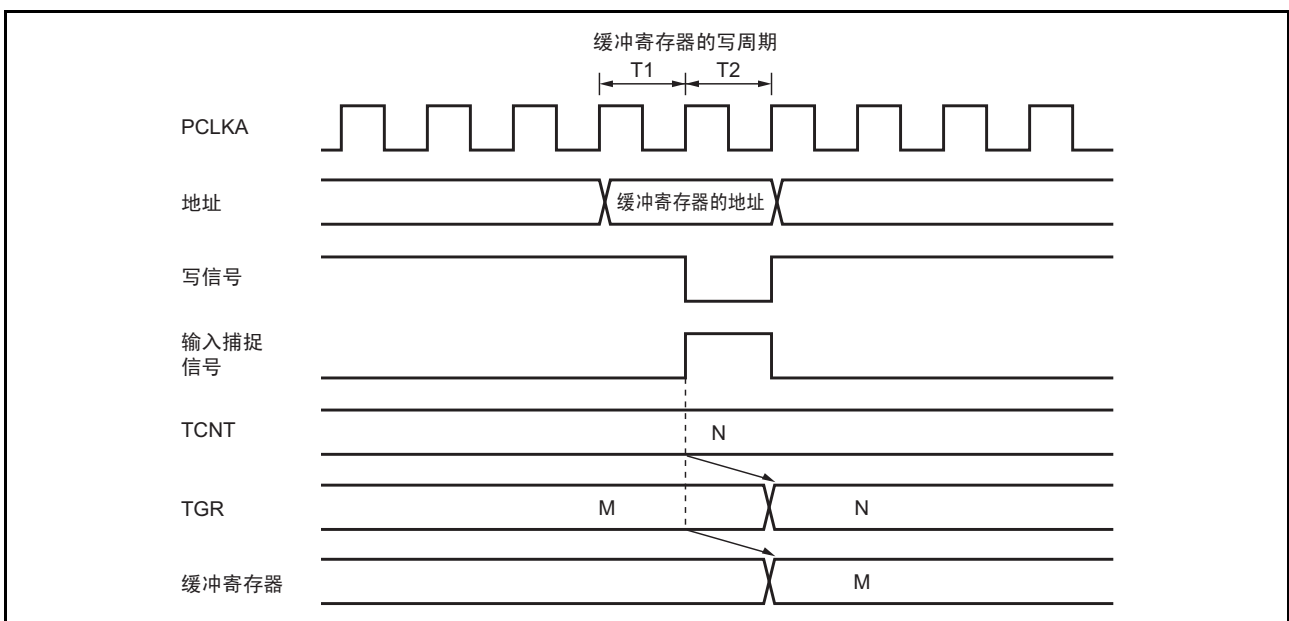


图 21.134 缓冲寄存器的写和输入捕捉的竞争

21.6.12 级联中的 MTU2.TCNT 的写和上溢 / 下溢的竞争

如果级联定时器的计数器 (MTU1.TCNT 和 MTU2.TCNT) 并且在 MTU1.TCNT 计数的瞬间 (MTU2.TCNT 上溢 / 下溢的瞬间) 和 MTU2.TCNT 的写周期中的 T2 状态发生竞争, 就写 MTU2.TCNT 而禁止 MTU1.TCNT 的计数信号。此时, MTU1.TGRA 作为比较匹配寄存器运行, 如果和 MTU1.TCNT 的值相同, 就产生比较匹配信号。

如果选择 MTU1.TCNT 计数时钟作为 MTU0 的输入捕捉源, MTU0.TGRA ~ TGRD 就进行输入捕捉运行。如果选择 MTU0.TGRC 的比较匹配 / 输入捕捉作为 MTU1.TGRB 的输入捕捉源, MTU1.TGRB 就进行输入捕捉运行。

此时序如图 21.135 所示。

如果在级联运行时设定清除 TCNT, 就必须进行 MTU1 和 MTU2 的同步设定。

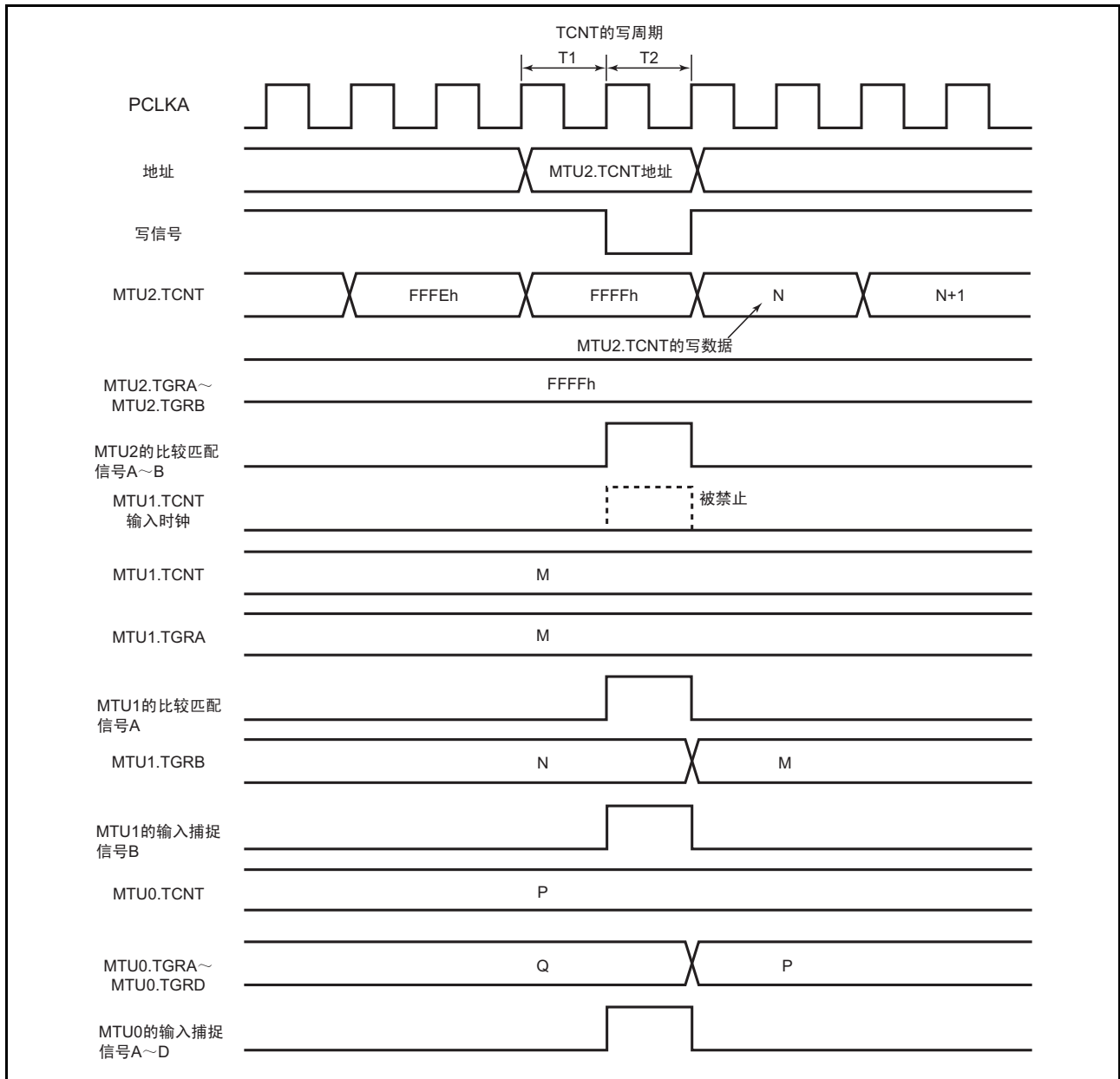


图 21.135 级联中的 MTU2.TCNT 的写和上溢 / 下溢的竞争

21.6.13 互补 PWM 模式停止时的计数器值

如果 MTU3.TCNT 和 MTU4.TCNT (MTU6.TCNT 和 MTU7.TCNT) 在互补 PWM 运行时停止计数运行, MTU3.TCNT (MTU6.TCNT) 就变为定时器的死区时间寄存器 (TDDRA (TDDRB)) 的值, MTU4.TCNT (MTU7.TCNT) 变为“0000h”。

一旦重新开始互补 PWM 运行, 计数器就自动从初始状态开始计数。

此说明图如图 21.136 所示。

如果要在其他运行模式中开始计数, 就必须给 MTU3.TCNT 和 MTU4.TCNT (MTU6.TCNT 和 MTU7.TCNT) 设定计数初始值。

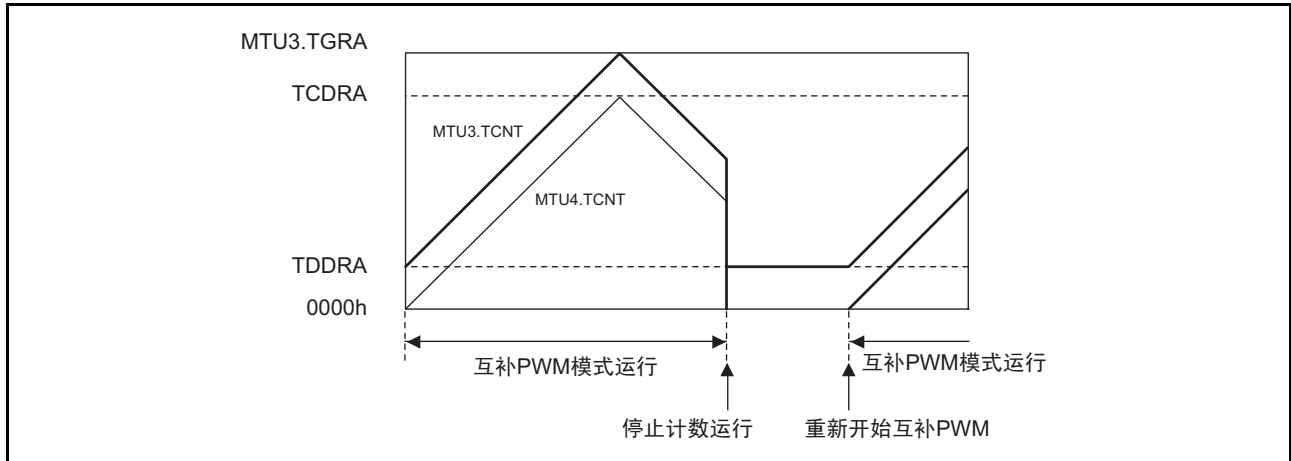


图 21.136 互补 PWM 模式停止时的计数器值

21.6.14 互补 PWM 模式中的缓冲运行的设定

在互补 PWM 模式中, 必须通过缓冲运行改写 PWM 周期设定寄存器 (MTU3.TGRA、MTU6.TGRA)、定时器的周期数据寄存器 (TCDRA、TCDRB) 和占空比设定寄存器 (MTU3.TGRB、MTU3.TGRA、MTU4.TGRB、MTU6.TGRB、MTU7.TGRA、MTU7.TGRB), 并且将 MTU4.TMDR1 (MTU7.TMDR1) 的 BFA 位和 BFB 位置“0”。如果将 MTU4.TMDR1 (MTU7.TMDR1) 的 BFA 位置“1”, 就无法进行 MTIOC4C (MTIOC7C) 引脚的波形输出。同样, 如果将 MTU4.TMDR1 (MTU7.TMDR1) 的 BFB 位置“1”, 就无法进行 MTIOC4D (MTIOC7D) 引脚的波形输出。

按照 MTU3.TMDR1 (MTU6.TMDR1) 的 BFA 位和 BFB 位的设定, 进行互补 PWM 模式的 MTU3 和 MTU4 (MTU6 和 MTU7) 的缓冲运行。如果将 MTU3.TMDR1 (MTU6.TMDR1) 的 BFA 位置“1”, 就在 MTU3.TGRC (MTU6.TGRC) 用作 MTU3.TGRA (MTU6.TGRA) 的缓冲寄存器的同时, MTU4.TGRC (MTU7.TGRC) 用作 MTU4.TGRA (MTU7.TGRA) 的缓冲寄存器, 并且 TCBRA (TCBRB) 用作 TCDRA (TCDRB) 的缓冲寄存器。

21.6.15 复位同步 PWM 模式的缓冲运行和比较匹配标志

要设定为在复位同步 PWM 模式中进行缓冲运行时，必须将 MTU4.TMDR1 (MTU7.TMDR1) 的 BFA 位和 BFB 位置“0”。如果将 MTU4.TMDR1 (MTU7.TMDR1) 的 BFA 位置“1”，就不能进行 MTIOC4C (MTIOC7C) 引脚的波形输出。同样，如果将 MTU4.TMDR1 (MTU7.TMDR1) 的 BFB 位置“1”，就不能进行 MTIOC4D (MITOC7D) 引脚的波形输出。

按照 MTU3.TMDR1 (MTU6.TMDR1) 的 BFA 位和 BFB 位的设定，进行复位同步 PWM 模式的 MTU3 和 MTU4 (MTU6 和 MTU7) 的缓冲运行。例如，如果将 MTU3.TMDR1 (MTU6.TMDR1) 的 BFA 位置“1”，就在 MTU3.TGRC (MTU6.TGRC) 用作 MTU3.TGRA (MTU6.TGRA) 的缓冲寄存器的同时，MTU4.TGRC (MTU7.TGRC) 用作 MTU4.TGRA (MTU7.TGRA) 的缓冲寄存器。

当 MTU3.TGRC (MTU6.TGRC) 和 MTU3.TGRD (MTU6.TGRD) 用作缓冲寄存器时，MTU3.TSR 和 MTU4.TSR (MTU6.TSR 和 MTU7.TSR) 的 TGFC 位和 TGFD 位不被置位。

将 MTU3.TMDR1 (MTU6.TMDR1) 的 BFA 位和 BFB 位置“1”并且将 MTU4.TMDR1 (MTU7.TMDR1) 的 BFA 位和 BFB 位置“0”时的 MTU3.TGR (MTU6.TGR)、MTU4.TGR (MTU7.TGR)、MTIOC3 (MTIOC6)、MTIOC4 (MTIOC7) 的运行例子如图 21.137 所示。

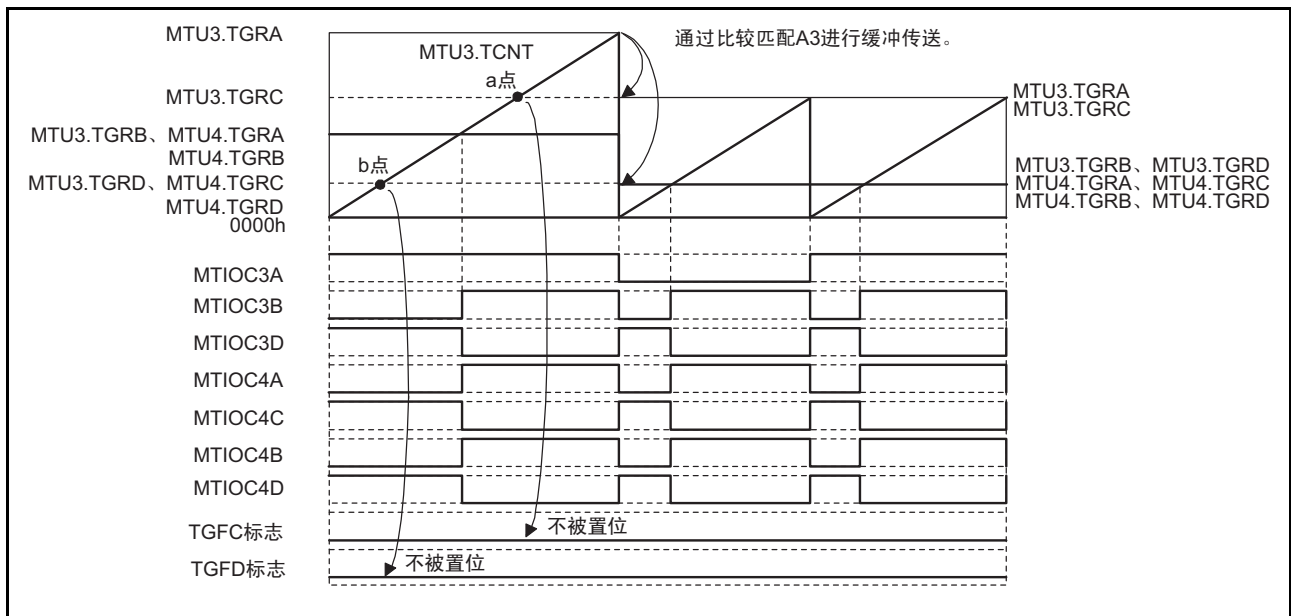


图 21.137 复位同步 PWM 模式的缓冲运行和比较匹配标志

21.6.16 复位同步 PWM 模式的上溢标志

如果设定复位同步 PWM 模式并且将 TSTRA (TSTRB) 的 CST3 (CST6) 位置“1”，就开始 MTU3.TCNT 和 MTU4.TCNT (MTU6.TCNT 和 MTU7.TCNT) 的计数运行。此时，MTU4.TCNT (MTU7.TCNT) 的计数时钟源和计数边沿服从 MTU3.TCR (MTU6.TCR) 的设定。

在复位同步 PWM 模式中，当周期寄存器 MTU3.TGRA (MTU6.TGRA) 的设定值为“FFFFh”并且指定 MTU3.TGRA (MTU6.TGRA) 的比较匹配为计数器清除源时，如果 MTU3.TCNT 和 MTU4.TCNT (MTU6.TCNT 和 MTU7.TCNT) 递增计数到“FFFFh”，就发生和 MTU3.TGRA (MTU6.TGRA) 的比较匹配，并且清除 MTU3.TCNT 和 MTU4.TCNT (MTU6.TCNT 和 MTU7.TCNT)。此时，TSR 寄存器的上溢标志 TCFV 标志不变为“1”。

在复位同步 PWM 模式中，当周期寄存器 MTU3.TGRA (MTU6.TGRA) 的设定值为“FFFFh”并且指定 MTU3.TGRA (MTU6.TGRA) 的比较匹配为计数器清除源而不进行同步设定时的 TCFV 标志的运行例子如图 21.138 所示。

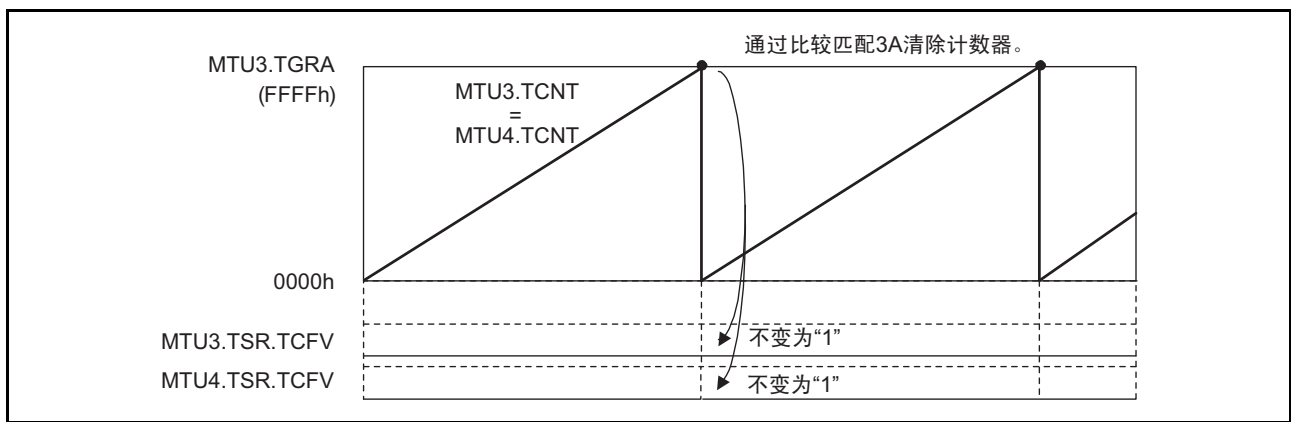


图 21.138 复位同步 PWM 模式的上溢标志

21.6.17 上溢 / 下溢和计数器清除的竞争

如果同时发生上溢 / 下溢和计数器清除，就不将 TSR 寄存器的 TCFV/TCFU 标志置“1”而优先清除 TCNT。

将 TGR 寄存器的比较匹配作为清除源并且给 TGR 寄存器设定“FFFFh”时的运行时序如图 21.139 所示。

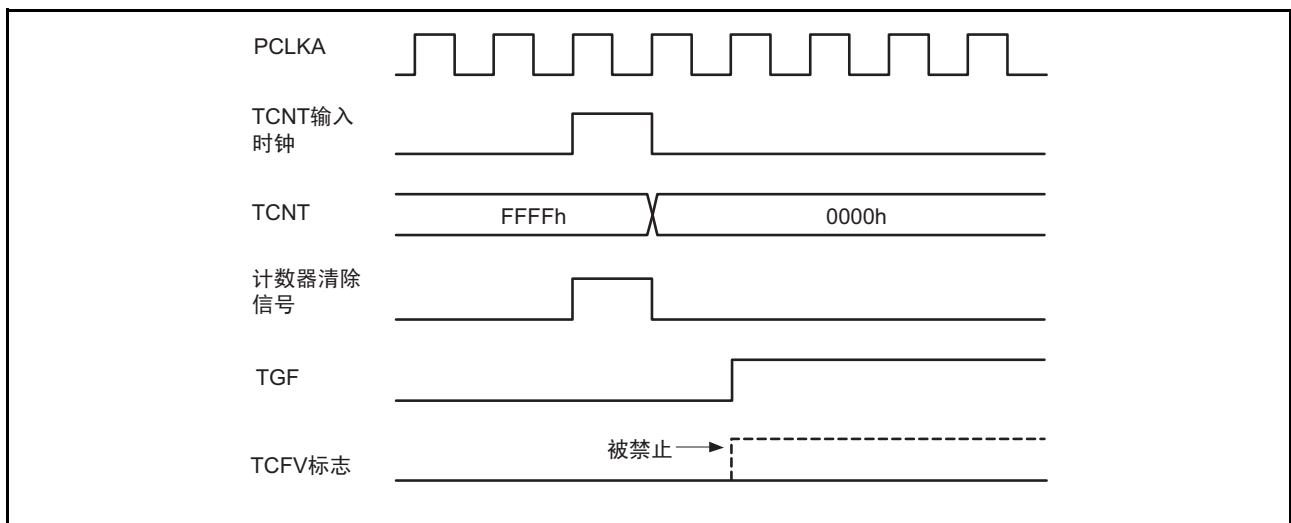


图 21.139 上溢和计数器清除的竞争

21.6.18 TCNT 的写和上溢 / 下溢的竞争

即使在 TCNT 的写周期中的 T2 状态发生递增计数 / 递减计数和上溢 / 下溢，也优先写 TCNT 而 TSR 寄存器的 TCFV/TCFU 标志不变为“1”。

TCNT 的写和上溢竞争时的运行时序如图 21.140 所示。

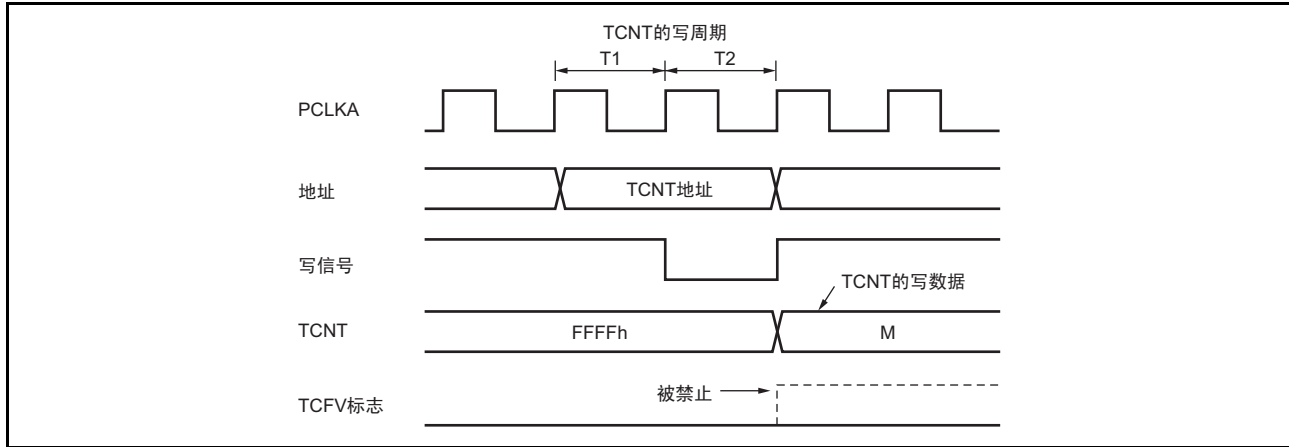


图 21.140 TCNT 的写和上溢的竞争

21.6.19 从正常运行或者 PWM 模式 1 转移到复位同步 PWM 模式时的注意事项

要从 MTU3 和 MTU4 (MTU6 和 MTU7) 的正常运行或者 PWM 模式 1 转移到复位同步 PWM 模式时，如果在将输出引脚 (MTIOC3B、MTIOC3D、MTIOC4A、MTIOC4C、MTIOC4B、MTIOC4D、MTIOC6B、MTIOC6D、MTIOC7A、MTIOC7C、MTIOC7B、MTIOC7D) 置为高电平的状态下停止计数器并且在转移到复位同步 PWM 模式后进行运行，就不能正确地进行引脚的初始输出，因此必须注意。

在从正常运行转移到复位同步 PWM 模式时，必须在给 MTU3.TIORH、MTU3.TIORL、MTU4.TIORH、MTU4.TIORL (MTU6.TIORH、MTU6.TIORL、MTU7.TIORH、MTU7.TIORL) 寄存器写“11h”并且将输出引脚初始化为低电平后设定寄存器的初始值“00h”，然后进行模式移行。

在从 PWM 模式 1 转移到复位同步 PWM 模式时，必须暂且转移到正常运行，在将输出引脚初始化为低电平后设定寄存器的初始值“00h”，然后转移到复位同步 PWM 模式。

21.6.20 互补 PWM 模式、复位同步 PWM 模式的输出电平

当 MTU3 和 MTU4 (MTU6 和 MTU7) 为互补 PWM 模式或者复位同步 PWM 模式时，通过定时器的输出控制寄存器 (TOCR1A、TOCR1B) 的 OLSP 位和 OLSN 位设定 PWM 波形的输出电平。并且必须将 TIOR 寄存器设定为“00h”。

在互补 PWM 模式中将 TDER.TDER 位设定为“0” (不生成死区时间) 时的反相输出电平与 TOCR1.OLSN 位的设定无关，为通过 TOCR1.OLSP 位的设定产生的正相输出的取反电平。

21.6.21 级联中的 MTU1.TCNT 和 MTU2.TCNT 的同时输入捕捉

在将定时器的计数器 1 和计数器 2 (MTU1.TCNT 和 MTU2.TCNT) 级联用作 32 位计数器时, 即使 MTIOC1A 和 MTIOC2A 或者 MTIOC1B 和 MTIOC2B 同时进行输入捕捉的输入, 也会与内部时钟同步将输入到 MTU1.TCNT 和 MTU2.TCNT 的外部输入捕捉信号取到内部, 此时可能因 MTIOC1A 和 MTIOC2A 或者 MTIOC1B 和 MTIOC2B 的取时序产生偏差而不能正确地捕捉级联计数器的值。

例如, MTU1.TCNT (高 16 位的计数器) 应该捕捉由 MTU2.TCNT (低 16 位的计数器) 的上溢产生的递增计数值, 却会捕捉递增计数前的计数值。此时, 应该将 MTU1.TCNT 为 “FFF1h” 以及 MTU2.TCNT 为 “0000h” 的值传送到 MTU1.TGRA 和 MTU2.TGRA 或者 MTU1.TGRB 和 MTU2.TGRB, 却会误传送 MTU1.TCNT 为 “FFF0h” 以及 MTU2.TCNT 为 “0000h” 的值。

在 MTU 中追加了通过 1 个输入捕捉的输入引脚同时捕捉 MTU1.TCNT 和 MTU2.TCNT 的功能。如果使用本功能, 就能在不发生 MTU1.TCNT 和 MTU2.TCNT 的捕捉时序不一致的情况下执行 32 位计数器的读取操作。详细内容请参照 “21.2.9 定时器的输入捕捉控制寄存器 (TICCR) ”。

21.6.22 中断减少功能 2

在使用中断减少功能 2 并且 MTU4.TADCORA 的值和 MTU4.TADCORB 的值的间隔短时, 有可能不能正确地减少次数进行计数并且不发生期待时序的 A/D 转换请求。必须在以下的设定条件下使用:

在 MTU6 和 MTU7 中, 需要对 MTU7.TADCORA 和 MTU7.TADCORB 进行相同的设定。

(1) 当使用中断减少功能 2 并且中断减少次数为 “0” 时

- MTU4.TADCORA 寄存器值与 MTU4.TADCORB 寄存器值之间的间隔 ≥ 4
- MTU4.TADCORA 的比较间隔 ≥ 4 个 PCLKA (将 MTU4.TADCORA 寄存器的更新值设定为: 更新前的值 + 大于等于 4 的值或者更新前的值 - 小于等于 4 的值)
- MTU4.TADCORB 的比较间隔 ≥ 4 个 PCLKA (将 MTU4.TADCORB 寄存器的更新值设定为: 更新前的值 + 大于等于 4 的值或者更新前的值 - 小于等于 4 的值)

(2) 当使用中断减少功能 2 并且中断减少次数大于等于 “1” 时

- MTU4.TADCORA 寄存器值与 MTU4.TADCORB 寄存器值之间的间隔 ≥ 2
- MTU4.TADCORB 的比较间隔 ≥ 2 个 PCLKA (将 MTU4.TADCORB 寄存器的更新值设定为: 更新前的值 + 大于等于 2 的值或者更新前的值 - 小于等于 2 的值)

21.7 MTU 输出引脚的初始化方法

21.7.1 运行模式

MTU 有以下 6 种运行模式，能在任意的模式中进行波形输出。

- 正常模式 (MTU0~4、6、7)
- PWM 模式 1 (MTU0~4、6、7)
- PWM 模式 2 (MTU0~2)
- 相位计数模式 1~4 (MTU1、2)
- 互补 PWM 模式 (MTU3、4、6、7)
- 复位同步 PWM 模式 (MTU3、4、6、7)

在此说明各模式中的 MTU 输出引脚的初始化方法。

21.7.2 因运行过程中的异常而重新设定时的运行

如果在 MTU 运行过程中发生异常，就必须通过系统切断 MTU 的输出。要切断输出时，必须通过 I/O 端口的端口方向寄存器 (PDR)、端口输出数据寄存器 (PODR) 和端口模式寄存器 (PMR) 将引脚切换为通用输出端口，并且输出无效电平。通过 TIOR 寄存器将 MTU 引脚设定为输出禁止。通过 TOERA 寄存器和 TOERB 寄存器设定互补 PWM 输出 (MTIOC3B、MTIOC3D、MTIOC4A、MTIOC4B、MTIOC4C、MTIOC4D、MTIOC6B、MTIOC6D、MTIOC7A、MTIOC7B、MTIOC7C、MTIOC7D)。另外，对于 PWM 输出引脚，也能使用端口输出允许 3 (POE3)，通过硬件来切断输出。因运行过程中的异常等而重新设定时的引脚初始化步骤以及重新设定后在其他运行模式中重新开始时的步骤如下所示。

如上所述，因为 MTU 有 6 个运行模式，所以有 36 种模式转移的组合，但是有通道和模式的组合中不存在的转移。模式转移的组合一览表如表 21.74 所示。

表中使用下述的符号表示：

Normal: 正常模式 PWM1: PWM 模式 1 PWM2: PWM 模式 2
PCM: 相位计数模式 1~4 CPWM: 互补 PWM 模式 RPWM: 复位同步 PWM 模式

表 21.74 模式转移的组合

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	none	none
PCM	(17)	(18)	(19)	(20)	none	none
CPWM	(21)	(22)	none	none	(23) (24)	(25)
RPWM	(26)	(27)	none	none	(28)	(29)

21.7.3 因运行过程中的异常而对引脚进行初始化的步骤以及模式转移的概要

- 在转移到定时器的I/O控制寄存器 (TIOR) 选择的引脚输出电平模式 (Normal、PWM1、PWM2、PCM) 时, 必须通过设定TIOR对引脚进行初始化。
- 在PWM模式1中不将波形输出到MTIOcNB/MTIOcND (n=3、4、6、7) 引脚。如果没有输出对应引脚的其他模块就变为高阻抗状态。如果有应该输出的电平时, 就通过I/O端口的端口方向寄存器 (PDR)、端口输出数据寄存器 (PODR) 和端口模式寄存器 (PMR) 设定通用输出端口。
- 在PWM模式2中不将波形输出到周期寄存器的引脚。如果没有输出对应引脚的其他模块就变为高阻抗状态。如果有应该输出的电平时, 就通过I/O端口的端口方向寄存器 (PDR)、端口输出数据寄存器 (PODR) 和端口模式寄存器 (PMR) 设定通用输出端口。
- 在正常模式或者PWM模式2中, 如果TGRC和TGRD都用作缓冲寄存器, 就不将波形输出到引脚。如果没有输出对应引脚的其他模块就变为高阻抗状态。如果有应该输出的电平时, 就通过I/O端口的端口方向寄存器 (PDR)、端口输出数据寄存器 (PODR) 和端口模式寄存器 (PMR) 设定通用输出端口。
- 在PWM模式1中, 如果TGRC或者TGRD用作缓冲寄存器, 就不将波形输出到引脚。如果没有输出对应引脚的其他模块就变为高阻抗状态。如果有应该输出的电平时, 就通过I/O端口的端口方向寄存器 (PDR)、端口输出数据寄存器 (PODR) 和端口模式寄存器 (PMR) 设定通用输出端口。
- 在转移到通过定时器的输出控制寄存器 (TOCR1A、TOCR2A、TOCR1B、TOCR2B) 选择的引脚输出电平模式 (CPWM、RPWM) 时, 必须通过定时器的输出主控允许寄存器 (TOERA、TOERB) 将MTU3和MTU4 (MTU6或MTU7) 设定为输出禁止。此时, 如果没有输出对应引脚的其他模块就变为高阻抗状态。如果有应该输出的电平时, 就通过I/O端口的端口方向寄存器 (PDR)、端口输出数据寄存器 (PODR) 和端口模式寄存器 (PMR) 设定通用输出端口。转移到正常模式后通过TIOR寄存器进行初始化, 在将TIOR恢复为初始值后按照模式的设定步骤 (TOCR1A的设定、TOCR2A的设定、TMDR1的设定和TOERA的设定 (TOCR1B的设定、TOCR2B的设定、TMDR1的设定和TOERB的设定) 运行。

注. 在没有特别要求的情况下, 各项记述中的 n 为通道号。

根据表 21.74 的组合 No. 对引脚进行初始化的步骤如下所示，有效电平为 Low 电平。

(1) 在正常模式的运行过程中发生异常并且在正常模式中重新开始时的运行

在正常模式中发生异常并且重新设定后在正常模式中重新开始时的说明图如图 21.141 所示。

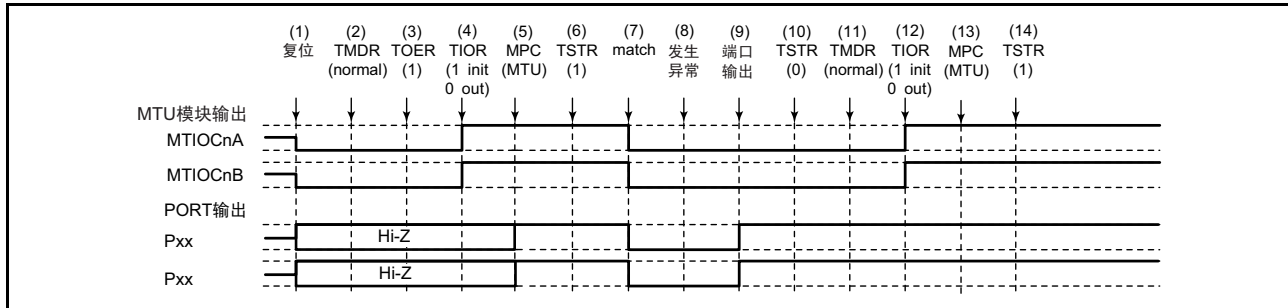


图 21.141 在正常模式中发生异常并且在正常模式中恢复的情况

- (1) 复位后，MTU 的输出为 Low 电平，端口为高阻抗。
- (2) 复位后，TMDR1 寄存器被设定为正常模式。
- (3) 在 MTU3 和 MTU4 中，必须在通过 TIOR 寄存器对引脚进行初始化前用 TOERA 寄存器允许输出。
- (4) 必须通过 TIOR 寄存器对引脚进行初始化
(这是初始输出为 High 电平并且在比较匹配时输出 Low 电平的例子)。
- (5) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。
- (6) 通过 TSTR 寄存器开始计数运行。
- (7) 在发生比较匹配时输出低电平。
- (8) 发生异常。
- (9) 必须通过 I/O 端口的端口方向寄存器 (PDR)、端口输出数据寄存器 (PODR) 和端口模式寄存器 (PMR) 将引脚切换为通用输出端口，并且输出无效电平。
- (10) 通过 TSTR 寄存器停止计数运行。
- (11) 在正常模式中重新开始的情况下不需要此步骤。
- (12) 必须通过 TIOR 寄存器对引脚进行初始化。
- (13) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。
- (14) 通过 TSTR 寄存器重新开始。

(2) 在正常模式的运行过程中发生异常并且在 PWM 模式 1 中重新开始时的运行

在正常模式中发生异常并且重新设定后在 PWM 模式 1 中重新开始时的说明图如图 21.142 所示。

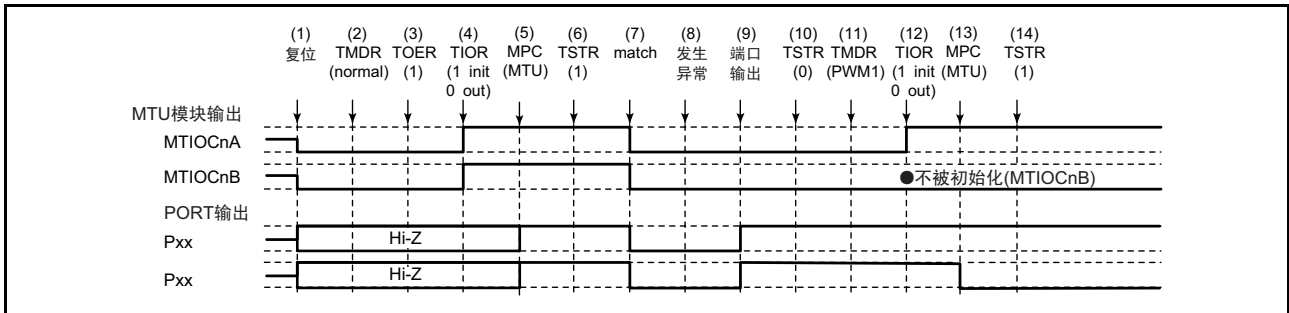


图 21.142 在正常模式中发生异常并且在 PWM 模式 1 中恢复的情况

(1) ~ (10) 和图 21.141 通用。

(11) 设定 PWM 模式 1。

(12) 必须通过 TIOR 寄存器对引脚进行初始化 (在 PWM 模式 1 中不将波形输出到 MTIOCnB (MTIOCnD) 引脚。如果有应该输出的电平, 必须通过 I/O 端口的端口方向寄存器 (PDR) 和端口输出数据寄存器 (PODR) 设定通用输出端口)。

(13) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(14) 通过 TSTR 寄存器重新开始。

(3) 在正常模式的运行过程中发生异常并且在 PWM 模式 2 中重新开始时的运行

在正常模式中发生异常并且重新设定后在 PWM 模式 2 中重新开始时的说明图如图 21.143 所示。

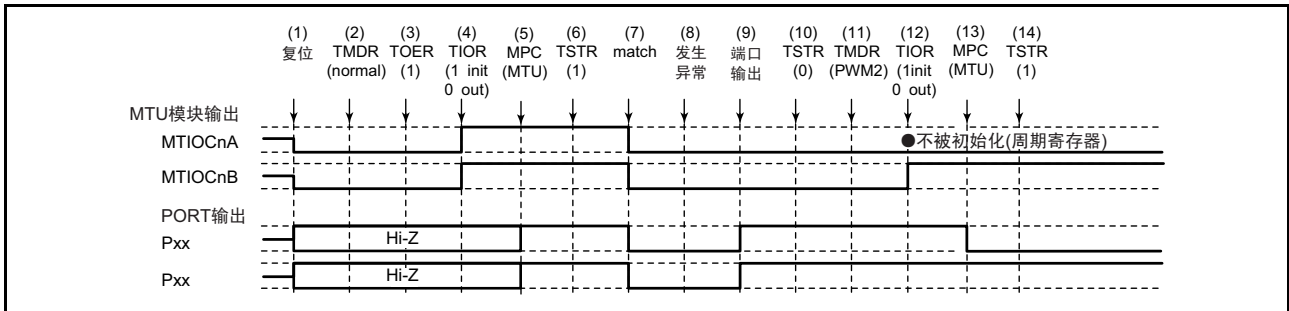


图 21.143 在正常模式中发生异常并且在 PWM 模式 2 中恢复的情况

(1) ~ (10) 和图 21.141 通用。

(11) 设定 PWM 模式 2。

(12) 必须通过 TIOR 寄存器对引脚进行初始化 (在 PWM 模式 2 中不将波形输出到周期寄存器的引脚。如果有应该输出的电平, 必须通过 I/O 端口的端口方向寄存器 (PDR) 和端口输出数据寄存器 (PODR) 设定通用输出端口)。

(13) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(14) 通过 TSTR 寄存器重新开始。

注. 只有 MTU0 ~ 2 能设定 PWM 模式 2, 因此不需要设定 TOERA 寄存器。

(4) 在正常模式的运行过程中发生异常并且在相位计数模式中重新开始时的运行

在正常模式中发生异常并且重新设定后在相位计数模式中重新开始时的说明图如图 21.144 所示。

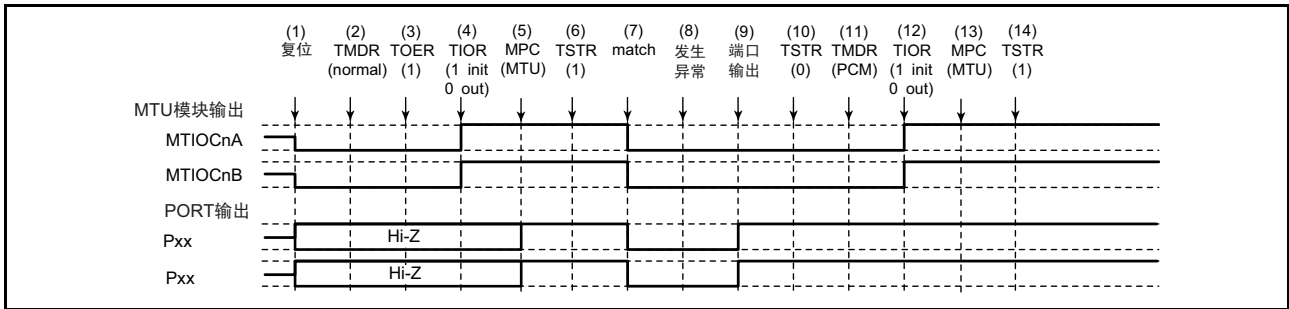


图 21.144 在正常模式中发生异常并且在相位计数模式中恢复的情况

(1) ~ (10) 和图 21.141 通用。

(11) 设定相位计数模式。

(12) 必须通过 TIOR 寄存器对引脚进行初始化。

(13) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(14) 通过 TSTRA 寄存器重新开始。

注. 只有 MTU1 和 MTU2 能设定相位计数模式, 因此不需要设定 TOERA 寄存器。

(5) 在正常模式的运行过程中发生异常并且在互补 PWM 模式中重新开始时的运行

在正常模式中发生异常并且重新设定后在互补 PWM 模式中重新开始时的说明图如图 21.145 所示。

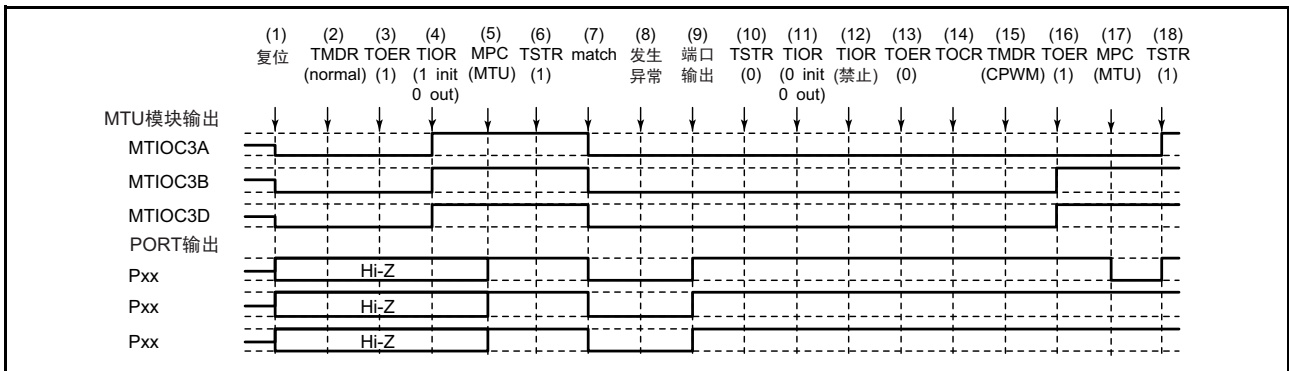


图 21.145 在正常模式中发生异常并且在互补 PWM 模式中恢复的情况

(1) ~ (10) 和图 21.141 通用。

(11) 必须通过 TIOR 寄存器对正常模式的波形生成部进行初始化。

(12) 必须通过 TIOR 寄存器禁止正常模式的波形生成部的运行。

(13) 必须通过 TOERA 寄存器禁止 MTU3 和 MTU4 的输出。

(14) 必须通过 TOCR1A、TOCR2A 寄存器选择互补 PWM 的输出电平以及允许或者禁止周期输出。

(15) 设定互补 PWM。

(16) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。

(17) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(18) 通过 TSTRA 寄存器重新开始。

(6) 在正常模式的运行过程中发生异常并且在复位同步 PWM 模式中重新开始时的运行

在正常模式中发生异常并且重新设定后在复位同步 PWM 模式中重新开始时的说明图如图 21.146 所示。

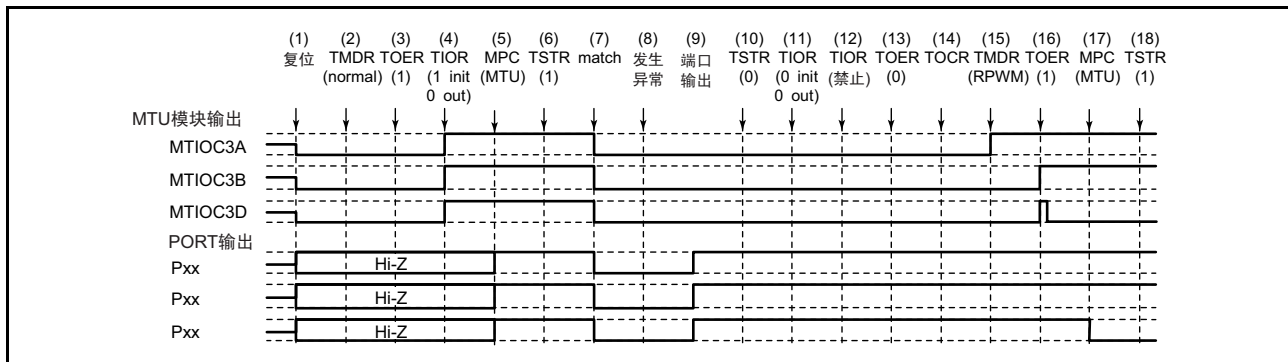


图 21.146 在正常模式中发生异常并且复位同步 PWM 模式中恢复的情况

(1) ~ (13) 和图 21.141 通用。

(14) 必须通过 TOCR1A、TOCR2A 寄存器选择复位同步 PWM 的输出电平以及允许或者禁止周期输出。

(15) 设定复位同步 PWM。

(16) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。

(17) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(18) 通过 TSTRA 寄存器重新开始。

(7) 在 PWM 模式 1 的运行过程中发生异常并且在正常模式中重新开始时的运行

在 PWM 模式 1 中发生异常并且重新设定后在正常模式中重新开始时的说明图如图 21.147 所示。

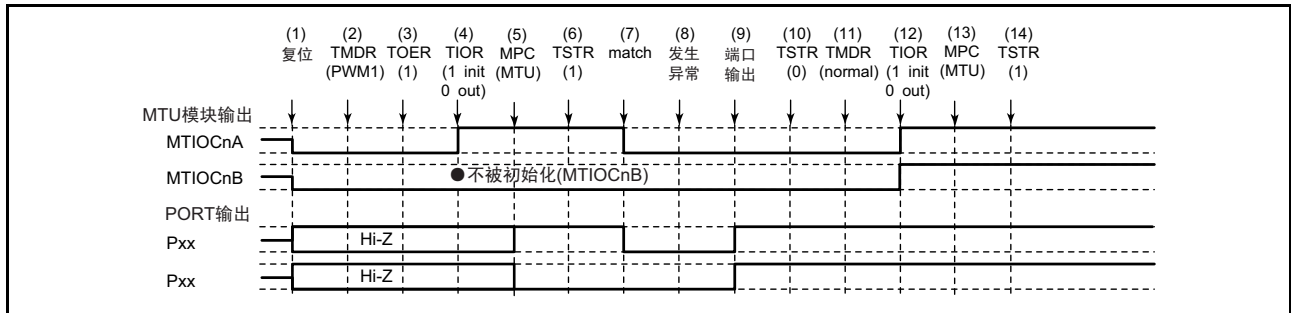


图 21.147 在 PWM 模式 1 中发生异常并且在正常模式中恢复的情况

- (1) 复位后，MTU 的输出为 Low 电平，端口为高阻抗。
- (2) 必须设定 PWM 模式 1。
- (3) 在 MTU3 和 MTU4 中，必须在通过 TIOR 寄存器对引脚进行初始化前用 TOERA 寄存器允许输出。
- (4) 必须通过 TIOR 寄存器对引脚进行初始化（这是初始输出为 High 电平并且在比较匹配时输出 Low 电平的例子。在 PWM 模式 1 中不对 MTIOcNB 进行初始化）。
- (5) 必须通过 MPC 和 I/O 端口的端口模式寄存器（PMR）将引脚置为 MTU 输出。
- (6) 通过 TSTR 寄存器开始计数运行。
- (7) 在比较匹配时输出 Low 电平。
- (8) 发生异常。
- (9) 必须通过 I/O 端口的端口方向寄存器（PDR）、端口输出数据寄存器（PODR）和端口模式寄存器（PMR）将引脚切换为通用输出端口，并且输出无效电平。
- (10) 通过 TSTR 寄存器停止计数运行。
- (11) 必须设定正常模式。
- (12) 必须通过 TIOR 寄存器对引脚进行初始化。
- (13) 必须通过 MPC 和 I/O 端口的端口模式寄存器（PMR）将引脚置为 MTU 输出。
- (14) 通过 TSTR 寄存器重新开始。

(8) 在 PWM 模式 1 的运行过程中发生异常并且在 PWM 模式 1 中重新开始时的运行

在 PWM 模式 1 中发生异常并且重新设定后在 PWM 模式 1 中重新开始时的说明图如图 21.148 所示。

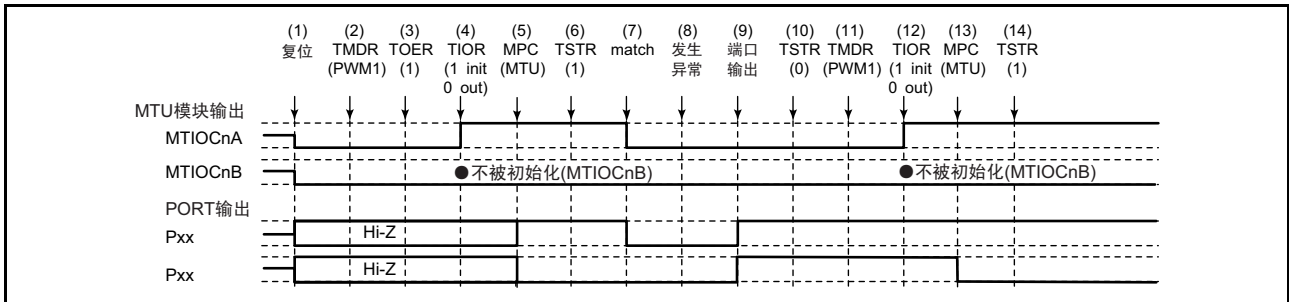


图 21.148 在 PWM 模式 1 中发生异常并且在 PWM 模式 1 中恢复的情况

(1) ~ (10) 和图 21.147 通用。

(11) 在 PWM 模式 1 中重新开始的情况下不需要此步骤。

(12) 必须通过 TIOR 寄存器对引脚进行初始化（在 PWM 模式 1 中不将波形输出到 MTIOCnB (MTIOCnD) 引脚。如果有应该输出的电平，必须通过 I/O 端口的端口方向寄存器 (PDR) 和端口输出数据寄存器 (PODR) 设定通用输出端口)。

(13) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(14) 通过 TSTR 寄存器重新开始。

(9) 在 PWM 模式 1 的运行过程中发生异常并且在 PWM 模式 2 中重新开始时的运行

在 PWM 模式 1 中发生异常并且重新设定后在 PWM 模式 2 中重新开始时的说明图如图 21.149 所示。

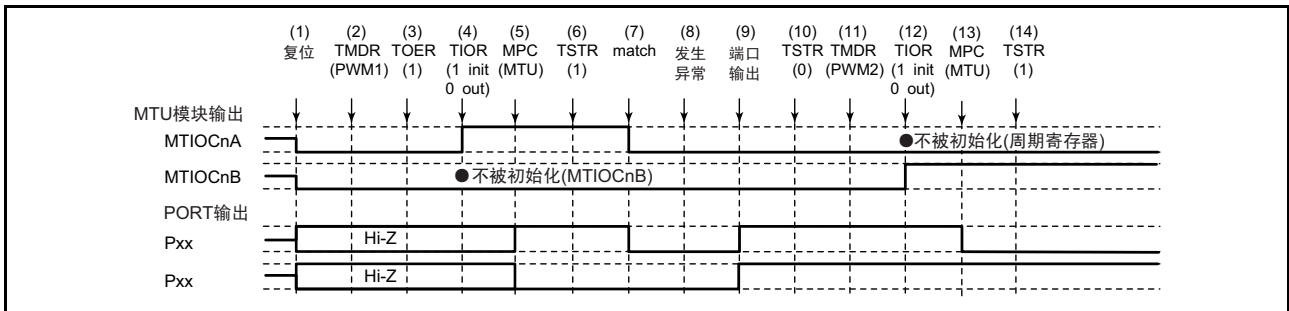


图 21.149 在 PWM 模式 1 中发生异常并且在 PWM 模式 2 中恢复的情况

(1) ~ (10) 和图 21.147 通用。

(11) 设定 PWM 模式 2。

(12) 必须通过 TIOR 寄存器对引脚进行初始化（在 PWM 模式 2 中不将波形输出到周期寄存器的引脚。如果有应该输出的电平，必须通过 I/O 端口的端口方向寄存器 (PDR) 和端口输出数据寄存器 (PODR) 设定通用输出端口)。

(13) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(14) 通过 TSTR 寄存器重新开始。

注. 只有 MTU0 ~ 2 能设定 PWM 模式 2，因此不需要设定 TOERA 寄存器。

(10) 在 PWM 模式 1 的运行过程中发生异常并且在相位计数模式中重新开始时的运行

在 PWM 模式 1 中发生异常并且重新设定后在相位计数模式中重新开始时的说明图如图 21.150 所示。

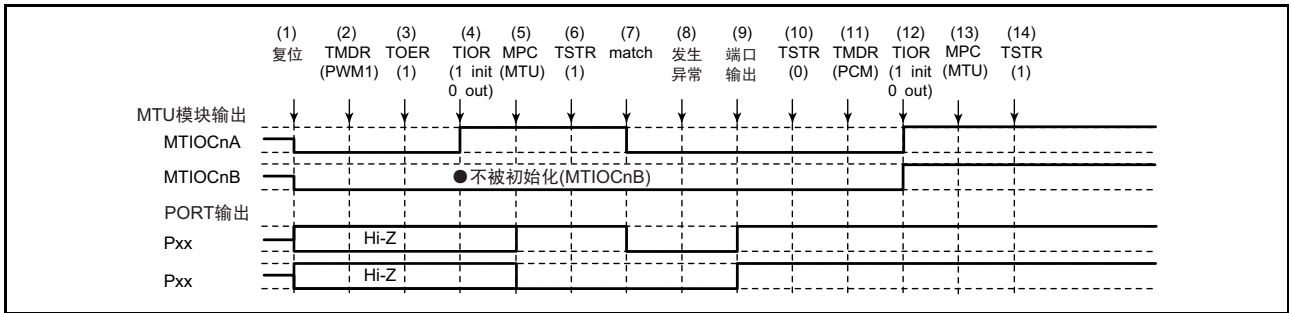


图 21.150 在 PWM 模式 1 中发生异常并且在相位计数模式中恢复的情况

(1) ~ (10) 和图 21.147 通用。

- (11) 设定相位计数模式。
- (12) 必须通过 TIOR 寄存器对引脚进行初始化。
- (13) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。
- (14) 通过 TSTR 寄存器重新开始。

注. 只有 MTU1 和 MTU2 能设定相位计数模式, 因此不需要设定 TOERA 寄存器。

(11) 在 PWM 模式 1 的运行过程中发生异常并且在互补 PWM 模式中重新开始时的运行

在 PWM 模式 1 中发生异常并且重新设定后在互补 PWM 模式中重新开始时的说明图如图 21.151 所示。

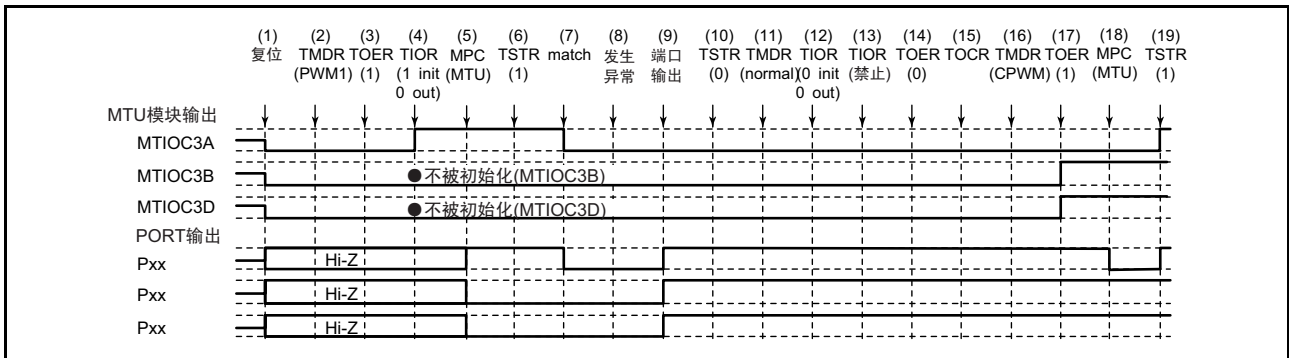


图 21.151 在 PWM 模式 1 中发生异常并且在互补 PWM 模式中恢复的情况

(1) ~ (10) 和图 21.147 通用。

- (11) 为了对波形生成部进行初始化, 必须设定正常模式。
- (12) 必须通过 TIOR 寄存器对 PWM 模式 1 的波形生成部进行初始化。
- (13) 必须通过 TIOR 寄存器禁止 PWM 模式 1 的波形生成部的运行。
- (14) 必须通过 TOERA 寄存器禁止 MTU3 和 MTU4 的输出。
- (15) 必须通过 TOCR1A、TOCR2A 寄存器选择互补 PWM 的输出电平以及允许或者禁止周期输出。
- (16) 设定互补 PWM。
- (17) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。
- (18) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。
- (19) 通过 TSTR 寄存器重新开始。

(12) 在 PWM 模式 1 的运行过程中发生异常并且在复位同步 PWM 模式中重新开始时的运行

在 PWM 模式 1 中发生异常并且重新设定后在复位同步 PWM 模式中重新开始时的说明图如图 21.152 所示。

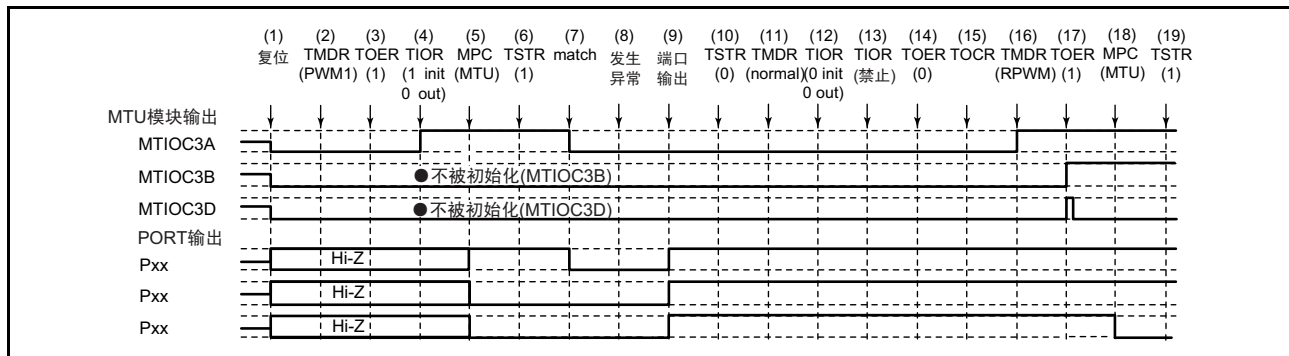


图 21.152 在 PWM 模式 1 中发生异常并且在复位同步 PWM 模式中恢复的情况

(1) ~ (14) 和图 21.151 通用。

(15) 必须通过 TOCR1A、TOCR2A 寄存器选择复位同步 PWM 的输出电平以及允许或者禁止周期输出。

(16) 设定复位同步 PWM。

(17) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。

(18) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(19) 通过 TSTRA 寄存器重新开始。

(13) 在 PWM 模式 2 的运行过程中发生异常并且在正常模式中重新开始时的运行

在 PWM 模式 2 中发生异常并且重新设定后在正常模式中重新开始时的说明图如图 21.153 所示。

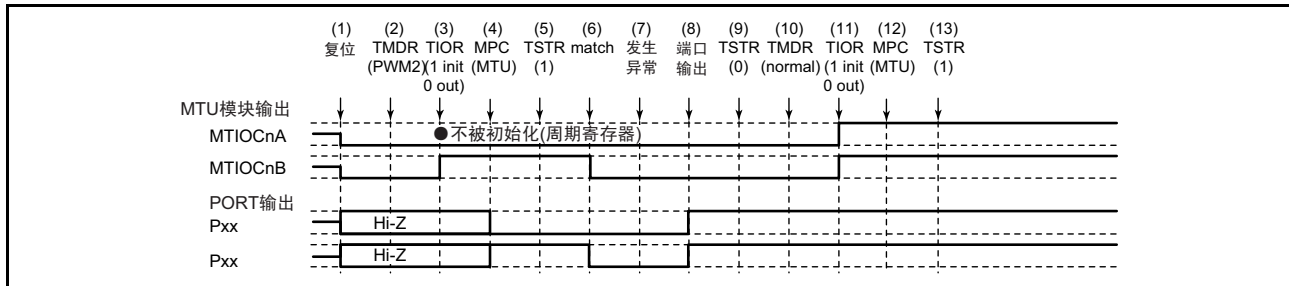


图 21.153 在 PWM 模式 2 中发生异常并且在正常模式中恢复的情况

- (1) 复位后，MTU 的输出为 Low 电平，端口为高阻抗。
- (2) 必须设定 PWM 模式 2。
- (3) 必须通过 TIOR 寄存器对引脚进行初始化（这是初始输出为 High 电平并且在比较匹配输出时 Low 电平的例子。在 PWM 模式 2 中不对周期寄存器的引脚进行初始化，这是 MTIOcNA 为周期寄存器的例子）。
- (4) 必须通过 MPC 和 I/O 端口的端口模式寄存器（PMR）将引脚置为 MTU 输出。
- (5) 通过 TSTR 寄存器开始计数运行。
- (6) 在比较匹配时输出 Low 电平。
- (7) 发生异常。
- (8) 必须通过 I/O 端口的端口方向寄存器（PDR）、端口输出数据寄存器（PODR）和端口模式寄存器（PMR）将引脚切换为通用输出端口，并且输出无效电平。
- (9) 通过 TSTR 寄存器停止计数运行。
- (10) 必须设定正常模式。
- (11) 必须通过 TIOR 寄存器对引脚进行初始化。
- (12) 必须通过 MPC 和 I/O 端口的端口模式寄存器（PMR）将引脚置为 MTU 输出。
- (13) 通过 TSTR 寄存器重新开始。

(14) 在 PWM 模式 2 的运行过程中发生异常并且在 PWM 模式 1 中重新开始时的运行

在 PWM 模式 2 中发生异常并且重新设定后在 PWM 模式 1 中重新开始时的说明图如图 21.154 所示。

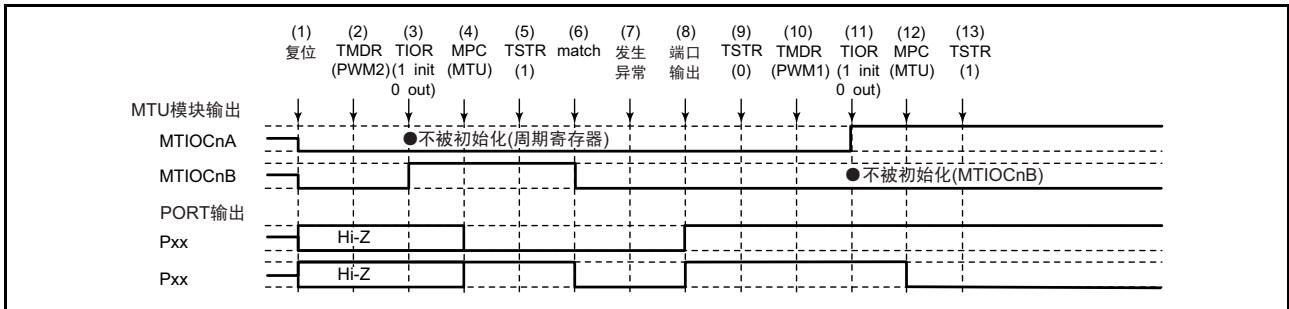


图 21.154 在 PWM 模式 2 中发生异常并且在 PWM 模式 1 中恢复的情况

(1) ~ (9) 和图 21.153 通用。

(10) 设定 PWM 模式 1。

(11) 必须通过 TIOR 寄存器对引脚进行初始化 (在 PWM 模式 1 中不将波形输出到 MTIOcNB (MTIOcND) 引脚。如果有应该输出的电平, 必须通过 I/O 端口的端口方向寄存器 (PDR) 和端口输出数据寄存器 (PODR) 设定通用输出端口)。

(12) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(13) 通过 TSTR 寄存器重新开始。

(15) 在 PWM 模式 2 的运行过程中发生异常并且在 PWM 模式 2 中重新开始时的运行

在 PWM 模式 2 中发生异常并且重新设定后在 PWM 模式 2 中重新开始时的说明图如图 21.155 所示。

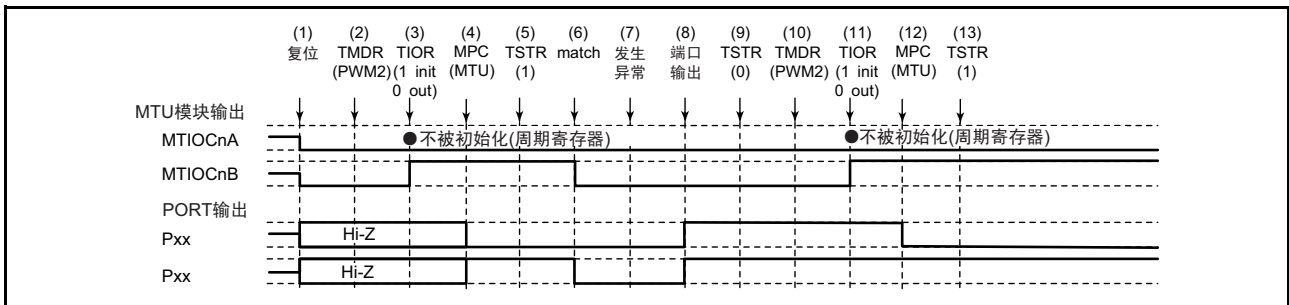


图 21.155 在 PWM 模式 2 中发生异常并且在 PWM 模式 2 中恢复的情况

(1) ~ (9) 和图 21.153 通用。

(10) 在 PWM 模式 2 中重新开始的情况下不需要此步骤。

(11) 必须通过 TIOR 寄存器对引脚进行初始化 (在 PWM 模式 2 中不将波形输出到周期寄存器的引脚。如果有应该输出的电平, 必须通过 I/O 端口的端口方向寄存器 (PDR) 和端口输出数据寄存器 (PODR) 设定通用输出端口)。

(12) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(13) 通过 TSTR 寄存器重新开始。

(16) 在 PWM 模式 2 的运行过程中发生异常并且在相位计数模式中重新开始时的运行

在 PWM 模式 2 中发生异常并且重新设定后在相位计数模式中重新开始时的说明图如图 21.156 所示。

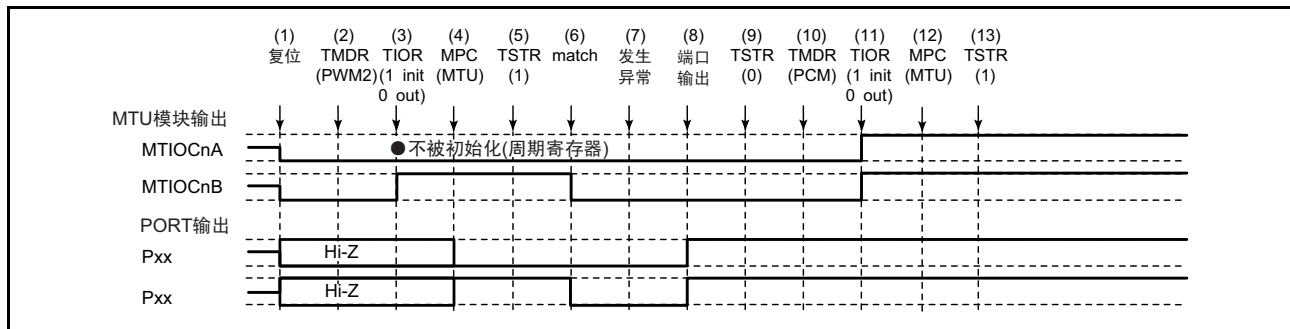


图 21.156 在 PWM 模式 2 中发生异常并且在相位计数模式中恢复的情况

(1) ~ (9) 和图 21.153 通用。

(10) 设定相位计数模式。

(11) 必须通过 TIOR 寄存器对引脚进行初始化。

(12) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(13) 通过 TSTR 寄存器重新开始。

(17) 在相位计数模式的运行过程中发生异常并且在正常模式中重新开始时的运行

在相位计数模式中发生异常并且重新设定后在正常模式中重新开始时的说明图如图 21.157 所示。

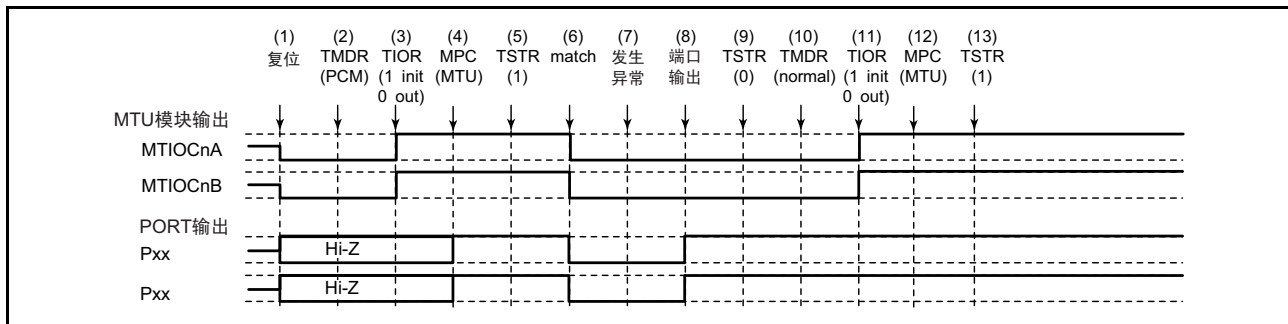


图 21.157 在相位计数模式中发生异常并且在正常模式中恢复的情况

- (1) 复位后，MTU 的输出为 Low 电平，端口为高阻抗。
- (2) 必须设定相位计数模式。
- (3) 必须通过 TIOR 寄存器对引脚进行初始化（这是初始输出为 High 电平并且在比较匹配时输出 Low 电平的例子）。
- (4) 必须通过 MPC 和 I/O 端口的端口模式寄存器（PMR）将引脚置为 MTU 输出。
- (5) 通过 TSTR 寄存器开始计数运行。
- (6) 在比较匹配时输出 Low 电平。
- (7) 发生异常。
- (8) 必须通过 I/O 端口的端口方向寄存器（PDR）、端口输出数据寄存器（PODR）和端口模式寄存器（PMR）将引脚切换为通用输出端口，并且输出无效电平。
- (9) 通过 TSTR 寄存器停止计数运行。
- (10) 必须在正常模式中进行设定。
- (11) 必须通过 TIOR 寄存器对引脚进行初始化。
- (12) 必须通过 MPC 和 I/O 端口的端口模式寄存器（PMR）将引脚置为 MTU 输出。
- (13) 通过 TSTR 寄存器重新开始。

(18) 在相位计数模式的运行过程中发生异常并且在 PWM 模式 1 中重新开始时的运行

在相位计数模式中发生异常并且重新设定后在 PWM 模式 1 中重新开始时的说明图如图 21.158 所示。

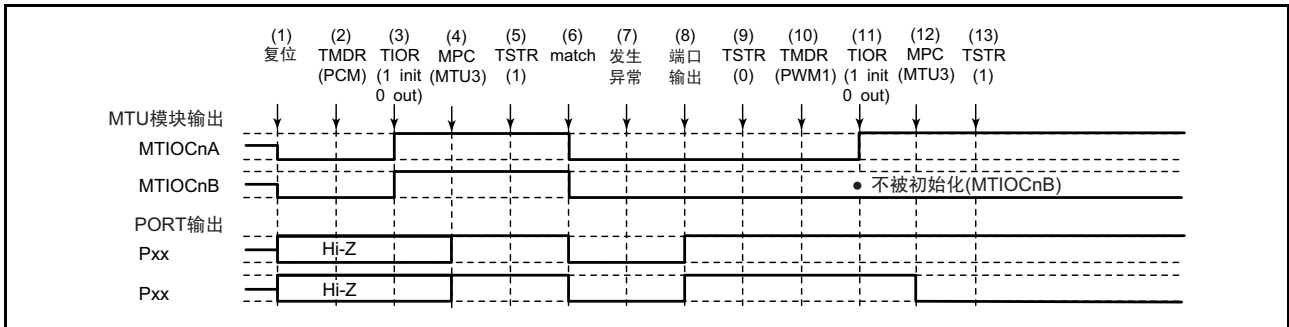


图 21.158 在相位计数模式中发生异常并且在 PWM 模式 1 中恢复的情况

(1) ~ (9) 和图 21.157 通用。

(10) 设定 PWM 模式 1。

(11) 必须通过 TIOR 寄存器对引脚进行初始化（在 PWM 模式 1 中不将波形输出到 MTIOcNB (MTIOcND) 引脚。如果有应该输出的电平，必须通过 I/O 端口的端口方向寄存器 (PDR) 和端口输出数据寄存器 (PODR) 设定通用输出端口)。

(12) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(13) 通过 TSTR 寄存器重新开始。

(19) 在相位计数模式的运行过程中发生异常并且在 PWM 模式 2 中重新开始时的运行

在相位计数模式中发生异常并且重新设定后在 PWM 模式 2 中重新开始时的说明图如图 21.159 所示。

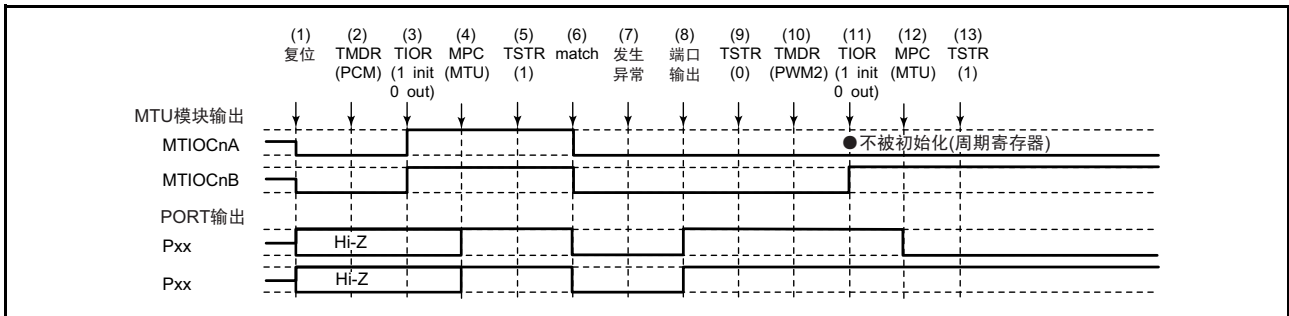


图 21.159 在相位计数模式中发生异常并且在 PWM 模式 2 中恢复的情况

(1) ~ (9) 和图 21.157 通用。

(10) 设定 PWM 模式 2。

(11) 必须通过 TIOR 寄存器对引脚进行初始化（在 PWM 模式 2 中不将波形输出到周期寄存器的引脚。如果有应该输出的电平，必须通过 I/O 端口的端口方向寄存器 (PDR) 和端口输出数据寄存器 (PODR) 设定通用输出端口)。

(12) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(13) 通过 TSTR 寄存器重新开始。

(20) 在相位计数模式的运行过程中发生异常并且在相位计数模式中重新开始时的运行

在相位计数模式中发生异常并且重新设定后在相位计数模式中重新开始时的说明图如图 21.160 所示。

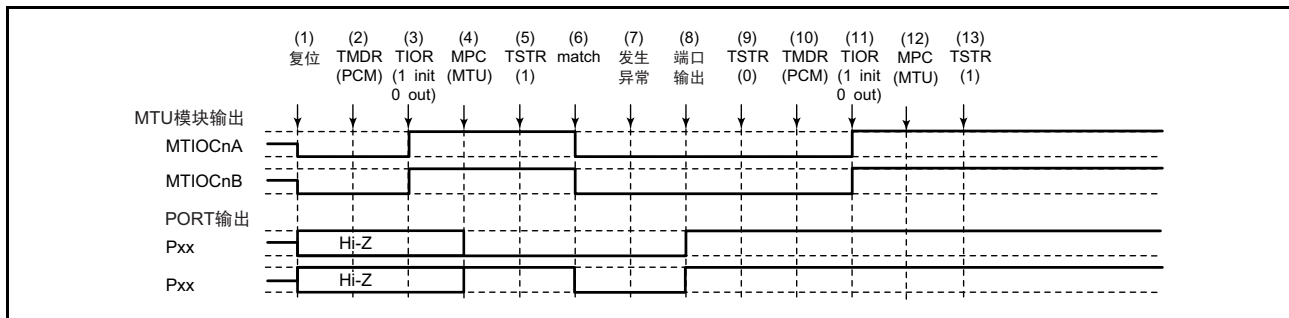


图 21.160 在相位计数模式中发生异常并且在相位计数模式中恢复的情况

(1) ~ (9) 和图 21.157 通用。

(10) 在相位计数模式重新开始的情况下不需要此步骤。

(11) 必须通过 TIOR 寄存器对引脚进行初始化。

(12) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(13) 通过 TSTR 寄存器重新开始。

(21) 在互补 PWM 模式的运行过程中发生异常并且在正常模式中重新开始时的运行

在互补 PWM 模式中发生异常并且重新设定后在正常模式中重新开始的说明图如图 21.161 所示。

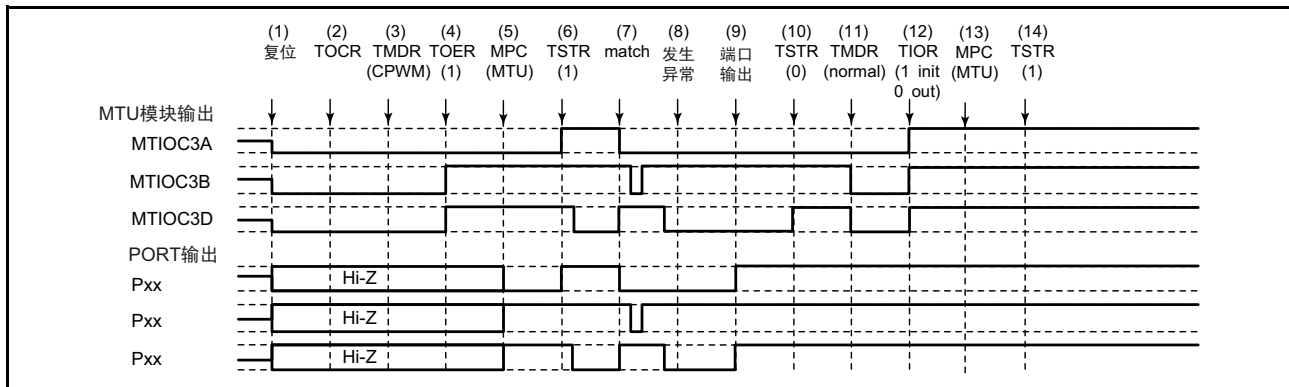


图 21.161 在互补 PWM 模式中发生异常并且在正常模式中恢复的情况

- (1) 复位后，MTU 的输出为 Low 电平，端口为高阻抗。
- (2) 必须通过 TOCR1A、TOCR2A 寄存器选择互补 PWM 的输出电平以及允许或者禁止周期输出。
- (3) 设定互补 PWM。
- (4) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。
- (5) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。
- (6) 通过 TSTRA 寄存器开始计数运行。
- (7) 通过发生比较匹配输出互补 PWM 波形。
- (8) 发生异常。
- (9) 必须通过 I/O 端口的端口方向寄存器 (PDR)、端口输出数据寄存器 (PODR) 和端口模式寄存器 (PMR) 将引脚切换为通用输出端口，并且输出无效电平。
- (10) 通过 TSTRA 寄存器停止计数运行 (MTU 输出为互补 PWM 输出的初始值)。
- (11) 必须设定正常模式 (MTU 的输出为 Low 电平)。
- (12) 必须通过 TIOR 寄存器对引脚进行初始化。
- (13) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。
- (14) 通过 TSTRA 寄存器重新开始。

(22) 在互补 PWM 模式的运行过程中发生异常并且在 PWM 模式 1 中重新开始时的运行

在互补 PWM 模式中发生异常并且重新设定后在 PWM 模式 1 中重新开始时的说明图如图 21.162 所示。

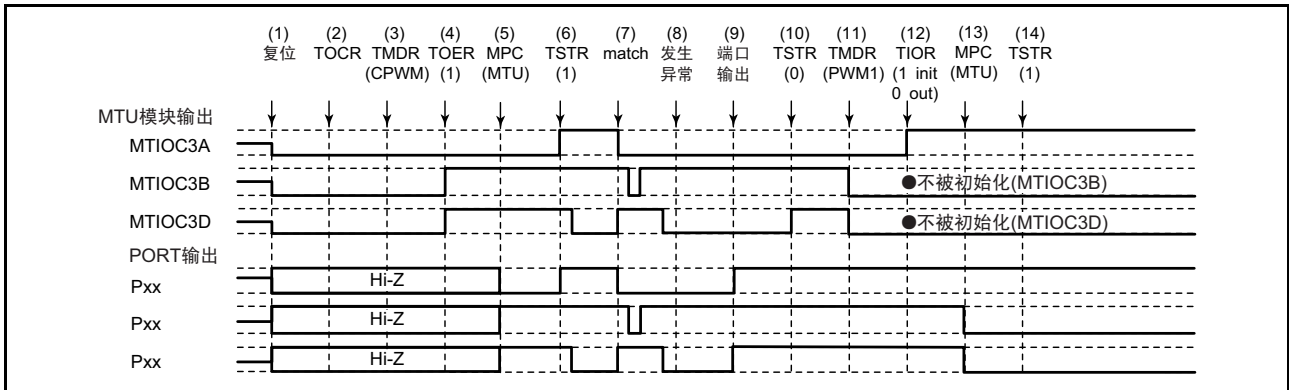


图 21.162 在互补 PWM 模式中发生异常并且在 PWM 模式 1 中恢复的情况

(1) ~ (10) 和图 21.161 通用。

(11) 必须设定 PWM 模式 1 (MTU 的输出为 Low 电平)。

(12) 必须通过 TIOR 寄存器对引脚进行初始化 (在 PWM 模式 1 中不将波形输出到 MTIOCnB (MTIOCnD) 引脚。如果有应该输出的电平, 必须通过 I/O 端口的端口方向寄存器 (PDR) 和端口输出数据寄存器 (PODR) 设定通用输出端口)。

(13) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(14) 通过 TSTR 寄存器重新开始。

(23) 在互补 PWM 模式的运行过程中发生异常并且在互补 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常并且重新设定后在互补 PWM 模式中重新开始时的说明图如图 21.163 所示 (从停止对周期和占空比进行计数时的值重新开始的情况)。

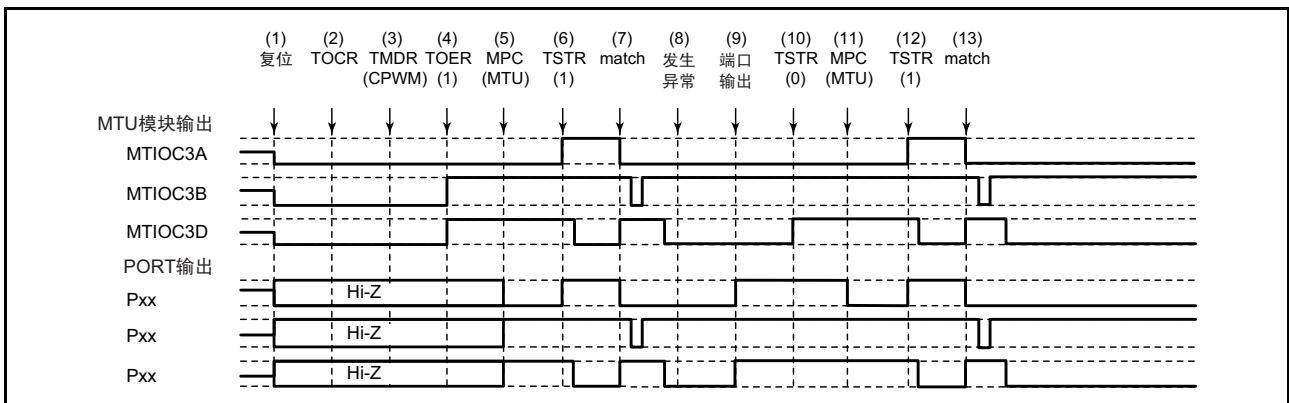


图 21.163 在互补 PWM 模式中发生异常并且在互补 PWM 模式中恢复的情况 (1)

(1) ~ (10) 和图 21.161 通用。

(11) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(12) 通过 TSTR 寄存器重新开始。

(13) 通过发生比较匹配输出互补 PWM 波形。

(24) 在互补 PWM 模式的运行过程中发生异常并且在互补 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常并且重新设定后在互补 PWM 模式中重新开始时的说明图如图 21.164 所示 (从重新设定的周期和占空比的值重新开始的情况)。

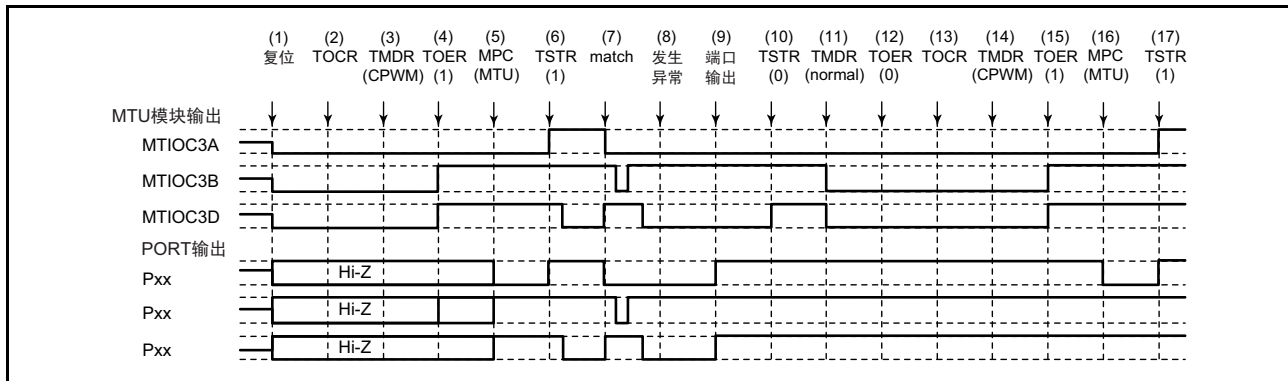


图 21.164 在互补 PWM 模式中发生异常并且在互补 PWM 模式中恢复的情况 (2)

(1) ~ (10) 和图 21.161 通用。

(11) 必须设定正常模式并且设定新的设定值 (MTU的输出为Low电平)。

(12) 必须通过TOERA寄存器禁止MTU3和MTU4的输出。

(13) 必须通过TOCR1A、TOCR2A寄存器选择互补PWM模式的输出电平以及允许或者禁止周期输出。

(14) 设定互补PWM。

(15) 必须通过TOERA寄存器允许MTU3和MTU4的输出。

(16) 必须通过MPC和I/O端口的端口模式寄存器 (PMR) 将引脚置为MTU输出。

(17) 通过TSTRA寄存器重新开始。

(25) 在互补 PWM 模式的运行过程中发生异常并且在复位同步 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常并且重新设定后在复位同步 PWM 模式中重新开始时的说明图如图 21.165 所示。

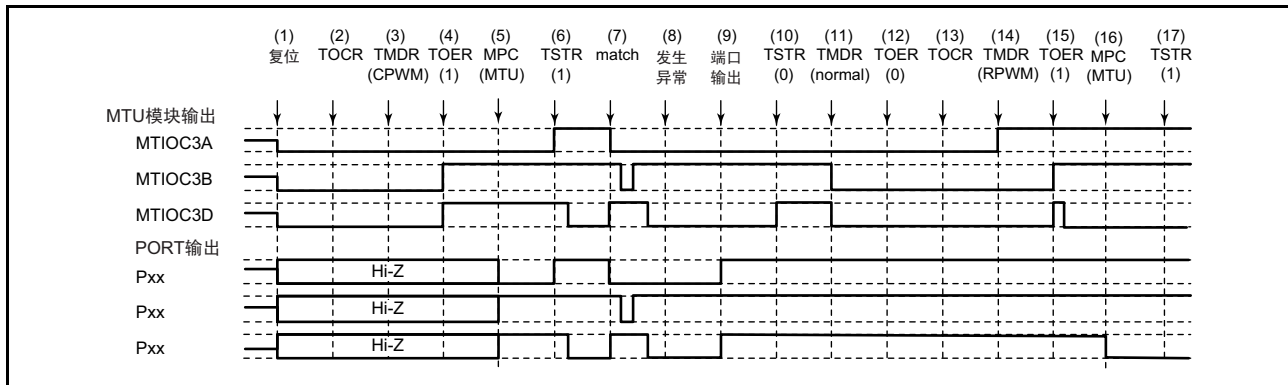


图 21.165 在互补 PWM 模式中发生异常并且在复位同步 PWM 模式中恢复的情况

(1) ~ (10) 和图 21.161 通用。

(11) 必须设定正常模式 (MTU 的输出为 Low 电平)。

(12) 必须通过 TOERA 寄存器禁止 MTU3 和 MTU4 的输出。

(13) 必须通过 TOCR1A、TOCR2A 寄存器选择复位同步 PWM 模式的输出电平以及允许或者禁止周期输出。

(14) 设定复位同步 PWM。

(15) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。

(16) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(17) 通过 TSTRA 寄存器重新开始。

(26) 在复位同步 PWM 模式的运行过程中发生异常并且在正常模式中重新开始时的运行

在复位同步 PWM 模式中发生异常并且重新设定后在正常模式中重新开始时的说明图如图 21.166 所示。

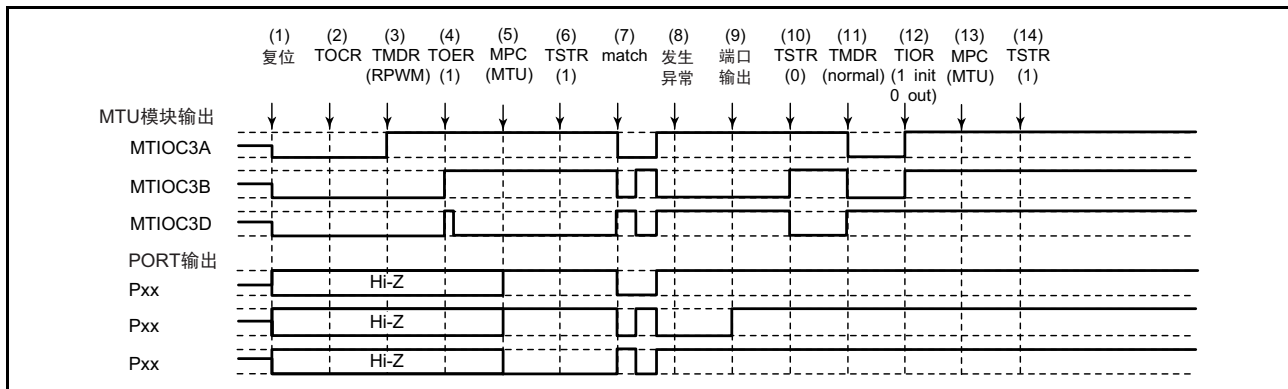


图 21.166 在复位同步 PWM 模式中发生异常并且在正常模式中恢复的情况

- (1) 复位后，MTU 的输出为 Low 电平，端口为高阻抗。
- (2) 必须通过 TOCR1A、TOCR2A 寄存器选择复位同步 PWM 的输出电平以及允许或者禁止周期输出。
- (3) 设定复位同步 PWM。
- (4) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。
- (5) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。
- (6) 通过 TSTRA 寄存器开始计数运行。
- (7) 通过发生比较匹配输出复位同步 PWM 波形。
- (8) 发生异常。
- (9) 必须通过 I/O 端口的端口方向寄存器 (PDR)、端口输出数据寄存器 (PODR) 和端口模式寄存器 (PMR) 将引脚切换为通用输出端口，并且输出无效电平。
- (10) 通过 TSTRA 寄存器停止计数运行 (MTU 的输出为复位同步 PWM 输出的初始值)。
- (11) 必须设定正常模式 (MTU 输出的正相侧为 Low 电平，反相侧为 High 电平)。
- (12) 必须通过 TIOR 寄存器对引脚进行初始化。
- (13) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。
- (14) 通过 TSTRA 寄存器重新开始。

(27) 在复位同步 PWM 模式的运行过程中发生异常并且在 PWM 模式 1 中重新开始时的运行

在复位同步 PWM 模式中发生异常并且重新设定后在 PWM 模式 1 中重新开始时的说明图如图 21.167 所示。

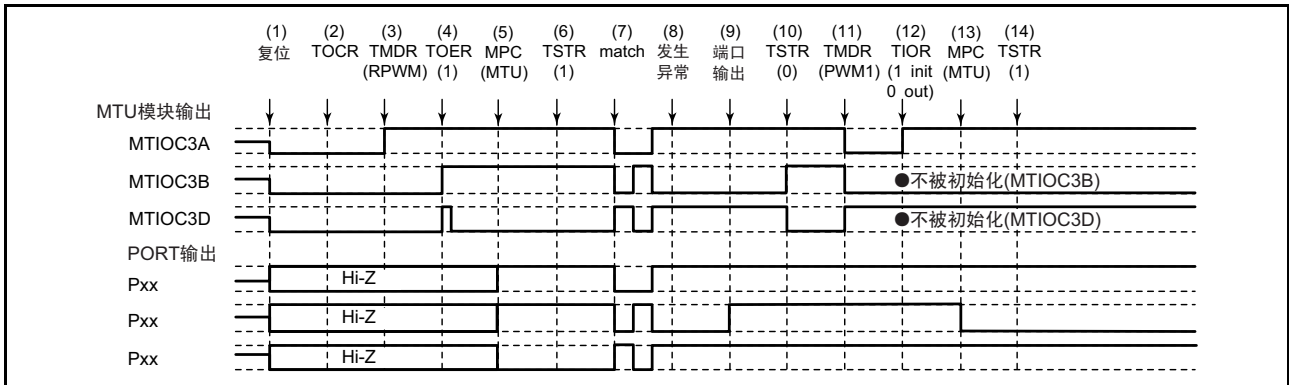


图 21.167 在复位同步 PWM 模式中发生异常并且在 PWM 模式 1 中恢复的情况

(1) ~ (10) 和图 21.166 通用。

(11) 必须设定 PWM 模式 1 (MTU 输出的正相侧为 Low 电平, 反相侧为 High 电平)。

(12) 必须通过 TIOR 寄存器对引脚进行初始化 (在 PWM 模式 1 中不将波形输出到 MTIOCnB (MTIOCnD) 引脚。如果有应该输出的电平, 必须通过 I/O 端口的端口方向寄存器 (PDR) 和端口输出数据寄存器 (PODR) 设定通用输出端口)。

(13) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(14) 通过 TSTR 寄存器重新开始。

(28) 在复位同步 PWM 模式的运行过程中发生异常并且在互补 PWM 模式中重新开始时的运行

在复位同步 PWM 模式中发生异常并且重新设定后在互补 PWM 模式中重新开始时的说明图如图 21.168 所示。

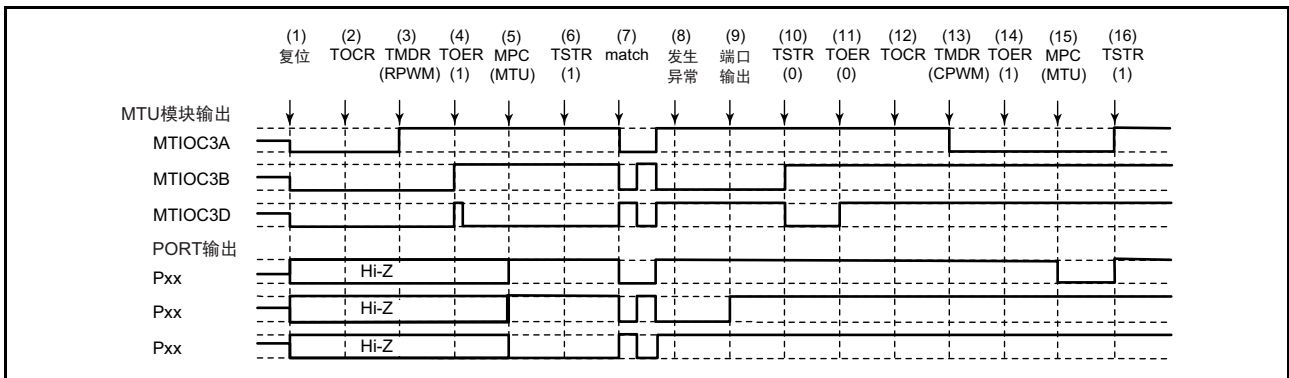


图 21.168 在复位同步 PWM 模式中发生异常并且在互补 PWM 模式中恢复的情况

(1) ~ (10) 和图 21.166 通用。

(11) 必须通过 TOERA 寄存器禁止 MTU3 和 MTU4 的输出。

(12) 必须通过 TOCR1A、TOCR2A 寄存器选择互补 PWM 的输出电平以及允许或者禁止周期输出。

(13) 设定互补 PWM (MTU 的周期输出引脚为 Low 电平)。

(14) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。

(15) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(16) 通过 TSTR 寄存器重新开始。

(29) 在复位同步 PWM 模式的运行过程中发生异常并且在复位同步 PWM 模式中重新开始时的运行

在复位同步 PWM 模式中发生异常并且重新设定后在复位同步 PWM 模式中重新开始时的说明图如图 21.169 所示。

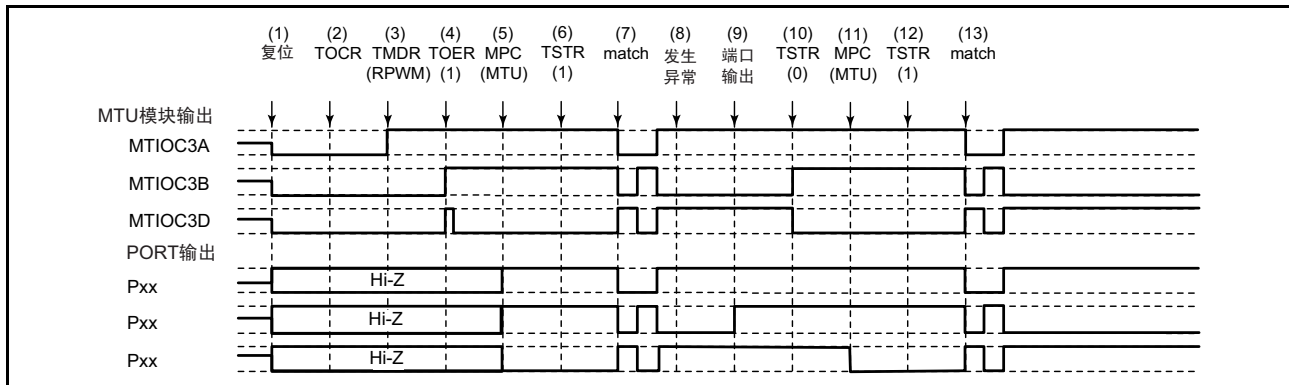


图 21.169 在复位同步 PWM 模式中发生异常并且在复位同步 PWM 模式中恢复的情况

(1) ~ (10) 和图 21.166 通用。

(11) 必须通过 MPC 和 I/O 端口的端口模式寄存器 (PMR) 将引脚置为 MTU 输出。

(12) 通过 TSTR 寄存器重新开始。

(13) 通过发生比较匹配输出复位同步 PWM 波形。

22. 端口输出允许 3 (POE3)

根据 POE0#、POE8#、POE10#、POE11# 引脚的输入变化、MTU 的互补 PWM 输出引脚 (MTIOC3B、MTIOC3D、MTIOC4A、MTIOC4B、MTIOC4C 和 MTIOC4D 或者 MTIOC6B、MTIOC6D、MTIOC7A、MTIOC7B、MTIOC7C 和 MTIOC7D) 的输出状态、GPT 输出引脚 (GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B、GTIOC2A、GTIOC2B) 的输出状态或者寄存器的设定，端口输出允许 3 (POE3) 能将 MTU 的互补 PWM 输出引脚以及 MTU0 引脚 (MTIOC0A、MTIOC0B、MTIOC0C、MTIOC0D)、GPT 引脚 (GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B、GTIOC2A、GTIOC2B、GTIOC3A、GTIOC3B) 中为多路复用的引脚置为高阻抗状态，同时还能发行中断请求。即使检测到时钟发生电路停止振荡，也能将 MTU 的互补 PWM 输出引脚以及 MTU0 引脚、GPT 引脚中为多路复用的引脚置为高阻抗状态。

22.1 概要

POE3 的规格和框图分别如表 22.1 和图 22.1 所示。

表 22.1 POE3 的规格

项目	内容
功能	<ul style="list-style-type: none"> • 能给 POE0#、POE8#、POE10# 和 POE11# 的各输入引脚设定下降沿、PCLK/8×16 次、PCLK/16×16 次或者 PCLK/128×16 次的 Low 电平采样。 • 能通过 POE0#、POE8#、POE10# 和 POE11# 引脚的下降沿或者 Low 电平采样，将 MTU 的互补 PWM 输出引脚以及 MTU0 引脚、GPT 引脚置为高阻抗状态。 • 当检测到时钟发生电路停止振荡时，能将 MTU 的互补 PWM 输出引脚以及 MTU0 引脚、GPT 引脚置为高阻抗状态。 • 如果在对 MTU 的互补 PWM 输出引脚或者 GPT 输出引脚的输出电平进行比较的同时，有效电平的输出至少持续输出了 1 个周期，就能将 MTU 的互补 PWM 输出引脚或者 GPT 输出引脚置为高阻抗状态。 • 能通过 12 位 A/D 转换器 (S12ADB) 的比较器检测，将 MTU 的互补 PWM 输出引脚以及 MTU0 引脚、GPT 引脚置为高阻抗状态。 • 能通过设定 POE3 的寄存器，将 MTU 的互补 PWM 输出引脚以及 MTU0 引脚、GPT 引脚置为高阻抗状态。 • 根据输入电平的采样或者输出电平的比较结果，可能会发生各种中断。

如图 22.1 的框图所示，POE3 由输入电平的检测电路、输出电平的比较电路和高阻抗请求 / 中断请求的生成电路构成。

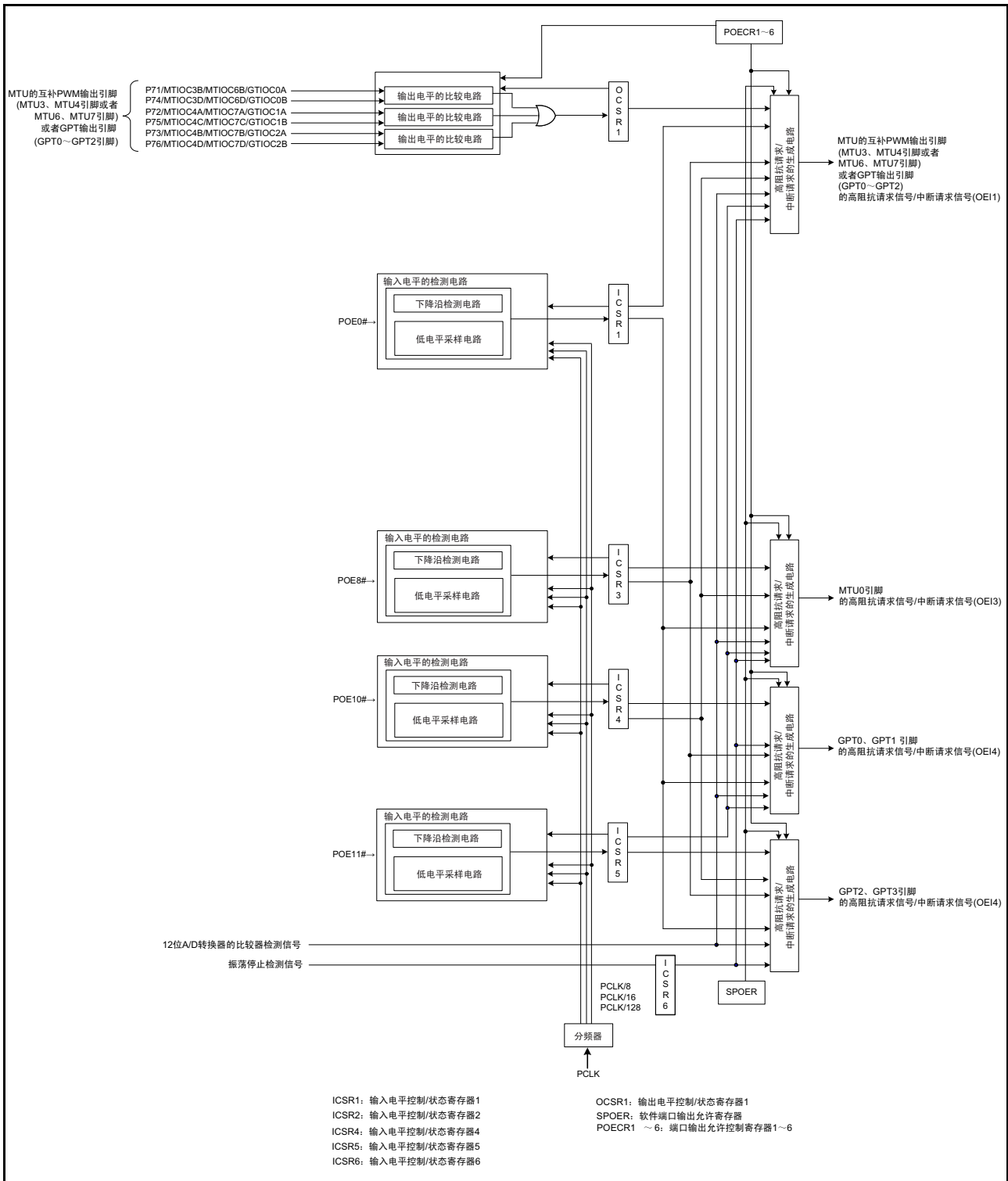


图 22.1 POE3 的框图

POE3 使用的输入 / 输出引脚如表 22.2 所示。

表 22.2 POE3 的输入 / 输出引脚

引脚名	输入 / 输出	功能
POE0#	输入	将 MTU 的互补 PWM 输出引脚 (MTU3、MTU4 引脚或者 MTU6、MTU7 引脚) 置为高阻抗状态的请求信号。 也能通过设定寄存器将 MTU0 引脚、GPT 引脚置为高阻抗状态。
POE8#	输入	将 MTU0 的引脚置为高阻抗状态的请求信号。 也能通过设定寄存器将 MTU 的互补 PWM 输出引脚 (MTU3、MTU4 引脚或者 MTU6、MTU7 引脚)、GPT 引脚置为高阻抗状态。
POE10#	输入	将 GPT0、GPT1 的引脚置为高阻抗状态的请求信号。 也能通过设定寄存器将 MTU 的互补 PWM 输出引脚 (MTU3、MTU4 引脚或者 MTU6、MTU7 引脚)、MTU0 引脚、GPT2 引脚、GPT3 引脚置为高阻抗状态。
POE10#	输入	将 GPT2、GPT3 的引脚置为高阻抗状态的请求信号。 也能通过设定寄存器将 MTU 的互补 PWM 输出引脚 (MTU3、MTU4 引脚或者 MTU6、MTU7 引脚)、MTU0 引脚、GPT0 引脚、GPT1 引脚置为高阻抗状态。

通过表 22.3 所示的引脚组合进行输出电平的比较。

表 22.3 引脚组合

引脚组合	输入 / 输出	功能
MTIOC3B 和 MTIOC3D	输出	当 2 个引脚的有效电平 (在 MTUn.TOCR1A.TOCS 位为“0”的情况下, MTUn.TOCR1A.OLSP 位是“0”时为 Low 电平输出, 是“1”时为 High 电平输出。或者在 MTUn.TOCR1A.TOCS 位为“1”的情况下, MTUn.TOCR2A.OLS3N、MTUn.TOCR2A.OLS3P、MTUn.TOCR2A.OLS2N、MTUn.TOCR2A.OLS2P、MTUn.TOCR2A.OLS1N、MTUn.TOCR2A.OLS1P 位是“0”时为 Low 电平输出, 是“1”时为 High 电平输出) 至少同时持续输出了 1 个外围时钟 (PCLK) 周期时, 能将 MTU 的互补 PWM 输出引脚 (MTU3、MTU4 引脚) 置为高阻抗状态。能通过 POE3 的寄存器设定对哪个组合进行输出比较并且进行高阻抗控制。
MTIOC4A 和 MTIOC4C	输出	
MTIOC4B 和 MTIOC4D	输出	
MTIOC6B 和 MTIOC6D	输出	当 2 个引脚的有效电平 (在 MTUn.TOCR1B.TOCS 位为“0”的情况下, MTUn.TOCR1B.OLSP 位是“0”时为 Low 电平输出, 是“1”时为 High 电平输出。或者在 MTUn.TOCR1B.TOCS 位为“1”的情况下, MTUn.TOCR2B.OLS3N、MTUn.TOCR2B.OLS3P、MTUn.TOCR2B.OLS2N、MTUn.TOCR2B.OLS2P、MTUn.TOCR2B.OLS1N、MTUn.TOCR2B.OLS1P 位是“0”时为 Low 电平输出, 是“1”时为 High 电平输出) 至少同时持续输出了 1 个外围时钟 (PCLK) 周期时, 能将 MTU 的互补 PWM 输出引脚 (MTU6、MTU7 引脚) 置为高阻抗状态。能通过 POE3 的寄存器设定对哪个组合进行输出比较并且进行高阻抗控制。
MTIOC7A 和 MTIOC7C	输出	
MTIOC7B 和 MTIOC7D	输出	
GTIOC0A 和 GTIOC0B	输出	当 2 个引脚的有效电平 (在 ALR1.OLSG2B、ALR1.OLSG2A、ALR1.OLSG1B、ALR1.OLSG1A、ALR1.OLSG0B、ALR1.OLSG0A 位是“0”时为 Low 电平输出, 是“1”时为 High 电平输出) 至少同时持续输出了 1 个外围时钟 (PCLK) 周期时, 能将 GPT 输出引脚 (GPT0 ~ GPT2 引脚) 置为高阻抗状态。能通过 POE3 的寄存器设定对哪个组合进行输出比较并且进行高阻抗控制。
GTIOC1A 和 GTIOC1B	输出	
GTIOC2A 和 GTIOC2B	输出	

22.2 寄存器说明

通过复位对 POE3 的寄存器进行初始化。

22.2.1 输入电平控制 / 状态寄存器 1 (ICSR1)

地址 0008 C4C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE0F	—	—	—	PIE1	—	—	—	—	—	—	—	POE0M[1:0]
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
b1-b0	POE0M[1:0]	POE0 模式选择位	b1 b0 0 0: 在 POE0# 输入的下降沿接受请求 0 1: 在每 PCLK/8 时钟对 POE0# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。 1 0: 在每 PCLK/16 时钟对 POE0# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。 1 1: 在每 PCLK/128 时钟对 POE0# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。	R/W (注 1)
b2	—	保留位	读写值都为“0”。	R/W
b8	PIE1	端口中断允许 1 位	0: 禁止中断请求 1: 允许中断请求	R/W
b11-b9	—	保留位	读写值都为“0”。	R/W
b12	POE0F	POE0 标志	0: POE0# 引脚无高阻抗请求 1: POE0# 引脚有高阻抗请求	R/(W) (注 2)
b15-b13	—	保留位	读写值都为“0”。	R/(W)

注 1. 复位后, 只能写 1 次。

注 2. 为了将标志置“0”, 只能在读“1”后写“0”。

ICSR1 寄存器是选择 POE0# 引脚的输入模式、控制中断的允许 / 禁止以及表示各状态的寄存器。

POE0M[1:0] 位 (POE0 模式选择位)

这些位选择 POE0# 引脚的输入模式。

PIE1 位 (端口中断允许 1 位)

当 POE0F 标志为“1”时, 此位指定是否请求中断。

POE0F 标志 (POE0 标志)

这是表示给 POE0# 引脚输入了高阻抗请求的标志。

[为“0”的条件]

- 读“1”后写“0”时

[为“1”的条件]

- POE0# 引脚发生 POE0M[1:0] 位设定的输入时

22.2.2 输出电平控制 / 状态寄存器 1 (OCSR1)

地址 0008 C4C2h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OSF1	—	—	—	—	—	OCE1	OIE1	—	—	—	—	—	—	—	—
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
b7-b0	—	保留位	读写值都为“0”。	R/W
b8	OIE1	输出短路的中断允许 1 位	0: 禁止中断请求 1: 允许中断请求	R/W
b9	OCE1	输出短路的高阻抗允许 1 位	0: 不将引脚置为高阻抗 1: 将引脚置为高阻抗	R/W (注 1)
b14-b10	—	保留位	读写值都为“0”。	R/W
b15	OSF1	输出短路标志 1	0: 不同时变为有效电平 1: 同时变为有效电平	R/(W) (注 2)

注 1. 复位后, 只能写 1 次。

注 2. 为了将标志置“0”, 只能在读“1”后写“0”。

OCSR1 寄存器是允许或者禁止输出电平的比较、控制中断的允许 / 禁止以及表示各状态的寄存器。

OIE1 位 (输出短路的中断允许 1 位)

当 OSF1 标志为“1”时, 此位指定是否请求中断。

OCE1 位 (输出短路的高阻抗允许 1 位)

当 OSF1 标志为“1”时, 此位指定是否将引脚置为高阻抗。

OSF1 标志 (输出短路标志 1)

此标志表示在对 P71 ~ P76 的 MTU 互补 PWM 输出引脚 (MTU3、MTU4 引脚或者 MTU6、MTU7 引脚) 或者 GPT 输出引脚 (GPT0 ~ GPT2 引脚) 进行比较的 3 组 2 相输出中, 至少有 1 组同时为有效电平。

[为“0”的条件]

- 读“1”后写“0”时

[为“1”的条件]

- 3 组 2 相输出中至少有 1 组同时为有效电平时

22.2.3 有效电平寄存器 1 (ALR1)

地址 0008 C4DAh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	MTUCHSEL	OLSEN	—	OLSG2B	OLSG2A	OLSG1B	OLSG1A	OLSG0B	OLSG0A
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	OLSG0A	MTIOC3B/GTIOC0A-A 的有效电平设定位	0: Low 电平有效 1: High 电平有效	R/W (注1)
b1	OLSG0B	MTIOC3D/GTIOC0B-A 的有效电平设定位	0: Low 电平有效 1: High 电平有效	R/W (注1)
b2	OLSG1A	MTIOC4A/GTIOC1A-A 的有效电平设定位	0: Low 电平有效 1: High 电平有效	R/W (注1)
b3	OLSG1B	MTIOC4C/GTIOC1B-A 的有效电平设定位	0: Low 电平有效 1: High 电平有效	R/W (注1)
b4	OLSG2A	MTIOC4B/GTIOC2A-A 的有效电平设定位	0: Low 电平有效 1: High 电平有效	R/W (注1)
b5	OLSG2B	MTIOC4D/GTIOC2B-A 的有效电平设定位	0: Low 电平有效 1: High 电平有效	R/W (注1)
b6	—	保留位	读写值都为“0”。	R/W
b7	OLSEN	有效电平的设定有效位	0: 无效 1: 有效	R/W (注1)
b8	MTUCHSEL	MTU 输出有效电平的通道设定位	0: 选择 MTU3、MTU4 (TOCR1A、TOCR2A) 1: 选择 MTU6、MTU7 (TOCR1B、TOCR2B)	R/W (注1)
b15-b9	—	保留位	读写值都为“0”。	R/W

注 1. 复位后, 只能写 1 次。

ALR1 寄存器是在通过 OCSR1 寄存器检测到 MTU 输出和 GPT 输出的输出短路时, 设定 MTU 输出和 GPT 输出的有效电平的寄存器。

OLSG0A 位 (MTIOC3B/MTIOC6B/GTIOC0A 的有效电平设定位)

此位设定 P71 的 MTIOC3B 输出、MTIOC6B 输出和 GTIOC0A 输出的有效电平。当 OLSG0A 位是“0”时 Low 电平为有效电平, 是“1”时 High 电平为有效电平, 并且进行输出短路的检测。

OLSG0B 位 (MTIOC3D/MTIOC6D/GTIOC0B 的有效电平设定位)

此位设定 P74 的 MTIOC3D 输出、MTIOC6D 输出和 GTIOC0B 输出的有效电平。当 OLSG0B 位是“0”时 Low 电平为有效电平, 是“1”时 High 电平为有效电平, 并且进行输出短路的检测。

OLSG1A 位 (MTIOC4A/MTIOC7A/GTIOC1A 的有效电平设定位)

此位设定 P72 的 MTIOC4A 输出、MTIOC7A 输出和 GTIOC1A 输出的有效电平。当 OLSG1A 位是“0”时 Low 电平为有效电平, 是“1”时 High 电平为有效电平, 并且进行输出短路的检测。

OLSG1B 位 (MTIOC4C/MTIOC7C/GTIOC1B 的有效电平设定位)

此位设定 P75 的 MTIOC4C 输出、MTIOC7C 输出和 GTIOC1B 输出的有效电平。当 OLSG1B 位是“0”时 Low 电平为有效电平，是“1”时 High 电平为有效电平，并且进行输出短路的检测。

OLSG2A 位 (MTIOC4B/MTIOC7B/GTIOC2A 的有效电平设定位)

此位设定 P73 的 MTIOC4B 输出、MTIOC7B 输出和 GTIOC2A 输出的有效电平。当 OLSG2A 位是“0”时 Low 电平为有效电平，是“1”时 High 电平为有效电平，并且进行输出短路的检测。

OLSG2B 位 (MTIOC4D/MTIOC7D/GTIOC2B 的有效电平设定位)

此位设定 P76 的 MTIOC4D 输出、MTIOC7D 输出和 GTIOC2B 输出的有效电平。当 OLSG2B 位是“0”时 Low 电平为有效电平，是“1”时 High 电平为有效电平，并且进行输出短路的检测。

OLSEN 位 (有效电平的设定有效位)

此位设定通过 OLSGnm 位 (n=0~2、m=A、B) 设定的有效电平有效或者无效。当 OLSEN 位为“0”时，OLSGnm 位的设定无效，MTU 输出的有效电平取决于 MTU.TOCR1 寄存器和 MTU.TOCR2 寄存器的设定，而 MTU.TOCR1 寄存器和 MTU.TOCR2 寄存器则通过 MTUCHSEL 位来设定；当 OLSEN 位为“1”时，MTU 输出 OLSGnm 位设定的有效电平。

只有在 OLSEN 位为“1”时才能设定 GPT 输出的有效电平。在进行 GPT 输出的输出短路检测时，必须将 OLSEN 位置“1”，并且通过 OLSGnm 位设定 GPT 输出的有效电平。

MTUCHSEL 位 (MTU 输出有效电平的通道设定位)

此位设定是选择 MTU3、MTU4 (MTUn.TOCR1A、MTUn.TOCR2A) 还是选择 MTU6、MTU7 (MTUm.TOCR1B、MTUm.TOCR2B) 作为 MTU 的互补 PWM 输出引脚。(n=3、4、m=6、7)

如果使用 MTU 的互补 PWM 输出，必须在复位后，并且将引脚设定为输出前，设定 MTUCHSEL 位。

22.2.4 输入电平控制 / 状态寄存器 3 (ICSR3)

地址 0008 C4C8h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	POE8F	—	—	POE8E	PIE3	—	—	—	—	—	—	POE8M[1:0]	—
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
b1-b0	POE8M[1:0]	POE8 模式选择位	b1 b0 0 0: 在 POE8# 输入的下降沿接受请求 0 1: 在每 PCLK/8 时钟对 POE8# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。 1 0: 在每 PCLK/16 时钟对 POE8# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。 1 1: 在每 PCLK/128 时钟对 POE8# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。	R/W (注 1)
b7-b2	—	保留位	读写值都为“0”。	R/W
b8	PIE3	端口中断允许 3 位	0: 禁止中断请求 1: 允许中断请求	R/W
b9	POE8E	POE8 高阻抗允许位	0: 不将引脚置为高阻抗 1: 将 M 引脚置为高阻抗	R/W (注 1)
b11-b10	—	保留位	读写值都为“0”。	R/W
b12	POE8F	POE8 标志	0: POE8# 引脚无高阻抗请求 1: POE8# 引脚有高阻抗请求	R/(W) (注 2)
b15-b13	—	保留位	读写值都为“0”。	R/W

注 1. 复位后, 只能写 1 次。

注 2. 为了将标志置“0”, 只能在读“1”后写“0”。

ICSR3 寄存器是选择 POE8# 引脚的输入模式, 控制中断的允许 / 禁止以及表示各状态的寄存器。

POE8M[1:0] 位 (POE8 模式选择位)

这些位选择 POE8# 引脚的输入模式。

PIE3 位 (端口中断允许 3 位)

当 POE8F 标志为“1”时, 此位指定是否请求中断。

POE8E 位 (POE8 高阻抗允许位)

当 POE8F 标志为“1”时, 此位指定是否将引脚置为高阻抗。

POE8F 标志 (POE8 标志)

这是表示给 POE8# 引脚输入了高阻抗请求的标志。

[为“0”的条件]

- 读“1”后写“0”时

[为“1”的条件]

- POE8# 引脚发生 POE8M[1:0] 位设定的输入时

22.2.5 输入电平控制 / 状态寄存器 4 (ICSR4)

地址 0008 C4D6h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	POE10F	—	—	POE10E	PIE4	—	—	—	—	—	—	—	POE10M[1:0]
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
b1-b0	POE10M[1:0]	POE10 模式选择位	b1 b0 0 0: 在 POE10# 输入的下降沿接受请求 0 1: 在每 PCLK/8 时钟对 POE10# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。 1 0: 在每 PCLK/16 时钟对 POE10# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。 1 1: 在每 PCLK/128 时钟对 POE10# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。	R/W (注 1)
b7-b2	—	保留位	读写值都为“0”。	R/W
b8	PIE4	端口中断允许 4 位	0: 禁止中断请求 1: 允许中断请求	R/W
b9	POE10E	POE10 高阻抗允许位	0: 不将引脚置为高阻抗 1: 将引脚置为高阻抗	R/W (注 1)
b11-b10	—	保留位	读写值都为“0”。	R/W
b12	POE10F	POE10 标志	0: POE10# 引脚无高阻抗请求 1: POE10# 引脚有高阻抗请求	R/(W) (注 2)
b15-b13	—	保留位	读写值都为“0”。	R/W

注 1. 复位后, 只能写 1 次。

注 2. 为了将标志置“0”, 只能在读“1”后写“0”。

ICSR4 寄存器是选择 POE10# 引脚的输入模式、控制中断的允许 / 禁止以及表示各状态的寄存器。

POE10M[1:0] 位 (POE10 模式选择位)

这些位选择 POE10# 引脚的输入模式。

PIE4 位 (端口中断允许 4 位)

当 POE10F 标志为“1”时, 此位指定是否请求中断。

POE10E 位 (POE10 高阻抗允许位)

当 POE10F 标志为“1”时, 此位指定是否将引脚置为高阻抗。

POE10F 标志 (POE10 标志)

这是表示给 POE10# 引脚输入了高阻抗请求的标志。

[为“0”的条件]

- 读“1”后写“0”时

[为“1”的条件]

- POE10# 引脚发生 POE10M[1:0] 位设定的输入时

22.2.6 输入电平控制 / 状态寄存器 5 (ICSR5)

地址 0008 C4D8h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	POE11F	—	—	POE11E	PIE5	—	—	—	—	—	—	—	POE11M[1:0]
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
b1-b0	POE11M[1:0]	POE11 模式选择位	b1 b0 0 0: 在 POE11# 输入的下降沿接受请求 0 1: 在每个 PCLK/8 时钟对 POE11# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。 1 0: 在每个 PCLK/16 时钟对 POE11# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。 1 1: 在每个 PCLK/128 时钟对 POE11# 输入的 Low 电平进行 16 次采样并且全部为 Low 电平时, 接受请求。	R/W (注 1)
b7-b2	—	保留位	读写值都为“0”。	R/W
b8	PIE5	端口中断允许 5 位	0: 禁止中断请求 1: 允许中断请求	R/W
b9	POE11E	POE11 高阻抗允许位	0: 不将引脚置为高阻抗 1: 将引脚置为高阻抗	R/W (注 1)
b11-b10	—	保留位	读写值都为“0”。	R/W
b12	POE11F	POE11 标志	0: POE11# 引脚无高阻抗请求 1: POE11# 引脚有高阻抗请求	R/(W) (注 2)
b15-b13	—	保留位	读写值都为“0”。	R/W

注 1. 复位后, 只能写 1 次。

注 2. 为了将标志置“0”, 只能在读“1”后写“0”。

ICSR5 寄存器是选择 POE11# 引脚的输入模式、控制中断的允许 / 禁止以及表示各状态的寄存器。

POE11M[1:0] 位 (POE11 模式选择位)

这些位选择 POE11# 引脚的输入模式。

PIE5 位 (端口中断允许 5 位)

当 POE11F 标志为“1”时, 此位指定是否请求中断。

POE11E 位 (POE11 高阻抗允许位)

当 POE11F 标志为“1”时, 此位指定是否将引脚置为高阻抗。

POE11F 标志 (POE11 标志)

这是表示给 POE11# 引脚输入了高阻抗请求的标志。

[为“0”的条件]

- 读“1”后写“0”时

[为“1”的条件]

- POE11# 引脚发生 POE11M[1:0] 位设定的输入时

22.2.7 软件端口输出允许寄存器 (SPOER)

地址 0008 C4CAh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	GPT23HIZ	GPT01HIZ	MTUCH0HIZ	—	MTUCH34HIZ
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	MTUCH34HIZ	MTU3、MTU4 或者 MTU6、MTU7 的输出高阻抗允许位	0: 不置为高阻抗状态 1: 置为高阻抗状态	R/W
b1	—	保留位	读写值都为“0”。	R/W
b2	MTUCH0HIZ	MTU0 的输出高阻抗允许位	0: 不置为高阻抗状态 1: 置为高阻抗状态	R/W
b3	GPT01HIZ	GPT0、GPT1 的输出高阻抗允许位	0: 不置为高阻抗状态 1: 置为高阻抗状态	R/W
b4	GPT23HIZ	GPT2、GPT3 的输出高阻抗允许位	0: 不置为高阻抗状态 1: 置为高阻抗状态	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W

SPOER 寄存器是对引脚进行高阻抗控制的寄存器。

MTUCH34HIZ 位 (MTU3、MTU4 或者 MTU6、MTU7 的输出高阻抗允许位)

此位对 P71 ~ P76 的 MTU 的互补 PWM 输出引脚 (MTIOC3B、MTIOC3D、MTIOC4A、MTIOC4B、MTIOC4C、MTIOC4D 或者 MTIOC6B、MTIOC6D、MTIOC7A、MTIOC7B、MTIOC7C、MTIOC7D) 或者 GPT 输出引脚 (GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B、GTIOC2A、GTIOC2B) 进行高阻抗控制。

[为“0”的条件]

- 复位
- 读“1”后写“0”时

[为“1”的条件]

- 写“1”时

MTUCH0HIZ 位 (MTU0 的输出高阻抗允许位)

此位对 MTU0 的引脚进行高阻抗控制。

[为“0”的条件]

- 复位
- 读“1”后写“0”时

[为“1”的条件]

- 写“1”时

GPT01HIZ 位 (GPT0、GPT1 的输出高阻抗允许位)

此位对 PD4 ~ PD7 的 GPT0、GPT1 引脚 (GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B) 进行高阻抗控制。

[为“0”的条件]

- 复位
- 读“1”后写“0”时

[为“1”的条件]

- 写“1”时

GPT23HIZ 位 (GPT2、GPT3 的输出高阻抗允许位)

此位对 P00、P01、PD3、PB6、PB7 的 GPT2、GPT3 引脚 (GTIOC2A、GTIOC2B、GTIOC3A、GTIOC3B) 进行高阻抗控制。

[为“0”的条件]

- 复位
- 读“1”后写“0”时

[为“1”的条件]

- 写“1”时

22.2.8 端口输出允许控制寄存器 1 (POECR1)

地址 0008 C4CBh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	MTU0D ZE	MTU0C ZE	MTU0B ZE	MTU0A ZE
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	MTU0AZE	MTU CH0A 的高阻抗允许位	0: 不置为高阻抗 1: 置为高阻抗	R/W (注1)
b1	MTU0BZE	MTU CH0B 的高阻抗允许位	0: 不置为高阻抗 1: 置为高阻抗	R/W (注1)
b2	MTU0CZE	MTU CH0C 的高阻抗允许位	0: 不置为高阻抗 1: 置为高阻抗	R/W (注1)
b3	MTU0DZE	MTU CH0D 的高阻抗允许位	0: 不置为高阻抗 1: 置为高阻抗	R/W (注1)
b7-b4	—	保留位	读写值都为“0”。	R/W

注 1. 复位后, 只能写 1 次。

POECR1 寄存器是对 MTU0 引脚进行高阻抗控制的寄存器。

MTU0AZE 位 (MTU CH0A 的高阻抗允许位)

当通过 ICSR3.POE8F 标志、SPOER.MTUCH0HIZ 位、ICSR6.OSTSTE 位和 POECR5 寄存器追加选择的 ICSRn.POE_mF (n=1、4、5, m=0、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志中的任意一个为“1”时, 此位设定是否将 P31、PB3 的 MTU0 引脚的 MTIOC0A 输出置为高阻抗状态。

MTU0BZE 位 (MTU CH0B 的高阻抗允许位)

当通过 ICSR3.POE8F 标志、SPOER.MTUCH0HIZ 位、ICSR6.OSTSTE 位和 POECR5 寄存器追加选择的 ICSRn.POE_mF (n=1、4、5, m=0、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志中的任意一个为“1”时, 此位设定是否将 P30、PB2 的 MTU0 引脚的 MTIOC0B 输出置为高阻抗状态。

MTU0CZE 位 (MTU CH0C 的高阻抗允许位)

当通过 ICSR3.POE8F 标志、SPOER.MTUCH0HIZ 位、ICSR6.OSTSTE 位和 POECR5 寄存器追加选择的 ICSRn.POE_mF (n=1、4、5, m=0、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志中的任意一个为“1”时, 此位设定是否将 PB1 的 MTU0 引脚的 MTIOC0C 输出置为高阻抗状态。

MTU0DZE 位 (MTU CH0D 的高阻抗允许位)

当通过 ICSR3.POE8F 标志、SPOER.MTUCH0HIZ 位、ICSR6.OSTSTE 位和 POECR5 寄存器追加选择的 ICSRn.POE_mF (n=1、4、5, m=0、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志中的任意一个为“1”时, 此位设定是否将 PB0 的 MTU0 引脚的 MTIOC0D 输出置为高阻抗状态。

22.2.9 端口输出允许控制寄存器 2 (POE2CR2)

地址 0008 C4CCh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	MTU3BDZE	MTU4ACZE	MTU4BDZE	—	—	—	—	—	—	—	—
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

复位后的值

位	符号	位名	功能	R/W
b2-b0	—	保留位	读写值都为“1”。	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W
b8	MTU4BDZE	MTU CH4BD 的高阻抗允许位	0: 不置为高阻抗 1: 置为高阻抗。	R/W (注1)
b9	MTU4ACZE	MTU CH4AC 的高阻抗允许位	0: 不置为高阻抗 1: 置为高阻抗。	R/W (注1)
b10	MTU3BDZE	MTU CH3BD 的高阻抗允许位	0: 不置为高阻抗 1: 置为高阻抗。	R/W (注1)
b15-b11	—	保留位	读写值都为“0”。	R/W

注1. 复位后, 只能写1次。

POE2CR2 寄存器是对 P71 ~ P76 的 MTU 的互补 PWM 输出引脚 (MTU3、MTU4、MTU6、MTU7 引脚) 和 GPT 输出引脚 (GPT0 ~ GPT2 引脚) 进行高阻抗控制的寄存器。

MTU4BDZE 位 (MTU CH4BD 的高阻抗允许位)

当通过 OCSR1.OSF1 标志、ICSR1.POE0F 标志、SPOER.MTUCH34HIZ 位、ICSR6.OSTSTE 位和 POE2CR4 寄存器追加选择的 ICSRn.POE_mF (n=3 ~ 5, m=8、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志中的任意一个为“1”时, 此位设定是否将 P73、P76 的 MTU4/MTU7/GPT2 引脚的 MTIOC4B 输出 / MTIOC7B 输出 / GTIOC2A 输出和 MTIOC4D 输出 / GTIOC7D 输出 / GTIOC2B 输出置为高阻抗状态。

MTU4ACZE 位 (MTU CH4AC 的高阻抗允许位)

当通过 OCSR1.OSF1 标志、ICSR1.POE0F 标志、SPOER.MTUCH34HIZ 位、ICSR6.OSTSTE 位和 POE2CR4 寄存器追加选择的 ICSRn.POE_mF (n=3 ~ 5, m=8、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志中的任意一个为“1”时, 此位设定是否将 P72、P75 的 MTU4/MTU7/GPT1 引脚的 MTIOC4A 输出 / MTIOC7A 输出 / GTIOC1A 输出和 MTIOC4C 输出 / MTIOC7C 输出 / GTIOC1B 输出置为高阻抗状态。

MTU3BDZE 位 (MTU CH3BD 的高阻抗允许位)

当通过 OCSR1.OSF1 标志、ICSR1.POE0F 标志、SPOER.MTUCH34HIZ 位、ICSR6.OSTSTE 位和 POE2CR4 寄存器追加选择的 ICSRn.POE_mF (n=3 ~ 5, m=8、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志中的任意一个为“1”时, 此位设定是否将 P71、P74 的 MTU3/MTU6/GPT0 引脚的 MTIOC3B 输出 / MTIOC6B 输出 / GTIOC0A 输出和 MTIOC3D 输出 / MTIOC6D 输出 / GTIOC0B 输出置为高阻抗状态。

22.2.10 端口输出允许控制寄存器 3 (POE3)

地址 0008 C4CEh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
—	—	—	—	—	—	GPT3ABZE	GPT2ABZE	—	—	—	—	—	—	GPT1ABZE	GPT0ABZE	
复位后的值	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

位	符号	位名	功能	R/W
b0	GPT0ABZE	GPT CH0AB 的高阻抗允许位	0: 不置为高阻抗 1: 置为高阻抗。	R/W (注1)
b1	GPT1ABZE	GPT CH1AB 的高阻抗允许位	0: 不置为高阻抗 1: 置为高阻抗。	R/W (注1)
b7-b2	—	保留位	读写值都为“0”。	R/W
b8	GPT2ABZE	GPT CH2AB 的高阻抗允许位	0: 不置为高阻抗 1: 置为高阻抗。	R/W (注1)
b9	GPT3ABZE	GPT CH3AB 的高阻抗允许位	0: 不置为高阻抗 1: 置为高阻抗。	R/W (注1)
b15-b10	—	保留位	读写值都为“0”。	R/W

注 1. 复位后, 只能写 1 次。

POE3 寄存器是对 GPT 引脚进行高阻抗控制的寄存器。

GPT0ABZE 位 (GPT CH0AB 的高阻抗允许位)

当通过 ICSR4.POE10F 标志、SPOER.GPT0IHIZ 位、ICSR6.OSTSTE 位和 POE3 寄存器追加选择的 ICSRn.POE_mF (n=1、3、5, m=0、8、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志中的任意一个为“1”时, 此位设定是否将 PD6、PD7 的 GPT0 引脚的 GTIOC0A 输出和 GTIOC0B 输出置为高阻抗状态。

GPT1ABZE 位 (GPT CH1AB 的高阻抗允许位)

当通过 ICSR4.POE10F 标志、SPOER.GPT0IHIZ 位、ICSR6.OSTSTE 位和 POE3 寄存器追加选择的 ICSRn.POE_mF (n=1、3、5, m=0、8、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志中的任意一个为“1”时, 此位设定是否将 PD4、PD5 的 GPT1 引脚的 GTIOC1A 输出和 GTIOC1B 输出置为高阻抗状态。

GPT2ABZE 位 (GPT CH2AB 的高阻抗允许位)

当通过 ICSR5.POE11F 标志、SPOER.GPT23HIZ 位、ICSR6.OSTSTE 位和 POE3 寄存器追加选择的 ICSRn.POE_mF (n=1、3、4, m=0、8、10) 标志和 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志中的任意一个为“1”时, 此位设定是否将 PD3、PB6、PB7 的 GPT2 引脚的 GTIOC2A 输出和 GTIOC2B 输出置为高阻抗状态。

GPT3ABZE 位 (GPT CH3AB 的高阻抗允许位)

当通过 ICSR5.POE11F 标志、SPOER.GPT23HIZ 位、ICSR6.OSTSTE 位和 POE3 寄存器追加选择的 ICSRn.POE_mF (n=1、3、4, m=0、8、10) 标志和 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志中的任意一个为“1”时, 此位设定是否将 P00、P01 的 GPT3 引脚的 GTIOC3A 输出和 GTIOC3B 输出置为高阻抗状态。

22.2.11 端口输出允许控制寄存器 4 (POECR4)

地址 0008 C4D0h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	IC5ADD MT34ZE	IC4ADD MT34ZE	IC3ADD MT34ZE	—	—	CMADD MT34ZE
复位后的值	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

位	符号	位名	功能	R/W
b0	CMADDMT34ZE	MTU CH34 的高阻抗 CFLAG 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b1	—	保留位	读写值都为“1”。	R/W
b2	—	保留位	读写值都为“0”。	R/W
b3	IC3ADDMT34ZE	MTU CH34 的高阻抗 POE8F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b4	IC4ADDMT34ZE	MTU CH34 的高阻抗 POE10F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b5	IC5ADDMT34ZE	MTU CH34 的高阻抗 POE11F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b9-b6	—	保留位	读写值都为“0”。	R/W
b10	—	保留位	读写值都为“1”。	R/W
b15-b11	—	保留位	读写值都为“0”。	R/W

注 1. 复位后, 只能写 1 次。

POECR4 寄存器是扩展 MTU 的互补 PWM 输出引脚 (MTU3、MTU4、MTU6、MTU7 引脚) 和 GPT 输出引脚 (GPT0 ~ GPT2 引脚) 的高阻抗控制条件的寄存器。

CMADDMT34ZE 位 (MTU CH34 的高阻抗 CFLAG 追加位)

将 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志追加到 P71 ~ P76 的 MTU3、MTU4、MTU6、MTU7、GPT0 ~ GPT2 引脚 (MTIOC3B/MTIOC3D/MTIOC6B/MTIOC6D/GTIOC0A/GTIOC0B/MTIOC4A/MTIOC4C/MTIOC7A/MTIOC7C/GTIOC1A/GTIOC1B/MTIOC4B/MTIOC4D/MTIOC7B/MTIOC7D/GTIOC2A/GTIOC2B) 的高阻抗控制条件。但是, 如果通过 S12AD.ADCMPFR.CjFLAG 标志将引脚置为高阻抗, 就不发生 OEIn (n=1、3、4) 中断。

IC3ADDMT34ZE 位 (MTU CH34 的高阻抗 POE8F 追加位)

将 ICSR3.POE8F 标志追加到 P71 ~ P76 的 MTU3、MTU4、MTU6、MTU7、GPT0 ~ GPT2 引脚 (MTIOC3B/MTIOC3D/MTIOC6B/MTIOC6D/GTIOC0A/GTIOC0B/MTIOC4A/MTIOC4C/MTIOC7A/MTIOC7C/GTIOC1A/GTIOC1B/MTIOC4B/MTIOC4D/MTIOC7B/MTIOC7D/GTIOC2A/GTIOC2B) 的高阻抗控制条件。

IC4ADDMT34ZE 位 (MTU CH34 的高阻抗 POE10F 追加位)

将 ICSR4.POE10F 标志追加到 P71 ~ P76 的 MTU3、MTU4、MTU6、MTU7、GPT0 ~ GPT2 引脚 (MTIOC3B/MTIOC3D/MTIOC6B/MTIOC6D/GTIOC0A/GTIOC0B/MTIOC4A/MTIOC4C/MTIOC7A/MTIOC7C/GTIOC1A/GTIOC1B/MTIOC4B/MTIOC4D/MTIOC7B/MTIOC7D/GTIOC2A/GTIOC2B) 的高阻抗控制条件。

IC5ADDMT34ZE 位 (MTU CH34 的高阻抗 POE11F 追加位)

将 ICSR5.POE11F 标志追加到 P71 ~ P76 的 MTU3、MTU4、MTU6、MTU7、GPT0 ~ GPT2 引脚 (MTIOC3B/MTIOC3D/MTIOC6B/MTIOC6D/GTIOC0A/GTIOC0B/MTIOC4A/MTIOC4C/MTIOC7A/MTIOC7C/GTIOC1A/GTIOC1B/MTIOC4B/MTIOC4D/MTIOC7B/MTIOC7D/GTIOC2A/GTIOC2B) 的高阻抗控制条件。

22.2.12 端口输出允许控制寄存器 5 (POECSR5)

地址 0008 C4D2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	IC5ADD MT0ZE	IC4ADD MT0ZE	—	—	IC1ADD MT0ZE	CMADD MT0ZE
复位后的值	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

位	符号	位名	功能	R/W
b0	CMADDMT0ZE	MTU0 CH0 的高阻抗 CFLAG 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注1)
b1	IC1ADDMT0ZE	MTU CH0 的高阻抗 POE0F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注1)
b2	—	保留位	读写值都为“0”。	R/W
b3	—	保留位	读写值都为“1”。	R/W
b4	IC4ADDMT0ZE	MTU CH0 的高阻抗 POE10F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注1)
b5	IC5ADDMT0ZE	MTU CH0 的高阻抗 POE11F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注1)
b15-b6	—	保留位	读写值都为“0”。	R/W

注 1. 复位后, 只能写 1 次。

POECSR5 寄存器是扩展 MTU0 引脚的高阻抗控制条件的寄存器。

CMADDMT0ZE 位 (MTU CH0 的高阻抗 CFLAG 追加位)

将 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志追加到 MTU0 引脚 (MTIOC0A、MTIOC0B、MTIOC0C、MTIOC0D) 的高阻抗控制条件。但是, 如果通过 S12AD.ADCMPFR.CjFLAG 标志将引脚置为高阻抗, 就不发生 OEIn (n=1、3、4) 中断。

IC1ADDMT0ZE 位 (MTU CH0 的高阻抗 POE0F 追加位)

将 ICSR1.POE0F 标志追加到 MTU0 引脚 (MTIOC0A、MTIOC0B、MTIOC0C、MTIOC0D) 的高阻抗控制条件。

IC4ADDMT0ZE 位 (MTU CH0 的高阻抗 POE10F 追加位)

将 ICSR4.POE10F 标志追加到 MTU0 引脚 (MTIOC0A、MTIOC0B、MTIOC0C、MTIOC0D) 的高阻抗控制条件。

IC5ADDMT0ZE 位 (MTU CH0 的高阻抗 POE11F 追加位)

将 ICSR5.POE11F 标志追加到 MTU0 引脚 (MTIOC0A、MTIOC0B、MTIOC0C、MTIOC0D) 的高阻抗控制条件。

22.2.13 端口输出允许控制寄存器 6 (POECR6)

地址 0008 C4D4h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	IC4ADDGP T23ZE	IC3ADDGP T23ZE	—	IC1ADDGP T23ZE	CMADDGP T23ZE	—	—	IC5ADDGP T01ZE	—	IC3ADDGP T01ZE	—	IC1ADDGP T01ZE	CMADDGP T01ZE
复位后的值	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

位	符号	位名	功能	R/W
b0	CMADDGPT01ZE	GPT CH01 的高阻抗 CFLAG 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b1	IC1ADDGPT01ZE	GPT CH01 的高阻抗 POE0F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b2	—	保留位	读写值都为“0”。	R/W
b3	IC3ADDGPT01ZE	GPT CH01 的高阻抗 POE8F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b4	—	保留位	读写值都为“1”。	R/W
b5	IC5ADDGPT01ZE	GPT CH01 的高阻抗 POE11F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b7-b6	—	保留位	读写值都为“0”。	R/W
b8	CMADDGPT23ZE	GPT CH23 的高阻抗 CFLAG 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b9	IC1ADDGPT23ZE	GPT CH23 的高阻抗 POE0F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b10	—	保留位	读写值都为“0”。	R/W
b11	IC3ADDGPT23ZE	GPT CH23 的高阻抗 POE8F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b12	IC4ADDGPT23ZE	GPT CH23 的高阻抗 POE10F 追加位	0: 不追加到高阻抗控制条件 1: 追加到高阻抗控制条件。	R/W (注 1)
b13	—	保留位	读写值都为“1”。	R/W
b15-b14	—	保留位	读写值都为“0”。	R/W

注 1. 复位后, 只能写 1 次。

POECR6 寄存器是扩展 GPT0 ~ GPT3 引脚的高阻抗控制条件的寄存器。

CMADDGPT01ZE 位 (GPT CH01 的高阻抗 CFLAG 追加位)

将 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志追加到 PD4 ~ PD7 的 GPT0、GPT1 引脚 (GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B) 的高阻抗控制条件。但是, 如果通过 S12AD.ADCMPFR.CjFLAG 标志将引脚置为高阻抗, 就不发生 OEIn (n=1、3、4) 中断。

IC1ADDGPT01ZE 位 (GPT CH01 的高阻抗 POE0F 追加位)

将 ICSR1.POE0F 标志追加到 PD4 ~ PD7 的 GPT0、GPT1 引脚 (GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B) 的高阻抗控制条件。

IC3ADDGPT01ZE 位 (GPT CH01 的高阻抗 POE8F 追加位)

将 ICSR3.POE8F 标志追加到 PD4 ~ PD7 的 GPT0、GPT1 引脚 (GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B) 的高阻抗控制条件。

IC5ADDGPT01ZE 位 (GPT CH01 的高阻抗 POE11F 追加位)

将 ICSR5.POE11F 标志追加到 PD4 ~ PD7 的 GPT0、GPT1 引脚 (GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B) 的高阻抗控制条件。

CMADDGPT23ZE 位 (GPT CH23 的高阻抗 CFLAG 追加位)

将 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志追加到 PD3、PB6、PB7、P00、P01 的 GPT2、GPT3 引脚 (GTIOC2A、GTIOC2B、GTIOC3A、GTIOC3B) 的高阻抗控制条件。但是, 如果通过 S12AD.ADCMPFR.CjFLAG 标志将引脚置为高阻抗, 就不发生 OEIn (n=1、3、4) 中断。

IC1ADDGPT23ZE 位 (GPT CH23 的高阻抗 POE0F 追加位)

将 ICSR1.POE0F 标志追加到 PD3、PB6、PB7、P00、P01 的 GPT2、GPT3 引脚 (GTIOC2A、GTIOC2B、GTIOC3A、GTIOC3B) 的高阻抗控制条件。

IC3ADDGPT23ZE 位 (GPT CH23 的高阻抗 POE8F 追加位)

将 ICSR3.POE8F 标志追加到 PD3、PB6、PB7、P00、P01 的 GPT2、GPT3 引脚 (GTIOC2A、GTIOC2B、GTIOC3A、GTIOC3B) 的高阻抗控制条件。

IC4ADDGPT23ZE 位 (GPT CH23 的高阻抗 POE10F 追加位)

将 ICSR4.POE10F 标志追加到 PD3、PB6、PB7、P00、P01 的 GPT2、GPT3 引脚 (GTIOC2A、GTIOC2B、GTIOC3A、GTIOC3B) 的高阻抗控制条件。

22.2.14 输入电平控制 / 状态寄存器 6 (ICSR6)

地址 0008 C4DCh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	OSTSTF	—	—	OSTSTE	—	—	—	—	—	—	—	—	POE11M[1:0]
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
b8-b0	—	保留位	读写值都为“0”。	R/W
b9	OSTSTE	OSTST 高阻抗允许位	0: 不将 MTU 的互补 PWM 输出引脚、MTU0 引脚以及 GPT 引脚置为高阻抗 1: 将 MTU 的互补 PWM 输出引脚、MTU0 引脚以及 GPT 引脚置为高阻抗	R/W (注 1)
b11-b10	—	保留位	读写值都为“0”。	R/W
b12	OSTSTF	OSTST 高阻抗标志	0: 无振荡停止高阻抗请求 1: 有振荡停止高阻抗请求	R/W (注 2)
b15-b13	—	保留位	读写值都为“0”。	R/W

注 1. 复位后, 只能写 1 次。

注 2. 为了将标志置“0”, 只能在读“1”后写“0”。

ICSR6 寄存器是表示振荡停止时的控制和状态寄存器。

OSTSTE 位 (OSTST 高阻抗允许位)

此位设定在检测到振荡停止时是否将 MTU 的互补 PWM 输出引脚、MTU0 引脚以及 GPT 引脚置为高阻抗。

OSTSTF 标志 (OSTST 高阻抗标志)

这是表示振荡停止高阻抗请求的状态标志。

振荡停止时此标志为“1”。如果要将 OSTSTF 标志置“0”, 必须在振荡停止检测信号无效的状态下写“0”。在振荡停止检测信号有效的状态下即使给 OSTSTF 标志写“0”, OSTSTF 标志也不变为“0”。必须在清除 OSTSTF 标志后确认已变为“0”。

[为“0”的条件]

- 读“1”后写“0”时

[为“1”的条件]

- 检测到振荡停止时

22.3 运行说明

高阻抗控制的对象引脚以及置为高阻抗的条件如表 22.4 所示。

表 22.4 高阻抗控制的对象引脚及其条件 (1/3)

引脚	条件	详细内容
MTU3B/MTU3D 引脚或者 MTU6B/MTU6D 引脚 (MTIOC3B、MTIOC3D、 MTIOC6B、MTIOC6D)	<ul style="list-style-type: none"> • POE0# 引脚的输入电平检测 • MTIOC3B 输出和 MTIOC3D 输出或者 MTIOC6B 输出和 MTIOC6D 输出的输出电平比较 • SPOER 寄存器设定 • 振荡停止检测 • 通过 POECR4 寄存器追加的条件 <ul style="list-style-type: none"> - 12 位 A/D 转换器的比较器检测 - POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测 	MTU3BDZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICSR6.OSTSTE) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=000 ~ 002)
MTU4A/MTU4C 引脚或者 MTU7A/MTU7C 引脚 (MTIOC4A、MTIOC4C、 MTIOC7A、MTIOC7C)	<ul style="list-style-type: none"> • POE0# 引脚的输入电平检测 • MTIOC4A 输出和 MTIOC4C 输出或者 MTIOC7A 输出和 MTIOC7C 输出的输出电平比较 • SPOER 寄存器设定 • 振荡停止检测 • 通过 POECR4 寄存器追加的条件 <ul style="list-style-type: none"> - 12 位 A/D 转换器的比较器检测 - POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测 	MTU4ACZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICSR6.OSTSTE) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=000 ~ 002)
MTU4B/MTU4D 引脚或者 MTU7B/MTU7D 引脚 (MTIOC4B、MTIOC4D、 MTIOC7B、MTIOC7D)	<ul style="list-style-type: none"> • POE0# 引脚的输入电平检测 • MTIOC4B 输出和 MTIOC4D 输出或者 MTIOC7B 输出和 MTIOC7D 输出的输出电平比较 • SPOER 寄存器设定 • 振荡停止检测 • 通过 POECR4 寄存器追加的条件 <ul style="list-style-type: none"> - 12 位 A/D 转换器的比较器检测 - POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测 	MTU4BDZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICSR6.OSTSTE) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=000 ~ 002)
GPT0 引脚 (GTIOC0A (PD7)、 GTIOC0B (PD6))	<ul style="list-style-type: none"> • POE0# 引脚的输入电平检测 • GTIOC0A 输出和 GTIOC0B 输出的输出电平比较 • SPOER 寄存器设定 • 振荡停止检测 • 通过 POECR4 寄存器追加的条件 <ul style="list-style-type: none"> - 12 位 A/D 转换器的比较器检测 - POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测 	MTU3BDZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICSR6.OSTSTE) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=000 ~ 002)

表 22.4 高阻抗控制的对象引脚及其条件 (2/3)

引脚	条件	详细内容
GPT1 引脚 (GTIOC1A (PD5)、 GTIOC1B (PD4))	<ul style="list-style-type: none"> POE0# 引脚的输入电平检测 GTIOC1A 输出和 GTIOC1B 输出的输出电平比较 SPOER 寄存器设定 振荡停止检测 通过 POECR4 寄存器追加的条件 <ul style="list-style-type: none"> 12 位 A/D 转换器的比较器检测 POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测 	MTU4ACZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICSR6.OSTSTE) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=000 ~ 002)
GPT2 引脚 (GTIOC2A (PD3)、 GTIOC2B (PB6、PB7))	<ul style="list-style-type: none"> POE0# 引脚的输入电平检测 GTIOC2A 输出和 GTIOC2B 输出的输出电平比较 SPOER 寄存器设定 振荡停止检测 通过 POECR4 寄存器追加的条件 <ul style="list-style-type: none"> 12 位 A/D 转换器的比较器检测 POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测 	MTU4BDZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICSR6.OSTSTE) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=000 ~ 002)
MTU0A 引脚 (MTIOC0A (P31、PB3))	<ul style="list-style-type: none"> POE8# 引脚的输入电平检测 SPOER 寄存器设定 振荡停止检测 通过 POECR5 寄存器追加的条件 <ul style="list-style-type: none"> 12 位 A/D 转换器的比较器检测 POE0# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测 	MTU0AZE• ((POE8F•POE8E) +(MTUCH0HIZ) +(ICSR6.OSTSTE) +(CMADDMT0ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDMT0ZE•POE0F) +(IC4ADDMT0ZE•POE10E•POE10F) +(IC5ADDMT0ZE•POE11E•POE11F)) (n=000 ~ 002)
MTU0B 引脚 (MTIOC0B (P30、PB2))	<ul style="list-style-type: none"> POE8# 引脚的输入电平检测 SPOER 寄存器设定 振荡停止检测 通过 POECR5 寄存器追加的条件 <ul style="list-style-type: none"> 12 位 A/D 转换器的比较器检测 POE0# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测 	MTU0BZE• ((POE8F•POE8E) +(MTUCH0HIZ) +(ICSR6.OSTSTE) +(CMADDMT0ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDMT0ZE•POE0F) +(IC4ADDMT0ZE•POE10E•POE10F) +(IC5ADDMT0ZE•POE11E•POE11F)) (n=000 ~ 002)
MTU0C 引脚 (MTIOC0C)	<ul style="list-style-type: none"> POE8# 引脚的输入电平检测 SPOER 寄存器设定 振荡停止检测 通过 POECR5 寄存器追加的条件 <ul style="list-style-type: none"> 12 位 A/D 转换器的比较器检测 POE0# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测 	MTU0CZE• ((POE8F•POE8E) +(MTUCH0HIZ) +(ICSR6.OSTSTE) +(CMADDMT0ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDMT0ZE•POE0F) +(IC4ADDMT0ZE•POE10E•POE10F) +(IC5ADDMT0ZE•POE11E•POE11F)) (n=000 ~ 002)

表 22.4 高阻抗控制的对象引脚及其条件 (3/3)

引脚	条件	详细内容
MTU0D 引脚 (MTIOC0D)	<ul style="list-style-type: none"> POE8# 引脚的输入电平检测 SPOER 寄存器设定 振荡停止检测 通过 POECR5 寄存器追加的条件 <ul style="list-style-type: none"> 12 位 A/D 转换器的比较器检测 POE0# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测 	MTU0DZE• ((POE8F•POE8E) +(MTUCH0HIZ) +(ICSR6.OSTSTE) +(CMADDMT0ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDMT0ZE•POE0F) +(IC4ADDMT0ZE•POE10E•POE10F) +(IC5ADDMT0ZE•POE11E•POE11F)) (n=000 ~ 002)
GPT0 引脚 (GTIOC0A (PD7)、 GTIOC0B (PD6))	<ul style="list-style-type: none"> POE10# 引脚的输入电平检测 SPOER 寄存器设定 振荡停止检测 通过 POECR6 寄存器追加的条件 <ul style="list-style-type: none"> 12 位 A/D 转换器的比较器检测 POE0# 引脚、POE8# 引脚、POE11# 引脚的输入电平检测 	GPT0ABZE• ((POE10F•POE10E) +(GPT01HIZ) +(ICSR6.OSTSTE) +(CMADDGPT01ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDGPT01ZE•POE0F) +(IC3ADDGPT01ZE•POE8E•POE8F) +(IC5ADDGPT01ZE•POE11E•POE11F)) (n=000 ~ 002)
GPT1 引脚 (GTIOC1A (PD5)、 GTIOC1B (PD4))	<ul style="list-style-type: none"> POE10# 引脚的输入电平检测 SPOER 寄存器设定 振荡停止检测 通过 POECR6 寄存器追加的条件 <ul style="list-style-type: none"> 12 位 A/D 转换器的比较器检测 POE0# 引脚、POE8# 引脚、POE11# 引脚的输入电平检测 	GPT1ABZE• ((POE10F•POE10E) +(GPT01HIZ) +(ICSR6.OSTSTE) +(CMADDGPT01ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDGPT01ZE•POE0F) +(IC3ADDGPT01ZE•POE8E•POE8F) +(IC5ADDGPT01ZE•POE11E•POE11F)) (n=000 ~ 002)
GPT2 引脚 (GTIOC2A (PD3)、 GTIOC2B (PB6、PB7))	<ul style="list-style-type: none"> POE11# 引脚的输入电平检测 SPOER 寄存器设定 振荡停止检测 通过 POECR6 寄存器追加的条件 <ul style="list-style-type: none"> 12 位 A/D 转换器的比较器检测 POE0# 引脚、POE8# 引脚、POE10# 引脚的输入电平检测 	GPT2ABZE• ((POE11F•POE11E) +(GPT23HIZ) +(ICSR6.OSTSTE) +(CMADDGPT23ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDGPT23ZE•POE0F) +(IC3ADDGPT23ZE•POE8E•POE8F) +(IC4DDGPT23ZE•POE10E•POE10F)) (n=000 ~ 002)
GPT3 引脚 (GTIOC3A、GTIOC3B)	<ul style="list-style-type: none"> POE11# 引脚的输入电平检测 SPOER 寄存器设定 振荡停止检测 通过 POECR6 寄存器追加的条件 <ul style="list-style-type: none"> 12 位 A/D 转换器的比较器检测 POE0# 引脚、POE8# 引脚、POE10# 引脚的输入电平检测 	GPT3ABZE• ((POE11F•POE11E) +(GPT23HIZ) +(ICSR6.OSTSTE) +(CMADDGPT23ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDGPT23ZE•POE0F) +(IC3ADDGPT23ZE•POE8E•POE8F) +(IC4ADDGPT23ZE•POE10E•POE10F)) (n=000 ~ 002)

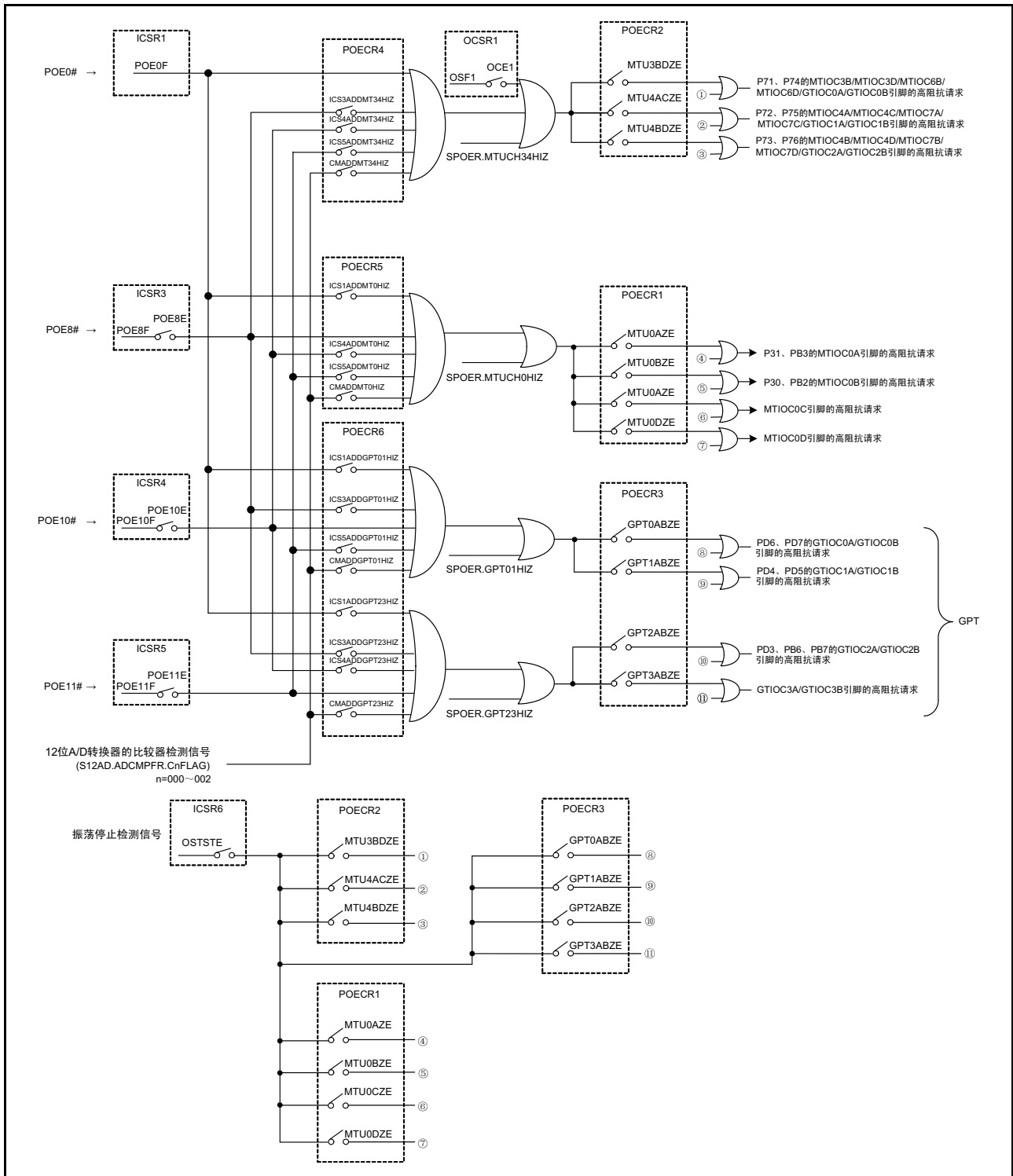


图 22.2 高阻抗控制的对象引脚和条件

22.3.1 输入电平的检测

当 POE0#、POE8#、POE10#、POE11# 引脚产生 ICSR1、ICSR3 ~ ICSR5 寄存器设定的输入条件时，就将 MTU 的互补 PWM 输出引脚 (MTU3、MTU4 引脚或者 MTU6、MTU7 引脚) 以及 MTU0 引脚、GPT 引脚置为高阻抗状态。但是，在 MTU 的互补 PWM 输出引脚以及 MTU0 引脚、GPT 引脚中为多路复用的引脚即使在没有选择 MTU、GPT 功能时，也能置为高阻抗。

(1) 下降沿检测

当 POE0#、POE8#、POE10#、POE11# 引脚从 High 电平变为 Low 电平时，将 MTU 的互补 PWM 输出引脚以及 MTU0 引脚、GPT 引脚中为多路复用的引脚置为高阻抗状态。

从 POE0#、POE8#、POE10#、POE11# 引脚输入到将引脚置为高阻抗的时序例子如图 22.3 所示。

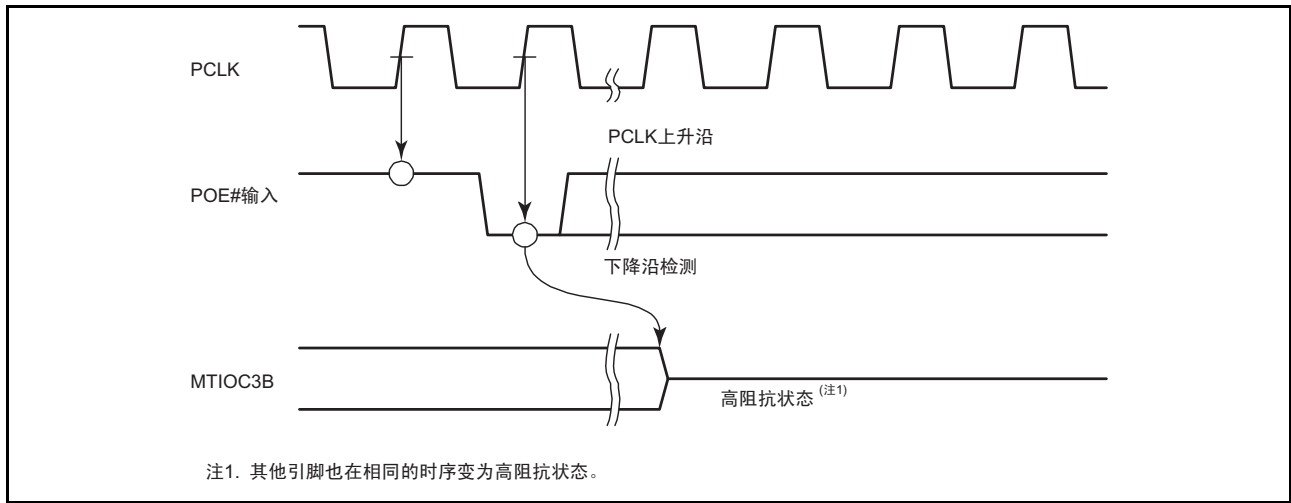


图 22.3 下降沿检测

(2) Low 电平检测

Low 电平检测如图 22.4 所示。通过 ICSR1、ICSR3 ~ ICSR5 寄存器设定的采样时钟连续对 Low 电平进行 16 次采样。此时，只要有 1 次检测到 High 电平，就不接受请求。

从采样时钟到 MTU 的互补 PWM 输出引脚以及 MTU0 引脚、GPT 引脚变为高阻抗状态的时序和下降沿检测、Low 电平检测相同。

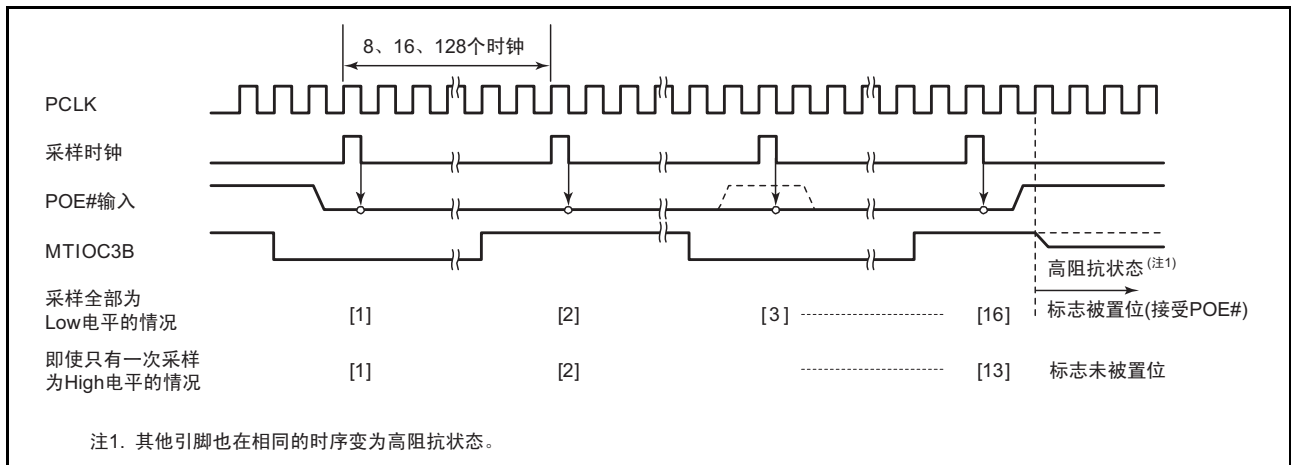


图 22.4 Low 电平检测

22.3.2 输出电平的比较

以 MTIOC3B 和 MTIOC3D 的组合为例，输出电平的比较如图 22.5 所示，其他引脚的组合也一样。

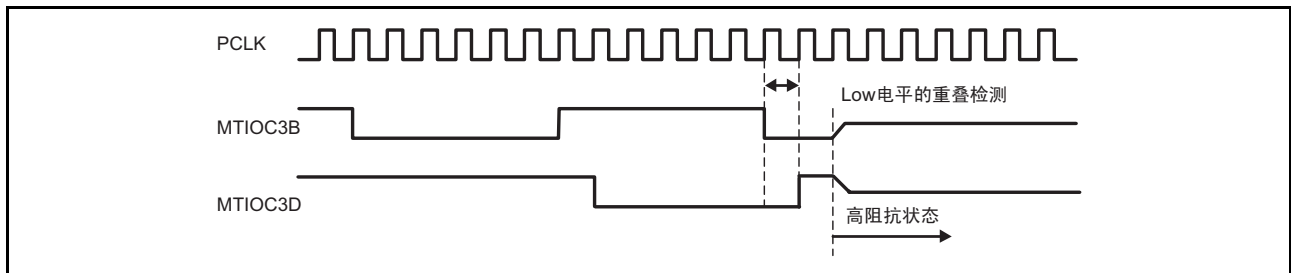


图 22.5 输出电平的比较

22.3.3 通过寄存器进行的高阻抗控制

通过软件端口输出允许寄存器 (SPOER) 直接对 MTU 引脚 (MTU0、MTU3、MTU4、MTU6、MTU7) 和 GPT 引脚进行高阻抗控制。

通过将 SPOER.CH34HIZ 位置“1”，将端口输出允许控制寄存器 2 (POECR2) 设定的 MTU3、MTU4 引脚置为高阻抗状态。

对于其他引脚，通过 SPOER 寄存器的位设定，同样也能进行高阻抗控制。

22.3.4 通过振荡停止检测进行的高阻抗控制

设定输入电平控制/状态寄存器 6 (ICSR6) 的 OSTSTE 位，并且在通过时钟发生电路的振荡停止检测电路检测到振荡停止时，将端口输出允许控制寄存器 2 (POECR2) 设定的 MTU3、MTU4 或者 MTU6、MTU7 引脚、端口输出允许控制寄存器 1 (POECR1) 设定的 MTU0 引脚以及端口输出允许控制寄存器 3 (POECR3) 设定的 GPT 引脚置为高阻抗状态。

22.3.5 通过比较器检测进行的高阻抗控制

能通过 12 位 A/D 转换器的内部比较器检测，将 MTU 的互补 PWM 输出引脚、MTU0 引脚和 GPT 引脚置为高阻抗状态。

例如，通过将 POECR4.CMADDMT34ZE 位置“1”，将 S12AD.ADCMPFR.CjFLAG (j=000 ~ 002) 标志追加到 MTU3、MTU4 引脚的高阻抗控制条件，就在进行比较器检测时将端口输出允许控制寄存器 2 (POECR2) 设定的 MTU3、MTU4 引脚置为高阻抗状态。

对于其他引脚，通过设定 POECR1 ~ POECR6 寄存器，同样也能进行高阻抗控制。

22.3.6 高阻抗控制条件的追加功能

能通过设定端口输出允许控制寄存器 4 ~ 6 (POECR4 ~ 6)，追加 MTU 的互补 PWM 输出引脚、MTU0 引脚和 GPT 引脚的高阻抗控制条件。

例如，能将以下条件追加到 MTU3、MTU4 引脚的高阻抗控制条件。

- 将 POECR4.CMADDMT34ZE 位置“1”，追加比较器检测
- 将 POECR4.IC3ADDMT34ZE 位置“1”，追加通过 POE8# 引脚进行的输入电平检测
- 将 POECR4.IC4ADDMT34ZE 位置“1”，追加通过 POE10# 引脚进行的输入电平检测
- 将 POECR4.IC5ADDMT34ZE 位置“1”，追加通过 POE11# 引脚进行的输入电平检测

对于其他引脚，通过设定 POECR4 ~ POECR6 寄存器，同样也能进行高阻抗控制条件的追加。

22.3.7 高阻抗状态的解除

因输入电平检测而变为高阻抗状态的引脚通过复位恢复到初始状态，或者通过清除 ICSR1.POE0F 标志、ICSR3.POE8F 标志、ICSR4.POE10F 标志、ICSR5.POE11F 标志解除高阻抗状态。但是，在通过 ICSR1.POE0M[1:0] 位、ICSR3.POE8M[1:0] 位、ICSR4.POE10M[1:0] 位、ICSR5.POE11M[1:0] 位设定为 Low 电平采样时，如果不是在 POE0#、POE8#、POE10#、POE11# 引脚输入 High 电平后对 High 电平进行了采样以后，即使给标志写“0”也无效并且标志不变为“0”。

因输出电平检测而变为高阻抗状态引脚通过复位恢复到初始状态，或者通过将 OCSR1.OSF1 标志置“0”，解除高阻抗状态。如果要将 OCSR1.OSF1 标志、OCSR2.OSF2 标志置“0”，必须先设定为“从引脚输出无效电平”。能通过设定 MTU 内的寄存器、GPT 内的寄存器以及 ALR1 寄存器输出无效电平。

因比较器检测而变为高阻抗状态引脚通过复位恢复到初始状态，或者通过将 ICSR1.POE0F 标志、ICSR3.POE8F 标志、ICSR4.POE10F 标志、ICSR5.POE11F 标志置“0”，解除高阻抗状态。而且，还能通过将 S12AD.ADCMPFR.CnFLAG (n=000 ~ 002) 标志置“0”，解除高阻抗状态。但是，如果比较器输入不在设定的范围内，即使给标志写“0”也无效并且标志不变为“0”。

22.4 中断

如果在进行输入电平的检测或者输出电平的比较时条件匹配，POE3 就能在产生中断请求后发生中断。中断的种类和产生中断请求的条件如表 22.5 所示。

表 22.5 中断请求的种类和条件

名称	中断源	中断标志	条件
OE11	输出允许中断 1	POE0F、OSF1	PIE1•POE0F+OIE1•OSF1
OE13	输出允许中断 3	POE8F	PIE3•POE8F
OE14	输出允许中断 4	POE10F、POE11F	PIE4•POE10F+PIE5•POE11F

22.5 使用时的注意事项

在使用 POE3 时，不能转移到软件待机模式和深度软件待机模式。因为在软件待机模式和深度软件待机模式中 POE3 停止运行，所以不能对引脚进行高阻抗控制。

23. 通用 PWM 定时器 (GPT)

RX63T 群内置由 4 个通道的 16 位定时器构成的通用 PWM 定时器 (GPT)。GPT 最大能以 100MHz 运行。

23.1 概要

GPT 的规格如表 23.1、GPT 的功能一览表如表 23.2、GPT 的框图如图 23.1 所示。

表 23.1 GPT 的规格

项目	内容
功能	<ul style="list-style-type: none"> • 16 位 ×4 个通道 • 各计数器进行递增计数或者递减计数 (锯齿波)、递增 / 递减计数 (三角波) • 能给各通道选择独立的时钟源 • 每个通道有 2 个输入 / 输出引脚 • 每个通道有 2 个用于输出比较 / 输入捕捉的寄存器 • 4 个缓冲寄存器, 分别对应各通道的 2 个输出比较 / 输入捕捉寄存器。在不进行缓冲运行时, 作为比较寄存器运行。 • 在输出比较运行时, 能分别在波峰 / 波谷进行缓冲运行, 并且生成左右不对称的 PWM 波形。 • 给各通道装载用于帧周期的寄存器 (能通过上溢 / 下溢中断) • 能同步运行各计数器 • 同步运行模式 (支持同时或者以任意的时序进行相位移位) • 能在 PWM 运行时生成死区时间 • 组合 3 个计数器, 能生成带死区时间的三相 PWM 波形 • 能通过外部 / 内部触发开始、清除或者停止计数 • 内部触发源有比较器检测、软件和比较匹配 • 能通过分频定时器模块时钟 (PCLKA) 的计数时钟来测量被分频的 IWDTC 专用低速时钟 (IWDTCCLK) 的边沿 (振荡异常检测)

表 23.2 GPT 的功能一览表 (1/2)

项目	GPT0	GPT1	GPT2	GPT3
计数时钟	PCLKA PCLKA/2 PCLKA/4 PCLKA/8	PCLKA PCLKA/2 PCLKA/4 PCLKA/8	PCLKA PCLKA/2 PCLKA/4 PCLKA/8	PCLKA PCLKA/2 PCLKA/4 PCLKA/8
输出比较 / 输入捕捉寄存器 (GTCCR)	GTCCRA GTCCRB	GTCCRA GTCCRB	GTCCRA GTCCRB	GTCCRA GTCCRB
比较 / 缓冲寄存器	GTCCRC GTCCRD GTCCRE GTCCRF	GTCCRC GTCCRD GTCCRE GTCCRF	GTCCRC GTCCRD GTCCRE GTCCRF	GTCCRC GTCCRD GTCCRE GTCCRF
周期设定寄存器	GTPR	GTPR	GTPR	GTPR
周期设定缓冲寄存器	GTPBR GTPDBR	GTPBR GTPDBR	GTPBR GTPDBR	GTPBR GTPDBR
输入 / 输出引脚	GTIOC0A GTIOC0B	GTIOC1A GTIOC1B	GTIOC2A GTIOC2B	GTIOC3A GTIOC3B

表 23.2 GPT 的功能一览表 (2/2)

项目		GPT0	GPT1	GPT2	GPT3
外部触发输入引脚		GTETRG			
计数器清除源		GTPR 的比较匹配、输入捕捉、比较器检测、GTETRG 引脚输入、GTIOC3A/B 引脚输入、GTIOC3A/B 的内部输出 (输出比较)			
比较匹配 输出	低电平输出	○	○	○	○
	高电平输出	○	○	○	○
	交替输出	○	○	○	○
输入捕捉功能		○	○	○	○
同步运行		○	○	○	○
相位移位开始		○	○	○	○
死区时间自动附加功能		○	○	○	○
PWM 模式		○	○	○	○
缓冲运行		○	○	○	○
单触发运行		○	○	○	○
DTC 的启动		全部的中断源			
A/D 转换开始触发		GTADTRA、GTADTRB 的比较匹配	GTADTRA、GTADTRB 的比较匹配	GTADTRA、GTADTRB 的比较匹配	GTADTRA、GTADTRB 的比较匹配
中断源		5 个中断源 • GTCCRA 比较匹配 / 输入捕捉 (GTCIA0) • GTCCRB 比较匹配 / 输入捕捉 (GTCIB0) • GTCCRC 比较匹配 / GTCCRD 比较匹配 / 死区时间错误 (GTICIC0) • GTCCRE 比较匹配 / GTCCRF 比较匹配 (GTCIE0) • GTCNT 上溢 (GTPR 比较匹配) / GTCNT 下溢 (GTCIV0)	5 个中断源 • GTCCRA 比较匹配 / 输入捕捉 (GTCIA1) • GTCCRB 比较匹配 / 输入捕捉 (GTCIB1) • GTCCRC 比较匹配 / GTCCRD 比较匹配 / 死区时间错误 (GTICIC1) • GTCCRE 比较匹配 / GTCCRF 比较匹配 (GTCIE1) • GTCNT 上溢 (GTPR 比较匹配) / GTCNT 下溢 (GTCIV1)	5 个中断源 • GTCCRA 比较匹配 / 输入捕捉 (GTCIA2) • GTCCRB 比较匹配 / 输入捕捉 (GTCIB2) • GTCCRC 比较匹配 / GTCCRD 比较匹配 / 死区时间错误 (GTICIC2) • GTCCRE 比较匹配 / GTCCRF 比较匹配 (GTCIE2) • GTCNT 上溢 (GTPR 比较匹配) / GTCNT 下溢 (GTCIV2)	5 个中断源 • GTCCRA 比较匹配 / 输入捕捉 (GTCIA3) • GTCCRB 比较匹配 / 输入捕捉 (GTCIB3) • GTCCRC 比较匹配 / GTCCRD 比较匹配 / 死区时间错误 (GTICIC3) • GTCCRE 比较匹配 / GTCCRF 比较匹配 (GTCIE3) • GTCNT 上溢 (GTPR 比较匹配) / GTCNT 下溢 (GTCIV3)
共同中断源		外部触发 / IWDTCCLK 计数功能中断 (LOCOI)			
中断减少功能		减少 GTCNT 上溢 (GTPR 比较匹配) / GTCNT 下溢 (GTCIV0) 中断 (有联动其他中断和 A/D 转换开始请求的功能)	减少 GTCNT 上溢 (GTPR 比较匹配) / GTCNT 下溢 (GTCIV0) 中断 (有联动其他中断和 A/D 转换开始请求的功能)	减少 GTCNT 上溢 (GTPR 比较匹配) / GTCNT 下溢 (GTCIV0) 中断 (有联动其他中断和 A/D 转换开始请求的功能)	减少 GTCNT 上溢 (GTPR 比较匹配) / GTCNT 下溢 (GTCIV3) 中断 (有联动其他中断和 A/D 转换开始请求的功能)

○: 能

—: 不能

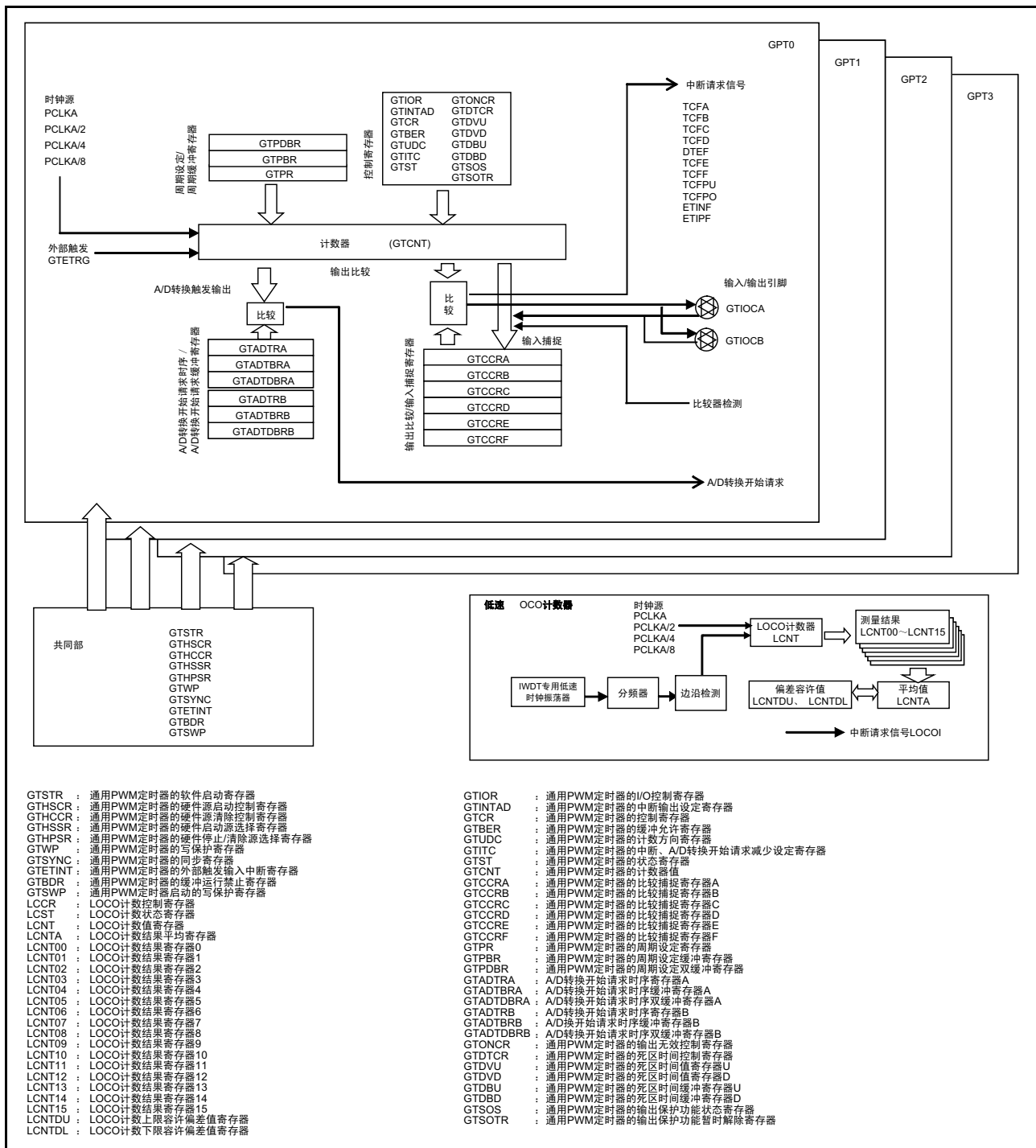


图 23.1 GPT 的框图

GPT 使用的输入 / 输出引脚如表 23.3 所示。

表 23.3 GPT 的输入 / 输出引脚

通道	引脚名	输入 / 输出	功能
共同	GTETRG	输入	外部触发的输入引脚
GPT0	GTIOC0A	输入 / 输出	GTCCRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚
	GTIOC0B	输入 / 输出	GTCCRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚
GPT1	GTIOC1A	输入 / 输出	GTCCRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚
	GTIOC1B	输入 / 输出	GTCCRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚
GPT2	GTIOC2A	输入 / 输出	GTCCRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚
	GTIOC2B	输入 / 输出	GTCCRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚
GPT3	GTIOC3A	输入 / 输出	GTCCRA 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚
	GTIOC3B	输入 / 输出	GTCCRB 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚

23.2 寄存器说明

23.2.1 通用 PWM 定时器的软件启动寄存器 (GTSTR)

地址 000C 2000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CST3	CST2	CST1	CST0
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	CST0	GPT0.GTCNT 计数开始位	0: 停止计数 1: 计数	R/W
b1	CST1	GPT1.GTCNT 计数开始位		R/W
b2	CST2	GPT2.GTCNT 计数开始位		R/W
b3	CST3	GPT3.GTCNT 计数开始位		R/W
b15-b4	—	保留位	读写值都为“0”。	R/W

GTSTR 寄存器是设定 GPTn.GTCNT (n=0 ~ 3) 计数器的计数 / 停止计数的寄存器。

CSTn 位 (GPTn.GTCNT 计数开始位) (n=0 ~ 3)

此位选择 GPTn.GTCNT 计数器的计数 / 停止计数。

忽略写操作被禁止的 CSTn 位 (通过通用 PWM 定时器启动的写保护寄存器 (GTSWP 的 SWPn 位进行设定) 的写操作。

也能通过 GTHSCR 寄存器设定由硬件源引起的计数器的计数 / 停止计数。如果通过硬件源启动计数, 此位就自动变为“1”, 如果通过硬件源停止计数, 此位就自动变为“0”。

23.2.2 通用 PWM 定时器的硬件源启动控制寄存器 (GTHSCR)

地址 000C 2004h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CPHW3[1:0]		CPHW2[1:0]		CPHW1[1:0]		CPHW0[1:0]		CSHW3[1:0]		CSHW2[1:0]		CSHW1[1:0]		CSHW0[1:0]	
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
b1-b0	CSHW0[1:0]	GPT0.GTCNT 硬件源计数开始位	0 0: 未通过硬件源开始计数 0 1: 在硬件源的上升沿开始计数 1 0: 在硬件源的下降沿开始计数 1 1: 在硬件源的双边沿开始计数	R/W
b3-b2	CSHW1[1:0]	GPT1.GTCNT 硬件源计数开始位		R/W
b5-b4	CSHW2[1:0]	GPT2.GTCNT 硬件源计数开始位		R/W
b7-b6	CSHW3[1:0]	GPT3.GTCNT 硬件源计数开始位		R/W
b9-b8	CPHW0[1:0]	GPT0.GTCNT 硬件源计数开始位	0 0: 未通过硬件源停止计数 0 1: 在硬件源的上升沿停止计数 1 0: 在硬件源的下降沿停止计数 1 1: 在硬件源的双边沿停止计数	R/W
b11-b10	CPHW1[1:0]	GPT1.GTCNT 硬件源计数开始位		R/W
b13-b12	CPHW2[1:0]	GPT2.GTCNT 硬件源计数开始位		R/W
b15-b14	CPHW3[1:0]	GPT3.GTCNT 硬件源计数开始位		R/W

GTHSCR 寄存器是设定通过硬件源引起的 GPTn.GTCNT (n=0 ~ 3) 计数器的计数 / 停止计数的寄存器。如果同时发生由硬件源引起的 GPTn.GTCNT 计数器的计数和停止计数, 就优先计数。

CSHWn[1:0] 位 (GPTn.GTCNT 硬件源计数开始位) (n=0 ~ 3)

通过硬件源启动 GPTn.GTCNT 计数器的计数。

如果通过硬件源启动计数, GTSTR 寄存器的对应位就自动变为“1”。

通过 GTHSSR 寄存器选择硬件源。

CPHWn[1:0] 位 (GPTn.GTCNT 硬件源计数停止位) (n=0 ~ 3)

通过硬件源停止 GPTn.GTCNT 计数器的计数。

如果通过硬件源停止计数, GTSTR 寄存器的对应位就自动变为“0”。

通过 GTHPSR 寄存器选择硬件源。

23.2.3 通用 PWM 定时器的硬件源清除控制寄存器 (GTHCCR)

地址 000C 2006h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	CCSW3	CCSW2	CCSW1	CCSW0	CCHW3[1:0]	CCHW2[1:0]	CCHW1[1:0]	CCHW0[1:0]				
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
b1-b0	CCHW0[1:0]	GPT0.GTCNT 硬件源计数器清除位	0 0: 未通过硬件源清除计数器 0 1: 在硬件源的上升沿清除计数器 1 0: 在硬件源的下降沿清除计数器 1 1: 在硬件源的双边沿清除计数器	R/W
b3-b2	CCHW1[1:0]	GPT1.GTCNT 硬件源计数器清除位		R/W
b5-b4	CCHW2[1:0]	GPT2.GTCNT 硬件源计数器清除位		R/W
b7-b6	CCHW3[1:0]	GPT3.GTCNT 硬件源计数器清除位		R/W
b8	CCSW0	GPT0.GTCNT 计数器清除位	写“1”时清除计数器。写“1”后自动恢复“0”。 读取值为“0”。	R/W
b9	CCSW1	GPT1.GTCNT 计数器清除位		R/W
b10	CCSW2	GPT2.GTCNT 计数器清除位		R/W
b11	CCSW3	GPT3.GTCNT 计数器清除位		R/W
b15-b12	—	保留位	读写值都为“0”。	

GTHCCR 寄存器是设定通过硬件源清除 GPTn.GTCNT (n=0 ~ 3) 计数器的寄存器。

当设定为通过硬件源清除 GPTn.GTCNT (n=0 ~ 3) 计数器时, 不管 GPTn.GTCNT 计数器正在计数 (GTSTR.CSTn=1) 还是停止计数 (GTSTR.CSTn=0), 都通过硬件源进行计数器清除。

CCHWn[1:0] 位 (GPTn.GTCNT 硬件源计数器清除位) (n=0 ~ 3)

通过硬件源清除 GPTn.GTCNT 计数器。

通过 GTHPSR 寄存器选择硬件源。

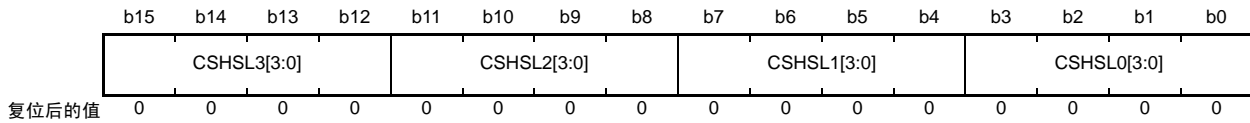
当 CCHWn[1:0] 位为“01b”、“10b”、“11b”时, 重复接受硬件源。

CCSWn 位 (GPTn.GTCNT 计数器清除位) (n=0 ~ 3)

如果写“1”, GPTn.GTCNT 计数器就变为“0”。写“1”后自动恢复“0”。读取值为“0”。

23.2.4 通用 PWM 定时器的硬件启动源选择寄存器 (GTHSSR)

地址 000C 2008h



位	符号	位名	功能	R/W
b3-b0	CSHSL0[3:0]	GPT0.GTCNT 硬件计数器启动源选择位	b3 b0 0 0 0 0: AN000 的比较器检测 0 0 0 1: AN001 的比较器检测 0 0 1 0: AN002 的比较器检测 0 0 1 1: 不能设定 0 1 1 1: 不能设定 1 0 0 0: GTIOC3A 引脚输入 1 0 0 1: GTIOC3B 引脚输入 1 0 1 0: GTIOC3A 内部输出 (注1) (输出比较) 1 0 1 1: GTIOC3B 内部输出 (注1) (输出比较) 1 1 0 0: GTETRG 引脚输入 上述以外: 不能设定	R/W
b7-b4	CSHSL1[3:0]	GPT1.GTCNT 硬件计数器启动源选择位	b7 b4 0 0 0 0: AN000 的比较器检测 0 0 0 1: AN001 的比较器检测 0 0 1 0: AN002 的比较器检测 0 0 1 1: 不能设定 0 1 1 1: 不能设定 1 0 0 0: GTIOC3A 引脚输入 1 0 0 1: GTIOC3B 引脚输入 1 0 1 0: GTIOC3A 内部输出 (注1) (输出比较) 1 0 1 1: GTIOC3B 内部输出 (注1) (输出比较) 1 1 0 0: GTETRG 引脚输入 上述以外: 不能设定	R/W
b11-b8	CSHSL2[3:0]	GPT2.GTCNT 硬件计数器启动源选择位	b11 b8 0 0 0 0: AN000 的比较器检测 0 0 0 1: AN001 的比较器检测 0 0 1 0: AN002 的比较器检测 0 0 1 1: 不能设定 0 1 1 1: 不能设定 1 0 0 0: GTIOC3A 引脚输入 1 0 0 1: GTIOC3B 引脚输入 1 0 1 0: GTIOC3A 内部输出 (注1) (输出比较) 1 0 1 1: GTIOC3B 内部输出 (注1) (输出比较) 1 1 0 0: GTETRG 引脚输入 上述以外: 不能设定	R/W

位	符号	位名	功能	R/W
b15-b12	CSHSL3[3:0]	GPT3.GTCNT 硬件计数器启动源选择位	b15 b12 0 0 0 0: AN000 的比较器检测 0 0 0 1: AN001 的比较器检测 0 0 1 0: AN002 的比较器检测 0 0 1 1: 不能设定 0 1 1 1: 不能设定 1 0 0 0: GTIOC3A 引脚输入 1 0 0 1: GTIOC3B 引脚输入 1 0 1 0: 不能设定 1 0 1 1: 不能设定 1 1 0 0: GTETRG 引脚输入 上述以外: 不能设定	R/W

注 1. 在 48 引脚版中无输出引脚，但是能用作启动源。

GTHSSR 寄存器是设定 GPTn.GTCNT (n=0 ~ 3) 计数器的计数器启动硬件源的寄存器。
 必须在将 GTHSCR.CSHWn[1:0] 位置“0”后更改硬件源。

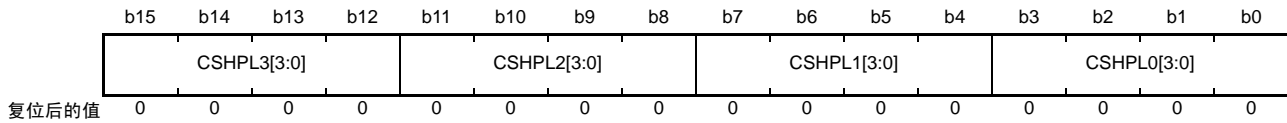
CSHSLn[3:0] 位 (GPTn.GTCNT 硬件计数器启动源选择位) (n=0 ~ 3)

这些位选择 GPTn.GTCNT 计数器的计数器启动硬件源。

当选择“1000b”作为硬件源时，必须将 GPT3.GTIOR.GTIOA[5:0] 位的 b5 置“0”，将 GPT3.GTONCR.OAE 位置“0”。当选择“1001b”作为硬件源时，必须将 GPT3.GTIOR.GTIOB[5:0] 位的 b5 置“0”，将 GPT3.GTONCR.OBE 位置“0”。

23.2.5 通用 PWM 定时器的硬件停止 / 清除源选择寄存器 (GTHPSR)

地址 000C 200Ah



位	符号	位名	功能	R/W
b3-b0	CSHPL0[3:0]	GPT0.GTCNT 硬件计数器停止 / 清除源选择位	b3 b0 0 0 0 0: AN000 的比较器检测 0 0 0 1: AN001 的比较器检测 0 0 1 0: AN002 的比较器检测 0 0 1 1: 不能设定 0 1 1 1: 不能设定 1 0 0 0: GTIOC3A 引脚输入 1 0 0 1: GTIOC3B 引脚输入 1 0 1 0: GTIOC3A 内部输出 (输出比较) 1 0 1 1: GTIOC3B 内部输出 (输出比较) 1 1 0 0: GTETRG 引脚输入 上述以外: 不能设定	R/W
b7-b4	CSHPL1[3:0]	GPT1.GTCNT 硬件计数器停止 / 清除源选择位	b7 b4 0 0 0 0: AN000 的比较器检测 0 0 0 1: AN001 的比较器检测 0 0 1 0: AN002 的比较器检测 0 0 1 1: 不能设定 0 1 1 1: 不能设定 1 0 0 0: GTIOC3A 引脚输入 1 0 0 1: GTIOC3B 引脚输入 1 0 1 0: GTIOC3A 内部输出 (输出比较) 1 0 1 1: GTIOC3B 内部输出 (输出比较) 1 1 0 0: GTETRG 引脚输入 上述以外: 不能设定	R/W
b11-b8	CSHPL2[3:0]	GPT2.GTCNT 硬件计数器停止 / 清除源选择位	b11 b8 0 0 0 0: AN000 的比较器检测 0 0 0 1: AN001 的比较器检测 0 0 1 0: AN002 的比较器检测 0 0 1 1: 不能设定 0 1 1 1: 不能设定 1 0 0 0: GTIOC3A 引脚输入 1 0 0 1: GTIOC3B 引脚输入 1 0 1 0: GTIOC3A 内部输出 (输出比较) 1 0 1 1: GTIOC3B 内部输出 (输出比较) 1 1 0 0: GTETRG 引脚输入 上述以外: 不能设定	R/W

位	符号	位名	功能	R/W
b15-b12	CSHPL3[3:0]	GPT3.GTCNT 硬件计数器 停止 / 清除源选择位	b15 b12 0 0 0 0: AN000 的比较器检测 0 0 0 1: AN001 的比较器检测 0 0 1 0: AN002 的比较器检测 0 0 1 1: 不能设定 0 1 1 1: 不能设定 1 0 0 0: GTIOC3A 引脚输入 1 0 0 1: GTIOC3B 引脚输入 1 0 1 0: 不能设定 1 0 1 1: 不能设定 1 1 0 0: GTETRG 引脚输入 上述以外: 不能设定	R/W

GTHPSR 寄存器是设定停止 / 清除 GPTn.GTCNT (n=0 ~ 3) 计数器的硬件源的寄存器。
 必须在将 GTHSCR.CPHWn[1:0] 位、GTHCCR.CCHWn[1:0] 位置“0”后更改清除源。

CSHPLn[3:0] 位 (GPTn.GTCNT 硬件计数器停止 / 清除源选择位) (n=0 ~ 3)

这些位选择 GPTn.GTCNT 计数器的计数器停止 / 清除的硬件源。

23.2.6 通用 PWM 定时器的写保护寄存器 (GTWP)

地址 000C 200Ch

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	—	—	WP3	WP2	WP1	WP0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	WP0	GPT0 寄存器的写禁止位	0: 允许写寄存器 1: 禁止写寄存器	R/W
b1	WP1	GPT1 寄存器的写禁止位		R/W
b2	WP2	GPT2 寄存器的写禁止位		R/W
b3	WP3	GPT3 寄存器的写禁止位		R/W
b15-b4	—	保留位	读写值都为“0”。	R/W

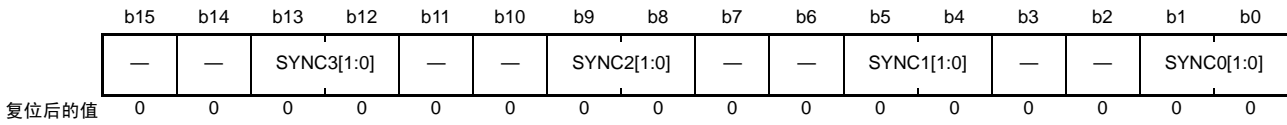
为了防止误写，GTWP 寄存器允许或者禁止写寄存器。

WPn 位 (GPTn 寄存器的写禁止位) (n=0 ~ 3)

这些位选择允许或者禁止写 GPTn 寄存器。

23.2.7 通用 PWM 定时器的同步寄存器 (GTSYNC)

地址 000C 200Eh



位	符号	位名	功能	R/W
b1-b0	SYNC0[1:0]	GPT0.GTCNT 的同步计数器清除源选择位	b1 b0 0 0: 通过 GPT0 的清除源清除 GPT0.GTCNT (不同步清除) 0 1: 通过 GPT1 的清除源同步清除 GPT0.GTCNT 1 0: 通过 GPT2 的清除源同步清除 GPT0.GTCNT 1 1: 通过 GPT3 的清除源同步清除 GPT0.GTCNT	R/W
b3-b2	—	保留位	读写值都为“0”。	R/W
b5-b4	SYNC1[1:0]	GPT1.GTCNT 的同步计数器清除源选择位	b5 b4 0 0: 通过 GPT0 的清除源清除 GPT1.GTCNT (不同步清除) 0 1: 通过 GPT1 的清除源同步清除 GPT1.GTCNT 1 0: 通过 GPT2 的清除源同步清除 GPT1.GTCNT 1 1: 通过 GPT3 的清除源同步清除 GPT1.GTCNT	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W
b9-b8	SYNC2[1:0]	GPT2.GTCNT 的同步计数器清除源选择位	b9 b8 0 0: 通过 GPT0 的清除源清除 GPT2.GTCNT (不同步清除) 0 1: 通过 GPT1 的清除源同步清除 GPT2.GTCNT 1 0: 通过 GPT2 的清除源同步清除 GPT2.GTCNT 1 1: 通过 GPT3 的清除源同步清除 GPT2.GTCNT	R/W
b11-b10	—	保留位	读写值都为“0”。	R/W
b13-b12	SYNC3[1:0]	GPT3.GTCNT 的同步计数器清除源选择位	b13 b12 0 0: 通过 GPT0 的清除源同步清除 GPT3.GTCNT 0 1: 通过 GPT1 的清除源同步清除 GPT3.GTCNT 1 0: 通过 GPT2 的清除源同步清除 GPT3.GTCNT 1 1: 通过 GPT3 的清除源清除 GPT3.GTCNT (不同步清除)	R/W
b15-b14	—	保留位	读写值都为“0”。	R/W

GTSYNC 寄存器是设定同步清除或者同步运行的 GPTn.GTCNT 计数器清除源的寄存器。必须在 GPTn.GTCNT (n=0 ~ 3) 计数器停止计数的状态下进行写操作。

SYNCn[1:0] 位 (GPTn.GTCNT 的同步计数器清除源选择位) (n=0 ~ 3)

这些位选择通过哪个通道的计数器清除源清除 GPTn.GTCNT 计数器。如果设定 SYNCn[1:0] 位, 必须事先将 GPTn.GTCR.CCLR[1:0] 位置“11b” (通过正在进行同步清除 / 同步运行的其他计数器的清除源清除)。

23.2.8 通用 PWM 定时器的外部触发输入中断寄存器 (GTETINT)

地址 000C 2010h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	ETINF	ETIPF	—	—	—	—	—	—	ETINEN	ETIPEN
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	ETIPEN	外部触发输入上升沿的中断请求允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b1	ETINEN	外部触发输入下降沿的中断请求允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W
b8	ETIPF	外部触发输入上升沿的中断请求标志	0: 无中断请求 1: 有中断请求	R/W (注 1)
b9	ETINF	外部触发输入下降沿的中断请求标志	0: 无中断请求 1: 有中断请求	R/W (注 1)
b15-b10	—	保留位	读写值都为“0”。	R/W

注 1. 只能写用于将标志置“0”的“0”。

GTETINT 寄存器是设定禁止或者允许由外部触发输入引脚 (GTETRG) 产生的中断的寄存器。中断请求作为 LOCOI 中断请求产生。

ETIPEN 位 (外部触发输入上升沿的中断请求允许位)

此位选择允许或者禁止由外部触发输入的上升沿产生中断请求。

ETINEN 位 (外部触发输入下降沿的中断请求允许位)

此位选择允许或者禁止由外部触发输入的下降沿产生中断请求。

ETIPF 标志 (外部触发输入上升沿的中断请求标志)

这是由外部触发输入的上升沿产生中断请求的标志。

[为“1”的条件]

- 在检测到外部触发输入的上升沿时

[为“0”的条件]

- 给 ETIPF 标志写“0”时

ETINF 标志 (外部触发输入下降沿的中断请求标志)

这是由外部触发输入的下降沿产生中断请求的标志。

[为“1”的条件]

- 在检测到外部触发输入的下降沿时

[为“0”的条件]

- 给 ETINF 标志写“0”时

23.2.9 通用 PWM 定时器的缓冲运行禁止寄存器 (GTBDR)

地址 000C 2014h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BD3[3]	BD3[2]	BD3[1]	BD3[0]	BD2[3]	BD2[2]	BD2[1]	BD2[0]	BD1[3]	BD1[2]	BD1[1]	BD1[0]	BD0[3]	BD0[2]	BD0[1]	BD0[0]
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	BD0[0]	GPT0.GTCCR 缓冲运行禁止位	0: 允许缓冲运行 1: 禁止缓冲运行	R/W
b1	BD0[1]	GPT0.GTPR 缓冲运行禁止位		R/W
b2	BD0[2]	GPT0.GTADTR 缓冲运行禁止位		R/W
b3	BD0[3]	GPT0.GTDV 缓冲运行禁止位		R/W
b4	BD1[0]	GPT1.GTCCR 缓冲运行禁止位		R/W
b5	BD1[1]	GPT1.GTPR 缓冲运行禁止位		R/W
b6	BD1[2]	GPT1.GTADTR 缓冲运行禁止位		R/W
b7	BD1[3]	GPT1.GTDV 缓冲运行禁止位		R/W
b8	BD2[0]	GPT2.GTCCR 缓冲运行禁止位		R/W
b9	BD2[1]	GPT2.GTPR 缓冲运行禁止位		R/W
b10	BD2[2]	GPT2.GTADTR 缓冲运行禁止位		R/W
b11	BD2[3]	GPT2.GTDV 缓冲运行禁止位		R/W
b12	BD3[0]	GPT3.GTCCR 缓冲运行禁止位		R/W
b13	BD3[1]	GPT3.GTPR 缓冲运行禁止位		R/W
b14	BD3[2]	GPT3.GTADTR 缓冲运行禁止位		R/W
b15	BD3[3]	GPT3.GTDV 缓冲运行禁止位		R/W

GTBDR 寄存器是一次性设定允许或者禁止各通道的缓冲运行的寄存器。如果没有通过 GTBER 寄存器将缓冲运行置为有效，即使将 GTBDR 寄存器的各位置“0”（允许缓冲运行），也不进行缓冲运行。

BDn[0] 位 (GPTn.GTCCR 缓冲运行禁止位) (n=0 ~ 3)

禁止组合 GPTn 的 GTCCRA 寄存器、GTCCRC 寄存器和 GTCCRD 寄存器的缓冲运行以及组合 GPTn 的 GTCCRB 寄存器、GTCCRE 寄存器和 GTCCRF 寄存器的缓冲运行。

BDn[1] 位 (GPTn.GTPR 缓冲运行禁止位) (n=0 ~ 3)

此位禁止组合 GPTn 的 GTPR 寄存器、GTPBR 寄存器和 GTPDBR 寄存器的缓冲运行。

BDn[2] 位 (GPTn.GTADTR 缓冲运行禁止位) (n=0 ~ 3)

此位禁止组合 GPTn 的 GTADTRA 寄存器、GTADTBRA 寄存器和 GTADTDBRA 寄存器的缓冲运行以及 GPTn 的 GTADTRB 寄存器、GTADTBRB 寄存器和 GTADTDBRB 寄存器的缓冲运行。

BDn[3] 位 (GPTn.GTDV 缓冲运行禁止位) (n=0 ~ 3)

此位禁止组合 GPTn 的 GTDVU 寄存器和 GTDBU 寄存器的缓冲运行以及 GPTn 的 GTDVD 寄存器和 GTDBD 寄存器的缓冲运行。

23.2.10 通用 PWM 定时器启动的写保护寄存器 (GTSWP)

地址 000C 2018h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	—	—	SWP3	SWP2	SWP1	SWP0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	SWP0	GTSTR.GST0 位的写禁止位	0: 允许写寄存器 1: 禁止写寄存器	R/W
b1	SWP1	GTSTR.GST1 位的写禁止位		R/W
b2	SWP2	GTSTR.GST2 位的写禁止位		R/W
b3	SWP3	GTSTR.GST3 位的写禁止位		R/W
b15-b4	—	保留位	读写值都为“0”。	R/W

GTSWP 寄存器允许或者禁止写防止误写的 GTSTR 寄存器。

SWPn 位 (GTSTR.CSTn 位的写禁止位) (n=0 ~ 3)

此位允许或者禁止写 GTSTR.CSTn 位。

如果设定为禁止, 就忽视写 GTSTR.CSTn 位。

但是, 当通过 GTHSCR 寄存器设定为因硬件源开始或者停止计数器的计数时, 即使通过设定 SWPn 位禁止了写 GTSTR.CSTn 位, 在因硬件源开始或者停止计数器计数的状态下能写 GTSTR.CSTn 位。

23.2.11 LOCO 计数控制寄存器 (LCCR)

地址 000C 2080h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LPSC[1:0]		TPSC[1:0]		LCNTAT	LCTO[2:0]			—	LCINTO	LCINTD	LCINTC	—	LCNTS	LCNTCR	LCNTE
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	LCNTE	IWDTCLK 计数功能允许位	0: IWDTCLK 计数功能停止 1: IWDTCLK 计数功能运行	R/W
b1	LCNTCR	IWDTCLK 计数值清除位	写“1”后将 LCNT 寄存器置“0”。写“1”后自动回复“0”。读取值为“0”。	R/W
b2	LCNTS	IWDTCLK 计数值设定位	写“1”后给 LCNT01 ~ LCNT15 寄存器设定 LCNT00 寄存器的值。写“1”后自动回复“0”。读取值为“0”。	R/W
b3	—	保留位	读写值都为“0”。	R/W
b4	LCINTC	IWDTCLK 分频时钟上升沿中断允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b5	LCINTD	IWDTCLK 计数值偏差超中断允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b6	LCINTO	LCNT 上溢中断允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b7	—	保留位	读写值都为“0”。	R/W
b10-b8	LCTO[2:0]	IWDTCLK 分频时钟上升沿中断减少次数设定位	b ¹⁰ b ⁸ 0 0 0: 不减少 0 0 1: 不能设定 0 1 0: 不能设定 0 1 1: 不能设定 1 0 0: 减少次数 7 (每 8 次计数 1 次) 1 0 1: 减少次数 15 (每 16 次计数 1 次) 1 1 0: 减少次数 127 (每 128 次计数 1 次) 1 1 1: 减少次数 255 (每 256 次计数 1 次)	R/W
b11	LCNTAT	IWDTCLK 计数结果减少设定位	0: 不减少 1: 减少	R/W
b13-b12	TPSC[1:0]	IWDTCLK 计数时钟选择位	b ¹³ b ¹² 0 0: PCLKA (定时器模块时钟) 0 1: PCLKA/2 (定时器模块时钟 /2) 1 0: PCLKA/4 (定时器模块时钟 /4) 1 1: PCLKA/8 (定时器模块时钟 /8)	R/W
b15-b14	LPSC[1:0]	IWDTCLK 分频时钟选择位	b ¹⁵ b ¹⁴ 0 0: 1 0 1: 1/16 1 0: 1/128 1 1: 1/256	R/W

LCCR 寄存器是设定 IWDT 专用低速时钟 (IWDTCLK) 的计数功能的寄存器。使用 IWDTCLK 的计数功能时, 必须让独立看门狗定时器 (IWDT) 也运行。

LCNTE 位 (IWDTCLK 计数功能允许位)

此位设定 IWDTCLK 的计数功能停止或者运行。

LCNTCR 位 (IWDTCLK 计数值清除位)

此位将 LCNT 计数器置“0”。

如果写“1”，就将计数值置“0”。写“1”后自动回复“0”。读取值为“0”。

LCNTS 位 (IWDTCLK 计数值设定位)

此位给 LCNT01 ~ LCNT15 寄存器设定 LCNT00 寄存器的值。

如果写“1”，就给 LCNT01 ~ LCNT15 寄存器设定 LCNT00 寄存器的值。写“1”后自动回复“0”。读取值为“0”。

LCINTC 位 (IWDTCLK 分频时钟上升沿中断允许位)

此位允许或者禁止由 IWDTCLK 分频时钟的上升沿产生的中断。中断请求作为 LOCOI 中断请求产生。

LCINTD 位 (IWDTCLK 计数值偏差超中断允许位)

此位允许或者禁止由 IWDTCLK 计数值的偏差超出产生的中断。中断请求作为 LOCOI 中断请求产生。

LCINTO 位 (LCNT 上溢中断允许位)

此位允许或者禁止由 LCNT 计数器的上溢产生的中断。中断请求作为 LOCOI 中断请求产生。

LCTO[2:0] 位 (IWDTCLK 分频时钟上升沿中断减少次数设定位)

这些位设定 IWDTCLK 分频时钟上升沿中断的减少次数。

LCNTAT 位 (IWDTCLK 计数结果减少设定位)

此位设定是否以 LCTO[2:0] 位设定的次数减少 IWDTCLK 计数结果传送到 LCNTn (n=00 ~ 15) 的时序。

TPSC[1:0] 位 (IWDTCLK 计数时钟选择位)

这些位选择对 IWDTCLK 分频时钟计数的时钟 (LCNT 运行时钟)。

LPSC[1:0] 位 (IWDTCLK 分频时钟选择位)

这些位选择 IWDTCLK 分频时钟的分频率。

如果要将 LPSC[1:0] 位置“00”以外的值，必须将 IWDT.IWDTCR.CKS[3:0] 位置“00xx”以外的值。

23.2.12 LOCO 计数状态寄存器 (LCST)

地址 000C 2082h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	—	—	—	LISO	LISD	LISC
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	LISC	IWDTCLK 分频时钟上升沿中断请求标志	0: 无中断请求 1: 有中断请求	R/W (注 1)
b1	LISD	IWDTCLK 计数值偏差超出中断请求标志	0: 无中断请求 1: 有中断请求	R/W (注 1)
b2	LISO	LCNT 上溢中断请求标志	0: 无中断请求 1: 有中断请求	R/W (注 1)
b15-b3	—	保留位	读写值都为“0”。	R/W

注 1. 只能写用于清除标志置的“0”。

LCST 寄存器是表示 IWDTCLK 分频时钟的计数状态的寄存器。

LISC 标志 (IWDTCLK 分频时钟上升沿中断请求标志)

这是由 IWDTCLK 分频时钟的上升沿产生中断请求的标志。在清除标志前不产生由 IWDTCLK 分频时钟的上升沿引起的下一个中断请求。

[为“1”的条件]

- 在 LCCR.LCINTC=1 的状态下 IWDTCLK 分频时钟上升时

[为“0”的条件]

- 给 LISC 标志写“0”时

LISD 标志 (IWDTCLK 计数值偏差超出中断请求标志)

这是由 IWDTCLK 计数值的偏差超出产生中断请求的标志。在标志置“0”前不产生由 IWDTCLK 计数值的偏差超出引起的下一个中断请求。

[为“1”的条件]

- 在 LCCR.LCINTD=1 的状态下 IWDTCLK 计数值器发生偏差超出时

[为“0”的条件]

- 给 LISD 标志写“0”时

LISO 标志 (LCNT 上溢中断请求标志)

这是由 LCNT 计数器的上溢产生中断请求的标志。在标志置“0”前不产生由 LCNT 计数器的上溢引起的下一个中断请求。

[为“1”的条件]

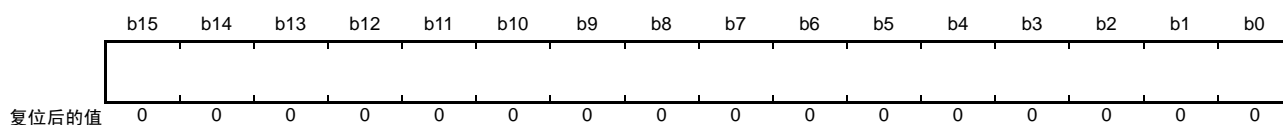
- 在 LCCR.LCINTO=1 的状态下 LCNT 计数器发生上溢时

[为“0”的条件]

- 给 LISO 标志写“0”时

23.2.13 LOCO 计数值寄存器 (LCNT)

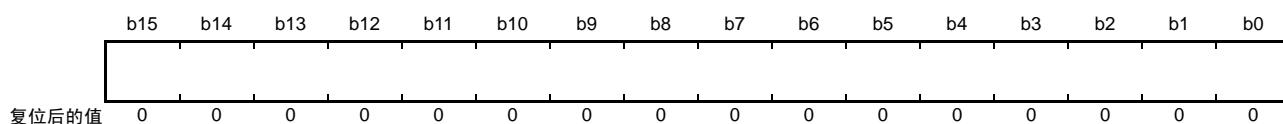
地址 000C 2084h



LCNT 寄存器是对 IWDTCLK 分频时钟进行计数的只读寄存器。

23.2.14 LOCO 计数结果平均寄存器 (LCNTA)

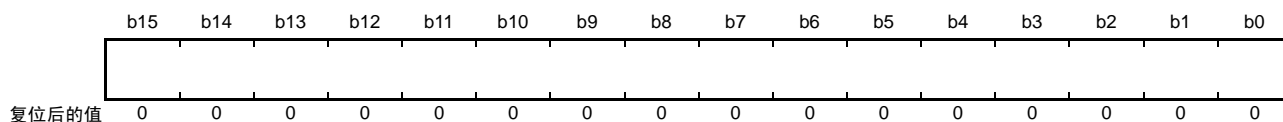
地址 000C 2086h



LCNTA 寄存器是表示 IWDTCLK 分频时钟计数结果 (LCNT00 ~ LCNT15) 的平均值的只读寄存器。

23.2.15 LOCO 计数结果寄存器 n (LCNTn) (n=00 ~ 15)

地址 LCNT00 000C 2088h、LCNT01 000C 208Ah、LCNT02 000C 208Ch、LCNT03 000C 208Eh、LCNT04 000C 2090h、LCNT05 000C 2092h、
LCNT06 000C 2094h、LCNT07 000C 2096h、LCNT08 000C 2098h、LCNT09 000C 209Ah、LCNT10 000C 209Ch、LCNT11 000C 209Eh、
LCNT12 000C 20A0h、LCNT13 000C 20A2h、LCNT14 000C 20A4h、LCNT15 000C 20A6h

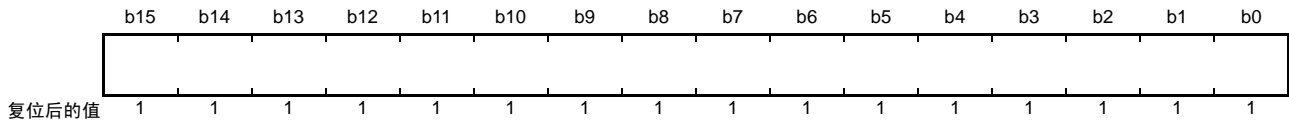


LCNTn 寄存器是表示 IWDTCLK 分频时钟计数结果的寄存器。只能在停止计数 (LCCR.LCNTTE 位 =0) 时进行写操作。

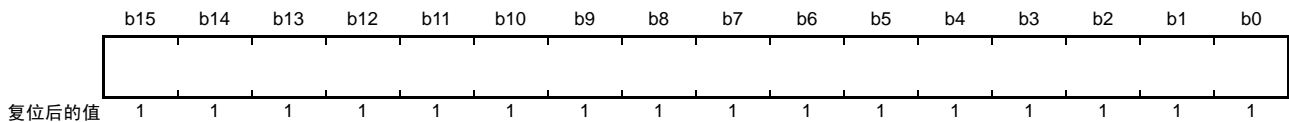
LCNT00 寄存器的计数值是最新结果。

23.2.16 LOCO 计数上限 / 下限容许偏差值寄存器 (LCNTDU、LCNTDL)

地址 LCNTDU 000C 20A8h



地址 LCNTDL 000C 20AAh

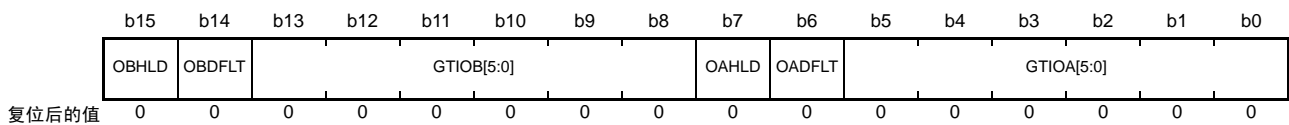


LCNTDU 寄存器和 LCNTDL 寄存器是设定 IWDTCCLK 分频时钟计数值的容许偏差的寄存器。

当 LCNT00 寄存器的值大于上限容许值 (LCNTA+LCNTDU) 或者低于下限容许值 (LCNTA-LCNTDL) 时, 产生 IWDTCCLK 计数值偏差超出中断请求。

23.2.17 通用 PWM 定时器的 I/O 控制寄存器 (GTIOR)

地址 GPT0.GTIOR 000C 2100h、GPT1.GTIOR 000C 2180h、GPT2.GTIOR 000C 2200h、GPT3.GTIOR 000C 2280h



位	符号	位名	功能	R/W
b5-b0	GTIOA[5:0]	GTIOCnA 引脚功能选择位	请参照表 23.4。	R/W
b6	OADFLT	GTIOCnA 引脚计数停止时的输出值位	0: 在停止计数时从 GTIOCnA 引脚输出 Low 电平 1: 在停止计数时从 GTIOCnA 引脚输出 High 电平	R/W
b7	OAHL	GTIOCnA 引脚计数开始 / 停止时的输出保持位	0: 计数开始 / 停止时的 GTIOCnA 引脚的输出电平取决于寄存器的设定值。 1: 保持计数开始 / 停止时的 GTIOCnA 引脚的输出电平	R/W
b13-b8	GTIOB[5:0]	GTIOCnB 引脚功能选择位	请参照表 23.4。	R/W
b14	OBDFLT	GTIOCnB 引脚计数停止时的输出值位	0: 在停止计数时从 GTIOCnB 引脚输出 Low 电平 1: 在停止计数时从 GTIOCnB 引脚输出 High 电平	R/W
b15	OBHLD	GTIOCnB 引脚计数开始 / 停止时的输出保持位	0: 计数开始 / 停止时的 GTIOCnB 引脚的输出电平取决于寄存器的设定值。 1: 保持计数开始 / 停止时的 GTIOCnB 引脚的输出电平	R/W

注 1. n=0 ~ 3

GPTn.GTIOR 寄存器是设定 GTIOCnA 引脚和 GTIOCnB 引脚 (n=0 ~ 3) 的功能的寄存器。各通道有 1 个 GPTn.GTIOR 寄存器。

GTIOA[5:0] 位 (GTIOCnA 引脚功能选择位)

此位选择 GTIOCnA 引脚的功能。详细内容请参照表 23.4。

OADFLT 位 (GTIOCnA 引脚计数停止时的输出值位)

在停止计数时，此位设定从 GTIOCnA 引脚输出 Low 电平还是 High 电平。

OAHL D 位 (GTIOCnA 引脚计数开始 / 停止时的输出保持位)

在计数开始 / 停止时，此位设定保持 GTIOCnA 引脚的输出电平还是取决于寄存器的设定值。

[将 OAHL D 位置 “0” 时]

- 在计数开始时，输出 GTIOR 寄存器的 bit4 指定的值
- 在计数停止时，输出 OADFLT 位指定的值
- 在计数停止时更改 OADFLT 位的值的情况下，立即反映到输出

[将 OAHL D 位置 “1” 时]

- 在计数开始 / 停止时保持输出

GTIOB[5:0] 位 (GTIOCnB 引脚功能选择位)

这些位选择 GTIOCnB 引脚的功能。详细内容请参照表 23.4。

OBDFLT 位 (GTIOCnB 引脚计数停止时的输出值位)

在停止计数时，此位设定从 GTIOCnB 引脚输出 Low 电平还是 High 电平。

OBHL D 位 (GTIOCnB 引脚计数开始 / 停止时的输出保持位)

在计数开始 / 停止时，此位设定保持 GTIOCnB 引脚的输出电平还是取决于寄存器的设定值。

[将 OBHL D 位置 “0” 时]

- 在计数开始时，输出 GTIOR 寄存器的 bit4 指定的值
- 在计数停止时，输出 OBDFLT 位指定的值
- 在计数停止时更改 OBDFLT 位的值的情况下，立即反映到输出

[将 OBHL D 位置 “1” 时]

- 在计数开始 / 停止时保持输出

表 23.4 GTIOA[5:0] 位 (GTIOB[5:0] 位) 的设定 (1/2)

GTIOA/B[5:0] 位						功能			
b5	b4	b3	b2	b1	b0	b5	b4	b3-b2	b1-b0
0	0	0	0	0	0	比较匹配	初始输出为 Low 电平	在周期结束时保持输出	通过 GPTn.GTCCRA/B 的比较匹配保持输出
0	0	0	0	0	1				通过 GPTn.GTCCRA/B 的比较匹配输出 Low 电平
0	0	0	0	1	0				通过 GPTn.GTCCRA/B 的比较匹配输出 High 电平
0	0	0	0	1	1				通过 GPTn.GTCCRA/B 的比较匹配进行交替输出
0	0	0	1	0	0			在周期结束时输出 Low 电平	通过 GPTn.GTCCRA/B 的比较匹配保持输出
0	0	0	1	0	1				通过 GPTn.GTCCRA/B 的比较匹配输出 Low 电平
0	0	0	1	1	0				通过 GPTn.GTCCRA/B 的比较匹配输出 High 电平
0	0	0	1	1	1				通过 GPTn.GTCCRA/B 的比较匹配进行交替输出
0	0	1	0	0	0		在周期结束时输出 High 电平	通过 GPTn.GTCCRA/B 的比较匹配保持输出	
0	0	1	0	0	1			通过 GPTn.GTCCRA/B 的比较匹配输出 Low 电平	
0	0	1	0	1	0			通过 GPTn.GTCCRA/B 的比较匹配输出 High 电平	
0	0	1	0	1	1			通过 GPTn.GTCCRA/B 的比较匹配进行交替输出	
0	0	1	1	0	0		在周期结束时交替输出	通过 GPTn.GTCCRA/B 的比较匹配保持输出	
0	0	1	1	0	1			通过 GPTn.GTCCRA/B 的比较匹配输出 Low 电平	
0	0	1	1	1	0			通过 GPTn.GTCCRA/B 的比较匹配输出 High 电平	
0	0	1	1	1	1			通过 GPTn.GTCCRA/B 的比较匹配进行交替输出	
0	1	0	0	0	0	初始输出为 High 电平	在周期结束时保持输出	通过 GPTn.GTCCRA/B 的比较匹配保持输出	
0	1	0	0	0	1			通过 GPTn.GTCCRA/B 的比较匹配输出 Low 电平	
0	1	0	0	1	0			通过 GPTn.GTCCRA/B 的比较匹配输出 High 电平	
0	1	0	0	1	1			通过 GPTn.GTCCRA/B 的比较匹配进行交替输出	

表 23.4 GTIOA[5:0] 位 (GTIOB[5:0] 位) 的设定 (2/2)

GTIOA/B[5:0]						功能			
b5	b4	b3	b2	b1	b0	b5	b4	b3-b2	b1-b0
0	1	0	1	0	0	比较匹配	初始输出为 High 电平	在周期结束时输出 Low 电平	通过 GPTn.GTCCRA/B 的比较匹配保持输出
0	1	0	1	0	1				通过 GPTn.GTCCRA/B 的比较匹配输出 Low 电平
0	1	0	1	1	0				通过 GPTn.GTCCRA/B 的比较匹配输出 High 电平
0	1	0	1	1	1				通过 GPTn.GTCCRA/B 的比较匹配进行交替输出
0	1	1	0	0	0				在周期结束时输出 High 电平
0	1	1	0	0	1			通过 GPTn.GTCCRA/B 的比较匹配输出 Low 电平	
0	1	1	0	1	0			通过 GPTn.GTCCRA/B 的比较匹配输出 High 电平	
0	1	1	0	1	1			通过 GPTn.GTCCRA/B 的比较匹配进行交替输出	
0	1	1	1	0	0			在周期结束时交替输出	通过 GPTn.GTCCRA/B 的比较匹配保持输出
0	1	1	1	0	1				通过 GPTn.GTCCRA/B 的比较匹配输出 Low 电平
0	1	1	1	1	0				通过 GPTn.GTCCRA/B 的比较匹配输出 High 电平
0	1	1	1	1	1				通过 GPTn.GTCCRA/B 的比较匹配进行交替输出
1	x	x	x	0	0			输入捕捉	don't care
1	x	x	x	0	1	在下降沿输入捕捉			
1	x	x	x	1	0	在双边沿输入捕捉			
1	x	x	x	1	1				

注. x: don't care

- 注 1. 周期结束在锯齿波时表示上溢 (递增计数时的 $GTCNT=GTTPR$) 或者下溢 (递减计数时的 $GTCNT=0$), 在三角波时表示波谷 ($GTCNT=0$)。
- 注 2. 在进行比较匹配运行时, 如果周期结束和 GTCCRA/B 寄存器的比较匹配时序相同, 就在锯齿波 PWM 模式中优先 b3-b2 的设定, 在其他模式中优先 b1-b0 的设定。
- 注 3. 只通过 GTIOR 寄存器设定为比较匹配时, 并不输出到引脚, 还必须另外进行 GTONCR 寄存器的设定。

23.2.18 通用 PWM 定时器的中断输出设定寄存器 (GTINTAD)

地址 GPT0.GTINTAD 000C 2102h、GPT1.GTINTAD 000C 2182h、GPT2.GTINTAD 000C 2202h、GPT3.GTINTAD 000C 2282h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADTRBD EN	ADTRBU EN	ADTRAD EN	ADTRAU EN	EINT	—	—	—	GTINTPR[1:0]	GTINTF	GTINTE	GTINTD	GTINTC	GTINTB	GTINTA	
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	GTINTA	GTCCRA 比较匹配 / 输入捕捉中断允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b1	GTINTB	GTCCRB 比较匹配 / 输入捕捉中断允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b2	GTINTC	GTCCRC 比较匹配中断允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b3	GTINTD	GTCCRD 比较匹配中断允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b4	GTINTE	GTCCRE 比较匹配中断允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b5	GTINTF	GTCCRF 比较匹配中断允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b7-b6	GTINTPR[1:0]	GTPR 比较匹配中断允许位	b7 b6 0 0: 禁止中断请求 0 1: 锯齿波时在上溢、三角波时在波峰 允许中断请求 1 0: 锯齿波时在下溢、三角波时在波谷 允许中断请求 1 1: 锯齿波时在上溢和下溢、三角波时 在波峰和波谷都允许中断请求	R/W
b10-b8	—	保留位	读写值都为“0”。	R/W
b11	EINT	死区时间错误中断允许位	0: 禁止中断请求 1: 允许中断请求	R/W
b12	ADTRAUEN	GTADTRA 比较匹配 (递增计数) A/D 转 换开始请求允许位	0: 禁止 A/D 转换开始请求 1: 允许 A/D 转换开始请求	R/W
b13	ADTRADEN	GTADTRA 比较匹配 (递减计数) A/D 转 换开始请求允许位	0: 禁止 A/D 转换开始请求 1: 允许 A/D 转换开始请求	R/W
b14	ADTRBUEN	GTADTRB 比较匹配 (递增计数) A/D 转 换开始请求允许位	0: 禁止 A/D 转换开始请求 1: 允许 A/D 转换开始请求	R/W
b15	ADTRBDEN	GTADTRB 比较匹配 (递减计数) A/D 转 换开始请求允许位	0: 禁止 A/D 转换开始请求 1: 允许 A/D 转换开始请求	R/W

GTINTAD 寄存器是设定允许或者禁止中断请求以及 A/D 转换开始请求的寄存器。

GTINTA 位 (GTCCRA 比较匹配 / 输入捕捉中断允许位)

此位允许或者禁止由 GTCCRA 寄存器的比较匹配 / 输入捕捉产生中断请求 (GTCIA)。

GTINTB 位 (GTCCRB 比较匹配 / 输入捕捉中断允许位)

此位允许或者禁止由 GTCCRB 寄存器的比较匹配 / 输入捕捉产生中断请求 (GTCIB)。

GTINTC 位 (GTCCRC 比较匹配中断允许位)

此位允许或者禁止由 GTCCRC 寄存器的比较匹配产生中断请求 (GTCIC)。

GTINTD 位 (GTCCRD 比较匹配中断允许位)

此位允许或者禁止由 GTCCRD 寄存器的比较匹配产生中断请求 (GTCID)。中断请求作为 GTCIC 中断产生。

GTINTE 位 (GTCCRE 比较匹配中断允许位)

此位允许或者禁止由 GTCCRE 寄存器的比较匹配产生中断请求 (GTCIE)。

GTINTF 位 (GTCCRF 比较匹配中断允许位)

此位允许或者禁止由 GTCCRF 寄存器的比较匹配产生中断请求 (GTCIF)。中断请求作为 GTCIE 中断产生。

GTINTPR[1:0] 位 (GTPR 比较匹配中断允许位)

这些位允许或者禁止由 GTPR 寄存器的比较匹配 (GTCNT 计数器的上溢) / GTCNT 计数器的下溢产生中断请求 (GTCIV)。

EINT 位 (死区时间错误中断允许位)

此位允许或者禁止由发生死区时间错误产生中断请求 (GTCIC)。中断请求作为 GTCIC 中断产生。

ADTRAUEN (GTADTRA 比较匹配 (递增计数) A/D 转换开始请求允许位)

此位允许或者禁止由和 GTCNT 计数器递增计数时的 GTADTRA 寄存器比较匹配产生 A/D 转换开始请求。

ADRADEN (GTADTRA 比较匹配 (递减计数) A/D 转换开始请求允许位)

此位允许或者禁止由和 GTCNT 计数器递减计数时的 GTADTRA 寄存器比较匹配产生 A/D 转换开始请求。

ADTRBUEN (GTADTRB 比较匹配 (递增计数) A/D 转换开始请求允许位)

此位允许或者禁止由和 GTCNT 计数器递增计数时的 GTADTRB 寄存器比较匹配产生 A/D 转换开始请求。

ADTRBDEN (GTADTRB 比较匹配 (递减计数) A/D 转换开始请求允许位)

此位允许或者禁止由和 GTCNT 计数器递减计数时的 GTADTRB 寄存器比较匹配产生 A/D 转换开始请求。

23.2.19 通用 PWM 定时器的控制寄存器 (GTCR)

地址 GPT0.GTCR 000C 2104h、GPT1.GTCR 000C 2184h、GPT2.GTCR 000C 2204h、GPT3.GTCR 000C 2284h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	CCLR[1:0]	—	—	TPCS[1:0]	—	—	—	—	—	—	—	MD[2:0]	—	—
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
b2-b0	MD[2:0]	模式选择位	b2 b0 0 0 0: 锯齿波 PWM 模式 (能单 / 双缓冲) 0 0 1: 锯齿波单触发脉冲模式 (固定为缓冲运行) 0 1 0: 不能设定 0 1 1: 不能设定 1 0 0: 三角波 PWM 模式 1 (在波谷以 16 位传送) (能单 / 双缓冲) 1 0 1: 三角波 PWM 模式 2 (在波峰 / 波谷以 16 位传送) (能单 / 双缓冲) 1 1 0: 三角波 PWM 模式 3 (在波谷以 32 位传送) (固定为缓冲运行) 1 1 1: 不能设定	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W
b9-b8	TPCS[1:0]	定时器的预分频器选择位	b9 b8 0 0: PCLKA (定时器模块时钟) 0 1: PCLKA/2 (定时器模块时钟 / 2) 1 0: PCLKA/4 (定时器模块时钟 / 4) 1 1: PCLKA/8 (定时器模块时钟 / 8)	R/W
b11-b10	—	保留位	读写值都为“0”。	R/W
b13-b12	CCLR[1:0]	计数器清除源选择位	b13 b12 0 0: 不能设定下述清除源 0 1: 通过 GTCCRA 寄存器的输入捕捉清除 1 0: 通过 GTCCRB 寄存器的输入捕捉清除 1 1: 通过正在进行同步清除 / 同步运行的其他计数器清除源清除	R/W
b15-b14	—	保留位	读写值都为“0”。	R/W

GTCR 寄存器是控制 GTCNT 计数器的寄存器。

必须在 GTCNT 计数器停止计数的状态下设定 GTCR 寄存器。

MD[2:0] 位 (模式选择位)

此位选择 GPT 的运行模式。

TPCS[1:0] 位 (定时器的预分频器选择位)

这些位选择 GTCNT 计数器的时钟。各通道能独立选择时钟源。

CCLR[1:0] 位 (计数器清除源选择位)

这些位选择 GTCNT 计数器的清除源。

如果在锯齿波的情况下选择同步清除, 此时的同步清除与通过自身的上溢 / 下溢进行的清除相同, 进行引脚输出 / 缓冲运行, 但是, 上溢标志 / 下溢标志不被置位; 如果在三角波的情况下选择同步清除, 此时的同步清除只进行计数器的清除, 计数器的值变为“0”, 但是, 不作为“波谷”处理。

通过计数器清除源选择“01b”、“10b”或者“11b”时, 不管 GPTn.GTCNT 计数器正在计数 (GTSTR.CSTn=1), 还是停止计数 (GTSTR.CSTn=0) (n=0 ~ 3), 都通过计数器清除源清除计数器。

23.2.20 通用 PWM 定时器的缓冲允许寄存器 (GTBER)

地址 GPT0.GTBER 000C 2106h、GPT1.GTBER 000C 2186h、GPT2.GTBER 000C 2206h、GPT3.GTBER 000C 2286h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	ADTDB	ADTTB[1:0]	—	ADTDA	ADTTA[1:0]	—	CCRS WT	—	PR[1:0]	CCRB[1:0]	CCRA[1:0]	—	—	—	—
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
b1-b0	CCRA[1:0]	GTCCRA 缓冲运行位	b1 b0 0 0: 不进行缓冲运行 0 1: 进行单缓冲运行 (GTCCRA↔GTCCRC) 1 x: 进行双缓冲运行 (GTCCRA↔GTCCRC↔GTCCRD)	R/W
b3-b2	CCRB[1:0]	GTCCRB 缓冲运行位	b3 b2 0 0: 不缓冲运行 0 1: 进行单缓冲运行 (GTCCRB↔GTCCRE) 1 x: 进行双缓冲运行 (GTCCRB↔GTCCRE↔GTCCRF)	R/W
b5-b4	PR[1:0]	GTPR 缓冲运行位	b5 b4 0 0: 不缓冲运行 0 1: 进行单缓冲运行 (GTPBR⇒GTPR) 1 x: 进行双缓冲运行 (GTPDBR⇒GTPBR⇒GTPR)	R/W
b6	CCRSWT	GTCCRA、GTCCRB 强制缓冲运行位	如果写“1”，就强制进行 GTCCRA、GTCCRB 的缓冲传送。写“1”后自动回复“0”。读取值为“0”。	R/W
b7	—	保留位	读写值都为“0”。	R/W
b9-b8	ADTTA[1:0]	GTADTRA 缓冲传送时序选择位	• 三角波的情况 b9 b8 0 0: 不传送 0 1: 在波峰进行传送 1 0: 在波谷进行传送 1 1: 在波峰 / 波谷进行传送 • 锯齿波的情况 b9 b8 0 0: 不传送 0 0 以外: 在上溢 (递增计数)、下溢 (递减计数) 时进行传送	R/W
b10	ADTDA	GTADTRA 双缓冲运行位	0: 进行单缓冲运行 (GTADTBRA⇒GTADTRA) 1: 进行双缓冲运行 (GTADTBRA⇒GTADTBRA⇒GTADTRA)	R/W
b11	—	保留位	读写值都为“0”。	R/W
b13-b12	ADTTB[1:0]	GTADTRB 缓冲传送时序选择位	• 三角波的情况 b13 b12 0 0: 不传送 0 1: 在波峰进行传送 1 0: 在波谷进行传送 1 1: 在波峰 / 波谷进行传送 • 锯齿波的情况 b13 b12 0 0: 不传送 0 0 以外: 在上溢 (递增计数)、下溢 (递减计数) 时进行传送	R/W

位	符号	位名	功能	R/W
b14	ADTDB	GTADTRB 双缓冲运行位	0: 进行单缓冲运行 (GTADTBRB→GTADTRB) 1: 进行双缓冲运行 (GTADTDBRB→GTADTBRB → GTADTRB)	R/W
b15	—	保留位	读写值都为 "0"。	R/W

GTBER 寄存器是设定缓冲运行的寄存器。

必须在 GTCNT 计数器停止计数的状态下设定 GTBER 寄存器。

CCRA[1:0] 位 (GTCCRA 缓冲运行位)

这些位设定 GTCCRA 寄存器、GTCCRC 寄存器和 GTCCRD 寄存器组合的缓冲运行。如果因为 GTCR 寄存器设定的运行模式使缓冲运行受到限制，就优先 GTCR 寄存器的设定 (注 1)。

CCRB[1:0] 位 (GTCCRB 缓冲运行位)

这些位设定 GTCCRB 寄存器、GTCCRE 寄存器和 GTCCRF 寄存器组合的缓冲运行。如果因为 GTCR 寄存器设定的运行模式使缓冲运行受到限制，就优先 GTCR 寄存器的设定 (注 1)。

PR[1:0] 位 (GTPR 缓冲运行位)

这些位设定 GTPR 寄存器、GTPBR 寄存器和 GTPDBR 寄存器组合的缓冲运行。

CCRSWT 位 (GTCCRA、GTCCRB 强制缓冲运行位)

如果给 CCRSWT 位写 "1"，就强制进行 GTCCRA 寄存器和 GTCCRB 寄存器的缓冲传送。写 "1" 后自动回复 "0"。读取值为 "0"。

只在停止计数并且设定为比较匹配运行时有效。

ADTTA[1:0] 位 (GTADTRA 缓冲传送时序选择位)

这些位设定 GTADTRA 寄存器、GTADTBRA 寄存器和 GTADTDBRA 寄存器的缓冲运行的传送时序。

ADTDA 位 (GTADTRA 双缓冲运行位)

此位设定 GTADTRA 寄存器、GTADTBRA 寄存器和 GTADTDBRA 寄存器组合的缓冲运行。

ADTTB[1:0] 位 (GTADTRB 缓冲传送时序选择位)

这些位设定 GTADTRB 寄存器、GTADTBRB 寄存器和 GTADTDBRB 寄存器的缓冲运行的传送时序。

ADTDB 位 (GTADTRB 双缓冲运行位)

此位设定 GTADTRB 寄存器、GTADTBRB 寄存器和 GTADTDBRB 寄存器组合的缓冲运行。

注 1. 在锯齿波单触发脉冲模式或者三角波 PWM 模式 3 (在波谷以 32 位传送) 的情况下，固定为缓冲运行。

23.2.21 通用 PWM 定时器的计数方向寄存器 (GTUDC)

地址 GPT0.GTUDC 000C 2108h、GPT1.GTUDC 000C 2188h、GPT2.GTUDC 000C 2208h、GPT3.GTUDC 000C 2288h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	UDF	UD
复位后的值	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	UD	计数方向设定位	0: GTCNT 计数器递减计数 1: GTCNT 计数器递增计数	R/W
b1	UDF	计数方向强制设定位	0: 不强制设定 1: 强制设定	R/W
b15-b2	—	保留位	读写值都为“0”。	R/W

GTUDC 寄存器是设定 GTCNT 计数器的计数方向 (递增 / 递减) 的寄存器。

- 锯齿波的情况
 - 如果在递增计数时将 UD 位置“0”，就在发生上溢 (GTCNT=GTPR) 时转换计数方向。
 - 如果在递减计数时将 UD 位置“1”，就在发生下溢 (GTCNT=0) 时转换计数方向。
 - 如果在停止计数时 UDF 位为“0”的状态下，将 UD 位从“1”更改为“0”，初始计数运行就为递增计数，并且在发生上溢 (GTCNT=GTPR) 时转换计数方向。
 - 如果在停止计数时 UDF 位为“0”的状态下，将 UD 位从“0”更改为“1”，初始计数运行就为递减计数，并且在发生下溢 (GTCNT=0) 时转换计数方向。
 - 如果在停止计数时将 UDF 位置“1”，此时的 UD 位的值就反映到开始计数后的计数方向。
- 三角波的情况
 - 即使在计数时更改 UD 位的值，也不反映到计数方向。
 - 即使在停止计数时 UDF 位为“0”的状态下更改 UD 位的值，也不反映到开始计数后的计数方向。
 - 如果在停止计数时将 UDF 位置“1”，此时的 UD 位的值反映到开始计数后的计数方向。

UD 位 (计数方向设定位)

此位设定 GTCNT 计数器的计数方向 (递增 / 递减)。

UDF 位 (计数方向强制设定位)

此位将 GTCNT 计数器开始运行时的计数方向强制设定为 UD 位的值。

计数时只能给此位写“0”。

如果在停止计数时给此位写“1”，就必须在开始计数前将此位回复到“0”。

23.2.22 通用 PWM 定时器的中断、A/D 转换开始请求减少设定寄存器 (GTITC)

地址 GPT0.GTITC 000C 210Ah、GPT1.GTITC 000C 218Ah、GPT2.GTITC 000C 220Ah、GPT3.GTITC 000C 228Ah

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	ADTBL	—	ADTAL	—	IVTT[2:0]		IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA	
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	ITLA	GTCCRA 比较匹配 / 输入捕捉中断联动位	0: 不联动 GTCIV 中断减少功能 1: 联动 GTCIV 中断减少功能	R/W
b1	ITLB	GTCCRB 比较匹配 / 输入捕捉中断联动位	0: 不联动 GTCIV 中断减少功能 1: 联动 GTCIV 中断减少功能	R/W
b2	ITLC	GTCCRC 比较匹配中断联动位	0: 不联动 GTCIV 中断减少功能 1: 联动 GTCIV 中断减少功能	R/W
b3	ITLD	GTCCRD 比较匹配中断联动位	0: 不联动 GTCIV 中断减少功能 1: 联动 GTCIV 中断减少功能	R/W
b4	ITLE	GTCCRE 比较匹配中断联动位	0: 不联动 GTCIV 中断减少功能 1: 联动 GTCIV 中断减少功能	R/W
b5	ITLF	GTCCRF 比较匹配中断联动位	0: 不联动 GTCIV 中断减少功能 1: 联动 GTCIV 中断减少功能	R/W
b7-b6	IVTC[1:0]	GTCIV 中断减少功能选择位	b7 b6 0 0: 不减少 0 1: 对锯齿波时的上溢、三角波时的波峰进行计数并且减少功能有效 1 0: 对锯齿波时的下溢、三角波时的波谷进行计数并且减少功能有效 1 1: 对锯齿波时的上溢 / 下溢、三角波波峰 / 波谷都进行计数并且减少功能有效	R/W
b10-b8	IVTT[2:0]	GTCIV 中断减少次数选择位	b10 b8 0 0 0: 不减少 0 0 1: 减少次数为 1 次 0 1 0: 减少次数为 2 次 0 1 1: 减少次数为 3 次 1 0 0: 减少次数为 4 次 1 0 1: 减少次数为 5 次 1 1 0: 减少次数为 6 次 1 1 1: 减少次数为 7 次	R/W
b11	—	保留位	读取值都为“0”。	R/W
b12	ADTAL	GTADTRA 的 A/D 转换开始请求联动位	0: 不联动 GTCIV 中断减少功能 1: 联动 GTCIV 中断减少功能	R/W
b13	—	保留位	读取值都为“0”。	R/W
b14	ADTBL	GTADTRB 的 A/D 转换开始请求联动位	0: 不联动 GTCIV 中断减少功能 1: 联动 GTCIV 中断减少功能	R/W
b15	—	保留位	读取值都为“0”。	R/W

GTITC 寄存器是设定 GTCNT 计数器的上溢 (GTPR 寄存器的比较匹配) / 下溢中断 (GTCTV) 减少功能、以及设定其他中断以及 A/D 转换开始请求是否联动 GTCIV 中断减少功能的寄存器。死区时间错误中断不能联动 GTCIV 中断减少功能。当设定为中断减少功能时, 状态标志的变化也减少。

ITLA 位 (GTCCRA 比较匹配 / 输入捕捉中断联动位)

此位设定是否将 GTCCRA 寄存器的比较匹配 / 输入捕捉中断 (GTCIA) 与 GTCIV 中断减少功能联动。

ITLB 位 (GTCCRB 比较匹配 / 输入捕捉中断联动位)

此位设定是否将 GTCCRB 寄存器的比较匹配 / 输入捕捉中断 (GTCIB) 与 GTCIV 中断减少功能联动。

ITLC 位 (GTCCRC 比较匹配中断联动位)

此位设定是否将 GTCCRC 寄存器的比较匹配中断 (GTCIC) 与 GTCIV 中断减少功能联动。

ITLD 位 (GTCCRD 比较匹配中断联动位)

此位设定是否将 GTCCRD 寄存器的比较匹配中断 (GTCID) 与 GTCIV 中断减少功能联动。

ITLE 位 (GTCCRE 比较匹配中断联动位)

此位设定是否将 GTCCRE 寄存器的比较匹配中断 (GTCIE) 与 GTCIV 中断减少功能联动。

ITLF 位 (GTCCRF 比较匹配中断联动位)

此位设定是否将 GTCCRF 寄存器的比较匹配中断 (GTCIF) 与 GTCIV 中断减少功能联动。

IVTC[1:0] 位 (GTCIV 中断减少功能选择位)

此位选择 GTPR 寄存器的比较匹配 (GTCNT 上溢) /GTCNT 下溢中断 (GTCIV) 的减少功能。

IVTT[2:0] 位 (GTCIV 中断减少次数选择位)

此位选择 GTPT 寄存器的比较匹配 (GTCNT 上溢) /GTCNT 下溢中断 (GTCIV) 的减少次数。
必须在将 IVTC[1:0] 位置“00b”后才能更改 IVTT[2:0] 位。

ADTAL 位 (GTADTRA 的 A/D 转换开始请求联动位)

此位设定是否将通过 GTADTRA 寄存器的比较匹配产生的 A/D 转换请求与 GTCIV_n 中断减少功能联动。

ADTBL 位 (GTADTRB 的 A/D 转换开始请求联动位)

此位设定是否将通过 GTADTRB 寄存器的比较匹配产生的 A/D 转换请求与 GTCIV_n 中断减少功能联动。

23.2.23 通用 PWM 定时器的状态寄存器 (GTST)

地址 GPT0.GTST 000C 210Ch、GPT1.GTST 000C 218Ch、GPT2.GTST 000C 220Ch、GPT3.GTST 000C 228Ch

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TUCF	—	—	—	DTEF	ITCNT[2:0]		TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA	
复位后的值	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	TCFA	输入捕捉 / 比较匹配 A	0: 未发生 GTCCRA 寄存器的输入捕捉 / 比较匹配 1: 发生 GTCCRA 寄存器的输入捕捉 / 比较匹配	R/W (注 1)
b1	TCFB	输入捕捉 / 比较匹配 B	0: 未发生 GTCCRB 寄存器的输入捕捉 / 比较匹配 1: 发生 GTCCRB 寄存器的输入捕捉 / 比较匹配	R/W (注 1)
b2	TCFC	比较匹配标志 C	0: 未发生 GTCCRC 寄存器的比较匹配 1: 发生 GTCCRC 寄存器的比较匹配	R/W (注 1)
b3	TCFD	比较匹配标志 D	0: 未发生 GTCCRD 寄存器的比较匹配 1: 发生 GTCCRD 寄存器的比较匹配	R/W (注 1)
b4	TCFE	比较匹配标志 E	0: 未发生 GTCCRE 寄存器的比较匹配 1: 发生 GTCCRE 寄存器的比较匹配	R/W (注 1)
b5	TCFF	比较匹配标志 F	0: 未发生 GTCCRF 寄存器的比较匹配 1: 发生 GTCCRF 寄存器的比较匹配	R/W (注 1)
b6	TCFPO	上溢标志	0: 未发生上溢或者波峰 1: 发生上溢或者波峰	R/W (注 1)
b7	TCFPU	下溢标志	0: 未发生下溢或者波谷 1: 发生下溢或者波谷	R/W (注 1)
b10-b8	ITCNT[2:0]	GTCIV 中断减少次数计数器	定时器的中断减少次数计数器	R
b11	DTEF	死区时间错误标志	0: 未发生死区时间错误 1: 发生死区时间错误	R
b14-b12	—	保留位	读写值都为“0”。	R/W
b15	TUCF	计数方向标志	0: GPTn.GTCNT 计数器递减计数 1: GPTn.GTCNT 计数器递增计数	R

注 1. 只能写用于清除标志的“0”。

GTST 寄存器是表示 GPT 状态寄存器。

TCFA 标志 (输入捕捉 / 比较匹配标志 A)

这是表示发生 GTCCRA 寄存器的输入捕捉 / 比较匹配的状态标志。

[为“1”的条件]

- 在 GTCCRA 寄存器用作比较匹配寄存器的情况下，GTCNT=GTCCRA 时
- 在 GTCCRA 寄存器用作输入捕捉寄存器的情况下，通过输入捕捉信号将 GTCNT 计数器的值传送到 GTCCRA 寄存器时

[为“0”的条件]

- 给 TCFA 标志写“0”时

TCFB 标志 (输入捕捉 / 比较匹配标志 B)

这是表示发生 GTCCRB 寄存器的输入捕捉 / 比较匹配的状态标志。

[为“1”的条件]

- 在 GTCCRB 寄存器用作比较匹配寄存器的情况下, GTCNT=GTCCRB 时
- 在 GTCCRB 寄存器用作输入捕捉寄存器的情况下, 通过输入捕捉信号将 GTCNT 计数器的值传送到 GTCCRB 寄存器时

[为“0”的条件]

- 给 TCFB 标志写“0”时

TCFC 标志 (比较匹配标志 C)

这是表示发生 GTCCRC 寄存器的比较匹配的状态标志。

当 GTCCRC 寄存器缓冲运行时, 不进行比较匹配。

[为“1”的条件]

- GTCNT=GTCCRC 时

[为“0”的条件]

- 给 TCFC 标志写“0”时

[不进行比较匹配的条件]

- GTCR.MD[2:0]=001b (锯齿波单触发脉冲模式)
- GTCR.MD[2:0]=110b (三角波 PWM 模式 3)
- GTBER.CCRA[1:0]=01b、10b、11b (GTCCRC 寄存器缓冲运行)

TCFD 标志 (比较匹配标志 D)

这是表示发生 GTCCRD 寄存器的比较匹配的状态标志。

当 GTCCRD 寄存器缓冲运行时, 不进行比较匹配。

[为“1”的条件]

- GTCNT=GTCCRD 时

[为“0”的条件]

- 给 TCFD 标志写“0”时

[不进行比较匹配的条件]

- GTCR.MD[2:0]=001b (锯齿波单触发脉冲模式)
- GTCR.MD[2:0]=110b (三角波 PWM 模式 3)
- GTBER.CCRA[1:0]=10b、11b (GTCCRD 寄存器缓冲运行)

TCFE 标志 (比较匹配标志 E)

这是表示发生 GTCCRE 寄存器的比较匹配的状态标志。
当 GTCCRE 寄存器缓冲运行时, 不进行比较匹配。

[为“1”的条件]

- GTCNT=GTCCRE时

[为“0”的条件]

- 给TCFE标志写“0”时

[不进行比较匹配的条件]

- GTCR.MD[2:0]=001b (锯齿波单触发脉冲模式)
- GTCR.MD[2:0]=110b (三角波PWM模式3)
- GTBER.CCRB[1:0]=01b、10b、11b (GTCCRE寄存器缓冲运行)

TCFF 标志 (比较匹配标志 F)

这是表示发生 GTCCRF 寄存器的比较匹配的状态标志。
当 GTCCRF 寄存器缓冲运行时, 不进行比较匹配。

[为“1”的条件]

- GTCNT=GTCCRF时

[为“0”的条件]

- 给TCFF标志写“0”时

[不进行比较匹配的条件]

- GTCR.MD[2:0]=001b (锯齿波单触发脉冲模式)
- GTCR.MD[2:0]=110b (三角波PWM模式3)
- GTBER.CCRB[1:0]=10b、11b (GTCCRF寄存器缓冲运行)

TCFPO 标志 (上溢标志)

这是表示发生上溢或者波峰的标志。

[为“1”的条件]

- 在锯齿波时发生上溢 (递增计数时GTCNT=GTPR)
- 在三角波时产生波峰 (GTCNT=GTPR)

[为“0”的条件]

- 给TCFPO标志写“0”时

TCFPU 标志 (下溢标志)

这是表示发生下溢或者波谷的标志。

[为“1”的条件]

- 在锯齿波时发生下溢 (递减计数时GTCNT=0)
- 在三角波时产生波谷 (GTCNT=0)

[为“0”的条件]

- 给TCFPU标志写“0”时

ITCNT[2:0] 位 (GTCIV 中断减少次数计数器)

在使用 GTCIV 中断减少功能 (将 GTITC.IVTC[1:0] 位设定为不为“00b”) 时, 每产生 1 次 GTCIV 中断源就递增 1。

[为“0”的条件]

- 未使用 GTCIV 中断减少功能时 (GTITC.IVTC[1:0] 位为“00b”、GTITC.IVTT[2:0] 位为“000b”时)
- GTCIV 中断减少次数相同时 (通过 GTITC.IVTT[2:0] 位设定的减少次数与 ITCNT[2:0] 位的值相同时)

DTEF 标志 (死区时间错误标志)

这是表示自动附加死区时间后的定时器输出交替点超出定时器周期的标志。

如果自动附加死区时间后的定时器输出交替点回复到定时器周期内, 此位就变为“0”。DTEF 标志是只读标志 (即使写“0”也不变为“0”)。

如果允许由 DTEF 标志引起的中断 (GTINTAD.EINT=1), 每当 DTEF 标志从“0”变为“1”时, 就发生 GTCIC 中断。

[为“1”的条件]

- 自动附加死区时间后的定时器输出交替点超出定时器周期时

[为“0”的条件]

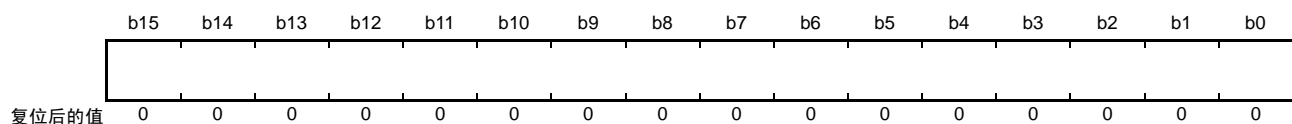
- 自动附加死区时间后的定时器输出交替点在定时器周期内时

TUCF 标志 (计数方向标志)

这是表示 GTCNT 计数器的计数方向的标志。

23.2.24 通用 PWM 定时器的计数器 (GTCNT)

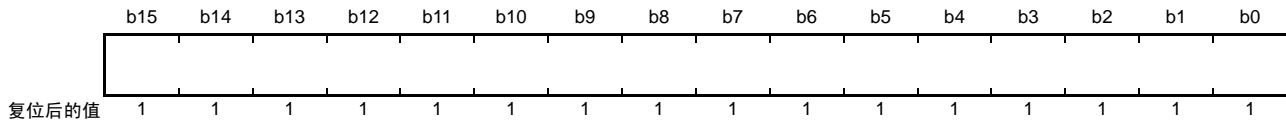
地址 GPT0.GTCNT 000C 210Eh、GPT1.GTCNT 000C 218Eh、GPT2.GTCNT 000C 220Eh、GPT3.GTCNT 000C 228Eh



GTCNT 计数器是 16 位可读写计数器。各通道有 1 个 (共计 4 个) GTCNT 计数器。不能在计数时进行写操作, 只能在停止计数时进行写操作。禁止以 8 位为单位存取 GTCNT 计数器, 必须以 16 位为单位进行存取。

23.2.25 通用 PWM 定时器的比较捕捉寄存器 m (GTCCRm) (m=A ~ F)

地址 GPT0.GTCCRA 000C 2110h、GPT1.GTCCRA 000C 2190h、GPT2.GTCCRA 000C 2210h、GPT3.GTCCRA 000C 2290h
 GPT0.GTCCRB 000C 2112h、GPT1.GTCCRB 000C 2192h、GPT2.GTCCRB 000C 2212h、GPT3.GTCCRB 000C 2292h
 GPT0.GTCCRC 000C 2114h、GPT1.GTCCRC 000C 2194h、GPT2.GTCCRC 000C 2214h、GPT3.GTCCRC 000C 2294h
 GPT0.GTCCRD 000C 2116h、GPT1.GTCCRD 000C 2196h、GPT2.GTCCRD 000C 2216h、GPT3.GTCCRD 000C 2296h
 GPT0.GTCCRE 000C 2118h、GPT1.GTCCRE 000C 2198h、GPT2.GTCCRE 000C 2218h、GPT3.GTCCRE 000C 2298h
 GPT0.GTCCRF 000C 211Ah、GPT1.GTCCRF 000C 219Ah、GPT2.GTCCRF 000C 221Ah、GPT3.GTCCRF 000C 229Ah



GTCCRm 寄存器是 16 位可读写寄存器。各通道有 6 个（共计 24 个）GTCCRm 寄存器。

GTCCRA 寄存器和 GTCCRB 寄存器是输出比较 / 输入捕捉兼用的寄存器。GTCCRC 寄存器和 GTCCRE 寄存器是比较匹配寄存器，但是也能用作 GTCCRA 寄存器和 GTCCRB 寄存器的缓冲寄存器。GTCCRD 寄存器和 GTCCRF 寄存器是比较匹配寄存器，但是也能用作 GTCCRC 寄存器和 GTCCRE 寄存器的缓冲寄存器（GTCCRA 寄存器和 GTCCRB 寄存器的双缓冲寄存器）。

23.2.26 通用 PWM 定时器的周期设定寄存器 (GTPR)

地址 GPT0.GTPR 000C 211Ch、GPT1.GTPR 000C 219Ch、GPT2.GTPR 000C 221Ch、GPT3.GTPR 000C 229Ch



GTPR 寄存器是 16 位可读写寄存器。用于设定 GTCNT 计数器的最大计数值。各通道有 1 个（共计 4 个）GTPR 寄存器。

在锯齿波时，计数周期为 GTPR 值 $\times 1$ ，在三角波时，计数周期为 GTPR 值 $\times 2$ 。

23.2.27 通用 PWM 定时器的周期设定缓冲寄存器 (GTPBR)

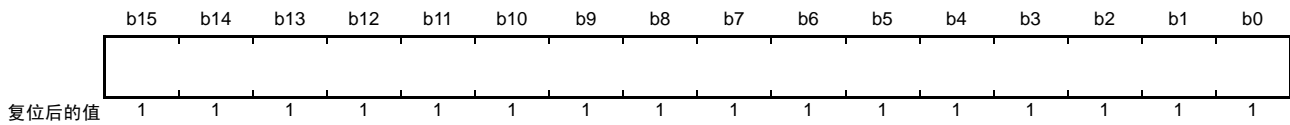
地址 GPT0.GTPBR 000C 211Eh、GPT1.GTPBR 000C 219Eh、GPT2.GTPBR 000C 221Eh、GPT3.GTPBR 000C 229Eh



GTPBR 寄存器是 16 位可读写寄存器。用作 GTPR 寄存器的缓冲寄存器。各通道有 1 个（共计 4 个）GTPBR 寄存器。

23.2.28 通用 PWM 定时器的周期设定双缓冲寄存器 (GTPDBR)

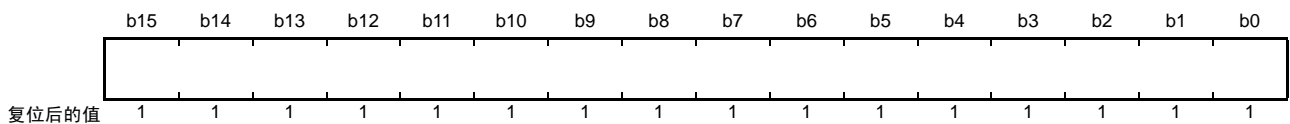
地址 GPT0.GTPDBR 000C 2120h、GPT1.GTPDBR 000C 21A0h、GPT2.GTPDBR 000C 2220h、GPT3.GTPDBR 000C 22A0h



GTPDBR 寄存器是 16 位可读写寄存器。用作 GTPBR 寄存器的缓冲寄存器 (GTPR 寄存器的双缓冲寄存器)。各通道有 1 个 (共计 4 个) GTPDBR 寄存器。

23.2.29 A/D 转换开始请求时序寄存器 m (GTADTRm) (m=A、B)

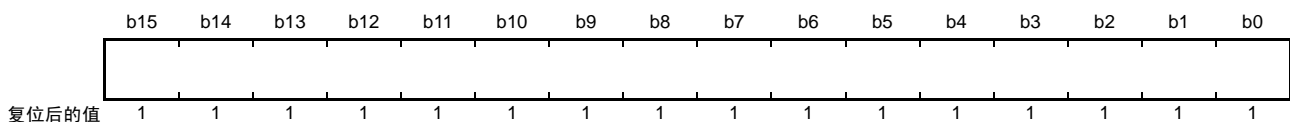
地址 GPT0.GTADTRA 000C 2124h、GPT1.GTADTRA 000C 21A4h、GPT2.GTADTRA 000C 2224h、GPT3.GTADTRA 000C 22A4h
GPT0.GTADTRB 000C 212Ch、GPT1.GTADTRB 000C 21ACh、GPT2.GTADTRB 000C 222Ch、GPT3.GTADTRB 000C 22ACh



GTADTRm 寄存器是 16 位可读写寄存器。用于设定 A/D 转换开始请求的时序。当 GTADTRm 寄存器的值与 GTCNT 计数器的值相同时, 产生 A/D 转换开始请求。各通道有 2 个 (共计 8 个) GTADTRm 寄存器。禁止以 8 位为单位存取 GTADTRm 寄存器, 必须以 16 位为单位进行存取。

23.2.30 A/D 转换开始请求时序缓冲寄存器 m (GTADTBRm) (m=A、B)

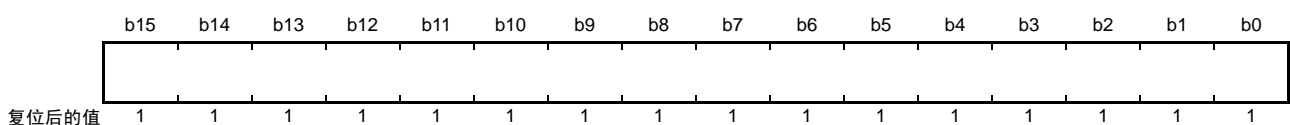
地址 GPT0.GTADTBRA 000C 2126h、GPT1.GTADTBRA 000C 21A6h、GPT2.GTADTBRA 000C 2226h、GPT3.GTADTBRA 000C 22A6h
GPT0.GTADTBRB 000C 212Eh、GPT1.GTADTBRB 000C 21AEh、GPT2.GTADTBRB 000C 222Eh、GPT3.GTADTBRB 000C 22AEh



GTADTBRm 寄存器是 16 位可读写寄存器。用作 GTADTRm 寄存器的缓冲寄存器。各通道有 2 个 (共计 8 个) GTADTBRm 寄存器。禁止以 8 位为单位存取 GTADTBRm 寄存器, 必须以 16 位为单位进行存取。

23.2.31 A/D 转换开始请求时序双缓冲寄存器 m (GTADTDBRm) (m=A、B)

地址 GPT0.GTADTDBRA 000C 2128h、GPT1.GTADTDBRA 000C 21A8h、GPT2.GTADTDBRA 000C 2228h、GPT3.GTADTDBRA 000C 22A8h
GPT0.GTADTDBRB 000C 2130h、GPT1.GTADTDBRB 000C 21B0h、GPT2.GTADTDBRB 000C 2230h、GPT3.GTADTDBRB 000C 22B0h



GTADTDBRm 寄存器是 16 位可读写寄存器。用作 GTADTBR 寄存器的缓冲寄存器 (GTADTR 的双缓冲寄存器)。各通道有 2 个 (共计 8 个) GTADTDBRm 寄存器。不能以 8 位为单位存取 GTADTDBRm 寄存器, 必须以 16 位为单位进行存取。

23.2.32 通用 PWM 定时器的输出无效控制寄存器 (GTONCR)

地址 GPT0.GTONCR 000C 2134h、GPT1.GTONCR 000C 21B4h、GPT2.GTONCR 000C 2234h、GPT3.GTONCR 000C 22B4h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OBE	OAE	—	SWN	—	—	—	NFV		NFS[3:0]			NVB	NVA	NEB	NEA
复位后的值	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	NEA	GTIOCnA 引脚无效控制允许位	0: 不允许 1: 允许	R/W
b1	NEB	GTIOCnB 引脚无效控制允许位	0: 不允许 1: 允许	R/W
b2	NVA	GTIOCnA 引脚无效值设定位	0: 在无效控制时将 GTIOCnA 引脚置“0” 1: 在无效控制时将 GTIOCnA 引脚置“1”	R/W
b3	NVB	GTIOCnB 引脚无效值设定位	0: 在无效控制时将 GTIOCnB 引脚置“0” 1: 在无效控制时将 GTIOCnB 引脚置“1”	R/W
b7-b4	NFS[3:0]	GTIOC 输出无效源选择位	b7 b4 0 0 0 0: AN000 的比较器检测 0 0 0 1: AN001 的比较器检测 0 0 1 0: AN002 的比较器检测 0 0 1 1: 不能设定 0 1 0 0: 不能设定 0 1 0 1: 不能设定 0 1 1 0: 不能设定 0 1 1 1: GTETRГ 引脚输入 1 x x x: 软件控制 (通过 SWN 位控制)	R/W
b8	NFV	无效源极性选择位	0: 在无效源变为“0”时进行无效控制 1: 在无效源变为“1”时进行无效控制	R/W
b11-b9	—	保留位	读写值都为“0”。	R/W
b12	SWN	软件无效控制位	• 当 NFV 位为“0”时 0: 进行无效控制 1: 不进行无效控制 • 当 NFV 位为“1”时 0: 不进行无效控制 1: 进行无效控制	R/W
b13	—	保留位	读写值都为“0”。	R/W
b14	OAE	GTIOCnA 引脚输出允许位	0: 不进行引脚输出 1: 进行引脚输出	R/W
b15	OBE	GTIOCnB 引脚输出允许位	0: 不进行引脚输出 1: 进行引脚输出	R/W

注 1. n=0 ~ 3

GTONCR 寄存器是设定 GTIOCnA 引脚输出和 GTIOCnB 引脚输出的无效控制的寄存器。

NEA 位 (GTIOcNA 引脚无效控制位)

此位允许 GTIOcNA 引脚输出的无效控制。

NEB 位 (GTIOcNB 引脚无效控制位)

此位允许 GTIOcNB 引脚输出的无效控制。

NVA 位 (GTIOcNA 引脚无效值设定位)

此位设定 GTIOcNA 引脚输出无效控制时的输出值。

NVB 位 (GTIOcNB 引脚无效控制位)

此位允许 GTIOcNB 引脚输出无效控制时的输出值。

NFS[3:0] 位 (GTIOC 输出无效源选择位)

这些位选择 GTIOcNA 引脚输出和 GTIOcNB 引脚输出的无效源。

NFV 位 (无效源极性选择位)

此位选择 GTIOcNA 引脚输出和 GTIOcNB 引脚输出的无效源极性。

SWN 位 (软件无效控制位)

此位设定是否进行 GTIOcNA 引脚输出和 GTIOcNB 引脚输出的无效控制。
当选择软件控制作为无效源 (NFS[3] 位为“1”) 时, 此位有效。

OAE 位 (GTIOcNA 引脚输出允许位)

此位选择是否进行 GTIOcNA 引脚输出。只在设定为比较匹配 (GTIOR 寄存器的 bit5=0) 时 OAE 位的设定有效。

OBE 位 (GTIOcNB 引脚输出允许位)

此位选择是否进行 GTIOcNB 引脚输出。只在设定为比较匹配 (GTIOR 寄存器的 bit13=0) 时 OBE 位的设定有效。

23.2.33 通用 PWM 定时器的死区时间控制寄存器 (GTDTCCR)

地址 GPT0.GTDTCCR 000C 2136h、GPT1.GTDTCCR 000C 21B6h、GPT2.GTDTCCR 000C 2236h、GPT3.GTDTCCR 000C 22B6h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	TDFER	—	—	TDBDE	TDBUE	—	—	—	TDE
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	TDE	反相波形设定位	0: 不使用 GTDVU 寄存器、GTDVD 寄存器, 而个别设定 GTCCRB 寄存器 1: 使用 GTDVU 寄存器、GTDVD 寄存器, 将带死区时间的反相波形的比较匹配值自动设定到 GTCCRB 寄存器。	R/W
b3-b1	—	保留位	读写值都为“0”。	R/W
b4	TDBUE	GTDVU 缓冲运行允许位	0: 禁止 GTDVU 寄存器的缓冲运行 1: 允许 GTDVU 寄存器的缓冲运行	R/W
b5	TDBDE	GTDVD 缓冲运行运行位	0: 禁止 GTDVD 寄存器的缓冲运行 1: 允许 GTDVD 寄存器的缓冲运行	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W
b8	TDFER	GTDVD 设定位	0: 个别设定 GTDVU 寄存器、GTDVD 寄存器 1: 将 GTDVU 寄存器的写入值自动设定到 GTDVD 寄存器	R/W
b15-b9	—	保留位	读写值都为“0”。	R/W

GTDTCCR 寄存器是设定允许自动设定带死区时间的反相波形的比较匹配值的寄存器。

TDE 位 (反相波形设定位)

此位设定是否使用 GTDVU 寄存器和 GTDVD 寄存器。当使用 GTDVU 寄存器和 GTDVD 寄存器时, 将由正相波形的比较匹配值 (GTCCRRA 寄存器) 和死区时间值 (GTDVU 寄存器和 GTDVD 寄存器) 算出的带死区时间的反相波形的比较匹配值自动设定到 GTCCRB 寄存器。

在锯齿波 PWM 模式中忽视 TDE 位的设定, 也不自动设定此位。

自动设定的 GTCCRB 寄存器的上限值 / 下限值如下所示。当算出的 GTCCRB 寄存器值在上限值 / 下限值的范围外时, 就给 GTCCRB 寄存器设定上限值 / 下限值, 并且 GPTn.GTST.DTEF 标志变为“1”。

- 三角波的情况
上限值: $GTPR - 1$
下限值: 递增计数时为“1”, 递减计数时为“0”。
- 锯齿波单触发脉冲模式的情况
上限值: GTPR
下限值: 0

TDBUE 位 (GTDVU 缓冲运行允许位)

此位允许 GTDVU 寄存器和 GTDBU 寄存器组合的缓冲运行。
缓冲传送时序在三角波时为波谷, 在锯齿波时为上溢或者下溢。

TDBDE 位 (GTDVD 缓冲运行允许位)

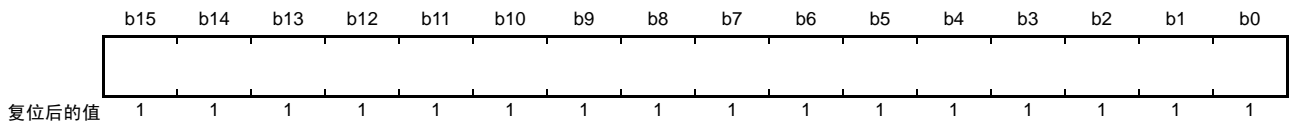
此位允许 GTDVD 寄存器和 GTDBD 寄存器组合的缓冲运行。
缓冲传送时序在三角波时为波谷，在锯齿波时为上溢或者下溢。
同时将 TDFER 位置“1”时，优先 TDFER 位的设定。

TDFER 位 (GTDVD 设定位)

此位设定是否将 GTDVU 寄存器的写入值自动设定到 GTDVD 寄存器。

23.2.34 通用 PWM 定时器的死区时间值寄存器 m (GTDVm) (m=U、D)

地址 GPT0.GTDVU 000C 2138h、GPT1.GTDVU 000C 21B8h、GPT2.GTDVU 000C 2238h、GPT3.GTDVU 000C 22B8h
GPT0.GTDVD 000C 213Ah、GPT1.GTDVD 000C 21BAh、GPT2.GTDVD 000C 223Ah、GPT3.GTDVD 000C 22BAh



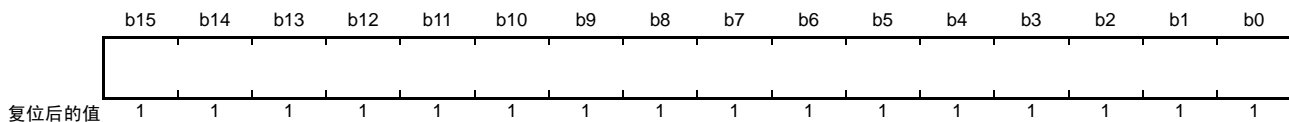
GTDVm 寄存器是 16 位可读写寄存器。用于设定生成带死区时间的 PWM 波形的死区时间。各通道分别有 2 个用于递增计数的 GTDVU 寄存器和用于递减计数的 GTDVD 寄存器，共计 8 个 GTDVm 寄存器。

禁止设定超出周期的死区时间值。能通过读 GTCCRB 寄存器读设定值。在使用 GTDVm 寄存器时禁止写 GTCCRB 寄存器。如果将死区时间值设定为“0”，就输出无死区时间的波形。

禁止以 8 位为单位存取 GTDVm 寄存器，必须以 16 位为单位进行存取。

23.2.35 通用 PWM 定时器的死区时间缓冲寄存器 m (GTDBm) (m=U、D)

地址 GPT0.GTDBU 000C 213Ch、GPT1.GTDBU 000C 21BCh、GPT2.GTDBU 000C 223Ch、GPT3.GTDBU 000C 22BCh
GPT0.GTDBD 000C 213Eh、GPT1.GTDBD 000C 21BEh、GPT2.GTDBD 000C 223Eh、GPT3.GTDBD 000C 22BEh



GTDBm 寄存器是 16 位可读写寄存器。用作 GTDVm 寄存器的缓冲寄存器。各通道有 2 个 (共计 8 个) GTDBm 寄存器。

23.2.36 通用 PWM 定时器的输出保护功能状态寄存器 (GTSOS)

地址 GPT0.GTSOS 000C 2140h、GPT1.GTSOS 000C 21C0h、GPT2.GTSOS 000C 2240h、GPT3.GTSOS 000C 22C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOS [1:0]	
复位后的值	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

x: 不定值

位	符号	位名	功能	R/W
b1-b0	SOS[1:0] 位	输出保护功能状态位	b1 b0 0 0: 通常运行 0 1: 保护状态 (通过波谷或者波峰的传送设定为 $GTCCRA=0$) 1 0: 保护状态 (通过波谷的传送设定为 $GTCCRA \geq GTPR$) 1 1: 保护状态 (通过波峰的传送设定为 $GTCCRA \geq GTPR$)	R
b7-b2	—	保留位	读取值为“0”，写操作无效。	R
b8-b9	—	保留位	读取值为不定值，写操作无效。	R
b15-b10	—	保留位	读取值为“0”，写操作无效。	R

GTSOS 寄存器是表示输出保护功能状态的状态寄存器。只在三角波模式中自动设定死区时间 (GTDTCCR.TDE 位 =1) 时输出保护功能有效。

SOS 位 (输出保护功能状态位)

此位表示三角波 PWM 模式中的输出保护功能的状态。有关输出保护功能的详细内容，请参照“23.7.4 GTIOC 引脚输出的输出保护功能”。

23.2.37 通用 PWM 定时器的输出保护功能暂时解除寄存器 (GTSOTR)

地址 GPT0.GTSOTR 000C 2142h、GPT1.GTSOTR 000C 21C2h、GPT2.GTSOTR 000C 2242h、GPT3.GTSOTR 000C 22C2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOTR
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	SOTR	输出保护功能暂时解除位	0: 不解除保护状态 1: 解除保护状态	R/W
b15-b1	—	保留位	读写值都为“0”。	R/W

GTSOTR 寄存器是在输出保护状态时暂时解除 GTIOCnB 引脚输出的保护状态的寄存器。

只能在 GTSOS.SOS[1:0] 位为“10b” (表示通过波谷传送 $GTCCRA \geq GTPR$ 的保护状态) 时进行解除。其他保护状态不被解除。

SOTR 位 (输出保护功能暂时解除位)

此位设定在输出保护状态时是否暂时解除 GTIOCnB 引脚输出的保护状态。

如果将 SOTR 位置“1”，就解除第一个波谷以后的输出保护功能。如果将 SOTR 位置“0”，就重新开始第一个波谷以后的输出保护功能。

23.3 运行说明

23.3.1 基本运行

各通道的 16 位定时器进行递增计数、递减计数或者递增 / 递减计数。通过 GTPG 寄存器控制定时器周期。

如果 GTCNT 计数器的值与 GTCCRA 寄存器、GTCCRB 寄存器的值相同，就能分别更改 GTIOcnA 输出引脚和 GTIOcnB 输出引脚 (n=0 ~ 3)。另外，GTIOcnA 引脚和 GTIOcnB 引脚能用作输入引脚，GTCCRA 寄存器和 GTCCRB 寄存器能用作输入捕捉寄存器。

GTCCRC 寄存器和 GTCCRD 寄存器能作为 GTCCRA 寄存器的缓冲寄存器运行，GTCCRE 寄存器和 GTCCRF 寄存器能作为 GTCCRB 寄存器的缓冲寄存器运行。

23.3.1.1 计数器的运行

(1) 周期计数运行 (递增计数时)

如果将 GTSTR 寄存器的对应 CST 位置“1”，各通道的计数器就开始递增计数。当 GTCNT 计数器的值和 GTPR 寄存器的值相同 (上溢)，将 GTST.TCFPO 标志置“1”。此时，如果 GTINTAD.GTINTPR[0] 位为“1”，就请求 GTCIV 中断。在发生上溢后，GTCNT 计数器从“0000h”开始继续递增计数。

递增计数时的周期计数运行例子如图 23.2 所示。

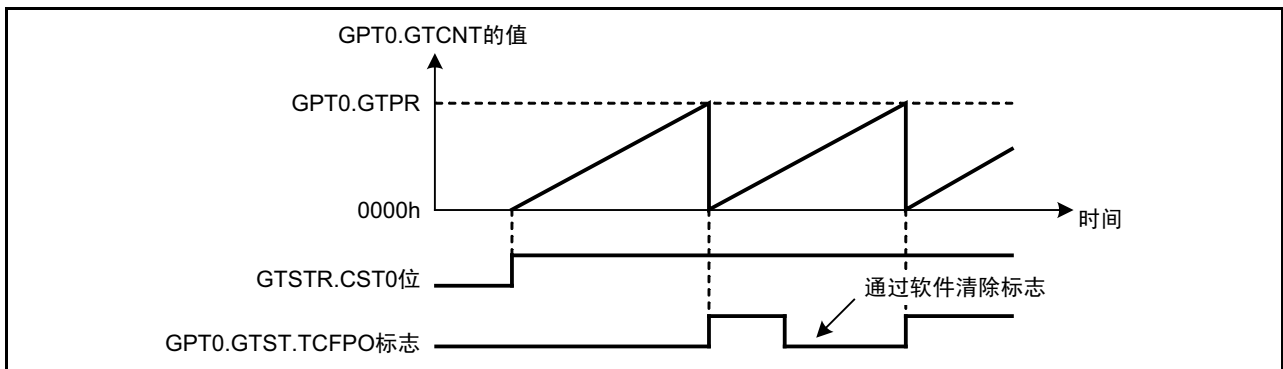


图 23.2 周期计数的运行例子 (递增计数时)

递增计数时的周期计数运行设定例子如图 23.3 所示。

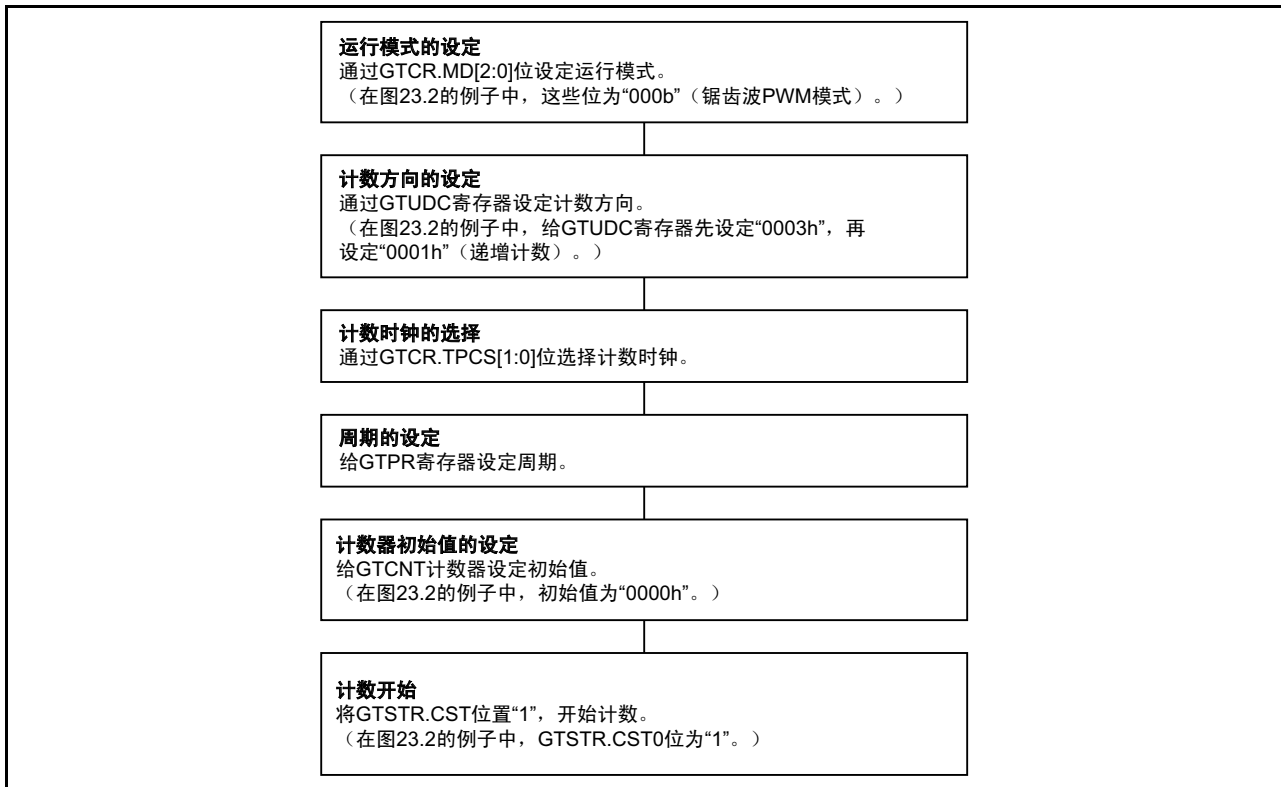


图 23.3 周期计数运行的设定例子 (递增计数时)

(2) 周期计数运行 (递减计数时)

各通道的计数器能通过设定 GTUDC 寄存器进行递减计数。如果将 GTCNT 计数器的值置“0”(下溢), GTST.TCFPU 标志就变为“1”。此时, 如果 GTINTAD.GTINTPR[1] 位为“1”, 就请求 GTCIV 中断。在发生下溢后, GTCNT 计数器从 GTPR 寄存器的值开始继续递减计数。

递减计数时的周期计数运行例子如图 23.4 所示。

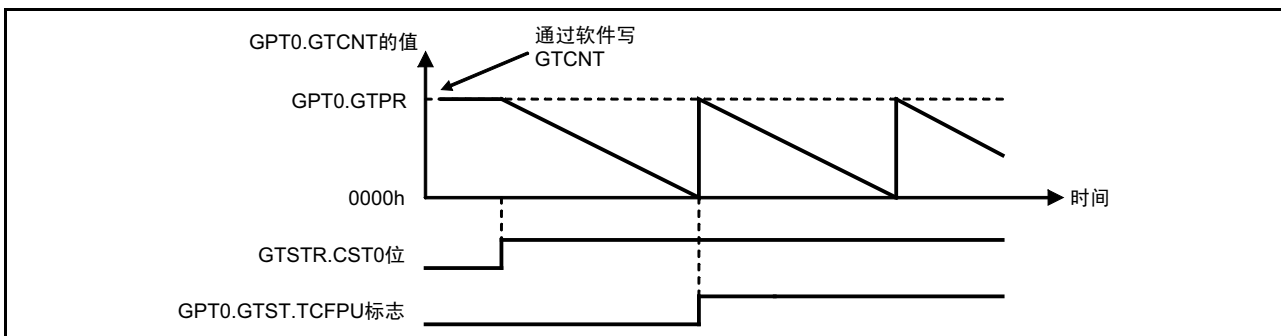


图 23.4 周期计数的运行例子 (递减计数时)

递减计数时的周期计数运行设定例子如图 23.5 所示。

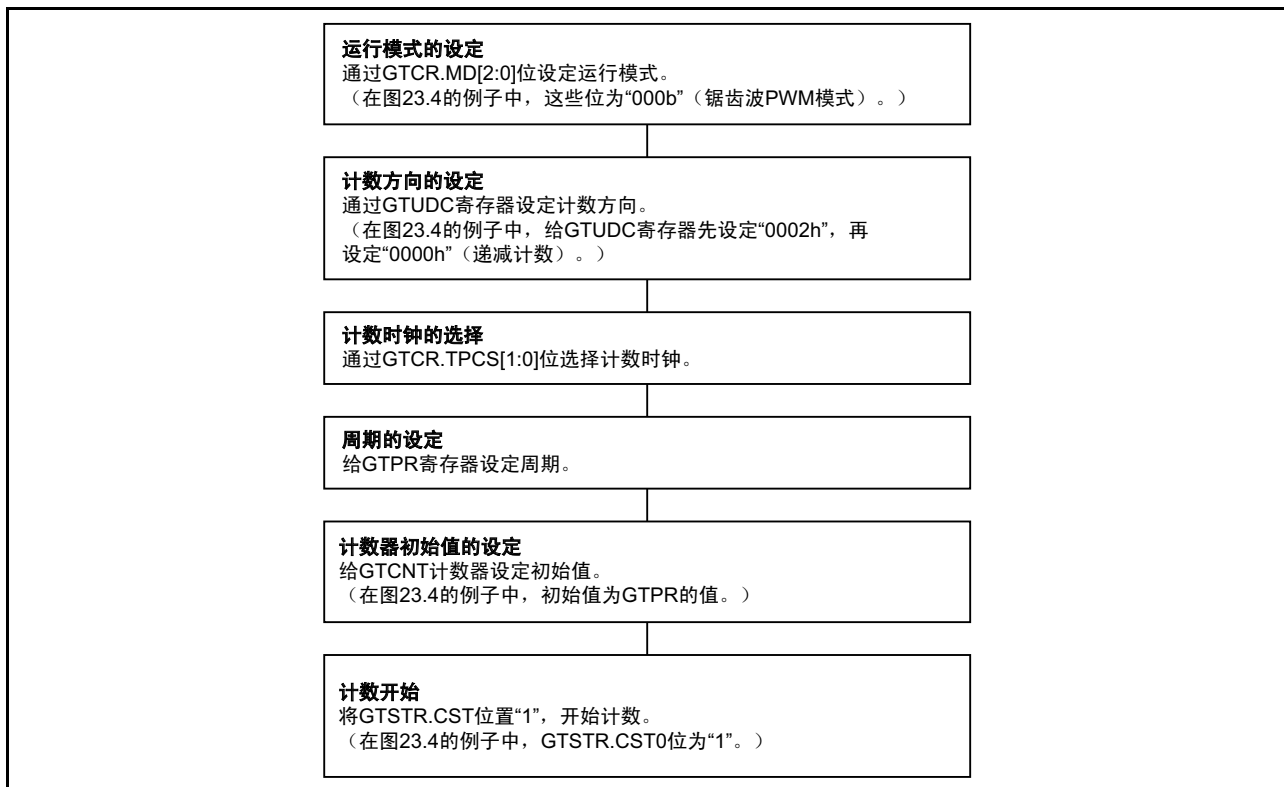


图 23.5 周期计数运行的设定例子 (递减计数时)

23.3.1.2 通过比较匹配进行的波形输出功能

如果 GPTn.GTCNT 计数器的值与 GPTn.GTCCRA 寄存器、GPTn.GTCCRB 寄存器的值相同, 就能分别从 GTIOCnA 输出引脚和 GTIOCnB 输出引脚输出 Low 电平、High 电平、或者进行交替输出。(n: 通道号)

即使在取决于 GPTn.GTPR 的“周期结束”时, 也能从 GTIOCnA 输出引脚和 GTIOCnB 输出引脚输出 Low 电平、High 电平、或者进行交替输出。“周期结束”如下所示。

- 在锯齿波进行递增计数的情况: 当 GPTn.GTCNT=GPTn.GTPR 时 (上溢)
- 在锯齿波进行递减计数的情况: 当 GPTn.GTCNT=0 时 (下溢)
- 在三角波的情况: 当 GPTn.GTCNT=0 时 (波谷)

(1) Low 电平输出 /High 电平输出

通过与 GTCCRA 寄存器、GTCCRB 寄存器的比较匹配进行的 Low 电平输出 /High 电平输出的运行例子如图 23.6 所示。

在此例子中假设通道 0 进行递增计数，并且通过与 GPT0.GTCCRA 寄存器的比较匹配从 GTIOC0A 引脚输出 High 电平，通过与 GPT0.GTCCRB 寄存器的比较匹配从 GTIOC0B 引脚输出 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。

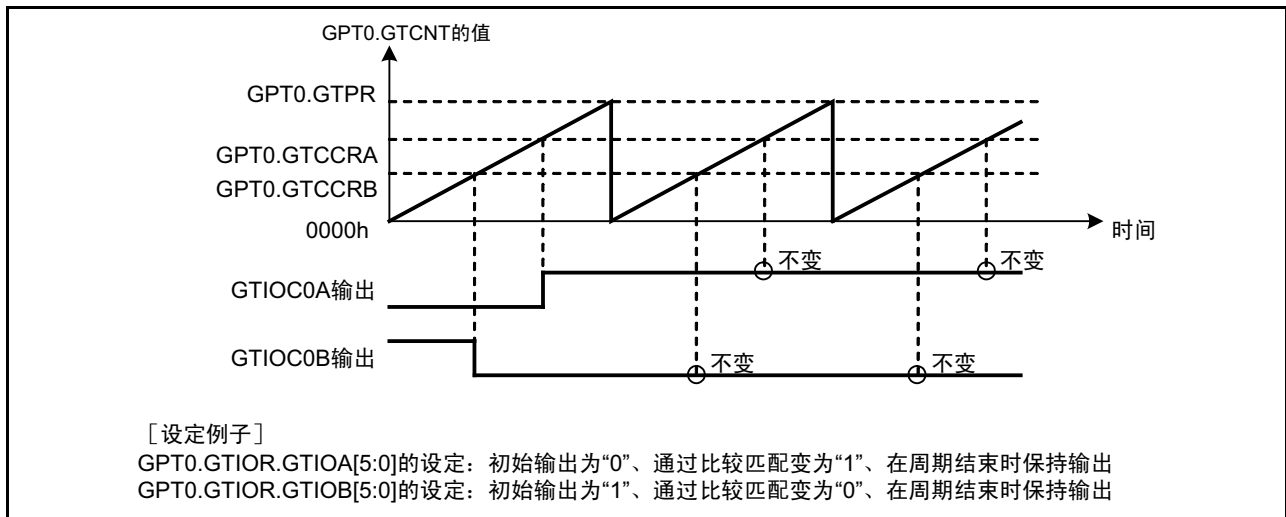


图 23.6 Low 电平输出 /High 电平输出的运行例子

Low 电平输出 /High 电平输出运行的设定例子如图 23.7 所示。



图 23.7 Low 电平输出 /High 电平输出运行的设定例子

(2) 交替输出

通过与 GTCCRA 寄存器、GTCCRB 寄存器的比较匹配进行的交替输出运行例子如图 23.8、图 23.9 所示。

在图 23.8 的例子中假设通道 0 进行递增计数，并且通过与 GPT0.GTCCRA 寄存器、GPT0.GTCCRB 寄存器的比较匹配从 GTIOC0A 引脚和 GTIOC0B 引脚进行交替输出。

在图 23.9 的例子中假设通道 0 进行递增计数，并且通过与 GPT0.GTCCRA 寄存器的比较匹配从 GTIOC0A 引脚进行交替输出，在周期结束时从 GTIOC0B 引脚进行交替输出。

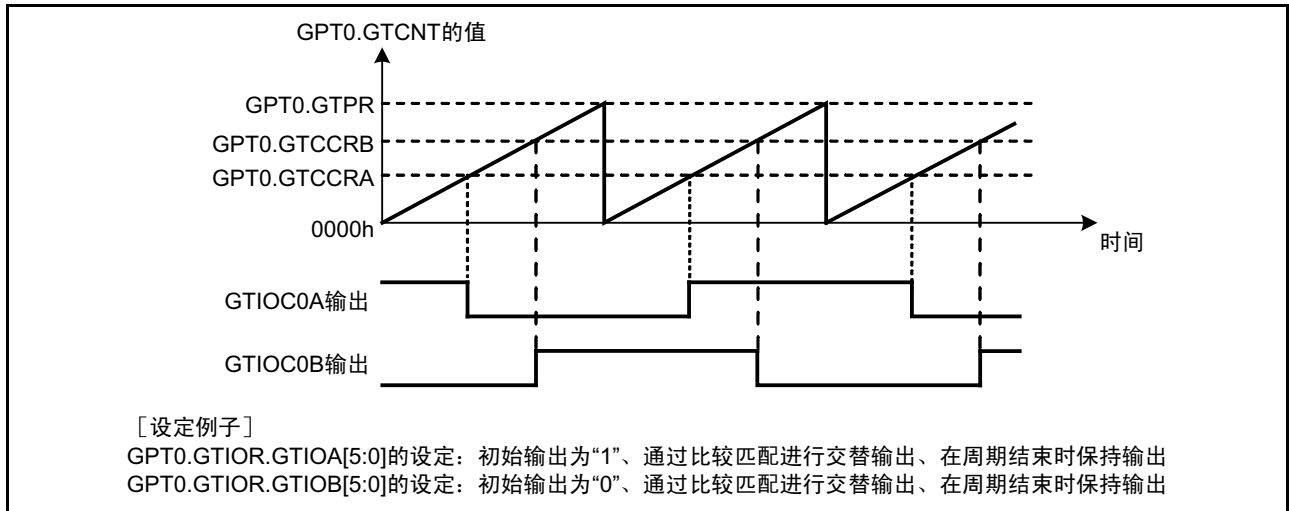


图 23.8 交替输出的运行例子 (1)

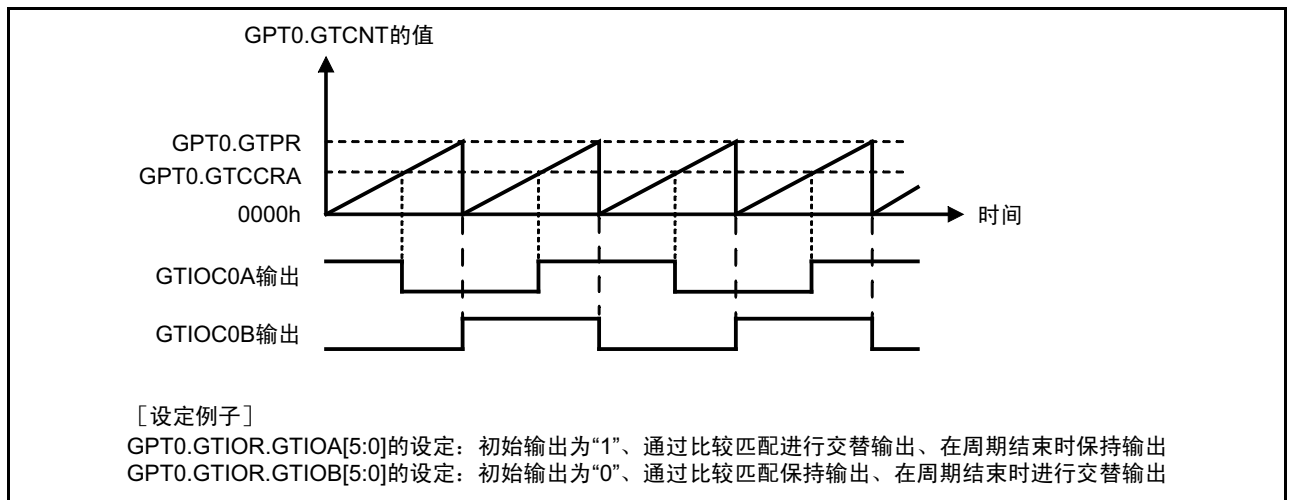


图 23.9 交替输出的运行例子 (2)

交替输出运行的设定例子如图 23.10 所示



图 23.10 交替输出运行的设定例子

23.3.1.3 输入捕捉功能

能检测 GTIOCnA 输入引脚和 GTIOCnB 输入引脚的边沿，并且将 GPTn.GTCNT 计数器的值分别传送到 GPTn.GTCCRA 寄存器和 GPTn.GTCCRB 寄存器 (n: 通道号)。从上升沿、下降沿和双边沿中选择检测边沿。

输入捕捉功能的运行例子如图 23.11 所示。

在此例子中假设通道 0 进行递增计数，并且在 GTIOC0A 输入引脚的双边沿进行输入捕捉，在 GTIOC0B 输入引脚的上升沿进行输入捕捉。

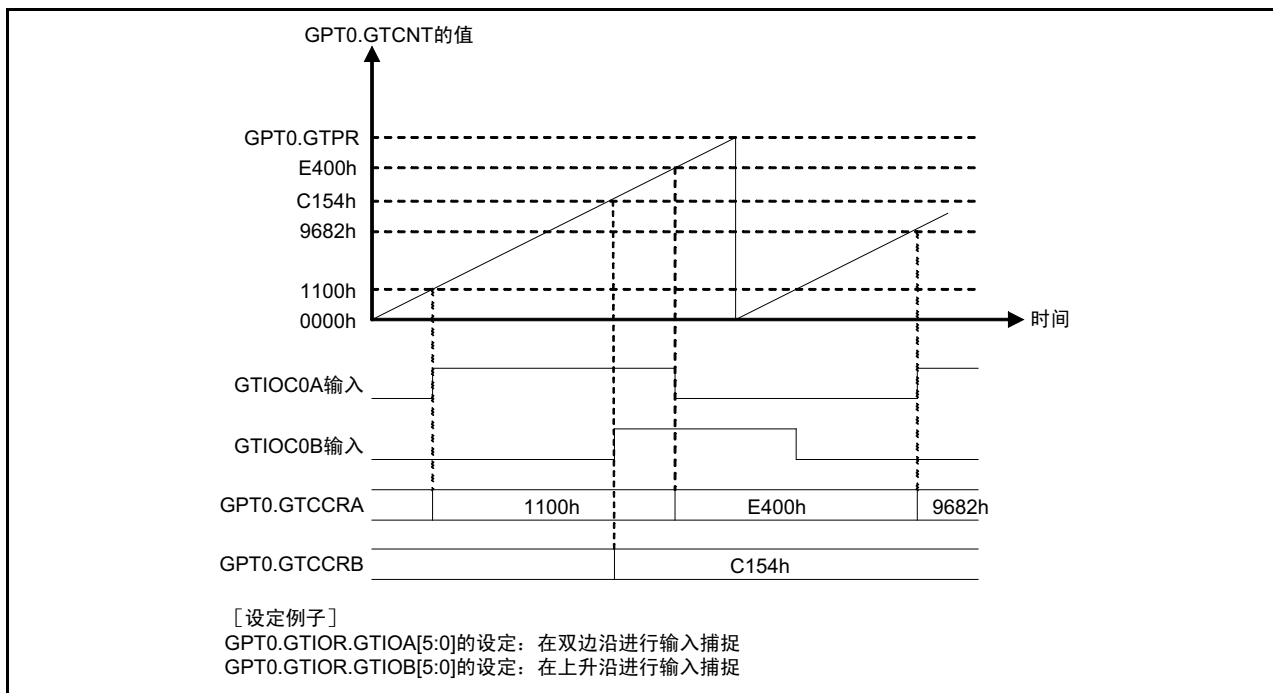


图 23.11 输入捕捉的运行例子

输入捕捉运行的设定例子如图 23.12 所示。



图 23.12 输入捕捉运行的设定例子

23.3.2 缓冲运行

能通过设定 GTBER 寄存器进行以下的缓冲运行。

- 进行 GTCCRA 寄存器和 GTCCRC 寄存器、GTCCRD 寄存器组合的缓冲运行
- 进行 GTCCRB 寄存器和 GTCCRE 寄存器、GTCCRF 寄存器组合的缓冲运行
- 进行 GTPR 寄存器和 GTPBR 寄存器、GTPDBR 寄存器组合的缓冲运行
- 进行 GTADTRA 寄存器和 GTADTBRA 寄存器、GTADTDBRA 寄存器组合的缓冲运行
- 进行 GTADTRB 寄存器和 GTADTBRB 寄存器、GTADTDBRB 寄存器组合的缓冲运行

或者，能通过设定 GTDTCR 寄存器进行以下的缓冲运行。

- 进行 GTDVU 寄存器和 GTDBU 寄存器组合的缓冲运行
- 进行 GTDVD 寄存器和 GTDBD 寄存器组合的缓冲运行

23.3.2.1 GTPR 寄存器的缓冲运行

GTPBR 寄存器作为 GTPR 寄存器的缓冲寄存器运行，GTPDBR 寄存器作为 GTPBR 寄存器的缓冲寄存器（GTPR 寄存器的双缓冲寄存器）运行。

锯齿波时的缓冲传送时序为发生上溢（递增计数）或者下溢（递减计数）时，三角波时的缓冲传送时序为波谷。

让 GTPR 寄存器进行双缓冲运行时，必须将 GTBER.PR[1:0] 位设定为“10b”或者“11b”，让 GTPR 寄存器进行单缓冲运行时，必须将 GTBER.PR[1:0] 位设定为“01b”，不让 GTPR 寄存器进行缓冲运行时，必须将 GTBER.PR[1:0] 位设定为“00b”。

GTPR 寄存器的缓冲运行例子如图 23.13 ~ 图 23.15，GTPR 寄存器的缓冲运行设定例子如图 23.16 所示。

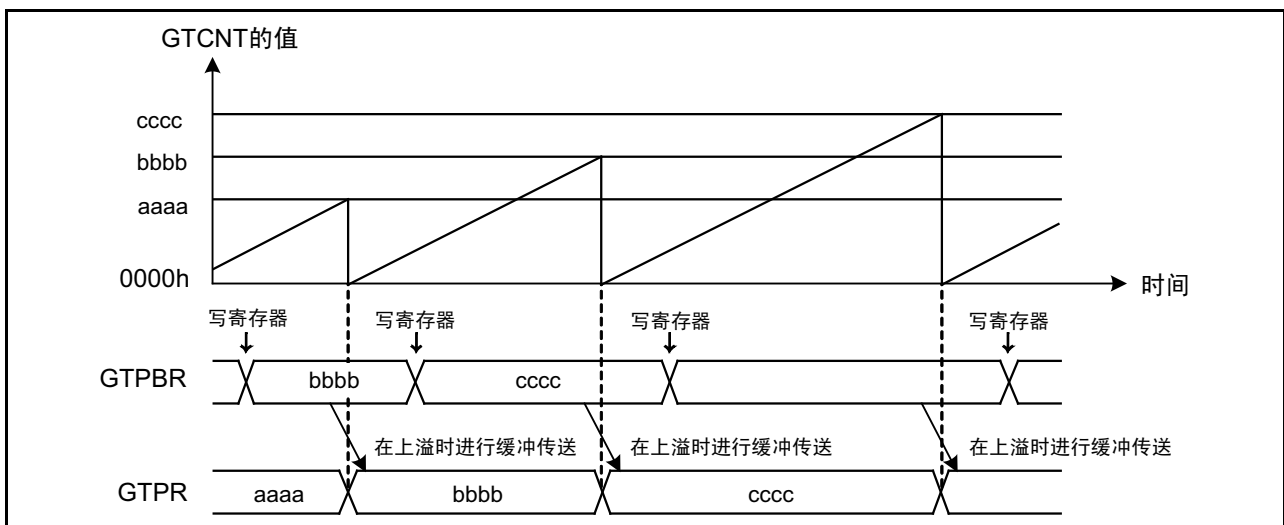


图 23.13 GTPR 寄存器的缓冲运行例子（在锯齿波进行递增计数的情况）

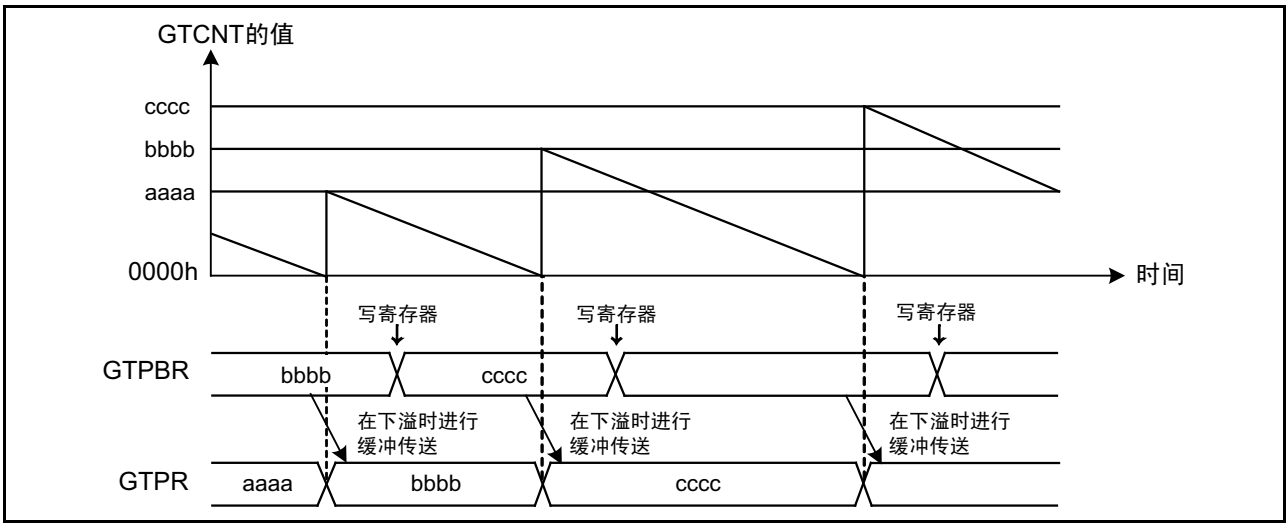


图 23.14 GTPR 寄存器的缓冲运行例子 (在锯齿波进行递减计数的情况)

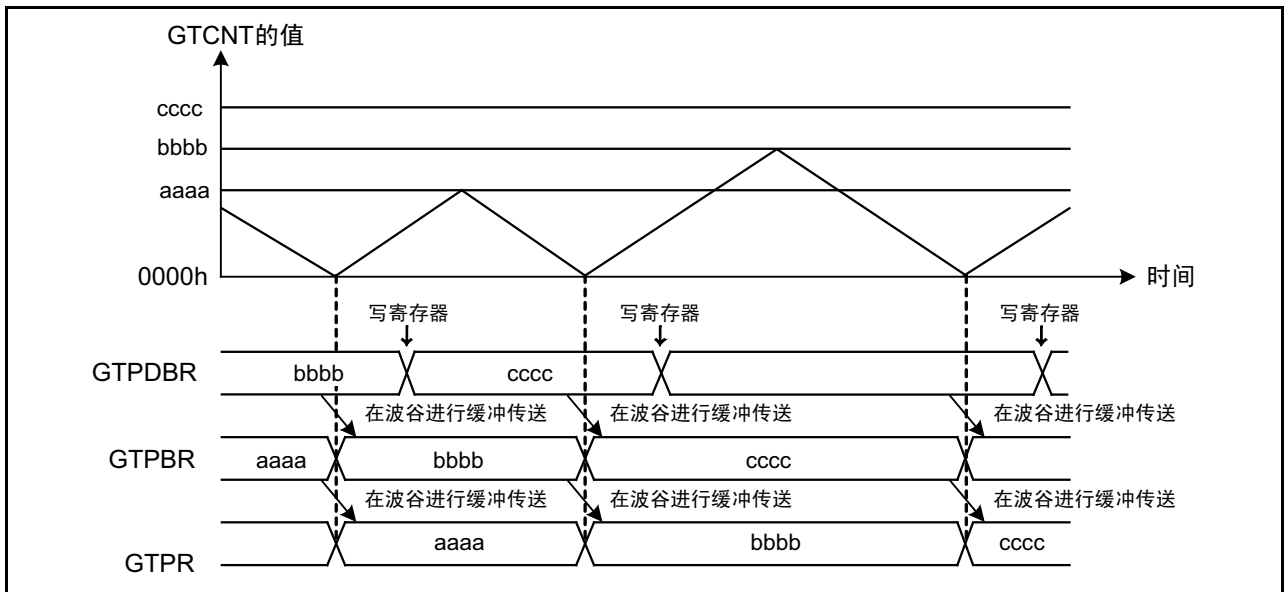


图 23.15 GTPR 寄存器的缓冲运行例子 (三角波的情况)

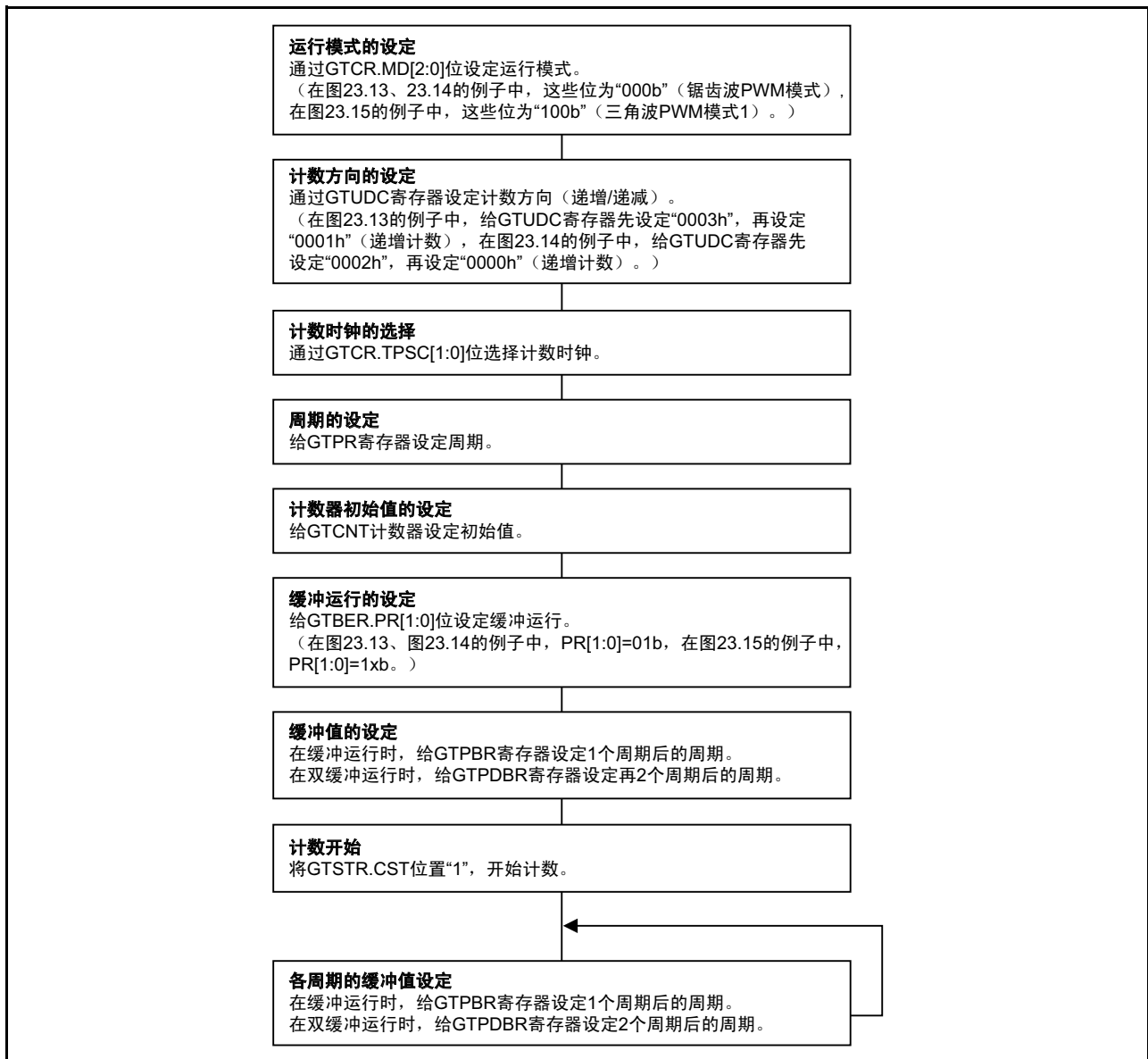


图 23.16 GTPR 寄存器的缓冲运行设定例子

23.3.2.2 GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行

GTCCRC 寄存器作为 GTCCRA 寄存器的缓冲寄存器运行，GTCCRD 寄存器作为 GTCCRC 寄存器的缓冲寄存器（GTCCRA 寄存器的双缓冲寄存器）运行。同样，GTCCRE 寄存器作为 GTCCRB 寄存器的缓冲寄存器运行，GTCCRF 寄存器作为 GTCCRE 寄存器的缓冲寄存器（GTCCRB 寄存器的双缓冲寄存器）运行。

让 GTCCRA 寄存器和 GTCCRB 寄存器进行双缓冲运行时，必须分别将 GTBER.CCRA[1:0] 位和 GTBER.CCRB[1:0] 位设定为“10b”或者“11b”，让 GTCCRA 寄存器和 GTCCRB 寄存器进行单缓冲运行时，必须将 GTBER.CCRA[1:0] 位和 GTBER.CCRB[1:0] 位设定为“01b”，不让 GTCCRA 寄存器和 GTCCRB 寄存器进行冲运行时，必须将 GTBER.CCRA[1:0] 位和 GTBER.CCRB[1:0] 位设定为“00b”。

(1) GTCCRA 寄存器和 GTCCRB 寄存器作为输出比较寄存器运行的情况

锯齿波时的缓冲传送时序为发生上溢（递增计数）或者下溢（递减计数）时，三角波时的缓冲传送时序为波谷。

GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行例子如图 23.17 ~ 图 23.19，GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行设定例子如图 23.20 所示。

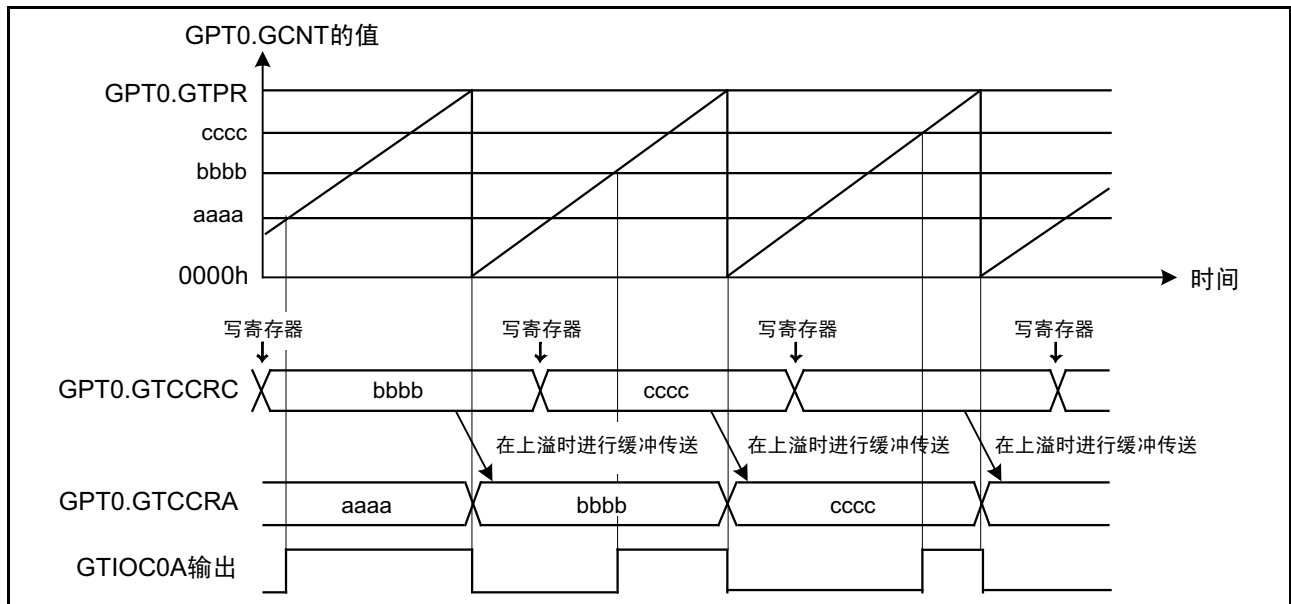


图 23.17 GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行例子
（输出比较、在锯齿波进行递增计数、在 GTCCRA 比较匹配时输出 High 电平、在周期结束时输出 Low 电平的情况）

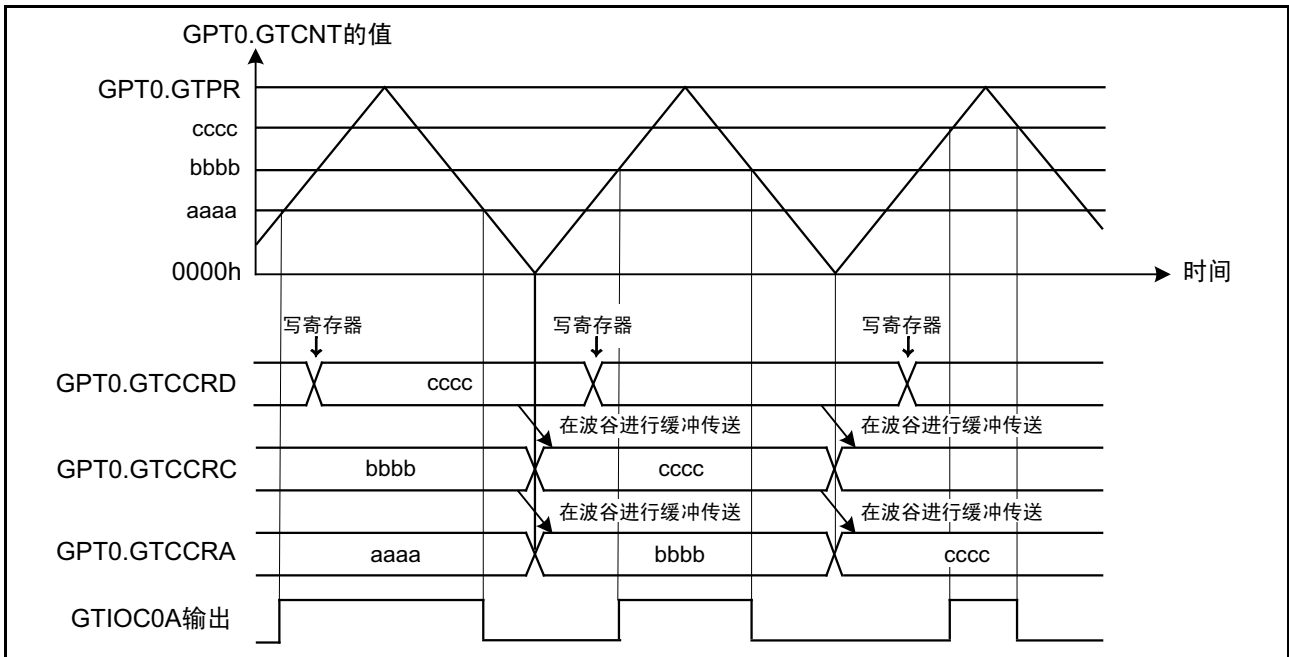


图 23.18 GTCCRA 寄存器和 GTCCRB 寄存器的双缓冲运行例子
 (输出比较、三角波、在波谷进行缓冲传送、在 GTCCRA 比较匹配时进行交替输出、在周期结束时保持输出的情况)

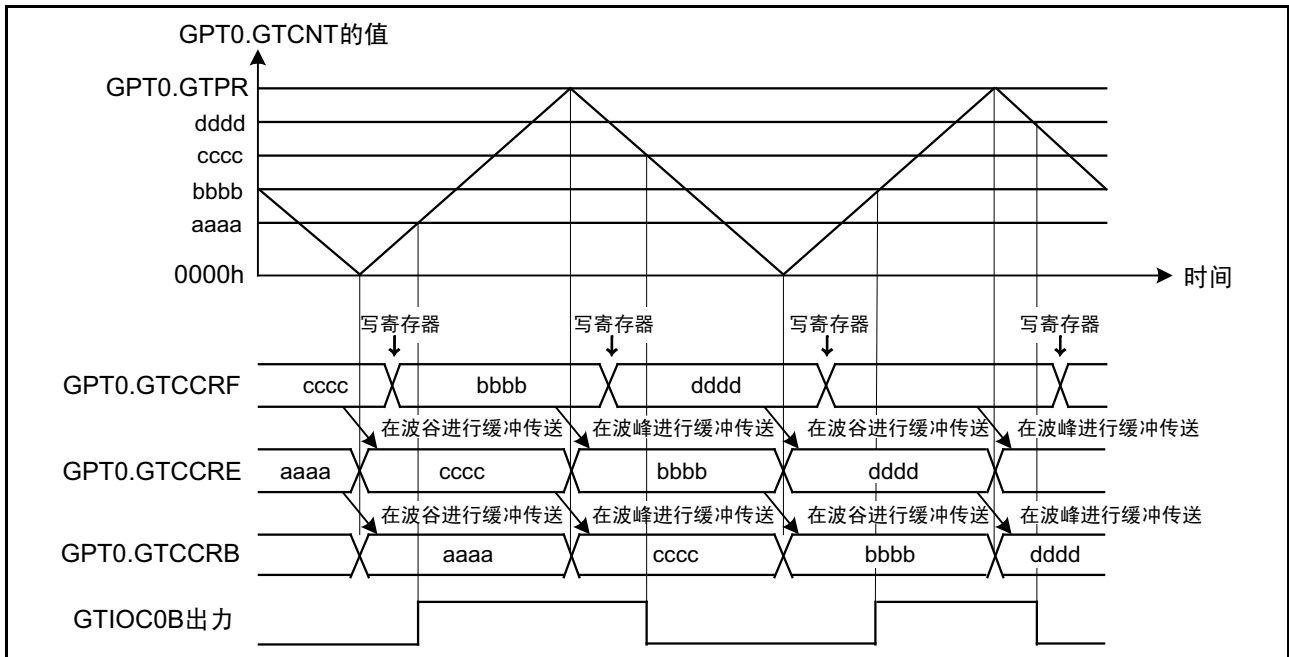


图 23.19 GTCCRA 寄存器和 GTCCRB 寄存器的双缓冲运行例子
 (输出比较、三角波、在波峰和波谷都进行缓冲传送、在 GTCCRB 比较匹配时进行交替输出、在周期结束时保持输出的情况)

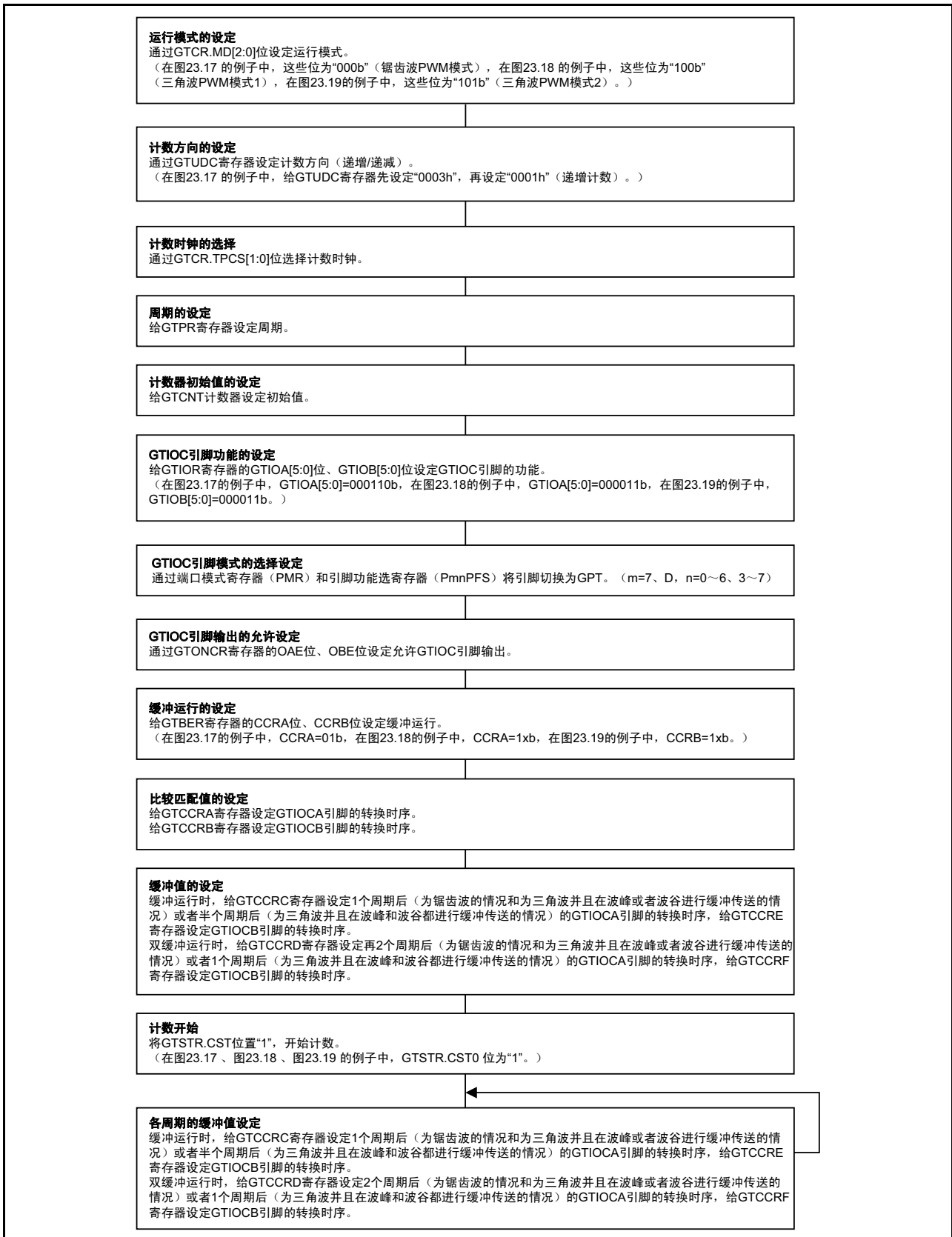


图 23.20 GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行设定例子 (输出比较的情况)

(2) GTCCRA 寄存器和 GTCCRB 寄存器作为输入捕捉寄存器运行的情况

缓冲传送的时序为发生输入捕捉时。如果发生输入捕捉，就在将 GTCNT 计数器的值传送到 GTCCRA 寄存器和 GTCCRB 寄存器的同时，将保存在 GTCCRA 寄存器和 GTCCRB 寄存器的值传送到缓冲寄存器。

GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行例子如图 23.21、图 23.22，GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行设定例子如图 23.23 所示。

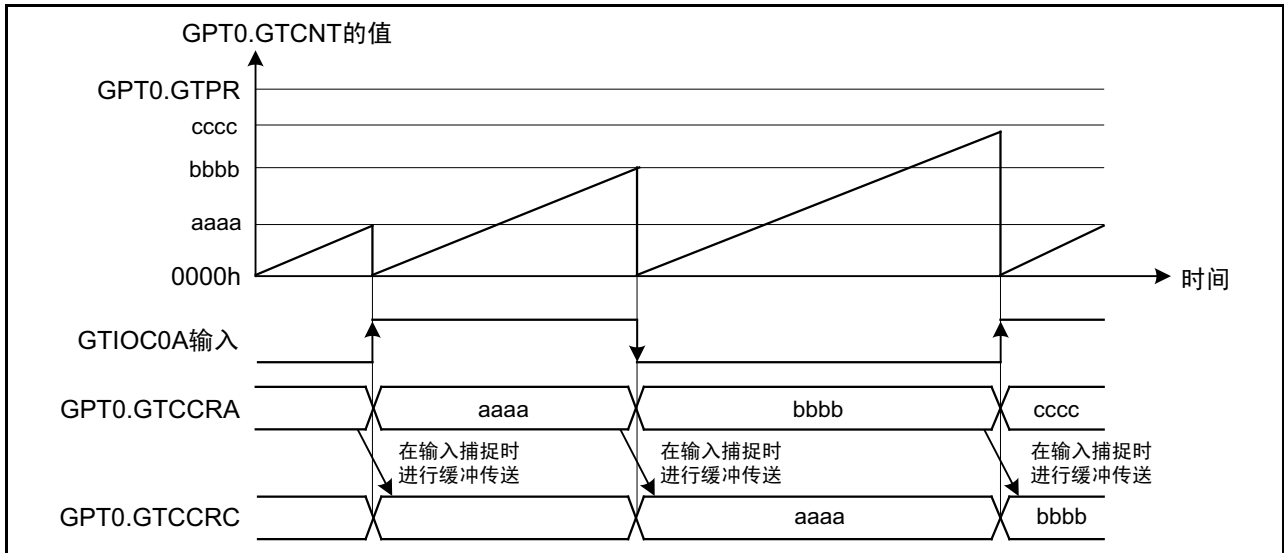


图 23.21 GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行例子
(在锯齿波进行递增计数、在 GTIOC0A 输入的双边沿进行输入捕捉、在 GTCCRA 寄存器的输入捕捉时进行 GTCNT 计数器清除的情况)

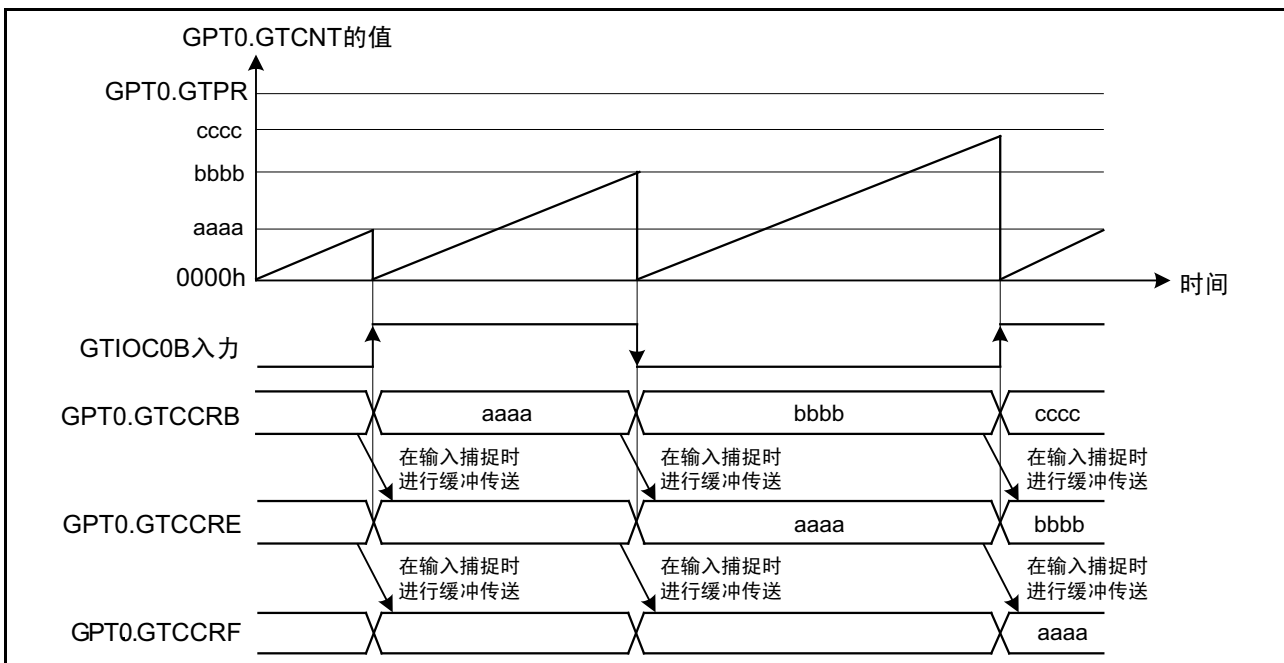


图 23.22 GTCCRA 寄存器和 GTCCRB 寄存器的双缓冲运行例子
(在锯齿波进行递增计数、在 GTIOC0B 输入的双边沿进行输入捕捉、在 GTCCRB 寄存器的输入捕捉时进行 GTCNT 计数器清除的情况)

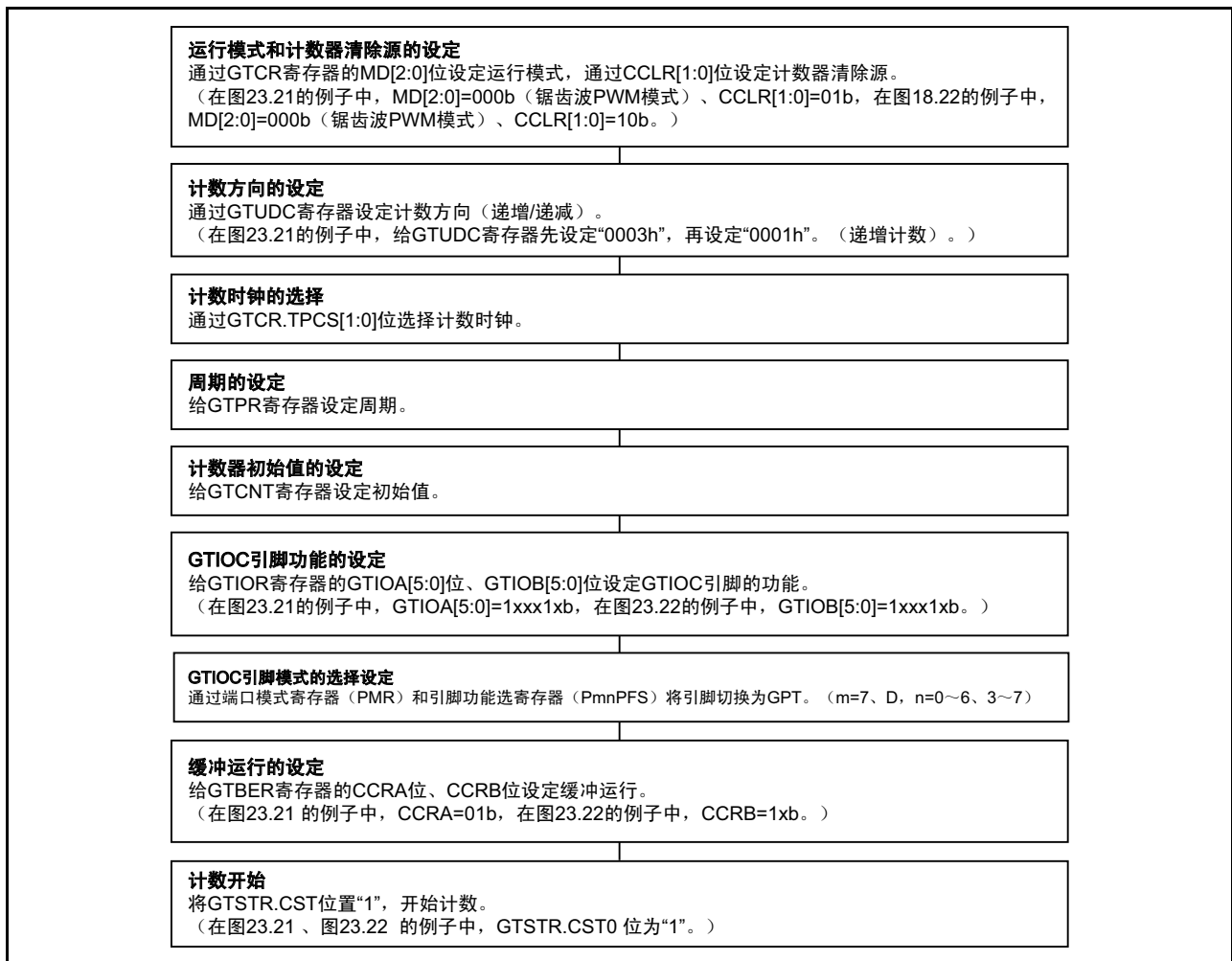


图 23.23 GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行设定例子 (输入捕捉的情况)

23.3.2.3 GTADTRA 寄存器和 GTADTRB 寄存器的缓冲运行

GTADTBRA 寄存器作为 GTADTRA 寄存器的缓冲寄存器运行，GTADTDBRA 寄存器作为 GTADTBRA 寄存器的缓冲寄存器（GTADTRA 寄存器的双缓冲寄存器）运行。同样，GTADTBRB 寄存器作为 GTADTRB 寄存器的缓冲寄存器运行，GTADTDBRB 寄存器作为 GTADTBRB 寄存器的缓冲寄存器（GTADTRB 寄存器的双缓冲寄存器）运行。

让 GTADTRA 寄存器和 GTADTRB 寄存器进行双缓冲运行时，必须分别将 GTBER.ADTDA 位和 GTBER.ADTDB 位设定为“1”，让 GTADTRA 寄存器和 GTADTRB 寄存器进行单缓冲运行时，必须将 GTBER.ADTDA 位和 GTBER.ADTDB 位设定为“0”，不让 GTADTRA 寄存器和 GTADTRB 寄存器进行缓冲运行时，必须分别将 GTBER.ADTTA[1:0] 位和 GTBER.ADTTB[1:0] 位设定为“00b”。

能通过 GTBER.ADTTA[1:0] 位设定缓冲传送的时序。锯齿波时的缓冲时序为发生上溢（递增计数）或者下溢（递减计数）时，三角波时的缓冲传送时序在 GTBER.ADTTA[1:0] 位是“01b”时为波峰、“10b”时为波谷，“11b”时为波谷和波峰。

GTADTRA 寄存器和 GTADTRB 寄存器的缓冲运行例子如图 23.24 ~ 图 23.26，GTADTRA 寄存器和 GTADTRB 寄存器的缓冲运行设定例子如图 23.27 所示。

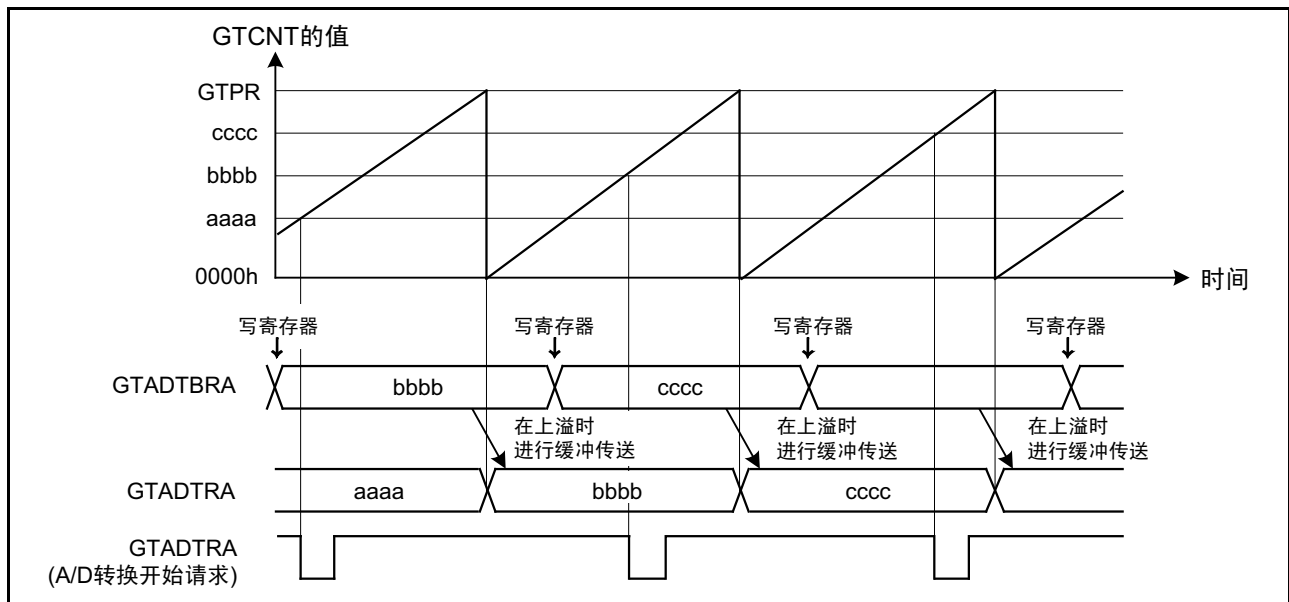


图 23.24 GTADTRA 寄存器和 GTADTRB 寄存器的缓冲运行例子
（在锯齿波进行递增计数、在递增计数时产生 A/D 转换开始请求的情况）

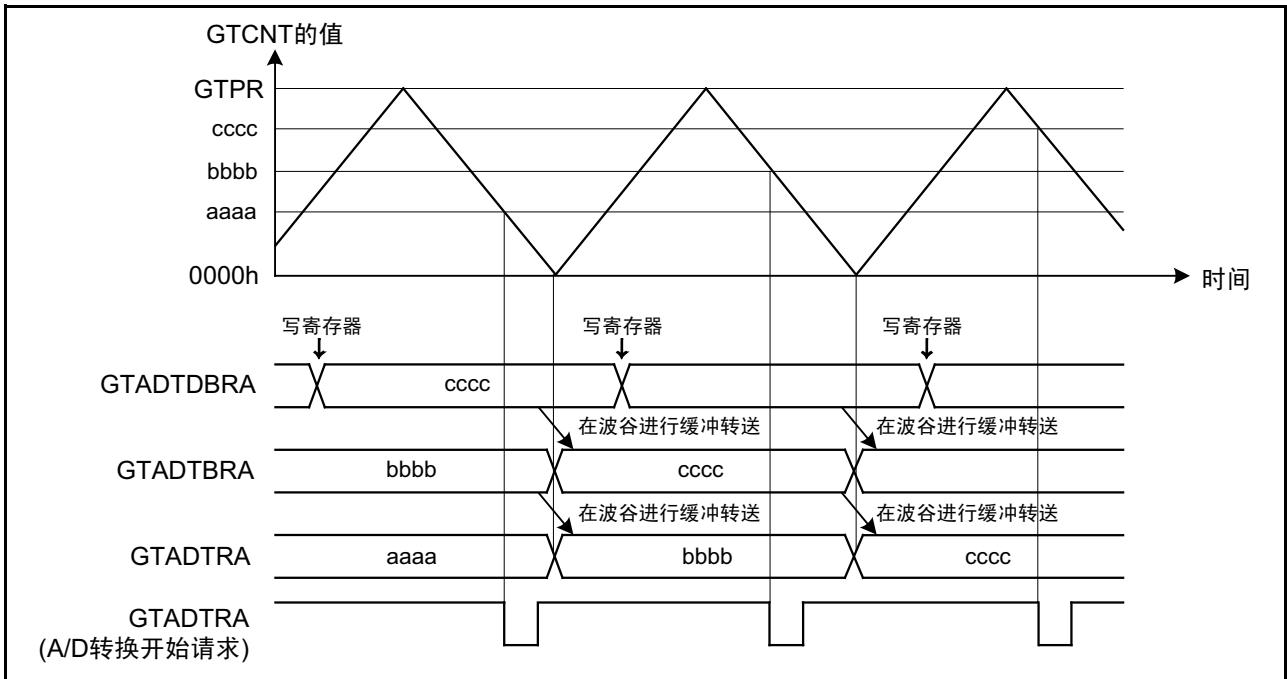


图 23.25 GTADTRA 寄存器和 GTADTBRA 寄存器的双缓冲运行例子
(三角波、在波谷进行缓冲转送、在递减计数时产生 A/D 转换开始请求的情况)

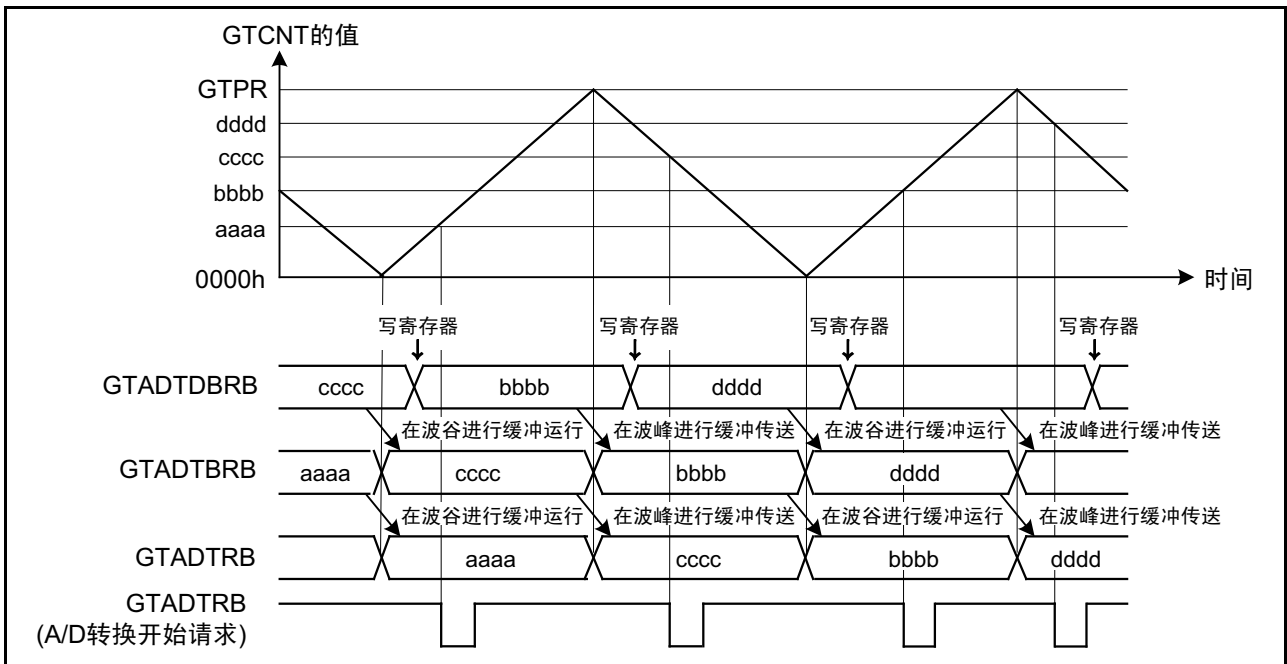


图 23.26 GTADTRA 寄存器和 GTADTRB 寄存器的双缓冲运行例子
(三角波、在波谷和波峰都进行缓冲转送、在递增计数和递减计数时都产生 A/D 转换请求的情况)

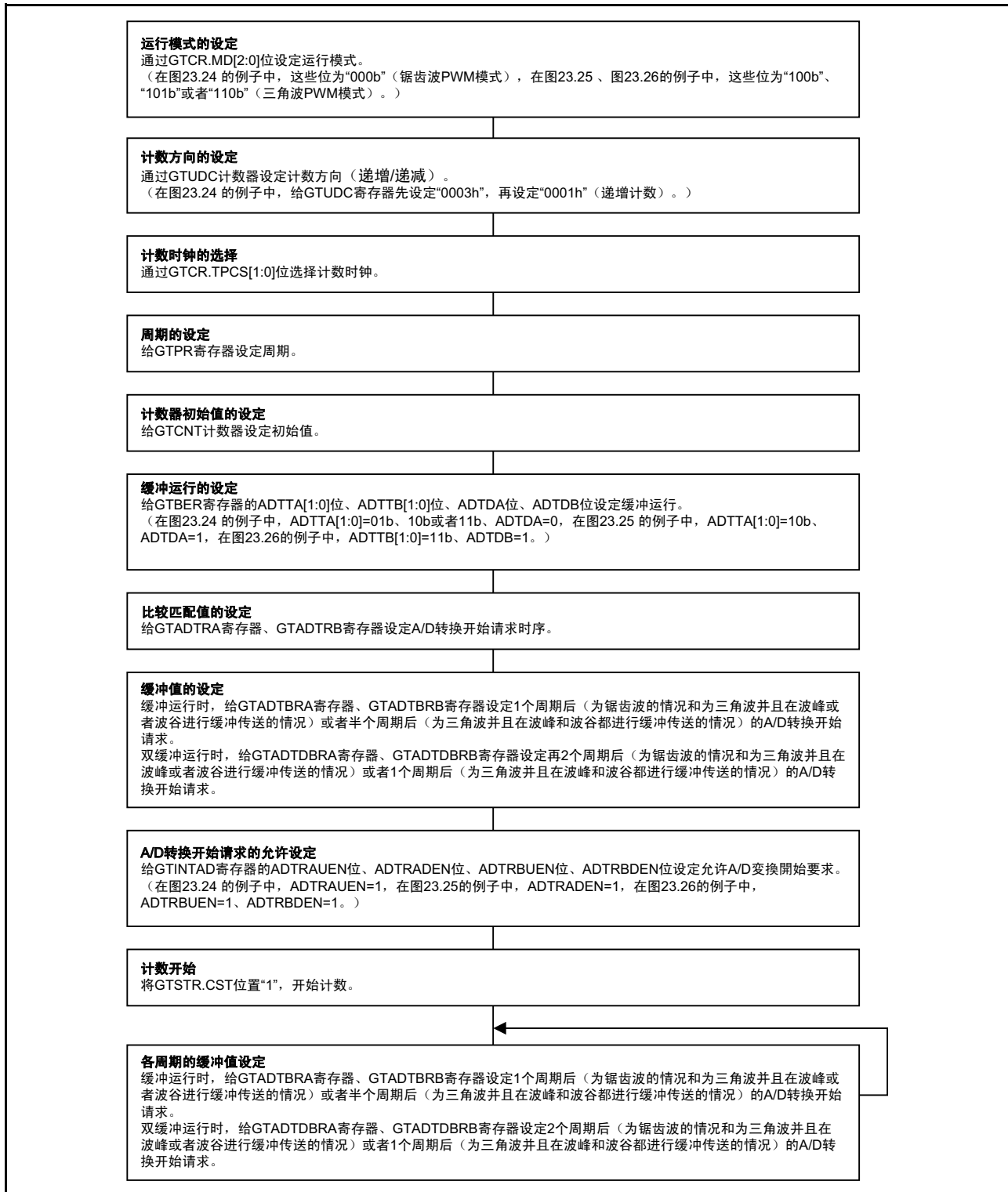


图 23.27 GTADTRA 寄存器和 GTADTRB 寄存器的缓冲运行设定例子

23.3.3 PWM 输出运行模式

能通过 GPTn.GTCNT 计数器和 GPTn.GTCCRA 寄存器、GPTn.GTCCRB 寄存器的比较匹配，对 GTIOCnA 引脚和 GTIOCnB 引脚输出 PWM 波形 (n: 通道号)。能给全部通道独立设定运行模式，也能进行通道间的同步运行。

能通过设定 GTDTCR 寄存器、GTDVU 寄存器和 GTDVD 寄存器，将用于带死区时间的反相波形的比较匹配值自动设定到 GTCCRB 寄存器。

(1) 锯齿波 PWM 模式

锯齿波 PWM 模式是将周期设定到 GPTn.GTPR 寄存器，使 GPTn.GTCNT 计数器进行锯齿波 (半波) 运行，并且通过 GPTn.GTCCRA 寄存器、GPTn.GTCCRB 寄存器的比较匹配将 PWM 波形输出到 GTIOCnA 引脚和 GTIOCnB 引脚的模式 (n: 通道数)。能通过 GTIOR 寄存器将引脚的输出值设定为在比较匹配时进行 Low 电平输出 / High 电平输出 / 交替输出，在周期结束时进行 Low 电平输出 / High 电平输出 / 交替输出。

锯齿波 PWM 模式的运行例子如图 23.28，锯齿波 PWM 模式的设定例子如图 23.29 所示。

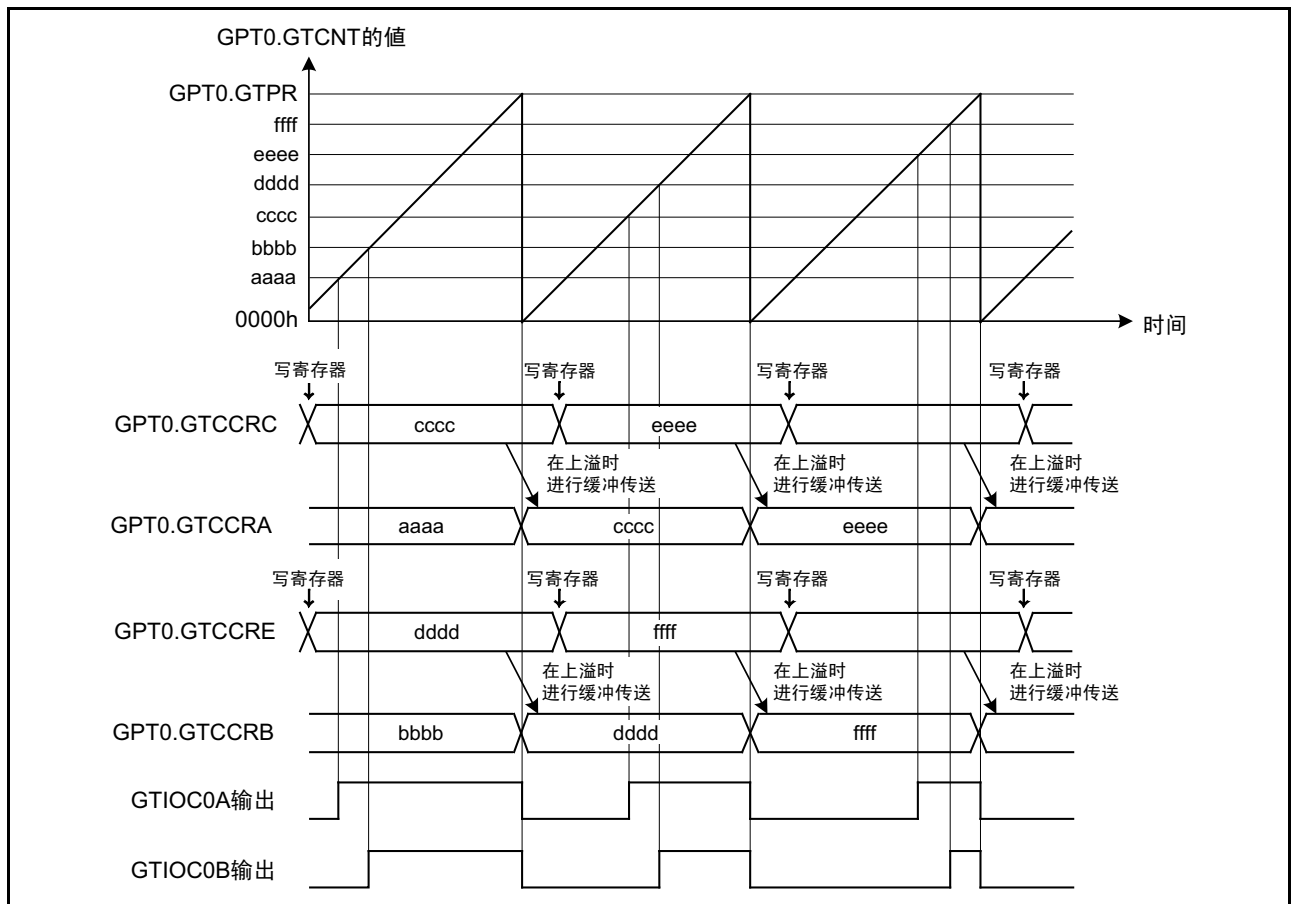


图 23.28 锯齿波 PWM 模式的运行例子
(递增计数、缓冲运行、在 GTCCRA/B 比较匹配时进行 High 电平输出、在周期结束时进行 Low 电平输出的情况)

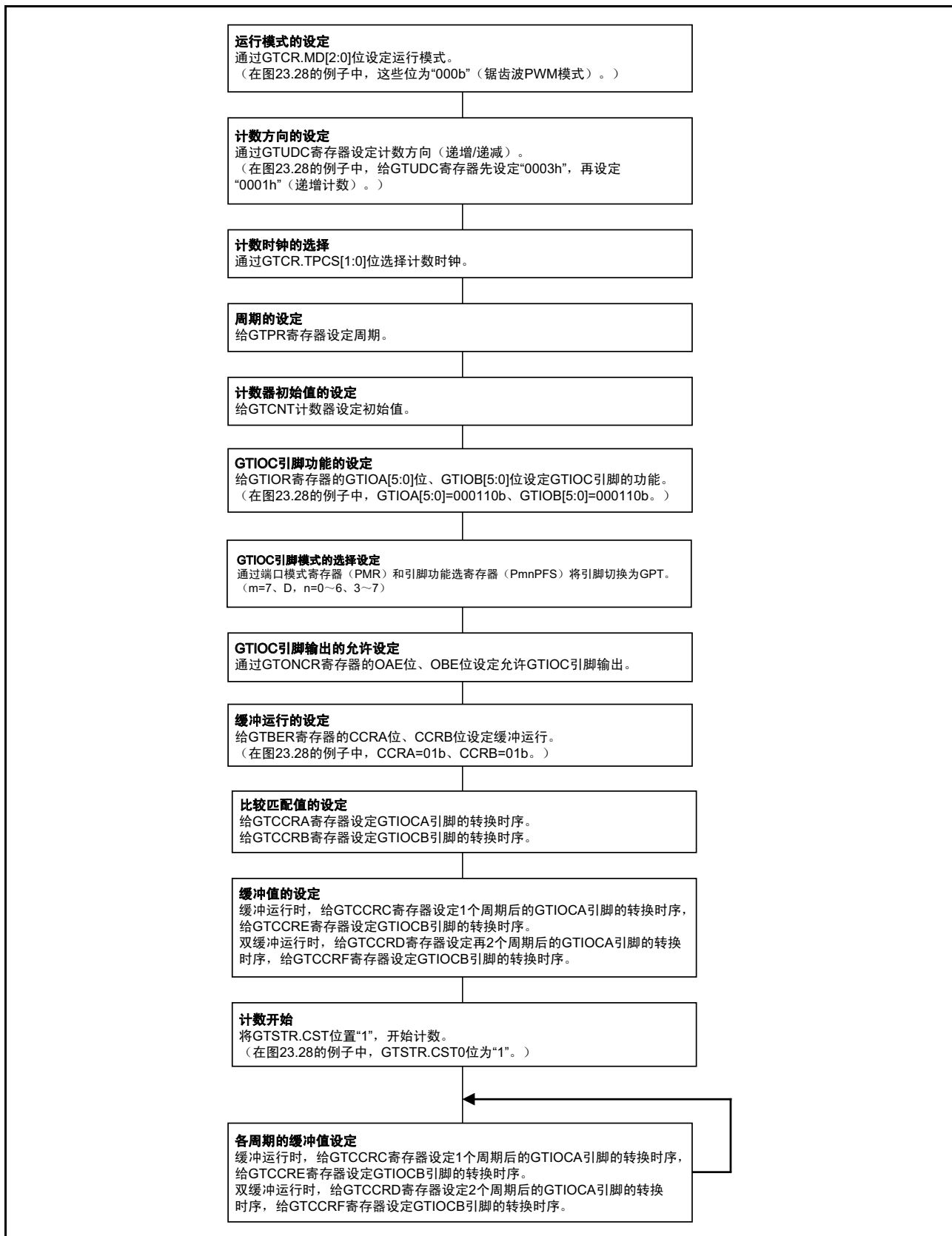


图 23.29 锯齿波 PWM 模式的设定例子

(2) 锯齿波单触发脉冲模式

锯齿波单触发脉冲模式是将周期设定到 GPTn.GTPR 寄存器，使 GPTn.GTCNT 计数器进行锯齿波（半波）运行，并且固定为缓冲运行，通过 GPTn.GTCCRA 寄存器、GPTn.GTCCRB 寄存器的比较匹配，将 PWM 波形输出到 GTIOCnA 引脚和 GTIOCnB 引脚的模式（n：通道号）。锯齿波单触发脉冲模式的缓冲运行与通常的缓冲运行不同，在周期结束时，进行从 GTCCRC 寄存器到 GTCCRA 寄存器、从 GTCCRE 寄存器到 GTCCRB 寄存器、从 GTCCRD 寄存器到暂存器 A、从 GTCCRF 寄存器到暂存器 B 的缓冲传送。而且在 GTCCRA 寄存器的比较匹配时进行从暂存器 A 到 GTCCRA 寄存器的缓冲传送，在 GTCCRB 寄存器的比较匹配时进行从暂存器 B 到 GTCCRB 寄存器的缓冲传送。能通过 GTIOR 寄存器将引脚的输出值设定为在比较匹配时进行 Low 电平输出 /High 电平输出 / 交替输出，在周期结束时进行 Low 电平输出 /High 电平输出 / 交替输出。

能通过设定 GTDTCR 寄存器、GTDVU 寄存器和 GTDVD 寄存器，将用于带死区时间的反相波形的比较匹配值自动设定到 GTCCRB 寄存器。

锯齿波单触发脉冲模式的运行例子如图 23.30，锯齿波单触发脉冲模式的设定例子如图 23.31 所示。

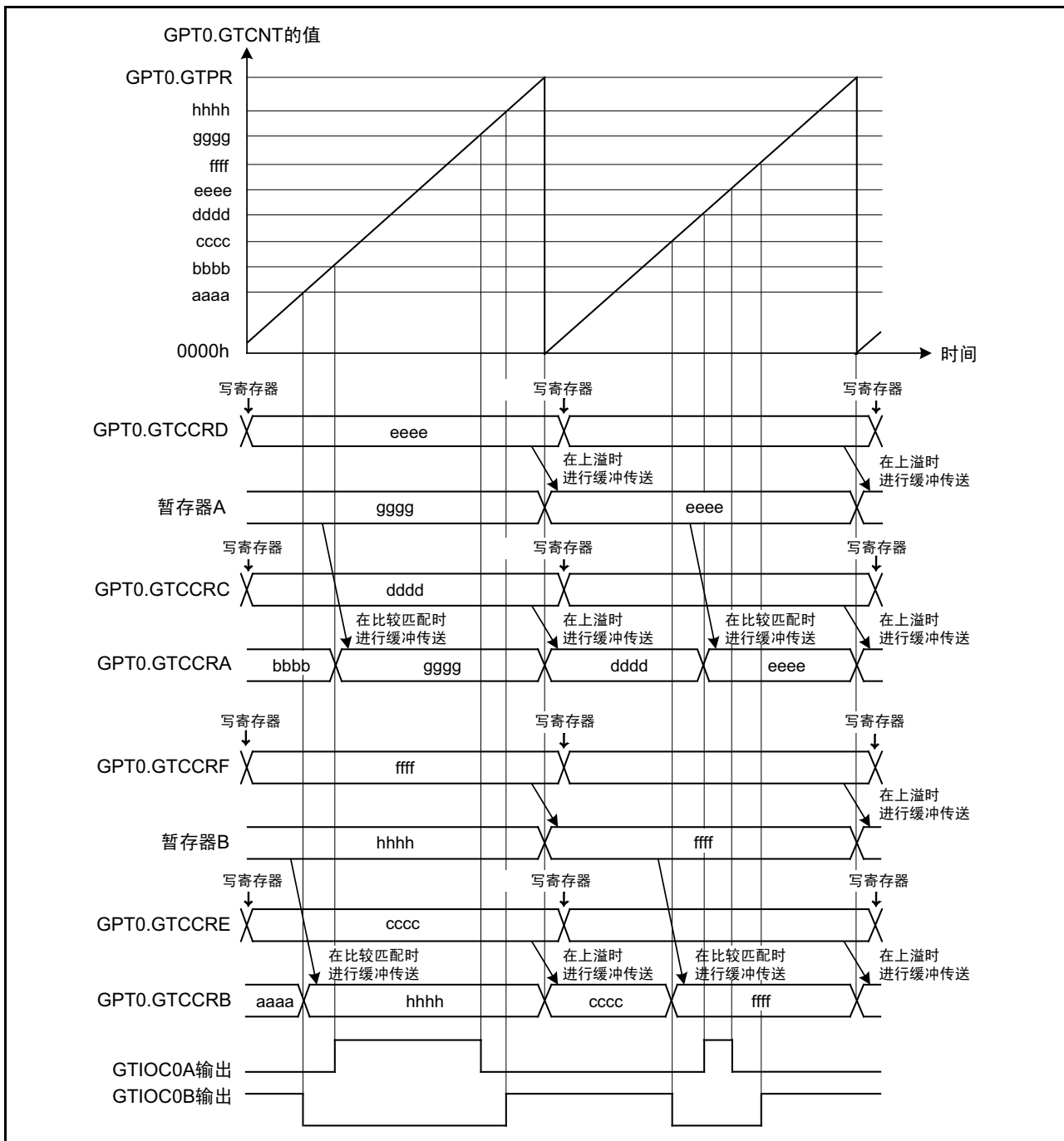


图 23.30 锯齿波单触发脉冲模式的运行例子
 (递增计数、在计数开始时 GTIOC0A 引脚为 Low 电平输出 /GTIOC0B 引脚为 High 电平输出、在 GTCRA/B 比较匹配时进行交替输出、在周期结束保持输出的情况)

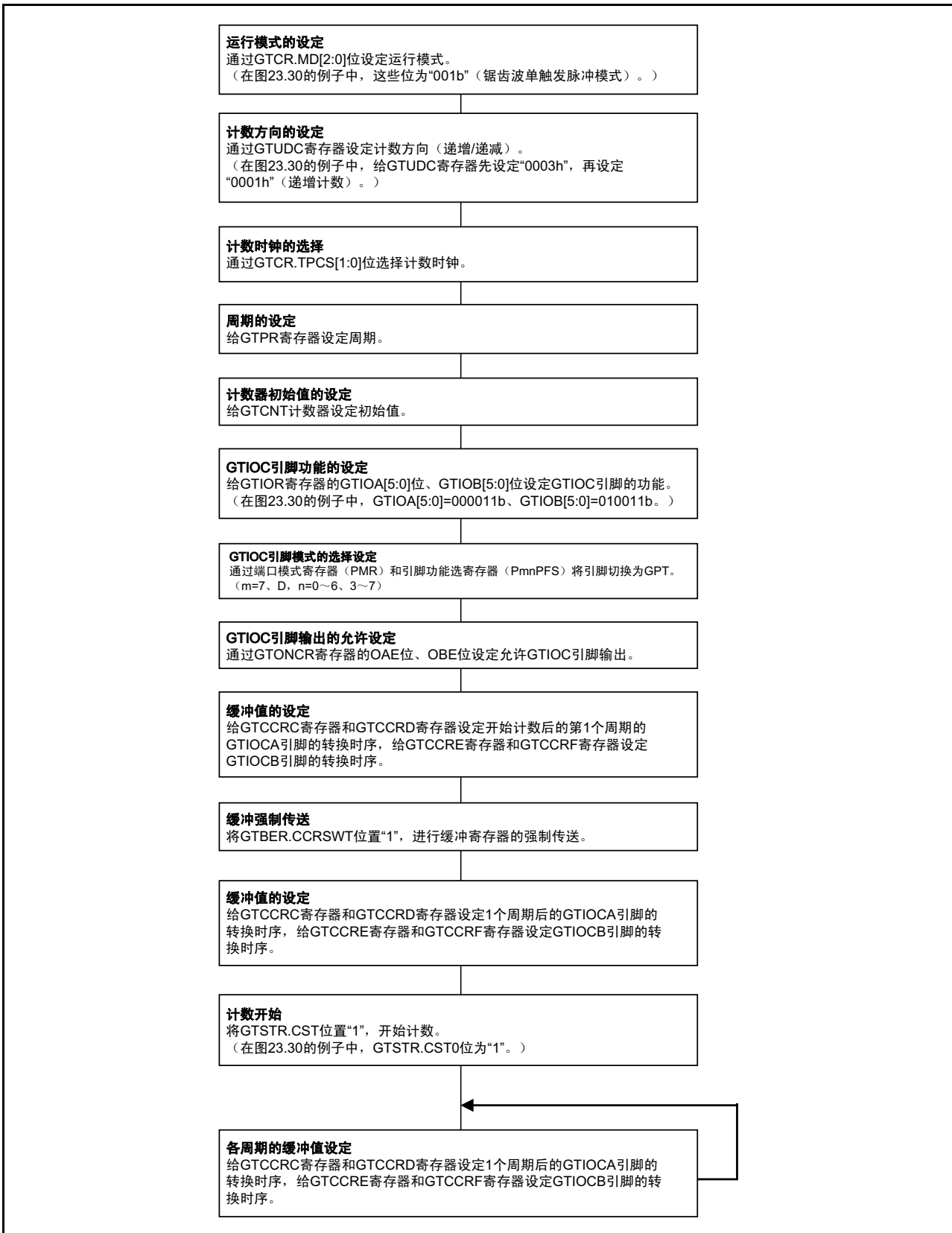


图 23.31 锯齿波单触发脉冲模式的设定例子

(3) 三角波 PWM 模式 1 (在波谷进行 16 位传送)

三角波 PWM 模式 1 是将周期设定到 GPTn.GTPR 寄存器，使 GPTn.GTCNT 进行三角波 (全波) 运行，并且通过 GPTn.GTCCRA 寄存器、GPTn.GTCCRB 寄存器的比较匹配，将 PWM 波形输出到 GTIOCnA 引脚和 GTIOCnB 引脚的模式 (n: 通道号)。缓冲运行的时序为波谷。能通过 GTIOR 寄存器将引脚的输出值设定为在比较匹配时进行 Low 电平输出 /High 电平输出 / 交替输出，在周期结束时进行 Low 电平输出 /High 电平输出 / 交替输出。

能通过设定 GTDTCR 寄存器、GTDVU 寄存器和 GTDVD 寄存器，将用于带死区时间的反相波形的比较匹配值自动设定到 GTCCRB 寄存器。

三角波 PWM 模式 1 的运行例子如图 23.32，三角波 PWM 模式 1 的设定例子如图 23.33 所示。

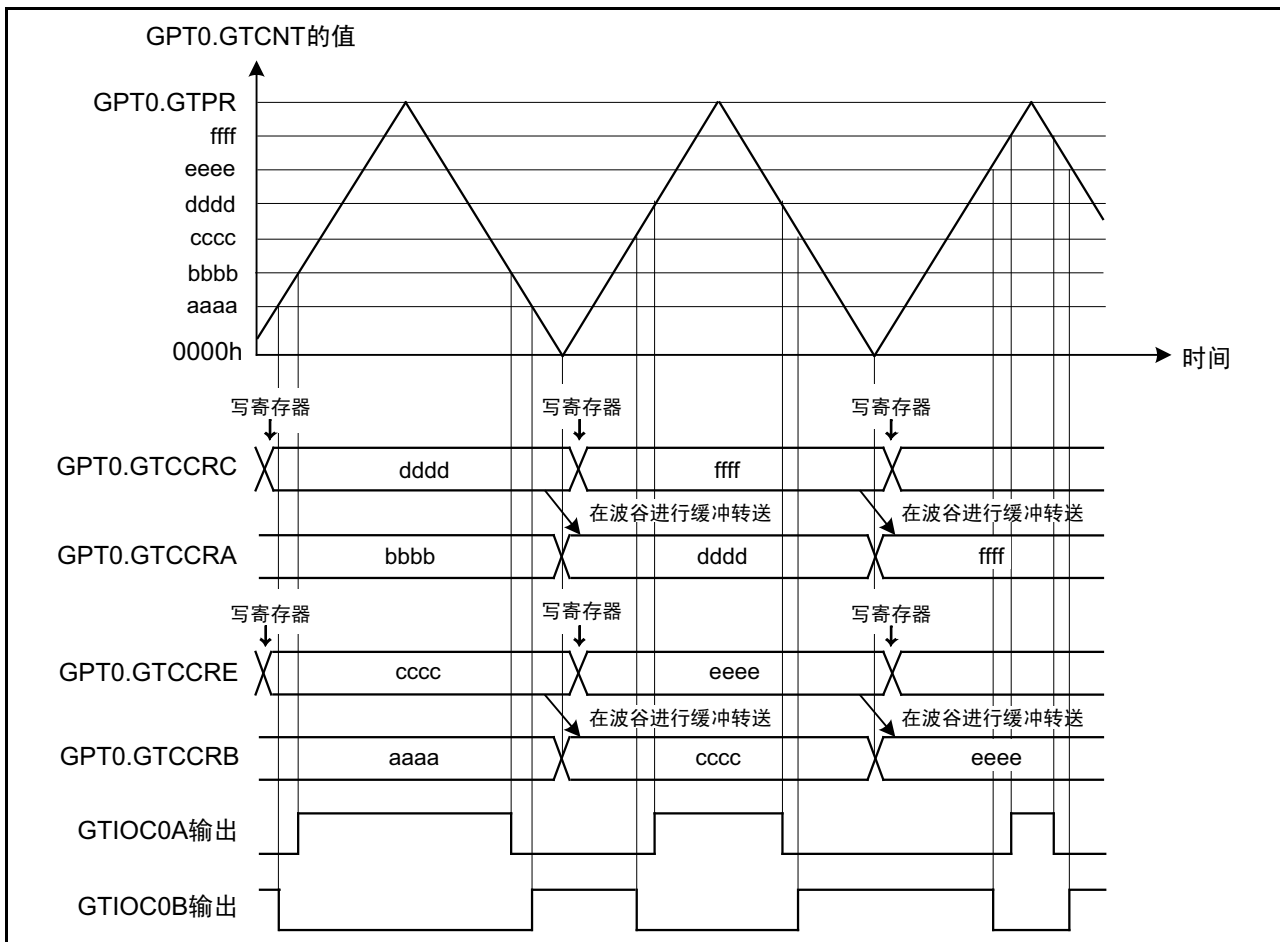


图 23.32 三角波 PWM 模式 1 的运行例子

(缓冲运行、在计数开始时 GTIOC0A 引脚为 Low 电平输出 /GTIOC0B 引脚为 High 电平输出、在 GTCCRA/B 比较匹配时进行交替输出、在周期结束时保持输出的情况)

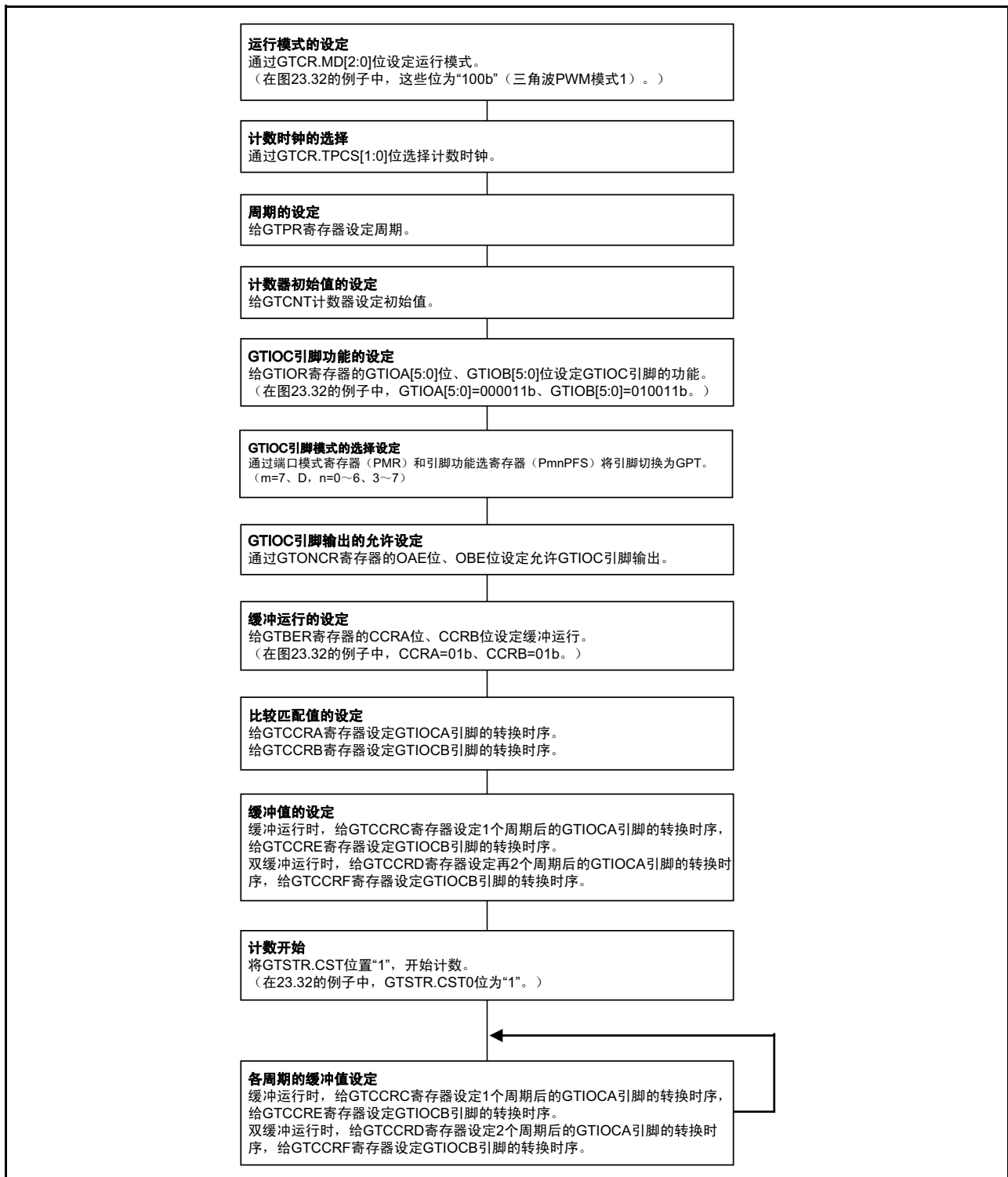


图 23.33 三角波 PWM 模式 1 的设定例子

(4) 三角波 PWM 模式 2 (在波峰和波谷进行 16 位传送)

三角波 PWM 模式 2 与三角波 PWM 模式 1 同样，是将周期设定到 GPTn.GTPR 寄存器，使 GPTn.GTCNT 进行三角波 (全波) 运行，并且通过 GPTn.GTCCRA 寄存器、GPTn.GTCCRB 寄存器的比较匹配，将 PWM 波形输出到 GTIOCnA 引脚和 GTIOCnB 引脚的模式 (n: 通道号)。但是，缓冲运行的时序为波谷和波谷。能通过 GTIOR 寄存器将引脚的输出值设定为在比较匹配时进行 Low 电平输出 /High 电平输出 / 交替输出，在周期结束时进行 Low 电平输出 /High 电平输出 / 交替输出。

能通过设定 GTDTCR 寄存器、GTDVU 寄存器和 GTDVD 寄存器，将用于带死区时间的反相波形的比较匹配值自动设定到 GTCCRB 寄存器。

三角波 PWM 模式 2 的运行例子如图 23.34，三角波 PWM 模式 2 的设定例子如图 23.35 所示。

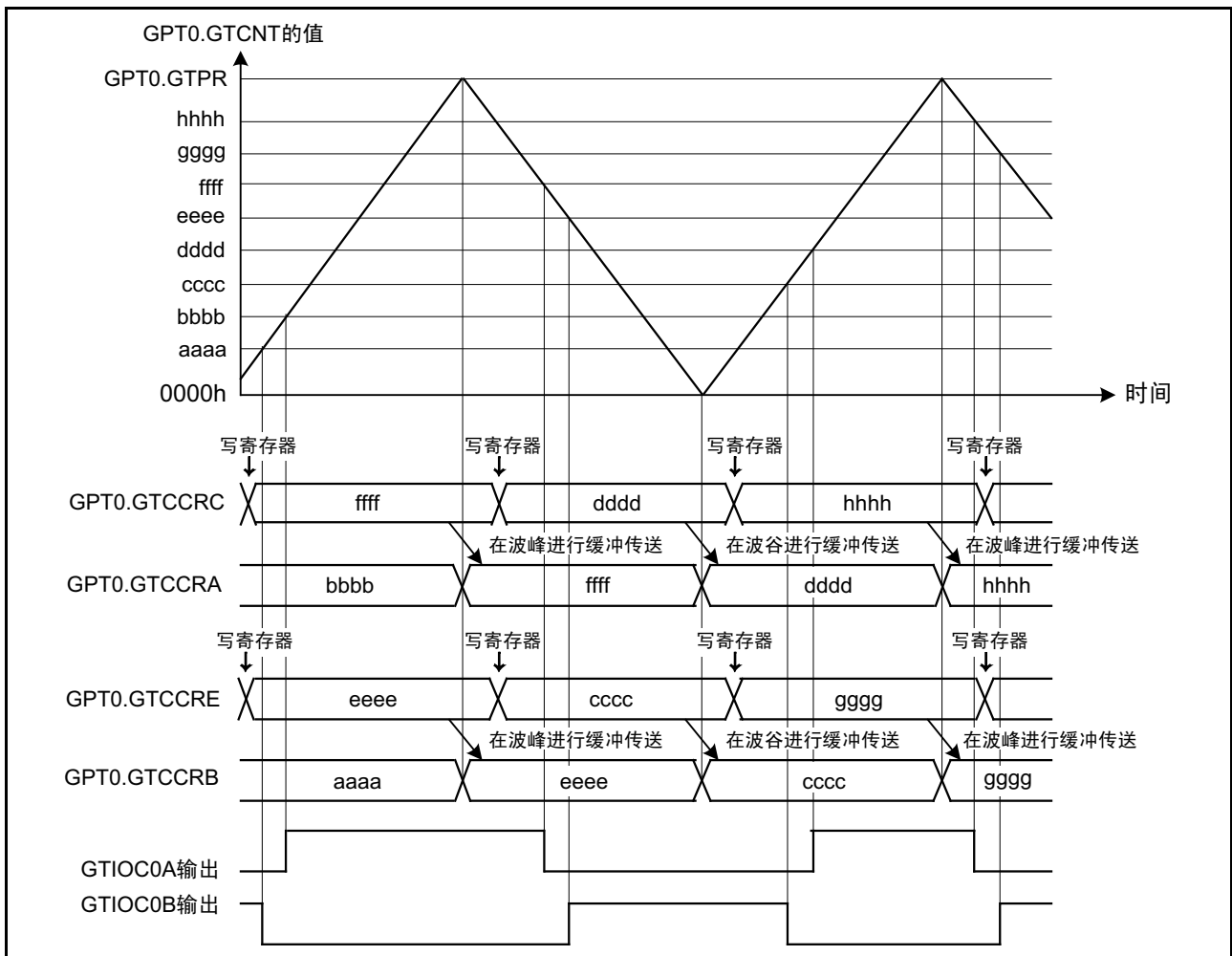


图 23.34 三角波 PWM 模式 2 的运行例子
 (缓冲运行、在计数开始时 GTIOC0A 引脚为 Low 电平输出 /GTIOC0B 引脚为 High 高电平输出、在 GTCCRA/B 比较匹配时进行交替输出、在周期结束时保持输出的情况)

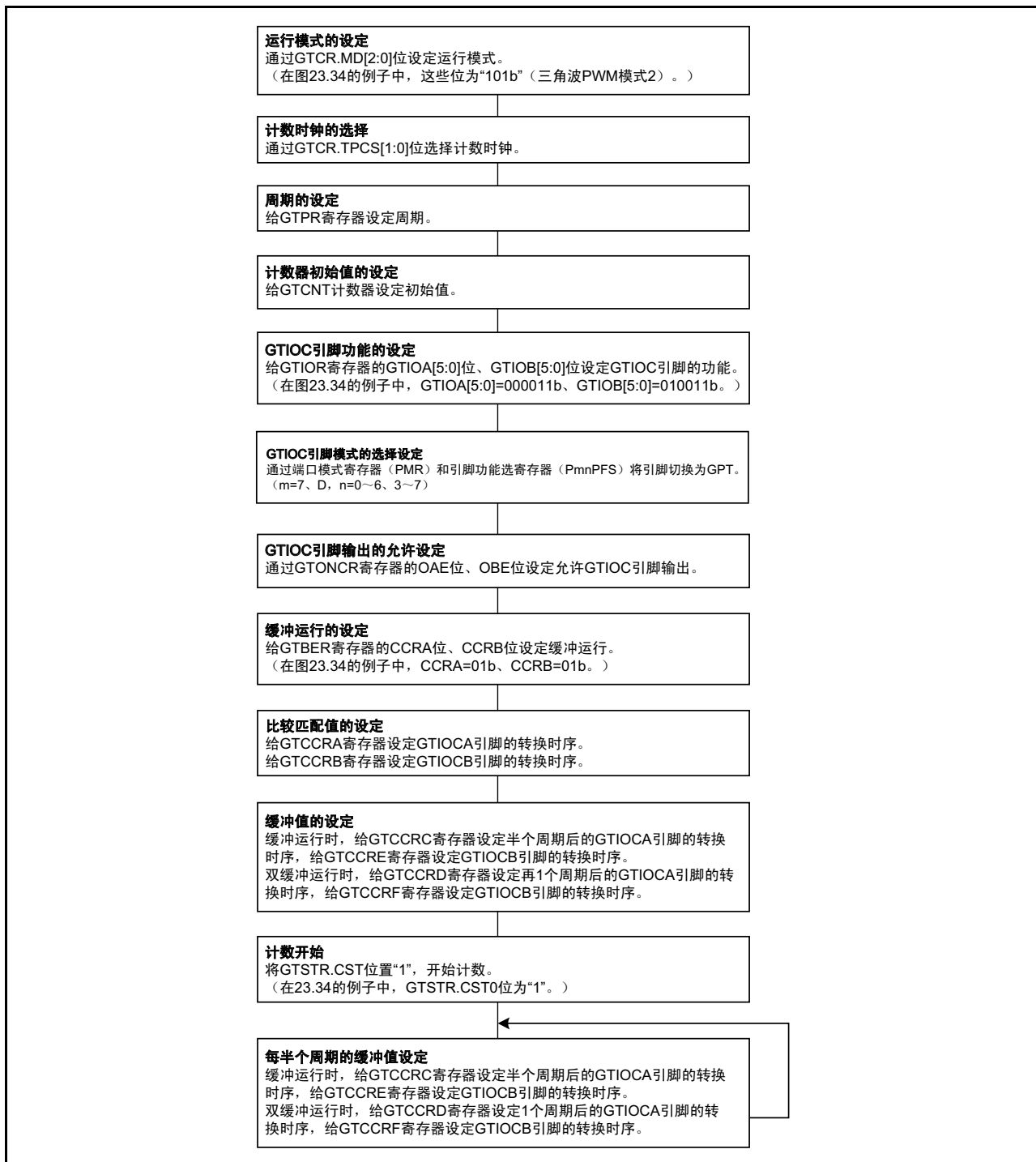


图 23.35 三角波 PWM 模式 2 的设定例子

(5) 三角波 PWM 模式 3 (在波谷进行 32 位传送)

三角波 PWM 模式 3 是将周期设定到 GPTn.GTPR 寄存器, 使 GPTn.GTCNT 进行三角波 (全波) 运行, 并且固定为缓冲运行, 通过 GPTn.GTCCRA 寄存器、GPTn.GTCCRB 寄存器的比较匹配, 将 PWM 波形输出到 GTIOCnA 引脚和 GTIOCnB 引脚的模式 (n: 通道号)。三角波 PWM 模式 3 的缓冲运行与通常的缓冲运行不同, 在波谷进行从 GTCCRC 寄存器到 GTCCRA 寄存器、从 GTCCRE 寄存器到 GTCCRB 寄存器、从 GTCCRD 寄存器到暂存器 A、从 GTCCRF 寄存器到暂存器 B 的缓冲传送。而且在波峰进行从暂存器 A 到 GTCCRA 寄存器、从暂存器 B 到 GTCCRB 寄存器的缓冲传送。能通过 GTIOR 寄存器将引脚的输出值设定为在比较匹配时进行 Low 电平输出 / High 电平输出 / 交替输出, 在周期结束时进行 Low 电平输出 / High 电平输出 / 交替输出。

能通过设定 GTDTCR 寄存器、GTDVU 寄存器和 GTDVD 寄存器, 将用于带死区时间的反相波形的比较匹配值自动设定到 GTCCRB 寄存器。

三角波 PWM 模式 3 的运行例子如图 23.36, 三角波 PWM 模式 3 的设定例子如图 23.37 所示。

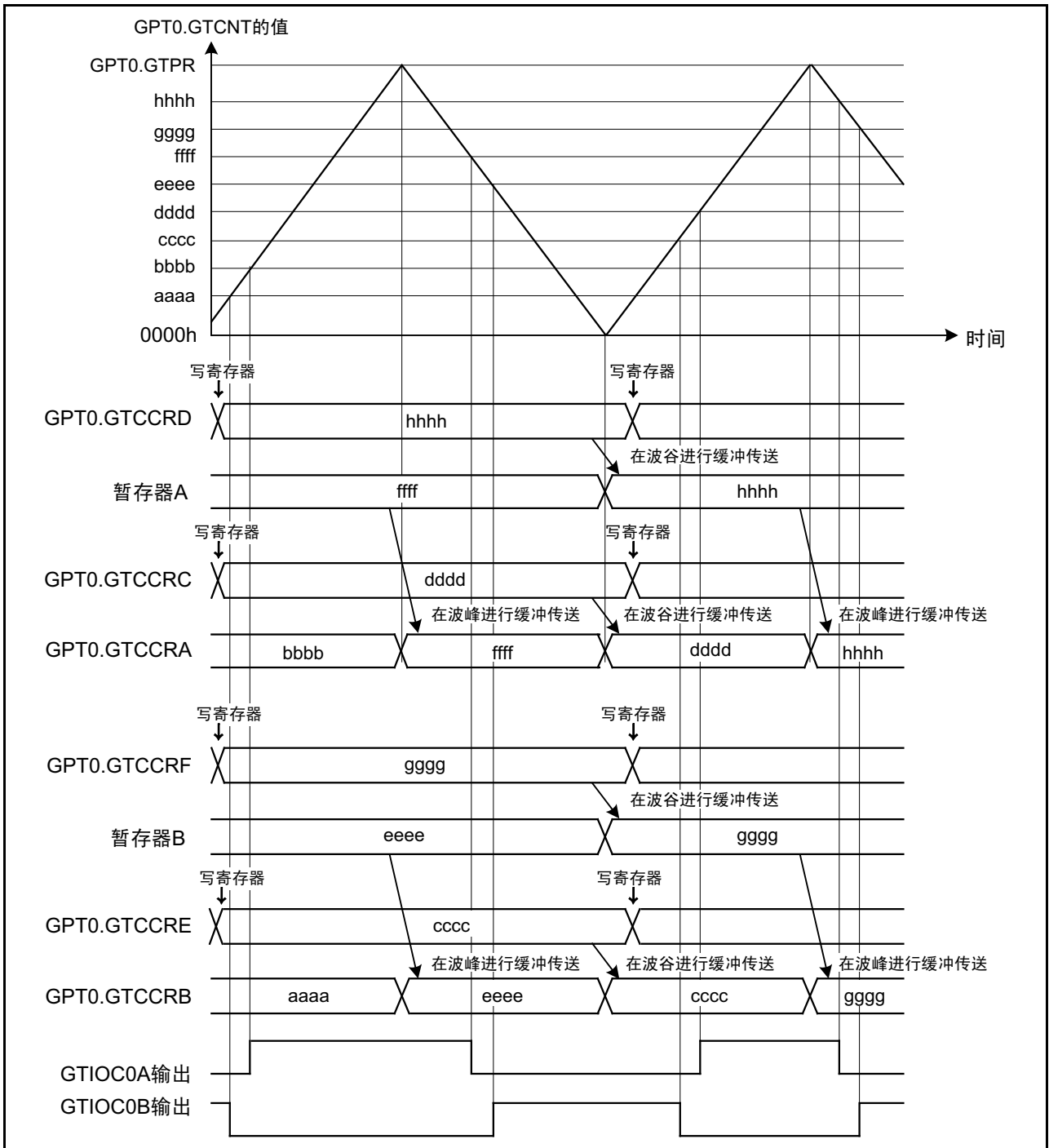


图 23.36 三角波 PWM 模式 3 的运行例子

(在计数开始时 GTIOC0A 引脚为 Low 电平输出 /GTIOC0B 引脚为 High 电平输出、在 GTCRA/ B 比较匹配时进行交替输出、在周期结束时保持输出的情况)

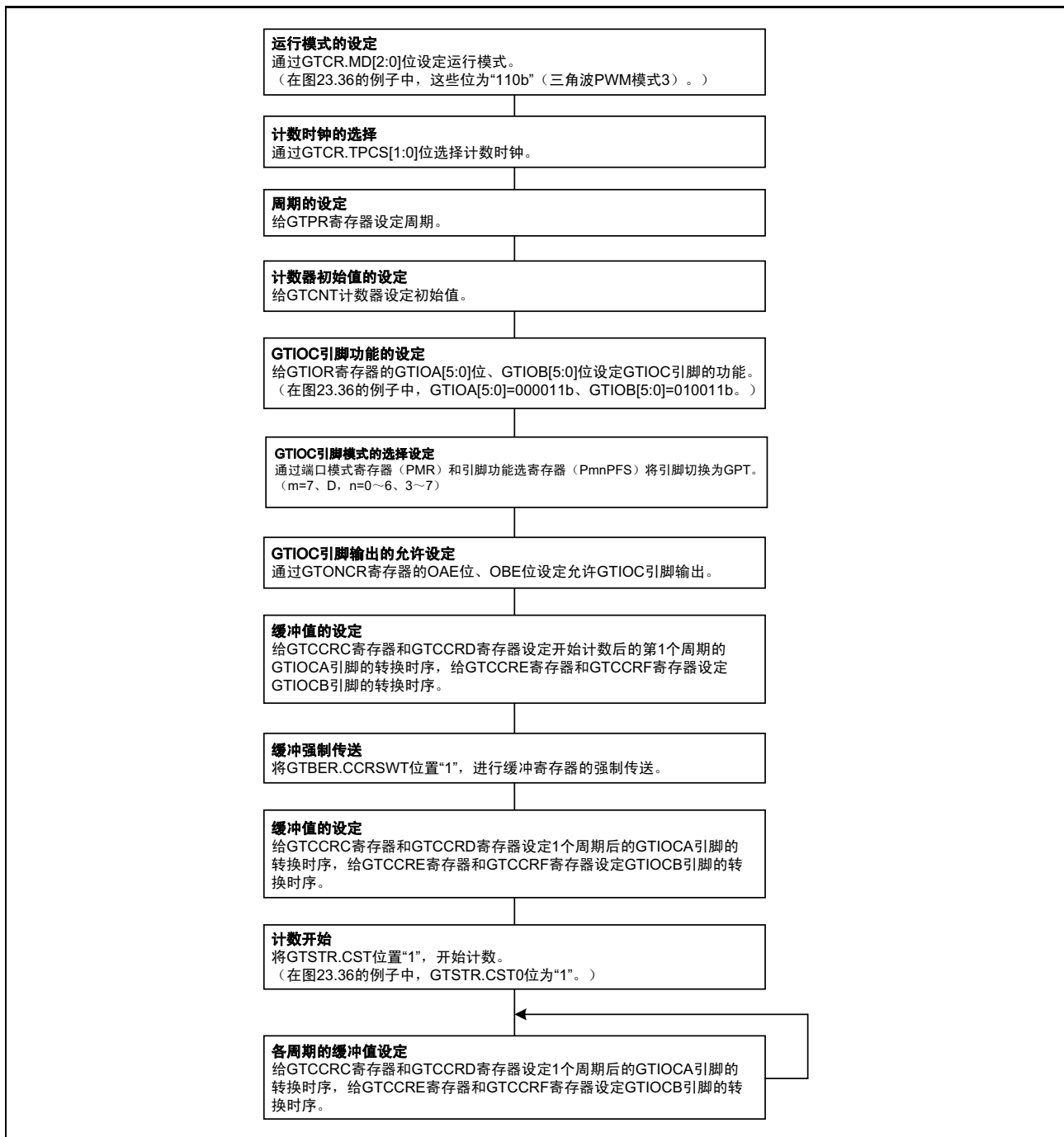


图 23.37 三角波 PWM 模式 3 的设定例子

23.3.4 死区时间自动设定功能

能通过设定 GTDTCR 寄存器，从正相波形的比较匹配值 (GTCCRA 寄存器的值) 和死区时间值 (GTDVU 寄存器和 GTDVD 寄存器的值) 生成带死区时间的反相波形的比较匹配值，并且自动设定到 GTCCRB 寄存器。

死区时间自动设定功能用于锯齿波单触发脉冲模式和全部的三角波 PWM 模式。

能在前侧和后侧独立设定死区时间。通过 GTDVU 寄存器设定对应于反相波形前侧转换时序的死区时间，通过 GTDVD 寄存器设定对应于反相波形后侧转换时序的死区时间。也能在前侧和后侧使用共同的死区时间。

另外，GTDBU 寄存器能用作 GTDVU 寄存器的缓冲寄存器。同样，GTDBD 寄存器能用作 GTDVD 寄存器的缓冲寄存器。缓冲传送时序为周期结束 (锯齿波时 GTCNT 为发生上溢 (递增计数) 或者下溢 (递减计数) 时，三角波时为波谷) 时。

在使用死区时间自动设定功能时，禁止写 GTCCRB 寄存器，也禁止设定超出周期的死区时间。能通过读 GTCCRB 寄存器的值确认死区时间的自动设定值。

死区时间自动设定功能的运行例子如图 23.38 ~ 图 23.40，死区时间自动设定功能的设定例子如图 23.41、图 23.42 所示。

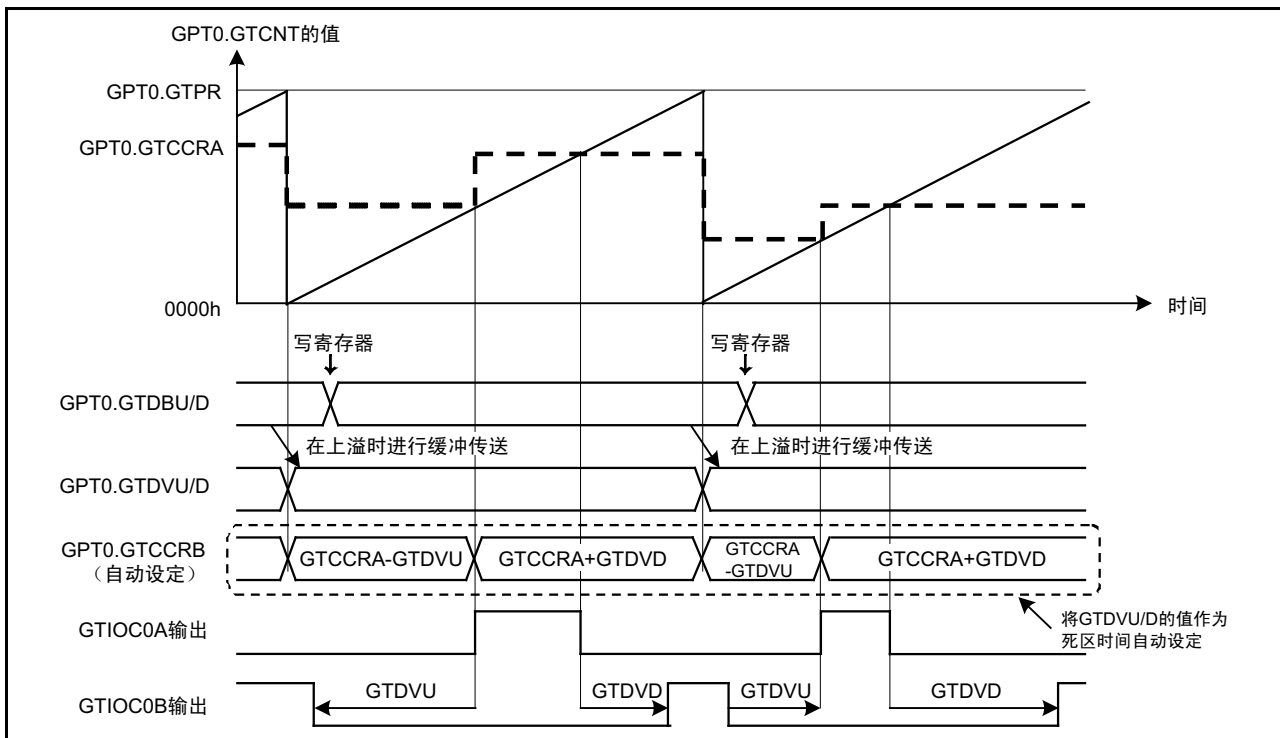


图 23.38 死区时间自动设定功能的运行例子
(锯齿波单触发脉冲模式、GTDVU/D 寄存器进行缓冲运行、有效电平为 High 电平的情况)

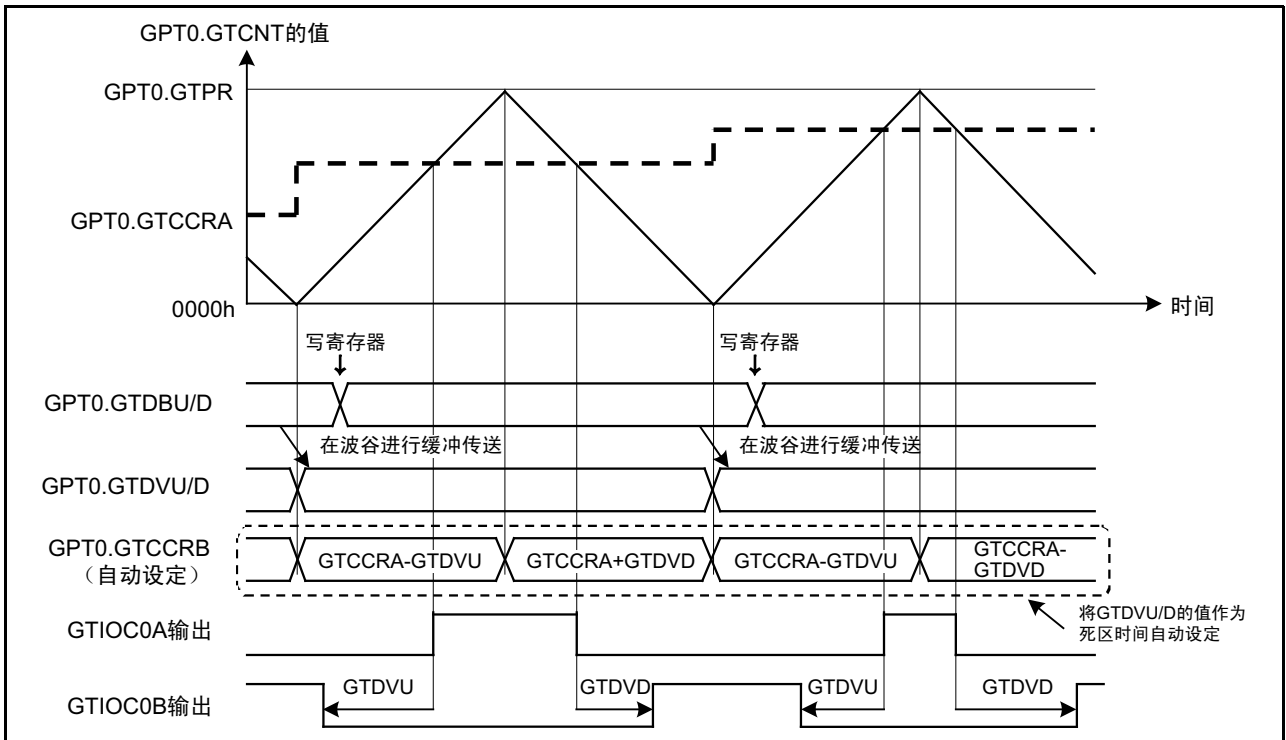


图 23.39 带死区时间自动设定功能的运行例子
(三角波 PWM 模式 1、GTDVU/D 寄存器进行缓冲运行、有效电平为 High 电平的情况)

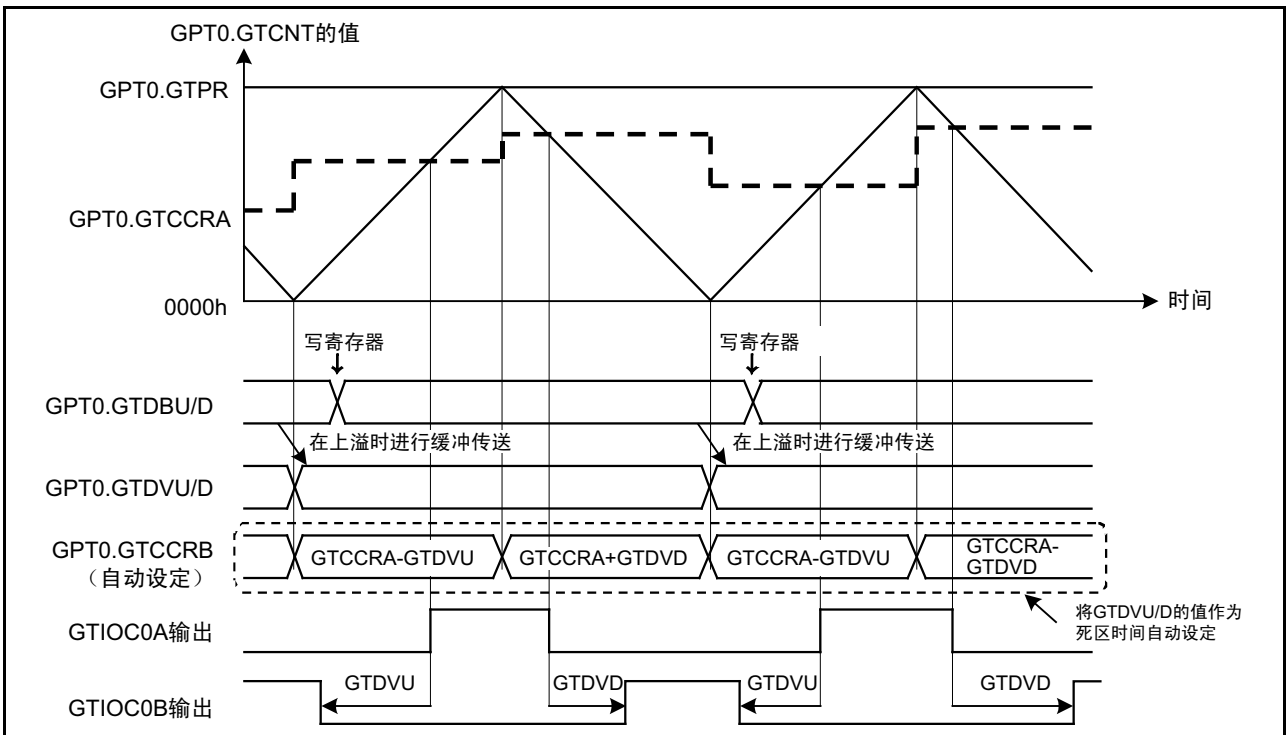


图 23.40 死区时间自动设定功能的运行例子
(三角波 PWM 模式 2 和 3、GTDVU/D 寄存器进行缓冲运行、有效电平为 High 电平的情况)

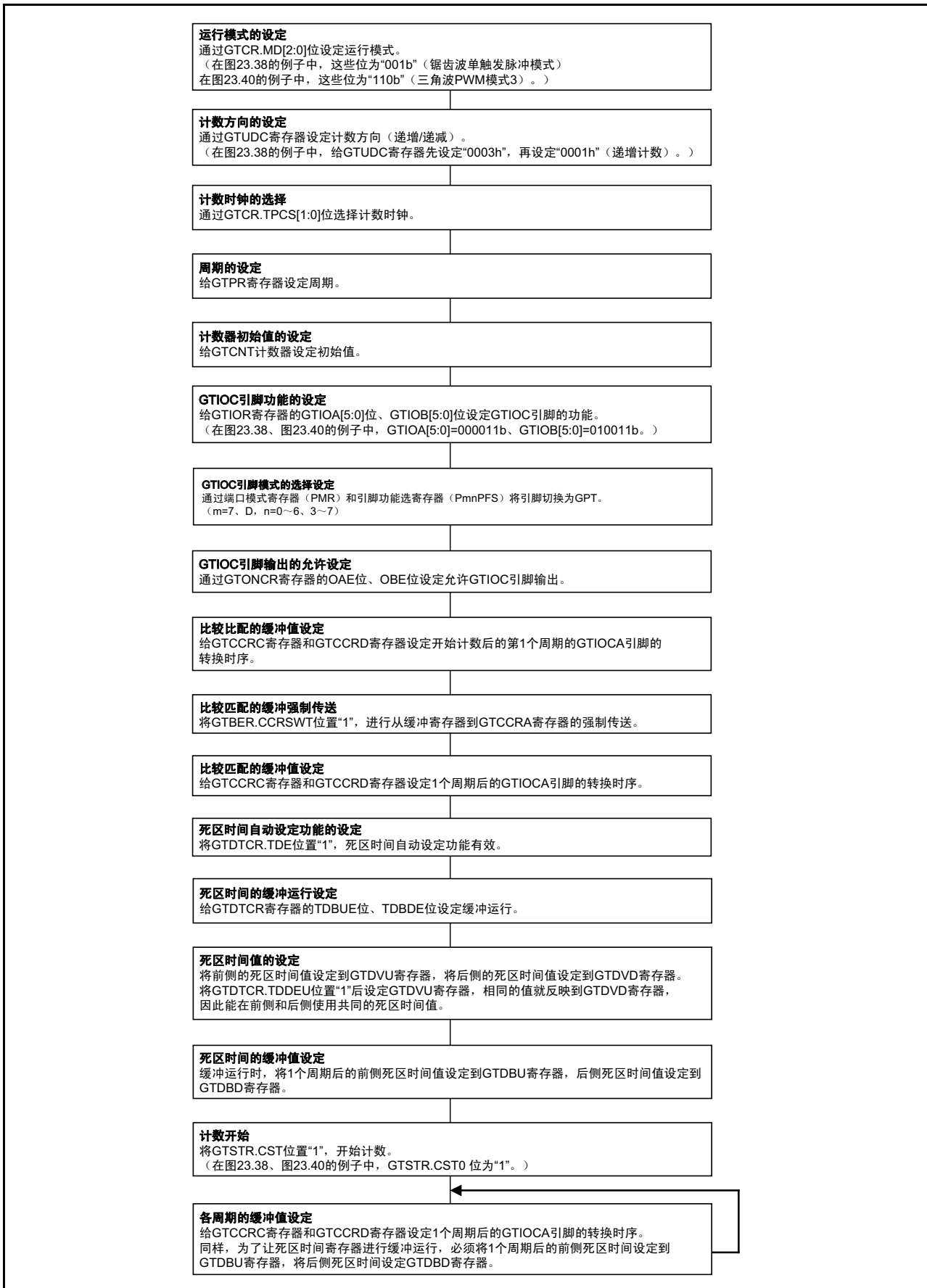


图 23.41 死区时间自动设定功能的设定例子 (锯齿波单触发脉冲模式、三角波 PWM 模式 3 的情况)

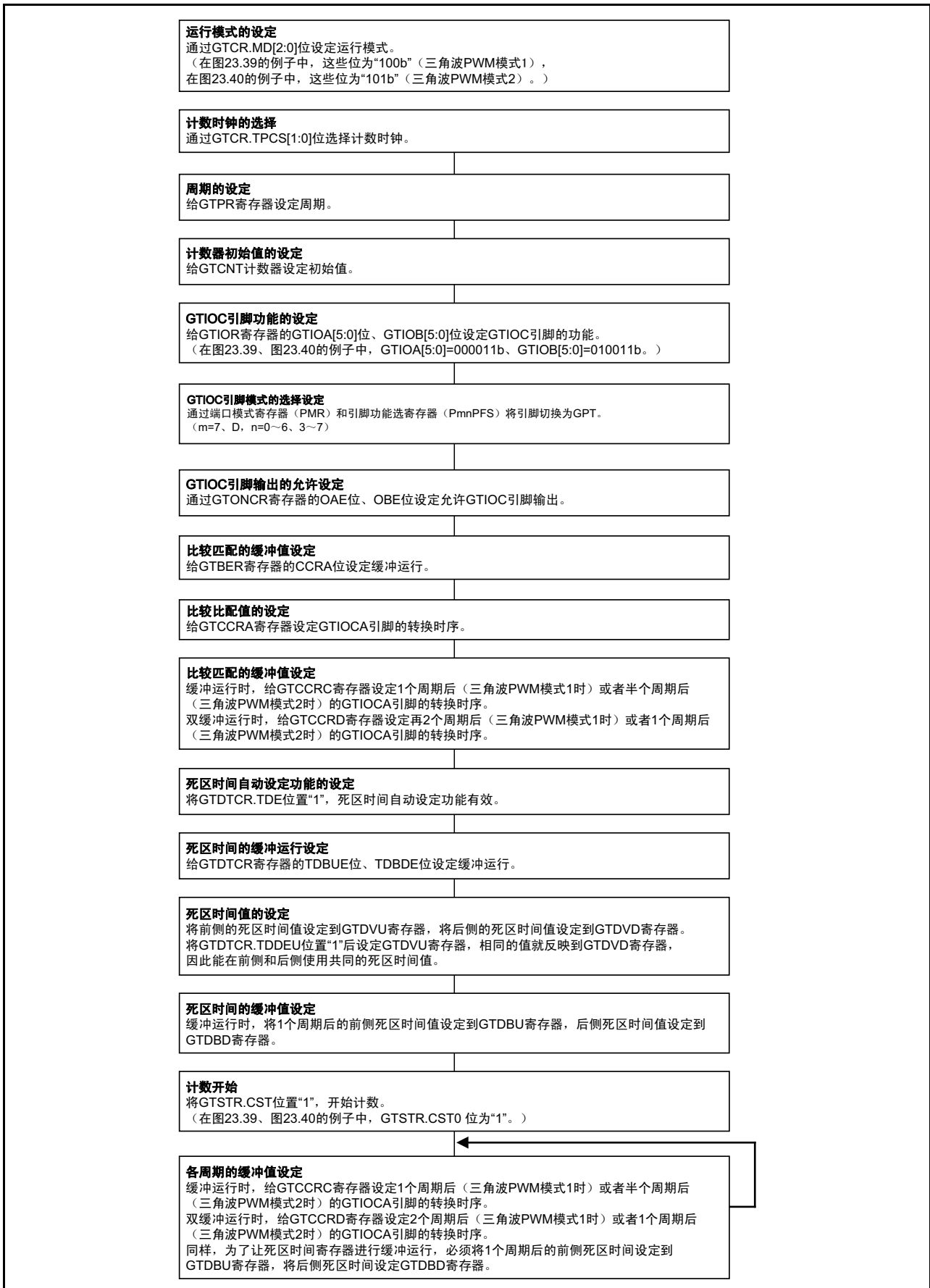


图 23.42 死区时间自动设定功能的设定例子 (三角波 PWM 模式 1 和 2 的情况)

23.3.5 计数方向转换功能

能通过更改 GTUDC.UD 位的值，转换 GTCNT 计数器的计数方向。

为锯齿波的情况时，如果在计数时更改 GTUDC.UD 位的值，就在发生上溢（在递增计数时更改）或者下溢（递减计数时更改）时转换计数方向。如果在停止计数并且 GTUDC.UDF 位为“0”的状态下更改 GTUDC.UD 位的值，计数开始后不反映此值，但是在发生上溢或者下溢时转换计数方向。如果在停止计数时将 GTUDC.UDF 位置“1”，就从计数开始后反映此时的 GTUDC.UD 位的值。

为三角波的情况时，即使在计数时更改 GTUDC.UD 位的值也不转换计数方向。同样，即使在停止计数并且 GTUDC.UDF 位为“0”的状态下更改 GTUDC.UD 位的值也不反映此值。如果在停止计数时将 GTUDC.UDF 位置“1”，就从计数开始后反映此时的 GTUDC.UD 位的值。

在锯齿波的计数过程中转换计数方向时，如果是递增计数，就将递增计数开始后的 GTPR 寄存器值反映到计数周期，如果是递减计数，就将递减计数开始前的 GTPR 寄存器值反映到计数周期。

计数方向转换功能的运行例子如图 23.43 所示。

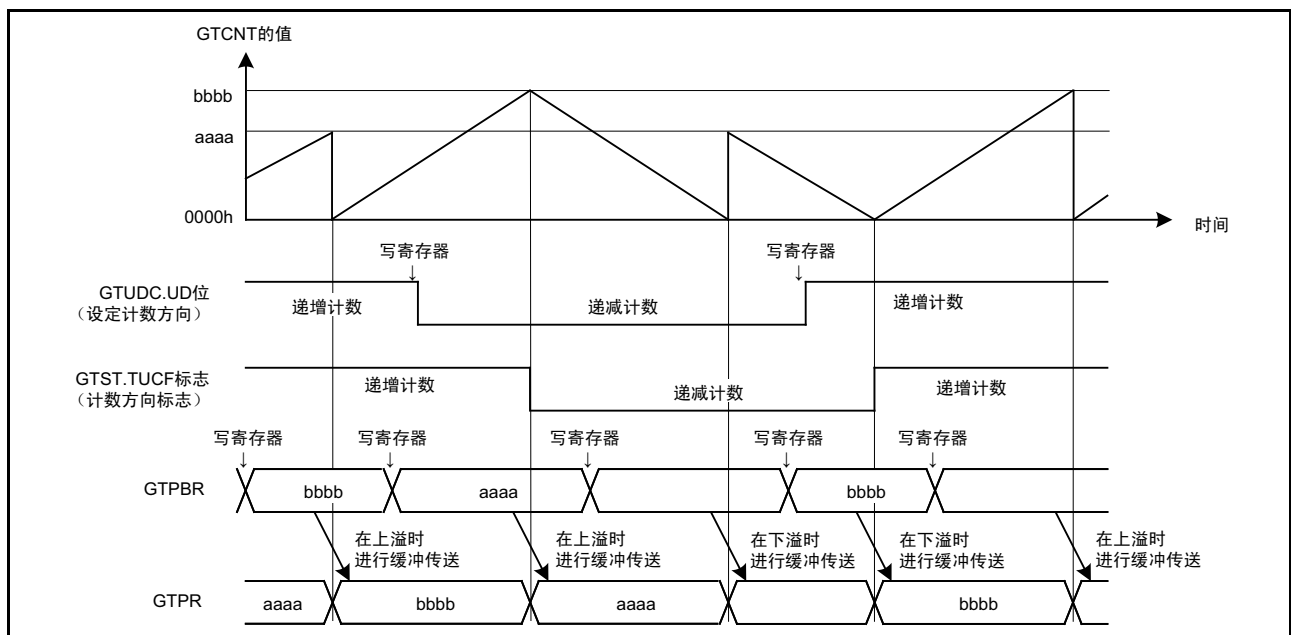


图 23.43 计数方向转换功能的运行例子（缓冲运行时）

23.3.6 硬件启动 / 停止、清除运行

能通过 RX63T 群内置的硬件源对 GTCNT 计数器的启动、停止以及清除进行控制。

硬件源有 4 种，分别是 GTETRГ 引脚输入、比较器检测、GTIOC3A 和 GTIOC3B 的引脚输入以及 GTIOC3A 和 GTIOC3B 的内部输出（输出比较）。另外，还能通过 GTCCRA 寄存器和 GTCCRB 寄存器的输入捕捉进行计数器清除。

23.3.6.1 硬件启动运行

能通过硬件源控制 GTCNT 计数器的启动。通过 GTHSSR.CSHSL 位选择让计数器启动的硬件源，通过 GTHSCR.CSHW 位指定硬件源的转换边沿并且允许计数。

通过硬件源启动计数的运行例子如图 23.44 所示，通过硬件源启动计数运行的设定例子如图 23.45 所示。

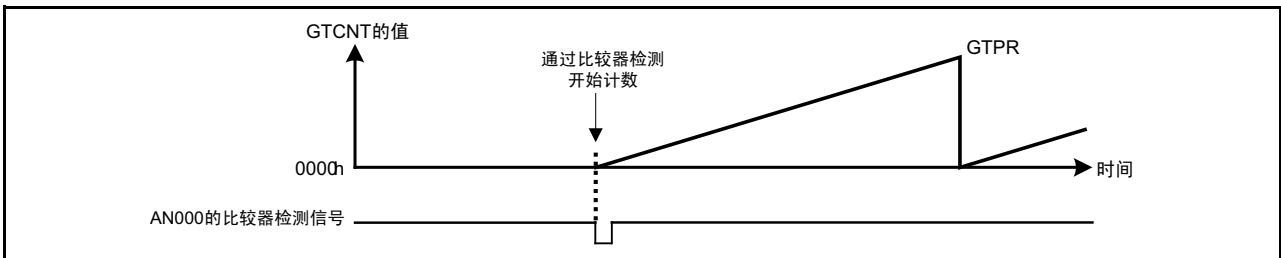


图 23.44 通过硬件源启动计数的运行例子（通过 AN000 的比较器检测开始计数的情况）

<p>运行模式的设定 通过 GTCR.MD[2:0] 位设定运行模式。 (在图 23.44 的例子中，这些位为“000b”（锯齿波 PWM 模式）。)</p>
<p>计数方向的设定 通过 GTUDC 寄存器设定计数方向（递增/递减）。 (在图 23.44 的例子中，给 GTUDC 寄存器先设定“0003h”，再设定“0001h”（递增计数）。)</p>
<p>计数时钟的选择 通过 GTCR.TPCS[1:0] 位选择计数时钟。</p>
<p>周期的设定 给 GTPR 寄存器设定周期。</p>
<p>计数器初始值的设定 给 GTCNT 计数器设定初始值。 (在图 23.44 的例子中，初始值为“0000h”。)</p>
<p>硬件计数的开始设定 通过 GTHSSR.CSHSL 位选择让计数开始的硬件源。 通过 GTHSCR.CSHW 位指定硬件源的边沿，允许计数，进入等待通过硬件源开始计数的状态。 (在图 23.44 的例子中，GTHSSR.CSHSL_n = 0000b、GTHSCR.CSHW_n = 11b (n=0、1、2、3)。)</p>
<p>硬件源的运行设定 设定 GTHSSR.CSHSL 位选择的硬件源的运行，开始计数。 (在图 23.44 的例子中，设定为 AN000 的比较器运行。)</p>

图 23.45 通过硬件源启动计数运行的设定例子

23.3.6.2 硬件停止运行

能通过硬件源控制 GTCNT 计数器的停止。通过 GTHPSR.CSHPL 位选择让计数器停止的硬件源，通过 GTHSCR.CPHW 位指定硬件源的转换边沿并且允许停止计数。

通过硬件源停止计数的运行例子如图 23.46 所示，通过硬件源停止计数运行的设定例子如图 23.47 所示。此例是通过 GTIOC3A 内部输出（输出比较）的双边沿停止计数，并且通过 GTIOC3B 内部输出（输出比较）的双边沿重新开始计数的例子。

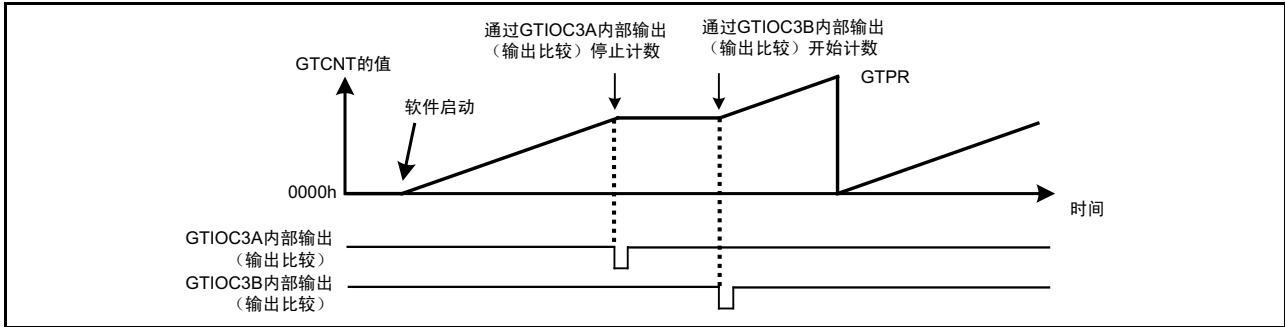


图 23.46 通过硬件源停止计数的运行例子
(通过软件启动、通过 GTIOC3A 内部输出（输出比较）停止、通过 GTIOC3B 内部输出（输出比较）重新启动)

<p>运行模式的设定 通过GTCR.MD[2:0]位设定运行模式。 (在图23.46的例子中, 这些位为“000b”(锯齿波PWM模式)。)</p>
<p>计数方向的设定 通过GTUDC寄存器设定计数方向(递增/递减)。 (在图23.46的例子中, 给GTUDC寄存器先设定“0003h”, 再设定“0001h”(递增计数)。)</p>
<p>计数时钟的选择 通过GTCR.TPCS[1:0]位选择计数时钟。</p>
<p>周期的设定 给GTPR寄存器设定周期。</p>
<p>计数器初始值的设定 给GTCNT计数器设定初始值。 (在图23.46的例子中, 初始值为“0000h”。)</p>
<p>硬件计数的开始设定 通过GTHSSR.CSHSL位选择让计数开始的硬件源。 通过GTHSCR.CSHW位指定硬件源的边沿, 允许计数, 进入等待通过硬件源开始计数的状态。 (在图23.46的例子中, GTHSSR.CSHSLn= 1010b、GTHSCR.CSHWn= 11b (n=0~2)。)</p>
<p>硬件计数的停止设定 通过GTHPSR.CSHPL位选择让计数停止的硬件源。 通过GTHSCR.CPHW位指定硬件源的边沿, 允许停止计数, 进入等待通过硬件源停止计数的状态。 (在图23.46的例子中, GTHPSR.CSHPLn= 1011b、GTHSCR.CPHWn= 11b (n=0~2)。)</p>
<p>硬件源的运行设定 设定GTHSSR.CSHSL位、GTHPSR.CSHPL位选择的硬件源的运行, 开始或者停止计数。 (在图23.46的例子中, 设定GPT3.GTCCRA比较匹配、GPT3.GTCCRB比较匹配和GPT3.GTIOR位, 之后将GTSTR.CST位置“1”, 开始计数。)</p>

图 23.47 通过硬件源停止计数运行的设定例子

通过硬件源开始或者停止计数的运行例子如图 23.48 所示。通过硬件源开始或者停止计数运行的设定例子如图 23.49 所示。在此例子中，外部输入触发 GTETRG 在 High 电平区间进行计数。

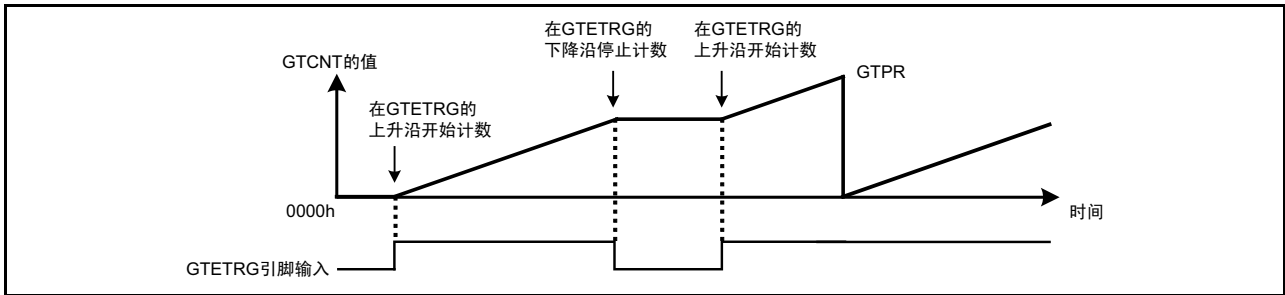


图 23.48 通过硬件源开始或者停止计数的运行例子
(在 GTETRG 引脚输入的上升沿开始计数，在 GTETRG 引脚输入的下沿停止计数)

<p>运行模式的设定 通过GTCR.MD[2:0]位设定运行模式。 (在图23.48的例子中，这些位为“000b”(锯齿波PWM模式)。)</p>
<p>计数方向的设定 通过GTUDC寄存器设定计数方向(递增/递减)。 (在图23.48的例子中，给GTUDC寄存器先设定“0003h”，再设定“0001h”(递增计数)。)</p>
<p>计数时钟的选择 通过GTCR.TPCS[1:0]位选择计数时钟。</p>
<p>周期的设定 给GTPR寄存器设定周期。</p>
<p>计数器初始值的设定 给GTCNT计数器设定初始值。 (在图23.48的例子中，初始值为“0000h”。)</p>
<p>硬件计数的开始设定 通过GTHSSR.CSHSL位选择让计数开始的硬件源。 通过GTHSCR.CSHW位指定硬件源的边沿，允许计数，进入等待通过硬件源开始计数的状态。 (在图23.48的例子中，GTHSSR.CSHSLn= 1100b、GTHSCR.CSHWn= 01b (n=0~3)。)</p>
<p>硬件计数的停止设定 通过GTHPSR.CSHPL位选择让计数停止的硬件源。 通过GTHSCR.CPHW位指定硬件源的边沿，允许停止计数，进入等待通过硬件源停止计数的状态。 (在图23.48的例子中，GTHPSR.CSHPLn= 1100b、GTHSCR.CPHWn= 10b (n=0~3)。)</p>
<p>硬件源的运行设定 设定GTHSSR.CSHSL位、GTHPSR.CSHPL位选择的硬件源的运行，开始或者停止计数。 (在图23.48的例子中，设定GTETRG引脚的运行。)</p>

图 23.49 通过硬件源开始或者停止计数运行的设定例子

23.3.6.3 硬件清除运行

能通过硬件源控制 GTCNT 计数器的清除。通过 GTHPSR.CSHPL 位选择清除计数器的硬件源，通过 GTHCCR.CCHW 位指定硬件源的转换边沿并且允许清除计数器。

另外，通过设定 GTCR.CCLR[1:0] 位，还能由 GTCCRA 寄存器和 GTCCRB 寄存器的输入捕捉进行计数器清除。

即使发生通过硬件源清除计数器和通过软件源清除计数器，也不发生 GTCIV 中断（上溢/下溢中断）。

通过硬件源进行计数器清除的运行例子如图 23.50、图 23.51 所示，通过硬件源进行计数器清除运行的设定例子如图 23.52 所示。在此例子中，通过 GTIOC3A 引脚输入的双边沿开始计数，并且通过 GTIOC3B 引脚输入的双边沿停止或者清除计数的例子。

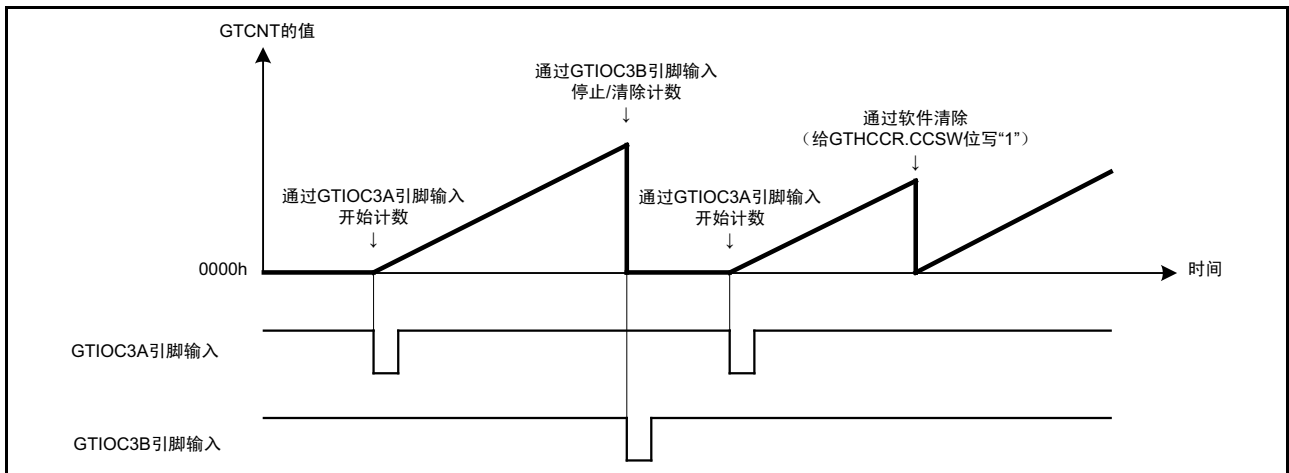


图 23.50 通过硬件源进行计数器清除的运行例子
(在锯齿波进行递增计数、通过 GTIOC3A 引脚输入开始计数、通过 GTIOC3B 引脚输入停止或者清除计数)

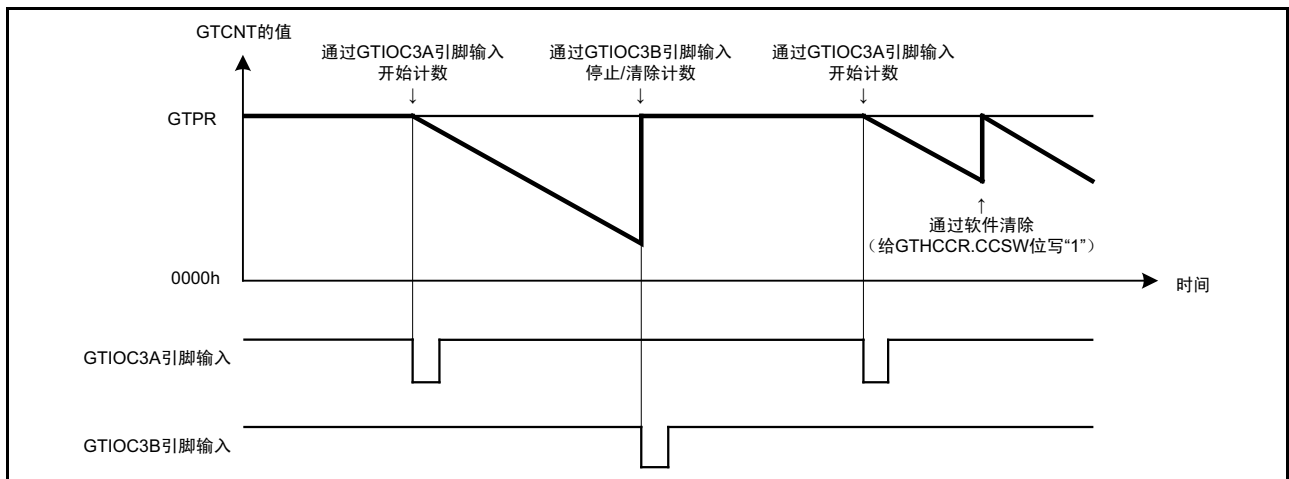


图 23.51 通过硬件源进行计数器清除的运行例子
(在锯齿波进行递减计数、通过 GTIOC3A 引脚输入开始计数、通过 GTIOC3B 引脚输入停止或者清除计数)

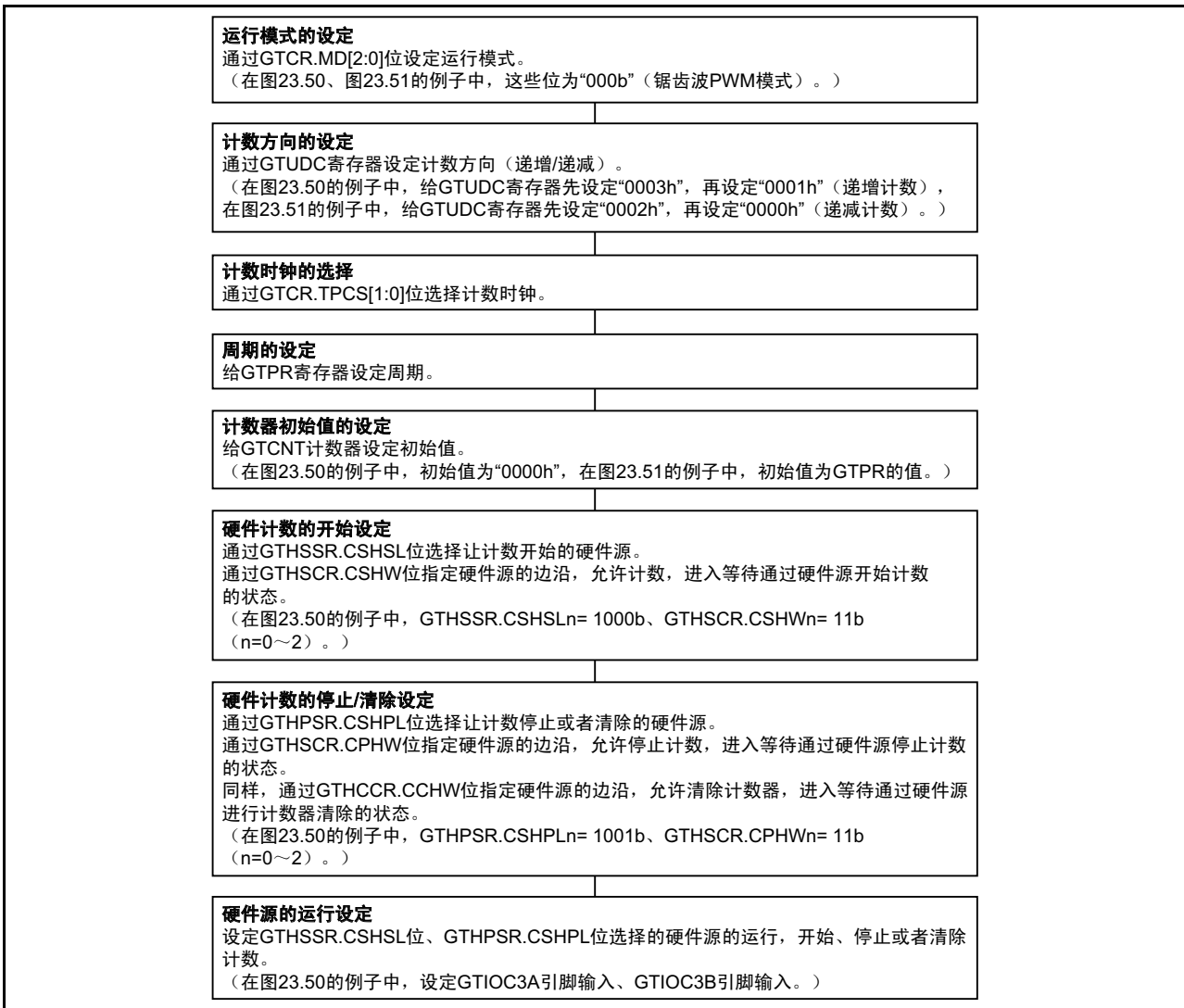


图 23.52 通过硬件源进行计数器清除运行的设定例子

即使通过硬件源进行计数器清除, 也不发生 GTCIV 中断 (上溢 / 下溢中断)。同样, 即使通过软件进行计数器清除, 也不发生 GTCIV 中断。

通过硬件进行计数器清除和 GTCIV 中断的关系如图 23.53 所示。

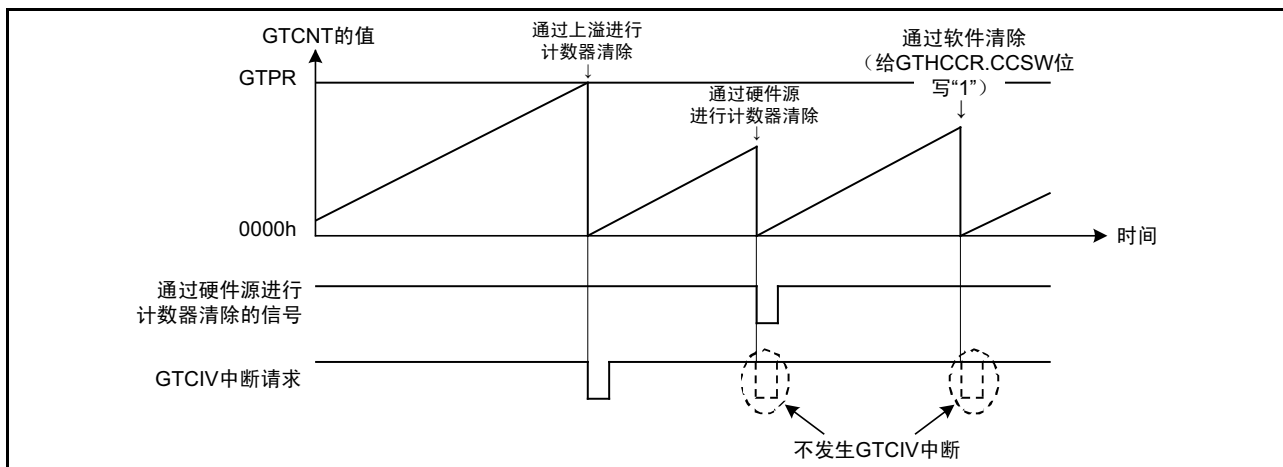


图 23.53 通过硬件进行计数器清除和 GTCIV 中断的关系

23.3.7 同步运行

能进行通道间的同步运行（同步清除、同步启动）。

23.3.7.1 同步清除运行

能控制通道间的同步清除。将需要进行同步清除的通道的 GTCR.CCLR[1:0] 位置“11b”，然后通过 GTSYNC.SYNCn[1:0] (n=0 ~ 3) 位设定通过哪个通道的清除源进行同步清除。

同步清除的运行例子和设定例子分别如图 23.54、图 23.56 所示。在此例子中，通过 GPT0.GTCNT 的清除源（上溢）同步清除 GPT1.GTCNT 和 GPT2.GTCNT 的例子。

不发生已被同步清除的通道间的同步清除（同步清除不被传播）。

已被同步清除的通道间的同步清除运行例子和设定例子分别如图 23.55、图 23.56 所示。在此例子中，通过 GPT0.GTCNT 的清除源（上溢）同步清除 GPT1.GTCNT，通过 GPT1.GTCNT 的清除源（上溢）同步清除 GPT2.GTCNT 的例子。通过 GPT0.GTCNT 的清除源（上溢）同步清除的 GPT1.GTCNT 的同步清除不传播到 GPT2.GTCNT。

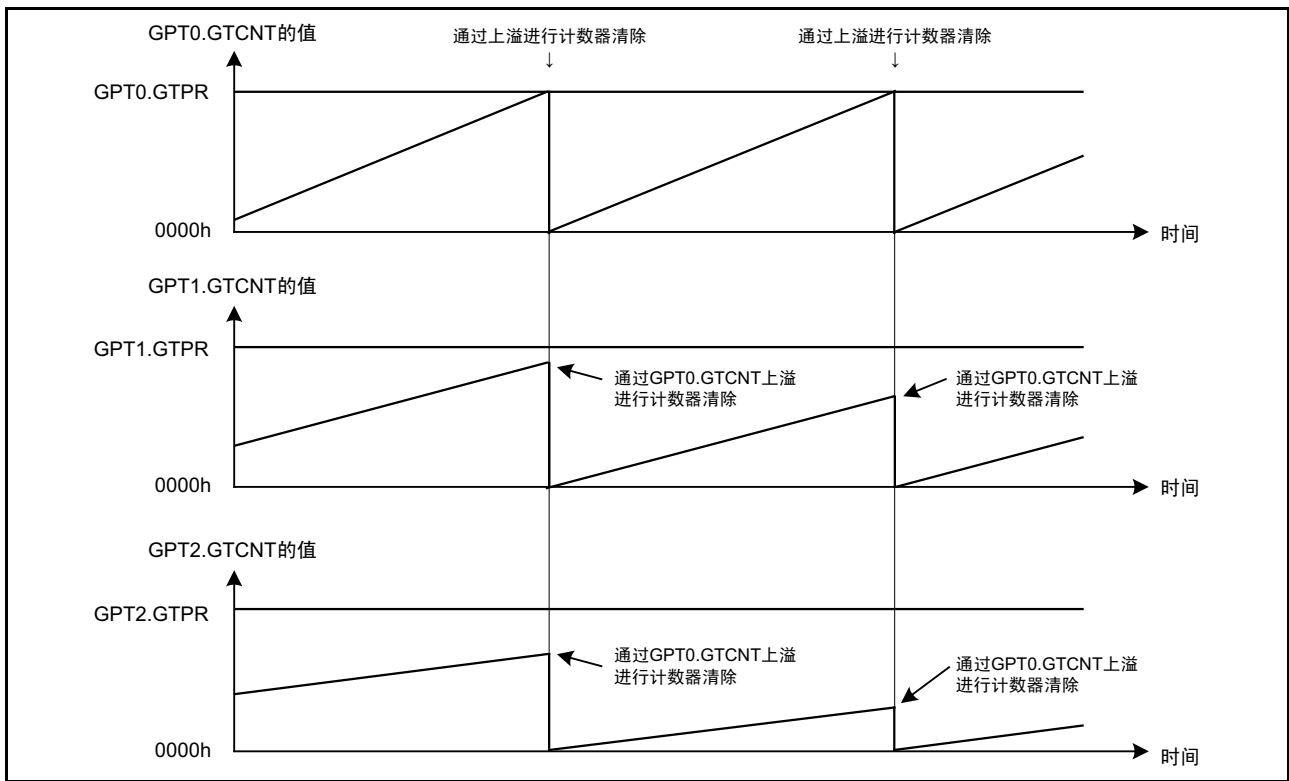


图 23.54 同步清除的运行例子（通过 GPT0.GTCNT 的清除源同步清除 GPT1.GTCNT 和 GPT2.GTCNT）

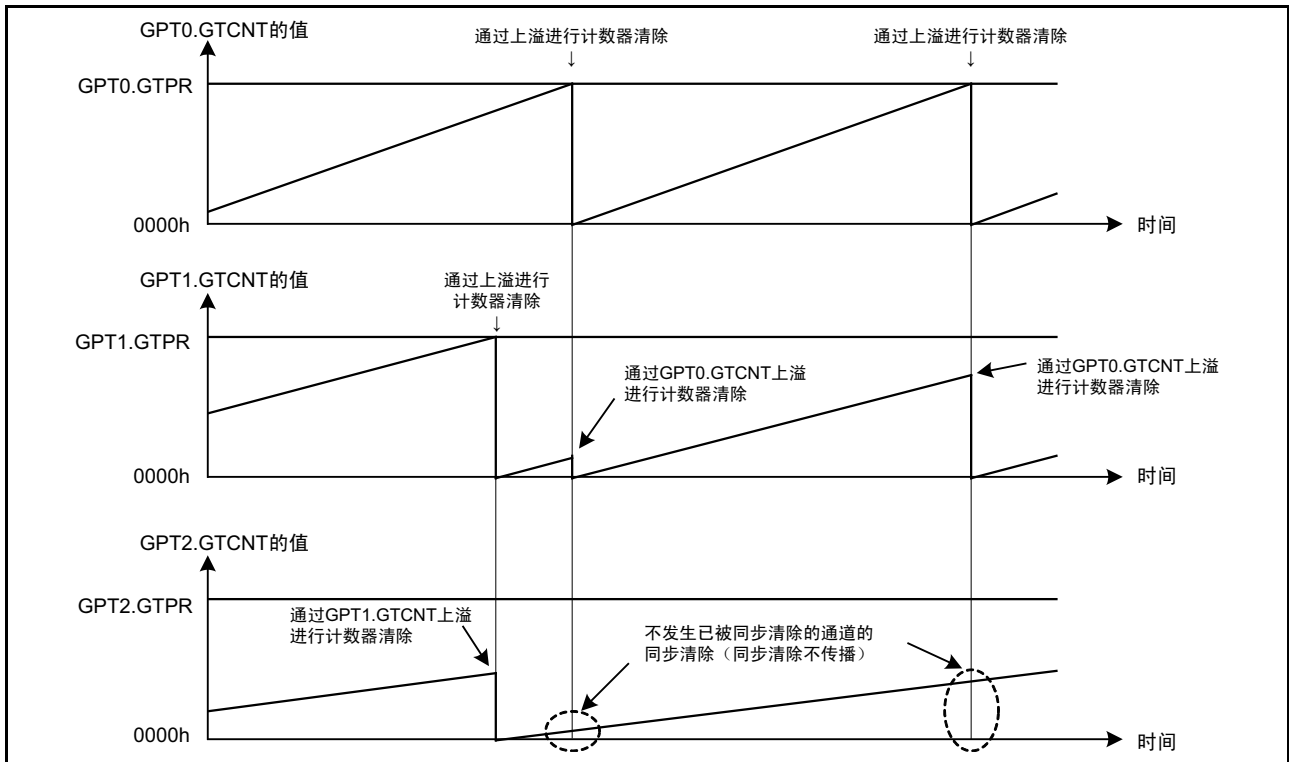


图 23.55 同步清除的运行例子
 (通过 GPT0.GTCNT 的清除源同步清除 GPT1.GTCNT、通过 GPT1.GTCNT 的清除源同步清除 GPT2.GTCNT)

<p>运行模式的设定 通过各通道的GTCR.MD[2:0]位设定运行模式。 (在图23.54、图23.55的例子中, 这些位为“000b”(锯齿波PWM模式)。)</p>
<p>计数方向的设定 通过各通道的GTUDC寄存器设定计数方向(递增/递减)。 (在图23.54、图23.55的例子中, 给GTUDC寄存器先设定“0003h”。再设定“0001h”(递增计数)。)</p>
<p>计数时钟的选择 通过各通道的GTCR.TPCS[1:0]位选择计数时钟。</p>
<p>周期的设定 给各通道的GTPR寄存器设定周期。</p>
<p>计数器初始值的设定 给各通道的GTCNT计数器设定初始值。</p>
<p>同步计数器清除的设定 通过各通道的GTCR.CCLR[1:0]位选择计数器清除源。将需要进行同步清除的通道 的GTCR.CCLR[1:0]位设定为“11b”。 通过GTSYNC.SYNCn[1:0]位设定通过哪个通道的清除源进行同步清除(n=0~3)。 (在图23.54的例子中, GPT0.GTCR.CCLR[1:0]=00b、GPT1.GTCR.CCLR[1:0]=11b、 GPT2.GTCR.CCLR[1:0]=11b、GTSYNC.SYNC1[1:0]=00b、 GTSYNC.SYNC2[1:0]=00b。) (在图23.55的例子中, GPT0.GTCR.CCLR[1:0]=00b、GPT1.GTCR.CCLR[1:0]=11b、 GPT2.GTCR.CCLR[1:0]=11b、GTSYNC.SYNC1[1:0]=00b、 GTSYNC.SYNC2[1:0]=01b。)</p>
<p>计数开始 将GTSTR.CST 位置“1”, 开始计数。</p>

图 23.56 同步清除运行的设定例子

23.3.7.2 同步启动运行

(1) 通过软件同时启动

能通过同时将 $GTSTR.CSTn$ ($n=0 \sim 3$) 位置“1”，同时开始各通道的计数运行。
通过软件同时启动的运行例子如图 23.57 所示。

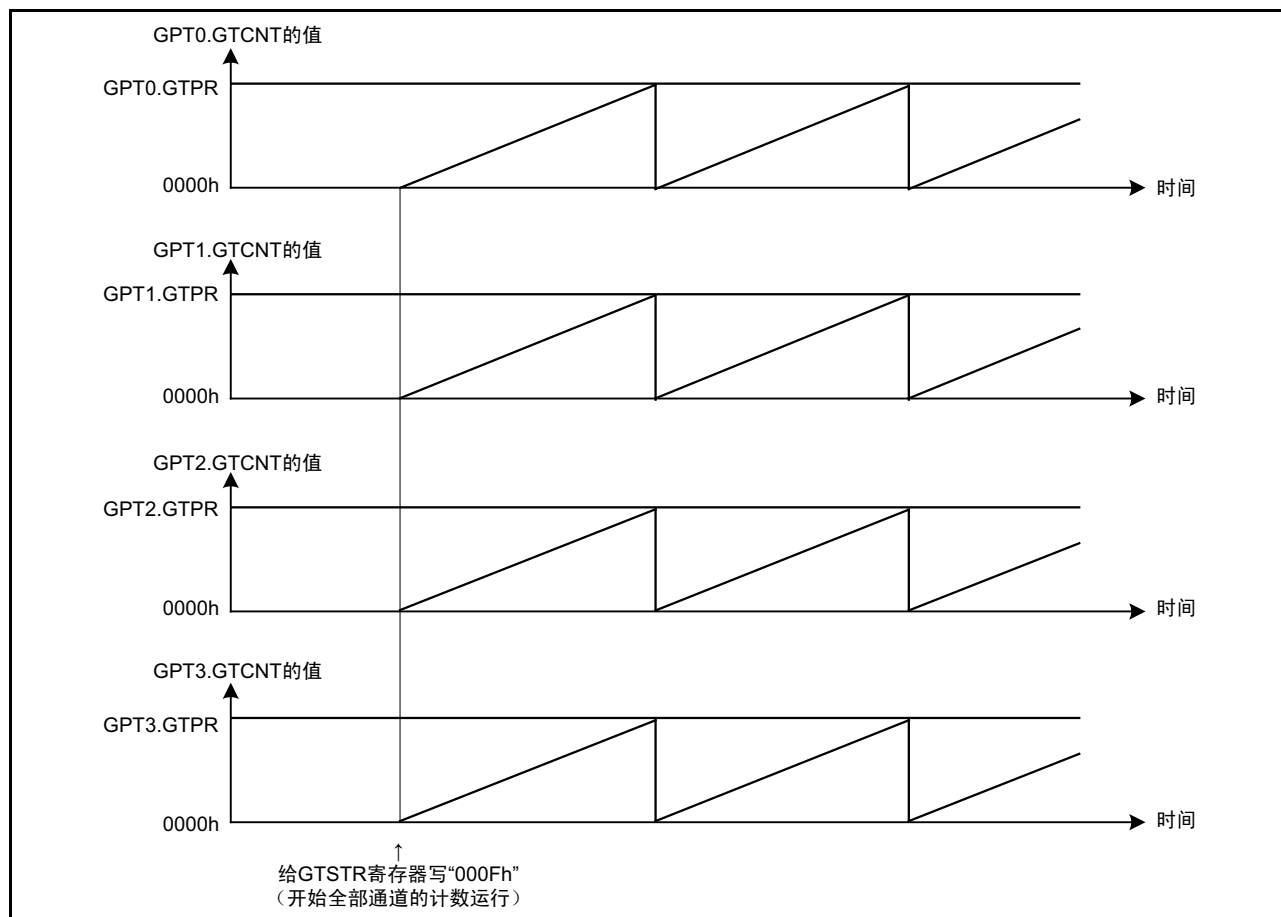


图 23.57 通过软件同时启动的运行例子 (计数周期 (GTPR 的值) 相同时)

(2) 通过软件进行相位启动

在计数开始前，事先设定各通道的 GTCNT 值，通过同时将 GTSTR.CSTn (n=0 ~ 3) 位置“1”，能在各通道间进行具有相位差的计数运行。

通过软件进行相位启动的运行例子如图 23.58 所示。

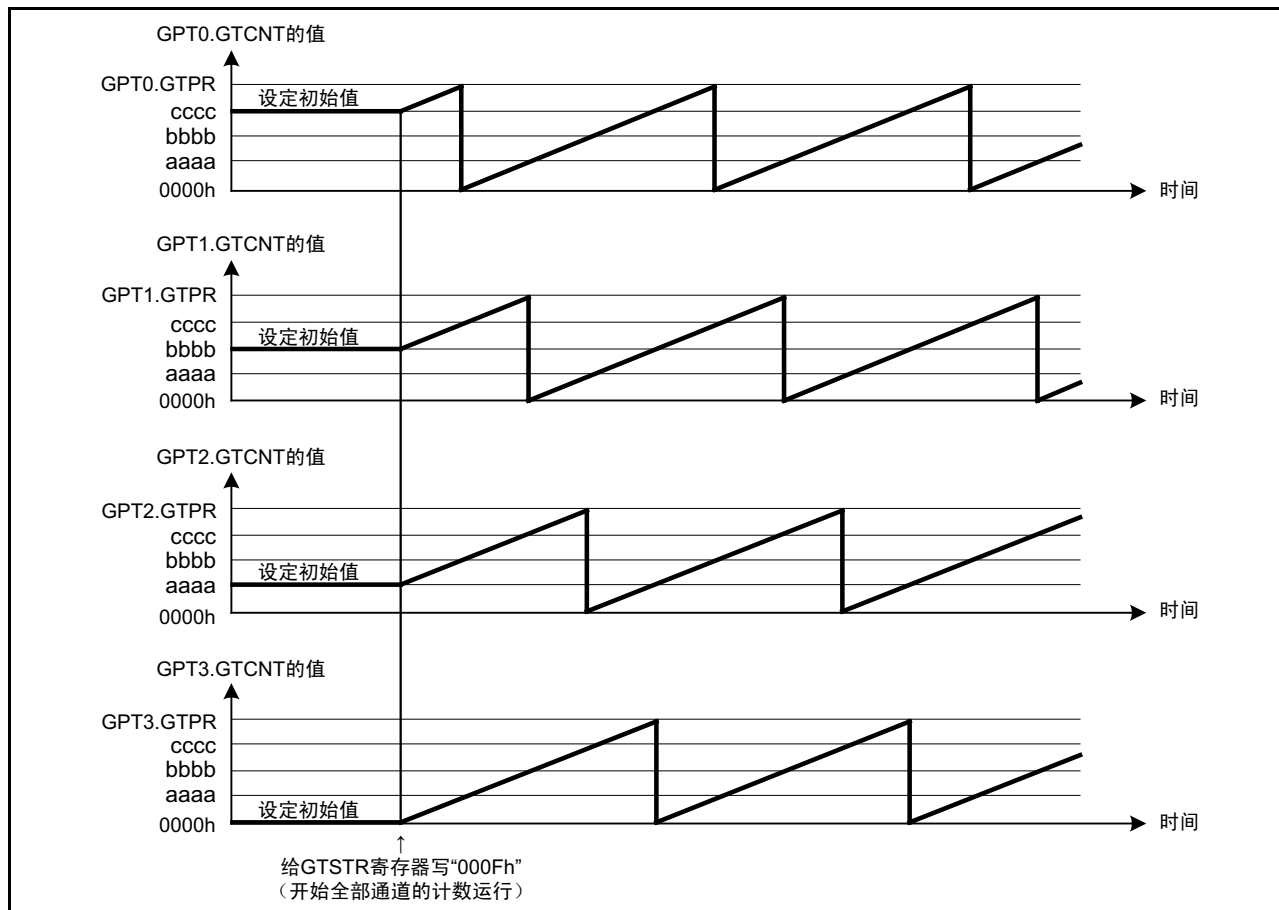


图 23.58 通过软件进行相位启动的运行例子 (计数周期 (GTPR 的值) 相同时)

(3) 通过硬件源同时启动

能通过 RX63T 群内置的硬件源，同时开始各通道的计数运行。

硬件源有 4 种，分别是 GTETR_G 引脚输入、比较器检测、GTIOC3A 和 GTIOC3B 的引脚输入、GTIOC3A 和 GTIOC3B 的内部输出（输出比较）。

通过硬件源同步启动的运行例子和设定例子分别如图 23.59、图 23.60 所示。这是通过 AN000 的比较器检测开始全部通道的计数运行的例子。

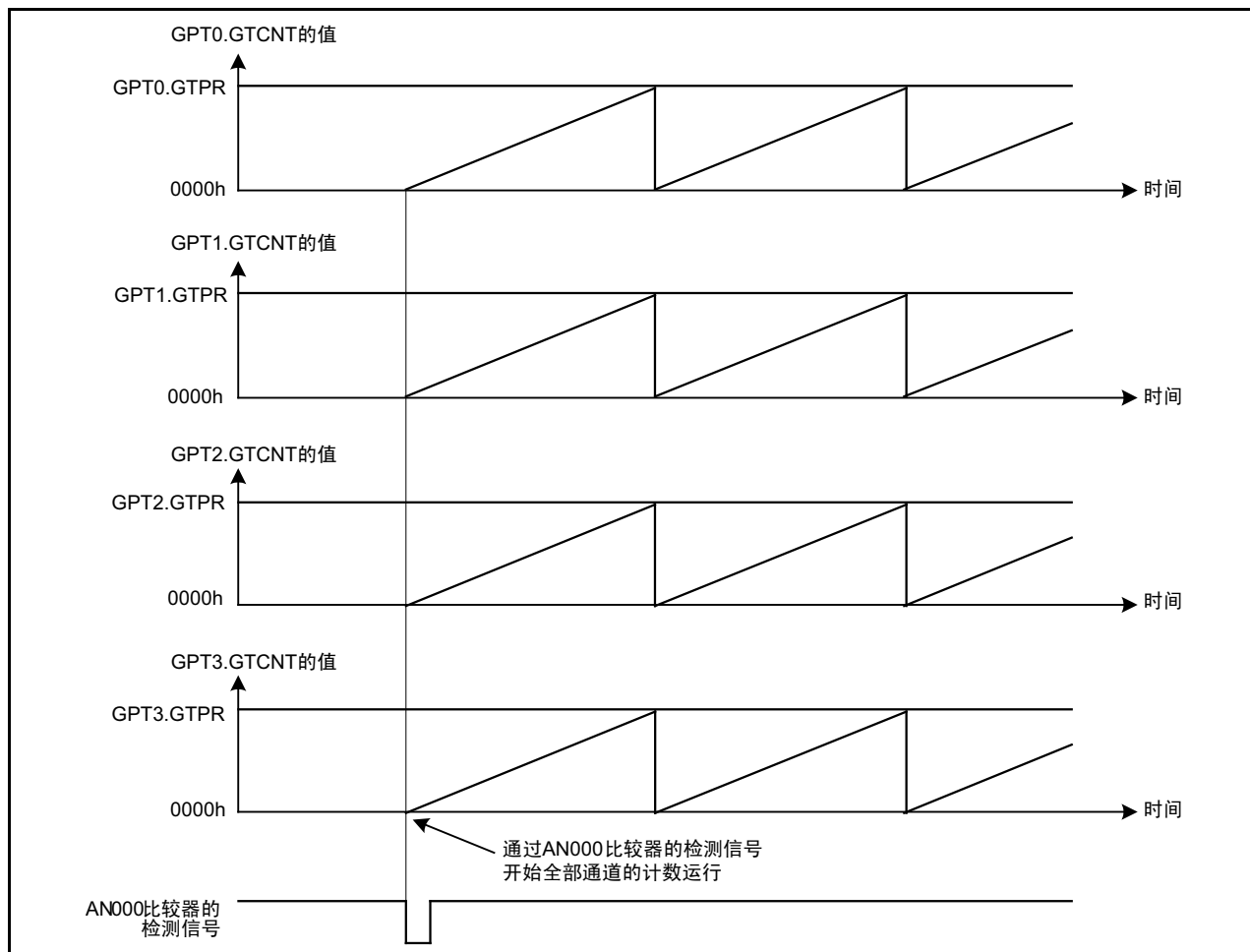


图 23.59 通过硬件源同步启动的运行例子（计数周期（GTPR 的值）相同时）

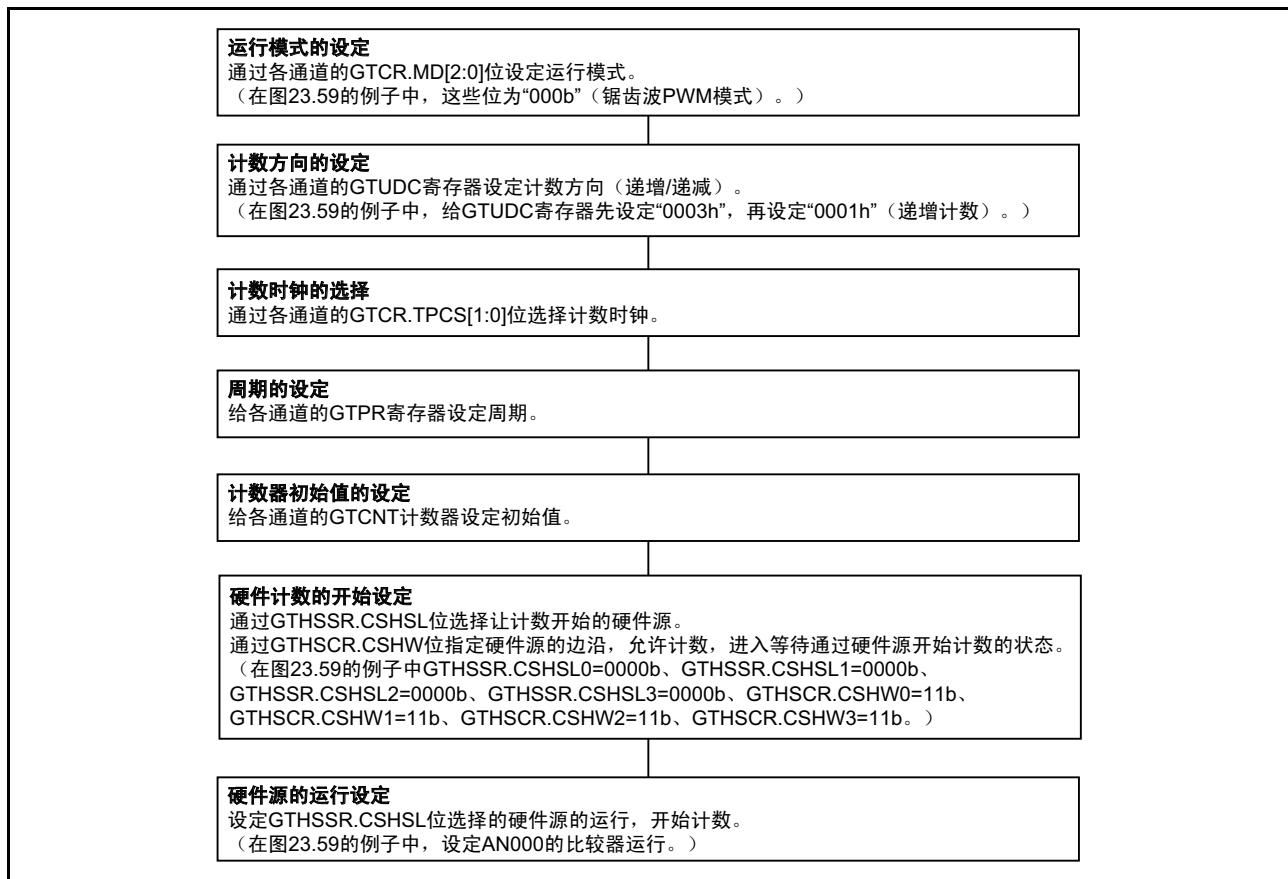


图 23.60 通过硬件源同步启动的运行设定例子

(4) 通过硬件源进行相位启动

能通过 RX63T 群内置的硬件源, 在各通道间开始进行具有相位差的计数运行。

硬件源有 4 种, 分别是 GTETRГ 引脚输入、比较器检测、GTIOC3A 和 GTIOC3B 的引脚输入、GTIOC3A 和 GTIOC3B 的内部输出 (输出比较)。

通过硬件源进行相位启动的运行例子和设定例子分别如图 23.61、图 23.62 所示。这是 GPT3.GTCNT 和 GPT0.GTCNT 通过软件同时开始计数运行, GPT1.GTCNT 和 GPT2.GTCNT 通过 GTIOC3A 和 GTIOC3B 的内部输出 (输出比较) 开始计数运行的例子。

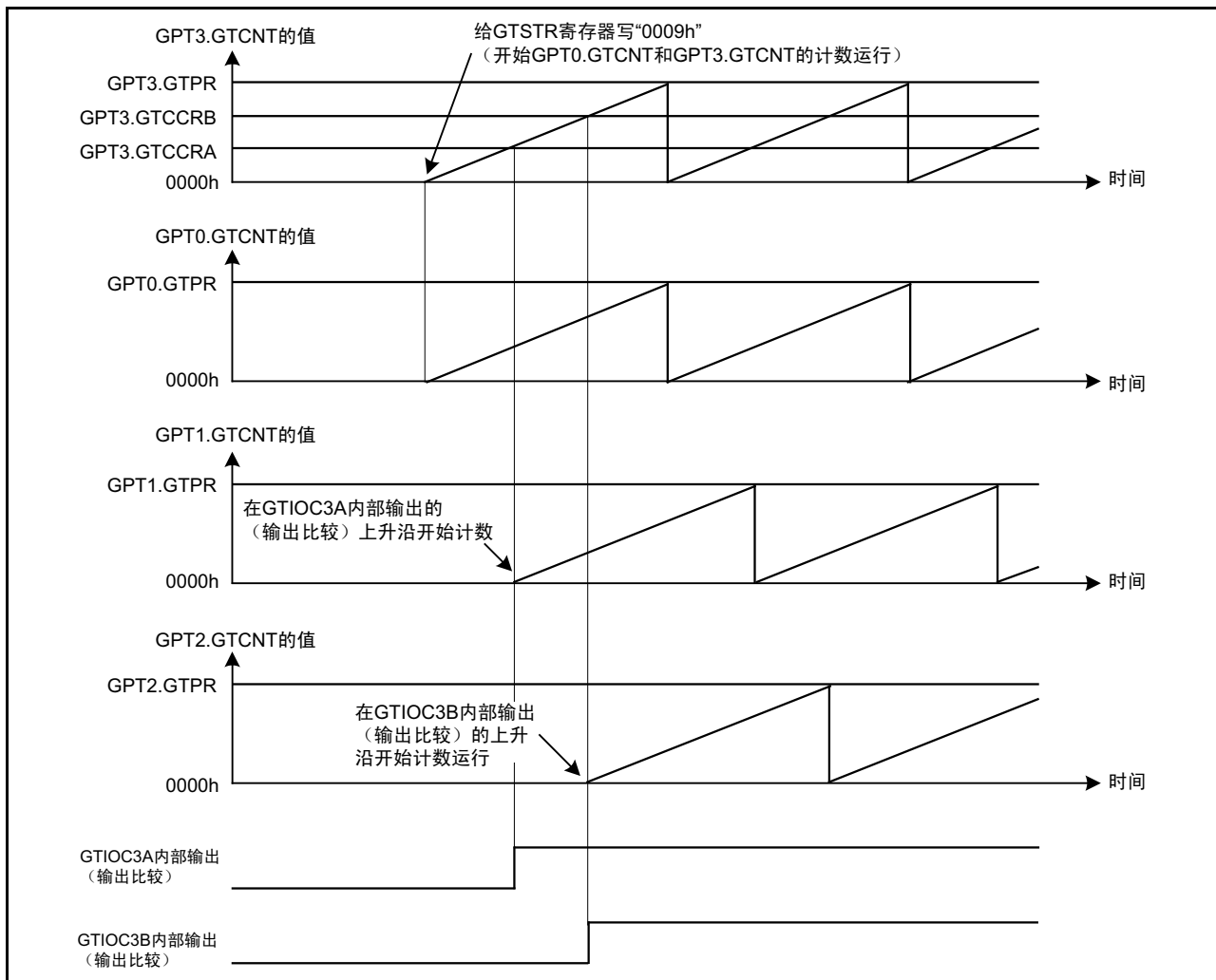


图 23.61 通过硬件源进行相位启动的运行例子 (计数周期 (GTPR 的值) 相同时)

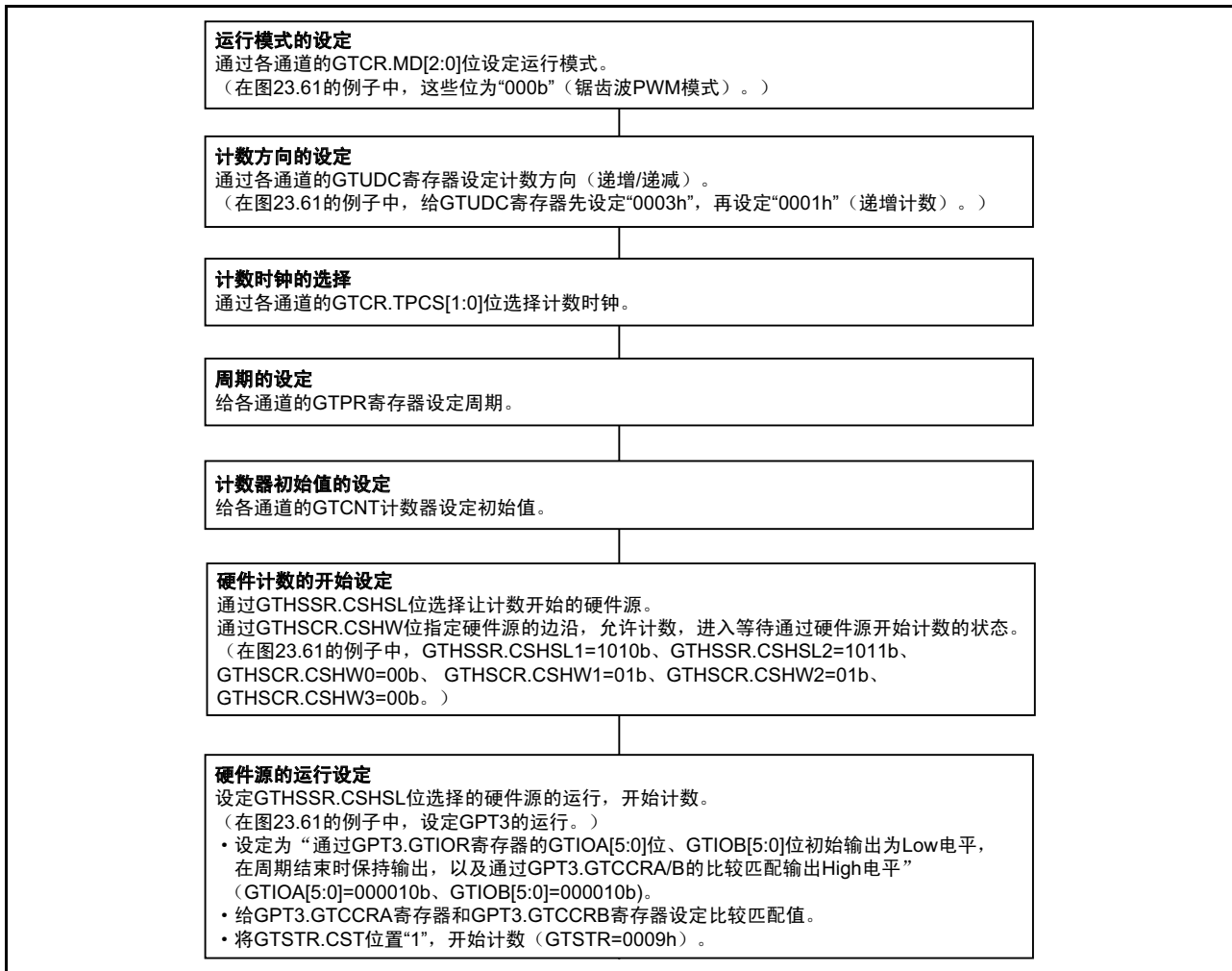


图 23.62 通过硬件源进行相位启动运行的设定例子

23.3.8 PWM 输出的运行例子

(1) 同步 PWM 输出

能通过通道间的同步运行输出最多 4 个通道的 8 相联动 PWM 波形。

图 23.63 是在锯齿波 PWM 模式中让全部的通道进行同步运行，并且输出 8 相 PWM 波形的例子。在此例子中如下设定 GTIOCnA 输出，初始输出为 Low 电平输出、通过 GTCCRA 寄存器的比较匹配进行 High 电平输出、在周期结束时进行 Low 电平输出；并且如下设定 GTIOCnB 输出，初始输出为 Low 电平输出、通过 GTCCRB 寄存器的比较匹配进行 High 电平输出、在周期结束时进行 Low 电平输出。

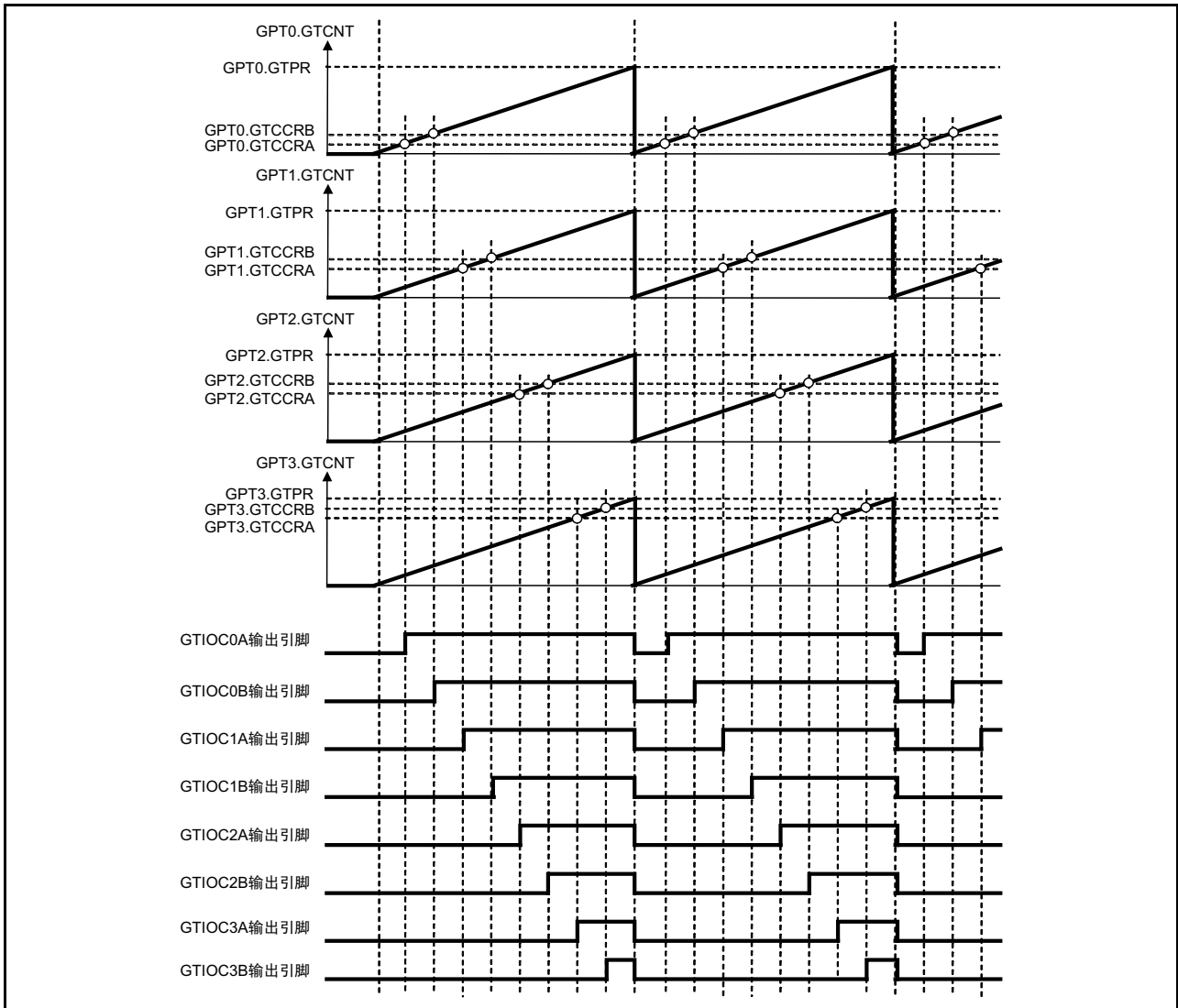


图 23.63 同步 PWM 输出的例子

(2) 锯齿波 3 相互补 PWM 输出

图 23.64 是在锯齿波 PWM 模式中让 3 个通道进行同步运行，并且输出 3 相互补 PWM 波形的例子。在此例子中如下设定 GTIOcN_A 输出，初始输出为 Low 电平输出、通过 GTCCRA 寄存器的比较匹配进行 High 电平输出、在周期结束时进行 Low 电平输出；并且如下设定 GTIOcN_B 输出，初始输出为 High 电平输出、通过 GTCCRB 寄存器的比较匹配进行 Low 电平输出、在周期结束时进行 High 电平输出。

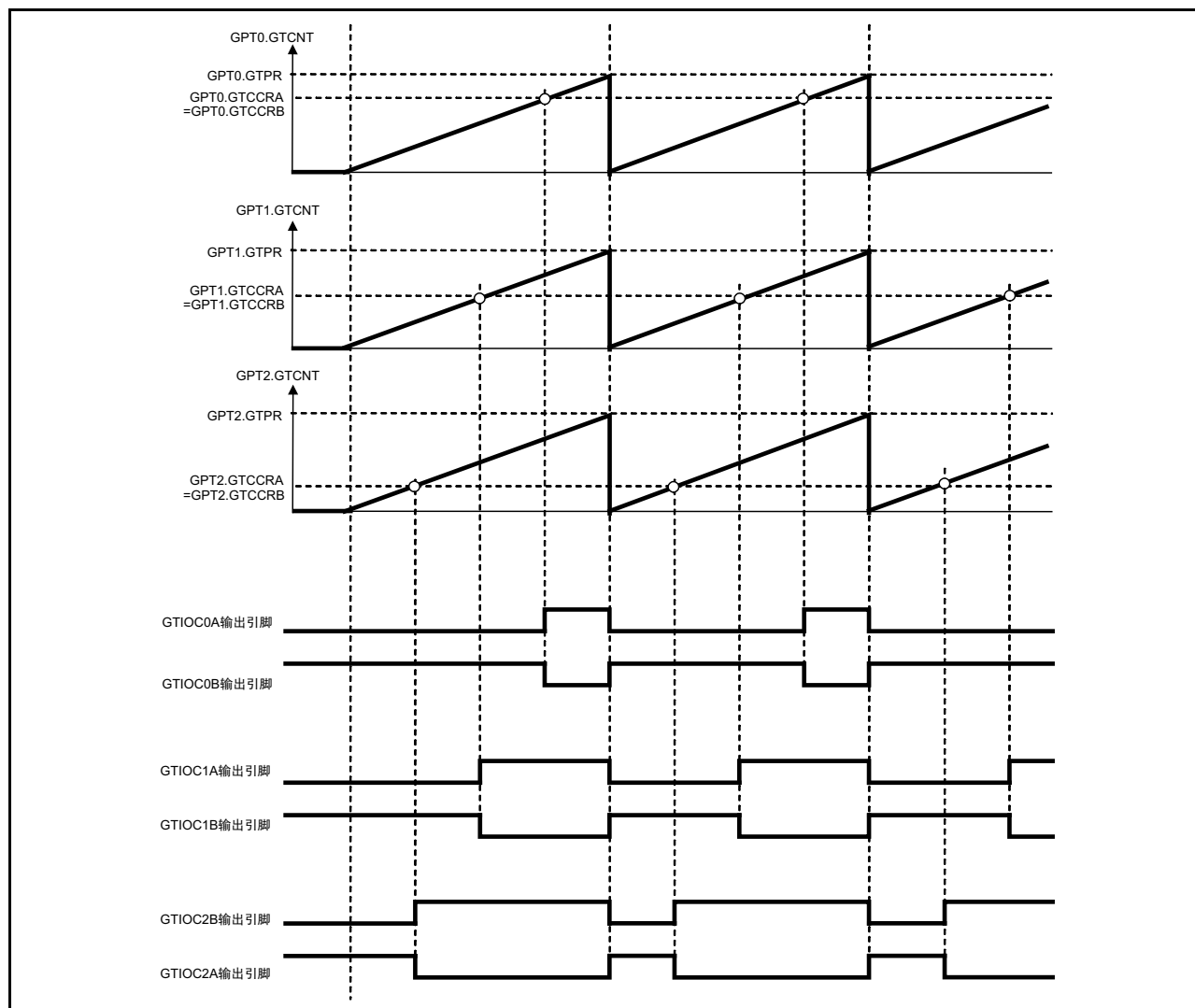


图 23.64 锯齿波 3 相互补 PWM 输出的例子

(3) 锯齿波 3 相互补 PWM 输出 (自动设定死区时间)

图 23.65 是在自动设定了死区时间的锯齿波单触发脉冲模式中让 3 个通道进行同步运行，并且输出 3 相互补 PWM 波形的例子。在此例子中如下设定 GTIOCnA 输出，初始输出为 Low 电平输出、通过 GTCCRA 寄存器的比较匹配进行交替输出、在周期结束时保持输出；并且如下设定 GTIOCnB 输出，初始输出为 High 电平输出、通过 GTCCRB 寄存器的比较匹配进行交替输出、在周期结束时保持输出。

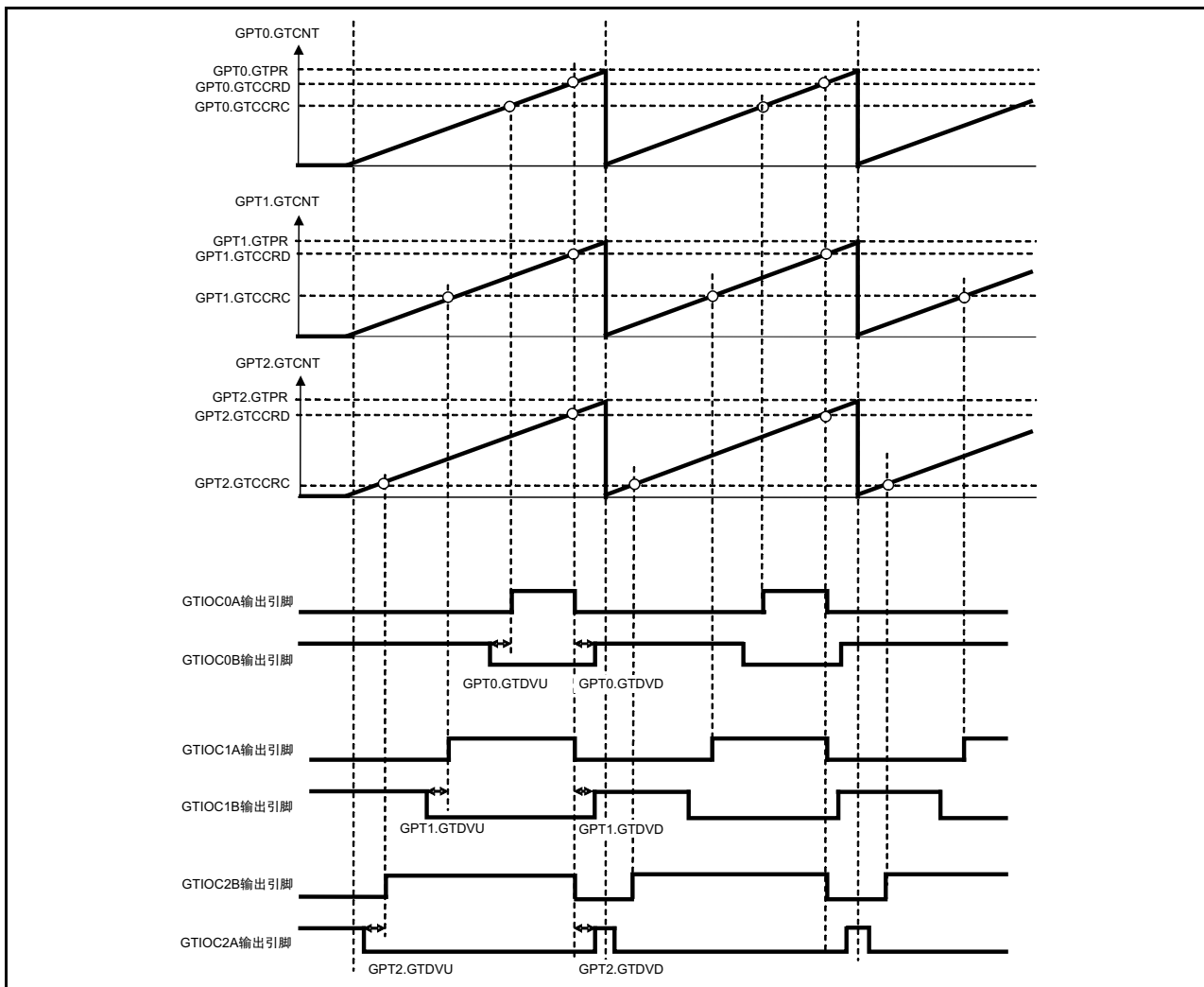


图 23.65 锯齿波 3 相互补 PWM 输出的例子 (自动设定死区时间)

(4) 三角波 3 相互补 PWM 输出

图 23.66 是在三角波 PWM 模式 1 中让 3 个通道进行同步运行，并且输出 3 相互补 PWM 波形的例子。在此例子中如下设定 GTIOCnA 输出，初始输出为 Low 电平输出、通过 GTCCRA 寄存器的比较匹配进行交替输出、在周期结束时保持输出；并且如下设定 GTIOCnB 输出，初始输出为 High 电平输出、通过 GTCCRB 寄存器的比较匹配进行交替输出、在周期结束时保持输出。

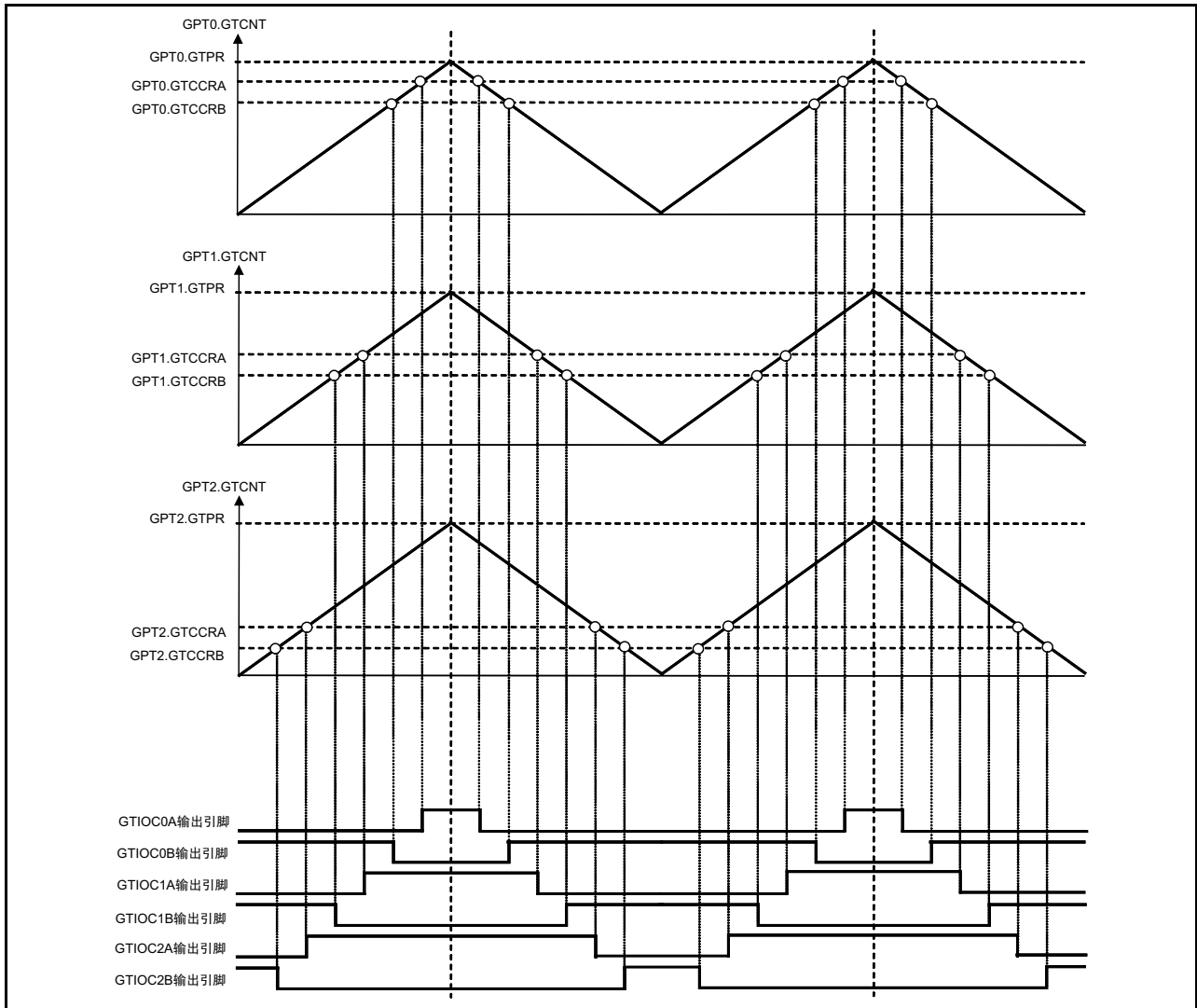


图 23.66 三角波 3 相互补 PWM 输出的例子

(5) 三角波 3 相互补 PWM 输出 (自动设定死区时间)

图 23.67 是在自动设定了死区时间的三角波 PWM 模式 1 中让 3 个通道进行同步运行，并且输出 3 相互补 PWM 波形的例子。在此例子中如下设定 GTIOCnA 输出，初始输出为 Low 电平输出、通过 GTCCRA 寄存器的比较匹配进行交替输出、在周期结束时保持输出；并且如下设定 GTIOCnB 输出，初始输出为 High 电平输出、通过 GTCCRB 寄存器的比较匹配进行交替输出、在周期结束时保持输出。

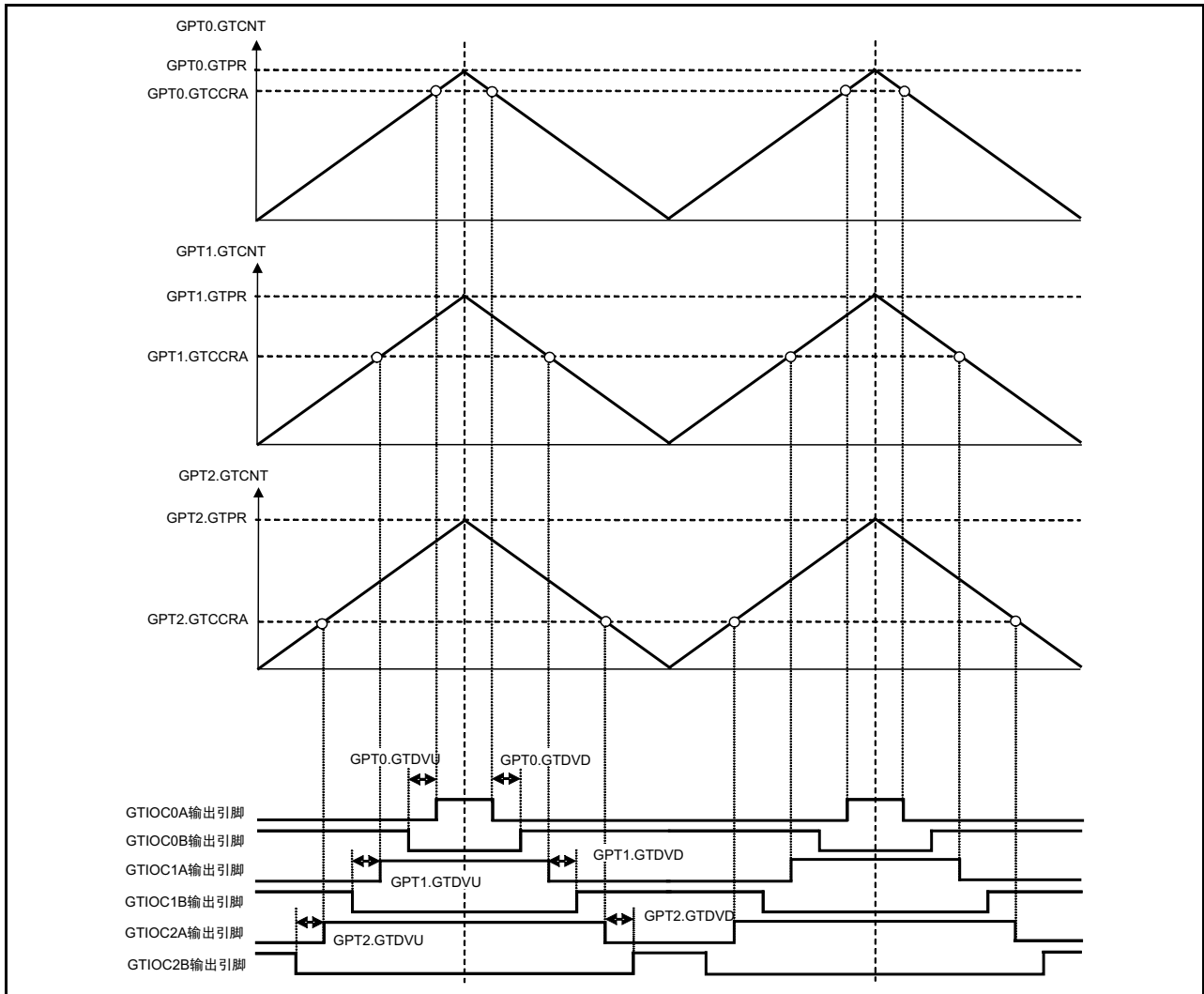


图 23.67 三角波 3 相互补 PWM 输出的例子 (自动设定死区时间)

(6) 非对称三角波 3 相互补 PWM 输出 (自动设定死区时间)

图 23.68 是在自动设定了死区时间的三角波 PWM 模式 3 中让 3 个通道进行同步运行，并且输出 3 相互补 PWM 波形的例子。在此例子中如下设定 GTIOCnA 输出，初始输出为 Low 电平输出、通过 GTCCRA 寄存器的比较匹配进行交替输出、在周期结束时保持输出；并且如下设定 GTIOCnB 输出，初始输出为 High 电平输出、通过 GTCCRB 寄存器的比较匹配进行交替输出、在周期结束时保持输出。

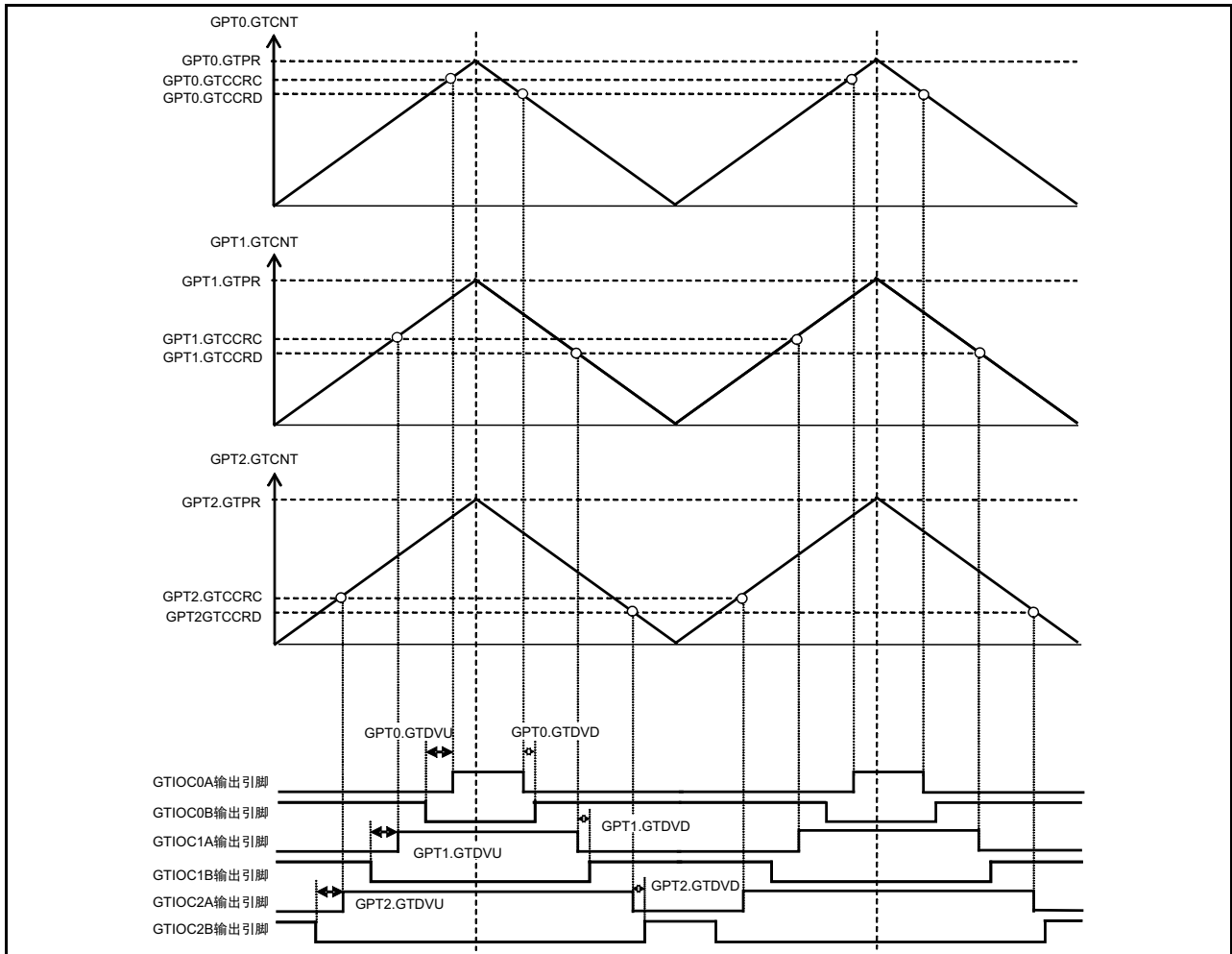


图 23.68 非对称三角波形 3 相互补 PWM 输出的例子 (自动设定死区时间)

23.4 中断源

23.4.1 中断源和优先级

GPT 的中断源有 4 种，分别是 GTCCR 寄存器的输入捕捉 / 比较匹配、GTCNT 计数器的上溢 (GTPR 寄存器的比较匹配) / 下溢、死区时间错误、IWDTCCLK 计数功能中断。各中断源分别具有专用的状态标志和产生中断请求的控制位，并且能独立允许或者禁止中断请求的产生。

如果产生中断源，GTST 寄存器对应的状态标志就变为“1”。此时，如果 GTINTAD 寄存器对应的中断请求允许 / 禁止位为“1”，就请求中断。但是，因为忽视该状态标志为“1”时的中断请求，所以为了能重新发生中断，必须将该状态标志置“0”。

通过中断控制器可更改通道间的优先级。通道间的优先级已被固定。详细内容请参照“14. 中断控制器 (ICUb)”，GPT 中断源的一览表如表 23.5 所示。

表 23.5 GPT 的中断源

通道	名称	中断源	中断标志	DTC 启动	优先级	
0	GTCIA0	GPT0.GTCCRA 的输入捕捉 / 比较匹配	TCFA	能	高 ↑ 低	
	GTCIB0	GPT0.GTCCRB 的输入捕捉 / 比较匹配	TCFB	能		
	GTCIC0	GPT0.GTCCRC 的比较匹配	TCFC	能		
		GPT0.GTCCRD 的比较匹配	TCFD			
		死区时间错误	DTEF			
	GTCIE0	GPT0.GTCCRE 的比较匹配	TCFE	能		
		GPT0.GTCCRF 的比较匹配	TCFF			
	GTCIV0	GPT0.GTCNT 的上溢 (GPT0.GTPR 的比较匹配)	TCFPO	能		
		GPT0.GTCNT 的下溢	TCFPU			
	LOCOI	IWDTCLK 计数功能中断	LCNT 的上溢	LISO		能
			IWDTCLK 计数值偏差超出	LISD		
			IWDTCLK 分频时钟的上升沿	LISC		
		外部输入触发	外部触发下降沿的输入	ETINF		
外部触发上升沿的输入			ETIPF			
1	GTCIA1	GPT1.GTCCRA 的输入捕捉 / 比较匹配	TCFA	能		
	GTCIB1	GPT1.GTCCRB 的输入捕捉 / 比较匹配	TCFB	能		
	GTCIC1	GPT1.GTCCRC 的比较匹配	TCFC	能		
		GPT1.GTCCRD 的比较匹配	TCFD			
		死区时间错误	DTEF			
	GTCIE1	GPT1.GTCCRE 的比较匹配	TCFE	能		
		GPT1.GTCCRF 的比较匹配	TCFF			
	GTCIV1	GPT1.GTCNT 的上溢 (GPT1.GTPR 的比较匹配)	TCFPO	能		
		GPT1.GTCNT 的下溢	TCFPU			
	2	GTCIA2	GPT2.GTCCRA 的输入捕捉 / 比较匹配	TCFA	能	
		GTCIB2	GPT2.GTCCRB 的输入捕捉 / 比较匹配	TCFB	能	
GTCIC2		GPT2.GTCCRC 的比较匹配	TCFC	能		
		GPT2.GTCCRD 的比较匹配	TCFD			
		死区时间错误	DTEF			
GTCIE2		GPT2.GTCCRE	TCFE	能		
		GPT2.GTCCRF	TCFF			
GTCIV2		GPT2.GTCNT 的上溢 (GPT2.GTPR 的比较匹配)	TCFPO	能		
		GPT2.GTCNT 的下溢	TCFPU			
3	GTCIA3	GPT3.GTCCRA 的输入捕捉 / 比较匹配	TCFA	能		
	GTCIB3	GPT3.GTCCRB 的输入捕捉 / 比较匹配	TCFB	能		
	GTCIC3	GPT3.GTCCRC 的比较匹配	TCFC	能		
		GPT3.GTCCRD 的比较匹配	TCFD			
		死区时间错误	DTEF			
	GTCIE3	GPT3.GTCCRE 的比较匹配	TCFE	能		
		GPT3.GTCCRF 的比较匹配	TCFF			
	GTCIV3	GPT3.GTCNT 的上溢 (GPT3.GTPR 的比较匹配)	TCFPO	能		
		GPT3.GTCNT 的下溢	TCFPU			

注. 表示复位后的初始状态。通过中断控制器可更改通道间的优先级。

(1) 输入捕捉 / 比较匹配中断

如果发生各通道的 GTCCR 寄存器的输入捕捉 / 比较匹配，GTST 寄存器的对应状态标志就变为“1”。此时，如果 GTINTAD 寄存器的对应中断允许位为“1”，就请求中断。

(2) 上溢 / 下溢 (周期) 中断

能通过设定各通道的 GTPR 寄存器，发生决定中断间隔的周期中断。

在锯齿波并且进行递增计数的情况下，如果在 GTCNT 计数器的值和 GTPR 寄存器的值相同时 (上溢) GTST.TCFPO 标志为“1”，GTST.TCFPU 标志就变为“1”；在锯齿波并且进行递减计数的情况下，如果 GTCNT 计数器的值为“0” (下溢)，GTST.TCFPU 标志就变为“1”。此时，如果将 GTINTAD.GTINTPR[1:0] 位的对应位设定为“01b”、“10b”或者“11b”，就请求 GTCIV 中断。

在三角波的情况下，如果 GTCNT 计数器的值和 GTPR 寄存器的值相同 (波峰)，GTST.TCFPO 标志就变为“1”，并且在 GTCNT 计数器的值为“0” (波谷) 时，GTST.TCFPU 标志就变为“1”。此时，如果将 GTINTAD.GTINTPR[1:0] 位的对应位设定为“01b”、“10b”或者“11b”，就请求 GTCIV 中断。

(3) IWDTCCLK 计数功能中断

如果 LCCR 寄存器的对应中断允许位为“1”，通过发生 IWDTCCLK 分频时钟的上升沿检测、IWDTCCLK 计数值偏差超出或者 LCNT 计数器的上溢，LCST 寄存器的对应状态标志就变为“1”，并且请求 LOCOI 中断。

同样，通过发生 GTETRG 外部触发输入的上升沿或者下降沿检测，GTETINT 寄存器的对应状态标志就变为“1”。此时，如果 GTETINT 寄存器的对应中断允许位为“1”，就请求 LOCOI 中断。

(4) 死区时间错误中断

当设定为自动设定死区时间时，如果自动附加后的定时器输出交替时序超出定时器周期，GTST.DTEF 标志就变为“1”。此时，如果 GTINTAD.EINT 位为“1”，就请求 LOCOI 中断。

表 23.6 中断信号、中断允许位和状态标志

中断信号	中断允许	状态标志
GTCIV	GTINTAD[7:6](GTINTPR[1:0])	GTST[7] (TCFPU)
		GTST[6] (TCFPO)
GTCIE	GTINTAD[5] (GTINTF)	GTST[5] (TCFF)
	GTINTAD[4] (GTINTE)	GTST[4] (TCFE)
GTCIC	GTINTAD[11](EINT)	GTST[11] (DTEF)
	GTINTAD[3] (GTINTD)	GTST[3] (TCFD)
	GTINTAD[2] (GTINTC)	GTST[2] (TCFC)
GTCIB	GTINTAD[1] (GTINTB)	GTST[1] (TCFB)
GTCIA	GTINTAD[0] (GTINTA)	GTST[0] (TCFA)
LOCOI	LCCR[6] (LCINTO)	LCST[2] (LISO)
	LCCR[5] (LCINTD)	LCST[1] (LISD)
	LCCR[4] (LCINTC)	LCST[0] (LISC)
	GTETINT[1] (ETINEN)	GTETINT[9] (ETINF)
	GTETINT[0] (ETIPEN)	GTETINT[8] (ETIPF)

(5) 同时使用中断源时的注意事项

在多个中断源重叠的中断 GTCICn、GTCIE_n、GTCIV_n、LOCOI (n=0、1、2、3) 中，因某个中断源（例如，GTCIC_n 的 TCFC 标志）使 ICUA 的中断请求标志（IR 标志）变为“1”的期间，即使产生与该中断源（GTCIC_n 的 TCFC 标志）重叠的另一个中断源（GTCIC_n 的 TCFD 标志）的中断请求，也忽视该中断源 GTCIC_n 的 TCFD 标志的中断请求。

因此，在同时使用重叠的多个中断源时，必须先确认中断处理程序中使用的全部中断源的标志，然后分别根据被置为有效的中断源的标志进行处理。

23.4.2 DTC 的启动

能通过各通道的中断启动 DTC。详细内容请参照“14. 中断控制器 (ICUb)”、“18. 数据传送控制器 (DTCa)”。

但是，与中断处理同样，也忽视对应状态标志为“1”时的 DTC 启动请求，因此为了重新产生 DTC 启动请求，必须将对应的状态标志置“0”。

23.4.3 中断、A/D 转换开始请求的减少功能

能通过设定 GTITC 寄存器减少 GTCNT 计数器的上溢 (GTPR 寄存器的比较匹配) / 下溢中断 (GTCIV)。联动 GTCIV 中断减少功能减少其他中断和 A/D 转换开始请求。但是，死区时间错误中断不能联动 GTCIV 中断减少功能。中断减少后，对应的状态标志的变化也减少，并且在状态标志为“1”的期间保持中断减少功能的运行。

在三角波中对波峰和波谷进行计数并且减少功能有效时，如果将减少次数设定为奇数，根据减少计数器的开始时序，不产生只在波峰或者只在波谷的 GTCIV 中断请求。在三角波中对波峰和波谷进行计数并且减少功能有效时，如果使用只在波峰或者只在波谷产生的 GTCIV 中断，就必须将减少次数设定为偶数。

同样，在锯齿波中一边转换计数方向一边对上溢和下溢进行计数并且减少功能有效时，有可能不产生只在发生上溢或者只在发生下溢时的 GTCIV 中断。在锯齿波中一边转换计数方向一边对上溢和下溢进行计数并且减少功能有效时，如果使用只在发生上溢或者只在发生下溢时产生的 GTCIV 中断，就必须在充分检测减少状态后再使用。

必须在暂时解除减少功能 (GTITC.IVIC[1:0]=00b) 后，才能更改减少次数。

减少功能的运行例子如图 23.69 ~ 图 23.74 所示。

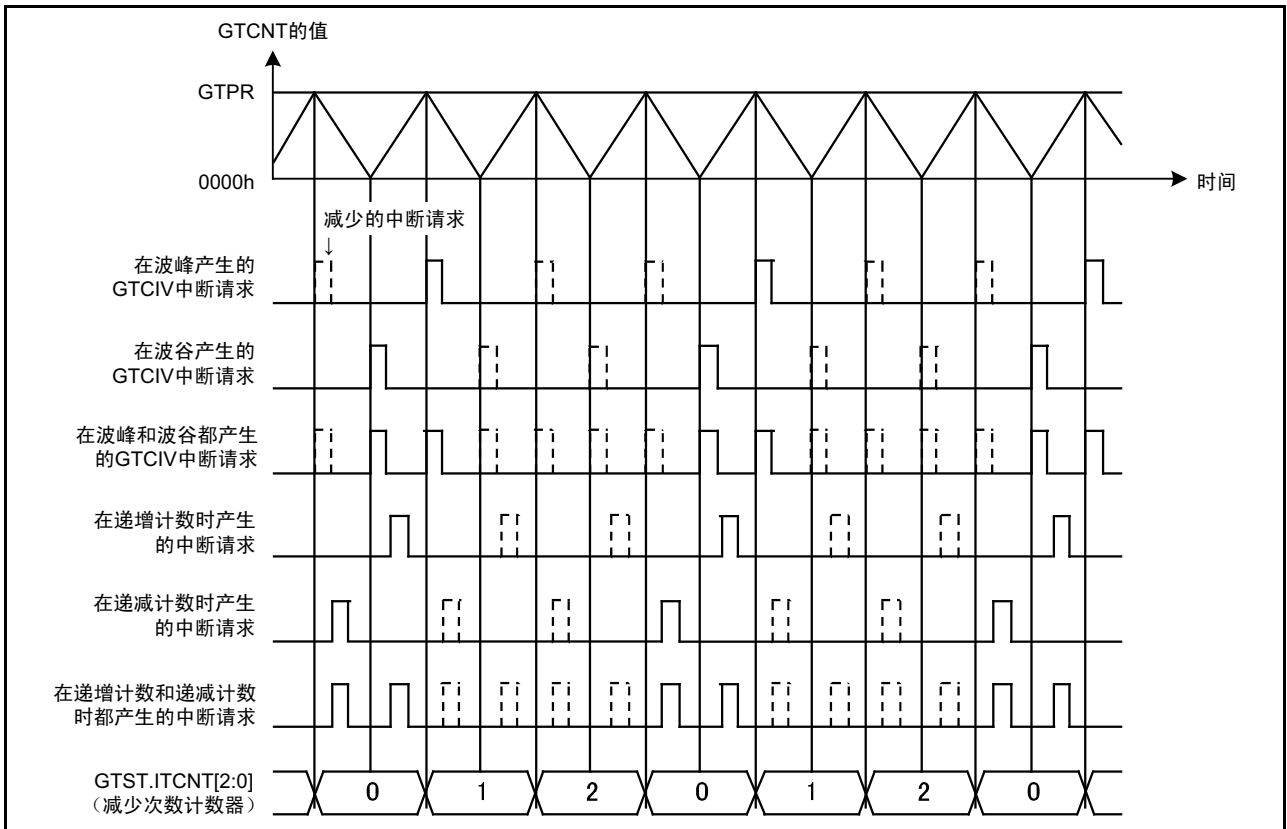


图 23.69 中断减少功能的运行例子 (三角波、对波谷进行计数并且减少功能有效、减少次数为 2 次)

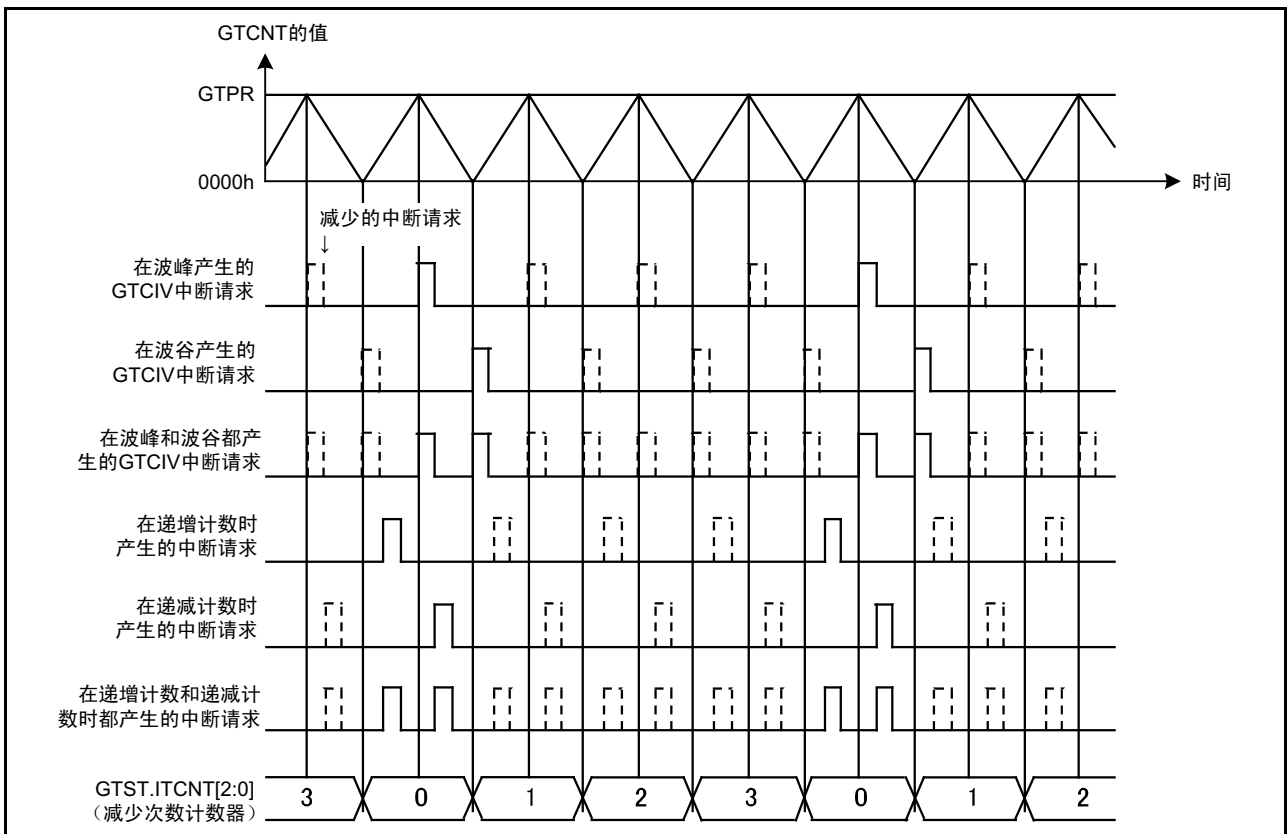


图 23.70 中断减少功能的运行例子 (三角波、对波谷进行计数并且减少功能有效、减少次数为 3 次)

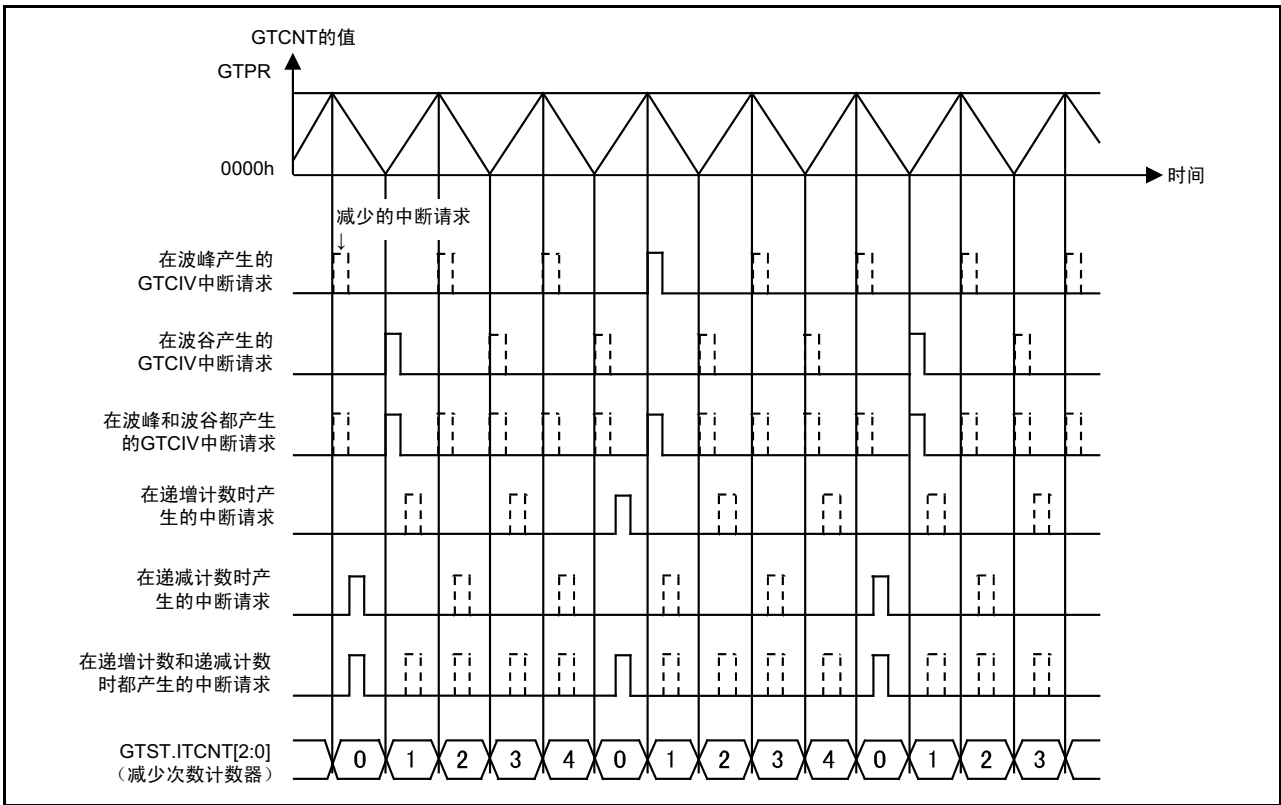


图 23.71 中断减少功能的运行例子
(三角波、对波谷和波峰进行计数并且减少功能有效、减少次数为 4 次)

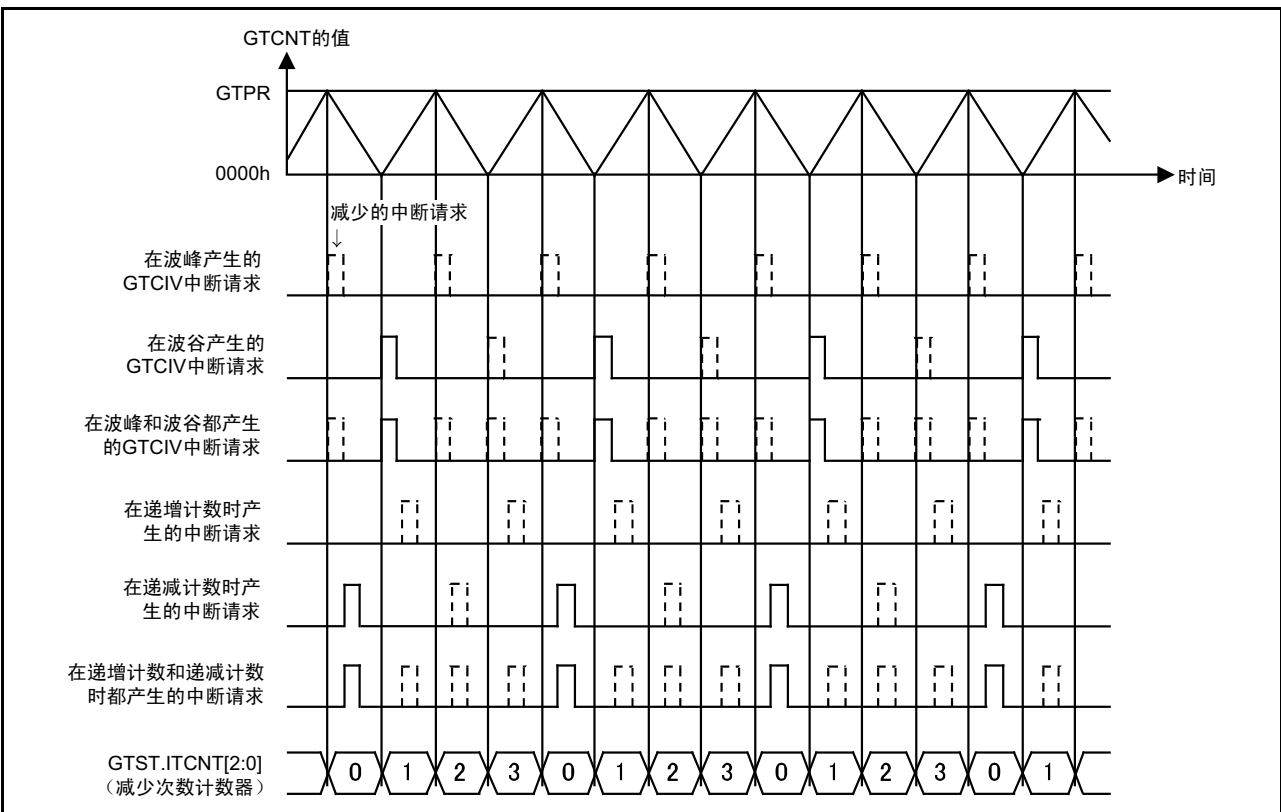


图 23.72 中断减少功能的运行例子
(三角波、对波谷和波峰进行计数并且减少功能有效、减少次数为 3 次、在递增计数时开始减少)

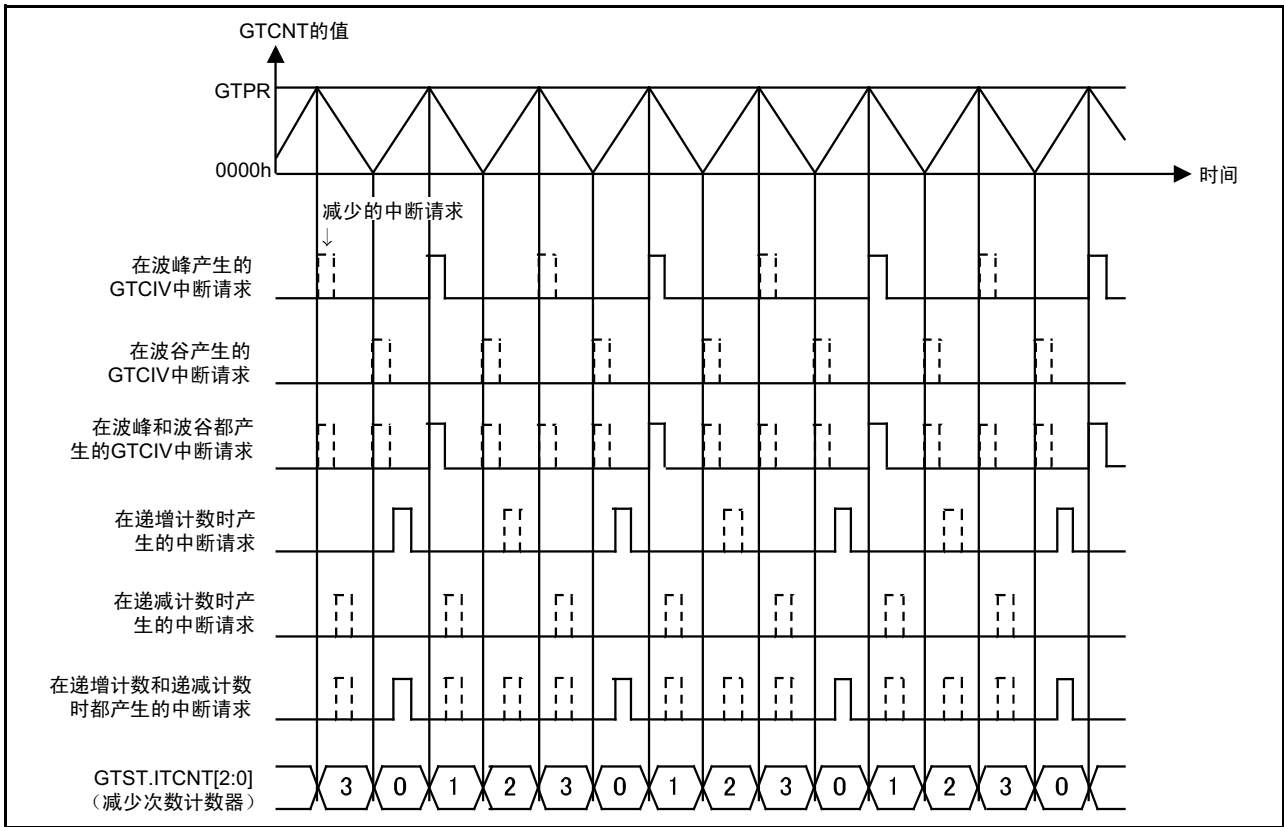


图 23.73 中断减少功能的运行例子
 (三角波、对波谷和波峰进行计数并且减少功能有效、减少次数为 3 次、在递减计数时开始减少)

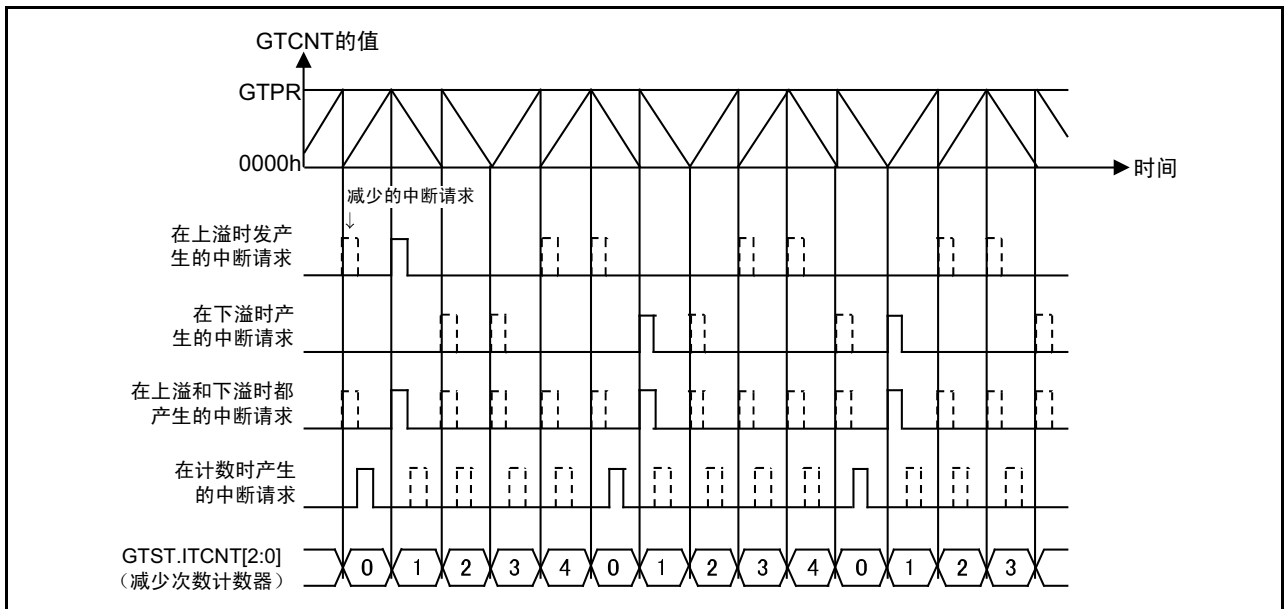


图 23.74 中断减少功能的运行例子
 (锯齿波、一边转换计数方向一边对上溢和下溢进行计数并且减少功能有效、减少次数为 4 次)

23.5 A/D 转换开始请求

能通过 GTCNT 计数器和 GTADTRA 寄存器、GTADTRB 寄存器的比较匹配产生 A/D 转换开始请求。能只在递增计数、只在递减计数、或者递增和递减计数时产生 A/D 转换开始请求。

GTADTRA 寄存器和 GTADTRB 寄存器各有两个缓冲寄存器。能用作 GTADTRA 寄存器和 GTADTBRA 寄存器、GTADTDBRA 寄存器组合的缓冲运行，以及 GTADTRB 寄存器和 GTADTBRB 寄存器、GTADTDBRB 寄存器组合的缓冲运行。

A/D 转换开始请求的运行例子如图 23.75， A/D 转换开始请求运行的设定例子如图 23.76 所示。

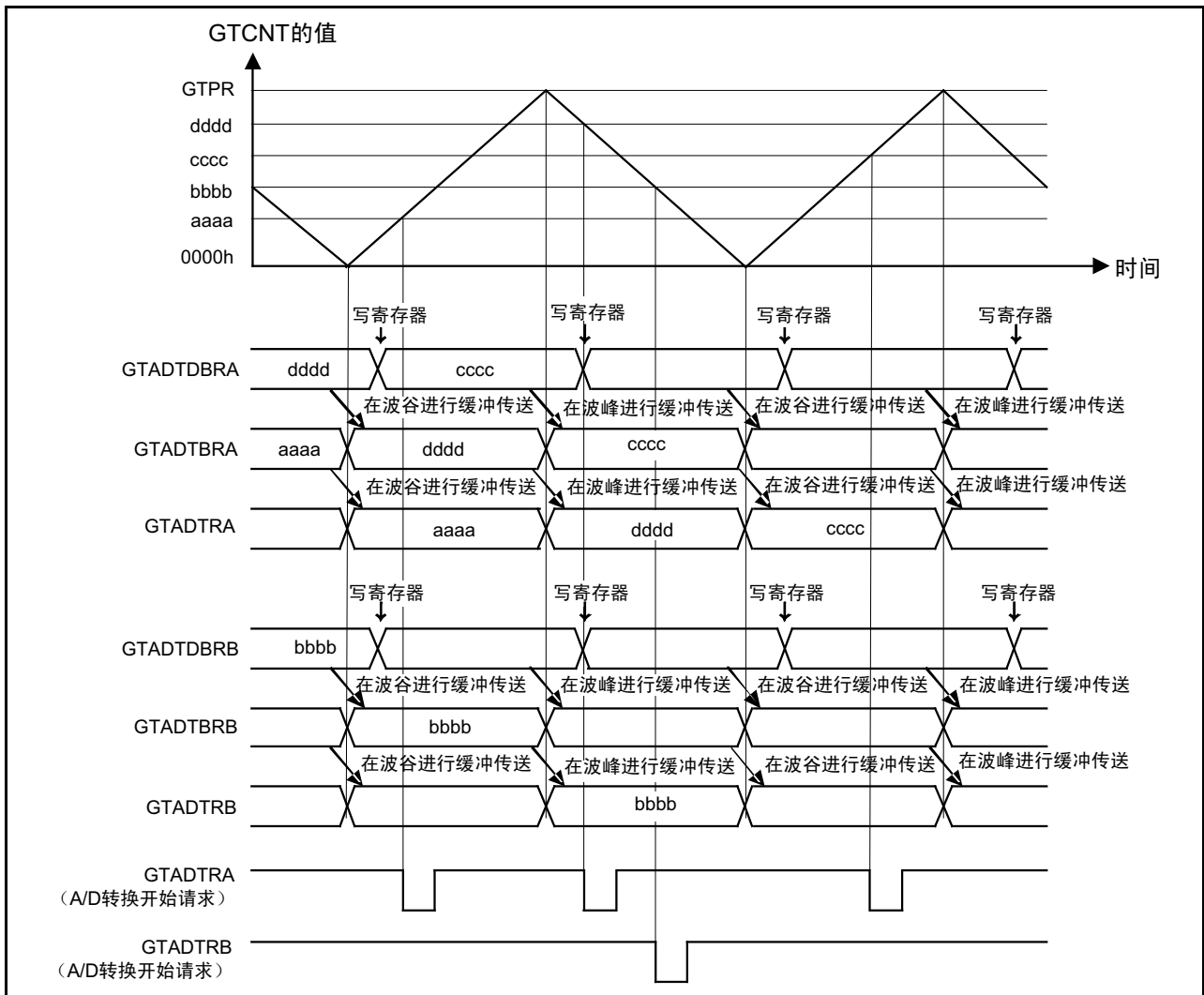


图 23.75 A/D 转换开始请求的运行例子

(三角波、双缓冲运行、在波谷和波峰进行缓冲传送、GTADTRA0 在递增计数和递减计数时都产生 A/D 转换开始请求， GTADTRB0 在递减计数时产生 A/D 转换开始请求)

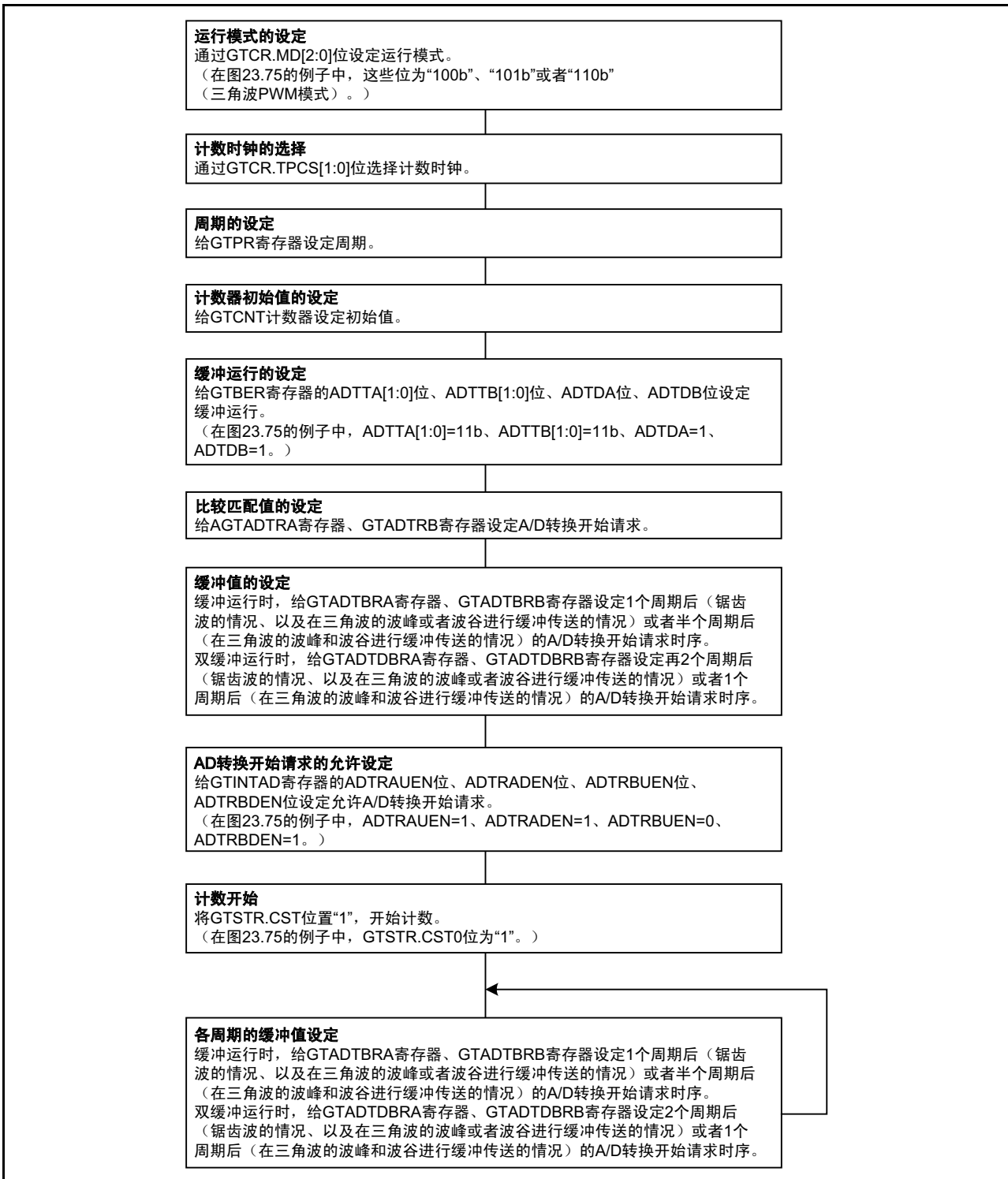


图 23.76 A/D 转换开始请求运行的设定例子

23.6 IWDTCCLK 计数功能

IWDTCCLK 计数功能测量 IWDTC 专用低速时钟 (IWDTCCLK) 的周期。通过此功能还能检测出主时钟振荡频率的异常。

能从 IWDTCCLK 时钟的 1 分频时钟、16 分频时钟、128 分频时钟和 256 分频时钟中选择作为测量对象的 IWDTCCLK 时钟 (IWDTCCLK 分频时钟)。能从 PCLKA 的 1 分频时钟、2 分频时钟、4 分频时钟和 8 分频时钟中选择测量 IWDTCCLK 分频时钟的计数时钟。

LCNT 计数器通过计数时钟对 IWDTCCLK 分频时钟进行计数。前 16 次的计数结果保存到 LCNT00 ~ LCNT15 寄存器 (最新的计数结果保存在 LCNT00 寄存器)。计算前 16 次的计数结果的平均值, 并且保存到 LCNTA 寄存器。

当检测到 IWDTCCLK 分频时钟的上升沿时, 产生 IWDTCCLK 分频时钟上升沿中断请求。能减少 IWDTCCLK 分频时钟上升沿中断请求, 此时, 计数结果也减少。如果计数结果 (LCNT00 寄存器的值) 超出上限值/下限值, 就能产生 IWDTCCLK 偏差超出中断请求。能通过 LCNTDU 寄存器和 LCNTDL 寄存器设定上限值和下限值。上限值为“LCNTA 寄存器的值+LCNTDU 寄存器的值”, 下限值为“LCNTA 寄存器的值-LCNTDL 寄存器的值”。如果因为 IWDTCCLK 分频时钟振荡慢而使 LCNT 计数器发生上溢, 就产生 LCNT 上溢中断请求。另外, 由于发生的中断全部作为 LOCOI 中断被输出, 所以必须在中断发生后读各状态标志, 并且判断中断源。

使用 IWDTCCLK 计数功能时, 必须使独立看门狗定时器 (IWDTC) 运行。

IWDTCCLK 计数功能的周期设定例子如表 23.7 所示。

IWDTCCLK 计数功能的运行例子和设定例子分别如图 23.77、图 23.78 所示。

表 23.7 IWDTCCLK 计数功能的周期设定例子

IWDTCCLK 分频时钟		计数时钟		计数结果的理想值
LCCR.LPSC[1:0] 位的设定	频率	LCCR.TPSC[1:0] 位的设定	频率 (当 PCLKA=100MHz 时)	
00 (1 分频)	125kHz	00 (1 分频)	100MHz	320h
01 (16 分频)	7.81kHz	00 (1 分频)	100MHz	3200h
10 (128 分频)	976Hz	10 (4 分频)	25MHz	6400h
11 (256 分频)	488Hz	11 (8 分频)	12.5MHz	6400h

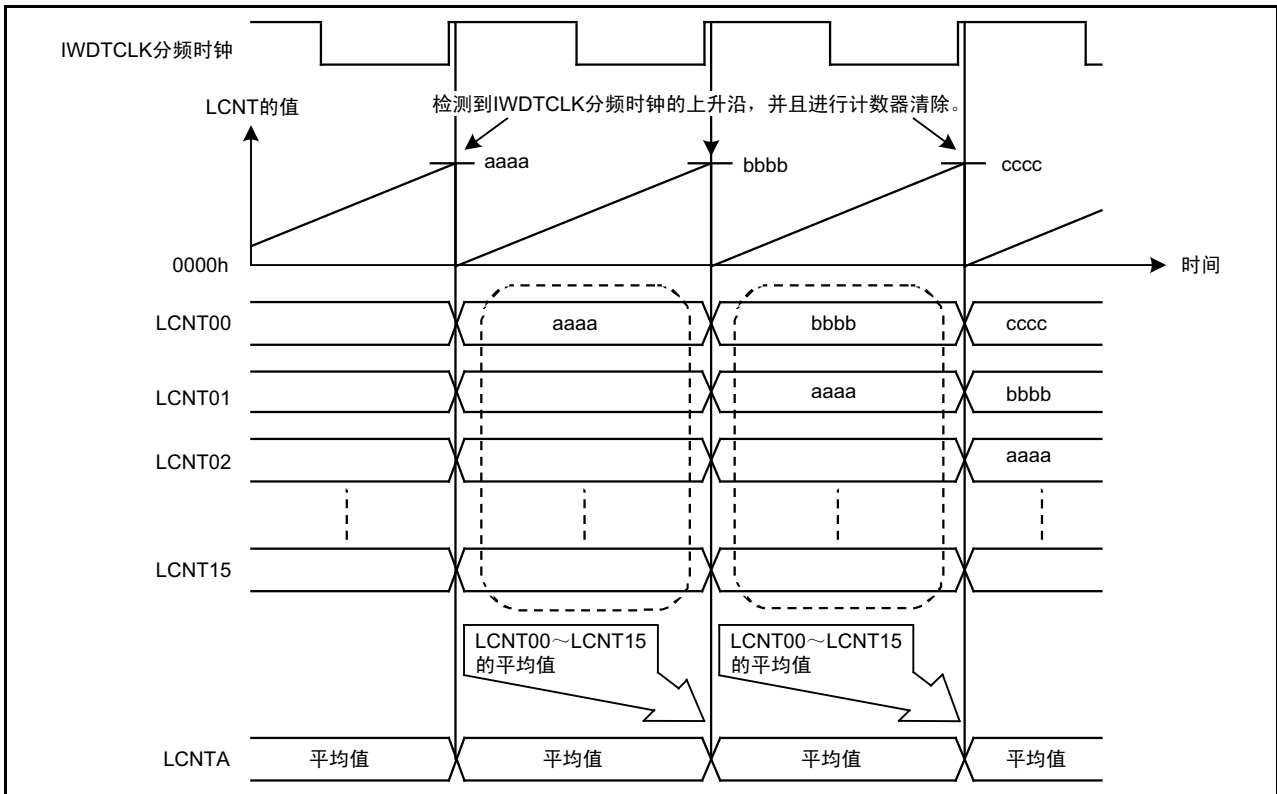


图 23.77 IWDTCCLK 计数功能的运行例子

<p>计数时钟的选择 通过LCCR.LPSC[1:0]位选择IWDTCCLK分频时钟。 通过LCCR.TPSC[1:0]位选择计数时钟。</p>
<p>IWDT的设置 使IWDT运行。(详细内容请参照“26. 独立看门狗定时器 (IWDTa)”) 给LCCR.LPSC[1:0]位设定“00b”以外的值时, 必须将IWDT.IWDTCR.CKS[3:0]位设定为“00xx”以外的值。</p>
<p>计数结果寄存器的初始化 给LCNT00~LCNT15寄存器设定初始值。 如果将LCCR.LCNTS位置“1”, LCNT00寄存器的值就被设定到LCNT01~LCNT15寄存器。</p>
<p>计数开始 将LCCR.LCNTE位置“1”, 如果检测到IWDTCCLK分频时钟的上升沿, 就开始计数。</p>

图 23.78 IWDTCCLK 计数功能的设定例子

能减少 IWDTCCLK 分频时钟上升沿中断请求, 此时, 计数结果也减少。通过 LCCR.LCTO[2:0] 位选择减少次数为 7 次、15 次、127 次或者 255 次。通过 LCCR.LCNTAT 位选择是否减少计数结果。

不减少计数结果时的运行例子如图 23.79、减少计数结果时的运行例子如图 23.80 所示。

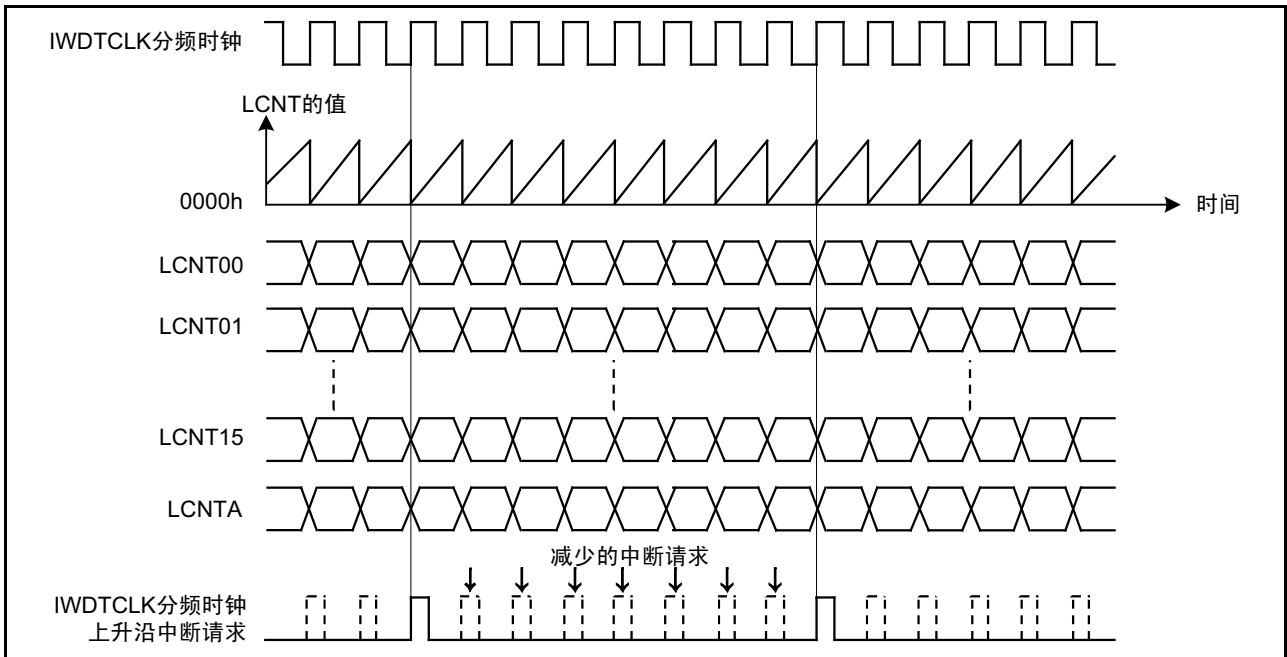


图 23.79 IWDTCCLK 计数减少功能的运行例子 (减少次数为 7 次、不减少计数结果)

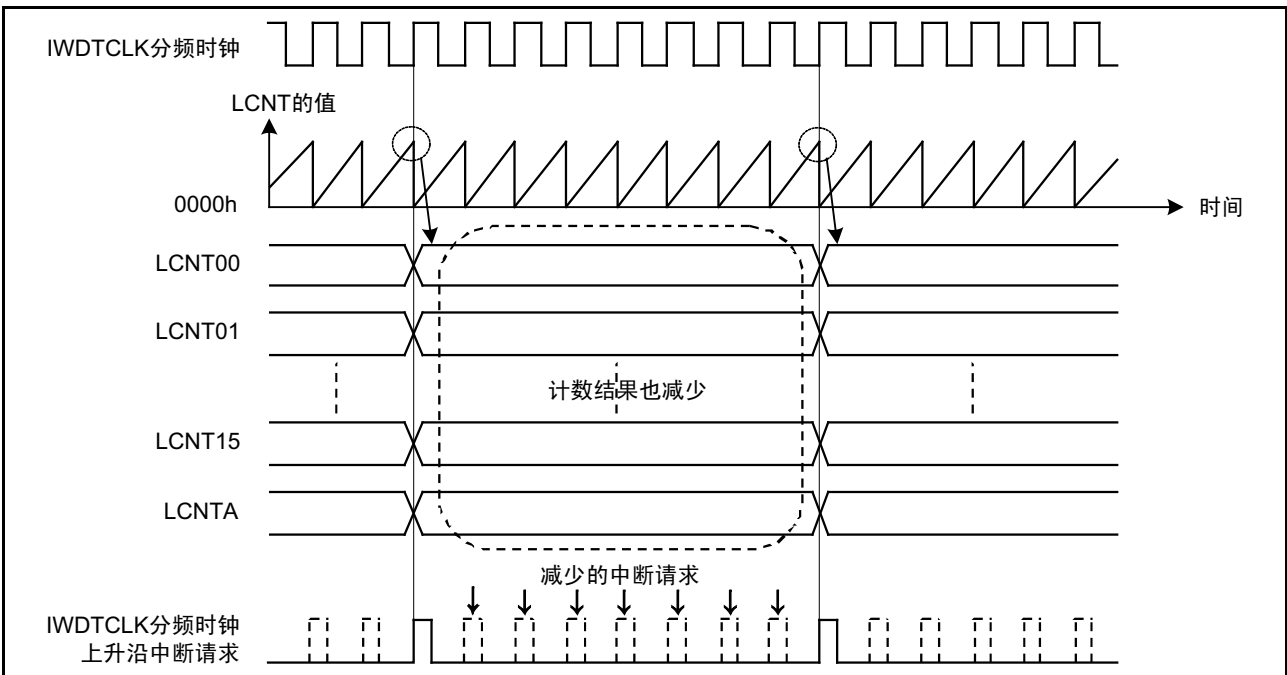


图 23.80 IWDTCCLK 计数减少功能的运行例子 (减少次数为 7 次、减少计数结果)

23.7 保护功能

23.7.1 寄存器的写保护

为了防止误写寄存器，能通过设定 GTWP.WPn (n=0 ~ 3) 位给每个通道设定禁止写寄存器。
禁止写的寄存器如下所示：

GPTn.GTIOR、GPTn.GTINTAD、GPTn.GTCR、GPTn.GTBER、GPTn.GTUDC、GPTn.GTITC、
GPTn.GTST、GPTn.GTCNT、GPTn.GTCCRA、GPTn.GTCCRB、GPTn.GTCCRC、GPTn.GTCCRD、
GPTn.GTCCRE、GPTn.GTCCRF、GPTn.GTPR、GPTn.GTPBR、GPTn.GTPDBR、GPTn.GTADTRA、
GPTn.GTADTBRA、GPTn.GTADTDBRA、GPTn.GTADTRB、GPTn.GTADTBRB、GPTn.GTADTDBRB、
GPTn.GTONCR、GPTn.GTDTCR、GPTn.GTDVU、GPTn.GTDVD、GPTn.GTDBU、GPTn.GTDBD、
GPTn.GTSOS、GPTn.GTSOS、GPTn.GTSOTR

23.7.2 缓冲运行的抑制

当缓冲寄存器的写操作赶不上缓冲传送时序时，能通过设定 GTBDR 寄存器禁止缓冲运行。如果在写缓冲寄存器前将 GTBDR 寄存器的对应位置“1”（禁止缓冲运行），在全部缓冲寄存器的写操作结束后将 GTBDR 寄存器的对应位置“0”（允许缓冲运行），即使在写缓冲寄存器的过程中产生缓冲传送条件，也能暂时禁止缓冲传送。

缓冲运行的抑制例子如图 23.81 所示。

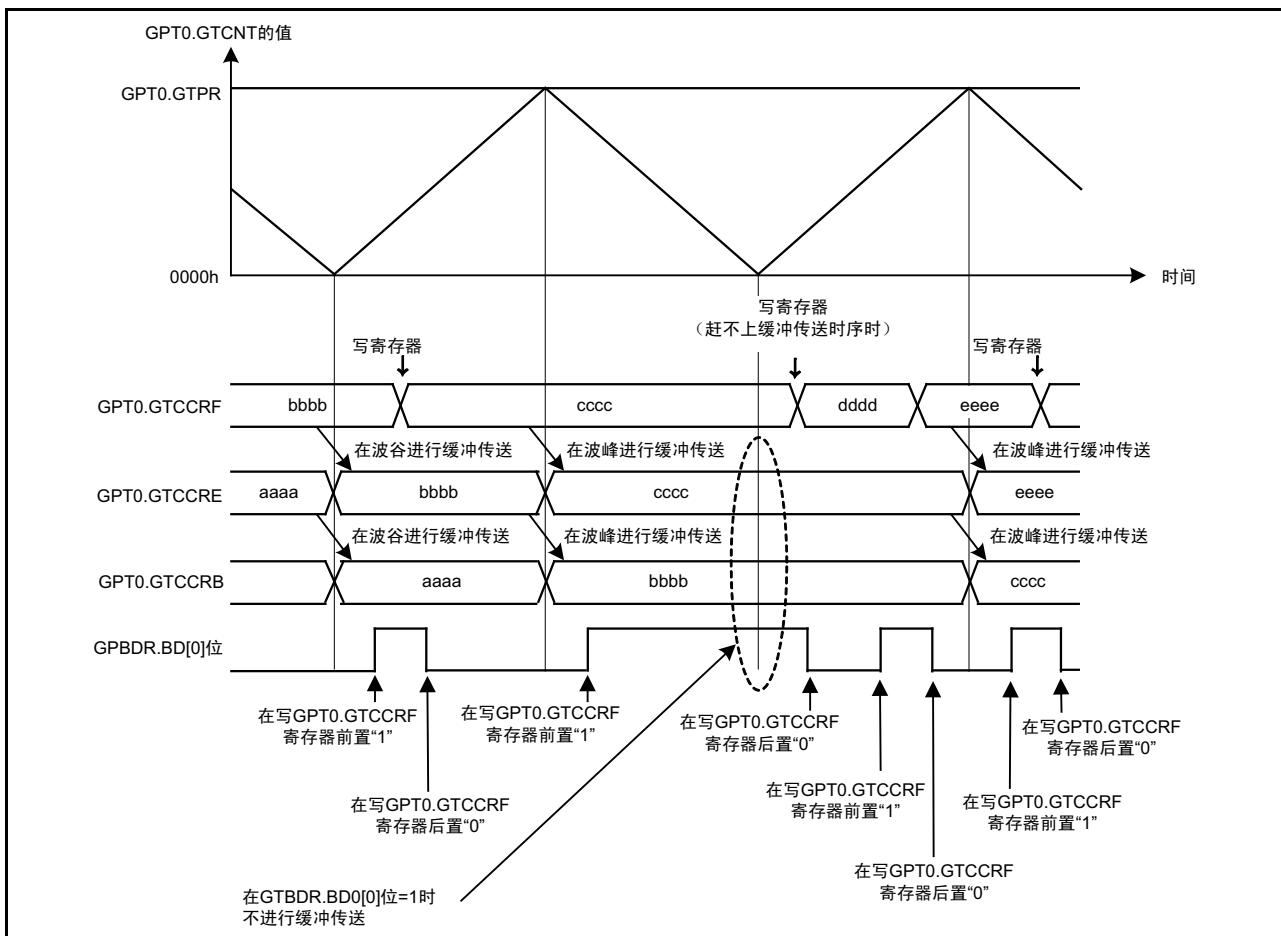


图 23.81 缓冲运行的抑制例子（三角波、双缓冲运行、在波谷和波峰都进行缓冲传送）

23.7.3 GTIOC 引脚输出的无效控制

为了在系统发生异常时进行保护，能通过设定 GTONCR 寄存器对 GTIOC 引脚输出进行无效控制（无效电平）。无效控制源有 3 种，分别是比较器检测、GTETRIG 引脚输入、写 GTONCR.SWN 位。

GTIOC 引脚输出的无效控制运行例子如图 23.82 所示。

在进行无效控制时，在同一周期内，即使无效控制源消失，无效控制也不被解除，而是在下一个周期被解除。

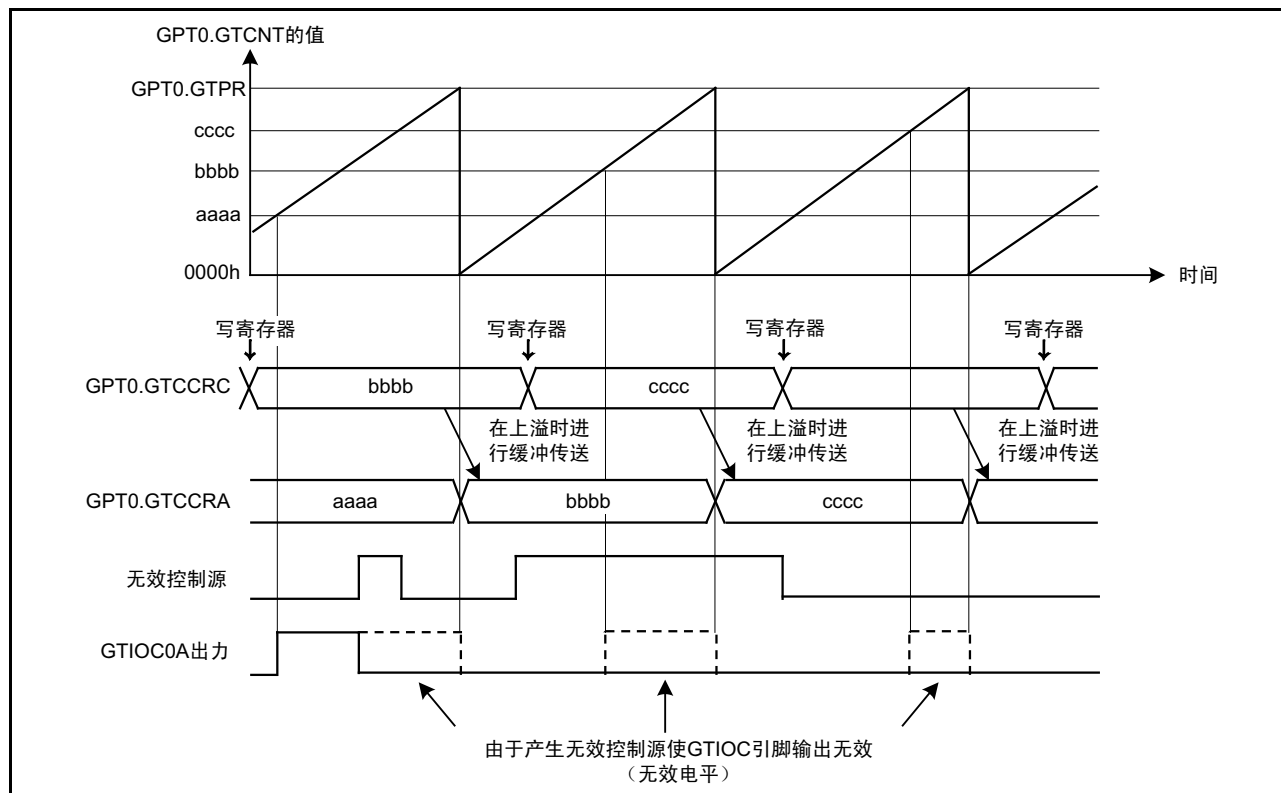


图 23.82 GTIOC 引脚输出的无效控制运行例子

（在锯齿波进行递增计数、缓冲运行、有效电平为 High 电平、通过 GTCCRA 的比较匹配进行 High 电平输出、在周期结束时进行 Low 电平输出）

23.7.4 GTIOC 引脚输出的输出保护功能

在给 GTCCRA 寄存器设定了异常值 (不在 $0 < GTCCRA < GTPR$ 范围的值) 时, 如果在三角波 PWM 模式中自动设定了死区时间 (GDTDCR.TDE 位为“1”), GTIOC 引脚输出的输出保护功能 (抑制功能) 就运行。

能通过读 GTSOS.SOS[1:0] 位确认输出保护功能的运行状态。

输出保护功能的状态转移如图 23.83 所示。

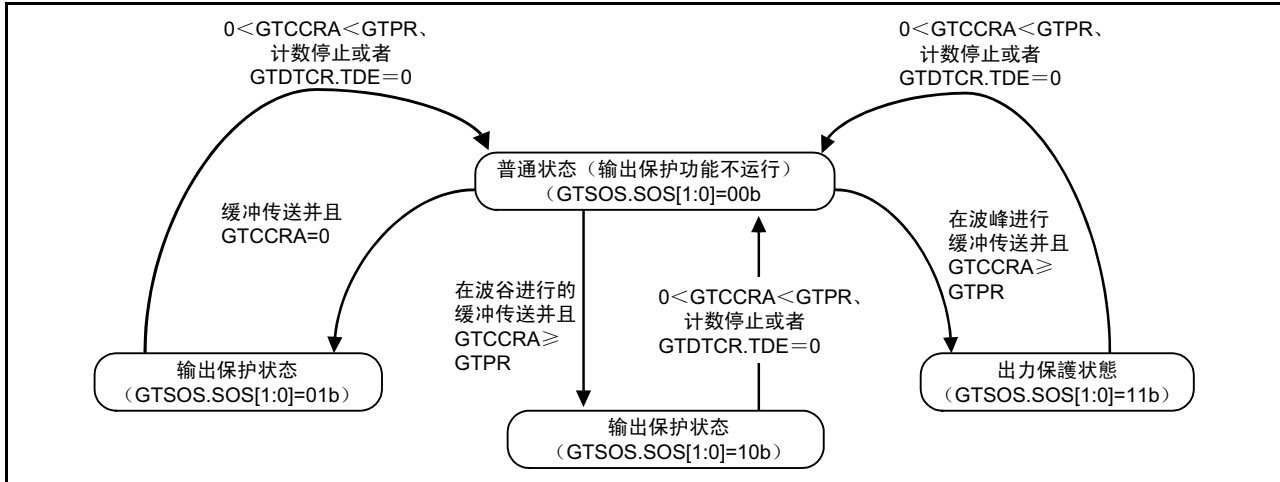


图 23.83 输出保护功能

(1) 在进行缓冲传送并且 GTCCRA 为“0”时的输出保护功能

在波谷进行缓冲传送并且 GTCCRA 为“0”时的输出保护功能的运行例子如图 23.84、图 23.85 所示, 在波峰进行缓冲传送并且 GTCCRA 为“0”时的输出保护功能的运行例子如图 23.86、图 23.87 所示。

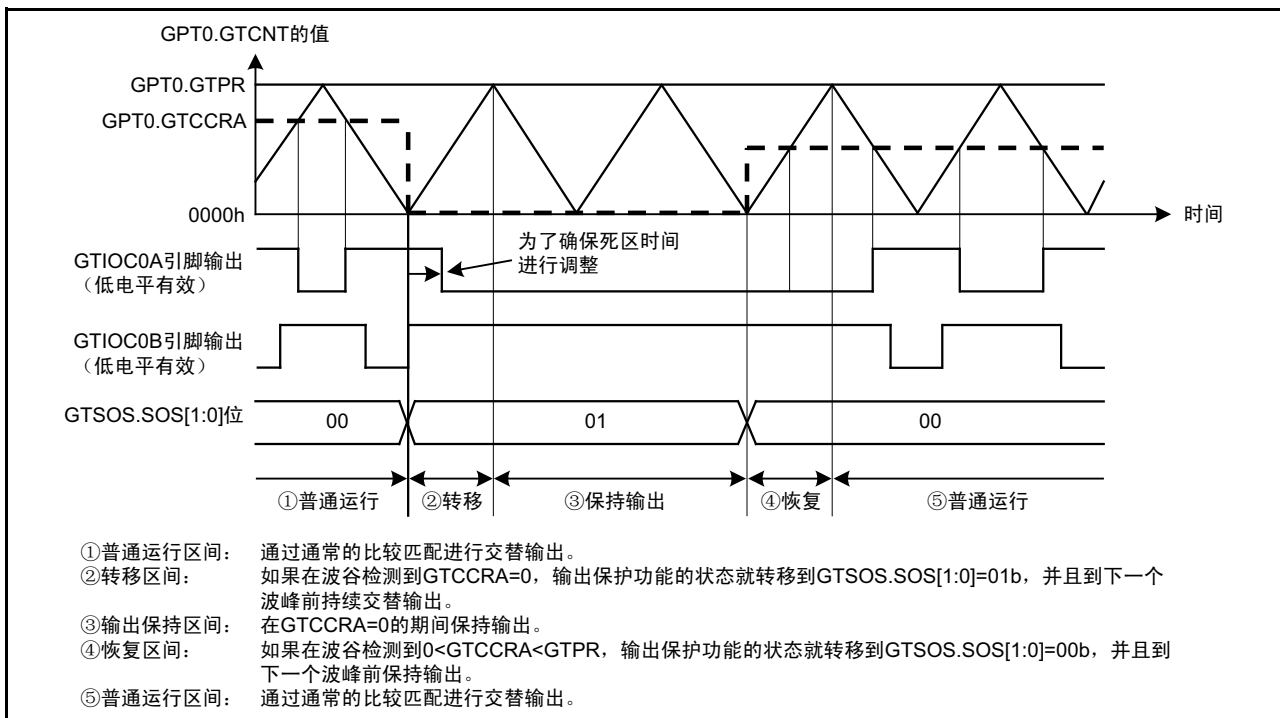


图 23.84 在波谷进行缓冲传送并且 GTCCRA 为“0”时的输出保护功能的运行例子 (在波谷进行缓冲传送时恢复 $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

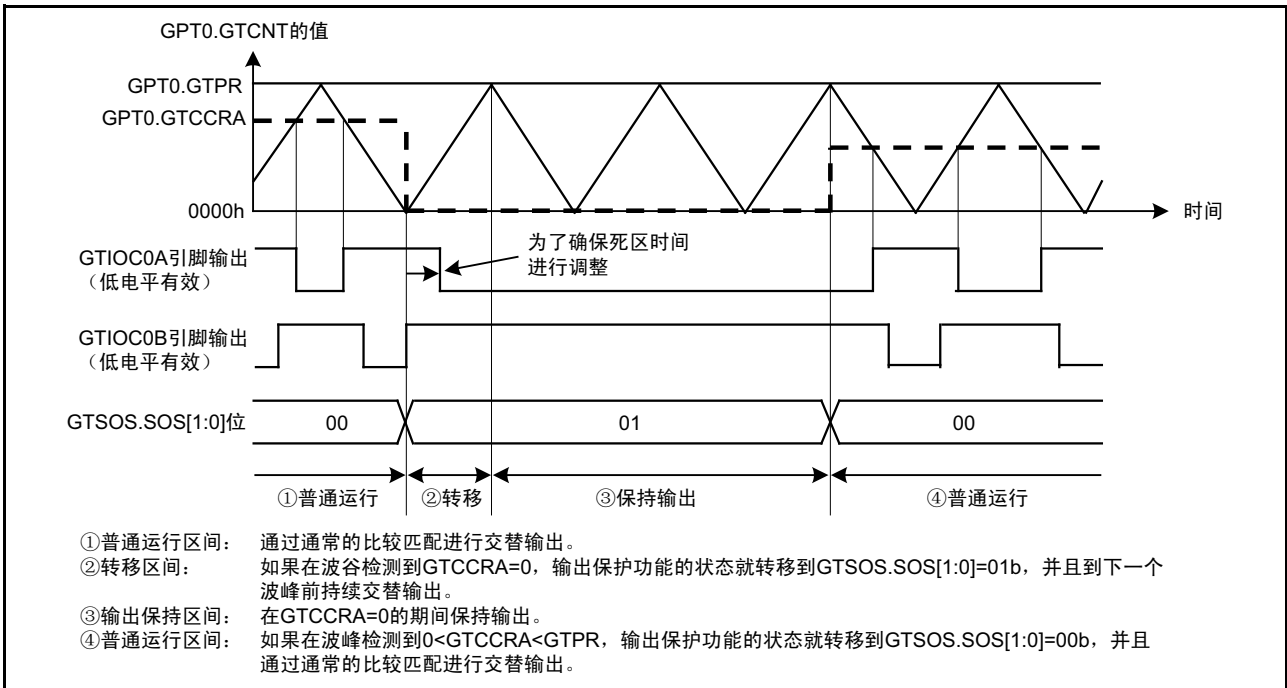


图 23.85 在波谷进行缓冲传送并且 GTCCRA 为 “0” 时的输出保护功能的运行例子
 (在波峰进行缓冲传送时恢复 $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

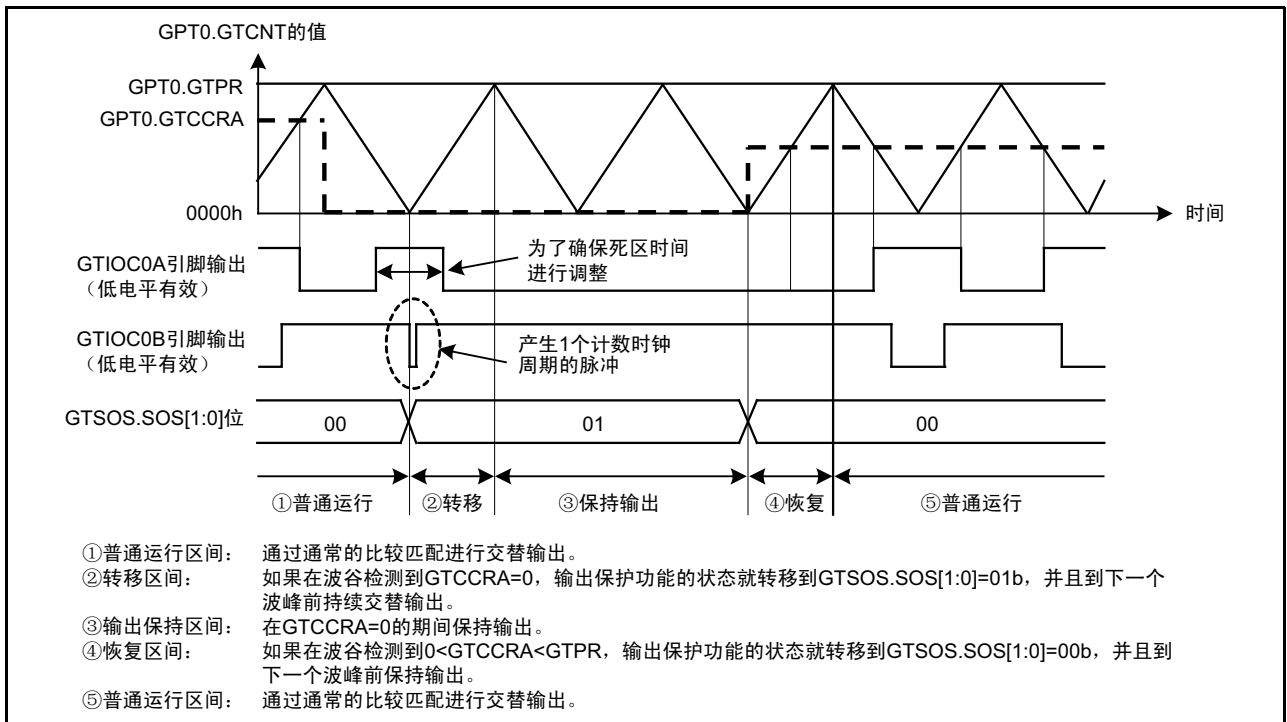


图 23.86 在波峰进行缓冲传送并且 GTCCRA 为 “0” 时的输出保护功能的运行例子
 (在波谷进行缓冲传送时恢复 $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

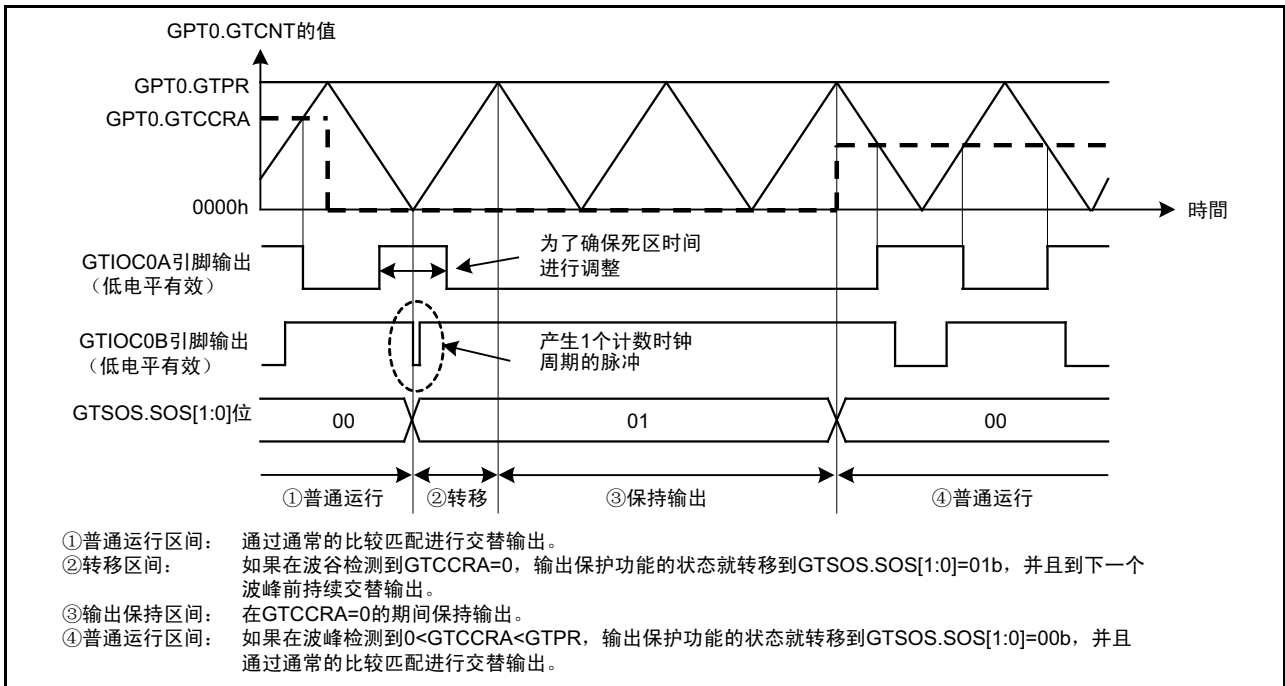


图 23.87 在波峰进行缓冲传送并且 GTCCRA 为 “0” 时的输出保护功能的运行例子
 （在波峰进行缓冲传送时恢复 $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平）

(2) 在波谷进行缓冲传送并且设定为 $GTCCRA \geq GTPR$ 时的输出保护功能

在波谷进行缓冲传送并且设定为 $GTCCRA \geq GTPR$ 时的输出保护功能的运行例子如图 23.88、图 23.89 所示。

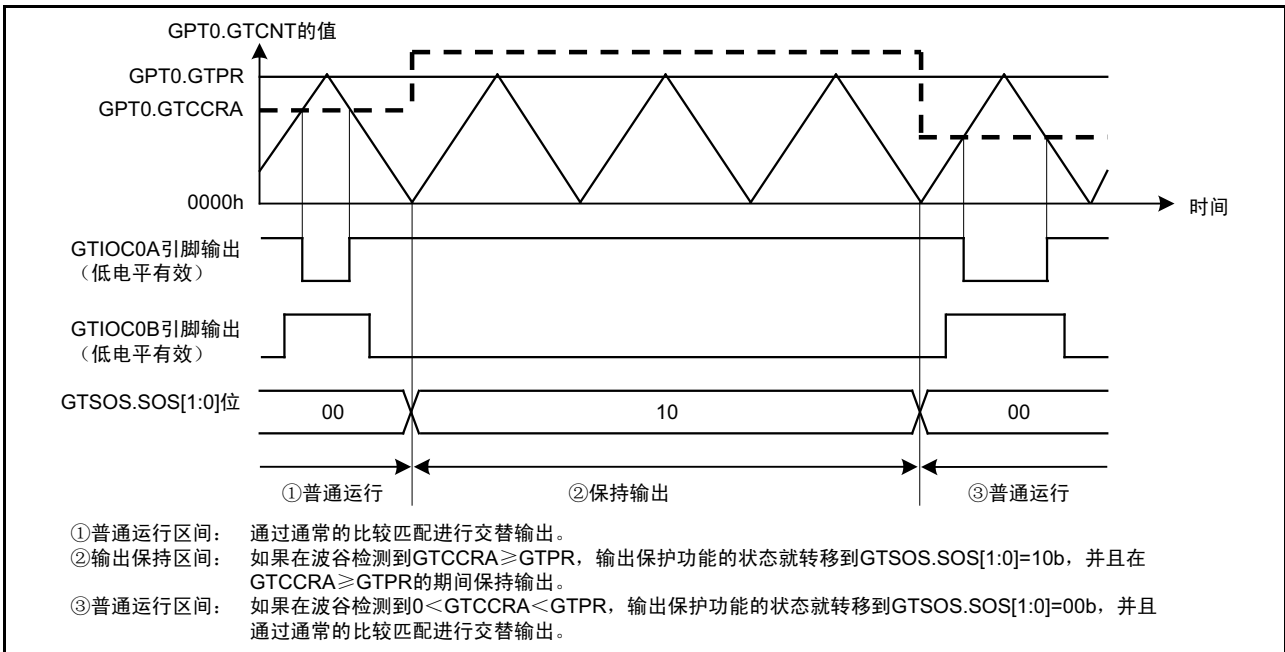


图 23.88 在波谷进行缓冲传送并且设定为 $GTCCRA \geq GTPR$ 时的输出保护功能的运行例子
 （在波谷进行缓冲传送时恢复 $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平）

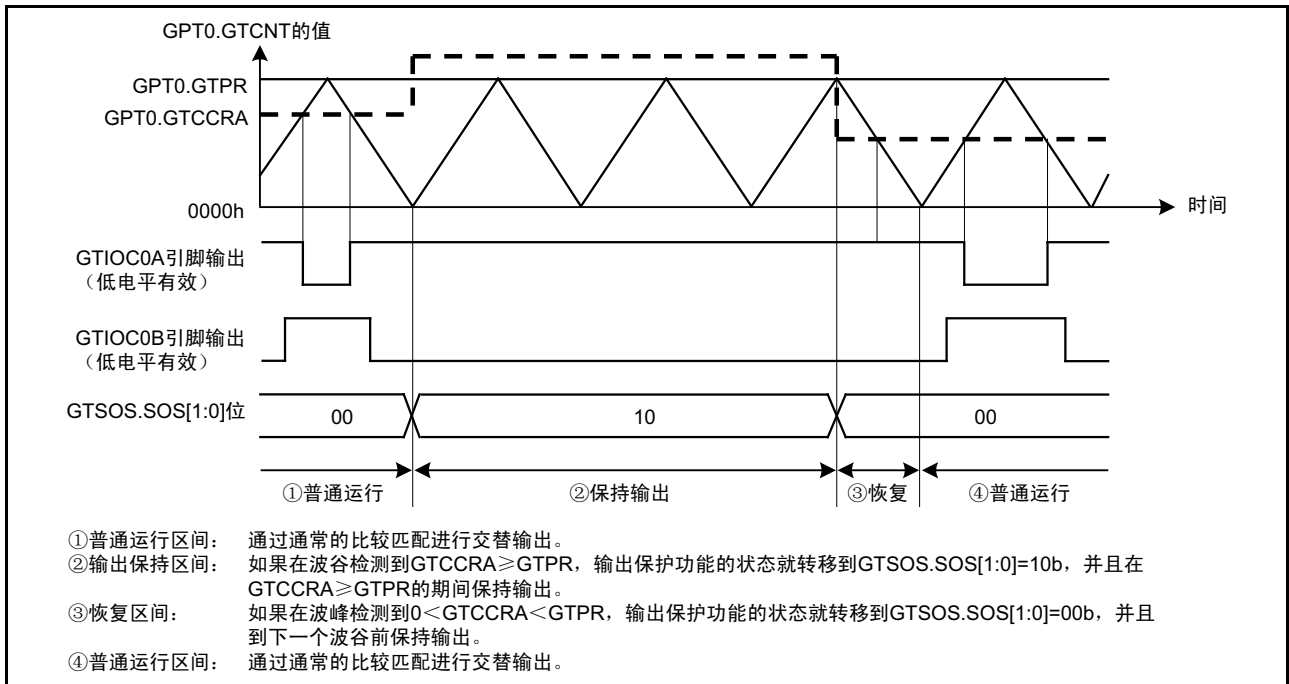


图 23.89 在波谷进行缓冲传送并且设定为 $GTCCRA \geq GTPR$ 时的输出保护功能的运行例子
 (在波峰进行缓冲传送时恢复 $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

(3) 在波峰进行缓冲传送并且设定为 $GTCCRA \geq GTPR$ 时的输出保护功能

在波峰进行缓冲传送并且设定为 $GTCCRA \geq GTPR$ 时的输出保护功能的运行例子如图 23.90、图 23.91 所示。

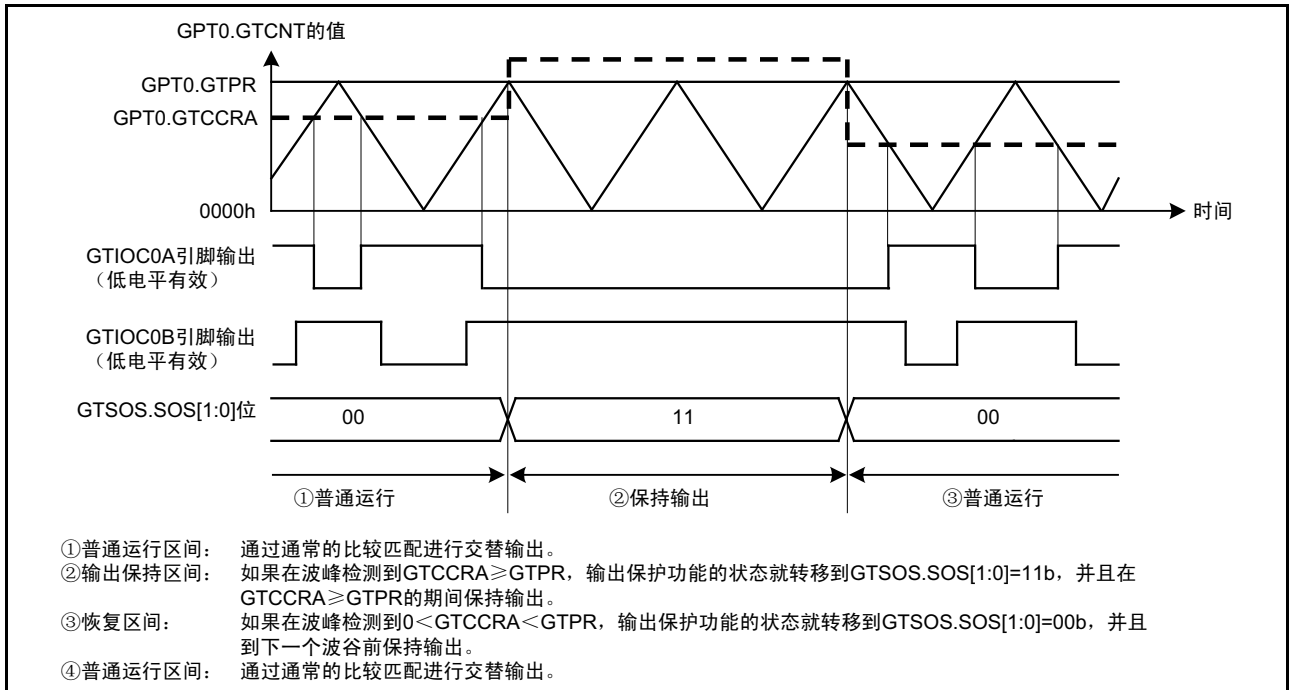


图 23.90 在波峰进行缓冲传送并且设定为 $GTCCRA \geq GTPR$ 时的输出保护功能的运行例子
 (在波峰进行缓冲传送时恢复 $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

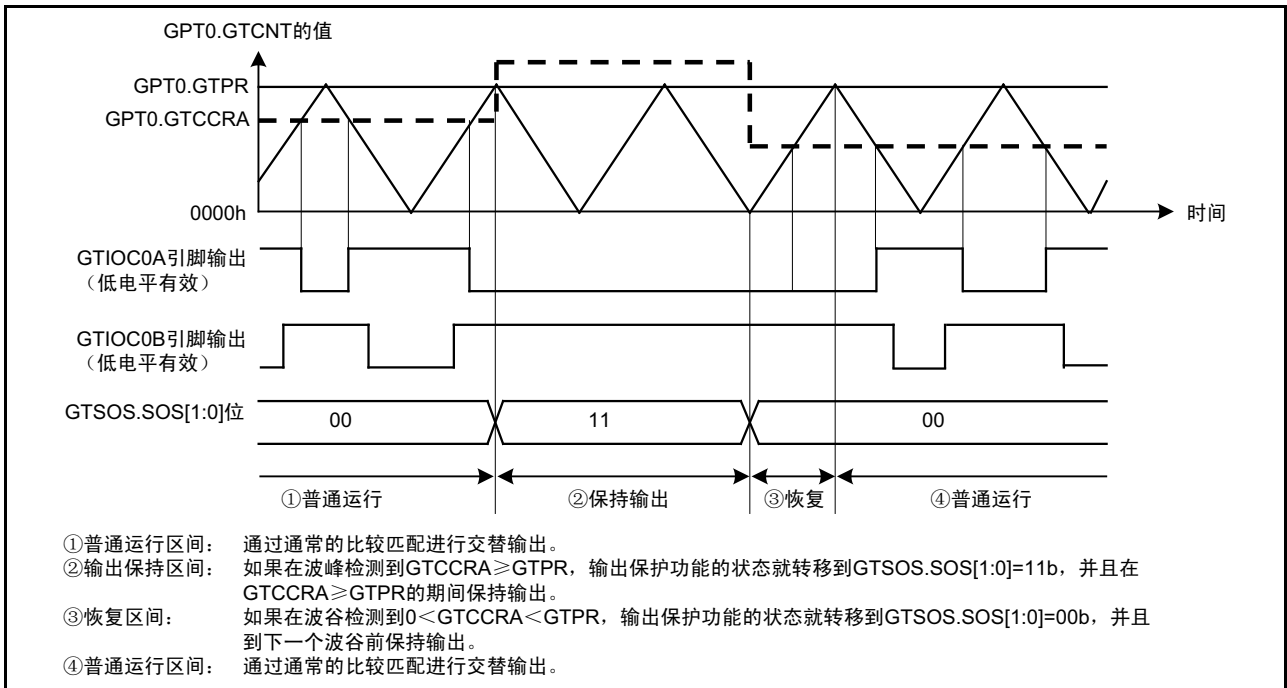


图 23.91 在波谷进行缓冲传送并且设定为 $GTCCRA \geq GTPR$ 时的输出保护功能的运行例子 (在波谷进行缓冲传送时恢复 $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

(4) 输出保护功能的注意事项

在将 GTCCRA 寄存器设定在 $0 < GTCCRA < GTPR$ 范围内的状态下, 开始计数。

即使在计数时给 GTCCRA 寄存器设定了异常值 (不在 $0 < GTCCRA < GTPR$ 范围的值), 通过输出保护功能也能使正相或者反相输出为无效输出。但是, 如果在给 GTCCRA 寄存器设定了异常值的状态下开始计数, 输出保护功能就无法正常运行。

23.7.5 通过 POE 功能进行 GTIOC 引脚输出的高阻抗控制

作为系统发生异常时的保护措施, 能通过端口输出运行 (POE) 功能, 进行 GTIOC 引脚输出的高阻抗控制。

详细内容请参照“22. 端口输出允许 3 (POE3)”。

23.8 输出引脚的初始化方法

23.8.1 复位后的引脚设定

GPT 的寄存器在复位时被初始化。必须在进行端口模式的选择设定 (PMR 寄存器、PmnPFS 寄存器), GTIOR 寄存器、GTONCR.OAE 位和 GTONCR.OBE 位的设定, 以及将 GPT 功能输出到外部引脚后, 开始计数。

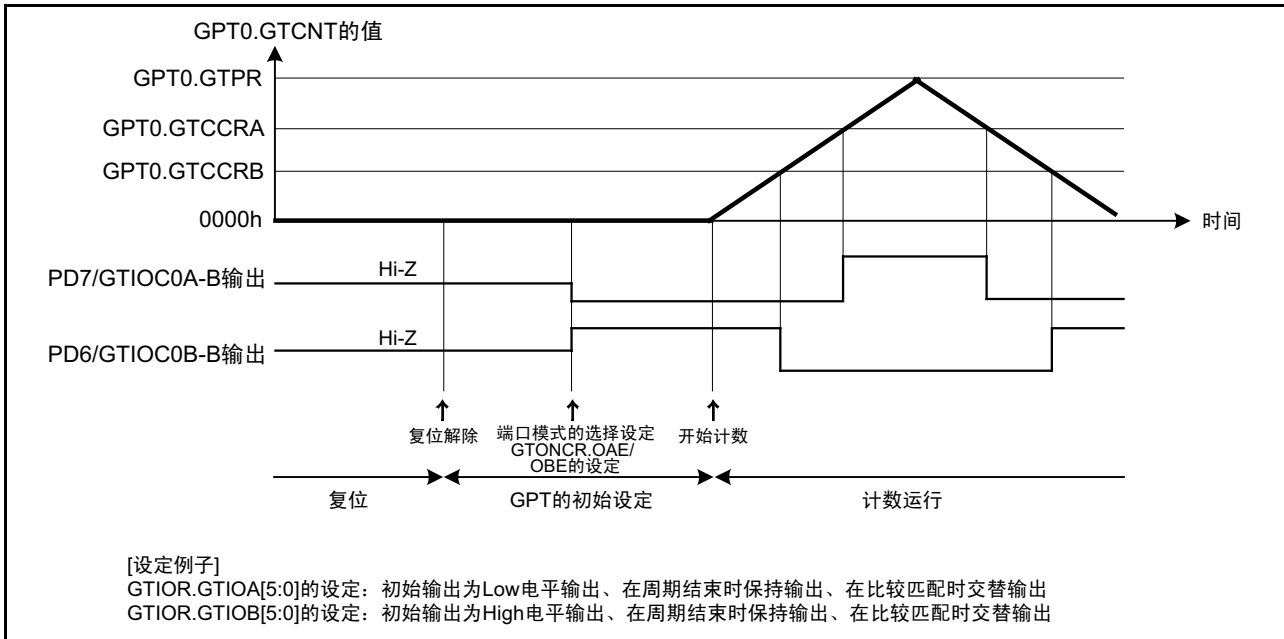


图 23.92 复位后的引脚设定例子

23.8.2 因运行中的异常等引起的引脚初始化

从 GPT 的运行过程中发生异常等开始到引脚初始化前的引脚处理, 有以下方法:

1. 将 GTIOR 寄存器的 OAHLD 位和 OBHLD 位置“1”, 在停止计数时保持输出。
2. 将 GTIOR 寄存器的 OAHLD 位和 OBHLD 位置“0”, 给 GTIOR 寄存器的 OADFLT 位和 OBDFLT 位设定任意的输出值, 在停止计数时输出任意的值。
3. 与 MTU3 同样, 事先通过 I/O 端口的 PDR 寄存器、PODR 寄存器和 PMR 寄存器设定为“在通用输出端口时输出任意的值”, 在发生异常时将 GTONCR 寄存器的 OAE 位和 OBE 位置“0”, 将 PMR 寄存器的对应引脚的控制位置“0”, 并且将引脚作为通用输出端口输出任意的值。
4. 使用端口输出允许 3 (POE3) 的 POE 功能, 将输出置为高阻抗。

在自动设定死区时间时, 必须在停止计数后将 GTDTCR.TDE 位置“0”。

在停止计数时, 只有因 GPT 以外的外部源变化的寄存器发生变化, 其他寄存器不变。在重新开始计数后, 持续运行。

在停止计数时, 只有在将各寄存器初始化后才能重新开始计数。

23.9 使用时的注意事项

23.9.1 模块停止功能的设定

能通过模块停止控制寄存器设定禁止或者运行 GPT 的运行。在设定为初始值时，GPT 停止运行。能通过解除模块停止状态存取寄存器。详细内容请参照“11. 低功耗功能”。

23.9.2 比较匹配运行时的 GTCCRn 寄存器的设定 (n=A、B、C、D、E、F)

(1) 在三角波 PWM 模式中自动设定死区时间的情况

必须将 GTCCRA 寄存器设定在 $GTDVU < GTCCRA$ 、 $GTDVD < GTCCRA$ 、 $GTCCRA < GTPR$ 的范围内。

如果在计数运行时设定 $GTCCRA=0$ 或者 $GTCCRA \geq GTPR$ ，输出保护功能就运行。

必须在将 GTCCRA 寄存器设定在 $0 < GTCCRA < GTPR$ 的范围内状态下，开始计数。如果在设定在 $0 < GTCCRA < GTPR$ 的外围外的状态下开始计数，输出保护功能就无法正常运行。

详细内容请参照“23.7.4 GTIOC 引脚输出的输出保护功能”。

(2) 在三角波 PWM 模式中不自动设定死区时间的情况

必须将 GTCCRA 寄存器设定在 $0 < GTCCRA < GTPR$ 的范围内。如果设定为 $GTCCRA=0$ 或者 $GTCCRA=GTPR$ ，只有在 $GTCCRA=0$ 或者 $GTCCRA=GTPR$ 成立时在周期内发生比较匹配。另外，如果设定为 $GTCCRA > GTPR$ ，就不发生比较匹配。

同样，必须将 GTCCRB 寄存器设定在 $0 < GTCCRB < GTPR$ 的范围内。如果设定为 $GTCCRB=0$ 或者 $GTCCRB=GTPR$ ，只有在 $GTCCRB=0$ 或者 $GTCCRB=GTPR$ 成立时在周期内发生比较匹配。另外，如果设定为 $GTCCRB > GTPR$ ，就不发生比较匹配。

(3) 在锯齿波单触发脉冲模式中自动设定死区时间的情况

GTCCRC 寄存器和 GTCCRD 寄存器必须满足以下的限制。否则有可能无法获得确保了死区时间的正常输出波形。

- 递增计数时: $GTCCRC < GTCCRD$ 、 $GTCCRC > GTDVU$ 、 $GTCCRD < GTPR - GTDVD$
- 递减计数时: $GTCCRC > GTCCRD$ 、 $GTCCRC < GTPR - GTDVU$ 、 $GTCCRD > GTDVD$

同样，GTCCRE 寄存器和 GTCCRF 寄存器必须满足以下的限制。否则有可能无法获得确保了死区时间的正常输出波形。

- 递增计数时: $GTCCRE < GTCCRF$ 、 $GTCCRE > GTDVU$ 、 $GTCCRF < GTPR - GTDVD$
- 递减计数时: $GTCCRE > GTCCRF$ 、 $GTCCRE < GTPR - GTDVU$ 、 $GTCCRF > GTDVD$

(4) 在锯齿波单触发脉冲模式中自动设定死区时间的情况

GTCCRC 寄存器和 GTCCRD 寄存器必须满足以下的限制。否则比较匹配就不发生 2 次，也无法获得脉冲输出。

- 递增计数时： $0 < GTCCRC < GTCCRD < GTPR$
- 递减计数时： $GTPR > GTCCRC > GTCCRD > 0$

同样，GTCCRE 寄存器和 GTCCRF 寄存器必须满足以下的限制。否则比较匹配就不发生 2 次，也无法获得脉冲输出。

- 递增计数时： $0 < GTCCRE < GTCCRF < GTPR$
- 递减计数时： $GTPR > GTCCRE > GTCCRF > 0$

(5) 在锯齿波 PWM 模式的情况

必须将 GTCCRA 寄存器设定在 $0 < GTCCRA < GTPR$ 的范围内。如果设定为 $GTCCRA=0$ 或者 $GTCCRA=GTPR$ ，只有在 $GTCCRA=0$ 或者 $GTCCRA=GTPR$ 成立时在周期内发生比较匹配。另外，如果设定为 $GTCCRA > GTPR$ ，就不发生比较匹配。

同样，必须将 GTCCRB 寄存器设定在 $0 < GTCCRB < GTPR$ 的范围内。如果设定为 $GTCCRB=0$ 或者 $GTCCRB=GTPR$ ，只有在 $GTCCRB=0$ 或者 $GTCCRB=GTPR$ 成立时在周期内发生比较匹配。另外，如果设定为 $GTCCRB > GTPR$ ，就不发生比较匹配。

23.9.3 安全停止定时器的方法

当通过写 GTSTR 寄存器来停止定时器与 GPT 的比较匹配中断时序发生竞争时，有可能在写 GTSTR 寄存器后发生中断。

如果按照以下的步骤停止定时器，就能安全的使定时器停止运行，并且在停止后不发生比较匹配中断。

1. 通过 ICU 的中断请求允许寄存器 (IER15~IER18) 禁止中断请求。
2. 通过 GPT 的中断输出设定寄存器 (GTINTAD) 禁止中断请求。
3. 将 GTSTR 寄存器的 CSTn 位置“0”。

23.9.4 使用 IWDTCCLK 计数功能时的低功耗功能的设定

在 IWDTCCLK 计数运行时，通过模块停止控制寄存器或者待机控制寄存器使 GPT 在停止运行后重新开始运行时，必须在 GPT 停止运行前设定 LCCR.CNTE=0，停止 IWDTCCLK 计数。在 GPT 重新开始运行后设定 LCCR.LCNTE=1，重新开始 IWDTCCLK 计数。模块停止控制寄存器和待机控制寄存器的详细内容请参照“11. 低功耗功能”。

24. 比较匹配定时器 (CMT)

RX63T 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，能按设定的各周期发生中断。

24.1 概要

CMT 的规格如表 24.1 所示。

CMT (单元 0) 的框图如图 24.1 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 24.1 CMT 的规格

项目	功能
计数时钟	<ul style="list-style-type: none"> 4 种分频时钟 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。

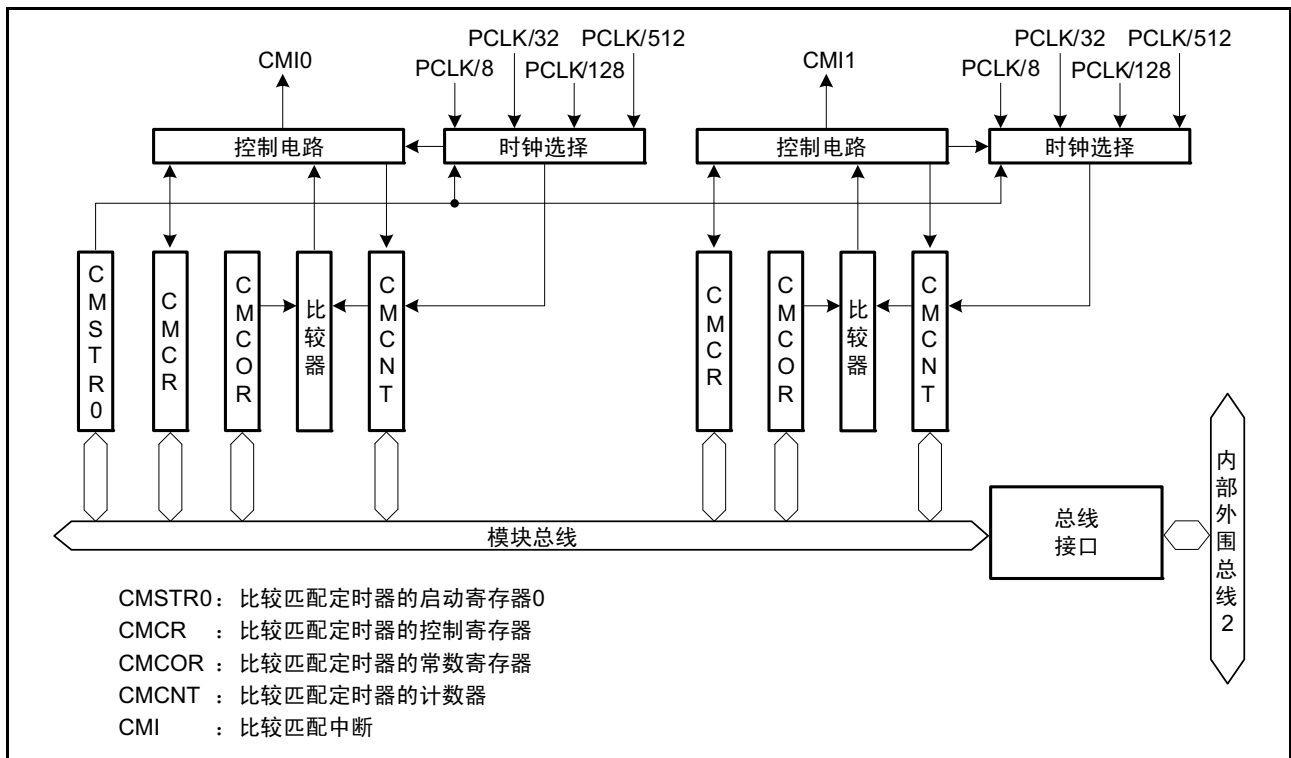


图 24.1 CMT (单元 0) 的框图

24.2 寄存器说明

24.2.1 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR1	STR0
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	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

24.2.2 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	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

24.2.3 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	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 (CMIn) 1: 允许比较匹配中断 (CMIn)	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

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 种分频时钟中选择输入到 CMCNT 计数器的计数时钟。

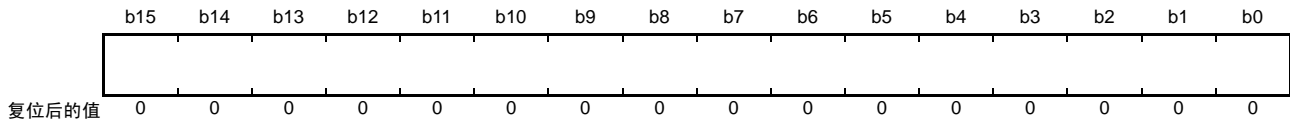
如果将 CMSTRm.STRn 位 (m=0、1, n=0~3) 置“1”, 对应的 CMCNT 计数器就通过 CKS[1:0] 位选择的时钟开始递增计数。

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 (CMIn) (n=0~3) 的发生。

24.2.4 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



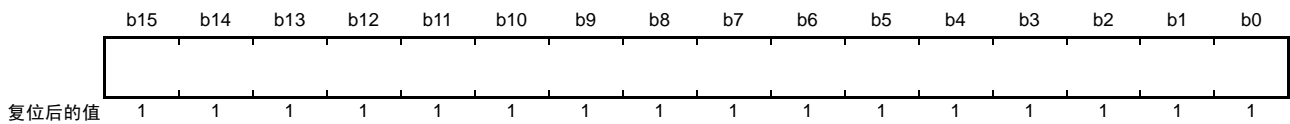
CMCNT 计数器是用于产生中断请求的可读写递增计数器。

如果通过 CMCR.CKS[1:0] 位选择分频时钟并且将 CMSTRm.STRn 位 (m=0、1, n=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 (CMI_n) (n=0 ~ 3)。

24.2.5 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 计数器比较匹配的周期的可读写寄存器。

24.3 运行说明

24.3.1 周期计数

如果通过 CMC.R.CKS[1:0] 位选择分频时钟并且将 CMSTRm.STRN 位 (m=0、1, n=0 ~ 3) 置“1”，CMCNT 计数器就通过所选时钟开始递增计数。

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”，并且发生比较匹配中断 (CMIn) (n=0 ~ 3)。CMCNT 计数器从“0000h”重新开始递增计数，CMCNT 计数器的运行如图 24.2 所示。

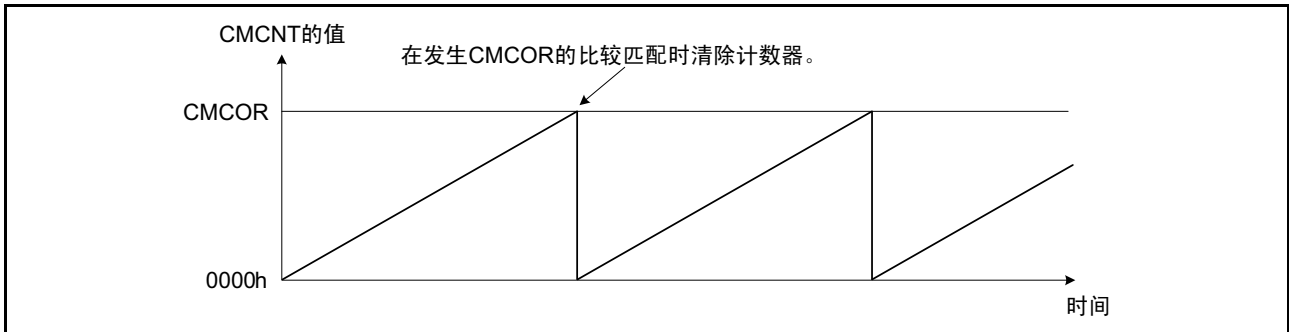


图 24.2 CMCNT 计数器的运行

24.3.2 CMCNT 计数器的计数时序

能通过 CMC.R.CKS[1:0] 位从外围模块时钟 (PCLK) 分频后的 4 种分频时钟 (PCLK/8、PCLK/32、PCLK/128、PCLK/512) 中选择输入到 CMCNT 计数器的计数时钟，此时的 CMCNT 计数器的计数时序如图 24.3 所示。

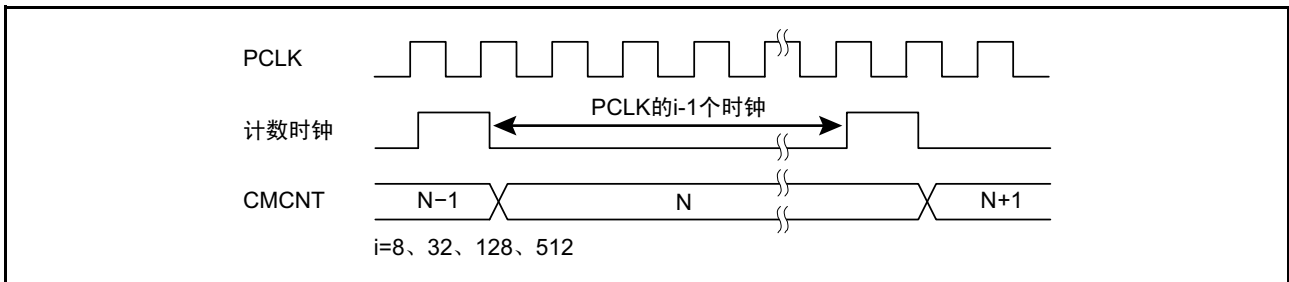


图 24.3 CMCNT 计数器的计数时序

24.4 中断

24.4.1 中断源

CMT 的各通道有比较匹配中断 (CMI_n) (n=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“14. 中断控制器 (ICUb)”。

表 24.2 CMT 的中断源

名称	中断源	DTC 的启动	DMAC 的启动
CMI0	CMT0.CMCNT 和 CMT0.CMCOR 的比较匹配	能	能
CMI1	CMT1.CMCNT 和 CMT1.CMCOR 的比较匹配	能	能
CMI2	CMT2.CMCNT 和 CMT2.CMCOR 的比较匹配	能	能
CMI3	CMT3.CMCNT 和 CMT3.CMCOR 的比较匹配	能	能

24.4.2 比较匹配中断的发生时序

在 CMCNT 计数器的值和 CMCOR 寄存器的值相同时，发生比较匹配中断 (CMI_n) (n=0 ~ 3)。

在 CMCNT 计数器的值和 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 输入时钟前，不产生比较匹配信号。

比较匹配中断的发生时序如图 24.4 所示。

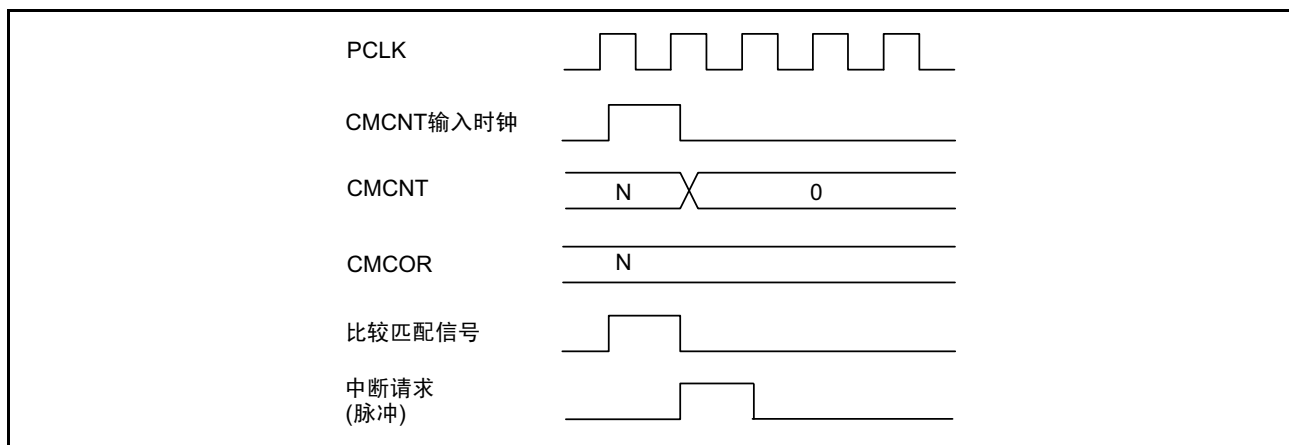


图 24.4 比较匹配中断的发生时序

24.5 使用时的注意事项

24.5.1 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 的运行，初始值时停止 CMT 的运行。能通过解除模块停止状态，存取寄存器。详细内容请参照“11. 低功耗功能”。

24.5.2 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 24.5 所示。

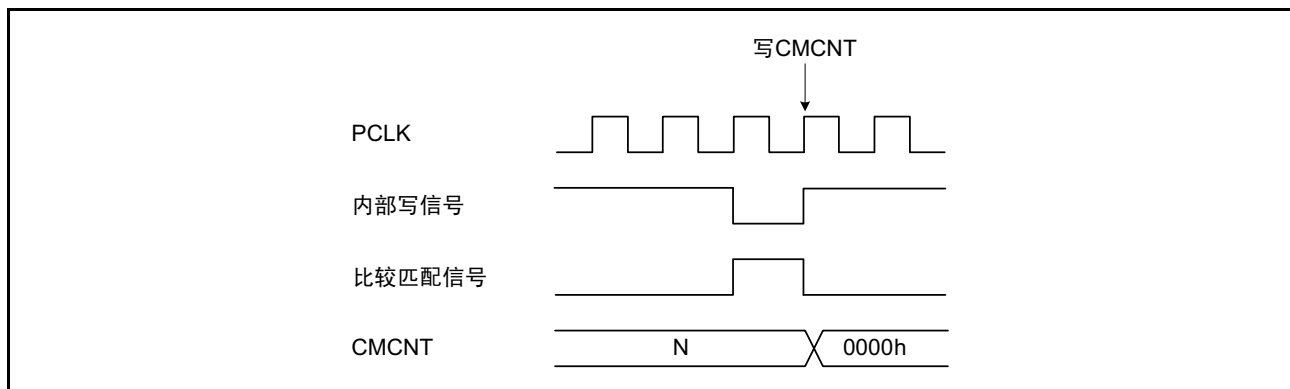


图 24.5 CMCNT 计数器的写和比较匹配的竞争

24.5.3 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 24.6 所示。

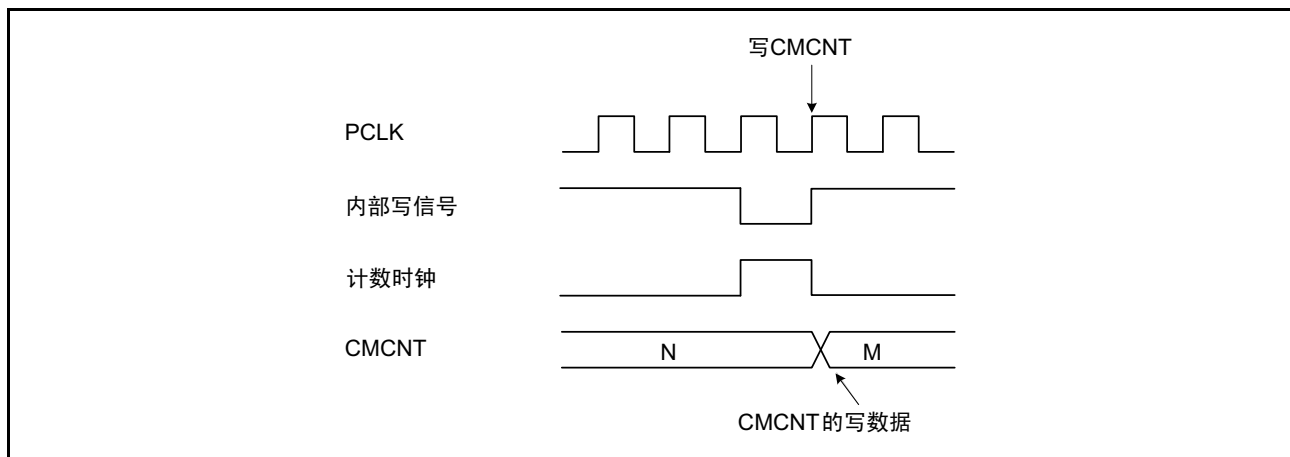


图 24.6 CMCNT 计数器的写和递增计数的竞争

25. 看门狗定时器 (WDTA)

看门狗定时器 (WDT) 检测程序的失控。

WDT 内置 14 位递减计数器，如果递减的计数值发生下溢，就能通过复位输出对本 LSI 进行复位，也能选择中断请求的发生。能通过刷新递减计数器的计数值，将计数值返回初始值并且重新开始计数。另外，能设定可刷新期间。如果在可刷新期间进行刷新，就能对计数器进行初始化并且再次进行计数；如果在可刷新期间外进行刷新，就输出复位或者中断请求。因此，能通过调整刷新间隔来检测程序失控。在发生下溢或者不在可刷新期间进行刷新后，WDT 停止计数。在寄存器启动模式中，在刷新后自动重新开始计数；在自动启动模式中，在输出复位或者中断请求后自动重新开始计数。

25.1 概要

WDT 有自动启动模式和寄存器启动模式两种模式，自动启动模式是在解除复位后自动开始计数的模式，寄存器启动模式是在解除复位后通过刷新（写寄存器）开始计数的模式。

在自动启动模式中，在解除复位前给选项功能选择寄存器 0 (OFS0) 设定时钟分频比、刷新的窗口起始 / 结束位置、超时期间、下溢时的复位输出 / 非屏蔽中断请求输出。

在寄存器启动模式中，在解除复位后并且在通过刷新开始计数前给寄存器设定时钟分频比、刷新的窗口起始 / 结束位置、超时期间、下溢时的复位输出 / 非屏蔽中断请求输出。

通过 WDT 启动模式选择位 (OFS0.WDTSTRT) 选择自动启动模式或者寄存器启动模式。

当选择自动启动模式时 (OFS0.WDTSTRT=0)，WDT 控制寄存器 (WDTCR) 和 WDT 复位控制寄存器 (WDTRCR) 的设定无效而 OFS0 寄存器的设定有效。

当选择寄存器启动模式时 (OFS0.WDTSTRT=1)，OFS0 寄存器的设定无效而 WDTCR 寄存器和 WDTRCR 寄存器的设定有效。

WDT 的规格和框图分别如表 25.1 和图 25.1 所示。

表 25.1 WDT 的规格 (1/2)

项目	内容
计数源	外围时钟 (PCLK)
时钟分频比	4 分频、64 分频、128 分频、512 分频、2048 分频、8192 分频
计数	通过 14 位递减计数器进行递减计数。
计数开始条件	<ul style="list-style-type: none"> 复位后自动开始计数 (自动启动模式)。 通过刷新 (在给 WDTRR 寄存器写 "00h" 后写 "FFh") 开始计数 (寄存器启动模式)。
计数停止条件	<ul style="list-style-type: none"> 复位 (递减计数器和寄存器返回初始值) 当发生下溢或者刷新错误时 重新开始计数 (自动启动模式: 自动、寄存器启动模式: 刷新)
窗口功能	能设定窗口起始 / 结束位置 (刷新允许 / 禁止期间)。
复位输出源	<ul style="list-style-type: none"> 当递减计数器发生下溢时 在刷新允许期间外进行刷新 (刷新错误) 时
中断请求输出源	<ul style="list-style-type: none"> 在递减计数器发生下溢时产生非屏蔽中断 (WUNI) 时 在刷新允许期间外进行刷新 (刷新错误) 时
读计数值	能通过读 WDTSR 寄存器，读递减计数器的计数值。
输出信号 (内部信号)	<ul style="list-style-type: none"> 复位输出 中断请求输出

表 25.1 WDT 的规格 (2/2)

项目	内容
自动启动模式 (由选项功能选择寄存器 0 (OFS0) 控制)	<ul style="list-style-type: none"> 选择复位后的时钟分频比 (OFS0.WDTCKS[3:0] 位)。 选择看门狗定时器的超时期限 (OFS0.WDTPOPS[1:0] 位)。 选择看门狗定时器的窗口起始位置 (OFS0.WDTRPSS[1:0] 位)。 选择看门狗定时器的窗口结束位置 (OFS0.WDTRPES[1:0] 位)。 选择复位输出或者中断请求输出 (OFS0.WDTRSTIRQS 位)。
寄存器启动模式 (由 WDT 寄存器控制)	<ul style="list-style-type: none"> 选择刷新后的时钟分频比 (WDTCR.CKS[3:0] 位)。 选择看门狗定时器的超时期限 (WDTCR.TOPS[1:0] 位)。 选择看门狗定时器的窗口起始位置 (WDTCR.RPSS[1:0] 位)。 选择看门狗定时器的窗口结束位置 (WDTCR.RPES[1:0] 位)。 选择复位输出或者中断请求输出 (WDTCR.RSTIRQS 位)。

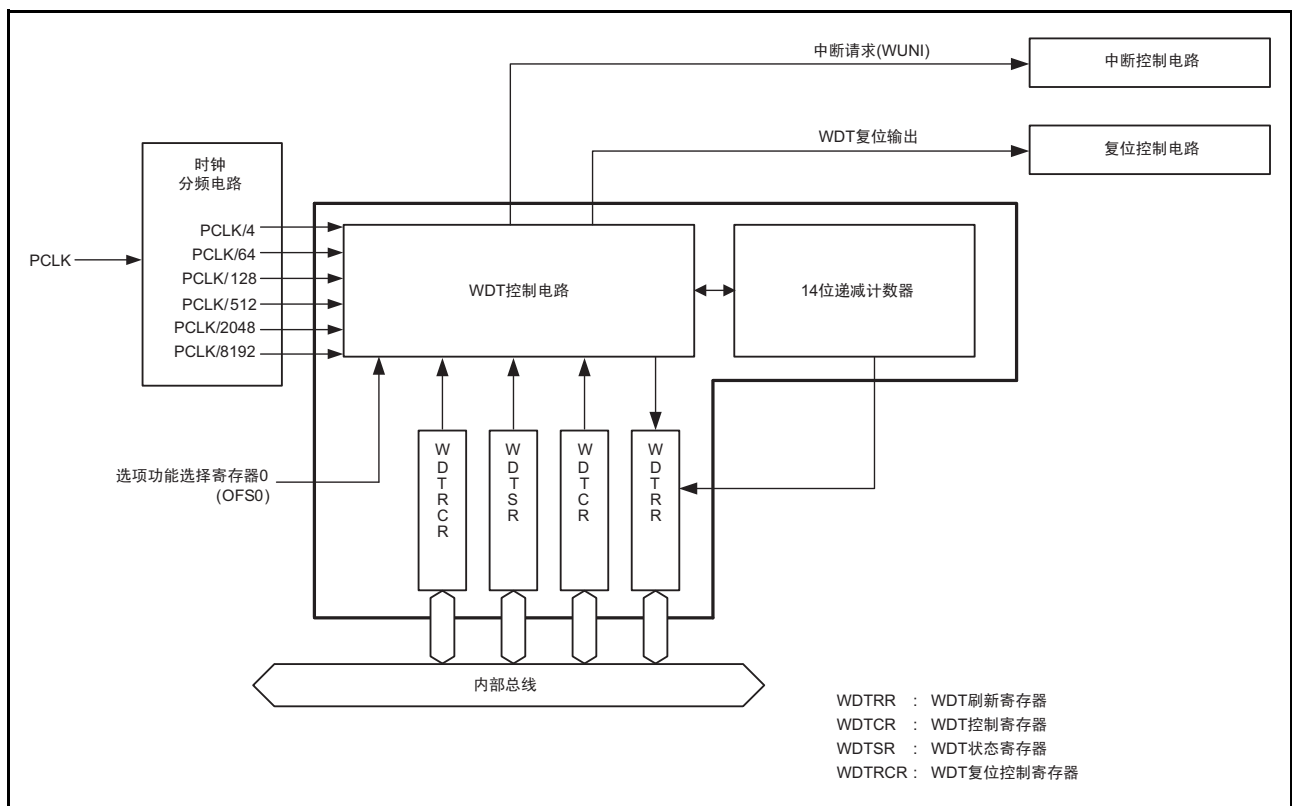
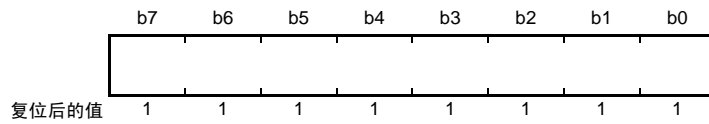


图 25.1 WDT 的框图

25.2 寄存器说明

25.2.1 WDT 刷新寄存器 (WDTRR)

地址 0008 8020h



位	功能	R/W
b7-b0	通过在写“00h”后写“FFh”进行刷新。	R/W

WDTRR 寄存器是刷新 WDT 递减计数器的寄存器。

在刷新允许期间，通过在给 WDTRR 寄存器写“00h”后写“FFh”（刷新），刷新 WDT 递减计数器。

在自动启动模式中，如果递减计数器被刷新，就从选项功能选择寄存器 0 的 WDT 超时期间选择位 (OFS0.WDTPS[1:0]) 设定的值开始递减计数。在寄存器启动模式中，如果递减计数器被刷新，就从 WDT 控制寄存器的超时期间选择位 (WDTCR.TOPS[1:0]) 设定的值开始递减计数，也能通过复位解除后的第 1 次刷新，从 WDTCR.TOPS[1:0] 位设定的值开始递减计数。

如果写“00h”，读取值就为“00h”；如果写“00h”以外的值，读取值就为“FFh”。

有关刷新的详细内容，请参照“25.3.3 刷新”。

25.2.2 WDT 控制寄存器 (WDTCR)

地址 0008 8022h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	RPSS[1:0]	—	—	RPES[1:0]	CKS[3:0]			—	—	TOPS[1:0]				
0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

复位后的值

位	符号	位名	功能	R/W
b1-b0	TOPS[1:0]	超时期间选择位	b1 b0 0 0: 1024 个周期 (03FFh) 0 1: 4096 个周期 (0FFFh) 1 0: 8192 个周期 (1FFFh) 1 1: 16384 个周期 (3FFFh)	R/W
b3-b2	—	保留位	读取值为“0”，写操作无效。	R
b7-b4	CKS[3:0]	时钟分频比选择位	b7 b4 0 0 0 1: PCLK/4 0 1 0 0: PCLK/64 1 1 1 1: PCLK/128 0 1 1 0: PCLK/512 0 1 1 1: PCLK/2048 1 0 0 0: PCLK/8192 上述以外: 不能设定	R/W
b9-b8	RPES[1:0]	窗口结束位置选择位	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (没有设定窗口的结束位置)	R/W
b11-b10	—	保留位	读取值为“0”，写操作无效。	R
b13-b12	RPSS[1:0]	窗口起始位置选择位	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (没有设定窗口的开始位置)	R/W
b15-b14	—	保留位	读取值为“0”，写操作无效。	R

WDTCR 寄存器有写操作限制，详细内容请参照“25.3.2 WDTCR 寄存器和 WDTRCR 寄存器的写控制”。

在自动启动模式中，WDTCR 寄存器的设定无效而选项功能选择寄存器 0 (OFS0) 的设定有效。OFS0 寄存器的设定和 WDTCR 寄存器各位的设定相同。详细内容请参照“25.3.8 选项功能选择寄存器 0 (OFS0) 和 WDT 寄存器的对应”。

TOPS[1:0] 位 (超时期间选择位)

这些位以 CKS[3:0] 位设定的分频时钟为 1 个周期，从 1024 个周期、4096 个周期、8192 个周期和 16384 个周期中选择递减计数器发生下溢为止的超时期间。

从刷新后到下溢为止的时间 (PCLK 数) 取决于 CKS[3:0] 位和 TOPS[1:0] 位的组合。

CKS[3:0] 位和 TOPS[1:0] 位的设定与超时期间、PCLK 数的关系如表 25.2 所示。

表 25.2 超时期间设定表

CKS[3:0] 位				TOPS[1:0] 位		时钟分频比	超时期间 (周期数)	PCLK 数
b7	b6	b5	b4	b1	b0			
0	0	0	1	0	0	PCLK/4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	PCLK/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	PCLK/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	PCLK/512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	PCLK/2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	PCLK/8192	1024	8388608
				0	1		4096	33554432
				1	0		8192	67108864
				1	1		16384	134217728

CKS[3:0] 位 (时钟分频比选择位)

这些位从 4 分频、64 分频、128 分频、512 分频、2048 分频和 8192 分频中选择 WDT 对外围时钟 (PCLK) 进行分频的分频比。能通过和 TOPS[1:0] 位的组合设定, 将 WDT 的计数期间设定在 PCLK 的 4096 ~ 134217728 个时钟之间。

RPES[1:0] 位 (窗口结束位置选择位)

这些位从计数期间的 75%、50%、25% 和 0% 中选择递减计数器的窗口结束位置。窗口结束位置选择小于窗口起始位置的值 (窗口起始位置 > 窗口结束位置)。如果设定窗口结束位置大于窗口起始位置的值, 就只有窗口起始位置的设定有效。

RPSS[1:0] 位和 RPES[1:0] 位设定的窗口起始 / 结束位置的计数器值根据 TOPS[1:0] 位的设定而变。

与 TOPS[1:0] 位的值对应的窗口起始 / 结束位置的计数器值如表 25.3 所示。

表 25.3 超时期间和窗口允许 / 结束位置的计数器值的对应表

TOPS[1:0] 位		超时期间		刷新允许 / 结束计数器值			
b1	b0	周期数	计数器值	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

RPSS[1:0] 位 (窗口起始位置选择位)

这些位从计数期间 (开始计数时为 100%、发生下溢时为 0%) 的 100%、75%、50% 和 25% 中选择递减计数器的窗口起始位置。从窗口起始位置到窗口结束位置的期间为刷新允许期间, 其他期间为刷新禁止期间。

RPSS[1:0] 位和 RPES[1:0] 位的设定值与刷新允许 / 禁止期间的关系如图 25.2 所示。

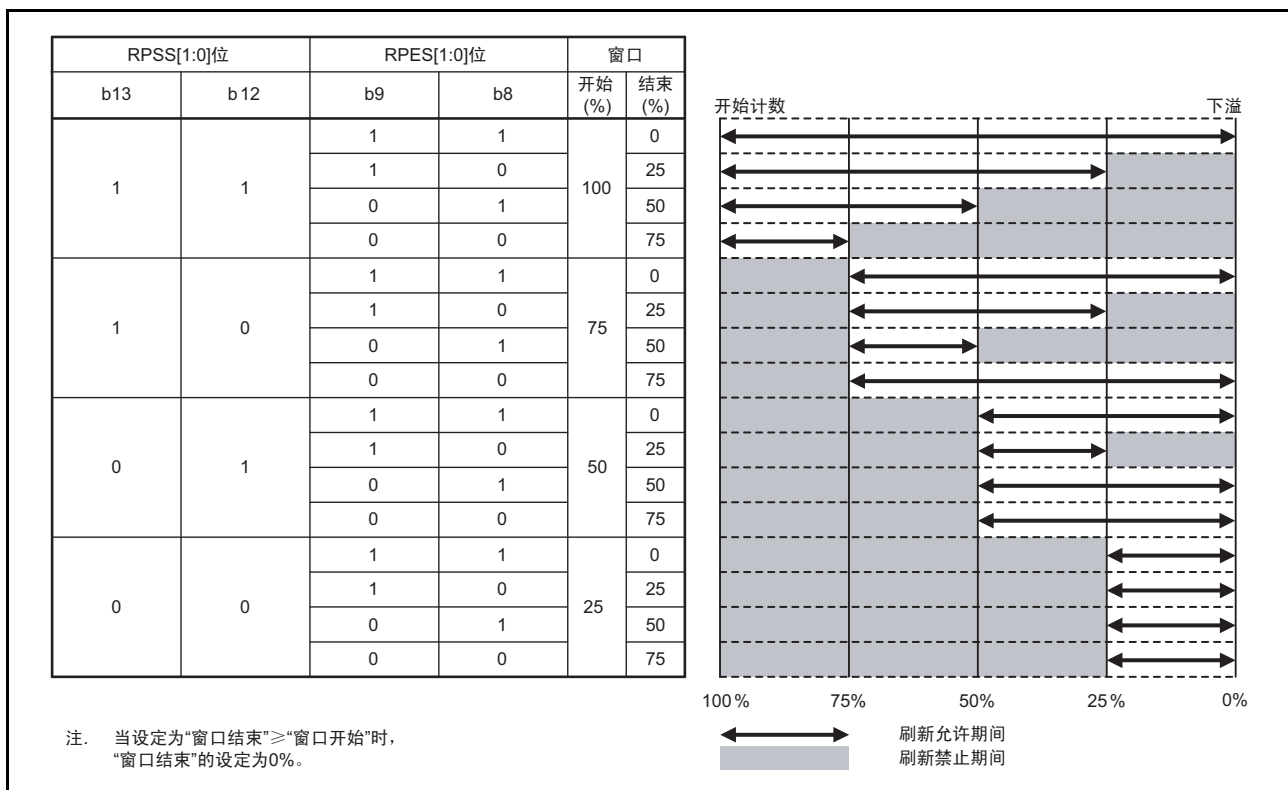
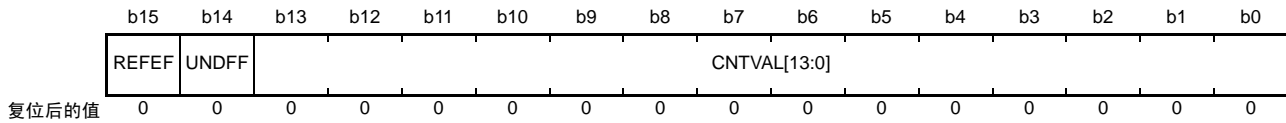


图 25.2 RPSS[1:0] 位、RPES[1:0] 位和刷新允许期间

25.2.3 WDT 状态寄存器 (WDTSR)

地址 0008 8024h



位	符号	位名	功能	R/W
b13-b0	CNTVAL[13:0]	递减计数器值的位	递减计数器的计数值	R
b14	UNDFE	下溢标志	0: 没有发生下溢 1: 发生下溢	R(W) (注1)
b15	REFEF	刷新错误标志	0: 没有发生刷新错误 1: 发生刷新错误	R(W) (注1)

注 1. 为了将标志置“0”，只能写“0”。

通过 WDT 的复位源而不通过其他复位源对 WDTSR 寄存器进行初始化。

CNTVAL[13:0] 位 (递减计数器值的位)

这些位能确认递减计数器的计数值。但是，读到的计数值有可能和递减计数器的实际值相差 1 个计数时钟。

UNDFE 标志 (下溢标志)

此标志能确认递减计数器的下溢发生状态。

当读取值为“1”时，表示递减计数器发生下溢；当读取值为“0”时，表示递减计数器没有发生下溢。

要将 UNDFE 标志的值置“0”时，必须给此标志写“0”。写“1”无效。

REFEF 标志 (刷新错误标志)

此标志能确认刷新错误 (刷新禁止期间的刷新) 的发生状态。

当读取值为“1”时，表示发生刷新错误；当读取值为“0”时，表示没有发生刷新错误。

要将 REFEF 标志的值置“0”时，必须给此标志写“0”。写“1”无效。

25.2.4 WDT 复位控制寄存器 (WDTRCR)

地址 0008 8026h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
复位后的值	1	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b6-b0	—	保留位	读取值为“0”，写操作无效。	R
b7	RSTIRQS	复位中断请求选择位	0: 允许输出非屏蔽中断请求 1: 允许输出复位	R/W

WDTRCR 寄存器有写操作限制，详细内容请参照“25.3.2 WDTCR 寄存器和 WDTRCR 寄存器的写控制”。

在自动启动模式中，WDTRCR 寄存器的设定无效而选项功能选择寄存器 0 (OFS0) 的设定有效。OFS0 寄存器的设定和 WDTRCR 寄存器各位的设定相同。详细内容请参照“25.3.8 选项功能选择寄存器 0 (OFS0) 和 WDT 寄存器的对应”。

25.2.5 选项功能选择寄存器 0 (OFS0)

有关 OFS0 寄存器，请参照“25.3.8 选项功能选择寄存器 0 (OFS0) 和 WDT 寄存器的对应”。

25.3 运行说明

25.3.1 各计数开始条件下的运行

通过选项功能选择寄存器 0 的 WDT 启动模式选择位 (OFS0.WDTSTRT) 选择 WDT 的启动模式。

当 OFS0.WDTSTRT 位为“1” (寄存器启动模式) 时，WDT 控制寄存器 (WDTCR) 和 WDT 复位控制寄存器 (WDTRCR) 的设定有效，通过刷新 WDT 刷新寄存器 (WDTRR) 开始计数。当 OFS0.WDTSTRT 位为“0” (自动启动模式) 时，OFS0 寄存器有效，在复位后自动开始计数。

25.3.1.1 寄存器启动模式

当选项功能选择寄存器 0 的 WDT 启动模式选择位 (OFS0.WDTSTRT) 为“1”时，为寄存器启动模式，WDT 控制寄存器 (WDTCR) 和 WDT 复位控制寄存器 (WDTRCR) 有效。

在解除复位后，给 WDTCR 寄存器设定时钟分频比、窗口起始 / 结束位置和超时期间，并且给 WDTRCR 寄存器设定复位输出或者中断请求输出。然后，通过刷新将超时期间选择位 (WDTCR.TOPS[1:0]) 选择的值设定到递减计数器并且开始递减计数。

此后，如果程序正常运行并且在刷新允许期间内进行了刷新，就在每次刷新时重新设定计数器值并且继续进行递减计数。在此期间，WDT 不输出复位。但是，如果因程序失控等无法刷新递减计数器而导致递减计数器发生下溢，或者因在刷新允许期间外进行刷新而发生刷新错误，WDT 就输出复位或者非屏蔽中断请求 (WUNI)。通过设定复位中断请求选择位 (WDTRCR.RSTIRQS)，选择输出复位或者中断请求。

以下条件下的运行例子如图 25.3 所示。

- WDT 启动模式选择位 (OFS0.WDTSTRT): “1” (寄存器启动模式)
- 复位中断请求选择位 (WDTRCR.RSTIRQS): “1” (允许输出复位)
- 窗口起始位置选择位 (WDTCR.RPSS[1:0]): “10b” (75%)
- 窗口结束位置选择位 (WDTCR.RPES[1:0]): “10b” (25%)

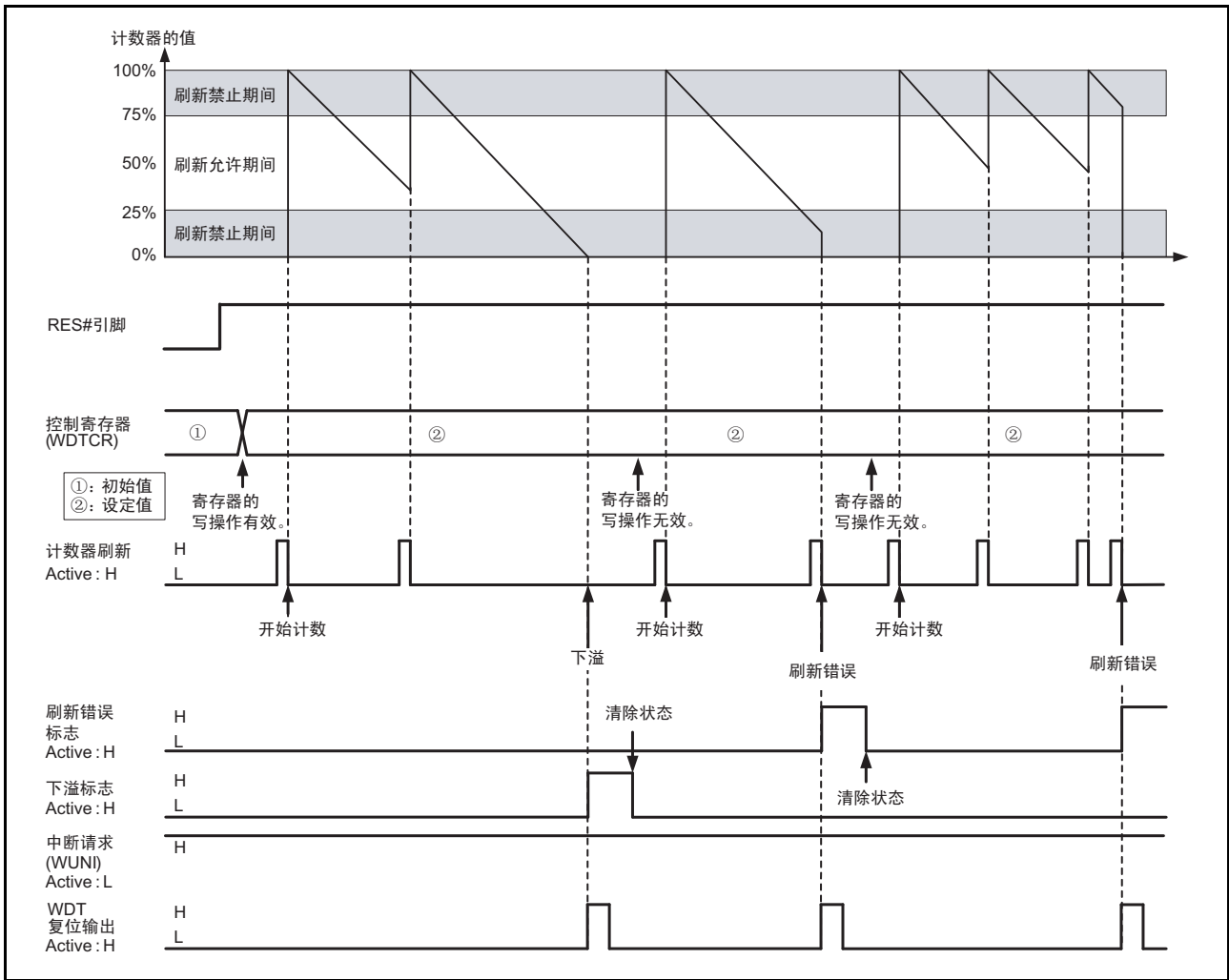


图 25.3 寄存器启动模式的运行例子

25.3.1.2 自动启动模式

当选项功能选择寄存器 0 的 WDT 启动模式选择位 (OFS0.WDTSTRT) 为“0”时, 为自动启动模式, WDT 控制寄存器 (WDTCR) 和 WDT 复位控制寄存器 (WDTRCR) 无效。

在复位期间, 通过选项功能选择寄存器 0 (OFS0) 设定时钟频率比、窗口起始 / 结束位置、超时期间、复位输出或者中断请求输出。然后通过解除复位将 WDT 超时期间选择位 (OFS0.WDTPS[1:0]) 设定的超时期间值设定到递减计数器并且自动开始递减计数。

此后, 如果程序正常运行并且在刷新允许期间内进行了刷新, 就在每次刷新时重新设定计数器值并且继续进行递减计数。在此期间, WDT 不输出复位。如果因程序失控等无法刷新递减计数器而导致递减计数器发生下溢, 或者因在刷新允许期间外进行刷新而发生刷新错误, WDT 就输出复位或者非屏蔽中断请求 (WUNI)。在输出 1 个计数周期的复位或者非屏蔽中断请求后, 递减计数器重新加载超时期间并且重新开始计数。通过设定 WDT 复位中断请求选择位 (OFS0.WDTRSTIRQS), 选择输出复位或者中断请求。

以下条件下的运行例子如图 25.4 所示。

- WDT 启动模式选择位 (OFS0.WDTSTRT): “0” (自动启动模式)
- 复位中断请求选择位 (OFS0.WDTRSTIRQS): “0” (允许输出非屏蔽中断请求)
- 窗口起始位置选择位 (OFS0.WDTRPSS[1:0]): “10b” (75%)
- 窗口结束位置选择位 (OFS0.WDTRPES[1:0]): “10b” (25%)

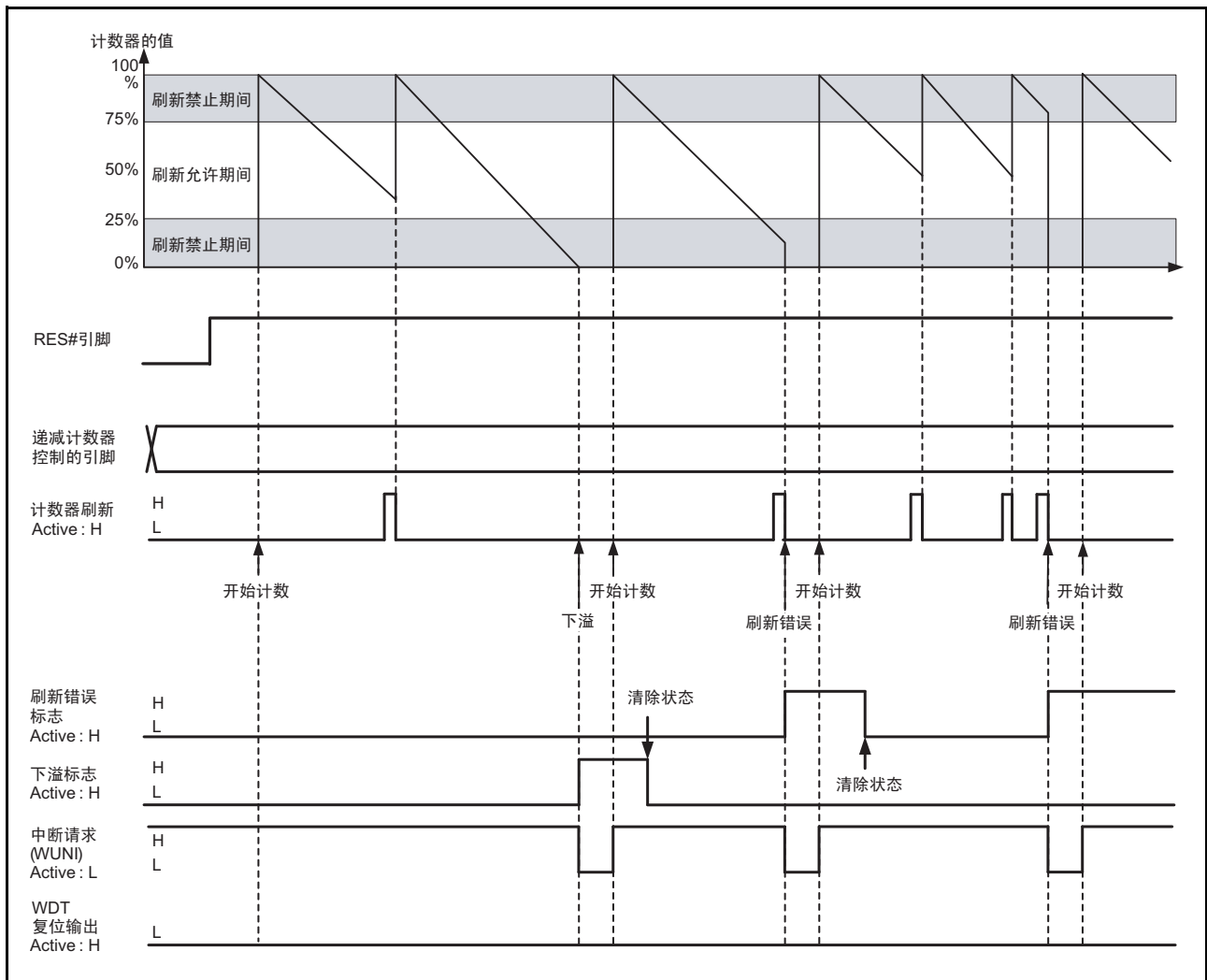


图 25.4 自动启动模式的运行例子

25.3.2 WDTCR 寄存器和 WDTRCR 寄存器的写控制

在从复位解除后到第 1 次刷新的期间，只能写 1 次 WDT 控制寄存器 (WDTCR) 和 WDT 复位控制寄存器 (WDTRCR)。

如果刷新 (开始计数) 或者写 WDTCR 寄存器和 WDTRCR 寄存器，WDT 的内部保护信号就变为“1”，此后保护对 WDTCR 寄存器和 WDTRCR 寄存器的写操作。

通过 WDT 的复位源而不通过其他复位源解除保护。

WDTCR 寄存器的写控制波形如图 25.5 所示。

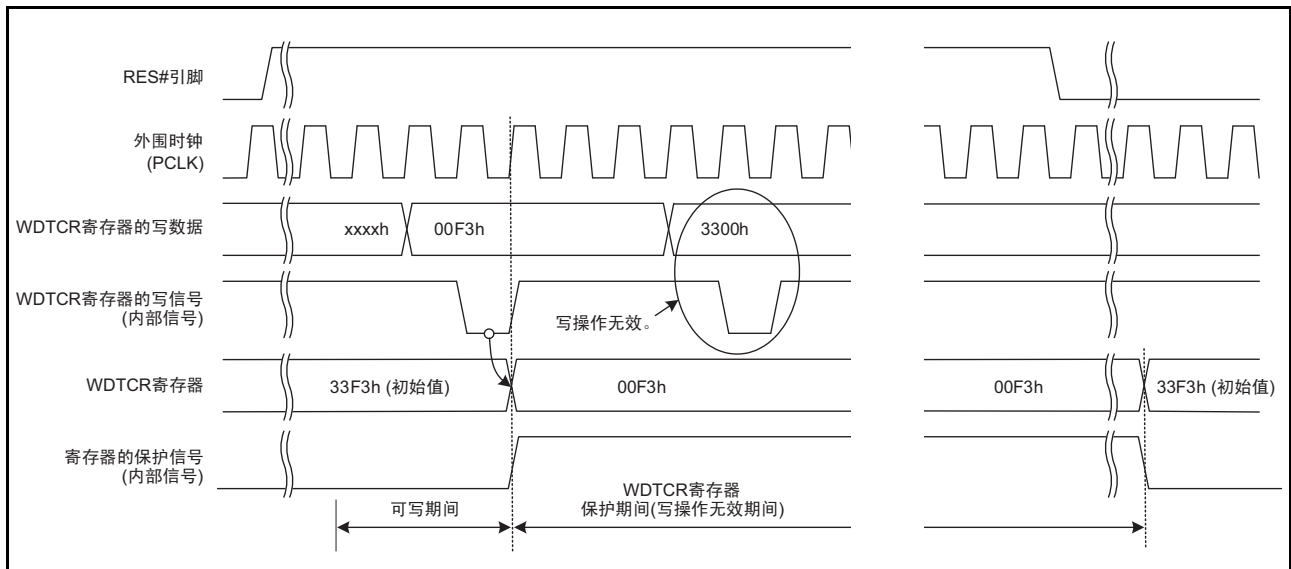


图 25.5 WDTCR 寄存器的写控制波形

25.3.3 刷新

要刷新递减计数器并且使递减计数器开始运行 (通过刷新开始计数) 时，在给 WDT 的刷新寄存器 (WDTRR) 写“00h”后接着写“FFh”。如果在写“00h”后写“FFh”以外的值，就不进行刷新。能通过再次按照“00h”→“FFh”的顺序写 WDTRR 寄存器，正常地进行刷新。

即使按照“00h” (第 1 次) →“00h” (第 2 次) 的顺序进行写操作，因为之后通过写“FFh”也能使“00h”→“FFh”顺序的写操作成立，所以“00h” (第 n-1 次) →“00h” (第 n 次) →“FFh”的写操作也有效并且进行刷新。同样，即使在“00h”之前写的不是“00h”，只要“00h”→“FFh”顺序的写操作成立，也进行刷新。另外，在给 WDTRR 寄存器写“00h”和“FFh”的期间，即使存取 WDTRR 以外的寄存器或者读 WDTRR 寄存器，也进行刷新。

【刷新有效的写操作例子】

- “00h”→“FFh”
- “00h” (第 n-1 次) →“00h” (第 n 次) →“FFh”
- “00h”→存取其他寄存器或者读 WDTRR 寄存器→“FFh”

【刷新无效的写操作例子】

- “23h” (不是“00h”) →“FFh”
- “00h”→“54h” (不是“FFh”)
- “00h”→“AAh” (不是“00h”和“FFh”) →“FFh”

作为刷新运行，即使在刷新允许期间外给 WDTRR 寄存器写“00h”，只要在刷新允许期间内给 WDTRR 寄存器写“FFh”，写操作就成立并且进行刷新。

在给 WDTRR 寄存器写“FFh”后，递减计数器的刷新时序最多需要 4 个计数周期（1 个周期的外围时钟 (PCLK) 数因时钟分频比选择位 (WDTCR.CKS[3:0]) 的设定值而不同）。因此，在从刷新允许期间结束位置开始到 4 个计数时钟或者递减计数器发生下溢的 4 个计数时钟前为止，必须给 WDTRR 寄存器写完“FFh”。能通过递减计数器值的位 (WDTSR.CNTVAL[13:0]) 确认递减计数器的值。

【刷新运行的时序例子】

- 当窗口起始位置为“1FFFh”时，即使在给 WDTRR 寄存器写“00h”后写“1FFFh”（例如“2002h”），只要在 WDTSR.CNTVAL[13:0] 位的值变为“1FFFh”后给 WDTRR 寄存器写“FFh”，也进行刷新。
- 当窗口结束位置为“1FFFh”时，如果在按照“00h”→“FFh”的顺序写 WDTRR 寄存器后立即读 WDTSR.CNTVAL[13:0] 位的值并且读取值大于等于“2003h”（“1FFFh”的 4 个计数时钟前），就进行刷新。
- 当刷新允许期间持续到“0000h”时，能在下溢前进行刷新。此时，如果在按照“00h”→“FFh”的顺序写 WDTRR 寄存器后立即读 WDTSR.CNTVAL[13:0] 位的值并且读取值大于等于“0003h”（下溢的 4 个计数时钟前），就不产生下溢而进行刷新。

时钟分频比为 PCLK/64 时的 WDT 刷新波形如图 25.6 所示。

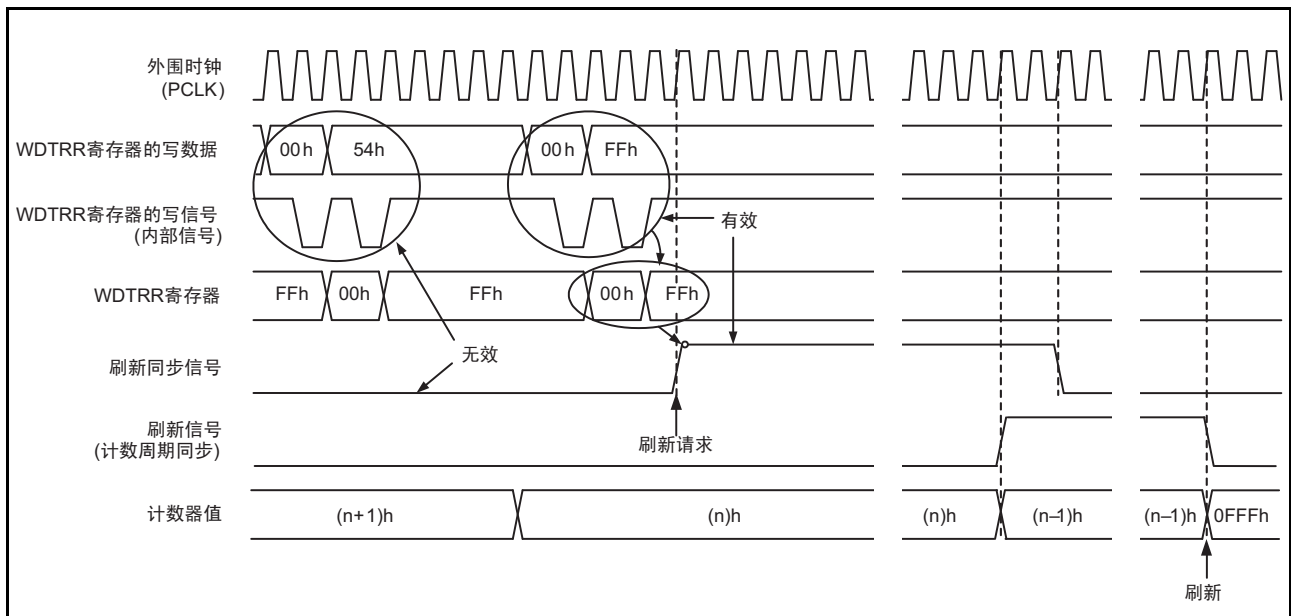


图 25.6 WDT 的刷新波形 (WDTCR.CKS[3:0]=0100b、WDTCR.TOPS[1:0]=01b)

25.3.4 状态标志

刷新错误标志 (WDTSR.REFEF) 和下溢标志 (WDTSR.UNDF) 保持 WDT 输出复位时的复位源或者发生 WDT 中断请求时的中断源。

在解除复位后或者在发生中断请求时，能通过读 WDTSR.REFEF 标志或者 WDTSR.UNDF 标志，确认复位源或者中断源的发生状态。

要将各标志的值置“0”时，必须写“0”。写“1”无效。

即使不将各标志置“0”，也不影响运行。如果不置“0”，就在下次 WDT 输出复位时清除旧的复位源并且写新的复位源，或者在下次 WDT 产生中断请求时清除旧的中断源并且写新的中断源。

另外，要读通过写“0”清除标志后被反映的值时，需要几个 PCLK 时钟（至少 5 个时钟）。

25.3.5 复位输出

如果在寄存器启动模式中将复位中断选择位 (WDTRCR.RSTIRQS) 置“1”或者在自动启动模式中将选项功能选择寄存器 0 的 WDT 复位中断请求选择位 (OFS0.WDTRSTIRQS) 置“1”，就在发生递减计数器下溢或者刷新错误时输出 1 个计数周期的复位。

在寄存器启动模式中，在输出复位后递减计数器保持初始状态 (ALL“0”)。在解除复位并且重新启动后，通过刷新重新设定计数器值并且开始递减计数。

在自动启动模式中，在输出复位后自动开始递减计数。

25.3.6 中断源

如果在寄存器启动模式中将复位中断选择位 (WDTRCR.RSTIRQS) 置“0”或者在自动启动模式中将选项功能选择寄存器 0 的 WDT 复位中断请求选择位 (OFS0.WDTRSTIRQS) 置“0”，就在发生递减计数器下溢或者刷新错误时产生非屏蔽中断 (WUNI)。

表 25.4 WDT 的中断源

名称	中断源	DTC 的启动	DMAC 的启动
WUNI	递减计数器的下溢 刷新错误	不能	不能

25.3.7 递减计数器值的读操作

WDT 将计数值保存到 WDT 状态寄存器的递减计数器 (WDTSR.CNTVAL[13:0] 位)。能通过读被保存到 WDTSR.CNTVAL[13:0] 位的值，确认计数值。

时钟分频比为 PCLK/64 时的递减计数器值的读处理如图 25.7 所示。

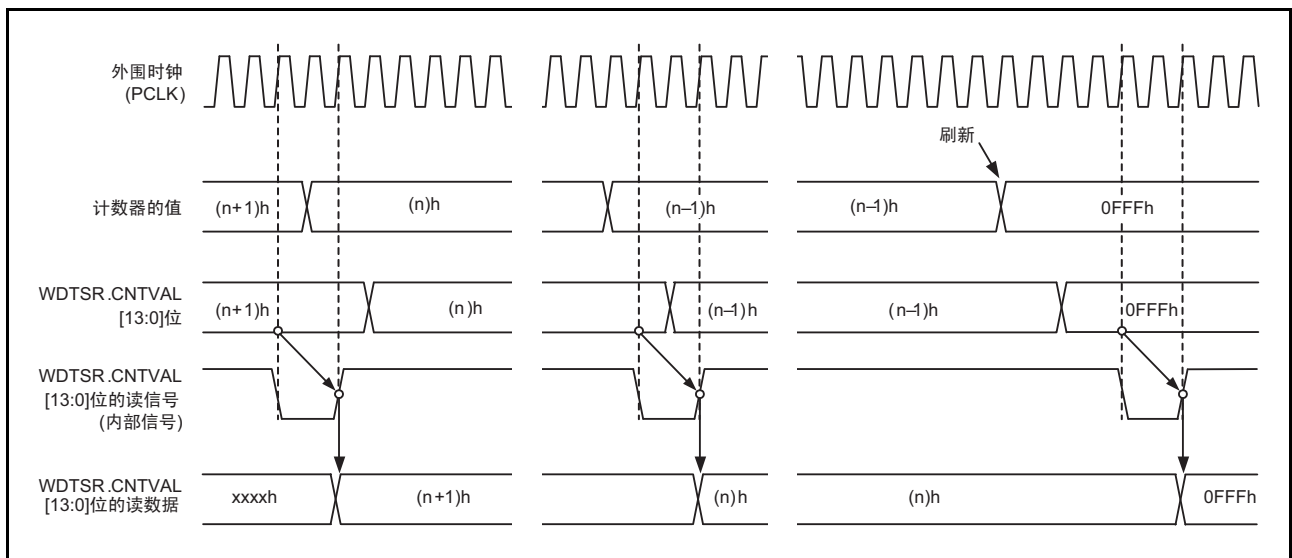


图 25.7 WDT 递减计数器值的读处理 (WDTCR.CKS[3:0]=0100b、WDTCR.TOPS[1:0]=01b)

25.3.8 选项功能选择寄存器 0 (OFS0) 和 WDT 寄存器的对应

通过选项功能选择寄存器 0 (OFS0) 进行的递减计数器控制、复位输出控制或者中断请求输出控制与 WDT 控制寄存器 (WDTCR) 和 WDT 复位控制寄存器 (WDTRCR) 的对应如表 25.5 所示。通过 WDT 启动模式选择位 (OFS0.WDTSTRT) 选择 OFS0 寄存器和 WDTCR 寄存器、WDTRCR 寄存器控制的有效或者无效。

必须在 WDT 运行过程中固定 OFS0 寄存器的设定。

有关 OFS0 寄存器，请参照“7.2.1 选项功能选择寄存器 0 (OFS0) ”。

表 25.5 选项功能选择寄存器 0 (OFS0) 和 WDT 寄存器的对应

控制	功能	OFS0 寄存器 (在自动启动模式中有效) OFS0.WDTSTRT=0	WDT 寄存器 (在寄存器启动模式中有效) OFS0.WDTSTRT=1
递减计数器	选择超时期间。	OFS0.WDTPPS[1:0]	WDTCR.TOPS[1:0]
	选择时钟分频比。	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	选择窗口起始位置。	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	选择窗口结束位置。	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
复位输出 / 中断请求输出	选择输出复位或者中断请求	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS

26. 独立看门狗定时器 (IWDTa)

独立看门狗定时器 (IWDT) 是和检测程序失控的传统看门狗定时器独立使用的看门狗定时器。

IWDT 内置 14 位递减计数器，如果递减的计数值发生下溢，就能通过复位输出对本 LSI 进行复位，也能选择中断请求的发生。能通过刷新递减计数器的计数值，将计数值返回初始值并且重新开始计数。另外，能设定可刷新期间。如果在可刷新期间进行刷新，就能对计数器进行初始化并且再次进行计数。如果在可刷新期间外进行刷新，就输出复位或者中断请求。因此，能通过调整刷新间隔检测程序失控。在发生下溢或者在可刷新期间外进行刷新后，IWDT 停止计数。在寄存器启动模式中，在刷新后自动重新开始计数；在自动启动模式中，在输出复位或者中断请求后自动重新开始计数。

26.1 概要

IWDT 有自动启动模式和寄存器启动模式两种模式，自动开始模式是在解除复位后自动开始计数的模式，寄存器启动模式是在解除复位后通过刷新（写寄存器）开始计数的模式。

在自动启动模式中，在解除复位前给选项功能选择寄存器 0 (OFS0) 设定时钟分频比、刷新的窗口起始 / 结束位置、超时期间、下溢时的复位输出 / 非屏蔽中断请求输出以及睡眠模式的计数停止控制。

在寄存器启动模式中，在解除复位后并且在通过刷新开始计数前给寄存器设定时钟分频比、刷新的窗口起始 / 结束位置、超时期间、下溢时的复位输出 / 非屏蔽中断请求输出以及睡眠模式的计数停止控制。

通过 IWDT 启动模式选择位 (OFS0.IWDTSTRT) 选择自动启动模式或者寄存器启动模式。

当选择自动启动模式时 (OFS0.IWDTSTRT=0)，IWDT 控制寄存器 (IWDTCR)、IWDT 复位控制寄存器 (IWDTRCR) 和 IWDT 计数停止控制寄存器 (IWDTCSPTPR) 的设定无效而 OFS0 寄存器的设定有效。

当选择寄存器启动模式时 (OFS0.IWDTSTRT=1)，OFS0 寄存器的设定无效而 IWDTCR、IWDTRCR 和 IWDTCSPTPR 寄存器的设定有效。

IWDT 的规格如表 26.1 所示。

表 26.1 IWDT 的规格 (1/2)

项目	内容
计数源	IWDT 专用低速时钟 (IWDTCLK)
时钟分频比	1 分频、16 分频、32 分频、64 分频、128 分频、256 分频
计数	通过 14 位递减计数器进行递减计数。
计数开始条件	<ul style="list-style-type: none"> 复位后自动开始计数 (自动启动模式)。 通过刷新 (在给 IWDTRR 寄存器写 "00h" 后写 "FFh") 开始计数 (寄存器启动模式)。
计数停止条件	<ul style="list-style-type: none"> 复位 (递减计数器和寄存器返回初始值) 当发生下溢或者刷新错误时 重新开始计数 (自动启动模式: 自动、寄存器启动模式: 刷新)
窗口功能	能设定窗口起始 / 结束位置 (刷新允许 / 禁止期间)。
复位输出源	<ul style="list-style-type: none"> 当递减计数器发生下溢时 在刷新允许期间外进行刷新 (刷新错误) 时
中断请求输出源	<ul style="list-style-type: none"> 在递减计数器发生下溢时产生非屏蔽中断 (WUNI) 时 在刷新允许期间外进行刷新 (刷新错误) 时
读计数值	能通过读 IWDTSR 寄存器，读递减计数器的计数值。
输出信号 (内部信号)	<ul style="list-style-type: none"> 复位输出 中断请求输出 睡眠模式的计数停止控制输出

表 26.1 IWDT 的规格 (2/2)

项目	内容
自动启动模式 (由选项功能选择寄存器 0 (OFS0) 控制)	<ul style="list-style-type: none"> 选择复位后的时钟分频 (OFS0.IWDTCKS[3:0] 位)。 选择看门狗定时器的超时间 (OFS0.IWDTTOPS[1:0] 位)。 选择看门狗定时器的窗口起始位置 (OFS0.IWDTRPSS[1:0] 位)。 选择看门狗定时器的窗口结束位置 (OFS0.IWDTRPES[1:0] 位)。 选择复位输出或者中断请求输出 (OFS0.IWDRSTIRQS 位)。 选择在向睡眠模式、软件待机模式、深度软件待机模式或者全模块时钟停止模式转移时停止递减计数 (OFS0.IWDTSLCSTP 位)。
寄存器启动模式 (由 IWDT 寄存器控制)	<ul style="list-style-type: none"> 选择刷新后的时钟分频比 (IWDTCR.CKS[3:0] 位)。 选择看门狗定时器的超时间 (IWDTCR.TOPS[1:0] 位)。 选择看门狗定时器的窗口起始位置 (IWDTCR.RPSS[1:0] 位)。 选择看门狗定时器的窗口结束位置 (IWDTCR.RPES[1:0] 位)。 选择复位输出或者中断请求输出 (IWDTCR.RSTIRQS 位)。 选择在向睡眠模式、软件待机模式、深度软件待机模式或者全模块时钟停止模式转移时停止递减计数 (IWDTCSTPR.SLCSTP 位)。

在使用 IWDT 时, 为了使 IWDT 在外围时钟 (PCLK) 停止的情况下运行, 需要外围时钟 (PCLK) 和 IWDT 专用低速时钟 (IWDTCLK)。总线接口部和寄存器部通过 PCLK 运行, 14 位递减计数器和控制电路通过 IWDTCLK 运行。

通过同步电路连接外围时钟运行块和 IWDT 专用低速时钟运行块之间的信号。

IWDT 的框图如图 26.1 所示。

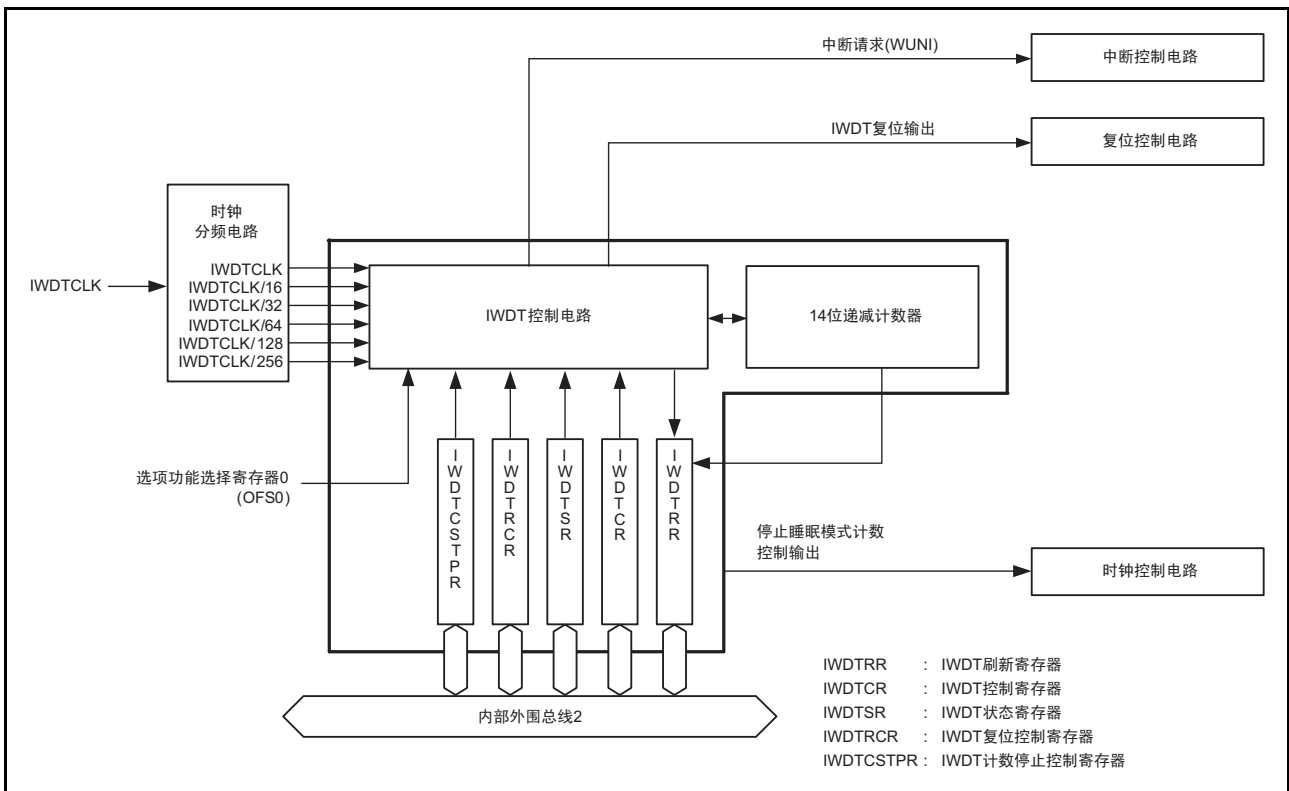
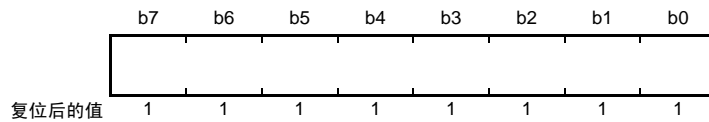


图 26.1 IWDT 的框图

26.2 寄存器说明

26.2.1 IWDT 刷新寄存器 (IWDTRR)

地址 0008 8030h



位	功能	R/W
b7-b0	通过在写“00h”后写“FFh”进行刷新。	R/W

IWDTRR 寄存器是刷新 IWDT 递减计数器的寄存器。

在刷新允许期间，通过在给 IWDTRR 寄存器写“00h”后写“FFh”（刷新），刷新 IWDT 递减计数器。

在自动启动模式中，如果递减计数器被刷新，就从选项功能选择寄存器 0 的 IWDT 超时期间选择位 (OFS0.IWDTTOPS[1:0]) 设定的值开始递减计数。在寄存器启动模式中，如果递减计数器被刷新，就从 IWDT 控制寄存器的超时期间选择位 (IWDTCR.TOPS[1:0]) 设定的值开始递减计数，也能通过解除复位后的第 1 次刷新，从 IWDTCR.TOPS[1:0] 位设定的值开始递减计数。

如果写“00h”，读取值就为“00h”；如果写“00h”以外的值，读取值就为“FFh”。

有关刷新的详细内容，请参照“26.3.3 刷新”。

26.2.2 IWDT 控制寄存器 (IWDTCR)

地址 0008 8032h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	RPSS[1:0]	—	—	RPES[1:0]	CKS[3:0]			—	—	TOPS[1:0]				
0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

复位后的值

位	符号	位名	功能	R/W
b1-b0	TOPS[1:0]	超时期间选择位	b1 b0 0 0: 1024 个周期 (03FFh) 0 1: 4096 个周期 (0FFFh) 1 0: 8192 个周期 (1FFFh) 1 1: 16384 个周期 (3FFFh)	R/W
b3-b2	—	保留位	读取值为“0”，写操作无效。	R
b7-b4	CKS[3:0]	时钟分频比选择位	b7 b4 0 0 0 0: IWDTCLK 0 0 1 0: IWDTCLK/16 0 0 1 1: IWDTCLK/32 0 1 0 0: IWDTCLK/64 1 1 1 1: IWDTCLK/128 0 1 0 1: IWDTCLK/256 上述以外: 不能设定	R/W
b9-b8	RPES[1:0]	窗口结束位置选择位	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (没有设定窗口的结束位置)	R/W
b11-b10	—	保留位	读取值为“0”，写操作无效。	R
b13-b12	RPSS[1:0]	窗口起始位置选择位	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (没有设定窗口的开始位置)	R/W
b15-b14	—	保留位	读取值为“0”，写操作无效。	R

IWDTCR 寄存器有写操作限制。详细内容请参照“26.3.2 IWDTCR 寄存器、IWDTRCR 寄存器和 IWDTCSTPR 寄存器的写控制”。

在自动启动模式中，IWDTCR 寄存器的设定无效而选项功能选择寄存器 0 (OFS0) 的设定有效。OFS0 寄存器的设定和 IWDTCR 寄存器各位的设定相同。详细内容请参照“26.3.8 选项功能选择寄存器 0 (OFS0) 和 IWDT 寄存器的对应”。

TOPS[1:0] 位 (超时期间选择位)

这些位以 CKS[3:0] 位设定的分频时钟为 1 个周期，从 1024 个周期、4096 个周期、8192 个周期和 16384 个周期中选择递减计数器发生下溢为止的超时期间。

从刷新后到下溢为止的时间 (IWDTCLK 数) 取决于 CKS[3:0] 位和 TOPS[1:0] 位的组合。

CKS[3:0] 位和 TOPS[1:0] 位的设定与超时期间、IWDTCLK 数的关系如表 26.2 所示。

表 26.2 超时期间设定表

CKS[3:0] 位				TOPS[1:0] 位		时钟分频比	超时期间 (周期数)	IWDTCLK 数
b7	b6	b5	b4	b1	b0			
0	0	0	0	0	0	IWDTCLK	1024	1024
				0	1		4096	4096
				1	0		8192	8192
				1	1		16384	16384
0	0	1	0	0	0	IWDTCLK/16	1024	16384
				0	1		4096	65536
				1	0		8192	131072
				1	1		16384	262144
0	0	1	1	0	0	IWDTCLK/32	1024	32768
				0	1		4096	131072
				1	0		8192	262144
				1	1		16384	524288
0	1	0	0	0	0	IWDTCLK/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	IWDTCLK/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	0	1	0	0	IWDTCLK/256	1024	262144
				0	1		4096	1048576
				1	0		8192	2097152
				1	1		16384	4194304

CKS[3:0] 位 (时钟分频比选择位)

这些位从 1 分频、16 分频、32 分频、64 分频、128 分频和 256 分频中选择 IWDT 对 IWDTCLK 进行分频的分频比。能通过和 TOPS[1:0] 位的组合，将 IWDT 的计数期间设定在 IWDTCLK 的 1024 ~ 4194304 个时钟之间。

RPES[1:0] 位 (窗口结束位置选择位)

这些位从计数期间的 75%、50%、25% 和 0% 中选择递减计数器的窗口结束位置。窗口结束位置选择小于窗口起始位置的值 (窗口起始位置 > 窗口结束位置)。如果设定窗口结束位置大于窗口起始位置的值，就只有窗口起始位置的设定有效。

RPSS[1:0] 位和 RPES[1:0] 位设定的窗口起始 / 结束位置的计数器值根据 TOPS[1:0] 位的设定而变。

与 TOPS[1:0] 位的值对应的窗口起始 / 结束位置的计数值如表 26.3 所示。

表 26.3 超时期间和窗口允许 / 结束位置的计数值的对应表

TOPS[1:0] 位		超时期间		刷新允许 / 结束计数器值			
b1	b0	周期数	计数器值	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

RPSS[1:0] 位 (窗口起始位置选择位)

这些位从计数期间 (开始计数时为 100%、发生下溢时为 0%) 的 100%、75%、50% 和 25% 中选择递减计数器的窗口起始位置。从窗口起始位置到窗口结束位置的期间为刷新允许期间, 其他期间为刷新禁止期间。

RPSS[1:0] 位和 RPES[1:0] 位的设定值与刷新允许 / 禁止期间的关系如图 26.2 所示。

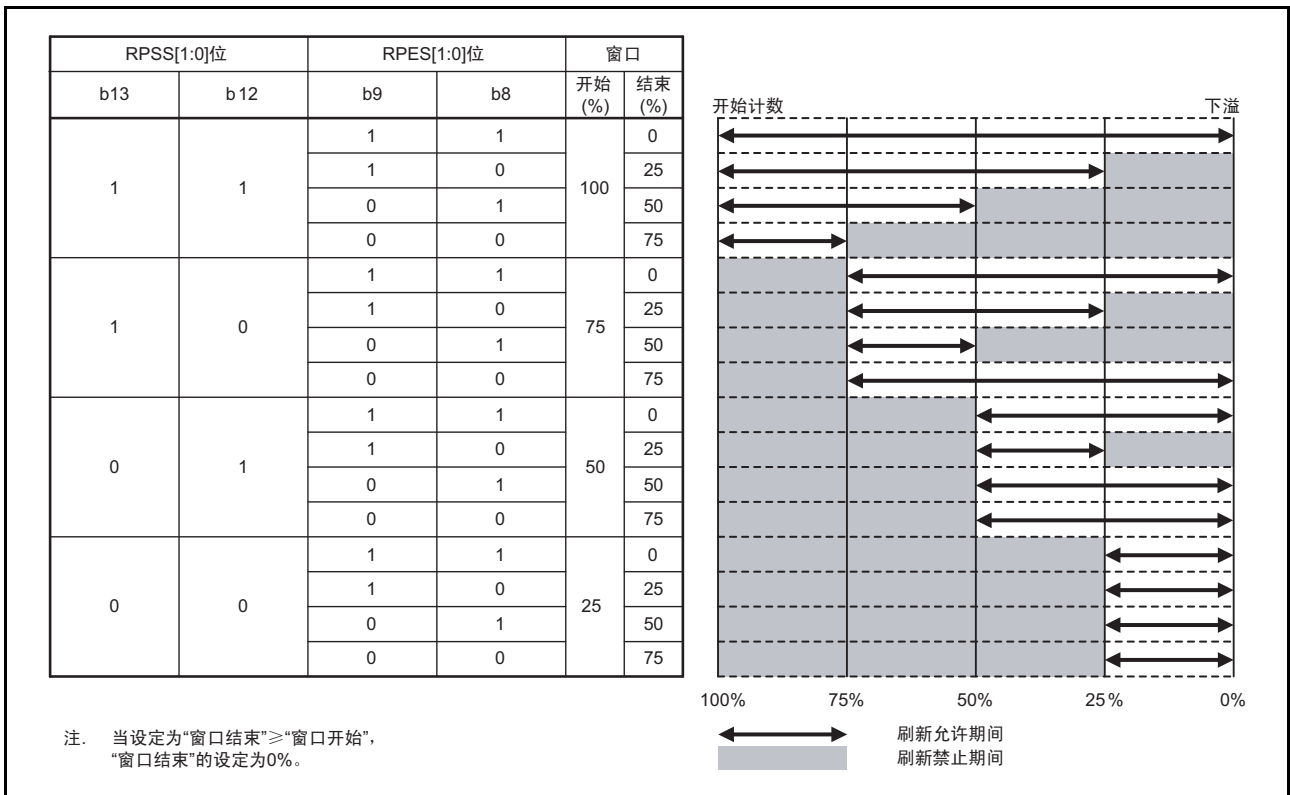
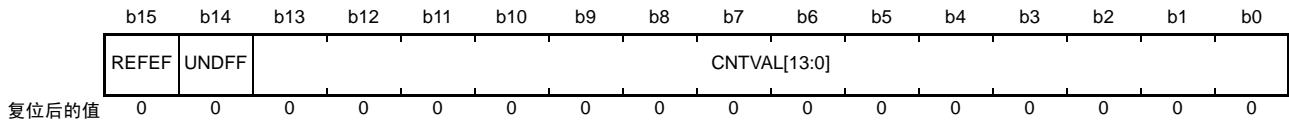


图 26.2 RPSS[1:0] 位、RPES[1:0] 位和刷新允许期间

26.2.3 IWDT 状态寄存器 (IWDTSR)

地址 0008 8034h



位	符号	位名	功能	R/W
b13-b0	CNTVAL[13:0]	递减计数器值的位	递减计数器的计数值	R
b14	UNDFE	下溢标志	0: 没有发生下溢 1: 发生下溢	R(W) (注1)
b15	REFEF	刷新错误标志	0: 没有发生刷新错误 1: 发生刷新错误	R(W) (注1)

注1. 为了将标志置“0”，只能写“0”。

通过 IWDT 的复位源而不通过其他复位源对 IWDTSR 寄存器进行初始化。

CNTVAL[13:0] 位 (递减计数器值的位)

这些位能确认递减计数器的计数值。但是，读到的计数值有可能和递减计数器的实际值相差 1 个计数时钟。

UNDFE 标志 (下溢标志)

此标志能确认递减计数器的下溢发生状态。

当读取值为“1”时，表示递减计数器发生下溢；当读取值为“0”时，表示递减计数器没有发生下溢。

要将 UNDFE 标志的值置“0”时，必须给此标志写“0”。写“1”无效。

REFEF 标志 (刷新错误标志)

此标志能确认刷新错误 (刷新禁止期间的刷新) 的发生状态。

当读取值为“1”时，表示发生刷新错误；当读取值为“0”时，表示没有发生刷新错误。

要将 REFEF 标志的值置“0”时，必须给此标志写“0”。写“1”无效。

26.2.4 IWDT 复位控制寄存器 (IWDTRCR)

地址 0008 8036h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIR QS	—	—	—	—	—	—	—
复位后的值	1	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b6-b0	—	保留位	读取值为“0”，写操作无效。	R
b7	RSTIRQS	复位中断请求选择位	0: 允许输出非屏蔽中断请求 1: 允许输出复位	R/W

IWDTRCR 寄存器有写操作限制，详细内容请参照“26.3.2 IWDTCR 寄存器、IWDTRCR 寄存器和 IWDTCSTPR 寄存器的写控制”。

在自动启动模式中，IWDTRCR 寄存器的设定无效而选项功能选择寄存器 0 (OFS0) 的设定有效。OFS0 寄存器的设定和 IWDTRCR 寄存器各位的设定相同。详细内容请参照“26.3.8 选项功能选择寄存器 0 (OFS0) 和 IWDT 寄存器的对应”。

26.2.5 IWDT 计数停止控制寄存器 (IWDTCSTPR)

地址 0008 8038h

	b7	b6	b5	b4	b3	b2	b1	b0
	SLCSTP	—	—	—	—	—	—	—
复位后的值	1	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b6-b0	—	保留位	读取值为“0”，写操作无效。	R
b7	SLCSTP	睡眠模式的计数停止控制位	0: 计数停止无效 1: 在向睡眠模式、软件待机模式、深度软件待机模式和全模块时钟停止模式转移时计数停止有效	R/W

IWDTCSTPR 寄存器有写操作限制，详细内容请参照“26.3.2 IWDTCR 寄存器、IWDTRCR 寄存器和 IWDTCSTPR 寄存器的写控制”。

在自动启动模式中，IWDTCSTPR 寄存器的设定无效而选项功能选择寄存器 0 (OFS0) 的设定有效。OFS0 寄存器的设定和 IWDTCSTPR 寄存器各位的设定相同。详细内容请参照“26.3.8 选项功能选择寄存器 0 (OFS0) 和 IWDT 寄存器的对应”。

SLCSTP 位 (睡眠模式的计数停止控制位)

此位选择在向睡眠模式、软件待机模式、深度软件待机模式和全模块时钟停止模式转移时停止计数。

26.2.6 选项功能选择寄存器 0 (OFS0)

有关 OFS0 寄存器，请参照“26.3.8 选项功能选择寄存器 0 (OFS0) 和 IWDT 寄存器的对应”。

26.3 运行说明

26.3.1 各计数开始条件下的运行

通过选项功能选择寄存器 0 的 IWDT 启动模式选择位 (OFS0.IWDTSTRT) 选择 IWDT 的启动模式。

当 OFS0.IWDTSTRT 位为“1” (寄存器启动模式) 时, IWDT 控制寄存器 (IWDTCR)、IWDT 复位控制寄存器 (IWDTRCR) 和 IWDT 计数停止控制寄存器 (IWDTCSTPR) 的设定有效, 通过刷新 IWDT 刷新寄存器 (IWDTRR) 开始计数。当 OFS0.IWDTSTRT 位为“0” (自动启动模式) 时, OFS0 寄存器有效, 在复位后自动开始计数。

26.3.1.1 寄存器启动模式

当选项功能选择寄存器 0 的 IWDT 启动模式选择位 (OFS0.IWDTSTRT) 为“1”时, 为寄存器启动模式, IWDT 控制寄存器 (IWDTCR)、IWDT 复位控制寄存器 (IWDTRCR) 和 IWDT 计数停止控制寄存器 (IWDTCSTPR) 有效。

在解除复位后, 给 IWDTCR 寄存器设定时钟分频比、窗口起始 / 结束位置和超时期间, 给 IWDTRCR 寄存器设定复位输出或者中断请求输出, 给 IWDTCSTPR 寄存器设定向低功耗模式转移时的 IWDT 递减计数器的计数停止控制。然后, 通过刷新将超时期间选择位 (IWDTCR.TOP[1:0]) 选择的值设定到递减计数器并且开始递减计数。

此后, 如果程序正常运行并且在刷新允许期间内进行了刷新, 就在每次刷新时重新设定计数器值并且继续进行递减计数。在此期间, IWDT 不输出复位。但是, 如果因程序失控等无法刷新递减计数器而导致递减计数器发生下溢, 或者因在刷新允许期间外进行刷新而发生刷新错误, IWDT 就输出复位或者非屏蔽中断请求 (WUNI)。通过设定复位中断请求选择位 (IWDTRCR.RSTIRQS), 选择输出复位或者中断请求。

以下条件下的运行例子如图 26.3 所示。

- IWDT 启动模式选择位 (OFS0.IWDTSTRT): “1” (寄存器启动模式)
- 复位中断请求选择位 (IWDTRCR.RSTIRQS): “1” (允许输出复位)
- 窗口起始位置选择位 (IWDTCR.RPSS[1:0]): “10b” (75%)
- 窗口结束位置选择位 (IWDTCR.RPES[1:0]): “10b” (25%)

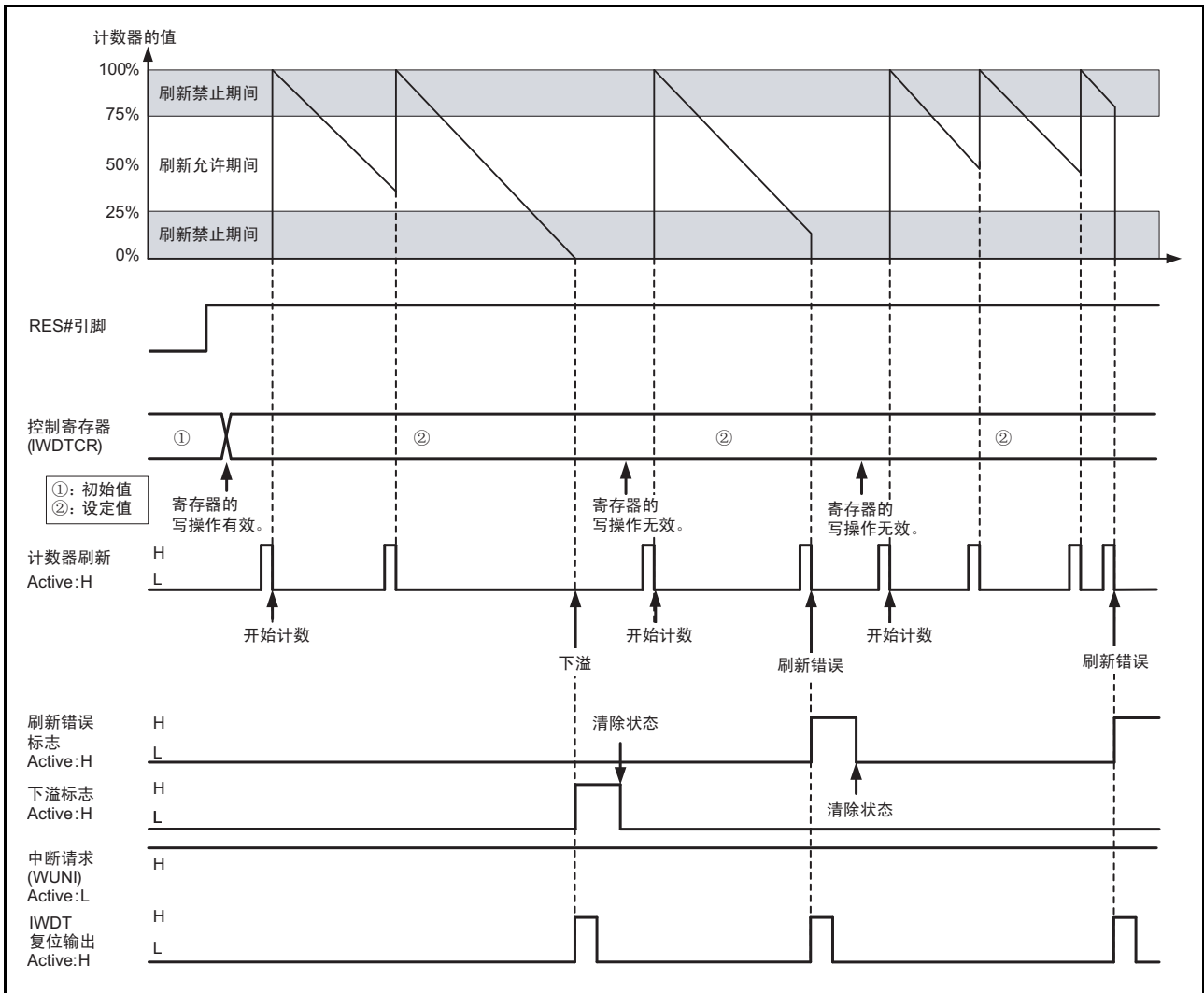


图 26.3 寄存器启动模式的运行例子

26.3.1.2 自动启动模式

当选项功能选择寄存器 0 的 IWDT 启动模式选择位 (OFS0.IWDTSTRT) 为“0”时, 为自动启动模式, IWDT 控制寄存器 0 (IWDTCR)、IWDT 复位控制寄存器 (IWDTRCR) 和 IWDT 计数停止控制寄存器 (IWDTCSTPR) 无效。

在复位期间, 通过 OFS0 寄存器设定时钟分频比、窗口起始 / 结束位置、超时期间、复位输出或者中断请求输出以及向低功耗模式转移时的 IWDT 递减计数器的计数停止控制。然后通过解除复位将 IWDT 超时期间选择位 (OFS0.IWDTTOPS[1:0]) 设定的超时期间值设定到递减计数器并且自动开始递减计数。

此后, 如果程序正常运行并且在刷新允许期间内进行了刷新, 就在每次刷新时重新设定计数器值并且继续进行递减计数。在此期间, IWDT 不输出复位。如果因程序失控等无法刷新递减计数器而导致递减计数器发生下溢, 或者因在刷新允许期间外进行刷新而发生刷新错误, IWDT 就输出复位或者非屏蔽中断请求 (WUNI)。在输出 1 个计数周期的复位或者非屏蔽中断请求后, 递减计数器重新加载超时期间并且重新开始计数。通过设定 IWDT 复位中断请求选择位 (OFS0.IWDRSTIRQS), 选择输出复位或者中断请求。

以下条件下的运行例子如图 26.4 所示。

- IWDT 启动模式选择位 (OFS0.IWDTSTRT): “0” (自动启动模式)
- 复位中断请求选择位 (OFS0.IWDTRSTIRQS): “0” (允许输出非屏蔽中断请求)
- 窗口起始位置选择位 (OFS0.IWDTRPSS[1:0]): “10b” (75%)
- 窗口结束位置选择位 (OFS0.IWDTRPES[1:0]): “10b” (25%)

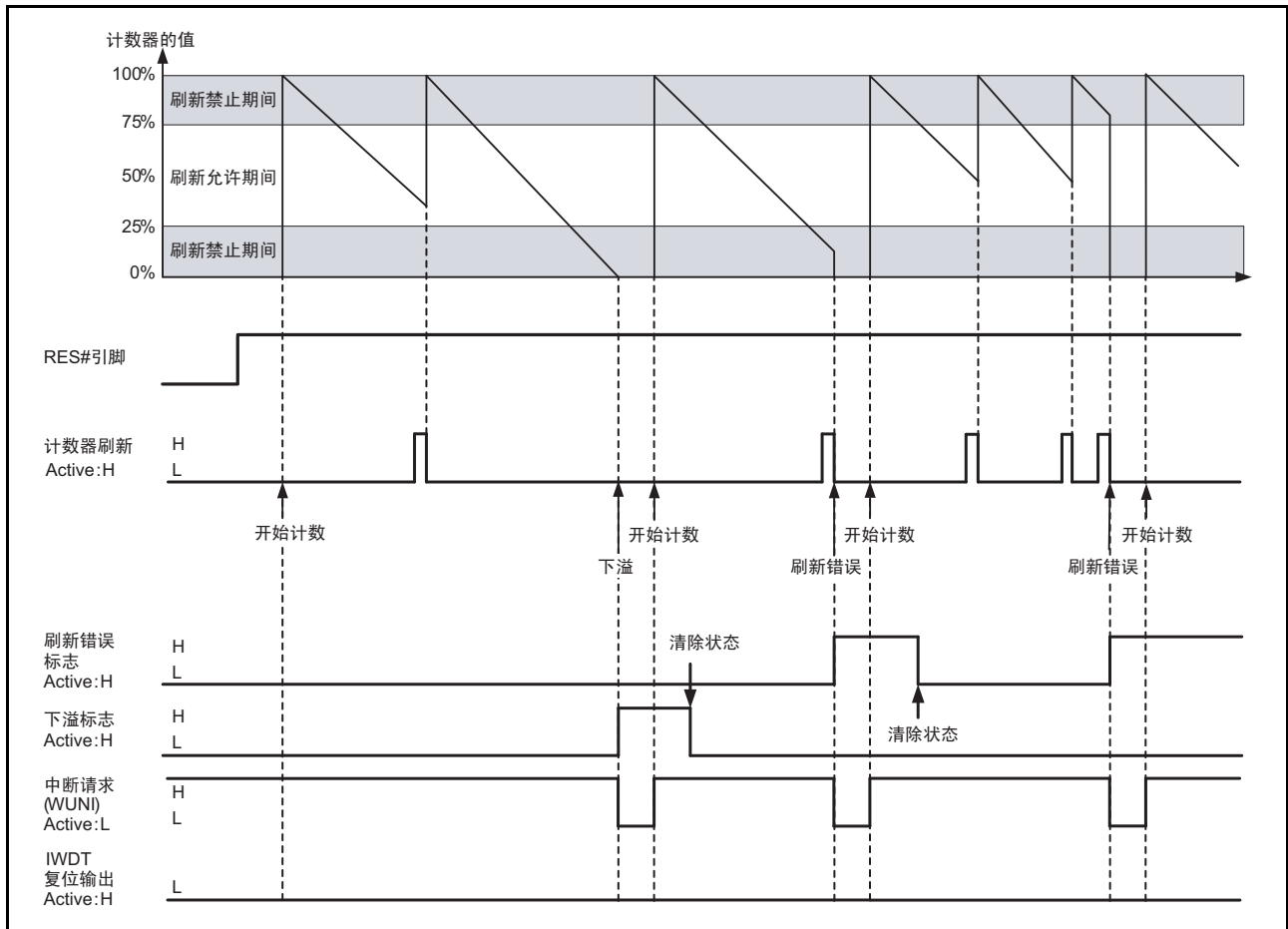


图 26.4 自动启动模式的运行例子

26.3.2 IWDTCR 寄存器、IWDTRCR 寄存器和 IWDTCS TPR 寄存器的写控制

在从复位解除后到第 1 次刷新的期间，只能写 1 次 IWDT 控制寄存器 (IWDTCR)、IWDT 复位控制寄存器 (IWDTRCR) 和 IWDT 计数停止控制寄存器 (IWDTCS TPR)。

如果刷新 (开始计数) 或者写 IWDTCR、IWDTRCR、IWDTCS TPR 寄存器，IWDT 的内部保护信号就变为“1”，此后保护对 IWDTCR、IWDTRCR 和 IWDTCS TPR 寄存器的写操作。

通过 IWDT 的复位源而不通过其他复位源解除保护。

IWDTCR 寄存器的写控制波形如图 26.5 所示。

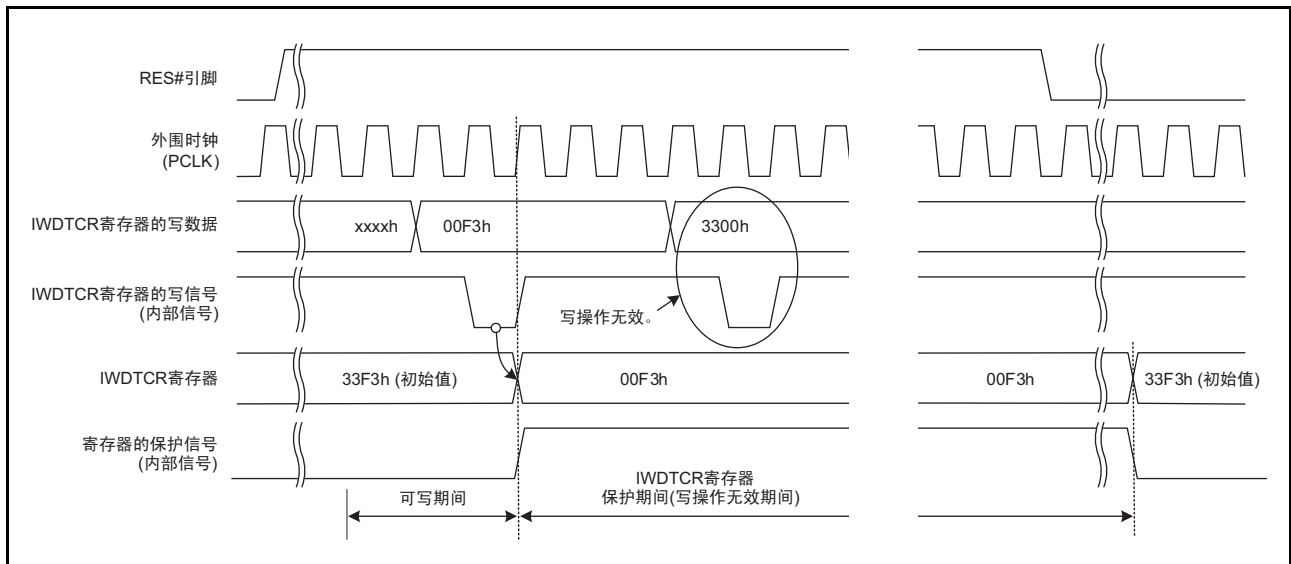


图 26.5 IWDTCR 寄存器的写控制波形

26.3.3 刷新

要刷新递减计数器并且使递减计数器开始运行 (通过刷新开始计数) 时，在给 IWDT 刷新寄存器 (IWDTRR) 写“00h”后接着写“FFh”。如果在写“00h”后写“FFh”以外的值，就不进行刷新。能通过再次按照“00h”→“FFh”的顺序写 IWDTRR 寄存器，正常地进行刷新。

即使按照“00h” (第 1 次) →“00h” (第 2 次) 的顺序进行写操作，因为之后通过写“FFh”能使“00h”→“FFh”顺序的写操作成立，所以“00h” (第 n-1 次) →“00h” (第 n 次) →“FFh”的写操作也有效并且进行刷新。同样，即使在“00h”之前写的不是“00h”，只要“00h”→“FFh”顺序的写操作成立，也进行刷新。另外，在给 IWDTRR 寄存器写“00h”和“FFh”的期间，即使存取 IWDTRR 以外的寄存器或者读 IWDTRR 寄存器，也进行刷新。

【刷新有效的写操作例子】

“00h”→“FFh”

“00h” (第 n-1 次) →“00h” (第 n 次) →“FFh”

“00h”→存取其他寄存器或者读 IWDTRR 寄存器 →“FFh”

【刷新无效的写操作例子】

“23h” (不是“00h”) →“FFh”

“00h”→“54h” (不是“FFh”)

“00h”→“AAh” (不是“00h”和“FFh”) →“FFh”

作为刷新运行，即使在刷新允许期间外给 IWDTRR 寄存器写“00h”，只要在刷新允许期间内给 IWDTRR 寄存器写“FFh”，写操作就成立并且进行刷新。

在给 IWDTRR 寄存器写“FFh”后，递减计数器的刷新时序最多需要 4 个计数周期（1 个周期期间的 IWDT 专用低速时钟 (IWDTCLK) 数因时钟分频比选择位 (IWDTCR.CKS[3:0]) 的设定值而不同）。因此，在从刷新允许期间结束位置开始到 4 个计数周期或者递减计数器发生下溢的 4 个计数周期前为止，必须给 IWDTRR 寄存器写完“FFh”。能通过递减计数器值的位 (IWDTSR.CNTVAL[13:0]) 确认递减计数器的值。

【刷新运行的时序例子】

- 当窗口起始位置为“1FFFh”时，即使在给 IWDTRR 寄存器写“00h”后写“1FFFh”（例如“2002h”），只要在 IWDTSR.CNTVAL[13:0] 位的值变为“1FFFh”后给 IWDTRR 寄存器写“FFh”，也进行刷新。
- 当窗口结束位置为“1FFFh”时，如果在按照“00h”→“FFh”的顺序写 IWDTRR 寄存器后立即读 IWDTSR.CNTVAL[13:0] 位的值并且读取值大于等于“2003h”（“1FFFh”的 4 个计数时钟前），就进行刷新。
- 当刷新允许期间持续到“0000h”时，能在下溢前进行刷新，此时，如果在按照“00h”→“FFh”的顺序写 IWDTRR 寄存器后立即读 IWDTSR.CNTVAL[13:0] 位的值并且读取值大于等于“0003h”（下溢的 4 个计数时钟前），就不产生下溢而进行刷新。

PCLK > IWDTCLK 并且时钟分频比为 IWDTCLK 时的 IWDT 的刷新波形如图 26.6 所示，PCLK < IWDTCLK 并且时钟分频比为 IWDTCLK/16 时的 IWDT 的刷新波形如图 26.7 所示。

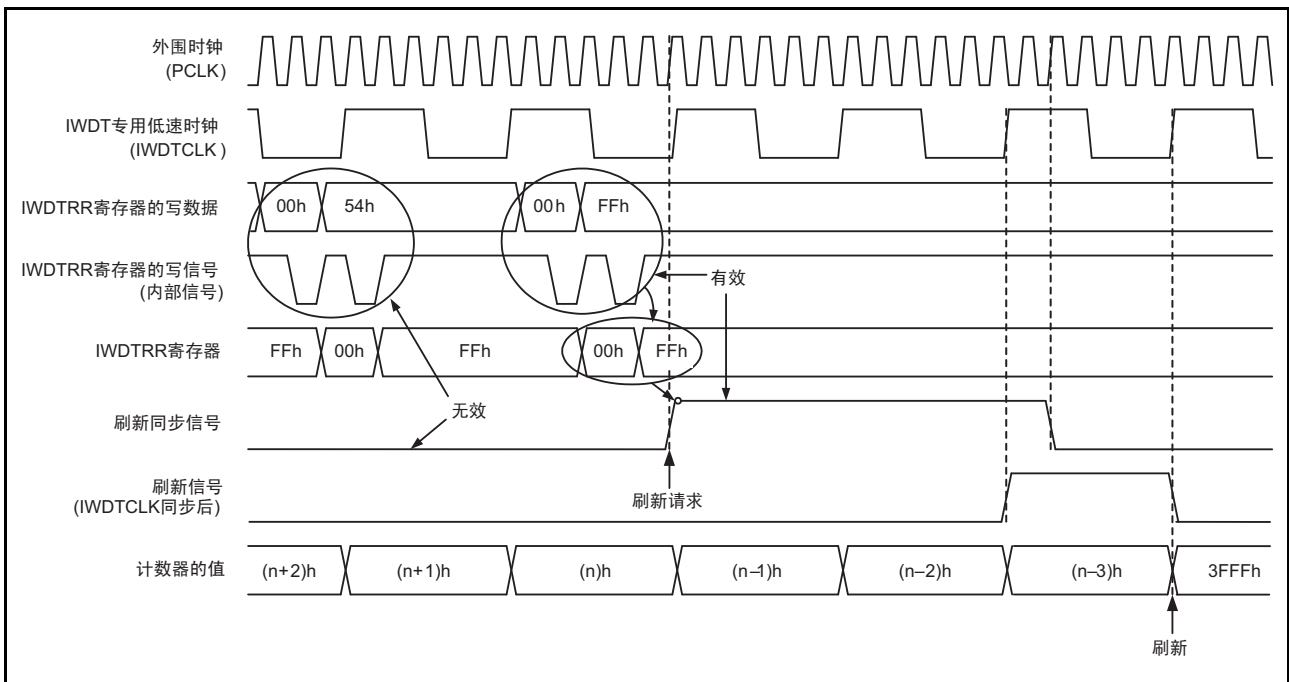


图 26.6 IWDT 的刷新波形 (IWDTCR.CKS[3:0]=0000b、IWDTCR.TOPS[1:0]=11b)

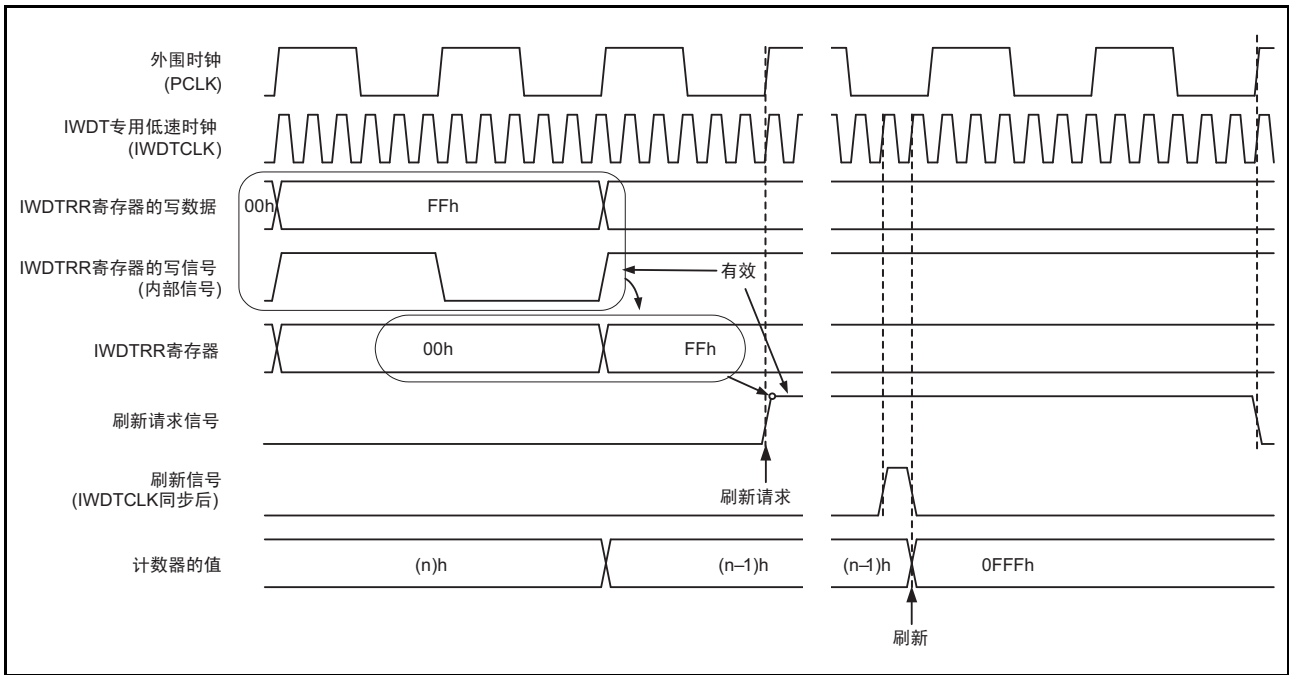


图 26.7 IWDT 的刷新波形 (IWDTCR.CKS[3:0]=0010b、IWDTCR.TOPS[1:0]=01b)

26.3.4 状态标志

刷新错误标志 (IWDTSR.REFEF) 和下溢标志 (IWDTSR.UNDF) 保持 IWDT 输出复位时的复位源或者发生中断请求时的中断源。

在解除复位后或者在发生中断请求时，能通过读 IWDTSR.REFEF 标志或者 IWDTSR.UNDF 标志，确认复位源或者中断源的发生状态。

要将各标志的值置“0”时，必须写“0”。写“1”无效。

即使不将各标志置“0”，也不影响运行。如果不置“0”，就在下次 IWDT 输出复位时清除以前的复位源并且写新的复位源，或者在下次 IWDT 发生中断请求时清除以前的中断源并且写新的中断源。

另外，要读通过写“0”清除标志后被反映的值时，需要几个 IWDTCLK 时钟（至少 3 个时钟）和几个 PCLK 时钟（至少 2 个时钟）。

26.3.5 复位输出

如果在寄存器启动模式中将复位中断选择位 (IWDTRCR.RSTIRQS) 置“1”或者在自动启动模式中将选项功能选择寄存器 0 的 IWDT 复位中断请求选择位 (OFS0.IWDRSTIRQS) 置“1”，就在发生递减计数器下溢或者刷新错误时输出 1 个计数周期的复位。

在寄存器启动模式中，在输出复位后递减计数器保持初始状态 (ALL“0”)。在解除复位并且重新启动后，通过刷新重新设定计数器值并且开始递减计数。

在自动启动模式中，在输出复位后自动开始递减计数。

26.3.6 中断源

如果在寄存器启动模式中将复位中断选择位 (IWDTRCR.RSTIRQS) 置“0”或者在自动启动模式中将选项功能选择寄存器 0 的 IWDT 复位中断请求选择位 (OFS0.IWDTRSTIRQS) 置“0”，就在发生递减计数器下溢或者刷新错误时产生非屏蔽中断 (WUNI)。

表 26.4 IWDT 的中断源

名称	中断源	DTC 的启动	DMAC 的启动
WUNI	递减计数器的下溢 刷新错误	不能	不能

26.3.7 递减计数器值的读操作

因为 IWDT 的递减计数器通过 IWDT 专用低速时钟 (IWDTCLK) 运行，所以不能直接读计数器值。因此，IWDT 通过外围时钟 (PCLK) 与计数器值同步并且保存到 IWDT 状态寄存器的递减计数器 (IWDTSR.CNTVAL[13:0] 位)。能通过读被保存到 IWDTSR.CNTVAL[13:0] 位的值，间接确认计数器值。

因为读操作需要几个 PCLK 时钟 (最多 4 个时钟)，所以读到的计数值和递减计数器的实际值有可能相差 1 个计数时钟。

PCLK > IWDTCLK 并且时钟分频比为 IWDTCLK 时的 IWDT 递减计数器值的读处理如图 26.8 所示，PCLK < IWDTCLK 并且时钟分频比为 IWDTCLK/16 时的 IWDT 递减计数器值的读处理如图 26.9 所示。

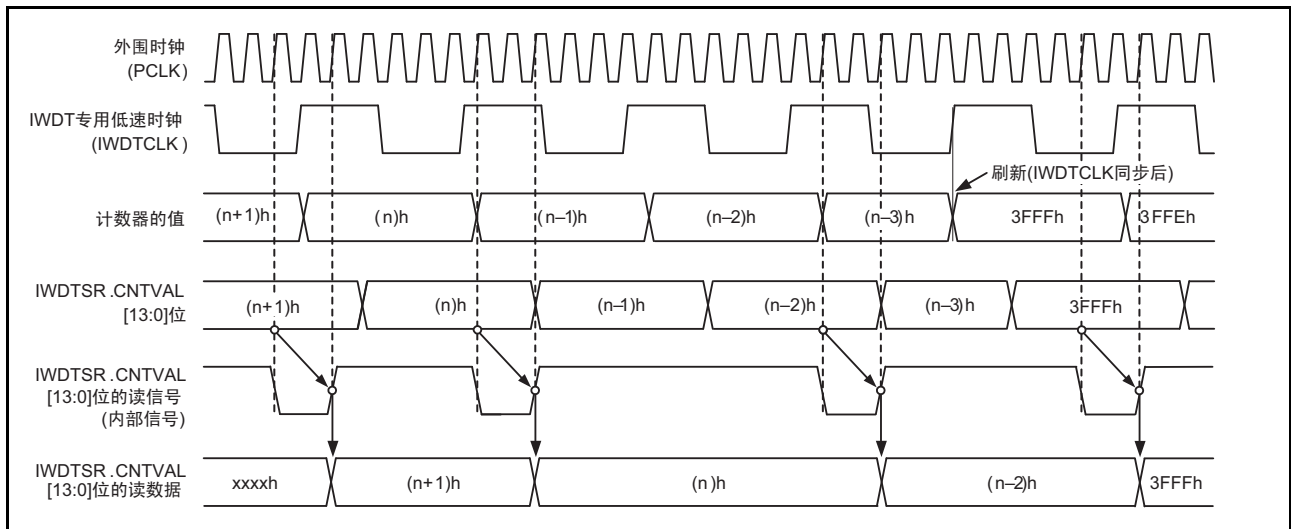


图 26.8 IWDT 递减计数器值的读处理 (IWDTCR.CKS[3:0]=0000b、IWDTCR.TOPS[1:0]=11b)

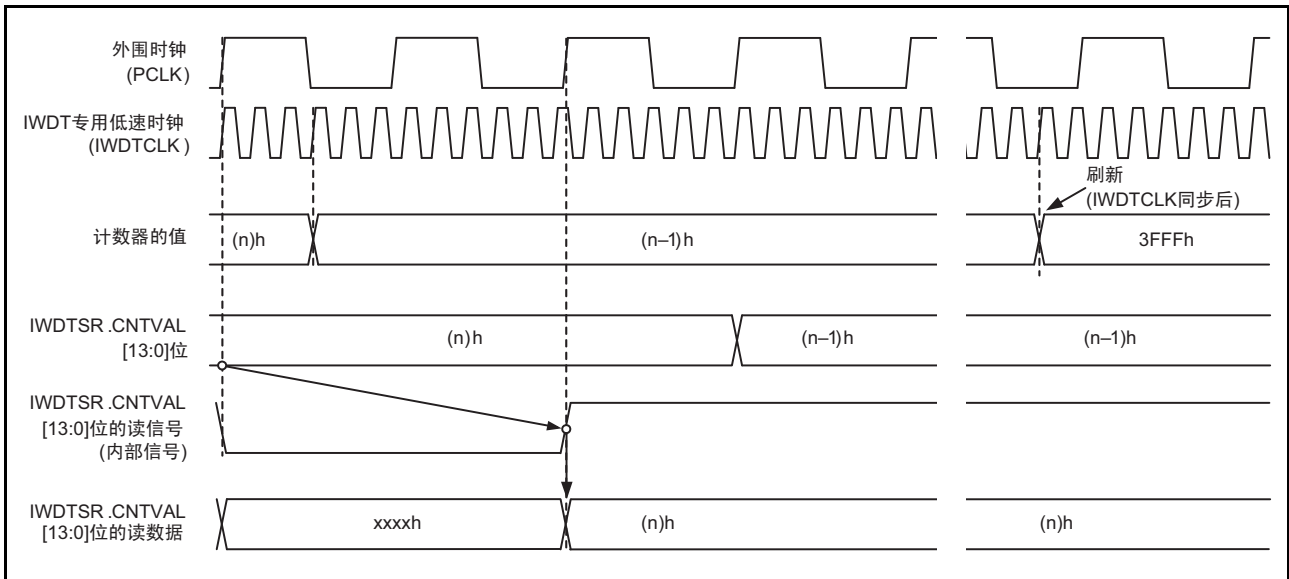


图 26.9 IWDT 递减计数器值的读处理 (IWDTCR.CKS[3:0]=0010b、IWDTCR.TOPS[1:0]=11b)

26.3.8 选项功能选择寄存器 0 (OFS0) 和 IWDT 寄存器的对应

通过选项功能选择寄存器 0 (OFS0) 进行的递减计数器控制、复位输出控制、中断请求输出控制、计数停止控制和 IWDT 控制寄存器 (IWDTCR)、IWDT 复位控制寄存器 (IWDTRCR)、IWDT 计数停止控制寄存器 (IWDTCSSTPR) 的对应如表 26.5 所示。通过 IWDT 启动模式选择位 (OFS0.IWDTSTRT) 选择 OFS0 寄存器和 IWDTCR、IWDTRCR、IWDTCSSTPR 寄存器控制的有效或者无效。

必须在 IWDT 运行过程中固定 OFS0 寄存器的设定。

有关 OFS0 寄存器，请参照“7.2.1 选项功能选择寄存器 0 (OFS0)”。

表 26.5 选项功能选择寄存器 0 (OFS0) 和 IWDT 寄存器的对应

控制	功能	OFS0 寄存器 (在自动启动模式中有效) OFS0.IWDTSTRT=0	IWDT 寄存器 (在寄存器启动模式中有效) OFS0.IWDTSTRT=1
递减计数器	选择超时期间。	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	选择时钟分频比。	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	选择窗口起始位置。	OFS0.IWDRPSS[1:0]	IWDTCR.RPSS[1:0]
	选择窗口结束位置。	OFS0.IWDRPES[1:0]	IWDTCR.RPES[1:0]
复位输出 / 中断请求输出	选择输出复位或者中断请求。	OFS0.IWDRSTIRQS	IWDTCR.RSTIRQS
计数停止	选择在睡眠模式中停止计数。	OFS0.IWDTSLCSTP	IWDTCSSTPR.SLCSTP

26.4 使用时的注意事项

26.4.1 刷新

在设定刷新时序时，必须考虑 PCLK 和 IWDTCLK 的精度，设定在误差范围内即使周期发生变化也能刷新新的值。

27. 串行通信接口 (SCIc、SCIId)

RX63T 群具有 3 个独立通道的串行通信接口 (SCI: Serial Communication Interface)。SCI 由 SCIc 模块 (SCI0、SCI1) 和 SCIId 模块 (SCI12) 构成。

SCIc (SCI0、SCI1) 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, 支持对应 ISO/IEC 7816-3 (Identification Card) 规格的智能卡 (IC 卡) 接口, 并且支持简易 I²C 总线接口的单主控模式的运行和简易 SPI 接口。

SCIId (SCI12) 除了 SCIc 的功能以外, 还支持由 Start Frame 和 Information Frame 构成的扩展串行通信协议。

27.1 概要

SCIc 的规格、SCIId 的规格和 SCI 各通道功能一览表分别如表 27.1、表 27.2 和表 27.3 所示。

SCI0、SCI1 的框图和 SCI12 (SCIId) 的框图分别如图 27.1 和图 27.2 所示。

表 27.1 SCIc 的规格 (1/2)

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 • 简易 I²C 总线 • 简易 SPI 接口
传送速度		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 27.4 ~ 表 27.6。
数据传送		可选择 LSB first 或者 MSB first。(注 1)
中断源		发送结束、发送数据空、接收数据满、接收错误、开始条件 / 重新开始条件 / 停止条件生成结束 (用于简易 I ² C 模式)
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
	硬件流程控制	能控制使用 CTSn 引脚和 RTSn 引脚的发送 / 接收。
	中止的检测	在发生帧错误时, 能通过直接读 RXDn 引脚的电平来检测中止。
	时钟源	可选择内部时钟或者外部时钟。
	多处理器通信功能	多个处理器之间的串行通信功能
	噪声消除	在 RXDn 引脚的输入线路中内置数字噪声消除器。
时钟同步模式	数据长度	8 位
	接收错误的检测	溢出错误
	硬件流程控制	能控制使用 CTSn 引脚和 RTSn 引脚的发送 / 接收。
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	支持正向协议或者反向协议。

表 27.1 SC1c 的规格 (2/2)

项目	内容	
简易 I ² C 模式	通信格式	I ² C 总线格式
	运行模式	主控 (只限于单主控模式的运行)
	传送速度	最大 384kbps
	噪声消除	在 SSCLn 引脚和 SSDAn 引脚的输入线路中内置数字噪声消除器。 能调整噪声消除的宽度。
简易 SPI 模式	数据长度	8 位
	错误的检测	溢出错误
	SS 输入引脚功能	当 SSn# 引脚为 High 电平时, 能将输出引脚置为高阻抗。
	时钟设定	可选择 4 种时钟相位和时钟极性的设定。

注 1. 在简易 I²C 模式中只能使用 MSB first。

表 27.2 SC1d 的规格 (1/2)

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 • 简易 I²C 总线 • 简易 SPI 接口 	
传送速度	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 27.4 ~ 表 27.7。	
数据传送	可选择 LSB first 或者 MSB first。(注 1)	
中断源	发送结束、发送数据空、接收数据满、接收错误、开始条件 / 重新开始条件 / 停止条件生成结束 (用于简易 I ² C 模式)	
低功耗功能	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
	硬件流程控制	能控制使用 CTSn 引脚和 RTSn 引脚的发送 / 接收。
	中止的检测	在发生帧错误时, 能通过直接读 RXDn 引脚的电平来检测中止。
	时钟源	可选择内部时钟或者外部时钟。
	多处理器通信功能	多个处理器之间的串行通信功能
时钟同步模式	数据长度	8 位
	接收错误的检测	溢出错误
	硬件流程控制	能控制使用 CTSn 引脚和 RTSn 引脚的发送 / 接收。
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	支持正向协议或者反向协议。

表 27.2 SCId 的规格 (2/2)

项目	内容	
简易 I ² C 模式	通信格式	I ² C 总线格式
	运行模式	主控 (只限于单主控模式的运行)
	传送速度	最大 384kbps
	噪声消除	在 SSCLn 引脚和 SSDAn 引脚的输入线路中内置数字噪声消除器。 能调整噪声消除的宽度。
简易 SPI 模式	数据长度	8 位
	错误的检测	溢出错误
	SS 输入引脚功能	当 SSn# 引脚为 High 电平时, 能将输出引脚置为高阻抗。
	时钟设定	可选择 4 种时钟相位和时钟极性的设定。
扩展串行模式	Start Frame 发送	<ul style="list-style-type: none"> 能输出 Break Field Low width, 有输出结束中断功能。 有总线冲突检测功能和检测中断功能。
	Start Frame 接收	<ul style="list-style-type: none"> 能检测 Break Field Low width, 有检测结束中断功能。 有 Control Field 0 和 Control Field 1 的数据比较 / 相同中断功能。 能在 Control Field 1 中设定主 / 次 2 种比较数据。 能在 Control Field 1 中设定优先级中断位。 也支持没有 Break Field 的 Start Frame。 也支持没有 Control Field 0 的 Start Frame。 有位数率测量功能
	输入 / 输出控制功能	<ul style="list-style-type: none"> 可选择 TXDX12/RXDX12 信号的极性。 能给 RXDX12 信号设定数字滤波器功能。 能进行兼用 RXDX12 引脚和 TXDX12 引脚的半双工通信。 可选择 RXDX12 引脚接收数据的采样时序。 当扩展串行模式控制部为 OFF 时, 能将 RXDX12 接收信号直接输出到 SCIc。
	定时器功能	<ul style="list-style-type: none"> 能用作重加载定时器功能。

注 1. 在简易 I²C 模式中只能使用 MSB first。

表 27.3 SCI 通道功能一览表

项目	SCI0、SCI1	SCI12
异步模式	○	○
时钟同步模式	○	○
智能卡接口模式	○	○
简易 I ² C 模式	○	○
简易 SPI 模式	○	○
扩展串行模式	—	○

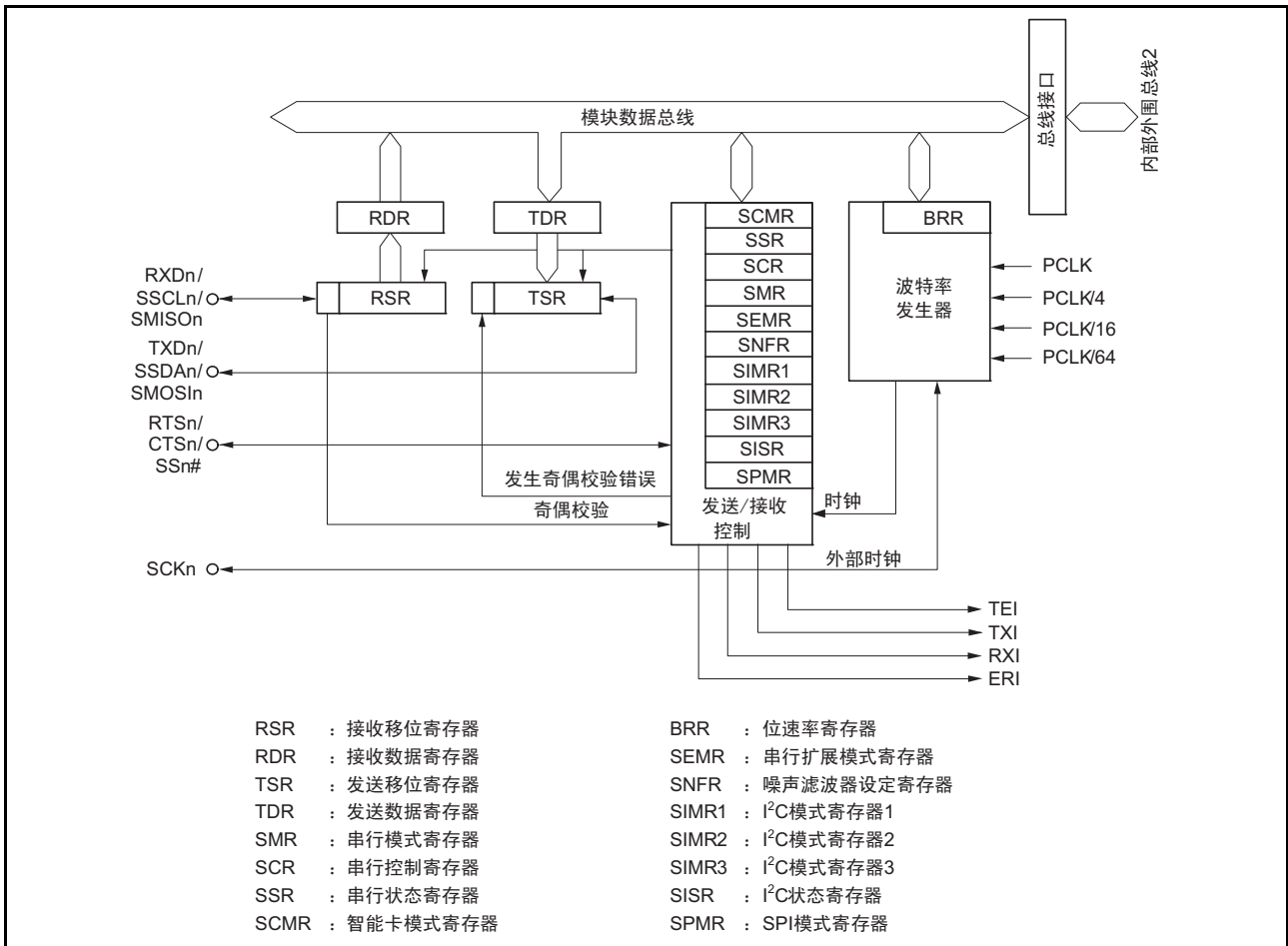


图 27.1 SCl0、SCl1 的框图

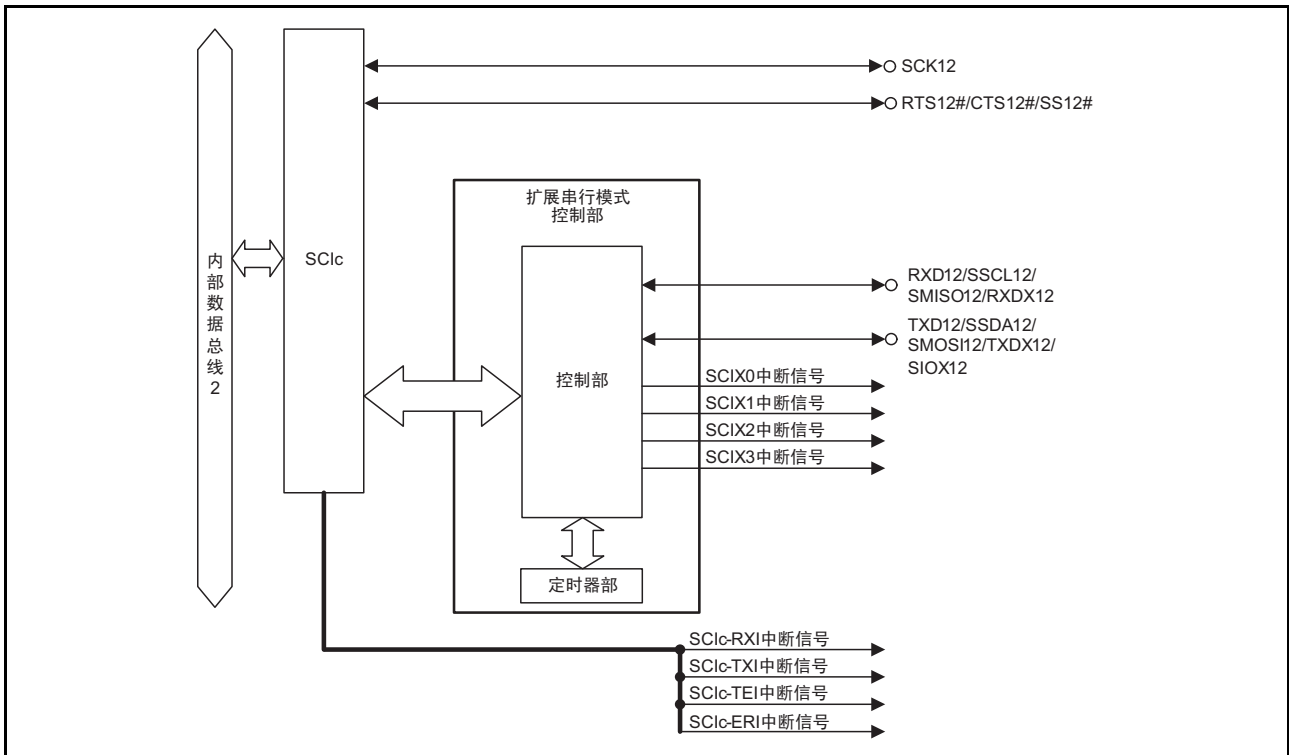


图 27.2 SCl12 (SCId) 的框图

各模式中的 SCI 使用的输入 / 输出引脚如表 27.4 ~ 表 27.7 所示。

表 27.4 SCI 的输入 / 输出引脚 (异步 / 时钟同步模式)

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
	CTS0#/RTS0#	输入 / 输出	SCI0 发送 / 接收开始控制的输入 / 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
	CTS1#/RTS1#	输入 / 输出	SCI1 发送 / 接收开始控制的输入 / 输出引脚
SCI12	SCK12	输入 / 输出	SCI12 的时钟输入 / 输出引脚
	RXD12	输入	SCI12 的接收数据输入引脚
	TXD12	输出	SCI12 的发送数据输出引脚
	CTS12#/RTS12#	输入 / 输出	SCI12 发送 / 接收开始控制的输入 / 输出引脚

表 27.5 SCI 的输入 / 输出引脚 (简易 I²C 模式)

通道	引脚名	输入 / 输出	功能
SCI0	SSCL0	输入 / 输出	SCI0 的 I ² C 时钟输入 / 输出引脚
	SSDA0	输入 / 输出	SCI0 的 I ² C 数据输入 / 输出引脚
SCI1	SSCL1	输入 / 输出	SCI1 的 I ² C 时钟输入 / 输出引脚
	SSDA1	输入 / 输出	SCI1 的 I ² C 数据输入 / 输出引脚
SCI12	SSCL12	输入 / 输出	SCI12 的 I ² C 时钟输入 / 输出引脚
	SSDA12	输入 / 输出	SCI12 的 I ² C 数据输入 / 输出引脚

表 27.6 SCI 的输入 / 输出引脚 (简易 SPI 模式)

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
	SMISO0	输入 / 输出	SCI0 的从属发送数据输入 / 输出引脚
	SMOSI0	输入 / 输出	SCI0 的主控发送数据输入 / 输出引脚
	SS0#	输入	SCI0 的片选输入引脚
SCI1	SCK1	输入 / 输出	SCI1 的时钟输入 / 输出引脚
	SMISO1	输入 / 输出	SCI1 的从属发送数据输入 / 输出引脚
	SMOSI1	输入 / 输出	SCI1 的主控发送数据输入 / 输出引脚
	SS1#	输入	SCI1 的片选输入引脚
SCI12	SCK12	输入 / 输出	SCI12 的时钟输入 / 输出引脚
	SMISO12	输入 / 输出	SCI12 的从属发送数据输入 / 输出引脚
	SMOSI12	输入 / 输出	SCI12 的主控发送数据输入 / 输出引脚
	SS12#	输入	SCI12 的片选输入引脚

表 27.7 SCI 的输入 / 输出引脚 (扩展串行模式)

通道	引脚名	输入 / 输出	功能
SCI12	RDX12	输入	SCI12 的接收数据输入引脚
	TXDX12	输出	SCI12 的发送数据输出引脚
	SIOX12	输入 / 输出	SCI12 发送 / 接收数据的输入 / 输出引脚

27.2 寄存器说明

27.2.1 接收移位寄存器 (RSR)

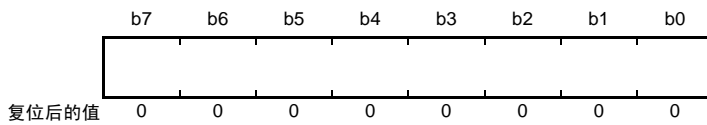
RSR 寄存器是将 RXDn 引脚输入的串行数据转换为并行数据的接收移位寄存器。

如果接收 1 帧的数据，数据就自动传送到 RDR 寄存器。

CPU 不能直接存取 RSR 寄存器。

27.2.2 接收数据寄存器 (RDR)

地址 SCI0.RDR 0008 A005h、SCI1.RDR 0008 A025h、SCI12.RDR 0008 B305h



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。

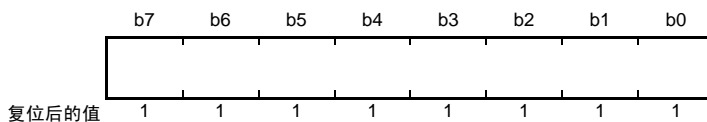
因为 RSR 寄存器和 RDR 寄存器是双缓冲结构，所以能连续接收。

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。

27.2.3 发送数据寄存器 (TDR)

地址 SCI0.TDR 0008 A003h、SCI1.TDR 0008 A023h、SCI12.TDR 0008 B303h



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。

27.2.4 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。

27.2.5 串行模式寄存器 (SMR)

注. 在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 A000h、SCI1.SMR 0008 A020h、SCI12.SMR 0008 B300h

b7	b6	b5	b4	b3	b2	b1	b0
CM	CHR	PE	PM	STOP	MP	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	MP	多处理器模式位	(只在异步模式中有效) 0: 禁止多处理器通信功能 1: 允许多处理器通信功能	R/W (注4)
b3	STOP	停止位长度位	(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	字符长度位	(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“27.2.9 位速率寄存器 (BRR)”中的 n 的值。

注 2. 在时钟同步模式中, 设定无效, 数据长度固定为 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。

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“27.2.9 位速率寄存器 (BRR) ”。

MP 位 (多处理器模式位)

此位选择禁止或者允许多处理器通信功能。在多处理器模式中，PE 位和 PM 位的设定无效。

STOP 位 (停止位长度位)

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)。

在多处理器模式中，PM 位的设定无效。

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。

与 PE 位的设定无关，不在多处理器格式中附加奇偶校验位，也不进行奇偶校验。

CHR 位 (字符长度位)

此位选择发送 / 接收数据的数据长度。

在时钟同步模式中，数据长度固定为 8 位。

(2) 智能卡接口模式 (SCMR.SMIF 位 =1)

地址 SCI0.SMR 0008 A000h、SCI1.SMR 0008 A020h、SCI12.SMR 0008 B300h

b7	b6	b5	b4	b3	b2	b1	b0
GM	BLK	PE	PM	BCP[1:0]	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	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: 在正常模式中运行 1: 在块传送模式中运行	R/W (注3)
b7	GM	GSM 模式位	0: 在正常模式中运行 1: 在 GSM 模式中运行	R/W (注3)

注1. n 为设定值的 10 进制数, 表示“27.2.9 位速率寄存器 (BRR)”中的 n 的值。

注2. S 表示“27.2.9 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。

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“27.2.9 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“27.6.4 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“27.6.2 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“27.6.3 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 标志的置位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“27.6.6 串行数据的发送 (块传送模式除外)”和“27.6.8 时钟的输出控制”。

27.2.6 串行控制寄存器 (SCR)

注. 在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 A002h、SCI1.SCR 0008 A022h、SCI12.SCR 0008 B302h

b7	b6	b5	b4	b3	b2	b1	b0
TIE	RIE	TE	RE	MPIE	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	(异步模式) b1 b0 0 0: 内部波特率发生器 根据 I/O 端口的设定, 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 x: 外部时钟 必须从 SCKn 引脚输入频率为 16 倍位速率的时钟。在 SEMR.ABCS 位为“1”时, 必须输入 8 倍频率的时钟。 (时钟同步模式) b1 b0 0 x: 内部时钟 SCKn 引脚为时钟的输出引脚。 1 x: 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 中断请求 1: 允许 TEI 中断请求	R/W
b3	MPIE	多处理器中断允许位	(在异步模式中 SMR.MP 位为“1”时有效) 0: 通常的接收运行 1: 跳读多处理器位为“0”的接收数据, 禁止 SSR. ORER 和 SSR.FER 各状态标志的置位 (“1”)。如果接收到多处理器位是“1”的数据, 就自动清除 (“0”) MPIE 位, 返回到通常的接收运行。	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 中断请求和 ERI 中断请求 1: 允许 RXI 中断请求和 ERI 中断请求	R/W
b7	TIE	发送中断允许位	0: 禁止 TXI 中断请求 1: 允许 TXI 中断请求	R/W

x: Don't care

注 1. 只能在 TE 位和 RE 位都为“0”时写这些位。

注 2. 如果 SMR.CM 位为“1”, 就只能在 TE 位和 RE 位都为“0”时写“1”。一旦将 TE 位或者 RE 位置“1”, 就只能给 TE 位和 RE 位写“0”。如果 SMR.CM 位为“0”, 就能在任意时候写此位。

CKE[1:0] 位 (时钟允许位)

这些位选择时钟源和 SCK_n 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 中断请求。

能通过将 TEIE 位置“0”来禁止 TEI 中断请求。

在简易 I²C 模式 (SIMR1.IICM=1) 中, 将开始 / 重新开始 / 停止条件生成结束中断 (STI 中断) 分配给 TEI 中断。此时, 也能通过 TEIE 位允许或者禁止 STI 中断请求。

MPIE 位 (多处理器中断允许位)

如果将 MPIE 位置“1”, 就跳读多处理器位为“0”的接收数据, 并且 SSR. ORER 和 SSR. FER 的各状态标志不变为“1”。如果接收多处理器位为“1”的数据, MPIE 位就自动清除并且返回到通常的接收运行。详细内容请参照“27.4 多处理器通信功能”。

在接收包括 SSR.MPB 位为“0”的接收数据时, 不将接收数据从 RSR 寄存器传送到 RDR 寄存器, 也不检测接收错误并且不将 ORER 和 FER 的各标志置位 (“1”)。

如果接收包括 MPB 位为“1”的接收数据, 就将 MPB 位置“1”, MPIE 位自动变为“0”, 并且允许 RXI 中断请求和 ERI 中断请求 (SCR 的 RIE 位为“1”的情况) 以及允许 ORER 和 FER 各标志的置位 (“1”)。

在不使用多处理器通信功能时, 不能给 MPIE 位写“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, 如果在异步模式中检测到起始位, 或者在时钟同步模式中检测到同步时钟输入, 就开始串行接收。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 和 ERI 的中断请求。

能通过将 RIE 位置“0”来禁止 RXI 中断请求。

通过在从 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”, 解除 ERI 中断请求。

TIE 位 (发送中断允许位)

此位允许或者禁止通知 TXI 中断请求。

能通过将 TIE 位置“0”来禁止 TXI 中断请求。

(2) 智能卡接口模式 (SCMR.SMIF 位 =1)

地址 SCI0.SCR 0008 A002h、SCI1.SCR 0008 A022h、SCI12.SCR 0008 B302h

b7	b6	b5	b4	b3	b2	b1	b0
TIE	RIE	TE	RE	MPIE	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 位为“0”的情况 b1 b0 0 0: 禁止输出 (根据 I/O 端口的设定, SCKn 引脚能用作输入 / 输出端口) 0 1: 时钟输出 1 x: 不能设定 SMR.GM 位为“1”的情况 b1 b0 0 0: 固定为 Low 电平输出 x 1: 时钟输出 1 0: 固定为 High 电平输出 	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	MPIE	多处理器中断允许位	在智能卡接口模式中, 必须置“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 中断请求和 ERI 中断请求 1: 允许 RXI 中断请求和 ERI 中断请求	R/W
b7	TIE	发送中断允许位	0: 禁止 TXI 中断请求 1: 允许 TXI 中断请求	R/W

x: Don't care

注 1. 只能在 TE 位和 RE 位都为“0”时写这些位。

注 2. 如果 SMR.CM 位为“1”, 就只能在 TE 位和 RE 位都为“0”时写“1”。一旦将 TE 位或者 RE 位置“1”, 就只能给 TE 位和 RE 位写“0”。如果 SMR.CM 位为“0”, 就能在任意时候写此位。

有关各中断源, 请参照“27.11 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 模式中动态更改时钟的输出, 详细内容请参照“27.6.8 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”。

MPIE 位 (多处理器中断允许位)

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 和 ERI 的中断请求。

能通过将 RIE 位置“0”来禁止 RXI 中断请求。

通过在从 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”, 解除 ERI 中断请求。

TIE 位 (发送中断允许位)

此位允许或者禁止通知 TXI 中断请求。

能通过将 TIE 位置“0”来禁止 TXI 中断请求。

27.2.7 串行状态寄存器 (SSR)

注. 在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 A004h、SCI1.SSR 0008 A024h、SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	ORER	FER	PER	TEND	MPB	MPBT
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
b0	MPBT	多处理器位的传送位	设定给发送帧附加的多处理器位的值。	R/W
b1	MPB	多处理器位	是接收帧中的多处理器位的值。	R
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. 为了清除标志, 只能写“0”。

MPB 位 (多处理器位)

此位保存接收帧中的多处理器位的值。此位在 SCR.RE 位为“0”时不变。

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]

- 当 SCR.TE 位为“0” (禁止串行发送) 时
在将 SCR.TE 位从“0”置为“1”时, TEND 标志不受影响而保持“1”的状态。
- 在发送字符的最后一位时未更新 TDR 寄存器时

[为“0”的条件]

- 在 SCR.TE 位为“1”的状态下给 TDR 寄存器写发送数据时
在通过给 TDR 寄存器写发送数据来清除 TEND 标志时, 必须读 TEND 标志, 确认 TEND 标志是“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数据时发生了奇偶校验错误并且异常结束。

[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标志被置“1”的状态下，以后的接收数据不传送到RDR寄存器。

[为“0”的条件]

- 在读到“1”的状态后写“0”（必须在写“0”后确认PER标志已被清除）时
即使将SCR.RE位清“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数据时发生了帧错误并且异常结束。

[为“1”的条件]

- 当停止位为“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标志被置“1”的状态下，以后的接收数据不传送到RDR寄存器。

[为“0”的条件]

- 在读到“1”的状态后写“0”（必须在写“0”后确认FER标志已被清除）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)

此标志表示在接收数据时发生了溢出错误并且异常结束。

[为“1”的条件]

- 当不读RDR寄存器的接收数据而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标志被置“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]

- 在读到“1”的状态后写“0”（必须在写“0”后确认ORER标志已被清除）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 位 =1)

地址 SCI0.SSR 0008 A004h、SCI1.SSR 0008 A024h、SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	ORER	ERS	PER	TEND	MPB	MPBT
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
b0	MPBT	多处理器位的传送位	在智能卡接口模式中，必须置“0”。	R/W
b1	MPB	多处理器位	在智能卡接口模式中不使用，必须置“0”。	R
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. 为了清除标志，只能写“0”。

MPBT 位 (多处理器位的传送位)

在智能卡接口模式中，必须将此位置“0”。

MPB 位 (多处理器位)

在智能卡接口模式中，不使用此位，必须将此位置“0”。

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，将此标志置“1”。

[为“1”的条件]

- 当 SCR.TE 位为“0” (禁止串行发送) 时
在将 SCR.TE 位从“0”置为“1”时，TEND 标志不受影响而保持“1”的状态。
- 在发送 1 字节数据后的一定时间之后 ERS 标志变为“0”并且未更新 TDR 寄存器时
根据寄存器的设定，置位的时序如下：
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]

- 在 SCR.TE 位为“1”的状态下给 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数据时发生了奇偶校验错误并且异常结束。

[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标志被置“1”的状态下，以后的接收数据不传送到RDR寄存器。

[为“0”的条件]

- 在读到“1”的状态后写“0”（必须在写“0”后确认PER标志已被清除）时
即使将SCR.RE位清“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]

- 当对错误信号的Low电平进行采样时

[为“0”的条件]

- 在读到“1”的状态后写“0”时

ORER 标志 (溢出错误标志)

此标志表示在接收数据时发生了溢出错误并且异常结束。

[为“1”的条件]

- 当不读RDR寄存器的接收数据而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标志被置“1”的状态下，不能继续进行以后的串行接收。

[为“0”的条件]

- 在读到“1”的状态后写“0”（必须在写“0”后确认ORER标志已被清除）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。

27.2.8 智能卡模式寄存器 (SCMR)

地址 SCI0.SCMR 0008 A006h、SCI1.SCMR 0008 A026h、SCI12.SCMR 0008 B306h

b7	b6	b5	b4	b3	b2	b1	b0
BCP2	—	—	—	SDIR	SINV	—	SMIF
复位后的值	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	发送 / 接收数据反相位	0: 将 TDR 寄存器的内容直接发送, 并且将接收数据直接保存到 RDR 寄存器。 1: 将 TDR 寄存器的内容取反后发送, 并且将接收数据取反后保存到 RDR 寄存器。	R/W (注1)
b3	SDIR	发送 / 接收数据传送方向位 (注2)	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3) 0 0 1: 128 个时钟 (S=128) (注3) 0 1 0: 186 个时钟 (S=186) (注3) 0 1 1: 512 个时钟 (S=512) (注3) 1 0 0: 32 个时钟 (S=32) (注3) (初始值) 1 0 1: 64 个时钟 (S=64) (注3) 1 1 0: 372 个时钟 (S=372) (注3) 1 1 1: 256 个时钟 (S=256) (注3)	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. 在选择简易 I²C 模式时, 必须将 SDIR 位设定为“1”（MSB first）。

注 3. S 表示“27.2.9 位速率寄存器 (BRR)”中的 S 的值。

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 (发送 / 接收数据反相位)

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 (发送 / 接收数据传送方向位)

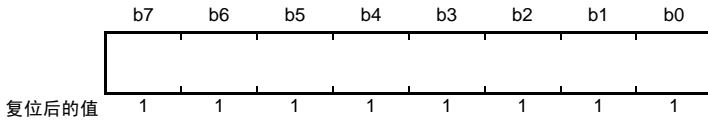
选择串行 / 并行转换的方向

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。

27.2.9 位速率寄存器 (BRR)

地址 SCI0.BRR 0008 A001h、SCI1.BRR 0008 A021h、SCI12.BRR 0008 B301h



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、多处理器通信、时钟同步模式、智能卡接口模式、简易 SPI 模式和简易 I²C 模式中，BRR 寄存器的设定值 N 和位速率 B 的关系如表 27.8 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 27.8 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、多处理器通信	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、简易SPI		$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
智能卡接口		$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	误差(%) = $\left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
简易I ² C(注1)		$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B : 位速率 (bps)

N : 波特率发生器的 BRR 设定值 (0 ≤ N ≤ 255)

PCLK : 工作频率 (MHz)

n 和 S : 取决于下表中 SMR 的设定值。

注 1. 必须调整位数率，使简易 I²C 模式中的 SCL 输出的 High/Low 电平宽度满足 I²C 规格。

表 27.9 SCL High/Low 电平宽度的计算式

模式	SCL	计算式 (秒 (s))
I ² C	High 宽度 (min 值)	$(N + 1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Low 宽度 (min 值)	$(N + 1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$

表 27.10 时钟源的设定

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

表 27.11 智能卡接口模式中的基本时钟设定

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 27.12 和表 27.13 所示，在各工作频率下能设定的最大位速率如表 27.13 所示；时钟同步模式和简易 SPI 模式中的 BRR 寄存器值 N 的设定例子如表 27.15 所示，智能卡接口模式中的 BRR 寄存器值 N 的设定例子如表 27.15 所示，简易 I²C 模式中的 BRR 寄存器值 N 的设定例子如表 27.15 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“27.6.4 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 27.14 和表 27.16 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 27.12 所示的位速率的 2 倍。

表 27.12 对应位速率的 BRR 设定例子 (异步模式) (1)

位速率 (bps)	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

表 27.12 对应位速率的 BRR 设定例子 (异步模式) (2)

位速率 (bps)	工作频率 PCLK (MHz)											
	12.288			14			16			17.2032		
	n	N	误差 (%)	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	3	75	0.48
150	2	159	0.00	2	181	0.16	2	207	0.16	2	223	0.00
300	2	79	0.00	2	90	0.16	2	103	0.16	2	111	0.00
600	1	159	0.00	1	181	0.16	1	207	0.16	1	223	0.00
1200	1	79	0.00	1	90	0.16	1	103	0.16	1	111	0.00
2400	0	159	0.00	0	181	0.16	0	207	0.16	0	223	0.00
4800	0	79	0.00	0	90	0.16	0	103	0.16	0	111	0.00
9600	0	39	0.00	0	45	-0.93	0	51	0.16	0	55	0.00
19200	0	19	0.00	0	22	-0.93	0	25	0.16	0	27	0.00
31250	0	11	2.40	0	13	0.00	0	15	0.00	0	16	1.20
38400	0	9	0.00	—	—	—	0	12	0.16	0	13	0.00

注. 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 27.12 对应位速率的 BRR 设定例子 (异步模式) (3)

位速率 (bps)	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

表 27.12 对应位速率的 BRR 设定例子 (异步模式) (4)

位速率 (bps)	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

注. 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 27.13 各工作频率下的最大位速率 (异步模式)

PCLK (MHz)	最大位速率 (bps)	n	N
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0
25	781250	0	0
30	937500	0	0
33	1031250	0	0
50	1562500	0	0

注. 如果将 SEMR.ABCS 位置“1”，位速率就变为原来的 2 倍。

表 27.14 输入外部时钟时的最大位速率 (异步模式)

PCLK (MHz)	外部输入时钟 (MHz)	最大位速率 (bps)	
		SEMR.ABCS 位 =0	SEMR.ABCS 位 =1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000
25	6.2500	390625	781250
30	7.5000	468750	937500
33	8.2500	515625	1031250
50	12.500	781250	1562500

注. 这是 SEMR.ABCS 位为“0”时的例子。

表 27.15 对应位速率的 BRR 设定例子 (时钟同步模式、简易 SPI 模式)

位速率 (bps)	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	—	—	—	—	—	—	—	—
2M	0	0 ^(注1)	—	—	0	1	—	—	—	—	—	—	—	—	—	—
2.5M			0	0 ^(注1)			0	1	—	—	0	2	—	—	0	4
4M					0	0 ^(注1)	—	—	—	—	—	—	—	—	—	—
5M							0	0 ^(注1)	—	—	—	—	—	—	—	—
6.25M									0	0 ^(注1)	—	—	—	—	0	1
7.5M											0	0 ^(注1)	—	—	—	—
8.25M													0	0 ^(注1)	—	—

空栏：不能设定。

—：能设定，但是会出现误差。

注 1. 不能进行连续的发送和接收。

表 27.16 输入外部时钟时的最大位速率 (时钟同步模式、简易 SPI 总线模式)

PCLK (MHz)	外部输入时钟 (MHz)	最大位速率 (bps)
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3
25	4.1667	4166666.7
30	5.0000	5000000.0
33	5.5000	5500000.0
50	8.3333	8333333.3

表 27.17 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)

位数率 (bps)	PCLK (MHz)	n	N	误差 (%)
9600	7.1424	0	0	0.00
	10.00	0	1	30
	10.7136	0	1	25
	13.00	0	1	8.99
	14.2848	0	1	0.00
	16.00	0	1	12.01
	18.00	0	2	15.99
	20.00	0	2	6.66
	25.00	0	3	12.49
	30.00	0	3	5.01
	33.00	0	4	7.59
50.00	0	6	0.01	

表 27.18 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 (bps)	n	N
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
16.00	21505	0	0
18.00	24194	0	0
20.00	26882	0	0
25.00	33602	0	0
30.00	40323	0	0
33.00	44355	0	0
50.00	67205	0	0

表 27.19 对应位速率的 BRR 设定例子 (简易 I²C 模式) (1)

位速率 (bps)	工作频率 PCLK (MHz)											
	8			10			16			20		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
10k	0	24	0.0	0	31	-2.3	1	12	-3.8	1	15	-2.3
25k	0	9	0.0	0	12	-3.8	1	4	0.0	1	6	-10.7
50k	0	4	0.0	0	6	-10.7	1	2	-16.7	1	3	-21.9
100k	0	2	-16.7	0	3	-21.9	0	4	0.0	0	6	-10.7
250k	0	0	0.0	0	1	-37.5	0	1	0.0	0	2	-16.7
350k										0	1	-10.7

表 27.19 对应位速率的 BRR 设定例子 (简易 I²C 模式) (2)

位速率 (bps)	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 (%)
10k	1	19	-2.3	1	23	-2.3	1	25	-0.8	2	9	-2.3
25k	1	7	-2.3	1	9	-6.3	1	10	-6.3	2	3	-2.3
50k	1	3	-2.3	1	4	-6.3	1	5	-14.1	2	1	-2.3
100k	1	1	-2.3	1	2	-21.9	1	2	-14.1	1	3	-2.3
250k	0	3	-21.9	0	3	-6.3	0	4	-17.5	0	6	-10.7
350k	0	2	-25.6	0	2	-10.7	0	2	-1.8	0	4	-10.7

表 27.20 各位速率设定的 SCL High/Low 电平宽度最小值 (简易 I²C 模式) (1)

SCL High/Low 电平宽度 min 值 (μs)	工作频率 PCLK (MHz)											
	8			10			16			20		
	n	N	High/Low 电平宽度	n	N	High/Low 电平宽度	n	N	High/Low 电平宽度	n	N	High/Low 电平宽度
10k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.5/52.00	1	15	44.80/51.20
25k	0	9	17.50/20.00	0	12	18.2/20.80	1	4	17.50/20.00	1	6	19.60/22.40
50k	0	4	8.75/10.00	0	6	9.80/11.20	1	2	10.50/12.00	1	3	11.20/12.80
100k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.37/5.00	0	6	4.90/5.60
250k	0	0	1.75/2.00	0	1	2.80/3.20	0	1	1.75/2.00	0	2	2.10/2.40
350k										0	1	1.40/1.60

表 27.20 各位速率设定的 SCL High/Low 电平宽度最小值 (简易 I²C 模式) (2)

SCL High/Low 电平宽度 min 值 (μs)	工作频率 PCLK (MHz)											
	25			30			33			50		
	n	N	High/Low 电平宽度	n	N	High/Low 电平宽度	n	N	High/Low 电平宽度	n	N	High/Low 电平宽度
10k	1	19	44.80/51.20	1	23	44.80/51.20	1	25	44.12/50.42	2	9	44.80/51.20
25k	1	7	17.92/20.48	1	9	18.66/21.33	1	10	18.66/21.33	2	3	17.92/20.48
50k	1	3	8.96/10.24	1	4	9.33/10.66	1	5	10.18/11.63	2	1	8.96/10.24
100k	1	1	4.48/5.12	1	2	5.60/6.40	1	2	5.09/5.81	1	3	4.48/5.12
250k	0	3	2.24/2.56	0	3	1.86/2.13	0	4	2.12/2.42	0	6	1.96/2.24
350k	0	2	1.68/1.92	0	2	1.40/1.60	0	2	1.27/1.45	0	4	1.40/1.60

27.2.10 串行扩展模式寄存器 (SEMR)

地址 SCI0.SEMR 0008 A007h、SCI1.SEMR 0008 A027h、SCI12.SEMR 0008 B307h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	NFEN	ABCS	—	—	—	—
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	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)
b5	NFEN	数字噪声滤波器功能允许位	(异步模式) 0: RXDn 输入信号的噪声消除功能无效 1: RXDn 输入信号的噪声消除功能有效 (简易 I ² C 模式) 0: SSCLn 和 SSDAn 输入信号的噪声消除功能无效 1: SSCLn 和 SSDAn 输入信号的噪声消除功能有效 在上述以外的模式中, 必须将 NFEN 位置“0”。	R/W (注 1)
b7-b6	—	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。

NFEN 位 (数字噪声滤波器功能允许位)

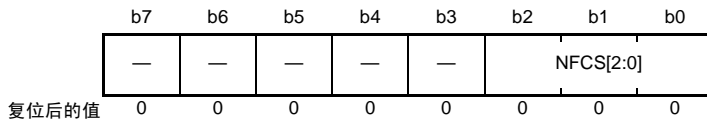
此位选择数字噪声滤波器功能的有效或者无效。

如果置为有效, 就在异步模式中消除 RXDn 输入信号的噪声, 而在简易 I²C 模式中消除 SSDAn 和 SSCLn 输入信号的噪声。

在上述以外的模式中, 必须将 NFEN 位置“0”, 使数字噪声滤波器功能无效。如果将数字噪声滤波器功能置为无效, 输入信号就直接作为内部信号而被传递。

27.2.11 噪声滤波器的设定寄存器 (SNFR)

地址 SCI0.SNFR 0008 A008h、SCI1.SNFR 0008 A028h、SCI12.SNFR 0008 B308h



位	符号	位名	功能	R/W
b2-b0	NFCS[2:0]	噪声滤波器的时钟选择位	<p>在异步模式中，以基本时钟为基准。</p> <p>b2 b0 0 0 0: 将 1 分频时钟用于噪声滤波器</p> <p>在简易 I²C 模式中，以 SMR.CKS[1:0] 位所选内部波特率发生器的时钟源为基准。</p> <p>b2 b0 0 0 1: 将 1 分频时钟用于噪声滤波器 0 1 0: 将 2 分频时钟用于噪声滤波器 0 1 1: 将 4 分频时钟用于噪声滤波器 1 0 0: 将 8 分频时钟用于噪声滤波器</p> <p>上述以外：不能设定</p>	R/W (注 1)
b7-b3	—	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）时写这些位。

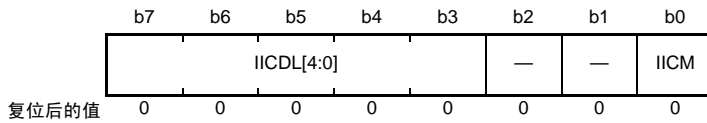
NFCS[2:0] 位（噪声滤波器的时钟选择位）

这些位选择数字噪声滤波器的采样时钟。

在异步模式中使用噪声滤波器的情况下，必须置“000b”；在简易 I²C 模式中使用噪声滤波器的情况下，必须置“001b”～“100b”中的任意一个值。

27.2.12 I²C 模式寄存器 1（SIMR1）

地址 SCI0.SIMR1 0008 A009h、SCI1.SIMR1 0008 A029h、SCI12.SIMR1 0008 B309h



位	符号	位名	功能	R/W
b0	IICM	简易 I ² C 模式选择位	SMIF IICM 0 0: 串行接口模式 (异步模式、时钟同步模式或者简易 SPI 模式) 0 1: 简易 I ² C 模式 1 0: 智能卡接口模式 1 1: 不能设定	R/W (注 1)
b2-b1	—	保留位	读写值都为“0”。	R/W
b7-b3	IICDL[4:0]	SSDA 输出延迟选择位	(以内部波特率发生器的时钟源为基准) b7 b3 00000: 没有输出延迟 00001: 0~1 个周期 00010: 1~2 个周期 00011: 2~3 个周期 00100: 3~4 个周期 00101: 4~5 个周期 : 11110: 29~30 个周期 11111: 30~31 个周期	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）时写这些位。

SIMR1 寄存器是用于选择简易 I²C 模式和 SSDA 输出延迟段数的寄存器。

IICM 位（简易 I²C 模式选择位）

此位通过和 SCMR.SMIF 位的组合来选择运行模式。

IICDL[4:0] 位（SSDA 输出延迟选择位）

这些位选择相对于 SSCLn 引脚输出的下降沿的 SSDAn 引脚输出的延迟。能将内部波特率发生器的时钟源作为 1 个周期，选择无延迟~ 31 个周期。内部波特率发生器的时钟源是指通过设定 SMR.CKS[1:0] 位将 PCLK 进行分频的时钟。在简易 I²C 模式中必须将这些位置“00001b”~“11111b”中的任意一个值，在其他模式中必须置“00000b”。

27.2.13 I²C 模式寄存器 2（SIMR2）

地址 SCI0.SIMR2 0008 A00Ah、SCI1.SIMR2 0008 A02Ah、SCI12.SIMR2 0008 B30Ah

b7	b6	b5	b4	b3	b2	b1	b0
—	—	IICACKT	—	—	—	IICCSC	IICINTM
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	IICINTM	I ² C 中断模式选择位	0: 使用 ACK/NACK 中断 1: 使用接收中断和发送中断	R/W (注1)
b1	IICCSC	时钟同步位	0: 不进行时钟同步 1: 进行时钟同步	R/W (注1)
b4-b2	—	保留位	读写值都为“0”。	R/W
b5	IICACKT	ACK 发送数据位	0: 发送 ACK 1: 发送 NACK 或者接收 ACK/NACK	R/W
b7-b6	—	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）时写这些位。

SIMR2 寄存器是用于选择简易 I²C 模式的发送 / 接收控制的寄存器。

IICINTM 位（I²C 中断模式选择位）

此位选择简易 I²C 模式中的中断请求源。

IICCSC 位（时钟同步位）

在为了使其他设备插入等待而将 SSCLn 引脚置为 Low 电平的情况下，如果要与内部生成的 SSCLn 时钟取得同步，就将 IICCSC 位置“1”。

如果将 IICCSC 位置“0”，就不进行 SSCLn 时钟的同步。与 SSCLn 引脚的输入无关，根据 BRR 寄存器设定的位速率生成 SSCLn 时钟。

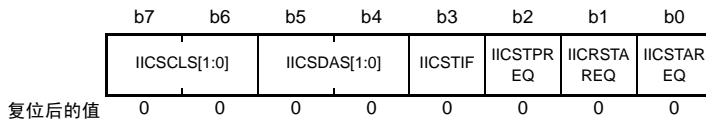
除了调试以外，必须将 IICCSC 位置“1”。

IICACKT 位（ACK 发送数据位）

此位保存发送数据的 ACK 位。必须在接收 ACK/NACK 位时将此位置“1”。

27.2.14 I²C 模式寄存器 3 (SIMR3)

地址 SCI0.SIMR3 0008 A00Bh、SCI1.SIMR3 0008 A02Bh、SCI12.SIMR3 0008 B30Bh



位	符号	位名	功能	R/W
b0	IICSTAREQ	开始条件生成位	0: 不生成开始条件 1: 生成开始条件 (注1、注3、注4)	R/W
b1	IICRSTAREQ	重新开始条件生成位	0: 不生成重新开始条件 1: 生成重新开始条件 (注2、注3、注4)	R/W
b2	IICSTPREQ	停止条件生成位	0: 不生成停止条件 1: 生成停止条件 (注2、注3、注4)	R/W
b3	IICSTIF	开始 / 重新开始 / 停止条件的生成结束标志	0: 没有各条件生成请求的状态或者正在生成的状态 1: 各条件生成结束的状态	R/W
b5-b4	IICSDAS[1:0]	SSDA 输出选择位	b5 b4 0 0: 串行数据输出 0 1: 生成开始条件、重新开始条件和停止条件 1 0: SSDAn 引脚输出 Low 电平 1 1: SSDAn 引脚为高阻抗状态	R/W
b7-b6	IICSCLS[1:0]	SSCL 输出选择位	b7 b6 0 0: 串行时钟输出 0 1: 生成开始条件、重新开始条件和停止条件 1 0: SSCLn 引脚输出 Low 电平 1 1: SSCLn 引脚为高阻抗状态	R/W

注 1. 必须确认总线状态并且在总线空闲的状态下生成开始条件。

注 2. 必须确认总线状态并且在总线忙的状态下生成重新开始条件或者停止条件。

注 3. 不能将 IICSTAREQ 位、IICRSTAREQ 位或者 IICSTPREQ 位中的 2 位或者全部置“1”。

注 4. 必须在将 IICSTIF 标志置“0”后生成各条件。

SIMR3 寄存器用于对简易 I²C 模式的开始条件和停止条件的生成以及对 SSDAn 引脚和 SSCLn 引脚输出值的固定进行控制。

IICSTAREQ 位 (开始条件生成位)

在生成开始条件时, 必须在将 IICSTAREQ 位置“1”的同时将 IICSDAS[1:0] 位和 IICSCLS[1:0] 位分别置“01b”。

[为“1”的条件]

- 当写“1”时

[为“0”的条件]

- 当开始条件生成结束时

IICRSTAREQ 位 (重新开始条件生成位)

在生成重新开始条件时, 必须在将 IICRSTAREQ 位置“1”的同时将 IICSDAS[1:0] 位和 IICSCLS[1:0] 位分别置“01b”。

[为“1”的条件]

- 当写“1”时

[为“0”的条件]

- 当重新开始条件生成结束时

IICSTPREQ 位 (停止条件生成位)

在生成停止条件时, 必须在将 IICSTPREQ 位置“1”的同时将 IICSDAS[1:0] 位和 IICSCLS[1:0] 位分别置“01b”。

[为“1”的条件]

- 当写“1”时

[为“0”的条件]

- 当停止条件生成结束时

IICSTIF 标志 (开始 / 重新开始 / 停止条件的生成结束标志)

此标志表示在生成各条件后生成结束的状态。必须在将 IICSTIF 标志置“0”后, 通过 IICSTAREQ 位、IICRSTAREQ 位和 IICSTPREQ 位生成各条件。

在通过 SCR.TEIE 位允许中断请求的状态下, 在 IICSTIF 标志为“1”时输出开始 / 重新开始 / 停止条件的生成结束中断 (STI) 请求。

[为“1”的条件]

- 当开始、重新开始和停止的各条件生成结束时 (如果和为“0”的条件发生冲突, 就优先为“0”的条件。)

[为“0”的条件]

- 当写“0”时 (必须确认 IICSTIF 标志已变为“0”)
- 当 SIMR1.IICM 位为“0”时 (简易 I²C 模式除外)
- 当 SCR.TE 位为“0”时

IICSDAS 位 (SSDA 输出选择位)

此位控制 SSDA_n 引脚的输出。

在正常运行时, 必须给 IICSDAS 位和 IICSCLS 位设定相同的值。

IICSCLS 位 (SSCL 输出选择位)

此位控制 SSCL_n 引脚的输出。

在正常运行时, 必须给 IICSCLS 位和 IICSDAS 位设定相同的值。

27.2.15 I²C 状态寄存器 (SISR)

地址 SC10.SISR 0008 A00Ch、SC11.SISR 0008 A02Ch、SC112.SISR 0008 B30Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IICACKR
复位后的值	0	0	x	x	0	x	0	0

x: 不定值

位	符号	位名	功能	R/W
b0	IICACKR	ACK 接收数据标志	0: 接收 ACK 1: 接收 NACK	R/W (注1)
b1	—	保留位	读写值都为“0”。	R/W
b2	—	保留位	读取值为不定值。	R
b3	—	保留位	读写值都为“0”。	R/W
b5-b4	—	保留位	读取值为不定值。	R
b7-b6	—	保留位	读写值都为“0”。	R/W

注 1. 为了清除标志, 只能写“0”。

SISR 寄存器监视简易 I²C 模式的相关状态。

IICACKR 标志 (ACK 接收数据标志)

此标志能读已接收的 ACK/NACK 位。

在接收 ACK/NACK 的位的 SSCLn 时钟上升时更新 IICACK 标志。

27.2.16 SPI 模式寄存器 (SPMR)

地址 SCI0.SPMR 0008 A00Dh、SCI1.SPMR 0008 A02Dh、SCI12.SPMR 0008 B30Dh

b7	b6	b5	b4	b3	b2	b1	b0
CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	SSE	SS 引脚功能允许位	0: 禁止 SS 引脚功能 1: 允许 SS 引脚功能	R/W (注1)
b1	CTSE	CTS 允许位	0: 禁止 CTS 功能 (RTS 输出功能有效) 1: 允许 CTS 功能	R/W (注1)
b2	MSS	主控从属选择位	0: TXDn 引脚: 发送、RXDn 引脚: 接收 (主控模式) 1: TXDn 引脚: 接收、RXDn 引脚: 发送 (从属模式)	R/W (注1)
b3	—	保留位	读写值都为“0”。	R/W
b4	MFF	模式故障标志	0: 没有模式故障错误 1: 有模式故障错误	R/W (注2)
b5	—	保留位	读写值都为“0”。	R/W
b6	CKPOL	时钟极性选择位	0: 没有时钟极性反相 1: 有时钟极性反相	R/W (注1)
b7	CKPH	时钟相位选择位	0: 没有时钟延迟 1: 有时钟延迟	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) 时写这些位。

注2. 为了清除标志, 只能写“0”。

SPMR 寄存器是用于选择异步模式和时钟同步模式扩展设定的寄存器。

SSE 位 (SS 引脚功能允许位)

在使用 SSn# 引脚进行发送和接收控制时 (简易 SPI 模式), 将此位置“1”, 而在其他通信模式中, 必须将此位置“0”。即使在简易 SPI 模式中, 如果用于主控模式 (SCR.CKE[1:0] 位 =00b 并且 MSS 位 =0) 的单主控, 就不需要使用主控侧的 SSn# 引脚进行发送和接收控制, 并且将 SSE 位置“0”。不能将 SSE 位和 CTSE 位都置为有效 (如果设定, 两位就都变为无效)。

CTSE 位 (CTS 允许位)

在将 SSn# 引脚用作 CTS 控制信号输入引脚进行发送和接收控制时, 将此位置“1”。在将此位置“0”的状态下输出 RTS 信号。在智能卡接口模式、简易 SPI 模式和简易 I²C 模式中, 必须将此位置“0”。不能将 CTSE 位和 SSE 位都置为有效 (即使设定也无效)。

MSS 位 (主控从属选择位)

此位在简易 SPI 模式中选择主控模式和从属模式。如果将 MSS 位置“1”，TXDn 引脚和 RXDn 引脚的功能就相反，从 TXDn 引脚输入接收数据并且从 RXDn 引脚输出发送数据。

在简易 SPI 模式以外的模式中，必须将此位置“0”。

MFF 标志 (模式故障标志)

此标志表示发生了模式故障错误。

在多主控模式中，必须通过读 MFF 标志来判断模式故障错误。

[为“1”的条件]

- 在设定为简易 SPI 模式的主控模式 (SSE 位=1 并且 MSS 位=0) 的情况下 SSn# 引脚的输入变为 Low 电平时。

[为“0”的条件]

- 在读“1”的状态后写“0”时

CKPOL 位 (时钟极性选择位)

此位选择 SCKn 引脚的时钟输出极性，详细内容请参照图 27.50。

在简易 SPI 模式和时钟同步模式以外的模式中，必须将此位置“0”。

CKPH 位 (时钟相位选择位)

此位选择 SCKn 引脚的时钟输出的相位设定，详细内容请参照图 27.50。

在简易 SPI 模式和时钟同步模式以外的模式中，必须将此位置“0”。

27.2.17 扩展串行模式有效寄存器（ESMER）

地址 SC112.ESMER 0008 B320h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	ESME
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	ESME	扩展串行模式有效位	0: 扩展串行模式无效 1: 扩展串行模式有效	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

ESME 位（扩展串行模式有效位）

当 ESME 位为“1”时，扩展串行模式控制部有效。

当将 ESME 位置“0”时，为以下状态：

- 扩展串行模式控制部为被初始化的状态。

表 27.21 ESME 位的设定和定时器运行模式的运行保证

ESME 位	定时器模式	Break Field Low width 判定模式	Break Field Low width 输出模式
0	○（注1）	×	×
1	○	○	○

○：需要运行保证。×：不需要运行保证。

注 1. 只在选择 PCLK 时运行。

27.2.18 控制寄存器 0（CR0）

地址 SC112.CR0 0008 B321h

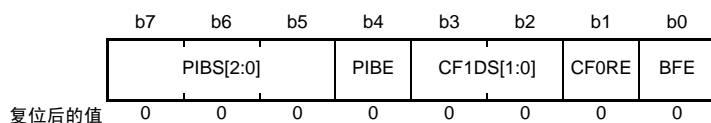
b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	BRME	RXDSF	SFSF	—
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	—	保留位	读写值都为“0”。	R/W
b1	SFSF	Start Frame 状态标志	0: Start Frame 检测功能无效状态 1: Start Frame 检测功能有效状态	R
b2	RXDSF	RXDX12 输入状态标志	0: RXDX12 输入允许状态 1: RXDX12 输入禁止状态	R
b3	BRME	位速率测量允许位	0: 位速率测量无效 1: 位速率测量有效	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

27.2.19 控制寄存器 1 (CR1)

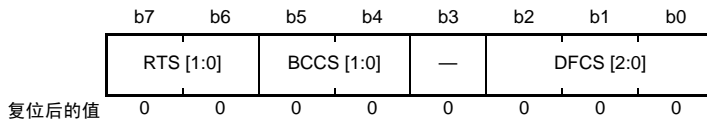
地址 SCI12.CR1 0008 B322h



位	符号	位名	功能	R/W
b0	BFE	Break Field 允许位	0: Break Field 的检测无效 1: Break Field 的检测有效	R/W
b1	CF0RE	Control Field 0 接收允许位	0: Control Field 0 的接收无效 1: Control Field 0 的接收有效	R/W
b3-b2	CF1DS[1:0]	Control Field 1 数据寄存器选择位	b3 b2 0 0: 选择 PCF1DR 作为比较数据 0 1: 选择 SCF1DR 作为比较数据 1 0: 选择 PCF1DR 和 SCF1DR 作为比较数据 1 1: 不能设定	R/W
b4	PIBE	优先级中断位的允许位	0: 优先级中断位无效 1: 优先级中断位有效	R/W
b7-b5	PIBS[2:0]	优先级中断位的选择位	b7 b5 0 0 0: Control Field 1 第 0 位 0 0 1: Control Field 1 第 1 位 0 1 0: Control Field 1 第 2 位 0 1 1: Control Field 1 第 3 位 1 0 0: Control Field 1 第 4 位 1 0 1: Control Field 1 第 5 位 1 1 0: Control Field 1 第 6 位 1 1 1: Control Field 1 第 7 位	R/W

27.2.20 控制寄存器 2 (CR2)

地址 SCI12.CR2 0008 B323h



位	符号	位名	功能	R/W
b2-b0	DFCS[2:0]	RXD12 信号数字滤波器的时钟选择位	b2 b0 0 0 0: 滤波器无效 0 0 1: 滤波器有效 (SCI 基本时钟) 0 1 0: 滤波器有效 (SCI 基本时钟 /8) 0 1 1: 滤波器有效 (SCI 基本时钟 /16) 1 0 0: 滤波器有效 (SCI 基本时钟 /32) 1 0 1: 滤波器有效 (SCI 基本时钟 /64) 1 1 0: 滤波器有效 (SCI 基本时钟 /128) 1 1 1: 不能设定	R/W
b3	—	保留位	读写值都为“0”。	R/W
b5-b4	BCCS[1:0]	总线冲突检测时钟选择位	b5 b4 0 0: SCI 基本时钟 0 1: SCI 基本时钟的 2 分频 1 0: SCI 基本时钟的 4 分频 1 1: 不能设定	R/W
b7-b6	RTS[1:0]	RXD12 接收采样时序选择位	<ul style="list-style-type: none"> • SCI12.SEMR.ABCS 位为“0”的情况 b7 b6 0 0: SCI 基本时钟的第 8 个上升沿 0 1: SCI 基本时钟的第 10 个上升沿 1 0: SCI 基本时钟的第 12 个上升沿 1 1: SCI 基本时钟的第 14 个上升沿 • SCI12.SEMR.ABCS 位为“1”的情况 b7 b6 0 0: SCI 基本时钟的第 4 个上升沿 0 1: SCI 基本时钟的第 5 个上升沿 1 0: SCI 基本时钟的第 6 个上升沿 1 1: SCI 基本时钟的第 7 个上升沿 	R/W

注. 当 SCI12.SEMR.ABCS 为“0”时, SCI 基本时钟为 1 个数据期间的 1/16 个周期; 当 SCI12.SEMR.ABCS 为“1”时, SCI 基本时钟为 1 个数据期间的 1/8 个周期。在使用 SCI 基本时钟的情况下, 必须将 SCI12.SCR.TE 位置“1”。

27.2.21 控制寄存器 3（CR3）

地址 SCI12.CR3 0008 B324h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	SDST
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	SDST	Start Frame 检测开始位	0: 不进行 Start Frame 的检测 1: 进行 Start Frame 的检测	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

SDST 位（Start Frame 检测开始位）

如果将 SDST 位置“1”，就开始检测 Start Frame。读取值为“0”。

27.2.22 端口控制寄存器（PCR）

地址 SCI12.PCR 0008 B325h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	SHARPS	—	—	RDXPS	TXDXPS
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	TXDXPS	TXDX12 信号极性选择位	0: 不将 TXDX12 信号极性进行反相输出 1: 将 TXDX12 信号极性进行反相输出	R/W
b1	RDXPS	RDX12 信号极性选择位	0: 不将 RDX12 极性进行反相输入 1: 将 RDX12 极性进行反相输入	R/W
b3-b2	—	保留位	读写值都为“0”。	R/W
b4	SHARPS	TXDX12/RDX12 引脚兼用选择位	0: TXDX12 引脚和 RDX12 引脚独立 1: TXDX12 引脚和 RDX12 引脚兼用	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W

SHARPS 位（TXDX12/RDX12 引脚兼用选择位）

当 SHARPS 位为“1”时，能进行 TXDX12 引脚和 RDX12 引脚兼用的半双工通信。

27.2.23 中断控制寄存器 (ICR)

地址 SCI12.ICR 0008 B326h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	AEDIE	BCDIE	PIBDIE	CF1MIE	CF0MIE	BFDIE
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	BFDIE	Break Field Low width 检测中断允许位	0: 禁止 Break Field Low width 检测中断 1: 允许 Break Field Low width 检测中断	R/W
b1	CF0MIE	Control Field 0 相同中断允许位	0: 禁止 Control Field 0 相同中断 1: 允许 Control Field 0 相同中断	R/W
b2	CF1MIE	Control Field 1 相同中断允许位	0: 禁止 Control Field 1 相同中断 1: 允许 Control Field 1 相同中断	R/W
b3	PIBDIE	优先级中断位的检测中断允许位	0: 禁止优先级中断位检测中断 1: 允许优先级中断位检测中断	R/W
b4	BCDIE	总线冲突检测中断允许位	0: 禁止总线冲突检测中断 1: 允许总线冲突检测中断	R/W
b5	AEDIE	有效边沿检测中断允许位	0: 禁止有效边沿检测中断 1: 允许有效边沿检测中断	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W

27.2.24 状态寄存器 (STR)

地址 SC12.STR 0008 B327h

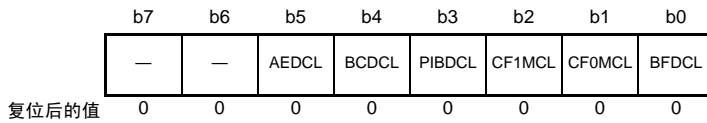
b7	b6	b5	b4	b3	b2	b1	b0
—	—	AEDF	BCDF	PIBDF	CF1MF	CF0MF	BFDF
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	BFDF	Break Field Low width 检测标志	[为“1”的条件] <ul style="list-style-type: none"> 当检测到 Break Field Low width 时 当 Break Field Low width 输出结束时 当定时器发生下溢时 [为“0”的条件] <ul style="list-style-type: none"> 当给 STCR.BFDCL 位写“1”时 	R
b1	CF0MF	Control Field 0 相同标志	[为“1”的条件] <ul style="list-style-type: none"> 当 Control Field 0 的接收数据和设定的数据相同时 [为“0”的条件] <ul style="list-style-type: none"> 当给 STCR.CF0MCL 位写“1”时 	R
b2	CF1MF	Control Field 1 相同标志	[为“1”的条件] <ul style="list-style-type: none"> 当 Control Field 1 的接收数据和设定的数据相同时 [为“0”的条件] <ul style="list-style-type: none"> 当给 STCR.CF1MCL 位写“1”时 	R
b3	PIBDF	优先级中断位检测标志	[为“1”的条件] <ul style="list-style-type: none"> 当检测到优先级中断位时 [为“0”的条件] <ul style="list-style-type: none"> 当给 STCR.PIBDCL 位写“1”时 	R
b4	BCDF	总线冲突检测标志	[为“1”的条件] <ul style="list-style-type: none"> 当检测到总线冲突时 [为“0”的条件] <ul style="list-style-type: none"> 当给 STCR.BCDCL 位写“1”时 	R
b5	AEDF	有效边沿检测标志	[为“1”的条件] <ul style="list-style-type: none"> 当检测到有效边沿时 [为“0”的条件] <ul style="list-style-type: none"> 当给 STCR.AEDCL 位写“1”时 	R
b7-b6	—	保留位	读取值为“0”，写操作无效。	R

27.2.25 状态清除寄存器 (STCR)

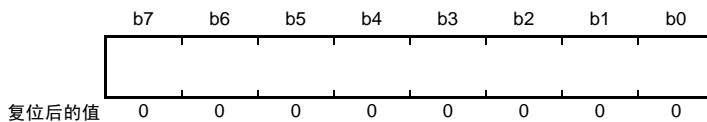
地址 SC112.STCR 0008 B328h



位	符号	位名	功能	R/W
b0	BFDFCL	BFDF 清除位	如果将 BFDFCL 位置“1”，就清除 STR.BFDF 标志。读取值为“0”。	R/W
b1	CF0MCL	CF0MF 清除位	如果将 CF0MCL 位置“1”，就清除 STR.CF0MF 标志。读取值为“0”。	R/W
b2	CF1MCL	CF1MF 清除位	如果将 CF1MCL 位置“1”，就清除 STR.CF1MF 标志。读取值为“0”。	R/W
b3	PIBDCL	PIBDF 清除位	如果将 PIBDFCL 位置“1”，就清除 STR.PIBDF 标志。读取值为“0”。	R/W
b4	BCDCL	BCDF 清除位	如果将 BCDCL 位置“1”，就清除 STR.BCDF 标志。读取值为“0”。	R/W
b5	AEDCL	AEDF 清除位	如果将 AEDCL 位置“1”，就清除 STR.AEDF 标志。读取值为“0”。	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W

27.2.26 Control Field 0 数据寄存器 (CF0DR)

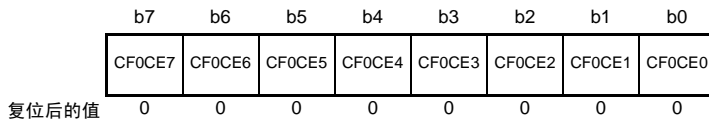
地址 SC112.CF0DR 0008 B329h



CF0DR 寄存器是保存 Control Field 0 比较数据的 8 位可读写寄存器。

27.2.27 Control Field 0 比较允许寄存器（CF0CR）

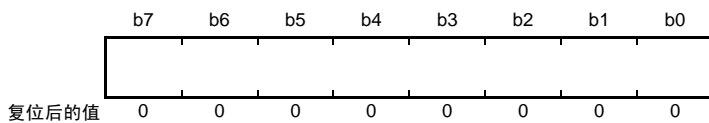
地址 SCI12.CF0CR 0008 B32Ah



位	符号	位名	功能	R/W
b0	CF0CE0	Control Field 0 bit0 的比较允许位	0: Control Field 0 bit0 的比较无效 1: Control Field 0 bit0 的比较有效	R/W
b1	CF0CE1	Control Field 0 bit1 的比较允许位	0: Control Field 0 bit1 的比较无效 1: Control Field 0 bit1 的比较有效	R/W
b2	CF0CE2	Control Field 0 bit2 的比较允许位	0: Control Field 0 bit2 的比较无效 1: Control Field 0 bit2 的比较有效	R/W
b3	CF0CE3	Control Field 0 bit3 的比较允许位	0: Control Field 0 bit3 的比较无效 1: Control Field 0 bit3 的比较有效	R/W
b4	CF0CE4	Control Field 0 bit4 的比较允许位	0: Control Field 0 bit4 的比较无效 1: Control Field 0 bit4 的比较有效	R/W
b5	CF0CE5	Control Field 0 bit5 的比较允许位	0: Control Field 0 bit5 的比较无效 1: Control Field 0 bit5 的比较有效	R/W
b6	CF0CE6	Control Field 0 bit6 的比较允许位	0: Control Field 0 bit6 的比较无效 1: Control Field 0 bit6 的比较有效	R/W
b7	CF0CE7	Control Field 0 bit7 的比较允许位	0: Control Field 0 bit7 的比较无效 1: Control Field 0 bit7 的比较有效	R/W

27.2.28 Control Field 0 接收数据寄存器（CF0RR）

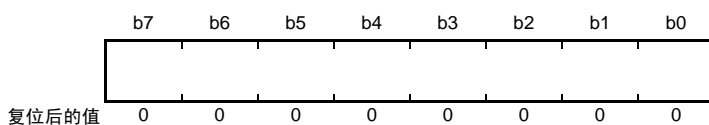
地址 SCI12.CF0RR 0008 B32Bh



CF0RR 寄存器是保存 Control Field 0 接收数据的 8 位可读寄存器。CPU 和 DTC 不能写 CF0RR 寄存器。

27.2.29 主 Control Field 1 数据寄存器（PCF1DR）

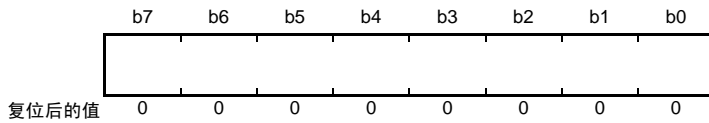
地址 SCI12.PCF1DR 0008 B32Ch



PCF1DR 寄存器是保存 Control Field 1 主比较数据的 8 位可读写寄存器。

27.2.30 次 Control Field 1 数据寄存器 (SCF1DR)

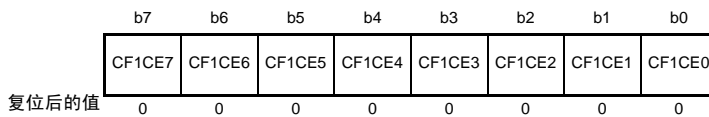
地址 SCI12.SCF1DR 0008 B32Dh



SCF1DR 寄存器是保存 Control Field 1 次比较数据的 8 位可读写寄存器。

27.2.31 Control Field 1 比较允许寄存器 (CF1CR)

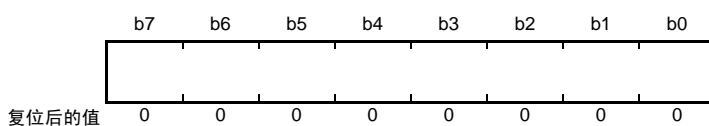
地址 SCI12.CF1CR 0008 B32Eh



位	符号	位名	功能	R/W
b0	CF1CE0	Control Field 1 bit0 的比较允许位	0: Control Field 1 bit0 的比较无效 1: Control Field 1 bit0 的比较有效	R/W
b1	CF1CE1	Control Field 1 bit1 的比较允许位	0: Control Field 1 bit1 的比较无效 1: Control Field 1 bit1 的比较有效	R/W
b2	CF1CE2	Control Field 1 bit2 的比较允许位	0: Control Field 1 bit2 的比较无效 1: Control Field 1 bit2 的比较有效	R/W
b3	CF1CE3	Control Field 1 bit3 的比较允许位	0: Control Field 1 bit3 的比较无效 1: Control Field 1 bit3 的比较有效	R/W
b4	CF1CE4	Control Field 1 bit4 的比较允许位	0: Control Field 1 bit4 的比较无效 1: Control Field 1 bit4 的比较有效	R/W
b5	CF1CE5	Control Field 1 bit5 的比较允许位	0: Control Field 1 bit5 的比较无效 1: Control Field 1 bit5 的比较有效	R/W
b6	CF1CE6	Control Field 1 bit6 的比较允许位	0: Control Field 1 bit6 的比较无效 1: Control Field 1 bit6 的比较有效	R/W
b7	CF1CE7	Control Field 1 bit7 的比较允许位	0: Control Field 1 bit7 的比较无效 1: Control Field 1 bit7 的比较有效	R/W

27.2.32 Control Field 1 接收数据寄存器 (CF1RR)

地址 SCI12.CF1RR 0008 B32Fh



CF1RR 寄存器是保存 Control Field 1 接收数据的 8 位可读寄存器。CPU 和 DTC 不能写 CF1RR 寄存器。

27.2.33 定时器的控制寄存器（TCR）

地址 SCI12.TCR 0008 B330h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	TCST
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	TCST	定时器的计数开始位	0: 定时器停止计数 1: 定时器开始计数	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

27.2.34 定时器的模式寄存器（TMR）

地址 SCI12.TMR 0008 B331h

b7	b6	b5	b4	b3	b2	b1	b0
—	TCSS[2:0]		TWRC	—	TOMS[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	TOMS[1:0]	定时器运行模式选择位（注1）	b1 b0 0 0: 定时器模式 0 1: Break Field Low width 判定模式 1 0: Break Field Low width 输出模式 1 1: 不能设定	R/W
b2	—	保留位	读写值都为“0”。	R/W
b3	TWRC	计数器的写控制位	0: 写重加载寄存器和计数器 1: 只写重加载寄存器	R/W
b6-b4	TCSS[2:0]	定时器的计数时钟源选择位（注1）	b6 b4 0 0 0: PCLK 0 0 1: PCLK/2 0 1 0: PCLK/4 0 1 1: PCLK/8 1 0 0: PCLK/16 1 0 1: PCLK/32 1 1 0: PCLK/64 1 1 1: PCLK/128	R/W
b7	—	保留位	读写值都为“0”。	R/W

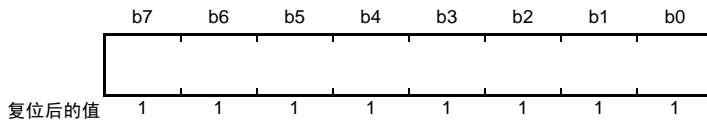
注1. 必须在定时器停止计数时（TCST=0）改写 TOMS[1:0] 位和 TCSS[2:0] 位。

TWRC 位（计数器的写控制位）

在写 TCNT 寄存器和 TPRES 寄存器时，此位选择是只写重加载寄存器还是写重加载寄存器和计数器。

27.2.35 定时器的预分频寄存器 (TPRE)

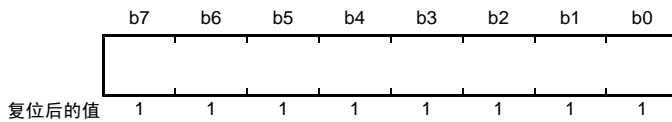
地址 SCI12.TPRE 0008 B332h



TPRE 寄存器由 8 位重加载寄存器、读缓冲器和计数器构成，初始值都为“FFh”。计数器通过 TMR.TCSS[2:0] 位选择的计数时钟源进行递减计数。如果发生下溢，就将重加载寄存器的值装入到计数器，而且下溢为 TCNT 计数器的计数时钟源。重加载寄存器和读缓冲器被分配在同一地址，在写此寄存器时，将值写到重加载寄存器；在读此寄存器时，读取值为传送到读缓冲器的计数值。在将重加载寄存器的值装入到计数器时，需要 1 个系统运行时钟。

27.2.36 定时器的计数器 (TCNT)

地址 SCI12.TCNT 0008 B333h



TCNT 计数器由 8 位重加载寄存器、读缓冲器和计数器构成，初始值都为“FFh”。对 TPRE 寄存器的下溢进行递减计数，如果 TCNT 计数器发生下溢，就将重加载寄存器的值装入到计数器。重加载寄存器和读缓冲器被分配在同一地址，在写此计数器时，将值写到重加载寄存器；在读此计数器时，读取值为传送到读缓冲器的计数值。在将重加载寄存器的值装入到计数器时，需要 1 个系统运行时钟。

27.3 异步模式的运行

异步串行通信的一般数据格式如图 27.3 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, 就视为起始位并且开始串行通信。

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。

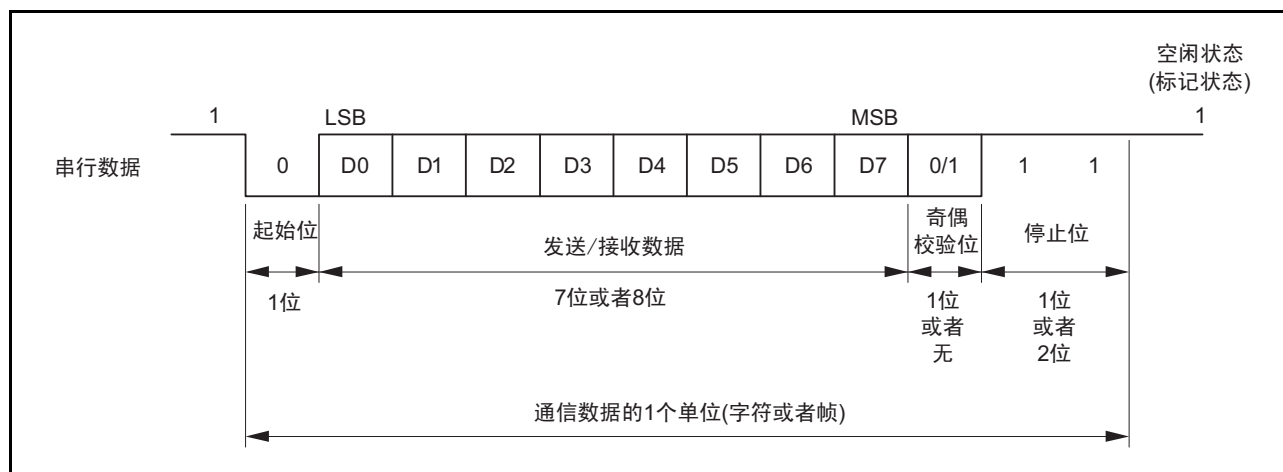


图 27.3 异步串行通信的数据格式 (8 位数据 / 有奇偶校验 / 2 个停止位的例子)

27.3.1 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 27.22 所示。

串行发送 / 接收格式有 12 种，能通过设定 SMR 寄存器进行选择。有关多处理器通信功能的详细内容，请参照“27.4 多处理器通信功能”。

表 27.22 串行发送 / 接收格式 (异步模式)

SMR 的设定				串行发送/接收格式和帧长												
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12	
0	0	0	0	S	8位数据								STOP			
0	0	0	1	S	8位数据								STOP	STOP		
0	1	0	0	S	8位数据								P	STOP		
0	1	0	1	S	8位数据								P	STOP	STOP	
1	0	0	0	S	7位数据							STOP				
1	0	0	1	S	7位数据							STOP	STOP			
1	1	0	0	S	7位数据							P	STOP			
1	1	0	1	S	7位数据							P	STOP	STOP		
0	—	1	0	S	8位数据								MPB	STOP		
0	—	1	1	S	8位数据								MPB	STOP	STOP	
1	—	1	0	S	7位数据							MPB	STOP			
1	—	1	1	S	7位数据							MPB	STOP	STOP		

- S : 起始位
- STOP : 停止位
- P : 奇偶校验位
- MPB : 多处理器位

27.3.2 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 27.4 所示，通过在基本时钟的第 8 个周期（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。

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100 \quad [\%] \quad \dots \text{表达式(1)}$$

- 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

$$M = \{ 0.5 - 1/(2 \times 16) \} \times 100 \quad [\%] = 46.875\%$$

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。

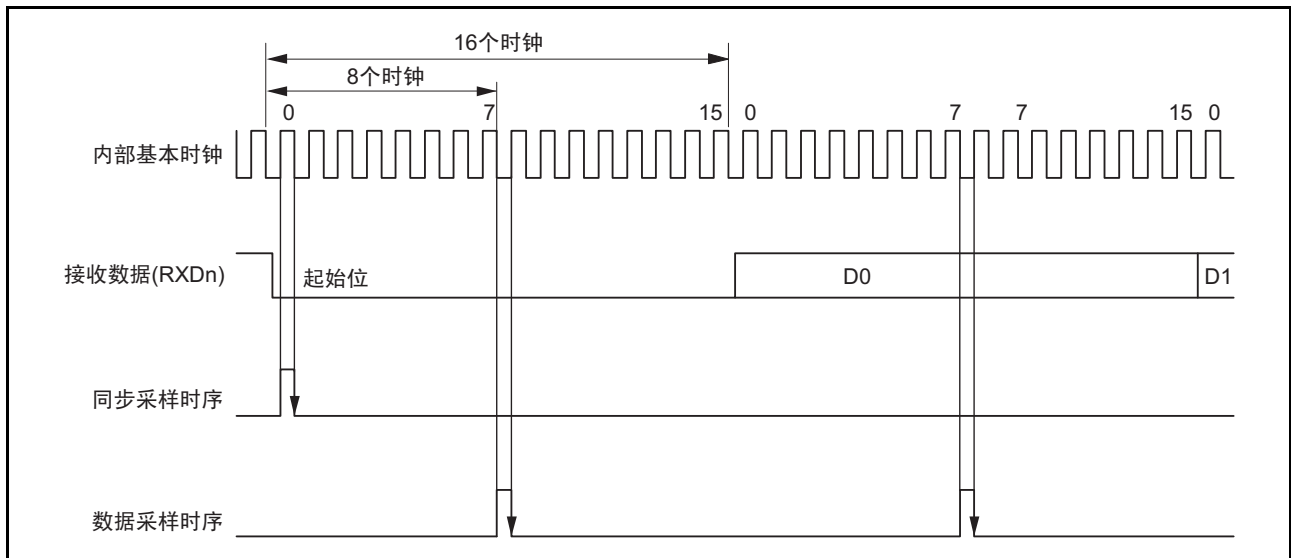


图 27.4 异步模式的接收数据采样时序

27.3.3 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。

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 27.5 所示，时钟在发送数据的中央上升。

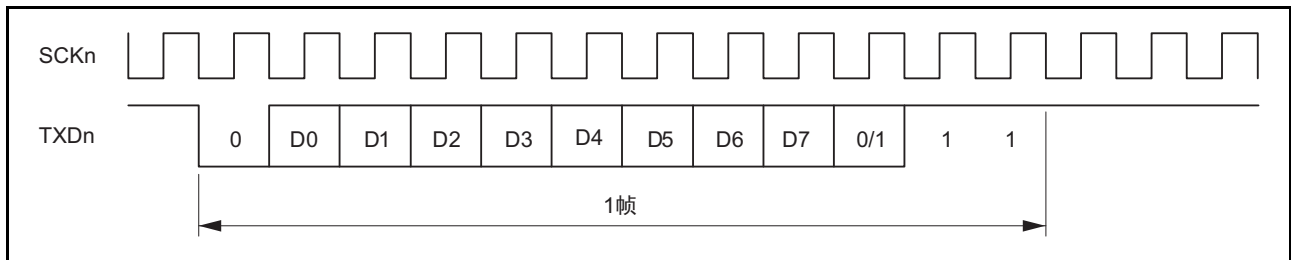


图 27.5 输出时钟和发送数据的相位关系
(异步模式: `SMR.CHR=0`、`SMR.PE=1`、`SMR.MP=0`、`SMR.STOP=1`)

27.3.4 CTS 和 RTS 功能

`CTS` 功能是使用 `CTSn#` 引脚的输入进行发送控制的功能。

如果将 `SPMR.CTSE` 位置“1”，`CTS` 功能就有效。在 `CTS` 功能有效时并且只在 `CTSn#` 引脚的输入为 `Low` 电平时，开始发送。

如果在发送过程中将 `CTSn#` 引脚置为 `Low` 电平，正在发送的帧就不受影响而继续发送。

`RTS` 功能是使用 `RTSn#` 引脚的输出进行接收请求的功能。如果处于可接收状态，就输出 `Low` 电平。输出 `Low` 电平或者 `High` 电平的条件如下：

[为 `Low` 电平的条件]

当满足以下全部条件时

- `SCR.RE` 位为“1”
- 不处于接收状态。
- 在读取前没有接收数据。
- `SSR.ORER`、`SSR.FER`、`SSR.PER` 标志全部为“0”。

[为 `High` 电平的条件]

当不满足 `Low` 电平的条件时

27.3.5 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 27.6 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”置为“0”或者从“0”置为“1”，就在 SCR.TIE 位为“1”时产生 TXI 中断请求。

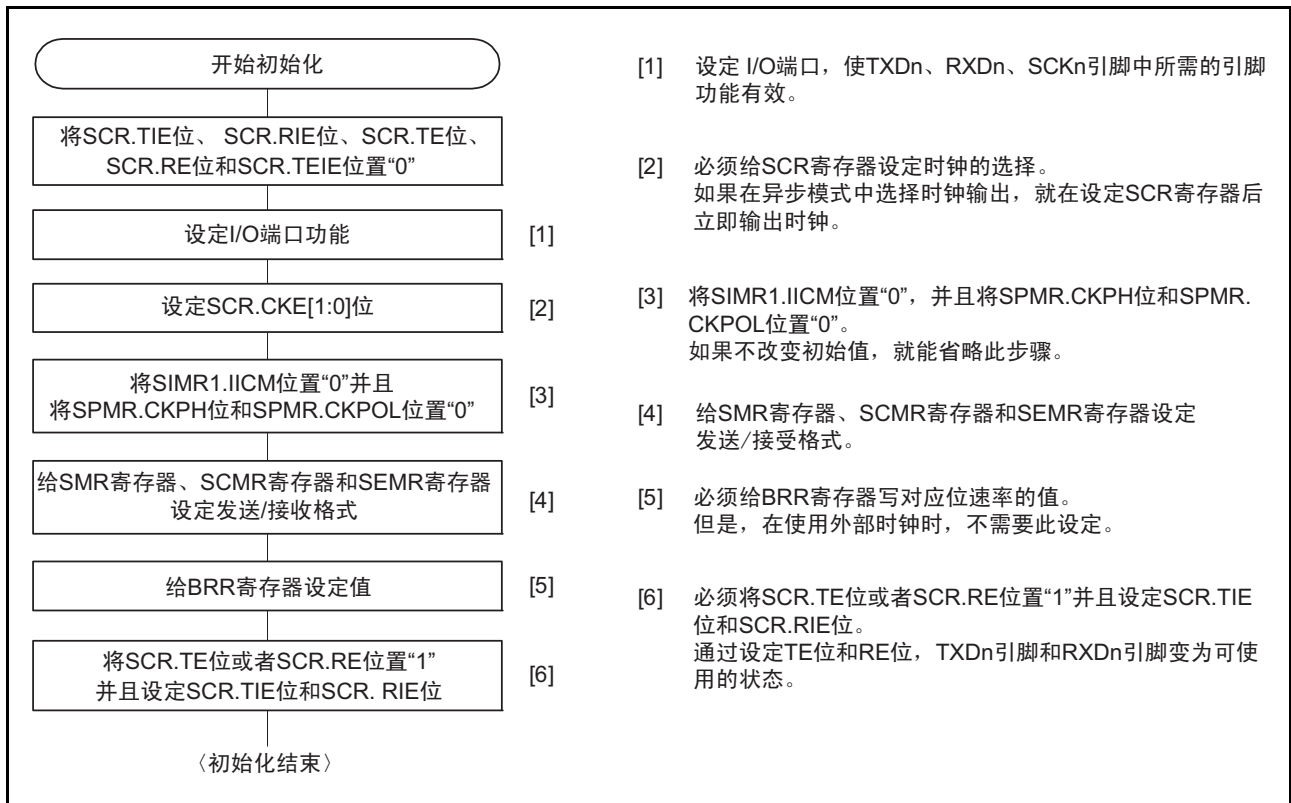


图 27.6 SCI 的初始化流程图例子 (异步模式)

27.3.6 串行数据的发送 (异步模式)

异步模式的串行发送例子如图 27.7 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，通过在将SCR.TIE位置“1”后将SCR.TE位置“1”，或者用1条指令同时将这2个位置“1”，产生TXI中断请求。
2. 在SPMR.CTSE位为“0”（禁止CTS功能）或者CTS_n#引脚的输入为Low电平的状态下，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。当使用TEI中断请求时，在TXI中断请求处理程序中给TDR寄存器写最后的发送数据，然后将SCR.TIE位置“0”（禁止TXI中断请求）并且将SCR.TEIE位置“1”（允许TEI中断请求）。
3. 从TXD_n引脚依次发送起始位、发送数据、奇偶校验位或者多处理器位（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，就在SPMR.CTSE位为“0”（禁止CTS功能）或者CTS_n#引脚的输入为Low电平的状态下，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。此时，如果SCR.TEIE位为“1”，就将SSR.TEND标志置“1”并且产生TEI中断请求。

串行发送的流程图例子如图 27.8 所示。

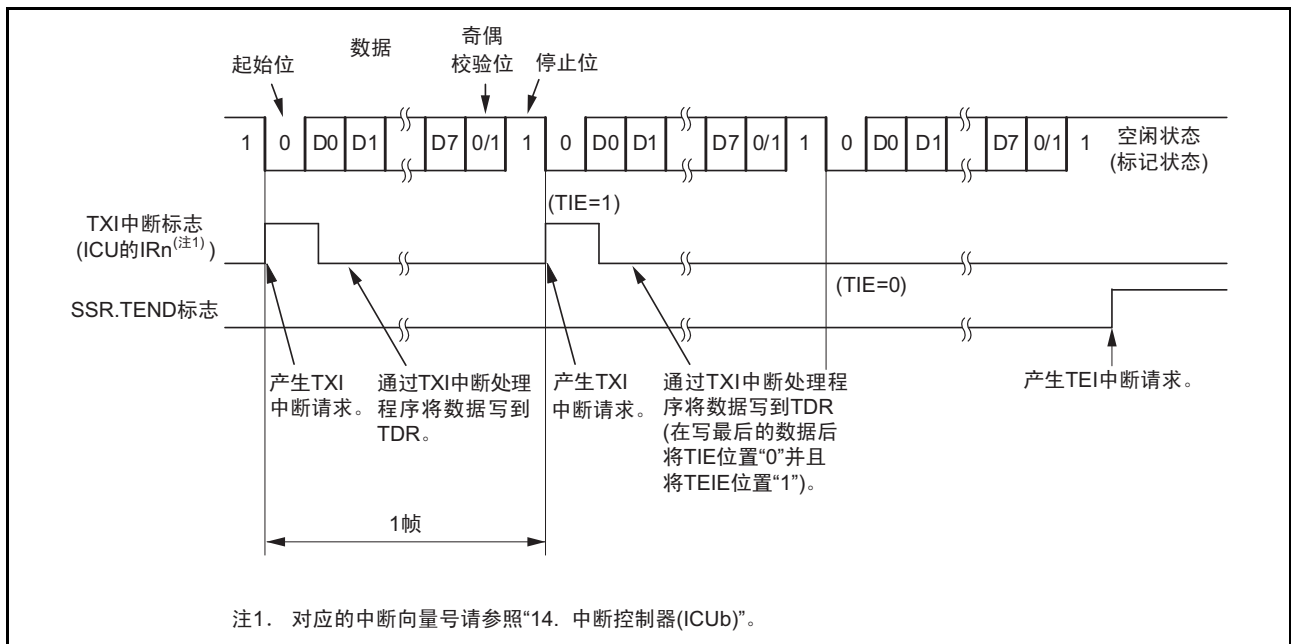


图 27.7 异步模式的串行发送（正在发送~发送结束的情况）例子
（8 位数据 / 有奇偶校验 / 1 个停止位的例子）

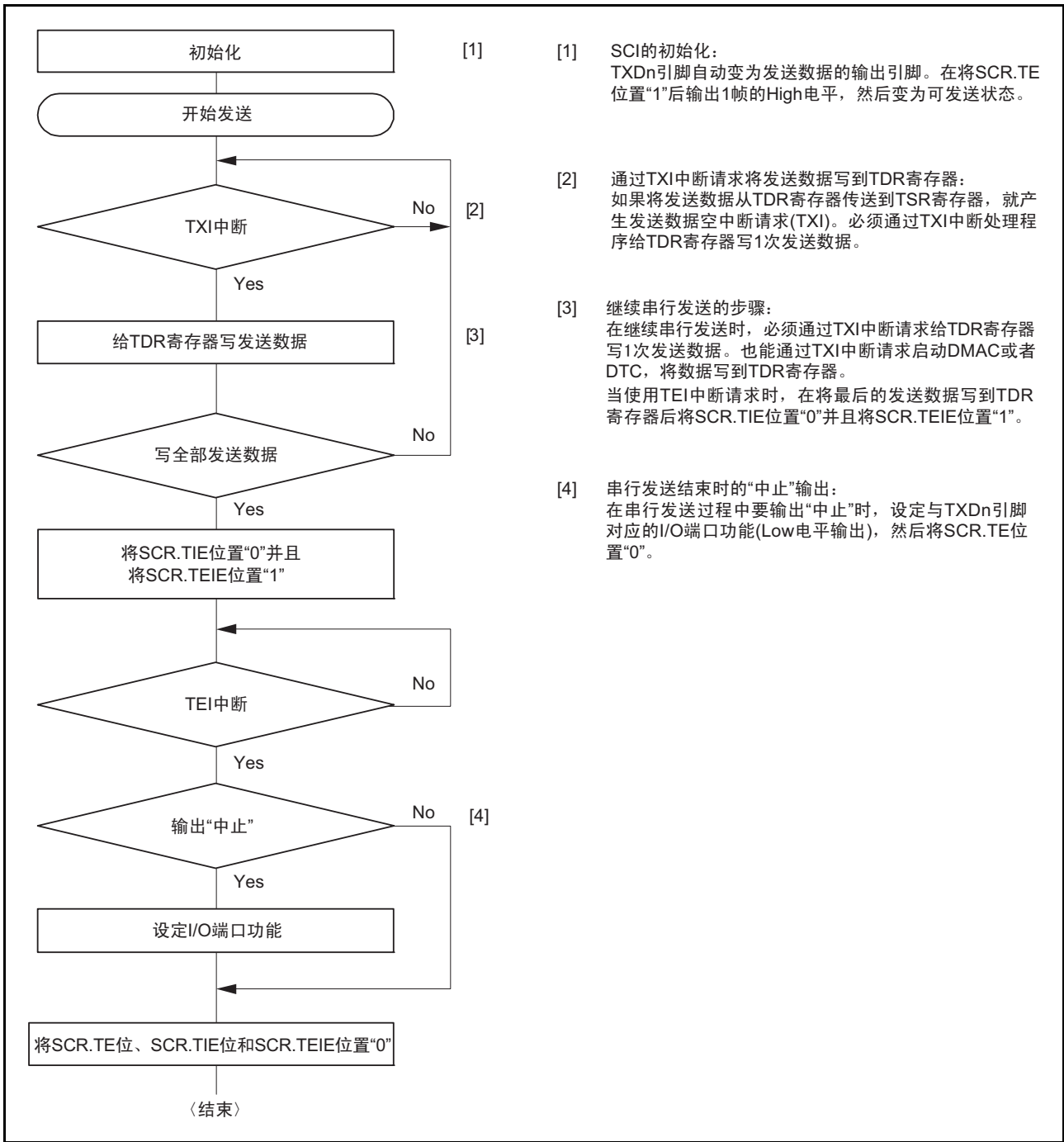


图 27.8 异步模式的串行发送的流程图例子

27.3.7 串行数据的接收 (异步模式)

异步模式的串行接收例子如图 27.9 和图 27.10 所示。
在接收串行数据时，SCI 的运行如下：

1. 如果将SCR.RE位置“1”，就将RTSn#引脚的输出置为Low电平。
2. 监视通信线路，如果检测到起始位，就与内部取得同步将接收数据取到RDR寄存器，并且检查奇偶校验位和停止位。
3. 如果发生溢出错误，就将SSR.ORER标志置位。此时，如果SCR.RIE位为“1”，就产生ERI中断请求。接收数据不传送到RDR寄存器。
4. 如果检测到奇偶校验错误，就将SSR.PER标志置位，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. 如果检测到帧错误（停止位为“0”时），就将SSR.FER标志置位，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。
6. 当正常接收时，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，进行连续的接收。如果读被传送到RDR寄存器的接收数据，就将RTSn#引脚的输出置为Low电平。

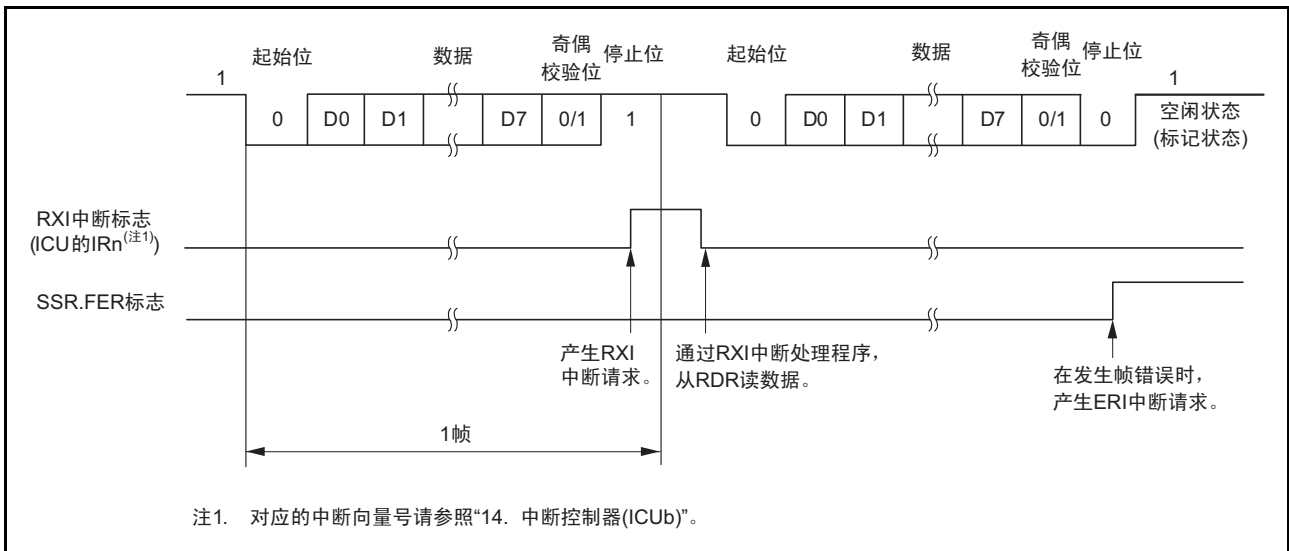


图 27.9 异步模式的串行接收例子 (1) (不使用 RTS 功能的情况)
(8 位数据 / 有奇偶校验 / 1 个停止位的例子)

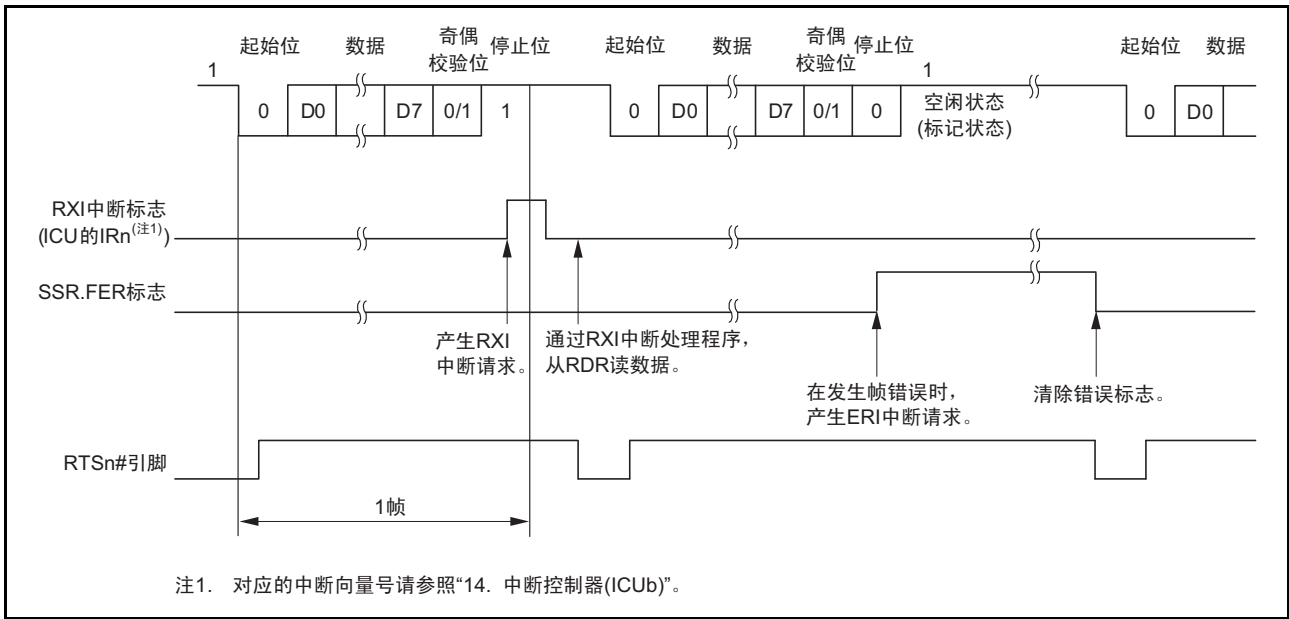


图 27.10 异步模式的串行接收例子 (2) (使用 RTS 功能的情况)
(8 位数据 / 有奇偶校验 / 1 个停止位的例子)

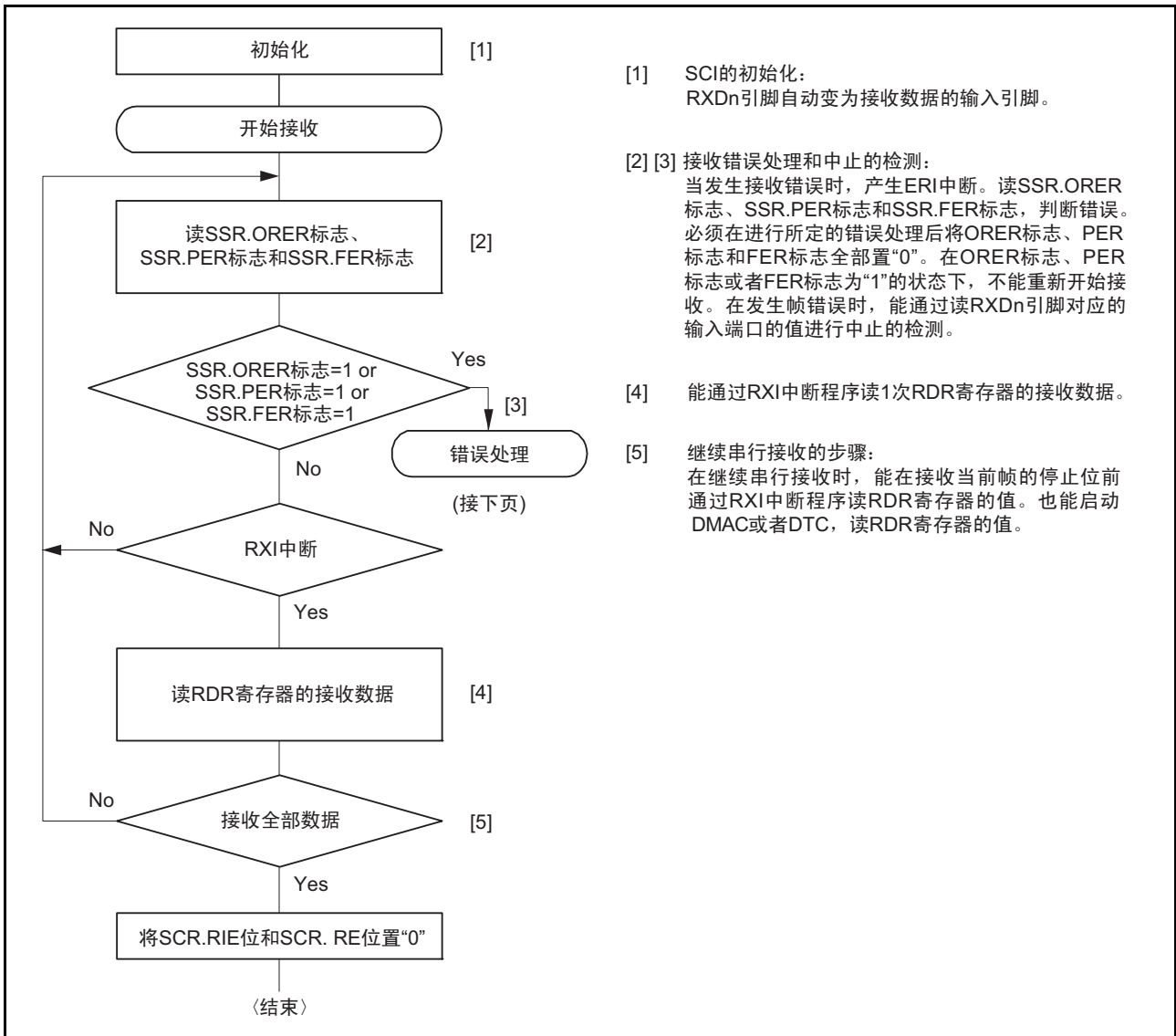
检测到接收错误时的 SSR 寄存器各状态标志的状态和接收数据的处理如表 27.23 所示。

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。在接收错误标志被置位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 27.11 和图 27.12 所示。

表 27.23 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



- [1] SCI的初始化:
RXDn引脚自动变为接收数据的输入引脚。
- [2] [3] 接收错误处理和中止的检测:
当发生接收错误时, 产生ERI中断。读SSR. ORER标志、SSR. PER标志和SSR. FER标志, 判断错误。必须在进行所定的错误处理后将ORER标志、PER标志和FER标志全部置“0”。在ORER标志、PER标志或者FER标志为“1”的状态下, 不能重新开始接收。在发生帧错误时, 能通过读RXDn引脚对应的输入端口的值进行中止的检测。
- [4] 能通过RXI中断程序读1次RDR寄存器的接收数据。
- [5] 继续串行接收的步骤:
在继续串行接收时, 能在接收当前帧的停止位前通过RXI中断程序读RDR寄存器的值。也能启动DMAC或者DTC, 读RDR寄存器的值。

图 27.11 异步模式的串行接收的流程图例子 (1)

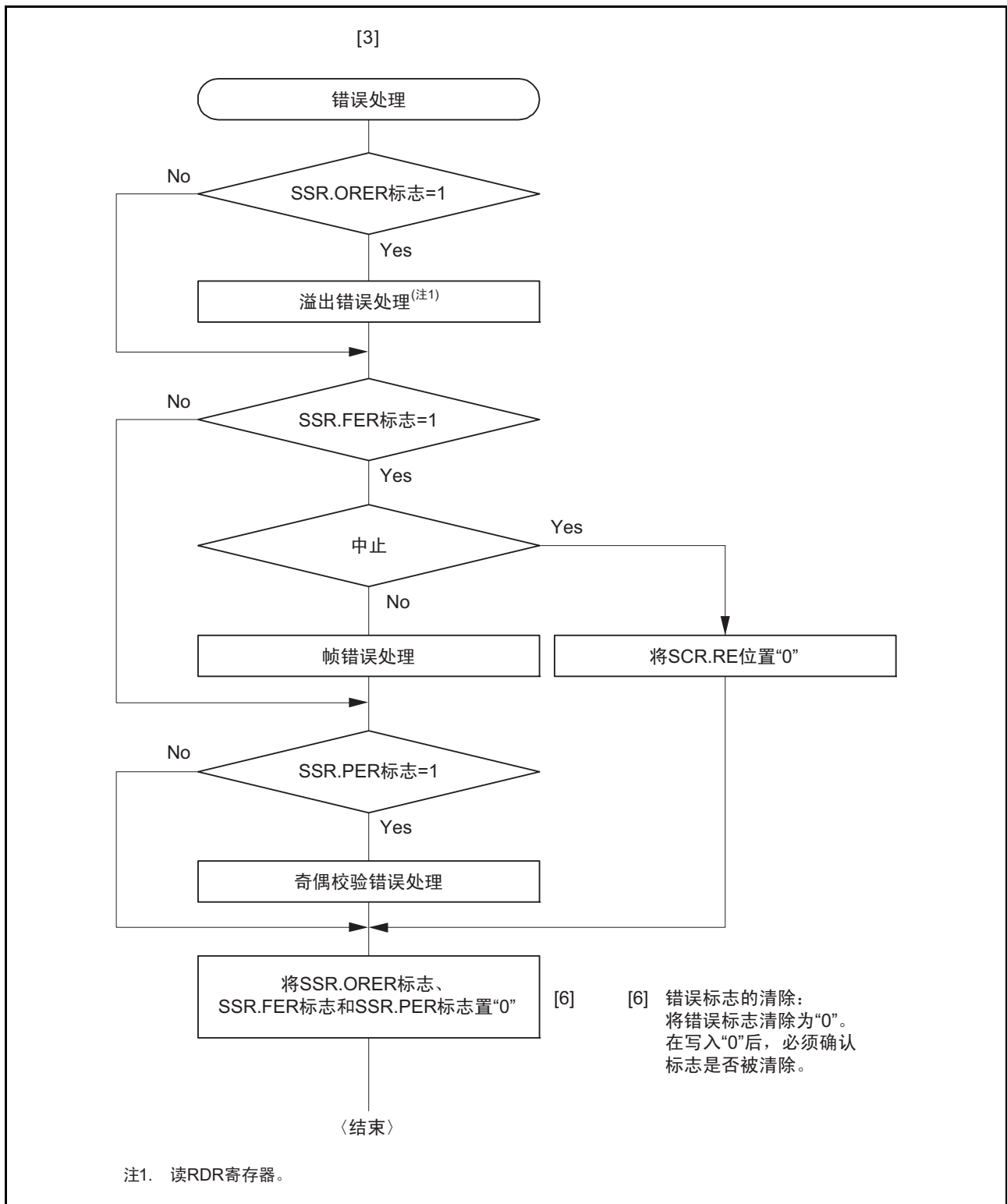


图 27.12 异步模式的串行接收的流程图例子 (2)

27.4 多处理器通信功能

如果使用多处理器通信功能，就能通过附加多处理器位的异步串行通信，在多个处理器之间共享通信线路进行数据的发送和接收。在多处理器通信中，给接收站分配各自特有的 ID 码。串行通信周期由指定接收站的 ID 发送周期和指定接收站的数据发送周期构成，用多处理器位区分 ID 发送周期和数据发送周期。当多处理器位是“1”时，为 ID 发送周期；当多处理器位是“0”时，为数据发送周期。使用多处理器格式的处理器之间的通信例子如图 27.13 所示。发送站首先发送多处理器位为“1”的接收站 ID 码，接着发送多处理器位为“0”的发送数据。如果接收站接收到多处理器位为“1”的通信数据，就将接收数据和本站的 ID 比较，如果相同，就继续接收被发送的通信数据；如果不相同，就在再次接收到下一个多处理器位为“1”的通信数据之前跳读通信数据。

SCI 为了支持此功能，设有 SCR.MPIE 位。如果将 MPIE 置“1”，就在接收到多处理器位为“1”的数据之前，禁止将接收数据从 RSR 寄存器传送到 RDR 寄存器，禁止检测接收错误并且禁止将 SRR.ORER 和 SRR.FER 各状态标志置位。如果接收到多处理器位为“1”的字符，就在 SSR.MPBT 位被置“1”的同时自动清除 SCR.MPIE，然后返回到通常的接收运行状态。此时，如果 SCR.RIE 位为“1”，就产生 RXI 中断。

在指定多处理器格式时，奇偶校验位的指定无效。除此以外与通常的异步模式相同，多处理器通信时的时钟也和通常的异步模式相同。

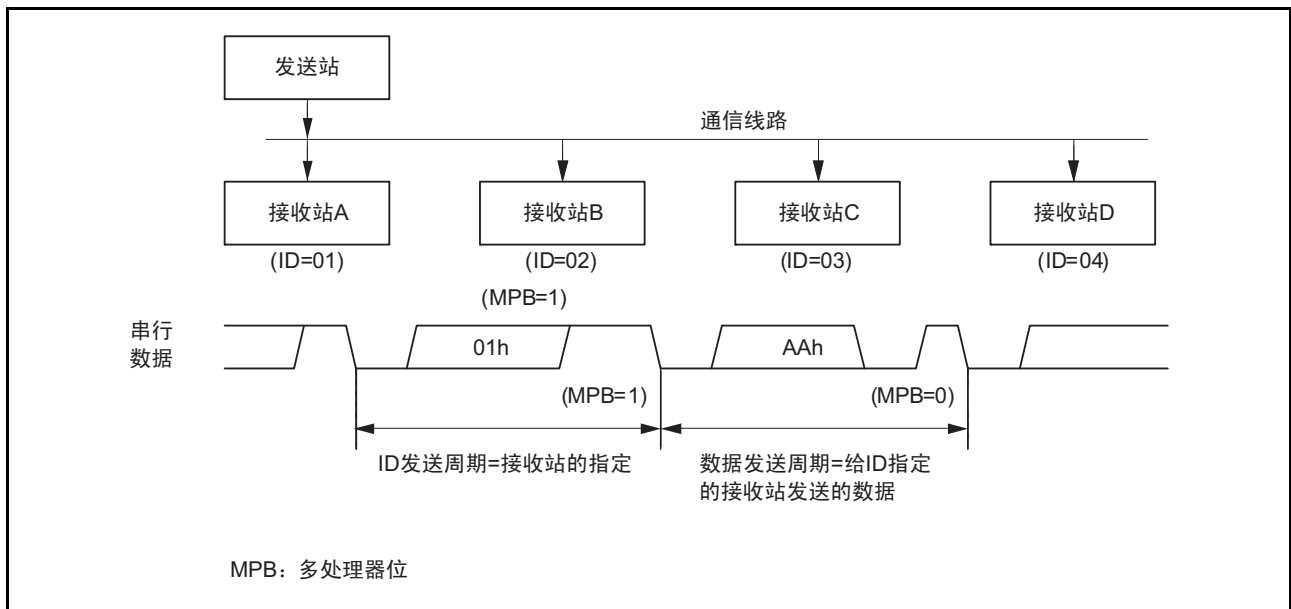


图 27.13 使用多处理器格式的通信的例子（将数据“AAh”发送到接收站 A 的例子）

27.4.1 多处理器串行数据的发送

多处理器数据处理的流程图例子如图 27.14 所示。必须在 ID 发送周期中将 SSR.MPBT 位置“1”后发送 ID 码，在数据发送周期中将 SSR.MPBT 位置“0”后发送数据。其他运行和异步模式的运行相同。

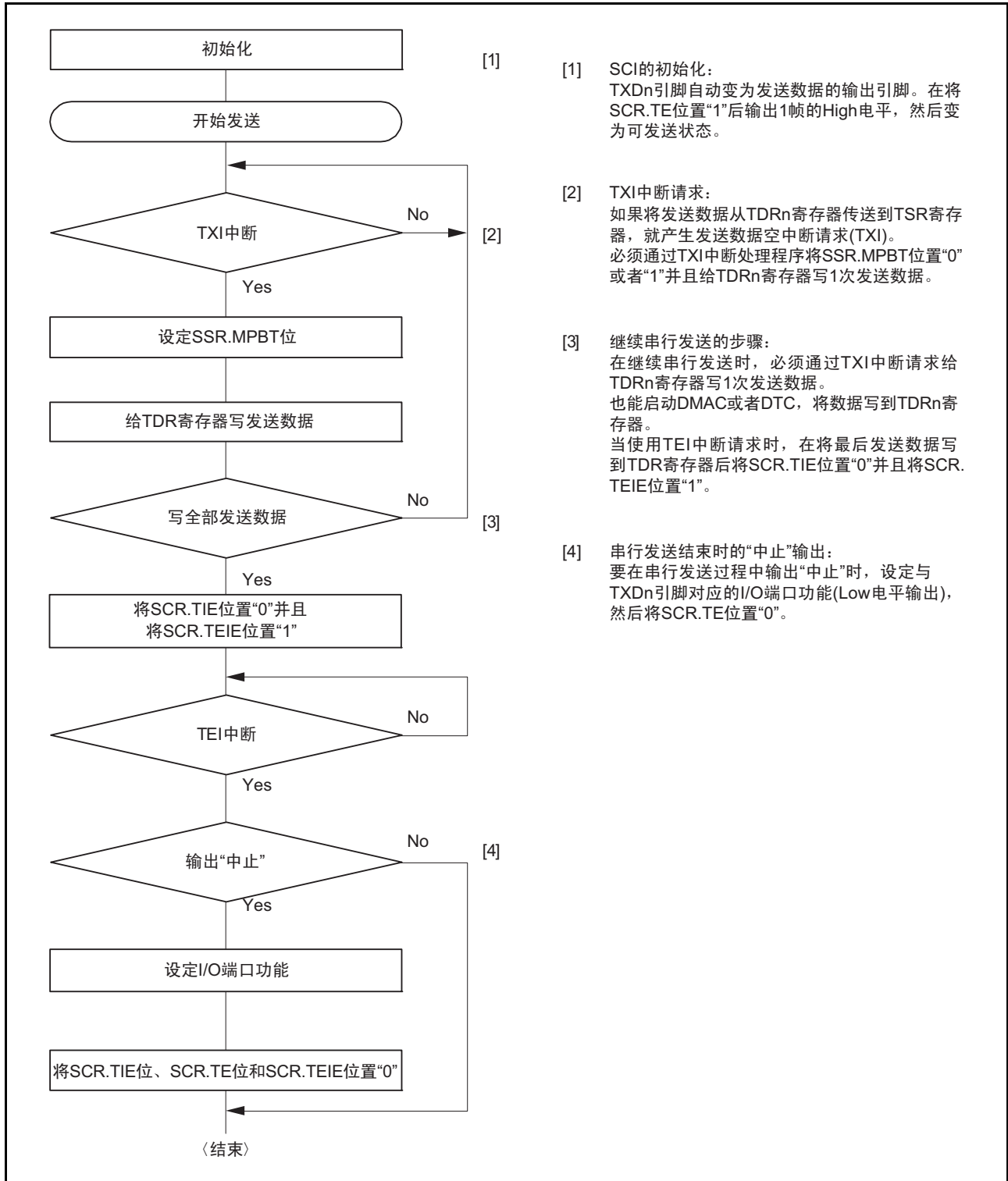


图 27.14 多处理器串行发送的流程图例子

27.4.2 多处理器串行数据的接收

多处理器数据接收的流程图例子如图 27.16 和图 27.17 所示。如果将 SCR.MPIE 位置“1”，就在接收到多处理器位为“1”的通信数据之前跳读通信数据。如果接收到多处理器位为“1”的通信数据，就将接收数据传送到 RDR 寄存器，此时产生 RXI 中断请求。其他运行和异步模式的运行相同。

接收时的运行例子如图 27.15 所示。

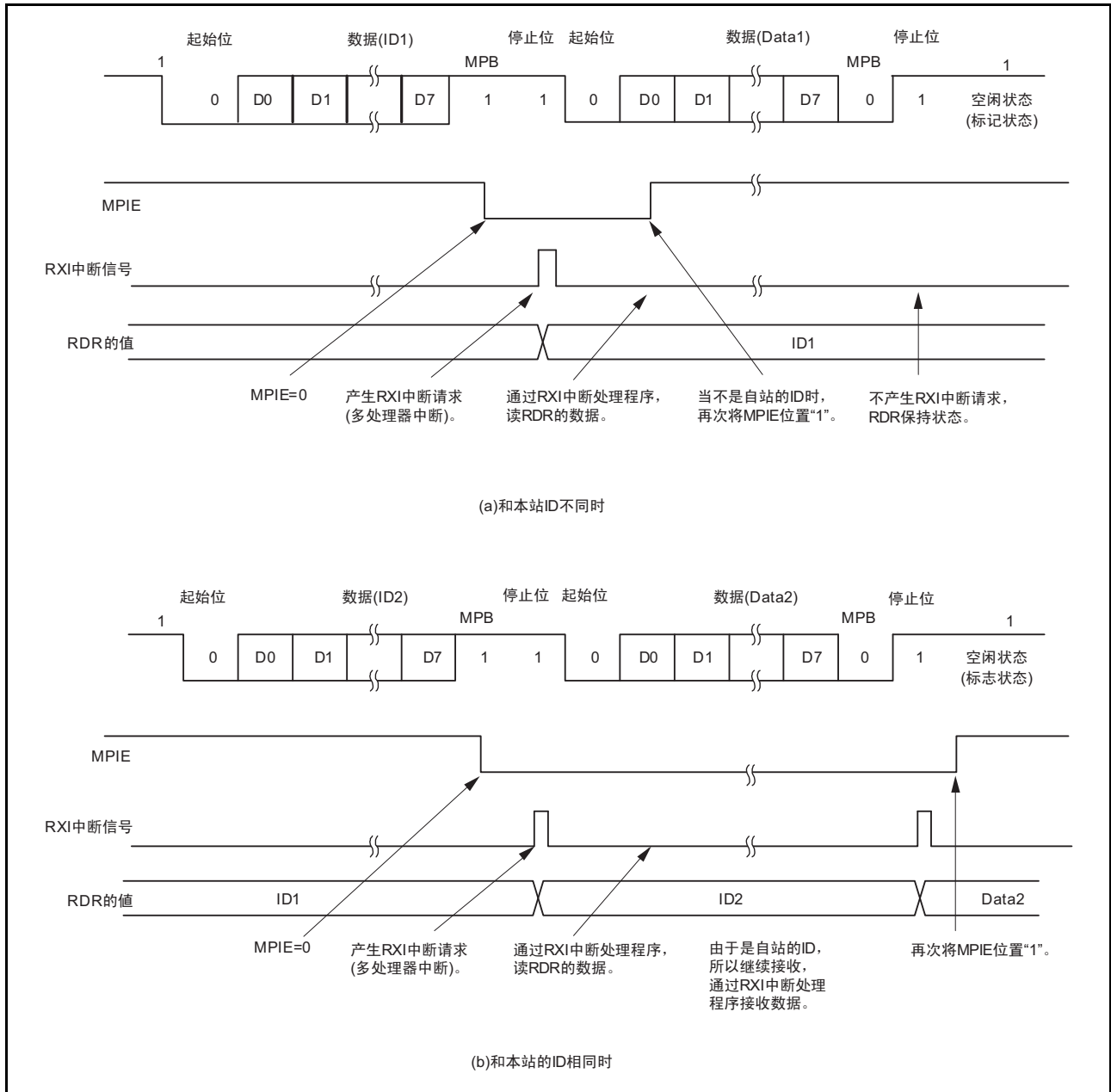


图 27.15 SCI 接收时的运行例子 (8 位数据 / 有多处理器位 / 1 个停止位的例子)

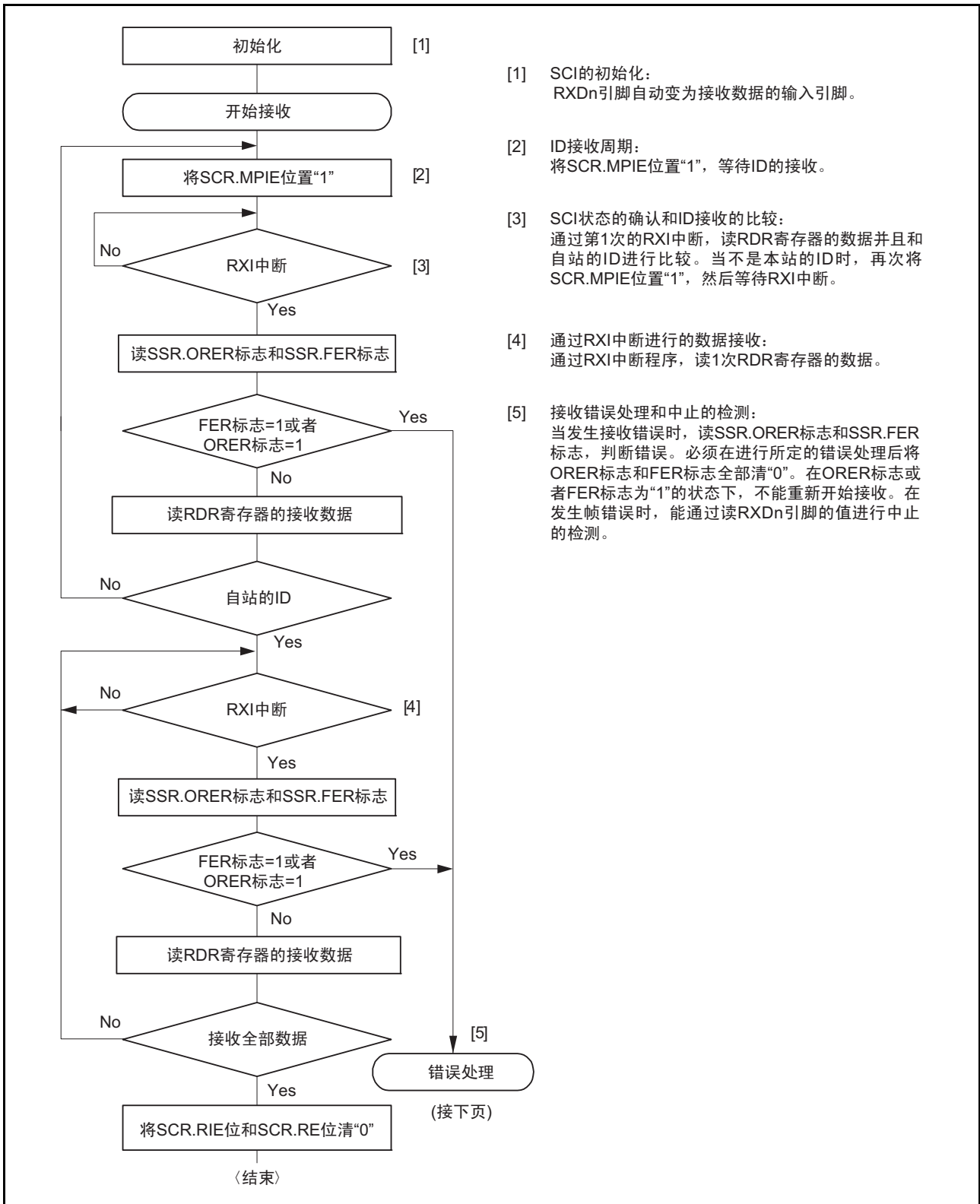


图 27.16 多处理器串行接收的流程图例子 (1)

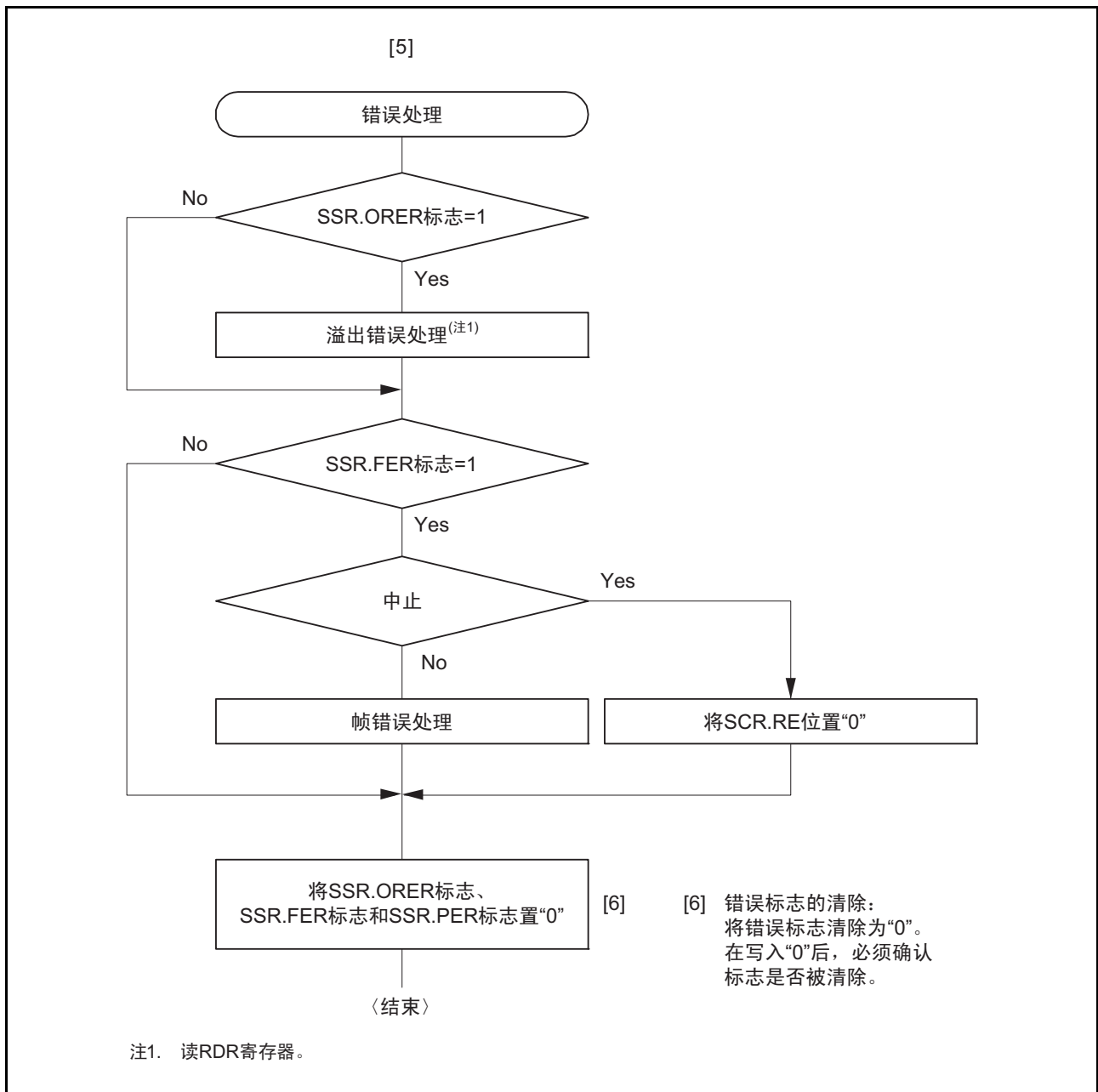


图 27.17 多处理器串行接收的流程图例子 (2)

27.5 时钟同步模式的运行

时钟同步串行通信的数据格式如图 27.18 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，进行连续的接收和发送。

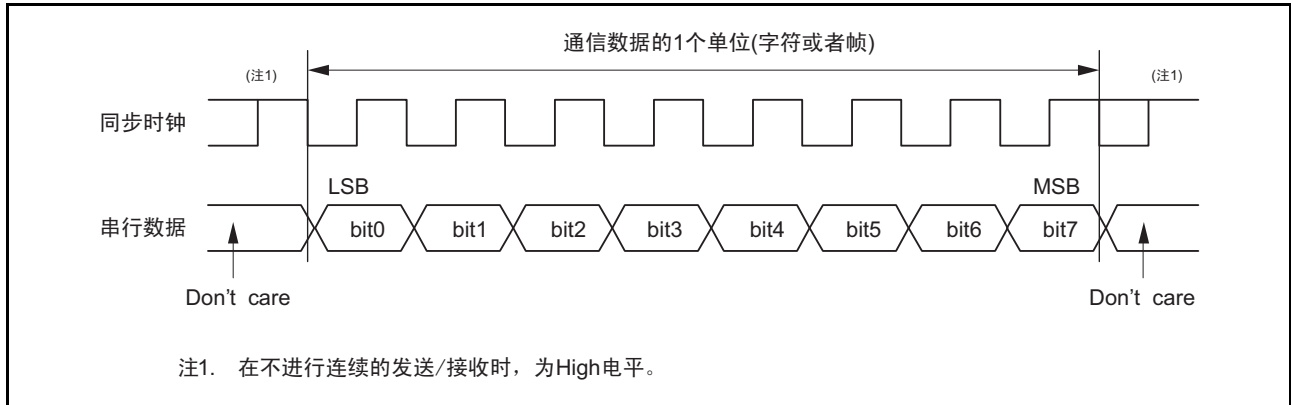


图 27.18 时钟同步串行通信的数据格式 (LSB first)

27.5.1 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，当只进行接收时，在 CTS 功能有效并且 CTSn# 引脚的输入为 High 电平或者发生溢出错误或者将 SCR.RE 位置“0”前输出同步时钟。在 CTS 功能有效的情况下，如果在结束帧的接收时 CTSn# 引脚的输入为 High 电平，就停止输出同步时钟。

27.5.2 CTS 和 RTS 功能

CTS 功能是在选择内部时钟的情况下使用 CTSn# 引脚的输入进行发送和接收开始控制的功能。如果将 SPMR.CTSE 位置“1”，CTS 功能就有效。

在 CTS 功能有效时并且只在 CTSn# 引脚的输入为 Low 电平时，开始发送和接收。

RTS 功能是在选择外部同步时钟的情况下使用 RTSn# 引脚的输出进行发送和接收开始请求的功能。如果能进行串行通信的状态，就输出 Low 电平。输出 Low 电平或者 High 电平的条件如下：

[为 Low 电平的条件]

当满足以下全部条件时

- SCR.RE 位或者 SCR.TE 位为“1”。
- 不处于发送和接收状态。
- 在读取前没有接收数据 (SCR.RE 位为“1”的情况)。
- 写完发送数据 (SCR.TE 位为“1”的情况)。
- SSR.ORER 标志为“0”。

[为 High 电平的条件]

当不满足 Low 电平的条件时

27.5.3 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 27.19 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”置为“0”或者从“0”置为“1”，就在 SCR.TIE 位为“1”时产生 TXI 中断。

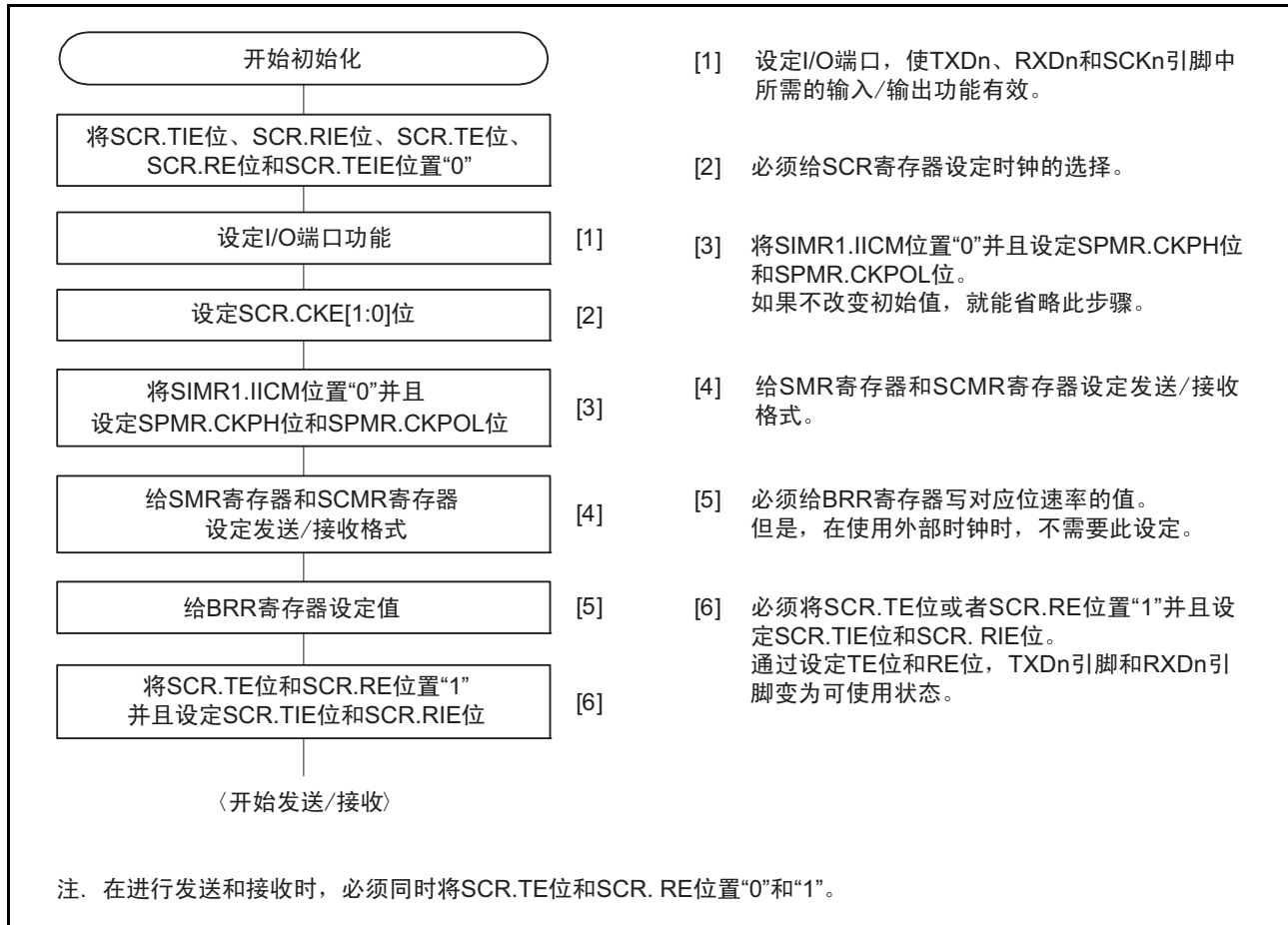


图 27.19 SCI 的初始化流程图例子 (时钟同步模式)

27.5.4 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 27.20 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，通过在将 SCR.TIE 位置“1”后将 SCR.TE 位置“1”，或者用 1 条指令同时将这 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。当使用 TEI 中断请求时，在 TXI 中断请求处理程序中给 TDR 寄存器写最后的发送数据，然后将 SCR.TIE 位置“0”（禁止 TXI 中断请求）并且将 SCR.TEIE 位置“1”（允许 TEI 中断请求）。
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。当 SPMR.CTSE 位为“1”（允许 CTS 功能）时，在等到 CTS 输入信号变为 Low 电平后开始输出时钟。
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，就将 SSR.TEND 标志置“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 27.21 所示。

不能在接收错误标志（SSR.ORER、SSR.FER、SSR.PER）被置“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：即使将 SCR.RE 位置“0”，也不清除接收错误标志。

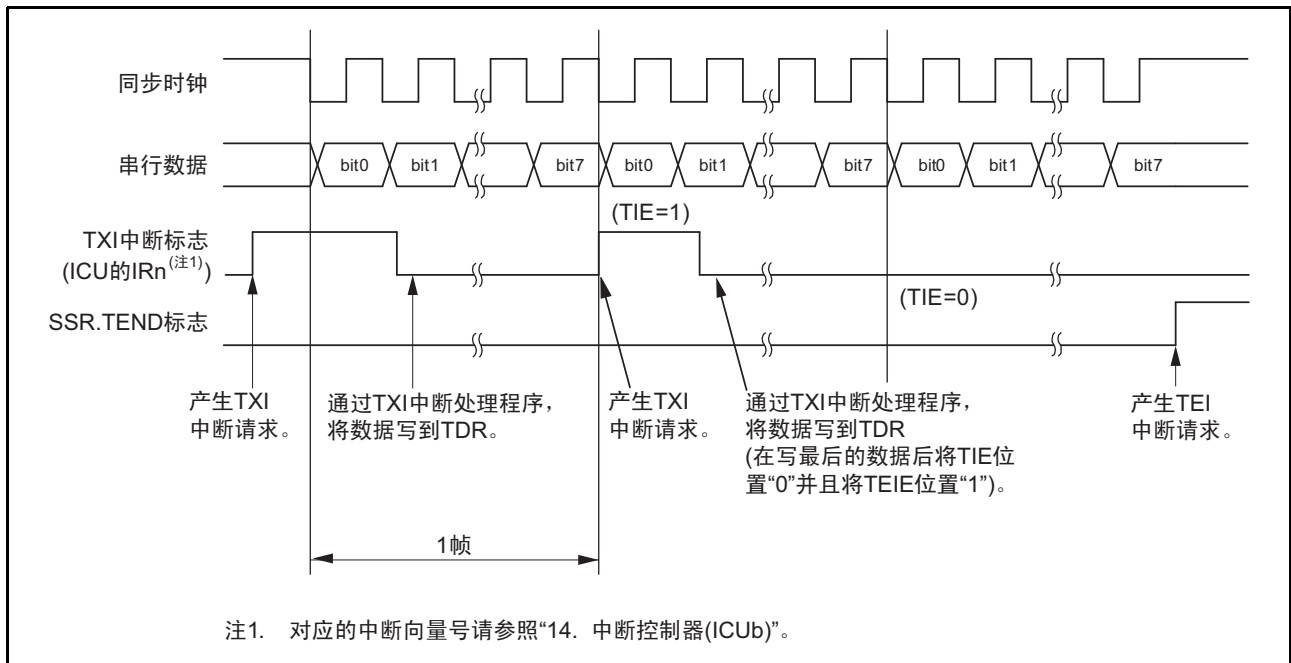


图 27.20 时钟同步模式的串行发送 (正在发送~发送结束) 例子

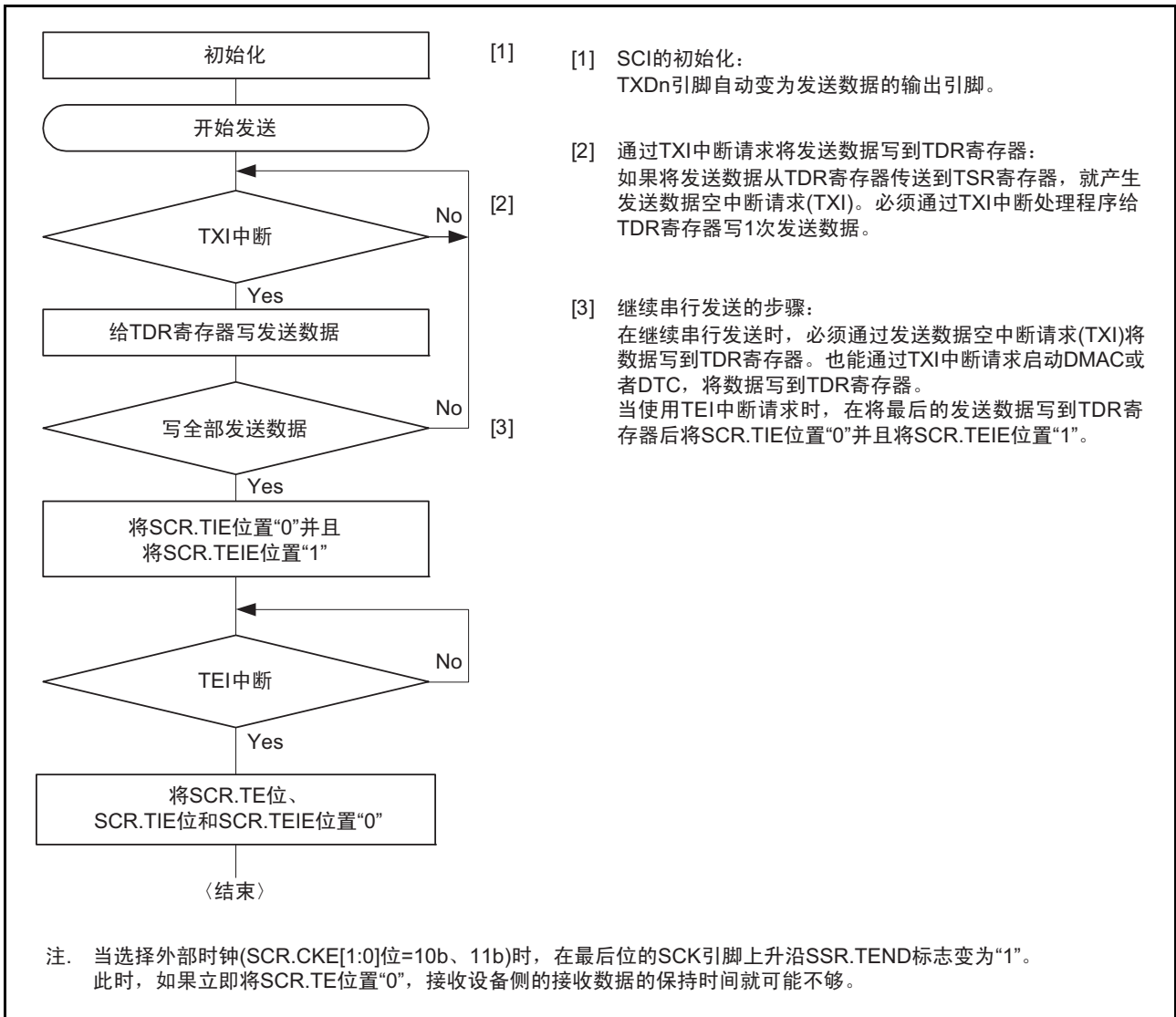


图 27.21 时钟同步模式的串行发送的流程图例子

27.5.5 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 27.22 和图 27.23 所示。

在接收串行数据时, SCI 的运行如下:

1. 如果将SCR.RE位置“1”, 就将RTS信号输出置为Low电平 (使用RTS功能的情况)。
2. SCI与同步时钟的输入同步或者与输出同步对内部进行初始化后开始接收, 并且将接收数据取到RSR寄存器。
3. 当发生溢出错误时, 将SSR.ORER标志置位。此时, 如果SCR.RIE位为“1”, 就产生ERI中断请求。接收数据不传送到RDR寄存器。
4. 当正常接收时, 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, 进行连续的接收。如果读被传送到RDR寄存器的接收数据, 就将RTS信号输出置为Low电平 (使用RTS功能的情况)。

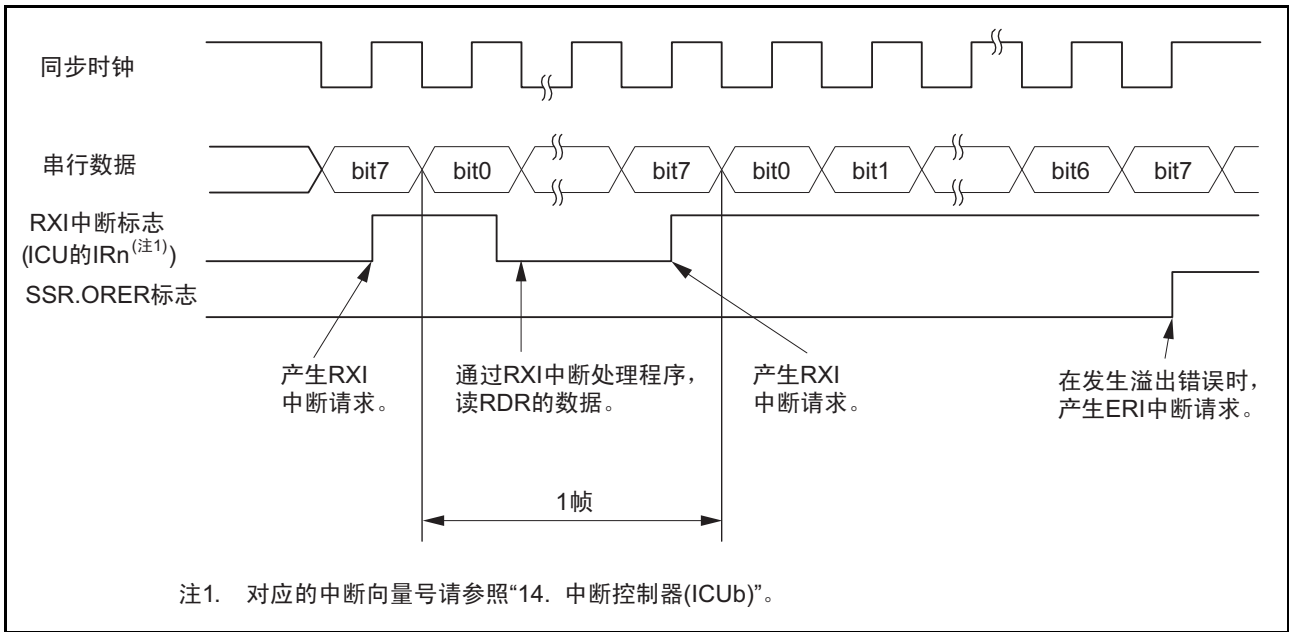


图 27.22 时钟同步模式的串行接收例子 (1) (不使用 RTS 功能的情况)

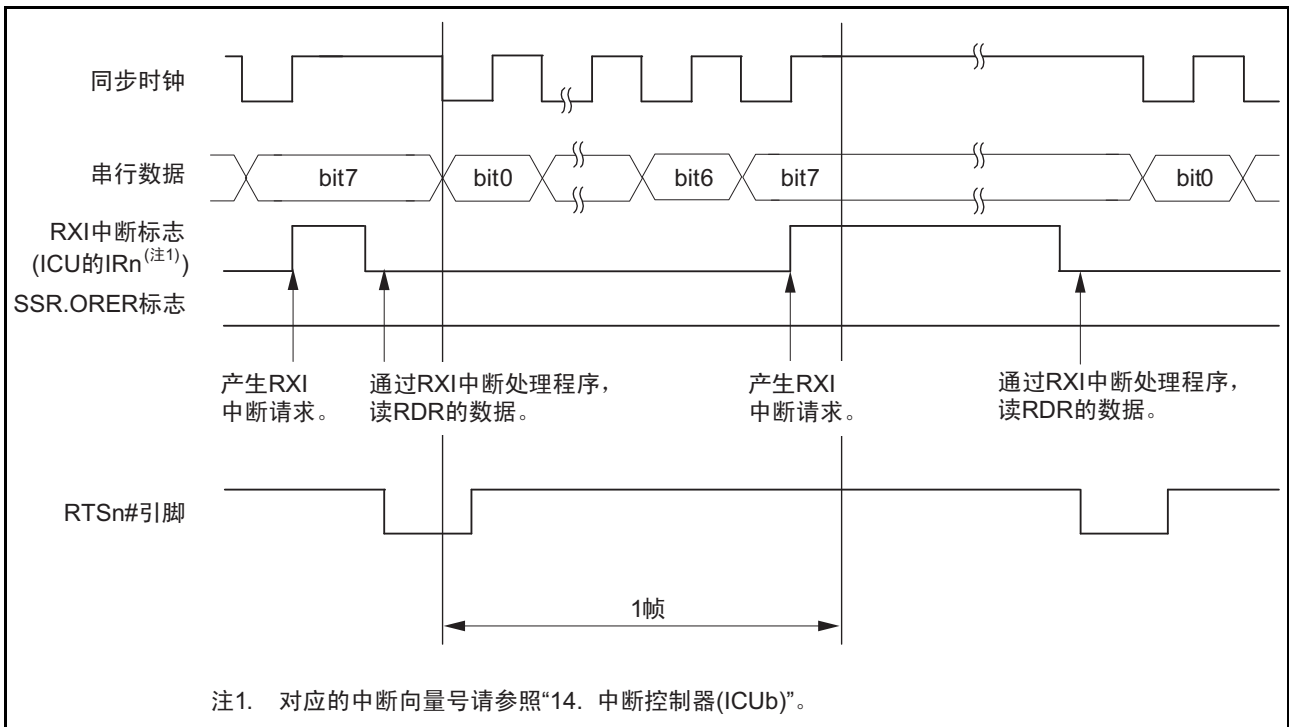


图 27.23 时钟同步模式的串行接收例子 (2) (使用 RTS 功能的情况)

在接收错误标志被置位的状态下，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 27.24 所示。

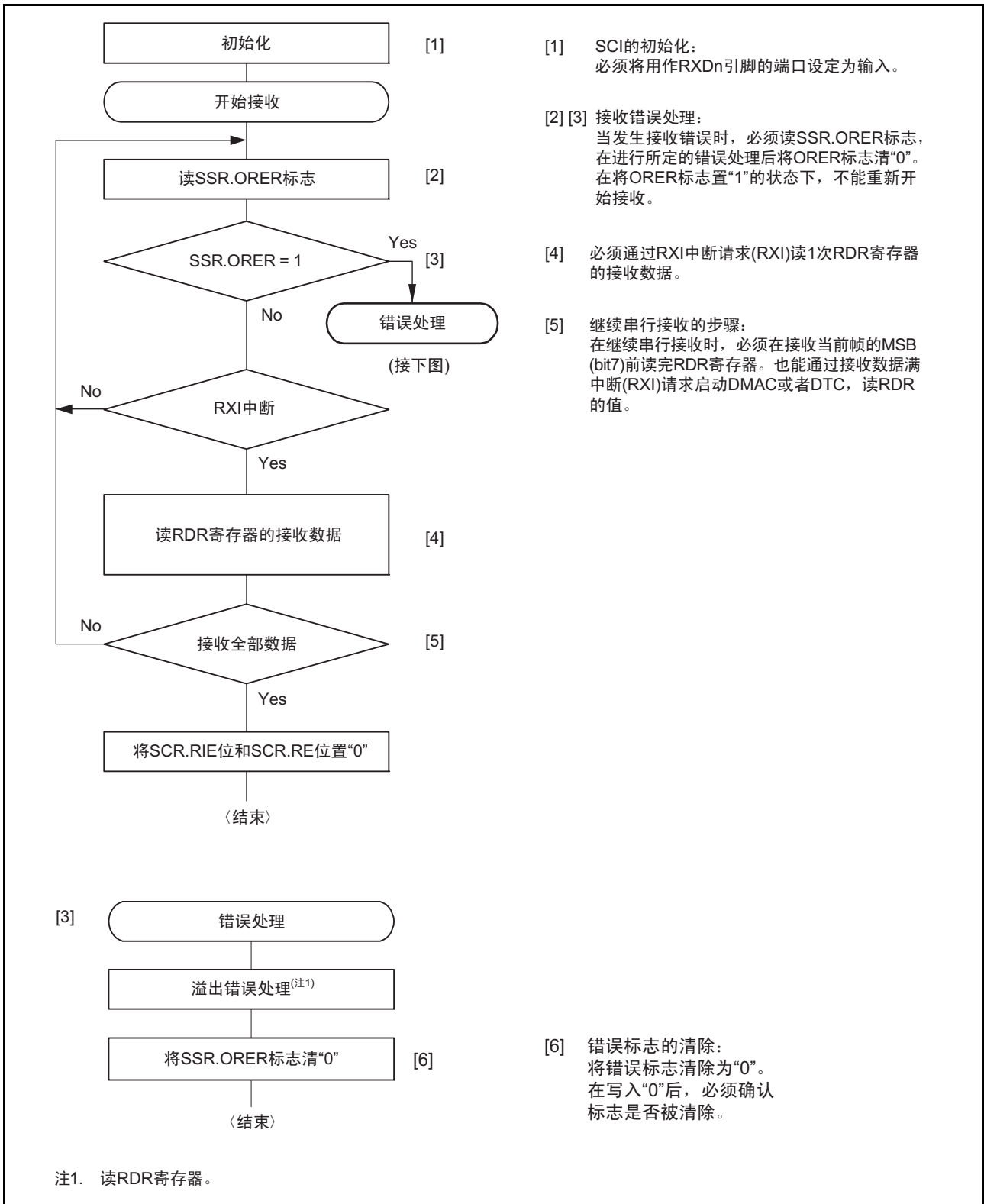


图 27.24 时钟同步模式的串行接收的流程图例子

27.5.6 串行数据的同时发送和接收 (时钟同步模式)

时钟同步模式的同时串行发送和接收的流程图例子如图 27.25 所示。

在对 SCI 进行初始化后, 必须按照以下步骤进行同时串行发送和接收。

要从发送转换为同时发送和接收时, 必须通过为“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, 用 1 条指令将 SCR.TIE 位、SCR.RIE 位、SCR.TE 位和 SCR.RE 位同时置“1”。

要从接收转换为同时发送和接收时, 必须在确认 SCI 处于接收结束状态后, 先将 SCR.RIE 位和 SCR.RE 位置“0”, 然后确认错误标志 (SSR.ORER、SSR.FER、SSR.PER) 已被置“0”, 最后用 1 条指令将 SCR.TIE 位、SCR.RIE 位、SCR.TE 位和 SCR.RE 位同时置“1”。

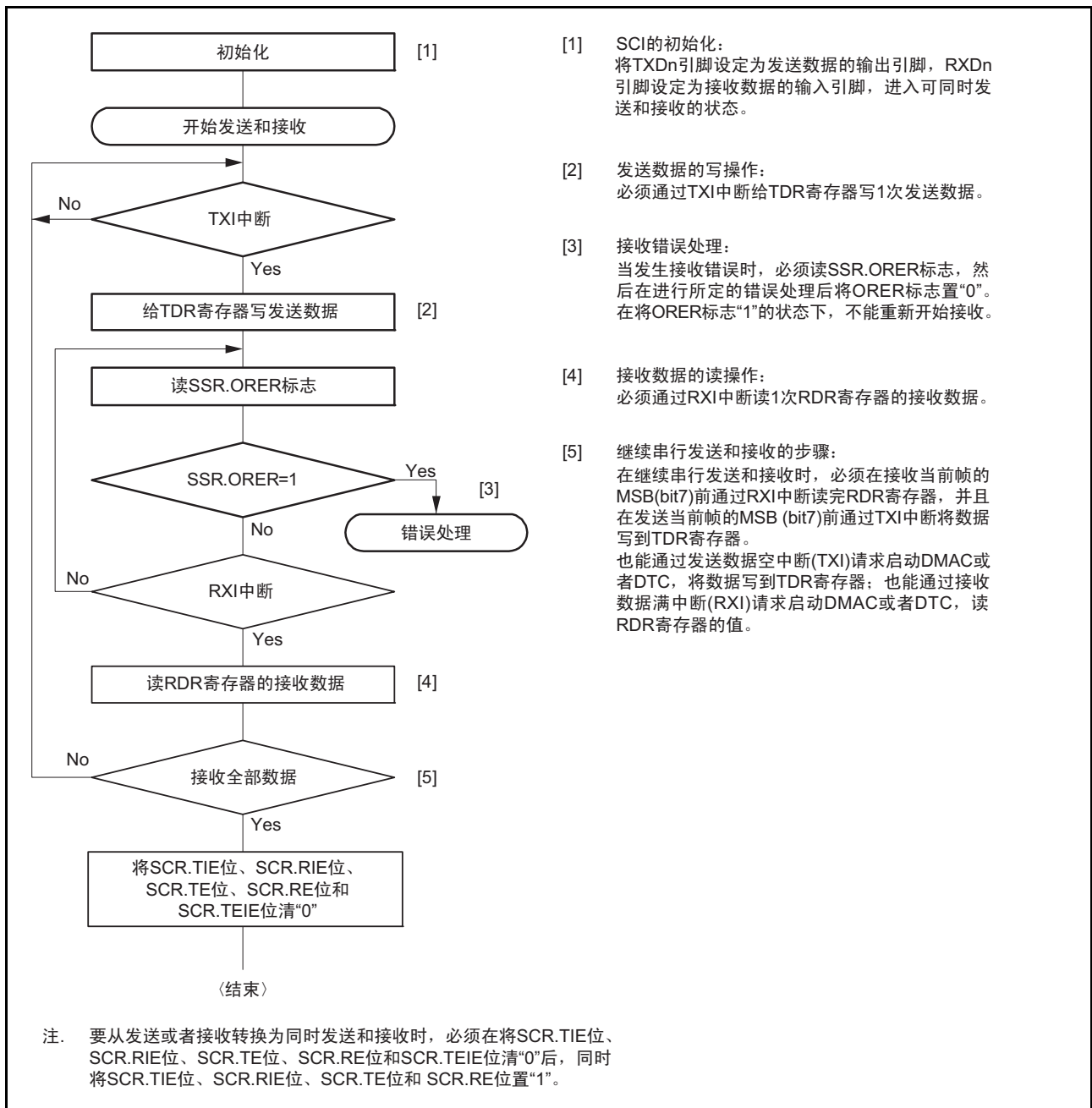


图 27.25 时钟同步模式的同时串行发送和接收的流程图例子

27.6 智能卡接口模式的运行

作为 SCI 的扩展功能，支持符合 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。

27.6.1 连接例子

智能卡 (IC 卡) 的连接例子如图 27.26 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”，就能将发送和接收闭合连接进行自诊断。

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 RX63T 群的输出端口用于复位信号的输出。

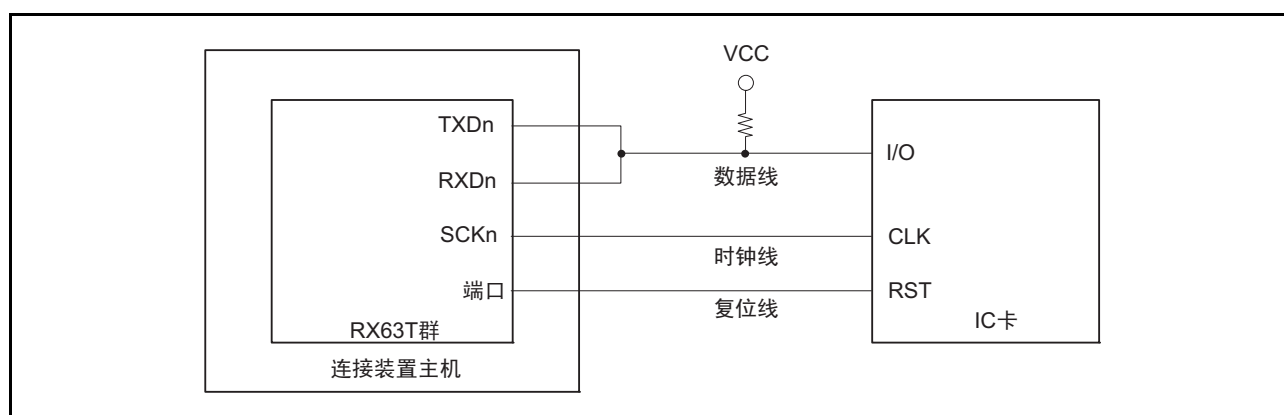


图 27.26 智能卡 (IC 卡) 的连接例子

27.6.2 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 27.27 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。

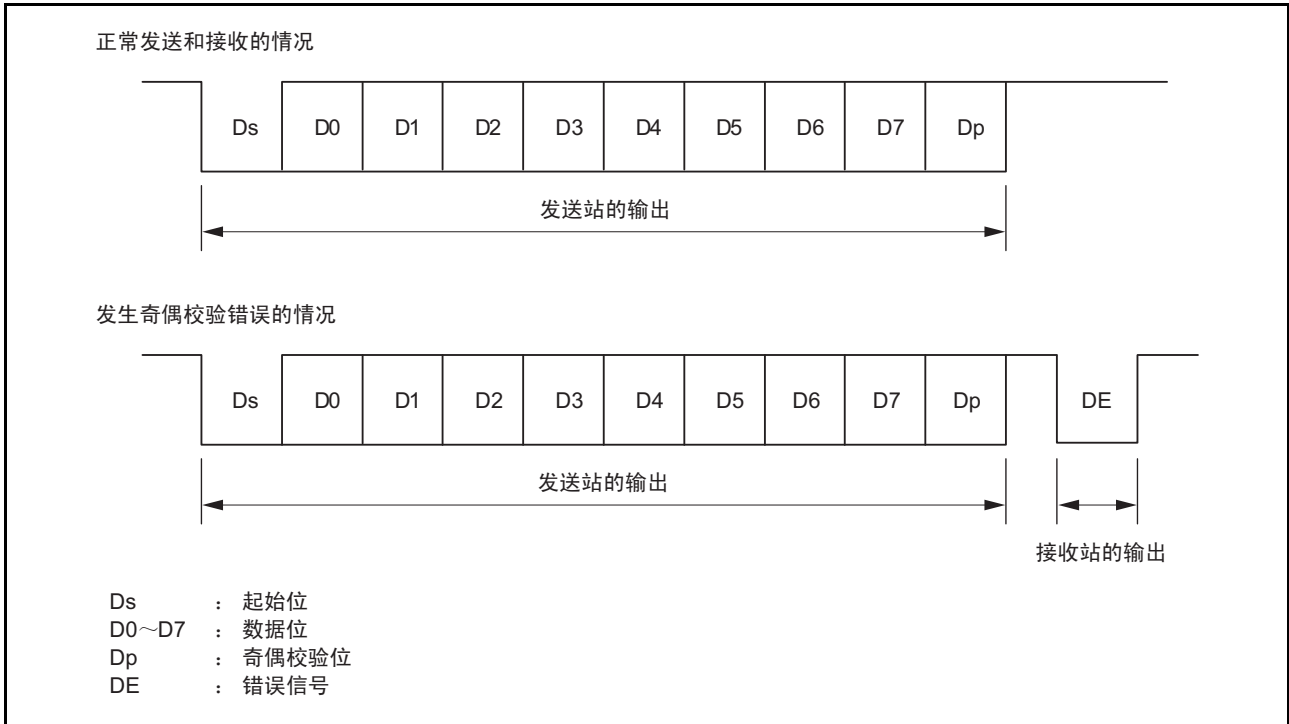


图 27.27 智能卡接口模式的数据格式

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 27.28 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 27.28 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”。

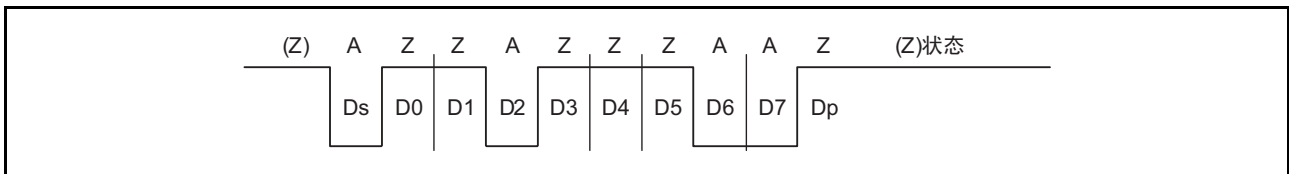


图 27.28 正向协议 (SCMR.SDIR 位 =0、SCMR.SINV 位 =0、SMR.PM 位 =0)

(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 27.29 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 RX63T 群中，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。

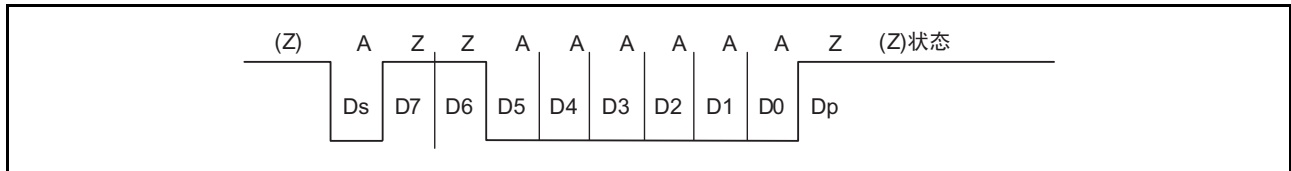


图 27.29 反向协议 (SCMR.SDIR 位 =1、SCMR.SINV 位 =1、SMR.PM 位 =1)

27.6.3 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 标志被置位，所以必须在接收下一帧的奇偶校验位前清除此位。
- 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，将 SSR.TEND 标志置位。
- 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”。

27.6.4 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 27.30 所示，通过在基本时钟的 16 个、32 个、186 个、128 个、46 个、64 个、93 个、256 个周期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：

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100 \quad [\%]$$

M：接收容限(%)

N：对应时钟的位速率比(N=32、64、372、256)

D：时钟的占空比(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：

$$M = \{0.5 - 1/(2 \times 372)\} \times 100 [\%] = 49.866\%$$

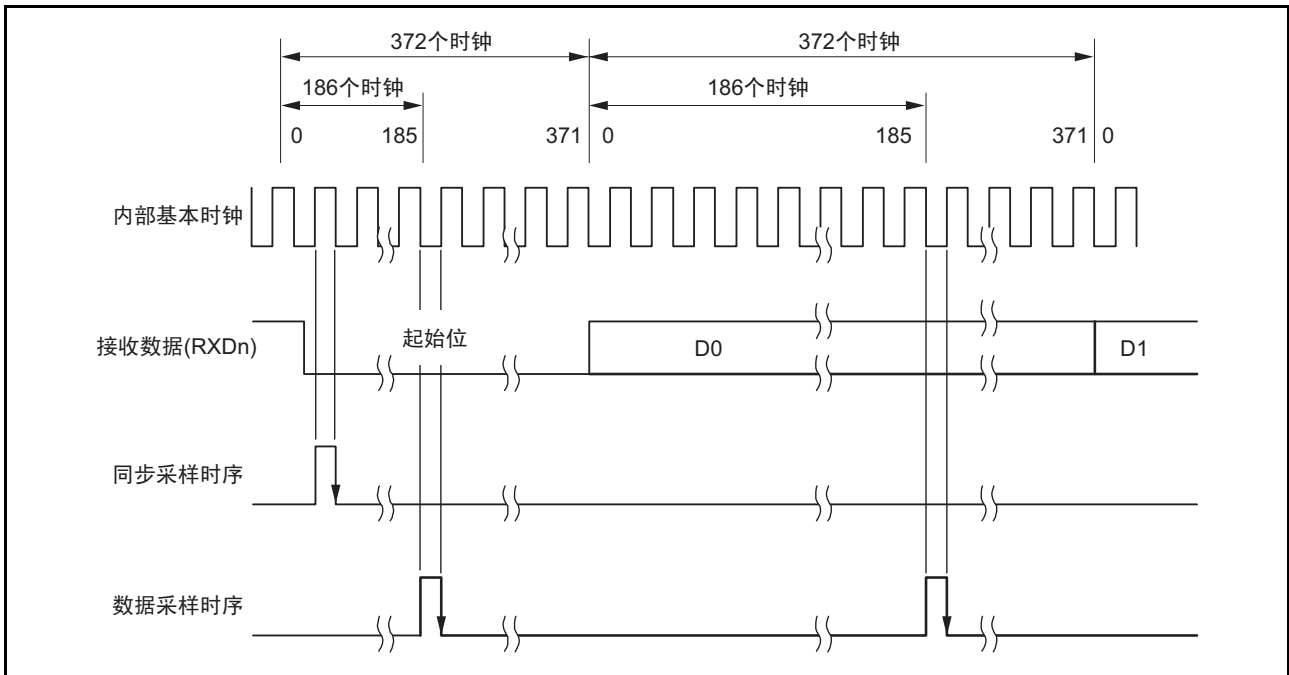


图 27.30 智能卡接口模式的接收数据的采样时序 (使用 372 倍时钟的情况)

27.6.5 SCI 的初始化 (智能卡接口模式)

在发送和接收数据前，必须按照以下步骤对 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. 必须设定 I/O 端口，使 TXDn 引脚、RXDn 引脚和 SCKn 引脚中所需的输入/输出功能有效。
3. 必须将 SSR 寄存器的错误标志 (ORER、ERS、PER) 置“0”。
4. 必须将 SIMR1.IICM 位置“0”并且将 SPMR.CKPH 位和 SPMR.CKPOL 位置“0”。
(如果不改变初始值，就能省略此步骤)
5. 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”。
6. 必须设定 SCMR.SDIR 位、SCMR.SINV 位和 SCMR.SMIF 位。TXDn 引脚和 RXDn 引脚为高阻抗状态。
7. 给 BRR 寄存器设定与位速率对应的值。
8. 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 CKE[0] 位置“1”，就从 SCKn 引脚输出时钟。
9. 必须设定 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 位同时置位。

要从接收模式转换为发送模式时，必须在确认接收已结束后从初始化开始进行，并且将 TE 位置“1”以及将 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，必须在确认发送已结束后从初始化开始进行，并且将 TE 位置“0”以及将 RE 位置“1”。能通过 SSR.TEND 标志确认发送的结束。

27.6.6 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 27.31 所示。

1. 如果在结束1帧的发送后对接收侧的错误信号进行采样, 就将SSR.ERS标志置“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清除ERS标志。
2. 在接收到错误信号的帧中, 不将SSR.TEND标志置位。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, 就不将ERS标志置位。
4. 在判断出包括重新传送的1帧的发送已结束后, 将SSR.TEND标志置位。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 27.33 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, 如果将 SCR.TIE 位置“1”, 就在 SSR.TEND 标志被置“1”时产生 TXI 中断请求。如果预先将 TXI 中断请求设定为 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, 将 TEND 标志自动置“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”的状态并且不启动 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, 在发生错误时不自动清除 ERS 标志, 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, 并且清除 ERS 标志。

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“17. DMA 控制器 (DMACA)”和“18. 数据传送控制器 (DTCa)”。

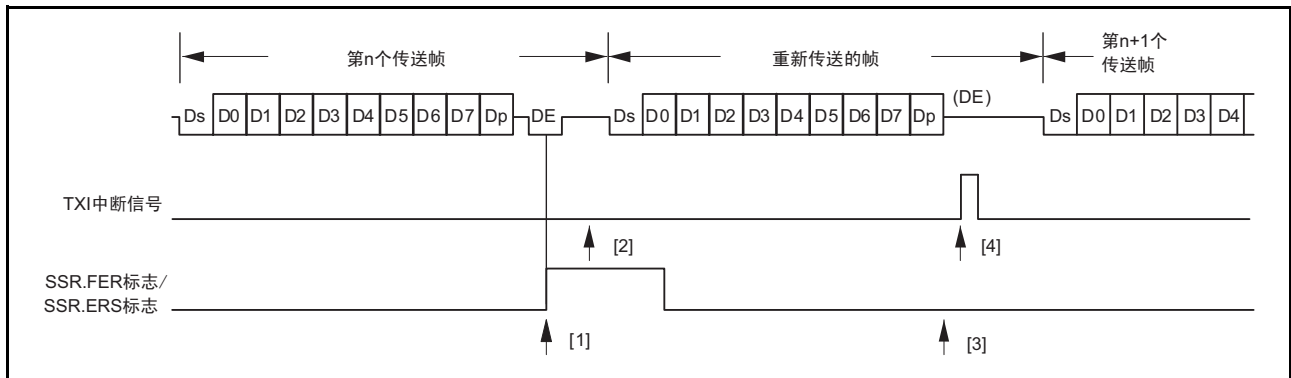


图 27.31 SCI 发送模式的重新传送 (发送时的重新传送)

SSR.TEND 标志的置位时序取决于 SMR.GM 位的设定，TEND 标志的产生时序如图 27.32 所示。

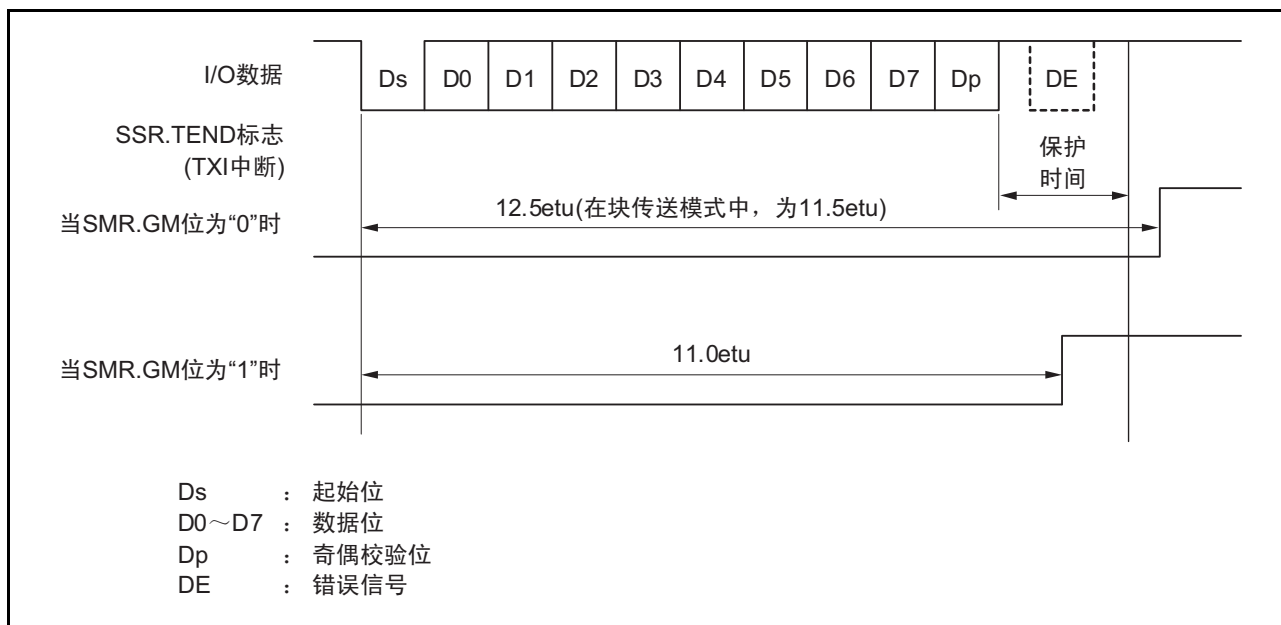


图 27.32 发送时的 SSR.TEND 标志的产生时序

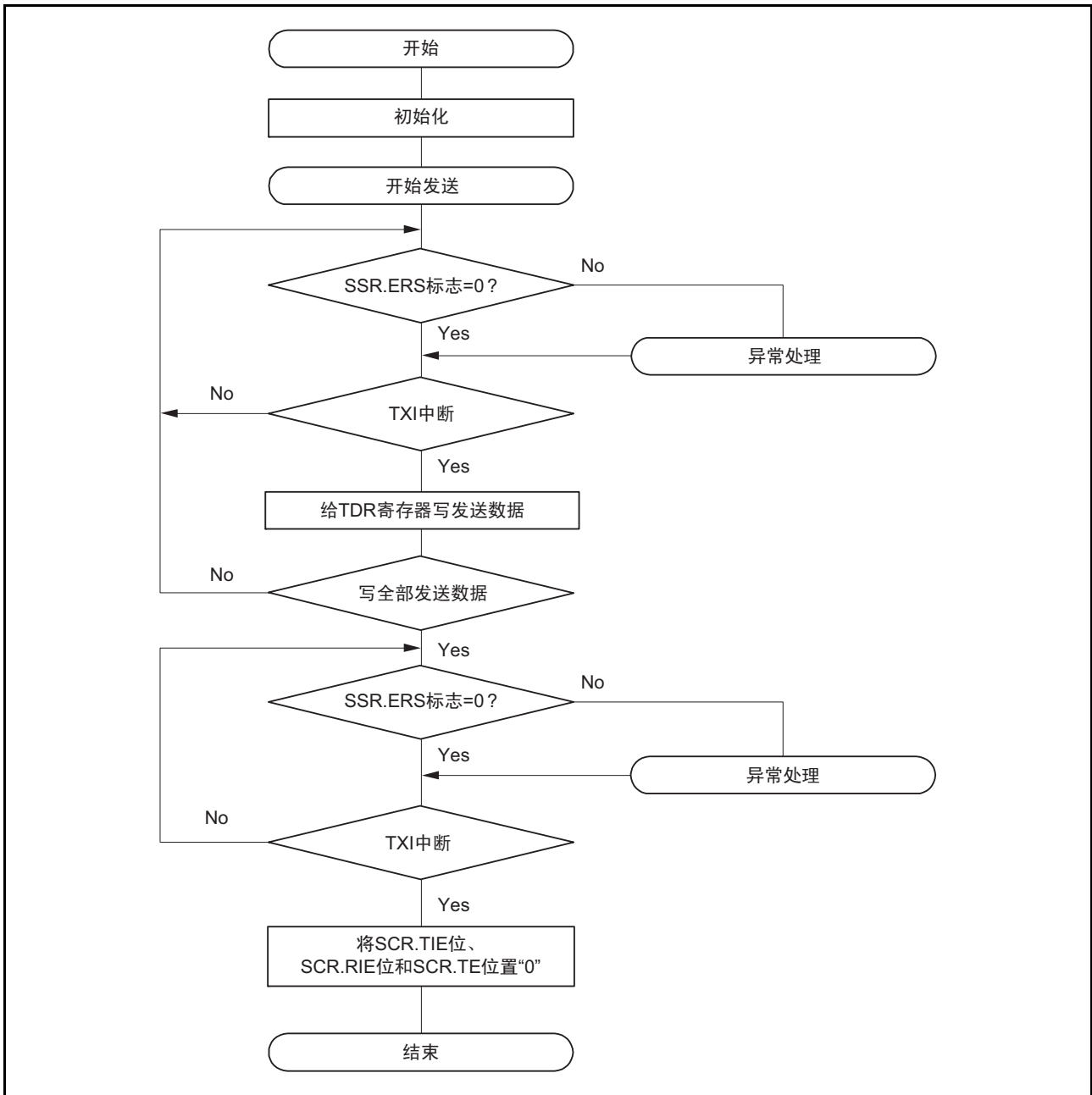


图 27.33 智能卡接口发送的流程图例子

27.6.7 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 27.34 所示。

1. 如果在接收数据时检测到奇偶校验错误, 就将 **SSR.PER** 标志置“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清除 **PER** 标志。
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, 就不将 **SSR.PER** 标志置位。
4. 如果在判断出接收正常结束后 **RIE** 位为“1”, 就产生 **RXI** 中断请求。

串行接收的流程图例子如图 27.35 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** 标志被置“1”, 就产生接收错误中断 (**ERI**) 请求, 因此必须清除错误标志。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。

即使在接收时发生奇偶校验错误并且 **PER** 标志被置“1”, 也将接收数据传送到 **RDR** 寄存器, 因此能读此数据。

注. 有关块传送模式, 请参照“27.3 异步模式的运行”。

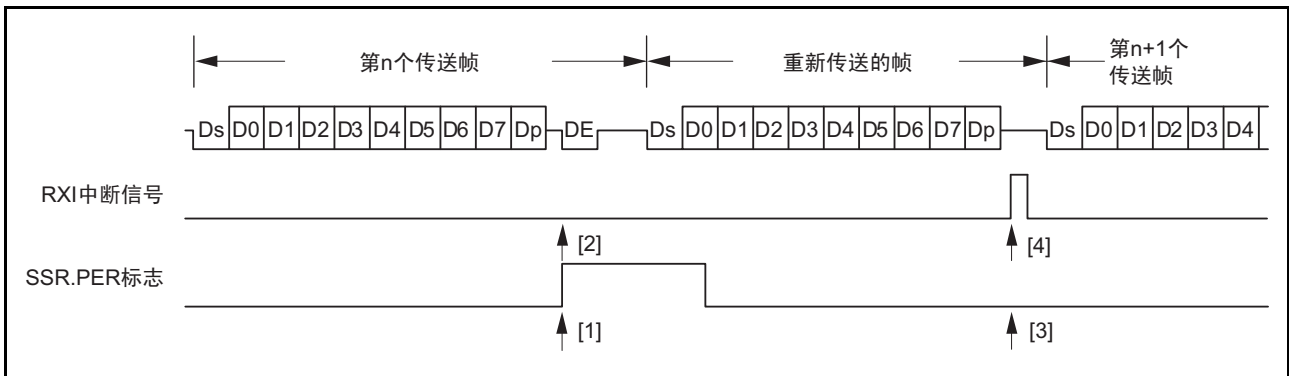


图 27.34 SCI 接收模式的重新传送 (接收时的重新传送)

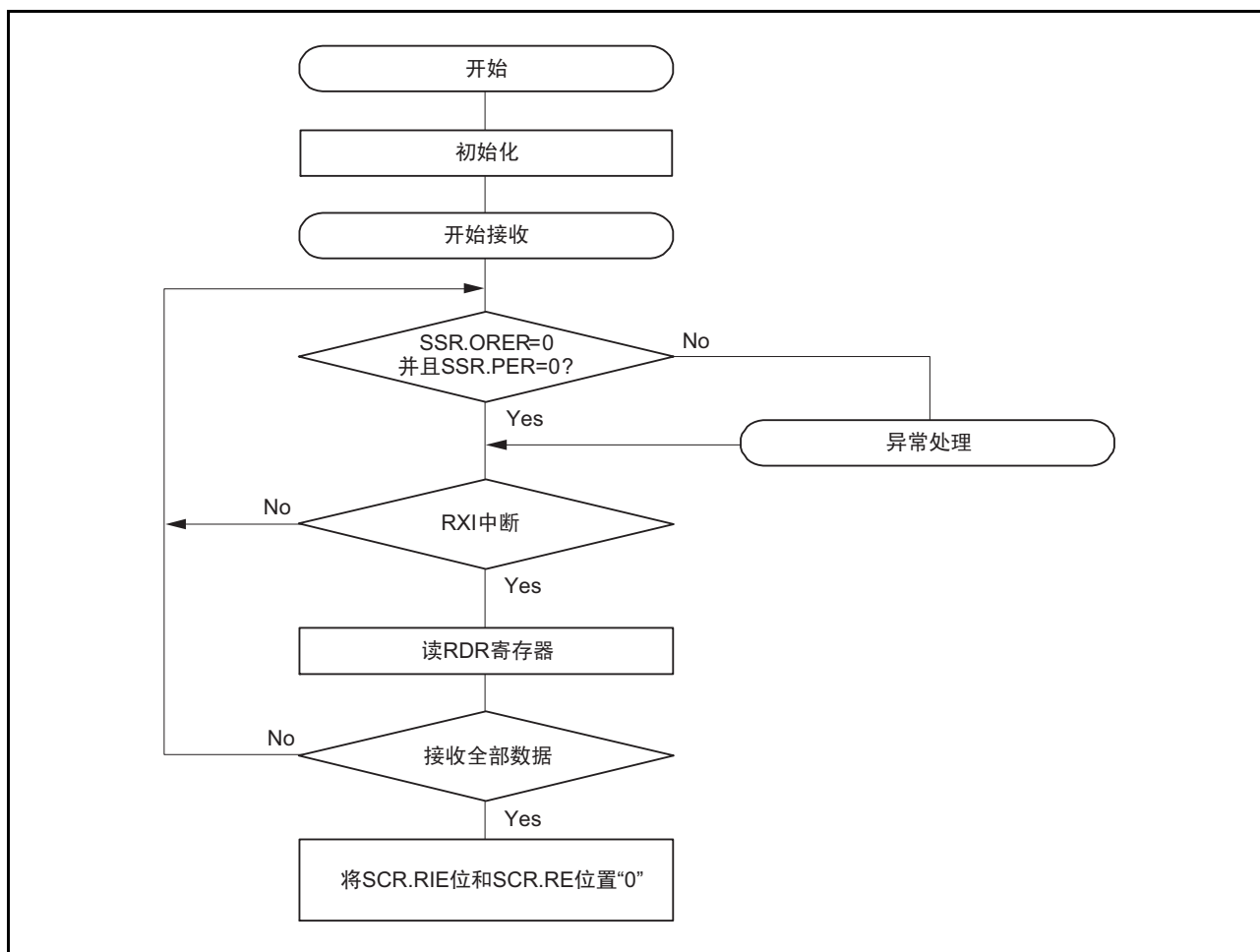


图 27.35 智能卡接口接收的流程图例子

27.6.8 时钟的输出控制

当 SMR.GM 位为“1”时，能通过 SCR.CKE[1:0] 位固定时钟输出。此时，能将时钟脉冲的最小宽度设定为指定的宽度。

时钟输出固定时序如图 27.36 所示，这是在 GM 位为“1”并且 CKE1 位为“0”的状态下控制 CKE0 位的例子。

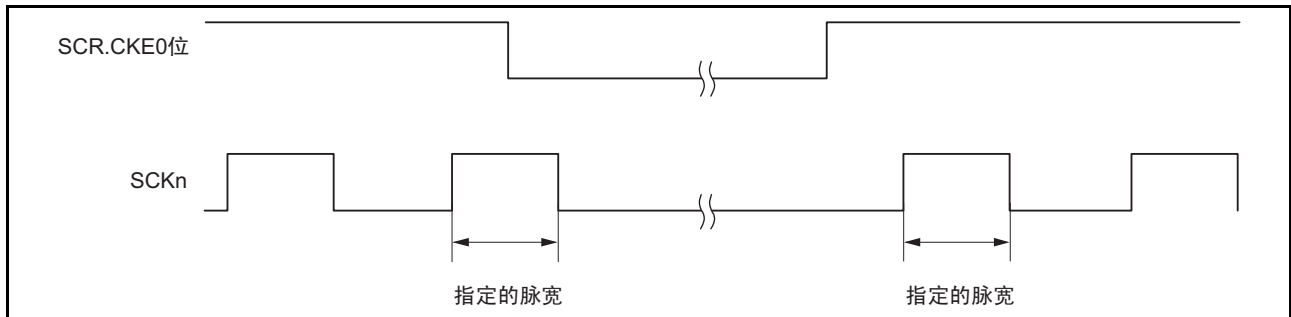


图 27.36 时钟输出固定时序

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，必须按照以下步骤进行转换处理：

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. 必须设定 SCR.CKE[1] 位和 I/O 端口功能，将 SCKn 引脚固定为指定的输出。
3. 必须设定 SMR 寄存器和 SCMR 寄存器，转换为智能卡接口模式的运行。
4. 必须将 SCR.CKE[0] 位置“1”，开始时钟的输出。

(2) 转换模式时

(a) 从智能卡接口模式转移到软件待机模式的情况

1. 必须设定 I/O 端口，使 SCKn 引脚变为软件待机模式中所需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.CKE[1] 位设定软件待机模式时的输出固定状态的值。
3. 必须给 SCR.CKE[0] 位写“0”，停止时钟。
4. 必须等待 1 个串行时钟周期。在此期间，保持占空比并且以指定的电平固定时钟的输出。
5. 必须转移到软件待机模式状态。

(b) 从软件待机模式返回到智能卡接口模式的情况

6. 必须解除软件待机模式状态。
7. 必须将 SCR.CKE[0] 位置“1”，输出时钟。开始以正常的占空比生成信号。

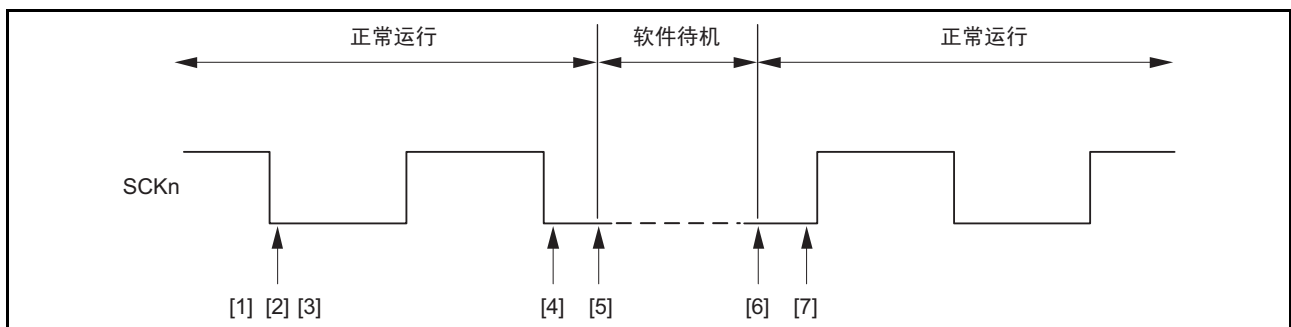


图 27.37 时钟的停止和重新启动的步骤

27.7 简易 I²C 模式的运行

简易 I²C 总线格式由 8 位数据和 1 个应答位构成。接在开始条件和重新开始条件之后的帧为从属地址帧，用于主控设备指定通信对象的从属设备。指定的从属设备有效到指定新的从属设备或者满足停止条件为止。从 MSB 开始依次发送各帧中的 8 位数据。

I²C 总线格式和总线时序分别如图 27.38 和图 27.39 所示。

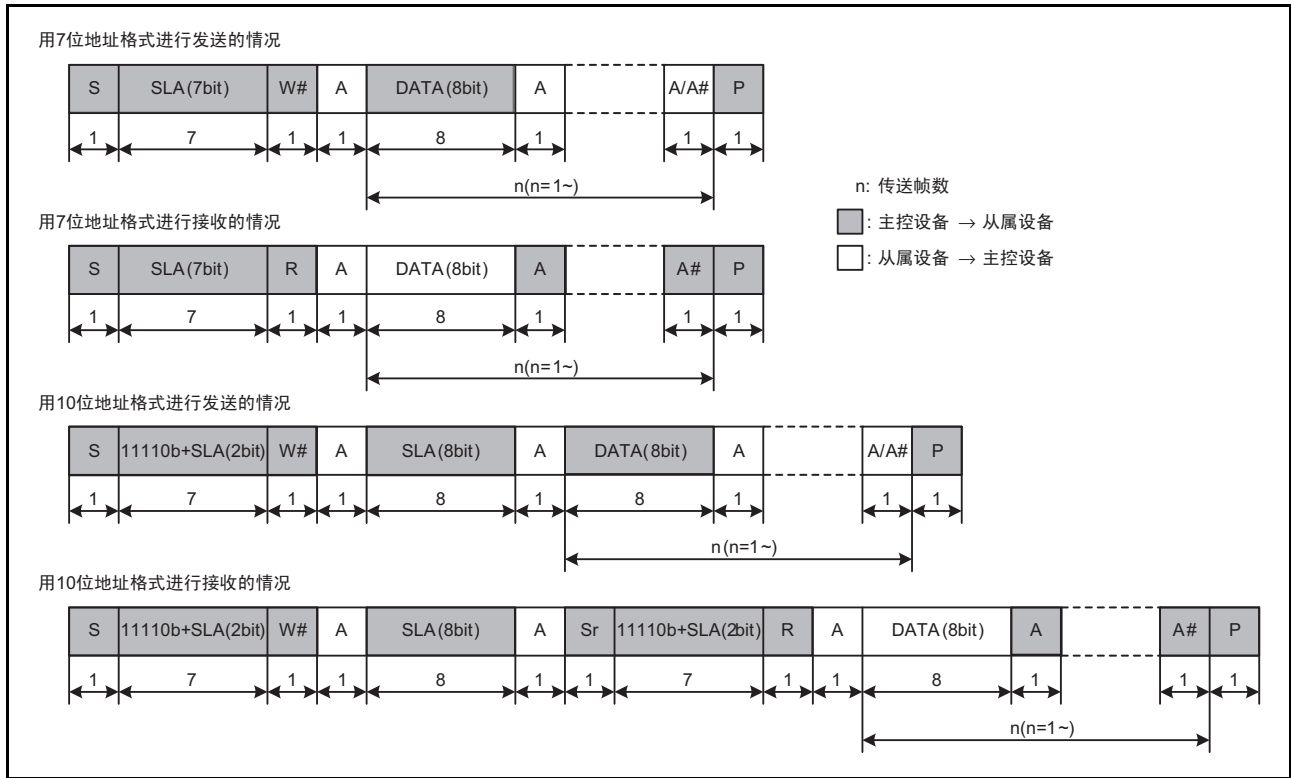


图 27.38 I²C 总线格式

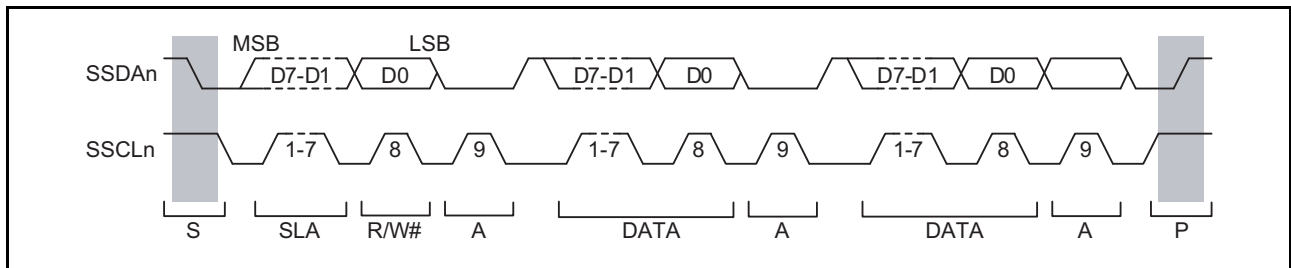


图 27.39 I²C 总线时序 (SLA 为 7 位的情况)

- S : 表示开始条件。主控设备在 SSCLn 线为 High 电平的状态下将 SSDAn 线从 High 电平变为 Low 电平。
- SLA : 表示从属地址。主控设备选择从属设备。
- R/W# : 表示发送和接收的方向。当 R/W# 为“1”时，将数据从从属设备发送到主控设备；当 R/W# 为“0”时，将数据从主控设备发送到从属设备。
- A/A# : 表示应答 (在主控发送模式中，从属设备返回应答；在主控接收模式中，主控设备返回应答)。ACK 返回 Low 电平，NACK 返回 High 电平。
- Sr : 表示重新开始条件。主控设备在 SSCLn 线为 High 电平的状态下经过准备时间后将 SSDAn 线从 High 电平变为 Low 电平。
- DATA : 表示发送和接收数据。
- P : 表示停止条件。主控设备在 SSCLn 线为 High 电平的状态下将 SSDAn 线从 Low 电平变为 High 电平。

27.7.1 开始条件、重新开始条件和停止条件的生成

通过给 SIMR3.IICSTAREQ 位写“1”生成开始条件。开始条件的生成如下所示：

- 降低SSDAn线的电平（从High电平变为Low电平），SSCLn线保持释放状态。
- 确保BRR寄存器所设位速率的一半时间以及开始条件的保持时间。
- 降低SSCLn线的电平（从High电平变为Low电平），将SIMR3.IICSTAREQ位清“0”并且输出开始条件的生成中断请求。

通过给 SIMR3.IICRSTAREQ 位写“1”生成重新开始条件。重新开始条件的生成如下所示：

- 释放SSDAn线，SSCLn线保持Low电平。
- 确保BRR寄存器所设位速率的一半时间以及SSCLn线的Low电平时间。
- 释放SSCLn线（从Low电平变为High电平）。
- 在检测到SSCLn线的High电平后，确保BRR寄存器所设位速率的一半时间以及重新开始条件的准备时间。
- 降低SSDAn线的电平（从High电平变为Low电平）。
- 确保BRR寄存器所设位速率的一半时间以及重新开始条件的保持时间。
- 降低SSCLn线的电平（从High电平转移到Low电平），将SIMR3.IICRSTAREQ位清“0”并且输出重新开始条件的生成中断请求。

通过给 SIMR3.IICSTPREQ 位写“1”生成停止条件。停止条件的生成如下所示：

- 降低SSDAn线的电平（从High电平变为Low电平），SSCLn线保持Low电平。
- 确保BRR寄存器所设位速率的一半时间以及SSCLn线的Low电平时间。
- 释放SSCLn线（从Low电平变为High电平）。
- 在检测到SSCLn线的High电平后，确保BRR寄存器所设位速率的一半时间以及停止条件的准备时间。
- 释放SSDAn线的电平（从Low电平变为High电平），将SIMR3.IICSTPREQ位清“0”并且输出停止条件的生成中断请求。

开始条件、重新开始条件和停止条件生成的运行时序如图 27.40 所示。

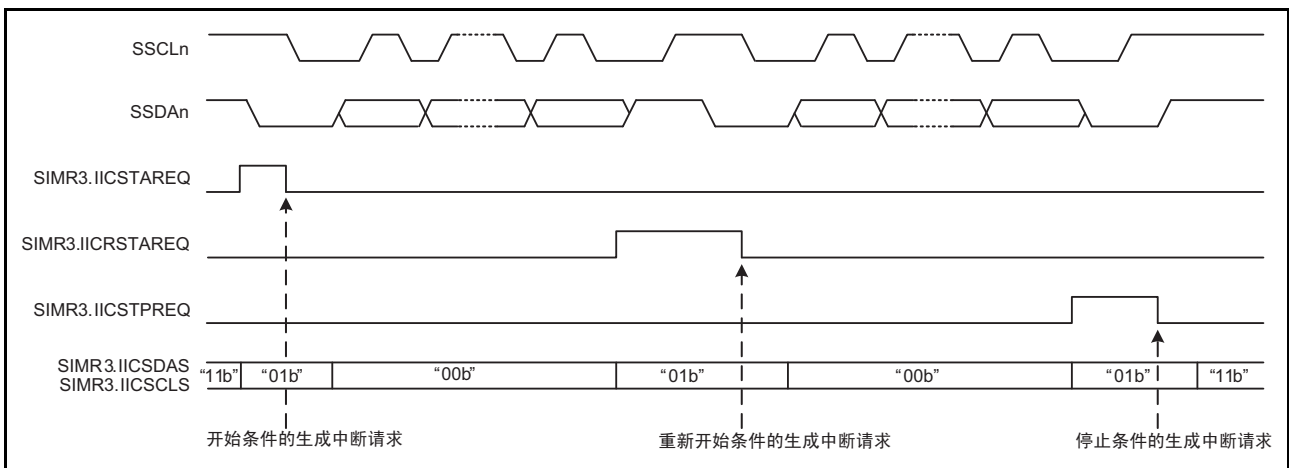


图 27.40 开始条件、重新开始条件和停止条件生成的运行时序

27.7.2 时钟同步

有时为了使通信对象的从属设备插入等待而将 SSCLn 线置为 Low 电平。如果将 SIMR2.IICCSC 位置“1”，就在内部 SSCLn 时钟和 SSCLn 引脚的输入电平不同时对同步的取得进行控制。

当 SIMR2.IICCSC 位为“1”时，如果内部 SSCLn 时钟从 Low 电平变为 High 电平，就在 SSCLn 引脚的输入为 Low 电平期间停止 High 电平期间的计数，一旦 SSCLn 引脚的输入变为 High 电平，就开始 High 电平期间的计数。此时，从 SSCLn 引脚变为 High 电平到开始 High 电平期间的计数为止，需要 SSCLn 引脚输入延迟、SSCLn 引脚输入的噪声滤波器延迟（2 ~ 3 个的噪声滤波器的采样时钟周期）和内部处理延迟（1 ~ 2 个 PCLK 周期）的时间。在此期间，即使其他设备不将 SSCLn 线置为 Low 电平，也会延长内部 SSCLn 时钟的 High 电平时间。

当 SIMR2.IICCSC 位为“1”时，与 SSCLn 引脚输入和内部 SSCLn 时钟的逻辑与同步进行数据的发送和接收；当 SIMR2.IICCSC 位为“0”时，与内部 SSCLn 时钟同步进行数据的接收和发送。

在从发行开始条件、重新开始条件和停止条件的生成请求后到内部 SSCLn 时钟从 Low 电平变为 High 电平为止的期间，如果从属设备插入等待，生成结束就延长此期间的的时间。

如果在内部 SSCLn 时钟变为 High 电平后从属设备插入等待，即使在此等待期间也不停止，而发行生成结束的中断请求，但是不保证条件的生成。

时钟同步的运行例子如图 27.41 所示。

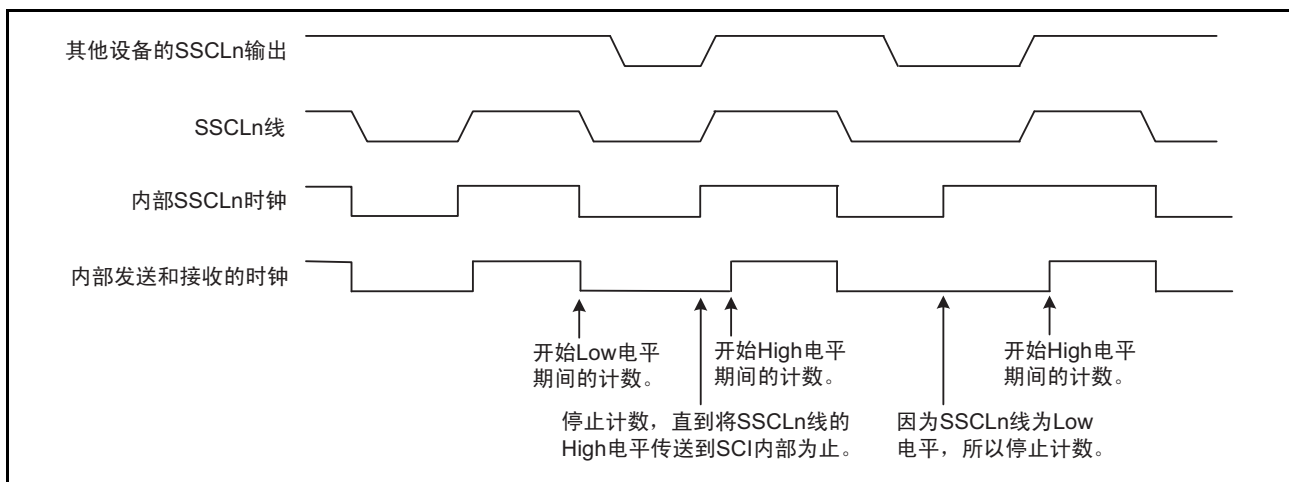


图 27.41 时钟同步的运行例子

27.7.3 SSDA 输出延迟

对于 SSCLn 引脚输出的下降沿，能通过 SIMR1.IICDL[4:0] 位延迟 SSDAn 引脚的输出。延迟时间能以内部波特率发生器的时钟源为基准（以 PCLK 为基准通过 SMR.CKS[1:0] 选择的分频时钟），选择 0 ~ 31 个周期的延迟时间。延迟 SSDAn 引脚输出的对象为开始条件信号、重新开始条件信号、停止条件信号、8 位发送数据和应答。

如果 SSDA 输出延迟时间比 SSCLn 引脚输出的下降时间短，SSDAn 引脚的输出就可能在 SSCLn 引脚输出的下降过程中开始变化，使从属设备发生误动作。因此必须将 SSDA 输出延迟设定为大于 SSCLn 引脚输出的下降时间的最大值（在 I²C 的标准模式和快速模式中为 300ns）。

SSDA 输出延迟的时序如图 27.42 所示。

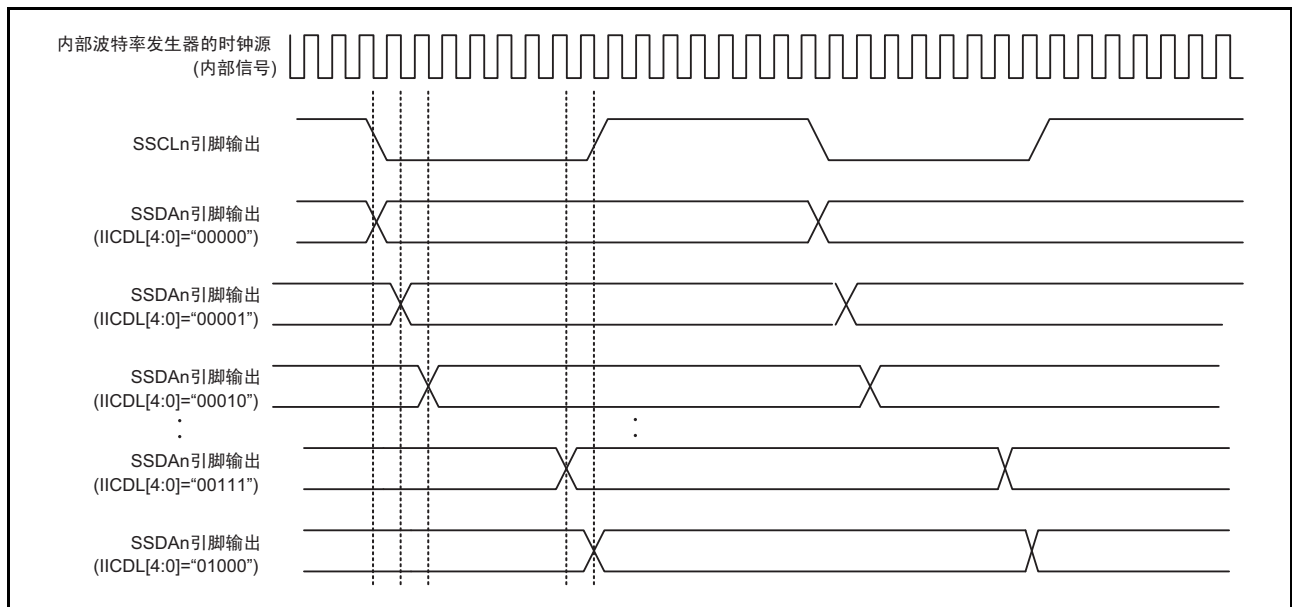


图 27.42 SSDA 输出延迟的时序

27.7.4 SCI 的初始化 (简易 I²C 模式)

必须在发送和接收数据前给 SCR 寄存器写初始值“00h”，并且按照图 27.43 的流程图例子进行初始化。

必须在将 SCR 寄存器置为初始值后更改运行模式和通信格式。另外，在简易 I²C 模式中，必须在端口侧设定通信端口的漏极开路。

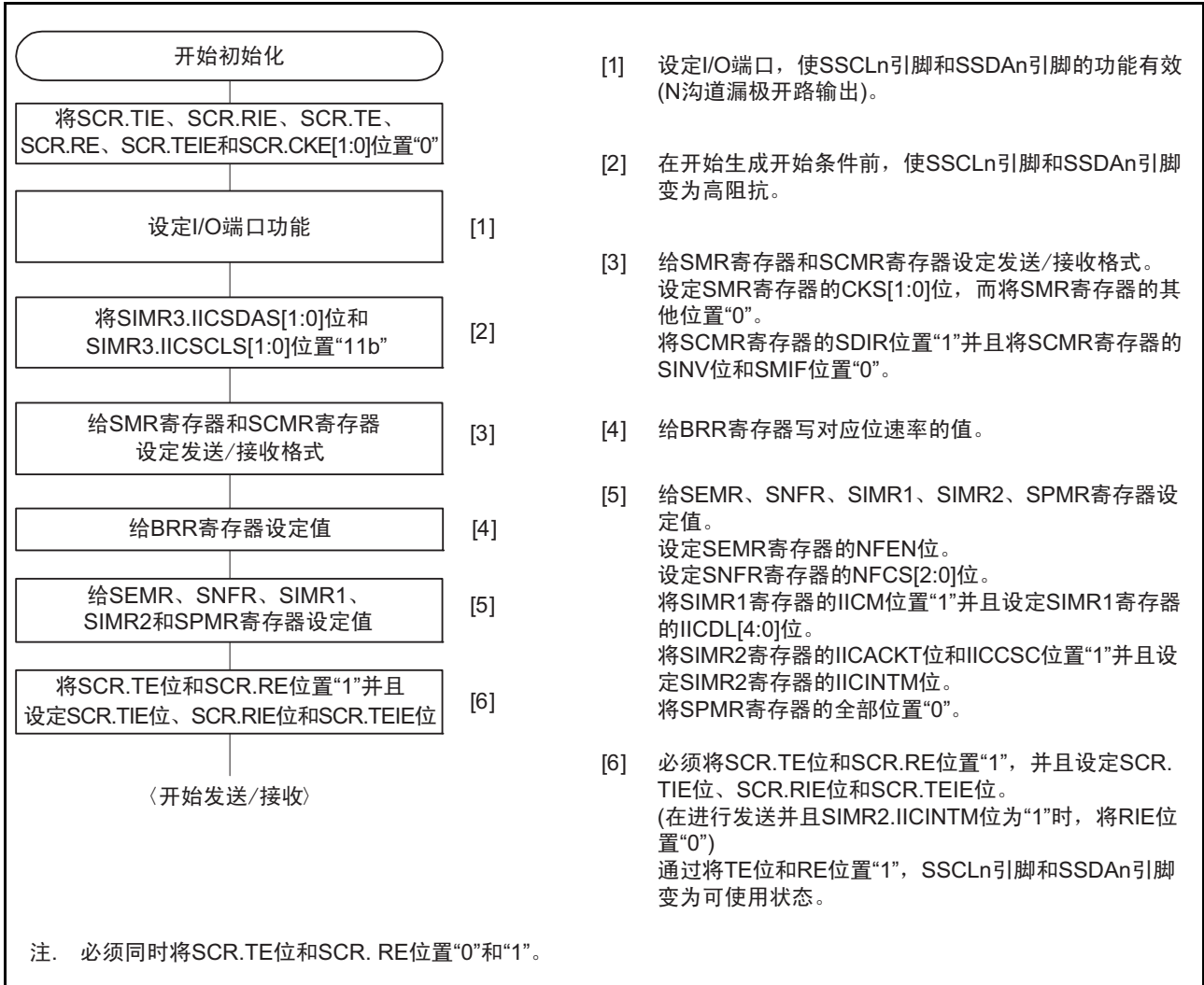


图 27.43 SCI 的初始化流程图例子 (简易 I²C 模式)

27.7.5 主控发送 (简易 I²C 模式)

简易 I²C 模式的主控发送运行例子如图 27.44 和图 27.45 所示，数据发送的流程图例子如图 27.46 所示。这些例子都假设将 SIMR2.IICINTM 位置“1”（使用接收中断和发送中断）并且将 SCR.RIE 位置“0”（禁止接收中断请求）。有关 STI 中断，请参照表 27.28。

在 10 位从属地址模式中，重复 2 次图 27.46 中的步骤 [3] ~ [4]。

简易 I²C 模式中的发送结束中断 (TXI) 发生在 1 帧的通信结束时，和时钟同步发送时的 TXI 中断请求发生时序不同。

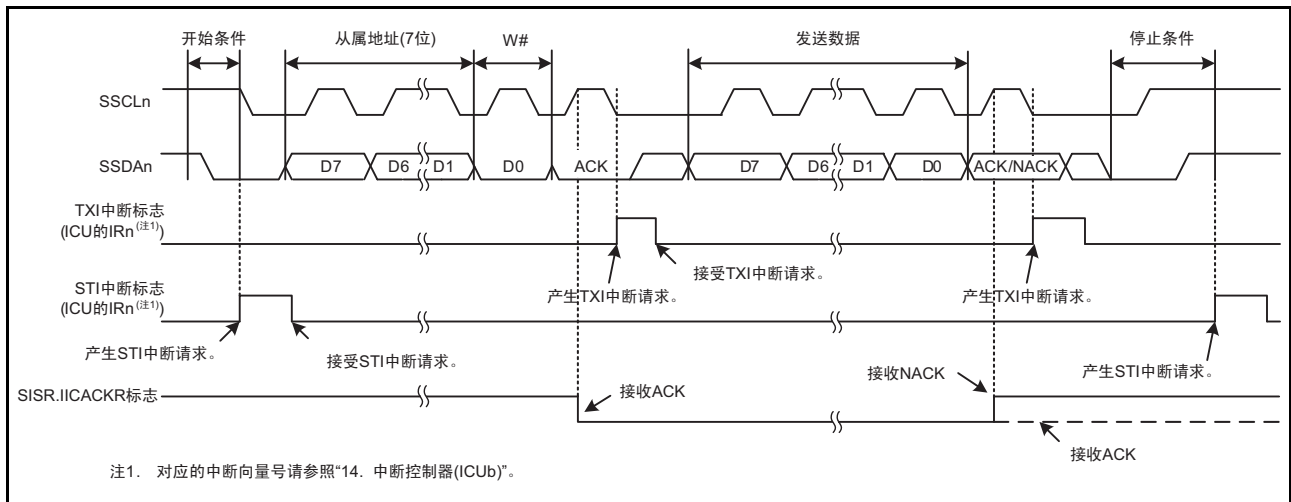


图 27.44 简易 I²C 总线模式的主控发送运行例子 1 (使用 7 位从属地址、发送中断和接收中断的情况)

如果在主控发送时将 SIMR2.IICINTM 位置“0”（使用 ACK 中断和 NACK 中断），就以 ACK 中断为触发，启动 DTC 或者 DMAC 并且发送所需字节数的数据。如果接收到 NACK，就以 NACK 中断为触发，中止发送并且进行重新发送等错误处理。

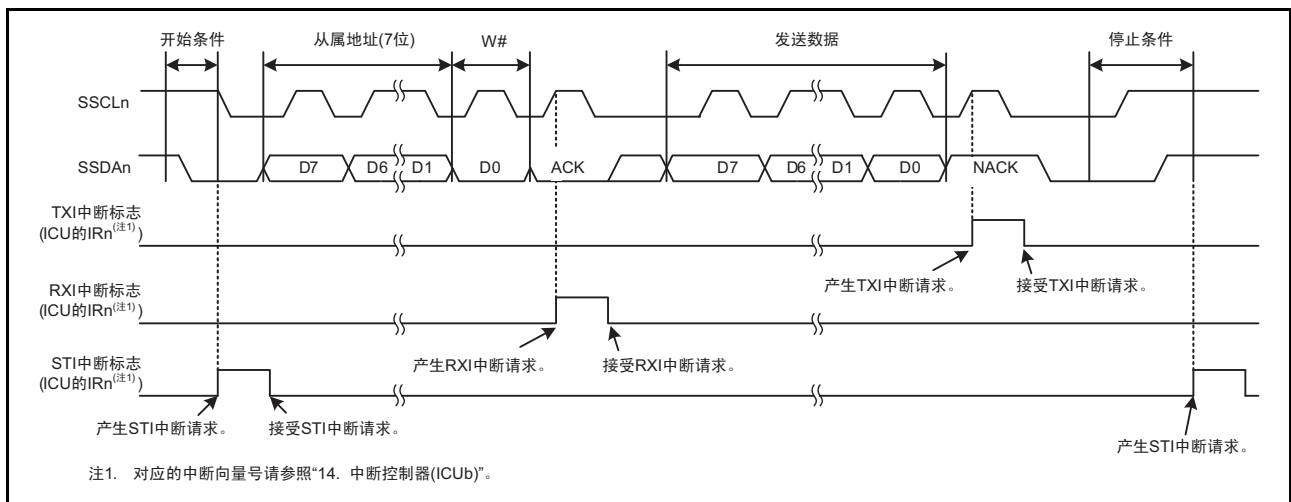


图 27.45 简易 I²C 总线模式的主控发送运行例子 2 (使用 7 位从属地址、ACK 中断和 NACK 中断的情况)

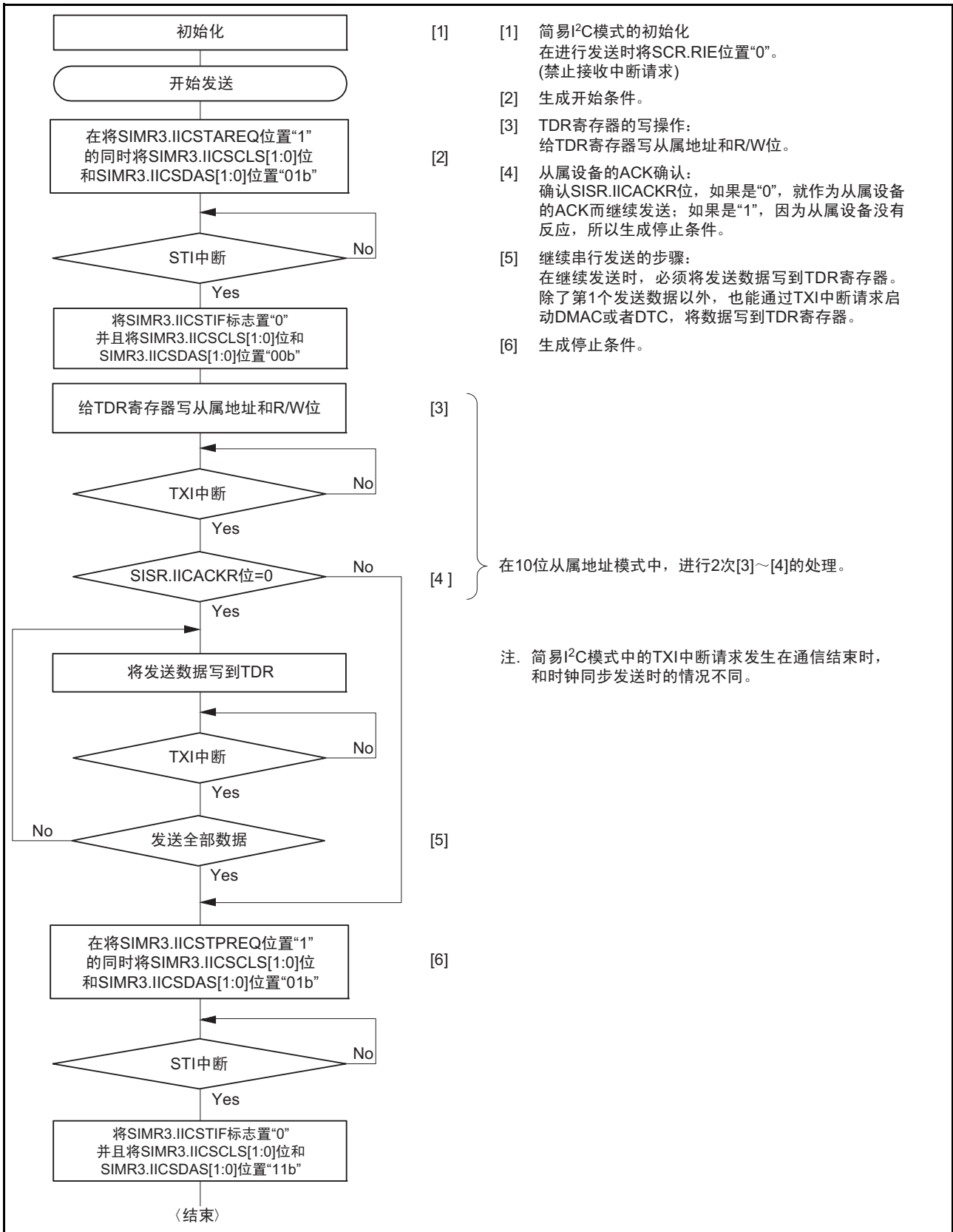


图 27.46 简易 I²C 模式的主控发送运行流程图例子 (使用发送中断和接收中断的情况)

27.7.6 主控接收 (简易 I²C 模式)

简易 I²C 模式的主控接收运行例子和主控接收的流程图例子分别如图 27.47 和图 27.48 所示。这些例子都假设将 SIMR2.IICINTM 位置“1” (使用接收中断和发送中断)。

简易 I²C 模式的发送结束中断 (TXI) 发生在 1 帧的通信结束时, 和时钟同步发送时的 TXI 中断请求发生时序不同。

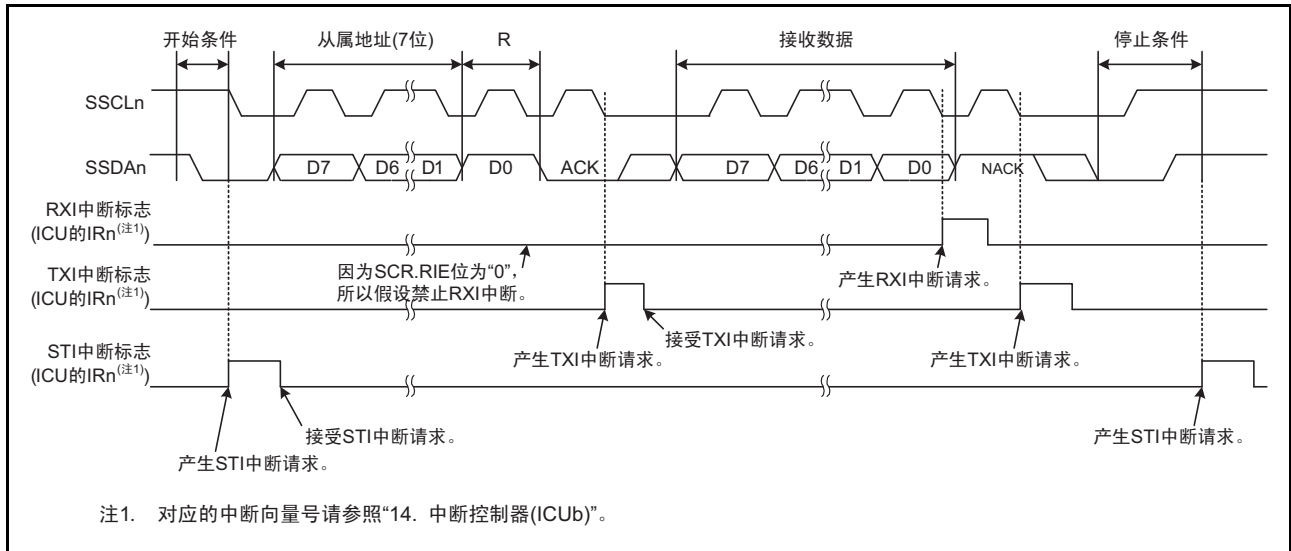


图 27.47 简易 I²C 总线模式的主控接收运行例子 (使用 7 位从属地址、发送中断和接收中断的情况)

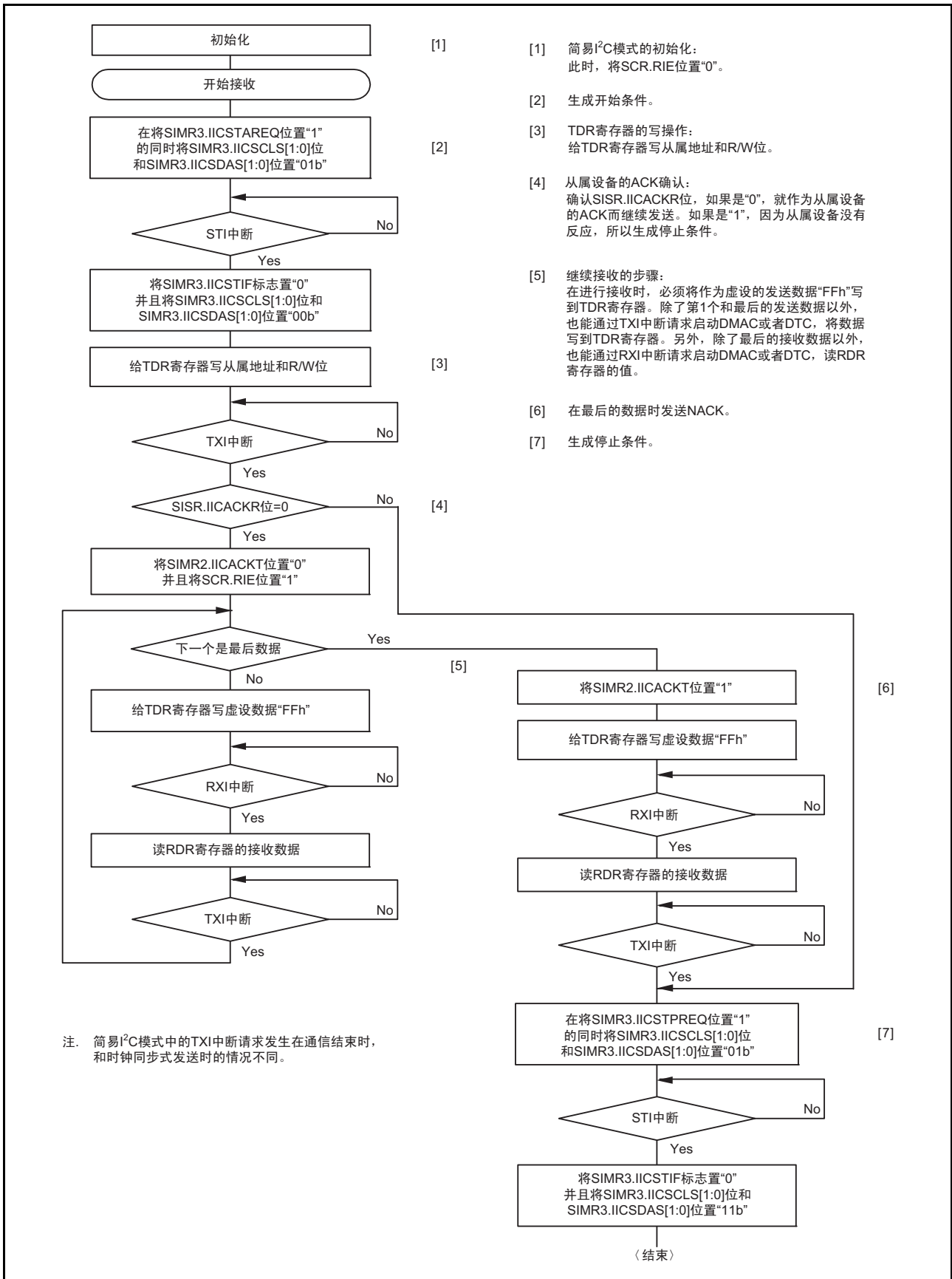


图 27.48 简易 I²C 模式的主控接收运行流程图例子 (使用发送中断和接收中断的情况)

27.8 简易 SPI 模式的运行

作为 SCI 的扩展功能，支持 1 个或者多个主控设备能对多个从属设备进行通信的简易 SPI 模式。

通过设定时钟同步模式 (SCMR.SMIF 位 =0、SIMR1.IICM 位 =0、SMR.CM 位 =1) 并且将 SPMR.SSE 位置“1”，进入简易 SPI 模式。在简易 SPI 模式用途中用于主控模式的单主控时，不需要主控侧的 SS 引脚功能，将 SPMR.SSE 位置“0”。

简易 SPI 模式的连接例子如图 27.49 所示。必须通过通用端口控制主控设备的 SS 信号的输出。

简易 SPI 模式和时钟同步模式一样，与时钟脉冲同步进行数据的发送和接收。通信数据的 1 个字符由 8 位数据构成，不能附加奇偶校验位。能通过将 SCMR.SINV 位置“1”，将发送 / 接收数据取反。

SCI 内部的发送部和接收部独立，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，所以能通过发送过程中写下一个发送数据并且在接收过程中读前一个接收数据，进行连续的发送和接收。

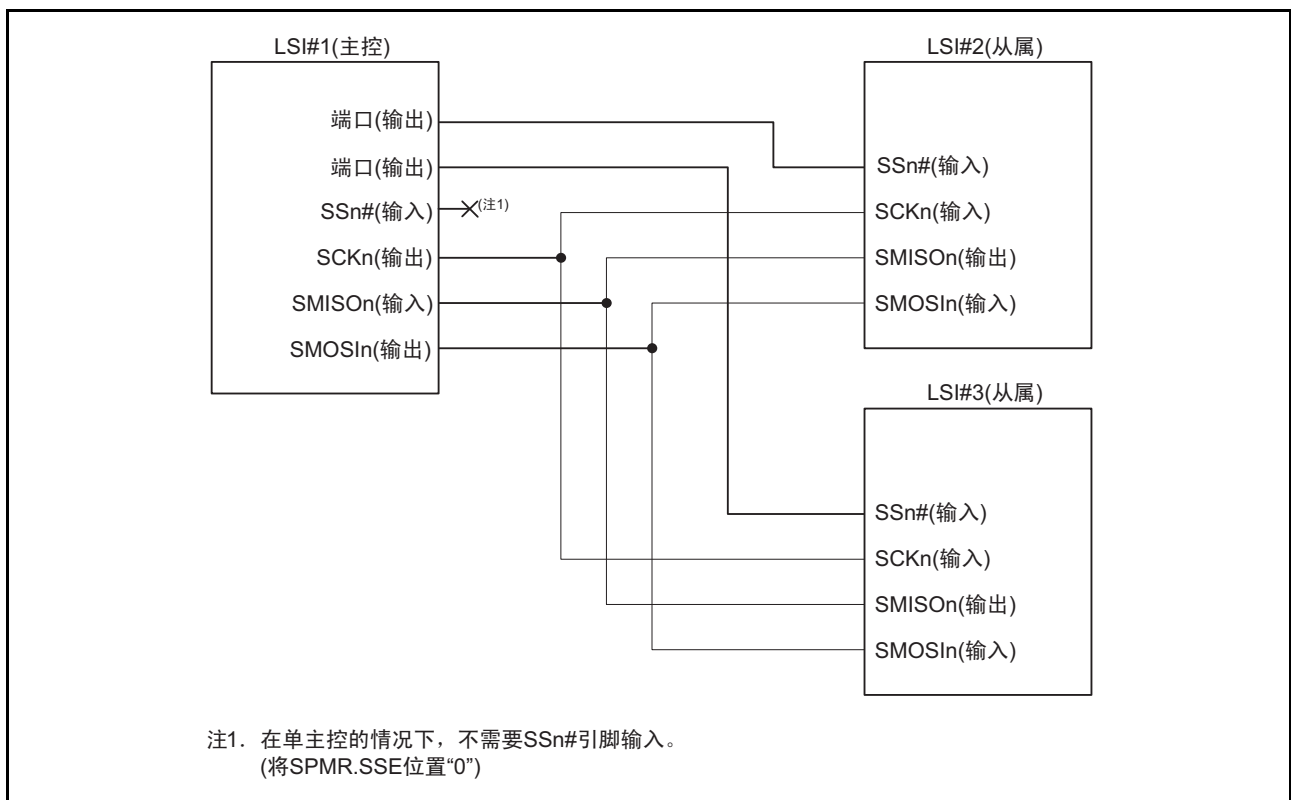


图 27.49 简易 SPI 模式的连接例子 (单主控 (SPMR.SSE 位 =0))

27.8.1 主控模式、从属模式和各引脚的状态

在简易 SPI 模式中，主控模式 (SCR.CKE[1:0] 位 =00 或者 01 并且 SPMR.MSS 位 =0) 和从属模式 (SCR.CKE[1:0] 位 =10 或者 11 并且 SPMR.MSS 位 =1) 中的各引脚的输入 / 输出方向不同。

模式、SSn# 引脚输入和各引脚状态的关系如表 27.24 所示。

表 27.24 模式、SSn# 引脚输入和各引脚状态的关系

模式	SSn# 引脚输入	SMOSIn 引脚状态	SMISOn 引脚状态	SCKn 引脚状态
主控模式 (注1)	High 电平 (能通信)	发送数据输出 (注2)	接收数据输入	时钟输出 (注3)
	Low 电平 (不能通信)	高阻抗	接收数据输入 (无效)	高阻抗
从属模式	High 电平 (不能通信)	接收数据输入 (无效)	高阻抗	时钟输入 (无效)
	Low 电平 (能通信)	接收数据输入	发送数据输出	时钟输入

注 1. 在单主控时 (SPMR.SSE 位 =0) 为可通信状态，与 SSn# 引脚的输入电平无关 (和 SSn# 引脚的输入为 High 电平时等效)。不使用 SSn# 引脚，而此引脚能用作其他用途。

注 2. 在禁止发送时 (SCR.TE 位 =0) 为高阻抗状态。

注 3. 在多主控 (SPMR.SSE 位 =1) 并且禁止发送和接收时 (SCR.TE 位 =00 并且 SCR.RE 位 =00) 为高阻抗状态。

27.8.2 主控模式中的 SS 功能

通过将 SCR.CKE[1:0] 位置 “00” 并且将 SPMR.MSS 位置 “0”，进入主控模式。

在单主控时 (SPMR.SSE 位 =0) 不使用 SSn# 引脚，与 SSn# 引脚输入的电平无关，能进行发送和接收。

在多主控 (SPMR.SSE 位 =1) 并且 SSn# 引脚的输入为 High 电平时，作为表示没有其他主控或者其他主控没有进行发送和接收的主控，从 SCKn 引脚输出时钟并且进行发送和接收。在多主控 (SPMR.SSE 位 =1) 并且 SSn# 引脚的输入为 Low 电平时，表示有其他主控并且正在进行发送和接收。此时，SCI 将 SMOSIn 引脚和 SCKn 引脚的输出置为高阻抗，不开始发送和接收。在发生模式故障错误时，SPMR.MFF 标志为 “1”。在多主控时，必须通过读 SPMR.MFF 标志来进行错误处理。即使在发送和接收过程中发生模式故障，也不停止发送和接收，而在发送和接收结束后，将 SMOSIn 引脚和 SCKn 引脚的输出置为高阻抗。

必须通过通用端口控制主控的 SS 信号输出。

27.8.3 从属模式中的 SS 功能

通过将 SCR.CKE[1:0] 位置 “10” 并且将 SPMR.MSS 位置 “1”，进入从属模式。

当 SSn# 引脚的输入为 High 电平时，RXDn 引脚的输出为高阻抗，忽视 SCKn 引脚的时钟输入；当 SSn# 引脚的输入为 Low 电平时，SCKn 引脚的时钟输入有效并且能进行发送和接收。

如果在发送 / 接收过程中，SSn# 引脚输入从 Low 电平变为 High 电平，就将 RXDn 引脚输出置为高阻抗，而且内部的发送 / 接收运行继续，直到根据 SCKn 引脚的时钟输入结束 1 个字符的发送 / 接收后停止运行，并且发生中断 (TXI、RXI、TEI 中的任意一个)。

27.8.4 时钟和发送 / 接收数据的关系

能通过 SPMR.CKPOL 位和 SPMR.CKPH 位从 4 种时钟中选择用于发送和接收的时钟。时钟和发送 / 接收数据的关系如图 27.50 所示。在从属模式和从属模式中，时钟和发送 / 接收数据的关系都相同 (和 SSn# 引脚的输入为 High 电平时等效)。SSn# 引脚能用于其他用途。详细内容请参照 “27.8.2 主控模式中的 SS 功能”。

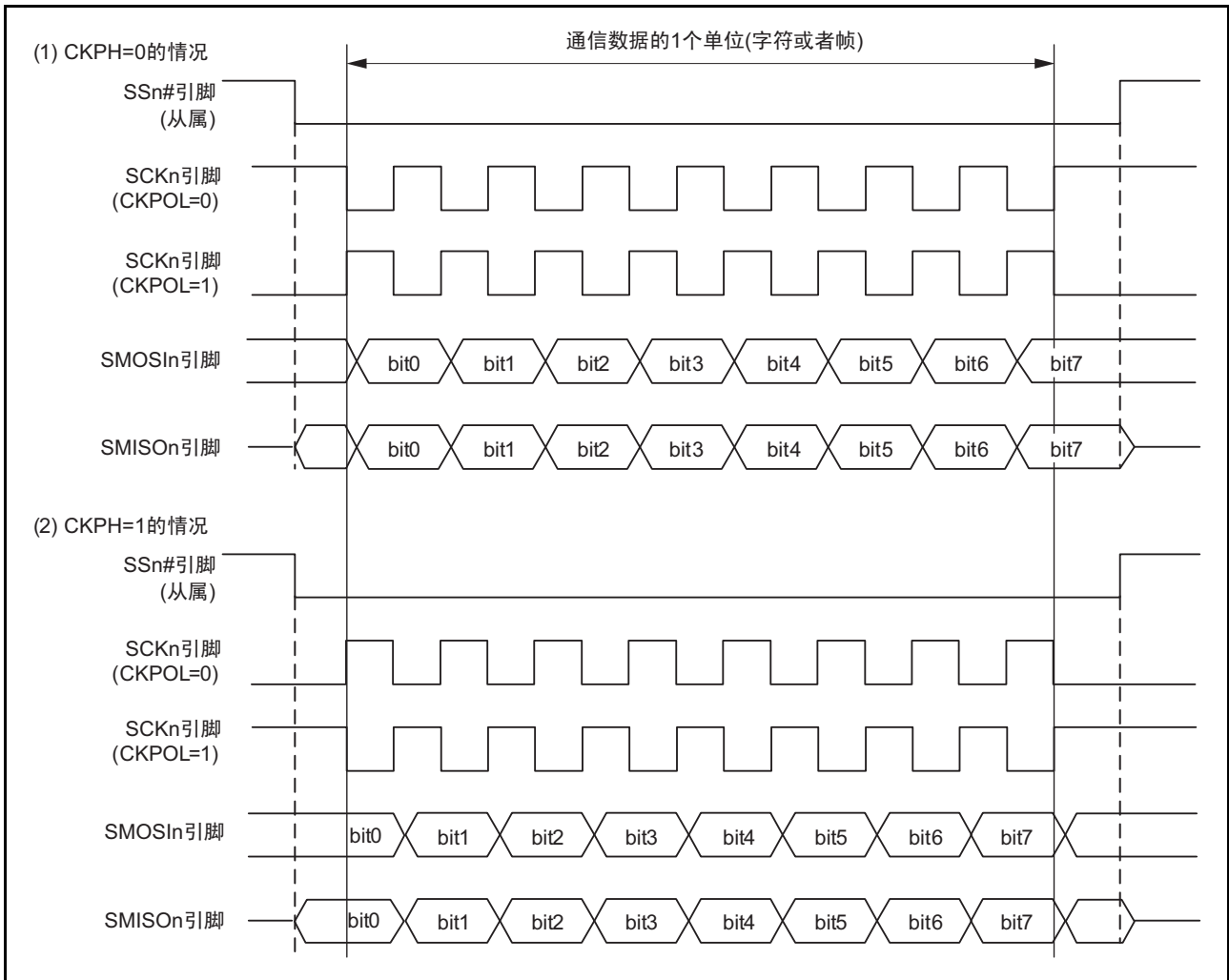


图 27.50 简易 SPI 模式的时钟和发送 / 接收数据的关系

27.8.5 SCI 的初始化 (简易 SPI 模式)

和时钟同步模式的初始化步骤 (图 27.19 的 SCI 的初始化流程图例子) 相同。SPMR.CKPOL 位和 SPMR.CKPH 位选择的时钟种类适合于主控设备和从属设备。

必须在将 SCR 寄存器置为初始值后进行初始化以及更改运行模式和通信格式。

必须注意: 即使将 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 TE 位从 “1” 置为 “0” 或者从 “0” 置为 “1”, 就在 SCR.TIE 位为 “1” 时产生 TXI 中断。

27.8.6 串行数据的发送和接收 (简易 SPI 模式)

在主机模式中, 如果在开始发送和接收前将发送和接收目标的从属设备的 SSn# 引脚置为 Low 电平并且结束发送和接收, 就将发送和接收目标的从属设备的 SSn# 引脚置为 High 电平。其他步骤和时钟同步模式相同。

27.9 扩展串行模式控制部的运行说明

27.9.1 串行通信协议

如图 27.51 所示，SCId 的扩展串行模式控制部和 SCIc 一起实现由 Start Frame 和 Information Frame 构成的串行通信协议。

Start Frame 由 Break Field、Control Field 0 和 Control Field 1 构成。Information Frame 能由几个 Data Field、CRC16 Upper Field 和 CRC16 Lower Field 构成。

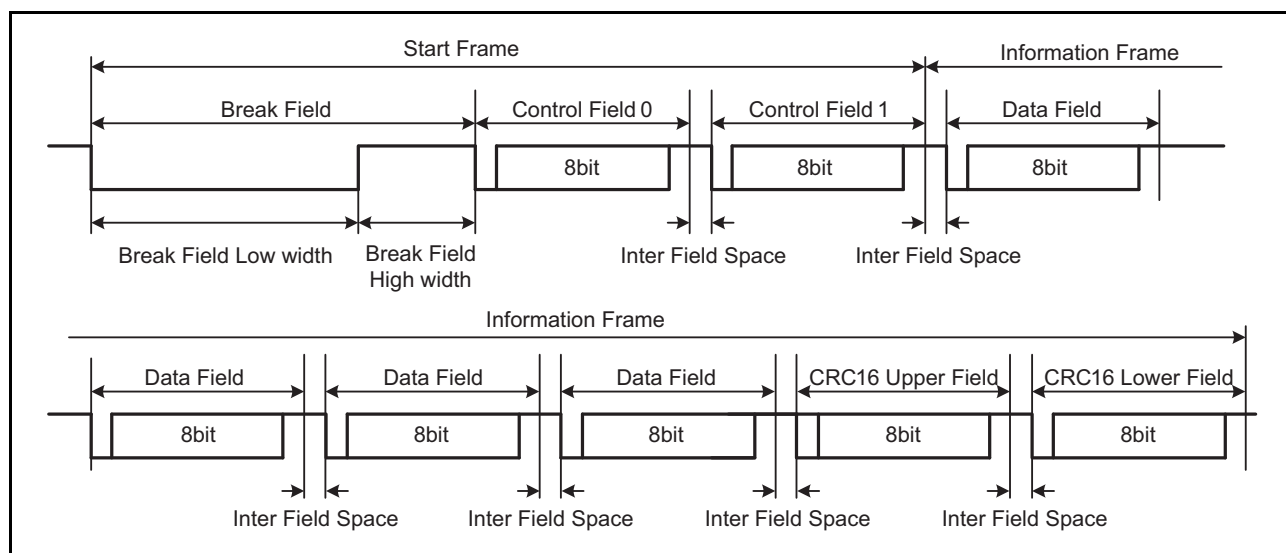


图 27.51 扩展串行模式控制部的串行通信协议

27.9.2 Start Frame 发送

发送由 Break Field Low width、Control Field 0 和 Control Filed1 构成的 Start Frame 时的运行例子如图 27.52 所示，发送 Start Frame 的流程图如图 27.53 和图 27.54 所示。

扩展串行模式控制部在 Start Frame 发送时的运行如下所示。SCI12 用于异步模式。

- (1) 在将定时器的运行模式设定为 Break Field Low width 输出模式的状态下，如果给 TCR 的 TCST 位写“1”，定时器就开始计数，在 TCNT 和 TPRE 所设期间从 TXDX12 引脚输出 Low 电平。
- (2) 如果定时器发生下溢，TXDX12 引脚就进行反相输出并且将 STR 的 BFDF 位置“1”。此时，如果 ICR 的 BFDIE 位为“1”，就产生 SCIX0 中断。
- (3) 给 TCR 的 TCST 位写“0”，停止定时器的计数，使用 SCI12 发送 Control Field 0 的数据。必须在从输出 Break Field Low width 后到发生下一次下溢前停止计数。
- (4) 在发送完 Control Field 0 的数据后，使用 SCI12 发送 Control Field 1 的数据。
- (5) 在发送完 Control Field 1 的数据后，使用 SCI12 进行 Information Frame 的通信。

必须根据 Start Frame 的结构省略 Break Field 和 Control Field 0。

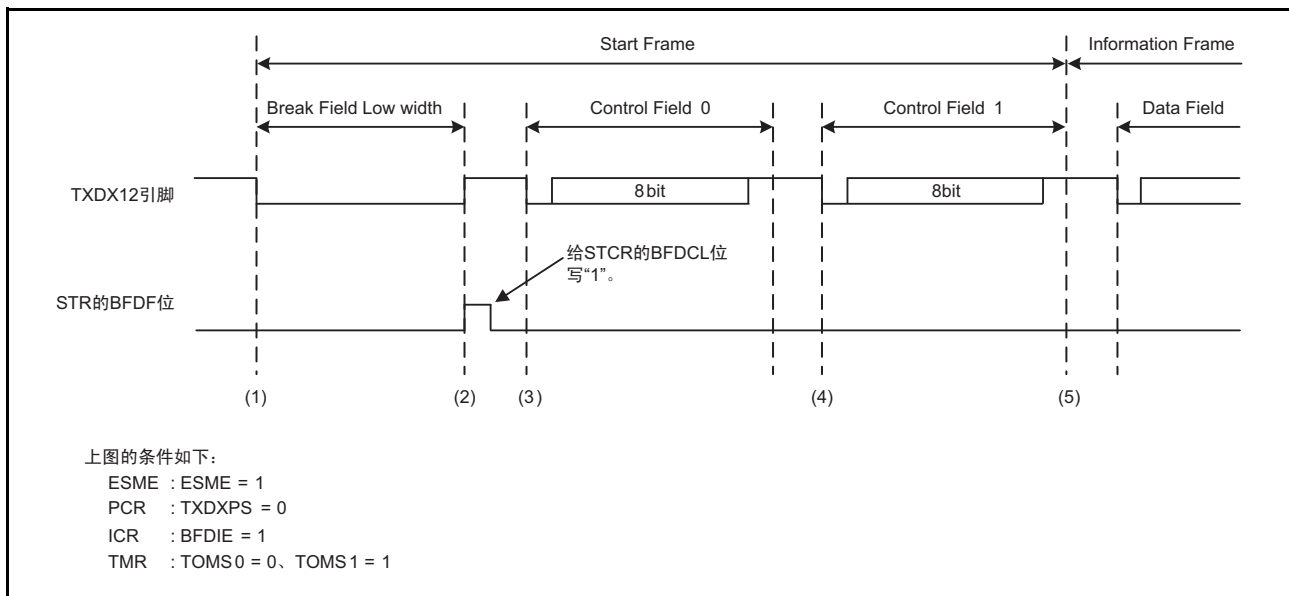


图 27.52 发送 Start Frame 时的运行例子

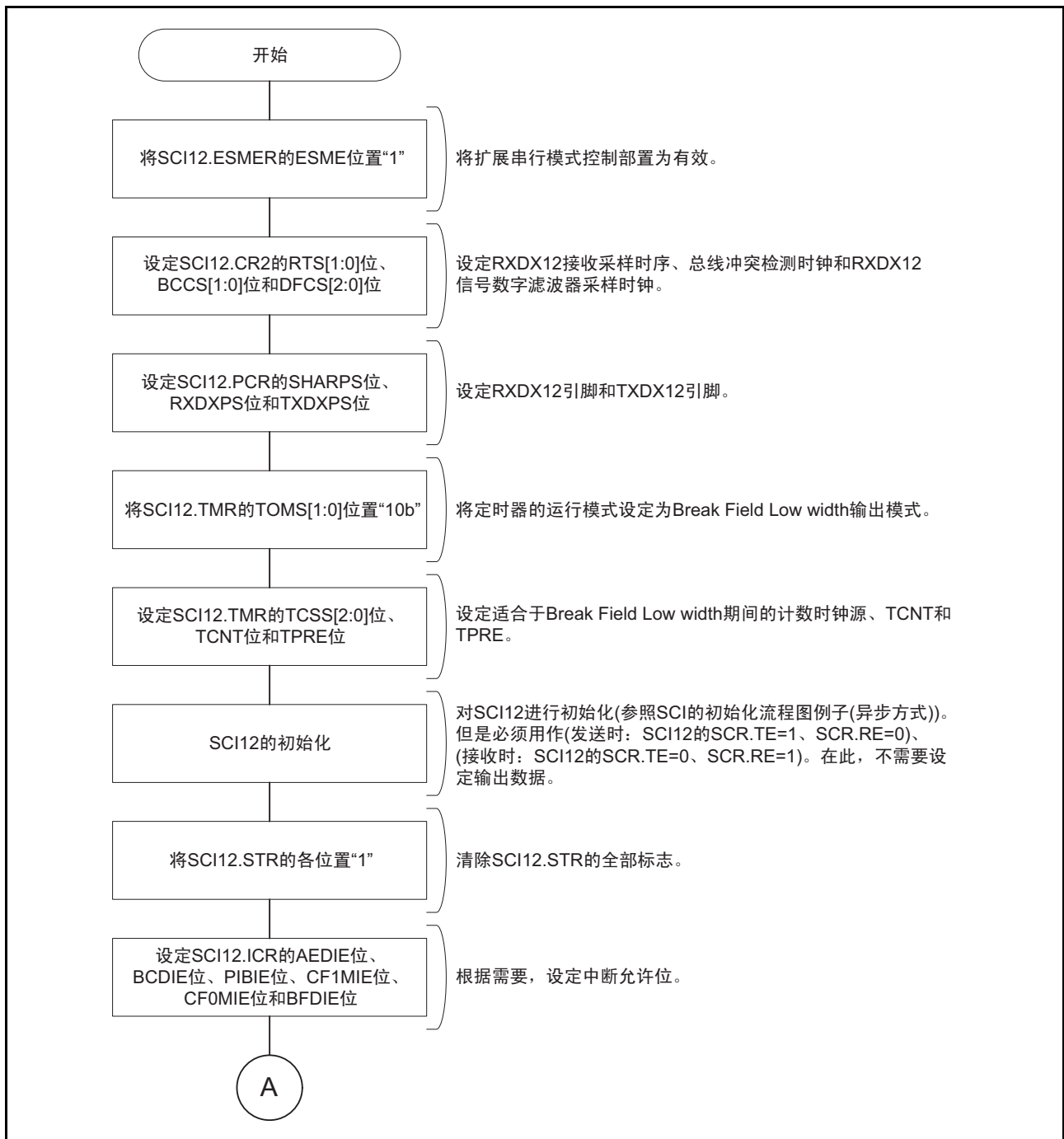


图 27.53 Start Frame 的发送流程图例子 (1)

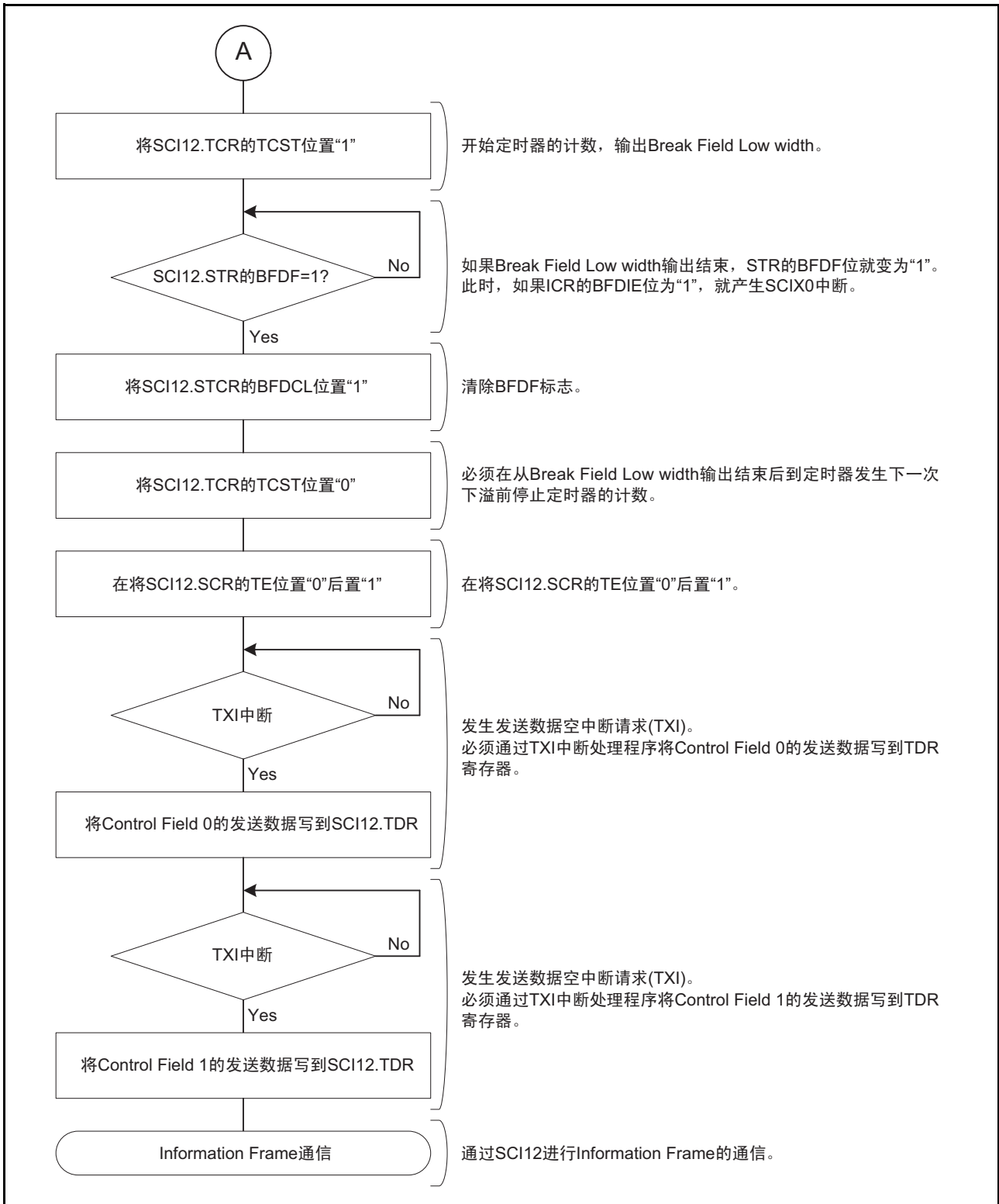


图 27.54 Start Frame 的发送流程图例子 (2)

27.9.3 Start Frame 接收

扩展串行模式控制部能检测到如表 27.25 所示结构的 Start Frame。

表 27.25 Start Frame 的结构

位的设定		Start Frame 的结构
BFE	CF0RE	
0	0	
0	1	
1	0	
1	1	

接收由 Break Field Low width、Control Field 0 和 Control Field 1 构成的 Start Frame 时的运行例子如图 27.55 所示。接收 Start Frame 的流程图如图 27.56 和图 27.57 所示，扩展串行模式控制部的状态转移图如图 27.58 所示。扩展串行模式控制部在接收 Start Frame 时的运行如下所示。SCI12 用于异步模式。

- (1) 如果在将定时器的运行模式设定为 Break Field Low width 检测模式后给 CR3 的 SDST 位写“1”，就能检测 Break Field Low width。此时，禁止将 RXDX12 输入到 SCI12。
- (2) 如果从 RXDX12 引脚至少输入定时器的 TCNT 和 TPRES 设定期间的 Low 电平，就检测为 Break Field Low width。此时，将 STR 的 BFDL 位置“1”，如果 ICR 的 BFDIE 位为“1”，就产生 SCIX0 中断。
- (3) 如果在检测到 Break Field Low width 后 RXDX12 引脚的输入为 High 电平，CR0 的 RXDSF 位就变为“0”，通过 SCI12 开始接收 Control Field 0。
- (4) 如果 Control Field 0 接收到的数据和 CF0DR 设定的数据相同，就将 STR 的 CF0MF 位置“1”。此时，如果 ICR 的 CF0MIE 位为“1”，就产生 SCIX1 中断，而且通过 SCI12 开始接收 Control Field 1。如果 Control Field 0 接收到的数据和 CF0DR 设定的数据不同，就转移到检测 Break Field Low width 前的状态。
- (5) 如果 Control Field 1 接收到的数据和 PCF1DR 或者 SCF1DR 设定的数据相同，就将 STR 的 CF1MF 位置“1”。此时，如果 ICR 的 CF1MIE 位为“1”，就产生 SCIX1 中断，而且使用 SCI12 进行 Information Frame 的通信。如果 Control Field 1 接收到的数据和 PCF1DR 或者 SCF1DR 设定的数据都不同，就转移到检测 Break Field Low width 前的状态。

必须根据 Start Frame 的结构省略 Break Field 和 Control Field 0 的处理。

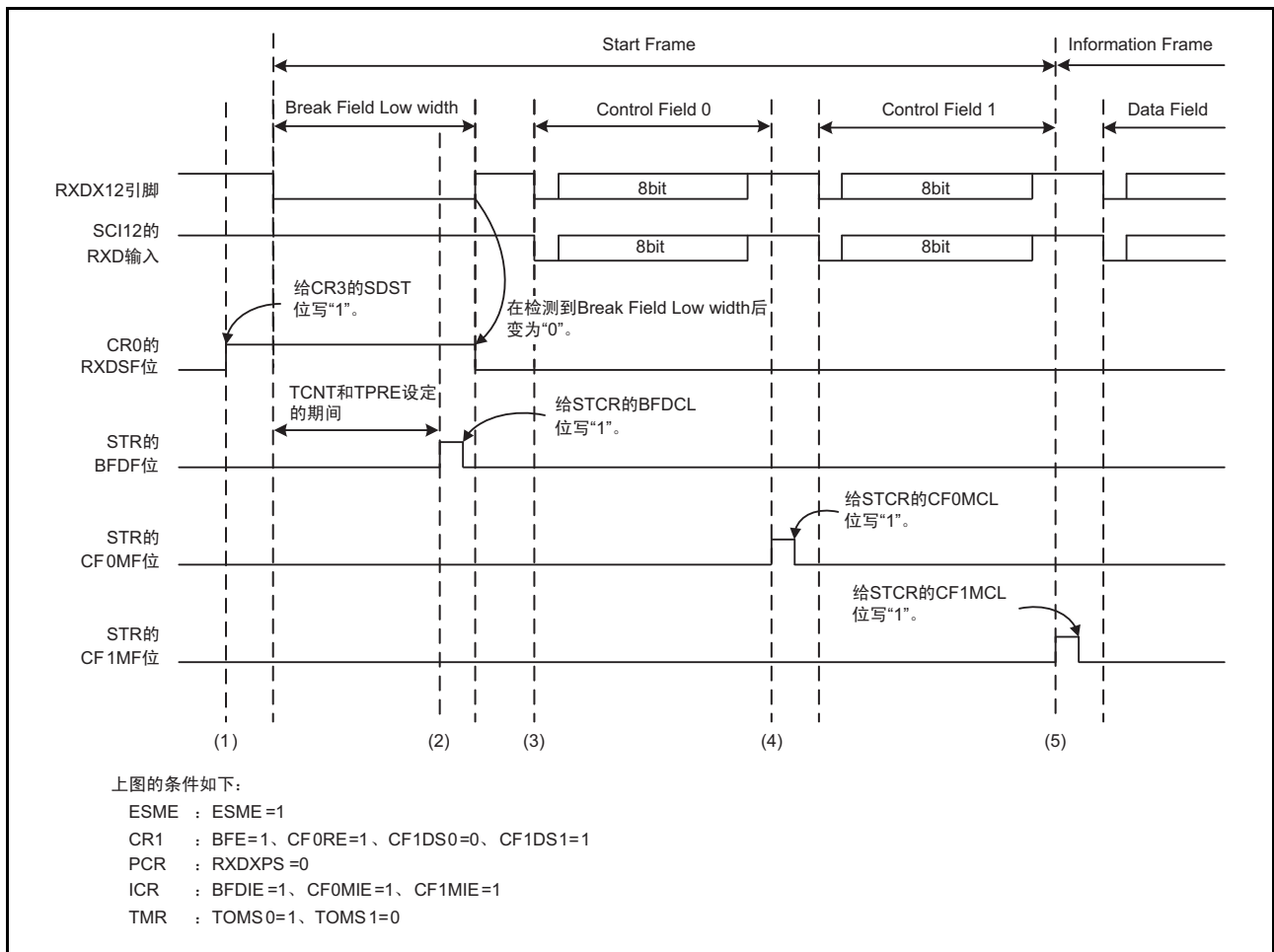


图 27.55 接收 Start Frame 时的运行例子

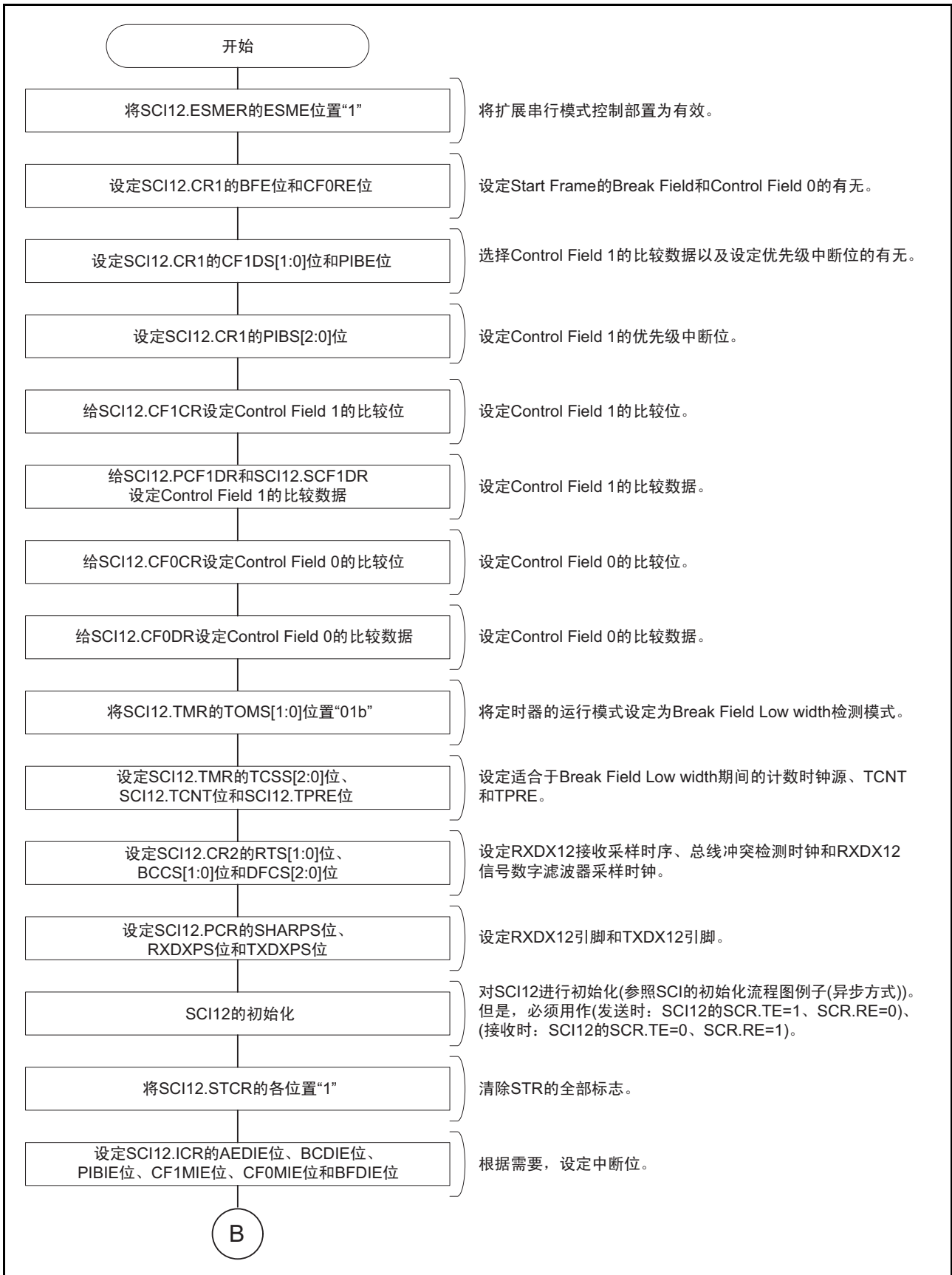


图 27.56 Start Frame 的接收流程图例子 (1)

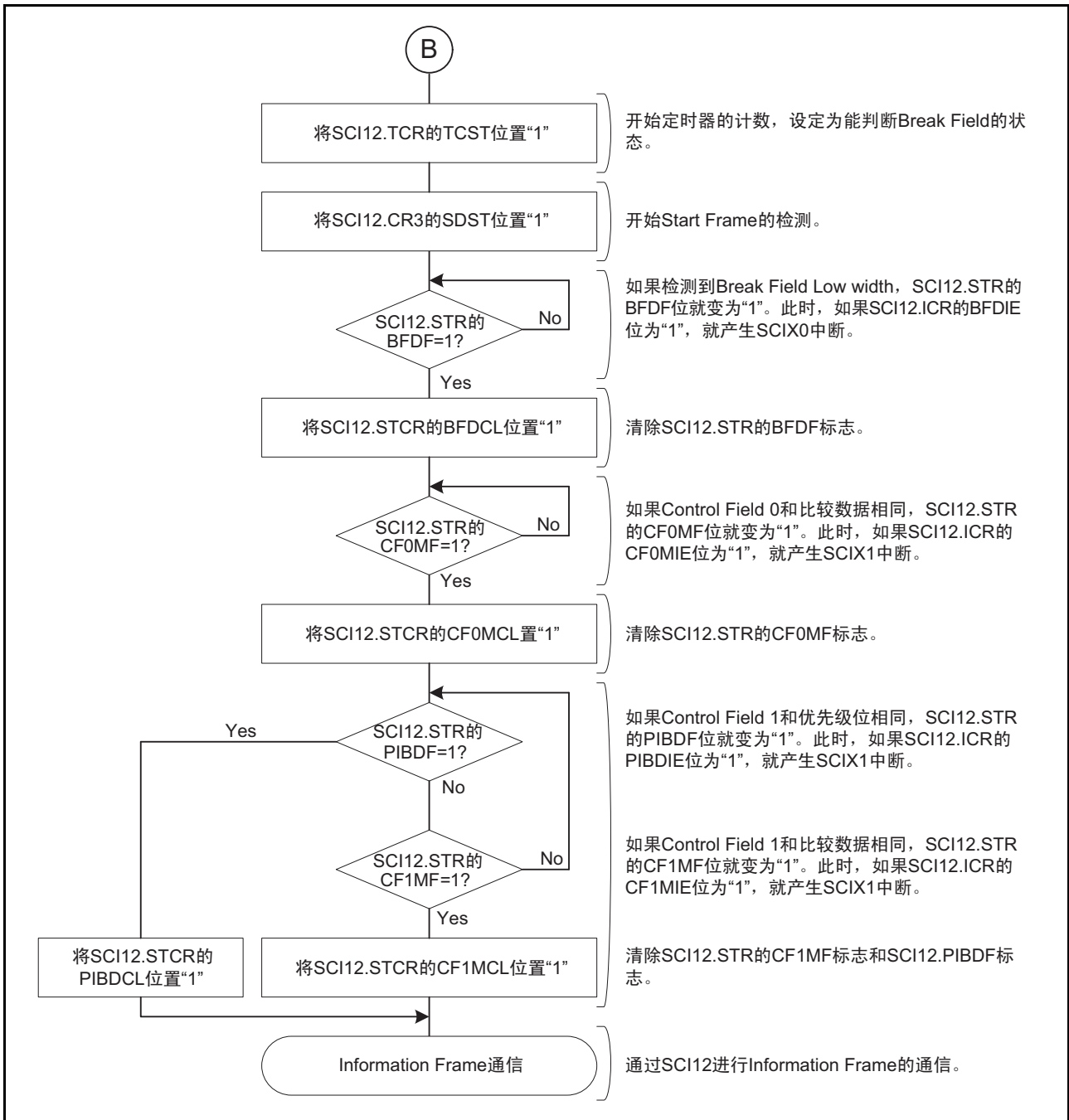


图 27.57 Start Frame 的接收流程图例子 (2)

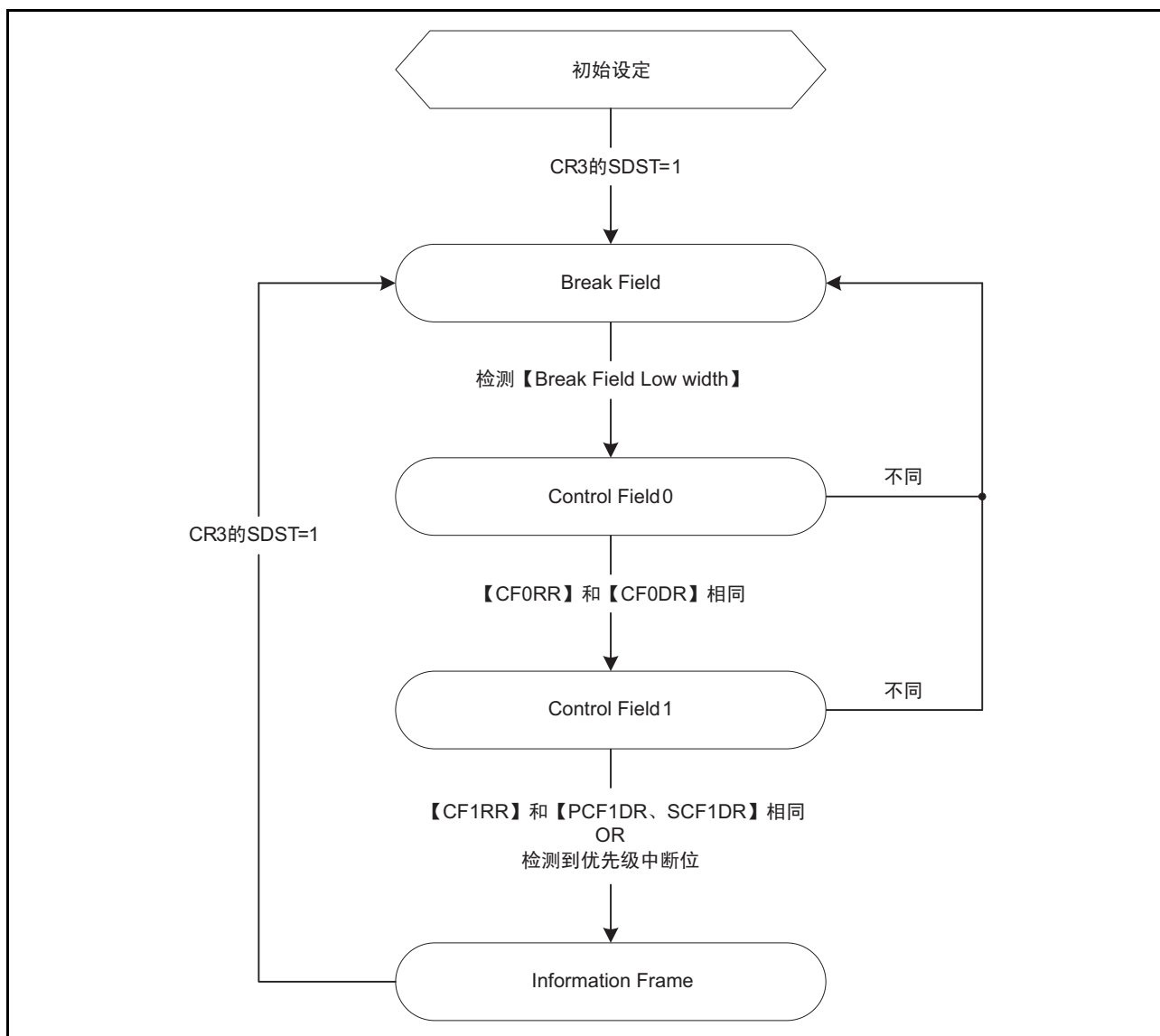


图 27.58 接收 Start Frame 时的状态转移图

27.9.3.1 优先级中断位

接收使用了优先级中断位的 Start Frame 时的运行例子如图 27.59 所示。通过将 CR1 的 PIBE 位置“1”，优先级中断位变为有效。

扩展串行模式控制部在接收使用了优先级中断位的 Start Frame 时的运行如下所示：

- (1) ~ (4) 和图 27.55 中的接收 Start Frame 时的运行例子 (1) ~ (4) 相同。
- (5) 如果 CR1 的 PIBS[2:0] 位指定的位数据和 PCF1DR 设定的数据相同，就将 STR 的 PIBDF 位置“1”。此时，如果 ICR 的 PIBDIE 位为“1”，就产生 SCIX1 中断，而且通过 SCI12 进行 Information Frame 的通信。如果 Control Field 1 接收到的数据和 PCF1DR 或者 SCF1DR 设定的数据都不同并且也检测不到优先级中断位，就转移到检测 Break Field Low width 前的状态。

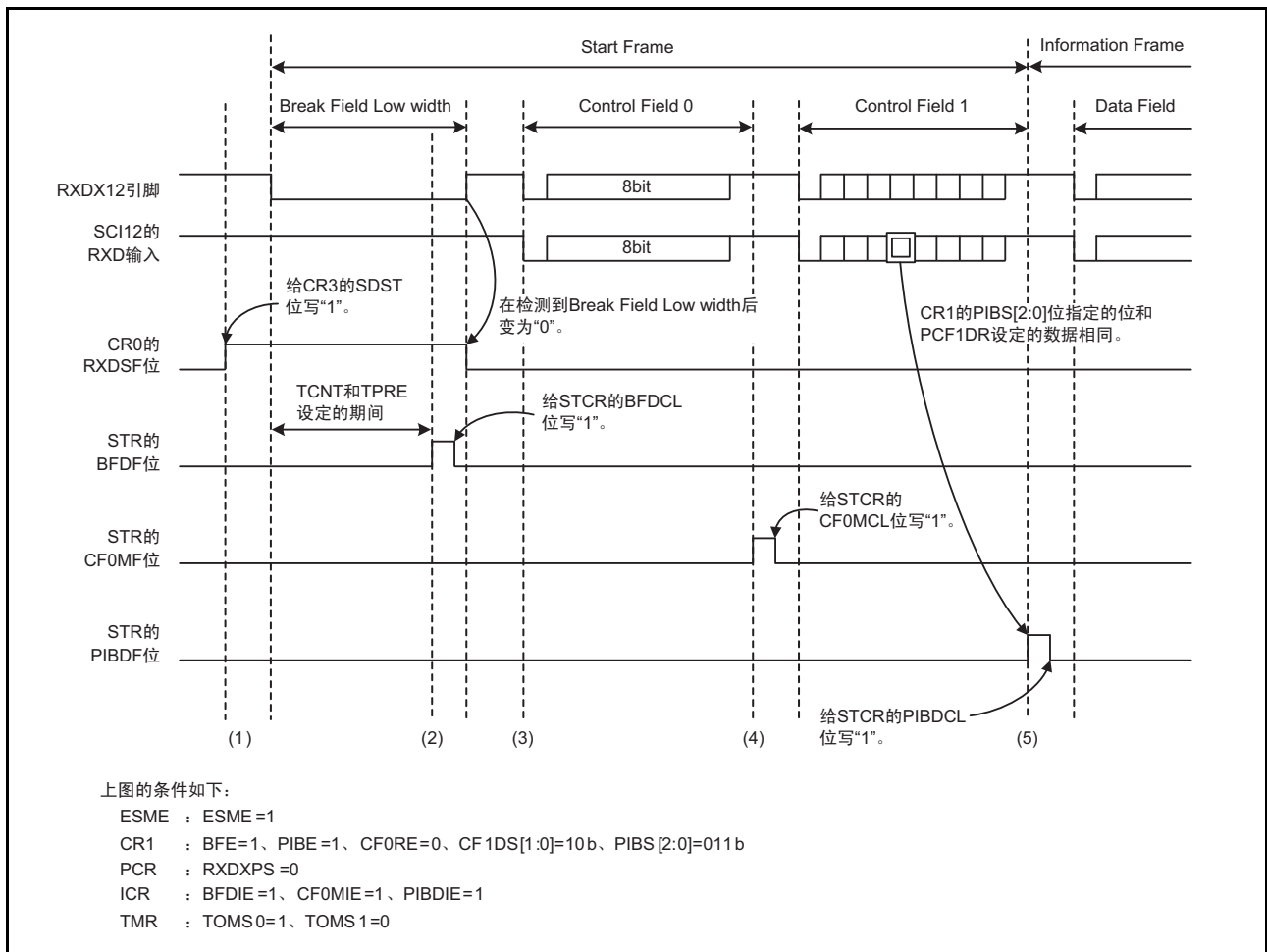


图 27.59 接收 Start Frame 时的运行例子（使用优先级中断位的情况）

27.9.4 总线冲突检测功能

在ESMER的ESME位和SCI12.SCR的TE位都为“1”的状态下输出Break Field Low width以及通过SCI12发送数据的过程中，总线冲突检测功能有效。

总线冲突检测功能的运行例子如图27.60所示。通过CR2的BCCS[1:0]所设总线冲突检测时钟对TXDX12引脚的输出和RXDX12引脚的输入进行采样，如果连续3次不同，STR的BCDF位就变为“1”。此时，如果ICR的BCDIE位为“1”，就产生SCIX2中断。

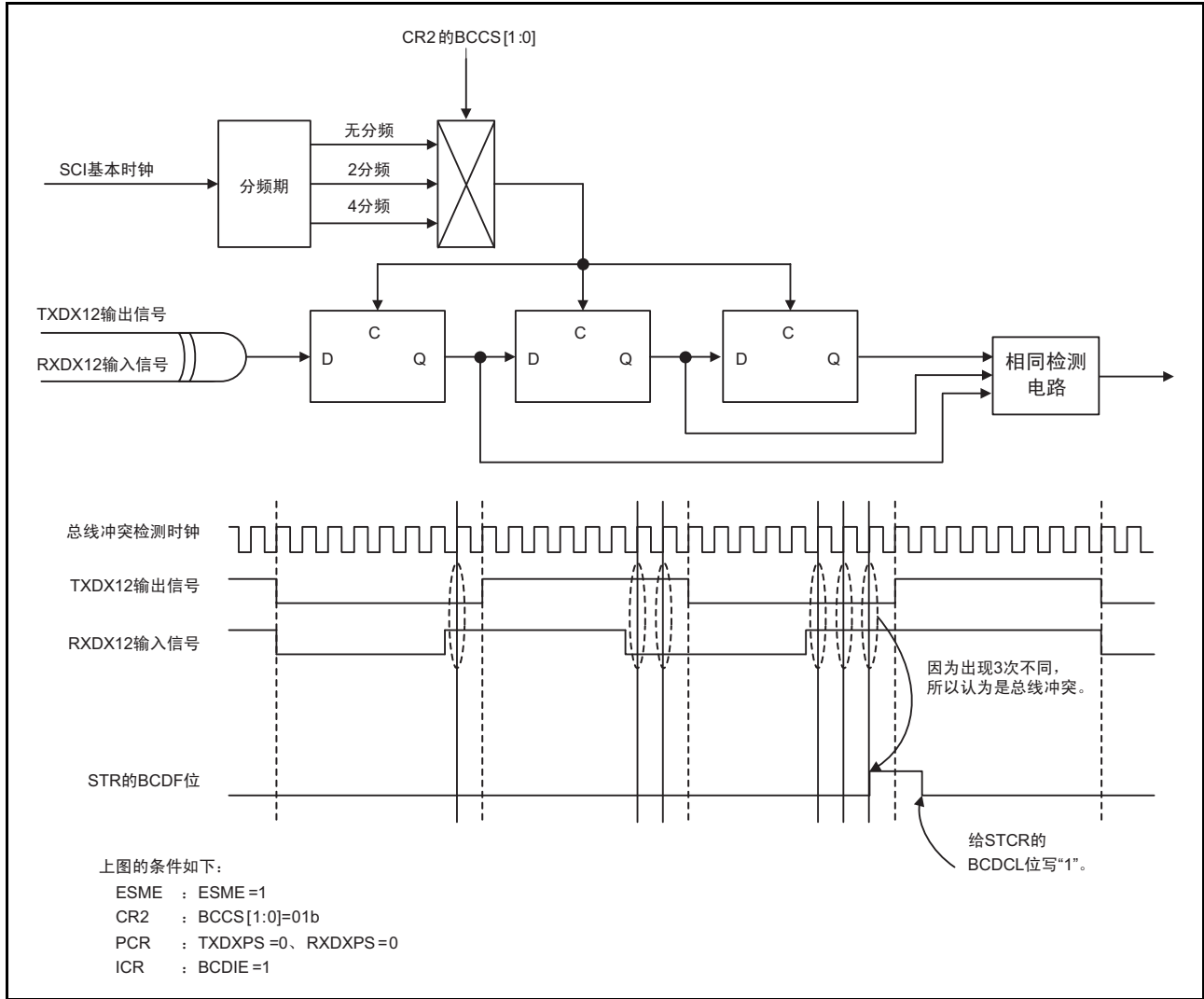


图 27.60 总线冲突检测功能的运行例

27.9.5 RXDX12 引脚输入的数字滤波器功能

能通过数字滤波器电路将 RXDX12 引脚的输入信号取到内部。数字滤波器电路由 3 段串联的触发电路和相同检测电路构成。通过 CR2 的 DFCS[2:0] 位所选时钟对 RXDX12 引脚的输入信号进行采样，如果 3 个锁存器的输出信号相同，就将此电平传送到后段；如果不相同，就保持以前的值。即，如果至少保持 3 个采样时钟相同的电平就识别为信号，但是如果低于 3 个采样时钟就将信号的变化判断为噪声，而不识别为信号变化。数字滤波器功能的运行例子如图 27.61 所示。

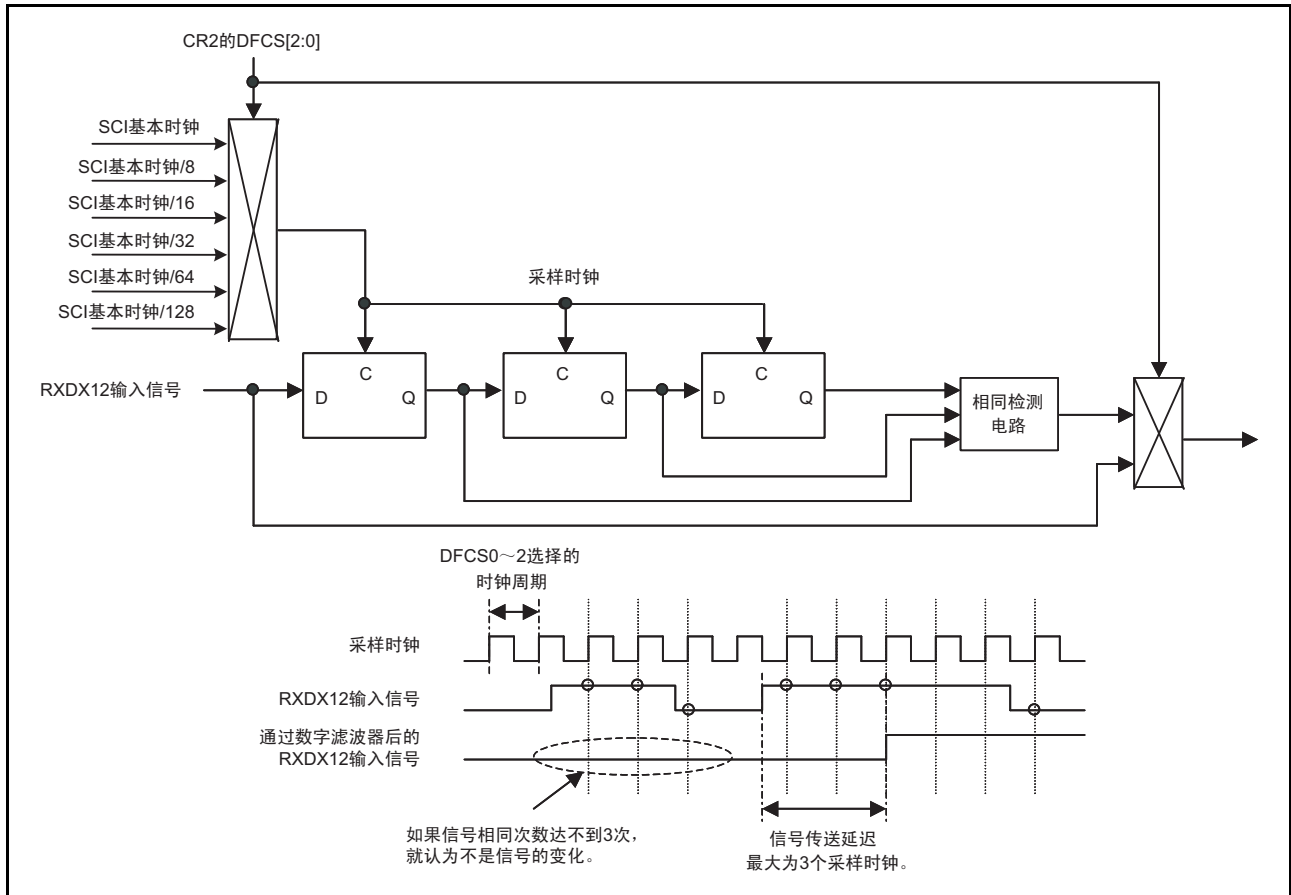


图 27.61 数字滤波器功能的运行例子

27.9.6 位速率测量功能

此功能是测量 RXDX12 引脚输入信号的上升沿和下降沿期间或者下降沿和上升沿期间的功能。位速率测量功能的运行例子如图 27.62 所示。

1. 如果给 CR0 的 BRME 位写“1”，位速率测量就有效。只有在要进行测量时才能将 BRME 位置“1”。即使将 BRME 位置“1”，也不在 Break Field 中测量位速率。
2. 在检测到 Break Field Low width 后，如果 RXDX12 引脚的输入变为 High 电平，就开始测量位速率。
3. 在开始测量位速率后，如果从 RXDX12 引脚输入有效边沿（上升沿和下降沿），定时器就将当时的计数值保持到读缓冲器并且重新加载计数器。此时，如果 ICR 的 AEDIE 位为“1”，就产生 SCIX3 中断。通过读 TCNT 和 TPRES 来解除保持状态。
4. 能从有效边沿期间的计数值计算出位速率，并且通过更改 SCI12 的设定来调整位速率。要在 Control Field 1 相同后将位速率测量功能置为无效时，必须给 CR0 的 BRME 位写“0”。

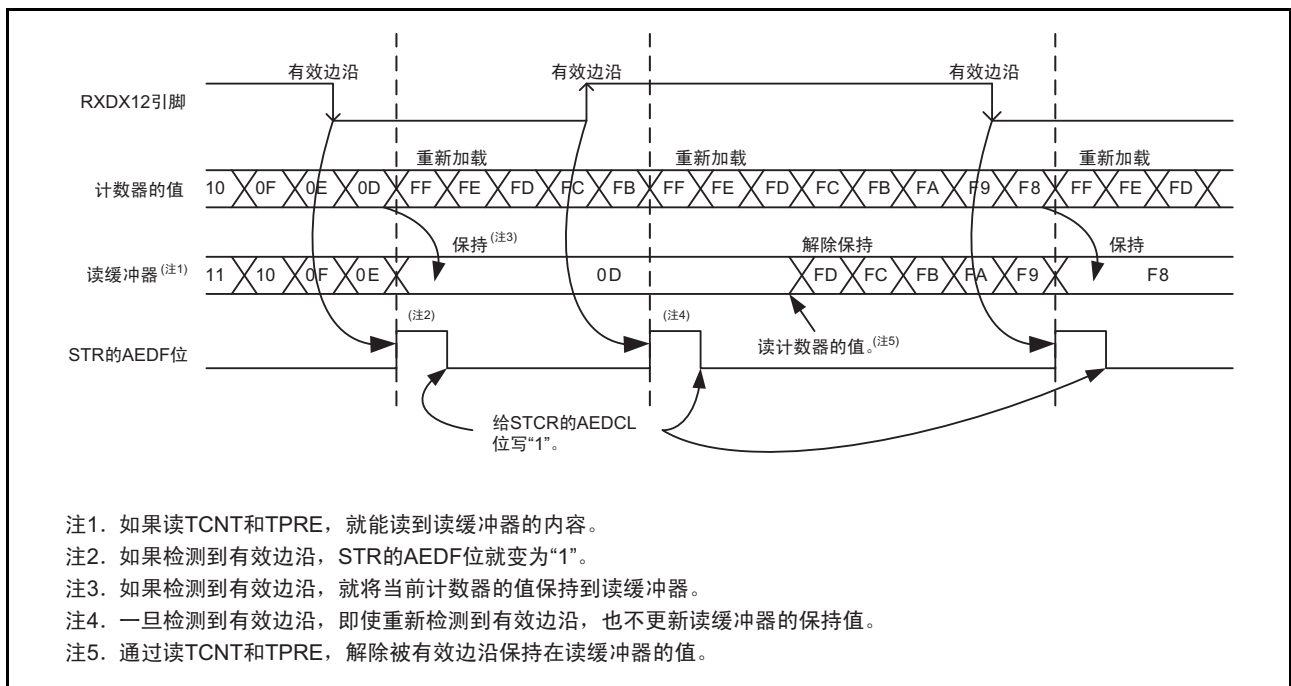


图 27.62 位速率测量功能的运行例子

27.9.7 RXDX12 接收数据的采样时序选择功能

扩展串行模式控制部能通过 CR2 的 RTS0 和 RTS1 从 SCI 基本时钟的 8 个时钟的上升沿、10 个时钟的上升沿、12 个时钟的上升沿或者 14 个时钟的上升沿中选择 SCI12 的 RXDX12 接收数据的采样时序。当 SCI12 的 SEMR 的 ABCS 位为“1”时，能从 PCLK 的 4 个时钟的上升沿、5 个时钟的上升沿、6 个时钟的上升沿和 7 个时钟的上升沿中选择接收数据的采样时序。RXDX12 接收数据的采样时序如图 27.63 所示。

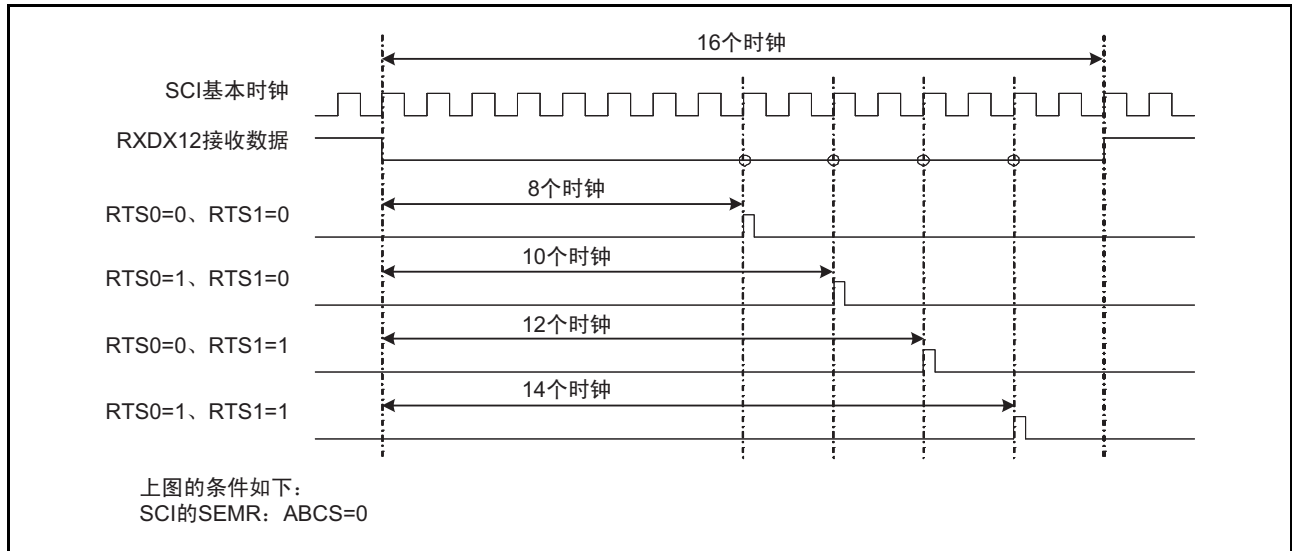


图 27.63 RXDX12 接收数据的采样时序

27.9.8 定时器

定时器有以下运行模式。

(1) Break Field Low width 输出模式

此模式在发送 Start Frame 时从 TXDX12 引脚输出 Break Field Low width 的 Low 电平。如果将 TMR 的 TOMS0 位置“0”并且将 TMR 的 TOMS1 位置“1”，就为 Break Field Low width 输出模式的运行。通过 TMR 的 TCSS[2:0] 位选择计数时钟源。如果给 TCR 的 TCST 位写“1”，就将 TXDX12 引脚的输出置为 Low 电平并且开始计数。如果定时器发生下溢，就将 TXDX12 引脚的输出置为 High 电平并且 STR 的 BFDF 位变为“1”。此时，如果 ICR 的 BFDIE 位为“1”，就产生 SCIX0 中断。如果给 TCR 的 TCST 位写“0”，就在重新加载 TPRES 和 TCNT 后停止计数。必须在 Break Field Low width 输出结束后到定时器再次发生下溢前停止计数。Break Field Low width 输出模式的运行例子如图 27.64 所示。

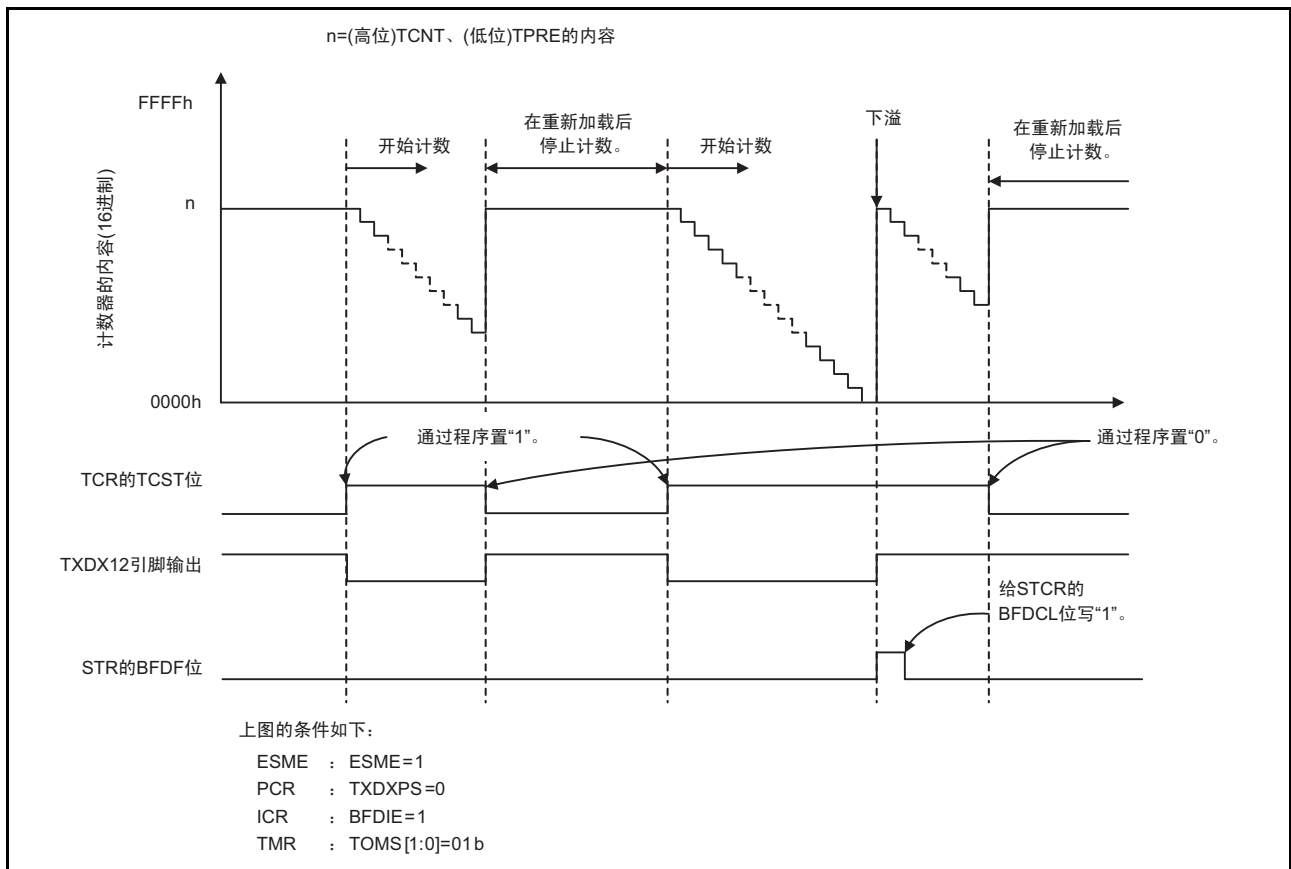


图 27.64 Break Field Low width 输出模式的运行例子

(2) Break Field Low width 判定模式

此模式在接收 Start Frame 时判定从 RXDX12 引脚输入的 Break Field Low width。如果将 TMR 的 TOMS0 位置“1”并且将 TMR 的 TOMS1 位置“0”，就为 Break Field Low width 判定模式的运行。通过 TMR 的 TCSS[2:0] 位选择计数时钟源。如果给 TCR 的 TCST 位写“1”，就进入能判定 Break Field Low width 的状态。如果从 RXDX12 引脚输入 Low 电平，就开始判定；如果从 RXDX12 引脚输入 High 电平，就重新加载 TPRES 和 TCNT 并且进入能判定 Break Field Low width 的状态。如果在 Break Field Low width 的判定过程中定时器发生下溢，STR 的 BDFD 位就变为“1”。此时，如果 ICR 的 BFDIE 位为“1”，就产生 SCIX0 中断。如果在数据通信过程中不能让定时器发生下溢或者发生中断，就必须在判定 Break Field Low width 后停止定时器。Break Field Low width 判定模式的运行例子如图 27.65 所示。

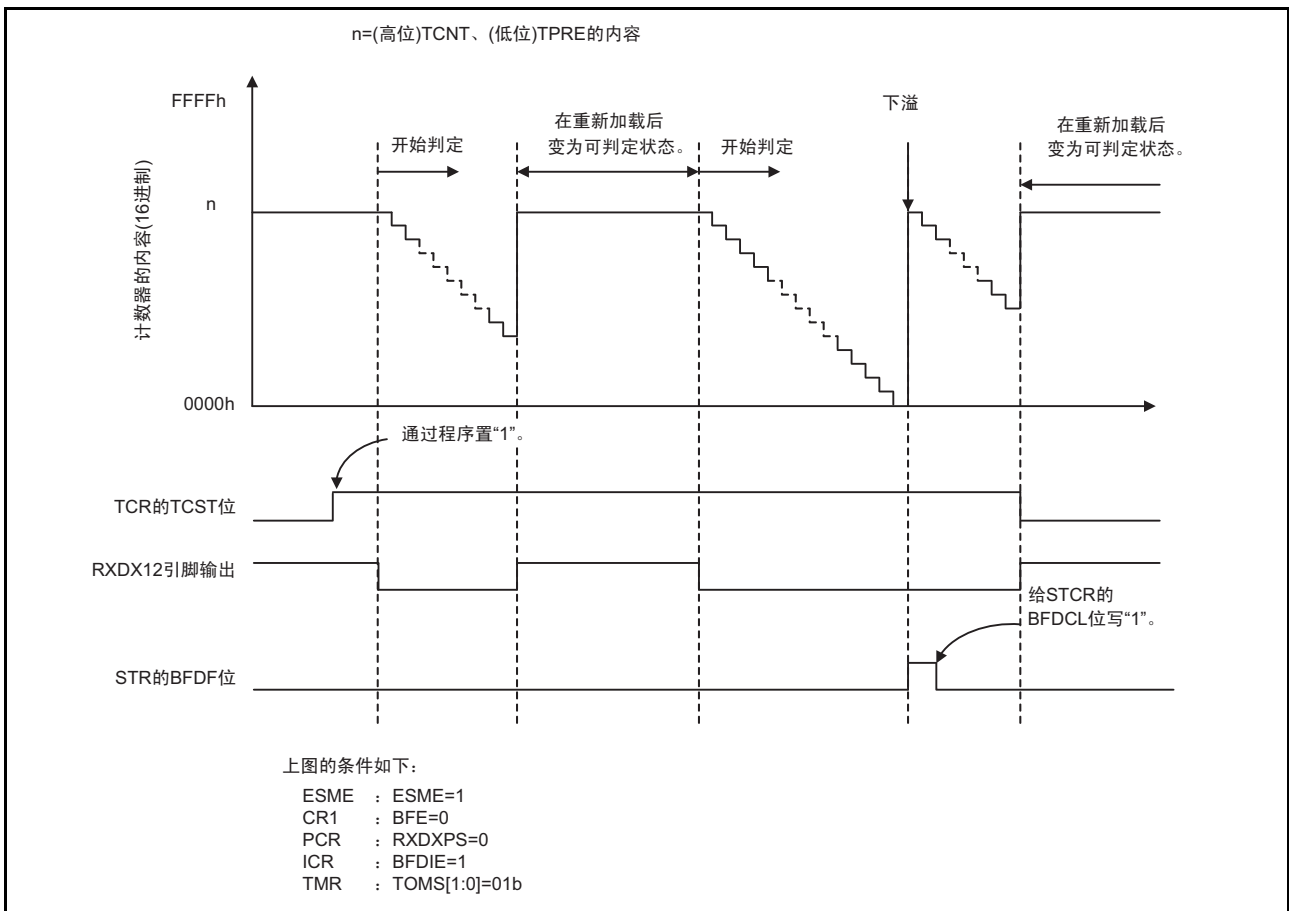


图 27.65 Break Field Low width 判定模式的运行例子

(3) 定时器模式

在此模式中将内部时钟作为计数时钟源。如果将 TMR 的 TOMS0 位和 TMR 的 TOMS1 位都置“0”，就为定时器模式运行。通过 TMR 的 TCSS[2:0] 位选择计数时钟源。如果给 TCR 的 TCST 位写“1”，就开始计数；如果给 TCST 位写“0”，就停止计数。通过输入到 TPRES 的计数时钟源的周期，TPRES 进行递减计数。将 TPRES 的下溢置为计数时钟源，TCNT 进行递减计数。如果定时器发生下溢，STR 的 BDFD 位就变为“1”。此时，如果 ICR 的 BFDIE 位为“1”，就产生 SCIX0 中断。

27.10 噪声消除功能

用于噪声消除功能的噪声滤波器的结构如图 27.66 所示。噪声滤波器由 2 段触发电路和相同检测电路构成。根据设定的采样周期，如果引脚的采样电平 3 次相同，就将相同的电平传送到内部，并且相同电平被持续传送到内部，直到引脚的采样电平再次 3 次相同为止。

在异步模式中，能将噪声消除功能用于 RXDn 的输入信号。采样周期为基本时钟的周期（在 SEMR.ABCS 位为“0”时，为 1 位时间的 1/16 周期；在 SEMR.ABCS 位为“1”时，为 1 位时间的 1/8 周期）。

在简易 I²C 模式中，能将噪声消除功能用于 SSDAn 和 SSCLn 的输入信号。通过设定 SNFR.NFCS[2:0] 位，从内部波特率发生器的时钟源的 1/2/4/8 分频时钟中选择采样时钟。

如果在噪声滤波器有效的状态下停止基本时钟，就在基本时钟重新开始输入时从停止时的噪声滤波器状态开始运行。如果在输入基本时钟期间将 SCR.TE 位和 SCR.RE 位都置“0”，噪声滤波器的触发器就全部被初始化为“1”，当重新开始接收时的输入数据为“1”时，作为检测到相同信号而被传送到内部；当输入数据为“0”时，噪声滤波器的输出保持初始值，直到引脚的采样电平 3 次相同为止。

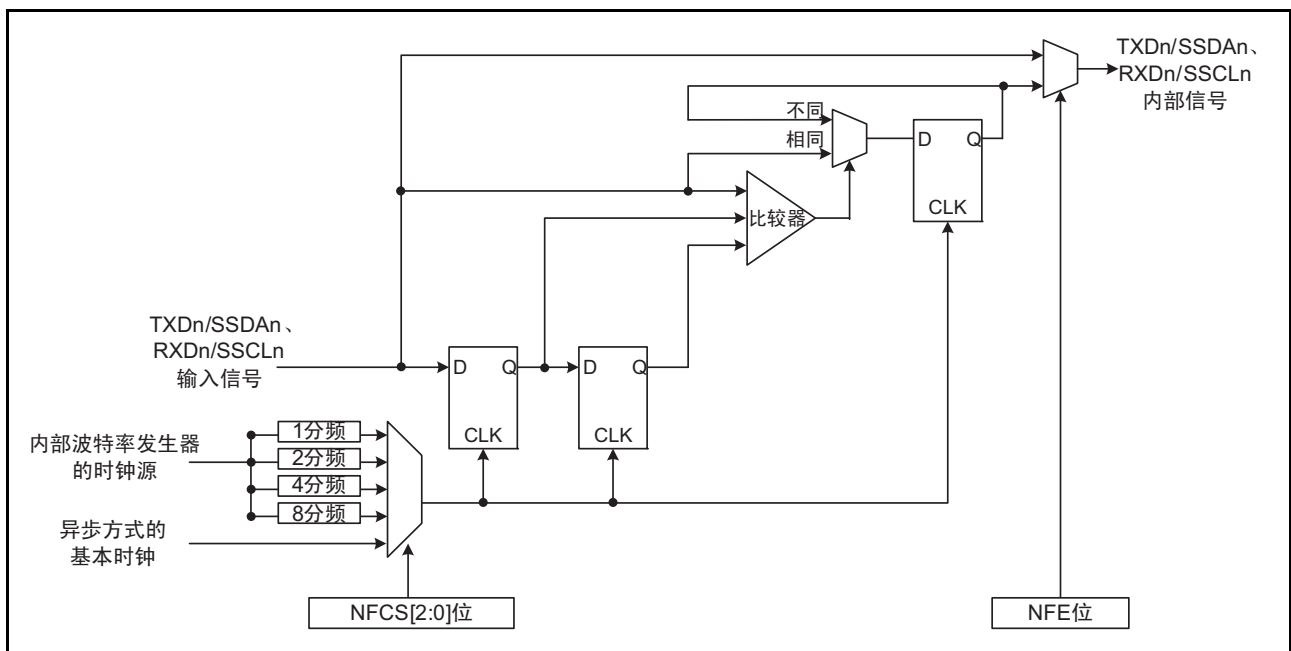


图 27.66 数字噪声滤波器电路的框图

27.11 中断源

27.11.1 TXI 中断和 RXI 中断的缓冲运行

对于 TXI 中断和 RXI 中断，即使在中断控制器的中断状态标志为“1”时满足中断产生条件，也不将中断请求输出到中断控制器而保持在内部（在内部能保持的容量是每个中断源 1 个请求）。

如果中断控制器的中断状态标志为“0”，就将保持的中断请求输出到中断控制器。如果输出被保持的中断请求，就自动清除此中断的内部保持。也能通过将对应的中断允许位（SCR.TIE 位或者 SCR.RIE 位）置“0”来清除保持在内部的中断请求。

27.11.2 串行通信接口模式和简易 SPI 模式的中断

串行通信接口模式和简易 SPI 模式的中断源如表 27.26 所示。各中断源分配有不同的中断向量，能通过 SCR 寄存器的允许位独立允许各中断源。

如果在 SCR.TIE 位为“1”时将发送数据从 TDR 寄存器传送到 TSR 寄存器，就产生 TXI 中断请求。通过在将 SCR.TIE 位置“1”后将 SCR.TE 位置“1”，或者用 1 条指令同时将 SCR.TIE 位和 SCR.TE 位置“1”，也产生 TXI 中断请求。能在通过 TXI 中断请求启动 DTC 或者 DMAC 后进行数据传送。

如果在 SCR.TIE 位为“0”的状态下将 SCR.TE 位置“1”以及在 SCR.TE 位为“1”的状态下将 SCR.TIE 位置“1”，就不产生 TXI 中断请求。（注 1）

在 SCR.TEIE 位为“1”时，如果在发送数据的最后位之前不给 TDR 寄存器写下一个数据，SSR.TEND 标志就变为“1”并且产生 TEI 中断请求。另外，在将 SCR.TE 位置“1”后到给 TDR 寄存器写发送数据为止的期间，SSR.TEND 标志保持“1”的状态，如果将 SCR.TEIE 位置“1”，就产生 TEI 中断请求。

如果给 TDR 寄存器写数据，就清除 SSR.TEND 标志并且取消 TEI 中断请求，但是需要取消时间。

如果在 SCR.RIE 位为“1”时将接受数据保存到 RDR 寄存器，就产生 RXI 中断请求。能在通过 RXI 中断请求启动 DTC 或者 DMAC 后进行数据传送。

如果在 SCR.RIE 位为“1”时将 SSR.ORER 标志、SSR.FER 标志或者 SSR.PER 标志置“1”，就产生 ERI 中断请求。此时，不产生 RXI 中断请求。能通过清除 SSR.ORER、SSR.FER 和 SSR.PER 的全部位，取消 ERI 中断请求。

注 1. 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发送数据时，必须通过 TXI 中断对应的中断控制器的中断请求允许位而不是 SCR.TIE 位，控制中断的禁止和允许。从而能防止 TXI 中断请求（用于发送新数据）的抑止。

表 27.26 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（注 1）	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	

注 1. FER 标志和 PER 标志只在异步模式中有效。

27.11.3 智能卡接口模式的中断

智能卡接口模式的中断源如表 27.27 所示，不能使用发送结束中断（TEI）请求。

表 27.27 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。如果在发送时将 SSR.TEND 标志置“1”，就产生 TXI 中断请求。如果预先将 TXI 中断请求设定为 DTC 或者 DMAC 的启动源，就在通过 TXI 中断请求启动 DTC 或者 DMAC 后传送发送数据。在通过 DTC 或者 DMAC 进行数据传送时，将 TEND 标志自动置“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，不自动清除 SSR.ERS 标志，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，清除 ERS 标志。

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。有关 DTC 或者 DMAC 的设定方法，请参照“17. DMA 控制器（DMACA）”和“18. 数据传送控制器（DTCa）”。

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 中断请求设定为 DTC 或者 DMAC 的启动源，就在通过 RXI 中断请求启动 DTC 或者 DMAC 后传送接收数据。如果发生错误，就将错误标志置位。因此，不启动 DTC 或者 DMAC 而向 CPU 请求 ERI 中断，所以必须清除错误标志。

27.11.4 简易 I²C 模式的中断

在简易 I²C 模式中，有表 27.28 所示的中断源。STI 中断分配给发送结束中断 (TEI) 请求，不使用接收错误中断 (ERI) 请求。

简易 I²C 模式也能使用 DTC 或者 DMAC 进行发送和接收。

当 SIMR2.IICINTM 位为“1”时，在 SSCLn 引脚的第 8 个下降沿产生 RXI 中断请求。如果预先将 RXI 中断请求设定为 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。另外，在 SSCLn 引脚的第 9 个下降沿 (应答位) 产生 TXI 中断请求。如果预先将 TXI 中断请求设定为 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。

当 SIMR2.IICINTM 位为“0”时，在 SSCLn 引脚的第 9 个下降沿 (应答位) SSDAn 引脚的输入为 Low 电平时，产生 RXI 中断请求 (检测到 ACK)；在 SSDAn 引脚的输入为 High 电平时，产生 TXI 中断请求 (检测到 NACK)。如果预先将 RXI 中断请求设定为 DTC 或者 DMAC 的启动源，就能在通过 RXI 中断请求启动 DTC 或者 DMAC 后进行接收数据或者发送数据的传送。

要使用 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设定 SCI。

如果使用 SIMR3.IICSTAREQ、SIMR3.IICRSTAREQ 和 SIMR3.IICSTPREQ 的各位生成开始条件、重新开始条件和停止条件，就在生成结束时产生 STI 中断请求。

表 27.28 SCI 中断源

名称	中断源	中断标志	DTC 的启动	DMAC 的启动	优先级
RXI	接收、ACK 检测	—	能	能	高 ↑ 低
TXI	发送、NACK 检测	—	能 (注 1)	能 (注 1)	
STI	开始条件、重新开始条件、 停止条件生成结束	IICSTIF	不能	不能	

注 1. 只有在 SIMR2.IICINTM 位为“1” (选择接收中断和发送中断) 时才能启动 DTC 和 DMAC。

27.11.5 扩展串行模式控制部的中断请求

SCId 的扩展串行模式控制部生成的中断请求有 SCIX0 中断 (检测到 Break Field Low width)、SCIX1 中断 (Control Field 0 相同, Control Field 1 相同, 检测到优先级中断位)、SCIX2 中断 (检测到总线冲突) 和 SCIX3 中断 (检测到有效边沿) 共 6 种。如果发生各中断源，就将状态标志置“1”。各中断请求的内容如表 27.29 所示。

表 27.29 扩展串行模式控制部的中断请求

中断请求	状态标志	中断源
SCIX0 中断 (检测到 Break Field Low width)	BDFD	<ul style="list-style-type: none"> 当检测到长于定时器所设期间的 Break Field Low width 时 在定时器所设期间 Break Field Low width 输出结束时 当定时器发生下溢时
SCIX1 中断 (Control Field 0 相同)	CF0MF	当 Control Field 0 的接收数据和 CF0DR 中设定的数据相同时
SCIX1 中断 (Control Field 1 相同)	CF1MF	当 Control Field 1 的接收数据和 PCF1DR 或者 SCF1DR 中设定的数据相同时
SCIX1 中断 (检测到优先级中断位)	PIBDF	当优先级中断位中指定的位数据和 PCF1DR 中设定的数据相同时
SCIX2 中断 (检测到总线冲突)	BCDF	通过总线冲突检测时钟对 TXDX12 引脚的输出和 RXDX12 引脚的输入进行采样，连续 3 次不同时
SCIX3 中断 (检测到有效边沿)	AEDF	在测量位数率的过程中检测到有效边沿时

27.12 使用时的注意事项

27.12.1 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, 复位后的值为停止 SCI 的运行。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“11. 低功耗功能”。

27.12.2 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 标志被置“1” (发生帧错误), 并且 SSR.PER 标志也有可能被置“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 标志也会再次被置“1”。

27.12.3 标记状态和中止的发送

当 SCR.TE 位为“0” (禁止串行发送) 时, 能通过设定 I/O 端口功能, 将 TXDn 引脚用作能选择输入 / 输出方向和电平的 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。

为了在将 SCR.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), 通过设定 I/O 端口功能将 TXDn 引脚设定为输出 High 电平。另外, 要在发送数据时发送中止的情况下, 必须在通过设定 I/O 端口功能将 TXDn 引脚设定为输出 Low 电平后将 SCR.TE 位置“0”。如果将 SCR.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, 并且根据 I/O 端口功能的设定, 从 TXDn 引脚输出 Low 电平或者 High 电平。

27.12.4 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) 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), 也不能将接收错误标志置“0”。

27.12.5 有关 TDR 寄存器的写操作

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。

27.12.6 时钟同步发送时的限制事项

如果将外部时钟源用于同步时钟, 就必须在通过 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。

27.12.7 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。

27.12.8 有关开始通信的注意事项

如果在开始通信时中断控制器的中断状态标志为“1”，就必须在允许运行（将 SCR.TE 位或者 SCR.RE 位置“1”）前按以下步骤清除中断请求：

- 确认通信处于停止状态（SCR.TE 位或者 SCR.RE 位为“0”）。
- 将对应的中断允许位（SCR.TIE 位或者 SCR.RIE 位）置“0”。
- 读对应的中断允许位（SCR.TIE 位或者 SCR.RIE 位），确认是“0”。
- 将中断控制器的中断状态标志置“0”。

27.12.9 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、SCR.TE 位 =0、SCR.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。通过清除 TE 位，对 TSR 寄存器、TDR 寄存器和 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。如果在发送过程中进行转移，发送中的数据就不确定。

在解除低功耗状态后不更改发送模式而进行发送时，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。

发送时的软件待机模式转移的流程图例子如图 27.67 所示，软件待机模式转移时的端口的引脚状态如图 27.68 和图 27.70 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。如果在接收时进行转移，正在接收的数据就无效。

在解除低功耗状态后不更改接收模式而进行接收时，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。

接收时的软件待机模式转移的流程图例子如图 27.70 所示。

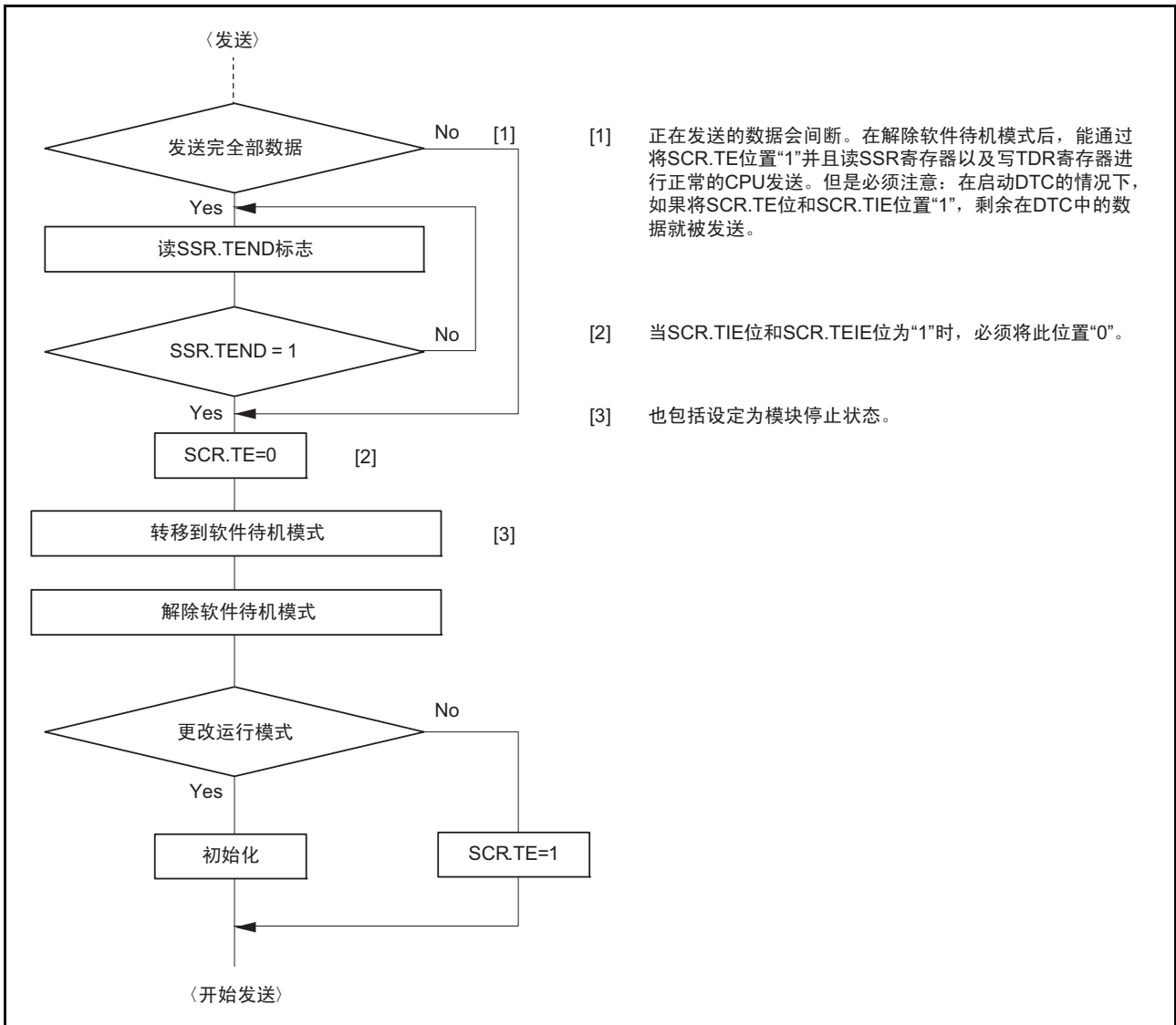


图 27.67 发送时向软件待机模式转移的流程图例子

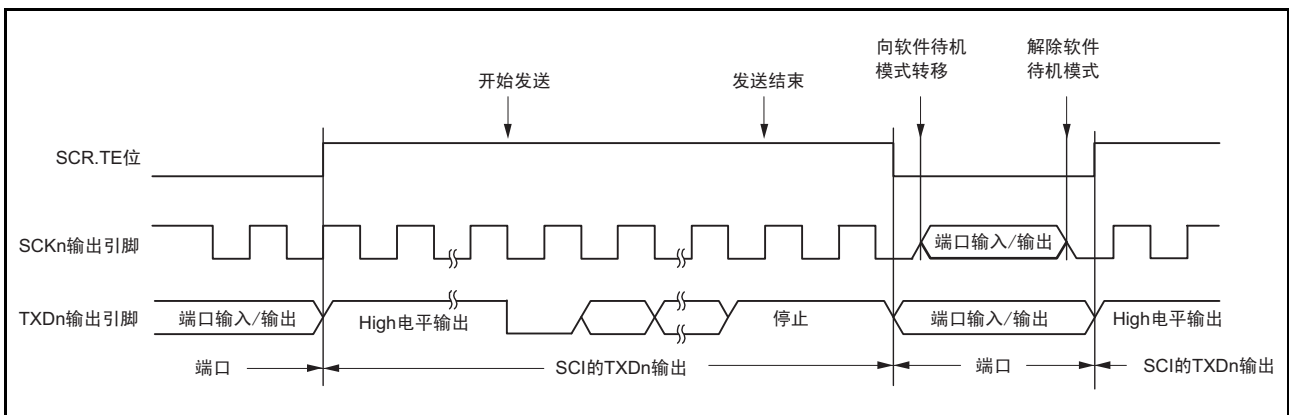


图 27.68 向软件待机模式转移时的端口引脚状态 (内部时钟、异步发送)

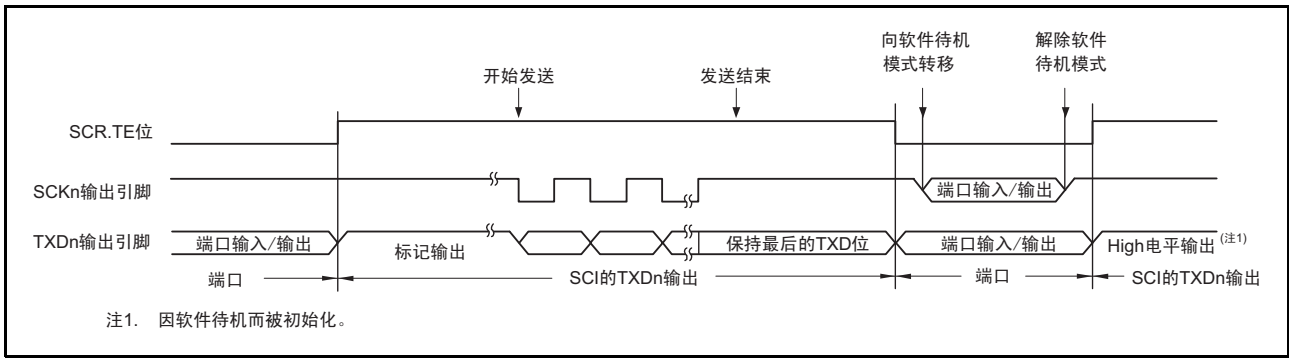


图 27.69 向软件待机模式转移时的端口引脚状态 (内部时钟、时钟同步发送)

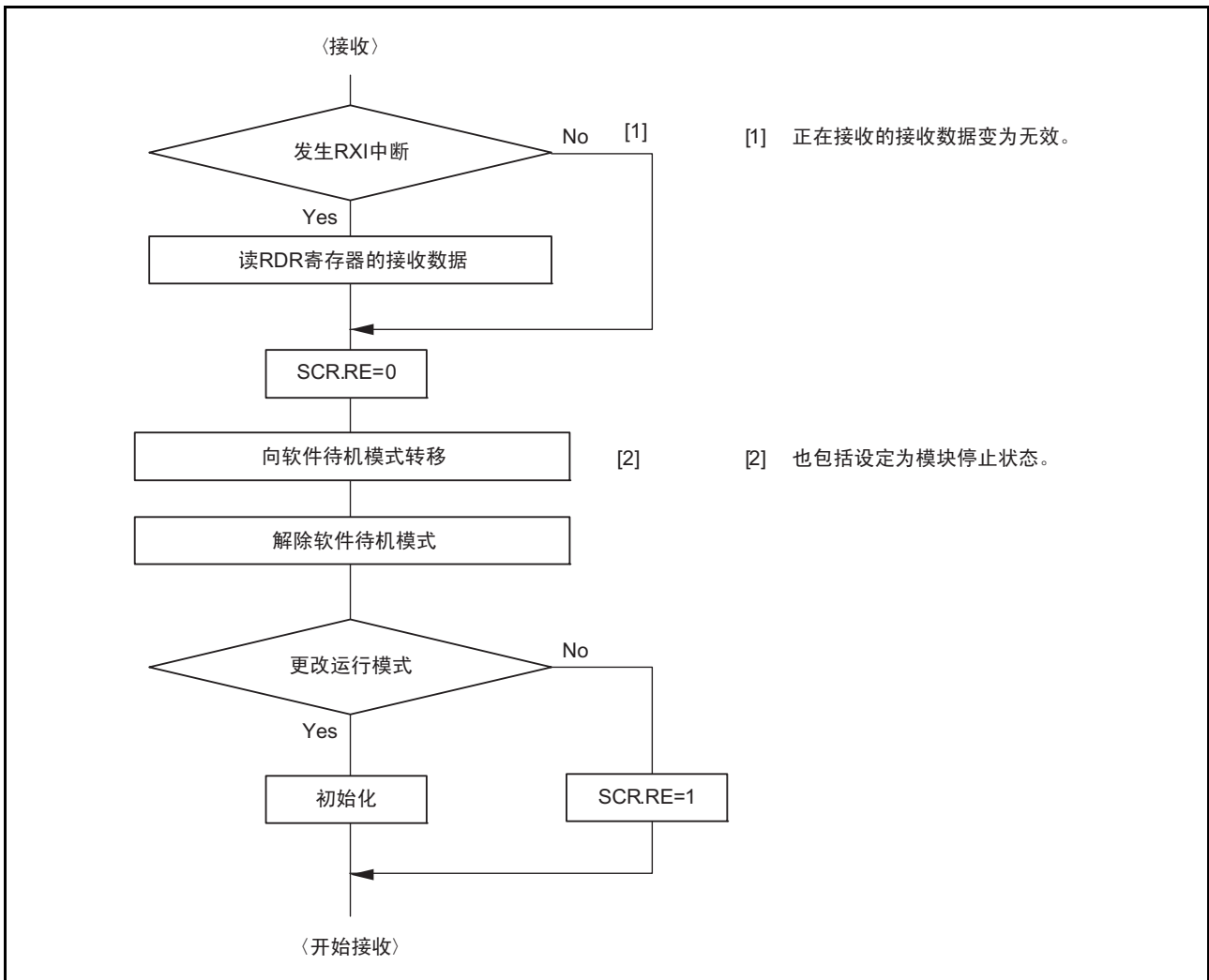


图 27.70 接收时向软件待机模式转移的流程图例子

27.12.10 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 输入必须满足：High 电平脉冲期间和 Low 电平脉冲期间必须至少为 2 个 PCLK，周期至少为 6 个 PCLK。

27.12.11 简易 SPI 模式的限制事项

(1) 主控模式

- 必须配合由 SPMR.CKPH 位和 SPMR.CKPOL 位设定的发送和接收时钟的初始值，用电阻上拉（下拉）时钟线。
- 当设定为有时钟延迟（SPMR.CKPH 位=1）时，如图 27.71 所示，在 SCKn 引脚的最后时钟的前一个边沿产生接收数据满中断（RXI 中断）。此时，必须注意：如果立即将 SCR.TE 位和 SCR.RE 位置“0”，SCKn 引脚的输出就变为高阻抗，最后发送和接收时钟的时钟脉宽变短。另外，还必须注意：如果在发生 RXI 中断后立即将连接从属设备的 SSn# 引脚输入信号置为 High 电平，从属设备就可能发生误动作。
- 在多主控模式中，如果在发送和接收字符的中途发生模式故障错误，就在 SS# 引脚的输入为 Low 电平期间 SCKn 引脚的输出为高阻抗，并且停止给连接的从属设备提供发送和接收时钟。为了避免在重新开始发送和接收时发生位的错位，必须重新设定连接的从属设备。

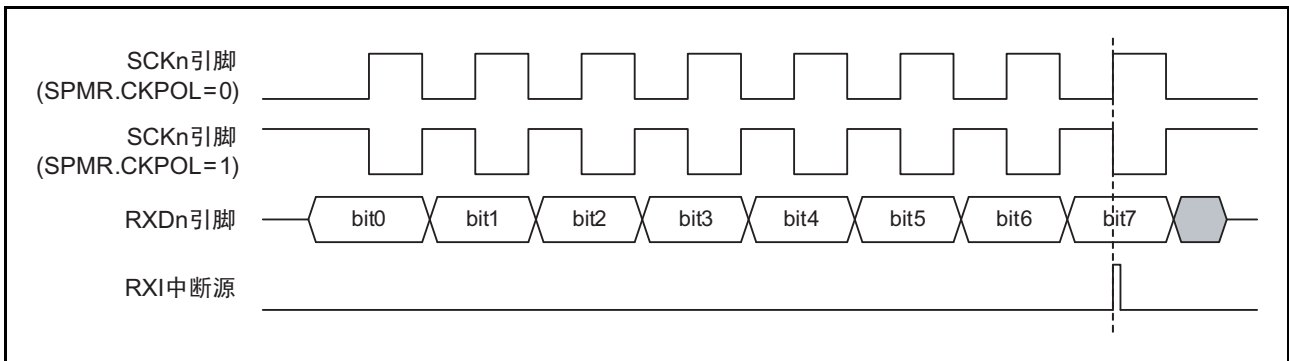


图 27.71 简易 SPI 模式（有时钟延迟）的 RXI 中断产生时序

(2) 从属模式

- 从给 TDR 寄存器写发送数据并且给 SSn# 引脚输入 Low 电平到开始输入外部时钟为止，必须至少确保 5 个 PCLK 的时间。
- 由主控提供的外部时钟必须和传送数据的长度相同。
- 必须在数据传送的开始前和结束后控制 SSn# 引脚的输入。
- 如果在发送和接收字符的中途将 SSn# 引脚的输入从 Low 电平变为 High 电平，就必须将 SCR.TE 位和 SCR.RE 位置“0”，并且在设定后从第一字节开始重新传送。

27.12.12 扩展串行模式控制部的使用限制事项 1

如果将 PCR 的 SHARPS 位置“1”，就在以下情况下 TXDX12/RXDX12 引脚变为输出引脚：

- 当 SCId 定时器运行在 Break Field Low width 输出模式中并且将 TCR 的 TCST 位置“1”时（从将 TCR 的 TCST 位置“1”到输出 Low 电平为止，最多输出 1 个定时器计数时钟源周期的 High 电平）
- 当 SCI12.SCR 的 TE 位为“1”时

27.12.13 扩展串行模式控制部的使用限制事项 2

即使将扩展串行模式置为有效，也生成 SCIc 的中断请求。因为 SCIId 在接收起始帧的过程中使用 SCIc 的中断请求事件，所以不能使用 SCIc 的中断请求。有以下 2 种对应方法。而且，在检测到接收错误时，必须按照图 27.72 的流程图的例子，清除 SCIc 的错误标志并且对 SCIId 控制部进行初始化。

1. 必须将 SCIc 的 SCR.RIE 位置“0”，禁止输出中断请求。此时，因为在发生接收错误时不产生 ERI 中断，所以必须在接收完起始帧时检查 SCIc 的 SSR 寄存器的错误标志。在从接收完起始帧到接收完信息帧的第 1 字节为止的期间，必须将 SCIc 的 SCR.RIE 位置“1”。
2. 必须将 SCIc 的 SCR.RIE 位置“1”，禁止 ICU 的 RXI 中断而允许 ICU 的 ERI 中断。在从接收完起始帧到接收完信息帧的第 1 字节为止的期间，必须清除 ICU 的 RXI 中断对应的 IRn.IR 标志，允许 ICU 的 RXI 中断。

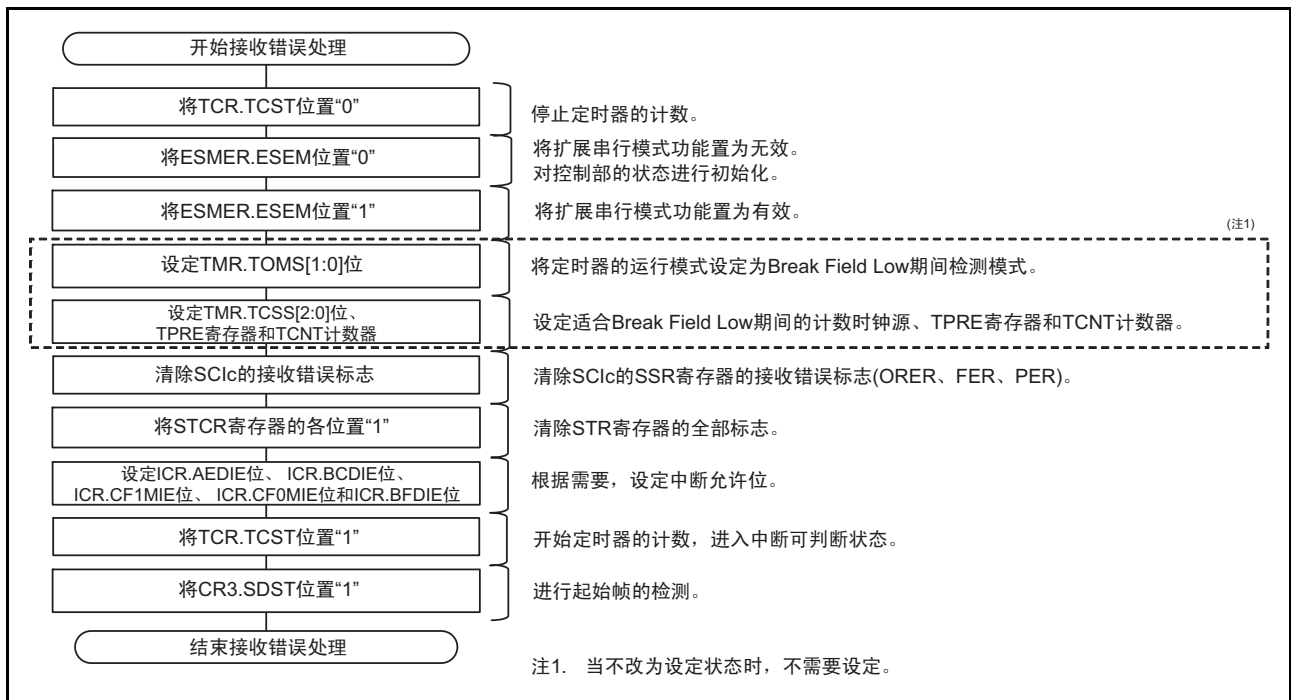


图 27.72 接收错误处理的流程图例子（接收起始帧的过程中）

28. I²C 总线接口 (RIIC)

RX63T 群内置 1 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, 内置了子集功能。

28.1 概要

RIIC 的规格和框图分别如表 28.1 和图 28.1 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 28.2 所示, RIIC 使用的输入 / 输出引脚如表 28.2 所示。

表 28.1 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。 自动确保与所设传送速度对应的各种准备时间、保持时间和总线空闲时间。
传送速度	~ 400kbps
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。
条件发行和条件检测	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"> 在发送时自动装入应答位。 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
低功耗功能	能设定为模块停止状态。

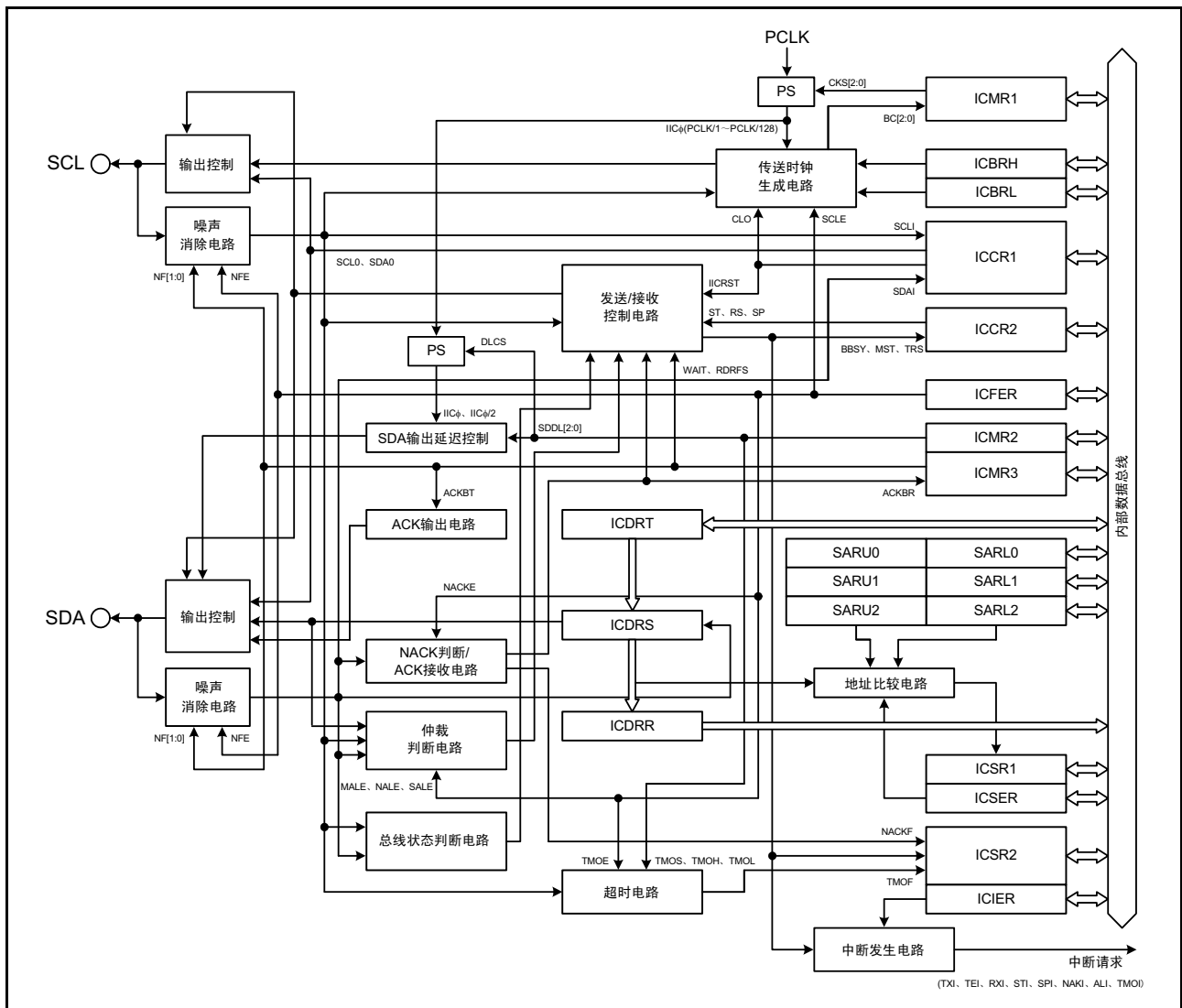
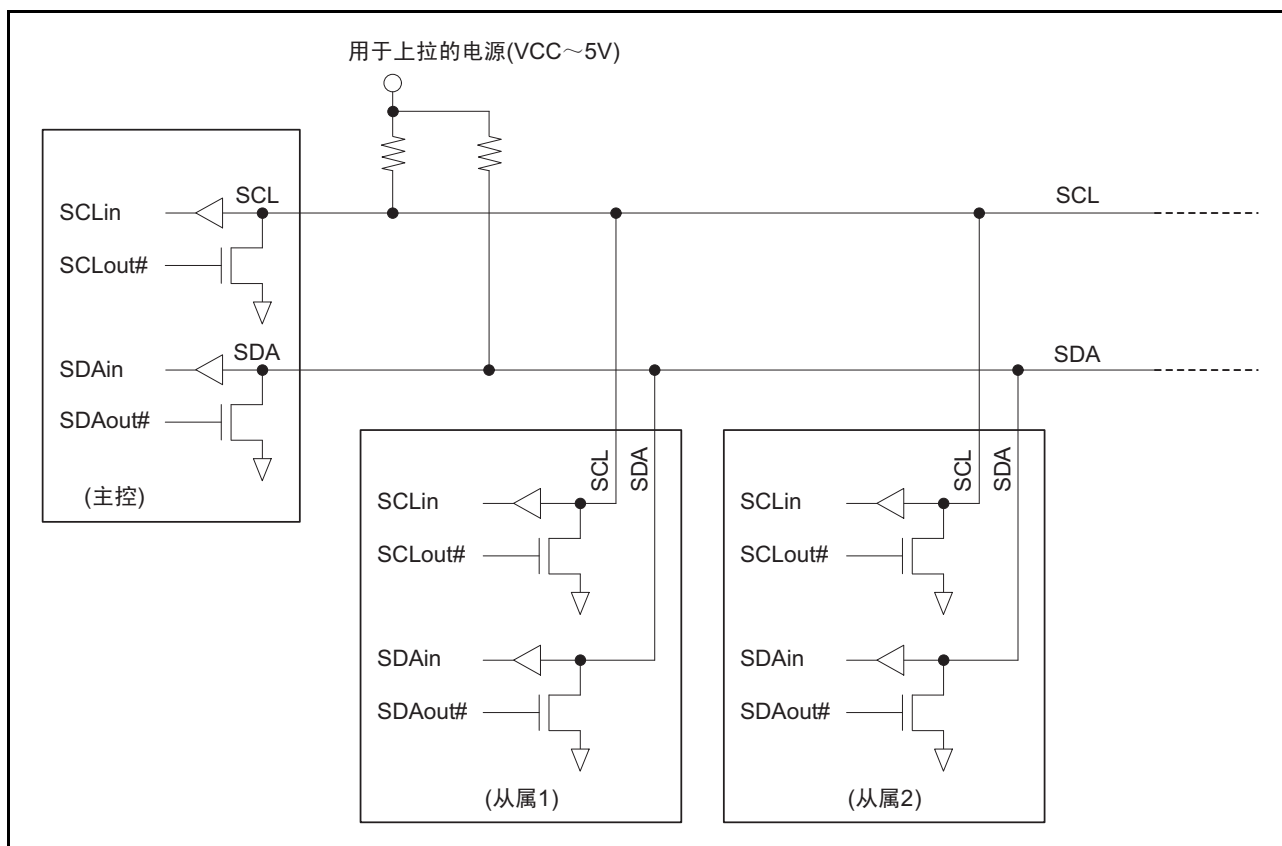


图 28.1 RIIC 的框图

图 28.2 输入 / 输出引脚的外部电路连接例子 (I²C 总线的结构例子)

当选择 I²C 总线时 (ICMR3.SMBS 位 =0)，

RIIC 各信号的输入电平为 CMOS 电平；当选择 SMBus 时 (ICMR3.SMBS 位 =1)，RIIC 各信号的输入电平为 TTL 电平。

表 28.2 RIIC 的输入 / 输出引脚

通道	引脚名	输入 / 输出	功能
RIIC0	SCL	输入 / 输出	串行时钟的输入 / 输出引脚
	SDA	输入 / 输出	串行数据的输入 / 输出引脚

28.2 寄存器说明

28.2.1 I²C 总线控制寄存器 1 (ICCR1)

地址 RIIC0.ICCR1 0008 8300h

	b7	b6	b5	b4	b3	b2	b1	b0
	CE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
复位后的值	0	0	0	1	1	1	1	1

位	符号	位名	功能	R/W
b0	SDAI	SDA 总线输入监视位	0: SDA 引脚输入为 Low 电平 1: SDA 引脚输入为 High 电平	R
b1	SCLI	SCL 总线输入监视位	0: SCL 引脚输入为 Low 电平 1: SCL 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 引脚为 Low 电平输出 1: SDA 引脚为高阻抗 写时 0: 将 SDA 引脚改为 Low 电平 1: 将 SDA 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 引脚为 Low 电平输出 1: SCL 引脚为高阻抗 写时 0: 将 SCL 引脚改为 Low 电平输出 1: 将 SCL 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/SDA 的输出锁存)	R/W
b7	ICE	I ² C 总线接口允许位	0: 禁止 (SCL、SDA 引脚为非驱动状态) 1: 允许 (SCL、SDA 引脚为驱动状态) (通过与 IICRST 位组合, 选择 RIIC 复位和内部复位)	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。

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，必须将此位置“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“28.11.2 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 28.3 RIIC 复位的种类”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“28.14 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 引脚 /SDA 引脚置为高阻抗，然后释放总线。

注 1. 在从属模式中和主控设备进行通信时，如果因总线发生意外停机而通过 IICRST 位进行内部复位，RIIC 就可能变为和主控设备的状态不同的状态 (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设备进行恢复处理。在从属模式中将 SCL 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，就必须在进行内部复位后从主控设备发行重新开始条件，或者在发行停止条件后发行开始条件，重新开始通信。如果只单独对从属设备进行复位，并且在没有从主控设备发行开始条件或者重新开始条件的情况下重新开始通信，就可能因双方运行状态的差异而导致不同步。

表 28.3 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 SCL 引脚和 SDA 引脚的驱动状态和非驱动状态。另外，ICE 位通过与 IICRST 位的组合，能进行 2 种复位。有关复位的种类，请参照“表 28.3 RIIC 复位的种类”。

在使用 RIIC 时，必须将 ICE 位设定为“1”。在 ICE 位为“1”时，SCL 引脚和 SDA 引脚变为驱动状态。

在不使用 RIIC 时，必须将 ICE 位设定为“0”。在 ICE 位为“0”时，SCL 引脚和 SDA 引脚变为非驱动状态。

另外，不能通过设定多功能引脚控制器 (MPC) 将 SCL 引脚和 SDA 引脚分配到 RIIC。必须注意；如果分配到 RIIC，就进行从属地址的比较运行。

28.2.2 I²C 总线控制寄存器 2 (ICCR2)

地址 RIIC0.ICCR2 0008 8301h

b7	b6	b5	b4	b3	b2	b1	b0
BBSY	MST	TRS	—	SP	RS	ST	—
0	0	0	0	0	0	0	0

复位后的值

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。

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“28.10 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]

- 当写“1”时

[为“0”的条件]

- 当写“0”时
- 当结束开始条件的发行时
- 当 ICSR2.AL 标志变为“1” (仲裁失败) 时
- 当给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

注. 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“28.10 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]

- 当写“0”时
- 当结束重新开始条件的发行或者检测到开始条件时
- 当 ICSR2.AL 标志变为“1”（仲裁失败）时
- 当给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

注. 不能在发行停止条件的过程中将 RS 位置“1”。

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“28.10 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]

- 当写“0”时
- 当结束停止条件的发行或者检测到停止条件时
- 当 ICSR2.AL 标志变为“1”（仲裁失败）时
- 当检测到开始条件和重新开始条件时
- 当给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

注. 不能在 BBSY 标志为“0”（总线空闲）时写此位。

不能在发行重新开始条件的过程中将 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”，并且运行模式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 R/W# 位接收到“1”时
- 在 ICMR1.MTWP 位为“1”的状态下写“1”时

[为“0”的条件]

- 当检测到停止条件时
- 当 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 ICSE 寄存器的有效地址匹配, 并且 R/W# 位接收到“0” (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
- 当给 ICCR1.IICRST 位写“1”并且进行 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 MST 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”, 并且运行模式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]

- 当检测到停止条件时
- 当 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
- 当给 ICCR1.IICRST 位写“1”并且进行 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 SCL 线为 High 电平的状态下 SDA 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 SCL 线为 High 电平的状态下 SDA 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]

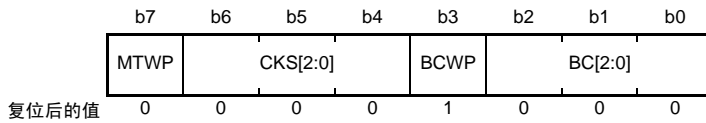
- 当检测到开始条件时

[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

28.2.3 I²C 总线模式寄存器 1 (ICMR1)

地址 RIIC0.ICMR1 0008 8302h



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。

BC[2:0] 位 (位计数器)

这些位是在 SCL 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 SCL 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”。

28.2.4 I²C 总线模式寄存器 2 (ICMR2)

地址 RIIC0.ICMR2 0008 8303h

b7	b6	b5	b4	b3	b2	b1	b0
DLCS	SDDL[2:0]		TMWE	TMOH	TMOL	TMOS	
0	0	0	0	0	1	1	0

复位后的值

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 SCL 线为 Low 电平时, 禁止计数。 1: 在 SCL 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 SCL 线为 High 电平时, 禁止计数。 1: 在 SCL 线为 High 电平时, 允许计数。	R/W
b3	TMWE	超时内部计数器的写允许位	0: 禁止写超时功能的内部计数器 1: 允许写超时功能的内部计数器	R/W
b6-b4	SDDL[2:0]	SDA 输出延迟计数器	<ul style="list-style-type: none"> 当 ICMR2.DLCS 位为“0” (IICϕ) 时 b6 b4 000: 没有输出延迟 001: 1 个 IICϕ 周期 010: 2 个 IICϕ 周期 011: 3 个 IICϕ 周期 100: 4 个 IICϕ 周期 101: 5 个 IICϕ 周期 110: 6 个 IICϕ 周期 111: 7 个 IICϕ 周期 当 ICMR2.DLCS 位为“1” (IIC$\phi/2$) 时 b6 b4 000: 没有输出延迟 001: 1 ~ 2 个 IICϕ 周期 010: 3 ~ 4 个 IICϕ 周期 011: 5 ~ 6 个 IICϕ 周期 100: 7 ~ 8 个 IICϕ 周期 101: 9 ~ 10 个 IICϕ 周期 110: 11 ~ 12 个 IICϕ 周期 111: 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 $\phi/2$) 为 SDA 输出延迟计数器的时钟源 (注 1)	R/W

注 1. 只在 SCL 为 Low 电平时 DLCS 为“1” (IIC $\phi/2$) 的设定有效。在 SCL 为 High 电平时 DLCS 为“1”的设定为无效并且为内部基准时钟 (IIC ϕ)。

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 SCL 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“28.11.1 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 SCL 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 SCL 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。

TMWE 位 (超时内部计数器的写允许位)

此位选择是否将超时内部计数器 (TMOCNTL/TMOCNTU) 分配到从属地址寄存器 (SARL0/SARU0) 的地址。

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“28.5 SDA 输出延迟功能”。

注. 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 1) 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)，否则就可能引起通信设备的通信故障，或者根据总线状态视为开始条件或者停止条件。

注 1. 数据有效时间 / 应答有效时间

3450ns (～ 100kbps: 标准模式 [Sm])

900ns (～ 400kbp: 快速模式 [fm])

28.2.5 I²C 总线模式寄存器 3 (ICMR3)

地址 RIIC0.ICMR3 0008 8304h

b7	b6	b5	b4	b3	b2	b1	b0
SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[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	NF[1:0]	噪声滤波器的段数选择位	b1 b0 0 0: 消除不超过 1 个 IIC ϕ 的噪声 (滤波器为 1 段) 0 1: 消除不超过 2 个 IIC ϕ 的噪声 (滤波器为 2 段) 1 0: 消除不超过 3 个 IIC ϕ 的噪声 (滤波器为 3 段) 1 1: 消除不超过 4 个 IIC ϕ 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” (接收 ACK) 1: 应答位接收“1” (接收 NACK)	R
b3	ACKBT	发送应答位	0: 应答位发送“0” (发送 ACK) 1: 应答位发送“1” (发送 NACK)	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, 此位为“1”。 (在第 8 个时钟的下降沿, SCL 线不保持 Low 电平) 1: 在 SCL 时钟的第 8 个时钟上升时, 此位为“1”。 (在第 8 个时钟的下降沿, SCL 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。

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. 必须注意: 用噪声滤波器消除的噪声宽度必须小于 SCL 线 High/Low 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
- 当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. 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 SCL 线的 Low 电平。

当 RDRFS 位为“0”时, 在第 8 个时钟的下降沿 SCL 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 SCL 线保持 Low 电平。通过写 ACKBT 位来解除此 SCL 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 SCL 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, SCL 线保持 Low 电平。因此能按字节接收数据。

注. 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 IC SER.HOAE 位有效。

28.2.6 I²C 总线功能允许寄存器 (ICFER)

地址 RIIC0.ICFER 0008 8305h

b7	b6	b5	b4	b3	b2	b1	b0
—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
0	1	1	1	0	0	1	0

复位后的值

位	符号	位名	功能	R/W
b0	TMOE	超时检测功能有效位	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有效位	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 同步电路有效位	0: SCL 同步电路无效 1: SCL 同步电路有效	R/W
b7	—	保留位	读写值都为“0”。	R/W

TMOE 位 (超时检测功能有效位)

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“28.11.1 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设备, 或者 2 个以上 (包括 2 个) 的主控设备同时选择了相同的从属设备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设备，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，此位选择在从从属设备接收到 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“28.8.2 NACK 接收传送中止功能”。

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 SCL 线的状态无关，RIIC 输出 ICBRH 寄存器和 ICBRL 寄存器所设传送速度的 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速度时才能将 SCLE 位置“0”。

28.2.7 I²C 总线状态允许寄存器 (ICSER)

地址 RIIC0.ICSER 0008 8306h

b7	b6	b5	b4	b3	b2	b1	b0
HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
0	0	0	0	1	0	0	1

复位后的值

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

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 位 (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“28.7.3 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。

28.2.8 I²C 总线中断允许寄存器 (ICIER)

地址 RIIC0.ICIER 0008 8307h

b7	b6	b5	b4	b3	b2	b1	b0
TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
0	0	0	0	0	0	0	0

复位后的值

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

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)。

28.2.9 I²C 总线状态寄存器 1 (ICSR1)

地址 RIIC0.ICSR1 0008 8308h

b7	b6	b5	b4	b3	b2	b1	b0
HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
0	0	0	0	0	0	0	0

复位后的值

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[6:0] 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[1:0] 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[6:0] 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[1:0] 匹配, 并且后续的地址与 SARL1 寄存器匹配时 (在 SARL1 一致判断帧的 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[6:0] 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[1:0] 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 • 当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 • 当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 • 当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”。

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[6:0] 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[1:0] 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]

- 在读“1”后写“0”时
- 当检测到停止条件时
- 当给 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[6:0] 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[1:0] 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[1:0] 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]

- 在读“1”后写“0”时
- 当检测到停止条件时
- 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。
- 当给 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 ICSEr.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]

- 在读“1”后写“0”时
- 当检测到停止条件时
- 在 ICSEr.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 ICSEr.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。
- 当给 ICCR1.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]

- 在读“1”后写“0”时
- 当检测到停止条件时
- 当给 ICMR3.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。
- 当给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

28.2.10 I²C 总线状态寄存器 2 (ICSR2)

地址 RIIC0.ICSR2 0008 8309h

b7	b6	b5	b4	b3	b2	b1	b0
TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
0	0	0	0	0	0	0	0

复位后的值

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”。

TMOF 标志 (超时检测标志)

如果 SCL 线的状态在一定的期间内不发生变化, 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 ICMR2.TMOH 位、ICMR2.TMOL 位、ICMR2.TMOS 位所选条件的期间内, SCL 线的状态没有变化时

[为“0”的条件]

- 在读“1”后写“0”时
- 当给 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 线的电平, 如果输出数据和 SDA 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 也能在可控模式中检测发送 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 位为 “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 线上的信号状态不同（内部 SDA 输出为 High 电平输出（=SDA 引脚为高阻抗）而检测到 SDA 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 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 位为 “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 线上的信号状态不同时

【从属仲裁失败的检测有效时：ICFER.SALE 位为 “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 线上的信号状态不同时

[为 “0” 的条件]

- 在读 “1” 后写 “0” 时
- 当给 ICCR1.IICRST 位写 “1” 并且进行 RIIC 复位或者内部复位时

表 28.4 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	x	x	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 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。
x	1	x	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。
x	x	1	1	发送数据不同	在 从属发送模式中，发送数据和总线状态不同。

x: Don't care

START 标志（开始条件检测标志）

[为 “1” 的条件]

- 当检测到开始条件（包括重新开始条件）时

[为 “0” 的条件]

- 在读 “1” 后写 “0” 时
- 当检测到停止条件时
- 当给 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]

- 当检测到停止条件时

[为 “0” 的条件]

- 在读 “1” 后写 “0” 时
- 当给 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.NACKF 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]

- 在读“1”后写“0”时
- 当给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

注. 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]

- 在读“1”后写“0”时
- 当读 ICDRR 寄存器时
- 当给 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]

- 在读“1”后写“0”时
- 当给 ICDRT 寄存器写数据时
- 当检测到停止条件时
- 当给 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]

- 当将数据从 ICDRT 寄存器传送到 ICDRS 寄存器并且 ICDRT 寄存器为空时
- 当 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. 当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
- 当接收的从属地址匹配并且 TRS 位为“1”时

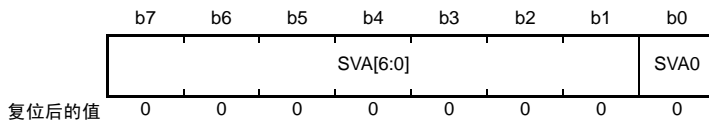
[为“0”的条件]

- 当给 ICDRT 寄存器写数据时
- 当 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
- 当给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

注. 如果在 ICFER.NACKF 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”。

28.2.11 从属地址寄存器 Ly (SARLy) (y=0 ~ 2)

地址 RIIC0.SARL0 0008 830Ah、RIIC0.SARL1 0008 830Ch、RIIC0.SARL2 0008 830Eh



位	符号	位名	功能	R/W
b0	SVA0	10 位地址的最低位	必须设定从属地址。 • 当 SARUy.FS 位为“0”（选择 7 位地址格式）时，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[6:0] 位合并为 10 位从属地址的低 8 位地址。	R/W
b7-b1	SVA[6:0]	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[6:0] 位为 7 位从属地址。 • 当 SARUy.FS 位“1”（选择 10 位地址格式）时，SVA[6:0] 位和 SVA0 位合并为 10 位从属地址的低 8 位地址。	R/W

SVA0 位（10 位地址的最低位）

在选择 10 位地址格式时（SARUy.FS 位 =1），用作 10 位地址的最低位，此位和 SVA[6:0] 位一起设定 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。

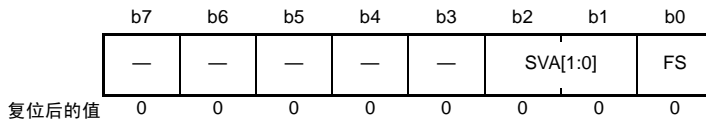
SVA[6:0] 位（7 位地址 /10 位地址的低位）

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 位一起用作 10 位地址的低 8 位。

当 IC SER.SARyE 位为“0”时，忽视设定值。

28.2.12 从属地址寄存器 Uy (SARUy) (y=0 ~ 2)

地址 RIIC0.SARU0 0008 830Bh、RIIC0.SARU1 0008 830Dh、RIIC0.SARU2 0008 830Fh



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[1:0]	10 位地址的高位	必须设定从属地址。 • 当 SARUy.FS 位为“0”（选择 7 位地址格式）时，SVA[1:0] 位无效。 • 当 SARUy.FS 位为“1”（选择 10 位地址格式）时，SVA[1:0] 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

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、SARUy 寄存器) 为 7 位地址或者 10 位地址。

当 ICSEr.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[6:0] 位的设定值有效，忽视 SVA[1:0] 位和 SARLy.SVA0 位的设定值。

当 ICSEr.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 y 为 10 位地址格式，SVA[1:0] 位和 SARLy 寄存器的设定值有效。

当 ICSEr.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[1:0] 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 ICSEr.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。

28.2.13 I²C 总线位速率低电平寄存器 (ICBRL)

地址 RIIC0.ICBRL 0008 8310h



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。

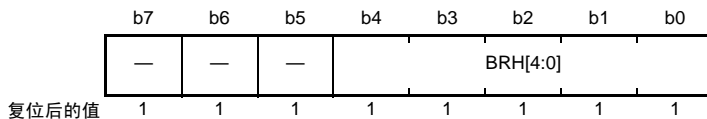
在 SCL 自动保持 Low 电平时（参照“28.8 SCL 的 Low 电平自动保持功能”），ICBRL 寄存器用作数据准备时间的确保持存器。因此，在 RIIC 总是用于从属模式的情况下，设定值不能小于数据准备时间（注 1）的值。

通过 ICMR1.CKS[2:0] 位选择的内部基准时钟源（IIC ϕ ），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]）

28.2.14 I²C 总线位速率高电平寄存器 (ICBRH)

地址 RIIC0.ICBRH 0008 8311h



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 寄存器是用于设定 SCL 时钟的 High 电平宽度的 5 位寄存器，在主机模式中有效。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 传送速度和 SCL 时钟的占空比。

$$\text{传送速度} = 1 / \{ [(ICBRH+1) + (ICBRL+1)] / IIC\phi \text{ (注 1)} + \text{SCL 线的上升时间 [tr]} + \text{SCL 的下降时间 [tf]} \}$$

$$\text{占空比} = \{ \text{SCL 线的上升时间 [tr]} \text{ (注 2)} + (ICBRH+1) / IIC\phi \} / \{ \text{SCL 线的下降时间 [tf]} \text{ (注 2)} + (ICBRL+1) / IIC\phi \}$$

注 1. IIC ϕ = PCLK × 10⁶ × 分频比

注 2. SCL 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 28.5 所示。

表 28.5 对应传送速度的 ICBRH 寄存器和 ICBRL 寄存器的设定例子 (1)

传送速度 (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)

表 28.5 对应传送速度的 ICBRH 寄存器和 ICBRL 寄存器的设定例子 (2)

传送速度 (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)

表 28.5 对应传送速度的 ICBRH 寄存器和 ICBRL 寄存器的设定例子 (3)

传送速度 (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)

注. 计算的设定例子是假设:

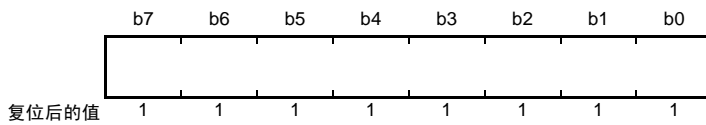
SCL 线的上升时间 (tr) 不超过 ~100kbps, [Sm] 为 1000ns 或者 ~400kbps, [Fm] 为 300ns,

SCL 线的下降时间 (tf) 不超过 ~400kbps, [Sm/Fm] 为 300ns。

有关 SCL 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。

28.2.15 I²C 总线发送数据寄存器 (ICDRT)

地址 RIIC0.ICDRT 0008 8312h



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。

28.2.16 I²C 总线接收数据寄存器 (ICDRR)

地址 RIIC0.ICDRR 0008 8313h



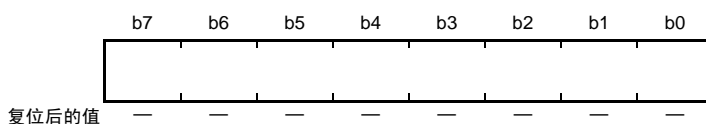
如果 1 字节数据接收结束, 就能将接收数据从 I²C 总线移位寄存器 (ICDRS) 传送到 ICDRR 寄存器, 进入能接收下一个数据的状态。

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 SCL 时钟自动保持 Low 电平。

28.2.17 I²C 总线移位寄存器 (ICDRS)



ICDRS 寄存器是用于发送和接收数据的移位寄存器。

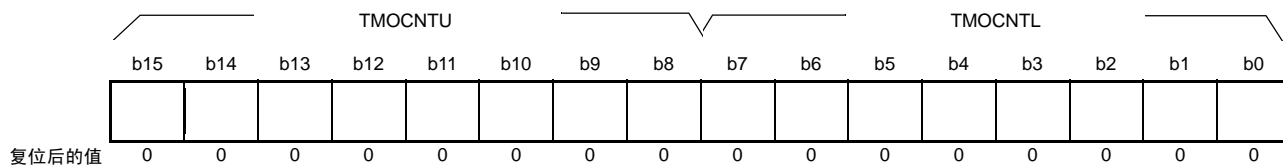
在发送时, 将发送数据从 ICDRT 寄存器传送到 ICDRS 寄存器, 从 SDA 引脚发送数据。在接收时, 一旦 1 字节数据接收结束, 就将数据从 ICDRS 寄存器传送到 ICDRR 寄存器。

不能直接存取 ICDRS 寄存器。

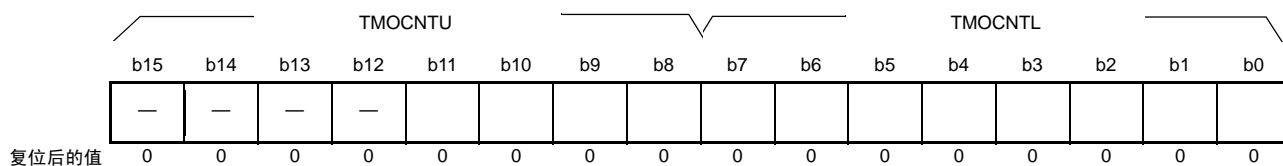
28.2.18 超时内部计数器 (TMOCNT)

地址 RIIC0.TMOCNTL 0008 830Ah (注1)
RIIC0.TMOCNTU 0008 830Bh (注1)

- ICMR2.TMOS = 0 (长模式) 时



- ICMR2.TMOS = 1 (短模式) 时



注1. 必须注意: 该寄存器与 SARL0 寄存器、SARU0 寄存器地址相同。

位	符号	位名	功能	R/W
b15-b8	TMOCNTU	超时内部计数器	超时内部计数器的高位 (注1)	W (注2)
b7-b0	TMOCNTL		超时内部计数器的低位	

注1. 当 TMOS=1 (短模式) 时, b15-b12 为保留位。可以写, 但是写入值无效。

注2. 不能读超时内部计数器的值。如果读, 读取值为“FFFFh”。

复位时, 如果 ICCR1.IICRST=1、或者 ICFER.TOME=1 并且 ICMR1.CKS[2:0]=000b (PCLK/1), 就在通过 ICMR2 的 TMOH/TMOL 位设定的计数器清除条件 (检测到 SCL 上升沿 / 下降沿) 成立时, 将超时内部计数器 (TMOCNTL/TMOCNTU) 初始化为“0000h”。

28.3 运行说明

28.3.1 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，用于指定主控设备通信对象的从属设备。在指定新的从属设备或者发行停止条件前，指定的从属设备有效。

I²C 总线的格式及其总线时序分别如图 28.3 和图 28.4 所示。

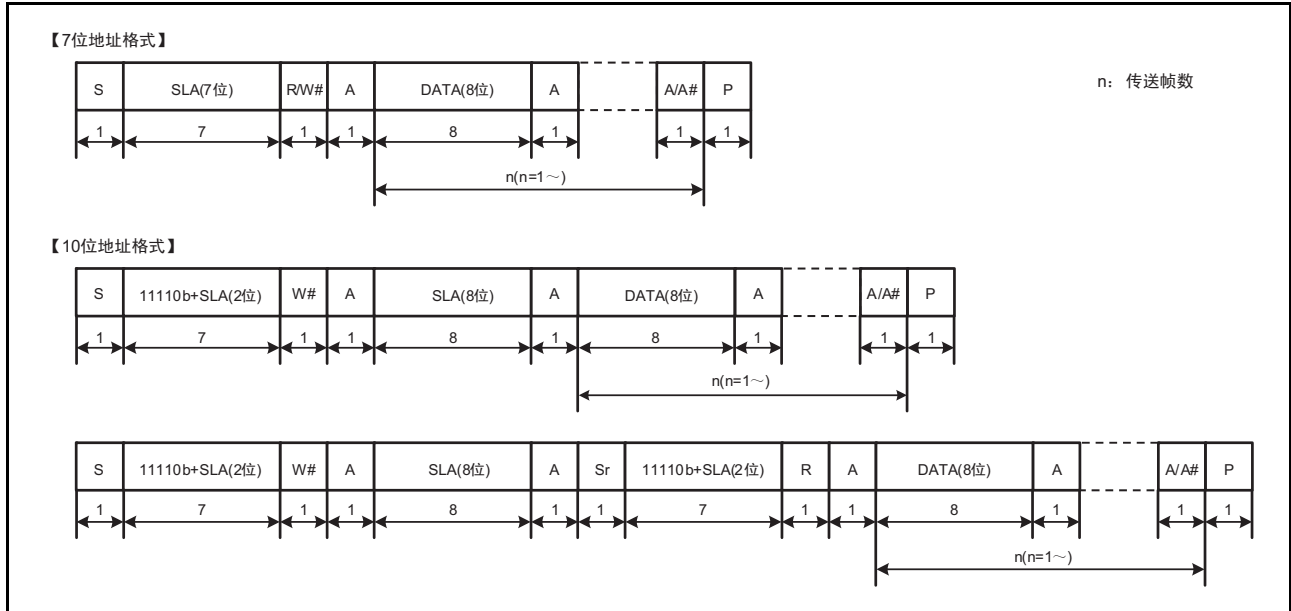


图 28.3 I²C 总线格式

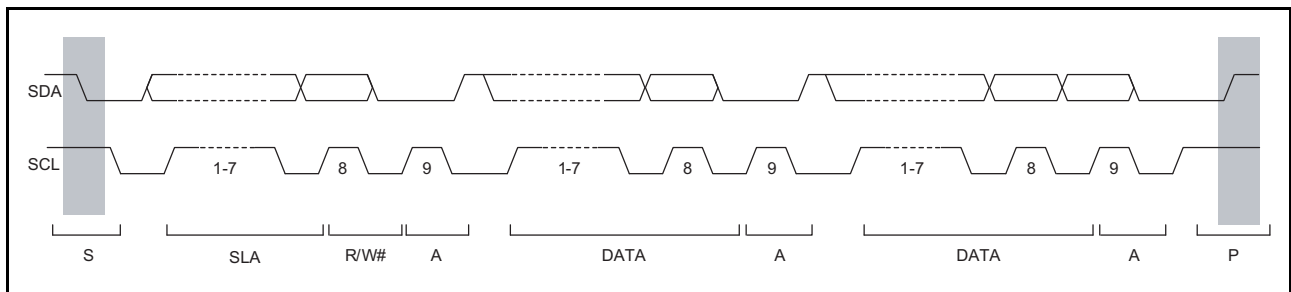


图 28.4 I²C 总线时序 (SLA 为 7 位)

- S : 表示开始条件。主控设备在 SCL 线为 High 电平的状态下将 SDA 线从 High 电平变为 Low 电平。
- SLA : 表示从属地址。主控设备选择从属设备。
- R/W# : 表示发送和接收的方向。当 R/W# 为“1”时，将数据从从属设备发送到主控设备；当 R/W# 为“0”时，将数据从主控设备发送到从属设备。
- A : 表示应答。接收设备将 SDA 线置为 Low 电平（在发送模式中，从属设备返回应答；在接收模式中，主控设备返回应答）。
- Sr : 表示重新开始条件。主控设备在 SCL 线为 High 电平的状态下经过准备时间后将 SDA 线从 High 电平变为 Low 电平。
- DATA : 表示发送和接收的数据。
- P : 表示停止条件。主控设备在 SCL 线为 High 电平的状态下将 SDA 线从 Low 电平变为 High 电平。

28.3.2 初始设定

在开始发送或者接收数据时，必须按照图 28.5 所示的步骤对 RIIC 进行初始化。



图 28.5 RIIC 的初始化流程图例子

28.3.3 主控发送

在主控发送模式中，主控设备的 RIIC 输出 SCL 时钟和发送数据，从属设备返回应答。主控发送模式的使用例子和运行时序分别如图 28.6 和图 28.7 ~ 图 28.9 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”（SCL 引脚和 SDA 引脚为非驱动状态）的状态下将 ICCR1.IICRST 位置“1”（RIIC 复位）后，将 ICCR1.ICE 位置“1”（内部复位）。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）并且必须根据需要进行其他寄存器（有关 RIIC 的初始设定请参照图 28.5）。必须在设定完所需的寄存器后将 ICCR1.IICRST 位置“0”（解除复位）。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 信号和 SDA 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 标志是“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设备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 标志是“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 线保持为 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 标志是“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。

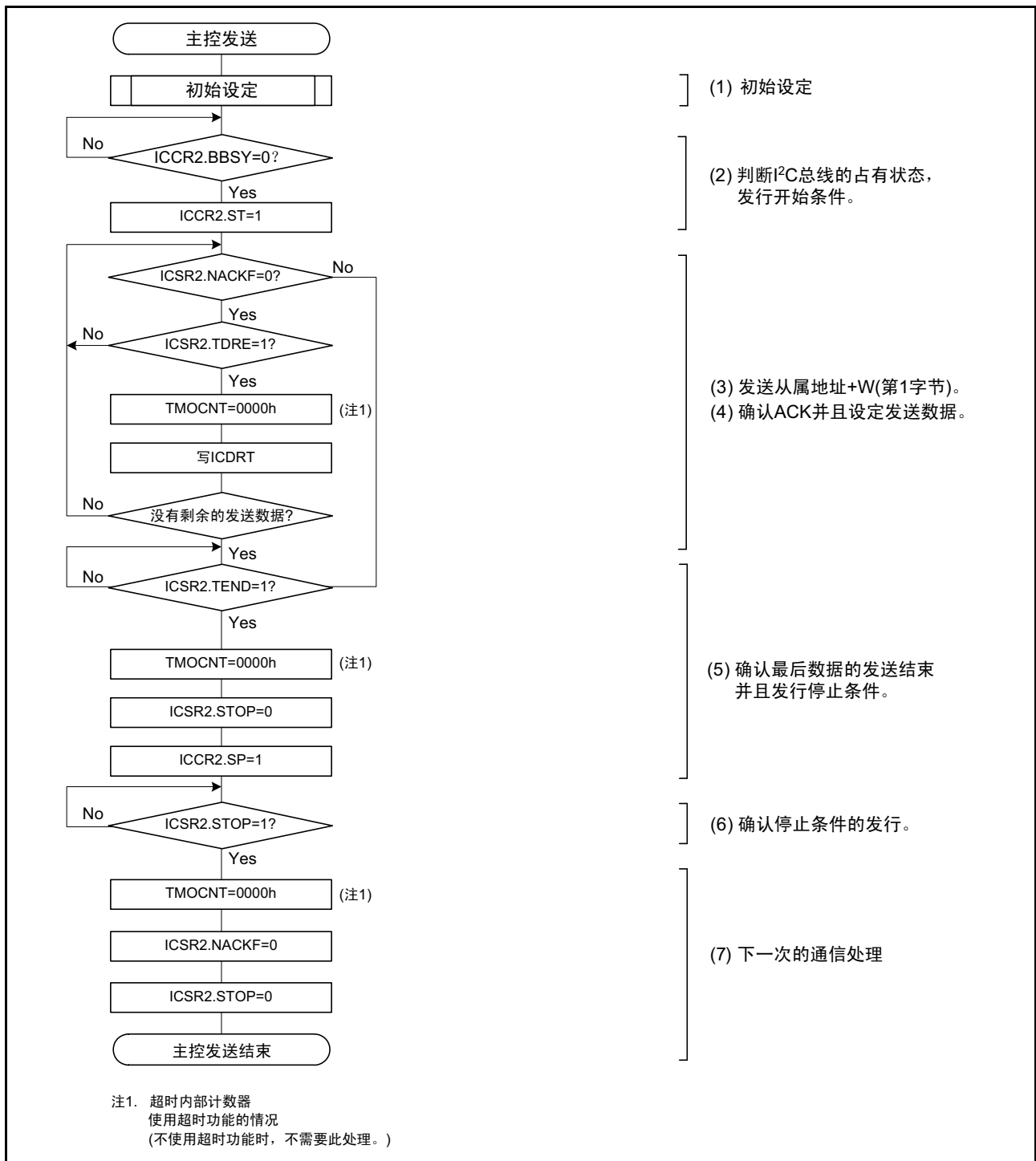


图 28.6 主控发送模式的流程图例子

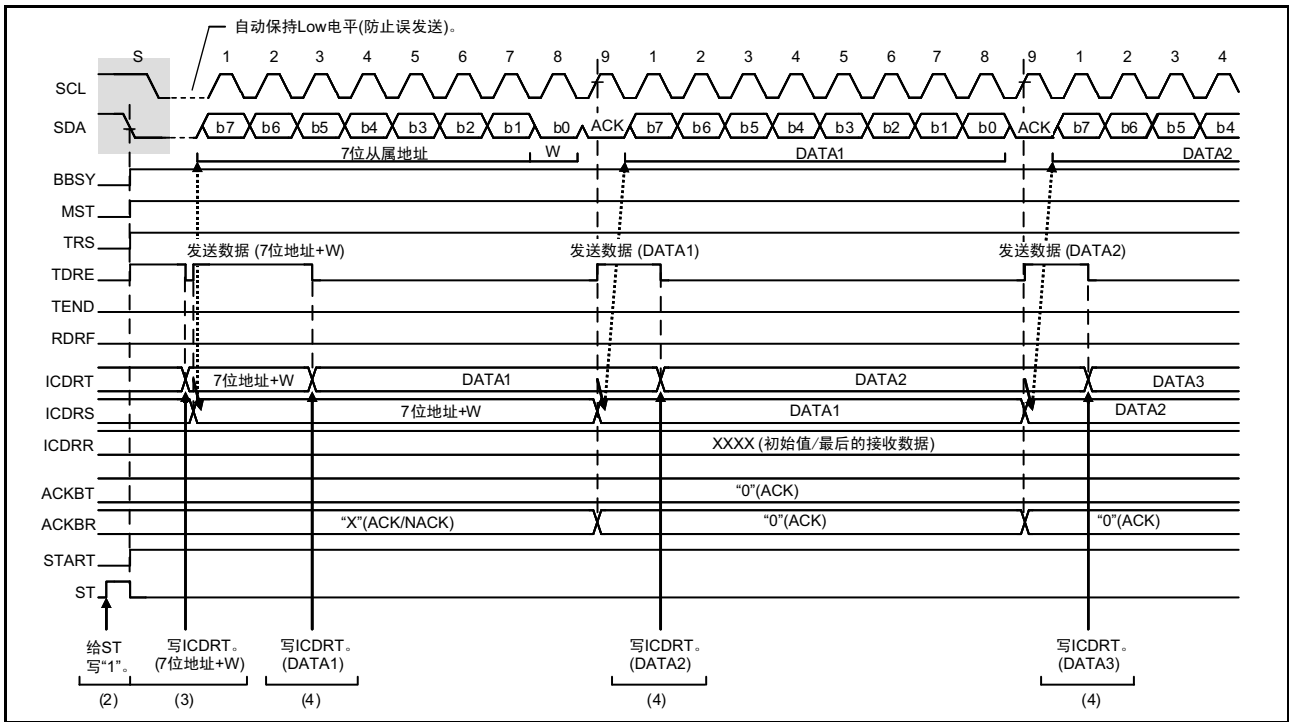


图 28.7 主控发送模式的运行时序 (1) (7 位地址格式的情况)

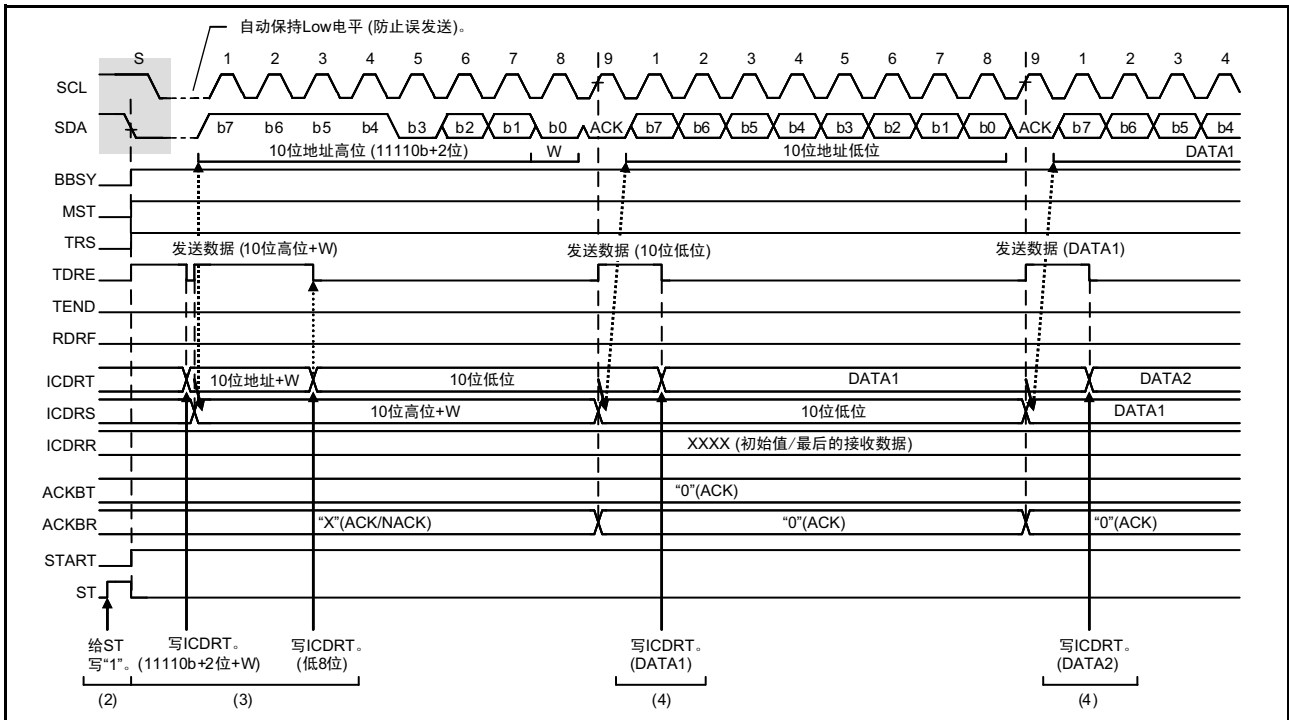


图 28.8 主控发送模式的运行时序 (2) (10 位地址格式的情况)

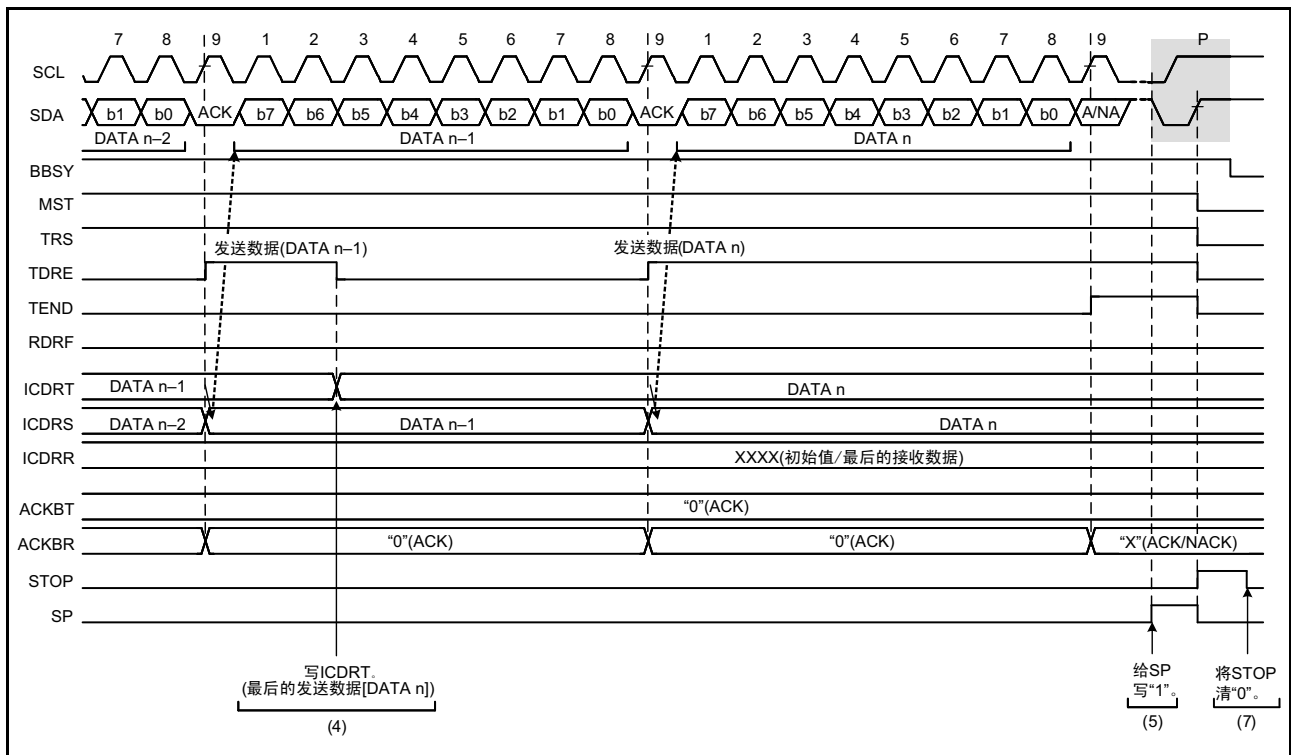


图 28.9 主控发送模式的运行时序 (3)

28.3.4 主控接收

在**主控接收模式**中，主控设备的**RIIC**输出**SCL**时钟，从属设备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设备，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28.10和图28.11～图28.13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”（**SCL**引脚和**SDA**引脚为非驱动状态）的状态下将**ICCR1.IICRST**位置“1”（**RIIC**复位）后，将**ICCR1.ICE**位置“1”（内部复位）。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（ $y=0\sim 2$ ）并且必须根据需要进行其他的寄存器（有关**RIIC**的初始设定请参照图28.5）。必须在设定完所需的寄存器后将**ICCR1.IICRST**位置“0”（解除复位）。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**信号和**SDA**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**TRIS**位为“1”而自动变为“1”。
3. 必须在确认**ICSR2.TDRE**标志是“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**TRIS**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设备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**标志是“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。另外，当下一个接收字节为最后字节-1时，必须在读**ICDRR**寄存器（第(最后字节-2)个字节）前将**ICMR3.WAIT**位置“1”（有**WAIT**）。因此，即使在将**ICMR3.ACKBT**位置“1”（**NACK**）的后续步骤6处理迟于其他中断等的情况下，也能在最后字节时输出**NACK**，并且同时能在接收最后字节时的第9个时钟的下降沿将**SCL**线固定为**Low**电平，进入能发行停止条件的状态。
6. 在**ICMR3.RDRFS**位为“0”并且需要将“下一个数据接收后通信结束”通知从属设备时，必须将**ICMR3.ACKBT**位置“1”（**NACK**）。
7. 在读**ICDRR**寄存器（第(最后字节-1)个字节）后，必须先确认**ICSR2.RDRF**标志是“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**标志是“1”后，必须将**ICSR2.NACKF**和**ICSR2.STOP**标志置“0”，以便进行下一次的通信。

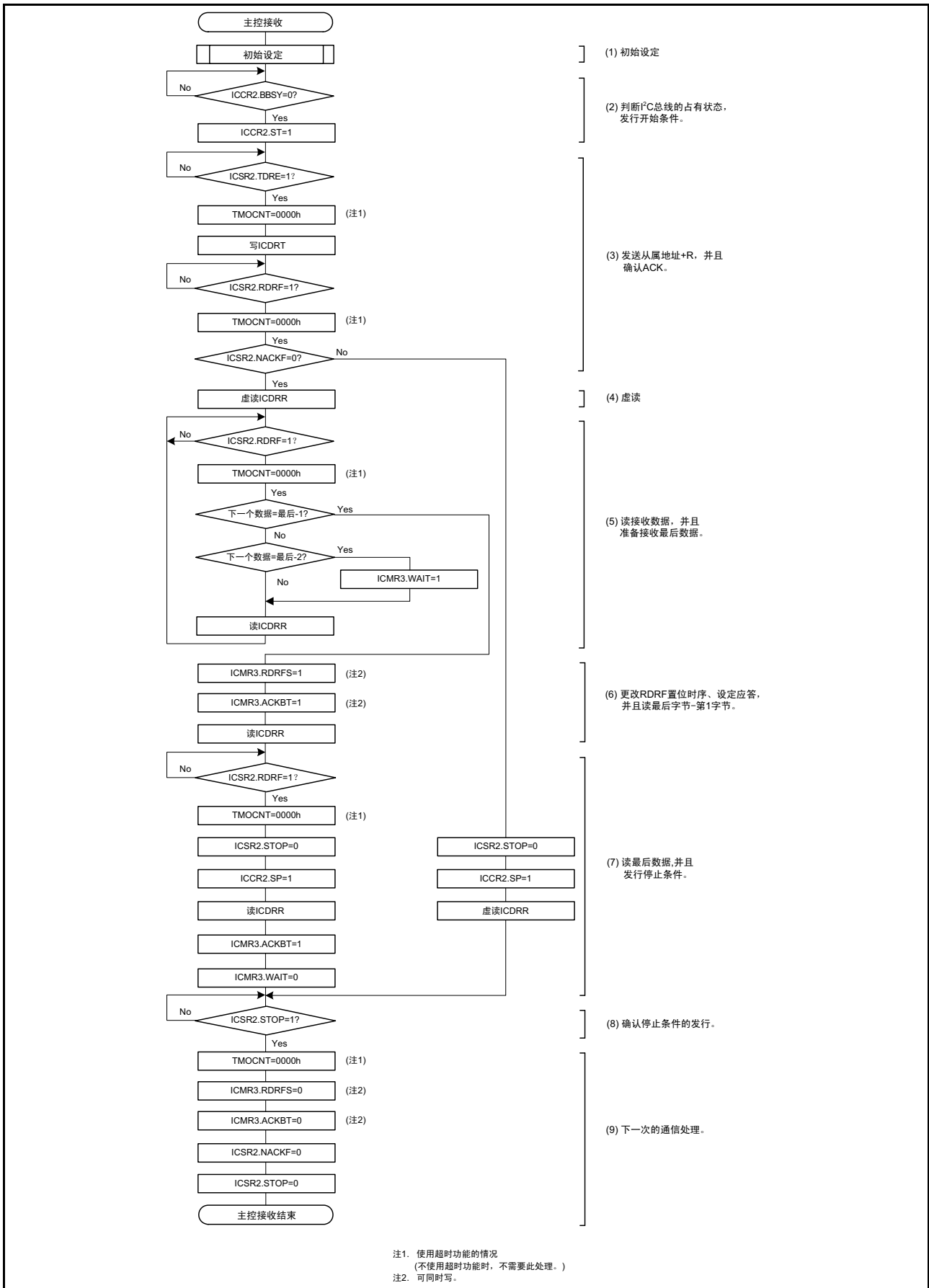


图 28.10 主控接收模式的流程图例子 (7 位地址格式的情况)

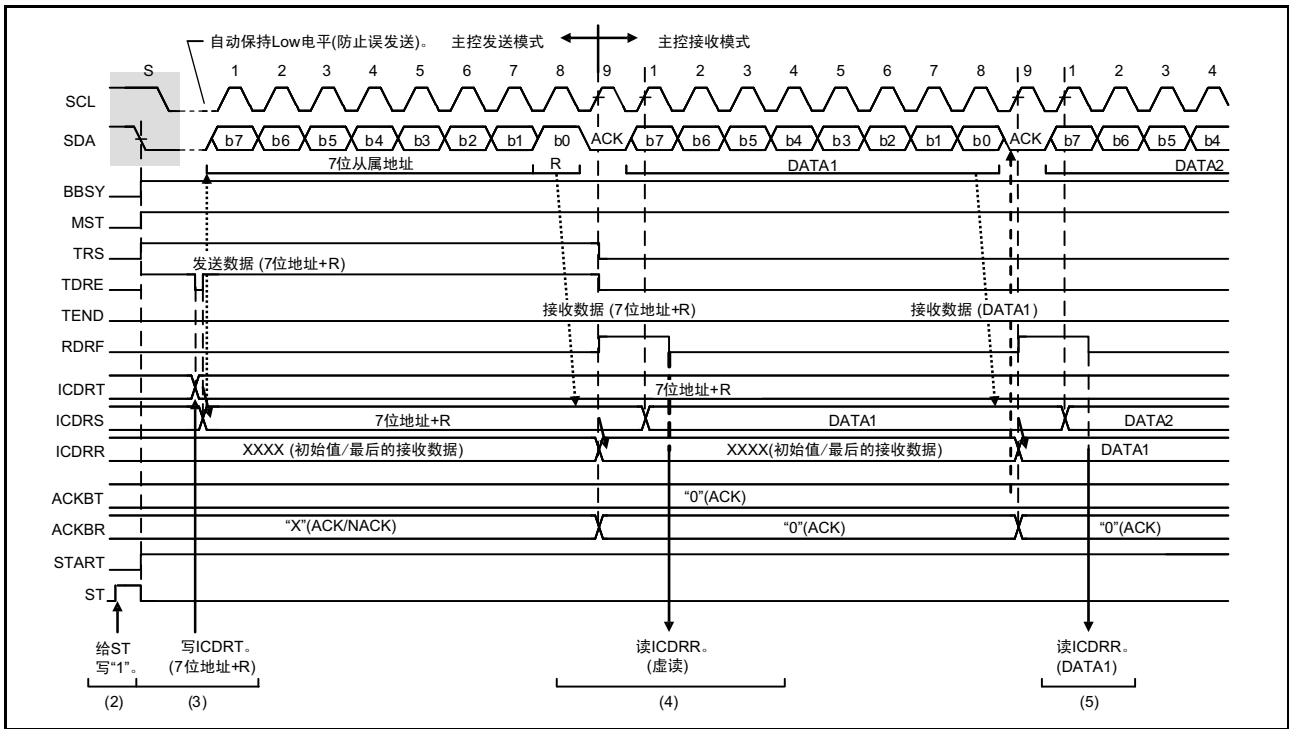


图 28.11 主控接收模式的运行时序 (1) (7 位地址格式并且 RDRFS 位为“0”的情况)

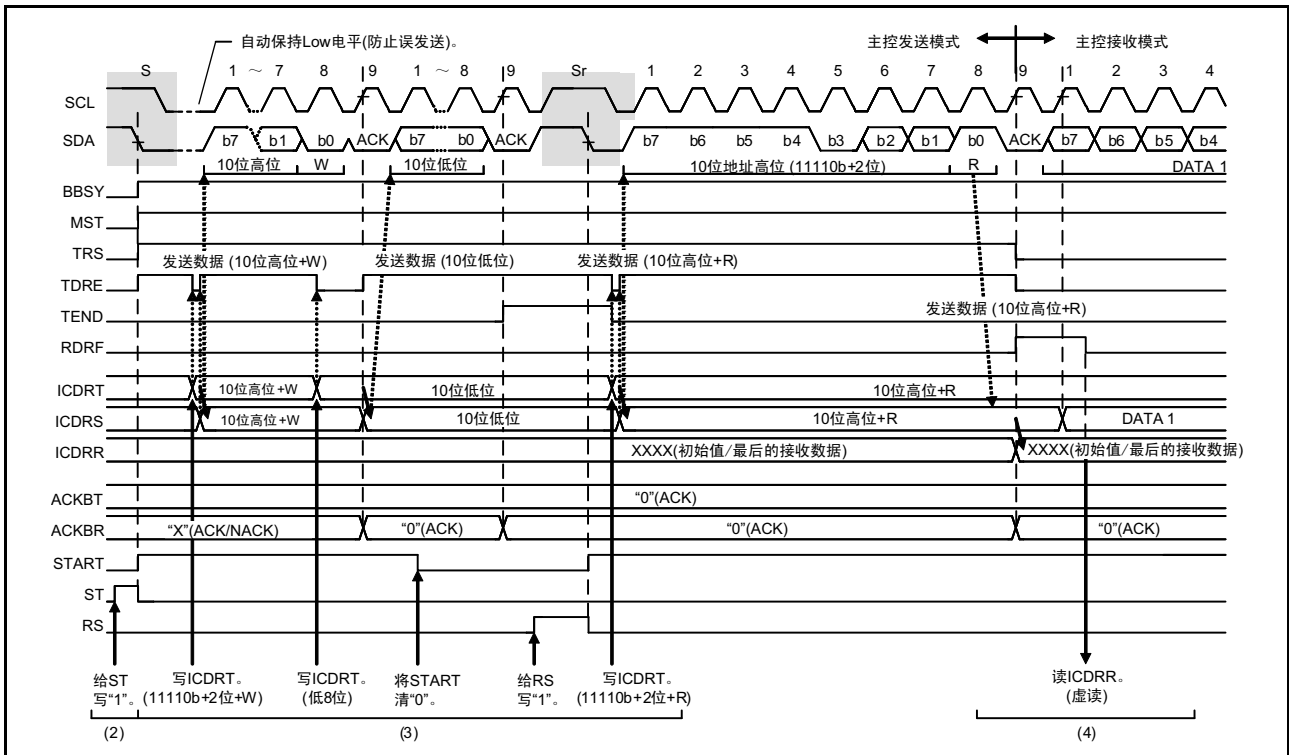


图 28.12 主控接收模式的运行时序 (2) (10 位地址格式并且 RDRFS 位为“0”的情况)

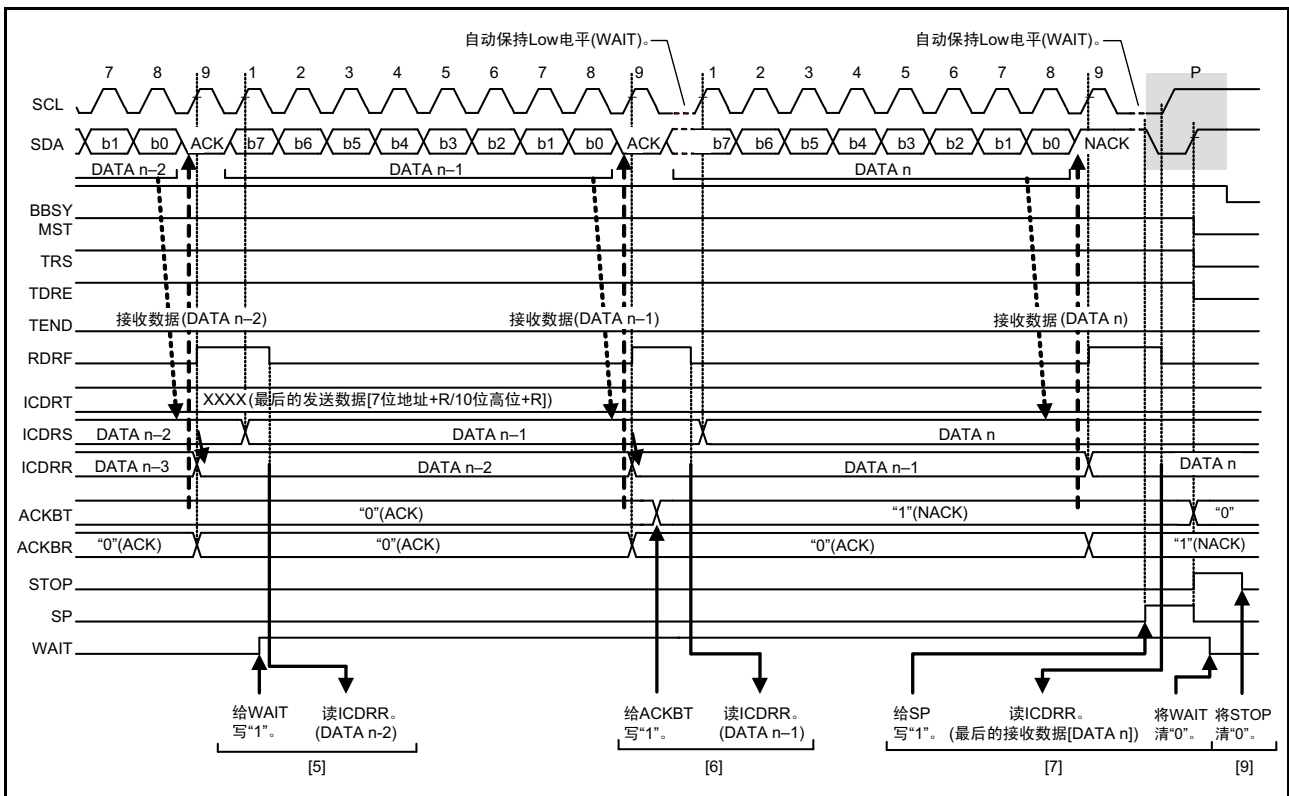


图 28.13 主控接收模式的运行时序 (3) (RDRFS 位为“0”的情况)

28.3.5 从属发送

在从属发送模式中，主控设备输出 SCL 时钟，从属设备的 RIIC 发送数据，并且主控设备返回应答。

从属发送模式的使用例子和运行时序分别如图 28.14 和图 28.15、图 28.16 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 28.5 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 标志是“1”后将发送数据写到 ICDRT 寄存器。此时，如果在 ICFER.NACKF 位为“1”的状态下主控设备没有应答（接收到 NACK），RIIC 就中止下一次的通信。
4. 在 ICSR2.NACKF 标志变为“1”或者将最后的发送数据写到 ICDRT 寄存器后，必须在 ICSR2.TDRE 标志为“1”的状态下等待 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 SCL 线保持为 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 SCL 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 标志是“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。

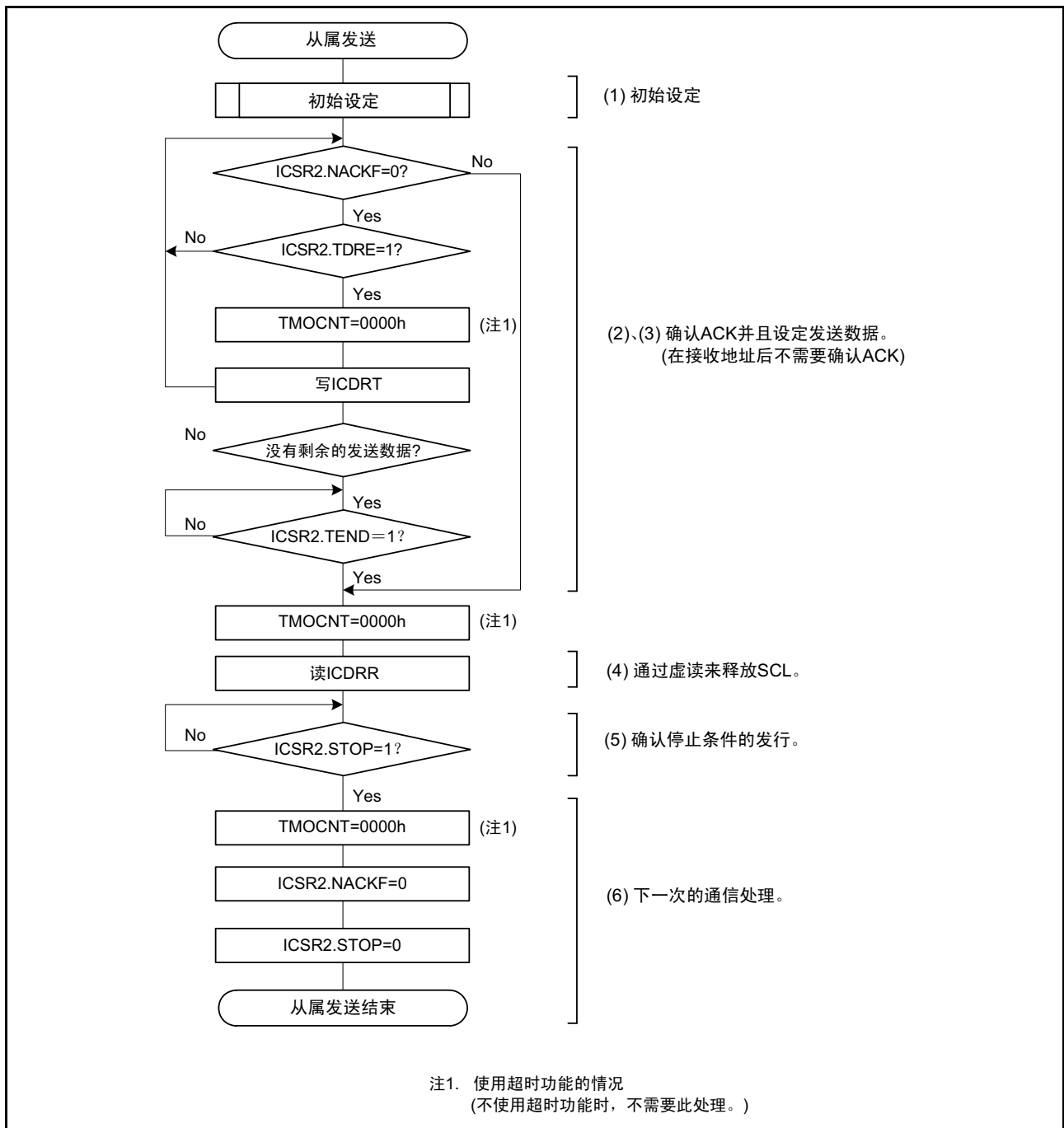


图 28.14 从属发送模式的流程图例子

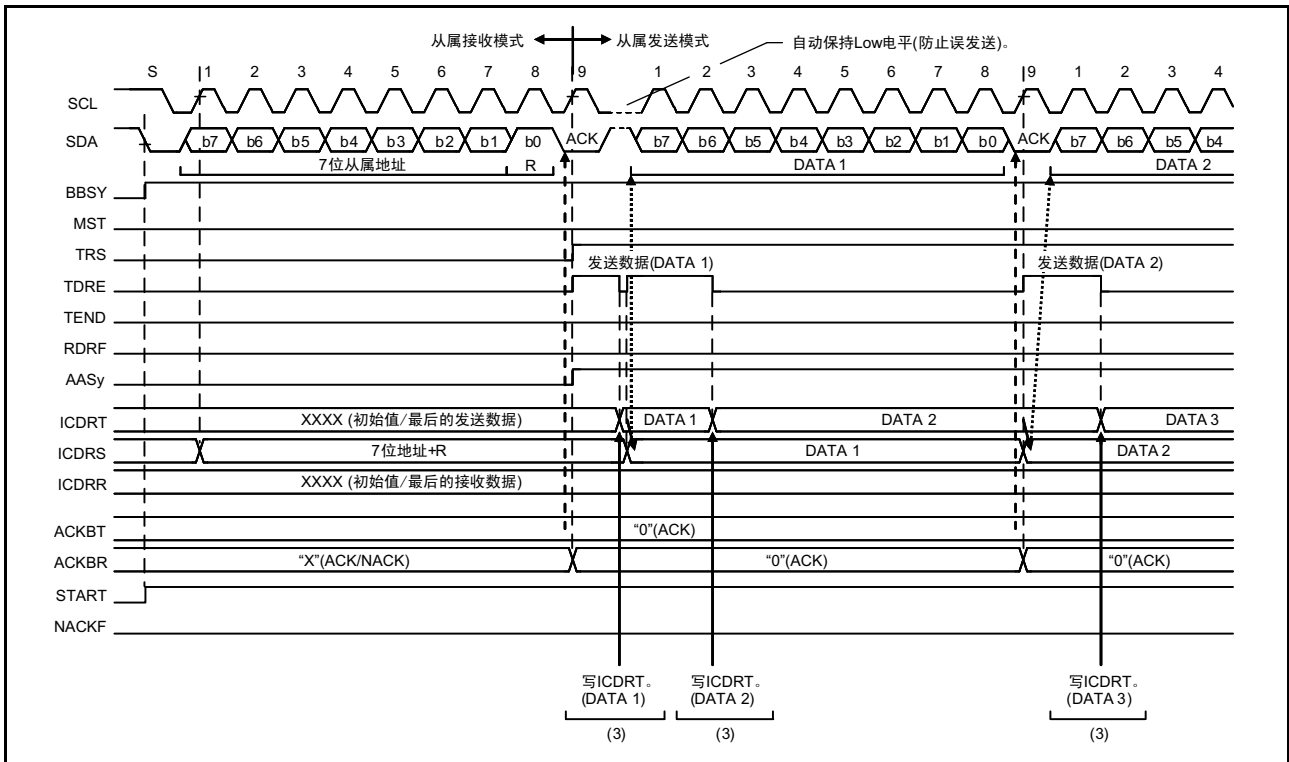


图 28.15 从属发送模式的运行时序 (1) (7 位地址格式的情况)

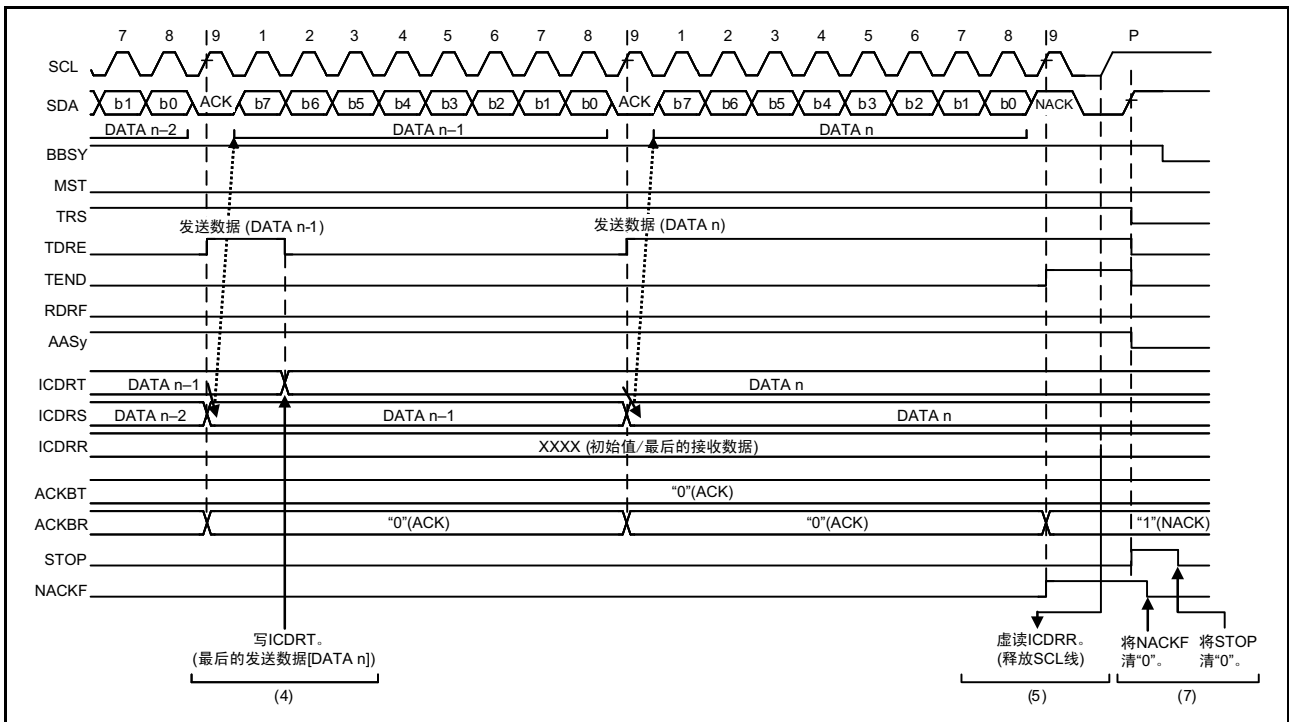


图 28.16 从属发送模式的运行时序 (2)

28.3.6 从属接收

在从属接收模式中，主控设备输出 SCL 时钟和发送数据，从属设备的 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 28.17 和图 28.18、图 28.19 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 28.5 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 位或者 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 标志是“0”并且 ICSR2.RDRF 标志是“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 SCL 线保持为 Low 电平。通过读 ICDRR 寄存器来解除被保持的 Low 电平，RIIC 释放 SCL 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 标志是“1”后，必须将 ICSR2.STOP 标志置“0”，以便进行下一次的通信。

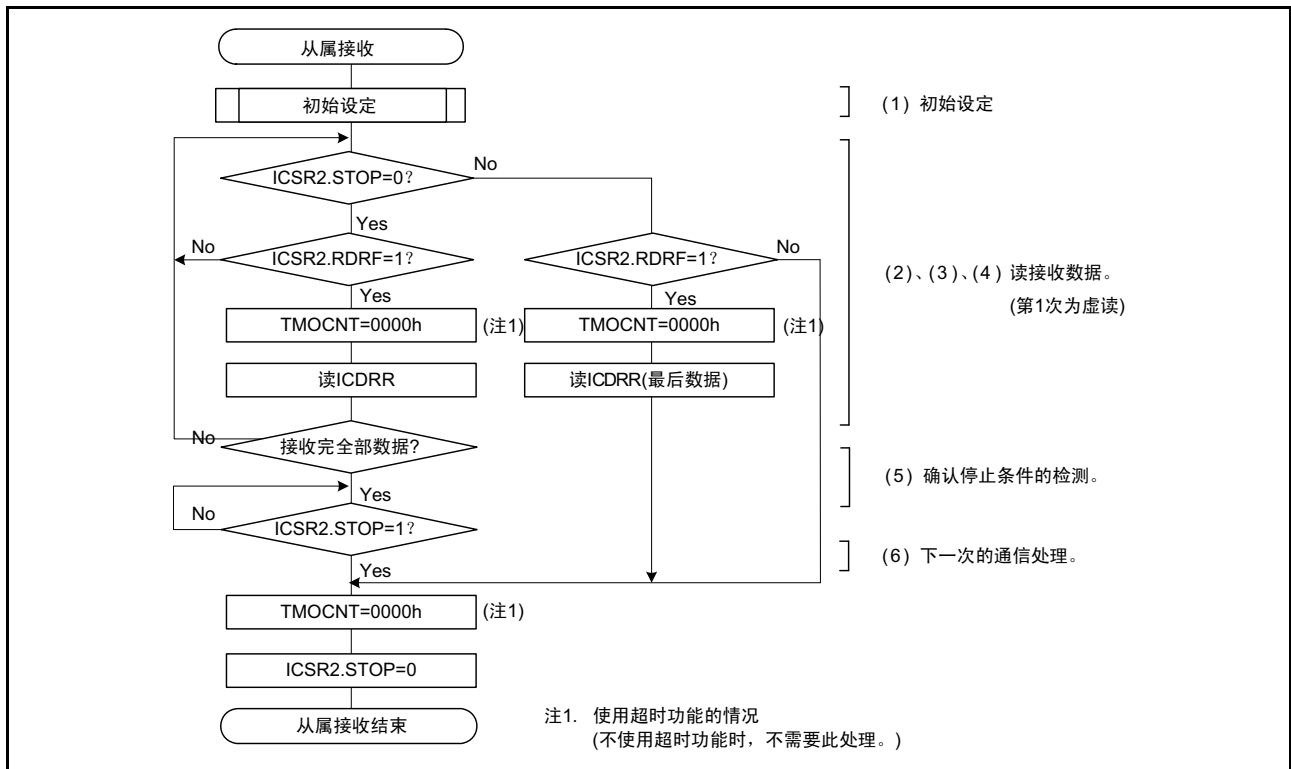


图 28.17 从属接收模式的流程图例子

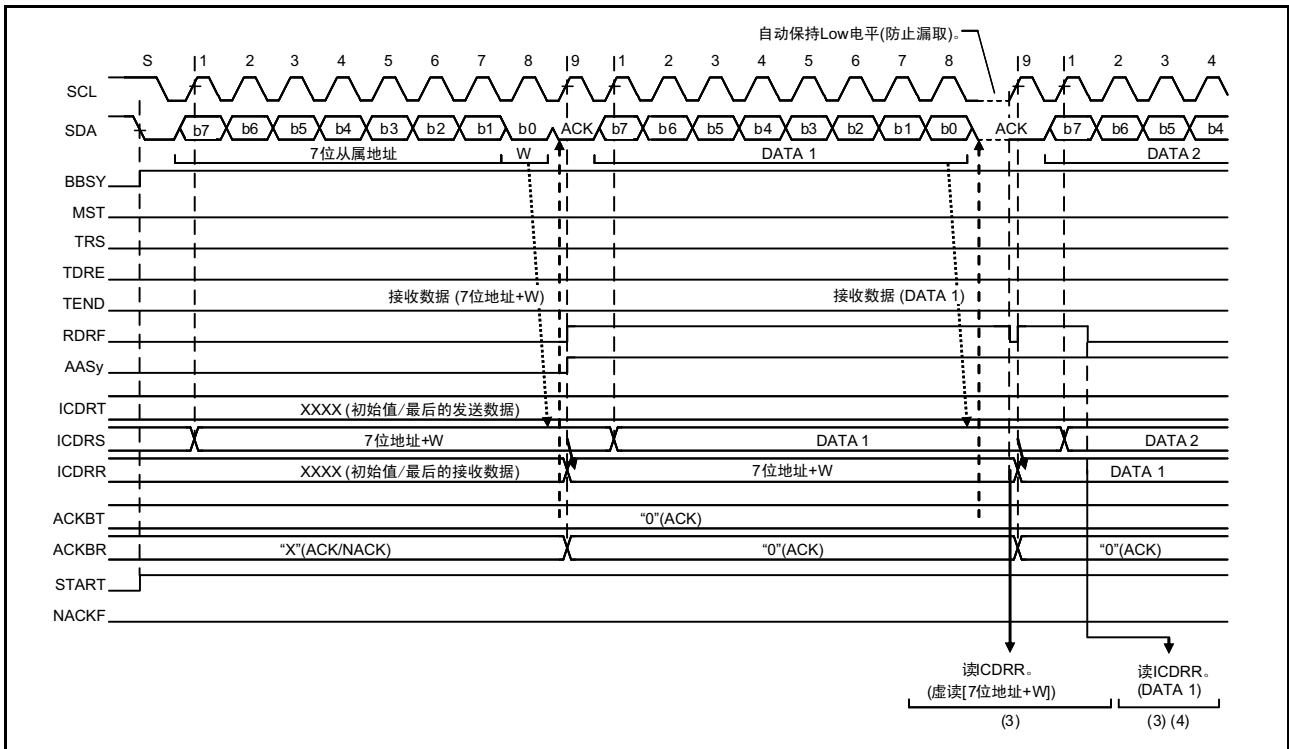


图 28.18 从属接收模式的运行时序 (1) (7 位地址格式并且 RDRFS 位为“0”的情况)

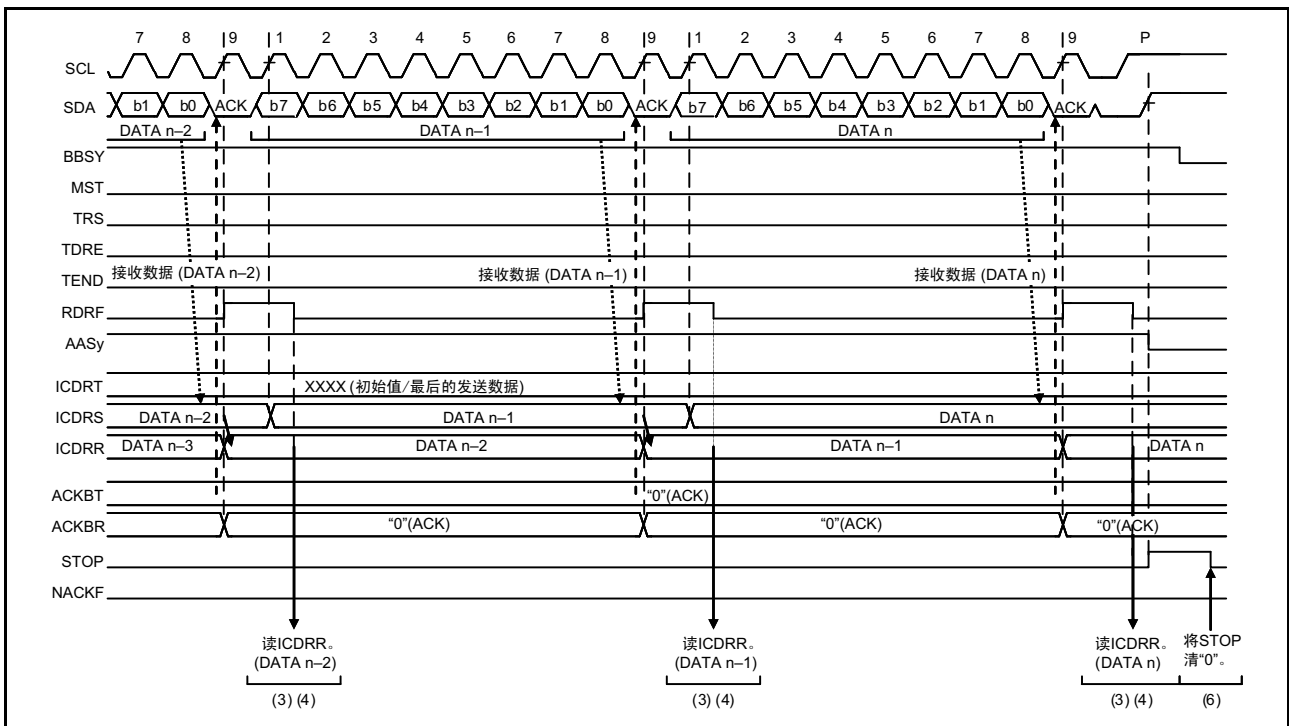


图 28.19 从属接收模式的运行时序 (2) (RDRFS 位为“0”的情况)

28.4 SCL 同步电路

如果 RIIC 的 SCL 时钟生成电路检测到 SCL 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 SCL 线驱动为 Low 电平。如果检测到 SCL 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 SCL 线的 Low 电平驱动, 然后释放 SCL 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设备的竞争而使 SCL 时钟和 SCL 时钟发生冲突。如果 SCL 时钟发生冲突, 主控设备就需要与 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 SCL 线, 并且一边逐位取得同步, 一边生成 SCL 时钟。

RIIC 检测到 SCL 线的上升沿并且在对 ICBRH 寄存器设定的 High 电平进行计数的过程中, 如果因其他主控设备的 SCL 时钟输出而使 SCL 线下降, RIIC 就在检测到 SCL 线的下降沿时中止 High 电平宽度的递增计数, 并且在 SCL 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 SCL 线的 Low 电平驱动, 然后释放 SCL 线。此时, 如果其他主控设备的 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。当其他主控设备结束 Low 电平输出时, 释放 SCL 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。

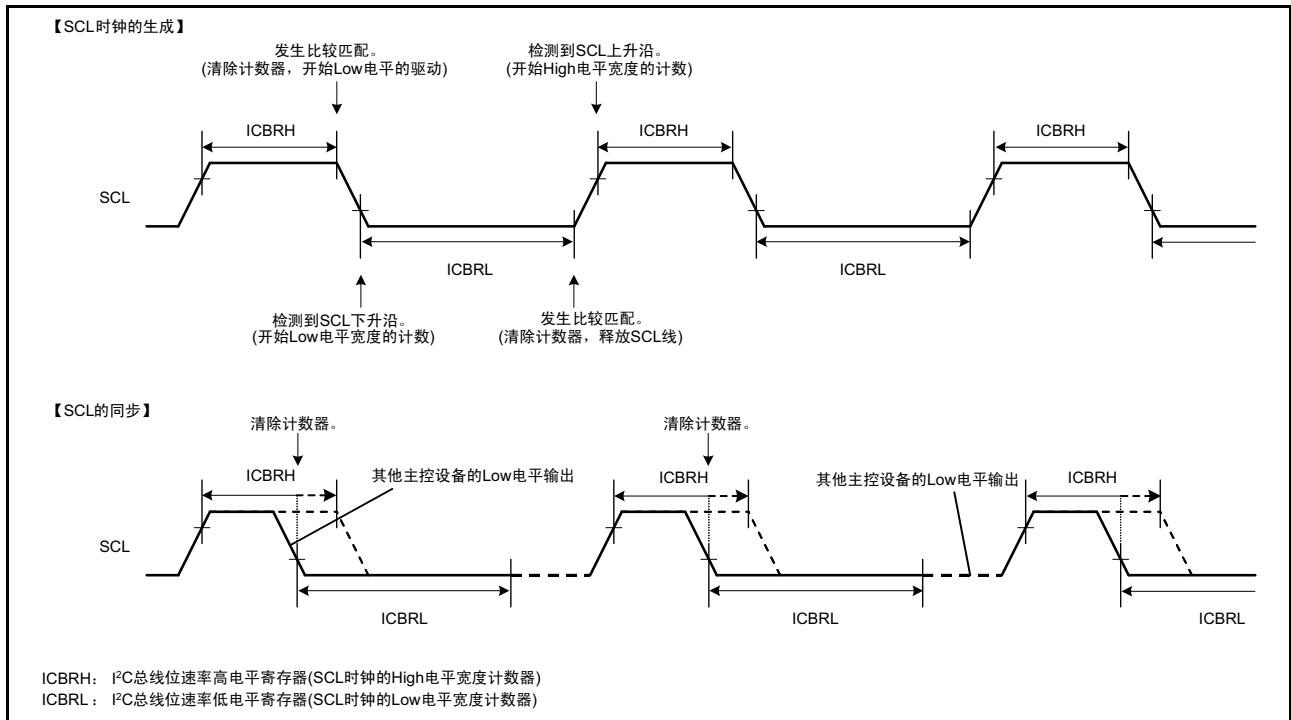


图 28.20 RIIC 的 SCL 时钟生成和 SCL 同步

28.5 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）。

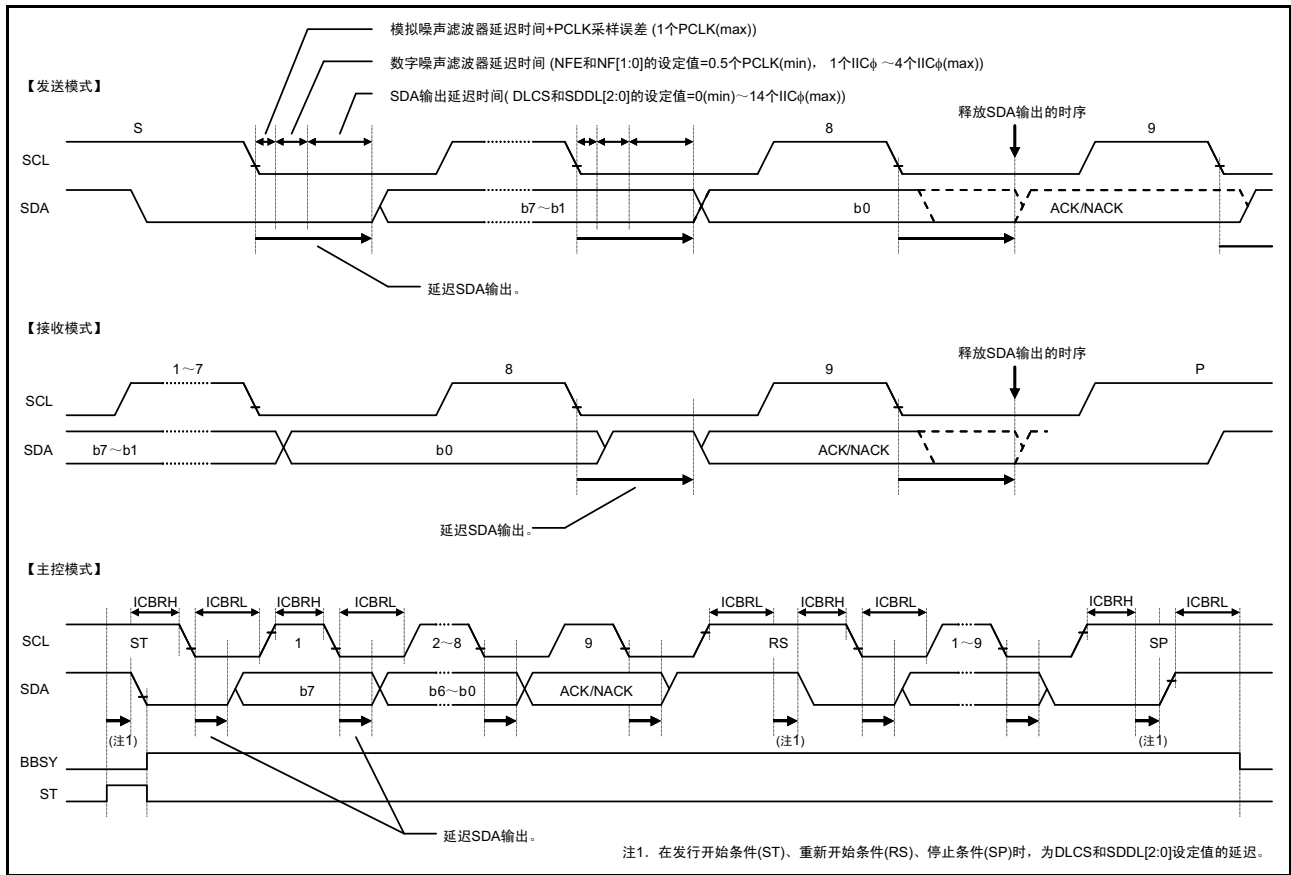


图 28.21 SDA 输出延迟时序

28.6 数字噪声滤波器电路

SCL 引脚和 SDA 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 28.22 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 IIC ϕ 周期。

在 IIC ϕ 的下降沿对 SCL 引脚的输入信号（或者 SDA 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。

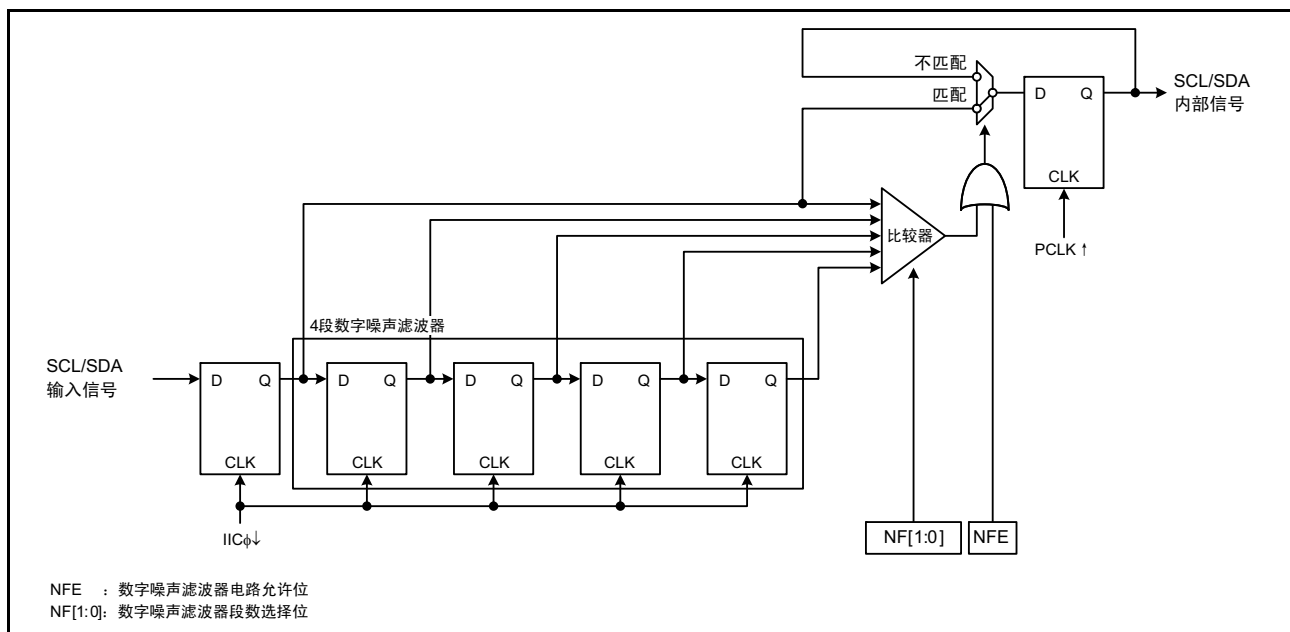


图 28.22 数字噪声滤波器电路的框图

28.7 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。

28.7.1 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 IC_{SER}.SAR_yE 位 (y=0 ~ 2) 为“1”时，能检测到 SAR_{Uy}/SAR_{Ly} 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 IC_{SR1}.AAS_y 标志 (y=0 ~ 2) 置“1”，然后根据后续的 R/W# 位将 IC_{SR2}.RDRF 标志或者 IC_{SR2}.TDRE 标志置“1”。因此，能产生接收数据满中断 (IC_{RXI}) 或者发送数据空中断 (IC_{TXI})，并且能通过确认 AAS_y 标志判断指定了哪个从属地址。

AAS_y 标志变为“1”的时序如图 28.23 ~ 图 28.25 所示。

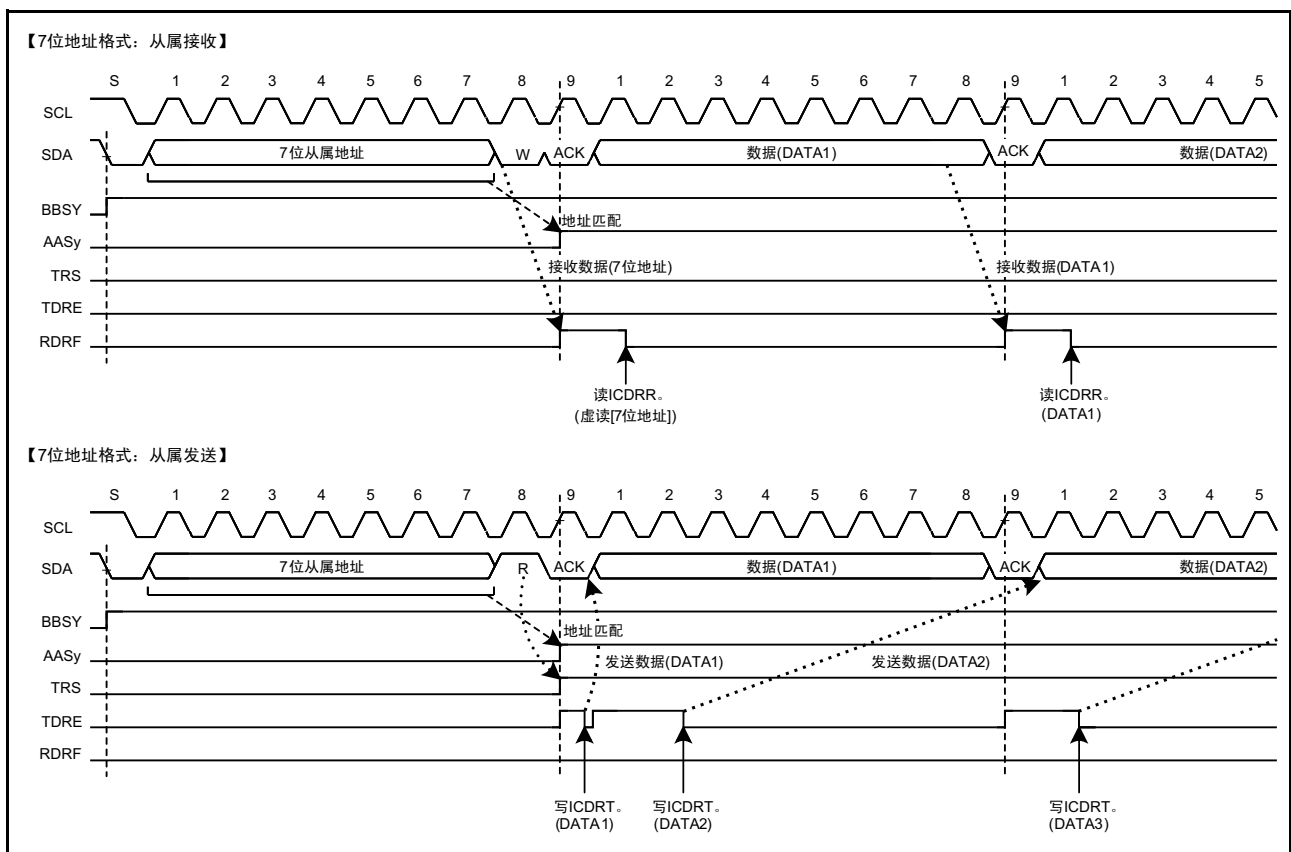


图 28.23 选择 7 位地址格式时的 AAS_y 标志变为“1”的时序

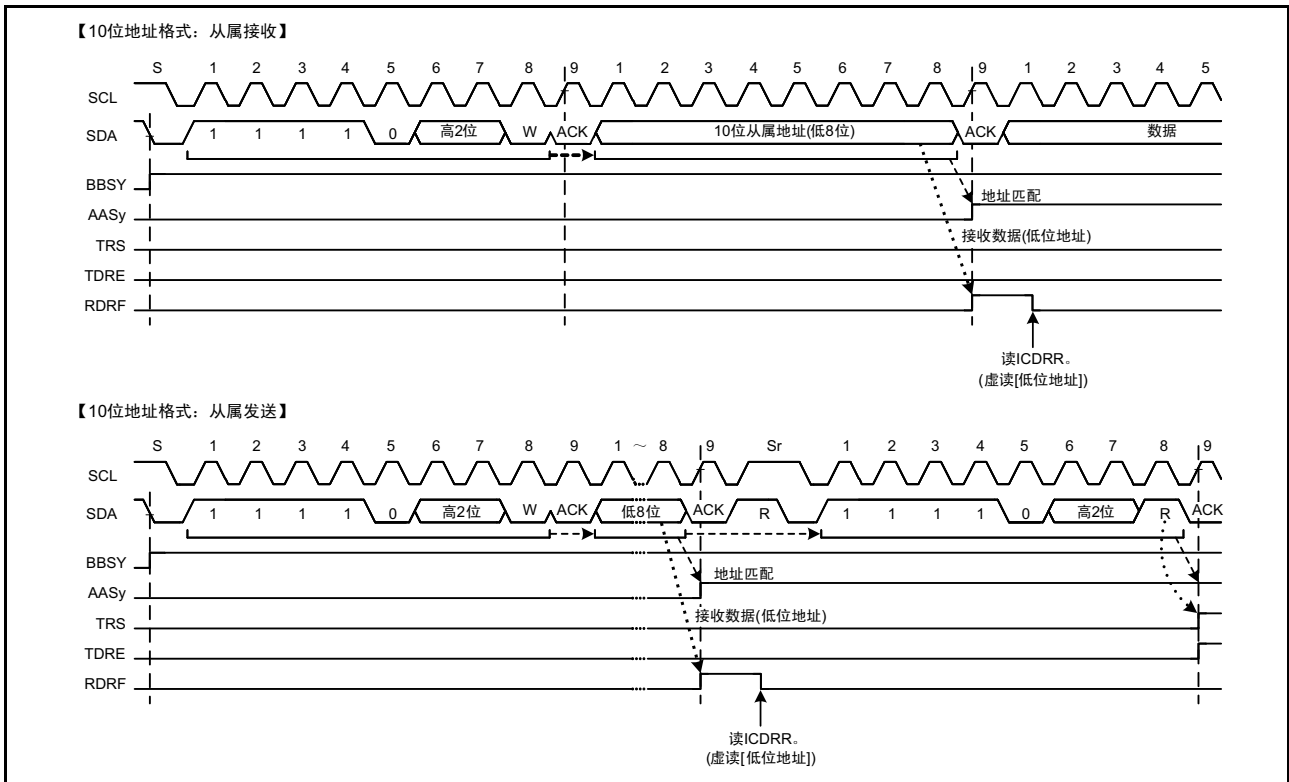


图 28.24 选择 10 位地址格式时的 AASy 标志变为“1”的时序

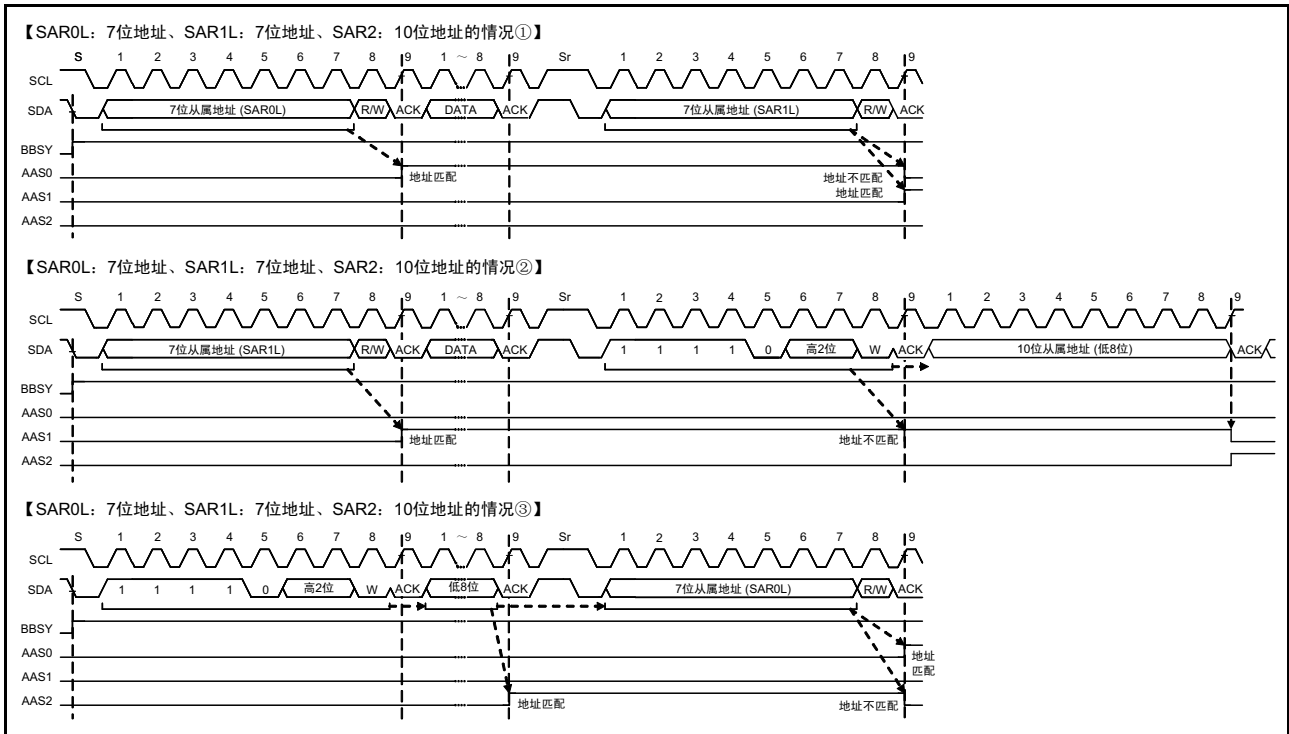


图 28.25 10 位地址格式同时存在时的 AASy 标志变为“1”/“0”的时序

28.7.2 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)，并且能通过确认 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。

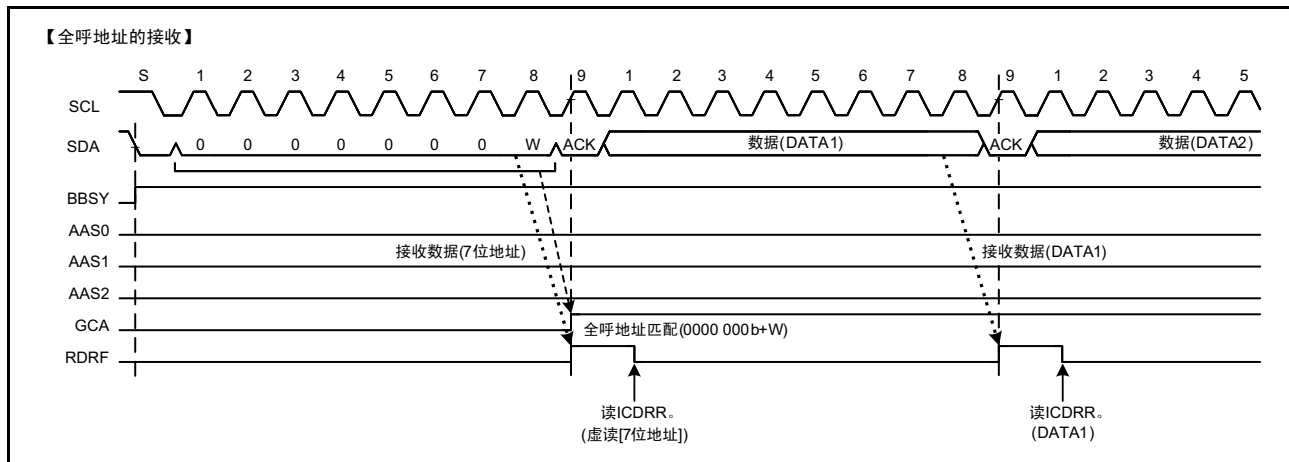


图 28.26 接收全呼地址时的 GCA 标志变为 “1” 的时序

28.7.3 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 是“1”后检查 DID 标志，能确认已接收到设备 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]) 和正常的发送数据一样，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。

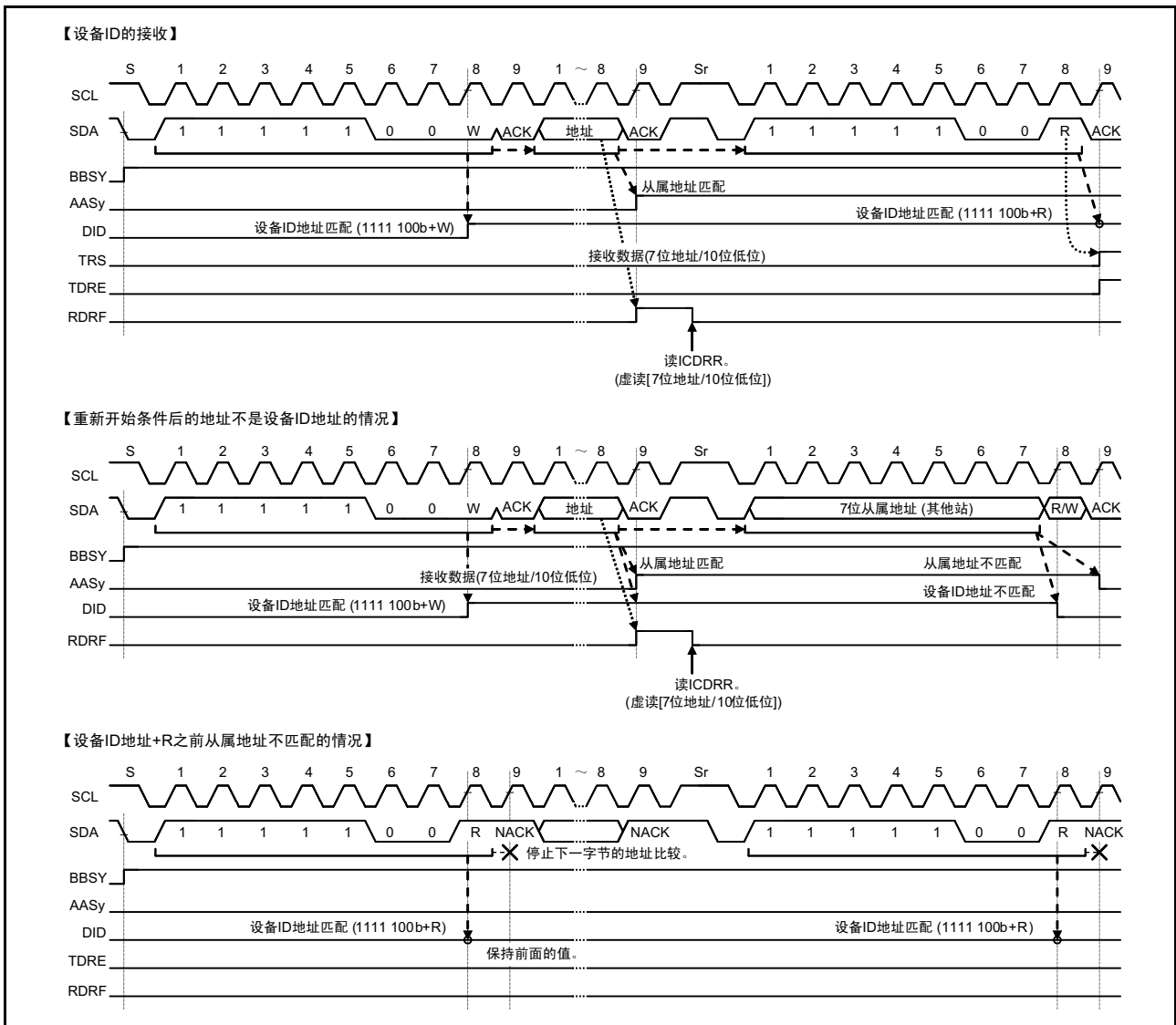


图 28.27 接收设备 ID 地址时的 AASy 标志和 DID 标志的置位 / 清除时序

28.7.4 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 IC SR1.HOA 标志置“1”，并且在 Wr 位的情况下 (R/W# 位接收到“0”)，将 IC SR2.RDRF 标志置“1”。因此，能产生接收数据满中断 (ICRXI)，并且能通过确认 HOA 标志，视为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 (R/W# 位接收到“1”)，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。

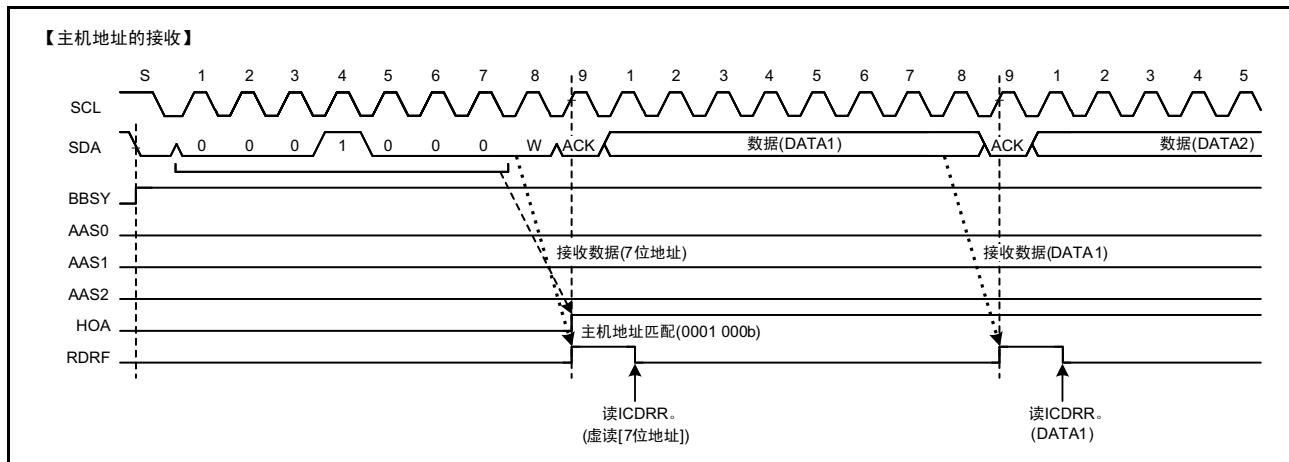


图 28.28 接收主机地址时的 HOA 标志变为“1”的时序

28.8 SCL 的 Low 电平自动保持功能

28.8.1 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 SCL 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

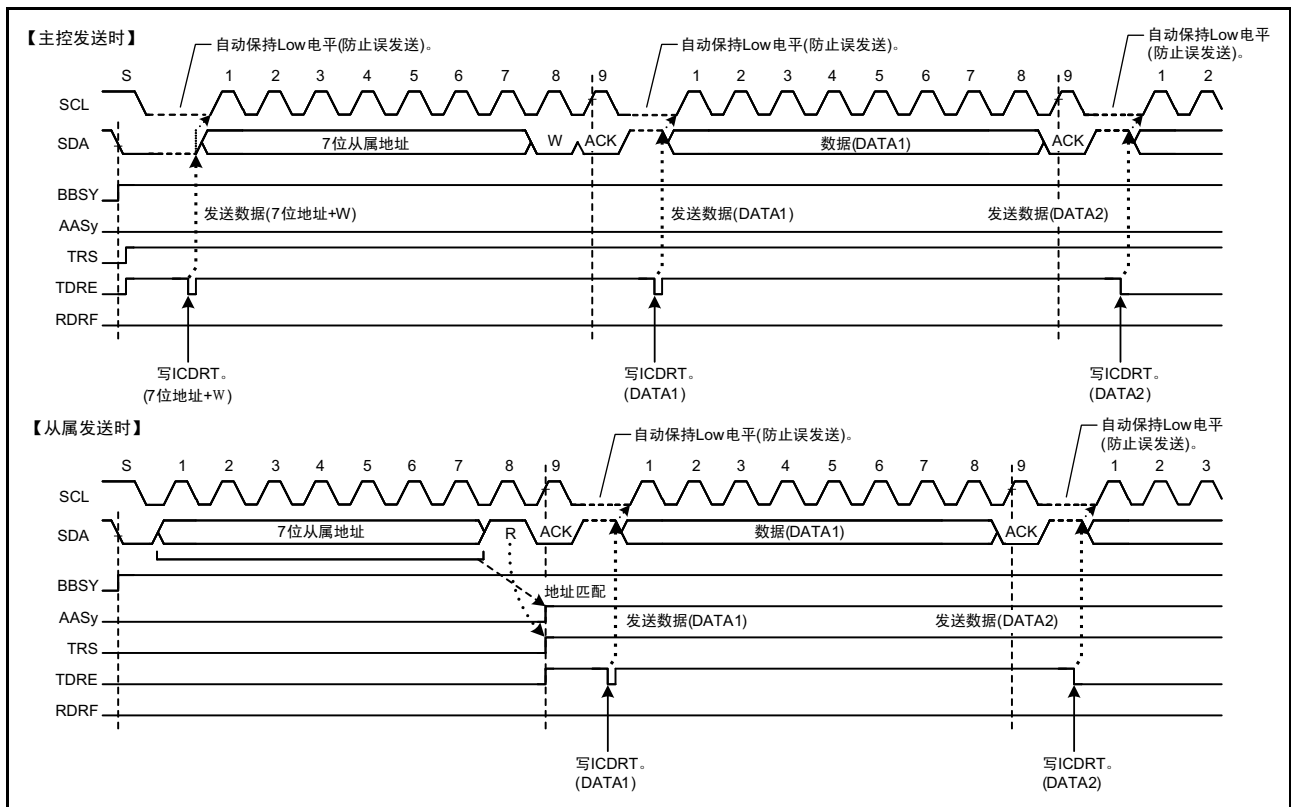


图 28.29 发送模式的 Low 电平自动保持运行

28.8.2 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 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 =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”, 然后通过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。

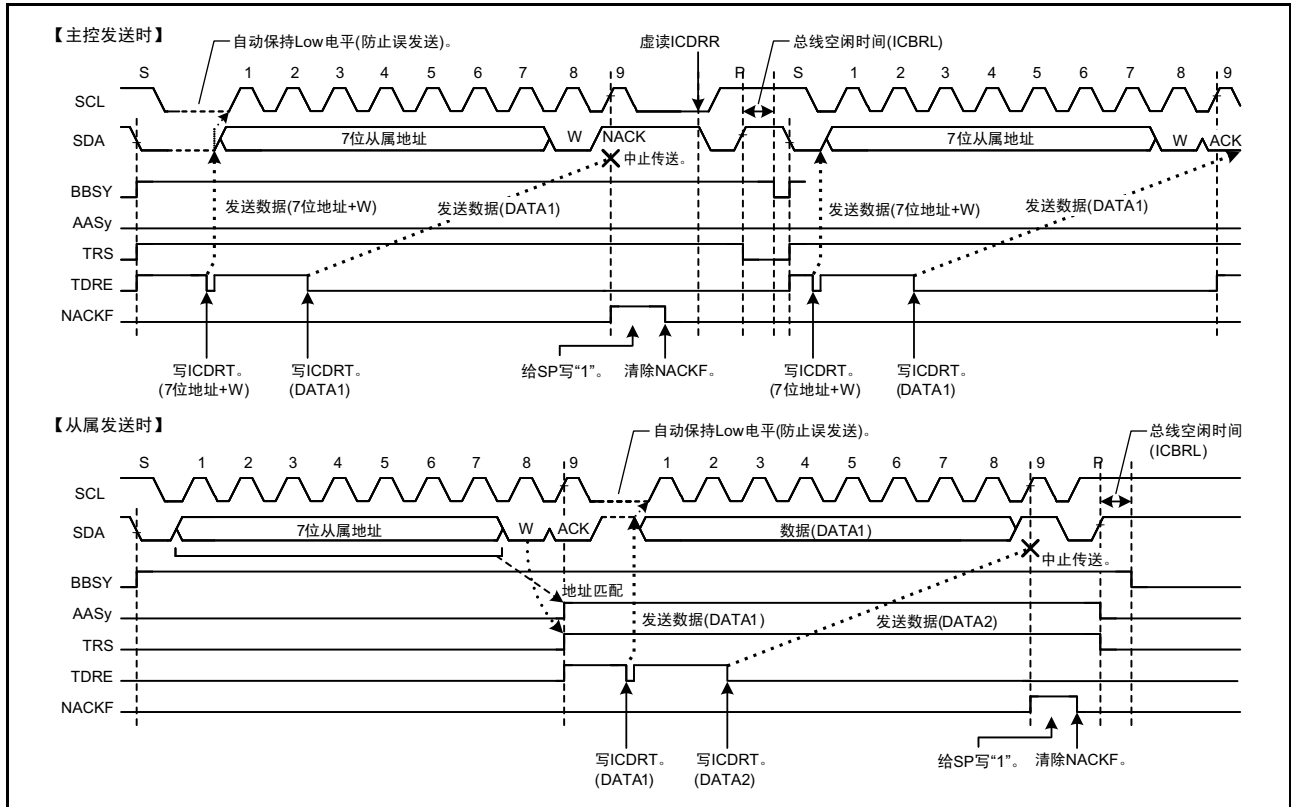


图 28.30 接收 NACK 时的传送中止 (NACKE 位为 “1”)

28.8.3 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 SCLn 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 电平的自动保持功能

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 SCL 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 个时钟的上升沿。如果检测到第 8 个时钟的下降沿，就自动保持 SCL 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。

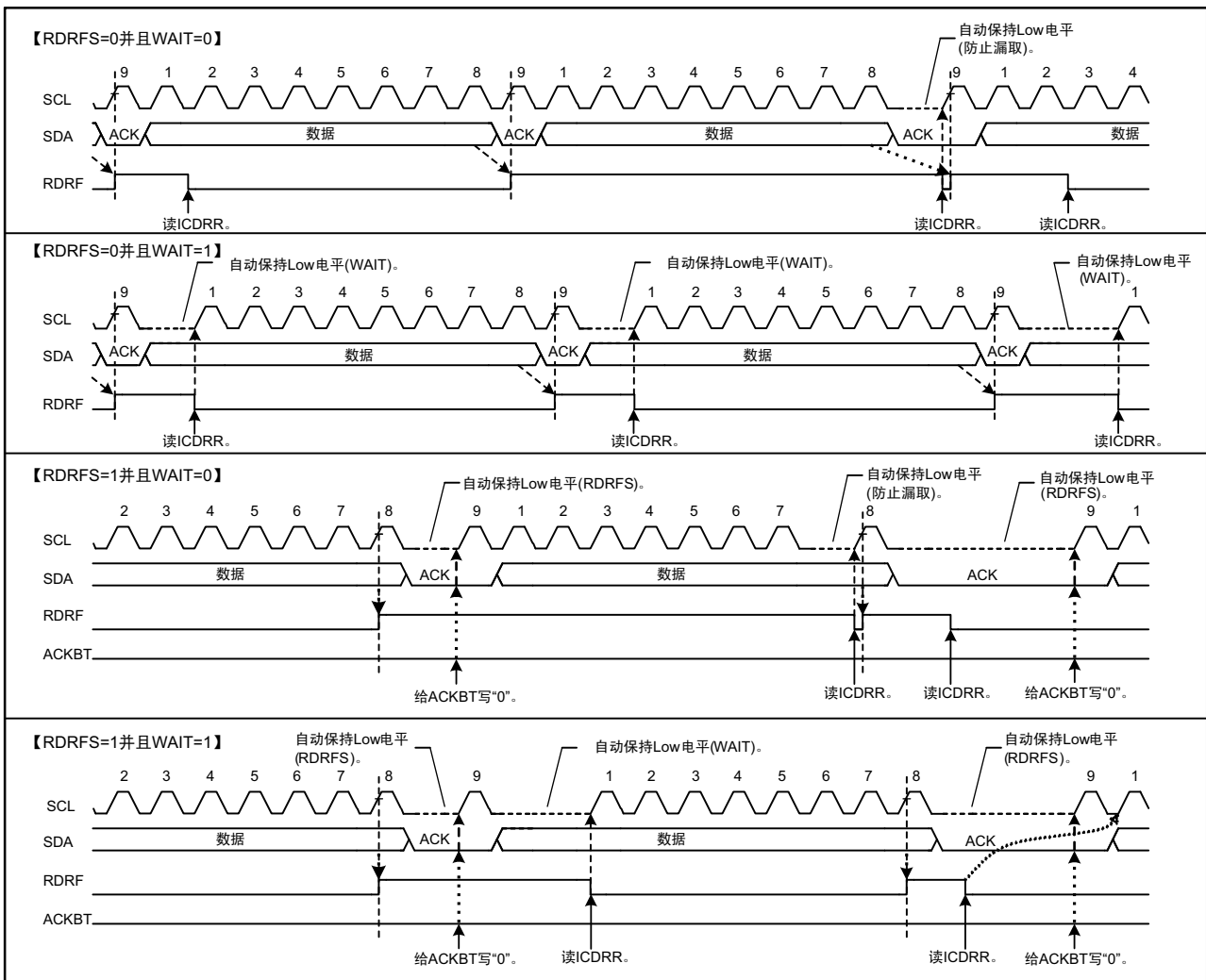


图 28.31 接收模式的 Low 电平自动保持（RDRFS 位和 WAIT 位）

28.9 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。

28.9.1 主控仲裁失败检测功能 (MALE 位)

在发行开始条件时，RIIC 将 SDA 线置为 Low 电平。但是，如果其他主控设备提前发行开始条件后将 SDA 线置为 Low 电平，就产生仲裁失败并且优先其他主控设备的通信。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设备的通信。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）和 SDA 线上的信号状态不同（自身发送的 SDA 输出为 High 电平输出（=SDA 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 信号和 SDA 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）和 SDA 线上的信号状态不同时

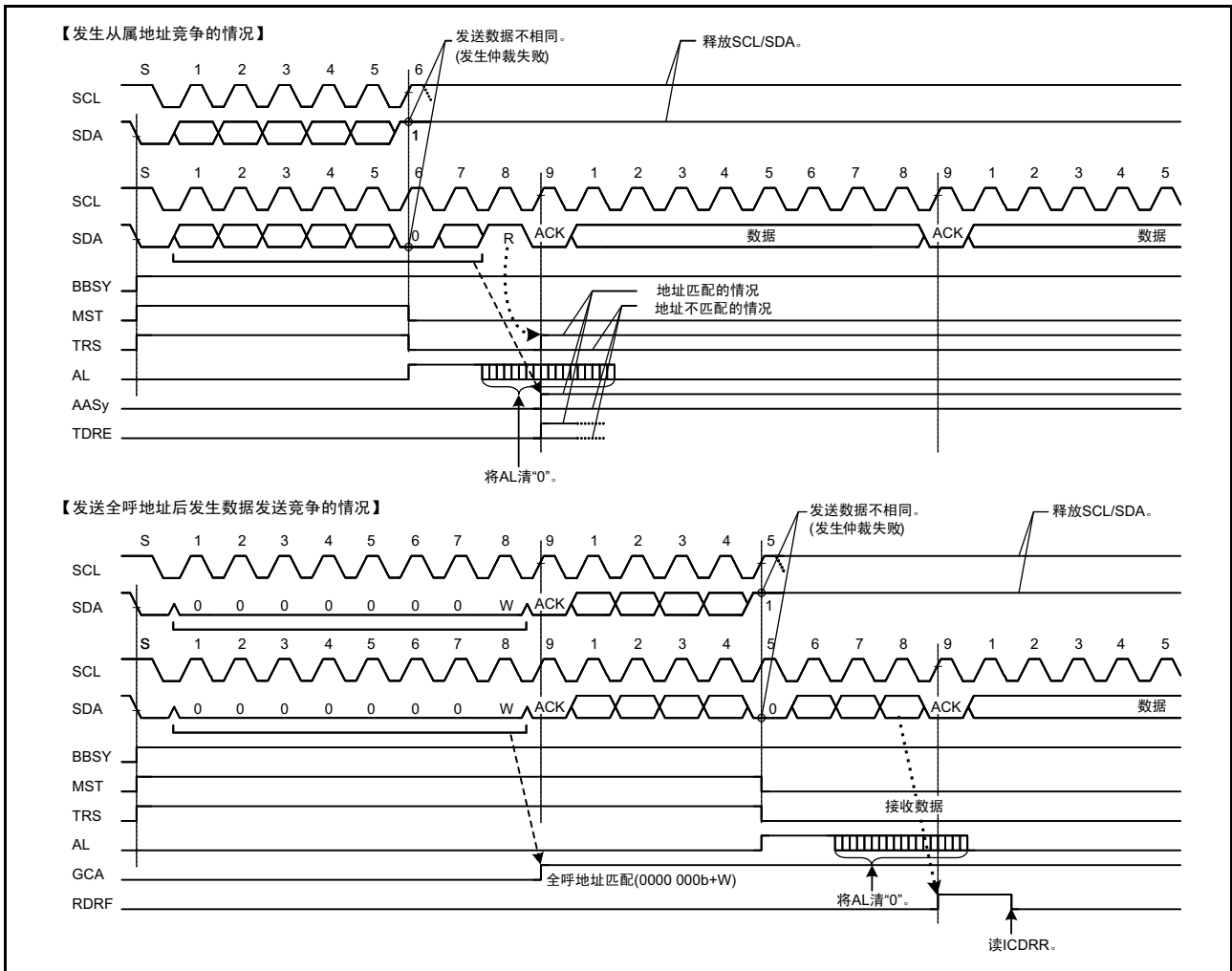


图 28.32 主控仲裁失败检测的运行例子 (MALE 位为“1”的情况)

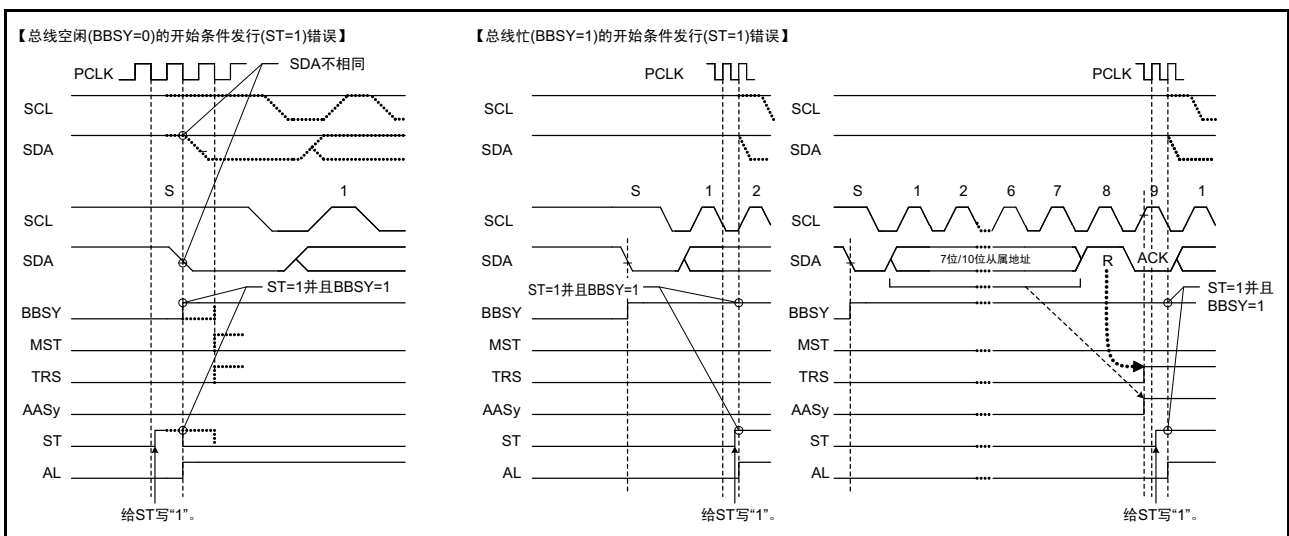


图 28.33 发行开始条件时的仲裁失败 (MALE 位为“1”的情况)

28.9.2 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 信号和 SDA 线上的信号状态不同（自身发送的 SDA 输出为 High 电平输出 (=SDA 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。此仲裁失败功能主要在多主控系统中 2 个或者 2 个以上的主控设备同时从同一个从属设备接收数据时发生 NACK 发送和 ACK 发送的冲突。这种情况发生在 2 个或者 2 个以上的主控设备通过 1 个从属设备交换共同信息时。NACK 发送仲裁失败检测的运行例子如图 28.34 所示。

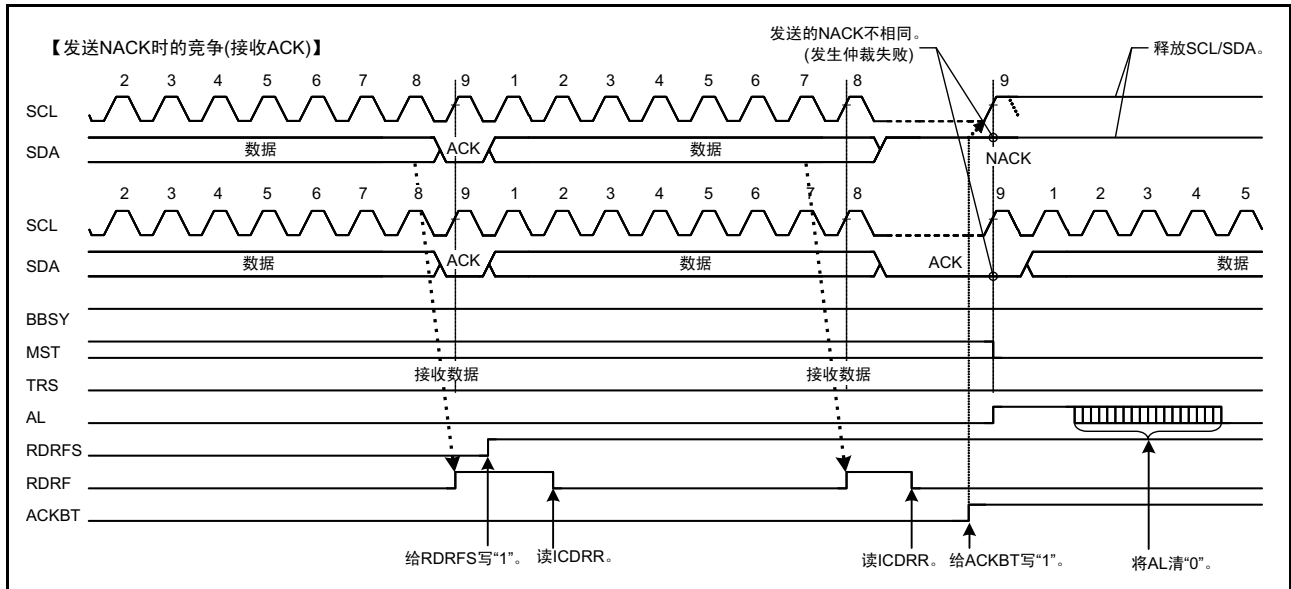


图 28.34 NACK 发送仲裁失败检测的运行例子 (NALE 位为“1”的情况)

以 2 个主控设备（主控 A、主控 B）和 1 个从属设备连接总线的情况为例进行说明。假设主控 A 从从属设备接收 2 字节的数据，主控 B 从从属设备接收 4 字节的数据。

在主控 A 和主控 B 同时存取从属设备的情况下，因为从属地址相同，所以主控 A 和主控 B 在存取从属设备时不发生仲裁失败，因此主控 A 和主控 B 都取得了总线权。然后，主控 A 在从从属设备接收完最后 2 字节数据时发送 NACK，主控 B 因为没有从从属设备接收到需要的 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设备的竞争失败，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 信号和 SDA 线上的信号状态不同（接收到 ACK）。

28.9.3 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) 和 SDA 线上的信号状态不同 (自身发送的 SDA 输出为 High 电平输出 (=SDA 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) 和 SDA 线上的信号状态不同。

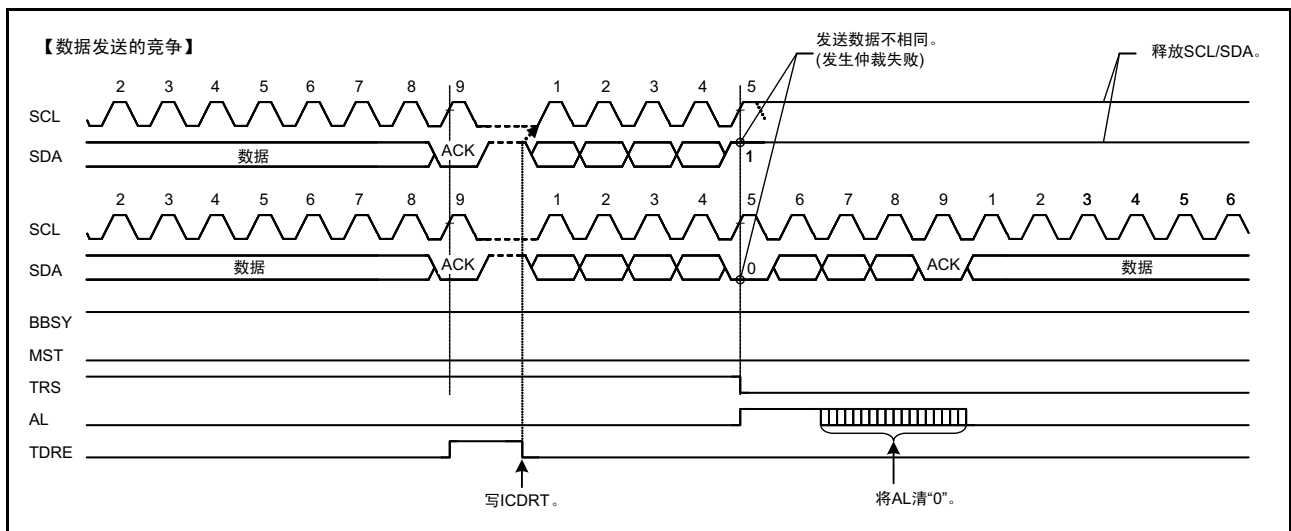


图 28.35 从属仲裁失败检测的运行例子 (SALE 位为 “1” 的情况)

28.10 开始条件、重新开始条件和停止条件的发行功能

28.10.1 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 SDA 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 SCL 线下降（从 High 电平变为 Low 电平）。
4. 在检测到 SCL 线的 Low 电平后，确保 ICBRL 寄存器所设时间的 SCL 线的 Low 电平宽度。

28.10.2 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 SDA 线。
2. 确保 ICBRL 寄存器所设时间的 SCL 线的 Low 电平宽度。
3. 释放 SCL 线（从 Low 电平变为 High 电平）。
4. 在检测到 SCL 线的高电平后，确保 ICBRL 寄存器所设时间的重新开始条件的准备时间。
5. 使 SDA 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 SCL 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 SCL 线的 Low 电平宽度。

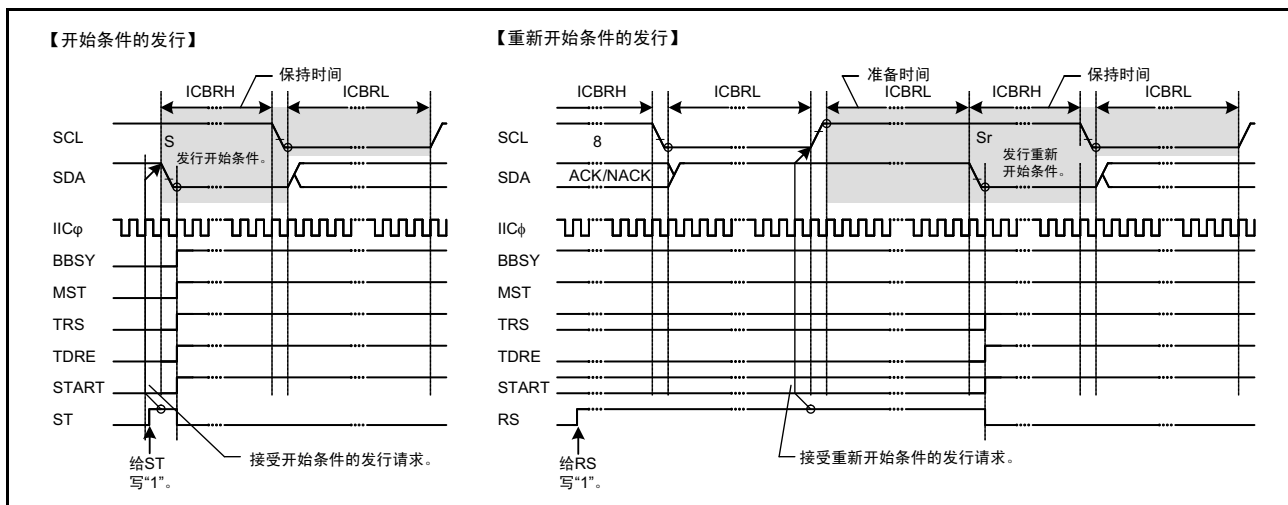


图 28.36 开始条件 / 重新开始条件的发行时序 (ST 位、RS 位)

28.10.3 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 线下降（从 High 电平变为 Low 电平）
- 确保 ICBRL 寄存器所设时间的 SCL 线的 Low 电平宽度
- 释放 SCL 线（从 Low 电平变为 High 电平）
- 在检测到 SCL 线的高电平后，确保 ICBRH 寄存器所设时间的停止条件的准备时间
- 释放 SDA 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）

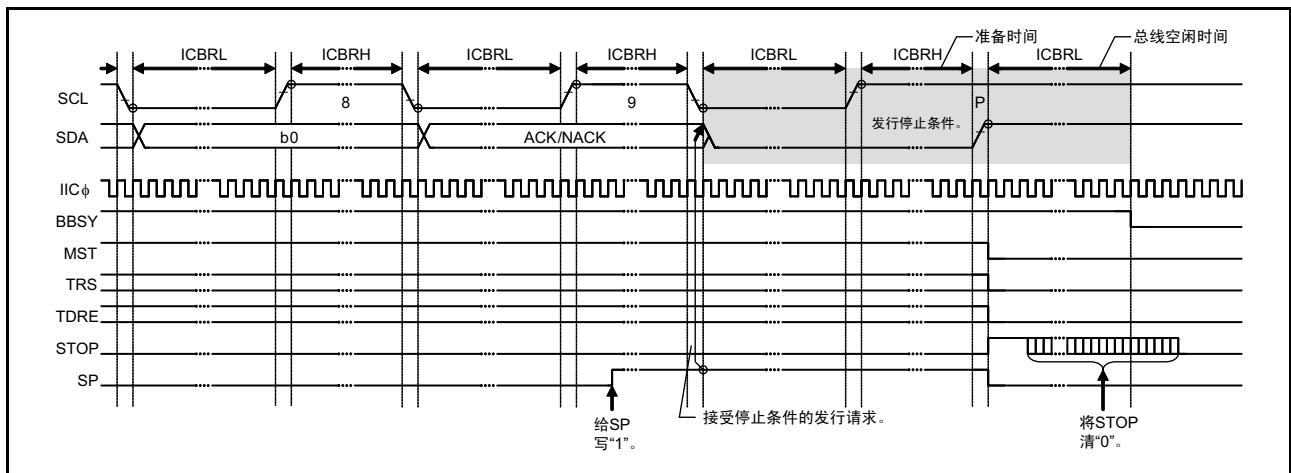


图 28.37 停止条件的发行时序 (SP 位)

28.11 总线意外停机

如果 I²C 总线主要因噪声等的影响而在主控设备和从属设备之间变为不同步，就可能在 SCL 线和 SDA 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 线来检测总线意外停机状态）、SCL 时钟的追加输出功能（用于解除因不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 线 /SDA 线输出了 Low 电平还是通信设备输出了 Low 电平。

28.11.1 超时检测功能

RIIC 具有超时检测功能，检测 SCL 线在一定时间后没有变化的状态。在总线忙的状态下，RIIC 能通过检测到 SCL 线的 Low 电平或者 High 电平被固定了一定的时间以上，并且检测到总线的异常状态。

超时检测功能监视 SCL 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 SCL 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 SCL 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 SCL 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 SCL 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。如果将 ICMR2.TMOH 位和 ICMR2.TMOL 位都置“0”，就不进行内部计数。

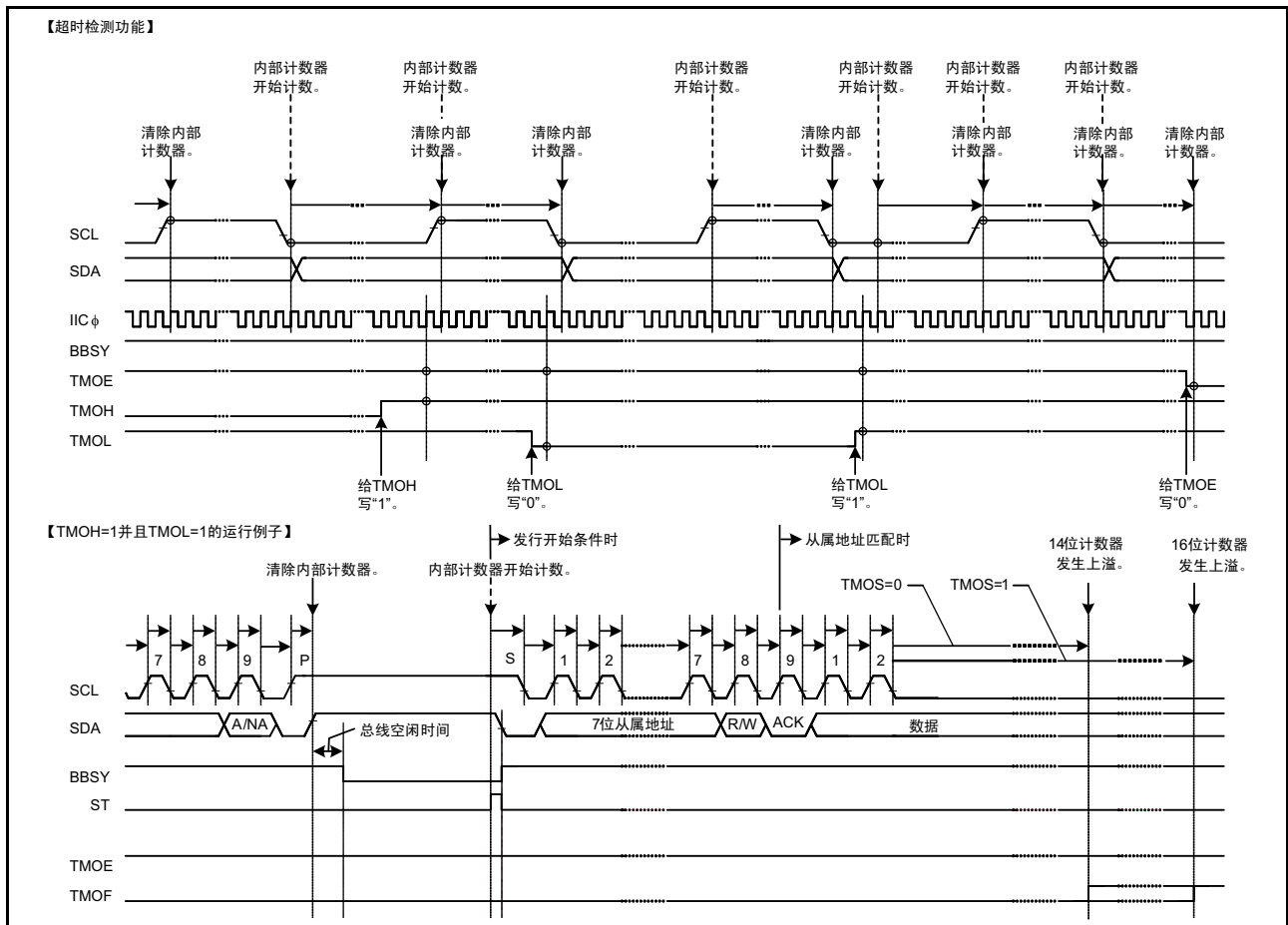


图 28.38 超时检测功能（TMOE 位、TMOS 位、TMOH 位、TMOL 位）

28.11.2 SCL 时钟追加输出功能

RIIC 具有 SCL 时钟的追加输出功能，在主控模式中用于释放因与从属设备不同步而使从属设备的 SDA 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，主要用于在主控模式中从属设备在 SDA 线固定为 Low 电平的状态而无法发行停止条件时释放从属设备的 SDA 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 寄存器所设传送速度的 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 位是“0”后写“1”，连续输出追加时钟。

在可控模式中，因噪声等的影响引起与从属设备不同步，导致从属设备将 SDA 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，释放从属设备的 SDA 线 Low 电平的固定状态，恢复总线状态。能通过检查 ICCR1.SDAI 位，确认此从属设备的 SDA 线的释放。为了在确认从属设备的 SDA 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 位的值和 SDA 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）下或者在可控模式（ICCR2.MST 位为“1”并且 BBSY 标志为“1”的状态）中
- 当通信设备没有保持 SCL 线的 Low 电平状态时

SCL 时钟追加输出功能（CLO 位）如图 28.39 所示。

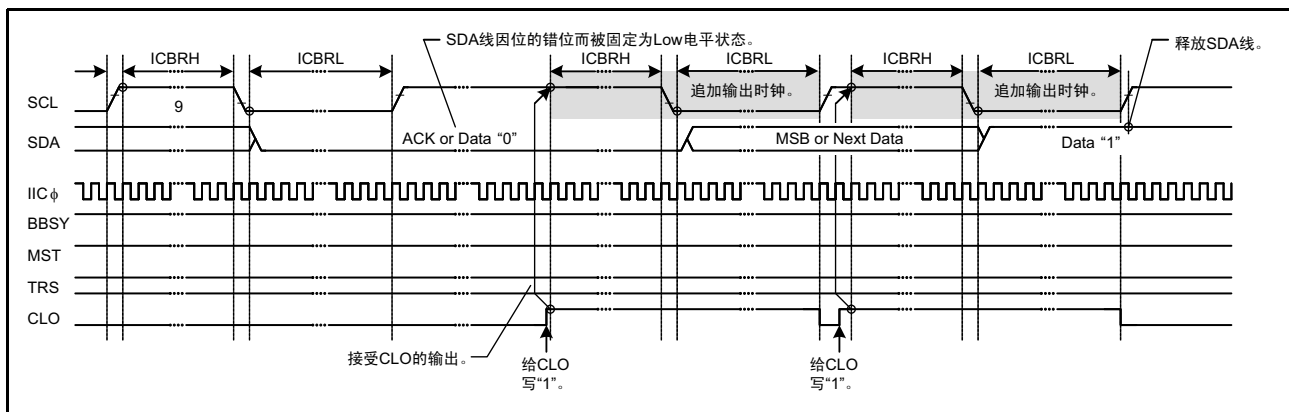


图 28.39 SCL 时钟追加输出功能（CLO 位）

28.11.3 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 引脚 /SDA 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设备的不同步，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“28.14 复位状况”。

28.12 SMBus 运行

RIIC 能进行以 SMBus (Ver.2.0) 为基准的通信。要进行 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, 将传送速度设定在 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 用作从属设备时, 不需要设定传送速度, 但是 ICBRL 的设定值至少为数据准备时间 (250ns) 的值。

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 SARU_y.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。

28.12.1 SMBus 超时测量

(1) 从属设备的超时测量

SMBus 通信的从属设备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

在通过从属设备进行超时测量时, 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 MTU3 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 [从属设备] $JT_{\text{LOW:SEXT}}$: 25ms (max) 以内。

如果通过 MTU3 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), 从属设备就需要释放总线。从属设备释放总线时, 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 SCL 引脚 /SDA 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) 主控设备的超时测量

SMBus 通信的主控设备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

在通过主控设备进行超时测量时, 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 MTU3 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 [主控设备] $JT_{\text{LOW:MEXT}}$: 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 个时钟上升沿); 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 MTU3 测量的时间超过 SMBus 规格的时钟 Low 电平的累积时间 [主控设备] $JT_{\text{LOW:MEXT}}$: 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), 主控设备就需要中止处理。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设备的处理。

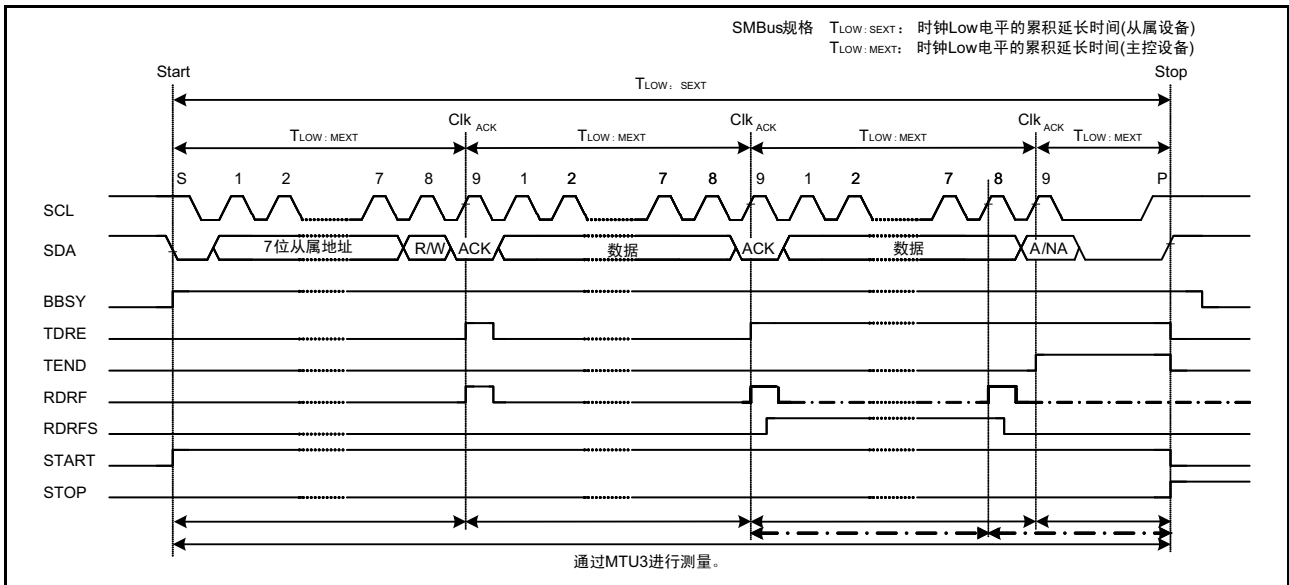


图 28.40 SMBus 超时测量

28.12.2 数据包错误码 (PEC)

RX63T 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“30. 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 PEC 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 SCL 线保持为 Low 电平。

28.12.3 SMBus 主机通知协议 /Notify ARP master

在 SMBus 中，从属设备能临时变为主控设备，将自己的从属地址通知给 SMBus 主机 (或者 ARP 主控)，或者向 SMBus 主机 (或者 ARP 主控) 请求自己的从属地址。

将 RX63T 群用作 SMBus 主机 (或者 ARP 主控) 时，需要将 从属设备发送的主机地址 (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。

28.13 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)、接收结束、发送数据空、发送结束共 4 种。

中断一览表如表 28.6 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 28.6 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	接收结束	—	能	能	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	低	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迟。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”。

28.13.1 ICTXI 中断和 ICRXI 中断的缓冲运行

对于 ICTXI 中断和 ICRXI 中断, 即使在 ICU.IRn.IR 标志为“1”时满足中断产生条件, 也不将中断请求输出到 ICU 而保持在内部 (在内部能保持的容量是每个中断源 1 个请求)。

如果 ICU.IRn.IR 标志变为“0”, 就将保持的中断请求输出到 ICU。在一般使用的状态下, 自动清除保持在内部的中断请求。

也能通过将对应的外围中断允许位置“0”来清除保持在内部的中断请求。

28.14 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 28.7 所示。

表 28.7 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	

28.15 使用时的注意事项

28.15.1 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。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“11. 低功耗功能”。

28.15.2 有关开始通信的注意事项

如果在开始通信 (ICCR1.ICE 位 =1) 时 ICU.IRn.IR 标志为“1”, 就必须在允许运行前按以下步骤清除中断请求。

1. 确认 ICCR1.ICE 位是“0”。
2. 将对应的外围中断允许位 (ICIER.TIE 等) 置“0”。
3. 读对应的外围中断允许位 (ICIER.TIE 等), 确认是“0”。
4. 将 ICU.IRn.IR 标志置“0”。

29. 串行外围接口 (RSPI)

29.1 概要

RX63T 群内置独立的 1 个通道的串行外围接口 (RSPI)。

RSPI 能进行全双工同步串行通信，还内置和多个处理器、外围设备进行高速通信的功能。

RSPI 的规格和框图分别如表 29.1 和图 29.1 所示。

在本章中，引脚名和信号名使用的 i 表示 0 ~ 3。RSPI 命令寄存器 m (SPCMD m) 使用的 m 表示 0 ~ 7。

表 29.1 RSPI 的规格

项目	内容
通道数	1 个通道
RSPI 传送功能	<ul style="list-style-type: none"> 能使用 MOSI (Master Out Slave In)、MISO (Master In Slave Out)、SSL (Slave Select) 和 RSPCK (RSPI Clock) 信号，通过 SPI 运行 (4 线式) 或者时钟同步运行 (3 线式) 进行串行通信。 只能进行发送。 能在主控模式或者从属模式中进行串行通信。 能更改串行传送时钟的极性。 能更改串行传送时钟的相位。
数据格式	<ul style="list-style-type: none"> 能选择 MSB first 或者 LSB first。 能将传送位长改为 8 位、9 位、10 位、11 位、12 位、13 位、14 位、15 位、16 位、20 位、24 位或者 32 位。 发送 / 接收缓冲器为 128 位。 1 次发送 / 接收最多传送 4 帧 (1 帧最多 32 位)。
位速率	<ul style="list-style-type: none"> 在主控模式中，通过内置波特率发生器将 PCLK 分频后，生成 RSPCK (最大分频比为 4096 分频) 在从属模式中，外部输入时钟作为串行时钟使用 (最大频率为 PCLK 的 8 分频) High 电平宽度: PCLK 的 4 个周期、Low 电平宽度: PCLK 的 4 个周期
缓冲器结构	发送 / 接收缓冲器的结构为双缓冲结构。
错误检测	<ul style="list-style-type: none"> 模式故障错误检测 溢出错误检测 奇偶校验错误检测
SSL 控制功能	<ul style="list-style-type: none"> 每个通道有 4 个 SSL 信号 (SSLA0 ~ SSLA3)。 在设定为单主控模式时，输出 SSLA0 ~ SSLA3 信号。 在设定为多主控模式时，SSLA0 信号为输入信号，SSLA1 ~ SSLA3 信号为输出信号或者不使用。 在设定为从属模式时，SSLA0 信号为输入信号，不使用 SSLA1 ~ SSLA3 信号。 能设定从 SSL 输出有效开始到 RSPCK 运行为止的延迟 (RSPCK 延迟)。 设定范围: 1 ~ 8 个 RSPCK 设定单位: 1 个 RSPCK 能设定从 RSPCK 停止开始到 SSL 输出无效为止的延迟 (SSL 无效延迟)。 设定范围: 1 ~ 8 个 RSPCK 设定单位: 1 个 RSPCK 能设定下次存取 of SSL 输出有效的等待 (下次存取延迟)。 设定范围: 1 ~ 8 个 RSPCK 设定单位: 1 个 RSPCK SSL 极性变更功能
主控传送时的控制方式	<ul style="list-style-type: none"> 能按顺序循环执行由最多 8 个命令构成的传送。 能给各命令设定以下项目: SSL 信号值、位速率、RSPCK 极性 / 相位、传送数据长度、LSB/MSB first、突发、RSPCK 延迟、SSL 无效延迟、下次存取延迟 能通过写发送缓冲器来启动传送。 能设定 SSL 无效时的 MOSI 信号值。
中断源	<ul style="list-style-type: none"> 可屏蔽的中断源 RSPI 接收中断 (接收缓冲器满) RSPI 发送中断 (发送缓冲器空) RSPI 错误中断 (模式故障错误、溢出错误和奇偶校验错误) RSPI 空闲中断 (RSPI 空闲)
其他功能	<ul style="list-style-type: none"> RSPI 初始化功能 环回模式功能
低功耗功能	能设定为模块停止状态。

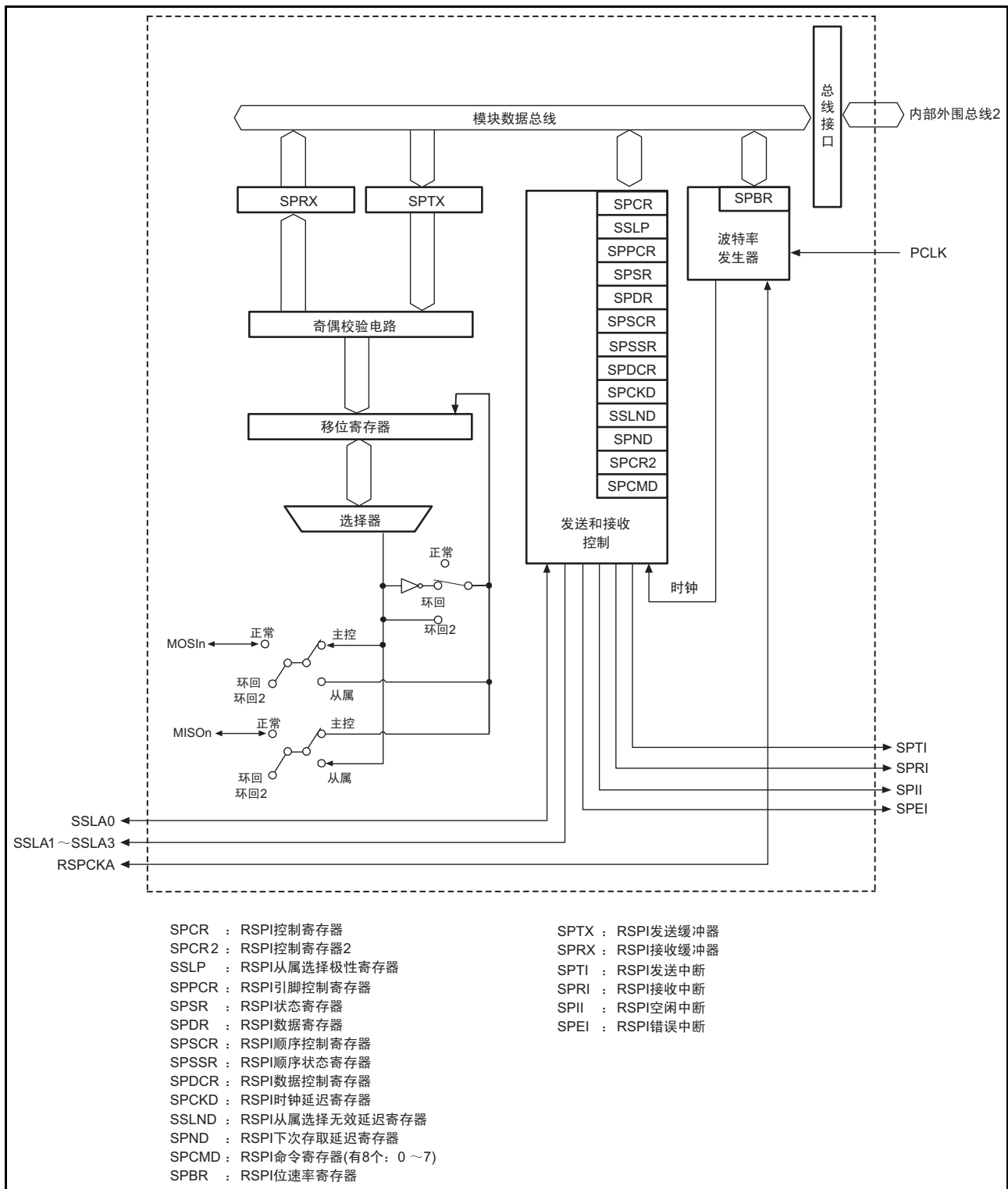


图 29.1 RSPI 的框图

RSPI 使用的输入 / 输出引脚如表 29.2 所示。

在设定为单主控模式时，RSPI 自动将 SSLA0 引脚的输入 / 输出方向更改为输出；在设定为多主控模式和从属模式时，RSPI 自动将 SSLA0 引脚的输入 / 输出方向更改为输入。根据主控模式 / 从属模式的设定和 SSLA0 引脚的输入电平，RSPI 自动更改 RSPCKA、MOSIA 和 MISOA 引脚的输入 / 输出方向。

详细内容请参照“29.3.2 RSPI 引脚的控制”。

表 29.2 RSPI 的输入 / 输出引脚

通道	引脚名	输入 / 输出	功能
RSPI0	RSPCKA	输入 / 输出	时钟输入 / 输出引脚
	MOSIA	输入 / 输出	主控发送数据输入 / 输出引脚
	MISOA	输入 / 输出	从属发送数据输入 / 输出引脚
	SSLA0	输入 / 输出	从属选择输入 / 输出引脚
	SSLA1	输出	从属选择输出引脚
	SSLA2	输出	从属选择输出引脚
	SSLA3	输出	从属选择输出引脚

29.2 寄存器说明

29.2.1 RSPI 控制寄存器 (SPCR)

地址 RSPi0.SPCR 0008 8380h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	SPMS	RSPI 模式选择位	0: SPI 运行 (4 线式) 1: 时钟同步运行 (3 线式)	R/W
b1	TXMD	通信运行模式选择位	0: 全双工同步串行通信 1: 只进行发送运行的串行通信	R/W
b2	MODFEN	模式故障错误检测允许位	0: 禁止模式故障错误的检测 1: 允许模式故障错误的检测	R/W
b3	MSTR	RSPI 主控 / 从属模式选择位	0: 从属模式 1: 主控模式	R/W
b4	SPEIE	RSPI 错误中断允许位	0: 禁止 RSPI 错误中断请求的产生 1: 允许 RSPI 错误中断请求的产生	R/W
b5	SPTIE	RSPI 发送中断允许位	0: 禁止 RSPI 发送中断请求的产生 1: 允许 RSPI 发送中断请求的产生	R/W
b6	SPE	RSPI 功能允许位	0: RSPI 功能无效 1: RSPI 功能有效	R/W
b7	SPRIE	RSPI 接收中断允许位	0: 禁止 RSPI 接收中断请求的产生 1: 允许 RSPI 接收中断请求的产生	R/W

如果在 SPCR.SPE 位为“1”的状态下更改 SPCR.MSTR 位、SPCR.MODFEN 位和 SPCR.TXMD 位的设定值，就不保证以后的运行。

SPMS 位 (RSPI 模式选择位)

此位选择 SPI 运行 (4 线式) 或者时钟同步运行 (3 线式)。

在进行时钟同步运行时，不使用 SSLA0 ~ 3 引脚而使用 RSPCKA 引脚、MOSIA 引脚和 MISOA 引脚这 3 个引脚进行通信。在 主控模式中 (SPCR.MSTR=1) 进行时钟同步运行时，能将 SPCMDm.CPHA 位置“0”或者“1”；在从属模式中 (SPCR.MSTR=0) 进行时钟同步运行时，就不保证将 CPHA 位置“0”时的运行。

TXMD 位 (发送运行模式选择位)

此位选择全双工同步串行通信或者只进行发送运行的串行通信。

在将 TXMD 位置“1”后进行通信时，只进行发送运行而不进行接收运行 (参照“29.3.6 通信运行模式”)。如果将 TXMD 置“1”，就不能使用接收缓冲器满的中断请求。

MODFEN 位 (模式故障错误检测允许位)

此位允许或者禁止模式故障错误检测 (参照“29.3.8 错误检测”)。RSPI 根据 MODFEN 位和 MSTR 位的组合，决定 SSLA0 ~ 3 引脚的输入 / 输出方向 (参照“29.3.2 RSPI 引脚的控制”)。

MSTR 位 (RSPI 主控 / 从属模式选择位)

此位选择 RSPI 的主控模式或者从属模式。RSPI 根据 MSTR 位的设定, 决定 RSPCKA、MOSIA、MISOA 和 SSLA0 ~ 3 引脚的方向。

SPEIE 位 (RSPI 错误中断允许位)

如果在 RSPI 检测到模式故障错误后将 SPSR.MODF 标志置“1”, 或者在 RSPI 检测到溢出错误后将 SPSR.OVRF 标志置“1”, 或者在 RSPI 检测到奇偶校验错误后将 SPSR.PERF 标志置“1”, 此位就允许或者禁止 RSPI 错误中断请求的产生。详细内容请参照“29.3.8 错误检测”。

SPE 位 (RSPI 功能允许位)

此位选择 RSPI 功能的有效或者无效。

当 SPSR.MODF 位为“1”时, 不能将 SPE 位置“1”。详细内容请参照“29.3.8 错误检测”。

如果将 SPE 位置“0”, 就将 RSPI 功能置为无效并且对一部分模块功能进行初始化。详细内容请参照“29.3.9 RSPI 的初始化”。通过将 SPE 位从“0”的状态变为“1”的状态或者从“1”的状态变为“0”的状态, 产生 RSPI 发送中断请求。

29.2.2 RSPI 从属选择极性寄存器 (SSLP)

地址 RSP10.SSLP 0008 8381h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	SSL0P	SSL0 信号极性设定位	0: SSL0 信号的 Low 电平有效 1: SSL0 信号的 High 电平有效	R/W
b1	SSL1P	SSL1 信号极性设定位	0: SSL1 信号的 Low 电平有效 1: SSL1 信号的 High 电平有效	R/W
b2	SSL2P	SSL2 信号极性设定位	0: SSL2 信号的 Low 电平有效 1: SSL2 信号的 High 电平有效	R/W
b3	SSL3P	SSL3 信号极性设定位	0: SSL3 信号的 Low 电平有效 1: SSL3 信号的 High 电平有效	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

如果在 SPCR.SPE 位为“1”的状态下改写 SSLP 寄存器, 就不保证以后的运行。

29.2.3 RSPI 引脚控制寄存器 (SPPCR)

地址 RSPI0.SPPCR 0008 8382h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	SPLP	RSPI 环回位	0: 正常模式 1: 环回模式 (发送数据的反相 = 接收数据)	R/W
b1	SPLP2	RSPI 环回 2 位	0: 正常模式 1: 环回模式 (发送数据 = 接收数据)	R/W
b3-b2	—	保留位	读写值都为“0”。	R/W
b4	MOIFV	MOSI 空闲固定值位	0: MOSI 空闲时的 MOSIA 引脚的输出值为“0” 1: MOSI 空闲时的 MOSIA 引脚的输出值为“1”	R/W
b5	MOIFE	MOSI 空闲值固定允许位	0: MOSI 输出值为上次传送的最后数据 1: MOSI 输出值为 MOIFV 位的设定值	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W

如果在 SPCR.SPE 位为“1”的状态下改写 SPPCR 寄存器，就不保证以后的运行。

SPLP 位 (RSPI 环回位)

此位选择 RSPI 的引脚模式。

当 SPLP 位被置“1”时，如果 SPCR.MSTR 位为“1”，RSPI 就切断 MISOA 引脚和移位寄存器之间的线路，而连接移位寄存器的输入线路和输出线路（取反）；如果 SPCR.MSTR 位为“0”，RSPI 就在从属运行时切断 MOSIA 引脚和移位寄存器之间的线路，而连接移位寄存器的输入线路和输出线路（取反）（环回模式）。

SPLP2 位 (RSPI 环回 2 位)

此位选择 RSPI 的引脚模式。

当 SPLP2 位被置“1”时，如果 SPCR.MSTR 位为“1”，RSPI 就切断 MISOA 引脚和移位寄存器之间的线路，而连接移位寄存器的输入线路和输出线路；如果 SPCR.MSTR 位为“0”，RSPI 就在从属运行时切断 MOSIA 引脚和移位寄存器之间的线路，而连接移位寄存器的输入线路和输出线路（环回模式）。

MOIFV 位 (MOSI 空闲固定值位)

在主机模式中，当 MOIFE 位为“1”时，此位选择 SSL 无效期间（包括突发传送中的 SSL 保持期间）的 MOSIA 引脚输出值。

MOIFE 位 (MOSI 空闲值固定允许位)

主机模式的 RSPI 在 SSL 无效期间（包含突发传送中的 SSL 保持期间）使用此位固定 MOSIA 的输出值。当 MOIFE 位为“0”时，RSPI 在 SSL 无效期间内将上次串行传送的最后数据输出到 MOSIA；当 MOIFE 位为“1”时，RSPI 将 MOIFV 位设定的固定值输出到 MOSIA。

29.2.4 RSPI 状态寄存器 (SPSR)

地址 RSPI0.SPSR 0008 8383h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	PERF	MODF	DLNF	OVRF
复位后的值	x	0	x	0	0	0	0	0

x: 不定值

位	符号	位名	功能	R/W
b0	OVRF	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)
b1	IDLNF	RSPI 空闲标志	0: RSPI 为空闲状态 1: RSPI 为传送状态	R
b2	MODF	模式故障错误标志	0: 未发生模式故障错误 1: 发生模式故障错误	R/(W) (注1)
b3	PERF	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	—	保留位	读写值都为“0”。	R/W
b5	—	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“0”。	R/W
b7	—	保留位	读取值为不定值, 只能写“1”。	R/W

注1. 为了清除标志, 只能在读“1”后写“0”。

OVRF 标志 (溢出错误标志)

此标志表示溢出错误的发生状况。

[为“1”的条件]

- 在 SPCR.TXMD 位为“0”并且接收缓冲器有未读数据的状态下串行传送结束时

[为“0”的条件]

- 在 OVRF 标志为“1”的状态下读 SPSR 寄存器后给 OVRF 标志写“0”时

IDLNF 标志 (RSPI 空闲标志)

此标志表示 RSPI 的传送状况。

[为“1”的条件]

【主控模式】

- 当上述[为“0”的条件]的【主控模式】中的条件一个也不满足时

【从属模式】

- 当 SPCR.SPE 位为“1” (RSPI 功能有效) 时

[为“0”的条件]

【主控模式】

- 当 SPCR.SPE 位为“0” (RSPI 的初始化) 时
- 当发送缓冲器 (SPTX) 为空 (未设定下一个传送数据) 时
- 当 SPSR.SPCP[2:0] 位为“000b” (顺序控制的开头) 时
- 当 RSPI 内部定序器转移到空闲状态时 (在下次存取延迟前结束运行的状态)

当满足上述 1. 条件或者满足上述 2. ~ 4. 的全部条件时

【从属模式】

- 当 SPCR.SPE 位为“0” (RSPI 的初始化) 时

MODF 标志 (模式故障错误标志)

此标志表示发生了模式故障错误。

[为“1”的条件]

【多主控模式】

- 在SPCR.MSTR位为“1”（主控模式）并且SPCR.MODFEN位为“1”（允许模式故障错误的检测）的状态下，SSLAi引脚的输入电平为有效电平并且RSPI检测到模式故障错误时

【从属模式】

- 在SPCR.MSTR位为“0”（从属模式）并且SPCR.MODFEN位为“1”（允许模式故障错误的检测）的状态下，在数据传送所需的RSPCK周期结束前SSLAi引脚无效并且RSPI检测到模式故障错误时

SSLAi信号的有效电平取决于SSLP.SSLiP位（SSL信号极性设定位）。

[为“0”的条件]

- 在MODF标志为“1”的状态下读SPSR寄存器后给MODF标志写“0”时

PERF 标志 (奇偶校验错误标志)

此标志表示发生了奇偶校验错误。

[为“1”的条件]

- 在SPCR.TXMD位为“0”并且SPCR2.SPPE位为“1”的状态下串行传送结束，并且检测到奇偶校验错误时

[为“0”的条件]

- 在PERF标志为“1”的状态下读SPSR寄存器后给PERF标志写“0”时

29.2.5 RSPI 数据寄存器 (SPDR)

地址 RSPI0.SPDR 0008 8384h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
复位后的值	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
SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
复位后的值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDR 寄存器是保存 RSPI 发送 / 接收数据的缓冲器。发送缓冲器和接收缓冲器是独立的缓冲器。SPDR 寄存器的结构图如图 29.2 所示。

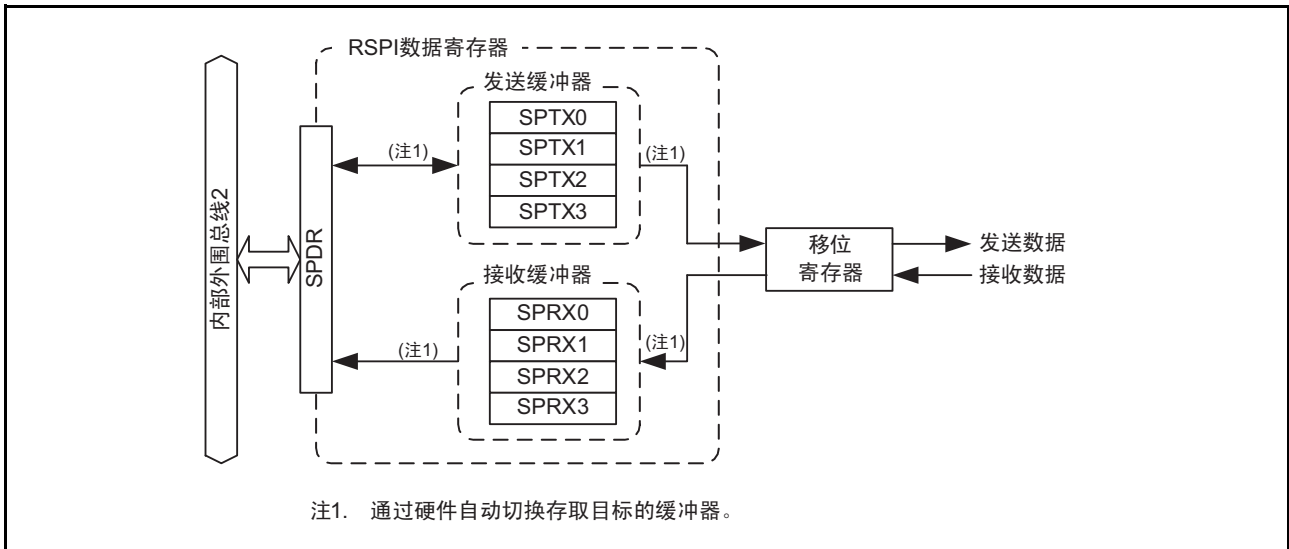


图 29.2 SPDR 寄存器的结构图

发送缓冲器和接收缓冲器分别有4个缓冲器，能通过RSPI数据控制寄存器的帧数设定位 (SPDCR.SPFC[1:0]) 设定要使用的缓冲器个数。这8个缓冲器被映像到SPDR寄存器的1个地址。

能通过写SPDR寄存器，将值写到发送缓冲器 (SPTX0 ~ 3) 并且发送写的的数据。

如果接收完数据，接收缓冲器就保存接收数据。当发生溢出错误时，不更新接收缓冲器的值。

当数据长度不是32位时，将SPTX_n (n=0 ~ 3) 的非参照位保存到SPRX_n (n=0 ~ 3) 的非参照位。例如，当数据长度是9位时，将SPTX_n[31:9] 保存到SPRX_n[31:9] (SPTX_n[8:0] 保存接收数据)。

(1) 总线接口

SPDR 寄存器有 4 个 32 位发送缓冲器和 4 个 32 位接收缓冲器，合计 32 字节。此 32 字节被映像到 SPDR 寄存器的 4 字节空间。通过 RSPI 数据控制寄存器的 RSPI 长字存取 / 字存取设定位 (SPDCR.SPLW) 设定的存取长度，进行 SPDR 寄存器的存取。

必须以往 LSB 靠紧的格式写发送数据。以往 LSB 靠紧的格式保存接收数据。

SPDR 寄存器的读写操作如下所示。

(a) 写操作

发送缓冲器有发送缓冲器写指针，通过写 SPDR 寄存器，指针自动切换到下一个缓冲器。

发送缓冲器的总线接口 (写时) 的结构图如图 29.3 所示。

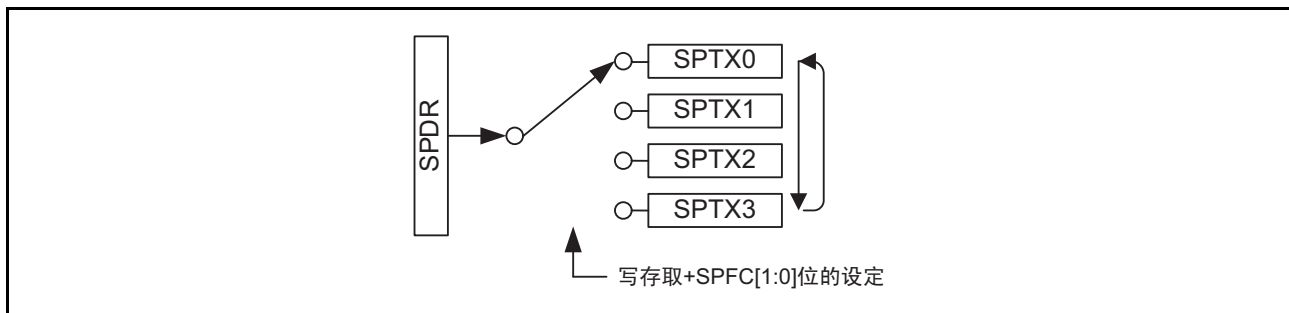


图 29.3 SPDR 寄存器的结构图 (写时)

发送缓冲器写指针的切换顺序因 RSPI 数据控制寄存器的帧数设定位 (SPDCR.SPFC[1:0]) 的设定而不同。

- SPFC[1:0] 位的设定和 SPTX0~3 的切换顺序
 - 当 SPFC[1:0] 位为“00b”时，SPTX0→SPTX0→SPTX0→…
 - 当 SPFC[1:0] 位为“01b”时，SPTX0→SPTX1→SPTX0→SPTX1→…
 - 当 SPFC[1:0] 位为“10b”时，SPTX0→SPTX1→SPTX2→SPTX0→SPTX1→…
 - 当 SPFC[1:0] 位为“11b”时，SPTX0→SPTX1→SPTX2→SPTX3→SPTX0→SPTX1→…

如果在 RSPI 控制寄存器的 RSPI 功能允许位 (SPCR.SPE) 为“0”的状态下写“1”，下一个写目标就为 SPTX0。

必须在发生 RSPI 发送中断后，将 RSPI 数据控制寄存器 (SPDCR) 的帧数设定位 (SPFC[1:0]) 所设帧数的发送数据写到发送缓冲器 (SPTXn)。在发生下一个 RSPI 发送中断前，即使写发送缓冲器 (SPTXn) 也放弃 SPDR 寄存器的写存取。

(b) 读操作

能通过读 SPDR 寄存器，读到接收缓冲器 (SPRX0 ~ 3) 或者发送缓冲器 (SPTX0 ~ 3) 的值。能通过 RSPI 数据控制寄存器的 RSPI 接收/发送数据选择位 (SPDCR.SPRDTC) 选择是读接收缓冲器还是读发送缓冲器。

在 SPDR 寄存器的读结构中，有 2 个独立的指针 (接收缓冲器读指针和发送缓冲器读指针)。通过读 SPDR 寄存器，指针自动切换到下一个缓冲器。

接收缓冲器和发送缓冲器的总线接口 (读时) 的结构图如图 29.4 所示。

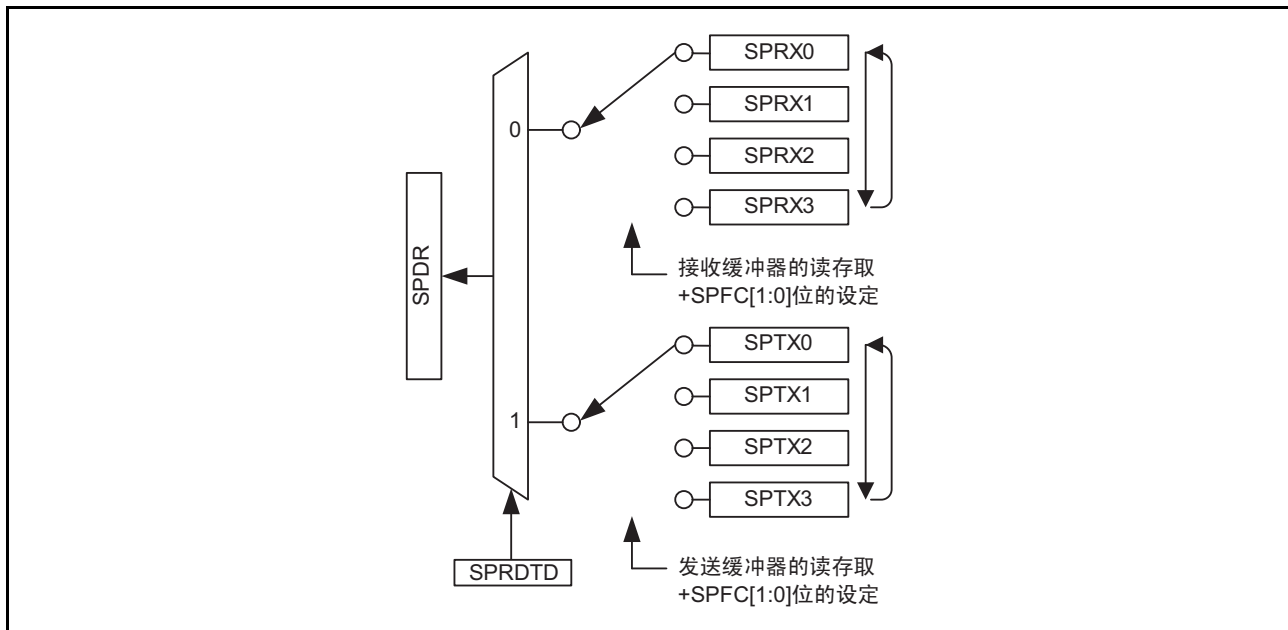


图 29.4 SPDR 寄存器的结构图 (读时)

接收缓冲器读指针的切换顺序和发送缓冲器读指针相同。如果读发送缓冲器，就能读到刚写入发送缓冲器的值。读时，只更新由 RSPI 数据控制寄存器的 RSPI 接收/发送数据选择位 (SPDCR.SPRDTC) 选择的读缓冲器的缓冲器读指针，而其他缓冲器读指针保持不变。

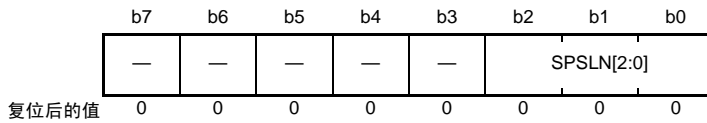
如果在 RSPI 控制寄存器的 RSPI 功能允许位 (SPCR.SPE) 为“0”的状态下写“1”，下一个读目标就由 SPRXn 的缓冲器读指针指向 SPTX0。

在发生 RSPI 发送中断以及写 RSPI 数据控制寄存器的帧数设定位 (SPDCR.SPFC[1:0]) 所设帧数的发送数据之后，如果在发生下一个 RSPI 发送中断之前读发送缓冲器，读取值就全部为“0”。

在发生 RSPI 发送中断后，如果写 RSPI 数据控制寄存器的帧数设定位 (SPDCR.SPFC[1:0]) 所设帧数的发送数据，就清除 SPTXn 的缓冲器读指针。

29.2.6 RSPI 顺序控制寄存器 (SPSCR)

地址 RSPI0.SPSCR 0008 8388h



位	符号	位名	功能	R/W
b2-b0	SPSLN[2:0]	RSPI 顺序长度设定位	b2 b0 顺序长度 参照的 SPCMD0 ~ 7 寄存器 (序号) 0 0 0: 1 0→0→... 0 0 1: 2 0→1→0→... 0 1 0: 3 0→1→2→0→... 0 1 1: 4 0→1→2→3→0→... 1 0 0: 5 0→1→2→3→4→0→... 1 0 1: 6 0→1→2→3→4→5→0→... 1 1 0: 7 0→1→2→3→4→5→6→0→... 1 1 1: 8 0→1→2→3→4→5→6→7→0→... 根据所设定的顺序长度, 更改要参照的 SPCMD0 ~ 7 寄存器和参照顺序。SPSLN[2:0] 位的设定值和顺序长度、RSPI 要参照的 SPCMD0 ~ 7 寄存器的关系如上所述。另外, 从属模式的 RSPI 能随时参照 SPCMD0 寄存器。	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

SPSCR 寄存器是在 RSPI 主控运行时设定顺序长度的寄存器。如果在 SPCR.MSTR 位和 SPCR.SPE 位都为“1”的状态下改写 SPSCR.SPSSLN[2:0] 位, 就必须在 SPSR.IDLNF 标志为“0”时进行改写。

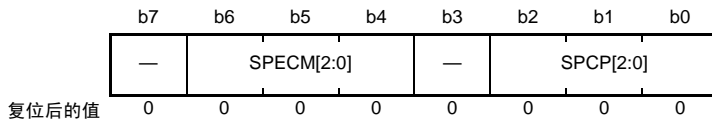
SPSLN[2:0] 位 (RSPI 顺序长度设定位)

这些位设定主控模式的 RSPI 进行顺序运行时的顺序长度。主控模式的 RSPI 根据 SPSLN[2:0] 位所设定的顺序长度, 更改要参照的 SPCMD0 ~ 7 寄存器和参照顺序。

从属模式的 RSPI 能随时参照 SPCMD0 寄存器。

29.2.7 RSPI 顺序状态寄存器 (SPSSR)

地址 RSPI0.SPSSR 0008 8389h



位	符号	位名	功能	R/W
b2-b0	SPCP[2:0]	RSPI 命令指针位	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b3	—	保留位	读取值为“0”。	R
b6-b4	SPECM[2:0]	RSPI 错误命令位	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b7	—	保留位	读取值为“0”。	R

SPSSR 寄存器表示 RSPI 主控运行时的顺序控制状态。

SPSSR 寄存器的写操作无效。

SPCP[2:0] 位 (RSPI 命令指针位)

这些位通过 RSPI 的顺序控制，表示当前指针所指向的 SPCMDm 寄存器。

有关 RSPI 的顺序控制，请参照“29.3.10.1 主控模式的运行”。

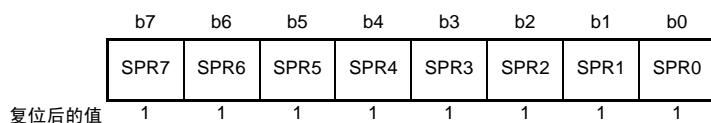
SPECM[2:0] 位 (RSPI 错误命令位)

这些位通过 RSPI 顺序控制，表示在检测到错误时 SPCP[2:0] 位指定的 SPCMDm 寄存器。只有在检测到错误时，RSPI 才能更新 SPECM[2:0] 位。如果 SPSR.OVRF 位和 MODF 标志都为“0”并且没有发生错误，SPECM[2:0] 位的值就没有含义。

有关 RSPI 的错误检测功能，请参照“29.3.8 错误检测”；有关 RSPI 的顺序控制，请参照“29.3.10.1 主控模式的运行”。

29.2.8 RSPI 位速率寄存器 (SPBR)

地址 RSPI0.SPBR 0008 838Ah



SPBR 寄存器用于设定主控模式中的位速率。如果在 SPCR.MSTR 和 SPCR.SPE 位都为“1”的状态下改写 SPBR 寄存器，就不保证以后的运行。

在从属模式中使用 RSPI 时，与 SPBR 寄存器和 SPCMDm.BRDV[1:0] 位（位速率分频设定位）无关，取决于输入时钟的位速率（必须使用满足电特性的位速率）。

位速率取决于 SPBR 寄存器的设定值和 SPCMDm.BRDV[1:0] 位设定值的组合，位速率的计算式如下。在计算式中，n 为 SPBR 寄存器的设定值（0、1、2、……、255），N 为 BRDV[1:0] 位的设定值（0、1、2、3）。

$$\text{位速率} = \frac{f(\text{PCLK})}{2 \times (n+1) 2^N}$$

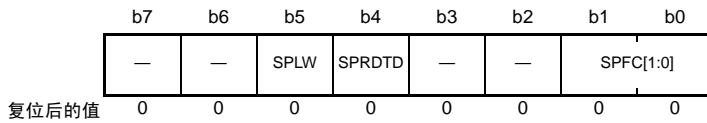
SPBR 寄存器、BRDV[1:0] 位的设定值和位速率的关系例子如表 29.3 所示。

表 29.3 SPBR 寄存器、BRDV[1:0] 位的设定值和位速率

SPBR 寄存器的设定值 (n)	BRDV[1:0] 位的设定值 (N)	分频比	位速率			
			PCLK=32MHz	PCLK=36MHz	PCLK=40MHz	PCLK=50MHz
0	0	2	16.0Mbps	18.0Mbps	20.0Mbps	25.0Mbps
1	0	4	8.00Mbps	9.00Mbps	10.0Mbps	12.5Mbps
2	0	6	5.33Mbps	6.00Mbps	6.67Mbps	8.33Mbps
3	0	8	4.00Mbps	4.50Mbps	5.00Mbps	6.25Mbps
4	0	10	3.20Mbps	3.60Mbps	4.00Mbps	5.00Mbps
5	0	12	2.67Mbps	3.00Mbps	3.33Mbps	4.16Mbps
5	1	24	1.33Mbps	1.50Mbps	1.67Mbps	2.08Mbps
5	2	48	667kbps	750kbps	833kbps	1.04Mbps
5	3	96	333kbps	375kbps	417kbps	521kbps
255	3	4096	7.81kbps	8.80kbps	9.78kbps	12.2kbps

29.2.9 RSPI 数据控制寄存器 (SPDCR)

地址 RSPi0.SPDCR 0008 838Bh



位	符号	位名	功能	R/W
b1-b0	SPFC[1:0]	帧数设定位	b1 b0 0 0: 1 帧 0 1: 2 帧 1 0: 3 帧 1 1: 4 帧	R/W
b3-b2	—	保留位	读写值都为“0”。	R/W
b4	SPRDTD	RSPI 接收 / 发送数据选择位	0: SPDR 读接收缓冲器 1: SPDR 读发送缓冲器 (但是, 在发送缓冲器为空时)	R/W
b5	SPLW	RSPI 长字存取 / 字存取设定位	0: 对 SPDR 寄存器进行字存取 1: 对 SPDR 寄存器进行长字存取	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W

通过 SPCMDm.SPB[3:0] 位、SPSCR.SPSSLN[2:0] 位和 SPDCR.SPFC[1:0] 位的组合, 启动 1 次发送和接收能进行最多 4 帧的发送和接收。

如果在 SPCR.SPE 位为“1”的状态下改写 SPDCR.SPFC[1:0] 位, 就必须在 SPSR.IDLNF 标志为“0”时进行改写。

SPFC[1:0] 位 (帧数设定位)

这些位设定 SPDR 寄存器能保存 (1 次传送启动) 的帧数。通过设定 SPSCR.SPSSLN[2:0] 位和 SPDCR.SPFC[1:0] 位, 启动 1 次发送和接收能进行最多 4 帧的发送和接收。另外, 通过设定 SPFC[1:0] 位, 更改用于发生 RSPI 接收中断和 RSPI 发送中断或者开始发送的帧数。SPDR 寄存器能保存的帧结构和发送 / 接收设定的组合例子如表 29.4 所示。如果设定组合例子中没有的组合, 就不保证以后的运行。

SPRDTD 位 (RSPI 接收 / 发送数据选择位)

此位选择 SPDR 寄存器的读取值是从接收缓冲器读取还是从发送缓冲器读取。

如果读发送缓冲器, 就能读到刚写入 SPDR 寄存器的值。

必须在发生 RSPI 发送中断后并且在写完 SPFC[1:0] 位所设帧数前, 读发送缓冲器。

详细内容请参照“29.2.5 RSPI 数据寄存器 (SPDR)”。

SPLW 位 (RSPI 长字存取 / 字存取设定位)

此位设定 SPDR 寄存器的存取宽度。当 SPLW 位为“0”时, 对 SPDR 寄存器进行字存取; 当 SPLW 位为“1”时, 对 SPDR 寄存器进行长字存取。

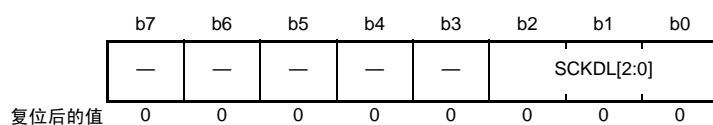
当 SPLW 位为“0”时, 必须给 SPCMDm.SPB[3:0] 位 (RSPI 数据长度设定位) 设定 8 ~ 16 位。如果设定 20 位、24 位或者 32 位, 就不保证运行。

表 29.4 SPSLN[2:0] 位和 SPFC[1:0] 位能设定的组合

设定	SPSLN[2:0]	SPFC[1:0]	1 个顺序 传送的帧数	发生接收缓冲器满中断或者发送缓冲器 有数据时的帧数
1-1	000	00	1	1
1-2	000	01	2	2
1-3	000	10	3	3
1-4	000	11	4	4
2-1	001	01	2	2
2-2	001	11	4	4
3	010	10	3	3
4	011	11	4	4
5	100	00	5	1
6	101	00	6	1
7	110	00	7	1
8	111	00	8	1

29.2.10 RSPI 时钟延迟寄存器 (SPCKD)

地址 RSPi0.SPCKD 0008 838Ch



位	符号	位名	说明	R/W
b2-b0	SCKDL[2:0]	RSPCK 延迟设定位	b2 b0 0 0 0: 1 个 RSPCK 0 0 1: 2 个 RSPCK 0 1 0: 3 个 RSPCK 0 1 1: 4 个 RSPCK 1 0 0: 5 个 RSPCK 1 0 1: 6 个 RSPCK 1 1 0: 7 个 RSPCK 1 1 1: 8 个 RSPCK	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

在 SPCMDm.SCKDEN 位为“1”的状态下，SPCKD 寄存器设定从 SSLAi 信号有效开始到 RSPCK 振荡的期间 (RSPCK 延迟)。如果在 SPCR.MSTR 位和 SPCR.SPE 位都为“1”的状态下改写 SPCKD 寄存器，就不保证以后的运行。

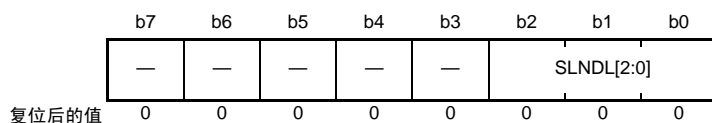
SCKDL[2:0] 位 (RSPCK 延迟设定位)

这些位设定 SPCMDm.SCKDEN 位为“1”时的 RSPCK 延迟值。

在从属模式中使用 RSPI 时，必须将 SCKDL[2:0] 位置“000b”。

29.2.11 RSPI 从属选择无效延迟寄存器 (SSLND)

地址 RSPI0.SSLND 0008 838Dh



位	符号	位名	功能	R/W
b2-b0	SLNDL[2:0]	SSL 无效延迟设定位	b2 b0 0 0 0: 1 个 RSPCK 0 0 1: 2 个 RSPCK 0 1 0: 3 个 RSPCK 0 1 1: 4 个 RSPCK 1 0 0: 5 个 RSPCK 1 0 1: 6 个 RSPCK 1 1 0: 7 个 RSPCK 1 1 1: 8 个 RSPCK	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

SSLND 寄存器设定主控模式的 RSPI 从发送串行传送最后的 RSPCK 边沿到将 SSLAi 信号置为无效的期间 (SSL 无效延迟)。如果在 SPCR.MSTR 位和 SPCR.SPE 位都为“1”的状态下改写 SSLND 寄存器, 就不保证以后的运行。

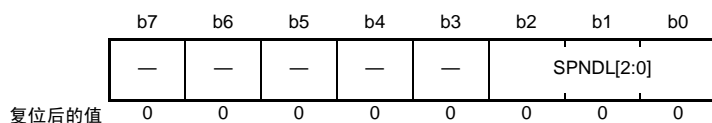
SLNDL[2:0] 位 (SSL 无效延迟设定位)

这些位设定主控模式的 RSPI 的 SSL 无效延迟值。

在从属模式中使用 RSPI 时, 必须将 SLNDL[2:0] 位置“000b”。

29.2.12 RSPI 下次存取延迟寄存器 (SPND)

地址 RSPi0.SPND 0008 838Eh



位	符号	位名	功能	R/W
b2-b0	SPNDL[2:0]	RSPI 下次存取延迟设定位	b2 b0 0 0 0: 1 个 RSPCK+2 个 PCLK 0 0 1: 2 个 RSPCK+2 个 PCLK 0 1 0: 3 个 RSPCK+2 个 PCLK 0 1 1: 4 个 RSPCK+2 个 PCLK 1 0 0: 5 个 RSPCK+2 个 PCLK 1 0 1: 6 个 RSPCK+2 个 PCLK 1 1 0: 7 个 RSPCK+2 个 PCLK 1 1 1: 8 个 RSPCK+2 个 PCLK	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

在 SPCMDm.SPNDEN 位为“1”的状态下，SPND 寄存器设定串行传送结束后的 SSLAi 信号的无效期间（下次存取延迟）。如果在 SPCR.MSTR 位和 SPCR.SPE 位都为“1”的状态下改写 SPND 寄存器，就不保证以后的运行。

SPNDL[2:0] 位 (RSPI 下次存取延迟设定位)

这些位设定 SPCMDm.SPNDEN 位为“1”时的下次存取延迟。

在从属模式中使用 RSPI 时，必须将 SPNDL[2:0] 位置“000b”。

29.2.13 RSPI 控制寄存器 2 (SPCR2)

地址 RSPi0.SPCR2 0008 838Fh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	PTE	SPIIE	SPOE	SPPE
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	SPPE	奇偶校验允许位	0: 不给发送数据附加奇偶校验位, 也不进行接收数据的奇偶校验。 1: 给发送数据附加奇偶校验位并且进行接收数据的奇偶校验 (当 SPCR.TXMD=0 时)。 给发送数据附加奇偶校验位, 但是不进行接收数据的奇偶校验 (当 SPCR.TXMD=1 时)。	R/W
b1	SPOE	奇偶校验模式位	0: 通过偶校验进行发送和接收 1: 通过奇校验进行发送和接收	R/W
b2	SPIIE	RSPI 空闲中断允许位	0: 禁止空闲中断请求的产生 1: 允许空闲中断请求的产生	R/W
b3	PTE	奇偶校验自诊断位	0: 奇偶校验电路自诊断功能无效 1: 奇偶校验电路自诊断功能有效	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

如果在 SPCR.SPE 位为“1”的状态下改写 SPCR2 寄存器的 SPPE 位和 SPOE 位的设定值, 就不保证以后的运行。

SPPE 位 (奇偶校验允许位)

此位选择奇偶校验功能的有效或者无效。

当 SPCR.TXMD 位为“0”并且 SPCR2.SPPE 位为“1”时, 给发送数据附加奇偶校验位并且进行接收数据的奇偶校验。

当 SPCR.TXMD 位为“1”并且 SPCR2.SPPE 位为“1”时, 给发送数据附加奇偶校验位, 但是不进行接收数据的奇偶校验。

SPOE 位 (奇偶校验模式位)

此位设定偶校验或者奇校验。

在决定偶校验的奇偶校验位时, 需要使奇偶校验位和发送 / 接收字符中“1”的个数的合计为偶数。同样, 在决定奇校验的奇偶校验位时, 需要使奇偶校验位和发送 / 接收字符中“1”的个数的合计为奇数。

SPOE 位只在 SPPE 位为“1”时有效。

SPIIE 位 (RSPI 空闲中断允许位)

当检测到 RSPI 为空闲状态并且 SPSR.IDLNF 标志为“0”时, 此位允许或者禁止 RSPI 空闲中断请求的产生。

PTE 位 (奇偶校验自诊断位)

这是为了确认奇偶校验功能处于正常状态而将奇偶校验电路的自诊断置为有效的位。

29.2.14 RSPI 命令寄存器 0 ~ 7 (SPCMD0 ~ SPCMD7)

地址 RSPi0.SPCMD0 0008 8390h、RSPi0.SPCMD1 0008 8392h、RSPi0.SPCMD2 0008 8394h、RSPi0.SPCMD3 0008 8396h、
RSPi0.SPCMD4 0008 8398h、RSPi0.SPCMD5 0008 839Ah、RSPi0.SPCMD6 0008 839Ch、RSPi0.SPCMD7 0008 839Eh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SCKDE	SLNDE	SPNDE	LSBF	SPB[3:0]			SSLKP	SSLA[2:0]		BRDV[1:0]		CPOL	CPHA		
复位后的值	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1

位	符号	位名	功能	R/W
b0	CPHA	RSPCK 相位设定位	0: 在奇数边沿进行数据采样, 在偶数边沿数据发生变化。 1: 在奇数边沿数据发生变化, 在偶数边沿进行数据采样。	R/W
b1	CPOL	RSPCK 极性设定位	0: 空闲时的 RSPCK 为 Low 电平 1: 空闲时的 RSPCK 为 High 电平	R/W
b3-b2	BRDV[1:0]	位速率分频设定位	b3 b2 0 0: 选择基本位速率 0 1: 选择基本位速率的 2 分频 1 0: 选择基本位速率的 4 分频 1 1: 选择基本位速率的 8 分频	R/W
b6-b4	SSLA[2:0]	SSL 信号有效设定位	b6 b4 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 1 x x: 不能设定 x: Don't care	R/W
b7	SSLKP	SSL 信号电平保持位	0: 在传送结束时将全部 SSL 信号置为无效 1: 在传送结束后到下次存取开始前保持 SSL 信号的电平	R/W
b11-b8	SPB[3:0]	RSPI 数据长度设定位	b11 b8 0100 ~ 0111: 8 位 1 0 0 0: 9 位 1 0 0 1: 10 位 1 0 1 0: 11 位 1 0 1 1: 12 位 1 1 0 0: 13 位 1 1 0 1: 14 位 1 1 1 0: 15 位 1 1 1 1: 16 位 0 0 0 0: 20 位 0 0 0 1: 24 位 0010、0011: 32 位	R/W
b12	LSBF	RSPI LSB first 位	0: MSB first 1: LSB first	R/W
b13	SPNDEN	RSPI 下次存取延迟设定允许位	0: 下次存取延迟为 1 个 RSPCK+2 个 PCLK 1: 下次存取延迟为 RSPI 下次存取延迟寄存器 (SPND) 的设定值	R/W
b14	SLNDEN	SSL 无效延迟设定允许位	0: SSL 无效延迟为 1 个 RSPCK 1: SSL 无效延迟为 RSPI 从属选择无效延迟寄存器 (SSLND) 的设定值	R/W
b15	SCKDEN	RSPCK 延迟设定允许位	0: RSPCK 延迟为 1 个 RSPCK 1: RSPCK 延迟为 RSPI 时钟延迟寄存器 (SPCKD) 的设定值	R/W

SPCMDm 寄存器设定主控模式的 RSPI 传送格式。1 个通道的 RSPI 有 8 个 (SPCMD0 ~ SPCMD7) RSPI 命令寄存器。SPCMD0 寄存器的部分位用于设定从属模式的 RSPI 传送格式。主控模式的 RSPI 根据 SPSCR.SPSSLN[2:0] 位的设定, 按顺序参照 SPCMDm 寄存器并且执行由所参照的 SPCMDm 寄存器设定的串行传送。

必须在发送缓冲器为空 (未设定下次传送的数据) 的状态下, 在参照该 SPCMDm 寄存器来设定要发送的数据前设定 SPCMDm 寄存器。

能通过 SPSSR.SPCP[2:0] 位, 确认主控模式的 RSPI 正在参照的 SPCMDm 寄存器。如果在 SPCR.MSTR 位为“0”并且 SPCR.SPE 位为“1”的状态下改写 SPCMDm 寄存器, 就不保证以后的运行。

CPHA 位 (RSPCK 相位设定位)

此位设定主控模式 / 从属模式的 RSPI 的 RSPCK 相位。在进行 RSPI 模块之间的数据通信时, 需要在模块之间设定相同的 RSPCK 相位。

CPOL 位 (RSPCK 极性设定位)

此位设定主控模式 / 从属模式的 RSPI 的 RSPCK 极性。在进行 RSPI 模块之间的数据通信时, 需要在模块之间设定相同的 RSPCK 极性。

BRDV[1:0] 位 (位速率分频设定位)

这些位设定位速率。位速率取决于 BRDV[1:0] 位和 SPBR 寄存器设定值的组合 (参照“29.2.8 RSPI 位速率寄存器 (SPBR)”)。SPBR 寄存器的设定值决定基本位速率。BRDV[1:0] 位的设定值选择基本位速率的无分频、2 分频、4 分频或者 8 分频。能分别对 SPCMDm 寄存器设定不同的 BRDV[1:0] 位。因此, 各命令能以不同的位速率进行串行传送。

SSLA[2:0] 位 (SSL 信号有效设定位)

这些位控制主控模式的 RSPI 进行串行传送时的 SSLAi 信号的有效。SSLA[2:0] 位的设定值控制 SSLAi 信号的有效。SSLAi 信号有效时的信号极性取决于 SSLP 寄存器的设定值。如果在多主控模式中将 SSLA[2:0] 位置“000b”, 就在全部 SSL 信号无效的状态下进行串行传送 (SSLA0 引脚为输入引脚)。

在从属模式中使用 RSPI 时, 必须将 SSLA[2:0] 位置“000b”。

SSLKP 位 (SSL 信号电平保持位)

在主控模式的 RSPI 进行串行传送时, 此位设定在从当前命令对应的 SSL 无效到下一个命令对应的 SSL 有效的期间是保持当前命令的 SSLAi 信号电平还是将此电平置为无效。

在从属模式中使用 RSPI 时, 必须将 SSLKP 位置“0”。

SPB[3:0] 位 (RSPI 数据长度设定位)

这些位设定主控模式 / 从属模式的 RSPI 传送数据长度。

LSBF 位 (RSPI LSB first 位)

此位设定是将主控模式 / 从属模式的 RSPI 数据格式设定为 MSB first 还是 LSB first。

SPNDEN 位 (RSPI 下次存取延迟允许位)

此位设定在从属模式的 RSPI 结束串行传送后从 SSLAi 信号无效到下次存取的 SSLAi 信号有效的期间 (下次存取延迟)。当 SPNDEN 位为“0”时, RSPI 将下次存取延迟设定为 1 个 RSPCK+2 个 PCLK; 当 SPNDEN 位为“1”时, RSPI 根据 SPND 寄存器的设定, 插入下次存取延迟。

在从属模式中使用 RSPI 时, 必须将 SPNDEN 位置“0”。

SLNDEN 位 (SSL 无效延迟设定允许位)

此位设定主控模式的 RSPI 从 RSPCK 振荡停止到 SSLAi 信号无效的期间 (SSL 无效延迟)。当 SLNDEN 位为“0”时, RSPI 将 SSL 无效延迟设定为 1 个 RSPCK; 当 SLNDEN 位为“1”时, RSPI 在经过 SSLND 寄存器设定的 RSPCK 延迟后将 SSL 置为无效。

在从属模式中使用 RSPI 时, 必须将 SLNDEN 位置“0”。

SCKDEN 位 (RSPCK 延迟设定允许位)

此位设定主控模式的 RSPI 从 SSLAi 信号有效到 RSPCK 振荡为止的期间 (RSPCK 延迟)。当 SCKDEN 位为“0”时, RSPI 将 RSPCK 延迟设定为 1 个 RSPCK; 当 SCKDEN 位为“1”时, RSPI 在经过 SPCKD 寄存器设定的 RSPCK 延迟后开始 RSPCK 的振荡。

在从属模式中使用 RSPI 时, 必须将 SCKDEN 位置“0”。

29.3 运行说明

在本节中，串行传送期间是指从开始驱动有效数据到取最后有效数据的期间。

29.3.1 RSPI 运行概要

RSPI 能在从属模式 (SPI 运行)、单主控模式 (SPI 运行)、多主控模式 (SPI 运行)、从属模式 (时钟同步运行) 和主控模式 (时钟同步运行) 中进行同步串行传送。能通过 SPCR.MSTR 位和 SPCR.MODFEN 位和 SPCR.SPMS 位设定 RSPI 的模式。RSPI 的模式和 SPCR 寄存器设定的关系以及各模式概要如表 29.5 所示。

表 29.5 RSPI 的模式和 SPCR 寄存器设定的关系以及各模式概要

模式	从属 (SPI 运行)	单主控 (SPI 运行)	多主控 (SPI 运行)	从属 (时钟同步运行)	主控 (时钟同步运行)
MSTR 位的设定	0	1	1	0	1
MODFEN 位的设定	0 or 1	0	1	0	0
SPMS 位的设定	0	0	0	1	1
RSPCKA 信号	输入	输出	输出 /Hi-Z	输入	输出
MOSIA 信号	输入	输出	输出 /Hi-Z	输入	输出
MISOA 信号	输出 /Hi-Z	输入	输入	输出	输入
SSLA0 信号	输入	输出	输入	Hi-Z (注1)	Hi-Z (注1)
SSLA1 ~ SSLA3 信号	Hi-Z (注1)	输出	输出 /Hi-Z	Hi-Z (注1)	Hi-Z (注1)
SSL 极性变更功能	有	有	有	—	—
传送率	~ PCLK/8	~ PCLK/2	~ PCLK/2	~ PCLK/8	~ PCLK/2
时钟源	RSPCK 输入	内部波特率发生器	内部波特率发生器	RSPCK 输入	内部波特率发生器
时钟极性	2 种	2 种	2 种	2 种	2 种
时钟相位	2 种	2 种	2 种	1 种 (CPHA=1)	2 种
开始传送的位	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
传送数据长度	8 ~ 32 位	8 ~ 32 位	8 ~ 32 位	8 ~ 32 位	8 ~ 32 位
突发传送	能 (CPHA=1)	能 (CPHA=0、1)	能 (CPHA=0、1)	—	—
RSPCK 延迟控制	无	有	有	无	有
SSL 无效延迟控制	无	有	有	无	有
下次存取延迟控制	无	有	有	无	有
传送启动方法	SSL 输入有效或者 RSPCK 振荡	通过发送缓冲器空中断请求，写发送缓冲器	通过发送缓冲器空中断请求，写发送缓冲器	RSPCK 振荡	通过发送缓冲器空中断请求，写发送缓冲器
顺序控制	无	有	有	无	有
发送缓冲器空检测	有	有	有	有	有
接收缓冲器满检测	有 (注2)	有 (注2)	有 (注2)	有 (注2)	有 (注2)
溢出错误检测	有 (注2)	有 (注2)	有 (注2)	有 (注2)	有 (注2)
奇偶校验错误检测	有 (注2) (注3)	有 (注2) (注3)	有 (注2) (注3)	有 (注2) (注3)	有 (注2) (注3)
模式故障错误检测	有 (MODFEN=1)	无	有	无	无

注 1. 在此模式中不使用。

注 2. 当 SPCR.TXMD 位为“1”时，不进行接收缓冲器满的检测、溢出错误的检测和奇偶校验错误的检测。

注 3. 当 SPCR2.SPPE 位为“0”时，不进行奇偶校验错误的检测。

29.3.2 RSPI 引脚的控制

RSPI 通过设定 SPCR.MSTR 位、SPCR.MODFEN 位、SPCR.SPMS 位来转换引脚状态。如果将 I/O 端口的 ODRn.Bi 位的设定值置“0”，RSPI 引脚就变为 CMOS 输出；如果置“1”，RSPI 引脚就变为漏极开路输出。引脚状态和各位设定值的关系如表 29.6 所示。I/O 端口的设定也必须相同。

表 29.6 RSPI 引脚的状态和控制位设定值的关系

模式	引脚	引脚状态 (注2)
单主控 (SPI 运行) (MSTR=1、MODFEN=0、SPMS=0)	RSPCKA	输出
	SSLA0 ~ 3	输出
	MOSIA	输出
	MISOA	输入
多主控 (SPI 运行) (MSTR=1、MODFEN=1、SPMS=0)	RSPCKA (注3)	输出 /Hi-Z
	SSLA0	输入
	SSLA1 ~ 3 (注3)	输出 /Hi-Z
	MOSIA (注3)	输出 /Hi-Z
	MISOA	输入
从属 (SPI 运行) (MSTR=0、SPMS=0)	RSPCKA	输入
	SSLA0	输入
	SSLA1 ~ 3 (注5)	Hi-Z (注1)
	MOSIA	输入
	MISOA (注4)	输出 /Hi-Z
主控 (时钟同步运行) (MSTR=1、MODFEN=0、SPMS=1)	RSPCKA	输出
	SSLA0 ~ 3 (注5)	Hi-Z (注1)
	MOSIA	输出
	MISOA	输入
从属 (时钟同步运行) (MSTR=0、SPMS=1)	RSPCKA	输入
	SSLA0 ~ 3 (注5)	Hi-Z (注1)
	MOSIA	输入
	MISOA	输出

注 1. 在此模式中不使用。

注 2. 未选择 RSPI 功能的多功能引脚不反映 RSPI 的设定值。

注 3. 当 SSLA0 为有效电平时，引脚状态为 Hi-Z。

注 4. 当 SSLA0 为无效电平或者 SPCR.SPE 位为“0”时，引脚状态为 Hi-Z。

注 5. 能用作 I/O 端口。

如表 29.7 所示，单主控模式 (SPI 运行) 和多主控模式 (SPI 运行) 的 RSPI 根据 SPPCR.MOIFE 位和 SPPCR.MOIFV 位的设定，决定 SSL 无效期间 (包括突发传送中的 SSL 保持期间) 的 MOSI 信号值。

表 29.7 SSL 无效期间的 MOSI 信号值的决定方法

MOIFE 位	MOIFV 位	SSL 无效期间的 MOSI 信号值
0	0、1	上次传送的最后数据
1	0	总是为 Low 电平。
1	1	总是为 High 电平。

29.3.3 RSPI 系统结构例子

29.3.3.1 单主控模式 / 单从属模式 (本 LSI= 主控设备)

本 LSI 用作主控设备时的单主控模式 / 单从属模式的 RSPI 系统结构例子如图 29.5 所示。在单主控模式 / 单从属模式的系统结构例子中，不使用本 LSI (主控设备) 的 SSLA0 ~ SSLA3 输出。通过将 RSPI 从属设备的 SSL 输入固定为 Low 电平，设定为能随时选择 RSPI 从属模式的状态 (注 1)。

本 LSI (主控设备) 随时驱动 RSPCKA 和 MOSIA，RSPI 从属设备随时驱动 MISO。

注 1. 在类似 SPCMDm.CPHA 位为“0”时的传送格式中，也有些从属设备不能将 SSL 信号固定为有效电平。在不能固定 SSL 信号时，必须将本 LSI 的 SSLAi 输出连接到从属设备的 SSL 输入。

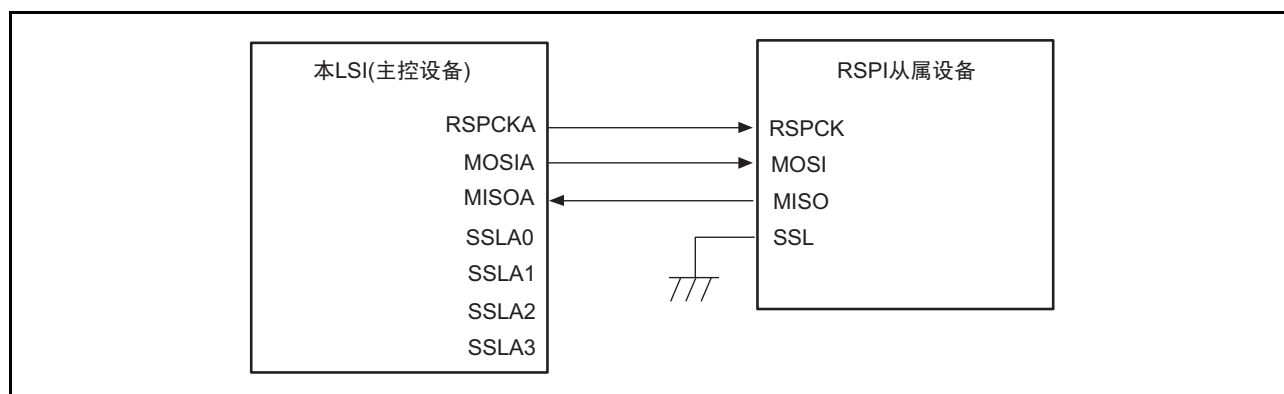


图 29.5 单主控模式 / 单从属模式的系统结构例子 (本 LSI= 主控设备)

29.3.3.2 单主控模式 / 单从属模式 (本 LSI= 从属设备)

将本 LSI 用作从属设备时的单主控模式 / 单从属模式的 RSPI 系统结构例子如图 29.6 所示。在将本 LSI 用作从属设备时，将 SSLA0 引脚用作 SSL 输入。RSPI 主控设备随时驱动 RSPCK 和 MOSI，本 LSI (从属设备) 随时驱动 MISOA (注 1)。

在 SPCMDm.CPHA 位为“1”的单从属模式的系统结构中，通过将本 LSI (从属设备) 的 SSLA0 输入引脚固定为 Low 电平，设定为能随时选择本 LSI (从属设备) 的状态并且也能进行串行传送 (图 29.7)。

注 1. 在 SSLA0 为无效电平时，引脚状态为 Hi-Z。

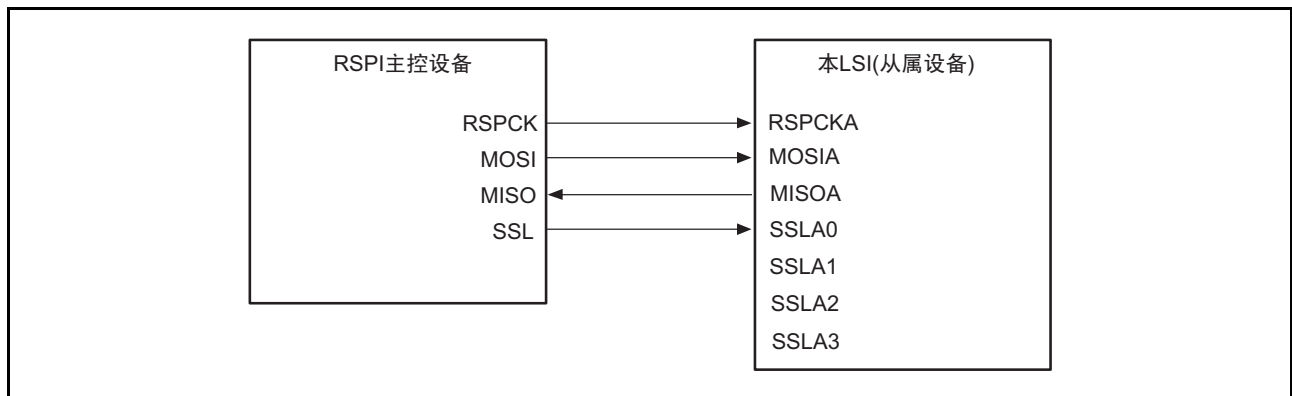


图 29.6 单主控模式 / 单从属模式的系统结构例子 (本 LSI= 从属设备、CPHA=0)

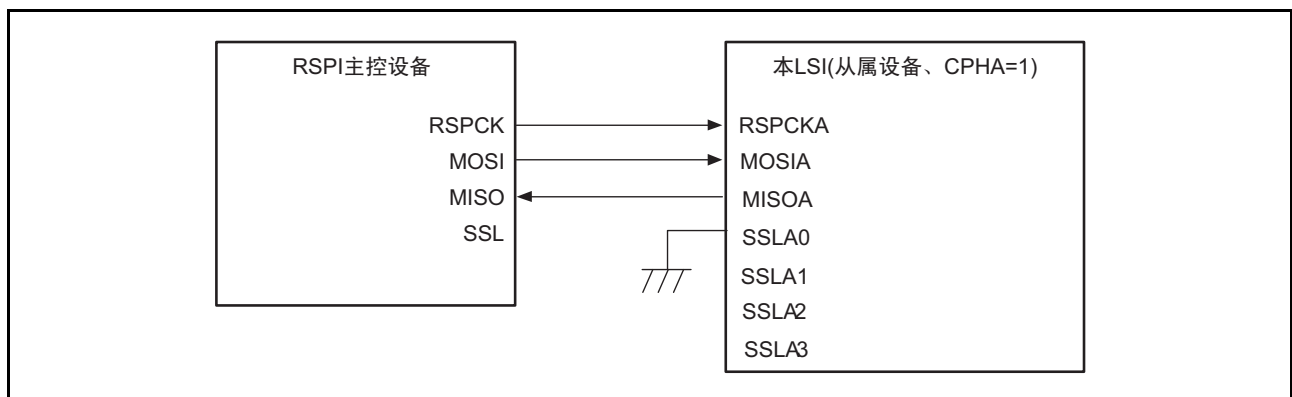


图 29.7 单主控模式 / 单从属模式的系统结构例子 (本 LSI= 从属设备、CPHA=1)

29.3.3.3 单主控模式 / 多从属模式 (本 LSI= 主控设备)

本 LSI 用作主控设备时的单主控模式 / 多从属模式的 RSPI 系统结构例子如图 29.8 所示。在图 29.8 的例子中，RSPI 系统由本 LSI (主控设备) 和 4 个从属设备 (RSPI 从属设备 0 ~ RSPI 从属设备 3) 构成。

将本 LSI (主控设备) 的 RSPCKA 输出和 MOSIA 输出连接 RSPI 从属设备 0 ~ RSPI 从属设备 3 的 RSPCK 输入和 MOSI 输入，RSPI 从属设备 0 ~ RSPI 从属设备 3 的 MISO 输出全部连接本 LSI (主控设备) 的 MISOA 输入，本 LSI (主控设备) 的 SSLA0 ~ SSLA3 输出分别连接 RSPI 从属设备 0 ~ RSPI 从属设备 3 的 SSL 输入。

本 LSI (主控设备) 随时驱动 RSPCK、MOSI 和 SSLA0 ~ SSLA3。在 RSPI 从属设备 0 ~ 从属设备 3 中，SSL 输入为 Low 电平的从属设备驱动 MISO。

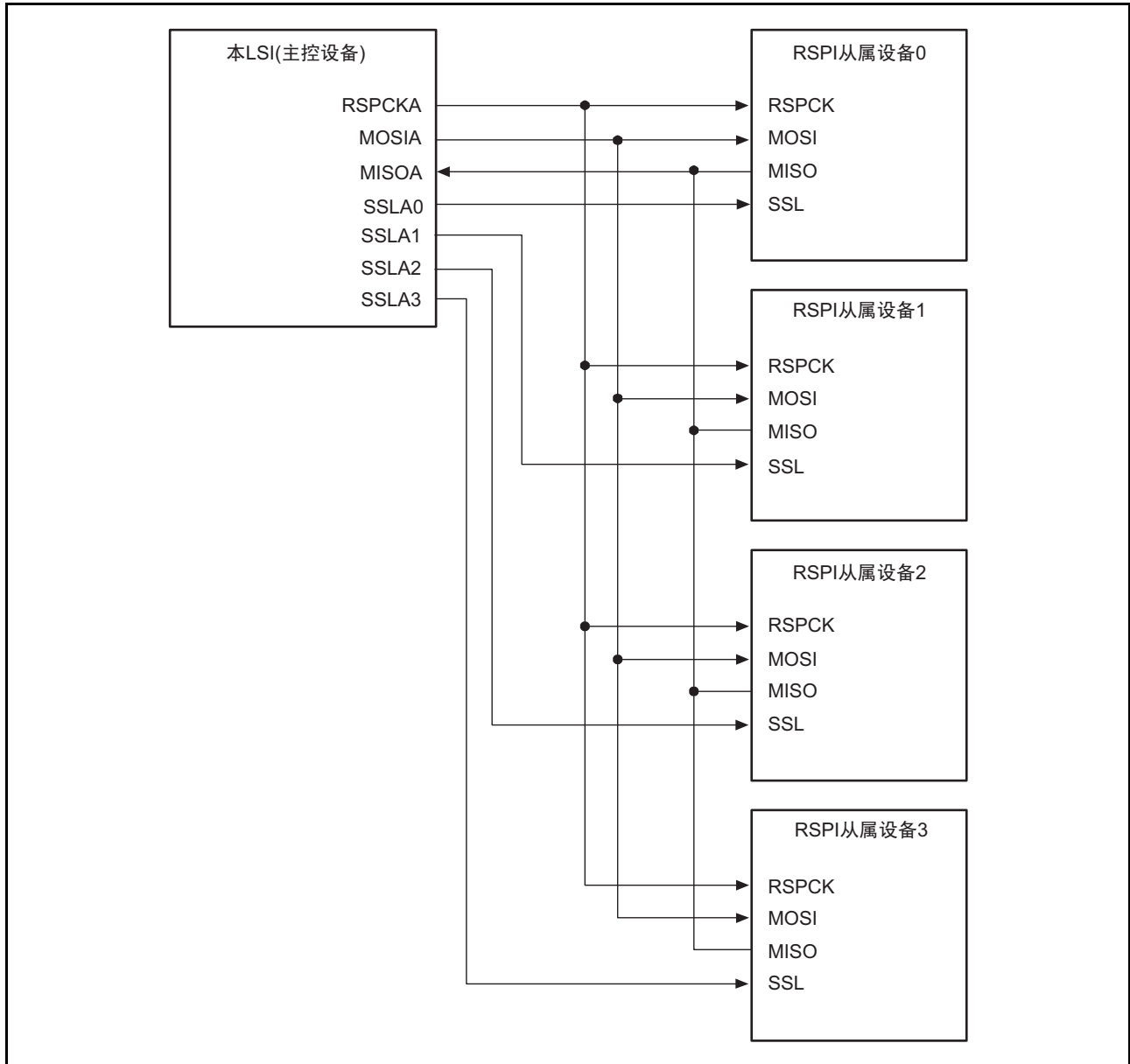


图 29.8 单主控模式 / 多从属模式的系统结构例子 (本 LSI= 主控设备)

29.3.3.4 单主控模式 / 多从属模式 (本 LSI= 从属设备)

本 LSI 用作从属设备时的单主控模式 / 多从属模式的 RSPI 系统结构例子如图 29.9 所示。在图 29.9 的例子中，RSPI 系统由 RSPI 主控设备和 2 个本 LSI (从属设备 X、从属设备 Y) 构成。

RSPI 主控设备的 RSPCK 输出和 MOSI 输出连接本 LSI (从属设备 X、从属设备 Y) 的 RSPCKA 输入和 MOSIA 输入，本 LSI (从属设备 X、从属设备 Y) 的 MISOA 输出连接 RSPI 主控设备的 MISO 输入，RSPI 主控设备的 SSLX 输出和 SSLY 输出连接本 LSI (从属设备 X、从属设备 Y) 的 SSLA0 输入。

RSPI 主控设备随时驱动 RSPCK、MOSI、SSLX 和 SSLY。在本 LSI (从属设备 X、从属设备 Y) 中，SSLA0 输入为 Low 电平的从属设备驱动 MISOA。

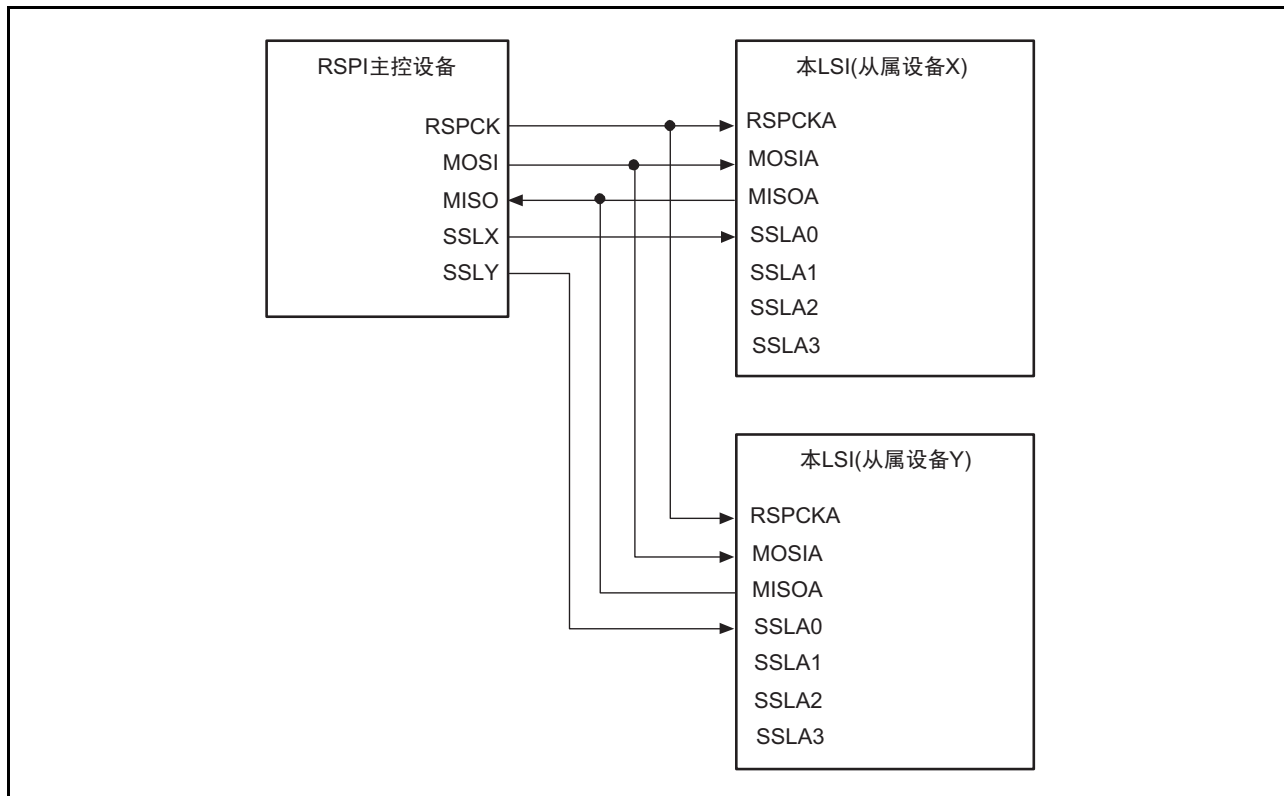


图 29.9 单主控模式 / 多从属模式的系统结构例子 (本 LSI= 从属设备)

29.3.3.5 多主控模式 / 多从属模式 (本 LSI= 主控设备)

本 LSI 用作主控设备时的多主控模式 / 多从属模式的 RSPI 系统结构例子如图 29.10 所示。在图 29.10 的例子中，RSPI 系统由 2 个本 LSI (主控设备 X、主控设备 Y) 和 2 个 RSPI 从属设备 (RSPI 从属设备 1、RSPI 从属设备 2) 构成。

本 LSI (主控设备 X、主控设备 Y) 的 RSPCKA 输出和 MOSIA 输出连接 RSPI 从属设备 1 和 RSPI 从属设备 2 的 RSPCK 输入和 MOSI 输入，RSPI 从属设备 1 和 RSPI 从属设备 2 的 MISO 输出连接本 LSI (主控设备 X、主控设备 Y) 的 MISOA 输入，本 LSI (主控设备 X) 任意的通用端口 Y 输出引脚连接本 LSI (主控设备 Y) 的 SSLA0 输入，本 LSI (主控设备 Y) 任意的通用端口 X 输出连接本 LSI (主控设备 X) 的 SSLA0 输入，本 LSI (主控设备 X、主控设备 Y) 的 SSLA1 输出和 SSLA2 输出连接 RSPI 从属设备 1 和 RSPI 从属设备 2 的 SSL 输入。在此结构例子中，因为系统只由 SSLA0 输入、连接从属设备的 SSLA1 输出和 SSLA2 输出构成，所以不使用本 LSI 的 SSLA3 输出。

当 SSLA0 输入电平为 High 电平时，本 LSI 驱动 RSPCKA、MOSIA、SSLA1 和 SSLA2；当 SSLA0 输入电平为 Low 电平时，检测模式故障错误，在将 RSPCKA、MOSIA、SSLA1 和 SSLA2 置为 Hi-Z 后，将 RSPI 总线权释放给其他主控设备。在 RSPI 从属设备 1 和 RSPI 从属设备 2 中，SSL 输入为 Low 电平的从属设备驱动 MISO。

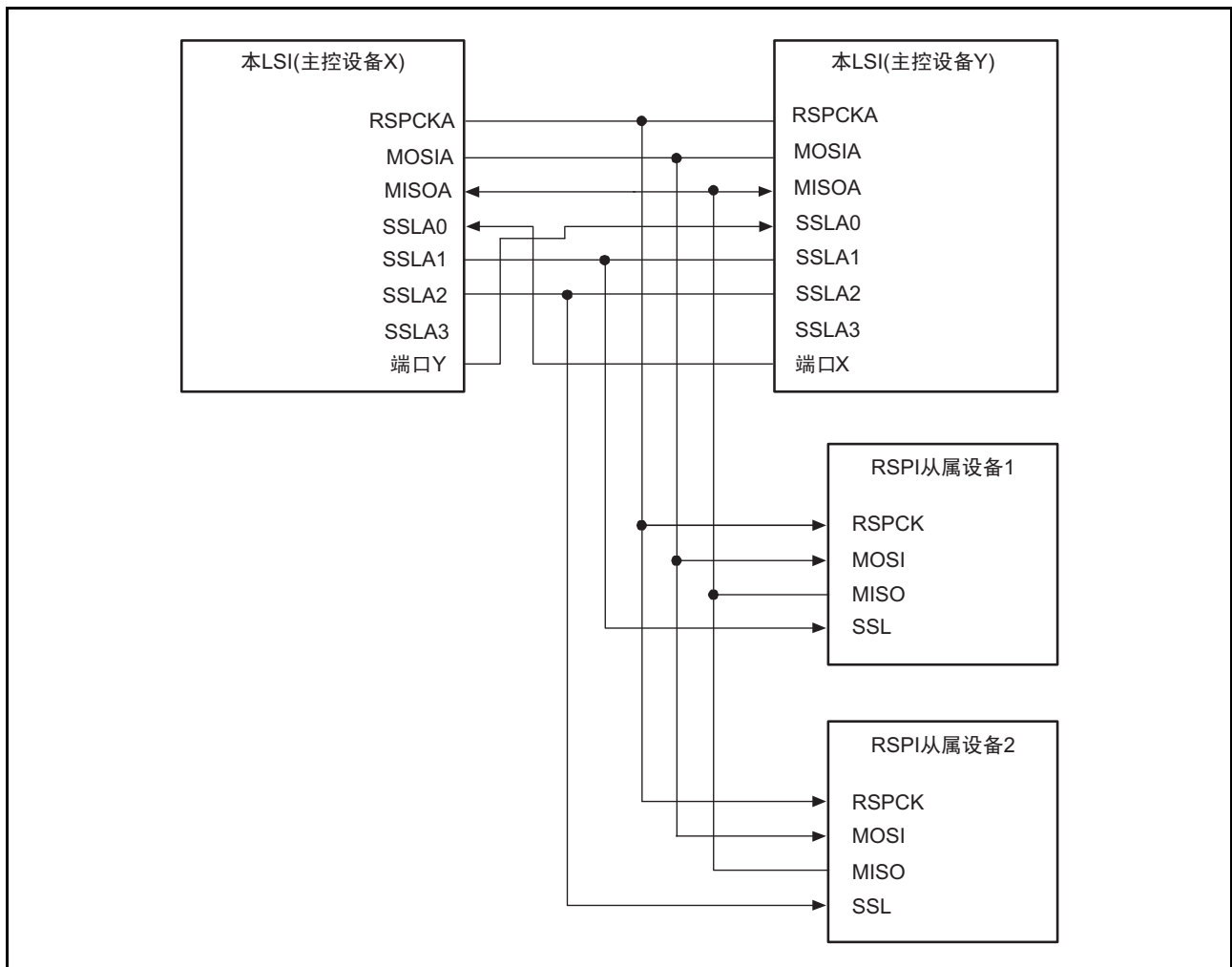


图 29.10 多主控模式 / 多从属模式的系统结构例子 (本 LSI= 主控设备)

29.3.3.6 主控模式 (时钟同步运行) / 从属模式 (时钟同步运行) (本 LSI= 主控设备)

本 LSI 用作主控设备时的主控模式 (时钟同步运行) / 从属模式 (时钟同步运行) 的 RSPI 系统结构例子如图 29.11 所示。在 主控模式 (时钟同步运行) / 从属模式 (时钟同步运行) 的结构中, 不使用本 LSI (主控设备) 的 SSLA0 ~ SSLA3。

本 LSI (主控设备) 随时启动 RSPCKA 和 MOSIA, RSPI 从属设备随时驱动 MISO。

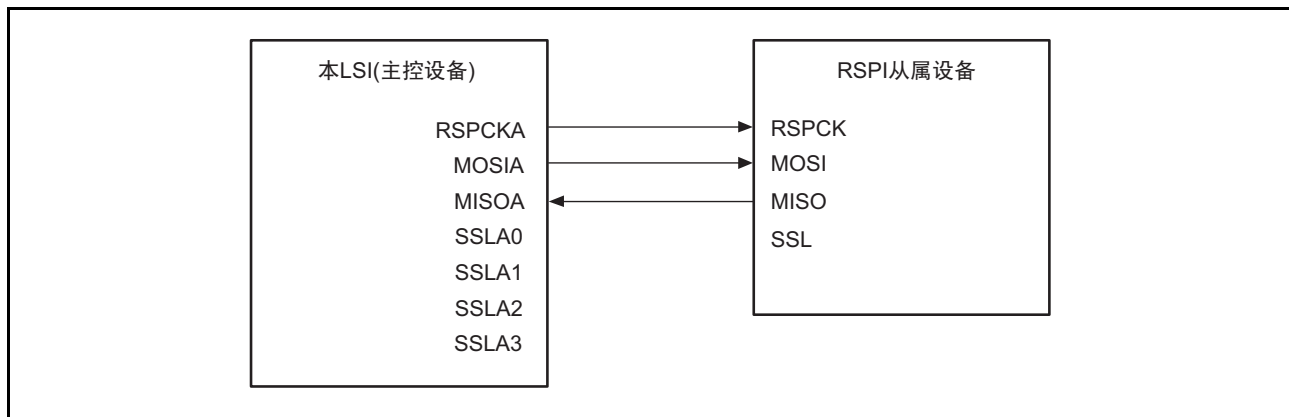


图 29.11 主控模式 (时钟同步运行) / 从属模式 (时钟同步运行) 的系统结构例子 (本 LSI= 主控设备)

29.3.3.7 主控模式 (时钟同步运行) / 从属模式 (时钟同步运行) (本 LSI= 从属设备)

本 LSI 用作从属设备时的主控模式 (时钟同步运行) / 从属模式 (时钟同步运行) 的 RSPI 系统结构例子如图 29.12 所示。在将本 LSI 用作从属设备 (时钟同步运行) 时, 本 LSI (从属设备) 随时驱动 MISOA, RSPI 主控设备随时驱动 RSPCK 和 MOSI。不使用本 LSI (从属设备) 的 SSLA0 ~ SSLA3。

只有在将 SPCMDm.CPHA 位置“1”的单从属模式的系统结构中, 本 LSI (从属设备) 才能进行串行传送。

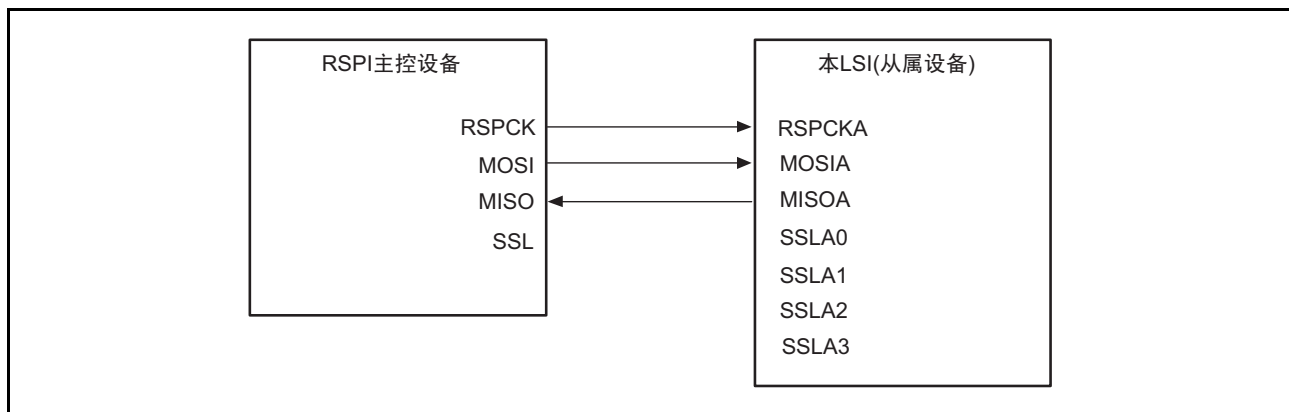


图 29.12 主控模式 (时钟同步运行) / 从属模式 (时钟同步运行) 的系统结构例子 (本 LSI= 从属设备, CPHA=1)

29.3.4 数据格式

RSPI 的数据格式取决于 RSPI 命令寄存器 m (SPCMD m) ($m=0 \sim 7$) 和 RSPI 控制寄存器 2 的奇偶校验允许位 (SPCR2.SPPE) 设定值。与 MSB/LSB first 无关, RSPI 将 RSPI 数据寄存器 (SPDR) 的 LSB 位开始的设定数据长度的范围作为传送数据进行处理。

发送和接收时的 1 帧数据格式如下所示。

(a) 奇偶校验功能无效的情况

在奇偶校验功能无效时, 进行 RSPI 命令寄存器 m 的 RSPI 数据长度设定位 (SPCMD m .SPB[3:0]) 所设位长的数据发送和接收。

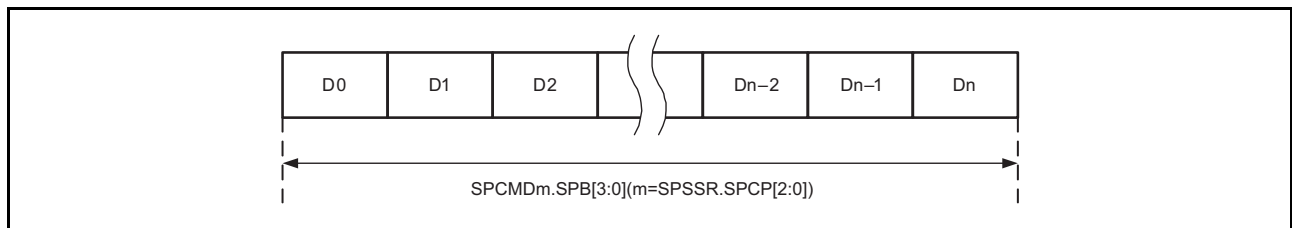


图 29.13 数据格式概要 (奇偶校验功能无效的情况)

(b) 奇偶校验功能有效的情况

在奇偶校验功能有效时, 进行 SPCMD m .SPB[3:0] 所设位长的数据发送和接收。但是, 最后位为奇偶校验位。

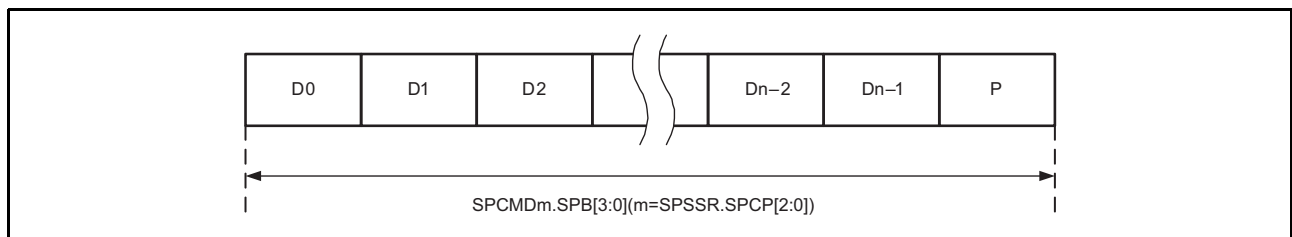


图 29.14 数据格式概要 (奇偶校验功能有效的情况)

29.3.4.1 奇偶校验功能无效的情况 (SPCR2.SPPE=0)

在奇偶校验功能无效时，将发送缓冲器的数据复制到移位寄存器而不进行加工。根据 MSB/LSB first 和位长的组合，对 RSPI 数据寄存器 (SPDR) 和移位寄存器的关系进行以下的说明。

(1) MSB first 传送 (32 位数据)

在奇偶校验功能无效的情况下，RSPI 进行 32 位数据的 MSB first 传送时的 SPDR 寄存器和移位寄存器的运行内容如图 29.15 所示。

在发送时，将发送缓冲器的 T31 ~ T00 复制到移位寄存器。按 T31→T30→…→T00 的顺序将移位寄存器的值进行移位，并且发送数据。

在接收时，将接收数据保存到移位寄存器的 bit0，并且按数据进行接收数据的移位。输入需要的 RSPCK，如果收集到 R31 ~ R00 的数据，就将移位寄存器的值复制到接收缓冲器。

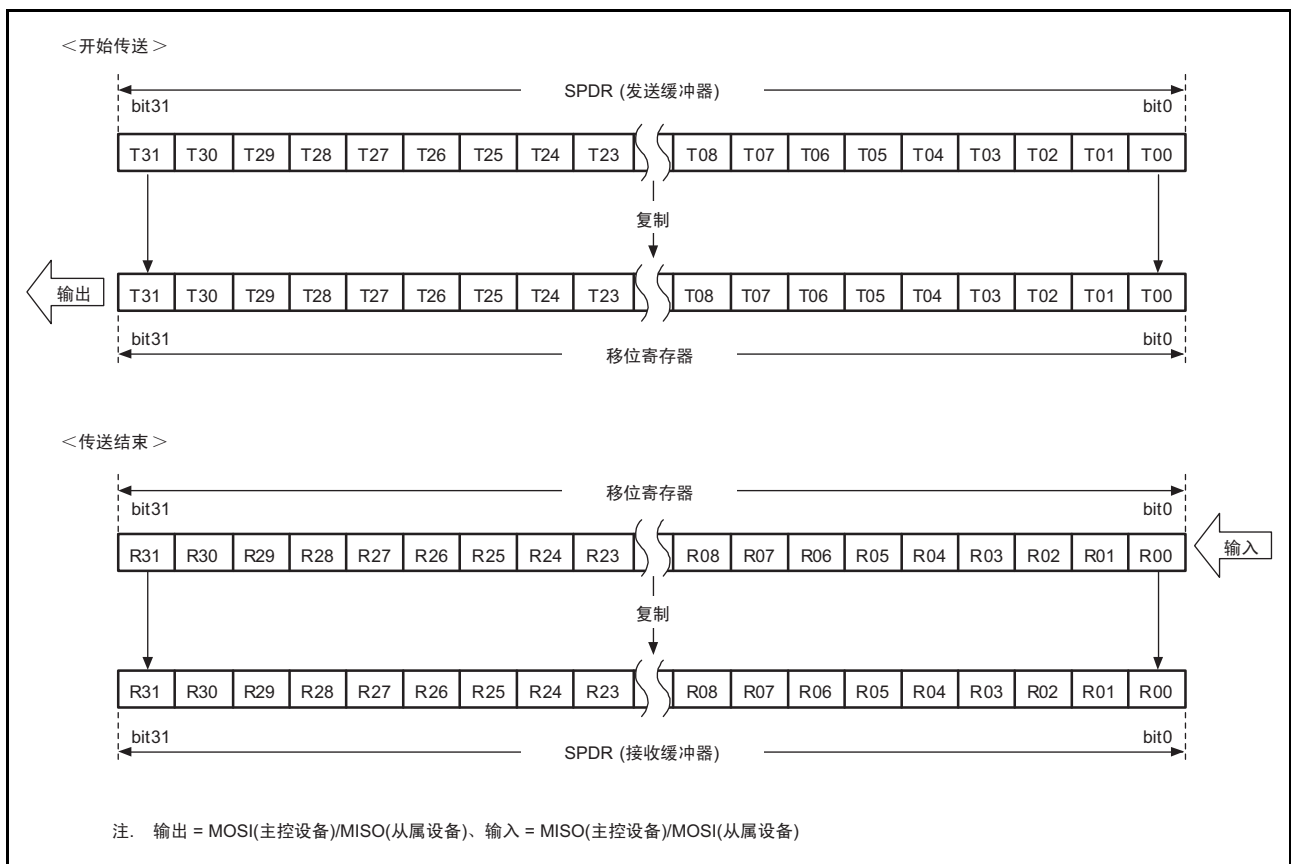


图 29.15 MSB first 传送 (32 位数据 / 奇偶校验功能无效)

(2) MSB first 传送 (24 位数据)

在奇偶校验功能无效的情况下，以对 32 位以外的数据进行 MSB first 传送为例，RSPI 进行 24 位数据传送时的 RSPI 数据寄存器 (SPDR) 和移位寄存器的运行内容如图 29.16 所示。

在发送时，将发送缓冲器的低 24 位 (T23 ~ T00) 复制到移位寄存器。按 T23→T22→...→T00 的顺序将移位寄存器的值进行移位，并且发送数据。

在接收时，将接收数据保存到移位寄存器的 bit0，并且按数据进行接收数据的移位。输入需要的 RSPCK，如果收集到 R23 ~ R00 的数据，就将移位寄存器的值复制到接收缓冲器。此时，将发送缓冲器的高 8 位保存到接收缓冲器的高 8 位。能在发送时预先给 T31 ~ T24 写“0”，使接收缓冲器的高 8 位变为“0”。

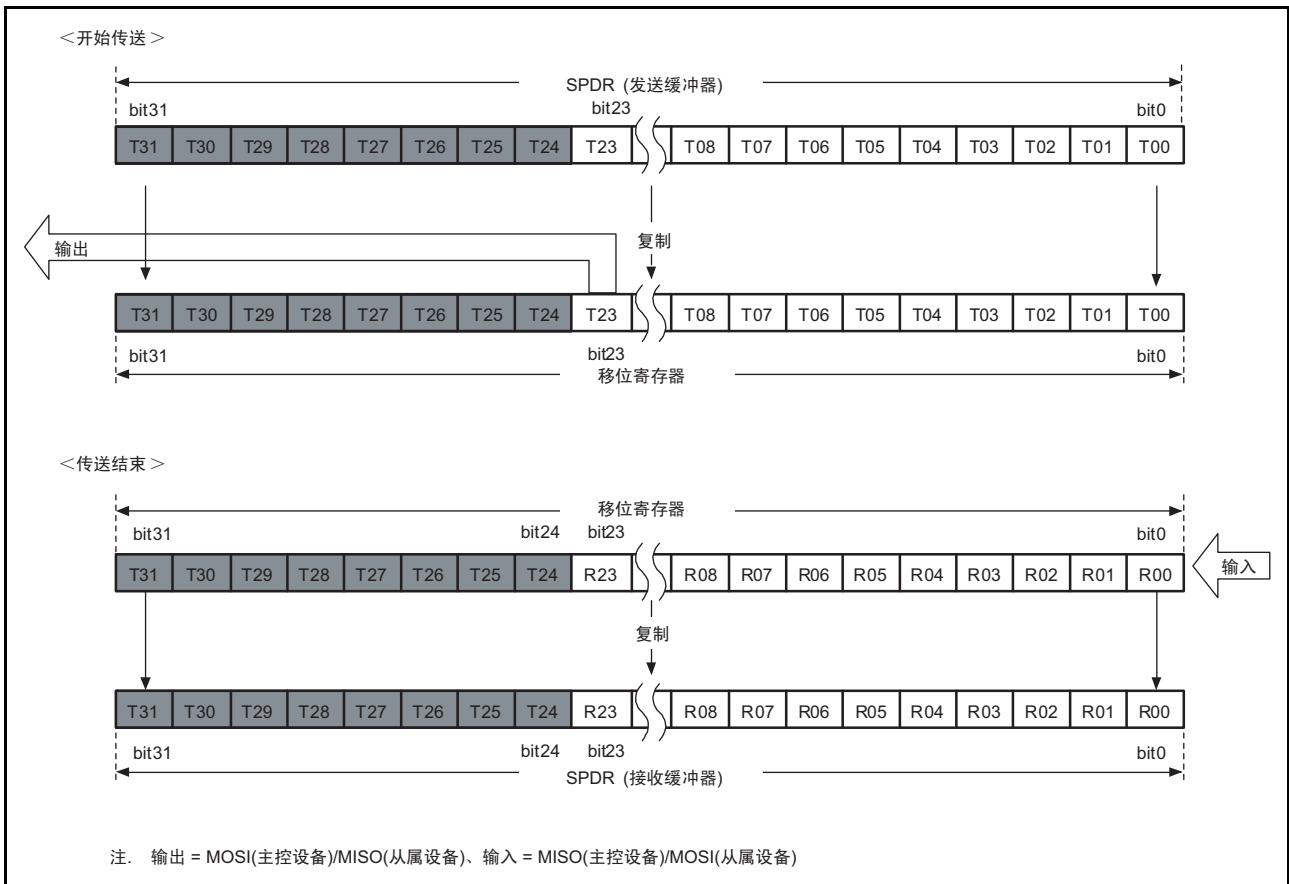


图 29.16 MSB first 传送 (24 位数据 / 奇偶校验功能无效)

(3) LSB first 传送 (32 位数据)

在奇偶校验功能无效的情况下，RSPI 进行 32 位数据长度的 LSB first 传送时的 RSPI 数据寄存器 (SPDR) 和移位寄存器的运行内容如图 29.17 所示。

在发送时，以位为单位将发送缓冲器的数据 (T31 ~ T00) 调换为 T00 ~ T31 的顺序，并且复制到移位寄存器。按 T00→T01→...→T31 的顺序将移位寄存器的值进行移位，并且发送数据。

在接收时，将最初的数据保存到移位寄存器的 bit0，并且按数据进行接收数据的移位。输入需要的 RSPCK，如果收集到 R00 ~ R31 的数据，就将移位寄存器的值复制到接收缓冲器。

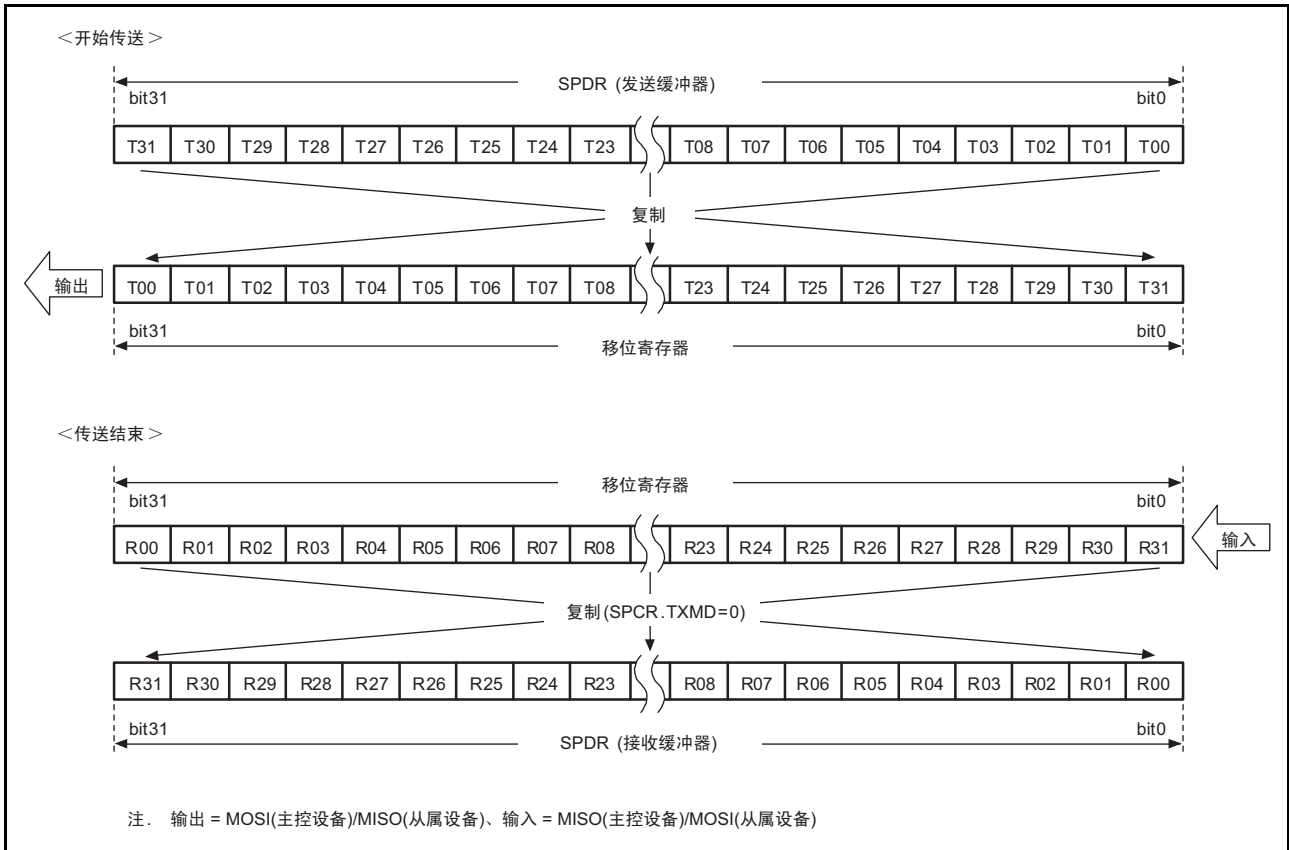


图 29.17 LSB first 传送 (32 位数据 / 奇偶校验功能无效)

(4) LSB first 传送 (24 位数据)

在奇偶校验功能无效的情况下, 以对 32 位以外的数据进行 LSB first 传送为例, RSPI 进行 24 位数据传送时的 RSPI 数据寄存器 (SPDR) 和移位寄存器的运行内容如图 29.18 所示。

在发送时, 以位为单位将发送缓冲器的低 24 位 (T23 ~ T00) 调换为 T00 ~ T23 的顺序, 并且复制到移位寄存器。按 T00→T01→…→T23 的顺序将移位寄存器的值进行移位, 并且发送数据。

在接收时, 将接收数据保存到移位寄存器的 bit8, 并且按数据进行接收数据的移位。输入需要的 RSPCK, 如果收集到 R00 ~ R23 的数据, 就将移位寄存器的值复制到接收缓冲器。

此时, 将发送缓冲器的高 8 位保存到接收缓冲器的高 8 位。能在发送时预先给 T31 ~ T24 写“0”, 使接收缓冲器的高 8 位变为“0”。

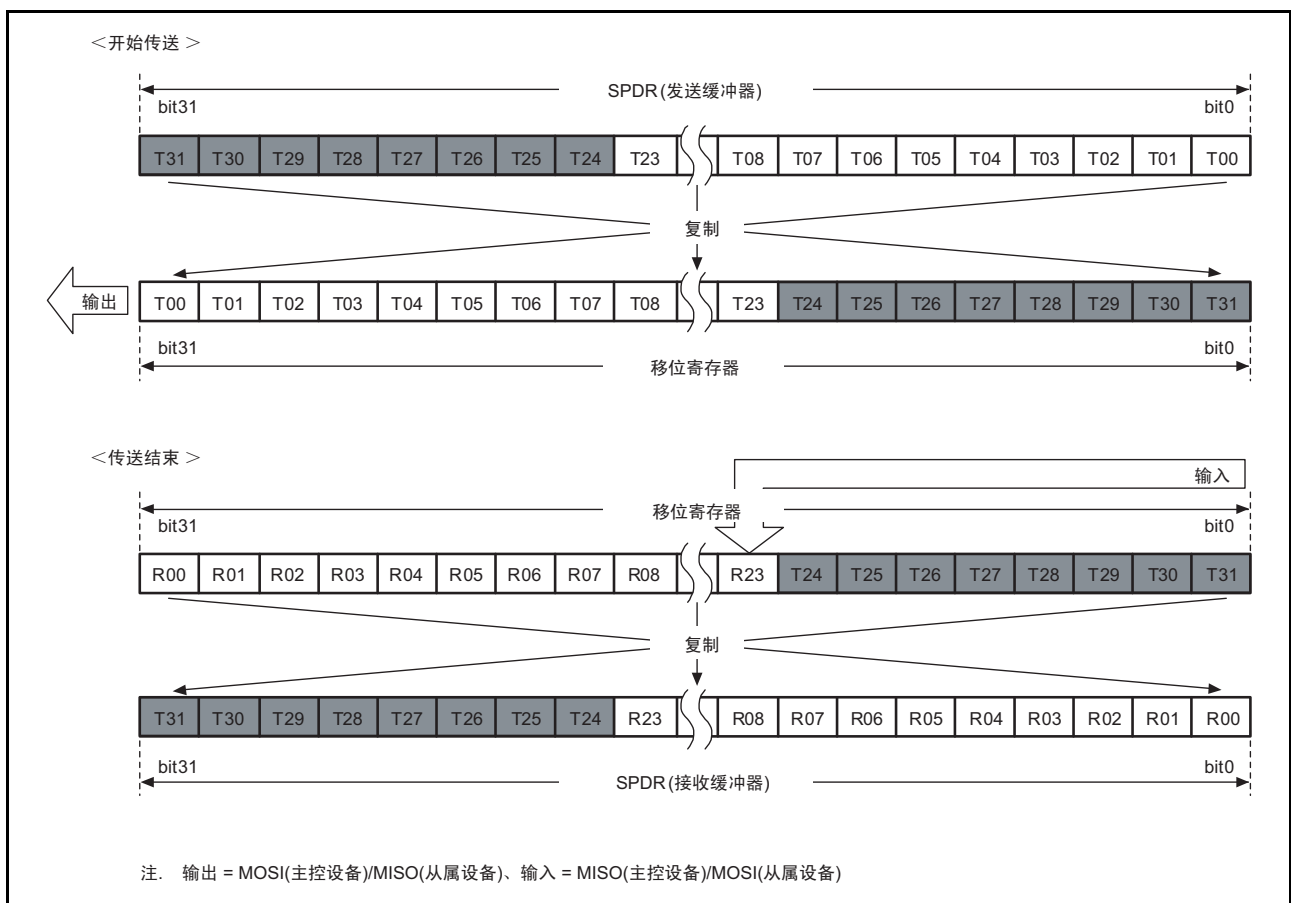


图 29.18 LSB first 传送 (24 位数据 / 奇偶校验功能无效)

29.3.4.2 奇偶校验功能有效的情况 (SPCR2.SPPE=1)

在奇偶校验功能有效时，将发送和接收数据的最低位转换为奇偶校验位。奇偶校验位的值由硬件进行计算和转换。

(1) MSB first 传送 (32 位数据)

在奇偶校验功能有效的情况下，RSPI 进行 32 位数据长度的 MSB first 传送时的 RSPI 数据寄存器 (SPDR) 和移位寄存器的运行内容如图 29.19 所示。

在发送时，先根据 T31 ~ T01 的数据进行奇偶校验位 (P) 的运算，替换最后位的 T00，然后复制到移位寄存器。按 T31→T30→...T01→P 的顺序发送数据。

在接收时，将接收数据保存到移位寄存器的 bit0，并且按数据进行接收数据的移位。输入需要的 RSPCK，如果收集到 R31 ~ P 的数据，就将移位寄存器的值复制到接收缓冲器。如果将数据复制到移位寄存器，就检查 R31 ~ P 的数据，判断奇偶校验错误。

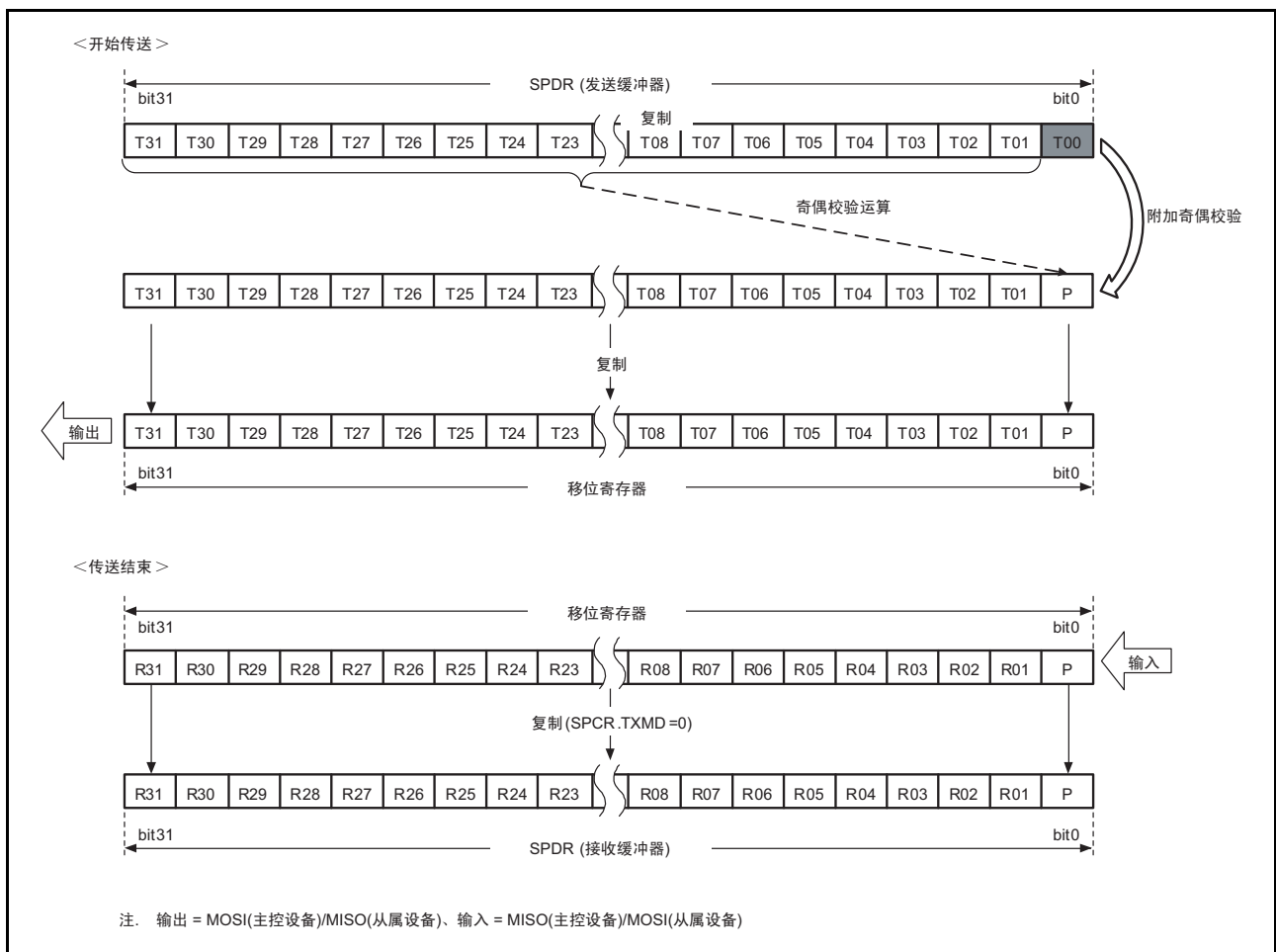


图 29.19 MSB first 传送 (32 位数据 / 奇偶校验功能有效)

(2) MSB first 传送 (24 位数据)

在奇偶校验功能有效的情况下，以对 32 位以外的数据进行 MSB first 传送为例，RSPI 进行 24 位数据传送时的 RSPI 数据寄存器 (SPDR) 和移位寄存器的运行内容如图 29.20 所示。

在发送时，先根据 T23 ~ T01 的数据进行奇偶校验位 (P) 的运算，替换最后位的 T00，然后复制到移位寄存器。按 T23→T22→…T01→P 的顺序发送数据。

在接收时，将接收数据保存到移位寄存器的 bit0，并且按数据进行接收数据的移位。输入需要的 RSPCK，如果收集到 R23 ~ P 的数据，就将移位寄存器的值复制到接收缓冲器。如果将数据复制到移位寄存器，就检查 R23 ~ P 的数据，判断奇偶校验错误。此时，将发送缓冲器的高 8 位保存到接收缓冲器的高 8 位。能在发送时预先给 T31 ~ T24 写“0”，使接收缓冲器的高 8 位变为“0”。

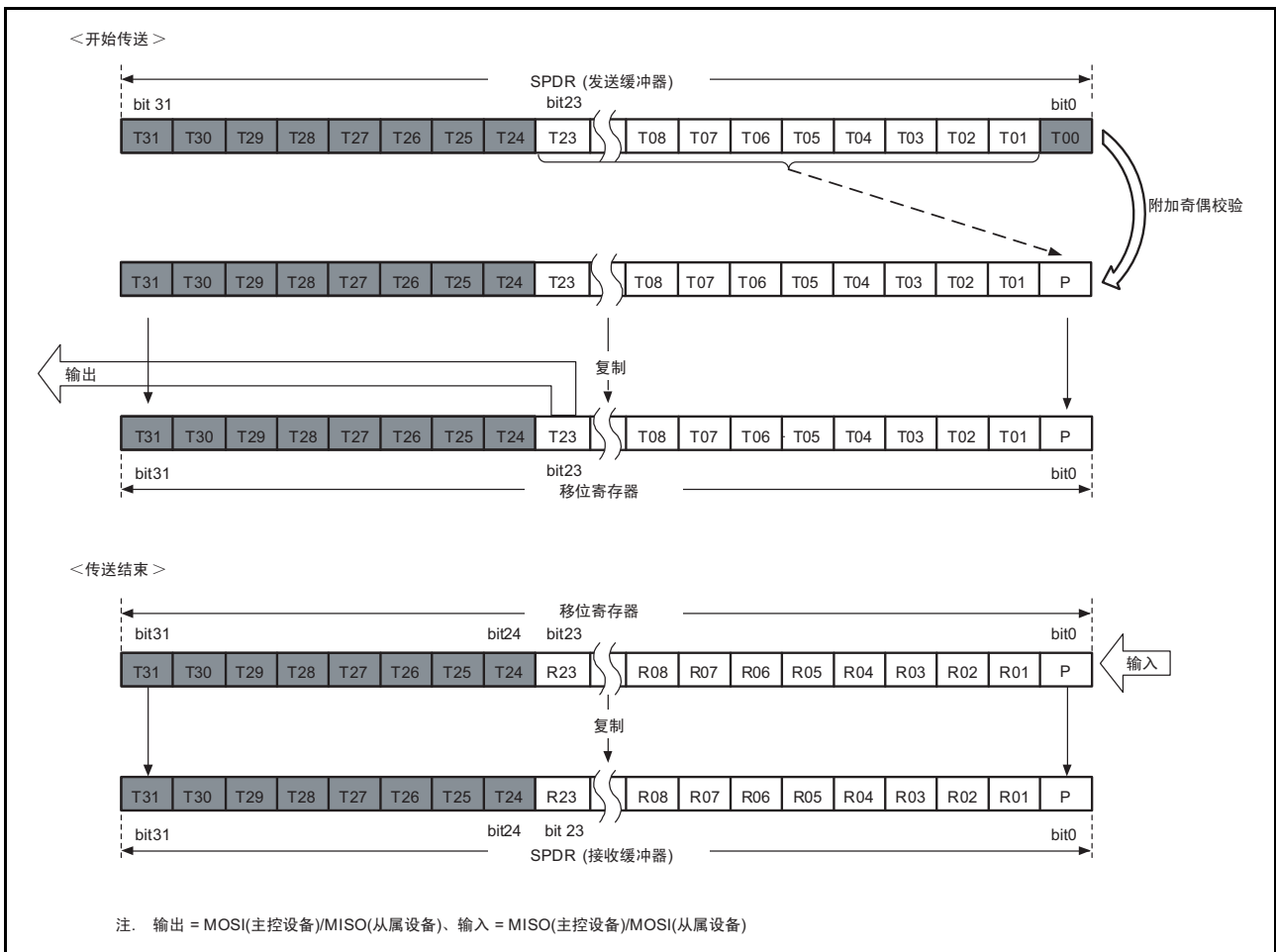


图 29.20 MSB first 传送 (24 位数据 / 奇偶校验功能有效)

(3) LSB first 传送 (32 位数据)

在奇偶校验功能有效的情况下, RSPI 进行 32 位数据长度的 LSB first 传送时的 RSPI 数据寄存器 (SPDR) 和移位寄存器的运行内容如图 29.21 所示。

在发送时, 先根据 T30 ~ T00 的数据进行奇偶校验位 (P) 的运算, 替换最后位的 T31, 然后复制到移位寄存器。按 T00→T01→…T30→P 的顺序发送数据。

在接收时, 将接收数据保存到移位寄存器的 bit0, 并且按数据进行接收数据的移位。输入需要的 RSPCK, 如果收集到 R00 ~ P 的数据, 就将移位寄存器的值复制到接收缓冲器。如果将数据复制到移位寄存器, 就检查 R00 ~ P 的数据, 判断奇偶校验错误。

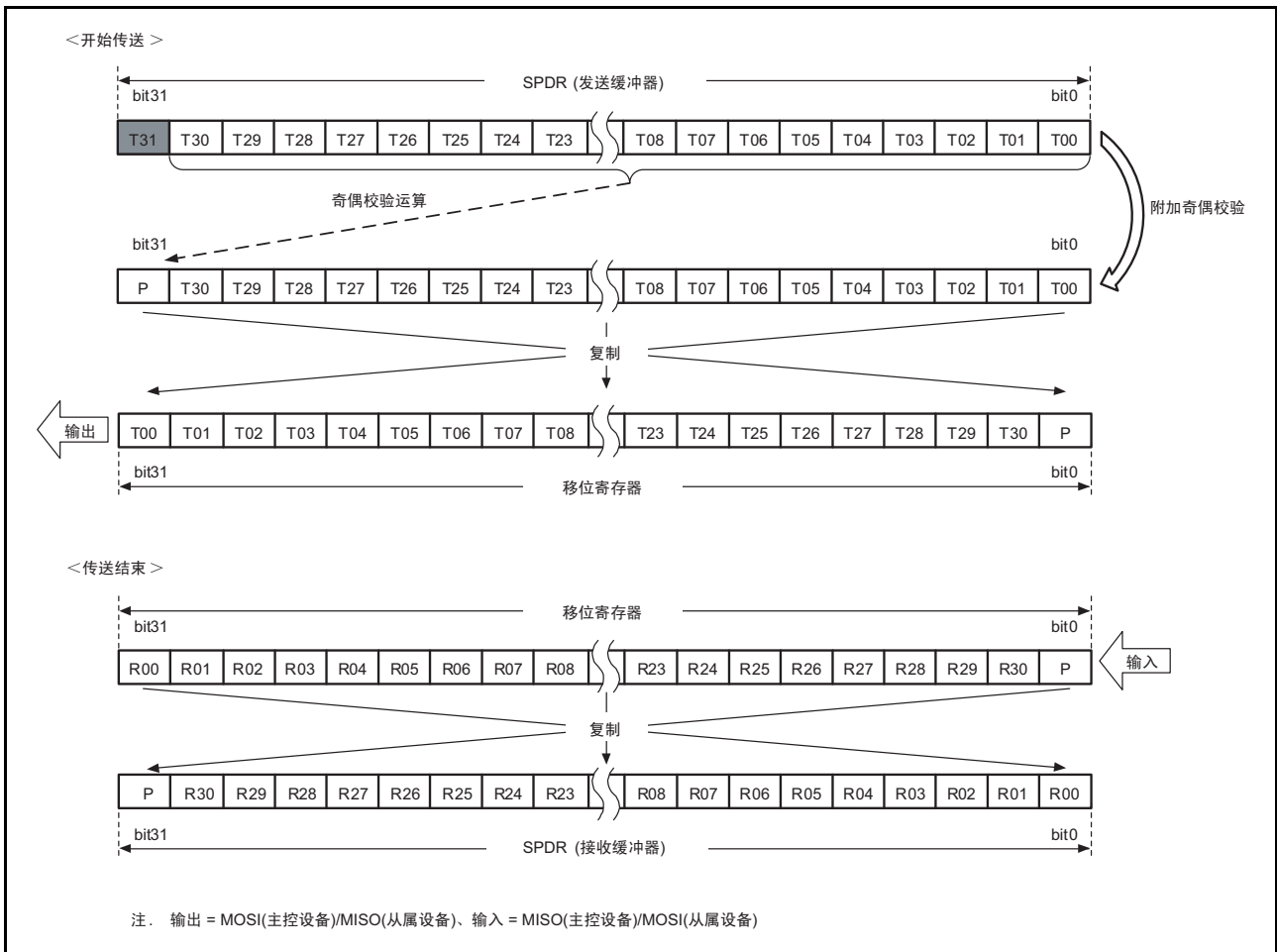


图 29.21 LSB first 传送 (32 位数据 / 奇偶校验功能有效)

(4) LSB first 传送 (24 位数据)

在奇偶校验功能有效的情况下, 以对 32 位以外的数据进行 LSB first 传送为例, RSPI 进行 24 位数据传送时的 RSPI 数据寄存器 (SPDR) 和移位寄存器的运行内容如图 29.22 所示。

在发送时, 先根据 T22 ~ T00 的数据进行奇偶校验位 (P) 的运算, 替换最后位的 T23, 然后复制到移位寄存器。按 T00→T01→…T22→P 的顺序发送数据。

在接收时, 将接收数据保存到移位寄存器的 bit8, 并且按数据进行接收数据的移位。输入需要的 RSPCK, 如果收集到 R00 ~ P 的数据, 就将移位寄存器的值复制到接收缓冲器。如果将数据复制到移位寄存器, 就检查 R00 ~ P 的数据, 判断奇偶校验错误。此时, 将发送缓冲器的高 8 位保存到接收缓冲器的高 8 位。能在发送时预先给 T31 ~ T24 写“0”, 使接收缓冲器的高 8 位变为“0”。

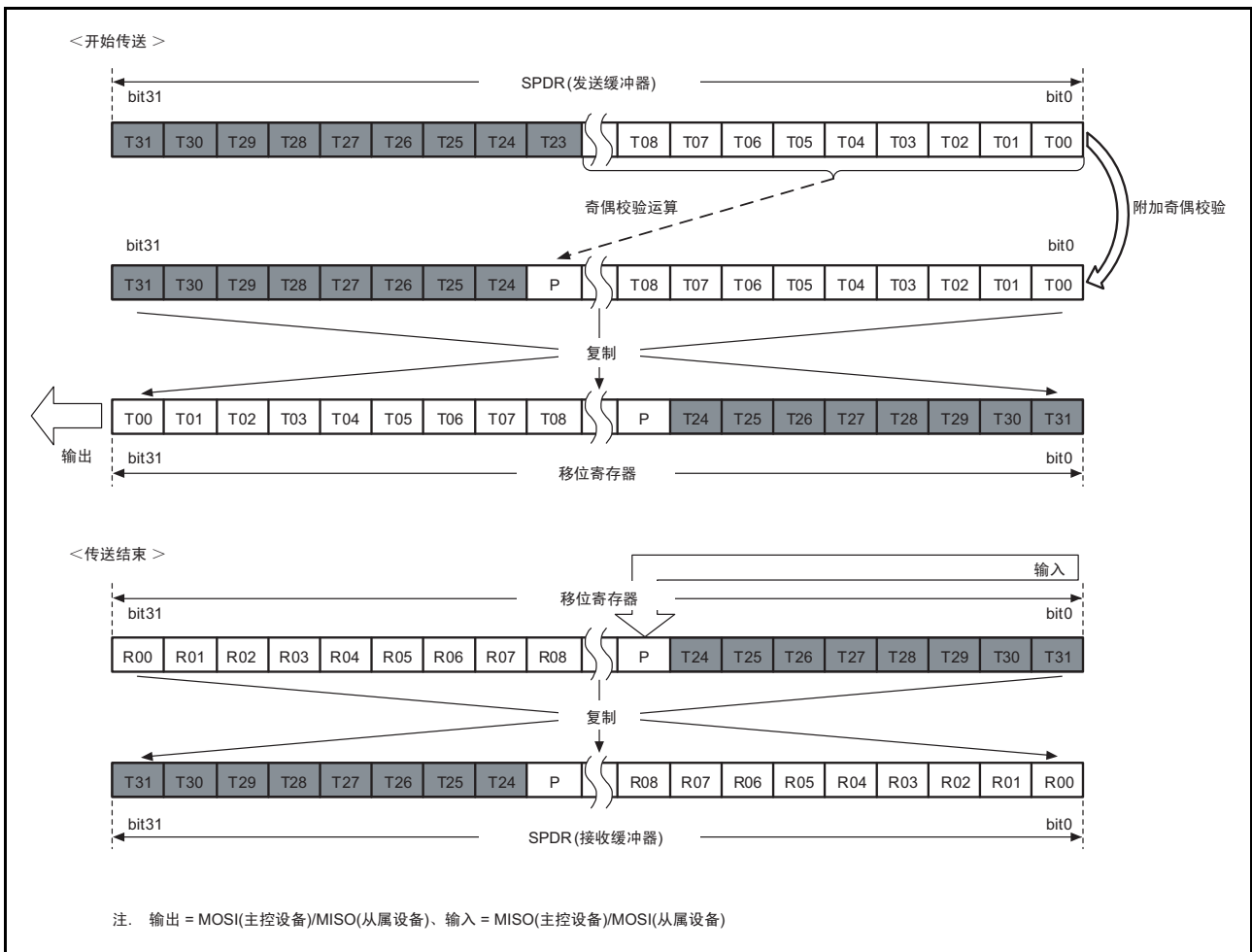


图 29.22 LSB first 传送 (24 位数据 / 奇偶校验功能有效)

29.3.5 传送格式

29.3.5.1 CPHA 位为“0”的情况

在 SPCMDm.CPHA 位为“0”的情况下，对 8 位数据进行串行传送时的传送格式例子如图 29.23 所示。但是，不保证 RSPI 为从属模式 (SPCR.MSTR=0) 并且 CPHA 位为“0”时的时钟同步运行 (SPCR.SPMS 位为“1”)。在图 29.23 中，RSPCKA (CPOL=0) 是 SPCMDm.CPOL 位为“0”时的 RSPCKA 信号波形，RSPCKA (CPOL=1) 是 SPCMDm.CPOL 位为“1”时的 RSPCKA 信号波形。采样时序表示 RSPI 将串行传送数据取到移位寄存器的时序。各信号的输入 / 输出方向取决于 RSPI 的设置，详细内容请参照“29.3.2 RSPI 引脚的控制”。

当 SPCMDm.CPHA 位为“0”时，在 SSLAi 信号的有效时序开始驱动 MOSIA 信号和 MISOA 信号的有效数据。在 SSLAi 信号有效后发生的第一个 RSPCKA 信号变化时序为第一个传送数据的取时序，在此时序之后，按每个 RSPCK 周期对数据进行采样。MOSIA 信号和 MISOA 信号的变化时序总是在传送数据取时序的 1/2 个 RSPCK 周期之后。CPOL 位的设定值不影响 RSPCK 信号的运行时序而只影响信号的极性。

t1 表示从 SSLAi 信号有效到 RSPCKA 振荡的期间 (RSPCK 延迟)，t2 表示从 RSPCKA 振荡停止到 SSLAi 信号无效的期间 (SSL 无效延迟)，t3 表示在串行传送结束后抑制下次传送的 SSLAi 信号有效的期间 (下次存取延迟)。t1、t2 和 t3 由 RSPI 系统上的主控设备进行控制。有关本 LSI 的 RSPI 为主控模式时的 t1、t2 和 t3，请参照“29.3.10.1 主控模式的运行”。

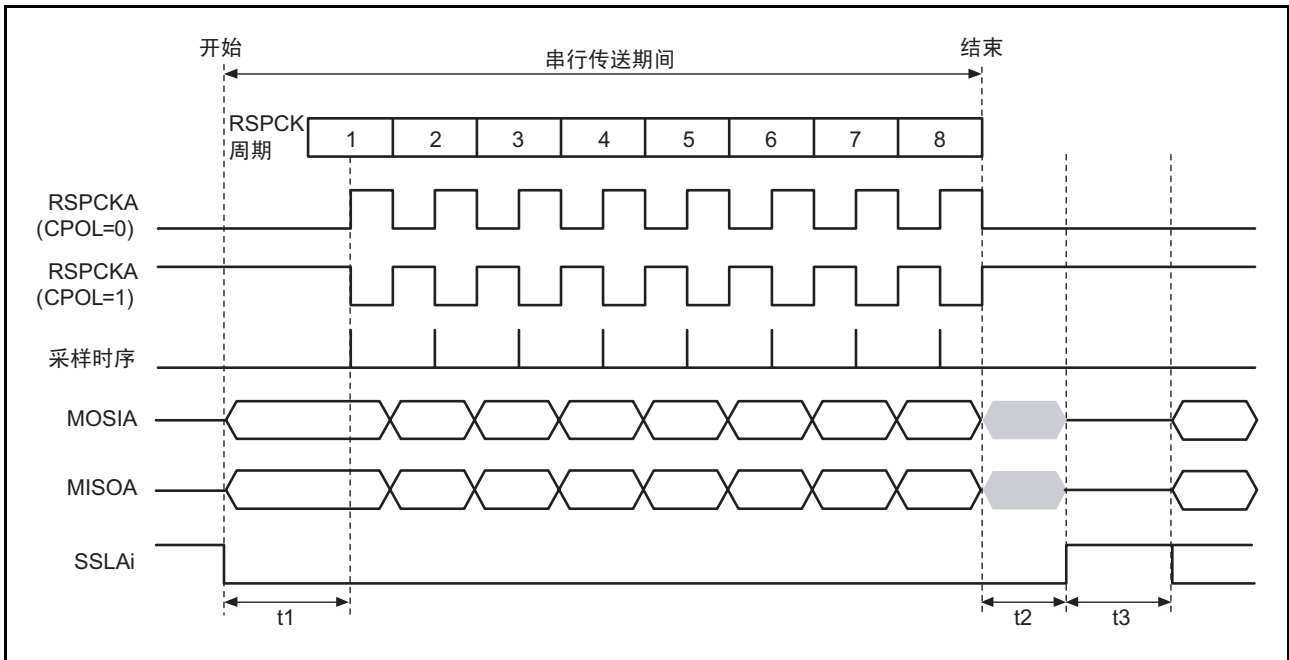


图 29.23 RSPI 传送格式 (CPHA 位 =0)

29.3.5.2 CPHA 位为“1”的情况

在 SPCMDm.CPHA 位为“1”的情况下，对 8 位数据进行串行传送时的传送格式例子如图 29.24 所示。但是，当 SPCR.SPMS 位为“1”时，不使用 SSLAi 信号而只使用 RSPCKA 信号、MOSIA 信号和 MISOA 信号进行通信。在图 29.24 中，RSPCKA (CPOL=0) 是 SPCMDm.CPOL 位为“0”时的 RSPCKA 信号波形，RSPCK (CPOL=1) 是 SPCMDm.CPOL 位为“1”的 RSPCKA 信号波形。采样时序表示 RSPI 将串行传送数据取到移位寄存器的时序。各信号的输入/输出方向取决于 RSPI 的模式（主控/从属模式），详细内容请参照“29.3.2 RSPI 引脚的控制”。

当 SPCMDm.CPHA 位为“1”时，在 SSLAi 信号的有效时序开始驱动 MISOA 信号的有效数据。在 SSLAi 信号有效后发生的第一个 RSPCKA 信号变化时序，开始输出 MOSIA 信号和 MISOA 信号的有效数据。在此时序后，按每个 RSPCK 周期更新数据。传送数据的取时序总是在此时序的 1/2 个 RSPCK 周期之后。SPCMDm.CPOL 位的设定值不影响 RSPCKA 信号的运行时序而只影响信号的极性。

t1、t2、t3 的内容和 CPHA 位为“0”的情况相同。有关本 LSI 的 RSPI 为主控模式时的 t1、t2 和 t3，请参照“29.3.10.1 主控模式的运行”。

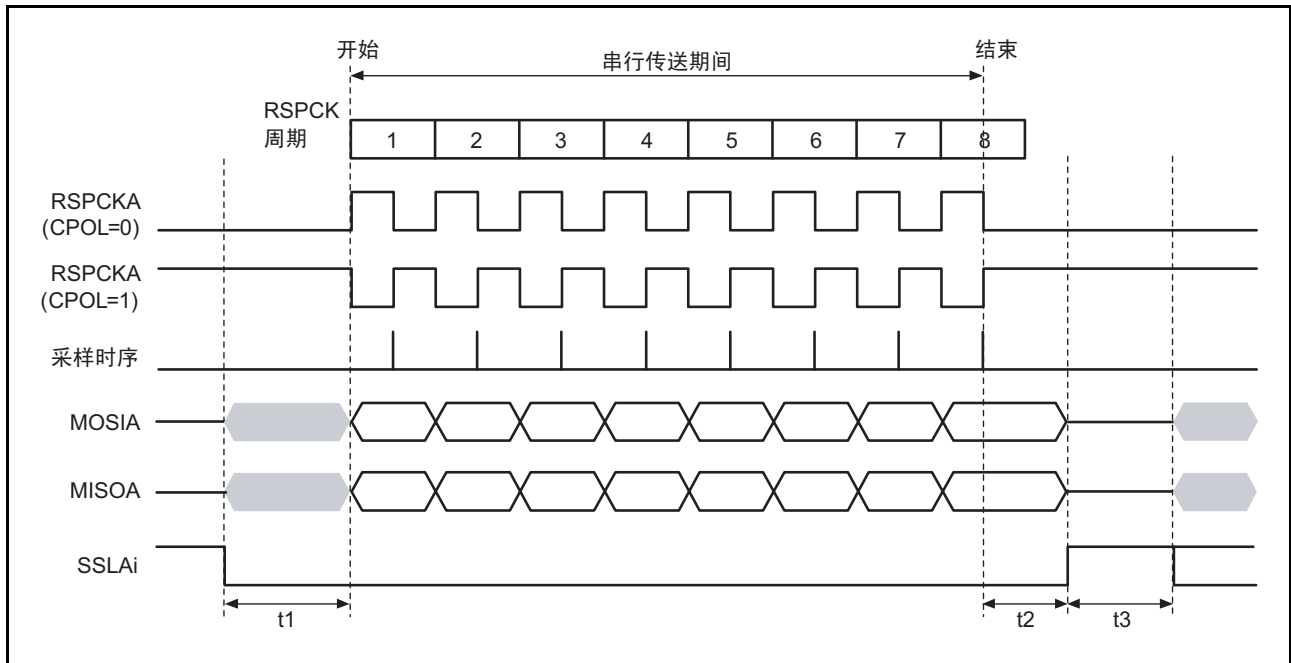


图 29.24 RSPI 传送格式 (CPHA 位 =1)

29.3.6 通信运行模式

通过设定 SPCR.TXMD 位，选择是进行全双工同步串行通信还是只进行发送的串行通信。图 29.25 和图 29.26 记载的“SPDR 存取”表示 SPDR 寄存器的存取状况，“W”表示写周期。

29.3.6.1 全双工同步串行通信 (SPCR.TXMD=0)

SPCR.TXMD 位为“0”时的运行例子如图 29.25 所示。在图 29.25 的例子中，SPDCR.SPFC[1:0] 位为“00b”，SPCMDm.CPHA 位为“1”并且 SPCMDm.CPOL 位为“0”，RSPI 进行 8 位串行传送。RSPCKA 波形下面记载的数字表示 RSPCK 周期数 (= 传送位数)。

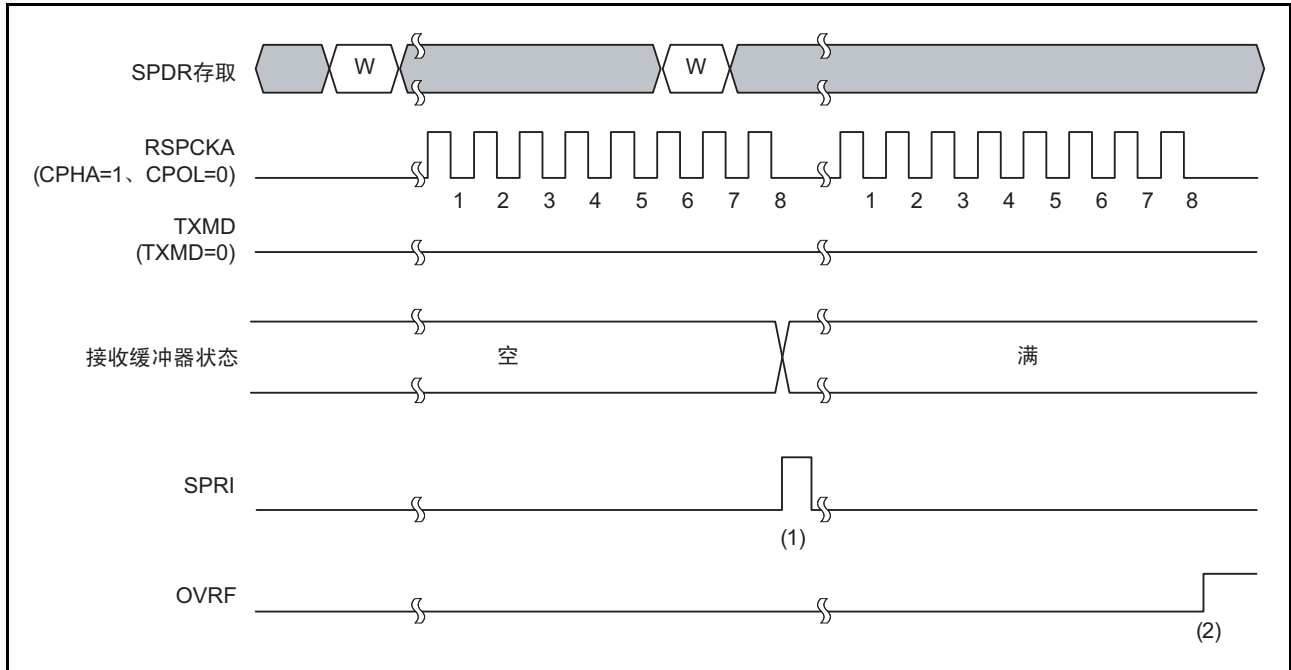


图 29.25 SPCR.TXMD 为“0”时的运行例子

以下说明在图中的 (1) 和 (2) 所示的时序中标志的运行内容：

- (1) 如果在 SPDR 寄存器的接收缓冲器为空的状态下结束串行传送，RSPI 就在产生接收缓冲器满中断请求 (SPRI) 后，将移位寄存器的接收数据复制到接收缓冲器。
- (2) 如果在 SPDR 寄存器的接收缓冲器有以前的接收数据的状态下结束串行传送，RSPI 就在将 SPSR.OVRF 标志置“1”后，放弃移位寄存器的接收数据。

29.3.6.2 只进行发送的运行 (SPCR.TXMD=1)

SPCR.TXMD 位为“1”时的运行例子如图 29.26 所示。在图 29.26 的例子中，SPDCR.SPFC[1:0] 位为“00b”，SPCMDm.CPHA 位为“1”并且 SPCMDm.CPOL 位为“0”，RSPI 进行 8 位串行传送。RSPCKA 波形下面记载的数字表示 RSPCK 周期数 (= 传送位数)。

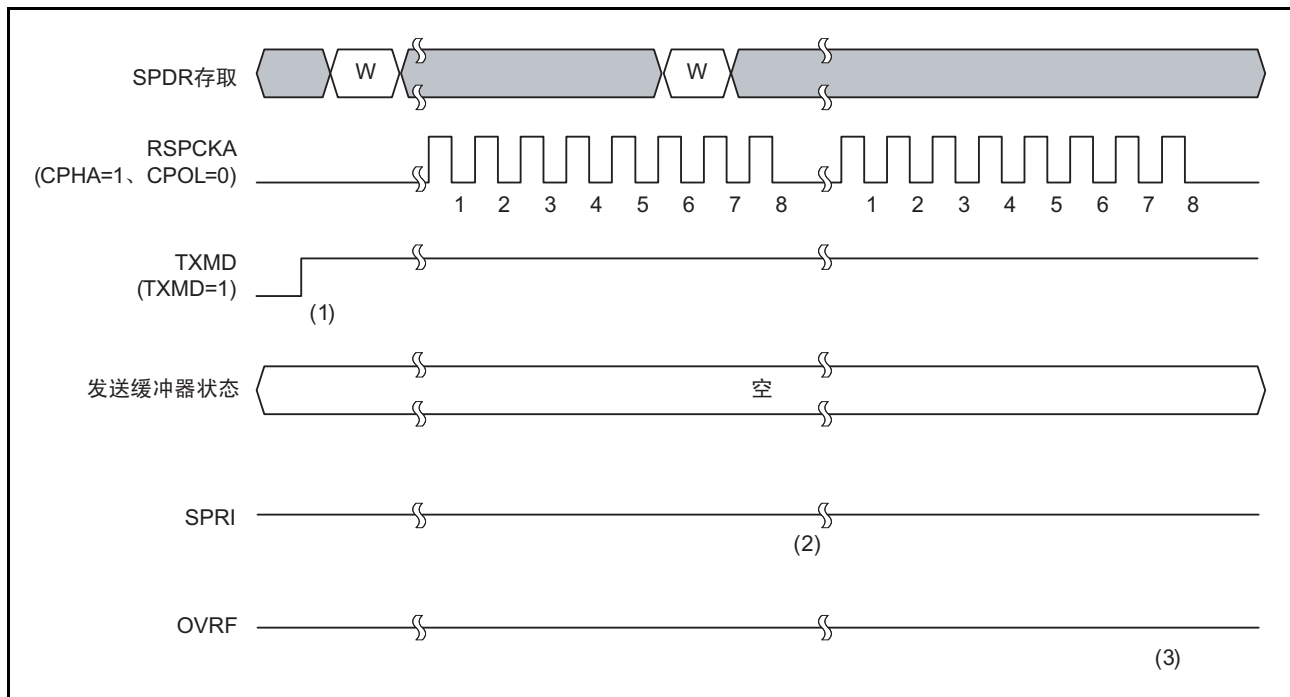


图 29.26 SPCR.TXMD 为“1”时的运行例子

以下说明在图中的 (1)、(2)、(3) 所示的时序中标志的运行内容:

- (1) 必须在确认接收缓冲器没有数据并且 SPSR.OVRF 标志是“0”后转移到只进行发送的运行 (SPCR.TXMD=1)。
- (2) 如果在 SPDR 寄存器的接收缓冲器为空的状态下结束串行传送,就在只进行发送的运行 (SPCR.TXMD=1) 时不将移位寄存器的数据复制到接收缓冲器。
- (3) 因为 SPDR 寄存器的接收缓冲器没有以前的接收数据,所以即使结束串行传送 SPSR.OVRF 标志也保持“0”,并且不将移位寄存器的数据复制到接收缓冲器。

在只进行发送的运行 (SPCR.TXMD=1), 发送数据而不接收数据。因此,在 (1)、(2)、(3) 各自的时序中, SPSR.OVRF 标志保持“0”。

29.3.7 发送缓冲器空中断 / 接收缓冲器满中断

RSPI 发送缓冲器空中断 (SPTI) 和 RSPI 接收缓冲器满中断 (SPRI) 的运行例子如图 29.27 所示。图 29.27 中记载的“SPDR 存取”表示 SPDR 寄存器的存取状况,“W”表示写周期,“R”表示读周期。在图 29.27 的例子中,SPCR.TXMD 位为“0”,SPDCR.SPFC[1:0] 位为“00b”,SPCMDm.CPHA 位为“1”并且 SPCMDm.CPOL 为“0”,RSPI 进行 8 位串行传送。RSPCKA 波形下面记载的数字表示 RSPCK 周期数 (= 传送位数)。

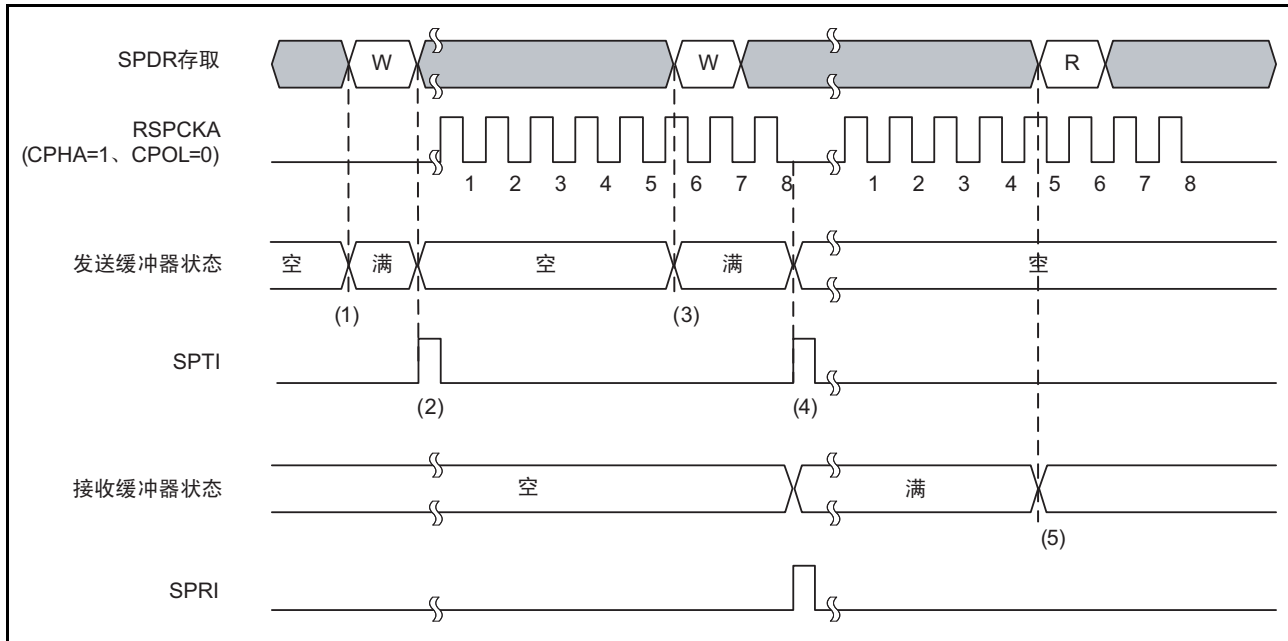


图 29.27 SPTI 中断和 SPRI 中断的运行例子

以下说明在图中的 (1) ~ (5) 所示的时序中中断的运行内容:

- (1) 在 SPDR 寄存器的发送缓冲器为空的 (未设定下次传送的数据) 状态下, 如果将发送数据写到 SPDR 寄存器, RSPI 就将数据写到发送缓冲器。
- (2) 如果移位寄存器为空, 就在 RSPI 将发送缓冲器的数据复制到移位寄存器后产生发送缓冲器空中断 (SPTI)。串行传送的开始方法取决于 RSPI 的模式 (参照“29.3.10 SPI 运行”和“(6) 软件处理流程”)。
- (3) 如果通过发送缓冲器空中断程序将发送数据写到 SPDR 寄存器, 就将数据传送到发送缓冲器。因为串行传送过程中的数据被保存在移位寄存器, 所以 RSPI 不将发送缓冲器的数据复制到移位寄存器。
- (4) 如果在 SPDR 寄存器的接收缓冲器为空的状态下结束串行传送, RSPI 就将移位寄存器的接收数据复制到接收缓冲器并且产生接收缓冲器满中断请求 (SPRI)。一旦串行传送结束, 移位寄存器就变空。因此, 如果在串行传送结束前发送缓冲器满, RSPI 就将发送缓冲器的数据复制到移位寄存器。即使在发生溢出错误的状态下不将接收数据从移位寄存器复制到接收缓冲器, 只要串行传送结束, RSPI 也判断为移位寄存器处于空的状态, 变为可将数据从发送缓冲器传送到移位寄存器的状态。
- (5) 如果通过接收缓冲器满中断程序读 SPDR 寄存器, 就能读到接收数据。

如果在发送缓冲器有未发送数据的状态下写 SPDR 寄存器, RSPI 就不更新发送缓冲器的数据。必须通过发送缓冲器空中断程序来写 SPDR 寄存器。在使用 RSPI 发送中断时, 必须将 SPCR 的 SPTIE 位置“1”。

在 RSPI 功能无效 (SPCR 的 SPE 位为“0”) 状态下, 必须将 SPTIE 位置“0”。

如果在接收缓冲器满的状态下结束串行传送, RSPI 就不将数据从移位寄存器复制到接收缓冲器, 而检测到溢出错误 (参照“29.3.8 错误检测”)。为了防止接收数据的溢出, 必须在下次串行传送结束前, 通过接收缓冲器满中断请求读接收数据。在使用 RSPI 接收中断时, 必须将 SPCR.SPRIE 位置“1”。

能通过发送 / 接收中断或者对应 ICU 的 IRn.IR 标志 (n= 中断向量号) 确认发送 / 接收缓冲器的状态。有关中断向量号, 请参照“14. 中断控制器 (ICUb)”。

29.3.8 错误检测

在正常的 RSPI 串行传送中，能对写在 SPDR 寄存器的发送缓冲器中的数据进行串行发送，并且能从 SPDR 寄存器的接收缓冲器读串行接收的数据。根据存取 SPDR 寄存器时的发送 / 接收缓冲器的状态以及串行传送开始 / 结束时的 RSPI 的状态，有可能发生异常传送。

如果发生部分异常传送，RSPI 就检测为溢出错误、奇偶校验错误或者模式故障错误。异常传送和 RSPI 错误检测功能的关系如表 29.8 所示。

表 29.8 异常传送的产生条件和 RSPI 的错误检测功能

	发生条件	RSPI 运行	错误检测
A	在发送缓冲器满的状态下写 SPDR 寄存器。	<ul style="list-style-type: none"> 保持发送缓冲器内容。 写数据丢失。 	无
B	在从属模式中未将发送数据设定到移位寄存器的状态下开始串行传送。	对上次串行传送时的接收数据进行串行发送。	无
C	在接收缓冲器为空的状态下读 SPDR 寄存器。	输出上次的串行接收数据。	无
D	在接收缓冲器满的状态下结束串行传送。	保持接收缓冲器内容并且串行接收数据丢失。	溢出错误检测
E	在进行全双工同步串行通信并且奇偶校验功能有效的状态下接收到错误的奇偶校验位。	奇偶校验错误标志有效。	奇偶校验错误检测
F	在多主控模式的串行传送空闲时 SSLA0 输入信号有效。	<ul style="list-style-type: none"> 停止驱动 RSPCKA、MOSIA、SSLA1 ~ 3 输出信号。 RSPI 功能为无效。 	模式故障错误检测
G	在多主控模式的串行传送过程中 SSLA0 输入信号有效。	<ul style="list-style-type: none"> 中止串行传送。 发送 / 接收数据丢失。 停止驱动 RSPCKA、MOSIA、SSLA1 ~ 3 输出信号。 RSPI 功能为无效。 	模式故障错误检测
H	在从属模式的串行传送过程中 SSLA0 输入信号无效。	<ul style="list-style-type: none"> 中止串行传送。 发送 / 接收数据丢失。 停止驱动 MISOA 输出信号。 RSPI 功能为无效。 	模式故障错误检测

对于表 29.8 的 A 所示的运行，RSPI 不检测错误。为了在写 SPDR 寄存器时不丢失数据，必须通过发送中断请求来写 SPDR 寄存器。

即使对于 B 所示的运行，RSPI 也不检测错误。在更新移位寄存器前启动的串行传送中，RSPI 发送上次串行传送时的接收数据，而不将 B 所示的运行作为错误进行处理。因为上次串行传送时的接收数据保存在 SPDR 寄存器的接收缓冲器，所以能正常读取（如果不在串行传送结束前读 SPDR 寄存器，就发生溢出错误）。

即使对于 C 所示的运行，RSPI 也不检测错误。为了只读需要的数据，必须通过接收中断来读 SPDR 寄存器。

有关 D 所示的溢出错误，请参照“29.3.8.1 溢出错误”；有关 E 所示的奇偶校验错误，请参照“29.3.8.2 奇偶校验错误”；有关 F ~ H 所示的模式故障错误，请参照“29.3.8.3 模式故障错误”。

有关发送中断和接收中断，请参照“29.3.7 发送缓冲器空中断 / 接收缓冲器满中断”。

29.3.8.1 溢出错误

如果在 SPDR 寄存器的接收缓冲器满的状态下结束串行传送，RSPI 就检测到溢出错误，然后将 SPSR.OVRF 标志置“1”。因为在 OVRF 标志为“1”的状态下 RSPI 不将移位寄存器的数据复制到接收缓冲器，所以发生错误前的数据保存在接收缓冲器。要将 OVRF 标志置“0”时，需要在 OVRF 标志为“1”的状态下读 SPSR 寄存器后给 OVRF 标志写“0”。

SPSR.OVRF 标志的运行如图 29.28 所示。图 29.28 中记载的 SPSR 存取和 SPDR 存取分别表示 SPSR 寄存器和 SPDR 寄存器的存取状况，“W”表示写周期，“R”表示读周期。在图 29.28 的例子中，SPCMDm.CPHA 位为“1”并且 SPCMDm.CPOL 位为“0”，RSPI 进行 8 位串行传送。RSPCKA 波形下面记载的数字表示 RSPCK 周期数 (= 传送位数)。

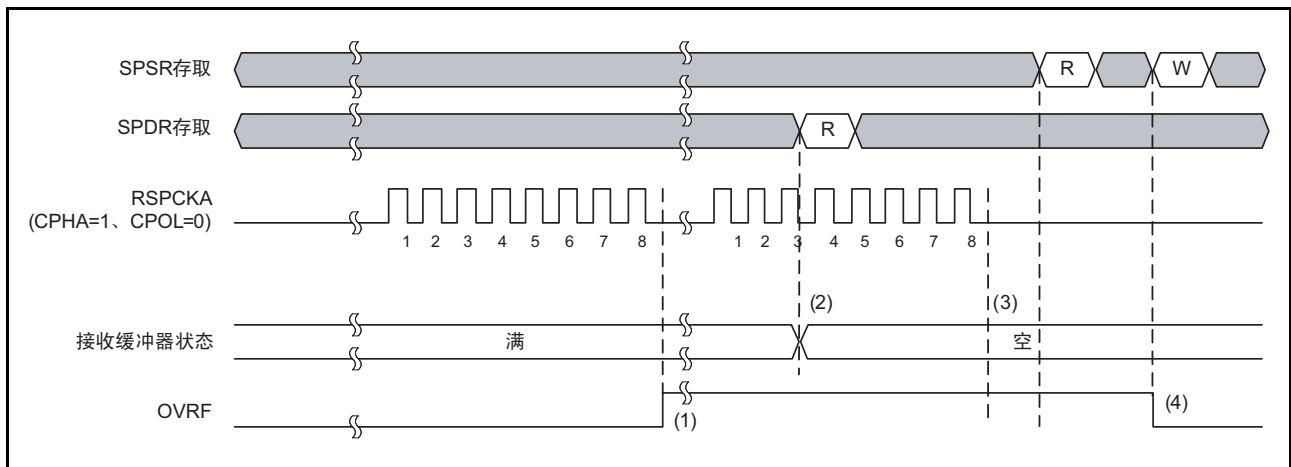


图 29.28 OVRF 标志的运行例子

以下说明在图中的 (1) ~ (4) 所示的时序中标志的运行内容：

- (1) 如果在接收缓冲器满的状态下结束串行传送，RSPI 就检测到溢出错误并且将 OVRF 标志置“1”。RSPI 不将移位寄存器的数据复制到接收缓冲器。即使 SPPE 位为“1”时，也不检测奇偶校验错误。在主控模式中，将指向 SPCMDm 寄存器的指针值复制到 SPSR.SPECM[2:0] 位。
- (2) 如果读 SPDR，RSPI 就能读到接收缓冲器的数据。即使接收缓冲器为空，OVRF 标志也不变为“0”。
- (3) 如果在 OVRF 标志为“1”的状态（溢出错误）下结束串行传送，RSPI 就不将移位寄存器的数据复制到接收缓冲器，也不产生接收缓冲器满中断。即使 SPPE 位为“1”，也不检测奇偶校验错误。在主控模式中，RSPI 不更新 SPECM[2:0] 位。在发生溢出错误的状态下，如果在没有将接收数据从移位寄存器复制到接收缓冲器的情况下结束串行传送，RSPI 就判断为移位寄存器处于空的状态，变为可将数据从发送缓冲器传送到移位寄存器的状态。
- (4) 如果在 OVRF 标志为“1”的状态下读 SPSR 寄存器后给 OVRF 标志写“0”，RSPI 就将 OVRF 标志置“0”。

能通过读 SPSR 寄存器或者通过 RSPI 错误中断并且读 SPSR 寄存器，确认溢出的发生。在进行串行传送时，必须通过在读 SPDR 寄存器后立即读 SPSR 寄存器等方法，尽早检测到溢出错误的发生。在主控模式中使用 RSPI 时，能通过读 SPSR.SPECM[2:0] 位，确认发生错误时的指向 SPCMDm 寄存器的指针值。

如果在发生溢出错误后 OVRF 标志变为“1”，就在 OVRF 标志变为“0”前无法进行正常的接收运行。

29.3.8.2 奇偶校验错误

在 SPCR.TXMD 位为“0”并且 SPCR2.SPPE 位为“1”的状态下，如果进行全双工同步串行通信并且结束传送，就判断为奇偶校验错误。当 RSPI 检测到接收数据有奇偶校验错误时，将 SPSR.PERF 标志置“1”。在 SPSR.OVRF 位为“1”的状态下，因为 RSPI 不将移位寄存器的数据复制到接收缓冲器，所以不对接收数据进行奇偶校验错误的检测。要将 PERF 标志置“0”时，需要在 PERF 标志为“1”的状态下读 SPSR 寄存器后给 PERF 标志写“0”。

OVRF 标志和 PERF 标志的运行如图 29.29 所示。图 29.29 中记载的“SPSR 存取”表示 SPSR 寄存器的存取状况，“W”表示写周期，“R”表示读周期。在图 29.29 的例子中，在 SPCR.TXMD 位为“0”并且 SPCR2.SPPE 位为“1”的状态下进行全双工同步串行通信。通过 SPCMDm.CPHA 位为“1”并且 SPCMDm.CPOL 位为“0”的设定，RSPI 进行 8 位串行传送。RSPCKA 波形下面记载的数字表示 RSPCK 周期数 (= 传送位数)。

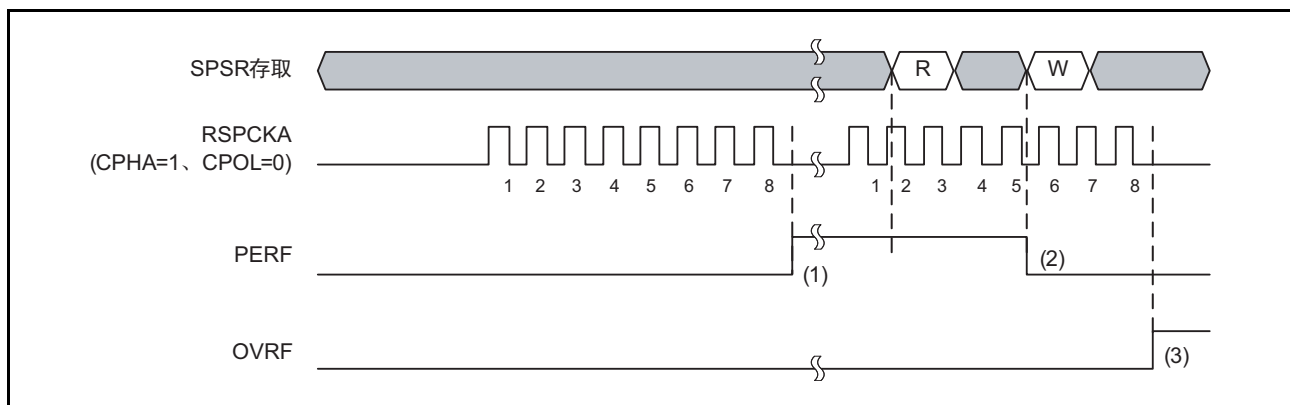


图 29.29 PERF 标志的运行例子

以下说明在图中 (1) ~ (3) 所示的时序中标志的运行内容：

- (1) 如果 RSPI 没有检测到溢出错误而结束串行传送，就将移位寄存器的数据复制到接收缓冲器。此时，RSPI 判断接收数据，如果检测到奇偶校验错误，就将 PERF 标志置“1”。在主机模式中，将指向 SPCMDm 寄存器的指针值复制到 SPSR.SPECM[2:0] 位。
- (2) 如果在 PERF 标志为“1”的状态下读 SPSR 寄存器后给 PERF 标志写“0”，PERF 标志就变为“0”。
- (3) 如果 RSPI 检测到溢出错误并且结束串行传送，就将移位寄存器的数据复制到接收缓冲器。此时，RSPI 不检测奇偶校验错误。

能通过读 SPSR 寄存器或者通过 RSPI 错误中断并且读 SPSR 寄存器，确认奇偶校验错误的发生。在进行串行传送时，必须通过读 SPSR 等方法，尽早检测到奇偶校验错误的发生。在主机模式中使用 RSPI 时，能通过读 SPSR.SPECM[2:0] 位，确认发生错误时的指向 SPCMDm 寄存器的指针值。

29.3.8.3 模式故障错误

当 SPCR.MSTR 位为“1”、SPCR.SPMS 位为“0”并且 SPCR.MODFEN 位为“1”时，RSPI 以多主控模式运行。如果给多主控模式的 RSPI 的 SSLA0 输入信号输入有效电平，RSPI 就检测到模式故障错误，然后将 SPSR.MODF 标志置“1”，与串行传送状态无关。如果检测到模式故障错误，RSPI 就将指向 SPCMDm 寄存器的指针值复制到 SPSSR.SPECM[2:0] 位。SSLA0 信号的有效电平取决于 SSLP.SSL0P。

当 MSTR 位为“0”时，RSPI 以从属模式运行。当从属模式的 RSPI 的 MODFEN 位为“1”并且 SPMS 位为“0”时，如果在串行传送期间（从开始驱动有效数据到取最后的有效数据前）使 SSLA0 输入信号无效，RSPI 就检测到模式故障错误。

如果 RSPI 检测到模式故障错误，就停止驱动输出信号并且清除 SPCR.SPE 位（参照“29.3.9 RSPI 的初始化”）。在多主控结构中，如果使用模式故障错误来驱动输出信号，就能停止 RSPI 功能并且释放主控权。

能通过读 SPSR 寄存器或者通过 RSPI 错误中断并且读 SPSR 寄存器，确认模式错误的发生。为了在不使用 RSPI 错误中断的情况下检测模式故障错误，需要轮询 SPSR 寄存器。在主导模式中使用 RSPI 时，能通过读 SPSSR.SPECM[2:0] 位，确认发生错误时的指向 SPCMDm 寄存器的指针值。

在 MODF 标志为“1”的状态下，RSPI 忽视给 SPE 位的写“1”。为了使 RSPI 功能在检测到模式故障错误后有效，必须将 MODF 标志置“0”。

29.3.9 RSPI 的初始化

如果给 SPCR.SPE 位写“0”或者 RSPI 通过模式故障错误检测将 SPE 位置“0”，RSPI 就使 RSPI 功能无效并且对部分模块功能进行初始化。如果发生系统复位，RSPI 就对全部模块功能进行初始化。以下说明通过将 SPCR.SPE 位置“0”进行的初始化以及通过系统复位进行的初始化。

29.3.9.1 通过清除 SPE 位进行的初始化

当 SPCR.SPE 位为“0”时，RSPI 进行以下所示的初始化：

- 中止正在进行的串行传送。
- 在从属模式中停止驱动输出信号（Hi-Z）。
- 对 RSPI 内部状态进行初始化。
- 将 RSPI 发送缓冲器清空。

在将 SPE 位置“0”进行初始化时，不对 RSPI 的控制位进行初始化。因此，只要重新将 SPE 位置“1”，就能在和初始化前相同的传送模式中启动 RSPI。

不对 SPSR.OVRF 和 SPSR.MODF 标志的值进行初始化，也不对 SPSSR 寄存器的值进行初始化。因此，在 RSPI 的初始化后，也能通过读接收缓冲器的数据，确认 RSPI 传送时的错误发生状况。

将发送缓冲器初始化为空状态。因此，如果在 RSPI 的初始化后将 SPCR.SPTIE 位置“1”，就产生 RSPI 发送中断。在通过 CPU 对 RSPI 进行初始化的情况下，为了禁止 RSPI 发送中断，必须在给 SPE 位写“0”的同时给 SPTIE 位写“0”。要在检测到模式故障错误后禁止 RSPI 发送中断时，必须通过错误处理程序给 SPTIE 位写“0”。

29.3.9.2 系统复位

在通过系统复位进行初始化时，除了“29.3.9.1 通过清除 SPE 位进行的初始化”记载的事项以外，还对用于 RSPI 控制的全部位、状态位和数据寄存器进行初始化，RSPI 完全被初始化。

29.3.10 SPI 运行

29.3.10.1 主控模式的运行

单主控模式运行和多主控模式运行的不同点在于模式故障错误检测（参照“29.3.8 错误检测”）。单主控模式的 RSPI 不检测模式故障错误，多主控模式的 RSPI 检测模式故障错误。本节说明单主控模式 / 多主控模式的通用运行。

(1) 串行传送的开始

在 RSPI 发送缓冲器为空（未设定下次传送的数据）的状态下，如果将数据写到 SPDR 寄存器，RSPI 就更新 SPDR 寄存器的发送缓冲器（SPTX）的数据。在将 SPDCR.SPFC[1:0] 位所设帧的数据写到 SPDR 寄存器后移位寄存器变空时，RSPI 在将发送缓冲器的数据复制到移位寄存器后开始串行传送。如果 RSPI 将发送数据复制到移位寄存器，就将移位寄存器状态改为满状态；如果串行传送结束，就将移位寄存器改为空状态。不能参照移位寄存器的状态。

有关 RSPI 传送格式的详细内容，请参照“29.3.5 传送格式”。SSLAi 输出引脚的极性取决于 SSLP 寄存器的设定值。

(2) 串行传送的结束

与 SPCMDm.CPHA 位无关，如果 RSPI 发送最后采样时序所对应的 RSPCKA 边沿，就结束串行传送。在接收缓冲器（SPRX）为空的情况下，在串行传送结束后将数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器。

最后的采样时序取决于传送数据的位长，主控模式的 RSPI 数据长度取决于 SPCMDm.SPB[3:0] 位的设定值，SSLAi 输出引脚的极性取决于 SSLP 寄存器的设定值。有关 RSPI 传送格式的详细内容，请参照“29.3.5 传送格式”。

(3) 顺序控制

主控模式时的传送格式取决于 SPSCR 寄存器、SPCMDm 寄存器、SPBR 寄存器、SPCKD 寄存器、SSLND 寄存器和 SPND 寄存器。

SPSCR 寄存器决定主控模式的 RSPI 要执行的串行传送的顺序结构。给 SPCMDm 寄存器设定 SSLAi 引脚的输出信号值、MSB/LSB first、数据长度、部分位速率、RSPCK 极性 / 相位以及设定是否要参照 SPCKD 寄存器、SSLND 寄存器和 SPND 寄存器；给 SPBR 寄存器设定部分位速率，给 SPCKD 设定 RSPI 时钟延迟值，给 SSLND 寄存器设定 SSL 无效延迟并且给 SPND 寄存器设定 RSPI 的下次存取延迟值。

RSPI 根据 SPSCR 寄存器设定的顺序长度，由部分或者全部 SPCMDm 寄存器构成顺序。RSPI 有构成顺序的 SPCMDm 寄存器的对应指针。能通过读 SPSSR.SPCP[2:0] 位来确认此指针的值。如果通过将 SPCR.SPE 位置“1”来允许 RSPI 功能，RSPI 就将指向命令的指针设定到 SPCMD0 寄存器，并且在开始串行传送时将 SPCMD0 寄存器的设定内容反映到传送格式。每当各数据传送的下次存取延迟期间结束时，RSPI 将指针进行递增。当构成顺序的最后命令对应的串行传送结束时，RSPI 将指针设定到 SPCMD0 寄存器，重复执行顺序。

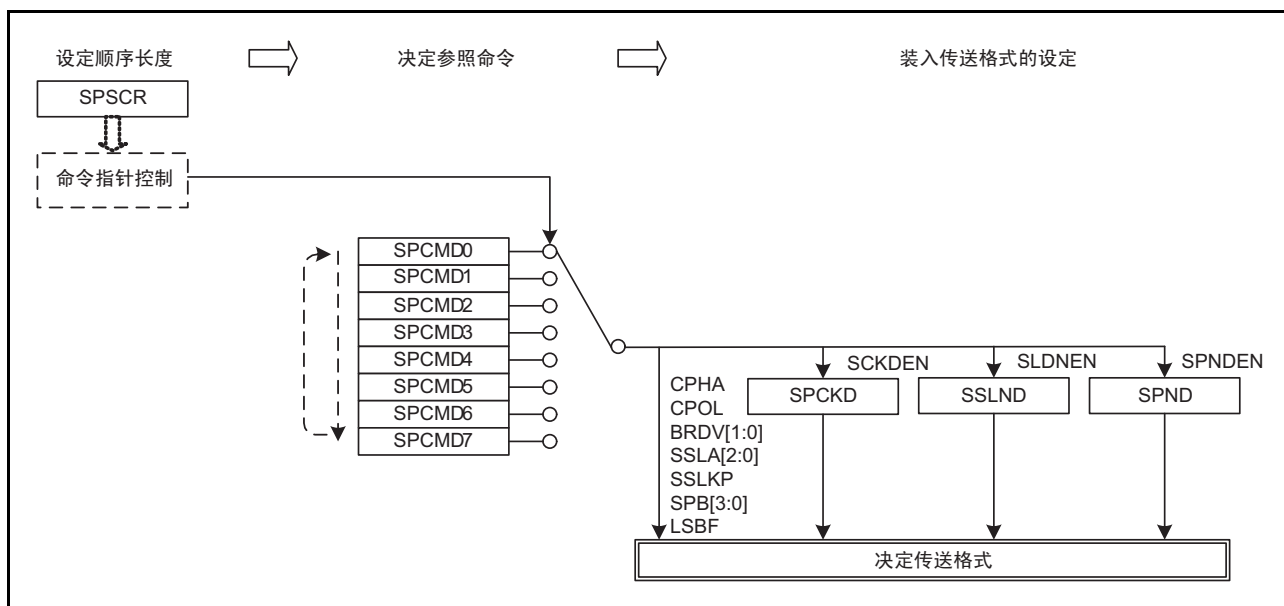


图 29.30 主控模式中的串行传送方式的决定方法 (SPI 运行)

在本章节中，将数据 (SPDR) 和设定 (SPCMDm) 合成 1 帧。

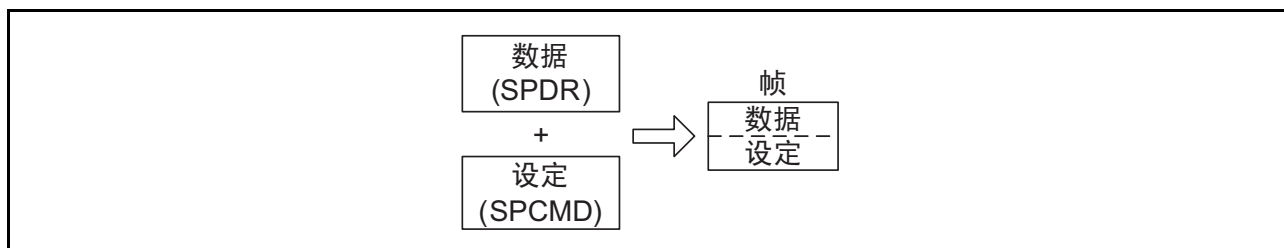


图 29.31 帧的概念图

通过表 29.4 的设定进行顺序运行时的命令和发送缓冲器 / 接收缓冲器的关系如图 29.32 所示。

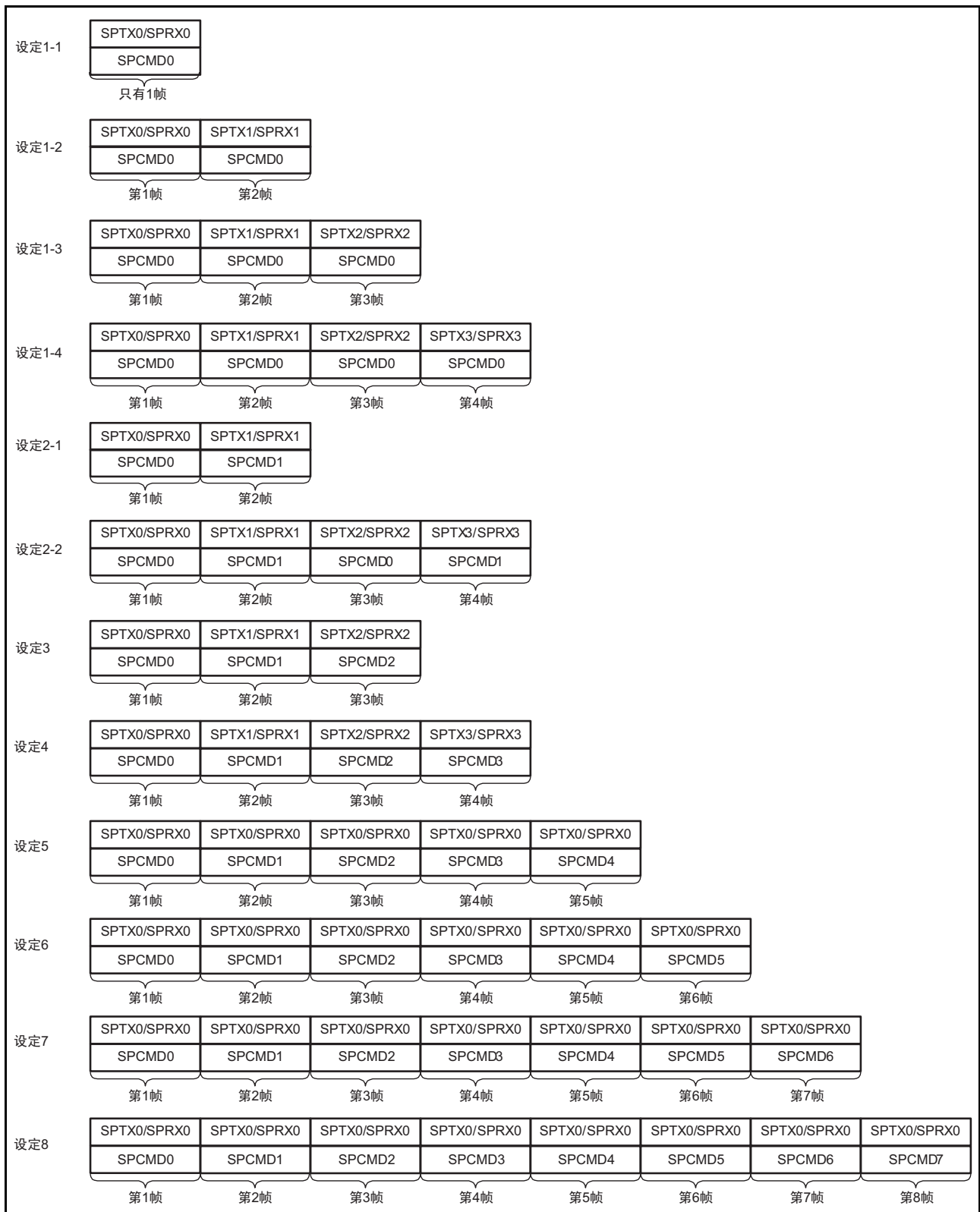


图 29.32 顺序运行时的 RSPI 命令寄存器和发送 / 接收缓冲器

(4) 突发传送

当 RSPI 正在进行的串行传送所参照的 SPCMDm.SSLKP 位为“1”时，RSPI 将串行传送过程中的 SSLAi 信号电平保持到下次串行传送的 SSLAi 信号开始有效为止。如果下次串行传送的 SSLAi 信号电平和正在进行的串行传送的 SSLAi 信号电平相同，RSPI 就能在保持 SSLAi 信号有效的状态下，连续进行串行传送（突发传送）。

通过设定 SPCMD0 寄存器和 SPCMD1 寄存器来实现突发传送时的 SSLAi 信号运行例子如图 29.33 所示。以下说明图 29.33 中记载的 (1) ~ (7) 的 RSPI 运行内容。SSLAi 输出信号的极性取决于 SSLP 寄存器的设定值。

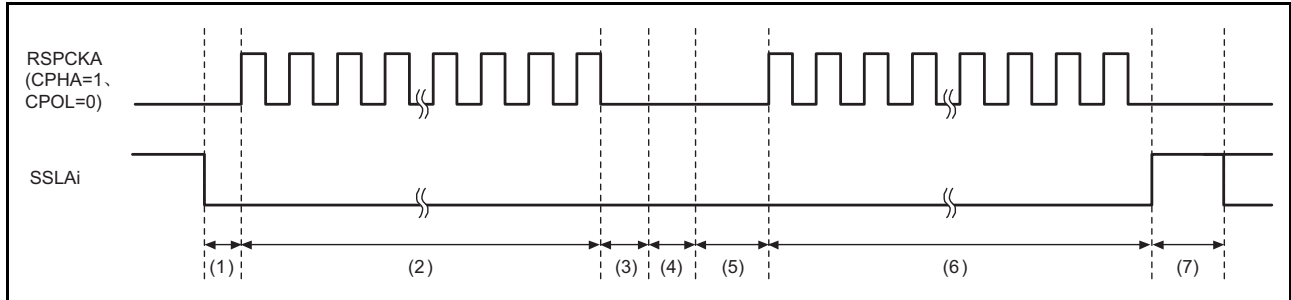


图 29.33 使用 SSLKP 位的突发传送运行例子

- (1) 根据 SPCMD0 寄存器，使 SSLAi 信号有效并且插入 RSPCK 延迟。
- (2) 根据 SPCMD0 寄存器进行串行传送。
- (3) 插入 SSL 无效延迟。
- (4) 因为 SPCMD0.SSLKP 位为“1”，所以保持 SPCMD0 寄存器中的 SSLAi 信号值。在此期间，保持的时间最短也要和下次存取 SPCMD0 寄存器的延迟相同。如果在经过最短期间后移位寄存器变空，就在将下次要传送的发送数据保存到移位寄存器前继续此期间。
- (5) 根据 SPCMD1 寄存器，使 SSLAi 信号有效并且插入 RSPCK 延迟。
- (6) 根据 SPCMD1 寄存器进行串行传送。
- (7) 因为 SPCMD1.SSLKP 位为“0”，所以 SSLAi 信号无效。根据 SPCMD1 寄存器插入下次的存取延迟。

在 SSLKP 位为“1”的 SPCMDm 寄存器所设 SSLAi 信号输出和下次传送使用的 SPCMDm 寄存器所设 SSLAi 信号输出不同的情况下，RSPI 在下次传送命令所对应的 SSLAi 信号有效时（图 29.33 的 (5)）切换 SSLAi 信号状态。必须注意：如果切换 SSLAi 信号，就可能因驱动 MISOA 的从属设备发生竞争而引起信号电平的冲突。

主控模式的 RSPI 在模块内部参照不使用 SSLKP 位时的 SSLAi 信号运行。即使 SPCMDm.CPHA 位为“0”，RSPI 也能使用在内部检测到的下次传送的有效 SSLAi 信号，正确地开始串行传送。因此，与 CPHA 位的设定值无关，能进行主控模式的突发传送（参照“29.3.10 SPI 运行”）。

(5) RSPCK 延迟 (t1)

主控模式的 RSPI 的 RSPCK 延迟值取决于 SPCMDm.SCKDEN 位的设定和 SPCKD 寄存器的设定。RSPI 通过指针控制来决定串行传送要参照的 SPCMDm 寄存器，并且使用所选的 SPCMDm.SCKDEN 位和 SPCKD 寄存器来决定如表 29.9 所示的串行传送时的 RSPCK 延迟值。有关 RSPCK 延迟的定义，请参照“29.3.5 传送格式”。

表 29.9 SCKDEN 位、SPCKD 寄存器和 RSPCK 延迟值的关系

SPCMDm.SCKDEN 位	SPCKD.SCKDL[2:0] 位	RSPCK 延迟值
0	000 ~ 111	1 个 RSPCK
1	000	1 个 RSPCK
	001	2 个 RSPCK
	010	3 个 RSPCK
	011	4 个 RSPCK
	100	5 个 RSPCK
	101	6 个 RSPCK
	110	7 个 RSPCK
	111	8 个 RSPCK

(6) SSL 无效延迟 (t2)

主控模式的 RSPI 的 SSL 无效延迟值取决于 SPCMDm.SLNDEN 位的设定和 SSLND 寄存器的设定。RSPI 通过指针控制来决定串行传送要参照的 SPCMDm 寄存器，并且使用所选的 SPCMDm.SLNDEN 位和 SSLND 寄存器来决定表 29.10 所示的串行传送时的 SSL 无效延迟值。有关 SSL 无效延迟的定义，请参照“29.3.5 传送格式”。

表 29.10 SLNDEN 位、SSLND 寄存器和 SSL 无效延迟值的关系

SPCMDm.SLNDEN 位	SSLND.SLNDL[2:0] 位	SSL 无效延迟值
0	000 ~ 111	1 个 RSPCK
1	000	1 个 RSPCK
	001	2 个 RSPCK
	010	3 个 RSPCK
	011	4 个 RSPCK
	100	5 个 RSPCK
	101	6 个 RSPCK
	110	7 个 RSPCK
	111	8 个 RSPCK

(7) 下次存取延迟 (t3)

主控模式的 RSPI 的下次存取延迟取决于 SPCMDm.SPNDEN 位的设定和 SPND 寄存器的设定。RSPI 通过指针控制来决定串行传送要参照的 SPCMDm 寄存器，并且使用所选的 SPCMDm.SPNDEN 位和 SPND 寄存器来决定如表 29.11 所示的串行传送时的 RSPCK 延迟。有关下次存取延迟的定义，请参照“29.3.5 传送格式”。

表 29.11 SPNDEN 位、SPND 寄存器和下次存取延迟值的关系

SPCMDm.SPNDEN 位	SPND.SPNDL[2:0] 位	下次存取延迟值
0	000 ~ 111	1 个 RSPCK+2 个 PCLK
1	000	1 个 RSPCK+2 个 PCLK
	001	2 个 RSPCK+2 个 PCLK
	010	3 个 RSPCK+2 个 PCLK
	011	4 个 RSPCK+2 个 PCLK
	100	5 个 RSPCK+2 个 PCLK
	101	6 个 RSPCK+2 个 PCLK
	110	7 个 RSPCK+2 个 PCLK
	111	8 个 RSPCK+2 个 PCLK

(8) 初始化流程

在 SPI 运行的情况下，在 主控模式中使用 RSPI 时的初始化流程例子如图 29.34 所示。有关中断控制器、DMAC 和输入 / 输出端口的设定方法，请参照各模块的说明。

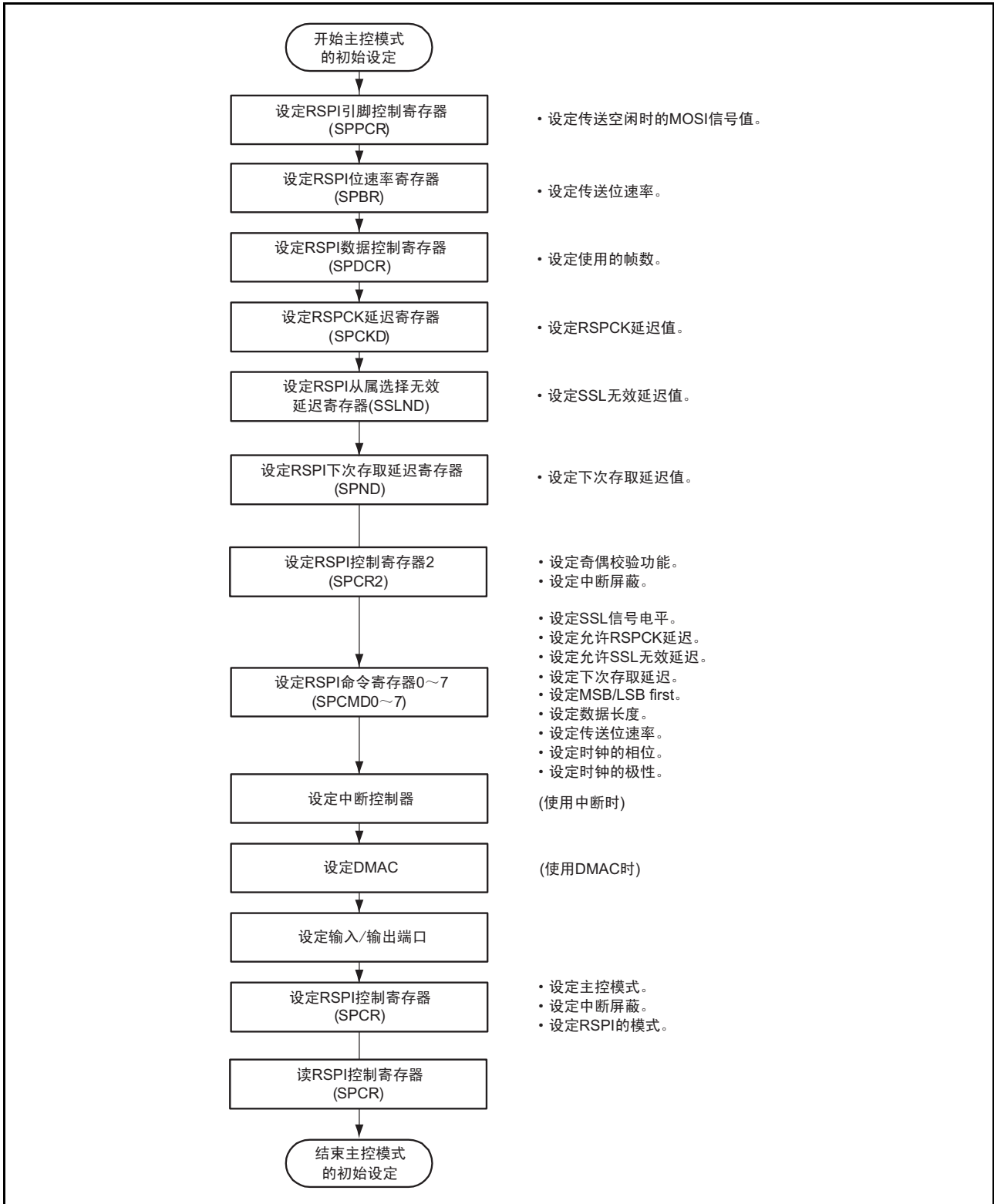


图 29.34 主控模式时的初始化流程例子 (SPI 运行)

(9) 软件处理流程

软件处理流程的例子如图 29.35 ~ 图 29.37 所示。

(a) 发送处理流程

能在进行发送的情况下，通过在写完最后数据后允许 SPII 中断，向 CPU 通知全部数据发送结束。

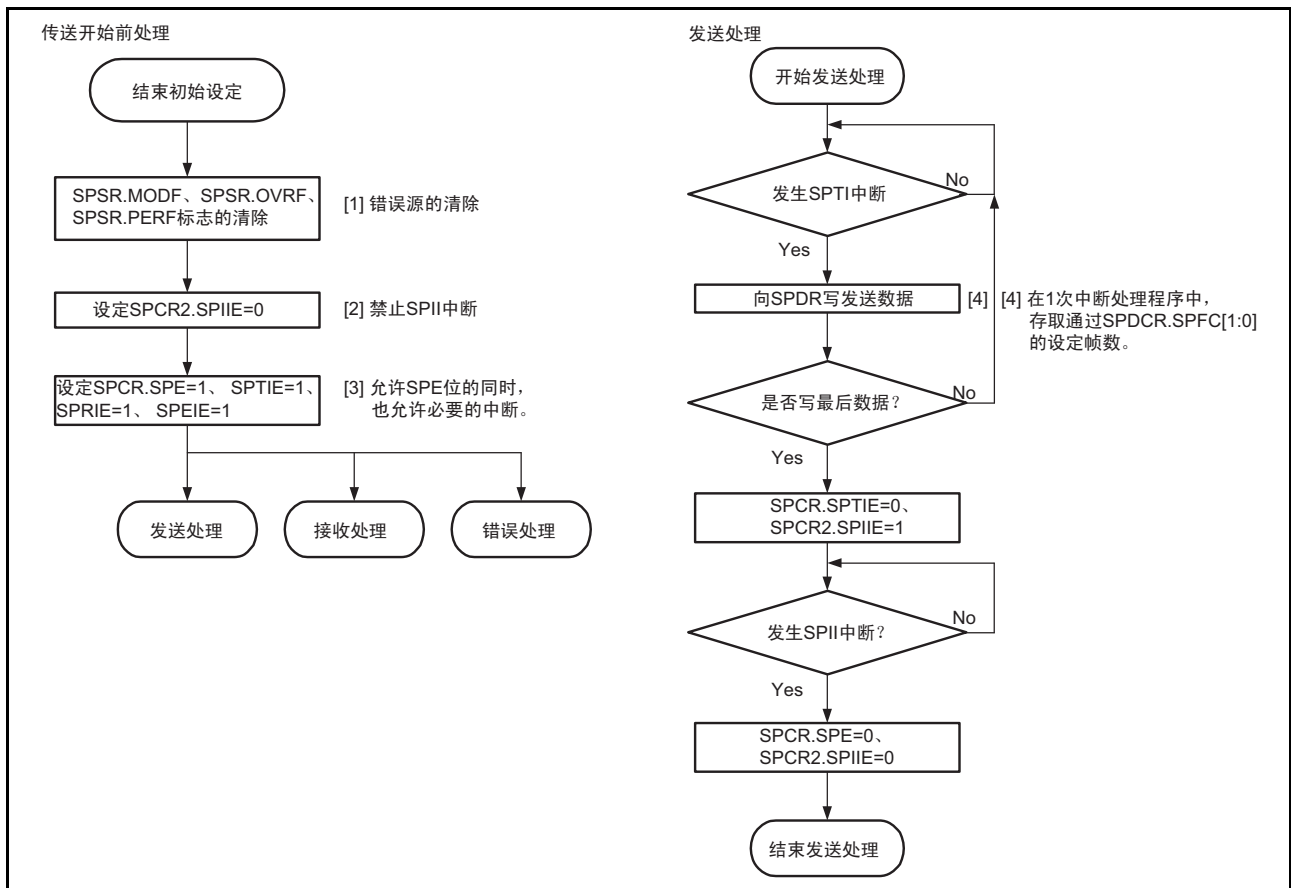


图 29.35 主控模式时的流程图 (发送)

(b) 接收处理流程

RSPI 不能只进行接收运行，总是需要发送运行。

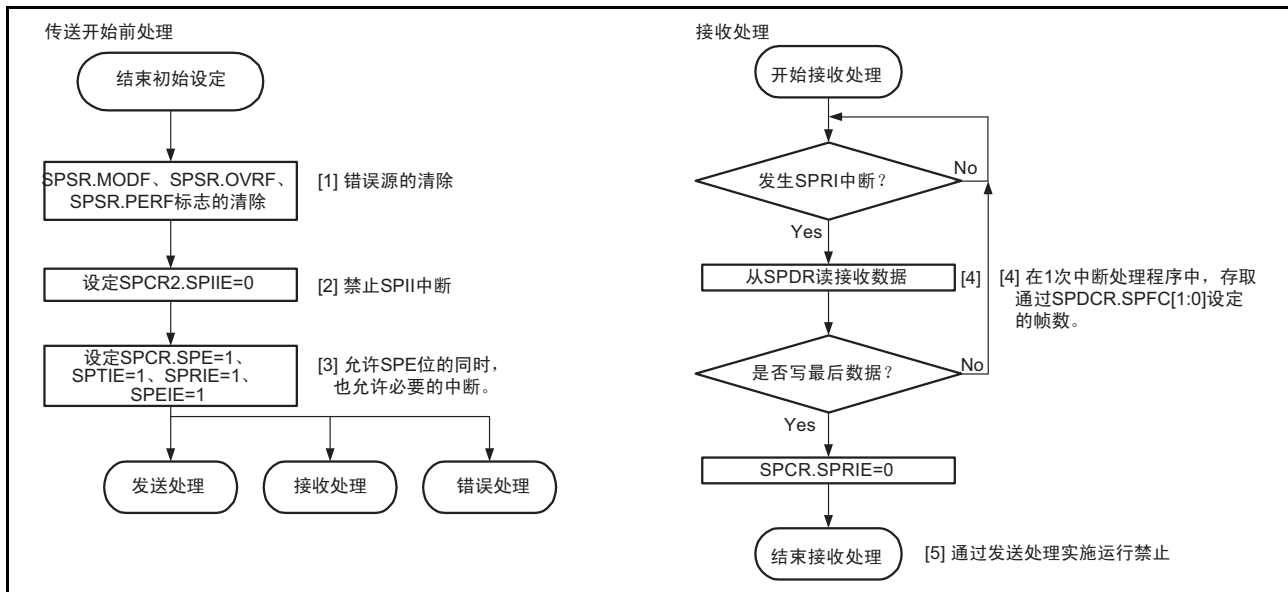


图 29.36 主控模式时的流程图 (接收)

(c) 错误处理流程

RSPI 有 3 种错误。在发生模式故障错误时，自动清除 SPCR.SPE 位，并且停止发送 / 接收运行。但是，因为其他错误源不清除 SPCR.SPE 位，所以发送 / 接收持续运行，因此，在因其他错误源（不是第 1 个错误源）发生错误时，由于 SPSSR.SPECM[2:0] 位被更新，所以推荐清除 SPCR.SPE 位并且停止发送 / 接收运行。

在发生错误时，ICU.IRN.IR 标志有可能保持 SPTI 中断请求或者 SPRI 中断请求，因此必须通过错误处理清除 ICU.IRn.IR 标志。

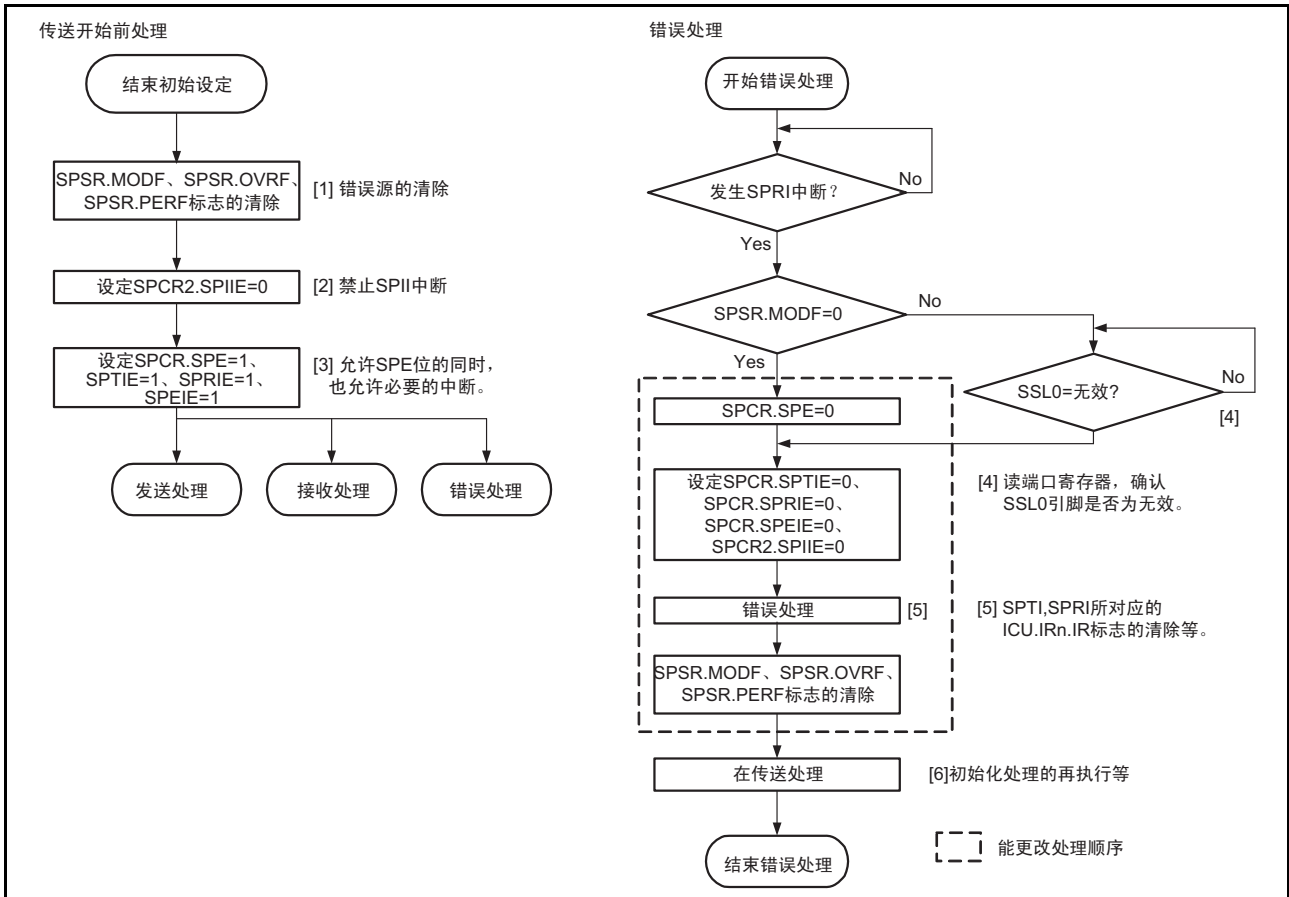


图 29.37 主控模式时的流程图（错误）

29.3.10.2 从属模式的运行

(1) 串行传送的开始

当 SPCMD0.CPHA 位为“0”时，如果 RSPI 检测到 SSLA0 输入信号有效，就需要开始向 MISOA 输出信号驱动有效数据。因此，在 CPHA 位为“0”时，SSLA0 输入信号的有效为串行传送的开始触发。

当 CPHA 位为“1”时，如果在 SSLA0 输入信号有效的状态下 RSPI 检测到最初的 RSPCKA 边沿，就需要开始向 MISOA 输出信号驱动有效数据。因此，在 CPHA 位为“1”时，SSLA0 信号有效状态下的最初的 RSPCKA 边沿为串行传送的开始触发。

如果 RSPI 在移位寄存器为空的状态下检测到串行传送的开始，就将移位寄存器改为满状态，并且不能在串行传送过程中将数据从发送缓冲器传送到移位寄存器。如果在开始串行传送前移位寄存器已满，RSPI 就保持移位寄存器的满状态。

与 CPHA 位的设定无关，RSPI 开始驱动 MISOA 输出信号的时序为 SSLA0 信号的有效时序。RSPI 的输出数据的有效或者无效因 CPHA 位的设定而发生变化。

有关 RSPI 传送格式的详细内容，请参照“29.3.5 传送格式”。SSLA0 输入信号的极性取决于 SSLP.SSL0P 位的设定值。

(2) 串行传送的结束

与 SPCMD0.CPHA 位无关，如果 RSPI 检测到相当于最后采样时序的 RSPCKA 边沿，就结束串行传送。在接收缓冲器为空的情况下，RSPI 在串行传送结束后将接收数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器。与接收缓冲器的状态无关，RSPI 在串行传送结束后将移位寄存器改为空状态。在串行传送开始到串行传送结束的期间，如果 RSPI 检测到 SSLA0 输入信号无效，就发生模式故障错误（参照“29.3.8 错误检测”）。

最后的采样时序取决于传送数据的位长，从属模式的 RSPI 的数据长度取决于 SPCMD0.SPB[3:0] 位的设定值，SSLA0 输入信号的极性取决于 SSLP.SSL0P 位的设定值。有关 RSPI 传送格式的详细内容，请参照“29.3.5 传送格式”。

(3) 单从属模式中的注意事项

当 SPCMD0.CPHA 位为“0”时，如果 RSPI 检测到 SSLA0 输入信号的有效边沿，就开始串行传送。在图 29.7 的例子所示的结构中，在单从属模式中使用 RSPI 时，因为 SSLA0 输入信号总是被固定为有效状态，所以 CPHA 位为“0”的 RSPI 无法正常开始串行传送。在 SSLA0 输入信号被固定为有效状态的结构中，要使从属模式的 RSPI 正常进行发送和接收，必须将 CPHA 位置“1”。如果需要将 CPHA 位置“0”，就不能固定 SSLA0 输入信号。

(4) 突发传送

当 SPCMD0.CPHA 位为“1”时，能在保持 SSLA0 输入信号有效的状态下进行连续的串行传送（突发传送）。当 CPHA 位为“1”时，从 SSLA0 输入信号有效状态的最初 RSPCKA 边沿到最后位的接收采样时序的期间相当于串行传送期间。即使 SSLA0 输入信号保持有效电平，因为能检测到存取的开始，所以仍然能应对突发传送。

当 CPHA 位为“0”时，不能正确进行突发传送第 2 次以后的串行传送。

(5) 初始化流程

在SPI运行的情况下，在从属模式中使用RSPI时的初始化流程例子如图29.38所示。有关中断控制器、DMAC和输入/输出端口的设定方法，请参照各块的说明。

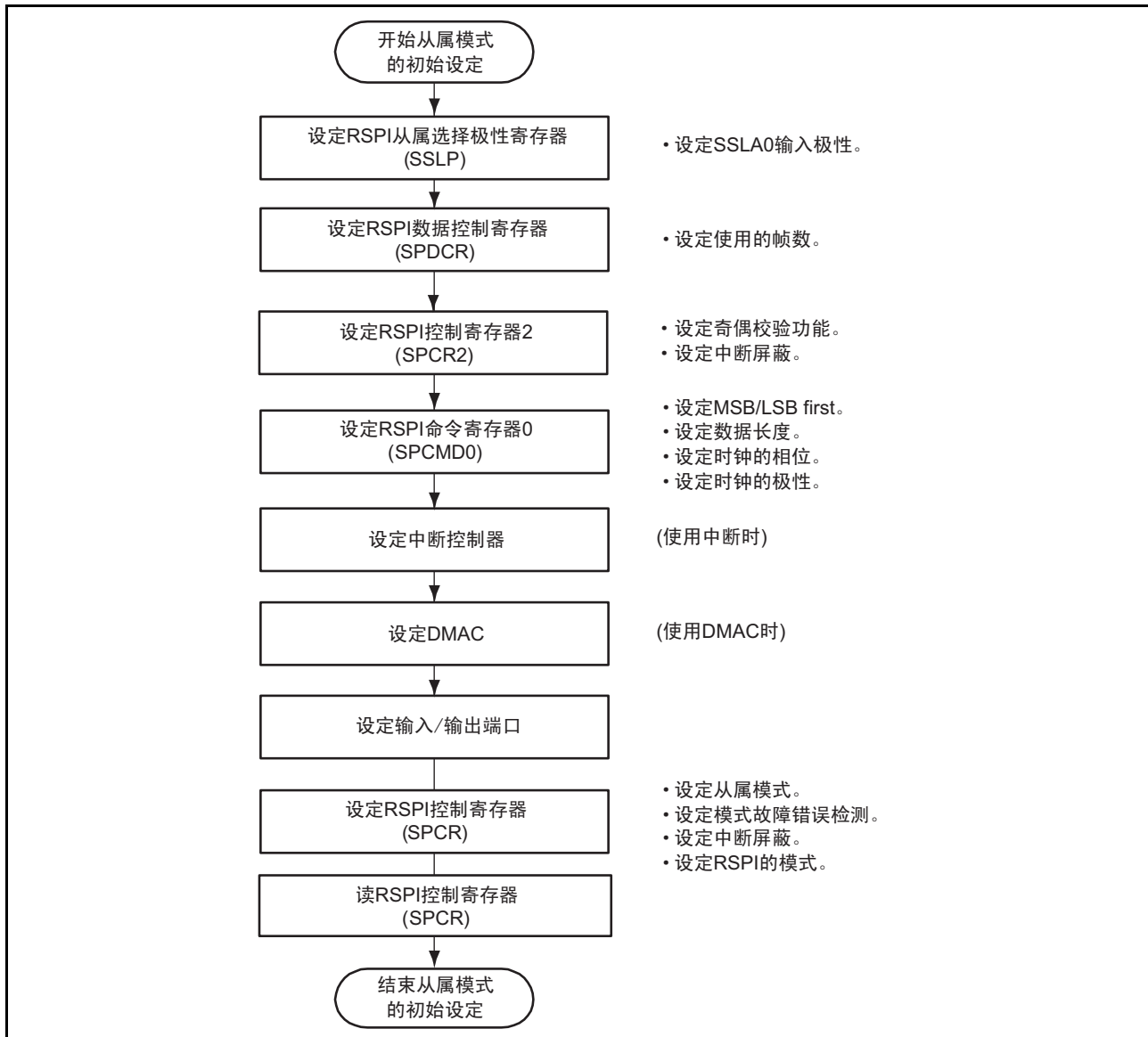


图 29.38 从属模式时的初始化流程例子 (SPI 运行)

(6) 软件处理流程

软件处理流程的例子如图 29.39 ~ 图 29.41 所示。

(a) 发送处理流程

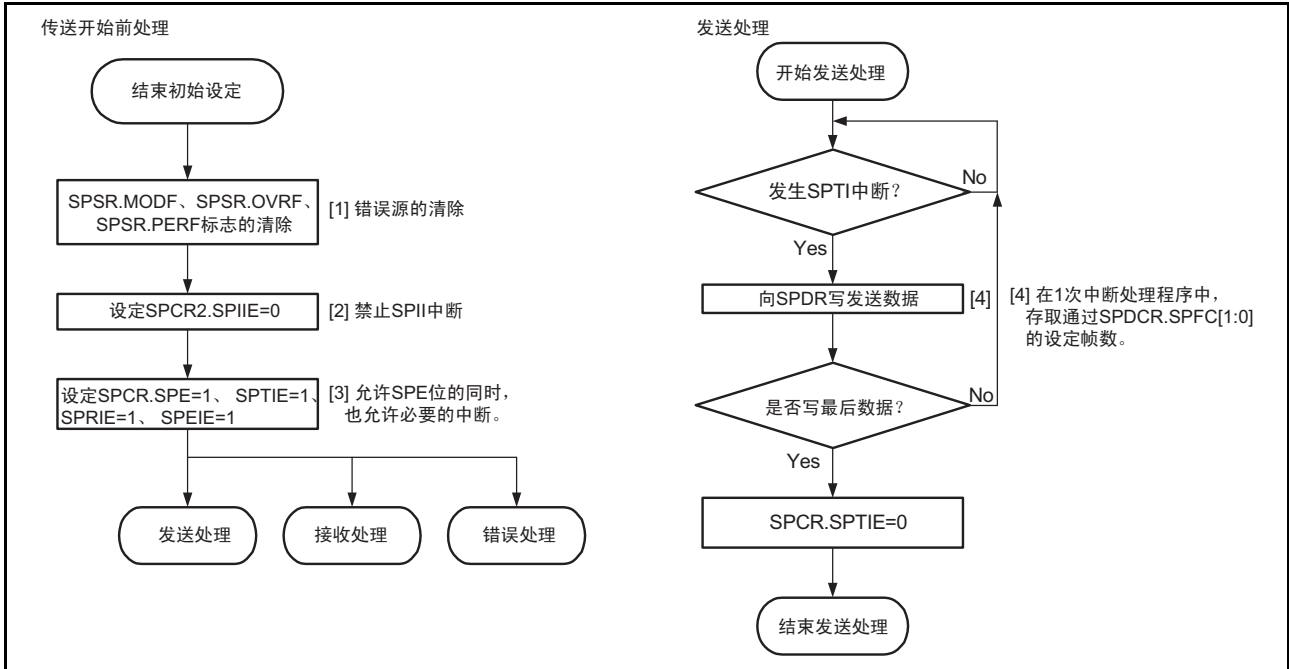


图 29.39 从属模式时的流程图 (发送)

(b) 接收处理流程

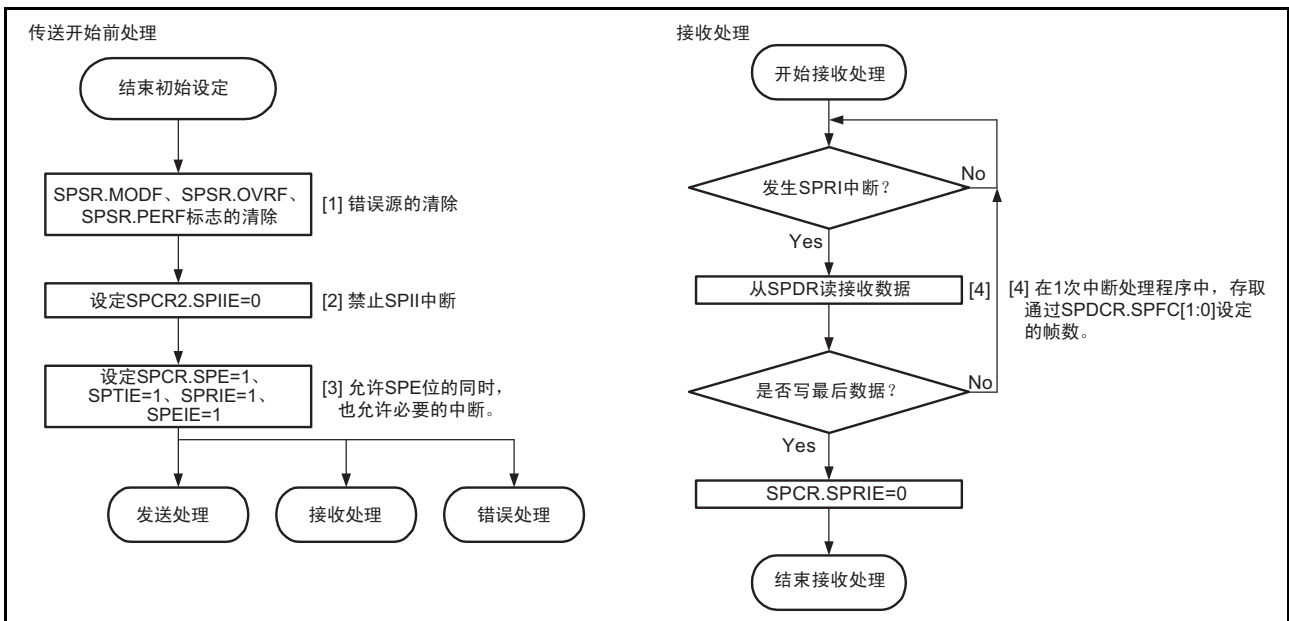


图 29.40 从属模式时的流程图 (接收)

(c) 错误处理流程

即使发生模式故障错误，从属运行也不断言引脚，能清除 SPSR.MODF 标志。

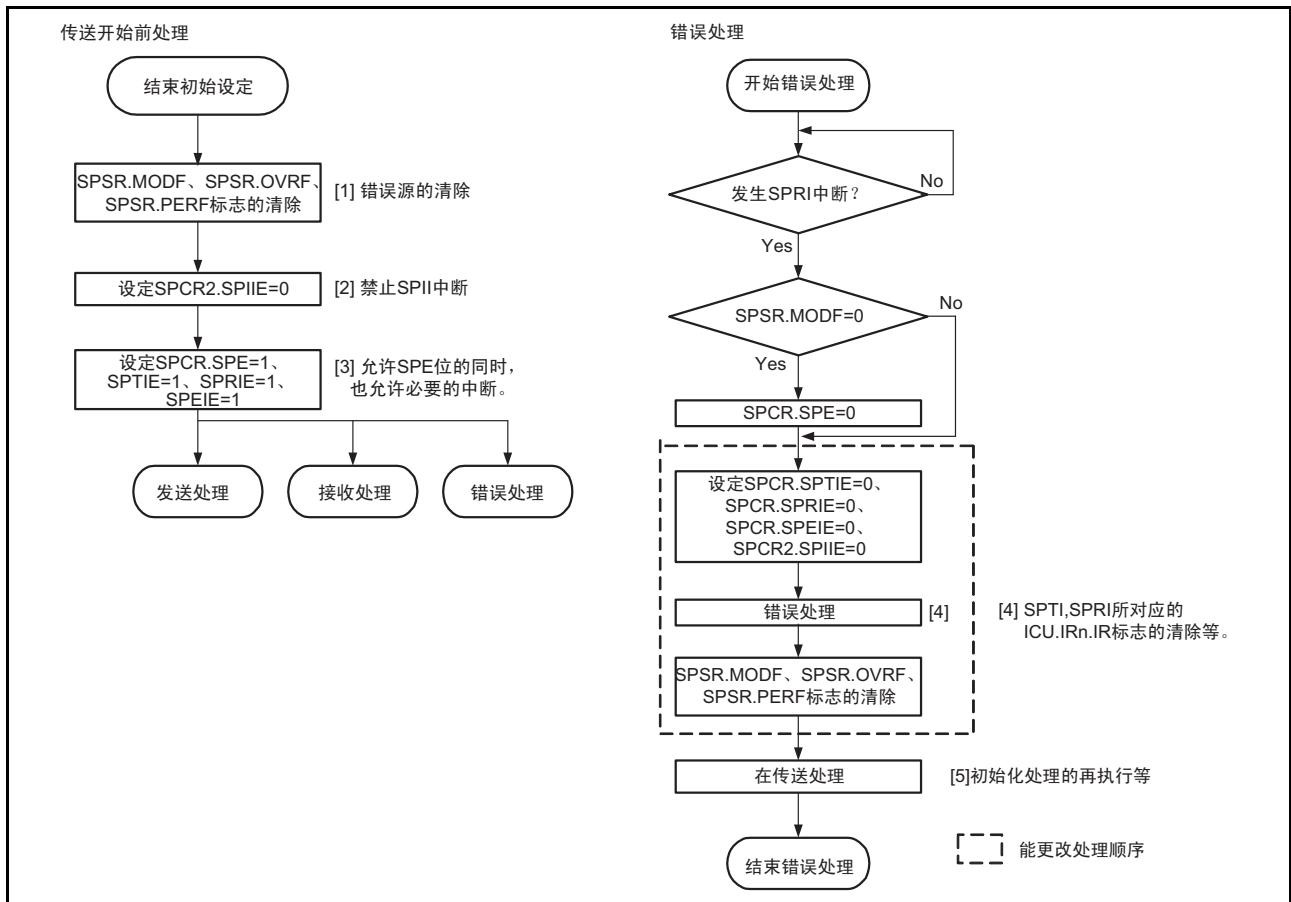


图 29.41 从属模式时的流程图 (错误处理)

29.3.11 时钟同步运行

在 SPCR.SPMS 位为“1”时，RSPI 进行时钟同步运行。在进行时钟同步运行时，不使用 SSLAi 引脚而使用 RSPCKA、MOSIA 和 MISOA 这 3 个引脚进行通信，SSLAi 引脚能用作 I/O 端口。

在进行时钟同步运行时，不使用 SSLAi 引脚进行通信，但是模块内部的运行和 SPI 运行模式相同。在主控运行或者从属运行中，能按照和 SPI 运行时相同的流程进行通信。因为不使用 SSLAi 引脚，所以检测不到模式故障错误。

在进行时钟同步运行时，无法保证从属模式中 (SPCR.MSTR=0) SPCMDm.CPHA 位为“0”时的运行。

29.3.12 主控模式的运行

(1) 串行传送的开始

在发送缓冲器为空的（未设定下次传送的数据）状态下，如果将数据写到 SPDR 寄存器，RSPI 就更新 SPDR 寄存器的发送缓冲器 (SPTX) 的数据。在将 SPCR.SPFC[1:0] 位所设数据写到 SPDR 寄存器后移位寄存器变空时，RSPI 在将发送缓冲器的数据复制到移位寄存器后开始串行传送。如果 RSPI 将发送数据复制到移位寄存器，就将移位寄存器改为满状态；如果串行传送结束，就将移位寄存器改为空状态。不能参照移位寄存器的状态。

有关 RSPI 传送格式的详细内容，请参照“29.3.5 传送格式”。但是，在进行时钟同步通信时，不使用 SSLA0 输出信号。

(2) 串行传送的结束

如果 RSPI 发送与最后采样时序对应的 RSPCKA 边沿，就结束串行传送。在接收缓冲器 (SPRX) 为空的情况下，在串行传送结束后将数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器。

最后的采样时序取决于传送数据的位长，主控模式的 RSPI 的数据长度取决于 SPCMDm.SPB[3:0] 位的设定值。有关 RSPI 传送格式的详细内容，请参照“29.3.5 传送格式”。但是，在进行时钟同步通信时，不使用 SSLA0 输出信号。

(3) 顺序控制

主控模式中的传送格式取决于 SPSCR 寄存器、SPCMDm 寄存器、SPBR 寄存器、SPCKD 寄存器、SSLND 寄存器或者 SPND 寄存器。在进行时钟同步运行时，不输出 SSLAi 信号，但是这些设定有效。

SPSCR 寄存器决定主控模式的 RSPI 要执行的串行传送的顺序结构。给 SPCMDm 寄存器设定 SSLAi 输出信号值、MSB/LSB first、数据长度、部分位速率、RSPCKA 极性 / 相位以及设定是否要参照 SPCKD 寄存器、SSLND 寄存器和 SPND 寄存器；给 SPBR 寄存器设定部分位速率，给 SPCKD 寄存器设定 RSPI 时钟延迟值，给 SSLND 寄存器设定 SSL 无效延迟并且给 SPND 寄存器设定下次存取延迟值。

RSPI 根据 SPSCR 寄存器设定的顺序长度，由部分或者全部 SPCMDm 寄存器构成顺序。RSPI 有构成顺序的 SPCMDm 寄存器的对应指针。能通过读 SPSSR.SPCP[2:0] 位来确认此指针的值。如果通过将 SPCR.SPE 位置“1”来允许 RSPI 运行，RSPI 就将指向命令的指针设定到 SPCMD0 寄存器，并且在开始串行传送时将 SPCMD0 寄存器的设定内容反映到传送格式。每当各数据传送的下次存取延迟期间结束时，RSPI 将指针进行递增。当构成顺序的最后命令对应的串行传送结束时，RSPI 将指针设定到 SPCMD0 寄存器，重复执行顺序。

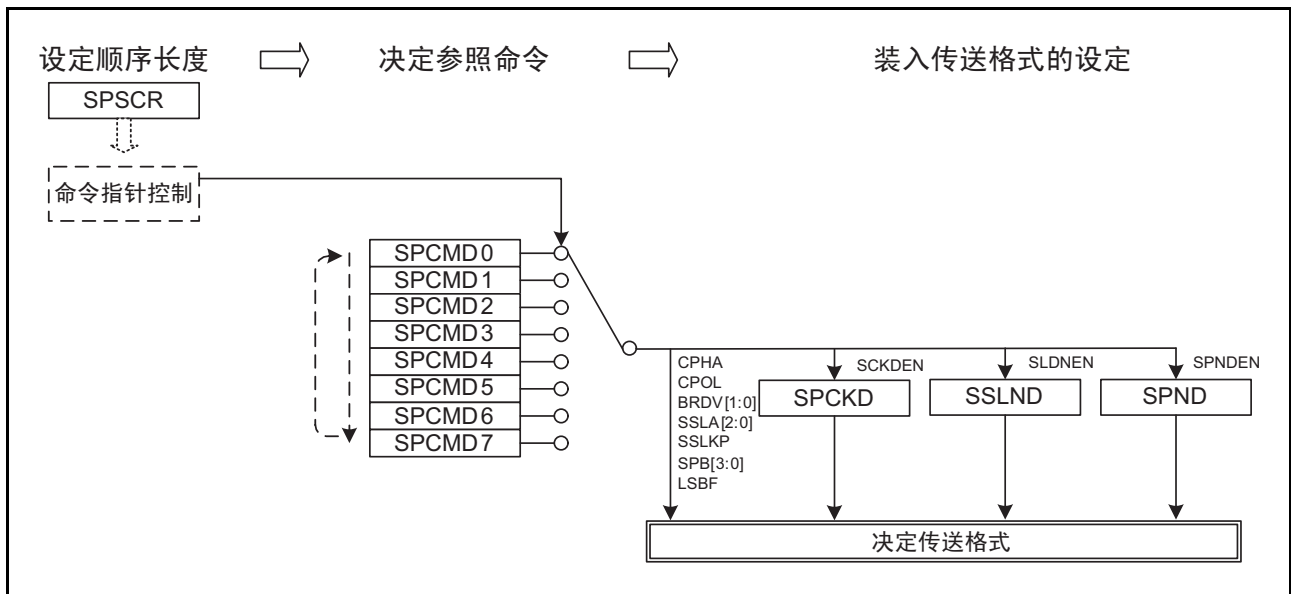


图 29.42 主控模式中的串行传送方式的决定方法 (时钟同步运行)

在本章节中，将数据 (SPDR) 和设定 (SPCMDm) 合成 1 帧。

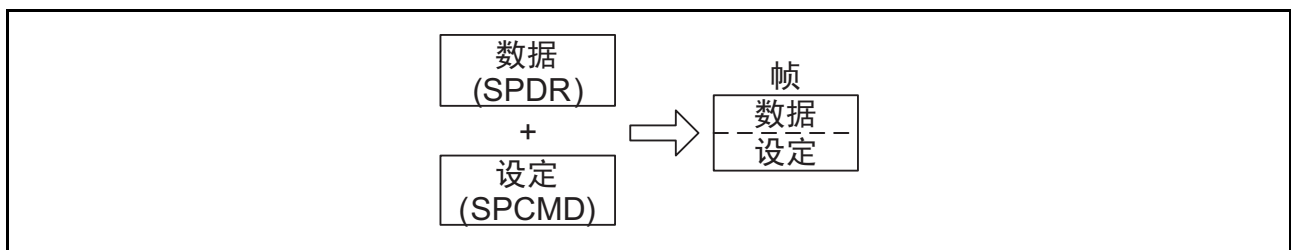


图 29.43 帧的概念图

通过表 29.4 的设置进行顺序运行时的命令和发送缓冲器 / 接收缓冲器的关系如图 29.45 所示。

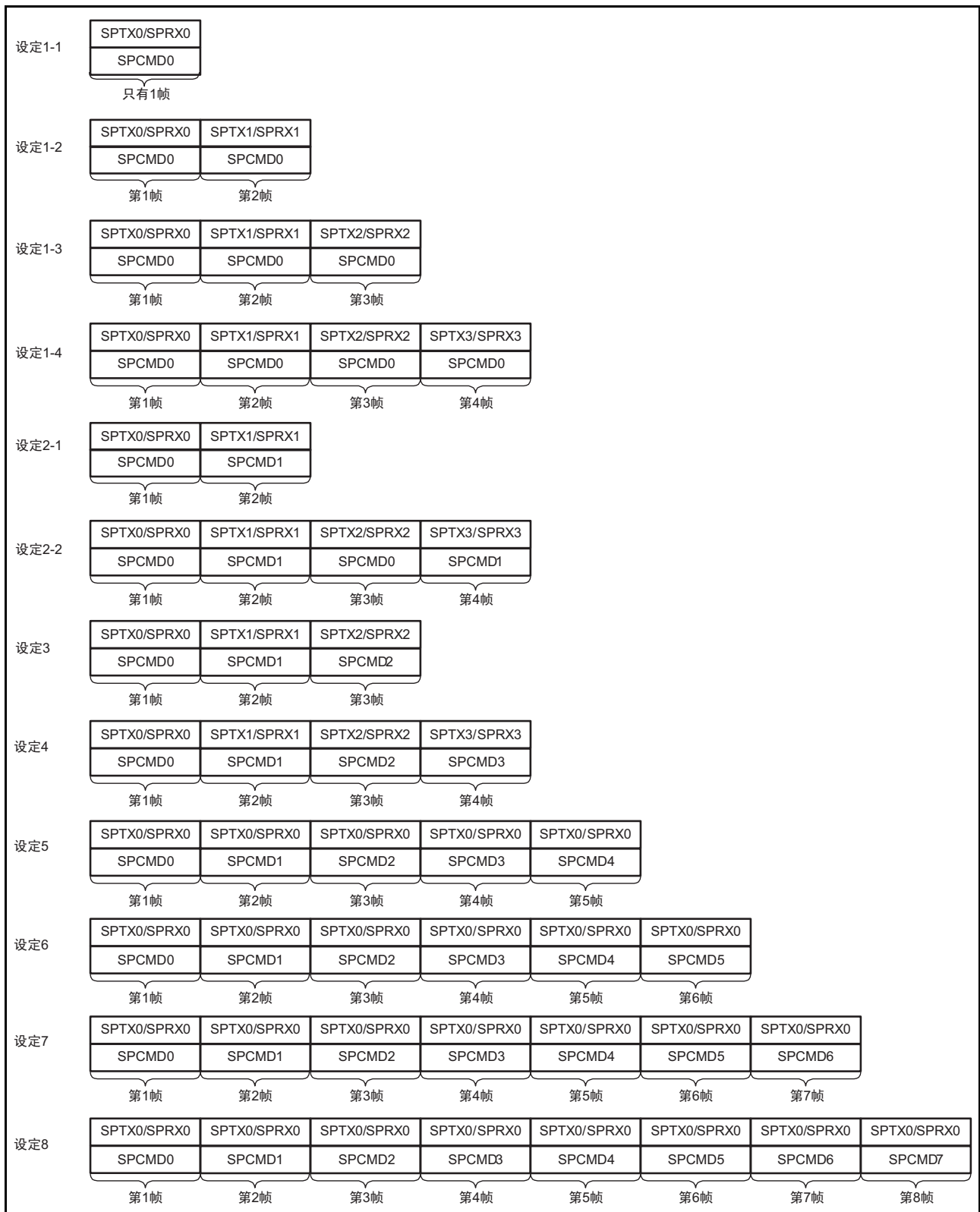


图 29.44 顺序运行时的 RSPI 命令寄存器和发送 / 接收缓冲器的对应

(4) 初始化流程

在时钟同步运行的情况下，在**主控模式**中使用**RSPI**时的初始化流程例子如图 29.45 所示。有关中断控制器、**DMAC** 和输入 / 输出端口的设定方法，请参照各块的说明。

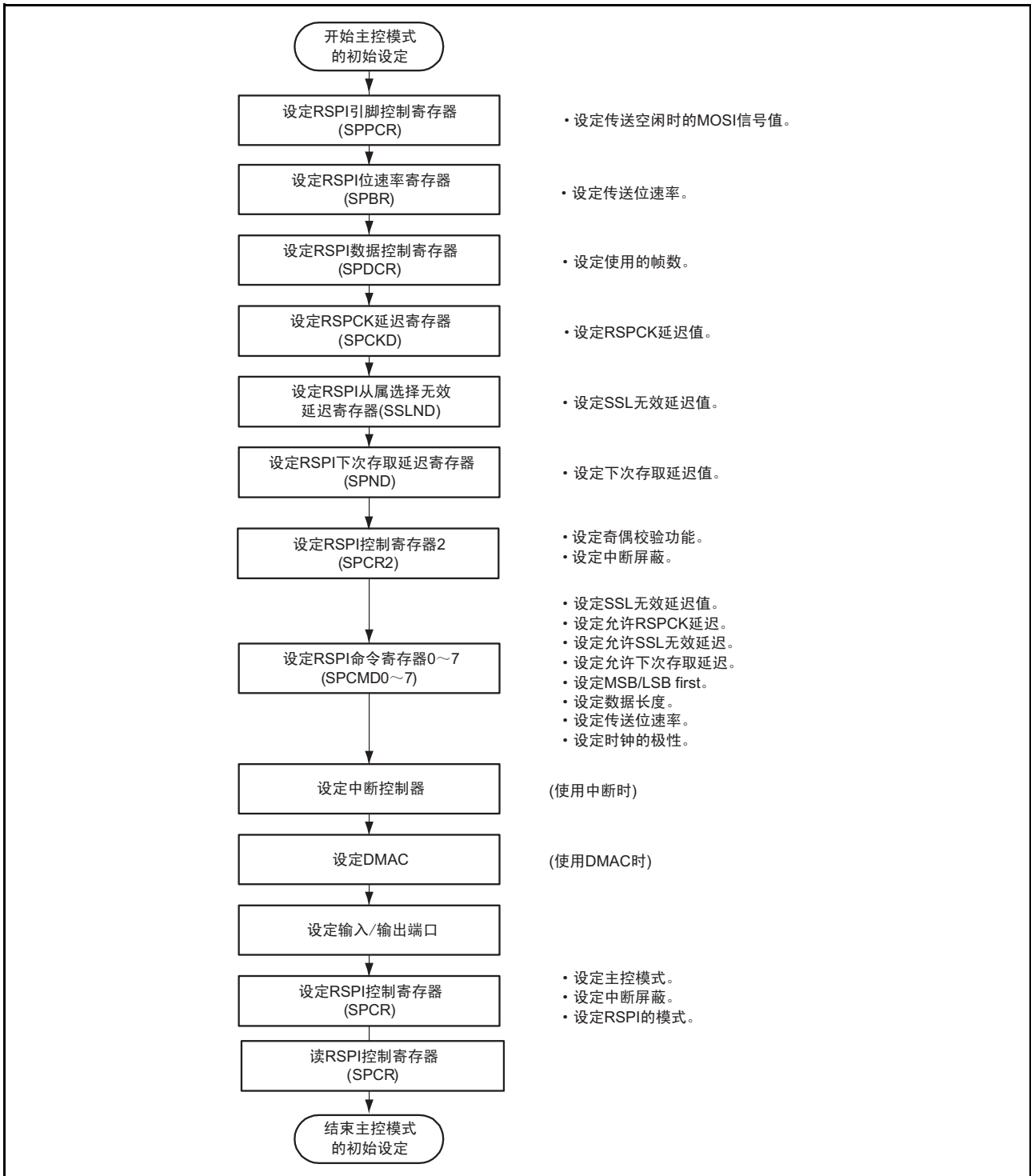


图 29.45 主控模式时的初始化流程例子 (时钟同步运行)

(5) 软件处理流程

时钟同步运行时的**主控模式**的软件处理与**SPI**运行时的**主控模式**的软件处理流程相同。详细内容请参照“29.3.10.1 (9) 软件处理流程”。但是，不发生模式故障错误。

29.3.13 从属模式的运行

(1) 串行传送的开始

在 SPCR.SPMS 位为“1”时，RSPI 的最初的 RSPCKA 边沿为串行传送的开始触发。

如果 RSPI 在移位寄存器为空的状态下检测到串行传送的开始，就将移位寄存器改为满状态，并且不能在串行传送过程中将数据从发送缓冲器复制到移位寄存器。如果在开始串行传送前移位寄存器已满，RSPI 就保持移位寄存器的满状态。

在 SPMS 位为“1”时，RSPI 随时驱动 MISOA 输出信号。

有关 RSPI 传送格式的详细内容，请参照“29.3.5 传送格式”。但是，在时钟同步运行时，不使用 SSL0 输入信号。

(2) 串行传送的结束

如果 RSPI 检测到相当于最后采样时序的 RSPCKA 边沿，就结束串行传送。在接收缓冲器为空的情况下，RSPI 在串行传送结束后将接收数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器。在串行传送结束后，RSPI 将移位寄存器改为空状态。最后的采样时序取决于传送数据的位长，从属模式的 RSPI 的数据长度取决于 SPCMD0.SPB[3:0] 位的设定值。有关 RSPI 传送格式的详细内容，请参照“29.3.5 传送格式”。

(3) 初始化流程

在时钟同步运行的情况下，在从属模式中使用 RSPI 时的初始化流程例子如图 29.46 所示。有关中断控制器、DMAC 和输入 / 输出端口的设定方法，请参照各块的说明。

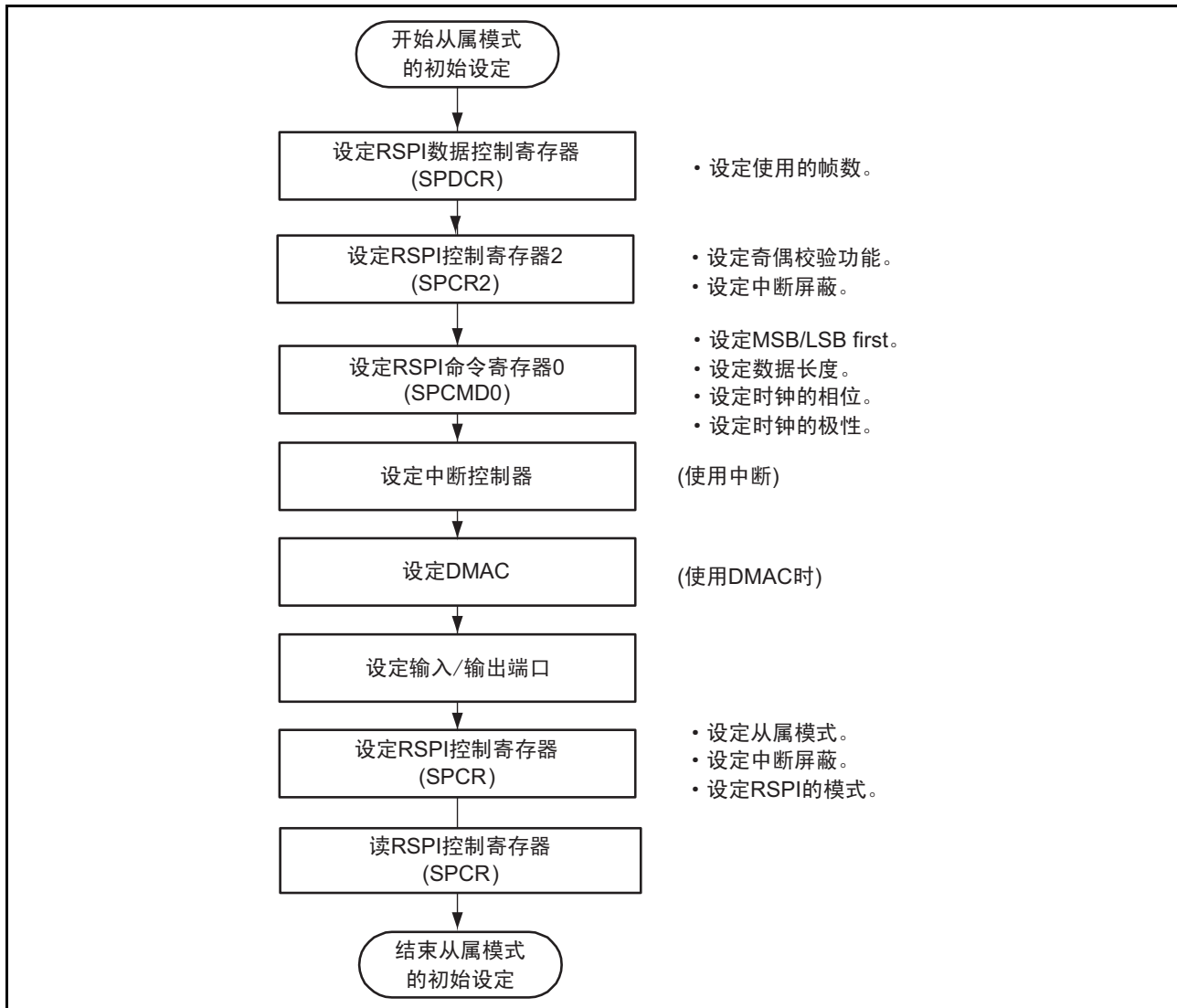


图 29.46 从属模式时的初始化流程例子 (时钟同步运行)

(4) 软件处理流程

时钟同步运行时的主控模式的软件处理与 SPI 运行时的主控模式的软件处理流程相同。详细内容，请参照“29.3.10.2 (6) 软件处理流程”。但是，不发生模式故障错误。

29.3.14 环回模式

在给 SPPCR.SPLP2 位或者 SPPCR.SPLP 位写“1”时，如果 SPCR.MSTR 位为“1”，RSPI 就切断 MISOA 引脚和移位寄存器之间的线路，而连接移位寄存器的输入线路和输出线路；如果 SPCR.MSTR 位为“0”，就切断 MOSIA 引脚和移位寄存器之间的线路，而连接移位寄存器的输入线路和输出线路。另外，如果 SPCR.MSTR 位为“1”，就切断 MOSIA 引脚和移位寄存器之间的线路；如果 SPCR.MSTR 位为“0”，就切断 MISOA 引脚和移位寄存器之间的线路。称此为环回模式。如果通过环回模式进行串行传送，RSPI 的发送数据或者发送数据取反后的数据就为 RSPI 的接收数据。

SPLP2 位和 SPLP 位的设定和接收数据的关系如表 29.12 所示，将主控模式的 RSPI 设定为环回模式 (SPPCR.SPLP2=0、SPPCR.SPLP=1) 时的移位寄存器输入 / 输出线路的结构如图 29.47 所示。

表 29.12 SPLP2 位、SPLP 位的设定和接收数据

SPPCR.SPLP2 位	SPPCR.SPLP 位	接收数据
0	0	MOSIA 引脚或者 MISOA 引脚的输入数据
0	1	发送数据取反后的数据
1	0	发送数据
1	1	发送数据

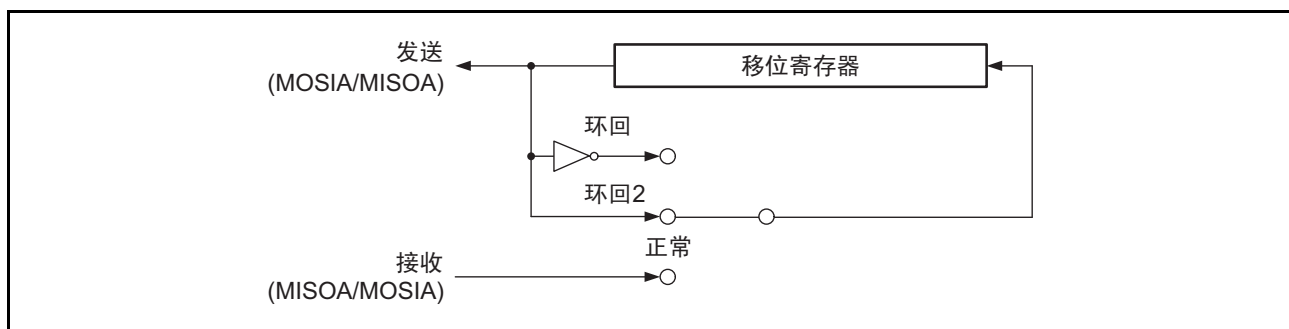


图 29.47 环回模式时的移位寄存器输入 / 输出结构 (主控模式)

29.3.15 奇偶校验位功能的自诊断

奇偶校验电路由发送数据的奇偶校验附加部分和接收数据的错误检测部分构成。要检测奇偶校验电路的奇偶校验附加部分和错误检测部分的故障时，必须按照如图 29.48 所示的流程，进行奇偶校验电路的自诊断。

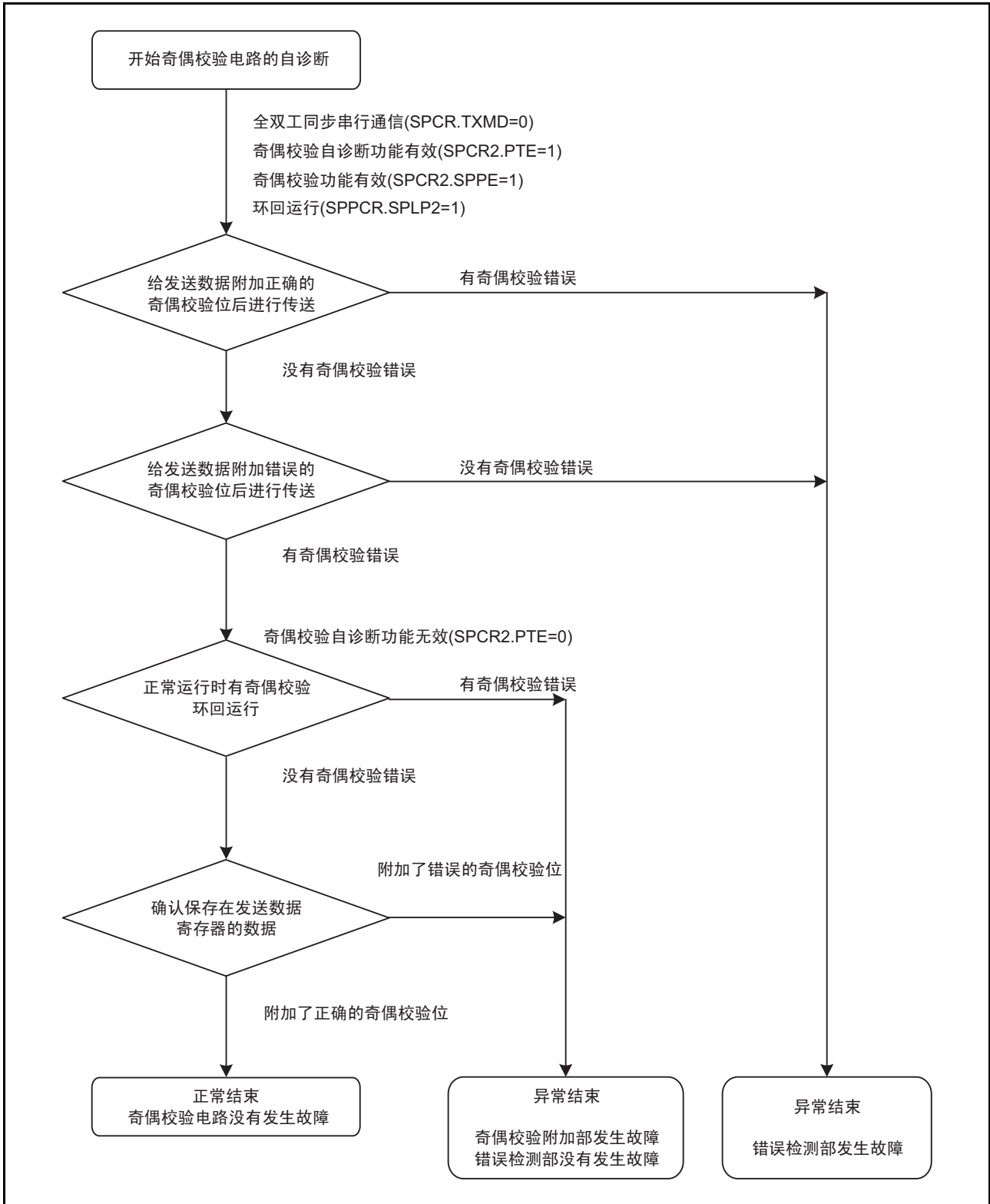


图 29.48 奇偶校验电路的自诊断流程

29.3.16 中断源

RSPI 的中断源有接收缓冲器满、发送缓冲器空、模式故障、溢出、奇偶校验错误和 RSPI 空闲。能通过接收缓冲器满或者发送缓冲器空的中断请求来启动 DTC 或者 DMAC，进行数据传送。

因为模式故障、溢出和奇偶校验错误的中断请求分配到 SPEI 的向量地址，所以需要通过标志判断中断源。RSPI 的中断源如表 29.13 所示。一旦表 29.13 所示的中断条件成立，就产生中断。对于接收缓冲器满和发送缓冲器空的中断源，必须通过数据传送来清除中断源。

在使用 DTC 或者 DMAC 进行发送或者接收时，必须先设定 DTC 或者 DMAC，然后在设定为允许状态后设定 RSPI。有关 DTC 或者 DMAC 的设定方法，请参照“17. DMA 控制器 (DMACA)”和“18. 数据传送控制器 (DTCa)”。

对于发送缓冲器空中断和接收缓冲器满中断，即使在 ICU.IRn.IR 标志为“1”时满足中断产生条件，也不将中断请求输出到 ICU 而保持在内部（在内部能保持的容量是每个中断源 1 个请求）。如果 ICU.IRn.IR 标志变为“0”，就将保持的中断请求输出到 ICU。如果输出被保持的中断请求，就自动清除保持的中断请求。也能通过将对应的中断允许位（SPCR.SPTIE 位或者 SPCR.SPRIE 位）置“0”来清除保持在内部的中断请求。

表 29.13 RSPI 的中断源

中断源	略称	中断条件	DMAC/DTC 的启动
接收缓冲器满	SPRI	在 SPCR.SPRIE 位为“1”的状态下接收缓冲器变满时	能
发送缓冲器空	SPTI	在 SPCR.SPTIE 位为“1”的状态下发送缓冲器变空时	能
RSPI 错误（模式故障、溢出、奇偶校验错误）	SPEI	在 SPCR.SPEIE 位为“1”的状态下 SPSR.MODF、SPSR.OVRF 和 SPSR.PERF 标志变为“1”时	不能
RSPI 空闲	SPII	在 SPCR2.SPIIE 位为“1”的状态下 IDLNF 标志变为“0”时	不能

29.4 使用时的注意事项

29.4.1 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 RSPI 的运行，复位后的值为停止 RSPI 的运行。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“11. 低功耗功能”。

29.4.2 低功耗功能的注意事项

当使用低功耗功能来降低 RSPI 功耗时，必须在将 SPCR.SPE 位置“0”并且结束通信后使用低功耗功能。

29.4.3 有关开始通信的注意事项

如果在 ICU.IRn.IR 标志为“1”的状态下开始通信，通信开始后的中断请求就被保持在内部，所以 ICU.IRn.IR 标志可能会导致意想不到的动作。

如果在开始通信时 ICU.IRn.IR 标志为“1”，就必须在允许运行（将 SPCR.SPE 位置“1”）前按以下步骤清除中断请求：

1. 确认通信处于停止状态（SPCR.SPE 位为“0”）。
2. 将对应的中断允许位（SPCR.SPTIE 位或者 SPCR.SPRIE 位）置“0”。
3. 读对应的中断允许位（SPCR.SPTIE 位或者 SPCR.SPRIE 位），确认是“0”。
4. 将 ICU.IRn.IR 标志置“0”。

30. CRC 运算器 (CRC)

CRC (Cyclic Redundancy Check) 运算器生成数据块的 CRC 码。

30.1 概要

CRC 运算器的规格和框图分别如表 30.1 和图 30.1 所示。

表 30.1 CRC 运算器的规格

项目	内容
CRC 运算的对象数据 (注 1)	对 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。

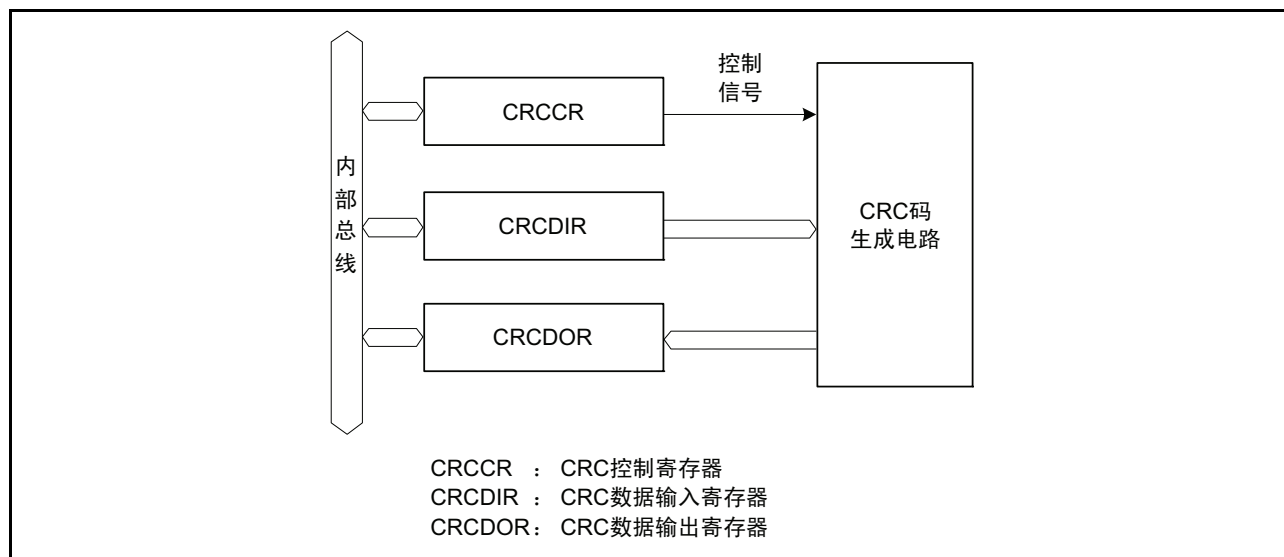
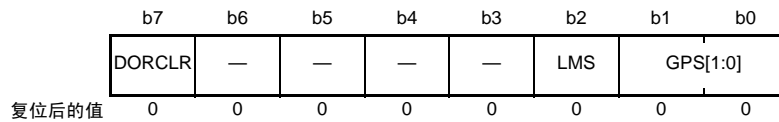


图 30.1 CRC 运算器的框图

30.2 寄存器说明

30.2.1 CRC 控制寄存器 (CRCCR)

地址 0008 8280h



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 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: 将进行 LSB first 通信时的 CRC 运算的 CRCDOR 寄存器值 (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: 将进行 MSB first 通信时的 CRC 运算的 CRCDOR 寄存器值 (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	1: 清除 CRCDOR 寄存器 (注 1) 读取值为“0”。	W

注 1. 只能写“1”。

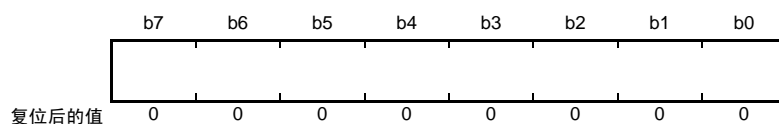
DORCLR 位 (CRCDOR 寄存器清除位)

如果将 DORCLR 位置“1”，CRCDOR 寄存器就变为“0000h”。

只能写“1”，读取值为“0”。

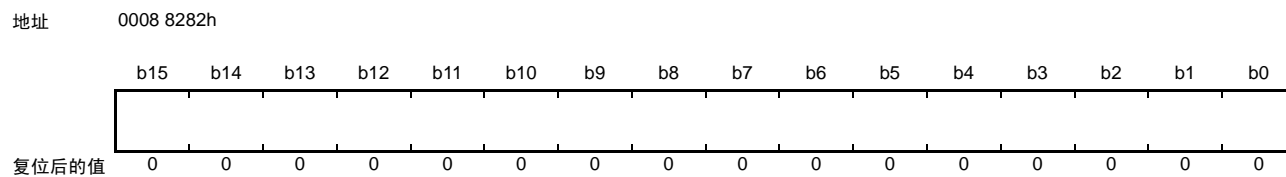
30.2.2 CRC 数据输入寄存器 (CRCDIR)

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。

30.2.3 CRC 数据输出寄存器 (CRCDOR)



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) 就被更新。

30.3 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)，生成数据 “F0h” 的 CRC 码的例子如下所示。

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。

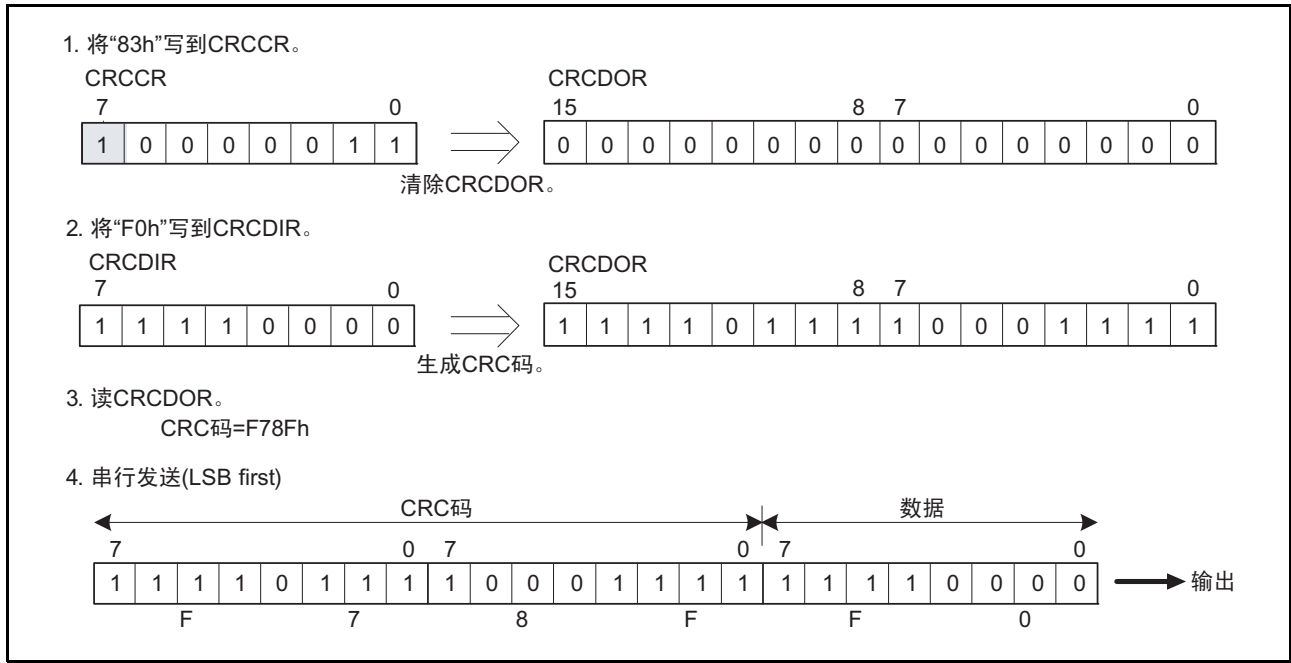


图 30.2 LSB first 的数据发送

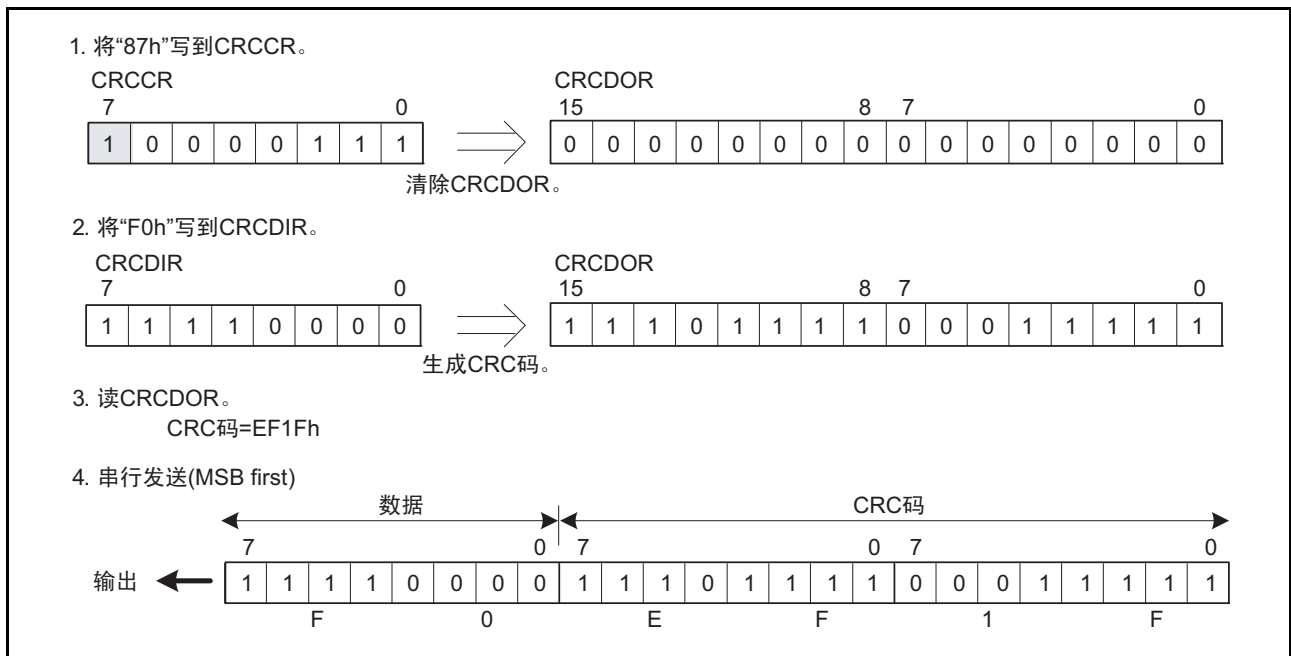


图 30.3 MSB first 的数据发送

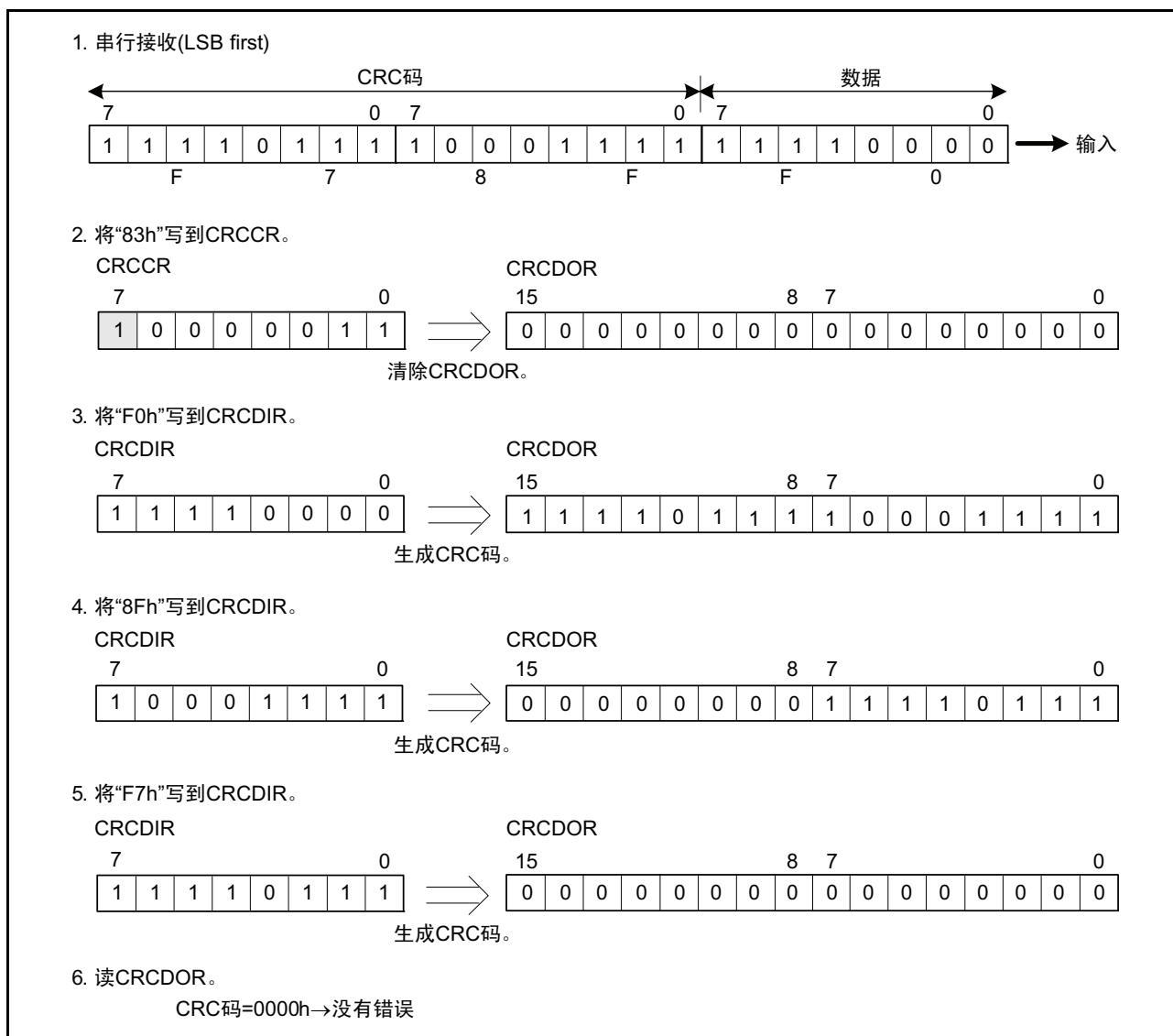


图 30.4 LSB first 的数据接收

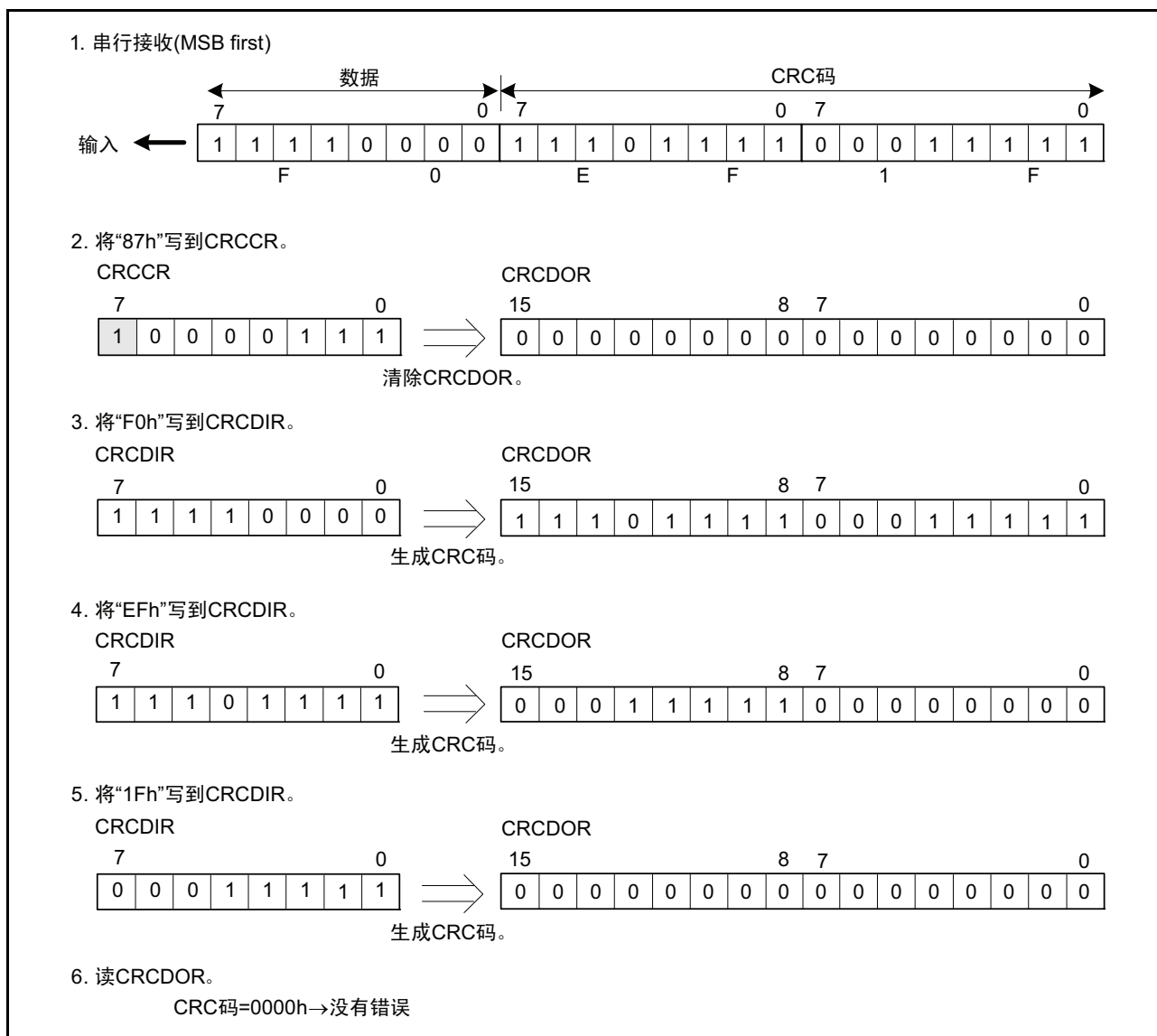


图 30.5 MSB first 的数据接收

30.4 使用时的注意事项

30.4.1 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 CRC 运算器的运行, 初始值为停止 CRC 运算器的运行。能通过解除模块停止状态, 使寄存器变为可存取的状态。详细内容请参照“11. 低功耗功能”。

30.5 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。

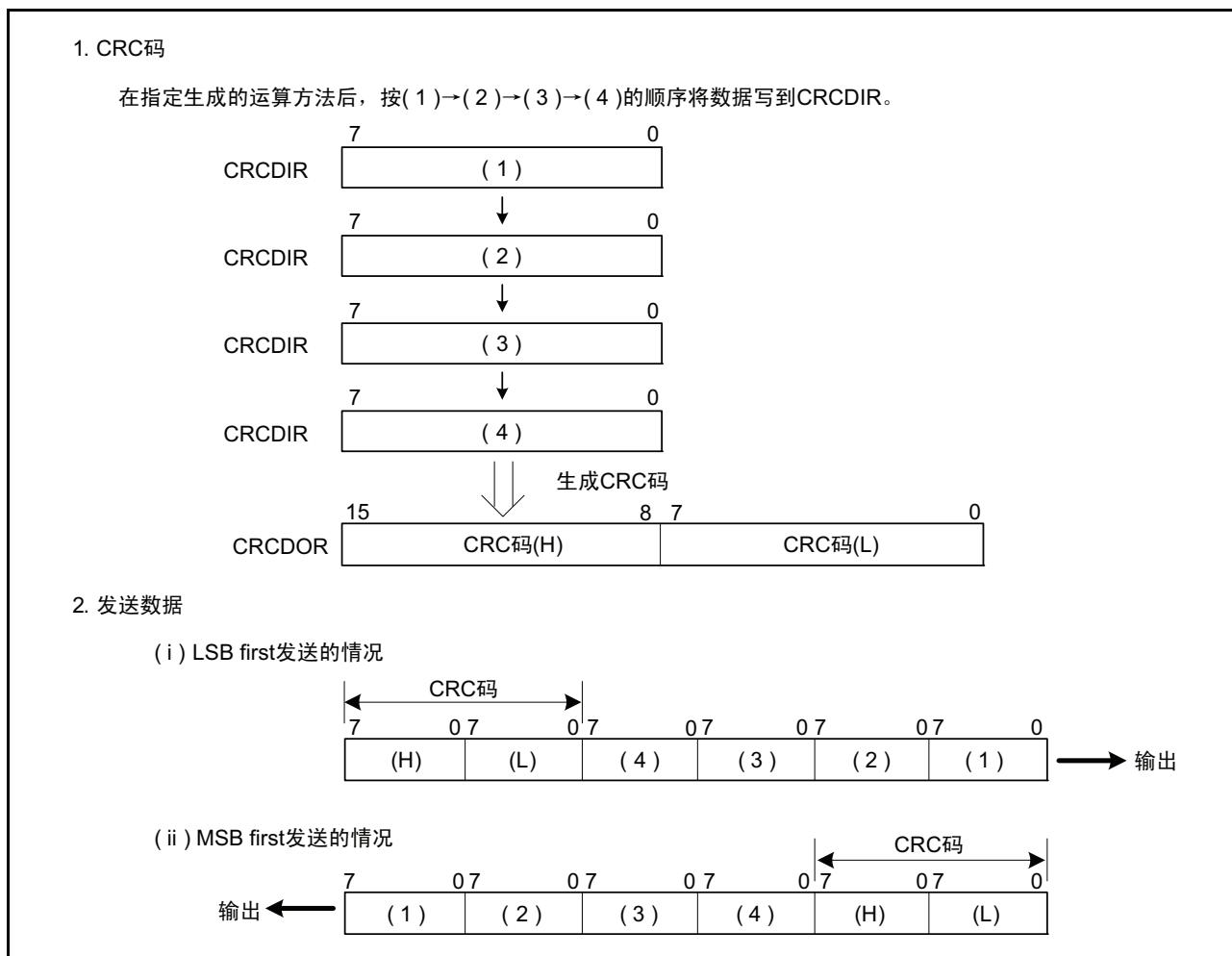


图 30.6 LSB first 和 MSB first 的发送数据

31. 12 位 A/D 转换器 (S12ADB)

31.1 概要

RX63T 群内置 1 个单元的逐次逼近方式的 12 位 A/D 转换器，能选择最多 8 个通道的模拟输入。

12 位 A/D 转换器通过逐次逼近方式将选择的最多 8 个通道的模拟输入转换为 12 位数字值。A/D 转换器的运行模式有 1 个周期扫描模式（按照从小到大的通道号顺序对任意选择的最多 8 个通道的模拟输入只进行 1 次转换）、连续扫描模式（按照从小到大的通道号顺序对任意选择的最多 8 个通道的模拟输入进行连续转换）和分组扫描模式（将任意选择的最多 8 个通道的模拟输入分为 2 组（组 A 和组 B），然后以组为单位按照从小到大的通道号顺序对所选通道的模拟输入进行转换）。

在分组扫描模式中，能通过分别选择组 A 和组 B 的扫描开始条件，在不同时序开始组 A 和组 B 的 A/D 转换。在设定为“组 A 的优先控制运行”时，除了上述运行外，还能在组 B 的 A/D 转换运行过程中接受组 A 的扫描，中止组 B 的 A/D 转换运行，并且优先开始组 A 的 A/D 转换运行。

在双触发模式中，通过 1 个周期扫描模式或者分组扫描模式（组 A）对任意选择的 1 个通道的模拟输入进行转换，并且将通过第 1 次 A/D 转换开始触发转换的数据和通过第 2 次 A/D 转换开始触发转换的数据分别保存到各自的寄存器（A/D 转换数据的双重化）。

在每次扫描的最开始执行 1 次自诊断，对在 12 位 A/D 转换器内部生成的 3 个电压值中的 1 个进行 A/D 转换。

12 位 A/D 转换器的规格、功能概要和框图分别如表 31.1、表 31.2 和图 31.1 所示。

表 31.1 12 位 A/D 转换器的规格 (1/2)

项目	内容
单元数	1 个单元
输入通道	最多 8 个通道
A/D 转换方式	逐次逼近方式
分辨率	12 位
转换时间	每个通道 1.0 μ s（A/D 转换时钟，以 ADCLK=50MHz 运行时）
A/D 转换时钟	能用以下的分频比设定外围模块时钟 PCLKB（注1）和 A/D 转换时钟 ADCLK（注1） PCLKB:ADCLK 分频比 =1:1、1:2、1:4、1:8 在时钟发生电路（CPG）中设定 ADCLK。
数据寄存器	<ul style="list-style-type: none"> • 用于模拟输入：8 个 • 用于双触发模式中的 A/D 转换数据双重化：1 个 • 用于双触发模式扩展运行时的 A/D 转换数据双重化：2 个 • 将 A/D 转换结果保存到 12 位 A/D 数据寄存器。 • 支持 A/D 转换结果的 8 位、10 位、12 位精度输出。 （能选择输出转换结果的右移 2 位或者右移 4 位） • 在加法运算模式中，将 A/D 转换结果的加法运算值以 14 位保存到 A/D 数据寄存器。 • 双触发模式（能在 1 个周期扫描模式和分组扫描模式中选择） • 将所选 1 个通道的模拟输入的第 1 次 A/D 转换数据保存到对象通道的数据寄存器，第 2 次 A/D 转换数据保存到双重化寄存器。 • 双触发模式扩展运行（只对特定的触发种类有效） 将所选 1 个通道的模拟输入的 A/D 转换数据保存到按触发种类准备的双重化寄存器。

表 31.1 12 位 A/D 转换器的规格 (2/2)

项目	内容
运行模式	<ul style="list-style-type: none"> • 1 个周期扫描模式： 对任意选择的最多 8 个通道的模拟输入只进行 1 次 A/D 转换。 • 连续扫描模式：对任意选择的最多 8 个通道的模拟输入重复进行 A/D 转换。 • 分组扫描模式 将最多 8 个通道的模拟输入分为组 A 和组 B，并且以组为单位对所选全部通道的模拟输入只进行 1 次 A/D 转换。 • 能通过分别选择组 A 和组 B 的转换开始条件，在不同的时序开始组 A 和组 B 的 A/D 转换。 • 分组扫描模式（选择组 A 优先控制时） 如果在组 B 的 A/D 转换运行过程中输入组 A 的触发，就中止组 B 的 A/D 转换运行，开始组 A 的 A/D 转换运行。 在组 A 的 A/D 转换运行结束后，重新开始组 B 的 A/D 转换运行（重新扫描）。
A/D 转换开始条件	<ul style="list-style-type: none"> • 软件触发 • 同步触发 多功能定时器脉冲单元（MTU3）、通用 PWM 定时器（GPT）的触发 • 异步触发 能通过外部触发 ADTRG0# 引脚开始 A/D 转换运行。
功能	<ul style="list-style-type: none"> • 采样 & 保持功能 • 通道专用采样 & 保持功能（3ch） • 采样状态数可变功能 • 12 位 A/D 转换器的自诊断功能 • A/D 转换值加法运算模式 • 放电功能 • 双触发模式（A/D 转换数据双重化功能） • 窗口比较器功能（3ch）
中断源	<ul style="list-style-type: none"> • 除双触发模式和分组扫描模式外，在 1 次扫描结束时产生扫描结束中断请求（S12ADI）。 • 在双触发模式中，在 2 次扫描结束时产生扫描结束中断请求（S12ADI）。 • 在分组扫描模式中，在组 A 的扫描结束时产生扫描结束中断请求（S12ADI）；在组 B 的扫描结束时产生组 B 专用的扫描结束中断请求（S12GBADI）。 • 在分组扫描模式的双触发模式中，在组 A 的 2 次扫描结束时产生扫描结束中断请求（S12ADI）；在组 B 的扫描结束时产生组 B 专用的扫描结束中断请求（S12GBADI）。 • 在比较器检测时产生中断请求（CMP0 ~ CMP2）。（也能用作 POE 源） • 能通过 S12ADI、S12GBADI 中断或者 CMP0 ~ CMP2 中断启动 DMA 控制器（DMAC）或者数据传送控制器（DTC）。
低功耗功能	<ul style="list-style-type: none"> • 能设定为模块停止状态（注 2）。

注 1. 外围模块时钟 PCLKB 为 SCKCR.PCKB[3:0] 位设定的频率，A/D 转换时钟 ADCLK 为 SCKCR.PCKD[3:0] 位设定的频率。

注 2. 详细内容请参照“11. 低功耗功能”。

表 31.2 12 位 A/D 转换器的功能概要 (1/2)

项目		功能	
模拟输入通道		AN000 ~ AN007	
A/D 转换 开始条件	软件	软件触发	能
	外部触发	触发输入引脚	ADTRG0#
	MTU3 的 触发	MTU0 的 TRGA 的比较匹配 / 输入捕捉	TRG0AN
		MTU1.TGRA 的比较匹配 / 输入捕捉	TRGA1N
		MTU2.TGRA 的比较匹配 / 输入捕捉	TRGA2N
		MTU3.TGRA 的比较匹配 / 输入捕捉	TRGA3N
		MTU4.TGRA 的比较匹配 / 输入捕捉 或者互补 PWM 模式中的 MTU4.TCNT 的下溢 (波谷)	TRGA4N
		MTU6.TGRA 的比较匹配 / 输入捕捉	TRGA6N
		MTU7.TGRA 的比较匹配 / 输入捕捉 或者互补 PWM 模式中的 MTU7.TCNT 的下溢 (波谷)	TRGA7N
		MTU0.TGRE 的比较匹配	TRG0N
		MTU4.TADCORA 和 MTU4.TCNT 的比较匹配	TRG4AN
		MTU4.TADCORB 和 MTU4.TCNT 的比较匹配	TRG4BN
		MTU4.TADCORA 和 MTU4.TCNT 的比较匹配 或者 MTU4.TADCORB 和 MTU4.TCNT 的比较匹配	TRG4AN 或者 TRG4BN
		MTU4.TADCORA 和 MTU4.TCNT 的比较匹配 与 MTU4.TADCORB 和 MTU4.TCNT 的比较匹配 (使用中断减少功能 2 时)	TRG4ABN
		MTU7.TADCORA 和 MTU7.TCNT 的比较匹配	TRG7AN
		MTU7.TADCORB 和 MTU7.TCNT 的比较匹配	TRG7BN
		MTU7.TADCORA 和 MTU7.TCNT 的比较匹配 或者 MTU7.TADCORB 和 MTU7.TCNT 的比较匹配	TRG7AN 或者 TRG7BN
		MTU7.TADCORA 和 MTU7.TCNT 的比较匹配 与 MTU7.TADCORB 和 MTU7.TCNT 的比较匹配 (使用中断减少功能 2 时)	TRG7ABN
		GPT 的触发	GPT0.GTADTRA 的比较匹配
	GPT0.GTADTRB 的比较匹配		GTADTRB0N
	GPT1.GTADTRA 的比较匹配		GTADTRA1N
	GPT1.GTADTRB 的比较匹配		GTADTRB1N
	GPT2.GTADTRA 的比较匹配		GTADTRA2N
	GPT2.GTADTRB 的比较匹配		GTADTRB2N
	GPT3.GTADTRA 的比较匹配		GTADTRA3N
	GPT3.GTADTRB 的比较匹配		GTADTRB3N
	GPT0.GTADTRA 的比较匹配或者 GPT0.GTADTRB 的比较匹配		GTADTRA0N 或者 GTADTRB0N
GPT1.GTADTRA 的比较匹配或者 GPT1.GTADTRB 的比较匹配	GTADTRA1N 或者 GTADTRB1N		
GPT2.GTADTRA 的比较匹配或者 GPT2.GTADTRB 的比较匹配	GTADTRA2N 或者 GTADTRB2N		
GPT3.GTADTRA 的比较匹配或者 GPT3.GTADTRB 的比较匹配	GTADTRA3N 或者 GTADTRB3N		
通道专用独立采样 & 保持功能	对象通道	AN000 ~ AN002	

表 31.2 12 位 A/D 转换器的功能概要 (2/2)

项目		功能	
窗口比较器	对象通道	AN000 ~ AN002	
	基准电压的 设定基准	由外部引脚指定	CVREFL: AN003、 CVREFH: AN007
		内部生成	1/8AVCC0、2/8AVCC0、 3/8AVCC0、4/8AVCC0、 5/8AVCC0、6/8AVCC0、 7/8AVCC0
噪声消除功能		通过 PCLK、PCLK/2、PCLK/4、 PCLK/8、PCLK/16、PCLK/128 对比较器检测结果进行 16 次采样	
中断		S12ADI、S12GBADI、 CMP0 ~ CMP2	
模块停止功能的设定 (注 1)		MSTPCRA.MSTPA17 位	

注 1. 详细内容请参照“11. 低功耗功能”。

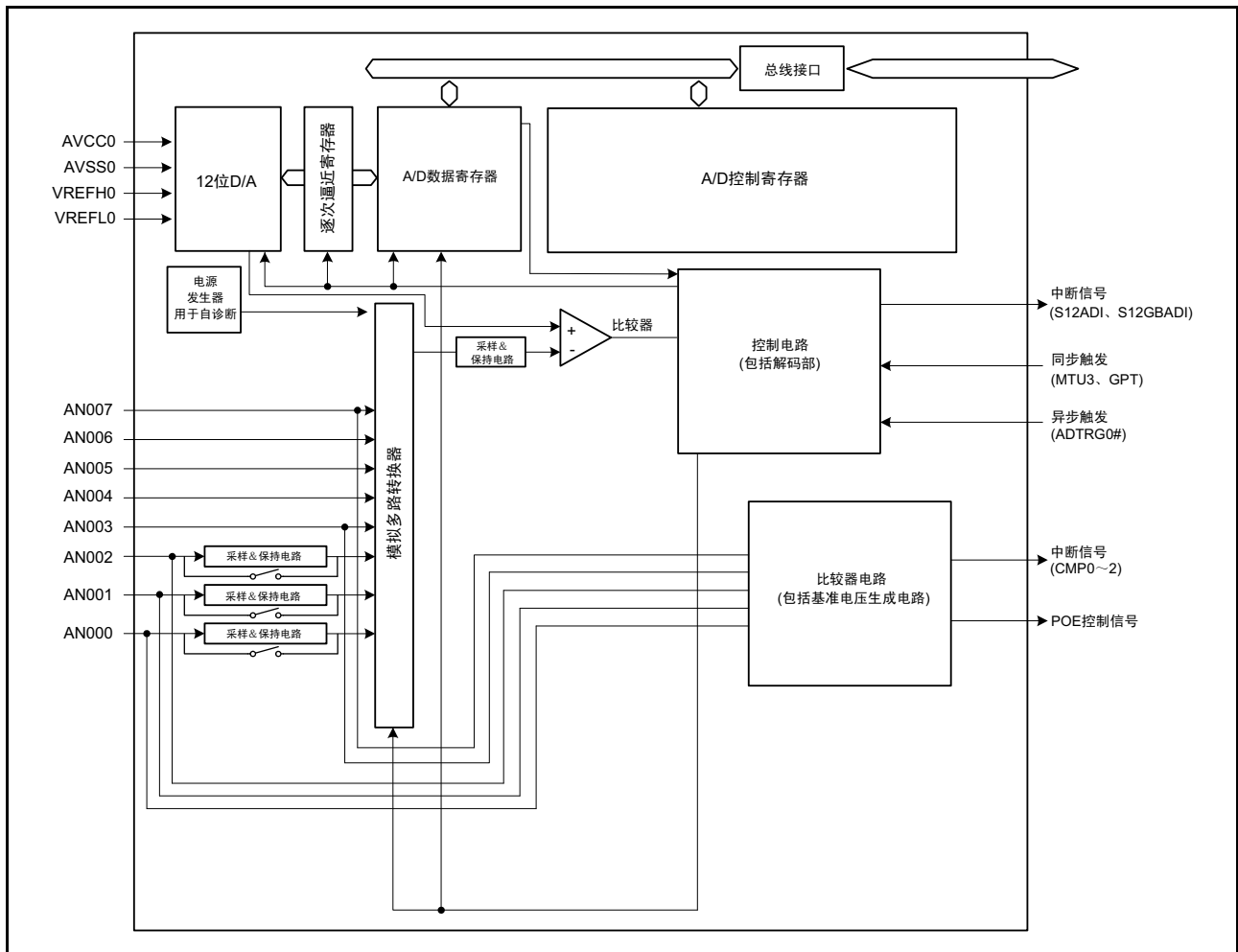


图 31.1 12 位 A/D 转换器的框图

12 位 A/D 转换器使用的输入引脚如表 31.3 所示。

表 31.3 12 位 A/D 转换器的输入引脚

引脚名	输入	功能
AVCC0	输入	模拟部的电源引脚
AVSS0	输入	模拟部的接地引脚
VREFH0	输入	基准电源引脚
VREFL0	输入	基准电源接地引脚
AN000	输入	模拟输入引脚 0
AN001	输入	模拟输入引脚 1
AN002	输入	模拟输入引脚 2
AN003/CVREFL	输入	模拟输入引脚 3/ 比较器 Low 电平侧的基准电压引脚 (在比较器运行并且选择外加外部引脚的基准电压时, 为比较器 Low 电平侧的基准电压引脚。)
AN004	输入	模拟输入引脚 4
AN005	输入	模拟输入引脚 5
AN006	输入	模拟输入引脚 6
AN007/CVREFH	输入	模拟输入引脚 7/ 比较器 High 电平侧的基准电压引脚 (在比较器运行并且选择外加外部引脚的基准电压时, 为比较器 High 电平侧的基准电压引脚。)
ADTRG0#	输入	用于 A/D 转换开始的外部触发输入引脚

31.2 寄存器说明

31.2.1 A/D 数据寄存器 y (ADDRy) (y=0 ~ 7)、A/D 数据双重化寄存器 (ADBLDR)、A/D 数据双重化寄存器 A (ADBLDRA)、A/D 数据双重化寄存器 B (ADBLDRB)

ADDRy 寄存器是 16 位只读寄存器，用于保存 A/D 转换结果。ADBLDR 寄存器是 16 位只读寄存器，用于保存通过双触发模式的第 2 次触发进行 A/D 转换后的 A/D 转换结果。ADBLDRA 寄存器和 ADBLDRB 寄存器是 16 位只读寄存器，用于保存通过双触发模式的扩展运行的触发进行 A/D 转换后的 A/D 转换结果。

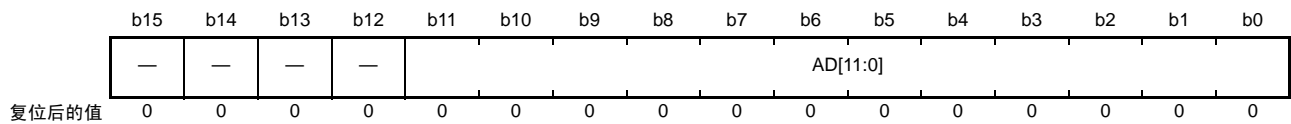
ADDRy、ADBLDR、ADBLDRA、ADBLDRB 寄存器的格式因以下条件而不同。

- A/D 数据寄存器格式选择位的设定值 (往右靠紧或者往左靠紧)
- A/D 数据寄存器位精度指定位的设定值 (12 位、10 位或者 8 位)
- A/D 转换值加法运算模式选择寄存器的设定值 (选择或者不选择 A/D 转换值加法运算模式)

各条件的格式如下所示。

- 设定为 12 位精度并且往右靠紧的格式时

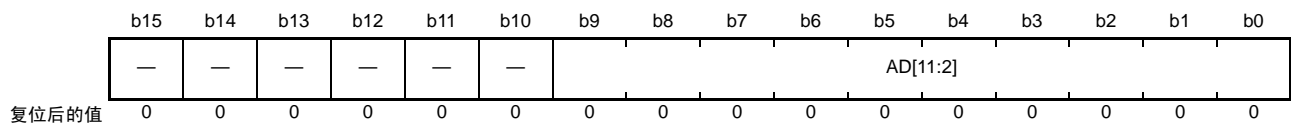
地址 ADDR0 0008 9020h、ADDR1 0008 9022h、ADDR2 0008 9024h、ADDR3 0008 9026h、ADDR4 0008 9028h、ADDR5 0008 902Ah、ADDR6 0008 902Ch、ADDR7 0008 902Eh、ADBLDR 0008 9018h、ADBLDRA 0008 9084h、ADBLDRB 0008 9086h



位	符号	位名	功能	R/W
b11-b0	AD[11:0]	转换值 11 ~ 0	12 位 A/D 转换值	R
b15-b12	—	保留位	读写值都为“0”。	R/W

- 设定为 10 位精度并且往右靠紧的格式时

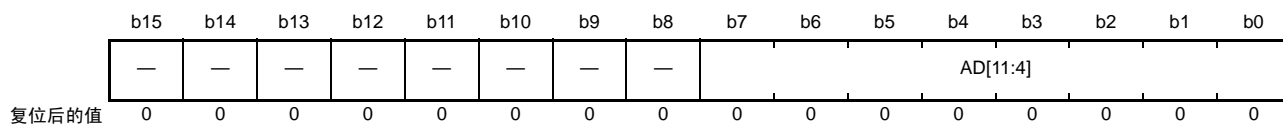
地址 ADDR0 0008 9020h、ADDR1 0008 9022h、ADDR2 0008 9024h、ADDR3 0008 9026h、ADDR4 0008 9028h、ADDR5 0008 902Ah、ADDR6 0008 902Ch、ADDR7 0008 902Eh、ADBLDR 0008 9018h、ADBLDRA 0008 9084h、ADBLDRB 0008 9086h



位	符号	位名	功能	R/W
b9-b0	AD[11:2]	转换值 11 ~ 2	12 位 A/D 转换值的高 10 位	R
b15-b10	—	保留位	读写值都为“0”。	R/W

- 设定为 8 位精度并且往右靠紧的格式时

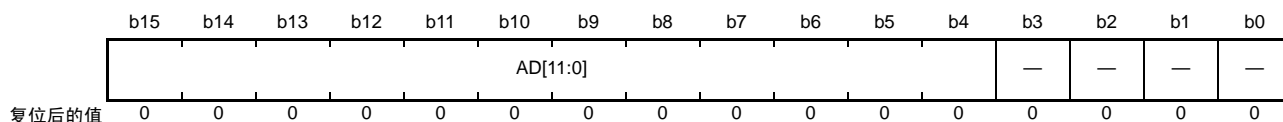
地址 ADDR0 0008 9020h、ADDR1 0008 9022h、ADDR2 0008 9024h、ADDR3 0008 9026h、ADDR4 0008 9028h、ADDR5 0008 902Ah、
ADDR6 0008 902Ch、ADDR7 0008 902Eh、ADBLDR 0008 9018h、ADBLDRA 0008 9084h、ADBLDRB 0008 9086h



位	符号	位名	功能	R/W
b7-b0	AD[11:4]	转换值 11 ~ 4	12 位 A/D 转换值的高 8 位	R
b15-b8	—	保留位	读写值都为“0”。	R/W

- 设定为 12 位精度并且往左靠紧的格式时

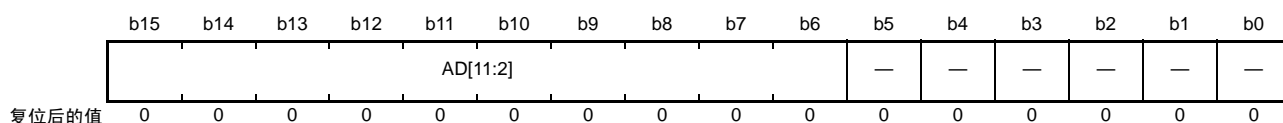
地址 ADDR0 0008 9020h、ADDR1 0008 9022h、ADDR2 0008 9024h、ADDR3 0008 9026h、ADDR4 0008 9028h、ADDR5 0008 902Ah、
ADDR6 0008 902Ch、ADDR7 0008 902Eh、ADBLDR 0008 9018h、ADBLDRA 0008 9084h、ADBLDRB 0008 9086h



位	符号	位名	功能	R/W
b3-b0	—	保留位	读写值都为“0”。	R/W
b15-b4	AD[11:0]	转换值 11 ~ 0	12 位 A/D 转换值	R

- 设定为 10 位精度并且往左靠紧的格式时

地址 ADDR0 0008 9020h、ADDR1 0008 9022h、ADDR2 0008 9024h、ADDR3 0008 9026h、ADDR4 0008 9028h、ADDR5 0008 902Ah、
ADDR6 0008 902Ch、ADDR7 0008 902Eh、ADBLDR 0008 9018h、ADBLDRA 0008 9084h、ADBLDRB 0008 9086h



位	符号	位名	功能	R/W
b5-b0	—	保留位	读写值都为“0”。	R/W
b15-b6	AD[11:2]	转换值 11 ~ 2	12 位 A/D 转换值的高 10 位	R

- 设定为 8 位精度并且往左靠紧的格式时

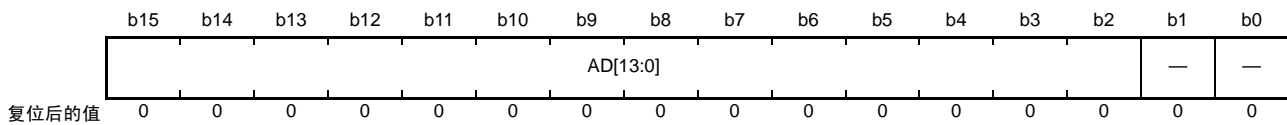
地址 ADDR0 0008 9020h、ADDR1 0008 9022h、ADDR2 0008 9024h、ADDR3 0008 9026h、ADDR4 0008 9028h、ADDR5 0008 902Ah、
ADDR6 0008 902Ch、ADDR7 0008 902Eh、ADDBLDR 0008 9018h、ADDBLDRA 0008 9084h、ADDBLDRB 0008 9086h



位	符号	位名	功能	R/W
b7-b0	—	保留位	读写值都为“0”。	R/W
b15-b8	AD[11:4]	转换值 11 ~ 4	12 位 A/D 转换值的高 8 位	R

- 选择 A/D 转换值加法运算模式时

地址 ADDR0 0008 9020h、ADDR1 0008 9022h、ADDR2 0008 9024h、ADDR3 0008 9026h、ADDR4 0008 9028h、ADDR5 0008 902Ah、
ADDR6 0008 902Ch、ADDR7 0008 902Eh、ADDBLDR 0008 9018h、ADDBLDRA 0008 9084h、ADDBLDRB 0008 9086h



位	符号	位名	功能	R/W
b1-b0	—	保留位	读写值都为“0”。	R/W
b15-b2	AD[13:0]	—	14 位 A/D 转换值的加法运算结果	R

如果设定为 A/D 转换值加法运算模式，AD[13:0] 位就表示对相同通道的 A/D 转换值进行加法运算后的值。而且 A/D 转换结果的加法运算值按照 14 位精度并且往左靠紧的格式保存到 A/D 数据寄存器，同时 ADCER.ADPRC[1:0] 位和 ADCER.ADRFMT 位的设定无效。

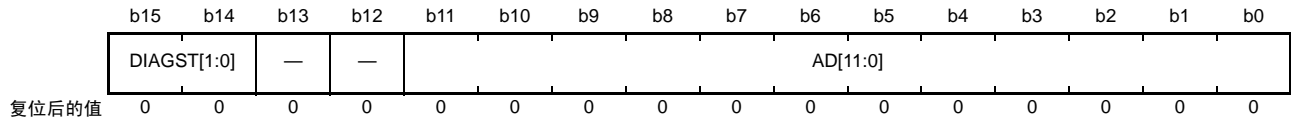
31.2.2 A/D 自诊断数据寄存器 (ADRD)

ADRD 寄存器是 16 位只读寄存器，用于保存通过 12 位 A/D 转换器的自诊断进行 A/D 转换后的结果。

根据 A/D 数据寄存器格式选择位 (ADCER.ADRFMT) 的设定，ADRD 寄存器的格式如下所示。ADRD 寄存器不能设定为 A/D 转换加法运算模式。

- ADCER.ADRFMT=0 (设定为往右靠紧的格式)

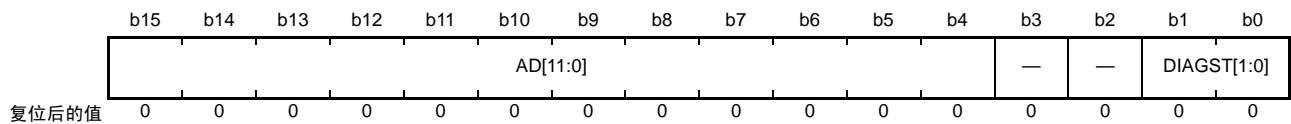
地址 0008 901Eh



位	符号	位名	功能	R/W
b11-b0	AD[11:0]	—	12 位 A/D 转换值	R
b13-b12	—	保留位	读写值都为“0”。	R/W
b15-b14	DIAGST[1:0]	自诊断状态位	^{b15 b14} 0 0: 表示自通电后还未执行 1 次自诊断 0 1: 表示执行了电压值为 0V 的自诊断 1 0: 表示执行了电压值为 VREFH0×1/2 的自诊断 1 1: 表示执行了电压值为 VREFH0 的自诊断 有关自诊断的详细内容，请参照“31.2.8 A/D 控制扩展寄存器 (ADCER)”。	R

- ADCER.ADRFMT=1 (设定为往左靠紧的格式)

地址 0008 901Eh



位	符号	位名	功能	R/W
b1-b0	DIAGST[1:0]	自诊断状态位	^{b1 b0} 0 0: 表示自通电后还未执行 1 次自诊断 0 1: 表示执行了电压值为 0V 的自诊断 1 0: 表示执行了电压值为 VREFH0×1/2 的自诊断 1 1: 表示执行了电压值为 VREFH0 的自诊断 有关自诊断的详细内容，请参照“31.2.8 A/D 控制扩展寄存器 (ADCER)”。	R
b3-b2	—	保留位	读写值都为“0”。	R/W
b15-b4	AD[11:0]	—	12 位 A/D 转换值	R

31.2.3 A/D 控制寄存器 (ADCSR)

地址 0008 9000h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADST	ADCS[1:0]	ADIE	—	—	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]					
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b4-b0	DBLANS [4:0]	双触发对象通道选择位	从 8 个通道中选择 1 个通道进行双触发对象的模拟输入。只在选择双触发模式时有效。	R/W
b5	—	保留位	读写值都为“0”。	R/W
b6	GBADIE	组 B 扫描结束中断允许位	0: 禁止在组 B 的扫描结束后发生 S12GBADI 中断 1: 允许在组 B 的扫描结束后发生 S12GBADI 中断	R/W
b7	DBLE	双触发模式选择位	0: 不选择双触发模式 1: 选择双触发模式	R/W
b8	EXTRG	触发选择位 (注 1)	0: 选择通过同步触发 (MTU3、GPT) 开始 A/D 转换 1: 选择通过异步触发 (ADTRG0#) 开始 A/D 转换	R/W
b9	TRGE	触发启动允许位	0: 禁止通过同步触发、异步触发开始 A/D 转换 1: 允许通过同步触发、异步触发开始 A/D 转换	R/W
b10-b11	—	保留位	读写值都为“0”。	R/W
b12	ADIE	扫描结束中断允许位	0: 禁止在扫描结束后发生 S12ADI 中断 1: 允许在扫描结束后发生 S12ADI 中断	R/W
b14-b13	ADCS[1:0]	扫描模式选择位	^{b14 b13} 0 0: 1 个周期扫描模式 0 1: 分组扫描模式 1 0: 连续扫描模式 1 1: 禁止设定	R/W
b15	ADST	A/D 转换开始位	0: 停止 A/D 转换 1: 开始 A/D 转换	R/W

注 1. 通过外部引脚 (异步触发) 开始 A/D 转换的方法:

在给外部引脚 (ADTRG0#) 输入 High 电平的状态下, 将 ADCSR.TRGE 位置“1”并且将 ADCSR.EXTRG 位置“1”。此后, 如果使 ADTRG0# 的信号变为 Low 电平, 就在检测到 ADTRG0# 的下降沿时开始扫描转换。此时, Low 电平输入的脉宽至少需要 1.5 个 PCLK 时钟。

ADCSR 寄存器是设定双触发模式和 A/D 转换开始触发、允许或者禁止扫描结束中断、选择扫描模式以及开始或者停止 A/D 转换的寄存器。

DBLANS[4:0] 位 (双触发对象通道选择位)

这些位选择在双触发模式中进行 A/D 转换数据双重化的 1 个通道。将通过双触发模式的第一次 A/D 转换开始触发对 DBLANS[4:0] 位选择的通道的模拟输入进行 A/D 转换后的结果保存到 A/D 数据寄存器 y, 将通过第二次 A/D 转换开始触发进行 A/D 转换后的结果保存到 A/D 数据双重化寄存器。双触发对象通道的选择表如表 31.4 所示。在双触发模式的加法运算模式中, 能通过 ADADS 寄存器选择 DBLANS[4:0] 位选择的通道。如果选择双触发模式, ADANSA 寄存器选择的通道就变为无效, 而 DBLANS[4:0] 位选择的 1 个通道变为进行 A/D 转换的通道。必须在 ADST 位为“0”时设定 DBLANS[4:0] 位 (不能在给 ADST 位写“1”的同时进行设定)。

表 31.4 DBLANS[4:0] 位的设定值与双触发对象通道的关系

DBLANS[4:0]	双触发对象通道
00000	AN000
00001	AN001
00010	AN002
00011	AN003
00100	AN004
00101	AN005
00110	AN006
00111	AN007

GBADIE 位 (组 B 扫描结束中断允许位)

此位允许或者禁止发生分组扫描模式中组 B 的扫描结束中断 (S12GBADI)。

DBLE 位 (双触发模式选择位)

双触发模式以 ADSTRGR.TRSA[4:0] 位选择的 MTU3 触发或者 GPT 触发开始扫描为条件进行以下两个运行。

1. 如果将 ADIE 位置“1”，在第 1 次扫描结束时就不输出扫描结束中断，而在第 2 次扫描结束时输出扫描结束中断。
2. 将 DBLANS[4:0] 位选择的模拟输入的第 1 次 A/D 转换数据保存到 A/D 数据寄存器 y，第 2 次 A/D 转换数据保存到 A/D 数据双重化寄存器。

如果将 DBLE 位置“1”，ADANSA 寄存器选择的通道就变为无效。在连续扫描模式中不能选择双触发模式。如果选择双触发模式，就不能使用软件触发。必须在 ADST 位为“0”时设定 DBLE 位 (不能在给 ADST 位写“1”的同时进行设定)。

EXTRG 位 (触发选择位)

此位选择是同步触发还是异步异步触发作为 A/D 转换开始触发。

TRGE 位 (触发启动允许位)

此位允许或者禁止通过同步触发或者异步触发开始 A/D 转换。
在分组扫描模式中，必须将此位置“1”。

ADIE 位 (扫描结束中断允许位)

除分组扫描模式中的组 B 外，此位允许或者禁止发生 A/D 扫描转换结束中断 (S12ADI)。

在设定为不选择双触发模式的情况下，如果在 1 次扫描结束时 ADIE 位为“1”，就发生 S12ADI 中断。

在选择双触发模式的情况下，只在通过 ADSTRGR.TRSA[4:0] 位选择的 MTU3 触发或者 GPT 触发开始扫描时，如果在第 2 次扫描结束时 ADIE 位为“1”，就发生 S12ADI 中断。

ADCS[1:0] 位 (扫描模式选择位)

这些位选择扫描转换模式。

在 1 个周期扫描模式中, 按照从小到大的通道号顺序对 ADANSA 寄存器选择的最多 8 个通道的模拟输入进行 A/D 转换, 如果所选通道的转换全部结束, 就停止扫描转换。

在连续扫描模式中, 在 ADCSR.ADST 位为“1”的期间, 按照从小到大的通道号顺序对 ADANSA 寄存器选择的最多 8 个通道的模拟输入进行 A/D 转换, 如果所选通道的转换全部结束, 就返回到最初的通道继续进行 A/D 转换。如果将 ADCSR.ADST 位置“0”, 就停止 A/D 转换。

在分组扫描模式中, 将 ADSTRGR.TRSA[4:0] 位选择的 MUT3 触发或者 GPT 触发作为 A/D 转换开始条件, 按照从小到大的通道号顺序对 ADANSA 寄存器选择的最多 8 个通道的模拟输入 (组 A) 进行 A/D 转换, 如果所选通道的 A/D 转换全部结束, 就停止 A/D 转换; 同样, 将 ADSTRGR.TRSA[4:0] 位选择的 MUT3 触发或者 GPT 触发作为 A/D 转换开始条件, 按照从小到大的通道号顺序对 ADANSB 寄存器选择的最多 8 个通道的模拟输入 (组 B) 进行 A/D 转换, 如果所选通道的 A/D 转换全部结束, 就停止 A/D 转换。在选择分组扫描模式时, 必须给组 A 和组 B 选择不同的通道和不同的触发。

必须在 ADST 位为“0”时设定 ADCS 位 (不能在给 ADST 位写“1”的同时进行设定)。

ADST 位 (A/D 转换开始位)

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、转换模式和转换对象模拟输入。

[为“1”的条件]

- 当通过软件写“1”时
- 将 ADCSR.EXTRG 位置“0”, ADCSR.TRGE 位置“1”并且检测到 ADSTRGR.TRSA[4:0] 位选择的同步触发 (MTU3、GPT) 时
- 在分组扫描模式中将 ADCSR.TRGE 位置“1”并且检测到 ADSTRGR.TRSA[4:0] 位选择的同步触发 (MTU3、GPT) 时
- 将 ADCSR.TRGE 位和 ADCSR.EXTRG 位置“1”, ADSTRGR.TRSA[4:0] 位置“00000b”并且检测到异步触发时
- 在组 A 优先控制运行模式有效 (ADCSR.ADCS[1:0] 位=01b 并且 ADGSPCR.PGS=1) 时, 检测到组 B 的触发并且开始组 B 的 A/D 转换时
- 在组 A 优先控制运行模式有效 (ADCSR.ADCS[1:0] 位=01b 并且 ADGSPCR.PGS=1) 时, 将 ADGSPCR.GBRSCN 位置“1”并且重新开始组 B 的 A/D 转换时
- 在组 A 优先控制运行模式有效 (ADCSR.ADCS[1:0] 位=01b 并且 ADGSPCR.PGS=1) 时, 将 ADGSPCR.GBRP 位置“1”并且重新开始组 B 的 A/D 转换时

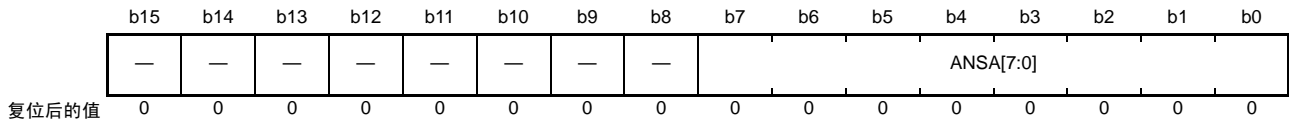
[为“0”的条件]

- 当通过软件写“0”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
- 在分组扫描模式中组 A 的扫描结束时
- 在分组扫描模式中组 B 的扫描结束时
- 在组 A 优先控制运行模式有效 (ADCSR.ADCS[1:0] 位=01b 并且 ADGSPCR.PGS=1) 时, 就在组 B 的 A/D 转换过程中检测到组 A 的触发并且中止组 B 的扫描时
- 在组 A 优先控制运行模式有效 (ADCSR.ADCS[1:0] 位=01b 并且 ADGSPCR.PGS=1) 时, 将 ADGSPCR.GBRP 位置“1”并且通过组 B 的重新开始触发进行的扫描结束时
- 在组 A 优先控制运行模式有效 (ADCSR.ADCS[1:0] 位=01b 并且 ADGSPCR.PGS=1) 时, 将 ADGSPCR.GBRSCN 位置“1”并且通过组 B 的触发进行的扫描结束时

注. 在组 A 优先控制运行模式有效 (ADCSR.ADCS[1:0] 位=01b 并且 ADGSPCR.PGS=1) 时, 不能将 ADST 位置“1”。
在组 A 优先控制运行模式有效 (ADCSR.ADCS[1:0] 位=01b 并且 ADGSPCR.PGS=1) 并且 ADGSPCR.GBRP=1 时, 不能将 ADST 位置“0”。如果要强制停止 A/D 转换, 就必须按照 ADST 位的清除步骤进行。

31.2.4 A/D 通道选择寄存器 A (ADANSA)

地址 0008 9004h



位	符号	位名	功能	R/W
b7-b0	ANSA[7:0]	A/D 转换通道选择位	0: 不将 AN000 ~ AN007 作为转换对象 1: 将 AN000 ~ AN007 作为转换对象	R/W
b15-b8	—	保留位	读写值都为“0”。	R/W

ADANSA 寄存器是选择 A/D 转换通道的模拟输入 AN000 ~ AN007 的寄存器。在分组扫描模式中，选择组 A 的通道。

ANSA[7:0] 位 (A/D 转换通道选择位)

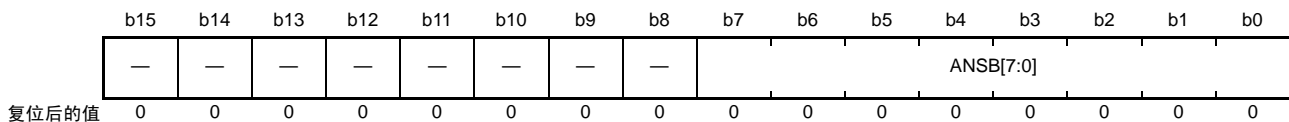
这些位选择 A/D 转换通道的模拟输入 AN000 ~ AN007，能任意设定要选择的通道和通道数。ANSA[0] 位对应 AN000，ANSA[7] 位对应 AN007。

在选择双触发模式时，ADCSR.DBLANS[4:0] 位选择的 1 个通道为组 A 的通道，ANSA[7:0] 位的设定无效。

必须在 ADCSR.ADST 位为“0”时设定 ANSA[7:0] 位。

31.2.5 A/D 通道选择寄存器 B (ADANSB)

地址 0008 9014h



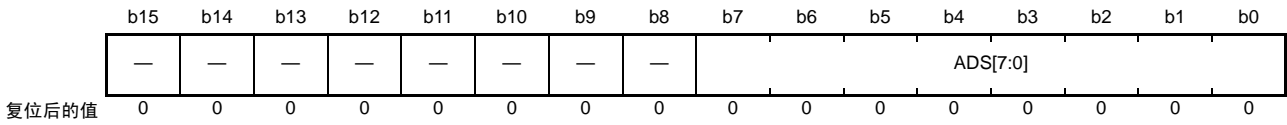
位	符号	位名	功能	R/W
b7-b0	ANSB[7:0]	A/D 转换通道选择位	0: 不将 AN000 ~ AN007 作为转换对象 1: 将 AN000 ~ AN007 作为转换对象	R/W
b15-b8	—	保留位	读写值都为“0”。	R/W

ADANSB 寄存器选择分组模式中进行组 B 的 A/D 转换的通道 AN000 ~ AN007。ADANSB 寄存器不能在其他的扫描模式中使用。从 ADANSA 寄存器或者双触发模式中的 ADCSR.DBLANS[4:0] 位选择的组 A 的通道以外的通道中，设定要选择的通道和通道数。ANSB[0] 位对应 AN000，ANSB[7] 位对应 AN007。

必须在 ADST 位为“0”时设定 ANSB[7:0] 位。

31.2.6 A/D 转换值加法运算模式选择寄存器 (ADADS)

地址 0008 9008h



位	符号	位名	功能	R/W
b7-b0	ADS[7:0]	A/D 转换值加法运算通道选择位	0: 不选择 AN000 ~ AN007 的 A/D 转换值加法运算模式 1: 选择 AN000 ~ AN007 的 A/D 转换值加法运算模式	R/W
b15-b8	—	保留位	读写值都为“0”。	R/W

ADADS 寄存器选择连续进行 2 ~ 4 次 A/D 转换后进行加法运算 (累加运算) 的 A/D 转换通道 0 ~ 7。

ADS[7:0] 位 (A/D 转换值加法运算通道选择位)

如果将通道号与 ADANSA.ANSA[n] 位 (n=0 ~ 7) 或者 ADCSR.DBLANS[4:0] 位和 ADANSB.ANSB[n] 位 (n=0 ~ 7) 选择的 A/D 转换通道相同的 ADS[n] 位置“1”，就对所选通道的模拟输入连续进行 ADADC.ADC[1:0] 位所设次数 (2 ~ 4 次) 的 A/D 转换，并且将加法运算 (累加运算) 值返回到 A/D 数据寄存器。通常，对不选择加法运算模式的 A/D 转换通道进行 1 次转换，然后将值返回到 A/D 数据寄存器。

必须在 ADCSR.ADST 位为“0”时设定 ADS[7:0] 位。

将 ADS[2] 位和 ADS[6] 位置“1”后的扫描运行顺序如图 31.2 所示。

在连续扫描模式 (ADCSR.ADCS[1:0]=10b) 中，设定 3 次 (ADADC.ADC[1:0]=11b) 加法运算，并且将 AN000 ~ AN007 作为被选择的对象 (ADANS0.ANSA[7:0]=FFh)。从 AN000 开始转换，对 AN002 连续进行 4 次转换，然后将加法运算 (累加运算) 值返回到 A/D 数据寄存器 2。此后，开始 AN003 的转换，对 AN006 连续进行 4 次转换，然后将加法运算 (累加运算) 值返回到 A/D 数据寄存器 6。在转换 AN007 后，按照同样的顺序重新从 AN000 开始转换。

不选择加法运算模式的通道的 A/D 数据寄存器格式取决于 ADCER.ADRFMT 位 (往右靠紧 / 往左靠紧) 的设定。

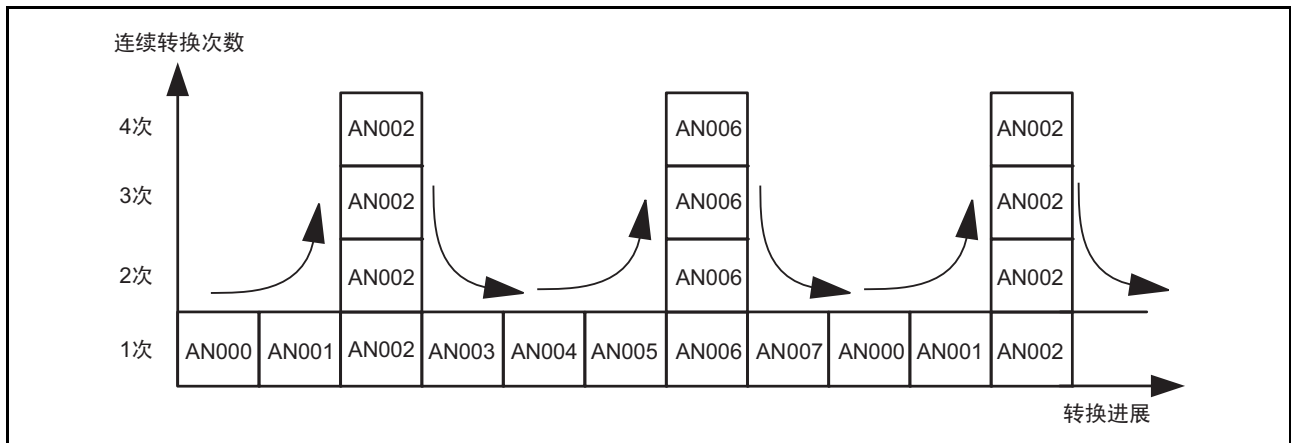
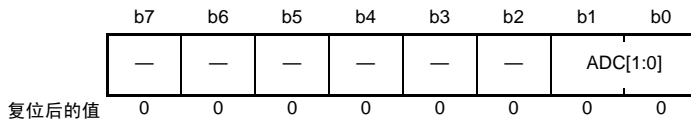


图 31.2 选择 ADADC.ADC[1:0] 为“11b”并且 ADS[2] 和 ADS[6] 都为“1”时的扫描转换顺序

31.2.7 A/D 转换值加法运算次数选择寄存器 (ADADC)

地址 0008 900Ch



位	符号	位名	功能	R/W
b1-b0	ADC[1:0]	加法运算次数选择位	b1 b0 0 0: 转换 1 次 (没有加法运算, 和一般转换一样) 0 1: 转换 2 次 (进行 1 次加法运算) 1 0: 转换 3 次 (进行 2 次加法运算) 1 1: 转换 4 次 (进行 3 次加法运算)	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

ADADC 寄存器给选择 A/D 转换值加法运算模式的通道的 A/D 转换设定加法运算次数。

ADC[1:0] 位 (加法运算次数选择位)

给选择 A/D 转换 (包括在双触发模式中, 通过 ADCSR.DBLANS[4:0] 位选择的通道的 A/D 转换) 和加法运算模式的通道的 A/D 转换设定共同的加法运算次数。

必须在 ADCSR.ADST 位为“0”时设定 ADC[1:0] 位。

31.2.8 A/D 控制扩展寄存器 (ADCER)

地址 0008 900Eh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]	—	—	ACE	DCE	—	ADPRC[1:0]	—	—	—
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	—	保留位	读写值都为“0”。	R/W
b2-b1	ADPRC[1:0]	A/D 数据寄存器的位精度指定位	b2 b1 0 0: 以 12 位精度保存到 A/D 数据寄存器 0 1: 以 10 位精度保存到 A/D 数据寄存器 1 0: 以 8 位精度保存到 A/D 数据寄存器 1 1: 不能设定	R/W
b3	—	保留位	读写值都为“0”。	R/W
b4	DCE	放电允许位	0: 不实施 A/D 转换结束后的放电运行 1: 实施 A/D 转换结束后的放电运行	R/W
b5	ACE	自动清除允许位	0: 禁止自动清除 1: 允许自动清除	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W
b9-b8	DIAGVAL [1:0]	自诊断转换电压选择位	b9 b8 0 0: 在自诊断有效时禁止设定 0 1: 使用 0V 电压进行自诊断 1 0: 使用 VREFH0×1/2 的电压进行自诊断 1 1: 使用 VREFH0 电压进行自诊断	R/W
b10	DIAGLD	自诊断模式选择位	0: 自诊断电压轮流模式 1: 自诊断电压固定模式	R/W
b11	DIAGM	自诊断允许位	0: 不执行 12 位 A/D 转换器的自诊断 1: 执行 12 位 A/D 转换器的自诊断	R/W
b14-b12	—	保留位	读写值都为“0”。	R/W
b15	ADRFMT	A/D 数据寄存器格式选择位	0: 将 A/D 数据寄存器的格式设定为往右靠紧 1: 将 A/D 数据寄存器的格式设定为往左靠紧	R/W

ADCER 寄存器设定自诊断模式、A/D 数据寄存器 y (ADDRy) 的格式以及自动清除。

ADPRC 位 (A/D 数据寄存器的位精度指定位)

此位选择是以 8 位精度、10 位精度还是 12 位精度将 A/D 转换结果保存到 ADDRy、ADDBLDR、ADDBLDRA、ADDBLDRB 寄存器。

DCE 位 (放电允许位)

此位选择是否在 A/D 转换结束后进行对象模拟引脚的放电。如果因故障造成 A/D 转换器的输入引脚开路, 就通过实施数次放电使转换结果为“0000h”的近似值, 并且能检测到引脚开路。

ACE 位 (自动清除允许位)

此位选择是否通过 CPU、DTC 或者 DMAC 读 ADDRy、ADRD、ADDBLDR、ADDBLDRA、ADDBLDRB 寄存器后自动清除对应的寄存器 (All“0”)。能通过自动清除检测到 ADDRy、ADRD、ADDBLDR、ADDBLDRA、ADDBLDRB 寄存器未更新的故障。

DIAGVAL[1:0] 位 (自诊断转换电压选择位)

详细内容请参照 ADCER.DIAGLD 位的说明。

在 ADCER.DIAGVAL[1:0] 位为“00b”的状态下将 ADCER.DIAGLD 位置“1”，就不执行自诊断。

DIAGLD 位 (自诊断模式选择位)

此位选择是对自诊断转换的 3 个电压值进行轮流转换，还是将电压值固定。

如果将 ADCER.DIAGLD 位置“0”，就按 $0V \rightarrow VREFH0 \times 1/2 \rightarrow VREFH0$ 的顺序轮流转换。复位后，如果选择自诊断电压轮流模式，就从 $0V$ 开始自诊断；如果选择自诊断电压固定模式，就固定为 ADCER.DIAGVAL[1:0] 位选择的电压进行转换。在自诊断电压轮流模式中，即使扫描转换结束，也不返回 $0V$ 。如果再次进行扫描转换，就接着上一次进行轮流转换。从自诊断电压固定模式转换到自诊断电压轮流模式时，从固定的电压值开始轮流转换。

必须在 ADST 位为“0”时设定 DIAGLD 位。

DIAGM 位 (自诊断允许位)

此位选择是否执行自诊断。

自诊断是检测 12 位 A/D 转换器故障的功能。对在内部生成的 $0V$ 、 $VREFH0 \times 1/2$ 和 $VREFH0$ 这 3 个电压值中的任意一个电压值进行转换。如果转换结束，就将转换的电压信息和转换值保存到自诊断数据寄存器 (ADRD)。之后，通过软件读 ADRD 寄存器，由此判断转换值是 (正常) 否 (异常) 在正常的范围内。在每次扫描的最开始执行 1 次自诊断，对 3 个电压值中的 1 个电压值进行 A/D 转换。自诊断的执行时间和 1 个通道的 A/D 转换时间相同。如果在分组扫描模式中选择自诊断，就分别执行组 A 和组 B 的自诊断。

必须在 ADST 位为“0”时设定 DIAGM 位。

ADRFMT 位 (A/D 数据寄存器格式选择位)

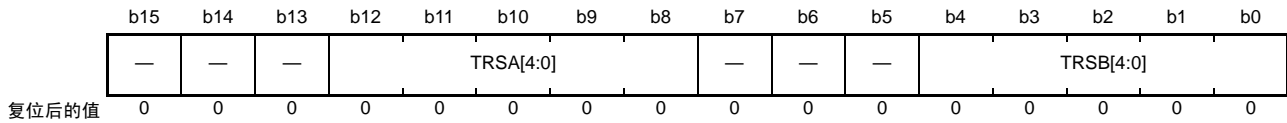
此位选择 ADDRy、ADRD、ADDBLDR、ADDBLDRA、ADDBLDRB 寄存器中保存的数据是往右靠紧还是往左靠紧。

在选择 A/D 转换值加法运算模式时，各数据寄存器的格式与 ADCER.ADRFMT 位的设定无关，固定为往左靠紧。

有关数据寄存器格式的详细内容，请参照“31.2.1 A/D 数据寄存器 y (ADDRy) (y=0 ~ 7)、A/D 数据双重化寄存器 (ADDBLDR)、A/D 数据双重化寄存器 A (ADDBLDRA)、A/D 数据双重化寄存器 B (ADDBLDRB)”、“31.2.2 A/D 自诊断数据寄存器 (ADRD)”。

31.2.9 A/D 开始触发选择寄存器 (ADSTRGR)

地址 0008 9010h



位	符号	位名	功能	R/W
b4-b0	TRSB[4:0]	组 B 专用 A/D 转换开始触发选择位	在分组扫描模式中，选择组 B 的 A/D 转换开始触发。	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W
b12-b8	TRSA[4:0]	A/D 转换开始触发选择位	在 1 个周期扫描模式和连续扫描模式中，选择 A/D 转换开始触发。在分组扫描模式中，选择组 A 的 A/D 转换开始触发。	R/W
b15-b13	—	保留位	读写值都为“0”。	R/W

ADSTRGR 寄存器是选择 A/D 转换开始触发的寄存器。

TRSB[4:0] 位 (组 B 专用 A/D 转换开始触发选择位)

此位选择组 B 所选的模拟输入的扫描开始触发。TRSB[4:0] 位只在分组扫描模式中需要设定，在其他扫描模式中不使用。组 B 的扫描开始触发禁止设定为软件触发和异步触发。因此，在分组扫描模式中，必须将 TRSB[4:0] 位设定为“00000”以外的值，将 ADCSR.TRGE 位设定为“1”。

在分组扫描模式的组 A 优先控制运行模式中，通过将 ADGSPCR.GBRP 位置“1”，能连续进行组 B 的 1 个周期扫描运行。在将 ADGSPCR.GBRP 位设定为“1”时，必须将 TRSB[4:0] 位设定为“1Fh”。另外，用于 A/D 转换的触发的发行间隔必须设定为不小于实际的扫描转换时间 (tSCAN)。如果发行间隔小于 tSCAN，就有可能使通过触发进行的 A/D 转换无效。

通过 TRSB[4:0] 位选择的 A/D 启动源一览表如表 31.5 所示。

TRSA[4:0] 位 (A/D 转换开始触发选择位)

这些位选择 1 个周期扫描模式和连续扫描模式中的 A/D 转换开始触发，还能选择分组扫描模式中组 A 所选模拟输入的扫描开始触发。在分组扫描模式或者双触发模式中进行扫描时，不能使用软件触发和异步触发。

- 在使用同步触发 (MTU3、GPT) 的 A/D 转换启动源时，必须将 ADCSR.TRGE 位置“1”并且将 ADCSR.EXTRG 位置“0”。
- 在使用异步触发 (ADTRG0#) 时，必须将 ADCSR.TRGE 位置“1”并且将 ADCSR.EXTRG 位置“1”。
- 软件触发 (ADCSR.ADST) 与 ADCSR.TRGE 位、ADCSR.EXTRG 位和 TRSA[4:0] 位的设定值无关，总是有效。

另外，用于 A/D 转换的触发的发行间隔必须设定为不小于实际的扫描转换时间 (tSCAN)。如果发行间隔小于 tSCAN，就有可能使通过触发进行的 A/D 转换无效。

通过 TRSA[4:0] 位选择的 A/D 启动源一览表如表 31.6 所示。

表 31.5 通过 TRSB[4:0] 位选择的 A/D 启动源一览表

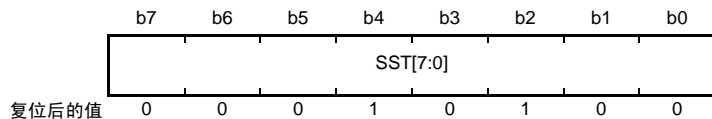
模块	启动源	备注	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
不选择触发源的状态			1	1	1	1	1
MTU3	TRGA0N	MTU0.TGRA 的比较匹配 / 输入捕捉	0	0	0	0	1
	TRGA1N	MTU1.TGRA 的比较匹配 / 输入捕捉	0	0	0	1	0
	TRGA2N	MTU2.TGRA 的比较匹配 / 输入捕捉	0	0	0	1	1
	TRGA3N	MTU3.TGRA 的比较匹配 / 输入捕捉	0	0	1	0	0
	TRGA4N	MTU4.TGRA 的比较匹配 / 输入捕捉或者 互补 PWM 模式中的 MTU4.TCNT 的下溢 (波谷)	0	0	1	0	1
	TRGA6N	MTU6.TGRA 的比较匹配 / 输入捕捉	0	0	1	1	0
	TRGA7N	MTU7.TGRA 的比较匹配 / 输入捕捉或者 互补 PWM 模式中的 MTU7.TCNT 的下溢 (波谷)	0	0	1	1	1
	TRG0N	MTU0.TGRE 的比较匹配	0	1	0	0	0
	TRG4AN	MTU4.TADCORA 和 MTU4.TCNT 的比较匹配	0	1	0	0	1
	TRG4BN	MTU4.TADCORB 和 MTU4.TCNT 的比较匹配	0	1	0	1	0
	TRG4AN 或者 TRG4BN	MTU4.TADCORA 和 MTU4.TCNT 的比较匹配或者 MTU4.TADCORB 和 MTU4.TCNT 的比较匹配	0	1	0	1	1
	TRG4ABN	MTU4.TADCORA 和 MTU4.TCNT 的比较匹配与 MTU4.TADCORB 和 MTU4.TCNT 的比较匹配 (使用中断减少功能 2 时)	0	1	1	0	0
	TRG7AN	MTU7.TADCORA 和 MTU7.TCNT 的比较匹配	0	1	1	0	1
	TRG7BN	MTU7.TADCORB 和 MTU7.TCNT 的比较匹配	0	1	1	1	0
	TRG7AN 或者 TRG7BN	MTU7.TADCORA 和 MTU7.TCNT 的比较匹配或者 MTU7.TADCORB 和 MTU7.TCNT 的比较匹配	0	1	1	1	1
TRG7ABN	MTU7.TADCORA 和 MTU7.TCNT 的比较匹配与 MTU7.TADCORB 和 MTU7.TCNT 的比较匹配 (使用中断减少功能 2 时)	1	0	0	0	0	
GPT	GTADTRA0N	GPT0.GTADTRA 的比较匹配	1	0	0	0	1
	GTADTRB0N	GPT0.GTADTRB 的比较匹配	1	0	0	1	0
	GTADTRA1N	GPT1.GTADTRA 的比较匹配	1	0	0	1	1
	GTADTRB1N	GPT1.GTADTRB 的比较匹配	1	0	1	0	0
	GTADTRA2N	GPT2.GTADTRA 的比较匹配	1	0	1	0	1
	GTADTRB2N	GPT2.GTADTRB 的比较匹配	1	0	1	1	0
	GTADTRA3N	GPT3.GTADTRA 的比较匹配	1	0	1	1	1
	GTADTRB3N	GPT3.GTADTRB 的比较匹配	1	1	0	0	0
	GTADTRA0N 或者 GTADTRB0N	GPT0.GTADTRA 的比较匹配或者 GPT0.GTADTRB 的比较匹配	1	1	0	0	1
	GTADTRA1N 或者 GTADTRB1N	GPT1.GTADTRA 的比较匹配或者 GPT1.GTADTRB 的比较匹配	1	1	0	1	0
	GTADTRA2N 或者 GTADTRB2N	GPT2.GTADTRA 的比较匹配或者 GPT2.GTADTRB 的比较匹配	1	1	0	1	1
	GTADTRA3N 或者 GTADTRB3N	GPT3.GTADTRA 的比较匹配或者 GPT3.GTADTRB 的比较匹配	1	1	1	0	0

表 31.6 通过 TRSA[4:0] 位选择的 A/D 启动源一览表

模块	启动源	备注	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
不选择触发源的状态			1	1	1	1	1
外部引脚	ADTRG0#	触发输入引脚	0	0	0	0	0
MTU3	TRGA0N	MTU0.TGRA 的比较匹配 / 输入捕捉	0	0	0	0	1
	TRGA1N	MTU1.TGRA 的比较匹配 / 输入捕捉	0	0	0	1	0
	TRGA2N	MTU2.TGRA 的比较匹配 / 输入捕捉	0	0	0	1	1
	TRGA3N	MTU3.TGRA 的比较匹配 / 输入捕捉	0	0	1	0	0
	TRGA4N	MTU4.TGRA 的比较匹配 / 输入捕捉或者 互补 PWM 模式中的 MTU4.TCNT 下溢 (波谷)	0	0	1	0	1
	TRGA6N	MTU6.TGRA 的比较匹配 / 输入捕捉	0	0	1	1	0
	TRGA7N	MTU7.TGRA 的比较匹配 / 输入捕捉或者 互补 PWM 模式中的 MTU7.TCNT 下溢 (波谷)	0	0	1	1	1
	TRG0N	MTU0.TGRE 的比较匹配 / 输入捕捉	0	1	0	0	0
	TRG4AN	MTU4.TADCORA 和 MTU4.TCNT 的比较匹配	0	1	0	0	1
	TRG4BN	MTU4.TADCORB 和 MTU4.TCNT 的比较匹配	0	1	0	1	0
	TRG4AN 或者 TRG4BN	MTU4.TADCORA 和 MTU4.TCNT 的比较匹配或者 MTU4.TADCORB 和 MTU4.TCNT 的比较匹配	0	1	0	1	1
	TRG4ABN	MTU4.TADCORA 和 MTU4.TCNT 的比较匹配与 MTU4.TADCORB 和 MTU4.TCNT 的比较匹配 (使用中断减少功能 2 时)	0	1	1	0	0
	TRG7AN	MTU7.TADCORA 和 MTU7.TCNT 的比较匹配	0	1	1	0	1
	TRG7BN	MTU7.TADCORB 和 MTU7.TCNT 的比较匹配	0	1	1	1	0
	TRG7AN 或者 TRG7BN	MTU7.TADCORA 和 MTU7.TCNT 的比较匹配或者 MTU7.TADCORB 和 MTU7.TCNT 的比较匹配	0	1	1	1	1
TRG7ABN	MTU7.TADCORA 和 MTU7.TCNT 的比较匹配与 MTU7.TADCORB 和 MTU7.TCNT 的比较匹配 (使用中断减少功能 2 时)	1	0	0	0	0	
GPT	GTADTRA0N	GPT0.GTADTRA 的比较匹配	1	0	0	0	1
	GTADTRB0N	GPT0.GTADTRB 的比较匹配	1	0	0	1	0
	GTADTRA1N	GPT1.GTADTRA 的比较匹配	1	0	0	1	1
	GTADTRB1N	GPT1.GTADTRB 的比较匹配	1	0	1	0	0
	GTADTRA2N	GPT2.GTADTRA 的比较匹配	1	0	1	0	1
	GTADTRB2N	GPT2.GTADTRB 的比较匹配	1	0	1	1	0
	GTADTRA3N	GPT3.GTADTRA 的比较匹配	1	0	1	1	1
	GTADTRB3N	GPT3.GTADTRB 的比较匹配	1	1	0	0	0
	GTADTRA0N 或者 GTADTRB0N	GPT0.GTADTRA 的比较匹配或者 GPT0.GTADTRB 的比较匹配	1	1	0	0	1
	GTADTRA1N 或者 GTADTRB1N	GPT1.GTADTRA 的比较匹配或者 GPT1.GTADTRB 的比较匹配	1	1	0	1	0
	GTADTRA2N 或者 GTADTRB2N	GPT2.GTADTRA 的比较匹配或者 GPT2.GTADTRB 的比较匹配	1	1	0	1	1
	GTADTRA3N 或者 GTADTRB3N	GPT3.GTADTRA 的比较匹配或者 GPT3.GTADTRB 的比较匹配	1	1	1	0	0

31.2.10 A/D 采样状态寄存器 n (ADSSTRn) (n=0 ~ 7)

地址 ADSSTR0 0008 9060h、ADSSTR1 0008 9073h、ADSSTR2 0008 9074h、ADSSTR3 0008 9075h、ADSSTR4 0008 9076h、
ADSSTR5 0008 9077h、ADSSTR6 0008 9078h、ADSSTR7 0008 9079h



位	符号	位名	功能	R/W
b7-b0	SST[7:0]	采样时间设定位	以 13 ~ 255 个状态为单位, 设定采样时间。	R/W

ADSSTRn 寄存器是设定模拟输入的采样时间的寄存器。

如果 1 个状态为 1 个 ADCLK (A/D 转换时钟) 的宽度并且 ADCLK 时钟为 50MHz, 1 个状态就为 20ns, 初始值为 20 个状态。能在模拟输入信号源阻抗高而采样时间不够或者 ADCLK 时钟为低速的情况下调整采样时间。必须在 ADCSR.ADST 位为“0”时设定 SST[7:0] 位。必须将采样时间的设定值设定为大于等于 13 个状态并且小于等于 255 个状态的值。另外, 必须将采样时间设定为大于等于 0.4μs。A/D 采样状态寄存器和对象通道的关系如表 31.7 所示。

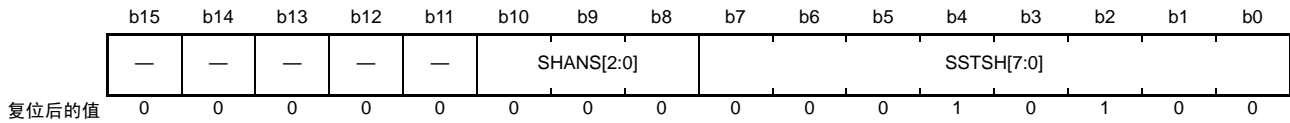
详细内容请参照“31.3.5 模拟输入的采样和扫描转换时间”

表 31.7 A/D 采样状态寄存器和对象通道的关系

位名	对象通道
ADSSTR0.SST[7:0] 位	AN000/ 自诊断
ADSSTR1.SST[7:0] 位	AN001
ADSSTR2.SST[7:0] 位	AN002
ADSSTR3.SST[7:0] 位	AN003
ADSSTR4.SST[7:0] 位	AN004
ADSSTR5.SST[7:0] 位	AN005
ADSSTR6.SST[7:0] 位	AN006
ADSSTR7.SST[7:0] 位	AN007

31.2.11 采样 & 保持电路控制寄存器 (ADSHCR)

地址 0008 9066h



位	符号	位名	功能	R/W
b7-b0	SSTSH[7:0]	采样时间采样 & 保持电路设定位	在 4 ~ 255 个状态间设定采样时间。	R/W
b10-b8	SHANS[2:0]	通道专用采样 & 保持电路的旁路选择位	选择使用 AN000 ~ AN002 的通道专用采样 & 保持电路或者不使用 AN000 ~ AN002 的通道专用采样 & 保持电路 (旁路) 0: 不使用通道专用采样 & 保持电路 (旁路) 1: 使用通道专用采样 & 保持电路	R/W
b15-b11	—	保留位	读写值都为“0”。	R/W

ADSHCR 寄存器是设定通道专用采样 & 保持电路的寄存器。

SSTSH[7:0] 位 (采样时间采样 & 保持电路设定位)

这些位设定通道专用采样 & 保持电路的采样时间。如果 1 个状态为 1 个 ADCLK (A/D 转换时钟) 的宽度并且 ADCLK 时钟为 50MHz, 1 个状态就为 20ns, 初始值为 20 个状态。能在模拟输入信号源阻抗高而采样时间不够或者 ADCLK 时钟为低速的情况下调整采样时间。必须在 ADCSR.ADST 位为“0”时设定 SSTSH[7:0] 位。必须将采样时间的设定值设定为大于等于 4 个状态并且小于等于 255 个状态的值。另外, 必须将采样时间设定为大于等于 0.6μs。例如, ADCLK 时钟为 50MHz 时, 采样时间设定值的下限为 30 个状态。

SHANS[2:0] 位 (通道专用采样 & 保持电路的旁路选择位)

这些位选择使用模拟输入 AN000 ~ AN002 的通道专用采样 & 保持电路或者不使用模拟输入 AN000 ~ AN002 的通道专用采样 & 保持电路 (旁路)。SHANS[0] 位为 AN000 的通道专用采样 & 保持电路的选择位, SHANS[1] 位为 AN001 的通道专用采样 & 保持电路的选择位, SHANS[2] 位为 AN002 的通道专用采样 & 保持电路的选择位。必须在 ADCSR.ADST 位为“0”时设定 SHANS[2:0] 位。

在双触发模式的组 A 优先控制运行模式中, 如果给组 B 选择 AN000 ~ AN002 中的任意一个, 就必须将通道专用采样 & 保持电路设定为旁路电路。

31.2.12 A/D 分组扫描优先控制寄存器 (ADGSPCR)

地址 0008 9080h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GBRP	—	—	—	—	—	—	—	—	—	—	—	—	—	GBRS CN	PGS
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	PGS	组 A 优先控制设定位 (注 1)	0: 不进行组 A 的优先控制运行 1: 进行组 A 的优先控制运行	R/W
b1	GBRSCN	组 B 重新启动设定位	(只在 PGS=1 时有效, 在 PGS=0 时为保留位) 0: 不重新启动因组 A 的优先控制而中止的组 B 的 A/D 转换运行 1: 重新启动因组 A 的优先控制而中止的组 B 的 A/D 转换运行	R/W
b14-b2	—	保留位	读写值都为“0”。	R/W
b15	GBRP	组 B 的 1 个周期扫描连续启动设定位 (注 2)	(只在 PGS=1 时有效, 在 PGS=0 时为保留位) 0: 不连续进行组 B 的 1 个周期扫描运行 1: 连续进行组 B 的 1 个周期扫描运行	R/W

注 1. 如果将 PGS 位置“1”，就必须将 ADCSR.ADCS[1:0] 位置“01b”（分组扫描模式），否则，不保证运行。

注 2. 如果将 GBRP 位置“1”，就与 GBRSCN 位的设定无关，连续进行组 B 的 1 个周期扫描。

ADGSPCR 寄存器是设定优先控制分组扫描模式中组 A 的 A/D 转换的寄存器。

PGS 位 (组 A 优先控制设定位)

此位优先控制组 A 的运行。在进行组 A 优先控制运行时，必须将此位置“1”。

如果将 PGS 位置“1”，就必须将 ADCSR.ADCS[1:0] 位置“01b”（分组扫描模式），否则，不保证运行。

如果要将 PGS 位置“0”，就必须根据“31.6.2 停止 A/D 转换时的注意事项”，通过软件进行清除。如果要将 PGS 位置“1”，就必须根据“31.3.4.4 组 A 优先控制运行”的步骤进行设定。

GBRSCN 位 (组 B 重新启动设定位)

此位设定组 A 优先控制时的组 B 的重新扫描运行。

如果将 GBRSCN 位置“1”，就在因组 A 的触发输入而中止组 B 的扫描运行后，等待组 A 的 A/D 转换运行结束，开始组 B 的重新扫描。另外，如果在组 A 的 A/D 转换运行过程中输入组 B 的触发，就在等待组 A 的 A/D 转换运行结束后进行组 B 的重新扫描。

但是，如果连续输入组 A 的触发，就不进行组 B 的重新扫描。

如果将 GBRSCN 位置“0”，就忽视在 A/D 转换运行过程中输入的触发。另外，必须在 ADCSR.ADST 位为“0”时设定 GBRSCN 位。

在 PGS 位为“1”时 GBRSCN 位的设定有效。

GBRP 位 (组 B 的 1 个周期扫描连续启动设定位)

此位设定连续进行组 B 的 1 个周期扫描。

如果将 GBRP 位置“1”，就启动组 B 的 1 个周期扫描。扫描结束后，自动重新开始组 B 的 1 个周期扫描。在因组 A 优先控制而中止组 B 的 A/D 转换运行后，如果组 A 的 A/D 转换运行结束，就自动重新开始组 B 的 1 个周期扫描。

必须事先将组 B 的触发输入置为无效后将 GBRP 位置“1”。在将 GBRP 位置“1”后，GBRSCN 位的设定无效。

必须在 ADCSR.ADST 位为“0”时设定 GBRP 位。

在 PGS 位为“1”时 GBRP 位的设定有效。

31.2.13 比较器运行模式选择寄存器 0 (ADCMPMD0)

地址 0008 90E0h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	CEN002[1:0]	CEN001[1:0]	CEN000[1:0]			
0	0	0	0	0	0	0	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	CEN000[1:0]	AN000 的比较器选择位	b1 b0 0 0: 不使用比较器 0 1: 作为 Low 电平比较器使用 (检测到比 Low 电平侧基准电压还要低的输入电压) 1 0: 作为 High 电平比较器使用 (检测到比 High 电平侧基准电压还要高的输入电压) 1 1: 作为窗口比较器使用 (检测到在 Low 电平侧基准电压~ High 电平侧基准电压范围外的输入电压)	R/W
b3-b2	CEN001[1:0]	AN001 的比较器选择位	b3 b2 0 0: 不使用比较器 0 1: 作为 Low 电平比较器使用 (检测到比 Low 电平侧基准电压还要低的输入电压) 1 0: 作为 High 电平比较器使用 (检测到比 High 电平侧基准电压还要高的输入电压) 1 1: 作为窗口比较器使用 (检测到在 Low 电平侧基准电压~ High 电平侧基准电压范围外的输入电压)	R/W
b5-b4	CEN002[1:0]	AN002 的比较器选择位	b5 b4 0 0: 不使用比较器 0 1: 作为 Low 电平比较器使用 (检测到比 Low 电平侧基准电压还要低的输入电压) 1 0: 作为 High 电平比较器使用 (检测到比 High 电平侧基准电压还要高的输入电压) 1 1: 作为窗口比较器使用 (检测到在 Low 电平侧基准电压~ High 电平侧基准电压范围外的输入电压)	R/W
b15-b6	—	保留位	读写值都为 "0"。	R/W

ADCMPMD0 寄存器是设定是否使用比较器的寄存器。

CENn[1:0] 位 (ANn 的比较器选择位) (n=000 ~ 002)

这些位设定是否使用各比较器以及运行模式。

31.2.14 比较器运行模式选择寄存器 1 (ADCMPMD1)

地址 0008 90E2h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	VSELLO	VSELH0	—	—	REFH[2:0]			—	REFL[2:0]		
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
b2-b0	REFL[2:0]	比较器 Low 电平侧基准电压的内部电压选择位	b2 b1 b0 0 0 0: 无效 0 0 1: AVCC0x1/8 0 1 0: AVCC0x2/8 0 1 1: AVCC0x3/8 1 0 0: AVCC0x4/8 1 0 1: AVCC0x5/8 1 1 0: AVCC0x6/8 1 1 1: AVCC0x7/8	R/W
b3	—	保留位	读写值都为“0”。	R/W
b6-b4	REFH[2:0]	比较器 High 电平侧基准电压的内部电压选择位	b6 b5 b4 0 0 0: 无效 0 0 1: AVCC0x1/8 0 1 0: AVCC0x2/8 0 1 1: AVCC0x3/8 1 0 0: AVCC0x4/8 1 0 1: AVCC0x5/8 1 1 0: AVCC0x6/8 1 1 1: AVCC0x7/8	R/W
b8-b7	—	保留位	读写值都为“0”。	R/W
b9	VSELH0	比较器 High 电平侧基准电压选择位	0: 从 AN007 引脚输入 High 电平侧基准电压 1: 将 REFH[2:0] 位选择的内部电压输入到 High 电平侧基准电压	R/W
b10	VSELLO	比较器 Low 电平侧基准电压选择位	0: 从 AN003 引脚输入 Low 电平侧基准电压 1: 将 REFL[2:0] 位选择的内部电压输入到 Low 电平侧基准电压	R/W
b15-b11	—	保留位	读写值都为“0”。	R/W

ADCMPMD1 寄存器是设定比较器的输入和基准电压的寄存器。

REFL[2:0] 位 (比较器 Low 电平侧基准电压的内部电压选择位)

这些位设定使用内部电压作为比较器 Low 电平侧基准电压时的电压值。

电压值的设定范围根据使用条件而不同, 请参照“36.6 振荡停止检测时序”的 REFL 电压范围。

REFH[2:0] 位 (比较器 High 电平侧基准电压的内部电压选择位)

这些位设定使用内部电压作为比较器 High 电平侧基准电压时的电压值。

电压值的设定范围根据使用条件而不同, 请参照“36.6 振荡停止检测时序”的 REFH 电压范围。

VSELH0 位 (比较器 High 电平侧基准电压选择位)

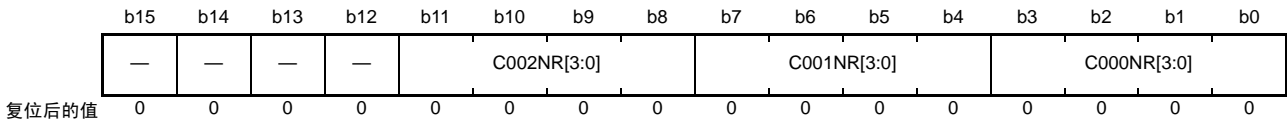
此位设定比较器 High 电平侧基准电压的输入方法。

VSELLO 位 (比较器 Low 电平侧基准电压选择位)

此位设定比较器 Low 电平侧基准电压的输入方法。

31.2.15 比较器的滤波器模式寄存器 0 (ADCMPNR0)

地址 0008 90E4h



位	符号	位名	功能	R/W
b3-b0	C000NR[3:0]	AN000 的比较器噪声消除滤波器模式选择位	b3 b0 0 0 0 0: 不对比较器检测结果进行采样 1 0 0 0: 用 PCLK 对比较器结果进行 16 次采样 1 0 0 1: 用 PCLK/2 对比较器结果进行 16 次采样 1 0 1 0: 用 PCLK/4 对比较器结果进行 16 次采样 1 0 1 1: 用 PCLK/8 对比较器结果进行 16 次采样 1 1 0 0: 用 PCLK/16 对比较器结果进行 16 次采样 1 1 0 1: 用 PCLK/128 对比较器结果进行 16 次采样 上述以外: 不能设定	R/W
b7-b4	C001NR[3:0]	AN001 的比较器噪声消除滤波器模式选择位	b7 b4 0 0 0 0: 不对比较器检测结果进行采样 1 0 0 0: 用 PCLK 对比较器结果进行 16 次采样 1 0 0 1: 用 PCLK/2 对比较器结果进行 16 次采样 1 0 1 0: 用 PCLK/4 对比较器结果进行 16 次采样 1 0 1 1: 用 PCLK/8 对比较器结果进行 16 次采样 1 1 0 0: 用 PCLK/16 对比较器结果进行 16 次采样 1 1 0 1: 用 PCLK/128 对比较器结果进行 16 次采样 上述以外: 不能设定	R/W
b11-b8	C002NR[3:0]	AN002 的比较器噪声消除滤波器模式选择位	b11 b8 0 0 0 0: 不对比较器检测结果进行采样 1 0 0 0: 用 PCLK 对比较器结果进行 16 次采样 1 0 0 1: 用 PCLK/2 对比较器结果进行 16 次采样 1 0 1 0: 用 PCLK/4 对比较器结果进行 16 次采样 1 0 1 1: 用 PCLK/8 对比较器结果进行 16 次采样 1 1 0 0: 用 PCLK/16 对比较器结果进行 16 次采样 1 1 0 1: 用 PCLK/128 对比较器结果进行 16 次采样 上述以外: 不能设定	R/W
b15-b12	—	保留位	读写值都为“0”。	R/W

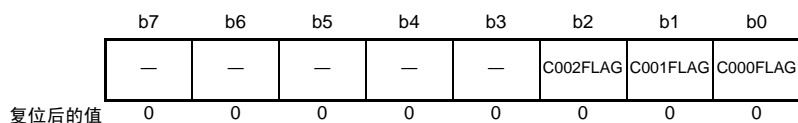
ADCMPNR0 寄存器设定比较器检测结果的噪声滤波器的运行。

CnNR[3:0] 位 (ANn 的比较器噪声消除滤波器模式选择位) (n=000 ~ 002)

这些位设定 ANn 的比较器检测结果的噪声滤波器的运行。在已设定的条件下对比较器检测结果进行采样, 如果全部检测到, ADCMPFR.CnFLAG 标志就被置位。此时, 能通过设定 ADCMPSEL 寄存器, 请求比较器中断 (CMP0 ~ CMP2) 和端口输出允许 3 (POE3) 的启动。

31.2.16 比较器检测标志寄存器 (ADCMPFR)

地址 0008 90E8h



位	符号	位名	功能	R/W
b0	C000FLAG	AN000 的比较器检测标志	0: 未检测比较器 1: 检测比较器	R/(W) (注1)
b1	C001FLAG	AN001 的比较器检测标志	0: 未检测比较器 1: 检测比较器	R/(W) (注1)
b2	C002FLAG	AN002 的比较器检测标志	0: 未检测比较器 1: 检测比较器	R/(W) (注1)
b7-b3	—	保留位	读写值都为“0”。	R/W

注 1. 只能写用于清除标志的“0”。

ADCMPFR 寄存器是表示各比较器的检测 / 未检测状态的寄存器。

CnFLAG 标志 (ANn 的比较器检测标志) (n=000 ~ 002)

这是表示各比较器的检测 / 未检测状态的标志。

[为“1”的条件]

- 在 ADCMPNR0.CnNR[3:0] (n=000 ~ 002) 位设定的条件下对比较器检测结果进行采样并且全部检测到时

[为“0”的条件]

- 通过软件写“0”时

31.2.17 比较器中断选择寄存器 (ADCMPSSEL)

地址 0008 90EAh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	POERQ 002	POERQ 001	POERQ 000	—	—	—	—	—	IE002	IE001	IE000
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	IE000	AN000 的比较器检测中断 (CMP0) 允许位	0: 禁止发生中断 1: 允许发生中断	R/W
b1	IE001	AN001 的比较器检测中断 (CMP1) 允许位	0: 禁止发生中断 1: 允许发生中断	R/W
b2	IE002	AN002 的比较器检测中断 (CMP2) 允许位	0: 禁止发生中断 1: 允许发生中断	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W
b8	POERQ 000	AN000 的比较器检测 POE 请求设定位	0: 禁止产生 AN000 的比较器检测时的 POE 请求 1: 允许产生 AN000 的比较器检测时的 POE 请求	R/W
b9	POERQ 001	AN001 的比较器检测 POE 请求设定位	0: 禁止产生 AN001 的比较器检测时的 POE 请求 1: 允许产生 AN001 的比较器检测时的 POE 请求	R/W
b10	POERQ 002	AN002 的比较器检测 POE 请求设定位	0: 禁止产生 AN002 的比较器检测时的 POE 请求 1: 允许产生 AN002 的比较器检测时的 POE 请求	R/W
b15-b10	—	保留位	读写值都为“0”。	R/W

ADCMPSSEL 寄存器是设定将比较器检测标志用作 CPU 中断源或者 POE 源的寄存器。

IE_n 位 (AN_n 的比较器检测中断允许位) (n=000 ~ 002)

这些位禁止或者允许发生各比较器检测中断 (CMP_n)。

POERQ_n 位 (AN_n 的比较器检测 POE 请求设定位) (n=000 ~ 002)

这些位禁止或者允许产生各比较器检测时的 POE 请求。POE 请求为 POERQ_n 位选择的比较器检测的逻辑或。

31.3 运行说明

31.3.1 扫描的运行说明

扫描是指依次对选择的通道的模拟输入进行 A/D 转换。

扫描转换的运行模式有 3 种，分别是 1 个周期扫描模式、连续扫描模式和分组扫描模式。

1 个周期扫描模式是指在对选择的至少 1 个通道进行 1 次扫描后结束的模式。连续扫描模式是指对选择的至少 1 个通道无限制地重复进行扫描，直到通过软件将 ADCSR.ADST 位清“0”（从“1”变为“0”）为止。分组扫描模式是指通过分别选择的触发开始组 A 和组 B 的扫描，并且在分别对组 A 和组 B 所选通道进行 1 次扫描后结束。

如果在 1 个周期扫描模式或者连续扫描模式中开始扫描转换，就从 ADANSA 寄存器选择的 ANn 中按照 n 从小到大的顺序开始 A/D 转换。在分组扫描模式中，组 A 从 ADANSA 寄存器选择的 ANn 中按照 n 从小到大的顺序开始 A/D 转换，组 B 从 ADANSB 寄存器选择的 ANn 中按照 n 从小到大的顺序开始 A/D 转换。

如果选择自诊断时，就在每次扫描的最开始执行 1 次自诊断，对 12 位 A/D 转换器内部生成的 3 个电压值中的 1 个电压值进行 A/D 转换。

双触发模式用于 1 个周期扫描模式或者分组扫描模式。如果允许双触发模式，只在通过 ADSTRGR.TRSA[4:0] 位选择的 MTU3 触发或者 GPT 触发启动扫描时，对 ADCSR.DBLANS[4:0] 位选择的 1 个通道的 A/D 转换数据进行双重化。

如果通过 ADSHCR.SHANS[2:0] 位设定 AN000 ~ AN002 中的任意 1 个使用通道专用采样 & 保持电路，就在每次扫描的第 1 个 A/D 转换开始前，进行对象模拟输入的采样 & 保持。

31.3.2 1 个周期扫描模式

31.3.2.1 基本运行（无通道专用采样 & 保持电路）

1 个周期扫描模式的基本运行是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. 如果通过软件触发、同步触发（MTU3、GPT）或者异步触发的输入将 ADCSR.ADST 位置“1”（开始 A/D 转换），就从 ADANSA 寄存器所选 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
2. 如果 1 个通道的 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. 在所选通道的 A/D 转换全部结束后，如果 ADCSR.ADIE 位已被置“1”（允许由扫描结束引起的 S12ADI 中断），就产生 S12ADI 中断请求。
4. ADST 位在 A/D 转换过程中保持“1”（开始 A/D 转换），一旦所选通道的 A/D 转换全部结束就自动清除，并且 12 位 A/D 转换器进入待机状态。

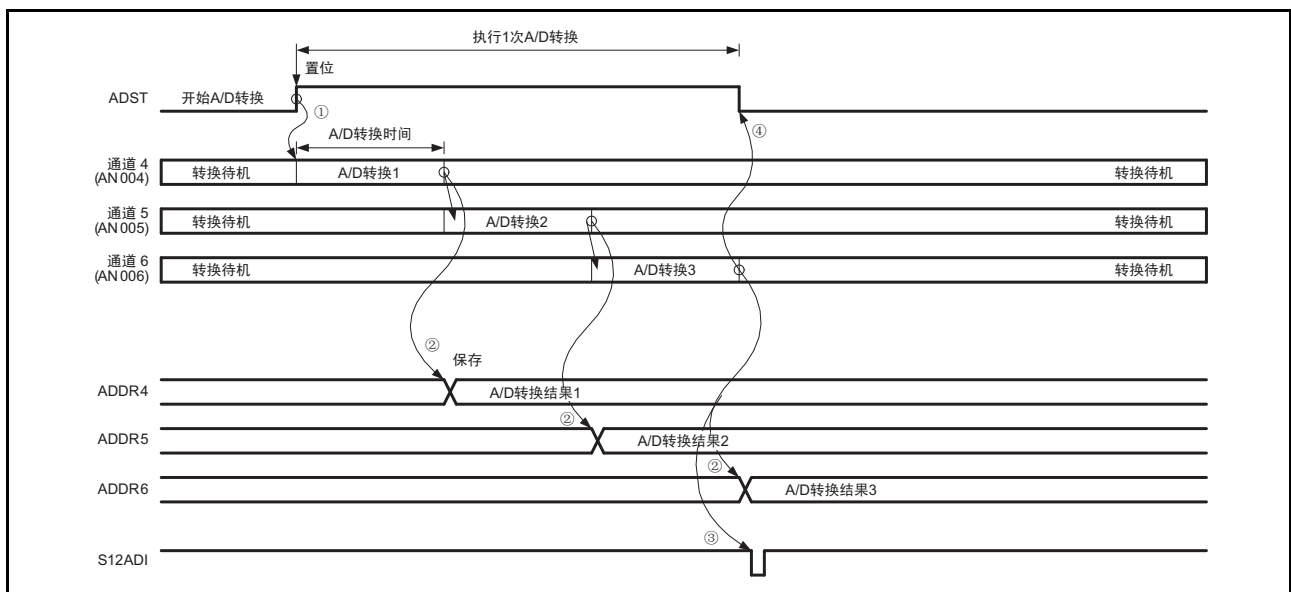


图 31.3 1 个周期扫描模式的运行例子（基本运行：选择 AN004 ~ AN006）

31.3.2.2 基本运行 (有通道专用采样 & 保持电路)

如果使用通道专用采样 & 保持电路,就在执行采样 & 保持后对所选的全部通道的模拟输入只进行 1 次 A/D 转换,如下所示。通过 ADShCR.SHANS[2:0] 位选择使用通道专用采样 & 保持电路的通道。

1. 如果通过软件触发、同步触发 (MTU3、GPT) 或者异步触发的输入将 ADCSR.ADST 位置“1” (开始 A/D 转换), 就开始对使用通道专用采样 & 保持电路的全部通道的模拟输入进行采样。
2. 在执行采样 & 保持后, 从 ADANSA 寄存器选择的通道 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
3. 如果 1 个通道的 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)。
4. 在所选通道的 A/D 转换全部结束后, 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断), 就产生 S12ADI 中断请求。
5. ADST 位在 A/D 转换过程中保持“1” (开始 A/D 转换), 一旦所选通道的 A/D 转换全部结束就自动清除, 并且 12 位 A/D 转换器进入待机状态。

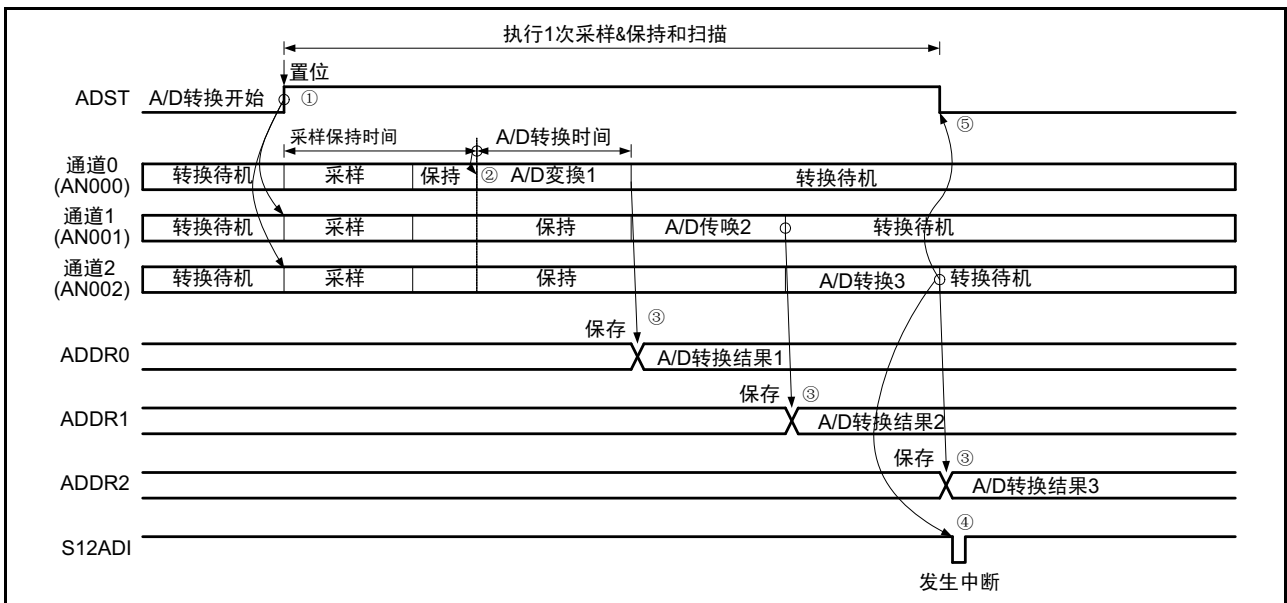


图 31.4 1 个周期扫描模式的运行例子 (有通道专用采样 & 保持电路)

31.3.2.3 通道选择和自诊断 (无通道专用采样 & 保持电路)

如果在选择通道的同时选择自诊断, 就对提供给 12 位 A/D 转换器的基准电压 VREFH0 (×0、×1/2、×1 中的任意 1 个) 进行 A/D 转换, 之后对所选通道的模拟输入只进行 1 次 A/D 转换, 如下所示。

1. 如果通过软件触发、同步触发 (MTU3、GPT) 或者异步触发的输入将 ADCSR.ADST 位置“1” (开始 A/D 转换), 就最先开始自诊断的 A/D 转换。
2. 在自诊断的 A/D 转换结束后, 将 A/D 转换结果保存到 A/D 自诊断数据寄存器 (ADRD), 然后从 ADANSA 寄存器选择的通道 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
3. 如果 1 个通道的 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)。
4. 在所选通道的 A/D 转换全部结束后, 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断), 就产生 S12ADI 中断请求。
5. ADST 位在 A/D 转换过程中保持“1” (开始 A/D 转换), 一旦所选通道的 A/D 转换全部结束就自动清除, 并且 12 位 A/D 转换器进入待机状态。

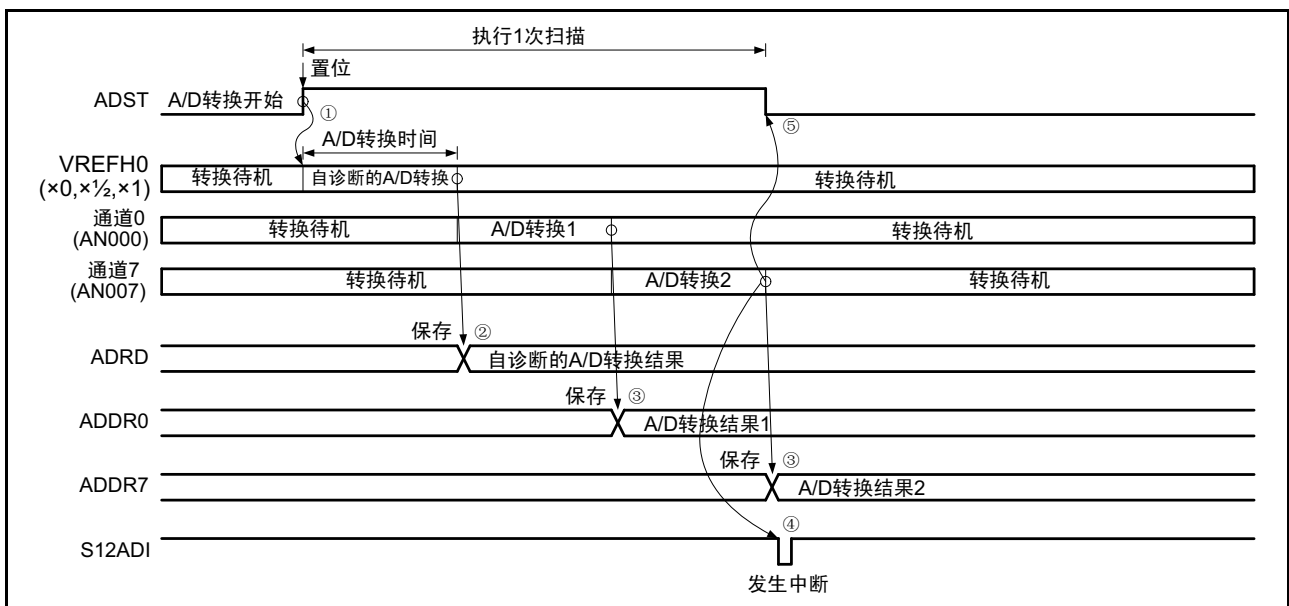


图 31.5 1 个周期扫描模式的运行例子 (基本运行和自诊断)

31.3.2.4 通道选择和自诊断 (有通道专用采样 & 保持电路)

如果在选择通道的同时选择自诊断, 并且使用通道专用采样 & 保持电路, 就在执行采样 & 保持后对提供给 12 位 A/D 转换器的基准电压 VREFH0 (x0、x1/2、x1 中的任意 1 个) 进行 A/D 转换, 之后对所选通道的模拟输入只进行 1 次 A/D 转换, 如下所示。

1. 如果通过软件触发、同步触发 (MTU3、GPT) 或者异步触发的输入将 ADCSR.ADST 位置“1” (开始 A/D 转换), 就开始对使用通道专用采样 & 保持电路的全部通道的模拟输入进行采样。
2. 在执行采样 & 保持后, 开始自诊断的 A/D 转换。
3. 在自诊断的 A/D 转换结束后, 将 A/D 转换结果保存到 A/D 自诊断数据寄存器 (ADRD), 然后从 ADANSA 寄存器选择的通道 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
4. 如果 1 个通道的 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)。
5. 在所选通道的 A/D 转换全部结束后, 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断), 就产生 S12ADI 中断请求。
6. ADST 位在 A/D 转换过程中保持“1” (开始 A/D 转换), 一旦所选通道的 A/D 转换全部结束就自动清除, 并且 12 位 A/D 转换器进入待机状态。

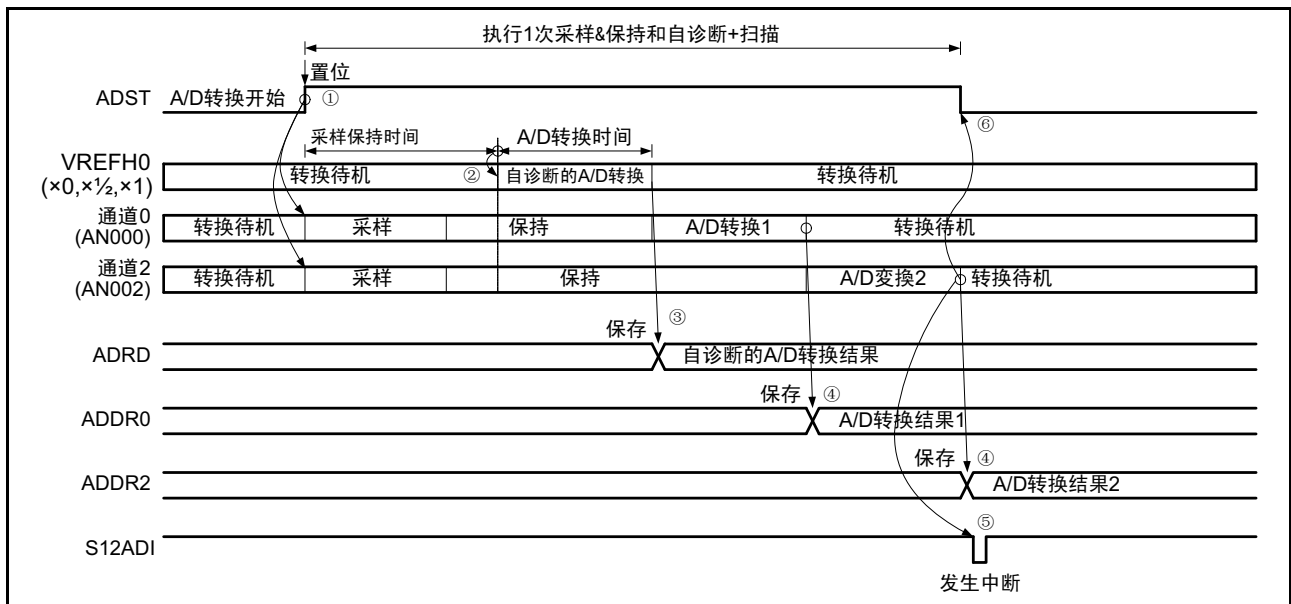


图 31.6 1 个周期扫描模式的运行例子 (有通道专用采样 & 保持电路和自诊断)

31.3.2.5 选择双触发模式时的运行

如果在 1 个周期扫描模式中选择双触发模式，就通过 MTU3 触发或者 GPT 触发执行 2 次连串的 1 个周期扫描模式，如下所示。

将自诊断设定为不选择。

如果将要双重化的通道号设定到 ADCSR.DBLANS[4:0] 位，并且将 ADCSR.DBLE 位置“1”，A/D 转换数据的双重化就有效。将 ADCSR.DBLE 位置“1”时，ADANSA 寄存器的通道选择就无效。另外，在选择双触发模式时，必须通过 ADSTRGR.TRSA[4:0] 位选择 MTU3 触发或者 GPT 触发，并且将 ADCSR.EXTRG 位置“0”、ADCSR.TRGE 位置“1”。而且不能使用软件触发。

1. 如果通过 MTU3 触发或者 GPT 触发的输入将 ADCSR.ADST 位置“1”（开始 A/D 转换），就开始对 ADCSR.DBLANS[4:0] 位选择的 1 个通道进行 A/D 转换。
2. 如果 1 个通道的 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. 自动清除 ADST 位，并且 12 位 A/D 转换器进入待机状态。此时，与 ADCSR.ADIE 位（允许由扫描结束引起的 S12ADI 中断）的设定无关，不发生 S12ADI 中断。
4. 如果通过第 2 次触发输入将 ADCSR.ADST 位置“1”（开始 A/D 转换），就开始对 ADCSR.DBLANS[4:0] 位选择的 1 个通道进行 A/D 转换。
5. 如果 A/D 转换结束，就将 A/D 转换结果保存到双触发模式专用的 A/D 数据双重化寄存器 (ADDBLDR)。
6. 如果 ADCSR.ADIE 位已被置“1”（允许由扫描结束引起的 S12ADI 中断），就产生 S12ADI 中断请求。
7. ADST 位在 A/D 转换过程中保持“1”（开始 A/D 转换），一旦 A/D 转换结束就自动清除，并且 12 位 A/D 转换器进入待机状态。

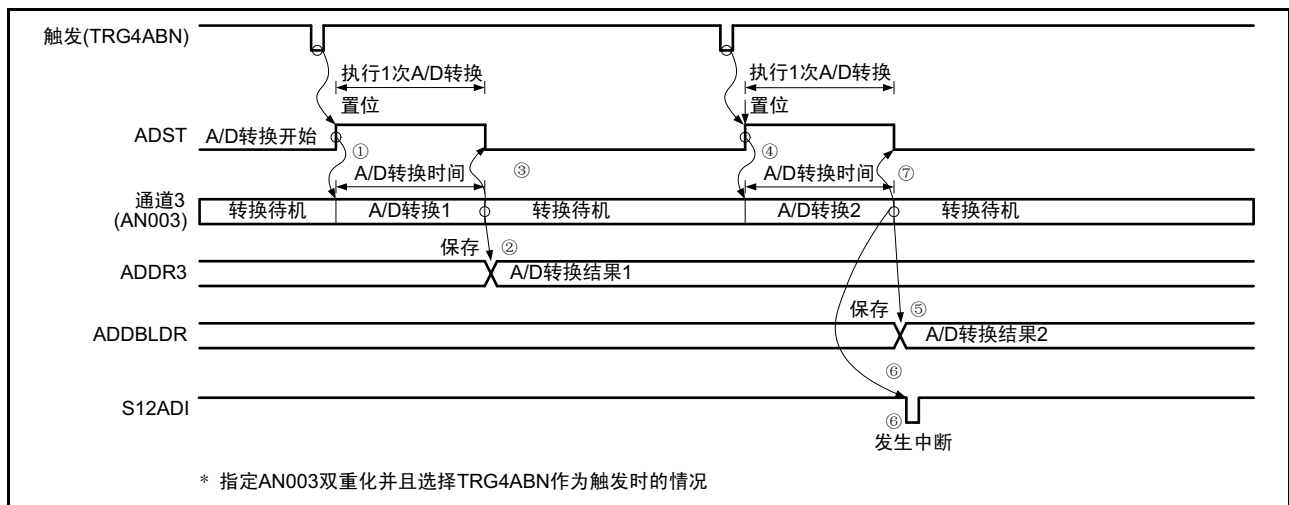


图 31.7 1 个周期扫描模式的运行例子（选择双触发模式、AN003 的双重化）

31.3.2.6 选择双触发模式时的扩展运行

如果在 1 个周期扫描模式中选择双触发模式，就在选择同步触发的 TRGnAN 或者 TRGnBN (n=4、7) 作为 A/D 转换开始触发 (给 ADSTRGR.TSRA[4:0] 位设定“0Bh”、“0Fh”) 时，又或者在选择 GTADTRAmN 或者 GTADTRBmN (m=0~3) 作为 A/D 转换开始触发 (给 ADSTRGR.TSRA[4:0] 位设定“19h”、“1Ah”、“1Bh”、“1Ch”) 时，在上述选择双触发模式时的运行中追加以下运行。

将通过同步触发 TRGnAN (n=4、7) 或者 GTADTRAmN (m=0~3) 开始的 A/D 转换的转换结果保存到 A/D 数据双重化寄存器 A (ADDBLDRA)，将通过同步触发 TRGnBN (n=4、7) 或者 GTADTRBmN (m=0~3) 开始的 A/D 转换的转换结果保存到 A/D 数据双重化寄存器 B (ADDBLDRB)。由此可见，同步触发源与保存寄存器对应，而且与触发输入的顺序无关，A/D 转换结果被保存到对应的保存寄存器。但是，与此同时，A/D 转换结果又按照触发输入的顺序，被保存到对应的 A/D 数据寄存器 (ADDRy) 和 A/D 数据双重化寄存器 (ADDBLDR)。

在双触发扩展模式中，在选择 TRGnAN 或者 TRGnBN (n=4、7)，又或者选择 GTADTRAmN 或者 GTADTRBmN (m=0~3) 时，如果同时产生 2 种触发源，A/D 转换结果就不根据触发源进行分配，而保存到数据双重化寄存器 B (ADDBLDRB)。

另外，如果在通过其中一个触发源进行 A/D 转换的过程中输入另一个触发源，就忽视该触发源，并且根据正在进行转换的触发源来进行分配。

选择同步触发 TRG4AN 或者 TRG4BN 作为 A/D 转换开始触发，并且第 1 次触发为 TRG4AN 时的运行如下所示。

1. 如果通过 TRG4AN 输入将 ADCSR.ADST 位置“1” (开始 A/D 转换)，就开始对 ADCSR.DBLANS[4:0] 位选择的 1 个通道进行 A/D 转换。
2. 如果 1 个通道的 A/D 转换结束，就将 A/D 转换结果保存到 A/D 数据双重化寄存器 A (ADDBLDRA) 和对应的 A/D 数据寄存器 (ADDRy)。
3. 自动清除 ADST 位，并且 12 位 A/D 转换器进入待机状态。此时，与 ADCSR.ADIE 位 (允许由扫描结束引起的 S12ADI 中断) 的设定无关，不发生 S12ADI 中断。
4. 如果通过 TRG4BN 输入将 ADCSR.ADST 位置“1” (开始 A/D 转换)，就开始对 ADCSR.DBLANS[4:0] 位选择的 1 个通道进行 A/D 转换。
5. 如果 A/D 转换结束，就将 A/D 转换结果保存到 A/D 数据双重化寄存器 B (ADDBLDRB) 和 A/D 数据双重化寄存器 (ADDBLDR)。
6. 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断)，就产生 S12ADI 中断请求。
7. ADST 位在 A/D 转换过程中保持“1” (开始 A/D 转换)，一旦 A/D 转换结束就自动清除，并且 12 位 A/D 转换器进入待机状态。

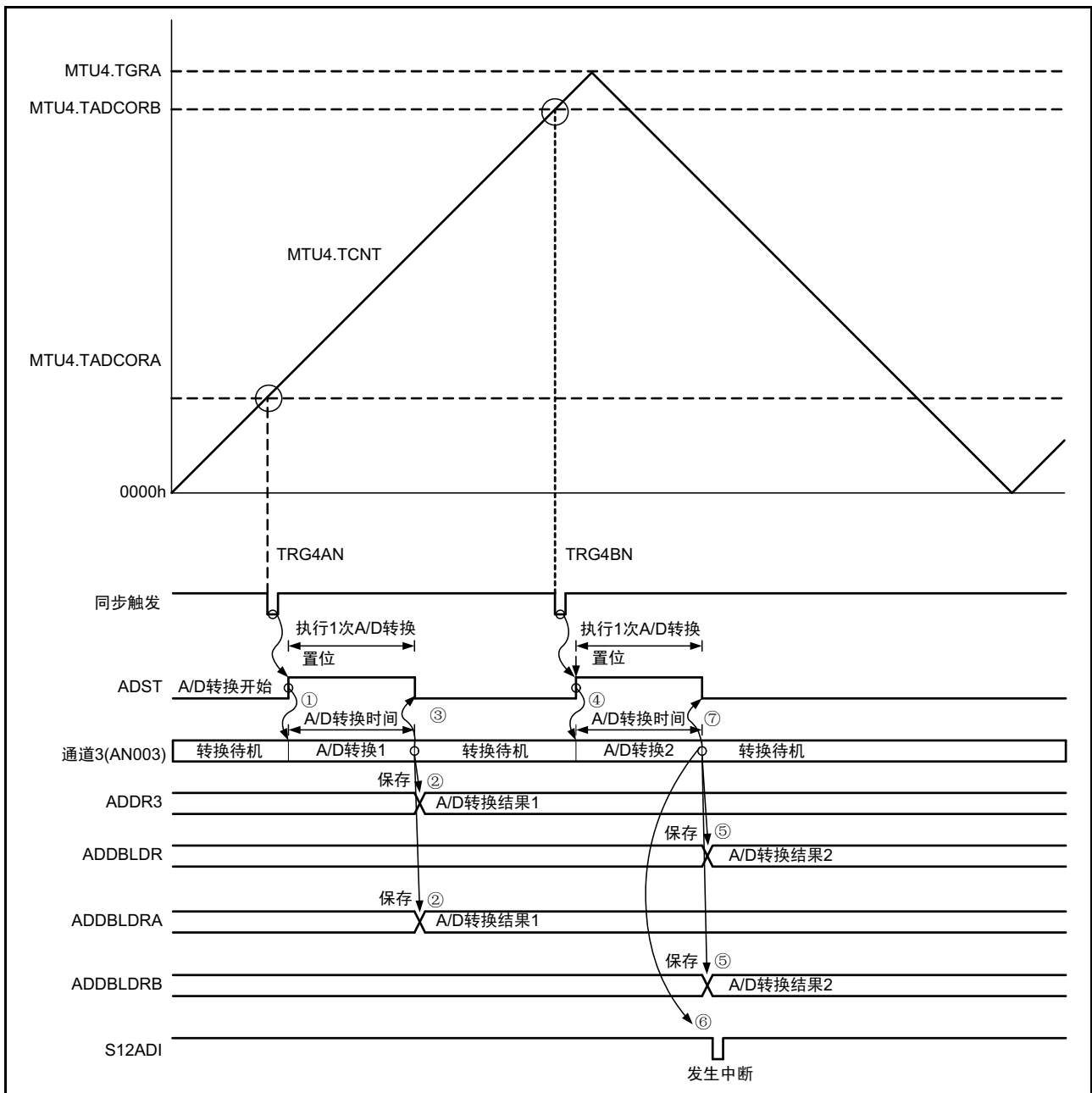


图 31.8 双触发模式的扩展运行例子 (1)
 (选择 AN003 双重化、选择 TRG4AN 或者 TRG4BN、并且第 1 次触发为 TRG4AN 的情况)

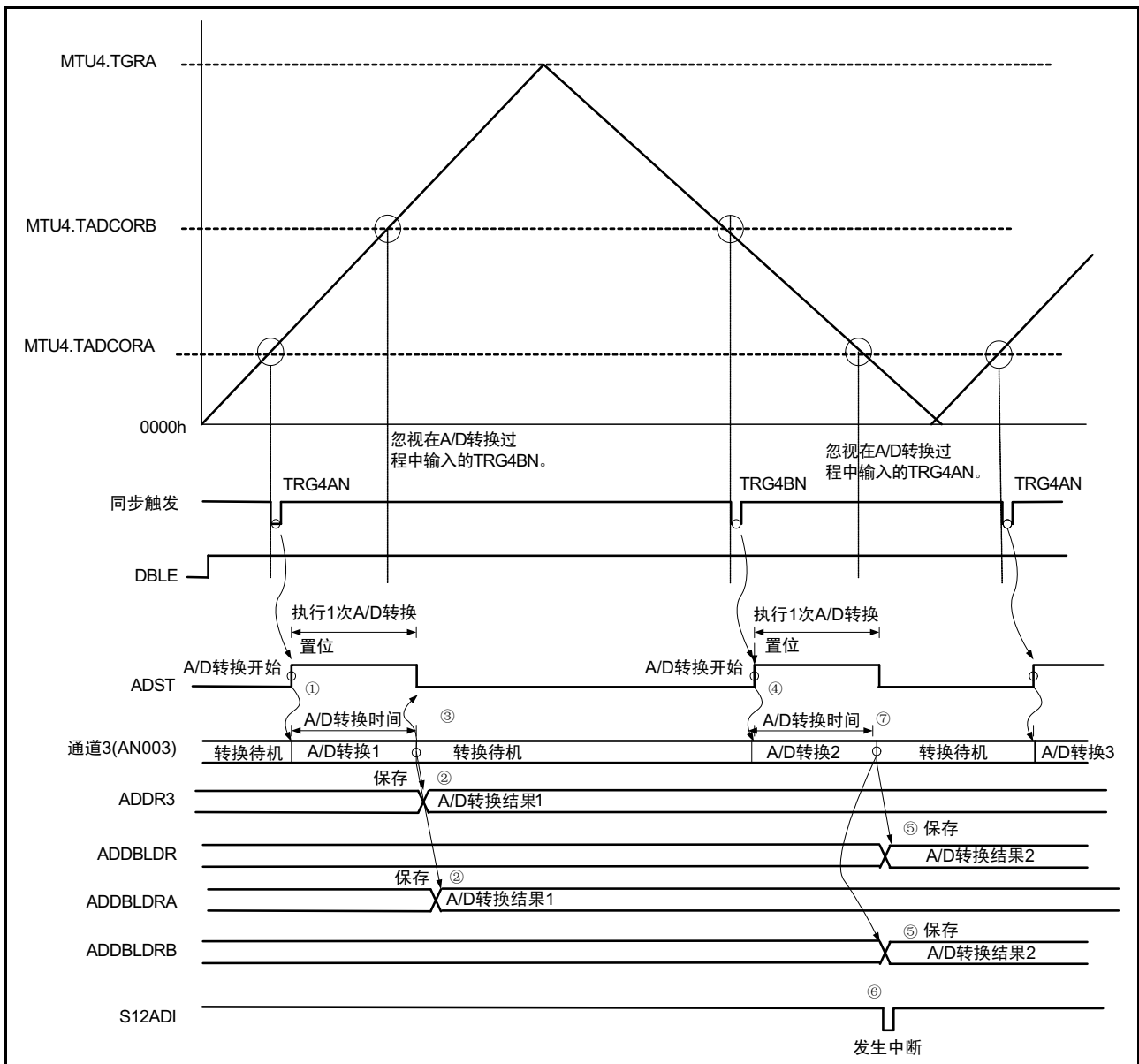


图 31.9 双触发模式的扩展运行例子 (2)
 (选择 AN003 双重化、选择 TRG4AN 或者 TRG4BN、并且第 1 次触发为 TRG4AN 的情况)

选择同步触发 TRG4AN 或者 TRG4BN 作为 A/D 转换开始触发，并且第 1 次触发为 TRG4BN 时的运行如下所示。

1. 如果通过 TRG4BN 输入将 ADCSR.ADST 位置“1”（开始 A/D 转换），就开始对 ADCSR.DBLANS[4:0] 位选择的 1 个通道进行 A/D 转换。
2. 如果 1 个通道的 A/D 转换结束，就将 A/D 转换结果保存到 A/D 数据双重化寄存器 B（ADDBLDRB）和对应的 A/D 数据寄存器（ADDRy）。
3. 自动清除 ADST 位，并且 12 位 A/D 转换器进入待机状态。此时，与 ADCSR.ADIE 位（允许由扫描结束引起的 S12ADI 中断）的设定无关，不发生 S12ADI 中断。
4. 如果通过 TRG4AN 输入将 ADCSR.ADST 位置“1”（开始 A/D 转换），就开始对 ADCSR.DBLANS[4:0] 位选择的 1 个通道进行 A/D 转换。
5. 如果 A/D 转换结束，就将 A/D 转换结果保存到 A/D 数据双重化寄存器 A（ADDBLDRA）和 A/D 数据双重化寄存器（ADDBLDR）。
6. 如果 ADCSR.ADIE 位已被置“1”（允许由扫描结束引起的 S12ADI 中断），就产生 S12ADI 中断请求。
7. ADST 位在 A/D 转换过程中保持“1”（开始 A/D 转换），一旦 A/D 转换结束就自动清除，并且 12 位 A/D 转换器进入待机状态。

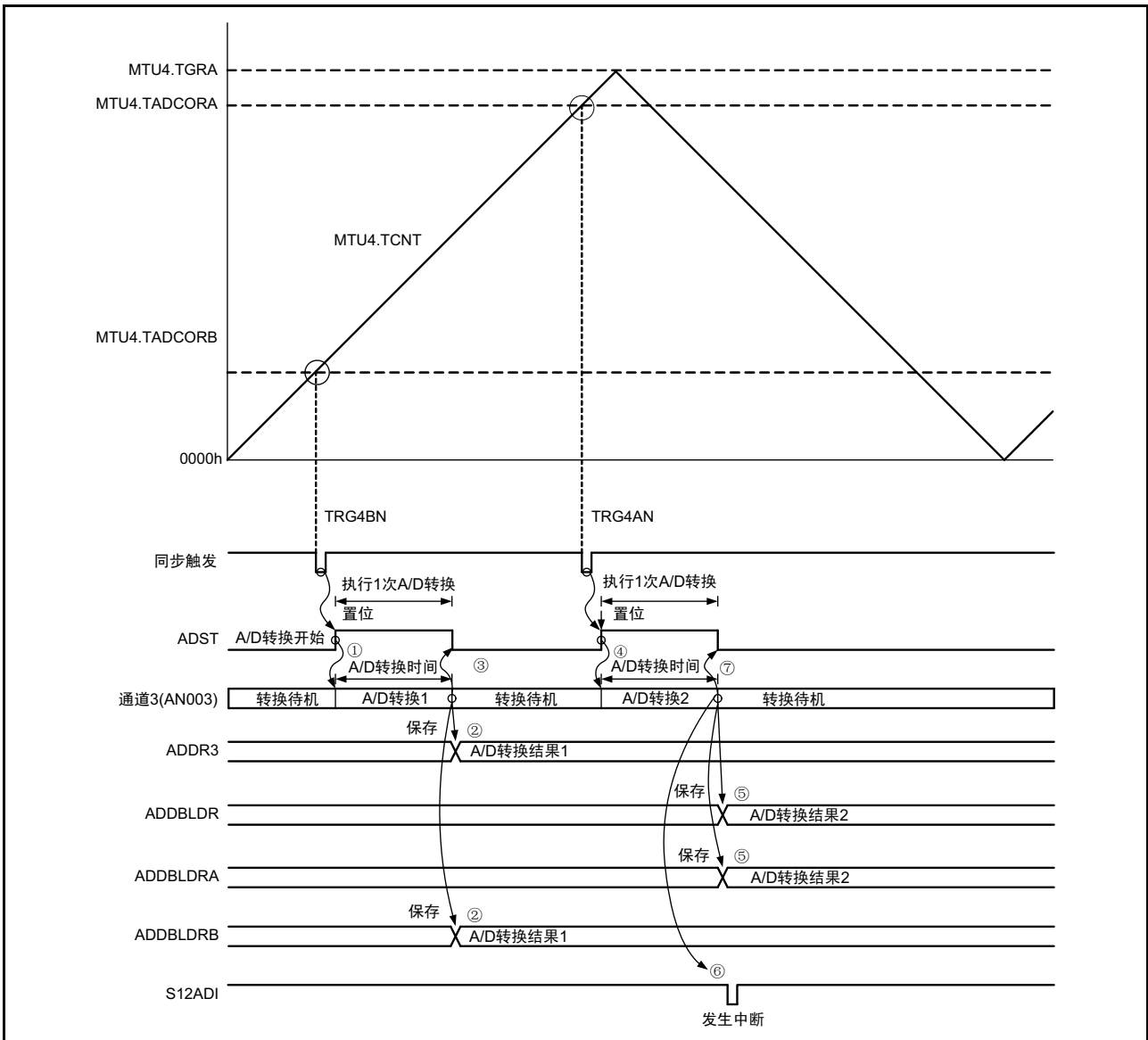


图 31.10 双触发模式的扩展运行例子 (2)
 (选择 AN003 双重化、选择 TRG4AN 或者 TRG4BN、并且第 1 次触发为 TRG4BN 的情况)

选择同步触发 GTADTRA0N 或者 GTADTRB0N 作为 A/D 转换开始触发，并且 2 种触发源同时产生的运行如下所示。

1. 如果通过第 1 次同时输入的 2 种触发源 GTADTRA0N、GTADTRB0N，将 ADCSR.ADST 位置“1”（开始 A/D 转换），就开始对 ADCSR.DBLANS[4:0] 位选择的组 A 的 1 个通道进行第 1 次 A/D 转换。
2. 如果 1 个通道的 A/D 转换结束，就将 A/D 转换结果保存到 A/D 数据双重化寄存器 B (ADDBLDRB) 和对应的 A/D 数据寄存器 (ADDRy)。
3. 自动清除 ADST 位，并且 12 位 A/D 转换器进入待机状态。此时，与 ADCSR.ADIE 位（允许由扫描结束引起的 S12ADI 中断）的设定无关，不发生 S12ADI 中断。
4. 如果通过第 2 次同时输入的 2 种触发源 GTADTRA0N、GTADTRB0N，将 ADCSR.ADST 位置“1”（开始 A/D 转换），就开始对 ADCSR.DBLANS[4:0] 位选择的组 A 的 1 个通道进行第 2 次 A/D 转换。
5. 如果 A/D 转换结束，就将 A/D 转换结果保存到 A/D 数据双重化寄存器 B (ADDBLDRB) 和 A/D 数据双重化寄存器 (ADDBLDR)。
6. 如果 ADCSR.ADIE 位已被置“1”（允许由扫描结束引起的 S12ADI 中断），就产生 S12ADI 中断请求。
7. ADST 位在 A/D 转换过程中保持“1”（开始 A/D 转换），一旦 A/D 转换结束就自动清除，并且 12 位 A/D 转换器进入待机状态。

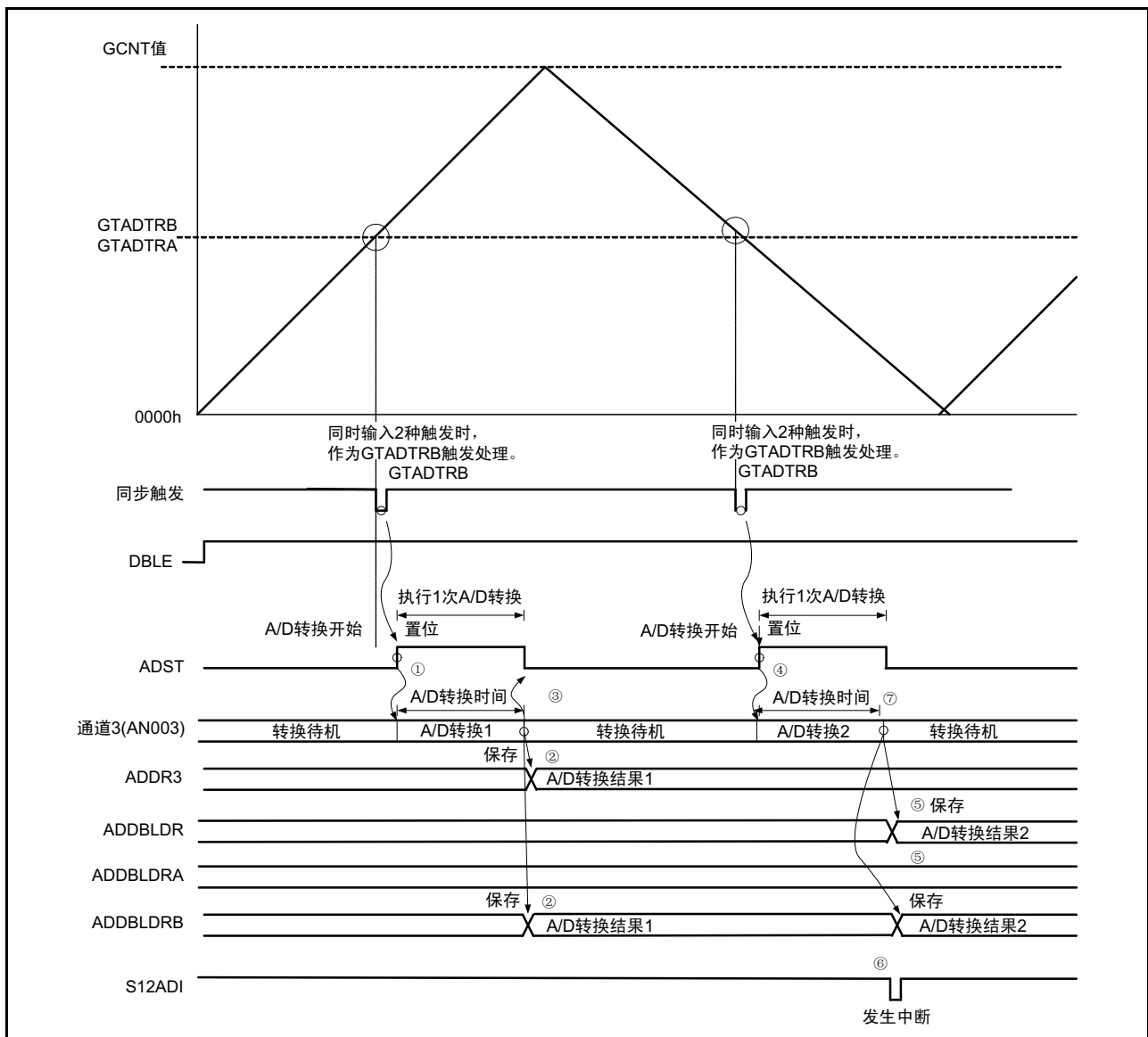


图 31.11 双触发模式的运行例子
(选择 AN003 双重化、选择 GTADTRA0N 或者 GTADTRB0N、并且 2 种触发源同时产生的情况)

31.3.3 连续扫描模式

31.3.3.1 基本运行 (无通道专用采样 & 保持电路)

连续扫描模式的基本运行是对所选通道的模拟输入重复进行 A/D 转换，如下所示。

1. 如果通过软件触发、同步触发 (MTU3、GPT) 或者异步触发的输入将 ADCSR.ADST 位置“1” (开始 A/D 转换)，就从 ADANSA 寄存器选择的 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
2. 如果 1 个通道的 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. 在所选通道的 A/D 转换全部结束后，如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断)，就产生 S12ADI 中断请求。另外，12 位 A/D 转换器继续从 ADANSA 寄存器选择的 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
4. 不自动清除 ADST 位，而在此位为“1” (开始 A/D 转换) 的期间重复步骤 2~3。如果将 ADST 位置“0” (停止 A/D 转换)，就中止 A/D 转换并且 12 位 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)，就重新开始从 ADANSA 寄存器选择的 ANn 中按照 n 从小到大的顺序进行 A/D 转换。

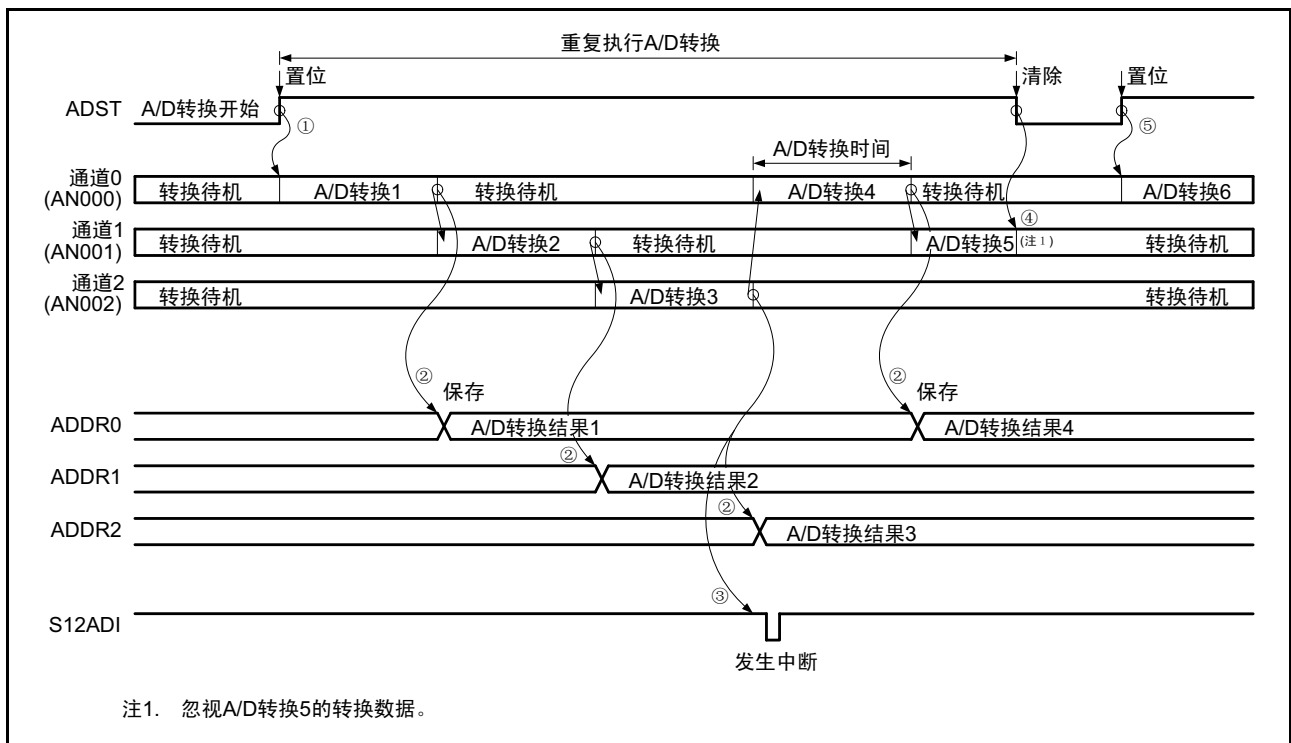


图 31.12 连续扫描模式的运行例子 (基本运行: 选择 AN000 ~ AN002)

31.3.3.2 基本运行 (有通道专用采样 & 保持电路)

如果使用通道专用采样 & 保持电路, 就在执行采样 & 保持后, 对选择的全部通道的模拟输入进行连续的 A/D 转换, 如下所示。通过 ADSHCR.SHANS[2:0] 位选择使用通道专用采样 & 保持电路的通道。

1. 如果通过软件触发或者同步触发 (MTU3、GPT) 的输入将 ADCSR.ADST 位置“1” (开始 A/D 转换), 就开始对使用通道专用采样 & 保持电路的全部通道的模拟输入进行采样。
2. 在执行采样 & 保持后, 从 ADANSA 寄存器选择的 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
3. 如果 1 个通道的 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)。
4. 在所选通道的 A/D 转换全部结束后, 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断), 就产生 S12ADI 中断请求。另外, 开始对使用通道专用采样 & 保持电路的通道的模拟输入全部进行采样。
5. 不自动清除 ADST 位, 而在此位为“1” (开始 A/D 转换) 的期间重复步骤 2~4。如果将 ADST 位置“0” (停止 A/D 转换), 就中止 A/D 转换并且 12 位 A/D 转换器进入待机状态。
6. 此后, 如果将 ADST 位置“1” (开始 A/D 转换), 就重新开始对使用通道专用采样 & 保持电路的全部通道的模拟输入进行采样。

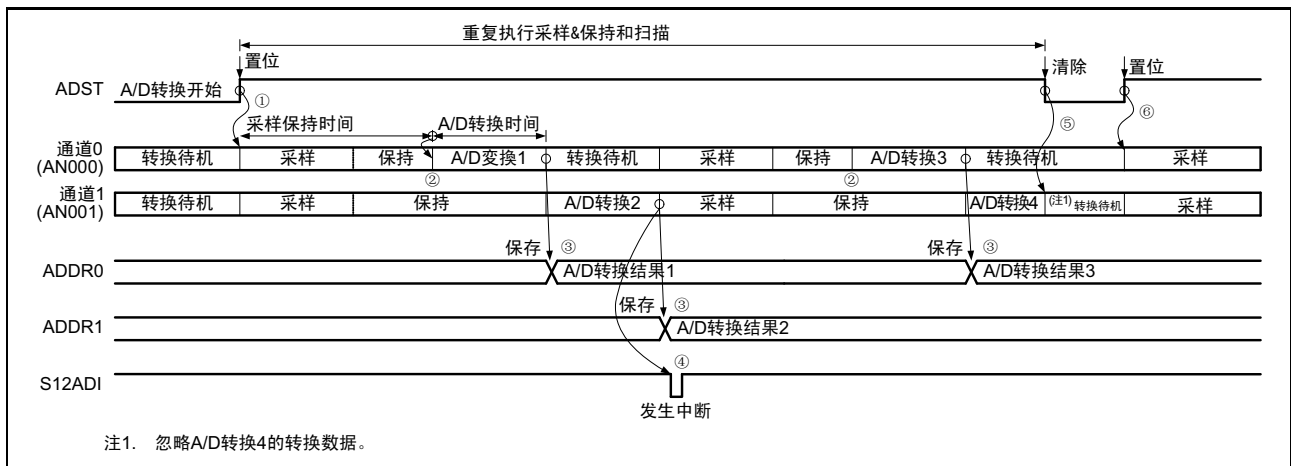


图 31.13 连续扫描模式的运行例子 (有通道专用采样 & 保持电路)

31.3.3.3 通道选择和自诊断 (无通道专用采样 & 保持电路)

如果在选择通道的同时选择自诊断, 就对提供给 12 位 A/D 转换器的基准电压 VREFH0 (×0、×1/2、×1 中的任意 1 个) 进行 A/D 转换, 之后对所选通道的模拟输入重复进行 A/D 转换, 如下所示。

1. 如果通过软件触发、同步触发 (MTU3、GPT) 或者异步触发的输入将 ADCSR.ADST 位置“1” (开始 A/D 转换), 就最先开始自诊断的 A/D 转换。
2. 如果自诊断的 A/D 转换结束, 就将 A/D 转换结果保存到 A/D 自诊断数据寄存器 (ADRD), 然后从 ADANSA 寄存器选择的 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
3. 如果 1 个通道的 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)。
4. 在所选通道的 A/D 转换全部结束后, 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断), 就产生 S12ADI 中断请求。另外, 12 位 A/D 转换器继续自诊断的 A/D 转换, 并且在转换结束后从 ADANSA 寄存器选择的 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
5. 不自动清除 ADST 位, 而在此位为“1” (开始 A/D 转换) 的期间重复步骤 2~4。如果将 ADST 位置“0” (停止 A/D 转换), 就中止 A/D 转换并且 12 位 A/D 转换器进入待机状态。
6. 此后, 如果将 ADST 位置“1” (开始 A/D 转换), 就重新开始自诊断的 A/D 转换。

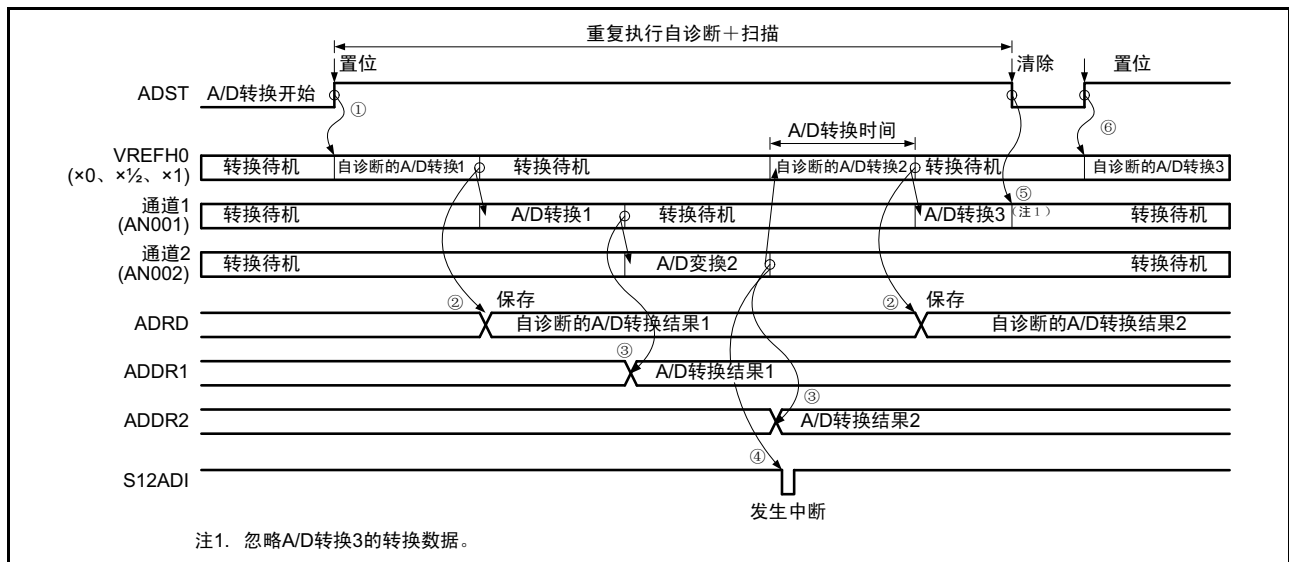


图 31.14 连续扫描模式的运行例子 (基本运行和自诊断)

31.3.3.4 通道选择和自诊断 (有通道专用采样 & 保持电路)

如果在选择通道的同时选择自诊断, 并且使用通道专用采样 & 保持电路, 就在执行采样 & 保持后, 对提供 12 位 A/D 转换器的基准电压 VREFH0 (x0、x1/2、x1 中的任意 1 个) 进行 A/D 转换, 之后对所选通道的模拟输入重复进行 A/D 转换, 如下所示。

1. 如果通过软件触发、同步触发 (MTU3、GPT) 或者异步触发的输入将 ADCSR.ADST 位置“1” (开始 A/D 转换), 就开始对使用通道专用采样 & 保持电路的全部通道的模拟输入进行采样。
2. 在执行采样 & 保持后, 开始自诊断的 A/D 转换。
3. 如果自诊断的 A/D 转换结束, 就将 A/D 转换结果保存到 A/D 自诊断数据寄存器 (ADDRD), 然后从 ADANSA 寄存器选择的 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
4. 如果 1 个通道的 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)。
5. 在所选通道的 A/D 转换全部结束后, 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断), 就产生 S12ADI 中断请求。另外, 开始对使用通道专用采样 & 保持电路的全部通道的模拟输入进行采样。
6. 不自动清除 ADST 位, 而在此位为“1” (开始 A/D 转换) 的期间重复步骤 2~5。如果将 ADST 位置“0” (停止 A/D 转换), 就中止 A/D 转换并且 12 位 A/D 转换器进入待机状态。
7. 此后, 如果将 ADST 位置“1” (开始 A/D 转换), 就重新开始对使用通道专用采样 & 保持电路的全部通道的模拟输入进行采样。

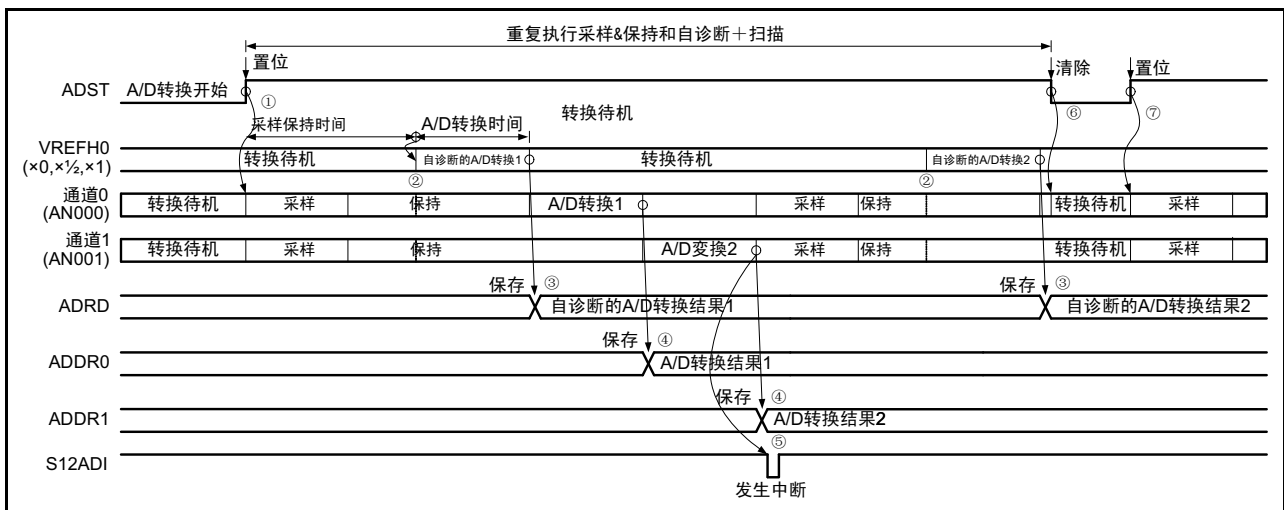


图 31.15 连续扫描模式的运行例子 (通道专用采样 & 保持电路运行和自诊断)

31.3.4 分组扫描模式

31.3.4.1 基本运行

分组扫描模式的基本运行是以 MTU3 触发或者 GPT 触发作为扫描开始条件，并且对组 A 和组 B 分别选择的全部通道的模拟输入只进行 1 次 A/D 转换，如下所示。组 A 和组 B 的扫描运行与 1 个周期扫描模式中的运行相同。

对于分组扫描模式中的触发设定，通过 ADSTRGR.TRSA[4:0] 位选择组 A 的触发，通过 ADSTRGR.TRSB[4:0] 位选择组 B 的触发。必须给组 A 和组 B 选择不同的触发，以避免同时发生组 A 和组 B 的 A/D 转换。另外，不能使用软件触发。

对于 A/D 转换的对象通道，通过 ADANSA 寄存器选择组 A 的通道，通过 ADANSB 寄存器选择组 B 的通道。组 A 和组 B 不能选择相同的通道。

在分组扫描模式中选择自诊断时，组 A 和组 B 分别执行自诊断。

通过 MTU3 触发进行分组扫描模式的运行例子如下所示。设定为组 A 通过 MTU3 的 TRG4AN 触发开始转换，组 B 通过 MTU3 的 TRG4BN 触发开始转换。

1. 通过 MTU3 的 TRG4AN 触发开始组 A 的扫描。
2. 在组 A 的扫描结束时，如果 ADCSR.ADIE 位已被置“1”（允许由扫描结束引起的 S12ADI 中断），就发生 S12ADI 中断。
3. 通过 MTU3 的 TRG4BN 触发开始组 B 的扫描。
4. 在组 B 的扫描结束时，如果 ADCSR.GBADIE 位已被置“1”（允许由扫描结束引起的 S12GBADI 中断），就发生 S12GBADI 中断。

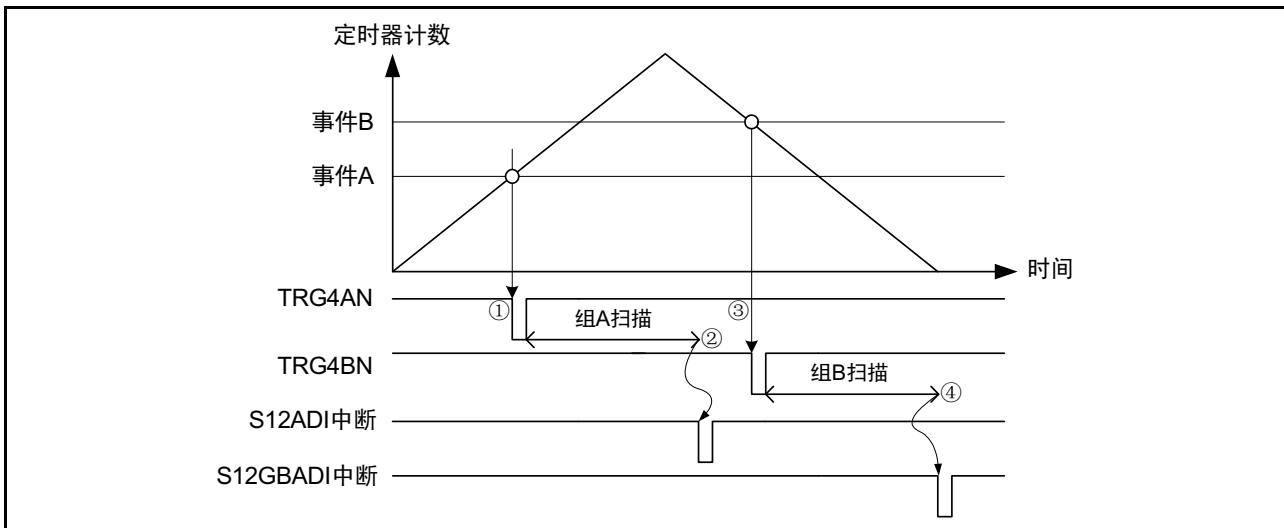


图 31.16 分组扫描模式的运行例子（通过 MTU3 触发进行的基本运行）

31.3.4.2 选择双触发模式时的运行

在分组扫描模式中选择双触发模式时，组 A 执行 2 次连串的通过 MTU3 触发或者 GPT 触发开始的 1 个周期扫描模式，组 B 的运行与通过 MTU3 触发或者 GPT 触发开始的 1 个周期扫描模式的运行相同。

对于分组扫描模式中的触发设定，通过 ADSTRGR.TRSA[4:0] 位选择组 A 的触发，通过 ADSTRGR.TRSB[4:0] 位选择组 B 的触发。必须给组 A 和组 B 选择不同的触发，以避免同时发生组 A 和组 B 的 A/D 转换。另外，不能使用软件触发和异步触发 (ADTRG0#)。

对于 A/D 转换的对象通道，通过 ADCSR.DBLANS[4:0] 位选择组 A 的通道，通过 ADANSB 寄存器选择组 B 的通道。组 A 和组 B 不能选择相同的通道。

在分组扫描模式中选择双触发模式时，不能选择自诊断。

如果将双重化的通道号设定到 ADCSR.DBLANS[4:0] 位，并且将 ADCSR.DBLE 位置“1”，A/D 转换数据的双重化就有效。

在通过 MTU3 触发开始的分组扫描模式中选择双触发模式的运行例子如下所示。设定为组 A 通过 MTU3 的 TRG4ABN 触发开始转换，组 B 通过 MTU3 的 TRG0AN 触发开始转换。

1. 通过 MTU3 的 TRG0AN 触发开始组 B 的扫描。
2. 在组 B 的扫描结束时，如果 ADCSR.GBADIE 位已被置“1”（允许由扫描结束引起的 S12GBADI 中断），就发生 S12GBADI 中断。
3. 通过 MTU3 的第 1 次 TRG4ABN 触发开始组 A 的第 1 次扫描。
4. 在组 A 的第 1 次扫描结束时，将转换数据保存到 ADDRy 寄存器，并且与 ADCSR.ADIE 位的设定无关，不产生 S12ADI 中断请求。
5. 通过 MTU3 的第 2 次 TRG4ABN 触发开始组 A 的第 2 次扫描。
6. 在组 A 的第 2 次扫描结束时，将转换数据保存到 ADDBLDR 寄存器，并且如果 ADCSR.ADIE 位已被置“1”（允许由扫描结束引起的 S12ADI 中断），就产生 S12ADI 中断请求。

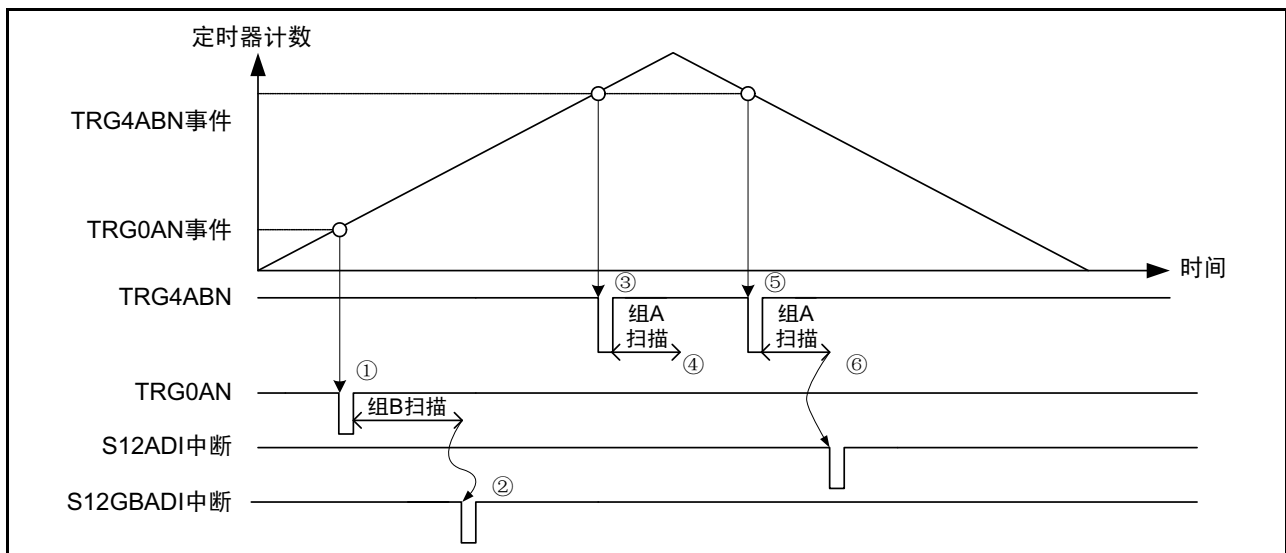


图 31.17 在分组扫描模式中选择双触发模式时的运行例子（通过 MTU3 触发进行的基本运行）

31.3.4.3 使用软件触发时的注意事项

如果在双触发模式中输入软件触发，就对所选通道进行扫描。如果将 ADCSR.ADIE 位置“1”（允许由扫描结束引起的 S12ADI 中断），就与第 1 次扫描、第 2 次扫描无关，发生 S12ADI 中断。即使通过软件触发开始的扫描是第 2 次扫描，也不进行数据的双重化。使用软件触发时的双触发运行注意例子如图 31.18 所示。

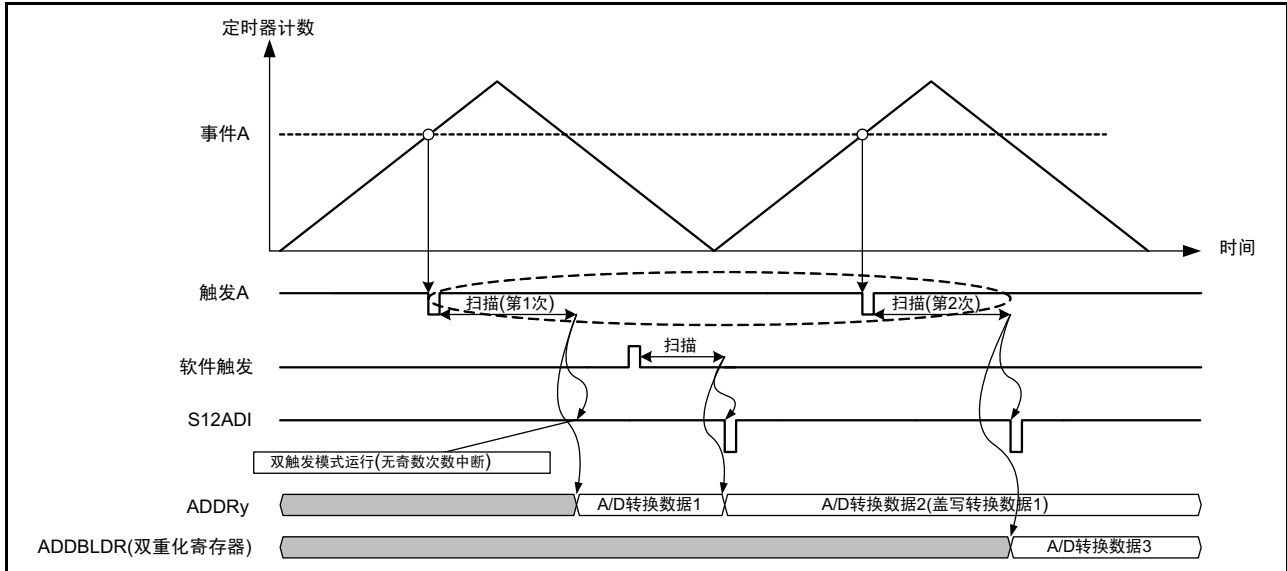


图 31.18 使用软件触发时的双触发运行注意例子

31.3.4.4 组 A 优先控制运行

如果在分组扫描模式中将 A/D 分组扫描模式优先控制寄存器 (ADGSPCR) 的 PGS 位置“1”，就进行组 A 优先控制运行。必须按照图 31.19 记载的步骤，进行 ADGSPCR 寄存器的 PGS 位置“1”的设定。如果进行流程图以外的设定，就不保证 A/D 转换的运行和保存的数据。

在分组扫描模式的基本运行中，即使在组 A 的 A/D 转换运行中输入组 B 的触发，或者在组 B 的 A/D 转换运行中输入组 A 的触发，都将忽视被输入的触发。在组 A 优先控制运行中，如果在组 B 的 A/D 转换运行中输入组 A 的触发，就中止组 B 的 A/D 转换运行，并且开始组 A 的 A/D 转换运行。当 ADGSPCR.GBRSCN 位为“0”时，在组 A 的 A/D 转换运行结束后进入待机状态。当 ADGSPCR.GBRSCN 位为“1”时，在组 A 的 A/D 转换运行结束后，自动从扫描的起始开始重新进行组 B 的 A/D 转换运行。ADGSPCR.GBRSCN 位的设定和 A/D 转换运行中输入触发时的运行如表 31.8 所示。

组 A 和组 B 的扫描运行与 1 个周期扫描模式的运行相同。另外，如果将 ADGSPCR.GBRP 位置“1”，组 B 的扫描运行就为连续执行 1 个周期扫描的运行。

在进行分组扫描模式的触发设定时，通过 ADSTRGR.TRSA[4:0] 位选择组 A 的同步触发，通过 ADSTRGR.TRSB[4:0] 位选择与组 A 的触发不同的组 B 的同步触发。如果要将 ADGSPCR.GBRP 位设定为“1”，就必须将 ADSTRGR.TRSB[4:0] 位设定为“1Fh”。另外，通过 ADANSA 寄存器选择组 A 的 A/D 转换对象通道，通过 ADANSB 寄存器选择不同与组 A 的组 B 的 A/D 转换对象通道。

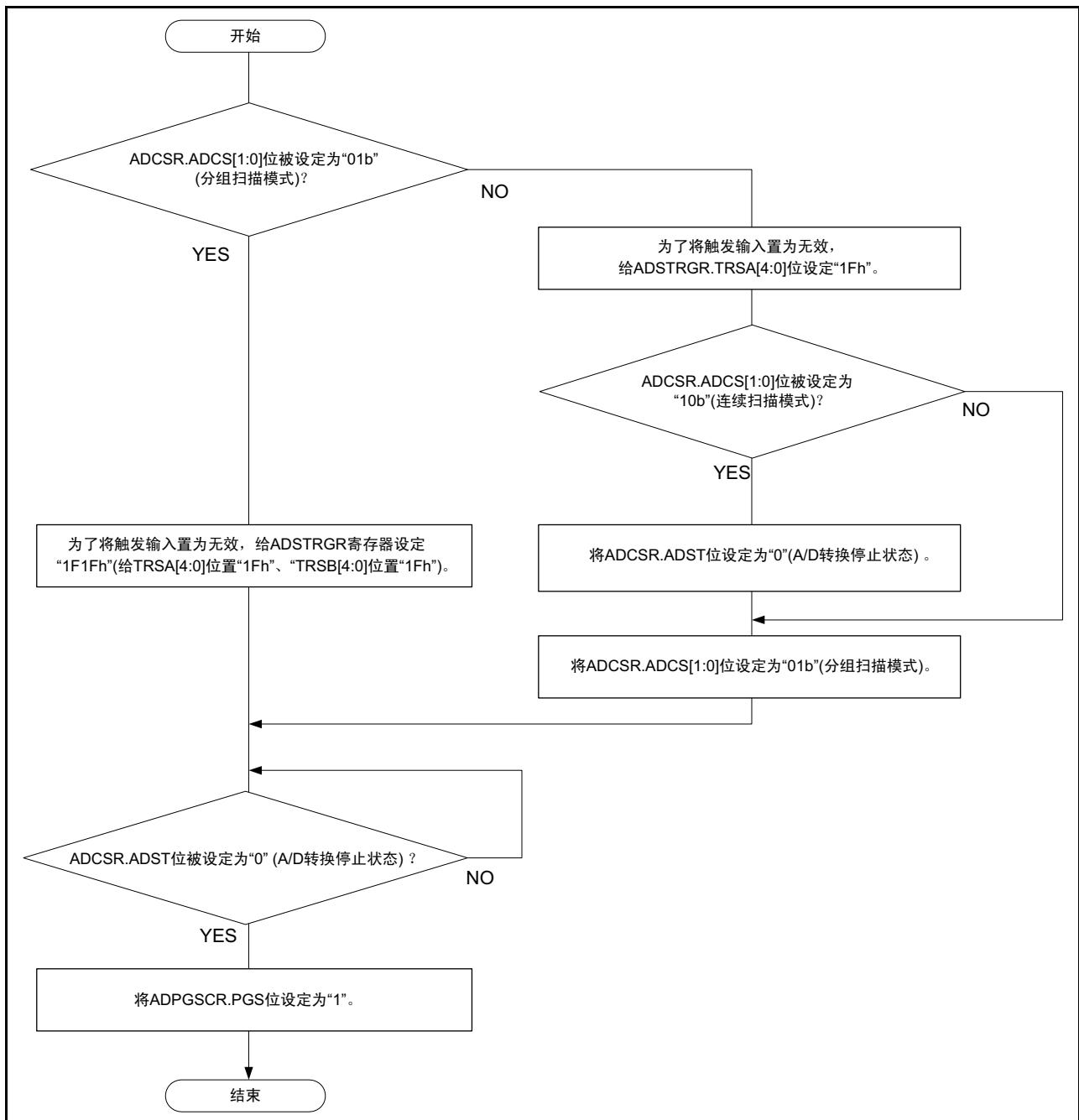


图 31.19 设定 ADPGSCR.PGS 位时的流程

表 31.8 根据 ADGSPCR.GBRSCN 位的设定进行的 A/D 转换运行控制

	触发输入	ADGSPCR.GBRSCN=0	ADGSPCR.GBRSCN=1
在组 A 的 A/D 转换运行过程中	组 A 的触发输入	触发输入无效	触发输入无效
	组 B 的触发输入	触发输入无效	在组 A 的 A/D 转换运行结束后, 进行组 B 的 A/D 转换运行。但是, 在连续输入组 A 的触发时, 组 B 的重新扫描运行因组 A 而被取消, 因此不进行组 B 的重新扫描。(注 1)
在组 B 的 A/D 转换运行过程中	组 A 的触发输入	中止组 B 的 A/D 转换运行, 开始组 A 的 A/D 转换运行。	<ul style="list-style-type: none"> 中止组 B 的 A/D 转换运行, 开始组 A 的 A/D 转换运行。 在组 A 的 A/D 转换运行结束后, 开始组 B 的 A/D 转换运行。
	组 B 的触发输入	触发输入无效	触发输入无效

注 1. 为了保证组 B 的重新扫描运行, 必须确保组 A 的最小触发间隔为 2 个扫描 (将组 A 的扫描时间 (tSCAN[GrA]) 作为 1 个扫描, 组 B 的扫描时间 (tSCAN[GrB]) 作为 1 个扫描, 共 2 个扫描) 的间隔; 组 B 的最小触发间隔为 3 个扫描 (将组 A 的扫描时间 (tSCAN[GrA]) 作为 1 个扫描, 组 B 的扫描时间 (tSCAN[GrB]) 作为 2 个扫描, 共 3 个扫描) 的间隔。

如果通过启动重新扫描进行的组 B 的 A/D 转换运行和组 A 的 A/D 转换运行重叠, 就取消重新扫描。另外, 如果通过重新扫描启动组 B 和通过触发请求启动组 B 重叠, 触发请求就变为无效, 因此也必须确保上述的触发间隔。

给组 A 选择通道 0, 组 B 选择通道 1 ~ 3 的分组扫描模式的组 A 优先控制运行例子 (ADGSPCR.GBRSCN=1、ADGSPCR.GBRP=0 时), 如下所示。

1. 如果通过组 B 的触发输入将 ADCSR.ADST 位置“1” (开始 A/D 转换), 就从 ADANSB 寄存器选择的通道 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
2. 如果 1 个通道的 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. 如果在组 B 的 A/D 转换运行过程中输入组 A 的触发, ADCSR.ADST 位就清“0”, 并且中止组 B 的 A/D 转换运行。此后, 如果 ADCSR.ADST 位变为“1” (开始 A/D 转换), 就从 ADANSA 寄存器选择的通道 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
4. 如果 1 个通道的 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)。
5. 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断), 就产生 S12ADI 中断请求。
6. 如果在自动清除 ADST 位后, ADST 位再次自动变为“1” (开始 A/D 转换), 就从 ADANSB 寄存器选择的通道 ANn 中按照 n 从小到大的顺序重新开始 A/D 转换。
7. 如果 1 个通道的 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)。
8. 如果 ADCSR.GBADIE 位已被置“1” (允许由扫描结束引起的 S12GBADI 中断), 就产生 S12GBADI 中断请求。
9. ADST 位在 A/D 转换过程中保持“1” (开始 A/D 转换), 一旦 A/D 转换结束就自动清除, 并且 12 位 A/D 转换器进入待机状态。

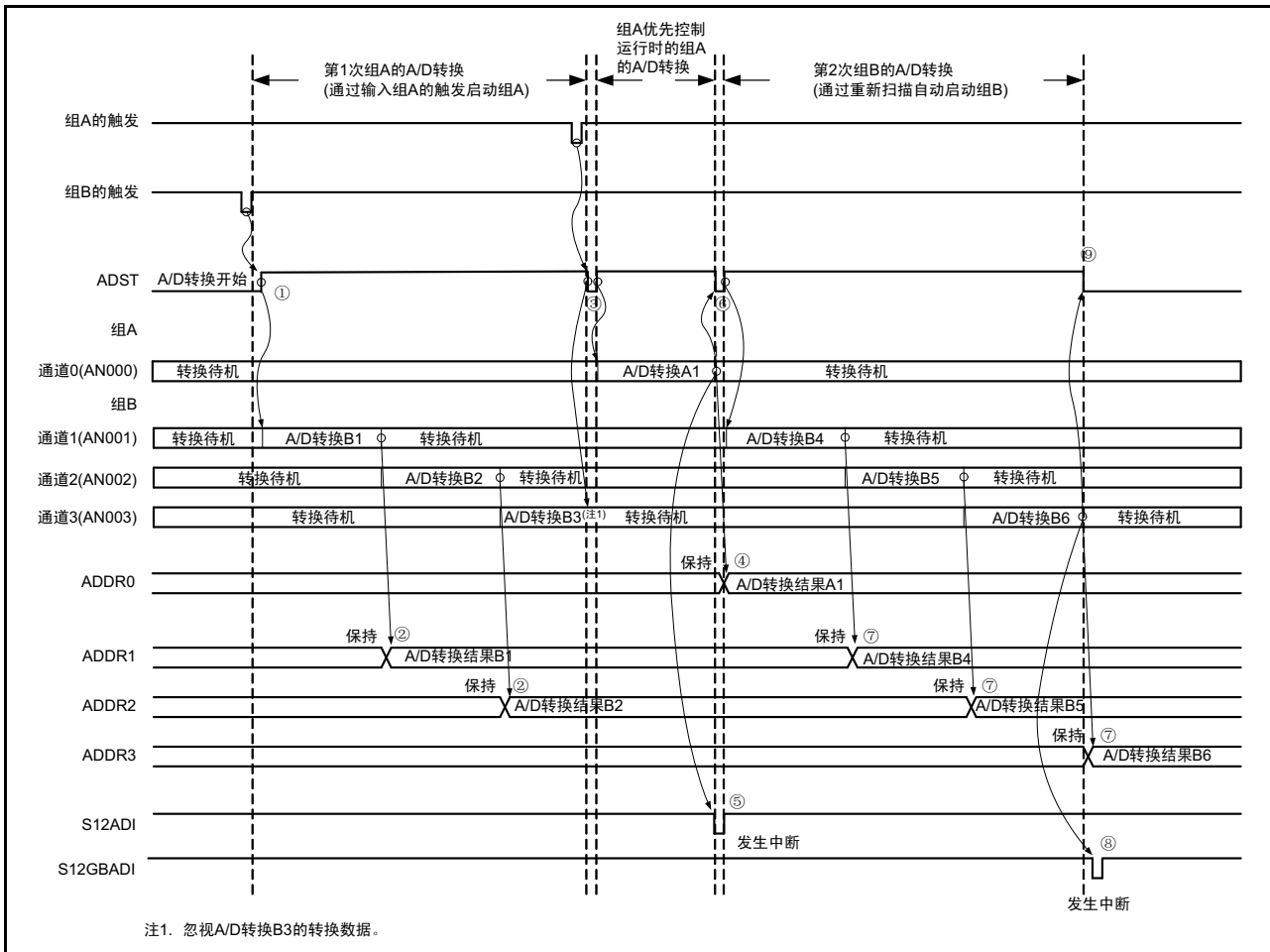


图 31.20 组 A 优先控制的运行例子 (ADGSPCR.GBRSCN=1、ADGSPCR.GBRP=0 时的运行) (1)

在组 A 优先控制运行 (ADGSPCR.GBRSCN=1、ADGSPCR.GBRP=0) 时, 给组 A 选择通道 0, 组 B 选择通道 1 ~ 3, 并且在组 A 的 A/D 转换运行中输入组 B 的触发时的重新扫描运行例子, 如下所示。

1. 如果通过组 A 的触发输入将 ADCSR.ADST 位置“1” (开始 A/D 转换), 就从 ADANSA 寄存器选择的组 A 的通道 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
2. 如果 1 个通道的 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. 如果在组 A 的 A/D 转换运行过程中输入组 B 的触发, 就在组 A 的 A/D 转换运行结束后进入能进行组 B 的 A/D 转换运行的状态。(但是, 在连续输入组 A 的触发时, 组 B 的重新扫描运行因组 A 而被取消, 因此, 不进行组 B 的重新扫描。)
4. 在组 A 的扫描结束后, 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断), 就产生 S12ADI 中断请求。
5. 在组 A 的扫描结束后, 通过启动组 B 的重新扫描, ADCSR.ADST 位自动设定为“1”。此后, 从 ADANSB 寄存器选择的组 B 的通道 ANn 中按照 n 从小到大的顺序重新开始 A/D 转换。
6. 如果 1 个通道的 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)。
7. 在通过启动重新扫描进行的组 B 的扫描结束后, 如果 ADCSR.GBADIE 位已被置“1” (允许由扫描结束引起的 S12GBADI 中断), 就产生 S12GBADI 中断请求。
8. ADST 位在 A/D 转换过程中保持“1” (开始 A/D 转换), 一旦 A/D 转换结束就自动清除, 并且 12 位 A/D 转换器进入待机状态。

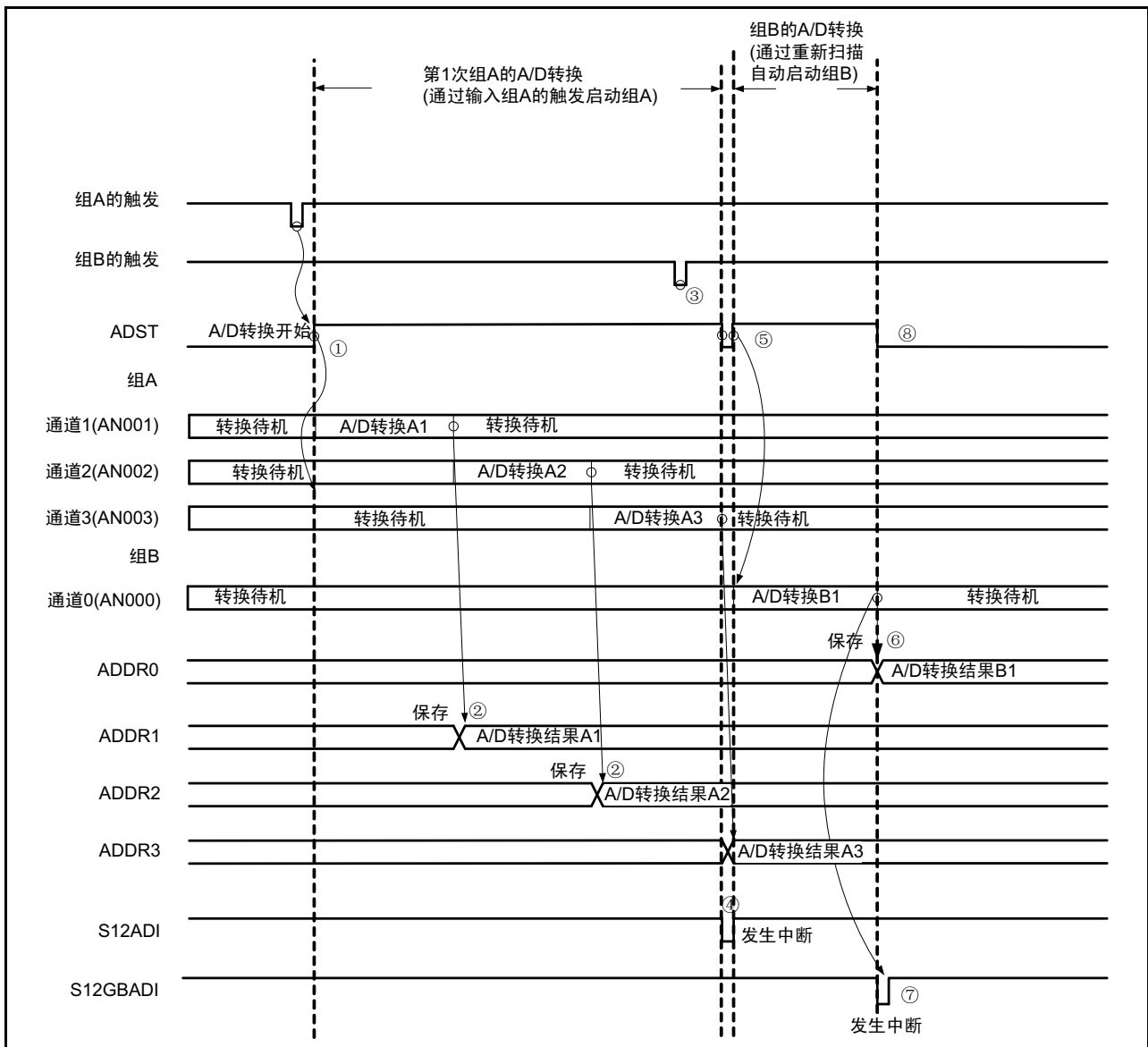


图 31.21 组 A 优先控制的运行例子 (ADGSPCR.GBRSCN=1、ADGSPCR.GBRP=0 时的运行) (2)

在组 A 优先控制运行 (ADGSPCR.GBRSCN=1、ADGSPCR.GBRP=0) 时, 给组 A 选择通道 0, 组 B 选择通道 1 ~ 3, 并且不能进行重新扫描的运行例子, 如下所示。

1. 如果通过组 B 的触发输入将 ADCSR.ADST 位置“1” (开始 A/D 转换), 就从 ADANSB 寄存器选择的组 B 的通道 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
2. 如果 1 个通道的 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. 如果在组 B 的 A/D 转换运行过程中输入组 A 的触发, ADCSR.ADST 位就清“0” (停止 A/D 转换), 并且中止组 B 的 A/D 转换运行。
4. 此后, 如果 ADCSR.ADST 位自动变为“1” (开始 A/D 转换), 就从 ADANSA 寄存器选择的组 A 的通道 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
5. 如果 1 个通道的 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)。
6. 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断), 就产生 S12ADI 中断请求。
7. 如果 ADGSPCR.GBRSCN 位已被置“1” (重新扫描运行有效), 就在组 A 的 A/D 转换运行结束后, 通过组 B 的重新扫描运行, 使 ADCSR.ADST 位自动变为“1”。此后, 从 ADANSB 寄存器选择的组 B 的通道 ANn 中按照 n 从小到大的顺序重新开始 A/D 转换。
8. 如果 1 个通道的 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)。
9. 如果在通过启动重新扫描进行的组 B 的 A/D 转换运行过程中, 输入组 A 的触发, ADCSR.ADST 位就清“0” (停止 A/D 转换), 并且中止组 B 的 A/D 转换运行。
10. 此后, 如果 ADCSR.ADST 位自动变为“1” (开始 A/D 转换), 就从 ADANSA 寄存器选择的组 A 的通道 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
11. 如果 1 个通道的 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)。
12. 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断), 就产生 S12ADI 中断请求。
13. 此后, ADST 位自动清“0”, 并且停止 A/D 转换。而因组 A 的 A/D 转换而被取消的组 B 的重新扫描运行也不再执行。

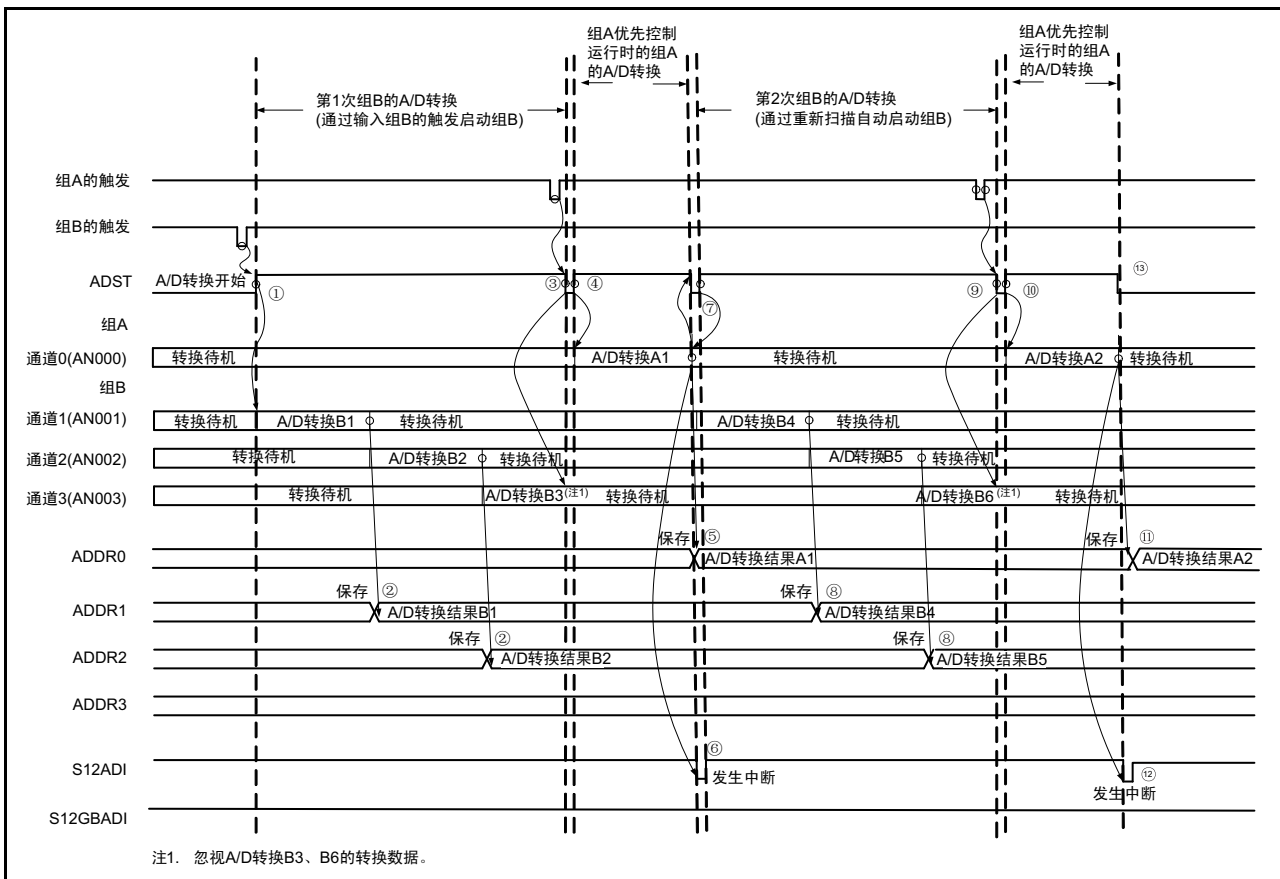


图 31.22 组 A 优先控制的运行例子 (ADGSPCR.GBRSCN=1、ADGSPCR.GBRP=0 时的运行) (3)

给组 A 选择通道 0, 组 B 选择通道 1 ~ 3 时的组 A 优先控制运行例子 (ADGSPCR.GBRSCN=0、ADGSPCR.GBRP=0 时), 如下所示。

1. 如果通过组 B 的触发输入将 ADCSR.ADST 位置“1” (开始 A/D 转换), 就从 ADANSB 寄存器选择的通道 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
2. 如果 1 个通道的 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. 如果在组 B 的 A/D 转换运行过程中输入组 A 的触发, ADCSR.ADST 位就清“0”, 并且中止组 B 的 A/D 转换运行。此后, 如果 ADCSR.ADST 位自动变为“1” (开始 A/D 转换), 就从 ADANSA 寄存器选择的通道 ANn 中按照 n 从小到大的顺序开始 A/D 转换。
4. 如果 1 个通道的 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)。
5. 如果 ADCSR.ADIE 位已被置“1” (允许由扫描结束引起的 S12ADI 中断), 就产生 S12ADI 中断请求。
6. ADST 位在 A/D 转换过程中保持“1” (开始 A/D 转换), 一旦 A/D 转换结束就自动清除, 并且 12 位 A/D 转换器进入待机状态。

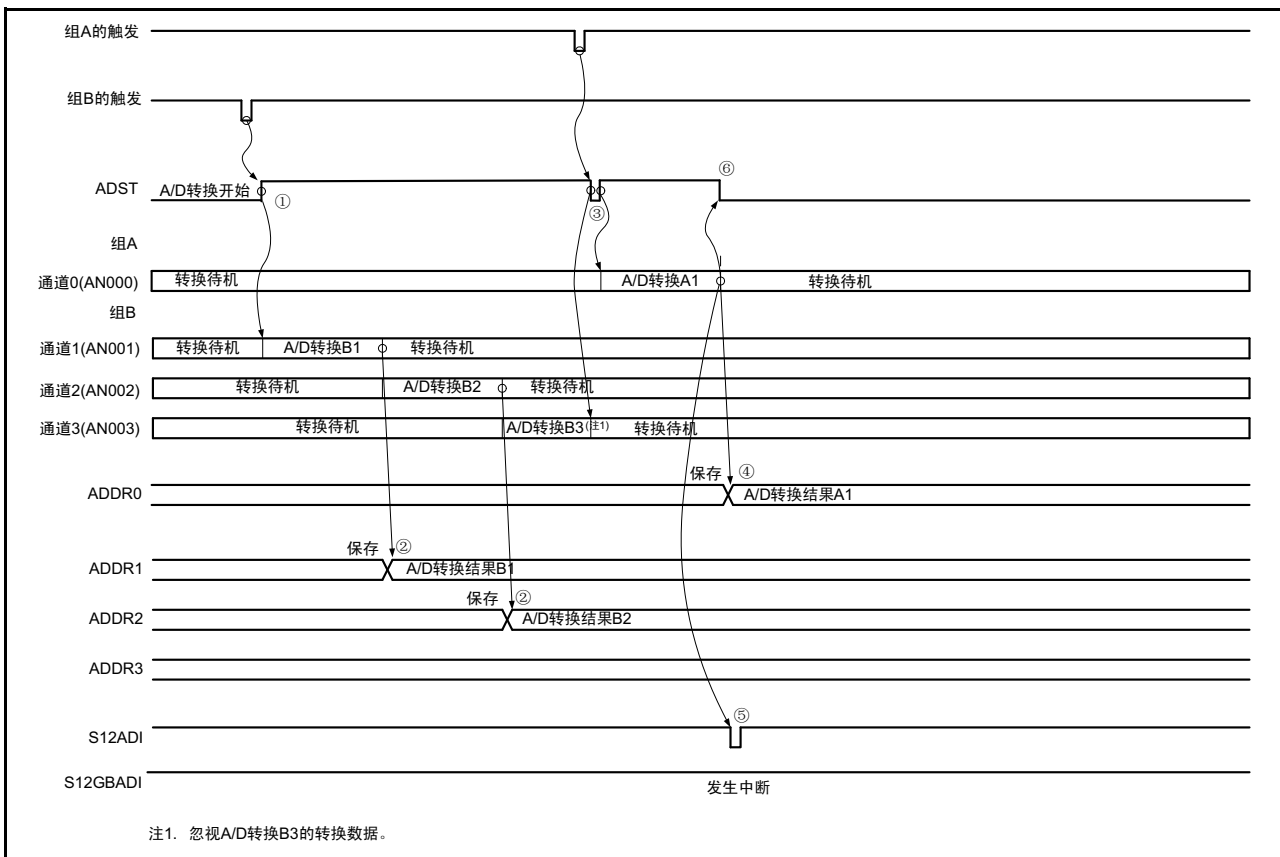


图 31.23 组 A 优先控制的运行例子 (ADGSPCR.GBRSCN=0、ADGSPCR.GBRP=0 时的运行)

给组 A 选择通道 0, 组 B 选择通道 1 ~ 3 时的组 A 优先控制运行例子 (ADGSPCR.GBRP=1), 如下所示。

1. 如果将ADGSPCR.GBRP置“1”, ADCSR.ADST位就被置“1”(开始A/D转换), 从ADANSB寄存器选择的通道ANn中按照n从小到大的顺序开始A/D转换。
2. 如果1个通道的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. 如果在组B的A/D转换运行过程中输入组A的触发, ADCSR.ADST位就清“0”, 并且中止组B的A/D转换运行。此后, 如果ADCSR.ADST位自动变为“1”(开始A/D转换), 就从ADANSA寄存器选择的通道ANn中按照n从小到大的顺序开始A/D转换。
4. 如果1个通道的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)。
5. 如果ADCSR.ADIE位已被置“1”(允许由扫描结束引起的S12ADI中断), 就产生S12ADI中断请求。
6. 如果在自动清除ADST位后, ADCSR.ADST位再次自动变为“1”(开始A/D转换), 就重新开始从ADANSB寄存器选择的通道ANn中按照n从小到大的顺序进行A/D转换。
7. 如果1个通道的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)。
8. 如果ADCSR.GBADIE位已被置“1”(允许由扫描结束引起的S12GBADI中断), 就产生S12GBADI中断请求。
9. 如果在自动清除ADST位后, ADCSR.ADST位再次自动变为“1”(开始A/D转换), 就重新开始从ADANSB寄存器选择的通道ANn中按照n从小到大的顺序进行A/D转换。在ADGSPCR.GBRP位置“1”的期间, 重复步骤6~9的运行, 并且禁止清除ADCSR.ADST位。如果要在ADGSPCR.GBRP位为“1”时强制停止A/D转换, 必须按照“图31.32 通过ADCSR.ADST位执行软件清除的设定流程”来进行设定。

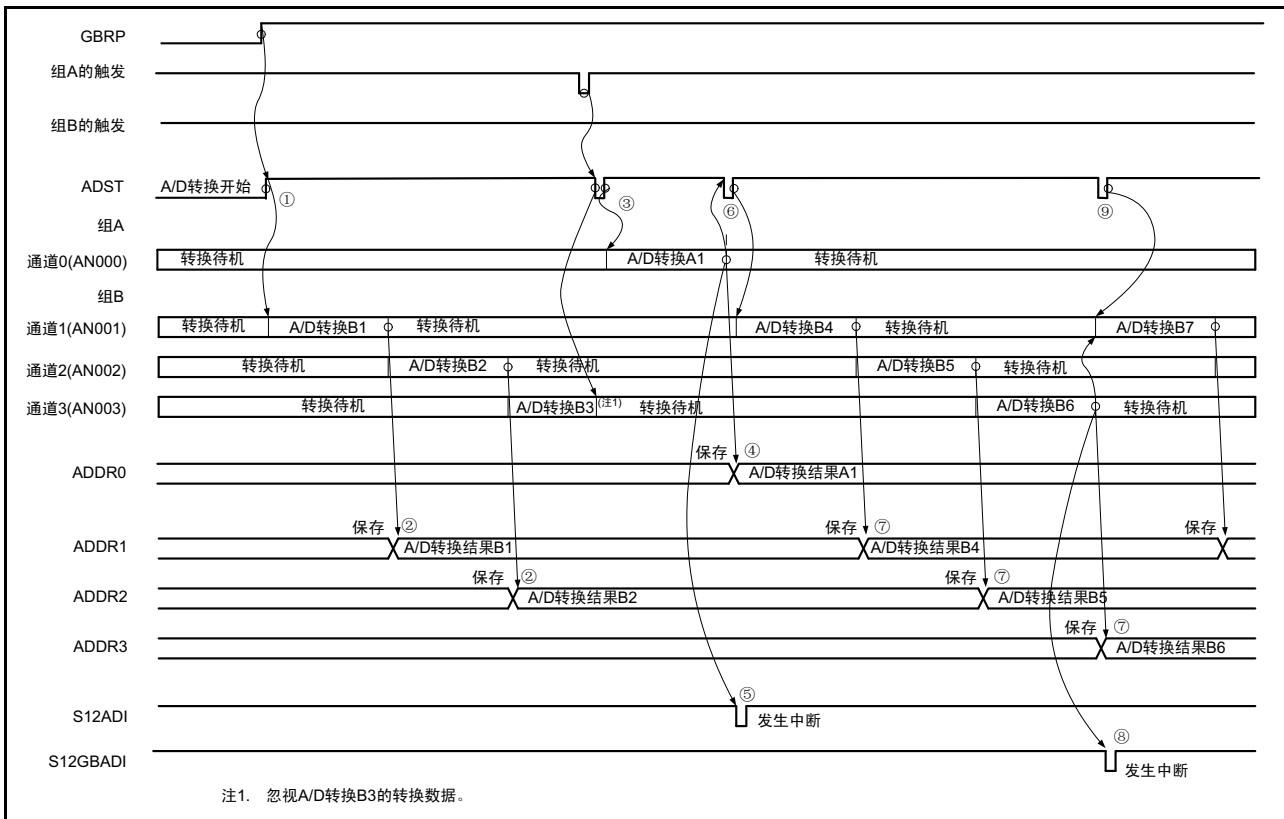


图 31.24 组 A 优先控制的运行例子 (ADGSPCR.GBRP=1 时的运行)

31.3.5 模拟输入的采样和扫描转换时间

能选择通过软件触发、MTU3 触发、GPT 触发或者 ADTRG0# (外部触发) 启动扫描转换。在扫描转换开始延迟时间 (t_D) 后, 执行通道专用采样 & 保持电路的采样、自诊断转换处理, 然后开始执行 A/D 转换处理。

在 1 个周期扫描模式中, 通过软件触发启动和通过 MTU3 触发、GPT 触发启动扫描转换的时序如图 31.25 所示。在 1 个周期扫描模式中, 通过 ADTRG0# (外部触发) 启动扫描转换的时序如图 31.26 所示。扫描转换时间 (t_{SCAN}) 包括扫描转换开始延迟时间 (t_D)、通道专用采样 & 保持电路采样时间 (t_{SH}) (注 1)、自诊断转换时间 (t_{DIAG}) (注 2)、A/D 转换处理时间 (t_{CONV})、通道专用采样 & 保持电路结束处理时间 (t_{SHED}) (注 3) 以及扫描转换结束延迟时间 (t_{ED})。A/D 转换处理时间 (t_{CONV}) 为输入采样时间 (t_{SPL}) 与逐次转换时间 (t_{SAM}) 的合计时间。

采样时间 (t_{SPL}) 是对 A/D 转换器的采样 & 保持电路进行电荷充电的时间。能在模拟输入的信号源阻抗高、采样时间不足, 或者 A/D 转换时钟 (ADCLK) 低速运行时, 通过 ADSSTR 寄存器对采样时间进行调整。

逐次转换时间 (t_{SAM}) 固定为 ADCLK 的 30 个状态。ADSSTR 寄存器的设定例子如表 31.9 所示。扫描转换时间如表 31.10 所示。

选择通道数 n 的 1 个周期扫描的扫描转换时间 (t_{SCAN}) 如下所示:

$$t_{SCAN} = t_D + t_{SPLSH} + t_{DIAG} + (t_{CONV} \times n) + t_{ED}$$

连续扫描的第 1 个周期为: 从 1 个周期扫描的 t_{SCAN} 中去掉 t_{ED} 后加上 t_{SHED} 的时间

从连续扫描的第 2 个周期开始固定为: $t_{SPLSH} + t_{DIAG} + (t_{CONV} \times n) + t_{SHED}$

注 1. 在不使用通道专用采样 & 保持电路时, t_{SH} 为“0”。

注 2. 在没有设定自诊断时, t_{DIAG} 为“0”。

注 3. 在不使用通道专用采样 & 保持电路时, t_{SHED} 为“0”。

这是执行连续扫描模式时所需要的时间。在执行 1 个周期扫描模式和分组扫描模式时, 包括在扫描结束处理时间 (t_{ED}) 中。

表 31.9 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)	“14h”	0.4 μ s (PCLK=ADCLK=50MHz 时)
在模拟输入的信号源阻抗高、采样时间不足时设定。	“15h” ~ “FF”	例: “FFh” 5.1 μ s (PCLK=ADCLK=50MHz 时)
在 ADCLK 小于 50MHz 时, 为了使采样时间小于初始值而设定。	“00h” ~ “13”	例: “10h” 0.4 μ s (PCLK=ADCLK=40MHz 时)

注 1. 必须设定为采样时间 $\geq 0.4\mu$ s。用以下表达式表示采样时间。

$$\text{采样时间}(\mu\text{s}) = \frac{\text{ADSSTR寄存器的设定值}}{\text{ADCLK}(\text{MHz})}$$

表 31.10 扫描转换时间 (通过 PCLK 和 ADCLK 的周期数表示)

项目	符号	种类 / 条件			单位	
		同步触发	异步触发	软件触发		
扫描开始处理时间 (注 1、注 2)	通过组 A 优先控制运行进行组 A 的 A/D 转换	有中止组 B (在因组 A 的 A/D 转换源停止组 B 的运行后, 启动组 A)	6 个 PCLK + 4 个 ADCLK	—	—	
	没有中止组 B (因组 A 的 A/D 转换源启动组 A)					3 个 PCLK + 4 个 ADCLK
	上述以外		2 个 PCLK + 4 个 ADCLK	4 个 PCLK + 4 个 ADCLK	2 个 PCLK + 4 个 ADCLK	
通道专用采样 & 保持电路处时间 (注 1)	采样时间	t_{SPLSH}	t_{SH}	ADSHCR.SSTSH[7:0] 设定值 (初始值 14h) × ADCLK		
	采样 -A/D 转换等待时间		t_W	10 个 ADCLK		
自诊断转换处理时间 (注 1)	采样时间	t_{DIAG}	t_{SPL}	ADSHCR.SSTSH[7:0] 设定值 (初始值 14h) × ADCLK		
	逐次转换时间			t_{SAM}	30 个 ADCLK	
A/D 转换处理时间 (注 1)	采样时间	t_{CONV}	t_{SPL}	ADSHCR.SSTSH[7:0] 设定值 (初始值 14h) × ADCLK		
	逐次转换时间			t_{SAM}	30 个 ADCLK	
通道专用采样 & 保持电路结束处理时间		t_{SHED}	2 个 ADCLK			
扫描结束处理时间 (注 1)		t_{ED}	1 个 PCLK + 3 个 ADCLK			

注 1. 有关 t_D 、 t_{SPLSH} 、 t_{DIAG} 、 t_{CONV} 和 t_{ED} 的各时序, 请参照图 31.25 和图 31.26。
 注 2. 这是从软件编程或者触发输入开始到 A/D 转换开始的最大时间。

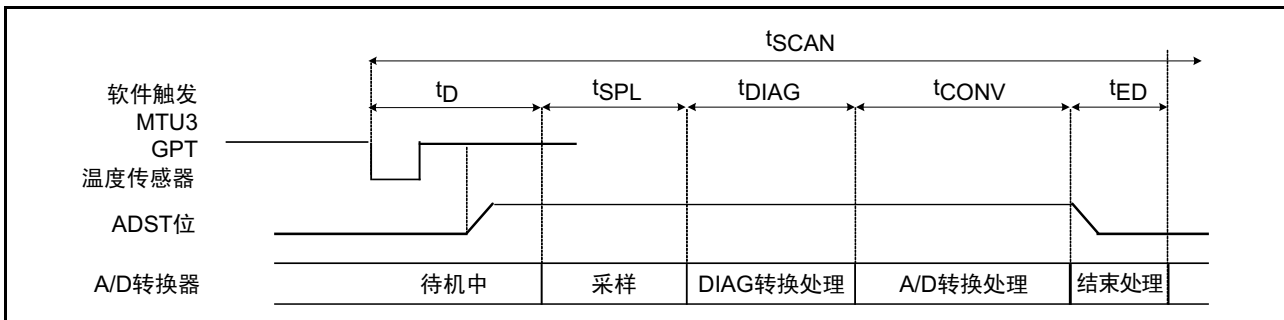


图 31.25 扫描转换的时序 (通过软件触发启动或者 MTU3 触发、GPT 触发启动的情况)

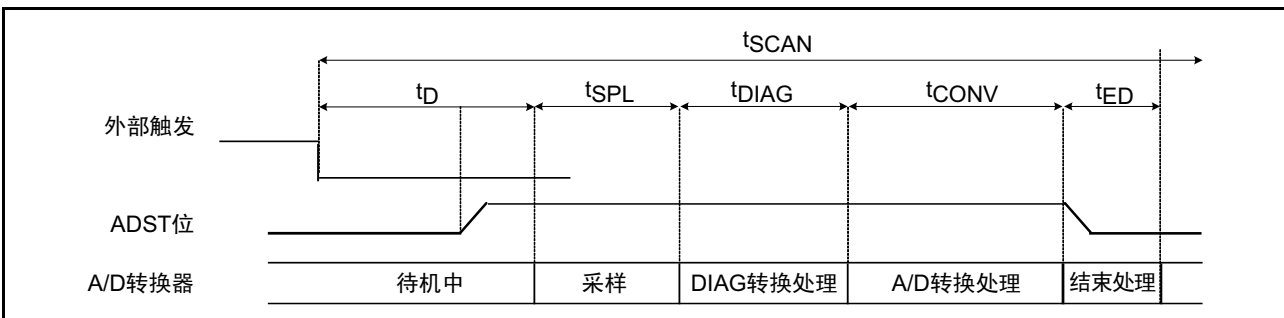


图 31.26 扫描转换的时序 (通过 ADTRG# 启动的情况)

31.3.6 寄存器的自动清除功能的使用例子

通过将 ADCER.ACE 位置“1”，能在 CPU、DTC 或者 DMAC 读 A/D 数据寄存器（ADDRy、ADRD、ADDBLDR、ADDBLDRB、ADDBLDRB）时，将 ADDRy、ADRD、ADDBLDR、ADDBLDRB、ADDBLDRB 寄存器自动清“0000h”。

能通过使用自动清除功能检测到 ADDRy、ADRD、ADDBLDR、ADDBLDRB、ADDBLDRB 寄存器的未更新故障。ADDRy 寄存器的自动清除功能无效 / 有效时的例子如下所示。

当 ADCER.ACE 位为“0”（禁止自动清除）时，如果因某种原因而无法将 A/D 转换结果（0222h）写到 ADDRy 寄存器，之前的数据（0111h）就为 ADDRy 寄存器的值。如果使用 A/D 转换结束中断将此 ADDRy 寄存器的值读到通用寄存器，就能将之前的数据（0111h）保存到通用寄存器等。但是，在进行未更新的检查时，需要边将之前的数据逐个保存到 RAM 或者通用寄存器边进行检查。

当 ADCER.ACE 位为“1”（允许自动清除）并且通过 CPU、DTC 或者 DMAC 读到 ADDRy 寄存器为“0111h”时，ADDRy 寄存器自动清“0000h”。此后，如果因某种原因而无法将 A/D 转换结果“0222h”传送到 ADDRy 寄存器，被清除的数据（0000h）就留作 ADDRy 寄存器的值。此时，如果使用 A/D 转换结束中断将 ADDRy 寄存器的值读到通用寄存器等，就将“0000h”保存到通用寄存器等。只要检查到读取的数据值为“0000h”，就能判断 ADDRy 寄存器发生未更新的故障。

31.3.7 A/D 转换值加法运算功能

对相同的通道连续进行 2 ~ 4 次 A/D 转换后，将此转换值的合计保存到数据寄存器。使用此结果的平均值，根据噪声成分能提高 A/D 转换精度，但是不保证一定能提高 A/D 转换精度。

31.3.8 模拟引脚放电功能

在进行 A/D 转换时，能通过执行对象模拟引脚的放电检测模拟引脚开路。

如果将 ADCER.DCE 位置“1”，就在结束 A/D 转换处理中的模拟引脚的采样后，进行模拟引脚的放电。因故障等造成模拟输入引脚开路时，能通过执行数次放电使转换结果变为“0000h”的近似值，并且检测引脚开路。

31.3.9 通过异步触发开始的 A/D 转换

能通过输入异步触发开始 AD 转换。使用异步触发开始 A/D 转换时，先将 A/D 转换开始触发选择位（ADSTRGR.TRGA[4:0]）置“00000b”，然后在给异步触发（ADTRG0# 引脚）输入 High 电平后，将 ADCSR.TRGE 位和 ADCSR.EXTRG 位都置“1”。异步触发的输入时序如图 31.27 所示。

有关从 ADST 位被值“1”到开始 A/D 转换为止的时间，请参照“31.6.3 开始和强制停止 A/D 转换时的运行时序”。

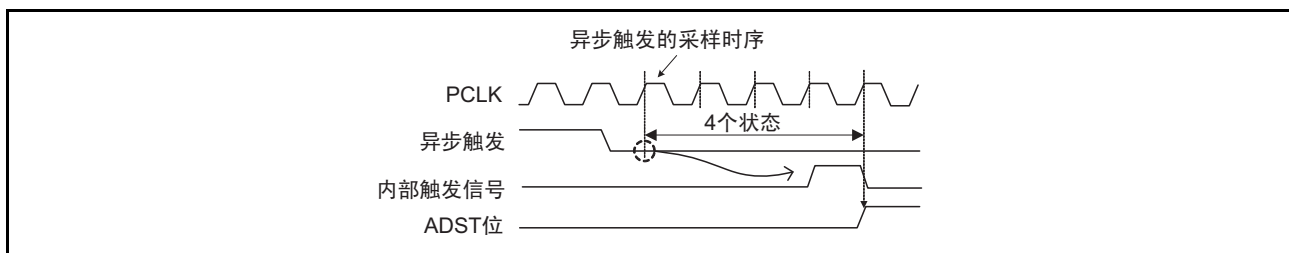


图 31.27 异步触发的输入时序

31.3.10 通过外围模块的同步触发开始的 A/D 转换

能通过 MTU3、GPT 的同步触发开始 A/D 转换。在通过同步触发开始 A/D 转换时，将 ADCSR.TRGE 位置“1”，将 ADCSR.EXTRG 位置“0”并且通过 ADSTRGR.TRSA[4:0] 位和 ADSTRGR.TRSB[4:0] 位设定为对应的启动源。

31.3.11 窗口比较器

在 ANn 引脚 (n=000 ~ 002) 中装载有窗口比较器。窗口比较器不但能作为窗口比较器运行 (检测给 ANn 引脚外加的不在 Low 电平侧基准电压 ~ High 电平侧基准电压范围内的输入电压) 还能作为 Low 电平比较器 (检测给 ANn 引脚外加的低于 Low 电平侧基准电压的输入电压) 和 High 电平比较器 (检测给 ANn 引脚外加的高于 High 电平侧基准电压的输入电压) 运行。通过 ADCMPMD0.CENn[1:0] 位 (n=000 ~ 002) 设定窗口比较器的运行模式。窗口比较器的基准电压能选择在 Low 电平侧通过 AN003/CVREFL 引脚从外部提供, 在 High 电平侧通过 AN007/CVREFH 引脚从外部提供; 或者选择使用内部生成的基准电压。窗口比较器内置对应窗口比较器检测信号的噪声消除滤波器, 能抑制因噪声导致的误检测。通过比较器检测信号向 CPU 生成比较器检测中断请求 (CMP0 ~ CMP2) 或者 POE 请求 (MTU3 的互补 PWM 输出引脚和 GPT 输出引脚的高阻抗请求)。窗口比较器的检测信号还能用作通用 PWM 定时器 (GPT) 的计数器控制触发, 以及输出无效控制源。详细内容请参照“23. 通用 PWM 定时器 (GPT)”。

窗口比较器的设定例子如下所示。

1. 通过 ADCMPMD1.VSELL0 位和 ADCMPMD1.VSELH0 位设定基准电压的外加方法。在选择内部生成的基准电压时, 通过 ADCMPMD1.REFL[2:0] 位设定 Low 电平侧基准电压、ADCMPMD1.REFH[2:0] 位设定 High 电平侧基准电压。
2. 通过 ADCMPNR0.CnNR[3:0] 位 (n=000 ~ 002) 给各比较器设定是否使用比较器检测结果的噪声消除滤波器。
3. 通过 ADCMPSEL.IEn 位 (n=000 ~ 002) 设定是否通过比较器检测信号向 CPU 请求比较器检测中断 (CMP0 ~ CMP2)。另外, 通过 ADCMPSEL.POERQm 位 (m=000 ~ 002) 设定是否通过比较器检测信号产生 POE 请求。由 ADCMPSEL.POERQm 位 (m=000 ~ 002) 的逻辑或生成 POE 请求。
4. 通过 ADCMPMD0.CENn[1:0] 位 (n=000 ~ 002) 设定检测对象引脚和窗口比较器的运行模式。

窗口比较器的运行例子如图 31.28 ~ 图 31.30 所示。

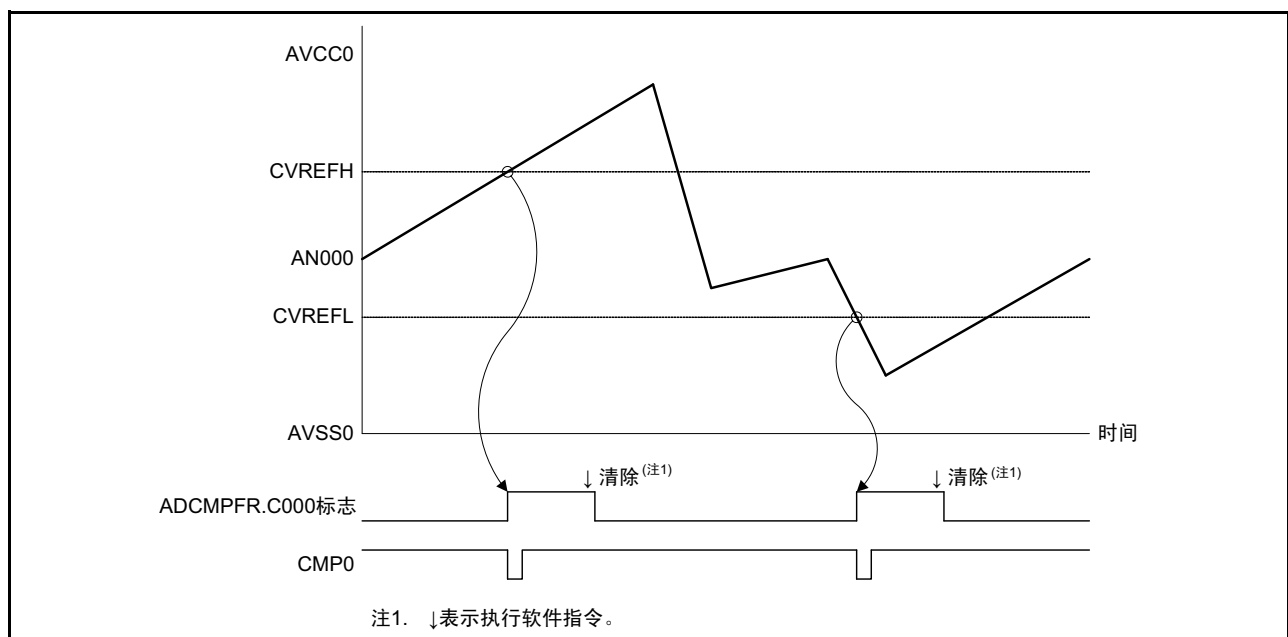


图 31.28 比较器的运行例子 (1) (选择 AN000, 并且 ADCMPMD0.CEN000=11b 的情况)

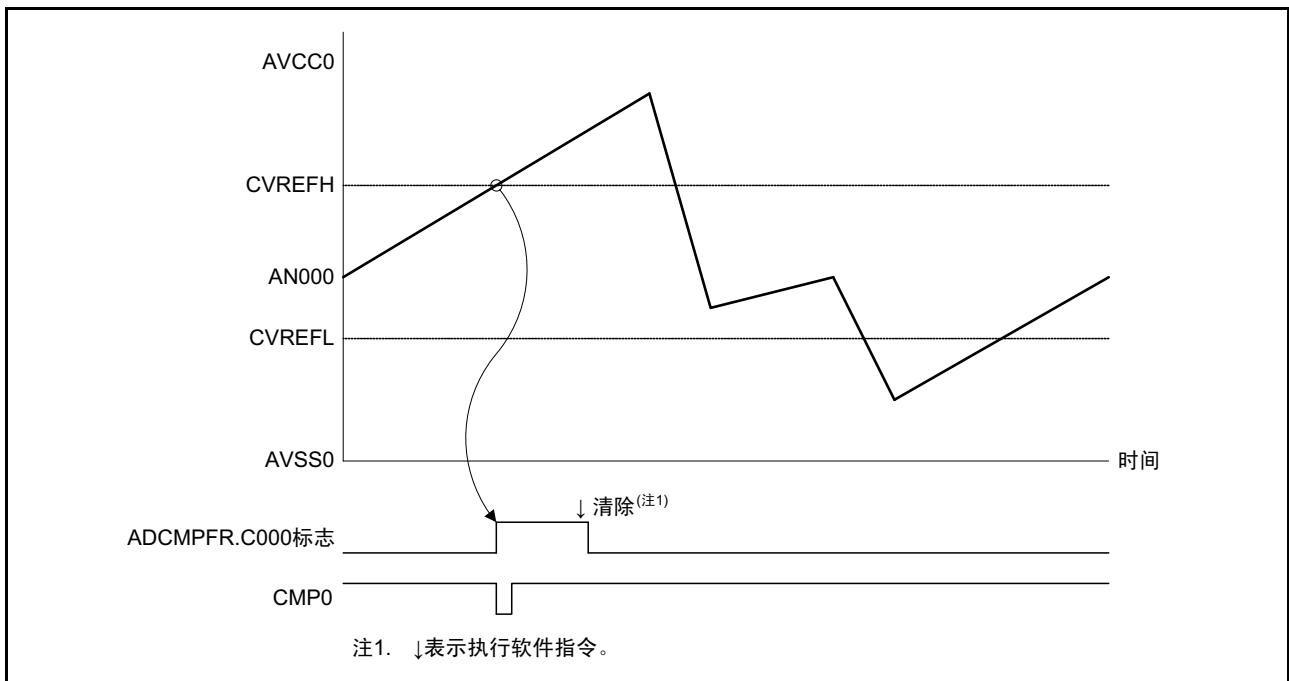


图 31.29 比较器的运行例子 (2) (选择 AN000, 并且 ADCMPMD0.CEN000=10b 的情况)

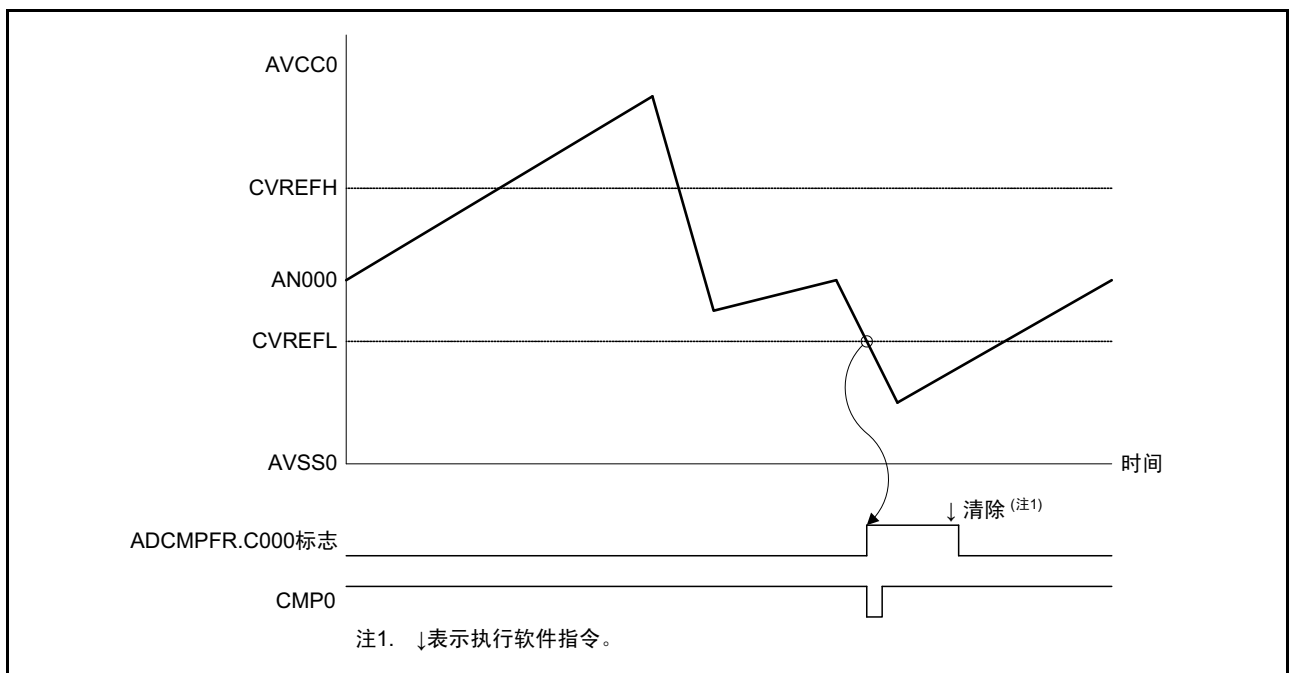


图 31.30 比较器的运行例子 (3) (选择 AN000, 并且 ADCMPMD0.CEN000=01b 的情况)

31.4 中断源和 DTC、DMAC 传送请求

31.4.1 扫描结束时的中断请求

12 位 A/D 转换器能向 CPU 请求发生扫描结束中断 (S12ADI 中断和 S12GBADI 中断)。

如果将 ADCSR.ADIE 位置“1”，就允许 S12ADI 中断；置“0”，就禁止 S12ADI 中断。如果将 ADCSR.GBADIE 位置“1”，就允许 S12GBADI 中断；置“0”，就禁止 S12GBADI 中断。

另外，在发生 S12ADI 中断和 S12GBADI 中断时能启动 DTC 或者 DMAC。如果通过 DTC 或者 DMAC 读由 S12ADI 中断或者 S12GBADI 中断转换的数据，就能无软件负担的实现连续转换。DTC 的设定请参照“18. 数据传送控制器 (DTCa) ”、DMAC 的设定请参照“17. DMA 控制器 (DMACA) ”。

31.4.2 比较器检测时的中断请求

比较器能向 CPU 请求发生比较器检测中断 (CMPm (m=0 ~ 2))。

如果将 ADCMPSEL.IEn (n=000 ~ 002) 位置“1”，就允许 CMPm 中断，置“0”，就禁止 CMPm 中断。

另外，在发生 CMPm 时能启动 DTC 或者 DMAC。DTC 的设定请参照“18. 数据传送控制器 (DTCa) ”、DMAC 的设定请参照“17. DMA 控制器 (DMACA) ”。

31.5 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 12 位 A/D 转换器的数字输出码数。
- 偏移误差
这是在数字输出从最小电压值“000000000000”变为“000000000001”时与模拟输入电压值的理想 A/D 转换特性的偏差。但是不包含量化误差。(图 31.31)
- 满刻度误差
这是在数字输出从“11111111110”变为“11111111111”时与模拟输入电压值的理想 A/D 转换特性的偏差。但是不包含量化误差。(图 31.31)
- 量化误差
这是 12 位 A/D 转换器固有的偏差，为 $1/2\text{LSB}$ 。(图 31.31)
- 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 31.31)
- 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。

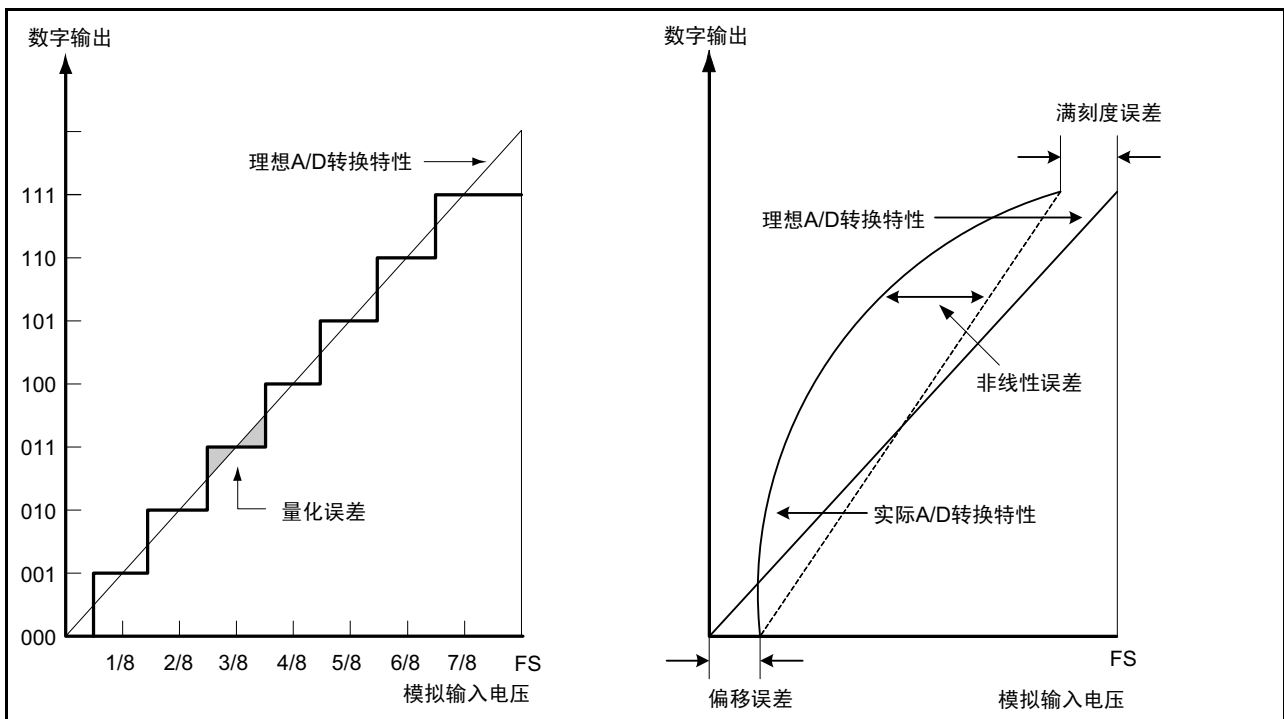


图 31.31 A/D 转换精度的定义 (3 位 A/D 转换器的例子)

31.6 使用时的注意事项

31.6.1 读数据寄存器的注意事项

必须以字为单位读 A/D 数据寄存器、A/D 数据双重化寄存器、以及 A/D 自诊断数据寄存器。为了避免因以字节为单位分 2 次读高位字节和低位字节而使第 1 次读到的 A/D 转换值和第 2 次读到的 A/D 转换值发生变化，不能以字节为单位读数据。

31.6.2 停止 A/D 转换时的注意事项

在选择异步触发或者同步触发为 A/D 转换开始条件时，为了停止 A/D 转换，必须按照图 31.32 的流程进行设定。

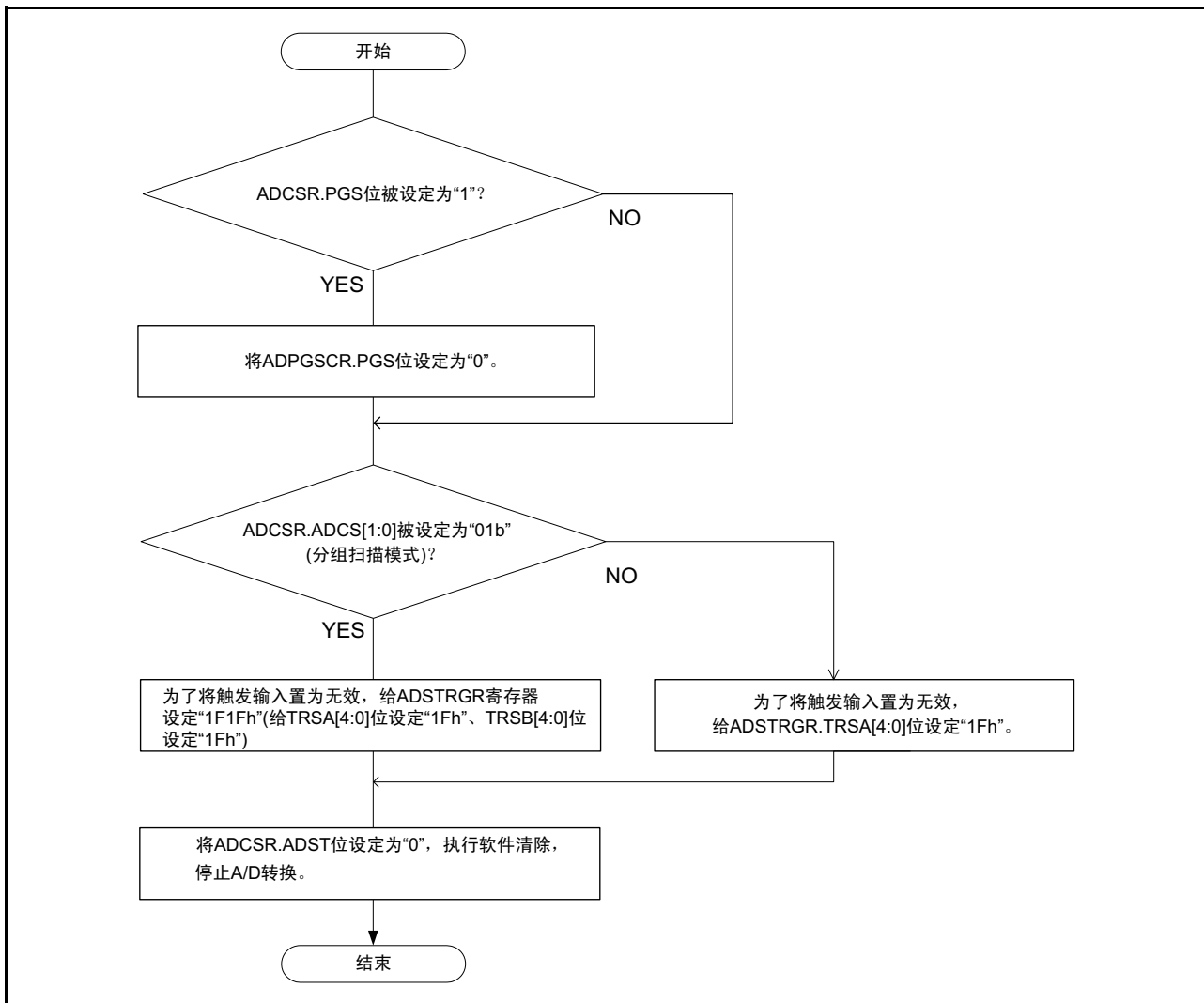


图 31.32 通过 ADCSR.ADST 位执行软件清除的设定流程

31.6.3 开始和强制停止 A/D 转换时的运行时序

在 12 位 A/D 转换器的模拟部停止的状态下，将 ADCSR.ADST 位置“1”，需要最多 4 个 ADCLK 时钟的时间来开始 12 位 A/D 转换器模拟部的运行。如果通过将 ADCSR.ADST 位置“0”来强制停止 A/D 转换，就需要最多 2 个 ADCLK 时钟的时间来停止 12 位 A/D 转换器模拟部的运行。

31.6.4 扫描结束中断处理的注意事项

在通过触发启动对相同模拟输入进行 2 次扫描时，从发生第 1 次扫描结束中断到通过第 2 次扫描进行第 1 个模拟输入的 A/D 转换结束为止，如果 CPU 未读完 A/D 转换数据，第 2 次的 A/D 转换数据就盖写第 1 次的 A/D 转换数据。

31.6.5 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 12 位 A/D 转换器的运行，初始值为停止 12 位 A/D 转换器的运行。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“11. 低功耗功能”。

31.6.6 向低功耗状态转移时的注意事项

要向模块停止模式或者软件待机模式转移时，必须停止 A/D 转换。在停止 A/D 转换时，需要确保从将 ADCSR.ADST 位置“0”到 12 位 A/D 转换器的模拟部停止为止的时间。为了确保此时间，必须按照以下步骤进行设定。

必须按照“图 31.32 通过 ADCSR.ADST 位执行软件清除的设定流程”，将 ADCSR.ADST 位设定为“0”。此后，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。

另外，在向模块停止模式、软件待机模式、深度软件待机模式转移时，12 位 A/D 转换器的一部分为运行待机状态。如果要使 12 位 A/D 转换器完全进入待机状态，必须将 MSTPCRA.MSTPA24 位置“1”。此时，在解除模块停止模式、软件待机模式、深度软件待机模式后，必须将 MSTPCRA.MSTPA24 位置“0”，并且在等待 10ms 后开始 A/D 转换。

31.6.7 有关容许信号源阻抗

为了实现 1.0 μ s 的高速转换，对于信号源阻抗不超过 3k Ω 的输入信号，本 LSI 的模拟输入能保证转换精度。在 1 个周期扫描模式中只进行 1 个引脚的转换并且外接大电容的情况下，因为输入负载实际上只有 10k Ω 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，所以有可能无法跟上微分系数大的模拟信号（例如至少 5mV/ μ s）（图 31.33）。在转换高速模拟信号或者在扫描模式中进行多个引脚的转换时，必须插入阻抗低的缓冲器。

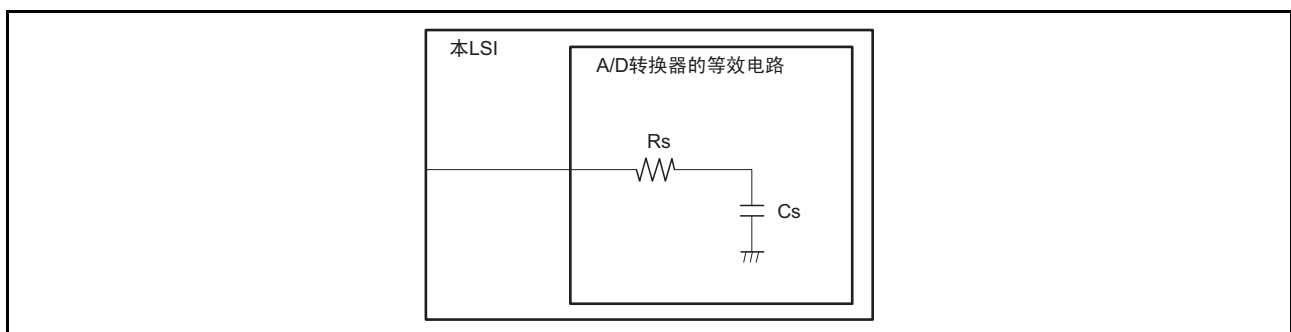


图 31.33 模拟输入引脚的内部等效电路

表 31.11 模拟引脚的规格

项目	min	max	单位
允许信号源阻抗 (注 1)	—	3	k Ω
引脚的内部等效电路	Rs	10	k Ω
	Cs	8	pF

注 1. 因模拟电源电压和模拟输入引脚而不同。详细内容请参照“36. 电特性”。

31.6.8 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，因此必须连接 AVSS0 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。

31.6.9 模拟电源引脚以外的设定范围

如果超过以下所示电压的设定范围使用 LSI，就可能给 LSI 的可靠性带来不良影响。

- 模拟输入电压的设定范围

给模拟输入引脚 ANn 外加的电压必须在 $VREFL0 \leq V_{AN} \leq VREFH0$ 的范围内。

- 各电源引脚 (AVCC0-AVSS0、VREFH0-VREFL0、VCC-VSS) 的关系

AVCC0、AVSS0 与 VCC、VSS 的关系必须是 $AVSS0 = VSS$ 。为了能形成最短的闭环，必须在各电源之间连接 $0.1\mu F$ 的电容器，如图 31.34 所示。必须在供电处连接为 $VREFL0 = AVSS0 = VSS$ 。在不使用 12 位 A/D 转换器时，必须使 $VREFH0 = AVCC0 = VCC$ 、 $VREFL0 = AVSS0 = VSS$ 。

- VREFH0、VREFL0 的设定范围

VREFH0 引脚的基准电压设定范围必须为 $VREFH0 \leq AVCC0$ 。

VREFL0 引脚的设定范围必须为 $VREFL0 = AVSS0 = VSS$ 。

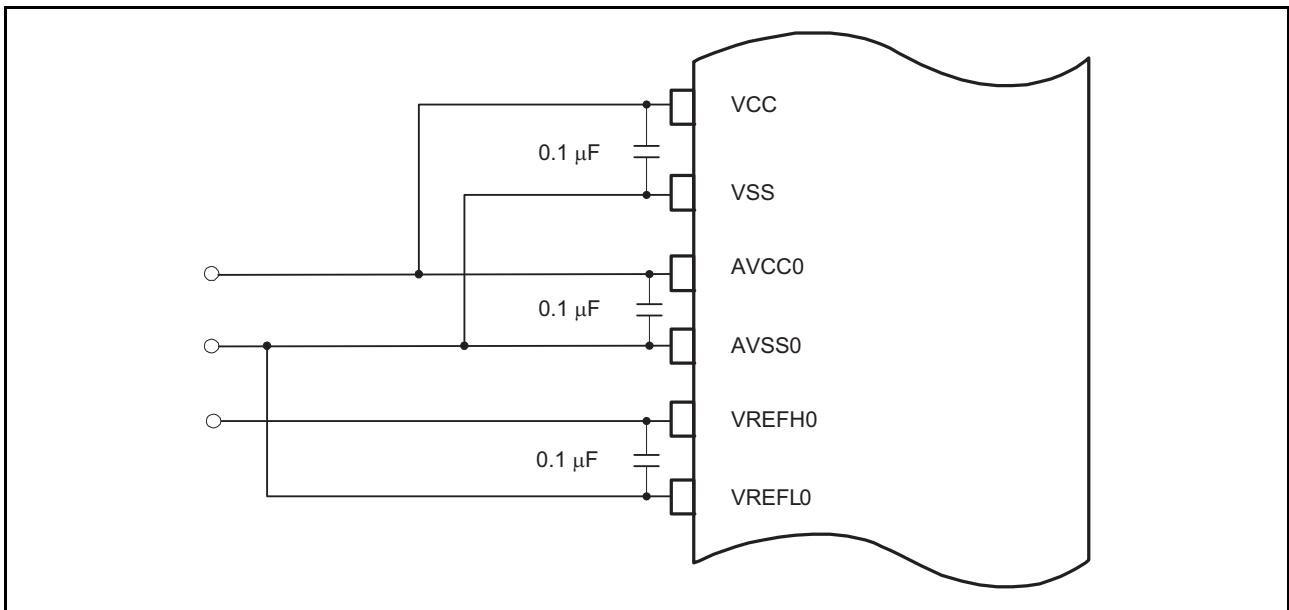


图 31.34 各电源引脚的连接例子

31.6.10 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 转换值的精度带来不良影响。必须通过模拟接地 (AVSS0) 将模拟输入引脚 (AN000 ~ AN007)、模拟基准电压 (VREFH0、VREFL0)、模拟电源 (AVCC0) 和数字电路分开，而且模拟接地 (AVSS0) 必须与电路板上稳定的数字接地 (VSS) 进行单点连接。

31.6.11 噪声对策的注意事项

如图 31.35 所示，为了防止因过大的电涌等异常电压给模拟输入引脚（AN000 ~ AN007）造成破坏，必须在 AVCC0 和 AVSS0 之间、VREFH0 和 VREFL0 之间连接电容，并且以模拟输入引脚（AN000 ~ AN007）为基准连接保护电路。

另外，连接模拟输入引脚（AN000 ~ AN007）的滤波器的电容器必须连接 VREFL0。

如图 31.35 所示，必须尽量将 $0.1\mu\text{F}$ 的电容器设置在引脚的附近。但是，因为一旦连接滤波器的电容器，就会导致模拟输入引脚（AN000 ~ AN007）的输入电流平均，从而有可能产生误差。因此，必须在对电路常数进行充分的检讨后再做决定。

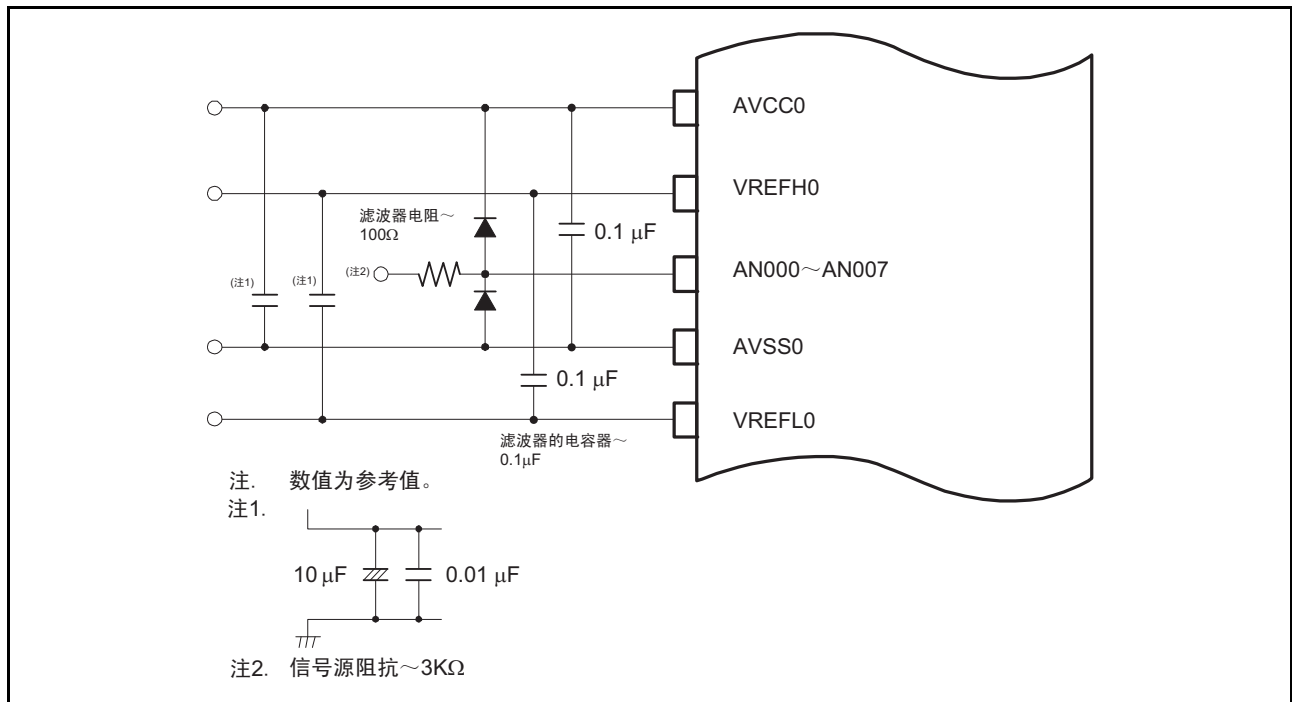


图 31.35 模拟输入保护电路的例子

32. 数据运算电路 (DOC)

32.1 概要

数据运算电路 (DOC) 对 16 位数据进行比较、加法运算或者减法运算。
数据运算电路 (DOC) 的规格和框图分别如表 32.1 和图 32.1 所示。

- 能比较 16 位数据, 并且在选择的条件下发生中断。
- 能对 16 位数据进行加法运算。
- 能对 16 位数据进行减法运算。

表 32.1 数据运行电路 (DOC) 的规格

项目	内容
数据运算功能	对 16 位数据进行比较、加法运算或者减法运算
低功耗功能	能设定为模块停止状态

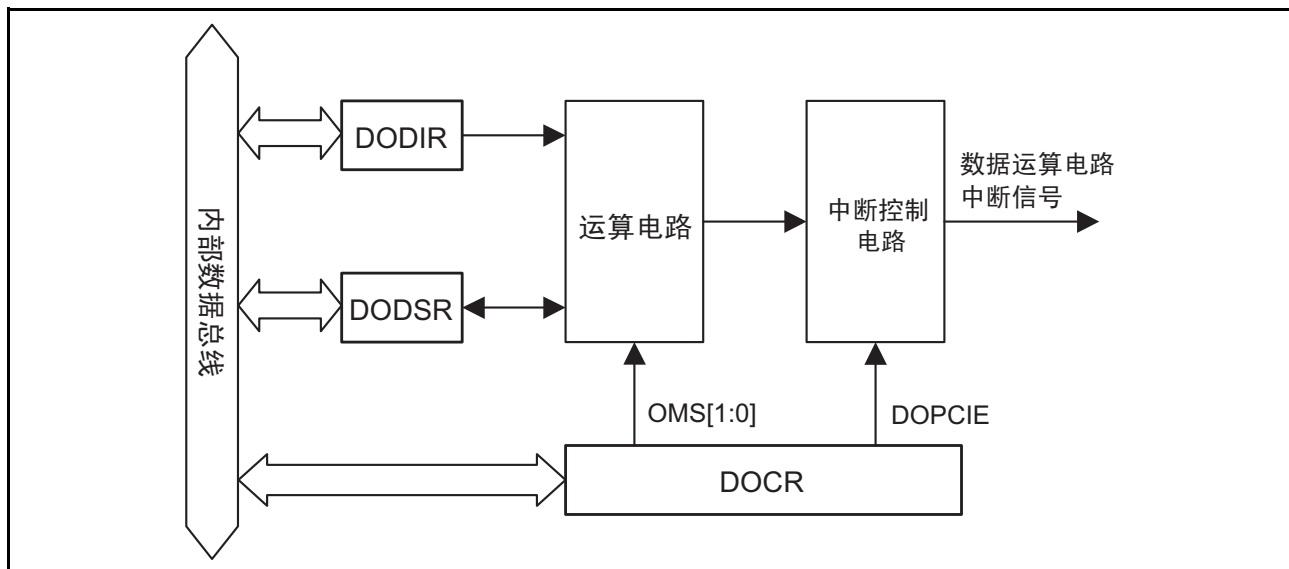


图 32.1 数据运算电路的框图

32.2 寄存器说明

32.2.1 DOC 控制寄存器 (DOCR)

地址 0008 B080h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DOPCFCL	DOPCF	DOPCIE	—	DCSEL	OMS [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	OMS[1:0]	运行模式选择位	b1 b0 0 0: 数据比较模式 0 1: 数据加法运算模式 1 0: 数据减法运算模式 1 1: 不能设定	R/W
b2	DCSEL (注 1)	检测条件选择位	0: 检测到数据比较的结果不相同 1: 检测到数据比较的结果相同	R/W
b3	—	保留位	读写值都为“0”。	R/W
b4	DOPCIE	数据运算电路中断允许位	0: 数据运算电路中断无效 1: 数据运算电路中断有效	R/W
b5	DOPCF	数据运算电路标志	显示运算结果	R
b6	DOPCFCL	DOPCF 清除位	0: 不进行任何运行 1: 清除 DOPCF 标志	R/W
b7	—	保留位	读写值都为“0”。	R/W

注 1. 只在数据比较模式中有效。

OMS[1:0] 位 (运行模式选择位)

通过设定此位选择数据运算电路的运行模式。

DCSEL 位 (检测条件选择位)

只在数据比较模式中有效。

通过设定此位选择数据比较模式中的结果检测条件。

DOPCIE 位 (数据运算电路中断允许位)

当此位为“1”时，允许数据运算电路中断。

DOPCF 标志 (数据运算电路标志)

[为“1”的条件]

- 当变为 DCSEL 位选择的条件时
- 当数据加法运算的结果大于“FFFFh”时
- 当数据减法运算的结果小于“0000h”时

[为“0”的条件]

- 给 DOPCFCL 位写“1”时

DOPCFCL 位 (DOPCF 清除位)

如果将此位置“1”，就清除 DOPCF 标志。

读取值为“0”。

32.2.2 DOC 数据输入寄存器 (DODIR)

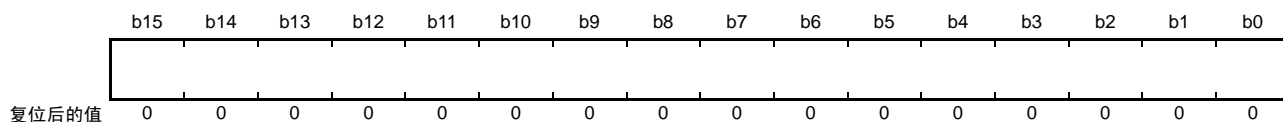
地址 0008 B082h



DODIR 寄存器是保存运算对象的 16 位数据的 16 位可读写寄存器。

32.2.3 DOC 数据设定寄存器 (DODSR)

地址 0008 B084h



DODSR 寄存器在数据比较模式中是保存基准 16 位数据的 16 位可读写寄存器。另外，在数据加法运算模式和数据减法运算模式中是保存运算结果的寄存器。

32.3 运算说明

32.3.1 数据比较模式

数据比较模式的运行例子如图 32.2 所示。

在数据比较模式中，数据运算电路如下运行。

1. 如果给 DOCR.OMS[1:0] 位写“00b”，就为数据比较模式。
2. 给 DODSR 寄存器设定作为基准的 16 位数据。
3. 给 DODIR 寄存器写要进行比较的 16 位数据。
4. 给 DODIR 寄存器写要进行比较的 16 位数据，直到写完全部要进行比较的数据。
5. 当写到 DODIR 寄存器的数据与设定到 DODSR 寄存器的数据不相同，DOCR.DOPCF 标志就被置“1”。另外，如果给 DOCR.DOPCIE 位设定“1”，就发生数据运算电路中断。

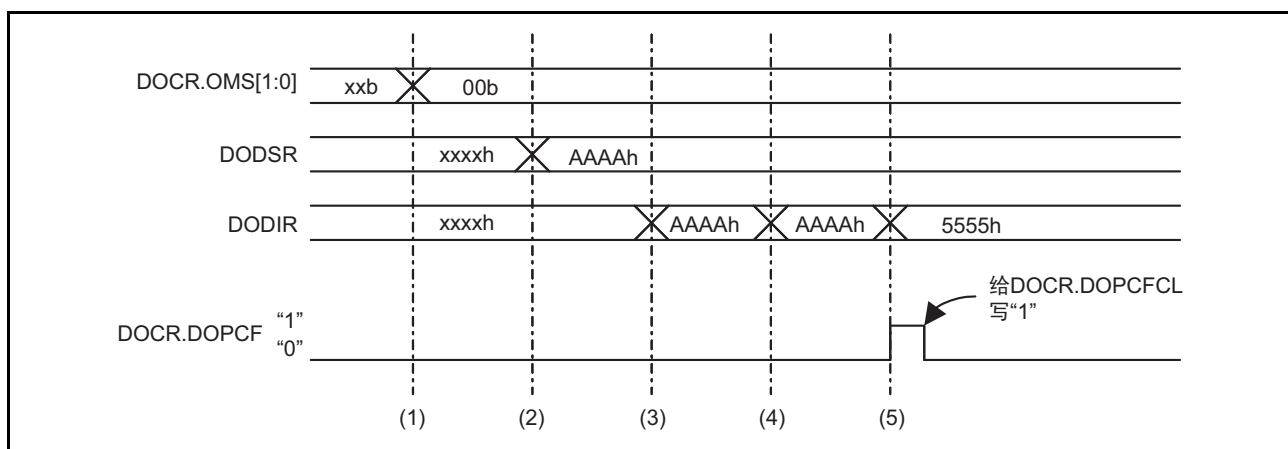


图 32.2 数据比较模式的运行例子

32.3.2 数据加法运算模式

数据加法运算模式的运行例子如图 32.3 所示。

在数据加法运算模式中，数据运算电路如下运行。

1. 如果给 DOCR.OMS[1:0] 位写“01b”，就为数据加法运算模式。
2. 给 DODSR 寄存器设定作为初始值的 16 位数据。
3. 给 DODIR 寄存器写要进行加法运算的 16 位数据。运算结果保存到 DODSR 寄存器。
4. 给 DODIR 寄存器写要进行加法运算的 16 位数据，直到写完全部要进行加法运算的数据。
5. 当运算结果大于“FFFFh”时，DOCR.DOPCF 标志就被置“1”。另外，如果给 DOCR.DOPCIE 位设定“1”，就发生数据运算电路中断。

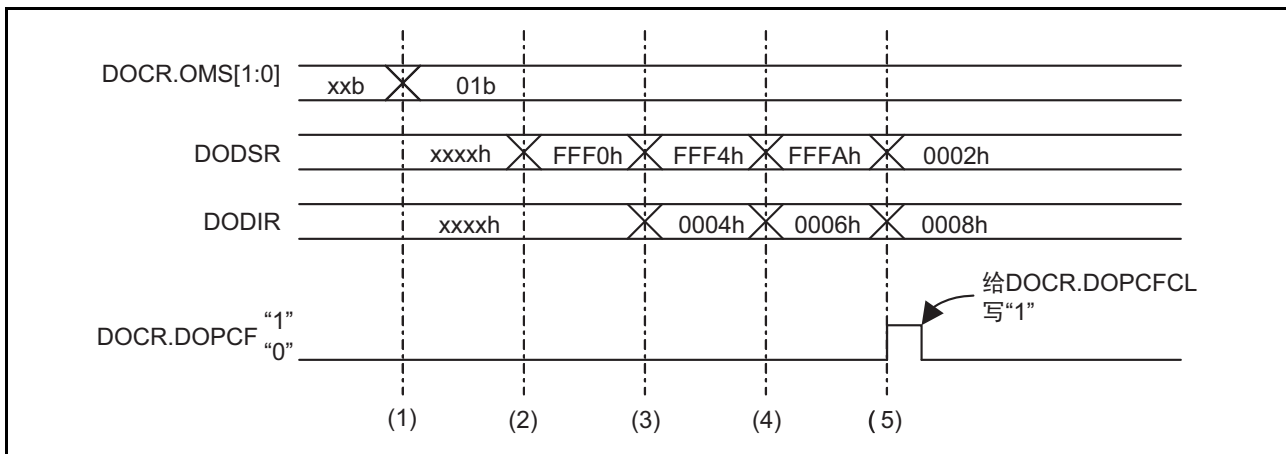


图 32.3 数据加法运算模式的运行例子

32.3.3 数据减法运算模式

数据减法运算模式的运行例子如图 32.4 所示。
在数据减法运算模式中，数据运算电路如下运行。

1. 如果给 DOCR.OMS[1:0] 位写“10b”，就为数据减法运算模式。
2. 给 DODSR 寄存器设定作为初始值的 16 位数据。
3. 给 DODIR 寄存器写要进行减法运算的 16 位数据。运算结果保存到 DODSR 寄存器。
4. 给 DODIR 寄存器写要进行减法运算的 16 位数据，直到写全部要进行减法运算的数据。
5. 当运算结果小于“0000h”时，DOCR.DOPCF 标志就被置“1”。另外，如果给 DOCR.DOPCIE 位设定“1”，就发生数据运算电路中断。

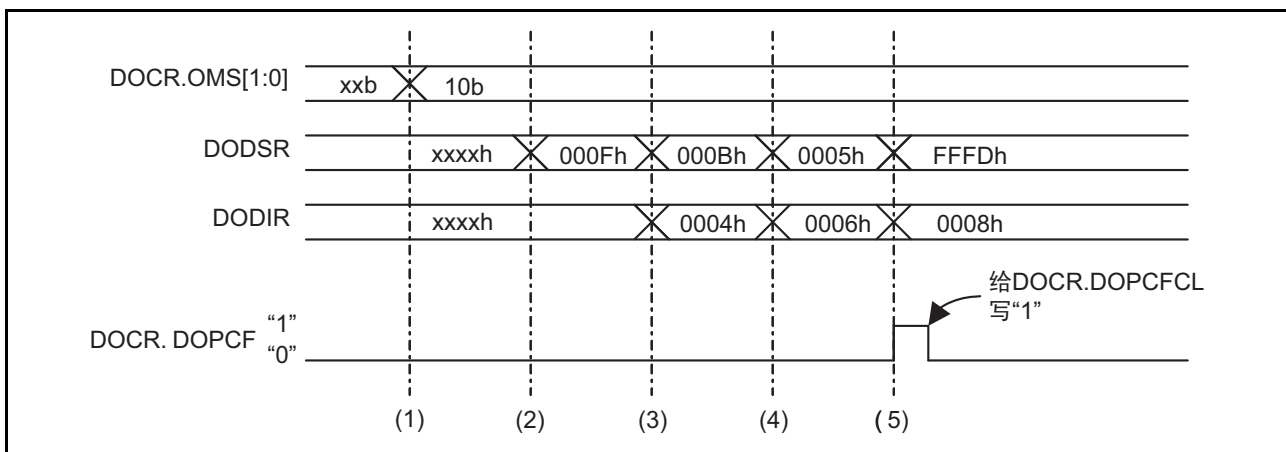


图 32.4 数据减法运算模式的运行例子

32.4 中断请求

数据运算电路生成的中断请求只有数据运算电路中断请求。如果产生中断源，状态标志就被置“1”。中断请求的内容如表 32.2 所示。

表 32.2 数据运算电路中断请求

中断请求	状态标志	中断源
数据运算电路中断	DOPCF	<ul style="list-style-type: none"> • 当变为 DOCR.DCSEL 位选择的条件时 • 当数据加法运算的结果大于“FFFFh”时 • 当数据减法运算的结果小于“0000h”时

32.5 使用时的注意事项

32.5.1 模块停止功能的设定

能通过模块停止控制寄存器 B (MSTPCRB) 设定禁止或者允许数据运算电路的运算。初始值时为停止数据运算电路的运行。能通过解除模块停止状态存取寄存器。详细内容请参照“11. 低功耗功能”。

33. RAM

RX63T 群内置高速静态 RAM。

33.1 概要

内部 RAM 的规格如表 33.1 所示。

表 33.1 内部 RAM 的规格

项目	内容
RAM 容量	8K 字节
存取	<ul style="list-style-type: none"> 以 1 个周期进行读写。 能选择内部 RAM 的有效或者无效（注 1）。
低功耗功能	能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“3.2.3 系统控制寄存器 1（SYSCR1）”。

内部 RAM 容量	RAM 地址
8K 字节	内部 RAM0: 0000 0000h ~ 0000 1FFFh

33.2 运行说明

33.2.1 低功耗功能

通过设定模块停止控制寄存器 C（MSTPCRC）停止给内部 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。

通过停止提供时钟，使内部 RAM0 变为模块停止状态。但是在复位后，内部 RAM0 为运行状态。

一旦进入模块停止状态，就无法存取内部 RAM。不能在存取内部 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“11. 低功耗功能”。

34. ROM（保存代码的闪存）

RX63T 群内置用于保存最多 64K 字节代码的闪存（ROM）和用于保存 8K 字节数据的闪存（E2 数据闪存）。本章说明用于保存代码的闪存，有关 E2 数据闪存，请参照“35. E2 数据闪存（保存数据的闪存）”。

34.1 概要

ROM 的规格如表 34.1 所示，ROM 容量和 ROM 地址的对应表如表 34.2 所示，ROM 和 E2 数据闪存的框图如图 34.1 所示。

表 34.1 ROM 的规格

项目	内容	
存储空间	• 最多 64K 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 进行改写的专用定序器（FCU）。 • 能通过给 FCU 发行命令，进行 ROM 的编程或者擦除。 • 如果读擦除状态的 ROM，就能以 32 位为单位读到“FFFF FFFFh”。 	
BGO（后台操作）功能	<ul style="list-style-type: none"> • 在对 E2 数据闪存进行编程或者擦除的期间，能执行分配在 ROM 区的程序。 • 在对 ROM 进行编程或者擦除的期间，CPU 能执行分配在 ROM/E2 数据闪存以外区域的程序 	
挂起 / 恢复功能	<ul style="list-style-type: none"> • 中止对 ROM 进行的编程或者擦除，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"> • 用户区的编程单位：128 字节 • 用户区的擦除单位：以块为单位 	
板上编程（2 种）	通过引导模式改写 <ul style="list-style-type: none"> • 使用异步串行接口（SCI1） • 自动调整通信速度 	
	通过用户程序中的 ROM 改写程序进行改写 <ul style="list-style-type: none"> • 不需要对系统进行复位就能对 ROM 进行改写 	
保护功能	软件保护功能	能通过 FENTRYR.FENTRY0 位、FWEPROR.FLWE[1:0] 位和锁定位防止意料不到的改写。
	命令锁定状态	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“36. 电特性”。	

表 34.2 ROM 容量和 ROM 地址的对应表

ROM 容量	ROM 地址
64K 字节	FFFF 0000h ~ FFFF FFFFh
48K 字节	FFFF 4000h ~ FFFF FFFFh
32K 字节	FFFF 8000h ~ FFFF FFFFh

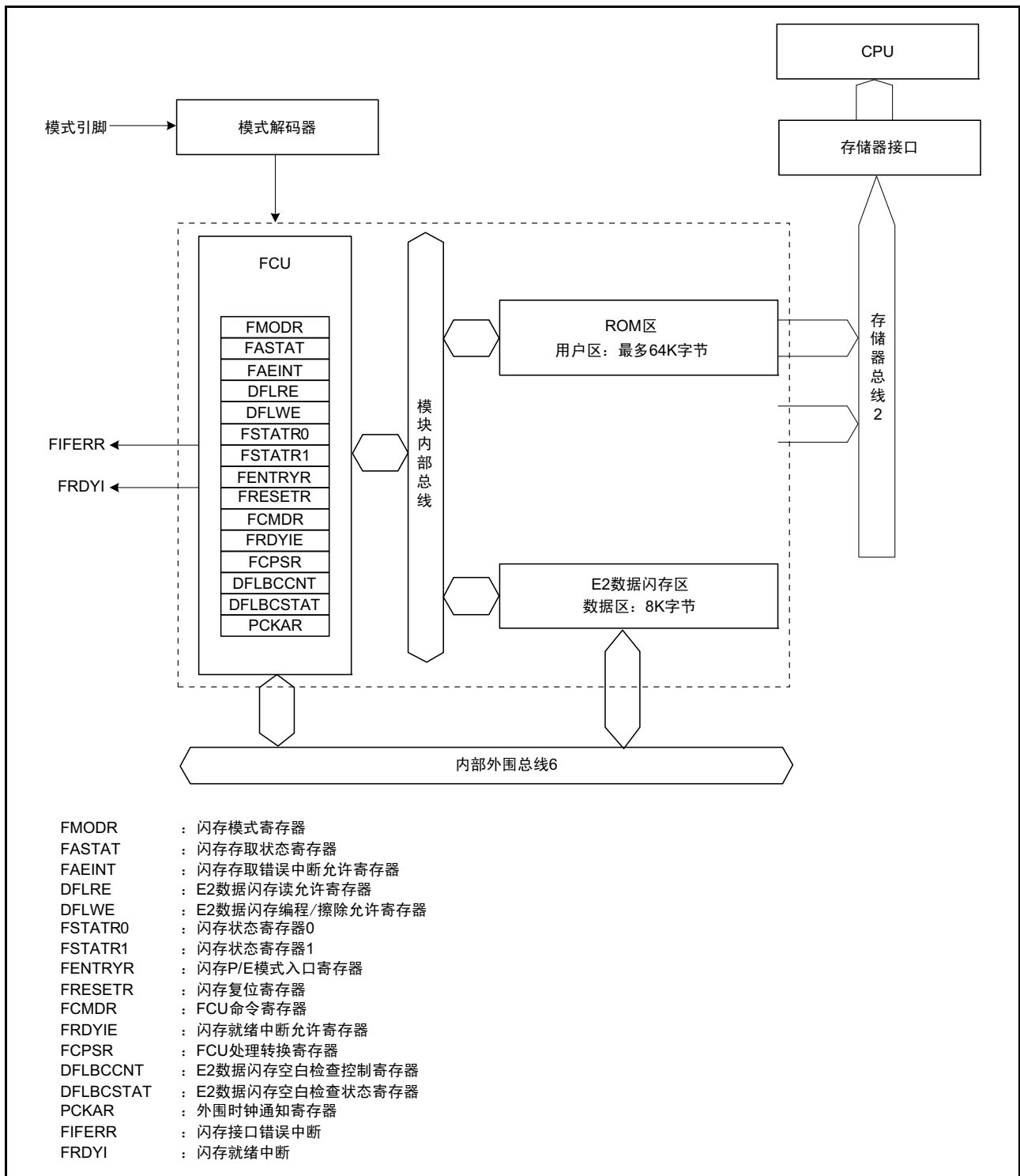


图 34.1 ROM 的框图

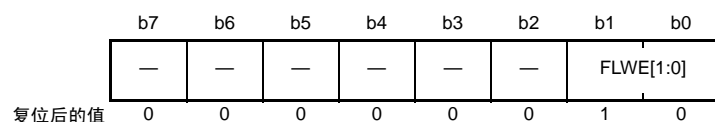
34.2 寄存器说明

部分寄存器也有 E2 数据闪存的相关位，本章只说明 ROM 的相关位的功能。有关 E2 数据闪存相关位功能的详细内容，请参照“35. E2 数据闪存（保存数据的闪存）”的“35.2 寄存器说明”。

P/E 表示编程 / 擦除。

34.2.1 闪存编程 / 擦除保护寄存器（FWEPROR）

地址 0008 C296h



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 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

在 RES# 引脚复位时、转移到软件待机模式时、转移到深度软件待机模式时、或者低电源电压时，对 FWEPROR 寄存器进行初始化。

FLWE[1:0] 位（闪存编程 / 擦除位）

这些位通过软件对闪存编程和擦除的执行进行保护。

34.2.2 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 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

在内部 ROM 为无效时，FMODR 寄存器的读数据为“00h”，不能写。

FRDMD 位（FCU 读模式选择位）

此位指定锁定位的读法。

在使用 E2 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“35. E2 数据闪存（保存数据的闪存）”。

34.2.3 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	E2 数据闪存的编程 / 擦除保护违反位	请参照“35. E2 数据闪存（保存数据的闪存）”。	R/(W) (注 1)
b1	DFLRPE	E2 数据闪存的读保护违反位	请参照“35. E2 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	E2 数据闪存的存取违反位	请参照“35. E2 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: 没有 ROM 存取错误 1: 有 ROM 存取错误	R/(W) (注 1)

注. 为了将标志置“0”，只能在读“1”后写“0”。

在内部 ROM 为无效时，FASTAT 寄存器的读数据为“00h”，不能写。如果 FASTAT.DFLWPE、FASTAT.DFLRPE、FASTAT.DFLAE、FASTAT.ROMAE 位中的任意一位变为“1”，FSTAT0.ILGLERR 位就变为“1”，并且 FCU 进入命令锁定状态（参照“34.8.2 命令锁定状态”）。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。

CMDLK 位（FCU 命令锁定位）

此位表示 FCU 处于命令锁定状态（参照“34.8.2 命令锁定状态”）。

[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 FCU 发行状态寄存器清除命令后

ROMAE 位（ROM 存取违反位）

此位表示有没有违反 ROM 的存取。

如果 ROMAE 位变为“1”，FSTATR0.ILGLERR 位就变为“1”，并且 FCU 进入命令锁定状态。

[为“1”的条件]

- 在 ROM P/E 正常模式的状态下对以下的 ROM 编程/擦除地址发行读存取命令时

ROM 容量	ROM 编程 / 擦除地址
	FENTRY0 位为“1”
64K 字节	00FF 0000h ~ 00FF FFFFh
48K 字节	00FF 4000h ~ 00FF FFFFh
32K 字节	00FF 8000h ~ 00FF FFFFh

- 当对以下的 ROM 编程/擦除地址发行存取命令时

ROM 容量	ROM 编程 / 擦除地址
	FENTRY0 位为“0”
64K 字节	00FF 0000h ~ 00FF FFFFh
48K 字节	00FF 4000h ~ 00FF FFFFh
32K 字节	00FF 8000h ~ 00FF FFFFh

- 在设定 FENTRYR 寄存器后转移到 ROM P/E 模式的状态下，对以下的 ROM 读地址发行读存取命令时

ROM 容量	ROM 编程 / 擦除地址
64K 字节	FFFF 0000h ~ FFFF FFFFh
48K 字节	FFFF 4000h ~ FFFF FFFFh
32K 字节	FFFF 8000h ~ FFFF FFFFh

[为“0”的条件]

- 在读“1”后写“0”时

34.2.4 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
	ROMAE IE	—	—	CMDLK IE	DFLAEIE	—	DFLRPE IE	DFLWPE IE
复位后的值	1	0	0	1	1	0	1	1

位	符号	位名	功能	R/W
b0	DFLWPEIE	E2 数据闪存的编程 / 擦除保护违反中断允许位	请参照“35. E2 数据闪存（保存数据的闪存）”。	R/W
b1	DFLRPEIE	E2 数据闪存的读保护违反中断允许位	请参照“35. E2 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	E2 数据闪存的存取违反中断允许位	请参照“35. E2 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

在内部 ROM 无效模式中，FAEINT 寄存器的读数据为“00h”，不能写。

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。

34.2.5 闪存状态寄存器 0（FSTATR0）

地址 007F FFB0h

b7	b6	b5	b4	b3	b2	b1	b0
FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD
1	0	0	0	0	0	0	0

复位后的值

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 没有检测到非法命令和 ROM/E2 数据闪存的非法存取 1: FCU 检测到非法命令或者 ROM/E2 数据闪存的非法存取	R
b7	FRDY	闪存就绪位	0: 正在处理编程或者擦除，正在处理编程或者擦除的中止，正在处理锁定位读 2 命令，正在处理外围时钟通知命令，或者正在处理 E2 数据闪存的空白检查（参照“35. E2 数据闪存（保存数据的闪存）”）。 1: 没有执行上述处理	R

在内部 ROM 为无效时，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“34.7 挂起”。

[为“1”的条件]

- 当开始进行编程的中止处理时

[为“0”的条件]

- 当接受恢复命令时

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“34.7 挂起”。

[为“1”的条件]

- 当开始进行擦除的中止处理时

[为“0”的条件]

- 当接受恢复命令时

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]

- 当接受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/E2 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“34.8.2 命令锁定状态”。

[为“1”的条件]

- 在编程过程中发生错误时
- 当对由锁定位保护的区域发行编程命令时

[为“0”的条件]

- 在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/E2 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“34.8.2 命令锁定状态”。

[为“1”的条件]

- 在擦除过程中发生错误时
- 当对由锁定位保护的区域发行擦除命令时

[为“0”的条件]

- 在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/E2 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“34.8.2 命令锁定状态”。

[为“1”的条件]

- 当FCU检测到非法命令时
- 当FCU检测到ROM/E2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”)
- 当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

34.2.6 闪存状态寄存器 1（FSTATR1）

地址 007F FFB1h

	b7	b6	b5	b4	b3	b2	b1	b0
	FCUERR	—	—	FLOCKST	—	—	—	—
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	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

在内部 ROM 为无效时，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。

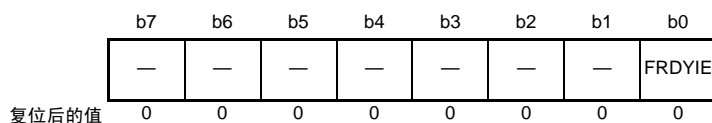
FCUERR 位（FCU 错误位）

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。

34.2.7 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

在内部 ROM 为无效时，FRDYIE 寄存器的读数据为“00h”，不能写。

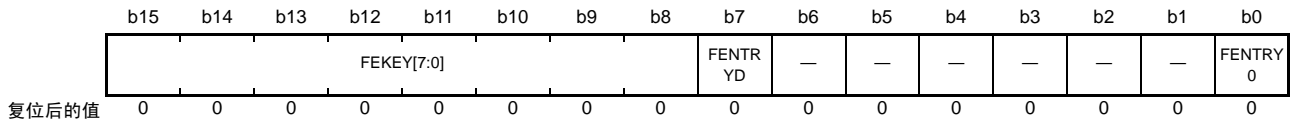
FRDYIE 位（闪存就绪中断允许位）

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（FSTATR0.FRDY 位从“0”变为“1”），就产生闪存就绪中断请求（FRDYI）。

34.2.8 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 为 ROM 读模式 1: ROM 为 ROM P/E 模式	R/W
b6-b1	—	保留位	读写值都为“0”。	R/W
b7	FENTRYD	E2 数据闪存 P/E 模式入口位	请参照“35. E2 数据闪存（保存数据的闪存）”。	R/W
b15-b8	FEKEY[7:0]	键码	控制是否改写 FENTRYR 寄存器。 如果改写 FENTRYR 寄存器，就必须以 16 位为单 位，给高 8 位写“AAh”、低 8 位写任意的值。	R/(W) (注 1)

注 1. 不保持写数据。

为了将 ROM/E2 数据闪存设定为 P/E 模式，使 FCU 能接受命令，需要将 FENTRYD 位或者 FENTRY0 位置“1”。但是，如果给 FENTRYR 寄存器写“AA01h”和“AA80h”以外的值，FSTATR0.ILGLERR 位就变为“1”，FCU 进入命令锁定状态。

如果要在存取 FENTRYR 寄存器后转移到 ROM 读模式，就必须在写 FENTRYR 寄存器后读该寄存器，确认已变为设定值后进行 ROM 读运行。

在内部 ROM 无效时，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。

FENTRY0 位（ROM P/E 模式入口位 0）

此位将 ROM 64K 字节的读地址：FFFF 0000h ~ FFFF FFFFh，编程 / 擦除地址：00FF 0000h ~ 00FF FFFFh；ROM 48K 字节的读地址：FFFF 4000h ~ FFFF FFFFh，编程 / 擦除地址：00FF 4000h ~ 00FF FFFFh；ROM 32K 字节的读地址：FFFF 8000h ~ FFFF FFFFh，编程 / 擦除地址：00FF 8000h ~ 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
- FSTATR0.FRDY 位为“1”
- 以字为单位给 FEKEY[7:0] 位写“AAh”

[为“1”的条件]

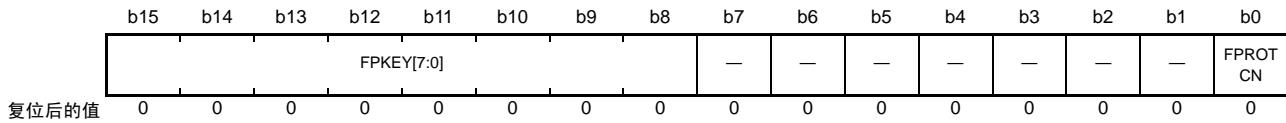
- 在满足写操作的有效条件并且 FENTRYR 寄存器为“0000h”的状态下，给 FENTRY0 位写“1”时

[为“0”的条件]

- 当以字节为单位进行写操作时
- 在 FEKEY[7:0] 位不为“AAh”的状态下，以字为单位进行写操作时
- 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

34.2.9 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	控制是否改写 FPROTR 寄存器。 如果改写 FPROTR 寄存器，就必须以 16 位为单 位，给高 8 位写“55h”、低 8 位写任意的值。	R/(W) (注1)

注 1. 不保持写数据。

在内部 ROM 为无效时，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]

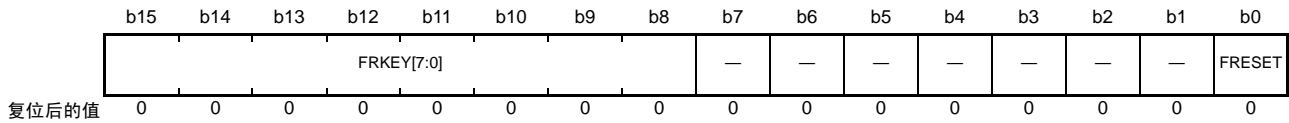
- 在 FENTRYR 寄存器的值不为“0000h”的状态下，以字为单位给 FPKEY[7:0]位写“55h”并且给 FPROTCN 位写“1”时

[为“0”的条件]

- 当以字节为单位进行写操作时
- 在 FPKEY[7:0]位不为“55h”的状态下，以字为单位进行写操作时
- 以字为单位给 FPKEY[7:0]位写“55h”并且给 FPROTCN 位写“0”时
- 当 FENTRYR 寄存器的值为“0000h”时

34.2.10 闪存复位寄存器（FRESETR）

地址 007F FFB6h



位	符号	位名	功能	R/W
b0	FRESET	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	控制是否改写 FRESETR 寄存器。 如果改写 FRESETR 寄存器，就必须以 16 位为单位，给高 8 位写“55h”、低 8 位置任意的值。	R/(W) (注1)

注 1. 不保持写数据。

在内部 ROM 为无效时，FRESETR 寄存器的读数据为“0000h”，不能写。

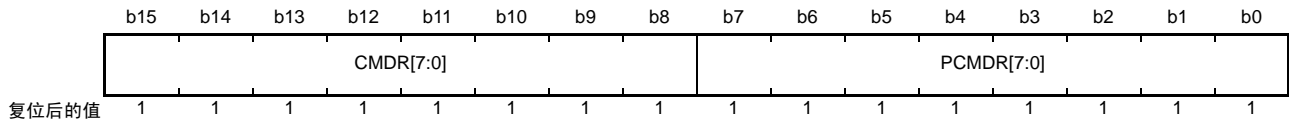
FRESET 位（闪存复位的位）

如果将 FRESET 位置“1”，ROM/E2 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/E2 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，FRESET 位为“1”的状态必须保持 tFCUR（参照“36. 电特性”）的时间。在 FRESET 位保持“1”的期间，必须禁止读 ROM/E2 数据闪存。因为在 FRESET 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。

34.2.11 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

在内部 ROM 为无效时，FCMDR 寄存器的读数据为“0000h”，写操作无效。

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 34.3 所示，空白检查的内容请参照 E2 数据闪存的“35.6 E2 数据闪存的编程 / 擦除”。

表 34.3 接受各命令后的 FCMDR 寄存器状态

命令	CMDR[7:0]	PCMDR[7:0]
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

34.2.12 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
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

在内部 ROM 为无效时，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/E2 数据闪存的擦除处理过程中发行 P/E 挂起命令时，此位选择擦除中止处理模式，详细内容请参照“34.7 挂起”。

34.2.13 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]								—	—
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	00h: 没有错误 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

在内部 ROM 为无效时，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。

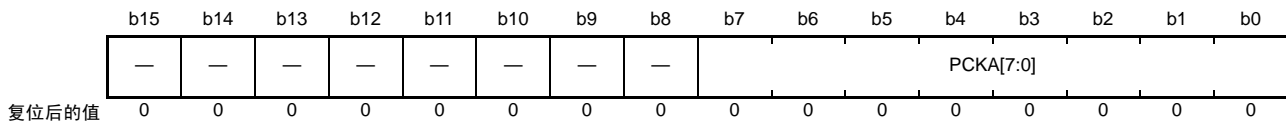
PEERRST[7:0] 位（P/E 错误状态位）

这些位表示在 ROM/E2 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”的状态下并且 FSTATR0.FRDY 位变为“1”时，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。

34.2.14 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/E2 数据闪存进行编程或者擦除时，这些位设定 FlashIF 时钟（FCLK）。	R/W
b15-b8	—	保留位	读写值都为“0”。	R/W

在内部 ROM 为无效时，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/E2 数据闪存进行编程或者擦除时，这些位设定 FlashIF 时钟（FCLK）。

必须在编程或者擦除前给 PCKA[7:0] 位设定 FCLK 的频率，然后发行外围时钟通知命令。不能在对 ROM/E2 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 FlashIF 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 的第 1 位进位。
- 将 36 转换为 2 进制数，给 PCKA[7:0] 位的高位设定“00h”，给低位设定“24h”（“0010 0100b”）。

注 1. 如果将 PCKA[7:0] 位设定在 4MHz ~ 50MHz 的范围外，就不能对 ROM/E2 数据闪存发行改写命令。

注 2. 如果给 PCKA[7:0] 位设定和 FCLK 不同的频率，就可能破坏 ROM/E2 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。

34.3 ROM 的存储区结构

RX63T 群的 ROM 由最多 64K 字节的用户区构成，ROM 的存储区结构如图 34.2 所示。
 必须注意：在进行读、编程或者擦除时，用户区的地址不同。

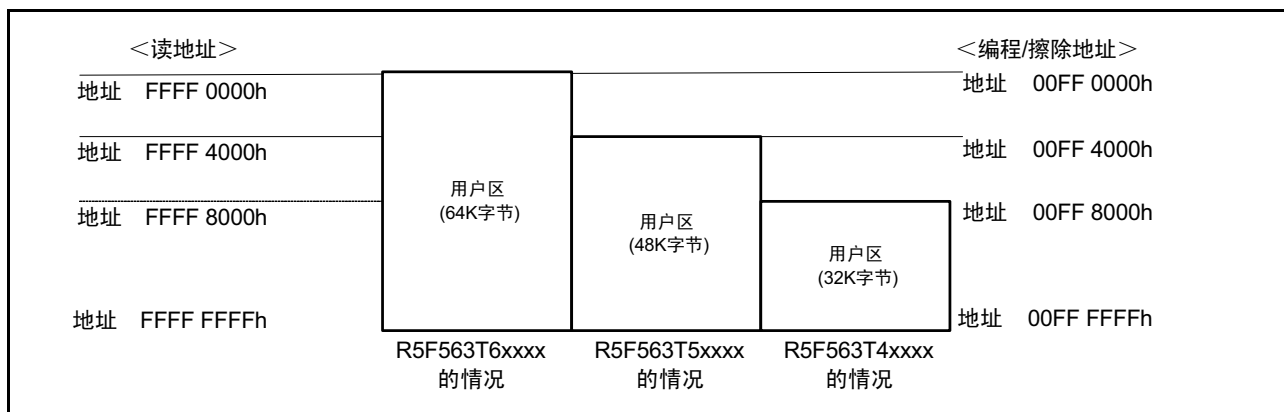


图 34.2 ROM 的存储区结构

34.4 块结构

用户区的擦除块的结构如图 34.3 所示。

用户区根据 ROM 容量被分割为以下的块，以块为单位进行擦除。以低位地址为“00h”开始的 128 字节为单位进行编程。

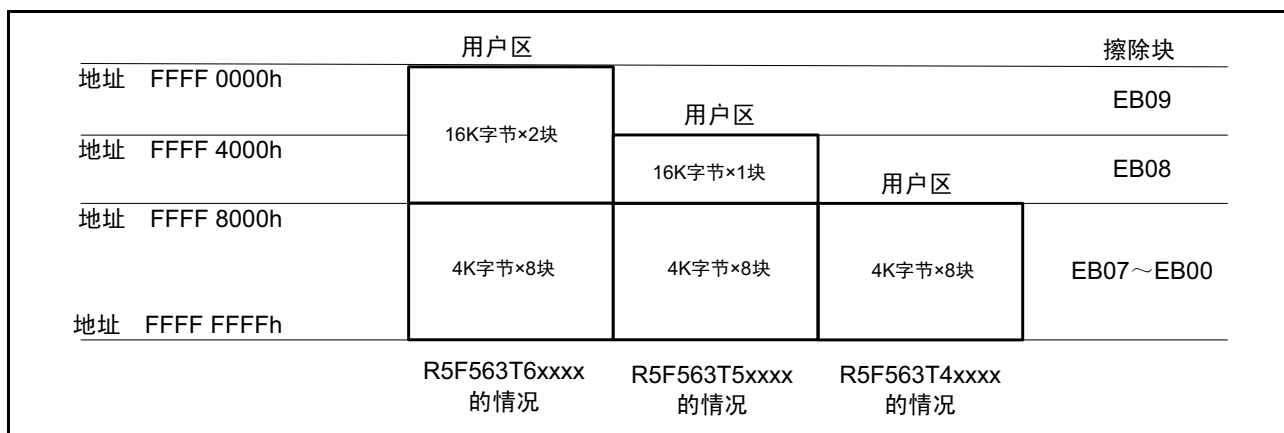


图 34.3 用户区的擦除块结构

34.5 ROM 相关的运行模式

RX63T 群的运行模式转移图如图 34.4 所示。

如果设定 MD 引脚，并且解除复位，就按图 34.4 进行转移。

有关 MD 引脚的设定值和 RX63T 群的运行模式的关系，请参照“3. 运行模式”。

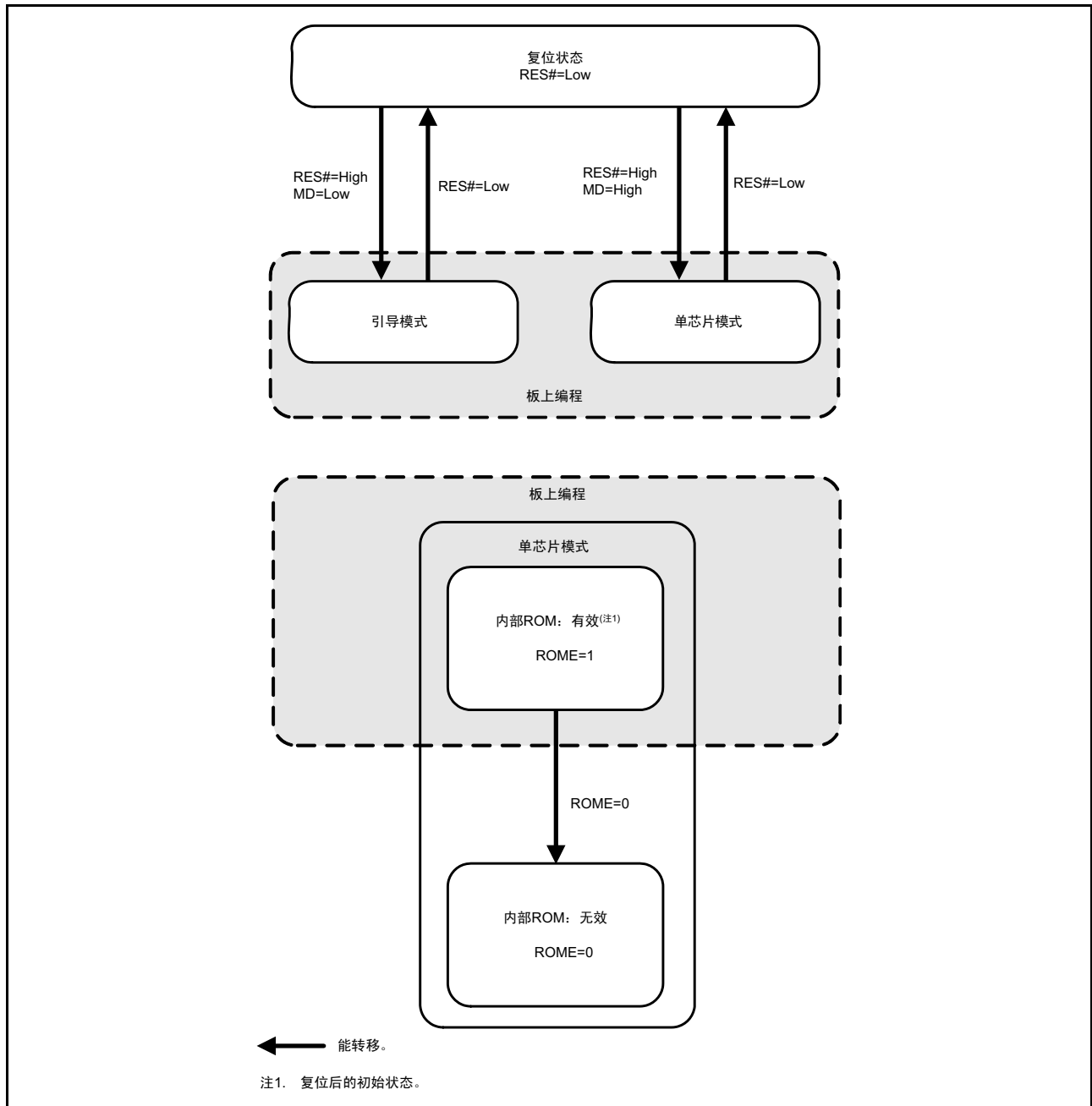


图 34.4 ROM 的运行模式转移图

在引导模式 / 单芯片模式（内部 ROM 有效）中，板上进行内部 ROM 的读 / 编程 / 擦除。

在各模式中，能编程 / 擦除的区域因复位时的启动区域等而不同。各模式的不同点如表 34.4 所示。

表 34.4 各模式的不同点

项目	引导模式	单芯片模式
编程 / 擦除环境	板上编程	
能编程 / 擦除的区域	用户区 / 数据区	用户区 / 数据区
块分割擦除	○（注 1）	○
复位时的启动程序	引导程序	用户程序

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“34.9.4 ID 码保护（引导模式）”、“34.9.2 引导模式的状态转移”。

- 在引导模式中，主机能经由SCI对用户区或者数据区进行编程、擦除和读。
- 在引导模式中，通过引导程序使用内部RAM。因此，不保持内部RAM的数据。

34.6 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和单芯片模式（内部 ROM 有效）中通用。

34.6.1 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 34.5 所示。

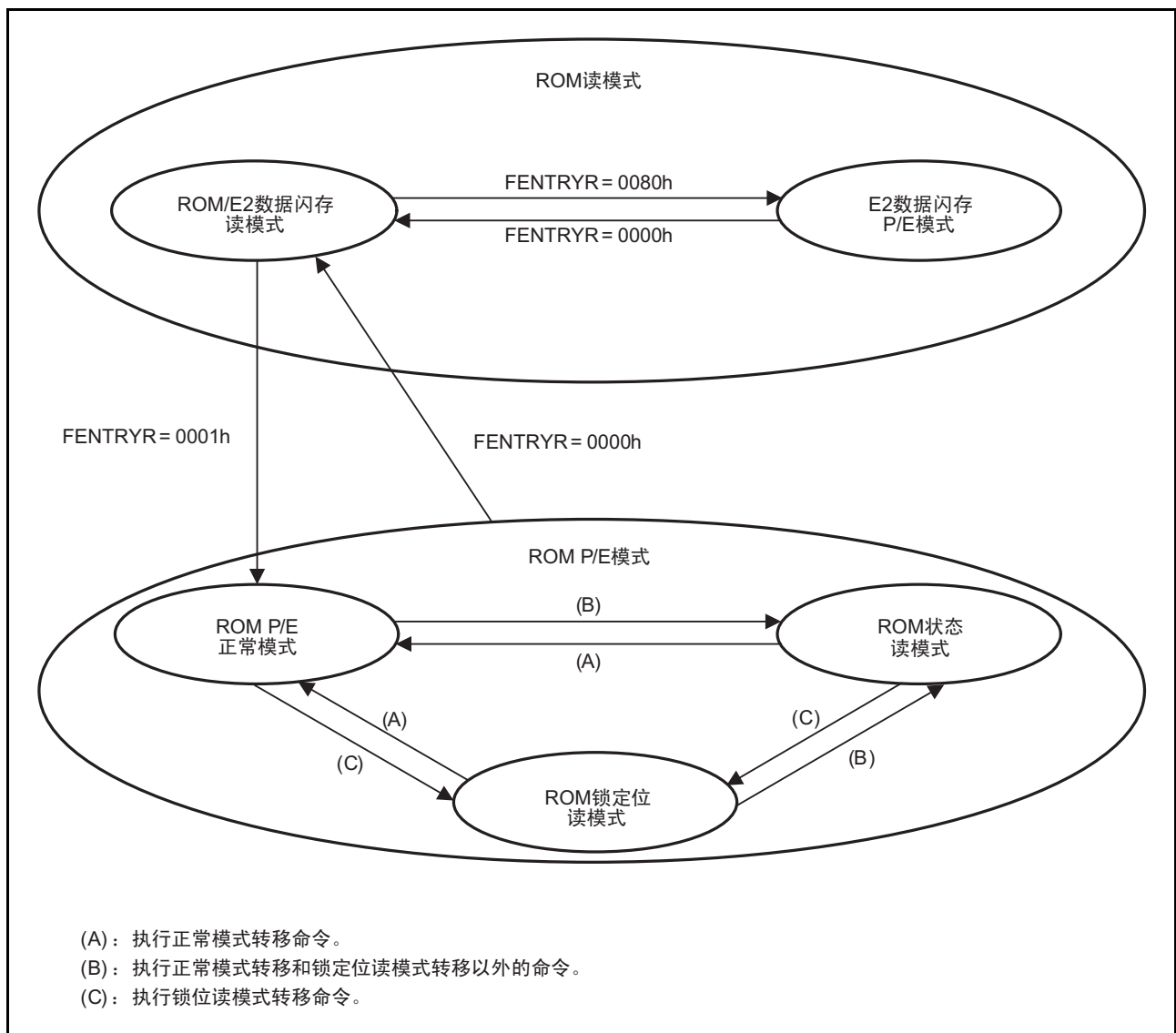


图 34.5 FCU 的模式转移图（ROM 相关）

34.6.1.1 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/E2 数据闪存读模式和 E2 数据闪存 P/E 模式。

(1) ROM/E2 数据闪存读模式

ROM/E2 数据闪存读模式是能读 ROM 和 E2 数据闪存的模式，不接受 FCU 命令。如果将 FENTRYR.FENTRY0 位和 FENTRYR.FENTRYD 位都置“0”，就转移到此模式。

(2) E2 数据闪存 P/E 模式

E2 数据闪存 P/E 模式是对 E2 数据闪存进行编程和擦除的模式，能高速读 ROM。此模式接受 E2 数据闪存的 FCU 命令，但是不接受 ROM 的 FCU 命令。如果将 FFENTRYR.FENTRY0 位置“0”并且将 FENTRYR.FENTRYD 位置“1”，就转移到此模式。

有关 E2 数据闪存 P/E 模式的详细内容，请参照“35. E2 数据闪存（保存数据的闪存）”的“35.6.1 FCU 的模式”。

34.6.1.2 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“34.8.2 命令锁定状态”）。

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.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 34.7 所示。

如果在 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“34.8.2 命令锁定状态”）。

(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 34.7 所示。

如果在 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 34.7 所示。

如果在 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。

34.6.2 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 34.5 所示。

表 34.5 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“34.6.3 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“34.6.3 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“34.6.3 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 128 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）/E2 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 E2 数据闪存的空白检查命令。如果对 E2 数据闪存发行锁定位读 2 命令，就进行 E2 数据闪存的空白检查（参照“35. E2 数据闪存（保存数据的闪存）”）。

通过对 ROM 的编程 / 擦除地址进行写存取来给 FCU 发行命令。FCU 命令的格式如表 34.6 所示。如果在 FCU 的特定条件下进行表 34.6 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“34.6.3 FCU 的模式和命令的关系”和“34.6.4 FCU 命令的使用方法”。

表 34.6 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 66 周期		第 67 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	0F0Fh	RA	0F0Fh	RA	D0h	—	—	—	—
编程	67	RA	E8h	RA	40h	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 RA: ROM 的编程 / 擦除地址

当ROM容量为64K字节并且FENTRYR.FENTRY0位为“1”时,为00FF 0000h~00FF FFFFh的任意地址;

当ROM容量为48K字节并且FENTRYR.FENTRY0位为“1”时,为00FF 4000h~00FF FFFFh的任意地址;

当ROM容量为32K字节并且FENTRYR.FENTRY0位为“1”时,为00FF 8000h~00FF FFFFh的任意地址。

WA: ROM 的编程目标地址

编程数据 128 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 ~ 64)

34.6.3 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 34.7 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“34.8.2 命令锁定状态”）。

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.FCUEERR 位的值，然后发行 FCU 命令。能根据 FASTAT.CMDLK 位的值确认有没有发生错误。FASTAT.CMDLK 位的值是 FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位和 FSTATR1.FCUEERR 位的值的逻辑或。

表 34.7 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	擦除挂起中的编程处理中	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 (FRDY=0)	命令锁定状态 (FRDY=1)	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	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	0	0
FSTATR0.ERSSPD 位	0	1	0	0	1	0/1	0/1	0	1	0/1	0/1	0	0	1	0
FSTATR0.PRGSPD 位	1	0	0	0	0	0/1	0/1	1	0	0/1	0/1	0	1	0	0
FASTAT.CMDLK 位	0	0	0	0	0	0	0	0	0	1	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	×	Δ	○	×	×	×	×	×	Δ	×	×	○	×	Δ	○
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	×	Δ	○	×	×	×	×	×	Δ	×	×	○	×	Δ	○

○：能接受、Δ：只能接受对非擦除中止块进行的编程、×：不能接受

34.6.4 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“34.6.3 FCU 的模式和命令的关系”。

34.6.4.1 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 34.5。

(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.FENTRY0 位置“1”。

在进行编程或者擦除时，必须以字节为单位给 FWEPROR 寄存器写“01h”，进入能进行编程和擦除的状态（参照“34.2.1 闪存编程 / 擦除保护寄存器（FWEPROR）”）。

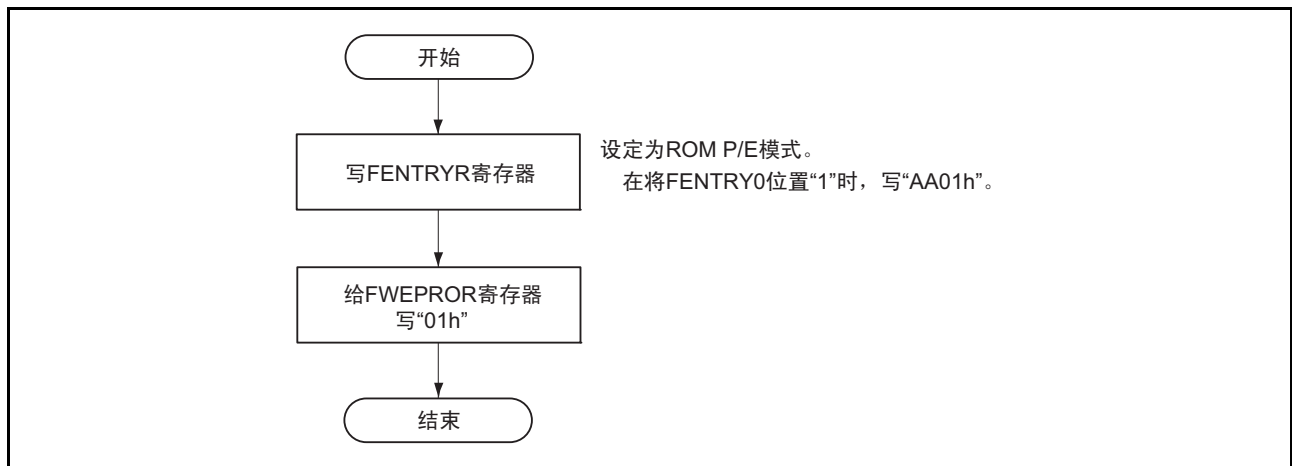


图 34.6 ROM P/E 模式的转移流程

(2) ROM 读模式的转移方法

要高速读 ROM 时，需要将 FENTRYR.FENTRY0 位置“0”，将 FCU 设定为 ROM 读模式。

还需要以字节为单位给 FWEPROR 寄存器写“02h”，进入不能编程和擦除的状态（参照“34.2.1 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。

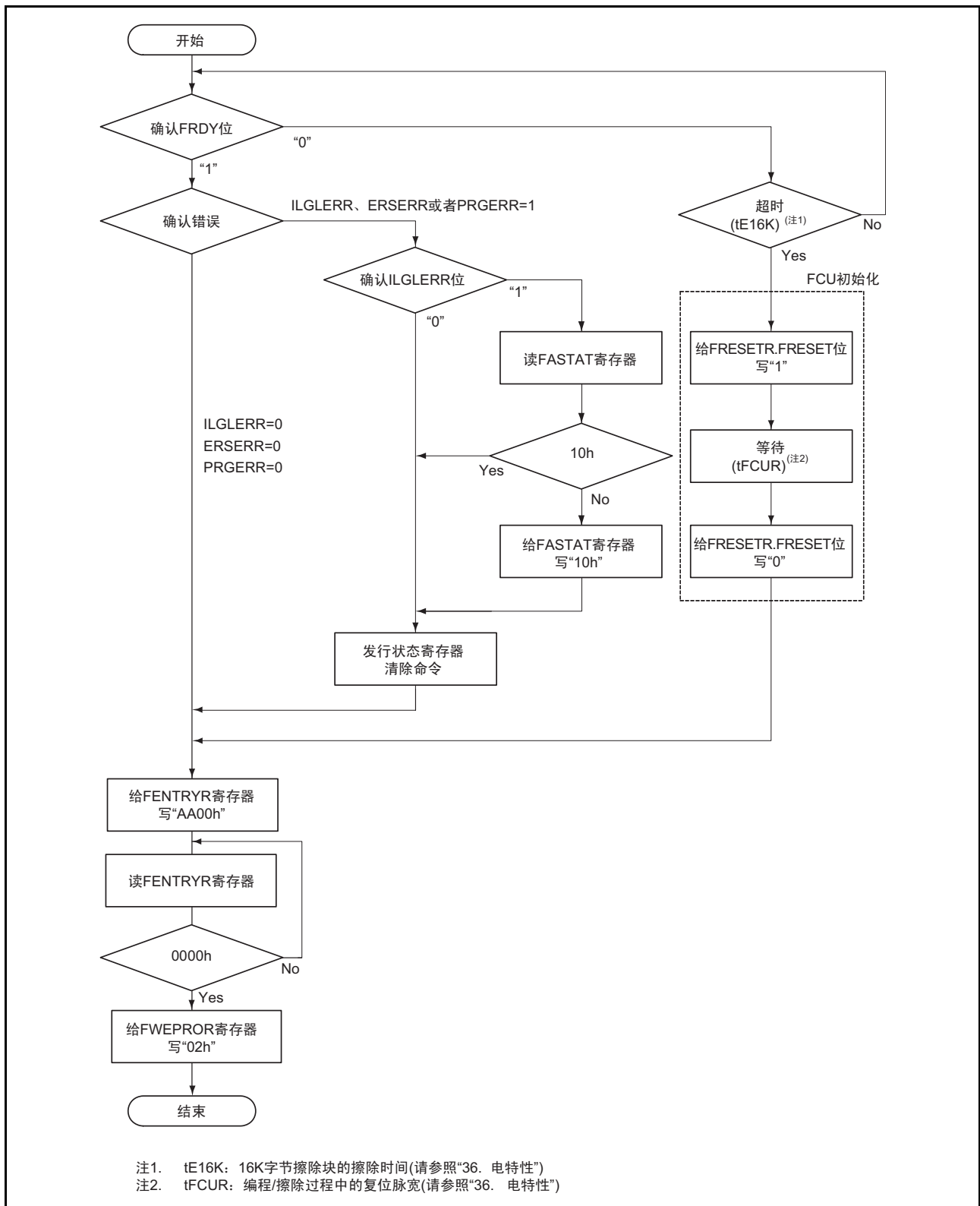


图 34.7 ROM 读模式的转移流程

(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“34.6.1 FCU 的模式”）以及在 ROM P/E 模式中发行正常模式转移命令（图 34.8）的方法。以字节为单位给 ROM 的编程 / 擦除地址写“FFh”，执行正常模式转移命令。

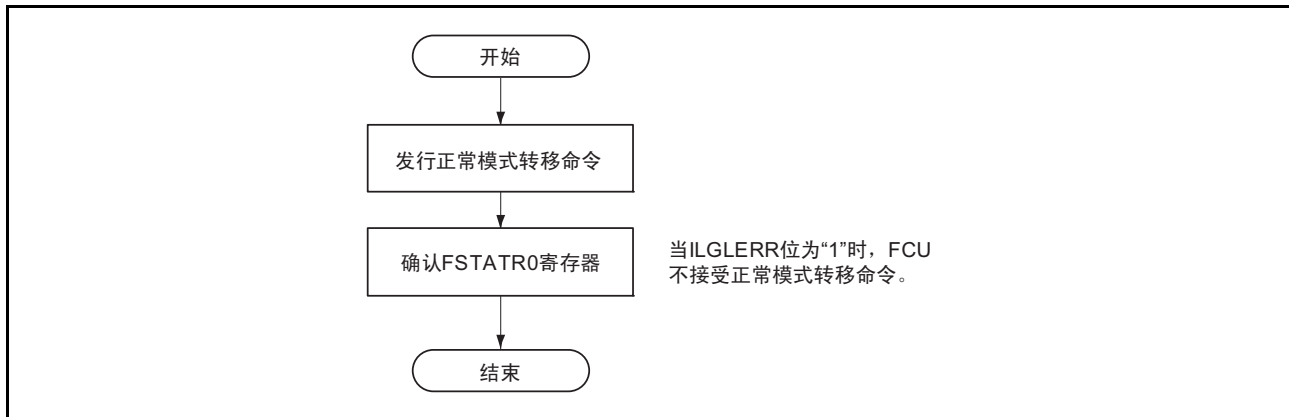


图 34.8 ROM P/E 正常模式的转移流程

(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 34.9 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 的编程 / 擦除地址进行读存取，确认 FSTATR0 的内容。

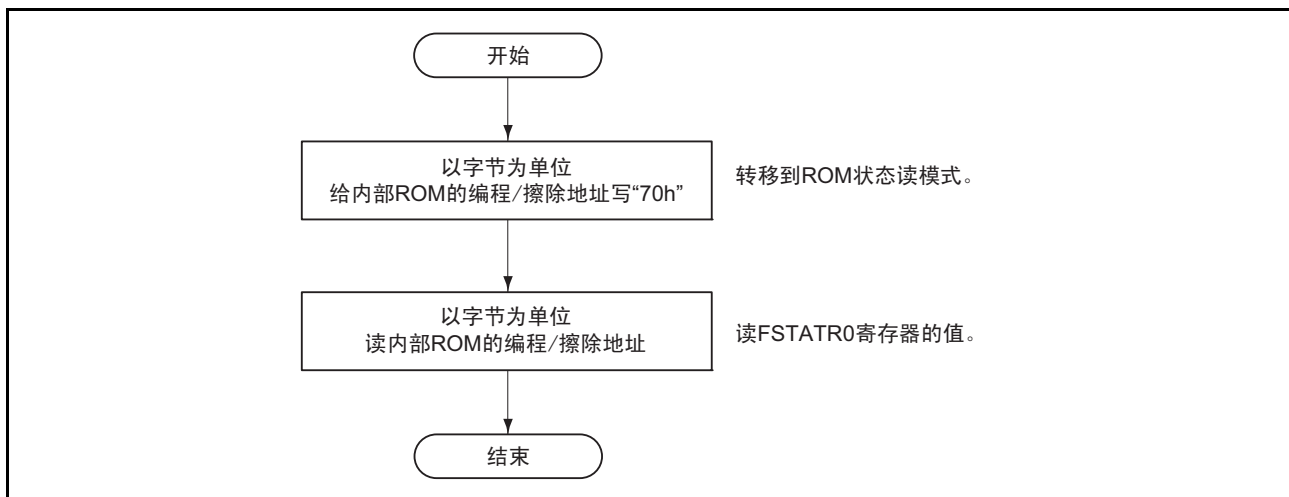


图 34.9 ROM 状态读模式的转移流程以及状态的确认方法

(5) ROM 锁定位读模式的转移方法

在 FMODR.FRDMMD 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 锁定位读模式后对 ROM 的编程 / 擦除地址进行读存取，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 34.10）。

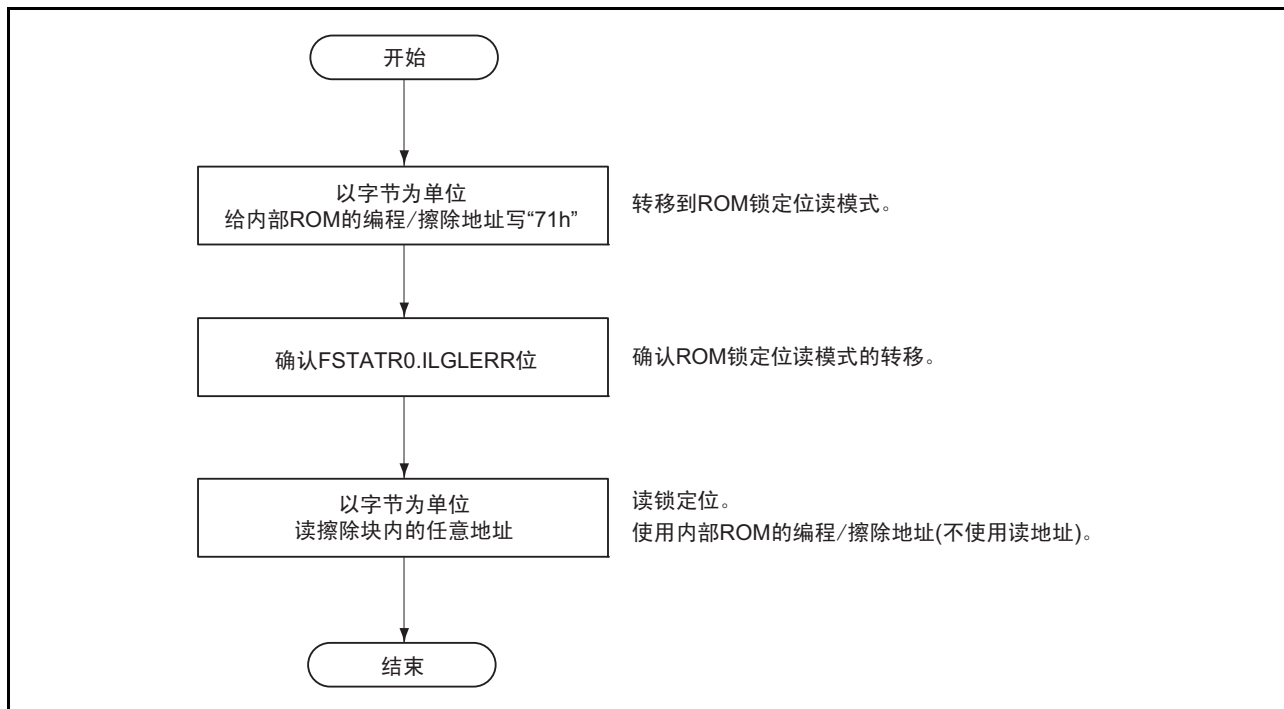


图 34.10 ROM 锁定位读模式的转移流程以及锁定位的读法

34.6.4.2 编程 / 擦除的方法步骤

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“34.6.3 FCU 的模式和命令的关系”。FCU 命令的概略流程如图 34.11 所示。

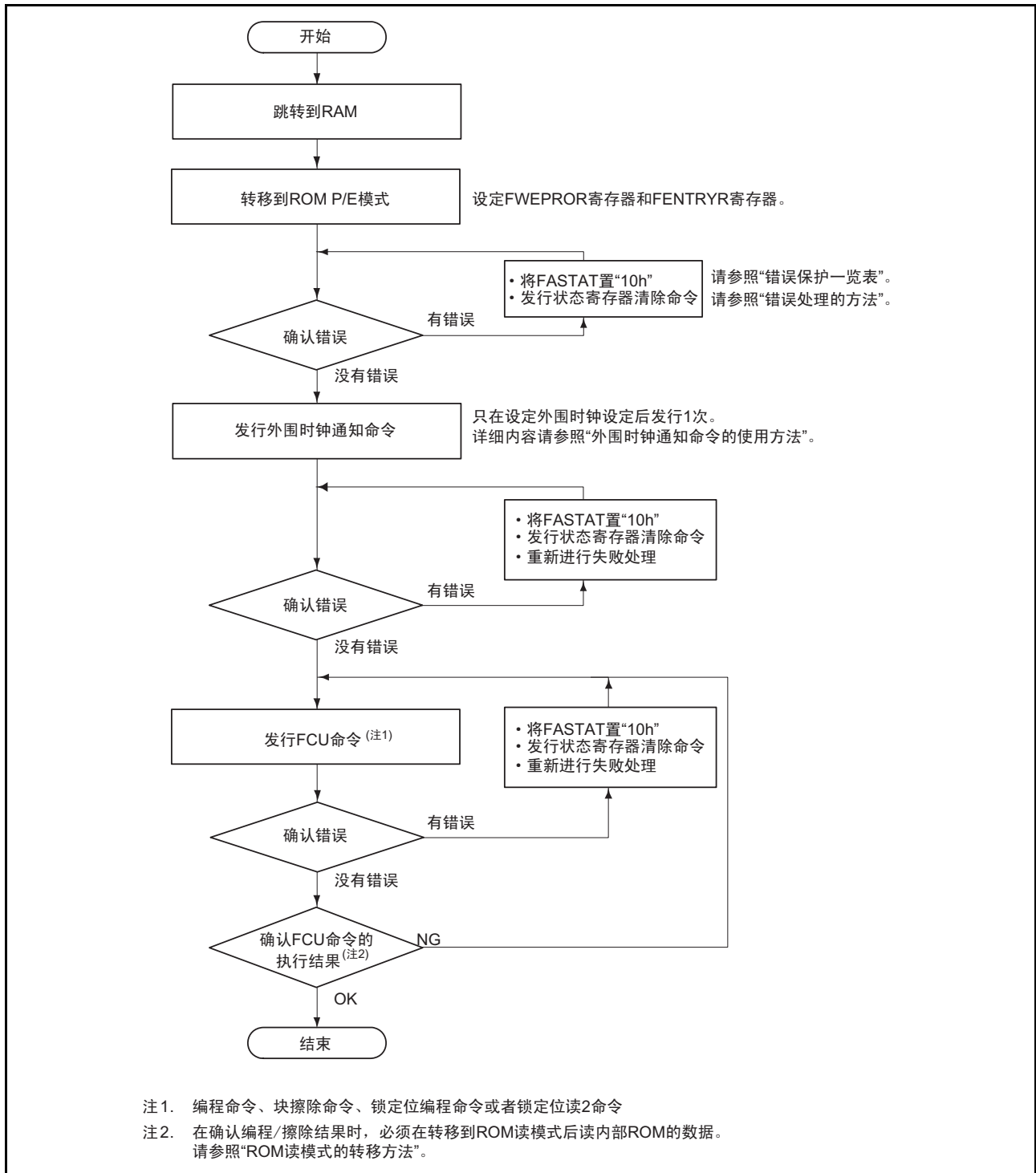


图 34.11 编程 / 擦除处理的概略流程

(1) 向 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 RAM 后跳转到 RAM。

(2) ROM P/E 模式的转移

需要设定 FENTRYR.FENTRY0 位和 FWEPROR 寄存器，并且将 FCU 设定为 ROM P/E 模式。详细内容请参照“34.6.4.1 (1) ROM P/E 模式的转移方法”。

(3) 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 FlashIF 时钟（FCLK）的频率。能设定的频率范围是 1 ~ 100MHz。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“34.8.2 命令锁定状态”）。另外，如果将 PCKAR 寄存器的 PCKA[7:0] 位设定在 4MHz ~ 50MHz 范围外，就不能对 ROM/E2 数据闪存发行改写命令。

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，以字节为单位给 ROM 的编程 / 擦除地址写“E9h”；在第 2 周期，以字节为单位给 ROM 的编程 / 擦除地址写“03h”。在命令的第 3 周期~第 5 周期，以字为单位进行写操作。在进行 3 次以字为单位给 ROM 的编程 / 擦除地址写数据“0F0Fh”后，如果在第 6 周期以字节为单位给 ROM 的编程 / 擦除地址写“D0h”，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.FENTRY0 位的设定而不同，必须指定与 FENTRYR.FENTRY0 位对应的地址。如果因 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“34.8.2 命令锁定状态”）。

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。

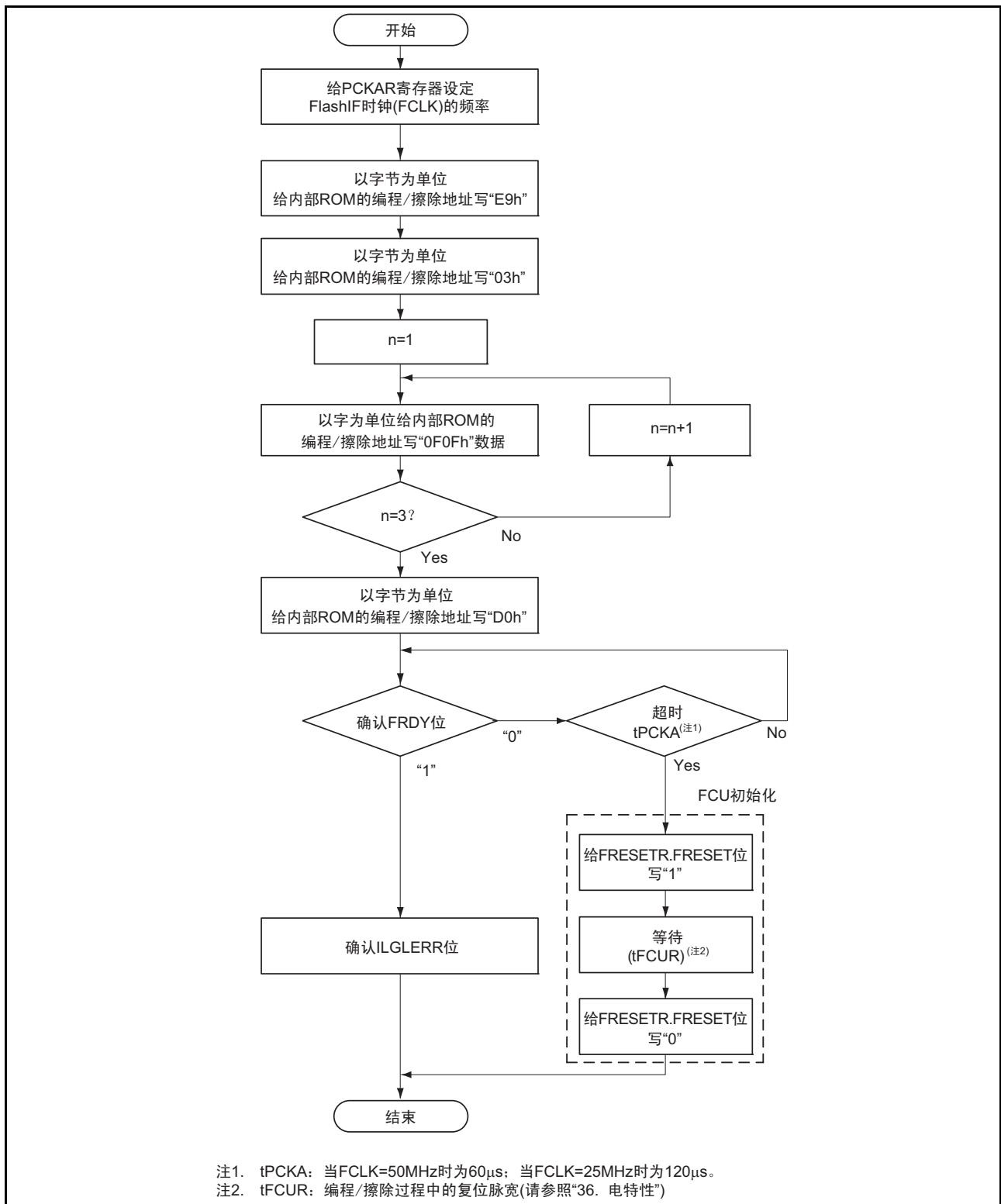


图 34.12 外围时钟通知命令的使用方法

(4) 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，以字节为单位给 ROM 的编程 / 擦除地址写“E8h”；在第 2 周期，以字节为单位给 ROM 的编程 / 擦除地址写“40h”。在第 3 周期的存取中，必须以字为单位给编程对象区的起始地址写编程数据。此时，起始地址必须使用调整为以 128 字节为边界的地址。在第 4 周期~第 66 周期，必须进行 63 次以字单位给 ROM 的编程 / 擦除地址写编程数据。如果在第 67 周期以字节为单位给 ROM 的编程 / 擦除地址写“D0h”，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 67 周期中能指定的地址因 FENTRYR.FENTRY0 位的设定而不同，必须指定与 FENTRYR.FENTRY0 位对应的地址。如果因 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“34.8.2 命令锁定状态”）。

如果在第 3 周期~第 66 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”。

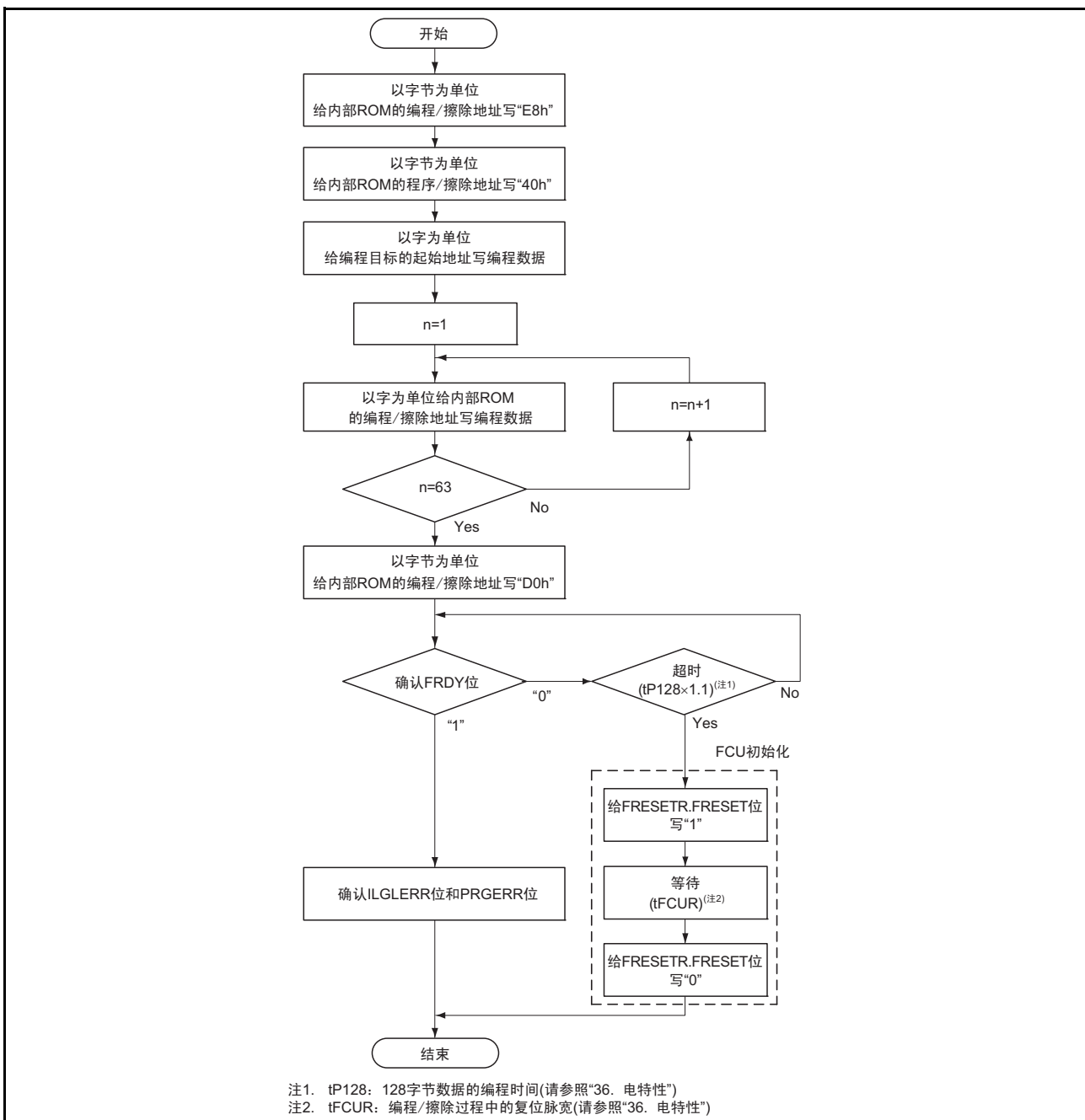


图 34.13 ROM 的编程方法

(5) 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，以字节为单位给 ROM 的编程 / 擦除地址写“20h”。如果在第 2 周期通过以字节为单位给擦除对象块内的任意地址写“D0h”，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。如果通过 CPU 读擦除状态的 ROM，就以 32 位为单位读到“FFFF FFFFh”。

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，就必须在将 FPROTR.FPROTCN 位置位后进行擦除。

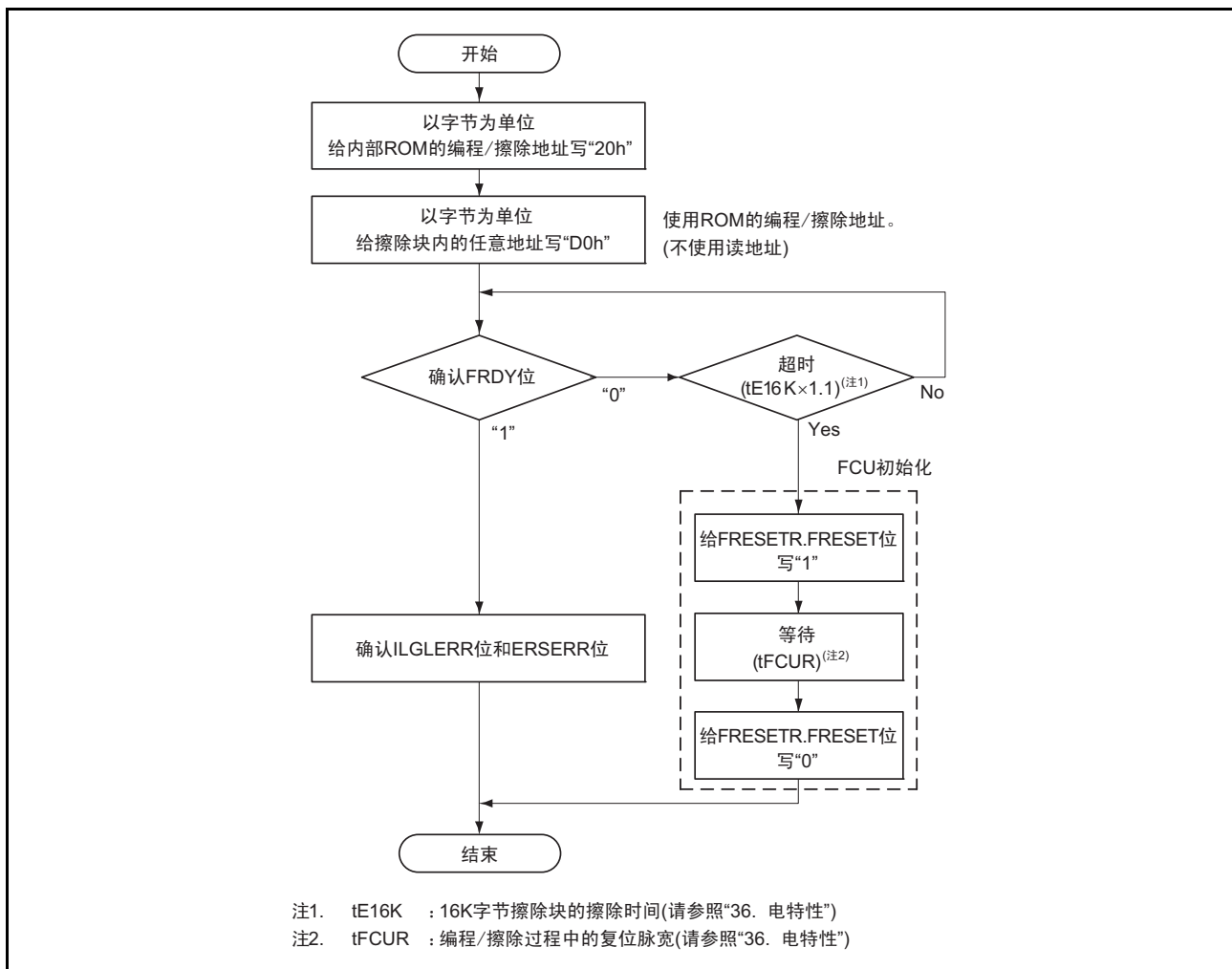


图 34.14 ROM 的擦除方法

(6) 锁定位的编程 / 擦除方法

用户区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，以字节为单位给 ROM 的编程 / 擦除地址写“77h”。如果在第 2 周期以字节为单位给要进行锁定位编程的擦除块内的任意地址写“D0h”，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。

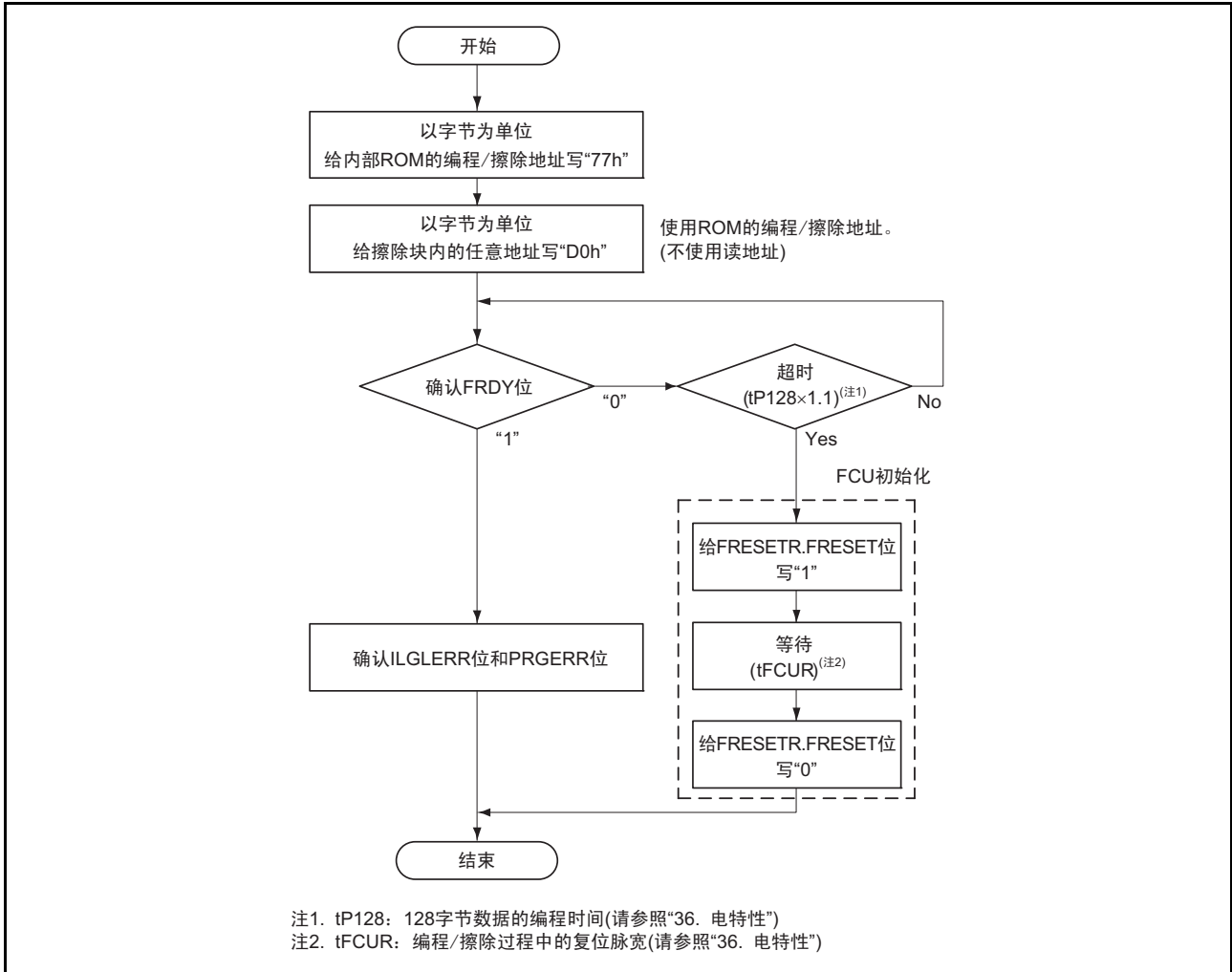


图 34.15 锁定位的编程设定方法

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。

(7) 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/擦除地址发行锁定读2命令。如果在锁定读2命令的第1周期以字节为单位写“71h”，在第2周期以字节为单位写“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“34.6.4.1 (5) ROM 锁定位读模式的转移方法”。

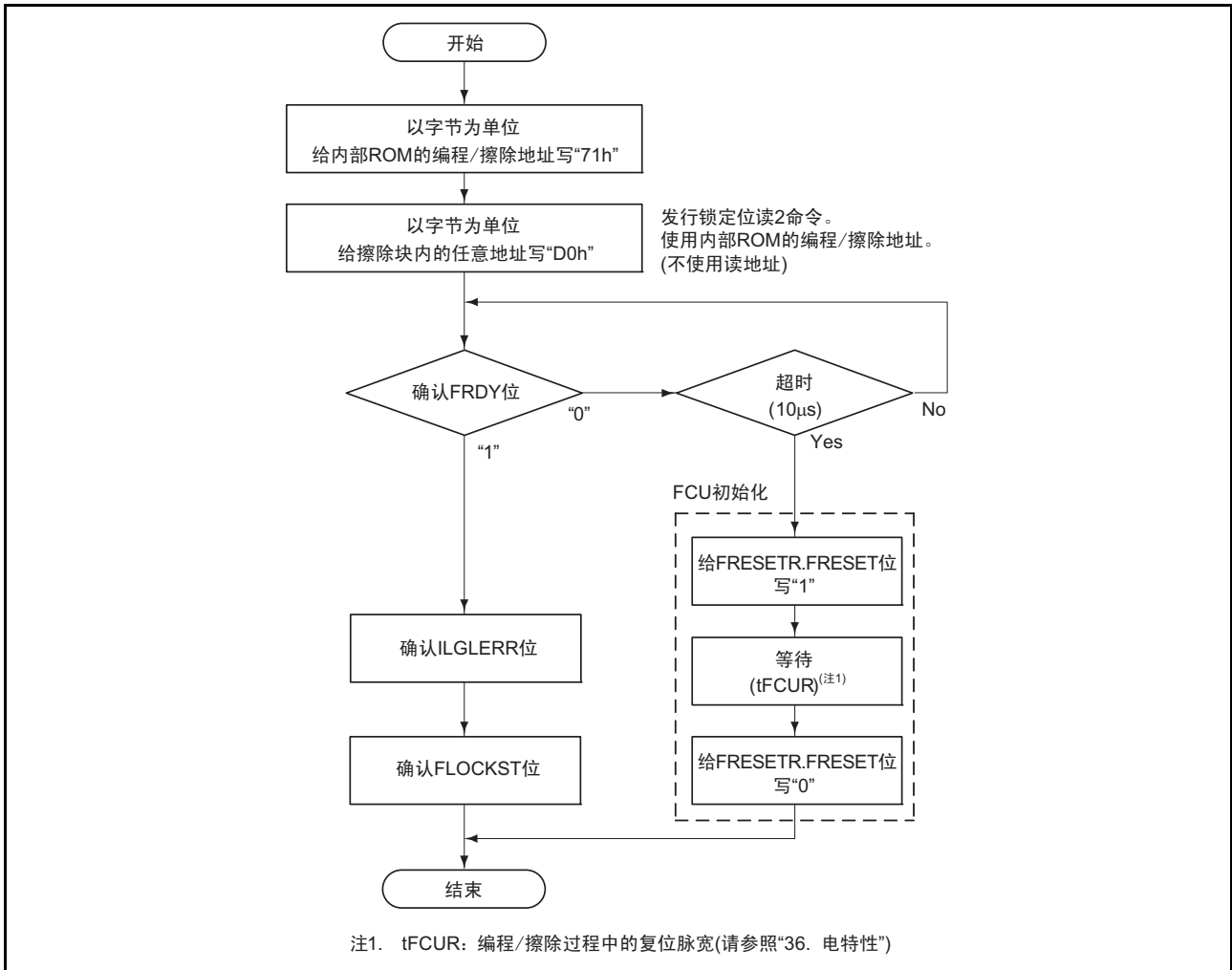


图 34.16 通过寄存器读方式读锁定位的方法

34.6.4.3 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“34.8 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“34.6.4.1 (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 FASTAT.DFLWPE 位的值。即使不清除这些位而发行状态寄存器清除命令，ILGLERR 位也不变为“0”。

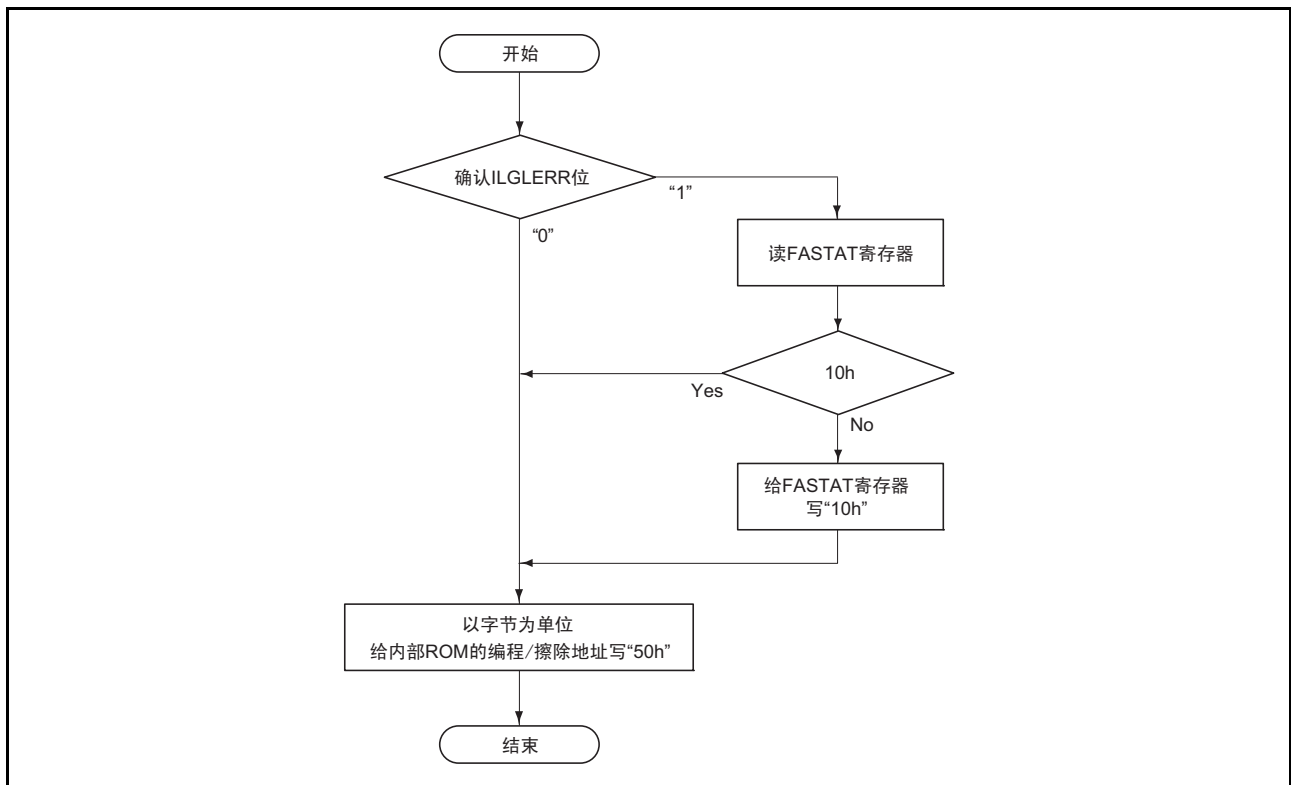


图 34.17 FSTATR0 寄存器的清除方法

(3) FCU 的初始化方法

在发行 FCU 命令后，如果因超时而使 FSTATR0.FR DY 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。无论在哪种情况下，FRESETR.FRESET 位为“1”的状态都必须保持 tFCUR 的时间（参照“36. 电特性”）。在 FRESET 位保持“1”的期间，禁止读 ROM/E2 数据闪存。因为在 FRESET 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 34.11 的处理。

34.6.4.4 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 位是“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 位是“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.FRNDY 位为“1”并且 FSTATR0.ERSSPD 位和 FSTATR0.PRGSPD 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，FRNDY 位变为“1”并且 ERSSPD 位或者 PRGSPD 位变为“1”。在发行 P/E 挂起命令后，必须通过检查 ERSSPD 位或者 PRGSPD 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，只要在后续的流程中发行 P/E 恢复命令，也发生非法命令错误，FCU 转移到命令锁定状态（参照“34.8.2 命令锁定状态”）。

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“34.7 挂起”。

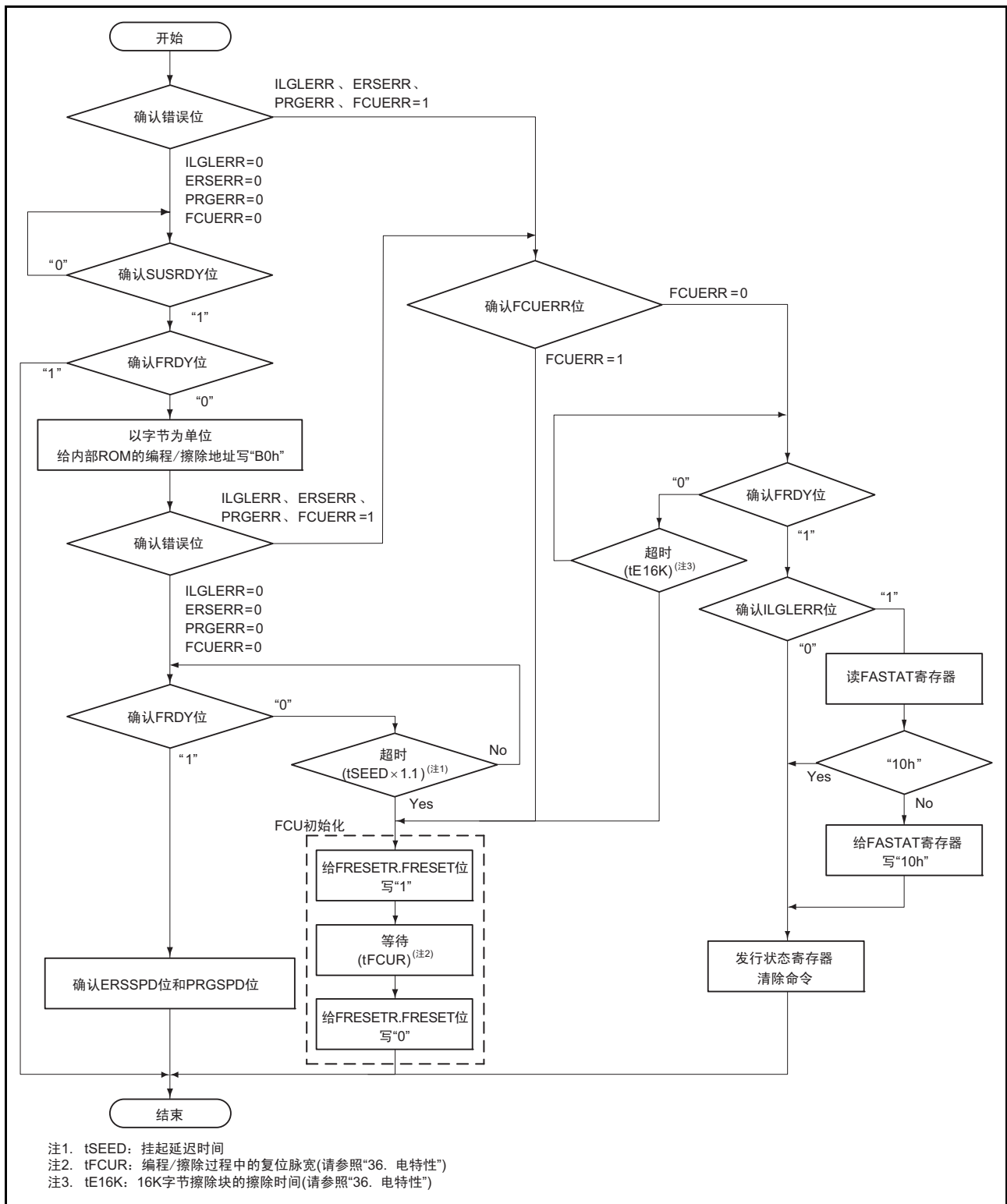


图 34.18 编程 / 擦除的挂起方法

(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。

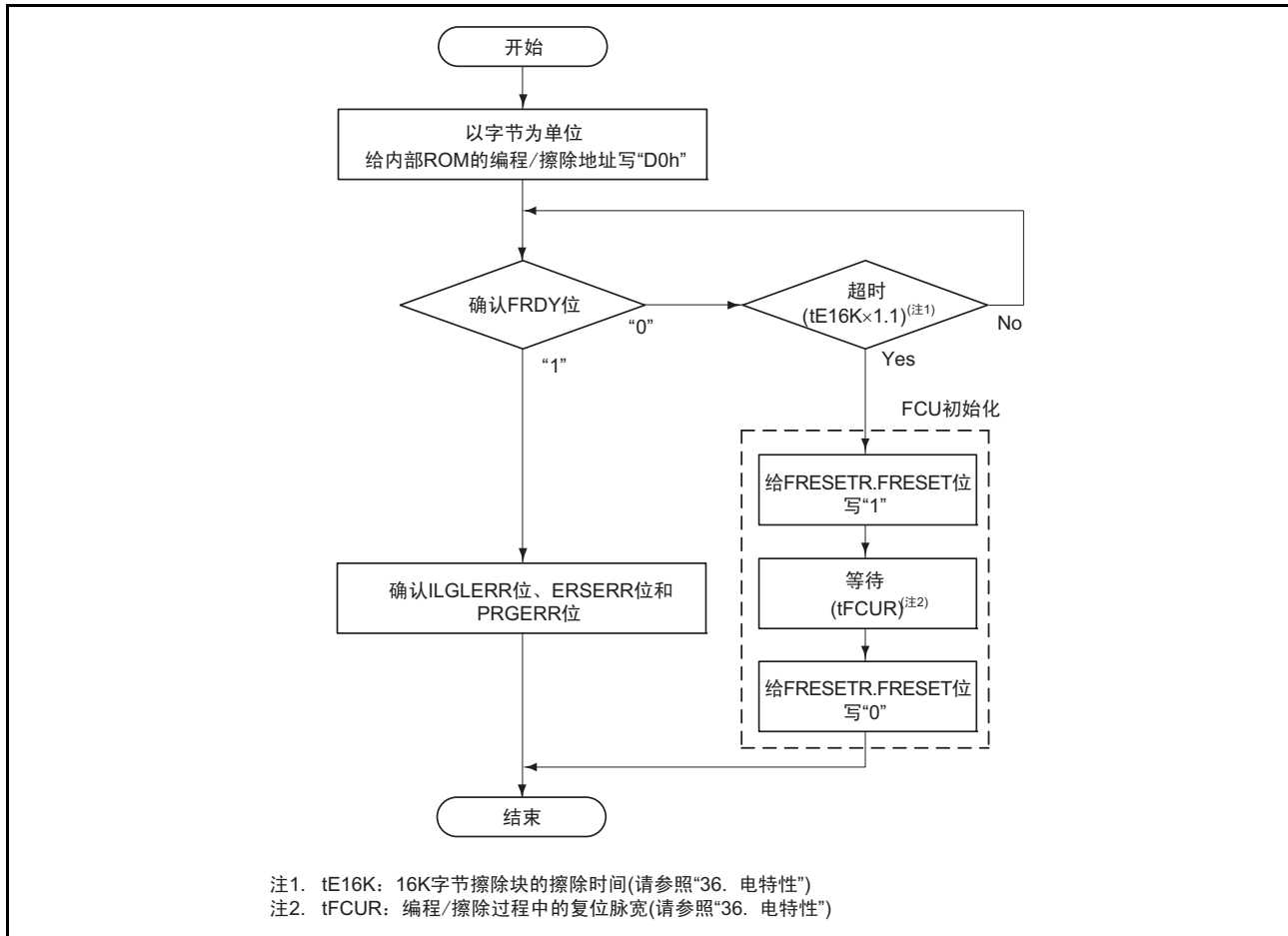


图 34.19 编程 / 擦除的恢复方法

34.7 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、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。

34.7.1 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 34.20 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，FCU 将 FSTATR0.SUSRDY 位置“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。

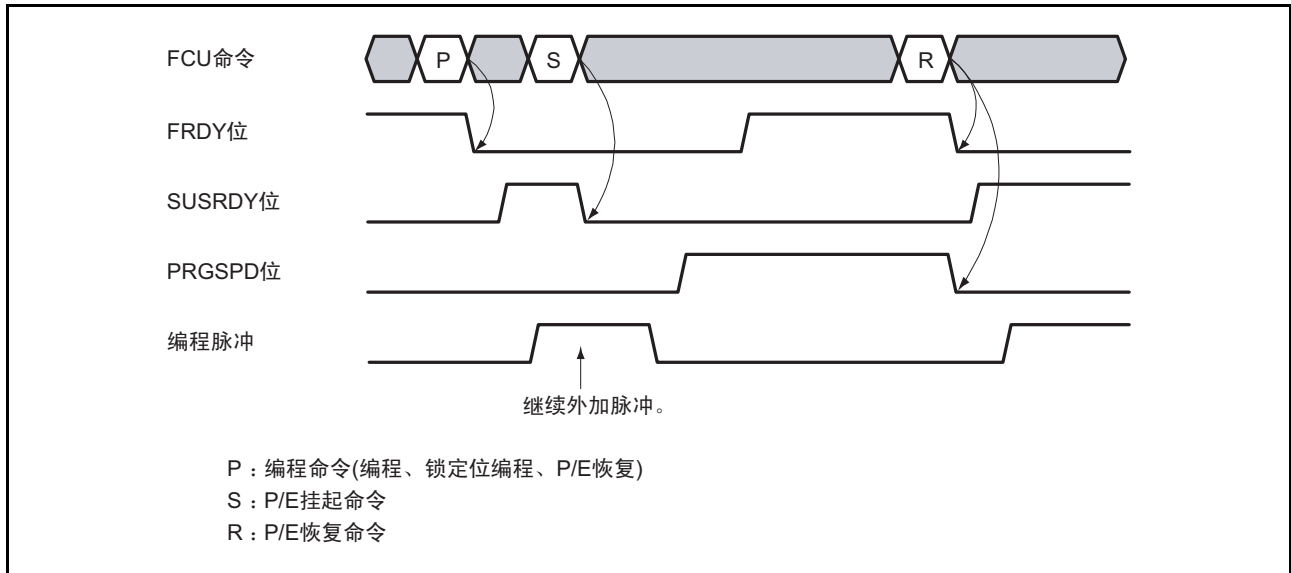


图 34.20 编程处理的中止

34.7.2 编程和擦除时的挂起（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 34.21 所示。

如果 FCU 接受擦除命令，就将 FSTATR0.FRDY 位清“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，FCU 将 FSTATR0.SUSRDY 位置“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 FSTATR0.ERSSPD 位置“1”。一旦中止处理结束，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。中止或者重新开始擦除处理时的 FRDY 位、SUSRDY 位和 ERSSPD 位的操作相同，而不取决于擦除挂起模式。

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。

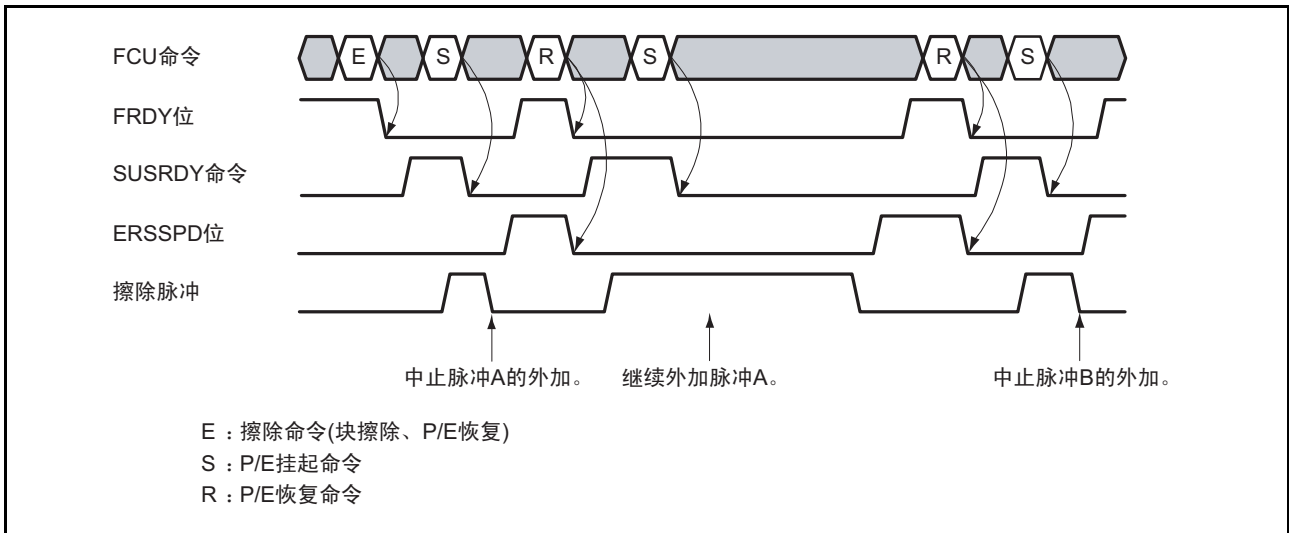


图 34.21 擦除处理的中止（挂起优先模式）

34.7.3 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 34.22 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。

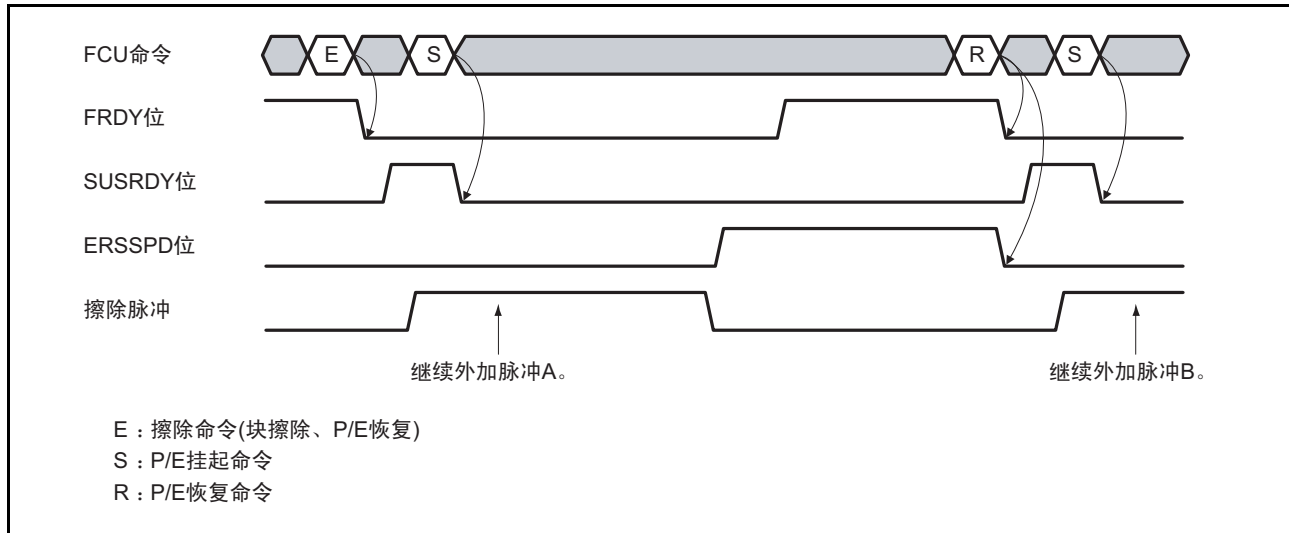


图 34.22 擦除处理的中止（擦除优先模式）

34.8 保护

ROM 的编程 / 擦除保护有 2 种，分别是软件保护和命令锁定状态。

34.8.1 软件保护

软件保护是通过设定控制寄存器和用户区的锁定位来禁止对 ROM 进行编程 / 擦除的状态。如果违反软件保护而发行 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.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“34.8.2 命令锁定状态”）。

(3) 通过锁定位进行的保护

用户区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而发行 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“34.8.2 命令锁定状态”）。

34.8.2 命令锁定状态

命令锁定状态是因误发行 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。

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、FASTAT.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 相关的命令锁定状态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、FASTAT.ROMAE 位）的关系如表 34.8 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 34.8 命令锁定状态的一览表（ROM 专用 +ROM/E2 数据闪存共用）

分类	内容	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE
FENTRYR 的设置错误	给 FENTRYR 寄存器设定“0001h”和“0008h”以外的值。	1	0	0	0	0
	挂起和恢复时的 FENTRYR 寄存器的设定不同。	1	0	0	0	0
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
	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
	给 PCKAR 寄存器设定 1 ~ 100MHz 以外的外围时钟。 (在设定 1 ~ 4MHz 和 50 ~ 100MHz 时, 不检测错误)	1	0	0	0	0
	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
	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
	在挂起状态下发行挂起命令。	1	0	0	0	0
	在挂起以外的状态下发行恢复命令。	1	0	0	0	0
	在编程挂起状态下发行编程 / 擦除 (编程、锁定位编程、块擦除) 命令。	1	0	0	0	0
	在擦除挂起状态下发行块擦除命令。	1	0	0	0	0
	在擦除挂起状态下发行擦除挂起对象区的编程命令 / 锁定位编程命令。	1	0	0	0	0
	在编程命令的第 2 个周期指定“80h”以外的数据。	1	0	0	0	0
	在命令锁定状态下发行命令。	1	0/1	0/1	0/1	0/1
擦除错误	在擦除处理过程中发生错误。	0	1	0	0	0
	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
编程错误	在编程处理过程中发生错误。	0	0	1	0	0
	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位编程命令。	0	0	1	0	0
FCU 错误	在 FCU 内部处理过程中发生错误。	0	0	0	1	0
ROM 存取违反	在 FENTRYR.FENTRY0 位为“1”时并且在 ROM P/E 正常模式中, 当用户区的容量为 64K 字节时, 对 00FF 0000h ~ 00FF FFFFh 发行读存取命令; 当用户区的容量为 48K 字节时, 对 00FF 4000h ~ 00FF FFFFh 发行读存取命令; 当用户区的容量为 32K 字节时, 对 00FF 8000h ~ 00FF FFFFh 发行读存取命令。	1	0	0	0	1
	在 FENTRYR.FENTRY0 位为“0”的情况下, 当用户区的容量为 64K 字节时, 对 00FF 0000h ~ 00FF FFFFh 发行读存取命令; 当用户区的容量为 48K 字节时, 对 00FF 4000h ~ 00FF FFFFh 发行读存取命令; 当用户区的容量为 32K 字节时, 对 00FF 8000h ~ 00FF FFFFh 发行读存取命令。	1	0	0	0	1
	在设定 FENTRYR 寄存器后转移到 ROM P/E 模式的状态下, 对 FFFF 0000h/FFFF 4000h/FFFF 8000h (注 1) ~ FFFF FFFFh 发行读存取命令。	1	0	0	0	1

注 1. 当用户区的容量为 64K 字节时, 为“FFFF 0000h”; 当用户区的容量为 48K 字节时, 为“FFFF 4000h”; 当用户区的容量为 32K 字节时, 为“FFFF 8000h”。

34.9 引导模式

34.9.1 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，对用户区或者数据区进行编程和擦除。为了进行主机和 RX63T 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 RX63T，就执行引导区中的程序。引导区中的程序在自动调整 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 34.23 所示。

内部 ROM 的相关输入 / 输出引脚如表 34.9 所示。

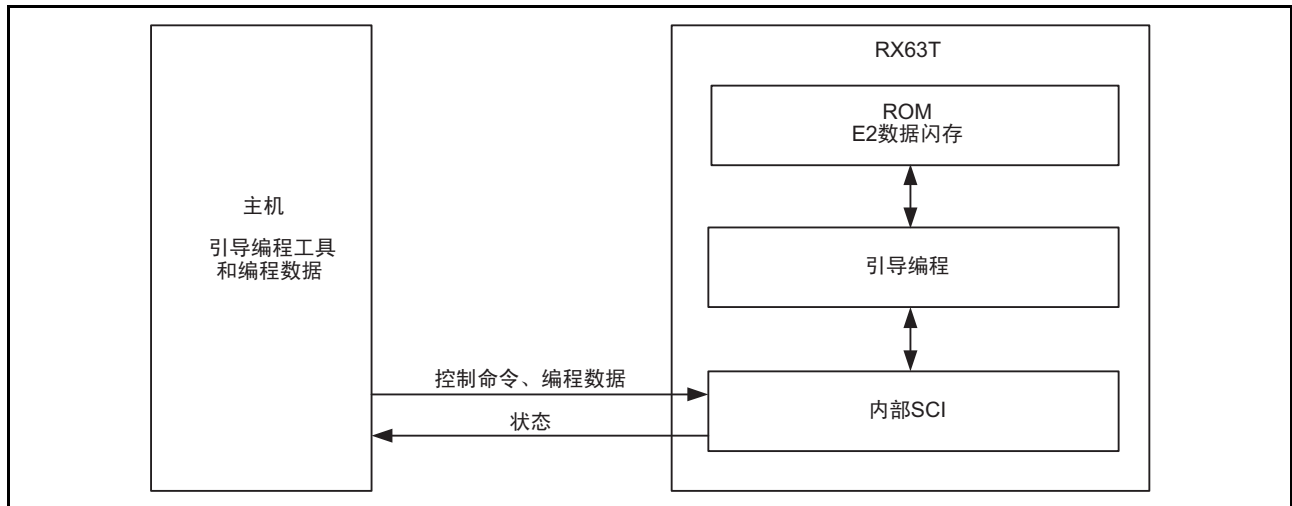


图 34.23 引导模式的系统结构

表 34.9 内部 ROM 的相关输入 / 输出引脚

引脚名称	输入 / 输出	功能
PD5/RXD1	输入	用于引导模式时的 SCI 的接收数据（用于主机通信）。
PD3/TXD1	输出	用于引导模式时的 SCI 的发送数据（用于主机通信）。
MD	输入	设定运行模式。

34.9.2 引导模式的状态转移

引导模式的状态转移图如图 34.24 所示。

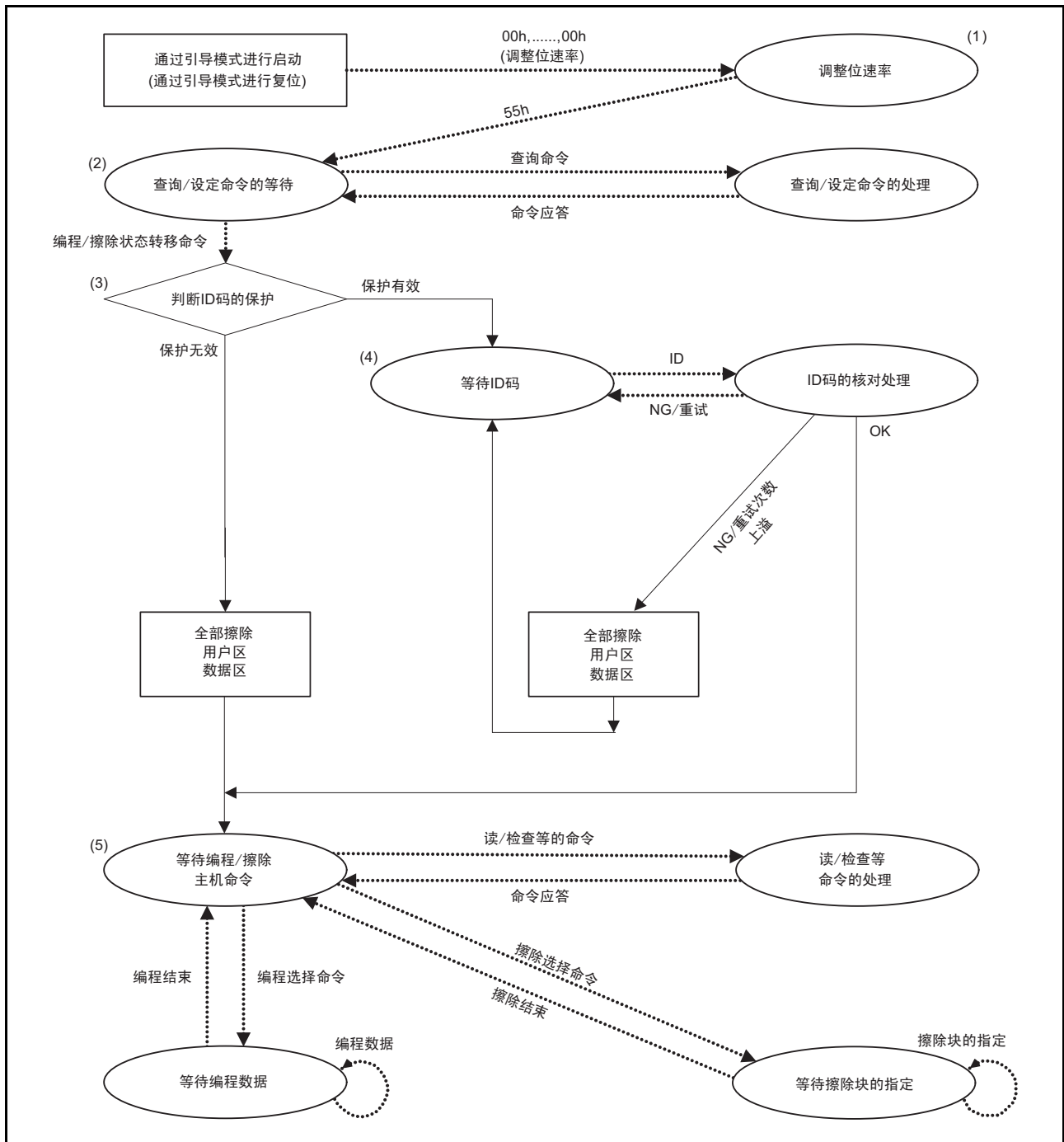


图 34.24 引导模式的状态转移图

(1) 位速率的调整

如果通过引导模式启动 RX63T，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 RX63T 发送到主机。此后，如果 RX63T 正常接收从主机送来的“55h”，就转移到查询 / 设定命令的等待状态。有关位速率调整的详细内容，请参照“34.9.3 位速率的自动调整”。

(2) 查询 / 设定命令的等待

这是查询区域大小、区域结构、区域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。有关查询设定命令的详细内容，请参照“34.9.5 查询 / 设定命令的等待”。

(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，全部擦除用户区和数据区，并且转移到编程 / 擦除命令等待状态。有关控制码和 ID 码的详细内容，请参照“34.9.4 ID 码保护（引导模式）”。

(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“34.9.4 ID 码保护（引导模式）”。

(5) 编程 / 擦除命令的等待

这是根据主机的命令进行编程或者擦除的状态。根据 RX63T 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 RX63T 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 RX63T 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 / 擦除命令状态的期间，用户区和数据区被全部擦除，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，还有用户区的校验和命令、空白检查（擦除检查）命令、存储器读命令、状态信息取命令等命令。

34.9.3 位速率的自动调整

如果通过引导模式启动 RX63T，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、没有奇偶校验，位速率必须设定为 9600bps 或者 19200bps。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，RX63T 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 RX63T。否则，就必须通过引导模式重新启动 RX63T，并且重新自动调整位速率。如果 RX63T 正常接收“55h”，就发送“E6h”。否则，就发送“FFh”。

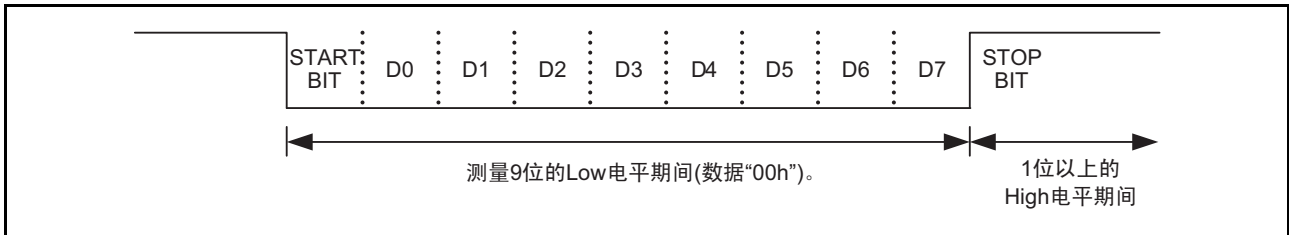


图 34.25 自动调整位速率时的 SCI 发送 / 接收格式



图 34.26 主机和 RX63T 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 RX63T 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 34.10 所示的条件下进行 SCI 的通信。

表 34.10 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
9600bps	4 ~ 20MHz（注 1）
19200bps	8 ~ 20MHz（注 1）

注 1. 但是，谐振器频率最大为 16MHz。

34.9.4 ID 码保护（引导模式）

这是禁止从 PC 等主机读、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 34.27 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 34.27 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 34.11 所示。

表 34.11 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: 转移到命令等待。 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: 转移到命令等待。 ID 码不同: 再次转移到 ID 码等待。
	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 开始依次设定“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 开始按依次设定“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
```

34.9.5 查询 / 设定命令的等待

在查询 / 设定命令等待状态下能使用的命令一览表如表 34.12 所示。在编程 / 擦除命令等待状态下，也能使用引导程序状态查询命令。其他命令只能在查询设定主机命令等待状态下使用。

表 34.12 查询 / 设定命令

命令名	功能
支持器件查询	查询器件码和系列名的产品型号。
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。
用户区信息查询	查询用户区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 RX63T 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。
引导程序状态查询	查询处理状态。

如果主机发送未定义的命令，RX63T 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 RX63T。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，RX63T 就发送命令错误应答。查询 / 设定命令等待状态下的主机命令使用例子如图 34.28 所示。

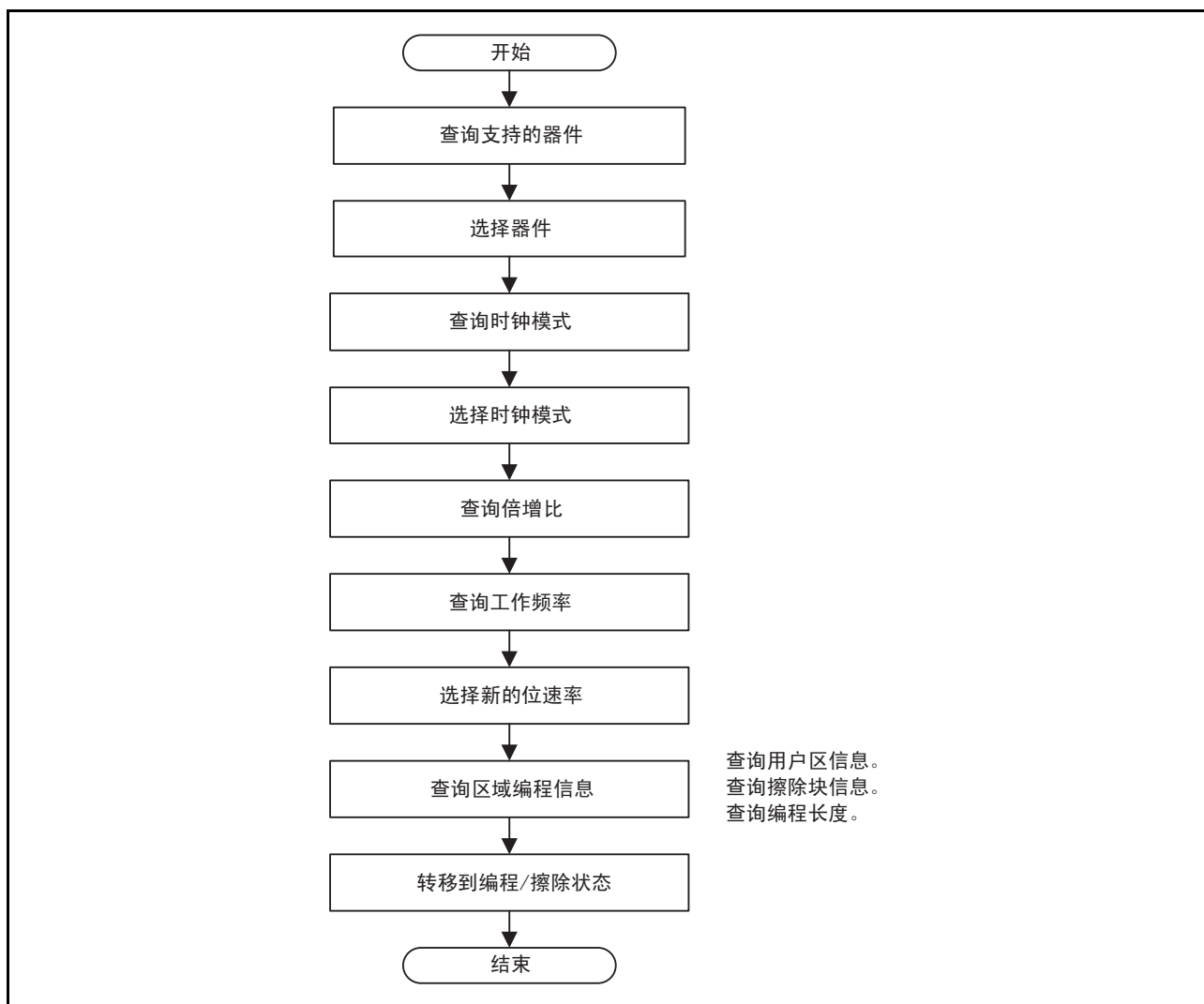


图 34.28 用户区 / 查询 / 设定命令的使用例子

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 RX63T 的命令，“应答”是从 RX63T 发送到主机的应答。“校验和”是指在合计 RX63T 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，RX63T 就通过引导编程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，RX63T 就只发送所选的器件信息。作为支持器件查询命令的应答，RX63T 按各自的顺序发送小端法指定和大端法指定的 2 个器件信息。

命令	20h		
应答	30h	长度	器件数
	字符数	器件码（小端法指定）	
	字符数	器件码（大端法指定）	
	SUM	系列名	

- 长度（1 字节）：器件数、字符数、器件码和系列名的数据总字节数
 器件数（1 字节）：引导编程支持的品种数
 字符数（1 字节）：器件码和系列名的字符数
 器件码（4 字节）：芯片识别码
 系列名（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，RX63T 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，就发送应答（06h）。如果不是能支持的器件或者送来的命令为非法命令，RX63T 就发送错误应答（90h）。必须根据要编程的数据，从支持器件查询命令发送的 2 个器件信息中选择某个字节序指定的器件码。

命令	10h	长度	器件码	SUM
应答	06h			
错误应答	90h	错误		

- 长度（1 字节）：器件码的字符数（是固定值，为“4”。）
 器件码（4 字节）：芯片系列名的 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: 器件码错误

(3) 时钟模式查询

如果主机发送时钟模式查询命令，RX63T 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，RX63T 就只发送所选时钟模式的信息。

命令	21h	
应答	31h	长度
	模式	
	SUM	

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，RX63T 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，RX63T 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，RX63T 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		

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: 时钟模式错误

(5) 倍增比查询

如果主机发送倍增比查询命令，RX63T 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	...
	SUM			

- 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，RX63T 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	
	SUM	

- 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) 用户区信息查询

如果主机查询用户区信息，RX63T 就发送用户区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		
	SUM		

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）：用户区的区域数（连续区域为 1 个区域）

区域的起始地址（4 字节）：用户区的起始地址

区域的结束地址（4 字节）：用户区的结束地址

SUM（1 字节）：校验和

(8) 擦除块信息查询

如果主机查询擦除块信息，RX63T 就发送用户区和数据区的合计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		

长度（2 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）：用户区的擦除块数

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

(9) 编程长度查询

如果主机查询编程长度，RX63T 就发送编程长度的信息。

命令	27h			
应答	37h	长度	编程长度	SUM

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

(10) 新位速率选择

如果主机发送新位速率选择命令，RX63T 就检查能否将内部 SCI 设定为指定的新位速率。如果能设定为新位速率，RX63T 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，RX63T 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），RX63T 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。

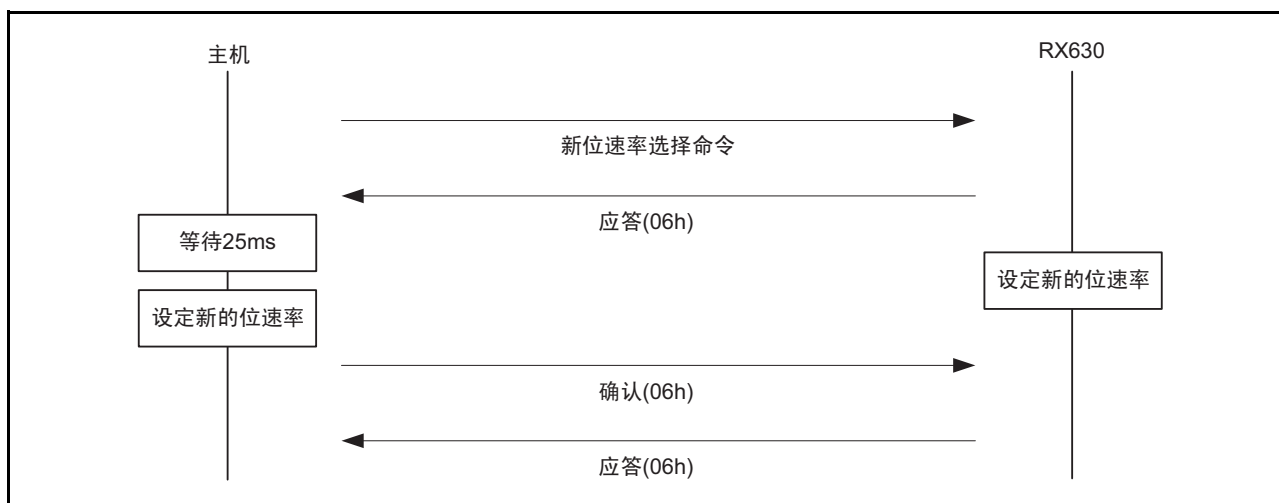


图 34.29 新位速率的选择顺序

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				

- 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）：RX63T 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（固定值：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（PCLKB）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RX63T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 f_{EX} 、倍增比2为 $M_{P\phi}$ 、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CKS[1:0]位的设定值为n，则用以下表达式求位速率的误差。

$$\text{误差 (\%)} = \left\{ \frac{f_{EX} \times M_{P\phi} \times 10^6}{B \times 64 \times 2^{2n-1} \times (n+1)} - 1 \right\} \times 100$$

- 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RX63T无法运行，就产生工作频率错误。RX63T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。

(11) 编程 / 擦除状态转移

如果主机发送编程 / 擦除状态转移命令，RX63T 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，全部擦除用户区 / 数据区。在全部擦除后，RX63T 发送应答（26h）并且转移到编程 / 擦除命令等待状态。如果发生错误而没有全部擦除，RX63T 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

ACK（1 字节）：ACK 码
26h: ID 码保护无效
16h: ID 码保护有效

(12) 引导程序状态查询

如果主机发送引导程序状态查询命令，RX63T 就发送当前的状态。能在查询 / 设定命令等待状态和编程 / 擦除主机命令等待状态下使用引导程序状态查询命令。

命令	4Fh			
应答	5Fh	长度	状态	错误

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）：RX63T 的状态（参照表 34.13）
错误（1 字节）：RX63T 的错误发生状况（参照表 34.14）

表 34.13 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 / 擦除命令等待状态的转移（位速率选择结束）。
31h	正在擦除用户区。
3Fh	等待编程 / 擦除命令。
4Fh	等待编程数据的接收。
5Fh	等待擦除块的指定。

表 34.14 错误的内容

代码	内容
00h	没有错误
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

34.9.6 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 34.15 所示。

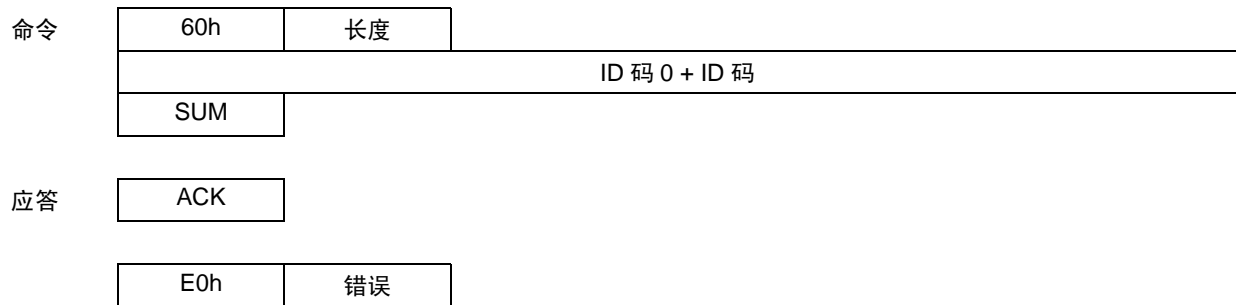
表 34.15 ID 码检查命令

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，RX63T 就发送命令错误应答。有关命令错误的内容，请参照“34.9.5 查询 / 设定命令的等待”。

(1) ID 码检查

如果主机发送 ID 码检查命令，RX63T 就将 ROM 中的控制码、ID 码和主机送来的代码进行比较，并且送回比较结果。



长度（1 字节）：ID 码的字节数（是固定值，为“16”。）

ID 码（16 字节）：ID 码 0（1 字节）+ ID 码（15 字节）

SUM（1 字节）：校验和

ACK（1 字节）：ACK 码

26h：对编程 / 擦除状态转移的应答

错误（1 字节）：错误码

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。

34.9.7 编程 / 擦除主机命令等待状态

在编程 / 擦除命令等待状态下能使用的命令一览表如表 34.16 所示。

表 34.16 编程 / 擦除命令

命令名	功能
用户 / 数据区编程选择	选择用户 / 数据区的编程
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
用户区校验和	用户区的校验和
用户区空白检查	用户区的空白检查
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
引导程序状态查询	查询 RX63T 的状态

如果主机发送未定义命令，RX63T 就发送命令错误应答。有关命令错误的内容，请参照“34.9.5 查询 / 设定命令的等待”

在进行 ROM 的编程时，主机在发送编程选择命令（选择用户区编程 / 数据区编程）后发送 256 字节编程命令。如果主机发送编程选择命令，RX63T 就进入编程数据等待状态（参照“34.9.2 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，RX63T 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，RX63T 就判断为编程结束，转移到编程 / 擦除命令等待状态。

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，RX63T 就进入擦除块指定等待状态（参照“34.9.2 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，RX63T 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，RX63T 就判断为擦除结束，转移到编程 / 擦除命令等待状态。

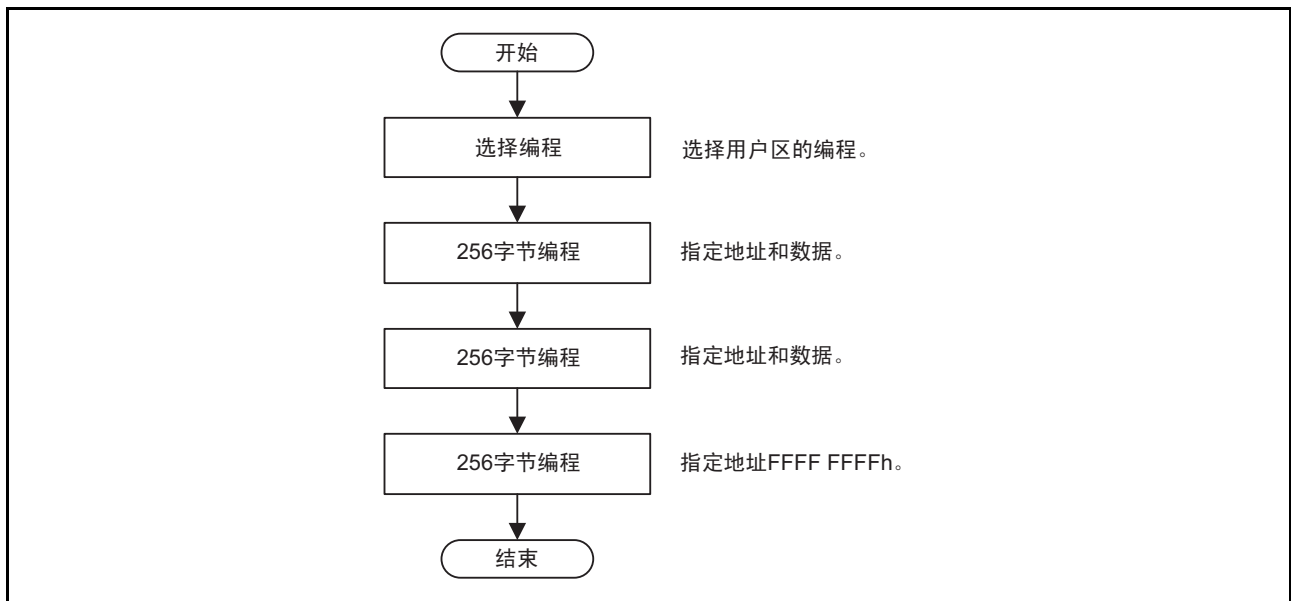


图 34.30 引导模式的 ROM 编程方法

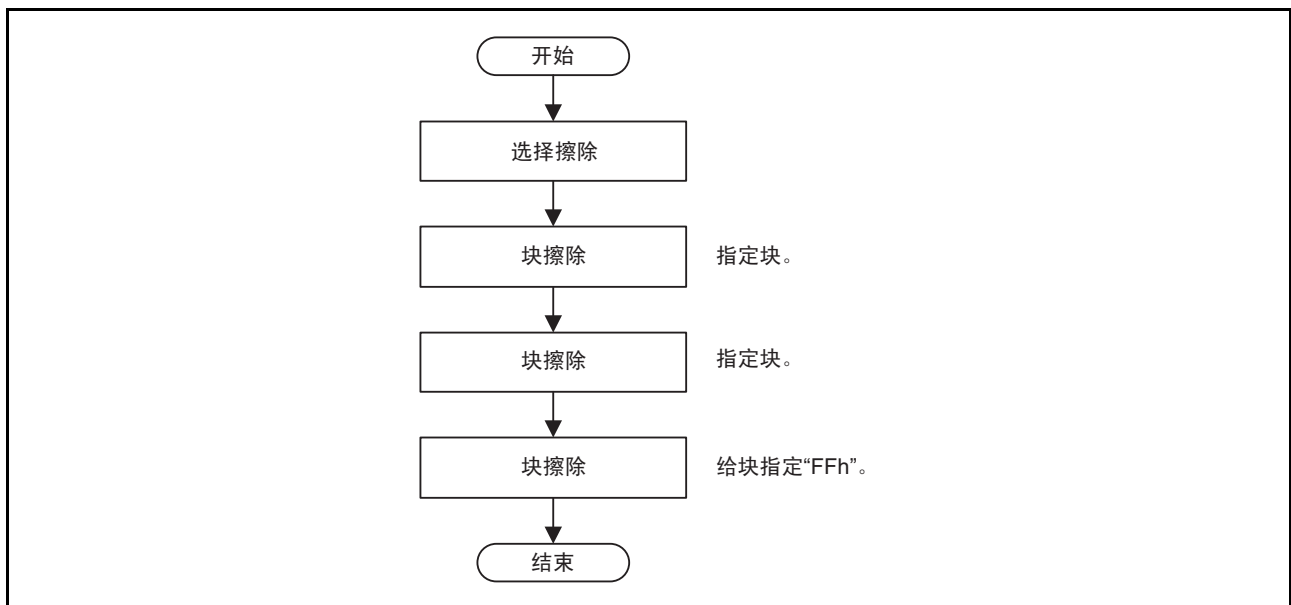


图 34.31 引导模式的 ROM 擦除方法

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 RX63T 的命令，“应答”是从 RX63T 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) 用户区 / 数据区编程选择

如果主机发送用户区 / 数据区编程选择命令，RX63T 就选择用户区编程程序，进入编程数据等待状态。

命令	43h
应答	06h

(2) 256 字节编程

如果主机发送 256 字节编程命令，RX63T 就进行 ROM 的编程。如果 ROM 的编程正常结束，RX63T 就发送应答（06h）。如果在编程处理过程中发生错误，RX63T 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		

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（地址不在指定的区内）
 53h: 发生编程错误，不能进行编程。

(3) 擦除选择

如果主机发送擦除选择命令，RX63T 就选择擦除程序，进入擦除块指定等待状态。

命令	48h
应答	06h

(4) 块擦除

如果主机发送块擦除命令，RX63T 就擦除 ROM。如果 ROM 擦除正常结束，RX63T 就发送应答（06h）。如果在擦除处理过程中发生错误，RX63T 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
------	-----	----

- 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。

(5) 存储器读

如果主机发送存储器读命令，RX63T 就读 ROM。如果读操作正常结束，RX63T 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，RX63T 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
------	-----	----

- 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）：读对象的区
01h: 用户区
- 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区内。
- 2Bh: 长度错误
 - 在读长度的选择中指定“00h”。
 - 读长度超过区的长度。
 - 由读起始地址和读长度计算出的地址不在区内。

(6) 用户区校验和

如果主机发送用户区校验和命令，RX63T 就发送以字节为单位将用户区的数据相加后的结果（校验和）。

命令	4Bh			
应答	5Bh	长度	区的校验和	SUM

长度（1 字节）：区的校验和的字节数（是固定值，为“4”。）

区的校验和（4 字节）：用户区的校验和结果
 用户区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）

(7) 用户区空白检查

如果主机发送用户区空白检查命令，RX63T 就检查用户区是否全部为擦除状态。如果用户区全部为擦除状态，RX63T 就发送应答（06h）。如果用户区有未擦除区域，RX63T 就发送错误应答（CDh、52h）。

命令	4Dh	
应答	06h	
错误应答	CDh	52h

(8) 读锁定位状态

如果主机发送读锁定位状态命令，RX63T 就读锁定位。如果读操作正常结束，RX63T 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，RX63T 就发送错误应答（F1h）。

命令	71h	长度	区域	A15 ~ A8	A23 ~ A16	A31~A24	SUM
----	-----	----	----	----------	-----------	---------	-----

A15 ~ A8（1 字节）：指定块的结束地址（15 ~ 8 位）

A23 ~ A16（1 字节）：指定块的结束地址（23 ~ 16 位）

A31 ~ A24（1 字节）：指定块的结束地址（31 ~ 24 位）

应答	状态
----	----

错误应答	F1h	错误
------	-----	----

长度（1 字节）：区域、A15 ~ A8、A23 ~ A16、A31 ~ A24 的数据总字节数（RX63T 为固定值“4”。）

区域（1 字节）：读对象的区
01h：用户区

A15 ~ A8（1 字节）：指定块的结束地址的 A15 ~ A8（15 ~ 8 位）

A23 ~ A16（1 字节）：指定块的结束地址的 A23 ~ A16（23 ~ 16 位）

A31 ~ A24（1 字节）：指定块的结束地址的 A31 ~ A24（31 ~ 24 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（地址不在指定的区内）

(9) 锁定位编程

如果主机发送锁定位编程命令，RX63T 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，RX63T 就发送应答（06h）。否则，RX63T 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					

- 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（RX63T 为固定值“4”。）
- 区域（1 字节）：锁定对象的区
01h: 用户区
- 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（地址不在指定的区内）
53h: 发生编程错误，不能设定为锁定状态。

(10) 锁定位有效

如果主机发送锁定位有效命令，RX63T 就将锁定位置为有效。

命令	7Ah
应答	06h

(11) 锁定位无效

如果主机发送锁定位无效命令，RX63T 就将锁定位置为无效。

命令	75h
应答	06h

(12) 引导程序状态查询

请参照“34.9.5 查询 / 设定命令的等待”。

34.10 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，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 到 ID 码 7 中设定“50h”、“72h”、“6Fh”、“74h”、“65h”、“63h”、“74h”时，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 34.27 相同。

表 34.17 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	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	保护有效	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。

34.11 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

如果在编程 / 擦除过程中发生 RES# 引脚复位，必须在电特性规定的工作电压范围内，经过至少 tRESWF（参照“36. 电特性”）的复位输入期间后解除复位。

如果在编程 / 擦除过程中通过 FRESETR.FRESET 位对 FCU 进行复位，就必须保持 tFCUR（参照“36. 电特性”）时间的复位状态。不能在对 FCU 进行复位的期间读编程 / 擦除的对象 ROM。

能使用编程 / 擦除中的看门狗定时器复位、独立看门狗定时器复位和软件复位。

(5) 编程 / 擦除过程中的非屏蔽中断的禁止

如果在编程 / 擦除过程中发生非屏蔽中断（NMI 引脚中断、振荡停止检测中断、WDT 下溢 / 刷新错误、IWDT 下溢 / 刷新错误、电压监视 1 中断、电压监视 2 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非屏蔽中断。（该注意事项仅适用于 ROM。）

(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 / 擦除的异常结束

在编程 / 擦除过程中，如果因超出工作电压范围的电压波动、复位、FRESETR.FRESET 位引起的 FCU 复位以及错误检测引起的命令锁定状态，以及因下述 (9) 的禁止事项，使编程 / 擦除无法正常结束，锁定位就可能变为“0”（保护状态）。此时，必须在将 FPROTR.FPROTCN 位置“1”的状态下发行块擦除命令，擦除锁定位。此后，必须重新执行没有正常结束的程序。

(8) 编程 / 擦除过程中的禁止事项

在编程 / 擦除过程中，对闪存外加高电压。为了防止损坏闪存，不能进行以下的操作：

- RX63T 的电源不在工作电压范围内。
- 更新 FWEPROR.FLWE[1:0] 位的值。
- 通过设定 SYSCR0.ROME 位来更改运行模式。
- 更改 SCKCR3 寄存器的时钟源选择位。
- 更改 FlashIF 时钟（FCLK）的分频比。
- 给 PCKAR 寄存器设定不同于 FCLK 的频率。
- 转移到全模块时钟停止模式、软件待机模式或者深度软件待机模式。

(9) 在引导模式中改写闪存时的注意事项

要在引导模式中改写闪存时，需要输入主时钟。在引导模式中，必须在 XTAL 引脚和 EXTAL 引脚之间连接“36.3.1 时钟时序”所示范围的谐振器。

(10) 引导模式中的 EXTAL 引脚的处理

在使用引导模式时，必须给 EXTAL 引脚输入外部时钟或者连接谐振器来提供时钟。

35. E2 数据闪存（保存数据的闪存）

RX63T 群内置用于保存最多 64K 字节代码的闪存（ROM）和用于保存 8K 字节数据的闪存（E2 数据闪存）。本章说明 E2 数据闪存，有关 ROM 请参照“34. ROM（保存代码的闪存）”。

35.1 概要

E2 数据闪存的规格如表 35.1 所示，ROM 和 E2 数据闪存外围的框图如图 35.1 所示。

表 35.1 E2 数据闪存的规格

项目	内容	
存储空间	数据区：8K 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 6 个 FCLK 周期进行读操作。	
编程 / 擦除方式	<ul style="list-style-type: none"> 内置对 ROM/E2 数据闪存进行改写的专用定序器（FCU）。 能通过给 FCU 发行命令，对 ROM/E2 数据闪存进行编程或者擦除。 	
BGO（后台操作）功能	<ul style="list-style-type: none"> 在对 E2 数据闪存进行编程或者擦除的期间，能执行分配在 ROM 区的程序。 在对 ROM 进行编程或者擦除的期间，CPU 能执行分配在 ROM/E2 数据闪存以外区域的程序。 	
挂起 / 恢复功能	<ul style="list-style-type: none"> 中止对 E2 数据闪存进行的编程或者擦除，CPU 能读 E2 数据闪存区（挂起）。 在中止后，能重新开始对 E2 数据闪存进行编程或者擦除（恢复）。 	
编程 / 擦除单位	<ul style="list-style-type: none"> 数据区的编程单位：2 字节 数据区的擦除单位：32 字节（256 块） 	
空白检查功能	<ul style="list-style-type: none"> 能执行用于确认 E2 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 2 字节或者 2K 字节。 	
板上编程（2 种）	通过引导模式改写 <ul style="list-style-type: none"> 使用异步串行接口（SCI1） 自动调整通信速度 通过用户程序中的 ROM 改写程序进行改写 <ul style="list-style-type: none"> 不需要对系统进行复位就能对 ROM 进行改写 	
保护功能	软件保护功能	<ul style="list-style-type: none"> 能通过 FENTRYR.FENTRYD 位、FWEPROR.FLWE[1:0] 位、DFLRE0 寄存器和 DFLWE0 寄存器防止意料不到的改写和读操作。 通过 DFLRE0 寄存器和 DFLWE0 寄存器进行的改写的单位为 2K 字节。
	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“36. 电特性”。	

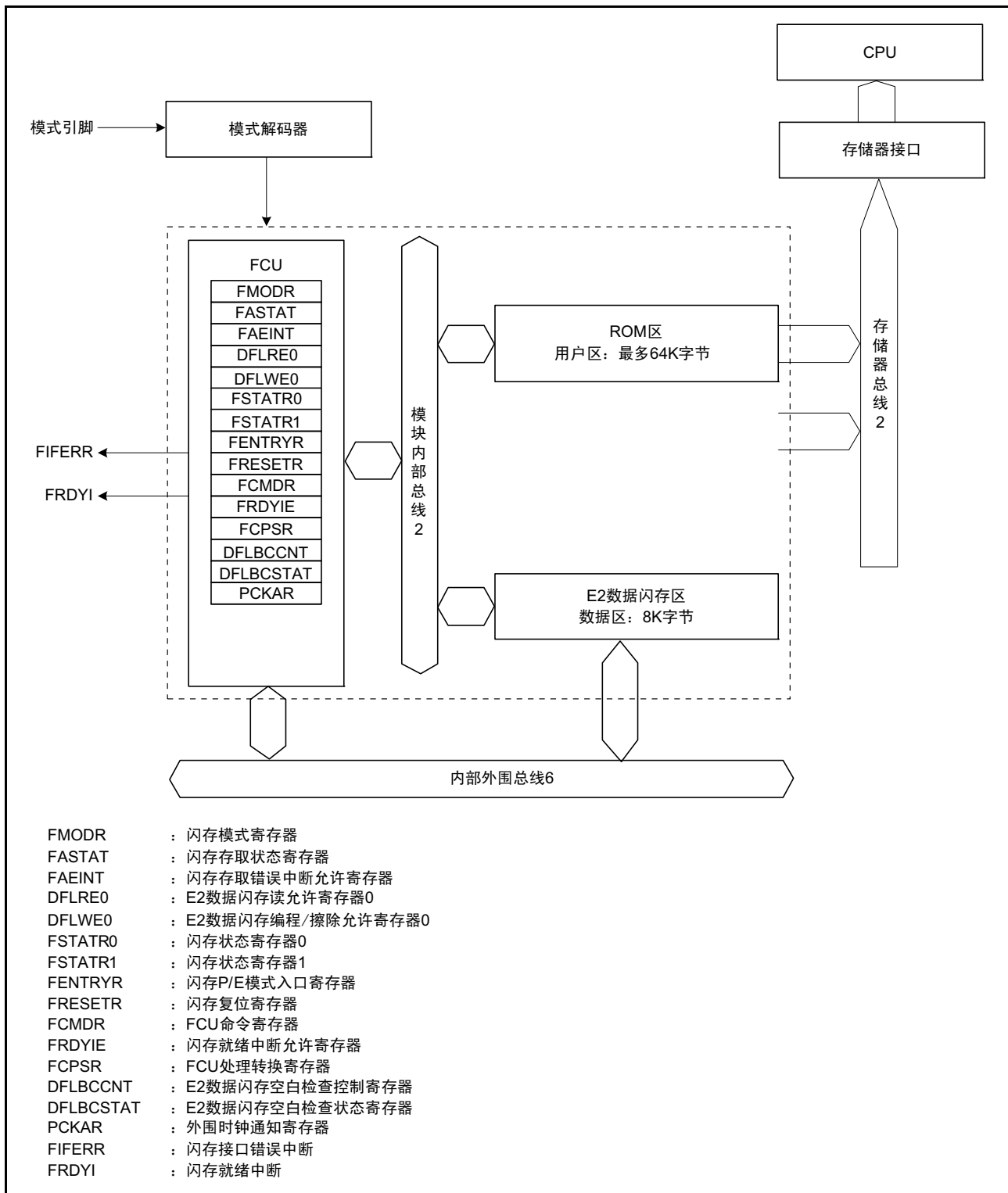


图 35.1 E2 数据闪存的框图

有关 E2 数据闪存相关的输入 / 输出引脚，请参照“表 34.9 内部 ROM 的相关输入 / 输出引脚”。

35.2 寄存器说明

部分寄存器也有 ROM 的相关位，本章只说明 E2 数据闪存相关位的功能。有关由 ROM/E2 数据闪存的公共位构成的寄存器（FRDYIE、FSTATR0、FSTATR1、FRESETR、FCMDR、FCPSR、PCKAR、FWEPROR）以及 ROM 专用位功能的详细内容，请参照“34. ROM（保存代码的闪存）”的“34.2 寄存器说明”。

P/E 表示编程 / 擦除。

35.2.1 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 在要转移到 E2 数据闪存锁定位读模式时进行设定。E2 数据闪存没有锁定位，如果在转移到锁定位读模式后读 E2 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

在使用空白检查命令时，必须将 FRDMD 位置“1”。

在内部 ROM 为无效时，FMODR 寄存器的读数据为“00h”，不能写。

FRDMD 位（FCU 读模式选择位）

此位选择是进行 E2 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“34. ROM（保存代码的闪存）”）。

35.2.2 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

位	符号	位名	功能	R/W
b0	DFLWPE	E2 数据闪存的编程 / 擦除保护违反位	0: 不发行违反 DFLWE0 寄存器设定的 E2 数据闪存的编程 / 擦除命令 1: 发行违反 DFLWE0 寄存器设定的 E2 数据闪存的编程 / 擦除命令	R/(W) (注1)
b1	DFLRPE	E2 数据闪存的读保护违反位	0: 不读违反 DFLRE0 寄存器设定的 E2 数据闪存 1: 读违反 DFLRE0 寄存器设定的 E2 数据闪存	R/(W) (注1)
b2	—	保留位	读写值都为“0”。	R/W
b3	DFLAE	E2 数据闪存的存取违反位	0: 不违反 E2 数据闪存的存取 1: 违反 E2 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“34. 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”。

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.DFLWPE、FASTAT.DFLRPE、FASTAT.DFLAE、FASTAT.ROMAE 寄存器的任意位置“1”，FASTAT0.ILGLERR 位就变为“1”，FCU 就进入命令锁定状态（参照“35.7.2 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。

DFLWPE 位（E2 数据闪存的编程 / 擦除保护违反位）

此位表示有没有违反由 DFLWE0 寄存器设定的编程 / 擦除保护。

如果 DFLWPE 位变为“1”，FSTATR0.ILGLERR 位就变为“1”并且 FCU 进入命令锁存状态。

有关 FSTATR0 寄存器，请参照“34.2.5 闪存状态寄存器 0（FSTATR0）”。

[为“1”的条件]

- 对通过 DFLWE0 寄存器设定为禁止编程或者擦除的 E2 数据闪存区发行编程 / 擦除命令。

[为“0”的条件]

- 在读“1”后写“0”时

DFLRPE 位（E2 数据闪存的读保护违反位）

此位表示有没有违反由 DFLRE0 寄存器设定的读保护。

如果 DFLRPE 位变为“1”，FSTATR0.ILGLERR 位就变为“1”并且 FCU 进入命令锁存状态。

有关 FSTATR0 寄存器，请参照“34.2.5 闪存状态寄存器 0（FSTATR0）”。

[为“1”的条件]

- 对通过 DFLRE0 寄存器设定为禁止读的 E2 数据闪存区发行读存取命令。

[为“0”的条件]

- 在读“1”后写“0”时

DFLAE 位（E2 数据闪存的存取违反位）

此位表示有没有违反 E2 数据闪存的存取。

如果 DFLAE 位变为“1”，FSTATR0.ILGLERR 位就变为“1”，并且 FCU 进入命令锁定状态。

有关 FSTATR0 寄存器，请参照“34.2.5 闪存状态寄存器 0（FSTATR0）”。

[为“1”的条件]

- 在 FENTRYR.FENTRYD 位为“1”时并且在 E2 数据闪存 P/E 正常模式中，对 E2 数据闪存区发行读存取命令。
- 在 FENTRYD 位为“0”的状态下，对 E2 数据闪存区发行写存取命令。
- 在 FENTRYR.FENTRY0 位为“1”的状态下，对 E2 数据闪存区发行存取命令。

[为“0”的条件]

- 在读“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“35.7.2 错误保护”）。

[为“1”的条件]

- 在 FCU 检测到错误并且转移到命令锁定状态后

[为“0”的条件]

- 在 FCU 发行状态寄存器清除命令后

35.2.3 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
	ROMAEI E	—	—	CMDLK IE	DFLAEIE	—	DFLRPE IE	DFLWPE IE
复位后的值	1	0	0	1	1	0	1	1

位	符号	位名	功能	R/W
b0	DFLWPEIE	E2 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	E2 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	E2 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“34. ROM（保存代码的闪存）”。	R/W

在内部 ROM 无效模式中，FAEINT 寄存器的读数据为“00h”，不能写。

DFLWPEIE 位（E2 数据闪存的编程 / 擦除保护违反中断允许位）

在发生 E2 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（E2 数据闪存的读保护违反中断允许位）

在发生 E2 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（E2 数据闪存的存取违反中断允许位）

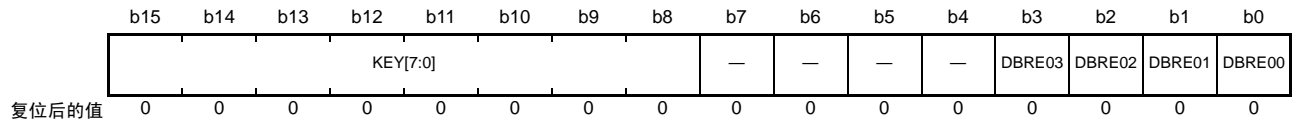
在发生 E2 数据闪存的存取违反并且 FASTAT.DFLAE 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的发生。

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。

35.2.4 E2 数据闪存读允许寄存器 0（DFLRE0）

地址 007F C440h



位	符号	位名	功能	R/W
b0	DBRE00	DB0000-DB0063（2K 字节）块的读允许位	0: 禁止读 1: 允许读	R/W
b1	DBRE01	DB0064-DB0127（2K 字节）块的读允许位		R/W
b2	DBRE02	DB0128-DB0191（2K 字节）块的读允许位		R/W
b3	DBRE03	DB0192-DB0256（2K 字节）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	控制是否改写 DFLRE0 寄存器。 如果改写 DFLRE0 寄存器，就必须以 16 位为单位，给高 8 位写“2Dh”、低 8 位写任意的值。	R/(W) (注 1)

注 1. 不保持写数据。

DFLRE0 寄存器允许或者禁止读数据区的 DB0000 ~ DB0255 块（参照图 35.3），以 2K 字节（64 个擦除块）为单位允许或者禁止读操作。

在内部 ROM 为无效时，DFLRE0 寄存器的读数据为“0000h”，不能写。

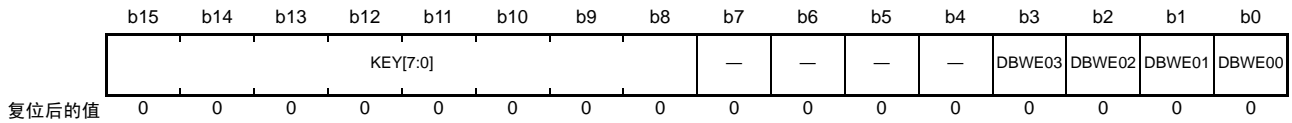
DBREj 位（DBj 块的读允许位）（j=00 ~ 03）

此位设定允许或者禁止读数据区的 DB0000 ~ DB0255 块。

DBREj 位用于控制 DBj 块的读操作。

35.2.5 E2 数据闪存编程 / 擦除允许寄存器 0（DFLWE0）

地址 007F C450h



位	符号	位名	功能	R/W
b0	DBWE00	DB0000-DB0063（2K 字节） 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	DBWE01	DB0064-DB0127（2K 字节） 块的编程 / 擦除允许位		R/W
b2	DBWE02	DB0128-DB0191（2K 字节） 块的编程 / 擦除允许位		R/W
b3	DBWE03	DB0192-DB0255（2K 字节） 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	控制是否改写 DFLWE0 寄存器。 如果改写 DFLWE0 寄存器，就必须以 16 位为单 位，给高 8 位写“1Eh”、低 8 位写任意的值。	R/(W) (注 1)

注 1. 不保持写数据。

DFLWE0 寄存器允许或者禁止对数据区的 DB0000 ~ DB0255 块（参照图 35.3）进行编程或者擦除，以 2K 字节（64 个擦除块）为单位允许或者禁止编程或者擦除。

在内部 ROM 为无效时，DFLWE0 寄存器的读数据为“0000h”，不能写。

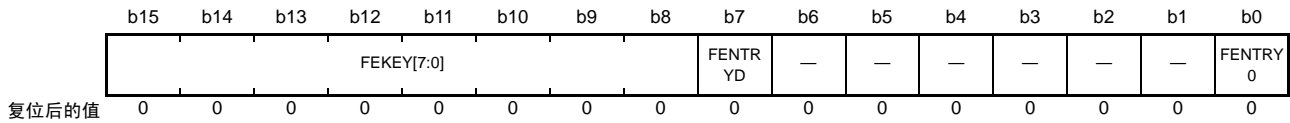
DBWEj 位（DBj 块的编程 / 擦除允许位）（j=00 ~ 03）

此位设定允许或者禁止对数据区的 DB0000 ~ DB0255 块进行编程或者擦除。

DBWEj 位用于控制 DBj 块的编程和擦除。

35.2.6 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“34. ROM（保存代码的闪存）”。	R/W
b6-b1	—	保留位	读写值都为“0”。	R/W
b7	FENTRYD	E2 数据闪存的 P/E 模式入口位	0: E2 数据闪存为读模式 1: E2 数据闪存为 P/E 模式	R/W
b15-b8	FEKEY[7:0]	键码	控制是否改写 FENTRYR 寄存器。 如果改写 FENTRYR 寄存器，就必须以 16 位为单 位，给高 8 位写“AAh”、低 8 位置任意的值。	R/(W) (注 1)

注 1. 不保持写数据。

为了将 ROM/E2 数据闪存设定为 P/E 模式，使 FCU 能接收命令，需要将 FENTRYD 位或者 FENTRY0 位置“1”。但是，如果给 FENTRYR 寄存器设定的值不是“AA01h”和“AA80h”，FSTATR0.ILGLERR 位就变为“1”，并且 FCU 进入命令锁定状态。

如果要在存取 FENTRYR 寄存器后转移到 ROM 读模式，就必须在写 FENTRYR 寄存器后读该寄存器，确认已变为设定值后进行 ROM 读运行。

在内部 ROM 为无效时，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 FSTATR0 寄存器，请参照“34.2.5 闪存状态寄存器 0（FSTATR0）”。

有关 FRESETR 寄存器，请参照“34.2.10 闪存复位寄存器（FRESETR）”。

FENTRYD 位（E2 数据闪存 P/E 模式入口位）

FENTRYD 位将 E2 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。
- FSTATR0.FRDI 位为“1”。
- 以字为单位给 FEKEY[7:0] 位写“AAh”。

[为“1”的条件]

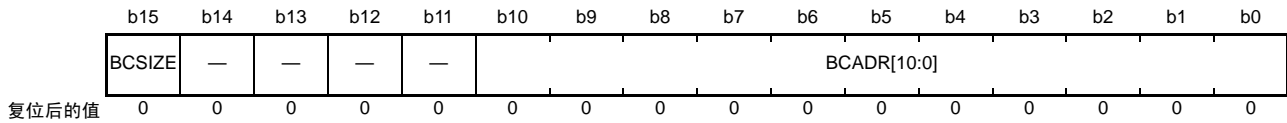
- 在满足写操作的有效条件并且 FENTRYR 寄存器为“0000h”的状态下，给 FENTRYD 位写“1”时

[为“0”的条件]

- 以字节为单位进行写操作时
- 在 FEKEY[7:0] 位不为“AAh”的状态下，以字为单位进行写操作时
- 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

35.2.7 E2 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
b10-b0	BCADR[10:0]	空白检查地址设定位	设定检查对象区的地址。	R/W
b14-b11	—	保留位	读写值都为“0”。	R/W
b15	BCSIZE	空白检查大小设定位	0: 空白检查对象区为 2 字节 1: 空白检查对象区为 2K 字节	R/W

在内部 ROM 为无效时，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“34.2.10 闪存复位寄存器（FRESETR）”。

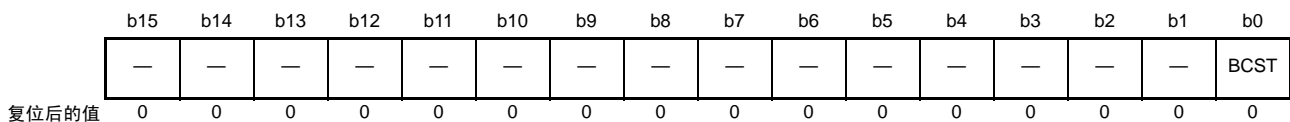
BCADR[10:0] 位（空白检查地址设定位）

在空白检查命令的检查对象区大小为 2 字节（BCSIZE 位为“0”）时，此位设定检查对象区的地址。BCADR[0] 位必须指定“0”。

当 BCSIZE 位为“0”时，检查对象区的起始地址是 DFLBCCNT 寄存器的设定值加上发行空白检查命令时指定的块起始地址（2K 字节单位）后的值。

35.2.8 E2 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

在内部 ROM 为无效时，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“34.2.10 闪存复位寄存器（FRESETR）”。

35.3 E2 数据闪存的存储区结构

RX63T 群的 E2 数据闪存由 8K 字节的数据区构成，E2 数据闪存的数据区结构如图 35.2 所示。在读、编程和擦除时，数据区的地址相同。

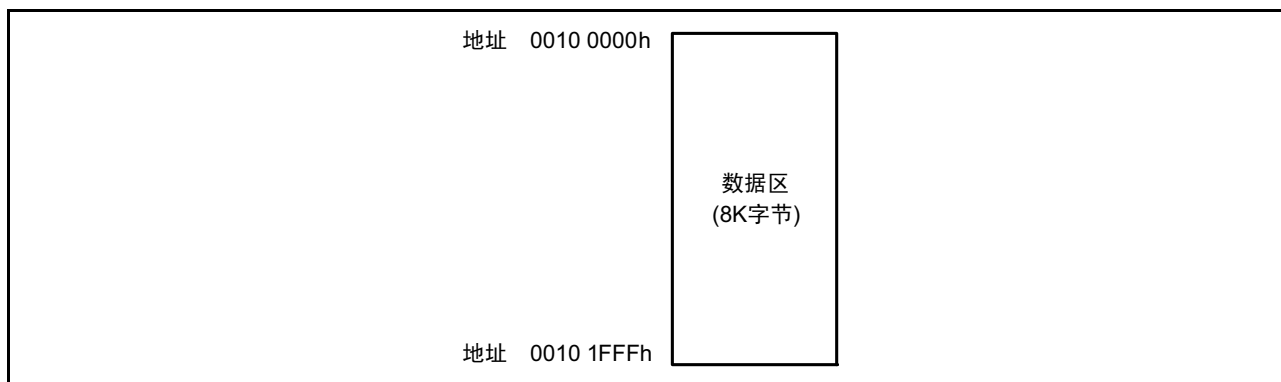


图 35.2 E2 数据闪存的数据区结构

35.4 块结构

数据区的擦除块结构如图 35.3 所示。数据区被分割为 32 字节（256 块），以块为单位进行擦除，以 2 字节为单位进行编程。以低位地址的 1 位为“0”的 2 字节为单位进行 2 字节编程

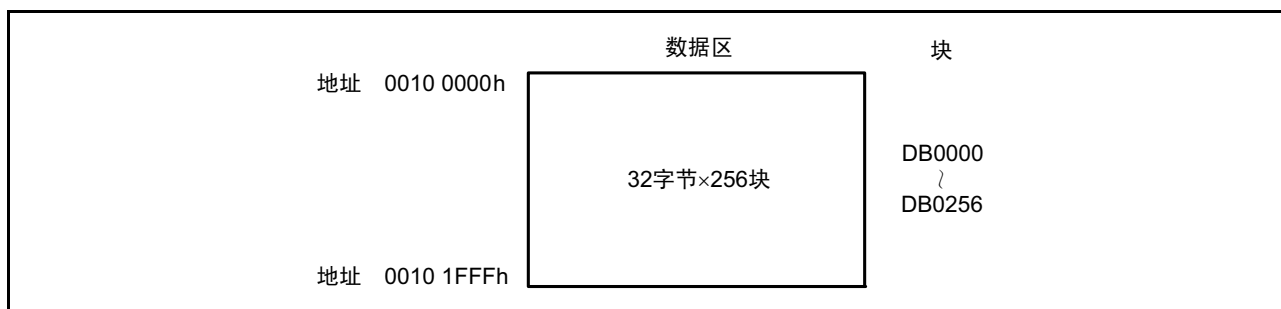


图 35.3 数据区的块分割

35.5 E2 数据闪存相关的运行模式

有关运行模式的转移，请参照“34.5 ROM 相关的运行模式”。

在引导模式 / 单芯片模式（内部 ROM 有效）中，能进行板上数据区的读、编程或者擦除。

各模式的不同点如表 35.2 所示。

表 35.2 各模式的不同点

项目	引导模式	单芯片模式（内部 ROM 有效）
编程 / 擦除环境	板上编程	
能编程和擦除的区	数据区	数据区
块分割擦除	○（注1）	○
复位时的启动程序	引导程序	用户程序

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“34.9.4 ID 码保护（引导模式）”、“34.9.2 引导模式的状态转移”。

- 在引导模式中，主机能经由SCI对数据区进行编程、擦除和读。
- 在引导模式中，通过引导程序使用内部RAM。因此，不保持内部RAM的数据。

35.6 E2 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 E2 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 E2 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和单芯片模式（内部 ROM 有效）中通用。

35.6.1 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 35.4 所示。

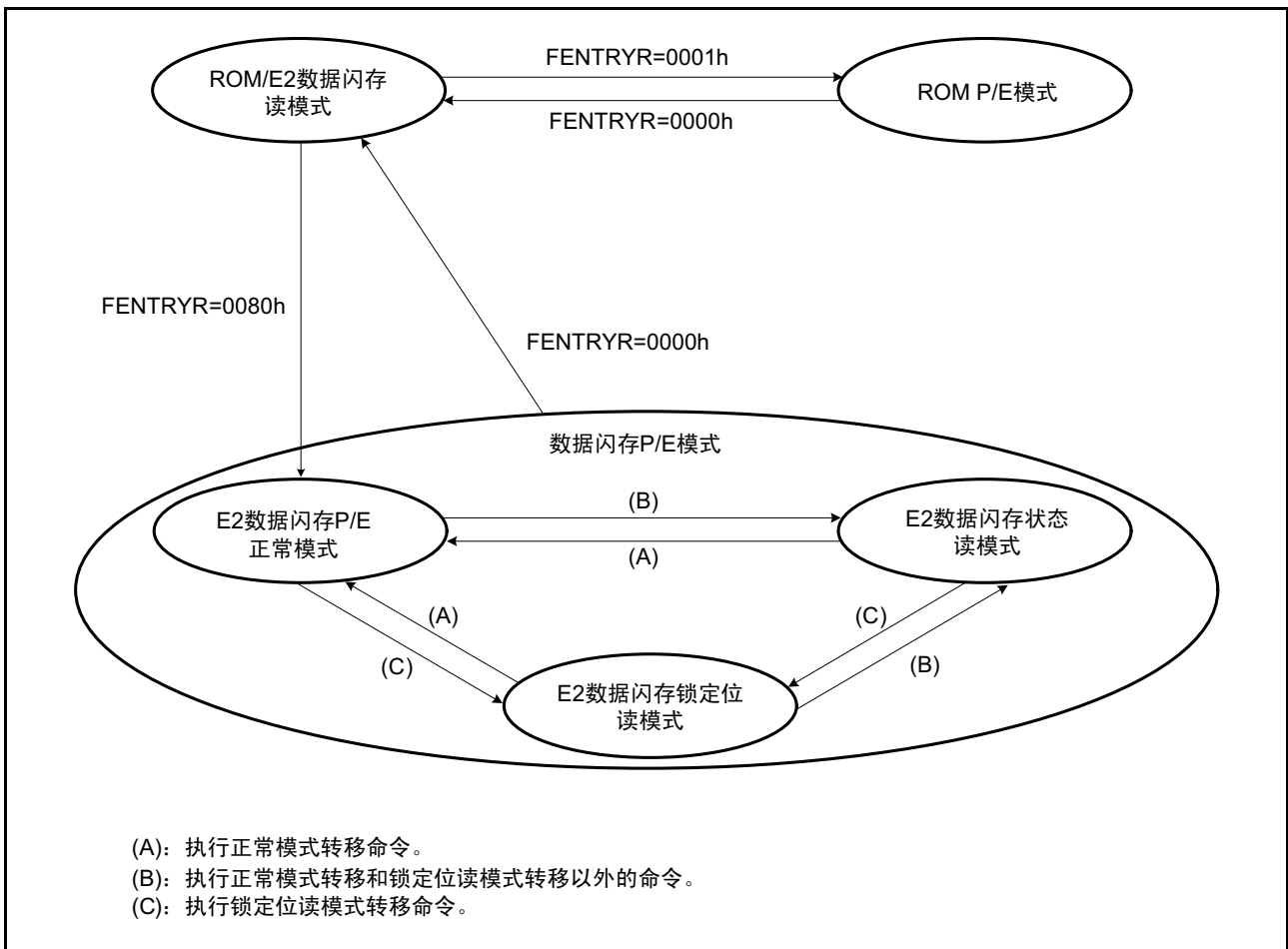


图 35.4 FCU 的模式转移图（E2 数据闪存相关）

35.6.1.1 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“34.6.1.2 ROM P/E 模式”。

35.6.1.2 ROM/E2 数据闪存读模式

ROM/E2数据闪存读模式是能读ROM和E2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.FENTRY0位置“0”，就转移到此模式。

35.6.1.3 E2 数据闪存 P/E 模式

E2 数据闪存 P/E 模式是对 E2 数据闪存进行编程和擦除的模式，不能读 E2 数据闪存。

E2数据闪存P/E模式有E2数据闪存P/E正常模式、E2数据闪存状态读模式、E2数据闪存锁定位读模式共3种。

(1) E2 数据闪存 P/E 正常模式

E2 数据闪存 P/E 正常模式是在对 E2 数据闪存进行编程或者擦除时最先转移的模式。如果在 ROM/E2 数据闪存读模式中将 FENTRYR.FENTRYD 位置“1”并且将 FENTRYR.FENTRY0 位置“0”，或者在 E2 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 35.7 所示。

(2) E2 数据闪存状态读模式

E2 数据闪存状态读模式是能读 E2 数据闪存状态的模式。

如果在 E2 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 E2 数据闪存状态读模式中的状态。能接受的命令如表 35.7 所示。

如果对 E2 数据闪存区发行读存取，就读 FSTATR0 寄存器的值。能高速读 ROM。

(3) E2 数据闪存锁定位读模式

E2 数据闪存锁定位读模式是能读 E2 数据闪存锁定位的模式。但是，因为 E2 数据闪存没有锁定位，所以不能读锁定位。

如果在 E2 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 E2 数据闪存锁定位读模式。能接受的命令如表 35.7 所示。

E2 数据闪存没有锁定位，如果对 E2 数据闪存区发行读存取，读数据就为不定值，但是不发生 E2 数据闪存的存取违反。能高速读 ROM。

35.6.2 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 E2 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 35.3 所示。

表 35.3 FCU 命令一览表（E2 数据闪存相关）

命令	功能
P/E 正常模式转移	转移到正常模式（参照“35.6.3 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“35.6.3 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“35.6.3 FCU 的模式和命令的关系”）。
外围时钟通知	设定 FlashIF 时钟（FCLK）的频率。
编程	对 E2 数据闪存进行编程（2 字节）。
块擦除	擦除 E2 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.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	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 E2 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 E2 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 35.4 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“34. ROM（保存代码的闪存）”的“34.6.2 FCU 命令一览表”。

如果在特定条件下进行如表 35.4 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“35.6.3 FCU 的模式和命令的关系”和“35.6.4 FCU 命令的使用方法”。

表 35.4 FCU 命令的格式（E2 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 N+3 周期	
		地址	数据	地址	数据	地址	数据	地址	数据
编程 (2 字节编程: N=1)	4	EA	E8h	EA	01h	WA	WDn	EA	D0h
空白检查	2	EA	71h	BA	D0h	—	—	—	—

地址列 EA: E2 数据闪存区的地址
0010 0000h ~ 0010 1FFFh 的任意地址

WA: 编程数据的调整后的起始地址

BA: E2 数据闪存的读 / 编程 / 擦除的允许 / 禁止块地址
读 / 编程 / 擦除的允许 / 禁止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)

35.6.3 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 35.5 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“35.7.2 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 FASTAT.CMDLK 位的值确认有没有发生错误，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 35.5 FCU 的模式 / 状态和能接受的命令的关系（E2 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	擦除挂起中的编程处理中	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 (FRDY=0)	命令锁定状态 (FRDY=1)	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	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	0	0
FSTATR0.ERSSPD 位	0	1	0	0	1	0/1	0	0	1	0/1	0/1	0	0	1	0
FSTATR0.PRGSPD 位	1	0	0	0	0	0/1	0	1	0	0/1	0/1	0	1	0	0
FASTAT.CMDLK 位	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
P/E 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

35.6.4 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 E2 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 E2 数据闪存的擦除状态的方法。有关其他 FCU 命令的使用方法，请参照“34. ROM（保存代码的闪存）”的“34.6.2 FCU 命令一览表”。

(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“34. ROM（保存代码的闪存）”的“34.6.4 FCU 命令的使用方法”。必须在将 FENTRYR.FENTRYD 位置“1”后设定 E2 数据闪存区内的地址。

(2) 编程方法

使用编程命令对 E2 数据闪存进行数据编程。

在编程命令的第 1 周期，以字节为单位给 E2 数据闪存区的地址写“E8h”，在第 2 周期，以字节为单位给 E2 数据闪存区的地址写编程字数（N）（注 1）。在命令的第 3～第 N+2 周期，必须以字为单位写编程数据。在第 3 周期的存取中，必须给编程对象区的起始地址写编程数据。必须将起始地址调整为以 2 字节为边界的地址。如果在进行 N 次以字单位给 E2 数据闪存区的地址写编程数据后，在第 N+3 周期以字节为单位给 E2 数据闪存区的地址写“D0h”，FCU 就开始对 E2 数据闪存进行编程处理。能通过 FSTATR0.FRDY 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”。另外，在将 DFLWE0 的编程 / 擦除保护置“1”后，必须允许包括写地址的块的编程 / 擦除。

E2 数据闪存的编程方法如图 35.5 所示。

注 1. 当进行 2 字节编程时，N 为“01h”。

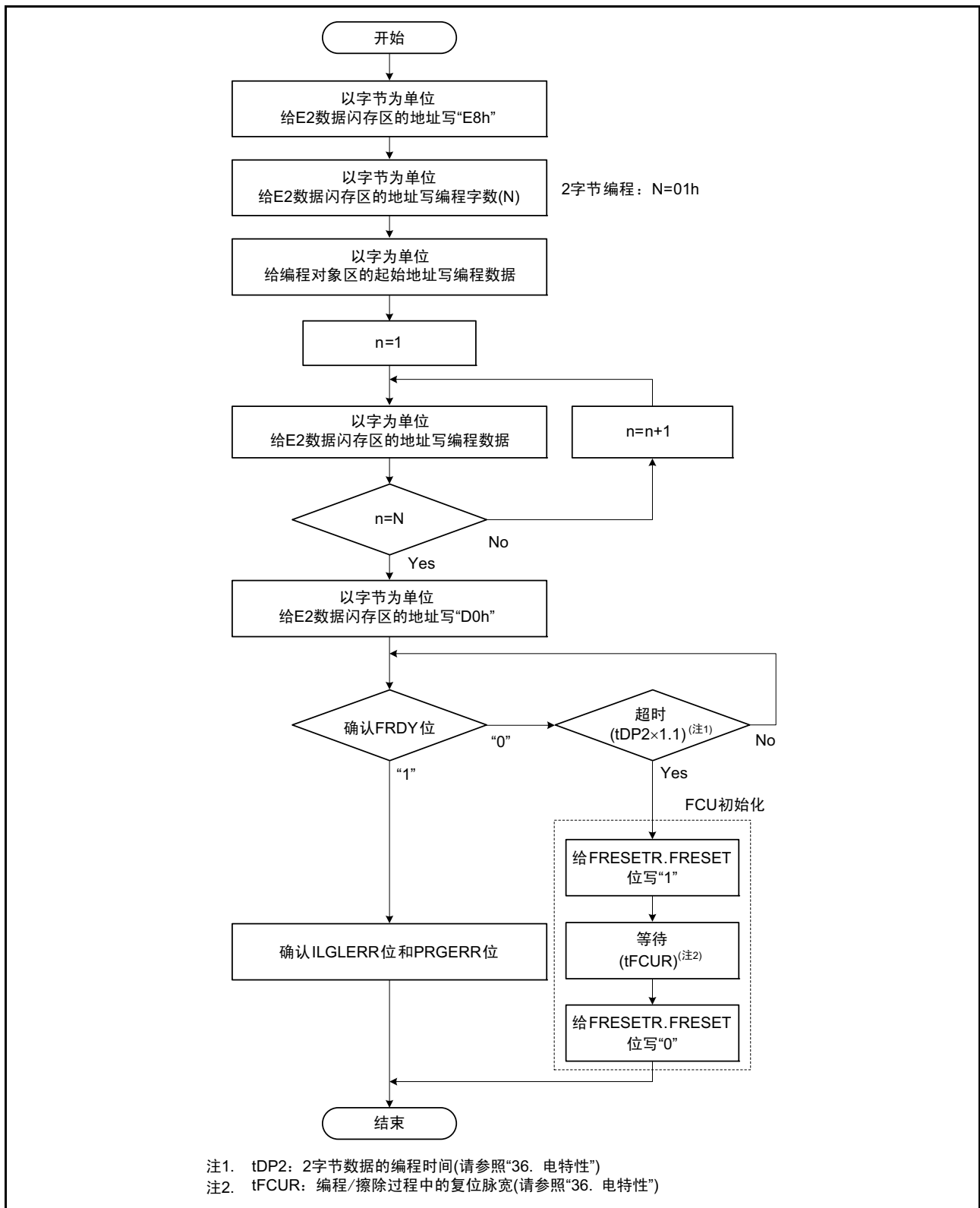


图 35.5 E2 数据闪存的编程

(3) 擦除方法

使用块擦除命令来擦除 E2 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“34. ROM（保存代码的闪存）”）。

必须注意：E2 数据闪存具有 DFLWE0 寄存器控制的编程 / 擦除保护功能。如果要在将通过 DFLWE0 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 E2 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（2K 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 2 字节区域进行空白检查。在空白检查命令的第 1 周期，以字节为单位给 E2 数据闪存区的地址写“71h”。在命令的第 2 周期，如果以字节为单位给包括空白检查对象区的擦除块内的任意地址写“D0h”，FCU 就开始进行 E2 数据闪存空白检查处理。能通过 FSTATR0.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。

E2 数据闪存空白检查方法如图 35.6 所示。

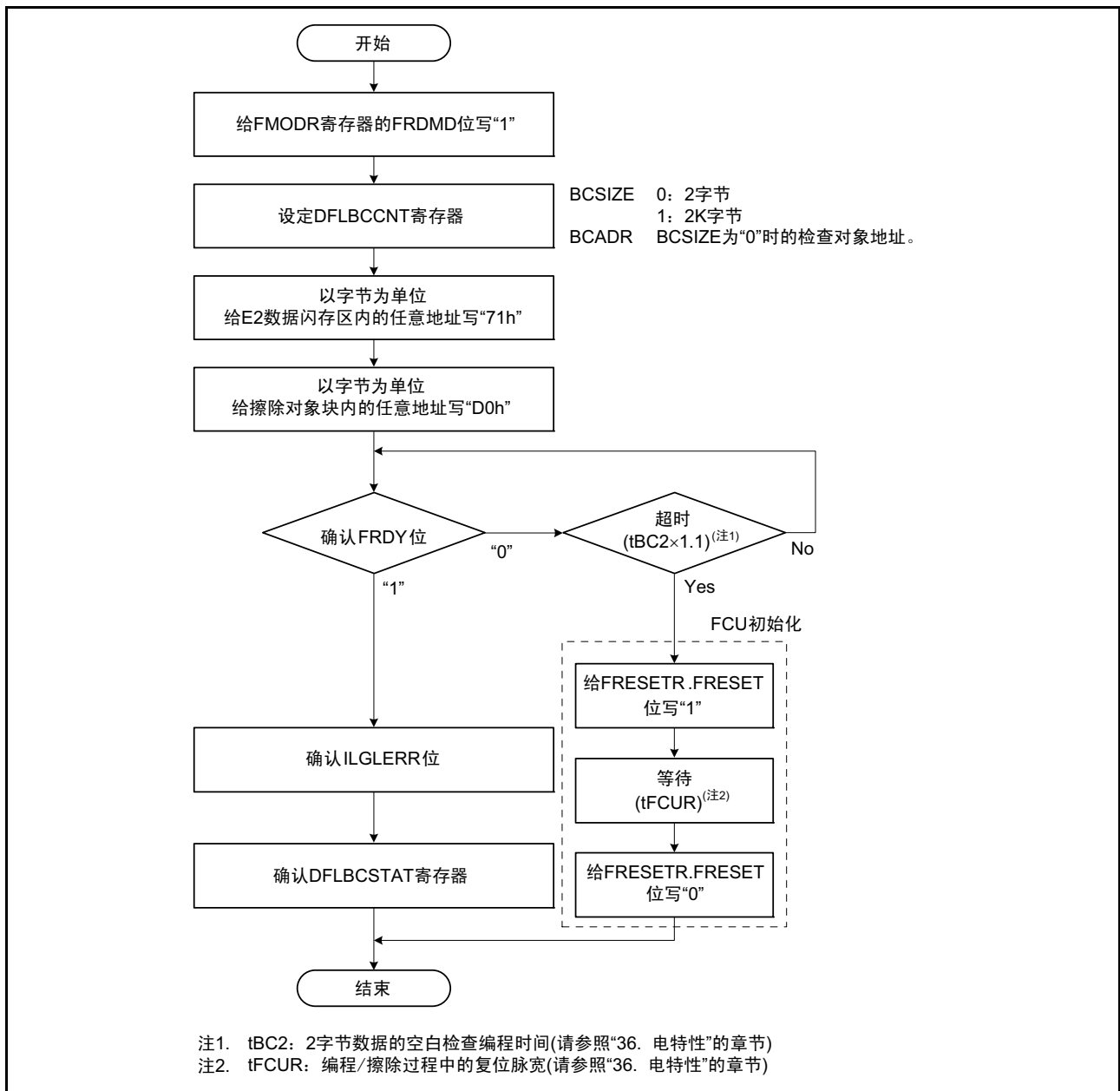


图 35.6 E2 数据闪存的空白检查

35.7 保护

E2 数据闪存的编程 / 擦除的保护有软件保护和错误保护。

35.7.1 软件保护

软件保护是通过设定控制寄存器禁止对 E2 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 E2 数据闪存发行编程 / 擦除命令或者读 E2 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/E2 数据闪存读模式。因为在 ROM/E2 数据闪存读模式中不接受 FCU 命令，所以 E2 数据闪存为禁止编程和擦除的状态。如果在 ROM/E2 数据闪存读模式中对 E2 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“35.7.2 错误保护”）。

(3) 通过 DFLWE0 寄存器进行的保护

当 DFLWE0.DBWE_j (j=00 ~ 03) 位为“0”时，数据区的 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“35.7.2 错误保护”）。

(4) 通过 DFLRE0 寄存器进行的保护

当 DFLRE0.DBRE_j (j=00 ~ 03) 位为“0”时，数据区的 DB_j 块为禁止读的状态。如果在 DBRE_j 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“35.7.2 错误保护”）。

35.7.2 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”），状态位（FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位、FSTATR1.FCUERR 位、FASTAT.DFLAE 位、FASTAT.DFLRPE 位、FASTAT.DFLWPE 位）中的任意位或者多个位就为“1”，禁止对 E2 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 E2 数据闪存相关的中断允许位（DFLAEIE 位、DFLRPEIE 位、DFLWPEIE 位）为“1”时，即使 FASTAT 寄存器的对应位（DFLAE 位、DFLRPE 位、DFLWPE 位）变为“1”，也产生 FIFERR 中断。

E2 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（FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位、FASTAT.DFLAE 位、FASTAT.DFLRPE 位、FASTAT.DFLWPE 位）的关系如表 35.6 所示。有关 ROM/E2 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、FCU 错误），请参照“34. ROM（保存代码的闪存）”的“34.8.2 命令锁定状态”。

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 35.6 错误保护一览表（E2 数据闪存专用）

分类	内容	ILGLERR	ERSERR	PRGERR	DFLAE	DFLRPE	DFLWPE
非法命令	在编程命令的第 2 周期指定“01h”以外的数据。	1	0	0	0	0	0
	在 FENTRYR.FENTRYD 位为“1”的状态下，对 E2 数据闪存区发行锁定位编程命令。	1	0	0	0	0	0
E2 数据闪存的存取违反	在 FENTRYD 位为“1”时并且在 E2 数据闪存 P/E 正常模式中，对 E2 数据闪存区发行读存取命令。	1	0	0	1	0	0
	在 FENTRYD 位为“0”的状态下，对 E2 数据闪存区发行写存取命令。	1	0	0	1	0	0
	在 FENTRYR.FENTRY0 位为“1”的状态下，对 E2 数据闪存区发行写存取命令。	1	0	0	1	0	0
E2 数据闪存的读保护违反	对通过 DFLRE0 寄存器设定为禁止读的 E2 数据闪存区发行读存取命令。	1	0	0	0	1	0
E2 数据闪存的写保护违反	对通过 DFLWE0 寄存器设定为禁止编程和擦除的 E2 数据闪存区发行编程 / 块擦除命令。	1	0	0	0	0	1

35.8 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，并且对数据区进行编程或者擦除。有关引导模式的系统结构和使用方法的详细内容，请参照“34.9 引导模式”。以下说明 E2 数据闪存固有的命令。

35.8.1 查询设定命令

E2 数据闪存固有的查询设定主机命令一览表如表 35.7 所示。在“34.9.5 查询 / 设定命令的等待”和“图 34.28 用户区 / 查询 / 设定命令的使用例子”所示的流程中，“查询 MAT 编程信息”使用数据区有无查询命令和数据区信息查询命令。

表 35.7 查询设定命令（E2 数据闪存固有）

主机命令名	功能
数据区有无查询	查询是否有数据区。
数据区信息查询	查询数据区的个数、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 RX63T 的命令，“应答”是从 RX63T 发送到主机的应答。“校验和”是指在合计 RX63T 发送的各字节时，为了得到“00h”而算出的字节数据。

(1) 数据区有无查询

如果主机发送数据区有无查询命令，RX63T 就发送表示有数据区的的信息。

命令	2Ah			
应答	3Ah	长度	有无区	SUM

长度（1 字节）：有无区的字符数（是固定值，为“1”。）

有无区（1 字节）：有无数据区（是固定值，为“21h”。）
21h: 有数据区

SUM（1 字节）：校验和

(2) 数据区信息查询

如果主机发送数据区信息查询命令，RX63T 就发送数据区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		
	SUM		

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）：数据区的区域数（连续区域为 1 个区域）

区域的起始地址（4 字节）：数据区的起始地址

区域的最后地址（4 字节）：数据区的结束地址

SUM（1 字节）：校验和

数据区的块结构信息包括在擦除块信息查询命令（参照“34.9.5 查询 / 设定命令的等待”）的应答中。

35.8.2 编程 / 擦除命令

E2 数据闪存固有的编程 / 擦除命令一览表如表 35.8 所示。E2 数据闪存固有的命令只有数据区校验和命令以及数据区空白检查命令，而编程命令、擦除命令、读命令和 ROM 通用。

在对数据区进行编程时，在发行用户区编程选择命令后，用 256 字节编程命令给编程地址指定数据区的地址。在擦除数据区时，在发行擦除选择命令后，用块擦除命令指定数据区的擦除块。数据区的擦除块信息包括在擦除块查询命令的应答中。在读数据区时，用存储器读命令选择用户区并且给读对象地址指定数据区的地址。

有关用户区编程选择命令、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“34.9.7 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“34.9.5 查询 / 设定命令的等待”。

表 35.8 编程 / 擦除主机命令（E2 数据闪存固有）

主机命令名	功能
数据区校验和	数据区的校验和
数据区空白检查	数据区的空白检查

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 RX63T 的命令，“应答”是从 RX63T 发送到主机的应答。“校验和”是指在合计 RX63T 发送的各字节时，为了得到“00h”而算出的字节数据。

(1) 数据区校验和

如果主机发送数据区校验和命令，RX63T 就发送以字节为单位将数据区的数据相加后的结果（校验和）。

命令

61h

应答

71h	长度	区的校验和	SUM
-----	----	-------	-----

长度（1 字节）：区的校验和的字节数（是固定值，为“4”。）

区校验和（4 字节）：数据区的校验和结果

SUM（1 字节）：校验和（应答数据的校验和）

(2) 数据区空白检查

如果主机发送数据区空白检查命令，RX63T 就检查数据区是否为全部擦除状态。如果数据区为全部擦除状态，RX63T 就发送应答（06h）。如果数据区有未擦除的区域，RX63T 就发送错误应答（E2h、52h）。

命令

62h

应答

06h

错误应答

E2h	52h
-----	-----

35.9 使用时的注意事项

(1) 启动复位后的数据区保护状态

因为 DFLRE0 寄存器和 DFLWE0 寄存器的初始值为“0000h”，所以启动复位后的数据区为禁止读、编程和擦除的状态。如果需要读数据区，就必须在设定 DFLRE0 寄存器后存取数据区。如果需要对数据区进行编程或者擦除，就必须在设定 DFLWE0 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“34. ROM（保存代码的闪存）”的“34.11 使用时的注意事项”。但是，在 E2 数据闪存的情况下，追加空白检查。必须将编程 / 擦除读作编程 / 擦除 / 空白检查。

36. 电特性

36.1 绝对最大额定值

表 36.1 绝对最大额定值

项目	符号	额定值	单位
电源电压	VCC	-0.3 ~ +4.6	V
输入电压 (5V 容许对应端口 (注1) 和端口 4 除外)	V _{in}	-0.3 ~ VCC+0.3	V
输入电压 (端口 4)	V _{in}	-0.3 ~ AVCC0+0.3	V
输入电压 (5V 容许对应端口 (注1))	V _{in}	-0.3 ~ +5.8	V
模拟电源电压	AVCC0 (注2)	-0.3 ~ +4.6	V
基准电源电压	VREFH0 (注2)	-0.3 ~ AVCC0+0.3	V
模拟输入电压 (端口 4)	V _{AN}	-0.3 ~ AVCC0+0.3	V
工作温度	T _{opr}	-40 ~ +85	°C
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，就可能造成 LSI 的永久性破坏。

注 1. 端口 0、端口 1、端口 2、端口 3、端口 7、端口 9、端口 A、端口 B 和端口 D 对应 5V 容许。

注 2. 在不使用 A/D 转换器时，不能将 AVCC0、VREFH0、VREFL0、AVSS0 引脚置为开路。AVCC0 引脚和 VREFH0 引脚必须连接 VCC，AVSS0 引脚和 VREFL0 引脚必须连接 VSS。

36.2 DC 特性

表 36.2 DC 特性 (1)

条件: $V_{CC}=2.7 \sim 3.6V$ 、 $V_{SS}=AV_{SS0}=V_{REFL0}=0V$ 、 $AV_{CC0}=3.0 \sim 3.6V$ 、 $V_{REFH0}=3.0V \sim AV_{CC0}$ 、 $T_a=T_{opr}$

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	V_{IH}	$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	V	
	MTU3 输入引脚	V_{IL}	-0.3	—	$V_{CC} \times 0.2$		
	POE3 输入引脚	ΔV_T	$V_{CC} \times 0.06$	—	—		
	SCI 输入引脚						
	A/D 触发输入引脚						
	GPT 输入引脚						
	RES#、NMI						
	RIIC 输入引脚 (IICBus 运行时)	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 4 (模拟兼用端口)	V_{IH}	$AV_{CC0} \times 0.8$	—	$AV_{CC0} + 0.3$		
		V_{IL}	-0.3	—	$AV_{CC0} \times 0.2$		
5V 容许对应端口 (注 1)	V_{IH}	$V_{CC} \times 0.8$	—	5.8			
	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}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$		
	EXTAL、TCK、RSPI 输入引脚		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$		
	RIIC 输入引脚 (SMBus 运行时)		2.1	—	$V_{CC} + 0.3$		
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	—	$V_{CC} \times 0.1$		
	EXTAL、TCK、RSPI 输入引脚		-0.3	—	$V_{CC} \times 0.2$		
	RIIC 输入引脚 (SMBus 运行时)		-0.3	—	0.8		
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	$I_{OL} = 1.0mA$
			—	—	0.4		$I_{OL} = 3mA$
	RIIC 引脚		—	—	0.6		$I_{OL} = 6mA$
输入泄露电流	RES#、MD 引脚、EMLE、 端口 4、端口 PE2	$ I_{in} $	—	—	1.0	μA	$V_{in} = 0V$ 、 $V_{in} = V_{CC}$
三态泄露电流 (OFF 状态)	5V 容许对应	$ I_{TS} $	—	—	5.0	μA	$V_{in} = 0V$ 、 $V_{in} = 5.5V$
输入电容	全部输入引脚 (端口 PB1、PB2 除外)	C_{in}	—	—	15	pF	$V_{in} = 0V$ 、 $f = 1MHz$ $T_a = 25^\circ C$
	端口 PB1、PB2		—	—	30		

注 1. 端口 0、端口 1、端口 2、端口 3、端口 7、端口 9、端口 A、端口 B、端口 D 对应 5V 容许。

表 36.3 DC 特性 (2)

条件: VCC=2.7 ~ 3.6V、VSS=AVSS0=VREFL0=0V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、T_a=T_{opr}

项目		符号	min	typ	max	单位	测量条件
消耗电流 (注1)	运行时	最大运行时 (注2)	—	—	60	mA	ICLK=100MHz PCLKA=100MHz PCLKB=50MHz PCLKD=50MHz FCLK=50MHz
		正常运行时 (注4)	—	25	—		
		BGO 运行时的增加 (注5)	—	15	—		
	睡眠时		—	25	35		
	全模块时钟停止时 (注4)		—	14	25	mA	
	待机时	软件待机时	—	0.2	6		
		深度软件待机时	—	16	40		
模拟 电源电流	12 位 A/D 转换中 (使用采样 & 保持电路时)		AI _{CC0}	—	3	4	mA
	12 位 A/D 转换中 (未使用采样 & 保持电路时)			—	2	3	
	窗口比较器 (1ch 运行时)			—	0.4	1	
	窗口比较器 (3ch 运行时)			—	0.5	1	
	12 位 A/D 转换待机时			—	25	32	
基准 电源电流	12 位 A/D 转换中		AI _{REFH0}	—	0.6	0.7	mA
	12 位 A/D 转换待机时			—	0.6	0.7	
VCC 上升斜率			SrVCC	8.4	—	20000	ms/V

注 1. 消耗电流值是指全部的输出引脚为无负载状态时的值。

注 2. 外围功能为时钟提供状态, BGO 运行除外。

注 3. 根据以下表达式, I_{CC} 取决于 f(ICLK)。(ICLK:PCLK=8:4)

$$I_{CC \max} = 0.87 \times f + 13 \quad (\text{最大运行时})$$

$$I_{CC \text{ typ}} = 0.35 \times f + 5 \quad (\text{正常运行时})$$

$$I_{CC \max} = 0.48 \times f + 12 \quad (\text{睡眠时})$$

注 4. 外围功能为时钟提供停止状态, BGO 运行除外。

注 5. 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。

表 36.4 输出容许电流

条件: VCC=2.7 ~ 3.6V、VSS=AVSS0=VREFL0=0V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、T_a=T_{opr}

项目	符号	min	typ	max	单位
输出 Low 电平允许电流 (每个引脚的平均值)	I _{OL}	—	—	2.0 (注1)	mA
输出 Low 电平允许电流 (每个引脚的最大值)	I _{OL}	—	—	4.0 (注1)	mA
输出 Low 电平允许电流 (总和)	ΣI _{OL}	—	—	32	mA
输出 High 电平允许电流 (每个引脚的平均值)	-I _{OH}	—	—	2.0	mA
输出 High 电平允许电流 (每个引脚的最大值)	-I _{OH}	—	—	4.0	mA
输出 High 电平允许电流 (总和)	Σ-I _{OH}	—	—	32	mA

注 1. RIIC 引脚为 I_{OL}=6mA(Max.)。

【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 36.4 中的值。

36.3 AC 特性

表 36.5 工作频率

条件: VCC=2.7 ~ 3.6V、VSS=AVSS0=VREFL0=0V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、 $T_a=T_{opr}$

项目		符号	min	typ	max	单位
工作频率	系统时钟 (ICLK)	f	—	—	100	MHz
	外围模块时钟 (PCLK)		—	—	50	
	定时器模块时钟 (PCLKA)		—	—	100	
	S12AD 时钟 (PCLKD)		—	—	50	
	FlashIF 时钟 (FCLK)		— (注1)	—	50	

注 1. 在改写 ROM 和 E2 数据闪存时, FCLK 必须大于等于 4MHz。

36.3.1 时钟时序

表 36.6 时钟时序

条件: VCC=2.7 ~ 3.6V、VSS=AVSS0=VREFL0=0V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、 $T_a=T_{opr}$

项目	符号	min	typ	max	单位	测量条件	
EXTAL 外部时钟输入的周期时间	t_{EXcyc}	50	—	250	ns	图 36.1	
EXTAL 外部时钟输入的 Low 电平脉宽	t_{EXL}	20	—	—	ns		
EXTAL 外部时钟输入的 High 电平脉宽	t_{EXH}	20	—	—	ns		
EXTAL 外部时钟上升时间	t_{EXr}	—	—	5	ns		
EXTAL 外部时钟下降时间	t_{EXf}	—	—	5	ns		
EXTAL 外部时钟输入的待机时间 (注1)	t_{EXWT}	1	—	—	ms		
主时钟振荡器振荡频率	f_{MAIN}	4	—	16	MHz		
主时钟振荡器稳定时间 (晶体)	$t_{MAINOSC}$	10	—	—	ms	图 36.2	
主时钟振荡器稳定待机时间 (晶体)	$t_{MAINOSCWT}$	20	—	—	ms		
LOCO 时钟周期时间	t_{cyc}	6.96	8	9.4	μ s		
低速内部振荡器振荡频率	f_{LOCO}	106.25	125	143.75	kHz		
LOCO 时钟振荡器稳定时间	t_{LOCOWT}	—	—	20	μ s	图 36.3	
PLL 时钟振荡器稳定时间	在主时钟振荡器稳定后开始 PLL 运行。	t_{PLL1}	—	—	500	μ s	图 36.4
PLL 时钟振荡器稳定待机时间		t_{PLLWT1}	1.5	—	—	ms	
PLL 时钟振荡器稳定时间	在主时钟振荡器稳定前开始 PLL 运行。	t_{PLL2}	10	—	—	ms	图 36.5
PLL 时钟振荡器稳定待机时间		t_{PLLWT2}	20	—	—	ms	

注 1. 这是在将主时钟振荡器停止位 (MOSCCR.MOSTP) 置“0” (运行) 后到能使用为止的时间。

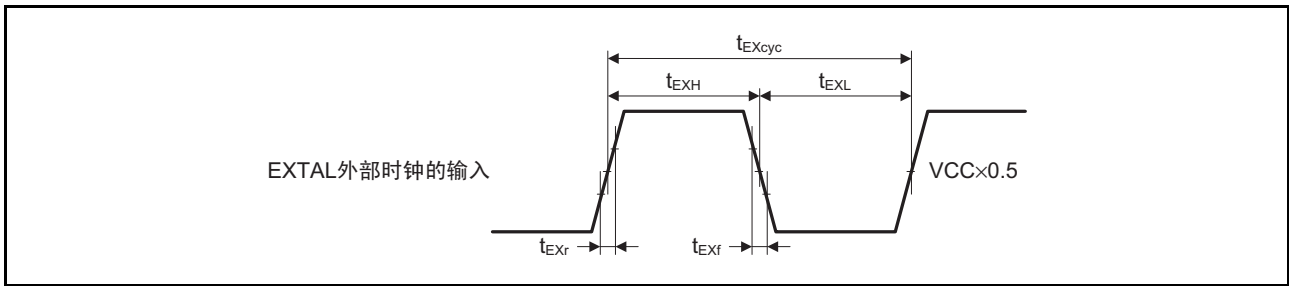


图 36.1 EXTAL 外部时钟输入的时序

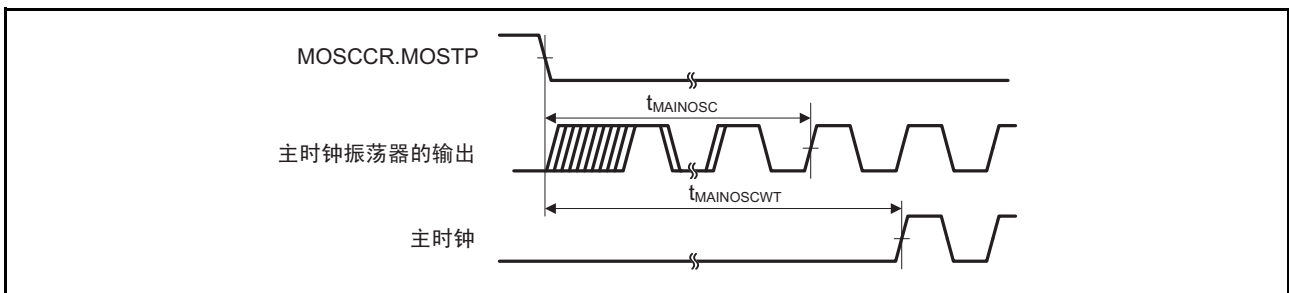


图 36.2 主时钟振荡开始的时序

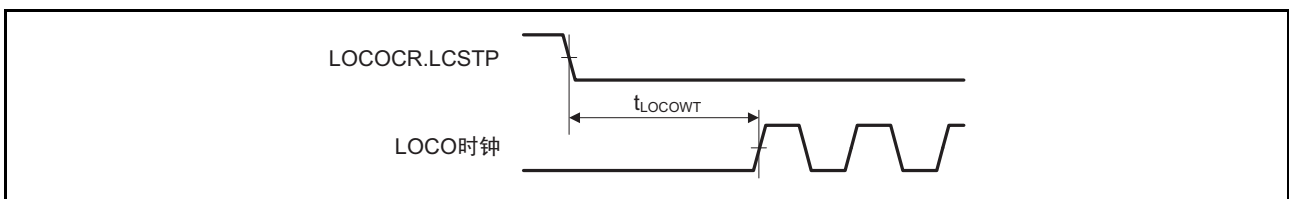


图 36.3 LOCO 时钟振荡开始的时序

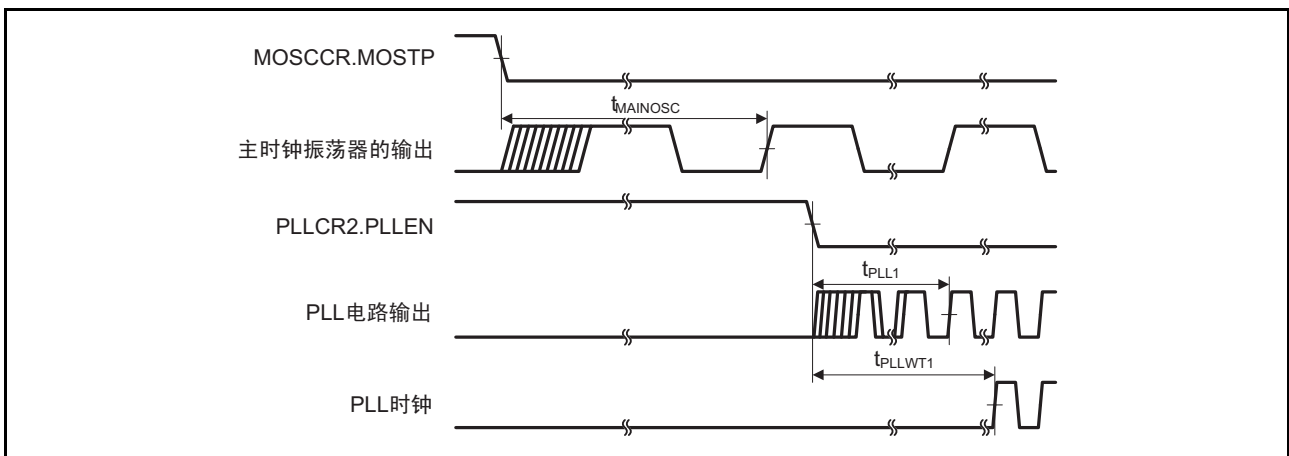


图 36.4 PLL 时钟振荡开始的时序（在主时钟振荡稳定后开始 PLL 运行的情况）

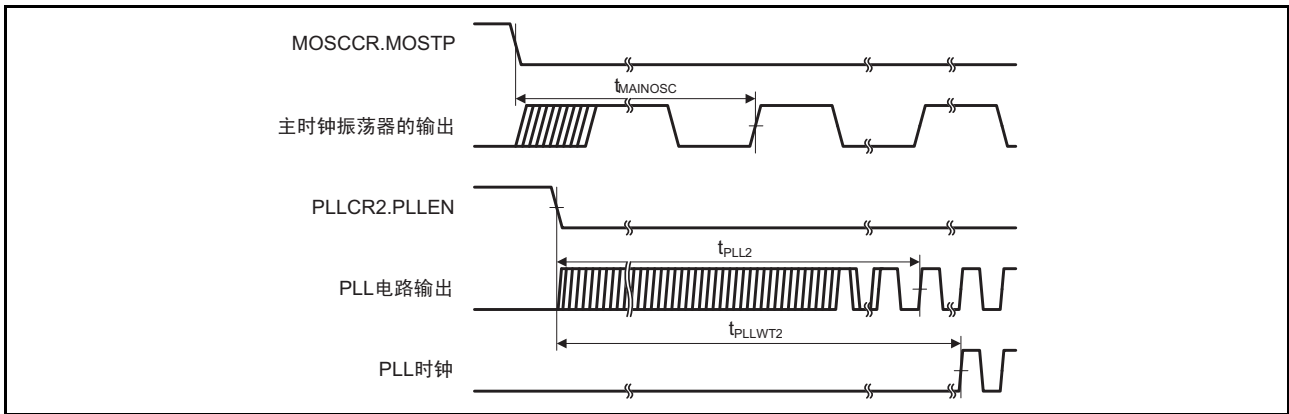


图 36.5 PLL 时钟振荡开始的时序（不等待主时钟振荡稳定就开始 PLL 运行的情况）

36.3.2 复位时序

表 36.7 复位时序

条件: VCC=2.7 ~ 3.6V、VSS=AVSS0=VREFL0=0V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、 $T_a=T_{opr}$

项目		符号	min	typ	max	单位	测量条件
RES# 脉宽	接通电源时	t_{RESWP}	2	—	—	ms	图 36.6
	深度软件待机模式	t_{RESWD}	1	—	—	ms	图 36.7
	软件待机模式、	t_{RESWS}	1	—	—	ms	
	上述以外（ROM 编程 / 擦除过程中、E2 数据闪存编程 / 擦除过程中、E2 数据闪存空白检查过程中除外）	t_{RESWF}	200	—	—	μ s	
RES# 解除后的待机时间	t_{RESWT}	59	—	60	t_{cyc}		
内部复位时间 (独立看门狗定时器复位、看门狗定时器复位、软件复位)	t_{RESW2}	112	—	120	t_{cyc}		

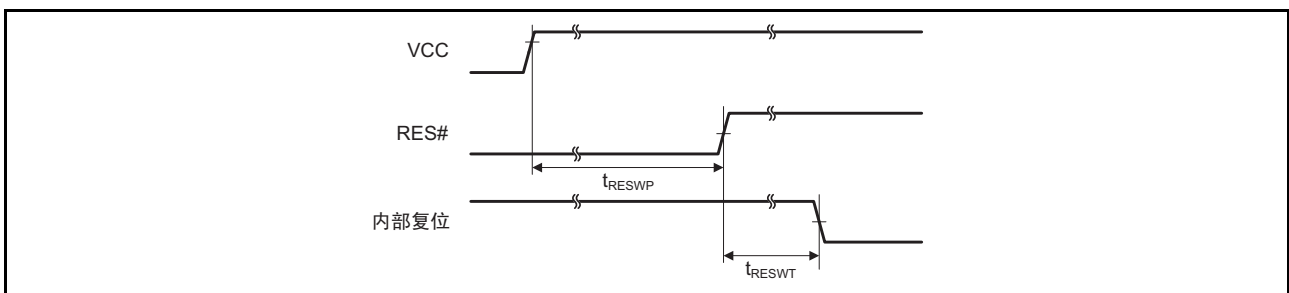


图 36.6 接通电源时的复位输入时序

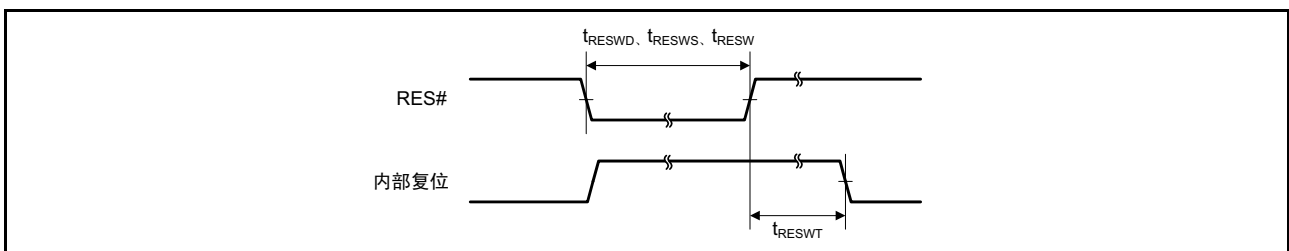


图 36.7 复位输入时序

36.3.3 从低功耗状态的恢复时序

表 36.8 从低功耗状态的恢复时序

条件: VCC=2.7 ~ 3.6V、VSS=AVSS0=VREFL0=0V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、 $T_a=T_{opr}$

项目		符号	min	typ	max	单位	测量条件
解除软件待机模式后的恢复时间	给主时钟振荡器 连接晶体谐振器	主时钟振荡器运行	t_{SBYMC}	10	—	—	ms
		主时钟振荡器、 PLL 电路运行	t_{SBYPC}	10	—	—	ms
	给主时钟振荡器 输入外部时钟	主时钟振荡器运行	t_{SBYEX}	1	—	—	ms
		主时钟振荡器、 PLL 电路运行	t_{SBYPE}	1	—	—	ms
	低速内部振荡器或者 IWDG 专用低速时钟 振荡器	t_{SBYLO}	—	—	800	μ s	
解除深度软件待机模式后的恢复时间		t_{DSBY}	—	—	1	ms	图 36.9
解除深度软件待机模式后的待机时间		t_{DSBYWT}	45	—	46	t_{cyc}	

注. 等待时间因执行 WAIT 指令时的各振荡器的状态而不同。多个振荡器振荡时的恢复时间和正在振荡的振荡器中恢复最费时间的振荡器单独振荡时的恢复时间相同。

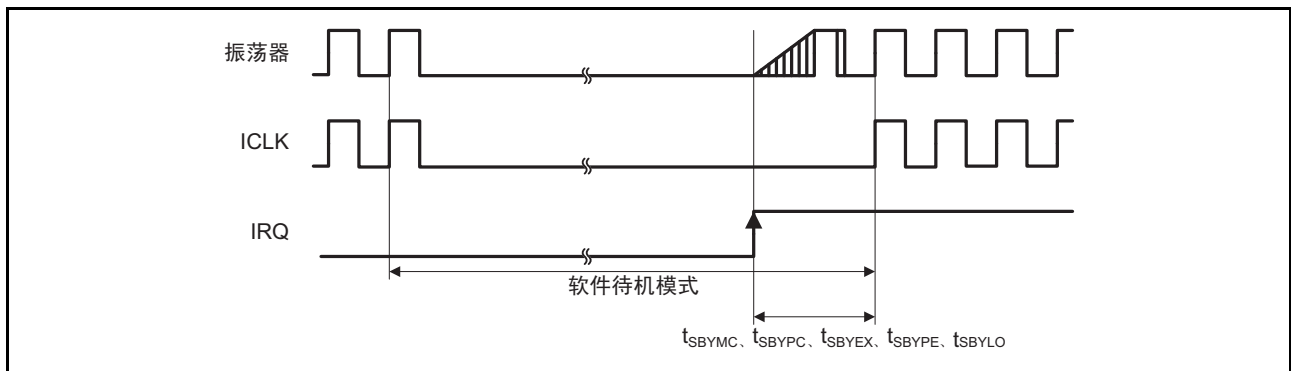


图 36.8 软件待机模式的解除时序

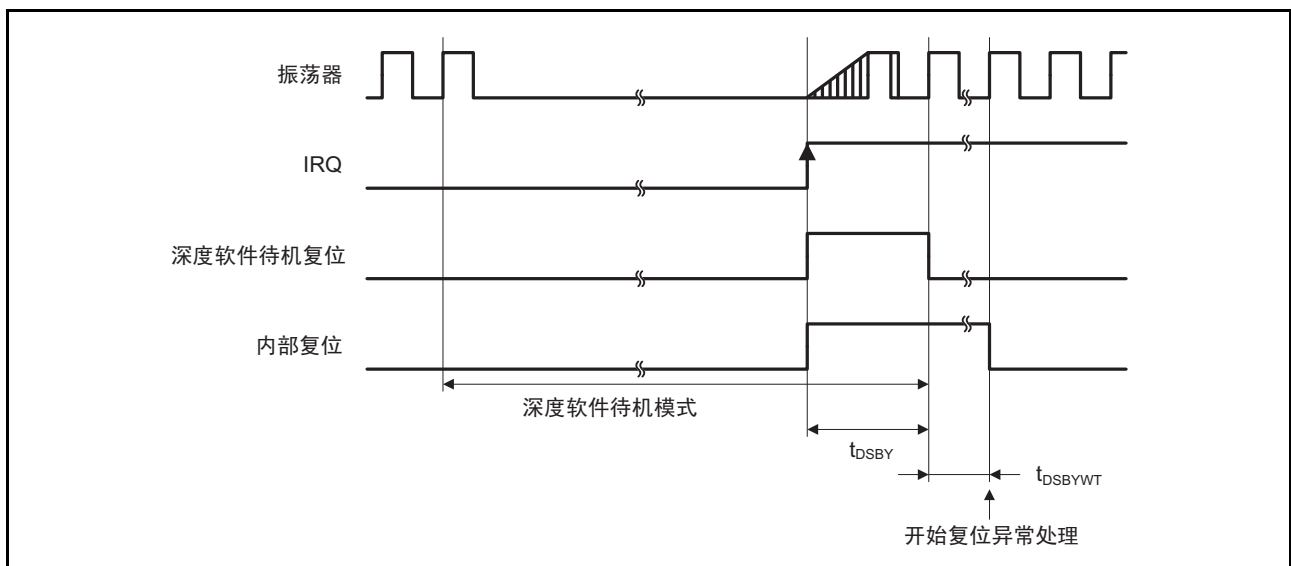


图 36.9 深度软件待机模式的解除时序

36.3.4 控制信号时序

表 36.9 控制信号时序

条件: $V_{CC}=2.7 \sim 3.6V$ 、 $V_{SS}=AVSS0=VREFL0=0V$ 、 $AVCC0=3.0 \sim 3.6V$ 、 $VREFH0=3.0V \sim AVCC0$ 、 $T_a=T_{opr}$

项目	符号	min	typ	max	单位 (注 1)	测量条件
NMI 脉宽	t_{NMIW}	200	—	—	ns	$t_{P_{cyc}} \times 2 \leq 200ns$ 、图 36.10
		2			$t_{P_{cyc}}$	$t_{P_{cyc}} \times 2 > 200ns$ 、图 36.10
IRQ# 脉宽	t_{IRQW}	200	—	—	ns	$t_{P_{cyc}} \times 2 \leq 200ns$ 、图 36.11
		2			$t_{P_{cyc}}$	$t_{P_{cyc}} \times 2 > 200ns$ 、图 36.11

注 1. $t_{P_{cyc}}$: PCLK 的周期

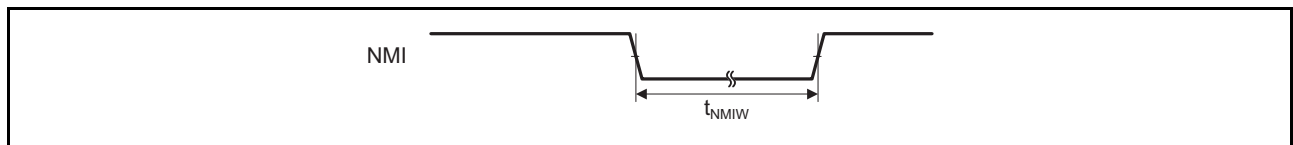


图 36.10 NMI 中断输入时序

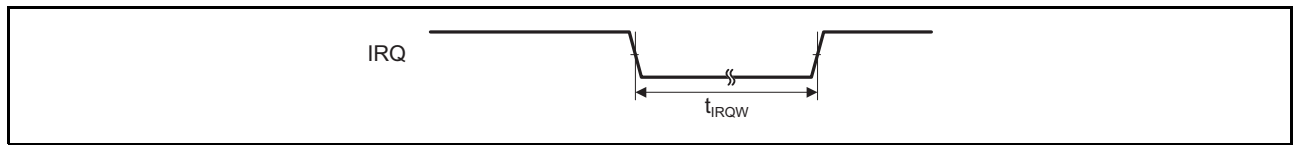


图 36.11 IRQ 中断输入时序

36.3.5 内部外围模块时序

表 36.10 内部外围模块时序 (1)

条件: VCC=2.7 ~ 3.6V、VSS=AVSS0=VREFL0=0V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、 $T_a=T_{opr}$

项目		符号	min	max	单位 (注1)	测量条件		
I/O 端口	输入数据脉宽	t_{PRW}	1.5	—	t_{Pcyc}	图 36.12		
MTU3	输入捕捉的输入脉宽	单边沿指定	t_{TICW}	3	—	t_{PAcyc}	图 36.13	
		双边沿指定		5	—			
	定时器时钟脉宽	单边沿指定	t_{TCKWH} 、 t_{TCKWL}	3	—	t_{PAcyc}	图 36.14	
双边沿指定		5		—				
相位测量模式		5	—					
POE3	POE# 输入脉宽	t_{POEW}	1.5	—	t_{Pcyc}	图 36.16		
GPT	输入捕捉的输入脉宽	单边沿指定	t_{GTICW}	3	—	t_{PAcyc}	图 36.15	
		双边沿指定		5	—			
SCI	输入时钟周期	异步	t_{Scyc}	4	—	t_{Pcyc}	图 36.17	
		时钟同步		6	—			
	输入时钟脉宽		t_{SCKW}	0.4	0.6	t_{Scyc}		
	输入时钟的上升时间		t_{SCKr}	—	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}	16	—	t_{Pcyc}		
		时钟同步		4	—			
	输出时钟脉宽		t_{SCKW}	0.4	0.6	t_{Scyc}		
	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	时钟同步	t_{TXD}	—	40	ns		图 36.18
	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	12 位 A/D 转换器的触发输入脉宽	t_{TRGW}	1.5	—	t_{Pcyc}	图 36.19		

注 1. t_{Pcyc} : PCLK 的周期、 t_{PAcyc} : PCLKA 的周期

表 36.12 内部外围模块时序 (3)

条件: $V_{CC}=2.7 \sim 3.6V$ 、 $V_{SS}=AVSS0=VREFL0=0V$ 、 $AVCC0=3.0 \sim 3.6V$ 、 $VREFH0=3.0V \sim AVCC0$ 、 $T_a=T_{opr}$

项目		符号	min	max	单位 (注1)	测量条件	
简易 SPI	SCK 时钟周期输出 (主控)	t_{SPCyc}	4	65536	t_{PCyc}	图 36.20	
	SCK 时钟周期输入 (从属)		8	65536			
	SCK 时钟的 High 电平脉宽	t_{SPCKWH}	0.4	0.6	t_{SPCyc}		
	SCK 时钟的 Low 电平脉宽	t_{SPCKWL}	0.4	0.6	t_{SPCyc}		
	SCK 时钟的上升 / 下降时间	t_{SPCKr} 、 t_{SPCKf}	—	20	ns		
	数据输入的准备时间	t_{SU}	40	—	ns	图36.21~ 图 36.24	
	数据输入的保持时间	t_H	40	—	ns		
	SS 输入的准备时间	t_{LEAD}	1	—	t_{SPCyc}		
	SS 输入的保持时间	t_{LAG}	1	—	t_{SPCyc}		
	数据输出的延迟时间	t_{OD}	—	40	ns		
	数据输出的保持时间	t_{OH}	10	—	ns		
	数据上升 / 下降时间	t_{Dr} 、 t_{Df}	—	20	ns		
	SS 输入的上升 / 下降时间	t_{SSLr} 、 t_{SSLf}	—	20	ns		
	从属存取时间	t_{SA}	—	5	t_{PCyc}		图 36.23
	从属输出释放时间	t_{REL}	—	5	t_{PCyc}		图 36.24

注 1. t_{PCyc} : PCLK 的周期

表 36.13 内部外围模块时序 (4)

条件: VCC=2.7 ~ 3.6V、VSS=AVSS0=VREFL0=0V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、 $T_a=T_{opr}$

项目		符号	min (注1、注2、注3)	max	单位	测量条件
RIIC (Standard-mode、)	SCL 输入的周期时间	t_{SCL}	$6(12) \times t_{IICcyc} + 1300$	—	ns	图 36.25
	SCL 输入的 High 电平脉宽	t_{SCLH}	$3(6) \times t_{IICcyc} + 300$	—	ns	
	SCL 输入的 Low 电平脉宽	t_{SCLL}	$3(6) \times t_{IICcyc} + 300$	—	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 输入的尖峰脉冲消除时间	t_{SP}	0	$1(4) \times t_{IICcyc}$	ns	
	SDA 输入的总线空闲时间	t_{BUF}	$3(6) \times t_{IICcyc} + 300$	—	ns	
	开始条件输入的保持时间	t_{STAH}	$t_{IICcyc} + 300$	—	ns	
	重新发送开始条件输入的准备时间	t_{STAS}	1000	—	ns	
	停止条件输入的准备时间	t_{STOS}	1000	—	ns	
	数据输入的准备时间	t_{SDAS}	$t_{IICcyc} + 50$	—	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	—	400	pF	
RIIC (Fast-mode)	SCL 输入的周期时间	t_{SCL}	$6(12) \times t_{IICcyc} + 600$	—	ns	
	SCL 输入的 High 电平脉宽	t_{SCLH}	$3(6) \times t_{IICcyc} + 300$	—	ns	
	SCL 输入的 Low 电平脉宽	t_{SCLL}	$3(6) \times t_{IICcyc} + 300$	—	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 输入的尖峰脉冲消除时间	t_{SP}	0	$1(4) \times t_{IICcyc}$	ns	
	SDA 输入的总线空闲时间	t_{BUF}	$3(6) \times t_{IICcyc} + 300$	—	ns	
	开始条件输入的保持时间	t_{STAH}	$t_{IICcyc} + 300$	—	ns	
	重新发送开始条件输入的准备时间	t_{STAS}	300	—	ns	
	停止条件输入的准备时间	t_{STOS}	300	—	ns	
	数据输入的准备时间	t_{SDAS}	$t_{IICcyc} + 50$	—	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	—	400	pF	

注. t_{IICcyc} : RIIC 的内部基准时钟 (IIC ϕ) 的周期。

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 36.14 内部外围模块时序 (5)

条件: VCC=2.7 ~ 3.6V、VSS=AVSS0=VREFL0=0V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、 $T_a=T_{opr}$

项目		符号	min (注1、注2)	max	单位	测量条件
简易 IIC (Standard-mode)	SCL 输入和 SDA 输入的上升时间	t_{Sr}	—	1000	ns	图 36.25
	SCL 输入和 SDA 输入的下降时间	t_{Sf}	—	300	ns	
	SCL 输入和 SDA 输入的尖峰脉冲消除时间	t_{SP}	0	$4 \times (1/PCLK)$	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	
简易 IIC (Fast-mode)	SCL 输入和 SDA 输入的上升时间	t_{Sr}	$20+0.1C_b$	300	ns	图 36.25
	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 输入的尖峰脉冲消除时间	t_{SP}	0	$4 \times (1/PCLK)$	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	

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。

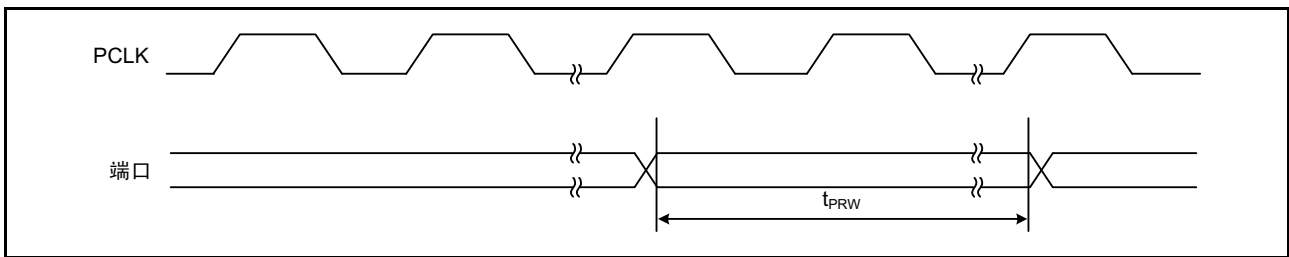


图 36.12 I/O 端口输入时序

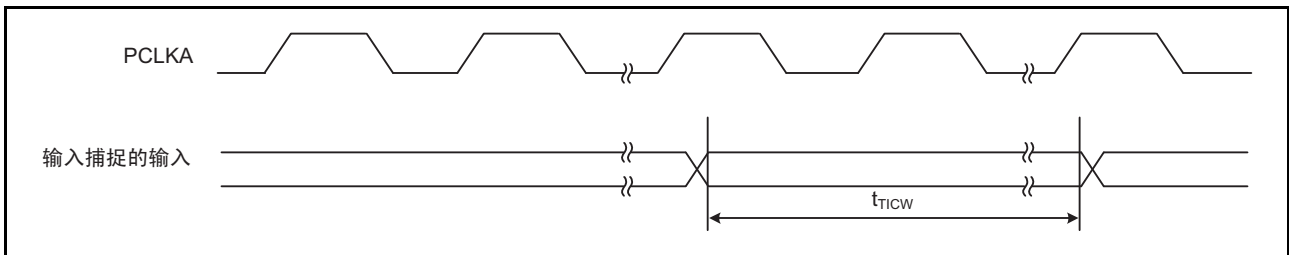


图 36.13 MTU3 输入 / 输出时序

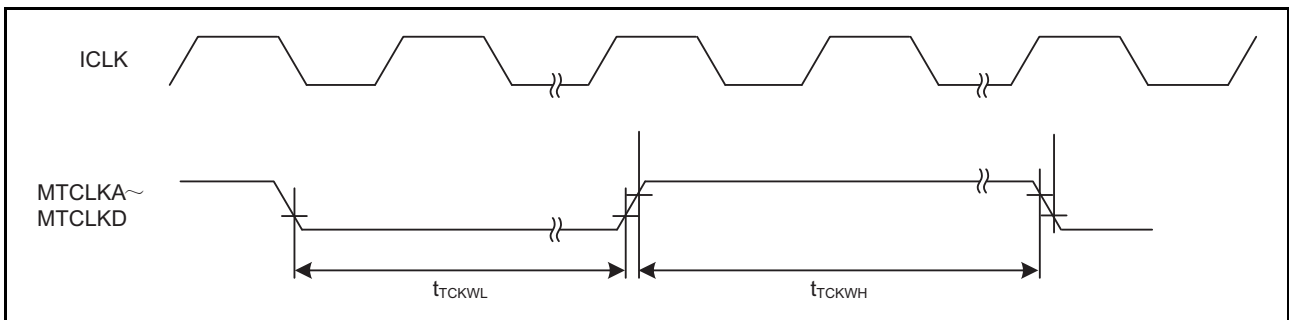


图 36.14 MTU3 时钟输入的时序

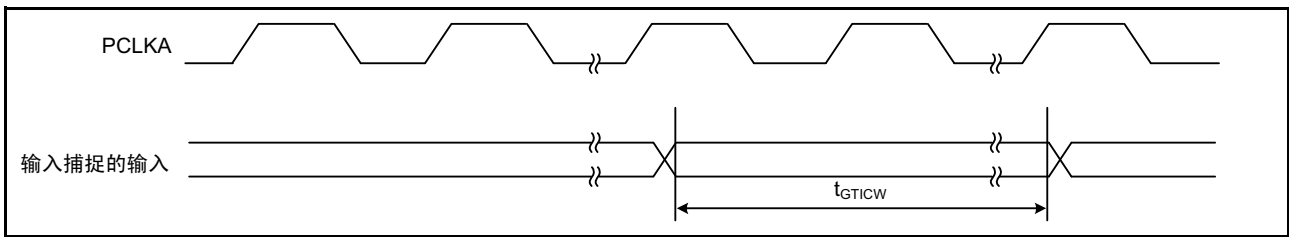


图 36.15 GPT 输入 / 输出时序

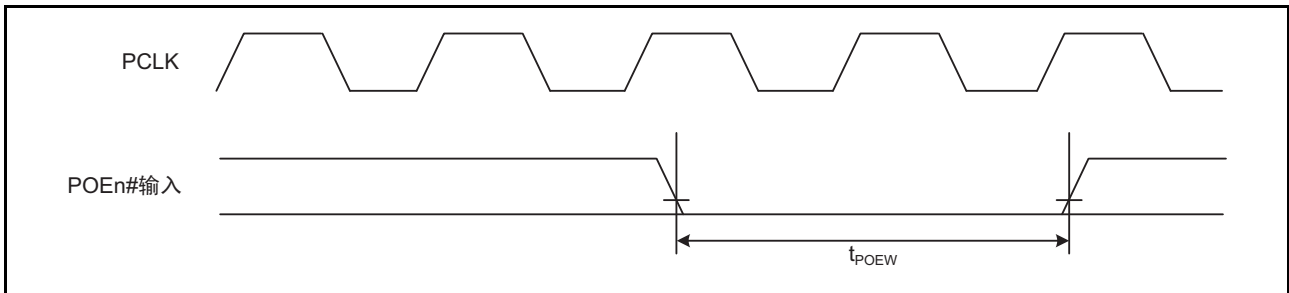


图 36.16 POE3# 输入时序

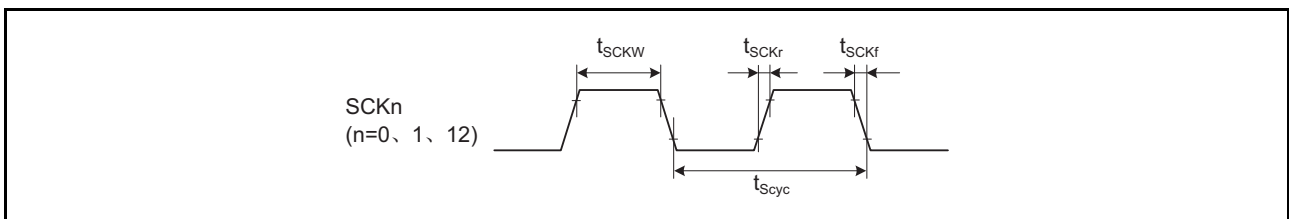


图 36.17 SCK 时钟输入时序

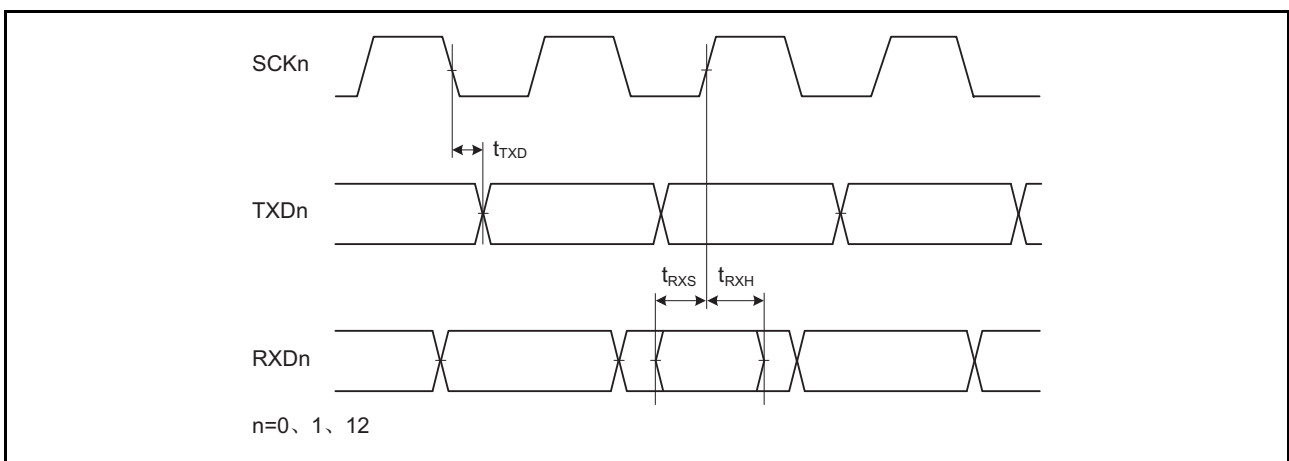


图 36.18 SCI 输入 / 输出时序 / 时钟同步模式

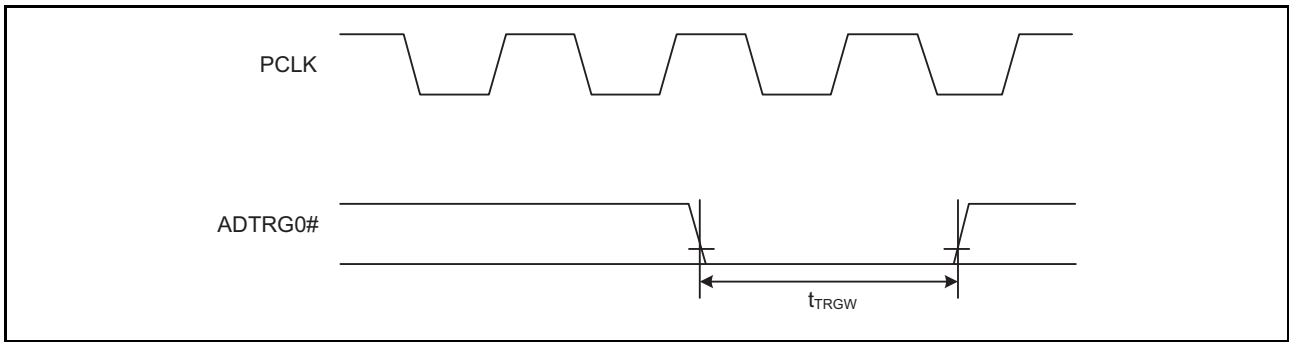


图 36.19 A/D 转换器外部触发输入的时序

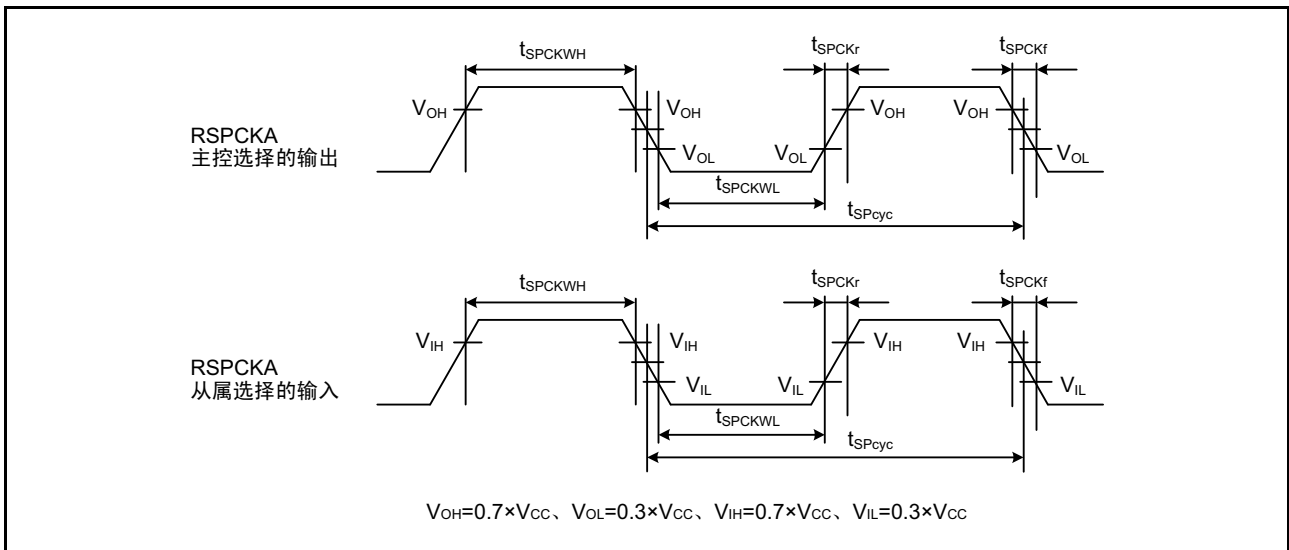


图 36.20 RSPCKA 时钟时序 / 简易 SPI 时钟时序

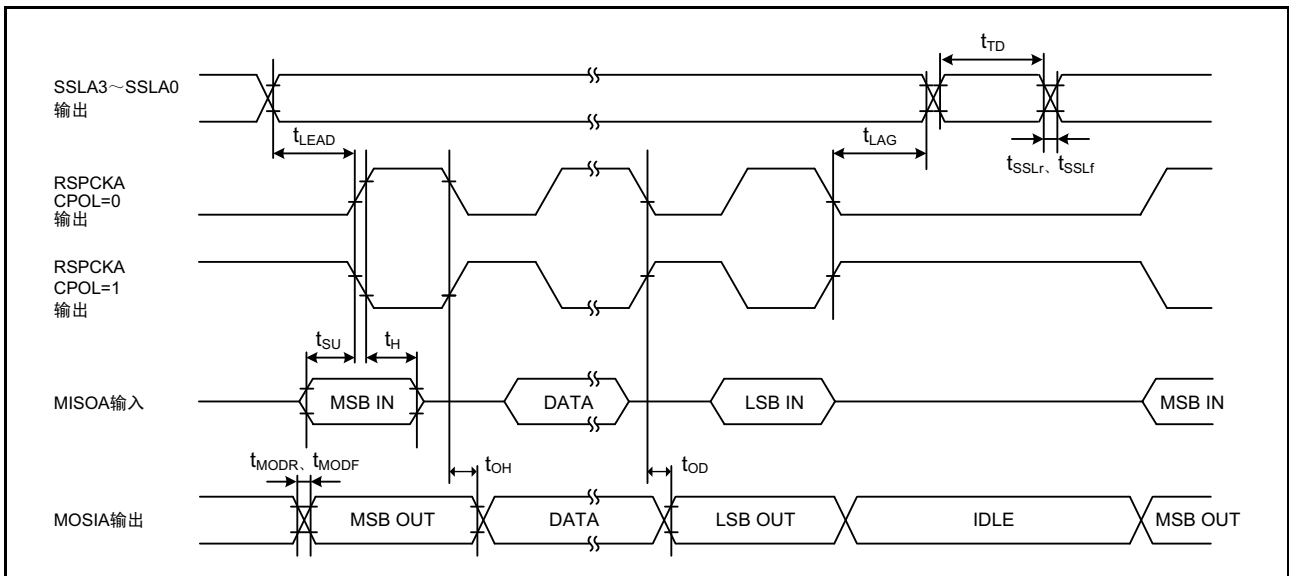


图 36.21 RSPCKA 时序 (主控、CPHA=0) / 简易 SPI 时序 (主控、CKPH=1)

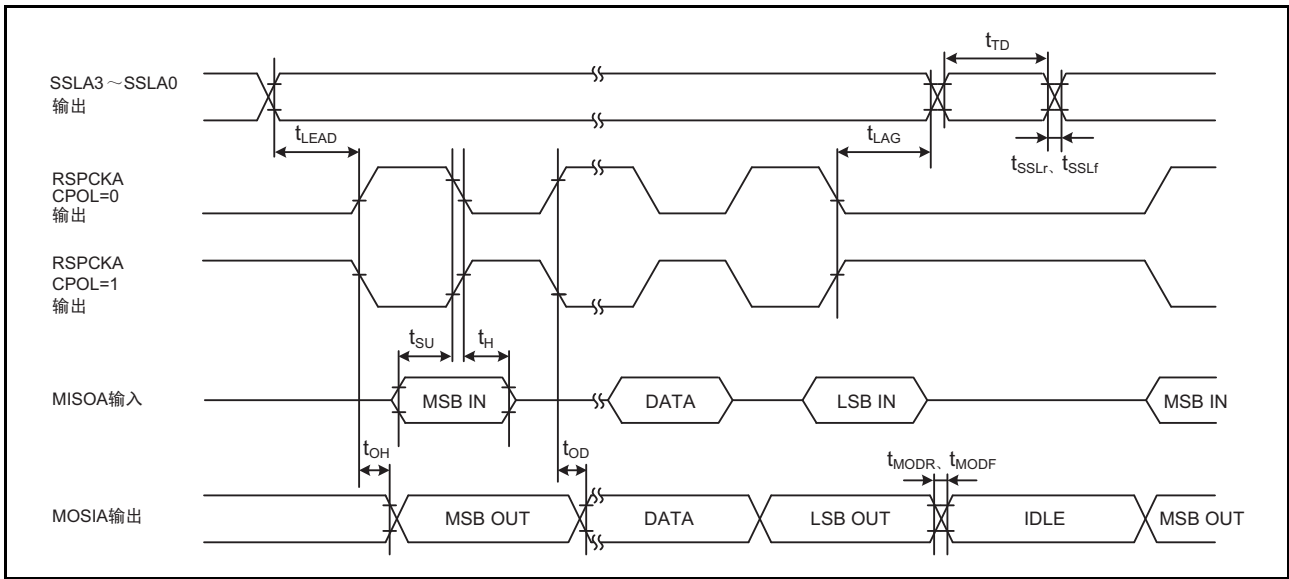


图 36.22 RSPI 时序 (主控、CPHA=1) / 简易 SPI 时序 (主控、CKPH=0)

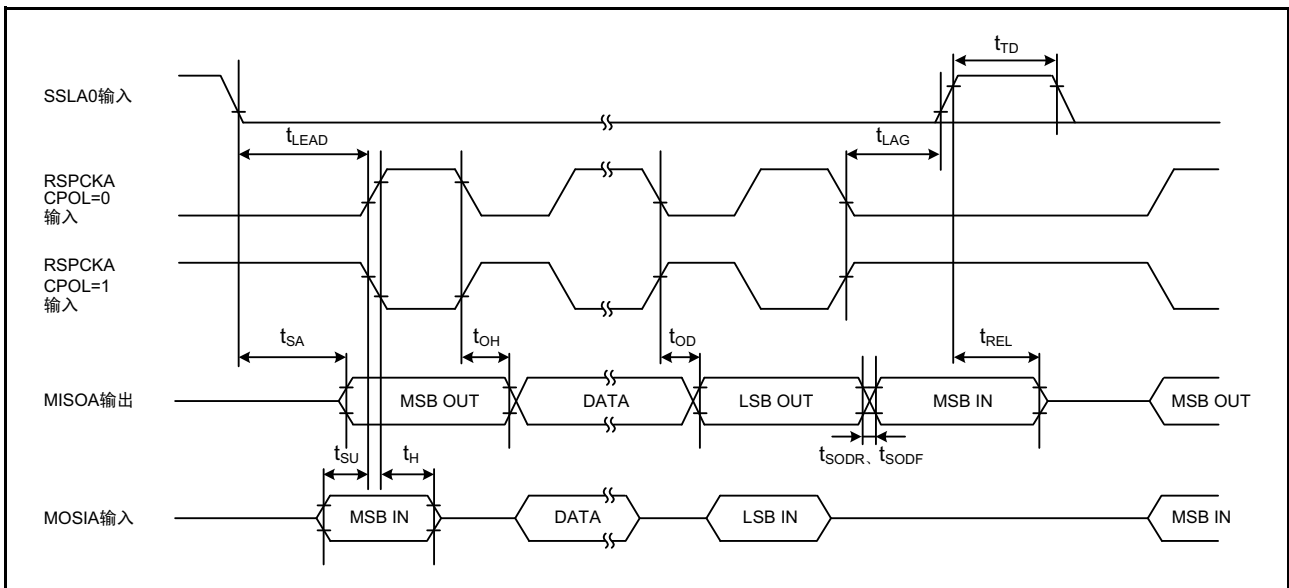


图 36.23 RSPI 时序 (从属、CPHA=0) / 简易 SPI 时序 (从属、CKPH=1)

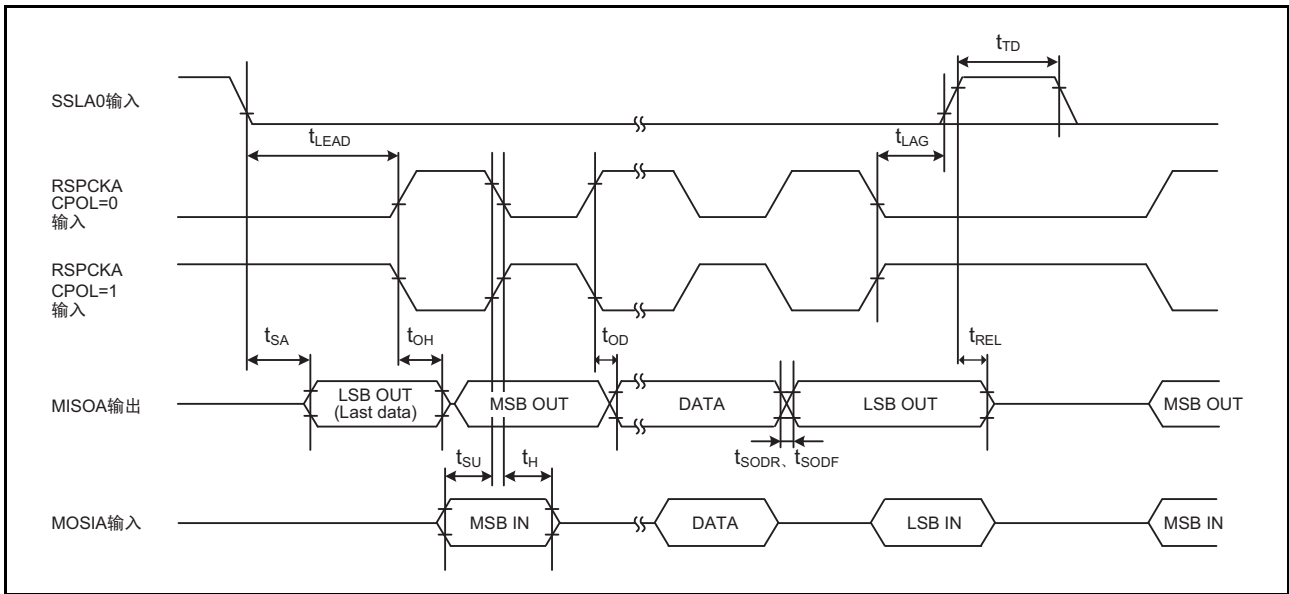


图 36.24 RSPCKA 时序 (从属、CPHA=1) / 简易 SPI 时序 (从属、CKPH=0)

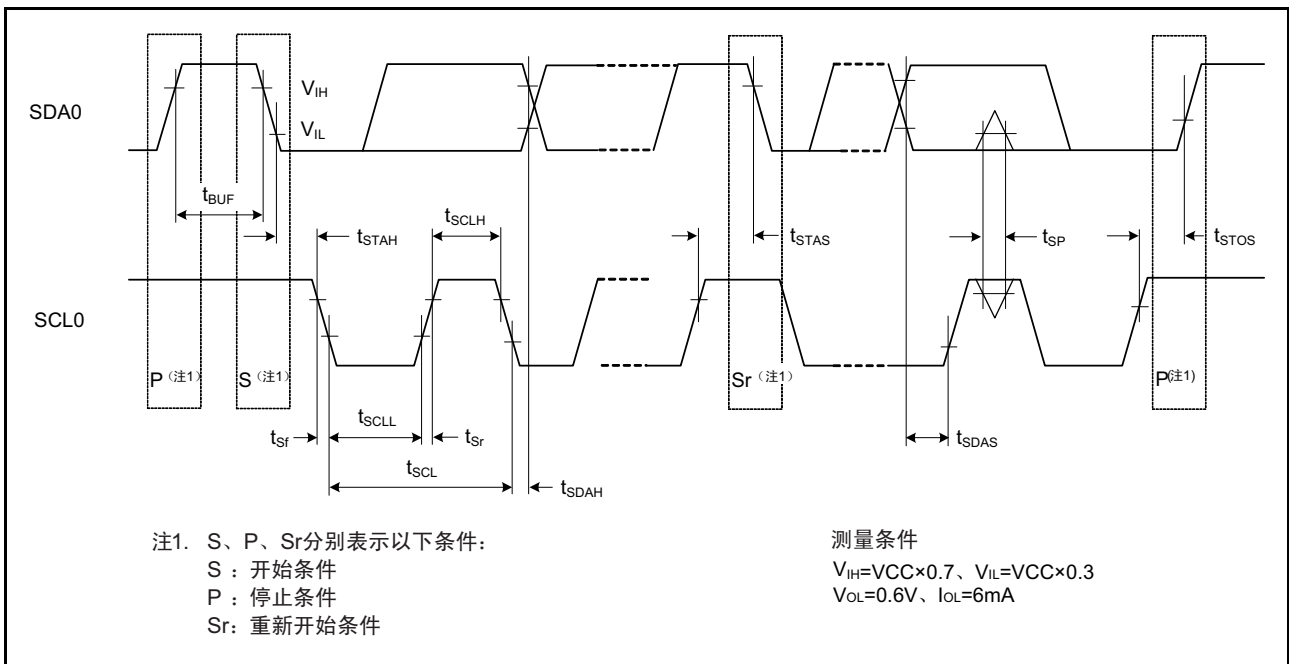


图 36.25 I2C 总线接口输入 / 输出时序 / 简易 I2C 总线接口输入 / 输出时序

36.4 A/D 转换特性

表 36.15 12 位 A/D 转换特性

条件: $V_{CC}=2.7 \sim 3.6V$ 、 $V_{SS}=AV_{SS0}=V_{REFL0}=0V$ 、 $AV_{CC0}=3.0 \sim 3.6V$ 、 $V_{REFH0}=3.0V \sim AV_{CC0}$ 、 $T_a=T_{opr}$

项目		min	typ	max	单位	测量条件
分辨率能		12	12	12	位	
转换时间 (注 1) (ADCLK= 50MHz 时)	每个引脚使用采样 & 保持电路时	1.6	—	—	μs	30 个采样 & 保持电路的采样 状态 20 个 A/D 转换器的采样状态
	每个引脚都未使用采样 & 保持电路时	1.0	—	—	μs	20 个 A/D 转换器的采样状态
模拟输入的电容		—	—	6	pF	
非线性误差		—	—	± 4.0	LSB	
偏移误差		—	—	± 7.5	LSB	
满刻度误差		—	—	± 7.5	LSB	
量化误差		—	± 0.5	—	LSB	
绝对精度	使用采样 & 保持电路时	—	—	± 8.0	LSB	$AV_{in}=0.25 \sim AV_{REFH}-0.25$
	未使用采样 & 保持电路时	—	—	± 8.0	LSB	$AV_{in}=AV_{REFL} \sim AV_{REFH}$
容许信号源阻抗		—	—	3.0	k Ω	

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状态数。

表 36.16 比较器特性

条件: $V_{CC}=2.7 \sim 3.6V$ 、 $V_{SS}=AV_{SS0}=V_{REFL0}=0V$ 、 $AV_{CC0}=3.0 \sim 3.6V$ 、 $V_{REFH0}=3.0V \sim AV_{CC0}$ 、 $T_a=T_{opr}$

项目	符号	min	typ	max	单位	测量条件
模拟输入的电容	C_{in}	—	—	6	pF	
REFH 引脚的偏移电压	V_{off}	—	—	5	mV	
REFL 引脚的偏移电压		—	—	5	mV	
REFH 输入电压范围	V_{in}	1.7	—	$AV_{CC0}-0.3$	V	
REFL 输入电压范围		0.3	—	$AV_{CC0}-1.7$	V	
REFH 响应时间	t_{CR}	—	—	0.5	μs	
REFL 响应时间	t_{CF}	—	—	0.5	μs	

36.5 上电复位电路和电压检测电路的特性

表 36.17 上电复位电路和电压检测电路的特性

条件: VCC=2.7 ~ 3.6V、VSS=AVSS0=VREFL0=0V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、 $T_a=T_{opr}$

项目		符号	min	typ	max	单位	测量条件
电压检测电平	上电复位 (POR)	V_{POR}	2.5	2.6	2.7	V	图 36.26
	电压检测电路 (LVD0)	V_{det0}	2.7	2.8	2.9		图 36.27
	电压检测电路 (LVD1)	V_{det1}	2.80	2.95	3.10		图 36.28
	电压检测电路 (LVD2)	V_{det2}	2.80	2.95	3.10		图 36.29
内部复位时间	上电复位时间 (POR)	t_{POR}	—	4.6	—	ms	图 36.26
	LVD0 复位时间 (LVD0)	t_{LVD0}	—	4.6	—		图 36.27
	LVD1 复位时间 (LVD1)	t_{LVD1}	—	0.9	—		图 36.28
	LVD2 复位时间 (LVD2)	t_{LVD2}	—	0.9	—		图 36.29
最小 VCC 降低时间 (注 1)		t_{VOFF}	200	—	—	μs	图 36.26、 图 36.27
响应延迟时间		t_{det}	—	—	200	μs	图 36.26~ 图 36.29
LVD 运行稳定时间 (选择 LVD 有效时)		$T_{d(E-A)}$	—	—	3	μs	图 36.28
滞后宽度 (LVD1、LVD2)		V_{LVH}	—	80	—	mV	图 36.29

注 1. 最小 VCC 降低时间是 VCC 低于 POR/LVD 的电压检测电平 V_{POR} 、 V_{det1} 、 V_{det2} 的 min 值的时间。

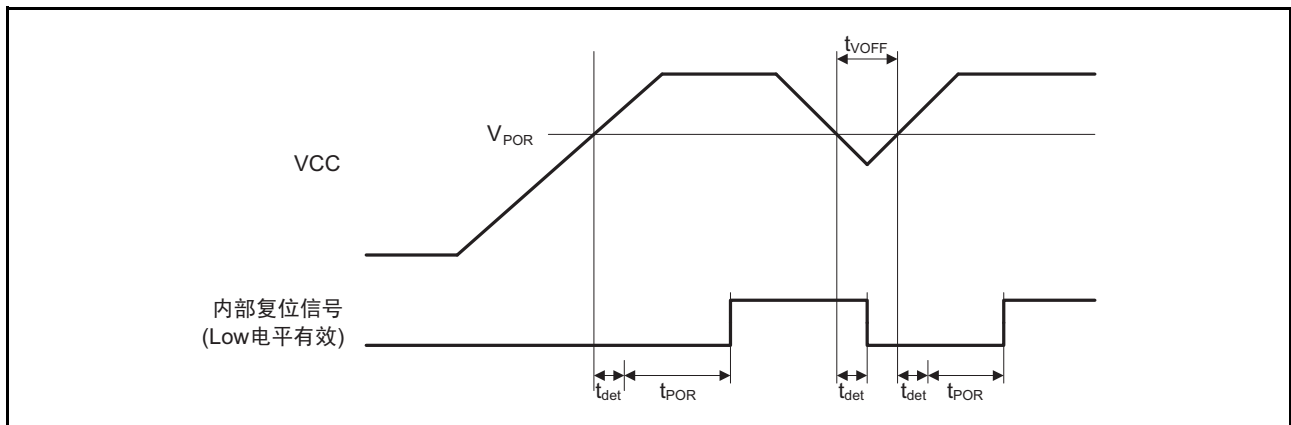


图 36.26 上电复位时序

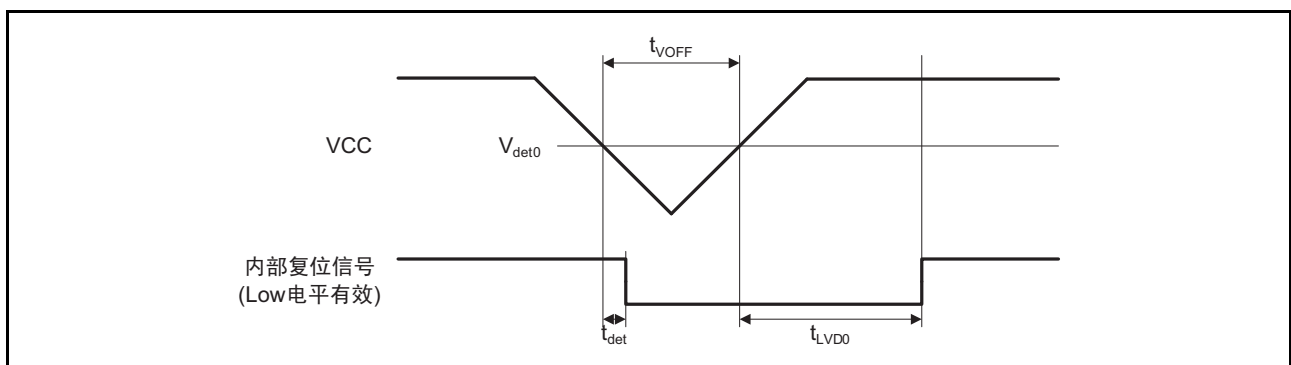


图 36.27 电压检测电路时序 (V_{det0})

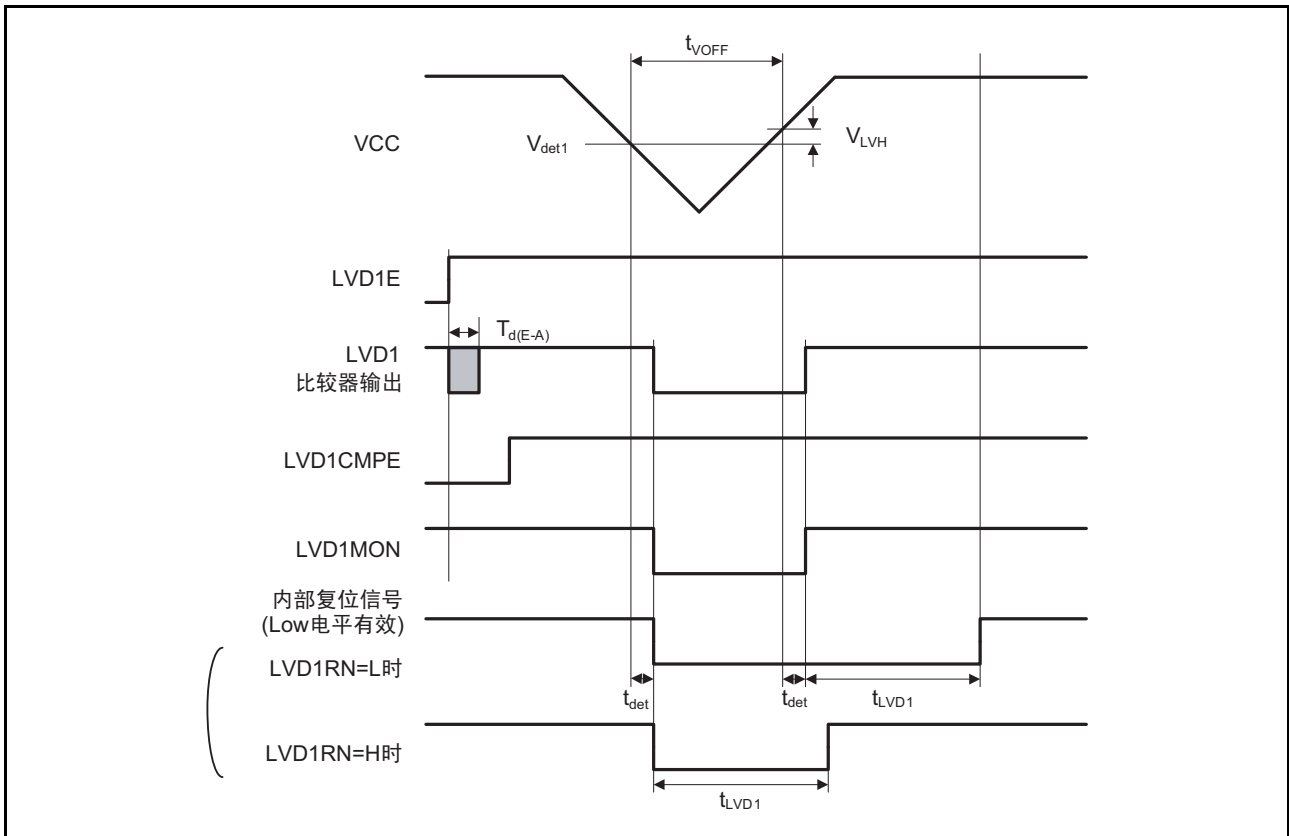


图 36.28 电压检测电路时序 (V_{det1})

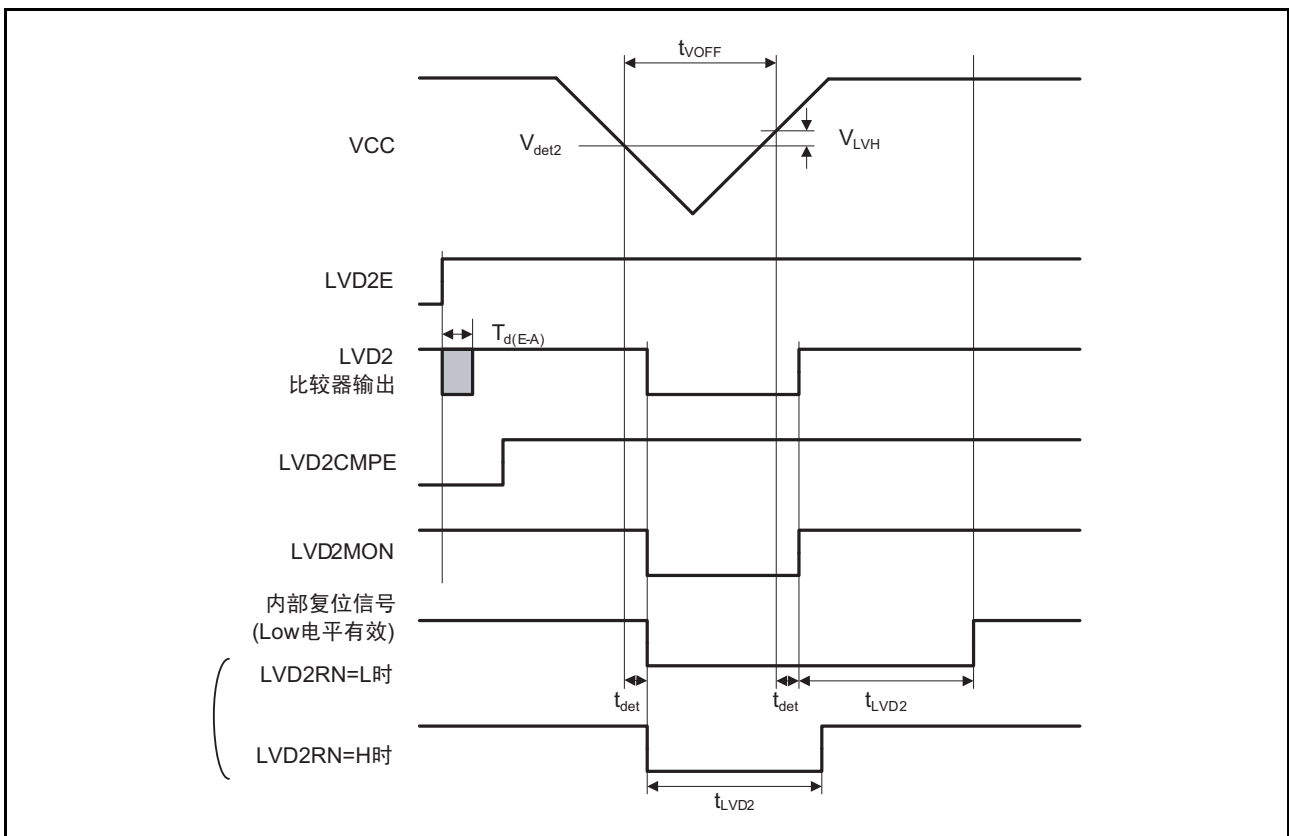


图 36.29 电压检测电路时序 (V_{det2})

36.6 振荡停止检测时序

表 36.18 振荡停止检测电路特性

条件: $VCC=2.7 \sim 3.6V$ 、 $VSS=AVSS0=VREFL0=0V$ 、 $AVCC0=3.0 \sim 3.6V$ 、 $VREFH0=3.0V \sim AVCC0$ 、 $T_a=T_{opr}$

项目	符号	min	typ	max	单位	测量条件
检测时间	t_{dr}	—	—	1.0	ms	图 36.30

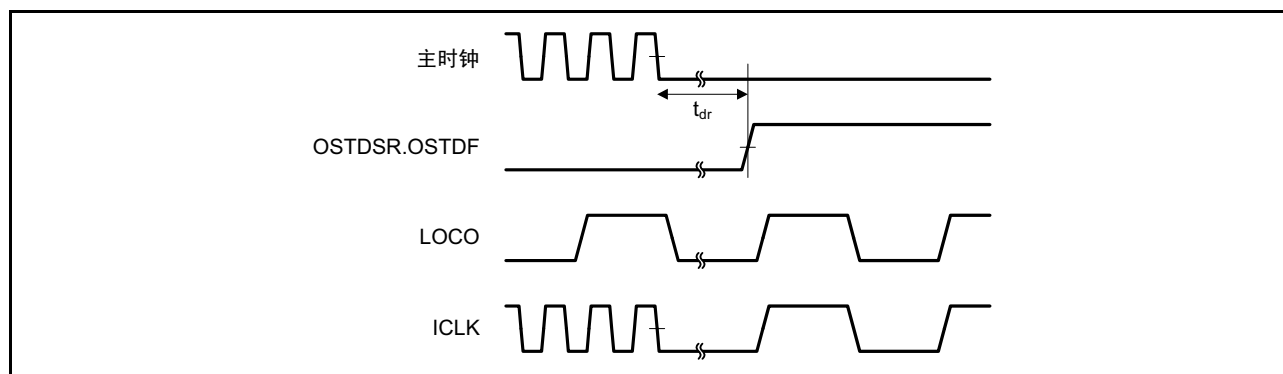


图 36.30 振荡停止检测时序

36.7 ROM（保存代码的闪存）特性

表 36.19 ROM（保存代码的闪存）特性

条件：VCC=2.7 ~ 3.6V、VSS=AVSS0=VREFL0=0V、AVCC0=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、 $T_a=T_{opr}$

项目	符号	min	typ	max	单位	测量条件		
编程时间	128 字节	t_{P128}	—	1	10	ms	FCLK=50MHz、 $N_{PEC} \leq 100$ 时	
	4K 字节	t_{P4K}	—	23	50	ms		
	16K 字节	t_{P16K}	—	90	200	ms		
	擦除时间	128 字节	t_{E128}	—	1.2	12	ms	FCLK=50MHz、 $N_{PEC} > 100$ 时
		4K 字节	t_{E4K}	—	27.6	60	ms	
		16K 字节	t_{E16K}	—	108	240	ms	
再编程 / 擦除次数 (注1)	4K 字节	t_{E4K}	—	25	60	ms	FCLK=50MHz、 $N_{PEC} \leq 100$ 时	
	16K 字节	t_{E16K}	—	100	240	ms		
	4K 字节	t_{E4K}	—	30	72	ms	FCLK=50MHz、 $N_{PEC} > 100$ 时	
	16K 字节	t_{E16K}	—	120	288	ms		
编程时的挂起延迟时间	t_{SPD}	—	—	120	μ s	图 36.31 FCLK=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	μ s			
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			
数据保持时间 (注3)	t_{DRP}	10	—	—	年			
FCU 复位时间	t_{FCUR}	35	—	—	μ s			

注 1. 再编程 / 擦除次数的定义:

再编程 / 擦除次数是指每块的擦除次数。如果再编程 / 擦除次数为 n ($n=1000$)，就能逐块擦除 n 次。例如，对于 4K 字节的块，如果将 128 字节的数据分 16 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（禁止盖写）。

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。

36.8 E2 数据闪存特性

表 36.20 E2 数据闪存特性

条件: $V_{CC}=2.7 \sim 3.6V$ 、 $V_{SS}=AVSS0=VREFL0=0V$ 、 $AVCC0=3.0 \sim 3.6V$ 、 $VREFH0=3.0V \sim AVCC0$ 、 $T_a=T_{opr}$

项目	符号	min	typ	max	单位	测量条件	
编程时间	2 字节	t_{DP2}	—	0.25	2	ms	FCLK=50MHz
擦除时间	32 字节	t_{DE32}	—	2	20	ms	FCLK=50MHz、 $N_{PEC} \leq 100$ 时
	32 字节	t_{DE32}	—	4	20	ms	FCLK=50MHz、 $N_{PEC} > 100$ 时
空白检查时间	2 字节	t_{DBC2}	—	—	30	μs	FCLK=50MHz
再编程 / 擦除次数 (注 1)		N_{DPEC}	100000 (注 2)	—	—	次	
编程时的挂起延迟时间		t_{DSPD}	—	—	120	μs	图 36.31 FCLK=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}	—	—	300	μs	
擦除时的挂起延迟时间 (在擦除优先模式中)		t_{DSEED}	—	—	300	μs	
数据的保持时间 (注 3)		t_{DDRP}	10	—	—	年	

注 1. 再编程 / 擦除次数的定义:

再编程 / 擦除次数是指每块的擦除次数。如果再编程 / 擦除次数为 n 回 ($n=100000$)，就能逐块擦除 n 次。例如，对于 32 字节的块，如果将 2 字节的数据分 16 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（禁止盖写）。

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。

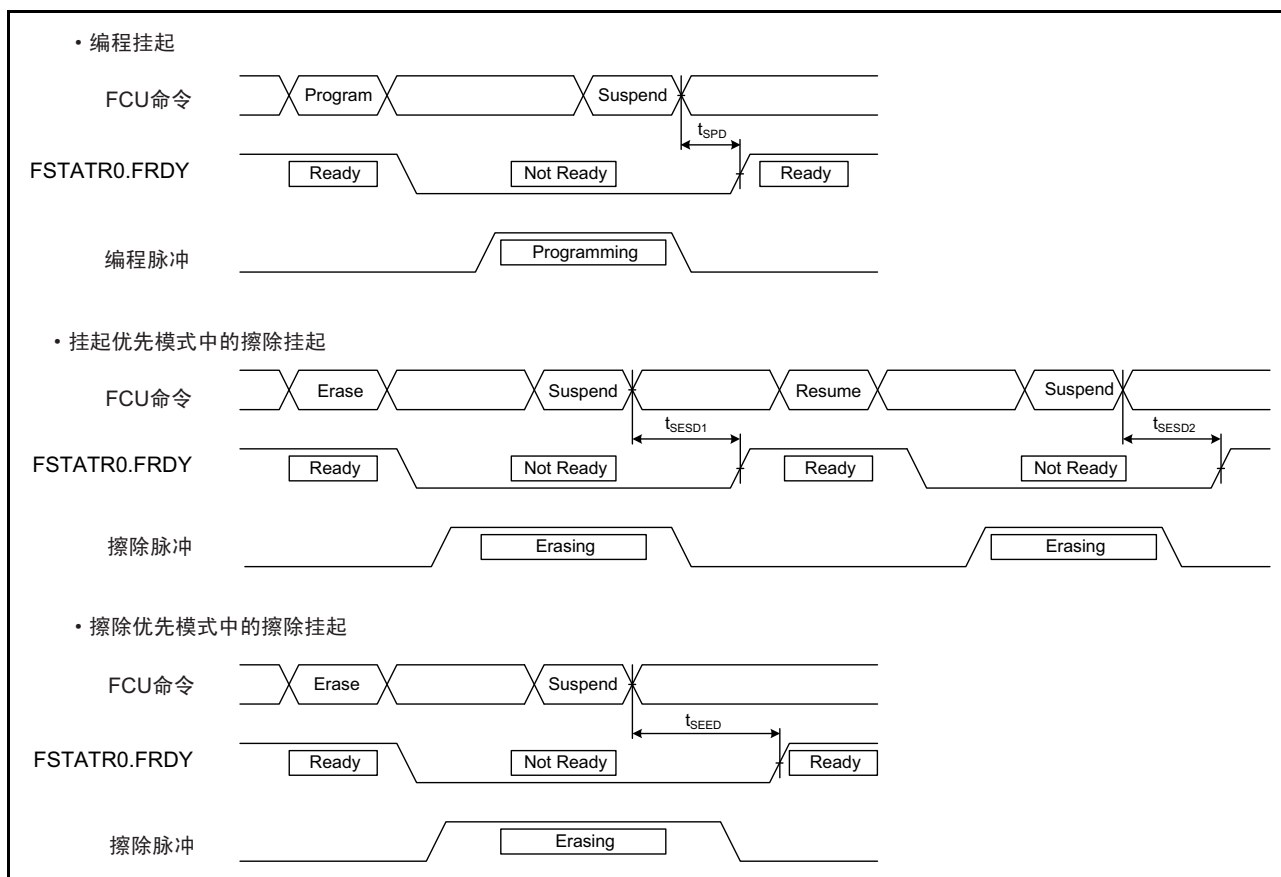


图 36.31 闪存的编程 / 擦除挂起时序

附录

附录 1. 各运行模式中的端口状态

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)	
					IOKEEP = 1 (注1)	IOKEEP = 0
P00/IRQ2-DS	全部模式	Hi-Z	Keep-O (注2)	Keep-O (注3)	Keep	Hi-Z
P01/IQR4-DS	全部模式	Hi-Z	Keep-O (注2)	Keep-O (注3)	Keep	Hi-Z
P10/IRQ0-DS	全部模式	Hi-Z	Keep-O (注2)	Keep-O (注3)	Keep	Hi-Z
P11/IRQ1-DS	全部模式	Hi-Z	Keep-O (注2)	Keep-O (注3)	Keep	Hi-Z
端口 2	全部模式	Hi-Z	Keep-O	Keep	Keep	Hi-Z
端口 3	全部模式	Hi-Z	Keep-O	Keep	Keep	Hi-Z
端口 4	全部模式	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
P70/IRQ5-DS	全部模式	Hi-Z	Keep-O (注2)	Keep-O (注3)	Keep	Hi-Z
P71 ~ P76	全部模式	Hi-Z	Keep-O	Keep	Keep	Hi-Z
P91、P92	全部模式	Hi-Z	Keep-O	Keep	Keep	Hi-Z
P93/IRQ1	全部模式	Hi-Z	Keep-O (注2)	Keep	Keep	Hi-Z
P94	全部模式	Hi-Z	Keep-O	Keep	Keep	Hi-Z
端口 A	全部模式	Hi-Z	Keep-O	Keep	Keep	Hi-Z
PB0 ~ PB3	全部模式	Hi-Z	Keep-O	Keep	Keep	Hi-Z
PB4/IRQ3-DS	全部模式	Hi-Z	Keep-O (注2)	Keep-O (注3)	Keep	Hi-Z
PB5/IRQ0	全部模式	Hi-Z	Keep-O (注2)	Keep	Keep	Hi-Z
PB6、PB7	全部模式	Hi-Z	Keep-O	Keep	Keep	Hi-Z
端口 D	全部模式	Hi-Z	Keep-O	Keep	Keep	Hi-Z
PE2/NMI	全部模式	Hi-Z	Hi-Z (注2)	Hi-Z (注3)	Hi-Z	Hi-Z

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。

Keep : 保持软件待机时的引脚状态。

HiZ : 高阻抗

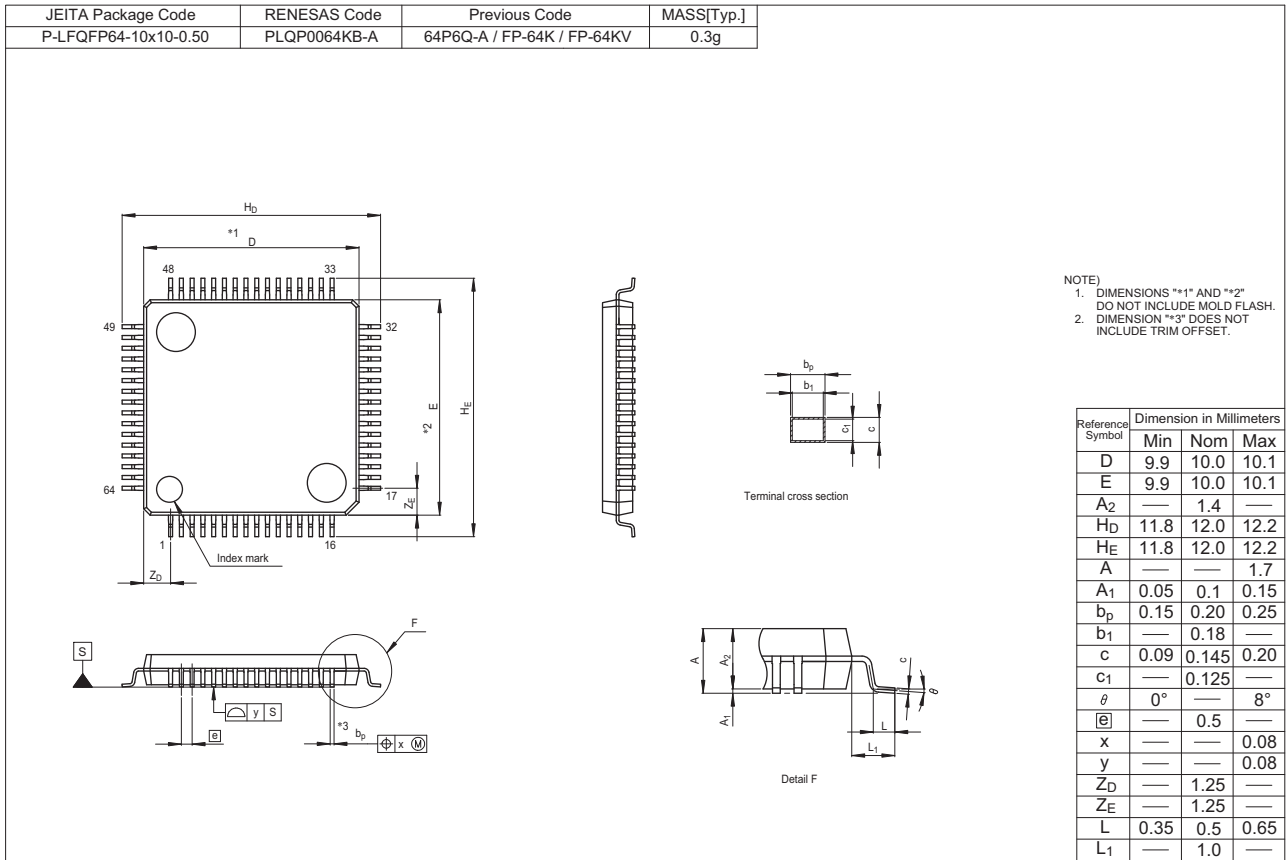
注 1. 将 DPSBYCR.IOKEEP 位置“0”前, 保持 I/O 端口的状态。

注 2. 当作为外部中断引脚使用时, 能在设定为软件待机解除源的情况下进行输入。

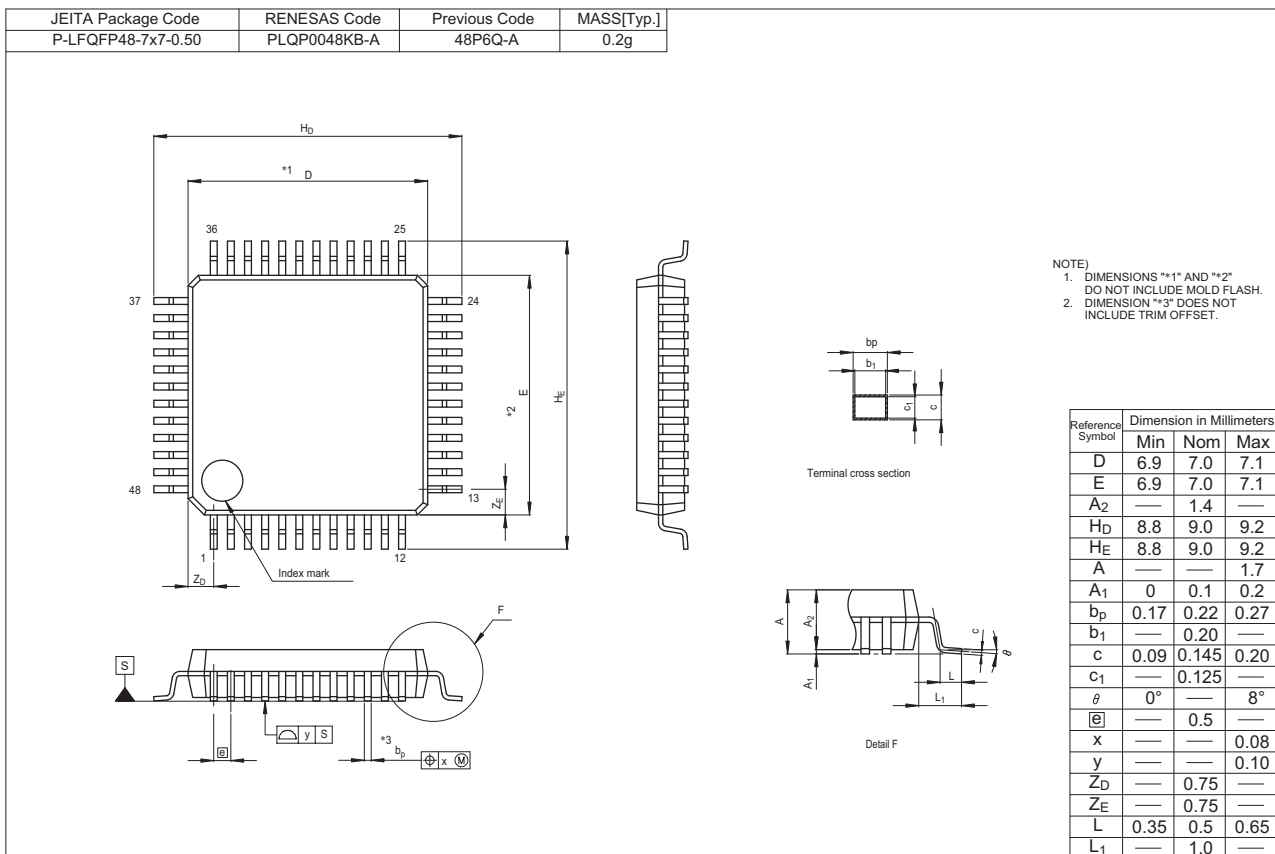
注 3. 能在设定为深度软件待机解除源的情况下进行输入。

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 64 引脚 LQFP (PLQP0064KB-A) 封装尺寸图



附图 2.2 48 引脚 LQFP (PLQP0048KB-A) 封装尺寸图

修订记录	RX63T 群 用户手册 硬件篇
------	------------------

Rev.	发行日	修订内容	
		页	修订处
1.00	2012.07.31	—	初版发行

RX63T 群
用户手册 硬件篇

Publication Date: Rev1.00 Jul 31, 2012

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: +49-211-65030, 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.**13F, No. 363, Fu Shing North Road, Taipei, Taiwan
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

RX63T群



瑞萨电子株式会社

R01UH0238CJ0100