

# RX62T 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.

## 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。

RX62T 群准备了以下的文献。请使用最新的文献。最新版本刊登在瑞萨电子的主页上。

文献的种类	记载内容	资料名	资料号
Shortsheet	硬件的概要	—	—
数据表	硬件的概要和电特性	RX62T Group Datasheet	R01DS0096EJ0100
硬件手册	硬件的说明（引脚配置、存储器映像、外围功能的说明、电特性、时序）和工作说明	RX62T 群 用户手册 硬件篇	本用户手册
软件手册	CPU 指令集的说明	RX 族 用户手册 软件篇	R01US0028CJ0100
应用注意事项	应用例子参考程序等	—	—
RENASAS TECHNICAL UPDATE	相关产品说明、文献等的快速公告	—	—

## 2 寄存器的表示

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

**X.X.X      ..... 寄存器**

地址    xxxx xxxxxh

b7	b6	b5	b4	b3	b2	b1	b0
—	.....	.....	.....	—	—	—	.....

复位后的值    x    0    0    0    0    0    0    0

位	符号	位名	功能	R/W <sup>(注1)</sup>
b0	.....0	.....位 (注2)	0: ..... (注3) 1: 不能设定。	R/W
b3-b1	—	保留位	读写值都为“0”。	R/W
b4	.....4	.....位	0: ..... 1: .....	R
b6-b5	.....[1:0]	.....位	0 0: ..... 0 1: ..... (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
SIM	Subscriber Identity Module	支持 ISO/IEC 7816 标准的接触型 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 概要 .....	2
1.2 产品一览表 .....	8
1.3 框图 .....	11
1.4 引脚排列图 .....	12
1.5 引脚功能 .....	27
2. CPU .....	32
2.1 特点 .....	32
2.2 CPU 寄存器组 .....	33
2.2.1 通用寄存器 (R0 ~ R15) .....	34
2.2.2 控制寄存器 .....	34
2.3 处理器模式 .....	41
2.3.1 管理模式 .....	41
2.3.2 用户模式 .....	41
2.3.3 特权指令 .....	41
2.3.4 处理器模式之间的转移 .....	41
2.4 数据类型 .....	42
2.4.1 整数 .....	42
2.4.2 浮点数 .....	42
2.4.3 位 .....	43
2.4.4 字符串 .....	43
2.5 字节序 .....	44
2.5.1 字节序的转换 .....	44
2.5.2 I/O 寄存器的存取 .....	47
2.5.3 存取 I/O 寄存器时的注意事项 .....	48
2.5.4 数据排列 .....	48
2.6 向量表 .....	49
2.6.1 固定向量表 .....	49
2.6.2 可向量表 .....	49
2.7 指令操作 .....	50
2.7.1 RMPA 指令和字符串操作指令的数据预取 .....	50
2.8 流水线 .....	51
2.8.1 概要 .....	51
2.8.2 指令和流水线处理 .....	52
2.8.3 指令处理时间的计算方法 .....	59
2.8.4 中断响应周期数 .....	59
3. 运行模式 .....	60
3.1 运行模式的种类和选择 .....	60
3.2 寄存器说明 .....	61
3.2.1 模式监视寄存器 (MDMONR) .....	61
3.2.2 模式状态寄存器 (MDSR) .....	62
3.2.3 系统控制寄存器 0 (SYSCR0) .....	63
3.2.4 系统控制寄存器 1 (SYSCR1) .....	64

3.3	运行模式的说明 .....	64
3.3.1	单芯片模式 .....	64
3.3.2	引导模式 .....	64
3.4	运行模式的转移 .....	65
3.4.1	通过模式引脚进行运行模式的转移 .....	65
3.4.2	通过寄存器进行运行模式的转移 .....	65
4.	地址空间 .....	66
4.1	地址空间 .....	66
5.	I/O 寄存器 .....	67
5.1	I/O 寄存器地址一览表（地址顺序） .....	69
5.2	I/O 寄存器位一览表 .....	97
6.	复位 .....	127
6.1	概要 .....	127
6.2	寄存器说明 .....	130
6.2.1	复位状态寄存器（RSTSR） .....	130
6.2.2	复位控制/状态寄存器（RSTCSR） .....	130
6.2.3	IWDT 状态寄存器（IWDTSR） .....	130
6.3	运行说明 .....	131
6.3.1	引脚复位 .....	131
6.3.2	上电复位 .....	131
6.3.3	电压监视复位 .....	132
6.3.4	深度软件待机复位 .....	132
6.3.5	独立看门狗定时器复位 .....	132
6.3.6	看门狗定时器复位 .....	132
6.4	复位发生源的判断 .....	133
6.5	使用时的注意事项 .....	133
6.5.1	设计电路板时的注意事项 .....	133
7.	电压检测电路（LVD） .....	134
7.1	概要 .....	134
7.2	寄存器说明 .....	135
7.2.1	复位状态寄存器（RSTSR） .....	135
7.2.2	低电压检测控制寄存器的键码寄存器（LVDKEYR） .....	135
7.2.3	低电压检测控制寄存器（LVDCR） .....	136
7.3	电压检测电路 .....	137
7.3.1	电压监视复位 .....	137
7.3.2	电压监视中断 .....	139
7.3.3	通过电压检测电路进行深度软件待机模式的解除 .....	141
8.	时钟发生电路 .....	142
8.1	概要 .....	142
8.2	寄存器说明 .....	143
8.2.1	系统时钟控制寄存器（SCKCR） .....	143
8.2.2	振荡停止检测控制寄存器（OSTDCR） .....	144
8.3	主时钟振荡器 .....	145
8.3.1	连接晶体谐振器的方法 .....	145
8.3.2	输入外部时钟的方法 .....	146
8.4	IWDT 专用低速内部振荡器 .....	146
8.5	振荡停止检测电路 .....	146



8.6	内部振荡电路 .....	146
8.7	PLL 电路 .....	146
8.8	分频器 .....	146
8.9	内部时钟 .....	147
8.9.1	系统时钟 (ICLK) .....	147
8.9.2	外围模块时钟 (PCLK) .....	147
8.9.3	内部振荡器时钟 (IWDTCCLK) .....	147
8.10	振荡停止检测功能 .....	148
8.10.1	振荡停止检测和检测后的运行 .....	148
8.10.2	振荡停止检测中断 .....	148
8.10.3	有关解除深度软件待机的注意事项 .....	148
8.11	使用时的注意事项 .....	149
8.11.1	有关时钟发生电路的注意事项 .....	149
8.11.2	有关谐振器的注意事项 .....	149
8.11.3	设计电路板时的注意事项 .....	150
9.	低功耗功能 .....	151
9.1	概要 .....	151
9.2	寄存器说明 .....	153
9.2.1	待机控制寄存器 (SBYCR) .....	154
9.2.2	模块停止控制寄存器 A (MSTPCRA) .....	155
9.2.3	模块停止控制寄存器 B (MSTPCRB) .....	157
9.2.4	模块停止控制寄存器 C (MSTPCRC) .....	158
9.2.5	深度待机控制寄存器 (DPSBYCR) .....	159
9.2.6	深度待机等待控制寄存器 (DPSWCR) .....	160
9.2.7	深度待机中断允许寄存器 (DPSIER) .....	161
9.2.8	深度待机中断标志寄存器 (DPSIFR) .....	162
9.2.9	深度待机中断边沿寄存器 (DPSIEGR) .....	163
9.2.10	复位状态寄存器 (RSTSR) .....	164
9.2.11	深度待机备用寄存器 (DPSBKRY) (y=0 ~ 31) .....	165
9.3	多时钟功能 .....	166
9.4	模块停止功能 .....	166
9.5	低功耗状态 .....	167
9.5.1	睡眠模式 .....	167
9.5.2	全模块时钟停止模式 .....	168
9.5.3	软件待机模式 .....	169
9.5.4	深度软件待机模式 .....	171
9.6	使用时的注意事项 .....	176
9.6.1	I/O 端口状态 .....	176
9.6.2	DTC 的模块停止 .....	176
9.6.3	内部外围模块的中断 .....	176
9.6.4	MSTPCRA、MSTPCRB、MSTPCRC 寄存器的写操作 .....	176
9.6.5	通过 DIRQnE 位 (n=1、0) 控制输入缓冲器 .....	176
9.6.6	深度软件待机模式的转移和中断的竞争 .....	176
9.6.7	WAIT 指令的执行时序 .....	176
10.	异常处理 .....	177
10.1	异常事件 .....	177
10.1.1	未定义指令异常 .....	177
10.1.2	特权指令异常 .....	177
10.1.3	存取异常 .....	177
10.1.4	浮点异常 .....	177

10.1.5	复位 .....	177
10.1.6	非屏蔽中断 .....	178
10.1.7	中断 .....	178
10.1.8	无条件陷阱 .....	178
10.2	异常处理步骤 .....	179
10.3	异常事件的接受 .....	180
10.3.1	接受时序和被保存的 PC 值 .....	180
10.3.2	向量和 PC、PSW 的保存场所 .....	180
10.4	接受异常 / 从异常返回时的硬件处理 .....	181
10.5	硬件预处理 .....	182
10.5.1	未定义指令异常 .....	182
10.5.2	特权指令异常 .....	182
10.5.3	存取异常 .....	182
10.5.4	浮点异常 .....	182
10.5.5	复位 .....	182
10.5.6	非屏蔽中断 .....	183
10.5.7	中断 .....	183
10.5.8	无条件陷阱 .....	183
10.6	从异常处理程序的返回 .....	184
10.7	异常事件的优先级 .....	184
11.	<b>中断控制器 (ICU) .....</b>	<b>185</b>
11.1	概要 .....	185
11.2	寄存器说明 .....	187
11.2.1	中断请求寄存器 i (IRi) (i= 中断向量号) .....	194
11.2.2	中断请求允许寄存器 m (IERm) (m=02h ~ 1Fh) .....	195
11.2.3	中断源优先级寄存器 m (IPRm) (m=00h ~ 90h) .....	196
11.2.4	高速中断设定寄存器 (FIR) .....	197
11.2.5	软件中断启动寄存器 (SWINTR) .....	198
11.2.6	DTC 启动允许寄存器 n (DTCERn) (n= 中断向量号) .....	199
11.2.7	IRQ 控制寄存器 n (IRQCRn) (n=0 ~ 7) .....	200
11.2.8	非屏蔽中断状态寄存器 (NMISR) .....	201
11.2.9	非屏蔽中断允许寄存器 (NMIER) .....	202
11.2.10	非屏蔽中断清除寄存器 (NMICLR) .....	203
11.2.11	NMI 引脚中断控制寄存器 (NMICR) .....	203
11.3	向量表 .....	204
11.3.1	中断向量表 .....	204
11.3.2	高速中断的向量表 .....	210
11.3.3	非屏蔽中断的向量表 .....	210
11.4	中断的运行说明 .....	211
11.4.1	中断检测 .....	211
11.4.2	中断请求的允许 / 禁止 .....	213
11.4.3	中断请求目标的选择 .....	213
11.4.4	优先级的判断 .....	214
11.4.5	高速中断 .....	215
11.4.6	外部引脚中断 .....	215
11.5	非屏蔽中断的运行说明 .....	215
11.6	从低功耗状态的返回 .....	216
11.6.1	从睡眠模式的返回 .....	216
11.6.2	从全模块时钟停止模式的返回 .....	216
11.6.3	从软件待机模式的返回 .....	216

11.7	使用时的注意事项 .....	217
11.7.1	使用 DTC 传送的通信运行注意事项 .....	217
12.	总线 .....	219
12.1	概要 .....	219
12.2	总线说明 .....	221
12.2.1	CPU 总线 .....	221
12.2.2	存储器总线 .....	221
12.2.3	内部主总线 .....	221
12.2.4	内部外围总线 .....	222
12.2.5	并行运行 .....	222
12.2.6	限制事项 .....	222
12.3	寄存器说明 .....	223
12.3.1	总线错误状态清除寄存器 (BERCLR) .....	223
12.3.2	总线错误监视允许寄存器 (BEREN) .....	223
12.3.3	总线错误状态寄存器 1 (BERSR1) .....	224
12.3.4	总线错误状态寄存器 2 (BERSR2) .....	225
12.4	总线错误监视部 .....	225
12.4.1	总线错误的种类 .....	225
12.4.2	发生总线错误时的运行 .....	225
12.4.3	总线错误的发生条件 .....	226
13.	存储器保护单元 (MPU) .....	227
13.1	概要 .....	227
13.1.1	存取控制的种类 .....	229
13.1.2	存取控制区域 .....	229
13.1.3	后台区域 .....	229
13.1.4	区域的重叠 .....	229
13.1.5	跨区域的指令和数据 .....	229
13.2	寄存器说明 .....	230
13.2.1	区域 n 起始页号寄存器 (RSPAGEn) (n=0 ~ 7) .....	231
13.2.2	区域 n 结束页号寄存器 (REPAGEn) (n=0 ~ 7) .....	232
13.2.3	存储器保护功能有效化寄存器 (MPEN) .....	233
13.2.4	后台存取控制寄存器 (MPBAC) .....	234
13.2.5	存储器保护错误状态清除寄存器 (MPECLR) .....	235
13.2.6	存储器保护错误状态寄存器 (MPESTS) .....	236
13.2.7	数据存储器保护错误地址寄存器 (MPDEA) .....	237
13.2.8	区域搜索地址寄存器 (MPSA) .....	237
13.2.9	区域搜索操作寄存器 (MPOPS) .....	238
13.2.10	区域无效化操作寄存器 (MPOPI) .....	238
13.2.11	指令命中区域寄存器 (MHITI) .....	239
13.2.12	数据命中区域寄存器 (MHITD) .....	241
13.3	功能 .....	243
13.3.1	存储器保护功能 .....	243
13.3.2	区域搜索功能 .....	243
13.3.3	存储器保护单元相关寄存器的保护 .....	243
13.3.4	存储器保护功能的存取判断流程 .....	244
13.4	存储器保护功能的使用步骤 .....	246
13.4.1	存取控制信息的设定 .....	246
13.4.2	存储器保护功能的有效化 .....	246
13.4.3	向用户模式的转移 .....	246
13.4.4	发生存储器保护错误时的处理 .....	246

14. 数据传送控制器 (DTC)	248
14.1 概要	248
14.2 寄存器说明	250
14.2.1 DTC 模式寄存器 A (MRA)	251
14.2.2 DTC 模式寄存器 B (MRB)	252
14.2.3 DTC 传送源地址寄存器 (SAR)	253
14.2.4 DTC 传送目标地址寄存器 (DAR)	253
14.2.5 DTC 传送计数寄存器 A (CRA)	254
14.2.6 DTC 传送计数寄存器 B (CRB)	255
14.2.7 DTC 控制寄存器 (DTCCR)	255
14.2.8 DTC 向量基址寄存器 (DTCVBR)	256
14.2.9 DTC 地址模式寄存器 (DTCADM0D)	256
14.2.10 DTC 模块启动寄存器 (DTCST)	257
14.2.11 DTC 状态寄存器 (DTCSTS)	258
14.3 启动源	259
14.3.1 传送信息的分配和 DTC 向量表	259
14.3.2 启动源和向量地址	261
14.4 运行说明	263
14.4.1 传送信息的跳读功能	265
14.4.2 传送信息的回写省略功能	266
14.4.3 正常传送模式	267
14.4.4 重复传送模式	268
14.4.5 块传送模式	269
14.4.6 链传送	270
14.4.7 运行时序	271
14.4.8 DTC 的执行状态	274
14.4.9 DTC 的总线权释放时序	274
14.5 DTC 的设定步骤	275
14.6 DTC 的使用例子	276
14.6.1 正常传送	276
14.6.2 计数器为“0”时的链传送	277
14.7 中断源	278
14.8 低功耗功能	278
14.9 使用时的注意事项	279
14.9.1 传送信息起始地址	279
14.9.2 传送信息的分配	279
14.9.3 中断控制器的 DTC 启动允许寄存器 (ICU.DTCERn) 的设定	280
14.9.4 给 DTC 启动源指定通信功能中断的情况	280
15. I/O 端口	281
15.1 112 引脚 LQFP 的 I/O 端口	281
15.1.1 概要	281
15.1.2 寄存器说明	285
15.1.3 端口的设定	303
15.1.4 输出允许设定一览表	304
15.1.5 未使用引脚的处理	308
15.2 100 引脚 LQFP 的 I/O 端口	309
15.2.1 概要	309
15.2.2 寄存器说明	312
15.2.3 端口的设定	328
15.2.4 输出允许设定一览表	329
15.2.5 未使用引脚的处理	333

15.3	80 引脚 LQFP 的 I/O 端口 .....	334
15.3.1	概要 .....	334
15.3.2	寄存器说明 .....	337
15.3.3	端口的设定 .....	352
15.3.4	输出允许设定一览表 .....	353
15.3.5	未使用引脚的处理 .....	356
15.4	64 引脚 LQFP 的 I/O 端口 .....	357
15.4.1	概要 .....	357
15.4.2	寄存器说明 .....	360
15.4.3	端口的设定 .....	372
15.4.4	输出允许设定一览表 .....	373
15.4.5	未使用引脚的处理 .....	376
15.5	输入 / 输出端口的结构 .....	377
15.6	使用时的注意事项 .....	379
15.6.1	输入缓冲控制寄存器 (PORTn.ICR) 的设定 .....	379
15.6.2	端口功能寄存器的设定 .....	379
15.6.3	有关输出允许设定的转换 .....	379
15.6.4	读端口寄存器 (PORT) 时的注意事项 .....	379
16.	多功能定时器脉冲单元 3 (MTU3) .....	380
16.1	概要 .....	380
16.2	寄存器说明 .....	386
16.2.1	定时器的控制寄存器 (TCR) .....	392
16.2.2	定时器的模式寄存器 1 (TMDR1) .....	396
16.2.3	定时器的模式寄存器 2 (TMDR2A、TMDR2B) .....	398
16.2.4	定时器的 I/O 控制寄存器 (TIOR) .....	398
16.2.5	定时器的比较匹配清除寄存器 (TCNTCMPCLR) .....	413
16.2.6	定时器的中断允许寄存器 (TIER) .....	414
16.2.7	定时器的状态寄存器 (TSR) .....	417
16.2.8	定时器的缓冲运行传送模式寄存器 (TBTM) .....	421
16.2.9	定时器的输入捕捉控制寄存器 (TICCR) .....	422
16.2.10	定时器的同步清除寄存器 (TSYCR) .....	423
16.2.11	定时器的计数器 (TCNT) .....	423
16.2.12	定时器的通用寄存器 (TGR) .....	424
16.2.13	定时器的启动寄存器 (TSTR) .....	425
16.2.14	定时器的同步寄存器 (TSYR) .....	427
16.2.15	定时器的计数器同步启动寄存器 (TCSYSTR) .....	429
16.2.16	定时器的读写允许寄存器 (TRWERA、TRWERB) .....	431
16.2.17	定时器的输出主控允许寄存器 (TOER) .....	432
16.2.18	定时器的输出控制寄存器 1 (TOCR1A、TOCR1B) .....	434
16.2.19	定时器的输出控制寄存器 2 (TOCR2A、TOCR2B) .....	436
16.2.20	定时器的输出电平缓冲寄存器 (TOLBRA、TOLBRB) .....	438
16.2.21	定时器的门控寄存器 A (TGCRA) .....	439
16.2.22	定时器的副计数器 (TCNTSA、TCNTSB) .....	440
16.2.23	定时器的周期数据寄存器 (TCDRA、TCDRB) .....	440
16.2.24	定时器的周期缓冲寄存器 (TCBRA、TCBRB) .....	441
16.2.25	定时器的死区时间数据寄存器 (TDDRA、TDDR B) .....	441
16.2.26	定时器的死区时间允许寄存器 (TDERA、TDERB) .....	441
16.2.27	定时器的缓冲传送设定寄存器 (TBTERA、TBTERB) .....	442
16.2.28	定时器的波形控制寄存器 (TWCRA、TWCRB) .....	443
16.2.29	定时器的 A/D 转换开始请求控制寄存器 (TADCR) .....	445
16.2.30	定时器的 A/D 转换开始请求周期设定寄存器 (TADCORA、TADCORB) .....	449

16.2.31	定时器的 A/D 转换开始请求周期设定缓冲寄存器 (TADCOBRA、TADCOBRB)	449
16.2.32	定时器的中断减少模式寄存器 (TITMRA、TITMRB)	450
16.2.33	定时器的中断减少设定寄存器 1 (TITCR1A、TITCR1B)	451
16.2.34	定时器的中断减少次数计数器 1 (TITCNT1A、TITCNT1B)	453
16.2.35	定时器的中断减少设定寄存器 2 (TITCR2A、TITCR2B)	455
16.2.36	定时器的中断减少次数计数器 2 (TITCNT2A、TITCNT2B)	457
16.2.37	和总线主控的接口	458
16.3	运行说明	459
16.3.1	基本运行	459
16.3.2	同步运行	463
16.3.3	缓冲运行	464
16.3.4	级联运行	468
16.3.5	PWM 模式	472
16.3.6	相位计数模式	477
16.3.7	复位同步 PWM 模式	483
16.3.8	互补 PWM 模式	486
16.3.9	A/D 转换开始请求的延迟功能	521
16.3.10	MTU0 ~ 4 - MTU6、7 的同步运行	526
16.3.11	外部脉宽的测量功能	528
16.3.12	死区时间的补偿功能	529
16.3.13	在互补 PWM 的“波峰 / 波谷”进行的 TCNT 捕捉运行	530
16.4	中断源	531
16.4.1	中断源和优先级	531
16.4.2	DTC 的启动	533
16.4.3	A/D 转换器的启动	533
16.5	运行时序	535
16.5.1	输入 / 输出时序	535
16.5.2	中断信号的时序	540
16.6	使用时的注意事项	546
16.6.1	模块时钟停止模式的设定	546
16.6.2	输入时钟的限制事项	546
16.6.3	设定周期时的注意事项	546
16.6.4	TCNT 的写和清除的竞争	547
16.6.5	TCNT 的写和递增计数的竞争	547
16.6.6	TGR 的写和比较匹配的竞争	548
16.6.7	缓冲寄存器的写和比较匹配的竞争	548
16.6.8	缓冲寄存器的写和 TCNT 清除的竞争	549
16.6.9	TGR 寄存器的读和输入捕捉的竞争	549
16.6.10	TGR 寄存器的写和输入捕捉的竞争	550
16.6.11	缓冲寄存器的写和输入捕捉的竞争	551
16.6.12	级联中的 MTU2.TCNT 的写和上溢 / 下溢的竞争	552
16.6.13	互补 PWM 模式停止时的计数器值	553
16.6.14	互补 PWM 模式中的缓冲运行的设定	553
16.6.15	复位同步 PWM 模式的缓冲运行和比较匹配标志	554
16.6.16	复位同步 PWM 模式的上溢标志	555
16.6.17	上溢 / 下溢和计数器清除的竞争	555
16.6.18	TCNT 的写和上溢 / 下溢的竞争	556
16.6.19	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	556
16.6.20	互补 PWM 模式、复位同步 PWM 模式的输出电平	556
16.6.21	级联中的 MTU1.TCNT 和 MTU2.TCNT 的同时输入捕捉	557
16.6.22	中断减少功能 2	557

16.7	MTU 输出引脚的初始化方法 .....	558
16.7.1	运行模式 .....	558
16.7.2	因运行过程中的异常而重新设定时的运行 .....	558
16.7.3	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 .....	559
17.	端口输出允许 3 (POE3) .....	576
17.1	概要 .....	576
17.2	寄存器说明 .....	579
17.2.1	输入电平控制 / 状态寄存器 1 (ICSR1) .....	580
17.2.2	输出电平控制 / 状态寄存器 1 (OCSR1) .....	581
17.2.3	有效电平寄存器 1 (ALR1) .....	582
17.2.4	输入电平控制 / 状态寄存器 2 (ICSR2) .....	584
17.2.5	输出电平控制 / 状态寄存器 2 (OCSR2) .....	585
17.2.6	输入电平控制 / 状态寄存器 3 (ICSR3) .....	586
17.2.7	输入电平控制 / 状态寄存器 4 (ICSR4) .....	587
17.2.8	输入电平控制 / 状态寄存器 5 (ICSR5) .....	588
17.2.9	软件端口输出允许寄存器 (SPOER) .....	589
17.2.10	端口输出允许控制寄存器 1 (POECR1) .....	591
17.2.11	端口输出允许控制寄存器 2 (POECR2) .....	592
17.2.12	端口输出允许控制寄存器 3 (POECR3) .....	594
17.2.13	端口输出允许控制寄存器 4 (POECR4) .....	595
17.2.14	端口输出允许控制寄存器 5 (POECR5) .....	597
17.2.15	端口输出允许控制寄存器 6 (POECR6) .....	598
17.3	运行说明 .....	600
17.3.1	输入电平的检测 .....	606
17.3.2	输出电平的比较 .....	607
17.3.3	通过寄存器进行的高阻抗控制 .....	607
17.3.4	通过振荡停止检测进行的高阻抗控制 .....	607
17.3.5	通过比较器检测进行的高阻抗控制 .....	607
17.3.6	高阻抗控制条件的追加功能 .....	607
17.3.7	高阻抗状态的解除 .....	608
17.4	中断 .....	608
17.5	使用时的注意事项 .....	608
18.	通用 PWM 定时器 (GPT) .....	609
18.1	概要 .....	609
18.2	寄存器说明 .....	613
18.2.1	通用 PWM 定时器的软件启动寄存器 (GTSTR) .....	616
18.2.2	通用 PWM 定时器的硬件源启动控制寄存器 (GTHSCR) .....	617
18.2.3	通用 PWM 定时器的硬件源清除控制寄存器 (GTHCCR) .....	618
18.2.4	通用 PWM 定时器的硬件启动源选择寄存器 (GTHSSR) .....	619
18.2.5	通用 PWM 定时器的硬件停止 / 清除源选择寄存器 (GTHPSR) .....	621
18.2.6	通用 PWM 定时器的写保护寄存器 (GTWP) .....	623
18.2.7	通用 PWM 定时器的同步寄存器 (GTSYNC) .....	624
18.2.8	通用 PWM 定时器的外部触发输入中断寄存器 (GTETINT) .....	625
18.2.9	通用 PWM 定时器的缓冲运行禁止寄存器 (GTBDR) .....	626
18.2.10	通用 PWM 定时器启动的写保护寄存器 (GTSWP) .....	627
18.2.11	LOCO 计数控制寄存器 (LCCR) .....	628
18.2.12	LOCO 计数状态寄存器 (LCST) .....	630
18.2.13	LOCO 计数值寄存器 (LCNT) .....	631
18.2.14	LOCO 计数结果平均寄存器 (LCNTA) .....	631
18.2.15	LOCO 计数结果寄存器 n (LCNTn) (n=00 ~ 15) .....	631

18.2.16	LOCO 计数上限 / 下限容许偏差值寄存器 (LCNTDU、LCNTDL) .....	632
18.2.17	通用 PWM 定时器的 I/O 控制寄存器 (GTIOR) .....	632
18.2.18	通用 PWM 定时器的中断输出设定寄存器 (GTINTAD) .....	636
18.2.19	通用 PWM 定时器的控制寄存器 (GTCR) .....	638
18.2.20	通用 PWM 定时器的缓冲允许寄存器 (GTBER) .....	639
18.2.21	通用 PWM 定时器的计数方向寄存器 (GTUDC) .....	641
18.2.22	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) .....	642
18.2.23	通用 PWM 定时器的状态寄存器 (GTST) .....	644
18.2.24	通用 PWM 定时器的计数器 (GTCNT) .....	647
18.2.25	通用 PWM 定时器的比较捕捉寄存器 m (GTCCRm) (m=A ~ F) .....	648
18.2.26	通用 PWM 定时器的周期设定寄存器 (GTPR) .....	648
18.2.27	通用 PWM 定时器的周期设定缓冲寄存器 (GTPBR) .....	648
18.2.28	通用 PWM 定时器的周期设定双缓冲寄存器 (GTPDBR) .....	649
18.2.29	A/D 转换开始请求时序寄存器 m (GTADTRm) (m=A、B) .....	649
18.2.30	A/D 转换开始请求时序缓冲寄存器 m (GTADTBRm) (m=A、B) .....	649
18.2.31	A/D 转换开始请求时序双缓冲寄存器 m (GTADTDBRm) (m=A、B) .....	649
18.2.32	通用 PWM 定时器的输出无效控制寄存器 (GTONCR) .....	650
18.2.33	死区时间控制寄存器 (GDTTCR) .....	652
18.2.34	通用 PWM 定时器的死区时间值寄存器 m (GTDVm) (m=U、D) .....	653
18.2.35	通用 PWM 定时器的死区时间缓冲寄存器 m (GTDBm) (m=U、D) .....	653
18.2.36	通用 PWM 定时器的输出保护功能状态寄存器 (GTSOS) .....	654
18.2.37	通用 PWM 定时器的输出保护功能暂时解除寄存器 (GTSOTR) .....	654
18.3	运行说明 .....	655
18.3.1	基本运行 .....	655
18.3.2	缓冲运行 .....	664
18.3.3	PWM 输出运行模式 .....	675
18.3.4	死区时间自动设定功能 .....	687
18.3.5	计数方向转换功能 .....	691
18.3.6	硬件启动 / 停止、清除运行 .....	692
18.3.7	同步运行 .....	697
18.3.8	PWM 输出的运行例子 .....	705
18.4	中断源 .....	711
18.4.1	中断源和优先级 .....	711
18.4.2	DTC 的启动 .....	714
18.4.3	中断、A/D 转换开始请求的减少功能 .....	714
18.5	A/D 转换开始请求 .....	718
18.6	LOCO 计数功能 .....	720
18.7	保护功能 .....	723
18.7.1	计数器的写保护 .....	723
18.7.2	缓冲运行的抑制 .....	723
18.7.3	GTIOC 引脚输出的无效控制 .....	724
18.7.4	GTIOC 引脚输出的输出保护功能 .....	725
18.7.5	通过 POE 功能进行 GTIOC 引脚输出的高阻抗控制 .....	729
18.8	输出引脚的初始化方法 .....	730
18.8.1	复位后的引脚设定 .....	730
18.8.2	因运行中的异常等引起的引脚初始化 .....	730
18.9	使用时的注意事项 .....	731
18.9.1	模块停止功能的设定 .....	731
18.9.2	比较匹配运行时的 GTCCRn 寄存器的设定 (n=A、B、C、D、E、F) .....	731
18.9.3	安全停止定时器的方法 .....	732
18.9.4	使用 LOCO 计数功能时的低功耗功能的设定 .....	732



19.	比较匹配定时器 (CMT)	733
19.1	概要	733
19.2	寄存器说明	734
19.2.1	比较匹配定时器的启动寄存器 0 (CMSTR0)	734
19.2.2	比较匹配定时器的启动寄存器 1 (CMSTR1)	735
19.2.3	比较匹配定时器的控制寄存器 (CMCR)	736
19.2.4	比较匹配定时器的计数器 (CMCNT)	737
19.2.5	比较匹配定时器的常数寄存器 (CMCOR)	737
19.3	运行说明	738
19.3.1	周期计数	738
19.3.2	CMCNT 计数器的计数时序	738
19.4	中断	739
19.4.1	中断源	739
19.4.2	比较匹配中断的发生时序	739
19.5	使用时的注意事项	740
19.5.1	模块停止功能的设定	740
19.5.2	比较匹配定时器的计数器 (CMCNT) 的写和比较匹配的竞争	740
19.5.3	比较匹配定时器的计数器 (CMCNT) 的写和递增计数的竞争	740
19.5.4	改写比较匹配定时器的控制寄存器 (CMCR) 时的注意事项	741
19.5.5	比较匹配定时器的计数器 (CMCNT) 和比较匹配常数寄存器 (CMCOR) 的注意事项	741
20.	看门狗定时器 (WDT)	742
20.1	概要	742
20.2	寄存器说明	743
20.2.1	定时器的计数器 (TCNT)	743
20.2.2	定时器的控制 / 状态寄存器 (TCSR)	744
20.2.3	复位控制 / 状态寄存器 (RSTCSR)	745
20.2.4	写窗口 A 寄存器 (WINA)	746
20.2.5	写窗口 B 寄存器 (WINB)	746
20.3	运行说明	747
20.3.1	看门狗定时器模式	747
20.3.2	间隔定时器模式	748
20.4	中断源	748
20.5	使用时的注意事项	749
20.5.1	存取寄存器时的注意事项	749
20.5.2	定时器的计数器 (TCNT) 的写和递增计数的竞争	750
20.5.3	CKS[2:0] 位的改写	750
20.5.4	看门狗定时器模式和间隔定时器模式的转换	751
20.5.5	看门狗定时器模式的内部复位	751
20.5.6	通过 WDTOVF# 信号进行系统复位	751
20.5.7	向看门狗定时器模式和软件待机模式的转移	751
21.	独立看门狗定时器 (IWDT)	752
21.1	概要	752
21.2	寄存器说明	753
21.2.1	IWDT 刷新寄存器 (IWDTRR)	753
21.2.2	IWDT 控制寄存器 (IWDTCR)	754
21.2.3	IWDT 状态寄存器 (IWDTSR)	756
21.3	运行说明	757
21.3.1	递减计数器的计数	757
21.3.2	IWDT 控制寄存器的写控制	758
21.3.3	刷新	758

21.3.4	状态标志 .....	759
21.4	使用时的注意事项 .....	759
21.4.1	向低功耗功能转移时的限制事项 .....	759
22.	串行通信接口 (SC1b) .....	760
22.1	概要 .....	760
22.2	串行通信接口模式 .....	762
22.2.1	寄存器说明 .....	762
22.2.2	异步模式的运行 .....	778
22.2.3	多处理器通信功能 .....	788
22.2.4	时钟同步模式的运行 .....	793
22.3	智能卡接口模式 .....	800
22.3.1	寄存器说明 .....	800
22.3.2	智能卡接口模式的运行 .....	808
22.3.3	数据格式 (块传送模式除外) .....	809
22.4	噪声消除功能 .....	817
22.5	中断源 .....	818
22.5.1	串行通信接口模式的中断 .....	818
22.5.2	智能卡接口模式的中断 .....	819
22.6	使用时的注意事项 .....	820
22.6.1	模块停止功能的设定 .....	820
22.6.2	有关中止的检测和处理 .....	820
22.6.3	标记状态和中止的发送 .....	820
22.6.4	有关接收错误标志和发送 (只限于时钟同步模式) .....	820
22.6.5	有关写 TDR .....	820
22.6.6	时钟同步发送时的限制事项 .....	820
22.6.7	使用 DTC 时的限制事项 .....	820
22.6.8	有关低功耗状态时的运行 .....	821
22.6.9	时钟同步模式的外部时钟输入 .....	823
23.	CRC 运算器 (CRC) .....	824
23.1	概要 .....	824
23.2	寄存器说明 .....	825
23.2.1	CRC 控制寄存器 (CRCCR) .....	825
23.2.2	CRC 数据输入寄存器 (CRCDIR) .....	826
23.2.3	CRC 数据输出寄存器 (CRCDOR) .....	826
23.3	CRC 运算器的运行说明 .....	827
23.4	使用时的注意事项 .....	830
23.4.1	模块待机功能的设定 .....	830
23.5	传送时的注意事项 .....	830
24.	I <sup>2</sup> C 总线接口 (RIIC) .....	831
24.1	概要 .....	831
24.2	寄存器说明 .....	834
24.2.1	I <sup>2</sup> C 总线控制寄存器 1 (ICCR1) .....	835
24.2.2	I <sup>2</sup> C 总线控制寄存器 2 (ICCR2) .....	838
24.2.3	I <sup>2</sup> C 总线模式寄存器 1 (ICMR1) .....	841
24.2.4	I <sup>2</sup> C 总线模式寄存器 2 (ICMR2) .....	842
24.2.5	I <sup>2</sup> C 总线模式寄存器 3 (ICMR3) .....	844
24.2.6	I <sup>2</sup> C 总线功能允许寄存器 (ICFER) .....	846
24.2.7	I <sup>2</sup> C 总线状态允许寄存器 (ICSER) .....	848
24.2.8	I <sup>2</sup> C 总线中断允许寄存器 (ICIER) .....	850

24.2.9	I <sup>2</sup> C 总线状态寄存器 1 (ICSR1)	852
24.2.10	I <sup>2</sup> C 总线状态寄存器 2 (ICSR2)	855
24.2.11	从属地址寄存器 Ly (SARLy) (y=0 ~ 2)	858
24.2.12	从属地址寄存器 Uy (SARUy) (y=0 ~ 2)	859
24.2.13	I <sup>2</sup> C 总线位速率低电平寄存器 (ICBRL)	860
24.2.14	I <sup>2</sup> C 总线位速率高电平寄存器 (ICBRH)	861
24.2.15	I <sup>2</sup> C 总线发送数据寄存器 (ICDRT)	863
24.2.16	I <sup>2</sup> C 总线接收数据寄存器 (ICDRR)	863
24.2.17	I <sup>2</sup> C 总线移位寄存器 (ICDRS)	863
24.2.18	超时内部计数器 (TMOCNT)	864
24.3	运行说明	865
24.3.1	通信数据的格式	865
24.3.2	初始设定	866
24.3.3	主控发送	867
24.3.4	主控接收	871
24.3.5	从属发送	874
24.3.6	从属接收	877
24.4	SCL 同步电路	879
24.5	SDA 输出延迟功能	880
24.6	数字噪声滤波器电路	881
24.7	地址匹配检测功能	882
24.7.1	从属地址匹配检测功能	882
24.7.2	全呼地址检测功能	884
24.7.3	设备 ID 地址检测功能	884
24.7.4	主机地址检测功能	886
24.8	SCL 的 Low 电平自动保持功能	887
24.8.1	发送数据的误发送防止功能	887
24.8.2	NACK 接收传送中止功能	888
24.8.3	接收数据漏取防止功能	888
24.9	仲裁失败检测功能	890
24.9.1	主控仲裁失败检测功能 (MALE 位)	890
24.9.2	NACK 发送仲裁失败检测功能 (NALE 位)	892
24.9.3	从属仲裁失败检测功能 (SALE 位)	893
24.10	开始条件、重新开始条件和停止条件的发行功能	894
24.10.1	开始条件的发行	894
24.10.2	重新开始条件的发行	894
24.10.3	停止条件的发行	895
24.11	总线意外停机	896
24.11.1	超时检测功能	896
24.11.2	SCL 时钟追加输出功能	898
24.11.3	RIIC/ 内部复位	898
24.12	SMBus 运行	899
24.12.1	SMBus 超时测量	899
24.12.2	数据包错误码 (PEC)	900
24.12.3	SMBus 主机通知协议 /Notify ARP master	900
24.13	中断源	901
24.14	复位状况	902
24.15	使用时的注意事项	903
24.15.1	模块停止功能的设定	903
24.15.2	输入缓冲控制寄存器的设定	903

25. CAN 模块 (CAN)	904
25.1 概要	904
25.2 寄存器说明	907
25.2.1 控制寄存器 (CTRL)	908
25.2.2 位配置寄存器 (BCR)	911
25.2.3 屏蔽寄存器 i (MKRi) (i=0 ~ 7)	913
25.2.4 FIFO 接收 ID 比较寄存器 0、1 (FIDCR0、FIDCR1)	914
25.2.5 屏蔽无效寄存器 (MKIVLR)	916
25.2.6 信箱寄存器 j (MBj) (j=0 ~ 31)	917
25.2.7 信箱中断允许寄存器 (MIER)	921
25.2.8 信息控制寄存器 j (MCTLj) (j=0 ~ 31)	923
25.2.9 接收 FIFO 控制寄存器 (RFCR)	926
25.2.10 接收 FIFO 指针控制寄存器 (RFPCR)	928
25.2.11 发送 FIFO 控制寄存器 (TFCR)	929
25.2.12 发送 FIFO 指针控制寄存器 (TFPCR)	931
25.2.13 状态寄存器 (STR)	932
25.2.14 信箱搜索模式寄存器 (MSMR)	935
25.2.15 信箱搜索状态寄存器 (MSSR)	936
25.2.16 通道搜索支持寄存器 (CSSR)	937
25.2.17 验收滤波器支持寄存器 (AFSR)	938
25.2.18 错误中断允许寄存器 (EIER)	939
25.2.19 错误中断源判断寄存器 (EIFR)	941
25.2.20 接收错误计数寄存器 (RECR)	943
25.2.21 发送错误计数寄存器 (TECR)	943
25.2.22 错误代码保存寄存器 (ECSR)	944
25.2.23 时戳寄存器 (TSR)	945
25.2.24 测试控制寄存器 (TCR)	946
25.3 运行模式	948
25.3.1 CAN 复位模式	949
25.3.2 CAN Halt 模式	950
25.3.3 CAN 睡眠模式	951
25.3.4 CAN 操作模式 (非总线断开状态)	951
25.3.5 CAN 操作模式 (总线断开状态)	952
25.4 CAN 通信速度的设定	953
25.4.1 CAN 时钟的设定	953
25.4.2 位时序的设定	953
25.4.3 位速率	954
25.5 信箱和屏蔽寄存器的结构	955
25.6 验收过滤功能和屏蔽功能	956
25.7 接收 / 发送	958
25.7.1 接收	959
25.7.2 发送	961
25.8 CAN 中断	962
26. 串行外围接口 (RSPI)	963
26.1 概要	963
26.2 寄存器说明	965
26.2.1 RSPI 控制寄存器 (SPCR)	966
26.2.2 RSPI 从属选择极性寄存器 (SSLP)	968
26.2.3 RSPI 引脚控制寄存器 (SPPCR)	969
26.2.4 RSPI 状态寄存器 (SPSR)	970
26.2.5 RSPI 数据寄存器 (SPDR)	973

26.2.6	RSPI 顺序控制寄存器 (SPSCR) .....	974
26.2.7	RSPI 顺序状态寄存器 (SPSSR) .....	975
26.2.8	RSPI 位速率寄存器 (SPBR) .....	976
26.2.9	RSPI 数据控制寄存器 (SPDCR) .....	977
26.2.10	RSPI 时钟延迟寄存器 (SPCKD) .....	980
26.2.11	RSPI 从属选择无效延迟寄存器 (SSLND) .....	981
26.2.12	RSPI 下次存取延迟寄存器 (SPND) .....	982
26.2.13	RSPI 控制寄存器 2 (SPCR2) .....	983
26.2.14	RSPI 命令寄存器 m (SPCMDm) (m=0 ~ 7) .....	984
26.3	运行说明 .....	987
26.3.1	RSPI 运行概要 .....	987
26.3.2	RSPI 引脚的控制 .....	988
26.3.3	RSPI 系统结构例子 .....	989
26.3.4	传送格式 .....	995
26.3.5	数据格式 .....	997
26.3.6	通信运行模式 .....	1005
26.3.7	发送缓冲器空中断 / 接收缓冲器满中断 .....	1007
26.3.8	错误检测 .....	1008
26.3.9	RSPI 的初始化 .....	1011
26.3.10	SPI 运行 .....	1012
26.3.11	时钟同步运行 .....	1022
26.3.12	错误处理 .....	1029
26.3.13	环回模式 .....	1031
26.3.14	奇偶校验位功能的自诊断 .....	1032
26.3.15	中断源 .....	1033
26.4	使用时的注意事项 .....	1033
26.4.1	主控模式中奇偶校验功能有效时的发送运行 .....	1033
27.	LIN 模块 (LIN) .....	1034
27.1	概要 .....	1034
27.2	寄存器说明 .....	1036
27.2.1	LIN 唤醒波特率选择寄存器 (LWBR) .....	1037
27.2.2	LIN 波特率预分频器 0 寄存器 (LBRP0) .....	1037
27.2.3	LIN 波特率预分频器 1 寄存器 (LBRP1) .....	1038
27.2.4	LIN 自测试控制寄存器 (LSTC) .....	1038
27.2.5	模式寄存器 (LOMD) .....	1039
27.2.6	间隔场设定寄存器 (LOBRK) .....	1040
27.2.7	间隔设定寄存器 (LOSPC) .....	1041
27.2.8	唤醒设定寄存器 (LOWUP) .....	1042
27.2.9	中断允许寄存器 (LOIE) .....	1043
27.2.10	错误检测允许寄存器 (LOEDE) .....	1044
27.2.11	控制寄存器 (LOC) .....	1045
27.2.12	发送控制寄存器 (LOTTC) .....	1046
27.2.13	模式状态寄存器 (LOMST) .....	1047
27.2.14	状态寄存器 (LOST) .....	1048
27.2.15	错误状态寄存器 (LOEST) .....	1049
27.2.16	响应场设定寄存器 (LORFC) .....	1050
27.2.17	ID 缓冲寄存器 (LOIDB) .....	1051
27.2.18	校验和缓冲寄存器 (LOCBR) .....	1052
27.2.19	数据 n 缓冲寄存器 (LODBn) (n=1 ~ 8) .....	1053

27.3	运行模式 .....	1054
27.3.1	LIN 复位模式 .....	1055
27.3.2	LIN 运行模式 .....	1055
27.3.3	LIN 唤醒模式 .....	1055
27.3.4	LIN 自测试模式 .....	1055
27.4	运行概要 .....	1056
27.4.1	报文头的发送 .....	1056
27.4.2	响应的发送 .....	1057
27.4.3	响应的接收 .....	1058
27.5	波特率发生器 .....	1059
27.6	数据的发送 / 接收 .....	1061
27.6.1	数据的发送 .....	1061
27.6.2	数据的接收 .....	1061
27.7	发送 / 接收数据的缓冲处理 .....	1062
27.7.1	LIN 帧的发送 .....	1062
27.7.2	LIN 帧的接收 .....	1063
27.8	唤醒的发送 / 接收 .....	1064
27.8.1	唤醒的发送运行 .....	1064
27.8.2	唤醒的接收运行 .....	1064
27.8.3	唤醒冲突 .....	1065
27.9	状态 .....	1066
27.10	错误状态 .....	1067
27.10.1	错误状态的种类 .....	1067
27.10.2	LIN 错误检测的对象时间区域 .....	1068
27.11	中断 .....	1069
27.12	LIN 自测试模式 .....	1070
27.12.1	向 LIN 自测试模式的转移 .....	1071
27.12.2	LIN 自测试模式中的发送 .....	1071
27.12.3	LIN 自测试中的接收 .....	1071
27.12.4	LIN 自测试模式的结束 .....	1071
28.	12 位 A/D 转换器 (S12ADA) .....	1072
28.1	概要 .....	1072
28.2	寄存器说明 .....	1077
28.2.1	A/D 数据寄存器 n (ADDRn) (n=0A、0B、1~3)、A/D 数据寄存器 Diag (ADRD) ...	1078
28.2.2	A/D 控制寄存器 (ADCSR) .....	1082
28.2.3	A/D 通道选择寄存器 (ADANS) .....	1084
28.2.4	A/D 控制扩展寄存器 (ADCER) .....	1087
28.2.5	A/D 开始触发选择寄存器 (ADSTRGR) .....	1089
28.2.6	A/D 可编程增益放大器寄存器 (ADPG) .....	1091
28.2.7	比较器运行模式选择寄存器 0 (ADCMPMD0) .....	1094
28.2.8	比较器运行模式选择寄存器 1 (ADCMPMD1) .....	1095
28.2.9	比较器的滤波器模式寄存器 0 (ADCMPNR0) .....	1097
28.2.10	比较器的滤波器模式寄存器 1 (ADCMPNR1) .....	1098
28.2.11	比较器检测标志寄存器 (ADCMPFR) .....	1099
28.2.12	比较器中断选择寄存器 (ADCMPSEL) .....	1100
28.2.13	A/D 采样状态寄存器 (ADSSTR) .....	1101
28.3	运行说明 .....	1102
28.3.1	单次模式 .....	1102
28.3.2	扫描转换运行的说明 .....	1103
28.3.3	模拟输入的采样和 A/D 转换时间 .....	1107
28.3.4	ADDRn、ADRD 寄存器的自动清除功能的使用例子 (n=0A、0B、1~3) .....	1108

28.3.5	双数据寄存器的运行（仅 ADDR0 寄存器）	1109
28.3.6	可编程增益放大器	1109
28.3.7	比较器	1110
28.3.8	通过外部触发开始的 A/D 转换	1111
28.3.9	通过外围模块的触发开始的 A/D 转换	1111
28.4	中断源和 DTC 传送请求	1111
28.4.1	A/D 转换的各 A/D 转换结束时的中断请求	1111
28.4.2	比较器检测时的中断请求	1111
28.5	使用时的注意事项	1112
28.5.1	模块停止功能的设定	1112
28.5.2	重新开始 A/D 转换时的注意事项	1112
28.5.3	停止 A/D 转换时的注意事项	1112
28.5.4	向低功耗状态转移时的注意事项	1112
28.5.5	有关容许信号源高阻抗	1113
28.5.6	对绝对精度的影响	1113
28.5.7	模拟电源引脚等的设定范围	1113
28.5.8	设计电路板时的注意事项	1114
28.5.9	噪声对策的注意事项	1114
29.	10 位 A/D 转换器（ADA）	1115
29.1	概要	1115
29.2	寄存器说明	1118
29.2.1	A/D 数据寄存器 n（ADDRn）（n=A ~ L）	1119
29.2.2	A/D 控制 / 状态寄存器（ADCSR）	1121
29.2.3	A/D 控制寄存器（ADCR）	1123
29.2.4	A/D 开始触发选择寄存器	1123
29.2.5	ADDRn 格式选择寄存器（ADDPn）（n=A ~ L）	1125
29.2.6	A/D 自诊断寄存器（ADDIAGR）	1126
29.2.7	A/D 采样状态寄存器（ADSSTR）	1127
29.3	运行说明	1128
29.3.1	单次模式	1128
29.3.2	扫描模式	1129
29.3.3	输入采样和 A/D 转换时间	1131
29.3.4	通过外部触发开始的 A/D 转换	1133
29.3.5	通过 MTU3、GPT 开始的 A/D 转换	1133
29.4	中断源	1134
29.5	A/D 转换精度的定义	1134
29.6	使用时的注意事项	1136
29.6.1	模块停止功能的设定	1136
29.6.2	停止 A/D 转换时的注意事项	1136
29.6.3	重新开始 A/D 转换时的注意事项	1136
29.6.4	向低功耗状态转移时的注意事项	1136
29.6.5	有关容许信号源阻抗	1137
29.6.6	对绝对精度的影响	1137
29.6.7	模拟电源引脚等的设定范围	1137
29.6.8	设计电路板时的注意事项	1138
29.6.9	噪声对策的注意事项	1138
30.	RAM	1139
30.1	概要	1139
30.2	运行说明	1139
30.2.1	低功耗功能	1139

31. ROM (保存代码的闪存)	1140
31.1 概要	1140
31.2 寄存器说明	1142
31.2.1 闪存模式寄存器 (FMODR)	1143
31.2.2 闪存存取状态寄存器 (FASTAT)	1144
31.2.3 闪存存取错误中断允许寄存器 (FAEINT)	1145
31.2.4 FCU RAM 允许寄存器 (FCURAME)	1146
31.2.5 闪存状态寄存器 0 (FSTATR0)	1147
31.2.6 闪存状态寄存器 1 (FSTATR1)	1149
31.2.7 闪存就绪中断允许寄存器 (FRDYIE)	1150
31.2.8 闪存 P/E 模式入口寄存器 (FENTRYR)	1151
31.2.9 闪存保护寄存器 (FPROTR)	1152
31.2.10 闪存复位寄存器 (FRESETR)	1153
31.2.11 FCU 命令寄存器 (FCMDR)	1154
31.2.12 FCU 处理转换寄存器 (FCPSR)	1155
31.2.13 闪存 P/E 状态寄存器 (FPESTAT)	1156
31.2.14 外围时钟通知寄存器 (PCKAR)	1157
31.2.15 闪存编程 / 擦除保护寄存器 (FWEPROR)	1158
31.3 ROM 的存储器 MAT 结构	1159
31.4 块结构	1159
31.5 ROM 相关的运行模式	1160
31.6 ROM 的编程 / 擦除	1162
31.6.1 FCU 的模式	1162
31.6.2 FCU 命令一览表	1164
31.6.3 FCU 的模式和命令的关系	1166
31.6.4 FCU 命令的使用方法	1167
31.7 挂起	1183
31.7.1 编程时的挂起	1183
31.7.2 擦除时的挂起 (挂起优先模式)	1184
31.7.3 擦除时的挂起 (擦除优先模式)	1185
31.8 保护	1186
31.8.1 软件保护	1186
31.8.2 错误保护	1186
31.9 引导模式	1188
31.9.1 系统结构	1188
31.9.2 ID 码保护	1188
31.9.3 引导模式的状态转移	1190
31.9.4 位速率的自动调整	1192
31.9.5 查询设定主机命令的等待状态	1193
31.9.6 ID 码等待状态	1203
31.9.7 编程 / 擦除主机命令等待状态	1204
31.10 on-chip 调试器的 ID 码保护	1211
31.11 ROM 码保护	1211
31.12 使用时的注意事项	1212
32. 数据闪存 (保存数据的数据闪存)	1213
32.1 概要	1213
32.2 寄存器说明	1215
32.2.1 闪存模式寄存器 (FMODR)	1216
32.2.2 闪存存取状态寄存器 (FASTAT)	1217
32.2.3 闪存存取错误中断允许寄存器 (FAEINT)	1219
32.2.4 数据闪存读允许寄存器 0 (DFLRE0)	1220



32.2.5	数据闪存读允许寄存器 1 (DFLRE1)	1221
32.2.6	数据闪存编程 / 擦除允许寄存器 0 (DFLWE0)	1222
32.2.7	数据闪存编程 / 擦除允许寄存器 1 (DFLWE1)	1223
32.2.8	闪存 P/E 模式入口寄存器 (FENTRYR)	1224
32.2.9	数据闪存空白检查控制寄存器 (DFLBCCNT)	1225
32.2.10	数据闪存空白检查状态寄存器 (DFLBCSTAT)	1225
32.3	数据闪存的存储器 MAT 结构	1226
32.4	块结构	1226
32.5	数据闪存相关的运行模式	1227
32.6	数据闪存的编程 / 擦除	1228
32.6.1	FCU 的模式	1228
32.6.2	FCU 命令一览表	1230
32.6.3	FCU 的模式和命令的关系	1231
32.6.4	FCU 命令的使用方法	1232
32.7	保护	1236
32.7.1	软件保护	1236
32.7.2	错误保护	1236
32.8	引导模式	1237
32.8.1	查询设定主机命令	1237
32.8.2	编程 / 擦除主机命令	1239
32.9	使用时的注意事项	1240
<b>33.</b>	<b>电特性</b>	<b>1241</b>
33.1	绝对最大额定值	1241
33.2	DC 特性	1242
33.3	AC 特性	1245
33.3.1	时钟的时序	1246
33.3.2	控制信号的时序	1249
33.3.3	内部外围模块的时序	1250
33.4	A/D 转换特性	1258
33.5	上电复位电路和电压检测电路的特性	1262
33.6	振荡停止检测时序	1264
33.7	ROM (保存代码的闪存) 特性	1265
33.8	数据闪存 (保存数据的闪存) 特性	1266
<b>附录</b>		<b>1268</b>
附录 1.	各运行模式中的端口状态	1268
附录 2.	封装尺寸图	1269

## RX62T 群

瑞萨 32 位单片机

R01UH0034CJ0131  
Rev.1.31  
2012.06.28

100MHz、32 位 RX MCU、内部 FPU、165DMIPS、  
从 12 位 ADC（3 个 S/H 电路、双数据寄存器、放大器、比较器）的 2 个单元、  
10 位 ADC 的 1 个单元、共 ADC 的 3 个单元中选择 7ch 进行同时采样、  
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  
165DMIPS 性能（以 100MHz 运行时）
- 32 位单精度浮点（符合 IEEE754 标准）
- 32×32→64 位运算结果（1 条指令）的累加器
- 32×32 位乘法、除法器（乘法运算指令为 1 个 CPU 时钟）
- 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：大幅缩短编码
- JTAG 调试功能和高速跟踪功能

#### ■工作电压

- 单 3.3V、单 5V、3.3V 产品能使用 5V 模拟电源

#### ■低功耗功能

- 4 种低功耗模式

#### ■内部主闪存（无等待）

- 以 100MHz 运行、读周期为 10n sec
- 当 CPU 全速读时，无等待
- 64K/128K/256K 字节容量
- 用于指令、操作数
- SCI、JTAG 的用户编程

#### ■内部数据闪存

- 最大 32K 字节（能进行 30K 次擦除）
- 不对 CPU 外加负荷的编程/擦除

#### ■内部 SRAM（无等待）

- 8K/16K 字节 SRAM
- 用于操作数、指令

#### ■DMA

- DTC：能在 1 个单元进行多个通道的传送

#### ■复位和电源电压控制

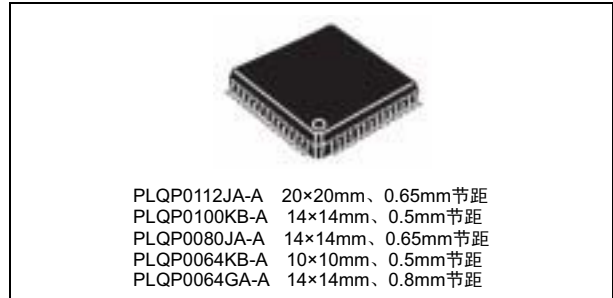
- 随时内部上电复位（POR）
- 能设定低电压检测功能（LVD）

#### ■时钟功能

- 外部晶体振荡器、内部 PLL 支持 8MHz～12.5MHz
- 用于 IWDG 的内部 125kHz LOCO 时钟
- 主时钟振荡停止检测功能（支持 IEC60730）

#### ■内置独立看门狗定时器（支持 IEC60730）

- 以 125kHz LOCO 时钟运行
- 无法通过软件停止的强 WDT



#### ■内置最多 7 个通信接口

- CAN（符合 ISO11898-1 标准）内置 32 个信箱
- SCI（3ch）  
异步模式（带噪声消除功能）/ 时钟同步模式 / 智能卡接口模式
- I<sup>2</sup>C 总线接口支持 SMBus（1ch）
- RSPI（1ch）
- LIN（1ch）

#### ■最多 16 个 16 位定时器

- 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 A/D 转换器 3 个单元共 20ch

- 能从 3 个单元中选择 7ch 进行同时采样
- 自诊断功能（支持 IEC60730）
- 12 位 ADC（2 个单元）：3 个采样 & 保持电路、双数据寄存器、放大器、比较器（8ch）
- 10 位 ADC（1 个单元）（12ch）

#### ■CRC（Cyclic Redundancy Check）运算器

- 监视通信数据（支持 IEC60730）
- 监视存储数据（支持 IEC60730）

#### ■最多 61 个输入 / 输出端口、最多 21 个输入端口

- PORT 寄存器：监视输出端口（支持 IEC60730）

#### ■工作环境温度

- -40°C～+85°C

## 1. 概要

### 1.1 概要

规格概要和各产品功能对应表分别如表 1.1 和表 1.2 所示。

表 1.1 规格概要 (1/5)

分类	模块 / 功能	说明
CPU	中央处理器	<ul style="list-style-type: none"> <li>最大工作频率：100MHz</li> <li>32 位 RX CPU</li> <li>指令的最短执行时间：1 条指令 1 个时钟</li> <li>地址空间：4G 字节、线性地址</li> <li>寄存器               <ul style="list-style-type: none"> <li>通用寄存器：32 位 ×16 个</li> <li>控制寄存器：32 位 ×9 个</li> <li>累加器：64 位 ×1 个</li> </ul> </li> <li>基本指令：73 种</li> <li>浮点运算指令：8 种</li> <li>DSP 功能指令：9 种</li> <li>寻址方式：10 种</li> <li>数据排列               <ul style="list-style-type: none"> <li>指令：小端法</li> <li>数据：可选择小端法或者大端法</li> </ul> </li> <li>32 位乘法器：32 位 ×32 位 →64 位</li> <li>除法器：32 位 ÷32 位 →32 位</li> <li>桶式移位器：32 位</li> <li>存储器保护单元 (MPU)</li> </ul>
	FPU	<ul style="list-style-type: none"> <li>单精度浮点数 (32 位)</li> <li>符合 IEEE754 规格的数据类型和异常</li> </ul>
存储器	ROM	<ul style="list-style-type: none"> <li>ROM 容量：最多 256K 字节</li> <li>2 种板上编程模式               <ul style="list-style-type: none"> <li>引导模式 (能使用 SCI 改写用户 MAT)</li> <li>用户编程模式</li> </ul> </li> <li>板外编程               <ul style="list-style-type: none"> <li>能使用 PROM 编程器改写用户模式</li> </ul> </li> </ul>
	RAM	RAM 容量：最多 16K 字节
	数据闪存	<ul style="list-style-type: none"> <li>数据 ROM 容量：最多 32K 字节</li> <li>支持 BGO (后台操作)</li> </ul>
MCU 运行模式		单芯片模式
时钟	时钟发生电路	<ul style="list-style-type: none"> <li>1 个电路：主时钟振荡电路</li> <li>内部振荡：IWDG 专用低速内部振荡器</li> <li>由 PLL 频率合成器和分频电路构成，可选择工作频率。</li> <li>振荡停止检测：有</li> <li>能分别设定系统时钟 (ICLK) 和外围模块时钟 (PCLK)。</li> <li>CPU、总线主控器等系统设备、MTU3 以及 GPT 与 ICLK 同步：8 ~ 100MHz</li> <li>外围模块与 PCLK 同步：8 ~ 50MHz</li> </ul>
复位		引脚复位、上电复位 (接通电源时自动上电复位)、电压监视复位、看门狗定时器复位、独立看门狗定时器复位、深度软件待机复位
电压检测电路 (LVD)		如果 VCC 低于电压检测电平 (Vdet)，就产生内部复位或者内部中断。
低功耗	低功耗功能	<ul style="list-style-type: none"> <li>模块停止功能</li> <li>4 种低功耗状态               <ul style="list-style-type: none"> <li>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</li> </ul> </li> </ul>

表 1.1 规格概要 (2/5)

分类	模块 / 功能	说明
中断	中断控制器 (ICU)	<ul style="list-style-type: none"> <li>• 外围功能中断: 101 个中断源</li> <li>• 外部中断: 9 个中断源 (NMI 引脚、IRQ0 ~ IRQ7 引脚)</li> <li>• 非屏蔽中断: 3 个中断源 (NMI 引脚、振荡停止检测中断、电压监视中断)</li> <li>• 能设定 16 个中断优先级。</li> </ul>
数据传送	数据传送控制器 (DTC)	<ul style="list-style-type: none"> <li>• 传送模式: 正常传送模式、重复传送模式、块传送模式</li> <li>• 启动源: 软件触发、外部中断、外围功能中断</li> </ul>
I/O 端口	可编程输入 / 输出端口	112 引脚 LQFP/100 引脚 LQFP/80 引脚 LQFP/64 引脚 LQFP 的 I/O 端口 <ul style="list-style-type: none"> <li>• 输入 / 输出: 61/55/44/37</li> <li>• 输入: 21/21/13/9</li> <li>• 漏极开路输出: 2/2/2/2 (I<sup>2</sup>C 总线接口引脚)</li> <li>• 大电流输出: 12/12/6/6 (0) (MTU3 引脚、GPT 引脚) (在 64 引脚 5V 版中无大电流输出)</li> <li>• 能随时读取引脚的状态</li> </ul>
定时器	多功能定时器脉冲单元 3 (MTU3)	<ul style="list-style-type: none"> <li>• 16 位 × 8 个通道</li> <li>• 最多能输入 / 输出 24 个脉冲以及输入 3 个脉冲</li> <li>• 各通道可选择 6 ~ 8 种计数时钟 (ICLK/1、ICLK/4、ICLK/16、ICLK/64、ICLK/256、ICLK/1024、MTCLKA、MTCLKB、MTCLKC、MTCLKD) (通道 5 可选择 4 种)</li> <li>• 24 个输出比较寄存器兼输入捕捉寄存器</li> <li>• 计数器清除运行 (能通过比较匹配 / 输入捕捉同时清除)</li> <li>• 同时写多个定时器的计数器 (TCNT)</li> <li>• 通过计数器的同步运行进行各寄存器的同步输入 / 输出</li> <li>• 缓冲运行</li> <li>• 级联运行</li> <li>• 38 种中断源</li> <li>• 自动传送寄存器数据</li> <li>• 脉冲输出模式 交替 / PWM / 互补 PWM / 复位同步 PWM</li> <li>• 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</li> <li>• 复位同步 PWM 模式 三相输出任意占空比的正相、反相 PWM 波形</li> <li>• 相位计数模式</li> <li>• 用于补偿死区时间的计数功能</li> <li>• 能生成 A/D 转换器的转换开始触发</li> <li>• A/D 转换开始减少功能</li> </ul>
	端口输出允许 3 (POE3)	<ul style="list-style-type: none"> <li>• 控制 MTU3 和 GPT 波形输出引脚的高阻抗 通过 POE0、POE4、POE8、POE10 和 POE11 这 5 个输入引脚启动</li> <li>• 通过输出短路检测 (检测到大电流引脚的输出同时变为有效电平) 启动</li> <li>• 通过 12 位 A/D 转换器模拟输入的比较器检测启动</li> <li>• 通过振荡停止检测启动</li> <li>• 通过软件启动</li> <li>• 在各 POE 输入和比较器检测时, 能选择让哪个输出引脚变为高阻抗。</li> </ul>

表 1.1 规格概要 (3 / 5)

分类	模块 / 功能	说明
定时器	通用 PWM 定时器 (GPT)	<ul style="list-style-type: none"> <li>• 16 位 ×4 个通道</li> <li>• 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)</li> <li>• 能给各通道选择独立的时钟源</li> <li>• 每个通道有 2 个输入 / 输出引脚</li> <li>• 每个通道有 2 个用于输出比较 / 输入捕捉的寄存器</li> <li>• 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。</li> <li>• 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。</li> <li>• 给各通道装载用于帧周期的寄存器 (能通过上溢 / 下溢中断)</li> <li>• 能同步运行各计数器</li> <li>• 同步运行模式 (支持同时或者以任意的时序移动相位)</li> <li>• 能在 PWM 运行时生成死区时间</li> <li>• 组合 3 个计数器, 能自动生成带死区时间的三相 PWM 波形</li> <li>• 能通过外部 / 内部触发开始、清除或者停止计数</li> <li>• 内部触发源有比较器检测、软件和比较匹配</li> <li>• 能通过分频系统时钟 (ICLK) 的计数时钟来测量被分频的 IWDTC 专用低速内部振荡器时钟的边沿 (振荡异常检测)。</li> </ul>
	比较匹配定时器 (CMT)	<ul style="list-style-type: none"> <li>• (16 位 ×2 个通道) ×2 个单元</li> <li>• 可选择 4 种时钟 (PCLK/8、PCLK/32、PCLK/128、PCLK/512)</li> </ul>
	看门狗定时器 (WDT)	<ul style="list-style-type: none"> <li>• 8 位 ×1 个通道</li> <li>• 能选择 8 种计数时钟 (PCLK/4、PCLK/64、PCLK/128、PCLK/512、PCLK/2048、PCLK/8192、PCLK/32768、PCLK/131072)</li> <li>• 能转换看门狗定时器模式和间隔定时器模式</li> </ul>
	独立看门狗定时器 (IWDTC)	<ul style="list-style-type: none"> <li>• 14 位 ×1 个通道</li> <li>• 计数时钟: IWDTC 专用低速内部振荡器</li> </ul>
通信功能	串行通信接口 (SCIb)	<ul style="list-style-type: none"> <li>• 3 个通道</li> <li>• 串行通信方式: 异步 / 时钟同步 / 智能卡接口</li> <li>• 多处理器通信功能</li> <li>• 能通过内部波特率发生器选择任意的位速率</li> <li>• 能选择 LSB first 或者 MSB first</li> <li>• 噪声消除功能 (仅在异步时有效)</li> </ul>
	I <sup>2</sup> C 总线接口 (RIIC)	<ul style="list-style-type: none"> <li>• 1 个通道</li> <li>• 通信格式 I<sup>2</sup>C 总线格式 / SMBus 格式 能选择主控 / 从属</li> </ul>
	CAN 模块 (CAN) (选项)	<ul style="list-style-type: none"> <li>• 1 个通道</li> <li>• 32 个信箱</li> </ul>
	串行外围接口 (RSPI)	<ul style="list-style-type: none"> <li>• 1 个单元</li> <li>• RSPI 传送功能 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) / 时钟同步运行 (3 线式) 进行串行通信。 能在主控模式 / 从属模式中进行串行通信。</li> <li>• 数据格式 能转换 MSB first 和 LSB first。 能将传送位长更改为 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 / 接收最多传送 4 帧 (1 帧最多为 32 位)。</li> <li>• 缓冲器结构 发送 / 接收缓冲器为双缓冲结构。</li> </ul>
	LIN 模块 (LIN)	<ul style="list-style-type: none"> <li>• 1 个通道 (LIN 主控器)</li> <li>• 能支持的 LIN 协议版本 1.3、2.0、2.1</li> </ul>

表 1.1 规格概要 (4 / 5)

分类	模块 / 功能	说明
A/D 转换器	12 位 A/D 转换器 (S12ADA)	<ul style="list-style-type: none"> <li>• 12 位 (4 个通道 × 2 个单元)</li> <li>• 分辨率: 12 位</li> <li>• 转换时间 每个通道 1.0μs (当 A/D 转换时钟 ADCLK=50MHz、AVCC0=4.0 ~ 5.5V 时) 每个通道 2.0μs (当 A/D 转换时钟 ADCLK=25MHz、AVCC0=3.0 ~ 3.6V 时)</li> <li>• 2 种运行模式 单次模式、扫描模式 (单次扫描模式、连续扫描模式、2 个通道扫描模式 (在 A/D 单元将输入分为 2 组, 能分别设定启动源))</li> <li>• 带采样 &amp; 保持功能 装载了单元通用的采样 &amp; 保持电路 还装载了专用的采样 &amp; 保持电路 (3 个通道 / 1 个单元)</li> <li>• 给各输入引脚装载了 A/D 转换寄存器</li> <li>• 各单元仅 1 个模拟输入 (AN000/AN100) 设置有 2 个转换结果寄存器</li> <li>• 3 种 A/D 转换开始方法 软件触发、定时器 (MTU3 和 GPT) 的触发、外部触发</li> <li>• 支持 8 位、10 位精度输出 能选择输出转换结果的 2 位或者 4 位右移</li> <li>• 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)</li> <li>• 通过可编程增益放大器放大输入信号的功能 (3 个通道 / 1 个单元) 放大率: 2.0 倍、2.5 倍、3.077 倍、3.636 倍、4.0 倍、4.444 倍、5.0 倍、5.714 倍、6.667 倍、10.0 倍、13.333 倍 (共 11 级)</li> <li>• 窗口比较器功能 (3 个通道 / 1 个单元)</li> </ul>
	10 位 A/D 转换器 (ADA)	<ul style="list-style-type: none"> <li>• 10 位 (12 个通道 × 1 个单元)</li> <li>• 分辨率: 10 位</li> <li>• 转换时间 每个通道 1.0μs (当 A/D 转换时钟 ADCLK=50MHz、AVCC=4.0 ~ 5.5V 时) 每个通道 2.0μs (当 A/D 转换时钟 ADCLK=25MHz、AVCC=3.0 ~ 3.6V 时)</li> <li>• 2 种运行模式 单次模式、扫描模式 (1 个周期扫描模式、连续扫描模式)</li> <li>• 带采样 &amp; 保持功能 装载了单元通用的采样 &amp; 保持电路</li> <li>• 给各输入引脚装载了 A/D 转换寄存器</li> <li>• 3 种 A/D 转换开始方法 软件触发、定时器 (MTU3 和 GPT) 的触发、外部触发</li> <li>• 支持 8 位精度输出 能选择输出转换结果的 2 位右移</li> <li>• 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 (AVSS、VREF×1/2、VREF)</li> </ul>
CRC 运算器 (CRC)	<ul style="list-style-type: none"> <li>• 对应以 8 位为单位的任意数据长度, 生成 CRC 码。</li> <li>• 能从 3 个多项式中选择: <math>X^8+X^2+X+1</math>、<math>X^{16}+X^{15}+X^2+1</math>、<math>X^{16}+X^{12}+X^5+1</math></li> <li>• 能选择生成用于 LSB first 通信的 CRC 码或者生成用于 MSB first 通信的 CRC 码。</li> </ul>	
工作频率	<p>ICLK: 8 ~ 100MHz PCLK: 8 ~ 50MHz</p>	

表 1.1 规格概要 (5 / 5)

分类	模块 / 功能	说明
电源电压		<ul style="list-style-type: none"> <li>• 3V 版 VCC=PLLVC=2.7 ~ 3.6V AVCC0=AVCC=3.0 ~ 3.6V 或者 4.0 ~ 5.5V VREFH0=3.0 ~ AVCC0 或者 4.0 ~ AVCC0 VREF=3.0 ~ AVCC 或者 4.0 ~ AVCC</li> <li>• 5V 版 VCC=PLLVC=4.0 ~ 5.5V AVCC0=AVCC=4.0 ~ 5.5V VREFH0=4.0 ~ AVCC0 VREF=4.0 ~ AVCC</li> </ul>
工作环境温度		-40 ~ +85°C
封装		112 引脚 LQFP (PLQP0112JA-A (20×20-0.65mm 节距)) 100 引脚 LQFP (PLQP0100KB-A (14×14-0.5mm 节距)) 80 引脚 LQFP (PLQP0080JA-A (14×14-0.65mm 节距)) 64 引脚 LQFP (PLQP0064KB-A (10×10-0.5mm 节距)) 64 引脚 LQFP (PLQP0064GA-A (14×14-0.8mm 节距))

表 1.2 RX62T 群的功能比较表

功能 引脚数		RX62T 群			
		112 引脚	100 引脚	80 引脚	64 引脚
数据传送	数据传送控制器 (DTC)	○			
中断控制器 (ICU)	NMI 引脚输入	○			
	IRQ 引脚输入	○ (8 个)		○ (4 个)	
定时器	多功能定时器脉冲单元 3 (MTU3)	○		○ (注)	
	通用 PWM 定时器 (GPT)	○		○ (注)	
	端口输出允许 3 (POE3)	○ (5 个 POE 引脚)		○ (3 个 POE 引脚)	
	比较匹配定时器 (CMT)	○			
	看门狗定时器 (WDT)	○			
	独立看门狗定时器 (IWDT)	○			
	通信功能	串行通信接口 (SCI)	○		
I <sup>2</sup> C 总线接口 (RIIC)		○			
CAN 模块 (CAN) (选项)		○			
LIN 模块 (LIN)		○			
串行外围接口 (RSPI)		○			
12 位 AD 转换器 (S12ADA)		○ (4 个通道 × 2 个单元)			
	3 个通道同时采样功能	○ (2 个单元)			
	可编程增益放大器	○ (3 个通道 × 2 个单元)			
	窗口比较器	○ (3 个通道 × 2 个单元)			
10 位 A/D 转换器 (ADA)	○ (12 个通道)		○ (4 个通道)	—	
CRC 运算器 (CRC)		○			
I/O 端口	输入 / 输出	61 个	55 个	44 个	37 个
	输入	21 个	21 个	13 个	9 个
封装		LQFP2020 (0.65mm 节距)	LQFP1414 (0.5mm 节距)	LQFP1414 (0.65mm 节距)	LQFP1010 (0.5mm 节距) LQFP1414 (0.8mm 节距)

○: 有 —: 无

注. 有关 MTU3 和 GPT, 引脚数因封装而不同。详细内容请参照各功能引脚一览表。另外, CAN 模块为选项。详细内容请参照表 1.3。



## 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	型号	封装			工作频率 (max)	ROM 容量	RAM 容量	数据闪存	工作电压		CAN 功能													
		封装外形	引脚节距	封装代码					VCC/ PLLVCC	AVCC/ AVCC0														
RX62T	R5F562TAADFH	LQFP2020-112	0.65mm	PLQP0112JA-A	100MHz	256K 字节	16K 字节	32K 字节	4.0 ~ 5.5V	4.0 ~ 5.5V	有													
	R5F562TAADFP	LQFP1414-100	0.5mm	PLQP0100KB-A																				
	R5F562TAADFF	LQFP1414-80	0.65mm	PLQP0080JA-A																				
	R5F562TAADFM	LQFP1010-64	0.5mm	PLQP0064KB-A																				
	R5F562TAADFK	LQFP1414-64	0.8mm	PLQP0064GA-A																				
	R5F562T7ADFH	LQFP2020-112	0.65mm	PLQP0112JA-A								128K 字节	8K 字节	8K 字节										
	R5F562T7ADFP	LQFP1414-100	0.5mm	PLQP0100KB-A																				
	R5F562T7ADFF	LQFP1414-80	0.65mm	PLQP0080JA-A																				
	R5F562T7ADFM	LQFP1010-64	0.5mm	PLQP0064KB-A																				
	R5F562T7ADFK	LQFP1414-64	0.8mm	PLQP0064GA-A																				
	R5F562T6ADFF	LQFP1414-80	0.65mm	PLQP0080JA-A														64K 字节	8K 字节					
	R5F562T6ADFM	LQFP1010-64	0.5mm	PLQP0064KB-A																				
	R5F562T6ADFK	LQFP1414-64	0.8mm	PLQP0064GA-A																				
	R5F562TABDFH	LQFP2020-112	0.65mm	PLQP0112JA-A		256K 字节	16K 字节	32K 字节	2.7 ~ 3.6V	3.0 ~ 3.6V 或者 4.0 ~ 5.5V														
	R5F562TABDFP	LQFP1414-100	0.5mm	PLQP0100KB-A																				
	R5F562TABDFF	LQFP1414-80	0.65mm	PLQP0080JA-A																				
	R5F562TABDFM	LQFP1010-64	0.5mm	PLQP0064KB-A																				
	R5F562TABDFK	LQFP1414-64	0.8mm	PLQP0064GA-A																				
	R5F562T7BDFH	LQFP2020-112	0.65mm	PLQP0112JA-A								128K 字节	8K 字节	8K 字节										
	R5F562T7BDFP	LQFP1414-100	0.5mm	PLQP0100KB-A																				
	R5F562T7BDFP	LQFP1414-80	0.65mm	PLQP0080JA-A																				
	R5F562T7BDFM	LQFP1010-64	0.5mm	PLQP0064KB-A																				
	R5F562T7BDFK	LQFP1414-64	0.8mm	PLQP0064GA-A																				
	R5F562T6BDFP	LQFP1414-80	0.65mm	PLQP0080JA-A														64K 字节	8K 字节					
	R5F562T6BDFM	LQFP1010-64	0.5mm	PLQP0064KB-A																				
	R5F562T6BDFK	LQFP1414-64	0.8mm	PLQP0064GA-A																				
	R5F562TADDFH	LQFP2020-112	0.65mm	PLQP0112JA-A		256K 字节	16K 字节	32K 字节	4.0 ~ 5.5V	4.0 ~ 5.5V														无
	R5F562TADDFP	LQFP1414-100	0.5mm	PLQP0100KB-A																				
	R5F562TADDFP	LQFP1414-80	0.65mm	PLQP0080JA-A																				
	R5F562TADDFM	LQFP1010-64	0.5mm	PLQP0064KB-A																				
	R5F562TADDFK	LQFP1414-64	0.8mm	PLQP0064GA-A																				
	R5F562T7DDFH	LQFP2020-112	0.65mm	PLQP0112JA-A								128K 字节	8K 字节	8K 字节										
	R5F562T7DDFP	LQFP1414-100	0.5mm	PLQP0100KB-A																				
	R5F562T7DDFF	LQFP1414-80	0.65mm	PLQP0080JA-A																				
	R5F562T7DDFM	LQFP1010-64	0.5mm	PLQP0064KB-A																				
	R5F562T7DDFK	LQFP1414-64	0.8mm	PLQP0064GA-A																				
	R5F562T6DDFF	LQFP1414-80	0.65mm	PLQP0080JA-A														64K 字节	8K 字节					
	R5F562T6DDFM	LQFP1010-64	0.5mm	PLQP0064KB-A																				
	R5F562T6DDFK	LQFP1414-64	0.8mm	PLQP0064GA-A																				
	R5F562TAEDFH	LQFP2020-112	0.65mm	PLQP0112JA-A		256K 字节	16K 字节	32K 字节	2.7 ~ 3.6V	3.0 ~ 3.6V 或者 4.0 ~ 5.5V														
	R5F562TAEDFP	LQFP1414-100	0.5mm	PLQP0100KB-A																				
	R5F562TAEDFF	LQFP1414-80	0.65mm	PLQP0080JA-A																				
	R5F562TAEDFM	LQFP1010-64	0.5mm	PLQP0064KB-A																				
	R5F562TAEDFK	LQFP1414-64	0.8mm	PLQP0064GA-A																				
	R5F562T7EDFH	LQFP2020-112	0.65mm	PLQP0112JA-A								128K 字节	8K 字节	8K 字节										
	R5F562T7EDFP	LQFP1414-100	0.5mm	PLQP0100KB-A																				
	R5F562T7EDFF	LQFP1414-80	0.65mm	PLQP0080JA-A																				
	R5F562T7EDFM	LQFP1010-64	0.5mm	PLQP0064KB-A																				
R5F562T7EDFK	LQFP1414-64	0.8mm	PLQP0064GA-A																					
R5F562T6EDFF	LQFP1414-80	0.65mm	PLQP0080JA-A	64K 字节	8K 字节																			
R5F562T6EDFM	LQFP1010-64	0.5mm	PLQP0064KB-A																					
R5F562T6EDFK	LQFP1414-64	0.8mm	PLQP0064GA-A																					

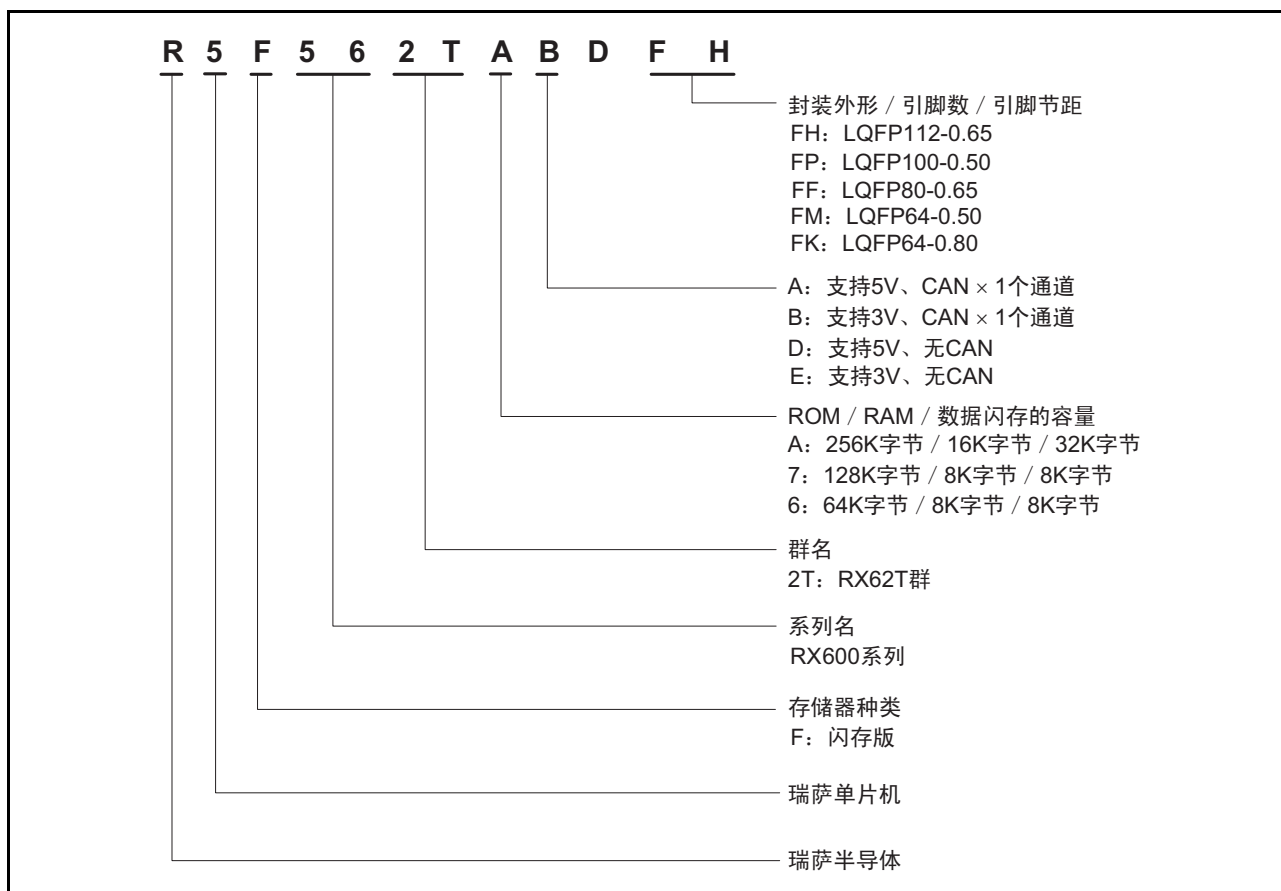


图 1.1 产品型号、存储容量和封装

1.3 框图

框图如图 1.2 所示。

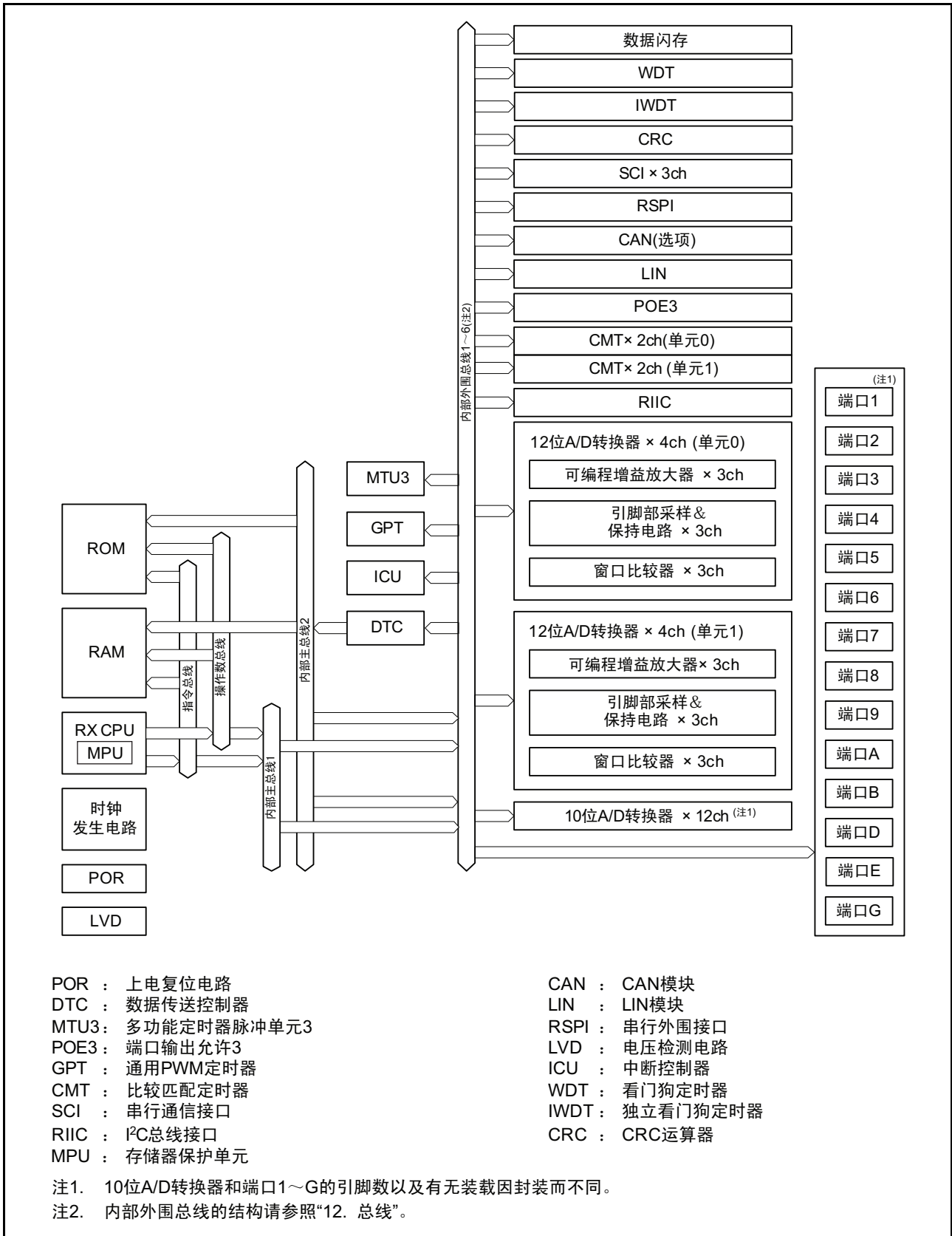


图 1.2 框图

1.4 引脚排列图

引脚排列图如图 1.3 ~ 图 1.6 所示，各功能引脚一览表如表 1.4 ~ 表 1.7 所示。

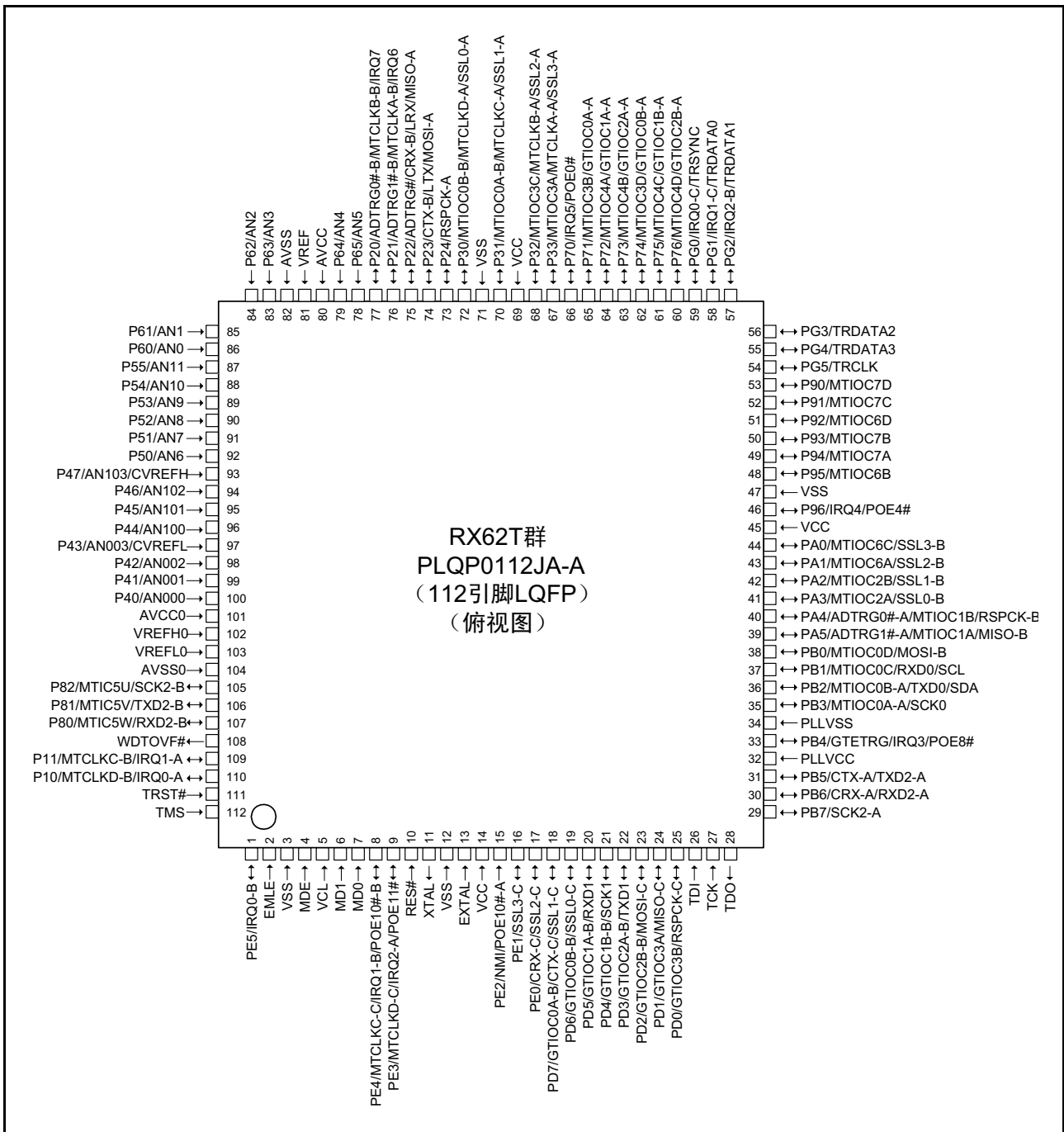


图 1.3 122 引脚 LQFP 的引脚排列图

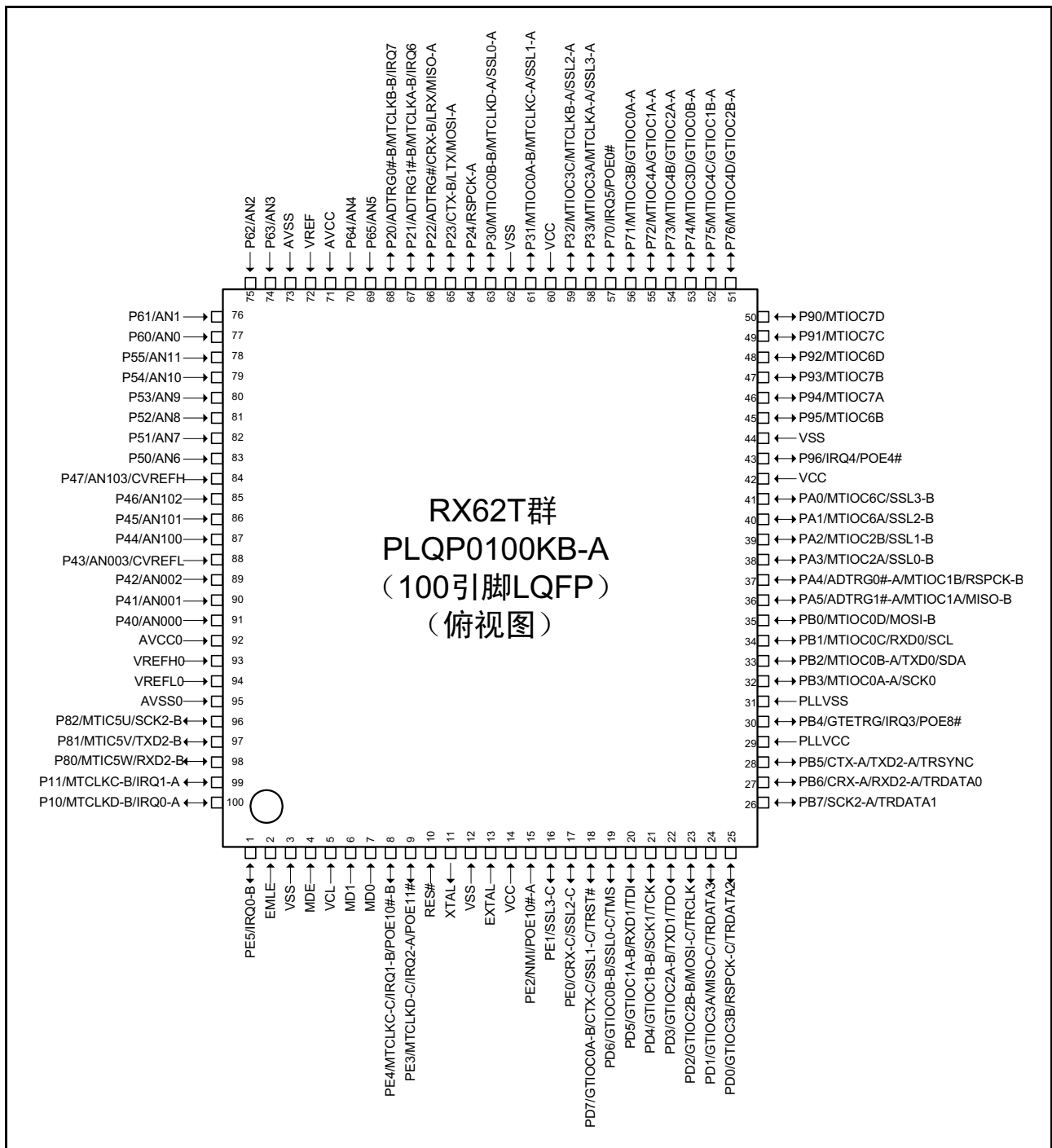


图 1.4 100 引脚 LQFP 的引脚排列图

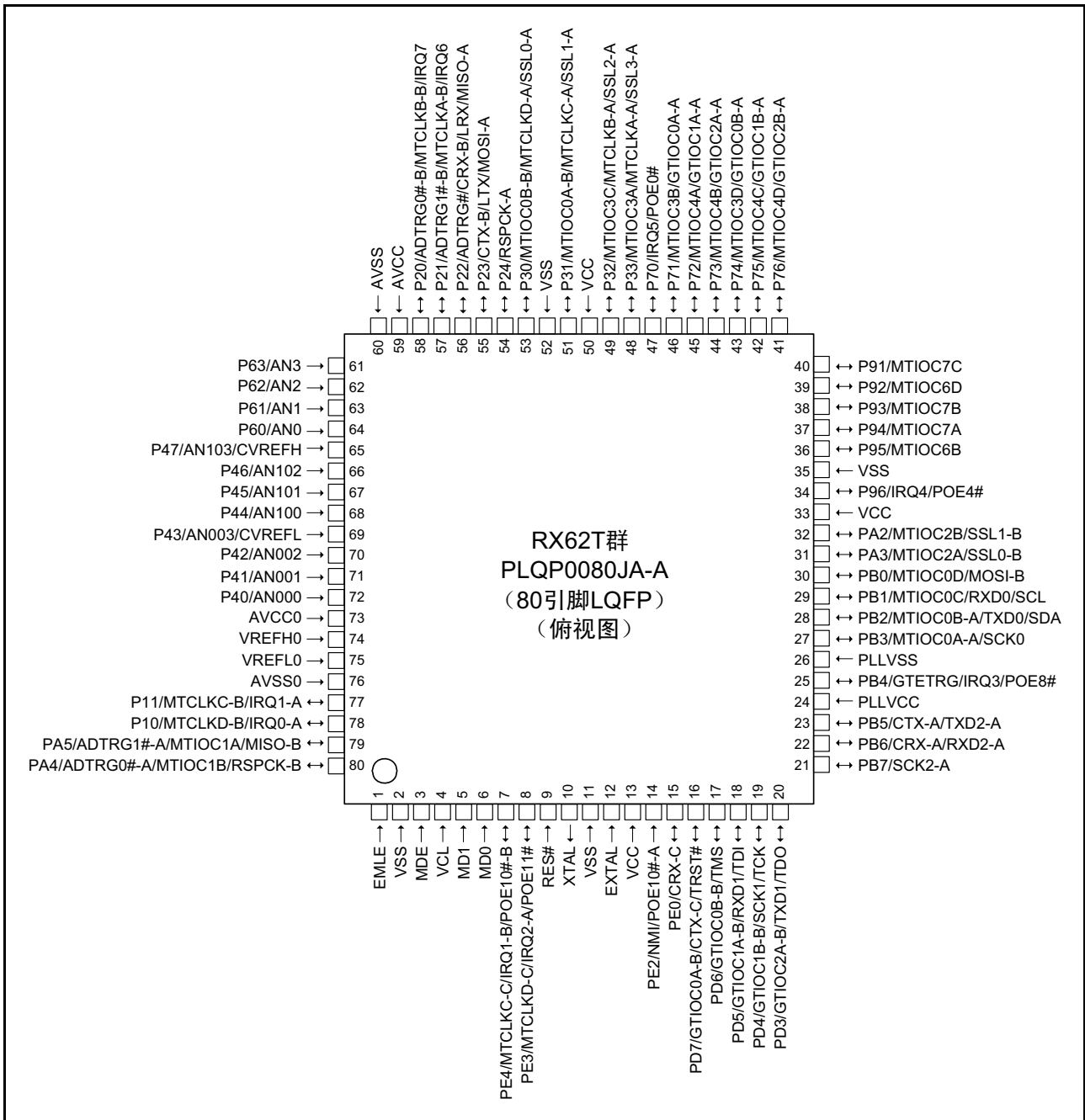


图 1.5 80 引脚 LQFP 的引脚排列图

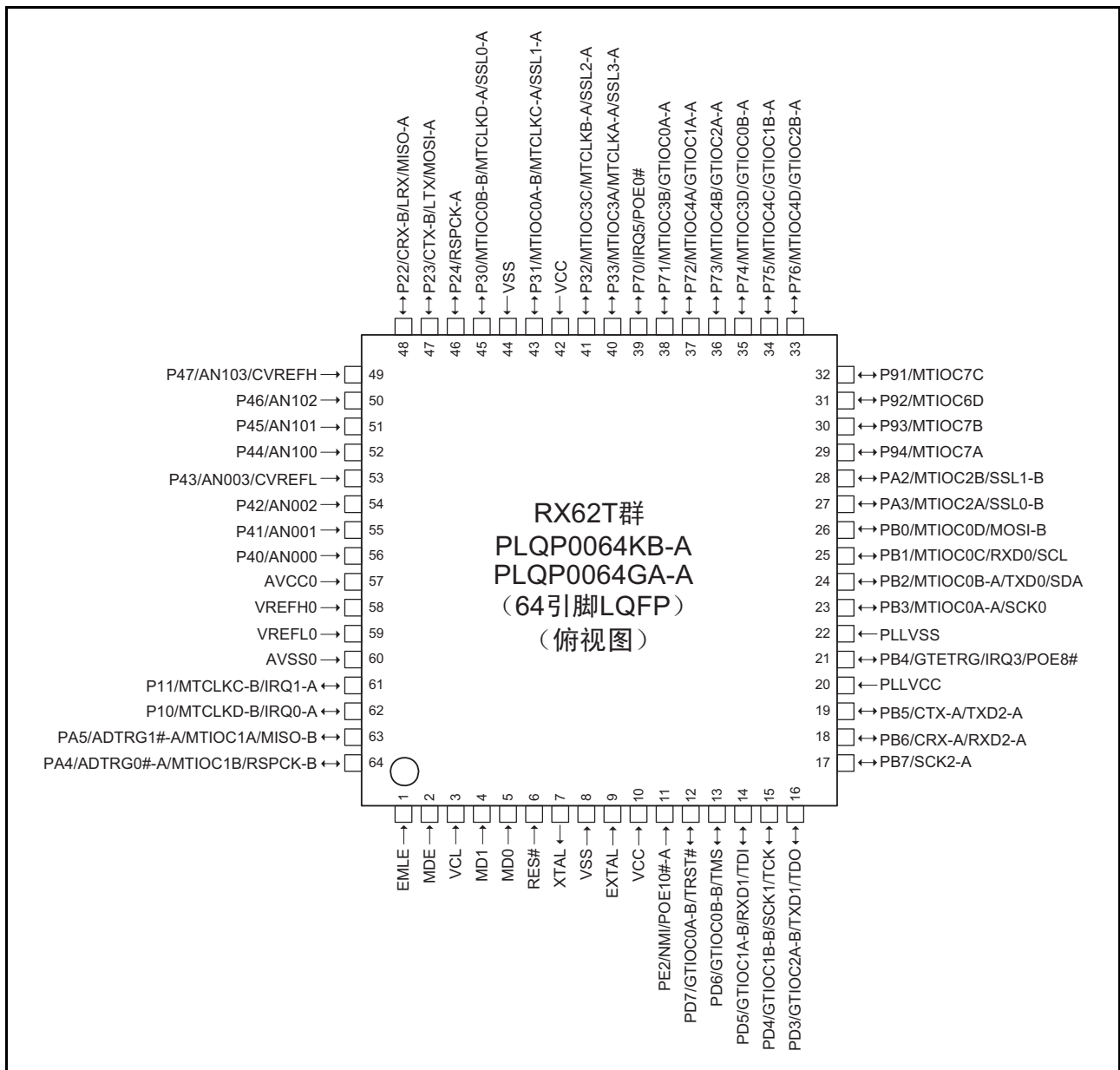


图 1.6 64 引脚 LQFP 的引脚排列图



表 1.4 各功能引脚一览表 (112 引脚 LQFP) (1/3)

引脚序号 (112 引脚 LQFP)	电源时钟 系统控制	I/O 端口	模拟	定时器	通信	中断	POE	调试
1		PE5				IRQ0-B		
2	EMLE							
3	VSS							
4	MDE							
5	VCL							
6	MD1							
7	MD0							
8		PE4		MTCLKC-C		IRQ1-B	POE10#-B	
9		PE3		MTCLKD-C		IRQ2-A	POE11#	
10	RES#							
11	XTAL							
12	VSS							
13	EXTAL							
14	VCC							
15		PE2				NMI	POE10#-A	
16		PE1			SSL3-C			
17		PE0			CRX-C/ SSL2-C			
18		PD7		GTIOC0A-B	CTX-C/ SSL1-C			
19		PD6		GTIOC0B-B	SSL0-C			
20		PD5		GTIOC1A-B	RXD1			
21		PD4		GTIOC1B-B	SCK1			
22		PD3		GTIOC2A-B	TXD1			
23		PD2		GTIOC2B-B	MOSI-C			
24		PD1		GTIOC3A	MISO-C			
25		PD0		GTIOC3B	RSPCK-C			
26								TDI
27								TCK
28								TDO
29		PB7			SCK2-A			
30		PB6			CRX-A/ RXD2-A			
31		PB5			CTX-A/ TXD2-A			
32	PLLVC							
33		PB4		GTETR		IRQ3	POE8#	
34	PLLVS							
35		PB3		MTIOC0A-A	SCK0			
36		PB2		MTIOC0B-A	TXD0/SDA			
37		PB1		MTIOC0C	RXD0/SCL			
38		PB0		MTIOC0D	MOSI-B			
39		PA5	ADTRG1#-A	MTIOC1A	MISO-B			
40		PA4	ADTRG0#-A	MTIOC1B	RSPCK-B			
41		PA3		MTIOC2A	SSL0-B			
42		PA2		MTIOC2B	SSL1-B			

表 1.4 各功能引脚一览表 (112 引脚 LQFP) (2/3)

引脚序号 (112 引脚 LQFP)	电源时钟 系统控制	I/O 端口	模拟	定时器	通信	中断	POE	调试
43		PA1		MTIOC6A	SSL2-B			
44		PA0		MTIOC6C	SSL3-B			
45	VCC							
46		P96				IRQ4	POE4#	
47	VSS							
48		P95		MTIOC6B				
49		P94		MTIOC7A				
50		P93		MTIOC7B				
51		P92		MTIOC6D				
52		P91		MTIOC7C				
53		P90		MTIOC7D				
54		PG5						TRCLK
55		PG4						TRDATA3
56		PG3						TRDATA2
57		PG2				IRQ2-B		TRDATA1
58		PG1				IRQ1-C		TRDATA0
59		PG0				IRQ0-C		TRSYNC
60		P76		MTIOC4D/ GTIOC2B-A				
61		P75		MTIOC4C/ GTIOC1B-A				
62		P74		MTIOC3D/ GTIOC0B-A				
63		P73		MTIOC4B/ GTIOC2A-A				
64		P72		MTIOC4A/ GTIOC1A-A				
65		P71		MTIOC3B/ GTIOC0A-A				
66		P70				IRQ5	POE0#	
67		P33		MTIOC3A/ MTCLKA-A	SSL3-A			
68		P32		MTIOC3C/ MTCLKB-A	SSL2-A			
69	VCC							
70		P31		MTIOC0A-B/ MTCLKC-A	SSL1-A			
71	VSS							
72		P30		MTIOC0B-B/ MTCLKD-A	SSL0-A			
73		P24			RSPCK-A			
74		P23			CTX-B/ LTX/ MOSI-A			
75		P22	ADTRG#		CRX-B/ LRX/ MISO-A			

表 1.4 各功能引脚一览表 (112 引脚 LQFP) (3 / 3)

引脚序号 (112 引脚 LQFP)	电源时钟 系统控制	I/O 端口	模拟	定时器	通信	中断	POE	调试
76		P21	ADTRG1#-B	MTCLKA-B		IRQ6		
77		P20	ADTRG0#-B	MTCLKB-B		IRQ7		
78		P65	AN5					
79		P64	AN4					
80	AVCC							
81	VREF							
82	AVSS							
83		P63	AN3					
84		P62	AN2					
85		P61	AN1					
86		P60	AN0					
87		P55	AN11					
88		P54	AN10					
89		P53	AN9					
90		P52	AN8					
91		P51	AN7					
92		P50	AN6					
93		P47	AN103/ CVREFH					
94		P46	AN102					
95		P45	AN101					
96		P44	AN100					
97		P43	AN003/ CVREFL					
98		P42	AN002					
99		P41	AN001					
100		P40	AN000					
101	AVCC0							
102	VREFH0							
103	VREFL0							
104	AVSS0							
105		P82		MTIC5U	SCK2-B			
106		P81		MTIC5V	TXD2-B			
107		P80		MTIC5W	RXD2-B			
108				WDTOVF#				
109		P11		MTCLKC-B		IRQ1-A		
110		P10		MTCLKD-B		IRQ0-A		
111								TRST#
112								TMS

表 1.5 各功能引脚一览表 (100 引脚 LQFP) (1/3)

引脚序号 (100 引脚 LQFP)	电源时钟 系统控制	I/O 端口	模拟	定时器	通信	中断	POE	调试
1		PE5				IRQ0-B		
2	EMLE							
3	VSS							
4	MDE							
5	VCL							
6	MD1							
7	MD0							
8		PE4		MTCLKC-C		IRQ1-B	POE10#-B	
9		PE3		MTCLKD-C		IRQ2-A	POE11#	
10	RES#							
11	XTAL							
12	VSS							
13	EXTAL							
14	VCC							
15		PE2				NMI	POE10#-A	
16		PE1			SSL3-C			
17		PE0			CRX-C/ SSL2-C			
18		PD7		GTIOC0A-B	CTX-C/ SSL1-C			TRST#
19		PD6		GTIOC0B-B	SSL0-C			TMS
20		PD5		GTIOC1A-B	RXD1			TDI
21		PD4		GTIOC1B-B	SCK1			TCK
22		PD3		GTIOC2A-B	TXD1			TDO
23		PD2		GTIOC2B-B	MOSI-C			TRCLK
24		PD1		GTIOC3A	MISO-C			TRDATA3
25		PD0		GTIOC3B	RSPCK-C			TRDATA2
26		PB7			SCK2-A			TRDATA1
27		PB6			CRX-A/ RXD2-A			TRDATA0
28		PB5			CTX-A/ TXD2-A			TRSYNC
29	PLLVC							
30		PB4		GTETRG		IRQ3	POE8#	
31	PLLVS							
32		PB3		MTIOC0A-A	SCK0			
33		PB2		MTIOC0B-A	TXD0/SDA			
34		PB1		MTIOC0C	RXD0/SCL			
35		PB0		MTIOC0D	MOSI-B			
36		PA5	ADTRG1#-A	MTIOC1A	MISO-B			
37		PA4	ADTRG0#-A	MTIOC1B	RSPCK-B			
38		PA3		MTIOC2A	SSL0-B			
39		PA2		MTIOC2B	SSL1-B			
40		PA1		MTIOC6A	SSL2-B			
41		PA0		MTIOC6C	SSL3-B			
42	VCC							

表 1.5 各功能引脚一览表 (100 引脚 LQFP) (2/3)

引脚序号 (100 引脚 LQFP)	电源时钟 系统控制	I/O 端口	模拟	定时器	通信	中断	POE	调试
43		P96				IRQ4	POE4#	
44	VSS							
45		P95		MTIOC6B				
46		P94		MTIOC7A				
47		P93		MTIOC7B				
48		P92		MTIOC6D				
49		P91		MTIOC7C				
50		P90		MTIOC7D				
51		P76		MTIOC4D/ GTIOC2B-A				
52		P75		MTIOC4C/ GTIOC1B-A				
53		P74		MTIOC3D/ GTIOC0B-A				
54		P73		MTIOC4B/ GTIOC2A-A				
55		P72		MTIOC4A/ GTIOC1A-A				
56		P71		MTIOC3B/ GTIOC0A-A				
57		P70				IRQ5	POE0#	
58		P33		MTIOC3A/ MTCLKA-A	SSL3-A			
59		P32		MTIOC3C/ MTCLKB-A	SSL2-A			
60	VCC							
61		P31		MTIOC0A-B/ MTCLKC-A	SSL1-A			
62	VSS							
63		P30		MTIOC0B-B/ MTCLKD-A	SSL0-A			
64		P24			RSPCK-A			
65		P23			CTX-B/ LTX/ MOSI-A			
66		P22	ADTRG#		CRX-B/ LRX/ MISO-A			
67		P21	ADTRG1#-B	MTCLKA-B		IRQ6		
68		P20	ADTRG0#-B	MTCLKB-B		IRQ7		
69		P65	AN5					
70		P64	AN4					
71	AVCC							
72	VREF							
73	AVSS							
74		P63	AN3					
75		P62	AN2					

表 1.5 各功能引脚一览表 (100 引脚 LQFP) (3/3)

引脚序号 (100 引脚 LQFP)	电源时钟 系统控制	I/O 端口	模拟	定时器	通信	中断	POE	调试
76		P61	AN1					
77		P60	AN0					
78		P55	AN11					
79		P54	AN10					
80		P53	AN9					
81		P52	AN8					
82		P51	AN7					
83		P50	AN6					
84		P47	AN103/ CVREFH					
85		P46	AN102					
86		P45	AN101					
87		P44	AN100					
88		P43	AN003/ CVREFL					
89		P42	AN002					
90		P41	AN001					
91		P40	AN000					
92	AVCC0							
93	VREFH0							
94	VREFL0							
95	AVSS0							
96		P82		MTIC5U	SCK2-B			
97		P81		MTIC5V	TXD2-B			
98		P80		MTIC5W	RXD2-B			
99		P11		MTCLKC-B		IRQ1-A		
100		P10		MTCLKD-B		IRQ0-A		

表 1.6 各功能引脚一览表 (80 引脚 LQFP) (1/3)

引脚序号 (80 引脚 LQFP)	电源时钟 系统控制	I/O 端口	模拟	定时器	通信	中断	POE	调试
1	EMLE							
2	VSS							
3	MDE							
4	VCL							
5	MD1							
6	MD0							
7		PE4		MTCLKC-C		IRQ1-B	POE10#-B	
8		PE3		MTCLKD-C		IRQ2-A	POE11#	
9	RES#							
10	XTAL							
11	VSS							
12	EXTAL							
13	VCC							
14		PE2				NMI	POE10#-A	
15		PE0			CRX-C			
16		PD7		GTIOC0A-B	CTX-C			TRST#
17		PD6		GTIOC0B-B				TMS
18		PD5		GTIOC1A-B	RXD1			TDI
19		PD4		GTIOC1B-B	SCK1			TCK
20		PD3		GTIOC2A-B	TXD1			TDO
21		PB7			SCK2-A			
22		PB6			CRX-A/ RXD2-A			
23		PB5			CTX-A/ TXD2-A			
24	PLLVCC							
25		PB4		GTETRG		IRQ3	POE8#	
26	PLLVSS							
27		PB3		MTIOC0A-A	SCK0			
28		PB2		MTIOC0B-A	TXD0/SDA			
29		PB1		MTIOC0C	RXD0/SCL			
30		PB0		MTIOC0D	MOSI-B			
31		PA3		MTIOC2A	SSL0-B			
32		PA2		MTIOC2B	SSL1-B			
33	VCC							
34		P96				IRQ4	POE4#	
35	VSS							
36		P95		MTIOC6B				
37		P94		MTIOC7A				
38		P93		MTIOC7B				
39		P92		MTIOC6D				
40		P91		MTIOC7C				
41		P76		MTIOC4D/ GTIOC2B-A				
42		P75		MTIOC4C/ GTIOC1B-A				

表 1.6 各功能引脚一览表 (80 引脚 LQFP) (2 / 3)

引脚序号 (80 引脚 LQFP)	电源时钟 系统控制	I/O 端口	模拟	定时器	通信	中断	POE	调试
43		P74		MTIOC3D/ GTIOC0B-A				
44		P73		MTIOC4B/ GTIOC2A-A				
45		P72		MTIOC4A/ GTIOC1A-A				
46		P71		MTIOC3B/ GTIOC0A-A				
47		P70				IRQ5	POE0#	
48		P33		MTIOC3A/ MTCLKA-A	SSL3-A			
49		P32		MTIOC3C/ MTCLKB-A	SSL2-A			
50	VCC							
51		P31		MTIOC0A-B/ MTCLKC-A	SSL1-A			
52	VSS							
53		P30		MTIOC0B-B/ MTCLKD-A	SSL0-A			
54		P24			RSPCK-A			
55		P23			CTX-B/ LTX/ MOSI-A			
56		P22	ADTRG#		CRX-B/ LRX/ MISO-A			
57		P21	ADTRG1#-B	MTCLKA-B		IRQ6		
58		P20	ADTRG0#-B	MTCLKB-B		IRQ7		
59	AVCC							
60	AVSS							
61		P63	AN3					
62		P62	AN2					
63		P61	AN1					
64		P60	AN0					
65		P47	AN103/ CVREFH					
66		P46	AN102					
67		P45	AN101					
68		P44	AN100					
69		P43	AN003/ CVREFL					
70		P42	AN002					
71		P41	AN001					
72		P40	AN000					
73	AVCC0							
74	VREFH0							
75	VREFL0							



表 1.6 各功能引脚一览表 (80 引脚 LQFP) (3 / 3)

引脚序号 (80 引脚 LQFP)	电源时钟 系统控制	I/O 端口	模拟	定时器	通信	中断	POE	调试
76	AVSS0							
77		P11		MTCLKC-B		IRQ1-A		
78		P10		MTCLKD-B		IRQ0-A		
79		PA5	ADTRG1#-A	MTIOC1A	MISO-B			
80		PA4	ADTRG0#-A	MTIOC1B	RSPCK-B			

表 1.7 各功能引脚一览表 (64 引脚 LQFP) (1/2)

引脚序号 (64 引脚 LQFP)	电源时钟 系统控制	I/O 端口	模拟	定时器	通信	中断	POE	调试
1	EMLE							
2	MDE							
3	VCL							
4	MD1							
5	MD0							
6	RES#							
7	XTAL							
8	VSS							
9	EXTAL							
10	VCC							
11		PE2				NMI	POE10#-A	
12		PD7		GTIOC0A-B				TRST#
13		PD6		GTIOC0B-B				TMS
14		PD5		GTIOC1A-B	RXD1			TDI
15		PD4		GTIOC1B-B	SCK1			TCK
16		PD3		GTIOC2A-B	TXD1			TDO
17		PB7			SCK2-A			
18		PB6			CRX-A/ RXD2-A			
19		PB5			CTX-A/ TXD2-A			
20	PLLVCC							
21		PB4		GTETRQ		IRQ3	POE8#	
22	PLLVSS							
23		PB3		MTIOC0A-A	SCK0			
24		PB2		MTIOC0B-A	TXD0/SDA			
25		PB1		MTIOC0C	RXD0/SCL			
26		PB0		MTIOC0D	MOSI-B			
27		PA3		MTIOC2A	SSL0-B			
28		PA2		MTIOC2B	SSL1-B			
29		P94		MTIOC7A				
30		P93		MTIOC7B				
31		P92		MTIOC6D				
32		P91		MTIOC7C				
33		P76		MTIOC4D/ GTIOC2B-A				
34		P75		MTIOC4C/ GTIOC1B-A				
35		P74		MTIOC3D/ GTIOC0B-A				
36		P73		MTIOC4B/ GTIOC2A-A				
37		P72		MTIOC4A/ GTIOC1A-A				

表 1.7 各功能引脚一览表 (64 引脚 LQFP) (2 / 2)

引脚序号 (64 引脚 LQFP)	电源时钟 系统控制	I/O 端口	模拟	定时器	通信	中断	POE	调试
38		P71		MTIOC3B/ GTIOC0A-A				
39		P70				IRQ5	POE0#	
40		P33		MTIOC3A/ MTCLKA-A	SSL3-A			
41		P32		MTIOC3C/ MTCLKB-A	SSL2-A			
42	VCC							
43		P31		MTIOC0A-B/ MTCLKC-A	SSL1-A			
44	VSS							
45		P30		MTIOC0B-B/ MTCLKD-A	SSL0-A			
46		P24			RSPCK-A			
47		P23			CTX-B/ LTX/ MOSI-A			
48		P22			CRX-B/ LRX/ MISO-A			
49		P47	AN103/ CVREFH					
50		P46	AN102					
51		P45	AN101					
52		P44	AN100					
53		P43	AN003/ CVREFL					
54		P42	AN002					
55		P41	AN001					
56		P40	AN000					
57	AVCC0							
58	VREFH0							
59	VREFL0							
60	AVSS0							
61		P11		MTCLKC-B		IRQ1-A		
62		P10		MTCLKD-B		IRQ0-A		
63		PA5	ADTRG1#-A	MTIOC1A	MISO-B			
64		PA4	ADTRG0#-A	MTIOC1B	RSPCK-B			

## 1.5 引脚功能

引脚功能一览表如表 1.8 所示。

表 1.8 引脚功能一览表 (1/5)

分类	引脚名	输入 / 输出	功能
电源	VCC	输入	电源引脚 必须连接系统电源。
	VCL	输入	必须通过 0.1 $\mu$ F 的电容连接 VSS，并且使电容器靠近引脚。
	VSS	输入	接地引脚 必须连接系统电源 (0V)。
	PLLVC	输入	用于 PLL 电路的电源引脚 必须连接系统电源。
	PLLVSS	输入	用于 PLL 电路的接地引脚
时钟	XTAL	输出	连接晶体谐振器的引脚
	EXTAL	输入	EXTAL 引脚也能输入外部时钟。
运行模式的控制	MD0、MD1、MDE	输入	设定运行模式。不能在运行中更改这些引脚。
系统控制	RES#	输入	复位引脚 当此引脚为 Low 电平时，进入复位状态。
	EMLE	输入	允许连接内部仿真器的引脚 当使用内部仿真器时，必须置为 High 电平，否则就必须置为 Low 电平。
内部仿真器	TRST#	输入	用于内部仿真器的引脚
	TMS	输入	如果将 EMLE 引脚置为 High 电平，就变为内部仿真器的专用引脚。
	TDI	输入	
	TCK	输入	
	TDO	输出	
	TRCLK	输出	输出用于与跟踪数据取得同步的时钟。80 引脚版和 64 引脚版中无此引脚。
	TRSYNC	输出	表示 TRDATA0 ~ TRDATA3 引脚的输出为有效数据。80 引脚版和 64 引脚版中无此引脚。
TRDATA0 ~ TRDATA3	输出	输出跟踪信息。80 引脚版和 64 引脚版中无此引脚。	
中断 (ICU)	NMI	输入	非屏蔽中断请求引脚
	IRQ0-A/IRQ0-B/IRQ0-C IRQ1-A/IRQ1-B/IRQ1-C IRQ2-A/IRQ2-B IRQ3 ~ IRQ7	输入	中断请求引脚 在 100 引脚版中无 IRQ0-C/IRQ1-C/IRQ2-B 引脚；在 80 引脚版中无 IRQ0-B/IRQ0-C/IRQ1-C/IRQ2-B 引脚；在 64 引脚版中无 IRQ0-B/IRQ0-C/IRQ1-B/IRQ1-C/IRQ2-A/IRQ2-B/IRQ4/IRQ6/IRQ7 引脚。

表 1.8 引脚功能一览表 (2/5)

分类	引脚名	输入 / 输出	功能
多功能定时器脉冲单元 3 (MTU3)	MTIOC0A-A/MTIOC0A-B MTIOC0B-A/MTIOC0B-B MTIOC0C、MTIOC0D	输入 / 输出	MTU0.TGRA ~ 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
	MTIOC1A、MTIOC1B	输入 / 输出	MTU1.TGRA 和 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
	MTIOC2A、MTIOC2B	输入 / 输出	MTU2.TGRA 和 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
	MTIOC3A、MTIOC3B MTIOC3C、MTIOC3D	输入 / 输出	MTU3.TGRA ~ 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 MTIOC3B 引脚和 MTIOC3D 引脚能用作大电流输出引脚。
	MTIOC4A、MTIOC4B MTIOC4C、MTIOC4D	输入 / 输出	MTU4.TGRA 和 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 全部引脚都能用作大电流输出引脚。
	MTIC5U、MTIC5V、 MTIC5W	输入	MTU5.TGRU、MTU5.TGRV 和 MTU5.TGRW 的输入捕捉的输入引脚 / 死区时间补偿功能的输入引脚 80 引脚版和 60 引脚版中无这些引脚。
	MTIOC6A、MTIOC6B MTIOC6C、MTIOC6D	输入 / 输出	MTU6.TGRA ~ 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 MTIOC6B 引脚和 MTIOC6D 引脚能用作大电流输出引脚。 80 引脚版中无 MTIOC6A/MTIOC6C 引脚；64 引脚版中无 MTIOC6A/MTIOC6B/MTIOC6C 引脚。
	MTIOC7A、MTIOC7B MTIOC7C、MTIOC7D	输入 / 输出	MTU7.TGRA ~ 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 全部引脚都能用作大电流输出引脚。 80 引脚版和 64 引脚版中无 MTIOC7D 引脚。
	MTCLKA-A/MTCLKA-B MTCLKB-A/MTCLKB-B MTCLKC-A/MTCLKC-B/ MTCLKC-C MTCLKD-A/MTCLKD-B/ MTCLKD-C	输入	输入外部时钟。 64 引脚版中无 MTCLKA-B/MTCLKB-B/MTCLKC-C/ MTCLKD-C 引脚。
通用 PWM 定时器 (GPT)	GTIOC0A-A/GTIOC0A-B GTIOC0B-A/GTIOC0B-B	输入 / 输出	GPT0.GTCCRA、GPT0.GTCCRB 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 GTIOC0A-A 引脚和 GTIOC0B-A 引脚能用作大电流输出引脚。
	GTIOC1A-A/GTIOC1A-B GTIOC1B-A/GTIOC1B-B	输入 / 输出	GPT1.GTCCRA、GPT1.GTCCRB 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 GTIOC1A-A 引脚和 GTIOC1B-A 引脚能用作大电流输出引脚。
	GTIOC2A-A/GTIOC2A-B GTIOC2B-A/GTIOC2B-B	输入 / 输出	GPT2.GTCCRA、GPT2.GTCCRB 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 GTIOC2A-A 引脚和 GTIOC2B-A 引脚能用作大电流输出引脚。 80 引脚版和 64 引脚版中无 GTIOC2B-B 引脚。
	GTIOC3A、GTIOC3B	输入 / 输出	GPT3.GTCCRA、GPT3.GTCCRB 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 80 引脚版和 64 引脚版中无这些引脚。
	GTETRG	输入	外部触发的输入引脚

表 1.8 引脚功能一览表 (3/5)

分类	引脚名	输入 / 输出	功能
端口输出允许 3 (POE3)	POE0#、POE4#、 POE8# POE10#-A/POE10#-B POE11#	输入	输入将 MTU3 和 GPT 的大电流引脚置为高阻抗状态的请求信号。 64 引脚版中无 POE4#/POE10#-B/POE11# 引脚。
看门狗定时器 (WDT)	WDTOVF#	输出	看门狗定时器模式中的计数器上溢信号的输出引脚 100 引脚版、80 引脚版以及 64 引脚版中无此引脚。
串行通信接口 (SC1b)	TXD0、TXD1、 TXD2-A/TXD2-B	输出	发送数据的输出引脚 80 引脚版和 64 引脚版中无 TXD2-B 引脚。
	RXD0、RXD1、 RXD2-A/RXD2-B	输入	接收数据的输入引脚 80 引脚版和 64 引脚版中无 RXD2-B 引脚。
	SCK0、SCK1、 SCK2-A/SCK2-B	输入 / 输出	时钟的输入 / 输出引脚 80 引脚版和 64 引脚版中无 SCK2-B 引脚。
I <sup>2</sup> C 总线接口 (RIIC)	SCL	输入 / 输出	I <sup>2</sup> C 总线接口的时钟输入 / 输出引脚 能通过 NMOS 漏极开路输出，直接驱动总线。
	SDA	输入 / 输出	I <sup>2</sup> C 总线接口的数据输入 / 输出引脚 能通过 NMOS 漏极开路输出，直接驱动总线。
CAN 模块 (CAN) (选项)	CRX-A/CRX-B/CRX-C	输入	CAN 的输入引脚 64 引脚版中无 CRX-C 引脚。
	CTX-A/CTX-B/CTX-C	输出	CAN 的输出引脚 64 引脚版中无 CTX-C 引脚。
LIN 模块 (LIN)	LRX	输入	LIN 的输入引脚
	LTX	输出	LIN 的输出引脚
串行外围接口 (RSPI)	RSPCK-A/RSPCK-B/ RSPCK-C	输入 / 输出	RSPI 的时钟输入 / 输出引脚 80 引脚版和 64 引脚版中无 RSPCK-C 引脚。
	MOSI-A/MOSI-B/MOSI-C	输入 / 输出	RSPI 的主控发送数据引脚 80 引脚版和 64 引脚版中无 MOSI-C 引脚。
	MISO-A/MISO-B/MISO-C	输入 / 输出	RSPI 的从属发送数据引脚 80 引脚版和 64 引脚版中无 MISO-C 引脚。
	SSL0-A/SSL0-B/SSL0-C	输入 / 输出	RSPI 的从属选择引脚
	SSL1-A/SSL1-B/SSL1-C SSL2-A/SSL2-B/SSL2-C SSL3-A/SSL3-B/SSL3-C	输出	80 引脚版和 64 引脚版中 SSL0-C/SSL1-C/SSL2-C/SSL3-C 引脚。

表 1.8 引脚功能一览表 (4 / 5)

分类	引脚名	输入 / 输出	功能
A/D 转换器	AN000 ~ AN003、 AN100 ~ AN103	输入	12 位 A/D 转换器的模拟输入引脚
	AN0 ~ AN11	输入	10 位 A/D 转换器的模拟输入引脚 80 引脚版中无 AN4 ~ AN11 引脚； 64 引脚版中无这些引脚。
	ADTRG0#-A/ADTRG0#-B ADTRG1#-A/ADTRG1#-B ADTRG#	输入	用于开始 A/D 转换的外部触发输入引脚 64 引脚版中无 ADTRG0#-B/ADTRG1#-B/ADTRG# 引脚。
	CVREFH	输入	用于比较器高电平基准电压的输入引脚
	CVREFL	输入	用于比较器低电平基准电压的输入引脚
	AVCC0	输入	12 位 A/D 转换器的模拟电源引脚 不使用 A/D 转换器时，必须连接系统电源。
	AVSS0	输入	12 位 A/D 转换器的接地引脚 必须连接系统电源 (0V)。
	VREFH0	输入	12 位 A/D 转换器的基准电源引脚 不使用 12 位 A/D 转换器时，必须连接系统电源。
	VREFL0	输入	12 位 A/D 转换器的基准电源的接地引脚 不使用 12 位 A/D 转换器时，必须连接系统电源 (0V)。
	AVCC	输入	10 位 A/D 转换器的模拟电源引脚 不使用 A/D 转换器时，必须连接系统电源。 64 引脚版中无此引脚。
	AVSS	输入	10 位 A/D 转换器的接地引脚 必须连接系统电源 (0V)。 64 引脚版中无此引脚。
	VREF	输入	10 位 A/D 转换器的基准电源引脚 不使用 10 位 A/D 转换器时，必须连接系统电源。 80 引脚版和 64 引脚版中无此引脚。

表 1.8 引脚功能一览表 (5/5)

分类	引脚名	输入 / 输出	功能
I/O 端口	P10、P11	输入 / 输出	2 位输入 / 输出引脚
	P20 ~ P24	输入 / 输出	5 位输入 / 输出引脚 64 引脚版中无 P20/P21 引脚。
	P30 ~ P33	输入 / 输出	4 位输入 / 输出引脚
	P40 ~ P47	输入	8 位输入引脚
	P50 ~ P55	输入	6 位输入引脚 80 引脚版和 64 引脚版中无这些引脚。
	P60 ~ P65	输入	6 位输入引脚 80 引脚版中无 P64/P65 引脚；64 引脚版中无这些引脚。
	P70 ~ P76	输入 / 输出	7 位输入 / 输出引脚
	P80 ~ P82	输入 / 输出	3 位输入 / 输出引脚 80 引脚版和 64 引脚版中无这些引脚。
	P90 ~ P96	输入 / 输出	7 位输入 / 输出引脚 80 引脚版中无 P90 引脚；64 引脚版中无 P90/P95/P96 引脚。
	PA0 ~ PA5	输入 / 输出	6 位输入 / 输出引脚 80 引脚版和 64 引脚中无 PA0/PA1 引脚。
	PB0 ~ PB7	输入 / 输出	8 位输入 / 输出引脚
	PD0 ~ PD7	输入 / 输出	8 位输入 / 输出引脚 80 引脚版和 64 引脚版中无 PD0/PD1/PD2 引脚。
	PE0、PE1、PE3 ~ PE5	输入 / 输出	5 位输入 / 输出引脚 80 引脚版中无 PE1/PE5 引脚；64 引脚版中无这些引脚。
	PE2	输入	1 位输入引脚
PG0 ~ PG5	输入 / 输出	6 位输入 / 输出引脚 100 引脚版、80 引脚版以及 64 引脚版中无这些引脚。	

注. 装载 / 非装载引脚因封装种类而不同。详细内容请参照表 1.4 ~ 表 1.7 的各功能引脚一览表。



## 2. CPU

RX62T 群是装载了 RX CPU 的处理器。

RX CPU 采用了可变长度指令格式。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
  - 可变长度指令格式（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 段流水线
  - 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 个由 DSP 功能指令使用的累加器。

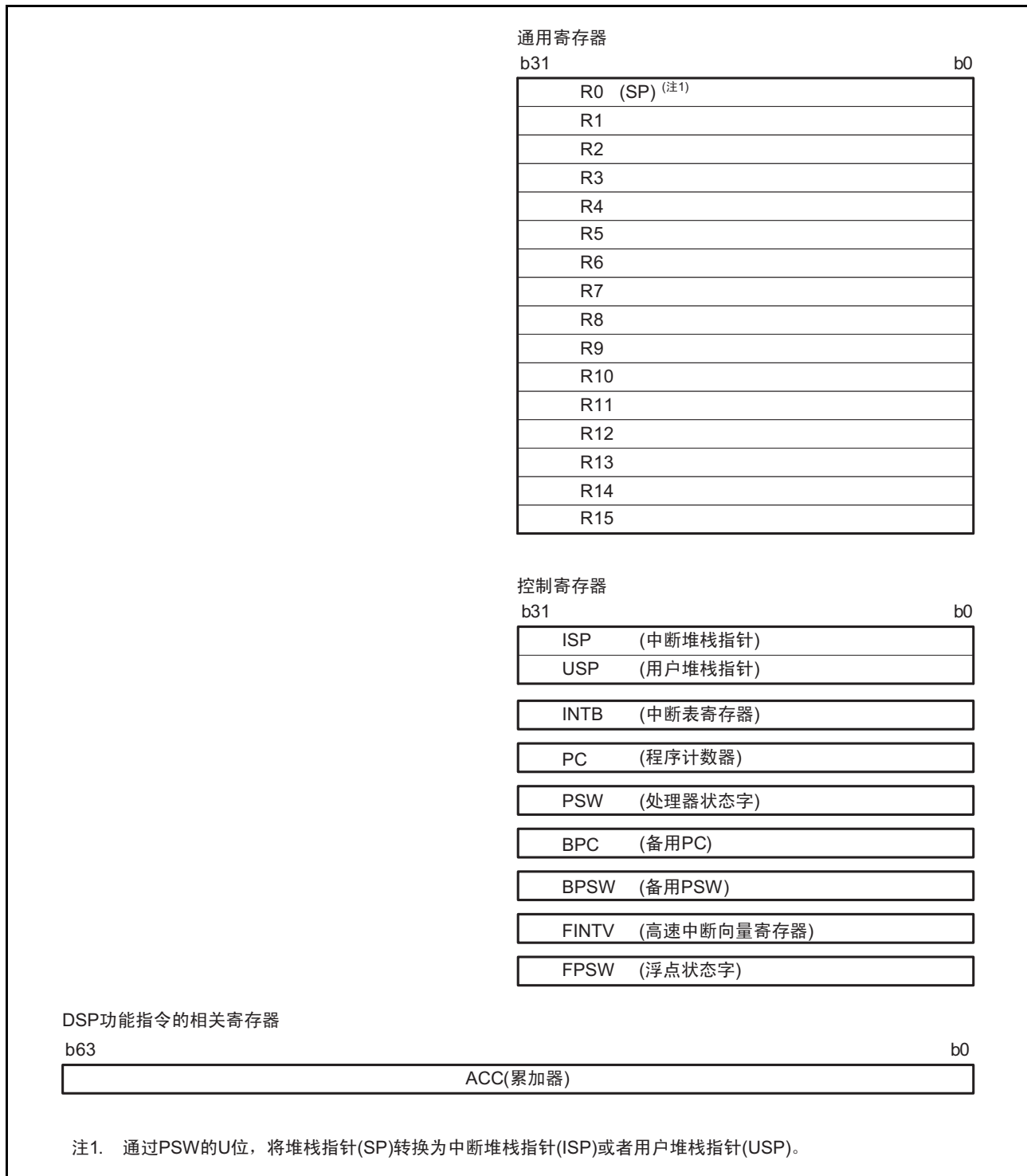


图 2.1 CPU 寄存器组

### 2.2.1 通用寄存器 (R0 ~ R15)

通用寄存器有 16 个 (R0 ~ R15)，用作数据寄存器和地址寄存器。

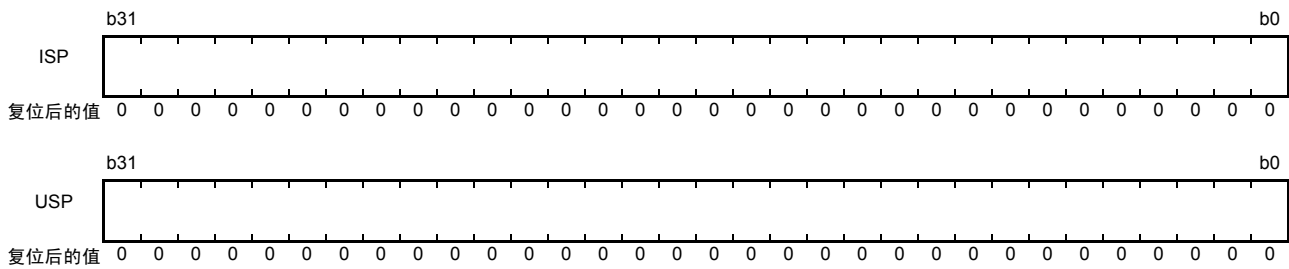
通用寄存器 R0 除了具有通用寄存器的功能以外，还有堆栈指针 (SP) 的功能。通过处理器状态字 (PSW) 的堆栈指针指定位 (U)，将 SP 转换为中断堆栈指针 (ISP) 或者用户堆栈指针 (USP)。

### 2.2.2 控制寄存器

控制寄存器有以下 9 个：

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

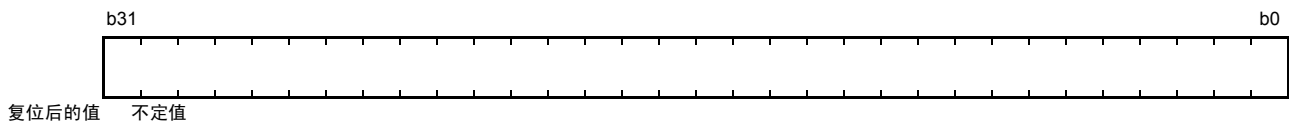
#### 2.2.2.1 中断堆栈指针 (ISP) / 用户堆栈指针 (USP)



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

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

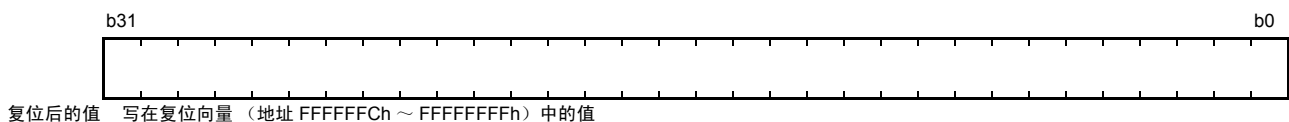
#### 2.2.2.2 中断表寄存器 (INTB)



必须给中断表寄存器 (INTB) 设定可变向量表的起始地址。

另外，必须给 INTB 寄存器设定 4 的倍数。

#### 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)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	IPL[3:0]				—	—	—	PM	—	—	U	I
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
—	—	—	—	—	—	—	—	—	—	—	—	O	S	Z	C
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	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用户模式时，必须在将被压栈的 PSW.PM 位置“1”后执行 RTE 指令、或者在将 BPSW.PM 位置“1”后执行 RTFI 指令。

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”。

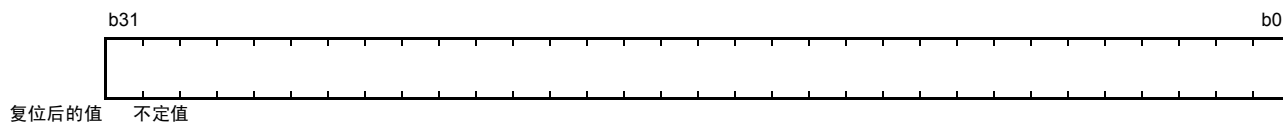
#### PM 位（处理器模式设定位）

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）

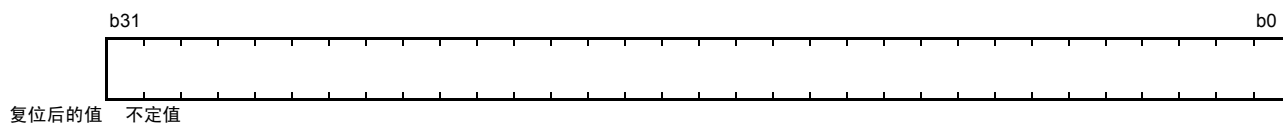
IPL[3:0] 位指定 0 级（最低）～ 15 级（最高）的 16 个处理器中断优先级。如果发生请求的中断的优先级高于处理器中断优先级，就允许该中断。在将 IPL[3:0] 位设定为“15 级”（Fh）时，禁止全部的中断。如果发生非屏蔽中断，IPL[3:0] 位就变为“15 级”（Fh）。如果发生中断，这些位就为所接受中断的优先级。

### 2.2.2.5 备用 PC (BPC)



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

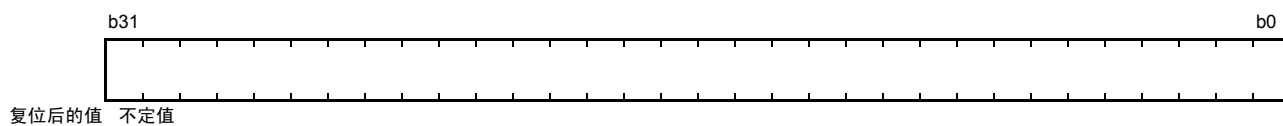
### 2.2.2.6 备用 PSW (BPSW)



备用 PSW (BPSW) 是为实现中断响应高速化而设计的寄存器。

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

### 2.2.2.7 高速中断向量寄存器 (FINTV)



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

## 2.2.2.8 浮点状态字 (FPSW)

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

位	符号	位名	功能	R/W
b1-b0	RM[1:0]	浮点舍入模式设定位	b1 b0 0 0: 向最近的值舍入 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
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

位	符号	位名	功能	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，确认是否发生异常处理。如果  $F_j$  标志为“1”，就在通过软件置“0”前保持“1” ( $j=X、U、Z、O、V$ )。

#### 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)

在发生 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) 和非安装处理时，对应的标志变为“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 FPU 运算指令而发生了 IEEE754 规格规定的 5 种异常时，控制 CPU 是否转移到异常处理。  
如果此位为“0”，就禁止异常处理；如果为“1”，就允许异常处理。

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

FU 标志（下溢标志）、FX 标志（精度异常标志）

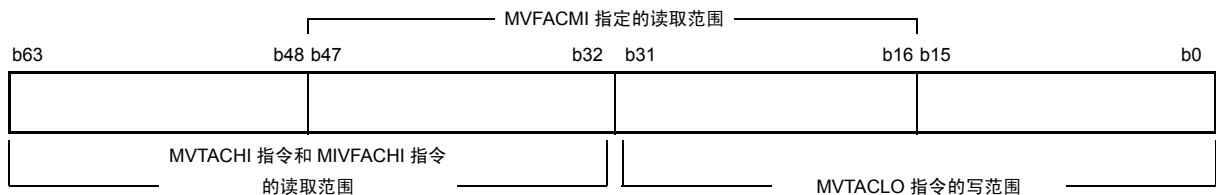
在异常处理允许位  $E_j$  为“0”（禁止异常处理）时，如果发生 IEEE754 规格规定的 5 种异常，对应的标志就变为“1”。

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

FS 标志（浮点错误概要标志）

此标志反映 FU、FZ、FO、FV 标志的逻辑或。

### 2.2.2.9 累加器（ACC）



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、RTE、MVTIPL、WAIT 指令。

### 2.3.4 处理器模式之间的转移

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

#### (1) 用户模式向管理模式的转移

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

#### (2) 管理模式向用户模式的转移

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

## 2.4 数据类型

RX CPU 能处理整数、浮点数、位、字符串共 4 种数据。

### 2.4.1 整数

整数有带符号整数和不带符号整数，带符号整数的负值用 2 的补数表现。

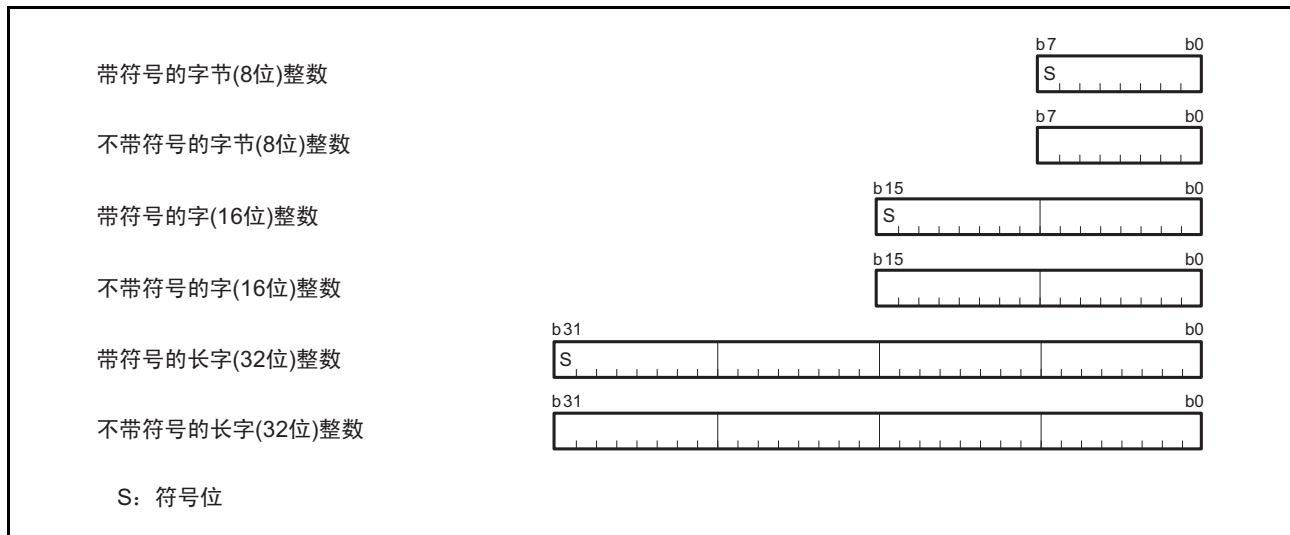


图 2.2 整数

### 2.4.2 浮点数

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

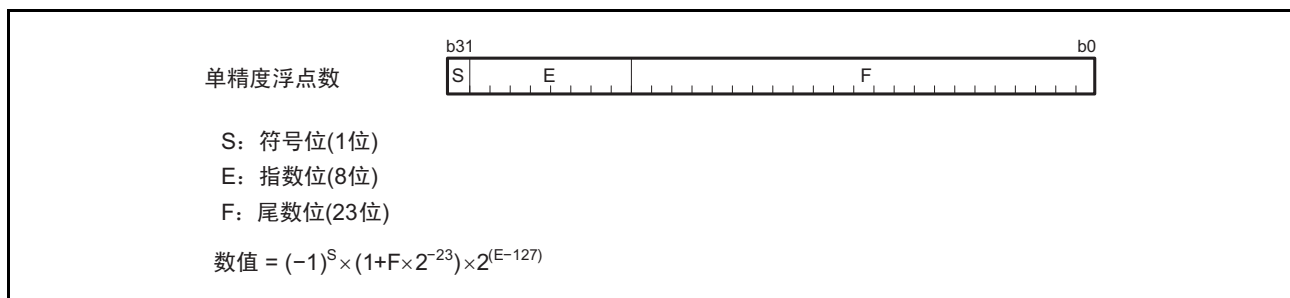


图 2.3 浮点数

浮点数支持以下数值：

- $0 < E < 255$  (规格化数 - Normal Numbers)
- $E = 0$  并且  $F = 0$  (零 - Signed Zero)
- $E = 0$  并且  $F > 0$  (非规格化数 - Subnormal Numbers) (注1)
- $E = 255$  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**、**BTST** 共 5 种指令。

通过对象寄存器和 31 ~ 0 的位号指定寄存器的位。

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

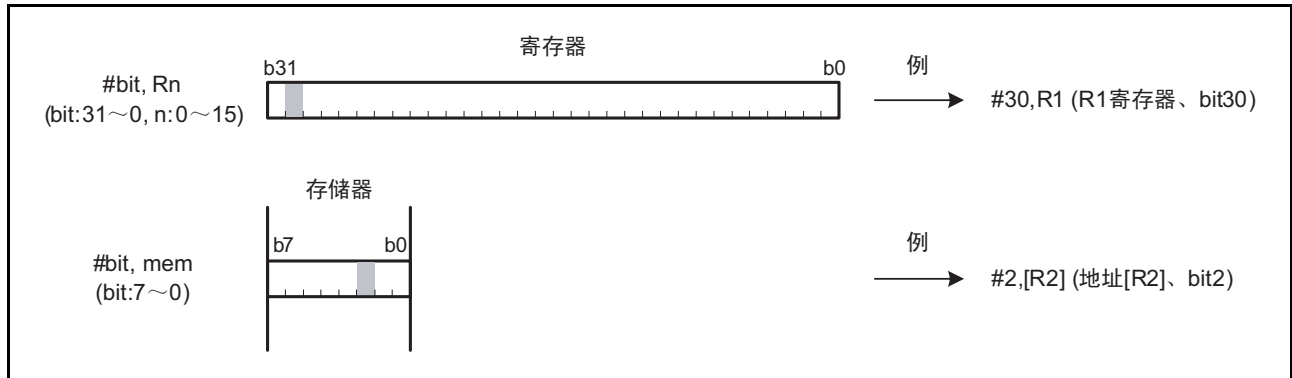


图 2.4 位

### 2.4.4 字符串

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

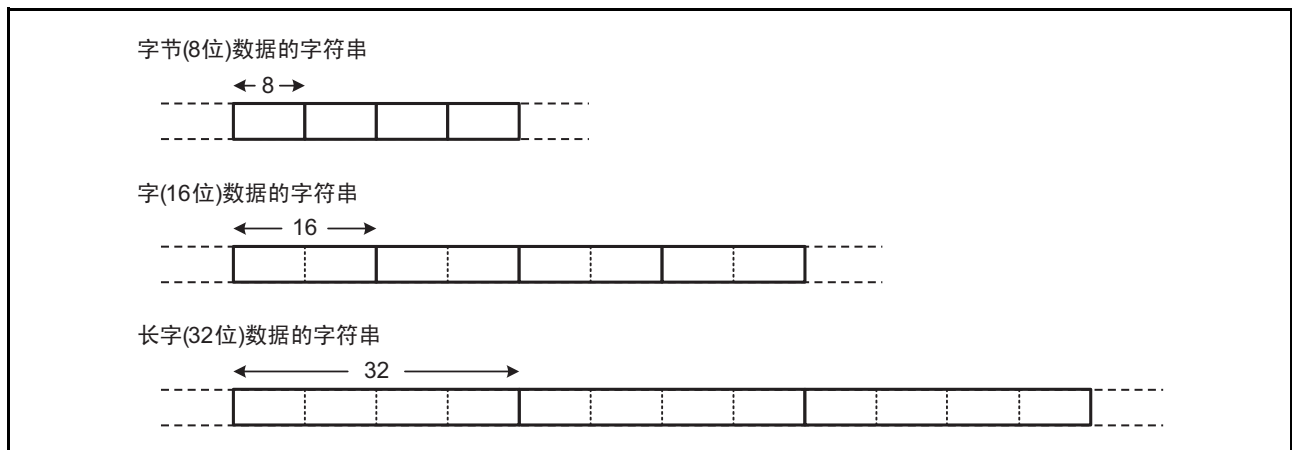


图 2.5 字符串

## 2.5 字节序

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 字节序的转换

在 RX62T 群中，能使用高位字节（MSB）为地址 0 的大端法和低位字节（LSB）为地址 0 的小端法两种字节数据的排列方法。

通过模式引脚（MDE）进行字节序的转换。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	从 LL 传送	—	—	—	—
地址 1	从 LH 传送	从 LL 传送	—	—	—
地址 2	从 HL 传送	从 LH 传送	从 LL 传送	—	—
地址 3	从 HH 传送	从 HL 传送	从 LH 传送	从 LL 传送	—
地址 4	—	从 HH 传送	从 HL 传送	从 LH 传送	从 LL 传送
地址 5	—	—	从 HH 传送	从 HL 传送	从 LH 传送
地址 6	—	—	—	从 HH 传送	从 HL 传送
地址 7	—	—	—	—	从 HH 传送

表 2.4 设定为大端法时的 32 位写操作

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

表 2.5 设定为小端法时的 16 位读操作

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

表 2.6 设定为大端法时的 16 位读操作

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

表 2.7 设定为小端法时的 16 位写操作

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

表 2.8 设定为大端法时的 16 位写操作

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

表 2.9 设定为小端法时的 8 位读操作

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

表 2.10 设定为大端法时的 8 位读操作

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

表 2.11 设定为小端法时的 8 位写操作

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

表 2.12 设定为大端法时的 8 位写操作

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

## 2.5.2 I/O 寄存器的存取

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



### 2.5.3 存取 I/O 寄存器时的注意事项

必须按照以下规则存取 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。

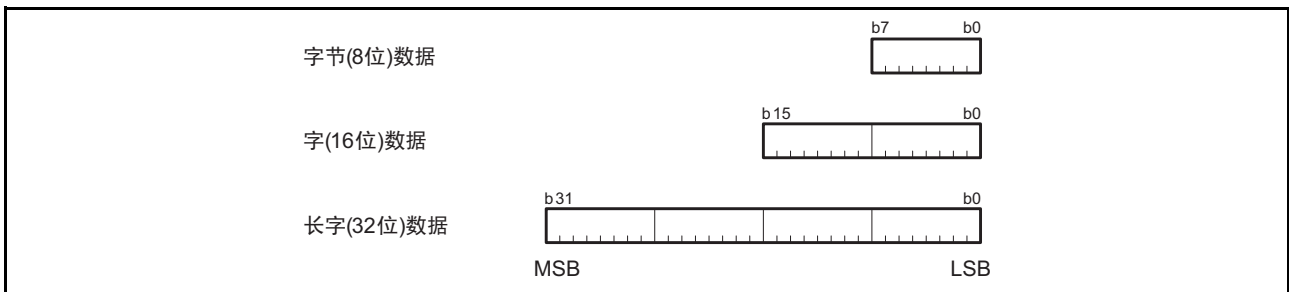


图 2.6 寄存器的数据排列

#### 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。

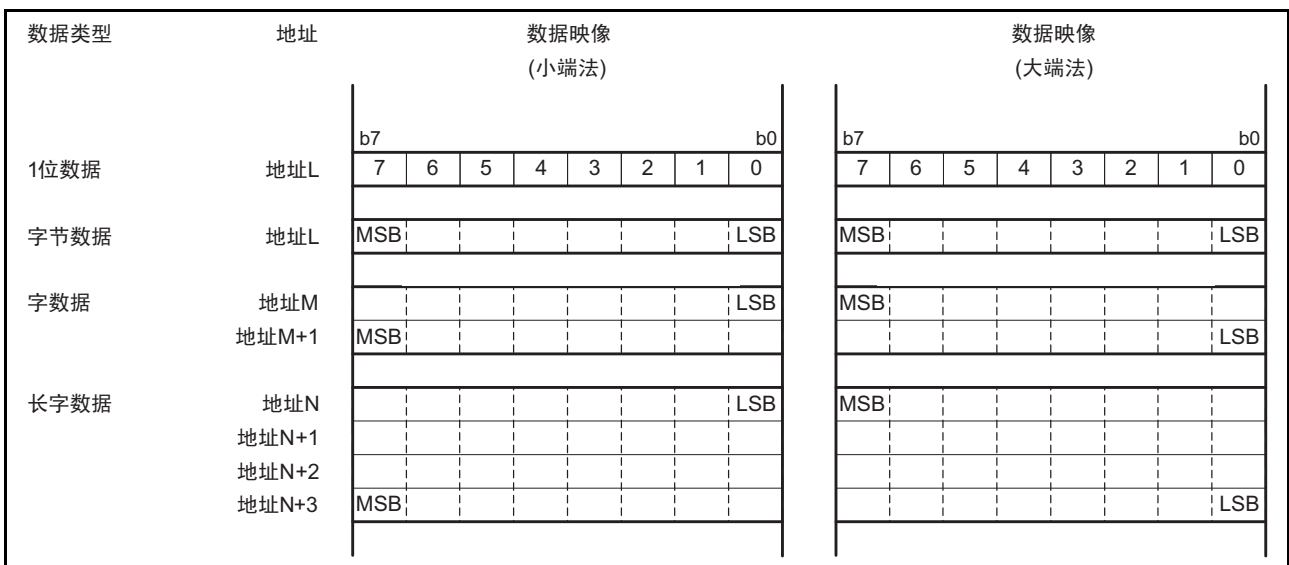


图 2.7 存储器的数据排列

## 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)	
FFFFFFFD0h	特权指令异常	
FFFFFFFD4h	存取异常	
FFFFFFFD8h	(保留区)	
FFFFFFDCh	未定义指令异常	
FFFFFFE0h	(保留区)	
FFFFFFE4h	浮点异常	
FFFFFFE8h	(保留区)	
FFFFFFECh	(保留区)	
FFFFFFF0h	(保留区)	
FFFFFFF4h	(保留区)	
FFFFFFF8h	非屏蔽中断	
FFFFFFFCh	复位	

图 2.8 固定向量表

### 2.6.2 可变向量表

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

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

另外，在中断源时，分配了各产品规定的向量号 (0 ~ 255)。有关中断的向量号，请参照“11. 中断控制器 (ICU)”。

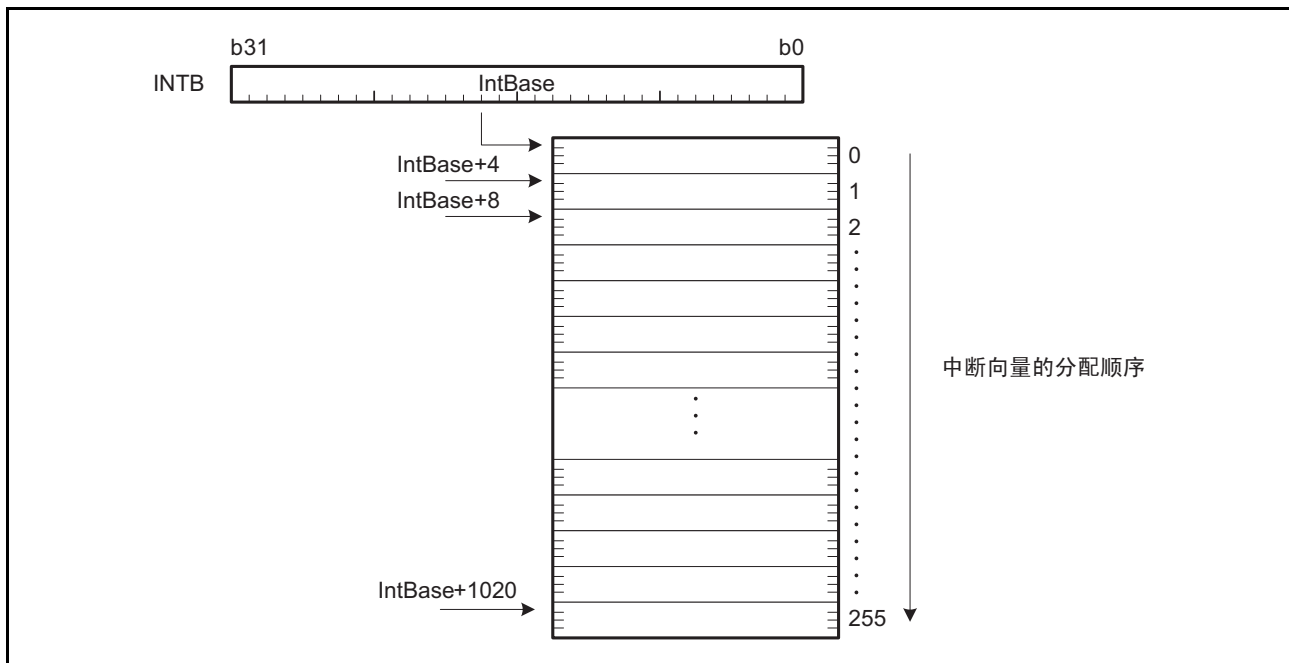


图 2.9 可变向量表

## 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 指令除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 和 OA2），只在存储器存取时使用此阶段。此阶段又分为 M1 和 M2 两个子阶段。RX CPU 中 M1 阶段和 M2 阶段各有 1 个存储器存取。

- M1 阶段（存储器存取阶段 1）

此阶段进行操作数的存储器存取（OA1）。

存储时：一旦总线接受写请求，就结束流水线处理。

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

- M2 阶段（存储器存取阶段 2）

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

#### (5) WB 阶段（回写阶段）

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

流水线结构及其运行如图 2.10 所示。

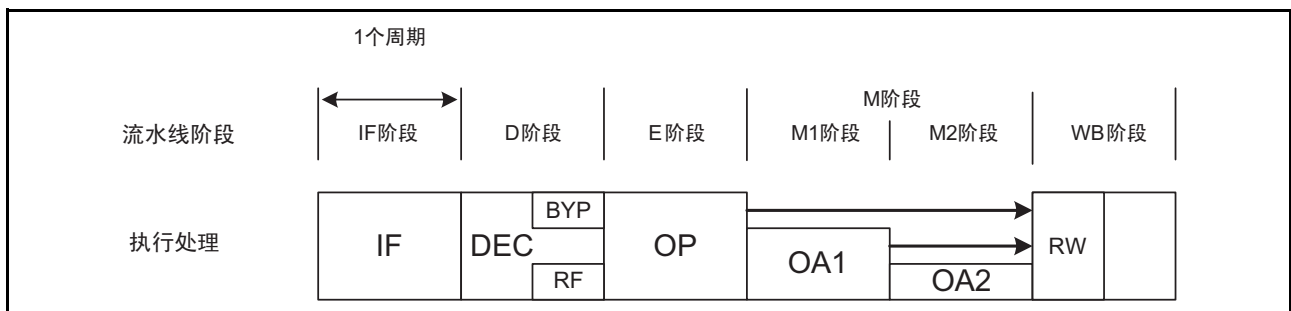


图 2.10 流水线结构及其运行

## 2.8.2 指令和流水线处理

表中操作数的表示法如下：

#IMM: 立即数

Rs、Rs2、Rd、Rd2、Ri、Rb: 通用寄存器

CR: 控制寄存器

dsp: dsp5、dsp8、dsp16、dsp24

pcdsp: pcdsp3、pcdsp8、pcdsp16、pcdsp24

### 2.8.2.1 转换为单一微操作的指令和流水线处理

转换为单一微操作的指令如下所示，周期数表示无等待存储器存取时的周期数。

表 2.13 转换为单一微操作的指令

指令	助记符 (当省略指令长度时, 为全长度共同的操作)	参照图	周期数
算术 / 逻辑运算指令 (寄存器之间、立即数 - 寄存器) DIV、DIVU、EMUL、EMULU、 RMPA 和 SATR 除外	• {ABS, ADC, ADD, AND, CMP, MAX, MIN, MUL, NEG, NOP, NOT, OR, ROLC, RORC, ROTL, ROTR, SAT, SBB, SHAR, SHLL, SHLR, SUB, TST, XOR} "#IMM, Rd"/"Rd" /"Rs, Rd"/"Rs, Rs2, Rd"	图 2.11	1
算术 / 逻辑运算指令 (除法)	• DIV "#IMM, Rd"/"Rs, Rd"	图 2.11	3 ~ 20 (注 1)
	• DIVU "#IMM, Rd"/"Rs, Rd"	图 2.11	2 ~ 18 (注 1)
传送指令 (寄存器之间、立即数 - 寄存器)	• {MOV, MOVU, REVL, REVW} "#IMM, Rd"/"Rs, Rd" • SCCnd "Rd" • {STNZ, STZ} "#IMM, Rd"	图 2.11	1
传送指令 (加载)	• {MOV, MOVU} "[Rs], Rd"/"dsp[Rs], Rd" /"[Rs+], Rd"/"[-Rs], Rd"/"Rs, [Ri, Rb]" • POP "Rd"	图 2.12	吞吐量: 1 等待时间: 2 (注 2)
传送指令 (存储)	• MOV "Rs, [Rd]"/"Rs, dsp[Rd]"/"Rs, [Rd+]" /"Rs, [-Rd]"/"Rs, [Ri, Rb]" • PUSH "Rs" • PUSHC "CR"	图 2.13	1
位操作指令 (寄存器)	• {BCLR, BNOT, BSET, BTST} "#IMM, Rd"/"Rs, Rd" • BMCnd "#IMM, Rd"	图 2.11	1
转移指令	• BCnd "pcdsp" • {BRA, BSR} "pcdsp"/"Rs" • {JMP, JSR} "Rs"	图 2.22	转移成立: 3 转移不成立: 1
浮点运算指令 (寄存器之间、立即数 - 寄存器)	• FCMP "#IMM, Rd"/"Rs, Rd"	图 2.11	1
系统控制指令	• CLRPSW, SETPSW "#IMM" • MVTC "#IMM, CR"/"Rs, CR" • MVFC "CR, Rd" • MVTIPL "#IMM"	—	1
DSP 功能指令	• {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”。

转换为单一微操作的指令操作如图 2.11 ~图 2.13 所示。

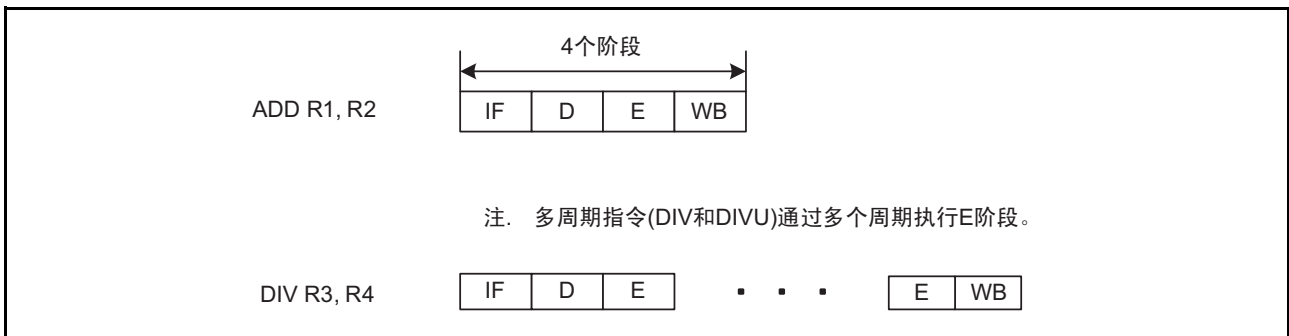


图 2.11 寄存器之间、立即数 - 寄存器的运算

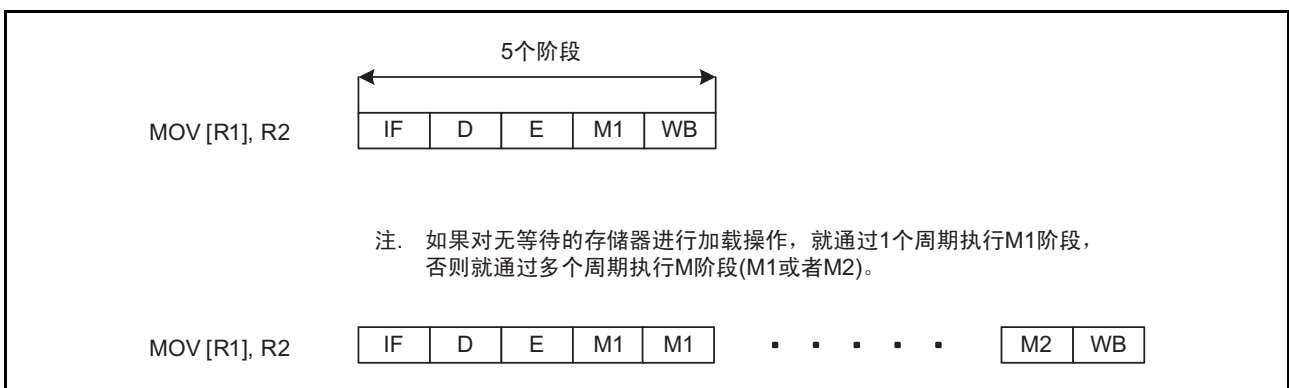


图 2.12 加载操作

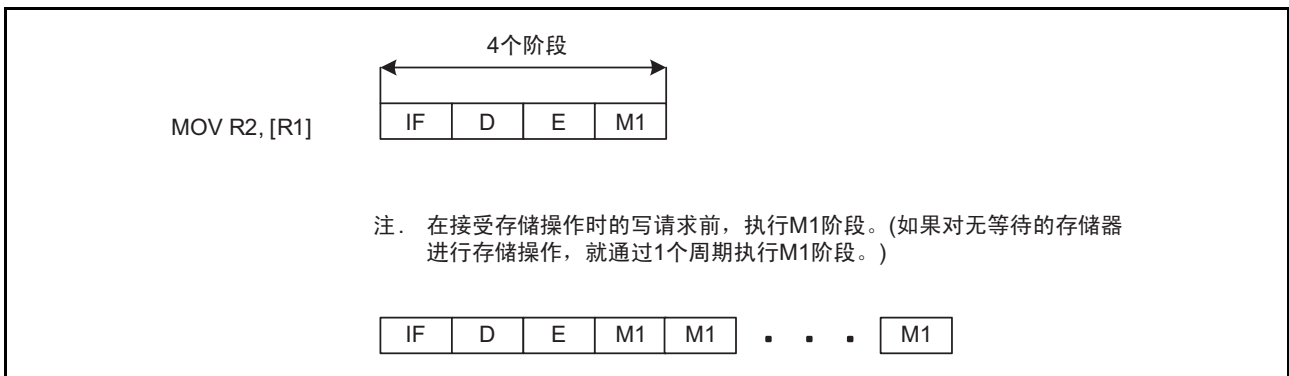


图 2.13 存储操作

## 2.8.2.2 转换为多个微操作的指令和流水线处理

转换为多个微操作的指令如下所示，周期数表示无等待存储器存取时的周期数。

表 2.14 转换为多个微操作的指令 (1/2)

指令	助记符 (当省略指令长度时, 为全长度共同的操作)	参照图	周期数
算术 / 逻辑运算指令 (存储器源操作数)	• {ADC, ADD, AND, CMP, MAX, MIN, MUL, OR, SBB, SUB, TST, XOR} "[Rs], Rd" / "dsp[Rs], Rd"	图 2.14	3
算术 / 逻辑运算指令 (除法)	• DIV "[Rs], Rd / dsp[Rs], Rd"	—	5 ~ 22
	• DIVU "[Rs], Rd / dsp[Rs], Rd"	—	4 ~ 20
算术 / 逻辑运算指令 (乘法 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)	• MOV "[Rs], [Rd]" / "dsp[Rs], [Rd]" /"[Rs], dsp[Rd]" / "dsp[Rs], [Rd]" • PUSH "[Rs]" / "dsp[Rs]"	图 2.15	3
位操作指令 (存储器源操作数)	• {BCLR, BNOT, BSET, BTST} "#IMM, [Rd]" /"#IMM, dsp[Rd]" • BMCnd "#IMM, [Rd]" / "#IMM, dsp[Rd]"	图 2.15	3
传送指令 (加载指令)	• POPC "CR"	—	吞吐量: 3 等待时间: 4 (注2)
传送指令 (多个寄存器的保存)	• PUSHM "Rs-Rs2"	—	n n 为寄存器的个数 (注3)
传送指令 (多个寄存器的恢复)	• POPM "Rs-Rs2"	—	吞吐量: n 等待时间: n+1 n 为寄存器的个数 (注2、注4)
传送指令 (寄存器之间的交换)	• XCHG "Rs, Rd"	图 2.17	2
传送指令 (存储器 - 寄存器的交换)	• XCHG "[Rs], Rd" / "dsp[Rs], Rd"	图 2.18	2
转移指令	• RTS	—	5
	• RTSD "#IMM"	—	5
	• RTSD "#IMM, Rd-Rd2"	—	吞吐量: n<5?5:1+n 等待时间: n<4?5:2+n n 为寄存器的个数 (注2)

表 2.14 转换为多个微操作的指令 (2/2)

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

注 1. floor(x): 小于等于 x 的最大整数

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

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

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

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



转换为多个微操作的指令操作如图 2.14 ~ 图 2.19 所示。

注. mop: 微操作; stall: 流水线延迟

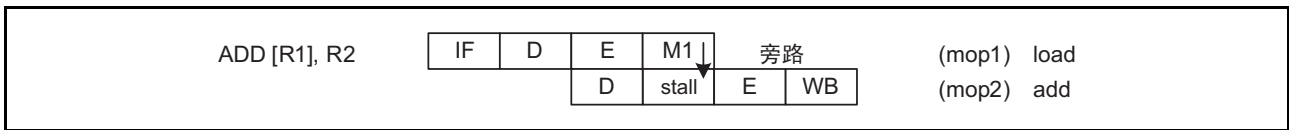


图 2.14 算术 / 逻辑运算指令 (存储器源操作数)

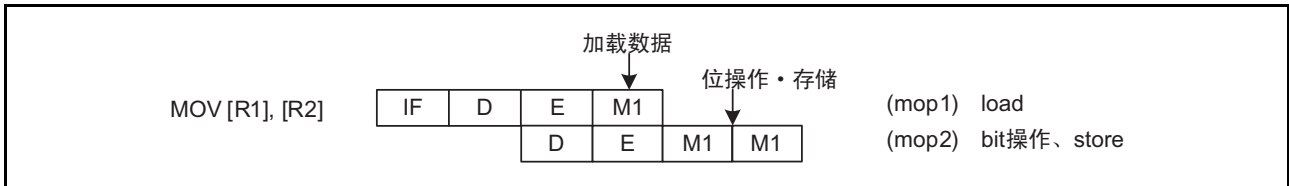


图 2.15 MOV 指令 (存储器之间的传送) 和位操作指令 (存储器源操作数)

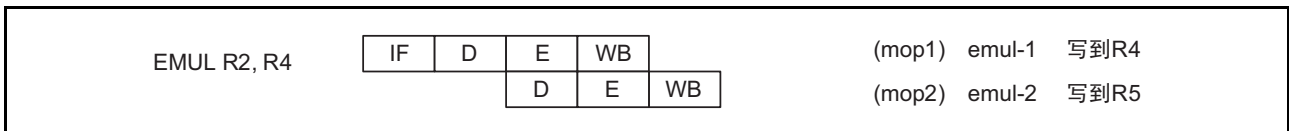


图 2.16 EMUL 指令和 EMULU 指令 (寄存器之间、寄存器 - 立即数)

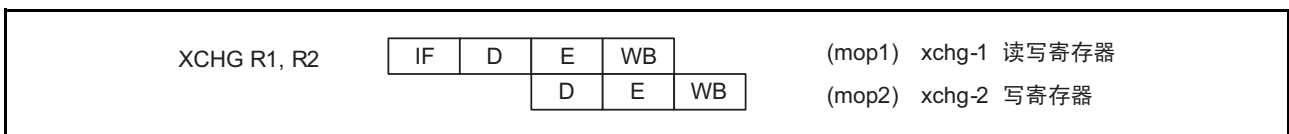


图 2.17 XCHG 指令 (寄存器)

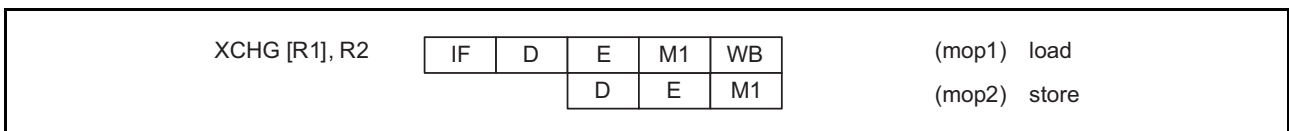


图 2.18 XCHG 指令 (存储器源操作数)

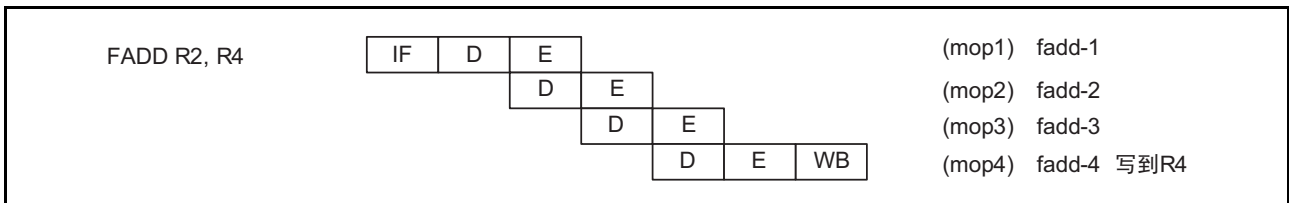


图 2.19 浮点运算指令 (寄存器之间、立即数 - 寄存器)

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

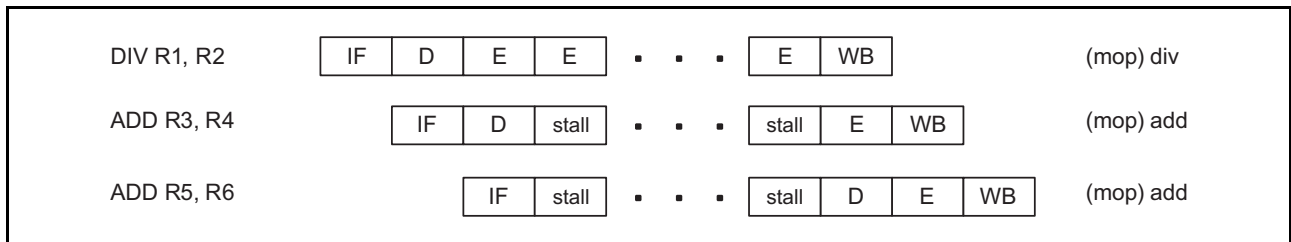


图 2.20 在执行 E 阶段过程中执行需要多个周期的指令的情况

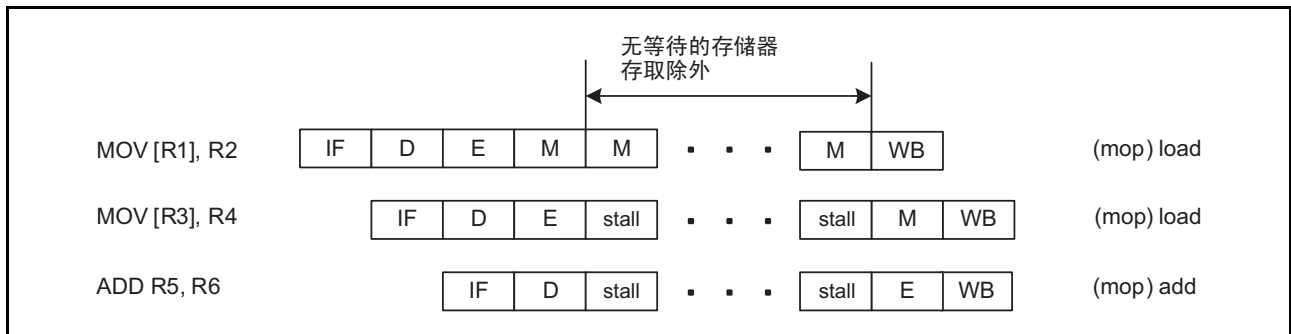


图 2.21 操作数存取没有在 1 个周期中结束的情况

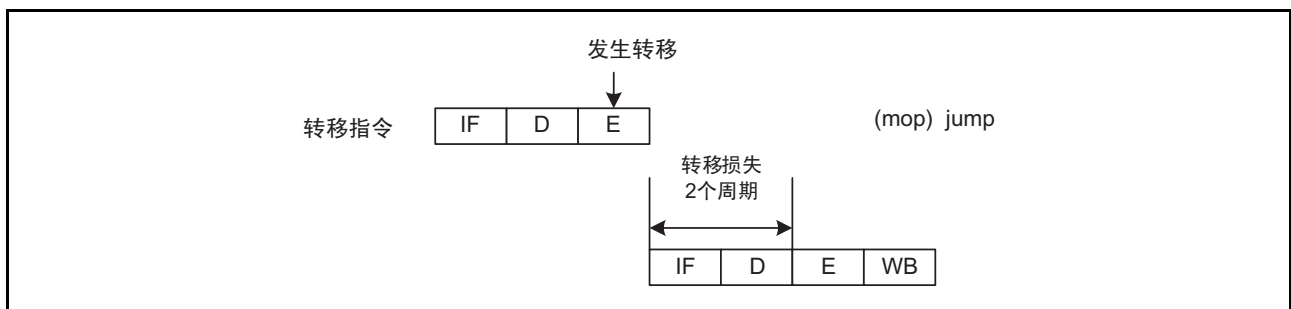


图 2.22 转移（在无条件的转移或者条件转移中条件成立的情况）

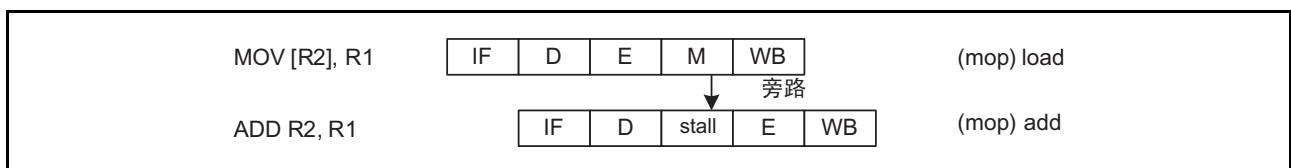


图 2.23 后续指令使用从存储器读取的操作数的情况

(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。

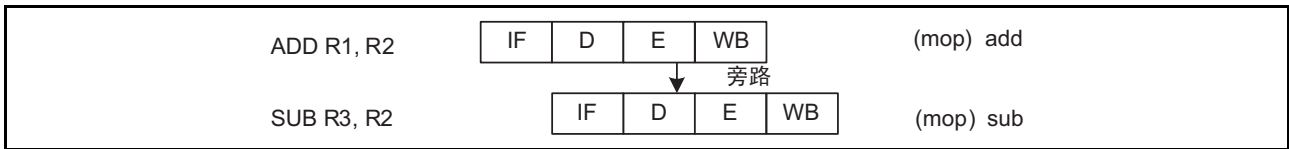


图 2.24 旁路

(b) 存储器加载和运算的 WB 阶段重叠的情况

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

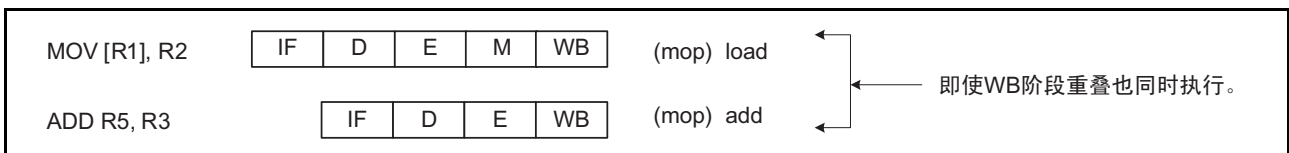


图 2.25 存储器加载和运算的 WB 阶段重叠的情况

(c) 在存储器加载结束前后续指令写相同寄存器的情况

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

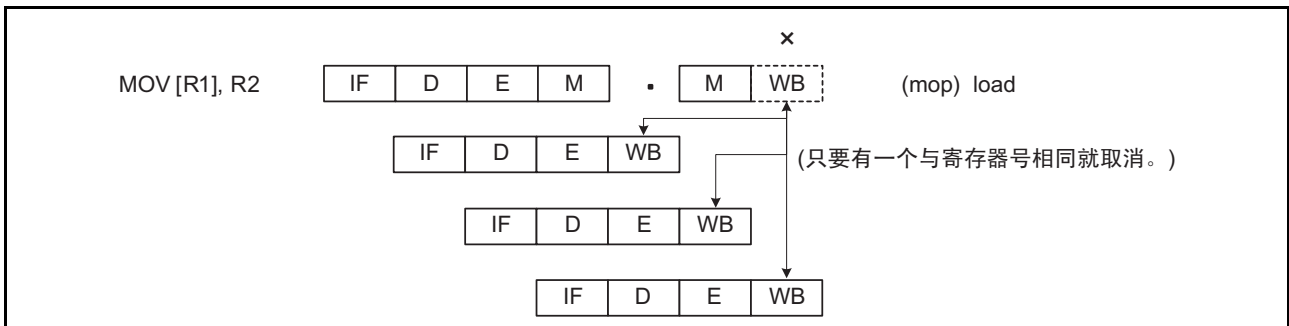


图 2.26 在结束存储器加载前，后续指令写相同寄存器的情况

(d) 后续指令不参照存储器加载的数据的情况

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

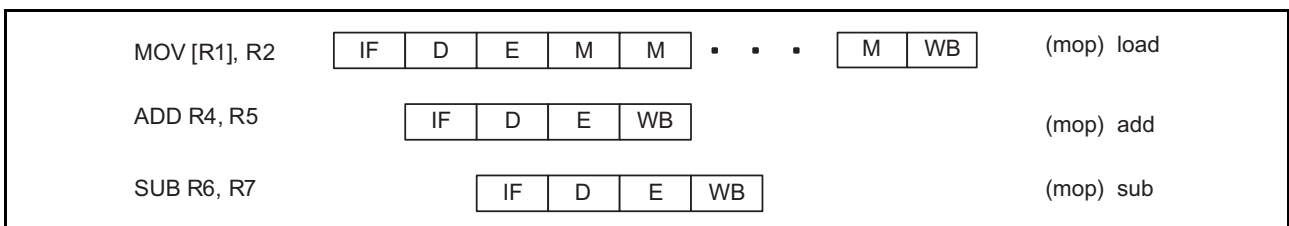


图 2.27 后续指令不参照存储器加载的数据的情况

### 2.8.3 指令处理时间的计算方法

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

- 计算周期数（参照表 2.13 和表 2.14）。
- 如果后续指令参照存储器加载的结果，存储器加载的指令周期数就为“等待时间”的周期数。否则，就为“吞吐量”的周期数。
- 如果引起取指令的延迟，就再追加周期数。
- 根据系统结构，存储器存取有时需要多个周期。RX62T 群的存储器存取周期数取决于机种。

### 2.8.4 中断响应周期数

中断响应处理的周期数如表 2.15 所示。

表 2.15 中断响应周期数

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

表 2.15 是 CPU 的存储器存取全部以无等待处理时的中断响应时间。RX62T 群装载了能无等待存取的内部 ROM 和内部 RAM。通过将程序（包括向量）分配到内部 ROM，将堆栈区分配到内部 RAM，能将中断响应周期数缩到最短。而且，异常处理程序的起始地址必须指定 8 字节边界。

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

中断接受时序取决于流水线状态。有关中断的接受时序，请参照“10.3.1 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

RX62T 群有 2 种运行模式，通过 MD1 引脚和 MD0 引脚以及系统控制寄存器 0（SYSCR0）的 ROME 位进行设定。

在设定各运行模式时，能选择 RX62T 群的字节序。通过 MDE 引脚设定字节序，有关 RX62T 群的字节序请参照“2.5 字节序”。

注. 不能在 LSI 运行时更改 MDE、MD1、MD0 引脚，也不能设定表 3.1 中没有的组合。

表 3.1 通过模式引脚选择的运行模式

模式引脚		SYSCR0 寄存器的初始状态	运行模式	内部 ROM（注1）
MD1	MD0	ROME		
0	1	1	引导模式	有效
1	1	1	单芯片模式	有效

注 1. 内部 ROM 有 ROM 和数据闪存，详细内容请参照“31. ROM（保存代码的闪存）”和“32. 数据闪存（保存数据的闪存）”。

表 3.2 通过寄存器选择的运行模式

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

注 1. 内部 ROM 有 ROM 和数据闪存，详细内容请参照“31. ROM（保存代码的闪存）”和“32. 数据闪存（保存数据的闪存）”。

表 3.3 字节序的选择

模式引脚	字节序
MDE	
0	小端法
1	大端法

## 3.2 寄存器说明

运行模式的相关寄存器一览表如表 3.4 所示。

表 3.4 运行模式的相关寄存器一览表

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

注. x: 不定值

### 3.2.1 模式监视寄存器 (MDMONR)

地址 0008 0000h

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

注 1. 因模式引脚 (MDE、MD1, MD0) 的设定而不同。

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

MDMONR 寄存器是监视模式引脚的寄存器。

## 3.2.2 模式状态寄存器 (MDSR)

地址 0008 0002h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	—	BOTS	—	—	—	IROM
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

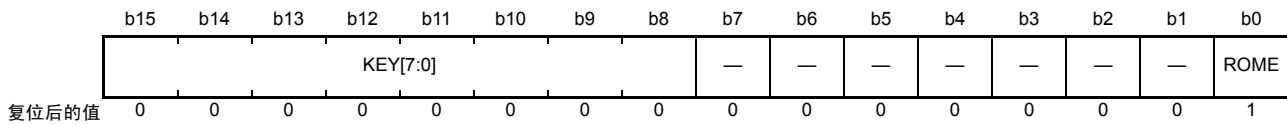
复位后的值

位	符号	位名	说明	R/W
b0	IROM	内部 ROM 的启动状态标志	0: 内部 ROM 无效。 1: 内部 ROM 有效。	R
b2-b1	—	保留位	读取值为“0”，写操作无效。	R
b3	—	保留位	读取值为“1”，写操作无效。	R
b4	BOTS	引导模式的启动标志	0: 非引导模式启动 1: 引导模式启动	R
b15-b5	—	保留位	读取值为“0”，写操作无效。	R

MDSR 寄存器是监视启动时的内部状态的寄存器。

### 3.2.3 系统控制寄存器 0 (SYSCR0)

地址 0008 0006h



位	符号	位名	说明	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	5Ah: 允许写 SYSCR0 寄存器 上述以外: 禁止写 SYSCR0 寄存器 读取值为“00h”。	R/W

SYSCR0 寄存器是选择内部 ROM 有效或者无效的寄存器。

#### ROME 位 (内部 ROM 有效位)

此位选择内部 ROM (ROM 和数据闪存) 的有效或者无效。

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

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

#### KEY[7:0] 位 (SYSCR0 键码)

这些位选择允许或者禁止写 SYSCR0 寄存器。

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



### 3.2.4 系统控制寄存器 1 (SYSCR1)

地址 0008 0008h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RAME
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

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

SYSCR1 寄存器是选择内部 RAM 的有效或者无效的寄存器。

#### RAME 位 (RAM 有效位)

此位选择内部 RAM 的有效或者无效。

一旦解除复位，RAME 位就被初始化为“1”。

不能在存取内部 RAM 时写“0”。

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

即使将 RAME 位置“0”，也保持内部 RAM 的值。

## 3.3 运行模式的说明

### 3.3.1 单芯片模式

在此模式中，内部 ROM 有效或者无效。全部 I/O 端口都能用作输入 / 输出端口。

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

### 3.3.2 引导模式

这是闪存的引导模式，除了闪存的编程 / 擦除以外，和单芯片模式中的运行相同。详细内容请参照“31. ROM (保存代码的闪存)”和“32. 数据闪存 (保存数据的闪存)”。

### 3.4 运行模式的转移

#### 3.4.1 通过模式引脚进行运行模式的转移

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

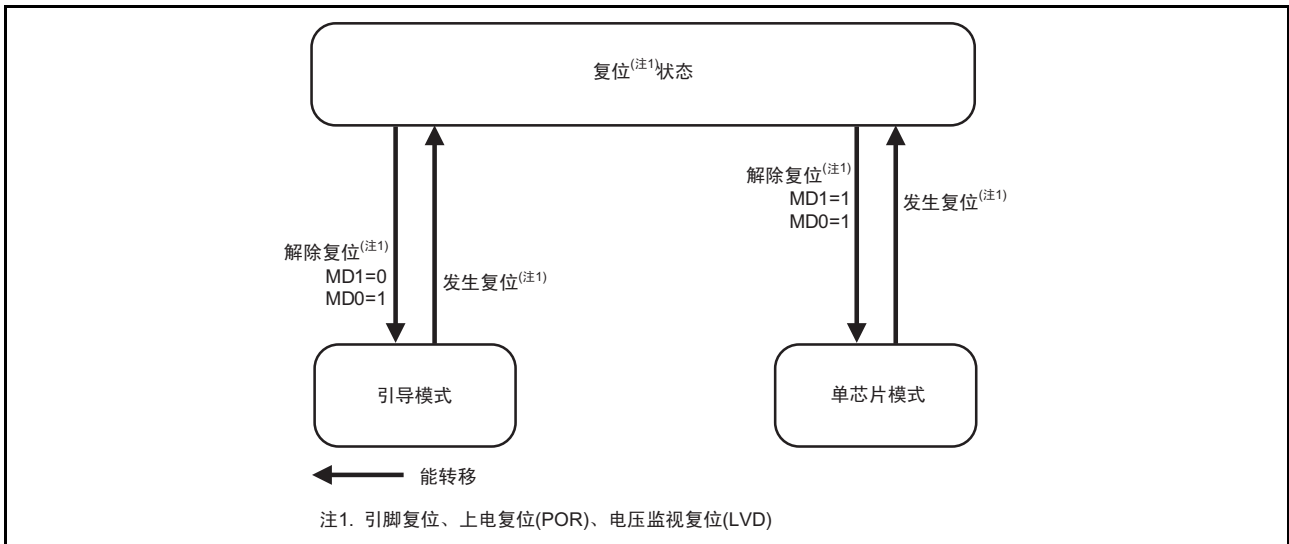


图 3.1 MD1 引脚、MD0 引脚的设定和运行模式

#### 3.4.2 通过寄存器进行运行模式的转移

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

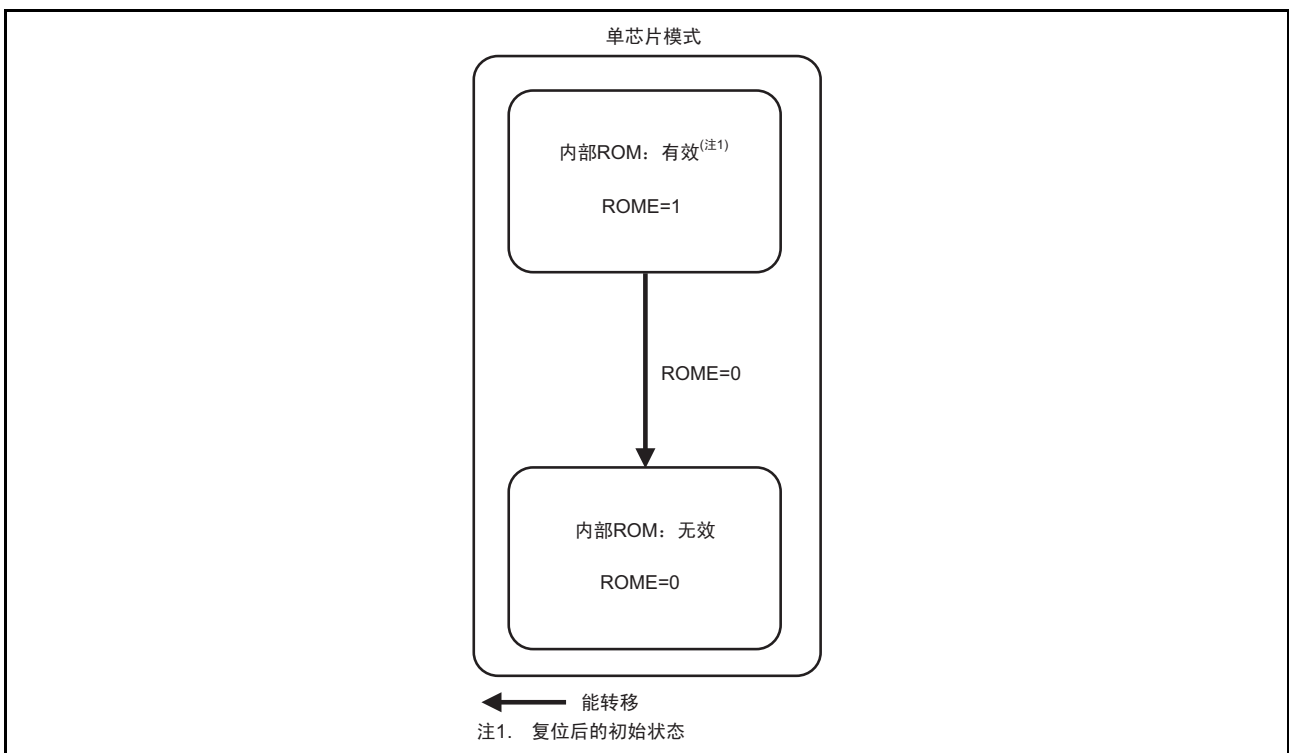


图 3.2 ROME 位的设定和运行模式

## 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。

存储器映像如图 4.1 所示。

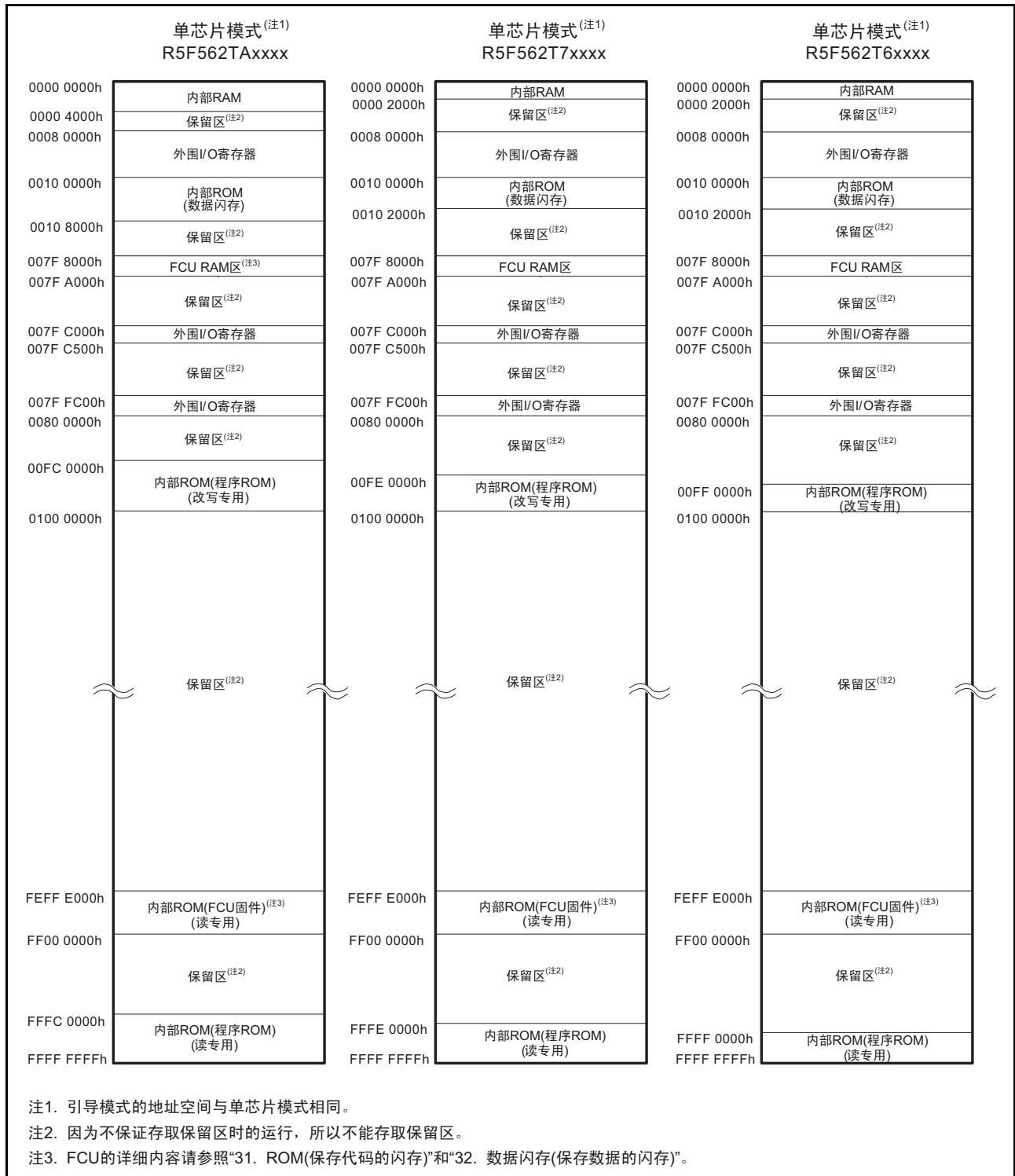


图 4.1 存储器映像 (RX62T 群)

## 5. I/O 寄存器

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

### (1) I/O 寄存器地址一览表（地址顺序）

- 从分配地址小的寄存器开始按顺序记载。
- 按模块符号进行分类。
- 存取状态数表示指定的基准时钟的状态数。
- 在I/O寄存器的区域，寄存器一览表中没有记载的地址区为保留区。禁止存取保留区。因为无法保证存取这些寄存器时的运行及其后续运行，所以禁止存取这些寄存器。
- 各寄存器指定有存取长度。禁止用指定外的存取长度进行存取。

### (2) I/O 寄存器位一览表

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

### (3) 写 I/O 寄存器时的注意事项

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

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 ]

- 将中断请求的允许位（ICU.IERm.IENj 位）（注1）置“0”，要在禁止中断的状态下执行后续指令时
- 在转移到低功耗状态的预处理后接着执行 WAIT 指令时

注 1 请参照“11.2.2 中断请求允许寄存器 m（IERm）（m=02h ~ 1Fh）”。

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

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

## [ 指令例子 ]

- I/O 寄存器为字节的情况

```
MOV.L #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。

## (4) I/O 寄存器的存取周期数

用以下表达式表示存取 I/O 寄存器时的存取周期数。(注 1)

$$\begin{aligned} \text{I/O 寄存器的存取周期数} = & \text{内部主总线 1 的总线周期数} + \\ & \text{分频时钟的同步周期数} + \\ & \text{内部外围总线 1、2、4、6 的总线周期数} \end{aligned}$$

内部外围总线 1、2、4、6 的总线周期数因存取的目标寄存器而不同。各寄存器的 I/O 寄存器的存取周期数请参照“表 5.1 I/O 寄存器地址一览表”。

在存取内部外围总线 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 的频率比以及总线的存取时序而不同。但是，因为内部主总线 1 的总线周期数加上分频时钟的同步周期数最多为 1 个 PCLK 周期，所以在表 5.1 中记载为 1 个 PCLK。

注 1. 这是 CPU 存取寄存器时，在和外部存储器的取指令或者不同总线主控器（DTC）的总线存取不发生竞争的情况下执行的周期数。

## 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 / 28）

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 0000h	SYSTEM	模式监视寄存器	MDMONR	16	16	3 个 ICLK
0008 0002h	SYSTEM	模式状态寄存器	MDSR	16	16	3 个 ICLK
0008 0006h	SYSTEM	系统控制寄存器 0	SYSCR0	16	16	3 个 ICLK
0008 0008h	SYSTEM	系统控制寄存器 1	SYSCR1	16	16	3 个 ICLK
0008 000Ch	SYSTEM	待机控制寄存器	SBYCR	16	16	3 个 ICLK
0008 0010h	SYSTEM	模块停止控制寄存器 A	MSTPCRA	32	32	3 个 ICLK
0008 0014h	SYSTEM	模块停止控制寄存器 B	MSTPCRB	32	32	3 个 ICLK
0008 0018h	SYSTEM	模块停止控制寄存器 C	MSTPCRC	32	32	3 个 ICLK
0008 0020h	SYSTEM	系统时钟控制寄存器	SCKCR	32	32	3 个 ICLK
0008 0040h	SYSTEM	振荡停止检测控制寄存器	OSTDCR	16	16	3 个 ICLK
0008 1300h	BSC	总线错误状态清除寄存器	BERCLR	8	8	2 个 ICLK
0008 1304h	BSC	总线错误监视允许寄存器	BEREN	8	8	2 个 ICLK
0008 1308h	BSC	总线错误状态寄存器 1	BERSR1	8	8	2 个 ICLK
0008 130Ah	BSC	总线错误状态寄存器 2	BERSR2	16	16	2 个 ICLK
0008 2400h	DTC	DTC 控制寄存器	DTCCR	8	8	2 个 ICLK
0008 2404h	DTC	DTC 向量基址寄存器	DTCVBR	32	32	2 个 ICLK
0008 2408h	DTC	DTC 地址模式寄存器	DTCADMOD	8	8	2 个 ICLK
0008 240Ch	DTC	DTC 模块启动寄存器	DTCST	8	8	2 个 ICLK
0008 240Eh	DTC	DTC 状态寄存器	DTCSTS	16	16	2 个 ICLK
0008 6400h	MPU	区域 0 起始页号寄存器	RSPAGE0	32	32	1 个 ICLK
0008 6404h	MPU	区域 0 结束页号寄存器	REPAGE0	32	32	1 个 ICLK
0008 6408h	MPU	区域 1 起始页号寄存器	RSPAGE1	32	32	1 个 ICLK
0008 640Ch	MPU	区域 1 结束页号寄存器	REPAGE1	32	32	1 个 ICLK
0008 6410h	MPU	区域 2 起始页号寄存器	RSPAGE2	32	32	1 个 ICLK
0008 6414h	MPU	区域 2 结束页号寄存器	REPAGE2	32	32	1 个 ICLK
0008 6418h	MPU	区域 3 起始页号寄存器	RSPAGE3	32	32	1 个 ICLK
0008 641Ch	MPU	区域 3 结束页号寄存器	REPAGE3	32	32	1 个 ICLK
0008 6420h	MPU	区域 4 起始页号寄存器	RSPAGE4	32	32	1 个 ICLK
0008 6424h	MPU	区域 4 结束页号寄存器	REPAGE4	32	32	1 个 ICLK
0008 6428h	MPU	区域 5 起始页号寄存器	RSPAGE5	32	32	1 个 ICLK
0008 642Ch	MPU	区域 5 结束页号寄存器	REPAGE5	32	32	1 个 ICLK
0008 6430h	MPU	区域 6 起始页号寄存器	RSPAGE6	32	32	1 个 ICLK
0008 6434h	MPU	区域 6 结束页号寄存器	REPAGE6	32	32	1 个 ICLK
0008 6438h	MPU	区域 7 起始页号寄存器	RSPAGE7	32	32	1 个 ICLK
0008 643Ch	MPU	区域 7 结束页号寄存器	REPAGE7	32	32	1 个 ICLK
0008 6500h	MPU	存储器保护功能有效化寄存器	MPEN	32	32	1 个 ICLK
0008 6504h	MPU	后台存取控制寄存器	MPBAC	32	32	1 个 ICLK
0008 6508h	MPU	存储器保护错误状态清除寄存器	MPECLR	32	32	1 个 ICLK
0008 650Ch	MPU	存储器保护错误状态寄存器	MPESTS	32	32	1 个 ICLK
0008 6514h	MPU	数据存储器保护错误地址寄存器	MPDEA	32	32	1 个 ICLK
0008 6520h	MPU	区域搜索地址寄存器	MPSA	32	32	1 个 ICLK
0008 6524h	MPU	区域搜索操作寄存器	MPOPS	16	16	1 个 ICLK
0008 6526h	MPU	区域无效化操作寄存器	MPOPI	16	16	1 个 ICLK
0008 6528h	MPU	指令命中区域寄存器	MHITI	32	32	1 个 ICLK
0008 652Ch	MPU	数据命中区域寄存器	MHITD	32	32	1 个 ICLK
0008 7010h	ICU	中断请求寄存器 016	IR016	8	8	2 个 ICLK

表 5.1 I/O 寄存器地址一览表 (2 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 7015h	ICU	中断请求寄存器 021	IR021	8	8	2 个 ICLK
0008 7017h	ICU	中断请求寄存器 023	IR023	8	8	2 个 ICLK
0008 701Bh	ICU	中断请求寄存器 027	IR027	8	8	2 个 ICLK
0008 701Ch	ICU	中断请求寄存器 028	IR028	8	8	2 个 ICLK
0008 701Dh	ICU	中断请求寄存器 029	IR029	8	8	2 个 ICLK
0008 701Eh	ICU	中断请求寄存器 030	IR030	8	8	2 个 ICLK
0008 701Fh	ICU	中断请求寄存器 031	IR031	8	8	2 个 ICLK
0008 702Ch	ICU	中断请求寄存器 044	IR044	8	8	2 个 ICLK
0008 702Dh	ICU	中断请求寄存器 045	IR045	8	8	2 个 ICLK
0008 702Eh	ICU	中断请求寄存器 046	IR046	8	8	2 个 ICLK
0008 702Fh	ICU	中断请求寄存器 047	IR047	8	8	2 个 ICLK
0008 7038h	ICU	中断请求寄存器 056	IR056	8	8	2 个 ICLK
0008 7039h	ICU	中断请求寄存器 057	IR057	8	8	2 个 ICLK
0008 703Ah	ICU	中断请求寄存器 058	IR058	8	8	2 个 ICLK
0008 703Bh	ICU	中断请求寄存器 059	IR059	8	8	2 个 ICLK
0008 703Ch	ICU	中断请求寄存器 060	IR060	8	8	2 个 ICLK
0008 7040h	ICU	中断请求寄存器 064	IR064	8	8	2 个 ICLK
0008 7041h	ICU	中断请求寄存器 065	IR065	8	8	2 个 ICLK
0008 7042h	ICU	中断请求寄存器 066	IR066	8	8	2 个 ICLK
0008 7043h	ICU	中断请求寄存器 067	IR067	8	8	2 个 ICLK
0008 7044h	ICU	中断请求寄存器 068	IR068	8	8	2 个 ICLK
0008 7045h	ICU	中断请求寄存器 069	IR069	8	8	2 个 ICLK
0008 7046h	ICU	中断请求寄存器 070	IR070	8	8	2 个 ICLK
0008 7047h	ICU	中断请求寄存器 071	IR071	8	8	2 个 ICLK
0008 7060h	ICU	中断请求寄存器 096	IR096	8	8	2 个 ICLK
0008 7062h	ICU	中断请求寄存器 098	IR098	8	8	2 个 ICLK
0008 7066h	ICU	中断请求寄存器 102	IR102	8	8	2 个 ICLK
0008 7067h	ICU	中断请求寄存器 103	IR103	8	8	2 个 ICLK
0008 706Ah	ICU	中断请求寄存器 106	IR106	8	8	2 个 ICLK
0008 7072h	ICU	中断请求寄存器 114	IR114	8	8	2 个 ICLK
0008 7073h	ICU	中断请求寄存器 115	IR115	8	8	2 个 ICLK
0008 7074h	ICU	中断请求寄存器 116	IR116	8	8	2 个 ICLK
0008 7075h	ICU	中断请求寄存器 117	IR117	8	8	2 个 ICLK
0008 7076h	ICU	中断请求寄存器 118	IR118	8	8	2 个 ICLK
0008 7077h	ICU	中断请求寄存器 119	IR119	8	8	2 个 ICLK
0008 7078h	ICU	中断请求寄存器 120	IR120	8	8	2 个 ICLK
0008 7079h	ICU	中断请求寄存器 121	IR121	8	8	2 个 ICLK
0008 707Ah	ICU	中断请求寄存器 122	IR122	8	8	2 个 ICLK
0008 707Bh	ICU	中断请求寄存器 123	IR123	8	8	2 个 ICLK
0008 707Ch	ICU	中断请求寄存器 124	IR124	8	8	2 个 ICLK
0008 707Dh	ICU	中断请求寄存器 125	IR125	8	8	2 个 ICLK
0008 707Eh	ICU	中断请求寄存器 126	IR126	8	8	2 个 ICLK
0008 707Fh	ICU	中断请求寄存器 127	IR127	8	8	2 个 ICLK
0008 7080h	ICU	中断请求寄存器 128	IR128	8	8	2 个 ICLK
0008 7081h	ICU	中断请求寄存器 129	IR129	8	8	2 个 ICLK
0008 7082h	ICU	中断请求寄存器 130	IR130	8	8	2 个 ICLK
0008 7083h	ICU	中断请求寄存器 131	IR131	8	8	2 个 ICLK
0008 7084h	ICU	中断请求寄存器 132	IR132	8	8	2 个 ICLK

表 5.1 I/O 寄存器地址一览表 (3 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 7085h	ICU	中断请求寄存器 133	IR133	8	8	2 个 ICLK
0008 7086h	ICU	中断请求寄存器 134	IR134	8	8	2 个 ICLK
0008 7087h	ICU	中断请求寄存器 135	IR135	8	8	2 个 ICLK
0008 7088h	ICU	中断请求寄存器 136	IR136	8	8	2 个 ICLK
0008 7089h	ICU	中断请求寄存器 137	IR137	8	8	2 个 ICLK
0008 708Ah	ICU	中断请求寄存器 138	IR138	8	8	2 个 ICLK
0008 708Bh	ICU	中断请求寄存器 139	IR139	8	8	2 个 ICLK
0008 708Ch	ICU	中断请求寄存器 140	IR140	8	8	2 个 ICLK
0008 708Dh	ICU	中断请求寄存器 141	IR141	8	8	2 个 ICLK
0008 708Eh	ICU	中断请求寄存器 142	IR142	8	8	2 个 ICLK
0008 708Fh	ICU	中断请求寄存器 143	IR143	8	8	2 个 ICLK
0008 7090h	ICU	中断请求寄存器 144	IR144	8	8	2 个 ICLK
0008 7091h	ICU	中断请求寄存器 145	IR145	8	8	2 个 ICLK
0008 7092h	ICU	中断请求寄存器 146	IR146	8	8	2 个 ICLK
0008 7095h	ICU	中断请求寄存器 149	IR149	8	8	2 个 ICLK
0008 7096h	ICU	中断请求寄存器 150	IR150	8	8	2 个 ICLK
0008 7097h	ICU	中断请求寄存器 151	IR151	8	8	2 个 ICLK
0008 7098h	ICU	中断请求寄存器 152	IR152	8	8	2 个 ICLK
0008 7099h	ICU	中断请求寄存器 153	IR153	8	8	2 个 ICLK
0008 70AAh	ICU	中断请求寄存器 170	IR170	8	8	2 个 ICLK
0008 70ABh	ICU	中断请求寄存器 171	IR171	8	8	2 个 ICLK
0008 70ACh	ICU	中断请求寄存器 172	IR172	8	8	2 个 ICLK
0008 70ADh	ICU	中断请求寄存器 173	IR173	8	8	2 个 ICLK
0008 70AEh	ICU	中断请求寄存器 174	IR174	8	8	2 个 ICLK
0008 70AFh	ICU	中断请求寄存器 175	IR175	8	8	2 个 ICLK
0008 70B0h	ICU	中断请求寄存器 176	IR176	8	8	2 个 ICLK
0008 70B1h	ICU	中断请求寄存器 177	IR177	8	8	2 个 ICLK
0008 70B2h	ICU	中断请求寄存器 178	IR178	8	8	2 个 ICLK
0008 70B3h	ICU	中断请求寄存器 179	IR179	8	8	2 个 ICLK
0008 70B4h	ICU	中断请求寄存器 180	IR180	8	8	2 个 ICLK
0008 70B5h	ICU	中断请求寄存器 181	IR181	8	8	2 个 ICLK
0008 70B6h	ICU	中断请求寄存器 182	IR182	8	8	2 个 ICLK
0008 70B7h	ICU	中断请求寄存器 183	IR183	8	8	2 个 ICLK
0008 70B8h	ICU	中断请求寄存器 184	IR184	8	8	2 个 ICLK
0008 70BAh	ICU	中断请求寄存器 186	IR186	8	8	2 个 ICLK
0008 70BBh	ICU	中断请求寄存器 187	IR187	8	8	2 个 ICLK
0008 70BCh	ICU	中断请求寄存器 188	IR188	8	8	2 个 ICLK
0008 70BDh	ICU	中断请求寄存器 189	IR189	8	8	2 个 ICLK
0008 70BEh	ICU	中断请求寄存器 190	IR190	8	8	2 个 ICLK
0008 70C0h	ICU	中断请求寄存器 192	IR192	8	8	2 个 ICLK
0008 70C1h	ICU	中断请求寄存器 193	IR193	8	8	2 个 ICLK
0008 70C2h	ICU	中断请求寄存器 194	IR194	8	8	2 个 ICLK
0008 70C3h	ICU	中断请求寄存器 195	IR195	8	8	2 个 ICLK
0008 70C4h	ICU	中断请求寄存器 196	IR196	8	8	2 个 ICLK
0008 70D6h	ICU	中断请求寄存器 214	IR214	8	8	2 个 ICLK
0008 70D7h	ICU	中断请求寄存器 215	IR215	8	8	2 个 ICLK
0008 70D8h	ICU	中断请求寄存器 216	IR216	8	8	2 个 ICLK
0008 70D9h	ICU	中断请求寄存器 217	IR217	8	8	2 个 ICLK



表 5.1 I/O 寄存器地址一览表 (4 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 70DAh	ICU	中断请求寄存器 218	IR218	8	8	2 个 ICLK
0008 70DBh	ICU	中断请求寄存器 219	IR219	8	8	2 个 ICLK
0008 70DCh	ICU	中断请求寄存器 220	IR220	8	8	2 个 ICLK
0008 70DDh	ICU	中断请求寄存器 221	IR221	8	8	2 个 ICLK
0008 70DEh	ICU	中断请求寄存器 222	IR222	8	8	2 个 ICLK
0008 70DFh	ICU	中断请求寄存器 223	IR223	8	8	2 个 ICLK
0008 70E0h	ICU	中断请求寄存器 224	IR224	8	8	2 个 ICLK
0008 70E1h	ICU	中断请求寄存器 225	IR225	8	8	2 个 ICLK
0008 70F6h	ICU	中断请求寄存器 246	IR246	8	8	2 个 ICLK
0008 70F7h	ICU	中断请求寄存器 247	IR247	8	8	2 个 ICLK
0008 70F8h	ICU	中断请求寄存器 248	IR248	8	8	2 个 ICLK
0008 70F9h	ICU	中断请求寄存器 249	IR249	8	8	2 个 ICLK
0008 70FEh	ICU	中断请求寄存器 254	IR254	8	8	2 个 ICLK
0008 711Bh	ICU	DTC 启动允许寄存器 027	DT CER027	8	8	2 个 ICLK
0008 711Ch	ICU	DTC 启动允许寄存器 028	DT CER028	8	8	2 个 ICLK
0008 711Dh	ICU	DTC 启动允许寄存器 029	DT CER029	8	8	2 个 ICLK
0008 711Eh	ICU	DTC 启动允许寄存器 030	DT CER030	8	8	2 个 ICLK
0008 711Fh	ICU	DTC 启动允许寄存器 031	DT CER031	8	8	2 个 ICLK
0008 712Dh	ICU	DTC 启动允许寄存器 045	DT CER045	8	8	2 个 ICLK
0008 712Eh	ICU	DTC 启动允许寄存器 046	DT CER046	8	8	2 个 ICLK
0008 7140h	ICU	DTC 启动允许寄存器 064	DT CER064	8	8	2 个 ICLK
0008 7141h	ICU	DTC 启动允许寄存器 065	DT CER065	8	8	2 个 ICLK
0008 7142h	ICU	DTC 启动允许寄存器 066	DT CER066	8	8	2 个 ICLK
0008 7143h	ICU	DTC 启动允许寄存器 067	DT CER067	8	8	2 个 ICLK
0008 7144h	ICU	DTC 启动允许寄存器 068	DT CER068	8	8	2 个 ICLK
0008 7145h	ICU	DTC 启动允许寄存器 069	DT CER069	8	8	2 个 ICLK
0008 7146h	ICU	DTC 启动允许寄存器 070	DT CER070	8	8	2 个 ICLK
0008 7147h	ICU	DTC 启动允许寄存器 071	DT CER071	8	8	2 个 ICLK
0008 7162h	ICU	DTC 启动允许寄存器 098	DT CER098	8	8	2 个 ICLK
0008 7166h	ICU	DTC 启动允许寄存器 102	DT CER102	8	8	2 个 ICLK
0008 7167h	ICU	DTC 启动允许寄存器 103	DT CER103	8	8	2 个 ICLK
0008 716Ah	ICU	DTC 启动允许寄存器 106	DT CER106	8	8	2 个 ICLK
0008 7172h	ICU	DTC 启动允许寄存器 114	DT CER114	8	8	2 个 ICLK
0008 7173h	ICU	DTC 启动允许寄存器 115	DT CER115	8	8	2 个 ICLK
0008 7174h	ICU	DTC 启动允许寄存器 116	DT CER116	8	8	2 个 ICLK
0008 7175h	ICU	DTC 启动允许寄存器 117	DT CER117	8	8	2 个 ICLK
0008 7179h	ICU	DTC 启动允许寄存器 121	DT CER121	8	8	2 个 ICLK
0008 717Ah	ICU	DTC 启动允许寄存器 122	DT CER122	8	8	2 个 ICLK
0008 717Dh	ICU	DTC 启动允许寄存器 125	DT CER125	8	8	2 个 ICLK
0008 717Eh	ICU	DTC 启动允许寄存器 126	DT CER126	8	8	2 个 ICLK
0008 7181h	ICU	DTC 启动允许寄存器 129	DT CER129	8	8	2 个 ICLK
0008 7182h	ICU	DTC 启动允许寄存器 130	DT CER130	8	8	2 个 ICLK
0008 7183h	ICU	DTC 启动允许寄存器 131	DT CER131	8	8	2 个 ICLK
0008 7184h	ICU	DTC 启动允许寄存器 132	DT CER132	8	8	2 个 ICLK
0008 7186h	ICU	DTC 启动允许寄存器 134	DT CER134	8	8	2 个 ICLK
0008 7187h	ICU	DTC 启动允许寄存器 135	DT CER135	8	8	2 个 ICLK
0008 7188h	ICU	DTC 启动允许寄存器 136	DT CER136	8	8	2 个 ICLK
0008 7189h	ICU	DTC 启动允许寄存器 137	DT CER137	8	8	2 个 ICLK

表 5.1 I/O 寄存器地址一览表 (5 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 718Ah	ICU	DTC 启动允许寄存器 138	DTCER138	8	8	2 个 ICLK
0008 718Bh	ICU	DTC 启动允许寄存器 139	DTCER139	8	8	2 个 ICLK
0008 718Ch	ICU	DTC 启动允许寄存器 140	DTCER140	8	8	2 个 ICLK
0008 718Dh	ICU	DTC 启动允许寄存器 141	DTCER141	8	8	2 个 ICLK
0008 718Eh	ICU	DTC 启动允许寄存器 142	DTCER142	8	8	2 个 ICLK
0008 718Fh	ICU	DTC 启动允许寄存器 143	DTCER143	8	8	2 个 ICLK
0008 7190h	ICU	DTC 启动允许寄存器 144	DTCER144	8	8	2 个 ICLK
0008 7191h	ICU	DTC 启动允许寄存器 145	DTCER145	8	8	2 个 ICLK
0008 7195h	ICU	DTC 启动允许寄存器 149	DTCER149	8	8	2 个 ICLK
0008 7196h	ICU	DTC 启动允许寄存器 150	DTCER150	8	8	2 个 ICLK
0008 7197h	ICU	DTC 启动允许寄存器 151	DTCER151	8	8	2 个 ICLK
0008 7198h	ICU	DTC 启动允许寄存器 152	DTCER152	8	8	2 个 ICLK
0008 7199h	ICU	DTC 启动允许寄存器 153	DTCER153	8	8	2 个 ICLK
0008 71AEh	ICU	DTC 启动允许寄存器 174	DTCER174	8	8	2 个 ICLK
0008 71AFh	ICU	DTC 启动允许寄存器 175	DTCER175	8	8	2 个 ICLK
0008 71B0h	ICU	DTC 启动允许寄存器 176	DTCER176	8	8	2 个 ICLK
0008 71B1h	ICU	DTC 启动允许寄存器 177	DTCER177	8	8	2 个 ICLK
0008 71B2h	ICU	DTC 启动允许寄存器 178	DTCER178	8	8	2 个 ICLK
0008 71B3h	ICU	DTC 启动允许寄存器 179	DTCER179	8	8	2 个 ICLK
0008 71B4h	ICU	DTC 启动允许寄存器 180	DTCER180	8	8	2 个 ICLK
0008 71B5h	ICU	DTC 启动允许寄存器 181	DTCER181	8	8	2 个 ICLK
0008 71B6h	ICU	DTC 启动允许寄存器 182	DTCER182	8	8	2 个 ICLK
0008 71B7h	ICU	DTC 启动允许寄存器 183	DTCER183	8	8	2 个 ICLK
0008 71B8h	ICU	DTC 启动允许寄存器 184	DTCER184	8	8	2 个 ICLK
0008 71BAh	ICU	DTC 启动允许寄存器 186	DTCER186	8	8	2 个 ICLK
0008 71BBh	ICU	DTC 启动允许寄存器 187	DTCER187	8	8	2 个 ICLK
0008 71BCh	ICU	DTC 启动允许寄存器 188	DTCER188	8	8	2 个 ICLK
0008 71BDh	ICU	DTC 启动允许寄存器 189	DTCER189	8	8	2 个 ICLK
0008 71BEh	ICU	DTC 启动允许寄存器 190	DTCER190	8	8	2 个 ICLK
0008 71C0h	ICU	DTC 启动允许寄存器 192	DTCER192	8	8	2 个 ICLK
0008 71C1h	ICU	DTC 启动允许寄存器 193	DTCER193	8	8	2 个 ICLK
0008 71C2h	ICU	DTC 启动允许寄存器 194	DTCER194	8	8	2 个 ICLK
0008 71C3h	ICU	DTC 启动允许寄存器 195	DTCER195	8	8	2 个 ICLK
0008 71C4h	ICU	DTC 启动允许寄存器 196	DTCER196	8	8	2 个 ICLK
0008 71D7h	ICU	DTC 启动允许寄存器 215	DTCER215	8	8	2 个 ICLK
0008 71D8h	ICU	DTC 启动允许寄存器 216	DTCER216	8	8	2 个 ICLK
0008 71DBh	ICU	DTC 启动允许寄存器 219	DTCER219	8	8	2 个 ICLK
0008 71DCh	ICU	DTC 启动允许寄存器 220	DTCER220	8	8	2 个 ICLK
0008 71DFh	ICU	DTC 启动允许寄存器 223	DTCER223	8	8	2 个 ICLK
0008 71E0h	ICU	DTC 启动允许寄存器 224	DTCER224	8	8	2 个 ICLK
0008 71F7h	ICU	DTC 启动允许寄存器 247	DTCER247	8	8	2 个 ICLK
0008 71F8h	ICU	DTC 启动允许寄存器 248	DTCER248	8	8	2 个 ICLK
0008 71FEh	ICU	DTC 启动允许寄存器 254	DTCER254	8	8	2 个 ICLK
0008 7202h	ICU	中断请求允许寄存器 02	IER02	8	8	2 个 ICLK
0008 7203h	ICU	中断请求允许寄存器 03	IER03	8	8	2 个 ICLK
0008 7205h	ICU	中断请求允许寄存器 05	IER05	8	8	2 个 ICLK
0008 7207h	ICU	中断请求允许寄存器 07	IER07	8	8	2 个 ICLK
0008 7208h	ICU	中断请求允许寄存器 08	IER08	8	8	2 个 ICLK

表 5.1 I/O 寄存器地址一览表 (6 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 720Ch	ICU	中断请求允许寄存器 0C	IER0C	8	8	2 个 ICLK
0008 720Dh	ICU	中断请求允许寄存器 0D	IER0D	8	8	2 个 ICLK
0008 720Eh	ICU	中断请求允许寄存器 0E	IER0E	8	8	2 个 ICLK
0008 720Fh	ICU	中断请求允许寄存器 0F	IER0F	8	8	2 个 ICLK
0008 7210h	ICU	中断请求允许寄存器 10	IER10	8	8	2 个 ICLK
0008 7211h	ICU	中断请求允许寄存器 11	IER11	8	8	2 个 ICLK
0008 7212h	ICU	中断请求允许寄存器 12	IER12	8	8	2 个 ICLK
0008 7213h	ICU	中断请求允许寄存器 13	IER13	8	8	2 个 ICLK
0008 7215h	ICU	中断请求允许寄存器 15	IER15	8	8	2 个 ICLK
0008 7216h	ICU	中断请求允许寄存器 16	IER16	8	8	2 个 ICLK
0008 7217h	ICU	中断请求允许寄存器 17	IER17	8	8	2 个 ICLK
0008 7218h	ICU	中断请求允许寄存器 18	IER18	8	8	2 个 ICLK
0008 721Ah	ICU	中断请求允许寄存器 1A	IER1A	8	8	2 个 ICLK
0008 721Bh	ICU	中断请求允许寄存器 1B	IER1B	8	8	2 个 ICLK
0008 721Ch	ICU	中断请求允许寄存器 1C	IER1C	8	8	2 个 ICLK
0008 721Eh	ICU	中断请求允许寄存器 1E	IER1E	8	8	2 个 ICLK
0008 721Fh	ICU	中断请求允许寄存器 1F	IER1F	8	8	2 个 ICLK
0008 72E0h	ICU	软件中断启动寄存器	SWINTR	8	8	2 个 ICLK
0008 72F0h	ICU	高速中断设定寄存器	FIR	16	16	2 个 ICLK
0008 7300h	ICU	中断源优先级寄存器 00	IPR00	8	8	2 个 ICLK
0008 7301h	ICU	中断源优先级寄存器 01	IPR01	8	8	2 个 ICLK
0008 7302h	ICU	中断源优先级寄存器 02	IPR02	8	8	2 个 ICLK
0008 7303h	ICU	中断源优先级寄存器 03	IPR03	8	8	2 个 ICLK
0008 7304h	ICU	中断源优先级寄存器 04	IPR04	8	8	2 个 ICLK
0008 7305h	ICU	中断源优先级寄存器 05	IPR05	8	8	2 个 ICLK
0008 7306h	ICU	中断源优先级寄存器 06	IPR06	8	8	2 个 ICLK
0008 7307h	ICU	中断源优先级寄存器 07	IPR07	8	8	2 个 ICLK
0008 7314h	ICU	中断源优先级寄存器 14	IPR14	8	8	2 个 ICLK
0008 7318h	ICU	中断源优先级寄存器 18	IPR18	8	8	2 个 ICLK
0008 7320h	ICU	中断源优先级寄存器 20	IPR20	8	8	2 个 ICLK
0008 7321h	ICU	中断源优先级寄存器 21	IPR21	8	8	2 个 ICLK
0008 7322h	ICU	中断源优先级寄存器 22	IPR22	8	8	2 个 ICLK
0008 7323h	ICU	中断源优先级寄存器 23	IPR23	8	8	2 个 ICLK
0008 7324h	ICU	中断源优先级寄存器 24	IPR24	8	8	2 个 ICLK
0008 7325h	ICU	中断源优先级寄存器 25	IPR25	8	8	2 个 ICLK
0008 7326h	ICU	中断源优先级寄存器 26	IPR26	8	8	2 个 ICLK
0008 7327h	ICU	中断源优先级寄存器 27	IPR27	8	8	2 个 ICLK
0008 7340h	ICU	中断源优先级寄存器 40	IPR40	8	8	2 个 ICLK
0008 7344h	ICU	中断源优先级寄存器 44	IPR44	8	8	2 个 ICLK
0008 7348h	ICU	中断源优先级寄存器 48	IPR48	8	8	2 个 ICLK
0008 7349h	ICU	中断源优先级寄存器 49	IPR49	8	8	2 个 ICLK
0008 7351h	ICU	中断源优先级寄存器 51	IPR51	8	8	2 个 ICLK
0008 7352h	ICU	中断源优先级寄存器 52	IPR52	8	8	2 个 ICLK
0008 7353h	ICU	中断源优先级寄存器 53	IPR53	8	8	2 个 ICLK
0008 7354h	ICU	中断源优先级寄存器 54	IPR54	8	8	2 个 ICLK
0008 7355h	ICU	中断源优先级寄存器 55	IPR55	8	8	2 个 ICLK
0008 7356h	ICU	中断源优先级寄存器 56	IPR56	8	8	2 个 ICLK
0008 7357h	ICU	中断源优先级寄存器 57	IPR57	8	8	2 个 ICLK

表 5.1 I/O 寄存器地址一览表 (7 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 7358h	ICU	中断源优先级寄存器 58	IPR58	8	8	2 个 ICLK
0008 7359h	ICU	中断源优先级寄存器 59	IPR59	8	8	2 个 ICLK
0008 735Ah	ICU	中断源优先级寄存器 5A	IPR5A	8	8	2 个 ICLK
0008 735Bh	ICU	中断源优先级寄存器 5B	IPR5B	8	8	2 个 ICLK
0008 735Ch	ICU	中断源优先级寄存器 5C	IPR5C	8	8	2 个 ICLK
0008 735Dh	ICU	中断源优先级寄存器 5D	IPR5D	8	8	2 个 ICLK
0008 735Eh	ICU	中断源优先级寄存器 5E	IPR5E	8	8	2 个 ICLK
0008 735Fh	ICU	中断源优先级寄存器 5F	IPR5F	8	8	2 个 ICLK
0008 7360h	ICU	中断源优先级寄存器 60	IPR60	8	8	2 个 ICLK
0008 7367h	ICU	中断源优先级寄存器 67	IPR67	8	8	2 个 ICLK
0008 7368h	ICU	中断源优先级寄存器 68	IPR68	8	8	2 个 ICLK
0008 7369h	ICU	中断源优先级寄存器 69	IPR69	8	8	2 个 ICLK
0008 736Ah	ICU	中断源优先级寄存器 6A	IPR6A	8	8	2 个 ICLK
0008 736Bh	ICU	中断源优先级寄存器 6B	IPR6B	8	8	2 个 ICLK
0008 736Ch	ICU	中断源优先级寄存器 6C	IPR6C	8	8	2 个 ICLK
0008 736Dh	ICU	中断源优先级寄存器 6D	IPR6D	8	8	2 个 ICLK
0008 736Eh	ICU	中断源优先级寄存器 6E	IPR6E	8	8	2 个 ICLK
0008 736Fh	ICU	中断源优先级寄存器 6F	IPR6F	8	8	2 个 ICLK
0008 7380h	ICU	中断源优先级寄存器 80	IPR80	8	8	2 个 ICLK
0008 7381h	ICU	中断源优先级寄存器 81	IPR81	8	8	2 个 ICLK
0008 7382h	ICU	中断源优先级寄存器 82	IPR82	8	8	2 个 ICLK
0008 7388h	ICU	中断源优先级寄存器 88	IPR88	8	8	2 个 ICLK
0008 7389h	ICU	中断源优先级寄存器 89	IPR89	8	8	2 个 ICLK
0008 738Ah	ICU	中断源优先级寄存器 8A	IPR8A	8	8	2 个 ICLK
0008 738Bh	ICU	中断源优先级寄存器 8B	IPR8B	8	8	2 个 ICLK
0008 7390h	ICU	中断源优先级寄存器 90	IPR90	8	8	2 个 ICLK
0008 7500h	ICU	IRQ 控制寄存器 0	IRQCR0	8	8	2 个 ICLK
0008 7501h	ICU	IRQ 控制寄存器 1	IRQCR1	8	8	2 个 ICLK
0008 7502h	ICU	IRQ 控制寄存器 2	IRQCR2	8	8	2 个 ICLK
0008 7503h	ICU	IRQ 控制寄存器 3	IRQCR3	8	8	2 个 ICLK
0008 7504h	ICU	IRQ 控制寄存器 4	IRQCR4	8	8	2 个 ICLK
0008 7505h	ICU	IRQ 控制寄存器 5	IRQCR5	8	8	2 个 ICLK
0008 7506h	ICU	IRQ 控制寄存器 6	IRQCR6	8	8	2 个 ICLK
0008 7507h	ICU	IRQ 控制寄存器 7	IRQCR7	8	8	2 个 ICLK
0008 7580h	ICU	非屏蔽中断状态寄存器	NMISR	8	8	2 个 ICLK
0008 7581h	ICU	非屏蔽中断允许寄存器	NMIER	8	8	2 个 ICLK
0008 7582h	ICU	非屏蔽中断清除寄存器	NMICLR	8	8	2 个 ICLK
0008 7583h	ICU	NMI 引脚中断控制寄存器	NMICR	8	8	2 个 ICLK
0008 8000h	CMT	比较匹配定时器的启动寄存器 0	CMSTR0	16	16	2~3 个 PCLK (注5)
0008 8002h	CMT0	比较匹配定时器的控制寄存器	CMCR	16	16	2~3 个 PCLK (注5)
0008 8004h	CMT0	比较匹配定时器的计数器	CMCNT	16	16	2~3 个 PCLK (注5)
0008 8006h	CMT0	比较匹配定时器的常数寄存器	CMCOR	16	16	2~3 个 PCLK (注5)
0008 8008h	CMT1	比较匹配定时器的控制寄存器	CMCR	16	16	2~3 个 PCLK (注5)
0008 800Ah	CMT1	比较匹配定时器的计数器	CMCNT	16	16	2~3 个 PCLK (注5)

表 5.1 I/O 寄存器地址一览表 (8 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 800Ch	CMT1	比较匹配定时器的常数寄存器	CMCOR	16	16	2~3个PCLK (注10)
0008 8010h	CMT	比较匹配定时器的启动寄存器 1	CMSTR1	16	16	2~3个PCLK (注5)
0008 8012h	CMT2	比较匹配定时器的控制寄存器	CMCR	16	16	2~3个PCLK (注5)
0008 8014h	CMT2	比较匹配定时器的计数器	CMCNT	16	16	2~3个PCLK (注5)
0008 8016h	CMT2	比较匹配定时器的常数寄存器	CMCOR	16	16	2~3个PCLK (注5)
0008 8018h	CMT3	比较匹配定时器的控制寄存器	CMCR	16	16	2~3个PCLK (注5)
0008 801Ah	CMT3	比较匹配定时器的计数器	CMCNT	16	16	2~3个PCLK (注5)
0008 801Ch	CMT3	比较匹配定时器的常数寄存器	CMCOR	16	16	2~3个PCLK (注5)
0008 8028h	WDT	定时器的控制 / 状态寄存器	TCSR	8	8	2~3个PCLK (注5)
0008 8028h	WDT	写窗口 A 寄存器	WINA	16	16	2~3个PCLK (注5)
0008 8029h	WDT	定时器的计数器	TCNT	8	8	2~3个PCLK (注5)
0008 802Ah	WDT	写窗口 B 寄存器	WINB	16	16	2~3个PCLK (注5)
0008 802Bh	WDT	复位控制 / 状态寄存器	RSTCSR	8	8	2~3个PCLK (注5)
0008 8030h	IWDT	IWDT 刷新寄存器	IWDTRR	8	8	2~3个PCLK (注5)
0008 8032h	IWDT	IWDT 控制寄存器	IWDTCR	16	16	2~3个PCLK (注5)
0008 8034h	IWDT	IWDT 状态寄存器	IWDTSR	16	16	2~3个PCLK (注5)
0008 8040h	AD0	A/D 数据寄存器 A	ADDRA	16	16	2~3个PCLK (注5)
0008 8042h	AD0	A/D 数据寄存器 B	ADDRB	16	16	2~3个PCLK (注5)
0008 8044h	AD0	A/D 数据寄存器 C	ADDRC	16	16	2~3个PCLK (注5)
0008 8046h	AD0	A/D 数据寄存器 D	ADDRD	16	16	2~3个PCLK (注5)
0008 8048h	AD0	A/D 数据寄存器 E	ADDRE	16	16	2~3个PCLK (注5)
0008 804Ah	AD0	A/D 数据寄存器 F	ADDRF	16	16	2~3个PCLK (注5)
0008 804Ch	AD0	A/D 数据寄存器 G	ADDRG	16	16	2~3个PCLK (注5)
0008 804Eh	AD0	A/D 数据寄存器 H	ADDRH	16	16	2~3个PCLK (注5)
0008 8050h	AD0	A/D 控制 / 状态寄存器	ADCSR	8	8	2~3个PCLK (注5)
0008 8051h	AD0	A/D 控制寄存器	ADCR	8	8	2~3个PCLK (注5)
0008 805Bh	AD0	A/D 采样状态寄存器	ADSSTR	8	8	2~3个PCLK (注5)
0008 805Dh	AD0	A/D 自诊断寄存器	ADDIAGR	8	8	2~3个PCLK (注5)
0008 8060h	AD0	A/D 数据寄存器 I	ADDRI	16	16	2~3个PCLK (注5)
0008 8062h	AD0	A/D 数据寄存器 J	ADDRJ	16	16	2~3个PCLK (注5)

表 5.1 I/O 寄存器地址一览表 (9/28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 8064h	AD0	A/D 数据寄存器 K	ADDRK	16	16	2~3 个 PCLK (注5)
0008 8066h	AD0	A/D 数据寄存器 L	ADDRL	16	16	2~3 个 PCLK (注5)
0008 8070h	AD0	A/D 开始触发选择寄存器	ADSTRGR	8	8	2~3 个 PCLK (注5)
0008 8072h	AD0	ADDRn 格式选择寄存器	ADDRPR	8	8	2~3 个 PCLK (注5)
0008 8240h	SCI0	串行模式寄存器	SMR (注1)	8	8	2~3 个 PCLK (注5)
0008 8241h	SCI0	位速率寄存器	BRR	8	8	2~3 个 PCLK (注5)
0008 8242h	SCI0	串行控制寄存器	SCR (注1)	8	8	2~3 个 PCLK (注5)
0008 8243h	SCI0	发送数据寄存器	TDR	8	8	2~3 个 PCLK (注5)
0008 8244h	SCI0	串行状态寄存器	SSR (注1)	8	8	2~3 个 PCLK (注5)
0008 8245h	SCI0	接收数据寄存器	RDR	8	8	2~3 个 PCLK (注5)
0008 8246h	SCI0	智能卡模式寄存器	SCMR	8	8	2~3 个 PCLK (注5)
0008 8247h	SCI0	串行扩展模式寄存器	SEMR	8	8	2~3 个 PCLK (注5)
0008 8240h	SMCI0	串行模式寄存器	SMR	8	8	2~3 个 PCLK (注5)
0008 8241h	SMCI0	位速率寄存器	BRR	8	8	2~3 个 PCLK (注5)
0008 8242h	SMCI0	串行控制寄存器	SCR	8	8	2~3 个 PCLK (注5)
0008 8243h	SMCI0	发送数据寄存器	TDR	8	8	2~3 个 PCLK (注5)
0008 8244h	SMCI0	串行状态寄存器	SSR	8	8	2~3 个 PCLK (注5)
0008 8245h	SMCI0	接收数据寄存器	RDR	8	8	2~3 个 PCLK (注5)
0008 8246h	SMCI0	智能卡模式寄存器	SCMR	8	8	2~3 个 PCLK (注5)
0008 8248h	SCI1	串行模式寄存器	SMR	8	8	2~3 个 PCLK (注5)
0008 8249h	SCI1	位速率寄存器	BRR	8	8	2~3 个 PCLK (注5)
0008 824Ah	SCI1	串行控制寄存器	SCR	8	8	2~3 个 PCLK (注5)
0008 824Bh	SCI1	发送数据寄存器	TDR	8	8	2~3 个 PCLK (注5)
0008 824Ch	SCI1	串行状态寄存器	SSR	8	8	2~3 个 PCLK (注5)
0008 824Dh	SCI1	接收数据寄存器	RDR	8	8	2~3 个 PCLK (注5)
0008 824Eh	SCI1	智能卡模式寄存器	SCMR	8	8	2~3 个 PCLK (注5)
0008 824Fh	SCI1	串行扩展模式寄存器	SEMR	8	8	2~3 个 PCLK (注5)
0008 8248h	SMCI1	串行模式寄存器	SMR	8	8	2~3 个 PCLK (注5)
0008 8249h	SMCI1	位速率寄存器	BRR	8	8	2~3 个 PCLK (注5)
0008 824Ah	SMCI1	串行控制寄存器	SCR	8	8	2~3 个 PCLK (注5)

表 5.1 I/O 寄存器地址一览表 (10 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 824Bh	SMC11	发送数据寄存器	TDR	8	8	2~3个PCLK (注5)
0008 824Ch	SMC11	串行状态寄存器	SSR	8	8	2~3个PCLK (注5)
0008 824Dh	SMC11	接收数据寄存器	RDR	8	8	2~3个PCLK (注5)
0008 824Eh	SMC11	智能卡模式寄存器	SCMR	8	8	2~3个PCLK (注5)
0008 8250h	SCI2	串行模式寄存器	SMR (注1)	8	8	2~3个PCLK (注5)
0008 8251h	SCI2	位速率寄存器	BRR	8	8	2~3个PCLK (注5)
0008 8252h	SCI2	串行控制寄存器	SCR (注1)	8	8	2~3个PCLK (注5)
0008 8253h	SCI2	发送数据寄存器	TDR	8	8	2~3个PCLK (注5)
0008 8254h	SCI2	串行状态寄存器	SSR (注1)	8	8	2~3个PCLK (注5)
0008 8255h	SCI2	接收数据寄存器	RDR	8	8	2~3个PCLK (注5)
0008 8256h	SCI2	智能卡模式寄存器	SCMR	8	8	2~3个PCLK (注5)
0008 8257h	SCI2	串行扩展模式寄存器	SEMR	8	8	2~3个PCLK (注5)
0008 8250h	SMC12	串行模式寄存器	SMR	8	8	2~3个PCLK (注5)
0008 8251h	SMC12	位速率寄存器	BRR	8	8	2~3个PCLK (注5)
0008 8252h	SMC12	串行控制寄存器	SCR	8	8	2~3个PCLK (注5)
0008 8253h	SMC12	发送数据寄存器	TDR	8	8	2~3个PCLK (注5)
0008 8254h	SMC12	串行状态寄存器	SSR	8	8	2~3个PCLK (注5)
0008 8255h	SMC12	接收数据寄存器	RDR	8	8	2~3个PCLK (注5)
0008 8256h	SMC12	智能卡模式寄存器	SCMR	8	8	2~3个PCLK (注5)
0008 8280h	CRC	CRC 控制寄存器	CRCCR	8	8	2~3个PCLK (注5)
0008 8281h	CRC	CRC 数据输入寄存器	CRCDIR	8	8	2~3个PCLK (注5)
0008 8282h	CRC	CRC 数据输出寄存器	CRCDOR	16	16	2~3个PCLK (注5)
0008 8300h	RIIC0	I <sup>2</sup> C 总线控制寄存器 1	ICCR1	8	8	2~3个PCLK (注5)
0008 8301h	RIIC0	I <sup>2</sup> C 总线控制寄存器 2	ICCR2	8	8	2~3个PCLK (注5)
0008 8302h	RIIC0	I <sup>2</sup> C 总线模式寄存器 1	ICMR1	8	8	2~3个PCLK (注5)
0008 8303h	RIIC0	I <sup>2</sup> C 总线模式寄存器 2	ICMR2	8	8	2~3个PCLK (注5)
0008 8304h	RIIC0	I <sup>2</sup> C 总线模式寄存器 3	ICMR3	8	8	2~3个PCLK (注5)
0008 8305h	RIIC0	I <sup>2</sup> C 总线功能允许寄存器	ICFER	8	8	2~3个PCLK (注5)
0008 8306h	RIIC0	I <sup>2</sup> C 总线状态允许寄存器	ICSER	8	8	2~3个PCLK (注5)
0008 8307h	RIIC0	I <sup>2</sup> C 总线中断允许寄存器	ICIER	8	8	2~3个PCLK (注5)

表 5.1 I/O 寄存器地址一览表 (11 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 8308h	RIIC0	I <sup>2</sup> C 总线状态寄存器 1	ICSR1	8	8	2~3 个 PCLK (注5)
0008 8309h	RIIC0	I <sup>2</sup> C 总线状态寄存器 2	ICSR2	8	8	2~3 个 PCLK (注5)
0008 830Ah	RIIC0	从属地址寄存器 L0	SARL0	8	8	2~3 个 PCLK (注5)
0008 830Ah	RIIC0	超时内部计数器 L	TMOCNL	16	16	2~3 个 PCLK (注5)
0008 830Bh	RIIC0	从属地址寄存器 U0	SARU0	8	8	2~3 个 PCLK (注5)
0008 830Bh	RIIC0	超时内部计数器 U	TMOCNTU	16	16	2~3 个 PCLK (注5)
0008 830Ch	RIIC0	从属地址寄存器 L1	SARL1	8	8	2~3 个 PCLK (注5)
0008 830Dh	RIIC0	从属地址寄存器 U1	SARU1	8	8	2~3 个 PCLK (注5)
0008 830Eh	RIIC0	从属地址寄存器 L2	SARL2	8	8	2~3 个 PCLK (注5)
0008 830Fh	RIIC0	从属地址寄存器 U2	SARU2	8	8	2~3 个 PCLK (注5)
0008 8310h	RIIC0	I <sup>2</sup> C 总线位速率低电平寄存器	ICBRL	8	8	2~3 个 PCLK (注5)
0008 8311h	RIIC0	I <sup>2</sup> C 总线位速率高电平寄存器	ICBRH	8	8	2~3 个 PCLK (注5)
0008 8312h	RIIC0	I <sup>2</sup> C 总线发送数据寄存器	ICDRT	8	8	2~3 个 PCLK (注5)
0008 8313h	RIIC0	I <sup>2</sup> C 总线接收数据寄存器	ICDRR	8	8	2~3 个 PCLK (注5)
0008 8380h	RSPI0	RSPI 控制寄存器	SPCR	8	8	2~3 个 PCLK (注5)
0008 8381h	RSPI0	RSPI 从属选择极性寄存器	SSLP	8	8	2~3 个 PCLK (注10)
0008 8382h	RSPI0	RSPI 引脚控制寄存器	SPPCR	8	8	2~3 个 PCLK (注5)
0008 8383h	RSPI0	RSPI 状态寄存器	SPSR	8	8	2~3 个 PCLK (注5)
0008 8384h	RSPI0	RSPI 数据寄存器	SPDR	32	16、32	2~3 个 PCLK (注5)
0008 8388h	RSPI0	RSPI 顺序控制寄存器	SPSCR	8	8	2~3 个 PCLK (注5)
0008 8389h	RSPI0	RSPI 顺序状态寄存器	SPSSR	8	8	2~3 个 PCLK (注5)
0008 838Ah	RSPI0	RSPI 位速率寄存器	SPBR	8	8	2~3 个 PCLK (注5)
0008 838Bh	RSPI0	RSPI 数据控制寄存器	SPDCR	8	8	2~3 个 PCLK (注5)
0008 838Ch	RSPI0	RSPI 时钟延迟寄存器	SPCKD	8	8	2~3 个 PCLK (注5)
0008 838Dh	RSPI0	RSPI 从属选择无效延迟寄存器	SSLND	8	8	2~3 个 PCLK (注5)
0008 838Eh	RSPI0	RSPI 下次存取延迟寄存器	SPND	8	8	2~3 个 PCLK (注5)
0008 838Fh	RSPI0	RSPI 控制寄存器 2	SPCR2	8	8	2~3 个 PCLK (注5)
0008 8390h	RSPI0	RSPI 命令寄存器 0	SPCMD0	16	16	2~3 个 PCLK (注5)
0008 8392h	RSPI0	RSPI 命令寄存器 1	SPCMD1	16	16	2~3 个 PCLK (注5)
0008 8394h	RSPI0	RSPI 命令寄存器 2	SPCMD2	16	16	2~3 个 PCLK (注5)



表 5.1 I/O 寄存器地址一览表 (12 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 8396h	RSPI0	RSPI 命令寄存器 3	SPCMD3	16	16	2~3 个 PCLK (注5)
0008 8398h	RSPI0	RSPI 命令寄存器 4	SPCMD4	16	16	2~3 个 PCLK (注5)
0008 839Ah	RSPI0	RSPI 命令寄存器 5	SPCMD5	16	16	2~3 个 PCLK (注5)
0008 839Ch	RSPI0	RSPI 命令寄存器 6	SPCMD6	16	16	2~3 个 PCLK (注5)
0008 839Eh	RSPI0	RSPI 命令寄存器 7	SPCMD7	16	16	2~3 个 PCLK (注5)
0008 9000h	S12AD	A/D 控制寄存器	ADCSR	8	8	2~3 个 PCLK (注5)
0008 9004h	S12AD	A/D 通道选择寄存器	ADANS	16	16	2~3 个 PCLK (注5)
0008 900Ah	S12AD	A/D 可编程增益放大器寄存器	ADPG	16	16	2~3 个 PCLK (注5)
0008 900Eh	S12AD	A/D 控制扩展寄存器	ADCER	16	16	2~3 个 PCLK (注5)
0008 9010h	S12AD	A/D 开始触发选择寄存器	ADSTRGR	16	16	2~3 个 PCLK (注5)
0008 9012h	S12AD	比较器运行模式选择寄存器 0	ADCMPMD0	16	16	2~3 个 PCLK (注5)
0008 9014h	S12AD	比较器运行模式选择寄存器 1	ADCMPMD1	16	16	2~3 个 PCLK (注5)
0008 9016h	S12AD	比较器的滤波器模式寄存器 0	ADCMPNR0	16	16	2~3 个 PCLK (注5)
0008 9018h	S12AD	比较器的滤波器模式寄存器 1	ADCMPNR1	16	16	2~3 个 PCLK (注5)
0008 901Ah	S12AD	比较器检测标准寄存器	ADCMPFR	8	8	2~3 个 PCLK (注5)
0008 901Ch	S12AD	比较器中断选择寄存器	ADCMPSEL	16	16	2~3 个 PCLK (注5)
0008 901Eh	S12AD0	A/D 数据寄存器 Diag	ADRD	16	16	2~3 个 PCLK (注5)
0008 9020h	S12AD0	A/D 数据寄存器 0A	ADDR0A	16	16	2~3 个 PCLK (注5)
0008 9022h	S12AD0	A/D 数据寄存器 1	ADDR1	16	16	2~3 个 PCLK (注5)
0008 9024h	S12AD0	A/D 数据寄存器 2	ADDR2	16	16	2~3 个 PCLK (注5)
0008 9026h	S12AD0	A/D 数据寄存器 3	ADDR3	16	16	2~3 个 PCLK (注5)
0008 9030h	S12AD0	A/D 数据寄存器 0B	ADDR0B	16	16	2~3 个 PCLK (注5)
0008 9060h	S12AD0	A/D 采样状态寄存器	ADSSTR	8	8	2~3 个 PCLK (注5)
0008 9080h	S12AD1	A/D 控制寄存器	ADCSR	8	8	2~3 个 PCLK (注5)
0008 9084h	S12AD1	A/D 通道选择寄存器	ADANS	16	16	2~3 个 PCLK (注5)
0008 908Ah	S12AD1	A/D 可编程增益放大器寄存器	ADPG	16	16	2~3 个 PCLK (注5)
0008 908Eh	S12AD1	A/D 控制扩展寄存器	ADCER	16	16	2~3 个 PCLK (注5)
0008 9090h	S12AD1	A/D 开始触发选择寄存器	ADSTRGR	16	16	2~3 个 PCLK (注5)
0008 909Eh	S12AD1	A/D 数据寄存器 Diag	ADRD	16	16	2~3 个 PCLK (注5)
0008 90A0h	S12AD1	A/D 数据寄存器 0A	ADDR0A	16	16	2~3 个 PCLK (注5)

表 5.1 I/O 寄存器地址一览表 (13 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 90A2h	S12AD1	A/D 数据寄存器 1	ADDR1	16	16	2~3 个 PCLK (注5)
0008 90A4h	S12AD1	A/D 数据寄存器 2	ADDR2	16	16	2~3 个 PCLK (注5)
0008 90A6h	S12AD1	A/D 数据寄存器 3	ADDR3	16	16	2~3 个 PCLK (注5)
0008 90B0h	S12AD1	A/D 数据寄存器 0B	ADDR0B	16	16	2~3 个 PCLK (注5)
0008 90E0h	S12AD1	A/D 采样状态寄存器	ADSSTR	8	8	2~3 个 PCLK (注5)
0008 C001h	PORT1	数据方向寄存器	DDR	8	8	2~3 个 PCLK (注5)
0008 C002h	PORT2	数据方向寄存器	DDR	8	8	2~3 个 PCLK (注5)
0008 C003h	PORT3	数据方向寄存器	DDR	8	8	2~3 个 PCLK (注5)
0008 C007h	PORT7	数据方向寄存器	DDR	8	8	2~3 个 PCLK (注5)
0008 C008h	PORT8	数据方向寄存器	DDR (注2、注3)	8	8	2~3 个 PCLK (注5)
0008 C009h	PORT9	数据方向寄存器	DDR	8	8	2~3 个 PCLK (注5)
0008 C00Ah	PORTA	数据方向寄存器	DDR	8	8	2~3 个 PCLK (注5)
0008 C00Bh	PORTB	数据方向寄存器	DDR	8	8	2~3 个 PCLK (注5)
0008 C00Dh	PORTD	数据方向寄存器	DDR	8	8	2~3 个 PCLK (注5)
0008 C00Eh	PORTE	数据方向寄存器	DDR	8	8	2~3 个 PCLK (注5)
0008 C010h	PORTG	数据方向寄存器	DDR (注1、注2、注3)	8	8	2~3 个 PCLK (注5)
0008 C021h	PORT1	数据寄存器	DR	8	8	2~3 个 PCLK (注5)
0008 C022h	PORT2	数据寄存器	DR	8	8	2~3 个 PCLK (注5)
0008 C023h	PORT3	数据寄存器	DR	8	8	2~3 个 PCLK (注5)
0008 C027h	PORT7	数据寄存器	DR	8	8	2~3 个 PCLK (注5)
0008 C028h	PORT8	数据寄存器	DR (注2、注3)	8	8	2~3 个 PCLK (注5)
0008 C029h	PORT9	数据寄存器	DR	8	8	2~3 个 PCLK (注5)
0008 C02Ah	PORTA	数据寄存器	DR	8	8	2~3 个 PCLK (注5)
0008 C02Bh	PORTB	数据寄存器	DR	8	8	2~3 个 PCLK (注5)
0008 C02Dh	PORTD	数据寄存器	DR	8	8	2~3 个 PCLK (注5)
0008 C02Eh	PORTE	数据寄存器	DR	8	8	2~3 个 PCLK (注5)
0008 C030h	PORTG	数据寄存器	DR (注1、注2、注3)	8	8	2~3 个 PCLK (注5)
0008 C041h	PORT1	端口寄存器	PORT	8	8	2~3 个 PCLK (注5)
0008 C042h	PORT2	端口寄存器	PORT	8	8	2~3 个 PCLK (注5)
0008 C043h	PORT3	端口寄存器	PORT	8	8	2~3 个 PCLK (注5)

表 5.1 I/O 寄存器地址一览表 (14 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 C044h	PORT4	端口寄存器	PORT	8	8	2~3个PCLK (注5)
0008 C045h	PORT5	端口寄存器	PORT (注2、注3)	8	8	2~3个PCLK (注5)
0008 C046h	PORT6	端口寄存器	PORT (注3)	8	8	2~3个PCLK (注5)
0008 C047h	PORT7	端口寄存器	PORT	8	8	2~3个PCLK (注5)
0008 C048h	PORT8	端口寄存器	PORT (注2、注3)	8	8	2~3个PCLK (注5)
0008 C049h	PORT9	端口寄存器	PORT	8	8	2~3个PCLK (注5)
0008 C04Ah	PORTA	端口寄存器	PORT	8	8	2~3个PCLK (注5)
0008 C04Bh	PORTB	端口寄存器	PORT	8	8	2~3个PCLK (注5)
0008 C04Dh	PORTD	端口寄存器	PORT	8	8	2~3个PCLK (注5)
0008 C04Eh	PORTE	端口寄存器	PORT	8	8	2~3个PCLK (注5)
0008 C050h	PORTG	端口寄存器	PORT (注1、注2、注3)	8	8	2~3个PCLK (注5)
0008 C061h	PORT1	输入缓冲控制寄存器	ICR	8	8	2~3个PCLK (注5)
0008 C062h	PORT2	输入缓冲控制寄存器	ICR	8	8	2~3个PCLK (注5)
0008 C063h	PORT3	输入缓冲控制寄存器	ICR	8	8	2~3个PCLK (注5)
0008 C064h	PORT4	输入缓冲控制寄存器	ICR	8	8	2~3个PCLK (注5)
0008 C065h	PORT5	输入缓冲控制寄存器	ICR (注2、注3)	8	8	2~3个PCLK (注5)
0008 C066h	PORT6	输入缓冲控制寄存器	ICR (注3)	8	8	2~3个PCLK (注5)
0008 C067h	PORT7	输入缓冲控制寄存器	ICR	8	8	2~3个PCLK (注5)
0008 C068h	PORT8	输入缓冲控制寄存器	ICR (注2、注3)	8	8	2~3个PCLK (注5)
0008 C069h	PORT9	输入缓冲控制寄存器	ICR	8	8	2~3个PCLK (注5)
0008 C06Ah	PORTA	输入缓冲控制寄存器	ICR	8	8	2~3个PCLK (注5)
0008 C06Bh	PORTB	输入缓冲控制寄存器	ICR	8	8	2~3个PCLK (注5)
0008 C06Dh	PORTD	输入缓冲控制寄存器	ICR	8	8	2~3个PCLK (注5)
0008 C06Eh	PORTE	输入缓冲控制寄存器	ICR	8	8	2~3个PCLK (注5)
0008 C070h	PORTG	输入缓冲控制寄存器	ICR (注1、注2、注3)	8	8	2~3个PCLK (注5)
0008 C108h	IOPORT	端口功能寄存器 8	PF8IRQ	8	8	2~3个PCLK (注5)
0008 C109h	IOPORT	端口功能寄存器 9	PF9IRQ	8	8	2~3个PCLK (注5)
0008 C10Ah	IOPORT	端口功能寄存器 A	PFAADC	8	8	2~3个PCLK (注5)
0008 C10Ch	IOPORT	端口功能寄存器 C	PFCMTU	8	8	2~3个PCLK (注5)
0008 C10Dh	IOPORT	端口功能寄存器 D	PFDGPT	8	8	2~3个PCLK (注5)

表 5.1 I/O 寄存器地址一览表 (15 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 C10Fh	IOPORT	端口功能寄存器 F	PFFSCI	8	8	2 ~ 3 个 PCLK (注5)
0008 C110h	IOPORT	端口功能寄存器 G	PFGSPI	8	8	2 ~ 3 个 PCLK (注5)
0008 C111h	IOPORT	端口功能寄存器 H	PFHSPI	8	8	2 ~ 3 个 PCLK (注5)
0008 C113h	IOPORT	端口功能寄存器 J	PFJCAN	8	8	2 ~ 3 个 PCLK (注5)
0008 C114h	IOPORT	端口功能寄存器 K	PFKLIN	8	8	2 ~ 3 个 PCLK (注5)
0008 C116h	IOPORT	端口功能寄存器 M	PFMPOE (注9)	8	8	2 ~ 3 个 PCLK (注5)
0008 C117h	IOPORT	端口功能寄存器 N	PFNPOE (注9)	8	8	2 ~ 3 个 PCLK (注5)
0008 C280h	SYSTEM	深度待机控制寄存器	DPSBYCR	8	8	4 ~ 5 个 PCLK (注5)
0008 C281h	SYSTEM	深度待机等待控制寄存器	DPSWCR	8	8	4 ~ 5 个 PCLK (注5)
0008 C282h	SYSTEM	深度待机中断允许寄存器	DPSIER	8	8	4 ~ 5 个 PCLK (注5)
0008 C283h	SYSTEM	深度待机中断标志寄存器	DPSIFR	8	8	4 ~ 5 个 PCLK (注5)
0008 C284h	SYSTEM	深度待机中断边沿寄存器	DPSIEGR	8	8	4 ~ 5 个 PCLK (注5)
0008 C285h	SYSTEM	复位状态寄存器	RSTSR	8	8	4 ~ 5 个 PCLK (注5)
0008 C289h	FLASH	闪存编程 / 擦除保护寄存器	FWEPROR	8	8	4 ~ 5 个 PCLK (注5)
0008 C28Ch	SYSTEM	控制低电压检测的键码寄存器	LVDKEYR	8	8	4 ~ 5 个 PCLK (注5)
0008 C28Dh	SYSTEM	低电压检测控制寄存器	LVDCCR	8	8	4 ~ 5 个 PCLK (注5)
0008 C290h	SYSTEM	深度待机备用寄存器 0	DPSBKR0	8	8	4 ~ 5 个 PCLK (注5)
0008 C291h	SYSTEM	深度待机备用寄存器 1	DPSBKR1	8	8	4 ~ 5 个 PCLK (注5)
0008 C292h	SYSTEM	深度待机备用寄存器 2	DPSBKR2	8	8	4 ~ 5 个 PCLK (注5)
0008 C293h	SYSTEM	深度待机备用寄存器 3	DPSBKR3	8	8	4 ~ 5 个 PCLK (注5)
0008 C294h	SYSTEM	深度待机备用寄存器 4	DPSBKR4	8	8	4 ~ 5 个 PCLK (注5)
0008 C295h	SYSTEM	深度待机备用寄存器 5	DPSBKR5	8	8	4 ~ 5 个 PCLK (注5)
0008 C296h	SYSTEM	深度待机备用寄存器 6	DPSBKR6	8	8	4 ~ 5 个 PCLK (注5)
0008 C297h	SYSTEM	深度待机备用寄存器 7	DPSBKR7	8	8	4 ~ 5 个 PCLK (注5)
0008 C298h	SYSTEM	深度待机备用寄存器 8	DPSBKR8	8	8	4 ~ 5 个 PCLK (注5)
0008 C299h	SYSTEM	深度待机备用寄存器 9	DPSBKR9	8	8	4 ~ 5 个 PCLK (注5)
0008 C29Ah	SYSTEM	深度待机备用寄存器 10	DPSBKR10	8	8	4 ~ 5 个 PCLK (注5)
0008 C29Bh	SYSTEM	深度待机备用寄存器 11	DPSBKR11	8	8	4 ~ 5 个 PCLK (注5)
0008 C29Ch	SYSTEM	深度待机备用寄存器 12	DPSBKR12	8	8	4 ~ 5 个 PCLK (注5)
0008 C29Dh	SYSTEM	深度待机备用寄存器 13	DPSBKR13	8	8	4 ~ 5 个 PCLK (注5)

表 5.1 I/O 寄存器地址一览表 (16 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 C29Eh	SYSTEM	深度待机备用寄存器 14	DPSBKR14	8	8	4 ~ 5 个 PCLK (注 5)
0008 C29Fh	SYSTEM	深度待机备用寄存器 15	DPSBKR15	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2A0h	SYSTEM	深度待机备用寄存器 16	DPSBKR16	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2A1h	SYSTEM	深度待机备用寄存器 17	DPSBKR17	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2A2h	SYSTEM	深度待机备用寄存器 18	DPSBKR18	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2A3h	SYSTEM	深度待机备用寄存器 19	DPSBKR19	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2A4h	SYSTEM	深度待机备用寄存器 20	DPSBKR20	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2A5h	SYSTEM	深度待机备用寄存器 21	DPSBKR21	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2A6h	SYSTEM	深度待机备用寄存器 22	DPSBKR22	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2A7h	SYSTEM	深度待机备用寄存器 23	DPSBKR23	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2A8h	SYSTEM	深度待机备用寄存器 24	DPSBKR24	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2A9h	SYSTEM	深度待机备用寄存器 25	DPSBKR25	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2AAh	SYSTEM	深度待机备用寄存器 26	DPSBKR26	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2ABh	SYSTEM	深度待机备用寄存器 27	DPSBKR27	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2ACh	SYSTEM	深度待机备用寄存器 28	DPSBKR28	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2ADh	SYSTEM	深度待机备用寄存器 29	DPSBKR29	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2AEh	SYSTEM	深度待机备用寄存器 30	DPSBKR30	8	8	4 ~ 5 个 PCLK (注 5)
0008 C2AFh	SYSTEM	深度待机备用寄存器 31	DPSBKR31	8	8	4 ~ 5 个 PCLK (注 5)
0008 C4C0h	POE	输入电平控制 / 状态寄存器 1	ICSR1	16	8、16	2 ~ 3PCLK (注 5)
0008 C4C2h	POE	输出电平控制 / 状态寄存器 1	OCSR1	16	8、16	2 ~ 3PCLK (注 5)
0008 C4C4h	POE	输入电平控制 / 状态寄存器 2	ICSR2	16	8、16	2 ~ 3PCLK (注 5)
0008 C4C6h	POE	输出电平控制 / 状态寄存器 2	OCSR2	16	8、16	2 ~ 3PCLK (注 5)
0008 C4C8h	POE	输入电平控制 / 状态寄存器 3	ICSR3	16	8、16	2 ~ 3PCLK (注 5)
0008 C4CAh	POE	软件端口输出允许寄存器	SPOER	8	8	2 ~ 3PCLK (注 5)
0008 C4CBh	POE	端口输出允许控制寄存器 1	POECR1	8	8	2 ~ 3PCLK (注 5)
0008 C4CCh	POE	端口输出允许控制寄存器 2	POECR2	16	16	2 ~ 3PCLK (注 5)
0008 C4CEh	POE	端口输出允许控制寄存器 3	POECR3	16	16	2 ~ 3PCLK (注 5)
0008 C4D0h	POE	端口输出允许控制寄存器 4	POECR4	16	16	2 ~ 3PCLK (注 5)
0008 C4D2h	POE	端口输出允许控制寄存器 5	POECR5	16	16	2 ~ 3PCLK (注 5)
0008 C4D4h	POE	端口输出允许控制寄存器 6	POECR6	16	16	2 ~ 3PCLK (注 5)

表 5.1 I/O 寄存器地址一览表 (17 / 28)

地址	模块符号	寄存器名	寄存器符号	位数	存取长度	存取周期数
0008 C4D6h	POE	输入电平控制 / 状态寄存器 4	ICSR4	16	8、16	2 ~ 3PCLK (注5)
0008 C4D8h	POE	输入电平控制 / 状态寄存器 5	ICSR5	16	8、16	2 ~ 3PCLK (注5)
0008 C4DAh	POE	有效电平寄存器 1	ALR1	16	8、16	2 ~ 3PCLK (注5)
0009 0200h ~ 0009 03FFh	CAN0 (注4)	信箱寄存器 0 ~ 31	MB0 ~ 31	128	8、16、32	2 ~ 3个PCLK (注5)
0009 0400h	CAN0 (注4)	屏蔽寄存器 0	MKR0	32	8、16、32	2 ~ 3个PCLK (注5)
0009 0404h	CAN0 (注4)	屏蔽寄存器 1	MKR1	32	8、16、32	2 ~ 3个PCLK (注5)
0009 0408h	CAN0 (注4)	屏蔽寄存器 2	MKR2	32	8、16、32	2 ~ 3个PCLK (注5)
0009 040Ch	CAN0 (注4)	屏蔽寄存器 3	MKR3	32	8、16、32	2 ~ 3个PCLK (注5)
0009 0410h	CAN0 (注4)	屏蔽寄存器 4	MKR4	32	8、16、32	2 ~ 3个PCLK (注5)
0009 0414h	CAN0 (注4)	屏蔽寄存器 5	MKR5	32	8、16、32	2 ~ 3个PCLK (注5)
0009 0418h	CAN0 (注4)	屏蔽寄存器 6	MKR6	32	8、16、32	2 ~ 3个PCLK (注5)
0009 041Ch	CAN0 (注4)	屏蔽寄存器 7	MKR7	32	8、16、32	2 ~ 3个PCLK (注5)
0009 0420h	CAN0 (注4)	FIFO 接收 ID 比较寄存器 0	FIDCR0	32	8、16、32	2 ~ 3个PCLK (注5)
0009 0424h	CAN0 (注4)	FIFO 接收 ID 比较寄存器 1	FIDCR1	32	8、16、32	2 ~ 3个PCLK (注5)
0009 0428h	CAN0 (注4)	屏蔽无效寄存器	MKIVLR	32	8、16、32	2 ~ 3个PCLK (注5)
0009 042Ch	CAN0 (注4)	信箱中断允许寄存器	MIER	32	8、16、32	2 ~ 3个PCLK (注5)
0009 0820h ~ 0009 083Fh	CAN0 (注4)	信息控制寄存器 0 ~ 31	MCTL0 ~ 31	8	8	2 ~ 3个PCLK (注5)
0009 0840h	CAN0 (注4)	控制寄存器	CTLR	16	8、16	2 ~ 3个PCLK (注5)
0009 0842h	CAN0 (注4)	状态寄存器	STR	16	8、16	2 ~ 3个PCLK (注5)
0009 0844h	CAN0 (注4)	位配置寄存器	BCR	32	8、16、32	2 ~ 3个PCLK (注5)
0009 0848h	CAN0 (注4)	接收 FIFO 控制寄存器	RFCR	8	8	2 ~ 3个PCLK (注5)
0009 0849h	CAN0 (注4)	接收 FIFO 指针控制寄存器	RFPCR	8	8	2 ~ 3个PCLK (注5)
0009 084Ah	CAN0 (注4)	发送 FIFO 控制寄存器	TFCR	8	8	2 ~ 3个PCLK (注5)
0009 084Bh	CAN0 (注4)	发送 FIFO 指针控制寄存器	TFPCR	8	8	2 ~ 3个PCLK (注5)
0009 084Ch	CAN0 (注4)	错误中断允许寄存器	EIER	8	8	2 ~ 3个PCLK (注5)
0009 084Dh	CAN0 (注4)	错误中断源判断寄存器	EIFR	8	8	2 ~ 3个PCLK (注5)
0009 084Eh	CAN0 (注4)	接收错误计数寄存器	RECR	8	8	2 ~ 3个PCLK (注5)
0009 084Fh	CAN0 (注4)	发送错误计数寄存器	TECR	8	8	2 ~ 3个PCLK (注5)
0009 0850h	CAN0 (注4)	错误代码保存寄存器	ECSR	8	8	2 ~ 3个PCLK (注5)
0009 0851h	CAN0 (注4)	通道搜索支持寄存器	CSSR	8	8	2 ~ 3个PCLK (注5)

表 5.1 I/O 寄存器地址一览表 (18 / 28)

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
0009 0852h	CAN0 (注4)	信箱搜索状态寄存器	MSSR	8	8	2 ~ 3个PCLK (注5)
0009 0853h	CAN0 (注4)	信箱搜索模式寄存器	MSMR	8	8	2 ~ 3个PCLK (注5)
0009 0854h	CAN0 (注4)	时戳寄存器	TSR	16	8、16	2 ~ 3个PCLK (注5)
0009 0856h	CAN0 (注4)	验收滤波器支持寄存器	AFSR	16	8、16	2 ~ 3个PCLK (注5)
0009 0858h	CAN0 (注4)	测试控制寄存器	TCR	8	8	2 ~ 3个PCLK (注5)
0009 4001h	LINO	LIN 唤醒波特率选择寄存器	LWBR	8	8	2 ~ 3PCLK (注5)
0009 4002h	LINO	LIN 波特率预分频器 0 寄存器	LBRP0	8	8、16	2 ~ 3PCLK (注5)
0009 4003h	LINO	LIN 波特率预分频器 1 寄存器	LBRP1	8	8、16	2 ~ 3PCLK (注5)
0009 4004h	LINO	LIN 自测试控制寄存器	LSTC	8	8	2 ~ 3PCLK (注5)
0009 4008h	LINO	模式寄存器	L0MD	8	8、16、32	2 ~ 3PCLK (注5)
0009 4009h	LINO	间隔场设定寄存器	L0BRK	8	8、16、32	2 ~ 3PCLK (注5)
0009 400Ah	LINO	间隔设定寄存器	L0SPC	8	8、16、32	2 ~ 3PCLK (注5)
0009 400Bh	LINO	唤醒设定寄存器	L0WUP	8	8、16、32	2 ~ 3PCLK (注5)
0009 400Ch	LINO	中断允许寄存器	L0IE	8	8、16	2 ~ 3PCLK (注5)
0009 400Dh	LINO	错误检测允许寄存器	L0EDE	8	8、16	2 ~ 3PCLK (注5)
0009 400Eh	LINO	控制寄存器	L0C	8	8	2 ~ 3PCLK (注5)
0009 4010h	LINO	发送控制寄存器	L0TC	8	8、16、32	2 ~ 3PCLK (注5)
0009 4011h	LINO	模式状态寄存器	L0MST	8	8、16、32	2 ~ 3PCLK (注5)
0009 4012h	LINO	状态寄存器	L0ST	8	8、16、32	2 ~ 3PCLK (注5)
0009 4013h	LINO	错误状态寄存器	L0EST	8	8、16、32	2 ~ 3PCLK (注5)
0009 4014h	LINO	响应场设定寄存器	L0RFC	8	8、16	2 ~ 3PCLK (注5)
0009 4015h	LINO	ID 缓冲寄存器	L0IDB	8	8、16	2 ~ 3PCLK (注5)
0009 4016h	LINO	校验和缓冲寄存器	L0CBR	8	8	2 ~ 3PCLK (注5)
0009 4018h	LINO	数据 1 缓冲寄存器	L0DB1	8	8、16、32	2 ~ 3PCLK (注5)
0009 4019h	LINO	数据 2 缓冲寄存器	L0DB2	8	8、16、32	2 ~ 3PCLK (注5)
0009 401Ah	LINO	数据 3 缓冲寄存器	L0DB3	8	8、16、32	2 ~ 3PCLK (注5)
0009 401Bh	LINO	数据 4 缓冲寄存器	L0DB4	8	8、16、32	2 ~ 3PCLK (注5)
0009 401Ch	LINO	数据 5 缓冲寄存器	L0DB5	8	8、16、32	2 ~ 3PCLK (注5)
0009 401Dh	LINO	数据 6 缓冲寄存器	L0DB6	8	8、16、32	2 ~ 3PCLK (注5)
0009 401Eh	LINO	数据 7 缓冲寄存器	L0DB7	8	8、16、32	2 ~ 3PCLK (注5)

表 5.1 I/O 寄存器地址一览表 (19 / 28)

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
0009 401Fh	LINO	数据 8 缓冲寄存器	L0DB8	8	8、16、32	2 ~ 3PCLK (注 5)
000C 1200h	MTU3	定时器的控制寄存器	TCR	8	8、16、32	5ICLK
000C 1201h	MTU4	定时器的控制寄存器	TCR	8	8	5ICLK
000C 1202h	MTU3	定时器的模式寄存器 1	TMDR1	8	8、16	5ICLK
000C 1203h	MTU4	定时器的模式寄存器 1	TMDR1	8	8	5ICLK
000C 1204h	MTU3	定时器的 I/O 控制寄存器 H	TIORH	8	8、16、32	5ICLK
000C 1205h	MTU3	定时器的 I/O 控制寄存器 L	TIORL	8	8	5ICLK
000C 1206h	MTU4	定时器的 I/O 控制寄存器 H	TIORH	8	8、16	5ICLK
000C 1207h	MTU4	定时器的 I/O 控制寄存器 L	TIORL	8	8	5ICLK
000C 1208h	MTU3	定时器的中断允许寄存器	TIER	8	8、16	5ICLK
000C 1209h	MTU4	定时器的中断允许寄存器	TIER	8	8	5ICLK
000C 120Ah	MTU	定时器的输出主控允许寄存器	TOERA	8	8	5ICLK
000C 120Dh	MTU	定时器的门控寄存器 A	TGCRA	8	8	5ICLK
000C 120Eh	MTU	定时器的输出控制寄存器 1A	TOCR1A	8	8、16	5ICLK
000C 120Fh	MTU	定时器的输出控制寄存器 2A	TOCR2A	8	8	5ICLK
000C 1210h	MTU3	定时器的计数器	TCNT	16	16、32	5ICLK
000C 1212h	MTU4	定时器的计数器	TCNT	16	16	5ICLK
000C 1214h	MTU	定时器的周期数据寄存器 A	TCRA	16	16、32	5ICLK
000C 1216h	MTU	定时器的死区时间数据寄存器 A	TDDRA	16	16	5ICLK
000C 1218h	MTU3	定时器的通用寄存器 A	TGRA	16	16、32	5ICLK
000C 121Ah	MTU3	定时器的通用寄存器 B	TGRB	16	16	5ICLK
000C 121Ch	MTU4	定时器的通用寄存器 A	TGRA	16	16、32	5ICLK
000C 121Eh	MTU4	定时器的通用寄存器 B	TGRB	16	16	5ICLK
000C 1220h	MTU	定时器的副计数器 A	TCNTSA	16	16、32	5ICLK
000C 1222h	MTU	定时器的周期缓冲寄存器 A	TCBRA	16	16	5ICLK
000C 1224h	MTU3	定时器的通用寄存器 C	TGRC	16	16、32	5ICLK
000C 1226h	MTU3	定时器的通用寄存器 D	TGRD	16	16	5ICLK
000C 1228h	MTU4	定时器的通用寄存器 C	TGRC	16	16、32	5ICLK
000C 122Ah	MTU4	定时器的通用寄存器 D	TGRD	16	16	5ICLK
000C 122Ch	MTU3	定时器的状态寄存器	TSR	8	8、16	5ICLK
000C 122Dh	MTU4	定时器的状态寄存器	TSR	8	8	5ICLK
000C 1230h	MTU	定时器的中断减少设定寄存器 1A	TITCR1A	8	8、16	5ICLK
000C 1231h	MTU	定时器的中断减少次数计数器 1A	TITCNT1A	8	8	5ICLK
000C 1232h	MTU	定时器的缓冲传送设定寄存器 A	TBTERA	8	8	5ICLK
000C 1234h	MTU	定时器的死区时间允许寄存器 A	TDERA	8	8	5ICLK
000C 1236h	MTU	定时器的输出电平缓冲寄存器 A	TOLBRA	8	8	5ICLK
000C 1238h	MTU3	定时器的缓冲运行传送模式寄存器	TBTM	8	8、16	5ICLK
000C 1239h	MTU4	定时器的缓冲运行传送模式寄存器	TBTM	8	8	5ICLK
000C 123Ah	MTU	定时器的中断减少模式寄存器 A	TITMRA	8	8	5ICLK
000C 123Bh	MTU	定时器的中断减少设定寄存器 2A	TITCR2A	8	8	5ICLK
000C 123Ch	MTU	定时器的中断减少次数计数器 2A	TITCNT2A	8	8	5ICLK
000C 1240h	MTU4	定时器的 A/D 转换开始请求控制寄存器	TADCR	16	16	5ICLK
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	5ICLK
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	5ICLK
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	5ICLK



表 5.1 I/O 寄存器地址一览表 (20 / 28)

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
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	5ICLK
000C 1260h	MTU	定时器的波形控制寄存器 A	TWCRA	8	8	5ICLK
000C 1270h	MTU	定时器的模式寄存器 2A	TMDR2A	8	8	5ICLK
000C 1272h	MTU3	定时器的通用寄存器 E	TGRE	16	16	5ICLK
000C 1274h	MTU4	定时器的通用寄存器 E	TGRE	16	16	5ICLK
000C 1276h	MTU4	定时器的通用寄存器 F	TGRF	16	16	5ICLK
000C 1280h	MTU	定时器的启动寄存器 A	TSTRA	8	8、16	5ICLK
000C 1281h	MTU	定时器的同步寄存器 A	TSYRA	8	8	5ICLK
000C 1282h	MTU	定时器的计数器同步启动寄存器	TCSYSTR	8	8	5ICLK
000C 1284h	MTU	定时器的读写允许寄存器 A	TRWERA	8	8	5ICLK
000C 1300h	MTU0	定时器的控制寄存器	TCR	8	8、16、32	5ICLK
000C 1301h	MTU0	定时器的模式寄存器 1	TMDR1	8	8	5ICLK
000C 1302h	MTU0	定时器的 I/O 控制寄存器 H	TIORH	8	8、16	5ICLK
000C 1303h	MTU0	定时器的 I/O 控制寄存器 L	TIORL	8	8	5ICLK
000C 1304h	MTU0	定时器的中断允许寄存器	TIER	8	8、16、32	5ICLK
000C 1305h	MTU0	定时器的状态寄存器	TSR	8	8	5ICLK
000C 1306h	MTU0	定时器的计数器	TCNT	16	16	5ICLK
000C 1308h	MTU0	定时器的通用寄存器 A	TGRA	16	16、32	5ICLK
000C 130Ah	MTU0	定时器的通用寄存器 B	TGRB	16	16	5ICLK
000C 130Ch	MTU0	定时器的通用寄存器 C	TGRC	16	16、32	5ICLK
000C 130Eh	MTU0	定时器的通用寄存器 D	TGRD	16	16	5ICLK
000C 1320h	MTU0	定时器的通用寄存器 E	TGRE	16	16、32	5ICLK
000C 1322h	MTU0	定时器的通用寄存器 F	TGRF	16	16	5ICLK
000C 1324h	MTU0	定时器的中断允许寄存器 2	TIER2	8	8、16	5ICLK
000C 1325h	MTU0	定时器的状态寄存器 2	TSR2	8	8	5ICLK
000C 1326h	MTU0	定时器的缓冲允许传送模式寄存器	TBTM	8	8	5ICLK
000C 1380h	MTU1	定时器的控制寄存器	TCR	8	8、16	5ICLK
000C 1381h	MTU1	定时器的模式寄存器 1	TMDR1	8	8	5ICLK
000C 1382h	MTU1	定时器的 I/O 控制寄存器	TIOR	8	8	5ICLK
000C 1384h	MTU1	定时器的中断允许寄存器	TIER	8	8、16、32	5ICLK
000C 1385h	MTU1	定时器的状态寄存器	TSR	8	8	5ICLK
000C 1386h	MTU1	定时器的计数器	TCNT	16	16	5ICLK
000C 1388h	MTU1	定时器的通用寄存器 A	TGRA	16	16、32	5ICLK
000C 138Ah	MTU1	定时器的通用寄存器 B	TGRB	16	16	5ICLK
000C 1390h	MTU1	定时器的输入捕捉控制寄存器	TICCR	8	8	5ICLK
000C 1400h	MTU2	定时器的控制寄存器	TCR	8	8、16	5ICLK
000C 1401h	MTU2	定时器的模式寄存器 1	TMDR1	8	8	5ICLK
000C 1402h	MTU2	定时器的 I/O 控制寄存器	TIOR	8	8	5ICLK
000C 1404h	MTU2	定时器的中断允许寄存器	TIER	8	8、16、32	5ICLK
000C 1405h	MTU2	定时器的状态寄存器	TSR	8	8	5ICLK
000C 1406h	MTU2	定时器的计数器	TCNT	16	16	5ICLK
000C 1408h	MTU2	定时器的通用寄存器 A	TGRA	16	16、32	5ICLK
000C 140Ah	MTU2	定时器的通用寄存器 B	TGRB	16	16	5ICLK
000C 1A00h	MTU6	定时器的控制寄存器	TCR	8	8、16、32	5ICLK
000C 1A01h	MTU7	定时器的控制寄存器	TCR	8	8	5ICLK
000C 1A02h	MTU6	定时器的模式寄存器 1	TMDR1	8	8、16	5ICLK
000C 1A03h	MTU7	定时器的模式寄存器 1	TMDR1	8	8	5ICLK
000C 1A04h	MTU6	定时器的 I/O 控制寄存器 H	TIORH	8	8、16、32	5ICLK

表 5.1 I/O 寄存器地址一览表 (21 / 28)

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
000C 1A05h	MTU6	定时器的 I/O 控制寄存器 L	TIORL	8	8	5ICLK
000C 1A06h	MTU7	定时器的 I/O 控制寄存器 H	TIORH	8	8、16	5ICLK
000C 1A07h	MTU7	定时器的 I/O 控制寄存器 L	TIORL	8	8	5ICLK
000C 1A08h	MTU6	定时器的中断允许寄存器	TIER	8	8、16	5ICLK
000C 1A09h	MTU7	定时器的中断允许寄存器	TIER	8	8	5ICLK
000C 1A0Ah	MTU	定时器的输出主控允许寄存器 B	TOERB	8	8	5ICLK
000C 1A0Eh	MTU	定时器的输出控制寄存器 1B	TOCR1B	8	8、16	5ICLK
000C 1A0Fh	MTU	定时器的输出控制寄存器 2B	TOCR2B	8	8	5ICLK
000C 1A10h	MTU6	定时器的计数器	TCNT	16	16、32	5ICLK
000C 1A12h	MTU7	定时器的计数器	TCNT	16	16	5ICLK
000C 1A14h	MTU	定时器的周期数据寄存器 B	TCDRB	16	16、32	5ICLK
000C 1A16h	MTU	定时器的死区时间数据寄存器 B	TDDRB	16	16	5ICLK
000C 1A18h	MTU6	定时器的通用寄存器 A	TGRA	16	16、32	5ICLK
000C 1A1Ah	MTU6	定时器的通用寄存器 B	TGRB	16	16	5ICLK
000C 1A1Ch	MTU7	定时器的通用寄存器 A	TGRA	16	16、32	5ICLK
000C 1A1Eh	MTU7	定时器的通用寄存器 B	TGRB	16	16	5ICLK
000C 1A20h	MTU	定时器的副计数器 B	TCNTSB	16	16、32	5ICLK
000C 1A22h	MTU	定时器的周期缓冲寄存器 B	TCBRB	16	16	5ICLK
000C 1A24h	MTU6	定时器的通用寄存器 C	TGRC	16	16、32	5ICLK
000C 1A26h	MTU6	定时器的通用寄存器 D	TGRD	16	16	5ICLK
000C 1A28h	MTU7	定时器的通用寄存器 C	TGRC	16	16、32	5ICLK
000C 1A2Ah	MTU7	定时器的通用寄存器 D	TGRD	16	16	5ICLK
000C 1A2Ch	MTU6	定时器的状态寄存器	TSR	8	8、16	5ICLK
000C 1A2Dh	MTU7	定时器的状态寄存器	TSR	8	8	5ICLK
000C 1A30h	MTU	定时器的中断减少设定寄存器 1B	TITCR1B	8	8、16	5ICLK
000C 1A31h	MTU	定时器的中断减少次数计数器 1B	TITCNT1B	8	8	5ICLK
000C 1A32h	MTU	定时器的缓冲传送设定寄存器 B	TBTERB	8	8	5ICLK
000C 1A34h	MTU	定时器的死区时间允许寄存器 B	TDERB	8	8	5ICLK
000C 1A36h	MTU	定时器的输出电平缓冲寄存器 B	TOLBRB	8	8	5ICLK
000C 1A38h	MTU6	定时器的缓冲运行传送模式寄存器	TBTM	8	8、16	5ICLK
000C 1A39h	MTU7	定时器的缓冲运行传送模式寄存器	TBTM	8	8	5ICLK
000C 1A3Ah	MTU	定时器的中断减少模式寄存器 B	TITMRB	8	8	5ICLK
000C 1A3Bh	MTU	定时器的中断减少设定寄存器 2B	TITCR2B	8	8	5ICLK
000C 1A3Ch	MTU	定时器的中断减少次数计数器 2B	TITCNT2B	8	8	5ICLK
000C 1A40h	MTU7	定时器的 A/D 转换开始请求控制寄存器	TADCR	16	16	5ICLK
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	5ICLK
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	5ICLK
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	5ICLK
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	5ICLK
000C 1A50h	MTU6	定时器的同步清除寄存器	TSYCR	8	8	5ICLK
000C 1A60h	MTU	定时器的波形控制寄存器 B	TWCRB	8	8	5ICLK
000C 1A70h	MTU	定时器的模式寄存器 2B	TMDR2B	8	8	5ICLK
000C 1A72h	MTU6	定时器的通用寄存器 E	TGRE	16	16	5ICLK
000C 1A74h	MTU7	定时器的通用寄存器 E	TGRE	16	16	5ICLK
000C 1A76h	MTU7	定时器的通用寄存器 F	TGRF	16	16	5ICLK
000C 1A80h	MTU	定时器的启动寄存器 B	TSTRB	8	8、16	5ICLK

表 5.1 I/O 寄存器地址一览表 (22 / 28)

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
000C 1A81h	MTU	定时器的同步寄存器 B	TSYRB	8	8	5ICLK
000C 1A84h	MTU	定时器的读写允许寄存器 B	TRWERB	8	8	5ICLK
000C 1C80h	MTU5	定时器的计数器 U	TCNTU	16	16、32	5ICLK
000C 1C82h	MTU5	定时器的通用寄存器 U	TGRU	16	16	5ICLK
000C 1C84h	MTU5	定时器的控制寄存器 U	TCRU	8	8	5ICLK
000C 1C86h	MTU5	定时器的 I/O 控制寄存器 U	TIORU	8	8	5ICLK
000C 1C90h	MTU5	定时器的计数器 V	TCNTV	16	16、32	5ICLK
000C 1C92h	MTU5	定时器的通用寄存器 V	TGRV	16	16	5ICLK
000C 1C94h	MTU5	定时器的控制寄存器 V	TCRV	8	8	5ICLK
000C 1C96h	MTU5	定时器的 I/O 控制寄存器 V	TIORV	8	8	5ICLK
000C 1CA0h	MTU5	定时器的计数器 W	TCNTW	16	16、32	5ICLK
000C 1CA2h	MTU5	定时器的通用寄存器 W	TGRW	16	16	5ICLK
000C 1CA4h	MTU5	定时器的控制寄存器 W	TCRW	8	8	5ICLK
000C 1CA6h	MTU5	定时器的 I/O 控制寄存器 W	TIORW	8	8	5ICLK
000C 1CB0h	MTU5	定时器的状态寄存器	TSR	8	8	5ICLK
000C 1CB2h	MTU5	定时器的中断允许寄存器	TIER	8	8	5ICLK
000C 1CB4h	MTU5	定时器的启动寄存器	TSTR	8	8	5ICLK
000C 1CB6h	MTU5	定时器的比较匹配清除寄存器	TCNTCMPCLR	8	8	5ICLK
000C 2000h	GPT	通用 PWM 定时器的软件启动寄存器	GTSTR	16	8、16、32	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
000C 200Ch	GPT	通用 PWM 定时器的写保护寄存器	GTWP	16	8、16、32	3 ~ 5ICLK (注 6)
000C 200Eh	GPT	通用 PWM 定时器的同步寄存器	GTSYNC	16	8、16、32	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
000C 2014h	GPT	通用 PWM 定时器的缓冲运行禁止寄存器	GTBDR	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2018h	GPT	通用 PWM 定时器启动的写保护寄存器	GTSWP	16	16、32	3 ~ 5ICLK (注 6)
000C 2080h	GPT	LOCO 计数控制寄存器	LCCR	16	8、16、32	3 ~ 5ICLK
000C 2082h	GPT	LOCO 计数状态寄存器	LCST	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2084h	GPT	LOCO 计数值寄存器	LCNT	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2086h	GPT	LOCO 计数结果平均寄存器	LCNTA	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2088h	GPT	LOCO 计数结果寄存器 0	LCNT00	16	8、16、32	3 ~ 5ICLK (注 6)
000C 208Ah	GPT	LOCO 计数结果寄存器 1	LCNT01	16	8、16、32	3 ~ 5ICLK (注 6)
000C 208Ch	GPT	LOCO 计数结果寄存器 2	LCNT02	16	8、16、32	3 ~ 5ICLK (注 6)
000C 208Eh	GPT	LOCO 计数结果寄存器 3	LCNT03	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2090h	GPT	LOCO 计数结果寄存器 4	LCNT04	16	8、16、32	3 ~ 5ICLK (注 6)

表 5.1 I/O 寄存器地址一览表 (23 / 28)

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
000C 2092h	GPT	LOCO 计数结果寄存器 5	LCNT05	16	8、16、32	3 ~ 5ICLK (注6)
000C 2094h	GPT	LOCO 计数结果寄存器 6	LCNT06	16	8、16、32	3 ~ 5ICLK (注6)
000C 2096h	GPT	LOCO 计数结果寄存器 7	LCNT07	16	8、16、32	3 ~ 5ICLK (注6)
000C 2098h	GPT	LOCO 计数结果寄存器 8	LCNT08	16	8、16、32	3 ~ 5ICLK (注6)
000C 209Ah	GPT	LOCO 计数结果寄存器 9	LCNT09	16	8、16、32	3 ~ 5ICLK (注6)
000C 209Ch	GPT	LOCO 计数结果寄存器 10	LCNT10	16	8、16、32	3 ~ 5ICLK (注6)
000C 209Eh	GPT	LOCO 计数结果寄存器 11	LCNT11	16	8、16、32	3 ~ 5ICLK (注6)
000C 20A0h	GPT	LOCO 计数结果寄存器 12	LCNT12	16	8、16、32	3 ~ 5ICLK (注6)
000C 20A2h	GPT	LOCO 计数结果寄存器 13	LCNT13	16	8、16、32	3 ~ 5ICLK (注6)
000C 20A4h	GPT	LOCO 计数结果寄存器 14	LCNT14	16	8、16、32	3 ~ 5ICLK (注6)
000C 20A6h	GPT	LOCO 计数结果寄存器 15	LCNT15	16	8、16、32	3 ~ 5ICLK (注6)
000C 20A8h	GPT	LOCO 计数上限容许偏差值寄存器	LCNTDU	16	8、16、32	3 ~ 5ICLK (注6)
000C 20AAh	GPT	LOCO 计数下限容许偏差值寄存器	LCNTDL	16	8、16、32	3 ~ 5ICLK (注6)
000C 2100h	GPT0	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	16	8、16、32	3 ~ 5ICLK (注6)
000C 2102h	GPT0	通用 PWM 定时器的中断输出设定寄存器	GTINTAD	16	8、16、32	3 ~ 5ICLK (注6)
000C 2104h	GPT0	通用 PWM 定时器的控制寄存器	GTCR	16	8、16、32	3 ~ 5ICLK (注6)
000C 2106h	GPT0	通用 PWM 定时器的缓冲允许寄存器	GTBER	16	8、16、32	3 ~ 5ICLK (注6)
000C 2108h	GPT0	通用 PWM 定时器的计数方向寄存器	GTUDC	16	8、16、32	3 ~ 5ICLK (注6)
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	3 ~ 5ICLK (注6)
000C 210Ch	GPT0	通用 PWM 定时器的状态寄存器	GTST	16	8、16、32	3 ~ 5ICLK (注6)
000C 210Eh	GPT0	通用 PWM 定时器的计数器	GTCNT	16	16	3 ~ 5ICLK (注6)
000C 2110h	GPT0	通用 PWM 定时器的比较捕捉寄存器 A	GTCCRA	16	16、32	3 ~ 5ICLK (注6)
000C 2112h	GPT0	通用 PWM 定时器的比较捕捉寄存器 B	GTCCRB	16	16、32	3 ~ 5ICLK (注6)
000C 2114h	GPT0	通用 PWM 定时器的比较捕捉寄存器 C	GTCCRC	16	16、32	3 ~ 5ICLK (注6)
000C 2116h	GPT0	通用 PWM 定时器的比较捕捉寄存器 D	GTCCRD	16	16、32	3 ~ 5ICLK (注6)
000C 2118h	GPT0	通用 PWM 定时器的比较捕捉寄存器 E	GTCCRE	16	16、32	3 ~ 5ICLK (注6)
000C 211Ah	GPT0	通用 PWM 定时器的比较捕捉寄存器 F	GTCCRF	16	16、32	3 ~ 5ICLK (注6)
000C 211Ch	GPT0	通用 PWM 定时器的周期设定寄存器	GTPR	16	16、32	3 ~ 5ICLK (注6)
000C 211Eh	GPT0	通用 PWM 定时器的周期设定缓冲寄存器	GTPBR	16	16、32	3 ~ 5ICLK (注6)
000C 2120h	GPT0	通用 PWM 定时器的周期设定双缓冲寄存器	GTPDBR	16	16、32	3 ~ 5ICLK (注6)

表 5.1 I/O 寄存器地址一览表 (24 / 28)

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
000C 2124h	GPT0	A/D 转换开始请求时序寄存器 A	GTADTRA	16	16、32	3 ~ 5ICLK (注 6)
000C 2126h	GPT0	A/D 转换开始请求时序缓冲寄存器 A	GTADTBRA	16	16、32	3 ~ 5ICLK (注 6)
000C 2128h	GPT0	A/D 转换开始请求时序双缓冲寄存器 A	GTADTDBRA	16	16、32	3 ~ 5ICLK (注 6)
000C 212Ch	GPT0	A/D 转换开始请求时序寄存器 B	GTADTRB	16	16、32	3 ~ 5ICLK (注 6)
000C 212Eh	GPT0	A/D 转换开始请求时序缓冲寄存器 B	GTADTBRB	16	16、32	3 ~ 5ICLK (注 6)
000C 2130h	GPT0	A/D 转换开始请求时序双缓冲寄存器 B	GTADTDBRB	16	16、32	3 ~ 5ICLK (注 6)
000C 2134h	GPT0	通用 PWM 定时器的输出无效控制寄存器	GTONCR	16	16、32	3 ~ 5ICLK (注 6)
000C 2136h	GPT0	通用 PWM 定时器的死区时间控制寄存器	GTDTCR	16	16、32	3 ~ 5ICLK (注 6)
000C 2138h	GPT0	通用 PWM 定时器的死区时间值寄存器	GTDVU	16	16、32	3 ~ 5ICLK (注 6)
000C 213Ah	GPT0	通用 PWM 定时器的死区时间值寄存器	GTDVD	16	16、32	3 ~ 5ICLK (注 6)
000C 213Ch	GPT0	通用 PWM 定时器的死区时间缓冲寄存器	GTDBU	16	16、32	3 ~ 5ICLK (注 6)
000C 213Eh	GPT0	通用 PWM 定时器的死区时间缓冲寄存器	GTDBD	16	16、32	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
000C 2180h	GPT1	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2182h	GPT1	通用 PWM 定时器的中断输出设定寄存器	GTINTAD	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2184h	GPT1	通用 PWM 定时器的控制寄存器	GTCR	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2186h	GPT1	通用 PWM 定时器的缓冲允许寄存器	GTBER	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2188h	GPT1	通用 PWM 定时器的计数方向寄存器	GTUDC	16	8、16、32	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
000C 218Ch	GPT1	通用 PWM 定时器的状态寄存器	GTST	16	8、16、32	3 ~ 5ICLK (注 6)
000C 218Eh	GPT1	通用 PWM 定时器的计数器	GTCNT	16	16	3 ~ 5ICLK (注 6)
000C 2190h	GPT1	通用 PWM 定时器的比较捕捉寄存器 A	GTCCRA	16	16、32	3 ~ 5ICLK (注 6)
000C 2192h	GPT1	通用 PWM 定时器的比较捕捉寄存器 B	GTCCRB	16	16、32	3 ~ 5ICLK (注 6)
000C 2194h	GPT1	通用 PWM 定时器的比较捕捉寄存器 C	GTCCRC	16	16、32	3 ~ 5ICLK (注 6)
000C 2196h	GPT1	通用 PWM 定时器的比较捕捉寄存器 D	GTCCRD	16	16、32	3 ~ 5ICLK (注 6)
000C 2198h	GPT1	通用 PWM 定时器的比较捕捉寄存器 E	GTCCRE	16	16、32	3 ~ 5ICLK (注 6)
000C 219Ah	GPT1	通用 PWM 定时器的比较捕捉寄存器 F	GTCCRF	16	16、32	3 ~ 5ICLK (注 6)
000C 219Ch	GPT1	通用 PWM 定时器的周期设定寄存器	GTPR	16	16、32	3 ~ 5ICLK (注 6)
000C 219Eh	GPT1	通用 PWM 定时器的周期设定缓冲寄存器	GTPBR	16	16、32	3 ~ 5ICLK (注 6)

表 5.1 I/O 寄存器地址一览表 (25 / 28)

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
000C 21A0h	GPT1	通用 PWM 定时器的周期设定双缓冲寄存器	GTPDBR	16	16、32	3 ~ 5ICLK (注 6)
000C 21A4h	GPT1	A/D 转换开始请求时序寄存器 A	GTADTRA	16	16、32	3 ~ 5ICLK (注 6)
000C 21A6h	GPT1	A/D 转换开始请求时序缓冲寄存器 A	GTADTBRA	16	16、32	3 ~ 5ICLK (注 6)
000C 21A8h	GPT1	A/D 转换开始请求时序双缓冲寄存器 A	GTADTDBRA	16	16、32	3 ~ 5ICLK (注 6)
000C 21ACh	GPT1	A/D 转换开始请求时序寄存器 B	GTADTRB	16	16、32	3 ~ 5ICLK (注 6)
000C 21AEh	GPT1	A/D 转换开始请求时序缓冲寄存器 B	GTADTBRB	16	16、32	3 ~ 5ICLK (注 6)
000C 21B0h	GPT1	A/D 转换开始请求时序双缓冲寄存器 B	GTADTDBRB	16	16、32	3 ~ 5ICLK (注 6)
000C 21B4h	GPT1	通用 PWM 定时器的输出无效控制寄存器	GTONCR	16	16、32	3 ~ 5ICLK (注 6)
000C 21B6h	GPT1	通用 PWM 定时器的死区时间控制寄存器	GTDTCR	16	16、32	3 ~ 5ICLK (注 6)
000C 21B8h	GPT1	通用 PWM 定时器的死区时间值寄存器	GTDVU	16	16、32	3 ~ 5ICLK (注 6)
000C 21BAh	GPT1	通用 PWM 定时器的死区时间值寄存器	GTDVD	16	16、32	3 ~ 5ICLK (注 6)
000C 21BCh	GPT1	通用 PWM 定时器的死区时间缓冲寄存器	GTDBU	16	16、32	3 ~ 5ICLK (注 6)
000C 21BEh	GPT1	通用 PWM 定时器的死区时间缓冲寄存器	GTDBD	16	16、32	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
000C 2200h	GPT2	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2202h	GPT2	通用 PWM 定时器的中断输出设定寄存器	GTINTAD	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2204h	GPT2	通用 PWM 定时器的控制寄存器	GTCR	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2206h	GPT2	通用 PWM 定时器的缓冲允许寄存器	GTBER	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2208h	GPT2	通用 PWM 定时器的计数方向寄存器	GTUDC	16	8、16、32	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
000C 220Ch	GPT2	通用 PWM 定时器的状态寄存器	GTST	16	8、16、32	3 ~ 5ICLK (注 6)
000C 220Eh	GPT2	通用 PWM 定时器的计数器	GTCNT	16	16	3 ~ 5ICLK (注 6)
000C 2210h	GPT2	通用 PWM 定时器的比较捕捉寄存器 A	GTCCRA	16	16、32	3 ~ 5ICLK (注 6)
000C 2212h	GPT2	通用 PWM 定时器的比较捕捉寄存器 B	GTCCRB	16	16、32	3 ~ 5ICLK (注 6)
000C 2214h	GPT2	通用 PWM 定时器的比较捕捉寄存器 C	GTCCRC	16	16、32	3 ~ 5ICLK (注 6)
000C 2216h	GPT2	通用 PWM 定时器的比较捕捉寄存器 D	GTCCRD	16	16、32	3 ~ 5ICLK (注 6)
000C 2218h	GPT2	通用 PWM 定时器的比较捕捉寄存器 E	GTCCRE	16	16、32	3 ~ 5ICLK (注 6)
000C 221Ah	GPT2	通用 PWM 定时器的比较捕捉寄存器 F	GTCCRF	16	16、32	3 ~ 5ICLK (注 6)
000C 221Ch	GPT2	通用 PWM 定时器的周期设定寄存器	GTPR	16	16、32	3 ~ 5ICLK (注 6)

表 5.1 I/O 寄存器地址一览表 (26 / 28)

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
000C 221Eh	GPT2	通用 PWM 定时器的周期设定缓冲寄存器	GTPBR	16	16、32	3 ~ 5ICLK (注 6)
000C 2220h	GPT2	通用 PWM 定时器的周期设定双缓冲寄存器	GTPDBR	16	16、32	3 ~ 5ICLK (注 6)
000C 2224h	GPT2	A/D 转换开始请求时序寄存器 A	GTADTRA	16	16、32	3 ~ 5ICLK (注 6)
000C 2226h	GPT2	A/D 转换开始请求时序缓冲寄存器 A	GTADTBRA	16	16、32	3 ~ 5ICLK (注 6)
000C 2228h	GPT2	A/D 转换开始请求时序双缓冲寄存器 A	GTADTDBRA	16	16、32	3 ~ 5ICLK (注 6)
000C 222Ch	GPT2	A/D 转换开始请求时序寄存器 B	GTADTRB	16	16、32	3 ~ 5ICLK (注 6)
000C 222Eh	GPT2	A/D 转换开始请求时序缓冲寄存器 B	GTADTBRB	16	16、32	3 ~ 5ICLK (注 6)
000C 2230h	GPT2	A/D 转换开始请求时序双缓冲寄存器 B	GTADTDBRB	16	16、32	3 ~ 5ICLK (注 6)
000C 2234h	GPT2	通用 PWM 定时器的输出无效控制寄存器	GTONCR	16	16、32	3 ~ 5ICLK (注 6)
000C 2236h	GPT2	通用 PWM 定时器的死区时间控制寄存器	GTDTCR	16	16、32	3 ~ 5ICLK (注 6)
000C 2238h	GPT2	通用 PWM 定时器的死区时间值寄存器	GTDVU	16	16、32	3 ~ 5ICLK (注 6)
000C 223Ah	GPT2	通用 PWM 定时器的死区时间值寄存器	GTDVD	16	16、32	3 ~ 5ICLK (注 6)
000C 223Ch	GPT2	通用 PWM 定时器的死区时间缓冲寄存器	GTDBU	16	16、32	3 ~ 5ICLK (注 6)
000C 223Eh	GPT2	通用 PWM 定时器的死区时间缓冲寄存器	GTDBD	16	16、32	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
000C 2280h	GPT3	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2282h	GPT3	通用 PWM 定时器的中断输出设定寄存器	GTINTAD	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2284h	GPT3	通用 PWM 定时器的控制寄存器	GTCR	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2286h	GPT3	通用 PWM 定时器的缓冲允许寄存器	GTBER	16	8、16、32	3 ~ 5ICLK (注 6)
000C 2288h	GPT3	通用 PWM 定时器的计数方向寄存器	GTUDC	16	8、16、32	3 ~ 5ICLK (注 6)
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	3 ~ 5ICLK (注 6)
000C 228Ch	GPT3	通用 PWM 定时器的状态寄存器	GTST	16	8、16、32	3 ~ 5ICLK (注 6)
000C 228Eh	GPT3	通用 PWM 定时器的计数器	GTCNT	16	16	3 ~ 5ICLK (注 6)
000C 2290h	GPT3	通用 PWM 定时器的比较捕捉寄存器 A	GTCCRA	16	16、32	3 ~ 5ICLK (注 6)
000C 2292h	GPT3	通用 PWM 定时器的比较捕捉寄存器 B	GTCCRB	16	16、32	3 ~ 5ICLK (注 6)
000C 2294h	GPT3	通用 PWM 定时器的比较捕捉寄存器 C	GTCCRC	16	16、32	3 ~ 5ICLK (注 6)
000C 2296h	GPT3	通用 PWM 定时器的比较捕捉寄存器 D	GTCCRD	16	16、32	3 ~ 5ICLK (注 6)
000C 2298h	GPT3	通用 PWM 定时器的比较捕捉寄存器 E	GTCCRE	16	16、32	3 ~ 5ICLK (注 6)
000C 229Ah	GPT3	通用 PWM 定时器的比较捕捉寄存器 F	GTCCRF	16	16、32	3 ~ 5ICLK (注 6)

表 5.1 I/O 寄存器地址一览表 (27 / 28)

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
000C 229Ch	GPT3	通用 PWM 定时器的周期设定寄存器	GTPR	16	16、32	3 ~ 5ICLK (注6)
000C 229Eh	GPT3	通用 PWM 定时器的周期设定缓冲寄存器	GTPBR	16	16、32	3 ~ 5ICLK (注6)
000C 22A0h	GPT3	通用 PWM 定时器的周期设定双缓冲寄存器	GTPDBR	16	16、32	3 ~ 5ICLK (注6)
000C 22A4h	GPT3	A/D 转换开始请求时序寄存器 A	GTADTRA	16	16、32	3 ~ 5ICLK (注6)
000C 22A6h	GPT3	A/D 转换开始请求时序缓冲寄存器 A	GTADTBRA	16	16、32	3 ~ 5ICLK (注6)
000C 22A8h	GPT3	A/D 转换开始请求时序双缓冲寄存器 A	GTADTDBRA	16	16、32	3 ~ 5ICLK (注6)
000C 22ACh	GPT3	A/D 转换开始请求时序寄存器 B	GTADTRB	16	16、32	3 ~ 5ICLK (注6)
000C 22AEh	GPT3	A/D 转换开始请求时序缓冲寄存器 B	GTADTBRB	16	16、32	3 ~ 5ICLK (注6)
000C 22B0h	GPT3	A/D 转换开始请求时序双缓冲寄存器 B	GTADTDBRB	16	16、32	3 ~ 5ICLK (注6)
000C 22B4h	GPT3	通用 PWM 定时器的输出无效控制寄存器	GTONCR	16	16、32	3 ~ 5ICLK (注6)
000C 22B6h	GPT3	通用 PWM 定时器的死区时间控制寄存器	GTDTCR	16	16、32	3 ~ 5ICLK (注6)
000C 22B8h	GPT3	通用 PWM 定时器的死区时间值寄存器	GTDVU	16	16、32	3 ~ 5ICLK (注6)
000C 22BAh	GPT3	通用 PWM 定时器的死区时间值寄存器	GTDVD	16	16、32	3 ~ 5ICLK (注6)
000C 22BCh	GPT3	通用 PWM 定时器的死区时间缓冲寄存器	GTDBU	16	16、32	3 ~ 5ICLK (注6)
000C 22BEh	GPT3	通用 PWM 定时器的死区时间缓冲寄存器	GTDBD	16	16、32	3 ~ 5ICLK (注6)
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	3 ~ 5ICLK (注6)
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	3 ~ 5ICLK (注6)
007F C402h	FLASH	闪存模式寄存器	FMODR	8	8	2 ~ 3 个 PCLK (注5)
007F C410h	FLASH	闪存存取状态寄存器	FASTAT	8	8	2 ~ 3 个 PCLK (注5)
007F C411h	FLASH	闪存存取错误中断允许寄存器	FAEINT	8	8	2 ~ 3 个 PCLK (注5)
007F C412h	FLASH	闪存就绪中断允许寄存器	FRDYIE	8	8	2 ~ 3 个 PCLK (注5)
007F C440h	FLASH	数据闪存读允许寄存器 0	DFLRE0	16	16	2 ~ 3 个 PCLK (注5)
007F C442h	FLASH	数据闪存读允许寄存器 1	DFLRE1	16	16	2 ~ 3 个 PCLK (注5)
007F C450h	FLASH	数据闪存编程 / 擦除允许寄存器 0	DFLWE0	16	16	2 ~ 3 个 PCLK (注5)
007F C452h	FLASH	数据闪存编程 / 擦除允许寄存器 1	DFLWE1	16	16	2 ~ 3 个 PCLK (注5)
007F C454h	FLASH	FCU RAM 允许寄存器	FCURAME	16	16	2 ~ 3 个 PCLK (注5)
007F FFB0h	FLASH	闪存状态寄存器 0	FSTATR0	8	8	2 ~ 3 个 PCLK (注5)
007F FFB1h	FLASH	闪存状态寄存器 1	FSTATR1	8	8	2 ~ 3 个 PCLK (注5)
007F FFB2h	FLASH	闪存 P/E 模式入口寄存器	FENTRYR	16	16	2 ~ 3 个 PCLK (注5)
007F FFB4h	FLASH	闪存保护寄存器	FPROTR	16	16	2 ~ 3 个 PCLK (注5)



表 5.1 I/O 寄存器地址一览表 (28 / 28)

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
007F FFB6h	FLASH	闪存复位寄存器	FRESETR	16	16	2 ~ 3 个 PCLK (注5)
007F FFBAh	FLASH	FCU 命令寄存器	FCMDR	16	16	2 ~ 3 个 PCLK (注5)
007F FFC8h	FLASH	FCU 处理转换寄存器	FCPSR	16	16	2 ~ 3 个 PCLK (注5)
007F FFCAh	FLASH	数据闪存空白检查控制寄存器	DFLBCCNT	16	16	2 ~ 3 个 PCLK (注5)
007F FFCCh	FLASH	闪存 P/E 状态寄存器	FPESTAT	16	16	2 ~ 3 个 PCLK (注5)
007F FFCEh	FLASH	数据闪存空白检查控制状态寄存器	DFLBCSTAT	16	16	2 ~ 3 个 PCLK (注5)
007F FFE8h	FLASH	外围时钟通知寄存器	PCKAR	16	16	2 ~ 3 个 PCLK (注5)

注 1. 不支持 100 引脚 LQFP 版。

注 2. 不支持 80 引脚 LQFP 版。

注 3. 不支持 64 引脚 LQFP 版。

注 4. 不支持无 CAN 功能版。

注 5. 因分频时钟的同步周期 (0 ~ 1 个 PCLK) 而不同。

注 6. 寄存器写周期数为 3 个 ICLK, 寄存器读周期数为 5 个 ICLK。

## 5.2 I/O 寄存器位一览表

外围模块的寄存器的地址和位名如下所示。

16 位寄存器或者 32 位寄存器分别以 8 位分 2 段或者 4 段表示。

表 5.2 I/O 寄存器位一览表 (1 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
SYSTEM	MDMONR	—	—	—	—	—	—	—	—
		MDE	—	—	—	—	—	MD1	MD0
SYSTEM	MDSR	—	—	—	—	—	—	—	—
		—	—	—	BOTS	—	—	—	IROM
SYSTEM	SYSCR0	KEY[7:0]							
		—	—	—	—	—	—	—	ROME
SYSTEM	SYSCR1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	RAME
SYSTEM	SBYCR	SSBY	—	—	STS[4:0]				
		—	—	—	—	—	—	—	—
SYSTEM	MSTPCRA	ACSE	—	—	MSTPA28	—	—	—	MSTPA24
		MSTPA23	—	—	—	—	—	MSTPA17	MSTPA16
		MSTPA15	MSTPA14	—	—	—	—	MSTPA9	—
		MSTPA7	—	—	—	—	—	—	—
SYSTEM	MSTPCRB	MSTPB31	MSTPB30	MSTPB29	—	—	—	—	—
		MSTPB23	—	MSTPB21	—	—	—	MSTPB17	—
		—	—	—	—	—	—	—	—
		MSTPB7	—	—	—	—	—	—	MSTPB0
SYSTEM	MSTPCRC	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MSTPC0
SYSTEM	SCKCR	—	—	—	—	ICK[3:0]			
		—	—	—	—	—	—	—	—
		—	—	—	—	PCK[3:0]			
		—	—	—	—	—	—	—	—
SYSTEM	OSTDCR	KEY[7:0]							
		OSTDE	OSTDF	—	—	—	—	—	—
BSC	BERCLR	—	—	—	—	—	—	—	STSCCLR
BSC	BEREN	—	—	—	—	—	—	—	IGAEN
BSC	BERSR1	—	MST[2:0]			—	—	—	IA
BSC	BERSR2	ADDR[12:0]							
		ADDR[12:0]						—	—
DTC	DTCCR	—	—	—	RRS	—	—	—	—
DTC	DTCVBR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
DTC	DTCADMOD	—	—	—	—	—	—	—	SHORT
DTC	DTCST	—	—	—	—	—	—	—	DTCST
DTC	DTCSTS	ACT	—	—	—	—	—	—	—
		VECN[7:0]							

表 5.2 I/O 寄存器位一览表 (2 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
MPU	RSPAGE0	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—		—	
MPU	REPAGE0	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			
MPU	RSPAGE1	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—		—	
MPU	REPAGE1	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			
MPU	RSPAGE2	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—		—	
MPU	REPAGE2	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			
MPU	RSPAGE3	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—		—	
MPU	REPAGE3	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			
MPU	RSPAGE4	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—		—	
MPU	REPAGE4	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			
MPU	RSPAGE5	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—		—	
MPU	REPAGE5	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			

表 5.2 I/O 寄存器位一览表 (3 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
MPU	RSPAGE6	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—	—	—	—
MPU	REPAGE6	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			V
MPU	RSPAGE7	RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]							
		RSPN[27:0]				—	—	—	—
MPU	REPAGE7	REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]							
		REPN[27:0]				UAC[2:0]			V
MPU	MPEN	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	MPEN
MPU	MPBAC	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	UBAC[2:0]			—
MPU	MPECLR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	CLR
MPU	MPESTS	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	DRW	DA	IA
MPU	MPDEA	DEA[31:0]							
		DEA[31:0]							
		DEA[31:0]							
		DEA[31:0]							
MPU	MPSA	SA[31:0]							
		SA[31:0]							
		SA[31:0]							
		SA[31:0]							
MPU	MPOPS	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	S
MPU	MPOPI	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	INV
MPU	MHITI	—	—	—	—	—	—	—	—
		HIT[7:0]							
		—	—	—	—	—	—	—	—
		—	—	—	—	UHACI[2:0]			—

表 5.2 I/O 寄存器位一览表 (4 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
MPU	MHITD	—	—	—	—	—	—	—	—
		HITD[7:0]							
		—	—	—	—	—	—	—	—
		UHACD[2:0]						—	—
ICU	IR016	—	—	—	—	—	—	—	IR
ICU	IR021	—	—	—	—	—	—	—	IR
ICU	IR023	—	—	—	—	—	—	—	IR
ICU	IR027	—	—	—	—	—	—	—	IR
ICU	IR028	—	—	—	—	—	—	—	IR
ICU	IR029	—	—	—	—	—	—	—	IR
ICU	IR030	—	—	—	—	—	—	—	IR
ICU	IR031	—	—	—	—	—	—	—	IR
ICU	IR044	—	—	—	—	—	—	—	IR
ICU	IR045	—	—	—	—	—	—	—	IR
ICU	IR046	—	—	—	—	—	—	—	IR
ICU	IR047	—	—	—	—	—	—	—	IR
ICU	IR056	—	—	—	—	—	—	—	IR
ICU	IR057	—	—	—	—	—	—	—	IR
ICU	IR058	—	—	—	—	—	—	—	IR
ICU	IR059	—	—	—	—	—	—	—	IR
ICU	IR060	—	—	—	—	—	—	—	IR
ICU	IR064	—	—	—	—	—	—	—	IR
ICU	IR065	—	—	—	—	—	—	—	IR
ICU	IR066	—	—	—	—	—	—	—	IR
ICU	IR067	—	—	—	—	—	—	—	IR
ICU	IR068	—	—	—	—	—	—	—	IR
ICU	IR069	—	—	—	—	—	—	—	IR
ICU	IR070	—	—	—	—	—	—	—	IR
ICU	IR071	—	—	—	—	—	—	—	IR
ICU	IR096	—	—	—	—	—	—	—	IR
ICU	IR098	—	—	—	—	—	—	—	IR
ICU	IR102	—	—	—	—	—	—	—	IR
ICU	IR103	—	—	—	—	—	—	—	IR
ICU	IR106	—	—	—	—	—	—	—	IR
ICU	IR114	—	—	—	—	—	—	—	IR
ICU	IR115	—	—	—	—	—	—	—	IR
ICU	IR116	—	—	—	—	—	—	—	IR
ICU	IR117	—	—	—	—	—	—	—	IR
ICU	IR118	—	—	—	—	—	—	—	IR
ICU	IR119	—	—	—	—	—	—	—	IR
ICU	IR120	—	—	—	—	—	—	—	IR
ICU	IR121	—	—	—	—	—	—	—	IR
ICU	IR122	—	—	—	—	—	—	—	IR
ICU	IR123	—	—	—	—	—	—	—	IR
ICU	IR124	—	—	—	—	—	—	—	IR
ICU	IR125	—	—	—	—	—	—	—	IR
ICU	IR126	—	—	—	—	—	—	—	IR
ICU	IR127	—	—	—	—	—	—	—	IR
ICU	IR128	—	—	—	—	—	—	—	IR
ICU	IR129	—	—	—	—	—	—	—	IR

表 5.2 I/O 寄存器位一览表 (5 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
ICU	IR130	—	—	—	—	—	—	—	IR
ICU	IR131	—	—	—	—	—	—	—	IR
ICU	IR132	—	—	—	—	—	—	—	IR
ICU	IR133	—	—	—	—	—	—	—	IR
ICU	IR134	—	—	—	—	—	—	—	IR
ICU	IR135	—	—	—	—	—	—	—	IR
ICU	IR136	—	—	—	—	—	—	—	IR
ICU	IR137	—	—	—	—	—	—	—	IR
ICU	IR138	—	—	—	—	—	—	—	IR
ICU	IR139	—	—	—	—	—	—	—	IR
ICU	IR140	—	—	—	—	—	—	—	IR
ICU	IR141	—	—	—	—	—	—	—	IR
ICU	IR142	—	—	—	—	—	—	—	IR
ICU	IR143	—	—	—	—	—	—	—	IR
ICU	IR144	—	—	—	—	—	—	—	IR
ICU	IR145	—	—	—	—	—	—	—	IR
ICU	IR146	—	—	—	—	—	—	—	IR
ICU	IR149	—	—	—	—	—	—	—	IR
ICU	IR150	—	—	—	—	—	—	—	IR
ICU	IR151	—	—	—	—	—	—	—	IR
ICU	IR152	—	—	—	—	—	—	—	IR
ICU	IR153	—	—	—	—	—	—	—	IR
ICU	IR170	—	—	—	—	—	—	—	IR
ICU	IR171	—	—	—	—	—	—	—	IR
ICU	IR172	—	—	—	—	—	—	—	IR
ICU	IR173	—	—	—	—	—	—	—	IR
ICU	IR174	—	—	—	—	—	—	—	IR
ICU	IR175	—	—	—	—	—	—	—	IR
ICU	IR176	—	—	—	—	—	—	—	IR
ICU	IR177	—	—	—	—	—	—	—	IR
ICU	IR178	—	—	—	—	—	—	—	IR
ICU	IR179	—	—	—	—	—	—	—	IR
ICU	IR180	—	—	—	—	—	—	—	IR
ICU	IR181	—	—	—	—	—	—	—	IR
ICU	IR182	—	—	—	—	—	—	—	IR
ICU	IR183	—	—	—	—	—	—	—	IR
ICU	IR184	—	—	—	—	—	—	—	IR
ICU	IR186	—	—	—	—	—	—	—	IR
ICU	IR187	—	—	—	—	—	—	—	IR
ICU	IR188	—	—	—	—	—	—	—	IR
ICU	IR189	—	—	—	—	—	—	—	IR
ICU	IR190	—	—	—	—	—	—	—	IR
ICU	IR192	—	—	—	—	—	—	—	IR
ICU	IR193	—	—	—	—	—	—	—	IR
ICU	IR194	—	—	—	—	—	—	—	IR
ICU	IR195	—	—	—	—	—	—	—	IR
ICU	IR196	—	—	—	—	—	—	—	IR
ICU	IR214	—	—	—	—	—	—	—	IR
ICU	IR215	—	—	—	—	—	—	—	IR
ICU	IR216	—	—	—	—	—	—	—	IR

表 5.2 I/O 寄存器位一览表 (6 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
ICU	IR217	—	—	—	—	—	—	—	IR
ICU	IR218	—	—	—	—	—	—	—	IR
ICU	IR219	—	—	—	—	—	—	—	IR
ICU	IR220	—	—	—	—	—	—	—	IR
ICU	IR221	—	—	—	—	—	—	—	IR
ICU	IR222	—	—	—	—	—	—	—	IR
ICU	IR223	—	—	—	—	—	—	—	IR
ICU	IR224	—	—	—	—	—	—	—	IR
ICU	IR225	—	—	—	—	—	—	—	IR
ICU	IR246	—	—	—	—	—	—	—	IR
ICU	IR247	—	—	—	—	—	—	—	IR
ICU	IR248	—	—	—	—	—	—	—	IR
ICU	IR249	—	—	—	—	—	—	—	IR
ICU	IR254	—	—	—	—	—	—	—	IR
ICU	DTCER027	—	—	—	—	—	—	—	DTCE
ICU	DTCER028	—	—	—	—	—	—	—	DTCE
ICU	DTCER029	—	—	—	—	—	—	—	DTCE
ICU	DTCER030	—	—	—	—	—	—	—	DTCE
ICU	DTCER031	—	—	—	—	—	—	—	DTCE
ICU	DTCER045	—	—	—	—	—	—	—	DTCE
ICU	DTCER046	—	—	—	—	—	—	—	DTCE
ICU	DTCER064	—	—	—	—	—	—	—	DTCE
ICU	DTCER065	—	—	—	—	—	—	—	DTCE
ICU	DTCER066	—	—	—	—	—	—	—	DTCE
ICU	DTCER067	—	—	—	—	—	—	—	DTCE
ICU	DTCER068	—	—	—	—	—	—	—	DTCE
ICU	DTCER069	—	—	—	—	—	—	—	DTCE
ICU	DTCER070	—	—	—	—	—	—	—	DTCE
ICU	DTCER071	—	—	—	—	—	—	—	DTCE
ICU	DTCER098	—	—	—	—	—	—	—	DTCE
ICU	DTCER102	—	—	—	—	—	—	—	DTCE
ICU	DTCER103	—	—	—	—	—	—	—	DTCE
ICU	DTCER106	—	—	—	—	—	—	—	DTCE
ICU	DTCER114	—	—	—	—	—	—	—	DTCE
ICU	DTCER115	—	—	—	—	—	—	—	DTCE
ICU	DTCER116	—	—	—	—	—	—	—	DTCE
ICU	DTCER117	—	—	—	—	—	—	—	DTCE
ICU	DTCER121	—	—	—	—	—	—	—	DTCE
ICU	DTCER122	—	—	—	—	—	—	—	DTCE
ICU	DTCER125	—	—	—	—	—	—	—	DTCE
ICU	DTCER126	—	—	—	—	—	—	—	DTCE
ICU	DTCER129	—	—	—	—	—	—	—	DTCE
ICU	DTCER130	—	—	—	—	—	—	—	DTCE
ICU	DTCER131	—	—	—	—	—	—	—	DTCE
ICU	DTCER132	—	—	—	—	—	—	—	DTCE
ICU	DTCER134	—	—	—	—	—	—	—	DTCE
ICU	DTCER135	—	—	—	—	—	—	—	DTCE
ICU	DTCER136	—	—	—	—	—	—	—	DTCE
ICU	DTCER137	—	—	—	—	—	—	—	DTCE
ICU	DTCER138	—	—	—	—	—	—	—	DTCE

表 5.2 I/O 寄存器位一览表 (7 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
ICU	DTCER139	—	—	—	—	—	—	—	DTCE
ICU	DTCER140	—	—	—	—	—	—	—	DTCE
ICU	DTCER141	—	—	—	—	—	—	—	DTCE
ICU	DTCER142	—	—	—	—	—	—	—	DTCE
ICU	DTCER143	—	—	—	—	—	—	—	DTCE
ICU	DTCER144	—	—	—	—	—	—	—	DTCE
ICU	DTCER145	—	—	—	—	—	—	—	DTCE
ICU	DTCER149	—	—	—	—	—	—	—	DTCE
ICU	DTCER150	—	—	—	—	—	—	—	DTCE
ICU	DTCER151	—	—	—	—	—	—	—	DTCE
ICU	DTCER152	—	—	—	—	—	—	—	DTCE
ICU	DTCER153	—	—	—	—	—	—	—	DTCE
ICU	DTCER174	—	—	—	—	—	—	—	DTCE
ICU	DTCER175	—	—	—	—	—	—	—	DTCE
ICU	DTCER176	—	—	—	—	—	—	—	DTCE
ICU	DTCER177	—	—	—	—	—	—	—	DTCE
ICU	DTCER178	—	—	—	—	—	—	—	DTCE
ICU	DTCER179	—	—	—	—	—	—	—	DTCE
ICU	DTCER180	—	—	—	—	—	—	—	DTCE
ICU	DTCER181	—	—	—	—	—	—	—	DTCE
ICU	DTCER182	—	—	—	—	—	—	—	DTCE
ICU	DTCER183	—	—	—	—	—	—	—	DTCE
ICU	DTCER184	—	—	—	—	—	—	—	DTCE
ICU	DTCER186	—	—	—	—	—	—	—	DTCE
ICU	DTCER187	—	—	—	—	—	—	—	DTCE
ICU	DTCER188	—	—	—	—	—	—	—	DTCE
ICU	DTCER189	—	—	—	—	—	—	—	DTCE
ICU	DTCER190	—	—	—	—	—	—	—	DTCE
ICU	DTCER192	—	—	—	—	—	—	—	DTCE
ICU	DTCER193	—	—	—	—	—	—	—	DTCE
ICU	DTCER194	—	—	—	—	—	—	—	DTCE
ICU	DTCER195	—	—	—	—	—	—	—	DTCE
ICU	DTCER196	—	—	—	—	—	—	—	DTCE
ICU	DTCER215	—	—	—	—	—	—	—	DTCE
ICU	DTCER216	—	—	—	—	—	—	—	DTCE
ICU	DTCER219	—	—	—	—	—	—	—	DTCE
ICU	DTCER220	—	—	—	—	—	—	—	DTCE
ICU	DTCER223	—	—	—	—	—	—	—	DTCE
ICU	DTCER224	—	—	—	—	—	—	—	DTCE
ICU	DTCER247	—	—	—	—	—	—	—	DTCE
ICU	DTCER248	—	—	—	—	—	—	—	DTCE
ICU	DTCER254	—	—	—	—	—	—	—	DTCE
ICU	IER02	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER03	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER05	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER07	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER08	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER0C	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER0D	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER0E	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0



表 5.2 I/O 寄存器位一览表 (8 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
ICU	IER0F	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER10	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER11	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER12	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER13	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER15	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER16	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER17	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER18	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER1A	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER1B	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER1C	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER1E	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	IER1F	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
ICU	SWINTR	—	—	—	—	—	—	—	SWINT
ICU	FIR	FIEN	—	—	—	—	—	—	—
		FVCT[7:0]							
ICU	IPR00	—	—	—	—	IPR[3:0]			
ICU	IPR01	—	—	—	—	IPR[3:0]			
ICU	IPR02	—	—	—	—	IPR[3:0]			
ICU	IPR03	—	—	—	—	IPR[3:0]			
ICU	IPR04	—	—	—	—	IPR[3:0]			
ICU	IPR05	—	—	—	—	IPR[3:0]			
ICU	IPR06	—	—	—	—	IPR[3:0]			
ICU	IPR07	—	—	—	—	IPR[3:0]			
ICU	IPR14	—	—	—	—	IPR[3:0]			
ICU	IPR18	—	—	—	—	IPR[3:0]			
ICU	IPR20	—	—	—	—	IPR[3:0]			
ICU	IPR21	—	—	—	—	IPR[3:0]			
ICU	IPR22	—	—	—	—	IPR[3:0]			
ICU	IPR23	—	—	—	—	IPR[3:0]			
ICU	IPR24	—	—	—	—	IPR[3:0]			
ICU	IPR25	—	—	—	—	IPR[3:0]			
ICU	IPR26	—	—	—	—	IPR[3:0]			
ICU	IPR27	—	—	—	—	IPR[3:0]			
ICU	IPR40	—	—	—	—	IPR[3:0]			
ICU	IPR44	—	—	—	—	IPR[3:0]			
ICU	IPR48	—	—	—	—	IPR[3:0]			
ICU	IPR49	—	—	—	—	IPR[3:0]			
ICU	IPR51	—	—	—	—	IPR[3:0]			
ICU	IPR52	—	—	—	—	IPR[3:0]			
ICU	IPR53	—	—	—	—	IPR[3:0]			
ICU	IPR54	—	—	—	—	IPR[3:0]			
ICU	IPR55	—	—	—	—	IPR[3:0]			
ICU	IPR56	—	—	—	—	IPR[3:0]			
ICU	IPR57	—	—	—	—	IPR[3:0]			
ICU	IPR58	—	—	—	—	IPR[3:0]			
ICU	IPR59	—	—	—	—	IPR[3:0]			
ICU	IPR5A	—	—	—	—	IPR[3:0]			
ICU	IPR5B	—	—	—	—	IPR[3:0]			

表 5.2 I/O 寄存器位一览表 (9 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
ICU	IPR5C	—	—	—	—	IPR[3:0]			
ICU	IPR5D	—	—	—	—	IPR[3:0]			
ICU	IPR5E	—	—	—	—	IPR[3:0]			
ICU	IPR5F	—	—	—	—	IPR[3:0]			
ICU	IPR60	—	—	—	—	IPR[3:0]			
ICU	IPR67	—	—	—	—	IPR[3:0]			
ICU	IPR68	—	—	—	—	IPR[3:0]			
ICU	IPR69	—	—	—	—	IPR[3:0]			
ICU	IPR6A	—	—	—	—	IPR[3:0]			
ICU	IPR6B	—	—	—	—	IPR[3:0]			
ICU	IPR6C	—	—	—	—	IPR[3:0]			
ICU	IPR6D	—	—	—	—	IPR[3:0]			
ICU	IPR6E	—	—	—	—	IPR[3:0]			
ICU	IPR6F	—	—	—	—	IPR[3:0]			
ICU	IPR80	—	—	—	—	IPR[3:0]			
ICU	IPR81	—	—	—	—	IPR[3:0]			
ICU	IPR82	—	—	—	—	IPR[3:0]			
ICU	IPR88	—	—	—	—	IPR[3:0]			
ICU	IPR89	—	—	—	—	IPR[3:0]			
ICU	IPR8A	—	—	—	—	IPR[3:0]			
ICU	IPR8B	—	—	—	—	IPR[3:0]			
ICU	IPR90	—	—	—	—	IPR[3:0]			
ICU	IRQCR0	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR1	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR2	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR3	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR4	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR5	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR6	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	IRQCR7	—	—	—	—	IRQMD[1:0]	—	—	—
ICU	NMISR	—	—	—	—	—	OSTST	LVDST	NMIST
ICU	NMIER	—	—	—	—	—	OSTEN	LVDEN	NMIEN
ICU	NMICLR	—	—	—	—	—	OSTCLR	—	NMICLR
ICU	NMICR	—	—	—	—	NMIMD	—	—	—
CMT	CMSTR0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR1	STR0
CMT0	CMCR	—	—	—	—	—	—	—	—
		—	CMIE	—	—	—	—	—	CKS[1:0]
CMT0	CMCNT								
CMT0	CMCOR								
CMT1	CMCR	—	—	—	—	—	—	—	—
		—	CMIE	—	—	—	—	—	CKS[1:0]
CMT1	CMCNT								
CMT1	CMCOR								
CMT	CMSTR1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR3	STR2

表 5.2 I/O 寄存器位一览表 (10 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
CMT2	CMCR	—	—	—	—	—	—	—	—
		—	CMIE	—	—	—	—	CKS[1:0]	
CMT2	CMCNT								
CMT2	CMCOR								
CMT3	CMCR	—	—	—	—	—	—	—	—
		—	CMIE	—	—	—	—	CKS[1:0]	
CMT3	CMCNT								
CMT3	CMCOR								
WDT	TCSR	—	TMS	TME	—	—	CKS[2:0]		
WDT	WINA								
WDT	TCNT								
WDT	WINB								
WDT	RSTCSR	WOVF	RSTE	—	—	—	—	—	—
IWDT	IWDTRR								
IWDT	IWDTCR	—	—	—	—	—	—	—	—
		CKS[3:0]				—	—	TOPS[1:0]	
IWDT	IWDTSR	—	UNDF	CNTVAL[13:0]					
		CNTVAL[13:0]							
AD0	ADDRA (注1)	—	—	—	—	—	—		
AD0	ADDRB (注1)	—	—	—	—	—	—		
AD0	ADDRC (注1)	—	—	—	—	—	—		
AD0	ADDRD (注1)	—	—	—	—	—	—		
AD0	ADDRE (注1)	—	—	—	—	—	—		
AD0	ADDRF (注1)	—	—	—	—	—	—		
AD0	ADDRG (注1)	—	—	—	—	—	—		
AD0	ADDRH (注1)	—	—	—	—	—	—		
AD0	ADCSR	—	ADIE	ADST	—	CH[3:0]			
AD0	ADCR	—	—	—	—	CKS[1:0]		MODE[1:0]	
AD0	ADSSTR								
AD0	ADDIAGR	—	—	—	—	—	—	DIAG[1:0]	
AD0	ADDRI (注1)	—	—	—	—	—	—		
AD0	ADDRJ (注1)	—	—	—	—	—	—		
AD0	ADDRK (注1)	—	—	—	—	—	—		

表 5.2 I/O 寄存器位一览表 (11 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
AD0	ADDRL (注1)	—	—	—	—	—	—		
AD0	ADSTRGR	—	—	—	ADSTRS[4:0]				
AD0	ADDPR	DPSEL	—	—	—	—	—	—	DPPRC
SCI0	SMR	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
SCI0	BRR								
SCI0	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SCI0	TDR								
SCI0	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
SCI0	RDR								
SCI0	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI0	SEMR	—	—	NFEN	ABCS	—	—	—	—
SMCI0	SMR	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
SMCI0	BRR								
SMCI0	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SMCI0	TDR								
SMCI0	SSR	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
SMCI0	RDR								
SMCI0	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI1	SMR	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
SCI1	BRR								
SCI1	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SCI1	TDR								
SCI1	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
SCI1	RDR								
SCI1	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI1	SEMR	—	—	NFEN	ABCS	—	—	—	—
SMCI1	SMR	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
SMCI1	BRR								
SMCI1	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SMCI1	TDR								
SMCI1	SSR	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
SMCI1	RDR								
SMCI1	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI2	SMR	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
SCI2	BRR								
SCI2	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SCI2	TDR								
SCI2	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
SCI2	RDR								
SCI2	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
SCI2	SEMR	—	—	NFEN	ABCS	—	—	—	—
SMCI2	SMR	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
SMCI2	BRR								
SMCI2	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
SMCI2	TDR								
SMCI2	SSR	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
SMCI2	RDR								
SMCI2	SCMR	BCP2	—	—	—	SDIR	SINV	—	SMIF
CRC	CRCCR	DORCLR	—	—	—	—	LMS	GPS[1:0]	

表 5.2 I/O 寄存器位一览表 (12 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
CRC	CRCDIR								
CRC	CRCDOR								
RIIC0	ICCR1	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
RIIC0	ICCR2	BBSY	MST	TRS	—	SP	RS	ST	—
RIIC0	ICMR1	MTWP	CKS[2:0]			BCWP	BC[2:0]		
RIIC0	ICMR2	DLCS	SDDL[2:0]			TMWE	TMOH	TMOL	TMOS
RIIC0	ICMR3	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
RIIC0	ICFER	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
RIIC0	ICSER	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
RIIC0	ICIER	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
RIIC0	ICSR1	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
RIIC0	ICSR2	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
RIIC0	SARL0	SVA[6:0]							SVA0
RIIC0	TMOCNTL								
RIIC0	SARU0	—	—	—	—	—	SVA[1:0]		FS
RIIC0	TMOCNTU								
RIIC0	SARL1	SVA[6:0]							SVA0
RIIC0	SARU1	—	—	—	—	—	SVA[1:0]		FS
RIIC0	SARL2	SVA[6:0]							SVA0
RIIC0	SARU2	—	—	—	—	—	SVA[1:0]		FS
RIIC0	ICBRL	—	—	—	BRL[4:0]				
RIIC0	ICBRH	—	—	—	BRH[4:0]				
RIIC0	ICDRT								
RIIC0	ICDRR								
RSPI0	SPCR	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
RSPI0	SSLP	—	—	—	—	SSLP3	SSLP2	SSLP1	SSLP0
RSPI0	SPPCR	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
RSPI0	SPSR	SPRF	—	SPTF	—	PERF	MODF	IDLNF	OVRF
RSPI0	SPDR	H[15:0]							
		H[15:0]							
		L[15:0]							
		L[15:0]							
RSPI0	SPSCR	—	—	—	—	—	SPSLN[2:0]		
RSPI0	SPSSR	—	SPECM[2:0]			—	SPCP[2:0]		
RSPI0	SPBR	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
RSPI0	SPDCR	—	—	SPLW	SPRDTD	SLSEL[1:0]		SPFC[1:0]	
RSPI0	SPCKD	—	—	—	—	—	SCKDL[2:0]		
RSPI0	SSLND	—	—	—	—	—	SLNDL[2:0]		
RSPI0	SPND	—	—	—	—	—	SPNDL[2:0]		
RSPI0	SPCR2	—	—	—	—	PTE	SPIIE	SPOE	SPPE
RSPI0	SPCMD0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI0	SPCMD1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI0	SPCMD2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	
RSPI0	SPCMD3	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]			BRDV[1:0]	CPOL	CPHA	

表 5.2 I/O 寄存器位一览表 (13 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
RSPI0	SPCMD4	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]		BRDV[1:0]		CPOL	CPHA	
RSPI0	SPCMD5	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]		BRDV[1:0]		CPOL	CPHA	
RSPI0	SPCMD6	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]		BRDV[1:0]		CPOL	CPHA	
RSPI0	SPCMD7	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			
		SSLKP	SSLA[2:0]		BRDV[1:0]		CPOL	CPHA	
S12AD0	ADCSR	ADST	ADCS[1:0]		ADIE	CKS[1:0]		TRGE	EXTRG
S12AD0	ADANS	—	—	CH[1:0]		—	PG002SEL	PG001SEL	PG000SEL
		—	—	—	—	—	PG002EN	PG001EN	PG000EN
S12AD0	ADPG	—	—	—	—	PG002GAIN[3:0]			
		PG001GAIN[3:0]			PG000GAIN[3:0]				
S12AD0	ADCER	ADRFMT	—	ADIEW	ADIE2	DIAGM	DIAGLD	DIAGVAL[1:0]	
		—	—	ACE	—	—	ADPRC[1:0]		SHBYP
S12AD0	ADSTRGR	—	—	—	ADSTRS1[4:0]				
		—	—	—	ADSTRS0[4:0]				
S12AD	ADCMPMD0	—	—	CEN102[1:0]		CEN101[1:0]		CEN100[1:0]	
		—	—	CEN002[1:0]		CEN001[1:0]		CEN000[1:0]	
S12AD	ADCMPMD1	—	VSELL1	VSELH1	CSEL1	—	VSELL0	VSELH0	CSEL0
		—	REFH[2:0]			—	REFL[2:0]		
S12AD	ADCMPNR0	—	—	—	—	C002NR[3:0]			
		C001NR[3:0]			C000NR[3:0]				
S12AD	ADCMPNR1	—	—	—	—	C102NR[3:0]			
		C101NR[3:0]			C100NR[3:0]				
S12AD	ADCMPFR	—	—	C102FLAG	C101FLAG	C100FLAG	C002FLAG	C001FLAG	C000FLAG
S12AD	ADCMPSEL	—	—	—	—	—	—	POERQ	IE
		—	—	SEL102	SEL101	SEL100	SEL002	SEL001	SEL000
S12AD0	ADDR (注2)	DIAGST[1:0]		—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD0	ADDR0A (注2)	—	—	—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD0	ADDR1 (注2)	—	—	—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD0	ADDR2 (注2)	—	—	—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD0	ADDR3 (注2)	—	—	—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD0	ADDR0B (注2)	—	—	—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD0	ADSSTR								
S12AD1	ADCSR	ADST	ADCS[1:0]		ADIE	CKS[1:0]		TRGE	EXTRG
S12AD1	ADANS	—	—	CH[1:0]		—	PG102SEL	PG101SEL	PG100SEL
		—	—	—	—	—	PG102EN	PG101EN	PG100EN
S12AD1	ADPG	—	—	—	—	PG102GAIN[3:0]			
		PG101GAIN[3:0]			PG100GAIN[3:0]				
S12AD1	ADCER	ADRFMT	—	ADIEW	ADIE2	DIAGM	DIAGLD	DIAGVAL[1:0]	
		—	—	ACE	—	—	ADPRC[1:0]		SHBYP
S12AD1	ADSTRGR	—	—	—	ADSTRS1[4:0]				
		—	—	—	ADSTRS0[4:0]				

表 5.2 I/O 寄存器位一览表 (14 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
S12AD1	ADRD (注2)	DIAGST[1:0]		—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD1	ADDR0A (注2)	—	—	—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD1	ADDR1 (注2)	—	—	—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD1	ADDR2 (注2)	—	—	—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD1	ADDR3 (注2)	—	—	—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD1	ADDR0B (注2)	—	—	—	—	AD11	AD10	AD9	AD8
		AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
S12AD1	ADSSTR								
PORT1	DDR	—	—	—	—	—	—	B1	B0
PORT2	DDR	—	—	—	B4	B3	B2	B1	B0
PORT3	DDR	—	—	—	—	B3	B2	B1	B0
PORT7	DDR	—	B6	B5	B4	B3	B2	B1	B0
PORT8	DDR	—	—	—	—	—	B2	B1	B0
PORT9	DDR	—	B6	B5	B4	B3	B2	B1	B0
PORTA	DDR	—	—	B5	B4	B3	B2	B1	B0
PORTB	DDR	B7	B6	B5	B4	B3	B2	B1	B0
PORTD	DDR	B7	B6	B5	B4	B3	B2	B1	B0
PORTE	DDR	—	—	B5	B4	B3	—	B1	B0
PORTG	DDR	—	—	B5	B4	B3	B2	B1	B0
PORT1	DR	—	—	—	—	—	—	B1	B0
PORT2	DR	—	—	—	B4	B3	B2	B1	B0
PORT3	DR	—	—	—	—	B3	B2	B1	B0
PORT7	DR	—	B6	B5	B4	B3	B2	B1	B0
PORT8	DR	—	—	—	—	—	B2	B1	B0
PORT9	DR	—	B6	B5	B4	B3	B2	B1	B0
PORTA	DR	—	—	B5	B4	B3	B2	B1	B0
PORTB	DR	B7	B6	B5	B4	B3	B2	B1	B0
PORTD	DR	B7	B6	B5	B4	B3	B2	B1	B0
PORTE	DR	—	—	B5	B4	B3	—	B1	B0
PORTG	DR	—	—	B5	B4	B3	B2	B1	B0
PORT1	PORT	—	—	—	—	—	—	B1	B0
PORT2	PORT	—	—	—	B4	B3	B2	B1	B0
PORT3	PORT	—	—	—	—	B3	B2	B1	B0
PORT4	PORT	B7	B6	B5	B4	B3	B2	B1	B0
PORT5	PORT	—	—	B5	B4	B3	B2	B1	B0
PORT6	PORT	—	—	B5	B4	B3	B2	B1	B0
PORT7	PORT	—	B6	B5	B4	B3	B2	B1	B0
PORT8	PORT	—	—	—	—	—	B2	B1	B0
PORT9	PORT	—	B6	B5	B4	B3	B2	B1	B0
PORTA	PORT	—	—	B5	B4	B3	B2	B1	B0
PORTB	PORT	B7	B6	B5	B4	B3	B2	B1	B0
PORTD	PORT	B7	B6	B5	B4	B3	B2	B1	B0
PORTE	PORT	—	—	B5	B4	B3	B2	B1	B0
PORTG	PORT	—	—	B5	B4	B3	B2	B1	B0
PORT1	ICR	—	—	—	—	—	—	B1	B0

表 5.2 I/O 寄存器位一览表 (15 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0		
PORT2	ICR	—	—	—	B4	B3	B2	B1	B0		
PORT3	ICR	—	—	—	—	B3	B2	B1	B0		
PORT4	ICR	B7	B6	B5	B4	B3	B2	B1	B0		
PORT5	ICR	—	—	B5	B4	B3	B2	B1	B0		
PORT6	ICR	—	—	B5	B4	B3	B2	B1	B0		
PORT7	ICR	—	B6	B5	B4	B3	B2	B1	B0		
PORT8	ICR	—	—	—	—	—	B2	B1	B0		
PORT9	ICR	—	B6	B5	B4	B3	B2	B1	B0		
PORTA	ICR	—	—	B5	B4	B3	B2	B1	B0		
PORTB	ICR	B7	B6	B5	B4	B3	B2	B1	B0		
PORTD	ICR	B7	B6	B5	B4	B3	B2	B1	B0		
PORTE	ICR	—	—	B5	B4	B3	—	B1	B0		
PORTG	ICR	—	—	B5	B4	B3	B2	B1	B0		
IOPORT	PF8IRQ	—	—	—	—	ITS1[1:0]		ITS0[1:0]			
IOPORT	PF9IRQ	—	—	—	—	—	ITS2	—	—		
IOPORT	PFAADC	—	—	—	—	—	—	ADTRG1S	ADTRG0S		
IOPORT	PFCMTU	TCLKS[1:0]		—	—	—	—	MTUS1	MTUS0		
IOPORT	PFDGPT	—	—	—	—	—	—	—	GPTS		
IOPORT	PFFSCI	—	—	—	—	—	SCI2S	—	—		
IOPORT	PFGSPI	SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	—		
IOPORT	PFHSPI	—	—	—	—	—	—	RSPIS[1:0]			
IOPORT	PFJCAN	CANS[1:0]		—	—	—	—	—	CANE		
IOPORT	PFKLIN	—	—	—	—	—	—	—	LINE		
IOPORT	PFMPOE	—	—	—	POE11E	POE10E	POE8E	POE4E	POE0E		
IOPORT	PFNPOE	POE10S	—	—	—	—	—	—	—		
SYSTEM	DPSBYCR	DPSBY	IOKEEP	—	—	—	—	—	—		
SYSTEM	DPSWCR	—	—	WTSTS[5:0]						—	—
SYSTEM	DPSIER	DNMIE	—	—	DLVDE	—	—	DIRQ1E	DIRQ0E		
SYSTEM	DPSIFR	DNMIF	—	—	DLVDF	—	—	DIRQ1F	DIRQ0F		
SYSTEM	DPSIEGR	DNMIEG	—	—	—	—	—	DIRQ1EG	DIRQ0EG		
SYSTEM	RSTSR	DPSRSTF	—	—	—	—	LVD2F	LVD1F	PORF		
FLASH	FWEPROR	—	—	—	—	—	—	FLWE[1:0]			
SYSTEM	LVDKEYR	—	—	—	—	—	—	—	—		
SYSTEM	LVDCR	LVD2E	LVD2RI	—	—	LVD1E	LVD1RI	—	—		
SYSTEM	DPSBKR0	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR1	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR2	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR3	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR4	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR5	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR6	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR7	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR8	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR9	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR10	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR11	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR12	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR13	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR14	—	—	—	—	—	—	—	—		
SYSTEM	DPSBKR15	—	—	—	—	—	—	—	—		



表 5.2 I/O 寄存器位一览表 (16 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
SYSTEM	DPSBKR16								
SYSTEM	DPSBKR17								
SYSTEM	DPSBKR18								
SYSTEM	DPSBKR19								
SYSTEM	DPSBKR20								
SYSTEM	DPSBKR21								
SYSTEM	DPSBKR22								
SYSTEM	DPSBKR23								
SYSTEM	DPSBKR24								
SYSTEM	DPSBKR25								
SYSTEM	DPSBKR26								
SYSTEM	DPSBKR27								
SYSTEM	DPSBKR28								
SYSTEM	DPSBKR29								
SYSTEM	DPSBKR30								
SYSTEM	DPSBKR31								
POE	ICSR1	—	—	—	POE0F	—	—	—	PIE1
		—	—	—	—	—	—	POE0M[1:0]	
POE	OCSR1	OSF1	—	—	—	—	—	OCE1	OIE1
		—	—	—	—	—	—	—	—
POE	ICSR2	—	—	—	POE4F	—	—	—	PIE2
		—	—	—	—	—	—	POE4M[1:0]	
POE	OCSR2	OSF2	—	—	—	—	—	OCE2	OIE2
		—	—	—	—	—	—	—	—
POE	ICSR3	—	—	—	POE8F	—	—	POE8E	PIE3
		—	—	—	—	—	—	POE8M[1:0]	
POE	SPOER	—	—	—	GPT23HIZ	GPT01HIZ	MTUCH0HIZ	MTUCH67HI Z	MTUCH34HI Z
POE	POECR1	—	—	—	—	MTU0DZE	MTU0CZE	MTU0BZE	MTU0AZE
POE	POECR2	—	—	—	—	—	MTU3BDZE	MTU4ACZE	MTU4BDZE
		—	—	—	—	—	MTU6BDZE	MTU7ACZE	MTU7BDZE
POE	POECR3	—	—	—	—	—	—	GPT3ABZE	GPT2ABZE
		—	—	—	—	—	—	GPT1ABZE	GPT0ABZE
POE	POECR4	—	—	IC5ADDMT6 7 ZE	IC4ADDMT6 7ZE	IC3ADDMT6 7ZE	—	IC1ADDMT6 7ZE	CMADDMT6 7ZE
		—	—	IC5ADDMT3 4 ZE	IC4ADDMT3 4ZE	IC3ADDMT3 4ZE	IC2ADDMT3 4ZE	—	CMADDMT3 4ZE
POE	POECR5	—	—	—	—	—	—	—	—
		—	—	IC5ADDMT0 ZE	IC4ADDMT0 ZE	—	IC2ADDMT0 ZE	IC1ADDMT0 ZE	CMADDMT0 ZE
POE	POECR6	—	—	—	IC4ADDGPT 23ZE	IC3ADDGPT 23ZE	IC2ADDGPT 23ZE	IC1ADDGPT 23ZE	CMADDGPT 23ZE
		—	—	IC5ADDGPT 01ZE	—	IC3ADDGPT 01ZE	IC2ADDGPT 01ZE	IC1ADDGPT 01ZE	CMADDGPT 01ZE
POE	ICSR4	—	—	—	POE10F	—	—	POE10E	PIE4
		—	—	—	—	—	—	POE10M[1:0]	
POE	ALR1	—	—	—	—	—	—	—	—
		OLSEN	—	OLSG2B	OLSG2A	OLSG1B	OLSG1A	OLSG0B	OLSG0A
POE	ICSR5	—	—	—	POE11F	—	—	POE11E	PIE5
		—	—	—	—	—	—	POE11M[1:0]	

表 5.2 I/O 寄存器位一览表 (17 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	
CAN0 (注3)	MB.ID	IDE	RTR	—	SID[10:0]					
		SID[10:0]					EID[17:0]			
		EID[17:0]								
		EID[17:0]								
	MB.DLC	—	—	—	—	—	—	—	—	
						DLC[3:0]				
MB.DATA0 ~ 7										
	MB.TS	TSH[7:0]								
TSL[7:0]										
CAN0 (注3)	MKR0	—	—	—	SID[10:0]					
		SID[10:0]					EID[17:0]			
		EID[17:0]								
		EID[17:0]								
CAN0 (注3)	MKR1	—	—	—	SID[10:0]					
		SID[10:0]					EID[17:0]			
		EID[17:0]								
		EID[17:0]								
CAN0 (注3)	MKR2	—	—	—	SID[10:0]					
		SID[10:0]					EID[17:0]			
		EID[17:0]								
		EID[17:0]								
CAN0 (注3)	MKR3	—	—	—	SID[10:0]					
		SID[10:0]					EID[17:0]			
		EID[17:0]								
		EID[17:0]								
CAN0 (注3)	MKR4	—	—	—	SID[10:0]					
		SID[10:0]					EID[17:0]			
		EID[17:0]								
		EID[17:0]								
CAN0 (注3)	MKR5	—	—	—	SID[10:0]					
		SID[10:0]					EID[17:0]			
		EID[17:0]								
		EID[17:0]								
CAN0 (注3)	MKR6	—	—	—	SID[10:0]					
		SID[10:0]					EID[17:0]			
		EID[17:0]								
		EID[17:0]								
CAN0 (注3)	MKR7	—	—	—	SID[10:0]					
		SID[10:0]					EID[17:0]			
		EID[17:0]								
		EID[17:0]								
CAN0 (注3)	FIDCR0	IDE	RTR	—	SID[10:0]					
		SID[10:0]					EID[17:0]			
		EID[17:0]								
		EID[17:0]								
CAN0 (注3)	FIDCR1	IDE	RTR	—	SID[10:0]					
		SID[10:0]					EID[17:0]			
		EID[17:0]								
		EID[17:0]								

表 5.2 I/O 寄存器位一览表 (18 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
CAN0 (注3)	MKIVLR								
CAN0 (注3)	MIER	—	—	—	—	—	—	—	—
CAN0 (注3)	MCTL.TX	TRMREQ	RECREQ	—	ONESHOT	—	TRMABT	TRMACTIVE	SENTDATA
CAN0 (注3)	MCTL.RX	TRMREQ	RECREQ	—	ONESHOT	—	MSGLOST	INVALIDDATA	NEWDATA
CAN0 (注3)	CTLR	—	—	RBOC	BOM[1:0]		SLPM	CANM[1:0]	
CAN0 (注3)	STR	—	RECST	TRMST	BOST	EPST	SLPST	HLTST	RSTST
CAN0 (注3)	BCR	TSEG1[3:0]			—	—	BRP[9:0]		
CAN0 (注3)	RFPCR	RFEST	RFWST	RFFST	RFMLF	RFUST[2:0]			RFE
CAN0 (注3)	TFPCR	TFEST	TFFST	—	—	TFUST[2:0]			TFE
CAN0 (注3)	EIER	BLIE	OLIE	ORIE	BORIE	BOEIE	EPIE	EWIE	BEIE
CAN0 (注3)	EIFR	BLIF	OLIF	ORIF	BORIF	BOEIF	EPIF	EWIF	BEIF
CAN0 (注3)	RECR	—	—	—	—	—	—	—	—
CAN0 (注3)	TECR	—	—	—	—	—	—	—	—
CAN0 (注3)	ECSR	EDPM	ADEF	BE0F	BE1F	CEF	AEF	FEF	SEF
CAN0 (注3)	CSSR	—	—	—	—	—	—	—	—
CAN0 (注3)	MSSR	SEST	—	—	MBNST[4:0]				
CAN0 (注3)	MSMR	—	—	—	—	—	MBSM[1:0]		
CAN0 (注3)	TSR	—	—	—	—	—	—	—	—
CAN0 (注3)	AFSR	—	—	—	—	—	—	—	—
CAN0 (注3)	TCR	—	—	—	—	—	TSTM[1:0]		TSTE
LINO	LWBR	—	—	—	—	—	—	—	LWBR0
LINO	LBRP0								
LINO	LBRP1								
LINO	LSTC								LSTM
LINO	L0MD	—	—	—	—	LCKS[1:0]		—	—
LINO	L0BRK	—	—	BDT[1:0]		BLT[3:0]			
LINO	L0SPC	—	—	IBS[1:0]		—	IBSH[2:0]		
LINO	L0WUP	WUTL[3:0]				—	—	—	—
LINO	L0IE	—	—	—	—	—	ERRIE	FRCIE	FTCIE
LINO	L0EDE	—	—	—	—	FERE	FTERE	PBERE	BERE
LINO	L0C	—	—	—	—	—	—	OM1	OM0
LINO	L0TC	—	—	—	—	—	—	RTS	FTS
LINO	L0MST	—	—	—	—	—	—	OMM1	OMM0
LINO	L0ST	HTRC	D1RC	—	—	ERR	—	FRC	FTC
LINO	L0EST	—	—	CSER	—	FER	FTER	PBER	BER

表 5.2 I/O 寄存器位一览表 (19 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
LIN0	L0RFC	—	FSM	CSM	RFT	RFDL[3:0]			
LIN0	L0IDB	IDP			ID				
LIN0	L0CBR								
LIN0	L0DB1								
LIN0	L0DB2								
LIN0	L0DB3								
LIN0	L0DB4								
LIN0	L0DB5								
LIN0	L0DB6								
LIN0	L0DB7								
LIN0	L0DB8								
MTU3	TCR	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
MTU4	TCR	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
MTU3	TMDR1	—	—	BFB	BFA	MD[3:0]			
MTU4	TMDR1	—	—	BFB	BFA	MD[3:0]			
MTU3	TIORH	IOB[3:0]				IOA[3:0]			
MTU3	TIORL	IOD[3:0]				IOC[3:0]			
MTU4	TIORH	IOB[3:0]				IOA[3:0]			
MTU4	TIORL	IOD[3:0]				IOC[3:0]			
MTU3	TIER	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
MTU4	TIER	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
MTU	TOERA	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
MTU	TGCRA	—	BDC	N	P	FB	WF	VF	UF
MTU	TOCR1A	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
MTU	TOCR2A	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
MTU3	TCNT								
MTU4	TCNT								
MTU	TCDRA								
MTU	TDDRA								
MTU3	TGRA								
MTU3	TGRB								
MTU4	TGRA								
MTU4	TGRB								
MTU	TCNTSA								
MTU	TCBRA								
MTU3	TGRC								
MTU3	TGRD								

表 5.2 I/O 寄存器位一览表 (20 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
MTU4	TGRC								
MTU4	TGRD								
MTU3	TSR	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
MTU4	TSR	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
MTU	TITCR1A	T3AEN	T3ACOR[2:0]			T4VEN	T4VCOR[2:0]		
MTU	TITCNT1A	—	T3ACNT[2:0]			—	T4VCNT[2:0]		
MTU	TBTERA	—	—	—	—	—	BTE[1:0]		
MTU	TDERA	—	—	—	—	—	—	TDER	
MTU	TOLBRA	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
MTU3	TBTM	—	—	—	—	—	—	TTSB	T TSA
MTU4	TBTM	—	—	—	—	—	—	TTSB	T TSA
MTU	TITMRA	—	—	—	—	—	—	—	TITM
MTU	TITCR2A	—	—	—	—	—	TRG4COR[2:0]		
MTU	TITCNT2A	—	—	—	—	—	TRG4CNT[2:0]		
MTU4	TADCR	BF[1:0]		—	—	—	—	—	—
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
MTU4	TADCORA								
MTU4	TADCORB								
MTU4	TADCOBRA								
MTU4	TADCOBRB								
MTU	TWCRA	CCE	—	—	—	—	—	—	WRE
MTU	TMDR2A	—	—	—	—	—	—	—	DRS
MTU3	TGRE								
MTU4	TGRE								
MTU4	TGRF								
MTU	TSTRA	CST4	CST3	—	—	—	CST2	CST1	CST0
MTU	TSYRA	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
MTU	TCSYSTR	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH6	SCH7
MTU	TRWERA	—	—	—	—	—	—	—	RWE
MTU0	TCR	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
MTU0	TMDR1	—	BFE	BFB	BFA	MD[3:0]			
MTU0	TIORH	IOB[3:0]				IOA[3:0]			
MTU0	TIORL	IOD[3:0]				IOC[3:0]			
MTU0	TIER	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
MTU0	TSR	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
MTU0	TCNT								
MTU0	TGRA								
MTU0	TGRB								

表 5.2 I/O 寄存器位一览表 (21 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
MTU0	TGRC								
MTU0	TGRD								
MTU0	TGRE								
MTU0	TGRF								
MTU0	TIER2	TTGE2	—	—	—	—	—	TGIEF	TGIEE
MTU0	TSR2	—	—	—	—	—	—	TGFF	TGFE
MTU0	TBTM	—	—	—	—	—	TTSE	TTSB	T TSA
MTU1	TCR	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]		
MTU1	TMDR1	—	—	—	—	MD[3:0]			
MTU1	TIOR	IOB[3:0]				IOA[3:0]			
MTU1	TIER	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
MTU1	TSR	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
MTU1	TCNT								
MTU1	TGRA								
MTU1	TGRB								
MTU1	TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE
MTU2	TCR	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]		
MTU2	TMDR1	—	—	—	—	MD[3:0]			
MTU2	TIOR	IOB[3:0]				IOA[3:0]			
MTU2	TIER	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
MTU2	TSR	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
MTU2	TCNT								
MTU2	TGRA								
MTU2	TGRB								
MTU6	TCR	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
MTU7	TCR	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
MTU6	TMDR1	—	—	BFB	BFA	MD[3:0]			
MTU7	TMDR1	—	—	BFB	BFA	MD[3:0]			
MTU6	TIORH	IOB[3:0]				IOA[3:0]			
MTU6	TIORL	IOD[3:0]				IOC[3:0]			
MTU7	TIORH	IOB[3:0]				IOA[3:0]			
MTU7	TIORL	IOD[3:0]				IOC[3:0]			
MTU6	TIER	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
MTU7	TIER	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
MTU	TOERB	—	—	OE7D	OE7C	OE6D	OE7B	OE7A	OE6B
MTU	TOCR1B	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
MTU	TOCR2B	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
MTU6	TCNT								

表 5.2 I/O 寄存器位一览表 (22 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
MTU7	TCNT								
MTU	TCDRB								
MTU	TDDRBR								
MTU6	TGRA								
MTU6	TGRB								
MTU7	TGRA								
MTU7	TGRB								
MTU	TCNTSB								
MTU	TGBRB								
MTU6	TGRC								
MTU6	TGRD								
MTU7	TGRC								
MTU7	TGRD								
MTU6	TSR	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
MTU7	TSR	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
MTU	TITCR1B	T6AEN	T6ACOR[2:0]			T7VEN	T7VCOR[2:0]		
MTU	TITCNT1B	—	T6ACNT[2:0]			—	T7VCNT[2:0]		
MTU	TBTERB	—	—	—	—	—	BTE[1:0]		—
MTU	TDERB	—	—	—	—	—	—	—	TDER
MTU	TOLBRB	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
MTU6	TBTM	—	—	—	—	—	—	TTSB	TTSA
MTU7	TBTM	—	—	—	—	—	—	TTSB	TTSA
MTU	TITMRB	—	—	—	—	—	—	—	TITM
MTU	TITCR2B	—	—	—	—	—	TRG7COR[2:0]		
MTU	TITCNT2B	—	—	—	—	—	TRG7CNT[2:0]		
MTU7	TADCR	BF[1:0]		—	—	—	—	—	—
		UT7AE	DT7AE	UT7BE	DT7BE	ITA6AE	ITA7VE	ITB6AE	ITB7VE
MTU7	TADCORA								
MTU7	TADCORB								
MTU7	TADCOBRA								
MTU7	TADCOBRB								
MTU6	TSYCR	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
MTU	TWCRB	CCE	—	—	—	—	—	SCC	WRE

表 5.2 I/O 寄存器位一览表 (23 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
MTU	TMDR2B	—	—	—	—	—	—	—	DRS
MTU6	TGRE								
MTU7	TGRE								
MTU7	TGRF								
MTU	TSTRB	CST7	CST6	—	—	—	—	—	—
MTU	TSYRB	SYNC7	SYNC6	—	—	—	—	—	—
MTU	TRWERB	—	—	—	—	—	—	—	RWE
MTU5	TCNTU								
MTU5	TGRU								
MTU5	TCRU	—	—	—	—	—	—	TPSC[1:0]	
MTU5	TIORU	—	—	—	IOC[4:0]				
MTU5	TCNTV								
MTU5	TGRV								
MTU5	TCRV	—	—	—	—	—	—	TPSC[1:0]	
MTU5	TIORV	—	—	—	IOC[4:0]				
MTU5	TCNTW								
MTU5	TGRW								
MTU5	TCRW	—	—	—	—	—	—	TPSC[1:0]	
MTU5	TIORW	—	—	—	IOC[4:0]				
MTU5	TSR	—	—	—	—	—	CMFU5	CMFV5	CMFW5
MTU5	TIER	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
MTU5	TSTR	—	—	—	—	—	CSTU5	CSTV5	CSTW5
MTU5	TCNTCMPCLR	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
GPT	GTSTR	—	—	—	—	—	—	—	—
		—	—	—	—	CST3	CST2	CST1	CST0
GPT	GTHSCR	CPHW3[1:0]		CPHW2[1:0]		CPHW1[1:0]		CPHW0[1:0]	
		CSHW3[1:0]		CSHW2[1:0]		CSHW1[1:0]		CSHW0[1:0]	
GPT	GTHCCR	—	—	—	—	CCSW3	CCSW2	CCSW1	CCSW0
		CCHW3[1:0]		CCHW2[1:0]		CCHW1[1:0]		CCHW0[1:0]	
GPT	GTHSSR	CSHSL3[3:0]				CSHSL2[3:0]			
		CSHSL1[3:0]				CSHSL0[3:0]			
GPT	GTHPSR	CSHPL3[3:0]				CSHPL2[3:0]			
		CSHPL1[3:0]				CSHPL0[3:0]			
GPT	GTWP	—	—	—	—	—	—	—	—
		—	—	—	—	WP3	WP2	WP1	WP0
GPT	GTSYNC	—	—	SYNC3[1:0]		—	—	SYNC2[1:0]	
		—	—	SYNC1[1:0]		—	—	SYNC0[1:0]	
GPT	GTETINT	—	—	—	—	—	—	ETINF	ETIPF
		—	—	—	—	—	—	ETINEN	ETIPEN
GPT	GTBDR	BD33	BD32	BD31	BD30	BD23	BD22	BD21	BD20
		BD13	BD12	BD11	BD10	BD03	BD02	BD01	BD00



表 5.2 I/O 寄存器位一览表 (24 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
GPT	GTSWP	—	—	—	—	—	—	—	—
		—	—	—	—	SWP3	SWP2	SWP1	SWP0
GPT	LCCR	LPSC[1:0]		TPSC[1:0]		LCNTAT	LCTO[2:0]		
		—	LCINTO	LCINTD	LCINTC	—	LCNTS	LCNTCR	LCNTE
GPT	LCST	—	—	—	—	—	—	—	—
		—	—	—	—	—	LISO	LISD	LISC
GPT	LCNT								
GPT	LCNTA								
GPT	LCNT00								
GPT	LCNT01								
GPT	LCNT02								
GPT	LCNT03								
GPT	LCNT04								
GPT	LCNT05								
GPT	LCNT06								
GPT	LCNT07								
GPT	LCNT08								
GPT	LCNT09								
GPT	LCNT10								
GPT	LCNT11								
GPT	LCNT12								
GPT	LCNT13								
GPT	LCNT14								
GPT	LCNT15								
GPT	LCNTDU								
GPT	LCNTDL								
GPT0	GTIOR	OBHLD	OBDFLT	GTIOB[5:0]					
		OAHL D	OADFLT	GTIOA[5:0]					
GPT0	GTINTAD	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN	EINT	—	—	—
		GTINTPR[1:0]		GTINTF	GTINTE	GTINTD	GTINTC	GTINTB	GTINTA

表 5.2 I/O 寄存器位一览表 (25 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
GPT0	GTCR	—	—	CCLR[1:0]		—	—	TPCS[1:0]	
		—	—	—	—	—	MD[2:0]		
GPT0	GTBER	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]	
		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
GPT0	GTUDC	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	UDF	UD
GPT0	GTITC	—	ADTBL	—	ADTAL	—	IVTT[2:0]		
		IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
GPT0	GTST	TUCF	—	—	—	DTEF	ITCNT[2:0]		
		TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
GPT0	GTCNT								
GPT0	GTCCRA								
GPT0	GTCCRB								
GPT0	GTCCRC								
GPT0	GTCCRD								
GPT0	GTCCRE								
GPT0	GTCCRF								
GPT0	GTPR								
GPT0	GTPBR								
GPT0	GTPDBR								
GPT0	GTADTRA								
GPT0	GTADTBRA								
GPT0	GTADTDBRA								
GPT0	GTADTRB								
GPT0	GTADTBRB								
GPT0	GTADTDBRB								
GPT0	GTONCR	OBE	OAE	—	SWN	—	—	—	NFV
		NFS[3:0]				NVB	NVA	NEB	NEA
GPT0	GTDTCR	—	—	—	—	—	—	—	TDFER
		—	—	TDBDE	TDBUE	—	—	—	TDE
GPT0	GTDVU								
GPT0	GTDVD								

表 5.2 I/O 寄存器位一览表 (26 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
GPT0	GTDBU								
GPT0	GTDBD								
GPT0	GTSOS	—	—	—	—	—	—	—	—
GPT0	GTSOTR	—	—	—	—	—	—	—	—
GPT1	GTIOR	OBHLD	OBDFLT	GTIOB[5:0]					
GPT1	GTINTAD	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN	EINT	—	—	—
GPT1	GTCR	—	—	CCLR[1:0]		—	—	TPCS[1:0]	
GPT1	GTBER	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]	
GPT1	GTUDC	—	—	—	—	—	—	—	—
GPT1	GTITC	—	ADTBL	—	ADTAL	—	IVTT[2:0]		
GPT1	GTST	TUCF	—	—	—	DTEF	ITCNT[2:0]		
GPT1	GTCCRA	—	—	—	—	—	—	—	—
GPT1	GTCCRB	—	—	—	—	—	—	—	—
GPT1	GTCCRC	—	—	—	—	—	—	—	—
GPT1	GTCCRD	—	—	—	—	—	—	—	—
GPT1	GTCCRE	—	—	—	—	—	—	—	—
GPT1	GTCCRF	—	—	—	—	—	—	—	—
GPT1	GTPR	—	—	—	—	—	—	—	—
GPT1	GTPBR	—	—	—	—	—	—	—	—
GPT1	GTPDBR	—	—	—	—	—	—	—	—
GPT1	GTADTRA	—	—	—	—	—	—	—	—
GPT1	GTADTBRA	—	—	—	—	—	—	—	—
GPT1	GTADTDBRA	—	—	—	—	—	—	—	—
GPT1	GTADTRB	—	—	—	—	—	—	—	—

表 5.2 I/O 寄存器位一览表 (27 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
GPT1	GTADTBRB								
GPT1	GTADTDBRB								
GPT1	GTONCR	OBE	OAE	—	SWN	—	—	—	NFV
		NFS[3:0]				NVB	NVA	NEB	NEA
GPT1	GTDTCR	—	—	—	—	—	—	—	TDFER
		—	—	TDBDE	TDBUE	—	—	—	TDE
GPT1	GTDVU								
GPT1	GTDVD								
GPT1	GTDBU								
GPT1	GTDBD								
GPT1	GTSOS	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	SOS[1:0]	
GPT1	GTSOTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	SOTR
GPT2	GTIOR	OBHLD	OBDFLT	GTIOB[5:0]					
		OAHL	OADFLT	GTIOA[5:0]					
GPT2	GTINTAD	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN	EINT	—	—	—
		GTINTPR[1:0]		GTINTF	GTINTE	GTINTD	GTINTC	GTINTB	GTINTA
GPT2	GTCR	—	—	CCLR[1:0]		—	—	TPCS[1:0]	
		—	—	—	—	—	MD[2:0]		
GPT2	GTBER	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]	
		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
GPT2	GTUDC	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	UDF	UD
GPT2	GTITC	—	ADTBL	—	ADTAL	—	IVTT[2:0]		
		IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
GPT2	GTST	TUCF	—	—	—	DTEF	ITCNT[2:0]		
		TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
GPT2	GCNT								
GPT2	GTCCRA								
GPT2	GTCCRB								
GPT2	GTCCRC								
GPT2	GTCCRD								
GPT2	GTCCRE								
GPT2	GTCCRF								
GPT2	GTPR								

表 5.2 I/O 寄存器位一览表 (28 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	
GPT2	GTPBR									
GPT2	GTPDBR									
GPT2	GTADTRA									
GPT2	GTADTBRA									
GPT2	GTADTDBRA									
GPT2	GTADTRB									
GPT2	GTADTBRB									
GPT2	GTADTDBRB									
GPT2	GTONCR	OBE	OAE	—	SWN	—	—	—	NFV	
		NFS[3:0]				NVB	NVA	NEB	NEA	
GPT2	GTDTCR	—	—	—	—	—	—	—	TDFER	
		—	—	TDBDE	TDBUE	—	—	—	TDE	
GPT2	GTDVU									
GPT2	GTDVD									
GPT2	GTDBU									
GPT2	GTDBD									
GPT2	GTSOS	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	SOS[1:0]		
GPT2	GTSOTR	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	—	SOTR	
GPT3	GTIOR	OBHLD	OBDFLT	GTIOB[5:0]						
		OAHLD	OADFLT	GTIOA[5:0]						
GPT3	GTINTAD	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN	EINT	—	—	—	
		GTINTPR[1:0]		GTINTF	GTINTE	GTINTD	GTINTC	GTINTB	GTINTA	
GPT3	GTCR	—	—	CCLR[1:0]		—	—	TPCS[1:0]		
		—	—	—	—	—	MD[2:0]			
GPT3	GTBER	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		
		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]		
GPT3	GTUDC	—	—	—	—	—	—	—	—	
		—	—	—	—	—	—	UDF	UD	
GPT3	GTITC	—	ADTBL	—	ADTAL	—	IVTT[2:0]			
		IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA	
GPT3	GTST	TUCF	—	—	—	DTEF	ITCNT[2:0]			
		TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA	
GPT3	GTCNT									
GPT3	GTCCRA									

表 5.2 I/O 寄存器位一览表 (29 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
GPT3	GTCCRB								
GPT3	GTCCRC								
GPT3	GTCCRD								
GPT3	GTCCRE								
GPT3	GTCCRF								
GPT3	GTPR								
GPT3	GTPBR								
GPT3	GTPDBR								
GPT3	GTADTRA								
GPT3	GTADTBRA								
GPT3	GTADTDBRA								
GPT3	GTADTRB								
GPT3	GTADTBRB								
GPT3	GTADTDBRB								
GPT3	GTONCR	OBE	OAE	—	SWN	—	—	—	NFV
		NFS[3:0]				NVB	NVA	NEB	NEA
GPT3	GTDTCR	—	—	—	—	—	—	—	TDFER
		—	—	TDBDE	TDBUE	—	—	—	TDE
GPT3	GTDVU								
GPT3	GTDVD								
GPT3	GTDBU								
GPT3	GTDBD	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	SOS[1:0]	
GPT3	GTSOS	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	SOS[1:0]	
GPT3	GTSOTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	SOTR
FLASH	FMODR	—	—	—	FRDMD	—	—	—	—
FLASH	FASTAT	ROMAE	—	—	CMDLK	DFLAE	—	DFLRPE	DFLWPE
FLASH	FAEINT	ROMAEIE	—	—	CMDLKIE	DFLAEIE	—	DFLRPEIE	DFLWPEIE
FLASH	FRDYIE	—	—	—	—	—	—	—	FRDYIE
FLASH	DFLRE0	KEY[7:0]							
		DBRE07	DBRE06	DBRE05	DBRE04	DBRE03	DBRE02	DBRE01	DBRE00

表 5.2 I/O 寄存器位一览表 (30 / 30)

模块符号	寄存器符号	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0
FLASH	DFLRE1	KEY[7:0]							
		DBRE15	DBRE14	DBRE13	DBRE12	DBRE11	DBRE10	DBRE09	DBRE08
FLASH	DFLWE0	KEY[7:0]							
		DBWE07	DBWE06	DBWE05	DBWE04	DBWE03	DBWE02	DBWE01	DBWE00
FLASH	DFLWE1	KEY[7:0]							
		DBWE15	DBWE14	DBWE13	DBWE12	DBWE11	DBWE10	DBWE09	DBWE08
FLASH	FCURAME	KEY[7:0]							
		—	—	—	—	—	—	—	FCRME
FLASH	FSTATR0	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD
FLASH	FSTATR1	FCUERR	—	—	FLOCKST	—	—	—	—
FLASH	FENTRYR	FEKEY[7:0]							
		FENTRYD	—	—	—	—	—	—	FENTRY0
FLASH	FPROTR	FPKEY[7:0]							
		—	—	—	—	—	—	—	FPROTCN
FLASH	FRESETR	FRKEY[7:0]							
		—	—	—	—	—	—	—	FRESET
FLASH	FCMDR	CMDR[7:0]							
		PCMDR[7:0]							
FLASH	FCPSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ESUSPMD
FLASH	DFLBCCNT	—	—	—	—	—	BCADR[7:0]		
		BCADR[7:0]						—	—
FLASH	FPESTAT	—	—	—	—	—	—	—	—
		PEERRST[7:0]							
FLASH	DFLBCSTAT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	BCST
FLASH	PCKAR	—	—	—	—	—	—	—	—
		PCKA[7:0]							

注. 有关 I/O 端口的相关寄存器 (0008 C001h ~ 0008 C116h)，本章节记载了 112 引脚 LQFP 版的位结构。寄存器和位结构因封装而不同，有关其他封装产品的寄存器和位结构，请参照“15. I/O 端口”。

注 1. 这是 ADDR.PDSEL 位为“0”、ADDPR.DPPRC 位为“0”（数据为 10 位精度，并且往 LSB 靠紧）时的位结构，详细内容请参照“29. 10 位 A/D 转换器 (ADA)”。

注 2. 这是 ADCER.ADRFMT 位为“0”（往右靠紧）并且 ADCER.ADPRC[1:0] 位为“00b”时的位结构，详细内容请参照“28. 12 位 A/D 转换器 (S12ADA)”。

注 3. 不支持无 CAN 功能版。

## 6. 复位

### 6.1 概要

复位有引脚复位、上电复位、电压监视复位、深度软件待机复位、独立看门狗定时器复位和看门狗定时器复位，复位名称和复位源如表 6.1 所示。

表 6.1 复位名称和复位源

复位名称	复位源
引脚复位	RES# 引脚的输入电压为 Low 电平。
上电复位	VCC 上升或者下降（监视电压：V <sub>POR</sub> ）。
电压监视复位	VCC 下降（监视电压：V <sub>det1</sub> 、V <sub>det2</sub> ）（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。
看门狗定时器复位	看门狗定时器发生上溢。

注 1. 有关监视电压（V<sub>det1</sub>、V<sub>det2</sub>、V<sub>POR</sub>），请参照“7. 电压检测电路（LVD）”和“33. 电特性”。

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



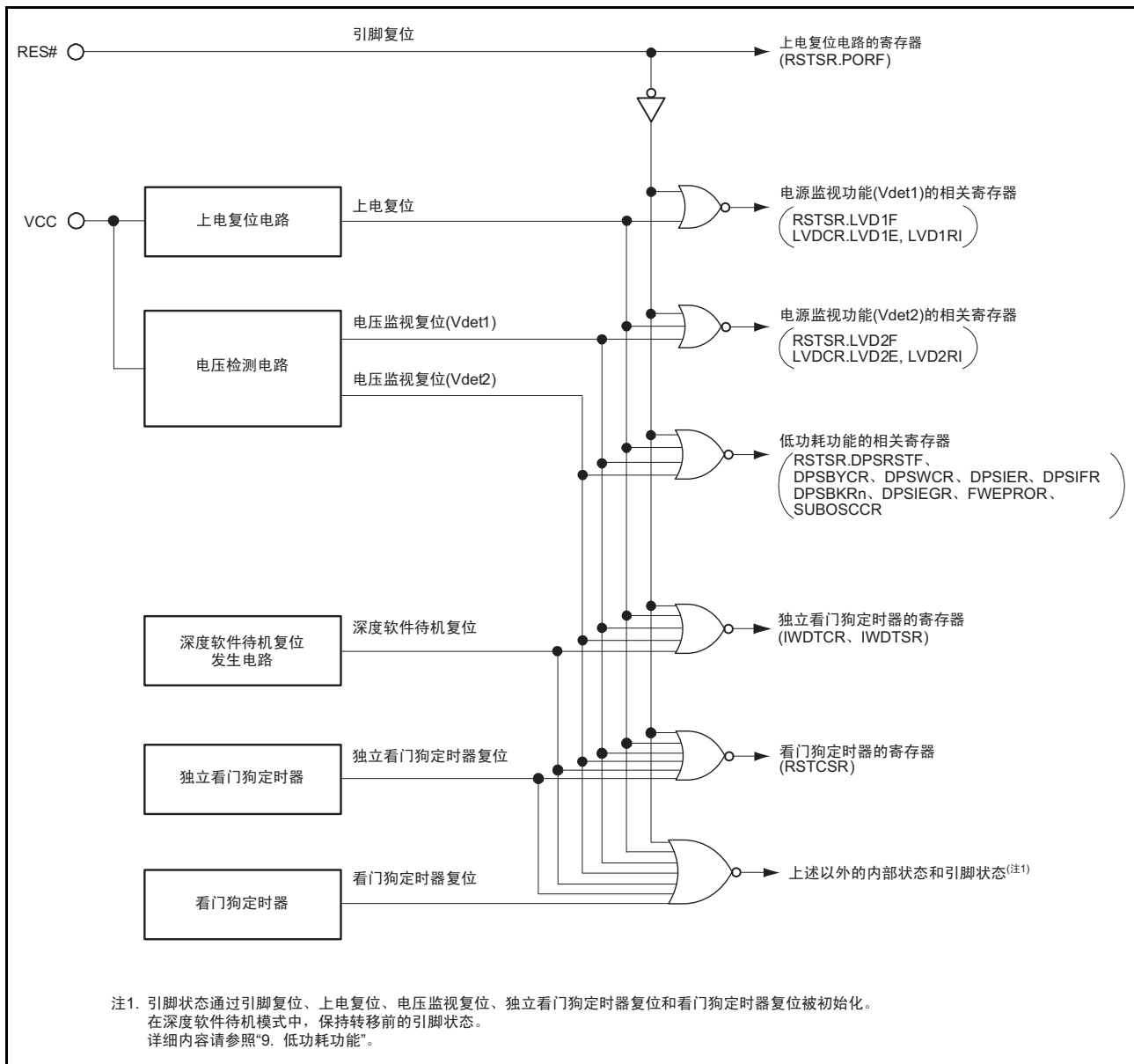


图 6.1 复位电路的框图

表 6.2 各种复位的初始化对象

复位对象	复位种类						
	引脚复位	上电复位	电压监视复位		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
			Vdet1	Vdet2			
上电复位电路的寄存器 RSTSR.PORF	○	—	—	—	—	—	—
电压监视功能 (Vdet1) 的 相关寄存器 RSTSR.LVD1F、 LVDCR.LVD1E、 LVDCR.LVD1RI	○	○	—	—	—	—	—
电压监视功能 (Vdet2) 的 相关寄存器 RSTSR.LVD2F、 LVDCR.LVD2E、 LVDCR.LVD2RI	○	○	○	—	—	—	—
低功耗功能的相关寄存器 RSTSR.DPSRSTF、 DPSBYCR、DPSWCR、 DPSIER、DPSIFR、 DPSBKRn、DPSIEGR、 FWEPROR、SUBOSCCR	○	○	○	○	—	—	—
独立看门狗定时器的寄存器 IWDTCR、IWDTSR	○	○	○	○	○	—	—
看门狗定时器的寄存器 RSTCSR	○	○	○	○	○	○	—
上述以外的寄存器和内部 状态	○	○	○	○	○	○	○
引脚状态	○	○	○	○	—	○	○

如果解除复位，就开始复位异常处理。有关复位异常处理，请参照“10. 异常处理”。  
复位的相关输入 / 输出引脚如表 6.3 所示。

表 6.3 复位的相关输入 / 输出引脚

引脚名	输入 / 输出	功能
RES#	输入	复位引脚

## 6.2 寄存器说明

复位的相关寄存器一览表如表 6.4 所示。  
各寄存器有表示各种复位发生源的位。

表 6.4 复位的相关寄存器一览表

寄存器名	符号	复位后的值 (注 1)	地址	存取长度
复位状态寄存器	RSTSR	x000 0xxxb	0008 C285h	8
复位控制 / 状态寄存器	RSTCSR	1Fh	0008 802Bh	8
IWDT 状态寄存器	IWDTSR	0000h	0008 8034h	16

注 1. 有效复位因各寄存器而不同，详细内容请参照“图 6.1 复位电路的框图”和“表 6.2 各种复位的初始化对象”。

### 6.2.1 复位状态寄存器 (RSTSR)

RSTSR 寄存器的详细内容请参照“9. 低功耗功能”。

### 6.2.2 复位控制 / 状态寄存器 (RSTCSR)

RSTCSR 寄存器的详细内容请参照“20. 看门狗定时器 (WDT)”。

### 6.2.3 IWDT 状态寄存器 (IWDTSR)

IWDTSR 寄存器的详细内容请参照“21. 独立看门狗定时器 (IWDT)”。

## 6.3 运行说明

### 6.3.1 引脚复位

这是 RES# 引脚引起的复位。

当 RES# 引脚为 Low 电平时，停止正在执行的全部处理，RX62T 群进入复位状态。

为了确保对 RX62T 群进行的复位，必须在接通电源时根据规定的振荡稳定时间，使 RES# 引脚保持 Low 电平，并且必须在运行中按照规定的复位脉宽，使 RES# 引脚保持 Low 电平。详细内容请参照“33. 电特性”。

### 6.3.2 上电复位

这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  $V_{POR}$ ，就在经过一定的时间（上电复位时间）后解除上电复位。上电复位时间是使外部电源和 LSI 稳定的时间。

如果在将 RES# 引脚置为 High 电平的状态下电源电压下降到低于等于  $V_{POR}$ ，就发生上电复位。如果此后 VCC 上升到高于  $V_{POR}$ ，就在经过上电复位时间后解除上电复位。

如果发生上电复位，RSTSR.PORF 标志就变为“1”。RSTSR.PORF 标志是只读寄存器，只在引脚复位时进行初始化。

上电复位的运行如图 6.2 所示。

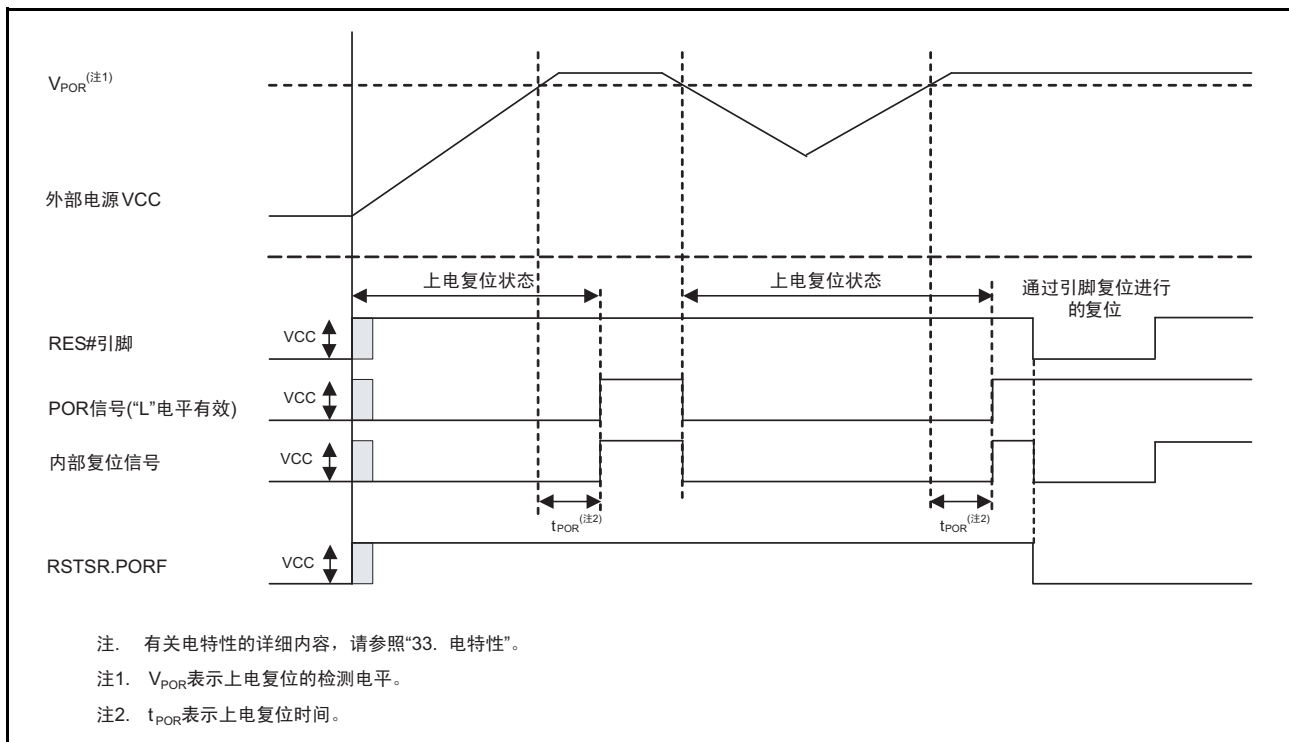


图 6.2 上电复位的运行

### 6.3.3 电压监视复位

这是电压检测电路引起的内部复位。

如果 VCC 低于等于 Vdet1，LVD1F 位就变为“1”。此时，如果 LVDCR 寄存器的 LVD1E 位为“1”（电压检测电路引起的复位或者中断有效）并且 LVD1RI 位为“0”（在检测到低电压时发生复位），电压检测电路就发生电压监视复位。

同样，如果 VCC 低于等于 Vdet2，LVD2F 位就变为“1”。此时，如果 LVDCR 寄存器的 LVD2E 位为“1”（电压检测电路引起的复位或者中断有效）并且 LVD2RI 位为“0”（在检测到低电压时发生复位），电压检测电路就产生电压监视复位。

如果此后 VCC 高于 Vdet1 或者 Vdet2，电压监视复位就在经过稳定时间（t<sub>POR</sub>）后被解除。

电压监视复位的详细内容请参照“7. 电压检测电路（LVD）”和“33. 电特性”。

### 6.3.4 深度软件待机复位

这是通过中断来解除深度软件待机模式时发生的内部复位。

如果解除深度软件待机模式，就在开始时钟振荡的同时产生深度软件待机复位。在经过深度软件待机等待时间设定位（DPSWCR.WTSTS[5:0]）所选的时间后，解除深度软件待机复位。

深度软件待机复位的详细内容请参照“9. 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。

在独立看门狗定时器发生下溢时发生独立看门狗定时器复位。独立看门狗定时器复位的详细内容请参照“21. 独立看门狗定时器（IWDG）”。

### 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。

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

看门狗定时器复位的详细内容请参照“20. 看门狗定时器（WDT）”。

## 6.4 复位发生源的判断

能通过读 RSTCSR 寄存器、IWDTSR 寄存器、RSTSR 寄存器和 LVDCR 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 6.3 所示。

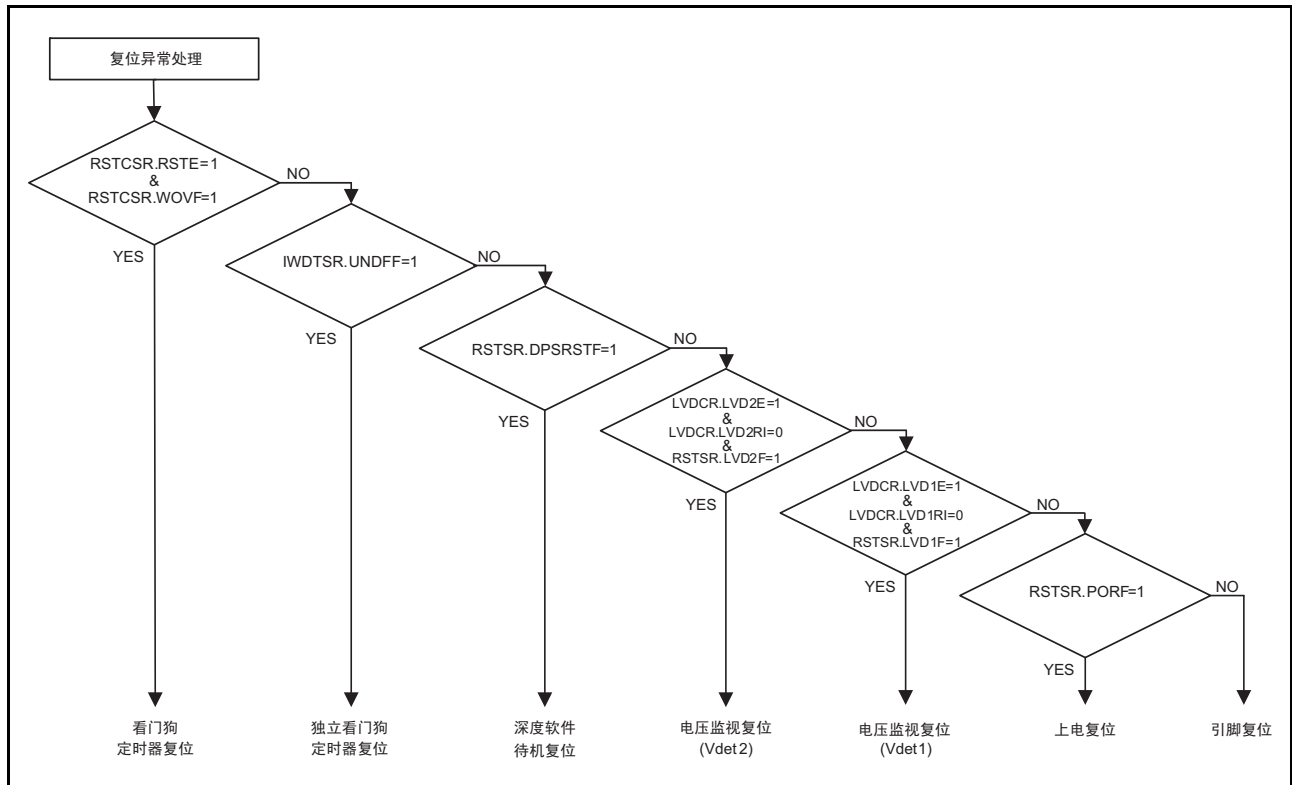


图 6.3 复位发生源的判断流程例子

## 6.5 使用时的注意事项

### 6.5.1 设计电路板时的注意事项

RX62T 群的 XTAL 引脚邻接复位引脚。因此，必须通过 GND 保护复位信号，以免受到时钟信号的影响。

## 7. 电压检测电路 (LVD)

### 7.1 概要

电压检测电路是监视 VCC 电压电平的电路。如果 VCC 下降到低于电压检测电平，电压检测电路就能对 LSI 内部进行复位，也能发生中断。

表 7.1 电压检测电路的规格

项目	内容
电压检测电路 1 (LVD1)	<ul style="list-style-type: none"> <li>能检测到电源电压 VCC 低于等于 Vdet1 (注 1)。</li> <li>能在检测到低电压时发生内部复位或者内部中断。</li> </ul>
电压检测电路 2 (LVD2)	<ul style="list-style-type: none"> <li>能检测到电源电压 VCC 低于等于 Vdet2 (注 1)。</li> <li>能在检测到低电压时发生内部复位或者内部中断。</li> </ul>

注 1. 有关 Vdet1 和 Vdet2，请参照“33. 电特性”。

电压检测电路的框图如图 7.1 所示。

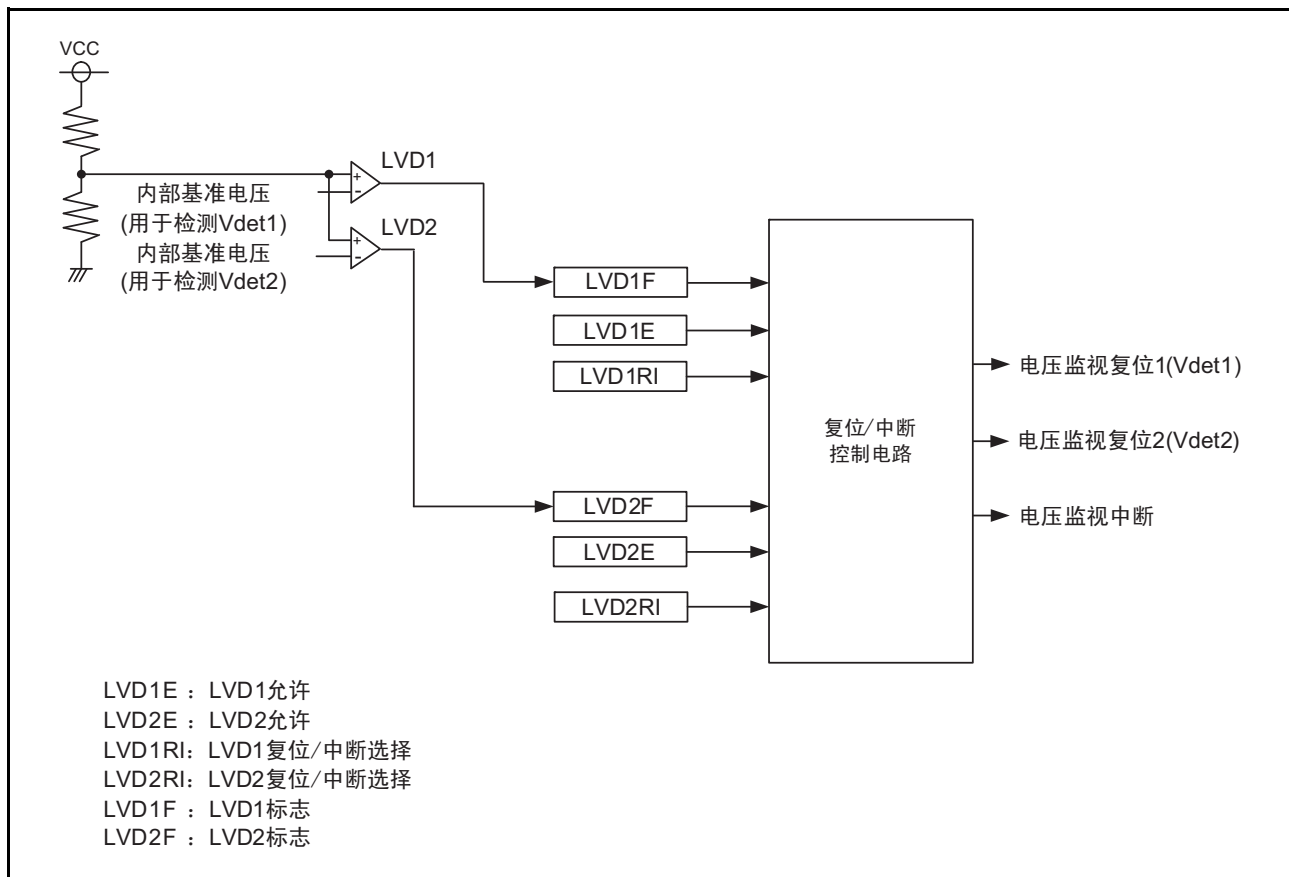


图 7.1 电压检测电路的框图

## 7.2 寄存器说明

电压检测电路的寄存器一览表如表 7.2 所示。

表 7.2 电压检测电路的寄存器一览表

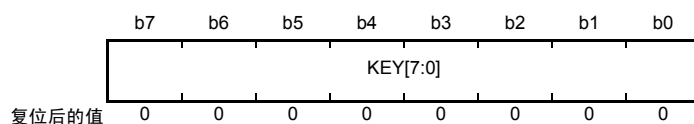
寄存器名	符号	复位后的值	地址	存取长度
复位状态寄存器	RSTSR	x000 0xxx <b>b</b>	0008 C285h	8
低电压检测控制寄存器的键码寄存器	LVDKEYR	00h	0008 C28Ch	8
低电压检测控制寄存器	LVDCCR	000x 000 <b>xb</b>	0008 C28Dh	8

### 7.2.1 复位状态寄存器 (RSTSR)

RSTSR 寄存器的详细内容请参照“9. 低功耗功能”。

### 7.2.2 低电压检测控制寄存器的键码寄存器 (LVDKEYR)

地址 0008 C28Ch



位	符号	位名	功能	R/W
b7-b0	KEY[7:0]	LVDCCR 键码	3Ch: LVDCCR 寄存器的写操作有效 上述以外: LVDCCR 寄存器的写操作无效	R/W

LVDKEYR 寄存器控制允许或者禁止写 LVDCCR 寄存器。  
能通过给 LVDKEYR 寄存器写键码，写 LVDCCR 寄存器。

#### KEY[7:0] 位 (LVDCCR 键码)

如果给 LVDKEYR 寄存器写“3Ch”，LVDCCR 寄存器的写操作就有效；如果写“3Ch”以外的值，LVDCCR 寄存器的写操作就无效。在写 LVDCCR 寄存器后，KEY[7:0] 位变为“00h”。



### 7.2.3 低电压检测控制寄存器 (LVDCR)

地址 0008 C28Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2E	LVD2RI	—	—	LVD1E	LVD1RI	—	—
复位后的值	0	0	0	x	0	0	0	x

x: 不定值

位	符号	位名	功能	R/W
b0	—	保留位	读取值为不定值，写操作无效。	R/W
b1	—	保留位	读写值都为“0”。	R/W
b2	LVD1RI	LVD1 复位 / 中断选择位	0: 当检测到低电压时，发生复位。 1: 当检测到低电压时，发生中断。	R/W
b3	LVD1E	LVD1 有效位	0: LVD1 无效 1: LVD1 有效	R/W
b4	—	保留位	读取值为不定值，写操作无效。	R/W
b5	—	保留位	读写值都为“0”。	R/W
b6	LVD2RI	LVD2 复位 / 中断选择位	0: 当检测到低电压时，发生复位。 1: 当检测到低电压时，发生中断。	R/W
b7	LVD2E	LVD2 有效位	0: LVD2 无效 1: LVD2 有效	R/W

LVDCR 寄存器是控制电压检测电路的寄存器。

通过引脚复位或者上电复位对 LVD1E 位和 LVD1RI 位进行初始化。

通过引脚复位、上电复位或者电压监视复位 1 (Vdet1) 对 LVD2E 位和 LVD2RI 位进行初始化。

在写 LVDCR 寄存器前，必须事先在 LVDKEYR.KEY[7:0] 写键码。

LVDCR 寄存器的设定和电压检测电路的状态如表 7.3 所示。

#### LVD1RI 位 (LVD1 复位 / 中断选择位)

当通过电压检测电路 1 检测到低电压时，此位选择是发生内部复位还是发生中断。

#### LVD1E 位 (LVD1 有效位)

此位通过电压检测电路 1 将复位或者中断置为有效。

#### LVD2RI 位 (LVD2 复位 / 中断选择位)

当通过电压检测电路 2 检测到低电压时，此位选择是发生内部复位还是发生中断。

#### LVD2E 位 (LVD2 有效位)

此位通过电压检测电路 2 将复位或者中断置为有效。

表 7.3 LVDCR 寄存器的设定和电压检测电路的状态

LVDCR 寄存器				电源检测电路 2 (LVD2)	电源检测电路 1 (LVD1)
LVD2E 位	LVD2RI 位	LVD1E 位	LVD1RI 位		
0	0	0	0	LVD2 无效	LVD1 无效
0	0	1	0	LVD2 无效	LVD1 有效 (复位)
0	0	1	1	LVD2 无效	LVD1 有效 (中断)
1	0	0	0	LVD2 有效 (复位)	LVD1 无效
1	1	0	0	LVD2 有效 (中断)	LVD1 无效
1	1	1	0	LVD2 有效 (中断)	LVD1 有效 (复位)
不能进行上述以外的设定。					

## 7.3 电压检测电路

### 7.3.1 电压监视复位

电压检测电路引起的电压监视复位如图 7.2 和图 7.3 所示。

如果 VCC 低于等于  $V_{det2}$ ，RSTSR.LVD2F 标志就变为“1”。此时，如果 LVDCR.LVD2E 位为“1”并且 LVDCR.LVD2RI 位为“0”，电压检测电路就发生电压监视复位。

同样，如果 VCC 低于等于  $V_{det1}$ ，RSTSR.LVD1F 标志就变为“1”。此时，如果 LVDCR.LVD1E 位为“1”并且 LVDCR.LVD1RI 位为“0”，电压检测电路就发生电压监视复位。

如果此后 VCC 高于  $V_{det1}$  或者  $V_{det2}$ ，电压监视复位就被解除。电压监视复位一旦被解除，就开始复位异常处理。

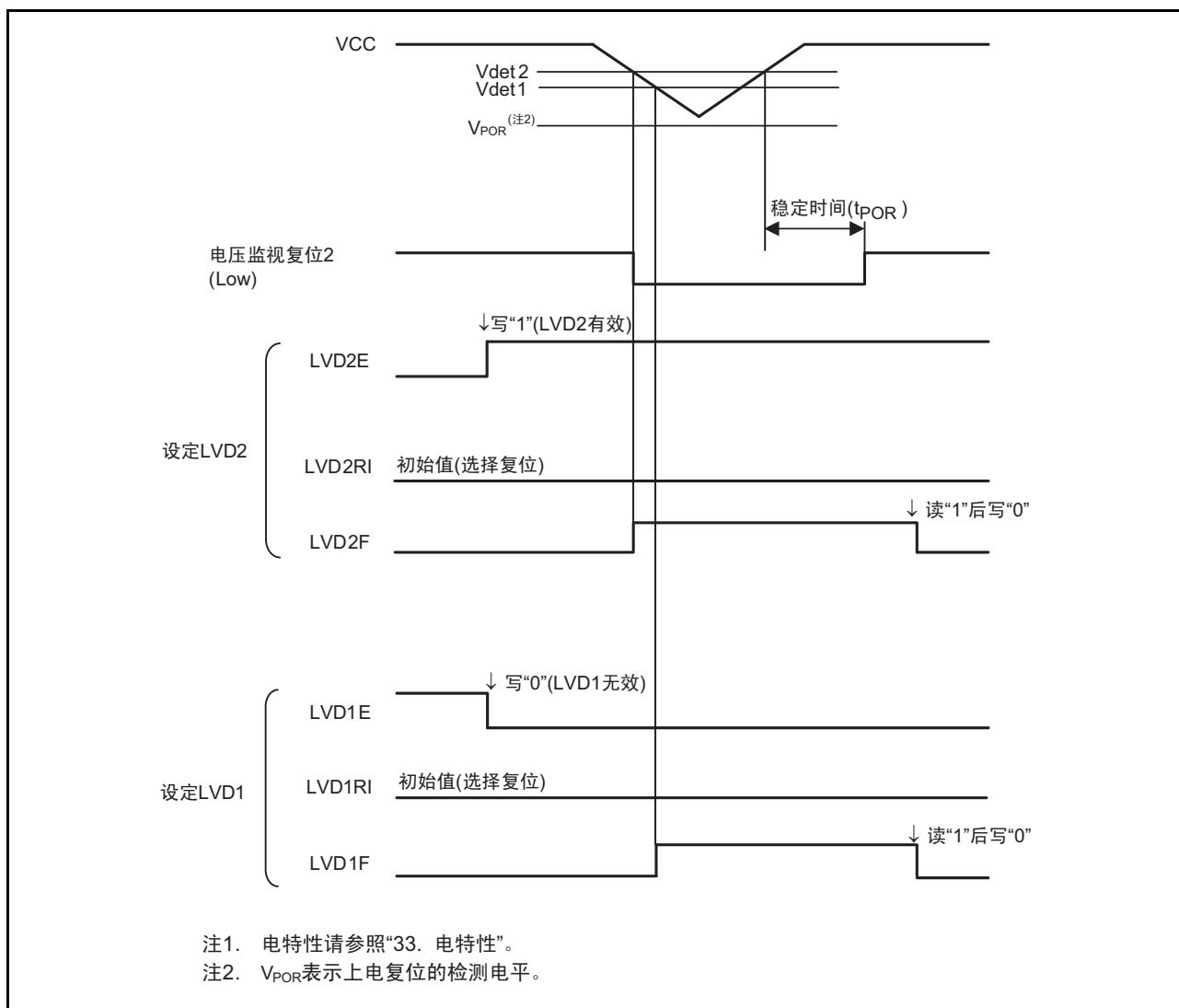


图 7.2 电压监视复位的时序 (1) (选择 LVD2 复位、LVD1 无效)

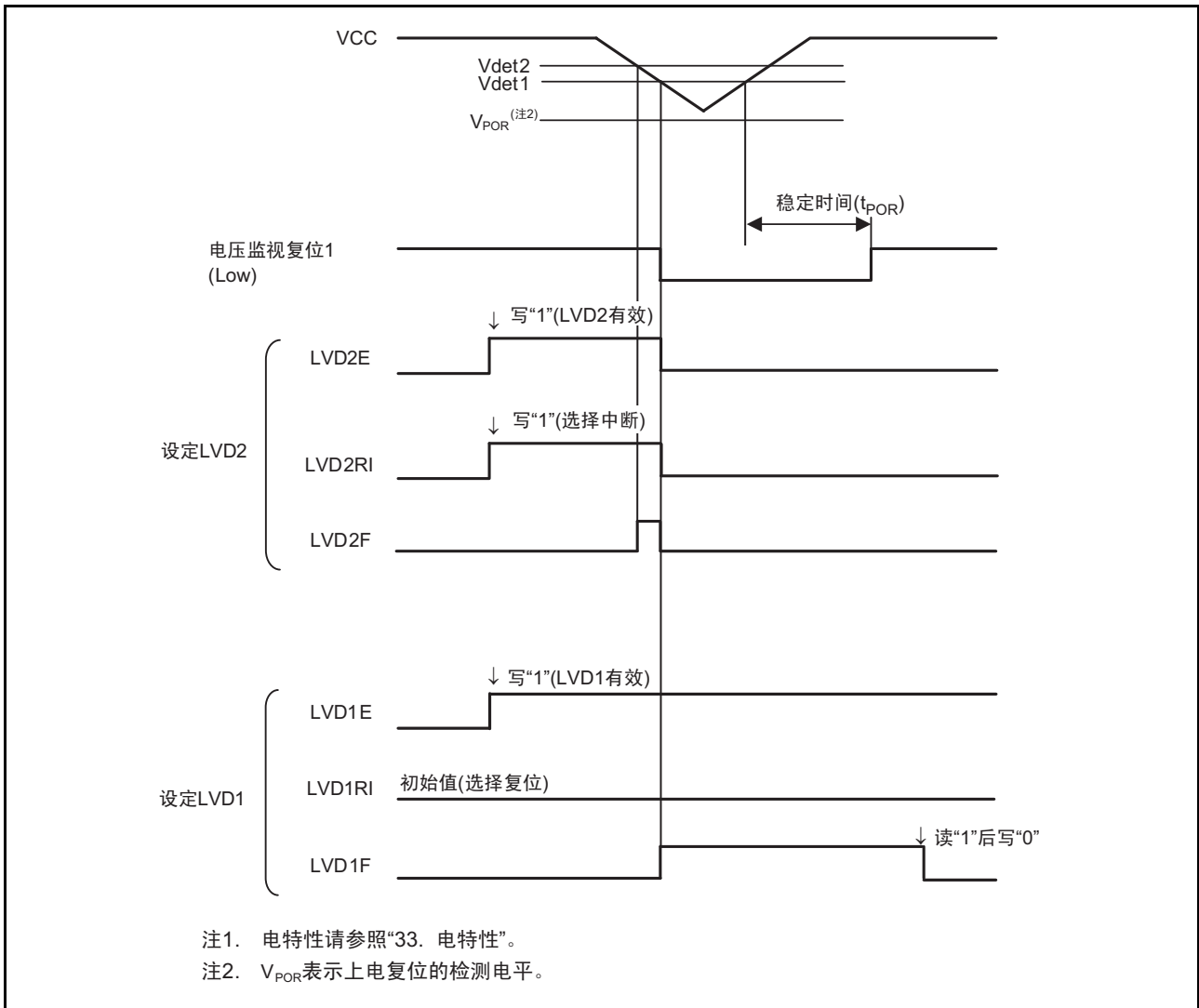


图 7.3 电压监视复位的时序 (2) (选择 LVD2 中断、选择 LVD1 复位)

### 7.3.2 电压监视中断

电压检测电路引起的电压监视中断如图 7.4 所示。

如果 VCC 低于等于  $V_{det1}$ , RSTSR.LVD1F 标志就变为“1”。此时, 如果 LVDCR.LVD1E 位和 LVDCR.LVD1RI 位都为“1”, 电压检测电路就请求电压监视中断。

同样, 如果 VCC 低于等于  $V_{det2}$ , RSTSR.LVD2F 标志就变为“1”。此时, 如果 LVDCR.LVD2E 位和 LVDCR.LVD2RI 位都为“1”, 电压检测电路就请求电压监视中断。

能通过在读“1”后给 RSTSR.LVD1F 标志写“0”, 将此标志置“0”。此时, 在 VCC 电压电平低于等于 LVD1 的检测电平 ( $V_{det1}$ ) 的状态下, 不能将 RSTSR.LVD1F 标志置“0”。能通过在给 RSTSR.LVD1F 标志写“0”后从 RSTSR.LVD1F 标志读“0”, 确认 VCC 高于等于  $V_{det1}$ 。

同样, 能通过在读“1”后给 RSTSR.LVD2F 标志写“0”, 将此标志置“0”。此时, 在 VCC 电压电平低于等于 LVD2 的检测电平 ( $V_{det2}$ ) 的状态下, 不能将 RSTSR.LVD2F 标志置“0”。能通过在给 RSTSR.LVD2F 标志写“0”后从 RSTSR.LVD2F 标志读“0”, 确认 VCC 高于等于  $V_{det2}$ 。

电压监视中断被分配给非屏蔽中断。有关中断处理的详细内容, 请参照“11. 中断控制器 (ICU)”。

电压监视中断的设定步骤例子如图 7.5 所示。

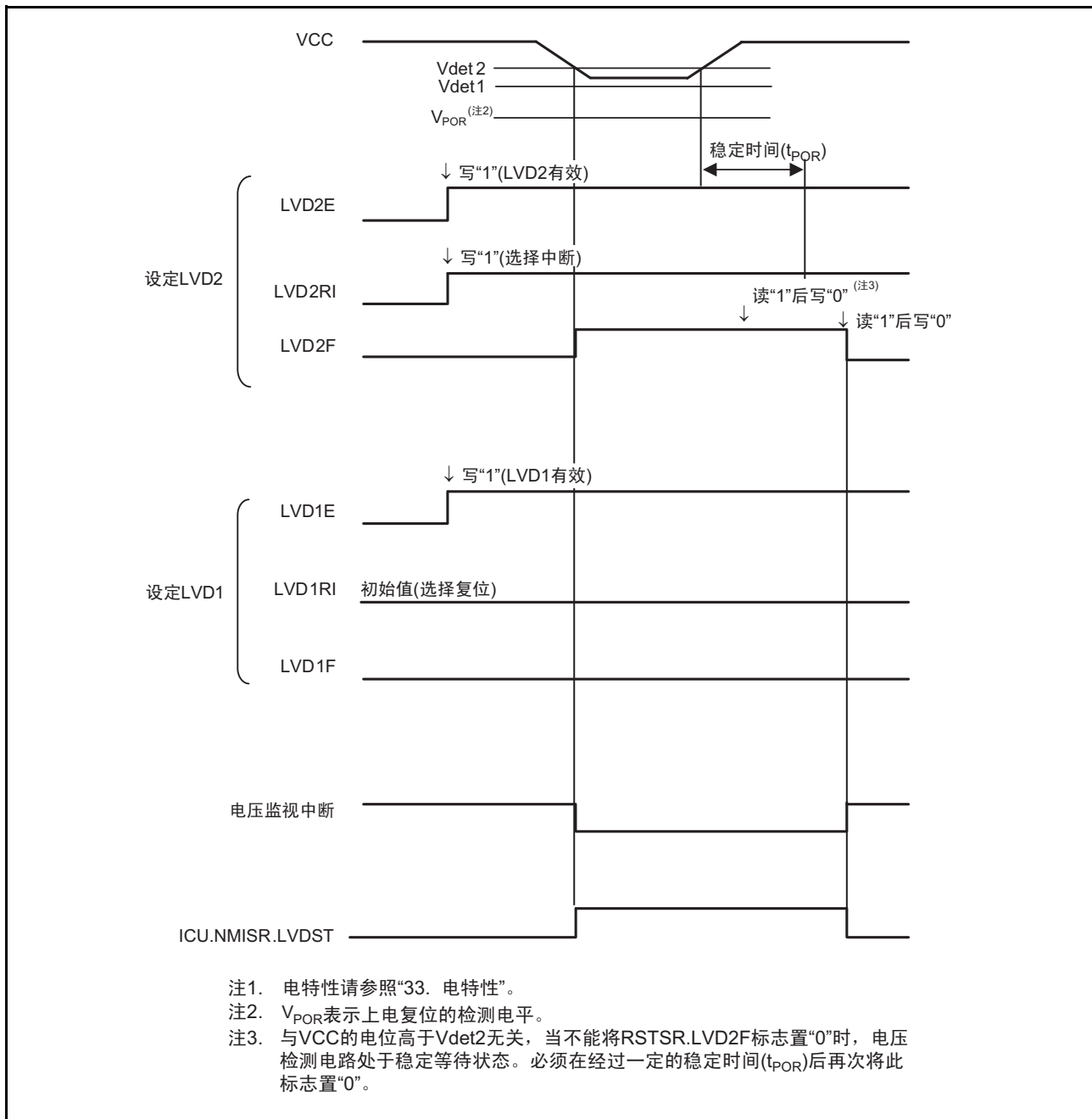


图 7.4 电压监视中断的时序（选择 LVD2 中断、选择 LVD1 复位）

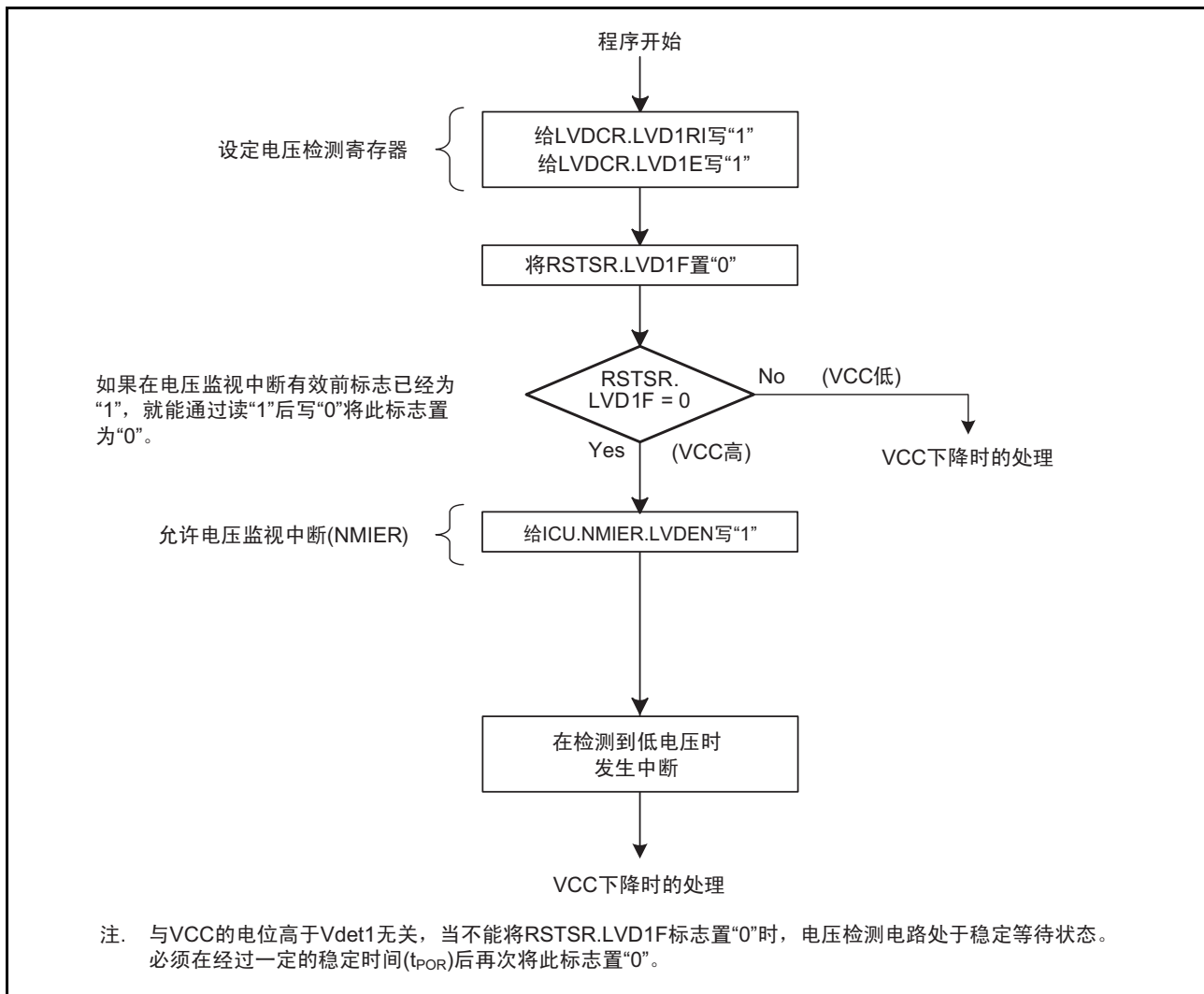


图 7.5 电压监视中断的设定步骤例子

### 7.3.3 通过电压检测电路进行深度软件待机模式的解除

如果 VCC 低于等于  $V_{det1}$ , RSTSR.LVD1F 标志就变为“1”。

此时, 如果 LVDCR.LVD1E 位、LVDCR.LVD1RI 位和 DPSIER.DLVDE 位都为“1”, DPSIFR.DLVDF 标志就变为“1”, 并通过电压监视中断来请求深度软件待机模式的解除。

同样, 如果 VCC 低于等于  $V_{det2}$ , RSTSR.LVD2F 标志就变为“1”。

此时, 如果 LVDCR.LVD2E 位、LVDCR.LVD2RI 位和 DPSIER.DLVDE 位都为“1”, DPSIFR.DLVDF 标志就变为“1”, 并通过电压监视中断来请求深度软件待机模式的解除。

有关深度软件待机模式, 请参照“9. 低功耗功能”。

## 8. 时钟发生电路

### 8.1 概要

RX62T 群内置了时钟发生电路，生成系统时钟（ICLK）、外围模块时钟（PCLK）和内部振荡器时钟（IWDTCLK）。

时钟发生电路由主时钟振荡器、IWDT 专用低速内部振荡器、振荡停止检测电路、内部振荡电路、PLL（Phase Locked Loop）电路、分频器和选择器构成。

时钟发生电路的规格和框图分别如表 8.1 和图 8.1 所示。

表 8.1 时钟发生电路的规格

项目	规格
用途	<ul style="list-style-type: none"> <li>生成给 CPU、DTC、MTU3、GPT、ROM 和 RAM 提供的系统时钟（ICLK）。</li> <li>生成给外围模块提供的外围模块时钟（PCLK）。</li> <li>生成给 IWDT 提供的内部振荡器时钟（IWDTCLK）。</li> </ul>
工作频率	ICLK: 8MHz ~ 100MHz PCLK: 8MHz ~ 50MHz IWDTCLK: 125kHz (Typ.) (注1) 时钟频率的设定限制: 维持 ICLK ≥ PCLK
能连接的谐振器或者附加电路	晶体谐振器
谐振器或者附加电路的连接引脚	主时钟: EXTAL、XTAL
输入时钟（EXTAL）的频率	8MHz ~ 12.5MHz
ICLK/PCLK 时钟的选择	<ul style="list-style-type: none"> <li>可给 ICLK/PCLK 分别选择 EXTAL×8、EXTAL×4、EXTAL×2、EXTAL×1。</li> </ul>
振荡停止检测功能	这是在检测到主时钟振荡器停止振荡时转换为内部振荡的功能。 这是将 MTU3 和 GPT 的引脚置为高阻抗的功能。

注 1. 详细内容请参照“33. 电特性”。

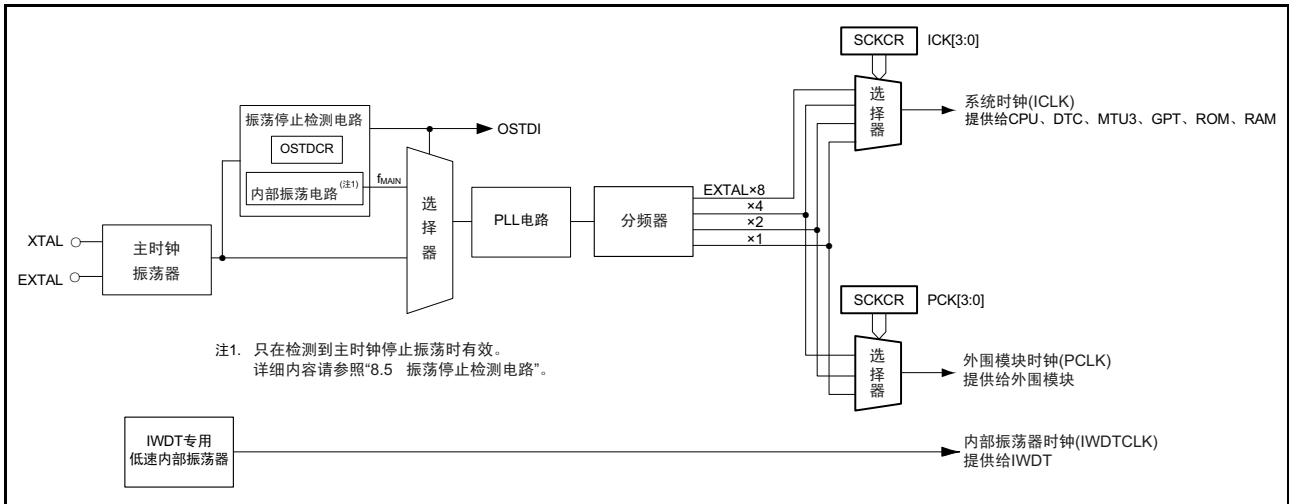


图 8.1 时钟发生电路的框图

时钟发生电路的输入 / 输出引脚如表 8.2 所示。

表 8.2 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	连接晶体谐振器的引脚
EXTAL	输入	EXTAL 引脚也可以输入外部时钟，详细内容请参照“8.3.2 输入外部时钟的方法”。

## 8.2 寄存器说明

时钟发生电路的寄存器一览表如表 8.3 所示。

表 8.3 时钟发生电路的寄存器一览表

寄存器名	符号	复位后的值	地址	存取长度
系统时钟控制寄存器	SCKCR	0202 0200h	0008 0020h	32
振荡停止检测控制寄存器	OSTDCR	0080h	0008 0040h	16

### 8.2.1 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	
复位后的值	—	—	—	—	ICK[3:0]			—	—	—	—	—	—	—	—	—	
	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
复位后的值	—	—	—	—	PCK[3:0]			—	—	—	—	—	—	—	—	—	
	0	0	0	0	0	0	1	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
b11-b8	PCK[3:0] (注 1)	外围模块时钟选择位	b11 b8 0001: ×4 0010: ×2 0011: ×1 上述以外: 不能设定	R/W
b16-b12	—	保留位	读写值都为“0”。	R/W
b17	—	保留位	读写值都为“1”。	R/W
b23-b18	—	保留位	读写值都为“0”。	R/W
b27-b24	ICK[3:0] (注 2)	系统时钟选择位	b27 b24 0000: ×8 0001: ×4 0010: ×2 0011: ×1 上述以外: 不能设定	R/W
b31-b28	—	保留位	读写值都为“0”。	R/W

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

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

SCKCR 寄存器选择系统时钟（ICLK）和外围模块时钟（PCLK）的频率。

#### PCK[3:0] 位（外围模块时钟选择位）

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

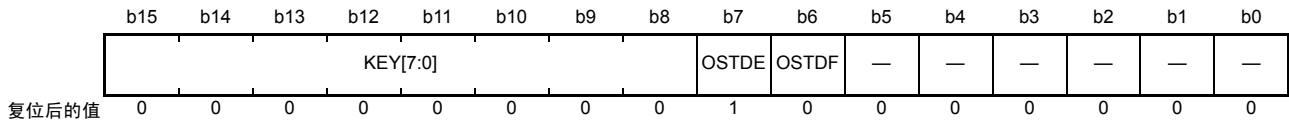
#### ICK[3:0] 位（系统时钟选择位）

这些位选择系统时钟（ICLK）的频率，并且表示输入时钟（EXTAL）的倍率。



## 8.2.2 振荡停止检测控制寄存器 (OSTDCR)

地址 0008 0040h



位	符号	位名	功能	R/W
b5-b0	—	保留位	读写值都为“0”。	R
b6	OSTDF	振荡停止检测标志	当 OSTDE 为“1”时： 0: 主时钟振荡器正常振荡 1: 检测到主时钟振荡器停止振荡  当 OSTDE 为“0”时： 读取值为“0”。	R
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
b15-b8	KEY[7:0]	OSTDCR 键码	ACh : 允许写 OSTDCR 寄存器 上述以外：禁止写 OSTDCR 寄存器 读取值为“0”。	R/W

OSTDCR 寄存器是控制振荡停止检测功能的寄存器。

### OSTDF 标志 (振荡停止检测标志)

此标志表示主时钟振荡器的状态。

当 OSTDF 标志为“1”时，表示检测到主时钟振荡器停止振荡。

### 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。

当 OSTDE 位为“1” (振荡停止检测功能有效) 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”后发行 WAIT 指令。

### KEY[7:0] 位 (OSTDCR 键码)

这些位选择允许或者禁止写 OSTDCR 寄存器。

必须在将 KEY[7:0] 位置“ACh”后给 OSTDE 位写值。当 KEY[7:0] 位的值不是“ACh”时，即使写 OSTDCR 寄存器，OSTDE 位的值也不变。

### 8.3 主时钟振荡器

提供时钟的方法有连接晶体谐振器和输入外部时钟两种。

#### 8.3.1 连接晶体谐振器的方法

连接晶体谐振器时的连接例子和阻尼电阻  $R_d$ （参考值）分别如图 8.2 和表 8.4 所示。

在连接晶体谐振器提供时钟的情况下，连接的晶体谐振器必须为 8 ~ 12.5MHz。

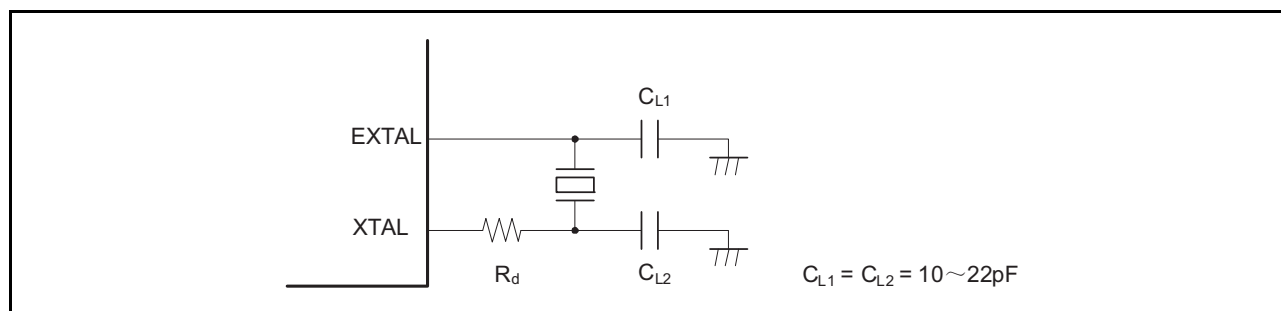


图 8.2 晶体谐振器的连接例子

表 8.4 阻尼电阻（参考值）

频率 (MHz)	8	10	12.5
$R_d$ ( $\Omega$ )	200	100	0

晶体谐振器的等效电路如图 8.3 所示，必须使用表 8.5 所示特性的晶体谐振器。

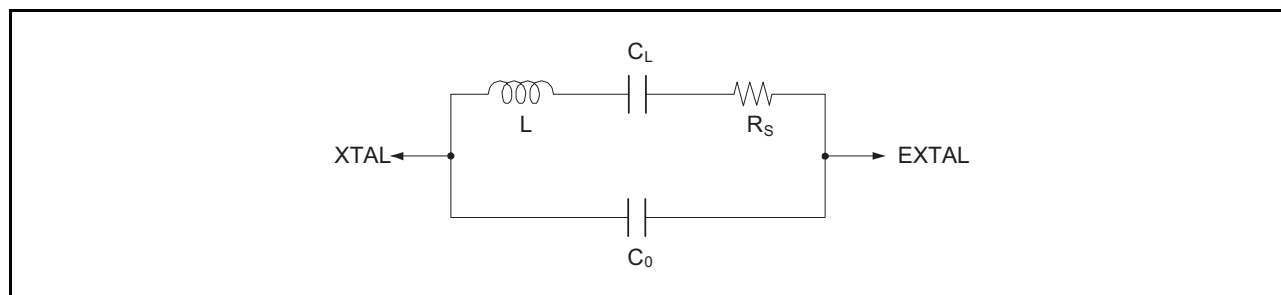


图 8.3 晶体谐振器的等效电路

表 8.5 晶体谐振器的特性（参考值）

频率 (MHz)	8	10	12.5
$R_S$ max ( $\Omega$ )	80	70	60
$C_0$ max (pF)	7		

### 8.3.2 输入外部时钟的方法

输入外部时钟的连接例子如图 8.4 所示。在将 XTAL 引脚置为开路时，寄生电容必须小于等于 10pF。在给 XTAL 引脚输入反相时钟时，必须在待机模式中将外部时钟置为 High 电平。

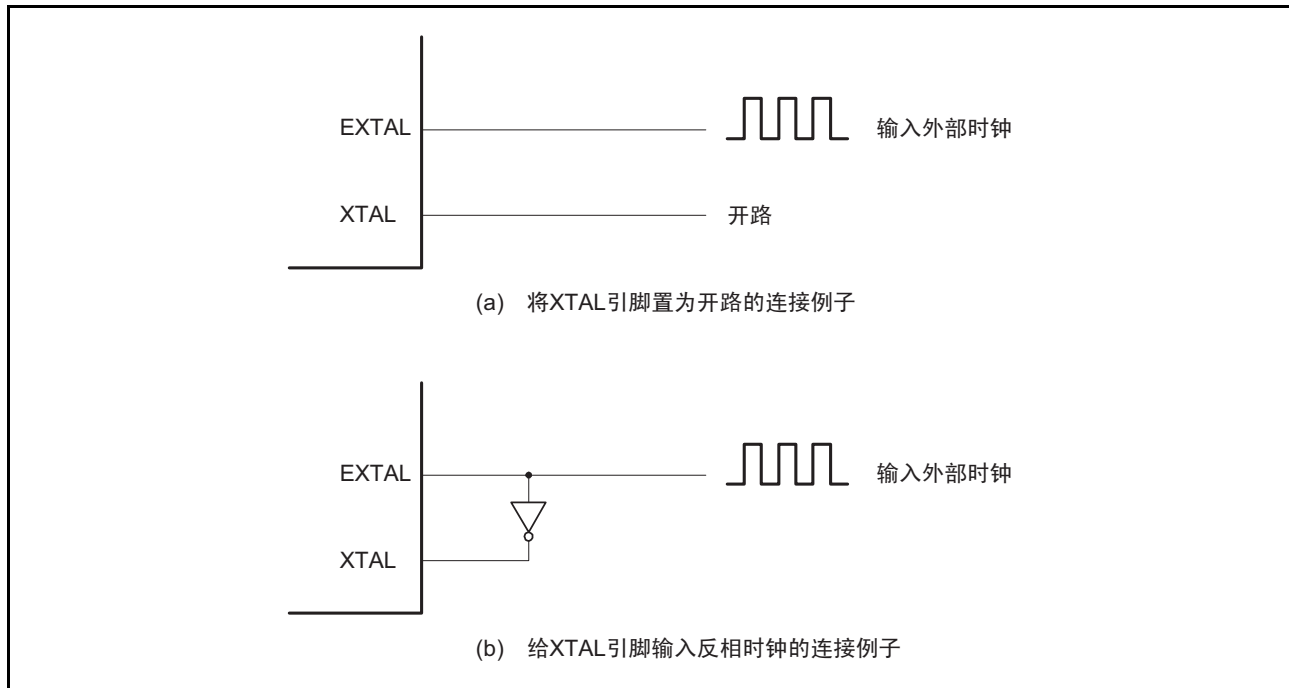


图 8.4 外部时钟的连接例子

## 8.4 IWDTC 专用低速内部振荡器

IWDTC 专用低速内部振荡器通过内部振荡生成内部振荡器时钟 (IWDTCCLK)。

## 8.5 振荡停止检测电路

振荡停止检测电路具有检测主时钟振荡器停止振荡以及代替停止的主时钟提供内部振荡电路输出的内部振荡时钟的功能。

详细内容请参照“8.10 振荡停止检测功能”。

## 8.6 内部振荡电路

内部振荡电路通过内部振荡生成内部振荡时钟。

## 8.7 PLL 电路

PLL 电路具有将振荡器的频率递增 8 倍的功能。

## 8.8 分频器

分频器对 PLL 时钟进行分频，生成 1/2、1/4、1/8 的时钟。如果改写 SCKCR.ICK[3:0] 位和 SCKCR.PCK[3:0] 位就以选择的频率运行。

## 8.9 内部时钟

内部时钟是通过 PLL 电路将外部输入时钟（EXTAL）递增 8 倍并且通过分频器进行 1 分频、2 分频、4 分频、8 分频后的时钟，或者是 IWDT 专用低速内部振荡器通过内部振荡生成的时钟。

内部时钟有以下 3 种时钟：

- CPU、DTC、MTU3、GPT、ROM 和 RAM 的运行时钟：系统时钟（ICLK）
- 外围模块的运行时钟：外围模块时钟（PCLK）
- IWDT 的运行时钟：内部振荡器时钟（IWDTCLK）

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

### 8.9.1 系统时钟（ICLK）

系统时钟（ICLK）是 CPU、DTC、MTU3、GPT、ROM 和 RAM 的运行时钟。

通过 SCKCR.ICK[3:0] 位设定 ICLK 的频率。

不能给 ICLK 设定低于外围模块时钟（PCLK）的频率。如果设定为低于外围模块时钟（PCLK）的频率，ICLK 的设定有效，但是，ICLK 的频率就和 PCLK 的频率相同。

### 8.9.2 外围模块时钟（PCLK）

外围模块时钟（PCLK）是外围模块的运行时钟。

通过 SCKCR.PCK[3:0] 位设定 PCLK 的频率。

不能给 PCLK 设定高于系统时钟（ICLK）的频率，否则 PCLK 的频率就为 ICLK 的设定频率。

### 8.9.3 内部振荡器时钟（IWDTCLK）

内部振荡器时钟（IWDTCLK）是 IWDT 的运行时钟。

IWDTCLK 是内部振荡器通过内部振荡生成的时钟。能通过 GPT 内的 LOCO 计数功能测量 IWDTCLK 的周期。详细内容请参照“18.6 LOCO 计数功能”。

## 8.10 振荡停止检测功能

### 8.10.1 振荡停止检测和检测后的运行

振荡停止检测功能是检测主时钟振荡器停止振荡并且代替停止的主时钟提供内部振荡电路输出的内部振荡时钟的功能。能在检测到振荡停止时产生中断请求，也能在检测到振荡停止时将 MTU3 和 GPT 的输出强制置为高阻抗状态。详细内容请参照“16. 多功能定时器脉冲单元 3 (MTU3)”、“17. 端口输出允许 3 (POE3)”、“18. 通用 PWM 定时器 (GPT)”和“附录 1. 各运行模式中的端口状态”。

在 RX62T 群因主时钟振荡器的异常而使输入时钟在一定期间内持续为“0”或者“1”（参照“33. 电特性”）的情况下，检测到主时钟振荡器的振荡停止。

如果检测到振荡停止，RX62T 群就通过内部振荡电路输出的内部振荡时钟继续运行。

内部振荡时钟运行时的系统时钟 (ICLK) 的频率如表 8.6 所示。

表 8.6 使用内部振荡电路时的系统时钟 (ICLK) 的频率

系统时钟 (ICLK)	min	typ	max
ICLK 的频率 (递增 8 倍)	$8 \times f_{\text{MAIN}}$	$8 \times f_{\text{MAIN}}$	$8 \times f_{\text{MAIN}}$
ICLK 的频率 (递增 4 倍)	$4 \times f_{\text{MAIN}}$	$4 \times f_{\text{MAIN}}$	$4 \times f_{\text{MAIN}}$
ICLK 的频率 (递增 2 倍)	$2 \times f_{\text{MAIN}}$	$2 \times f_{\text{MAIN}}$	$2 \times f_{\text{MAIN}}$
ICLK 的频率 (递增 1 倍)	$f_{\text{MAIN}}$	$f_{\text{MAIN}}$	$f_{\text{MAIN}}$

注. 有关  $f_{\text{MAIN}}$ ，请参照“33. 电特性”。

以振荡停止的检测为契机，自动从主时钟转换为内部振荡时钟。即使在检测到振荡停止后主时钟振荡器重新开始振荡，也通过内部振荡时钟继续运行。但是，只在引脚复位、上电复位和电压监视复位时，因为从内部振荡时钟转换为主时钟，所以在主时钟振荡器开始振荡后，通过主时钟运行。

振荡停止检测功能在通过全部复位对 LSI 内部进行初始化时有效。要使用此功能无效，必须将 OSTDCR.OSTDE 位置“0”。在检测到振荡停止并且通过内部振荡时钟运行的状态下，不能将 OSTDCR.OSTDE 位置“0”。

### 8.10.2 振荡停止检测中断

如果在振荡停止检测功能有效时检测到振荡停止，就产生振荡停止检测中断请求 (OSTDI)。振荡停止检测中断是非屏蔽中断。因为在复位解除后的初始状态下为“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“11. 中断控制器 (ICU)”。

在检测到振荡停止后通过内部振荡时钟运行的状态是系统发生某种异常的状态，只能实施对应异常的应急措施。

### 8.10.3 有关解除深度软件待机的注意事项

在深度软件待机模式中因主时钟振荡器发生异常引起不正常振荡的状态下，不能通过中断来解除深度软件待机模式。

但是，在主时钟振荡器不振荡的状态下，能通过引脚复位来解除深度软件待机模式。在确实需要解除深度软件待机模式时，必须通过引脚复位来解除。

## 8.11 使用时的注意事项

### 8.11.1 有关时钟发生电路的注意事项

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

ICLK=8MHz~100MHz, PCLK=8MHz~50MHz

2. 外围模块（DTC、MTU3、GPT 除外）全部以 PCLK 为基准运行。因此必须注意：在更改频率前后，定时器和 SCI 等的运行速度将发生变化。另外，用于解除软件待机模式的待机时间也因更改 PCLK 频率而发生变化，详细内容请参照“9.5.3.3 解除软件待机模式后的振荡稳定时间的设定”。
3. 在系统时钟（ICLK）、外围模块时钟（PCLK）之间， $ICLK \geq PCLK$  的关系成立，并且优先 ICLK 的设定。因此，如果设定没有满足此条件，SCKCR.PCK[3:0] 的设定就无效，PCLK 为 SCKCR.ICK[3:0] 位设定的时钟频率。
4. 在写 SCKCR 寄存器后到频率变更结束的期间，如果重写 SCKCR 寄存器，就忽视此写操作。如果继续写 SCKCR 寄存器，就必须确认是否能从 SCKCR 寄存器中读到最后写的值。
5. 在写 SCKCR 寄存器后到频率变更结束的期间，禁止转移到软件待机模式。如果在更改频率中途转移到软件待机模式，就无法保证以后的运行。在写 SCKCR 寄存器后，必须在通过系统时钟至少等待 11 个周期后发行 WAIT 指令。详细内容请参照“5. I/O 寄存器”。

### 8.11.2 有关谐振器的注意事项

因为有关谐振器的各种特性与用户的电路板设计密切相关，所以请用户参照本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绝对最大额定值。

### 8.11.3 设计电路板时的注意事项

在使用晶体谐振器时，必须尽量将谐振器和电容器排列在 XTAL 引脚和 EXTAL 引脚附近。如图 8.5 所示，在振荡电路的附近不能通过信号线，否则就可能因电感而不能正常振荡。

RX62T 群的 XTAL 引脚邻接复位引脚，因此，必须通过 GND 保护复位信号，以免受到时钟信号的影响。

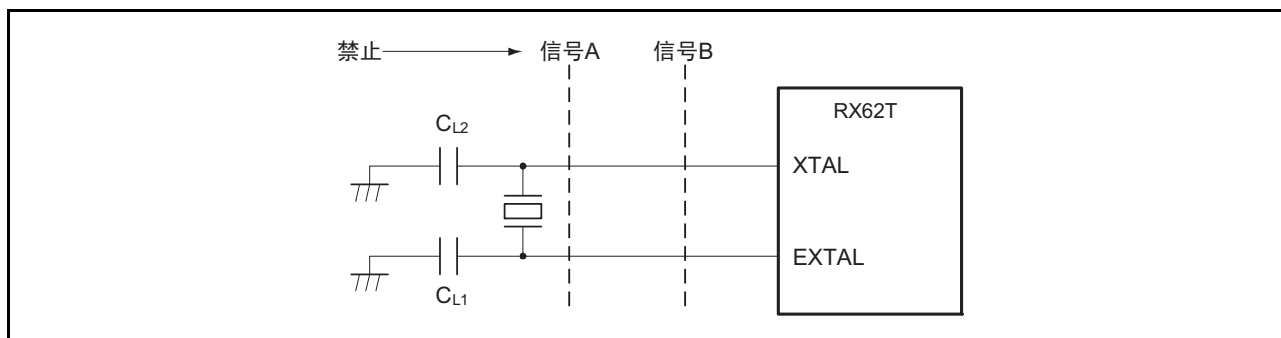


图 8.5 有关振荡电路的电路板设计的注意事项

外接 PLL 电路的推荐电路如图 8.6 所示，必须将 PLLVCC、PLLSS 和其他的 VCC、VSS 远离电路板的电源供给源，并且在引脚附近插入旁路电容器（CPB 和 CB）。

另外，PLLVCC 和 VCC 的电位必须相同。

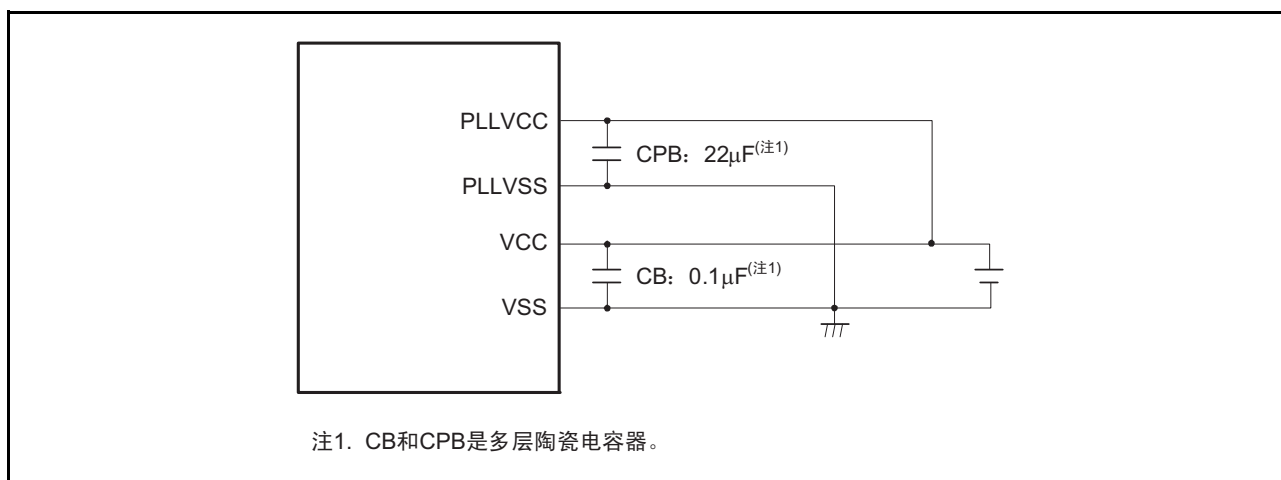


图 8.6 PLL 电路的外接推荐电路

## 9. 低功耗功能

### 9.1 概要

RX62T 群的低功耗功能有多时钟功能、模块停止功能和向低功耗状态转移的功能。

低功耗功能的规格如表 9.1 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 9.2 所示。

复位后，在正常的程序运行中，DTC 以外的模块为停止状态。

表 9.1 低功耗功能的规格

项目	内容
多时钟功能	能给系统时钟（ICLK）和外围模块时钟（PCLK）分别设定分频比。
模块停止功能	能停止各外围模块的功能。
向低功耗状态转移的功能	能进入使 CPU、外围模块和振荡器停止运行的低功耗状态。
低功耗状态	<ul style="list-style-type: none"> <li>• 睡眠模式</li> <li>• 全模块时钟停止模式</li> <li>• 软件待机模式</li> <li>• 深度软件待机模式</li> </ul>

表 9.2 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）
振荡器	运行	运行	停止	停止
CPU	停止（保持）	停止（保持）	停止（保持）	停止（不定值）
内部 RAM （0000 0000h ~ 0000 3FFFh）	运行（保持）	停止（保持）	停止（保持）	停止（不定值）
看门狗定时器（WDT）	运行	运行	停止（保持）	停止（不定值）
独立看门狗定时器（IWDT）	运行	运行	停止（保持）	停止（不定值）
电压检测电路	运行	运行	运行	运行
上电复位电路	运行	运行	运行	运行
外围模块	运行	停止（注 5）	停止（注 5）	停止（不定值）
I/O 引脚状态	运行	保持	保持	保持

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. 这是外部中断和部分内部中断（WDT、振荡停止检测和电压监视）。

注 2. 这是外部中断和部分内部中断（电压监视）。

注 3. 这是 NMI、IRQ0-A、IRQ1-A 和部分内部中断（电压监视）。但是，这些中断源只在深度待机中断允许寄存器（DPSIER）的对应位为“1”时有效。

注 4. 通过 RES# 引脚、上电复位、电压监视复位、看门狗定时器复位和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、WDT 复位和 IWDT 复位进行解除时，就转移到复位状态。

注 5. 外围模块保持原来的状态。



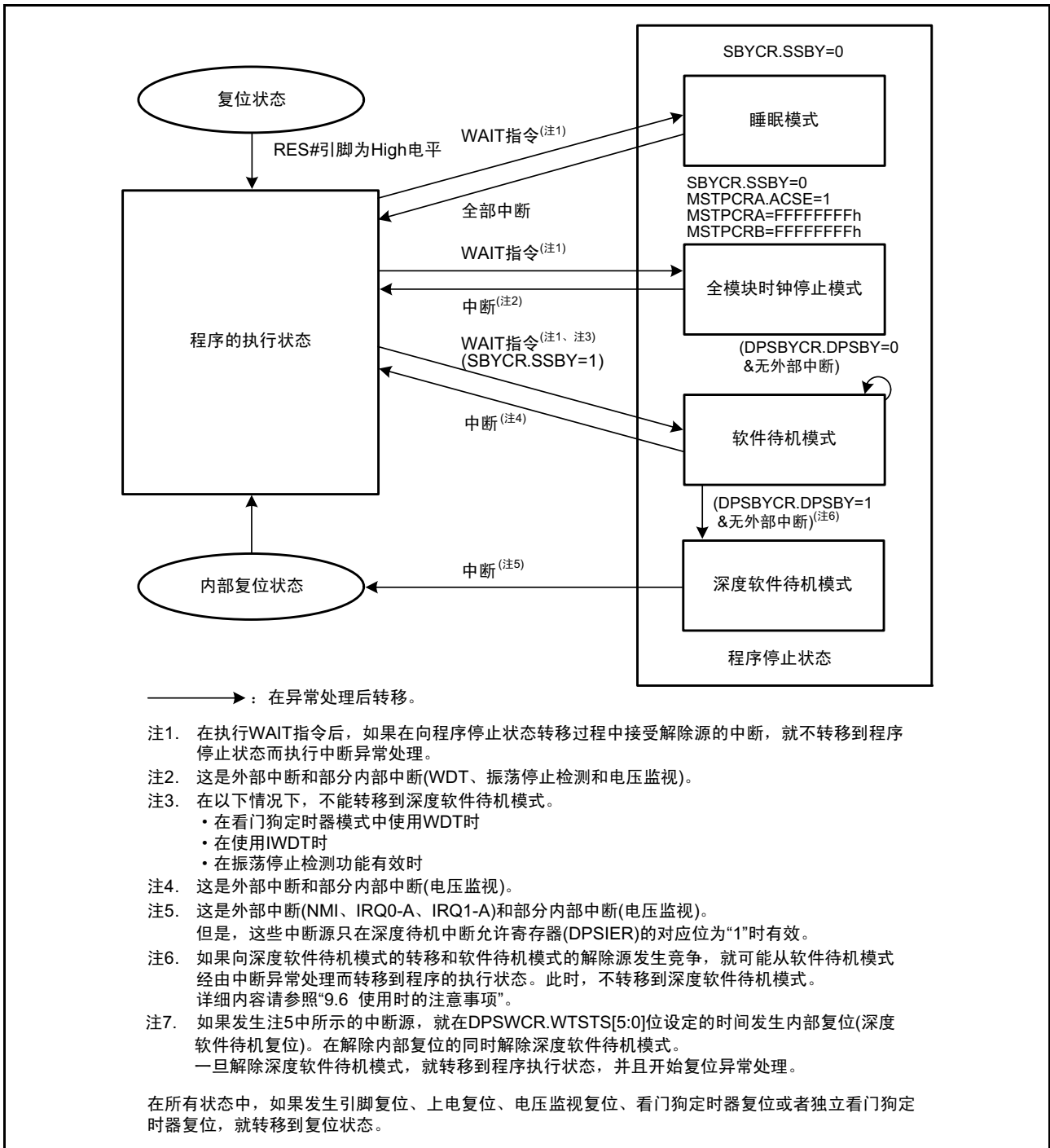


图 9.1 模式转移

## 9.2 寄存器说明

低功耗功能的相关寄存器如表 9.3 所示。有关系统时钟控制寄存器（SCKCR），请参照“9.2.1 待机控制寄存器（SBYCR）”。

表 9.3 低功耗功能的相关寄存器一览表

寄存器名	符号	复位后的值	地址	存取长度
待机控制寄存器	SBYCR	4F00h	0008 000Ch	16
模块停止控制寄存器 A	MSTPCRA	4xFF FFFFh	0008 0010h	32
模块停止控制寄存器 B	MSTPCRB	FFFF FFFFh	0008 0014h	32
模块停止控制寄存器 C	MSTPCRC	FFFF 0000h	0008 0018h	32
深度待机控制寄存器	DPSBYCR	31h	0008 C280h	8
深度待机等待控制寄存器	DPSWCR	0Fh	0008 C281h	8
深度待机中断允许寄存器	DPSIER	00h	0008 C282h	8
深度待机中断标志寄存器	DPSIFR	00h	0008 C283h	8
深度待机中断边沿寄存器	DPSIEGR	00h	0008 C284h	8
复位状态寄存器	RSTSR	x000 0xxxh	0008 C285h	8
深度待机备用寄存器 0	DPSBKR0	xxh (注 1)	0008 C290h	8
深度待机备用寄存器 1	DPSBKR1	xxh (注 1)	0008 C291h	8
深度待机备用寄存器 2	DPSBKR2	xxh (注 1)	0008 C292h	8
深度待机备用寄存器 3	DPSBKR3	xxh (注 1)	0008 C293h	8
深度待机备用寄存器 4	DPSBKR4	xxh (注 1)	0008 C294h	8
深度待机备用寄存器 5	DPSBKR5	xxh (注 1)	0008 C295h	8
深度待机备用寄存器 6	DPSBKR6	xxh (注 1)	0008 C296h	8
深度待机备用寄存器 7	DPSBKR7	xxh (注 1)	0008 C297h	8
深度待机备用寄存器 8	DPSBKR8	xxh (注 1)	0008 C298h	8
深度待机备用寄存器 9	DPSBKR9	xxh (注 1)	0008 C299h	8
深度待机备用寄存器 10	DPSBKR10	xxh (注 1)	0008 C29Ah	8
深度待机备用寄存器 11	DPSBKR11	xxh (注 1)	0008 C29Bh	8
深度待机备用寄存器 12	DPSBKR12	xxh (注 1)	0008 C29Ch	8
深度待机备用寄存器 13	DPSBKR13	xxh (注 1)	0008 C29Dh	8
深度待机备用寄存器 14	DPSBKR14	xxh (注 1)	0008 C29Eh	8
深度待机备用寄存器 15	DPSBKR15	xxh (注 1)	0008 C29Fh	8
深度待机备用寄存器 16	DPSBKR16	xxh (注 1)	0008 C2A0h	8
深度待机备用寄存器 17	DPSBKR17	xxh (注 1)	0008 C2A1h	8
深度待机备用寄存器 18	DPSBKR18	xxh (注 1)	0008 C2A2h	8
深度待机备用寄存器 19	DPSBKR19	xxh (注 1)	0008 C2A3h	8
深度待机备用寄存器 20	DPSBKR20	xxh (注 1)	0008 C2A4h	8
深度待机备用寄存器 21	DPSBKR21	xxh (注 1)	0008 C2A5h	8
深度待机备用寄存器 22	DPSBKR22	xxh (注 1)	0008 C2A6h	8
深度待机备用寄存器 23	DPSBKR23	xxh (注 1)	0008 C2A7h	8
深度待机备用寄存器 24	DPSBKR24	xxh (注 1)	0008 C2A8h	8
深度待机备用寄存器 25	DPSBKR25	xxh (注 1)	0008 C2A9h	8
深度待机备用寄存器 26	DPSBKR26	xxh (注 1)	0008 C2AAh	8
深度待机备用寄存器 27	DPSBKR27	xxh (注 1)	0008 C2ABh	8
深度待机备用寄存器 28	DPSBKR28	xxh (注 1)	0008 C2ACh	8
深度待机备用寄存器 29	DPSBKR29	xxh (注 1)	0008 C2ADh	8
深度待机备用寄存器 30	DPSBKR30	xxh (注 1)	0008 C2AEh	8
深度待机备用寄存器 31	DPSBKR31	xxh (注 1)	0008 C2AFh	8

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

### 9.2.1 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	—	—	STS[4:0]				—	—	—	—	—	—	—	—	—
复位后的值	0	1	0	0	1	1	1	1	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
b12-b8	STS[4:0]	待机定时器选择位	b12 b8 0 0 1 0 1: 待机时间 =64 个状态 0 0 1 1 0: 待机时间 =512 个状态 0 0 1 1 1: 待机时间 =1024 个状态 0 1 0 0 0: 待机时间 =2048 个状态 0 1 0 0 1: 待机时间 =4096 个状态 0 1 0 1 0: 待机时间 =16384 个状态 0 1 0 1 1: 待机时间 =32768 个状态 0 1 1 0 0: 待机时间 =65536 个状态 0 1 1 0 1: 待机时间 =131072 个状态 0 1 1 1 0: 待机时间 =262144 个状态 0 1 1 1 1: 待机时间 =524288 个状态 上述以外: 不能设定	R/W
b13	—	保留位	读写值都为“0”。	R/W
b14	—	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

SBYCR 寄存器是控制软件待机模式的寄存器。

#### STS[4:0] 位 (待机定时器选择位)

在通过外部中断解除软件待机模式时, 此位选择时钟稳定前的待机时间。

在使用晶体振荡的情况下, 请参照表 9.4, 根据工作频率选择大于等于振荡稳定时间的待机时间; 在使用外部时钟的情况下, 也需要 PLL 电路的稳定时间, 请参照表 9.4 设定待机时间。

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

#### SSBY 位 (软件待机位)

此位设定执行 WAIT 指令后的转移目标。

当 SSBY 位为“0”时, 根据 MSTPCRA 寄存器和 MSTPCRB 寄存器的设定, 在执行 WAIT 指令后转移到睡眠模式或者全模块时钟停止模式; 当 SSBY 位为“1”时, 在执行 WAIT 指令后转移到软件待机模式。此时, 如果 DPSBYCR.DPSBY 位为“1”, 就经由软件待机模式转移到深度软件待机模式, 详细内容请参照“9.5 低功耗状态”。

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

在看门狗定时器模式中使用 WDT 时, 或者在使用 IWDT 时, 或者在振荡停止检测功能有效时, 此位的设定值无效, 并且在执行 WAIT 指令后总是转移到睡眠模式或者全模块时钟停止模式。

## 9.2.2 模块停止控制寄存器 A (MSTPCRA)

地址 0008 0010h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ACSE	—	—	MSTPA 28	—	—	—	MSTPA 24	MSTPA 23	—	—	—	—	—	MSTPA 17	MSTPA 16
复位后的值	0	1	0	0	x	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
MSTPA 15	MSTPA 14	—	—	—	—	MSTPA 9	—	MSTPA 7	—	—	—	—	—	—	—
复位后的值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

x: 不定值

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 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 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 和 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 和 CMT1) 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b16	MSTPA16	12 位 A/D 转换器 (单元 1) 的模块停止设定位	对象模块: 12 位 ADC 单元 1 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b17	MSTPA17	12 位 A/D 转换器 (单元 0) 的模块停止设定位	对象模块: 12 位 ADC 单元 0 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b22-b18	—	保留位	读写值都为“1”。	R/W
b23	MSTPA23 (注 1)	10 位 A/D 转换器的模块停止设定位	对象模块: 10 位 ADC 0: 解除模块停止状态 1: 转移到模块停止状态	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: S12ADA 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	—	保留位	读取值为不定值, 只能写“1”。	R/W
b28	MSTPA28	数据传送控制器的模块停止设定位	对象模块: DTC 0: 解除模块停止状态 1: 转移到模块停止状态	R/W

位	符号	位名	功能	R/W
b29	—	保留位	读写值都为“1”。	R/W
b30	—	保留位	读写值都为“1”。	R/W
b31	ACSE (注2)	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

注 1. 在 64 引脚版中为保留位。读写值都为“1”。

注 2. 当 SBYCR.SSBY 位为“0”并且 MSTPCRA.ACSE 位为“0”时，在执行 WAIT 指令后转移到睡眠模式。

MSTPCRA 寄存器是控制模块停止状态的寄存器。

#### ACSE 位（全模块时钟停止模式允许位）

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

## 9.2.3 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 29	—	—	—	—	—	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
	—	—	—	—	—	—	—	—	MSTPB 7	—	—	—	—	—	—	MSTPB 0
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
b0	MSTPB0	CAN 模块停止设定位 (注 1)	对象模块: CAN 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b6-b1	—	保留位	读写值都为“1”。	R/W
b7	MSTPB7	LIN 模块停止设定位	对象模块: LIN 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b16-b8	—	保留位	读写值都为“1”。	R/W
b17	MSTPB17	串行外围接口的模块停止设定位	对象模块: RSPI 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 <sup>2</sup> C 总线接口的模块停止设定位	对象模块: RIIC 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
b28-b24	—	保留位	读写值都为“1”。	R/W
b29	MSTPB29	串行通信接口 2 的模块停止设定位	对象模块: SCI2 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b30	MSTPB30	串行通信接口 1 的模块停止设定位	对象模块: SCI1 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b31	MSTPB31	串行通信接口 0 的模块停止设定位	对象模块: SCI0 0: 解除模块停止状态 1: 转移到模块停止状态	R/W

注 1 在无 CAN 功能版中为保留位。读写值都为“1”。

MSTPCRB 寄存器是控制模块停止状态的寄存器。

### 9.2.4 模块停止控制寄存器 C (MSTPCRC)

地址 0008 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
复位后的值	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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 (注 1)	RAM 的模块停止设定位	对象模块: RAM (0000 0000h ~ 0000 3FFFh) 0: 解除模块停止状态 1: 转移到模块停止状态	R/W
b15-b1	—	保留位	读写值都为“0”。	R/W
b31-b16	—	保留位	读写值都为“1”。	R/W

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

MSTPCRC 寄存器是控制模块停止状态的寄存器。

### 9.2.5 深度待机控制寄存器 (DPSBYCR)

地址 0008 C280h

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

位	符号	位名	功能	R/W
b0	—	保留位	读写值都为“1”。	R/W
b3-b1	—	保留位	读写值都为“0”。	R/W
b5-b4	—	保留位	读写值都为“1”。	R/W
b6	IOKEEP	I/O 端口保持位	0: 在解除深度软件待机模式的同时, 解除 I/O 端口的保持状态。 1: 如果在解除深度软件待机模式后给 IOKEEP 位写“0”, 就解除 I/O 端口的保持状态。	R/W
b7	DPSBY	深度软件待机位	SSBY b7 0 0: 在执行 WAIT 指令后, 转移到睡眠模式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

DPSBYCR 寄存器是控制深度软件待机模式的寄存器。

通过 RES# 引脚的复位信号、上电复位和电压监视复位而不通过解除深度软件待机模式的内部复位信号, 对 DPSBYCR 寄存器进行初始化。

#### IOKEEP 位 (I/O 端口保持位)

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

#### DPSBY 位 (深度软件待机位)

此位控制深度软件待机模式的转移。

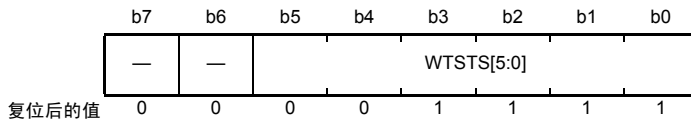
如果在 SBYCR.SSBY 位和 DPSBY 位都为“1”的状态下执行 WAIT 指令, 就经由软件待机模式转移到深度软件待机模式。如果通过外部中断引脚或者部分内部中断 (电压监视) 解除深度软件待机模式, DPSBY 位就保持“1”。要将此位置“0”时, 必须写“0”。

在看门狗定时器模式中使用 WDT 时, 或者在使用 IWDT 时, 或者在振荡停止检测功能有效时, 此位的设定值无效。此时, 即使在 SBYCR.SSBY 位和 DPSBY 位都为“1”的状态下, 也总是在执行 WAIT 指令后转移到睡眠模式或者全模块时钟停止模式。



## 9.2.6 深度待机等待控制寄存器 (DPSWCR)

地址 0008 C281h



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

在通过外部中断引脚或者部分内部中断（电压监视）解除深度软件待机模式时，通过 DPSWCR 寄存器选择时钟稳定前的 LSI 待机时间。

通过 RES# 引脚的复位信号、上电复位和电压监视复位而不通过解除深度软件待机模式的内部复位信号，对 DPSWCR 寄存器进行初始化。

### WTSTS[5:0] 位（深度软件待机的等待时间设定位）

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

在使用晶体振荡的情况下，请参照表 9.4，根据工作频率选择大于等于振荡稳定时间的待机时间。在使用外部时钟的情况下，也需要 PLL 电路的稳定时间，请参照表 9.4 设定待机时间。

在振荡稳定期间，以 EXTAL 输入时钟的频率进行计数。

## 9.2.7 深度待机中断允许寄存器 (DPSIER)

地址 0008 C282h

	b7	b6	b5	b4	b3	b2	b1	b0
	DNMIE	—	—	DLVDE	—	—	DIRQ1E	DIRQ0E
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	DIRQ0E	IRQ0 引脚允许位	0: 禁止通过 IRQ0 引脚解除深度软件待机模式 1: 允许通过 IRQ0 引脚解除深度软件待机模式	R/W
b1	DIRQ1E	IRQ1 引脚允许位	0: 禁止通过 IRQ1 引脚解除深度软件待机模式 1: 允许通过 IRQ1 引脚解除深度软件待机模式	R/W
b3-b2	—	保留位	读写值都为“0”	R/W
b4	DLVDE	LVD 深度待机解除信号允许位	0: 禁止通过电压监视中断解除深度软件待机模式 1: 允许通过电压监视中断解除深度软件待机模式	R/W
b6-b5	—	保留位	读写值都为“0”	R/W
b7	DNMIE	NMI 引脚允许位	0: 禁止通过 NMI 引脚解除深度软件待机模式 1: 允许通过 NMI 引脚解除深度软件待机模式	R/(W) (注1)

注 1. 在通过 RES# 引脚的复位信号、上电复位和电压监视复位解除初始化后，只能写 1 次“1”，以后的写操作无效。

DPSIER 寄存器允许或者禁止解除深度软件待机模式的外部中断引脚以及内部解除信号。

通过 RES# 引脚的复位信号、上电复位和电压监视复位而不通过解除深度软件待机模式的内部复位信号，对 DPSIER 寄存器进行初始化。

如果更改 DPSIER 寄存器的设定，对应引脚的输入缓冲器的内部状态就发生变化。此时，根据引脚的状态可能在内部产生边沿，并且 DPSIER 寄存器可能变为“1”。必须在向深度软件待机模式转移前，将 DPSIER 寄存器置“0”。

另外，在向深度软件待机模式转移时，DPSIER 寄存器为“0”的引脚的输入缓冲器无效。此时，根据引脚状态可能在内部产生上升沿，并且 DPSIER 寄存器可能变为“1”。但是，因为在 DPSIEGR 寄存器为“0”时不检测上升沿，所以 DPSIER 寄存器不变为“1”。

## 9.2.8 深度待机中断标志寄存器 (DPSIFR)

地址 0008 C283h

b7	b6	b5	b4	b3	b2	b1	b0
DNMIF	—	—	DLVDF	—	—	DIRQ1F	DIRQ0F
复位后的值	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	DIRQ0F	IRQ0 深度待机解除标志	0: 未产生由 IRQ0 引脚引起的解除请求 1: 产生由 IRQ0 引脚引起的解除请求	R/(W) (注1)
b1	DIRQ1F	IRQ1 深度待机解除标志	0: 未产生由 IRQ1 引脚引起的解除请求 1: 产生由 IRQ1 引脚引起的解除请求	R/(W) (注1)
b3-b2	—	保留位	读写值都为“0”	R/W
b4	DLVDF	LVD 深度待机解除标志	0: 未产生由电压监视信号引起的解除请求 1: 产生由电压监视信号引起的解除请求	R/(W) (注1)
b6-b5	—	保留位	读写值都为“0”	R/W
b7	DNMIF	NMI 深度待机解除标志	0: 未产生由 NMI 引脚引起的解除请求 1: 产生由 NMI 引脚引起的解除请求	R/(W) (注1)

注 1. 只能写“0”。

DPSIFR 寄存器是保持深度软件待机模式的解除请求的寄存器。

如果产生深度待机中断沿寄存器 (DPSIEGR) 设定的解除请求, DPSIFR 寄存器就变为“1”。即使在非深度软件待机模式的状态下产生解除请求, DPSIFR 寄存器也变为“1”, 因此必须在将 DPSIFR 寄存器置“0”后转移到深度软件待机模式。

通过 RES# 引脚的复位信号、上电复位和电压监视复位而不通过解除深度软件待机模式的内部复位信号, 对 DPSIFR 寄存器进行初始化。

### DIRQnF 标志 (IRQn 深度待机解除标志) (n=0、1)

此标志表示产生由 IRQn 引脚引起的解除请求。

[ 为“1”的条件 ]

- 产生由 DPSIEGR 寄存器选择的 IRQn 引脚引起的解除请求时

[ 为“0”的条件 ]

- 读“1”后写“0”时

### DLVDF 标志 (LVD 深度待机解除标志)

此标志表示产生由电压监视信号引起的解除请求。

[ 为“1”的条件 ]

- 产生由电压监视信号引起的解除请求时

[ 为“0”的条件 ]

- 读“1”后写“0”时

### DNMIF 标志 (NMI 深度待机解除标志)

此标志表示产生由 NMI 引脚引起的解除请求。

[ 为“1”的条件 ]

- 产生由 DPSIEGR 寄存器设定的 NMI 引脚引起的解除请求时

[ 为“0”的条件 ]

- 读“1”后写“0”时

## 9.2.9 深度待机中断边沿寄存器 (DPSIEGR)

地址 0008 C284h

	b7	b6	b5	b4	b3	b2	b1	b0
	DNMIEG	—	—	—	—	—	DIRQ1EG	DIRQ0EG
复位后的值	0	0	0	0	0	0	0	0

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

DPSIEGR 寄存器选择用于解除深度软件待机模式的解除信号的边沿。

通过 RES# 引脚的复位信号、上电复位和电压监视复位而不通过解除深度软件待机模式的内部复位信号，对 DPSIEGR 寄存器进行初始化。

### 9.2.10 复位状态寄存器 (RSTSR)

地址 0008 C285h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSRSTF	—	—	—	—	LVD2F	LVD1F	RORF
复位后的值	0/1 (注1)	0	0	0	0	0/1 (注1)	0/1 (注1)	0/1 (注1)

注1. 初始值取决于复位源。

位	符号	位名	功能	R/W
b0	PORF	上电复位标志	0: 未发生上电复位 1: 发生上电复位	R
b1	LVD1F	LVD1 检测标志	0: 未检测到 LVD1 1: 检测到 LVD1	R/(W) (注1)
b2	LVD2F	LVD2 检测标志	0: 未检测到 LVD2 1: 检测到 LVD2	R/(W) (注1)
b6-b3	—	保留位	读写值都为“0”。	R/W
b7	DPSRSTF	深度软件待机复位标志	0: 未产生由外部中断和部分内部中断（电压监视）引起的深度软件待机模式的解除源 1: 产生由外部中断和部分内部中断（电压监视）引起的深度软件待机模式的解除源	R/(W) (注1)

注1. 只能写用于清除标志的“0”。

RSTSR 寄存器是表示内部复位发生源的寄存器。

#### PORF 标志（上电复位标志）

此标志表示发生上电复位。

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

[为“1”的条件]

- 发生上电复位时

[为“0”的条件]

- 通过 RES# 引脚进行复位时

#### LVD1F 标志（LVD1 检测标志）

此标志表示检测到低于等于 Vdet1 电平的 VCC 电压。

通过 RES# 引脚的复位信号和上电复位而不通过解除深度软件待机模式的内部复位信号，对 LVD1F 标志进行初始化。

[为“1”的条件]

- 检测到低于等于 Vdet1 电平的 VCC 电压时

[为“0”的条件]

- 在 LVD1F.LVD1E 位为“1”并且 VCC 高于 Vdet1 并且经过稳定时间的状态下读“1”后写“0”时
- 通过 RES# 引脚进行复位时
- 发生上电复位时

**LVD2F 标志（LVD2 检测标志）**

此标志表示检测到低于等于  $V_{det2}$  电平的 VCC 电压。

通过 RES# 引脚的复位信号和上电复位而不通过解除深度软件待机模式的内部复位信号，对 LVD2F 标志进行初始化。

[ 为“1”的条件 ]

- 检测到低于等于  $V_{det2}$  电平的 VCC 电压时

[ 为“0”的条件 ]

- 在 LVDCR.LVD2E 位为“1”并且 VCC 高于  $V_{det2}$  并且经过稳定时间的状态下读“1”后写“0”时
- 通过 RES# 引脚进行复位时
- 发生上电复位时
- 发生由 LVD1 引起的复位时

**DPSRSTF 标志（深度软件待机复位标志）**

此标志表示通过 DPSIER 寄存器和 DPSIEGR 寄存器设定的外部中断源或者部分内部中断（电压监视），解除了深度软件待机模式并且发生了内部复位。

通过 RES# 引脚的复位信号、上电复位和电压监视复位而不通过解除深度软件待机模式的内部复位信号，对 DPSRSTF 标志进行初始化。

[ 为“1”的条件 ]

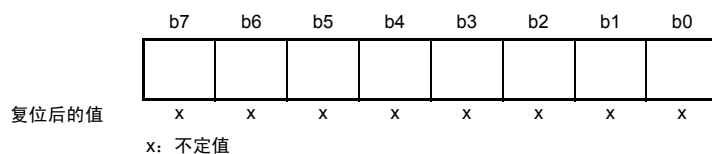
- 通过外部中断源或者部分内部中断（电压监视）解除了深度软件待机模式时

[ 为“0”的条件 ]

- 读“1”后写“0”时
- 通过 RES# 引脚进行复位时
- 发生上电复位时
- 发生由 LVD1 引起的复位时
- 发生由 LVD2 引起的复位时

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

地址 0008 C290h ~ 0008 C2AFh



DPSBKRY 寄存器能在深度软件待机模式中读写用于保存数据的 32 字节。

即使在不保持内部 RAM 数据的深度软件待机模式中，也保持此寄存器的值。

不对 DPSBKRY 寄存器进行初始化，刚接通电源后的寄存器的值为不定值。

### 9.3 多时钟功能

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

CPU 和总线主控通过 `ICK[3:0]` 位设定的运行时钟工作，外围模块通过 `PCK[3:0]` 位设定的运行时钟工作。详细内容请参照“8. 时钟发生电路”。

### 9.4 模块停止功能

能以内部外围模块为单位设定模块停止功能。

如果将 `MSTPCRA ~ MSTPCRC` 寄存器对应的 `MSTPyj` 位 ( $y=A \sim C, j=31 \sim 0$ ) 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 `MSTPyj` 位置“0”，解除模块停止状态并且模块在总线周期结束时重新开始运行。

在模块停止状态下，保持模块的内部状态。

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

## 9.5 低功耗状态

### 9.5.1 睡眠模式

#### 9.5.1.1 向睡眠模式的转移

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

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

在使用 WDT 时，如果转移到睡眠模式，WDT 就停止计数。

在使用睡眠模式时，必须在进行以下的设定后执行 WAIT 指令。

1. 将 CPU 的 PSW.I 位（注1）置“0”。
2. 将用于从睡眠模式返回的中断的请求目标设定为 CPU。
3. 将用于从睡眠模式返回的中断的优先级（注2）设定为高于 CPU 的 PSW.IPL[2:0] 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. 读最后执行写操作的 I/O 寄存器，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“11. 中断控制器（ICU）”。

#### 9.5.1.2 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、WDT 上溢复位或者 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，睡眠模式就被解除并且开始中断异常处理。如果可屏蔽中断被 CPU 屏蔽（中断优先级（注1）小于等于 CPU 的 PSW.IPL[3:0] 位（注2）的优先级），就不解除睡眠模式。
- 通过 RES# 引脚复位进行的解除  
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 WDT 上溢复位进行的解除  
通过 WDT 上溢的内部复位来解除睡眠模式。
- 通过 IWDT 下溢复位进行的解除  
通过 IWDT 下溢的内部复位来解除睡眠模式。
- 通过电压监视复位进行的解除  
通过电压监视电路的电压监视复位来解除睡眠模式。
- 通过上电复位进行的解除  
通过上电复位来解除睡眠模式。

注 1. 详细内容请参照“11. 中断控制器（ICU）”。

注 2. 详细内容请参照“2. CPU”。



## 9.5.2 全模块时钟停止模式

### 9.5.2.1 向全模块时钟停止模式的转移

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

- MSTPCRA.ACSE 位为“1”。
- 在由 MSTPCRA 寄存器和 MSTPCRB 寄存器控制的全部模块设定为模块停止状态（MSTPCRA、MSTPCRB=FFFFFFFh）。

在全模块时钟停止模式中，看门狗定时器、独立看门狗定时器、电压检测电路和上电复位以外的全部外围模块、CPU、总线控制器和 I/O 端口停止运行。

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

在使用全模块时钟停止模式时，必须在进行以下的设定后执行 WAIT 指令：

1. 将 CPU 的 PSW.I 位（注2）置“0”。
2. 将用于从全模块时钟停止模式返回的中断的请求目标设定为 CPU。
3. 将用于从全模块时钟停止模式返回的中断的优先级（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.IENn 位（注3）置“1”。
5. 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. 有时因 DTC 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 DTC 的 DTCST.DTCST 位置“0”，并且在不启动 DTC 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“11. 中断控制器（ICU）”。

### 9.5.2.2 全模块时钟停止模式的解除

在通过外部中断（NMI 引脚和 IRQ0 ~ IRQ7 引脚）、RES# 引脚、电压监视复位、上电复位或者内部中断（WDT、电压监视、振荡停止检测）解除全模块时钟停止模式后，经由异常处理状态转移到正常的程序执行状态。如果可屏蔽中断被 CPU 屏蔽（中断优先级（注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 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“11. 中断控制器（ICU）”。

注 2. 详细内容请参照“2. CPU”。

### 9.5.3 软件待机模式

#### 9.5.3.1 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。在此模式中，CPU、内部外围功能和振荡器的全部功能停止运行，但是保持 CPU 内部寄存器的值、内部 RAM 的数据、内部外围功能和 I/O 端口状态。在软件待机模式中，因为振荡器停止振荡，所以功耗明显降低。

必须在执行 WAIT 指令前将 DTC.DTCST.DTCST 位置“0”。

在看门狗定时器模式中使用 WDT 时，或者在使用 IWDT 时，不能转移到软件待机模式。必须在执行 WAIT 指令前停止 WDT。

在振荡停止检测功能有效时（注1），不能转移到软件待机模式。要向软件待机模式转移时，必须在将振荡停止检测功能设定为无效后发行 WAIT 指令。

在使用软件待机模式时，必须在进行以下的设定后执行 WAIT 指令：

1. 将 CPU 的 PSW.I 位（注2）置“0”。
2. 将用于从软件待机模式返回的中断的请求目标设定为 CPU。
3. 将用于从软件待机模式返回的中断的优先级（注3）设定为高于 CPU.PSW.IPL[3:0] 位（注2）的优先级。
4. 将用于从软件待机模式返回的中断的 IERm.IENn 位（注3）置“1”。
5. 执行 WAIT 指令（通过执行 WAIT 指令，CPU.PSW.I 位（注2）自动变为“1”）。

注 1. 在解除复位后，振荡停止检测功能（OSTDCR.OSTDE 位）有效。

注 2. 详细内容请参照“2. CPU”。

注 3. 详细内容请参照“11. 中断控制器（ICU）”。

#### 9.5.3.2 软件待机模式的解除

通过外部中断（NMI 引脚和 IRQ0 ~ IRQ7（注1））、部分内部中断（电压监视）、RES# 引脚复位、上电复位或者电压监视复位来解除软件待机模式。

##### 1. 通过中断进行的解除

如果通过 NMI、IRQ0 ~ IRQ7（注1）和电压监视输入中断请求信号，时钟就开始振荡，在经过 SBYCR.STS[4:0] 位选择的时间后给整个 RX62T 提供稳定的时钟，然后解除软件待机模式，开始中断异常处理。

##### 2. 通过 RES# 引脚复位进行的解除

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

##### 3. 通过上电复位进行的解除

如果因电源电压降低而发生上电复位，就解除软件待机模式。

##### 4. 通过电压监视复位进行的解除

如果因电源电压降低而发生电压监视复位，就解除软件待机模式并且时钟开始振荡。

注 1. 详细内容请参照“11. 中断控制器（ICU）”。

## 9.5.3.3 解除软件待机模式后的振荡稳定时间的设定

必须通过 SBYCR.STS[4:0] 位进行以下的设定：

1. 使用晶体振荡的情况  
设定 STS[4:0] 位，使待机时间大于等于振荡稳定时间。  
工作频率、STS[4:0] 位的设定与待机时间的对应如表 9.4 所示。
2. 使用外部时钟的情况  
需要 PLL 电路的稳定时间，请参照表 9.4 设定待机时间。

表 9.4 振荡稳定时间的设定

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

■：使用外部时钟时的推荐设定时间

■：使用晶体振荡时的推荐设定时间

注 1. PCLK 是外围模块分频器的输出时钟。

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

### 9.5.3.4 软件待机模式的应用例子

在 IRQ 引脚的下降沿转移到软件待机模式并且在 IRQ 引脚的上升沿解除软件待机模式的运行例子如图 9.2 所示。

此例中，在 ICU 的 `IRQCRi.IRQMD[1:0]` 位被设定为“01b”（下降沿）的状态下接受 IRQ 中断后，先将 `IRQMD[1:0]` 位置“10b”（上升沿），再将 `SBYCR.SSBY` 位置“1”，然后执行 WAIT 指令，转移到软件待机模式。此后，在 IRQ 引脚的上升沿解除软件待机模式。

要从软件待机模式返回时，还需要设定中断控制器（ICU），详细内容请参照“11. 中断控制器（ICU）”。

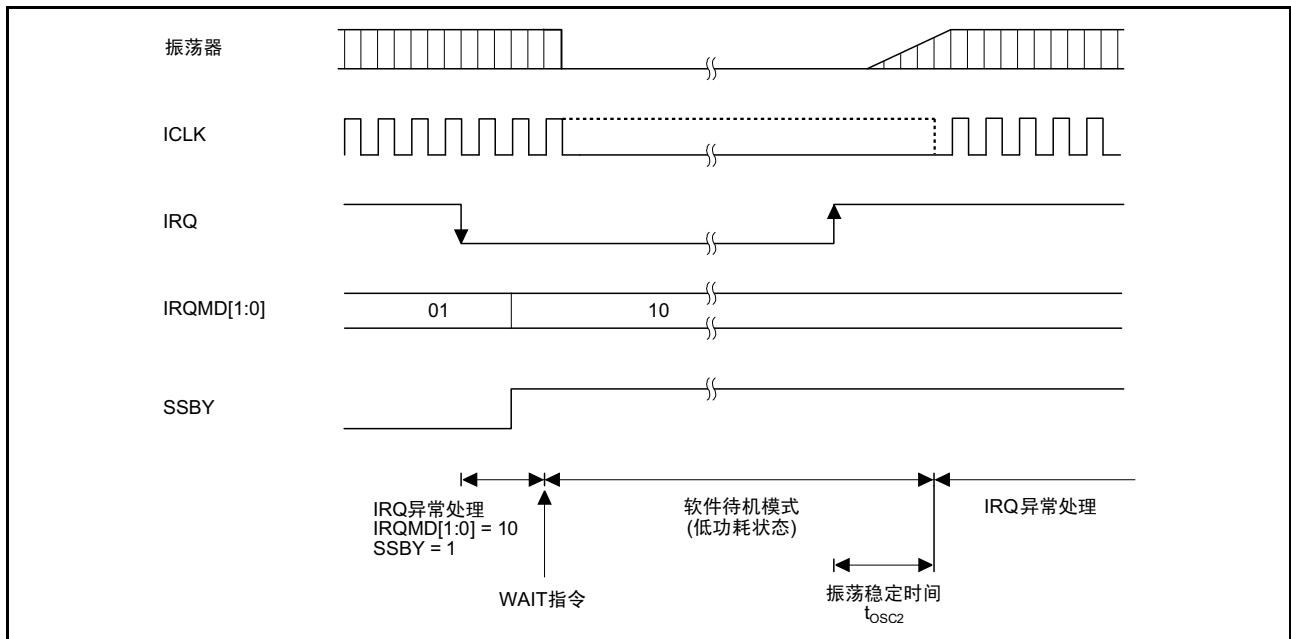


图 9.2 软件待机模式的应用例子

## 9.5.4 深度软件待机模式

### 9.5.4.1 向深度软件待机模式的转移

如果在 `SBYCR.SSBY` 位为“1”的状态下执行 WAIT 指令，就转移到软件待机模式（注 1）。此时，如果 `DPSBYCR.DPSBY` 位为“1”，就转移到深度软件待机模式。如果在转移到软件待机模式时发生和软件待机模式解除请求（NMI 和 `IRQ0 ~ IRQ7` 的中断请求、部分内部中断（电压监视））的竞争，就解除软件待机模式，与 `DPSBY` 位的设定无关。在经过 `SBYCR.STS[4:0]` 位选择的软件待机模式的振荡稳定时间后，开始中断异常处理。

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

在深度软件待机模式中，因为 CPU、内部外围功能、内部 RAM 和振荡器的全部功能停止运行，而且停止提供内部电源，所以功耗明显降低。此时，CPU 和内部外围功能的寄存器内容全部为不定值，内部 RAM 的数据也全部为不定值。

电压检测电路和上电复位电路也在深度软件待机模式中有效。

端口保持软件待机模式中的状态。

注 1. 必须在执行 WAIT 指令前满足向软件待机模式转移时的 DTC、WDT、IWD 和振荡停止检测功能的相关条件，详细内容请参照“9.5.3 软件待机模式”。

### 9.5.4.2 深度软件待机模式的解除

通过外部中断引脚（NMI 引脚、IRQ0-A 引脚和 IRQ1-A 引脚）、部分内部中断（电压监视）、RES# 引脚的复位、上电复位或者电压监视复位来解除深度软件待机模式。

#### 1. 通过外部中断或者内部中断进行的解除

DPSIFR 寄存器是保持深度软件待机模式解除源的寄存器，当产生解除请求时，此寄存器就变为“1”。此时，如果通过 DPSIER 寄存器允许解除源，就解除深度软件待机模式。

如果在 DPSIER.DNMIE 位或者 DPSIER.DIRQnE 位（n=1、0）允许的 NMI 引脚或者 IRQ0-A ~ IRQ1-A 引脚产生边沿，就将 DPSIFR.DNMIF 标志、DPSIFR.DIRQnF 标志置“1”。能通过 DPSIEGR 寄存器选择各引脚的上升沿和下降沿。

另外，如果发生 DPSIER.DLDVE 位允许的电压监视中断，就将 DLVDF 标志置“1”。

如果产生深度软件待机模式的解除请求，就在开始时钟振荡的同时开始提供内部电源，并且对整个 LSI 产生内部复位信号。

在经过 DPSWCR.WTSTS[5:0] 位选择的待机时间后，给整个 LSI 提供稳定的时钟并且解除内部复位。

如果在解除内部复位的同时解除深度软件待机模式，就开始复位异常处理。

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

#### 2. 通过 RES# 引脚复位进行的解除

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

#### 3. 通过上电复位进行的解除

如果因电源电压降低而发生上电复位，就解除深度软件待机模式。

#### 4. 通过电压监视复位进行的解除

如果因电源电压降低而发生电压监视复位，就解除深度软件待机模式。

### 9.5.4.3 解除深度软件待机模式时的引脚状态

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

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

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

不通过解除深度软件待机模式的内部复位对 IOKEEP 位进行初始化。

## 9.5.4.4 解除深度软件待机模式后的振荡稳定时间的设定

必须对 DPSWCR.WTSTS[5:0] 位进行以下的设定：

## 1. 使用晶体振荡的情况

设定 WTSTS[5:0] 位，使待机时间大于等于振荡稳定时间。

EXTAL 输入时钟的频率、WTSTS[5:0] 位的设定与待机时间的对应如表 9.5 所示。

## 2. 使用外部时钟的情况

需要 PLL 电路的稳定时间，请参照表 9.5 设定待机时间。

表 9.5 振荡稳定时间的设定

WTSTS5	WTSTS4	WTSTS3	WTSTS2	WTSTS1	WTSTS0	待机时间 (状态)	EXTAL 输入时钟的频率 (注1) (MHz)			单位	
							12	10	8		
0	0	0	0	0	0	(保留)	—	—	—	μs	
					1	(保留)	—	—	—		
				1	0	(保留)	—	—	—		
					1	(保留)	—	—	—		
			1	0	0	(保留)	—	—	—		
					1	64	5.3	6.4	8.0		
				1	0	512	42.7	51.2	64.0		
					1	1024	85.3	102.4	128.0		
		1	0	0	0	0	2048	170.7	204.8	256.0	ms
						1	4096	0.34	0.41	0.51	
					1	0	16384	1.37	1.64	2.05	
						1	32768	2.73	3.26	4.10	
				1	0	0	65536	5.46	6.55	8.19	
						1	131072	10.92	13.11	16.38	
					1	0	262144	21.85	26.21	32.77	
						1	524288	43.69	52.43	65.54	
1	x	x	x	x	x	(保留)	—	—	—		
						(保留)	—	—	—		

：使用外部时钟时的推荐设定时间

：使用晶体振荡时的推荐设定时间

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

9.5.4.5 深度软件待机模式的应用例子

在 IRQ 引脚的下降沿转移到深度软件待机模式，并且在 IRQ 引脚的上升沿解除深度软件待机模式的运行例子如图 9.3 所示。

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

此后，在 IRQ 引脚的上升沿解除深度软件待机模式。

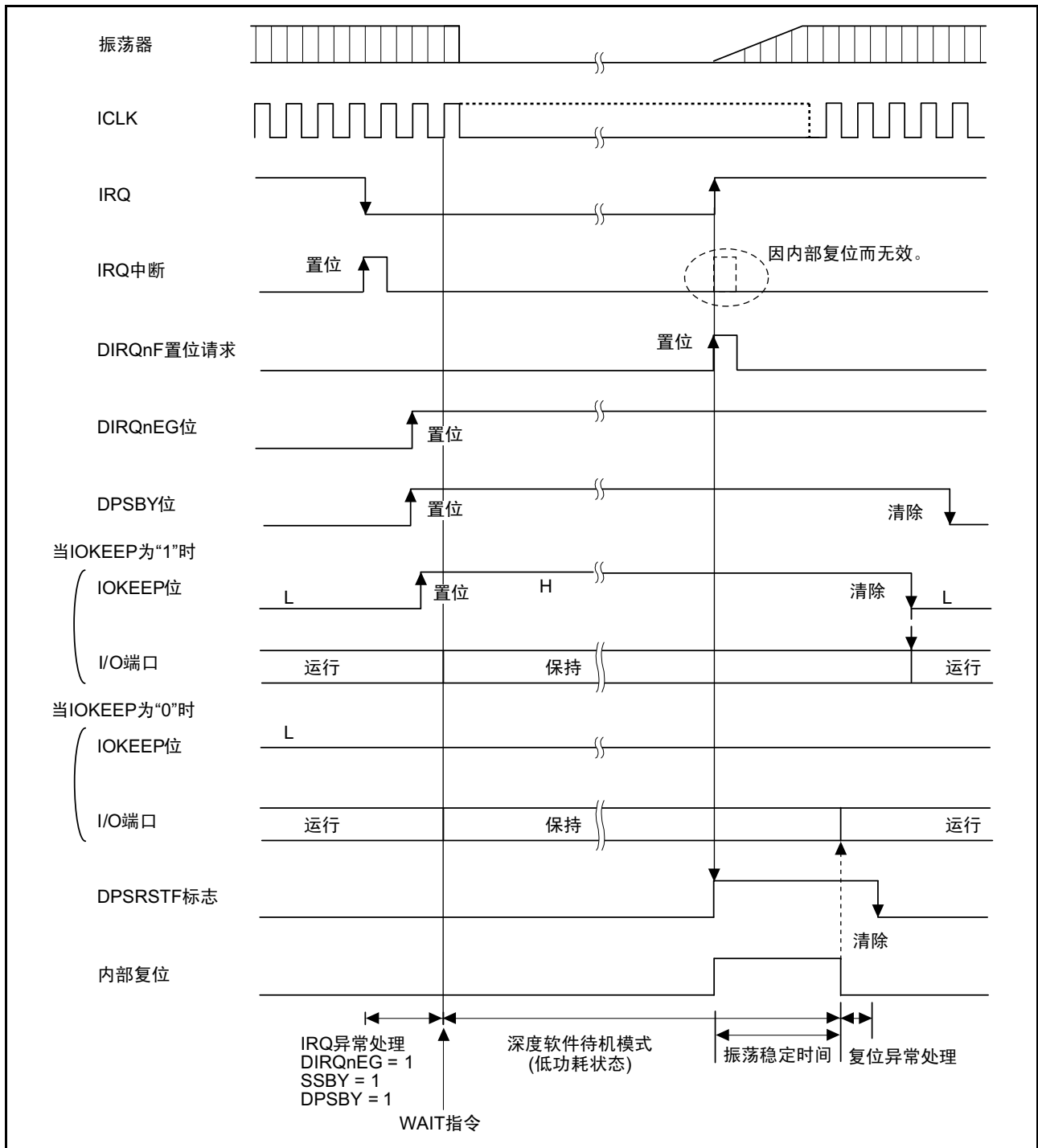


图 9.3 深度软件待机模式的应用例子

9.5.4.6 深度软件待机模式的流程图

使用深度软件待机模式时的流程图例子如图 9.4 所示。

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

在通过 RES# 引脚进行复位时，在进行各种设定后转移到深度软件待机模式。

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

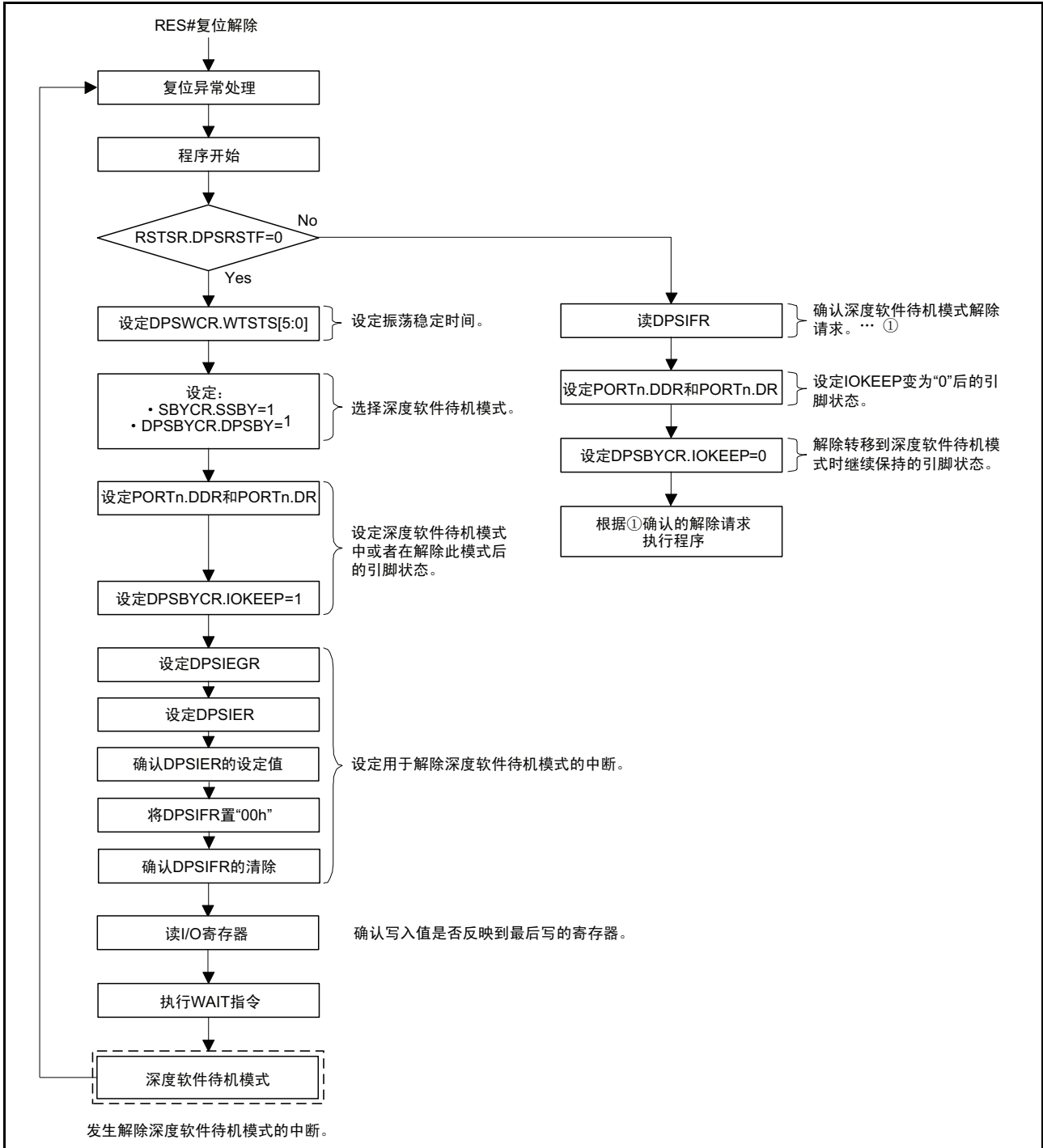


图 9.4 深度软件待机模式的流程图例子



## 9.6 使用时的注意事项

### 9.6.1 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。如果输出 High 电平，就无法降低输出电流的消耗电流。

### 9.6.2 DTC 的模块停止

必须在将 MSTPCRA.MSTPA28 位置“1”前，将 DTC 的 DTCST.DTCST 位置“0”，并且在未启动 DTC 的状态下进行设定。详细内容请参照“14. 数据传送控制器（DTC）”

### 9.6.3 内部外围模块的中断

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

### 9.6.4 MSTPCRA、MSTPCRB、MSTPCRC 寄存器的写操作

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

### 9.6.5 通过 DIRQnE 位（n=1、0）控制输入缓冲器

能通过将 DPSIER.DIRQnE 位（n=1、0）位置“1”，将 P10/IRQ0-A、P11/IRQ1-A 引脚的输入缓冲器设定为有效。因此必须注意：这些引脚的输入反映到 DPSIFR.DIRQnF 位（n=1、0）而不反映到中断控制器、外围模块和 I/O 端口。

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

### 9.6.6 深度软件待机模式的转移和中断的竞争

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

必须注意：在向深度软件待机模式的转移和软件待机模式的解除请求发生竞争时，需要执行中断异常处理程序。

### 9.6.7 WAIT 指令的执行时序

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

## 10. 异常处理

### 10.1 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 10.1 所示。

如果发生异常，处理器模式就转移到管理模式。

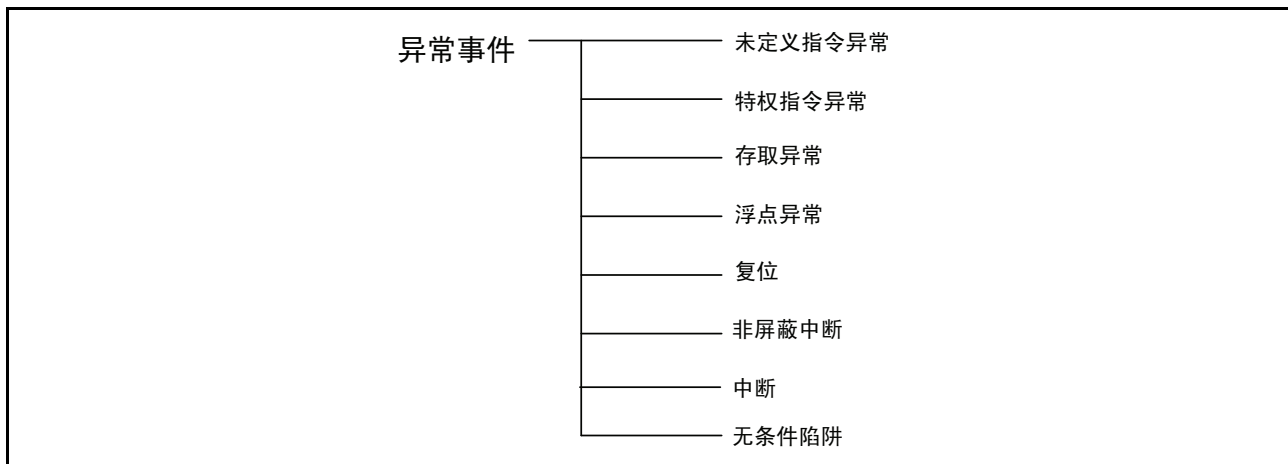


图 10.1 异常事件的种类

#### 10.1.1 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。

#### 10.1.2 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。

#### 10.1.3 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。

#### 10.1.4 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。

#### 10.1.5 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。

### 10.1.6 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。

### 10.1.7 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。

### 10.1.8 无条件陷阱

如果执行 INT 指令和 BRK 指令，就产生无条件陷阱。

### 10.2 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 10.2 所示。

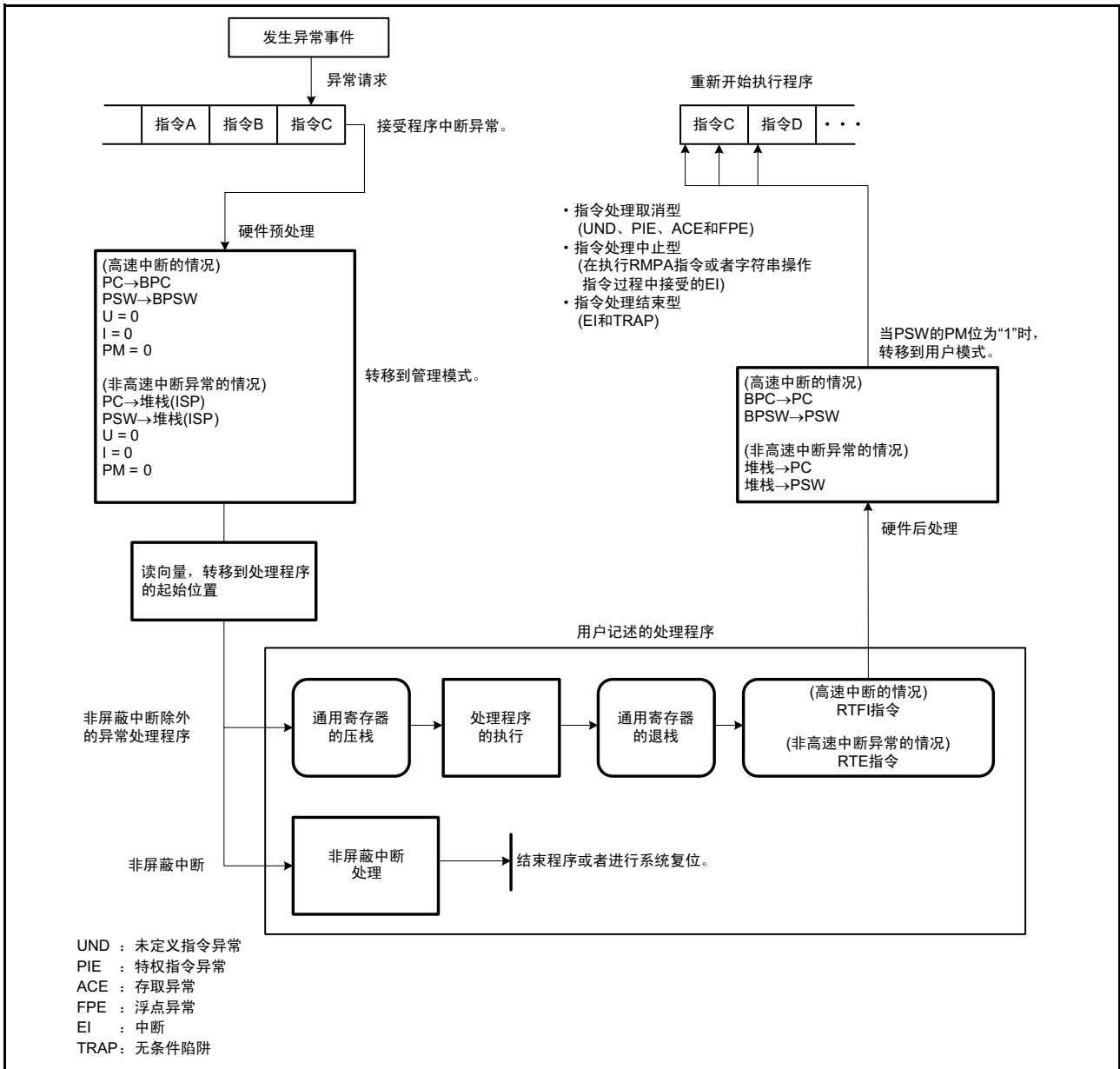


图 10.2 异常处理步骤的概要

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）的内容保存到备用程序计数器（BPC），处理器状态字（PSW）的内容保存到备用处理器状态字（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。

### 10.3 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。

#### 10.3.1 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 10.1 所示。

表 10.1 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		指令处理放弃型	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

#### 10.3.2 向量和 PC、PSW 的保存场所

各异常事件的向量、程序计数器（PC）和处理器状态字（PSW）的保存场所如表 10.2 所示。

表 10.2 向量和 PC、PSW 的保存场所

异常事件		向量	PC 和 PSW 的保存场所
未定义指令异常		固定向量表	堆栈
特权指令异常		固定向量表	堆栈
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

## 10.4 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

## 10.5 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。

### 10.5.1 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。

### 10.5.2 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。

### 10.5.3 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。

### 10.5.4 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。

### 10.5.5 复位

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

### 10.5.6 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。

### 10.5.7 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。

### 10.5.8 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。



## 10.6 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 10.3 所示的指令，就恢复异常处理顺序前保存到堆栈区或者控制寄存器（BPC 和 BPSW）的程序计数器（PC）和处理器状态字（PSW）的内容。


表 10.3 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

## 10.7 异常事件的优先级

异常事件的优先级如表 10.4 所示。如果同时发生多个异常，就先接受优先级高的事件。

表 10.4 异常事件的优先级

优先级		异常事件
高  低	1	复位
	2	非屏蔽中断
	3	中断
	4	指令存取异常
	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

## 11. 中断控制器 (ICU)

### 11.1 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。中断控制器的规格和框图分别如表 11.1 和图 11.1 所示。

表 11.1 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"> <li>• 外围模块的中断</li> <li>• 中断源数：101 个</li> <li>• 中断检测：边沿检测 / 电平检测</li> </ul> 给连接外围模块的各中断源规定了检测方法。
	外部引脚中断	<ul style="list-style-type: none"> <li>• IRQ7 ~ IRQ0 引脚的中断</li> <li>• 中断源数：8 个</li> <li>• 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。</li> </ul>
	软件中断	通过写寄存器产生中断。 <ul style="list-style-type: none"> <li>• 中断源数：1 个</li> </ul>
	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 的控制	能通过中断源启动 DTC。 DTC 启动源：87 个（78 个外围功能中断 + 8 个外部引脚中断 + 1 个软件中断）
非屏蔽中断	NMI 引脚中断	NMI 引脚的中断 <ul style="list-style-type: none"> <li>• 中断检测：下降沿 / 上升沿</li> </ul>
	电压监视中断	检测到低电源电压时的中断
	振荡停止检测中断	检测到振荡停止时的中断
从低功耗状态的返回		<ul style="list-style-type: none"> <li>• 睡眠模式：通过非屏蔽中断或者全部中断源返回。</li> <li>• 全模块时钟停止模式：通过非屏蔽中断、IRQ7 ~ IRQ0 中断、或者 WDT 中断返回。</li> <li>• 软件待机模式：通过非屏蔽中断或者 IRQ7 ~ IRQ0 中断返回。</li> </ul>

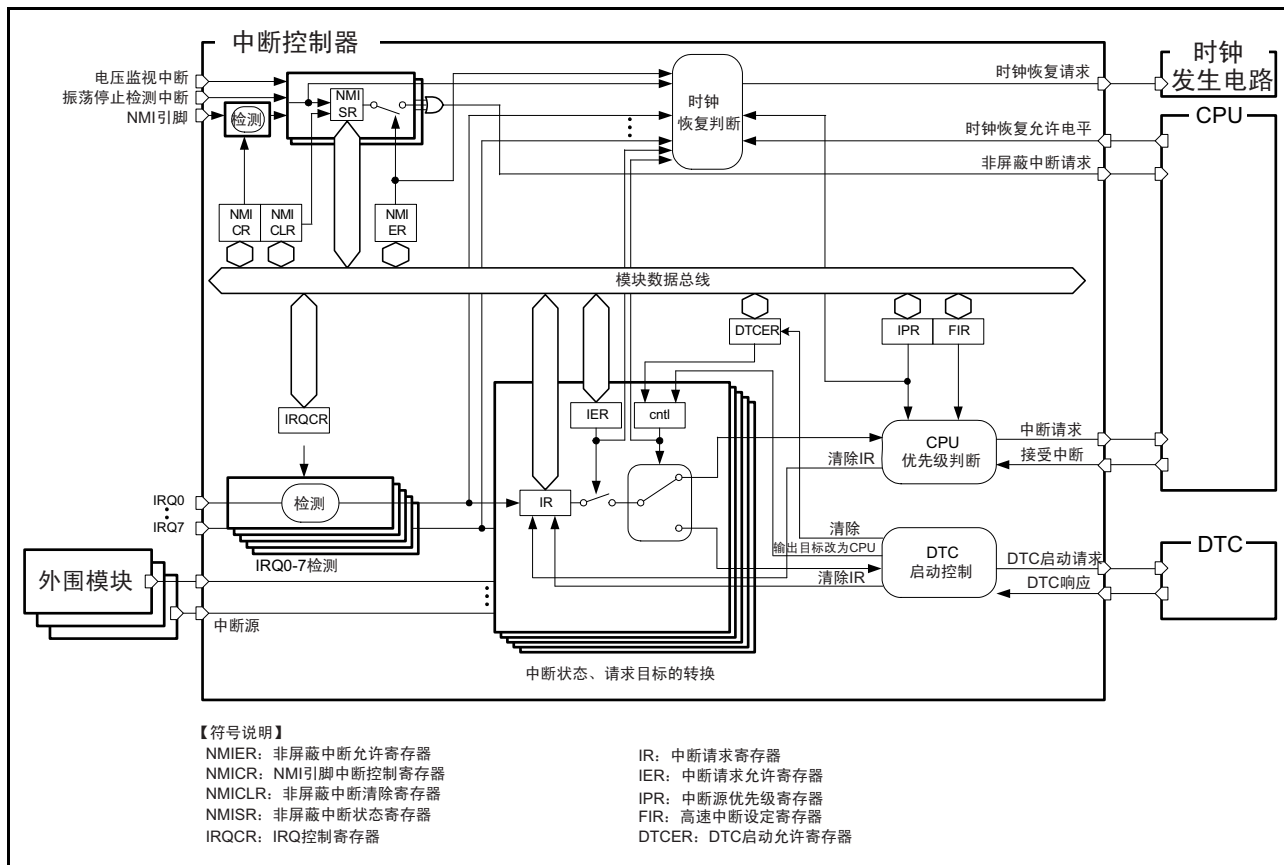


图 11.1 中断控制器的框图

中断控制器使用的输入 / 输出引脚如表 11.2 所示。

表 11.2 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
IRQ7 ~ IRQ0	输入	外部中断请求引脚

## 11.2 寄存器说明

中断控制器的寄存器一览表如表 11.3 所示。

表 11.3 中断控制器的寄存器一览表 (1/7)

寄存器名	符号	复位后的值	地址	存取长度
中断请求寄存器 016	IR016	00h	0008 7010h	8
中断请求寄存器 021	IR021	00h	0008 7015h	8
中断请求寄存器 023	IR023	00h	0008 7017h	8
中断请求寄存器 027	IR027	00h	0008 701Bh	8
中断请求寄存器 028	IR028	00h	0008 701Ch	8
中断请求寄存器 029	IR029	00h	0008 701Dh	8
中断请求寄存器 030	IR030	00h	0008 701Eh	8
中断请求寄存器 031	IR031	00h	0008 701Fh	8
中断请求寄存器 044	IR044	00h	0008 702Ch	8
中断请求寄存器 045	IR045	00h	0008 702Dh	8
中断请求寄存器 046	IR046	00h	0008 702Eh	8
中断请求寄存器 047	IR047	00h	0008 702Fh	8
中断请求寄存器 056	IR056	00h	0008 7038h	8
中断请求寄存器 057	IR057	00h	0008 7039h	8
中断请求寄存器 058	IR058	00h	0008 703Ah	8
中断请求寄存器 059	IR059	00h	0008 703Bh	8
中断请求寄存器 060	IR060	00h	0008 703Ch	8
中断请求寄存器 064	IR064	00h	0008 7040h	8
中断请求寄存器 065	IR065	00h	0008 7041h	8
中断请求寄存器 066	IR066	00h	0008 7042h	8
中断请求寄存器 067	IR067	00h	0008 7043h	8
中断请求寄存器 068	IR068	00h	0008 7044h	8
中断请求寄存器 069	IR069	00h	0008 7045h	8
中断请求寄存器 070	IR070	00h	0008 7046h	8
中断请求寄存器 071	IR071	00h	0008 7047h	8
中断请求寄存器 096	IR096	00h	0008 7060h	8
中断请求寄存器 098	IR098	00h	0008 7062h	8
中断请求寄存器 102	IR102	00h	0008 7066h	8
中断请求寄存器 103	IR103	00h	0008 7067h	8
中断请求寄存器 106	IR106	00h	0008 706Ah	8
中断请求寄存器 114	IR114	00h	0008 7072h	8
中断请求寄存器 115	IR115	00h	0008 7073h	8
中断请求寄存器 116	IR116	00h	0008 7074h	8
中断请求寄存器 117	IR117	00h	0008 7075h	8
中断请求寄存器 118	IR118	00h	0008 7076h	8
中断请求寄存器 119	IR119	00h	0008 7077h	8
中断请求寄存器 120	IR120	00h	0008 7078h	8
中断请求寄存器 121	IR121	00h	0008 7079h	8
中断请求寄存器 122	IR122	00h	0008 707Ah	8

表 11.3 中断控制器的寄存器一览表 (2/7)

寄存器名	符号	复位后的值	地址	存取长度
中断请求寄存器 123	IR123	00h	0008 707Bh	8
中断请求寄存器 124	IR124	00h	0008 707Ch	8
中断请求寄存器 125	IR125	00h	0008 707Dh	8
中断请求寄存器 126	IR126	00h	0008 707Eh	8
中断请求寄存器 127	IR127	00h	0008 707Fh	8
中断请求寄存器 128	IR128	00h	0008 7080h	8
中断请求寄存器 129	IR129	00h	0008 7081h	8
中断请求寄存器 130	IR130	00h	0008 7082h	8
中断请求寄存器 131	IR131	00h	0008 7083h	8
中断请求寄存器 132	IR132	00h	0008 7084h	8
中断请求寄存器 133	IR133	00h	0008 7085h	8
中断请求寄存器 134	IR134	00h	0008 7086h	8
中断请求寄存器 135	IR135	00h	0008 7087h	8
中断请求寄存器 136	IR136	00h	0008 7088h	8
中断请求寄存器 137	IR137	00h	0008 7089h	8
中断请求寄存器 138	IR138	00h	0008 708Ah	8
中断请求寄存器 139	IR139	00h	0008 708Bh	8
中断请求寄存器 140	IR140	00h	0008 708Ch	8
中断请求寄存器 141	IR141	00h	0008 708Dh	8
中断请求寄存器 142	IR142	00h	0008 708Eh	8
中断请求寄存器 143	IR143	00h	0008 708Fh	8
中断请求寄存器 144	IR144	00h	0008 7090h	8
中断请求寄存器 145	IR145	00h	0008 7091h	8
中断请求寄存器 146	IR146	00h	0008 7092h	8
中断请求寄存器 149	IR149	00h	0008 7095h	8
中断请求寄存器 150	IR150	00h	0008 7096h	8
中断请求寄存器 151	IR151	00h	0008 7097h	8
中断请求寄存器 152	IR152	00h	0008 7098h	8
中断请求寄存器 153	IR153	00h	0008 7099h	8
中断请求寄存器 170	IR170	00h	0008 70AAh	8
中断请求寄存器 171	IR171	00h	0008 70ABh	8
中断请求寄存器 172	IR172	00h	0008 70ACh	8
中断请求寄存器 173	IR173	00h	0008 70ADh	8
中断请求寄存器 174	IR174	00h	0008 70AEh	8
中断请求寄存器 175	IR175	00h	0008 70AFh	8
中断请求寄存器 176	IR176	00h	0008 70B0h	8
中断请求寄存器 177	IR177	00h	0008 70B1h	8
中断请求寄存器 178	IR178	00h	0008 70B2h	8
中断请求寄存器 179	IR179	00h	0008 70B3h	8
中断请求寄存器 180	IR180	00h	0008 70B4h	8
中断请求寄存器 181	IR181	00h	0008 70B5h	8
中断请求寄存器 182	IR182	00h	0008 70B6h	8

表 11.3 中断控制器的寄存器一览表 (3/7)

寄存器名	符号	复位后的值	地址	存取长度
中断请求寄存器 183	IR183	00h	0008 70B7h	8
中断请求寄存器 184	IR184	00h	0008 70B8h	8
中断请求寄存器 186	IR186	00h	0008 70BAh	8
中断请求寄存器 187	IR187	00h	0008 70BBh	8
中断请求寄存器 188	IR188	00h	0008 70BCh	8
中断请求寄存器 189	IR189	00h	0008 70BDh	8
中断请求寄存器 190	IR190	00h	0008 70BEh	8
中断请求寄存器 192	IR192	00h	0008 70C0h	8
中断请求寄存器 193	IR193	00h	0008 70C1h	8
中断请求寄存器 194	IR194	00h	0008 70C2h	8
中断请求寄存器 195	IR195	00h	0008 70C3h	8
中断请求寄存器 196	IR196	00h	0008 70C4h	8
中断请求寄存器 214	IR214	00h	0008 70D6h	8
中断请求寄存器 215	IR215	00h	0008 70D7h	8
中断请求寄存器 216	IR216	00h	0008 70D8h	8
中断请求寄存器 217	IR217	00h	0008 70D9h	8
中断请求寄存器 218	IR218	00h	0008 70DAh	8
中断请求寄存器 219	IR219	00h	0008 70DBh	8
中断请求寄存器 220	IR220	00h	0008 70DCh	8
中断请求寄存器 221	IR221	00h	0008 70DDh	8
中断请求寄存器 222	IR222	00h	0008 70DEh	8
中断请求寄存器 223	IR223	00h	0008 70DFh	8
中断请求寄存器 224	IR224	00h	0008 70E0h	8
中断请求寄存器 225	IR225	00h	0008 70E1h	8
中断请求寄存器 246	IR246	00h	0008 70F6h	8
中断请求寄存器 247	IR247	00h	0008 70F7h	8
中断请求寄存器 248	IR248	00h	0008 70F8h	8
中断请求寄存器 249	IR249	00h	0008 70F9h	8
中断请求寄存器 254	IR254	00h	0008 70FEh	8
DTC 启动允许寄存器 027	DTCER027	00h	0008 711Bh	8
DTC 启动允许寄存器 028	DTCER028	00h	0008 711Ch	8
DTC 启动允许寄存器 029	DTCER029	00h	0008 711Dh	8
DTC 启动允许寄存器 030	DTCER030	00h	0008 711Eh	8
DTC 启动允许寄存器 031	DTCER031	00h	0008 711Fh	8
DTC 启动允许寄存器 045	DTCER045	00h	0008 712Dh	8
DTC 启动允许寄存器 046	DTCER046	00h	0008 712Eh	8
DTC 启动允许寄存器 064	DTCER064	00h	0008 7140h	8
DTC 启动允许寄存器 065	DTCER065	00h	0008 7141h	8
DTC 启动允许寄存器 066	DTCER066	00h	0008 7142h	8
DTC 启动允许寄存器 067	DTCER067	00h	0008 7143h	8
DTC 启动允许寄存器 068	DTCER068	00h	0008 7144h	8
DTC 启动允许寄存器 069	DTCER069	00h	0008 7145h	8

表 11.3 中断控制器的寄存器一览表 (4/7)

寄存器名	符号	复位后的值	地址	存取长度
DTC 启动允许寄存器 070	DTCER070	00h	0008 7146h	8
DTC 启动允许寄存器 071	DTCER071	00h	0008 7147h	8
DTC 启动允许寄存器 098	DTCER098	00h	0008 7162h	8
DTC 启动允许寄存器 102	DTCER102	00h	0008 7166h	8
DTC 启动允许寄存器 103	DTCER103	00h	0008 7167h	8
DTC 启动允许寄存器 106	DTCER106	00h	0008 716Ah	8
DTC 启动允许寄存器 114	DTCER114	00h	0008 7172h	8
DTC 启动允许寄存器 115	DTCER115	00h	0008 7173h	8
DTC 启动允许寄存器 116	DTCER116	00h	0008 7174h	8
DTC 启动允许寄存器 117	DTCER117	00h	0008 7175h	8
DTC 启动允许寄存器 121	DTCER121	00h	0008 7179h	8
DTC 启动允许寄存器 122	DTCER122	00h	0008 717Ah	8
DTC 启动允许寄存器 125	DTCER125	00h	0008 717Dh	8
DTC 启动允许寄存器 126	DTCER126	00h	0008 717Eh	8
DTC 启动允许寄存器 129	DTCER129	00h	0008 7181h	8
DTC 启动允许寄存器 130	DTCER130	00h	0008 7182h	8
DTC 启动允许寄存器 131	DTCER131	00h	0008 7183h	8
DTC 启动允许寄存器 132	DTCER132	00h	0008 7184h	8
DTC 启动允许寄存器 134	DTCER134	00h	0008 7186h	8
DTC 启动允许寄存器 135	DTCER135	00h	0008 7187h	8
DTC 启动允许寄存器 136	DTCER136	00h	0008 7188h	8
DTC 启动允许寄存器 137	DTCER137	00h	0008 7189h	8
DTC 启动允许寄存器 138	DTCER138	00h	0008 718Ah	8
DTC 启动允许寄存器 139	DTCER139	00h	0008 718Bh	8
DTC 启动允许寄存器 140	DTCER140	00h	0008 718Ch	8
DTC 启动允许寄存器 141	DTCER141	00h	0008 718Dh	8
DTC 启动允许寄存器 142	DTCER142	00h	0008 718Eh	8
DTC 启动允许寄存器 143	DTCER143	00h	0008 718Fh	8
DTC 启动允许寄存器 144	DTCER144	00h	0008 7190h	8
DTC 启动允许寄存器 145	DTCER145	00h	0008 7191h	8
DTC 启动允许寄存器 149	DTCER149	00h	0008 7195h	8
DTC 启动允许寄存器 150	DTCER150	00h	0008 7196h	8
DTC 启动允许寄存器 151	DTCER151	00h	0008 7197h	8
DTC 启动允许寄存器 152	DTCER152	00h	0008 7198h	8
DTC 启动允许寄存器 153	DTCER153	00h	0008 7199h	8
DTC 启动允许寄存器 174	DTCER174	00h	0008 71AEh	8
DTC 启动允许寄存器 175	DTCER175	00h	0008 71AFh	8
DTC 启动允许寄存器 176	DTCER176	00h	0008 71B0h	8
DTC 启动允许寄存器 177	DTCER177	00h	0008 71B1h	8
DTC 启动允许寄存器 178	DTCER178	00h	0008 71B2h	8
DTC 启动允许寄存器 179	DTCER179	00h	0008 71B3h	8
DTC 启动允许寄存器 180	DTCER180	00h	0008 71B4h	8

表 11.3 中断控制器的寄存器一览表 (5/7)

寄存器名	符号	复位后的值	地址	存取长度
DTC 启动允许寄存器 181	DTCER181	00h	0008 71B5h	8
DTC 启动允许寄存器 182	DTCER182	00h	0008 71B6h	8
DTC 启动允许寄存器 183	DTCER183	00h	0008 71B7h	8
DTC 启动允许寄存器 184	DTCER184	00h	0008 71B8h	8
DTC 启动允许寄存器 186	DTCER186	00h	0008 71BAh	8
DTC 启动允许寄存器 187	DTCER187	00h	0008 71BBh	8
DTC 启动允许寄存器 188	DTCER188	00h	0008 71BCh	8
DTC 启动允许寄存器 189	DTCER189	00h	0008 71BDh	8
DTC 启动允许寄存器 190	DTCER190	00h	0008 71BEh	8
DTC 启动允许寄存器 192	DTCER192	00h	0008 71C0h	8
DTC 启动允许寄存器 193	DTCER193	00h	0008 71C1h	8
DTC 启动允许寄存器 194	DTCER194	00h	0008 71C2h	8
DTC 启动允许寄存器 195	DTCER195	00h	0008 71C3h	8
DTC 启动允许寄存器 196	DTCER196	00h	0008 71C4h	8
DTC 启动允许寄存器 215	DTCER215	00h	0008 71D7h	8
DTC 启动允许寄存器 216	DTCER216	00h	0008 71D8h	8
DTC 启动允许寄存器 219	DTCER219	00h	0008 71DBh	8
DTC 启动允许寄存器 220	DTCER220	00h	0008 71DCh	8
DTC 启动允许寄存器 223	DTCER223	00h	0008 71DFh	8
DTC 启动允许寄存器 224	DTCER224	00h	0008 71E0h	8
DTC 启动允许寄存器 247	DTCER247	00h	0008 71F7h	8
DTC 启动允许寄存器 248	DTCER248	00h	0008 71F8h	8
DTC 启动允许寄存器 254	DTCER254	00h	0008 71FEh	8
中断请求允许寄存器 02	IER02	00h	0008 7202h	8
中断请求允许寄存器 03	IER03	00h	0008 7203h	8
中断请求允许寄存器 05	IER05	00h	0008 7205h	8
中断请求允许寄存器 07	IER07	00h	0008 7207h	8
中断请求允许寄存器 08	IER08	00h	0008 7208h	8
中断请求允许寄存器 0C	IER0C	00h	0008 720Ch	8
中断请求允许寄存器 0D	IER0D	00h	0008 720Dh	8
中断请求允许寄存器 0E	IER0E	00h	0008 720Eh	8
中断请求允许寄存器 0F	IER0F	00h	0008 720Fh	8
中断请求允许寄存器 10	IER10	00h	0008 7210h	8
中断请求允许寄存器 11	IER11	00h	0008 7211h	8
中断请求允许寄存器 12	IER12	00h	0008 7212h	8
中断请求允许寄存器 13	IER13	00h	0008 7213h	8
中断请求允许寄存器 15	IER15	00h	0008 7215h	8
中断请求允许寄存器 16	IER16	00h	0008 7216h	8
中断请求允许寄存器 17	IER17	00h	0008 7217h	8
中断请求允许寄存器 18	IER18	00h	0008 7218h	8
中断请求允许寄存器 1A	IER1A	00h	0008 721Ah	8
中断请求允许寄存器 1B	IER1B	00h	0008 721Bh	8



表 11.3 中断控制器的寄存器一览表 (6/7)

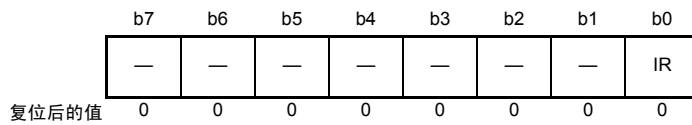
寄存器名	符号	复位后的值	地址	存取长度
中断请求允许寄存器 1C	IER1C	00h	0008 721Ch	8
中断请求允许寄存器 1E	IER1E	00h	0008 721Eh	8
中断请求允许寄存器 1F	IER1F	00h	0008 721Fh	8
软件中断启动寄存器	SWINTR	00h	0008 72E0h	8
高速中断设定寄存器	FIR	0000h	0008 72F0h	16
中断源优先级寄存器 00	IPR00	00h	0008 7300h	8
中断源优先级寄存器 01	IPR01	00h	0008 7301h	8
中断源优先级寄存器 02	IPR02	00h	0008 7302h	8
中断源优先级寄存器 03	IPR03	00h	0008 7303h	8
中断源优先级寄存器 04	IPR04	00h	0008 7304h	8
中断源优先级寄存器 05	IPR05	00h	0008 7305h	8
中断源优先级寄存器 06	IPR06	00h	0008 7306h	8
中断源优先级寄存器 07	IPR07	00h	0008 7307h	8
中断源优先级寄存器 14	IPR14	00h	0008 7314h	8
中断源优先级寄存器 18	IPR18	00h	0008 7318h	8
中断源优先级寄存器 20	IPR20	00h	0008 7320h	8
中断源优先级寄存器 21	IPR21	00h	0008 7321h	8
中断源优先级寄存器 22	IPR22	00h	0008 7322h	8
中断源优先级寄存器 23	IPR23	00h	0008 7323h	8
中断源优先级寄存器 24	IPR24	00h	0008 7324h	8
中断源优先级寄存器 25	IPR25	00h	0008 7325h	8
中断源优先级寄存器 26	IPR26	00h	0008 7326h	8
中断源优先级寄存器 27	IPR27	00h	0008 7327h	8
中断源优先级寄存器 40	IPR40	00h	0008 7340h	8
中断源优先级寄存器 44	IPR44	00h	0008 7344h	8
中断源优先级寄存器 48	IPR48	00h	0008 7348h	8
中断源优先级寄存器 49	IPR49	00h	0008 7349h	8
中断源优先级寄存器 51	IPR51	00h	0008 7351h	8
中断源优先级寄存器 52	IPR52	00h	0008 7352h	8
中断源优先级寄存器 53	IPR53	00h	0008 7353h	8
中断源优先级寄存器 54	IPR54	00h	0008 7354h	8
中断源优先级寄存器 55	IPR55	00h	0008 7355h	8
中断源优先级寄存器 56	IPR56	00h	0008 7356h	8
中断源优先级寄存器 57	IPR57	00h	0008 7357h	8
中断源优先级寄存器 58	IPR58	00h	0008 7358h	8
中断源优先级寄存器 59	IPR59	00h	0008 7359h	8
中断源优先级寄存器 5A	IPR5A	00h	0008 735Ah	8
中断源优先级寄存器 5B	IPR5B	00h	0008 735Bh	8
中断源优先级寄存器 5C	IPR5C	00h	0008 735Ch	8
中断源优先级寄存器 5D	IPR5D	00h	0008 735Dh	8
中断源优先级寄存器 5E	IPR5E	00h	0008 735Eh	8
中断源优先级寄存器 5F	IPR5F	00h	0008 735Fh	8

表 11.3 中断控制器的寄存器一览表 (7/7)

寄存器名	符号	复位后的值	地址	存取长度
中断源优先级寄存器 60	IPR60	00h	0008 7360h	8
中断源优先级寄存器 67	IPR67	00h	0008 7367h	8
中断源优先级寄存器 68	IPR68	00h	0008 7368h	8
中断源优先级寄存器 69	IPR69	00h	0008 7369h	8
中断源优先级寄存器 6A	IPR6A	00h	0008 736Ah	8
中断源优先级寄存器 6B	IPR6B	00h	0008 736Bh	8
中断源优先级寄存器 6C	IPR6C	00h	0008 736Ch	8
中断源优先级寄存器 6D	IPR6D	00h	0008 736Dh	8
中断源优先级寄存器 6E	IPR6E	00h	0008 736Eh	8
中断源优先级寄存器 6F	IPR6F	00h	0008 736Fh	8
中断源优先级寄存器 80	IPR80	00h	0008 7380h	8
中断源优先级寄存器 81	IPR81	00h	0008 7381h	8
中断源优先级寄存器 82	IPR82	00h	0008 7382h	8
中断源优先级寄存器 88	IPR88	00h	0008 7388h	8
中断源优先级寄存器 89	IPR89	00h	0008 7389h	8
中断源优先级寄存器 8A	IPR8A	00h	0008 738Ah	8
中断源优先级寄存器 8B	IPR8B	00h	0008 738Bh	8
中断源优先级寄存器 90	IPR90	00h	0008 7390h	8
IRQ 控制寄存器 0	IRQCR0	00h	0008 7500h	8
IRQ 控制寄存器 1	IRQCR1	00h	0008 7501h	8
IRQ 控制寄存器 2	IRQCR2	00h	0008 7502h	8
IRQ 控制寄存器 3	IRQCR3	00h	0008 7503h	8
IRQ 控制寄存器 4	IRQCR4	00h	0008 7504h	8
IRQ 控制寄存器 5	IRQCR5	00h	0008 7505h	8
IRQ 控制寄存器 6	IRQCR6	00h	0008 7506h	8
IRQ 控制寄存器 7	IRQCR7	00h	0008 7507h	8
非屏蔽中断状态寄存器	NMISR	00h	0008 7580h	8
非屏蔽中断允许寄存器	NMIER	00h	0008 7581h	8
非屏蔽中断清除寄存器	NMICLR	00h	0008 7582h	8
NMI 引脚中断控制寄存器	NMICR	00h	0008 7583h	8

### 11.2.1 中断请求寄存器 i (IRi) (i= 中断向量号)

地址 0008 7010h ~ 0008 70FEh



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

注 1. 在边沿检测源的情况下，能写用于清除标志的“0”。只能在“11.7 使用时的注意事项”的条件下写“1”。  
在电平检测源的情况下，不能写此位。

IRi 寄存器是中断请求的状态寄存器。

各中断源都有 IRi 寄存器，i 对应中断向量号。

有关中断源和中断向量号的对应，请参照“表 11.4 中断向量表”。

#### IR 标志 (中断状态标志)

这是中断请求的状态标志。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两种。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 IRQn 引脚中断，能通过设定 IRQCRn.IRQMD[1:0] 位 (n=0 ~ 7)，转换边沿检测和电平检测。有关各中断源的检测方法，请参照“表 11.4 中断向量表”。

##### (1) 边沿检测的情况

[为“1”的条件]

- 如果产生外围模块中断请求或者 IRQn 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产生，请参照各外围模块的章节。
- 禁止给 IR 标志写“1”，只能在“11.7 使用时的注意事项”的条件下写“1”。

[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 时，禁止给 IR 标志写“0”。

##### (2) 电平检测的情况

[为“1”的条件]

- 在产生外围模块中断请求或者 IRQn 引脚中断请求的期间，此标志总是为“1”。有关各外围模块中断请求的产生，请参照各外围模块的章节。

[为“0”的条件]

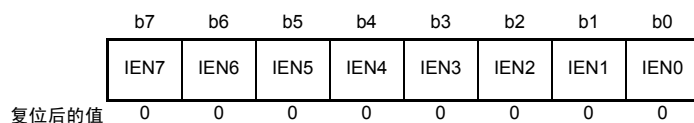
- 如果清除中断请求的输出源，此标志就变为“0”（即使中断请求目标接受中断请求，此标志也不变为“0”）。有关各外围模块中断请求的清除，请参照各外围模块的章节。

在通过电平检测使用 IRQn 引脚的情况下，要取消中断时，必须将 IRQn 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”。

## 11.2.2 中断请求允许寄存器 m (IERm) (m=02h ~ 1Fh)

地址 0008 7202h ~ 0008 721Fh



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”。

IERm 寄存器设定允许或者禁止向 CPU 请求中断以及设定允许或者禁止 DTC 的启动请求。

## IENj 位 (中断请求允许位) (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。

IRi.IR 标志不受 IERm.IENj 位的影响。即使 IERm.IENj 位为“0”, 在“11.2.1 中断请求寄存器 i (IRi) (i=中断向量号)”所示的条件下, IR 标志也会发生变化。

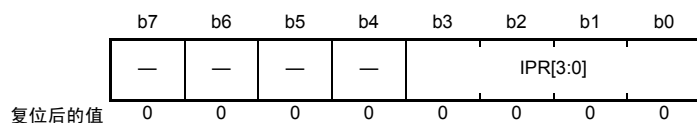
各中断源 (向量号) 都有 IERm.IENj 位。

中断源和 IERm.IENj 位的对应请参照“表 11.4 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“11.4.3 中断请求目标的选择”。

## 11.2.3 中断源优先级寄存器 m (IPRm) (m=00h ~ 90h)

地址 0008 7300h ~ 0008 7390h



位	符号	位名	功能	R/W																																																			
b3-b0	IPR[3:0]	中断优先级设定位	<table style="width: 100%; border: none;"> <tr> <td style="text-align: right;">b3</td> <td style="text-align: left;">b0</td> <td></td> </tr> <tr> <td>0 0 0 0</td> <td></td> <td>0 级 (禁止中断)</td> </tr> <tr> <td>0 0 0 1</td> <td></td> <td>1 级</td> </tr> <tr> <td>0 0 1 0</td> <td></td> <td>2 级</td> </tr> <tr> <td>0 0 1 1</td> <td></td> <td>3 级</td> </tr> <tr> <td>0 1 0 0</td> <td></td> <td>4 级</td> </tr> <tr> <td>0 1 0 1</td> <td></td> <td>5 级</td> </tr> <tr> <td>0 1 1 0</td> <td></td> <td>6 级</td> </tr> <tr> <td>0 1 1 1</td> <td></td> <td>7 级</td> </tr> <tr> <td>1 0 0 0</td> <td></td> <td>8 级</td> </tr> <tr> <td>1 0 0 1</td> <td></td> <td>9 级</td> </tr> <tr> <td>1 0 1 0</td> <td></td> <td>10 级</td> </tr> <tr> <td>1 0 1 1</td> <td></td> <td>11 级</td> </tr> <tr> <td>1 1 0 0</td> <td></td> <td>12 级</td> </tr> <tr> <td>1 1 0 1</td> <td></td> <td>13 级</td> </tr> <tr> <td>1 1 1 0</td> <td></td> <td>14 级</td> </tr> <tr> <td>1 1 1 1</td> <td></td> <td>15 级 (最高)</td> </tr> </table>	b3	b0		0 0 0 0		0 级 (禁止中断)	0 0 0 1		1 级	0 0 1 0		2 级	0 0 1 1		3 级	0 1 0 0		4 级	0 1 0 1		5 级	0 1 1 0		6 级	0 1 1 1		7 级	1 0 0 0		8 级	1 0 0 1		9 级	1 0 1 0		10 级	1 0 1 1		11 级	1 1 0 0		12 级	1 1 0 1		13 级	1 1 1 0		14 级	1 1 1 1		15 级 (最高)	R/W
b3	b0																																																						
0 0 0 0		0 级 (禁止中断)																																																					
0 0 0 1		1 级																																																					
0 0 1 0		2 级																																																					
0 0 1 1		3 级																																																					
0 1 0 0		4 级																																																					
0 1 0 1		5 级																																																					
0 1 1 0		6 级																																																					
0 1 1 1		7 级																																																					
1 0 0 0		8 级																																																					
1 0 0 1		9 级																																																					
1 0 1 0		10 级																																																					
1 0 1 1		11 级																																																					
1 1 0 0		12 级																																																					
1 1 0 1		13 级																																																					
1 1 1 0		14 级																																																					
1 1 1 1		15 级 (最高)																																																					
b7-b4	—	保留位	读写值都为“0”。	R/W																																																			

IPRm 寄存器是设定中断源优先级的寄存器。

各组中断源都有 IPRm 寄存器，m 为 00h ~ 90h 的连号。

中断源和中断组的对应请参照“表 11.4 中断向量表”。

## IPR[3:0] 位 (中断优先级设定位)

这些位是选择对应的中断源优先级的位。

由 IPR[3:0] 位选择的优先级只在判断向 CPU 请求中断的优先级时被参照，不影响 DTC 的传送请求。

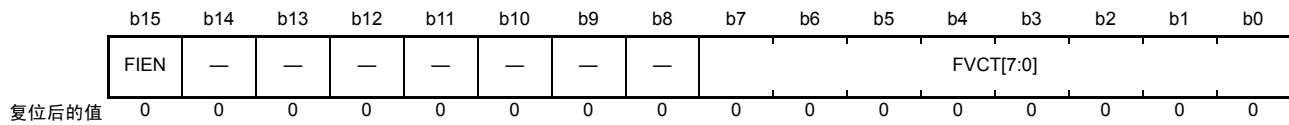
CPU 只接受优先级高于 PSW.IPL[3:0] 位所示优先级的中断请求并且进行中断处理。

如果同时产生多个中断请求，就通过 IPR[3:0] 位的设定值进行优先级的比较。如果同时产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 (IERm.IENj 位 =0) 中断请求的状态下写这些位。

### 11.2.4 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中断向量号。	R/W
b14-b8	—	保留位	读写值都为“0”。	R/W
b15	FIEN	高速中断允许位	0: 禁止高速中断 1: 允许高速中断	R/W

FIR 寄存器是设定高速中断功能的寄存器。

通过设定 FIR 寄存器实现的高速化功能只对 CPU 的中断请求有效，不影响 DTC 的传送请求。

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] 位设定的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 并且产生 FVCT[7:0] 位指定的向量号的中断请求，就作为高速中断向 CPU 输出中断请求，与 IPRm 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“11.6.3 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。

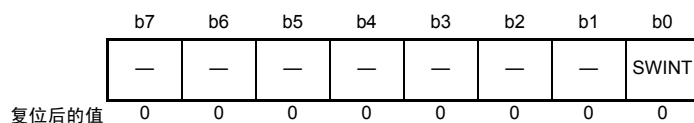
能设定的向量号请参照“表 11.4 中断向量表”。

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“10. 异常处理”和“11.4.5 高速中断”。

### 11.2.5 软件中断启动寄存器 (SWINTR)

地址 0008 72E0h



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”，读取值为“0”。

SWINTR 寄存器是产生软件中断请求的寄存器。

#### SWINT 位 (软件中断启动位)

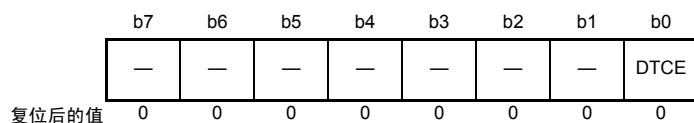
如果给 SWINT 位写“1”，中断请求寄存器 27 (IR27) 就变为“1”。

如果在将 DTC 启动允许寄存器 27 (DTCER27) 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 27 (DTCER27) 置“1”后给 SWINT 位写“1”，就发行 DTC 启动请求。

### 11.2.6 DTC 启动允许寄存器 n (DTCEr<sub>n</sub>) (n= 中断向量号)

地址 0008 711Bh ~ 0008 71FEh



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

DTCEr<sub>n</sub> 寄存器是选择要启动 DTC 的中断源的寄存器。

#### 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 ]

- 给 DTCE 位写“1”时

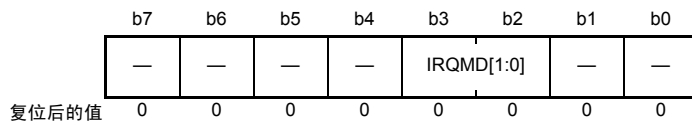
[ 为“0”的条件 ]

- 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)
- 给 DTCE 位写“0”时



## 11.2.7 IRQ 控制寄存器 n (IRQCRn) (n=0 ~ 7)

地址 0008 7500h ~ 0008 7507h



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

IRQCRn 寄存器是设定外部中断请求引脚 IRQ<sub>i</sub> (i=7 ~ 0) 的寄存器。

必须在对应的中断请求允许位为禁止中断请求 (IER<sub>m</sub>.IEN<sub>j</sub> 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<sub>i</sub>.IR 标志, 然后将中断请求允许位设定为允许。但是, 在更改为 Low 电平时不需要清除 IR 标志。

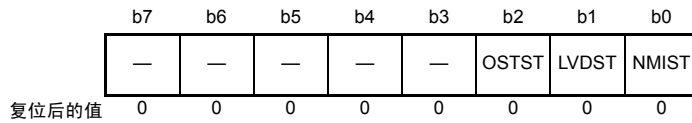
## IRQMD[1:0] 位 (IRQ 检测设定位)

这些位设定外部引脚中断源 (IRQ<sub>7</sub> ~ IRQ<sub>0</sub>) 的检测方法。

外部引脚中断的检测方法的设定请参照“11.4.6 外部引脚中断”。

### 11.2.8 非屏蔽中断状态寄存器 (NMISR)

地址 0008 7580h



位	符号	位名	功能	R/W
b0	NMIST	NMI 状态标志	0: 无 NMI 引脚中断请求 1: 有 NMI 引脚中断请求	R
b1	LVDST	电压监视中断状态标志	0: 无电压监视中断请求 1: 有电压监视中断请求	R
b2	OSTST	振荡停止检测中断状态标志	0: 无振荡停止检测中断请求 1: 有振荡停止检测中断请求	R
b7-b3	—	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 寄存器的全部位都为“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]

- 将 NMICR.NMIMD 位设定的边沿输入到 NMI 引脚时

[为“0”的条件]

- 给 NMICLR.NMICLR 位写“1”时

#### LVDST 标志 (电压监视中断状态标志)

此标志表示电压监视中断请求。

[为“1”的条件]

- 发生电压监视中断时

[为“0”的条件]

- 通过发生源清除中断时

#### 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]

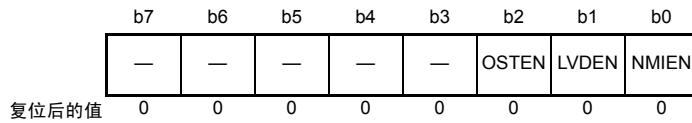
- 发生振荡停止检测中断时

[为“0”的条件]

- 给 NMICLR.OSTCLR 位写“1”时

### 11.2.9 非屏蔽中断允许寄存器 (NMIER)

地址 0008 7581h



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	LVDEN	电压监视中断允许位	0: 禁止电压监视中断 1: 允许电压监视中断	R/(W) (注 1)
b2	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)
b7-b3	—	保留位	读写值都为“0”。	R/W

注 1. 只能写 1 次“1”，以后的写操作无效。

NMIER 寄存器是允许使用非屏蔽中断的寄存器。

#### NMIEN 位 (NMI 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”。一旦允许了 NMI 引脚中断就无法禁止。

#### LVDEN 位 (电压监视中断允许位)

此位是允许使用电压监视中断的位。

只能写 1 次“1”，以后的写操作无效。

不能写“0”。一旦允许了电压监视中断就无法禁止。

#### 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”。一旦允许了振荡停止检测中断就无法禁止。

### 11.2.10 非屏蔽中断清除寄存器 (NMICLR)

地址 0008 7582h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	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”。如果写“1”，就将 NMISR.NMIST 标志置“0”。写“0”无效。	R/(W) (注1)
b1	—	保留位	读写值都为“0”。	R/W
b2	OSTCLR	OST 清除位	读取值为“0”。如果写“1”，就将 NMISR.OSTST 标志置“0”。写“0”无效。	R/(W) (注2)
b7-b3	—	保留位	读写值都为“0”。	R/W

注 1. 只能写用于清除 NMISR.NMIST 标志的“1”。

注 2. 只能写用于清除 NMISR.OSTST 标志的“1”。

NMICLR 寄存器是清除非屏蔽中断状态寄存器 (NMISR) 的寄存器。

#### NMICLR 位 (NMI 清除位)

如果写“1”，NMISR.NMIST 标志就变为“0”。  
不保持“1”的状态，读取值为“0”。

#### OSTCLR 位 (OST 清除位)

如果写“1”，NMISR.OSTST 标志就变为“0”。  
不保持“1”的状态，读取值为“0”。

### 11.2.11 NMI 引脚中断控制寄存器 (NMICR)

地址 0008 7583h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	NMIMD	—	—	—

复位后的值 0 0 0 0 0 0 0 0

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

NMICR 寄存器是设定 NMI 引脚中断的寄存器。

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

#### NMIMD 位 (NMI 检测设定位)

此位设定 NMI 引脚中断的检测方法。



表 11.4 中断向量表 (2/7)

优先级	中断请求发生源	名称	向量号	向量地址的偏移	中断源的检测方法	CPU 中断	DTC 启动	sstb 返回	sacs 返回	IER	IPR
高 ↑	—	保留	32	0080h	—	×	×	×	×	IER04.IEN0	—
		保留	33	0084h	—	×	×	×	×	IER04.IEN1	—
		保留	34	0088h	—	×	×	×	×	IER04.IEN2	—
		保留	35	008Ch	—	×	×	×	×	IER04.IEN3	—
		保留	36	0090h	—	×	×	×	×	IER04.IEN4	—
		保留	37	0094h	—	×	×	×	×	IER04.IEN5	—
		保留	38	0098h	—	×	×	×	×	IER04.IEN6	—
		保留	39	009Ch	—	×	×	×	×	IER04.IEN7	—
		保留	40	00A0h	—	×	×	×	×	IER05.IEN0	—
		保留	41	00A4h	—	×	×	×	×	IER05.IEN1	—
		保留	42	00A8h	—	×	×	×	×	IER05.IEN2	—
		保留	43	00ACh	—	×	×	×	×	IER05.IEN3	—
RSPI0	SPEI0	SPEI0	44	00B0h	电平	○	×	×	×	IER05.IEN4	IPR14
		SPRI0	45	00B4h	边沿	○	○	×	×	IER05.IEN5	
		SPTI0	46	00B8h	边沿	○	○	×	×	IER05.IEN6	
		SPII0	47	00BCh	电平	○	×	×	×	IER05.IEN7	
—	保留	保留	48	00C0h	—	×	×	×	×	IER06.IEN0	—
		保留	49	00C4h	—	×	×	×	×	IER06.IEN1	—
		保留	50	00C8h	—	×	×	×	×	IER06.IEN2	—
		保留	51	00CCh	—	×	×	×	×	IER06.IEN3	—
		保留	52	00D0h	—	×	×	×	×	IER06.IEN4	—
		保留	53	00D4h	—	×	×	×	×	IER06.IEN5	—
		保留	54	00D8h	—	×	×	×	×	IER06.IEN6	—
CAN0	ERS0	ERS0	56	00E0h	边沿	○	×	×	×	IER07.IEN0	IPR18
		RXF0	57	00E4h	边沿	○	×	×	×	IER07.IEN1	
		TXF0	58	00E8h	边沿	○	×	×	×	IER07.IEN2	
		RXM0	59	00ECh	边沿	○	×	×	×	IER07.IEN3	
		TXM0	60	00F0h	边沿	○	×	×	×	IER07.IEN4	
—	保留	保留	61	00F4h	—	×	×	×	×	IER07.IEN5	—
		保留	62	00F8h	—	×	×	×	×	IER07.IEN6	—
		保留	63	00FCh	—	×	×	×	×	IER07.IEN7	—
低											

表 11.4 中断向量表 (3/7)

优先级	中断请求发生源	名称	向量号	向量地址的偏移	中断源的检测方法	CPU 中断	DTC 启动	sstb 返回	sacs 返回	IER	IPR
高 ↑	外部引脚	IRQ0	64	0100h	边沿 / 电平	○	○	○	○	IER08.IEN0	IPR20
		IRQ1	65	0104h	边沿 / 电平	○	○	○	○	IER08.IEN1	IPR21
		IRQ2	66	0108h	边沿 / 电平	○	○	○	○	IER08.IEN2	IPR22
		IRQ3	67	010Ch	边沿 / 电平	○	○	○	○	IER08.IEN3	IPR23
		IRQ4	68	0110h	边沿 / 电平	○	○	○	○	IER08.IEN4	IPR24
		IRQ5	69	0114h	边沿 / 电平	○	○	○	○	IER08.IEN5	IPR25
		IRQ6	70	0118h	边沿 / 电平	○	○	○	○	IER08.IEN6	IPR26
		IRQ7	71	011Ch	边沿 / 电平	○	○	○	○	IER08.IEN7	IPR27
		保留	72	0120h	—	×	×	×	×	IER09.IEN0	—
		保留	73	0124h	—	×	×	×	×	IER09.IEN1	—
		保留	74	0128h	—	×	×	×	×	IER09.IEN2	—
		保留	75	012Ch	—	×	×	×	×	IER09.IEN3	—
		保留	76	0130h	—	×	×	×	×	IER09.IEN4	—
		保留	77	0134h	—	×	×	×	×	IER09.IEN5	—
		保留	78	0138h	—	×	×	×	×	IER09.IEN6	—
		保留	79	013Ch	—	×	×	×	×	IER09.IEN7	—
		保留	80	0140h	—	×	×	×	×	IER0A.IEN0	—
		保留	81	0144h	—	×	×	×	×	IER0A.IEN1	—
		保留	82	0148h	—	×	×	×	×	IER0A.IEN2	—
		保留	83	014Ch	—	×	×	×	×	IER0A.IEN3	—
	保留	84	0150h	—	×	×	×	×	IER0A.IEN4	—	
	保留	85	0154h	—	×	×	×	×	IER0A.IEN5	—	
	保留	86	0158h	—	×	×	×	×	IER0A.IEN6	—	
	保留	87	015Ch	—	×	×	×	×	IER0A.IEN7	—	
	—	保留	88	0160h	—	×	×	×	×	IER0B.IEN0	—
	—	保留	89	0164h	—	×	×	×	×	IER0B.IEN1	—
	—	保留	90	0168h	—	×	×	×	×	IER0B.IEN2	—
	—	保留	91	016Ch	—	×	×	×	×	IER0B.IEN3	—
	—	保留	92	0170h	—	×	×	×	×	IER0B.IEN4	—
	—	保留	93	0174h	—	×	×	×	×	IER0B.IEN5	—
	—	保留	94	0178h	—	×	×	×	×	IER0B.IEN6	—
	—	保留	95	017Ch	—	×	×	×	×	IER0B.IEN7	—
—	WDT	WOVI	96	0180h	边沿	○	—	—	○	IER0C.IEN0	IPR40
—	保留		97	0184h	—	×	×	×	×	IER0C.IEN1	—
—	ADA0	ADI0	98	0188h	边沿	○	○	—	—	IER0C.IEN2	IPR44
—	保留		99	018Ch	—	×	×	×	×	IER0C.IEN3	—
—	保留		100	0190h	—	×	×	×	×	IER0C.IEN4	—
—	保留		101	0194h	—	×	×	×	×	IER0C.IEN5	—
—	S12ADA0	S12ADI0	102	0198h	边沿	○	○	×	×	IER0C.IEN6	IPR48
—	S12ADA1	S12ADI1	103	019Ch	边沿	○	○	×	×	IER0C.IEN7	
—	保留		104	01A0h	—	×	×	×	×	IER0D.IEN0	—
—	保留		105	01A4h	—	×	×	×	×	IER0D.IEN1	—
低	比较器	CMP1	106	01A8h	边沿	○	○	×	×	IER0D.IEN2	IPR49

表 11.4 中断向量表 (4/7)

优先级	中断请求发生源	名称	向量号	向量地址的偏移	中断源的检测方法	CPU 中断	DTC 启动	sstb 返回	sacs 返回	IER	IPR
高 ↑	—	保留	107	01ACh	—	×	×	×	×	IER0D.IEN3	—
		保留	108	01B0h	—	×	×	×	×	IER0D.IEN4	—
		保留	109	01B4h	—	×	×	×	×	IER0D.IEN5	—
		保留	110	01B8h	—	×	×	×	×	IER0D.IEN6	—
		保留	111	01BCh	—	×	×	×	×	IER0D.IEN7	—
		保留	112	01C0h	—	×	×	×	×	IER0E.IEN0	—
		保留	113	01C4h	—	×	×	×	×	IER0E.IEN1	—
	MTU0	TGIA0	114	01C8h	边沿	○	○	×	×	IER0E.IEN2	IPR51
		TGIB0	115	01CCh	边沿	○	○	×	×	IER0E.IEN3	
		TGIC0	116	01D0h	边沿	○	○	×	×	IER0E.IEN4	
		TGID0	117	01D4h	边沿	○	○	×	×	IER0E.IEN5	IPR52
		TCIV0	118	01D8h	边沿	○	×	×	×	IER0E.IEN6	
		TGIE0	119	01DCh	边沿	○	×	×	×	IER0E.IEN7	
		TGIF0	120	01E0h	边沿	○	×	×	×	IER0F.IEN0	
MTU1	TGIA1	121	01E4h	边沿	○	○	×	×	IER0F.IEN1	IPR53	
	TGIB1	122	01E8h	边沿	○	○	×	×	IER0F.IEN2		
	TCIV1	123	01ECh	边沿	○	×	×	×	IER0F.IEN3	IPR54	
	TCIU1	124	01F0h	边沿	○	×	×	×	IER0F.IEN4		
MTU2	TGIA2	125	01F4h	边沿	○	○	×	×	IER0F.IEN5	IPR55	
	TGIB2	126	01F8h	边沿	○	○	×	×	IER0F.IEN6		
	TCIV2	127	01FCh	边沿	○	×	×	×	IER0F.IEN7	IPR56	
	TCIU2	128	0200h	边沿	○	×	×	×	IER10.IEN0		
MTU3	TGIA3	129	0204h	边沿	○	○	×	×	IER10.IEN1	IPR57	
	TGIB3	130	0208h	边沿	○	○	×	×	IER10.IEN2		
	TGIC3	131	020Ch	边沿	○	○	×	×	IER10.IEN3		
	TGID3	132	0210h	边沿	○	○	×	×	IER10.IEN4		
	TCIV3	133	0214h	边沿	○	×	×	×	IER10.IEN5	IPR58	
MTU4	TGIA4	134	0218h	边沿	○	○	×	×	IER10.IEN6	IPR59	
	TGIB4	135	021Ch	边沿	○	○	×	×	IER10.IEN7		
	TGIC4	136	0220h	边沿	○	○	×	×	IER11.IEN0		
	TGID4	137	0224h	边沿	○	○	×	×	IER11.IEN1		
	TCIV4	138	0228h	边沿	○	○	×	×	IER11.IEN2	IPR5A	
MTU5	TGIU5	139	022Ch	边沿	○	○	×	×	IER11.IEN3	IPR5B	
	TGIV5	140	0230h	边沿	○	○	×	×	IER11.IEN4		
	TGIW5	141	0234h	边沿	○	○	×	×	IER11.IEN5		
MTU6	TGIA6	142	0238h	边沿	○	○	×	×	IER11.IEN6	IPR5C	
	TGIB6	143	023Ch	边沿	○	○	×	×	IER11.IEN7		
	TGIC6	144	0240h	边沿	○	○	×	×	IER12.IEN0		
	TGID6	145	0244h	边沿	○	○	×	×	IER12.IEN1		
	TCIV6	146	0248h	边沿	○	×	×	×	IER12.IEN2	IPR5D	
	保留	147	024Ch	—	×	×	—	—	IER12.IEN3	—	
	保留	148	0250h	—	×	×	—	—	IER12.IEN4	—	
低											



表 11.4 中断向量表 (5/7)

优先级	中断请求发生源	名称	向量号	向量地址的偏移	中断源的检测方法	CPU 中断	DTC 启动	sstb 返回	sacs 返回	IER	IPR	
高 ↑	MTU7	TGIA7	149	0254h	边沿	○	○	×	×	IER12.IEN5	IPR5E	
		TGIB7	150	0258h	边沿	○	○	×	×	IER12.IEN6		
		TGIC7	151	025Ch	边沿	○	○	×	×	IER12.IEN7	IPR5F	
		TGID7	152	0260h	边沿	○	○	×	×	IER13.IEN0		
		TCIV7	153	0264h	边沿	○	○	×	×	IER13.IEN1	IPR60	
	—	保留	154	0268h	—	—	×	×	×	×	IER13.IEN2	—
		保留	155	026Ch	—	—	×	×	×	×	IER13.IEN3	—
		保留	156	0270h	—	—	×	×	×	×	IER13.IEN4	—
		保留	157	0274h	—	—	×	×	×	×	IER13.IEN5	—
		保留	158	0278h	—	—	×	×	×	×	IER13.IEN6	—
		保留	159	027Ch	—	—	×	×	×	×	IER13.IEN7	—
		保留	160	0280h	—	—	×	×	×	×	IER14.IEN0	—
		保留	161	0284h	—	—	×	×	×	×	IER14.IEN1	—
		保留	162	0288h	—	—	×	×	×	×	IER14.IEN2	—
		保留	163	028Ch	—	—	×	×	×	×	IER14.IEN3	—
	—	保留	164	0290h	—	—	×	×	×	×	IER14.IEN4	—
		保留	165	0294h	—	—	×	×	×	×	IER14.IEN5	—
	—	保留	166	0298h	—	—	×	×	×	×	IER14.IEN6	—
		保留	167	029Ch	—	—	×	×	×	×	IER14.IEN7	—
		保留	168	02A0h	—	—	×	×	×	×	IER15.IEN0	
	保留	169	02A4h	—	—	×	×	×	×	IER15.IEN1		
	POE	OEI1	170	02A8h	电平	电平	○	×	×	×	IER15.IEN2	IPR67
		OEI2	171	02ACh	电平	电平	○	×	×	×	IER15.IEN3	
		OEI3	172	02B0h	电平	电平	○	×	×	×	IER15.IEN4	
		OEI4	173	02B4h	电平	电平	○	×	×	×	IER15.IEN5	
	GPT0	GTCIA0	174	02B8h	边沿	边沿	○	○	×	×	IER15.IEN6	IPR68
		GTCIB0	175	02BCh	边沿	边沿	○	○	×	×	IER15.IEN7	
		GTCIC0	176	02C0h	边沿	边沿	○	○	×	×	IER16.IEN0	
		GTCIE0	177	02C4h	边沿	边沿	○	○	×	×	IER16.IEN1	IPR69
		GTCIV0	178	02C8h	边沿	边沿	○	○	×	×	IER16.IEN2	
		LOCO1	179	02CCh	边沿	边沿	○	○	×	×	IER16.IEN3	
GPT1	GTCIA1	180	02D0h	边沿	边沿	○	○	×	×	IER16.IEN4	IPR6A	
	GTCIB1	181	02D4h	边沿	边沿	○	○	×	×	IER16.IEN5		
	GTCIC1	182	02D8h	边沿	边沿	○	○	×	×	IER16.IEN6		
	GTCIE1	183	02DCh	边沿	边沿	○	○	×	×	IER16.IEN7	IPR6B	
	GTCIV1	184	02E0h	边沿	边沿	○	○	×	×	IER17.IEN0		
	保留	185	02E4h	—	—	×	×	×	×	IER17.IEN1		—
GPT2	GTCIA2	186	02E8h	边沿	边沿	○	○	×	×	IER17.IEN2	IPR6C	
	GTCIB2	187	02ECh	边沿	边沿	○	○	×	×	IER17.IEN3		
	GTCIC2	188	02F0h	边沿	边沿	○	○	×	×	IER17.IEN4		
	GTCIE2	189	02F4h	边沿	边沿	○	○	×	×	IER17.IEN5	IPR6D	
	GTCIV2	190	02F8h	边沿	边沿	○	○	×	×	IER17.IEN6		
	保留	191	02FCh	—	—	×	×	×	×	IER17.IEN7		—
低												

表 11.4 中断向量表 (6 / 7)

优先级	中断请求发生源	名称	向量号	向量地址的偏移	中断源的检测方法	CPU 中断	DTC 启动	sstb 返回	sacs 返回	IER	IPR
↑ 高	GPT3	GTCIA3	192	0300h	边沿	○	○	×	×	IER18.IEN0	IPR6E
		GTCIB3	193	0304h	边沿	○	○	×	×	IER18.IEN1	
		GTCIC3	194	0308h	边沿	○	○	×	×	IER18.IEN2	
		GTCIE3	195	030Ch	边沿	○	○	×	×	IER18.IEN3	IPR6F
		GTCIV3	196	0310h	边沿	○	○	×	×	IER18.IEN4	
		保留	197	0314h	—	×	×	×	×	IER18.IEN5	—
	—	保留	198	0318h	—	×	×	×	×	IER18.IEN6	—
		保留	199	031Ch	—	×	×	×	×	IER18.IEN7	—
		保留	200	0320h	—	×	×	×	×	IER19.IEN0	—
		保留	201	0324h	—	×	×	×	×	IER19.IEN1	—
		保留	202	0328h	—	×	×	×	×	IER19.IEN2	—
		保留	203	032Ch	—	×	×	×	×	IER19.IEN3	—
		保留	204	0330h	—	×	×	×	×	IER19.IEN4	—
		保留	205	0334h	—	×	×	×	×	IER19.IEN5	—
		保留	206	0338h	—	×	×	×	×	IER19.IEN6	—
		保留	207	033Ch	—	×	×	×	×	IER19.IEN7	—
		保留	208	0340h	—	×	×	×	×	IER1A.IEN0	—
		保留	209	0344h	—	×	×	×	×	IER1A.IEN1	—
	—	保留	210	0348h	—	×	×	×	×	IER1A.IEN2	—
		保留	211	034Ch	—	×	×	×	×	IER1A.IEN3	—
		保留	212	0350h	—	×	×	×	×	IER1A.IEN4	—
		保留	213	0354h	—	×	×	×	×	IER1A.IEN5	—
	SCI0	ERI0	214	0358h	电平	○	×	×	×	IER1A.IEN6	IPR80
		RXI0	215	035Ch	边沿	○	○	×	×	IER1A.IEN7	
		TXI0	216	0360h	边沿	○	○	×	×	IER1B.IEN0	
TEI0		217	0364h	电平	○	×	×	×	IER1B.IEN1		
SCI1	ERI1	218	0368h	电平	○	×	×	×	IER1B.IEN2	IPR81	
	RXI1	219	036Ch	边沿	○	○	×	×	IER1B.IEN3		
	TXI1	220	0370h	边沿	○	○	×	×	IER1B.IEN4		
	TEI1	221	0374h	电平	○	×	×	×	IER1B.IEN5		
SCI2	ERI2	222	0378h	电平	○	×	×	×	IER1B.IEN6	IPR82	
	RXI2	223	037Ch	边沿	○	○	×	×	IER1B.IEN7		
	TXI2	224	0380h	边沿	○	○	×	×	IER1C.IEN0		
	TEI2	225	0384h	电平	○	×	×	×	IER1C.IEN1		
低											

表 11.4 中断向量表 (7/7)

优先级	中断请求发生源	名称	向量号	向量地址的偏移	中断源的检测方法	CPU 中断	DTC 启动	sstb 返回	sacs 返回	IER	IPR
高 ↑	—	保留	226	0388h	—	×	×	×	×	IER1C.IEN2	—
		保留	227	038Ch	—	×	×	×	×	IER1C.IEN3	—
		保留	228	0390h	—	×	×	×	×	IER1C.IEN4	—
		保留	229	0394h	—	×	×	×	×	IER1C.IEN5	—
		保留	230	0398h	—	×	×	×	×	IER1C.IEN6	—
		保留	231	039Ch	—	×	×	×	×	IER1C.IEN7	—
		保留	232	03A0h	—	×	×	×	×	IER1D.IEN0	—
		保留	233	03A4h	—	×	×	×	×	IER1D.IEN1	—
		保留	234	03A8h	—	×	×	×	×	IER1D.IEN2	—
		保留	235	03ACh	—	×	×	×	×	IER1D.IEN3	—
		保留	236	03B0h	—	×	×	×	×	IER1D.IEN4	—
		保留	237	03B4h	—	×	×	×	×	IER1D.IEN5	—
		保留	238	03B8h	—	×	×	×	×	IER1D.IEN6	—
		保留	239	03BCh	—	×	×	×	×	IER1D.IEN7	—
		保留	240	03C0h	—	×	×	×	×	IER1E.IEN0	—
		保留	241	03C4h	—	×	×	×	×	IER1E.IEN1	—
		保留	242	03C8h	—	×	×	×	×	IER1E.IEN2	—
		保留	243	03CCh	—	×	×	×	×	IER1E.IEN3	—
		保留	244	03D0h	—	×	×	×	×	IER1E.IEN4	—
		保留	245	03D4h	—	×	×	×	×	IER1E.IEN5	—
RIIC0	—	ICEE10	246	03D8h	电平	○	×	×	×	IER1E.IEN6	IPR88
		ICRX10	247	03DCh	边沿	○	○	×	×	IER1E.IEN7	IPR89
		ICTX10	248	03E0h	边沿	○	○	×	×	IER1F.IEN0	IPR8A
		ICTE10	249	03E4h	电平	○	×	×	×	IER1F.IEN1	IPR8B
—	—	保留	250	03E8h	—	×	×	×	×	IER1F.IEN2	—
		保留	251	03ECh	—	×	×	×	×	IER1F.IEN3	—
		保留	252	03F0h	—	×	×	×	×	IER1F.IEN4	—
		保留	253	03F4h	—	×	×	×	×	IER1F.IEN5	—
LIN0	—	LIN0	254	03F8h	边沿	○	×	×	×	IER1F.IEN6	IPR90
低	—	保留	255	03FCh	—	×	×	×	×	IER1F.IEN7	—

○：能选择 x：不能选择

### 11.3.2 高速中断的向量表

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

### 11.3.3 非屏蔽中断的向量表

非屏蔽中断的向量表为“FFFF FFF8h”。

## 11.4 中断的运行说明

中断控制器进行以下的处理:

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

### 11.4.1 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  $IRQ_i$  引脚 ( $i=0 \sim 7$ ) 的外部中断请求, 能通过设定  $IRQCR_i.IRQMD[1:0]$  位, 转换边沿检测和电平检测。

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 11.4 中断向量表”。

#### 11.4.1.1 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  $IR_i.IR$  标志变化如图 11.2 所示。

在产生中断请求时的中断信号的变化位置,  $IR_i.IR$  标志变为“1”。当中断请求目标为 CPU 时, 如果接受中断,  $IR_i.IR$  标志就自动变为“0”。当中断请求目标为 DTC 时,  $IR$  标志的清除时序因 DTC 的传送设定和传送次数而不同。详细内容请参照表 11.5。不需要通过软件清除  $IR_i.IR$  标志。

关于中断向量为 64 ~ 71 的中断, 这些中断的中断信号产生时序与其他中断不同。在中断向量为 64 ~ 71 的  $IRQ$  引脚中断时, 从输入  $IRQ$  引脚开始的内部延迟增加 2 个  $PCLK$ 。

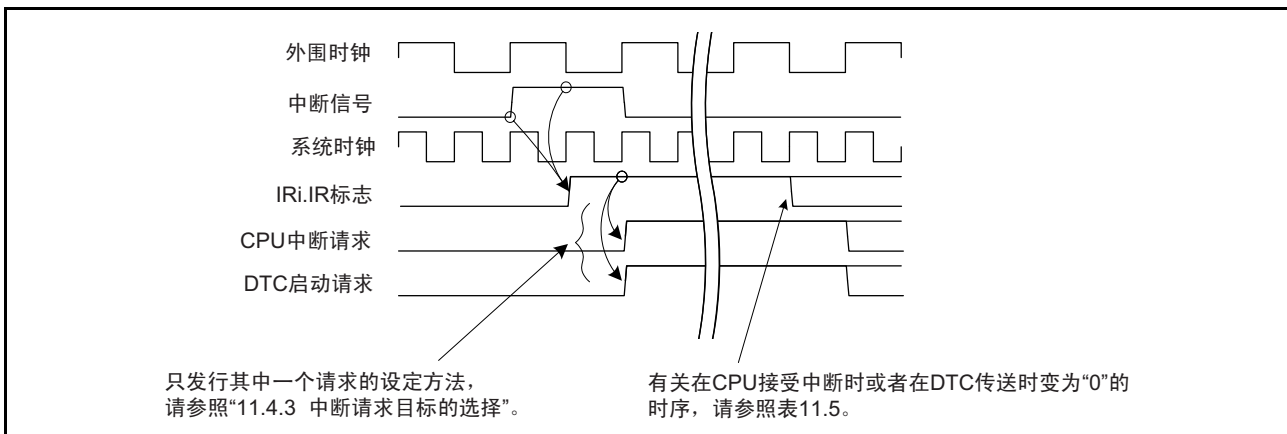


图 11.2 边沿检测时的  $IR_i.IR$  标志变化

在产生中断请求并且  $IR_i.IR$  标志为“1”的状态下, 忽视再次发生的中断请求。另外, GPT、MTU3、LIN 的全部中断以及 SI2ADA 的 CMPI 中断除了具有  $IR_i.IR$  标志外, 还具有用于判断多个中断源且不会自动变为“0”的状态标志。因此, 在此标志为“1”的状态下, 忽视再次发生的中断请求。如果要接受中断, 就必须将中断发生源的标志置“0”。有关目标状态标志的详细内容, 请参照各外围模块的章节。

如果在  $IR_i.IR$  标志变为“0”后产生中断请求,  $IR_i.IR$  标志就被重新置位。 $IR_i.IR$  标志被重新置位的时序如图 11.3 所示。

在通信功能 (SCI/RIIC/RSPI) 和 DTC 功能的组合情况下, 有可能忽视中断请求并且传送请求消失, 详细内容请参照“11.7 使用时的注意事项”。

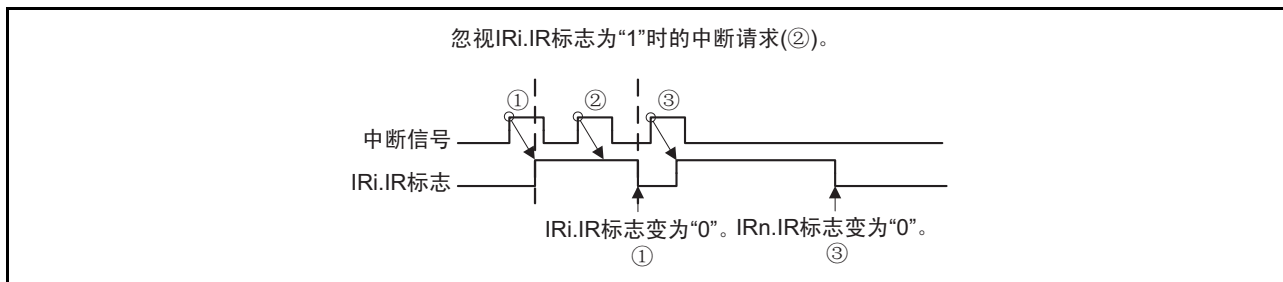


图 11.3 IRi.IR 标志被重新置位的时序

即使在 IRi.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），IRi.IR 标志也不受影响而保持“1”的状态。禁止中断时的 IRi.IR 标志变化如图 11.4 所示。

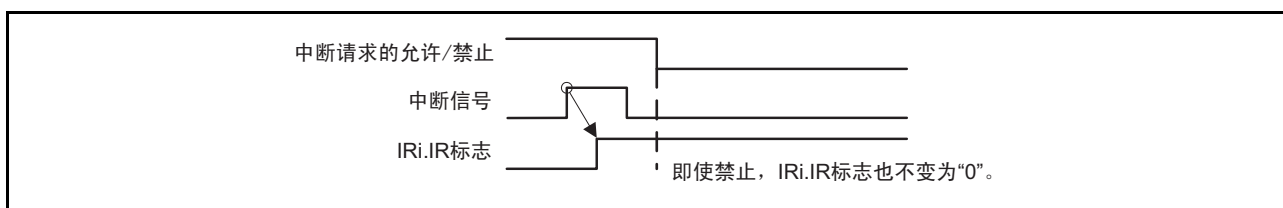


图 11.4 中断请求的禁止和IRi.IR标志的关系

### 11.4.1.2 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 IRi.IR 标志变化如图 11.5 所示。

在中断信号的有效期间，IRi.IR 标志保持“1”。要将 IRi.IR 标志置“0”时，必须将中断发生源的中断请求标志置“0”。

必须在确认中断发生源的中断请求标志变为“0”后结束中断处理程序。电平检测中断的处理步骤如图 11.6 所示。

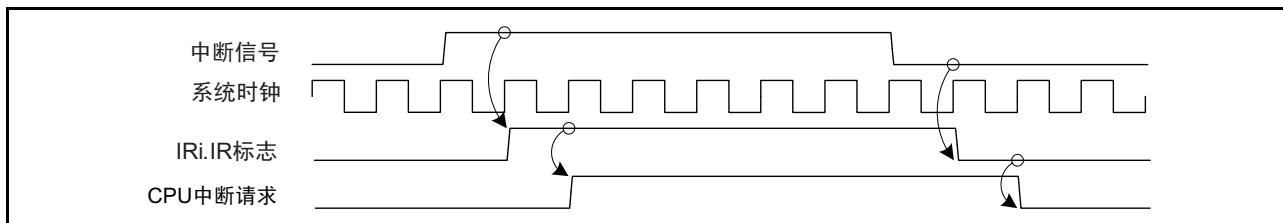


图 11.5 电平检测时的 IRi.IR 标志变化

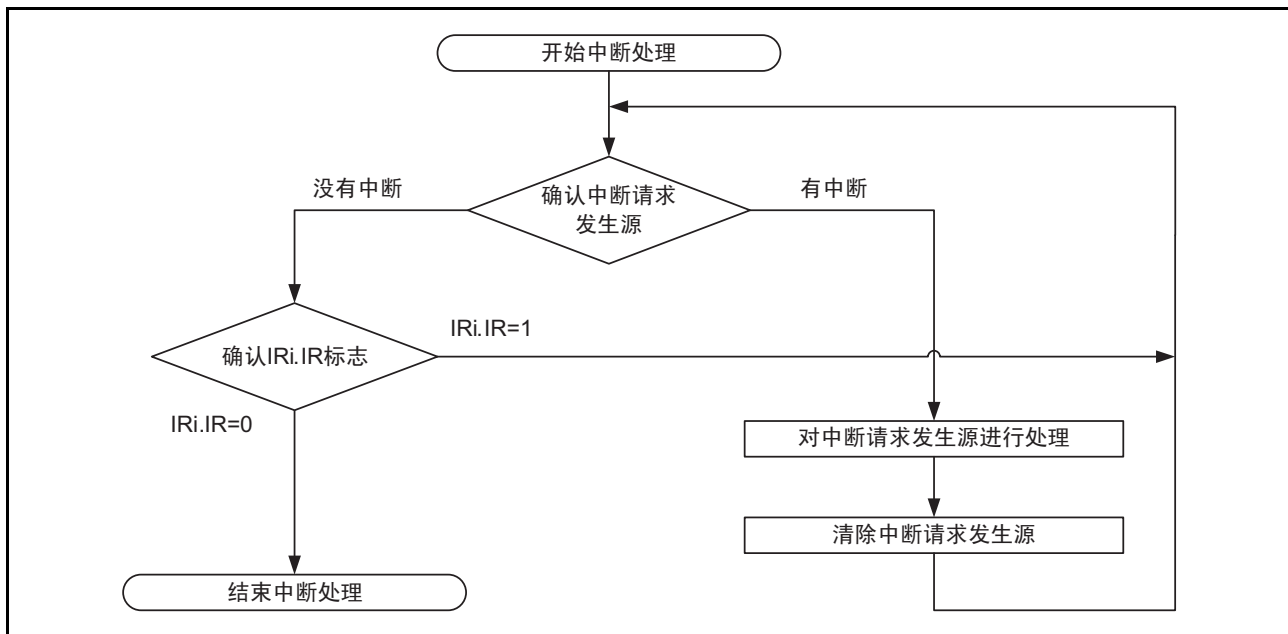


图 11.6 电平检测中断的处理步骤

### 11.4.2 中断请求的允许 / 禁止

要允许中断请求时，需要进行以下的设定：

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 IERm.IENj 位允许中断请求。

如果在中断发生源产生被允许输出的中断请求，对应的 IRi.IR 标志就变为“1”。

通过 IERm.IENj 位允许中断请求，将 IRi.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 IRi.IR 标志为“1”的中断请求。

IRi.IR 标志不受 IERm.IENj 位的影响。

### 11.4.3 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 11.4 中断向量表”所示。不能选择表 11.4 中没有“○”标记的中断请求目标。

如果通过 IRQ 引脚将 DTC 设定为中断请求目标，就必须将 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) DTC 的启动

必须在 IERm.IENj 位为“0”时进行以下的设定：

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”。
2. 将 DTC 模块启动位 (DTCST.DTCST) 位置“1”。

必须在进行了上述设定的状态下将 IERm.IENj 位置“1”。

DTC 的设定步骤请参照“14.5 DTC 的设定步骤”。

## (2) CPU 中断请求

中断请求目标不是 DTC 的中断源是 CPU 中断的对象。  
必须在未设定启动 DTC 的状态下将 IERm.IENj 位置“1”。

将 DTC 设定为中断请求目标时的运行如表 11.5 所示。

表 11.5 启动 DTC 时的运行

中断请求目标	DTC.MRB.DISEL	剩余的传送次数	每 1 次请求的运行	IRi.IR (注 1)(注 3)	传送后的中断请求目标
DTC (注 2)	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 DTCERn.DTCE 位并转换到 CPU。
	0	≠0	DTC 传送	在开始传送读取 DTC 传送息后的 DTC 数据时清除。	DTC
		=0	DTC 传送 → CPU 中断	在接受 CPU 中断时清除。	清除 DTCERn.DTCE 位并转换到 CPU。

注 1. 忽视在 IRi.IR 标志为“1”的状态下再次产生的中断请求 (DTC 启动请求)。

注 2. 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、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。

注 3. 如果忽视在 IRi.IR 标志为“1”的状态下再次产生的中断请求 (DTC 启动请求), 通信功能 (SCI/RIIC/RSPI) 就会发生问题。此时, 必须在不发生传送请求消失的状态下使用, 或者实施 DTC 的软件回避措施。详细内容请参照“11.7 使用时的注意事项”。

必须在 IERm.IENj 位为“0”时更改中断请求目标。

如果在设定 (1) 的 DTC 的启动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 传送的设定内容, 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 (1) DTC 的启动。

### 11.4.4 优先级的判断

中断控制器判断各中断请求目标的优先级。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为最优先, 其次优先中断优先级设定位 (IPRm.IPR[3:0]) 中值大的中断源。如果 IPRm.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 IPRm.IPR[3:0] 位的影响, 优先向量号小的中断源。

### 11.4.5 高速中断

高速中断功能是高速地进行 CPU 中断处理的功能，只对 CPU 的中断请求有效，不影响 DTC 的传送请求。

在通过 FIR.FVCT[7:0] 位给要设定为高速中断的中断源设定向量号并且通过 FIR.FIEN 位允许高速中断的状态下，如果发生对应的中断源，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 IPRm.IPR[3:0] 位的设定无关，是优先级最高的中断源。

高速中断的详细内容请参照“10. 异常处理”。

### 11.4.6 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”。
2. 设定 I/O 端口并进行确认。
3. 通过 IRQCRi.IRQMD[1:0] 位设定检测方法。
4. 将 IRi.IR 标志置“0”（边沿检测的情况）。
5. 当启动 DTC 时，设定 DTCERn.DTCE 位（在不设定的情况下，为 CPU 中断）。
6. 将 IERm.IENj 位置“1”。

## 11.5 非屏蔽中断的运行说明

非屏蔽中断有 NMI 引脚中断、电压监视中断和振荡停止检测中断三种。非屏蔽中断只限于向 CPU 请求的中断，不能启动 DTC。在包括高速中断在内的所有中断中，非屏蔽中断是最优先的中断。

与 CPU 的 PSW.I 位（中断允许位）、PSW.IPL[3:0] 位（处理器中断优先级）的状态无关，总是接受非屏蔽中断请求。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 寄存器的全部位是否为“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，设定 NMI 引脚的检测方法（NMICR.NMIMD 位）。
3. 在使用 NMI 引脚时，在给 NMICLR.NMICLR 位写“1”后将 NMISR.NMIST 标志置“0”。
4. 通过给非屏蔽中断允许寄存器（NMIER）允许的中断对应位置写“1”，允许使用非屏蔽中断。

如果给 NMIER 寄存器写“1”，就忽视以后对 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“10. 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”，清除振荡停止检测中断状态标志（NMISR.OSTST）。

有关电压监视中断状态标志（NMISR.LVDST）的清除，请参照“7. 电压检测电路（LVD）”。

不能在 NMIST 寄存器的任意状态标志为“1”的状态下发行 WAIT 指令。



## 11.6 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、软件待机模式状态返回的中断源如“表 11.4 中断向量表”所示。

详细内容请参照“9. 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。

### 11.6.1 从睡眠模式的返回

能通过非屏蔽中断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。
- 非屏蔽中断

已经通过 NMIER 寄存器允许对应的中断请求。

### 11.6.2 从全模块时钟停止模式的返回

能通过非屏蔽中断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。

### 11.6.3 从软件待机模式的返回

能通过非屏蔽中断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。

(对于设定为高速中断的中断源，除了高速中断设定寄存器 (FIR) 以外，还必须将对应的中断优先级 (IPRm) 设定为高于 CPU.PSW.IPL[3:0]。)

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。

## 11.7 使用时的注意事项

### 11.7.1 使用 DTC 传送的通信运行注意事项

RX62T 群在使用 DTC 传送的通信功能运行中需要注意以下事项。在产生中断请求并且 IRi.IR 标志为“1”的状态下，忽视再次产生的中断请求。在中断请求目标为 DTC 时，IR 标志清除时序因 DTC 的传送设定和传送次数而不同，因此根据设定的组合，传送请求有可能消失。

在组合使用通信功能 (SCI/RIIC/RSPI) 和 DTC 功能时，请按照以下注意事项进行使用。

#### (1) 使用 DTC 传送的通信运行的传送请求消失条件

需要注意的 DTC 功能的组合如表 11.6 所示。

表 11.6 需要注意的 DTC 功能的组合

	链传送的使用	每次传送中 CPU 通信中断的有无 (注1) (DISEL 的设定)	发生问题的可能性
DTC	不使用链传送	DISEL=0 【传送计数器 > 0】	无
		DISEL=0 【传送计数器 =0】(最后的传送)	无 (注2)
		DISEL=1	有
	使用链传送	DISEL=0 (传送计数器 > 0 并且链传送的最后为通信寄存器的存取)	无
		DISEL=0 (传送计数器 =0 (最后的传送) 并且链传送的最后为通信寄存器的存取)	无 (注2)
		DISEL=1	有

注 1. 通信中断是指 SCI/RIIC/RSPI 的发送数据空中断和接收数据满中断。

注 2. 必须注意：对于下一个通信包的传送请求，如果来不及清除 IRi.IR 标志，就可能发生问题。

#### (2) 在 DISEL 为“1”时使用 DTC 的情况

在数据传送 (DTC 运行) 后接受 CPU 中断时，自动清除 IR 标志。如果在 IR 标志被自动清除前发生下一个传送请求，传送请求就会消失。因为在 DISEL 为“1”时一定会发生 CPU 中断，所以必须实施下述的软件回避措施 (3)。

(3) 软件回避措施 (SCI、RIIC、RSPI) 流程图

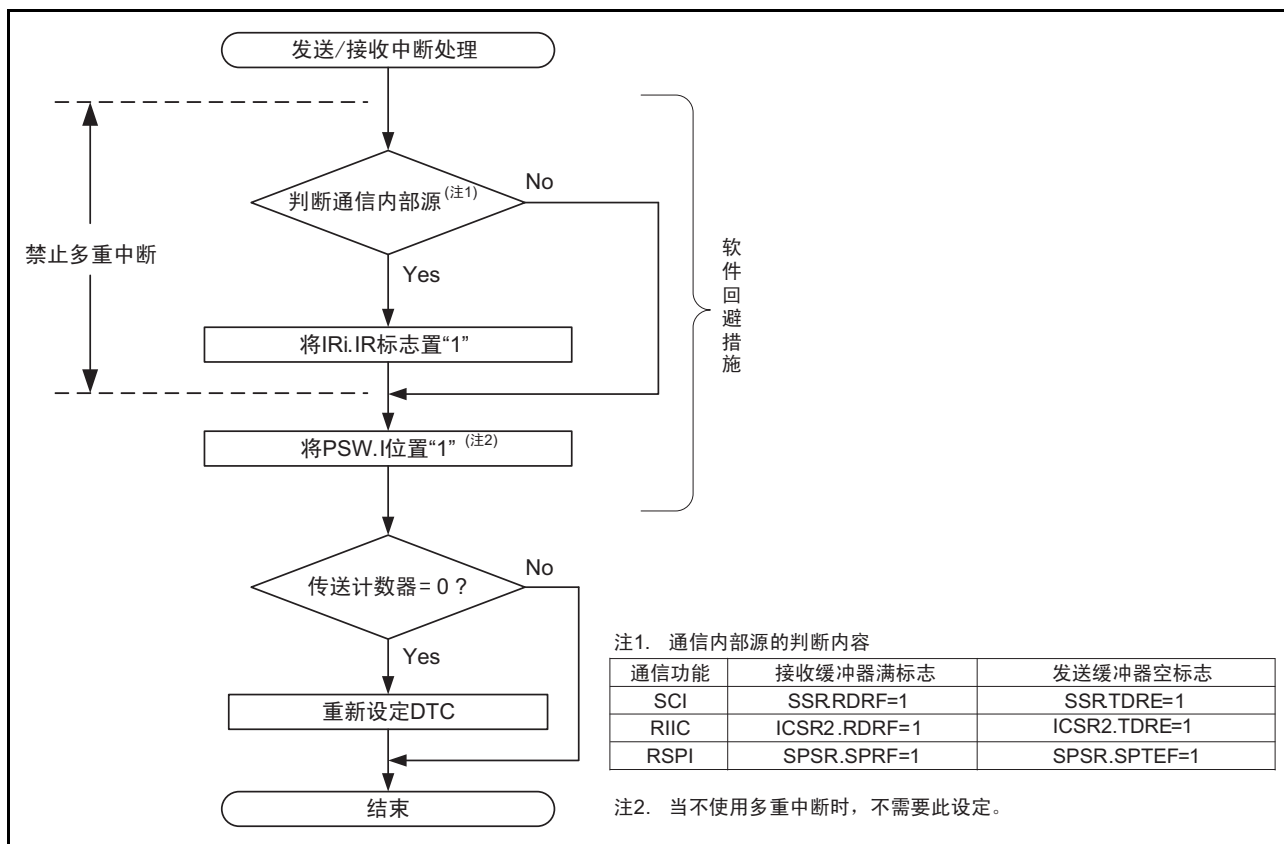


图 11.7 软件回避措施 (SCI、RIIC、RSPI) 流程图

## 12. 总线

### 12.1 概要

总线的结构图如图 12.1 所示，各总线的规格和各总线种类的地址对应表分别如表 12.1 和表 12.2 所示。

表 12.1 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"> <li>• 连接 CPU（指令）。</li> <li>• 连接内部存储器（内部 RAM 和内部 ROM）。</li> <li>• 与系统时钟（ICLK）同步运行。</li> </ul>
	操作数总线	<ul style="list-style-type: none"> <li>• 连接 CPU（操作数）。</li> <li>• 连接内部存储器（内部 RAM 和内部 ROM）。</li> <li>• 与系统时钟（ICLK）同步运行。</li> </ul>
存储器总线	存储器总线 1	• 连接内部 RAM。
	存储器总线 2	• 连接内部 ROM。
内部主总线	内部主总线 1	<ul style="list-style-type: none"> <li>• 连接 CPU。</li> <li>• 与系统时钟（ICLK）同步运行。</li> </ul>
	内部主总线 2	<ul style="list-style-type: none"> <li>• 连接 DTC。</li> <li>• 连接内部存储器（内部 RAM 和内部 ROM）。</li> <li>• 与系统时钟（ICLK）同步运行。</li> </ul>
内部外围总线	内部外围总线 1	<ul style="list-style-type: none"> <li>• 连接外围功能（总线错误监视部、中断等）。</li> <li>• 与系统时钟（ICLK）同步运行。</li> </ul>
	内部外围总线 2	<ul style="list-style-type: none"> <li>• 连接外围功能（WDT、CMT、CRC、SCI 等）。</li> <li>• 与外围模块时钟（PCLK）同步运行。</li> </ul>
	内部外围总线 4	<ul style="list-style-type: none"> <li>• 连接外围功能（MTU3、GPT）。</li> <li>• 与系统时钟（ICLK）同步运行。</li> </ul>
	内部外围总线 6	<ul style="list-style-type: none"> <li>• 连接内部 ROM（P/E）/ 数据。</li> <li>• 与外围模块时钟（PCLK）同步运行。</li> </ul>

【略称说明】 P/E: 编程 / 擦除

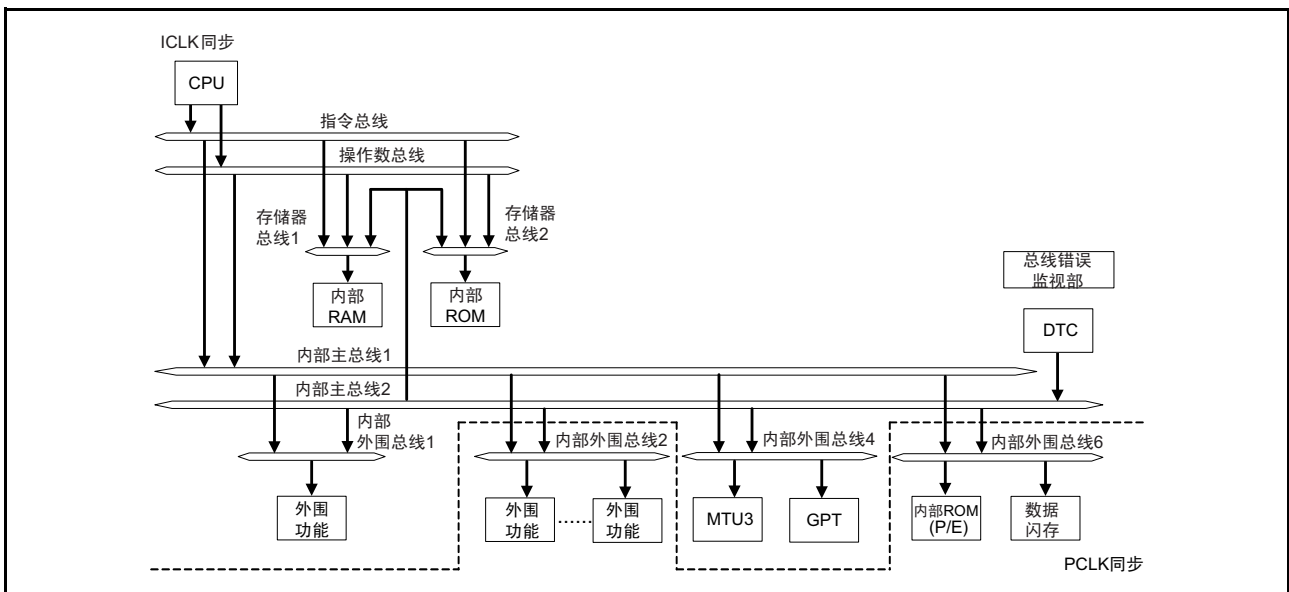


图 12.1 总线的结构图

表 12.2 各总线种类的地址对应表

地址	种类	内容	
0000 0000h ~ 0000 3FFFh	存储器总线 1	内部 RAM	
0000 4000h ~ 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	保留区	
000C 0000h ~ 000D FFFFh	内部外围总线 4	外围 I/O 寄存器	
000E 0000h ~ 000F FFFFh	保留区	保留区	
0010 0000h ~ 0011 FFFFh	内部外围总线 6	数据闪存	
0012 0000h ~ 007F 7FFFh		保留区	
007F 8000h ~ 007F 9FFFh		FUC RAM	
007F A000h ~ 007F BFFFh		保留区	
007F C000h ~ 007F C4FFh		外围 I/O 寄存器	
007F C500h ~ 007F FBFFh		保留区	
007F FC00h ~ 007F FFFFh		外围 I/O 寄存器	
0080 0000h ~ 00DF FFFFh		保留区	
00E0 0000h ~ 00FF FFFFh		内部 ROM (改写专用)	
0100 0000h ~ 7EFF FFFFh		保留区	保留区
8000 0000h ~ FFFF FFFFh		存储器总线 2	内部 ROM (读专用)

## 12.2 总线说明

### 12.2.1 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 而被 CPU 直接存取。内部 ROM 为只读区域，能被 CPU 直接存取，但是编程和擦除必须通过内部外围总线进行。

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

如果取指令和操作数存取的请求是在不同的总线（存储器总线 1、存储器总线 2 和内部主总线 1）上，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。

### 12.2.2 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。

2 条总线的优先级为内部主总线 2 > CPU 总线（操作数 > 取指令）。

### 12.2.3 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 CPU 以外的总线主控（DTC）使用的总线（内部主总线 2）构成。

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

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

2 条总线的优先级为内部主总线 2 > 内部主总线 1。如果通过 CPU 执行 XCHG 指令，就在通过 XCHG 指令进行的总线存取结束前不接受 CPU 以外的总线存取，而且在读以及回写 DTC 的传送信息过程中也不接受 DTC 以外的总线存取。

表 12.3 总线主控的优先级

优先级	总线主控
高	DTC
低	CPU

### 12.2.4 内部外围总线

连接内部外围总线的外围功能如表 12.4 所示。

表 12.4 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	<ul style="list-style-type: none"> <li>• 中断控制器</li> <li>• 总线错误监视部</li> </ul>
内部外围总线 2	<ul style="list-style-type: none"> <li>• 内部外围总线 1、4 以外的外围功能</li> <li>• 数据闪存</li> </ul>
内部外围总线 4	<ul style="list-style-type: none"> <li>• MTU3、GPT</li> </ul>
内部外围总线 6	<ul style="list-style-type: none"> <li>• 内部 ROM (P/E) / 数据闪存</li> </ul>

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

2 条总线的优先级为内部主总线 2 > 内部主总线 1。

### 12.2.5 并行运行

能在各总线主控存取不同的从属模块时并行运行。例如，CPU 能一边从内部 ROM 取指令，一边对内部 RAM 进行操作数存取，同时，DTC 还能进行外围和外围总线之间的传送。

并行运行的例子如图 12.2 所示。在此例中，CPU 能使用指令总线和操作数总线同时存取内部 ROM 和内部 RAM，而且在 CPU 存取内部 ROM 和内部 RAM 过程中，DTC 能使用内部主总线 2 同时存取外围总线。

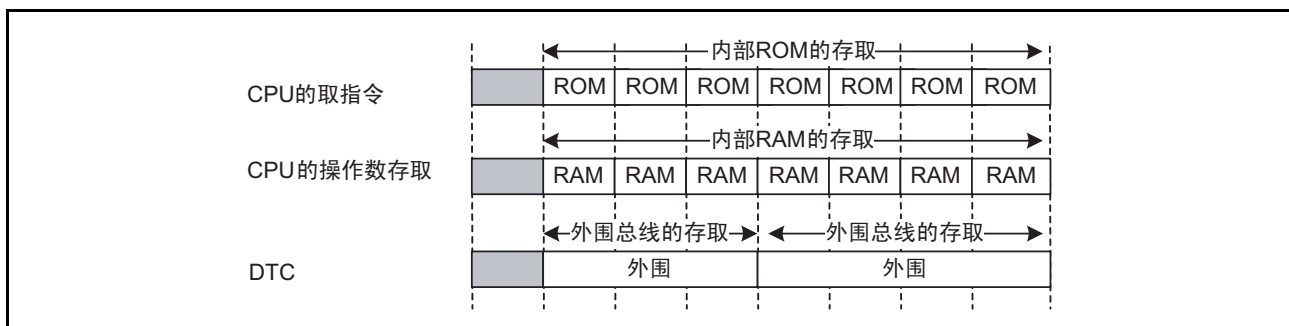


图 12.2 并行运行的例子

### 12.2.6 限制事项

#### (1) 有关 RMPA 指令和字符串操作指令的限制事项

禁止将 RMPA 指令和字符串指令的操作对象数据分配到 I/O 寄存器。否则，不保证运行。

## 12.3 寄存器说明

总线错误监视部的寄存器如表 12.5 所示。

表 12.5 总线错误监视部的寄存器一览表

寄存器名	符号	复位后的值	地址	存取长度
总线错误状态清除寄存器	BERCLR	00h	0008 1300h	8
总线错误监视允许寄存器	BEREN	00h	0008 1304h	8
总线错误状态寄存器 1	BERSR1	00h	0008 1308h	8
总线错误状态寄存器 2	BERSR2	0000h	0008 130Ah	16

### 12.3.1 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

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	R/(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)。

### 12.3.2 总线错误监视允许寄存器 (BEREN)

地址 0008 1304h

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

位	符号	位名	功能	R/W
b0	IGAEN	非法地址的存取检测允许位	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
b7-b1	—	保留位	读写值都为“0”。	R/W

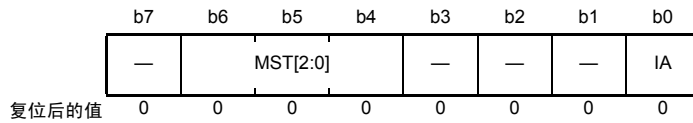
#### IGAEN 位 (非法地址的存取检测允许位)

此位设定允许或者禁止检测非法地址的存取。



### 12.3.3 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
b3-b1	—	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: 不能设定 0 1 0: 不能设定 0 1 1: DTC 1 0 0: 不能设定 1 0 1: 不能设定 1 1 0: 不能设定 1 1 1: 不能设定	R
b7	—	保留位	读取值为“0”，写操作无效。	R

BERSR1 寄存器是记录总线错误发生状态的寄存器。

BERSR1 寄存器记录是否发生非法地址的存取 (IA 位) 以及被哪个总线主控存取了 (MST[2:0] 位)。

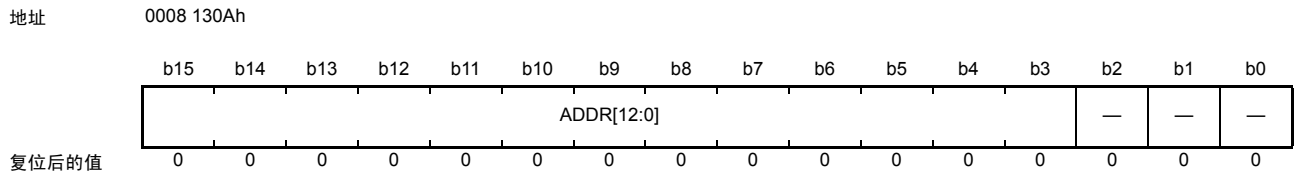
#### IA 位 (非法地址存取标志位)

此位表示是否发生了非法地址的存取错误。

#### MST[2:0] 位 (总线主控代码位)

这些位表示让总线发生错误的总线主控。

### 12.3.4 总线错误状态寄存器 2 (BERSR2)



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	发生总线错误的存取地址的高 13 位（以 512KB 为单位）	R

BERSR2 寄存器记录发生总线错误的地址的高 13 位。

#### ADDR[12:0]（总线错误发生地址位）

这些位表示发生总线错误的地址的高 13 位。

## 12.4 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。总线错误包括非法地址的存取。当存取非法区域时被检测为非法地址的存取。

### 12.4.1 总线错误的种类

总线错误包括非法地址存取。

当存取非法区域时被检测为非法地址存取。

#### 12.4.1.1 非法地址存取

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

- 存取了非法地址区域时  
非法地址存取错误的发生区域如表 12.6 所示。

#### 12.4.2 发生总线错误时的运行

一旦发生总线错误，就向 CPU 通知总线错误。当发生总线错误时，无法保证运行。

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

### 12.4.3 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 12.6 所示。

在没有发生总线错误的状态（总线错误状态寄存器 n（BERSRn）（n=1、2）被清除的状态）下，如果检测到非法地址的存取错误，就将当时的状态保存到 BERSRn 寄存器。一旦发生总线错误，即使以后再发生总线错误，在没有清除 BERSRn 的状态下也不保存当时的状态。

当 2 个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 12.6 发生的总线错误的种类

地址	内容	种类
		非法地址的存取
0000 0000h ~ 0000 3FFFh	内部 RAM（注 1）	—
0000 4000h ~ 0007 FFFFh	保留区	—
0008 0000h ~ 0009 0FFFh	外围 I/O 寄存器	—
0009 1000h ~ 0009 3FFFh		○
0009 4000h ~ 0009 41FFh		—
0009 4200h ~ 0009 FFFFh		○
000A 0000h ~ 000B FFFFh		○
000C 0000h ~ 000C 0FFFh		○
000C 1000h ~ 000C 27FFh		—
000C 2800h ~ 000D FFFFh		○
000E 0000h ~ 000F FFFFh		保留区
0010 0000h ~ 0011 FFFFh	数据闪存（注 1）	—
0012 0000h ~ 007F 7FFFh	保留区	○
007F 8000h ~ 007F 9FFFh	FCU RAM	—
007F A000h ~ 007F BFFFh	保留区	○
007F C000h ~ 007FC4FFh	外围 I/O 寄存器	—
007F C500h ~ 007F FBFFh	保留区	○
007F FC00h ~ 007F FFFFh	外围 I/O 寄存器	—
0080 0000h ~ 00DF FFFFh	保留区	—
00E0 0000h ~ 00FF FFFFh	内部 ROM（注 1）（改写专用）	—
0100 0000h ~ 7FFF FFFFh	保留区	○
8000 0000h ~ FFFF FFFFh	内部 ROM（注 1）（读专用）	—

—：不发生总线错误

○：发生总线错误

注 1. 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“30. RAM”、“31. ROM（保存代码的闪存）”和“32. 数据闪存（保存数据的闪存）”。

## 13. 存储器保护单元 (MPU)

### 13.1 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 13.1 和图 13.1 所示。

表 13.1 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)。
存储器保护的错误源判断	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。

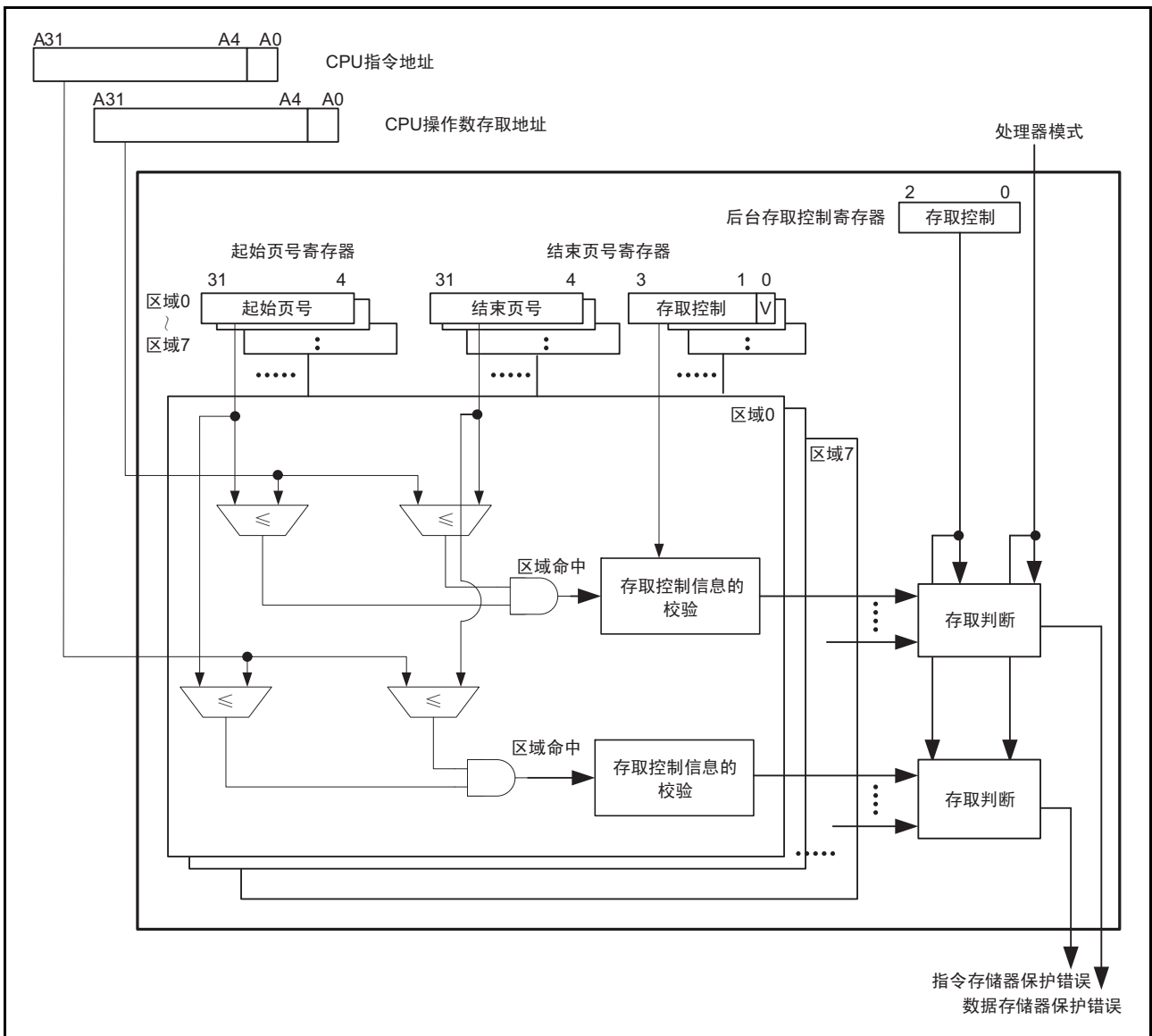


图 13.1 存储器保护单元的框图

### 13.1.1 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。

### 13.1.2 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。

### 13.1.3 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。

### 13.1.4 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。

### 13.1.5 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。

## 13.2 寄存器说明

存储器保护单元的寄存器一览如表 13.2 所示。

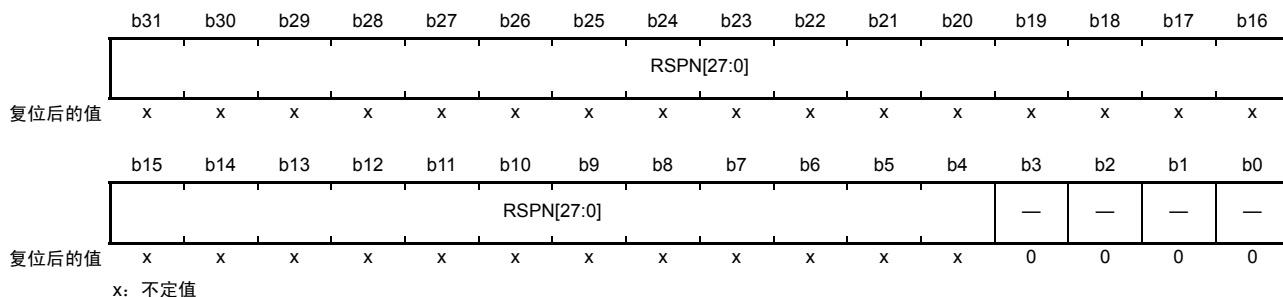
表 13.2 存储器保护单元的寄存器一览

寄存器名	符号	复位后的值 (注1)	地址	存取长度
区域 0 起始页号寄存器	RSPAGE0	xxxx xxx0h	0008 6400h	32
区域 0 结束页号寄存器	REPAGE0	xxxx xxx0h	0008 6404h	32
区域 1 起始页号寄存器	RSPAGE1	xxxx xxx0h	0008 6408h	32
区域 1 结束页号寄存器	REPAGE1	xxxx xxx0h	0008 640Ch	32
区域 2 起始页号寄存器	RSPAGE2	xxxx xxx0h	0008 6410h	32
区域 2 结束页号寄存器	REPAGE2	xxxx xxx0h	0008 6414h	32
区域 3 起始页号寄存器	RSPAGE3	xxxx xxx0h	0008 6418h	32
区域 3 结束页号寄存器	REPAGE3	xxxx xxx0h	0008 641Ch	32
区域 4 起始页号寄存器	RSPAGE4	xxxx xxx0h	0008 6420h	32
区域 4 结束页号寄存器	REPAGE4	xxxx xxx0h	0008 6424h	32
区域 5 起始页号寄存器	RSPAGE5	xxxx xxx0h	0008 6428h	32
区域 5 结束页号寄存器	REPAGE5	xxxx xxx0h	0008 642Ch	32
区域 6 起始页号寄存器	RSPAGE6	xxxx xxx0h	0008 6430h	32
区域 6 结束页号寄存器	REPAGE6	xxxx xxx0h	0008 6434h	32
区域 7 起始页号寄存器	RSPAGE7	xxxx xxx0h	0008 6438h	32
区域 7 结束页号寄存器	REPAGE7	xxxx xxx0h	0008 643Ch	32
存储器保护功能有效化寄存器	MPEN	0000 0000h	0008 6500h	32
后台存取控制寄存器	MPBAC	0000 0000h	0008 6504h	32
存储器保护错误状态清除寄存器	MPECLR	0000 0000h	0008 6508h	32
存储器保护错误状态寄存器	MPESTS	0000 0000h	0008 650Ch	32
数据存储器保护地址寄存器	MPDEA	xxxx xxxh	0008 6514h	32
区域搜索地址寄存器	MPSA	xxxx xxxh	0008 6520h	32
区域搜索操作寄存器	MPOPS	0000h	0008 6524h	16
区域无效化操作寄存器	MPOPI	0000h	0008 6526h	16
指令命中区域寄存器	MHITI	0000 0000h	0008 6528h	32
数据命中区域寄存器	MHITD	0000 0000h	0008 652Ch	32

注 1. x: 不定值

### 13.2.1 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

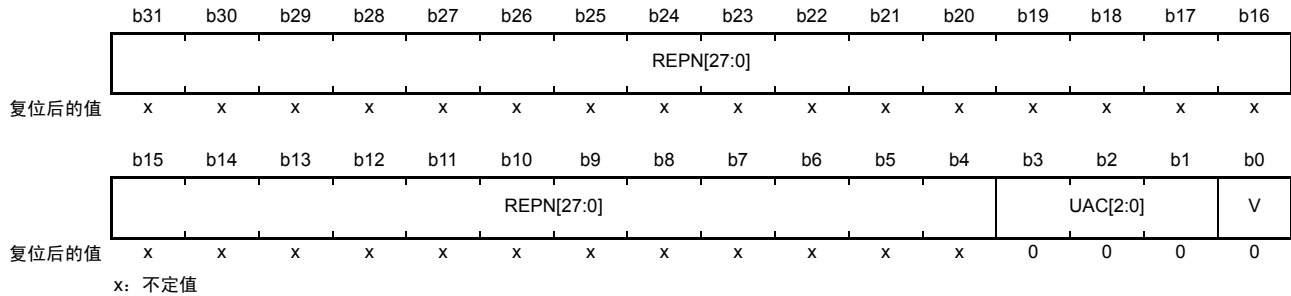
#### RSPN[27:0] 位 (区域起始页号位)

这些位设定区域起始页号。



## 13.2.2 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。

## 13.2.3 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。

## 13.2.4 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。

## 13.2.5 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	<b>【读时】</b> 0: 读 (固定) <b>【写时】</b> 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”。

## 13.2.6 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”。

### 13.2.7 数据存储保护错误地址寄存器 (MPDEA)

地址 0008 6514h



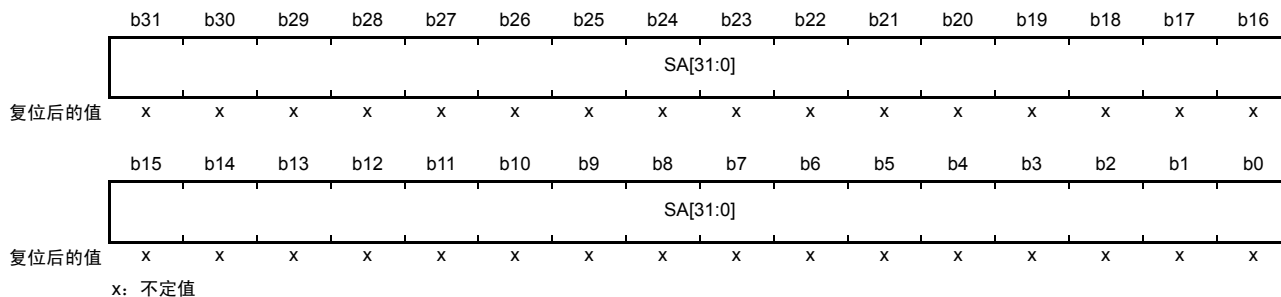
位	符号	位名	说明	R/W
b31-b0	DEA[31:0]	数据存储保护错误地址位	数据存储保护错误地址	R

#### DEA[31:0] 位 (数据存储保护错误地址位)

这些位保持因存取操作数而发生存储器保护错误的地址。

### 13.2.8 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 (REPAGEEn) 的区域结束地址进行比较的地址。

## 13.2.9 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	<b>【读时】</b> 0: 读 (固定) <b>【写时】</b> 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])。

## 13.2.10 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	<b>【读】</b> 0: 读 (固定) <b>【写】</b> 0: 不进行任何操作 1: 使全部存取控制区域 (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。

## 13.2.11 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”。

## 13.2.12 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”。

## 13.3 功能

### 13.3.1 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。

### 13.3.2 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。

### 13.3.3 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（取指令、DMA）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。

### 13.3.4 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 13.2 和图 13.3 所示。

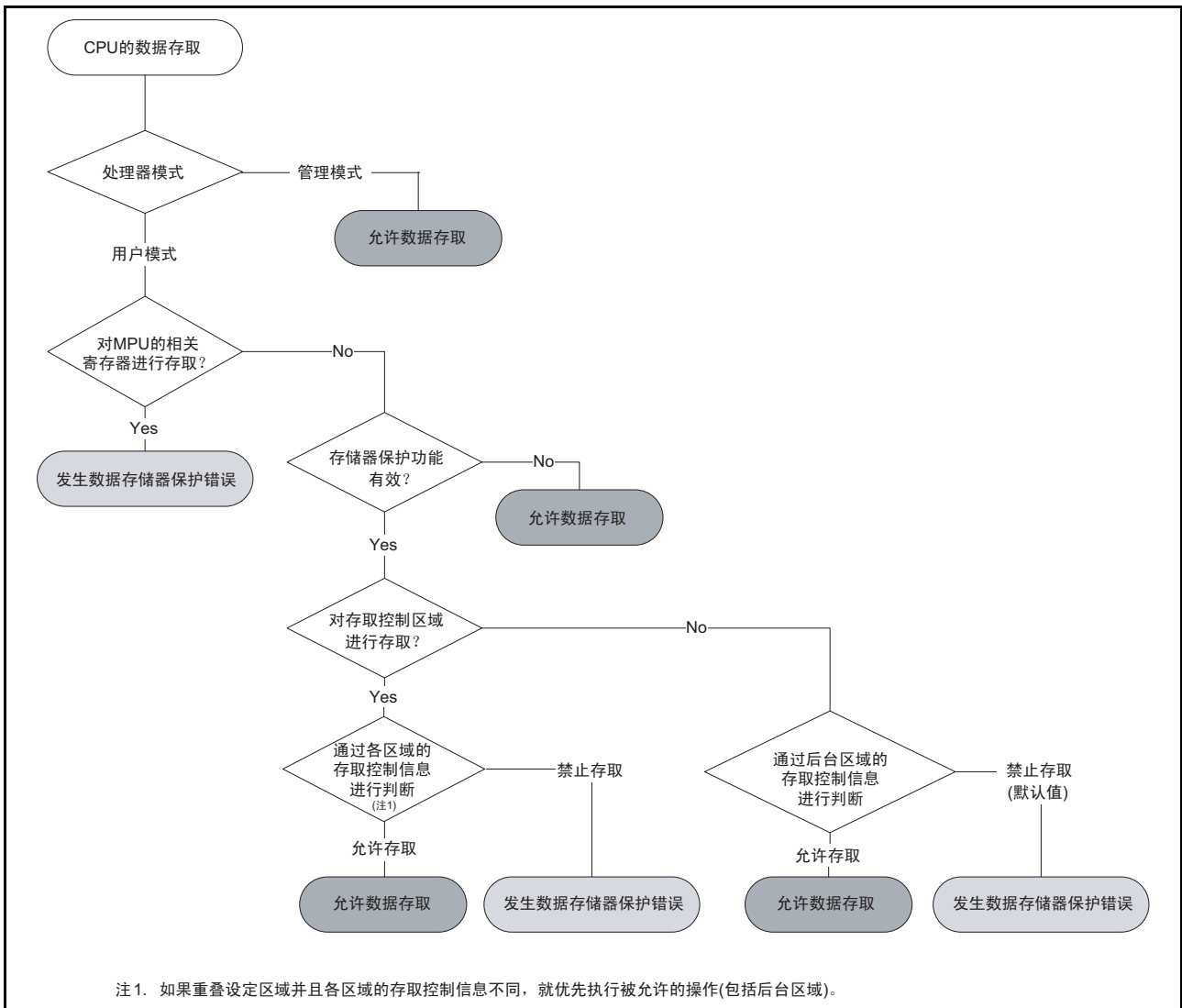


图 13.2 数据存取判断流程

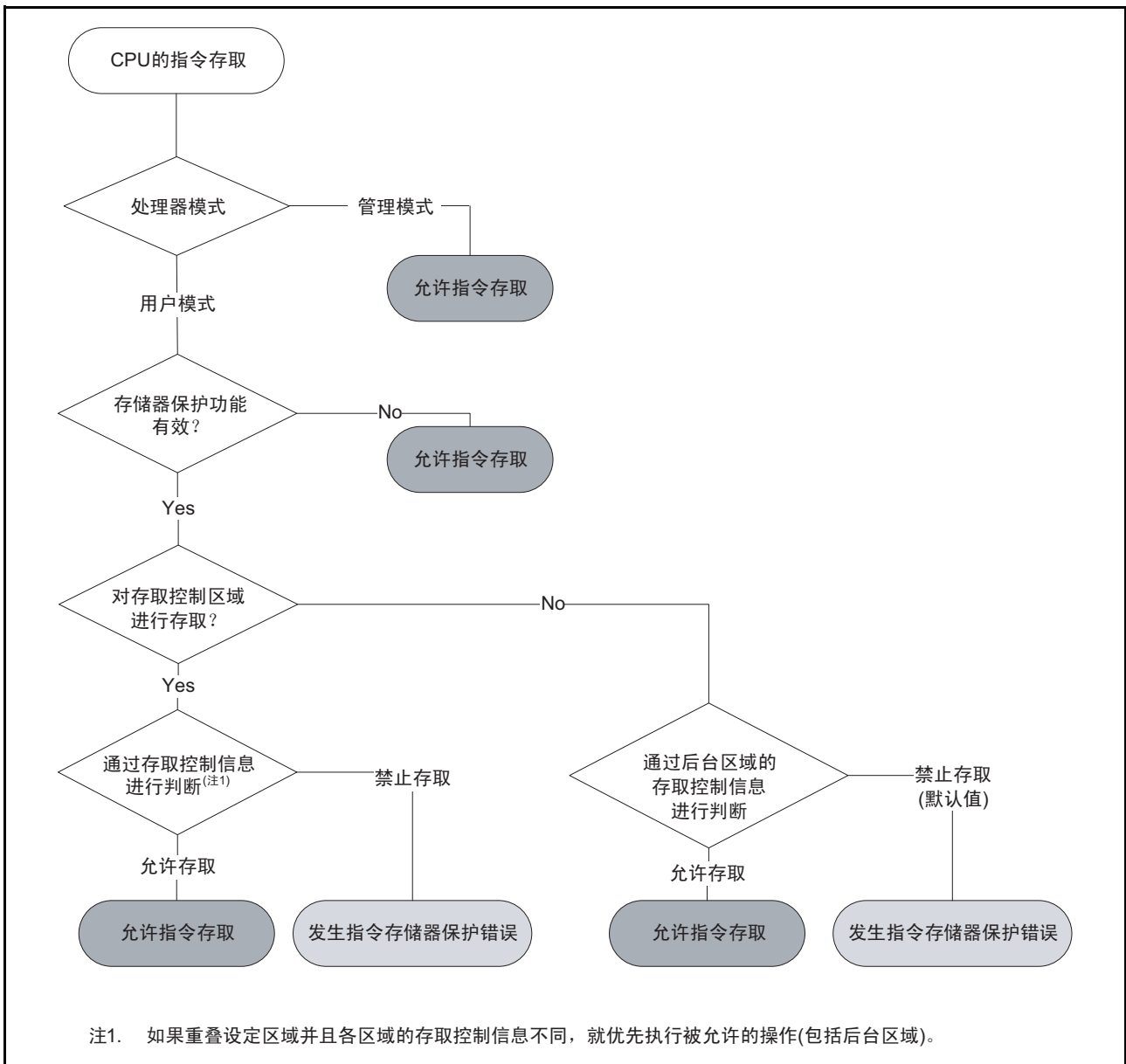


图 13.3 指令存取判断流程

## 13.4 存储器保护功能的使用步骤

### 13.4.1 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。

### 13.4.2 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”。

### 13.4.3 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。

### 13.4.4 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“10. 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。



## 14. 数据传送控制器 (DTC)

RX62T 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。

### 14.1 概要

DTC 的规格和框图分别如表 14.1 和图 14.1 所示。

表 14.1 DTC 的规格

项目	内容
传送模式	<ul style="list-style-type: none"> <li>• 正常传送模式 1 次启动进行 1 个数据的传送。</li> <li>• 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。</li> <li>• 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 255 个数据。</li> </ul>
传送通道	<ul style="list-style-type: none"> <li>• 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)。</li> <li>• 能对 1 个启动源进行多个的数据传送 (链传送)。</li> <li>• 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。</li> </ul>
传送空间	<ul style="list-style-type: none"> <li>• 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)</li> <li>• 在全地址模式中为 4G 字节。 (0000 0000h ~ FFFF FFFFh 中保留区除外的区域)</li> </ul>
数据传送单位	<ul style="list-style-type: none"> <li>• 1 个数据的位长: 8 位、16 位、32 位</li> <li>• 块大小的数据量: 1 ~ 255 个数据</li> </ul>
CPU 中断源	<ul style="list-style-type: none"> <li>• 能通过启动 DTC 的中断向 CPU 产生中断请求。</li> <li>• 在 1 次数据传送结束后向 CPU 产生中断请求。</li> <li>• 在指定数据量的数据传送结束后向 CPU 产生中断请求。</li> </ul>
跳读	<ul style="list-style-type: none"> <li>• 能指定传送信息的跳读。</li> </ul>
回写的省略	<ul style="list-style-type: none"> <li>• 当传送源地址被固定或者传送目标被固定时省略回写。</li> </ul>

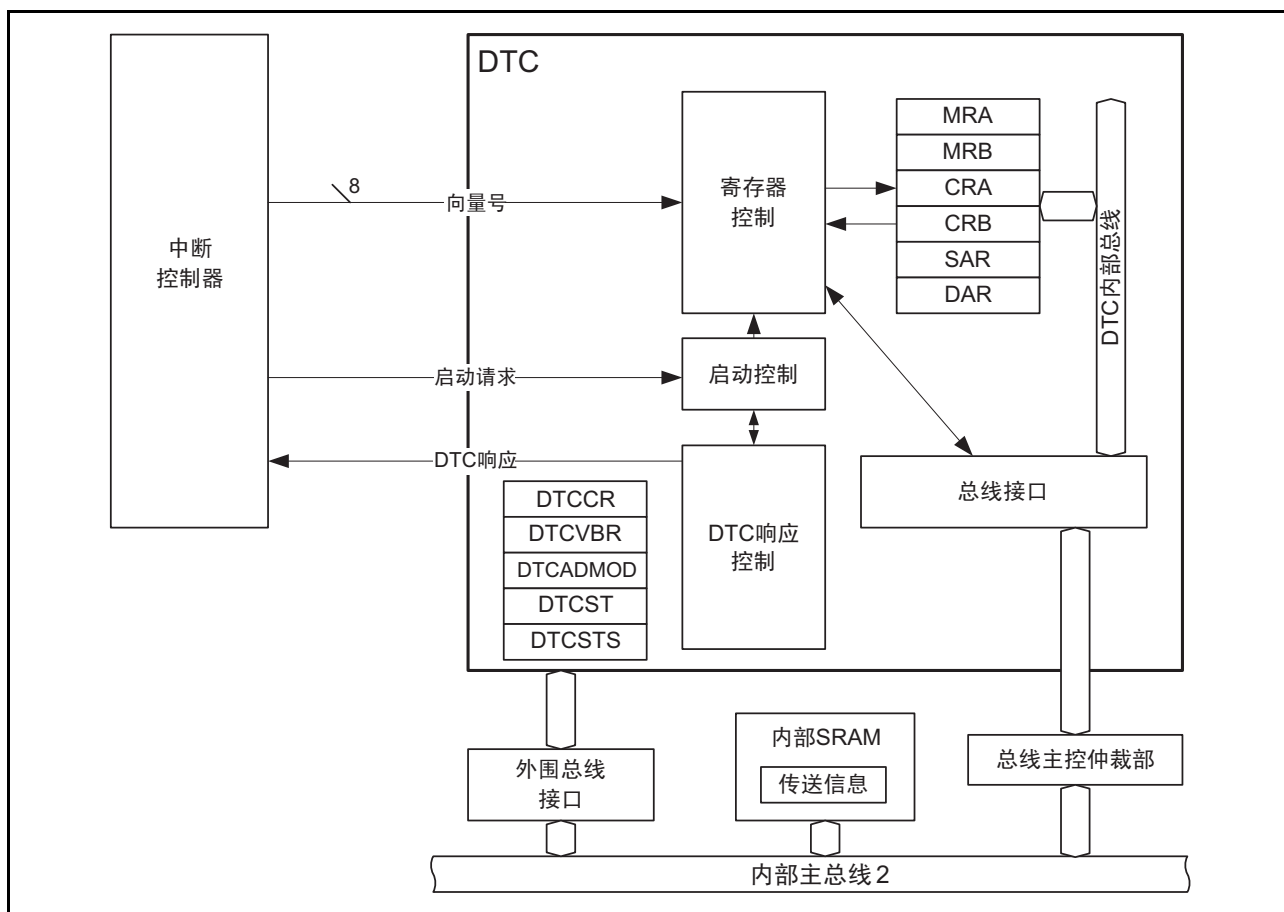


图 14.1 DTC 的框图

## 14.2 寄存器说明

DTC 的寄存器一览表如表 14.2 所示。

MRA、MRB、SAR、DAR、CRA、CRB 共 6 个寄存器是 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 RAM 区。如果产生 DTC 启动请求，就从 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 RAM 区。

表 14.2 DTC 的寄存器一览表

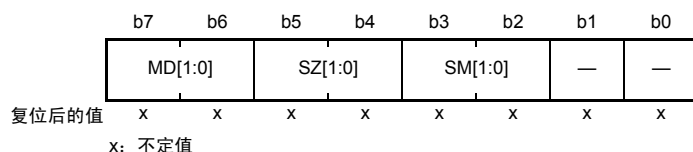
寄存器名	符号	复位后的值	地址	存取长度 (位)
DTC 模式寄存器 A	MRA	xxh	—	—
DTC 模式寄存器 B	MRB	xxh	—	—
DTC 传送源地址寄存器	SAR	xxxxxxxxh	—	—
DTC 传送目标地址寄存器	DAR	xxxxxxxxh	—	—
DTC 传送计数寄存器 A	CRA	xxxxh	—	—
DTC 传送计数寄存器 B	CRB	xxxxh	—	—
DTC 控制寄存器	DTCCR	08h	0008 2400h	8
DTC 向量基址寄存器	DTCVBR	00000000h	0008 2404h	32
DTC 地址模式寄存器	DTCADM0D	00h	0008 2408h	8
DTC 模块启动寄存器	DTCST	00h	0008 240Ch	8
DTC 状态寄存器	DTCSTS	0000h	0008 240Eh	16

x: 不定值

注. 要启动 DTC 时，需要设定中断控制器 (ICU) 的 DTCERn.DTCE 位 (n= 中断向量号) 和 IERm.IENj 位 (m=02h ~ 1Fh, j=7 ~ 0)，详细内容请参照“11. 中断控制器 (ICU)”。

## 14.2.1 DTC 模式寄存器 A (MRA)

地址 (不能由 CPU 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: 字节传送 0 1: 字传送 1 0: 长字传送 1 1: 不能设定	—
b7-b6	MD[1:0]	DTC 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	—

MRA 寄存器是选择 DTC 运行模式的寄存器。  
CPU 不能直接存取 MRA 寄存器。

## SM[1:0] 位 (传送源地址的寻址方式位)

这些位指定数据传送后的 SAR 寄存器的运行。

## SZ[1:0] 位 (DTC 数据传送长度位)

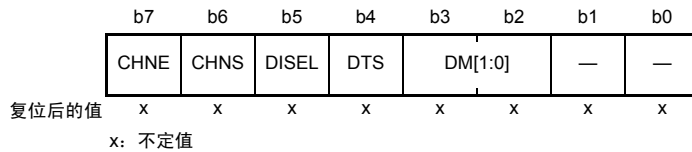
这些位指定传送数据的长度。

## MD[1:0] 位 (DTC 传送模式选择位)

这些位指定 DTC 传送模式。

## 14.2.2 DTC 模式寄存器 B (MRB)

地址 (不能由 CPU 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: 只在传送计数器为“0”时进行链传送	—
b7	CHNE	DTC 链传送允许位	0: 禁止链传送 1: 允许链传送	—

MRB 寄存器是选择 DTC 运行模式的寄存器。  
CPU 不能直接存取 MRB 寄存器。

## DM[1:0] 位 (传送目标地址的寻址方式位)

这些位指定数据传送后的 DAR 寄存器的运行。

## DTS 位 (DTC 传送模式选择位)

在重复传送模式或者块传送模式中, 此位指定是将传送源还是将传送目标作为重复区域或者块区域。

## DISEL 位 (DTC 中断选择位)

此位指定是在每次 DTC 数据传送时还是只在数据传送结束时向 CPU 请求中断。

## CHNS 位 (DTC 链传送选择位)

此位选择链传送的条件。

如果下一次传送是链传送, 就不判断指定次数的传送结束, 并且不清除中断状态标志, 也不向 CPU 请求中断。

## CHNE 位 (DTC 链传送允许位)

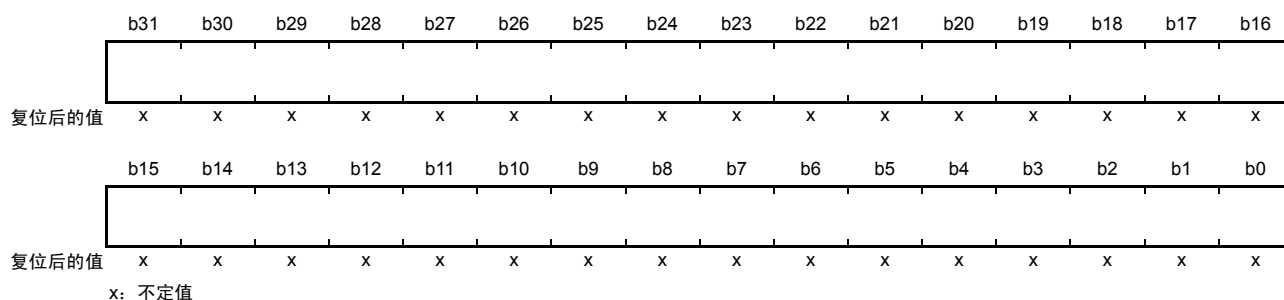
此位指定链传送。

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“14.4.6 链传送”。

### 14.2.3 DTC 传送源地址寄存器 (SAR)

地址 (不能由 CPU 直接存取)



SAR 寄存器是设定传送源起始地址的寄存器。

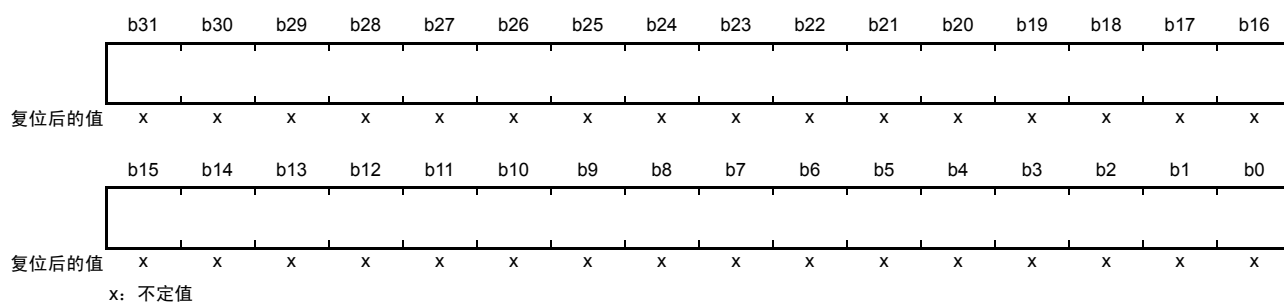
在全地址模式中 32 位有效。

在短地址模式中低 24 位有效，忽视高 8 位 (b31 ~ b24) 的设定，通过 b23 指定的值进行位扩展。

CPU 不能直接存取 SAR 寄存器。

### 14.2.4 DTC 传送目标地址寄存器 (DAR)

地址 (不能由 CPU 直接存取)



DAR 寄存器是设定传送目标起始地址的寄存器。

在全地址模式中 32 位有效。

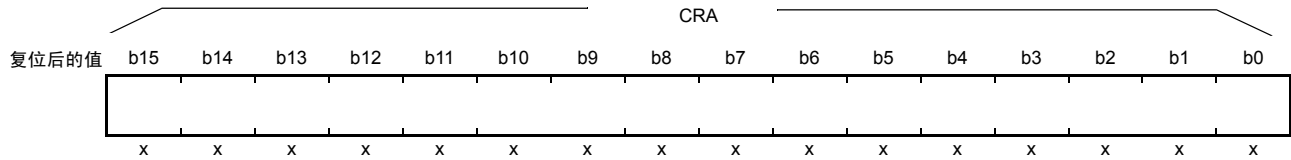
在短地址模式中低 24 位有效，忽视高 8 位 (b31 ~ b24) 的设定，通过 b23 指定的值进行位扩展。

CPU 不能直接存取 DAR 寄存器。

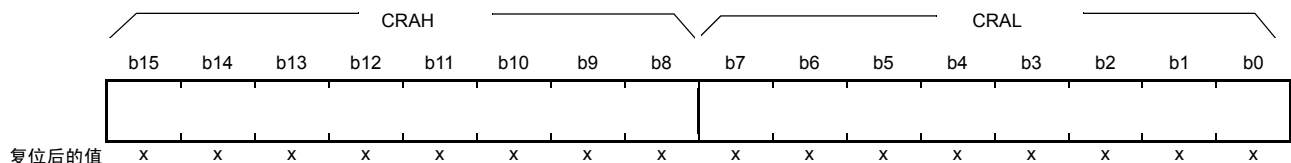
### 14.2.5 DTC 传送计数器 A (CRA)

地址 (不能由 CPU 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 1. 功能因传送模式而不同。

符号	寄存器名	功能	R/W
CRAL	传送计数器 A 的低位寄存器	设定传送次数。	—
CRAH	传送计数器 A 的高位寄存器		—

注. 在重复传送模式和块传送模式中, 必须给 CRAH 寄存器和 CRAL 寄存器设定相同的值。

CRA 寄存器是指定 DTC 传送次数的寄存器, 其功能因传送模式而不同。

CPU 不能直接存取 CRA 寄存器。

(1) 正常传送模式 (MRA.MD[1:0] 位 =00b)

在正常传送模式中, CRA 寄存器用作 16 位传送计数器。

当设定值为“0001h”时, 传送次数为 1 次; 当设定值为“FFFFh”时, 传送次数为 65535 次; 当设定值为“0000h”时, 传送次数为 65536 次。

每进行 1 次数据传送, CRA 寄存器就减 1。

(2) 重复传送模式 (MRA.MD[1:0] 位 =01b)

CRAH 寄存器保持传送次数, CRAL 寄存器用作 8 位传送计数器。

当设定值为“01h”时, 传送次数为 1 次; 当设定值为“FFh”时, 传送次数为 255 次; 当设定值为“00h”时, 传送次数为 256 次。

每进行 1 次数据传送, CRAL 寄存器就减 1。当 CRAL 寄存器变为“00h”时, 就传送 CRAH 寄存器的值。

(3) 块传送模式 (MRA.MD[1:0] 位 =10b)

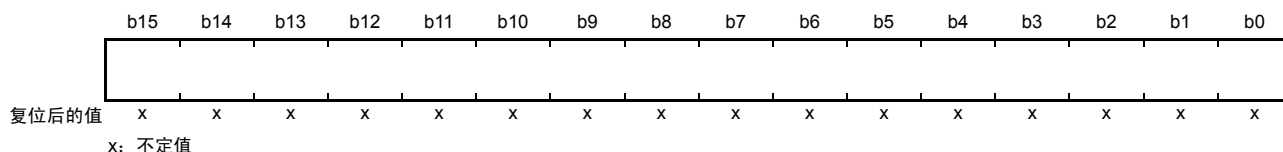
CRAH 寄存器保持块的大小, CRAL 寄存器用作 8 位块大小计数器。

当设定值为“01h”时, 传送次数为 1 次; 当设定值为“FFh”时, 传送次数为 255 次。不能设定“00h”。

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。

### 14.2.6 DTC 传送计数寄存器 B (CRB)

地址 (不能由 CPU 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。

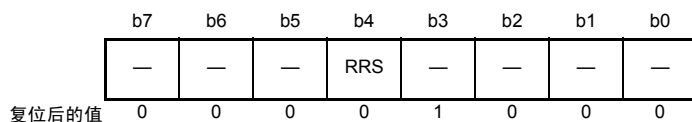
当设定值为“0001h”时，传送次数为 1 次；当设定值为“FFFFh”时，传送次数为 65535 次；当设定值为“0000h”时，传送次数为 65536 次。每进行 1 次数据传送，CRB 寄存器就减 1。

在正常传送模式和重复传送模式中，必须给 CRB 寄存器设定“FFFFh”。

CPU 不能直接存取 CRB 寄存器。

### 14.2.7 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

DTCCR 寄存器是选择 DTC 控制的寄存器。

#### 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。

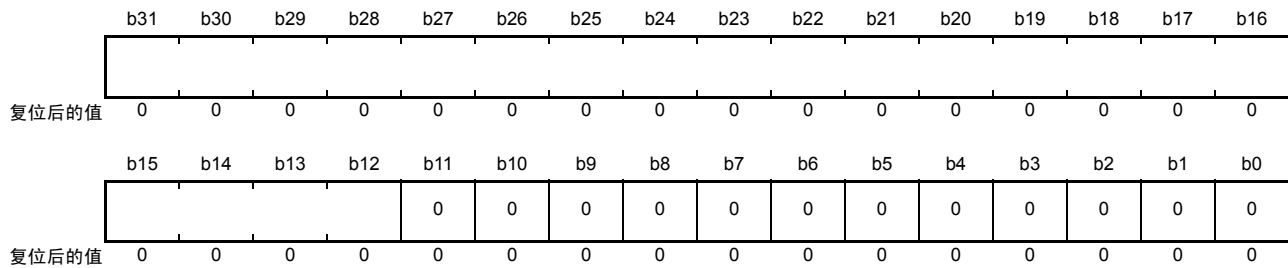
如果向量号相同并且 RRS 位为“1”，就不读传送信息而进行 DTC 的数据传送。如果上次的启动为链传送，就读传送信息，与 RRS 位的值无关。

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。



### 14.2.8 DTC 向量基址寄存器 (DTCVBR)

地址 0008 2404h



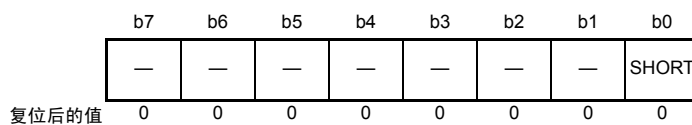
DTCVBR 寄存器设定在计算 DTC 向量表地址时的基址。

低 12 位 (b11-b0) 固定为“0”，写操作无效。

忽视高 4 位 (b31-b28) 的写操作，通过 b27 指定的值进行位扩展。

### 14.2.9 DTC 地址模式寄存器 (DTCADM0D)

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

DTCADM0D 寄存器是设定 DTC 能存取的区域寄存器。

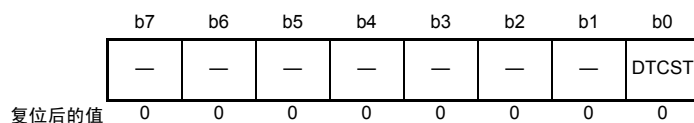
#### SHORT 位 (短地址模式设定位)

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)。

## 14.2.10 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 寄存器是设定 DTC 模块的运行或者停止的寄存器。

## DTCST 位 (DTC 模块启动位)

为了使 DTC 能接受启动请求, 必须将 DTCST 位置“1”。如果将 DTCST 位置“0”, 就不能接受新的启动请求。

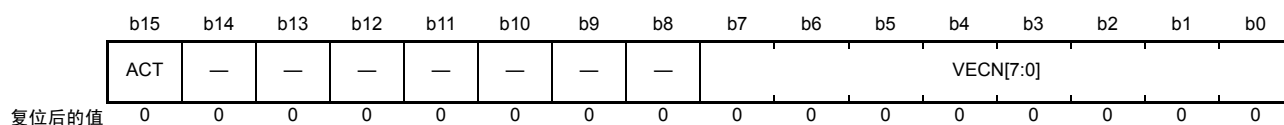
如果在运行过程中将此位改写为“0”, 已接受的启动请求就保持到处理结束为止。

在向模块停止状态以及向全模块时钟停止模式、软件待机模式、深度软件待机模式转移时, 必须将 DTCST 位置“0”。

有关向模块停止状态以及向全模块时钟停止模式、软件待机模式、深度软件待机模式的转移, 请参照“14.8 低功耗功能”和“9. 低功耗功能”。

## 14.2.11 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”），表示有效值。	R
b14-b8	—	保留位	读写值都为“0”。	R/W
b15	ACT	DTC 激活标志	0: 不在进行 DTC 传送 1: 正在进行 DTC 传送	R

DTCSTS 寄存器是表示 DTC 传送状态的寄存器。

## 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。

在读 DTCSTS 寄存器时，如果 ACT 标志为“1”（正在进行 DTC 传送），读到的 VECN[7:0] 位就为有效值；如果 ACT 标志为“0”（不在进行 DTC 传送），读到的 VECN[7:0] 位就为无效值。

启动源和向量号的关系请参照表 14.3。

## ACT 标志 (DTC 激活标志)

表示 DTC 的传送状态。

[ 为“1”的条件 ]

- 通过启动请求进行了 DTC 启动时

[ 为“0”的条件 ]

- 对于 1 次启动请求 DTC 运行结束时

### 14.3 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。

有关软件启动 (SWINT)，请参照“11. 中断控制器 (ICU)”。

在 1 次数据传送 (在链传送时，为连续传送的最后) 结束时，DTC 进行以下的运行：

- 在指定的总传送量的传送结束时，将  $DTCERn.DTCE$  位置“0”，然后向 CPU 请求中断。
- 在上述以外的传送结束时，如果  $MRB.DISEL$  位为“1”，就向 CPU 请求中断；如果  $MRB.DISEL$  位为“0”，就将作为启动源的中断状态标志 ( $IRi.IR$ ) 位置“0”。

#### 14.3.1 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，使基址 (起始地址) 的低 12 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 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  $DTCADMOD.SHORT$  位设定短地址模式 ( $SHORT$  位为“1”) 或者全地址模式 ( $SHORT$  位为“0”)。

DTC 向量表和传送信息的对应以及 RAM 区内传送信息的分配分别如图 14.2 和如图 14.3 所示。低位地址因分配区域的字节序而不同，详细内容请参照图 14.16。

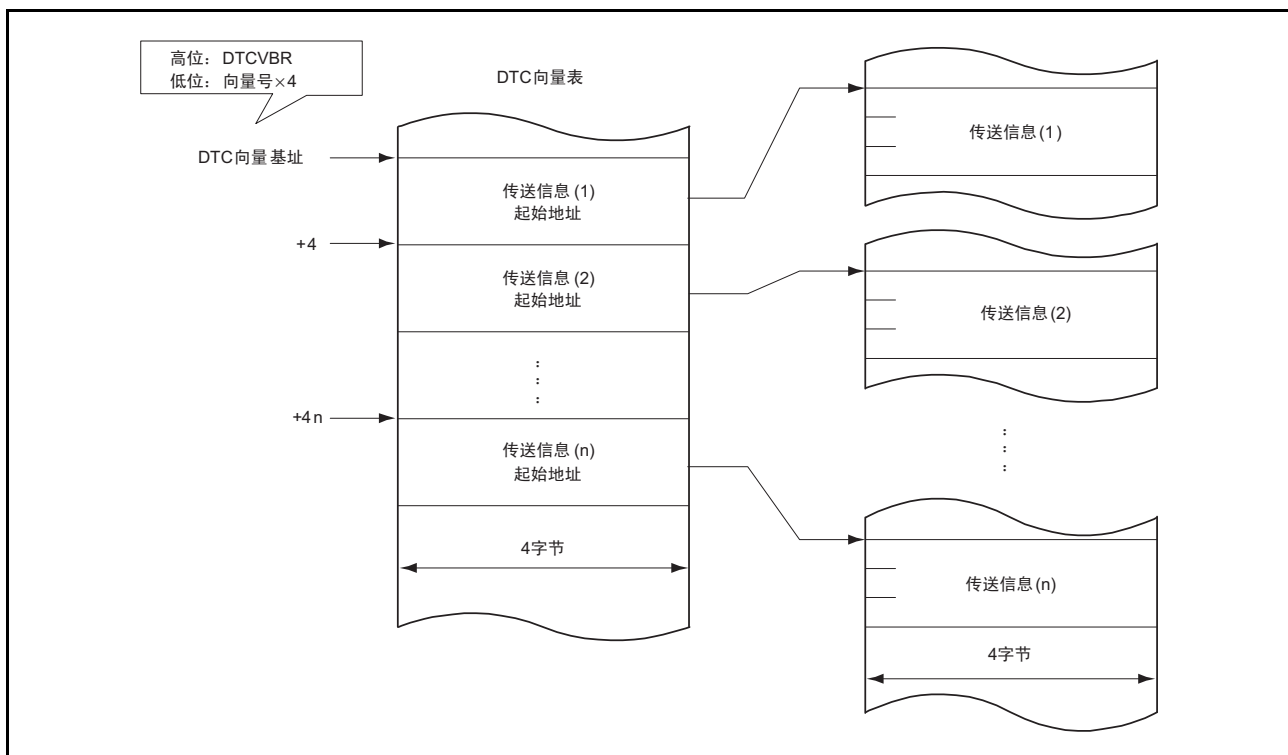


图 14.2 DTC 向量表和传送信息的对应

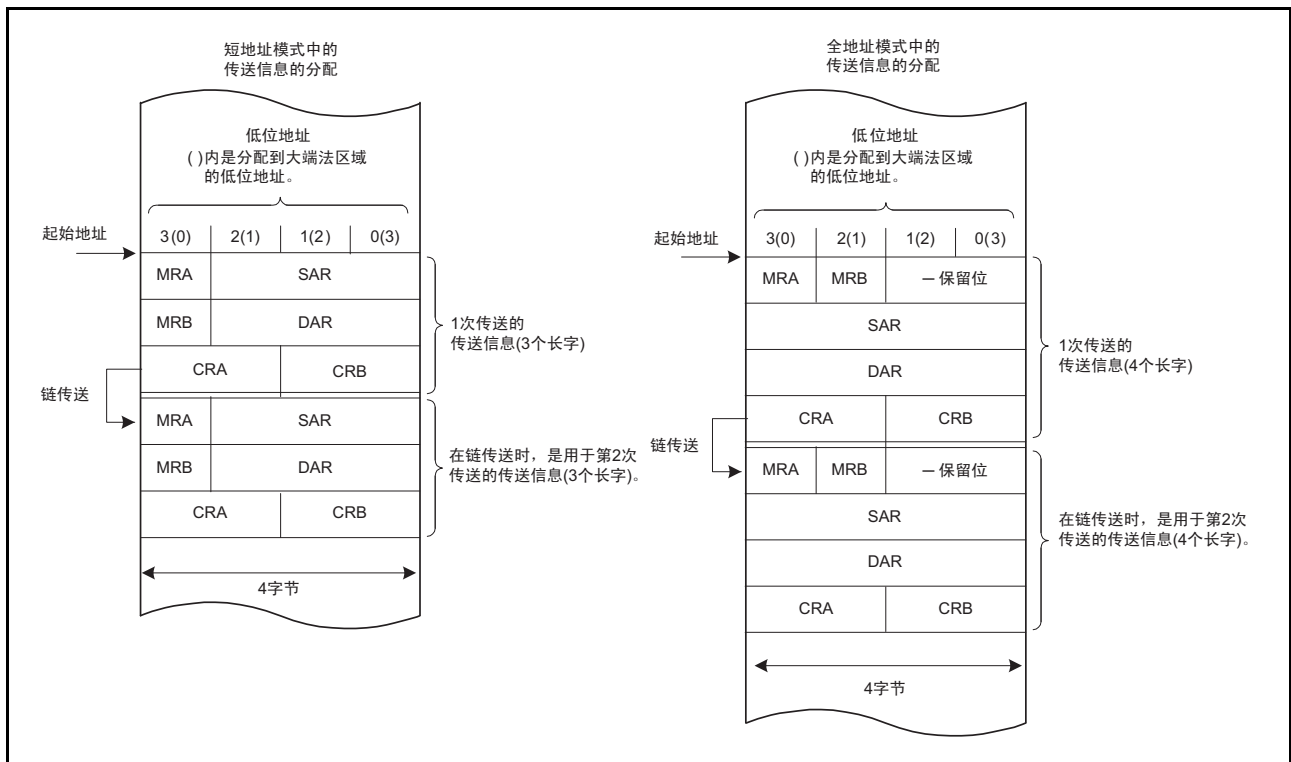


图 14.3 数据区内的传送信息的分配



表 14.3 中断源、DTC 向量地址和 ICU.DTCERn 寄存器的对应 (2 / 2)

启动请求发生源	启动源	向量号	DTC 向量地址的偏移量	ICU.DTCERn	优先级 (注 1)
MTU6	TGIA6	142	0238h	ICU.DTCER142	↑ 高
	TGIB6	143	023Ch	ICU.DTCER143	
	TGIC6	144	0240h	ICU.DTCER144	
	TGID6	145	0244h	ICU.DTCER145	
MTU7	TGIA7	149	0254h	ICU.DTCER149	
	TGIB7	150	0258h	ICU.DTCER150	
	TGIC7	151	025Ch	ICU.DTCER151	
	TGID7	152	0260h	ICU.DTCER152	
	TCIV7	153	0264h	ICU.DTCER153	
GPT0	GTCIA0	174	02B8h	ICU.DTCER174	
	GTCIB0	175	02BCh	ICU.DTCER175	
	GTCIC0	176	02C0h	ICU.DTCER176	
	GTCIE0	177	02C4h	ICU.DTCER177	
	GTCIV0	178	02C8h	ICU.DTCER178	
	LOCO1	179	02CCh	ICU.DTCER179	
GPT1	GTCIA1	180	02D0h	ICU.DTCER180	
	GTCIB1	181	02D4h	ICU.DTCER181	
	GTCIC1	182	02D8h	ICU.DTCER182	
	GTCIE1	183	02DCh	ICU.DTCER183	
	GTCIV1	184	02E0h	ICU.DTCER184	
GPT2	GTCIA2	186	02E8h	ICU.DTCER186	
	GTCIB2	187	02ECh	ICU.DTCER187	
	GTCIC2	188	02F0h	ICU.DTCER188	
	GTCIE2	189	02F4h	ICU.DTCER189	
	GTCIV2	190	02F8h	ICU.DTCER190	
GPT3	GTCIA3	192	0300h	ICU.DTCER192	
	GTCIB3	193	0304h	ICU.DTCER193	
	GTCIC3	194	0308h	ICU.DTCER194	
	GTCIE3	195	030Ch	ICU.DTCER195	
	GTCIV3	196	0310h	ICU.DTCER196	
SCI0	RXI0	215	035Ch	ICU.DTCER215	
	TXI0	216	0360h	ICU.DTCER216	
SCI1	RXI1	219	036Ch	ICU.DTCER219	
	TXI1	220	0370h	ICU.DTCER220	
SCI2	RXI2	223	037Ch	ICU.DTCER223	
	TXI2	224	0380h	ICU.DTCER224	
RIIC0	ICRXI0	247	03DCh	ICU.DTCER247	
	ICTXI0	248	03E0h	ICU.DTCER248	低

注 1. 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 DTC 传送过程中发生多个启动请求, 就在该传送结束时接受最高优先级的请求。如果在 DTC 模块启动寄存器 (DTCST) 为“0”的状态下发生多个启动请求, 就在 DTC 变为启动允许状态 (DTCST 为“1”)时接受最高优先级的请求。

## 14.4 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 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 14.4 所示。

表 14.4 DTC 的传送模式

传送模式	1 次启动请求能传送的数据长度	存储器地址的增减	能指定的传送次数
正常传送模式	1 字节 / 字 / 长字	增减 1、2、4 或者地址固定	1 ~ 65536 次
重复传送模式 (注 1)	1 字节 / 字 / 长字	增减 1、2、4 或者地址固定	1 ~ 256 次 (注 3)
块传送模式 (注 2)	CRAH 寄存器指定的块大小 (1 ~ 256 字节 / 字 / 长字)	增减 1、2、4 或者地址固定	1 ~ 65536 次

注 1. 将传送源或者传送目标设定为重复区域。

注 2. 将传送源或者传送目标设定为块区域。

注 3. 在指定次数的传送结束后，恢复初始状态并且继续（重复）运行。

能通过先将 MRB.CHNE 位置“1”，用 1 个启动源进行多次传送（链传送）。也能通过设定 MRB.CHNS 位，在传送计数器为“0”时进行链传送。

DTC 的运行流程图如图 14.4 所示，链传送的条件如表 14.5 所示（省略了第 2 次到第 3 次的传送组合以及第 3 次以后的传送组合）。



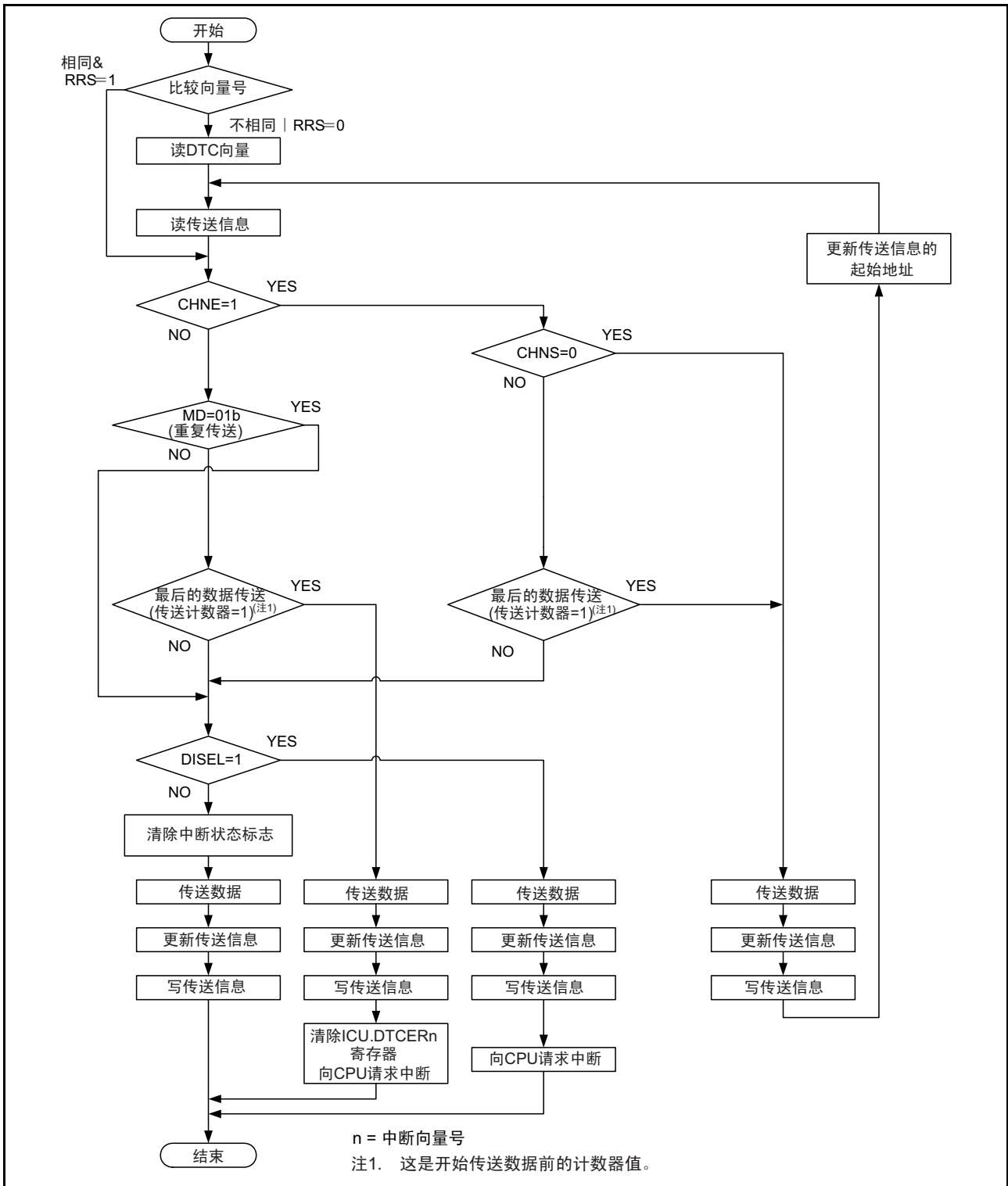


图 14.4 DTC 的运行流程图

表 14.5 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，最后传送时的计数器变化为 (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。

#### 14.4.1 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。

在产生 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，就必定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”，就必定读传送信息，与 RRS 位的值无关。跳读传送信息的运行例子如图 14.13 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。

### 14.4.2 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 14.6 所示。

与短地址模式和全地址模式的设定无关，必须回写 CRA 寄存器和 CRB 寄存器。在全地址模式中，必须省略 MRA 寄存器和 MRB 寄存器的回写。

表 14.6 传送信息的回写省略条件以及省略回写的寄存器

MRA.SM[1:0] 位		MRB.DM[1:0] 位		SAR 寄存器	DAR 寄存器
b3	b2	b3	b2		
0	0	0	0	省略	省略
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	省略	回写
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	回写	省略
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	回写	回写
1	0	1	1		
1	1	1	0		
1	1	1	1		

### 14.4.3 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 14.7 和图 14.5 所示。

表 14.7 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。

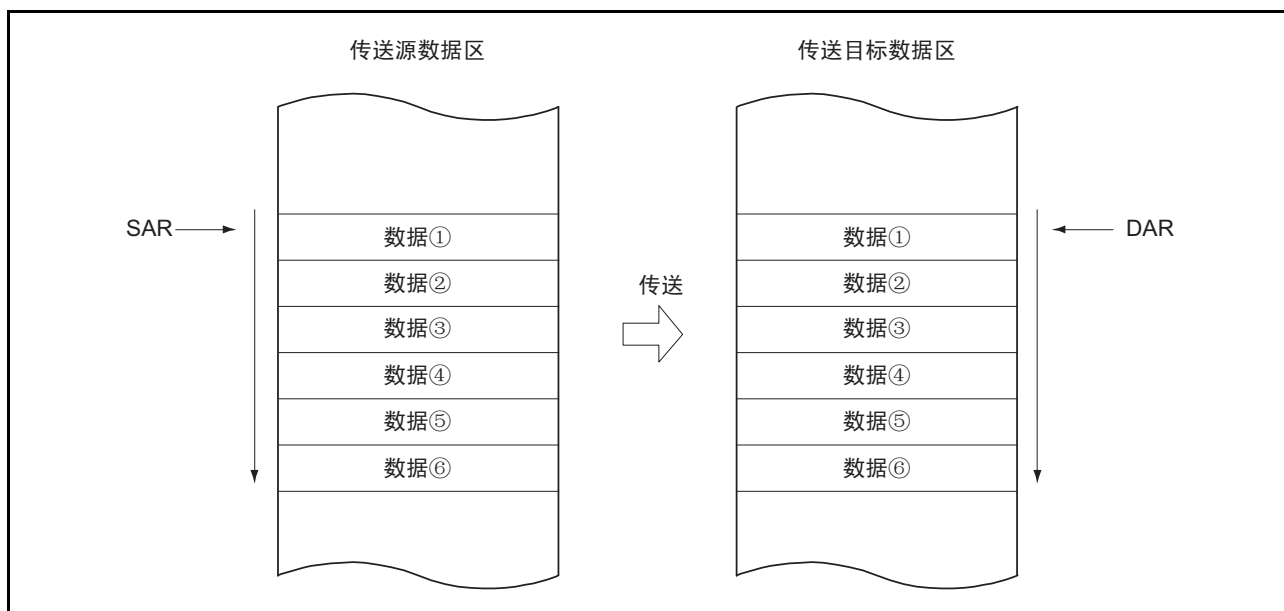


图 14.5 正常传送模式的存储器映像

#### 14.4.4 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 14.8 和图 14.6 所示。

表 14.8 重复传送模式的寄存器功能

寄存器	功能	写传送信息时被回写的值	
		CRAL≠1	CRAL=1
SAR	传送源地址	递增 / 递减 / 固定 (注 1)	(当 MRB.DTS 位为“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。

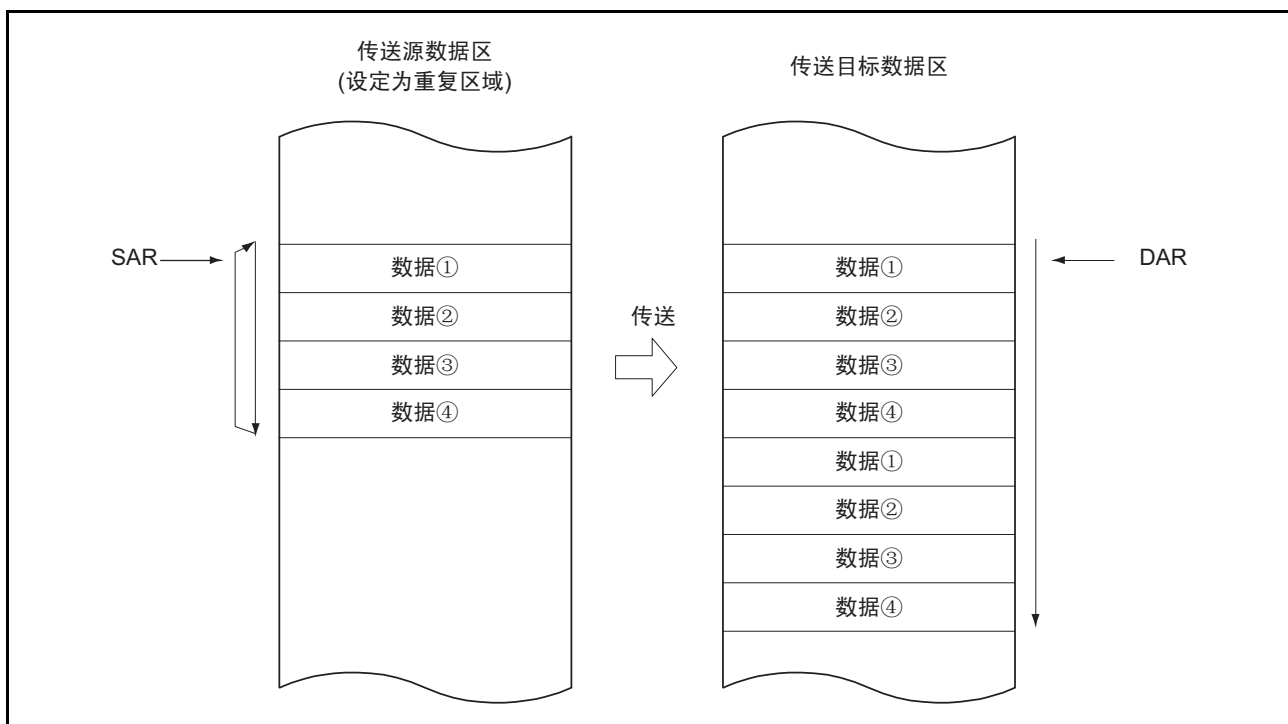


图 14.6 重复传送模式的存储器映像（将传送源设定为重复区域的情况）

### 14.4.5 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 ~ 255 字节（或者 1 ~ 255 字、1 ~ 255 长字）。

一旦指定的 1 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。

块传送次数能指定为 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 14.9 和图 14.7 所示。

表 14.9 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. 在地址固定模式的情况下，省略回写。

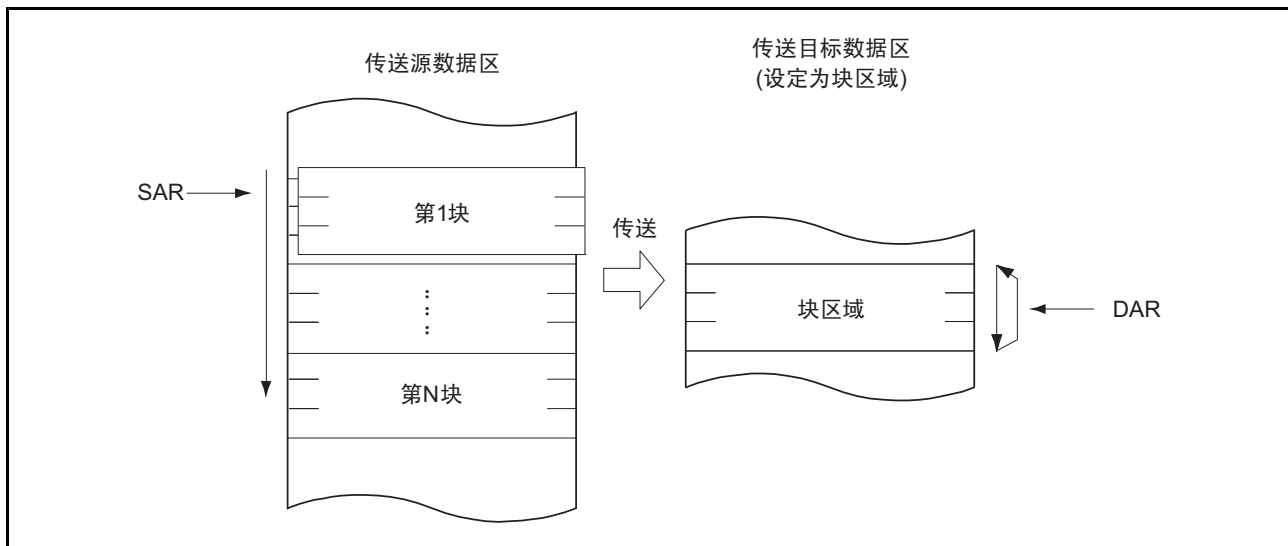


图 14.7 块传送模式的存储器映像（将传送目标指定为块区域的情况）

### 14.4.6 链传送

如果将 MRB.CHNE 位置“1”就能进行链传送。链传送是用 1 个启动源连续进行多个数据传送。

如果将 MRB.CHNE 位置“1”、MRB.CHNS 位置“0”，在指定的传送次数结束时，不向 CPU 请求中断，以及在将 MRB.DISEL 位置“1”（在每次进行 DTC 数据传送时向 CPU 请求中断）时，不向 CPU 中请求中断。并且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、MRB 寄存器。

链传送的运行如图 14.8 所示。

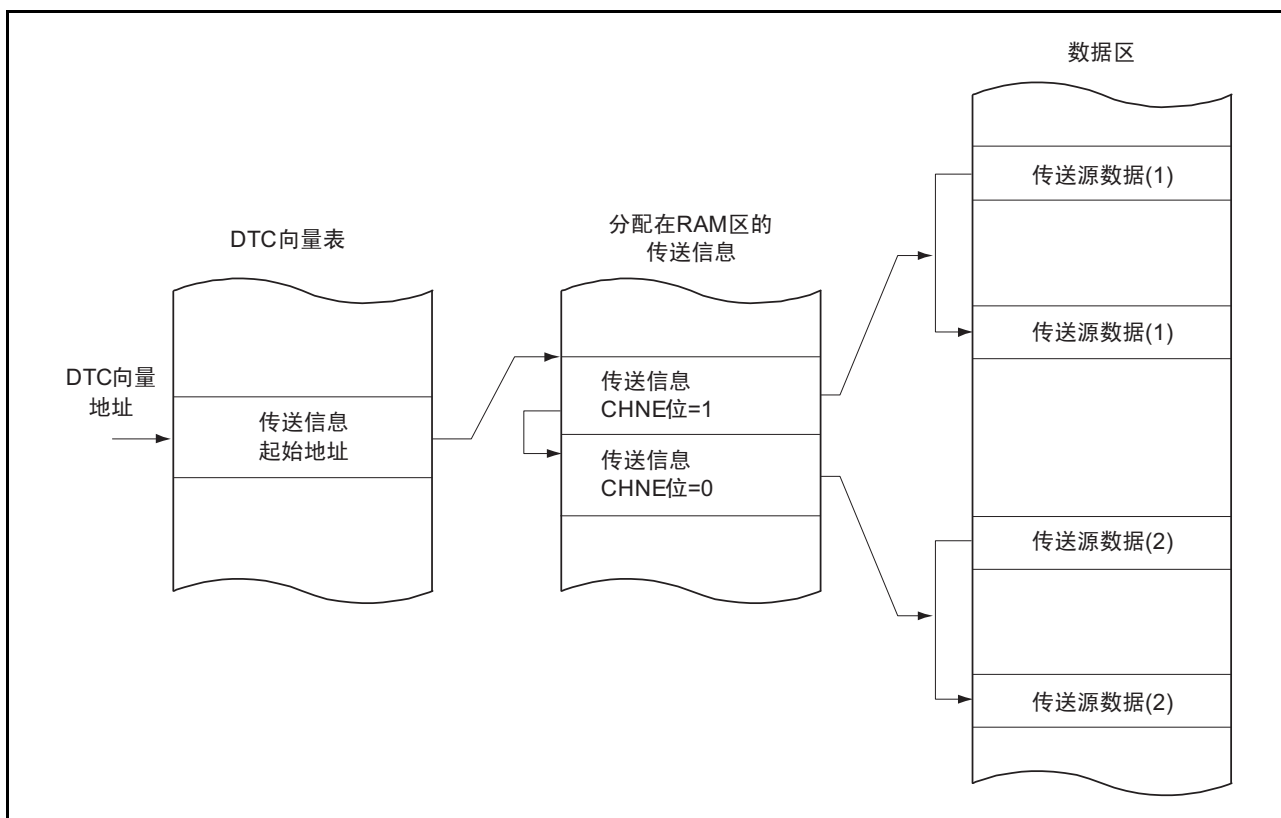


图 14.8 链传送的运行

如果将 MRB.CHNE 位置“1”、MRB.CHNS 位置“1”，只在指定的数据传送次数结束时（传送计数器变为“0”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。

链传送条件的详细内容请参照“表 14.5”。

### 14.4.7 运行时序

DTC 的运行时序例子如图 14.9 ~ 图 14.12 所示。

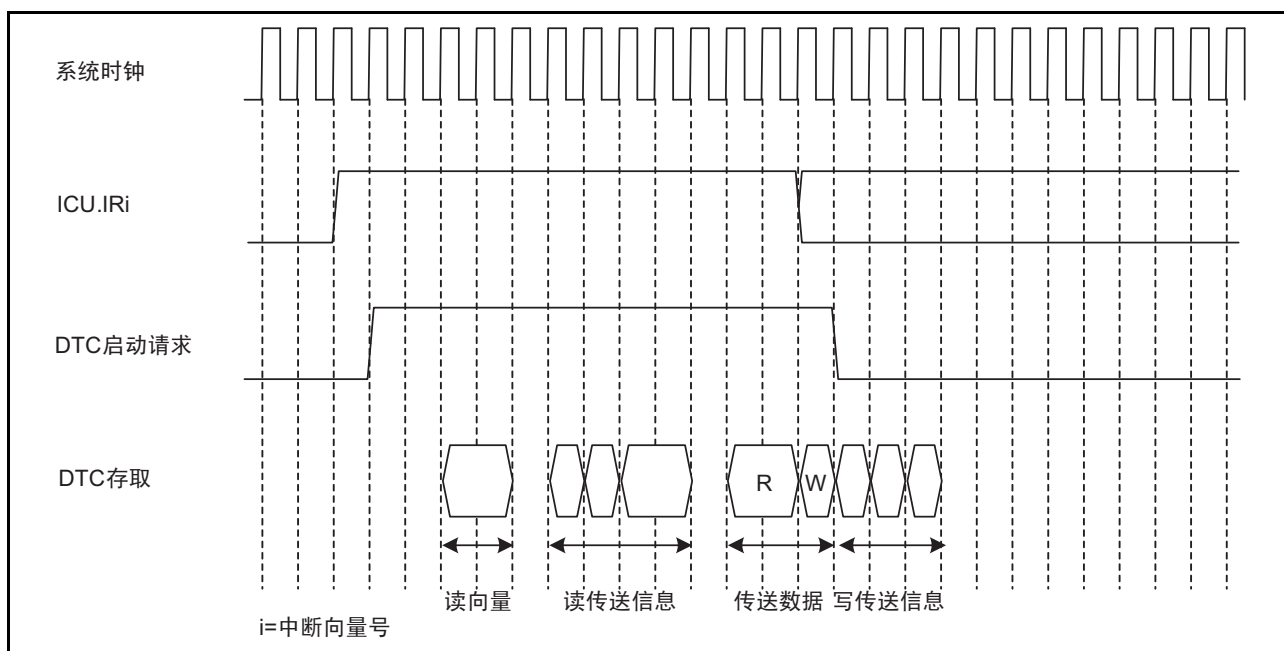


图 14.9 DTC 的运行时序例子 (1)  
(短地址模式、正常传送模式、重复传送模式的情况)

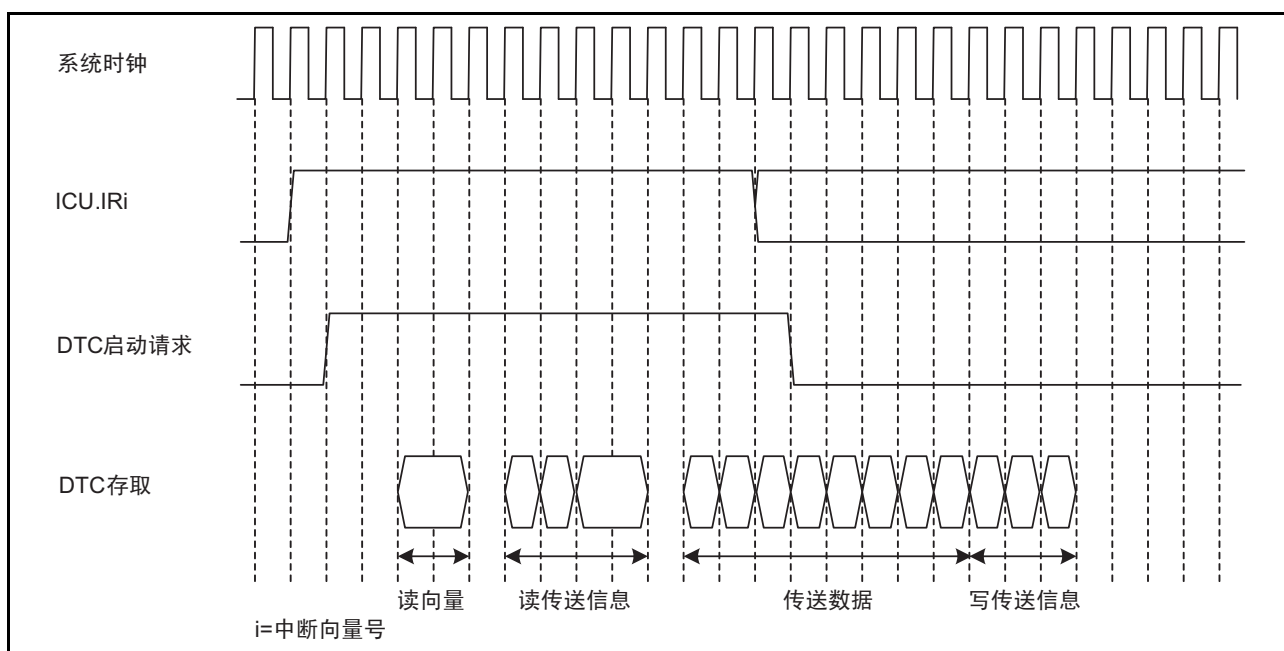


图 14.10 DTC 的运行时序例子 (2)  
(短地址模式、块传送模式、块大小为“4”的情况)



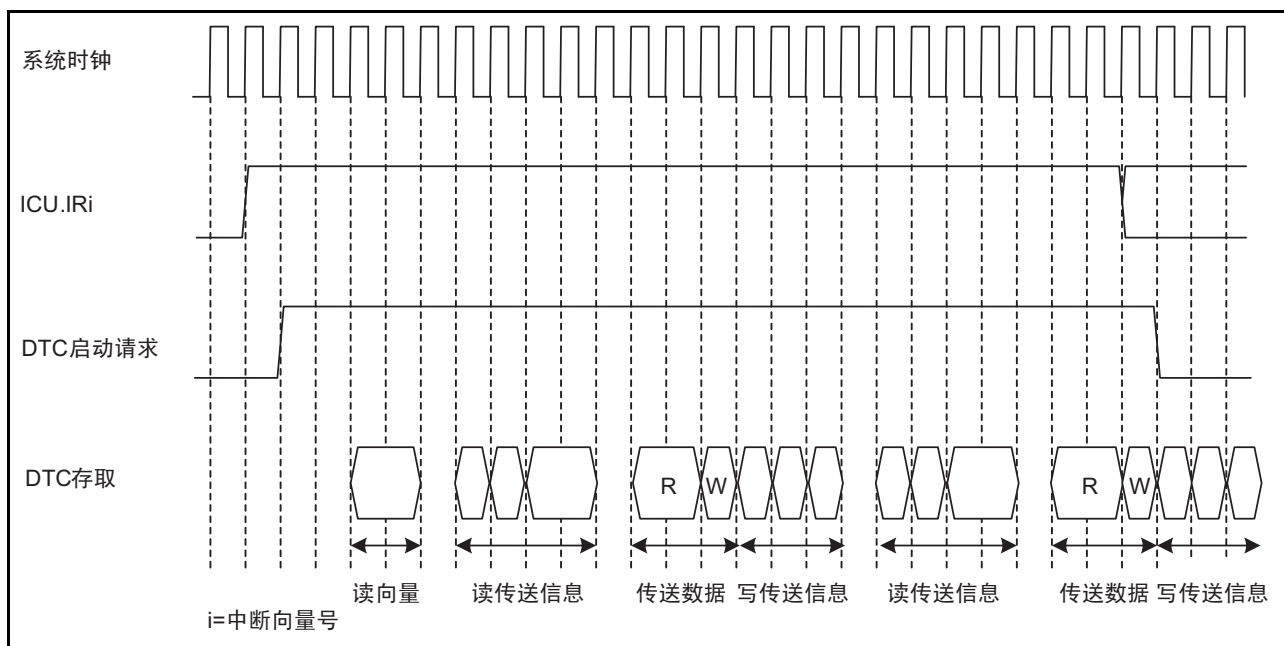


图 14.11 DTC 的运行时序例子 (3) (短地址模式、链传送的情况)

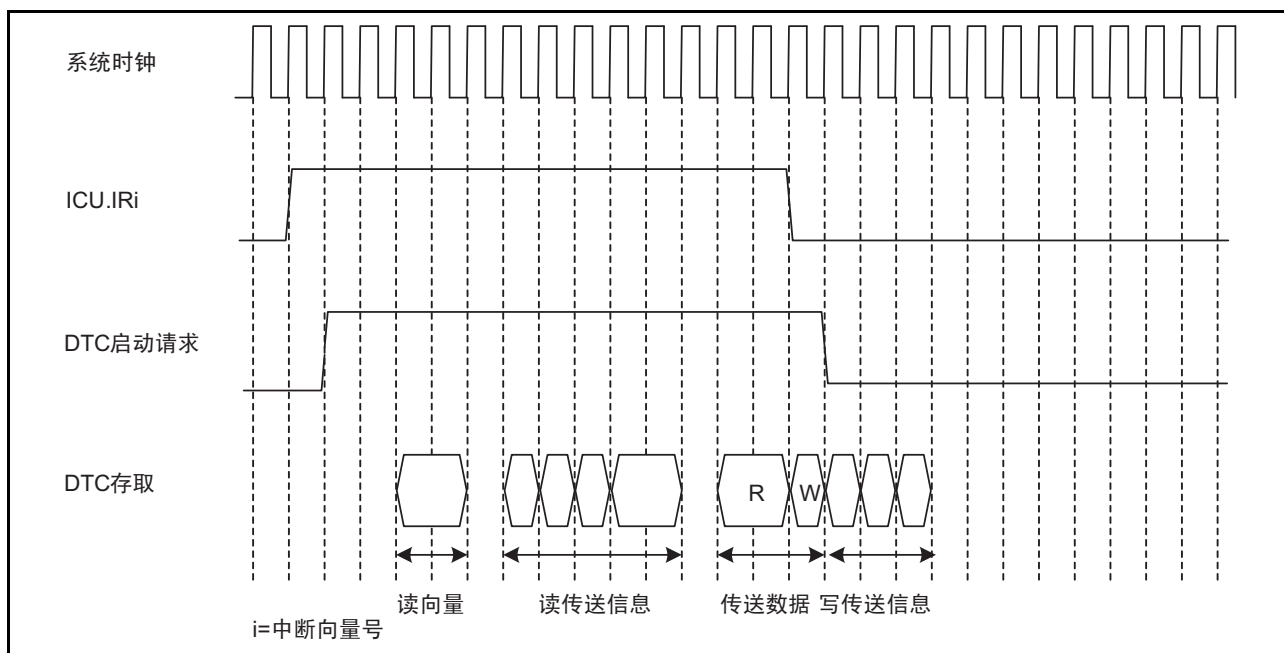


图 14.12 DTC 的运行时序例子 (4) (全地址模式、正常传送模式、重复传送模式的情况)

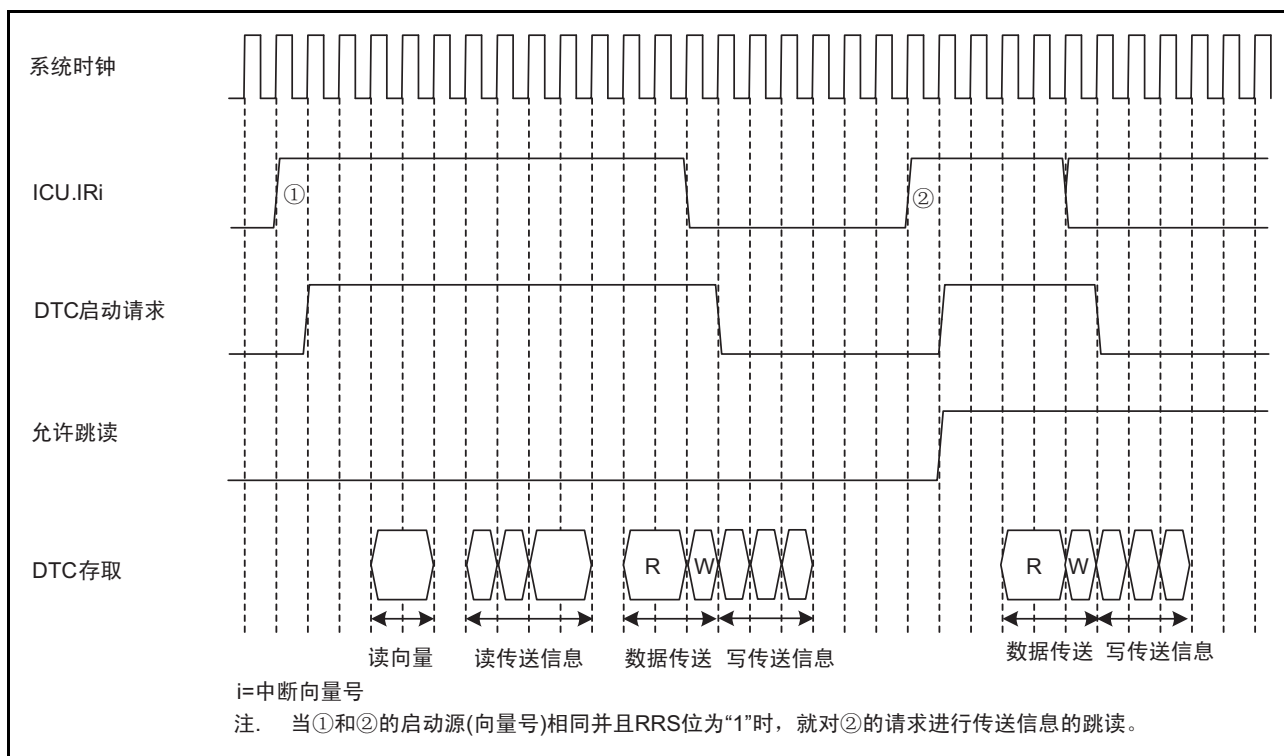


图 14.13 跳读传送信息的运行例子

### 14.4.8 DTC 的执行状态

DTC 的 1 次数据传送的执行状态如表 14.10 所示。

表 14.10 DTC 的执行状态

传送模式	读向量		读传送信息			写传送信息			读数据	写数据	内部运行	
	$Cv+1$	0 (注 1)	$4 \times Ci+1$ (注 2)	$3 \times Ci+1$ (注 3)	0 (注 1)	$3 \times Ci$ (注 4)	$2 \times Ci$ (注 5)	$Ci$ (注 6)	$Cr+1$	$Cw$	2	0 (注 1)
正常	$Cv+1$	0 (注 1)	$4 \times Ci+1$ (注 2)	$3 \times Ci+1$ (注 3)	0 (注 1)	$3 \times Ci$ (注 4)	$2 \times Ci$ (注 5)	$Ci$ (注 6)	$Cr+1$	$Cw$	2	0 (注 1)
重复	$Cv+1$	0 (注 1)	$4 \times Ci+1$ (注 2)	$3 \times Ci+1$ (注 3)	0 (注 1)	$3 \times Ci$ (注 4)	$2 \times Ci$ (注 5)	$Ci$ (注 6)	$Cr+1$	$Cw$	2	0 (注 1)
块 (注 7)	$Cv+1$	0 (注 1)	$4 \times Ci+1$ (注 2)	$3 \times Ci+1$ (注 3)	0 (注 1)	$3 \times Ci$ (注 4)	$2 \times Ci$ (注 5)	$Ci$ (注 6)	$P \times Cr$	$P \times Cw$	2	0 (注 1)

P: 块大小 (CRAH 寄存器和 CRAL 寄存器的设定值)

$Cv$ : 向量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

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。块大小等于“1”时为正常转送的周期数。

(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“30. RAM”、“31. ROM (保存代码的闪存)”和“5. I/O 寄存器”。)

各处理状态的执行顺序请参照“14.4.7 运行时序”

### 14.4.9 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。

## 14.5 DTC 的设定步骤

必须在使用 DTC 前设定 DTC 向量基址寄存器 (DTCVBR)。

必须按照图 14.14 的设定步骤进行 DTC 各启动源的设定。

必须将 DTC 模块启动位 (DTCST.DTCST 位) 置“1”。

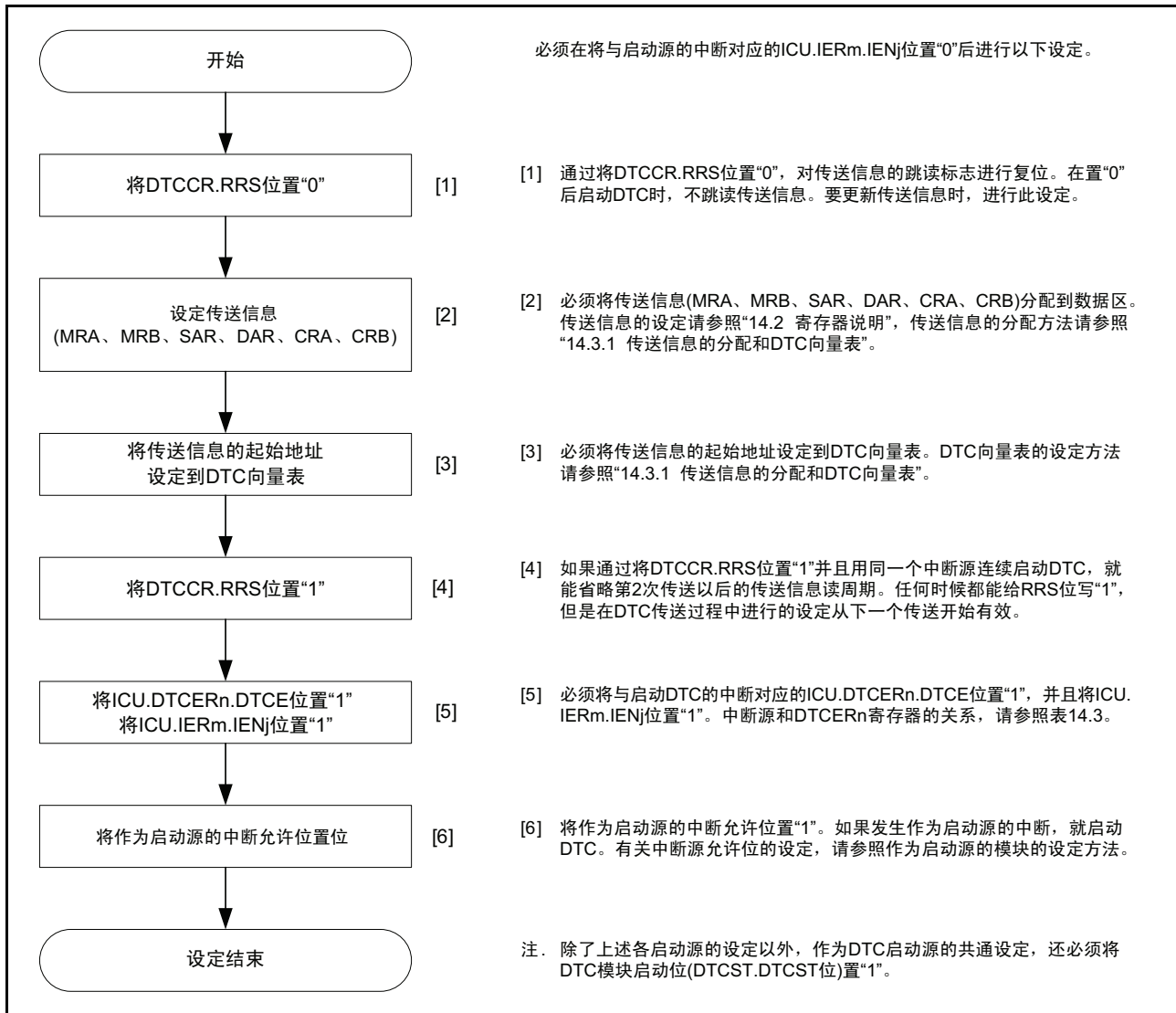


图 14.14 DTC 各启动源的设定步骤

## 14.6 DTC 的使用例子

### 14.6.1 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) 和字节传送 (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, MRB.DISEL 位 =0)，能给 MRB.DTS 位设定任意值；给 SAR 寄存器设定 SCIn.RDR 寄存器 (n=0 ~ 2) 的地址，给 DAR 寄存器设定用于保存数据的 RAM 起始地址，并且给 CRA 寄存器设定 128 (“0080h”)。必须给 CRB 寄存器设定 “FFFFh”。

#### (2) DTC 向量表的设定

将用于 RXI 中断的传送信息的起始地址设定到 DTC 向量表。

#### (3) ICU 的设定和 DTC 模块的启动

将对应的 ICU.DTCERn.DTCE 位置 “1”，将 ICU.IERm.IENj 位和 DTCST.DTCST 位置 “1”。

#### (4) SCI 的设定

将 SCIn.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 SCIn.RDR 寄存器传送到 RAM，并且进行 DAR 寄存器的递增和 CRA 寄存器的递减。

#### (6) 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。

### 14.6.2 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14.15所示。

1. 作为第1次数据传送，设定输入数据的正常传送模式。传送源地址必须固定，CRA寄存器为“0000h”（65536次），MRB.CHNE位为“1”（允许链传送），MRB.CHNS位为“1”（只在传送计数器为“0”时进行链传送），MRB.DISEL位为“0”（在指定的数据传送结束时向CPU请求中断）。
2. 必须在其他区域（ROM等）准备第1次数据传送的传送目标地址的每65536次的起始地址高8位。例如，当输入缓冲器为“20000h”~“21FFFFh”时，准备“21h”和“20h”。
3. 作为第2次数据传送，设定为用于重新设定第1次数据传送的传送目标地址的重复传送模式（传送源为重复区域），传送目标为第1次传送信息区的DAR寄存器的高8位。此时，必须将MRB.CHNE位置“0”（禁止链传送），并且将MRB.DISEL位置“0”（在指定的数据传送结束时向请求CPU中断）。当上述输入缓冲器为“20000h”~“21FFFFh”时，必须将传送计数器设定为“2”。
4. 通过中断进行65536次的第1次数据的传送。如果第1次数据传送的传送计数器变为“0”，就启动第2次数据传送。将第1次数据传送的传送源地址高8位设定为“21h”，第1次数据传送的传送目标地址低16位的传送计数器变为“0000h”。
5. 通过中断继续进行在第1次数据传送指定的65536次的第1次数据传送。如果第1次数据传送的传送计数器变为“0”，就启动第2次数据传送。将第1次数据传送的传送源地址高8位置“20h”，第1次的数据传送的传送目标地址低16位的传送计数器变为“0000h”。
6. 无限重复上述的第4步骤和第5步骤。因为第2次数据传送为重复传送模式，所以不向CPU请求中断。

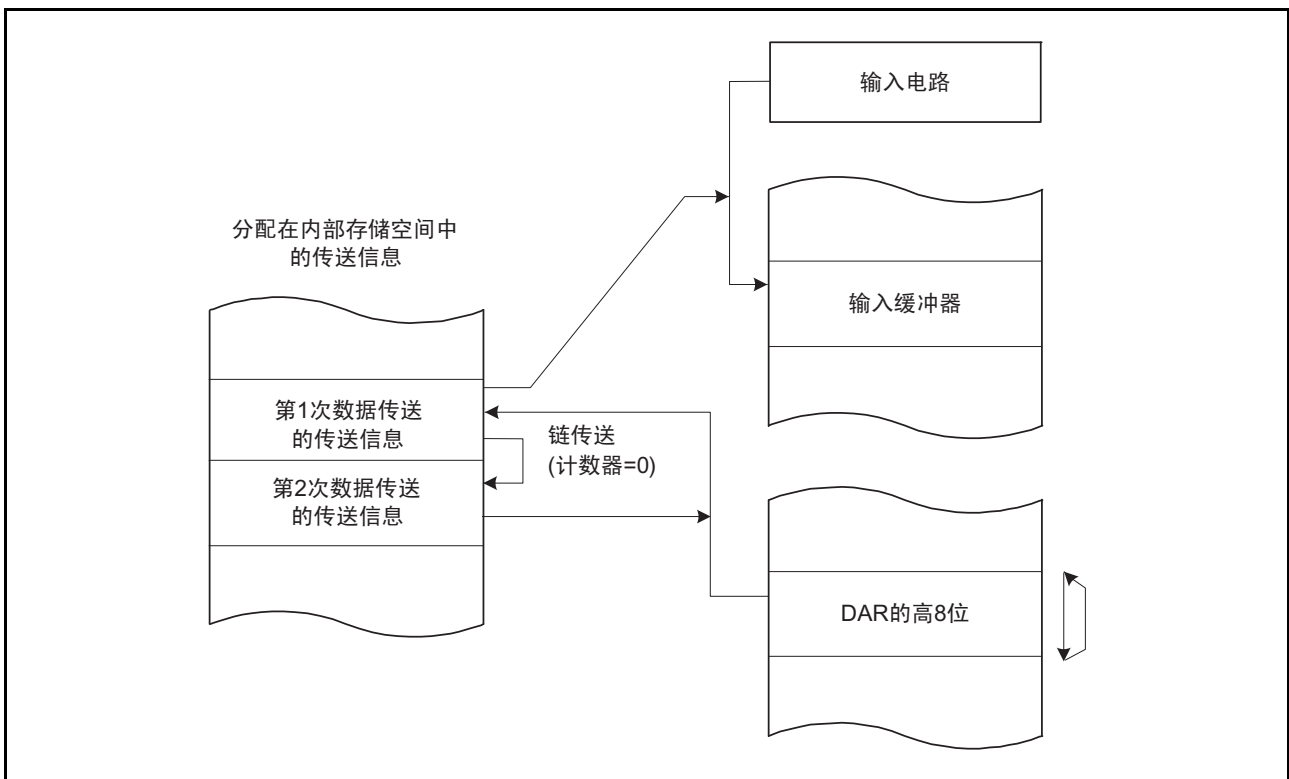


图 14.15 计数器为“0”时的链传送

## 14.7 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。

## 14.8 低功耗功能

在向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.DTCST 位置“0”（DTC 模块停止），然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”（向 DTC 模块停止状态转移），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。

通过将 MSTPCRA.MSTPA28 位置“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

在给 MSTPCRA.ACSE 位写“1”（允许全模块时钟停止模式）并且给包括 MSTPCRA.MSTPA28 位（向 DTC 模块停止状态转移的设定位）在内的 MSTPCRA.MSTPA<sub>n</sub> 和 MSTPCRB.MSTPB<sub>n</sub> 的全部位写“1”后，通过执行 WAIT 指令转移到全模块时钟停止模式。如果在执行 WAIT 指令时正在进行 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 模块停止状态。

### (3) 软件待机模式、深度软件待机模式

在给 SBYCR.SSBY 位写“1”（在执行 WAIT 指令后转移到软件待机模式）并且给 DPSBYCR.DPSBY 位写“0”（在执行 WAIT 指令后转移到软件待机模式）后，通过执行 WAIT 指令转移到软件待机模式。如果在执行 WAIT 指令时正在进行 DTC 传送，就在 DTC 传送结束后转移到软件待机模式。

如果在给 DPSBYCR.DPSBY 位写“1”（在执行 WAIT 指令后转移到深度软件待机模式）后执行 WAIT 指令，就转移到深度软件待机模式。

### (4) 低功耗功能的注意事项

有关 WAIT 指令和寄存器设定步骤，请参照“9.6.7 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，必须再次将 DTCST.DTCST 位置“1”。

## 14.9 使用时的注意事项

### 14.9.1 传送信息起始地址

必须给 DTC 向量表中指定的传送信息的起始地址指定地址  $4n$ ，否则地址的低 2 位就作为“00b”进行存取。

### 14.9.2 传送信息的分配

如图 14.16 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 的设定数据写到低位地址 0，将 CRB 的设定数据写到低位地址 2；在小端法的情况下，必须将 CRB 的设定数据写到低位地址 0，将 CRA 的设定数据写到低位地址 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、将 CRB 的设定数据分配到 32 位的 LSB 侧后写到低位地址 0。

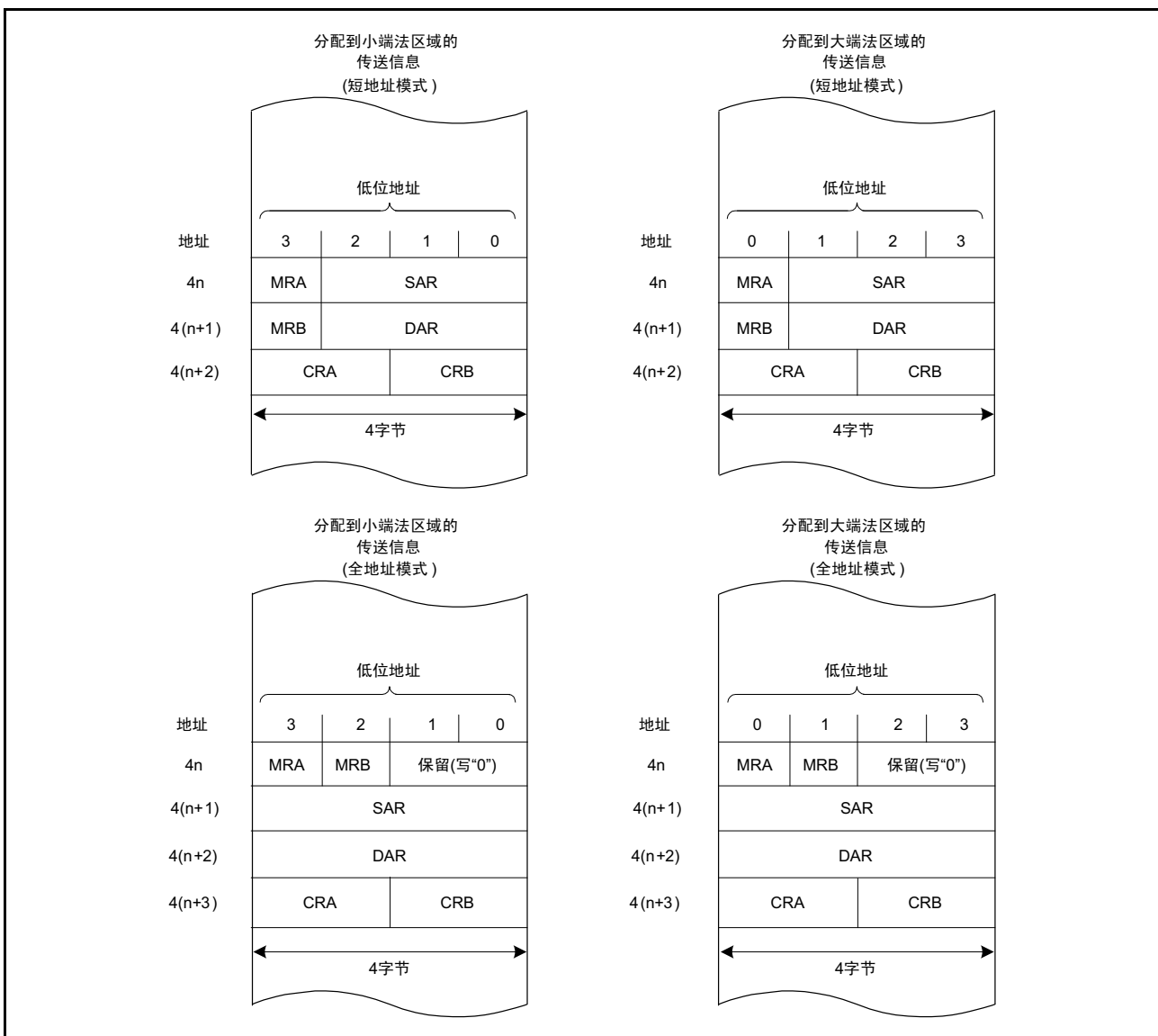


图 14.16 传送信息的分配



### 14.9.3 中断控制器的 DTC 启动允许寄存器 (ICU.DTCERn) 的设定

必须在 DTCST.DTCST 位为“0” (DTC 模块停止) 时设定 DTC 启动允许寄存器 (ICU.DTCERn 寄存器)。ICU.DTCERn 寄存器的详细内容请参照“11. 中断控制器 (ICU)”。

### 14.9.4 给 DTC 启动源指定通信功能中断的情况

RX62T 群在组合使用通信功能 (SCI/RIIC/RSPI) 和 DTC 功能时, 可能有注意事项。详细内容请参照“11.7 使用时的注意事项”。

## 15. I/O 端口

RX62T 群的 I/O 端口用作可编程输入 / 输出端口、外围功能的输入 / 输出和中断输入引脚。

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转换。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。

各端口有控制输入 / 输出的数据方向寄存器（DDR）、保存输出数据的数据寄存器（DR）、读引脚状态的端口寄存器（PORT）以及控制输入缓冲器的有效或者无效的输入缓冲控制寄存器（ICR）。

I/O 端口的结构因封装而不同，详细内容请参照各封装的章节。

112 引脚 LQFP 请参照“15.1 112 引脚 LQFP 的 I/O 端口”。

100 引脚 LQFP 请参照“15.2 100 引脚 LQFP 的 I/O 端口”。

80 引脚 LQFP 请参照“15.3 80 引脚 LQFP 的 I/O 端口”。

64 引脚 LQFP 请参照“15.4 64 引脚 LQFP 的 I/O 端口”。

### 15.1 112 引脚 LQFP 的 I/O 端口

RX62T 群的 112 引脚 LQFP 的 I/O 端口由端口 1 ~ 9、A、B、D、E、G 共 14 个端口构成，有 61 个输入 / 输出引脚。

#### 15.1.1 概要

I/O 端口的规格和端口功能一览表分别如表 15.1 和表 15.2 所示。

表 15.1 I/O 端口的规格（112 引脚 LQFP）

项目	内容
输入 / 输出引脚	61 个
输入引脚	21 个
端口	14 个端口（1 ~ 9、A、B、D、E、G）
漏极开路输出	2 个（RIIC）
大电流输出	12 个（MTU3 引脚、GPT 引脚）
施密特触发输入引脚	全部端口输入、CAN 输入、IRQ 输入、MTU3 输入、POE3 输入、RIIC 输入、SCI 输入、A/D 触发输入、NMI 输入、GPT 输入、LIN 输入
其他	<ul style="list-style-type: none"> <li>能驱动 1 个 TTL 负载和 30pF 的电容负载。</li> <li>在输出时驱动达林顿晶体管。</li> <li>能随时读引脚的状态。</li> </ul>

表 15.2 端口功能一览表 (1/3) (112 引脚 LQFP)

端口	概要	位	功能			CMOS 输入引脚	施密特触发 输入引脚	漏极开路 输出功能	大电流 输出引脚	
			输入 / 输出	输入	输出					
端口 1	与 MTU3 输入 和中断输入兼 用的通用输入 / 输出端口	0	P10	MTCLKD-B/IRQ0-A		—	全部输入功能	—	—	
		1	P11	MTCLKC-B/IRQ1-A			全部输入功能			
端口 2	与 MTU3 输入、 中断输入、 A/D 转换器输入、 RSPI 输入 / 输出、 LIN 输入 / 输出和 CAN 输入 / 输出 兼用的通用 输入 / 输出端口	0	P20	MTCLKB-B/IRQ7/ ADTRG0#-B		—	全部输入功能	—	—	
		1	P21	MTCLKA-B/IRQ6/ ADTRG1#-B			全部输入功能			
		2	P22/MISO-A	LRX/CRX-B/ ADTRG#		MISO-A	P22/LRX/CRX-B/ ADTRG#			
		3	P23/MOSI-A		LTX/CTX-B	MOSI-A	P23			
		4	P24/RSPCK-A			RSPCK-A	P24			
端口 3	与 MTU3 输入 / 输出和 RSPI 输入 / 输出兼用 的通用输入 / 输出 端口	0	P30/MTIOC0B-B/ SSL0-A	MTCLKD-A		SSL0-A	P30/MTIOC0B-B/ MTCLKD-A	—	—	
		1	P31/MTIOC0A-B	MTCLKC-A	SSL1-A	—	全部输入功能			
		2	P32/MTIOC3C	MTCLKB-A	SSL2-A		全部输入功能			
		3	P33/MTIOC3A	MTCLKA-A	SSL3-A		全部输入功能			
端口 4	与 A/D 转换器 输入兼用的通用 输入端口	0		P40/AN000		—	P40	—	—	
		1		P41/AN001			P41			
		2		P42/AN002			P42			
		3		P43/AN003/CVREFL			P43			
		4		P44/AN100			P44			
		5		P45/AN101			P45			
		6		P46/AN102			P46			
		7		P47/AN103/CVREFH			P47			
端口 5	与 A/D 转换器 输入兼用的通用 输入端口	0		P50/AN6		—	P50	—	—	
		1		P51/AN7			P51			
		2		P52/AN8			P52			
		3		P53/AN9			P53			
		4		P54/AN10			P54			
		5		P55/AN11			P55			
端口 6	与 A/D 转换器 输入兼用的通用 输入端口	0		P60/AN0		—	P60	—	—	
		1		P61/AN1			P61			
		2		P62/AN2			P62			
		3		P63/AN3			P63			
		4		P64/AN4			P64			
		5		P65/AN5			P65			
端口 7	与 POE3 输入、 中断输入、 MTU3 输入 / 输出 和 GPT 输入 / 输出 兼用的通用输入 / 输出 端口	0	P70	POE0#/IRQ5		—	全部输入功能	—	—	
		1	P71/MTIOC3B/ GTIOC0A-A				全部输入功能			○
		2	P72/MTIOC4A/ GTIOC1A-A				全部输入功能			○
		3	P73/MTIOC4B/ GTIOC2A-A				全部输入功能			○
		4	P74/MTIOC3D/ GTIOC0B-A				全部输入功能			○
		5	P75/MTIOC4C/ GTIOC1B-A				全部输入功能			○
		6	P76/MTIOC4D/ GTIOC2B-A				全部输入功能			○

表 15.2 端口功能一览表 (2/3) (112 引脚 LQFP)

端口	概要	位	功能			CMOS 输入引脚	施密特触发 输入引脚	漏极开路 输出功能	大电流 输出引脚		
			输入 / 输出	输入	输出						
端口 8	与 SCI 输入 / 输出和 MTU3 输入兼用的通用输入 / 输出端口	0	P80	RXD2-B/MTIC5W		—	全部输入功能	—	—		
		1	P81	MTIC5V	TXD2-B						
		2	P82/SCK2-B	MTIC5U							
端口 9	与 MTU3 输入 / 输出、POE3 输入和中断输入兼用的通用输入 / 输出端口	0	P90/MTIOC7D			—	全部输入功能	—	○		
		1	P91/MTIOC7C								
		2	P92/MTIOC6D								
		3	P93/MTIOC7B								
		4	P94/MTIOC7A								
		5	P95/MTIOC6B								
		6	P96	POE4#/IRQ4							
端口 A	与 MTU3 输入 / 输出、RSPI 输入 / 输出和 A/D 转换器输入兼用的通用输入 / 输出端口	0	PA0/MTIOC6C		SSL3-B	—	全部输入功能	—	—		
		1	PA1/MTIOC6A		SSL2-B						
		2	PA2/MTIOC2B		SSL1-B						
		3	PA3/MTIOC2A/ SSL0-B							SSL0-B	PA3/MTIOC2A
		4	PA4/MTIOC1B/ RSPCK-B	ADTRG0#-A						RSPCK-B	PA4/MTIOC1B/ ADTRG0#-A
		5	PA5/MTIOC1A/ MISO-B	ADTRG1#-A						MISO-B	PA5/MTIOC1A/ ADTRG1#-A
端口 B	与 MTU3 输入 / 输出、RSPI 输入 / 输出、RIIC 输入输出、SCI 输入 / 输出、GPT 输入、POE3 输入、中断输入和 CAN 输入 / 输出兼用的通用输入 / 输出端口	0	PB0/MTIOC0D/ MOSI-B			MOSI-B	全部输入功能	—	—		
		1	PB1/MTIOC0C/ SCL	RXD0		—	全部输入功能	○ (仅 SCL)	—		
		2	PB2/MTIOC0B-A/ SDA		TXD0						
		3	PB3/MTIOC0A-A/ SCK0								
		4	PB4	GTETRG/ POE8#/IRQ3							
		5	PB5		TXD2-A/ CTX-A/						
		6	PB6	RXD2-A/CRX-A							
		7	PB7/SCK2-A								
端口 D	与 GPT 输入 / 输出、RSPI 输入 / 输出、SCI 输入 / 输出和 CAN 输出兼用的通用输入 / 输出端口	0	PD0/GTIOC3B/ RSPCK-C							RSPCK-C	PD0/GTIOC3B
		1	PD1/GTIOC3A/ MISO-C			MISO-C	PD1/GTIOC3A				
		2	PD2/GTIOC2B-B/ MOSI-C			MOSI-C	PD2/GTIOC2B-B				
		3	PD3/GTIOC2A-B		TXD1	—	全部输入功能				
		4	PD4/GTIOC1B-B/ SCK1			—	全部输入功能				
		5	PD5/GTIOC1A-B	RXD1		—	全部输入功能				
		6	PD6/GTIOC0B-B/ SSL0-C			SSL0-C	PD6/GTIOC0B-B				
		7	PD7/GTIOC0A-B		SSL1-C/CTX-C	—	全部输入功能				

表 15.2 端口功能一览表 (3 / 3) (112 引脚 LQFP)

端口	概要	位	功能			CMOS 输入引脚	施密特触发 输入引脚	漏极开路 输出功能	大电流 输出引脚
			输入 / 输出	输入	输出				
端口 E	与 CAN 输入、 RSPI 输出、 POE3 输入、 MTU3 输入和 中断输入兼用 的通用输入 / 输 出端口	0	PE0	CRX-C	SSL2-C	—	全部输入功能	—	—
		1	PE1		SSL3-C		全部输入功能		
		2		PE2/POE10#-A/NMI			全部输入功能		
		3	PE3	MTCLKD-C/ POE11#/IRQ2-A			全部输入功能		
		4	PE4	MTCLKC-C/ POE10#-B/ IRQ1-B			全部输入功能		
		5	PE5	IRQ0-B			全部输入功能		
端口 G	与中断输入和 跟踪输出兼用 的通用输入 / 输 出端口	0	PG0	IRQ0-C	TRSYNC	—	全部输入功能	—	—
		1	PG1	IRQ1-C	TRDATA0		全部输入功能		
		2	PG2	IRQ2-B	TRDATA1		全部输入功能		
		3	PG3		TRDATA2		全部输入功能		
		4	PG4		TRDATA3		全部输入功能		
		5	PG5		TRCLK		全部输入功能		

### 15.1.2 寄存器说明

I/O 端口的寄存器一览表和各寄存器的有效位一览表分别如表 15.3 和表 15.4 所示。

表 15.3 I/O 端口的寄存器一览表 (1/2) (112 引脚 LQFP)

端口符号	寄存器名	寄存器符号	复位后的值	地址	存取长度
PORT1	数据方向寄存器	DDR	00h	0008 C001h	8
	数据寄存器	DR	00h	0008 C021h	8
	端口寄存器	PORT	不定值	0008 C041h	8
	输入缓冲控制寄存器	ICR	00h	0008 C061h	8
PORT2	数据方向寄存器	DDR	00h	0008 C002h	8
	数据寄存器	DR	00h	0008 C022h	8
	端口寄存器	PORT	不定值	0008 C042h	8
	输入缓冲控制寄存器	ICR	00h	0008 C062h	8
PORT3	数据方向寄存器	DDR	00h	0008 C003h	8
	数据寄存器	DR	00h	0008 C023h	8
	端口寄存器	PORT	不定值	0008 C043h	8
	输入缓冲控制寄存器	ICR	00h	0008 C063h	8
PORT4	端口寄存器	PORT	不定值	0008 C044h	8
	输入缓冲控制寄存器	ICR	00h	0008 C064h	8
PORT5	端口寄存器	PORT	不定值	0008 C045h	8
	输入缓冲控制寄存器	ICR	00h	0008 C065h	8
PORT6	端口寄存器	PORT	不定值	0008 C046h	8
	输入缓冲控制寄存器	ICR	00h	0008 C066h	8
PORT7	数据方向寄存器	DDR	00h	0008 C007h	8
	数据寄存器	DR	00h	0008 C027h	8
	端口寄存器	PORT	不定值	0008 C047h	8
	输入缓冲控制寄存器	ICR	00h	0008 C067h	8
PORT8	数据方向寄存器	DDR	00h	0008 C008h	8
	数据寄存器	DR	00h	0008 C028h	8
	端口寄存器	PORT	不定值	0008 C048h	8
	输入缓冲控制寄存器	ICR	00h	0008 C068h	8
PORT9	数据方向寄存器	DDR	00h	0008 C009h	8
	数据寄存器	DR	00h	0008 C029h	8
	端口寄存器	PORT	不定值	0008 C049h	8
	输入缓冲控制寄存器	ICR	00h	0008 C069h	8
PORTA	数据方向寄存器	DDR	00h	0008 C00Ah	8
	数据寄存器	DR	00h	0008 C02Ah	8
	端口寄存器	PORT	不定值	0008 C04Ah	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Ah	8
PORTB	数据方向寄存器	DDR	00h	0008 C00Bh	8
	数据寄存器	DR	00h	0008 C02Bh	8
	端口寄存器	PORT	不定值	0008 C04Bh	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Bh	8

表 15.3 I/O 端口的寄存器一览表 (2/2) (112 引脚 LQFP)

端口符号	寄存器名	寄存器符号	复位后的值	地址	存取长度
PORTD	数据方向寄存器	DDR	00h	0008 C00Dh	8
	数据寄存器	DR	00h	0008 C02Dh	8
	端口寄存器	PORT	不定值	0008 C04Dh	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Dh	8
PORTE	数据方向寄存器	DDR	00h	0008 C00Eh	8
	数据寄存器	DR	00h	0008 C02Eh	8
	端口寄存器	PORT	不定值	0008 C04Eh	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Eh	8
PORTG	数据方向寄存器	DDR	00h	0008 C010h	8
	数据寄存器	DR	00h	0008 C030h	8
	端口寄存器	PORT	不定值	0008 C050h	8
	输入缓冲控制寄存器	ICR	00h	0008 C070h	8
IOPORT	端口功能寄存器 8	PF8IRQ	00h	0008 C108h	8
	端口功能寄存器 9	PF9IRQ	00h	0008 C109h	8
	端口功能寄存器 A	PFAADC	00h	0008 C10Ah	8
	端口功能寄存器 C	PFCMTU	00h	0008 C10Ch	8
	端口功能寄存器 D	PDFMTU	00h	0008 C10Dh	8
	端口功能寄存器 F	PFSCI	00h	0008 C10Fh	8
	端口功能寄存器 G	PFSPi	00h	0008 C110h	8
	端口功能寄存器 H	PFHSPI	00h	0008 C111h	8
	端口功能寄存器 J	PFJCAN	00h	0008 C113h	8
	端口功能寄存器 K	PFKLIN	00h	0008 C114h	8
	端口功能寄存器 M	PFMPOE	00h	0008 C116h	8
	端口功能寄存器 N	PFNPOE	00h	0008 C117h	8

表 15.4 各寄存器的有效位一览表 (1/2) (112 引脚 LQFP)

寄存器符号	b7	b6	b5	b4	b3	b2	b1	b0
PORT1.DDR	x	x	x	x	x	x	○	○
PORT2.DDR	x	x	x	○	○	○	○	○
PORT3.DDR	x	x	x	x	○	○	○	○
PORT7.DDR	x	○	○	○	○	○	○	○
PORT8.DDR	x	x	x	x	x	○	○	○
PORT9.DDR	x	○	○	○	○	○	○	○
PORTA.DDR	x	x	○	○	○	○	○	○
PORTB.DDR	○	○	○	○	○	○	○	○
PORTD.DDR	○	○	○	○	○	○	○	○
PORTE.DDR	x	x	○	○	○	x	○	○
PORTG.DDR	x	x	○	○	○	○	○	○
PORT1.DR	x	x	x	x	x	x	○	○
PORT2.DR	x	x	x	○	○	○	○	○
PORT3.DR	x	x	x	x	○	○	○	○
PORT7.DR	x	○	○	○	○	○	○	○
PORT8.DR	x	x	x	x	x	○	○	○
PORT9.DR	x	○	○	○	○	○	○	○
PORTA.DR	x	x	○	○	○	○	○	○
PORTB.DR	○	○	○	○	○	○	○	○
PORTD.DR	○	○	○	○	○	○	○	○
PORTE.DR	x	x	○	○	○	x	○	○
PORTG.DR	x	x	○	○	○	○	○	○
PORT1.PORT	x	x	x	x	x	x	○	○
PORT2.PORT	x	x	x	○	○	○	○	○
PORT3.PORT	x	x	x	x	○	○	○	○
PORT4.PORT	○	○	○	○	○	○	○	○
PORT5.PORT	x	x	○	○	○	○	○	○
PORT6.PORT	x	x	○	○	○	○	○	○
PORT7.PORT	x	○	○	○	○	○	○	○
PORT8.PORT	x	x	x	x	x	○	○	○
PORT9.PORT	x	○	○	○	○	○	○	○
PORTA.PORT	x	x	○	○	○	○	○	○
PORTB.PORT	○	○	○	○	○	○	○	○
PORTD.PORT	○	○	○	○	○	○	○	○
PORTE.PORT	x	x	○	○	○	○	○	○
PORTG.PORT	x	x	○	○	○	○	○	○
PORT1.ICR	x	x	x	x	x	x	○	○
PORT2.ICR	x	x	x	○	○	○	○	○
PORT3.ICR	x	x	x	x	○	○	○	○
PORT4.ICR	○	○	○	○	○	○	○	○
PORT5.ICR	x	x	○	○	○	○	○	○
PORT6.ICR	x	x	○	○	○	○	○	○
PORT7.ICR	x	○	○	○	○	○	○	○
PORT8.ICR	x	x	x	x	x	○	○	○



表 15.4 各寄存器的有效位一览表 (2/2) (112 引脚 LQFP)

寄存器符号	b7	b6	b5	b4	b3	b2	b1	b0
PORT9.ICR	x	○	○	○	○	○	○	○
PORTA.ICR	x	x	○	○	○	○	○	○
PORTB.ICR	○	○	○	○	○	○	○	○
PORTD.ICR	○	○	○	○	○	○	○	○
PORTE.ICR	x	x	○	○	○	x	○	○
PORTG.ICR	x	x	○	○	○	○	○	○
IOPORT.PF8IRQ	x	x	x	x	○	○	○	○
IOPORT.PF9IRQ	x	x	x	x	x	○	x	x
IOPORT.PFAADC	x	x	x	x	x	x	○	○
IOPORT.PFCMTU	○	○	x	x	x	x	○	○
IOPORT.PFDGPT	x	x	x	x	x	x	x	○
IOPORT.PFFSCI	x	x	x	x	x	○	x	x
IOPORT.PFGSPI	○	○	○	○	○	○	○	x
IOPORT.PFHSPi	x	x	x	x	x	x	○	○
IOPORT.PFJCAN	○	○	x	x	x	x	x	○
IOPORT.PFKLIN	x	x	x	x	x	x	x	○
IOPORT.PFMPOE	x	x	x	○	○	○	○	○
IOPORT.PFNPOE	○	x	x	x	x	x	x	x

○: 有效位、x: 无效位 (保留位)

## 15.1.2.1 数据方向寄存器 (DDR)

地址 PORT1.DDR 0008 C001h、PORT2.DDR 0008 C002h、PORT3.DDR 0008 C003h、PORT7.DDR 0008 C007h、PORT8.DDR 0008 C008h、PORT9.DDR 0008 C009h、PORTA.DDR 0008 C00Ah、PORTB.DDR 0008 C00Bh、PORTD.DDR 0008 C00Dh、PORTE.DDR 0008 C00Eh、PORTG.DDR 0008 C010h

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

- 注 1. PORT1.DDR 的低 2 位有效，高 6 位是保留位。  
 PORT2.DDR 的低 5 位有效，高 3 位是保留位。  
 PORT3.DDR 的低 4 位有效，高 4 位是保留位。  
 PORT7.DDR 的低 7 位有效，高 1 位是保留位。  
 PORT8.DDR 的低 3 位有效，高 5 位是保留位。  
 PORT9.DDR 的低 7 位有效，高 1 位是保留位。  
 PORTA.DDR 的低 6 位有效，高 2 位是保留位。  
 PORTE.DDR 的 b5 ~ b3、b1、b0 有效，b7、b6、b2 是保留位。  
 PORTG.DDR 的低 6 位有效，高 2 位是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0	Pn0 输入 / 输出指定位	0: 输入端口 1: 输出端口	R/W
b1	B1	Pn1 输入 / 输出指定位		R/W
b2	B2	Pn2 输入 / 输出指定位		R/W
b3	B3	Pn3 输入 / 输出指定位		R/W
b4	B4	Pn4 输入 / 输出指定位		R/W
b5	B5	Pn5 输入 / 输出指定位		R/W
b6	B6	Pn6 输入 / 输出指定位		R/W
b7	B7	Pn7 输入 / 输出指定位		R/W

n=1 ~ 3、7 ~ 9、A、B、D、E、G

PORTn.DDR 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。

PORTn.DDR 寄存器 (n=1 ~ 3、7 ~ 9、A、B、D、E、G) 的各位分别对应端口 n 的各引脚，并且能以 1 位为单位进行指定。

## 15.1.2.2 数据寄存器 (DR)

地址 PORT1.DR 0008 C021h、PORT2.DR 0008 C022h、PORT3.DR 0008 C023h、PORT7.DR 0008 C027h、PORT8.DR 0008 C028h、PORT9.DR 0008 C029h、PORTA.DR 0008 C02Ah、PORTB.DR 0008 C02Bh、PORTD.DR 0008 C02Dh、PORTE.DR 0008 C02Eh、PORTG.DR 0008 C030h

	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

- 注 1. PORT1.DR 的低 2 位有效，高 6 位是保留位。  
 PORT2.DR 的低 5 位有效，高 3 位是保留位。  
 PORT3.DR 的低 4 位有效，高 4 位是保留位。  
 PORT7.DR 的低 7 位有效，高 1 位是保留位。  
 PORT8.DR 的低 3 位有效，高 5 位是保留位。  
 PORT9.DR 的低 7 位有效，高 1 位是保留位。  
 PORTA.DR 的低 6 位有效，高 2 位是保留位。  
 PORTE.DR 的 b5 ~ b3、b1、b0 有效，b7、b6、b2 是保留位。  
 PORTG.DR 的低 6 位有效，高 2 位是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0	Pn0 输出数据保存位	保存输出数据。	R/W
b1	B1	Pn1 输出数据保存位		R/W
b2	B2	Pn2 输出数据保存位		R/W
b3	B3	Pn3 输出数据保存位		R/W
b4	B4	Pn4 输出数据保存位		R/W
b5	B5	Pn5 输出数据保存位		R/W
b6	B6	Pn6 输出数据保存位		R/W
b7	B7	Pn7 输出数据保存位		R/W

n=1 ~ 3、7 ~ 9、A、B、D、E、G

PORTn.DR 寄存器 (n=1 ~ 3、7 ~ 9、A、B、D、E、G) 保存用作通用输出端口的引脚输出数据。

## 15.1.2.3 端口寄存器 (PORT)

地址 PORT1.PORT 0008 C041h、PORT2.PORT 0008 C042h、PORT3.PORT 0008 C043h、PORT4.PORT 0008 C044h、  
PORT5.PORT 0008 C045h、PORT6.PORT 0008 C046h、PORT7.PORT 0008 C047h、PORT8.PORT 0008 C048h、  
PORT9.PORT 0008 C049h、PORTA.PORT 0008 C04Ah、PORTB.PORT 0008 C04Bh、PORTD.PORT 0008 C04Dh、  
PORTE.PORT 0008 C04Eh、PORTG.PORT 0008 C050h

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

复位后的值 x x x x x x x x

- 注 1. PORT1.PORT 的低 2 位有效，高 6 位是保留位。  
PORT2.PORT 的低 5 位有效，高 3 位是保留位。  
PORT3.PORT 的低 4 位有效，高 4 位是保留位。  
PORT5.PORT 的低 6 位有效，高 2 位是保留位。  
PORT6.PORT 的低 6 位有效，高 2 位是保留位。  
PORT7.PORT 的低 7 位有效，高 1 位是保留位。  
PORT8.PORT 的低 3 位有效，高 5 位是保留位。  
PORT9.PORT 的低 7 位有效，高 1 位是保留位。  
PORTA.PORT 的低 6 位有效，高 2 位是保留位。  
PORTE.PORT 的低 6 位有效，高 2 位是保留位。  
PORTG.PORT 的低 6 位有效，高 2 位是保留位。

- 注 2. 保留位。读取值为“1”，写操作无效。

位	符号	位名	功能	R/W
b0	B0	Pn0 位	反映端口的引脚状态。	R
b1	B1	Pn1 位		R
b2	B2	Pn2 位		R
b3	B3	Pn3 位		R
b4	B4	Pn4 位		R
b5	B5	Pn5 位		R
b6	B6	Pn6 位		R
b7	B7	Pn7 位		R

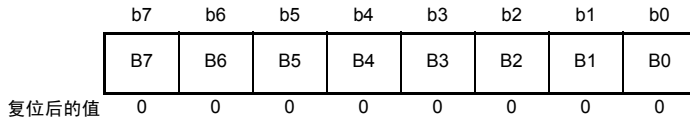
n=1 ~ 9、A、B、D、E、G

PORTn.PORT 寄存器是反映端口引脚状态的寄存器。

如果读 PORTn.PORT 寄存器 (n=1 ~ 9、A、B、D、E、G)，就能读到引脚的状态。

15.1.2.4 输入缓冲控制寄存器 (ICR)

地址 PORT1.ICR 0008 C061h、PORT2.ICR 0008 C062h、PORT3.ICR 0008 C063h、PORT4.ICR 0008 C064h、PORT5.ICR 0008 C065h、PORT6.ICR 0008 C066h、PORT7.ICR 0008 C067h、PORT8.ICR 0008 C068h、PORT9.ICR 0008 C069h、PORTA.ICR 0008 C06Ah、PORTB.ICR 0008 C06Bh、PORTD.ICR 0008 C06Dh、PORTE.ICR 0008 C06Eh、PORTG.ICR 0008 C070h



- 注 1. PORT1.ICR 的低 2 位有效，高 6 位是保留位。  
 PORT2.ICR 的低 5 位有效，高 3 位是保留位。  
 PORT3.ICR 的低 4 位有效，高 4 位是保留位。  
 PORT5.ICR 的低 6 位有效，高 2 位是保留位。  
 PORT6.ICR 的低 6 位有效，高 2 位是保留位。  
 PORT7.ICR 的低 7 位有效，高 1 位是保留位。  
 PORT8.ICR 的低 3 位有效，高 5 位是保留位。  
 PORT9.ICR 的低 7 位有效，高 1 位是保留位。  
 PORTA.ICR 的低 6 位有效，高 2 位是保留位。  
 PORTE.ICR 的 b5 ~ b3、b1、b0 有效，b7、b6、b2 是保留位。  
 PORTG.ICR 的低 6 位有效，高 2 位是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0 (注 1)	Pn0 输入缓冲器控制位	0: 对应引脚的输入缓冲器无效 1: 对应引脚的输入缓冲器有效	R/W
b1	B1 (注 1)	Pn1 输入缓冲器控制位		R/W
b2	B2 (注 1)	Pn2 输入缓冲器控制位		R/W
b3	B3 (注 1)	Pn3 输入缓冲器控制位		R/W
b4	B4 (注 1)	Pn4 输入缓冲器控制位		R/W
b5	B5 (注 1)	Pn5 输入缓冲器控制位		R/W
b6	B6 (注 1)	Pn6 输入缓冲器控制位		R/W
b7	B7 (注 1)	Pn7 输入缓冲器控制位		R/W

n=1 ~ 9、A、B、D、E、G

注 1. 在用作输入引脚时，必须将对应位置“1”。必须将不用作输入的引脚和用作模拟输入 / 输出引脚的位置“0”。

PORTn.ICR 寄存器是控制端口输入缓冲器的寄存器。

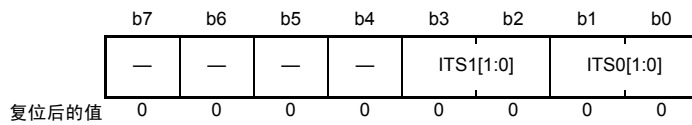
PORTn.ICR 寄存器 (n=1 ~ 9、A、B、D、E、G) 的各位分别对应端口 n 的各引脚，并且能以 1 位为单位进行指定。

在用作外围模块的输入引脚时，为了使对应引脚的输入缓冲器有效，需要预先将 PORTn.ICR 置“1”。如果在将 PORTn.ICR 置“0”的状态下用作外围模块的输入引脚，就将输入到外围模块的信号固定为 High 电平。

在更改 PORTn.ICR 寄存器的设定时，有可能因引脚的状态而在内部产生边沿，因此必须在不使用该输入引脚时更改 PORTn.ICR 寄存器的设定。例如，在 IRQ<sub>i</sub> (i=0 ~ 7) 输入时，必须在禁止该中断的状态下更改 PORTn.ICR 寄存器的设定。先将中断控制器的 IR<sub>i</sub>.IR 标志 (i=64 ~ 71 (IRQ 的中断向量号)) 置“0”，然后允许该中断。如果在更改 PORTn.ICR 寄存器的设定后产生边沿，就必须取消该边沿。

## 15.1.2.5 端口功能寄存器 8 (PF8IRQ)

地址 0008 C108h



位	符号	位名	功能	R/W
b1-b0	ITS0[1:0]	IRQ0 引脚选择位	b1 b0 0 0: 将 P10 设定为 IRQ0-A 输入引脚 0 1: 将 PE5 设定为 IRQ0-B 输入引脚 1 0: 将 PG0 设定为 IRQ0-C 输入引脚 1 1: 不能设定	R/W
b3-b2	ITS0[1:0]	IRQ1 引脚选择位	b3 b2 0 0: 将 P11 设定为 IRQ1-A 输入引脚 0 1: 将 PE4 设定为 IRQ1-B 输入引脚 1 0: 将 PG1 设定为 IRQ1-C 输入引脚 1 1: 不能设定	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

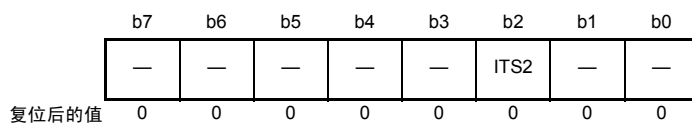
PF8IRQ 寄存器是选择 IRQ0、IRQ1 输入引脚的寄存器。

ITS<sub>i</sub> 位 (IRQ<sub>i</sub> 引脚选择位) (i=0、1)

此位选择 IRQ<sub>i</sub> 的输入引脚。

## 15.1.2.6 端口功能寄存器 9 (PF9IRQ)

地址 0008 C109h



位	符号	位名	功能	R/W
b1-b0	—	保留位	读写值都为“0”。	R/W
b2	ITS2	IRQ2 引脚选择位	0: 将 PE3 设定为 IRQ2-A 输入引脚 1: 将 PG2 设定为 IRQ2-B 输入引脚	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

PF9IRQ 寄存器是选择 IRQ2 输入引脚的寄存器。

ITS2 位 (IRQ2 引脚选择位)

此位选择 IRQ2 的输入引脚。

## 15.1.2.7 端口功能寄存器 A (PFAADC)

地址 0008 C10Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	ADTRG1S	ADTRG0S
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	ADTRG0S	ADTRG0# 输入选择位	0: 将 PA4 设定为 ADTRG0#-A 输入引脚 1: 将 P20 设定为 ADTRG0#-B 输入引脚	R/W
b1	ADTRG1S	ADTRG0# 输入选择位	0: 将 PA5 设定为 ADTRG1#-A 输入引脚 1: 将 P21 设定为 ADTRG1#-B 输入引脚	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

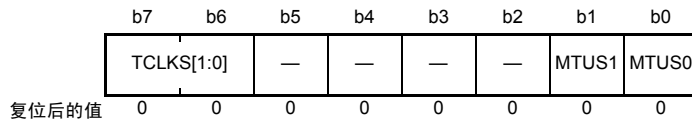
PFAADC 寄存器是选择 ADTRG0#、ADTRG1# 引脚的寄存器。

ADTRGnS 位 (ADTRGn# 输入选择位) (n=0、1)

此位选择 ADTRGn# 的输入引脚。

## 15.1.2.8 端口功能寄存器 C (PFCMTU)

地址 0008 C10Ch



位	符号	位名	功能	R/W
b0	MTUS0	MTU3 引脚选择 0 位	0: 选择 PB3 为 MTIOC0A-A 引脚 1: 选择 P31 为 MTIOC0A-B 引脚	R/W
b2	MTUS1	MTU3 引脚选择 1 位	0: 选择 PB2 为 MTIOC0B-A 引脚 1: 选择 P30 为 MTIOC0B-B 引脚	R/W
b5-b2	—	保留位	读写值都为“0”。	R/W
b7-b6	TCLKS[1:0]	MTCLK 引脚选择位	b7 b6 00: 将 P33 设定为 MTCLKA-A 输入引脚 将 P32 设定为 MTCLKB-A 输入引脚 将 P31 设定为 MTCLKC-A 输入引脚 将 P30 设定为 MTCLKD-A 输入引脚 01: 将 P21 设定为 MTCLKA-B 输入引脚 将 P20 设定为 MTCLKB-B 输入引脚 将 P11 设定为 MTCLKC-B 输入引脚 将 P10 设定为 MTCLKD-B 输入引脚 10: 将 PE4 设定为 MTCLKC-C 输入引脚 将 PE3 设定为 MTCLKD-C 输入引脚 (不能选择 MTCLKA 引脚和 MTCLKB 引脚) 11: 不能设定	R/W

PFCMTU 寄存器是选择 MTU3 引脚的寄存器。

MTUS<sub>i</sub> 位 (MTU3 引脚选择位) (i=0、1)

此位选择 MTU3 的输入 / 输出引脚。

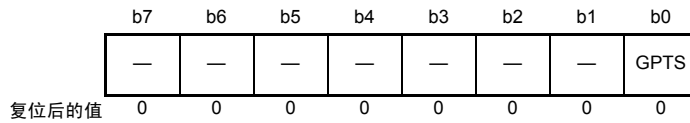
TCLKS[1:0] 位 (MTCLK 引脚选择位)

此位选择 MTU3 的 MTCLK 输入引脚。



## 15.1.2.9 端口功能寄存器 D (PFDGPT)

地址 0008 C10Dh



位	符号	位名	功能	R/W
b0	GPTS	GPT 引脚选择位	0: 选择 P71 为 GTIOC0A-A 引脚 选择 P74 为 GTIOC0B-A 引脚 选择 P72 为 GTIOC1A-A 引脚 选择 P75 为 GTIOC1B-A 引脚 选择 P73 为 GTIOC2A-A 引脚 选择 P76 为 GTIOC2B-A 引脚 1: 选择 PD7 为 GTIOC0A-B 引脚 选择 PD6 为 GTIOC0B-B 引脚 选择 PD5 为 GTIOC1A-B 引脚 选择 PD4 为 GTIOC1B-B 引脚 选择 PD3 为 GTIOC2A-B 引脚 选择 PD2 为 GTIOC2B-B 引脚	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

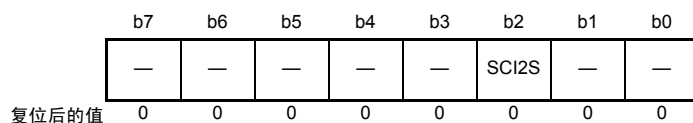
PFDGPT 寄存器是选择 GPT 引脚的寄存器。

## GPTS 位 (GPT 引脚选择位)

此位选择 GPT 的输入 / 输出引脚。

## 15.1.2.10 端口功能寄存器 F (PFFSCI)

地址 0008 C10Fh



位	符号	位名	功能	R/W
b1-b0	—	保留位	读写值都为“0”。	R/W
b2	SCI2S	SCI2 输入 / 输出引脚选择位	0: 将 PB6 设定为 RXD2-A 引脚 将 PB7 设定为 SCK2-A 引脚 将 PB5 设定为 TXD2-A 引脚 1: 将 P80 设定为 RXD2-B 引脚 将 P82 设定为 SCK2-B 引脚 将 P81 设定为 TXD2-B 引脚	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

PFFSCI 寄存器是选择 SCI2 引脚的寄存器。

## SCI2S 位 (SCI2 输入 / 输出引脚选择位)

此位选择 SCI 通道 2 的输入 / 输出引脚。

## 15.1.2.11 端口功能寄存器 G (PFGSPI)

地址 0008 C110h

b7	b6	b5	b4	b3	b2	b1	b0
SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	—
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	RSPCKE	RSPCK 输出允许位	0: RSPCK 引脚无效 1: RSPCK 引脚有效	R/W
b2	MOSIE	MOSI 输出允许位	0: MOSI 引脚无效 1: MOSI 引脚有效	R/W
b3	MISOE	MISO 输出允许位	0: MISO 引脚无效 1: MISO 引脚有效	R/W
b4	SSL0E	SSL0 输出允许位	0: SSL0 引脚无效 1: SSL0 引脚有效	R/W
b5	SSL1E	SSL1 输出允许位	0: SSL1 引脚无效 1: SSL1 引脚有效	R/W
b6	SSL2E	SSL2 输出允许位	0: SSL2 引脚无效 1: SSL2 引脚有效	R/W
b7	SSL3E	SSL3 输出允许位	0: SSL3 引脚无效 1: SSL3 引脚有效	R/W

PFGSPI 寄存器是设定 RSPCK 的相关输入 / 输出引脚的寄存器。

**RSPCKE 位 (RSPCK 输出允许位)**

此位选择允许或者禁止 RSPCK 引脚的输出。在使用 RSPCK 引脚时，必须将此位置“1”。

**MOSIE 位 (MOSI 输出允许位)**

此位选择允许或者禁止 MOSI 引脚的输出。在使用 MOSI 引脚时，必须将此位置“1”。

**MISOE 位 (MISO 输出允许位)**

此位选择允许或者禁止 MISO 引脚的输出。在使用 MISO 引脚时，必须将此位置“1”。

**SSL0E 位 (SSL0 输出允许位)**

此位选择允许或者禁止 SSL0 引脚的输出。在使用 SSL0 引脚时，必须将此位置“1”。

**SSL1E 位 (SSL1 输出允许位)**

此位选择允许或者禁止 SSL1 引脚的输出。在使用 SSL1 引脚时，必须将此位置“1”。

**SSL2E 位 (SSL2 输出允许位)**

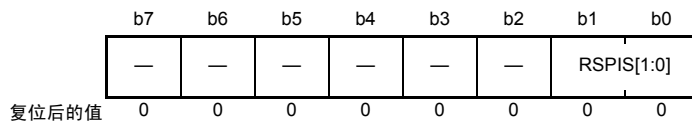
此位选择允许或者禁止 SSL2 引脚的输出。在使用 SSL2 引脚时，必须将此位置“1”。

**SSL3E 位 (SSL3 允许位)**

此位选择允许或者禁止 SSL3 引脚的输出。在使用 SSL3 引脚时，必须将此位置“1”。

## 15.1.2.12 端口功能寄存器 H (PFHSPI)

地址 0008 C111h



位	符号	位名	功能	R/W
b1-b0	RSPIS[1:0]	RSPI 引脚选择位	b1 b0 0 0: 将 P22 设定为 MISO-A 输入引脚 将 P23 设定为 MOSI-A 输入引脚 将 P24 设定为 RSPCK-A 输入引脚 将 P30 设定为 SSL0-A 输入引脚 将 P31 设定为 SSL1-A 输入引脚 将 P32 设定为 SSL2-A 输入引脚 将 P33 设定为 SSL3-A 输入引脚 0 1: 将 PA5 设定为 MISO-B 输入引脚 将 PB0 设定为 MOSI-B 输入引脚 将 PA4 设定为 RSPCK-B 输入引脚 将 PA3 设定为 SSL0-B 输入引脚 将 PA2 设定为 SSL1-B 输入引脚 将 PA1 设定为 SSL2-B 输入引脚 将 PA0 设定为 SSL3-B 输入引脚 1 0: 将 PD1 设定为 MISO-C 输入引脚 将 PD2 设定为 MOSI-C 输入引脚 将 PD0 设定为 RSPCK-C 输入引脚 将 PD6 设定为 SSL0-C 输入引脚 将 PD7 设定为 SSL1-C 输入引脚 将 PE0 设定为 SSL2-C 输入引脚 将 PE1 设定为 SSL3-C 输入引脚 1 1: 不能设定	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

PFHSPI 寄存器是设定 RSPI 的相关输入 / 输出引脚的寄存器。

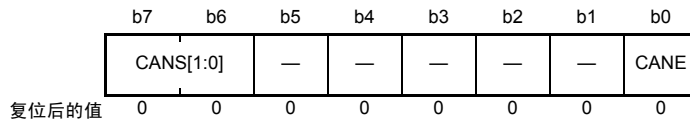
## RSPIS[1:0] 位 (RSPI 引脚选择位)

此位选择 RSPI 的输入 / 输出引脚。

RSPI 的各输入 / 输出引脚有允许位，选择允许位为“1”的引脚而不选择允许位为“0”的引脚。

## 15.1.2.13 端口功能寄存器 J (PFJCAN)

地址 0008 C113h



位	符号	位名	功能	R/W
b0	CANE	CAN 引脚允许位	0: CTX 引脚和 CRX 引脚无效 1: CTX 引脚和 CRX 引脚有效	R/W
b5-b1	—	保留位	读写值都为“0”。	R/W
b7-b6	CANS[1:0]	CAN 引脚选择位	b7 b6 00: 将 PB5 设定为 CTX-A 输入引脚 将 PB6 设定为 CRX-A 输入引脚 01: 将 P23 设定为 CTX-B 输入引脚 将 P22 设定为 CRX-B 输入引脚 10: 将 PE0 设定为 CRX-C 输入引脚 将 PD7 设定为 CTX-C 输入引脚 11: 不能设定	R/W

PFJCAN 寄存器是设定 CAN 的相关输入 / 输出引脚的寄存器。

## CANE 位 (CAN 引脚允许位)

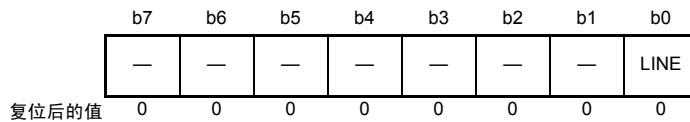
此位选择允许或者禁止 CAN 引脚。在使用 CAN 引脚时，必须将此位置“1”。

## CANS[1:0] 位 (CAN 引脚选择位)

选择 CAN 的输入 / 输出引脚。

## 15.1.2.14 端口功能寄存器 K (PFKLIN)

地址 0008 C114h



位	符号	位名	功能	R/W
b0	LINE	LIN 引脚允许位	0: LTX 引脚和 LRX 引脚无效 1: LTX 引脚和 LRX 引脚有效	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

PFKLIN 寄存器是设定 LIN 的相关输入 / 输出引脚的寄存器。

## LINE 位 (LIN 引脚允许位)

此位选择允许或者禁止 LIN 引脚。在使用 LIN 引脚时，必须将此位置“1”。

## 15.1.2.15 端口功能寄存器 M (PFMPOE)

地址 0008 C116h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	POE11E	POE10E	POE8E	POE4E	POE0E
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	POE0E	POE0# 输入允许位	0: 设定为 I/O 端口 1: 设定为 POEn# 输入引脚 (n=0、4、8、10、11)	R/W (注1)
b1	POE4E	POE4# 输入允许位		R/W (注1)
b2	POE8E	POE8# 输入允许位		R/W (注1)
b3	POE10E	POE10# 输入允许位		R/W (注1)
b4	POE11E	POE11# 输入允许位		R/W (注1)
b7-b5	—	保留位		读写值都为“0”。

注 1. 只有复位开始后的第 1 次写操作有效，不能进行第 2 次以后的写操作。

PFMPOE 寄存器是选择允许或者禁止 POEn# 输入引脚的寄存器。

为了防止系统的误动作，必须在复位开始后进行写操作。只有复位开始后的第 1 次写操作有效。

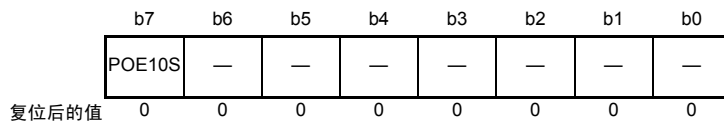
## POEnE 位 (POEn# 输入允许位) (n=0、4、8、10、11)

此位选择允许或者禁止对应的 POEn# 输入引脚。

在使用 POEn# 时，必须将对应的 POEnE 位置“1”。

## 15.1.2.16 端口功能寄存器 N (PFNPOE)

地址 0008 C117h



位	符号	位名	功能	R/W
b6-b0	—	保留位	读写值都为“0”。	R/W
b7	POE10S	POE10# 输入选择位	0: 将 PE2 设定为 POE10#-A 输入引脚 1: 将 PE4 设定为 POE10#-B 输入引脚	R/W (注1)

注 1. 只有复位开始后的第 1 次写操作有效，不能进行第 2 次以后的写操作。

PFNPOE 寄存器是选择允许或者禁止 POEn# 输入引脚的寄存器。

为了防止系统的误动作，必须在复位开始后进行写操作。只有复位开始后的第 1 次写操作有效。

## POE10S 位 (POE10# 输入选择位)

此位选择 POE10# 的输入引脚。

### 15.1.3 端口的设定

如果将各外围模块的引脚设定为有效，就能转换各端口的设定。

能通过设定各外围模块，独立设定用作输入的引脚。除端口寄存器的读、NMI 引脚的输入和 POEn# 引脚的输入以外，需要通过将输入缓冲控制寄存器（PORTn.ICR）的对应位置“1”，将输入缓冲器设定为有效。

对于用作输出和输入 / 输出的引脚，需要将各外围模块引脚的输出信号设定为有效。如果被相同端口多路复用的各外围模块输出信号的有效设定发生竞争，就根据外围模块的端口多路复用优先级，优先的外围模块功能变为有效。

外围模块的端口多路复用优先级一览表如表 15.5 所示。

表 15.5 外围模块的端口多路复用优先级一览表（112 引脚 LQFP）

优先级	模块名	输出引脚名	
高 ↑      ↓ 低	1	RSPI	RSPCK、MOSI、MISO、SSL0 ~ SSL3
	2	CAN	CTX
	3	LIN	LTX
	4	MTU0 ~ MTU7	MTIOC0A、MTIOC0B、MTIOC0C、MTIOC0D、MTIOC1A、MTIOC1B、MTIOC2A、MTIOC2B、MTIOC3A、MTIOC3B、MTIOC3C、MTIOC3D、MTIOC4A、MTIOC4B、MTIOC4C、MTIOC4D、MTIOC6A、MTIOC6B、MTIOC6C、MTIOC6D、MTIOC7A、MTIOC7B、MTIOC7C、MTIOC7D
	5	GPT0 ~ GPT3	GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B、GTIOC2A、GTIOC2B、GTIOC3A、GTIOC3B
	6	SCI0 ~ SCI2	SCK0 ~ SCK2、TXD0 ~ TXD2
	7	RIIC	SCL、SDA
	8	IOPORT	P10 ~ P11、P20 ~ P24、P30 ~ P33、P70 ~ P76、P80 ~ P82、P90 ~ P96、PA0 ~ PA5、PB0 ~ PB7、PD0 ~ PD7、PE0 ~ PE1、PE3 ~ PE5、PG0 ~ PG5



### 15.1.4 输出允许设定一览表

各端口的输出允许设定一览表如表 15.6 所示。

对应的输出信号的详细内容请参照各外围模块的寄存器说明。

能通过端口功能寄存器，更改各外围模块引脚名末尾带有 A ~ C 的任意一个字符的引脚功能。

表 15.6 各端口的输出允许设定一览表 (1/4) (112 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
P10	PORT1	P10		PORT1.DDR.B0=1
P11	PORT1	P11		PORT1.DDR.B1=1
P20	PORT2	P20		PORT2.DDR.B0=1
P21	PORT2	P21		PORT2.DDR.B1=1
P22	RSPI	MISO-A	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	PORT2	P22		PORT2.DDR.B2=1
P23	RSPI	MOSI-A	PFGSPI.MOSIE=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	CAN	CTX-B	PFJCAN.CANE =1 PFJCAN.CANS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	LIN	LTX	PFKLIN.LINE=1	(信号的输出状态取决于外围模块的设定)
	PORT2	P23		PORT2.DDR.B3=1
P24	RSPI	RSPCK-A	PFGSPI.RSPCKE =1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	PORT2	P24		PORT2.DDR.B4=1
P30	RSPI	SSL0-A	PFGSPI.SSL0E=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	MTU0	MTIOC0B-B	PFCMTU.MTUS1=1	(信号的输出状态取决于外围模块的设定)
	PORT3	P30		PORT3.DDR.B0=1
P31	RSPI	SSL1-A	PFGSPI.SSL1E=1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU0	MTIOC0A-B	PFCMTU.MTUS0=1	(信号的输出状态取决于外围模块的设定)
	PORT3	P31		PORT3.DDR.B1=1
P32	RSPI	SSL2-A	PFGSPI.SSL2E=1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU3	MTIOC3C		(信号的输出状态取决于外围模块的设定)
	PORT3	P32		PORT3.DDR.B2=1
P33	RSPI	SSL3-A	PFGSPI.SSL3E =1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU3	MTIOC3A		(信号的输出状态取决于外围模块的设定)
	PORT3	P33		PORT3.DDR.B3=1
P70	PORT7	P70		PORT7.DDR.B0=1
P71	MTU3	MTIOC3B		(信号的输出状态取决于外围模块的设定)
	GPT0	GTIOC0A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P71		PORT7.DDR.B1=1
P72	MTU4	MTIOC4A		(信号的输出状态取决于外围模块的设定)
	GPT1	GTIOC1A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P72		PORT7.DDR.B2=1
P73	MTU4	MTIOC4B		(信号的输出状态取决于外围模块的设定)
	GPT2	GTIOC2A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P73		PORT7.DDR.B3=1

表 15.6 各端口的输出允许设定一览表 (2/4) (112 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
P74	MTU3	MTIOC3D		(信号的输出状态取决于外围模块的设定)
	GPT0	GTIOC0B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P74		PORT7.DDR.B4=1
P75	MTU4	MTIOC4C		(信号的输出状态取决于外围模块的设定)
	GPT1	GTIOC1B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P75		PORT7.DDR.B5=1
P76	MTU4	MTIOC4D		(信号的输出状态取决于外围模块的设定)
	GPT2	GTIOC2B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P76		PORT7.DDR.B6=1
P80	PORT8	P80		PORT8.DDR.B0=1
P81	SCI2	TXD2-B	PFFSCI.SCI2S=1	SCI2.SCR.TE=1
	PORT8	P81		PORT8.DDR.B1=1
P82	SCI2	SCK2-B	PFFSCI.SCI2S=1	当 SCI2.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1; 当 SCI2.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORT8	P82		PORT8.DDR.B2=1
P90	MTU7	MTIOC7D		(信号的输出状态取决于外围模块的设定)
	I/O 端口	P90		PORT9.DDR.B0=1
P91	MTU7	MTIOC7C		(信号的输出状态取决于外围模块的设定)
	PORT9	P91		PORT9.DDR.B1=1
P92	MTU6	MTIOC6D		(信号的输出状态取决于外围模块的设定)
	PORT9	P92		PORT9.DDR.B2=1
P93	MTU7	MTIOC7B		(信号的输出状态取决于外围模块的设定)
	PORT9	P93		PORT9.DDR.B3=1
P94	MTU7	MTIOC7A		(信号的输出状态取决于外围模块的设定)
	PORT9	P94		PORT9.DDR.B4=1
P95	MTU6	MTIOC6B		(信号的输出状态取决于外围模块的设定)
	PORT9	P95		PORT9.DDR.B5=1
P96	PORT9	P96		PORT9.DDR.B6=1
PA0	RSPI	SSL3-B	PFGSPI.SSL3E=1 PFHSPI.RSPIS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	MTU6	MTIOC6C		(信号的输出状态取决于外围模块的设定)
	PORTA	PA0		PORTA.DDR.B0=1
PA1	RSPI	SSL2-B	PFGSPI.SSL2E=1 PFHSPI.RSPIS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	MTU6	MTIOC6A		(信号的输出状态取决于外围模块的设定)
	PORTA	PA1		PORTA.DDR.B1=1
PA2	RSPI	SSL1-B	PFGSPI.SSL1E=1 PFHSPI.RSPIS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	MTU2	MTIOC2B		(信号的输出状态取决于外围模块的设定)
	PORTA	PA2		PORTA.DDR.B2=1
PA3	RSPI	SSL0-B	PFGSPI.SSL0E=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU2	MTIOC2A		(信号的输出状态取决于外围模块的设定)
	PORTA	PA3		PORTA.DDR.B3=1

表 15.6 各端口的输出允许设定一览表 (3/4) (112 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
PA4	RSPI	RSPCK-B	PFGSPI.RSPCKE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU1	MTIOC1B		(信号的输出状态取决于外围模块的设定)
	PORTA	PA4		PORTA.DDR.B4=1
PA5	RSPI	MISO-B	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU1	MTIOC1A		(信号的输出状态取决于外围模块的设定)
	PORTA	PA5		PORTA.DDR.B5=1
PB0	RSPI	MOSI-B	PFGSPI.MOSIE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU0	MTIOC0D		(信号的输出状态取决于外围模块的设定)
	PORTB	PB0		PORTB.DDR.B0=1
PB1	MTU0	MTIOC0C		(信号的输出状态取决于外围模块的设定)
	RIIC	SCL		RIIC.ICCR1.ICE=1
	PORTB	PB1		PORTB.DDR.B1=1
PB2	MTU0	MTIOC0B-A	PFCMTU.MTUS1=0	(信号的输出状态取决于外围模块的设定)
	SCI0	TXD0		SCI0.SCR.TE=1
	RIIC	SDA		RIIC.ICCR1.ICE=1
	PORTB	PB2		PORTB.DDR.B2=1
PB3	MTU0	MTIOC0A-A	PFCMTU.MTUS0=0	(信号的输出状态取决于外围模块的设定)
	SCI0	SCK0		当 SCI0.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1; 当 SCI0.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTB	PB3		PORTB.DDR.B3=1
PB4	PORTB	PB4		PORTB.DDR.B4=1
PB5	CAN	CTX-A	PFJCAN.CANE=1 PFJCAN.CANS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	SCI2	TXD2-A	PFFSCI.SCI2S=0	SCI2.SCR.TE=1
	PORTB	PB5		PORTB.DDR.B5=1
PB6	PORTB	PB6		PORTB.DDR.B6=1
PB7	SCI2	SCK2-A	PFFSCI.SCI2S=0	当 SCI2.SCMR.SMIF =1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1; 当 SCI2.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTB	PB7		PORTB.DDR.B7=1
PD0	RSPI	RSPCK-C	PFGSPI.RSPCKE=1 PFHSPI.RSPIS[1:0]=10	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	GPT3	GTIOC3B		(信号的输出状态取决于外围模块的设定)
	PORTD	PD0		PORTD.DDR.B0=1

表 15.6 各端口的输出允许设定一览表 (4/4) (112 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
PD1	RSPI	MISO-C	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=10	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	GPT3	GTIOC3A		(信号的输出状态取决于外围模块的设定)
	PORTD	PD1		PORTD.DDR.B1=1
PD2	RSPI	MOSI-C	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=10	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	GPT2	GTIOC2B-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD2		PORTD.DDR.B2=1
PD3	GPT2	GTIOC2A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	SCI1	TXD1		SCI1.SCR.TE=1
	PORTD	PD3		PORTD.DDR.B3=1
PD4	GPT1	GTIOC1B-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	SCI1	SCK1		当 SCI1.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1; 当 SCI1.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTD	PD4		PORTD.DDR.B4=1
PD5	GPT1	GTIOC1A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD5		PORTD.DDR.B5=1
PD6	RSPI	SSL0-C	PFGSPI.SSL0E=1 PFHSPI.RSPIS[1:0]=10	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	GPT0	GTIOC0B-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD6		PORTD.DDR.B6=1
PD7	RSPI	SSL1-C	PFGSPI.SSL1E=1 PFHSPI.RSPIS[1:0]=10	(信号的输出状态取决于外围模块的设定)
	CAN	CTX-C	PFJCAN.CANE=1 PFJCAN.CANS[1:0]=10	(信号的输出状态取决于外围模块的设定)
	GPT0	GTIOC0A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD7		PORTD.DDR.B7=1
PE0	RSPI	SSL2-C	PFGSPI.SSL2E=1 PFHSPI.RSPIS[1:0]=10	(信号的输出状态取决于外围模块的设定)
	PORTE	PE0		PORTE.DDR.B0=1
PE1	RSPI	SSL3-C	PFGSPI.SSL3E=1 PFHSPI.RSPIS[1:0]=10	(信号的输出状态取决于外围模块的设定)
	PORTE	PE1		PORTE.DDR.B1=1
PE3	PORTE	PE3		PORTE.DDR.B3=1
PE4	PORTE	PE4		PORTE.DDR.B4=1
PE5	PORTE	PE5		PORTE.DDR.B5=1
PG0	PORTG	PG0		PORTG.DDR.B0=1
PG1	PORTG	PG1		PORTG.DDR.B1=1
PG2	PORTG	PG2		PORTG.DDR.B2=1
PG3	PORTG	PG3		PORTG.DDR.B3=1
PG4	PORTG	PG4		PORTG.DDR.B4=1
PG5	PORTG	PG5		PORTG.DDR.B5=1

### 15.1.5 未使用引脚的处理

未使用引脚的处理内容如表 15.7 所示。

表 15.7 未使用引脚的处理内容（112 引脚 LQFP）

引脚名	处理内容
EMLE	通过电阻连接 VSS（下拉）。
MD1、MD0	（必须用作模式引脚。）
MDE	（必须用作模式引脚。）
RES#	通过电阻连接 VCC（上拉）。
PE2/NMI/POE10#-A	通过电阻连接 VCC（上拉）。
EXTAL	（必须用作时钟引脚。）
XTAL	将引脚置为开路。
WDTOVF#	将引脚置为开路。
端口 1 ~ 3、7 ~ 9、 A、B、D、E、G	<ul style="list-style-type: none"> <li>每个引脚通过电阻连接 VCC（上拉）或者通过电阻连接 VSS（下拉）。</li> <li>也能在 PORTn.ICR 为初始值（输入缓冲器无效）的状态下将引脚置为开路（注 1）。</li> </ul>
端口 4	<ul style="list-style-type: none"> <li>每个引脚通过电阻连接 AVCC0（上拉）或者通过电阻连接 AVSS0（下拉）。</li> <li>也能在 PORTn.ICR 为初始值（输入缓冲器无效）的状态下将引脚置为开路（注 1）。</li> </ul>
端口 5 ~ 6	<ul style="list-style-type: none"> <li>每个引脚通过电阻连接 AVCC（上拉）或者通过电阻连接 AVSS（下拉）。</li> <li>也能在 PORTn.ICR 为初始值（输入缓冲器无效）的状态下将引脚置为开路（注 1）。</li> </ul>
VREFH0	连接 AVCC0。
VREFL0	连接 AVSS0。
VREF	连接 AVCC。
TRST#、TMS、TCK、 TDI	每个引脚通过电阻连接 VCC（上拉）或者通过电阻连接 VSS（下拉）
TDO	置为开路。

注 1. 不能改变 PORTn.ICR 寄存器的初始值，否则就可能有穿透电流流过。

## 15.2 100 引脚 LQFP 的 I/O 端口

RX62T 群的 100 引脚 LQFP 的 I/O 端口由端口 1 ~ 9、A、B、D、E 共 13 个端口构成，有 55 个输入 / 输出引脚。

### 15.2.1 概要

I/O 端口的规格和端口功能一览表分别如表 15.8 和表 15.9 所示。

表 15.8 I/O 端口的规格（100 引脚 LQFP）

项目	内容
输入 / 输出引脚	55 个
输入引脚	21 个
端口	13 个端口（1 ~ 9、A、B、D、E）
漏极开路输出	2 个（RIIC 引脚）
大电流输出	12 个（MTU3 引脚和 GPT 引脚）
施密特触发输入引脚	全部端口输入、CAN 输入、IRQ 输入、MTU3 输入、POE3 输入、RIIC 输入、SCI 输入、A/D 触发输入、NMI 输入、GPT 输入、LIN 输入
其他	<ul style="list-style-type: none"> <li>能驱动 1 个 TTL 负载和 30pF 的电容负载。</li> <li>在输出时驱动达林顿晶体管。</li> <li>能随时读引脚的状态。</li> </ul>

表 15.9 端口功能一览表（1 / 3）（100 引脚 LQFP）

端口	概要	位	功能			CMOS 输入引脚	施密特触发输入引脚	漏极开路输出功能	大电流输出引脚
			输入 / 输出	输入	输出				
端口 1	与 MTU3 输入和中断输入兼用的通用输入 / 输出端口	0	P10	MTCLKD-B/IRQ0-A		—	全部输入功能	—	—
		1	P11	MTCLKC-B/IRQ1-A			全部输入功能		
端口 2	与 MTU3 输入、中断输入、A/D 转换器输入、RSPI 输入 / 输出、LIN 输入 / 输出和 CAN 输入 / 输出兼用的通用输入 / 输出端口	0	P20	MTCLKB-B/IRQ7/ADTRG0#-B		—	全部输入功能	—	—
		1	P21	MTCLKA-B/IRQ6/ADTRG1#-B			全部输入功能		
		2	P22/MISO-A	LRX/CRX-B/ADTRG#		MISO-A	P22/LRX/CRX-B/ADTRG#		
		3	P23/MOSI-A		LTX/CTX-B	MOSI-A	P23		
端口 3	与 MTU3 输入 / 输出和 RSPI 输入 / 输出兼用的通用输入 / 输出端口	0	P30/MTIOC0B-B/SSL0-A	MTCLKD-A		SSL0-A	P30/MTIOC0B-B/MTCLKD-A	—	—
		1	P31/MTIOC0A-B	MTCLKC-A	SSL1-A	—	全部输入功能		
		2	P32/MTIOC3C	MTCLKB-A	SSL2-A		全部输入功能		
		3	P33/MTIOC3A	MTCLKA-A	SSL3-A		全部输入功能		

表 15.9 端口功能一览表 (2/3) (100 引脚 LQFP)

端口	概要	位	功能			CMOS 输入引脚	施密特触发 输入引脚	漏极开路 输出功能	大电流 输出引脚	
			输入 / 输出	输入	输出					
端口 4	与 A/D 转换器 输入兼用的通 用输入端口	0		P40/AN000		—	P40	—	—	
		1		P41/AN001			P41			
		2		P42/AN002			P42			
		3		P43/AN003/ CVREFL			P43			
		4		P44/AN100			P44			
		5		P45/AN101			P45			
		6		P46/AN102			P46			
		7		P47/AN103/ CVREFH			P47			
端口 5	与 A/D 转换器 输入兼用的通 用输入端口	0		P50/AN6		—	P50	—	—	
		1		P51/AN7			P51			
		2		P52/AN8			P52			
		3		P53/AN9			P53			
		4		P54/AN10			P54			
		5		P55/AN11			P55			
端口 6	与 A/D 转换器 输入兼用的通 用输入端口	0		P60/AN0		—	P60	—	—	
		1		P61/AN1			P61			
		2		P62/AN2			P62			
		3		P63/AN3			P63			
		4		P64/AN4			P64			
		5		P65/AN5			P65			
端口 7	与 POE3 输 入、中断输入、 MTU3 输入 / 输 出和 GPT 输入 / 输出兼用的通 用输入 / 输出端 口	0	P70	POE0#/IRQ5		—	全部输入功能	—	—	
		1	P71/MTIOC3B/ GTIOC0A-A				全部输入功能			○
		2	P72/MTIOC4A/ GTIOC1A-A				全部输入功能			○
		3	P73/MTIOC4B/ GTIOC2A-A				全部输入功能			○
		4	P74/MTIOC3D/ GTIOC0B-A				全部输入功能			○
		5	P75/MTIOC4C/ GTIOC1B-A				全部输入功能			○
		6	P76/MTIOC4D/ GTIOC2B-A				全部输入功能			○
端口 8	与 SCI 输入 / 输出和 MTU3 输入兼用的通 用输入 / 输出端 口	0	P80	RXD2-B/MTIC5W		—	全部输入功能	—	—	
		1	P81	MTIC5V	TXD2-B		全部输入功能			
		2	P82/SCK2-B	MTIC5U			全部输入功能			
端口 9	与 MTU3 输入 / 输出、POE3 输入和中断输 入兼用的通 用输入 / 输出端 口	0	P90/MTIOC7D			—	全部输入功能	—	○	
		1	P91/MTIOC7C				全部输入功能		○	
		2	P92/MTIOC6D				全部输入功能		○	
		3	P93/MTIOC7B				全部输入功能		○	
		4	P94/MTIOC7A				全部输入功能		○	
		5	P95/MTIOC6B				全部输入功能		○	
		6	P96	POE4#/IRQ4			全部输入功能		—	

表 15.9 端口功能一览表 (3/3) (100 引脚 LQFP)

端口	概要	位	功能			CMOS 输入引脚	施密特触发 输入引脚	漏极开路 输出功能	大电流 输出引脚	
			输入 / 输出	输入	输出					
端口 A	与 MTU3 输入 / 输出、RSPI 输入 / 输出和 A/D 转换器输入兼用的通用输入 / 输出端口	0	PA0/MTIOC6C		SSL3-B	—	全部输入功能	—	—	
		1	PA1/MTIOC6A		SSL2-B		全部输入功能			
		2	PA2/MTIOC2B		SSL1-B		全部输入功能			
		3	PA3/MTIOC2A/ SSL0-B			SSL0-B	PA3/MTIOC2A			
		4	PA4/MTIOC1B/ RSPCK-B	ADTRG0#-A		RSPCK-B	PA4/MTIOC1B/ ADTRG0#-A			
		5	PA5/MTIOC1A/ MISO-B	ADTRG1#-A		MISO-B	PA5/MTIOC1A/ ADTRG1#-A			
端口 B	与 MTU3 输入 / 输出、RSPI 输入 / 输出、RIIC 输入 / 输出、SCI 输入 / 输出、GPT 输入、POE3 输入、中断输入、CAN 输入 / 输出和跟踪输出兼用的通用输入 / 输出端口	0	PB0/MTIOC0D/ MOSI-B			MOSI-B	PB0/MTIOC0D	—	—	
		1	PB1/MTIOC0C/ SCL	RXD0		—	全部输入功能	○ (仅 SCL)	—	
		2	PB2/MTIOC0B-A/ SDA		TXD0		全部输入功能	○ (仅 SDA)	—	
		3	PB3/MTIOC0A-A/ SCK0				全部输入功能	—	—	
		4	PB4	GTETRG/ POE8#/IRQ3			全部输入功能			
		5	PB5		TXD2-A/ CTX-A/ TRSYNC		全部输入功能			
		6	PB6	RXD2-A/CRX-A	TRDATA0		全部输入功能			
		7	PB7/SCK2-A		TRDATA1		全部输入功能			
端口 D	与 GPT 输入 / 输出、RSPI 输入 / 输出、SCI 输入 / 输出、CAN 输出、跟踪输出和内部调试器输入 / 输出兼用的通用输入 / 输出端口	0	PD0/GTIOC3B/ RSPCK-C		TRDATA2	RSPCK-C	PD0/GTIOC3B	—	—	
		1	PD1/GTIOC3A/ MISO-C		TRDATA3	MISO-C	PD1/GTIOC3A			
		2	PD2/GTIOC2B-B/ MOSI-C		TRCLK		MOSI-C	PD2/GTIOC2B-B		
		3	PD3/GTIOC2A-B		TXD1/TDO	—	全部输入功能			
		4	PD4/GTIOC1B-B/ SCK1	TCK			—	全部输入功能		
		5	PD5/GTIOC1A-B	RXD1/TDI			—	全部输入功能		
		6	PD6/GTIOC0B-B/ SSL0-C	TMS			SSL0-C	PD6/GTIOC0B-B		
		7	PD7/GTIOC0A-B	TRST#	SSL1-C/CTX-C		—	全部输入功能		
端口 E	与 CAN 输入、RSPI 输出、POE3 输入、MTU3 输入和中断输入兼用的通用输入 / 输出端口	0	PE0	CRX-C	SSL2-C	—	全部输入功能	—	—	
		1	PE1		SSL3-C		全部输入功能			
		2		PE2/POE10#-A/NMI			全部输入功能			
		3	PE3	MTCLKD-C/ POE11#/IRQ2-A			全部输入功能			
		4	PE4	MTCLKC-C/ POE10#-B/ IRQ1-B			全部输入功能			
		5	PE5	IRQ0-B			全部输入功能			



## 15.2.2 寄存器说明

I/O 端口的寄存器一览表和各寄存器的有效位一览表分别如表 15.10 和表 15.11 所示。

表 15.10 I/O 端口的寄存器一览表 (1/2) (100 引脚 LQFP)

端口符号	寄存器名	寄存器符号	复位后的值	地址	存取长度
PORT1	数据方向寄存器	DDR	00h	0008 C001h	8
	数据寄存器	DR	00h	0008 C021h	8
	端口寄存器	PORT	不定值	0008 C041h	8
	输入缓冲控制寄存器	ICR	00h	0008 C061h	8
PORT2	数据方向寄存器	DDR	00h	0008 C002h	8
	数据寄存器	DR	00h	0008 C022h	8
	端口寄存器	PORT	不定值	0008 C042h	8
	输入缓冲控制寄存器	ICR	00h	0008 C062h	8
PORT3	数据方向寄存器	DDR	00h	0008 C003h	8
	数据寄存器	DR	00h	0008 C023h	8
	端口寄存器	PORT	不定值	0008 C043h	8
	输入缓冲控制寄存器	ICR	00h	0008 C063h	8
PORT4	端口寄存器	PORT	不定值	0008 C044h	8
	输入缓冲控制寄存器	ICR	00h	0008 C064h	8
PORT5	端口寄存器	PORT	不定值	0008 C045h	8
	输入缓冲控制寄存器	ICR	00h	0008 C065h	8
PORT6	端口寄存器	PORT	不定值	0008 C046h	8
	输入缓冲控制寄存器	ICR	00h	0008 C066h	8
PORT7	数据方向寄存器	DDR	00h	0008 C007h	8
	数据寄存器	DR	00h	0008 C027h	8
	端口寄存器	PORT	不定值	0008 C047h	8
	输入缓冲控制寄存器	ICR	00h	0008 C067h	8
PORT8	数据方向寄存器	DDR	00h	0008 C008h	8
	数据寄存器	DR	00h	0008 C028h	8
	端口寄存器	PORT	不定值	0008 C048h	8
	输入缓冲控制寄存器	ICR	00h	0008 C068h	8
PORT9	数据方向寄存器	DDR	00h	0008 C009h	8
	数据寄存器	DR	00h	0008 C029h	8
	端口寄存器	PORT	不定值	0008 C049h	8
	输入缓冲控制寄存器	ICR	00h	0008 C069h	8
PORTA	数据方向寄存器	DDR	00h	0008 C00Ah	8
	数据寄存器	DR	00h	0008 C02Ah	8
	端口寄存器	PORT	不定值	0008 C04Ah	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Ah	8
PORTB	数据方向寄存器	DDR	00h	0008 C00Bh	8
	数据寄存器	DR	00h	0008 C02Bh	8
	端口寄存器	PORT	不定值	0008 C04Bh	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Bh	8

表 15.10 I/O 端口的寄存器一览表 (2/2) (100 引脚 LQFP)

端口符号	寄存器名	寄存器符号	复位后的值	地址	存取长度
PORTD	数据方向寄存器	DDR	00h	0008 C00Dh	8
	数据寄存器	DR	00h	0008 C02Dh	8
	端口寄存器	PORT	不定值	0008 C04Dh	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Dh	8
PORTE	数据方向寄存器	DDR	00h	0008 C00Eh	8
	数据寄存器	DR	00h	0008 C02Eh	8
	端口寄存器	PORT	不定值	0008 C04Eh	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Eh	8
IOPORT	端口功能寄存器 8	PF8IRQ	00h	0008 C108h	8
	端口功能寄存器 A	PFAADC	00h	0008 C10Ah	8
	端口功能寄存器 C	PFCMTU	00h	0008 C10Ch	8
	端口功能寄存器 D	PFDGPT	00h	0008 C10Dh	8
	端口功能寄存器 F	PFHSCI	00h	0008 C10Fh	8
	端口功能寄存器 G	PFHSPi	00h	0008 C110h	8
	端口功能寄存器 H	PFHSPi	00h	0008 C111h	8
	端口功能寄存器 J	PFJCAN	00h	0008 C113h	8
	端口功能寄存器 K	PFKLiN	00h	0008 C114h	8
	端口功能寄存器 M	PFMPOE	00h	0008 C116h	8
	端口功能寄存器 N	PFNPOE	00h	0008 C117h	8

表 15.11 各寄存器的有效位一览表 (1/2) (100 引脚 LQFP)

寄存器符号	b7	b6	b5	b4	b3	b2	b1	b0
PORT1.DDR	x	x	x	x	x	x	○	○
PORT2.DDR	x	x	x	○	○	○	○	○
PORT3.DDR	x	x	x	x	○	○	○	○
PORT7.DDR	x	○	○	○	○	○	○	○
PORT8.DDR	x	x	x	x	x	○	○	○
PORT9.DDR	x	○	○	○	○	○	○	○
PORTA.DDR	x	x	○	○	○	○	○	○
PORTB.DDR	○	○	○	○	○	○	○	○
PORTD.DDR	○	○	○	○	○	○	○	○
PORTE.DDR	x	x	○	○	○	x	○	○
PORT1.DR	x	x	x	x	x	x	○	○
PORT2.DR	x	x	x	○	○	○	○	○
PORT3.DR	x	x	x	x	○	○	○	○
PORT7.DR	x	○	○	○	○	○	○	○
PORT8.DR	x	x	x	x	x	○	○	○
PORT9.DR	x	○	○	○	○	○	○	○
PORTA.DR	x	x	○	○	○	○	○	○
PORTB.DR	○	○	○	○	○	○	○	○
PORTD.DR	○	○	○	○	○	○	○	○
PORTE.DR	x	x	○	○	○	x	○	○

表 15.11 各寄存器的有效位一览表 (2/2) (100 引脚 LQFP)

寄存器符号	b7	b6	b5	b4	b3	b2	b1	b0
PORT1.PORT	x	x	x	x	x	x	○	○
PORT2.PORT	x	x	x	○	○	○	○	○
PORT3.PORT	x	x	x	x	○	○	○	○
PORT4.PORT	○	○	○	○	○	○	○	○
PORT5.PORT	x	x	○	○	○	○	○	○
PORT6.PORT	x	x	○	○	○	○	○	○
PORT7.PORT	x	○	○	○	○	○	○	○
PORT8.PORT	x	x	x	x	x	○	○	○
PORT9.PORT	x	○	○	○	○	○	○	○
PORTA.PORT	x	x	○	○	○	○	○	○
PORTB.PORT	○	○	○	○	○	○	○	○
PORTD.PORT	○	○	○	○	○	○	○	○
PORTE.PORT	x	x	○	○	○	○	○	○
PORT1.ICR	x	x	x	x	x	x	○	○
PORT2.ICR	x	x	x	○	○	○	○	○
PORT3.ICR	x	x	x	x	○	○	○	○
PORT4.ICR	○	○	○	○	○	○	○	○
PORT5.ICR	x	x	○	○	○	○	○	○
PORT6.ICR	x	x	○	○	○	○	○	○
PORT7.ICR	x	○	○	○	○	○	○	○
PORT8.ICR	x	x	x	x	x	○	○	○
PORT9.ICR	x	○	○	○	○	○	○	○
PORTA.ICR	x	x	○	○	○	○	○	○
PORTB.ICR	○	○	○	○	○	○	○	○
PORTD.ICR	○	○	○	○	○	○	○	○
PORTE.ICR	x	x	○	○	○	x	○	○
IOPORT.PF8IRQ	x	x	x	x	○	○	○	○
IOPORT.PFAADC	x	x	x	x	x	x	○	○
IOPORT.PFCMTU	○	○	x	x	x	x	○	○
IOPORT.PFDGPT	x	x	x	x	x	x	x	○
IOPORT.PFFSCI	x	x	x	x	x	○	x	x
IOPORT.PFGSPI	○	○	○	○	○	○	○	x
IOPORT.PFHSPI	x	x	x	x	x	x	○	○
IOPORT.PFJCAN	○	○	x	x	x	x	x	○
IOPORT.PFKLIN	x	x	x	x	x	x	x	○
IOPORT.PFMPOE	x	x	x	○	○	○	○	○
IOPORT.PFNPOE	○	x	x	x	x	x	x	x

○: 有效位、x: 无效位 (保留位)

## 15.2.2.1 数据方向寄存器 (DDR)

地址 PORT1.DDR 0008 C001h、PORT2.DDR 0008 C002h、PORT3.DDR 0008 C003h、PORT7.DDR 0008 C007h、PORT8.DDR 0008 C008h、PORT9.DDR 0008 C009h、PORTA.DDR 0008 C00Ah、PORTB.DDR 0008 C00Bh、PORTD.DDR 0008 C00Dh、PORTE.DDR 0008 C00Eh

	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

- 注 1. PORT1.DDR 的低 2 位有效，高 6 位是保留位。  
 PORT2.DDR 的低 5 位有效，高 3 位是保留位。  
 PORT3.DDR 的低 4 位有效，高 4 位是保留位。  
 PORT7.DDR 的低 7 位有效，高 1 位是保留位。  
 PORT8.DDR 的低 3 位有效，高 5 位是保留位。  
 PORT9.DDR 的低 7 位有效，高 1 位是保留位。  
 PORTA.DDR 的低 6 位有效，高 2 位是保留位。  
 PORTE.DDR 的 b5~b3、b1、b0 有效，b7、b6、b2 是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0	Pn0 输入 / 输出指定位	0: 输入端口 1: 输出端口	R/W
b1	B1	Pn1 输入 / 输出指定位		R/W
b2	B2	Pn2 输入 / 输出指定位		R/W
b3	B3	Pn3 输入 / 输出指定位		R/W
b4	B4	Pn4 输入 / 输出指定位		R/W
b5	B5	Pn5 输入 / 输出指定位		R/W
b6	B6	Pn6 输入 / 输出指定位		R/W
b7	B7	Pn7 输入 / 输出指定位		R/W

n=1~3、7~9、A、B、D、E

PORTn.DDR 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。

PORTn.DDR 寄存器 (n=1~3、7~9、A、B、D、E) 的各位分别对应端口 n 的各引脚，并且能以 1 位为单位进行指定。

## 15.2.2.2 数据寄存器 (DR)

地址 PORT1.DR 0008 C021h、PORT2.DR 0008 C022h、PORT3.DR 0008 C023h、PORT7.DR 0008 C027h、PORT8.DR 0008 C028h、PORT9.DR 0008 C029h、PORTA.DR 0008 C02Ah、PORTB.DR 0008 C02Bh、PORTD.DR 0008 C02Dh、PORTE.DR 0008 C02Eh

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

- 注 1. PORT1.DR 的低 2 位有效，高 6 位是保留位。  
 PORT2.DR 的低 5 位有效，高 3 位是保留位。  
 PORT3.DR 的低 4 位有效，高 4 位是保留位。  
 PORT7.DR 的低 7 位有效，高 1 位是保留位。  
 PORT8.DR 的低 3 位有效，高 5 位是保留位。  
 PORT9.DR 的低 7 位有效，高 1 位是保留位。  
 PORTA.DR 的低 6 位有效，高 2 位是保留位。  
 PORTE.DR 的 b5 ~ b3、b1、b0 有效，b7、b6、b2 是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0	Pn0 输出数据保存位	保存输出数据。	R/W
b1	B1	Pn1 输出数据保存位		R/W
b2	B2	Pn2 输出数据保存位		R/W
b3	B3	Pn3 输出数据保存位		R/W
b4	B4	Pn4 输出数据保存位		R/W
b5	B5	Pn5 输出数据保存位		R/W
b6	B6	Pn6 输出数据保存位		R/W
b7	B7	Pn7 输出数据保存位		R/W

n=1 ~ 3、7 ~ 9、A、B、D、E

PORTn.DR 寄存器 (n=1 ~ 3、7 ~ 9、A、B、D、E) 保存用作通用输出端口的引脚输出数据。

## 15.2.2.3 端口寄存器 (PORT)

地址 PORT1.PORT 0008 C041h、PORT2.PORT 0008 C042h、PORT3.PORT 0008 C043h、PORT4.PORT 0008 C044h、  
PORT5.PORT 0008 C045h、PORT6.PORT 0008 C046h、PORT7.PORT 0008 C047h、PORT8.PORT 0008 C048h、  
PORT9.PORT 0008 C049h、PORTA.PORT 0008 C04Ah、PORTB.PORT 0008 C04Bh、PORTD.PORT 0008 C04Dh、  
PORTE.PORT 0008 C04Eh

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

复位后的值 x x x x x x x x

- 注 1. PORT1.PORT 的低 2 位有效，高 6 位是保留位。  
PORT2.PORT 的低 5 位有效，高 3 位是保留位。  
PORT3.PORT 的低 4 位有效，高 4 位是保留位。  
PORT5.PORT 的低 6 位有效，高 2 位是保留位。  
PORT6.PORT 的低 6 位有效，高 2 位是保留位。  
PORT7.PORT 的低 7 位有效，高 1 位是保留位。  
PORT8.PORT 的低 3 位有效，高 5 位是保留位。  
PORT9.PORT 的低 7 位有效，高 1 位是保留位。  
PORTA.PORT 的低 6 位有效，高 2 位是保留位。  
PORTE.PORT 的低 6 位有效，高 2 位是保留位。

- 注 2. 保留位。读取值为 "1"，写操作无效。

位	符号	位名	功能	R/W
b0	B0	Pn0 位	反映端口的引脚状态。	R
b1	B1	Pn1 位		R
b2	B2	Pn2 位		R
b3	B3	Pn3 位		R
b4	B4	Pn4 位		R
b5	B5	Pn5 位		R
b6	B6	Pn6 位		R
b7	B7	Pn7 位		R

n=1 ~ 9、A、B、D、E

PORTn.PORT 寄存器是反映端口引脚状态的寄存器。

如果读 PORTn.PORT 寄存器 (n=1 ~ 9、A、B、D、E)，就能读到引脚的状态。

## 15.2.2.4 输入缓冲控制寄存器 (ICR)

地址 PORT1.ICR 0008 C061h、PORT2.ICR 0008 C062h、PORT3.ICR 0008 C063h、PORT4.ICR 0008 C064h、PORT5.ICR 0008 C065h、PORT6.ICR 0008 C066h、PORT7.ICR 0008 C067h、PORT8.ICR 0008 C068h、PORT9.ICR 0008 C069h、PORTA.ICR 0008 C06Ah、PORTB.ICR 0008 C06Bh、PORTD.ICR 0008 C06Dh、PORTE.ICR 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	0

- 注 1. PORT1.ICR 的低 2 位有效，高 6 位是保留位。  
 PORT2.ICR 的低 5 位有效，高 3 位是保留位。  
 PORT3.ICR 的低 4 位有效，高 4 位是保留位。  
 PORT5.ICR 的低 6 位有效，高 2 位是保留位。  
 PORT6.ICR 的低 6 位有效，高 2 位是保留位。  
 PORT7.ICR 的低 7 位有效，高 1 位是保留位。  
 PORT8.ICR 的低 3 位有效，高 5 位是保留位。  
 PORT9.ICR 的低 7 位有效，高 1 位是保留位。  
 PORTA.ICR 的低 6 位有效，高 2 位是保留位。  
 PORTE.ICR 的 b5~b3、b1、b0 有效，b7、b6、b2 是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0 (注 1)	Pn0 输入缓冲器控制位	0: 对应引脚的输入缓冲器无效 1: 对应引脚的输入缓冲器有效	R/W
b1	B1 (注 1)	Pn1 输入缓冲器控制位		R/W
b2	B2 (注 1)	Pn2 输入缓冲器控制位		R/W
b3	B3 (注 1)	Pn3 输入缓冲器控制位		R/W
b4	B4 (注 1)	Pn4 输入缓冲器控制位		R/W
b5	B5 (注 1)	Pn5 输入缓冲器控制位		R/W
b6	B6 (注 1)	Pn6 输入缓冲器控制位		R/W
b7	B7 (注 1)	Pn7 输入缓冲器控制位		R/W

n=1~9、A、B、D、E

注 1. 在用作输入引脚时，必须将对应位置“1”。必须将不用作输入的引脚和用作模拟输入引脚的位置“0”。

PORTn.ICR 寄存器是控制端口输入缓冲器的寄存器。

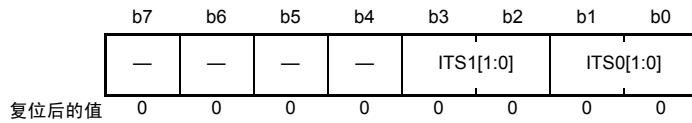
PORTn.ICR 寄存器 (n=1~9、A、B、D、E) 的各位分别对应端口 n 的各引脚，并且能以 1 位为单位进行指定。

在用作外围模块的输入引脚时，为了使对应引脚的输入缓冲器有效，需要预先将 PORTn.ICR 置“1”。如果在将 PORTn.ICR 置“0”的状态下用作外围模块的输入引脚，就将输入到外围模块的信号固定为 High 电平。

在更改 PORTn.ICR 寄存器的设定时，有可能因引脚的状态而在内部产生边沿，因此必须在不使用该输入引脚时更改 PORTn.ICR 寄存器的设定。例如，在 IRQi (i=0~7) 输入时，必须在禁止该中断的状态下更改 PORTn.ICR 寄存器的设定。先将中断控制器的 IRi.IR 标志 (i=64~71 (IRQ 的中断向量号)) 置“0”，然后允许该中断。如果在更改 PORTn.ICR 寄存器的设定后产生边沿，就必须取消该边沿。

## 15.2.2.5 端口功能寄存器 8 (PF8IRQ)

地址 0008 C108h



位	符号	位名	功能	R/W
b1-b0	ITS0[1:0]	IRQ0 引脚选择位	b1 b0 0 0: 将 P10 设定为 IRQ0-A 输入引脚 0 1: 将 PE5 设定为 IRQ0-B 输入引脚 1 0: 不能设定 1 1: 不能设定	R/W
b3-b2	ITS1[1:0]	IRQ1 引脚选择位	b3 b2 0 0: 将 P11 设定为 IRQ1-A 输入引脚 0 1: 将 PE4 设定为 IRQ1-B 输入引脚 1 0: 不能设定 1 1: 不能设定	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

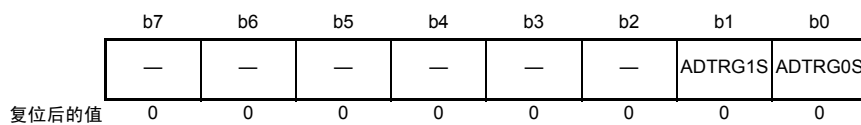
PF8IRQ 寄存器是选择 IRQ0、IRQ1 输入引脚的寄存器。

ITS<sub>i</sub> 位 (IRQ<sub>i</sub> 引脚选择位) (i=0、1)

此位选择 IRQ<sub>i</sub> 的输入引脚。

## 15.2.2.6 端口功能寄存器 A (PFAADC)

地址 0008 C10Ah



位	符号	位名	功能	R/W
b0	ADTRG0S	ADTRG0# 输入选择位	0: 将 PA4 设定为 ADTRG0#-A 输入引脚 1: 将 P20 设定为 ADTRG0#-B 输入引脚	R/W
b1	ADTRG1S	ADTRG1# 输入选择位	0: 将 PA5 设定为 ADTRG1#-A 输入引脚 1: 将 P21 设定为 ADTRG1#-B 输入引脚	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

PFAADC 寄存器是选择 ADTRG0#、ADTRG1# 引脚的寄存器。

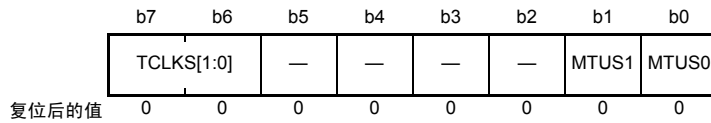
ADTRG<sub>n</sub>S 位 (ADTRG<sub>n</sub># 输入选择位) (n=0、1)

此位选择 ADTRG<sub>n</sub># 的输入引脚。



## 15.2.2.7 端口功能寄存器 C (PFCMTU)

地址 0008 C10Ch



位	符号	位名	功能	R/W
b0	MTUS0	MTU3 引脚选择 0 位	0: 选择 PB3 为 MTIOC0A-A 引脚 1: 选择 P31 为 MTIOC0A-B 引脚	R/W
b1	MTUS1	MTU3 引脚选择 1 位	0: 选择 PB2 为 MTIOC0B-A 引脚 1: 选择 P30 为 MTIOC0B-B 引脚	R/W
b5-b2	—	保留位	读写值都为“0”。	R/W
b7-b6	TCLKS[1:0]	MTCLK 引脚选择位	b7 b6 0 0: 将 P33 设定为 MTCLKA-A 输入引脚 将 P32 设定为 MTCLKB-A 输入引脚 将 P31 设定为 MTCLKC-A 输入引脚 将 P30 设定为 MTCLKD-A 输入引脚 0 1: 将 P21 设定为 MTCLKA-B 输入引脚 将 P20 设定为 MTCLKB-B 输入引脚 将 P11 设定为 MTCLKC-B 输入引脚 将 P10 设定为 MTCLKD-B 输入引脚 1 0: 将 PE4 设定为 MTCLKC-C 输入引脚 将 PE3 设定为 MTCLKD-C 输入引脚 (不能选择 MTCLKA 引脚和 MTCLKB 引脚) 1 1: 不能设定	R/W

PFCMTU 寄存器是选择 MTU3 引脚的寄存器。

#### MTUS<sub>i</sub> 位 (MTU3 引脚选择位) (i=0、1)

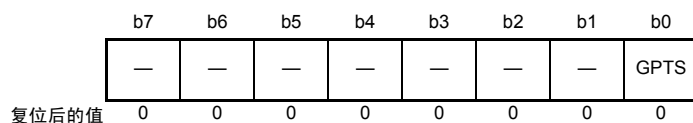
此位选择 MTU3 的输入 / 输出引脚。

#### TCLKS[1:0] 位 (MTCLK 引脚选择位)

此位选择 MTU3 的 MTCLK 输入引脚。

## 15.2.2.8 端口功能寄存器 D (PFDGPT)

地址 0008 C10Dh



位	符号	位名	功能	R/W
b0	GPTS	GPT 引脚选择位 6	0: 选择 P71 为 GTIOC0A-A 引脚 选择 P74 为 GTIOC0B-A 引脚 选择 P72 为 GTIOC1A-A 引脚 选择 P75 为 GTIOC1B-A 引脚 选择 P73 为 GTIOC2A-A 引脚 选择 P76 为 GTIOC2B-A 引脚 1: 选择 PD7 为 GTIOC0A-B 引脚 选择 PD6 为 GTIOC0B-B 引脚 选择 PD5 为 GTIOC1A-B 引脚 选择 PD4 为 GTIOC1B-B 引脚 选择 PD3 为 GTIOC2A-B 引脚 选择 PD2 为 GTIOC2B-B 引脚	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

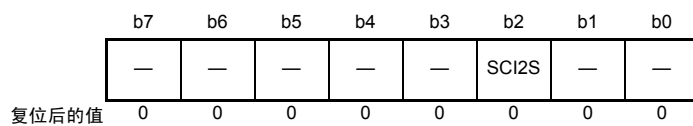
PFDGPT 寄存器是选择 GPT 引脚的寄存器。

## GPTS 位 (GPT 引脚选择位)

此位选择 GPT 的输入 / 输出引脚。

## 15.2.2.9 端口功能寄存器 F (PFFSCI)

地址 0008 C10Fh



位	符号	位名	功能	R/W
b1-b0	—	保留位	读写值都为“0”。	R/W
b2	SCI2S	SCI2 输入 / 输出引脚选择位	0: 将 PB6 设定为 RXD2-A 引脚 将 PB7 设定为 SCK2-A 引脚 将 PB5 设定为 TXD2-A 引脚 1: 将 P80 设定为 RXD2-B 引脚 将 P82 设定为 SCK2-B 引脚 将 P81 设定为 TXD2-B 引脚	R/W
b7	—	保留位	读写值都为“0”。	R/W

PFFSCI 寄存器是选择 SCI2 引脚的寄存器。

## SCI2S 位 (SCI2 输入 / 输出引脚选择位)

此位选择 SCI 通道 2 的输入 / 输出引脚。

## 15.2.2.10 端口功能寄存器 G (PFGSPI)

地址 0008 C110h

b7	b6	b5	b4	b3	b2	b1	b0
SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	—
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	RSPCKE	RSPCK 输出允许位	0: RSPCK 引脚无效 1: RSPCK 引脚有效	R/W
b2	MOSIE	MOSI 输出允许位	0: MOSI 引脚无效 1: MOSI 引脚有效	R/W
b3	MISOE	MISO 输出允许位	0: MISO 引脚无效 1: MISO 引脚有效	R/W
b4	SSL0E	SSL0 输出允许位	0: SSL0 引脚无效 1: SSL0 引脚有效	R/W
b5	SSL1E	SSL1 输出允许位	0: SSL1 引脚无效 1: SSL1 引脚有效	R/W
b6	SSL2E	SSL2 输出允许位	0: SSL2 引脚无效 1: SSL2 引脚有效	R/W
b7	SSL3E	SSL3 输出允许位	0: SSL3 引脚无效 1: SSL3 引脚有效	R/W

PFGSPI 寄存器是设定 RSPI 的相关输入 / 输出引脚的寄存器。

**RSPCKE 位 (RSPCK 输出允许位)**

此位选择允许或者禁止 RSPCK 引脚的输出。在使用 RSPCK 引脚时，必须将此位置“1”。

**MOSIE 位 (MOSI 输出允许位)**

此位选择允许或者禁止 MOSI 引脚的输出。在使用 MOSI 引脚时，必须将此位置“1”。

**MISOE 位 (MISO 输出允许位)**

此位选择允许或者禁止 MISO 引脚的输出。在使用 MISO 引脚时，必须将此位置“1”。

**SSL0E 位 (SSL0 输出允许位)**

此位选择允许或者禁止 SSL0 引脚的输出。在使用 SSL0 引脚时，必须将此位置“1”。

**SSL1E 位 (SSL1 输出允许位)**

此位选择允许或者禁止 SSL1 引脚的输出。在使用 SSL1 引脚时，必须将此位置“1”。

**SSL2E 位 (SSL2 输出允许位)**

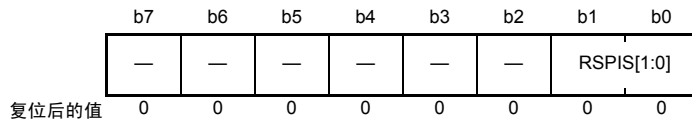
此位选择允许或者禁止 SSL2 引脚的输出。在使用 SSL2 引脚时，必须将此位置“1”。

**SSL3E 位 (SSL3 允许位)**

此位选择允许或者禁止 SSL3 引脚的输出。在使用 SSL3 引脚时，必须将此位置“1”。

## 15.2.2.11 端口功能寄存器 H (PFHSPI)

地址 0008 C111h



位	符号	位名	功能	R/W
b1-b0	RSPIS[1:0]	RSPI 引脚选择位	b1 b0 0 0: 将 P22 设定为 MISO-A 输入引脚 将 P23 设定为 MOSI-A 输入引脚 将 P24 设定为 RSPCK-A 输入引脚 将 P30 设定为 SSL0-A 输入引脚 将 P31 设定为 SSL1-A 输入引脚 将 P32 设定为 SSL2-A 输入引脚 将 P33 设定为 SSL3-A 输入引脚 0 1: 将 PA5 设定为 MISO-B 输入引脚 将 PB0 设定为 MOSI-B 输入引脚 将 PA4 设定为 RSPCK-B 输入引脚 将 PA3 设定为 SSL0-B 输入引脚 将 PA2 设定为 SSL1-B 输入引脚 将 PA1 设定为 SSL2-B 输入引脚 将 PA0 设定为 SSL3-B 输入引脚 1 0: 将 PD1 设定为 MISO-C 输入引脚 将 PD2 设定为 MOSI-C 输入引脚 将 PD0 设定为 RSPCK-C 输入引脚 将 PD6 设定为 SSL0-C 输入引脚 将 PD7 设定为 SSL1-C 输入引脚 将 PE0 设定为 SSL2-C 输入引脚 将 PE1 设定为 SSL3-C 输入引脚 1 1: 不能设定	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

PFHSPI 寄存器是设定 RSPI 的相关输入 / 输出引脚的寄存器。

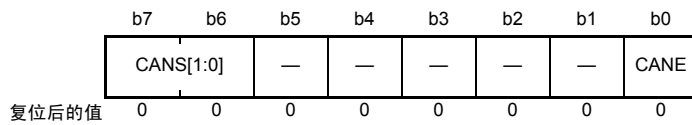
## RSPIS[1:0] 位 (RSPI 引脚选择位)

此位选择 RSPI 的输入 / 输出引脚。

RSPI 的各输入 / 输出引脚有允许位，选择允许位为“1”的引脚而不选择允许位为“0”的引脚。

## 15.2.2.12 端口功能寄存器 J (PFJCAN)

地址 0008 C113h



位	符号	位名	功能	R/W
b0	CANE	CAN 引脚允许位	0: CTX 引脚和 CRX 引脚无效 1: CTX 引脚和 CRX 引脚有效	R/W
b5-b1	—	保留位	读写值都为“0”。	R/W
b7-b6	CANS[1:0]	CAN 引脚选择位	b7 b6 00: 将 PB5 设定为 CTX-A 输入引脚 将 PB6 设定为 CRX-A 输入引脚 01: 将 P23 设定为 CTX-B 输入引脚 将 P22 设定为 CRX-B 输入引脚 10: 将 PE0 设定为 CRX-C 输入引脚 将 PD7 设定为 CTX-C 输入引脚 11: 不能设定	R/W

PFJCAN 寄存器是设定 CAN 的相关输入 / 输出引脚的寄存器。

## CANE 位 (CAN 引脚允许位)

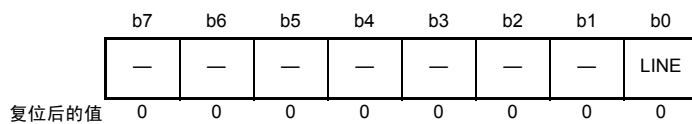
此位选择允许或者禁止 CAN 引脚。在使用 CAN 引脚时，必须将此位置“1”。

## CANS[1:0] 位 (CAN 引脚选择位)

选择 CAN 的输入 / 输出引脚。

## 15.2.2.13 端口功能寄存器 K (PFKLIN)

地址 0008 C114h



位	符号	位名	功能	R/W
b0	LINE	LIN 引脚允许位	0: LTX 引脚和 LRX 引脚无效 1: LTX 引脚和 LRX 引脚有效	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

PFKLIN 寄存器是设定 LIN 的相关输入 / 输出引脚的寄存器。

## LINE 位 (LIN 引脚允许位)

此位选择允许或者禁止 LIN 引脚。在使用 LIN 引脚时，必须将此位置“1”。

## 15.2.2.14 端口功能寄存器 M (PFMPOE)

地址 0008 C116h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	POE11E	POE10E	POE8E	POE4E	POE0E
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	POE0E	POE0# 输入允许位	0: 设定为 I/O 端口 1: 设定为 POEn# 输入引脚 (n=0、4、8、10、11)	R/W (注1)
b1	POE4E	POE4# 输入允许位		R/W (注1)
b2	POE8E	POE8# 输入允许位		R/W (注1)
b3	POE10E	POE10# 输入允许位		R/W (注1)
b4	POE11E	POE11# 输入允许位		R/W (注1)
b7-b5	—	保留位		读写值都为“0”。

注 1. 只有复位开始后的第 1 次写操作有效，不能进行第 2 次以后的写操作。

PFMPOE 寄存器是选择允许或者禁止 POEn# 输入引脚的寄存器。

为了防止系统的误动作，必须在复位开始后进行写操作。只有复位开始后的第 1 次写操作有效。

## POEnE 位 (POEn# 输入允许位) (n=0、4、8、10、11)

此位选择允许或者禁止对应的 POEn# 输入引脚。

在使用 POEn# 时，必须将对应的 POEnE 位置“1”。

## 15.2.2.15 端口功能寄存器 N (PFNPOE)

地址 0008 C117h

	b7	b6	b5	b4	b3	b2	b1	b0
	POE10S	—	—	—	—	—	—	—
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	POE10S	POE10# 输入选择位	0: 将 PE2 设定为 POE10#-A 输入引脚 1: 将 PE4 设定为 POE10#-B 输入引脚	R/W (注1)

注 1. 只有复位开始后的第 1 次写操作有效，不能进行第 2 次以后的写操作。

PFNPOE 寄存器是选择允许或者禁止 POEn# 输入引脚的寄存器。

为了防止系统的误动作，必须在复位开始后进行写操作。只有复位开始后的第 1 次写操作有效。

## POE10S 位 (POE10# 输入选择位)

此位选择 POE10# 的输入引脚。



### 15.2.3 端口的设定

如果将各外围模块的引脚设定为有效，就能转换各端口的设定。

能通过设定各外围模块，独立设定用作输入的引脚。除端口寄存器的读、NMI 引脚的输入和 POEn# 引脚的输入以外，需要通过将输入缓冲控制寄存器（PORTn.ICR）的对应位置“1”，将输入缓冲器设定为有效。

对于用作输出和输入 / 输出的引脚，需要将各外围模块引脚的输出信号设定为有效。如果被相同端口多路复用的各外围模块输出信号的有效设定发生竞争，就根据外围模块的端口多路复用优先级，优先的外围模块功能变为有效。

外围模块的端口多路复用优先级一览表如表 15.12 所示。

表 15.12 外围模块的端口多路复用优先级一览表（100 引脚 LQFP）

优先级	模块名	输出引脚名	
高 ↑       ↓ 低	1	RSPI	RSPCK、MOSI、MISO、SSL0 ~ SSL3
	2	CAN	CTX
	3	LIN	LTX
	4	MTU0 ~ MTU7	MTIOC0A、MTIOC0B、MTIOC0C、MTIOC0D、MTIOC1A、MTIOC1B、MTIOC2A、MTIOC2B、MTIOC3A、MTIOC3B、MTIOC3C、MTIOC3D、MTIOC4A、MTIOC4B、MTIOC4C、MTIOC4D、MTIOC6A、MTIOC6B、MTIOC6C、MTIOC6D、MTIOC7A、MTIOC7B、MTIOC7C、MTIOC7D
	5	GPT0 ~ GPT3	GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B、GTIOC2A、GTIOC2B、GTIOC3A、GTIOC3B
	6	SCI0 ~ SCI2	SCK0 ~ SCK2、TXD0 ~ TXD2
	7	RIIC	SCL、SDA
	8	IOPORT	P10 ~ P11、P20 ~ P24、P30 ~ P33、P70 ~ P76、P80 ~ P82、P90 ~ P96、PA0 ~ PA5、PB0 ~ PB7、PD0 ~ PD7、PE0 ~ PE1、PE3 ~ PE5

### 15.2.4 输出允许设定一览表

各端口的输出允许设定一览表如表 15.13 所示。

对应的输出信号的详细内容请参照各外围模块的寄存器说明。

能通过端口功能寄存器，更改各外围模块引脚名末尾带有 A ~ C 的任意一个字符的引脚功能。

表 15.13 各端口的输出允许设定一览表 (1/4) (100 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
P10	PORT1	P10		PORT1.DDR.B0=1
P11	PORT1	P11		PORT1.DDR.B1=1
P20	PORT2	P20		PORT2.DDR.B0=1
P21	PORT2	P21		PORT2.DDR.B1=1
P22	RSPI	MISO-A	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	PORT2	P22		PORT2.DDR.B2=1
P23	RSPI	MOSI-A	PFGSPI.MOSIE=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	CAN	CTX-B	PFJCAN.CANE =1 PFJCAN.CANS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	LIN	LTX	PFKLIN.LINE=1	(信号的输出状态取决于外围模块的设定)
	PORT2	P23		PORT2.DDR.B3=1
P24	RSPI	RSPCK-A	PFGSPI.RSPCKE=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	PORT2	P24		PORT2.DDR.B4=1
P30	RSPI	SSL0-A	PFGSPI.SSL0E=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	MTU0	MTIOC0B-B	PFCMTU.MTUS1=1	(信号的输出状态取决于外围模块的设定)
	PORT3	P30		PORT3.DDR.B0=1
P31	RSPI	SSL1-A	PFGSPI.SSL1E=1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU0	MTIOC0A-B	PFCMTU.MTUS0=1	(信号的输出状态取决于外围模块的设定)
	PORT3	P31		PORT3.DDR.B1=1
P32	RSPI	SSL2-A	PFGSPI.SSL2E=1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU3	MTIOC3C		(信号的输出状态取决于外围模块的设定)
	PORT3	P32		PORT3.DDR.B2=1
P33	RSPI	SSL3-A	PFGSPI.SSL3E=1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU3	MTIOC3A		(信号的输出状态取决于外围模块的设定)
	PORT3	P33		PORT3.DDR.B3=1
P70	PORT7	P70		PORT7.DDR.B0=1
P71	MTU3	MTIOC3B		(信号的输出状态取决于外围模块的设定)
	GPT0	GTIOC0A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P71		PORT7.DDR.B1=1
P72	MTU4	MTIOC4A		(信号的输出状态取决于外围模块的设定)
	GPT1	GTIOC1A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P72		PORT7.DDR.B2=1
P73	MTU4	MTIOC4B		(信号的输出状态取决于外围模块的设定)
	GPT2	GTIOC2A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P73		PORT7.DDR.B3=1
P74	MTU3	MTIOC3D		(信号的输出状态取决于外围模块的设定)
	GPT0	GTIOC0B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P74		PORT7.DDR.B4=1

表 15.13 各端口的输出允许设定一览表 (2/4) (100 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
P75	MTU4	MTIOC4C		(信号的输出状态取决于外围模块的设定)
	GPT1	GTIOC1B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P75		PORT7.DDR.B5=1
P76	MTU4	MTIOC4D		(信号的输出状态取决于外围模块的设定)
	GPT2	GTIOC2B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P76		PORT7.DDR.B6=1
P80	PORT8	P80		PORT8.DDR.B0=1
P81	SCI2	TXD2-B	PFFSCI.SCI2S=1	SCI2.SCR.TE=1
	PORT8	P81		PORT8.DDR.B1=1
P82	SCI2	SCK2-B	PFFSCI.SCI2S=1	当 SCI2.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1; 当 SCI2.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORT8	P82		PORT8.DDR.B2=1
P90	MTU7	MTIOC7D		(信号的输出状态取决于外围模块的设定)
	PORT9	P90		PORT9.DDR.B0=1
P91	MTU7	MTIOC7C		(信号的输出状态取决于外围模块的设定)
	PORT9	P91		PORT9.DDR.B1=1
P92	MTU6	MTIOC6D		(信号的输出状态取决于外围模块的设定)
	PORT9	P92		PORT9.DDR.B2=1
P93	MTU7	MTIOC7B		(信号的输出状态取决于外围模块的设定)
	PORT9	P93		PORT9.DDR.B3=1
P94	MTU7	MTIOC7A		(信号的输出状态取决于外围模块的设定)
	PORT9	P94		PORT9.DDR.B4=1
P95	MTU6	MTIOC6B		(信号的输出状态取决于外围模块的设定)
	PORT9	P95		PORT9.DDR.B5=1
P96	PORT9	P96		PORT9.DDR.B6=1
PA0	RSPI	SSL3-B	PFGSPI.SSL3E=1 PFHSPI.RSPIS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	MTU6	MTIOC6C		(信号的输出状态取决于外围模块的设定)
	PORTA	PA0		PORTA.DDR.B0=1
PA1	RSPI	SSL2-B	PFGSPI.SSL2E=1 PFHSPI.RSPIS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	MTU6	MTIOC6A		(信号的输出状态取决于外围模块的设定)
	PORTA	PA1		PORTA.DDR.B1=1
PA2	RSPI	SSL1-B	PFGSPI.SSL1E=1 PFHSPI.RSPIS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	MTU2	MTIOC2B		(信号的输出状态取决于外围模块的设定)
	PORTA	PA2		PORTA.DDR.B2=1
PA3	RSPI	SSL0-B	PFGSPI.SSL0E=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入 / 输出转换功能。)
	MTU2	MTIOC2A		(信号的输出状态取决于外围模块的设定)
	PORTA	PA3		PORTA.DDR.B3=1
PA4	RSPI	RSPCK-B	PFGSPI.RSPCKE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入 / 输出转换功能。)
	MTU1	MTIOC1B		(信号的输出状态取决于外围模块的设定)
	PORTA	PA4		PORTA.DDR.B4=1

表 15.13 各端口的输出允许设定一览表 (3/4) (100 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
PA5	RSPI	MISO-B	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU1	MTIOC1A		(信号的输出状态取决于外围模块的设定)
	PORTA	PA5		PORTA.DDR.B5=1
PB0	RSPI	MOSI-B	PFGSPI.MOSIE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU0	MTIOC0D		(信号的输出状态取决于外围模块的设定)
	PORTB	PB0		PORTB.DDR.B0=1
PB1	MTU0	MTIOC0C		(信号的输出状态取决于外围模块的设定)
	RIIC	SCL		RIIC.ICCR1.ICE=1
	PORTB	PB1		PORTB.DDR.B1=1
PB2	MTU0	MTIOC0B-A	PFCMTU.MTUS1=0	(信号的输出状态取决于外围模块的设定)
	SCI0	TXD0		SCI0.SCR.TE=1
	RIIC	SDA		RIIC.ICCR1.ICE=1
	PORTB	PB2		PORTB.DDR.B2=1
PB3	MTU0	MTIOC0A-A	PFCMTU.MTUS0=0	(信号的输出状态取决于外围模块的设定)
	SCI0	SCK0		当 SCI0.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1; 当 SCI0.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTB	PB3		PORTB.DDR.B3=1
PB4	PORTB	PB4		PORTB.DDR.B4=1
PB5	CAN	CTX-A	PFJCAN.CANE=1 PFJCAN.CANS[1:0] = 00	(信号的输出状态取决于外围模块的设定)
	SCI2	TXD2-A	PFFSCI.SCI2S = 0	SCI2.SCR.TE=1
	PORTB	PB5		PORTB.DDR.B5=1
PB6	PORTB	PB6		PORTB.DDR.B6=1
PB7	SCI2	SCK2-A	PFFSCI.SCI2S=0	当 SCI2.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1; 当 SCI2.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTB	PB7		PORTB.DDR.B7=1
PD0	RSPI	RSPCK-C	PFGSPI.RSPCKE=1 PFHSPI.RSPIS[1:0]=10	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	GPT3	GTIOC3B		(信号的输出状态取决于外围模块的设定)
	PORTD	PD0		PORTD.DDR.B0=1
PD1	RSPI	MISO-C	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=10	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	GPT3	GTIOC3A		(信号的输出状态取决于外围模块的设定)
	PORTD	PD1		PORTD.DDR.B1=1
PD2	RSPI	MOSI-C	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=10	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	GPT2	GTIOC2B-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD2		PORTD.DDR.B2=1

表 15.13 各端口的输出允许设定一览表 (4/4) (100 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
PD3	GPT2	GTIOC2A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	SCI1	TXD1		SCI1.SCR.TE=1
	PORTD	PD3		PORTD.DDR.B3=1
PD4	GPT1	GTIOC1B-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	SCI1	SCK1		当 SCI1.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1; 当 SCI1.SCMR.SMIF=0 时 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTD	PD4		PORTD.DDR.B4=1
PD5	GPT1	GTIOC1A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD5		PORTD.DDR.B5=1
PD6	RSPI	SSL0-C	PFGSPI.SSL0E=1 PFHSPI.RSPIS[1:0]=10	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	GPT0	GTIOC0B-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD6		PORTD.DDR.B6=1
PD7	RSPI	SSL1-C	PFGSPI.SSL1E=1 PFHSPI.RSPIS[1:0]=10	(信号的输出状态取决于外围模块的设定)
	CAN	CTX-C	PFJCAN.CANE=1 PFJCAN.CANS[1:0]=10	(信号的输出状态取决于外围模块的设定)
	GPT0	GTIOC0A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD7		PORTD.DDR.B7=1
PE0	RSPI	SSL2-C	PFGSPI.SSL2E=1 PFHSPI.RSPIS[1:0]=10	(信号的输出状态取决于外围模块的设定)
	PORTE	PE0		PORTE.DDR.B0=1
PE1	RSPI	SSL3-C	PFGSPI.SSL3E=1 PFHSPI.RSPIS[1:0]=10	(信号的输出状态取决于外围模块的设定)
	PORTE	PE1		PORTE.DDR.B1=1
PE3	PORTE	PE3		PORTE.DDR.B3=1
PE4	PORTE	PE4		PORTE.DDR.B4=1
PE5	PORTE	PE5		PORTE.DDR.B5=1

### 15.2.5 未使用引脚的处理

未使用引脚的处理内容如表 15.14 所示。

表 15.14 未使用引脚的处理内容（100 引脚 LQFP）

引脚名	处理内容
EMLE	通过电阻连接 VSS（下拉）。
MD1、MD0	（必须用作模式引脚。）
MDE	（必须用作模式引脚。）
RES#	通过电阻连接 VCC（上拉）。
PE2/NMI/POE10#-A	通过电阻连接 VCC（上拉）。
EXTAL	（必须用作时钟引脚。）
XTAL	将引脚置为开路。
端口 1 ~ 3、7 ~ 9、 A、B、D、E	<ul style="list-style-type: none"> <li>每个引脚通过电阻连接 VCC（上拉）或者通过电阻连接 VSS（下拉）。</li> <li>也能在 PORTn.ICR 为初始值（输入缓冲器无效）的状态下将引脚置为开路（注 1）。</li> </ul>
端口 4	<ul style="list-style-type: none"> <li>每个引脚通过电阻连接 AVCC0（上拉）或者通过电阻连接 AVSS0（下拉）。</li> <li>也能在 PORTn.ICR 为初始值（输入缓冲器无效）的状态下将引脚置为开路（注 1）。</li> </ul>
端口 5 ~ 6	<ul style="list-style-type: none"> <li>每个引脚通过电阻连接 AVCC（上拉）或者通过电阻连接 AVSS（下拉）。</li> <li>也能在 PORTn.ICR 为初始值（输入缓冲器无效）的状态下将引脚置为开路（注 1）。</li> </ul>
VREFH0	连接 AVCC0。
VREFL0	连接 AVSS0。
VREF	连接 AVCC

注 1. 不能改变 PORTn.ICR 寄存器的初始值，否则就可能有穿透电流流过。

### 15.3 80 引脚 LQFP 的 I/O 端口

RX62T 群的 80 引脚 LQFP 的 I/O 端口由端口 1 ~ 4、6、7、9、A、B、D、E 共 11 个端口构成，有 44 个输入 / 输出引脚。

#### 15.3.1 概要

I/O 端口的规格和端口功能一览表分别如表 15.15 和表 15.16 所示。

表 15.15 I/O 端口的规格（80 引脚 LQFP）

项目	内容
输入 / 输出引脚	44 个
输入引脚	13 个
端口	11 个端口（1 ~ 4、6、7、9、A、B、D、E）
漏极开路输出	2 个（RIIC 引脚）
大电流输出	6 个（MTU3 引脚和 GPT 引脚）
施密特触发输入引脚	全部端口输入、CAN 输入、IRQ 输入、MTU3 输入、POE3 输入、RIIC 输入、SCI 输入、A/D 触发输入、NMI 输入、GPT 输入、LIN 输入
其他	<ul style="list-style-type: none"> <li>能驱动 1 个 TTL 负载和 30pF 的电容负载。</li> <li>在输出时驱动达林顿晶体管。</li> <li>能随时读引脚的状态。</li> </ul>

表 15.16 端口功能一览表 (1/2) (80 引脚 LQFP)

端口	概要	位	功能			CMOS 输入引脚	施密特触发 输入引脚	漏极开路 输出功能	大电流 输出引脚	
			输入 / 输出	输入	输出					
端口 1	与 MTU3 输入 和中断输入兼 用的通用输入 / 输出端口	0	P10	MTCLKD-B/IRQ0-A		—	全部输入功能	—	—	
		1	P11	MTCLKC-B/IRQ1-A						
端口 2	与 MTU3 输入、 中断输入、 A/D 转换器输入、 RSPI 输入 / 输出、LIN 输入 / 输出和 CAN 输入 / 输出 兼用的通用 输入 / 输出端口	0	P20	MTCLKB-B/IRQ7/ ADTRG0#-B		—	全部输入功能	—	—	
		1	P21	MTCLKA-B/IRQ6/ ADTRG1#-B						
		2	P22/MISO-A	LRX/CRX-B/ ADTRG#	LTX/CTX-B	MISO-A	P22/LRX/CRX-B/ ADTRG#			
		3	P23/MOSI-A			MOSI-A	P23			
		4	P24/RSPCK-A			RSPCK-A	P24			
端口 3	与 MTU3 输入 / 输出和 RSPI 输入 / 输出兼用 的通用输入 / 输出 端口	0	P30/MTIOC0B-B/ SSL0-A	MTCLKD-A		SSL0-A	P30/MTIOC0B-B/ MTCLKD-A	—	—	
		1	P31/MTIOC0A-B	MTCLKC-A	SSL1-A	—	全部输入功能	—	—	
		2	P32/MTIOC3C	MTCLKB-A	SSL2-A		全部输入功能			
		3	P33/MTIOC3A	MTCLKA-A	SSL3-A		全部输入功能			
端口 4	与 A/D 转换器 输入兼用的通用 输入端口	0		P40/AN000		—	P40	—	—	
		1		P41/AN001			P41			
		2		P42/AN002			P42			
		3		P43/AN003/CVREFL			P43			
		4		P44/AN100			P44			
		5		P45/AN101			P45			
		6		P46/AN102			P46			
		7		P47/AN103/ CVREFH			P47			
端口 6	与 A/D 转换器 输入兼用的通用 输入端口	0		P60/AN0		—	P60	—	—	
		1		P61/AN1			P61			
		2		P62/AN2			P62			
		3		P63/AN3			P63			
端口 7	与 POE3 输入、 中断输入、 MTU3 输入 / 输出 和 GPT 输入 / 输出 兼用的通用输入 / 输出 端口	0	P70	POE0#/IRQ5		—	全部输入功能	—	—	
		1	P71/MTIOC3B/ GTIOC0A-A				全部输入功能			○
		2	P72/MTIOC4A/ GTIOC1A-A				全部输入功能			○
		3	P73/MTIOC4B/ GTIOC2A-A				全部输入功能			○
		4	P74/MTIOC3D/ GTIOC0B-A				全部输入功能			○
		5	P75/MTIOC4C/ GTIOC1B-A				全部输入功能			○
		6	P76/MTIOC4D/ GTIOC2B-A				全部输入功能			○
端口 9	与 MTU3 输入 / 输出、POE3 输入和中断输入 兼用的通用 输入 / 输出端口	1	P91/MTIOC7C			—	全部输入功能	—	—	
		2	P92/MTIOC6D				全部输入功能			
		3	P93/MTIOC7B				全部输入功能			
		4	P94/MTIOC7A				全部输入功能			
		5	P95/MTIOC6B				全部输入功能			
		6	P96	POE4#/IRQ4			全部输入功能			
端口 A	与 MTU3 输入 / 输出、RSPI 输入 / 输出和 A/D 转换器输入兼 用的通用输入 / 输出端口	2	PA2/MTIOC2B		SSL1-B	—	全部输入功能	—	—	
		3	PA3/MTIOC2A/ SSL0-B				SSL0-B			PA3/MTIOC2A
		4	PA4/MTIOC1B/ RSPCK-B	ADTRG0#-A			RSPCK-B			PA4/MTIOC1B/ ADTRG0#-A
		5	PA5/MTIOC1A/ MISO-B	ADTRG1#-A			MISO-B			PA5/MTIOC1A/ ADTRG1#-A



表 15.16 端口功能一览表 (2/2) (80 引脚 LQFP)

端口	概要	位	功能			CMOS 输入引脚	施密特触发 输入引脚	漏极开路 输出功能	大电流 输出引脚
			输入 / 输出	输入	输出				
端口 B	与 MTU3 输入 / 输出、RSPI 输入 / 输出、RIIC 输入 / 输出、SCI 输入 / 输出、GPT 输入、POE3 输入、中断输入和 CAN 输入 / 输出兼用的通用输入 / 输出端口	0	PB0/MTIOC0D/ MOSI-B			MOSI-B	PB0/MTIOC0D	—	—
		1	PB1/MTIOC0C/ SCL	RXD0		—	全部输入功能	○ (仅 SCI)	—
		2	PB2/MTIOC0B-A/ SDA		TXD0		全部输入功能	○ (仅 SDA)	—
		3	PB3/MTIOC0A-A/ SCK0				全部输入功能	—	—
		4	PB4	GTETRQ/ POE8#/IRQ3			全部输入功能		
		5	PB5		TXD2-A/ CTX-A/		全部输入功能		
		6	PB6	RXD2-A/CRX-A			全部输入功能		
		7	PB7/SCK2-A				全部输入功能		
端口 D	与 GPT 输入 / 输出、SCI 输入 / 输出、CAN 输出和内部调试器输入 / 输出兼用的通用输入 / 输出端口	3	PD3/GTIOC2A-B		TXD1/TDO	—	全部输入功能	—	—
		4	PD4/GTIOC1B-B/ SCK1	TCK		—	全部输入功能		
		5	PD5/GTIOC1A-B	RXD1/TDI		—	全部输入功能		
		6	PD6/GTIOC0B-B	TMS			全部输入功能		
		7	PD7/GTIOC0A-B	TRST#	CTX-C	—	全部输入功能		
端口 E	与 CAN 输入、POE3 输入、MTU3 输入和中断输入兼用的输入 / 输出端口	0	PE0	CRX-C		—	全部输入功能	—	—
		2		PE2/POE10#-A/NMI			全部输入功能		
		3	PE3	MTCLKD-C/ POE11#/IRQ2-A			全部输入功能		
		4	PE4	MTCLKC-C/ POE10#-B/ IRQ1-B			全部输入功能		

### 15.3.2 寄存器说明

I/O 端口的寄存器一览表和各寄存器的有效位一览表分别如表 15.17 和表 15.18 所示。

表 15.17 I/O 端口的寄存器一览表 (1/2) (80 引脚 LQFP)

端口符号	寄存器名	寄存器符号	复位后的值	地址	存取长度
PORT1	数据方向寄存器	DDR	00h	0008 C001h	8
	数据寄存器	DR	00h	0008 C021h	8
	端口寄存器	PORT	不定值	0008 C041h	8
	输入缓冲控制寄存器	ICR	00h	0008 C061h	8
PORT2	数据方向寄存器	DDR	00h	0008 C002h	8
	数据寄存器	DR	00h	0008 C022h	8
	端口寄存器	PORT	不定值	0008 C042h	8
	输入缓冲控制寄存器	ICR	00h	0008 C062h	8
PORT3	数据方向寄存器	DDR	00h	0008 C003h	8
	数据寄存器	DR	00h	0008 C023h	8
	端口寄存器	PORT	不定值	0008 C043h	8
	输入缓冲控制寄存器	ICR	00h	0008 C063h	8
PORT4	端口寄存器	PORT	不定值	0008 C044h	8
	输入缓冲控制寄存器	ICR	00h	0008 C064h	8
PORT6	端口寄存器	PORT	不定值	0008 C046h	8
	输入缓冲控制寄存器	ICR	00h	0008 C066h	8
PORT7	数据方向寄存器	DDR	00h	0008 C007h	8
	数据寄存器	DR	00h	0008 C027h	8
	端口寄存器	PORT	不定值	0008 C047h	8
	输入缓冲控制寄存器	ICR	00h	0008 C067h	8
PORT9	数据方向寄存器	DDR	00h	0008 C009h	8
	数据寄存器	DR	00h	0008 C029h	8
	端口寄存器	PORT	不定值	0008 C049h	8
	输入缓冲控制寄存器	ICR	00h	0008 C069h	8
PORTA	数据方向寄存器	DDR	00h	0008 C00Ah	8
	数据寄存器	DR	00h	0008 C02Ah	8
	端口寄存器	PORT	不定值	0008 C04Ah	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Ah	8
PORTB	数据方向寄存器	DDR	00h	0008 C00Bh	8
	数据寄存器	DR	00h	0008 C02Bh	8
	端口寄存器	PORT	不定值	0008 C04Bh	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Bh	8
PORTD	数据方向寄存器	DDR	00h	0008 C00Dh	8
	数据寄存器	DR	00h	0008 C02Dh	8
	端口寄存器	PORT	不定值	0008 C04Dh	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Dh	8

表 15.17 I/O 端口的寄存器一览表 (2/2) (80 引脚 LQFP)

端口符号	寄存器名	寄存器符号	复位后的值	地址	存取长度
PORTE	数据方向寄存器	DDR	00h	0008 C00Eh	8
	数据寄存器	DR	00h	0008 C02Eh	8
	端口寄存器	PORT	不定值	0008 C04Eh	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Eh	8
IOPORT	端口功能寄存器 8	PF8IRQ	00h	0008 C108h	8
	端口功能寄存器 A	PFAADC	00h	0008 C10Ah	8
	端口功能寄存器 C	PFCMTU	00h	0008 C10Ch	8
	端口功能寄存器 D	PFDGPT	00h	0008 C10Dh	8
	端口功能寄存器 G	PFGSPI	00h	0008 C110h	8
	端口功能寄存器 H	PFHSPI	00h	0008 C111h	8
	端口功能寄存器 J	PFJCAN	00h	0008 C113h	8
	端口功能寄存器 K	PFKLIN	00h	0008 C114h	8
	端口功能寄存器 M	PFMPOE	00h	0008 C116h	8
	端口功能寄存器 N	PFNPOE	00h	0008 C117h	8

表 15.18 各寄存器的有效位一览表 (1/2) (100 引脚 LQFP)

寄存器符号	b7	b6	b5	b4	b3	b2	b1	b0
PORT1.DDR	x	x	x	x	x	x	○	○
PORT2.DDR	x	x	x	○	○	○	○	○
PORT3.DDR	x	x	x	x	○	○	○	○
PORT7.DDR	x	○	○	○	○	○	○	○
PORT9.DDR	x	○	○	○	○	○	○	x
PORTA.DDR	x	x	○	○	○	○	x	x
PORTB.DDR	○	○	○	○	○	○	○	○
PORTD.DDR	○	○	○	○	○	x	x	x
PORTE.DDR	x	x	x	○	○	x	x	○
PORT1.DR	x	x	x	x	x	x	○	○
PORT2.DR	x	x	x	○	○	○	○	○
PORT3.DR	x	x	x	x	○	○	○	○
PORT7.DR	x	○	○	○	○	○	○	○
PORT9.DR	x	○	○	○	○	○	○	x
PORTA.DR	x	x	○	○	○	○	x	x
PORTB.DR	○	○	○	○	○	○	○	○
PORTD.DR	○	○	○	○	○	x	x	x
PORTE.DR	x	x	x	○	○	x	x	○
PORT1.PORT	x	x	x	x	x	x	○	○
PORT2.PORT	x	x	x	○	○	○	○	○
PORT3.PORT	x	x	x	x	○	○	○	○
PORT4.PORT	○	○	○	○	○	○	○	○
PORT6.PORT	x	x	x	x	○	○	○	○
PORT7.PORT	x	○	○	○	○	○	○	○

表 15.18 各寄存器的有效位一览表 (2/2) (100 引脚 LQFP)

寄存器符号	b7	b6	b5	b4	b3	b2	b1	b0
PORT9.PORT	x	○	○	○	○	○	○	x
PORTA.PORT	x	x	○	○	○	○	x	x
PORTB.PORT	○	○	○	○	○	○	○	○
PORTD.PORT	○	○	○	○	○	x	x	x
PORTE.PORT	x	x	x	○	○	○	x	○
PORT1.ICR	x	x	x	x	x	x	○	○
PORT2.ICR	x	x	x	○	○	○	○	○
PORT3.ICR	x	x	x	x	○	○	○	○
PORT4.ICR	○	○	○	○	○	○	○	○
PORT6.ICR	x	x	x	x	○	○	○	○
PORT7.ICR	x	○	○	○	○	○	○	○
PORT9.ICR	x	○	○	○	○	○	○	x
PORTA.ICR	x	x	○	○	○	○	x	x
PORTB.ICR	○	○	○	○	○	○	○	○
PORTD.ICR	○	○	○	○	○	x	x	x
PORTE.ICR	x	x	x	○	○	x	x	○
IOPORT.PF8IRQ	x	x	x	x	○	○	○	○
IOPORT.PFAADC	x	x	x	x	x	x	○	○
IOPORT.PFCMTU	○	○	x	x	x	x	○	○
IOPORT.PFDGPT	x	x	x	x	x	x	x	○
IOPORT.PFGSPI	○	○	○	○	○	○	○	x
IOPORT.PFHSPi	x	x	x	x	x	x	○	○
IOPORT.PFJCAN	○	○	x	x	x	x	x	○
IOPORT.PFKLIN	x	x	x	x	x	x	x	○
IOPORT.PFMPOE	x	x	x	○	○	○	○	○
IOPORT.PFNPOE	○	x	x	x	x	x	x	x

○: 有效位、x: 无效位 (保留位)

## 15.3.2.1 数据方向寄存器 (DDR)

地址 PORT1.DDR 0008 C001h、PORT2.DDR 0008 C002h、PORT3.DDR 0008 C003h、PORT7.DDR 0008 C007h、PORT9.DDR 0008 C009h、PORTA.DDR 0008 C00Ah、PORTB.DDR 0008 C00Bh、PORTD.DDR 0008 C00Dh、PORTE.DDR 0008 C00Eh

	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

- 注 1. PORT1.DDR 的低 2 位有效，高 6 位是保留位。  
 PORT2.DDR 的低 5 位有效，高 3 位是保留位。  
 PORT3.DDR 的低 4 位有效，高 4 位是保留位。  
 PORT7.DDR 的低 7 位有效，高 1 位是保留位。  
 PORT9.DDR 的 b6 ~ b1 有效，b7、b0 是保留位。  
 PORTA.DDR 的 b5 ~ b2 有效，b7、b6、b1、b0 是保留位。  
 PORTD.DDR 的高 5 位有效，低 3 位是保留位。  
 PORTE.DDR 的 b4、b3、b0 有效，b7 ~ b5、b2、b1 是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0	Pn0 输入 / 输出指定位	0: 输入端口 1: 输出端口	R/W
b1	B1	Pn1 输入 / 输出指定位		R/W
b2	B2	Pn2 输入 / 输出指定位		R/W
b3	B3	Pn3 输入 / 输出指定位		R/W
b4	B4	Pn4 输入 / 输出指定位		R/W
b5	B5	Pn5 输入 / 输出指定位		R/W
b6	B6	Pn6 输入 / 输出指定位		R/W
b7	B7	Pn7 输入 / 输出指定位		R/W

n=1 ~ 3、7、9、A、B、D、E

PORTn.DDR 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。

PORTn.DDR 寄存器 (n=1 ~ 3、7、9、A、B、D、E) 的各位分别对应端口 n 的各引脚，并且能以 1 位为单位进行指定。

## 15.3.2.2 数据寄存器 (DR)

地址 PORT1.DR 0008 C021h、PORT2.DR 0008 C022h、PORT3.DR 0008 C023h、PORT7.DR 0008 C027h、PORT9.DR 0008 C029h、  
PORTA.DR 0008 C02Ah、PORTB.DR 0008 C02Bh、PORTD.DR 0008 C02Dh、PORTE.DR 0008 C02Eh

	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

- 注 1. PORT1.DR 的低 2 位有效，高 6 位是保留位。  
 PORT2.DR 的低 5 位有效，高 3 位是保留位。  
 PORT3.DR 的低 4 位有效，高 4 位是保留位。  
 PORT7.DR 的低 7 位有效，高 1 位是保留位。  
 PORT9.DR 的 b6 ~ b1 有效，b7、b0 是保留位。  
 PORTA.DR 的 b5 ~ b2 有效，b7、b6、b1、b0 是保留位。  
 PORTD.DR 的高 5 位有效，低 3 位是保留位。  
 PORTE.DR 的 b4、b3、b0 有效，b7 ~ b5、b2、b1 是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0	Pn0 输出数据保存位	保存输出数据。	R/W
b1	B1	Pn1 输出数据保存位		R/W
b2	B2	Pn2 输出数据保存位		R/W
b3	B3	Pn3 输出数据保存位		R/W
b4	B4	Pn4 输出数据保存位		R/W
b5	B5	Pn5 输出数据保存位		R/W
b6	B6	Pn6 输出数据保存位		R/W
b7	B7	Pn7 输出数据保存位		R/W

n=1 ~ 3、7、9、A、B、D、E

PORTn.DR 寄存器 (n=1 ~ 3、7、9、A、B、D、E) 保存用作通用输出端口的引脚输出数据。

## 15.3.2.3 端口寄存器 (PORT)

地址 PORT1.PORT 0008 C041h、PORT2.PORT 0008 C042h、PORT3.PORT 0008 C043h、PORT4.PORT 0008 C044h、  
PORT6.PORT 0008 C046h、PORT7.PORT 0008 C047h、PORT9.PORT 0008 C049h、PORTA.PORT 0008 C04Ah、  
PORTB.PORT 0008 C04Bh、PORTD.PORT 0008 C04Dh、PORTE.PORT 0008 C04Eh

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

复位后的值 x x x x x x x x

- 注 1. PORT1.PORT 的低 2 位有效，高 6 位是保留位。  
PORT2.PORT 的低 5 位有效，高 3 位是保留位。  
PORT3.PORT 的低 4 位有效，高 4 位是保留位。  
PORT6.PORT 的低 4 位有效，高 4 位是保留位。  
PORT7.PORT 的低 7 位有效，高 1 位是保留位。  
PORT9.PORT 的 b6~b1 有效，b7、b0 是保留位。  
PORTA.PORT 的 b5~b2 有效，b7、b6、b1、b0 是保留位。  
PORTD.PORT 的高 5 位有效，低 3 位是保留位。  
PORTE.PORT 的 b4~b2、b0 有效，b7~b5、b1 是保留位。
- 注 2. 保留位。读取值为“1”，写操作无效。

位	符号	位名	功能	R/W
b0	B0	Pn0 位	反映端口的引脚状态。	R
b1	B1	Pn1 位		R
b2	B2	Pn2 位		R
b3	B3	Pn3 位		R
b4	B4	Pn4 位		R
b5	B5	Pn5 位		R
b6	B6	Pn6 位		R
b7	B7	Pn7 位		R

n=1~4、6、7、9、A、B、D、E

PORTn.PORT 寄存器是反映端口引脚状态的寄存器。

如果读 PORTn.PORT 寄存器 (n=1~4、6、7、9、A、B、D、E)，就能读到引脚的状态。

## 15.3.2.4 输入缓冲控制寄存器 (ICR)

地址 PORT1.ICR 0008 C061h、PORT2.ICR 0008 C062h、PORT3.ICR 0008 C063h、PORT4.ICR 0008 C064h、PORT6.ICR 0008 C066h、PORT7.ICR 0008 C067h、PORT9.ICR 0008 C069h、PORTA.ICR 0008 C06Ah、PORTB.ICR 0008 C06Bh、PORTD.ICR 0008 C06Dh、PORTE.ICR 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	0

- 注 1. PORT1.ICR 的低 2 位有效, 高 6 位是保留位。  
 PORT2.ICR 的低 5 位有效, 高 3 位是保留位。  
 PORT3.ICR 的低 4 位有效, 高 4 位是保留位。  
 PORT6.ICR 的低 4 位有效, 高 4 位是保留位。  
 PORT7.ICR 的低 7 位有效, 高 1 位是保留位。  
 PORT9.ICR 的 b6~b1 有效, b7、b0 是保留位。  
 PORTA.ICR 的 b5~b2 有效, b7、b6、b1、b0 是保留位。  
 PORTD.ICR 的高 5 位有效, 低 3 位是保留位。  
 PORTE.ICR 的 b4、b3、b0 有效, b7~b5、b2、b1 是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0 (注 1)	Pn0 输入缓冲器控制位	0: 对应引脚的输入缓冲器无效 1: 对应引脚的输入缓冲器有效	R/W
b1	B1 (注 1)	Pn1 输入缓冲器控制位		R/W
b2	B2 (注 1)	Pn2 输入缓冲器控制位		R/W
b3	B3 (注 1)	Pn3 输入缓冲器控制位		R/W
b4	B4 (注 1)	Pn4 输入缓冲器控制位		R/W
b5	B5 (注 1)	Pn5 输入缓冲器控制位		R/W
b6	B6 (注 1)	Pn6 输入缓冲器控制位		R/W
b7	B7 (注 1)	Pn7 输入缓冲器控制位		R/W

n=1~4、6、7、9、A、B、D、E

注 1. 在用作输入引脚时, 必须将对应位置“1”。必须将不用作输入的引脚和用作模拟输入引脚的位置“0”。

PORTn.ICR 寄存器是控制端口输入缓冲器的寄存器。

PORTn.ICR 寄存器 (n=1~4、6、7、9、A、B、D、E) 的各位分别对应端口 n 的各引脚, 并且能以 1 位为单位进行指定。

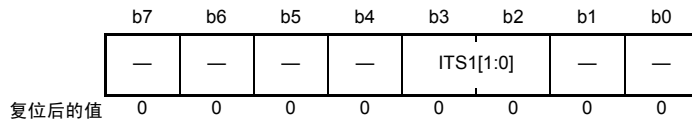
在用作外围模块的输入引脚时, 为了使对应引脚的输入缓冲器有效, 需要预先将 PORTn.ICR 置“1”。如果在将 PORTn.ICR 置“0”的状态下用作外围模块的输入引脚, 就将输入到外围模块的信号固定为 High 电平。

在更改 PORTn.ICR 寄存器的设定时, 有可能因引脚的状态而在内部产生边沿, 因此必须在不使用该输入引脚时更改 PORTn.ICR 寄存器的设定。例如, 在 IRQ<sub>i</sub> (i=0~7) 输入时, 必须在禁止该中断的状态下更改 PORTn.ICR 寄存器的设定。先将中断控制器的 IR<sub>i</sub>.IR 标志 (i=64~71 (IRQ 的中断向量号)) 置“0”, 然后允许该中断。如果在更改 PORTn.ICR 寄存器的设定后产生边沿, 就必须取消该边沿。



## 15.3.2.5 端口功能寄存器 8 (PF8IRQ)

地址 0008 C108h



位	符号	位名	功能	R/W
b1-b0	—	保留位	读写值都为“0”。	R/W
b3-b2	ITS1[1:0]	IRQ1 引脚选择位	b3 b2 0 0: 将 P11 设定为 IRQ1-A 输入引脚 0 1: 将 PE4 设定为 IRQ1-B 输入引脚 1 0: 保留 1 1: 保留	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

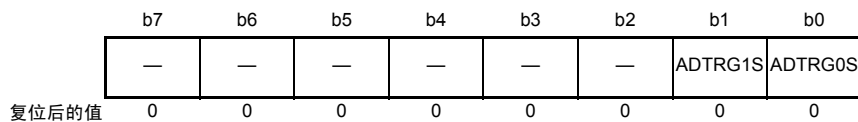
PF8IRQ 寄存器是选择 IRQ1 输入引脚的寄存器。

## ITS1 位 (IRQ1 引脚选择位)

此位选择 IRQ1 的输入引脚。

## 15.3.2.6 端口功能寄存器 A (PFAADC)

地址 0008 C10Ah



位	符号	位名	功能	R/W
b0	ADTRG0S	ADTRG0# 输入选择位	0: 将 PA4 设定为 ADTRG0#-A 输入引脚 1: 将 P20 设定为 ADTRG0#-B 输入引脚	R/W
b1	ADTRG1S	ADTRG1# 输入选择位	0: 将 PA5 设定为 ADTRG1#-A 输入引脚 1: 将 P21 设定为 ADTRG1#-B 输入引脚	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

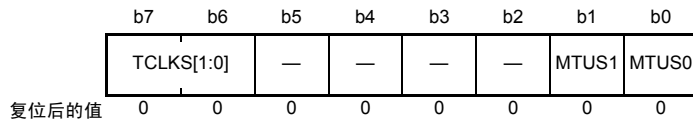
PFAADC 寄存器是选择 ADTRG0#、ADTRG1# 引脚的寄存器。

## ADTRGnS 位 (ADTRGn# 输入选择位) (n=0、1)

此位选择 ADTRGn# 的输入引脚。

## 15.3.2.7 端口功能寄存器 C (PFCMTU)

地址 0008 C10Ch



位	符号	位名	功能	R/W
b0	MTUS0	MTU3 引脚选择 0 位	0: 选择 PB3 为 MTIOC0A-A 引脚 1: 选择 P31 为 MTIOC0A-B 引脚	R/W
b1	MTUS1	MTU3 引脚选择 1 位	0: 选择 PB2 为 MTIOC0B-A 引脚 1: 选择 P30 为 MTIOC0B-B 引脚	R/W
b5-b2	—	保留位	读写值都为“0”。	R/W
b7-b6	TCLKS[1:0]	MTCLK 引脚选择位	b7 b6 0 0: 将 P33 设定为 MTCLKA-A 输入引脚 将 P32 设定为 MTCLKB-A 输入引脚 将 P31 设定为 MTCLKC-A 输入引脚 将 P30 设定为 MTCLKD-A 输入引脚 0 1: 将 P21 设定为 MTCLKA-B 输入引脚 将 P20 设定为 MTCLKB-B 输入引脚 将 P11 设定为 MTCLKC-B 输入引脚 将 P10 设定为 MTCLKD-B 输入引脚 1 0: 将 PE4 设定为 MTCLKC-C 输入引脚 将 PE3 设定为 MTCLKD-C 输入引脚 (不能选择 MTCLKA 引脚和 MTCLKB 引脚) 1 1: 不能设定	R/W

PFCMTU 寄存器是选择 MTU3 引脚的寄存器。

MTUS<sub>i</sub> 位 (MTU3 引脚选择位) (i=0、1)

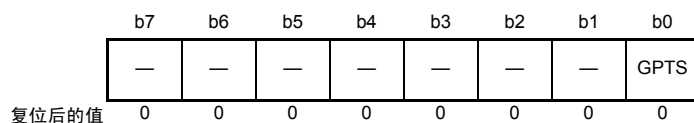
此位选择 MTU3 的输入 / 输出引脚。

TCLKS[1:0] 位 (MTCLK 引脚选择位)

此位选择 MTU3 的 MTCLK 输入引脚。

## 15.3.2.8 端口功能寄存器 D (PFDGPT)

地址 0008 C10Dh



位	符号	位名	功能	R/W
b0	GPTS	GPT 引脚选择位	0: 选择 P71 为 GTIOC0A-A 引脚 选择 P74 为 GTIOC0B-A 引脚 选择 P72 为 GTIOC1A-A 引脚 选择 P75 为 GTIOC1B-A 引脚 选择 P73 为 GTIOC2A-A 引脚 选择 P76 为 GTIOC2B-A 引脚 1: 选择 PD7 为 GTIOC0A-B 引脚 选择 PD6 为 GTIOC0B-B 引脚 选择 PD5 为 GTIOC1A-B 引脚 选择 PD4 为 GTIOC1B-B 引脚 选择 PD3 为 GTIOC2A-B 引脚 (不能选择 GTIOC2B-B 引脚)	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

PFDGPT 寄存器是选择 GPT 引脚的寄存器。

## GPTS 位 (GPT 引脚选择位)

此位选择 GPT 的输入 / 输出引脚。

## 15.3.2.9 端口功能寄存器 G (PFGSPI)

地址 0008 C110h

b7	b6	b5	b4	b3	b2	b1	b0
SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	—
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	RSPCKE	RSPCK 输出允许位	0: RSPCK 引脚无效 1: RSPCK 引脚有效	R/W
b2	MOSIE	MOSI 输出允许位	0: MOSI 引脚无效 1: MOSI 引脚有效	R/W
b3	MISOE	MISO 输出允许位	0: MISO 引脚无效 1: MISO 引脚有效	R/W
b4	SSL0E	SSL0 输出允许位	0: SSL0 引脚无效 1: SSL0 引脚有效	R/W
b5	SSL1E	SSL1 输出允许位	0: SSL1 引脚无效 1: SSL1 引脚有效	R/W
b6	SSL2E	SSL2 输出允许位	0: SSL2 引脚无效 1: SSL2 引脚有效	R/W
b7	SSL3E	SSL3 输出允许位	0: SSL3 引脚无效 1: SSL3 引脚有效	R/W

PFGSPI 寄存器是设定 RSPI 的相关输入 / 输出引脚的寄存器。

**RSPCKE 位 (RSPCK 输出允许位)**

此位选择允许或者禁止 RSPCK 引脚的输出。在使用 RSPCK 引脚时，必须将此位置“1”。

**MOSIE 位 (MOSI 输出允许位)**

此位选择允许或者禁止 MOSI 引脚的输出。在使用 MOSI 引脚时，必须将此位置“1”。

**MISOE 位 (MISO 输出允许位)**

此位选择允许或者禁止 MISO 引脚的输出。在使用 MISO 引脚时，必须将此位置“1”。

**SSL0E 位 (SSL0 输出允许位)**

此位选择允许或者禁止 SSL0 引脚的输出。在使用 SSL0 引脚时，必须将此位置“1”。

**SSL1E 位 (SSL1 输出允许位)**

此位选择允许或者禁止 SSL1 引脚的输出。在使用 SSL1 引脚时，必须将此位置“1”。

**SSL2E 位 (SSL2 输出允许位)**

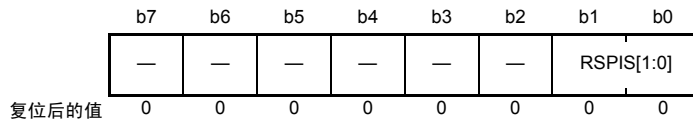
此位选择允许或者禁止 SSL2 引脚的输出。在使用 SSL2 引脚时，必须将此位置“1”。

**SSL3E 位 (SSL3 允许位)**

此位选择允许或者禁止 SSL3 引脚的输出。在使用 SSL3 引脚时，必须将此位置“1”。

## 15.3.2.10 端口功能寄存器 H (PFHSPI)

地址 0008 C111h



位	符号	位名	功能	R/W
b1-b0	RSPIS[1:0]	RSPI 引脚选择位	b1 b0 0 0: 将 P22 设定为 MISO-A 输入引脚 将 P23 设定为 MOSI-A 输入引脚 将 P24 设定为 RSPCK-A 输入引脚 将 P30 设定为 SSL0-A 输入引脚 将 P31 设定为 SSL1-A 输入引脚 将 P32 设定为 SSL2-A 输入引脚 将 P33 设定为 SSL3-A 输入引脚 0 1: 将 PA5 设定为 MISO-B 输入引脚 将 PB0 设定为 MOSI-B 输入引脚 将 PA4 设定为 RSPCK-B 输入引脚 将 PA3 设定为 SSL0-B 输入引脚 将 PA2 设定为 SSL1-B 输入引脚 (不能选择 SSL2 引脚和 SSL3 引脚) 1 0: 不能设定 1 1: 不能设定	R/W
b7	—	保留位	读写值都为“0”。	R/W

PFHSPI 寄存器是设定 RSPI 的相关输入 / 输出引脚的寄存器。

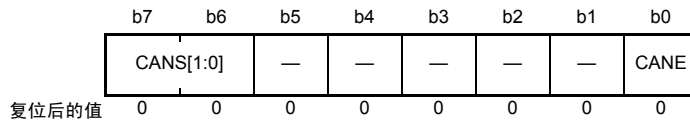
## RSPIS[1:0] 位 (RSPI 引脚选择位)

此位选择 RSPI 的输入 / 输出引脚。

RSPI 的各输入 / 输出引脚有允许位，选择允许位为“1”的引脚而不选择允许位为“0”的引脚。

## 15.3.2.11 端口功能寄存器 J (PFJCAN)

地址 0008 C113h



位	符号	位名	功能	R/W
b0	CANE	CAN 引脚允许位	0: CTX 引脚和 CRX 引脚无效 1: CTX 引脚和 CRX 引脚有效	R/W
b5-b1	—	保留位	读写值都为“0”。	R/W
b7-b6	CANS[1:0]	CAN 引脚选择位	b7 b6 00: 将 PB5 设定为 CTX-A 输入引脚 将 PB6 设定为 CRX-A 输入引脚 01: 将 P23 设定为 CTX-B 输入引脚 将 P22 设定为 CRX-B 输入引脚 10: 将 PE0 设定为 CRX-C 输入引脚 将 PD7 设定为 CTX-C 输入引脚 11: 不能设定	R/W

PFJCAN 寄存器是设定 CAN 的相关输入 / 输出引脚的寄存器。

## CANE 位 (CAN 引脚允许位)

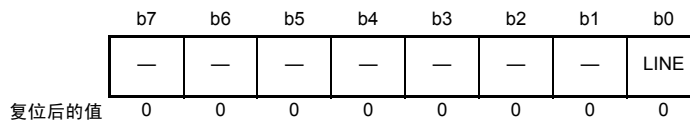
此位选择允许或者禁止 CAN 引脚。在使用 CAN 引脚时，必须将此位置“1”。

## CANS[1:0] 位 (CAN 引脚选择位)

选择 CAN 的输入 / 输出引脚。

## 15.3.2.12 端口功能寄存器 K (PFKLIN)

地址 0008 C114h



位	符号	位名	功能	R/W
b0	LINE	LIN 引脚允许位	0: LTX 引脚和 LRX 引脚无效 1: LTX 引脚和 LRX 引脚有效	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

PFKLIN 寄存器是设定 LIN 的相关输入 / 输出引脚的寄存器。

## LINE 位 (LIN 引脚允许位)

此位选择允许或者禁止 LIN 引脚。在使用 LIN 引脚时，必须将此位置“1”。

## 15.3.2.13 端口功能寄存器 M (PFMPOE)

地址 0008 C116h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	POE11E	POE10E	POE8E	POE4E	POE0E
复位后的值	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	POE0E	POE0# 输入允许位	0: 设定为 I/O 端口 1: 设定为 POEn# 输入引脚 (n=0、4、8、10、11)	R/W (注1)
b1	POE4E	POE4# 输入允许位		R/W (注1)
b2	POE8E	POE8# 输入允许位		R/W (注1)
b3	POE10E	POE10# 输入允许位		R/W (注1)
b4	POE11E	POE11# 输入允许位		R/W (注1)
b7-b5	—	保留位		读写值都为“0”。

注 1. 只有复位开始后的第 1 次写操作有效，不能进行第 2 次以后的写操作。

PFMPOE 寄存器是选择允许或者禁止 POEn# 输入引脚的寄存器。

为了防止系统的误动作，必须在复位开始后进行写操作。只有复位开始后的第 1 次写操作有效。

## POEnE 位 (POEn# 输入允许位) (n=0、4、8、10、11)

此位选择允许或者禁止对应的 POEn# 输入引脚。

在使用 POEn# 时，必须将对应的 POEnE 位置“1”。

## 15.3.2.14 端口功能寄存器 N (PFNPOE)

地址 0008 C117h

	b7	b6	b5	b4	b3	b2	b1	b0
	POE10S	—	—	—	—	—	—	—
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	POE10S	POE10# 输入选择位	0: 将 PE2 设定为 POE10#-A 输入引脚 1: 将 PE4 设定为 POE10#-B 输入引脚	R/W (注1)

注 1. 只有复位开始后的第 1 次写操作有效，不能进行第 2 次以后的写操作。

PFNPOE 寄存器是选择允许或者禁止 POEn# 输入引脚的寄存器。

为了防止系统的误动作，必须在复位开始后进行写操作。只有复位开始后的第 1 次写操作有效。

## POE10S 位 (POE10# 输入选择位)

此位选择 POE10# 的输入引脚。



### 15.3.3 端口的设定


如果将各外围模块的引脚设定为有效，就能转换各端口的设定。

能通过设定各外围模块，独立设定用作输入的引脚。除端口寄存器的读、NMI 引脚的输入和 POEn# 引脚的输入以外，需要通过将输入缓冲控制寄存器（PORTn.ICR）的对应位置“1”，将输入缓冲器设定为有效。

对于用作输出和输入 / 输出的引脚，需要将各外围模块引脚的输出信号设定为有效。如果被相同端口多路复用的各外围模块输出信号有效设定发生竞争，就根据外围模块的端口多路复用优先级，优先的外围模块功能变为有效。

外围模块的端口多路复用优先级一览表如表 15.19 所示。

表 15.19 外围模块的端口多路复用优先级一览表（80 引脚 LQFP）

优先级	模块名	输出引脚名	
高  低	1	RSPI	RSPCK、MOSI、MISO、SSL0 ~ SSL3
	2	CAN	CTX
	3	LIN	LTX
	4	MTU0 ~ MTU7	MTIOC0A、MTIOC0B、MTIOC0C、MTIOC0D、MTIOC1A、MTIOC1B、MTIOC2A、MTIOC2B、MTIOC3A、MTIOC3B、MTIOC3C、MTIOC3D、MTIOC4A、MTIOC4B、MTIOC4C、MTIOC4D、MTIOC6B、MTIOC6D、MTIOC7A、MTIOC7B、MTIOC7C
	5	GPT0 ~ GPT3	GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B、GTIOC2A、GTIOC2B
	6	SCI0 ~ SCI2	SCK0 ~ SCK2、TXD0 ~ TXD2
	7	RIIC	SCL、SDA
	8	IOPORT	P10 ~ P11、P20 ~ P24、P30 ~ P33、P70 ~ P76、P91 ~ P96、PA2 ~ PA5、PB0 ~ PB7、PD3 ~ PD7、PE0、PE3 ~ PE4

### 15.3.4 输出允许设定一览表

各端口的输出允许设定一览表如表 15.20 所示。

对应的输出信号的详细内容请参照各外围模块的寄存器说明。

能通过端口功能寄存器，更改各外围模块引脚名末尾带有 A ~ C 的任意一个字符的引脚功能。

表 15.20 各端口的输出允许设定一览表 (1/3) (80 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
P10	PORT1	P10		PORT1.DDR.B0=1
P11	PORT1	P11		PORT1.DDR.B1=1
P20	PORT2	P20		PORT2.DDR.B0=1
P21	PORT2	P21		PORT2.DDR.B1=1
P22	RSPI	MISO-A	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	PORT2	P22		PORT2.DDR.B2=1
P23	RSPI	MOSI-A	PFGSPI.MOSIE=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	CAN	CTX-B	PFJCAN.CANE=1 PFJCAN.CANS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	LIN	LTX	PFKLIN.LINE=1	(信号的输出状态取决于外围模块的设定)
	PORT2	P23		PORT2.DDR.B3=1
P24	RSPI	RSPCK-A	PFGSPI.RSPCKE=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	PORT2	P24		PORT2.DDR.B4=1
P30	RSPI	SSL0-A	PFGSPI.SSL0E=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	MTU0	MTIOC0B-B	PFCMTU.MTUS1=1	(信号的输出状态取决于外围模块的设定)
	PORT3	P30		PORT3.DDR.B0=1
P31	RSPI	SSL1-A	PFGSPI.SSL1E=1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU0	MTIOC0A-B	PFCMTU.MTUS0=1	(信号的输出状态取决于外围模块的设定)
	PORT3	P31		PORT3.DDR.B1=1
P32	RSPI	SSL2-A	PFGSPI.SSL2E=1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU3	MTIOC3C		(信号的输出状态取决于外围模块的设定)
	PORT3	P32		PORT3.DDR.B2=1
P33	RSPI	SSL3-A	PFGSPI.SSL3E=1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU3	MTIOC3A		(信号的输出状态取决于外围模块的设定)
	PORT3	P33		PORT3.DDR.B3=1
P70	PORT7	P70		PORT7.DDR.B0=1
P71	MTU3	MTIOC3B		(信号的输出状态取决于外围模块的设定)
	GPT0	GTIOC0A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P71		PORT7.DDR.B1=1
P72	MTU4	MTIOC4A		(信号的输出状态取决于外围模块的设定)
	GPT1	GTIOC1A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P72		PORT7.DDR.B2=1
P73	MTU4	MTIOC4B		(信号的输出状态取决于外围模块的设定)
	GPT2	GTIOC2A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P73		PORT7.DDR.B3=1

表 15.20 各端口的输出允许设定一览表 (2/3) (80 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
P74	MTU3	MTIOC3D		(信号的输出状态取决于外围模块的设定)
	GPT0	GTIOC0B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P74		PORT7.DDR.B4=1
P75	MTU4	MTIOC4C		(信号的输出状态取决于外围模块的设定)
	GPT1	GTIOC1B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P75		PORT7.DDR.B5=1
P76	MTU4	MTIOC4D		(信号的输出状态取决于外围模块的设定)
	GPT2	GTIOC2B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P76		PORT7.DDR.B6=1
P91	MTU7	MTIOC7C		(信号的输出状态取决于外围模块的设定)
	PORT9	P91		PORT9.DDR.B1=1
P92	MTU6	MTIOC6D		(信号的输出状态取决于外围模块的设定)
	PORT9	P92		PORT9.DDR.B2=1
P93	MTU7	MTIOC7B		(信号的输出状态取决于外围模块的设定)
	PORT9	P93		PORT9.DDR.B3=1
P94	MTU7	MTIOC7A		(信号的输出状态取决于外围模块的设定)
	PORT9	P94		PORT9.DDR.B4=1
P95	MTU6	MTIOC6B		(信号的输出状态取决于外围模块的设定)
	PORT9	P95		PORT9.DDR.B5=1
P96	PORT9	P96		PORT9.DDR.B6=1
PA2	RSPI	SSL1-B	PFGSPI.SSL1E=1 PFHSPI.RSPIS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	MTU2	MTIOC2B		(信号的输出状态取决于外围模块的设定)
	PORTA	PA2		PORTA.DDR.B2=1
PA3	RSPI	SSL0-B	PFGSPI.SSL0E=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU2	MTIOC2A		(信号的输出状态取决于外围模块的设定)
	PORTA	PA3		PORTA.DDR.B3=1
PA4	RSPI	RSPCK-B	PFGSPI.RSPCKE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU1	MTIOC1B		(信号的输出状态取决于外围模块的设定)
	PORTA	PA4		PORTA.DDR.B4=1
PA5	RSPI	MISO-B	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU1	MTIOC1A		(信号的输出状态取决于外围模块的设定)
	PORTA	PA5		PORTA.DDR.B5=1
PB0	RSPI	MOSI-B	PFGSPI.MOSIE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU0	MTIOC0D		(信号的输出状态取决于外围模块的设定)
	PORTB	PB0		PORTB.DDR.B0=1
PB1	MTU0	MTIOC0C		(信号的输出状态取决于外围模块的设定)
	RIIC	SCL		RIIC.ICCR1.ICE=1
	PORTB	PB1		PORTB.DDR.B1=1
PB2	MTU0	MTIOC0B-A	PFCMTU.MTUS1=0	(信号的输出状态取决于外围模块的设定)
	SCI0	TXD0		SCI0.SCR.TE=1
	RIIC	SDA		RIIC.ICCR1.ICE=1
	PORTB	PB2		PORTB.DDR.B2=1

表 15.20 各端口的输出允许设定一览表 (3/3) (80 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
PB3	MTU0	MTIOC0A-A	PFCMTU.MTUS0=0	(信号的输出状态取决于外围模块的设定)
	SCI0	SCK0		当 SCI0.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1; 当 SCI0.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTB	PB3		PORTB.DDR.B3=1
PB4	PORTB	PB4		PORTB.DDR.B4=1
PB5	CAN	CTX-A	PFJCAN.CANE=1 PFJCAN.CANS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	SCI2	TXD2-A	PFFSCI.SCI2S =0	SCI2.SCR.TE=1
	PORTB	PB5		PORTB.DDR.B5=1
PB6	PORTB	PB6		PORTB.DDR.B6=1
PB7	SCI2	SCK2-A	PFFSCI.SCI2S=0	当 SCI2.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1; 当 SCI2.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTB	PB7		PORTB.DDR.B7=1
PD3	GPT2	GTIOC2A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	SCI1	TXD1		SCI1.SCR.TE=1
	PORTD	PD3		PORTD.DDR.B3=1
PD4	GPT1	GTIOC1B-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	SCI1	SCK1		当 SCI1.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1; 当 SCI1.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTD	PD4		PORTD.DDR.B4=1
PD5	GPT1	GTIOC1A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD5		PORTD.DDR.B5=1
PD6	GPT0	GTIOC0B-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD6		PORTD.DDR.B6=1
PD7	CAN	CTX-C	PFJCAN.CANE=1 PFJCAN.CANS[1:0]=10	(信号的输出状态取决于外围模块的设定)
	GPT0	GTIOC0A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD7		PORTD.DDR.B7=1
PE0	PORTE	PE0		PORTE.DDR.B0=1
PE3	PORTE	PE3		PORTE.DDR.B3=1
PE4	PORTE	PE4		PORTE.DDR.B4=1

### 15.3.5 未使用引脚的处理

未使用引脚的处理内容如表 15.21 所示。

表 15.21 未使用引脚的处理内容（80 引脚 LQFP）

引脚名	处理内容
EMLE	通过电阻连接 VSS（下拉）。
MD1、MD0	（必须用作模式引脚。）
MDE	（必须用作模式引脚。）
RES#	通过电阻连接 VCC（上拉）。
PE2/NMI/POE10#-A	通过电阻连接 VCC（上拉）。
EXTAL	（必须用作时钟引脚。）
XTAL	将引脚置为开路。
端口 1 ~ 3、7、9、 A、B、D、E	<ul style="list-style-type: none"> <li>每个引脚通过电阻连接 VCC（上拉）或者通过电阻连接 VSS（下拉）。</li> <li>也能在 PORTn.ICR 为初始值（输入缓冲器无效）的状态下将引脚置为开路（注 1）。</li> </ul>
端口 4	<ul style="list-style-type: none"> <li>每个引脚通过电阻连接 AVCC0（上拉）或者通过电阻连接 AVSS0（下拉）。</li> <li>也能在 PORTn.ICR 为初始值（输入缓冲器无效）的状态下将引脚置为开路（注 1）。</li> </ul>
端口 6	<ul style="list-style-type: none"> <li>每个引脚通过电阻连接 AVCC（上拉）或者通过电阻连接 AVSS（下拉）。</li> <li>也能在 PORTn.ICR 为初始值（输入缓冲器无效）的状态下将引脚置为开路（注 1）。</li> </ul>
VREFH0	连接 AVCC0。
VREFL0	连接 AVSS0。

注 1. 不能改变 PORTn.ICR 寄存器的初始值，否则就可能有穿透电流流过。

## 15.4 64 引脚 LQFP 的 I/O 端口

### 15.4.1 概要

RX62T 群的 64 引脚 LQFP 的 I/O 端口由端口 1 ~ 4、7、9、A、B、D、E 共 10 个端口构成，有 37 个输入 / 输出引脚。

I/O 端口的规格和端口功能一览表分别如表 15.22 和表 15.23 所示。

表 15.22 I/O 端口的规格（64 引脚 LQFP）

项目	内容
输入 / 输出引脚	37 个
输入引脚	9 个
端口	10 个端口（1 ~ 4、7、9、A、B、D、E）
漏极开路输出	2 个（RIIC 引脚）
大电流输出	6 个（MTU3 引脚、GPT 引脚）
施密特触发输入引脚	全部端口输入、CAN 输入、IRQ 输入、MTU3 输入、POE3 输入、RIIC 输入、SCI 输入、A/D 触发输入、NMI 输入、GPT 输入、LIN 输入
其他	<ul style="list-style-type: none"> <li>• 能驱动 1 个 TTL 负载和 30pF 的电容负载。</li> <li>• 在输出时驱动达林顿晶体管。</li> <li>• 能随时读引脚的状态</li> </ul>

表 15.23 端口功能一览表 (1/2) (64 引脚 LQFP)

端口	概要	位	功能			CMOS 输入引脚	施密特触发 输入引脚	漏极开路 输出功能	大电流 输出引脚	
			输入 / 输出	输入	输出					
端口 1	与 MTU3 输入 和中断输入兼 用的通用输入 / 输出端口	0	P10	MTCLKD-B/IRQ0-A		—	全部输入功能	—	—	
		1	P11	MTCLKC-B/IRQ1-A			全部输入功能			
端口 2	与 RSPI 输入 / 输出、LIN 输 入 / 输出和 CAN 输入输出 兼用的通用输 入 / 输出端口	2	P22/MISO-A	LRX/CRX-B		MISO-A	P22/LRX/CRX-B	—	—	
		3	P23/MOSI-A		LTX/CTX-B	MOSI-A	P23			
		4	P24/RSPCK-A			RSPCK-A	P24			
端口 3	与 MTU3 输入 / 输出和 RSPI 输入 / 输出兼 用的通用输入 / 输出端口	0	P30/MTIOC0B-B/ SSL0-A	MTCLKD-A		SSL0-A	P30/MTIOC0B-B/ MTCLKD-A	—	—	
		1	P31/MTIOC0A-B	MTCLKC-A	SSL1-A	—	全部输入功能			
		2	P32/MTIOC3C	MTCLKB-A	SSL2-A		全部输入功能			
		3	P33/MTIOC3A	MTCLKA-A	SSL3-A		全部输入功能			
端口 4	与 A/D 转换器 输入兼用的通 用输入端口	0	P40	P40/AN000		—	P40	—	—	
		1	P41	P41/AN001			P41			
		2	P42	P42/AN002			P42			
		3	P43	P43/AN003/CVREFL			P43			
		4	P44	P44/AN100			P44			
		5	P45	P45/AN101			P45			
		6	P46	P46/AN102			P46			
		7	P47	P47/AN103/ CVREFH			P47			
端口 7	与 POE3 输 入、中断输入、 MTU3 输入 / 输出和 GPT 输 入 / 输出兼用 的通用输入 / 输出端 口	0	P70	POE0#/IRQ5		—	全部输入功能	—	—	
		1	P71/MTIOC3B/ GTIOC0A-A				全部输入功能			○
		2	P72/MTIOC4A/ GTIOC1A-A				全部输入功能			○
		3	P73/MTIOC4B/ GTIOC2A-A				全部输入功能			○
		4	P74/MTIOC3D/ GTIOC0B-A				全部输入功能			○
		5	P75/MTIOC4C/ GTIOC1B-A				全部输入功能			○
		6	P76/MTIOC4D/ GTIOC2B-A				全部输入功能			○
端口 9	与 MTU3 输入 / 输出兼用的通 用输入 / 输出 端 口	1	P91/MTIOC7C			—	全部输入功能	—	—	
		2	P92/MTIOC6D				全部输入功能			
		3	P93/MTIOC7B				全部输入功能			
		4	P94/MTIOC7A				全部输入功能			
端口 A	与 MTU3 输入 / 输出、RSPI 输 入 / 输出和 A/D 转换器输入兼 用的通用输入 / 输出端口	2	PA2/MTIOC2B		SSL1-B	—	全部输入功能	—	—	
		3	PA3/MTIOC2A/ SSL0-B				SSL0-B			PA3/MTIOC2A
		4	PA4/MTIOC1B/ RSPCK-B	ADTRG0#-A			RSPCK-B			PA4/MTIOC1B/ ADTRG0#-A
		5	PA5/MTIOC1A/ MISO-B	ADTRG1#-A			MISO-B			PA5/MTIOC1A/ ADTRG1#-A

表 15.23 端口功能一览表 (2/2) (64 引脚 LQFP)

端口	概要	位	功能			CMOS 输入引脚	施密特触发 输入引脚	漏极开路 输出功能	大电流 输出引脚	
			输入 / 输出	输入	输出					
端口 B	与 MTU3 输入 / 输出、RSPI 输入 / 输出、RIIC 输入 / 输出、SCI 输入 / 输出、GPT 输入、POE3 输入、中断输入和 CAN 输入 / 输出兼用的通用输入 / 输出端口	0	PB0/MTIOC0D/ MOSI-B			MOSI-B	PB0/MTIOC0D		—	
		1	PB1/MTIOC0C/ SCL	RXD0		—	全部输入功能	○ (仅 SCL)	—	
		2	PB2/MTIOC0B-A/ SDA		TXD0			全部输入功能	○ (仅 SDA)	—
		3	PB3/MTIOC0A-A/ SCK0					全部输入功能	—	—
		4	PB4	GTETRQ/ POE8#/IRQ3				全部输入功能		
		5	PB5			TXD2-A/ CTX-A		全部输入功能		
		6	PB6	RXD2-A/CRX-A				全部输入功能		
端口 D	与 GPT 输入 / 输出、SCI 输入 / 输出和内部调试器输入 / 输出兼用的通用输入 / 输出端口	3	PD3/GTIOC2A-B		TXD1/TDO	—	全部输入功能	—	—	
		4	PD4/GTIOC1B-B/ SCK1	TCK		—	全部输入功能			
		5	PD5/GTIOC1A-B	RXD1/TDI		—	全部输入功能			
		6	PD6/GTIOC0B-B	TMS		—	全部输入功能			
		7	PD7/GTIOC0A-B	TRST#		—	全部输入功能			
端口 E	与 POE3 输入和中断输入兼用的通用输入 / 输出端口	2		PE2/POE10#-A/NMI		—	全部输入功能	—	—	



### 15.4.2 寄存器说明

I/O 端口的寄存器一览表和各寄存器的有效位一览表分别如表 15.24 和表 15.25 所示。

表 15.24 I/O 端口的寄存器一览表 (64 引脚 LQFP)

端口符号	寄存器名	寄存器符号	复位后的值	地址	存取长度
PORT1	数据方向寄存器	DDR	00h	0008 C001h	8
	数据寄存器	DR	00h	0008 C021h	8
	端口寄存器	PORT	不定值	0008 C041h	8
	输入缓冲控制寄存器	ICR	00h	0008 C061h	8
PORT2	数据方向寄存器	DDR	00h	0008 C002h	8
	数据寄存器	DR	00h	0008 C022h	8
	端口寄存器	PORT	不定值	0008 C042h	8
	输入缓冲控制寄存器	ICR	00h	0008 C062h	8
PORT3	数据方向寄存器	DDR	00h	0008 C003h	8
	数据寄存器	DR	00h	0008 C023h	8
	端口寄存器	PORT	不定值	0008 C043h	8
	输入缓冲控制寄存器	ICR	00h	0008 C063h	8
PORT4	端口寄存器	PORT	不定值	0008 C044h	8
	输入缓冲控制寄存器	ICR	00h	0008 C064h	8
PORT7	数据方向寄存器	DDR	00h	0008 C007h	8
	数据寄存器	DR	00h	0008 C027h	8
	端口寄存器	PORT	不定值	0008 C047h	8
	输入缓冲控制寄存器	ICR	00h	0008 C067h	8
PORT9	数据方向寄存器	DDR	00h	0008 C009h	8
	数据寄存器	DR	00h	0008 C029h	8
	端口寄存器	PORT	不定值	0008 C049h	8
	输入缓冲控制寄存器	ICR	00h	0008 C069h	8
PORTA	数据方向寄存器	DDR	00h	0008 C00Ah	8
	数据寄存器	DR	00h	0008 C02Ah	8
	端口寄存器	PORT	不定值	0008 C04Ah	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Ah	8
PORTB	数据方向寄存器	DDR	00h	0008 C00Bh	8
	数据寄存器	DR	00h	0008 C02Bh	8
	端口寄存器	PORT	不定值	0008 C04Bh	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Bh	8
PORTD	数据方向寄存器	DDR	00h	0008 C00Dh	8
	数据寄存器	DR	00h	0008 C02Dh	8
	端口寄存器	PORT	不定值	0008 C04Dh	8
	输入缓冲控制寄存器	ICR	00h	0008 C06Dh	8
PORTE	端口寄存器	PORT	不定值	0008 C04Eh	8
IOPORT	端口功能寄存器 C	PFCMTU	00h	0008 C10Ch	8
	端口功能寄存器 D	PDFGPT	00h	0008 C10Dh	8
	端口功能寄存器 G	PFGSPI	00h	0008 C110h	8
	端口功能寄存器 H	PFHSPI	00h	0008 C111h	8
	端口功能寄存器 J	PFJCAN	00h	0008 C113h	8
	端口功能寄存器 K	PFKLIN	00h	0008 C114h	8
	端口功能寄存器 M	PFMPOE	00h	0008 C116h	8

表 15.25 各寄存器的有效位一览表 (64 引脚 LQFP)

寄存器符号	b7	b6	b5	b4	b3	b2	b1	b0
PORT1.DDR	x	x	x	x	x	x	○	○
PORT2.DDR	x	x	x	○	○	○	x	x
PORT3.DDR	x	x	x	x	○	○	○	○
PORT7.DDR	x	○	○	○	○	○	○	○
PORT9.DDR	x	x	x	○	○	○	○	x
PORTA.DDR	x	x	○	○	○	○	x	x
PORTB.DDR	○	○	○	○	○	○	○	○
PORTD.DDR	○	○	○	○	○	x	x	x
PORT1.DR	x	x	x	x	x	x	○	○
PORT2.DR	x	x	x	○	○	○	x	x
PORT3.DR	x	x	x	x	○	○	○	○
PORT7.DR	x	○	○	○	○	○	○	○
PORT9.DR	x	x	x	○	○	○	○	x
PORTA.DR	x	x	○	○	○	○	x	x
PORTB.DR	○	○	○	○	○	○	○	○
PORTD.DR	○	○	○	○	○	x	x	x
PORT1.PORT	x	x	x	x	x	x	○	○
PORT2.PORT	x	x	x	○	○	○	x	x
PORT3.PORT	x	x	x	x	○	○	○	○
PORT4.PORT	○	○	○	○	○	○	○	○
PORT7.PORT	x	○	○	○	○	○	○	○
PORT9.PORT	x	x	x	○	○	○	○	x
PORTA.PORT	x	x	○	○	○	○	x	x
PORTB.PORT	○	○	○	○	○	○	○	○
PORTD.PORT	○	○	○	○	○	x	x	x
PORTE.PORT	x	x	x	x	x	○	x	x
PORT1.ICR	x	x	x	x	x	x	○	○
PORT2.ICR	x	x	x	○	○	○	x	x
PORTP3.ICR	x	x	x	x	○	○	○	○
PORT4.ICR	○	○	○	○	○	○	○	○
PORT7.ICR	x	○	○	○	○	○	○	○
PORT9.ICR	x	x	x	○	○	○	○	x
PORTA.ICR	x	x	○	○	○	○	x	x
PORTB.ICR	○	○	○	○	○	○	○	○
PORTD.ICR	○	○	○	○	○	x	x	x
IOPORT.PFCMTU	○	○	x	x	x	x	○	○
IOPORT.PFDGPT	x	x	x	x	x	x	x	○
IOPORT.PFGSPI	○	○	○	○	○	○	○	x
IOPORT.PFHSPi	x	x	x	x	x	x	○	○
IOPORT.PFJCAN	○	○	x	x	x	x	x	○
IOPORT.PFKLIN	x	x	x	x	x	x	x	○
IOPORT.PFMPOE	x	x	x	x	○	○	x	○

○: 有效位、x: 无效位 (保留位)

## 15.4.2.1 数据方向寄存器 (DDR)

地址 PORT1.DDR 0008 C001h、PORT2.DDR 0008 C002h、PORT3.DDR 0008 C003h、PORT7.DDR 0008 C007h、PORT9.DDR 0008 C009h、PORTA.DDR 0008 C00Ah、PORTB.DDR 0008 C00Bh、PORTD.DDR 0008 C00Dh、PORTE.DDR 0008 C00Eh

	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

- 注 1. PORT1.DDR 的低 2 位有效，高 6 位是保留位。  
 PORT2.DDR 的 b4~b2 有效，b7~b5、b1、b0 是保留位。  
 PORT3.DDR 的低 4 位有效，高 4 位是保留位。  
 PORT7.DDR 的低 7 位有效，高 1 位是保留位。  
 PORT9.DDR 的 b4~b1 有效，b7~b5、b0 是保留位。  
 PORTA.DDR 的 b5~b2 有效，b7、b6、b1、b0 是保留位。  
 PORTD.DDR 的高 5 位有效，低 3 位是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0	Pn0 输入 / 输出指定位	0: 输入端口 1: 输出端口	R/W
b1	B1	Pn1 输入 / 输出指定位		R/W
b2	B2	Pn2 输入 / 输出指定位		R/W
b3	B3	Pn3 输入 / 输出指定位		R/W
b4	B4	Pn4 输入 / 输出指定位		R/W
b5	B5	Pn5 输入 / 输出指定位		R/W
b6	B6	Pn6 输入 / 输出指定位		R/W
b7	B7	Pn7 输入 / 输出指定位		R/W

n=1~3、7、9、A、B、D、E

PORTn.DDR 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。

PORTn.DDR 寄存器 (n=1~3、7、9、A、B、D、E) 的各位分别对应端口 n 的各引脚，并且能以 1 位为单位进行指定。

## 15.4.2.2 数据寄存器 (DR)

地址 PORT1.DR 0008 C021h、PORT2.DR 0008 C022h、PORT3.DR 0008 C023h、PORT7.DR 0008 C027h、PORT9.DR 0008 C029h、  
PORTA.DR 0008 C02Ah、PORTB.DR 0008 C02Bh、PORTD.DR 0008 C02Dh、PORTE.DR 0008 C02Eh

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

- 注 1. PORT1.DR 的低 2 位有效，高 6 位是保留位。  
PORT2.DR 的 b4~b2 有效，b7~b5、b1、b0 是保留位。  
PORT3.DR 的低 4 位有效，高 4 位是保留位。  
PORT7.DR 的低 7 位有效，高 1 位是保留位。  
PORT9.DR 的 b4~b1 有效，b7~b5、b0 是保留位。  
PORTA.DR 的 b5~b2 有效，b7、b6、b1、b0 是保留位。  
PORTD.DR 的高 5 位有效，低 3 位是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0	Pn0 输出数据保存位	保存输出数据。	R/W
b1	B1	Pn1 输出数据保存位		R/W
b2	B2	Pn2 输出数据保存位		R/W
b3	B3	Pn3 输出数据保存位		R/W
b4	B4	Pn4 输出数据保存位		R/W
b5	B5	Pn5 输出数据保存位		R/W
b6	B6	Pn6 输出数据保存位		R/W
b7	B7	Pn7 输出数据保存位		R/W

n=1~3、7、9、A、B、D、E

PORTn.DR 寄存器 (n=1~3、7、9、A、B、D、E) 保存用作通用输出端口的引脚输出数据。

## 15.4.2.3 端口寄存器 (PORT)

地址 PORT1.PORT 0008 C041h、PORT2.PORT 0008 C042h、PORT3.PORT 0008 C043h、PORT4.PORT 0008 C044h、  
PORT7.PORT 0008 C047h、PORT9.PORT 0008 C049h、PORTA.PORT 0008 C04Ah、PORTB.PORT 0008 C04Bh、  
PORTD.PORT 0008 C04Dh、PORTE.PORT 0008 C04Eh

b7	b6	b5	b4	b3	b2	b1	b0
B7	B6	B5	B4	B3	B2	B1	B0

复位后的值 x x x x x x x x

- 注 1. PORT1.PORT 的低 2 位有效，高 6 位是保留位。  
PORT2.PORT 的 b4~b2 有效，b7~b5、b1、b0 是保留位。  
PORT3.PORT 的低 4 位有效，高 4 位是保留位。  
PORT7.PORT 的低 7 位有效，高 1 位是保留位。  
PORT9.PORT 的 b4~b1 有效，b7~b5、b0 是保留位。  
PORTA.PORT 的 b5~b2 有效，b7、b6、b1、b0 是保留位。  
PORTD.PORT 的高 5 位有效，低 3 位是保留位。  
PORTE.PORT 的 b2 有效，b7~b3、b1、b0 是保留位。
- 注 2. 保留位。读取值为“1”，写操作无效。

位	符号	位名	功能	R/W
b0	B0	Pn0 位	反映端口的引脚状态。	R
b1	B1	Pn1 位		R
b2	B2	Pn2 位		R
b3	B3	Pn3 位		R
b4	B4	Pn4 位		R
b5	B5	Pn5 位		R
b6	B6	Pn6 位		R
b7	B7	Pn7 位		R

n=1~4、7、9、A、B、D、E

PORTn.PORT 寄存器是反映端口引脚状态的寄存器。

如果读 PORTn.PORT 寄存器 (n=1~4、7、9、A、B、D、E)，就能读到引脚的状态。

## 15.4.2.4 输入缓冲控制寄存器 (ICR)

地址 PORT1.ICR 0008 C061h、PORT2.ICR 0008 C062h、PORT3.ICR 0008 C063h、PORT4.ICR 0008 C064h、  
PORT7.ICR 0008 C067h、PORT9.ICR 0008 C069h、PORTA.ICR 0008 C06Ah、PORTB.ICR 0008 C06Bh、PORTD.ICR 0008 C06Dh

	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

- 注 1. PORT1.ICR 的低 2 位有效，高 6 位是保留位。  
 PORT2.ICR 的 b4 ~ b2 有效，b7 ~ b5、b1、b0 是保留位。  
 PORT3.ICR 的低 4 位有效，高 4 位是保留位。  
 PORT7.ICR 的低 7 位有效，高 1 位是保留位。  
 PORT9.ICR 的 b4 ~ b1 有效，b7 ~ b5、b0 是保留位。  
 PORTA.ICR 的 b5 ~ b2 有效，b7、b6、b1、b0 是保留位。  
 PORTD.ICR 的高 5 位有效，低 3 位是保留位。
- 注 2. 保留位的读写值都为“0”。

位	符号	位名	功能	R/W
b0	B0 (注 1)	Pn0 输入缓冲器控制位	0: 对应引脚的输入缓冲器无效 1: 对应引脚的输入缓冲器有效	R/W
b1	B1 (注 1)	Pn1 输入缓冲器控制位		R/W
b2	B2 (注 1)	Pn2 输入缓冲器控制位		R/W
b3	B3 (注 1)	Pn3 输入缓冲器控制位		R/W
b4	B4 (注 1)	Pn4 输入缓冲器控制位		R/W
b5	B5 (注 1)	Pn5 输入缓冲器控制位		R/W
b6	B6 (注 1)	Pn6 输入缓冲器控制位		R/W
b7	B7 (注 1)	Pn7 输入缓冲器控制位		R/W

n=1 ~ 4、7、9、A、B、D

注 1. 在用作输入引脚时，必须将对应位置“1”。必须将不用作输入的引脚和用作模拟输入引脚的位置“0”。

PORTn.ICR 寄存器是控制端口输入缓冲器的寄存器。

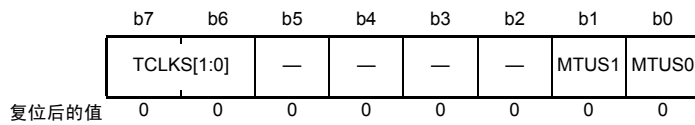
PORTn..ICR 寄存器 (n=1 ~ 4、7、9、A、B、D) 的各位分别对应端口 n 的各引脚，并且能以 1 位为单位进行指定。

在用作外围模块的输入引脚时，为了使对应引脚的输入缓冲器有效，需要预先将 PORTn.ICR 置“1”。如果在将 PORTn.ICR 置“0”的状态下用作外围模块的输入引脚，就将输入到外围模块的信号固定为 High 电平。

在更改 PORTn.ICR 寄存器的设定时，有可能因引脚的状态而在内部产生边沿，因此必须在不使用该输入引脚时更改 PORTn.ICR 寄存器的设定。例如，在 IRQ<sub>i</sub> (i=0 ~ 7) 输入时，必须在禁止该中断的状态下更改 PORTn.ICR 寄存器的设定。先将中断控制器的 IR<sub>i</sub>.IR 标志 (i=64 ~ 71 (IRQ 的中断向量号)) 置“0”，然后允许该中断。如果在更改 PORTn.ICR 寄存器的设定后产生边沿，就必须取消该边沿。

## 15.4.2.5 端口功能寄存器 C (PFCMTU)

地址 0008 C10Ch



位	符号	位名	功能	R/W
b0	MTUS0	MTU3 引脚选择 0 位	0: 选择 PB3 为 MTIOC0A-A 引脚 1: 选择 P31 为 MTIOC0A-B 引脚	R/W
b1	MTUS1	MTU3 引脚选择 1 位	0: 选择 PB2 为 MTIOC0B-A 引脚 1: 选择 P30 为 MTIOC0B-B 引脚	R/W
b5-b2	—	保留位	读写值都为“0”。	R/W
b7-b6	TCLKS[1:0]	MTCLK 引脚选择位	b7 b6 0 0: 将 P33 设定为 MTCLKA-A 输入引脚 将 P32 设定为 MTCLKB-A 输入引脚 将 P31 设定为 MTCLKC-A 输入引脚 将 P30 设定为 MTCLKD-A 输入引脚 0 1: 将 P11 设定为 MTCLKC-B 输入引脚 将 P10 设定为 MTCLKD-B 输入引脚 (不能选择 MTCLKA 引脚和 MTCLKB 引脚) 1 0: 不能设定 1 1: 不能设定	R/W

PFCMTU 寄存器是选择 MTU3 引脚的寄存器。

MTUS<sub>i</sub> 位 (MTU3 引脚选择位) (i=0、1)

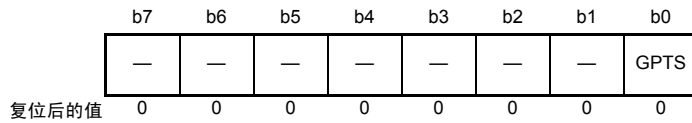
此位选择 MTU3 的输入 / 输出引脚。

TCLKS[1:0] 位 (MTCLK 引脚选择位)

此位选择 MTU3 的 MTCLK 输入引脚。

## 15.4.2.6 端口功能寄存器 D (PFDGPT)

地址 0008 C10Dh



位	符号	位名	功能	R/W
b0	GPTS	GPT 引脚选择位	0: 选择 P71 为 GTIOC0A-A 引脚 选择 P74 为 GTIOC0B-A 引脚 选择 P72 为 GTIOC1A-A 引脚 选择 P75 为 GTIOC1B-A 引脚 选择 P73 为 GTIOC2A-A 引脚 选择 P76 为 GTIOC2B-A 引脚 1: 选择 PD7 为 GTIOC0A-B 引脚 选择 PD6 为 GTIOC0B-B 引脚 选择 PD5 为 GTIOC1A-B 引脚 选择 PD4 为 GTIOC1B-B 引脚 选择 PD3 为 GTIOC2A-B 引脚 (不能选择 GTIOC2B-B 引脚)	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

PFDGPT 寄存器是选择 GPT 引脚的寄存器。

## GPTS 位 (GPT 引脚选择位)

此位选择 GPT 的输入 / 输出引脚。



## 15.4.2.7 端口功能寄存器 G (PFGSPI)

地址 0008 C110h

b7	b6	b5	b4	b3	b2	b1	b0
SSL3E	SSL2E	SSL1E	SSL0E	MISOE	MOSIE	RSPCKE	—
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	RSPCKE	RSPCK 输出允许位	0: RSPCK 引脚无效 1: RSPCK 引脚有效	R/W
b2	MOSIE	MOSI 输出允许位	0: MOSI 引脚无效 1: MOSI 引脚有效	R/W
b3	MISOE	MISO 输出允许位	0: MISO 引脚无效 1: MISO 引脚有效	R/W
b4	SSL0E	SSL0 输出允许位	0: SSL0 引脚无效 1: SSL0 引脚有效	R/W
b5	SSL1E	SSL1 输出允许位	0: SSL1 引脚无效 1: SSL1 引脚有效	R/W
b6	SSL2E	SSL2 输出允许位	0: SSL2 引脚无效 1: SSL2 引脚有效	R/W
b7	SSL3E	SSL3 输出允许位	0: SSL3 引脚无效 1: SSL3 引脚有效	R/W

PFGSPI 寄存器是设定 RSPI 的相关输入 / 输出引脚的寄存器。

**RSPCKE 位 (RSPCK 输出允许位)**

此位选择允许或者禁止 RSPCK 引脚的输出。在使用 RSPCK 引脚时，必须将此位置“1”。

**MOSIE 位 (MOSI 输出允许位)**

此位选择允许或者禁止 MOSI 引脚的输出。在使用 MOSI 引脚时，必须将此位置“1”。

**MISOE 位 (MISO 输出允许位)**

此位选择允许或者禁止 MISO 引脚的输出。在使用 MISO 引脚时，必须将此位置“1”。

**SSL0E 位 (SSL0 输出允许位)**

此位选择允许或者禁止 SSL0 引脚的输出。在使用 SSL0 引脚时，必须将此位置“1”。

**SSL1E 位 (SSL1 输出允许位)**

此位选择允许或者禁止 SSL1 引脚的输出。在使用 SSL1 引脚时，必须将此位置“1”。

**SSL2E 位 (SSL2 输出允许位)**

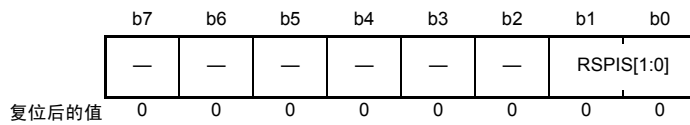
此位选择允许或者禁止 SSL2 引脚的输出。在使用 SSL2 引脚时，必须将此位置“1”。

**SSL3E 位 (SSL3 允许位)**

此位选择允许或者禁止 SSL3 引脚的输出。在使用 SSL3 引脚时，必须将此位置“1”。

## 15.4.2.8 端口功能寄存器 H (PFHSPI)

地址 0008 C111h



位	符号	位名	功能	R/W
b1-b0	RSPIS[1:0]	RSPI 引脚选择位	b1 b0 0 0: 将 P22 设定为 MISO-A 输入引脚 将 P23 设定为 MOSI-A 输入引脚 将 P24 设定为 RSPCK-A 输入引脚 将 P30 设定为 SSL0-A 输入引脚 将 P31 设定为 SSL1-A 输入引脚 将 P32 设定为 SSL2-A 输入引脚 将 P33 设定为 SSL3-A 输入引脚 0 1: 将 PA5 设定为 MISO-B 输入引脚 将 PB0 设定为 MOSI-B 输入引脚 将 PA4 设定为 RSPCK-B 输入引脚 将 PA3 设定为 SSL0-B 输入引脚 将 PA2 设定为 SSL1-B 输入引脚 (不能选择 SSL2 引脚和 SSL3 引脚) 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

PFHSPI 寄存器是设定 RSPI 的相关输入 / 输出引脚的寄存器。

## RSPIS[1:0] 位 (RSPI 引脚选择位)

此位选择 RSPI 的输入 / 输出引脚。

RSPI 的各输入 / 输出引脚有允许位，选择允许位为“1”的引脚而不选择允许位为“0”的引脚。

## 15.4.2.9 端口功能寄存器 J (PFJCAN)

地址 0008 C113h

b7	b6	b5	b4	b3	b2	b1	b0
CANS[1:0]	—	—	—	—	—	—	CANE
复位后的值	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	CANE	CAN 引脚允许位	0: CTX 引脚和 CRX 引脚无效 1: CTX 引脚和 CRX 引脚有效	R/W
b5-b1	—	保留位	读写值都为“0”。	R/W
b7-b6	CANS[1:0]	CAN 引脚选择位	b7 b6 0 0: 将 PB5 设定为 CTX-A 输入引脚 将 PB6 设定为 CRX-A 输入引脚 0 1: 将 P23 设定为 CTX-B 输入引脚 将 P22 设定为 CRX-B 输入引脚 1 0: 不能设定 1 1: 不能设定	R/W

PFJCAN 寄存器是设定 CAN 的相关输入 / 输出引脚的寄存器。

## CANE 位 (CAN 引脚允许位)

此位选择允许或者禁止 CAN 引脚。在使用 CAN 引脚时，必须将此位置“1”。

## CANS[1:0] 位 (CAN 引脚选择位)

选择 CAN 的输入 / 输出引脚。

## 15.4.2.10 端口功能寄存器 K (PFKLIN)

地址 0008 C114h

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

位	符号	位名	功能	R/W
b0	LINE	LIN 引脚允许位	0: LTX 引脚和 LRX 引脚无效 1: LTX 引脚和 LRX 引脚有效	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

PFKLIN 寄存器是设定 LIN 的相关输入 / 输出引脚的寄存器。

## LINE 位 (LIN 引脚允许位)

此位选择允许或者禁止 LIN 引脚。在使用 LIN 引脚时，必须将此位置“1”。

## 15.4.2.11 端口功能寄存器 M (PFMPOE)

地址 0008 C116h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	POE10E	POE8E	—	POE0E
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	POE0E	POE0# 输入允许位	0: 设定为 I/O 端口 1: 设定为 POE0# 输入引脚	R/W (注 1)
b1	—	保留位	读写值都为“0”。	R/W
b2	POE8E	POE8# 输入允许位	0: 设定为 I/O 端口 1: 设定为 POE8# 输入引脚	R/W (注 1)
b3	POE10E	POE10# 输入允许位	0: 设定为 I/O 端口 1: 设定为 POE10# 输入引脚	R/W (注 1)
b7-b4	—	保留位	读写值都为“0”。	R/W

注 1. 只有复位开始后的第 1 次写操作有效，不能进行第 2 次以后的写操作。

PFMPOE 寄存器是选择允许或者禁止 POEn# 输入引脚的寄存器。

为了防止系统的误动作，必须在复位开始后进行写操作。只有复位开始后的第 1 次写操作有效。

## POEnE 位 (POEn# 输入允许位) (n=0、8、10)

此位选择允许或者禁止对应的 POEn# 输入引脚。

在使用 POEn# 时，必须将对应的 POEnE 位置“1”。

### 15.4.3 端口的设定

如果将各外围模块的引脚设定为有效，就能转换各端口的设定。

能通过设定各外围模块，独立设定用作输入的引脚。除端口寄存器的读、NMI 的输入和 POEn# 引脚的输入以外，需要通过将输入缓冲控制寄存器（PORTn.ICR）的对应位置“1”，将输入缓冲器设定为有效。

对于用作输出和输入 / 输出的引脚，需要将各外围模块引脚的输出信号设定为有效。如果被相同端口多路复用的各外围模块输出信号的有效设定发生竞争，就根据外围模块的端口多路复用优先级，优先的外围模块功能变为有效。

外围模块的端口多路复用优先级一览表如表 15.26 所示。

表 15.26 外围模块的端口多路复用优先级一览表（64 引脚 LQFP）

优先级	模块名	输出引脚名	
高 ↑      ↓ 低	1	RSPI	RSPCK、MOSI、MISO、SSL0 ~ SSL3
	2	CAN	CTX
	3	LIN	LTX
	4	MTU0 ~ MTU7	MTIOC0A、MTIOC0B、MTIOC0C、MTIOC0D、MTIOC1A、MTIOC1B、MTIOC2A、MTIOC2B、MTIOC3A、MTIOC3B、MTIOC3C、MTIOC3D、MTIOC4A、MTIOC4B、MTIOC4C、MTIOC4D、MTIOC6D、MTIOC7A、MTIOC7B、MTIOC7C
	5	GPT0 ~ GPT3	GTIOC0A、GTIOC0B、GTIOC1A、GTIOC1B、GTIOC2A、GTIOC2B
	6	SCI0 ~ SCI2	SCK0 ~ SCK2、TXD0 ~ TXD2
	7	RIIC	SCL、SDA
	8	IOPORT	P10 ~ P11、P22 ~ P24、P30 ~ P33、P70 ~ P76、P91 ~ P94、PA2 ~ PA5、PB0 ~ PB7、PD3 ~ PD7

### 15.4.4 输出允许设定一览表

各端口的输出允许设定一览表如表 15.27 所示。

对应的输出信号的详细内容请参照各外围模块的寄存器说明。

能通过端口功能寄存器，更改各外围模块引脚名末尾带有 A ~ B 的任意一个字符的引脚功能。

表 15.27 各端口的输出允许设定一览表 (1/3) (64 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
P10	PORT1	P10		PORT1.DDR.B0=1
P11	PORT1	P11		PORT1.DDR.B1=1
P22	RSPI	MISO-A	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	PORT2	P22		PORT2.DDR.B2=1
P23	RSPI	MOSI-A	PFGSPI.MOSIE=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	CAN	CTX-B	PFJCAN.CANE=1 PFJCAN.CANS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	LIN	LTX	PFKLIN.LINE=1	(信号的输出状态取决于外围模块的设定)
	PORT2	P23		PORT2.DDR.B3=1
P24	RSPI	RSPCK-A	PFGSPI.RSPCKE=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	PORT2	P24		PORT2.DDR.B4=1
P30	RSPI	SSL0-A	PFGSPI.SSL0E=1 PFHSPI.RSPIS[1:0]=00	(除引脚允许以外，外围模块的设定有输入/输出转换功能。)
	MTU0	MTIOC0B-B	PFCMTU.MTUS1=1	(信号的输出状态取决于外围模块的设定)
	PORT3	P30		PORT3.DDR.B0=1
P31	RSPI	SSL1-A	PFGSPI.SSL1E=1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU0	MTIOC0A-B	PFCMTU.MTUS0=1	(信号的输出状态取决于外围模块的设定)
	PORT3	P31		PORT3.DDR.B1=1
P32	RSPI	SSL2-A	PFGSPI.SSL2E=1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU3	MTIOC3C		(信号的输出状态取决于外围模块的设定)
	PORT3	P32		PORT3.DDR.B2=1
P33	RSPI	SSL3-A	PFGSPI.SSL3E=1 PFHSPI.RSPIS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	MTU3	MTIOC3A		(信号的输出状态取决于外围模块的设定)
	PORT3	P33		PORT3.DDR.B3=1
P70	PORT7	P70		PORT7.DDR.B0=1
P71	MTU3	MTIOC3B		(信号的输出状态取决于外围模块的设定)
	GPT0	GTIOC0A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P71		PORT7.DDR.B1=1
P72	MTU4	MTIOC4A		(信号的输出状态取决于外围模块的设定)
	GPT1	GTIOC1A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P72		PORT7.DDR.B2=1
P73	MTU4	MTIOC4B		(信号的输出状态取决于外围模块的设定)
	GPT2	GTIOC2A-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P73		PORT7.DDR.B3=1
P74	MTU3	MTIOC3D		(信号的输出状态取决于外围模块的设定)
	GPT0	GTIOC0B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P74		PORT7.DDR.B4=1

表 15.27 各端口的输出允许设定一览表 (2/3) (64 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
P75	MTU4	MTIOC4C		(信号的输出状态取决于外围模块的设定)
	GPT1	GTIOC1B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P75		PORT7.DDR.B5=1
P76	MTU4	MTIOC4D		(信号的输出状态取决于外围模块的设定)
	GPT2	GTIOC2B-A	PFDGPT.GPTS=0	(信号的输出状态取决于外围模块的设定)
	PORT7	P76		PORT7.DDR.B6=1
P91	MTU7	MTIOC7C		(信号的输出状态取决于外围模块的设定)
	PORT9	P91		PORT9.DDR.B1=1
P92	MTU6	MTIOC6D		(信号的输出状态取决于外围模块的设定)
	PORT9	P92		PORT9.DDR.B2=1
P93	MTU7	MTIOC7B		(信号的输出状态取决于外围模块的设定)
	PORT9	P93		PORT9.DDR.B3=1
P94	MTU7	MTIOC7A		(信号的输出状态取决于外围模块的设定)
	PORT9	P94		PORT9.DDR.B4=1
PA2	RSPI	SSL1-B	PFGSPI.SSL1E=1 PFHSPI.RSPIS[1:0]=01	(信号的输出状态取决于外围模块的设定)
	MTU2	MTIOC2B		(信号的输出状态取决于外围模块的设定)
	PORTA	PA2		PORTA.DDR.B2=1
PA3	RSPI	SSL0-B	PFGSPI.SSL0E=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU2	MTIOC2A		(信号的输出状态取决于外围模块的设定)
	PORTA	PA3		PORTA.DDR.B3=1
PA4	RSPI	RSPCK-B	PFGSPI.RSPCKE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU1	MTIOC1B		(信号的输出状态取决于外围模块的设定)
	PORTA	PA4		PORTA.DDR.B4=1
PA5	RSPI	MISO-B	PFGSPI.MISOE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU1	MTIOC1A		(信号的输出状态取决于外围模块的设定)
	PORTA	PA5		PORTA.DDR.B5=1
PB0	RSPI	MOSI-B	PFGSPI.MOSIE=1 PFHSPI.RSPIS[1:0]=01	(除引脚允许以外, 外围模块的设定有输入/输出转换功能。)
	MTU0	MTIOC0D		(信号的输出状态取决于外围模块的设定)
	PORTB	PB0		PORTB.DDR.B0=1
PB1	MTU0	MTIOC0C		(信号的输出状态取决于外围模块的设定)
	RIIC	SCL		RIIC.ICCR1.ICE=1
	PORTB	PB1		PORTB.DDR.B1=1
PB2	MTU0	MTIOC0B-A	PFCMTU.MTUS1=0	(信号的输出状态取决于外围模块的设定)
	SCI0	TXD0		SCI0.SCR.TE=1
	RIIC	SDA		RIIC.ICCR1.ICE=1
	PORTB	PB2		PORTB.DDR.B2=1

表 15.27 各端口的输出允许设定一览表 (3/3) (64 引脚 LQFP)

端口	模块名	输出信号名	端口功能寄存器的设定	各外围模块的设定
PB3	MTU0	MTIOC0A-A	PFCMTU.MTUS0=0	(信号的输出状态取决于外围模块的设定)
	SCI0	SCK0		当 SCI0.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1 ; 当 SCI0.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTB	PB3		PORTB.DDR.B3=1
PB4	PORTB	PB4		PORTB.DDR.B4=1
PB5	CAN	CTX-A	PFJCAN.CANE=1 PFJCAN.CANS[1:0]=00	(信号的输出状态取决于外围模块的设定)
	SCI2	TXD2-A	PFFSCI.SCI2S=0	SCI2.SCR.TE=1
	PORTB	PB5		PORTB.DDR.B5=1
PB6	PORTB	PB6		PORTB.DDR.B6=1
PB7	SCI2	SCK2-A	PFFSCI.SCI2S=0	当 SCI2.SCMR.SMIF =1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1 ; 当 SCI2.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTB	PB7		PORTB.DDR.B7=1
PD3	GPT2	GTIOC2A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	SCI1	TXD1		SCI1.SCR.TE=1
	PORTD	PD3		PORTD.DDR.B3=1
PD4	GPT1	GTIOC1B-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	SCI1	SCK1		当 SCI1.SCMR.SMIF=1 时: 如果 SMR.GM=0、SCR.CKE[1:0]=01, 或者 SMR.GM=1, 则 SCR.TE=1 或者 SCR.RE=1 ; 当 SCI1.SCMR.SMIF=0 时: 如果 SMR.CM=0、SCR.CKE[1:0]=01, 或者 SMR.CM=1、SCR.CKE[1]=0, 则 SCR.TE=1 或者 SCR.RE=1。
	PORTD	PD4		PORTD.DDR.B4=1
PD5	GPT1	GTIOC1A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD5		PORTD.DDR.B5=1
PD6	GPT0	GTIOC0B-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD6		PORTD.DDR.B6=1
PD7	GPT0	GTIOC0A-B	PFDGPT.GPTS=1	(信号的输出状态取决于外围模块的设定)
	PORTD	PD7		PORTD.DDR.B7=1



### 15.4.5 未使用引脚的处理

未使用引脚的处理内容如表 15.28 所示。

表 15.28 未使用引脚的处理内容（64 引脚 LQFP）

引脚名	处理内容
EMLE	通过电阻连接 VSS（下拉）。
MD1、MD0	（必须用作模式引脚。）
MDE	（必须用作模式引脚。）
RES#	通过电阻连接 VCC（上拉）。
PE2/NMI/POE10#-A	通过电阻连接 VCC（上拉）。
EXTAL	（必须用作时钟引脚。）
XTAL	将引脚置为开路。
端口 1 ~ 3、7、9、 A、B、D、E	<ul style="list-style-type: none"> <li>每个引脚通过电阻连接 VCC（上拉）或者通过电阻连接 VSS（下拉）。</li> <li>也能在 PORTn.ICR 为初始值（输入缓冲器无效）的状态下将引脚置为开路（注 1）。</li> </ul>
端口 4	<ul style="list-style-type: none"> <li>每个引脚通过电阻连接 AVCC0（上拉）或者通过电阻连接 AVSS0（下拉）。</li> <li>也能在 PORTn.ICR 为初始值（输入缓冲器无效）的状态下将引脚置为开路（注 1）。</li> </ul>
VREFH0	连接 AVCC0。
VREFL0	连接 AVSS0。

注 1. 不能改变 PORTn.ICR 寄存器的初始值，否则就可能有穿透电流流过。

### 15.5 输入 / 输出端口的结构

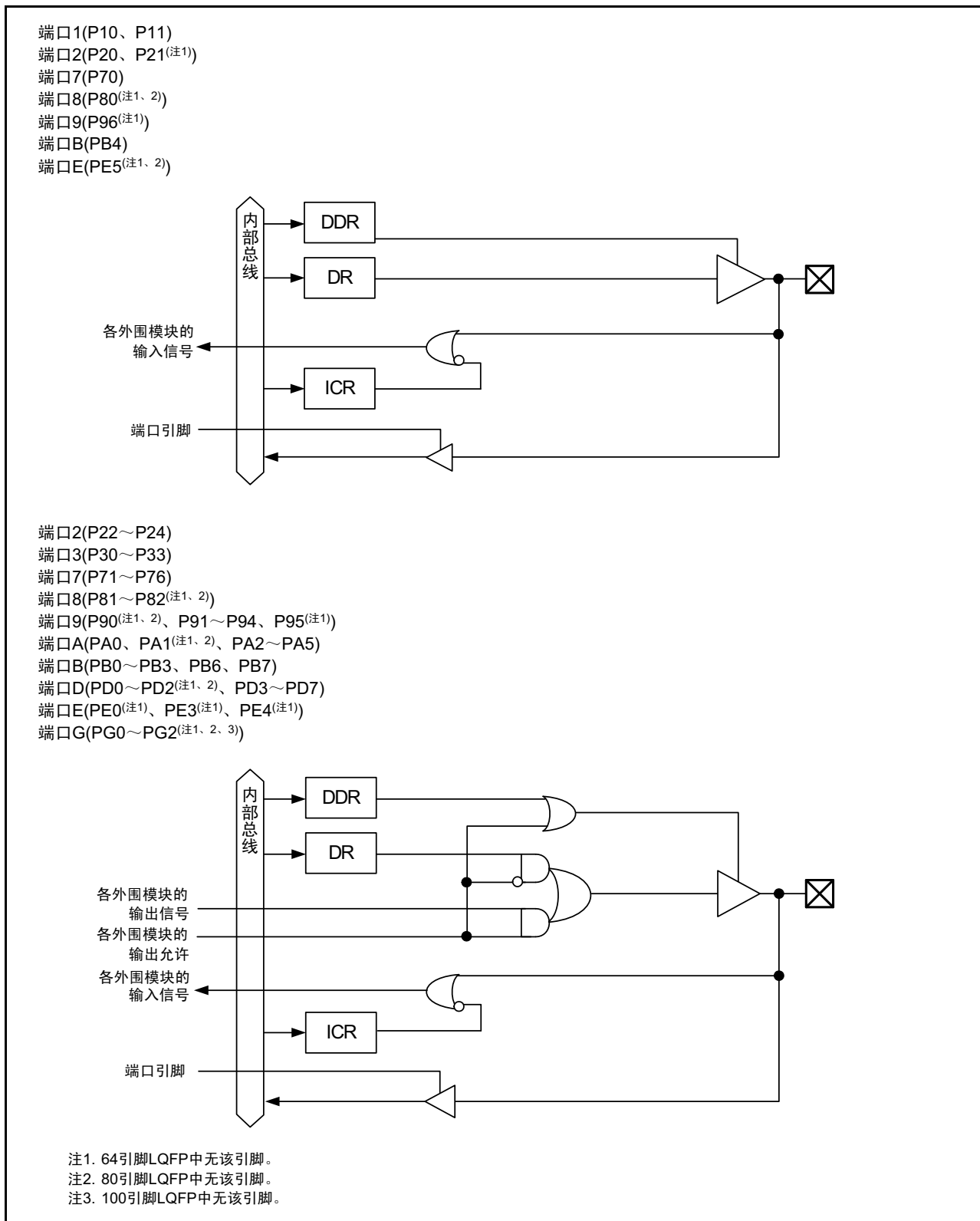


图 15.1 输入 / 输出端口的结构 (1)

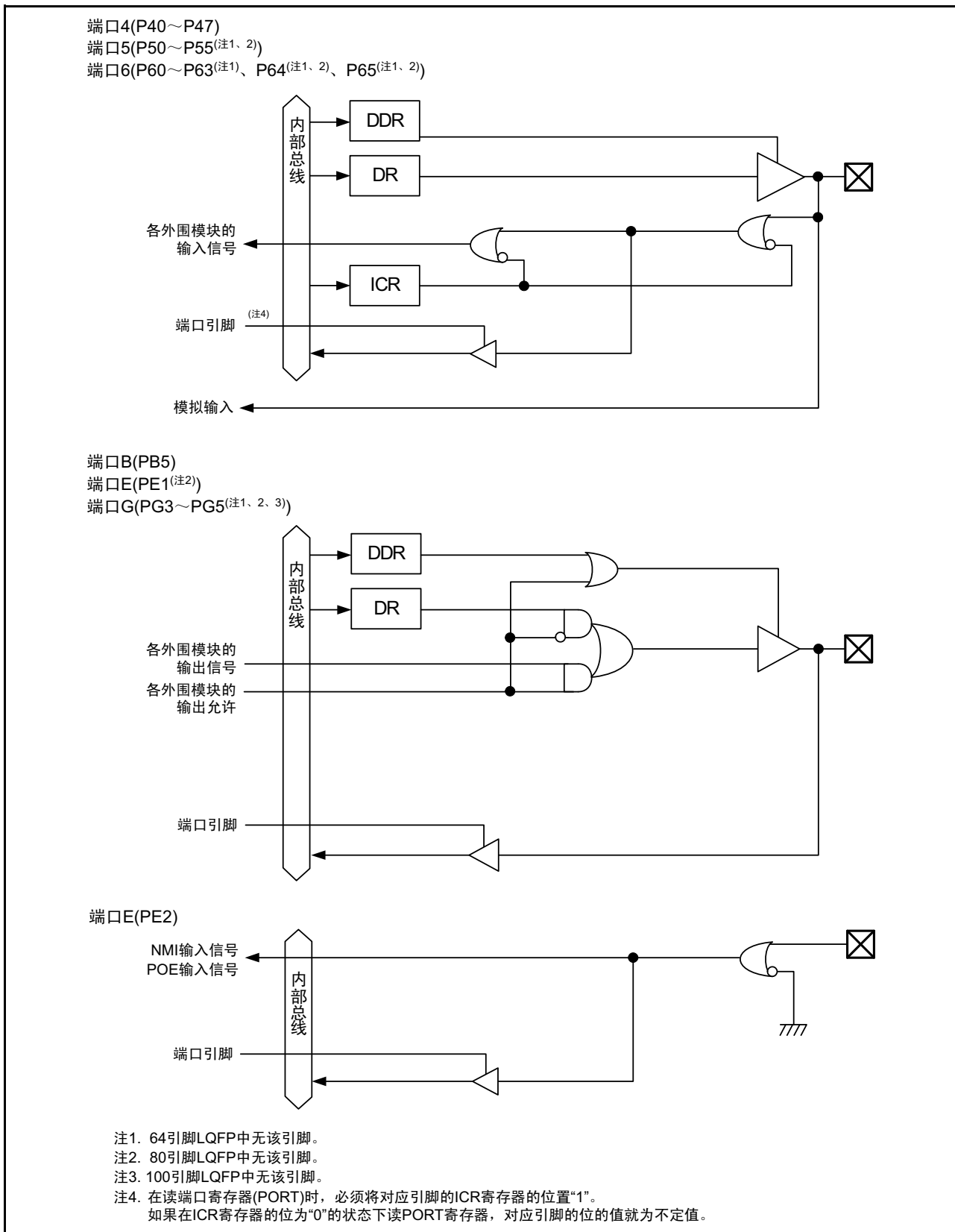


图 15.2 输入 / 输出端口的结构 (2)

## 15.6 使用时的注意事项

### 15.6.1 输入缓冲控制寄存器 (PORTn.ICR) 的设定

在更改 PORTn.ICR 寄存器的设定时, 有可能因引脚的状态而在内部产生边沿, 导致意想不到的运行。因此, 必须在将引脚固定为 High 电平的状态下或者在通过设定被分配到该引脚的外围模块将输入功能置为无效的状态下更改 PORTn.ICR 寄存器的设定。

如果通过设定 PORTn.ICR 寄存器将输入置为有效, 对于分配了多个输入功能的引脚, 就将引脚状态反映到全部的输入功能。对于未使用的输入功能, 必须注意各外围模块的设定。

在将引脚用作输出引脚时, 如果通过设定 PORTn.ICR 寄存器将输入置为有效, 输出数据就作为引脚状态被取入。必须通过设定 PORTn.ICR 寄存器将用作输出的引脚输入置为无效。

### 15.6.2 端口功能寄存器的设定

端口功能寄存器控制 I/O 端口。在设定各引脚的输入 / 输出时, 必须在选择输入 / 输出目标后将输入 / 输出置为有效。

在更改输入引脚时, 如果更改前的引脚电平和更改后的引脚电平不同, 就可能在内部产生边沿, 导致意想不到的运行。因此在更改输入引脚时, 必须按照以下步骤进行设定:

1. 通过设定与要更改的引脚功能对应的外围模块, 将输入功能置为无效。
2. 通过设定端口功能寄存器, 选择输入引脚。
3. 通过设定与要更改的引脚功能对应的外围模块, 将输入功能置为有效。

在更改输出引脚时, 如果更改前的引脚设定和更改后的引脚设定不同, 就可能从引脚输出内部边沿, 导致意想不到的运行。因此在更改输出引脚时, 必须按照以下步骤进行设定:

1. 通过设定与要更改的引脚功能对应的外围模块, 将输出功能置为无效。
2. 通过设定端口功能寄存器, 选择输出引脚。
3. 通过设定与要更改的引脚功能对应的外围模块, 将输出功能置为有效。

对于 1 个引脚功能, 如果同时有用于更改输入 / 输出目标的引脚选择位以及用于将引脚功能置为有效的允许位, 就必须在通过引脚选择位设定引脚的输入 / 输出目标后, 通过允许位将引脚功能置为有效。

### 15.6.3 有关输出允许设定的转换

因为禁止各端口输出允许设定的初始状态, 所以引脚为 Hi-Z 状态。如果进行输出允许设定的转换, 各端口就从 HiZ 状态转换为输出 (High 电平输出或者 Low 电平输出), 此时可能因 LSI 内部状态而从引脚输出边沿。

要想不产生边沿, 需要预先给端口数据寄存器 (DR) 设定值。

在从 Hiz 状态转换为 High 电平输出时, 必须按照以下步骤进行转换:

1. 将输出引脚的端口数据寄存器 (DR) 的位置“1” (将 LSI 内部状态置为 High 电平)。
2. 通过设定端口功能寄存器, 选择输出引脚。
3. 将输出引脚功能转换为输出允许设定。

在从 Hiz 状态转换为 Low 电平输出时, 必须按照以下步骤进行转换:

1. 将输出引脚的端口数据寄存器 (DR) 的位置“0” (将 LSI 内部状态置为 Low 电平)。
2. 通过设定端口功能寄存器, 选择输出引脚。
3. 将输出引脚功能转换为输出允许设定。

### 15.6.4 读端口寄存器 (PORT) 时的注意事项

在读端口 4、端口 5、端口 6 的端口寄存器 (PORT) 时, 为了使对应引脚的输入缓冲器有效, 必须预先将 PORTn.ICR 寄存器的对应位置“1”。

如果在 PORTn.ICR 寄存器的位为“0”的状态下读 PORTn.PORT 寄存器, 对应位的值就为不定值。

## 16. 多功能定时器脉冲单元 3 (MTU3)

### 16.1 概要

RX62T 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 16.1 所示, MTU 的功能一览表如表 16.2 所示, MTU 的框图如图 16.1 和图 16.2 所示。

表 16.1 MTU 的规格

项目	内容
脉冲输入 / 输出	最多 24 个
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><b>【MTU0 ~ 4、6、7】</b></p> <ul style="list-style-type: none"> <li>通过比较匹配进行波形输出。</li> <li>输入捕捉功能</li> <li>计数器清除运行</li> <li>同时写多个定时器的计数器 (TCNT)。</li> <li>通过比较匹配或者输入捕捉进行同时清除。</li> <li>通过计数器的同步运行进行各寄存器的同步输入 / 输出。</li> <li>通过和同步运行的组合进行最多 12 相的 PWM 输出。</li> </ul> <p><b>【MTU0、3、4、6、7】</b></p> <ul style="list-style-type: none"> <li>能设定缓冲运行。</li> </ul> <p><b>【MTU3、4、6、7】</b></p> <ul style="list-style-type: none"> <li>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 6 相共 12 相。</li> <li>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。</li> <li>能在互补 PWM 模式中设定双缓冲功能。</li> </ul> <p><b>【MTU3、4】</b></p> <ul style="list-style-type: none"> <li>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。</li> </ul> <p><b>【MTU1、2】</b></p> <ul style="list-style-type: none"> <li>能分别设定相位计数模式。</li> <li>能级联运行。</li> </ul> <p><b>【MTU5】</b></p> <ul style="list-style-type: none"> <li>能用作补偿死区时间的计数器。</li> </ul>
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 转换的开始请求延迟功能, 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。

表 16.2 MTU 的功能一览表 (1/2)

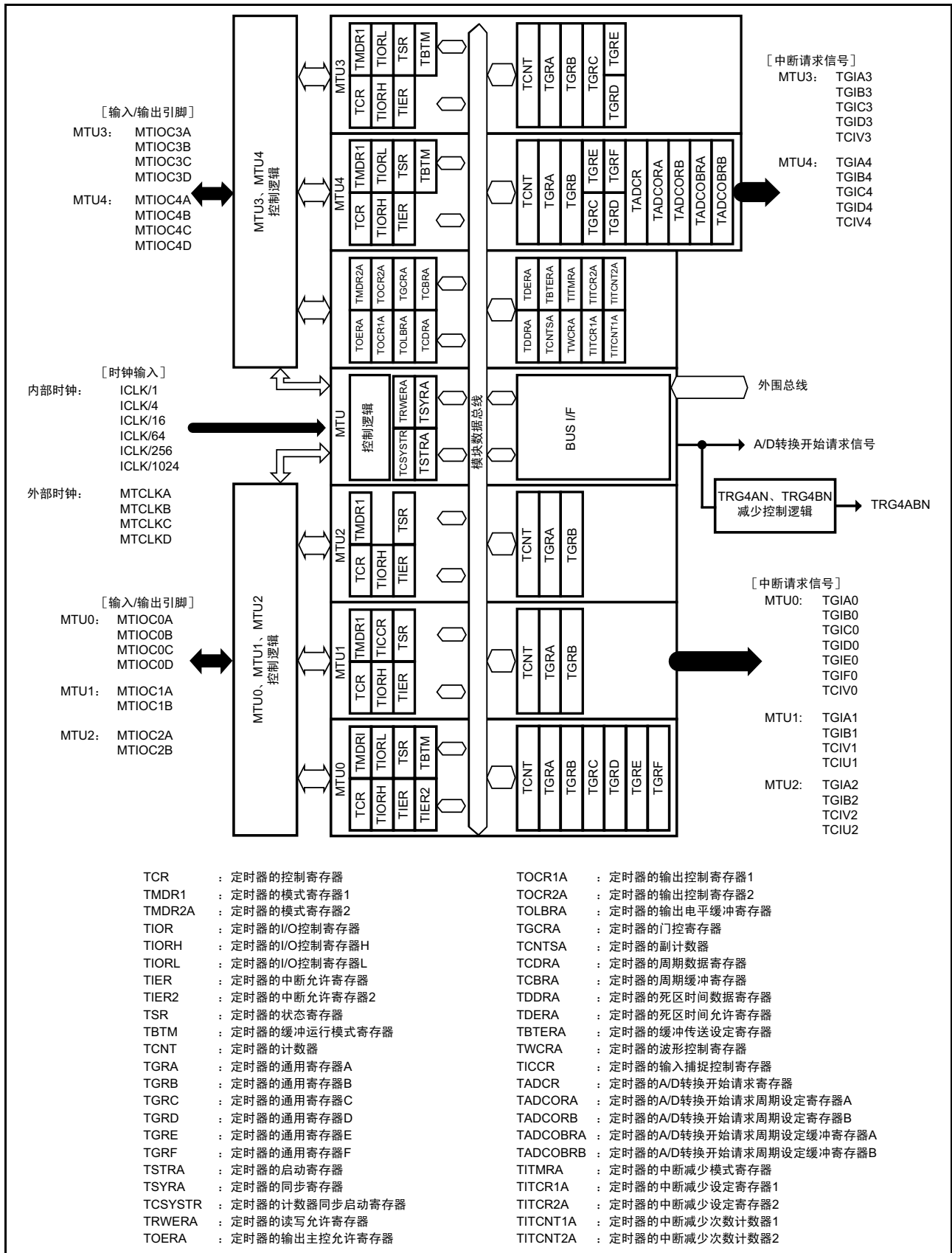
项目	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5	MTU6	MTU7	
计数时钟	ICLK/1 ICLK/4 ICLK/16 ICLK/64 MTCLKA MTCLKB MTCLKC MTCLKD	ICLK/1 ICLK/4 ICLK/16 ICLK/64 ICLK/256 MTCLKA MTCLKB	ICLK/1 ICLK/4 ICLK/16 ICLK/64 ICLK/1024 MTCLKA MTCLKB MTCLKC	ICLK/1 ICLK/4 ICLK/16 ICLK/64 ICLK/256 ICLK/1024 MTCLKA MTCLKB	ICLK/1 ICLK/4 ICLK/16 ICLK/64 ICLK/256 ICLK/1024 MTCLKA MTCLKB	ICLK/1 ICLK/4 ICLK/16 ICLK/64 ICLK/256 ICLK/1024 MTCLKA MTCLKB	ICLK/1 ICLK/4 ICLK/16 ICLK/64 ICLK/256 ICLK/1024	ICLK/1 ICLK/4 ICLK/16 ICLK/64 ICLK/256 ICLK/1024	ICLK/1 ICLK/4 ICLK/16 ICLK/64 ICLK/256 ICLK/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 16.2 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 转换的开始请求延迟功能	—	—	—	—	<ul style="list-style-type: none"> <li>在 TADCORA 和 TCNT 匹配时请求开始 A/D 转换</li> <li>在 TADCORB 和 TCNT 匹配时请求开始 A/D 转换</li> </ul>	—	—	<ul style="list-style-type: none"> <li>在 TADCORA 和 TCNT 匹配时请求开始 A/D 转换</li> <li>在 TADCORB 和 TCNT 匹配时请求开始 A/D 转换</li> </ul>
中断减少功能 1	—	—	—	<ul style="list-style-type: none"> <li>减少 TGRA 的比较匹配中断</li> </ul>	<ul style="list-style-type: none"> <li>减少 TCIV 中断</li> </ul>	—	<ul style="list-style-type: none"> <li>减少 TGRA 的比较匹配中断</li> </ul>	<ul style="list-style-type: none"> <li>减少 TCIV 中断</li> </ul>
中断减少功能 2	—	—	—	—	<ul style="list-style-type: none"> <li>减少 TADCORA 和 TCNT、以及 TADCORB 和 TCNT 的比较次数</li> </ul>	—	—	<ul style="list-style-type: none"> <li>减少 TADCORA 和 TCNT、以及 TADCORB 和 TCNT 的比较次数</li> </ul>
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“9. 低功耗功能”。



- TCR : 定时器的控制寄存器
- TMDR1 : 定时器的模式寄存器1
- TMDR2A : 定时器的模式寄存器2
- TIOR : 定时器的I/O控制寄存器
- TIORH : 定时器的I/O控制寄存器H
- TIORL : 定时器的I/O控制寄存器L
- TIER : 定时器的中断允许寄存器
- TIER2 : 定时器的中断允许寄存器2
- TSR : 定时器的状态寄存器
- TBTM : 定时器的缓冲运行模式寄存器
- TCNT : 定时器的计数器
- TGRA : 定时器的通用寄存器A
- TGRB : 定时器的通用寄存器B
- TGRC : 定时器的通用寄存器C
- TGRD : 定时器的通用寄存器D
- TGRE : 定时器的通用寄存器E
- TGRF : 定时器的通用寄存器F
- TSTRA : 定时器的启动寄存器
- TSYRA : 定时器的同步寄存器
- TCSYSTR : 定时器的计数器同步启动寄存器
- TRWERA : 定时器的读写允许寄存器
- TOERA : 定时器的输出主控允许寄存器

- TOCR1A : 定时器的输出控制寄存器1
- TOCR2A : 定时器的输出控制寄存器2
- TOLBRA : 定时器的输出电平缓缓冲寄存器
- TGCRA : 定时器的门控寄存器
- TCNTSA : 定时器的副计数器
- TCDRA : 定时器的周期数据寄存器
- TCBRA : 定时器的周期缓冲寄存器
- TDDRA : 定时器的死区时间数据寄存器
- TDERA : 定时器的死区时间允许寄存器
- TBTERA : 定时器的缓冲传送设定寄存器
- TWCRA : 定时器的波形控制寄存器
- TICCR : 定时器的输入捕捉控制寄存器
- TADCR : 定时器的A/D转换开始请求寄存器
- TADCORA : 定时器的A/D转换开始请求周期设定寄存器A
- TADCORB : 定时器的A/D转换开始请求周期设定寄存器B
- TADCOBRA : 定时器的A/D转换开始请求周期设定缓冲寄存器A
- TADCOBRB : 定时器的A/D转换开始请求周期设定缓冲寄存器B
- TITMRA : 定时器的中断减少模式寄存器
- TITCR1A : 定时器的中断减少设定寄存器1
- TITCR2A : 定时器的中断减少设定寄存器2
- TITCNT1A : 定时器的中断减少次数计数器1
- TITCNT2A : 定时器的中断减少次数计数器2

图 16.1 MTU 的框图 (MTU0 ~ 4)



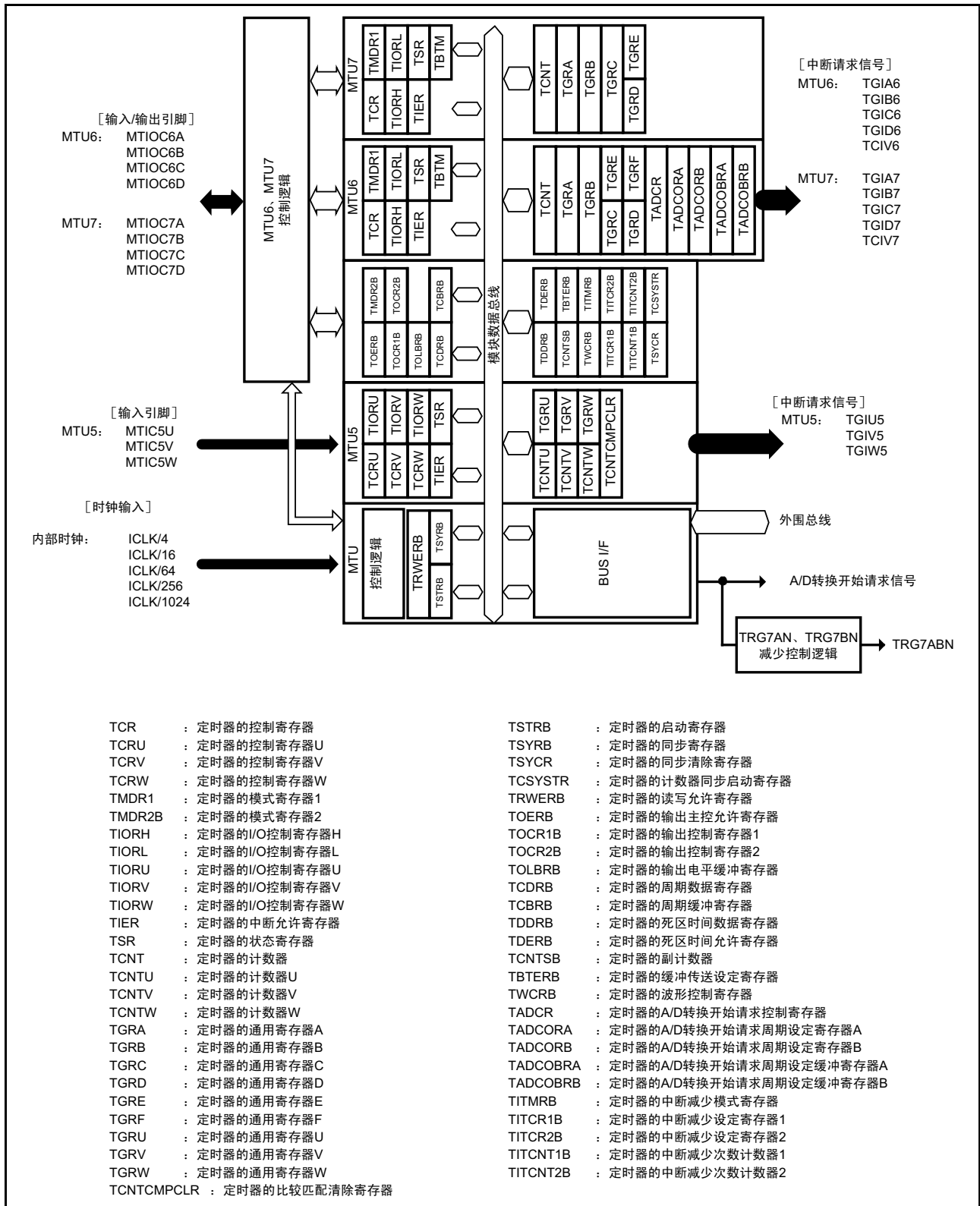


图 16.2 MTU 的框图 (MTU5 ~ 7)

MTU 使用的输入 / 输出引脚如表 16.3 所示。

表 16.3 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

## 16.2 寄存器说明

MTU 的寄存器一览表如表 16.4 所示。

表 16.4 MTU 的寄存器一览表 (1/6)

通道	寄存器名	寄存器符号	复位后的值	地址	存取长度
MTU0	定时器的控制寄存器	TCR	00h	000C 1300h	8、16、32
	定时器的模式寄存器 1	TMDR1	00h	000C 1301h	8
	定时器的 I/O 控制寄存器 H	TIORH	00h	000C 1302h	8、16
	定时器的 I/O 控制寄存器 L	TIORL	00h	000C 1303h	8
	定时器的中断允许寄存器	TIER	00h	000C 1304h	8、16、32
	定时器的状态寄存器	TSR	C0h	000C 1305h	8
	定时器的计数器	TCNT	0000h	000C 1306h	16
	定时器的通用寄存器 A	TGRA	FFFFh	000C 1308h	16、32
	定时器的通用寄存器 B	TGRB	FFFFh	000C 130Ah	16
	定时器的通用寄存器 C	TGRC	FFFFh	000C 130Ch	16、32
	定时器的通用寄存器 D	TGRD	FFFFh	000C 130Eh	16
	定时器的通用寄存器 E	TGRE	FFFFh	000C 1320h	16、32
	定时器的通用寄存器 F	TGRF	FFFFh	000C 1322h	16
	定时器的中断允许寄存器 2	TIER2	00h	000C 1324h	8、16
	定时器的状态寄存器 2	TSR2	C0h	000C 1325h	8
定时器的缓冲运行传送模式寄存器	TBTM	00h	000C 1326h	8	
MTU1	定时器的控制寄存器	TCR	00h	000C 1380h	8、16
	定时器的模式寄存器 1	TMDR1	00h	000C 1381h	8
	定时器的 I/O 控制寄存器	TIOR	00h	000C 1382h	8
	定时器的中断允许寄存器	TIER	00h	000C 1384h	8、16、32
	定时器的状态寄存器	TSR	C0h	000C 1385h	8
	定时器的计数器	TCNT	0000h	000C 1386h	16
	定时器的通用寄存器 A	TGRA	FFFFh	000C 1388h	16、32
	定时器的通用寄存器 B	TGRB	FFFFh	000C 138Ah	16
	定时器的输入捕捉控制寄存器	TICCR	00h	000C 1390h	8
MTU2	定时器的控制寄存器	TCR	00h	000C 1400h	8、16
	定时器的模式寄存器 1	TMDR1	00h	000C 1401h	8
	定时器的 I/O 控制寄存器	TIOR	00h	000C 1402h	8
	定时器的中断允许寄存器	TIER	00h	000C 1404h	8、16、32
	定时器的状态寄存器	TSR	C0h	000C 1405h	8
	定时器的计数器	TCNT	0000h	000C 1406h	16
	定时器的通用寄存器 A	TGRA	FFFFh	000C 1408h	16、32
	定时器的通用寄存器 B	TGRB	FFFFh	000C 140Ah	16

表 16.4 MTU 的寄存器一览表 (2/6)

通道	寄存器名	寄存器符号	复位后的值	地址	存取长度
MTU3	定时器的控制寄存器	TCR	00h	000C 1200h	8、16、32
	定时器的模式寄存器 1	TMDR1	00h	000C 1202h	8、16
	定时器的 I/O 控制寄存器 H	TIORH	00h	000C 1204h	8、16、32
	定时器的 I/O 控制寄存器 L	TIORL	00h	000C 1205h	8
	定时器的中断允许寄存器	TIER	00h	000C 1208h	8、16
	定时器的计数器	TCNT	0000h	000C 1210h	16、32
	定时器的通用寄存器 A	TGRA	FFFFh	000C 1218h	16、32
	定时器的通用寄存器 B	TGRB	FFFFh	000C 121Ah	16
	定时器的通用寄存器 C	TGRC	FFFFh	000C 1224h	16、32
	定时器的通用寄存器 D	TGRD	FFFFh	000C 1226h	16
	定时器的通用寄存器 E	TGRE	FFFFh	000C 1272h	16
	定时器的状态寄存器	TSR	C0h	000C 122Ch	8、16
	定时器的缓冲运行传送模式寄存器	TBTM	00h	000C 1238h	8、16
	MTU4	定时器的控制寄存器	TCR	00h	000C 1201h
定时器的模式寄存器 1		TMDR1	00h	000C 1203h	8
定时器的 I/O 控制寄存器 H		TIORH	00h	000C 1206h	8、16
定时器的 I/O 控制寄存器 L		TIORL	00h	000C 1207h	8
定时器的中断允许寄存器		TIER	00h	000C 1209h	8
定时器的计数器		TCNT	0000h	000C 1212h	16
定时器的通用寄存器 A		TGRA	FFFFh	000C 121Ch	16、32
定时器的通用寄存器 B		TGRB	FFFFh	000C 121Eh	16
定时器的通用寄存器 C		TGRC	FFFFh	000C 1228h	16、32
定时器的通用寄存器 D		TGRD	FFFFh	000C 122Ah	16
定时器的通用寄存器 E		TGRE	FFFFh	000C 1274h	16
定时器的状态寄存器		TGRF	FFFFh	000C 1276h	16
定时器的缓冲运行传送模式寄存器		TSR	C0h	000C 122Dh	8
定时器的控制寄存器		TBTM	00h	000C 1239h	8
定时器的 A/D 转换开始请求控制寄存器		TADCR	0000h	000C 1240h	16
定时器的 A/D 转换开始请求周期设定寄存器 A		TADCORA	FFFFh	000C 1244h	16、32
定时器的 A/D 转换开始请求周期设定寄存器 B		TADCORB	FFFFh	000C 1246h	16
定时器的 A/D 转换开始请求周期设定缓冲寄存器 A		TADCOBRA	FFFFh	000C 1248h	16、32
定时器的 A/D 转换开始请求周期设定缓冲寄存器 B		TADCOBRB	FFFFh	000C 124Ah	16

表 16.4 MTU 的寄存器一览表 (3/6)

通道	寄存器名	寄存器符号	复位后的值	地址	存取长度
MTU5	定时器的计数器 U	TCNTU	0000h	000C 1C80h	16、32
	定时器的通用寄存器 U	TGRU	FFFFh	000C 1C82h	16
	定时器的控制寄存器 U	TCRU	00h	000C 1C84h	8
	定时器的 I/O 控制寄存器 U	TIORU	00h	000C 1C86h	8
	定时器的计数器 V	TCNTV	0000h	000C 1C90h	16、32
	定时器的通用寄存器 V	TGRV	FFFFh	000C 1C92h	16
	定时器的控制寄存器 V	TCRV	00h	000C 1C94h	8
	定时器的 I/O 控制寄存器 V	TIORV	00h	000C 1C96h	8
	定时器的计数器 W	TCNTW	0000h	000C 1CA0h	16、32
	定时器的通用寄存器 W	TGRW	FFFFh	000C 1CA2h	16
	定时器的控制寄存器 W	TCRW	00h	000C 1CA4h	8
	定时器的 I/O 控制寄存器 W	TIORW	00h	000C 1CA6h	8
	定时器的状态寄存器	TSR	00h	000C 1CB0h	8
	定时器的中断允许寄存器	TIER	00h	000C 1CB2h	8
	定时器的启动寄存器	TSTR	00h	000C 1CB4h	8
	定时器的比较匹配清除寄存器	TCNTCMPCLR	00h	000C 1CB6h	8
MTU6	定时器的控制寄存器	TCR	00h	000C 1A00h	8、16、32
	定时器的模式寄存器 1	TMDR1	00h	000C 1A02h	8、16
	定时器的 I/O 控制寄存器 H	TIORH	00h	000C 1A04h	8、16、32
	定时器的 I/O 控制寄存器 L	TIORL	00h	000C 1A05h	8
	定时器的中断允许寄存器	TIER	00h	000C 1A08h	8、16
	定时器的计数器	TCNT	0000h	000C 1A10h	16、32
	定时器的通用寄存器 A	TGRA	FFFFh	000C 1A18h	16、32
	定时器的通用寄存器 B	TGRB	FFFFh	000C 1A1Ah	16
	定时器的通用寄存器 C	TGRC	FFFFh	000C 1A24h	16、32
	定时器的通用寄存器 D	TGRD	FFFFh	000C 1A26h	16
	定时器的通用寄存器 E	TGRE	FFFFh	000C 1A72h	16
	定时器的同步清除寄存器	TSYCR	00h	000C 1A50h	8
	定时器的状态寄存器	TSR	C0h	000C 1A2Ch	8、16
	定时器的缓冲运行传送模式寄存器	TBTM	00h	000C 1A38h	8、16

表 16.4 MTU 的寄存器一览表 (4/6)

通道	寄存器名	寄存器符号	复位后的值	地址	存取长度
MTU7	定时器的控制寄存器	TCR	00h	000C 1A01h	8
	定时器的模式寄存器 1	TMDR1	00h	000C 1A03h	8
	定时器的 I/O 控制寄存器 H	TIORH	00h	000C 1A06h	8、16
	定时器的 I/O 控制寄存器 L	TIORL	00h	000C 1A07h	8
	定时器的中断允许寄存器	TIER	00h	000C 1A09h	8
	定时器的计数器	TCNT	0000h	000C 1A12h	16
	定时器的通用寄存器 A	TGRA	FFFFh	000C 1A1Ch	16、32
	定时器的通用寄存器 B	TGRB	FFFFh	000C 1A1Eh	16
	定时器的通用寄存器 C	TGRC	FFFFh	000C 1A28h	16、32
	定时器的通用寄存器 D	TGRD	FFFFh	000C 1A2Ah	16
	定时器的通用寄存器 E	TGRE	FFFFh	000C 1A74h	16
	定时器的通用寄存器 F	TGRF	FFFFh	000C 1A76h	16
	定时器的状态寄存器	TSR	C0h	000C 1A2Dh	8
	定时器的缓冲运行传送模式寄存器	TBTM	00h	000C 1A39h	8
	定时器的 A/D 转换开始请求控制寄存器	TADCR	0000h	000C 1A40h	16
	定时器的 A/D 转换开始请求周期设定寄存器 A	TADCORA	FFFFh	000C 1A44h	16、32
	定时器的 A/D 转换开始请求周期设定寄存器 B	TADCORB	FFFFh	000C 1A46h	16
	定时器的 A/D 转换开始请求周期设定缓冲寄存器 A	TADCOBRA	FFFFh	000C 1A48h	16、32
	定时器的 A/D 转换开始请求周期设定缓冲寄存器 B	TADCOBRB	FFFFh	000C 1A4Ah	16
	MTU (MTU3、MTU4 共同)	定时器的输出主控允许寄存器 A	TOERA	C0h	000C 120Ah
MTU (MTU6、MTU7 共同)	定时器的输出主控允许寄存器 B	TOERB	C0h	000C 1A0Ah	8
MTU (MTU3、MTU4 共同)	定时器的门控寄存器 A	TGCRA	80h	000C 120Dh	8
MTU (MTU3、MTU4 共同)	定时器的输出控制寄存器 1A	TOCR1A	00h	000C 120Eh	8、16
MTU (MTU6、MTU7 共同)	定时器的输出控制寄存器 1B	TOCR1B	00h	000C 1A0Eh	8、16
MTU (MTU3、MTU4 共同)	定时器的输出控制寄存器 2A	TOCR2A	00h	000C 120Fh	8
MTU (MTU6、MTU7 共同)	定时器的输出控制寄存器 2B	TOCR2B	00h	000C 1A0Fh	8
MTU (MTU3、MTU4 共同)	定时器的周期数据寄存器 A	TCDRA	FFFFh	000C 1214h	16、32
MTU (MTU6、MTU7 共同)	定时器的周期数据寄存器 B	TCDRB	FFFFh	000C 1A14h	16、32
MTU (MTU3、MTU4 共同)	定时器的死区时间数据寄存器 A	TDDRA	FFFFh	000C 1216h	16
MTU (MTU3、MTU4 共同)	定时器的死区时间允许寄存器 A	TDERA	01h	000C 1234h	8
MTU (MTU6、MTU7 共同)	定时器的死区时间数据寄存器 B	TDDRB	FFFFh	000C 1A16h	16

表 16.4 MTU 的寄存器一览表 (5/6)

通道	寄存器名	寄存器符号	复位后的值	地址	存取长度
MTU (MTU6、MTU7 共同)	定时器的死区时间允许寄存器 B	TDERB	01h	000C 1A34h	8
MTU (MTU3、MTU4 共同)	定时器的副计数器 A	TCNTSA	0000h	000C 1220h	16、32
MTU (MTU6、MTU7 共同)	定时器的副寄存器 B	TCNTSB	0000h	000C 1A20h	16、32
MTU (MTU3、MTU4 共同)	定时器的周期缓冲寄存器 A	TCBRA	FFFFh	000C 1222h	16
MTU (MTU6、MTU7 共同)	定时器的周期缓冲寄存器 B	TCBRB	FFFFh	000C 1A22h	16
MTU (MTU3、MTU4 共同)	定时器的中断减少设定寄存器 1A	TITCR1A	00h	000C 1230h	8、16
MTU (MTU6、MTU7 共同)	定时器的中断减少设定寄存器 1B	TITCR1B	00h	000C 1A30h	8、16
MTU (MTU3、MTU4 共同)	定时器的中断减少设定寄存器 2A	TITCR2A	00h	000C 123Bh	8
MTU (MTU6、MTU7 共同)	定时器的中断减少设定寄存器 2B	TITCR2B	00h	000C 1A3Bh	8
MTU (MTU3、MTU4 共同)	定时器的中断减少次数计数器 1A	TITCNT1A	00h	000C 1231h	8
MTU (MTU6、MTU7 共同)	定时器的中断减少次数计数器 1B	TITCNT1B	00h	000C 1A31h	8
MTU (MTU3、MTU4 共同)	定时器的中断减少次数计数器 2A	TITCNT2A	00h	000C 123Ch	8
MTU (MTU6、MTU7 共同)	定时器的中断减少次数计数器 2B	TITCNT2B	00h	000C 1A3Ch	8
MTU (MTU3、MTU4 共同)	定时器的缓冲传送设定寄存器 A	TBTERA	00h	000C 1232h	8
MTU (MTU6、MTU7 共同)	定时器的缓冲传送设定寄存器 B	TBTERB	00h	000C 1A32h	8
MTU (MTU3、MTU4 共同)	定时器的输出电平缓冲寄存器 A	TOLBRA	00h	000C 1236h	8
MTU (MTU6、MTU7 共同)	定时器的输出电平缓冲寄存器 B	TOLBRB	00h	000C 1A36h	8
MTU (MTU3、MTU4 共同)	定时器的模式寄存器 2A	TMDR2A	00h	000C 1270h	8
MTU (MTU6、MTU7 共同)	定时器的模式寄存器 2B	TMDR2B	00h	000C 1A70h	8
MTU (MTU3、MTU4 共同)	定时器的减少模式寄存器 A	TITMRA	00h	000C 123Ah	8
MTU (MTU6、MTU7 共同)	定时器的减少模式寄存器 B	TITMRB	00h	000C 1A3Ah	8
MTU (MTU3、MTU4 共同)	定时器的波形控制寄存器 A	TWCRA	00h	000C 1260h	8
MTU (MTU6、MTU7 共同)	定时器的波形控制寄存器 B	TWCRB	00h	000C 1A60h	8
MTU (MTU0 ~ MTU4 共同)	定时器的开始寄存器 A	TSTRA	00h	000C 1280h	8、16

表 16.4 MTU 的寄存器一览表 (6/6)

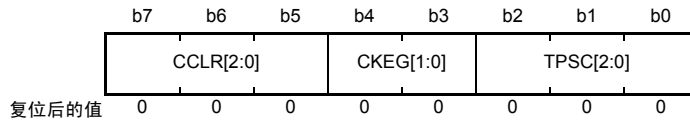
通道	寄存器名	寄存器符号	复位后的值	地址	存取长度
MTU (MTU6、MTU7 共同)	定时器的开始寄存器 B	TSTRB	00h	000C 1A80h	8、16
MTU (MTU0 ~ MTU4 共同)	定时器的同步寄存器 A	TSYRA	00h	000C 1281h	8
MTU (MTU6、MTU7 共同)	定时器的同步寄存器 B	TSYRB	00h	000C 1A81h	8
MTU (MTU0 ~ MTU7 共同)	定时器的计数器同步启动寄存器	TCSYSTR	00h	000C 1282h	8
MTU (MTU3、MTU4 共同)	定时器的读写允许寄存器 A	TRWERA	01h	000C 1284h	8
MTU (MTU6、MTU7 共同)	定时器的读写允许寄存器 B	TRWERB	01h	000C 1A84h	8



### 16.2.1 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 16.7 ~ 表 16.10。	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 16.5 和表 16.6。	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 16.7 ~ 表 16.10。

#### CKEG[1:0] 位 (时钟边沿选择位)

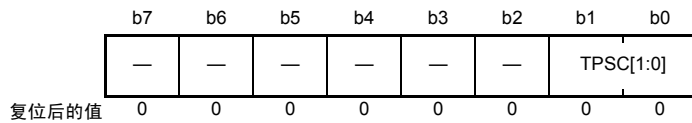
这些位选择输入时钟的边沿。如果在双边沿对内部时钟进行计数，输入时钟的周期就为 1/2 (例如: ICLK/4 的双边沿 = ICLK/2 的上升沿)。在 MTU1 和 MTU2 中使用相位计数模式时，忽视此设定而优先设定相位计数模式。内部时钟的边沿选择在输入时钟为 ICLK/4 或者慢于 ICLK/4 时有效。如果选择 ICLK/1 或者其他通道的上溢 / 下溢作为输入时钟，虽然能写值，但是在运行时为初始值。

#### 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 16.5 和表 16.6。

- 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[2:0]	定时器的预分频器选择位	请参照表 16.11。	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 16.11。

表 16.5 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 16.6 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 16.7 TPSC[2:0] (MTU0)

通道	bit2	bit1	bit0	说明
	TPSC2	TPSC1	TPSC0	
MTU0	0	0	0	内部时钟: 通过 ICLK/1 进行计数
	0	0	1	内部时钟: 通过 ICLK/4 进行计数
	0	1	0	内部时钟: 通过 ICLK/16 进行计数
	0	1	1	内部时钟: 通过 ICLK/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 16.8 TPSC[2:0] (MTU1)

通道	bit2	bit1	bit0	说明
	TPSC2	TPSC1	TPSC0	
MTU1	0	0	0	内部时钟: 通过 ICLK/1 进行计数
	0	0	1	内部时钟: 通过 ICLK/4 进行计数
	0	1	0	内部时钟: 通过 ICLK/16 进行计数
	0	1	1	内部时钟: 通过 ICLK/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 ICLK/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 16.9 TPSC[2:0] (MTU2)

通道	bit2	bit1	bit0	说明
	TPSC2	TPSC1	TPSC0	
MTU2	0	0	0	内部时钟: 通过 ICLK/1 进行计数
	0	0	1	内部时钟: 通过 ICLK/4 进行计数
	0	1	0	内部时钟: 通过 ICLK/16 进行计数
	0	1	1	内部时钟: 通过 ICLK/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 ICLK/1024 进行计数

注. 在 MTU2 为相位计数模式时, 此设定无效。

表 16.10 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 ICLK/1 进行计数
MTU4	0	0	1	内部时钟: 通过 ICLK/4 进行计数
MTU6	0	1	0	内部时钟: 通过 ICLK/16 进行计数
MTU7	0	1	1	内部时钟: 通过 ICLK/64 进行计数
	1	0	0	内部时钟: 通过 ICLK/256 进行计数
	1	0	1	内部时钟: 通过 ICLK/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 16.11 TPSC[1:0] (MTU5)

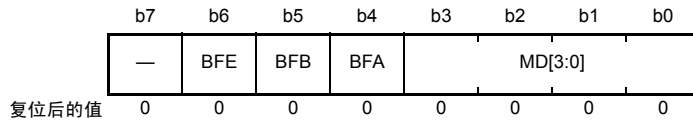
通道	bit1	bit0	说明
	TPSC1	TPSC0	
MTU5	0	0	内部时钟: 通过 ICLK/1 进行计数
	0	1	内部时钟: 通过 ICLK/4 进行计数
	1	0	内部时钟: 通过 ICLK/16 进行计数
	1	1	内部时钟: 通过 ICLK/64 进行计数

注. 在 MTU5 中, bit7 ~ 2 为保留位, 读写值都为“0”。

## 16.2.2 定时器的模式寄存器 1 (TMDR1)

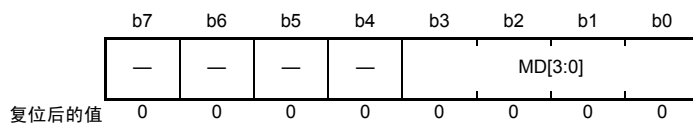
- MTU0.TMDR1

地址 MTU0.TMDR1 000C 1301h



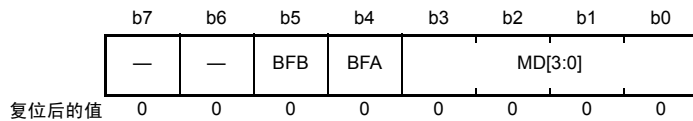
- MTU1.TMDR1、MRU2.TMDR1

地址 MTU1.TMDR1 000C 1381h、MTU2.TMDR1 000C 1401h



- 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



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 16.12。	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 16.12 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”。

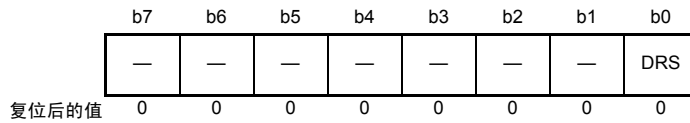
#### BFE 位 (缓冲运行 E 位)

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”。

## 16.2.3 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。

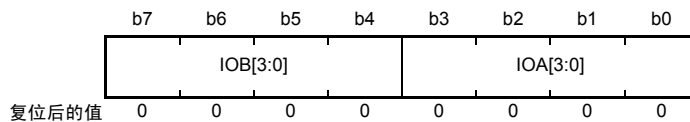
## DRS 位 (双缓冲选择位)

此位选择互补 PWM 模式中的双缓冲功能的有效 / 无效。

## 16.2.4 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

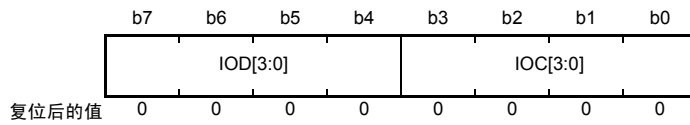
地址 MTU0.TIORH 000C 1302h、MTU1.TIOR 000C 1382h、MTU2.TIOR 000C 1402h、MTU3.TIORH 000C 1204h、MTU4.TIORH 000C 1206h、MTU6.TIORH 000C 1A04h、MTU7.TIORH 000C 1A06h



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 16.25、MTU1.TIOR: 表 16.27 MTU2.TIOR: 表 16.28、MTU3.TIORH: 表 16.29 MTU4.TIORH: 表 16.31、MTU6.TIORH: 表 16.33 MTU7.TIORH: 表 16.35	R/W
b7-b4	IOB[3:0]	I/O 控制 B 位	请参照下表: MTU0.TIORH: 表 16.13、MTU1.TIOR: 表 16.15、 MTU2.TIOR: 表 16.16、MTU3.TIORH: 表 16.17、 MTU4.TIORH: 表 16.19、MTU6.TIORH: 表 16.21、 MTU7.TIORH: 表 16.23	R/W

- MTU0.TIORL、MTU3.TIORL、MTU4.TIORL、MTU6.TIORL、MTU7.TIORL

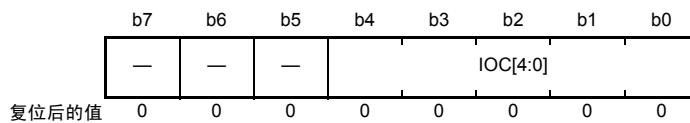
地址 MTU0.TIORL 000C 1303h、MTU3.TIORL 000C 1205h、MTU4.TIORL 000C 1207h、MTU6.TIORL 000C 1A05h、MTU7.TIORL 000C 1A07h



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 16.26、MTU3.TIORL：表 16.30、 MTU4.TIORL：表 16.32、MTU6.TIORL：表 16.34、 MTU7.TIORL：表 16.36	R/W
b7-b4	IOD[3:0]	I/O 控制 D 位	请参照下表： MTU0.TIORL：表 16.14、MTU3.TIORL：表 16.18、 MTU4.TIORL：表 16.20、MTU6.TIORL：表 16.22、 MTU7.TIORL：表 16.24	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 16.37	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 16.13 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 16.14 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		捕捉输入源为通道 1/ 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.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 16.15 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 16.16 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 16.17 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 16.18 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 16.19 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 16.20 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 16.21 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 16.22 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 16.23 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 16.24 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 16.25 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 16.26 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 16.27 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 16.28 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 16.29 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 16.30 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 16.31 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 16.32 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 16.33 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 16.34 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 16.35 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 16.36 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 16.37 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

## 16.2.5 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。

### 16.2.6 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<sub>n</sub> 标志变为“1”时，这些位允许或者禁止因 TSR.TGF<sub>n</sub> 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<sub>n</sub> 标志变为“1”时，这些位允许或者禁止因 TSR.TGF<sub>n</sub> 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<sub>n</sub>.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<sub>n</sub> 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<sub>n</sub> 位 (TGR 中断允许 5<sub>n</sub> 位)

在 MTU5.TSR.CMF<sub>n5</sub> 标志变为“1”时, 这些位允许或者禁止因 MTU5.TSR.CMF<sub>n5</sub> 标志而产生的中断请求 (TGIn5) (n=U、V、W)。

### 16.2.7 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，通道 1 ~ 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 (比较匹配标志 E)

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 (比较匹配标志 F)

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

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CMFU5	CMFV5	CMFW5
复位后的值	0	0	0	0	0	0	0	0

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	R/(W) (注1)
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	R/(W) (注1)
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	R/(W) (注1)
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。

### 16.2.8 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

- 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

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”。

## 16.2.9 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)

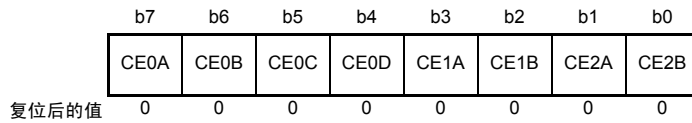
此位选择是否将 MTIOC2A 引脚追加到 MTU1.TGRA 的输入捕捉条件。

## 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。

## 16.2.10 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

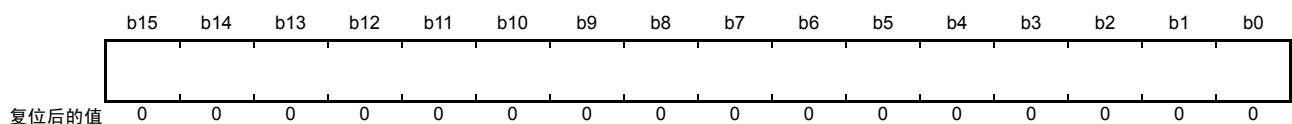
TSYCR 寄存器设定 MTU 的 MTU6.TCNT 和 MTU7.TCNT 的同步清除条件。MTU6 有一个 TSYCR。

CE<sub>nm</sub> 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<sub>n</sub>.TSR 的 TGF<sub>m</sub> 标志置位时清除。

## 16.2.11 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。

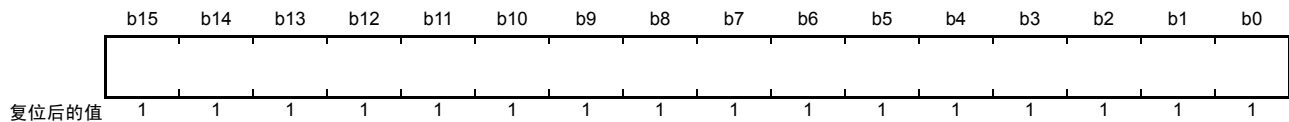
在复位时，TCNT 被初始化为“0000h”。

禁止以 8 位为单位存取 TCNT。必须以 16 位为单位进行存取。



## 16.2.12 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。

## 16.2.13 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。

## 16.2.14 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。

## 16.2.15 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

## 16.2.16 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)。



## 16.2.17 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

位	符号	位名	功能	R/W
b0	OE3B	主控允许 MTIOC3B 位	0: 禁止 MTU 输出 (注 1) 1: 允许 MTU 输出	R/W
b1	OE4A	主控允许 MTIOC4A 位	0: 禁止 MTU 输出 (注 1) 1: 允许 MTU 输出	R/W
b2	OE4B	主控允许 MTIOC4B 位	0: 禁止 MTU 输出 (注 1) 1: 允许 MTU 输出	R/W
b3	OE3D	主控允许 MTIOC3D 位	0: 禁止 MTU 输出 (注 1) 1: 允许 MTU 输出	R/W
b4	OE4C	主控允许 MTIOC4C 位	0: 禁止 MTU 输出 (注 1) 1: 允许 MTU 输出	R/W
b5	OE4D	主控允许 MTIOC4D 位	0: 禁止 MTU 输出 (注 1) 1: 允许 MTU 输出	R/W
b7-b6	—	保留位	读写值都为“1”。	R/W

注 1. 在设定为禁止 MTU 输出时, 如果要从各引脚输出无效电平, 就必须事先给 I/O 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定“向通用输入 / 输出端口输出无效电平”。详细内容请参照“15. I/O 端口.”。

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 16.36 和图 16.39)。

## 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 (注 1) 1: 允许 MTU 输出	R/W
b1	OE7A	主控允许 MTIOC7A 位	0: 禁止 MTU 输出 (注 1) 1: 允许 MTU 输出	R/W
b2	OE7B	主控允许 MTIOC7B 位	0: 禁止 MTU 输出 (注 1) 1: 允许 MTU 输出	R/W
b3	OE36D	主控允许 MTIOC6D 位	0: 禁止 MTU 输出 (注 1) 1: 允许 MTU 输出	R/W
b4	OE7C	主控允许 MTIOC7C 位	0: 禁止 MTU 输出 (注 1) 1: 允许 MTU 输出	R/W
b5	OE7D	主控允许 MTIOC7D 位	0: 禁止 MTU 输出 (注 1) 1: 允许 MTU 输出	R/W
b7-b6	—	保留位	读写值都为“1”。	R/W

注 1. 在设定为禁止 MTU 输出时, 如果要从各引脚输出无效电平, 就必须事先给 I/O 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定“向通用输入 / 输出端口输出无效电平”。详细内容请参照“15. I/O 端口.”。

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。

## 16.2.18 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
复位后的值	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 16.38。	R/W
b1	OLSN	输出电平选择 N 位 (注1、注3)	请参照表 16.39。	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、注4)	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 16.38 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 16.39 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 16.3 所示。

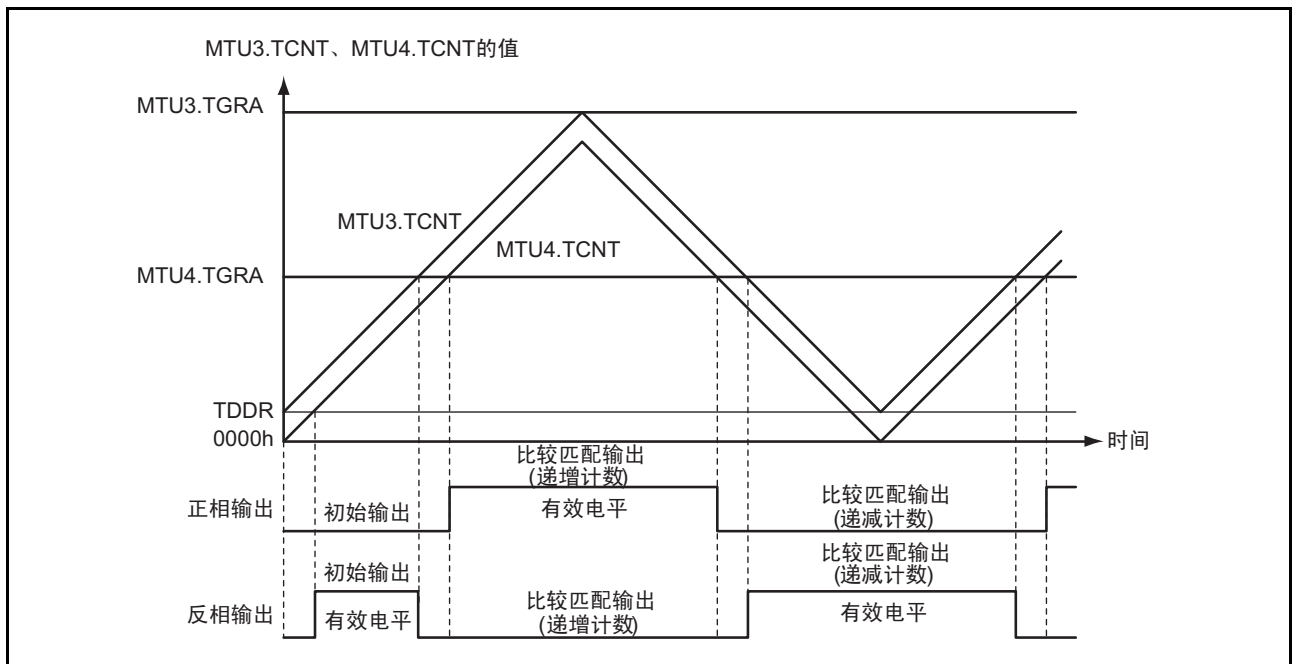


图 16.3 互补 PWM 模式的输出电平例子

## 16.2.19 定时器的输出控制寄存器 2 (TOCR2A、TOCR2B)

地址 MTU.TOCR2A 000C 120Fh、MTU.TOCR2B 000C 1A0Fh

b7	b6	b5	b4	b3	b2	b1	b0
BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
复位后的值		0	0	0	0	0	0

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 16.40。	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 16.41。	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 16.42。	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 16.43。	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 16.44。	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 16.45。	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 16.46。	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 16.40 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 16.41 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 16.42 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 16.43 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 16.44 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 16.45 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 16.46 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

## 16.2.20 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

b7	b6	b5	b4	b3	b2	b1	b0
—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
复位后的值	0	0	0	0	0	0	0

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 16.4 所示。

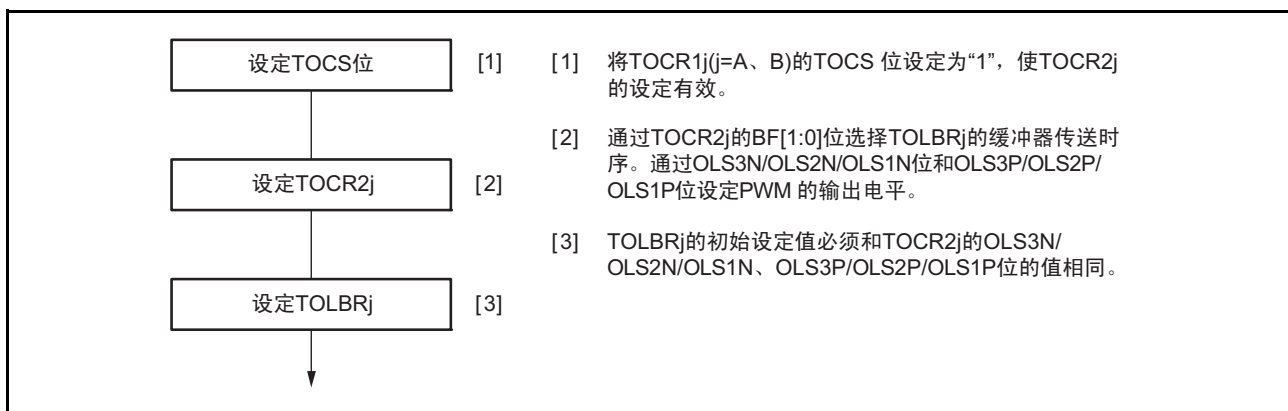


图 16.4 在缓冲运行中设定 PWM 输出电平时的设定步骤例子

## 16.2.21 定时器的门控寄存器 A (TGCR A)

地址 MTU.TGCR A 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 TGCR.FB 位为“1”时有效。此时，b0 ~ b2 的设定取代外部输入，请参照表 16.47。	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（TGCR A 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

TGCR A 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 TGCR A 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 TGCR A 寄存器功能的有效或者无效。

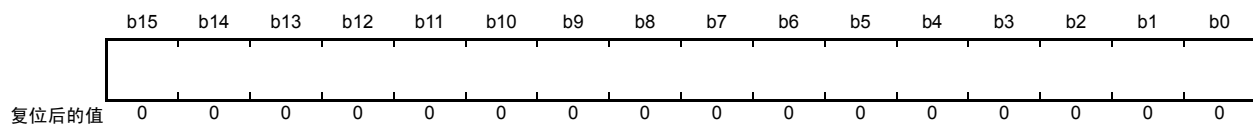


表 16.47 输出电平的选择功能

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

### 16.2.22 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”。

### 16.2.23 定时器的周期数据寄存器 (TCDRA、TCDRB)

地址 MTU.TCDRA 000C 1214h、MTU.TCDRB 000C 1A14h

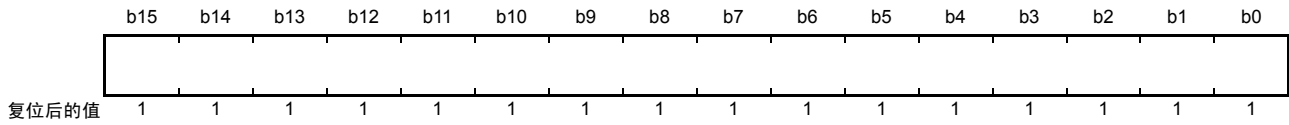


注. 禁止以 8 位为单位存取 TCDRA 寄存器和 TCDRB 寄存器。必须以 16 位为单位进行存取。

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”。

### 16.2.24 定时器的周期缓冲寄存器 (TCBRA、TCBRB)

地址 MTU.TCBRA 000C 1222h、MTU.TCBRB 000C 1A22h

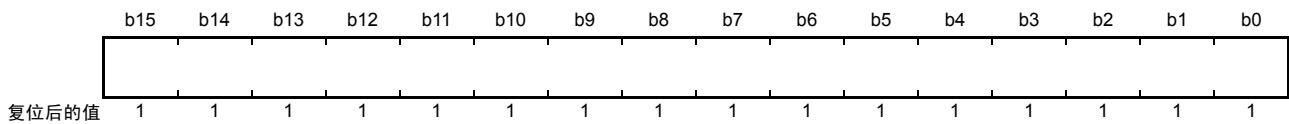


注. 禁止以 8 位为单位存取 TCBRA 寄存器和 TCBRB 寄存器。必须以 16 位为单位进行存取。

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”。

### 16.2.25 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

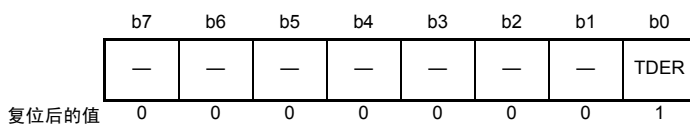


注. 禁止以 8 位为单位存取 TDDRA 寄存器和 TDDRB 寄存器。必须以 16 位为单位进行存取。

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”。

### 16.2.26 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)

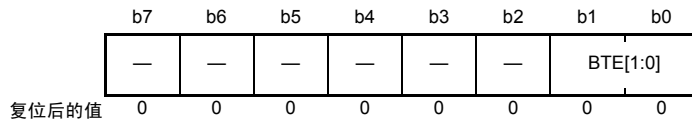
此位设定是否生成死区时间。

[ 为“0”的条件 ]

- 在 TDER 为“1”的状态下读 TDER 后，给 TDER 写“0”时

## 16.2.27 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 16.48。	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.TCBRB

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 16.48 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“16.3.8 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。

## 16.2.28 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 16.41。

[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 16.41。

[ 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

## 16.2.29 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 TCI4V 中断减少功能 1 1: A/D 转换开始请求 (TRG4BN) 联动 TCI4V 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 TCI4V 中断减少功能 1 1: A/D 转换开始请求 (TRG4AN) 联动 TCI4V 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 16.49。	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 16.49 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 16.50。	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，不能写“1”。

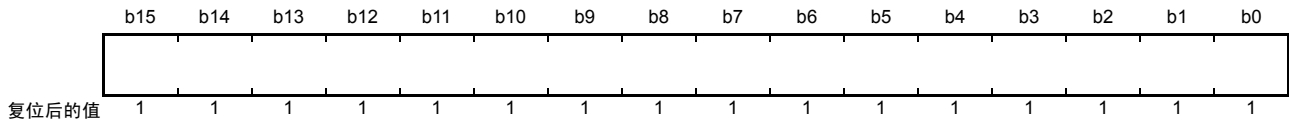
表 16.50 通过 TADCR.BF[1:0] 位进行传送时序的设定 (MTU7)

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。

### 16.2.30 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“16.3.9(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。

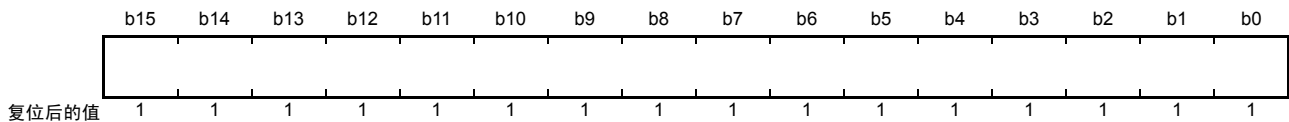
- (1) 当使用中中断减少功能 2 并且中断减少次数=0 时
- MTUn.TADCORA 寄存器值与 MTUn.TADCORB 寄存器值之间的间隔  $\geq 4$
  - MTUn.TADCORA 的比较间隔  $\geq 4$  个 ICLK (将 MTUn.TADCORA 寄存器的更新值设定为: 更新前的值 + 大于等于 4 的值或者更新前的值 - 小于等于 4 的值)
  - MTUn.TADCORB 的比较间隔  $\geq 4$  个 ICLK (将 MTUn.TADCORB 寄存器的更新值设定为: 更新前的值 + 大于等于 4 的值或者更新前的值 - 小于等于 4 的值)
- (2) 当使用中中断减少功能 2 并且中断减少次数  $\geq 1$  时
- MTUn.TADCORA 寄存器值与 MTUn.TADCORB 寄存器值之间的间隔  $\geq 2$
  - MTUn.TADCORB 的比较间隔  $\geq 2$  个 ICLK (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”。

### 16.2.31 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



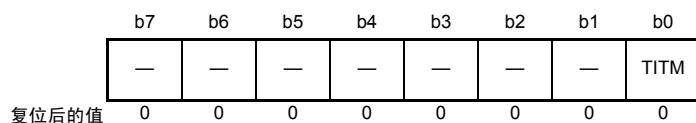
- 注. 禁止以 8 位为单位存取 TADCOBRA 寄存器和 TADCOBRB 寄存器。必须以 16 位为单位进行存取。

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”。

## 16.2.32 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 16.51.	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 16.51 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)

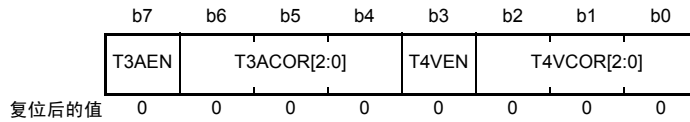
注 1. 通过设定 TITCR1A 寄存器和 TITCR1B 寄存器使中断减少功能 1 有效。

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。

## 16.2.33 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 16.52。	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 16.53。	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

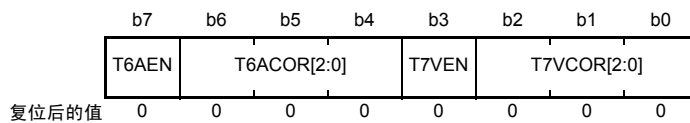
注 1. 如果将中断减少次数设定为“0”, 就不减少中断。

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 16.54。	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 16.55。	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 16.52 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 16.53 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 16.54 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。

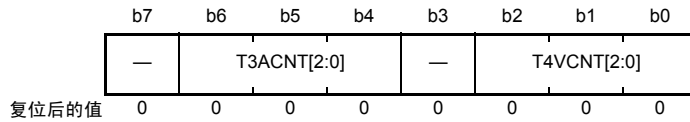
表 16.55 通过 T6ACOR[2:0] 位设定的中断减少次数

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。

## 16.2.34 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)

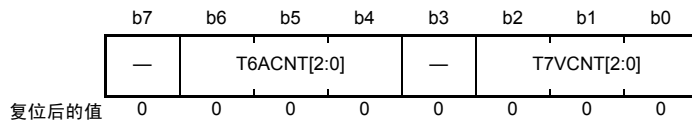
如果 TITCR1A.T3AEN 位为“1”，就在产生 TGIA3 中断源时递增 1。

[ 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。

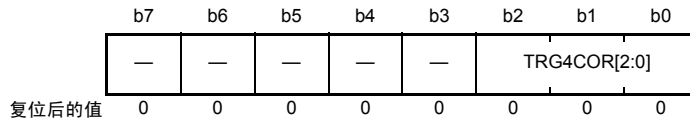
[ 为“0”的条件 ]

- 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

## 16.2.35 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 16.56。	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。

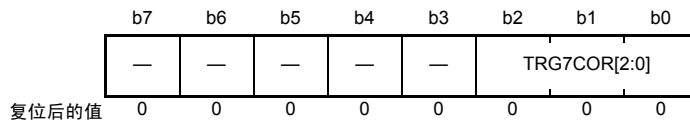
表 16.56 通过 TRG4COR[2:0] 位设定的中断减少次数

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 16.57。	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

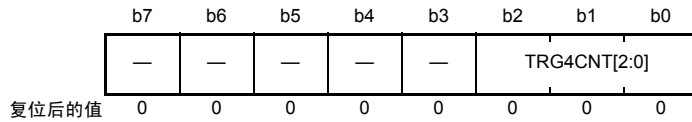
表 16.57 通过 TRG7COR[2:0] 位设定的中断减少次数

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。

## 16.2.36 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)

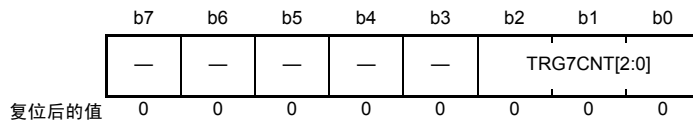
每当发生 TRG4AN 中断和 TRG4BN 中断时，从 TRG4COR[2:0] 位设定的值开始递减，直到计数器的值为“0”。在发生重加载时，TRG4AN 的中断和 TRG4BN 的中断有效。

[ 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

## 16.2.37 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、TDDR B)、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。

## 16.3 运行说明

### 16.3.1 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 16.5 所示。

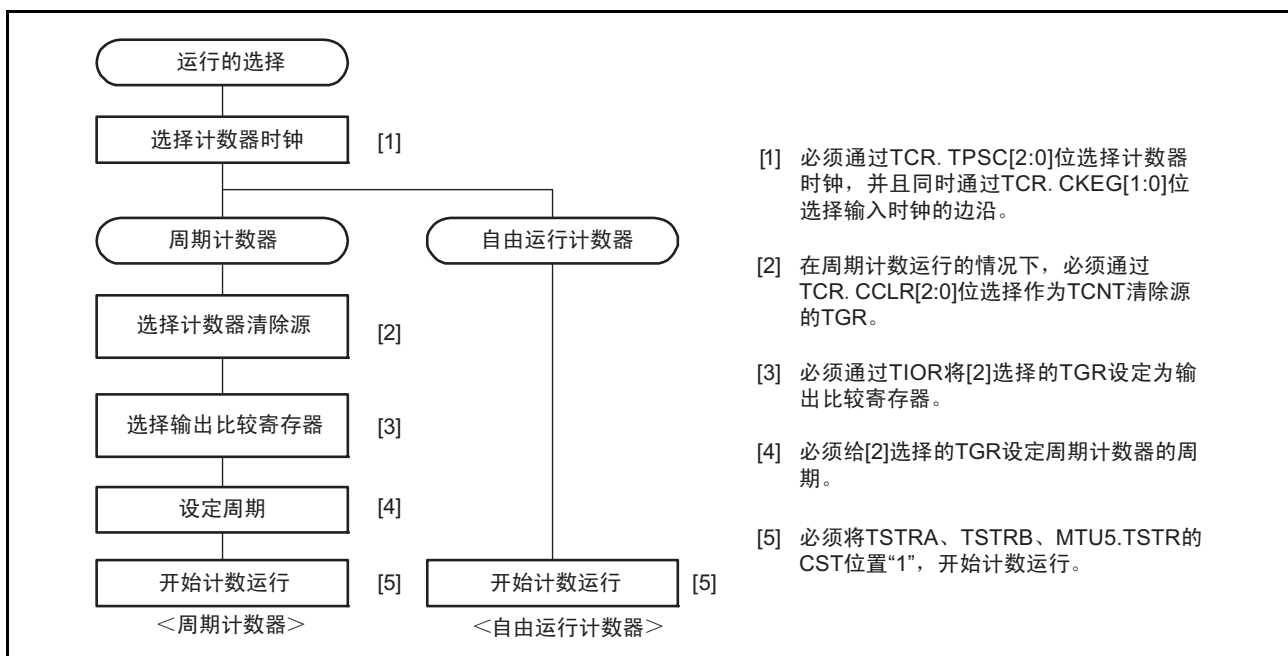


图 16.5 计数器运行的设定步骤例子

(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 16.6 所示。

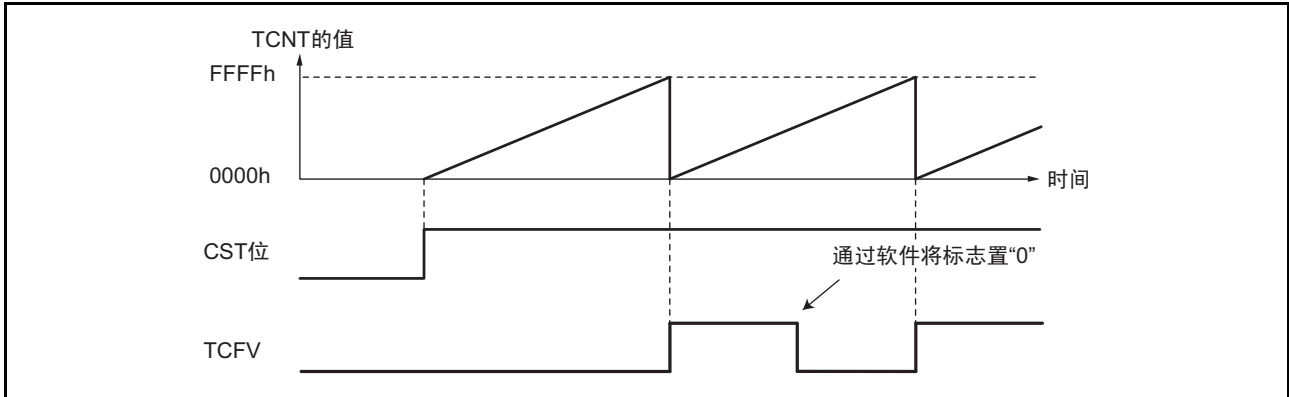


图 16.6 自由运行计数器的运行

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 16.7 所示。

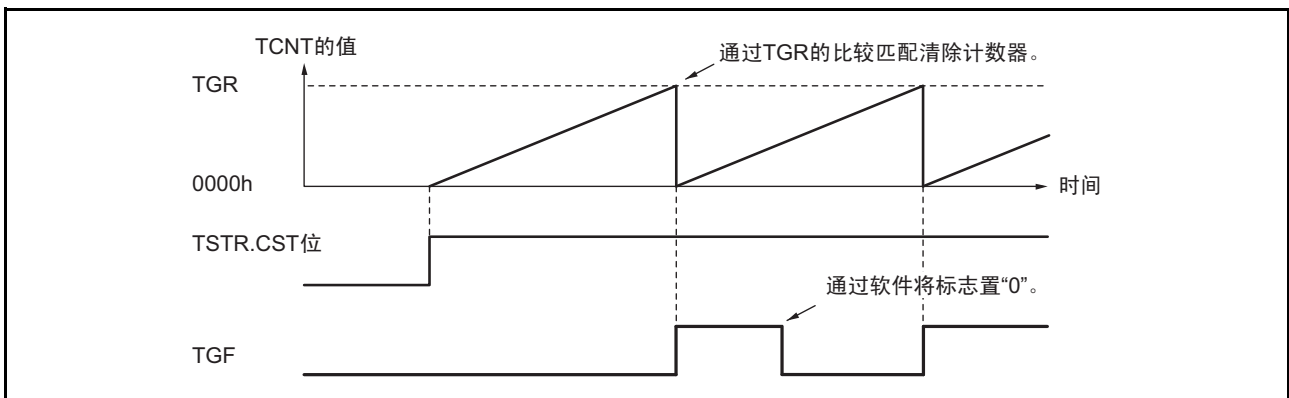


图 16.7 周期计数器的运行

(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 16.8 所示。

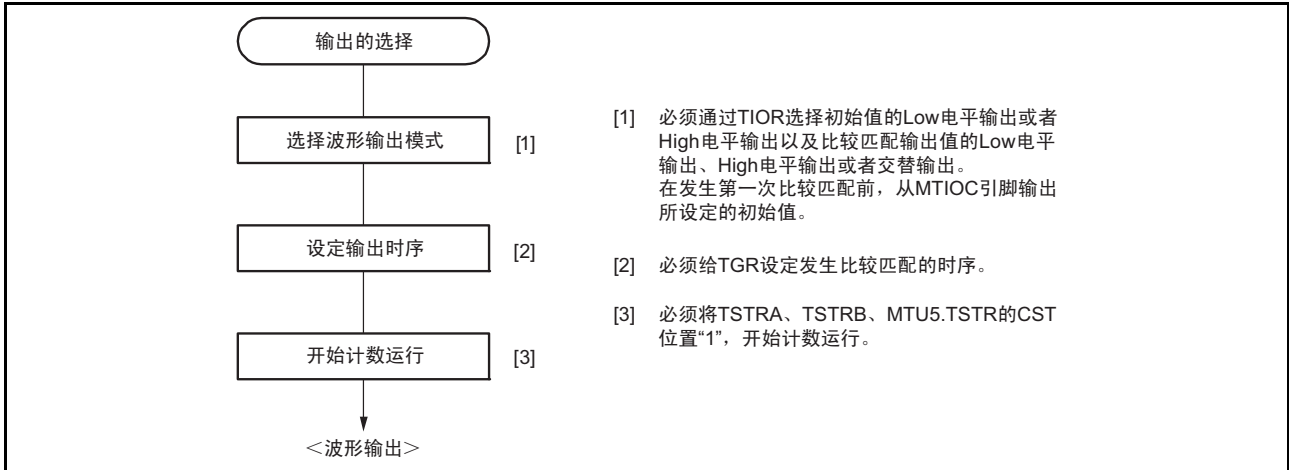


图 16.8 通过比较匹配进行波形输出的运行例子

(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 16.9 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。

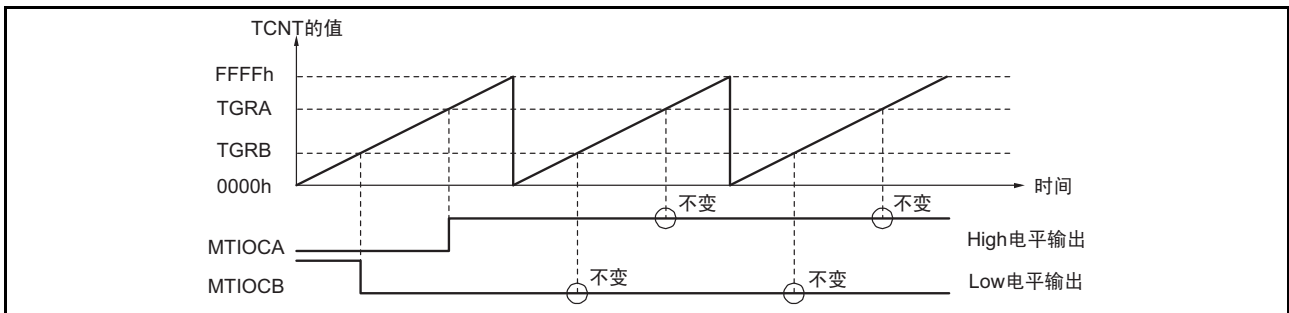


图 16.9 输出 Low 电平 /High 电平的运行例子

交替输出的运行例子如图 16.10 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。

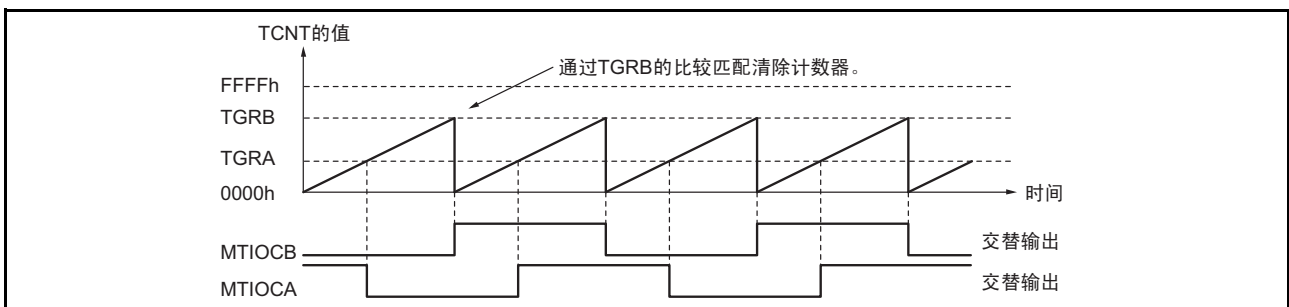


图 16.10 交替输出的运行例子

(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 ICLK/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 16.11 所示。

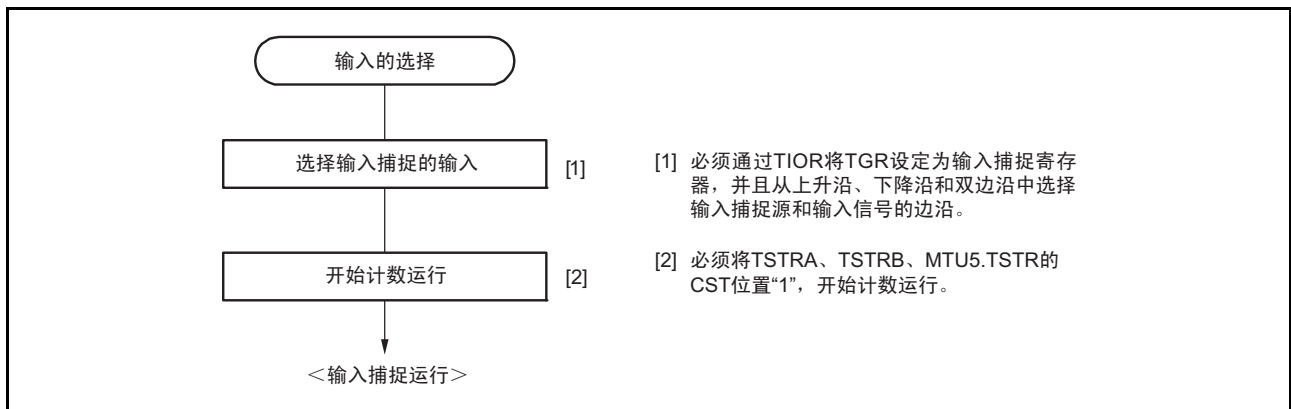


图 16.11 输入捕捉运行的设定例子

(b) 输入捕捉的运行例子

输入捕捉的运行例子如图 16.12 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。

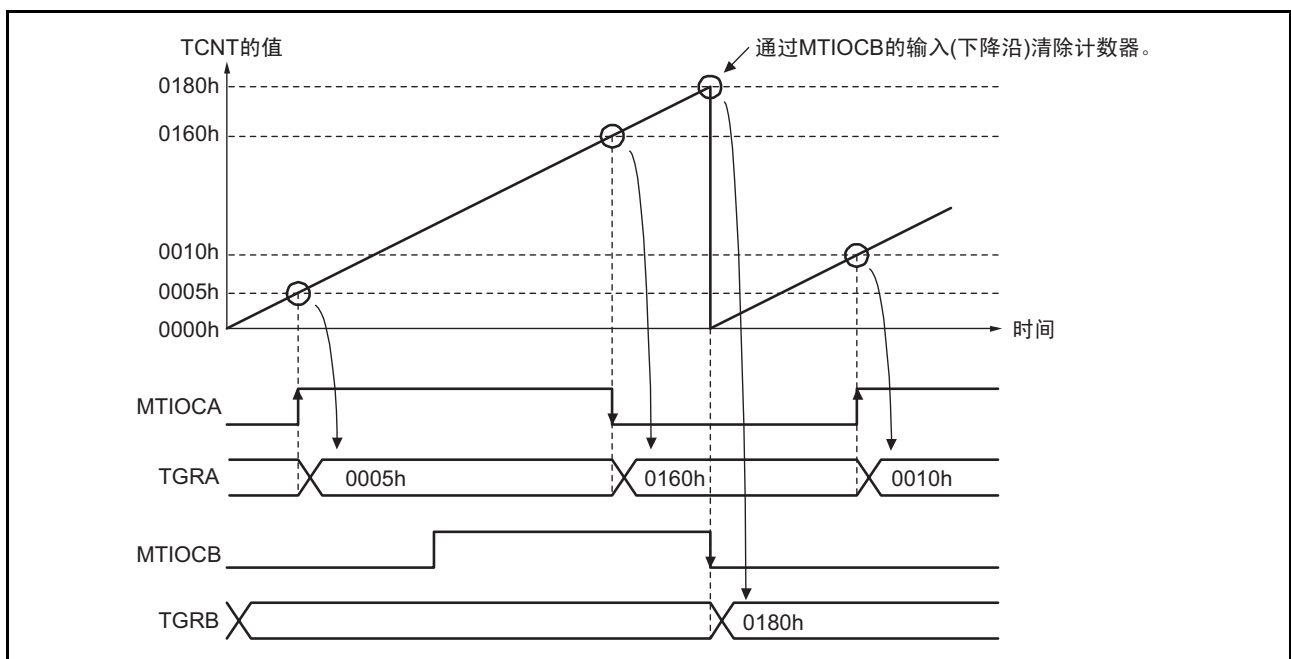


图 16.12 输入捕捉的运行例子

### 16.3.2 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 16.13 所示。

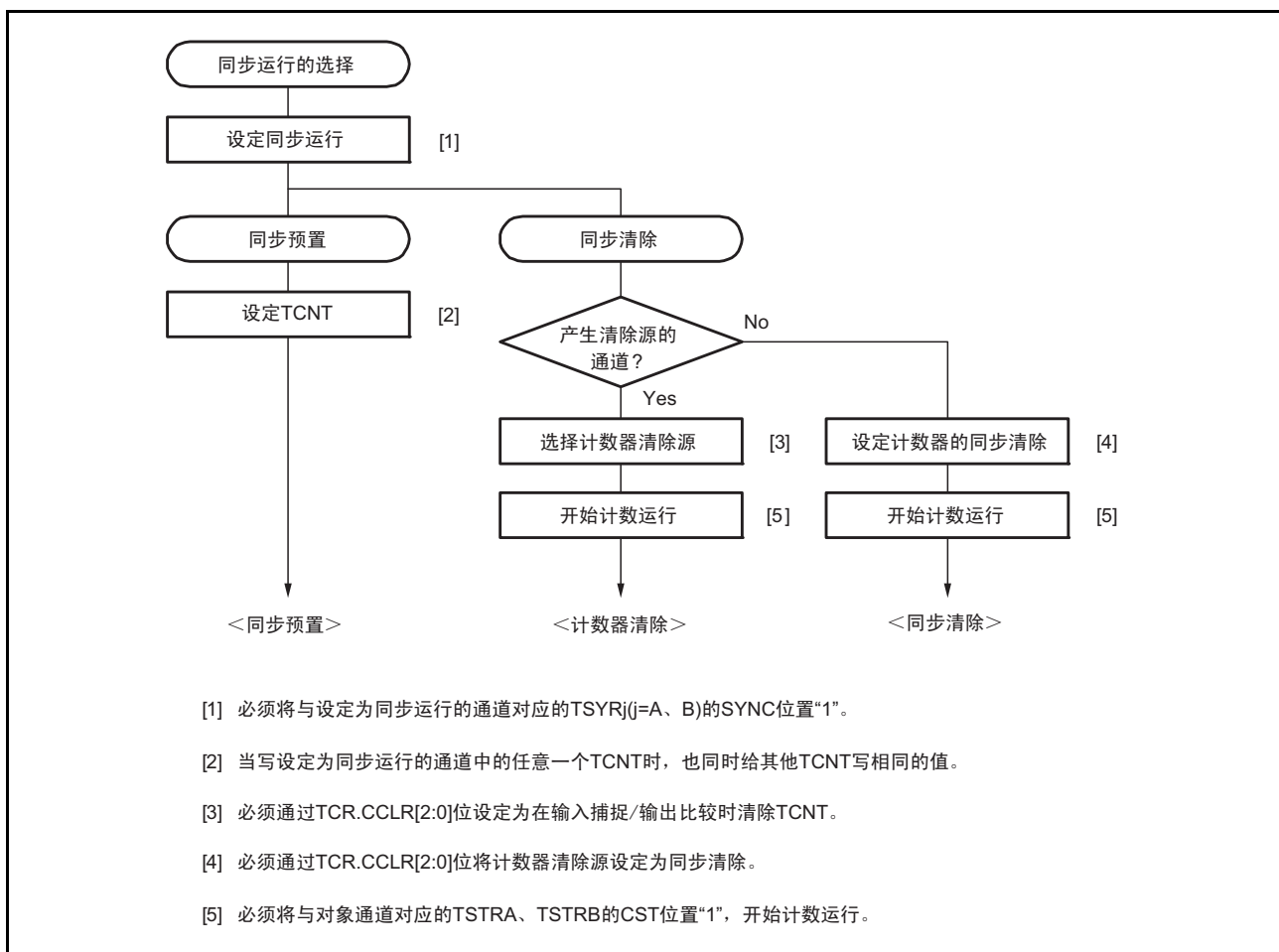


图 16.13 同步运行的设定步骤例子



## (2) 同步运行的例子

同步运行的例子如图 16.14 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 引脚输出三相 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“16.3.5 PWM 模式”。

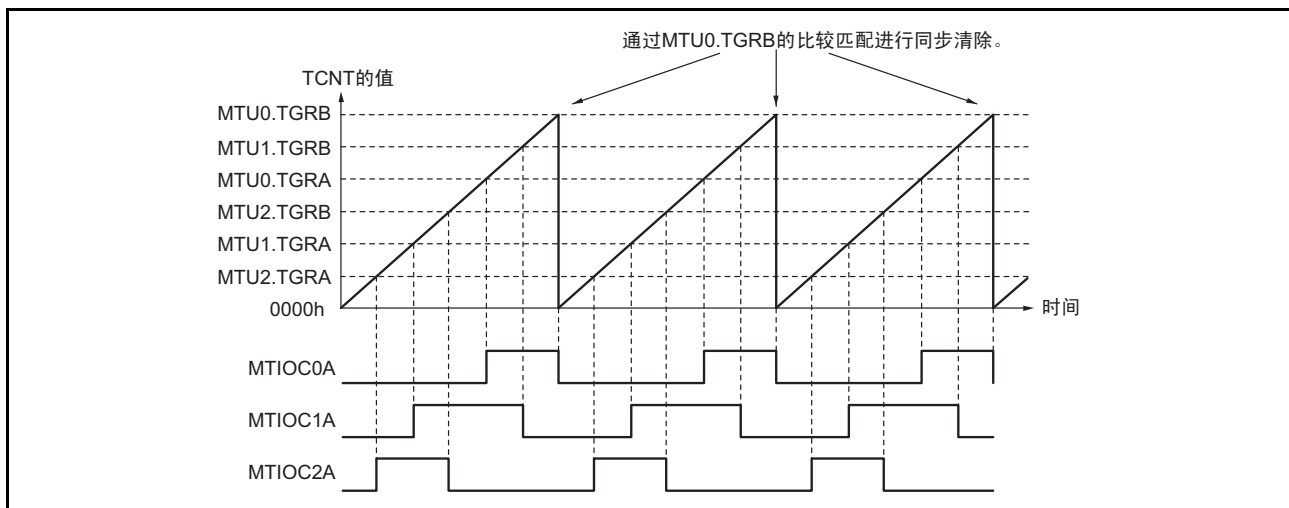


图 16.14 同步运行的例子

## 16.3.3 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 16.58 所示。

表 16.58 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 16.15 所示。

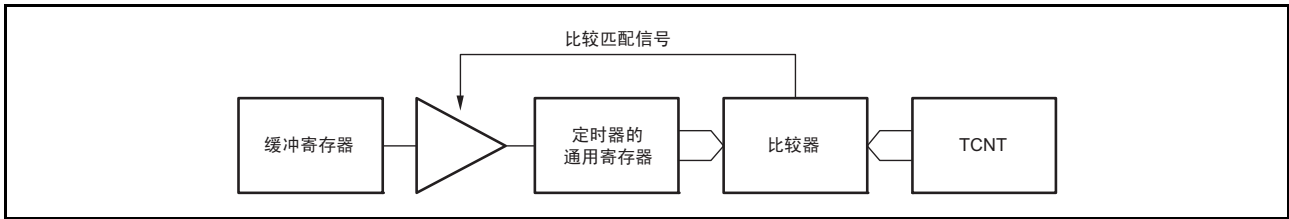


图 16.15 比较匹配的缓冲运行

- 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 16.16 所示。

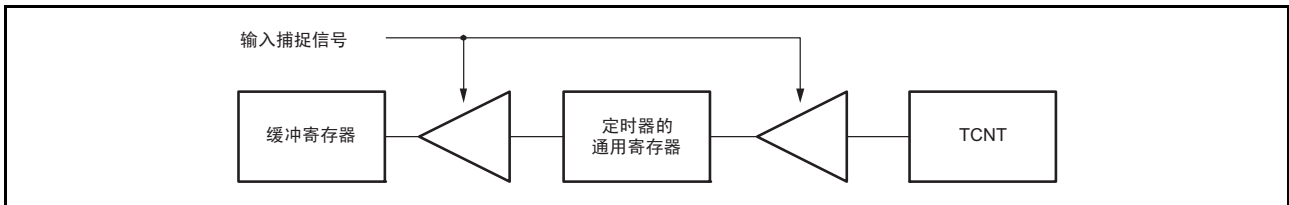


图 16.16 输入捕捉的缓冲运行

(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 16.17 所示。

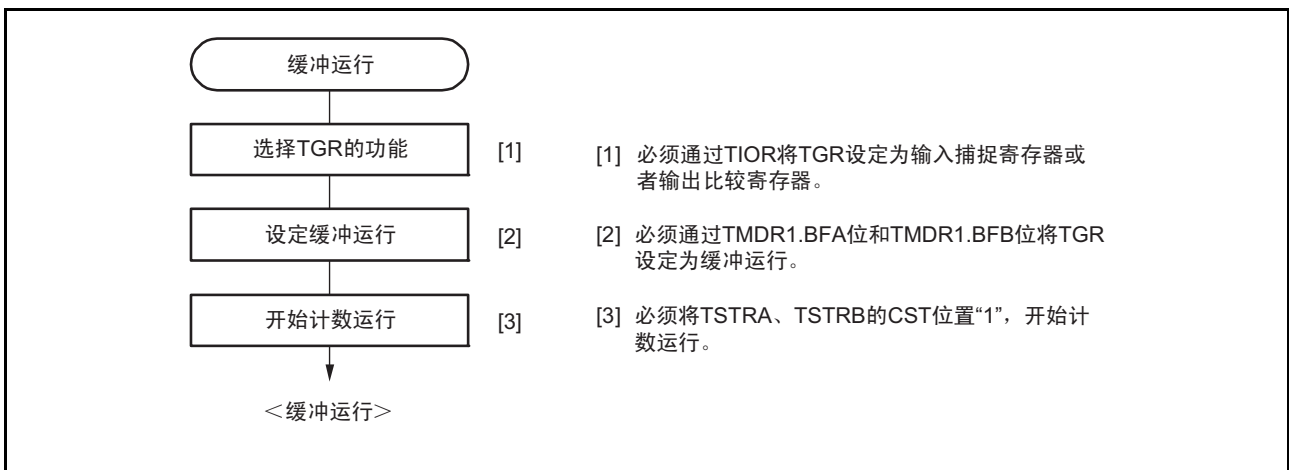


图 16.17 缓冲运行的设定步骤例子

(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 16.18 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“16.3.5 PWM 模式”。

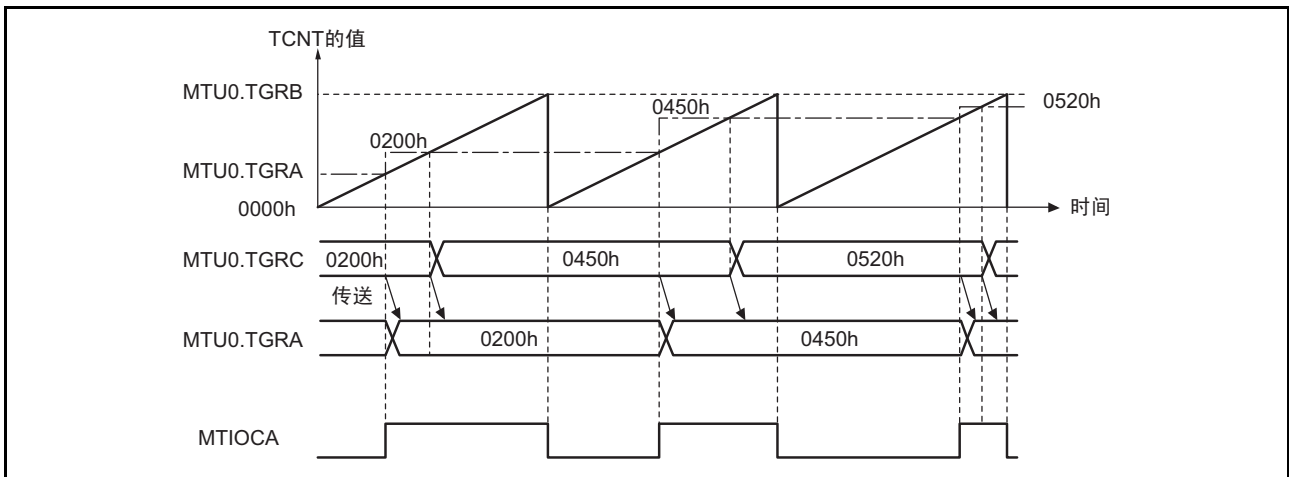


图 16.18 缓冲运行的例子 (1)

(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 16.19 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。

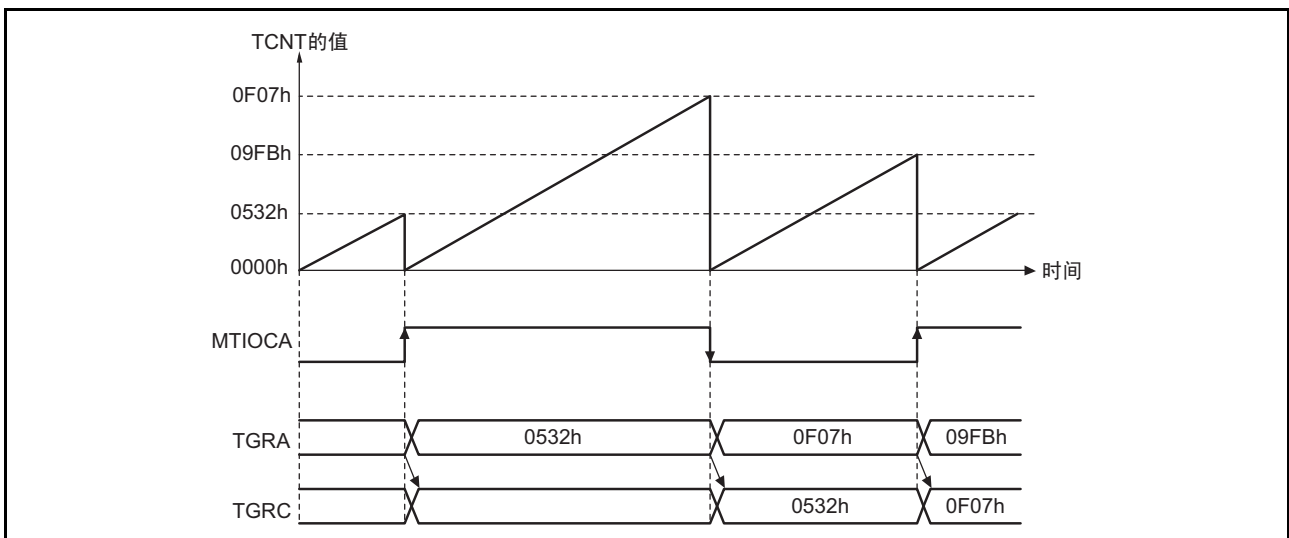


图 16.19 缓冲运行的例子 (2)

## (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 16.20 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”。

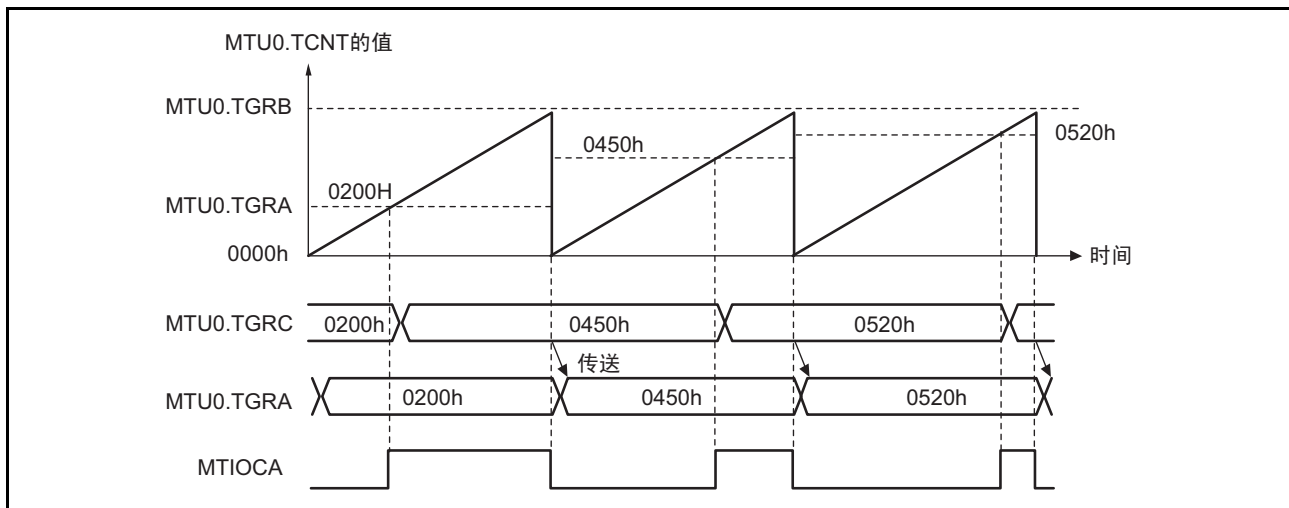


图 16.20 选择在清除 MTU0.TCNT 时进行从 MTU0.TGRC 到 MTU0.TGRA 的缓冲传送的运行例子

### 16.3.4 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 16.59 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 16.59 级联的组合

组合	高 16 位	低 16 位
通道 1 和通道 2	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作为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“16.6.21 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 16.60 所示。

表 16.60 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 16.21 所示。



图 16.21 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 16.22 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。

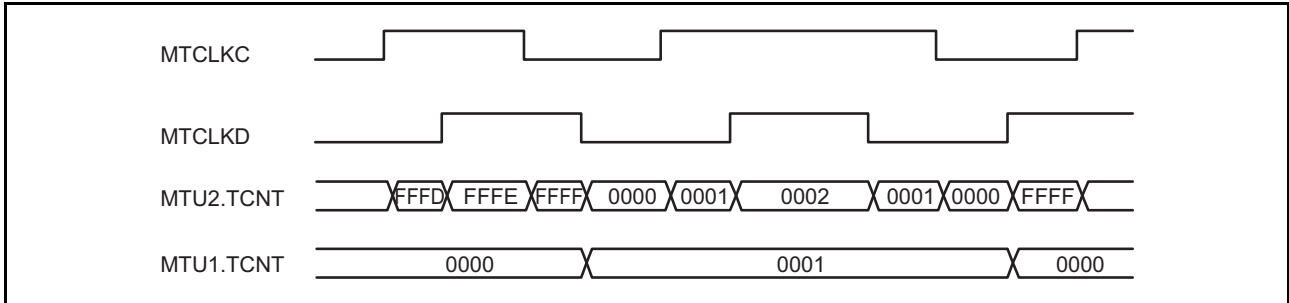


图 16.22 级联运行的例子 (a)

(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 16.23 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。

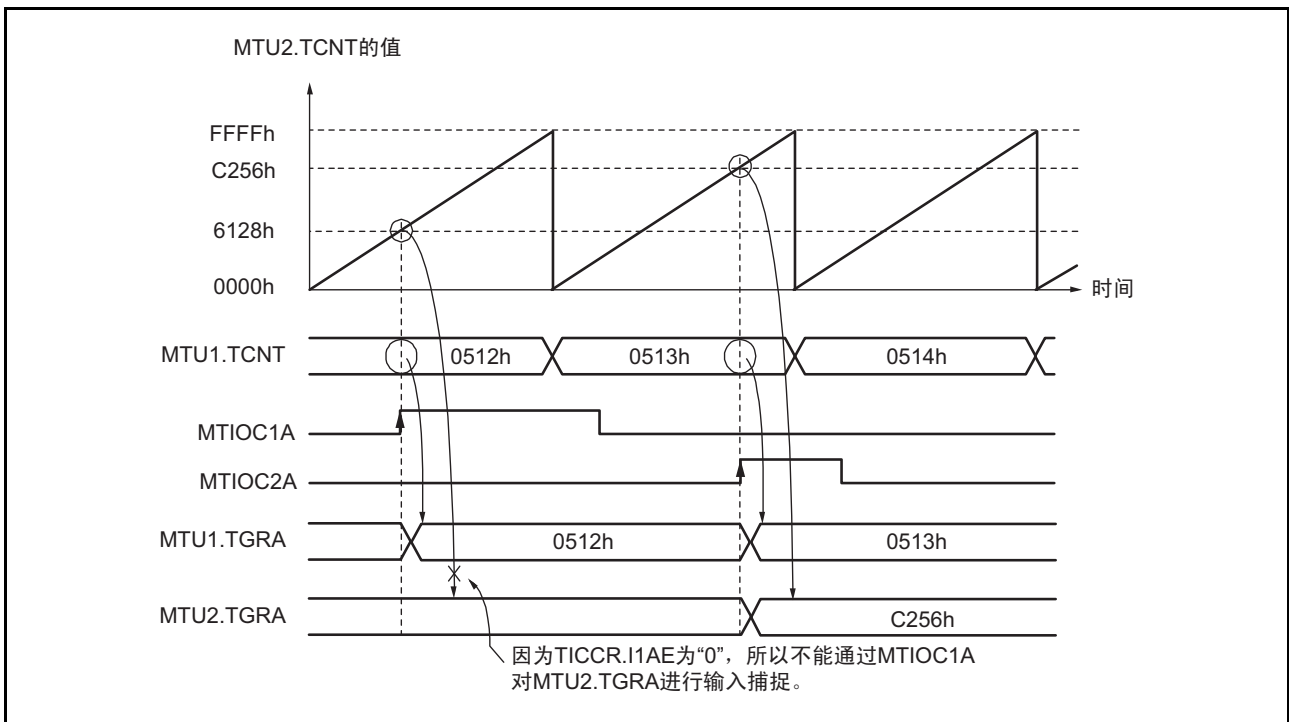


图 16.23 级联运行的例子 (b)

(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 16.24 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。

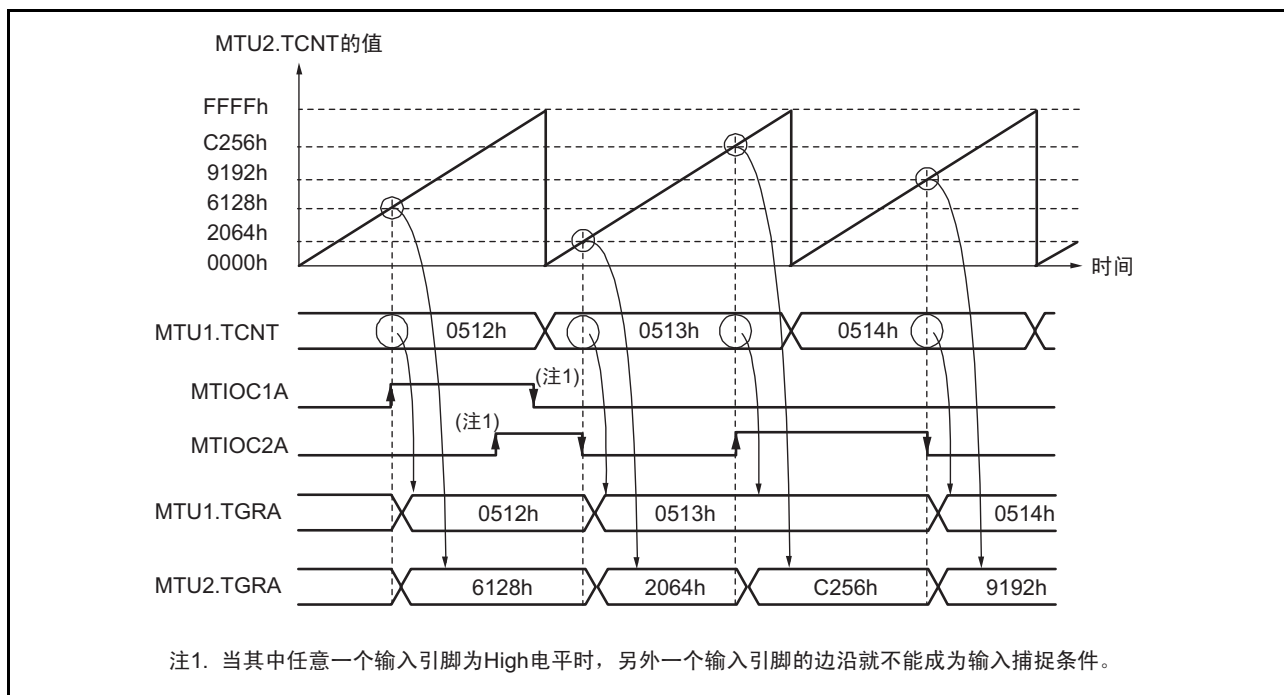


图 16.24 级联运行的例子 (c)

## (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 16.25 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。

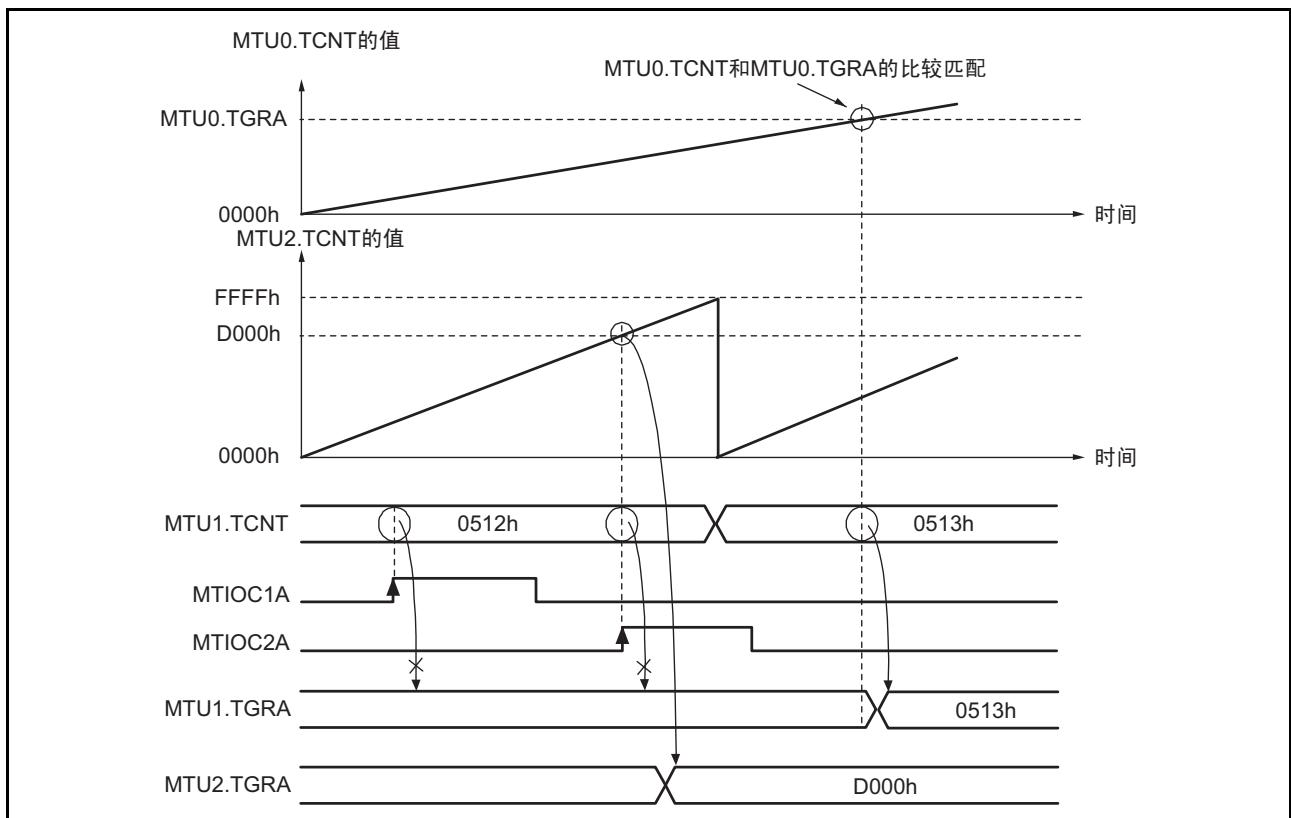


图 16.25 级联运行的例子 (d)



### 16.3.5 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电平可选择 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 12 相的 PWM 输出。

#### (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 16.61 所示。

表 16.61 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 16.26 所示。

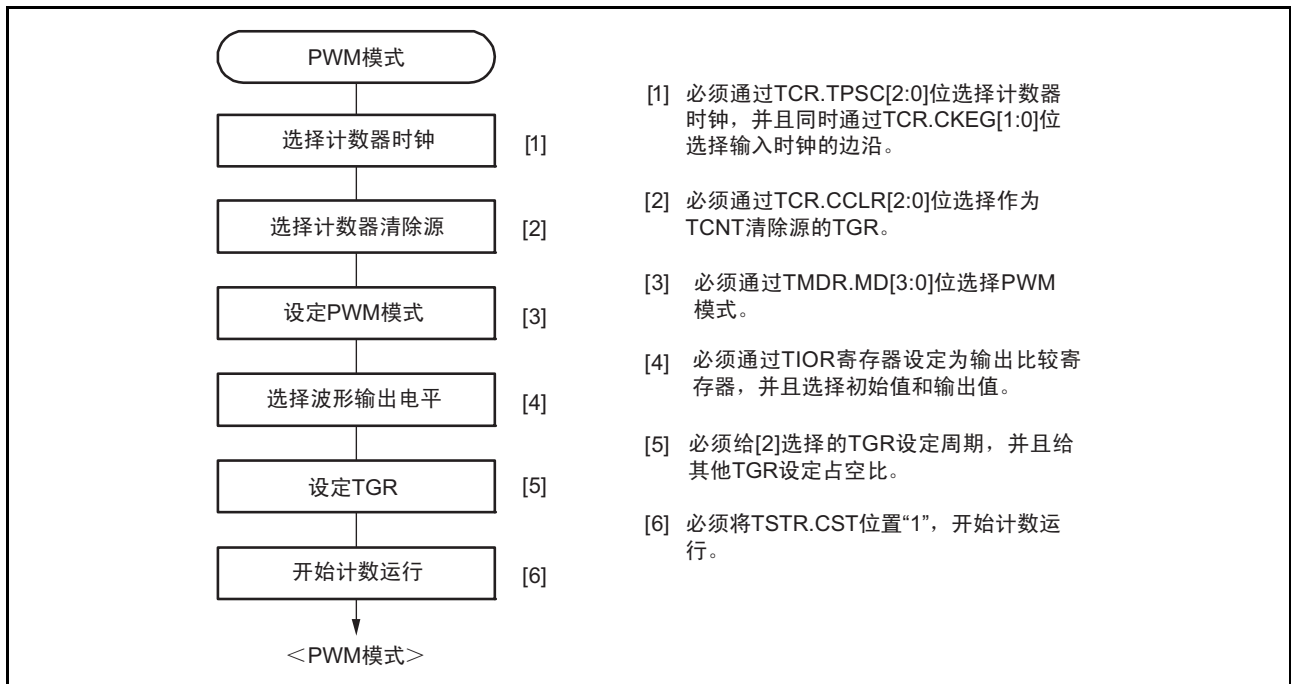


图 16.26 PWM 模式的设定步骤例子

(2) PWM 模式的运行例子

PWM 模式 1 的运行例子如图 16.27 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。

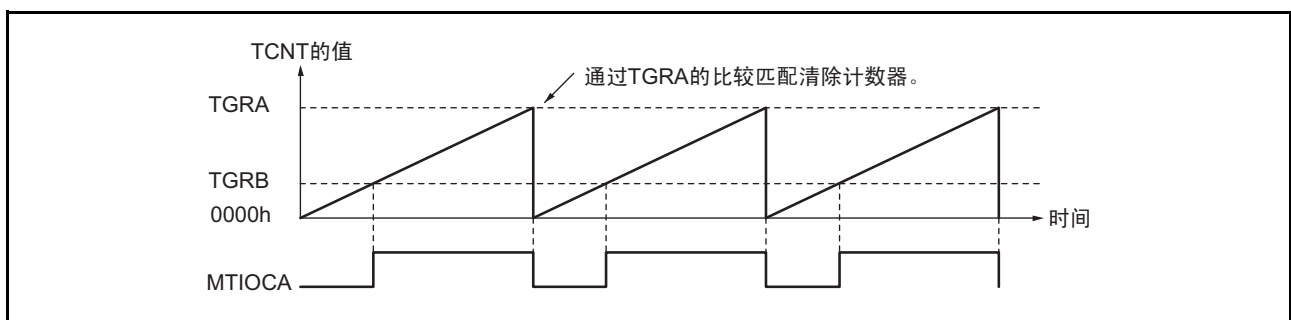


图 16.27 PWM 模式 1 的运行例子

PWM 模式 2 的运行例子如图 16.28 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。

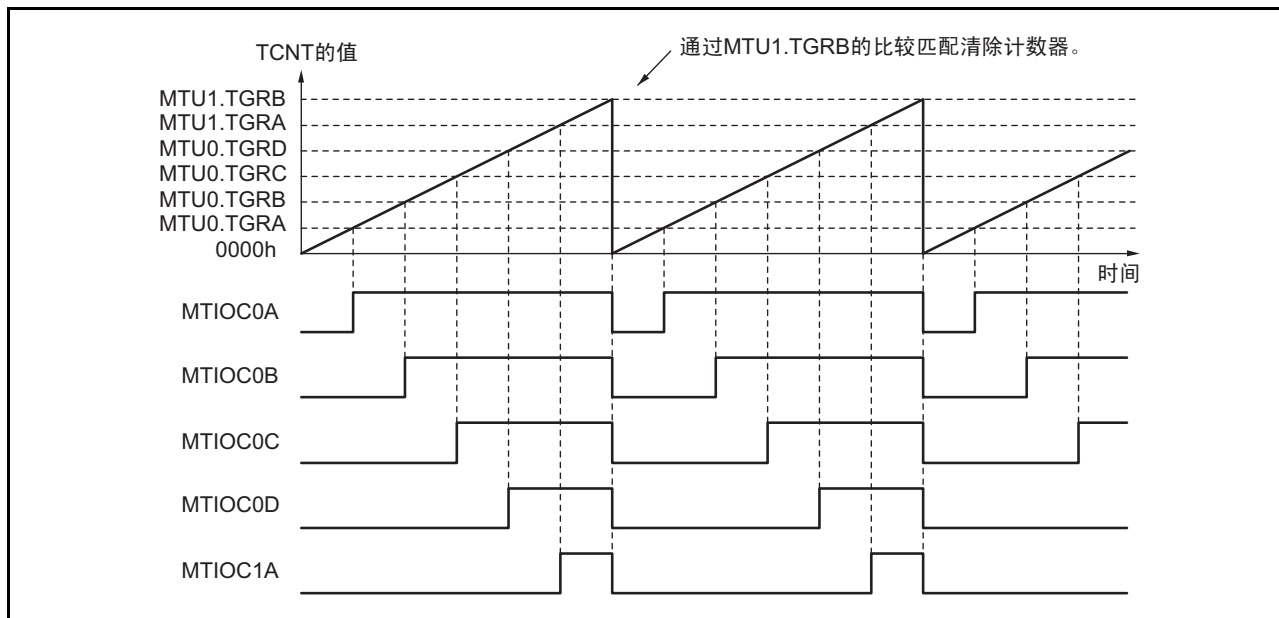


图 16.28 PWM 模式 2 的运行例子

在 PWM 模式中，占空比为 0% 和 100% 的 PWM 波形输出例子如图 16.29 所示。

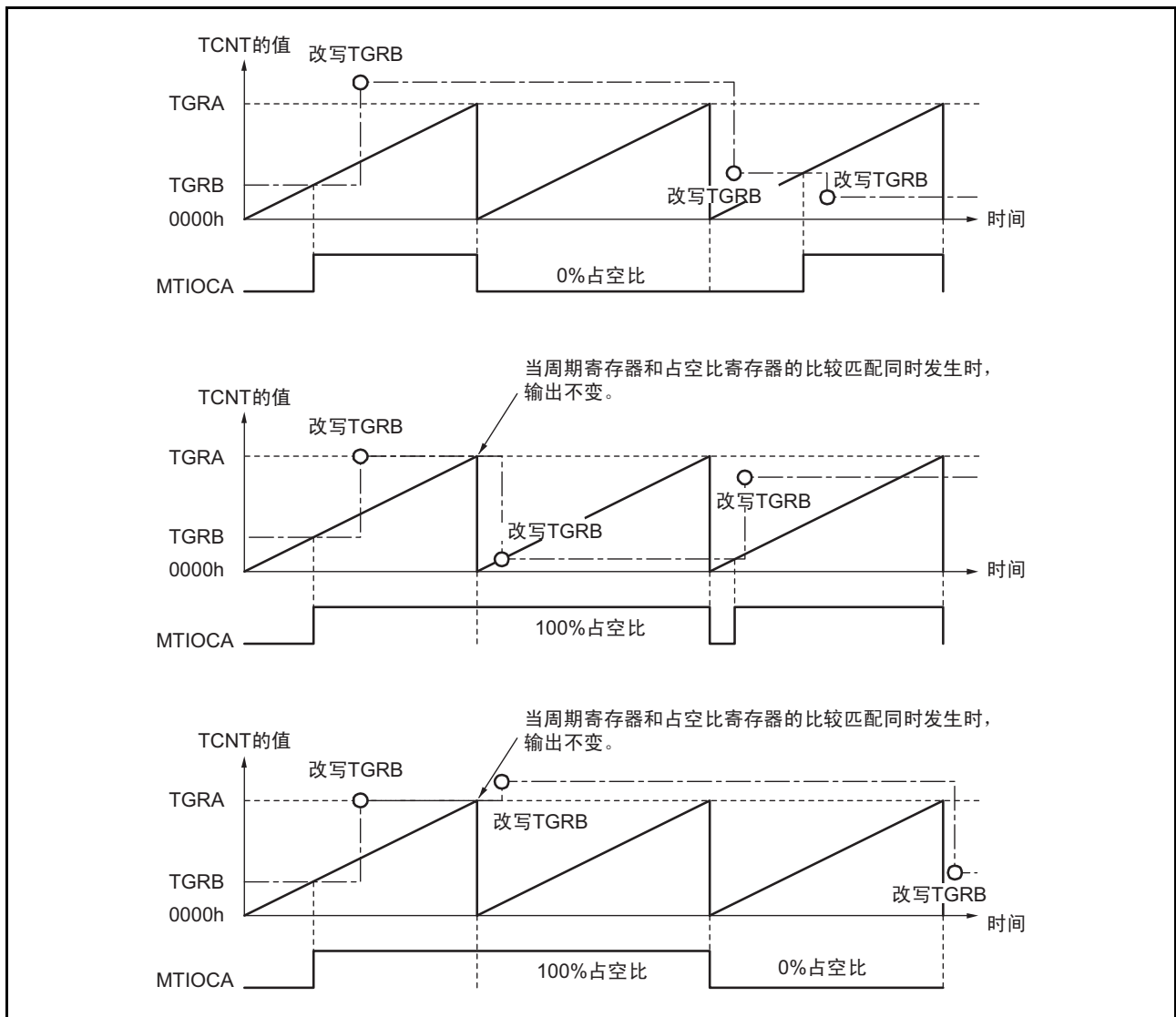


图 16.29 PWM 模式的运行例子 (0% 占空比和 100% 占空比的 PWM 波形输出例子)

### 16.3.6 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、TIOB 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 16.62 所示。

表 16.62 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 16.30 所示。

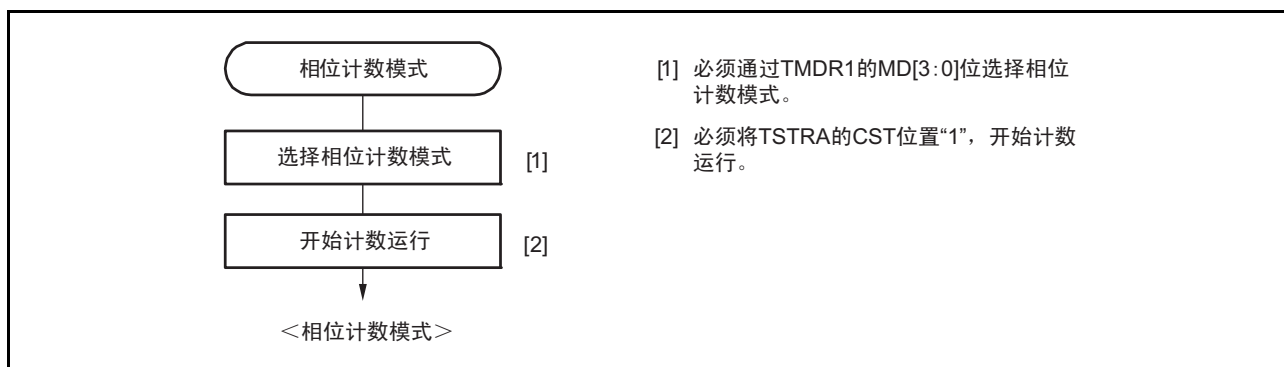


图 16.30 相位计数模式的设定步骤例子

(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 16.31 所示，TCNT 的递增 / 递减计数条件如表 16.63 所示。

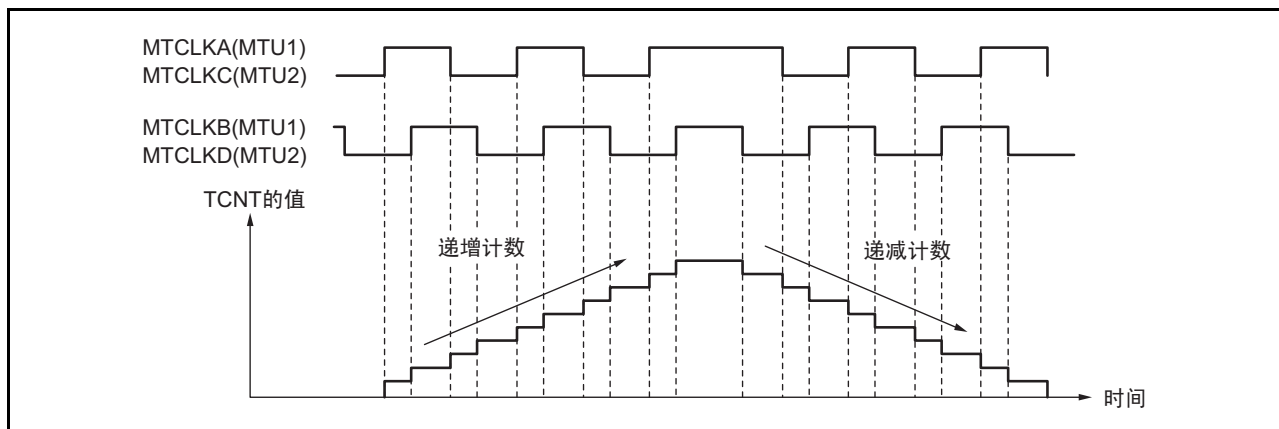


图 16.31 相位计数模式 1 的运行例子

表 16.63 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 16.32 所示，TCNT 的递增 / 递减计数条件如表 16.64 所示。

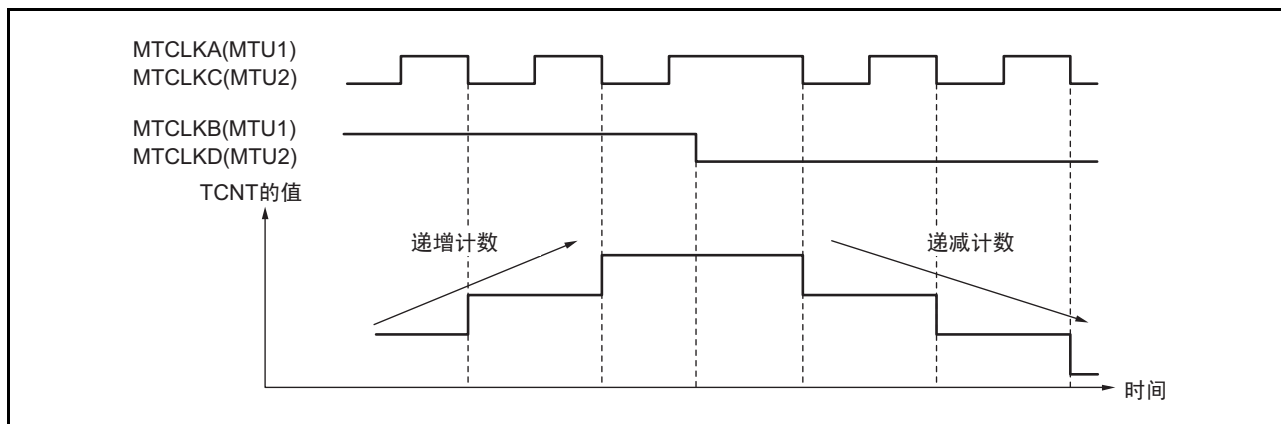


图 16.32 相位计数模式 2 的运行例子

表 16.64 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 16.33 所示，TCNT 的递增 / 递减计数条件如表 16.65 所示。

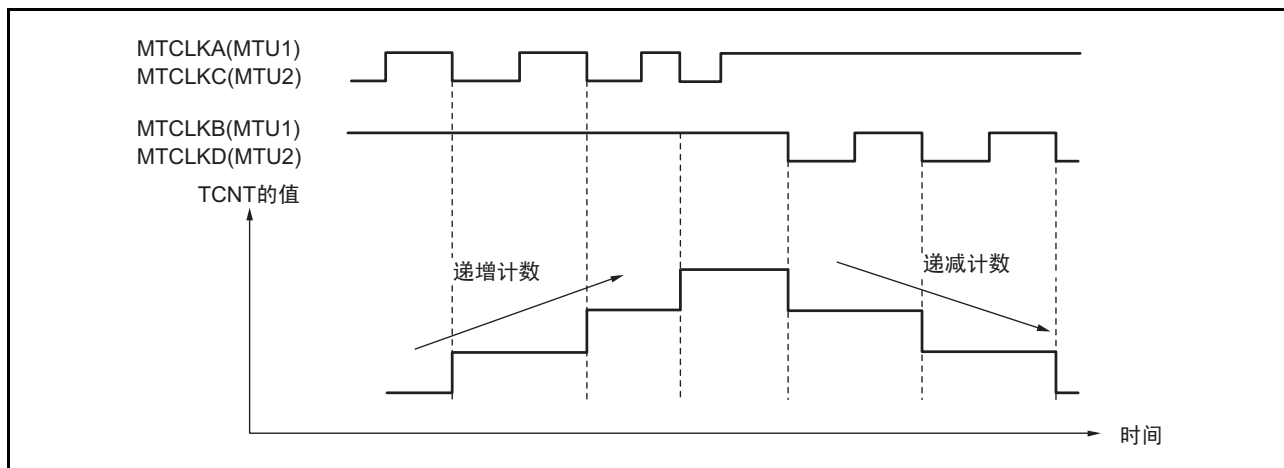


图 16.33 相位计数模式 3 的运行例子

表 16.65 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 16.34 所示，TCNT 的递增 / 递减计数条件如表 16.66 所示。

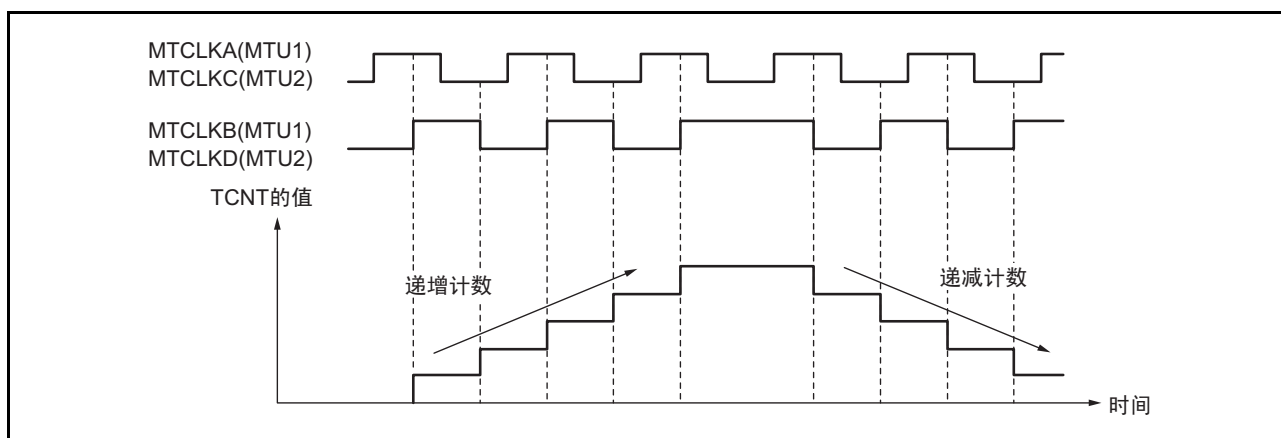


图 16.34 相位计数模式 4 的运行例子

表 16.66 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 16.35 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。

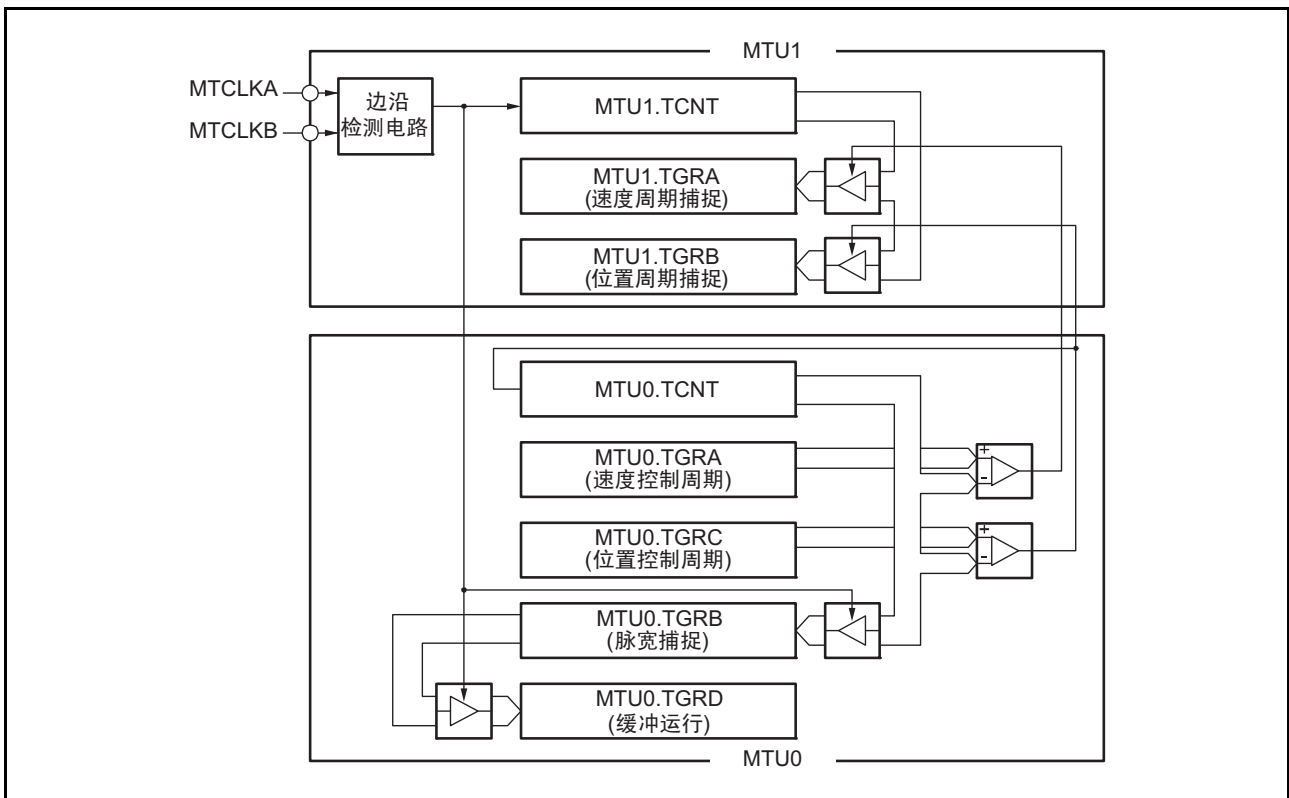


图 16.35 相位计数模式的应用例子

### 16.3.7 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）各三相共计 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 16.67、使用的寄存器的设定如表 16.68 所示。

表 16.67 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 16.68 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 16.36 所示。

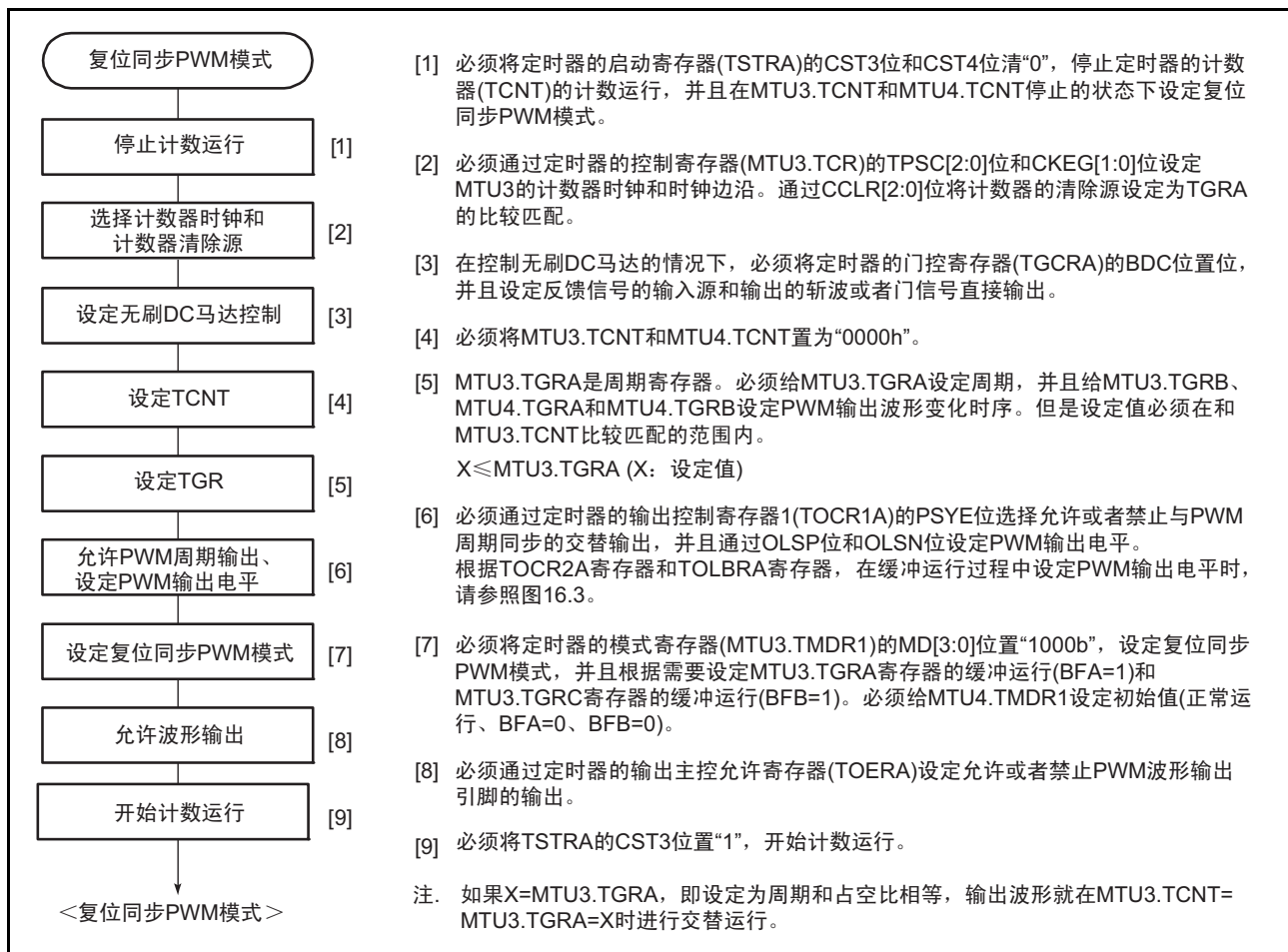


图 16.36 复位同步 PWM 模式的设定步骤例子

## (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 16.37 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。

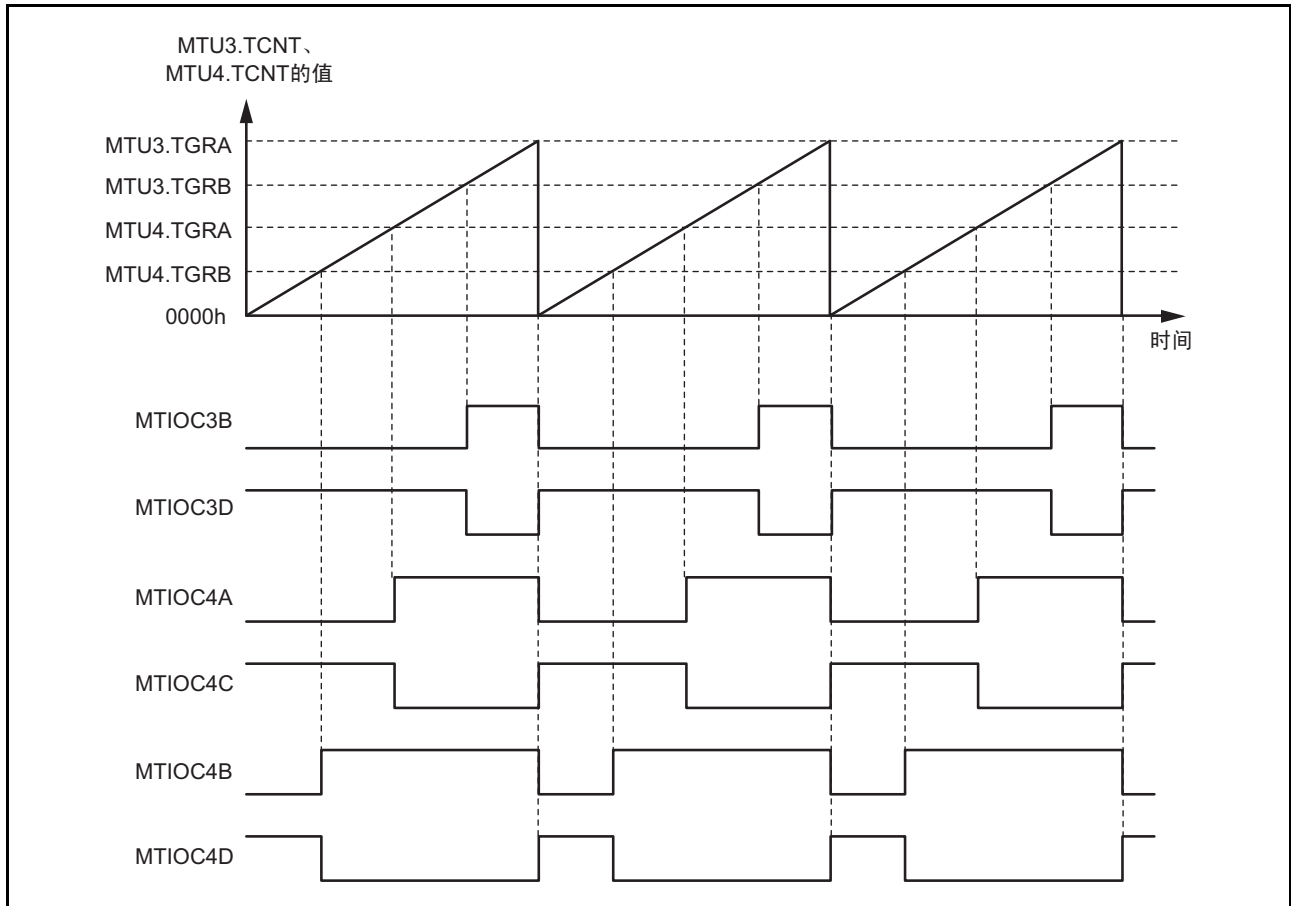


图 16.37 复位同步 PWM 模式的运行例子 (MTU3 和 MTU4、TOCR1A 的 OLSN=1、OLSP=1 的情况)

### 16.3.8 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 波形进行各三相共计 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 16.69、使用的寄存器的设定如表 16.70 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 16.69 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 16.70 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 16.71 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。

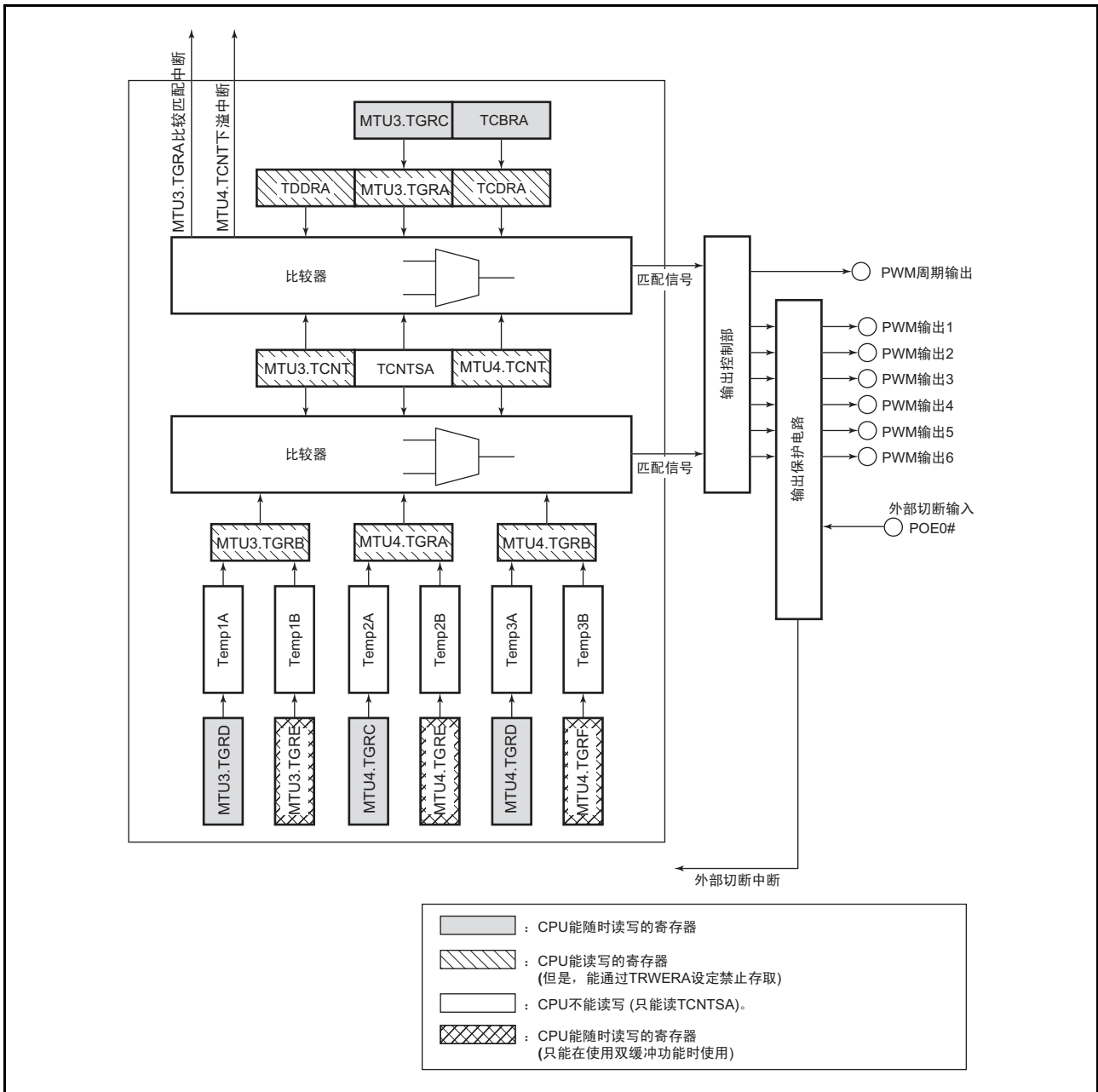


图 16.38 互补 PWM 模式中的 MTU3 和 MTU4 的框图

(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 16.39 所示。



图 16.39 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 16.40、互补 PWM 模式的运行例子如图 16.41 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。

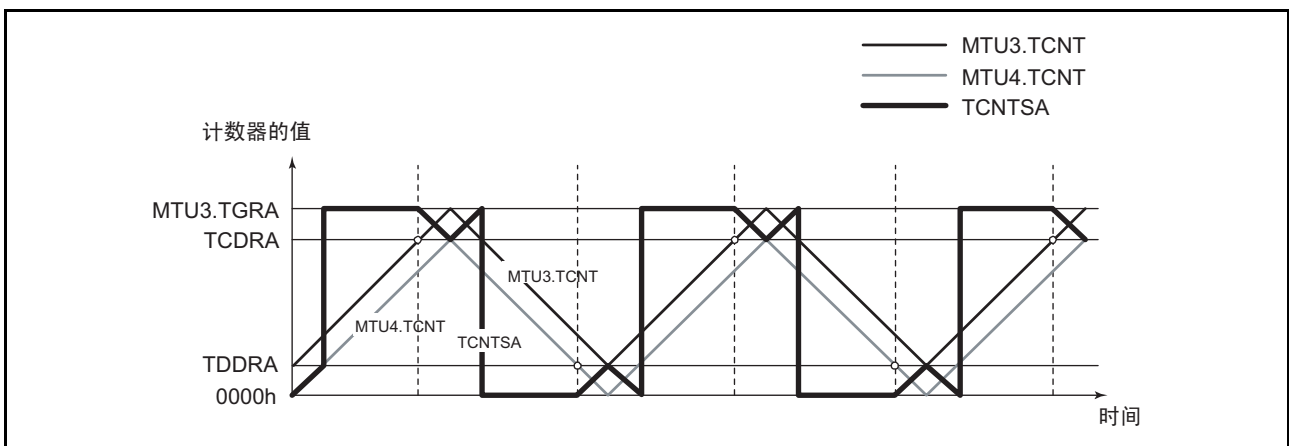


图 16.40 互补 PWM 模式的计数器运行

(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 16.41 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“16.3.8(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 16.41 所示。

在不向暂存器传送数据的 Tb (图 16.41 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。

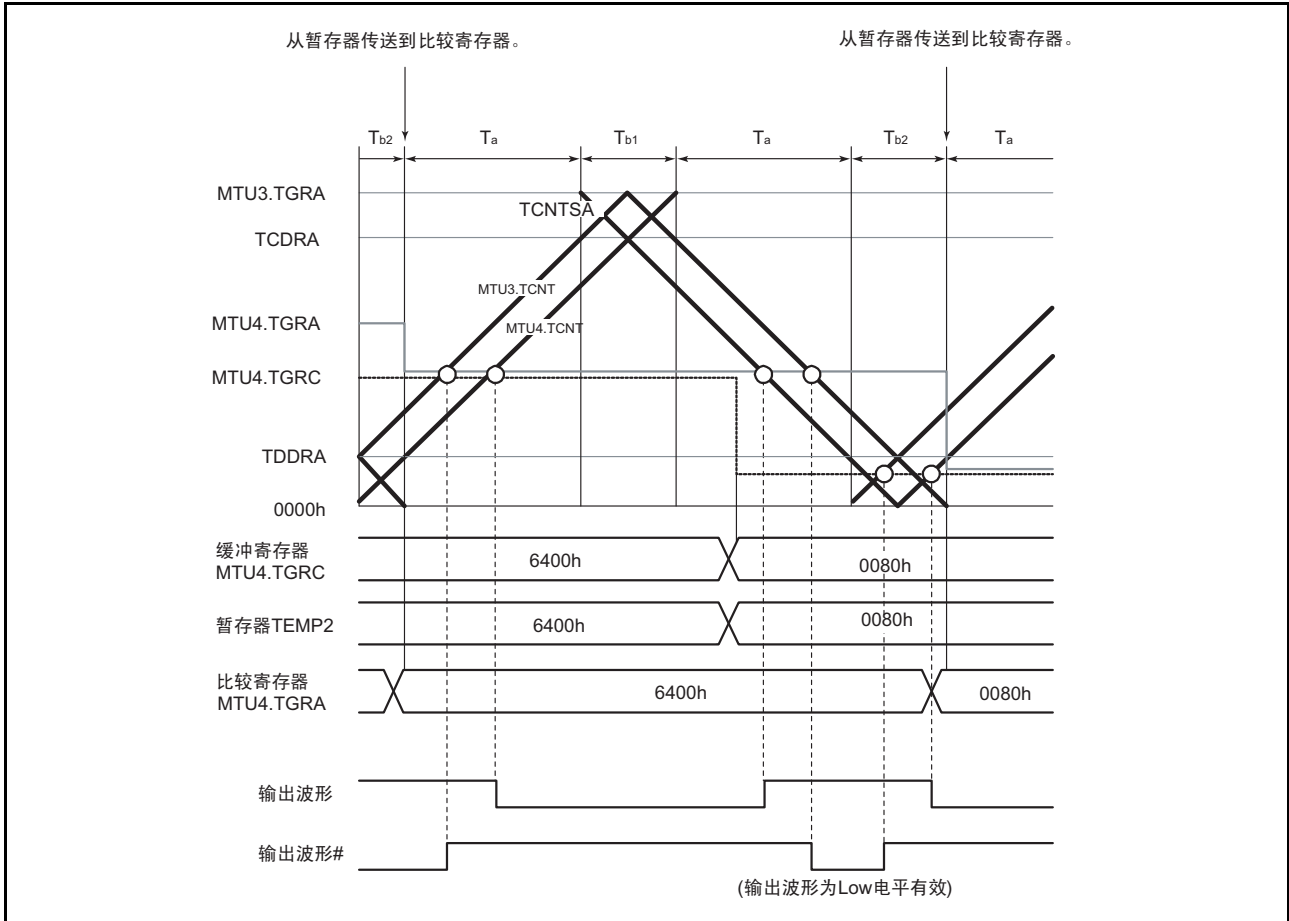


图 16.41 互补 PWM 模式的运行例子 (MTU3 和 MTU4)

## (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 16.72 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 相输出中的三相正相和三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 16.42 所示。

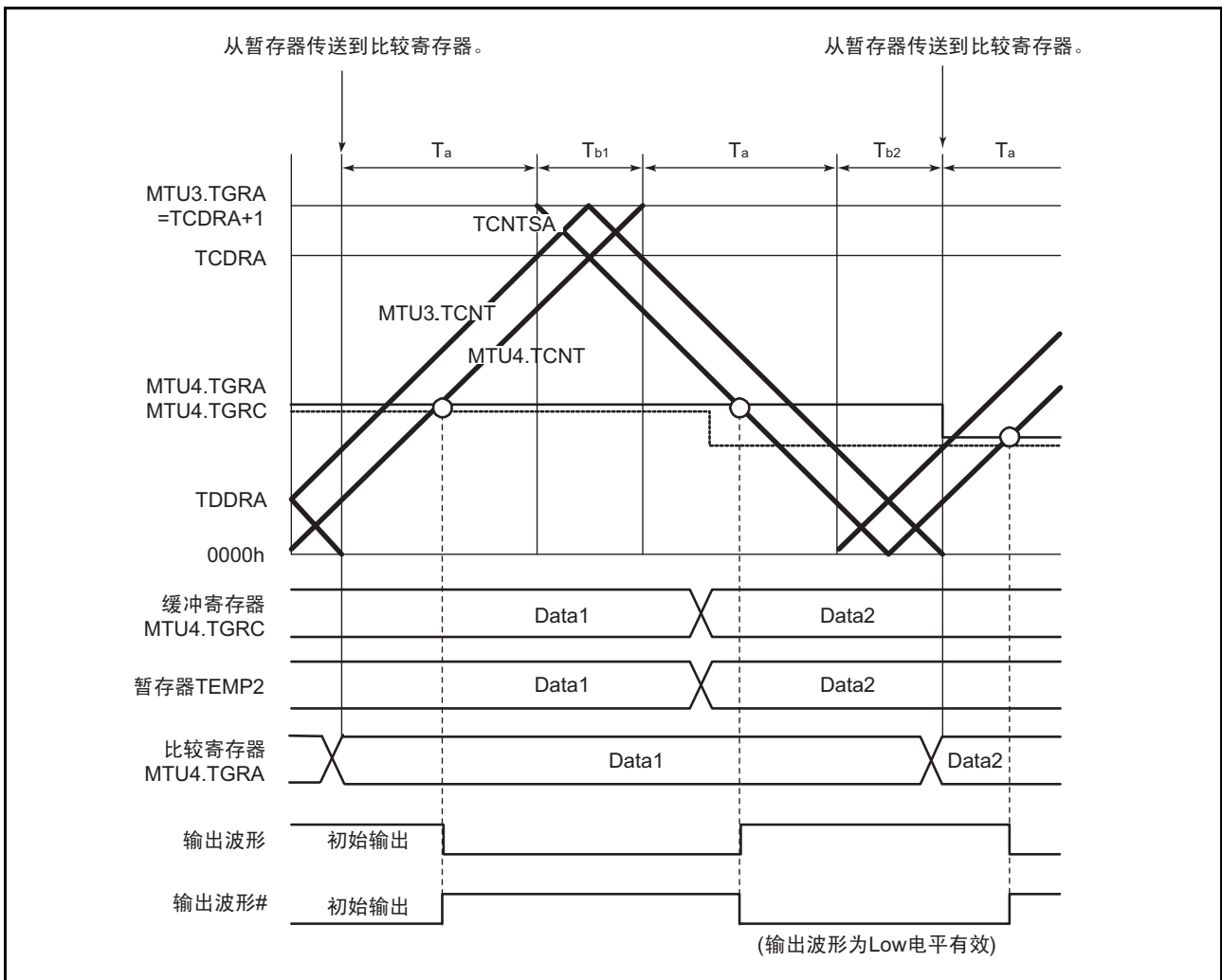


图 16.42 不生成死区时间的运行例子 (MTU3 和 MTU4)

### (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 16.43 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”。

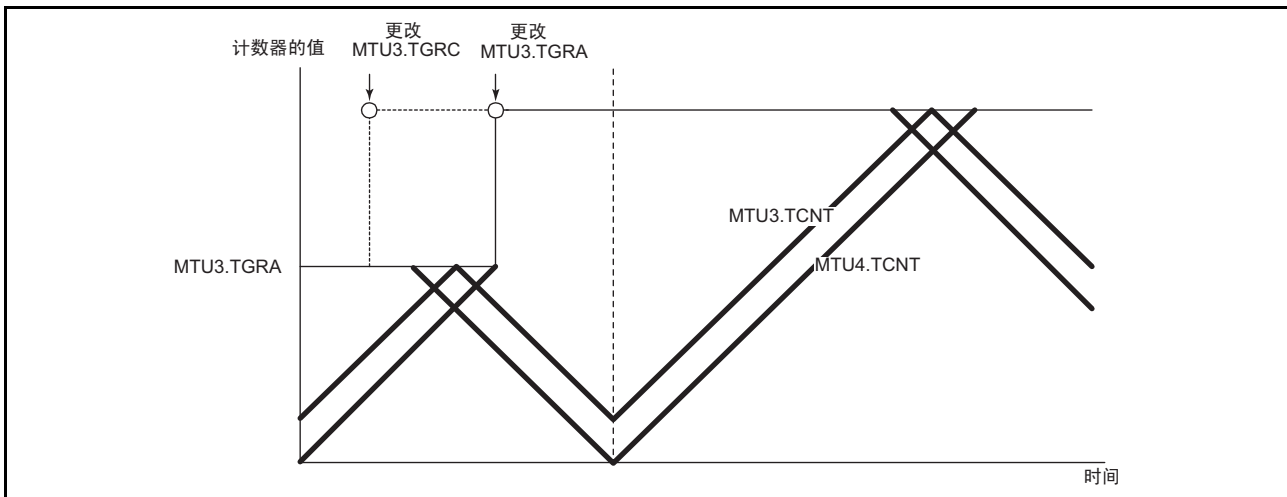


图 16.43 PWM 周期的更改例子 (MTU3 和 MTU4)

### (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 16.44 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“16.3.8(2)(s) 互补 PWM 模式的双缓冲功能”。



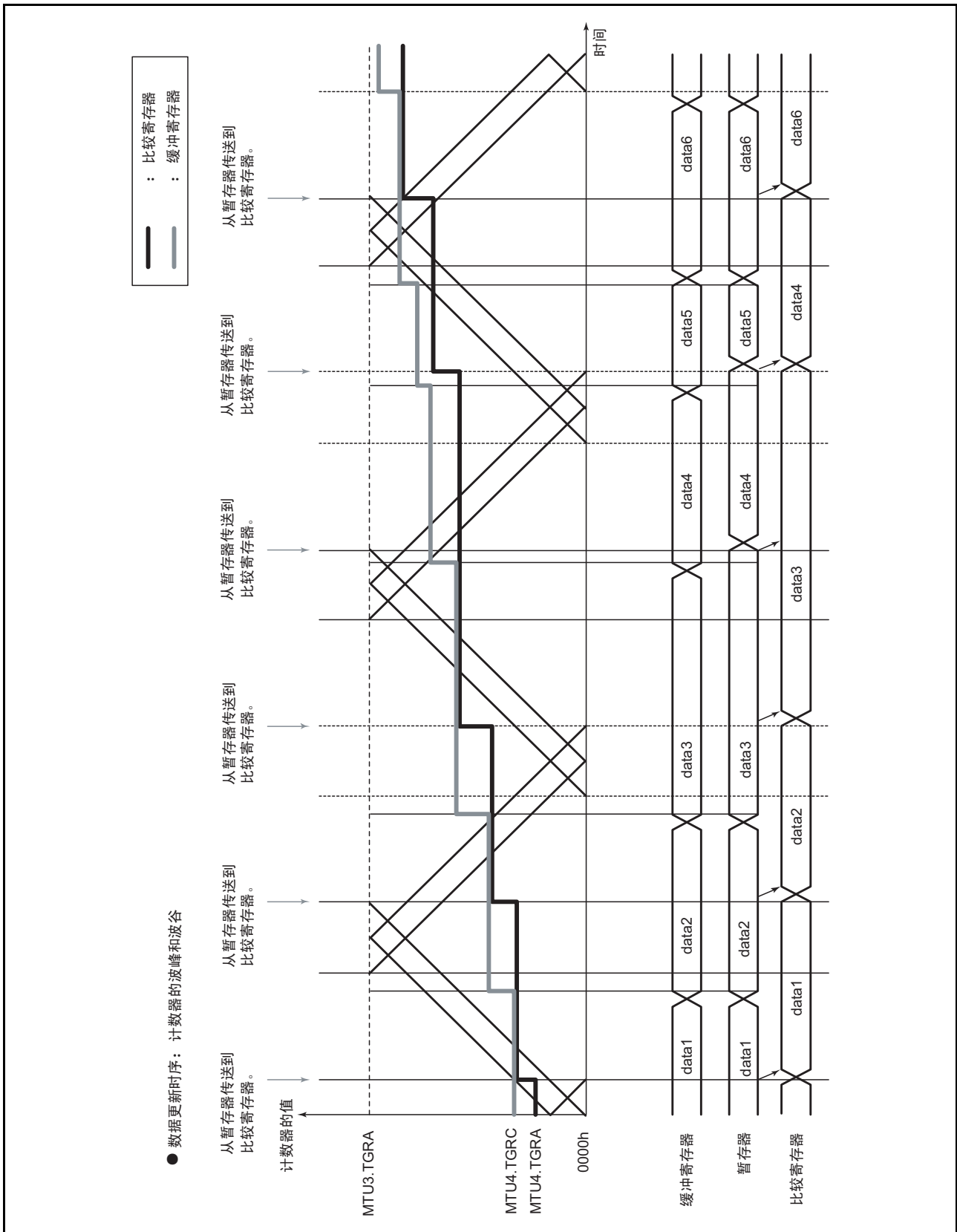


图 16.44 互补 PWM 模式的数据更新例子 (MTU3 和 MTU4)

(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 16.45 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 16.46 所示。

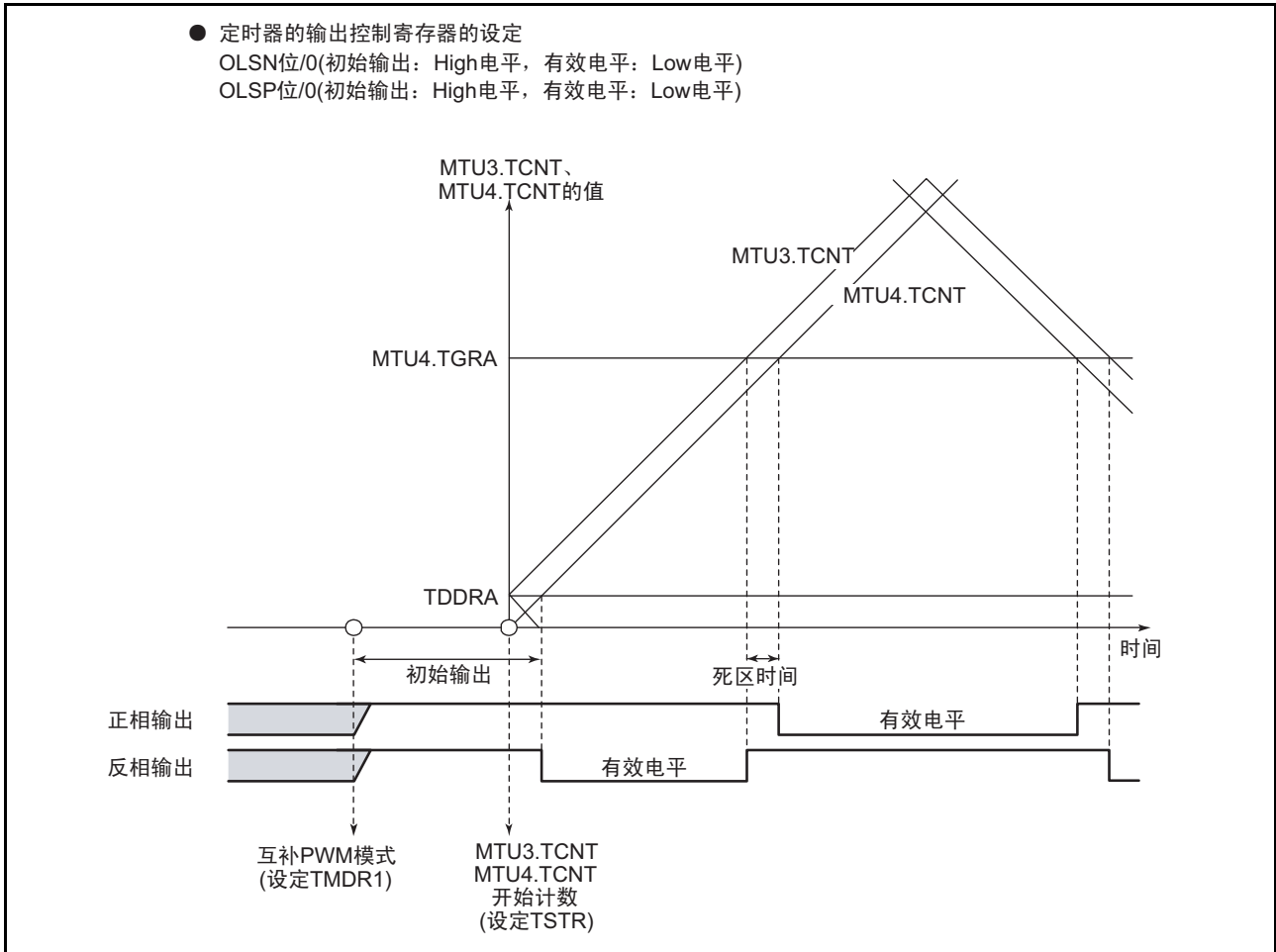


图 16.45 互补 PWM 模式的初始输出例子 (MTU3 和 MTU4) (1)

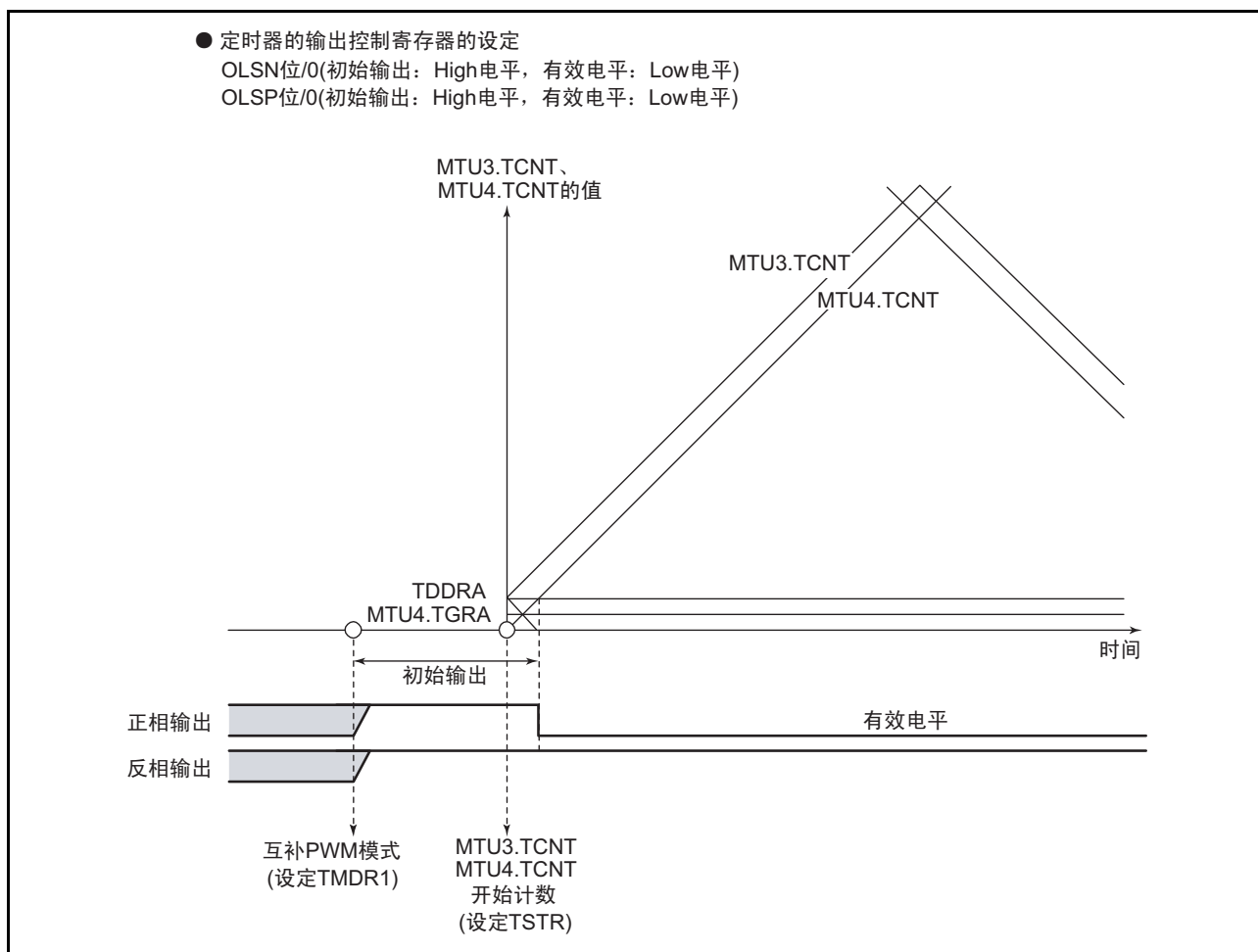


图 16.46 互补 PWM 模式的初始输出例子 (MTU3 和 MTU4) (2)

(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 波形进行三相输出，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 16.47 ~ 图 16.49 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 16.47 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 16.48 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 16.49 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。

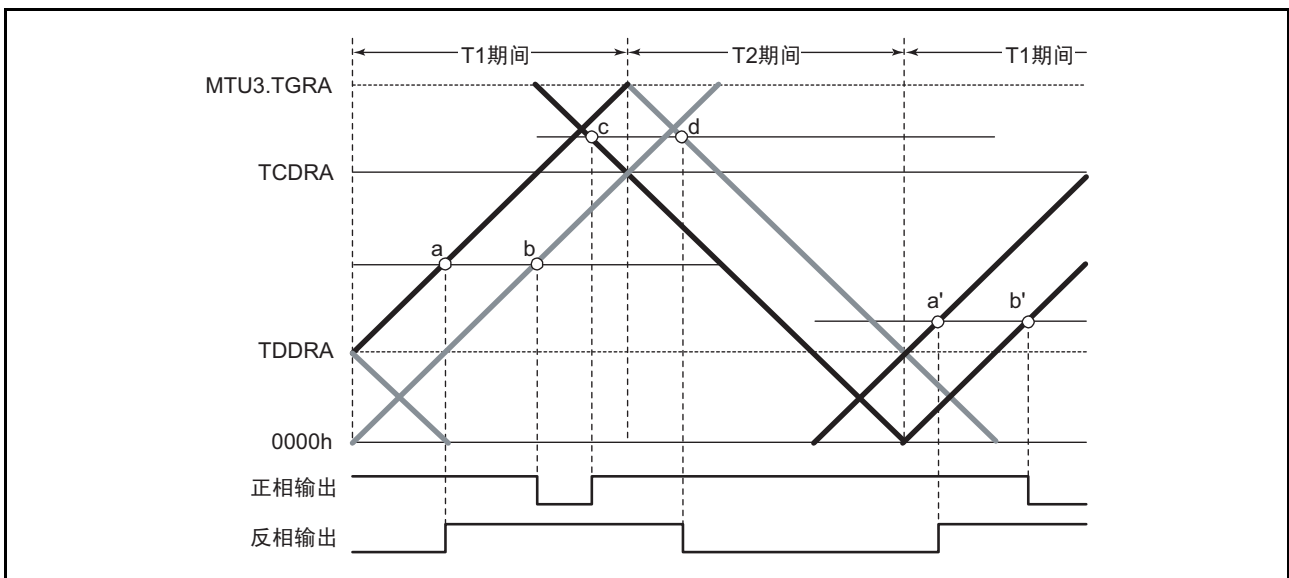


图 16.47 互补 PWM 模式的波形输出例子 (MTU3 和 MTU4) (1)

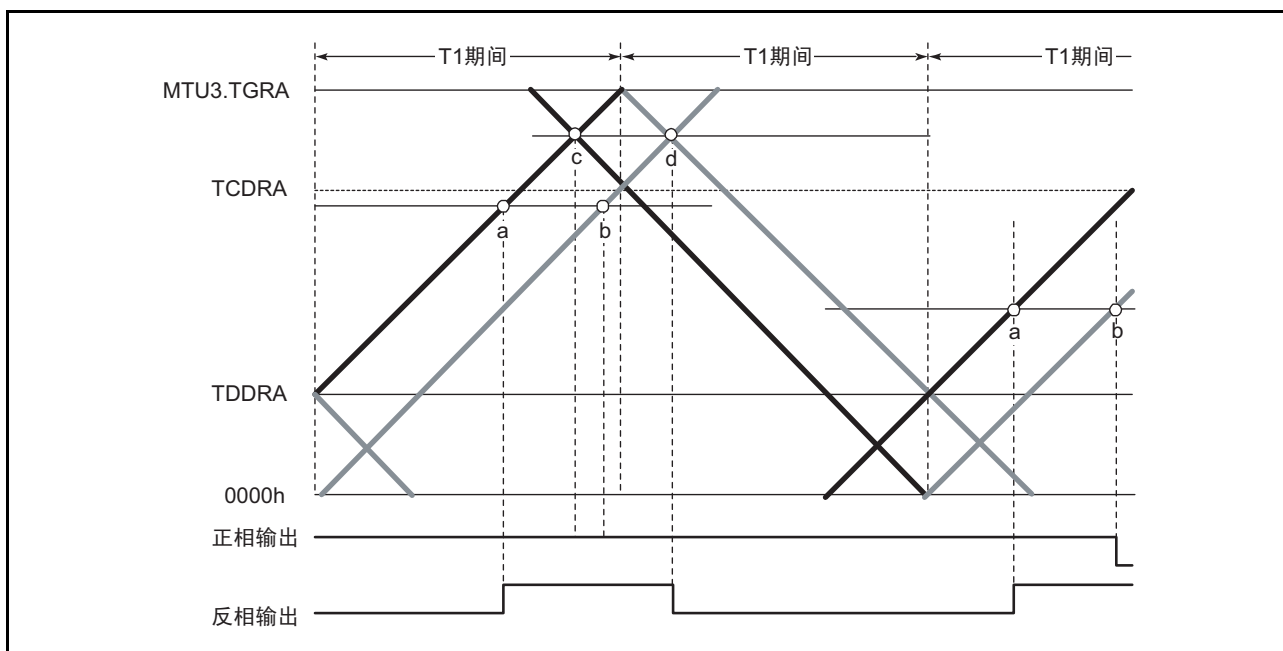


图 16.48 互补 PWM 模式的波形输出例子 (MTU3 和 MTU4) (2)

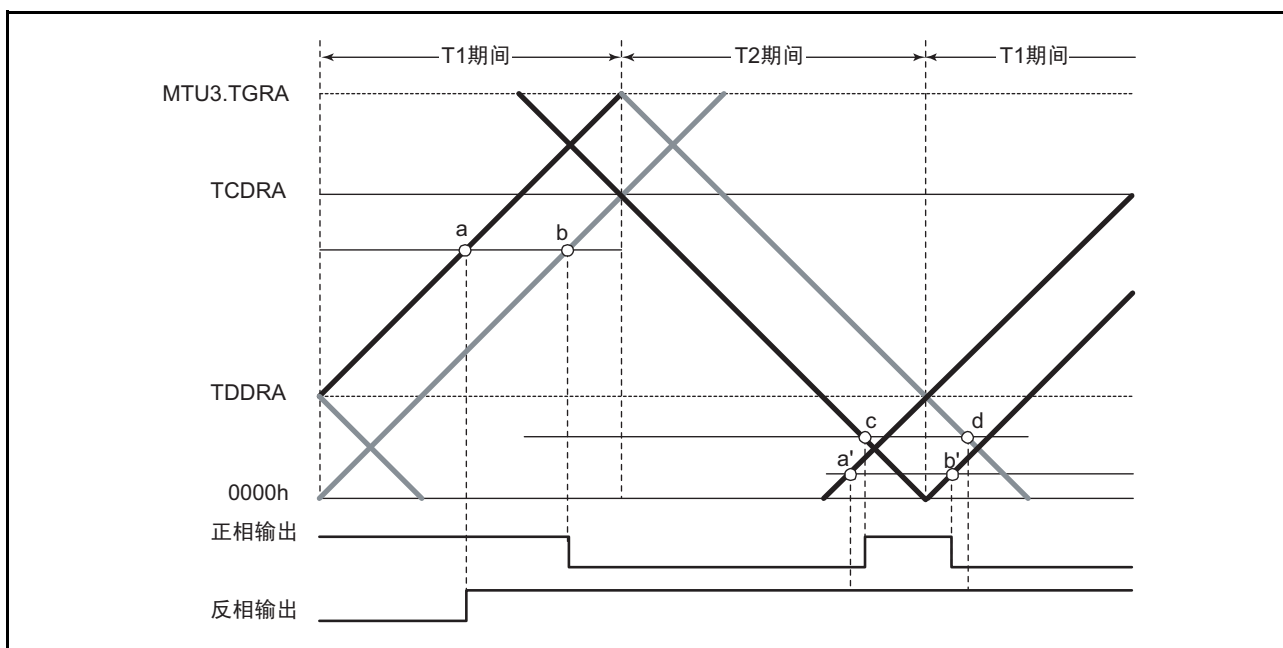


图 16.49 互补 PWM 模式的波形输出例子 (MTU3 和 MTU4) (3)

(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 16.50 ~ 图 16.54 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。

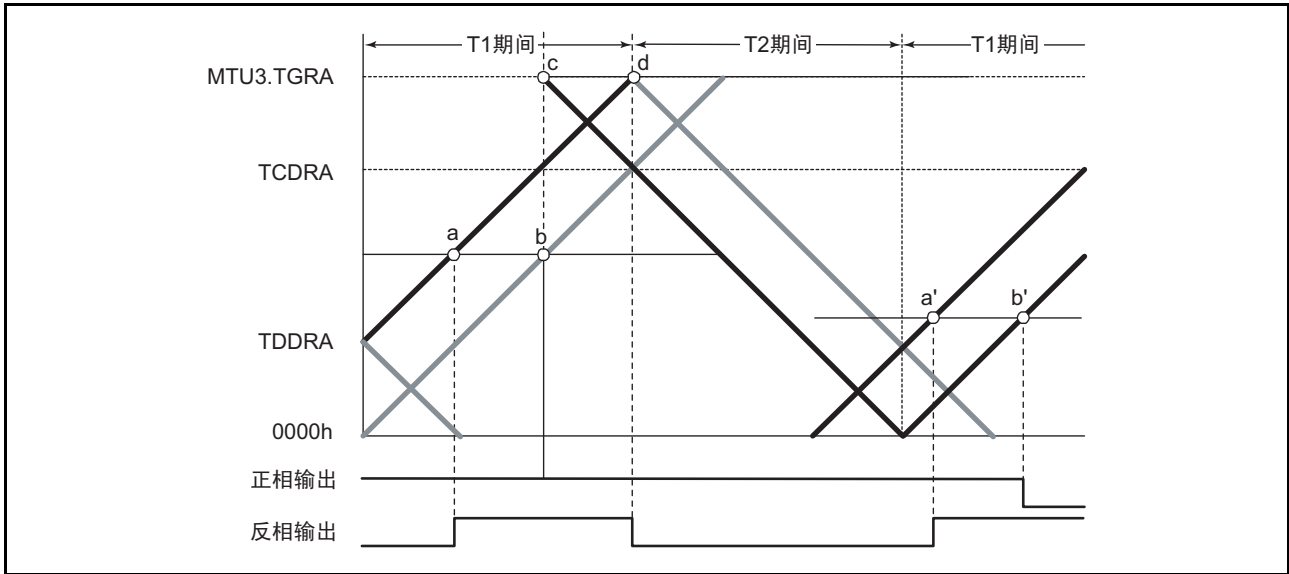


图 16.50 互补 PWM 模式的 0%、100% 波形输出例子 (MTU3 和 MTU4) (1)

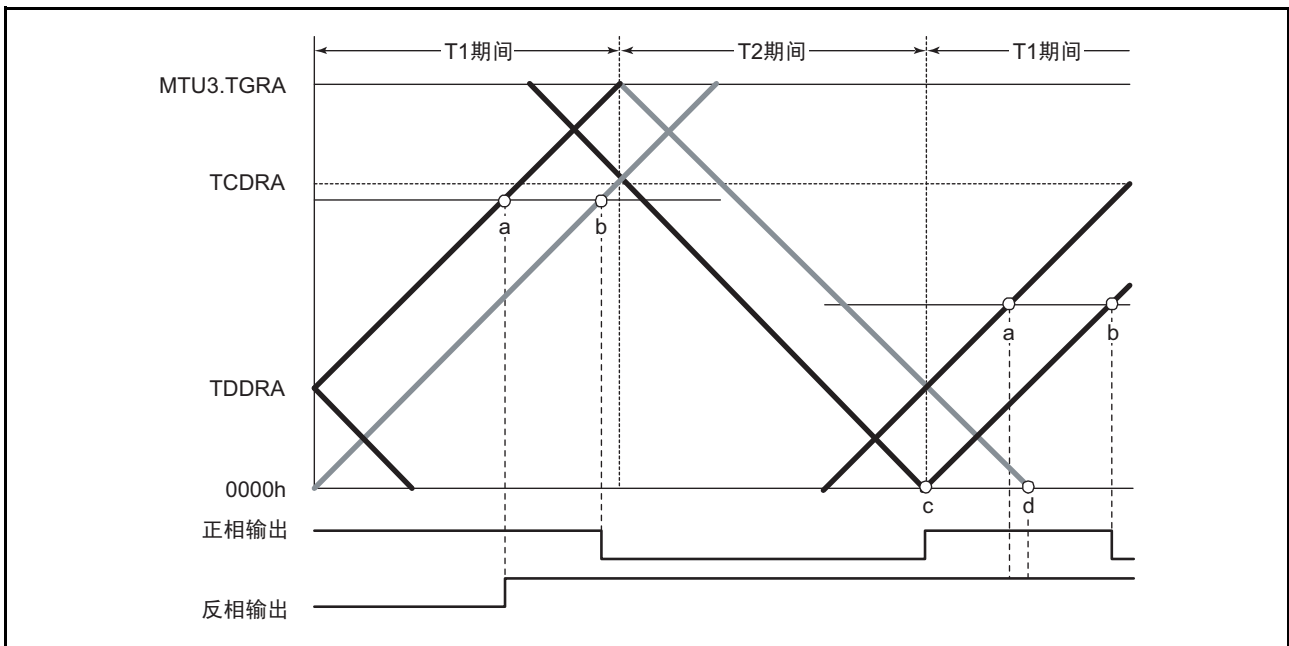


图 16.51 互补 PWM 模式的 0%、100% 波形输出例子 (MTU3 和 MTU4) (2)

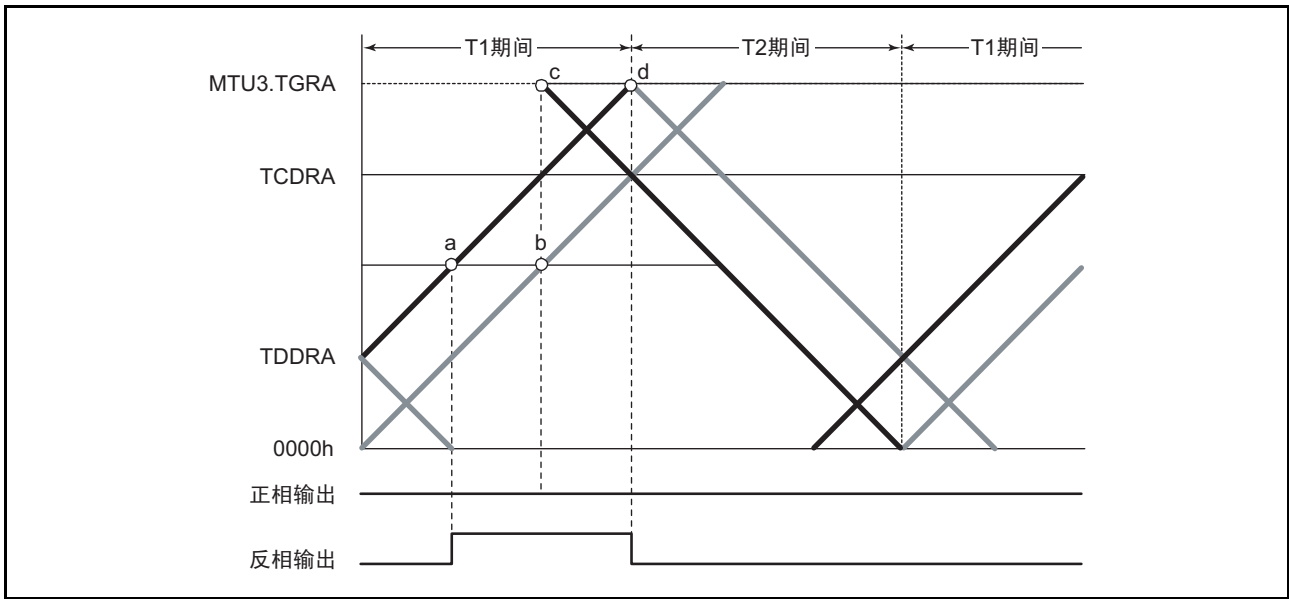


图 16.52 互补 PWM 模式的 0%、100% 波形输出例子 (MTU3 和 MTU4) (3)

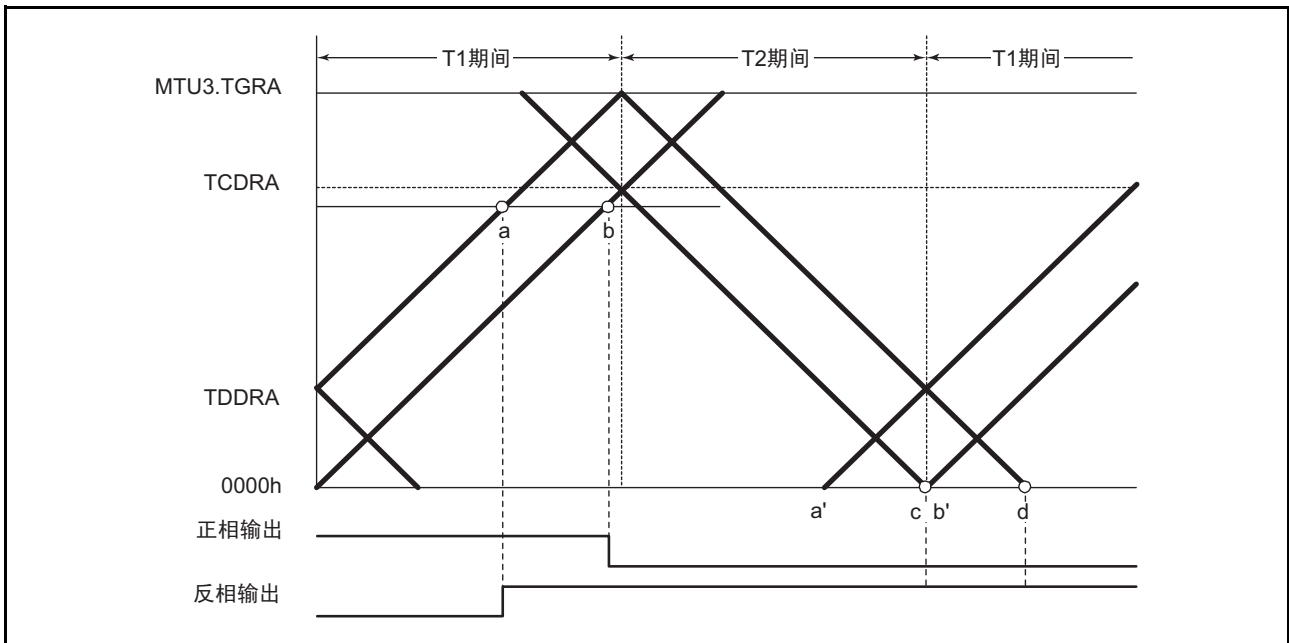


图 16.53 互补 PWM 模式的 0%、100% 波形输出例子 (MTU3 和 MTU4) (4)

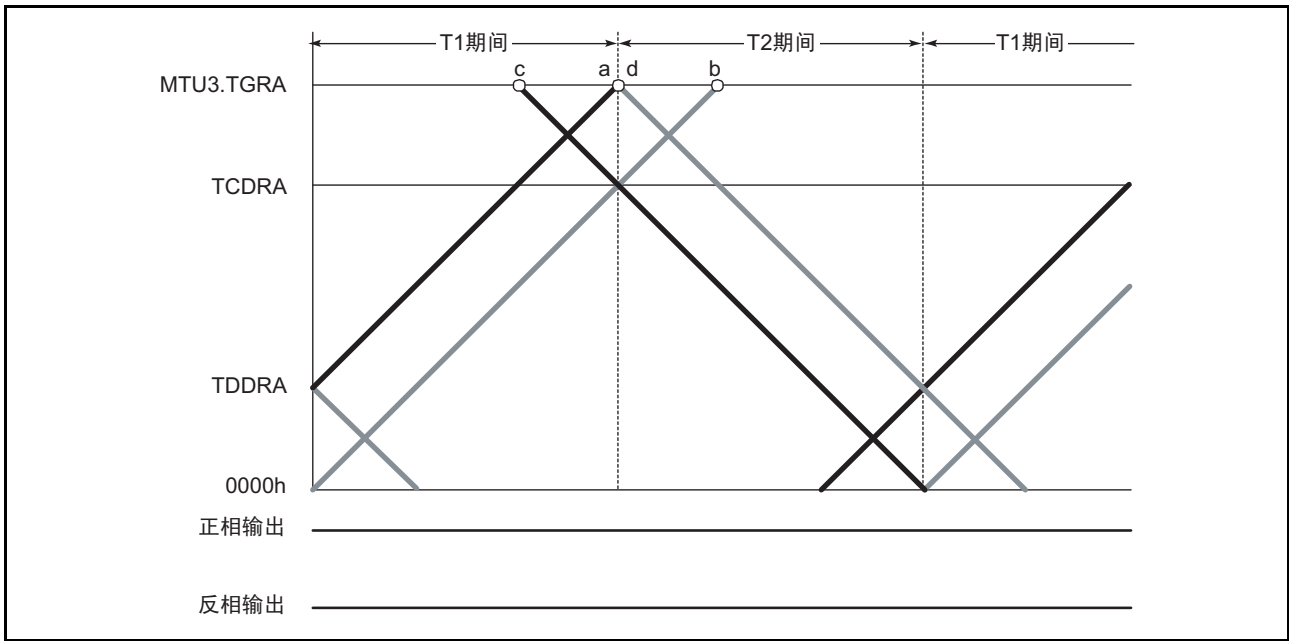


图 16.54 互补 PWM 模式的 0%、100% 波形输出例子 (MTU3 和 MTU4) (5)

(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 16.55 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。

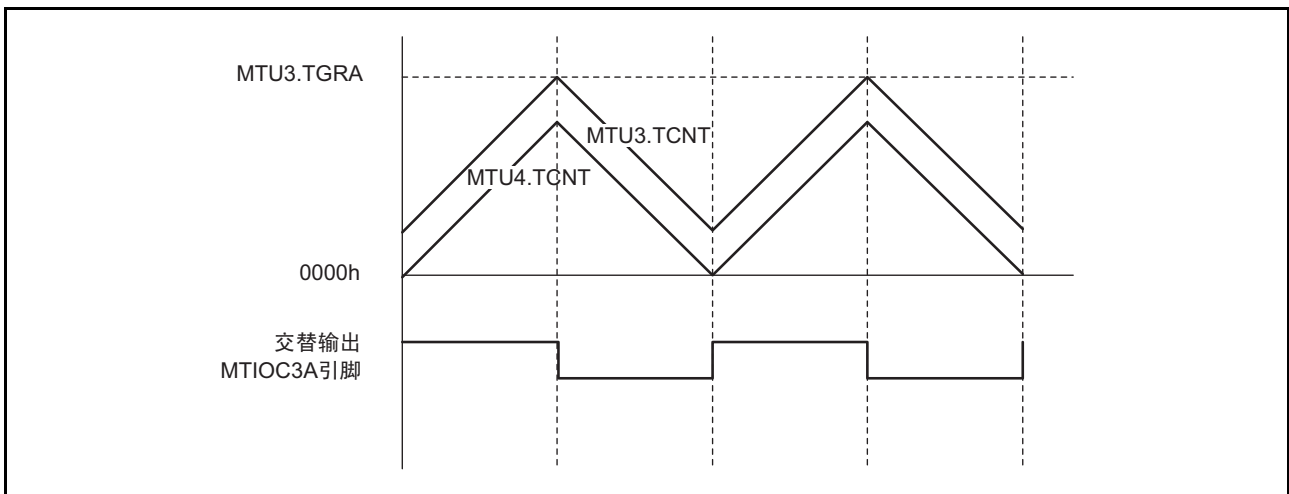


图 16.55 与 PWM 输出同步的交替输出波形例子 (MTU3 和 MTU4)



(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 16.56 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。

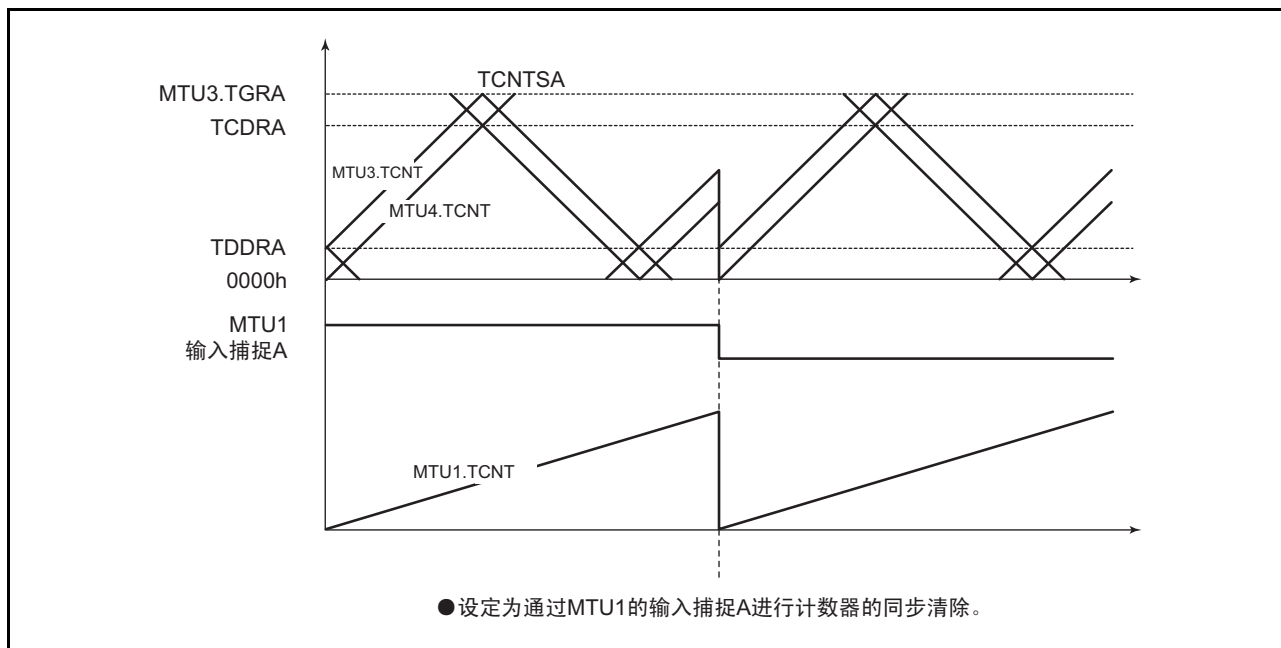


图 16.56 与其他通道同步的计数器清除 (MTU3 和 MTU4)

(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 16.57 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 16.57 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)。

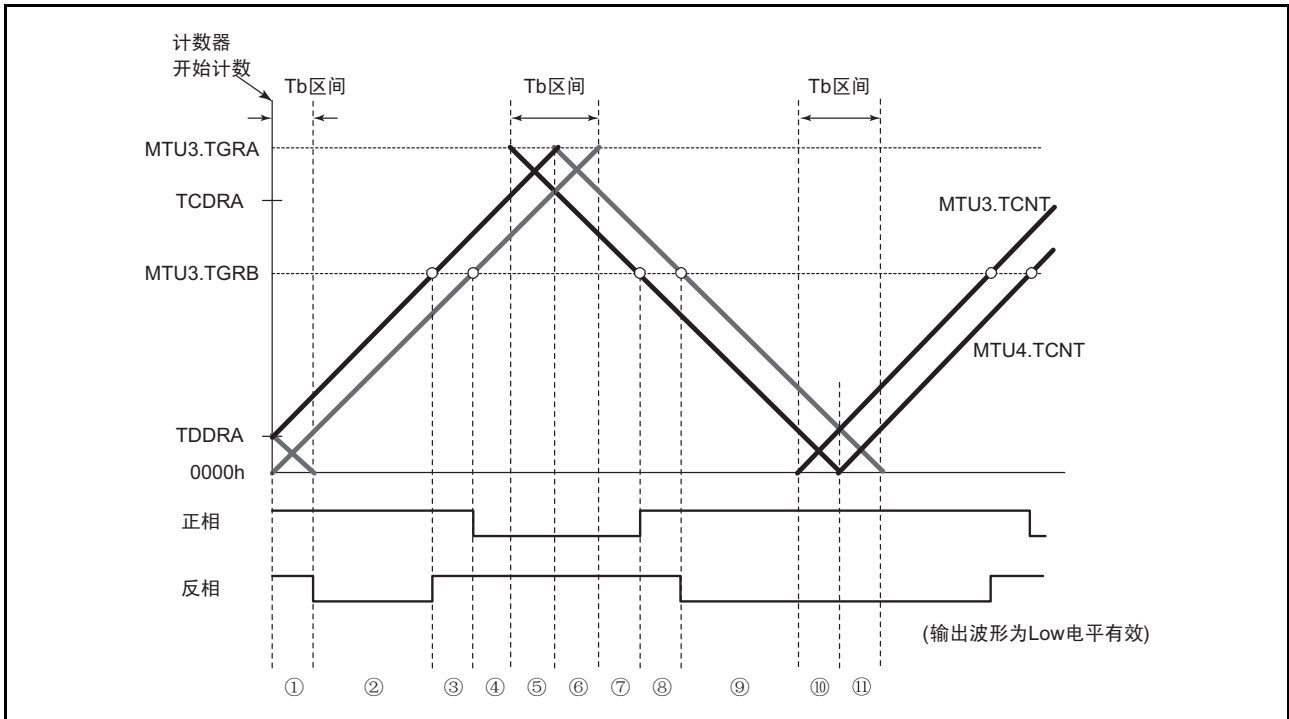


图 16.57 同步计数器清除的时序 (MTU3 和 MTU4)

- 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 16.58 所示。

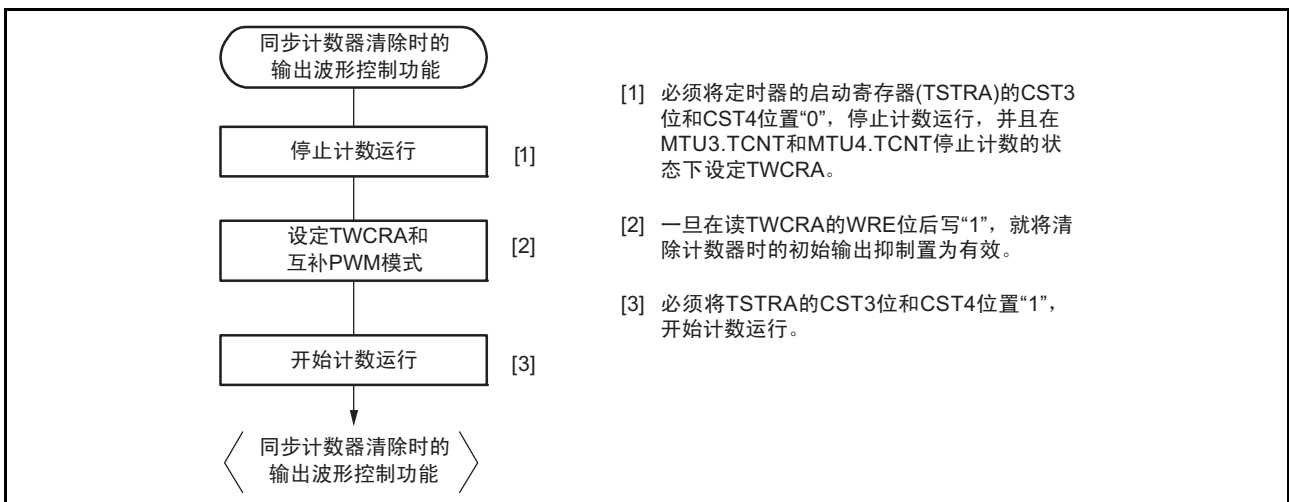


图 16.58 互补 PWM 模式中同步计数器清除时的输出波形控制设定步骤例子 (MTU3 和 MTU4)

• 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 16.59 ~图 16.62 所示。在此，图 16.59 ~图 16.62 的同步计数器清除时序分别是图 16.57 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。

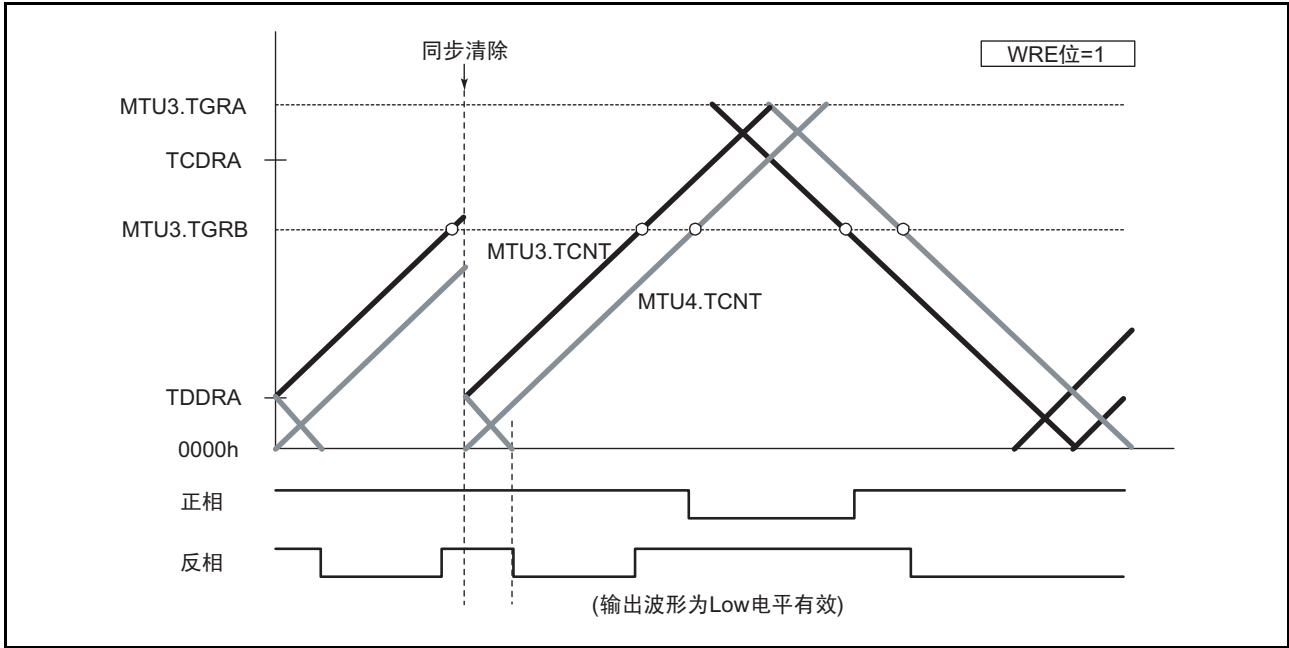


图 16.59 在递增计数过程中的死区时间发生同步清除的情况  
(图 16.57 的时序③、TWCRA 寄存器的 WRE 位 =1)

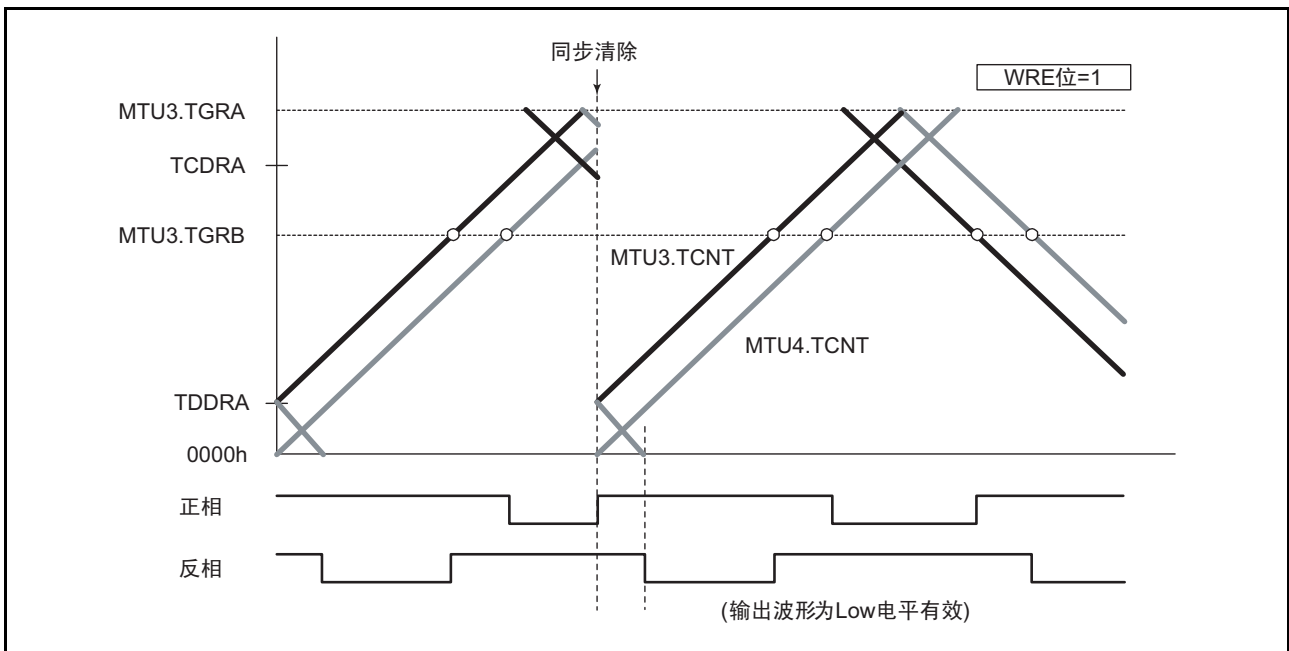


图 16.60 在波峰 Tb 区间发生同步清除的情况  
(图 16.57 的时序⑥、TWCRA 寄存器的 WRE 位 =1)

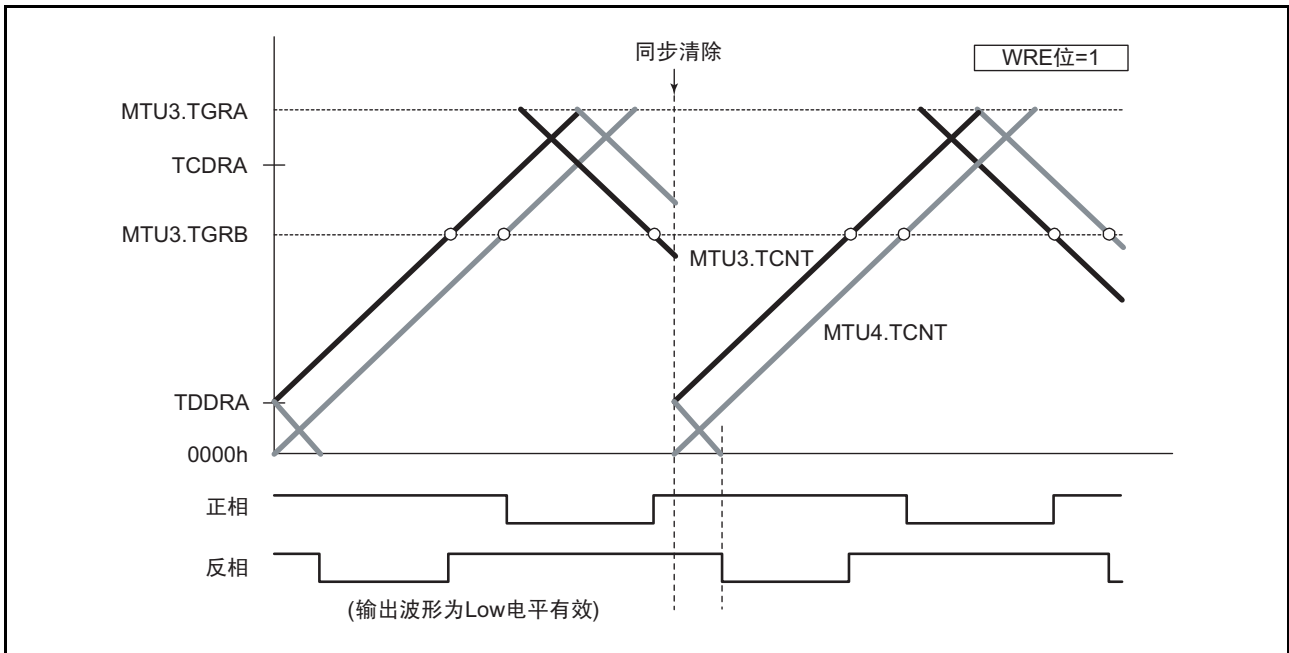


图 16.61 在递减计数过程中的死区时间发生同步清除的情况  
(图 16.57 的时序⑧、TWCRA 寄存器的 WRE 位 =1)

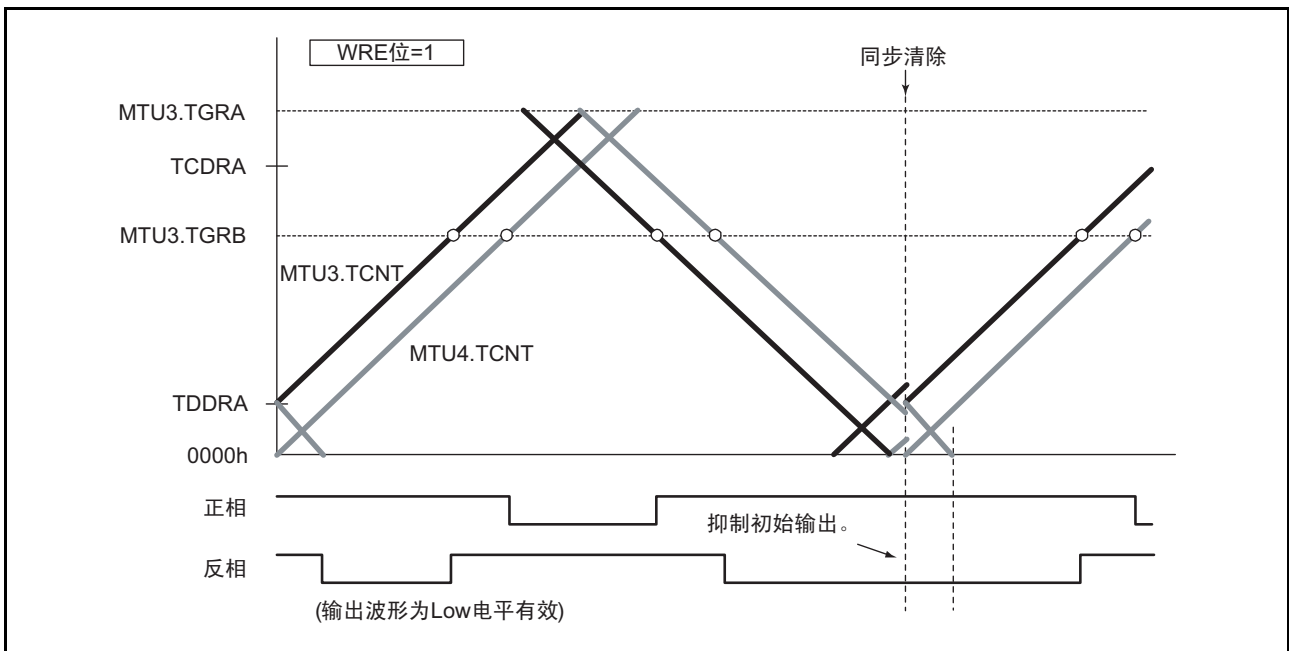


图 16.62 在波谷 Tb 区间发生同步清除的情况  
(图 16.57 的时序⑩、TWCRA 寄存器的 WRE 位 =1)

(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 16.63 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“16.3.10(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)”。

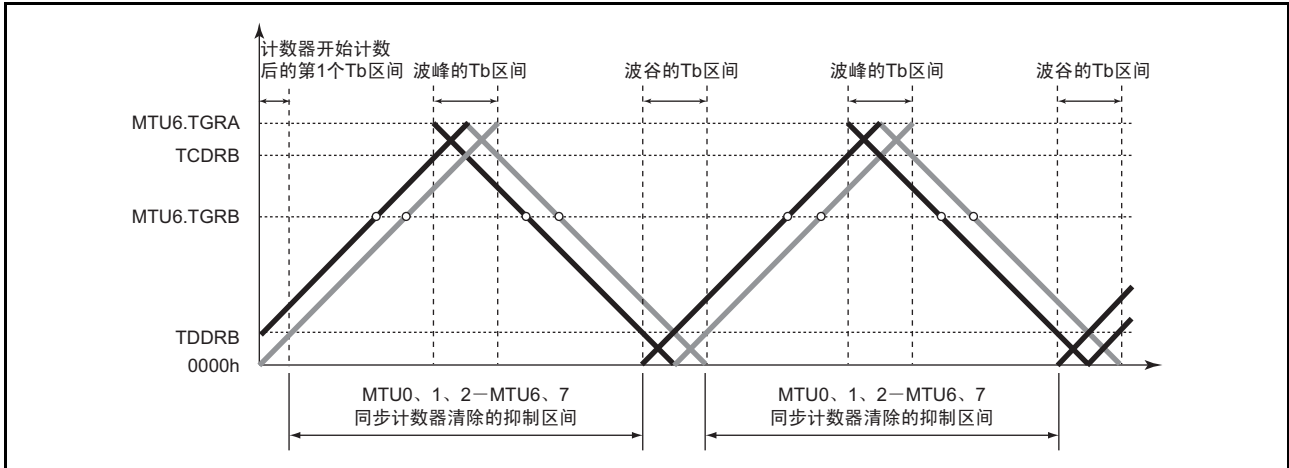


图 16.63 通过将 TWCRCB.SCC 位置位产生 MTU0、1、2 – MTU6、7 的同步清除抑制区间

• 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 16.64 所示。

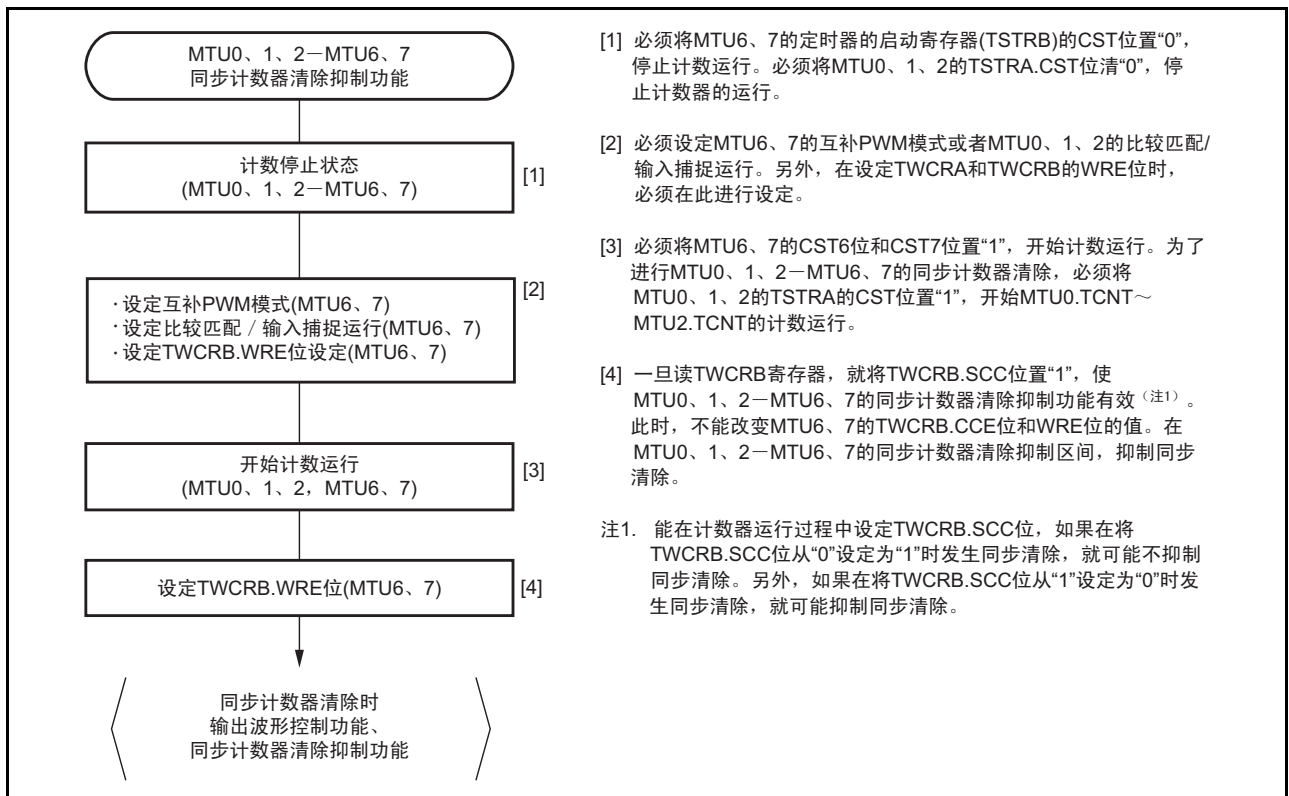


图 16.64 MTU0、1、2 – MTU6、7 的同步计数器清除抑制功能的设定步骤例子

- 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 16.65 ~ 图 16.68 所示。在此，图 16.65 ~ 图 16.68 的同步计数器清除时序分别是图 16.57 的③、⑥、⑧、⑪所示的时序。在此例中，将 MTU6 和 MTU7 的 TWCRB.WRE 位置“1”。

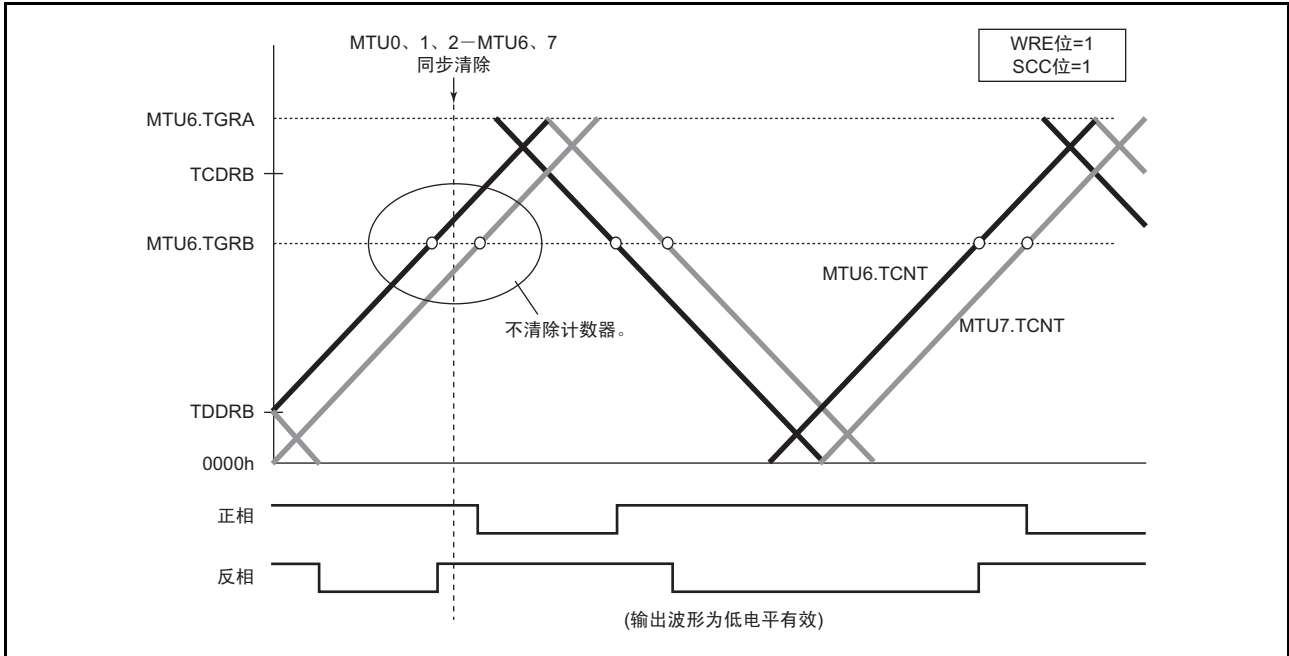


图 16.65 在递增计数过程中的死区时间发生同步清除的情况  
(图 16.57 的时序③、MTU6 和 MTU7 的 TWCRB 寄存器的 WRE 位 =1、SCC 位 =1)

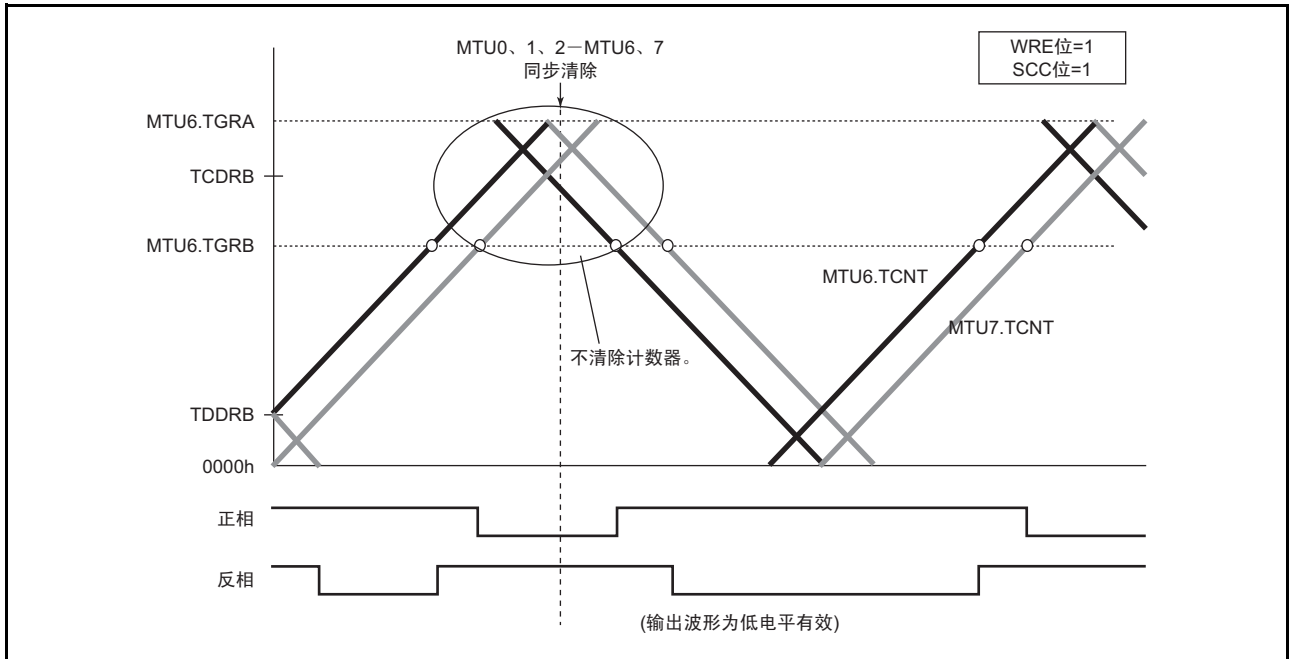


图 16.66 在波峰 Tb 区间发生同步清除的情况  
(图 16.57 的时序⑥、MTU6 和 MTU7 的 TWCRB 寄存器的 WRE 位 =1、SCC 位 =1)

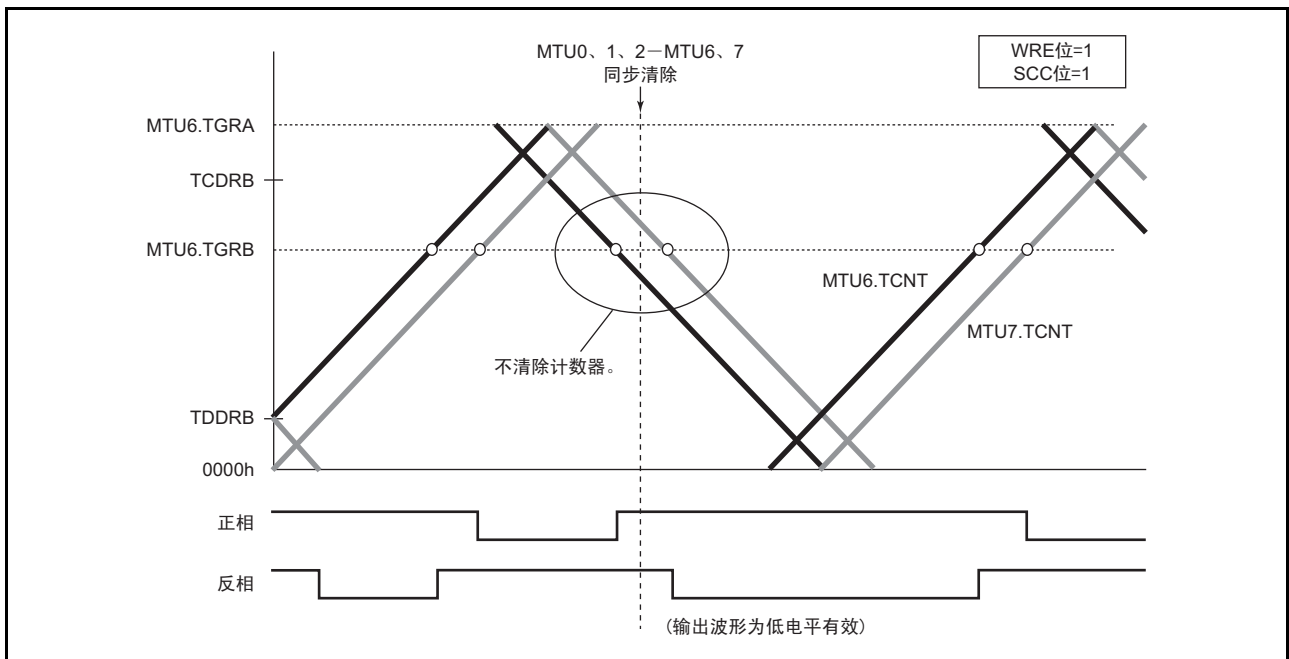


图 16.67 在递减计数过程中的死区时间发生同步清除的情况  
 (图 16.57 的时序⑧、MTU6 和 MTU7 的 TWCRB 寄存器的 WRE 位 =1、SCC 位 =1)

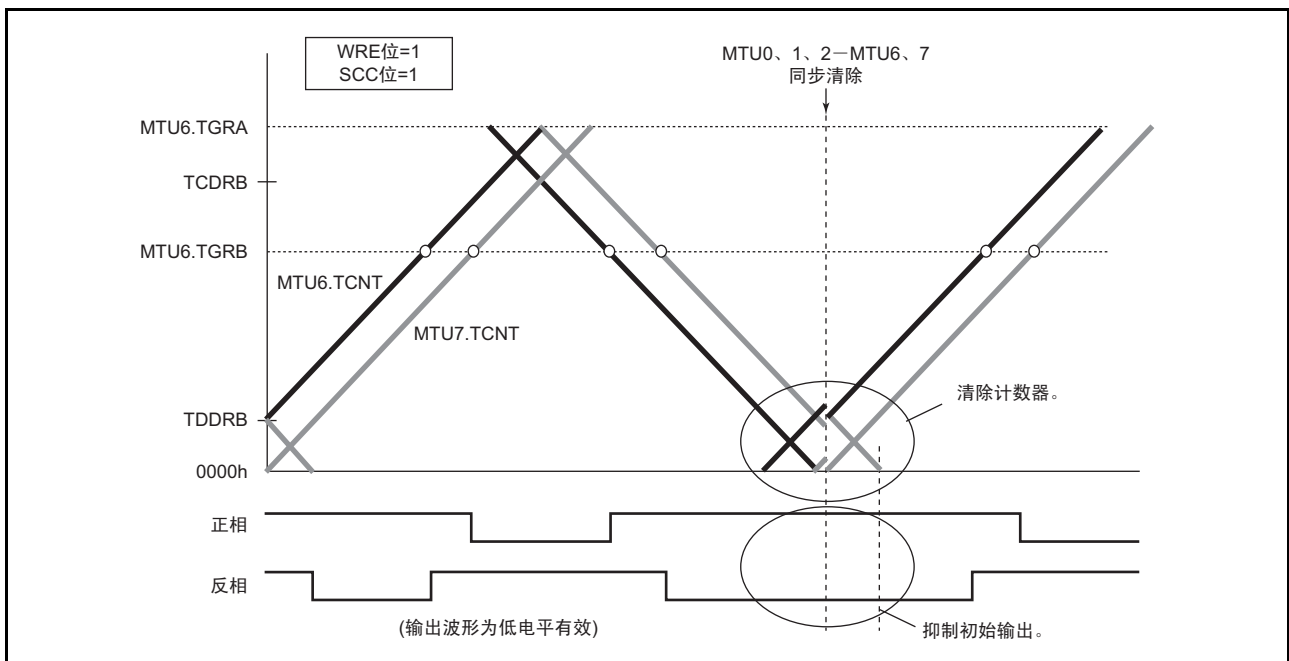


图 16.68 在波谷 Tb 区间发生同步清除的情况  
 (图 16.57 的时序⑩、MTU6 和 MTU7 的 TWCRB 寄存器的 WRE 位 =1、SCC 位 =1)

## (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 16.69 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”。

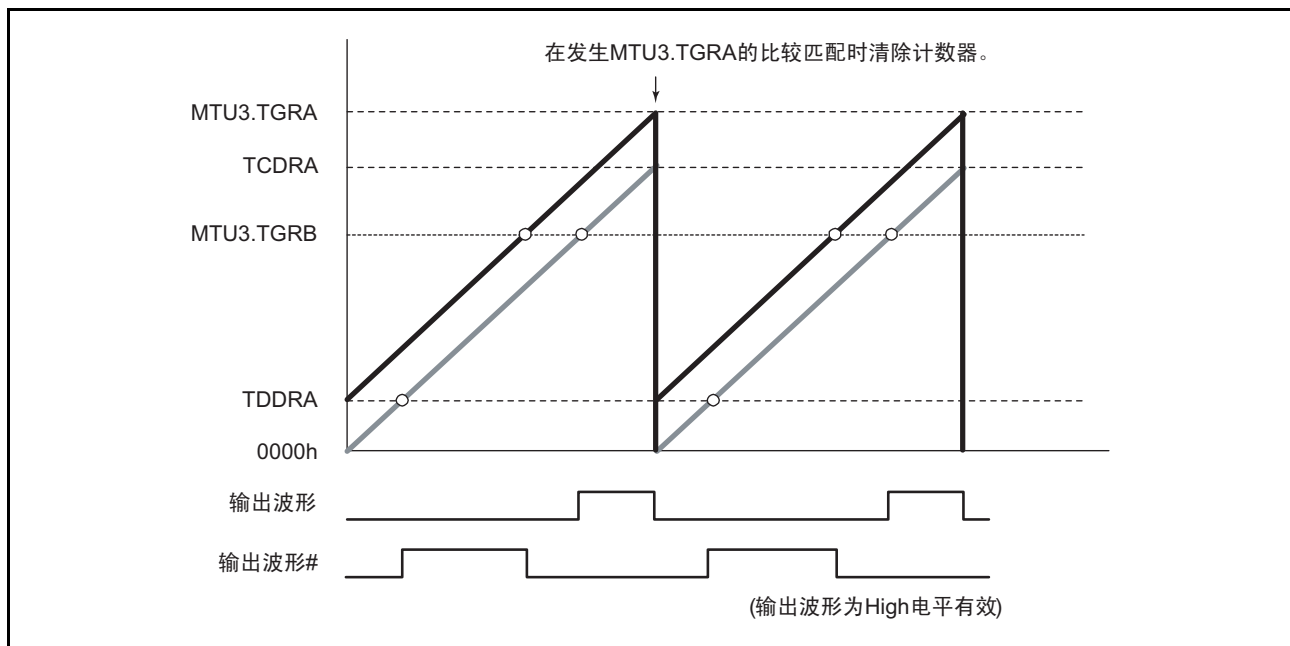


图 16.69 在 MTU3.TGRA 的比较匹配时清除计数器的运行例子



(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 16.70 ~ 图 16.73 所示。

在通过使用霍尔元件等检测到的外部信号进行三相无刷 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 (必须设定 I/O 端口的 ICR 寄存器)。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。

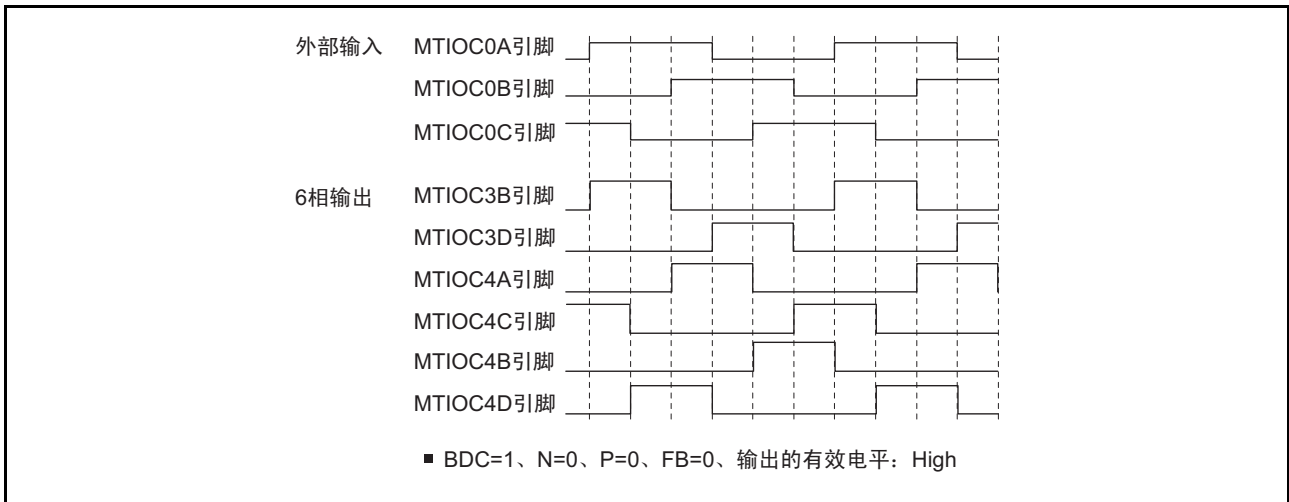


图 16.70 通过外部输入进行输出相转换的运行例子 (1)

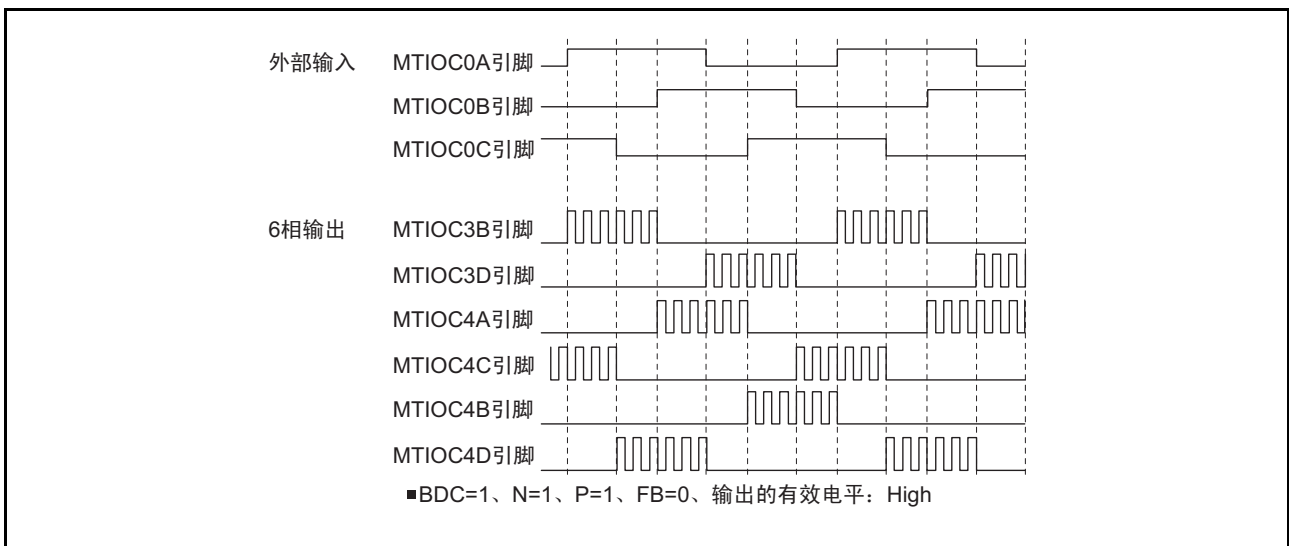


图 16.71 通过外部输入进行输出相转换的运行例子 (2)

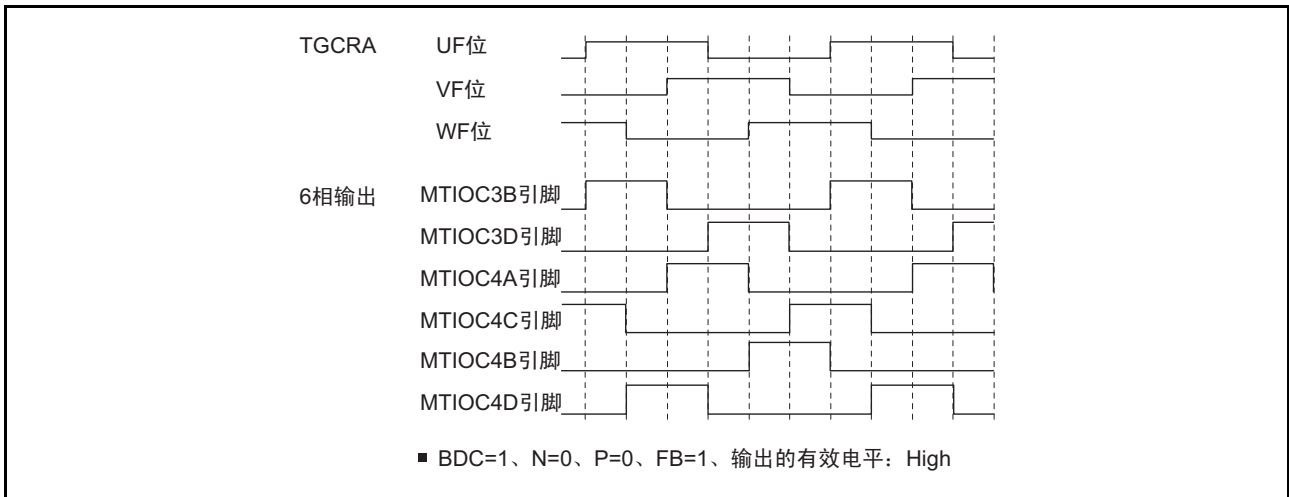


图 16.72 通过设定 UF 位、VF 位和 WF 位进行输出相转换的运行例子 (1)

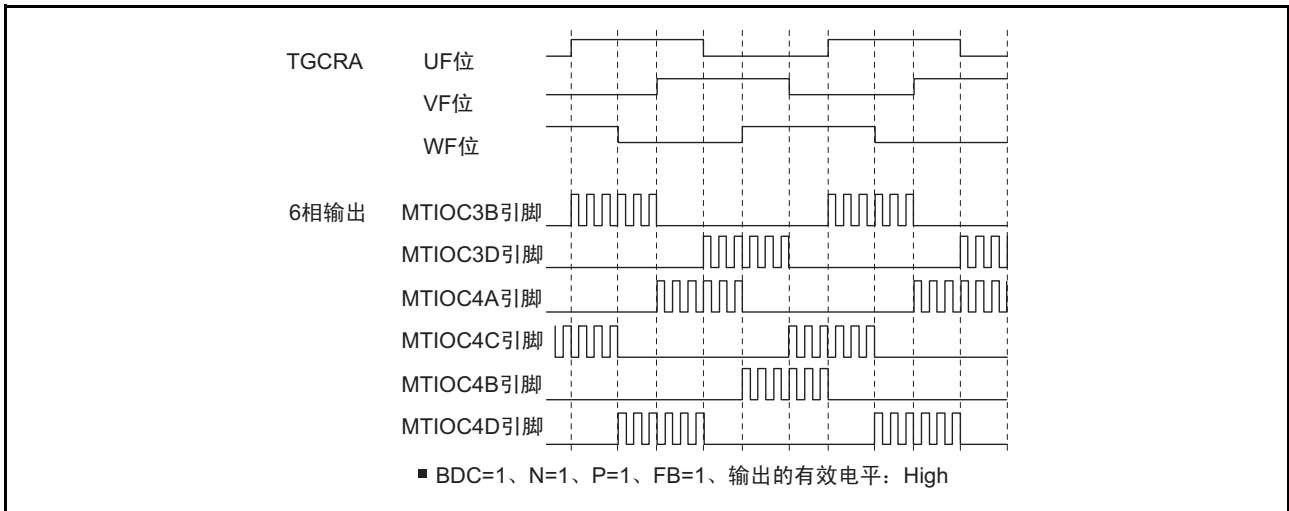


图 16.73 通过设定 UF 位、VF 位和 WF 位进行输出相转换的运行例子 (2)

(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 “16.3.8(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 16.74 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。

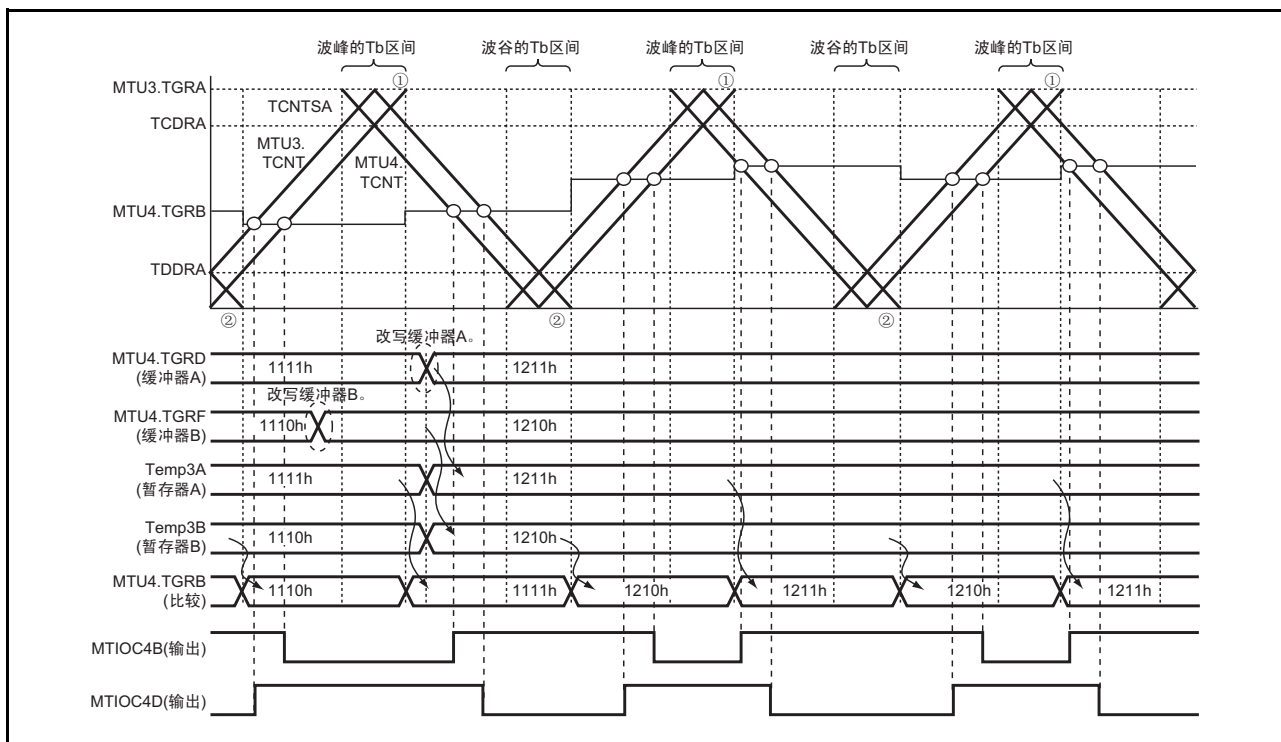


图 16.74 双缓冲功能的运行例子

缓冲器的写入值小于和大于 TDDRA 的值的情况分别如图 16.75 和图 16.76 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。

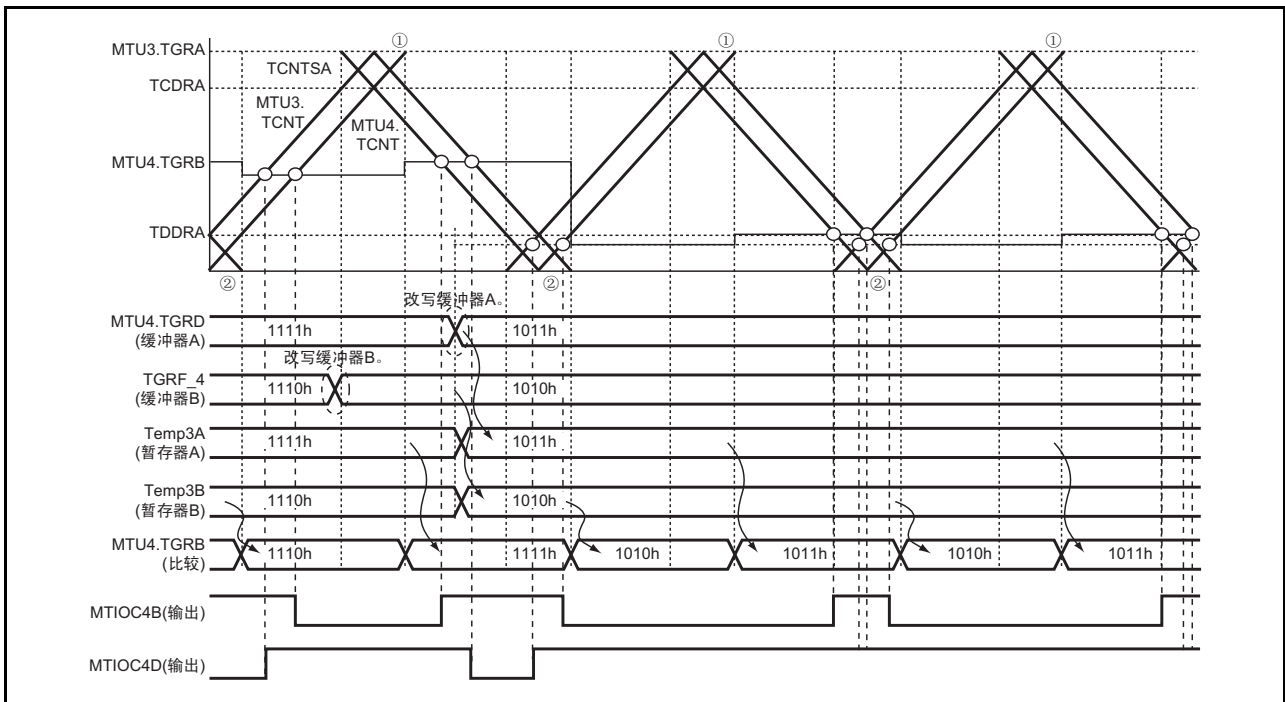


图 16.75 双缓冲功能的运行例子 (缓冲器的写入值小于 TDDRA 值的情况)

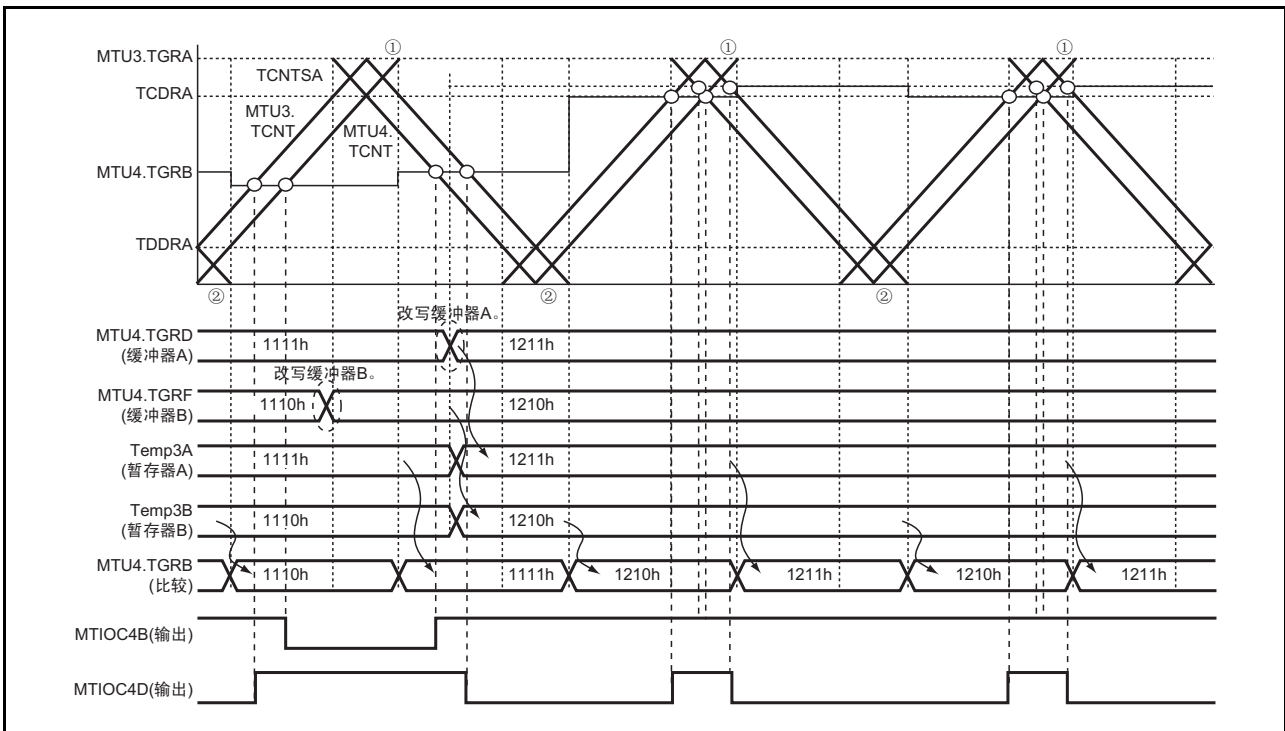


图 16.76 双缓冲功能的运行例子 (缓冲器的写入值大于 TCDRA 值的情况)

(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“16.3.9 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 16.77 和图 16.78 所示。

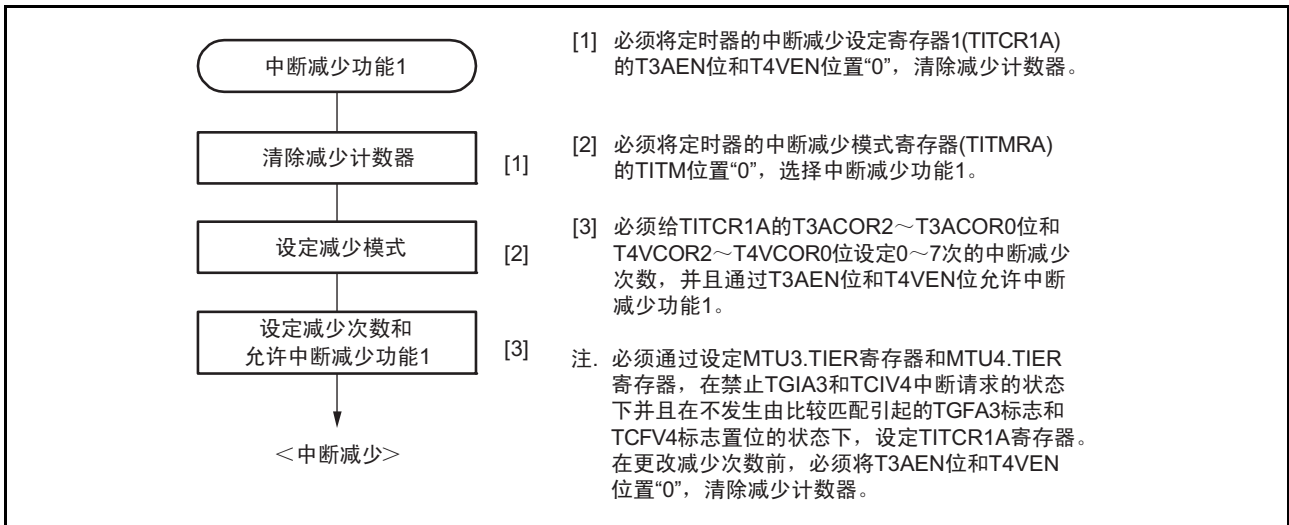


图 16.77 中断减少功能的设定步骤例子

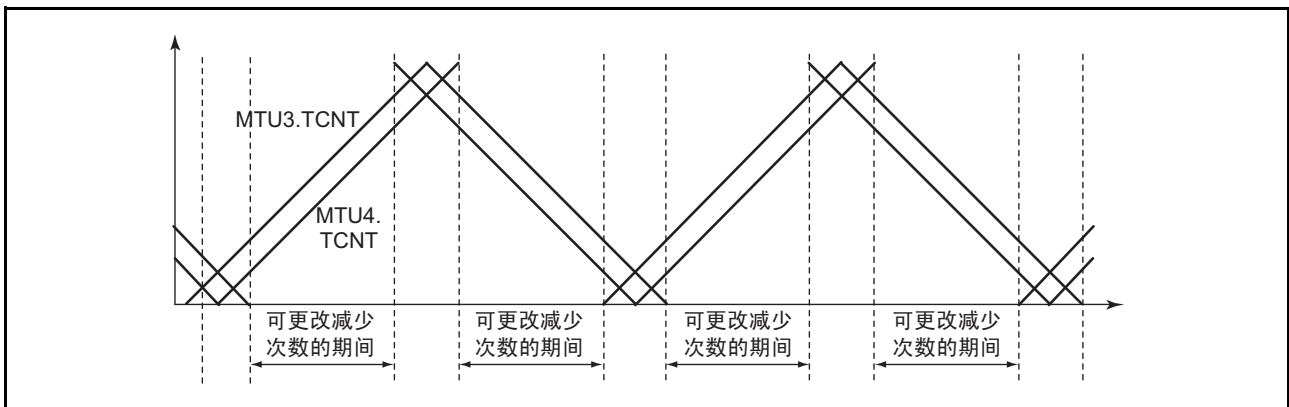


图 16.78 中断减少次数的可更改期间

## (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 16.79 所示。

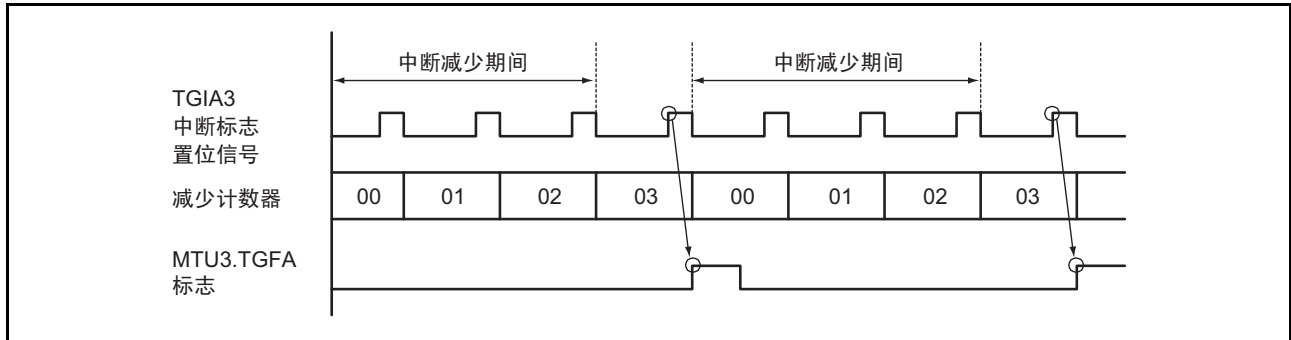


图 16.79 中断减少功能 1 的运行例子

## (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 16.80 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 16.81 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 16.82 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。

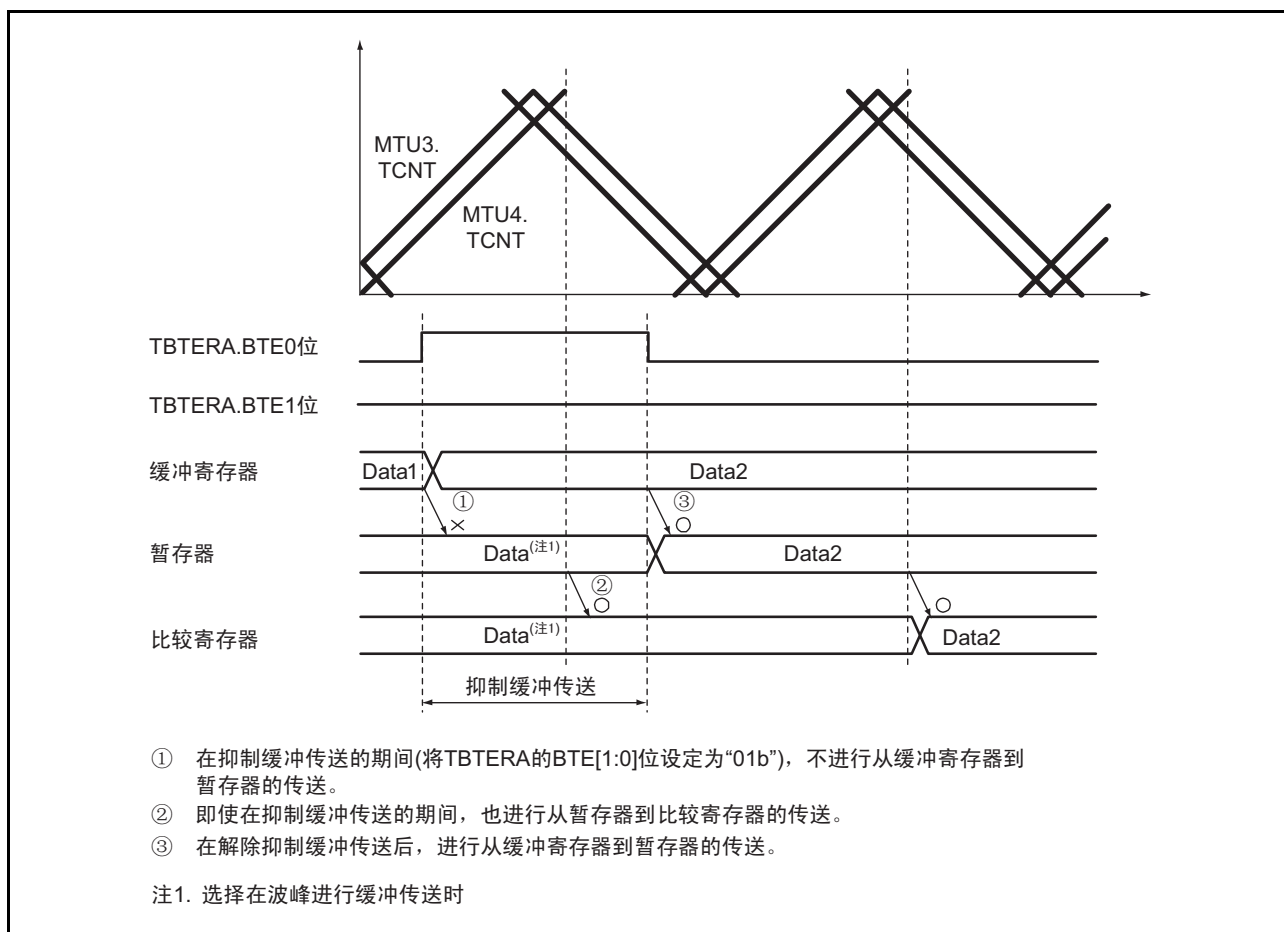


图 16.80 设定为抑制缓冲传送 (BTE[1:0]=01b) 时的运行例子

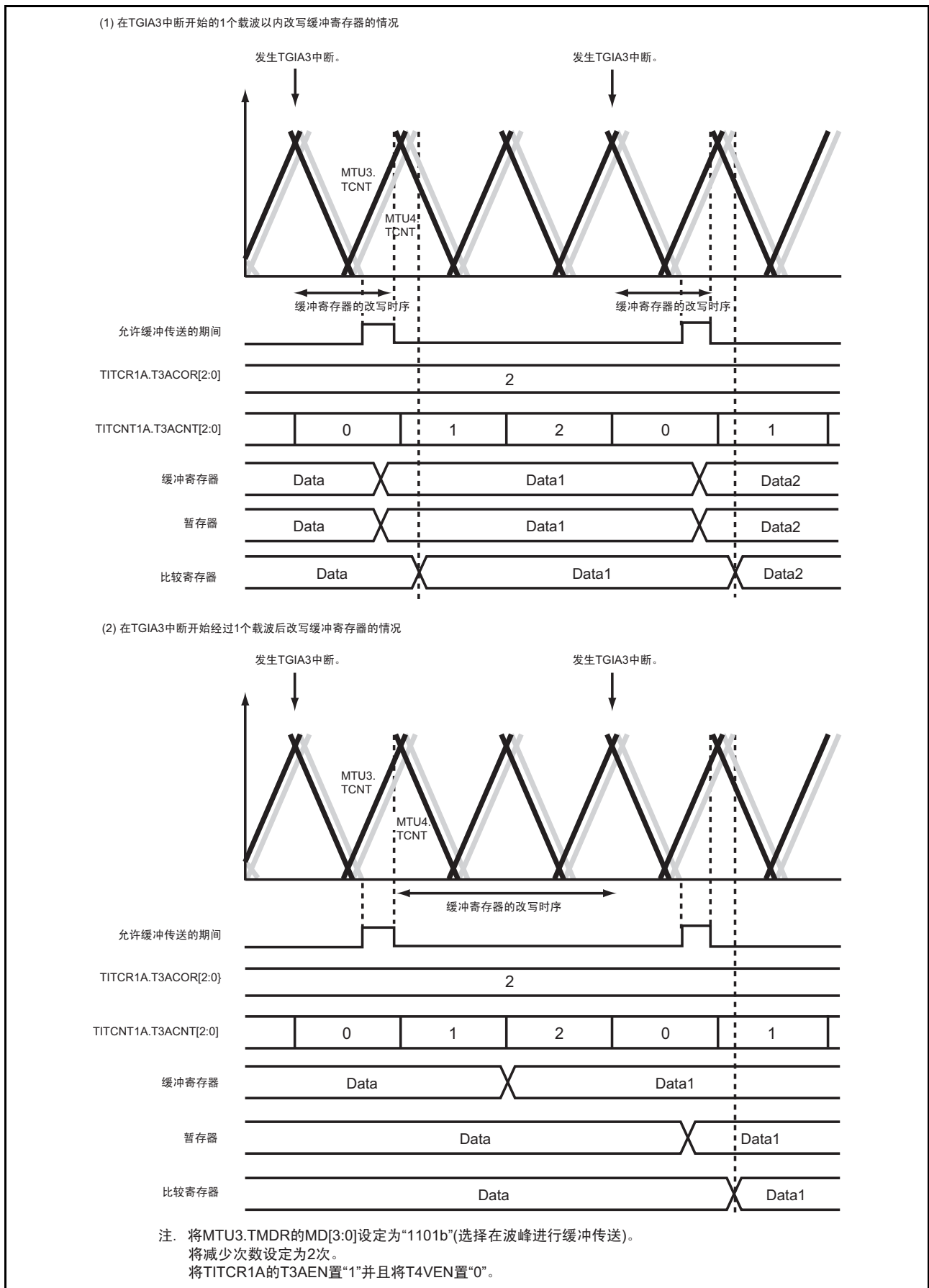


图 16.81 设定为缓冲传送联动中断减少功能 1 (BTE[1:0]=0) 时的运行例子



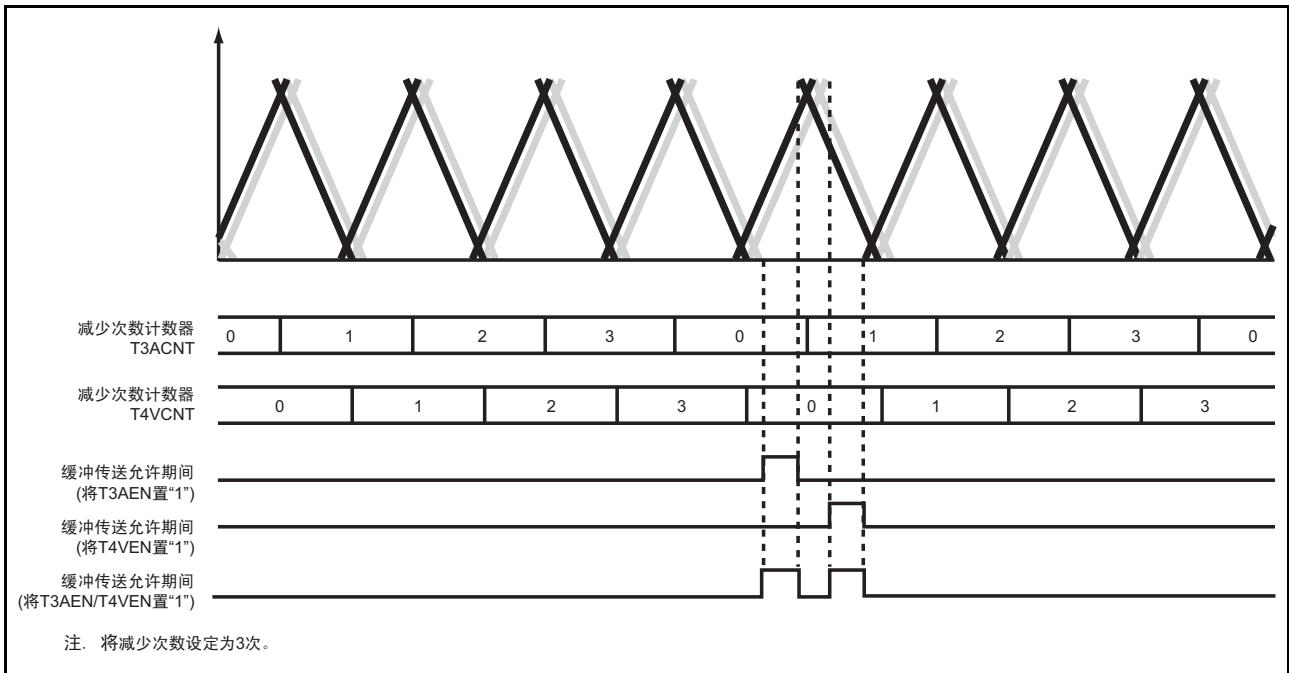


图 16.82 TITCR1A 寄存器的 T3AEN 位、T4VEN 位的设定和缓冲传送允许期间的关系

(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“17. 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 RX62T 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“8.10 振荡停止检测功能”。

### 16.3.9 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 16.83 所示。

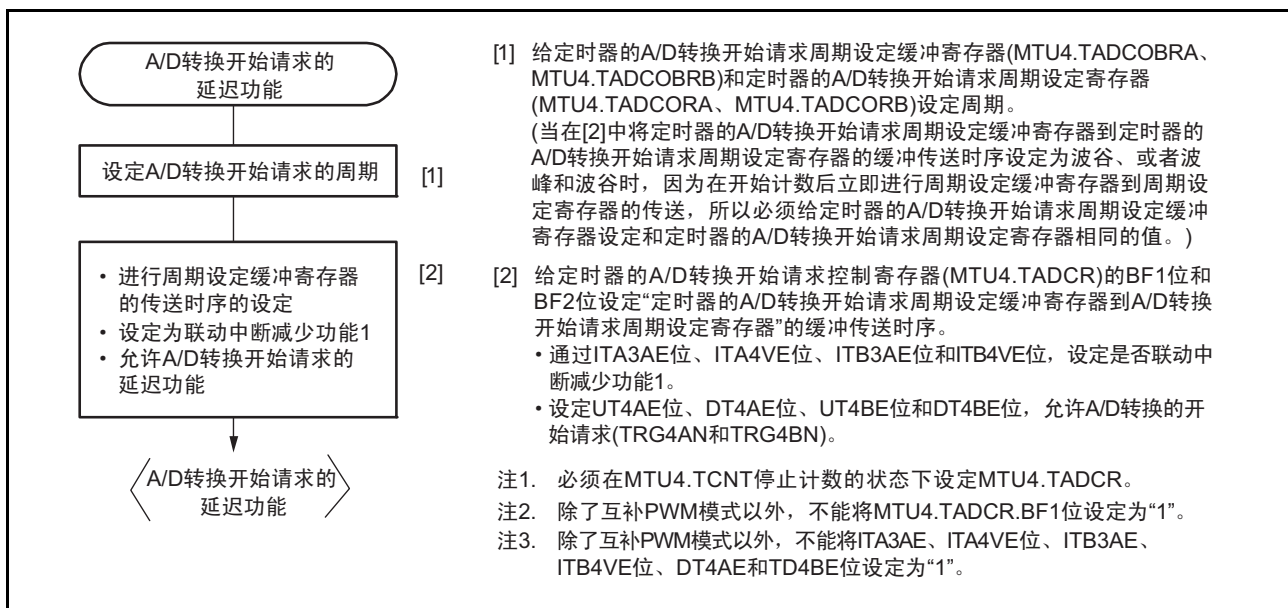


图 16.83 A/D 转换开始请求延迟功能的设定步骤例子

(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 16.84 所示。

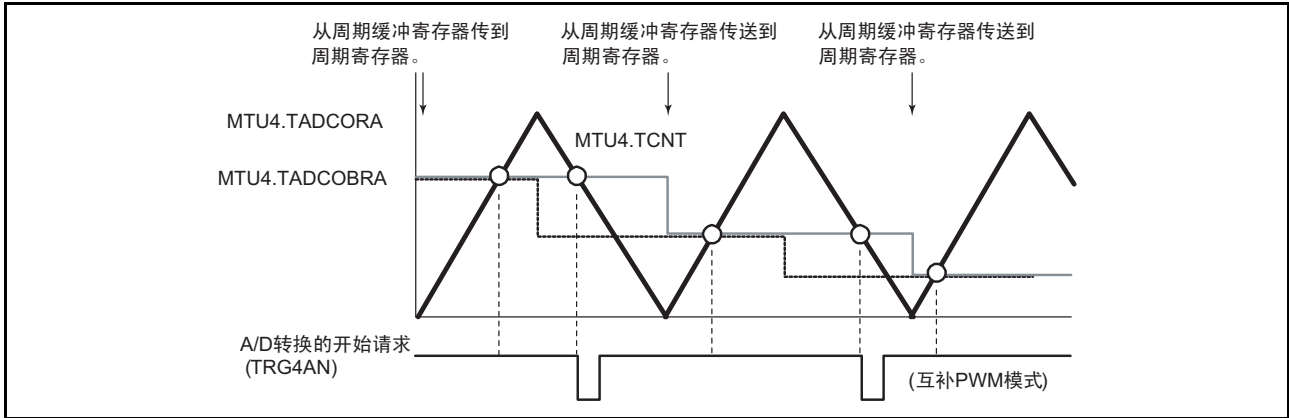


图 16.84 A/D 转换开始请求信号 (TRG4AN) 的基本运行例子

(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。

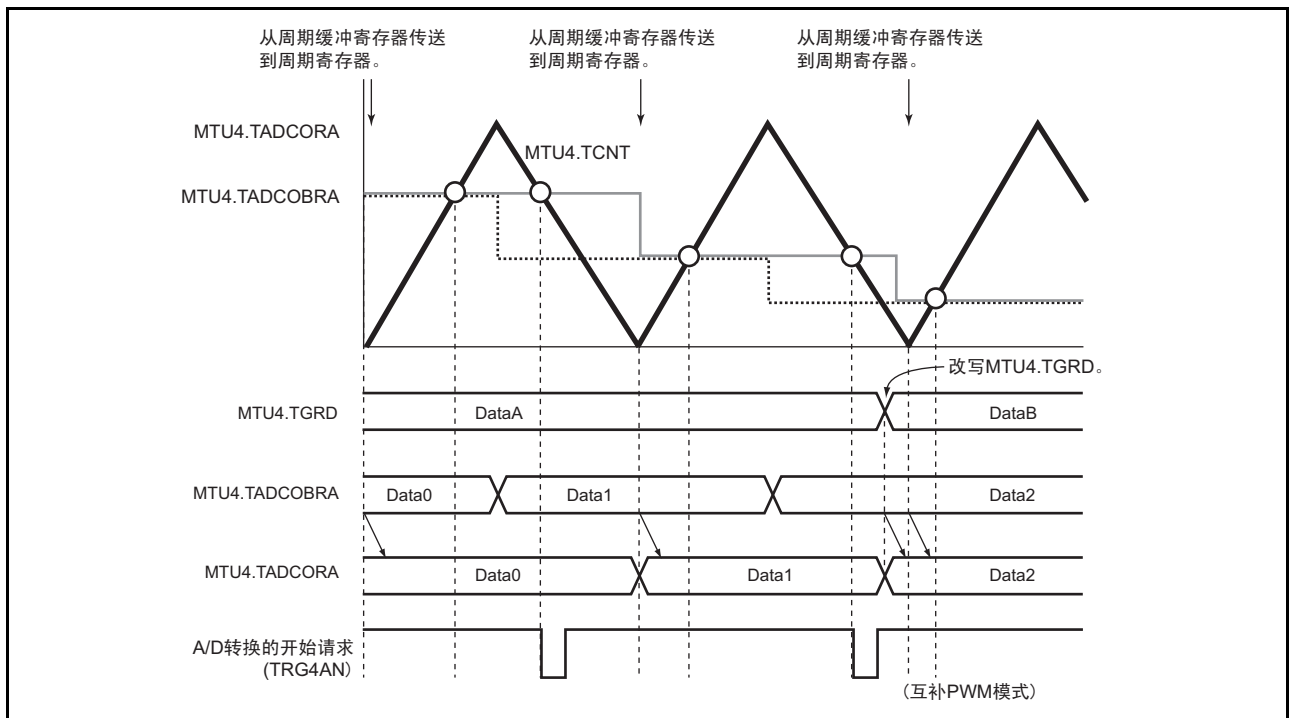


图 16.85 A/D 转换开始请求信号 (TRG4AN) 和缓冲传送的运行例子

(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.TCNT (MTU7.TCNT) 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 16.86 所示。

在 MTU4.TCNT (MTU7.TCNT) 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 16.87 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。

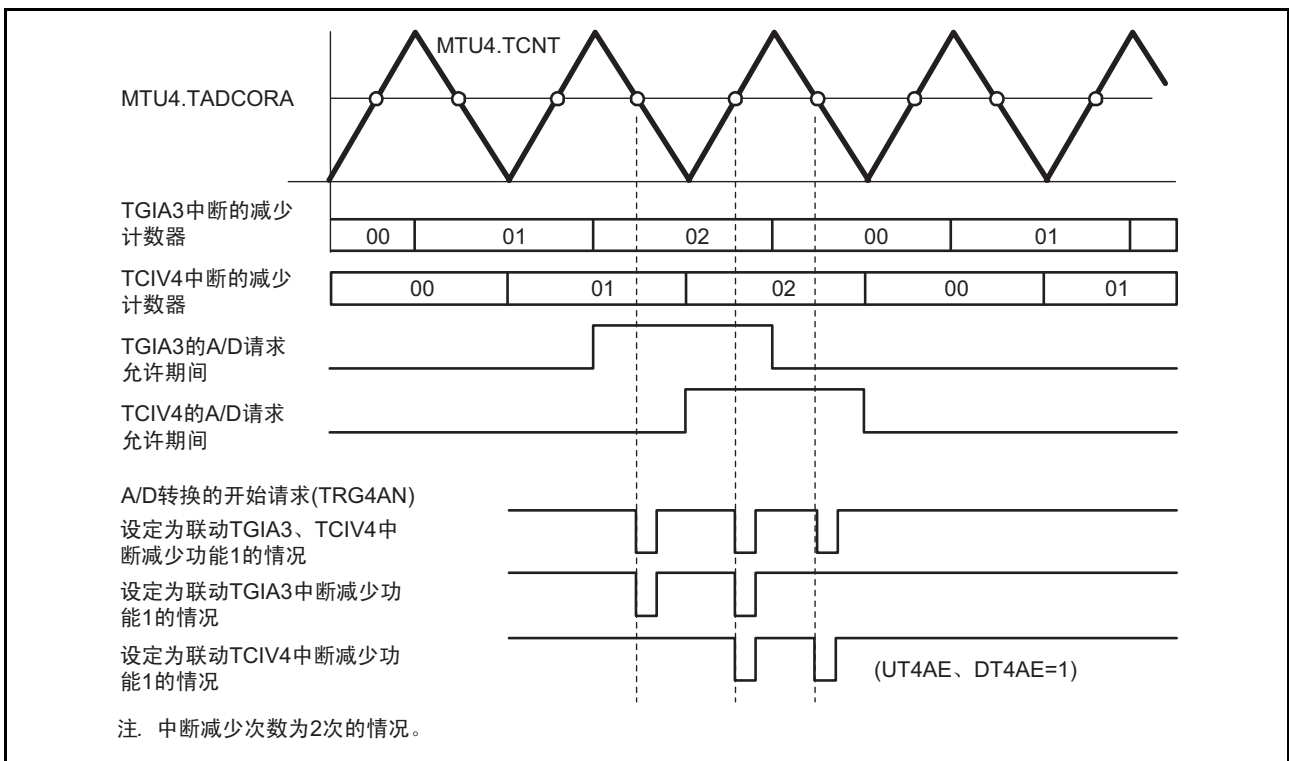


图 16.86 联动中断减少功能 1 时的 A/D 转换开始请求信号 (TRG4AN) 的运行例子 (UT4AE、DT4AE=1)

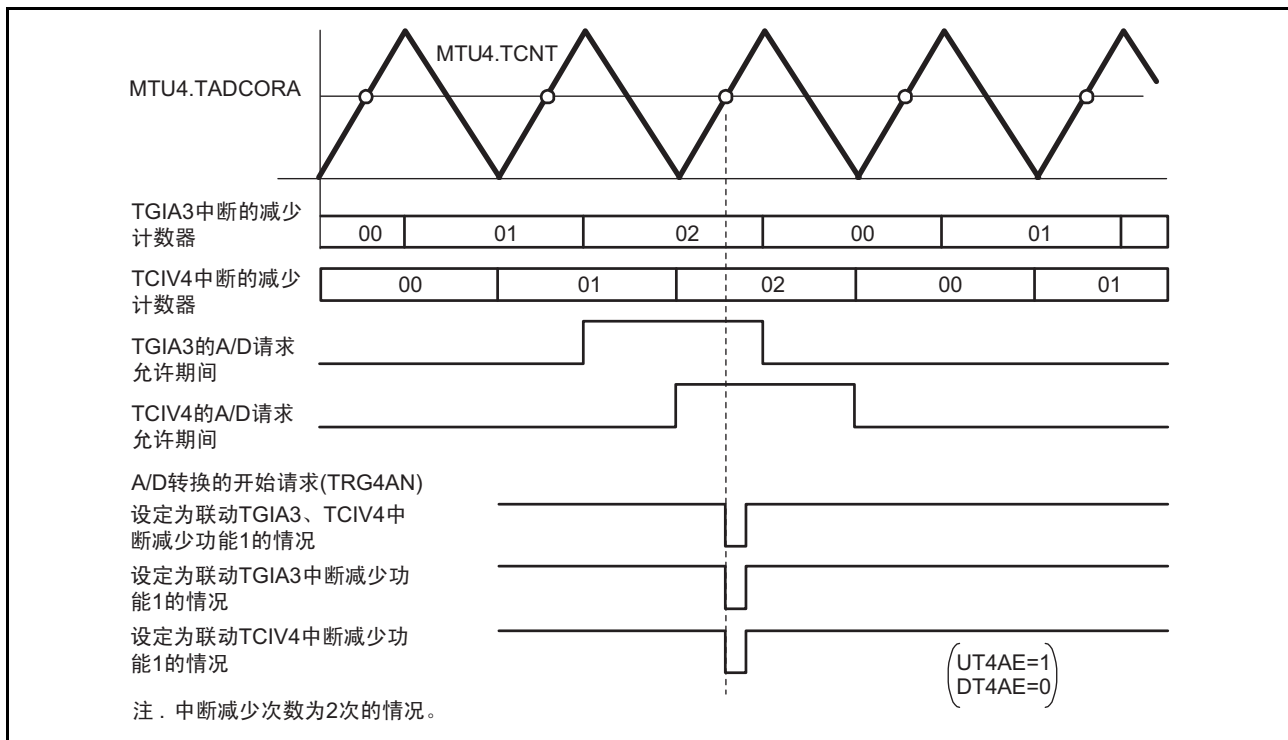


图 16.87 联动中断减少功能时的 A/D 转换开始请求信号 (TRG4AN) 的运行例子 (UT4AE=1、DT4AE=0)

#### (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 16.88 所示。

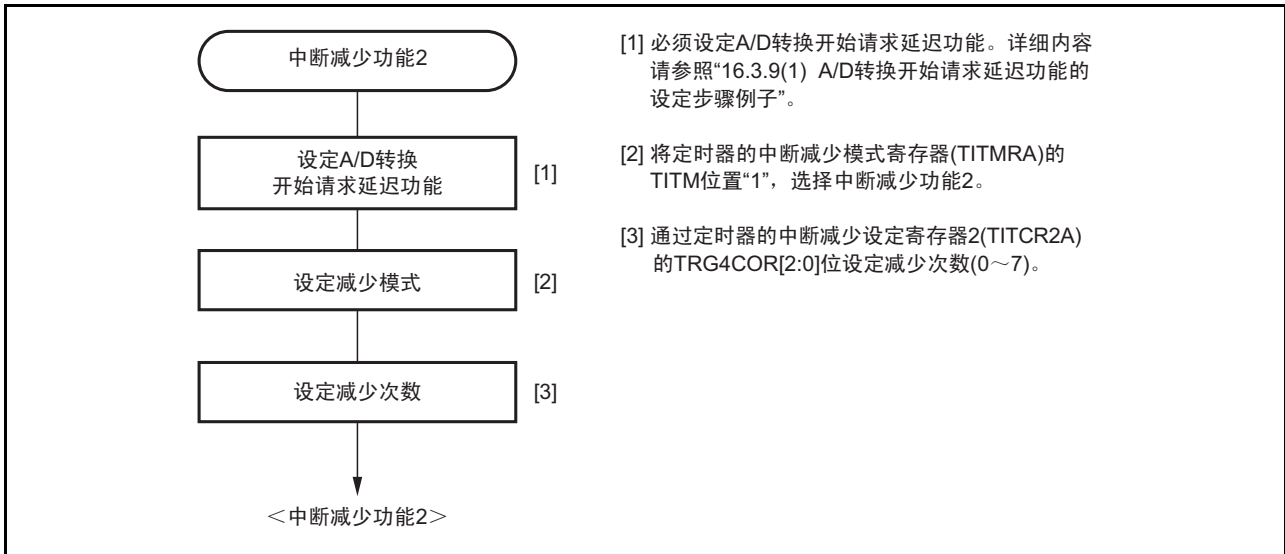


图 16.88 中断减少功能 2 的设定步骤例子

(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 16.89 所示。

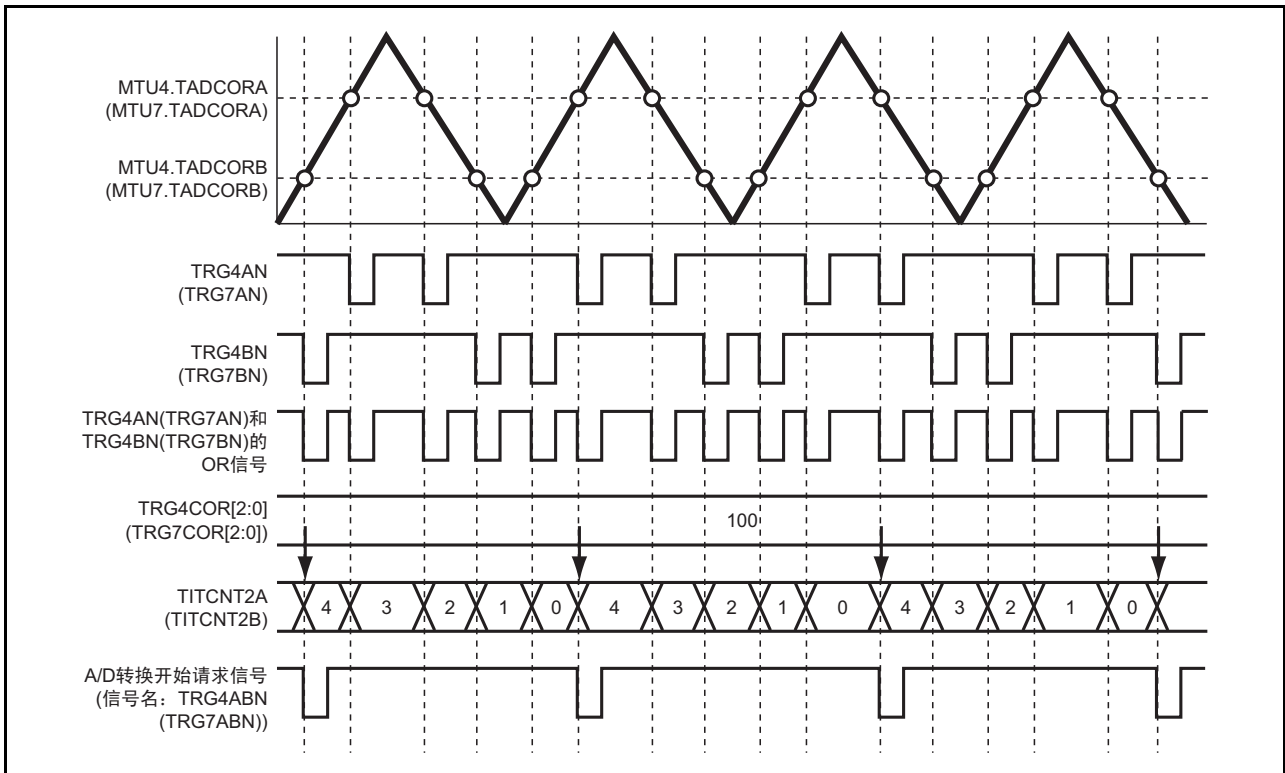


图 16.89 中断减少功能 2 的运行例子 (减少次数为 4 次的情况)

16.3.10 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 16.90 所示。

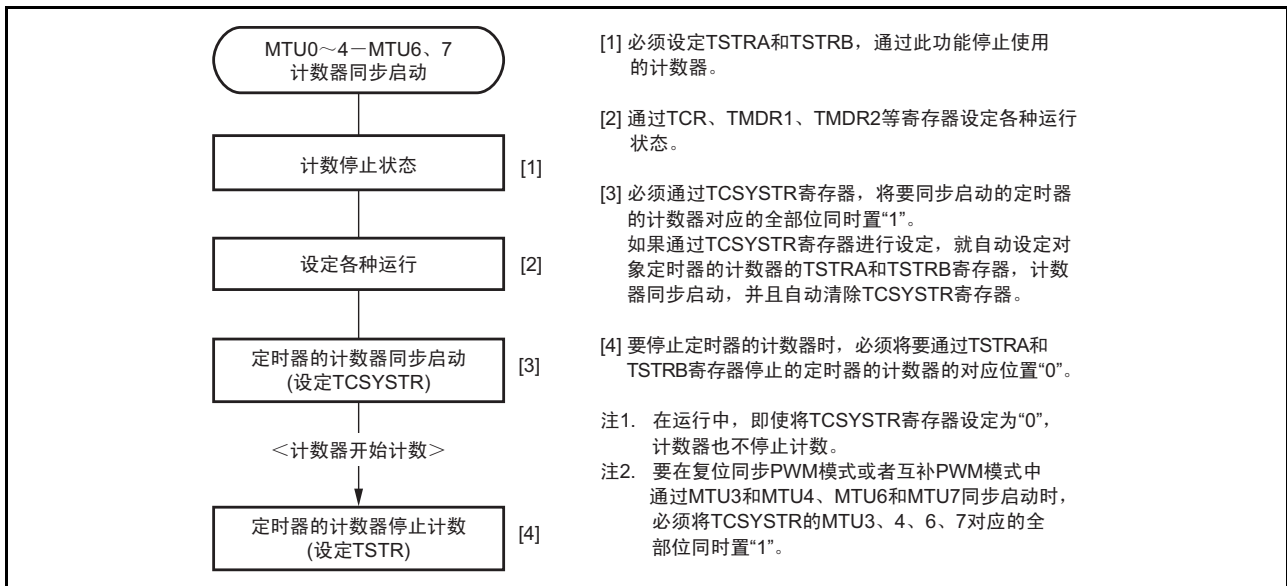


图 16.90 计数器同步启动的设定步骤例子

(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 16.91 所示。

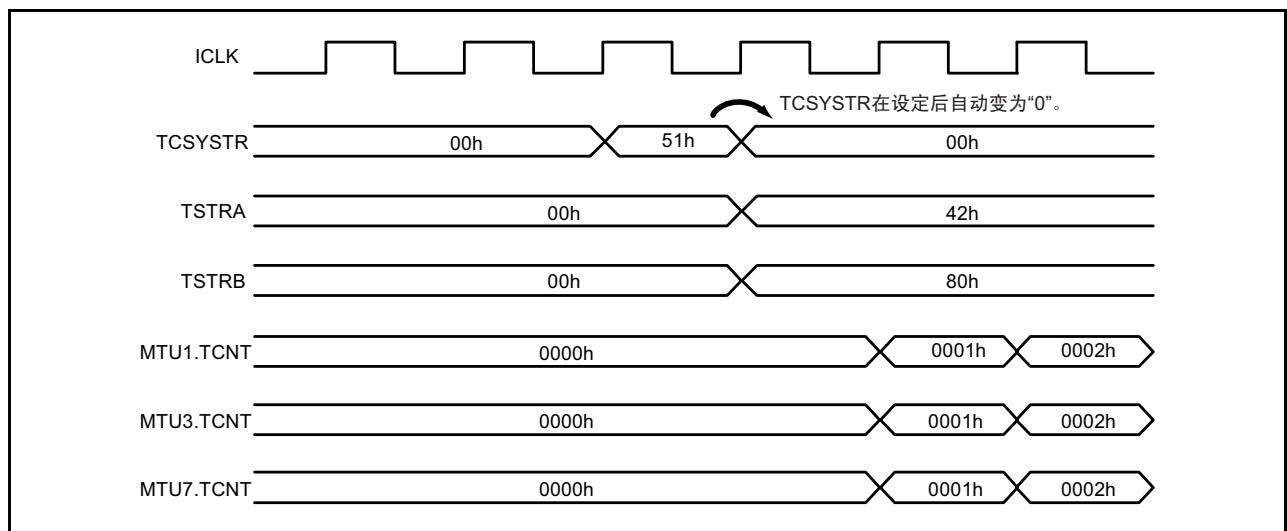


图 16.91 计数器同步启动的运行例子

(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 16.92 所示。

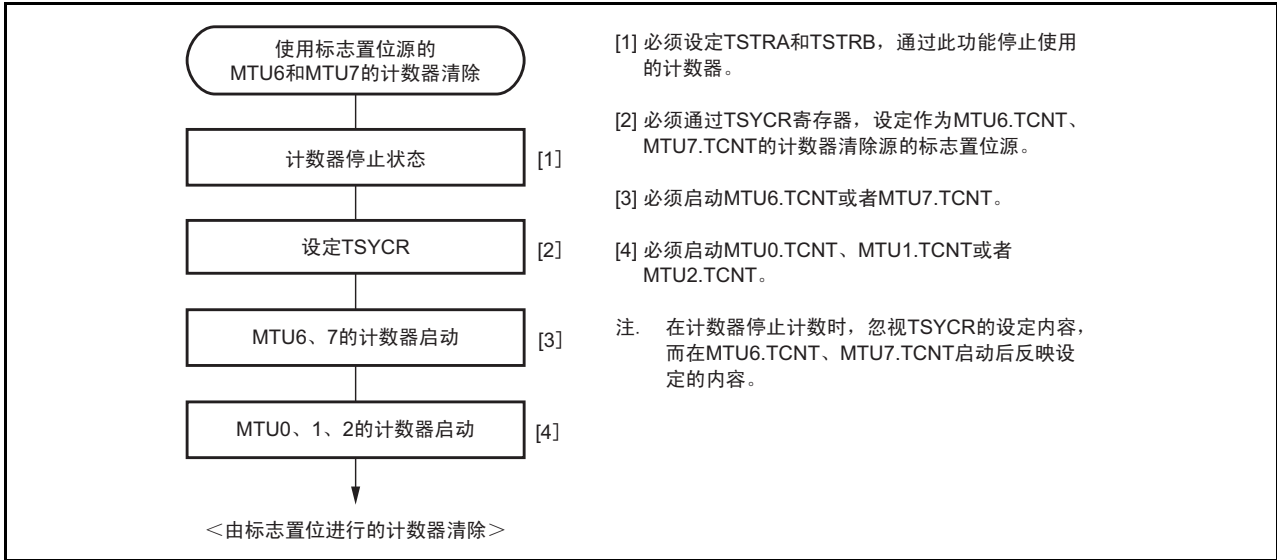


图 16.92 使用标志置位源的 MTU6 和 MTU7 的计数器清除的设定步骤例子

(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 16.93 和图 16.94 所示。

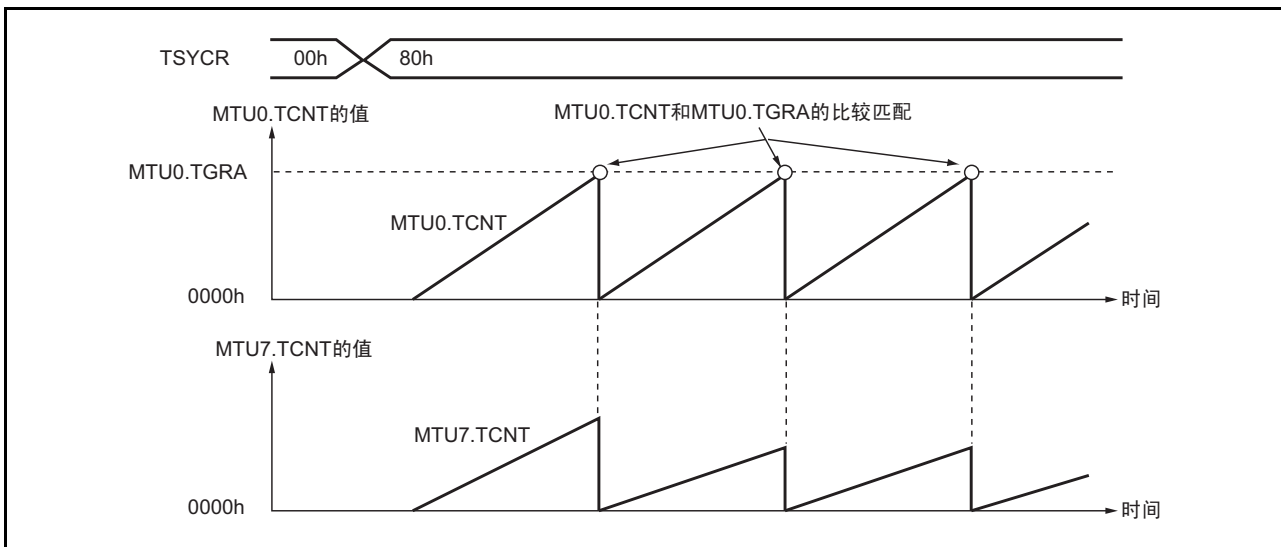


图 16.93 使用标志置位源的 MTU6 和 MTU7 的计数器清除的运行例子 (1)



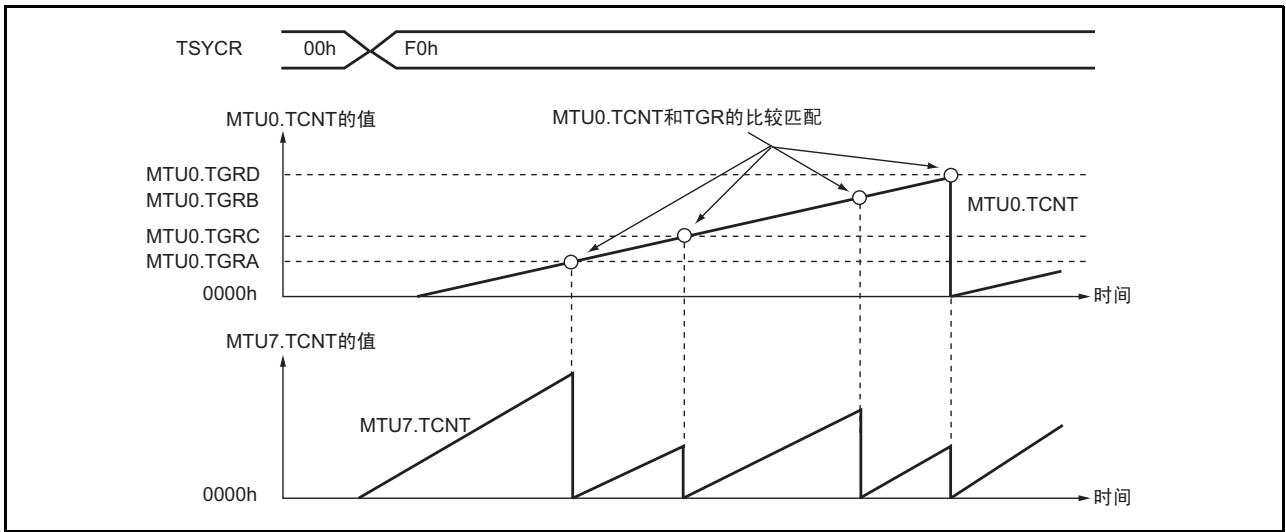


图 16.94 使用标志置位源的 MTU6 和 MTU7 的计数器清除的运行例子 (2)

### 16.3.11 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

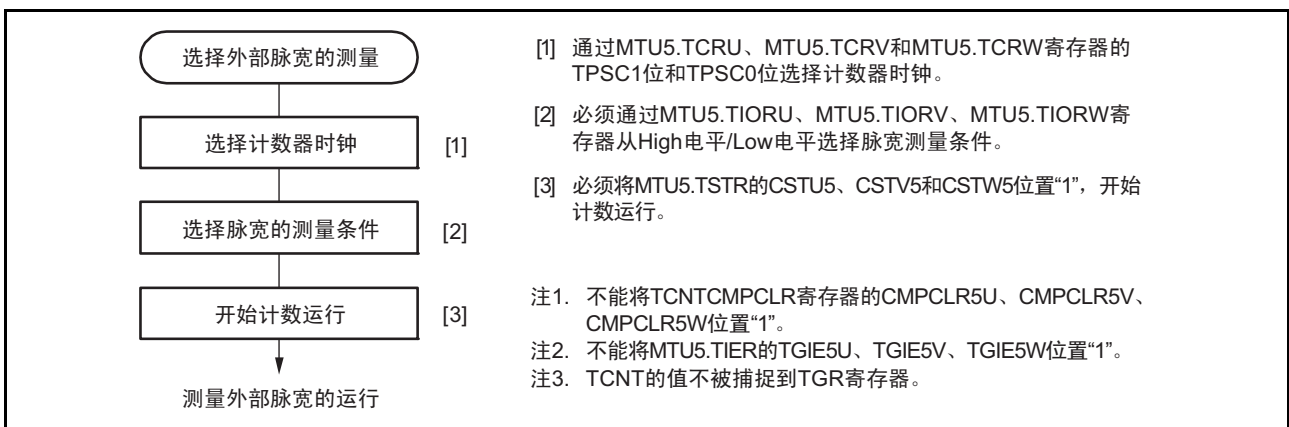


图 16.95 测量外部脉宽的设定步骤例子

#### (2) 测量外部脉宽的运行例子

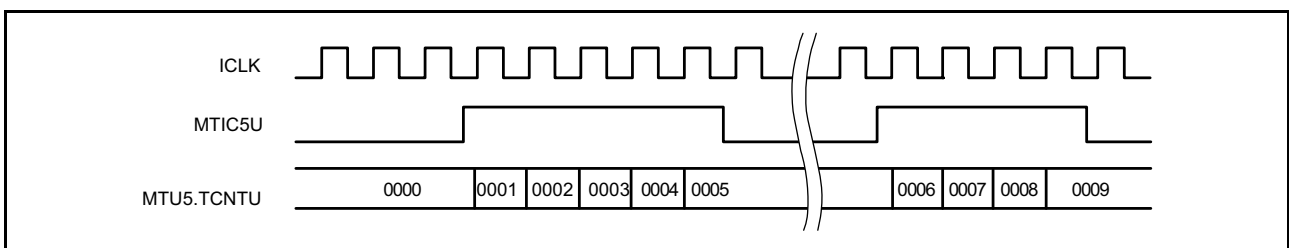


图 16.96 测量外部脉宽的运行例子 (测量 High 电平的脉宽)

### 16.3.12 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。

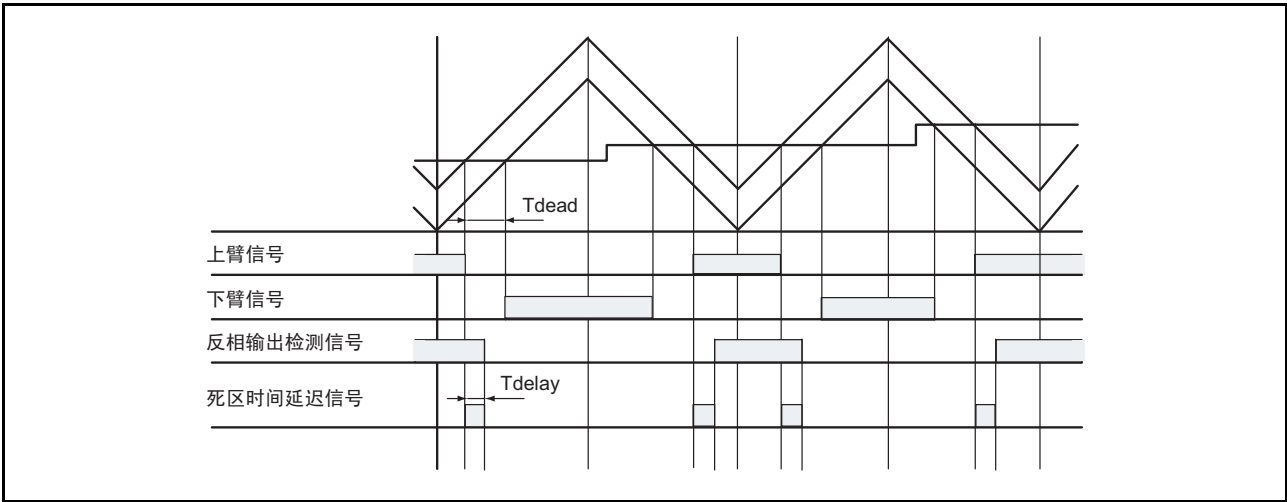


图 16.97 互补 PWM 运行时的死区时间延迟

#### (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 16.98 所示。

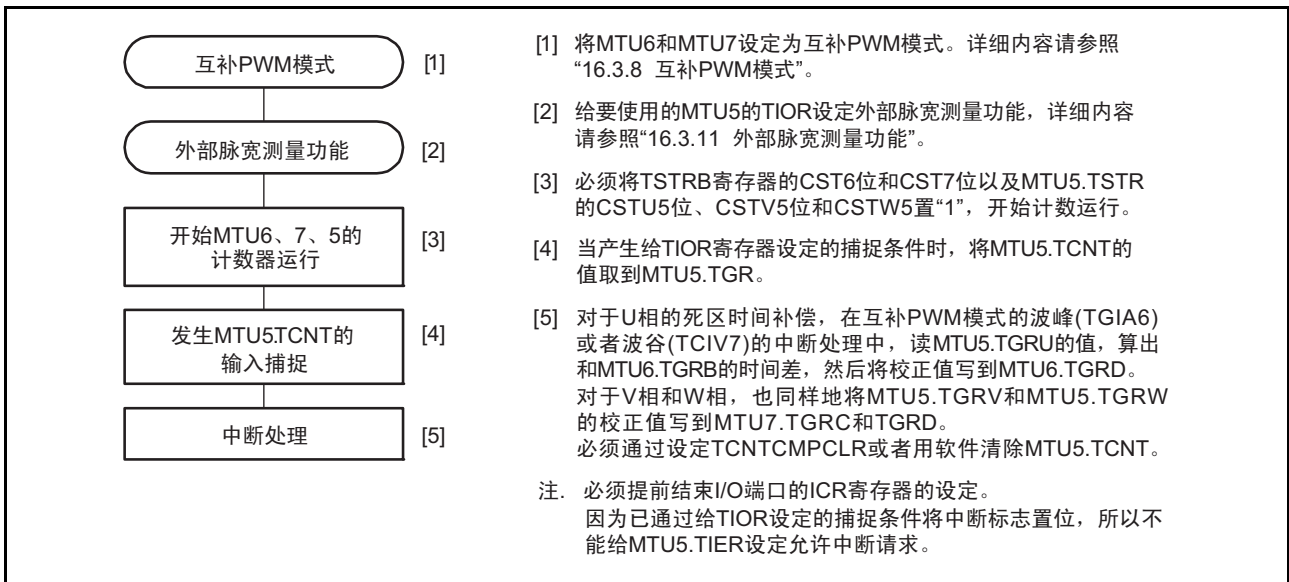


图 16.98 死区时间补偿功能的设定步骤例子

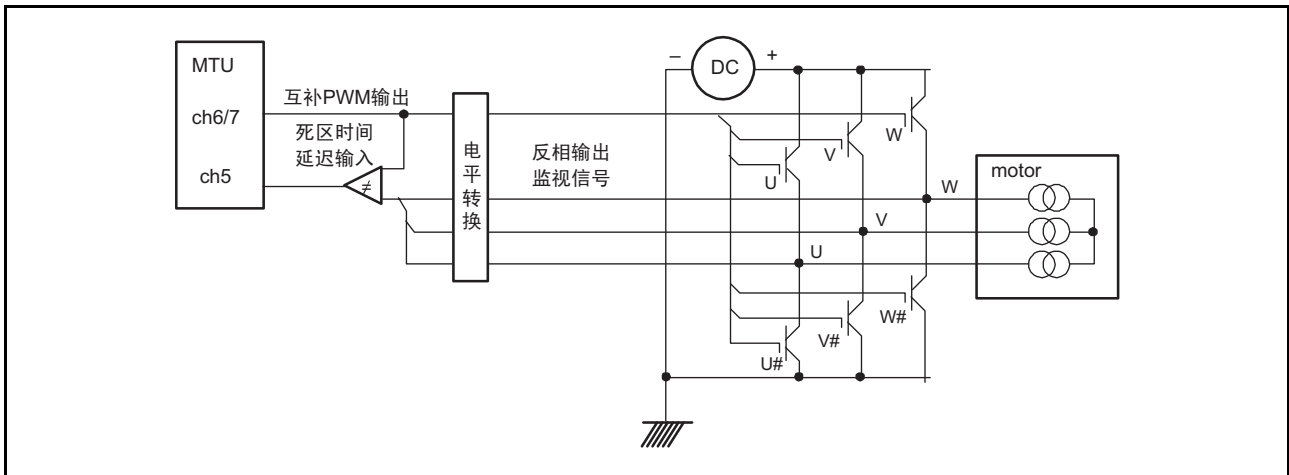


图 16.99 马达控制电路的结构例子

### 16.3.13 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 16.100 所示。

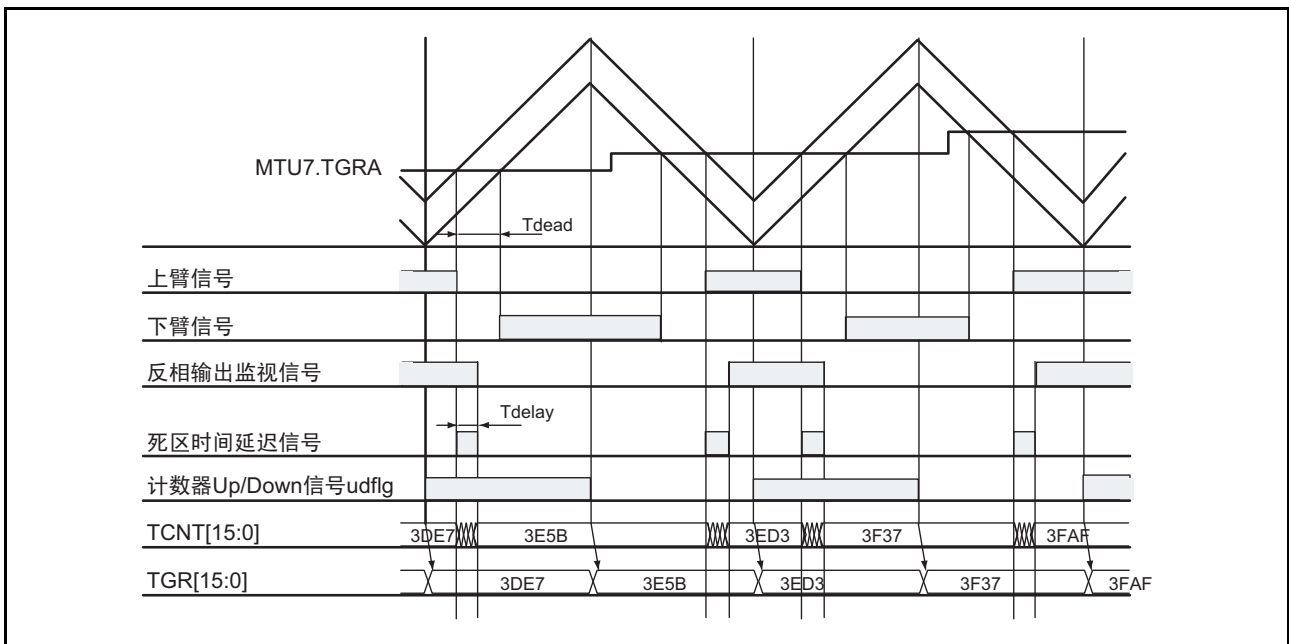


图 16.100 在互补 PWM 的“波峰 / 波谷”进行的 TCNT 捕捉运行

## 16.4 中断源

### 16.4.1 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“11. 中断控制器 (ICU)”。MTU 的中断源一览表如表 16.73 所示。

表 16.73 MTU 中断源

通道	名称	中断源	DTC 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。

## 16.4.2 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“14. 数据传送控制器 (DTC)”。

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”。

## 16.4.3 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 16.74 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“16.3.9 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 16.74 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。

## 16.5 运行时序

### 16.5.1 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 16.101 和图 16.102 所示，外部时钟运行（正常模式）时的 TCNT 计数时序如图 16.103，外部时钟运行（相位计数模式）的 TCNT 计数时序如图 16.104 所示。

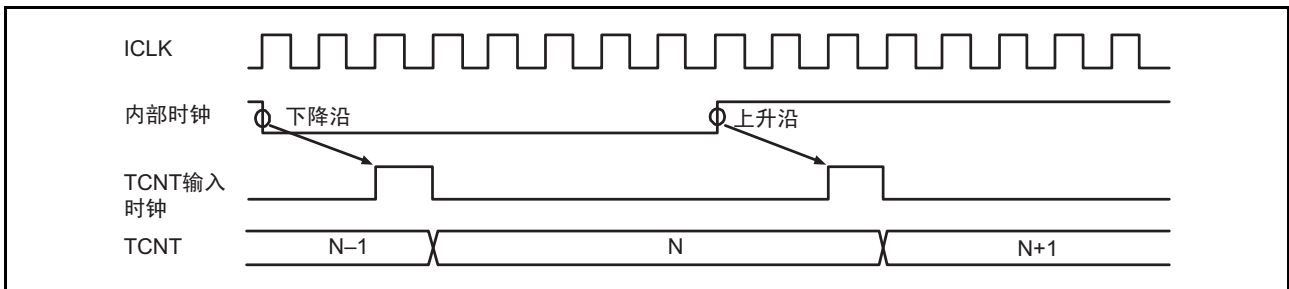


图 16.101 内部时钟运行时的计数时序 (MTU0 ~ 4、6、7)

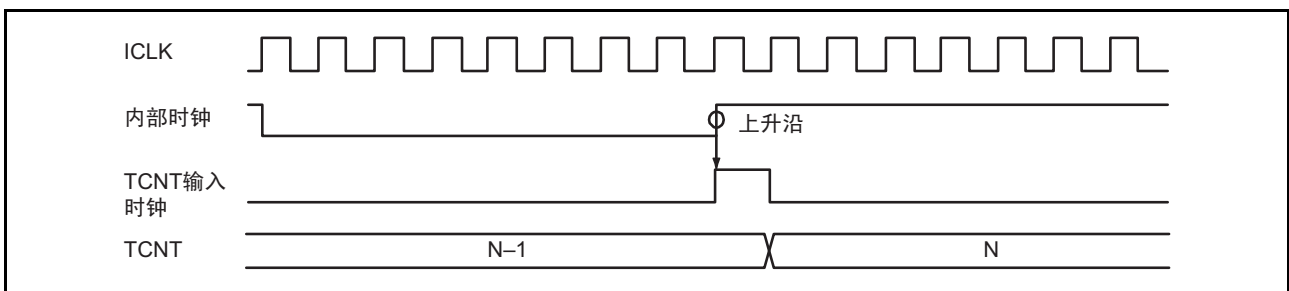


图 16.102 内部时钟运行时的计数时序 (MTU5)

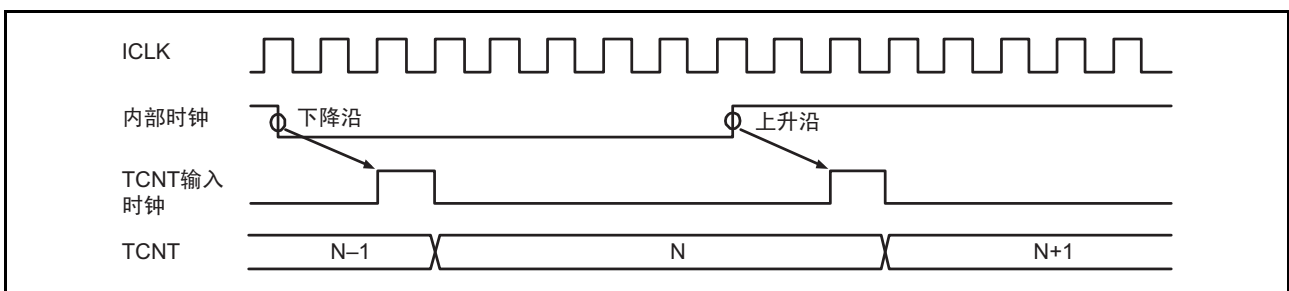


图 16.103 外部时钟运行时的计数时序 (MTU0 ~ 4)

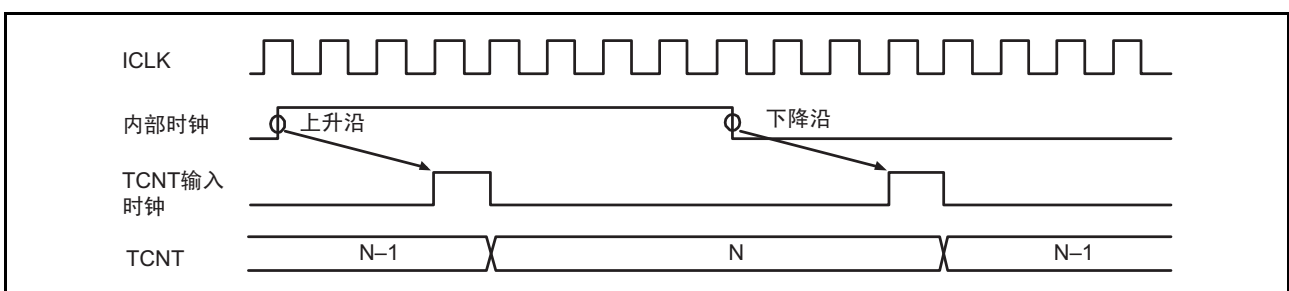


图 16.104 外部时钟运行时的计数时序 (相位计数模式)



## (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 16.105、输出比较的输出时序（互补 PWM 模式、复位同步 PWM 模式）如图 16.106 所示。

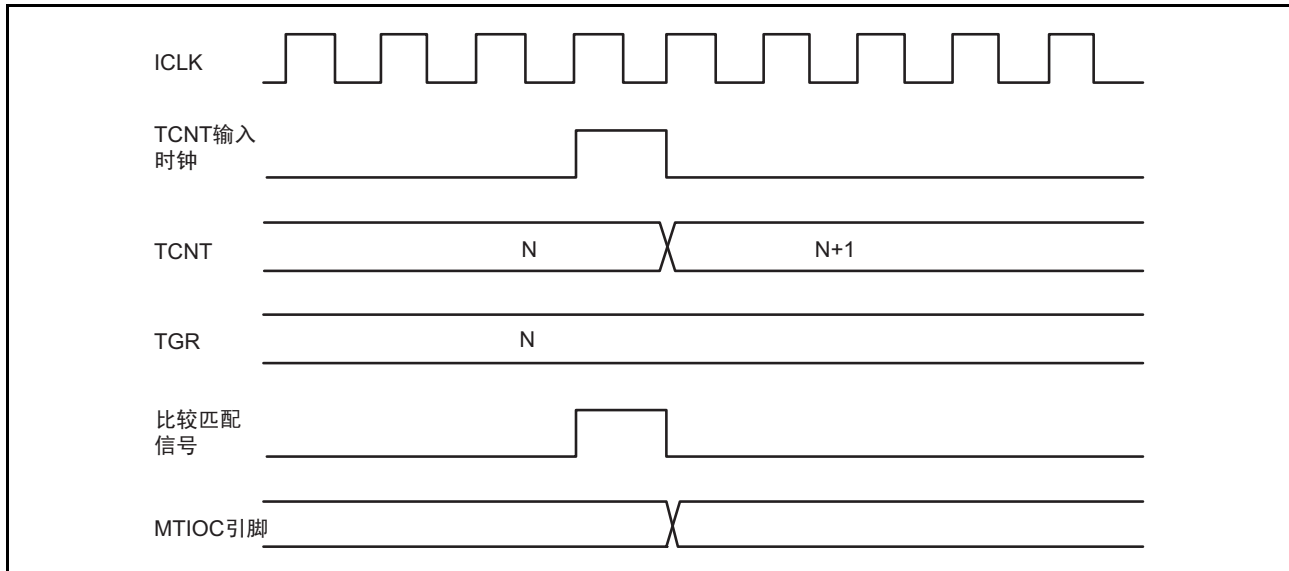


图 16.105 输出比较的输出时序（正常模式、PWM 模式）

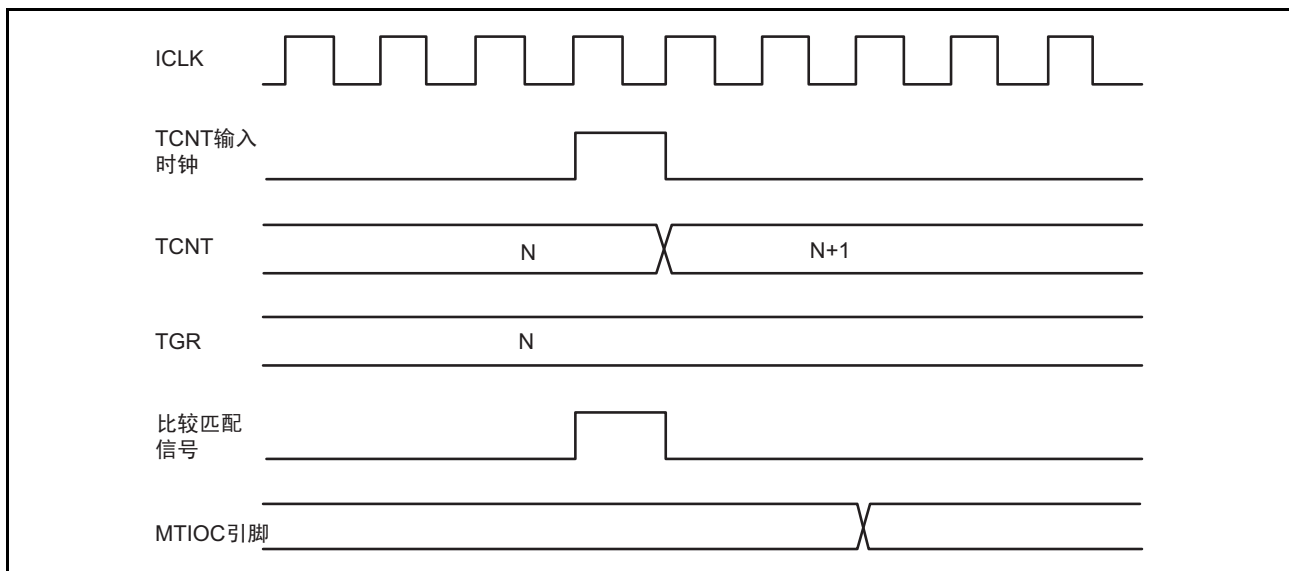


图 16.106 输出比较的输出时序（互补 PWM 模式、复位同步 PWM 模式）

(3) 输入捕捉信号的时序

输入捕捉的时序如图 16.107 所示。

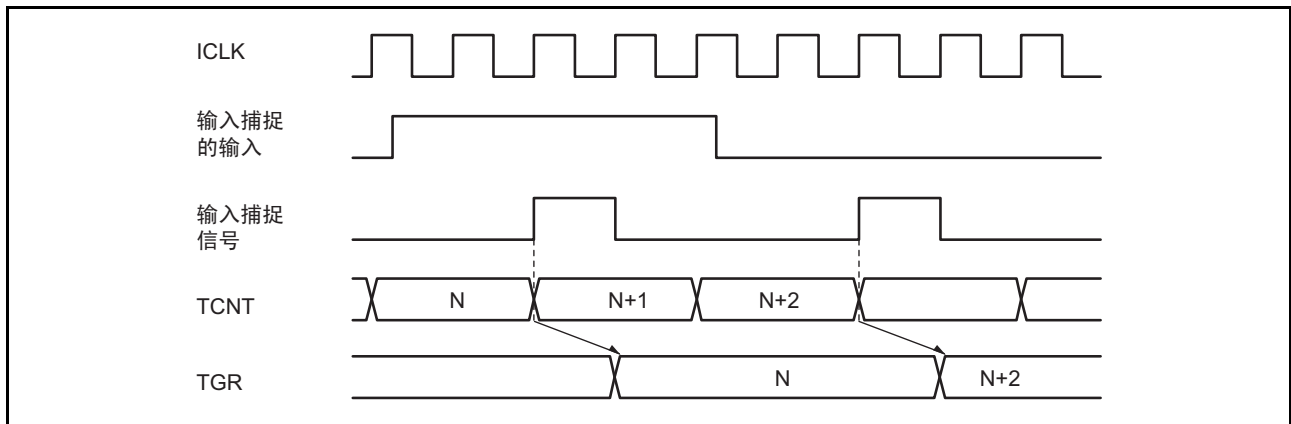


图 16.107 输入捕捉的输入信号时序

(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 16.108 和图 16.109 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 16.110 所示。

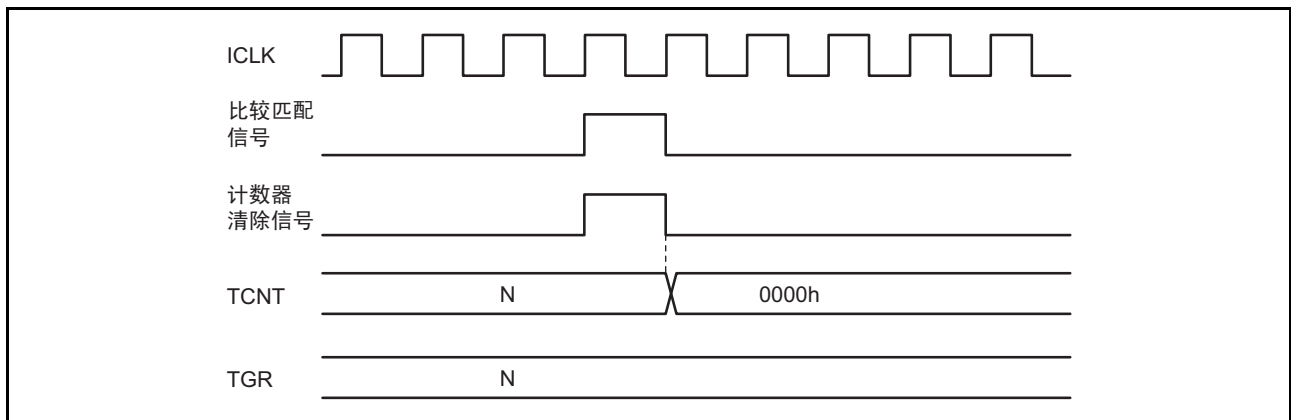


图 16.108 计数器清除时序 (比较匹配) (MTU0 ~ 4、6、7)

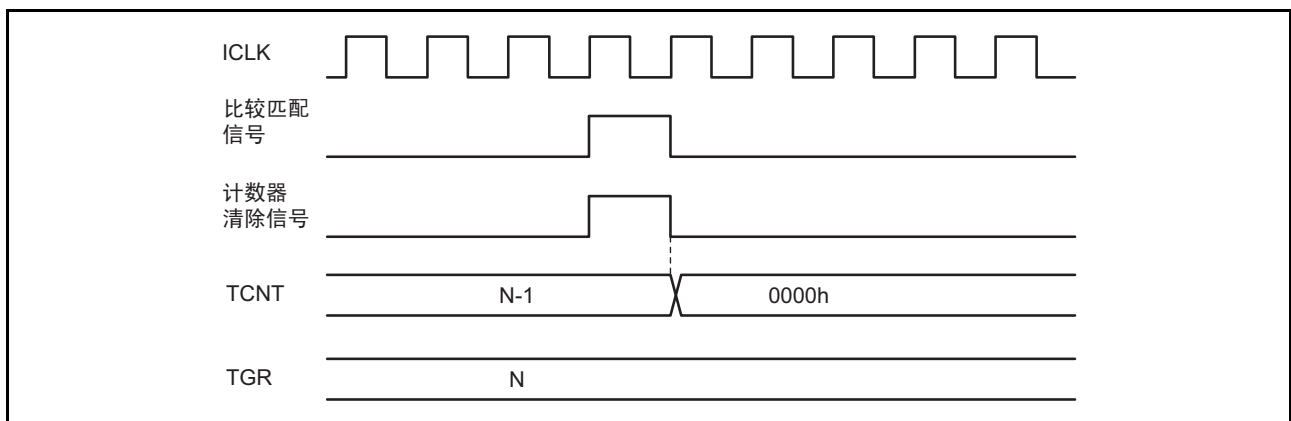


图 16.109 计数器清除时序 (比较匹配) (MTU5)

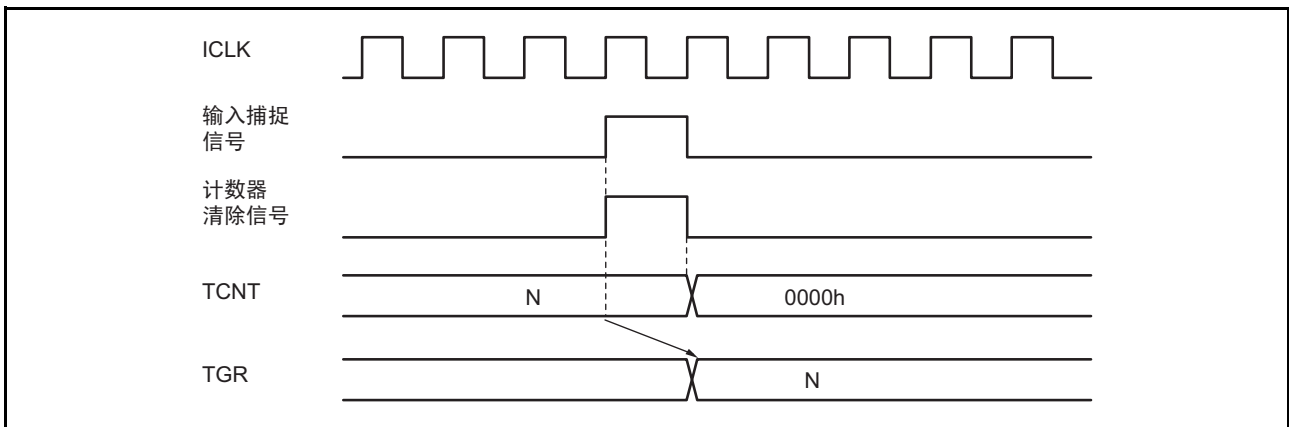


图 16.110 计数器清除时序 (输入捕捉) (MTU0 ~ 7)

(5) 缓冲运行的时序

缓冲运行的时序如图 16.111 ~ 图 16.113 所示。

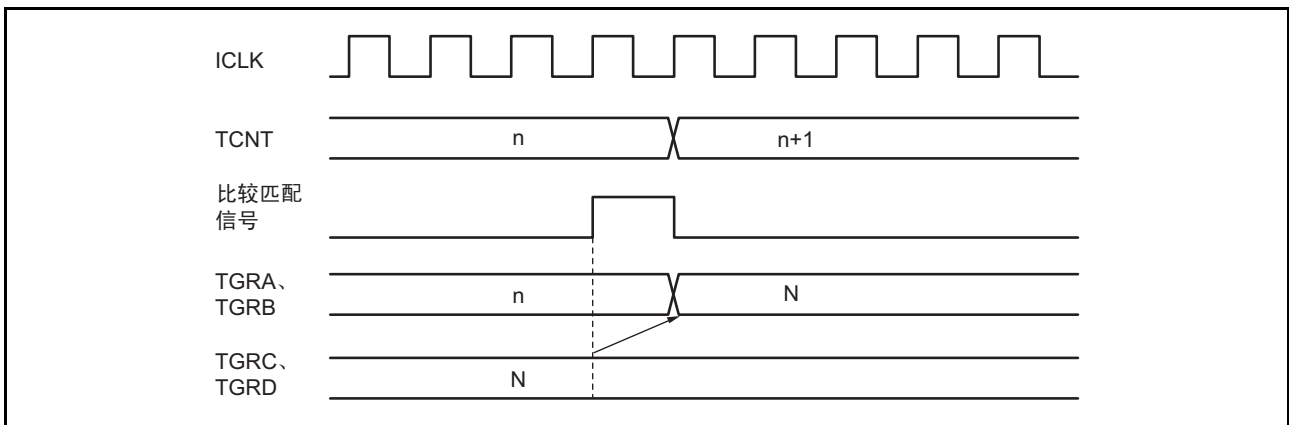


图 16.111 缓冲运行的时序 (比较匹配)

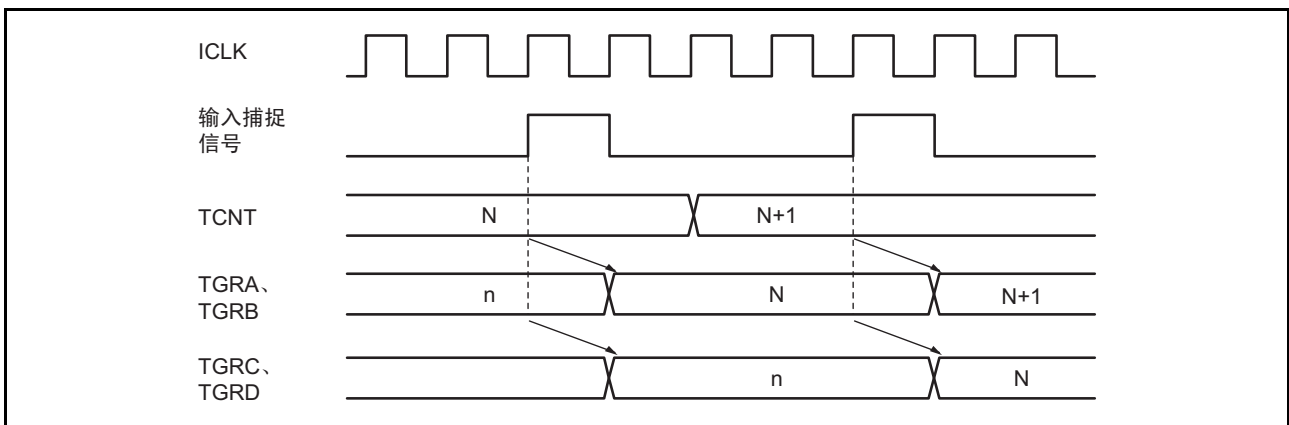


图 16.112 缓冲运行的时序 (输入捕捉)

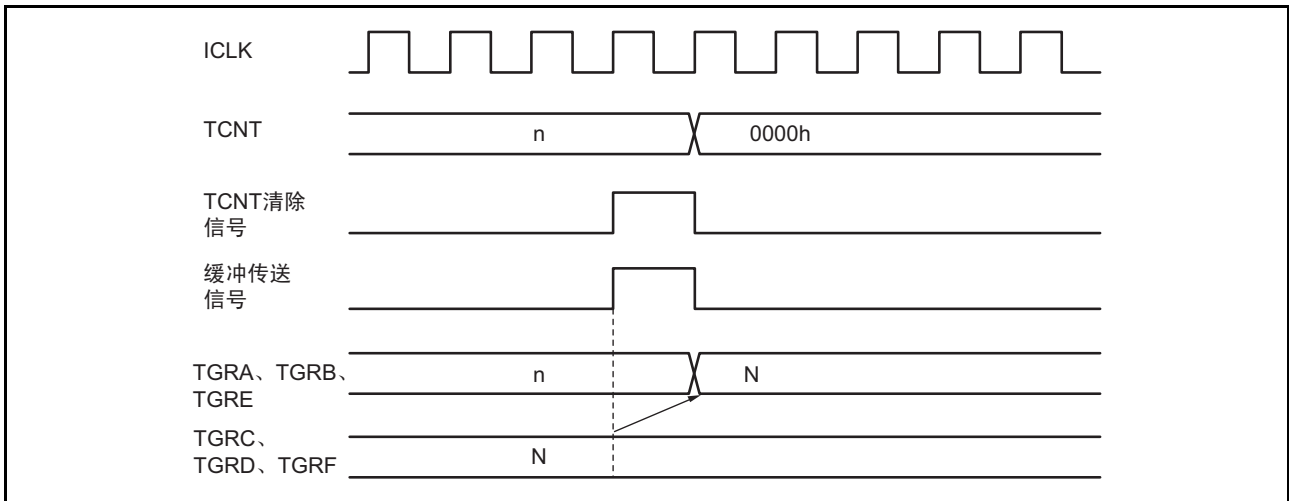


图 16.113 缓冲运行的时序 (清除 TCNT 的情况)

(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 16.114 ~ 图 16.116 所示。

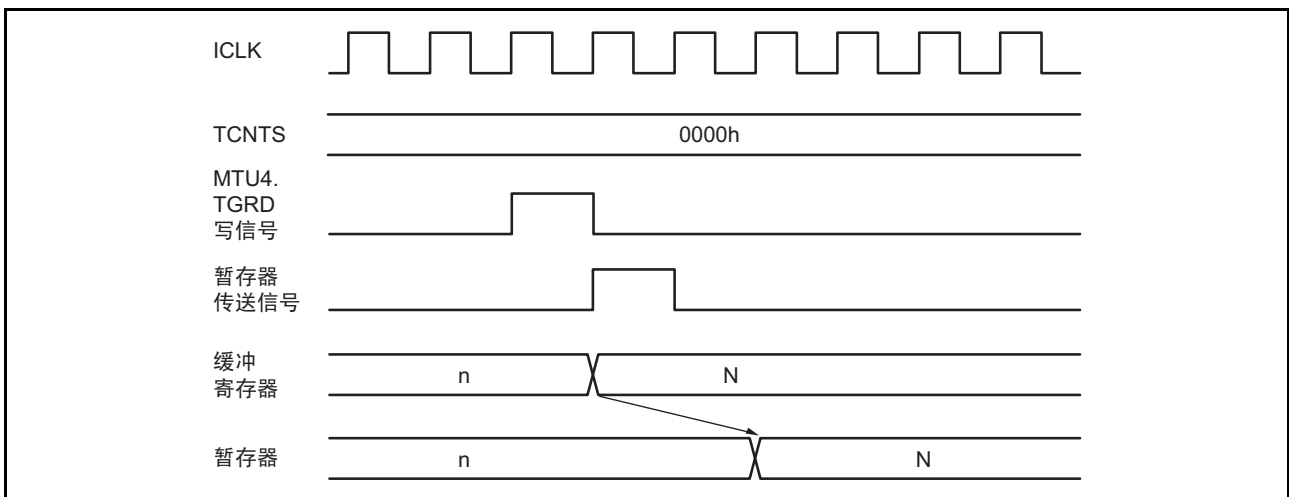


图 16.114 缓冲寄存器到暂存器的传送时序 (TCNTSA 停止)

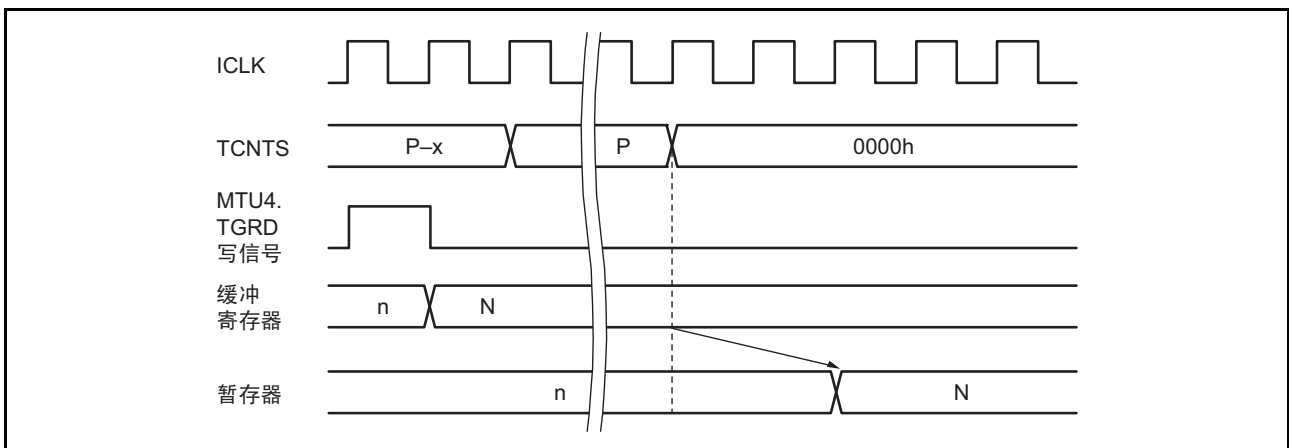


图 16.115 缓冲寄存器到暂存器的传送时序 (TCNTSA 正在运行)

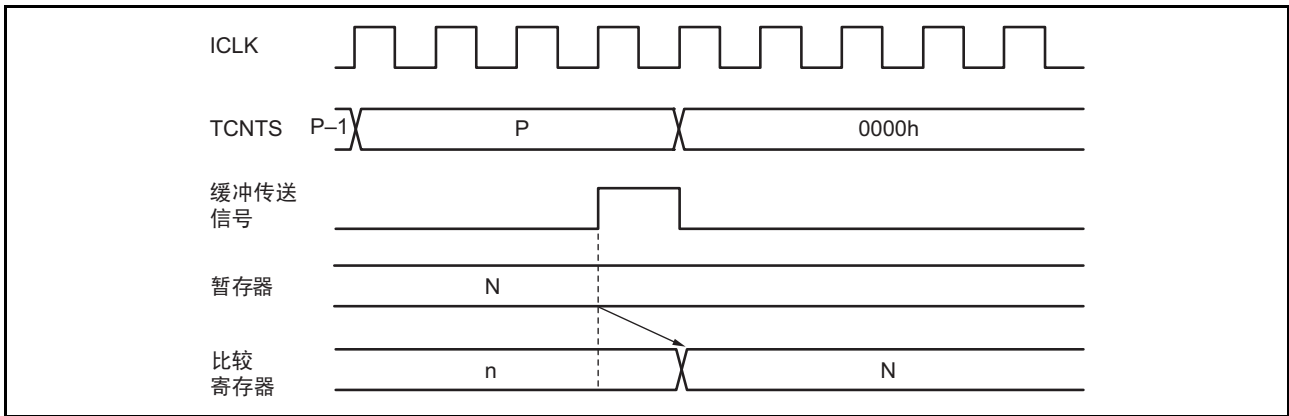


图 16.116 暂存器到比较寄存器的传送时序

### 16.5.2 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 16.117 和图 16.118 所示。

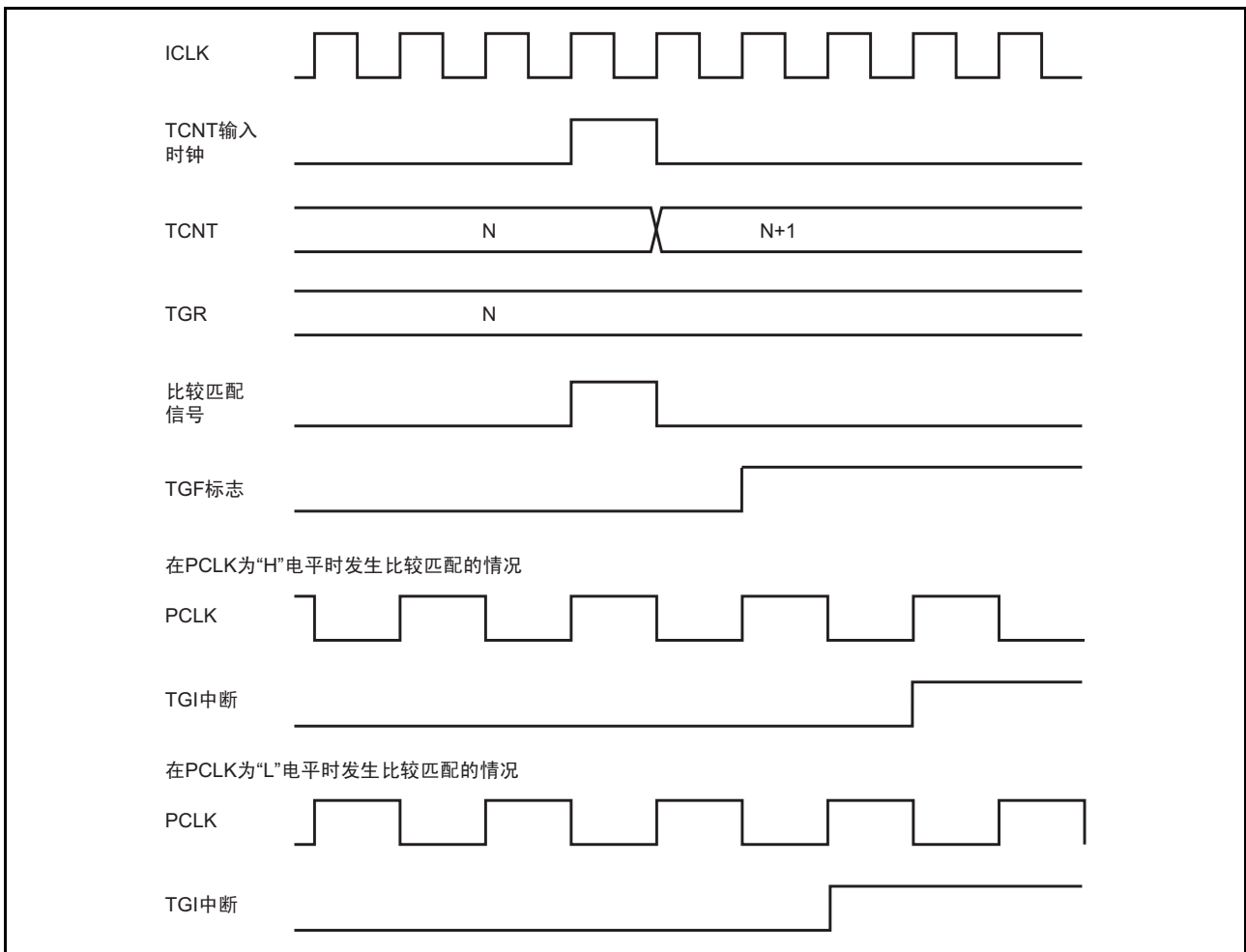


图 16.117 TGI 中断的时序 (比较匹配) (MTU0 ~ 4、6、7)

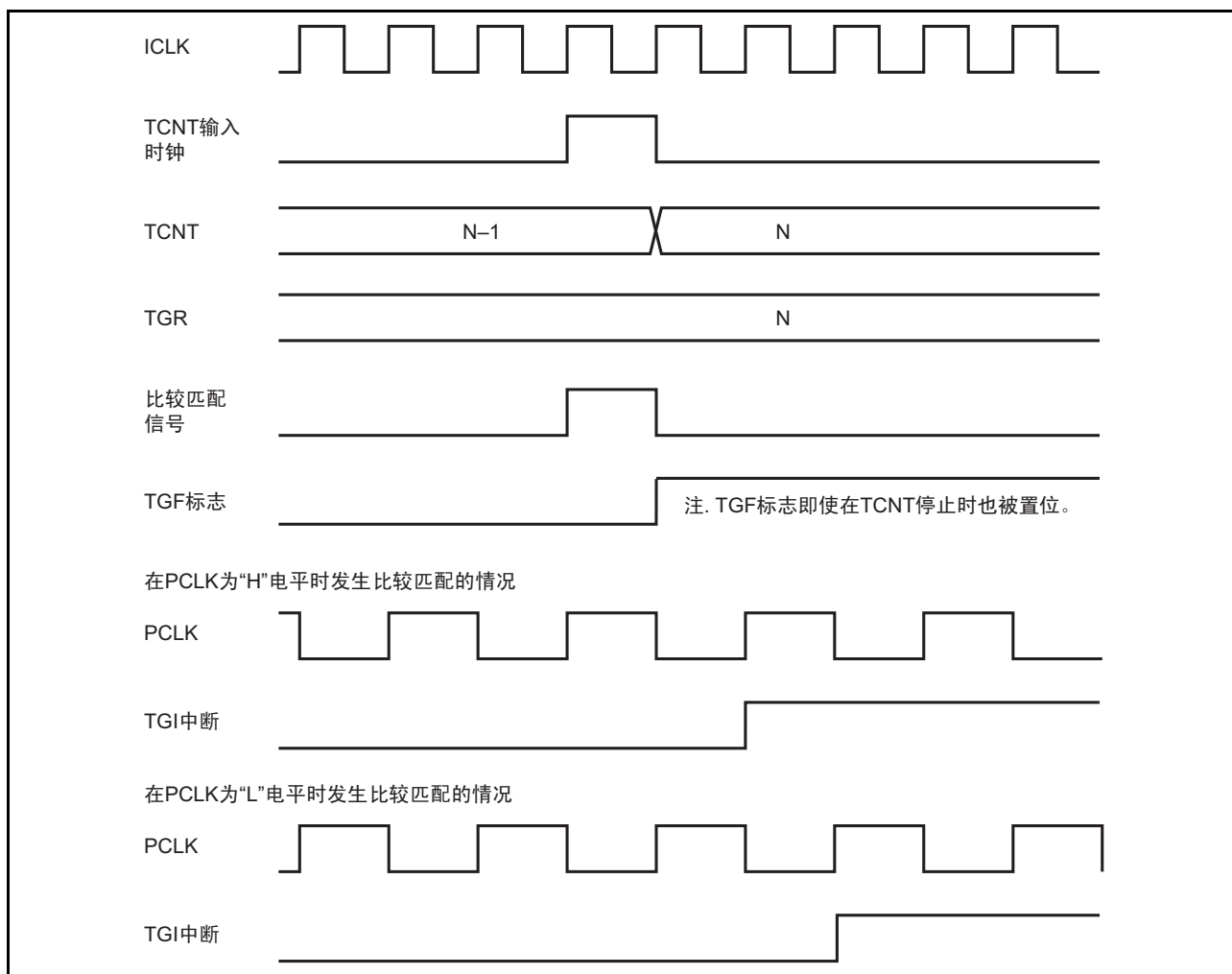


图 16.118 TGI 中断的时序 (比较匹配) (MTU5)

(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 16.119 和图 16.120 所示。

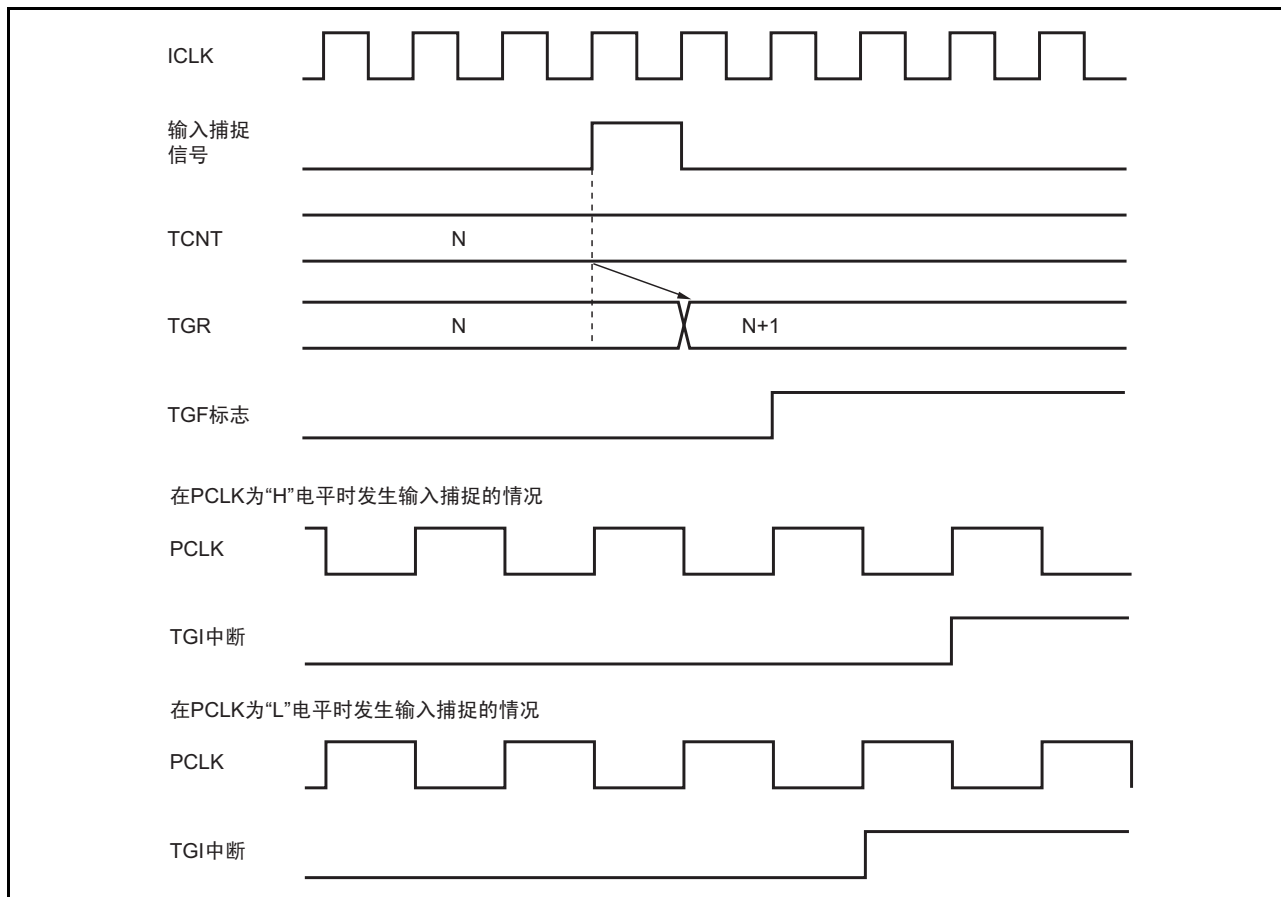


图 16.119 TGI 中断的时序 (输入捕捉) (MTU0 ~ 4、6、7)

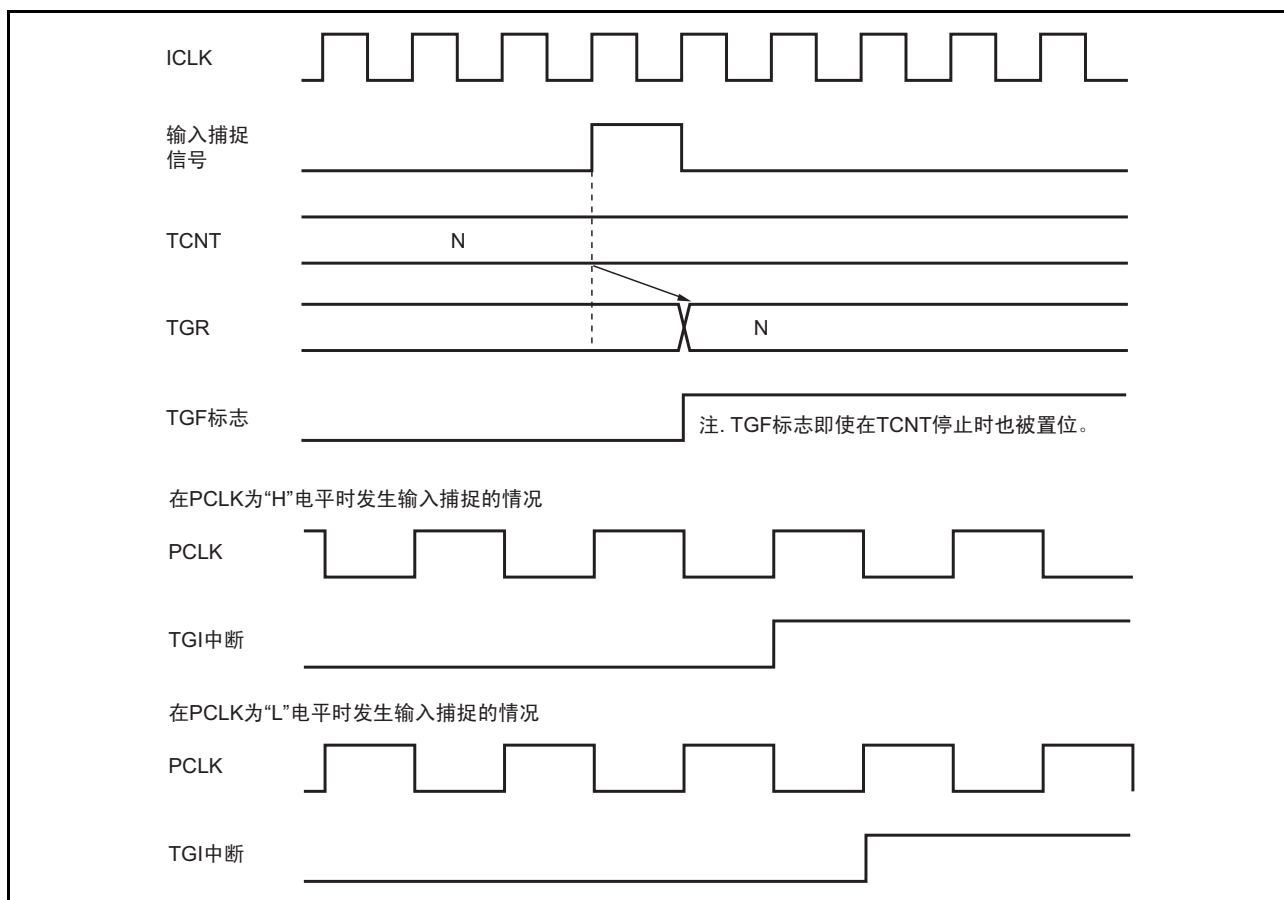


图 16.120 TGI 中断的时序 (输入捕捉) (MTU5)



(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 16.121 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 16.122 所示。

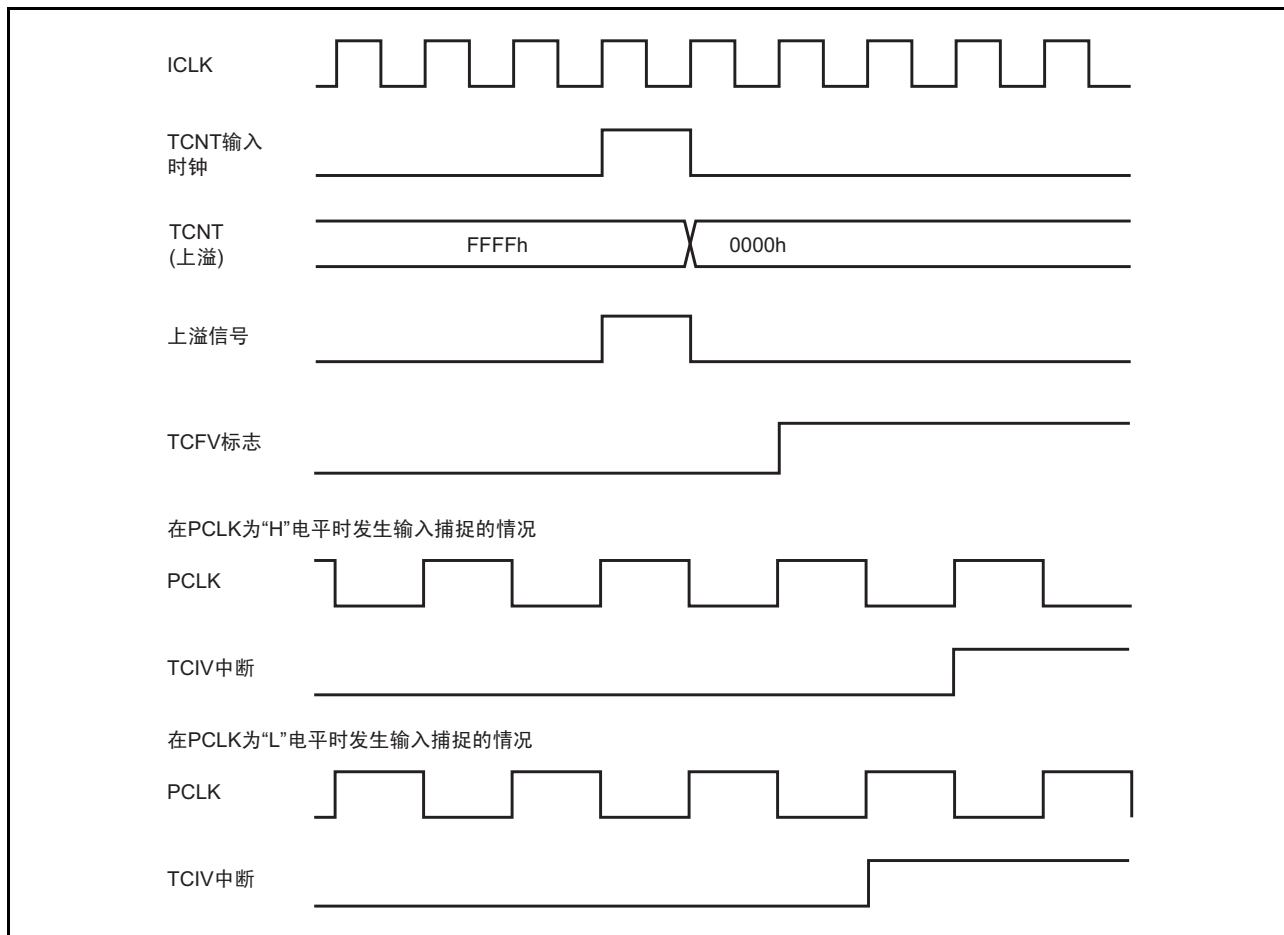


图 16.121 TCIV 中断的置位时序

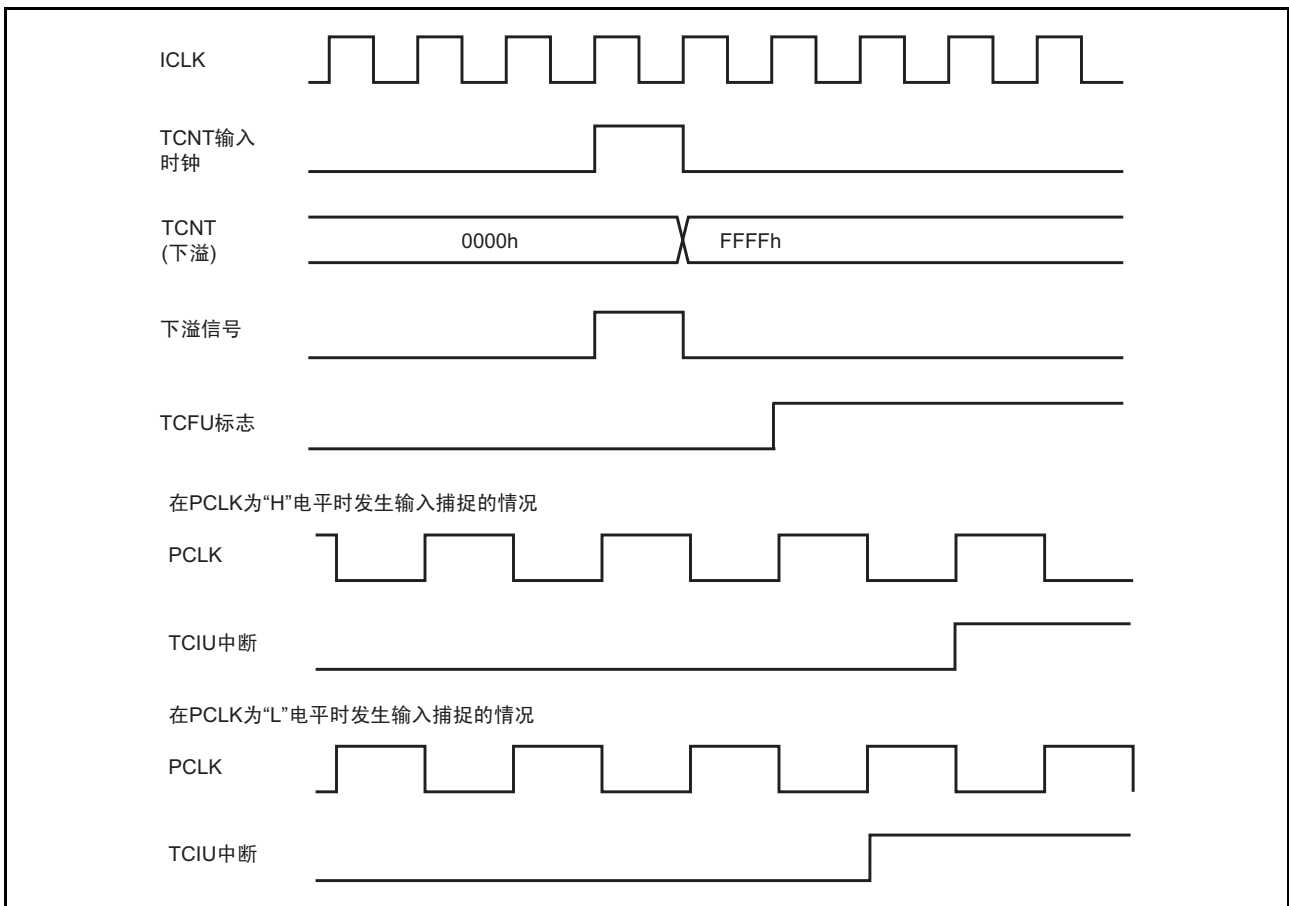


图 16.122 TCIU 中断的置位时序

(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 16.123 所示。

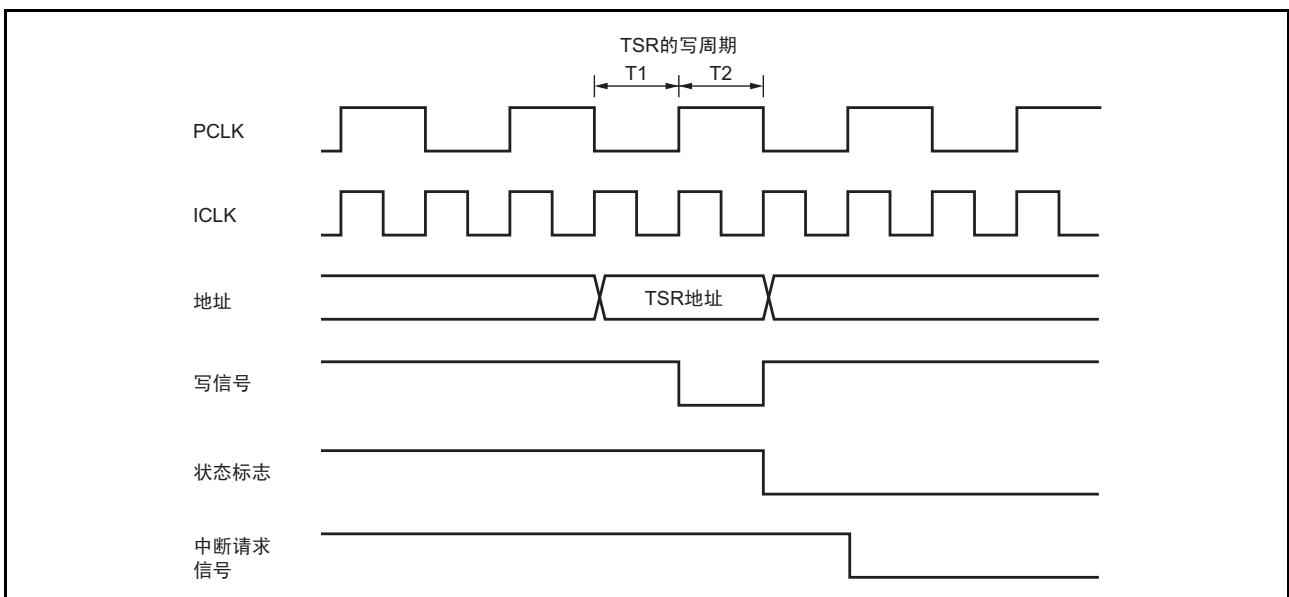


图 16.123 通过 CPU 进行的状态标志的清除时序 (MTU0 ~ 7)

## 16.6 使用时的注意事项

### 16.6.1 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“9. 低功耗功能”。

### 16.6.2 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 个状态；在双边沿的情况下，输入时钟的脉宽至少为 2.5 个状态。否则，就不能正常运行。

在相位计数模式中，2 个输入时钟的相位差和重叠都至少为 1.5 个状态，脉宽至少为 2.5 个状态。相位计数模式的输入时钟条件如图 16.124 所示。

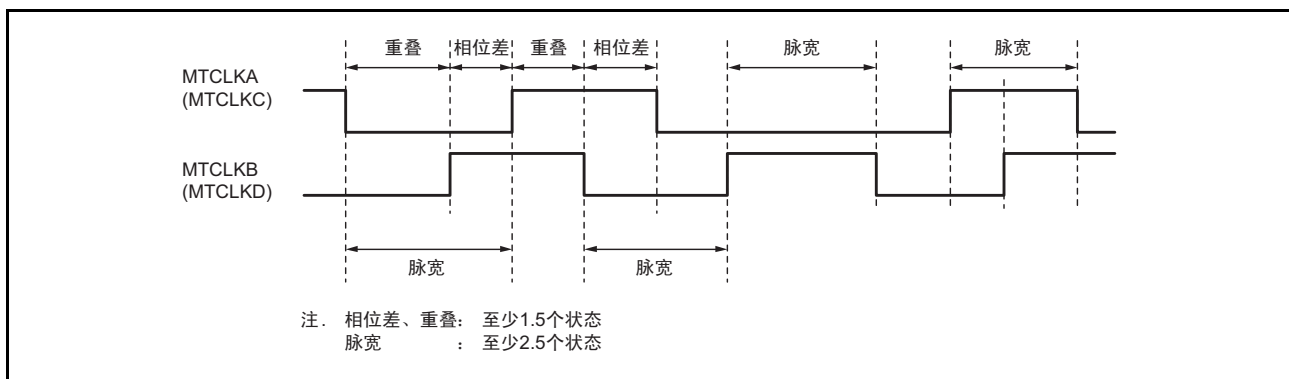


图 16.124 相位计数模式的相位差、重叠和脉宽

### 16.6.3 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

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

- MTU5的情况

$$f = \frac{ICLK}{N}$$

- f : 计数器的频率  
ICLK : MTU 时钟的工作频率  
N : TGR 的设定值

### 16.6.4 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 16.125 所示。

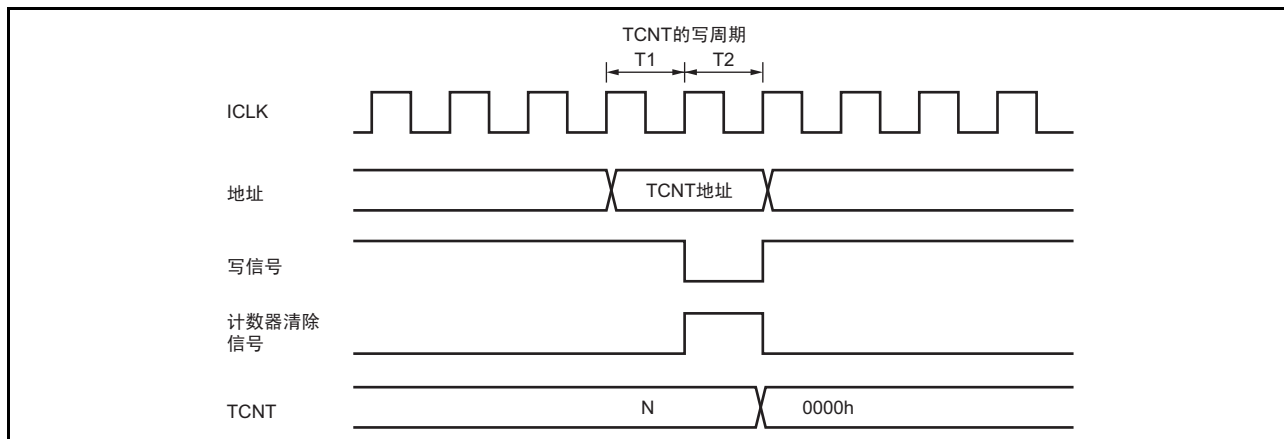


图 16.125 TCNT 的写和清除的竞争

### 16.6.5 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 16.126 所示。

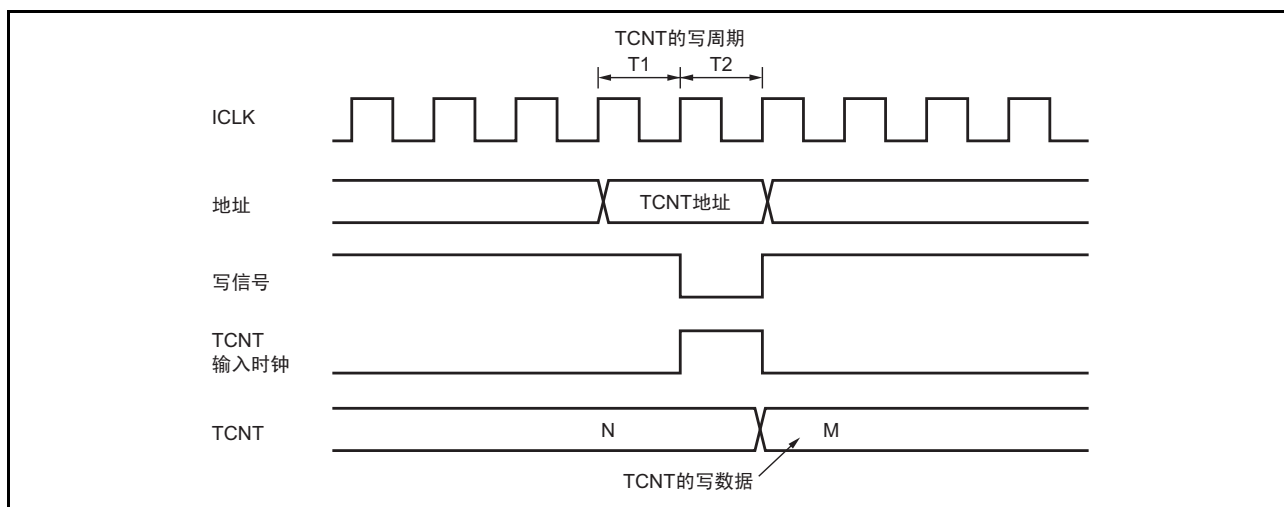


图 16.126 TCNT 的写和递增计数的竞争

### 16.6.6 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 16.127 所示。

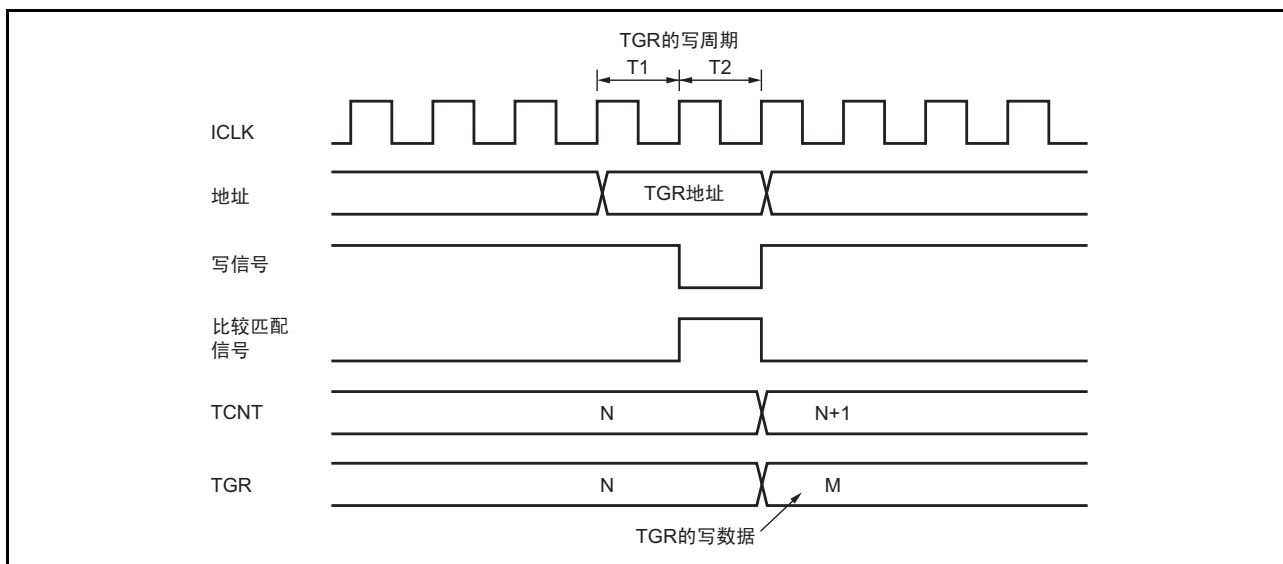


图 16.127 TGR 的写和比较匹配的竞争

### 16.6.7 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 16.128 所示。

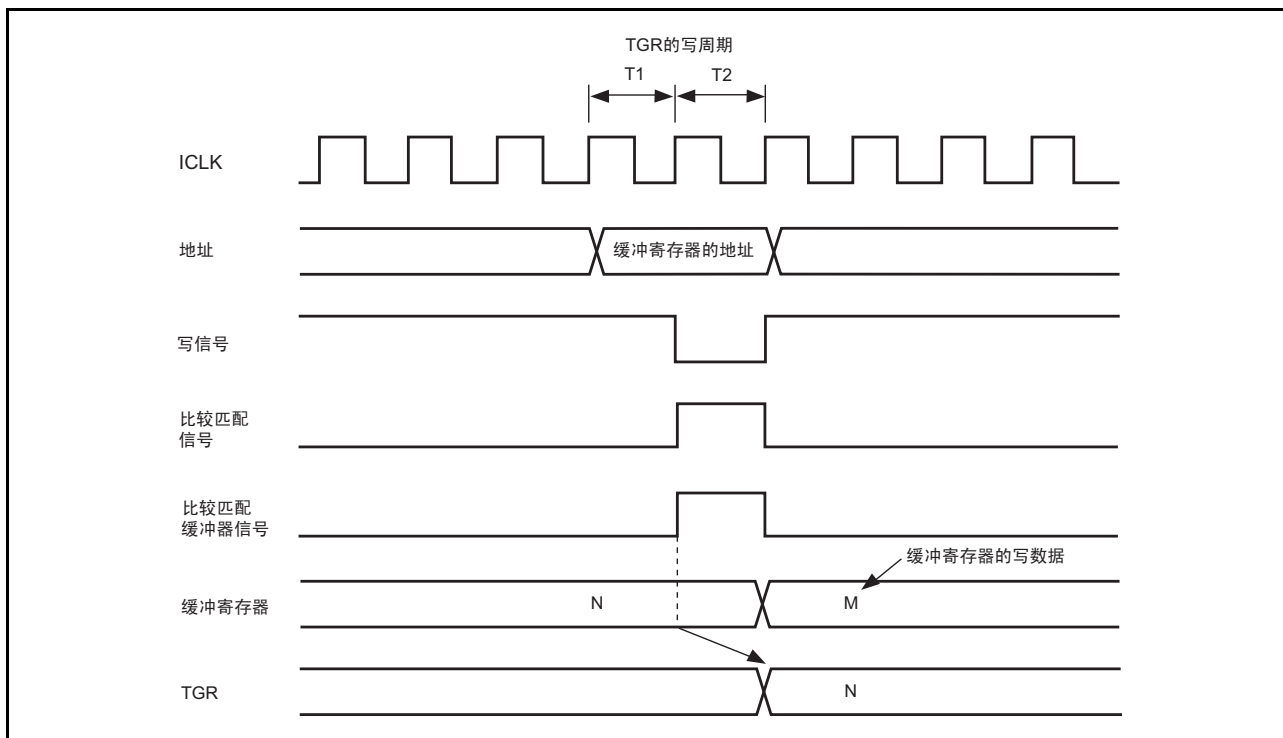


图 16.128 缓冲寄存器的写和比较匹配的竞争

### 16.6.8 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 16.129 所示。

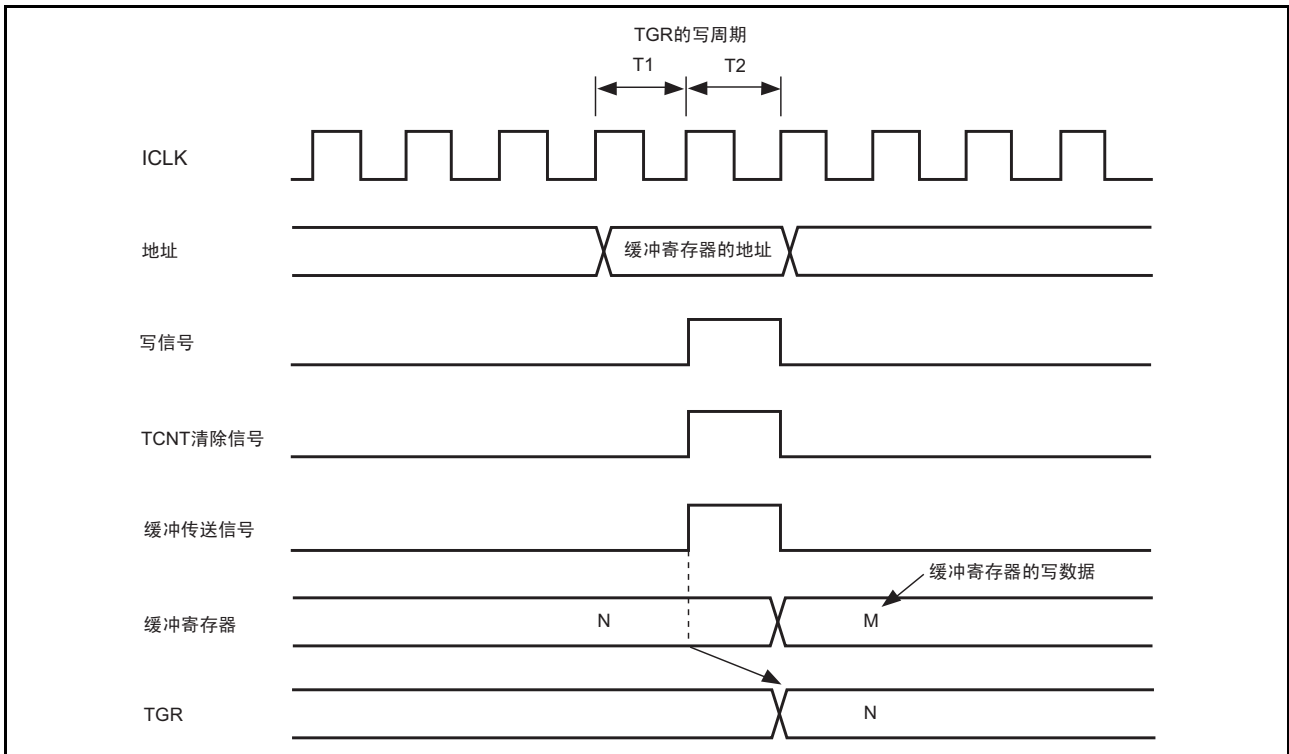


图 16.129 缓冲寄存器的写和 TCNT 清除的竞争

### 16.6.9 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 16.130 和图 16.131 所示。

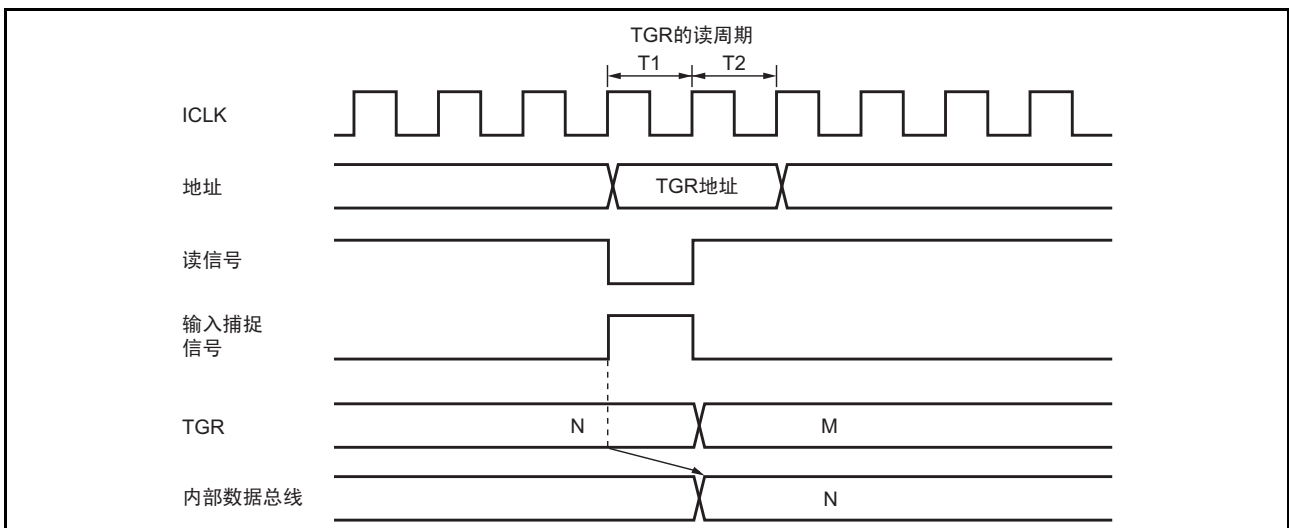


图 16.130 TGR 寄存器的读和输入捕捉的竞争 (MTU0 ~ 4、6、7)

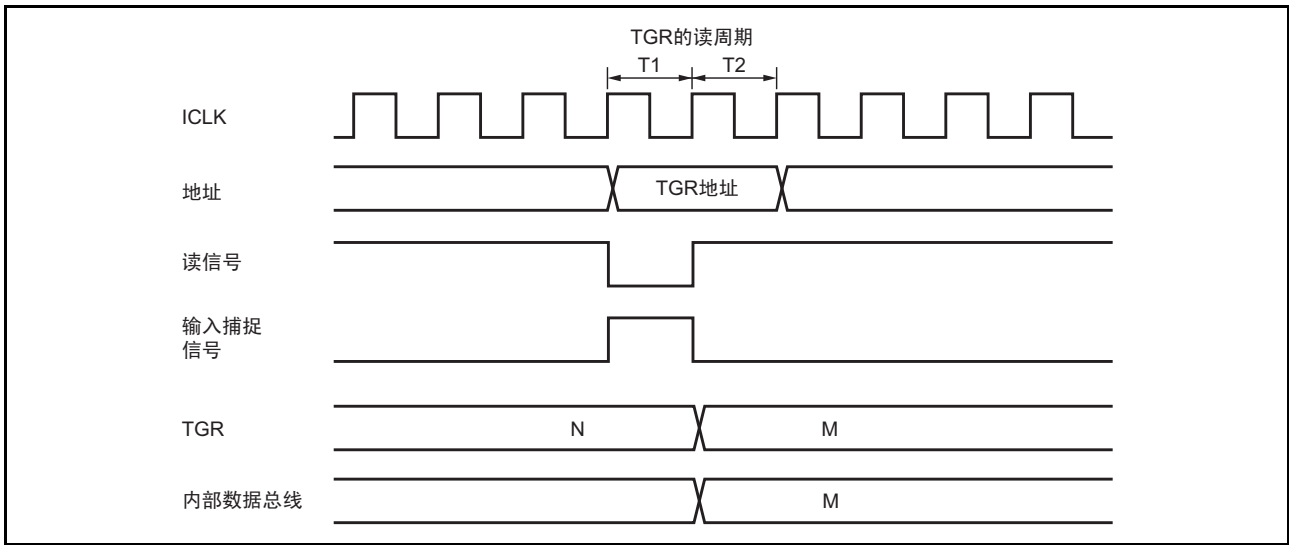


图 16.131 TGR 寄存器的读和输入捕捉的竞争 (MTU5)

### 16.6.10 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 16.132 和图 16.133 所示。

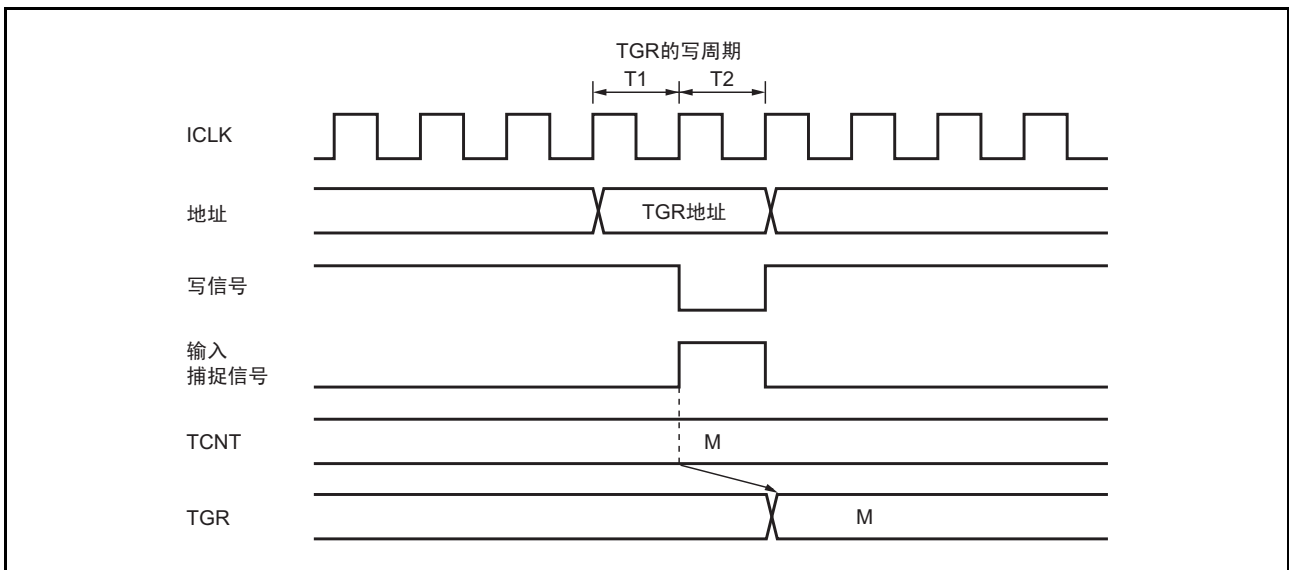


图 16.132 TGR 寄存器的写和输入捕捉的竞争 (MTU0 ~ 4、6、7)

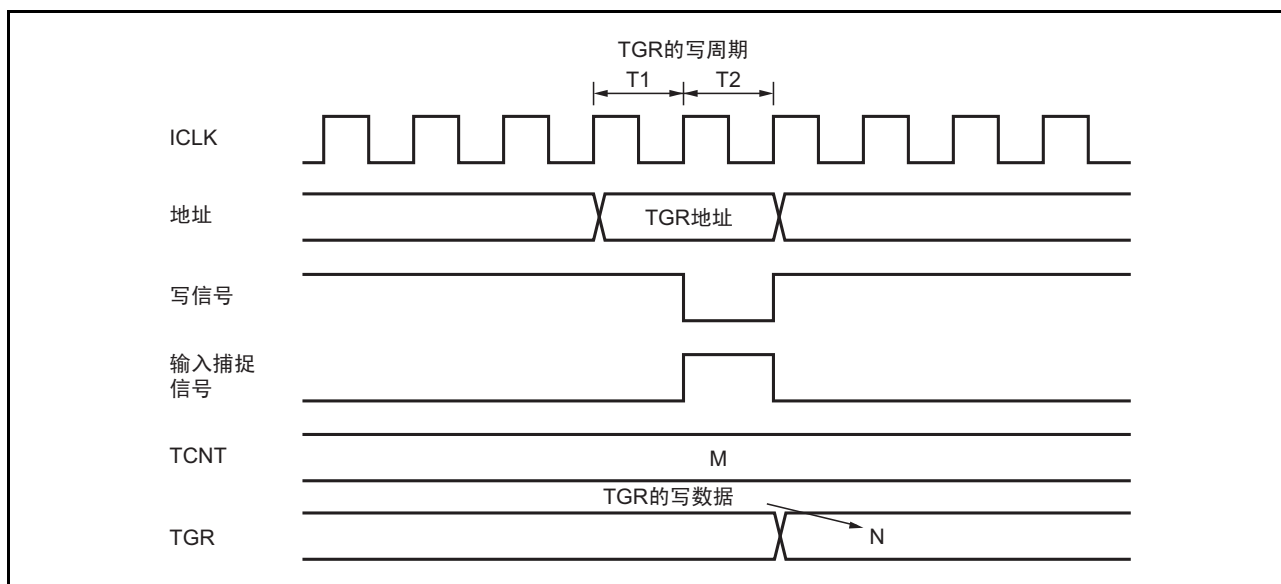


图 16.133 TGR 寄存器的写和输入捕捉的竞争 (MTU5)

### 16.6.11 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 16.134 所示。

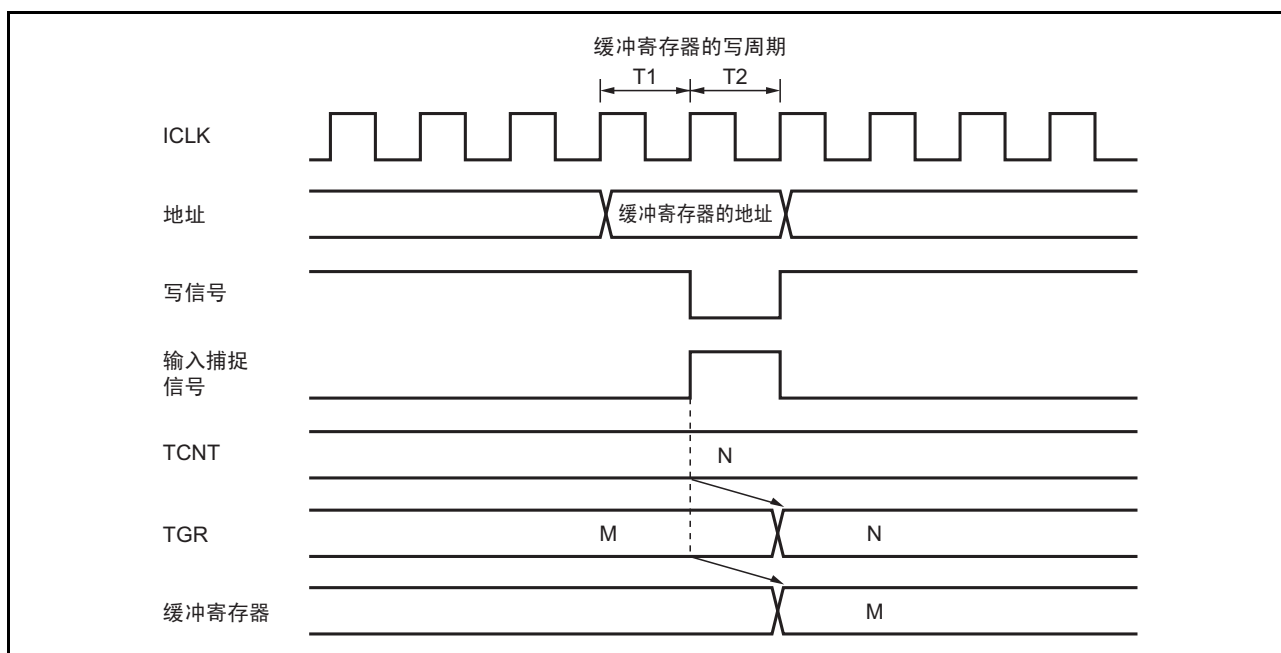


图 16.134 缓冲寄存器的写和输入捕捉的竞争



### 16.6.12 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 16.135 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。

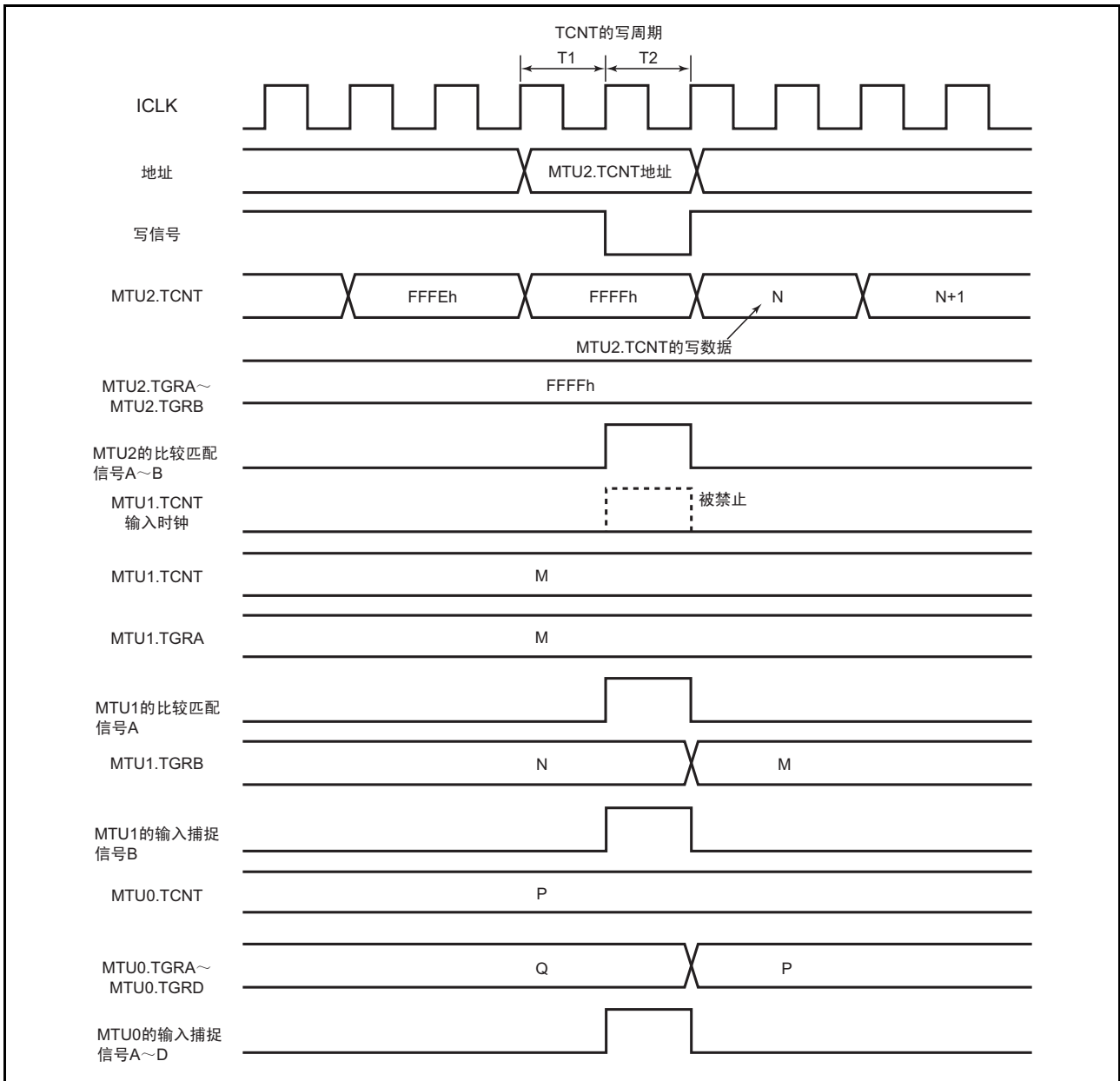


图 16.135 级联中的 MTU2.TCNT 的写和上溢 / 下溢的竞争

### 16.6.13 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 16.136 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。

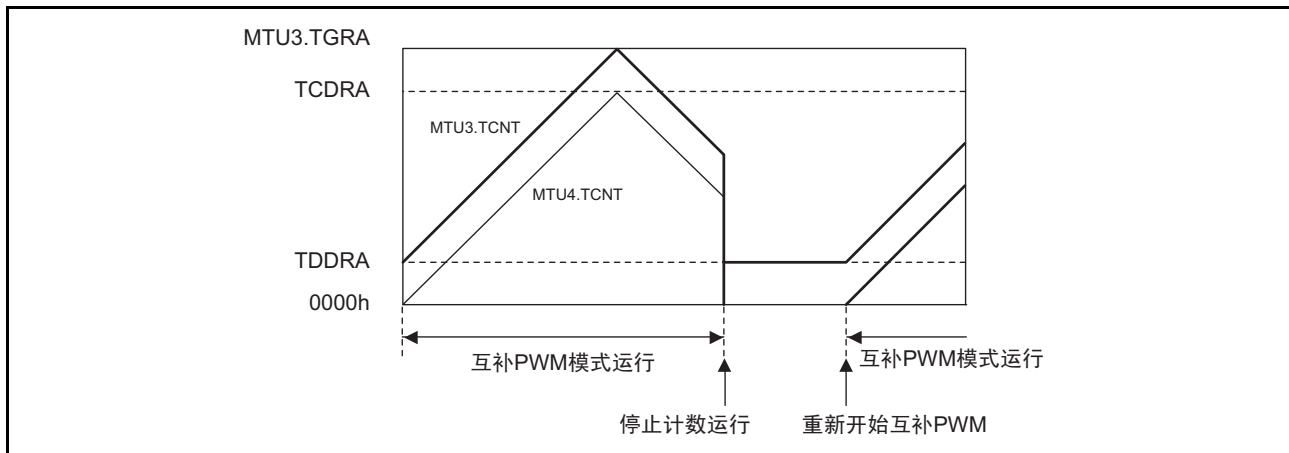


图 16.136 互补 PWM 模式停止时的计数器值

### 16.6.14 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。

### 16.6.15 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 16.137 所示。

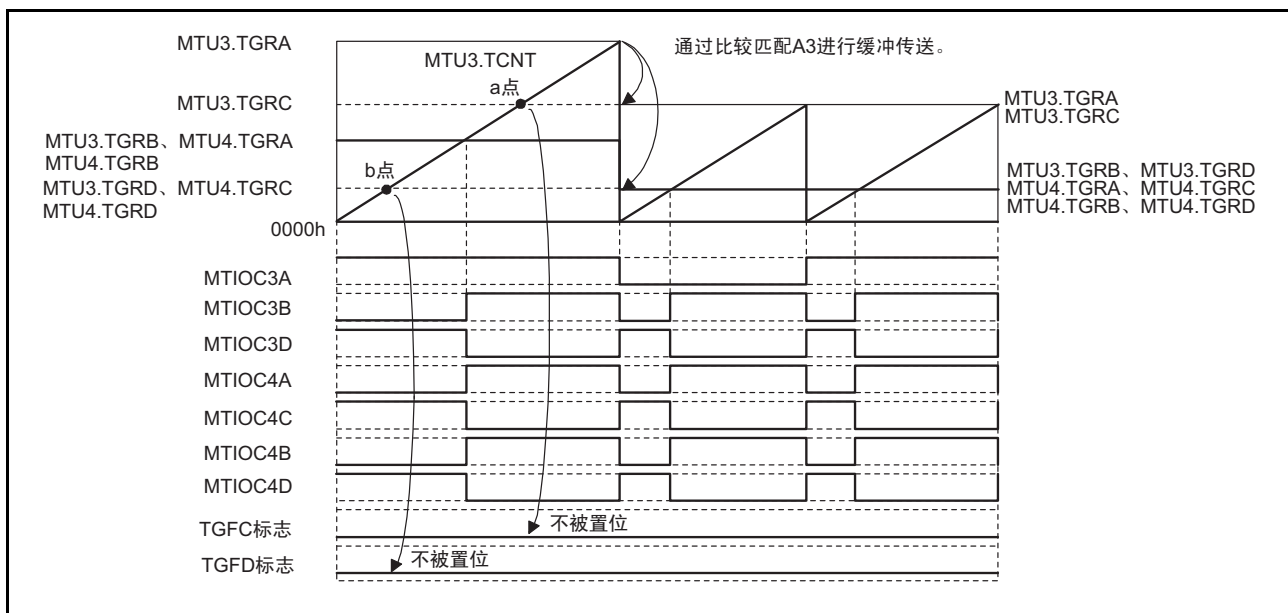


图 16.137 复位同步 PWM 模式的缓冲运行和比较匹配标志

### 16.6.16 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 16.138 所示。

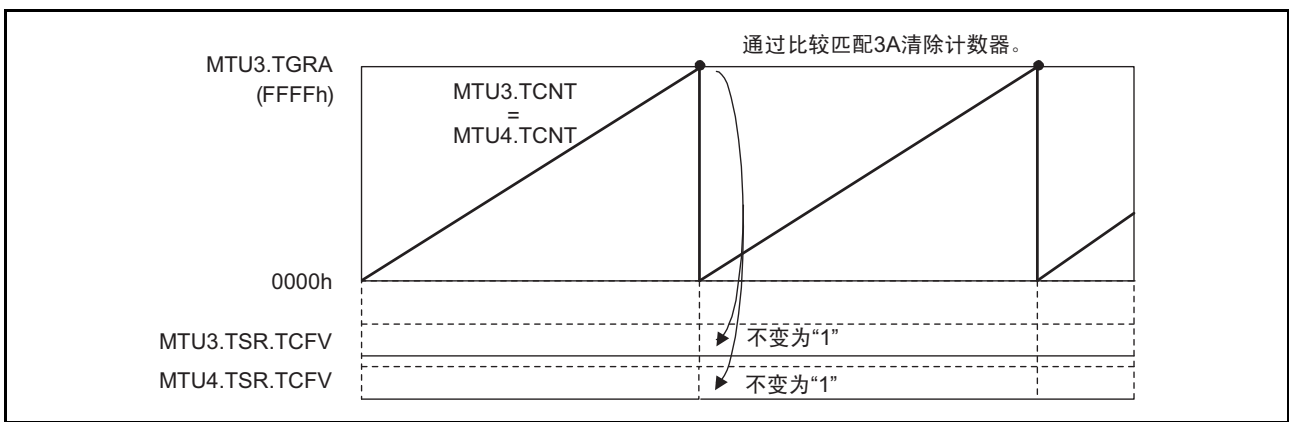


图 16.138 复位同步 PWM 模式的上溢标志

### 16.6.17 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 16.139 所示。

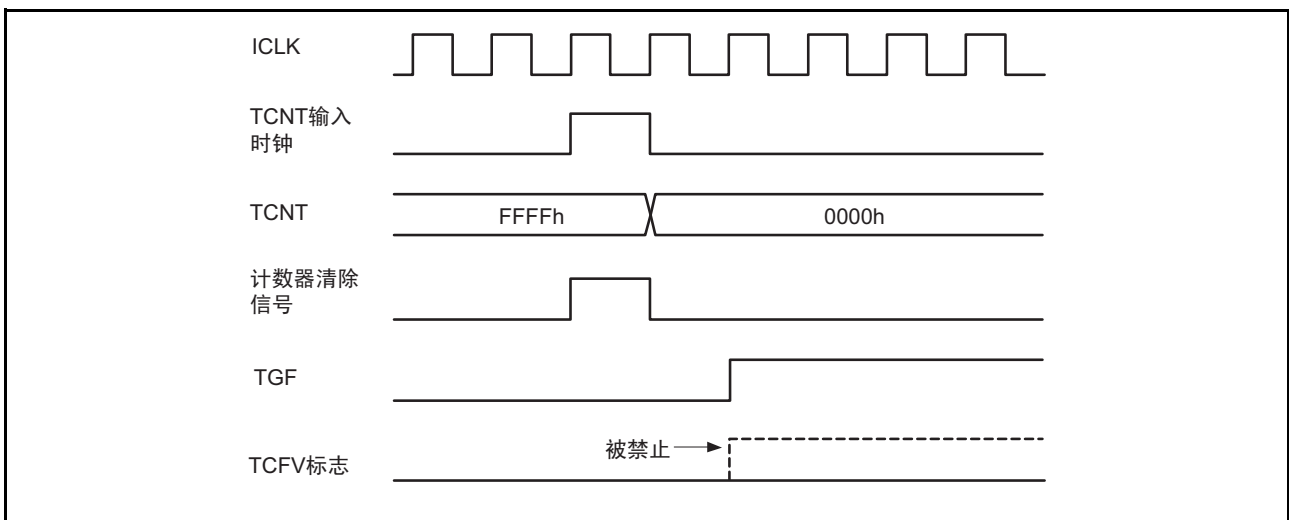


图 16.139 上溢和计数器清除的竞争

### 16.6.18 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 16.140 所示。

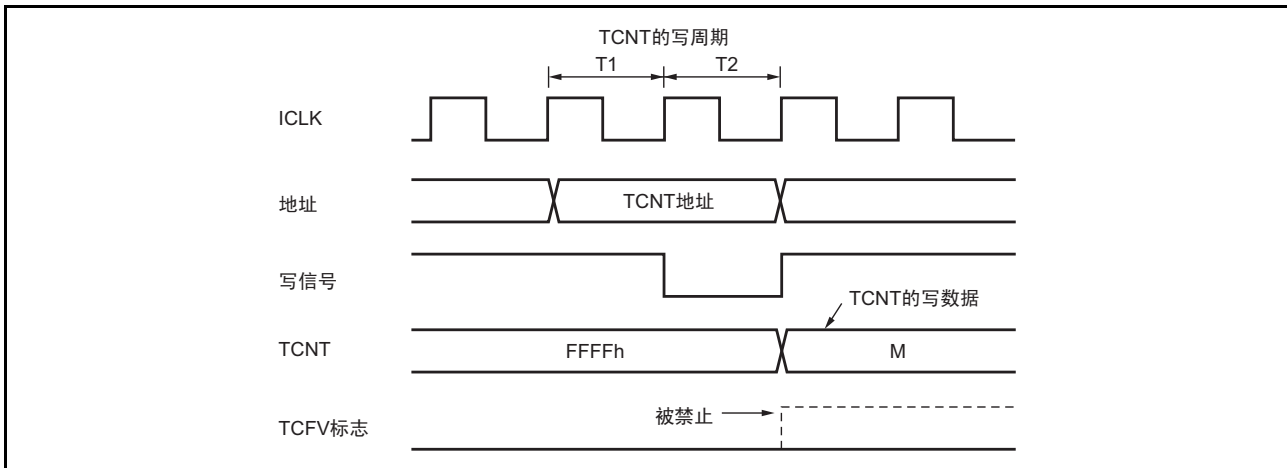


图 16.140 TCNT 的写和上溢的竞争

### 16.6.19 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。

### 16.6.20 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 位设定的正相输出的取反电平。

### 16.6.21 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 个输入捕捉的输入同时捕捉 MTU1.TCNT 和 MTU2.TCNT 的功能。如果使用本功能, 就能在不产生 MTU1.TCNT 和 MTU2.TCNT 的捕捉时序偏差的情况下执行 32 位计数器的读取操作。详细内容请参照 “16.2.9 定时器的输入捕捉控制寄存器 (TICCR) ”。

### 16.6.22 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  $\geq 4$
- MTU4.TADCORA 的比较间隔  $\geq 4$  个 ICLK (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  $\geq 4$  个 ICLK (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  $\geq 2$
- MTU4.TADCORB 的比较间隔  $\geq 2$  个 ICLK (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)

## 16.7 MTU 输出引脚的初始化方法

### 16.7.1 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。

### 16.7.2 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，需要预先通过 I/O 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定给通用输入 / 输出端口输出无效电平，禁止 MTU 的引脚输出，并且作为通用输出端口，将无效电平输出到引脚。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 16.75 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 16.75 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)

### 16.7.3 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端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 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端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 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端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 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端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 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端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 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 16.75 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 16.141 所示。

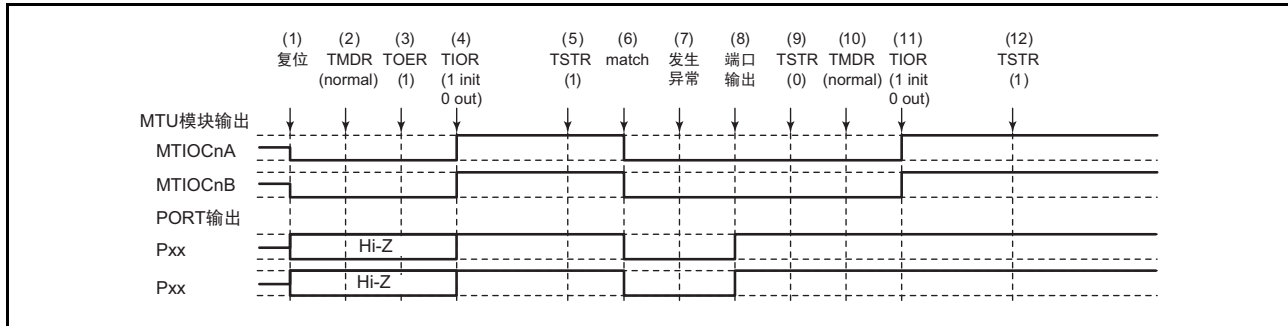


图 16.141 在正常模式中发生异常并且在正常模式中恢复的情况

- (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。  
另外，必须通过 I/O 端口的数据方向寄存器（DDR）和数据寄存器（DR）设定“将无效电平输出到通用输出端口”。
- (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) 通过 TSTR 寄存器开始计数运行。
- (6) 在发生比较匹配时输出低电平。
- (7) 发生异常。
- (8) 必须通过 TIOR 寄存器禁止 MTU 引脚输出，而作为端口输出，反相输出有效电平到引脚。
- (9) 通过 TSTR 寄存器停止计数运行。
- (10) 在正常模式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。
- (12) 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 16.142 所示。

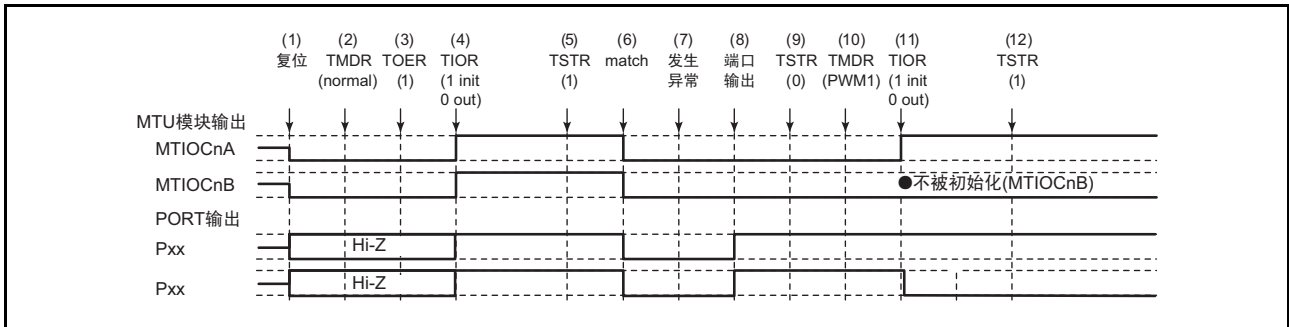


图 16.142 在正常模式中发生异常并且在 PWM 模式 1 中恢复的情况

(1) ~ (9) 和图 16.141 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 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定通用输出端口)。

(12) 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 16.143 所示。

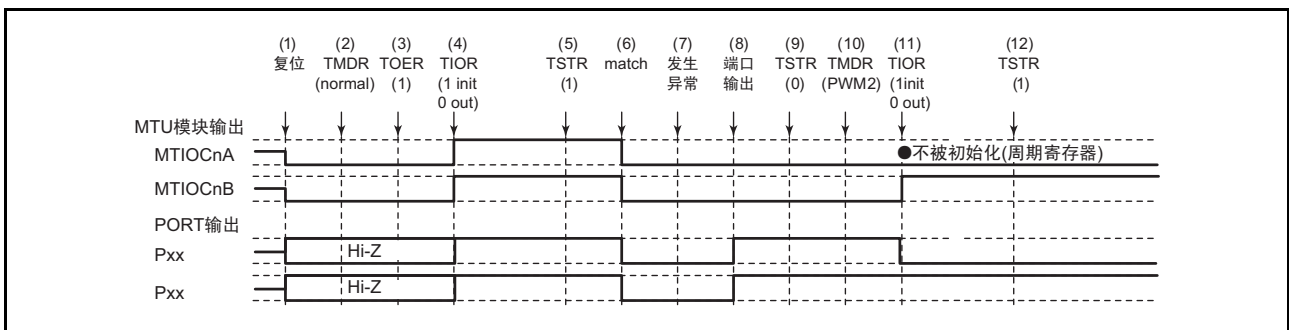


图 16.143 在正常模式中发生异常并且在 PWM 模式 2 中恢复的情况

(1) ~ (9) 和图 16.141 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 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定通用输出端口)。

(12) 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 16.144 所示。

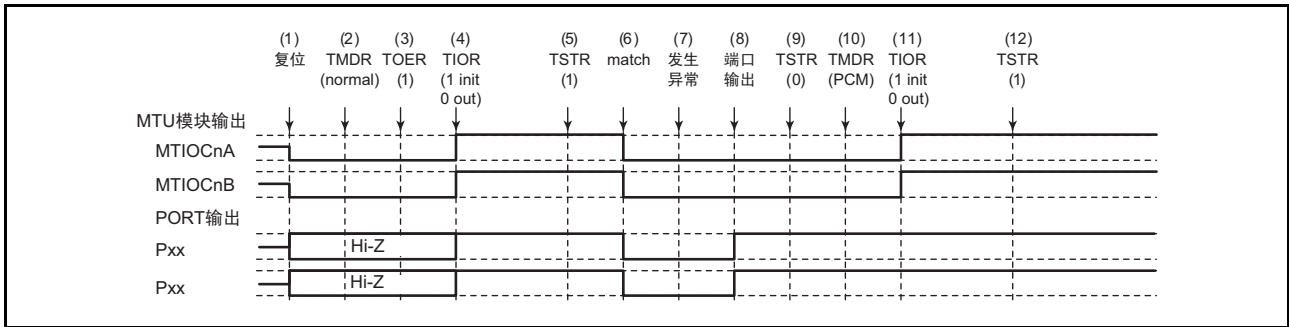


图 16.144 在正常模式中发生异常并且在相位计数模式中恢复的情况

- (1) ~ (9) 和图 16.141 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) 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。

(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 16.145 所示。

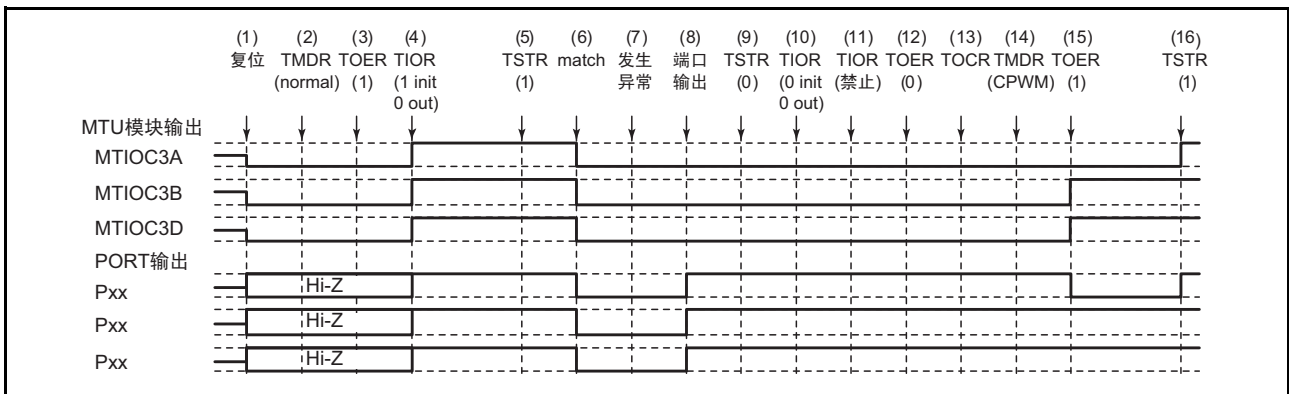


图 16.145 在正常模式中发生异常并且在互补 PWM 模式中恢复的情况

- (1) ~ (9) 和图 16.141 通用。
- (10) 必须通过 TIOR 寄存器对正常模式的波形生成部进行初始化。
- (11) 必须通过 TIOR 寄存器禁止正常模式的波形生成部的运行。
- (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 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) 通过 TSTR 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 16.146 所示。

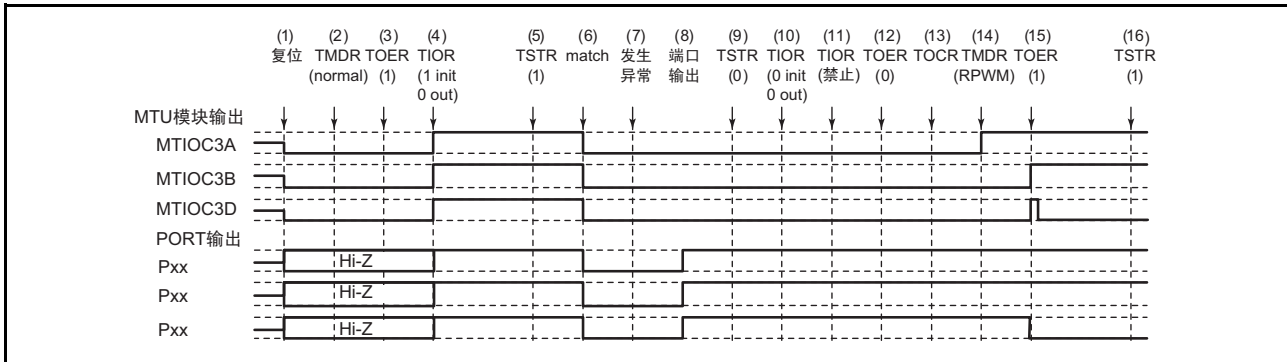


图 16.146 在正常模式中发生异常并且复位同步 PWM 模式中恢复的情况

- (1) ~ (12) 和图 16.141 通用。
- (13) 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。
- (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) 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 16.147 所示。

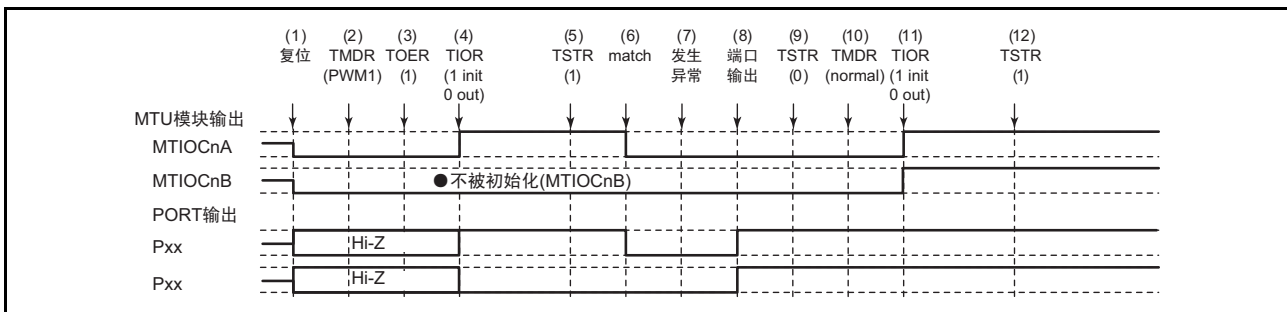


图 16.147 在 PWM 模式 1 中发生异常并且在正常模式中恢复的情况

- (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。另外，必须通过 I/O 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定“将无效电平输出到通用输出端口”。
- (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) 通过 TSTRA 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 TIOR 寄存器禁止 MTU 引脚输出，而作为通用输出端口，反相输出有效电平到引脚。
- (9) 通过 TSTRA 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) 通过 TSTRA 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 16.148 所示。

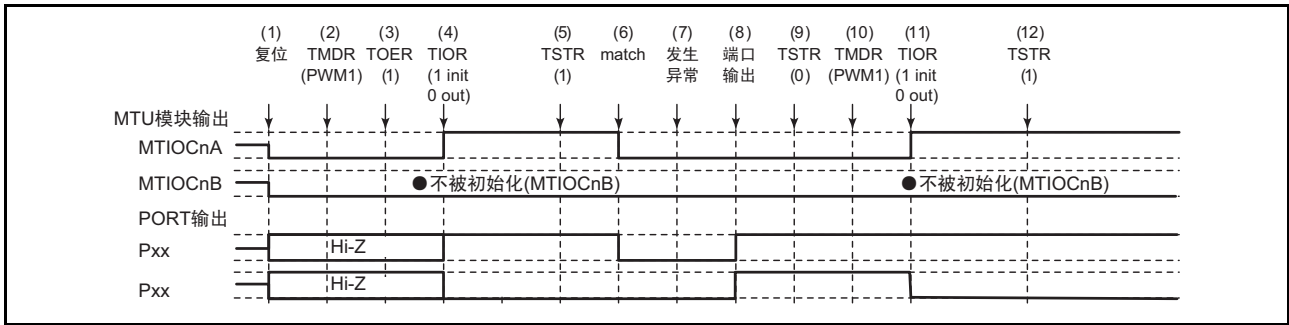


图 16.148 在 PWM 模式 1 中发生异常并且在 PWM 模式 1 中恢复的情况

(1) ~ (9) 和图 16.147 通用。

(10) 在 PWM 模式 1 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 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 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定通用输出端口)。

(12) 通过 TSTRA 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 16.149 所示。

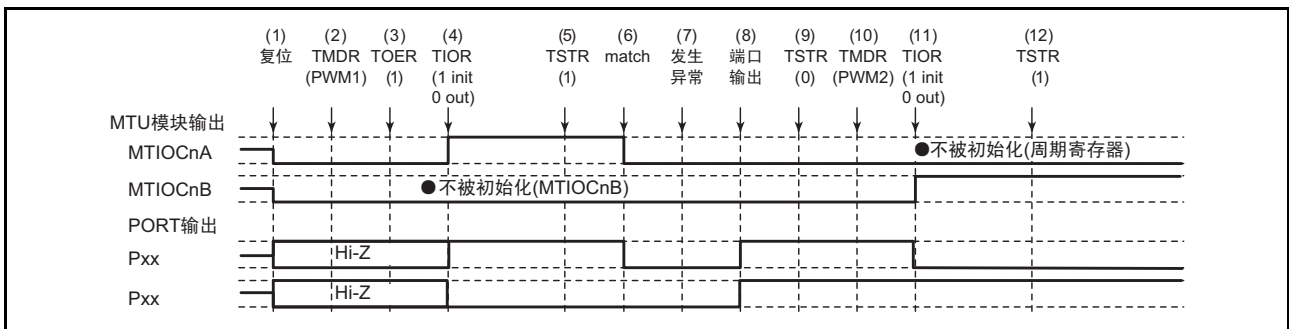


图 16.149 在 PWM 模式 1 中发生异常并且在 PWM 模式 2 中恢复的情况

(1) ~ (9) 和图 16.147 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 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定通用输出端口)。

(12) 通过 TSTRA 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 16.150 所示。

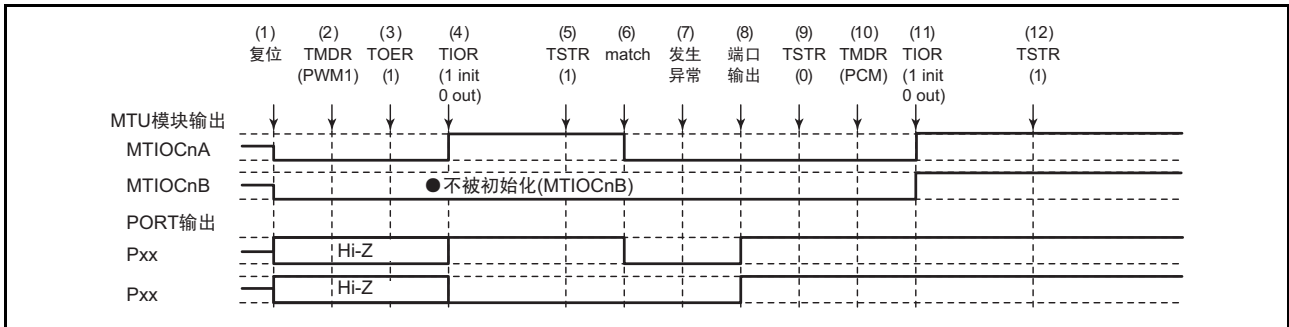


图 16.150 在 PWM 模式 1 中发生异常并且在相位计数模式中恢复的情况

- (1) ~ (9) 和图 16.147 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) 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 16.151 所示。

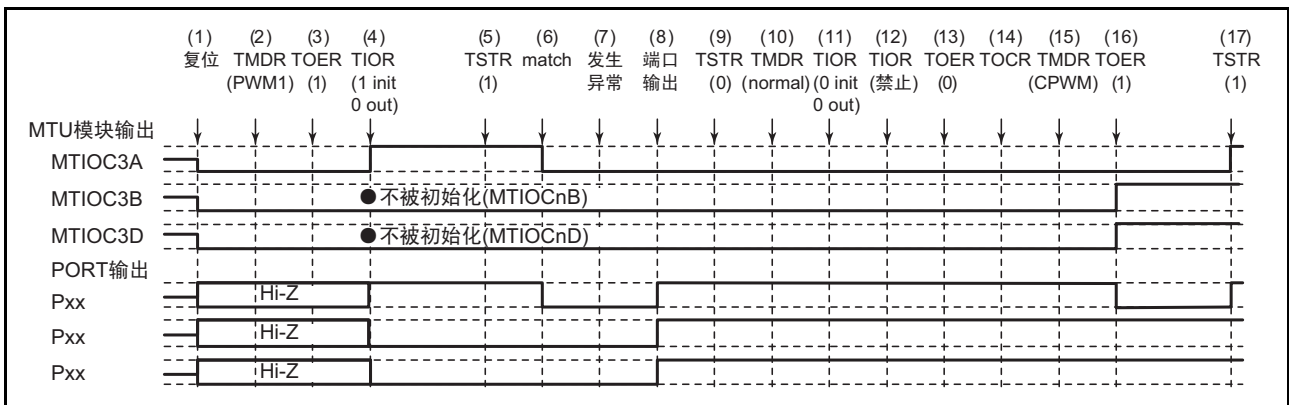


图 16.151 在 PWM 模式 1 中发生异常并且在互补 PWM 模式中恢复的情况

- (1) ~ (9) 和图 16.147 通用。
- (10) 为了对波形生成部进行初始化，必须设定正常模式。
- (11) 必须通过 TIOR 寄存器对 PWM 模式 1 的波形生成部进行初始化。
- (12) 必须通过 TIOR 寄存器禁止 PWM 模式 1 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) 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 模式中重新开始的说明图如图 16.152 所示。

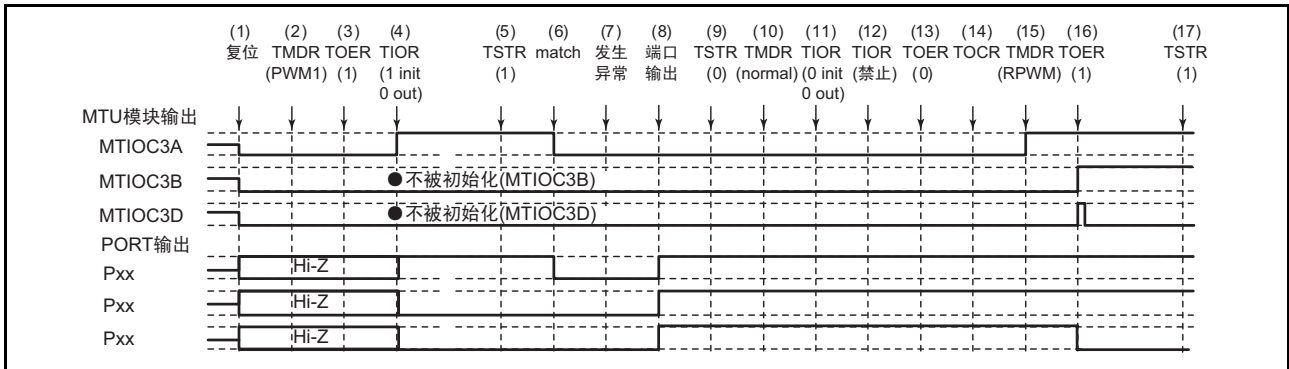


图 16.152 在 PWM 模式 1 中发生异常并且在复位同步 PWM 模式中恢复的情况

- (1) ~ (13) 和图 16.151 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) 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的说明图如图 16.153 所示。

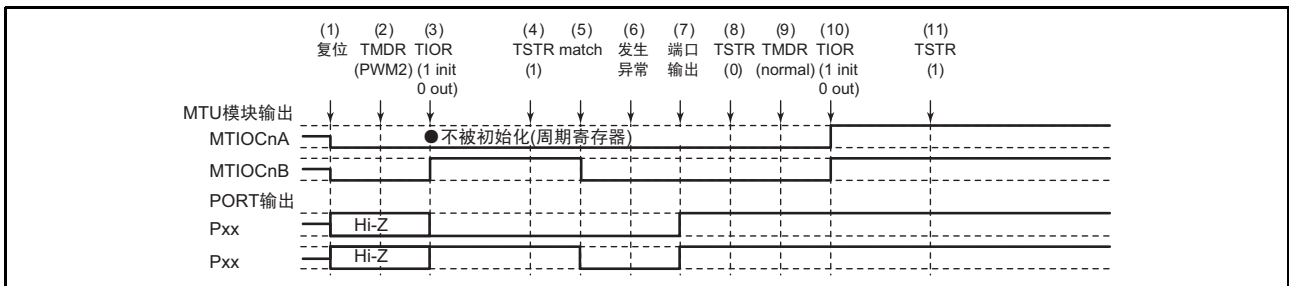


图 16.153 在 PWM 模式 2 中发生异常并且在正常模式中恢复的情况

- (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。  
另外，必须通过 I/O 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定“将无效电平输出到通用输出端口”。
- (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) 通过 TSTRA 寄存器开始计数运行。
- (5) 在比较匹配时输出 Low 电平。
- (6) 发生异常。
- (7) 必须通过 TIOR 寄存器禁止 MTU 引脚输出，而作为通用端口输出，反相输出有效电平到引脚。
- (8) 通过 TSTRA 寄存器停止计数运行。
- (9) 必须设定正常模式。
- (10) 必须通过 TIOR 寄存器对引脚进行初始化。
- (11) 通过 TSTRA 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 16.154 所示。

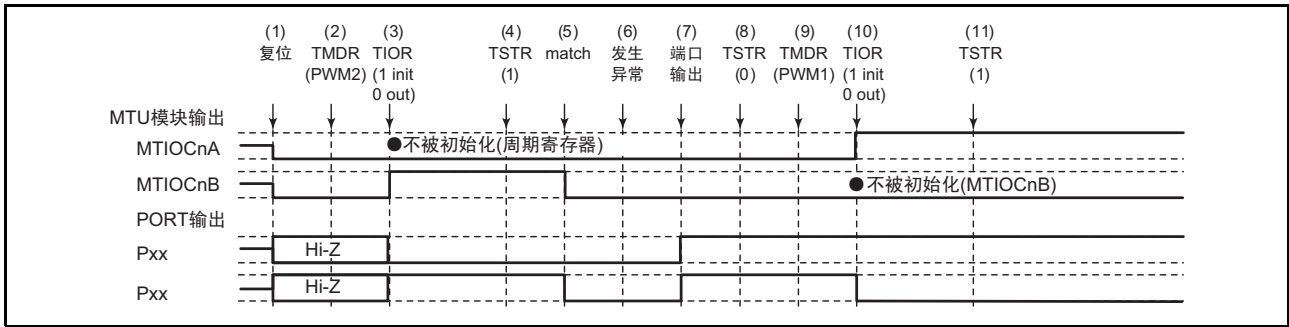


图 16.154 在 PWM 模式 2 中发生异常并且在 PWM 模式 1 中恢复的情况

(1) ~ (8) 和图 16.153 通用。

(9) 设定 PWM 模式 1。

(10) 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 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定通用输出端口)。

(11) 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 16.155 所示。

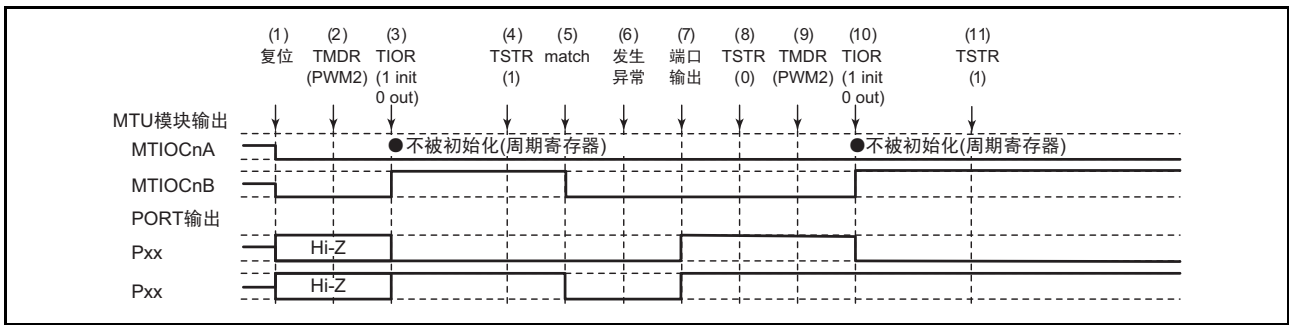


图 16.155 在 PWM 模式 2 中发生异常并且在 PWM 模式 2 中恢复的情况

(1) ~ (8) 和图 16.153 通用。

(9) 在 PWM 模式 2 中重新开始的情况下不需要此步骤。

(10) 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 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定通用输出端口)。

(11) 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 16.156 所示。

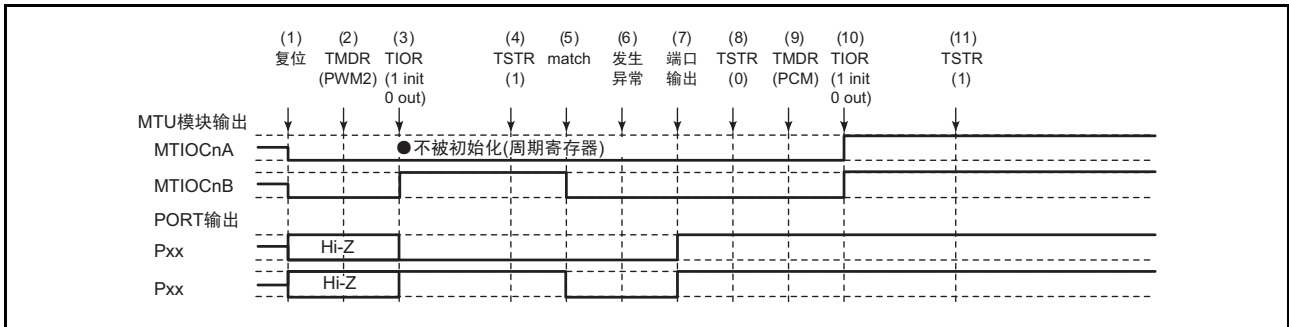


图 16.156 在 PWM 模式 2 中发生异常并且在相位计数模式中恢复的情况

- (1) ~ (8) 和图 16.153 通用。
- (9) 设定相位计数模式。
- (10) 必须通过 TIOR 寄存器对引脚进行初始化。
- (11) 通过 TSTRA 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 16.157 所示。

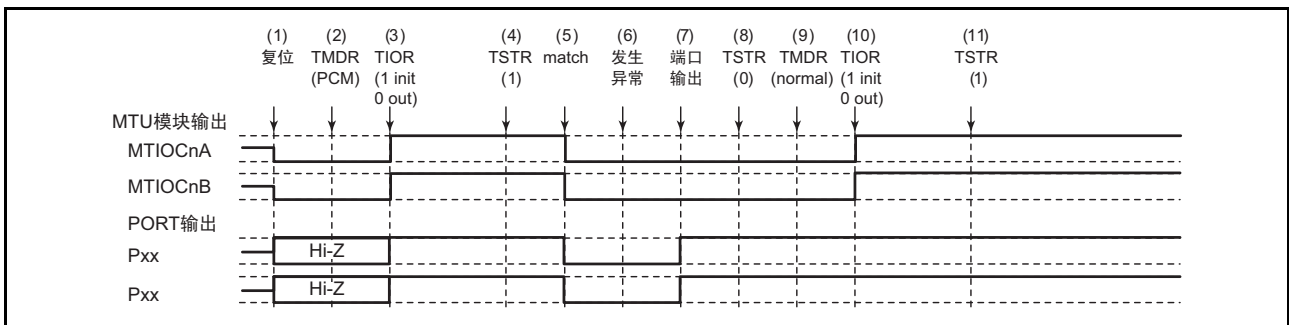


图 16.157 在相位计数模式中发生异常并且在正常模式中恢复的情况

- (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。  
另外，必须通过 I/O 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定“将无效电平输出到通用输出端口”。
- (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) 通过 TSTRA 寄存器开始计数运行。
- (5) 在比较匹配时输出 Low 电平。
- (6) 发生异常。
- (7) 必须通过 TIOR 寄存器禁止 MTU 引脚输出，而作为通用端口输出，输出无效电平到引脚。
- (8) 通过 TSTRA 寄存器停止计数运行。
- (9) 必须在正常模式中进行设定。
- (10) 必须通过 TIOR 寄存器对引脚进行初始化。
- (11) 通过 TSTRA 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 16.158 所示。

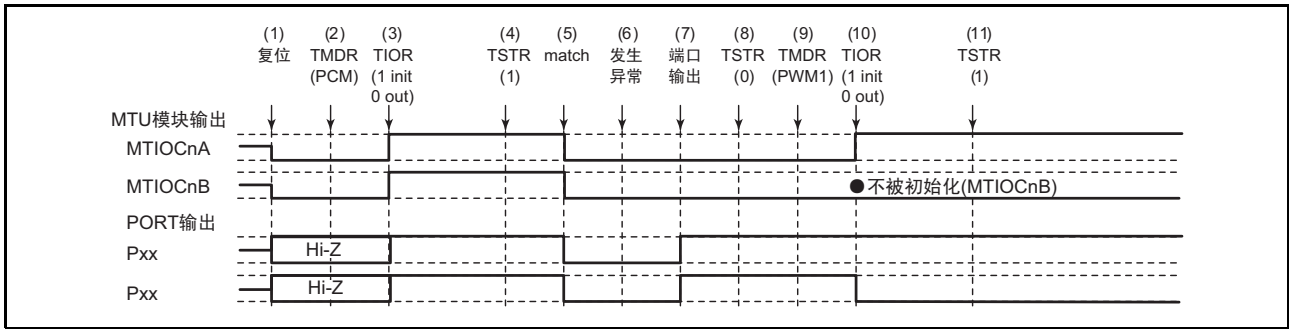


图 16.158 在相位计数模式中发生异常并且在 PWM 模式 1 中恢复的情况

- (1) ~ (8) 和图 16.157 通用。
- (9) 设定 PWM 模式 1。
- (10) 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 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定通用输出端口)。
- (11) 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 16.159 所示。

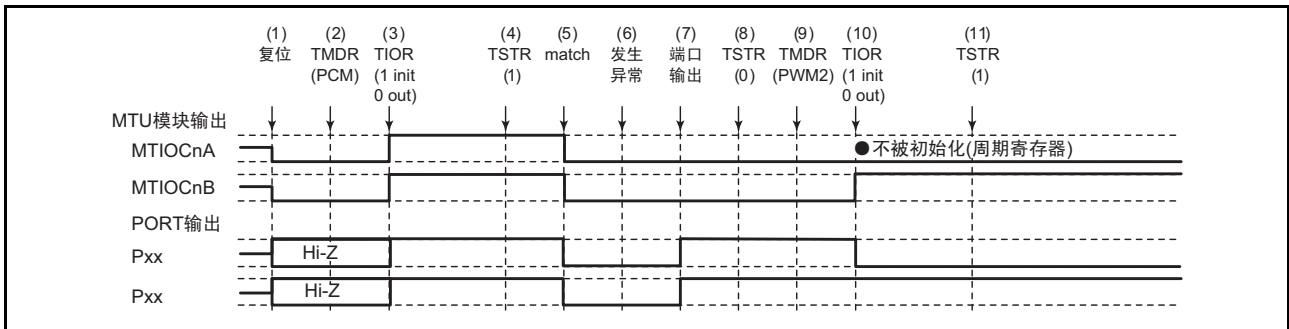


图 16.159 在相位计数模式中发生异常并且在 PWM 模式 2 中恢复的情况

- (1) ~ (8) 和图 16.157 通用。
- (9) 设定 PWM 模式 2。
- (10) 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 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定通用输出端口)。
- (11) 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 16.160 所示。

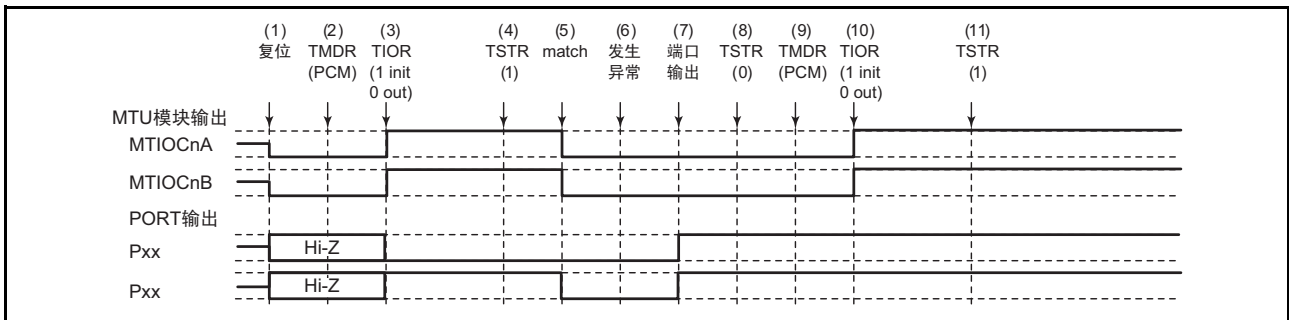


图 16.160 在相位计数模式中发生异常并且在相位计数模式中恢复的情况

- (1) ~ (8) 和图 16.157 通用。
- (9) 在相位计数模式重新开始的情况下不需要此步骤。
- (10) 必须通过 TIOR 寄存器对引脚进行初始化。
- (11) 通过 TSTRA 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时的说明图如图 16.161 所示。

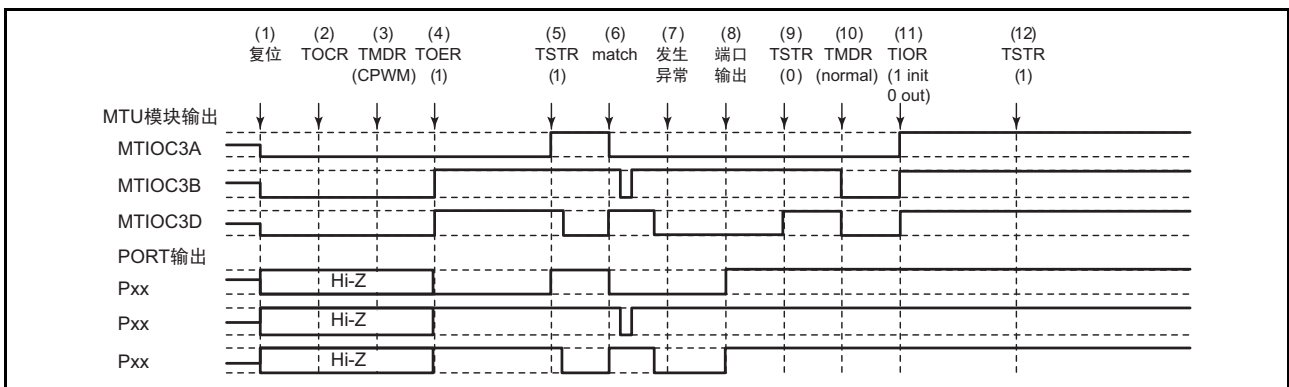


图 16.161 在互补 PWM 模式中发生异常并且在正常模式中恢复的情况

- (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。  
另外，必须通过 I/O 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定“将无效电平输出到通用输出端口”。
- (4) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。
- (5) 通过 TSTRA 寄存器开始计数运行。
- (6) 通过发生比较匹配输出互补 PWM 波形。
- (7) 发生异常。
- (8) 必须通过 TIOR 寄存器禁止 MTU 引脚输出，而作为端口输出，输出无效电平到引脚。
- (9) 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)。
- (10) 必须设定正常模式 (MTU 的输出为 Low 电平)。
- (11) 必须通过 TIOR 寄存器对引脚进行初始化。
- (12) 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 16.162 所示。

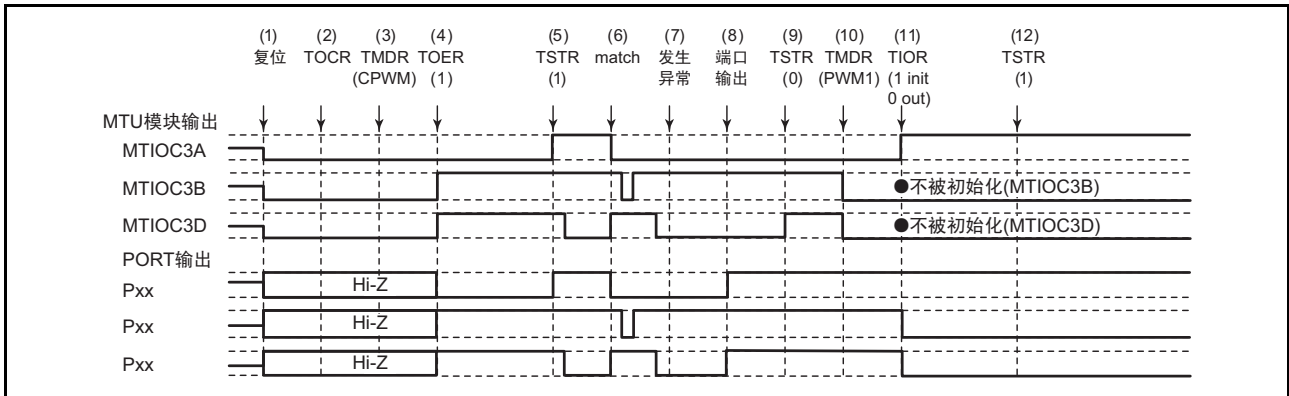


图 16.162 在互补 PWM 模式中发生异常并且在 PWM 模式 1 中恢复的情况

- (1) ~ (9) 和图 16.161 通用。
- (10) 必须设定 PWM 模式 1 (MTU 的输出为 Low 电平)。
- (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 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定通用输出端口)。
- (12) 通过 TSTRA 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 16.163 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)。

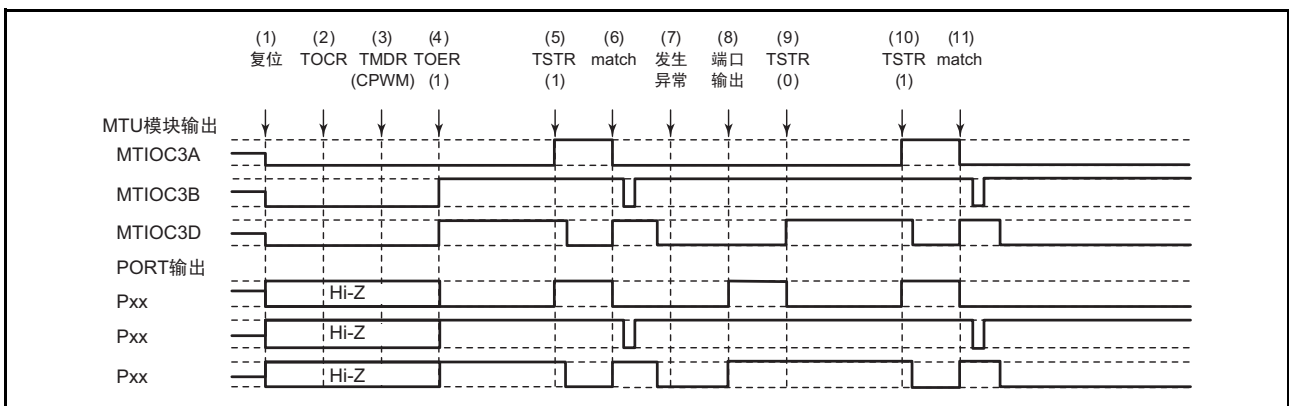


图 16.163 在互补 PWM 模式中发生异常并且在互补 PWM 模式中恢复的情况 (1)

- (1) ~ (9) 和图 16.161 通用。
- (10) 通过 TSTRA 寄存器重新开始。
- (11) 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 16.164 所示 (从重新设定的周期和占空比的值重新开始的情况)。

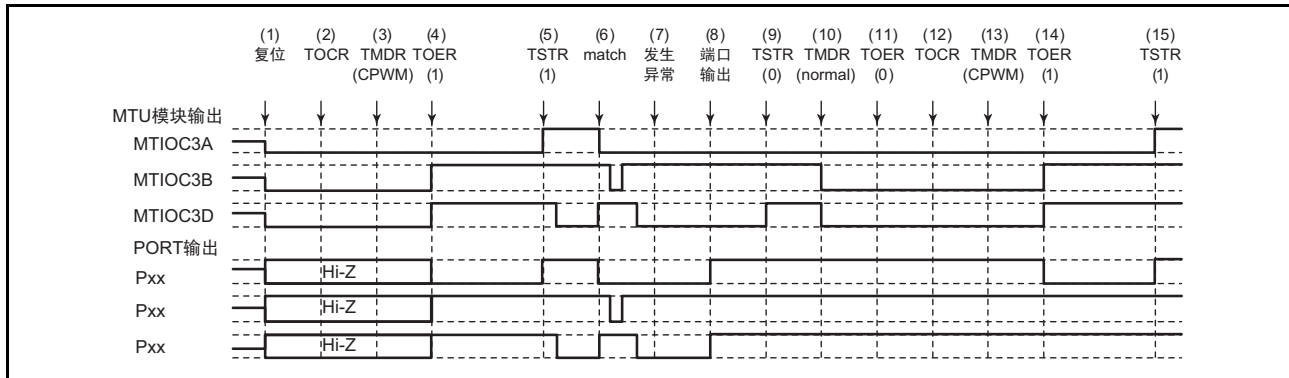


图 16.164 在互补 PWM 模式中发生异常并且在互补 PWM 模式中恢复的情况 (2)

(1) ~ (9) 和图 16.161 通用。

(10) 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)。

(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 模式的输出电平以及允许或者禁止周期输出。

(13) 设定互补 PWM。

(14) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。

(15) 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 16.165 所示。

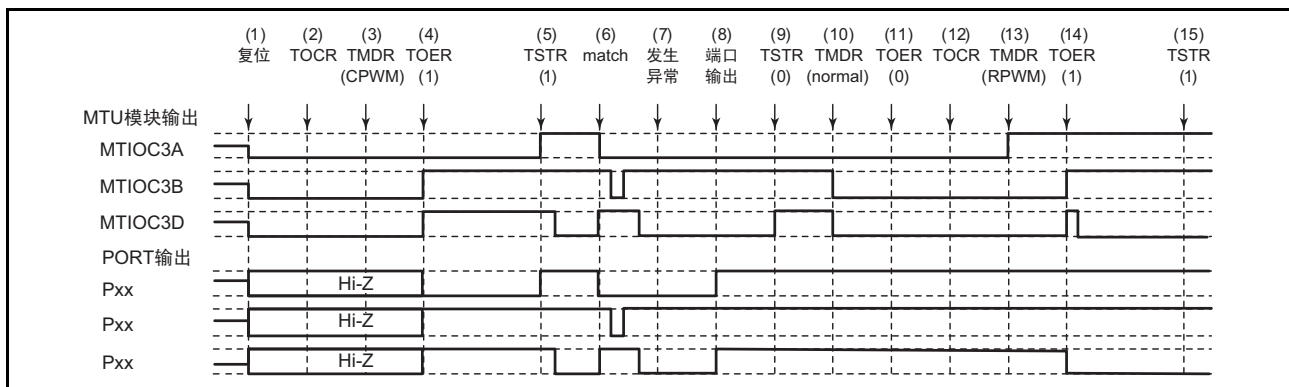


图 16.165 在互补 PWM 模式中发生异常并且在复位同步 PWM 模式中恢复的情况

(1) ~ (9) 和图 16.161 通用。

(10) 必须设定正常模式 (MTU 的输出为 Low 电平)。

(11) 必须通过 TOERA 寄存器禁止 MTU3 和 MTU4 的输出。

(12) 必须通过 TOCR1A、TOCR2A 寄存器选择复位同步 PWM 模式的输出电平以及允许或者禁止周期输出。

(13) 设定复位同步 PWM。

(14) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。

(15) 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 16.166 所示。

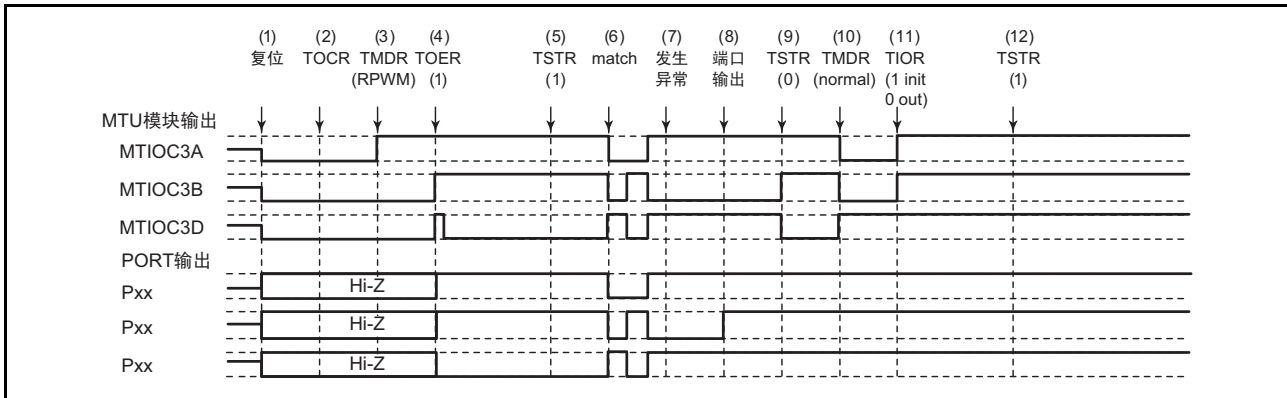


图 16.166 在复位同步 PWM 模式中发生异常并且在正常模式中恢复的情况

- (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。  
另外，必须通过 I/O 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定“将无效电平输出到通用输出端口”。
- (4) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。
- (5) 通过 TSTRA 寄存器开始计数运行。
- (6) 通过发生比较匹配输出复位同步 PWM 波形。
- (7) 发生异常。
- (8) 必须通过 TIOR 寄存器禁止 MTU 引脚输出，而作为端口输出，输出无效电平到引脚。
- (9) 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)。
- (10) 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)。
- (11) 必须通过 TIOR 寄存器对引脚进行初始化。
- (12) 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 16.167 所示。

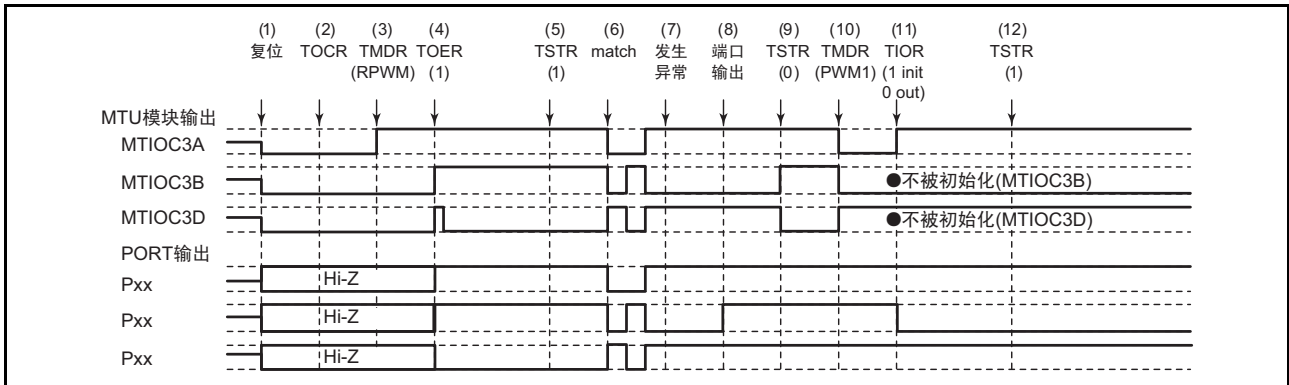


图 16.167 在复位同步 PWM 模式中发生异常并且在 PWM 模式 1 中恢复的情况

(1) ~ (9) 和图 16.166 通用。

(10) 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)。

(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 端口的数据方向寄存器 (DDR) 和数据寄存器 (DR) 设定通用输出端口)。

(12) 通过 TSTRA 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 16.168 所示。

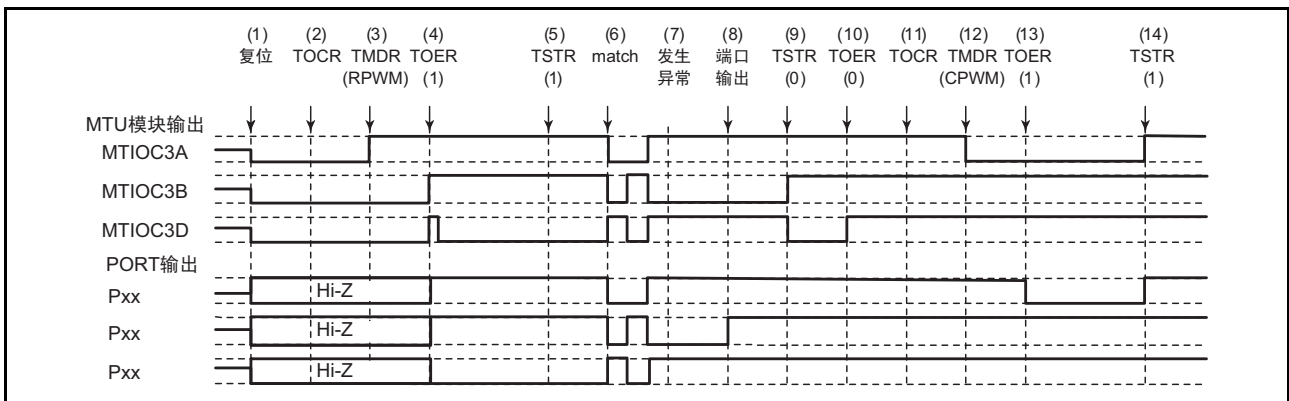


图 16.168 在复位同步 PWM 模式中发生异常并且在互补 PWM 模式中恢复的情况

(1) ~ (9) 和图 16.166 通用。

(10) 必须通过 TOERA 寄存器禁止 MTU3 和 MTU4 的输出。

(11) 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。

(12) 设定互补 PWM (MTU 的周期输出引脚为 Low 电平)。

(13) 必须通过 TOERA 寄存器允许 MTU3 和 MTU4 的输出。

(14) 通过 TSTRA 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 16.169 所示。

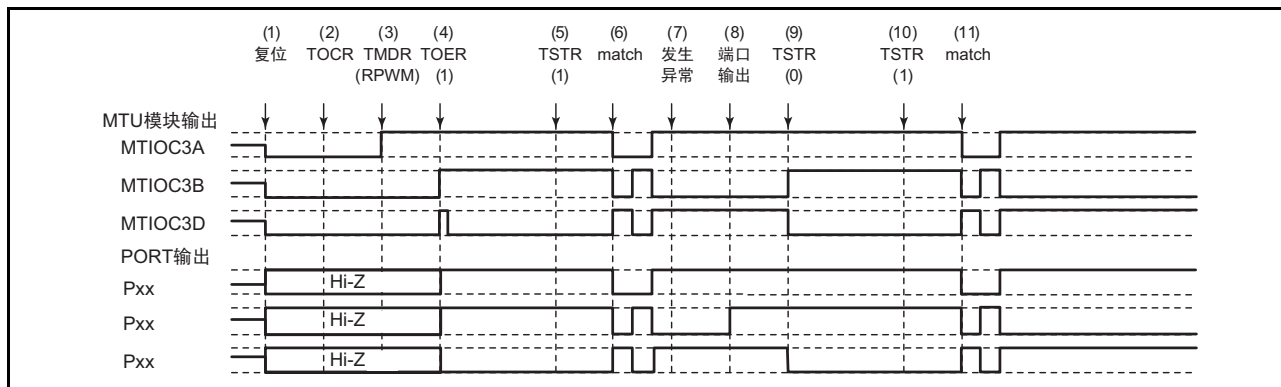


图 16.169 在复位同步 PWM 模式中发生异常并且在复位同步 PWM 模式中恢复的情况

- (1) ~ (9) 和图 16.166 通用。
- (10) 通过 TSTRA 寄存器重新开始。
- (11) 通过发生比较匹配输出复位同步 PWM 波形。



## 17. 端口输出允许 3 (POE3)

根据 POE0#、POE4#、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 以及 MTIOC6B、MTIOC6D、MTIOC7A、MTIOC7B、MTIOC7C 和 MTIOC7D) 的输出状态、GPT 大电流输出引脚 (GTIOC0A-A、GTIOC0B-A、GTIOC1A-A、GTIOC1B-A、GTIOC2A-A、GTIOC2B-A) 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。

### 17.1 概要

POE3 的规格和框图分别如表 17.1 和图 17.1 所示。

表 17.1 POE3 的规格

项目	内容
功能	<ul style="list-style-type: none"> <li>• 能给 POE0#、POE4#、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。</li> <li>• 能通过 POE0#、POE4#、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。</li> <li>• 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。</li> <li>• 如果在对 MTU 的互补 PWM 输出引脚或者 GPT 大电流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 大电流输出引脚置为高阻抗状态。</li> <li>• 能通过 12 位 A/D 转换器 (S12ADA) 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。</li> <li>• 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。</li> <li>• 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。</li> </ul>

如图 17.1 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。

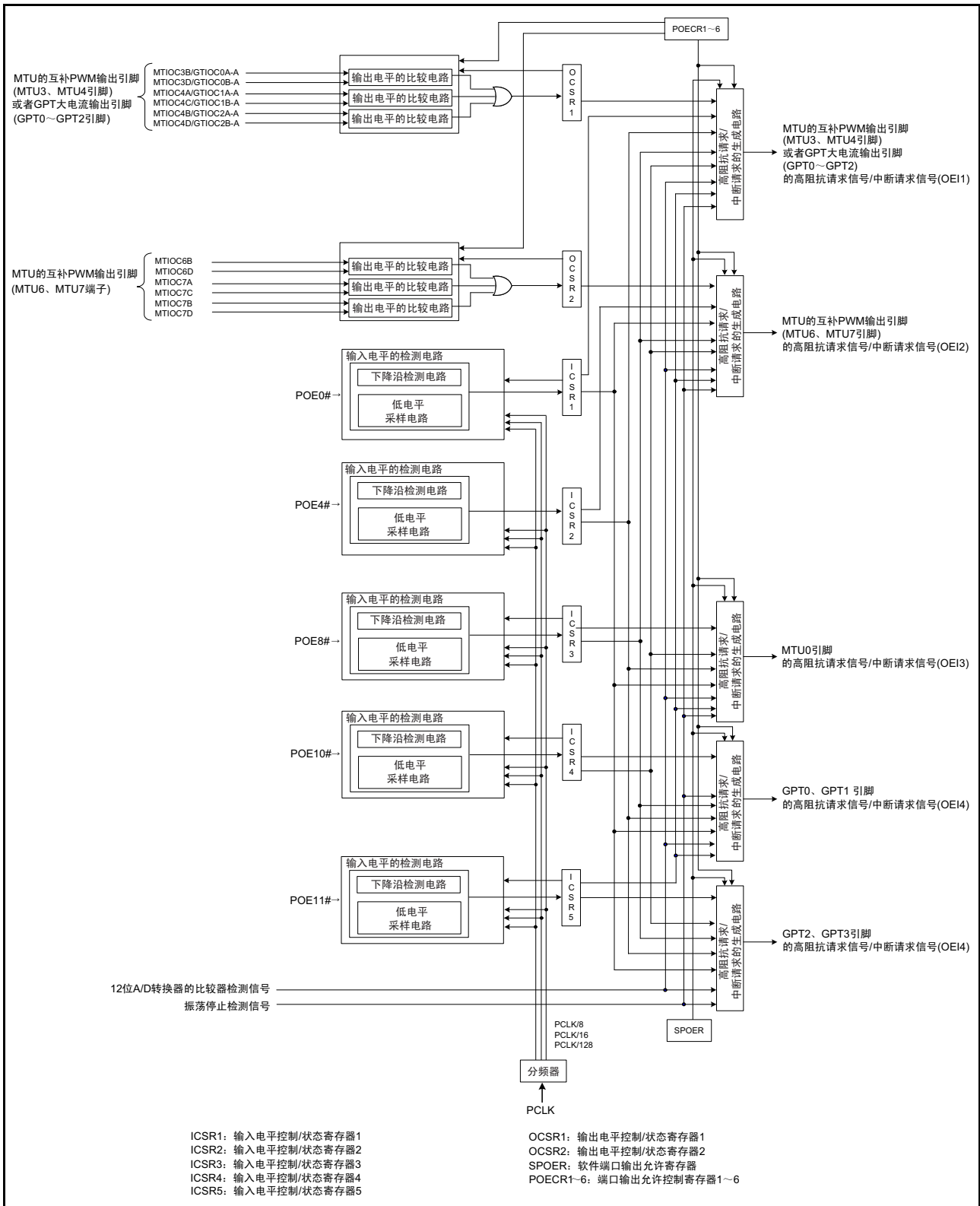


图 17.1 POE3 的框图

POE3 使用的输入 / 输出引脚如表 17.2 所示。

表 17.2 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) 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 (MTU6、MTU7 引脚)、MTU0 引脚、GPT 引脚置为高阻抗状态。
POE4#	输入	将 MTU 的互补 PWM 输出引脚 (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 MTU 的互补 PWM 输出引脚 (MTU3、MTU4 引脚)、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 引脚和 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 引脚和 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 引脚和 MTU6、MTU7 引脚)、MTU0 引脚、GPT0 引脚、GPT1 引脚置为高阻抗状态。

通过表 17.3 所示的引脚组合进行输出电平的比较。

表 17.3 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-A 和 GTIOC0B-A	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 大电流输出引脚 (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-A 和 GTIOC1B-A	输出	
GTIOC2A-A 和 GTIOC2B-A	输出	

## 17.2 寄存器说明

POE3 的寄存器一览表如表 17.4 所示。  
通过复位对 POE3 的寄存器进行初始化。

表 17.4 POE3 的寄存器一览表

寄存器名	符号	复位后的值	地址	存取长度
输入电平控制 / 状态寄存器 1	ICSR1	0000h	0008 C4C0h	8、16
输出电平控制 / 状态寄存器 1	OCSR1	0000h	0008 C4C2h	8、16
输入电平控制 / 状态寄存器 2	ICSR2	0000h	0008 C4C4h	8、16
输出电平控制 / 状态寄存器 2	OCSR2	0000h	0008 C4C6h	8、16
输入电平控制 / 状态寄存器 3	ICSR3	0000h	0008 C4C8h	8、16
输入电平控制 / 状态寄存器 4	ICSR4	0000h	0008 C4D6h	8、16
输入电平控制 / 状态寄存器 5	ICSR5	0000h	0008 C4D8h	8、16
软件端口输出允许寄存器	SPOER	00h	0008 C4CAh	8
端口输出允许控制寄存器 1	POECR1	00h	0008 C4CBh	8
端口输出允许控制寄存器 2	POECR2	0707h	0008 C4CCh	16
端口输出允许控制寄存器 3	POECR3	0303h	0008 C4CEh	16
端口输出允许控制寄存器 4	POECR4	0402h	0008 C4D0h	16
端口输出允许控制寄存器 5	POECR5	0008h	0008 C4D2h	16
端口输出允许控制寄存器 6	POECR6	2010h	0008 C4D4h	16
有效电平寄存器 1	ALR1	0000h	0008 C4DAh	8、16

## 17.2.1 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

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

## 17.2.2 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 MTU 的互补 PWM 输出引脚 (MTU3、MTU4 引脚) 或者 GPT 大电流输出引脚 (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

## 17.2.3 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
—	—	—	—	—	—	—	—	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)
b15-b8	—	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/GTIOC0A-A 的有效电平设定位)

此位设定 MTIOC3B 输出和 GTIOC0A-A 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/GTIOC0B-A 的有效电平设定位)

此位设定 MTIOC3D 输出和 GTIOC0B-A 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/GTIOC1A-A 的有效电平设定位)

此位设定 MTIOC4A 输出和 GTIOC1A-A 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/GTIOC1B-A 的有效电平设定位)

此位设定 MTIOC4C 输出和 GTIOC1B-A 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/GTIOC2A-A 的有效电平设定位)**

此位设定 MTIOC4B 输出和 GTIOC2A-A 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/GTIOC2B-A 的有效电平设定位)**

此位设定 MTIOC4D 输出和 GTIOC2B-A 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)**

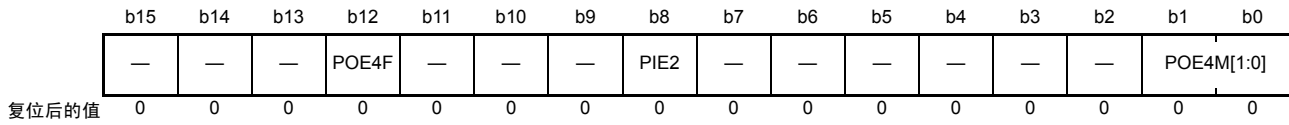
此位设定通过 OLSGnm 位 (n=0 ~ 2、m=A、B) 设定的有效电平有效或者无效。当 OLSEN 位为“0”时，OLSGnm 位的设定无效，MTU 输出 MTU.TOCR1 寄存器和 MTU.TOCR2 寄存器设定的有效电平；当 OLSEN 位为“1”时，MTU 输出 OLSGnm 位设定的有效电平。

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。



### 17.2.4 输入电平控制 / 状态寄存器 2 (ICSR2)

地址 0008 C4C4h



位	符号	位名	功能	R/W
b1-b0	POE4M[1:0]	POE4 模式选择位	b1 b0 00: 在 POE4# 输入的下降沿接受请求 01: 在每 PCLK/8 时钟对 POE4# 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。 10: 在每 PCLK/16 时钟对 POE4# 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。 11: 在每 PCLK/128 时钟对 POE4# 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	PIE2	端口中断允许 2 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	POE4F	POE4 标志	0: POE4# 引脚无高阻抗请求 1: POE4# 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”。

ICSR2 寄存器是选择 POE4# 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。

#### POE4M[1:0] 位 (POE4 模式选择位)

这些位选择 POE4# 引脚的输入模式。

#### PIE2 位 (端口中断允许 2 位)

当 POE4F 标志为“1”时, 此位指定是否请求中断。

#### POE4F 标志 (POE4 标志)

这是表示给 POE4# 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 ]

- POE4# 引脚发生 POE4M[1:0] 位设定的输入时

## 17.2.5 输出电平控制 / 状态寄存器 2 (OCSR2)

地址 0008 C4C6h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OSF2	—	—	—	—	—	OCE2	OIE2	—	—	—	—	—	—	—	—
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	OIE2	输出短路的中断允许 2 位	0: 禁止中断请求 1: 允许中断请求	R/W
b9	OCE2	输出短路的高阻抗允许 2 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	OSF2	输出短路标志 2	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”。

OCSR2 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。

## OIE2 位 (输出短路的中断允许 2 位)

当 OSF2 标志为“1”时, 此位指定是否请求中断。

## OCE2 位 (输出短路的高阻抗允许 2 位)

当 OSF2 标志为“1”时, 此位指定是否将引脚置为高阻抗。

## OSF2 标志 (输出短路标志 2)

此标志表示在对 MTU 的互补 PWM 输出引脚 (MTU6、MTU7 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

## 17.2.6 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

## 17.2.7 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 ]

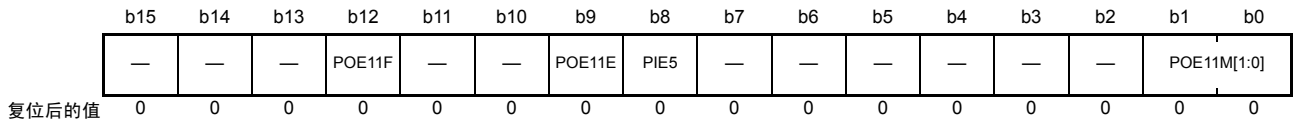
- 读“1”后写“0”时

[ 为“1”的条件 ]

- POE10# 引脚发生 POE10M[1:0] 位设定的输入时

### 17.2.8 输入电平控制 / 状态寄存器 5 (ICSR5)

地址 0008 C4D8h



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

## 17.2.9 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	MTUCH67HIZ	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 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	MTUCH67HIZ	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
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 的输出高阻抗允许位)

此位对 MTU 的互补 PWM 输出引脚 (MTIOC3B、MTIOC3D、MTIOC4A、MTIOC4B、MTIOC4C、MTIOC4D) 或者 GPT 大电流输出引脚 (GTIOC0A-A、GTIOC0B-A、GTIOC1A-A、GTIOC1B-A、GTIOC2A-A、GTIOC2B-A) 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

## MTUCH67HIZ 位 (MTU6、MTU7 的输出高阻抗允许位)

此位对 MTU 的互补 PWM 输出引脚 (MTIOC6B、MTIOC6D、MTIOC7A、MTIOC7B、MTIOC7C、MTIOC7D) 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 GPT0、GPT1 引脚 (GTIOC0A-B、GTIOC0B-B、GTIOC1A-B、GTIOC1B-B) 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 GPT2、GPT3 引脚 (GTIOC2A-B、GTIOC2B-B、GTIOC3A-B、GTIOC3B-B) 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

## 17.2.10 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
—	—	—	—	MTU0DZE	MTU0CZE	MTU0BZE	MTU0AZE
复位后的值	0	0	0	0	0	0	0

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、ICU.NMISR.OSTST 位和 POECR5 寄存器追加选择的 ICSRn.POE<sub>m</sub>F (n=1、2、4、5, m=0、4、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0~2、4~6) 标志中的任意一个为“1”时, 此位设定是否将 MTU0 引脚的 MTIOC0A-A 输出和 MTIOC0A-B 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、ICU.NMISR.OSTST 位和 POECR5 寄存器追加选择的 ICSRn.POE<sub>m</sub>F (n=1、2、4、5, m=0、4、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0~2、4~6) 标志中的任意一个为“1”时, 此位设定是否将 MTU0 引脚的 MTIOC0B-A 输出和 MTIOC0B-B 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、ICU.NMISR.OSTST 位和 POECR5 寄存器追加选择的 ICSRn.POE<sub>m</sub>F (n=1、2、4、5, m=0、4、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0~2、4~6) 标志中的任意一个为“1”时, 此位设定是否将 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、ICU.NMISR.OSTST 位和 POECR5 寄存器追加选择的 ICSRn.POE<sub>m</sub>F (n=1、2、4、5, m=0、4、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0~2、4~6) 标志中的任意一个为“1”时, 此位设定是否将 MTU0 引脚的 MTIOC0D 输出置为高阻抗状态。



## 17.2.11 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	—	—	—	—	—	MTU6BDZE	MTU7ACZE	MTU7BDZE	
复位后的值	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

位	符号	位名	功能	R/W
b0	MTU7BDZE	MTU CH7BD 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	MTU7ACZE	MTU CH7AC 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	MTU6BDZE	MTU CH6BD 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-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 MTU 的互补 PWM 输出引脚 (MTU3、MTU4、MTU6、MTU7 引脚) 和 GPT 大电流输出引脚 (GPT0 ~ GPT2 引脚) 进行高阻抗控制的寄存器。

## MTU7BDZE 位 (MTU CH7BD 的高阻抗允许位)

当通过 OCSR2.OSF2 标志、ICSR2.POE4F 标志、SPOER.MTUCH67HIZ 位、ICU.NMISR.OSTST 位和 POE2CR4 寄存器追加选择的 ICSRn.POE4mF (n=1、3~5, m=0、8、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0~2、4~6) 标志中的任意一个为“1”时, 此位设定是否将 MTU7 引脚的 MTIOC7B 输出和 MTIOC7D 输出置为高阻抗状态。

## MTU7ACZE 位 (MTU CH7AC 的高阻抗允许位)

当通过 OCSR2.OSF2 标志、ICSR2.POE4F 标志、SPOER.MTUCH67HIZ 位、ICU.NMISR.OSTST 位和 POE2CR4 寄存器追加选择的 ICSRn.POE4mF (n=1、3~5, m=0、8、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0~2、4~6) 标志中的任意一个为“1”时, 此位设定是否将 MTU7 引脚的 MTIOC7A 输出和 MTIOC7C 输出置为高阻抗状态。

## MTU6BDZE 位 (MTU CH6BD 的高阻抗允许位)

当通过 OCSR2.OSF2 标志、ICSR2.POE4F 标志、SPOER.MTUCH67HIZ 位、ICU.NMISR.OSTST 位和 POE2CR4 寄存器追加选择的 ICSRn.POE4mF (n=1、3~5, m=0、8、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0~2、4~6) 标志中的任意一个为“1”时, 此位设定是否将 MTU6 引脚的 MTIOC6B 输出和 MTIOC6D 输出置为高阻抗状态。

**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、ICU.NMISR.OSTST 位和 POE4CR4 寄存器追加选择的 ICSRn.POE<sub>m</sub>F (n=2 ~ 5, m=4、8、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0 ~ 2、4 ~ 6) 标志中的任意一个为“1”时, 此位设定是否将 MTU4/GPT2 引脚的 MTIOC4B 输出 / GTIOC2A-A 输出和 MTIOC4D 输出 /GTIOC2B-A 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、ICU.NMISR.OSTST 位和 POE4CR4 寄存器追加选择的 ICSRn.POE<sub>m</sub>F (n=2 ~ 5, m=4、8、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0 ~ 2、4 ~ 6) 标志中的任意一个为“1”时, 此位设定是否将 MTU4/GPT1 引脚的 MTIOC4A 输出 / GTIOC1A-A 输出和 MTIOC4C 输出 /GTIOC1B-A 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、ICU.NMISR.OSTST 位和 POE4CR4 寄存器追加选择的 ICSRn.POE<sub>m</sub>F (n=2 ~ 5, m=4、8、10、11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0 ~ 2、4 ~ 6) 标志中的任意一个为“1”时, 此位设定是否将 MTU3/GPT0 引脚的 MTIOC3B 输出 / GTIOC0A-A 输出和 MTIOC3D 输出 /GTIOC0B-A 输出置为高阻抗状态。

## 17.2.12 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.GPT01HIZ 位、ICU.NMISR.OSTST 位和 POE3 寄存器追加选择的 ICSRn.POE<sub>m</sub>F (n=1~3, 5, m=0, 4, 8, 11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0~2, 4~6) 标志中的任意一个为“1”时, 此位设定是否将 GPT0 引脚的 GTIOC0A-B 输出和 GTIOC0B-B 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.GPT01HIZ 位、ICU.NMISR.OSTST 位和 POE3 寄存器追加选择的 ICSRn.POE<sub>m</sub>F (n=1~3, 5, m=0, 4, 8, 11) 标志和 S12AD.ADCMPFR.CjFLAG (j=0~2, 4~6) 标志中的任意一个为“1”时, 此位设定是否将 GPT1 引脚的 GTIOC1A-B 输出和 GTIOC1B-B 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、ICU.NMISR.OSTST 位和 POE3 寄存器追加选择的 ICSRn.POE<sub>m</sub>F (n=1~4, m=0, 4, 8, 10) 标志和 S12AD.ADCMPFR.CjFLAG (j=0~2, 4~6) 标志中的任意一个为“1”时, 此位设定是否将 GPT2 引脚的 GTIOC2A-B 输出和 GTIOC2B-B 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、ICU.NMISR.OSTST 位和 POE3 寄存器追加选择的 ICSRn.POE<sub>m</sub>F (n=1~4, m=0, 4, 8, 10) 标志和 S12AD.ADCMPFR.CjFLAG (j=0~2, 4~6) 标志中的任意一个为“1”时, 此位设定是否将 GPT3 引脚的 GTIOC3A 输出和 GTIOC3B 输出置为高阻抗状态。

## 17.2.13 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 MT67ZE	IC4ADD MT67ZE	IC3ADD MT67ZE	—	IC1ADD MT67ZE	CMADD MT67ZE	—	—	IC5ADD MT34ZE	IC4ADD MT34ZE	IC3ADD MT34ZE	IC2ADD MT34ZE	—	CMADD MT34ZE
复位后的值	0	0	0	0	1	0	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	IC2ADDMT34ZE	MTU CH34 的高阻抗 POE4F 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)
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)
b7-b6	—	保留位	读写值都为“0”。	R/W
b8	CMADDMT67ZE	MTU CH67 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	IC1ADDMT67ZE	MTU CH67 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“1”。	R/W
b11	IC3ADDMT67ZE	MTU CH674 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	IC4ADDMT67ZE	MTU CH67 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	IC5ADDMT67ZE	MTU CH67 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-b14	—	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 大电流输出引脚 (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=0 ~ 2, 4 ~ 6) 标志追加到 MTU3、MTU4、GPT0 ~ GPT2 引脚 (MTIOC3B/MTIOC3D/GTIOC0A-A/GTIOC0B-A/MTIOC4A/MTIOC4C/GTIOC1A-A/GTIOC1B-A/MTIOC4B/MTIOC4D/GTIOC2A-A/GTIOC2B-A) 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 ~ 4) 中断。

**IC2ADDMT34ZE 位 (MTU CH34 的高阻抗 POE4F 追加位)**

将 ICSR2.POE4F 标志追加到 MTU3、MTU4、GPT0 ~ GPT2 引脚 (MTIOC3B/MTIOC3D/GTIOC0A-A/GTIOC0B-A/MTIOC4A/MTIOC4C/GTIOC1A-A/GTIOC1B-A/MTIOC4B/MTIOC4D/GTIOC2A-A/GTIOC2B-A) 的高阻抗控制条件。

**IC3ADDMT34ZE 位 (MTU CH34 的高阻抗 POE8F 追加位)**

将 ICSR3.POE8F 标志追加到 MTU3、MTU4、GPT0 ~ GPT2 引脚 (MTIOC3B/MTIOC3D/GTIOC0A-A/GTIOC0B-A/MTIOC4A/MTIOC4C/GTIOC1A-A/GTIOC1B-A/MTIOC4B/MTIOC4D/GTIOC2A-A/GTIOC2B-A) 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 MTU3、MTU4、GPT0 ~ GPT2 引脚 (MTIOC3B/MTIOC3D/GTIOC0A-A/GTIOC0B-A/MTIOC4A/MTIOC4C/GTIOC1A-A/GTIOC1B-A/MTIOC4B/MTIOC4D/GTIOC2A-A/GTIOC2B-A) 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 MTU3、MTU4、GPT0 ~ GPT2 引脚 (MTIOC3B/MTIOC3D/GTIOC0A-A/GTIOC0B-A/MTIOC4A/MTIOC4C/GTIOC1A-A/GTIOC1B-A/MTIOC4B/MTIOC4D/GTIOC2A-A/GTIOC2B-A) 的高阻抗控制条件。

**CMADDMT67ZE 位 (MTU CH67 的高阻抗 CFLAG 追加位)**

将 S12AD.ADCMPFR.CjFLAG (j=0 ~ 2、4 ~ 6) 标志追加到 MTU6、MTU7 引脚 (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D) 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 ~ 4) 中断。

**IC1ADDMT67ZE 位 (MTU CH67 的高阻抗 POE0F 追加位)**

将 ICSR1.POE0F 标志追加到 MTU6、MTU7 引脚 (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D) 的高阻抗控制条件。

**IC3ADDMT67ZE 位 (MTU CH67 的高阻抗 POE8F 追加位)**

将 ICSR3.POE8F 标志追加到 MTU6、MTU7 引脚 (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D) 的高阻抗控制条件。

**IC4ADDMT67ZE 位 (MTU CH67 的高阻抗 POE10F 追加位)**

将 ICSR4.POE10F 标志追加到 MTU6、MTU7 引脚 (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D) 的高阻抗控制条件。

**IC5ADDMT67ZE 位 (MTU CH67 的高阻抗 POE11F 追加位)**

将 ICSR5.POE11F 标志追加到 MTU6、MTU7 引脚 (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D) 的高阻抗控制条件。

## 17.2.14 端口输出允许控制寄存器 5 (POECR5)

地址 0008 C4D2h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	—	—	IC5ADD MT0ZE	IC4ADD MT0ZE	—	IC2ADD MT0ZE	IC1ADD MT0ZE	CMADD MT0ZE
复位后的值	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	IC2ADDMT0ZE	MTU CH0 的高阻抗 POE4F 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)
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。

POECR5 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=0 ~ 2, 4 ~ 6) 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 ~ 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。

## IC2ADDMT0ZE 位 (MTU CH0 的高阻抗 POE4F 追加位)

将 ICSR2.POE4F 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。

## 17.2.15 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
—	—	—	IC4ADD GPT23Z	IC3ADD GPT23Z	IC2ADD GPT23Z	IC1ADD GPT23Z	CMADD GPT23Z	—	—	IC5ADD GPT01Z	—	IC3ADD GPT01Z	IC2ADD GPT01Z	IC1ADD GPT01Z	CMADD GPT01Z
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	IC2ADDGPT01ZE	GPT CH01 的高阻抗 POE4F 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)
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	IC2ADDGPT23ZE	GPT CH23 的高阻抗 POE4F 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)
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=0 ~ 2, 4 ~ 6) 标志追加到 GPT0、GPT1 引脚 (GTIOC0A-B、GTIOC0B-B、GTIOC1A-B、GTIOC1B-B) 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 ~ 4) 中断。

## IC1ADDGPT01ZE 位 (GPT CH01 的高阻抗 POE0F 追加位)

将 ICSR1.POE0F 标志追加到 GPT0、GPT1 引脚 (GTIOC0A-B、GTIOC0B-B、GTIOC1A-B、GTIOC1B-B) 的高阻抗控制条件。

**IC2ADDGPT01ZE 位 (GPT CH01 的高阻抗 POE4F 追加位)**

将 ICSR2.POE4F 标志追加到 GPT0、GPT1 引脚 (GTIOC0A-B、GTIOC0B-B、GTIOC1A-B、GTIOC1B-B) 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 GPT0、GPT1 引脚 (GTIOC0A-B、GTIOC0B-B、GTIOC1A-B、GTIOC1B-B) 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 GPT0、GPT1 引脚 (GTIOC0A-B、GTIOC0B-B、GTIOC1A-B、GTIOC1B-B) 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=0 ~ 2、4 ~ 6) 标志追加到 GPT2、GPT3 引脚 (GTIOC2A-B、GTIOC2B-B、GTIOC3A-B、GTIOC3B-B) 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 ~ 4) 中断。

**IC1ADDGPT23ZE 位 (GPT CH23 的高阻抗 POE0F 追加位)**

将 ICSR1.POE0F 标志追加到 GPT2、GPT3 引脚 (GTIOC2A-B、GTIOC2B-B、GTIOC3A-B、GTIOC3B-B) 的高阻抗控制条件。

**IC2ADDGPT23ZE 位 (GPT CH23 的高阻抗 POE4F 追加位)**

将 ICSR2.POE4F 标志追加到 GPT2、GPT3 引脚 (GTIOC2A-B、GTIOC2B-B、GTIOC3A-B、GTIOC3B-B) 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 GPT2、GPT3 引脚 (GTIOC2A-B、GTIOC2B-B、GTIOC3A-B、GTIOC3B-B) 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 GPT2、GPT3 引脚 (GTIOC2A-B、GTIOC2B-B、GTIOC3A-B、GTIOC3B-B) 的高阻抗控制条件。



## 17.3 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 17.5 所示。

表 17.5 高阻抗控制的对象引脚及其条件 (1/5)

引脚	条件	详细内容
MTU3B/MTU3D 引脚 (MTIOC3B、MTIOC3D)	<ul style="list-style-type: none"> <li>• POE0# 引脚的输入电平检测</li> <li>• MTIOC3B 输出和 MTIOC3D 输出的输出电平比较</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR4 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12 位 A/D 转换器的比较器检测</li> <li>- POE4# 引脚、POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU3BDZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICU.NMISR.OSTST) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC2ADDMT34ZE•POE4F) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)
MTU4A/MTU4C 引脚 (MTIOC4A、MTIOC4C)	<ul style="list-style-type: none"> <li>• POE0# 引脚的输入电平检测</li> <li>• MTIOC4A 输出和 MTIOC4C 输出的输出电平比较</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR4 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12 位 A/D 转换器的比较器检测</li> <li>- POE4# 引脚、POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU4ACZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICU.NMISR.OSTST) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC2ADDMT34ZE•POE4F) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)
MTU4B/MTU4D 引脚 (MTIOC4B、MTIOC4D)	<ul style="list-style-type: none"> <li>• POE0# 引脚的输入电平检测</li> <li>• MTIOC4B 输出和 MTIOC4D 输出的输出电平比较</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR4 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12 位 A/D 转换器的比较器检测</li> <li>- POE4# 引脚、POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU4BDZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICU.NMISR.OSTST) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC2ADDMT34ZE•POE4F) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)

表 17.5 高阻抗控制的对象引脚及其条件 (2/5)

引脚	条件	详细内容
MTU6B/MTU6D 引脚 (MTIOC6B、MTIOC6D)	<ul style="list-style-type: none"> <li>• POE4# 引脚的输入电平检测</li> <li>• MTIOC6B 输出和 MTIOC6D 输出的输出电平比较</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POE4# 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12 位 A/D 转换器的比较器检测</li> <li>- POE0# 引脚、POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU6BDZE• ((POE4F +(OSF2•OCE2) +(MTUCH67HIZ) +(ICU.NMISR.OSTST) +(CMADDMT67ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDMT67ZE•POE0F) +(IC3ADDMT67ZE•POE8E•POE8F) +(IC4ADDMT67ZE•POE10E•POE10F) +(IC5ADDMT67ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)
MTU7A/MTU7C 引脚 (MTIOC7A、MTIOC7C)	<ul style="list-style-type: none"> <li>• POE4# 引脚的输入电平检测</li> <li>• MTIOC7A 输出和 MTIOC7C 输出的输出电平比较</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POE4# 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12 位 A/D 转换器的比较器检测</li> <li>- POE0# 引脚、POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU7ACZE• ((POE4F) +(OSF2•OCE2) +(MTUCH67HIZ) +(ICU.NMISR.OSTST) +(CMADDMT67ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDMT67ZE•POE0F) +(IC3ADDMT67ZE•POE8E•POE8F) +(IC4ADDMT67ZE•POE10E•POE10F) +(IC5ADDMT67ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)
MTU7B/MTU7D 引脚 (MTIOC7B、MTIOC7D)	<ul style="list-style-type: none"> <li>• POE4# 引脚的输入电平检测</li> <li>• MTIOC7B 输出和 MTIOC7D 输出的输出电平比较</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POE4# 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12 位 A/D 转换器的比较器检测</li> <li>- POE0# 引脚、POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU7BDZE• ((POE4F +(OSF2•OCE2) +(MTUCH67HIZ) +(ICU.NMISR.OSTST) +(CMADDMT67ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDMT67ZE•POE0F) +(IC3ADDMT67ZE•POE8E•POE8F) +(IC4ADDMT67ZE•POE10E•POE10F) +(IC5ADDMT67ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)
GPT0 引脚 (GTIOC0A-A、GTIOC0B-A)	<ul style="list-style-type: none"> <li>• POE0# 引脚的输入电平检测</li> <li>• GTIOC0A-A 输出和 GTIOC0B-A 输出的输出电平比较</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POE4# 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12 位 A/D 转换器的比较器检测</li> <li>- POE4# 引脚、POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU3BDZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICU.NMISR.OSTST) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC2ADDMT34ZE•POE4F) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)

表 17.5 高阻抗控制的对象引脚及其条件 (3/5)

引脚	条件	详细内容
GPT1 引脚 (GTIOC1A-A、GTIOC1B-A)	<ul style="list-style-type: none"> <li>• POE0# 引脚的输入电平检测</li> <li>• GTIOC1A-A 输出和 GTIOC1B-A 输出的输出电平比较</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR4 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12 位 A/D 转换器的比较器检测</li> <li>- POE4# 引脚、POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU4ACZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICU.NMISR.OSTST) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC2ADDMT34ZE•POE4F) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)
GPT2 引脚 (GTIOC2A-A、GTIOC2B-A)	<ul style="list-style-type: none"> <li>• POE0# 引脚的输入电平检测</li> <li>• GTIOC2A-A 输出和 GTIOC2B-A 输出的输出电平比较</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR4 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12 位 A/D 转换器的比较器检测</li> <li>- POE4# 引脚、POE8# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU4BDZE• ((POE0F) +(OSF1•OCE1) +(MTUCH34HIZ) +(ICU.NMISR.OSTST) +(CMADDMT34ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC2ADDMT34ZE•POE4F) +(IC3ADDMT34ZE•POE8E•POE8F) +(IC4ADDMT34ZE•POE10E•POE10F) +(IC5ADDMT34ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)
MTU0A 引脚 (MTIOC0A-A、MTIOC0A-B)	<ul style="list-style-type: none"> <li>• POE8# 引脚的输入电平检测</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR5 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12 位 A/D 转换器的比较器检测</li> <li>- POE0# 引脚、POE4# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU0AZE• ((POE8F•POE8E) +(MTUCH0HIZ) +(ICU.NMISR.OSTST) +(CMADDMT0ZE• S12AD.ADCMPSEL.POE•+ (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDMT0ZE•POE0F) +(IC2ADDMT0ZE•POE4F) +(IC4ADDMT0ZE•POE10E•POE10F) +(IC5ADDMT0ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)
MTU0B 引脚 (MTIOC0B-A、MTIOC0B-B)	<ul style="list-style-type: none"> <li>• POE8# 引脚的输入电平检测</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR5 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12 位 A/D 转换器的比较器检测</li> <li>- POE0# 引脚、POE4# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU0BZE• ((POE8F•POE8E) +(MTUCH0HIZ) +(ICU.NMISR.OSTST) +(CMADDMT0ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDMT0ZE•POE0F) +(IC2ADDMT0ZE•POE4F) +(IC4ADDMT0ZE•POE10E•POE10F) +(IC5ADDMT0ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)

表 17.5 高阻抗控制的对象引脚及其条件 (4/5)

引脚	条件	详细内容
MTU0C 引脚 (MTIOC0C)	<ul style="list-style-type: none"> <li>• POE8# 引脚的输入电平检测</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR5 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12位 A/D 转换器的比较器检测</li> <li>- POE0# 引脚、POE4# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU0CZE• ((POE8F•POE8E) +(MTUCH0HIZ) +(ICU.NMISR.OSTST) +(CMADDMT0ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDMT0ZE•POE0F) +(IC2ADDMT0ZE•POE4F) +(IC4ADDMT0ZE•POE10E•POE10F) +(IC5ADDMT0ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)
MTU0D 引脚 (MTIOC0D)	<ul style="list-style-type: none"> <li>• POE8# 引脚的输入电平检测</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR5 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12位 A/D 转换器的比较器检测</li> <li>- POE0# 引脚、POE4# 引脚、POE10# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	MTU0DZE• ((POE8F•POE8E) +(MTUCH0HIZ) +(ICU.NMISR.OSTST) +(CMADDMT0ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDMT0ZE•POE0F) +(IC2ADDMT0ZE•POE4F) +(IC4ADDMT0ZE•POE10E•POE10F) +(IC5ADDMT0ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)
GPT0 引脚 (GTIOC0A-B、GTIOC0B-B)	<ul style="list-style-type: none"> <li>• POE10# 引脚的输入电平检测</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR6 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12位 A/D 转换器的比较器检测</li> <li>- POE0# 引脚、POE4# 引脚、POE8# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	GPT0ABZE• ((POE10F•POE10E) +(GPT01HIZ) +(ICU.NMISR.OSTST) +(CMADDGPT01ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDGPT01ZE•POE0F) +(IC2ADDGPT01ZE•POE4F) +(IC3ADDGPT01ZE•POE8E•POE8F) +(IC5ADDGPT01ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)
GPT1 引脚 (GTIOC1A-B、GTIOC1B-B)	<ul style="list-style-type: none"> <li>• POE10# 引脚的输入电平检测</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR6 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12位 A/D 转换器的比较器检测</li> <li>- POE0# 引脚、POE4# 引脚、POE8# 引脚、POE11# 引脚的输入电平检测</li> </ul> </li> </ul>	GPT1ABZE• ((POE10F•POE10E) +(GPT01HIZ) +(ICU.NMISR.OSTST) +(CMADDGPT01ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDGPT01ZE•POE0F) +(IC2ADDGPT01ZE•POE4F) +(IC3ADDGPT01ZE•POE8E•POE8F) +(IC5ADDGPT01ZE•POE11E•POE11F)) (n=0 ~ 2、4 ~ 6)

表 17.5 高阻抗控制的对象引脚及其条件 (5/5)

引脚	条件	详细内容
GPT2 引脚 (GTIOC2A-B、GTIOC2B-B)	<ul style="list-style-type: none"> <li>• POE11# 引脚的输入电平检测</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR6 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12位 A/D 转换器的比较器检测</li> <li>- POE0# 引脚、POE4# 引脚、POE8# 引脚、POE10# 引脚的输入电平检测</li> </ul> </li> </ul>	GPT2ABZE• ((POE11F•POE11E) +(GPT23HIZ) +(ICU.NMISR.OSTST) +(CMADDGPT23ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDGPT23ZE•POE0F) +(IC2ADDGPT23ZE•POE4F) +(IC3ADDGPT23ZE•POE8E•POE8F) +(IC4DDGPT23ZE•POE10E•POE10F)) (n=0 ~ 2、4 ~ 6)
GPT3 引脚 (GTIOC3A、GTIOC3B)	<ul style="list-style-type: none"> <li>• POE11# 引脚的输入电平检测</li> <li>• SPOER 寄存器设定</li> <li>• 振荡停止检测</li> <li>• 通过 POECR6 寄存器追加的条件               <ul style="list-style-type: none"> <li>- 12位 A/D 转换器的比较器检测</li> <li>- POE0# 引脚、POE4# 引脚、POE8# 引脚、POE10# 引脚的输入电平检测</li> </ul> </li> </ul>	GPT3ABZE• ((POE11F•POE11E) +(GPT23HIZ) +(ICU.NMISR.OSTST) +(CMADDGPT23ZE• S12AD.ADCMPSEL.POE• (S12AD.ADCMPFR.CnFLAG)) +(IC1ADDGPT23ZE•POE0F) +(IC2ADDGPT23ZE•POE4F) +(IC3ADDGPT23ZE•POE8E•POE8F) +(IC4ADDGPT23ZE•POE10E•POE10F)) (n=0 ~ 2、4 ~ 6)

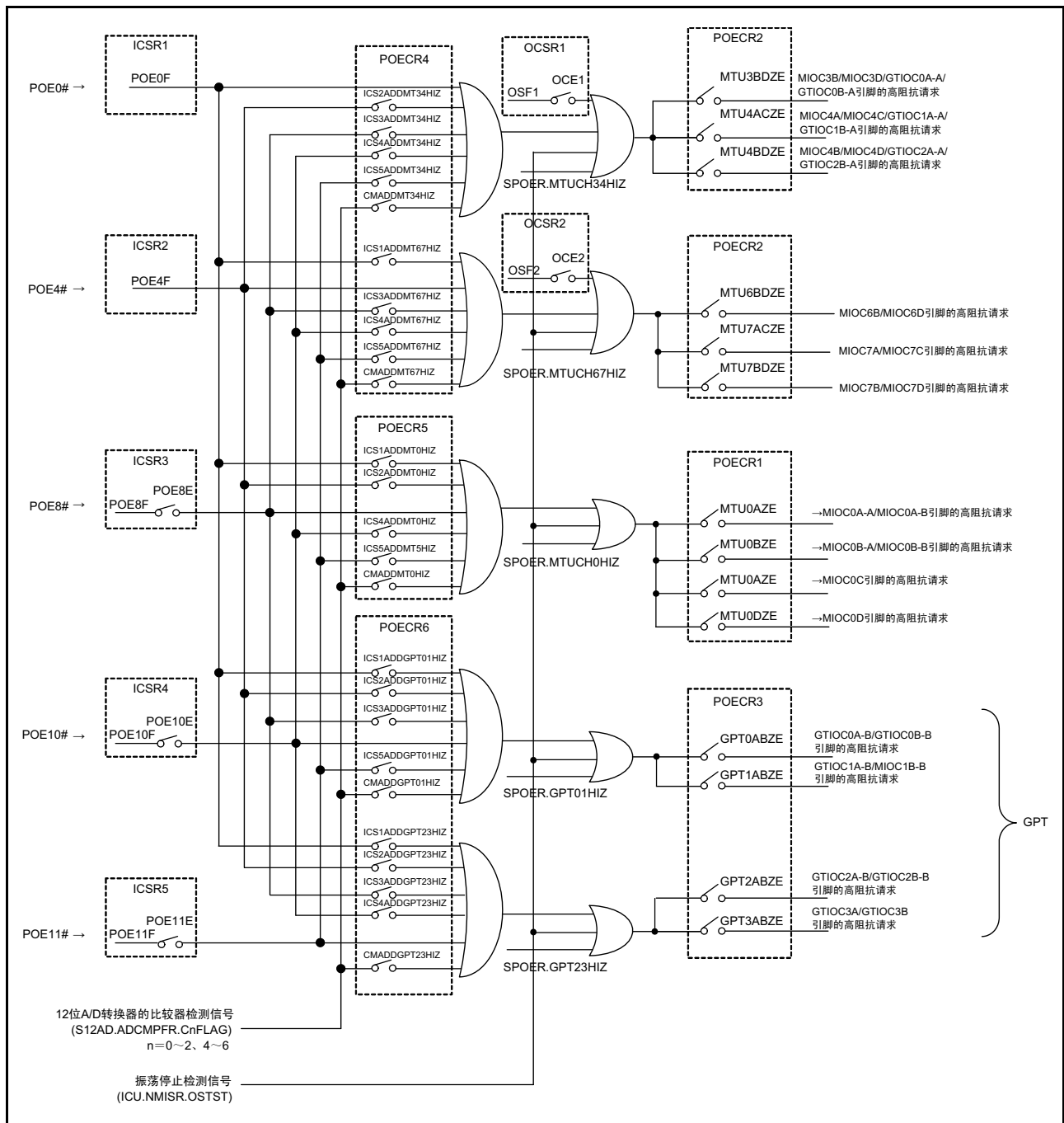


图 17.2 高阻抗控制的对象引脚和条件

### 17.3.1 输入电平的检测

当 POE0#、POE4#、POE8#、POE10#、POE11# 引脚产生 ICSR1 ~ 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 引脚和 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#、POE4#、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#、POE4#、POE8#、POE10#、POE11# 引脚输入到将引脚置为高阻抗的时序例子如图 17.3 所示。

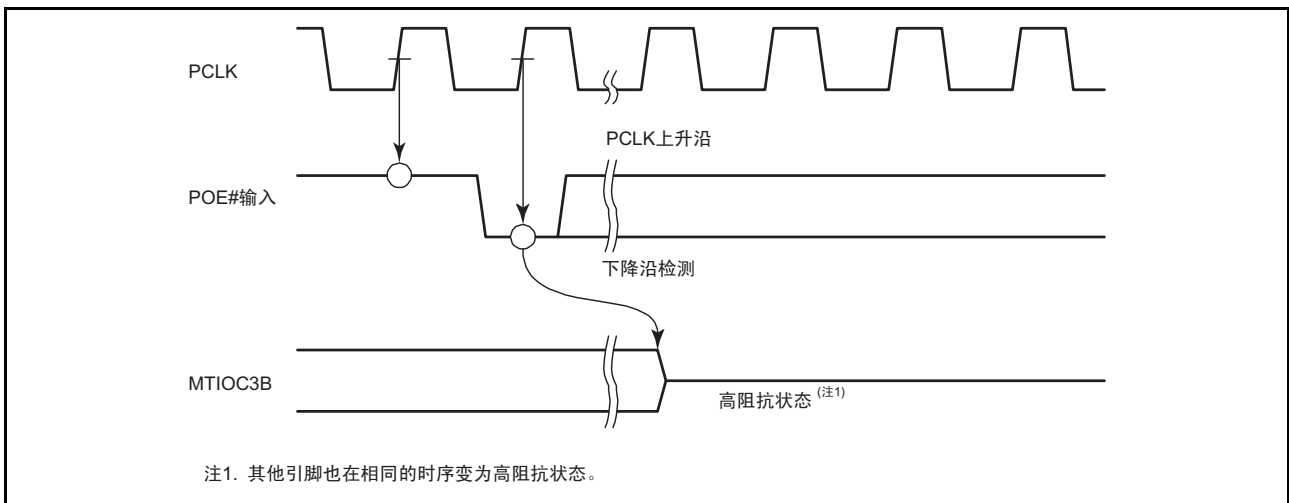


图 17.3 下降沿检测

#### (2) Low 电平检测

Low 电平检测如图 17.4 所示。通过 ICSR1 ~ 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。

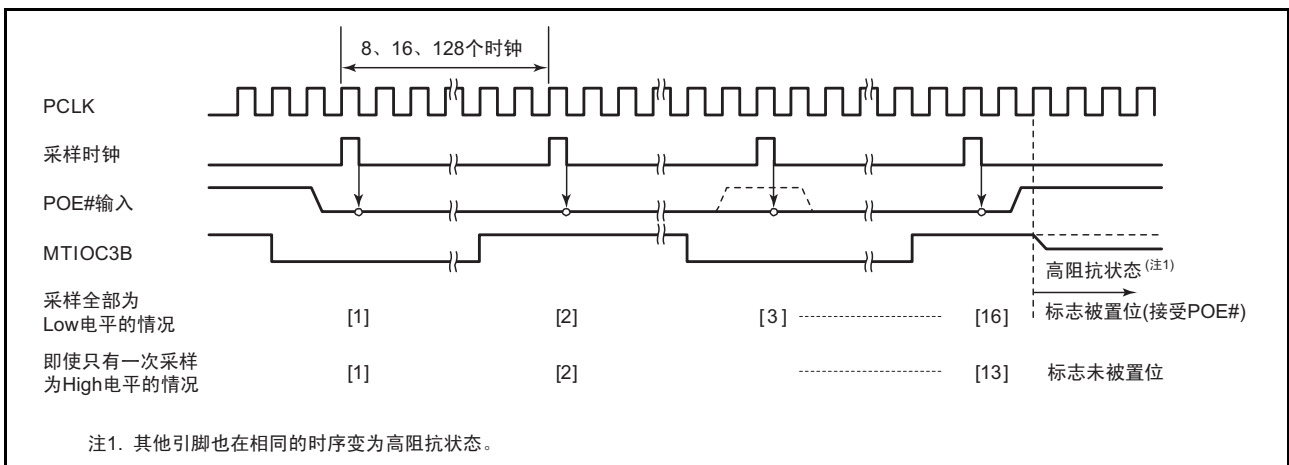


图 17.4 Low 电平检测

### 17.3.2 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 17.5 所示，其他引脚的组合也一样。

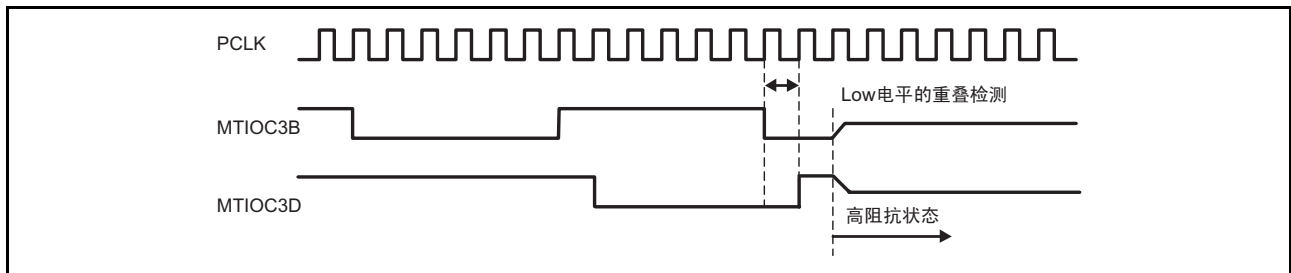


图 17.5 输出电平的比较

### 17.3.3 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。

### 17.3.4 通过振荡停止检测进行的高阻抗控制

如果通过时钟发生电路的振荡停止检测电路检测到振荡停止，就将端口输出允许控制寄存器 2 (POECR2) 设定的 MTU3、MTU4、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。

### 17.3.5 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=0~2、4~6) 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。

### 17.3.6 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.IC2ADDMT34ZE 位置“1”，追加通过 POE4# 引脚进行的输入电平检测
- 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。



### 17.3.7 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、ICSR2.POE4F 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、ICSR2.POE4M[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#、POE4#、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、OCSR2.OSF2 标志置“0”，解除高阻抗状态。但是，如果不是在从引脚输出了无效电平以后，即使给标志写“0”也无效并且标志不变为“0”。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、ICSR2.POE4F 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=0~2、4~6) 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”。

## 17.4 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 17.6 所示。

表 17.6 中断请求的种类和条件

名称	中断源	中断标志	条件
OEI1	输出允许中断 1	POE0F、OSF1	PIE1•POE0F+OIE1•OSF1
OEI2	输出允许中断 2	POE4F、OSF2	PIE2•POE4F+OIE2•OSF2
OEI3	输出允许中断 3	POE8F	PIE3•POE8F
OEI4	输出允许中断 4	POE10F、POE11F	PIE4•POE10F+PIE5•POE11F

### 17.5 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。

## 18. 通用 PWM 定时器 (GPT)

RX62T 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。

### 18.1 概要

GPT 的规格如表 18.1、GPT 的功能一览表如表 18.2、GPT 的框图如图 18.1 所示。

表 18.1 GPT 的规格

项目	内容
功能	<ul style="list-style-type: none"> <li>• 16 位 ×4 个通道</li> <li>• 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)</li> <li>• 能给各通道选择独立的时钟源</li> <li>• 每个通道有 2 个输入 / 输出引脚</li> <li>• 每个通道有 2 个用于输出比较 / 输入捕捉的寄存器</li> <li>• 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。</li> <li>• 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。</li> <li>• 给各通道装载用于帧周期的寄存器 (能通过上溢 / 下溢中断)</li> <li>• 能同步运行各计数器</li> <li>• 同步运行模式 (支持同时或者以任意的时序进行相位移位)</li> <li>• 能在 PWM 运行时生成死区时间</li> <li>• 组合 3 个计数器, 能生成带死区时间的三相 PWM 波形</li> <li>• 能通过外部 / 内部触发开始、清除或者停止计数</li> <li>• 内部触发源有比较器检测、软件和比较匹配</li> <li>• 能通过分频系统时钟 (ICLK) 的计数时钟来测量被分频的 IWDG 专用低速内部振荡器时钟的边沿 (振荡异常检测)</li> </ul>

表 18.2 GPT 的功能一览表 (1/2)

项目	GPT0	GPT1	GPT2	GPT3
计数时钟	ICLK ICLK/2 ICLK/4 ICLK/8	ICLK ICLK/2 ICLK/4 ICLK/8	ICLK ICLK/2 ICLK/4 ICLK/8	ICLK ICLK/2 ICLK/4 ICLK/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 18.2 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 /LOCO 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

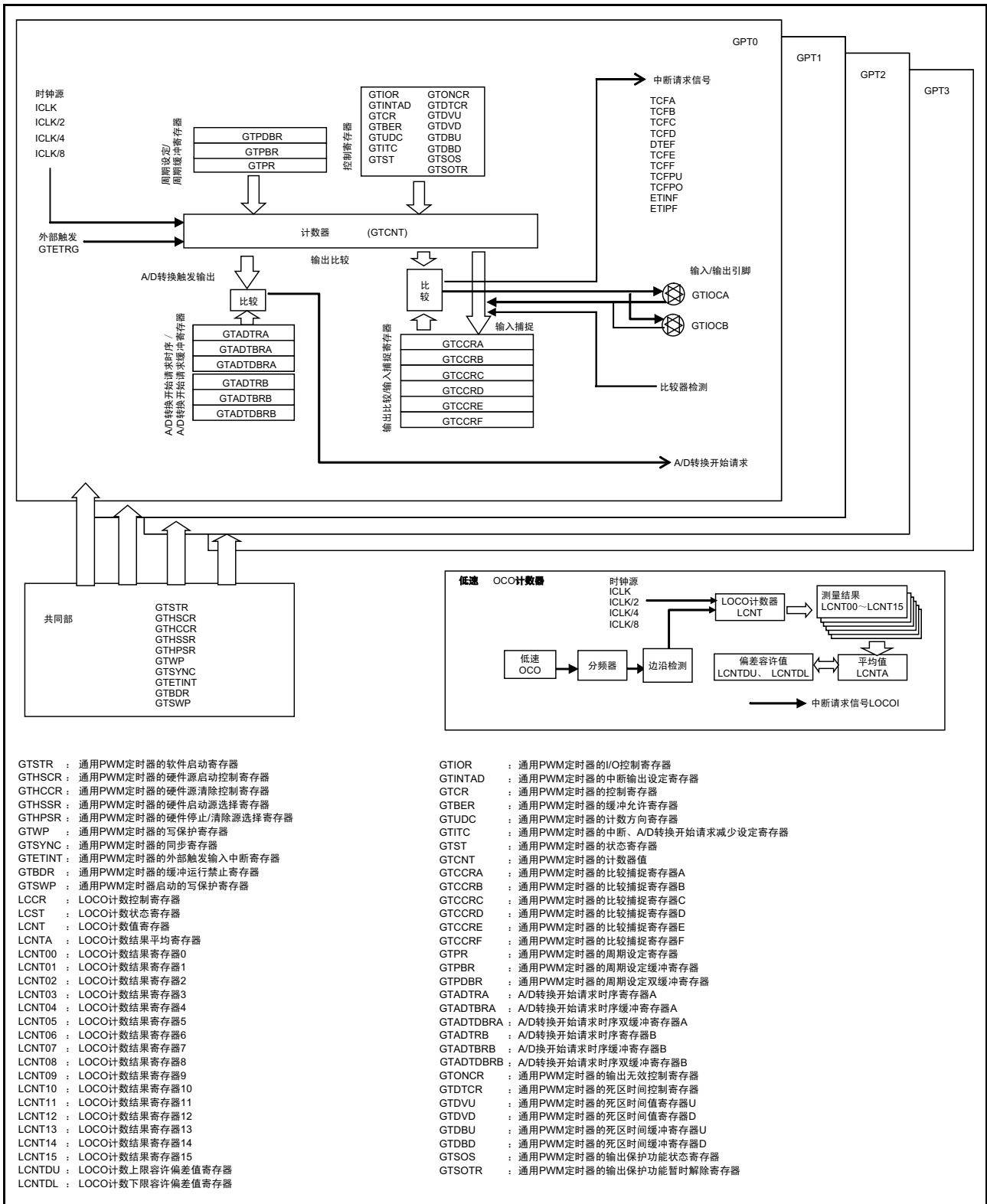


图 18.1 GPT 的框图

GPT 使用的输入 / 输出引脚如表 18.3 所示。

表 18.3 GPT 的输入 / 输出引脚

通道	引脚名	输入 / 输出	功能
GPT	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

## 18.2 寄存器说明

GPT 的寄存器一览表如表 18.4 所示。

表 18.4 MTU 的寄存器一览表 (1/4)

通道	寄存器名	寄存器符号	复位后的值	地址	存取长度
GPT	通用 PWM 定时器的软件启动寄存器	GTSTR	0000h	000C 2000h	8、16、32
	通用 PWM 定时器的硬件源启动控制寄存器	GTHSCR	0000h	000C 2004h	8、16、32
	通用 PWM 定时器的硬件源清除控制寄存器	GTHCCR	0000h	000C 2006h	8、16、32
	通用 PWM 定时器的硬件启动源选择寄存器	GTHSSR	0000h	000C 2008h	8、16、32
	通用 PWM 定时器的硬件停止 / 清除源选择寄存器	GTHPSR	0000h	000C 200Ah	8、16、32
	通用 PWM 定时器的写保护寄存器	GTWP	0000h	000C 200Ch	8、16、32
	通用 PWM 定时器的同步寄存器	GTSYNC	0000h	000C 200Eh	8、16、32
	通用 PWM 定时器的外部触发输入中断寄存器	GTETINT	0000h	000C 2010h	8、16、32
	通用 PWM 定时器的缓冲运行禁止寄存器	GTBDR	0000h	000C 2014h	8、16、32
	通用 PWM 定时器启动的写保护寄存器	GTSWP	0000h	000C 2018h	8、16、32
	LOCO 计数控制寄存器	LCCR	0000h	000C 2080h	8、16、32
	LOCO 计数状态寄存器	LCST	0000h	000C 2082h	8、16、32
	LOCO 计数值寄存器	LCNT	0000h	000C 2084h	8、16、32
	LOCO 计数结果平均寄存器	LCNTA	0000h	000C 2086h	8、16、32
	LOCO 计数结果寄存器 0	LCNT00	0000h	000C 2088h	8、16、32
	LOCO 计数结果寄存器 1	LCNT01	0000h	000C 208Ah	8、16、32
	LOCO 计数结果寄存器 2	LCNT02	0000h	000C 208Ch	8、16、32
	LOCO 计数结果寄存器 3	LCNT03	0000h	000C 208Eh	8、16、32
	LOCO 计数结果寄存器 4	LCNT04	0000h	000C 2090h	8、16、32
	LOCO 计数结果寄存器 5	LCNT05	0000h	000C 2092h	8、16、32
	LOCO 计数结果寄存器 6	LCNT06	0000h	000C 2094h	8、16、32
	LOCO 计数结果寄存器 7	LCNT07	0000h	000C 2096h	8、16、32
	LOCO 计数结果寄存器 8	LCNT08	0000h	000C 2098h	8、16、32
	LOCO 计数结果寄存器 9	LCNT09	0000h	000C 209Ah	8、16、32
	LOCO 计数结果寄存器 10	LCNT10	0000h	000C 209Ch	8、16、32
	LOCO 计数结果寄存器 11	LCNT11	0000h	000C 209Eh	8、16、32
	LOCO 计数结果寄存器 12	LCNT12	0000h	000C 20A0h	8、16、32
	LOCO 计数结果寄存器 13	LCNT13	0000h	000C 20A2h	8、16、32
	LOCO 计数结果寄存器 14	LCNT14	0000h	000C 20A4h	8、16、32
	LOCO 计数结果寄存器 15	LCNT15	0000h	000C 20A6h	8、16、32
LOCO 计数上限容许偏差值寄存器	LCNTDU	FFFFh	000C 20A8h	8、16、32	
LOCO 计数下限容许偏差置寄存器	LCNTDL	FFFFh	000C 20AAh	8、16、32	
GPT0	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	0000h	000C 2100h	8、16、32
	通用 PWM 定时器的中断输出设定寄存器	GTINTAD	0000h	000C 2102h	8、16、32
	通用 PWM 定时器的控制寄存器	GTCR	0000h	000C 2104h	8、16、32
	通用 PWM 定时器的缓冲允许寄存器	GTBER	0000h	000C 2106h	8、16、32
	通用 PWM 定时器的计数方向寄存器	GTUDC	0001h	000C 2108h	8、16、32
	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	0000h	000C 210Ah	8、16、32
	通用 PWM 定时器的状态寄存器	GTST	8000h	000C 210Ch	8、16、32
	通用 PWM 定时器的计数器	GTCNT	0000h	000C 210Eh	16
	通用 PWM 定时器的比较捕捉寄存器 A	GTCCRA	FFFFh	000C 2110h	16、32
	通用 PWM 定时器的比较捕捉寄存器 B	GTCCRB	FFFFh	000C 2112h	16、32
	通用 PWM 定时器的比较捕捉寄存器 C	GTCCRC	FFFFh	000C 2114h	16、32
	通用 PWM 定时器的比较捕捉寄存器 D	GTCCRD	FFFFh	000C 2116h	16、32

表 18.4 MTU 的寄存器一览表 (2/4)

通道	寄存器名	寄存器符号	复位后的值	地址	存取长度
GPT0	通用 PWM 定时器的比较捕捉寄存器 E	GTCCRE	FFFFh	000C 2118h	16、32
	通用 PWM 定时器的比较捕捉寄存器 F	GTCCRF	FFFFh	000C 211Ah	16、32
	通用 PWM 定时器的周期设定寄存器	GTPR	FFFFh	000C 211Ch	16、32
	通用 PWM 定时器的周期设定缓冲寄存器	GTPBR	FFFFh	000C 211Eh	16、32
	通用 PWM 定时器的周期设定双缓冲寄存器	GTPDBR	FFFFh	000C 2120h	16、32
	A/D 转换开始请求时序寄存器 A	GTADTRA	FFFFh	000C 2124h	16、32
	A/D 转换开始请求时序缓冲寄存器 A	GTADTBRA	FFFFh	000C 2126h	16、32
	A/D 转换开始请求时序双缓冲寄存器 A	GTADTDBRA	FFFFh	000C 2128h	16、32
	A/D 转换开始请求时序寄存器 B	GTADTRB	FFFFh	000C 212Ch	16、32
	A/D 转换开始请求时序缓冲寄存器 B	GTADTBRB	FFFFh	000C 212Eh	16、32
	A/D 转换开始请求时序双缓冲寄存器 B	GTADTDBRB	FFFFh	000C 2130h	16、32
	通用 PWM 定时器的输出无效控制寄存器	GTONCR	0100h	000C 2134h	16、32
	通用 PWM 定时器的死区时间控制寄存器	GTDTCR	0000h	000C 2136h	16、32
	通用 PWM 定时器的死区时间值寄存器 U	GTDVU	FFFFh	000C 2138h	16、32
	通用 PWM 定时器的死区时间值寄存器 D	GTDVD	FFFFh	000C 213Ah	16、32
	通用 PWM 定时器的死区时间缓冲寄存器 U	GTDBU	FFFFh	000C 213Ch	16、32
	通用 PWM 定时器的死区时间缓冲寄存器 D	GTDBD	FFFFh	000C 213Eh	16、32
	通用 PWM 定时器的输出保护功能状态寄存器	GTSOS	0x00h	000C 2140h	16、32
	通用 PWM 定时器的输出保护暂时解除寄存器	GTSOTR	0000h	000C 2142h	16、32
	GPT1	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	0000h	000C 2180h
通用 PWM 定时器的中断输出设定寄存器		GTINTAD	0000h	000C 2182h	8、16、32
通用 PWM 定时器的控制寄存器		GTCR	0000h	000C 2184h	8、16、32
通用 PWM 定时器的缓冲允许寄存器		GTBER	0000h	000C 2186h	8、16、32
通用 PWM 定时器的计数方向寄存器		GTUDC	0001h	000C 2188h	8、16、32
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	0000h	000C 218Ah	8、16、32
通用 PWM 定时器的状态寄存器		GTST	8000h	000C 218Ch	8、16、32
通用 PWM 定时器的计数器		GTCNT	0000h	000C 218Eh	16
通用 PWM 定时器的比较捕捉寄存器 A		GTCCRA	FFFFh	000C 2190h	16、32
通用 PWM 定时器的比较捕捉寄存器 B		GTCCRB	FFFFh	000C 2192h	16、32
通用 PWM 定时器的比较捕捉寄存器 C		GTCCRC	FFFFh	000C 2194h	16、32
通用 PWM 定时器的比较捕捉寄存器 D		GTCCRD	FFFFh	000C 2196h	16、32
通用 PWM 定时器的比较捕捉寄存器 E		GTCCRE	FFFFh	000C 2198h	16、32
通用 PWM 定时器的比较捕捉寄存器 F		GTCCRF	FFFFh	000C 219Ah	16、32
通用 PWM 定时器的周期设定寄存器		GTPR	FFFFh	000C 219Ch	16、32
通用 PWM 定时器的周期设定缓冲寄存器		GTPBR	FFFFh	000C 219Eh	16、32
通用 PWM 定时器的周期设定双缓冲寄存器		GTPDBR	FFFFh	000C 21A0h	16、32
A/D 转换开始请求时序寄存器 A		GTADTRA	FFFFh	000C 21A4h	16、32
A/D 转换开始请求时序缓冲寄存器 A		GTADTBRA	FFFFh	000C 21A6h	16、32
A/D 转换开始请求时序双缓冲寄存器 A		GTADTDBRA	FFFFh	000C 21A8h	16、32
A/D 转换开始请求时序寄存器 B		GTADTRB	FFFFh	000C 21ACh	16、32
A/D 转换开始请求时序缓冲寄存器 B		GTADTBRB	FFFFh	000C 21AEh	16、32
A/D 转换开始请求时序双缓冲寄存器 B		GTADTDBRB	FFFFh	000C 21B0h	16、32
通用 PWM 定时器的输出无效控制寄存器		GTONCR	0100h	000C 21B4h	16、32
通用 PWM 定时器的死区时间控制寄存器		GTDTCR	0000h	000C 21B6h	16、32
通用 PWM 定时器的死区时间值寄存器 U		GTDVU	FFFFh	000C 21B8h	16、32
通用 PWM 定时器的死区时间值寄存器 D		GTDVD	FFFFh	000C 21BAh	16、32
通用 PWM 定时器的死区时间缓冲寄存器 U		GTDBU	FFFFh	000C 21BCh	16、32
通用 PWM 定时器的死区时间缓冲寄存器 D		GTDBD	FFFFh	000C 21BEh	16、32

表 18.4 MTU 的寄存器一览表 (3/4)

通道	寄存器名	寄存器符号	复位后的值	地址	存取长度
GPT1	通用 PWM 定时器的输出保护功能状态寄存器	GTSOS	0x00h	000C 21C0h	16、32
	通用 PWM 定时器的输出保护暂时解除功能	GTSOTR	0000h	000C 21C2h	16、32
GPT2	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	0000h	000C 2200h	8、16、32
	通用 PWM 定时器的中断输出设定寄存器	GTINTAD	0000h	000C 2202h	8、16、32
	通用 PWM 定时器的控制寄存器	GTCR	0000h	000C 2204h	8、16、32
	通用 PWM 定时器的缓冲允许寄存器	GTBER	0000h	000C 2206h	8、16、32
	通用 PWM 定时器的计数方向寄存器	GTUDC	0001h	000C 2208h	8、16、32
	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	0000h	000C 220Ah	8、16、32
	通用 PWM 定时器的状态寄存器	GTST	8000h	000C 220Ch	8、16、32
	通用 PWM 定时器的计数器	GTCNT	0000h	000C 220Eh	16
	通用 PWM 定时器的比较捕捉寄存器 A	GTCCRA	FFFFh	000C 2210h	16、32
	通用 PWM 定时器的比较捕捉寄存器 B	GTCCRB	FFFFh	000C 2212h	16、32
	通用 PWM 定时器的比较捕捉寄存器 C	GTCCRC	FFFFh	000C 2214h	16、32
	通用 PWM 定时器的比较捕捉寄存器 D	GTCCRD	FFFFh	000C 2216h	16、32
	通用 PWM 定时器的比较捕捉寄存器 E	GTCCRE	FFFFh	000C 2218h	16、32
	通用 PWM 定时器的比较捕捉寄存器 F	GTCCRF	FFFFh	000C 221Ah	16、32
	通用 PWM 定时器的周期设定寄存器	GTPR	FFFFh	000C 221Ch	16、32
	通用 PWM 定时器的周期设定缓冲寄存器	GTPBR	FFFFh	000C 221Eh	16、32
	通用 PWM 定时器的周期设定双缓冲寄存器	GTPDBR	FFFFh	000C 2220h	16、32
	A/D 转换开始请求时序寄存器 A	GTADTRA	FFFFh	000C 2224h	16、32
	A/D 转换开始请求时序缓冲寄存器 A	GTADTBRA	FFFFh	000C 2226h	16、32
	A/D 转换开始请求时序双缓冲寄存器 A	GTADTDBRA	FFFFh	000C 2228h	16、32
	A/D 转换开始请求时序寄存器 B	GTADTRB	FFFFh	000C 222Ch	16、32
	A/D 转换开始请求时序缓冲寄存器 B	GTADTBRB	FFFFh	000C 222Eh	16、32
	A/D 转换开始请求时序双缓冲寄存器 B	GTADTDBRB	FFFFh	000C 2230h	16、32
	通用 PWM 定时器的输出无效控制寄存器	GTONCR	0100h	000C 2234h	16、32
	通用 PWM 定时器的死区时间控制寄存器	GTDTCR	0000h	000C 2236h	16、32
	通用 PWM 定时器的死区时间值寄存器 U	GTDVU	FFFFh	000C 2238h	16、32
	通用 PWM 定时器的死区时间值寄存器 D	GTDVD	FFFFh	000C 223Ah	16、32
	通用 PWM 定时器的死区时间缓冲寄存器 U	GTDBU	FFFFh	000C 223Ch	16、32
	通用 PWM 定时器的死区时间缓冲寄存器 D	GTDBD	FFFFh	000C 223Eh	16、32
	通用 PWM 定时器的输出保护功能状态寄存器	GTSOS	0x00h	000C 2240h	16、32
	通用 PWM 定时器的输出保护暂时解除寄存器	GTSOTR	0000h	000C 2242h	16、32
	GPT3	通用 PWM 定时器的 I/O 控制寄存器	GTIOR	0000h	000C 2280h
通用 PWM 定时器的中断输出设定寄存器		GTINTAD	0000h	000C 2282h	8、16、32
通用 PWM 定时器的控制寄存器		GTCR	0000h	000C 2284h	8、16、32
通用 PWM 定时器的缓冲允许寄存器		GTBER	0000h	000C 2286h	8、16、32
通用 PWM 定时器的计数方向寄存器		GTUDC	0001h	000C 2288h	8、16、32
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	0000h	000C 228Ah	8、16、32
通用 PWM 定时器的状态寄存器		GTST	8000h	000C 228Ch	8、16、32
通用 PWM 定时器的计数器		GTCNT	0000h	000C 228Eh	16
通用 PWM 定时器的比较捕捉寄存器 A		GTCCRA	FFFFh	000C 2290h	16、32
通用 PWM 定时器的比较捕捉寄存器 B		GTCCRB	FFFFh	000C 2292h	16、32
通用 PWM 定时器的比较捕捉寄存器 C		GTCCRC	FFFFh	000C 2294h	16、32
通用 PWM 定时器的比较捕捉寄存器 D		GTCCRD	FFFFh	000C 2296h	16、32
通用 PWM 定时器的比较捕捉寄存器 E		GTCCRE	FFFFh	000C 2298h	16、32
通用 PWM 定时器的比较捕捉寄存器 F		GTCCRF	FFFFh	000C 229Ah	16、32



表 18.4 MTU 的寄存器一览表 (4/4)

通道	寄存器名	寄存器符号	复位后的值	地址	存取长度
GPT3	通用 PWM 定时器的周期设定寄存器	GTPR	FFFFh	000C 229Ch	16、32
	通用 PWM 定时器的周期设定缓冲寄存器	GTPBR	FFFFh	000C 229Eh	16、32
	通用 PWM 定时器的周期设定双缓冲寄存器	GTPDBR	FFFFh	000C 22A0h	16、32
	A/D 转换开始请求时序寄存器 A	GTADTRA	FFFFh	000C 22A4h	16、32
	A/D 转换开始请求时序缓冲寄存器 A	GTADTBRA	FFFFh	000C 22A6h	16、32
	A/D 转换开始请求时序双缓冲寄存器 A	GTADTDBRA	FFFFh	000C 22A8h	16、32
	A/D 转换开始请求时序寄存器 B	GTADTRB	FFFFh	000C 22AC h	16、32
	A/D 转换开始请求时序缓冲寄存器 B	GTADTBRB	FFFFh	000C 22AEh	16、32
	A/D 转换开始请求时序双缓冲寄存器 B	GTADTDBRB	FFFFh	000C 22B0h	16、32
	通用 PWM 定时器的输出无效控制寄存器	GTONCR	0100h	000C 22B4h	16、32
	通用 PWM 定时器的死区时间控制寄存器	GTDTCR	0000h	000C 22B6h	16、32
	通用 PWM 定时器的死区时间值寄存器 U	GTDVU	FFFFh	000C 22B8h	16、32
	通用 PWM 定时器的死区时间值寄存器 D	GTDVD	FFFFh	000C 22BAh	16、32
	通用 PWM 定时器的死区时间缓冲寄存器 U	GTDBU	FFFFh	000C 22BCh	16、32
	通用 PWM 定时器的死区时间缓冲寄存器 D	GTDBD	FFFFh	000C 22BEh	16、32
	通用 PWM 定时器的输出保护功能状态寄存器	GTSOS	0x00h	000C 22C0h	16、32
	通用 PWM 定时器的输出保护暂时解除寄存器	GTSOTR	0000h	000C 22C2h	16、32

## 18.2.1 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
复位后的值	—	—	—	—	—	—	—	—	—	—	—	—	CST3	CST2	CST1	CST0
	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。

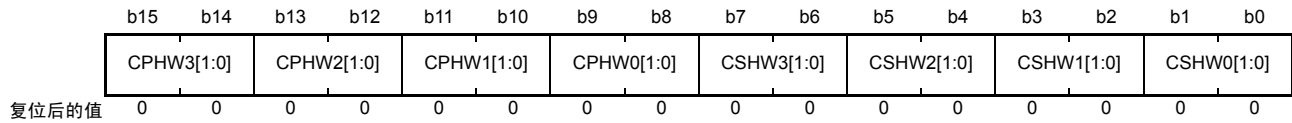
## CSTn 位 (GPTn.GTCNT 计数开始位) (n=0 ~ 3)

此位选择 GPTn.GTCNT 计数器的计数 / 停止计数。

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”。

## 18.2.2 通用 PWM 定时器的硬件源启动控制寄存器 (GTHSCR)

地址 000C 2004h



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。

## 18.2.3 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.CTSn=1) 还是停止计数 (GTSTR.CTSn=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。

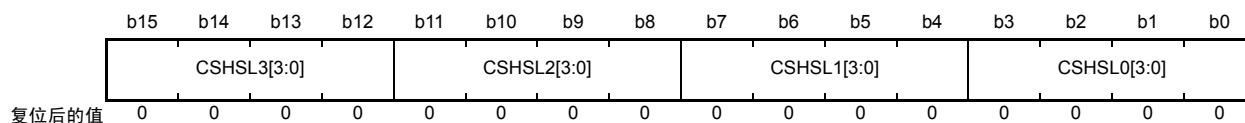
当 CCHWn[1:0] 位为“01b”、“10b”、“11b”时, 重复接受硬件源。

## 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”。

### 18.2.4 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 0 0: AN100 的比较器检测 0 1 0 1: AN101 的比较器检测 0 1 1 0: AN102 的比较器检测 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 0 0: AN100 的比较器检测 0 1 0 1: AN101 的比较器检测 0 1 1 0: AN102 的比较器检测 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
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 0 0: AN100 的比较器检测 0 1 0 1: AN101 的比较器检测 0 1 1 0: AN102 的比较器检测 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
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 0 0: AN100 的比较器检测 0 1 0 1: AN101 的比较器检测 0 1 1 0: AN102 的比较器检测 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 80 引脚版和 64 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。

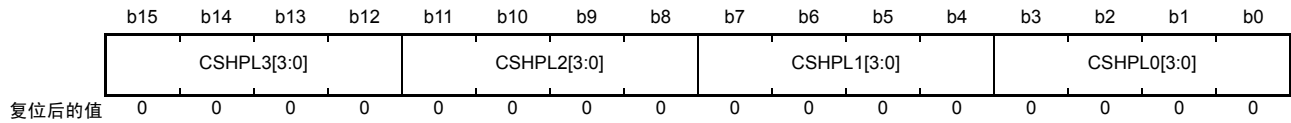
CSHSLn[3:0] 位 (GPTn.GTCNT 硬件计数器启动源选择位) (n=0 ~ 3)

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”。

## 18.2.5 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 0 0: AN100 的比较器检测 0 1 0 1: AN101 的比较器检测 0 1 1 0: AN102 的比较器检测 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 0 0: AN100 的比较器检测 0 1 0 1: AN101 的比较器检测 0 1 1 0: AN102 的比较器检测 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
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 0 0: AN100 的比较器检测 0 1 0 1: AN101 的比较器检测 0 1 1 0: AN102 的比较器检测 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
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 0 0: AN100 的比较器检测 0 1 0 1: AN101 的比较器检测 0 1 1 0: AN102 的比较器检测 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。

## 18.2.6 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 寄存器的写允许位	0: 允许写寄存器 1: 禁止写寄存器	R/W
b1	WP1	GPT1 寄存器的写允许位		R/W
b2	WP2	GPT2 寄存器的写允许位		R/W
b3	WP3	GPT3 寄存器的写允许位		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 寄存器的写允许位) (n=0 ~ 3)

这些位选择允许或者禁止写 GPTn 寄存器。



## 18.2.7 通用 PWM 定时器的同步寄存器 (GTSYNC)

地址 000C 200Eh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	SYNC3[1:0]	—	—	—	SYNC2[1:0]	—	—	—	—	SYNC1[1:0]	—	—	—	SYNC0[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	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)。

## 18.2.8 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

## 18.2.9 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。当 GTIOChA 引脚或者 GTIOChB 引脚用作输入捕捉引脚 (GPTn.GTIOR.GTIOA[5]=1 或者 GPTn.GTIOR.GTIOB[5]=1) 时，必须将此位置“0”。

## 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。

## 18.2.10 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.CTSn 位的写禁止位) (n=0 ~ 3)

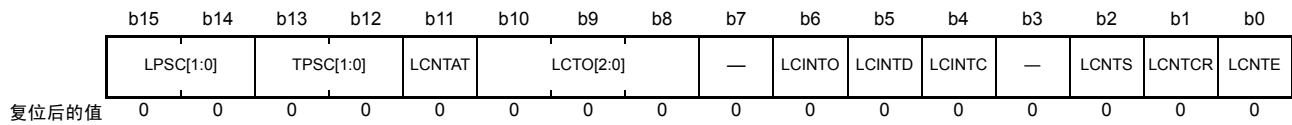
此位允许或者禁止写 GTSTR.CSTn 位。

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, 因硬件源开始或者停止计数器计数的状态也被写到 GTSTR.CSTn 位。

## 18.2.11 LOCO 计数控制寄存器 (LCCR)

地址 000C 2080h



位	符号	位名	功能	R/W
b0	LCNTE	LOCO 计数功能允许位	0: LOCO 计数功能停止 1: LOCO 计数功能运行	R/W
b1	LCNTCR	LOCO 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	LOCO 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	LOCO 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	LOCO 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]	LOCO 分频时钟上升沿中断减少次数设定位	b10 b8 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	LOCO 计数结果减少设定位	0: 不减少 1: 减少	R/W
b13-b12	TPSC[1:0]	LOCO 计数时钟选择位	b13 b12 0 0: ICLK (系统时钟) 0 1: ICLK/2 (系统时钟 /2) 1 0: ICLK/4 (系统时钟 /4) 1 1: ICLK/8 (系统时钟 /8)	R/W
b15-b14	LPSC[1:0]	LOCO 分频时钟选择位	b15 b14 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 专用低速内部振荡器 (LOCO) 的计数功能的寄存器。使用 LOCO 的计数功能时, 必须让独立看门狗定时器 (IWDT) 也运行。

**LCNTE 位 (LOCO 计数功能允许位)**

此位设定 LOCO 的计数功能停止或者运行。

**LCNTCR 位 (LOCO 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 (LOCO 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 (LOCO 分频时钟上升沿中断允许位)**

此位允许或者禁止由 LOCO 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 (LOCO 计数值偏差超中断允许位)**

此位允许或者禁止由 LOCO 计数值的偏差超产生的中断。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 (LOCO 分频时钟上升沿中断减少次数设定位)**

这些位设定 LOCO 分频时钟上升沿中断的减少次数。

**LCNTAT 位 (LOCO 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 LOCO 计数结果传送到 LCNTn (n=00 ~ 15) 的时序。

**TPSC[1:0] 位 (LOCO 计数时钟选择位)**

这些位选择对 LOCO 分频时钟计数的时钟 (LCNT 运行时钟)。

**LPSC[1:0] 位 (LOCO 分频时钟选择位)**

这些位选择 LOCO 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 IWDI.IWDTCR.CKS[3:0] 位置“00xx”以外的值。

## 18.2.12 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	LOCO 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	LOCO 计数值偏差超中断请求标志	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 LOCO 分频时钟的计数状态的寄存器。

## LISC 标志 (LOCO 分频时钟上升沿中断请求标志)

这是由 LOCO 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 LOCO 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 LOCO 分频时钟上升时

[ 为“0”的条件 ]

- 给 LISC 标志写“0”时

## LISD 标志 (LOCO 计数值偏差超中断请求标志)

这是由 LOCO 计数值的偏差超产生中断请求的标志。在标志置“0”前不产生由 LOCO 计数值的偏差超引起的下一个中断请求。

[ 为“1”的条件 ]

- 在 LCCR.LCINTD=1 的状态下 LOCO 计数值器发生偏差超时

[ 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

### 18.2.13 LOCO 计数值寄存器 (LCNT)

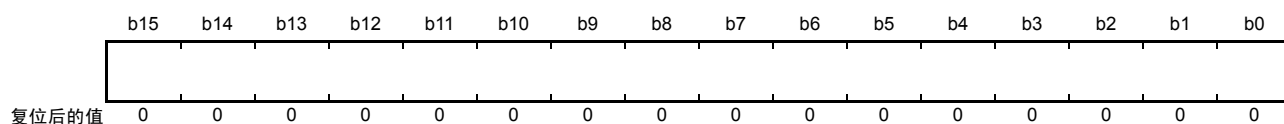
地址 000C 2084h



LCNT 寄存器是对 LOCO 分频时钟进行计数的只读寄存器。

### 18.2.14 LOCO 计数结果平均寄存器 (LCNTA)

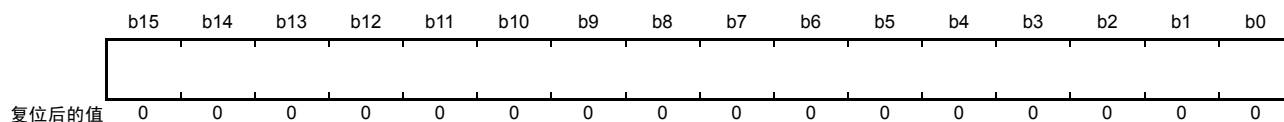
地址 000C 2086h



LCNTA 寄存器是表示 LOCO 分频时钟计数结果 (LCNT00 ~ LCNT15) 的平均值的只读寄存器。

### 18.2.15 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



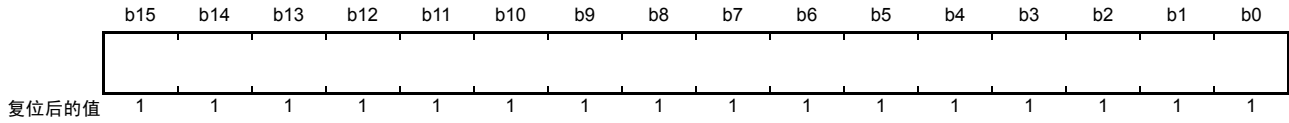
LCNTn 寄存器是表示 LOCO 分频时钟计数结果的寄存器。只能在停止计数 (LCCR.LCNTE 位 =0) 时进行写操作。

LCNT00 寄存器的计数值是最新结果。

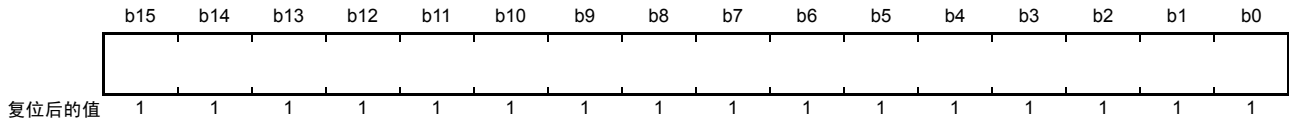


### 18.2.16 LOCO 计数上限 / 下限容许偏差值寄存器 (LCNTDU、LCNTDL)

地址 LCNTDU 000C 20A8h



地址 LCNTDL 000C 20AAh

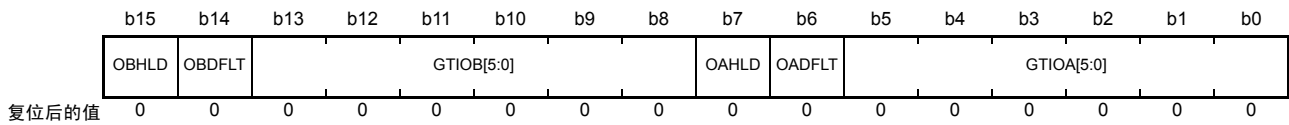


LCNTDU 寄存器和 LCNTDL 寄存器是设定 LOCO 分频时钟计数值的容许偏差的寄存器。

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 LOCO 计数值偏差超中断请求。

### 18.2.17 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 18.5。	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	OAHLD	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 18.5。	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 18.5。

**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 18.5。

**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 18.5 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 18.5 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。

## 18.2.18 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。

## 18.2.19 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: ICLK (系统时钟) 0 1: ICLK/2 (系统时钟 / 2) 1 0: ICLK/4 (系统时钟 / 4) 1 1: ICLK/8 (系统时钟 / 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.CTSn=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。

## 18.2.20 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]
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	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。

## 18.2.21 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”。

## 18.2.22 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、三角波时的波峰 / 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<sub>n</sub> 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<sub>n</sub> 中断减少功能联动。

## 18.2.23 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 ]

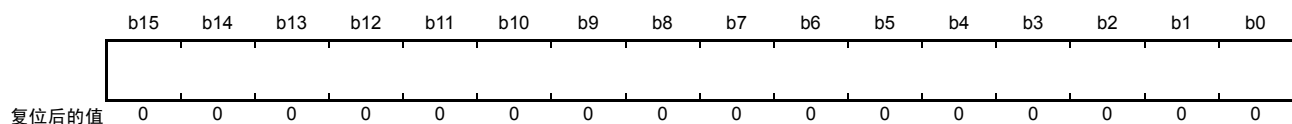
- 自动附加死区时间后的定时器输出交替点在定时器周期内时

**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。

**18.2.24 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

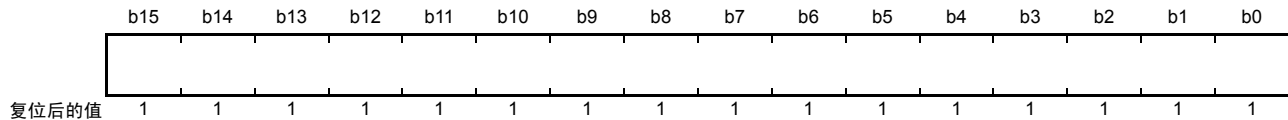


GTCNT 计数器是 16 位可读写计数器。各通道有 1 个 (共计 4 个) GTCNT 计数器。不能在计数时进行写操作, 只能在停止计数时进行写操作。禁止以 8 位为单位存取 GTCNT 计数器, 必须以 16 位为单位进行存取。



### 18.2.25 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

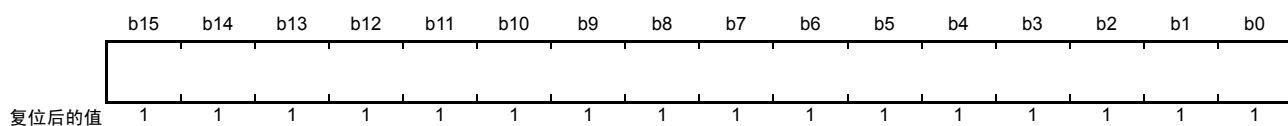


GTCCRm 寄存器是 16 位可读写寄存器。各通道有 6 个（共计 24 个）GTCCRm 寄存器。

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）。

### 18.2.26 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  $\times 2$ 。

### 18.2.27 通用 PWM 定时器的周期设定缓冲寄存器 (GTPBR)

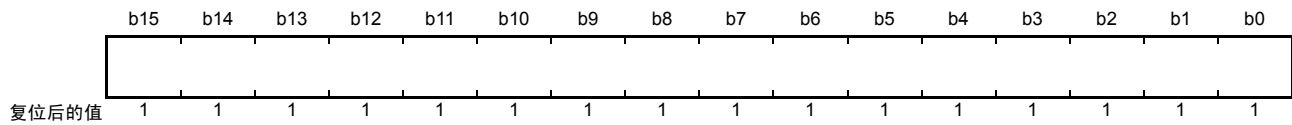
地址 GPT0.GTPBR 000C 211Eh、GPT1.GTPBR 000C 219Eh、GPT2.GTPBR 000C 221Eh、GPT3.GTPBR 000C 229Eh



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。

### 18.2.28 通用 PWM 定时器的周期设定双缓冲寄存器 (GTPDBR)

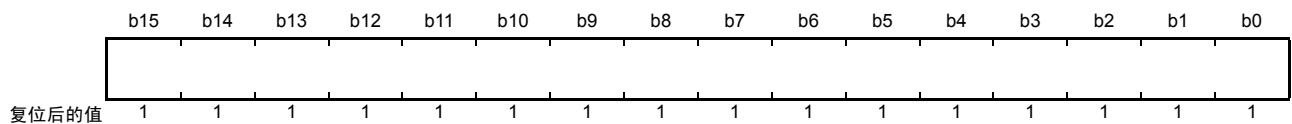
地址 GPT0.GTPDBR 000C 2120h、GPT1.GTPDBR 000C 21A0h、GPT2.GTPDBR 000C 2220h、GPT3.GTPDBR 000C 22A0h



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。

### 18.2.29 A/D 转换开始请求时序寄存器 m (GTADTRm) (m=A、B)

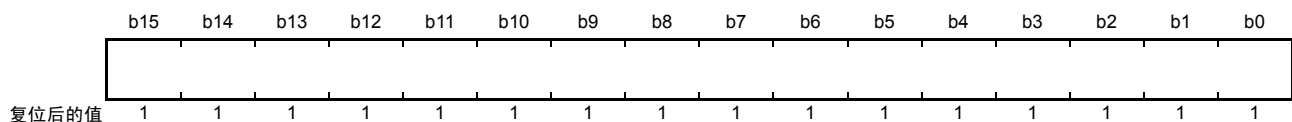
地址 GPT0.GTADTRA 000C 2124h、GPT1.GTADTRA 000C 21A4h、GPT2.GTADTRA 000C 2224h、GPT3.GTADTRA 000C 22A4h  
GPT0.GTADTRB 000C 212Ch、GPT1.GTADTRB 000C 21ACh、GPT2.GTADTRB 000C 222Ch、GPT3.GTADTRB 000C 22ACh



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。

### 18.2.30 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。

### 18.2.31 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。

## 18.2.32 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	1	0	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: AN100 的比较器检测 0 1 0 1: AN101 的比较器检测 0 1 1 0: AN102 的比较器检测 0 1 1 1: GTETRG 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。

## 18.2.33 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)**

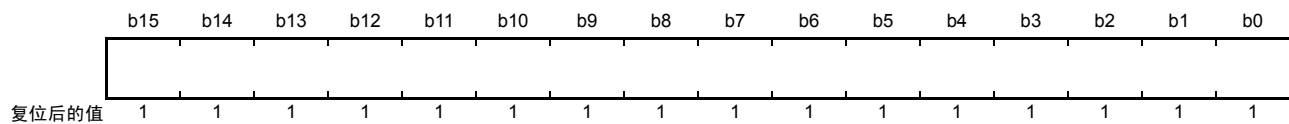
此位允许 GTDVD 寄存器和 GTDBD 寄存器组合的缓冲运行。  
缓冲传送时序在三角波时为波谷，在锯齿波时为上溢或者下溢。  
同时将 TDFER 位置“1”时，优先 TDFER 位的设定。

**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。

**18.2.34 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



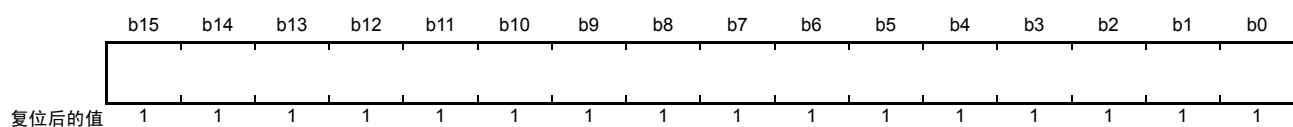
GTDVm 寄存器是 16 位可读写寄存器。用于设定生成带死区时间的 PWM 波形的死区时间。各通道分别有 2 个用于递增计数的 GTDVU 寄存器和用于递减计数的 GTDVD 寄存器，共计 8 个 GTDm 寄存器。

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 GTDm 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 GTDm 寄存器，必须以 16 位为单位进行存取。

**18.2.35 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 GTDm 寄存器的缓冲寄存器。各通道有 2 个 (共计 8 个) GTDBm 寄存器。

## 18.2.36 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“18.7.4 GTIOC 引脚输出的输出保护功能”。

## 18.2.37 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。

## 18.3 运行说明

### 18.3.1 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 GTIOCN A 输出引脚和 GTIOCN B 输出引脚 (n=0 ~ 3)。另外，GTIOCN A 引脚和 GTIOCN B 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。

#### 18.3.1.1 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 18.2 所示。

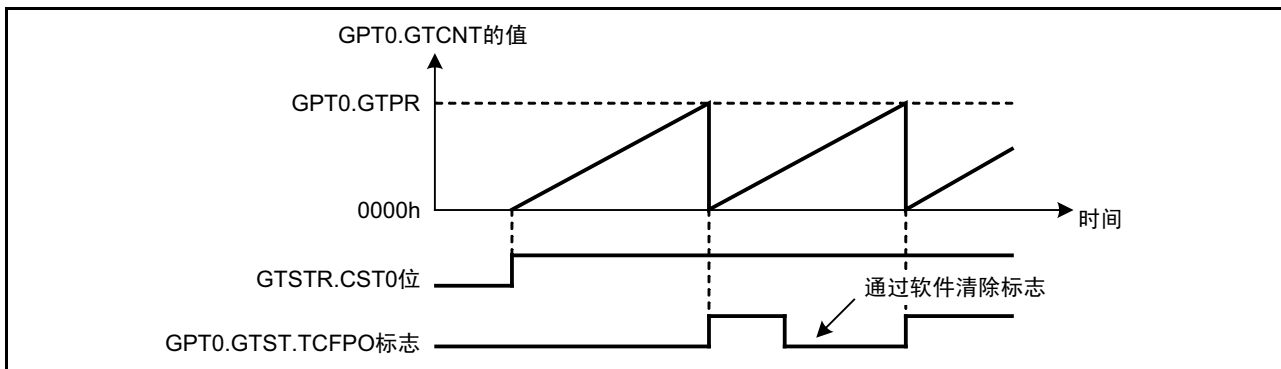


图 18.2 周期计数的运行例子 (递增计数时)

递增计数时的周期计数运行设定例子如图 18.3 所示。



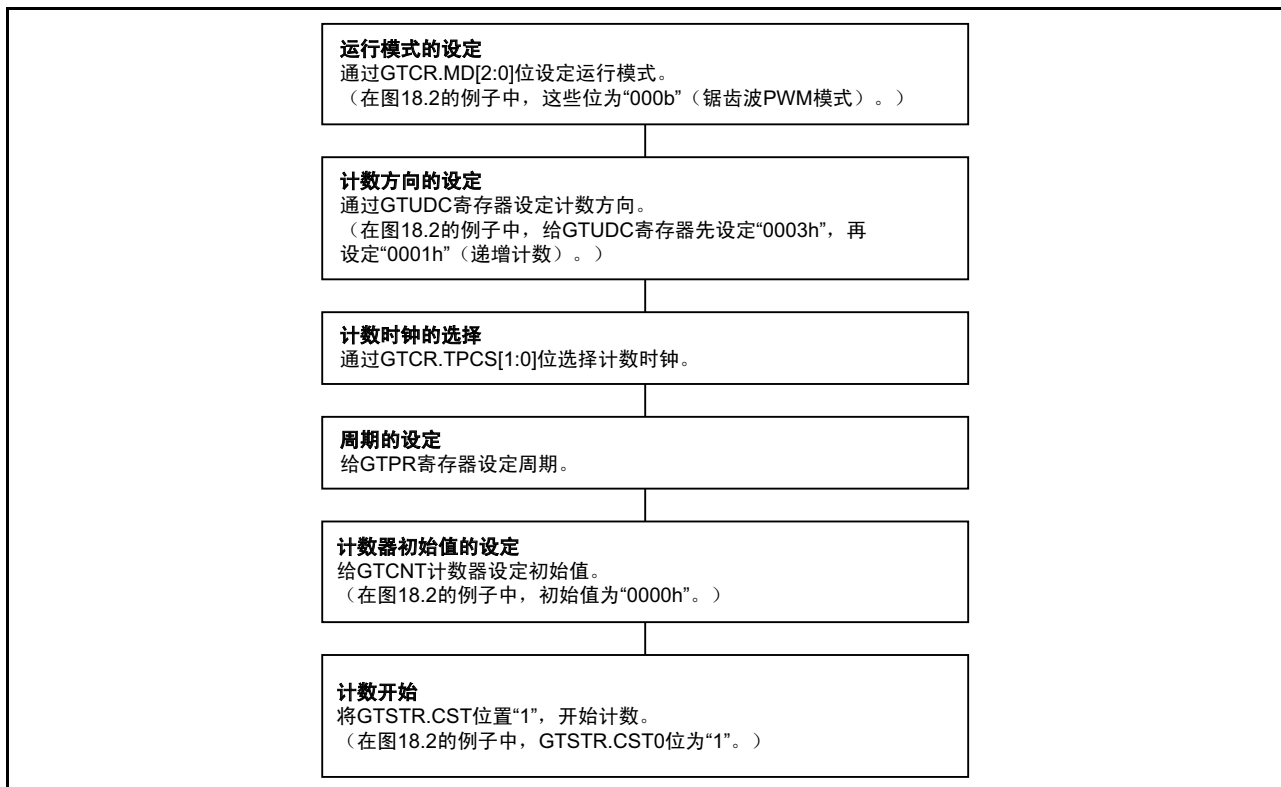


图 18.3 周期计数运行的设定例子 (递增计数时)

## (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 18.4 所示。

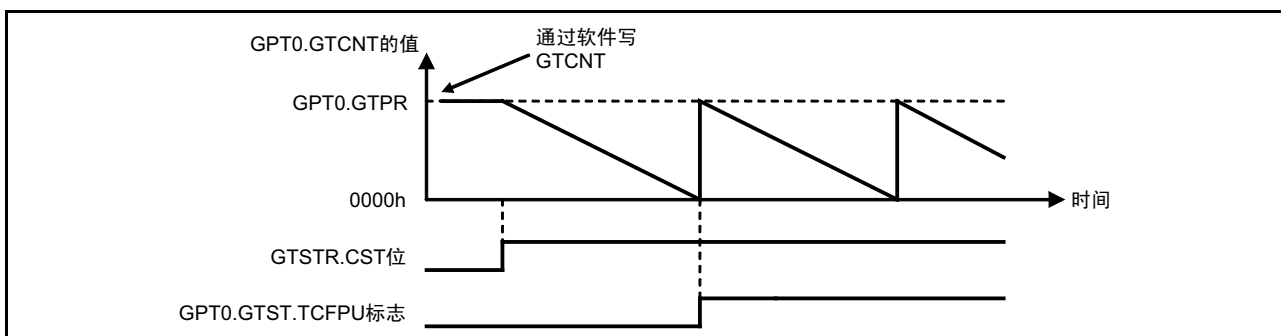


图 18.4 周期计数的运行例子 (递减计数时)

递减计数时的周期计数运行设定例子如图 18.5 所示。

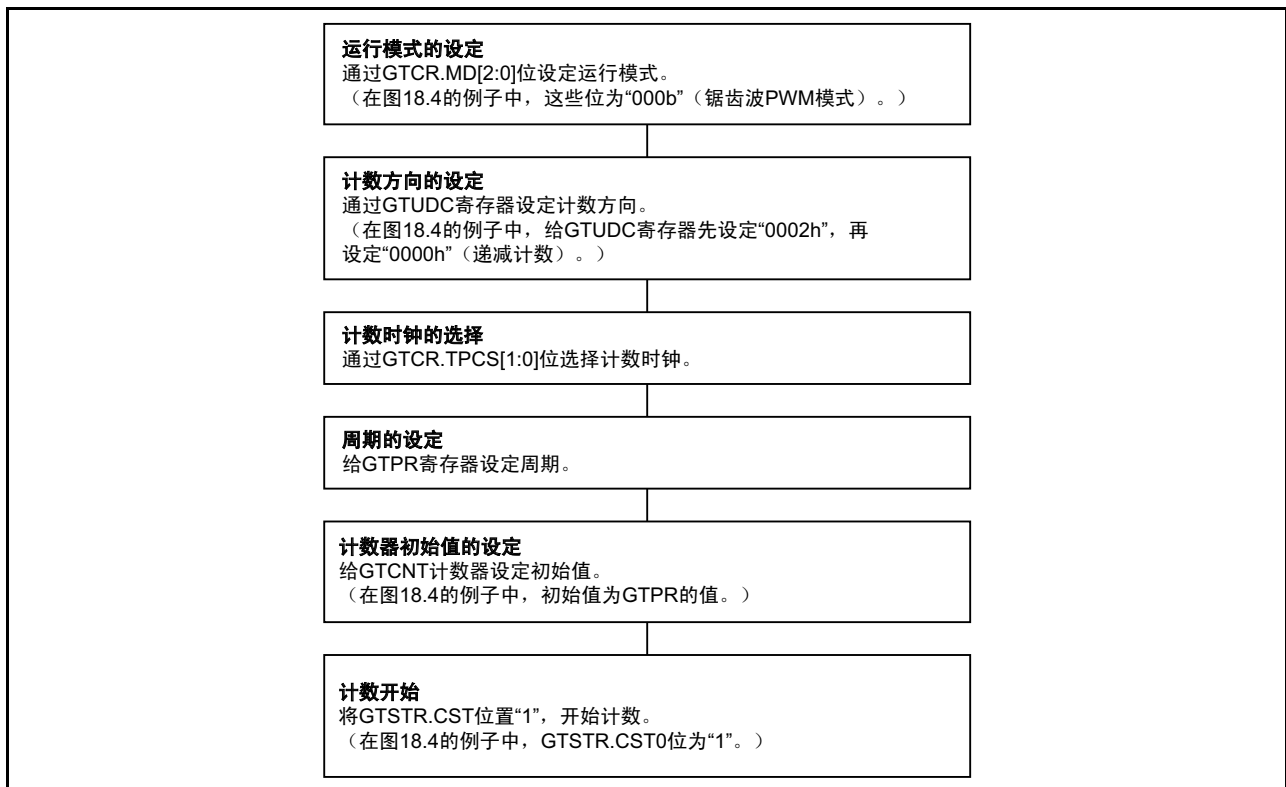


图 18.5 周期计数运行的设定例子 (递减计数时)

### 18.3.1.2 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 18.6 所示。

在此例子中假设 GPT0 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。

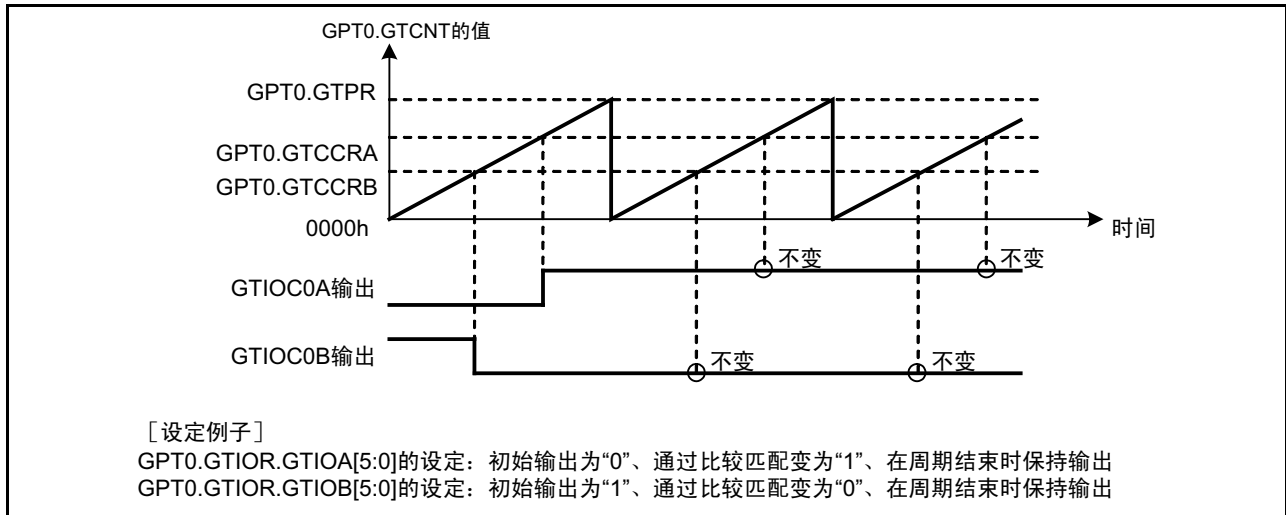


图 18.6 Low 电平输出 /High 电平输出的运行例子

Low 电平输出 /High 电平输出运行的设定例子如图 18.7 所示。



图 18.7 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 18.8、图 18.9 所示。

在图 18.8 的例子中假设 GPT0 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 18.9 的例子中假设 GPT0 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。

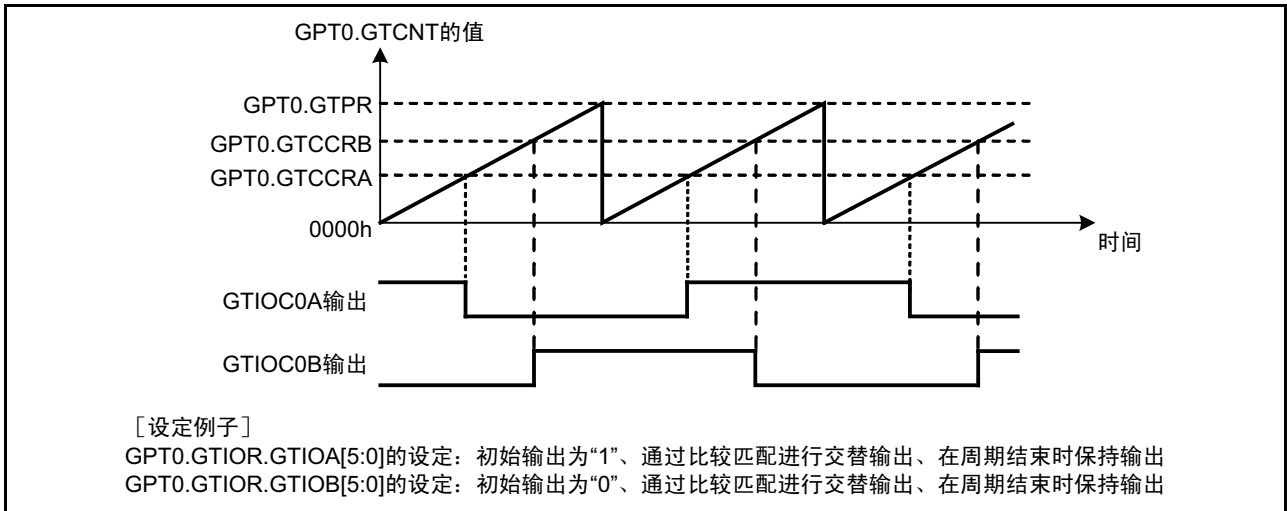


图 18.8 交替输出的运行例子 (1)

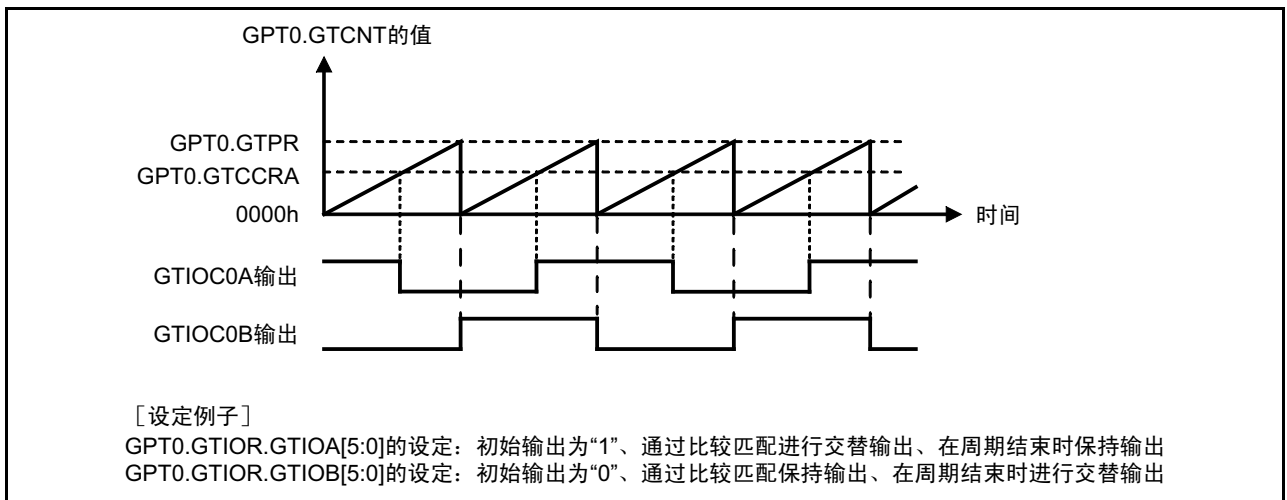


图 18.9 交替输出的运行例子 (2)

交替输出运行的设定例子如图 18.10 所示



图 18.10 交替输出运行的设定例子

18.3.1.3 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 18.11 所示。

在此例子中假设 GPT0 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。

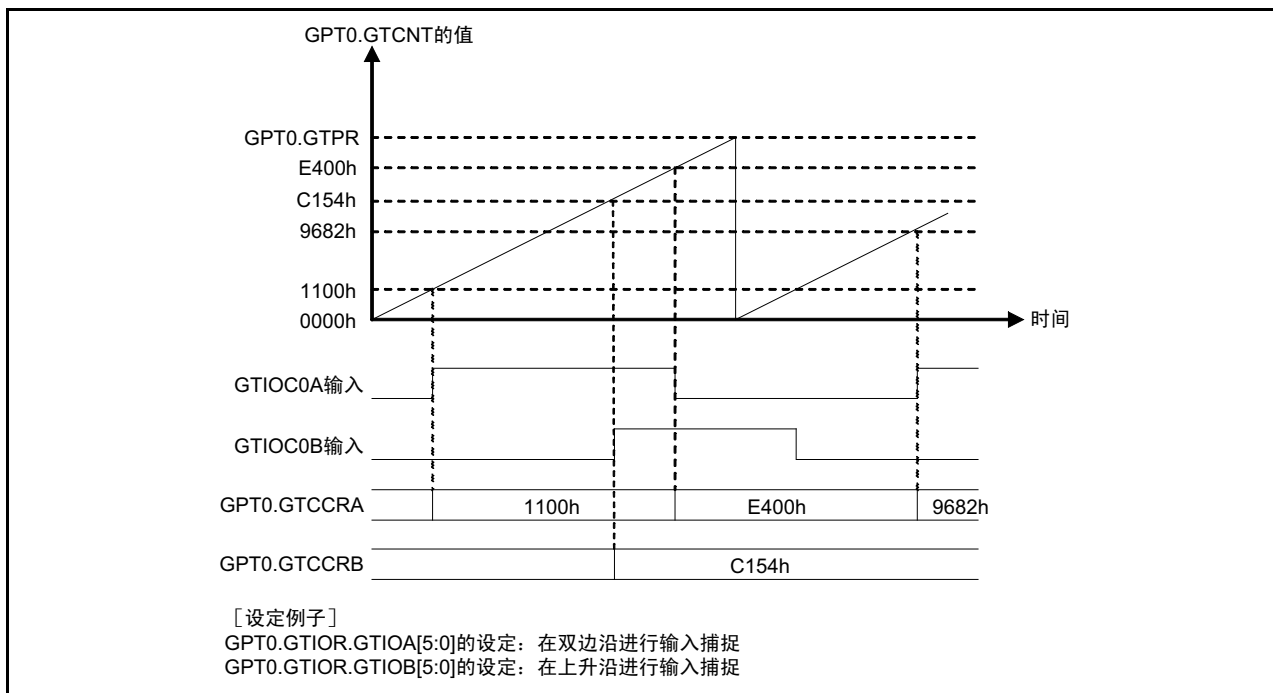


图 18.11 输入捕捉的运行例子

输入捕捉运行的设定例子如图 18.12 所示。

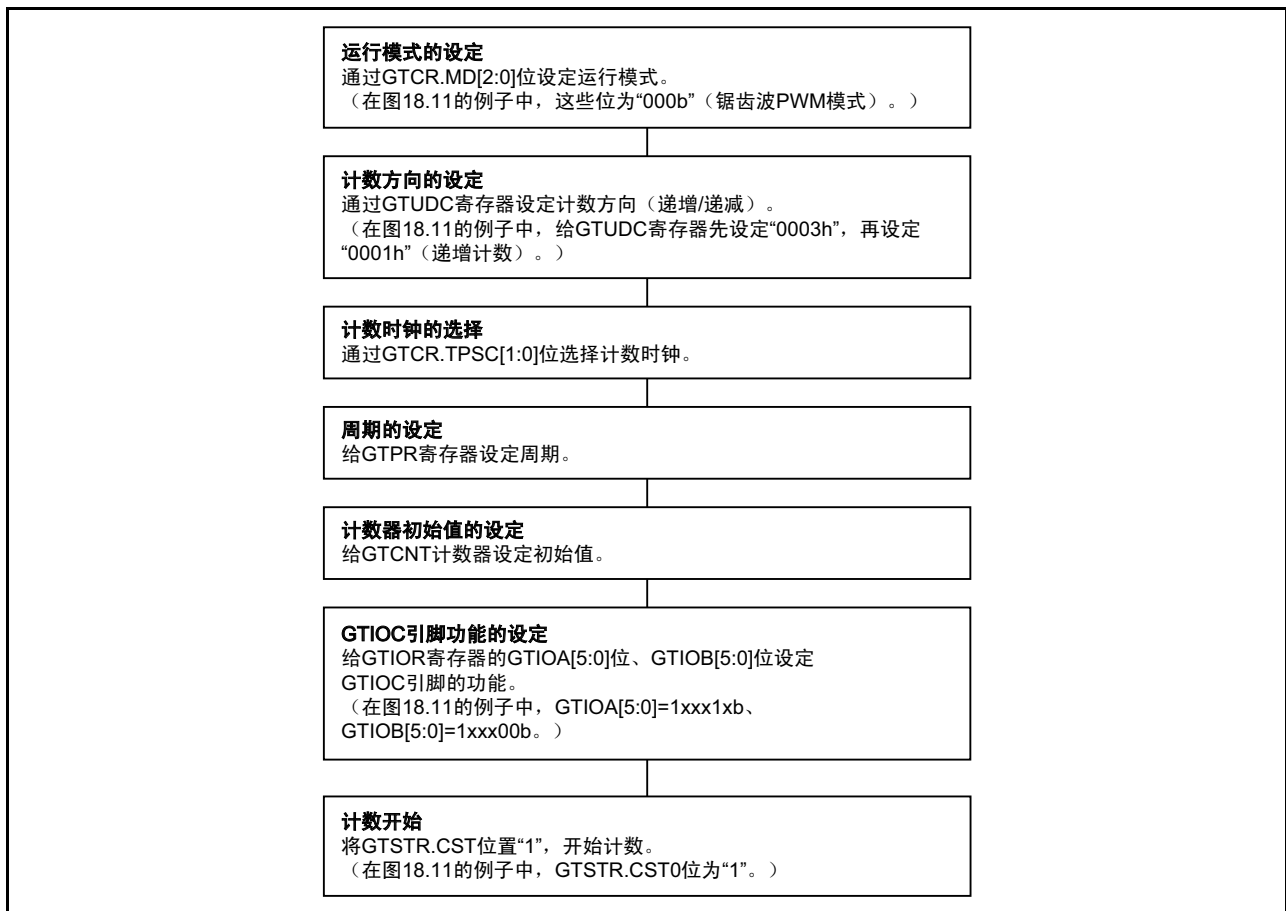


图 18.12 输入捕捉运行的设定例子



### 18.3.2 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

#### 18.3.2.1 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 18.13 ~ 图 18.15，GTPR 寄存器的缓冲运行设定例子如图 18.16 所示。

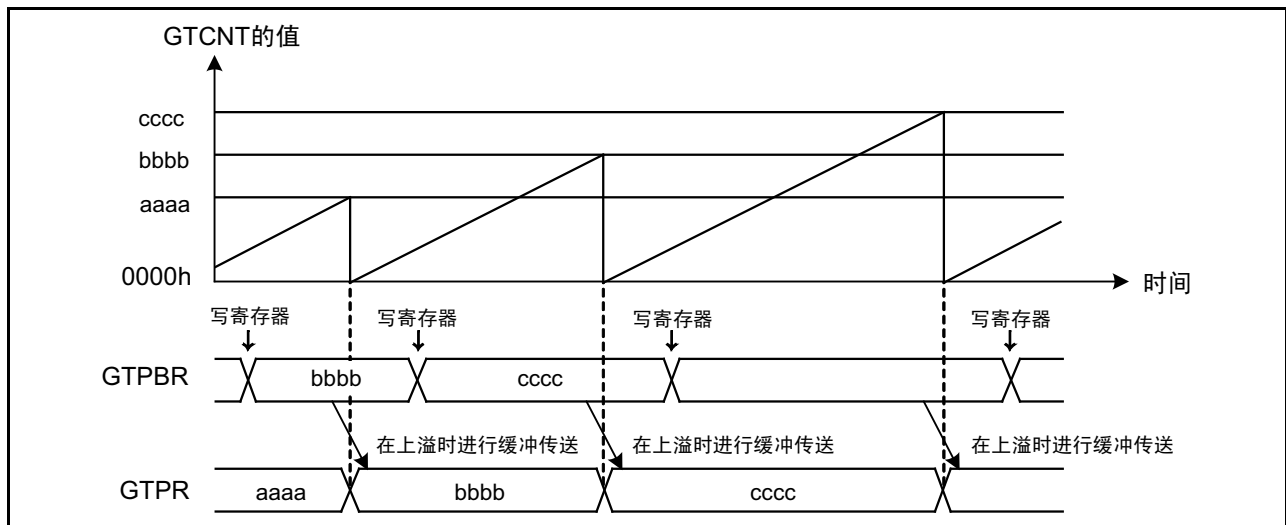


图 18.13 GTPR 寄存器的缓冲运行例子（在锯齿波进行递增计数的情况）

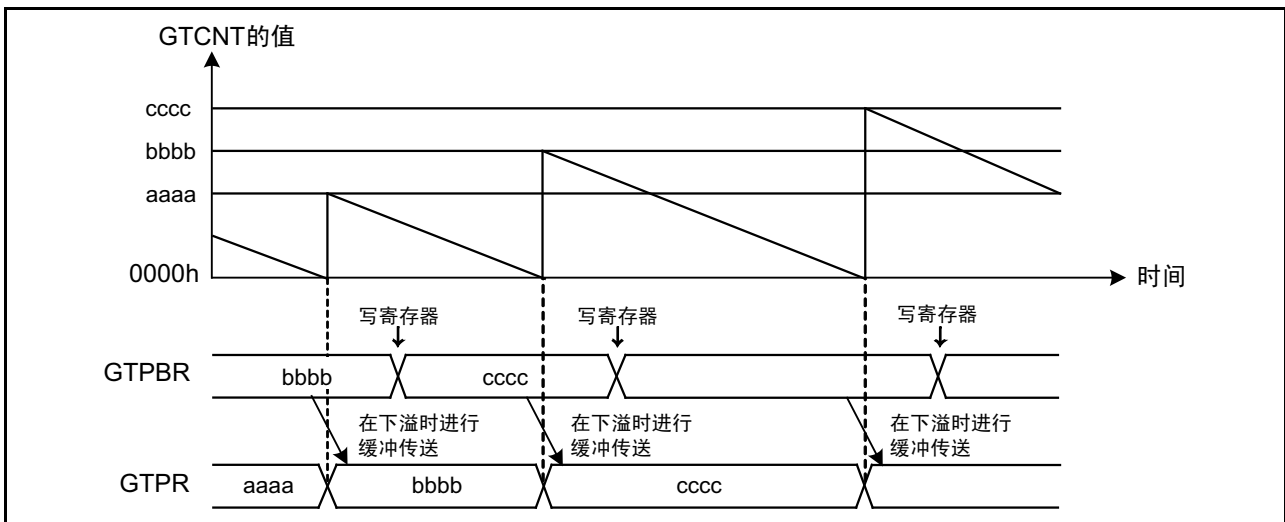


图 18.14 GTPR 寄存器的缓冲运行例子 (在锯齿波进行递减计数的情况)

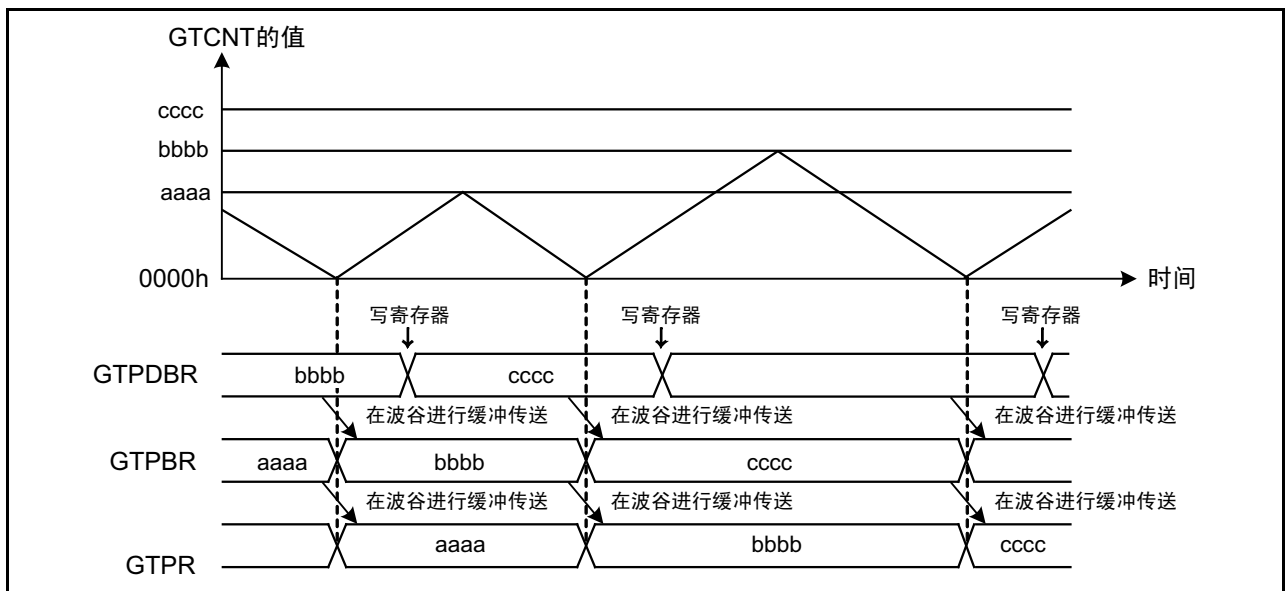


图 18.15 GTPR 寄存器的缓冲运行例子 (三角波的情况)

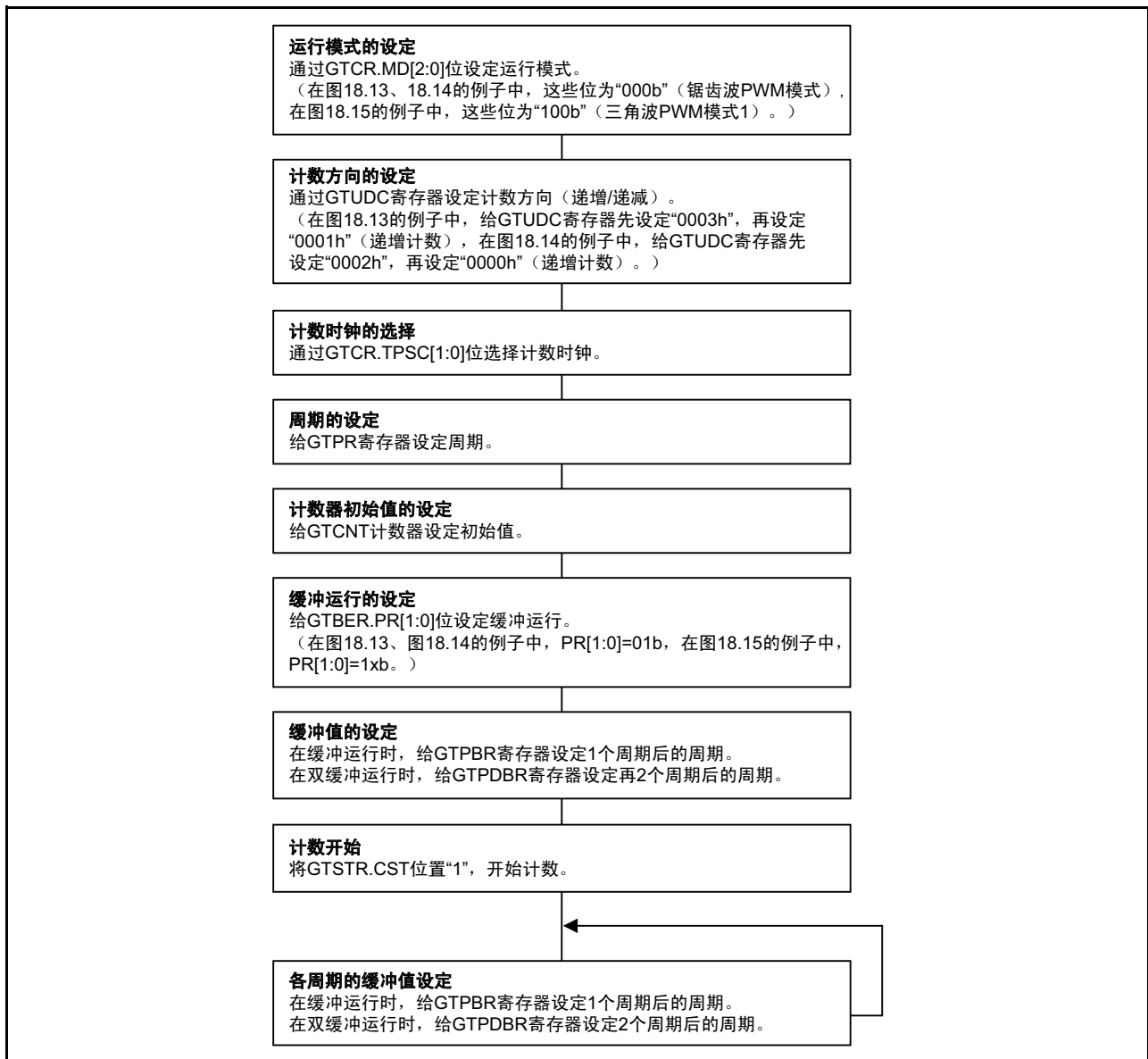


图 18.16 GTPR 寄存器的缓冲运行设定例子

## 18.3.2.2 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 18.17 ~ 图 18.19，GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行设定例子如图 18.20 所示。

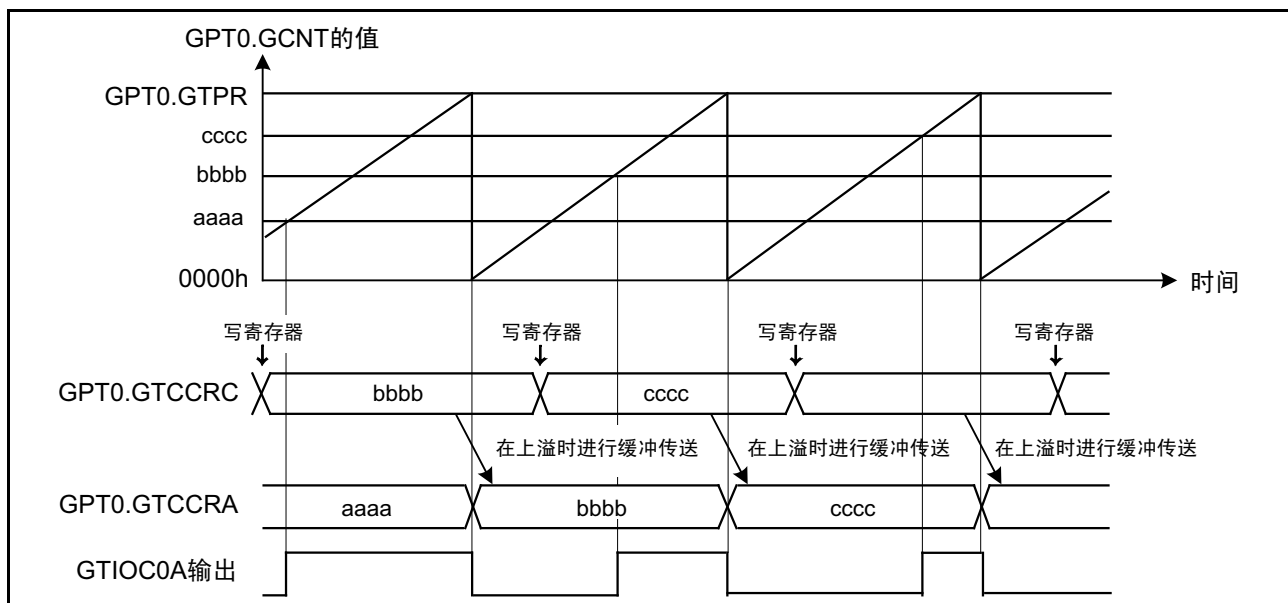


图 18.17 GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行例子  
(输出比较、在锯齿波进行递增计数、在 GTCCRA 比较匹配时输出 High 电平、在周期结束时输出 Low 电平的情况)

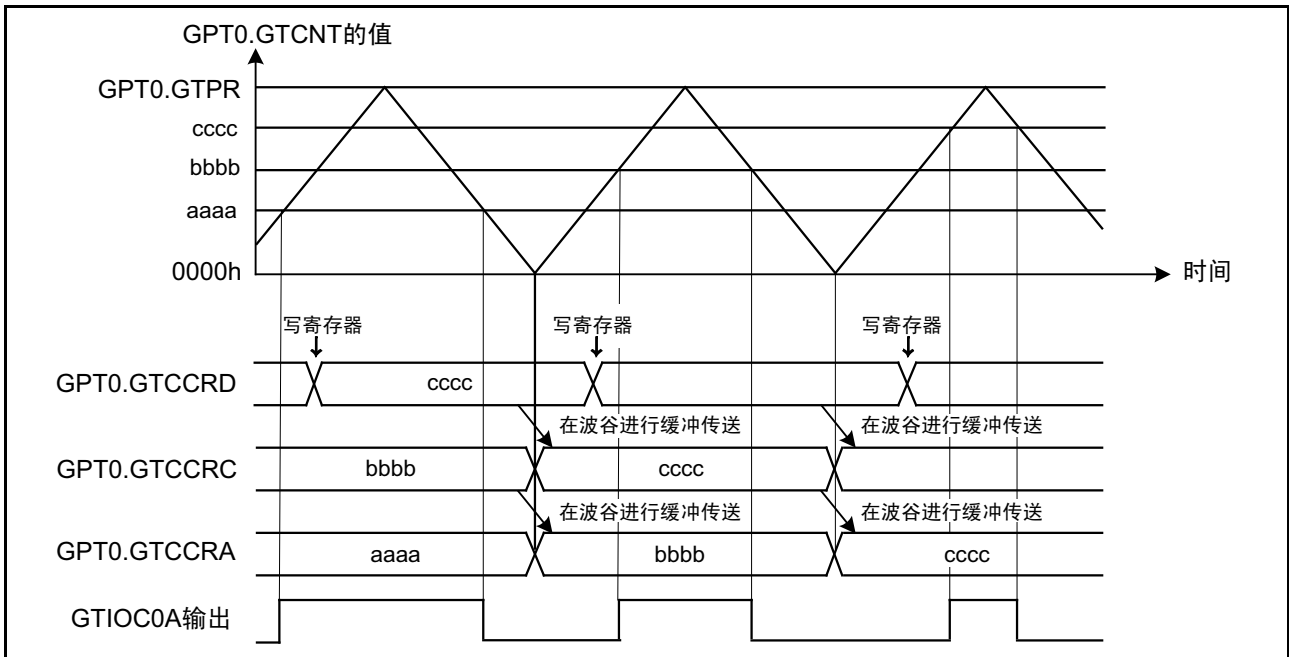


图 18.18 GTCCRA 寄存器和 GTCCRB 寄存器的双缓冲运行例子  
 (输出比较、三角波、在波谷进行缓冲传送、在 GTCCRA 比较匹配时进行交替输出、在周期结束时保持输出的情况)

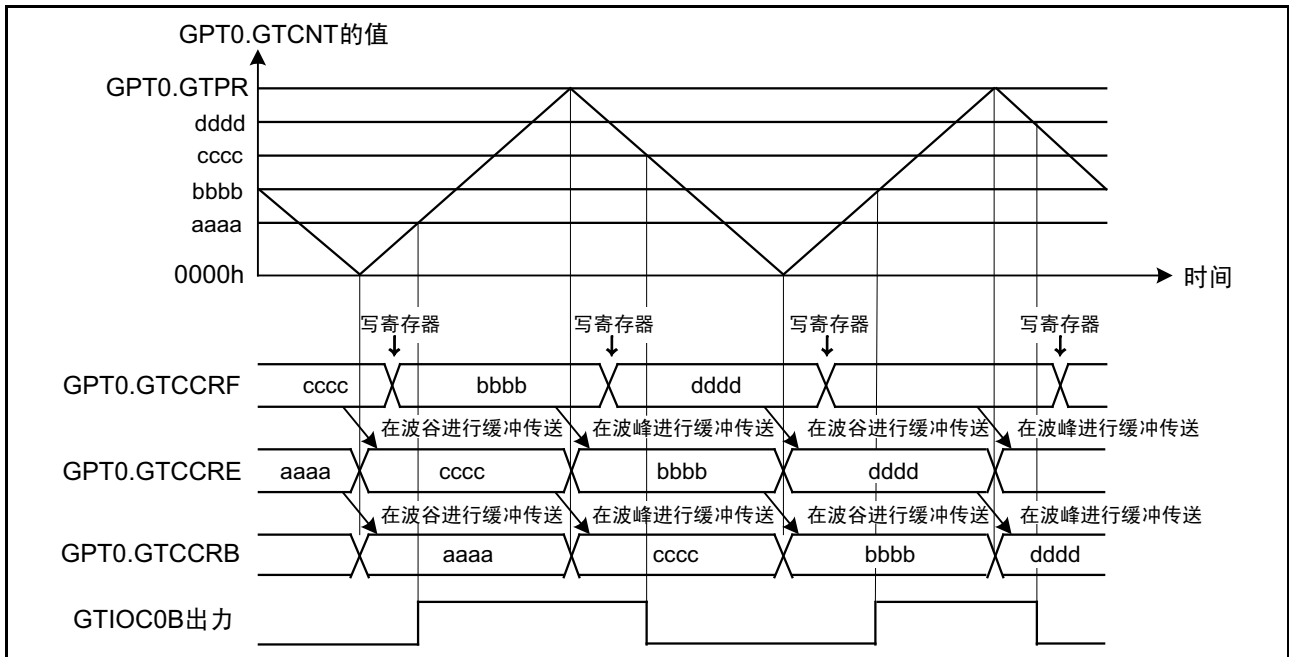


图 18.19 GTCCRA 寄存器和 GTCCRB 寄存器的双缓冲运行例子  
 (输出比较、三角波、在波峰和波谷都进行缓冲传送、在 GTCCRB 比较匹配时进行交替输出、在周期结束时保持输出的情况)

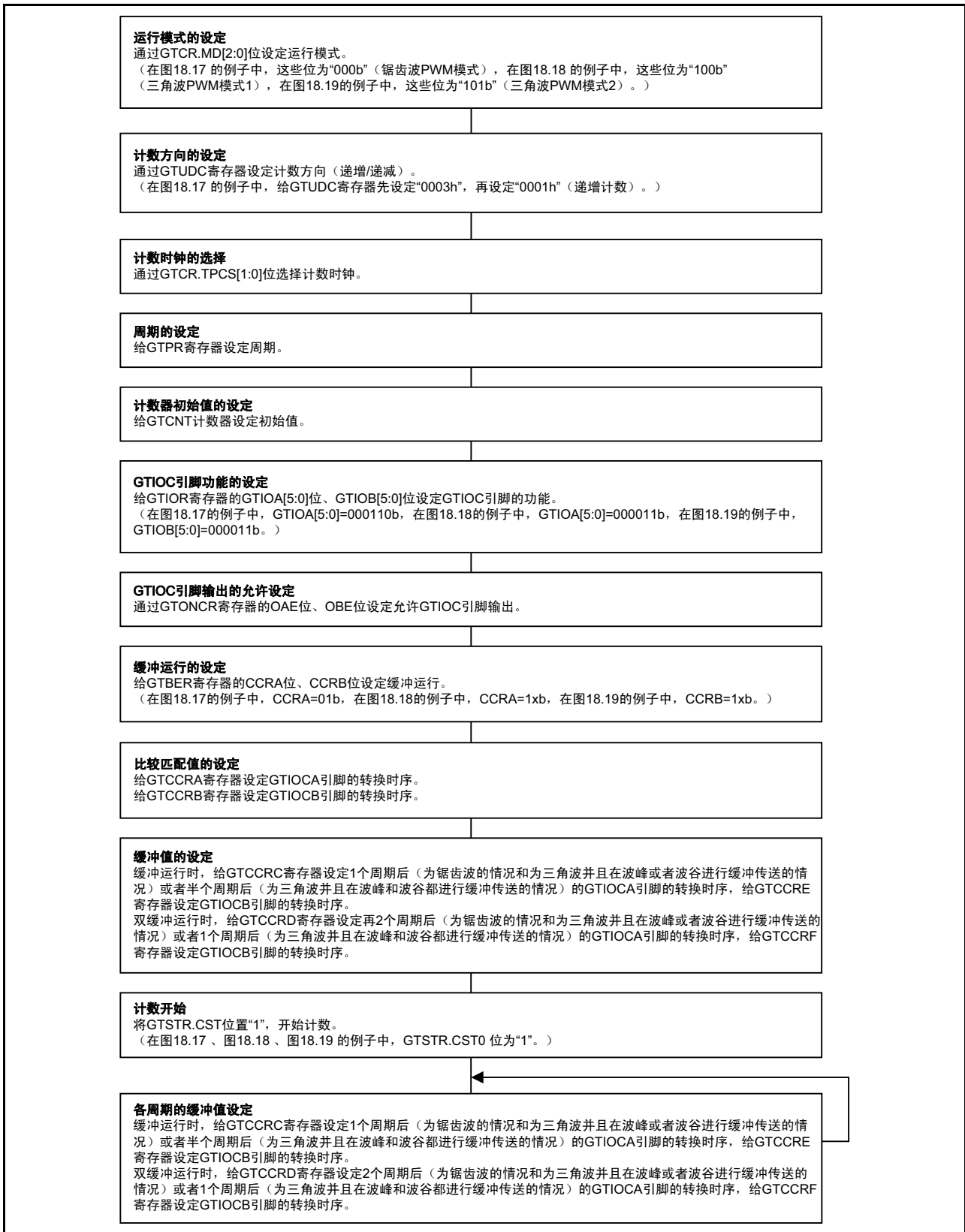


图 18.20 GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行设定例子 (输出比较的情况)

(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 18.21、图 18.22，GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行设定例子如图 18.23 所示。

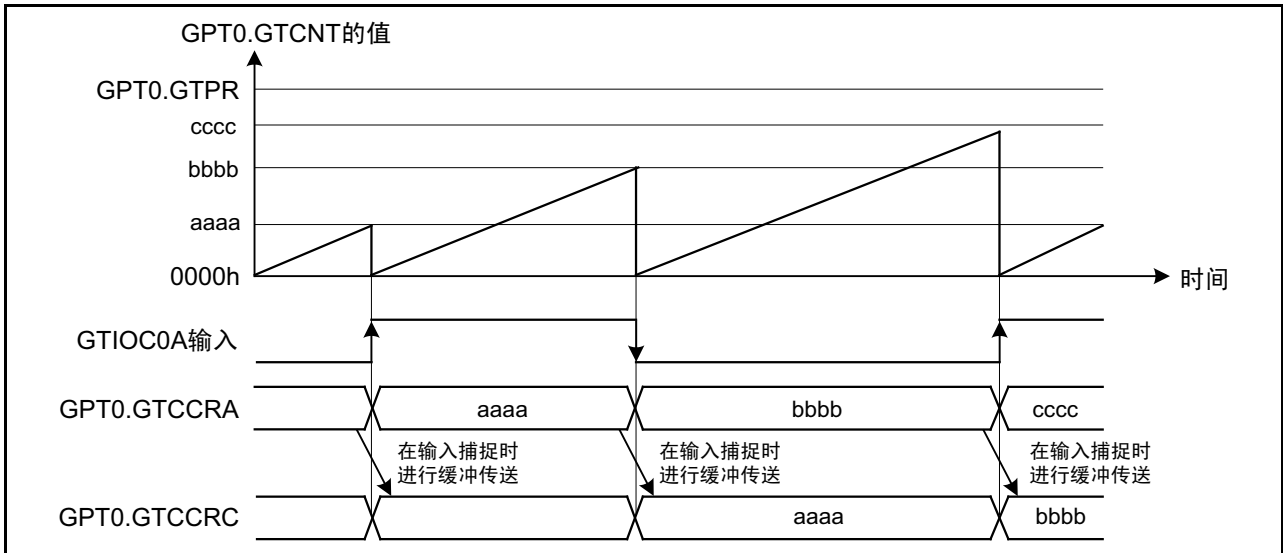


图 18.21 GTCCRA 寄存器和 GTCCRB 寄存器的缓冲运行例子  
(在锯齿波进行递增计数、在 GTIOC0A 输入的双边沿进行输入捕捉、在 GTCCRA 寄存器的输入捕捉时进行 GTCNT 计数器清除的情况)

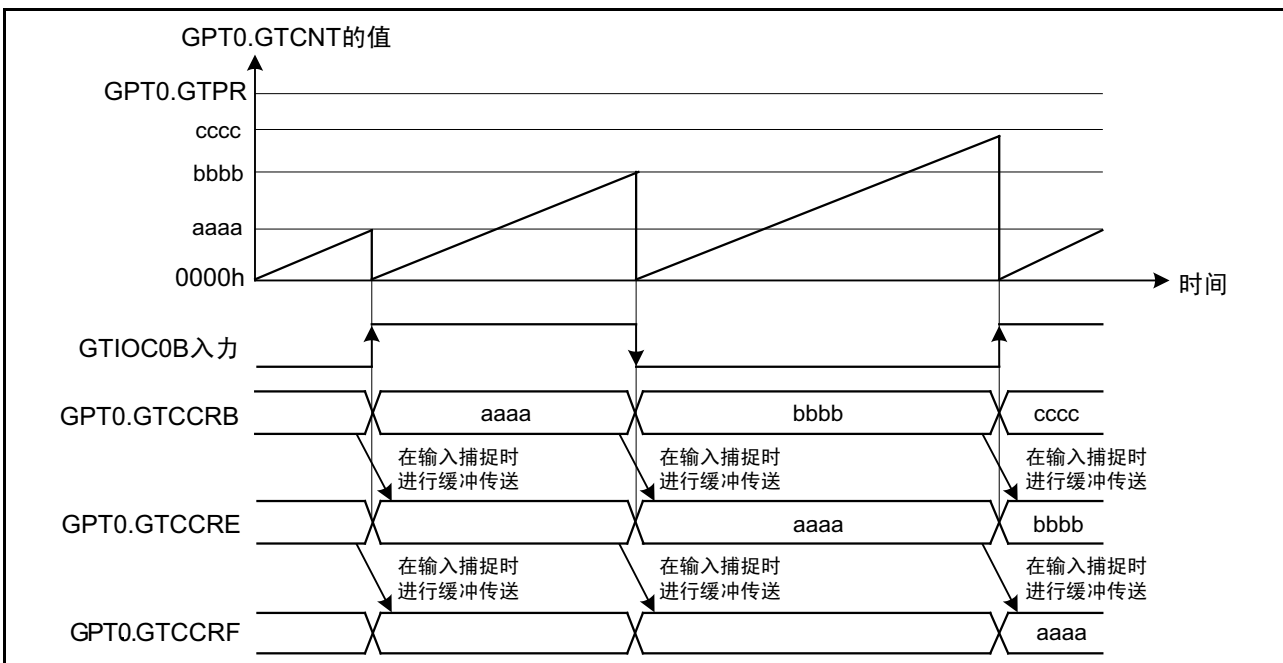


图 18.22 GTCCRA 寄存器和 GTCCRB 寄存器的双缓冲运行例子  
(在锯齿波进行递增计数、在 GTIOC0B 输入的双边沿进行输入捕捉、在 GTCCRB 寄存器的输入捕捉时进行 GTCNT 计数器清除的情况)

<p><b>运行模式和计数器清除源的设定</b>          通过GTCR寄存器的MD[2:0]位设定运行模式，通过CCLR[1:0]位设定计数器清除源。          (在图18.21的例子中，MD[2:0]=000b(锯齿波PWM模式)、CCLR[1:0]=01b，在图18.22的例子中，MD[2:0]=000b(锯齿波PWM模式)、CCLR[1:0]=10b。)</p>
<p><b>计数方向的设定</b>          通过GTUDC寄存器设定计数方向(递增/递减)。          (在图18.21的例子中，给GTUDC寄存器先设定“0003h”，再设定“0001h”。(递增计数)。)</p>
<p><b>计数时钟的选择</b>          通过GTCR.TPCS[1:0]位选择计数时钟。</p>
<p><b>周期的设定</b>          给GTPR寄存器设定周期。</p>
<p><b>计数器初始值的设定</b>          给GTCNT寄存器设定初始值。</p>
<p><b>GTIOC引脚功能的设定</b>          给GTIOR寄存器的GTIOA[5:0]位、GTIOB[5:0]位设定GTIOC引脚的功能。          (在图18.21的例子中，GTIOA[5:0]=1xxx1xb，在图18.22的例子中，GTIOB[5:0]=1xxx1xb。)</p>
<p><b>缓冲运行的设定</b>          给GTBER寄存器的CCRA位、CCRB位设定缓冲运行。          (在图18.21的例子中，CCRA=01b，在图18.22的例子中，CCRB=1xb。)</p>
<p><b>计数开始</b>          将GTSTR.CST位置“1”，开始计数。          (在图18.21、图18.22的例子中，GTSTR.CST0位为“1”。)</p>

图 18.23 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)



18.3.2.3 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 18.24 ~ 图 18.26，GTADTRA 寄存器和 GTADTRB 寄存器的缓冲运行设定例子如图 18.27 所示。

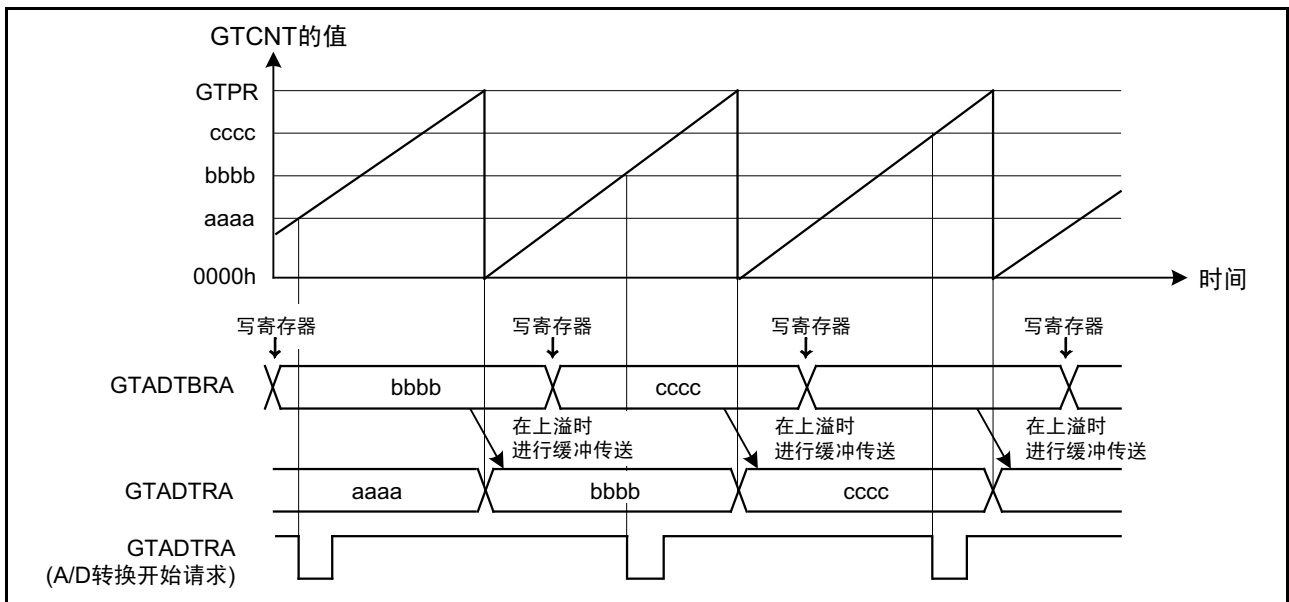


图 18.24 GTADTRA 寄存器和 GTADTRB 寄存器的缓冲运行例子  
（在锯齿波进行递增计数、在递增计数时产生 A/D 转换开始请求的情况）

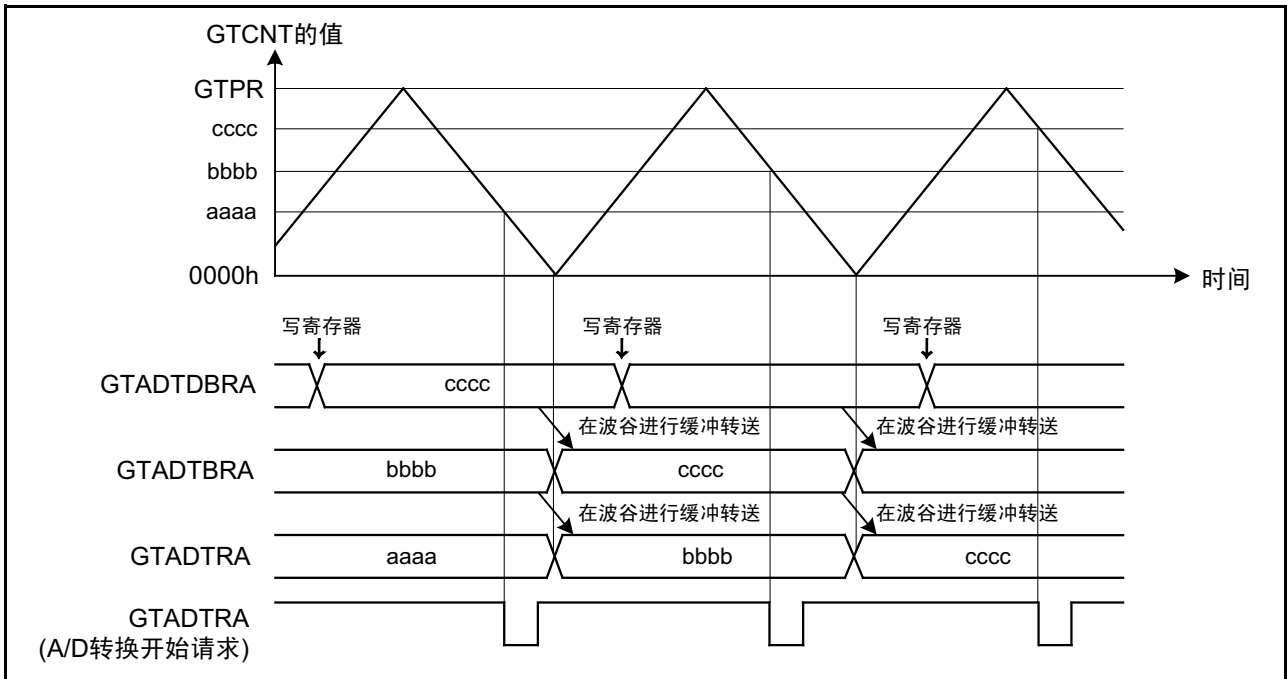


图 18.25 GTADTRA 寄存器和 GTADTRB 寄存器的双缓冲运行例子  
(三角波、在波谷进行缓冲转送、在递减计数时产生 A/D 转换请求的情况)

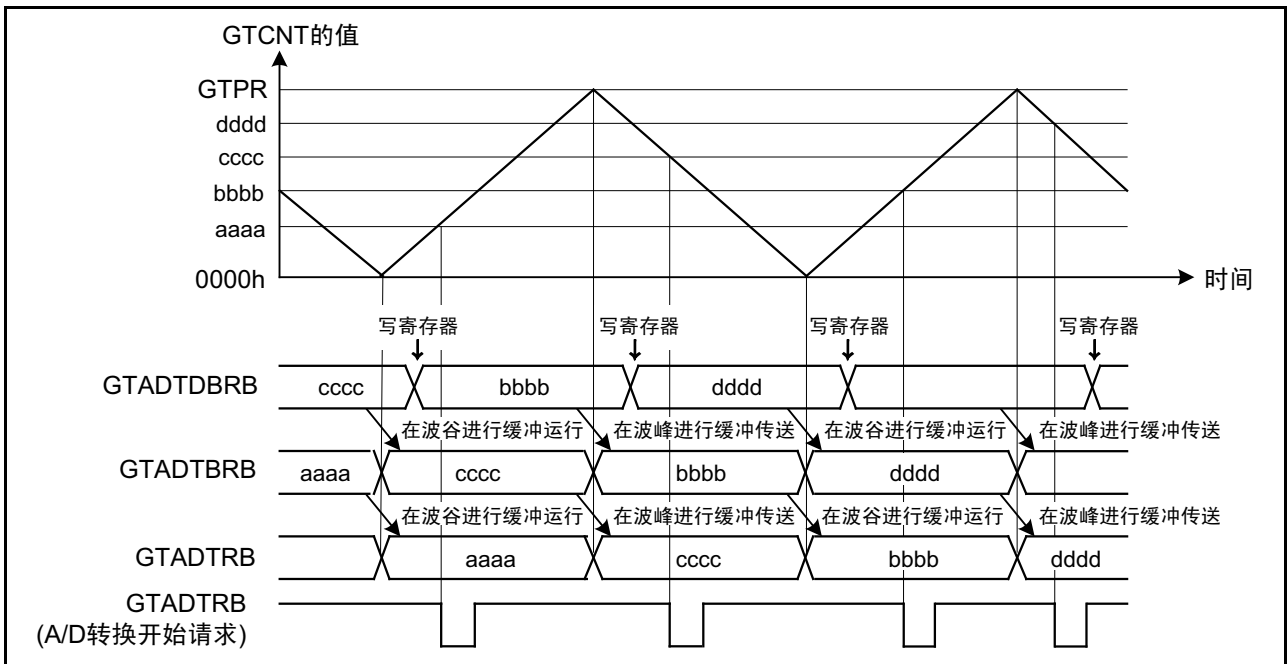


图 18.26 GTADTRA 寄存器和 GTADTRB 寄存器的双缓冲运行例子  
(三角波、在波谷和波峰都进行缓冲转送、在递增计数和递减计数时都产生 A/D 转换请求的情况)

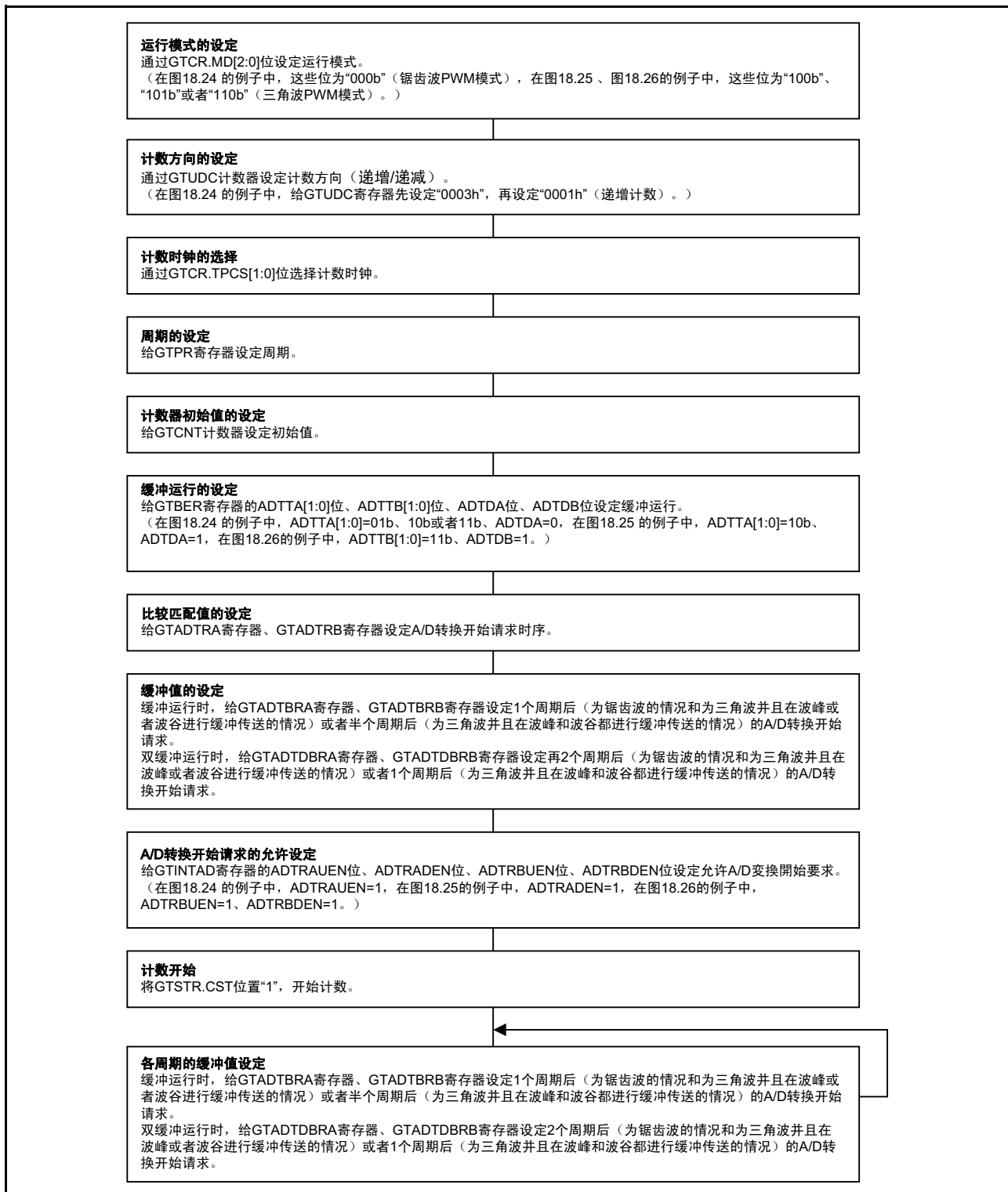


图 18.27 GTADTRA 寄存器和 GTADTRB 寄存器的缓冲运行设定例子

### 18.3.3 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 18.28，锯齿波 PWM 模式的设定例子如图 18.29 所示。

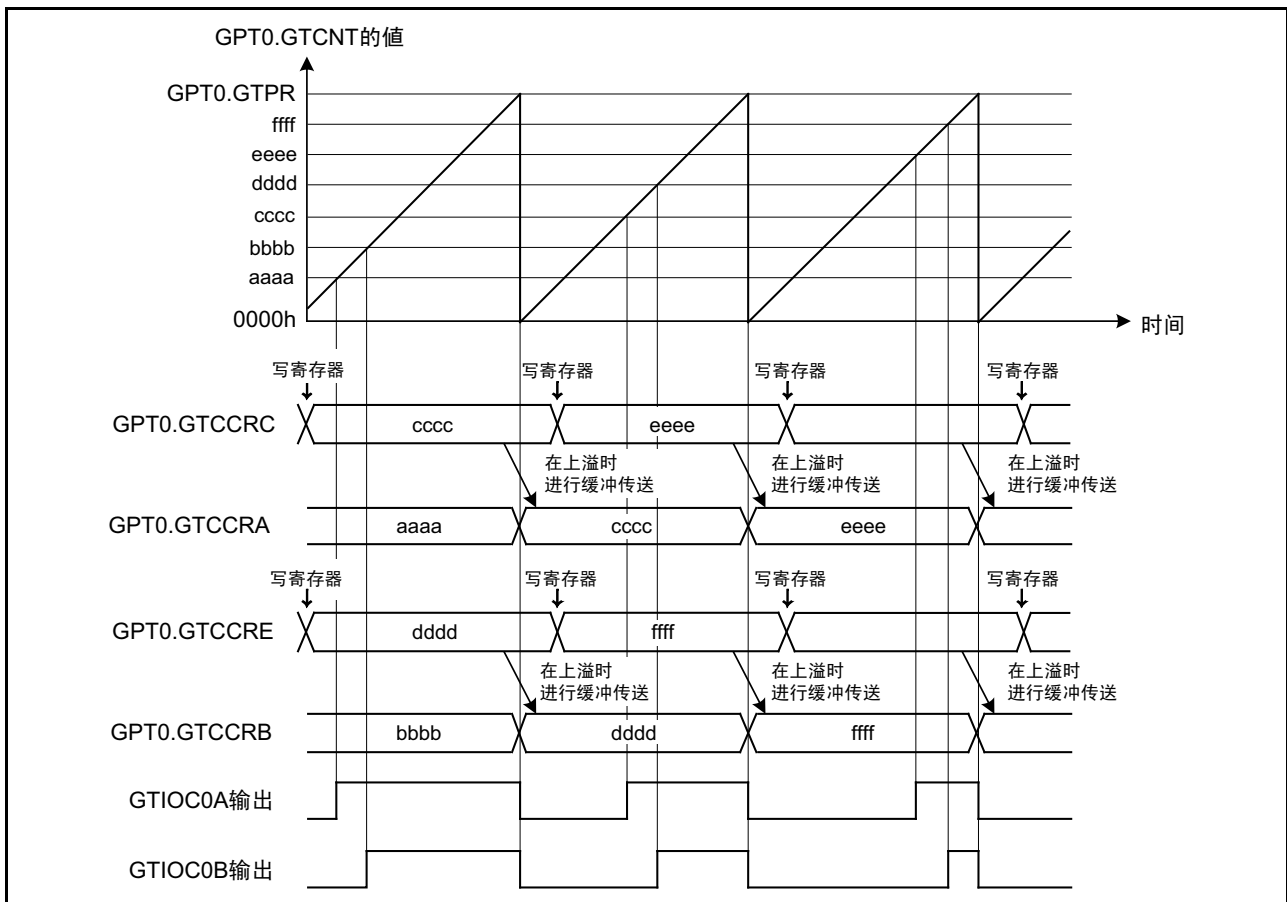


图 18.28 锯齿波 PWM 模式的运行例子  
(递增计数、缓冲运行、在 GTCCRA/B 比较匹配时进行 High 电平输出、在周期结束时进行 Low 电平输出的情况)

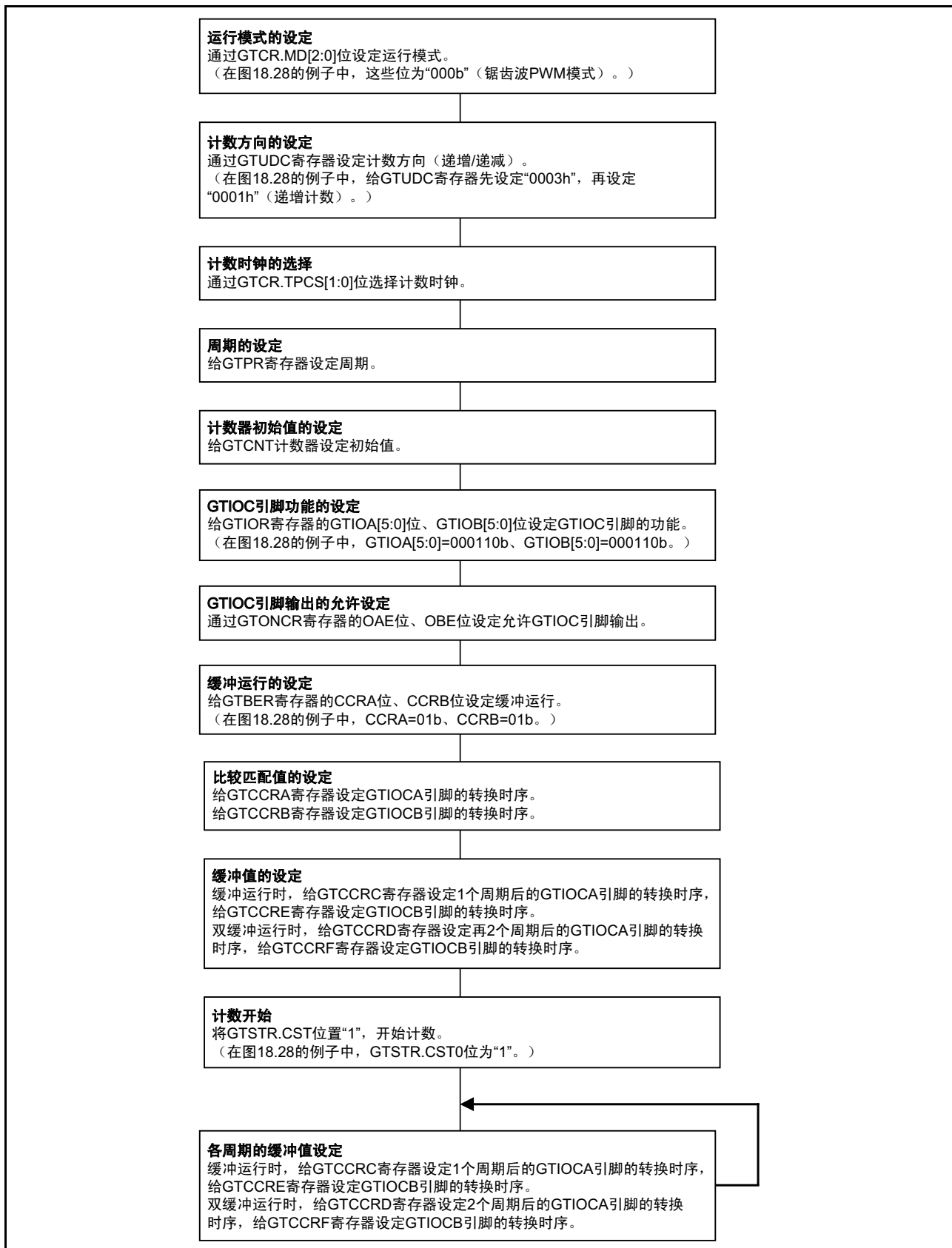


图 18.29 锯齿波 PWM 模式的设定例子

## (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 18.30，锯齿波单触发脉冲模式的设定例子如图 18.31 所示。

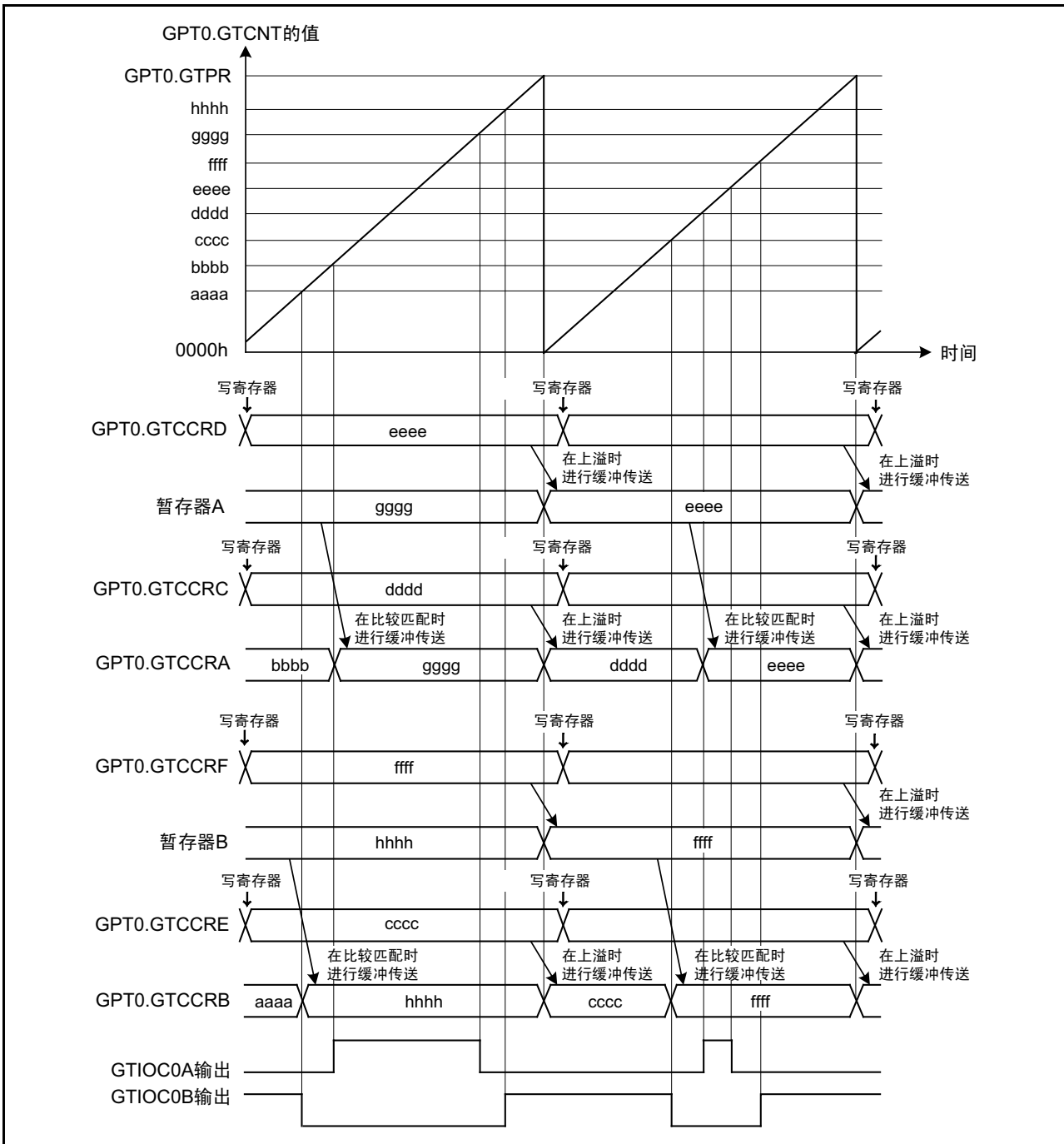


图 18.30 锯齿波单触发脉冲模式的运行例子  
 (递增计数、在开始计数时 GTIOC0A 引脚为 Low 电平输出 /GTIOC0B 引脚为 High 电平输出、在 GTCCRA/B 比较匹配时进行交替输出、在周期结束保持输出的情况)

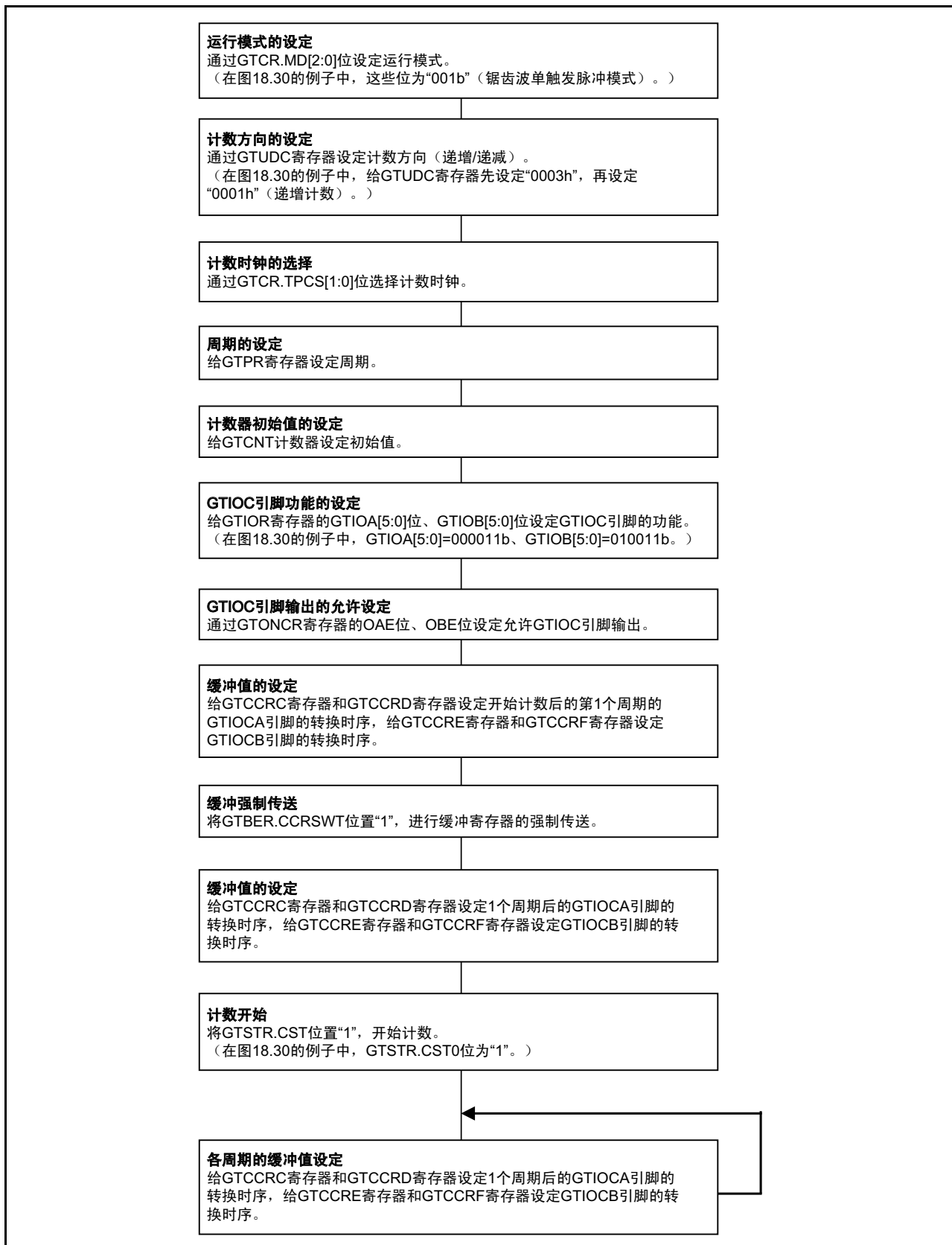


图 18.31 锯齿波单触发脉冲模式的设定例子



(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 18.32，三角波 PWM 模式 1 的设定例子如图 18.33 所示。

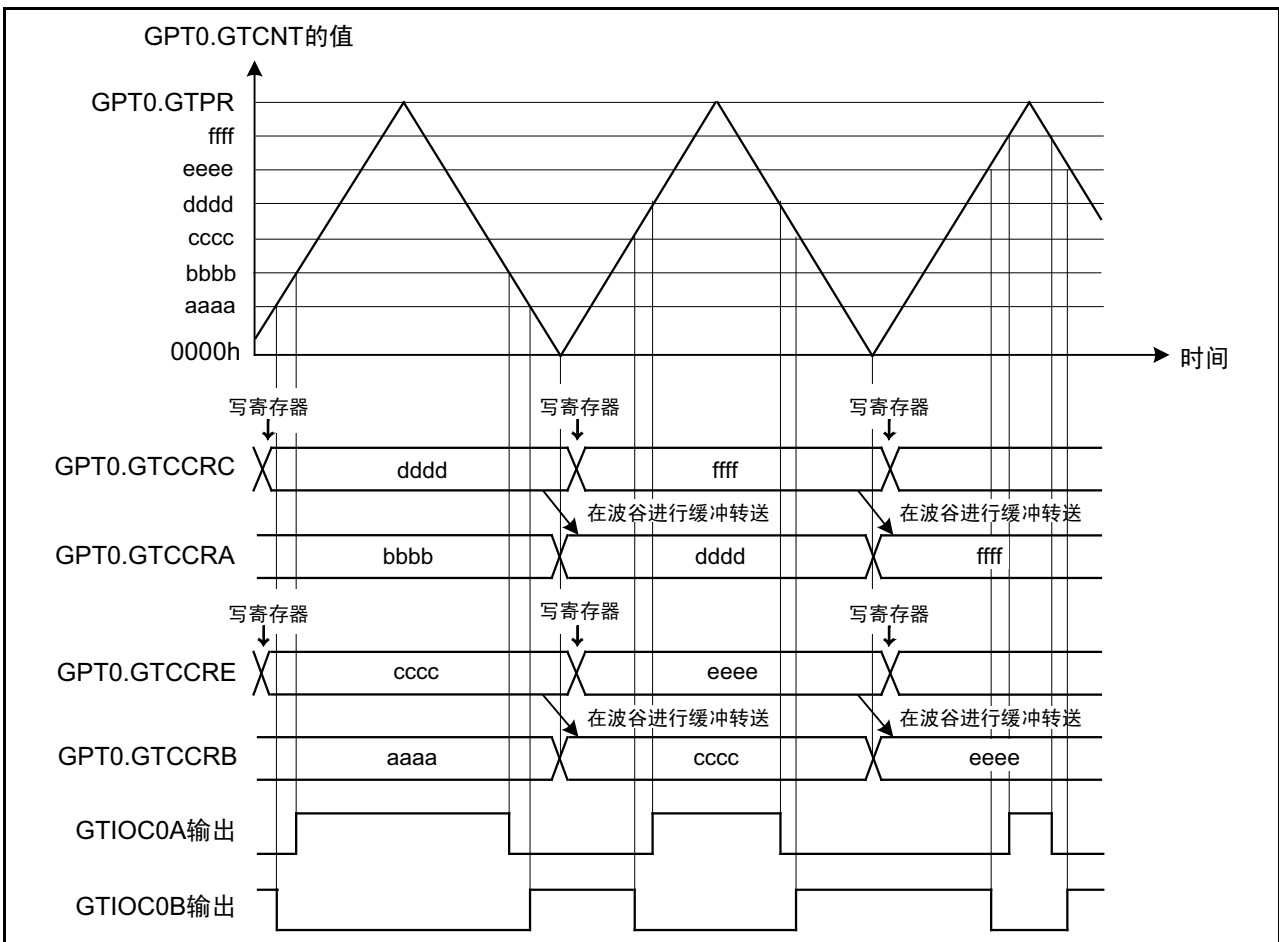


图 18.32 三角波 PWM 模式 1 的运行例子

(缓冲运行、在开始计数时 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)

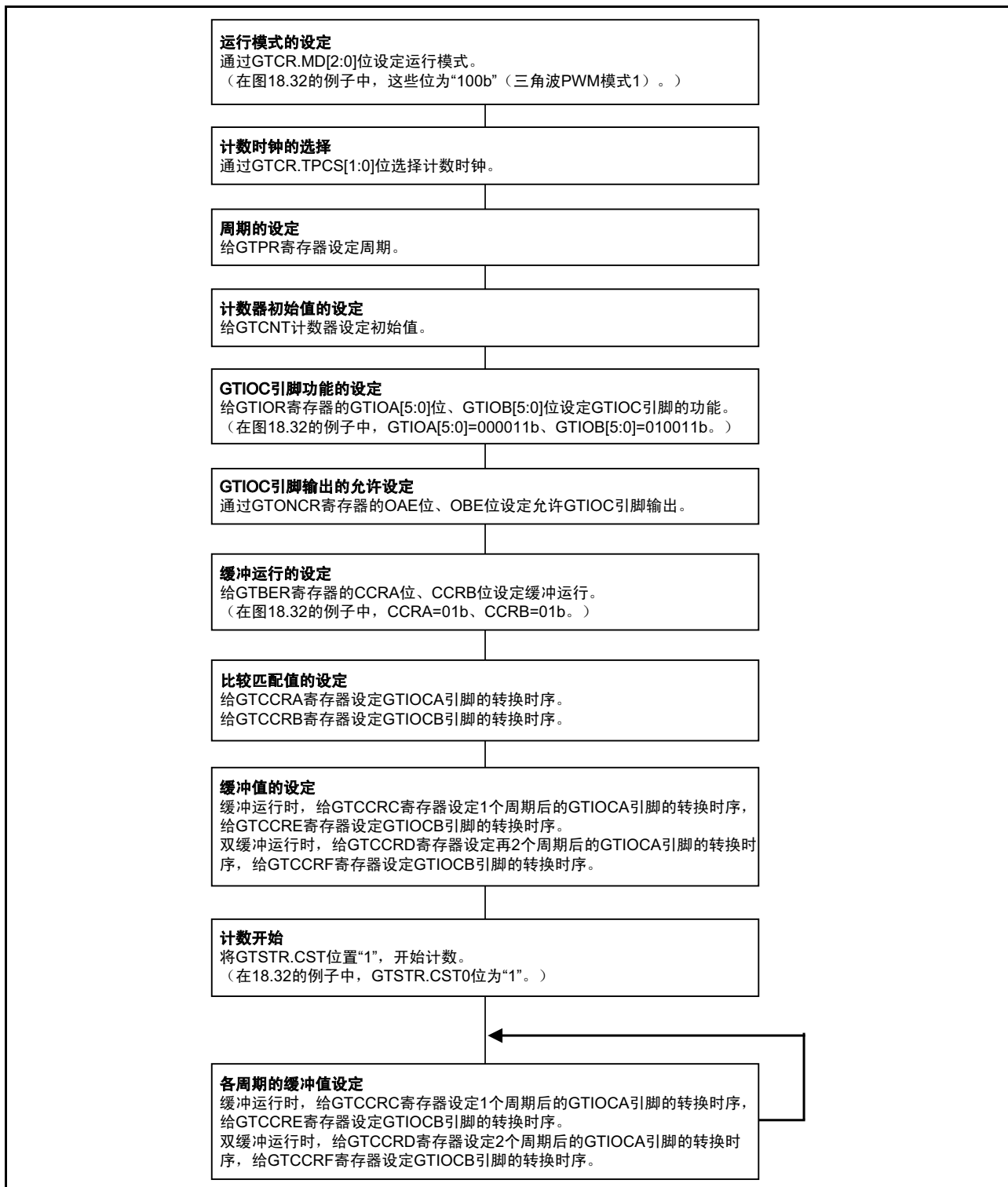


图 18.33 三角波 PWM 模式 1 的设定例子

(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 18.34，三角波 PWM 模式 2 的设定例子如图 18.35 所示。

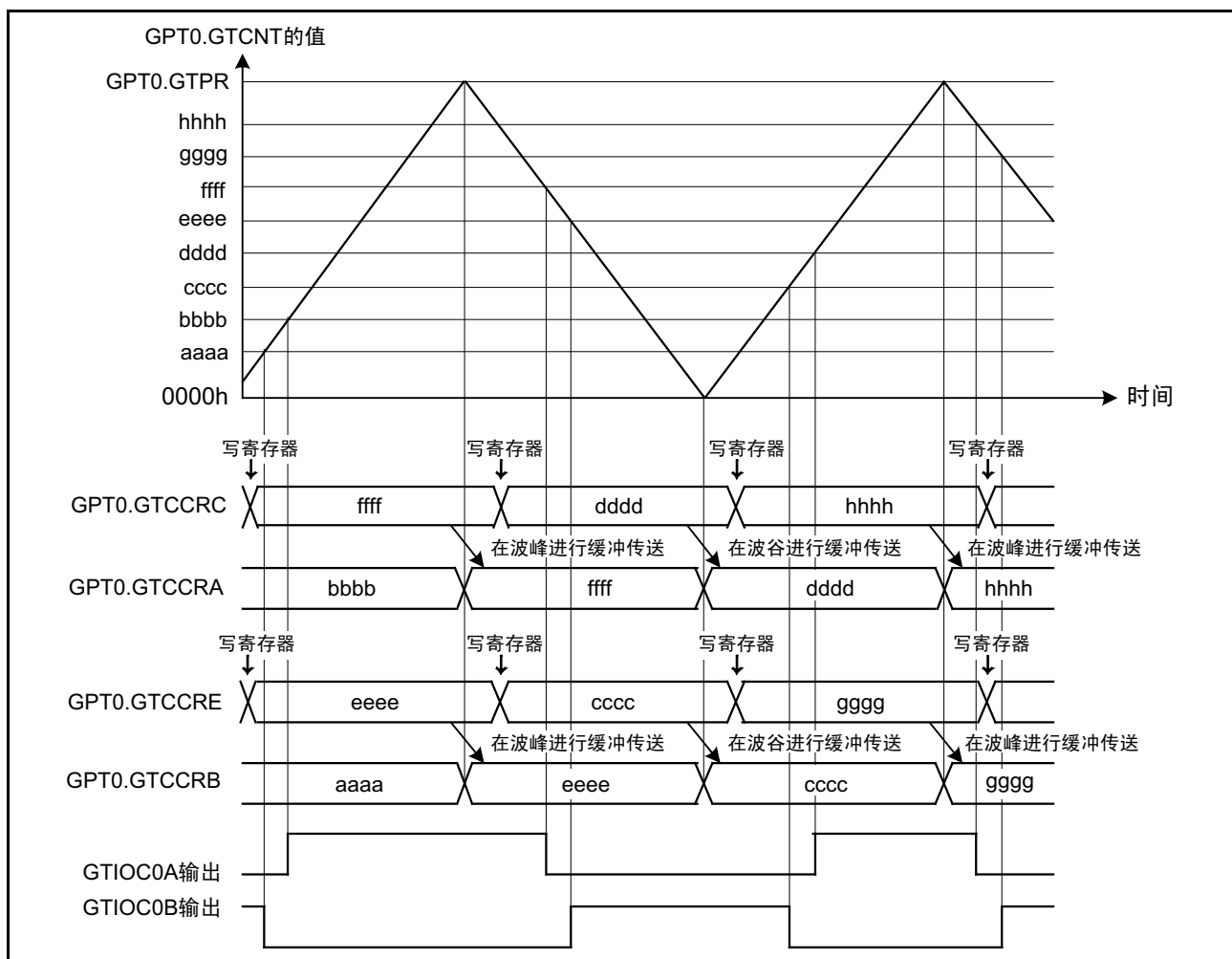


图 18.34 三角波 PWM 模式 2 的运行例子

(缓冲运行、在开始计数时 GTIOC0A 引脚为 Low 电平输出 /GTIOC0B 引脚 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)

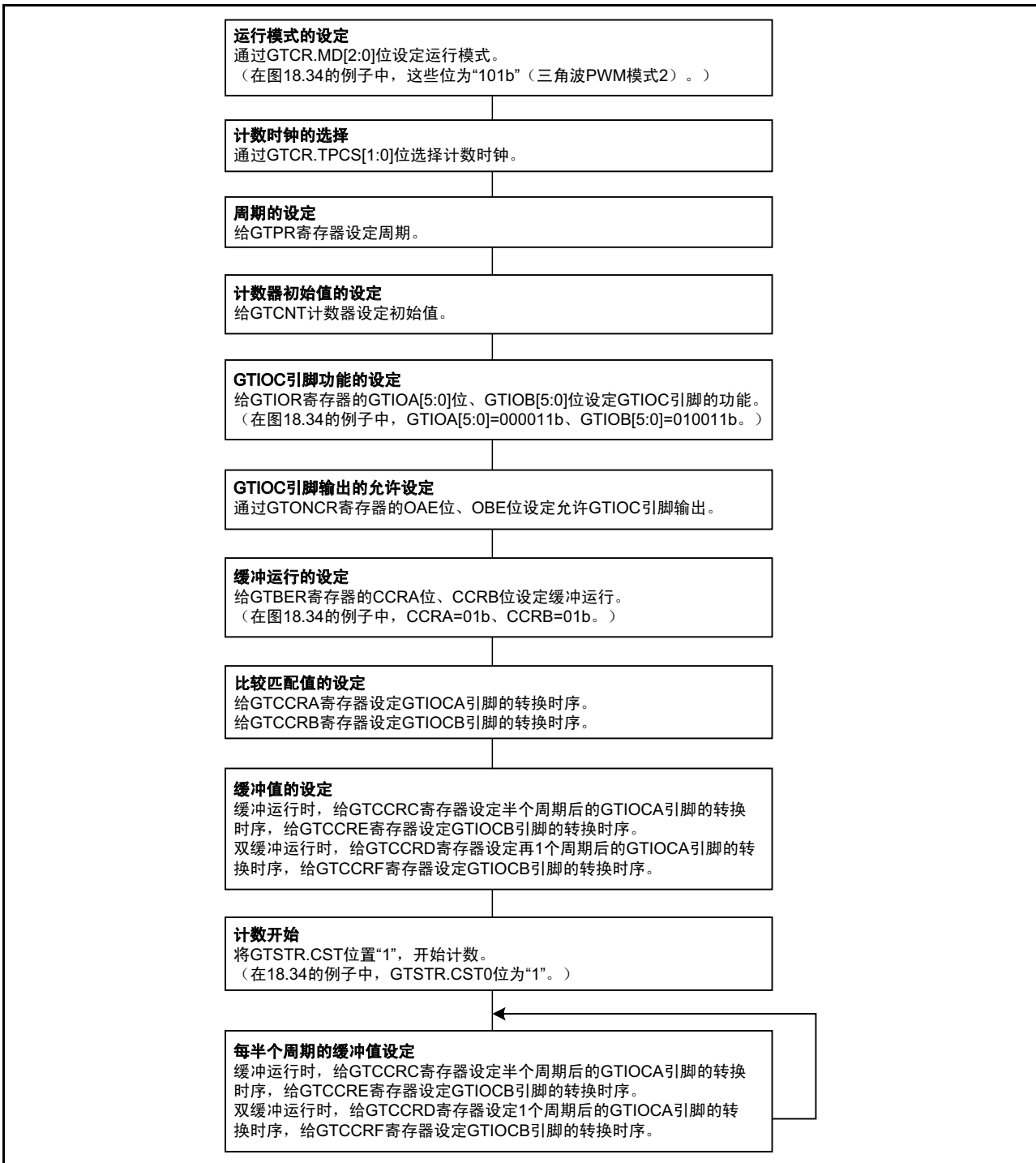


图 18.35 三角波 PWM 模式 2 的设定例子

### (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 18.36, 三角波 PWM 模式 3 的设定例子如图 18.37 所示。

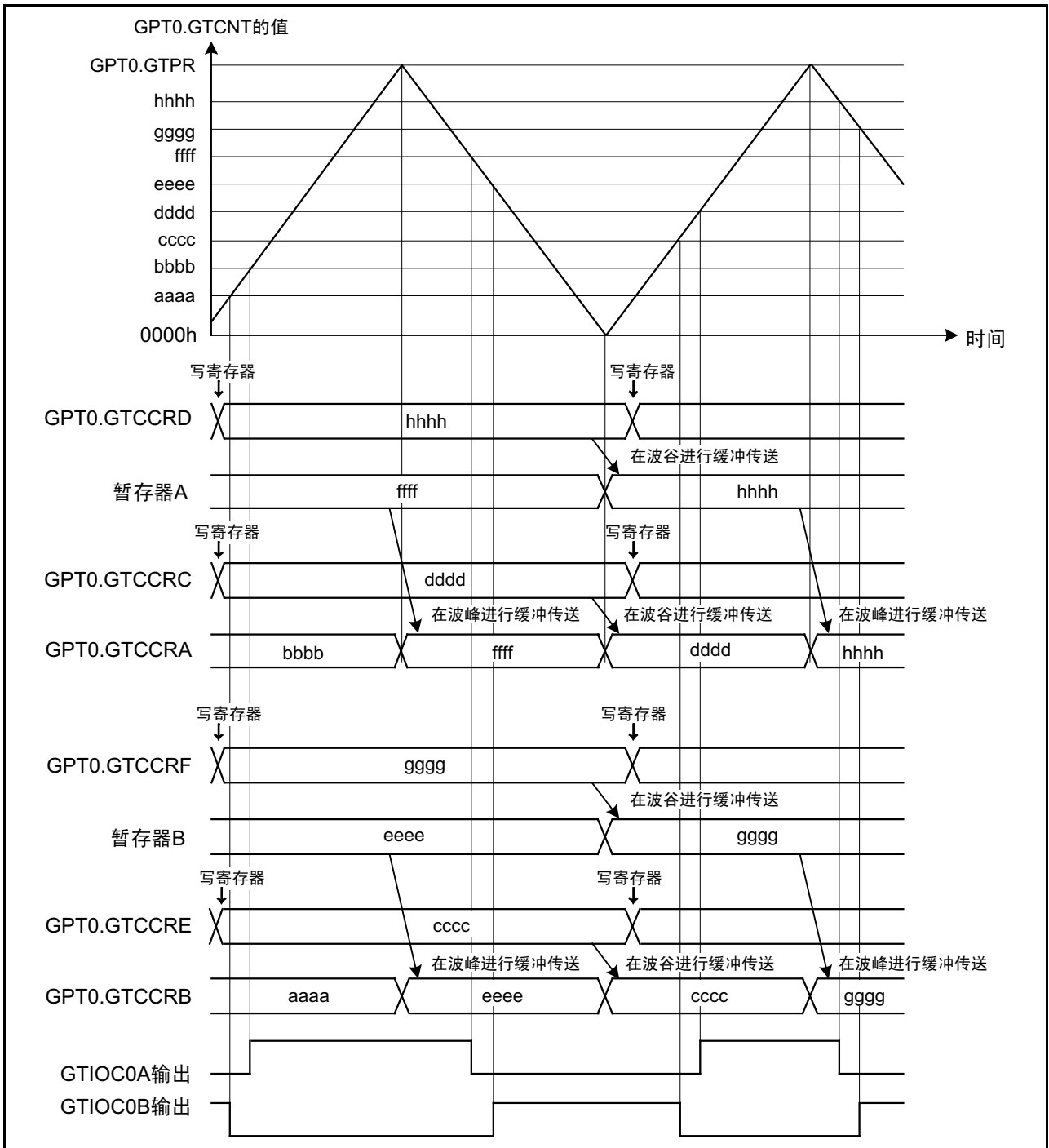


图 18.36 三角波 PWM 模式 3 的运行例子  
 (在开始计数时 GTIOC0A 引脚为 Low 电平输出 /GTIOC0B 引脚为 High 电平输出、在 GTCARA/ B 比较匹配时进行交替输出、在周期结束时保持输出的情况)

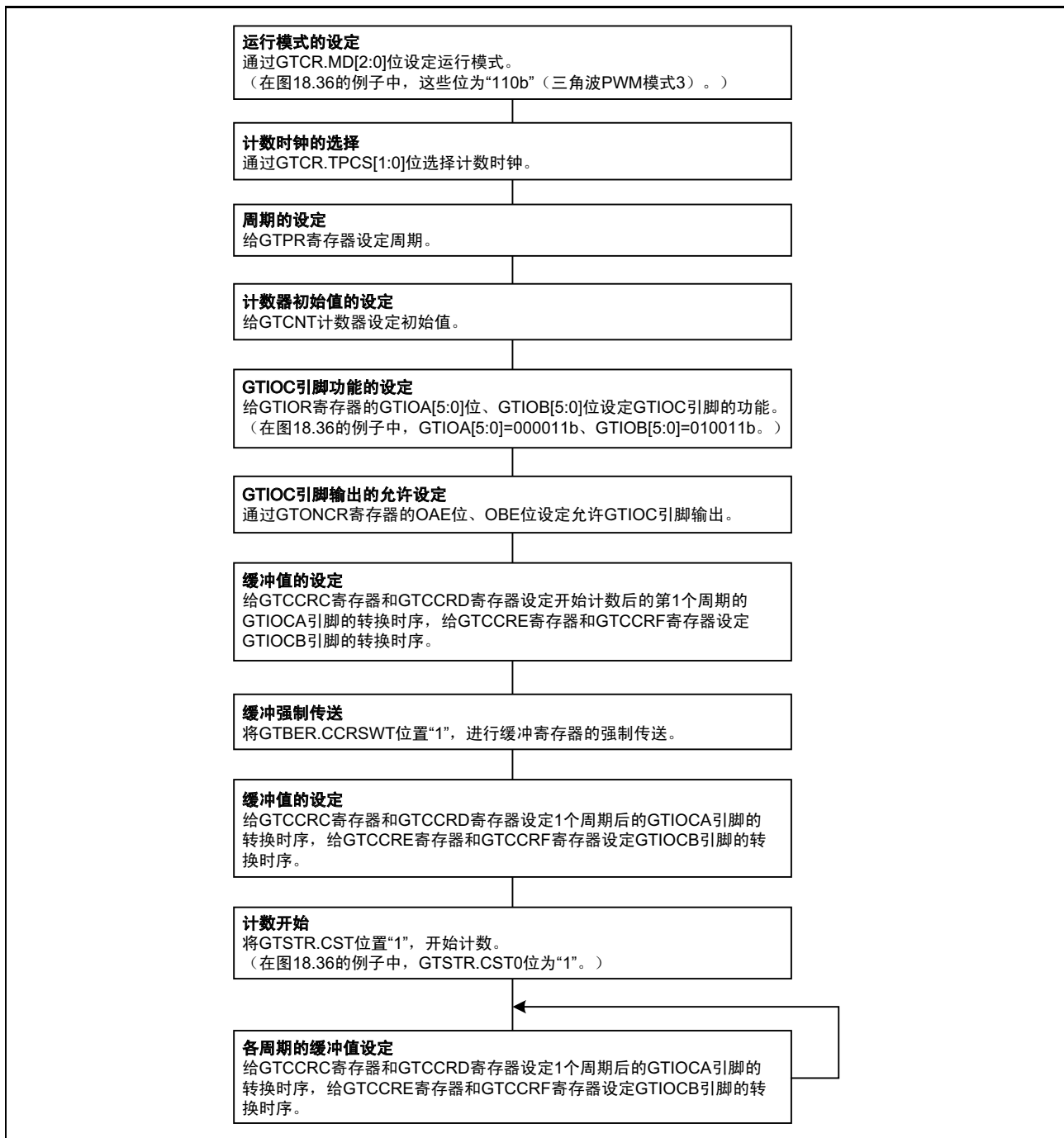


图 18.37 三角波 PWM 模式 3 的设定例子

### 18.3.4 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 18.38 ~ 图 18.40，死区时间自动设定功能的设定例子如图 18.41、图 18.42 所示。

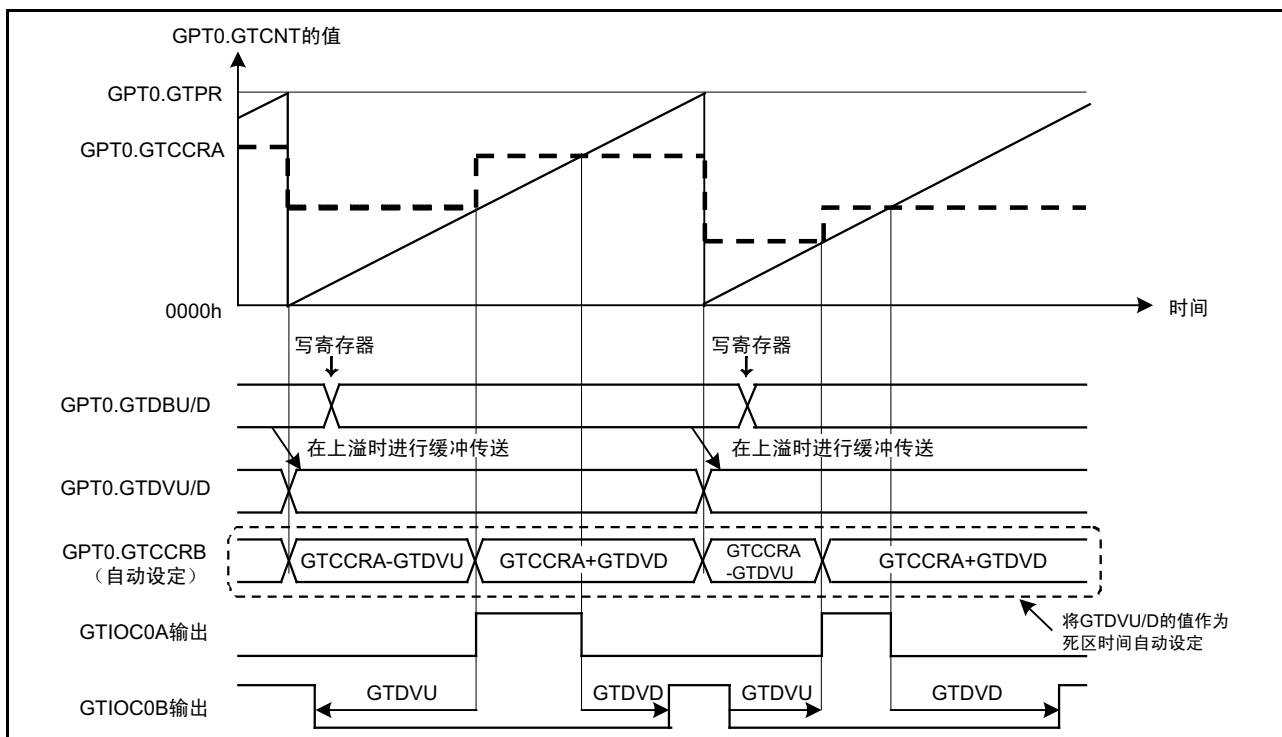


图 18.38 死区时间自动设定功能的运行例子  
(锯齿波单触发脉冲模式、GTDVU/D 寄存器进行缓冲运行、有效电平为 High 电平的情况)



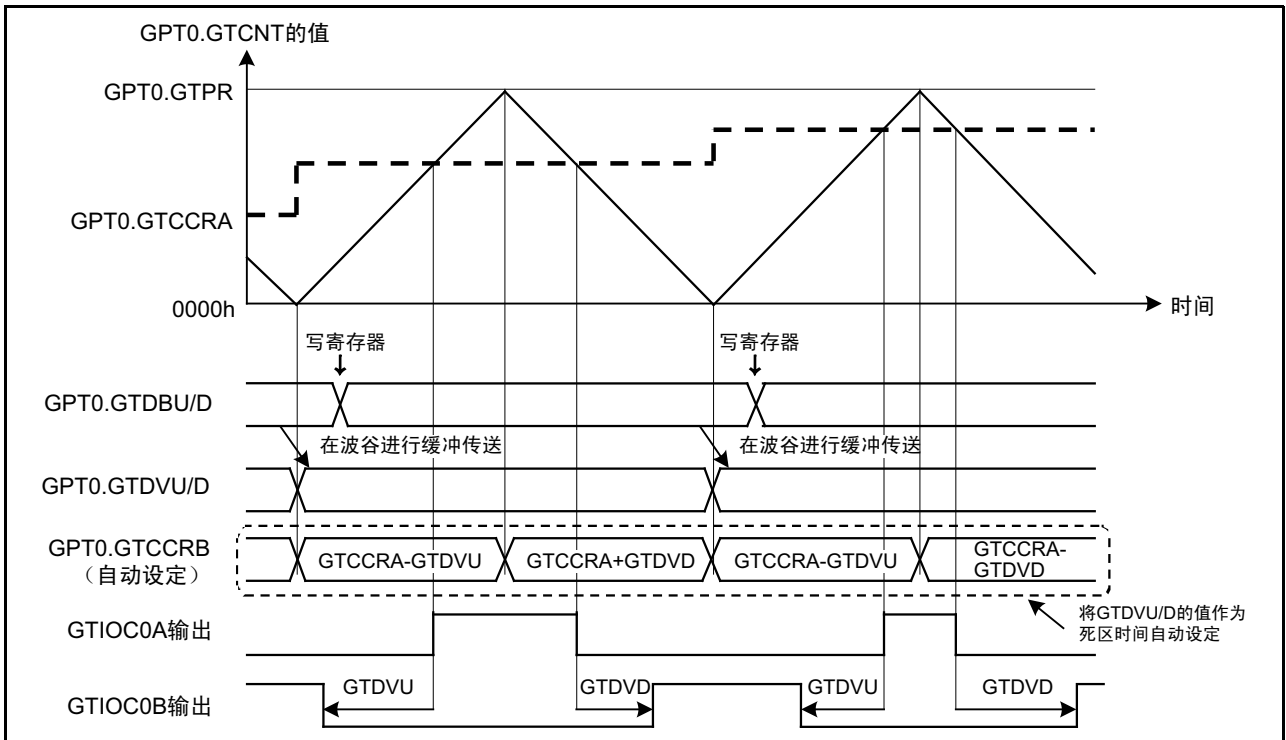


图 18.39 带死区时间自动设定功能的运行例子  
(三角波 PWM 模式 1、GTDVU/D 寄存器进行缓冲运行、有效电平为 High 电平的情况)

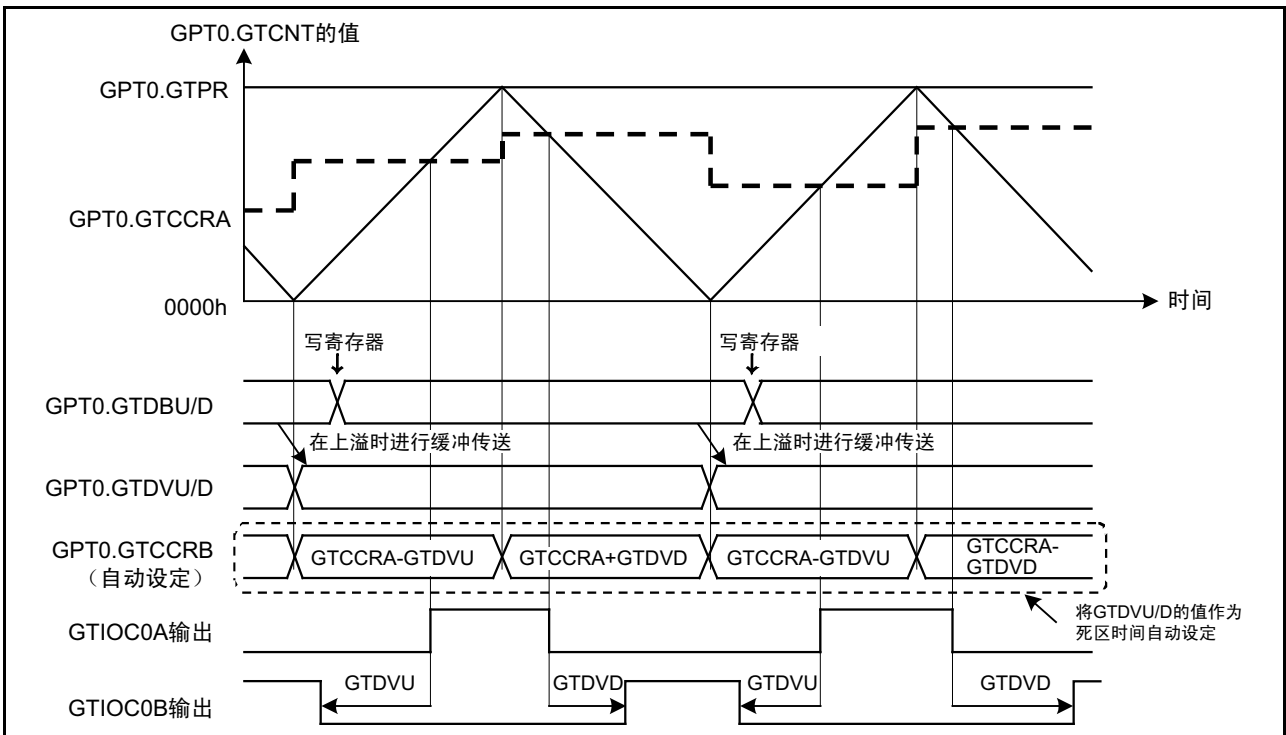


图 18.40 死区时间自动设定功能的运行例子  
(三角波 PWM 模式 2 和 3、GTDVU/D 寄存器进行缓冲运行、有效电平为 High 电平的情况)

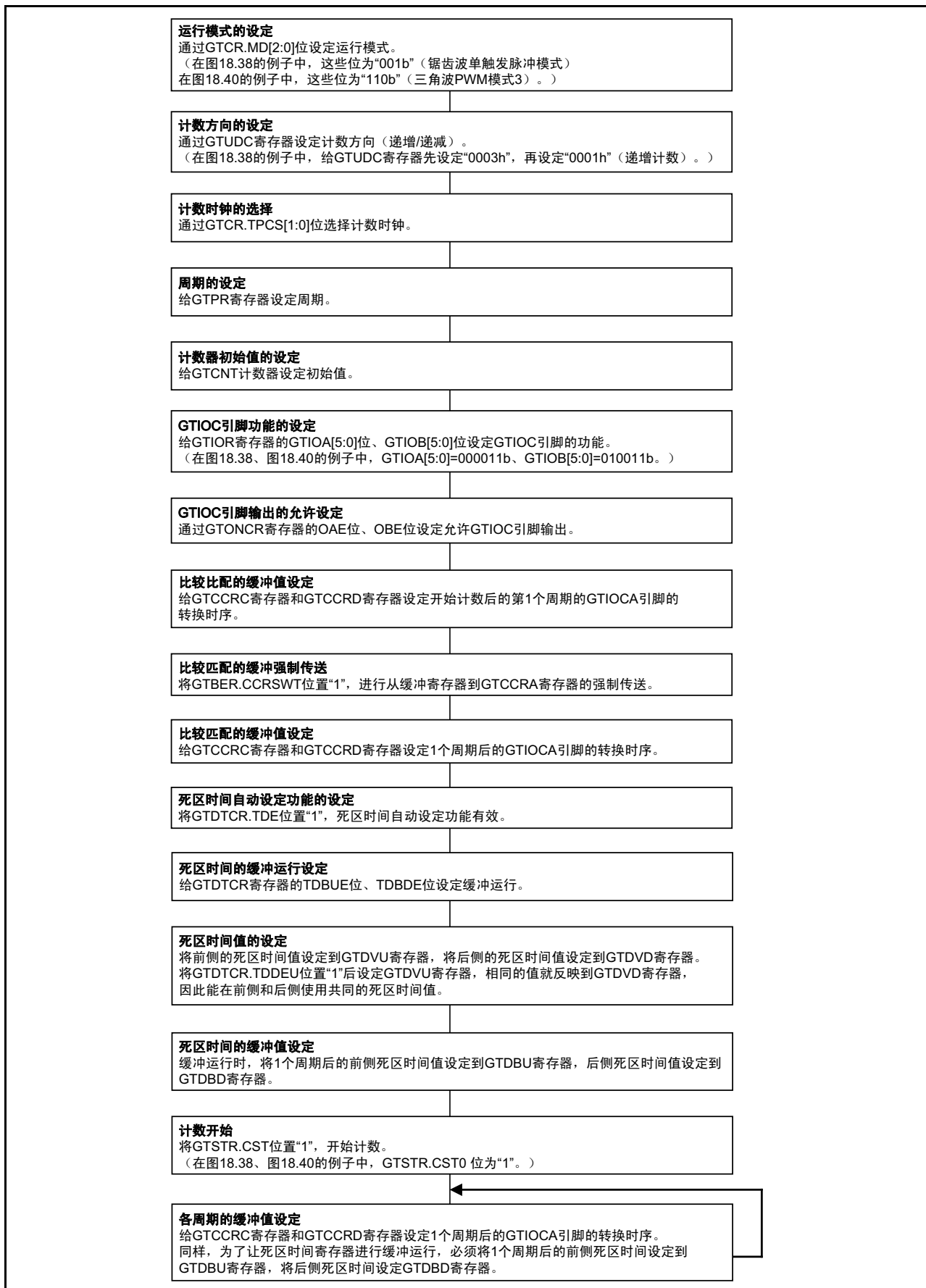


图 18.41 死区时间自动设定功能的设定例子 (锯齿波单触发脉冲模式、三角波 PWM 模式 3 的情况)

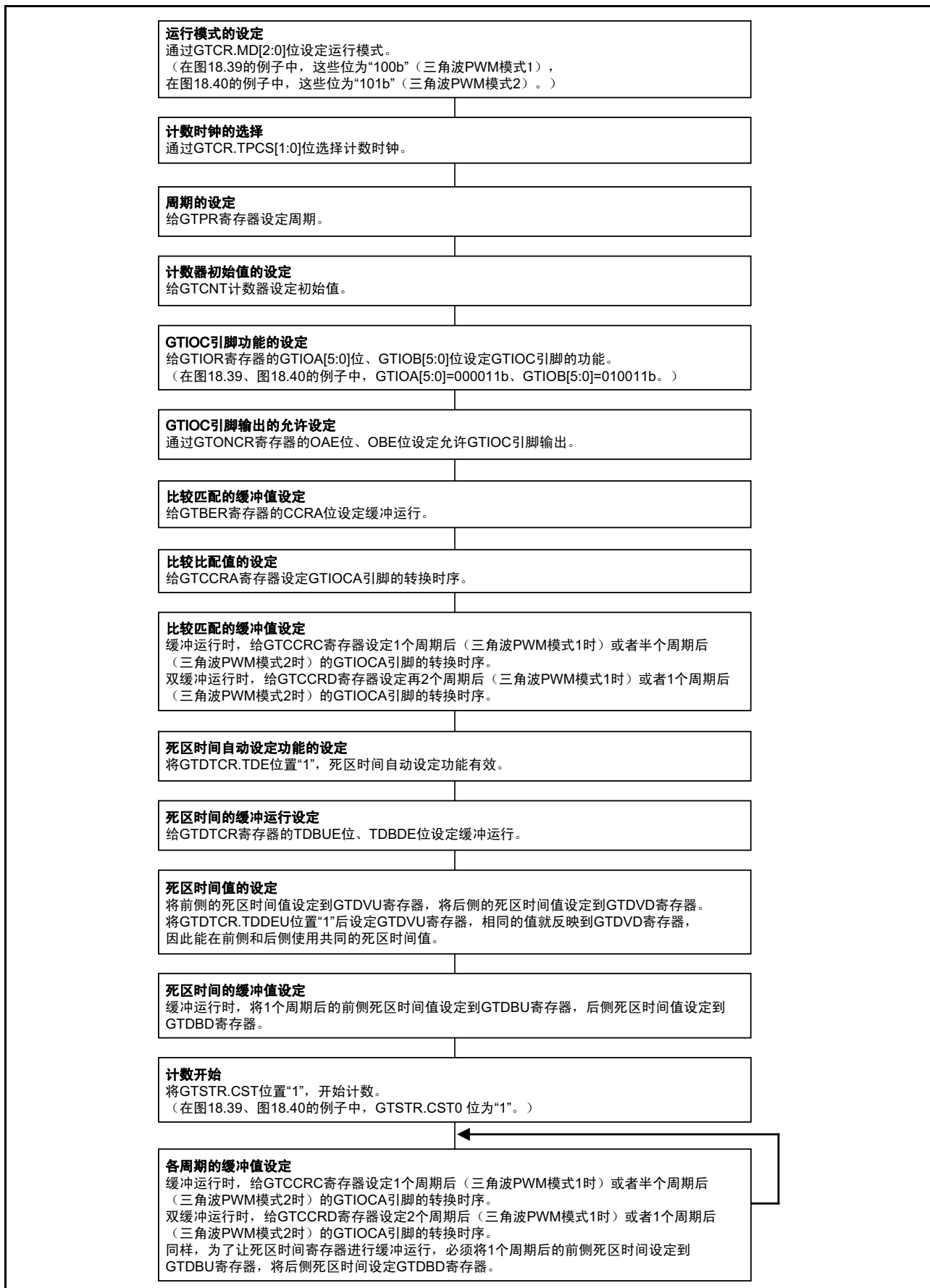


图 18.42 死区时间自动设定功能的设定例子 (三角波 PWM 模式 1 和 2 的情况)

### 18.3.5 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 18.43 所示。

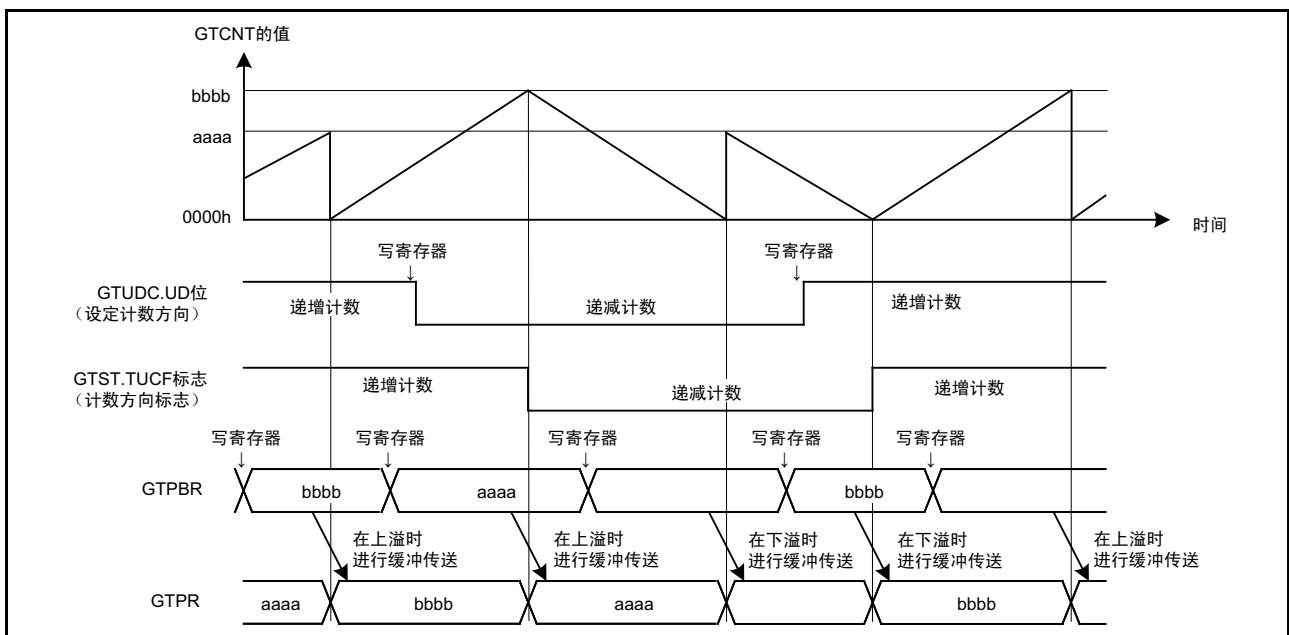


图 18.43 计数方向转换功能的运行例子（缓冲运行时）

### 18.3.6 硬件启动 / 停止、清除运行

能通过 RX62T 内部的硬件源对 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 GTETRGR 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。

#### 18.3.6.1 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 18.44 所示，通过硬件源启动计数运行的设定例子如图 18.45 所示。

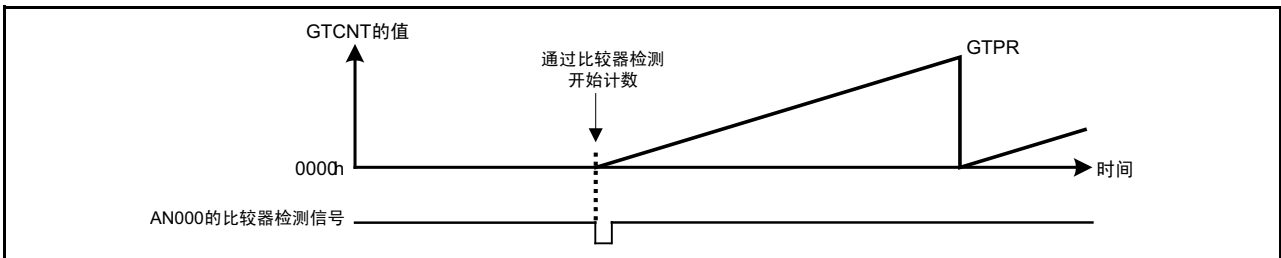


图 18.44 通过硬件源启动计数的运行例子（通过 AN000 的比较器检测开始计数的情况）

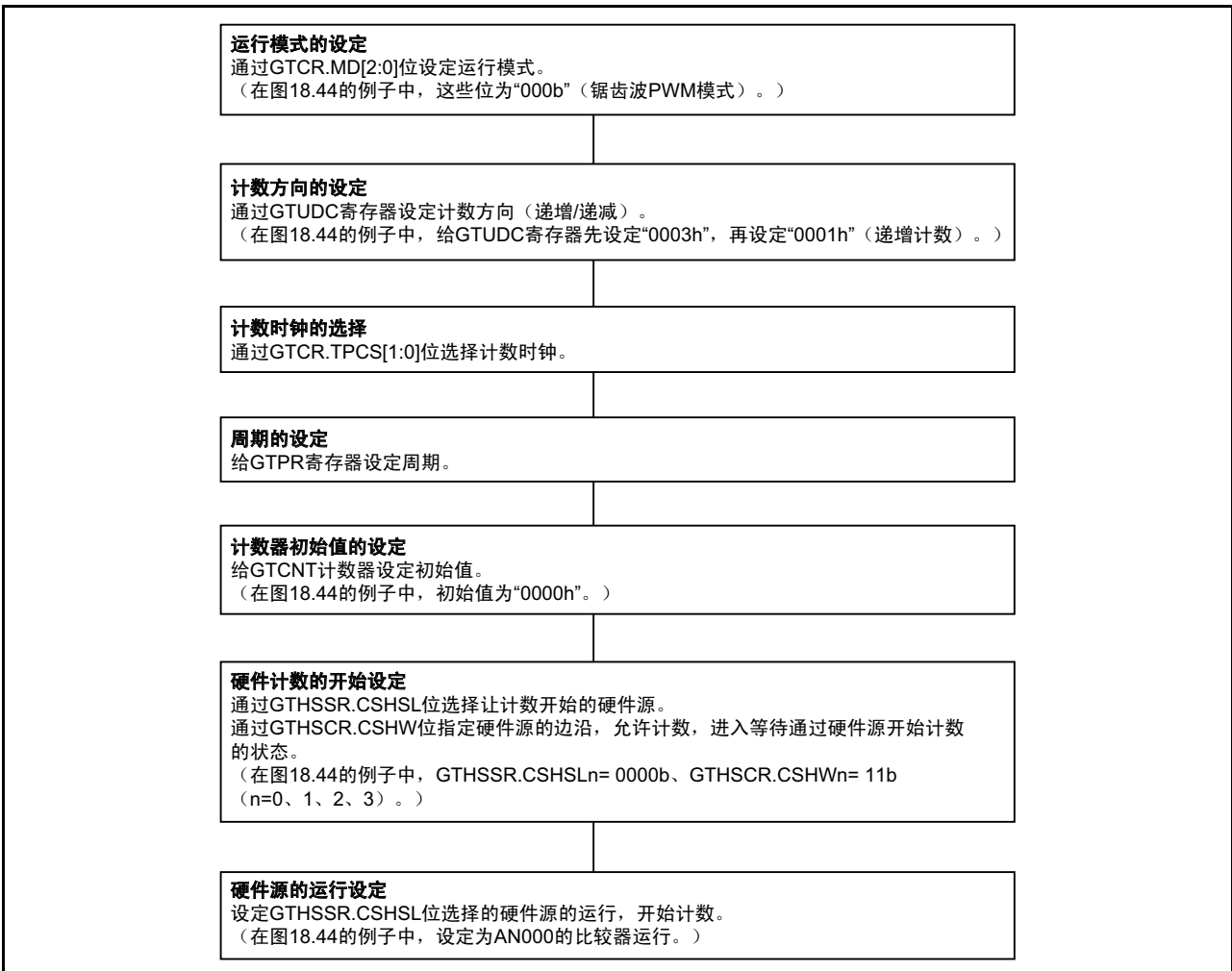


图 18.45 通过硬件源启动计数运行的设定例子

18.3.6.2 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 18.46 所示，通过硬件源停止计数运行的设定例子如图 18.47 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。

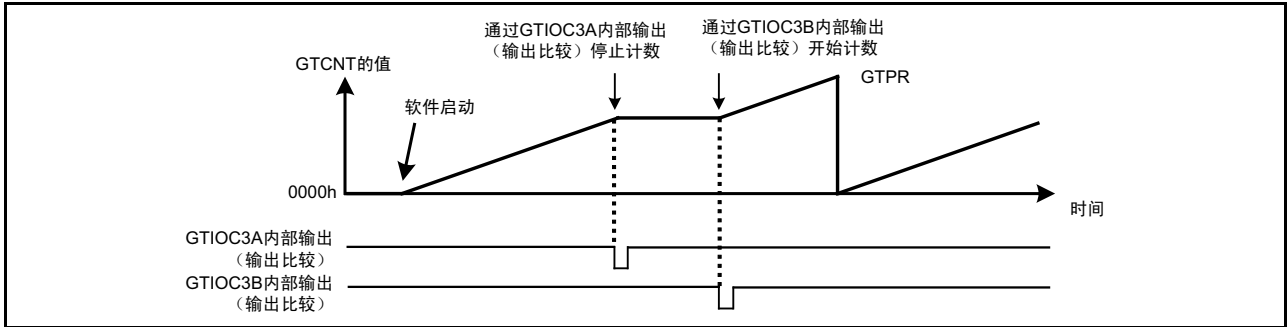


图 18.46 通过硬件源停止计数的运行例子

（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><b>运行模式的设定</b> 通过GTCR.MD[2:0]位设定运行模式。 （在图18.46的例子中，这些位为“000b”（锯齿波PWM模式）。）</p>
<p><b>计数方向的设定</b> 通过GTUDC寄存器设定计数方向（递增/递减）。 （在图18.46的例子中，给GTUDC寄存器先设定“0003h”，再设定“0001h”（递增计数）。）</p>
<p><b>计数时钟的选择</b> 通过GTCR.TPCS[1:0]位选择计数时钟。</p>
<p><b>周期的设定</b> 给GTPR寄存器设定周期。</p>
<p><b>计数器初始值的设定</b> 给GTCNT计数器设定初始值。 （在图18.46的例子中，初始值为“0000h”。）</p>
<p><b>硬件计数的开始设定</b> 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18.46的例子中，GTHSSR.CSHSLn= 1010b、GTHSCR.CSHWn= 11b（n=0~2）。）</p>
<p><b>硬件计数的停止设定</b> 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18.46的例子中，GTHPSR.CSHPLn= 1011b、GTHSCR.CPHWn= 11b（n=0~2）。）</p>
<p><b>硬件源的运行设定</b> 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18.46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 18.47 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 18.48 所示。通过硬件源开始或者停止计数运行的设定例子如图 18.49 所示。在此例子中，外部输入触发 GTETRГ 在 High 电平区间进行计数。

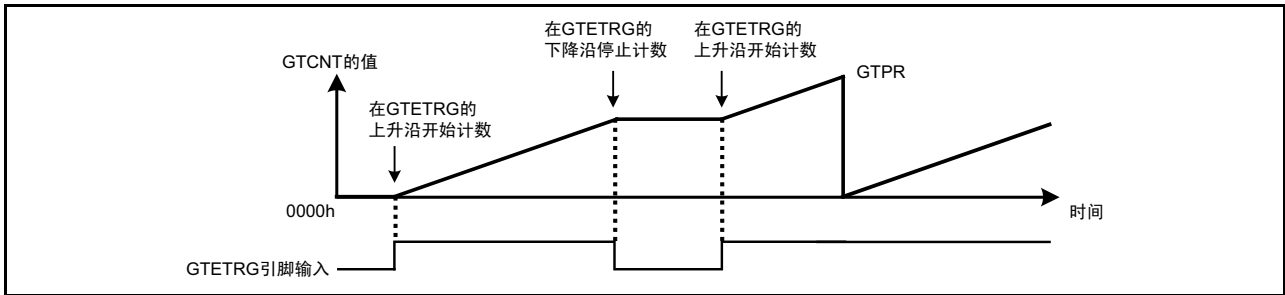


图 18.48 通过硬件源开始或者停止计数的运行例子  
(在 GTETRГ 引脚输入的上升沿开始计数，在 GTETRГ 引脚输入的下沿停止计数)

<p><b>运行模式的设定</b> 通过GTCR.MD[2:0]位设定运行模式。 (在图18.48的例子中，这些位为“000b”(锯齿波PWM模式)。)</p>
<p><b>计数方向的设定</b> 通过GTUDC寄存器设定计数方向(递增/递减)。 (在图18.48的例子中，给GTUDC寄存器先设定“0003h”，再设定“0001h”(递增计数)。)</p>
<p><b>计数时钟的选择</b> 通过GTCR.TPCS[1:0]位选择计数时钟。</p>
<p><b>周期的设定</b> 给GTPR寄存器设定周期。</p>
<p><b>计数器初始值的设定</b> 给GTCNT计数器设定初始值。 (在图18.48的例子中，初始值为“0000h”。)</p>
<p><b>硬件计数的开始设定</b> 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18.48的例子中，GTHSSR.CSHSLn= 1100b、GTHSCR.CSHWn= 01b (n=0~3)。)</p>
<p><b>硬件计数的停止设定</b> 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18.48的例子中，GTHPSR.CSHPLn= 1100b、GTHSCR.CPHWn= 10b (n=0~3)。)</p>
<p><b>硬件源的运行设定</b> 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18.48的例子中，设定GTETRГ引脚的运行。)</p>

图 18.49 通过硬件源开始或者停止计数运行的设定例子

18.3.6.3 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 18.50、图 18.51 所示，通过硬件源进行计数器清除运行的设定例子如图 18.52 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。

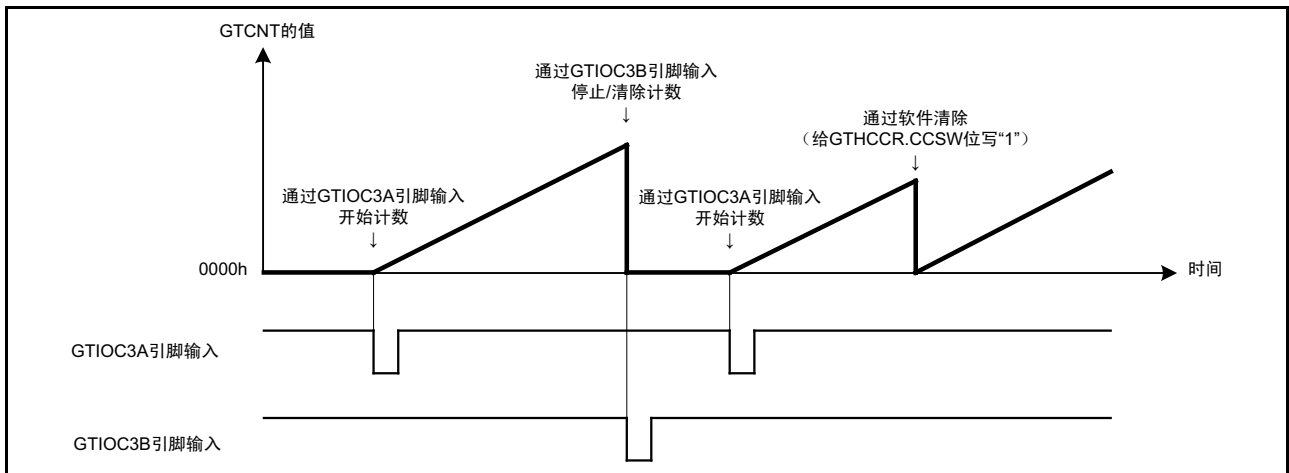


图 18.50 通过硬件源进行计数器清除的运行例子  
（在锯齿波进行递增计数、通过 GTIOC3A 引脚输入开始计数、通过 GTIOC3B 引脚输入停止或者清除计数）

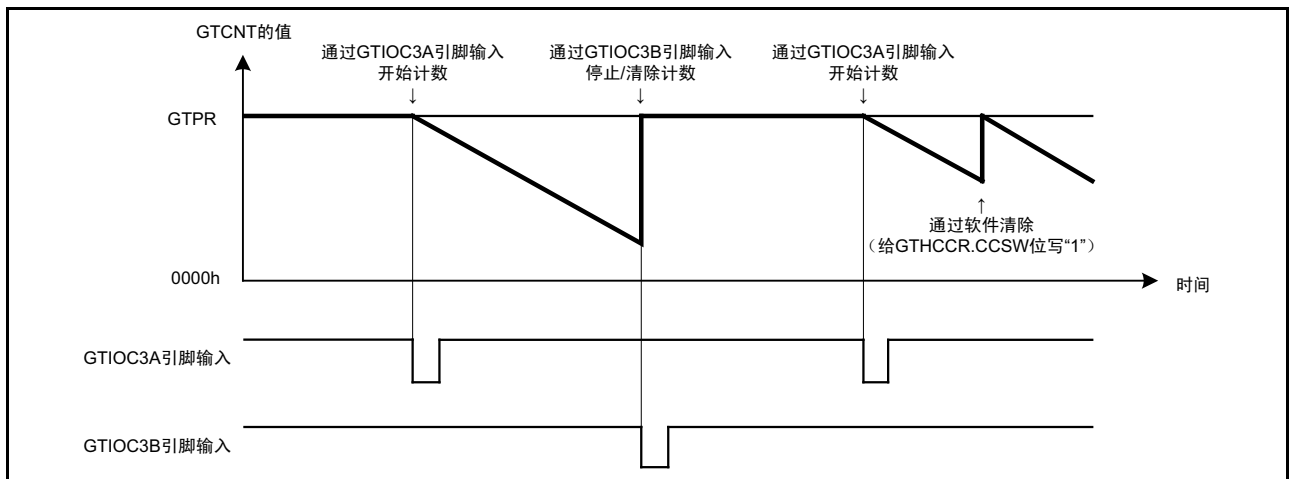


图 18.51 通过硬件源进行计数器清除的运行例子  
（在锯齿波进行递减计数、通过 GTIOC3A 引脚输入开始计数、通过 GTIOC3B 引脚输入停止或者清除计数）



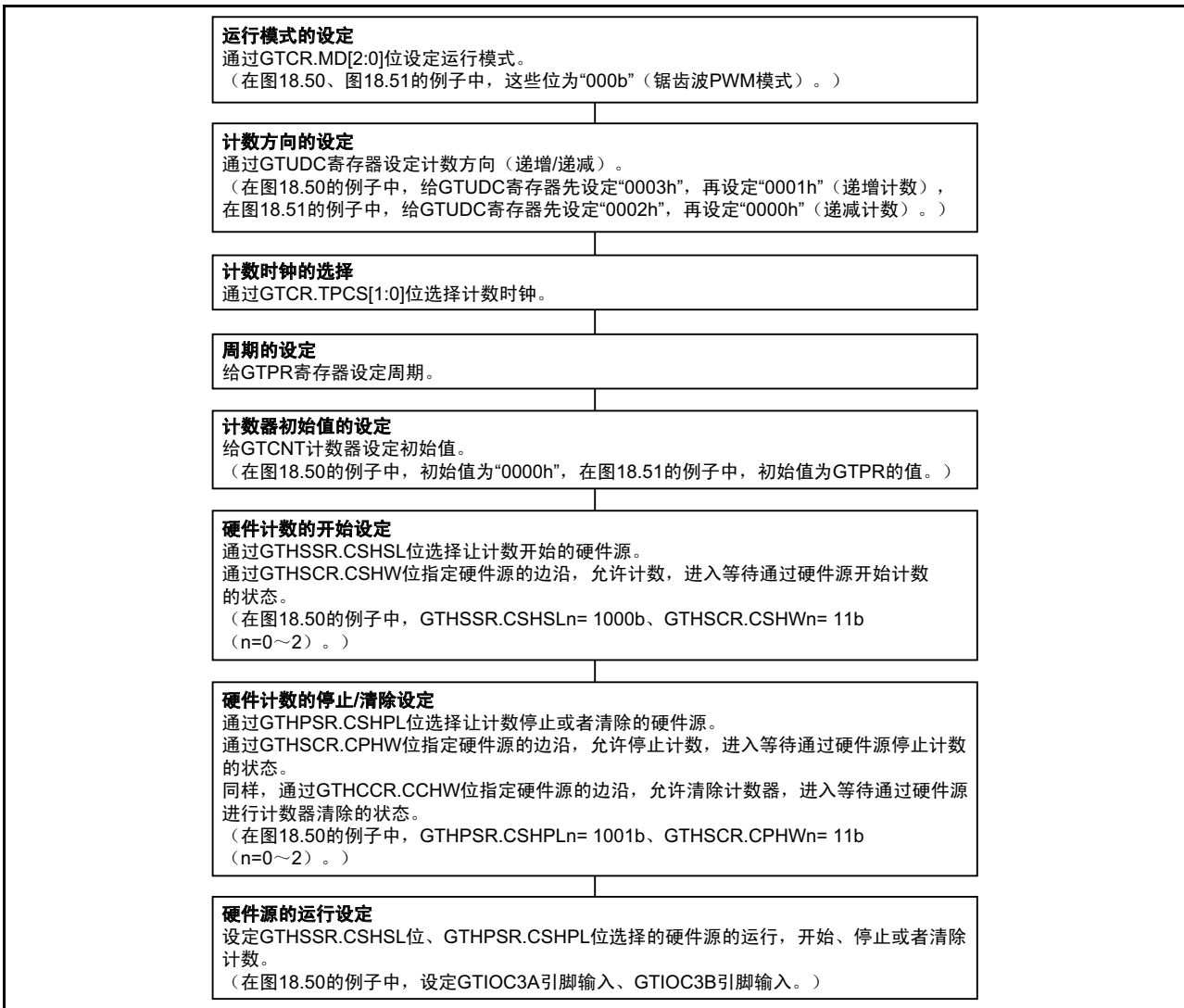


图 18.52 通过硬件源进行计数器清除运行的设定例子

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 18.53 所示。

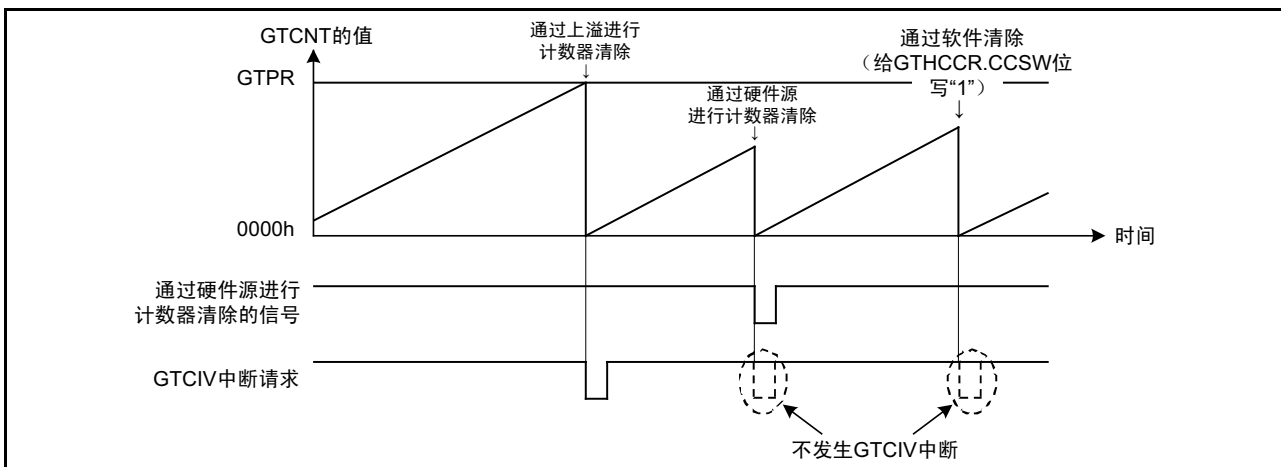


图 18.53 通过硬件进行计数器清除和 GTCIV 中断的关系

### 18.3.7 同步运行

能进行通道间的同步运行（同步清除、同步启动）。

#### 18.3.7.1 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位的  $GTCR.CCLR[1:0]$  位置“11b”，然后通过  $GTSYNC.SYNCn[1:0]$  ( $n=0 \sim 3$ ) 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 18.54、图 18.56 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 18.55、图 18.56 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  $GPT2.GTCNT$ 。

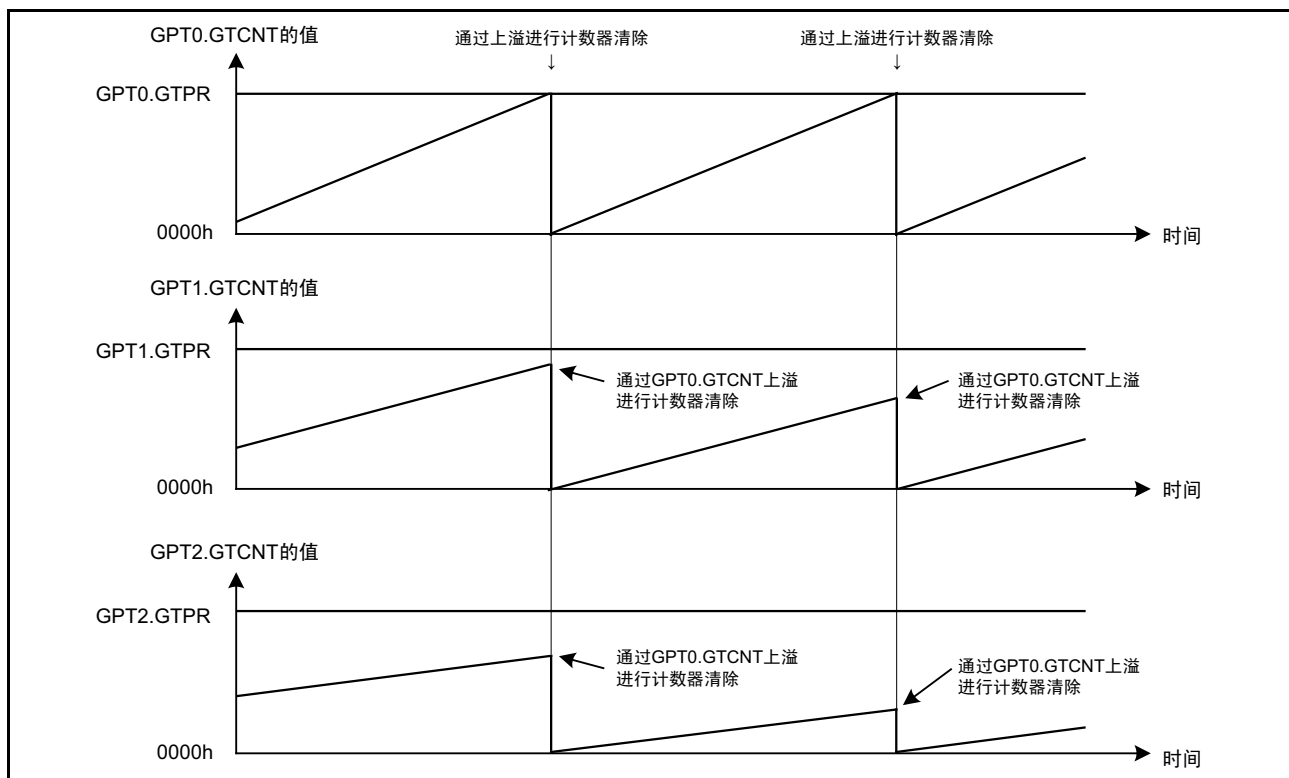


图 18.54 同步清除的运行例子（通过  $GPT0.GTCNT$  的清除源同步清除  $GPT1.GTCNT$  和  $GPT2.GTCNT$ ）

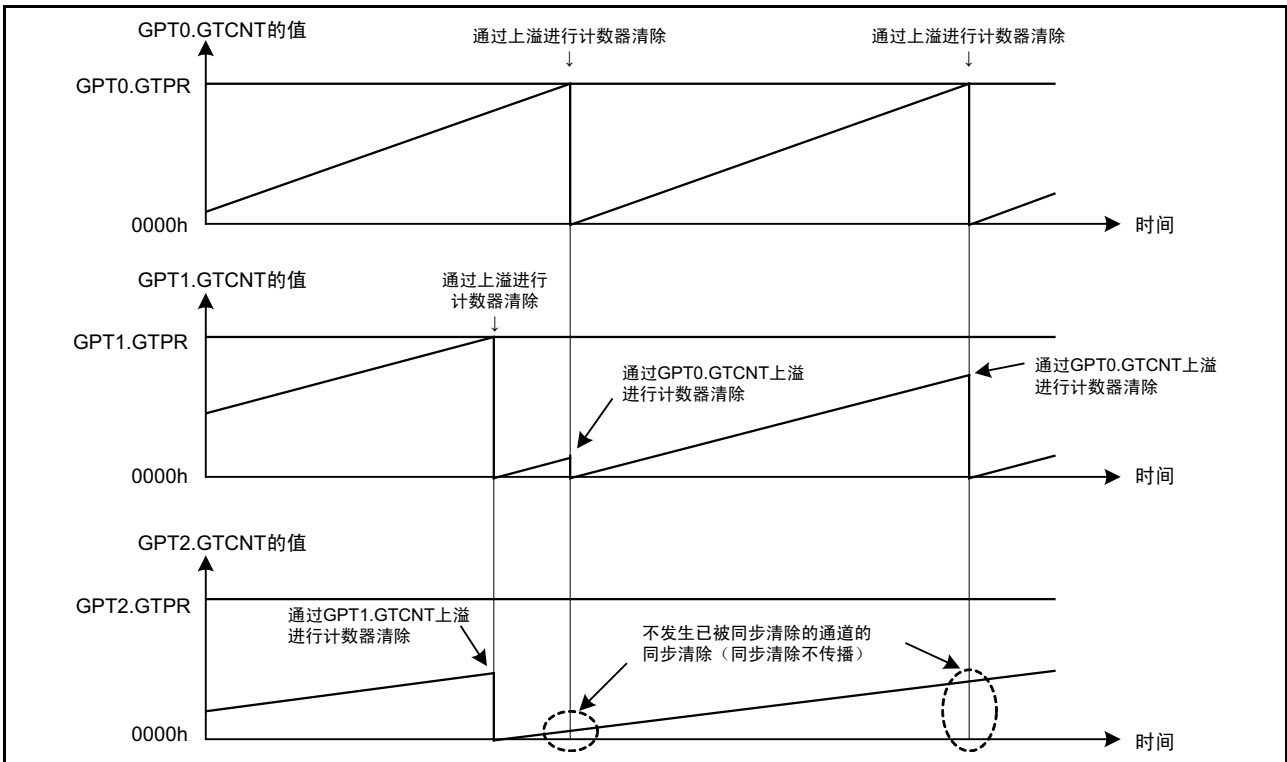


图 18.55 同步清除的运行例子  
 (通过 GPT0.GTCNT 的清除源同步清除 GPT1.GTCNT、通过 GPT1.GTCNT 的清除源同步清除 GPT2.GTCNT)

<p><b>运行模式的设定</b>                  通过各通道的GTCR.MD[2:0]位设定运行模式。                  (在图18.54、图18.55的例子中, 这些位为“000b”(锯齿波PWM模式)。)</p>
<p><b>计数方向的设定</b>                  通过各通道的GTUDC寄存器设定计数方向(递增/递减)。                  (在图18.54、图18.55的例子中, 给GTUDC寄存器先设定“0003h”。再设定“0001h”(递增计数)。)</p>
<p><b>计数时钟的选择</b>                  通过各通道的GTCR.TPCS[1:0]位选择计数时钟。</p>
<p><b>周期的设定</b>                  给各通道的GTPR寄存器设定周期。</p>
<p><b>计数器初始值的设定</b>                  给各通道的GTCNT计数器设定初始值。</p>
<p><b>同步计数器清除的设定</b>                  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18.54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18.55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><b>计数开始</b>                  将GTSTR.CST 位置“1”, 开始计数。</p>

图 18.56 同步清除运行的设定例子

## 18.3.7.2 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 18.57 所示。

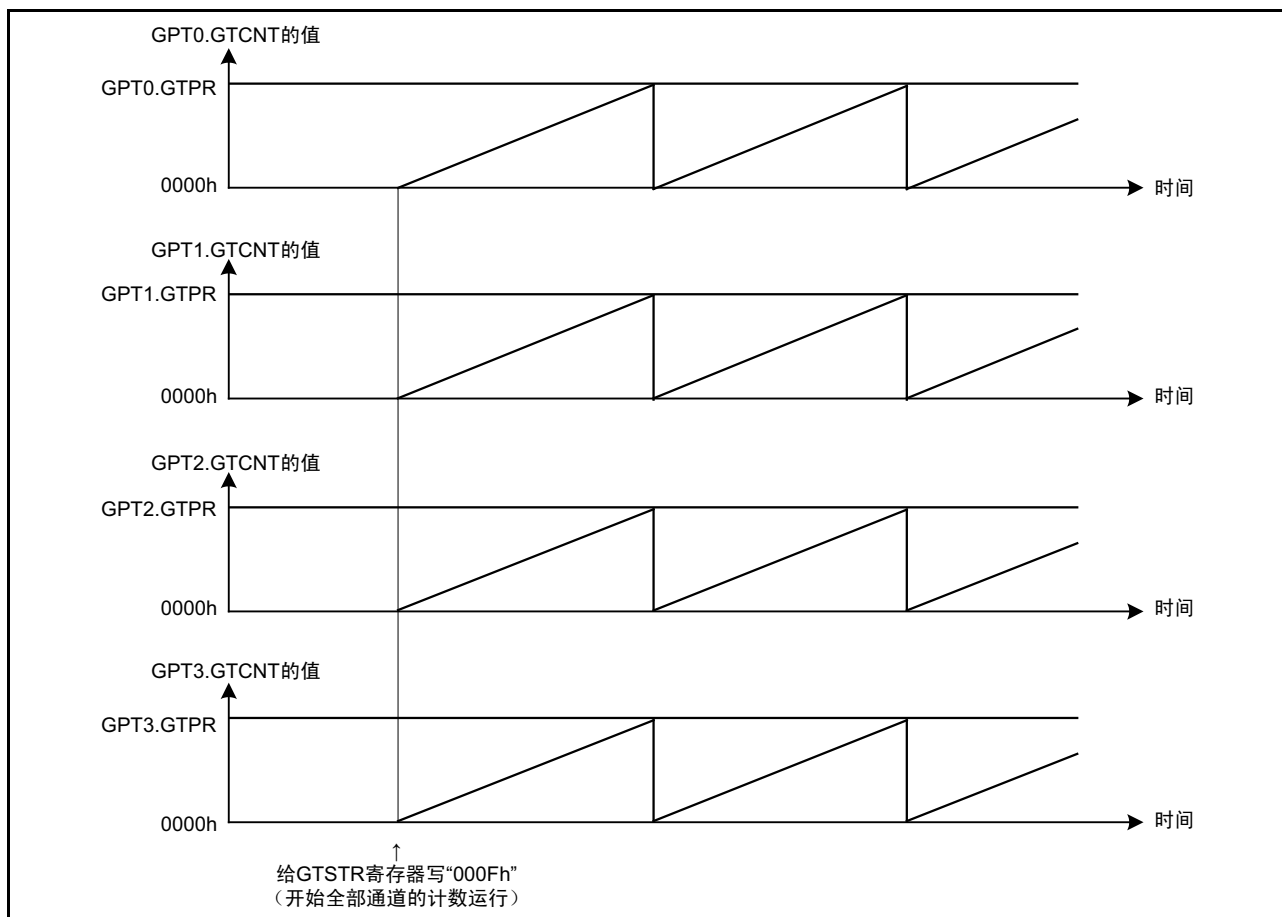


图 18.57 通过软件同时启动的运行例子 (计数周期 (GTPR 的值) 相同时)

(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 18.58 所示。

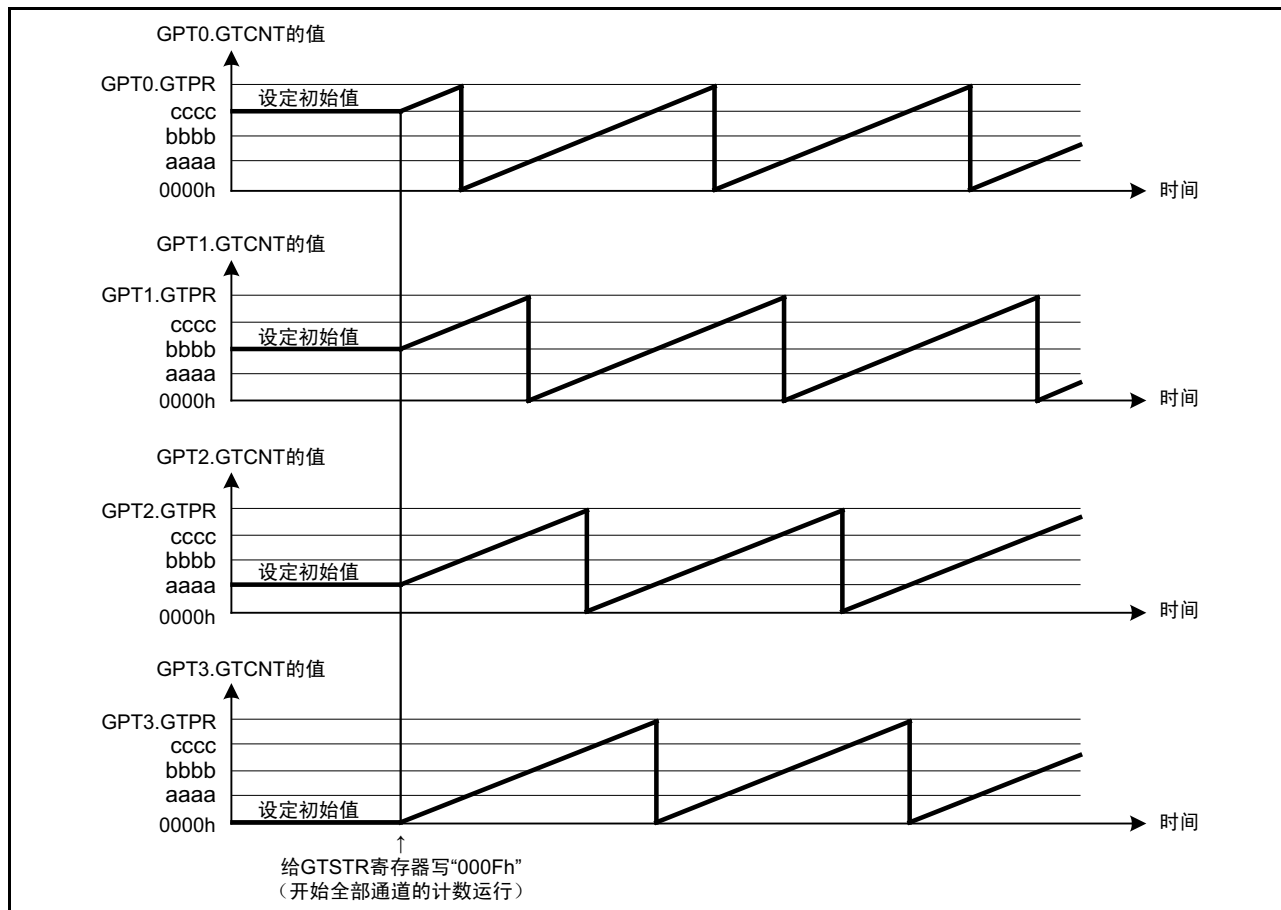


图 18.58 通过软件进行相位启动的运行例子 (计数周期 (GTPR 的值) 相同时)

(3) 通过硬件源同时启动

能通过 RX62T 内部的硬件源，同时开始各通道的计数运行。

硬件源有 4 种，分别是 GTETRG 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 18.59、图 18.60 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。

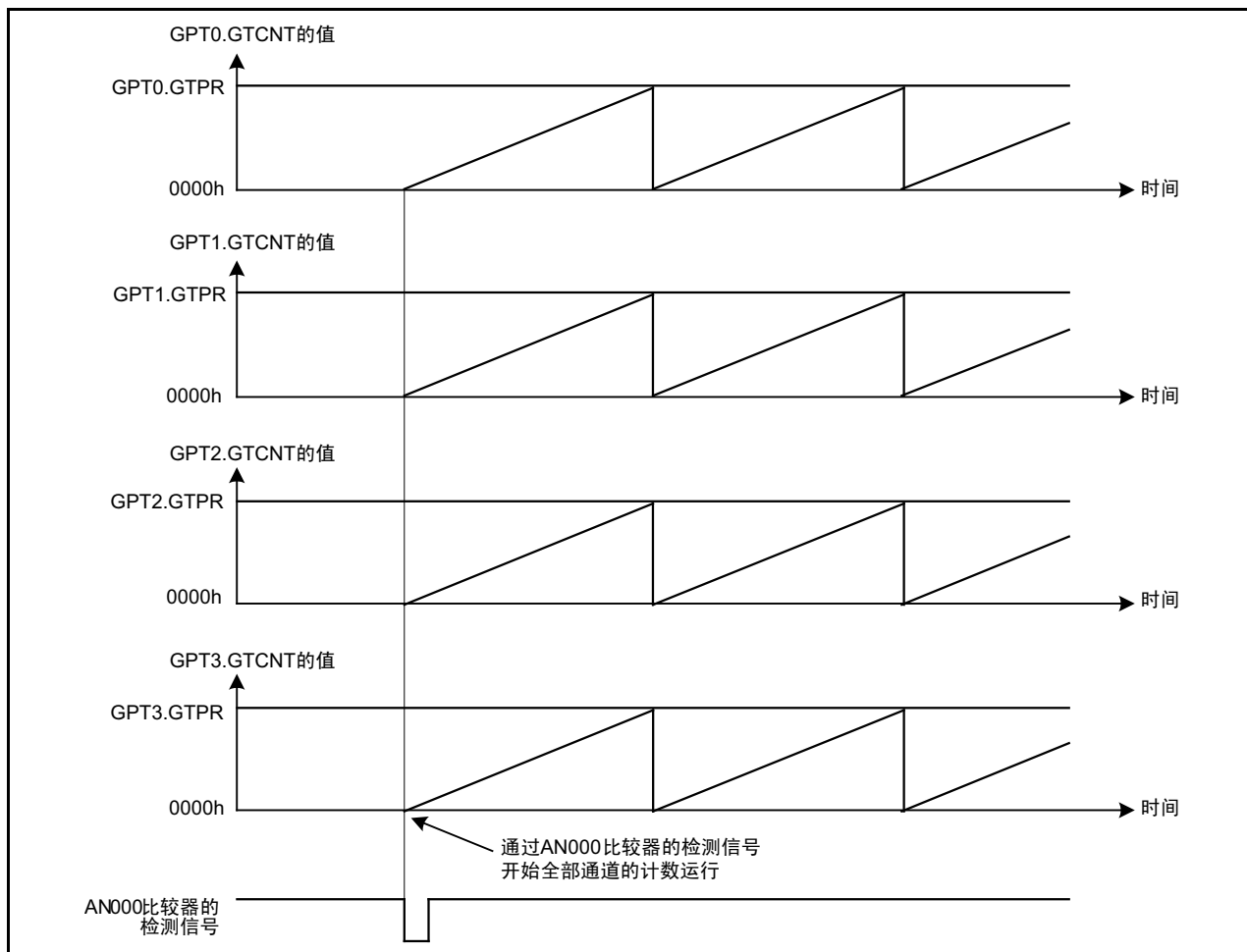


图 18.59 通过硬件源同步启动的运行例子（计数周期（GTPR 的值）相同时）

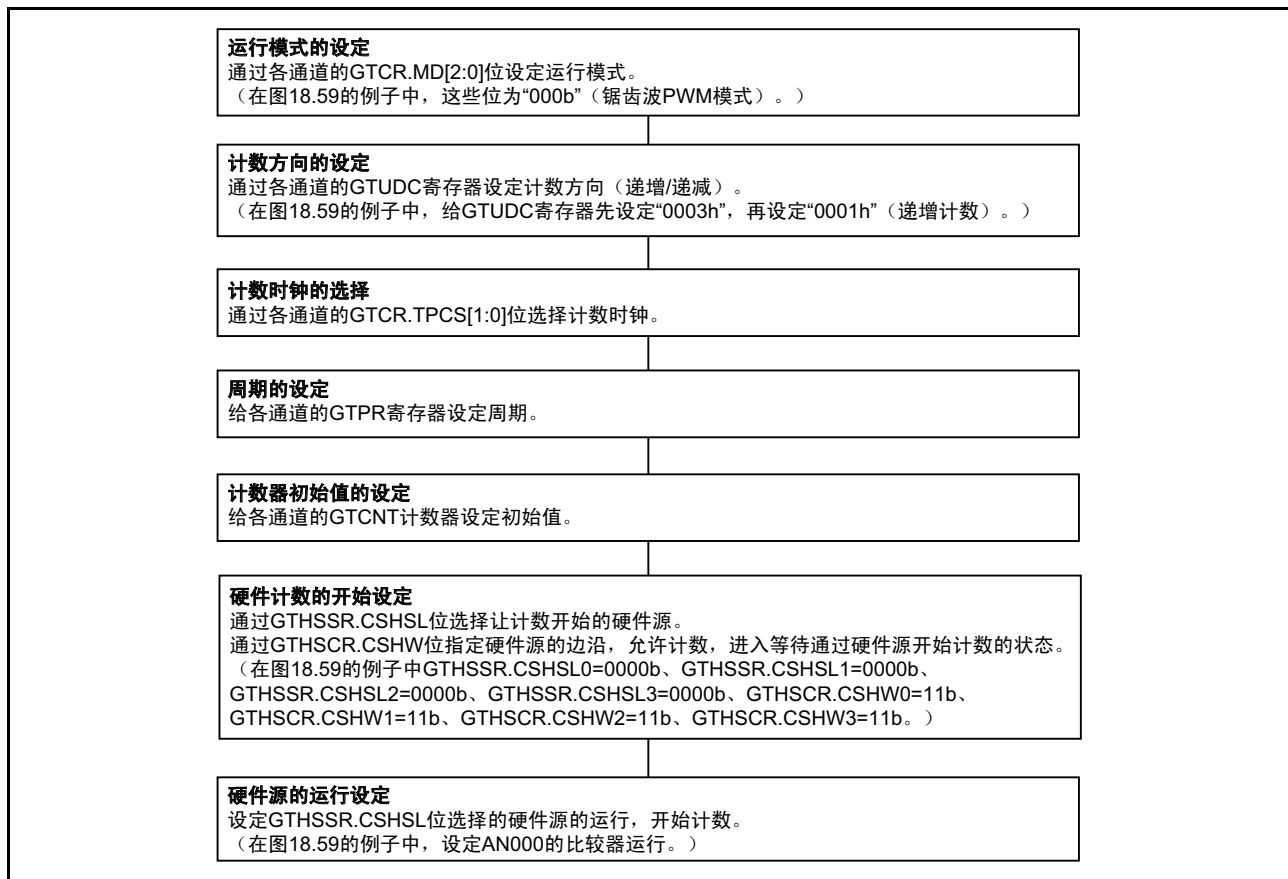


图 18.60 通过硬件源同步启动的运行设定例子

#### (4) 通过硬件源进行相位启动

能通过 RX62T 内部的硬件源, 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 18.61、图 18.62 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。

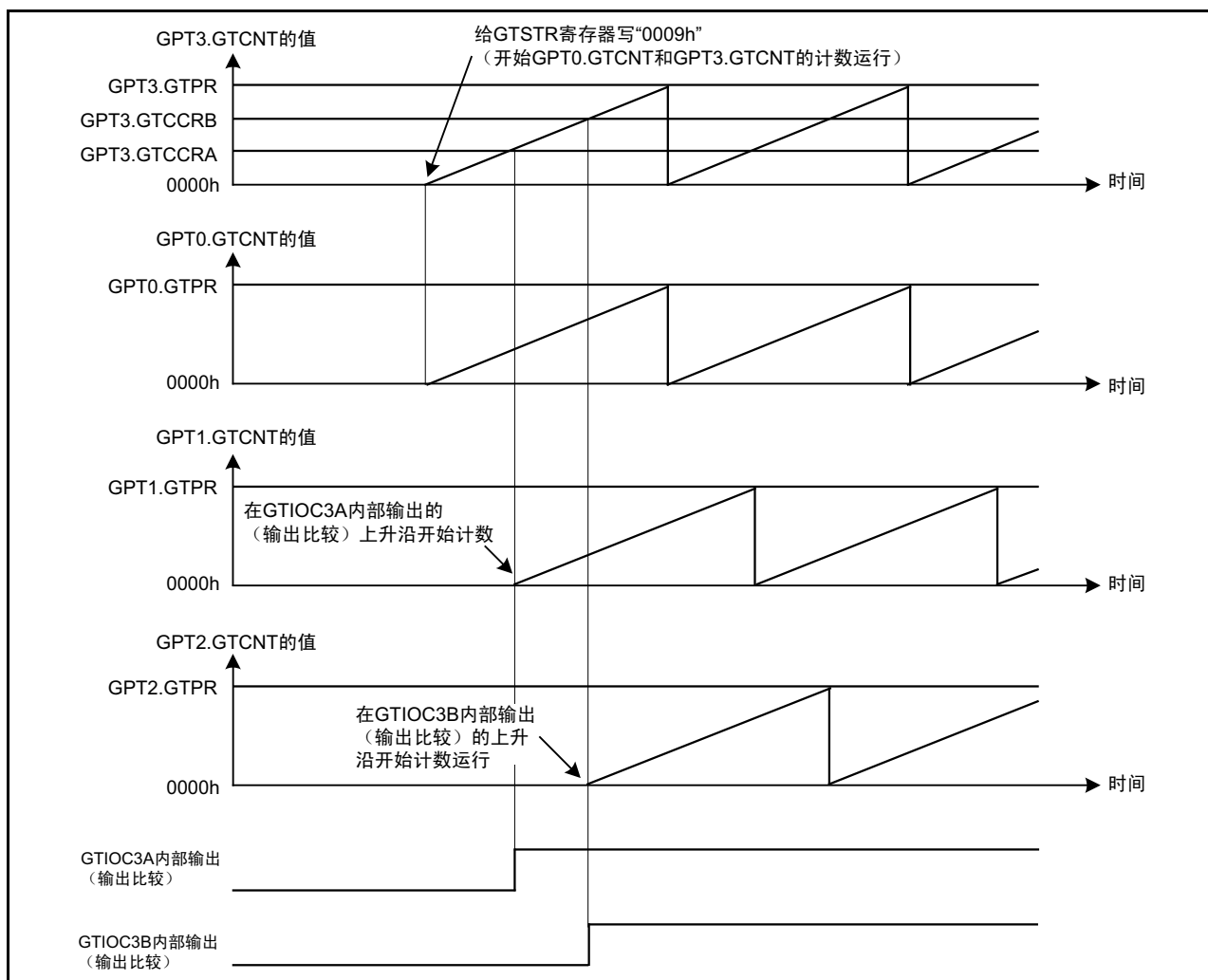


图 18.61 通过硬件源进行相位启动的运行例子 (计数周期 (GTPR 的值) 相同时)



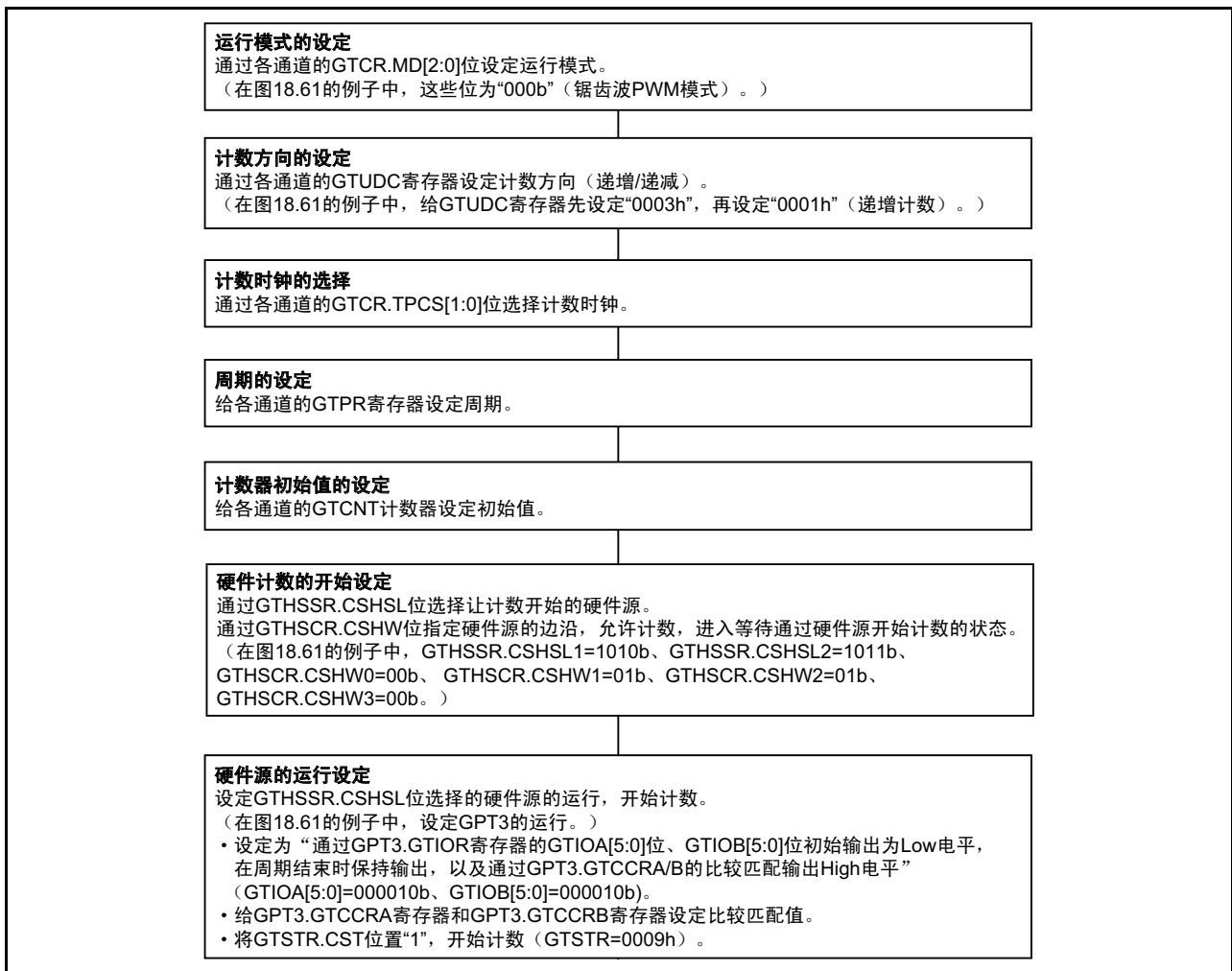


图 18.62 通过硬件源进行相位启动运行的设定例子

### 18.3.8 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 18.63 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。

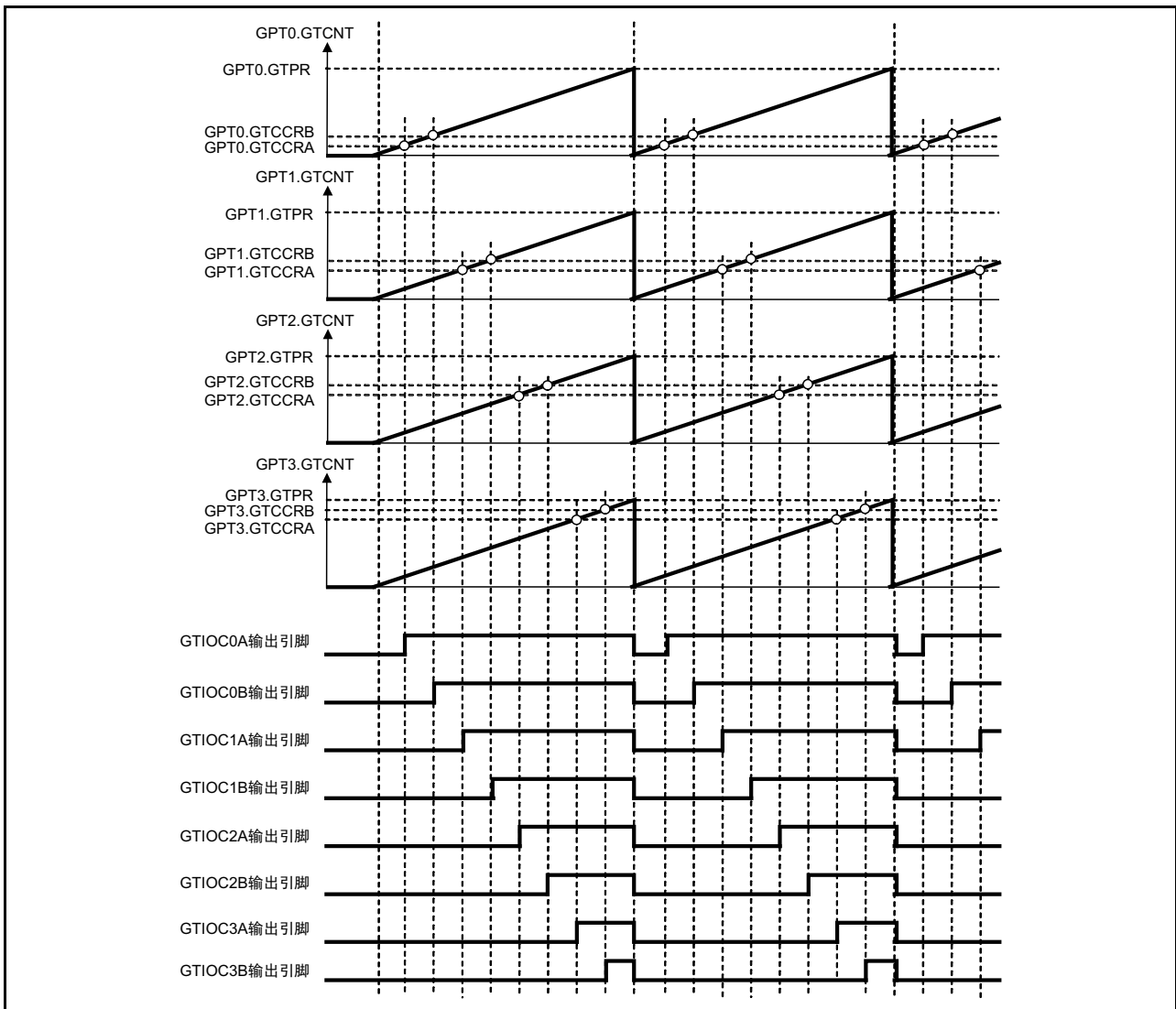


图 18.63 同步 PWM 输出的例子

## (2) 锯齿波三相互补 PWM 输出

图 18.64 是在锯齿波 PWM 模式中让 3 个通道进行同步运行，并且输出三相互补 PWM 波形的例子。在此例子中如下设定 GTIOcN<sub>A</sub> 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<sub>B</sub> 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。

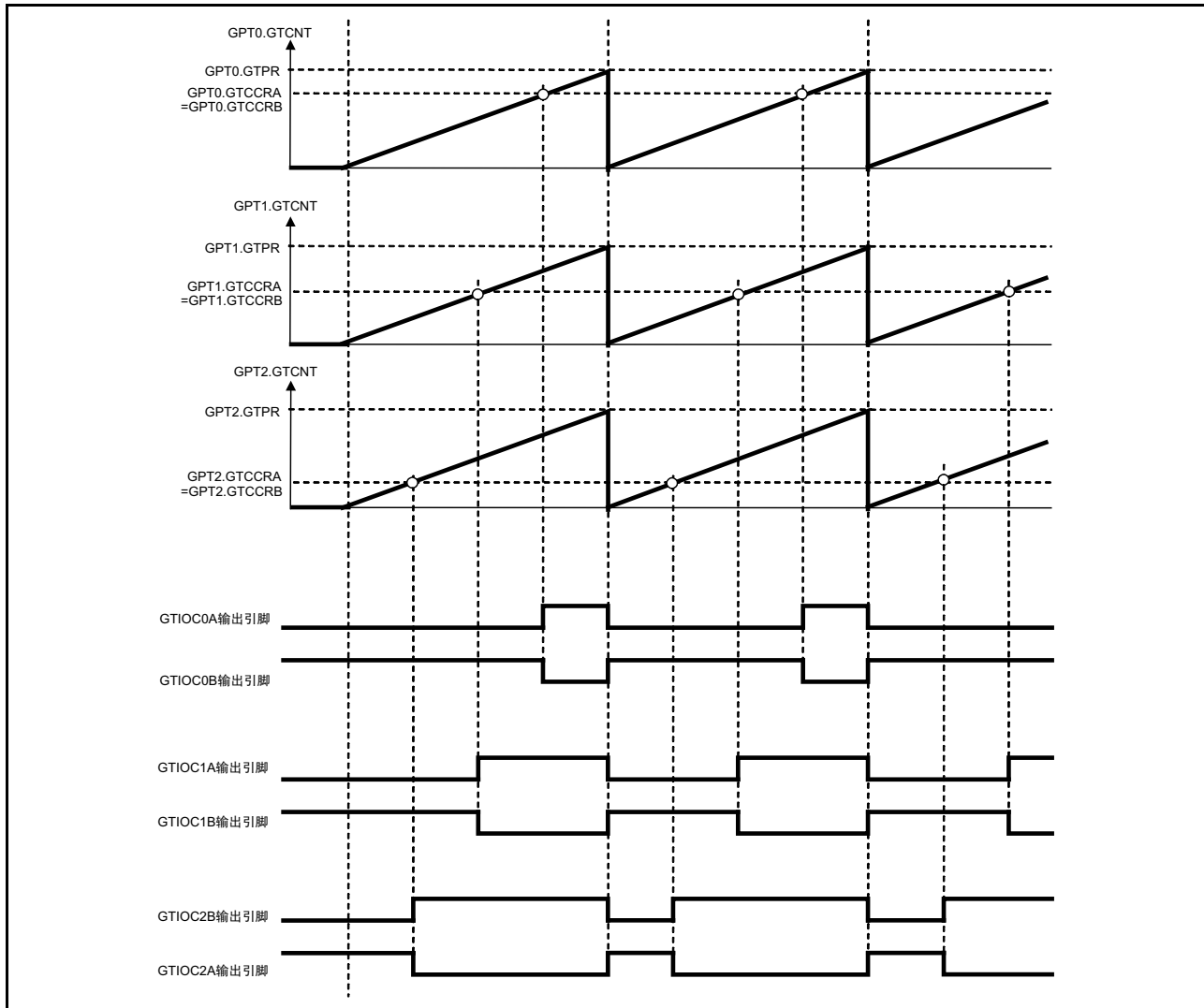


图 18.64 锯齿波三相互补 PWM 输出的例子

(3) 锯齿波三相互补 PWM 输出 (自动设定死区时间)

图 18.65 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，并且输出三相互补 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。

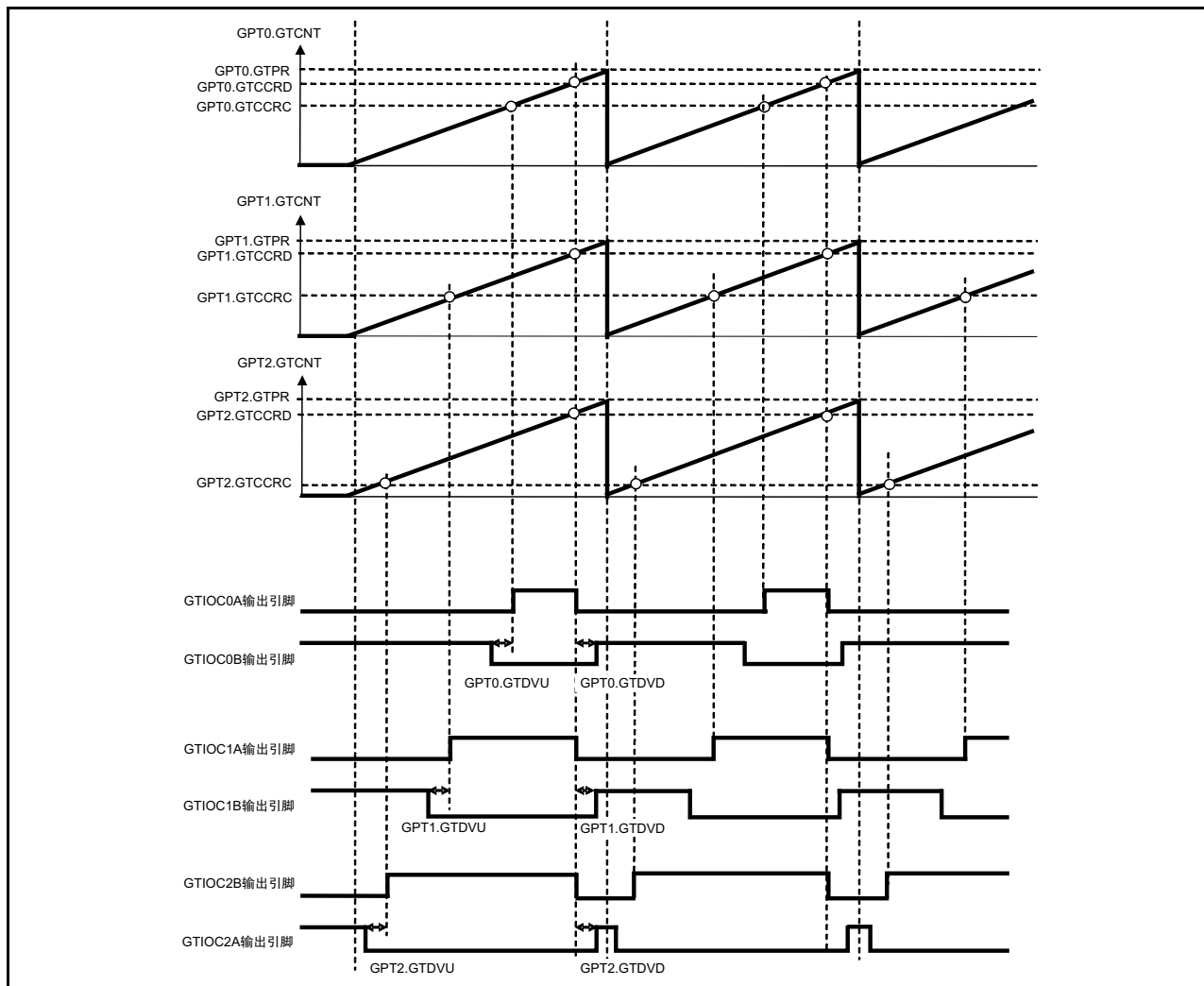


图 18.65 锯齿波三相互补 PWM 输出的例子 (自动设定死区时间)

## (4) 三角波三相互补 PWM 输出

图 18.66 是在三角波 PWM 模式 1 中让 3 个通道进行同步运行，并且输出三相互补 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。

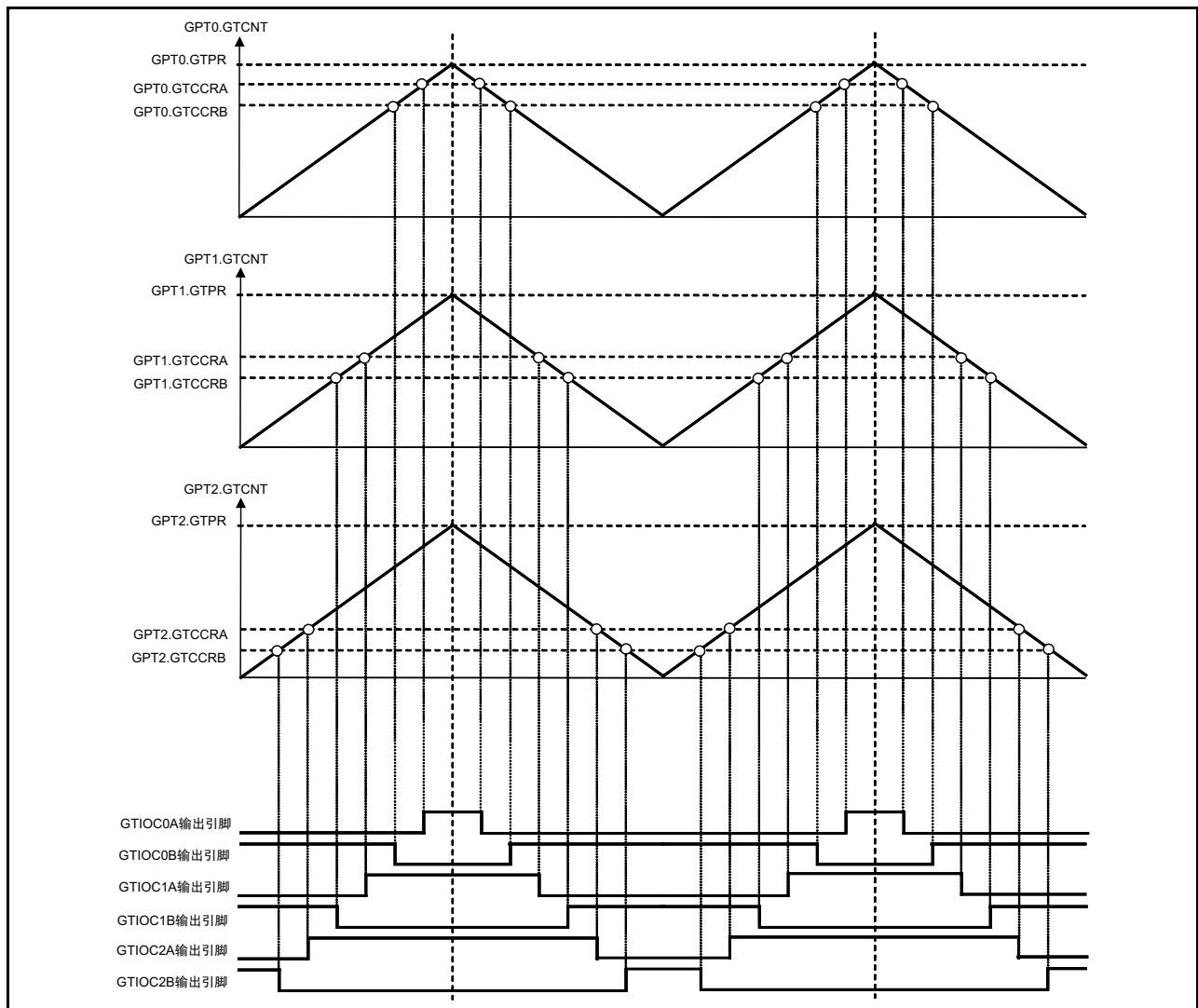


图 18.66 三角波三相互补 PWM 输出的例子

## (5) 三角波三相互补 PWM 输出 (自动设定死区时间)

图 18.67 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，并且输出三相互补 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。

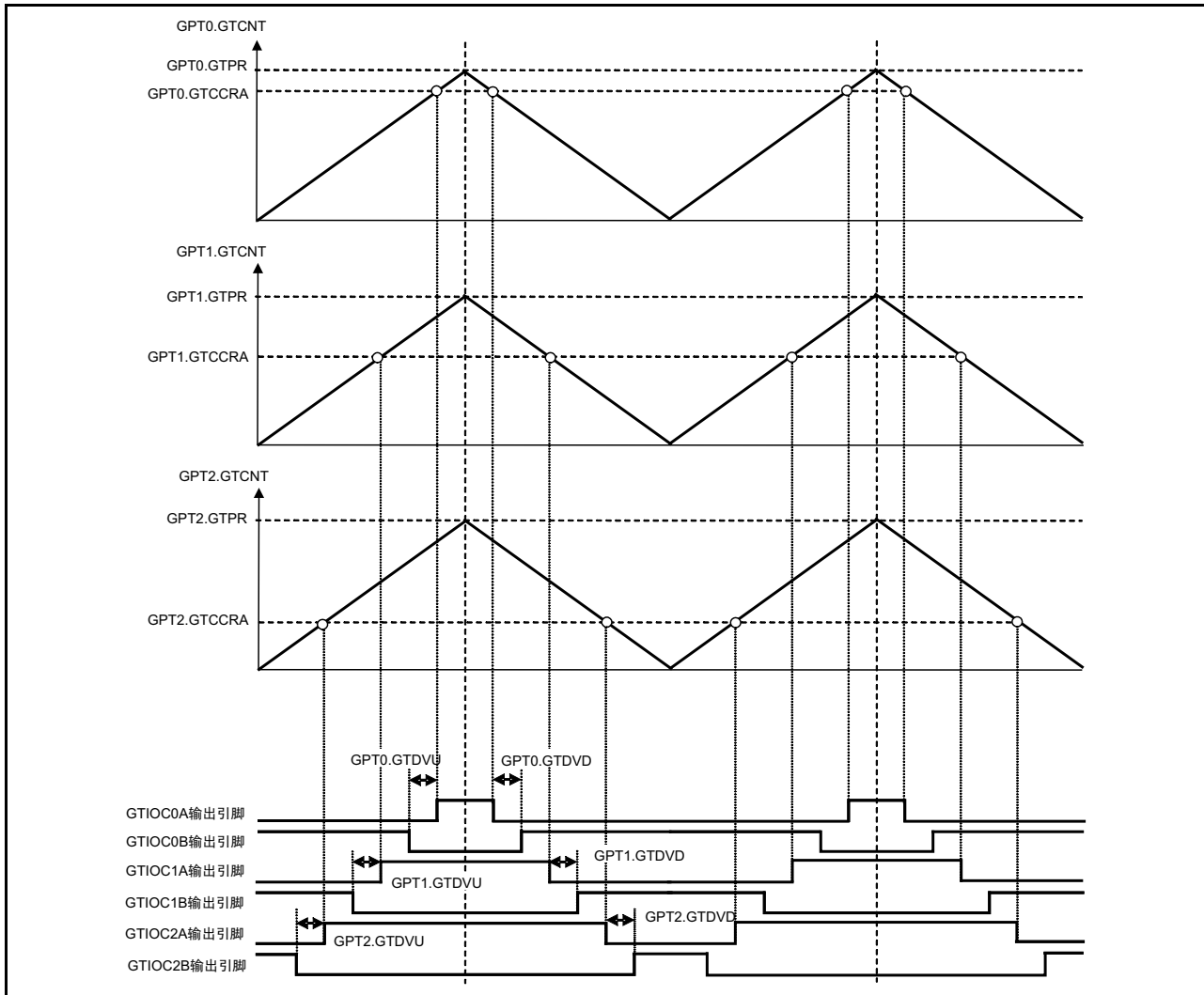


图 18.67 三角波三相互补 PWM 输出的例子 (自动设定死区时间)

(6) 非对称三角波三相互补 PWM 输出 (自动设定死区时间)

图 18.68 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，并且输出三相互补 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。

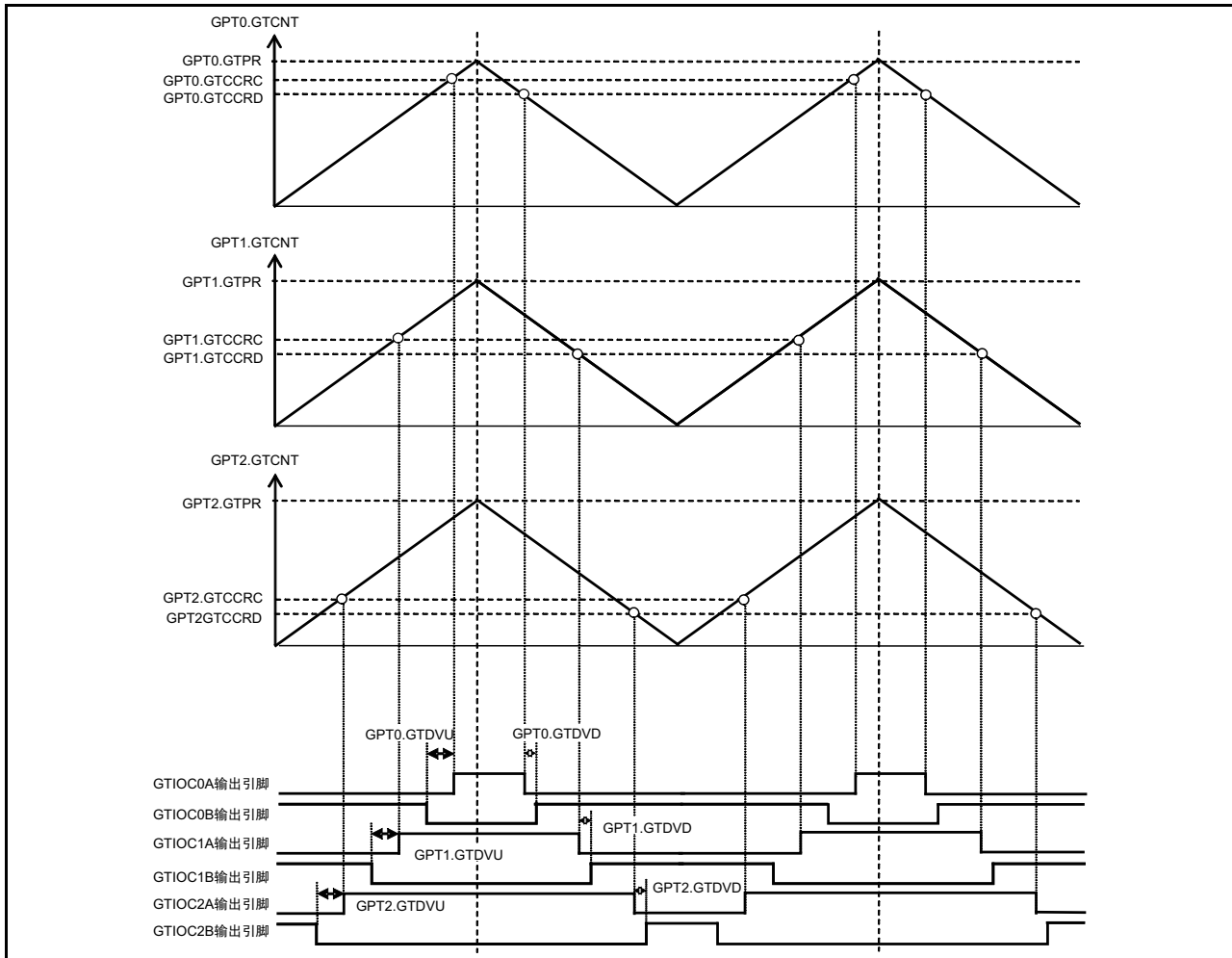


图 18.68 非对称三角波形三相互补 PWM 输出的例子 (自动设定死区时间)

## 18.4 中断源

### 18.4.1 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、LOCO 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，所以为了重新发送中断，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“11. 中断控制器 (ICU)”，GPT 中断源的一览表如表 18.6 所示。



表 18.6 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	LOCO 计数功能中断	LCNT 的上溢	LISO		能
			LOCO 计数值偏差超	LISD		
			LOCO 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) LOCO 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 LOCO 分频时钟的上升沿检测、LOCO 计数值偏差超或者 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 18.7 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<sub>n</sub>、GTCIV<sub>n</sub>、LOCOI (n=0、1、2、3) 中，在因某个中断源（例如，GTCIC<sub>n</sub> 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<sub>n</sub> 的 TCFC 标志）重叠的另一个中断源（GTCIC<sub>n</sub> 的 TCFD 标志）的中断请求，也忽视该中断源 GTCIC<sub>n</sub> 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。

## 18.4.2 DTC 的启动

能通过各通道的中断启动 DTC。详细内容请参照“11. 中断控制器 (ICU)”、“14. 数据传送控制器 (DTC)”。

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”。

## 18.4.3 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 18.69 ~ 图 18.74 所示。

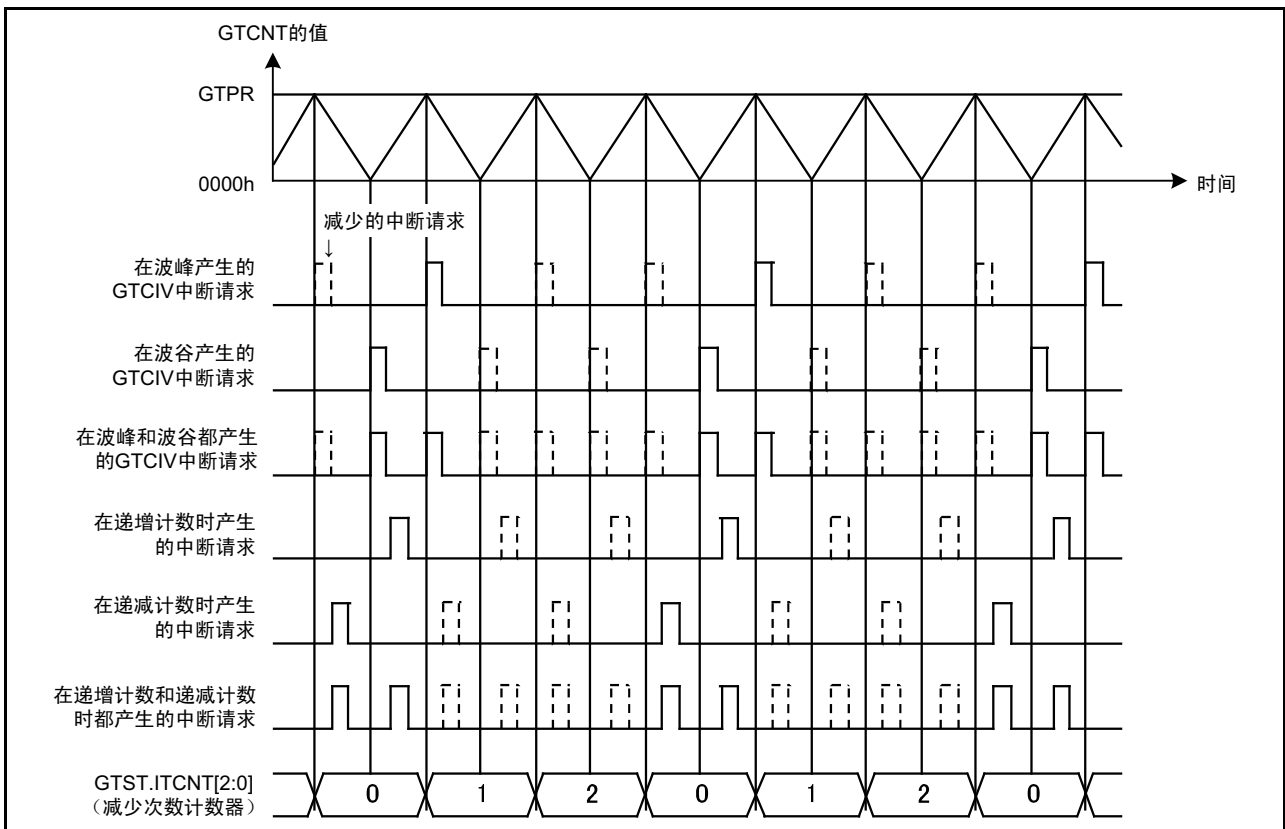


图 18.69 中断减少功能的运行例子 (三角波、对波峰进行计数并且减少功能有效、减少次数为 2 次)

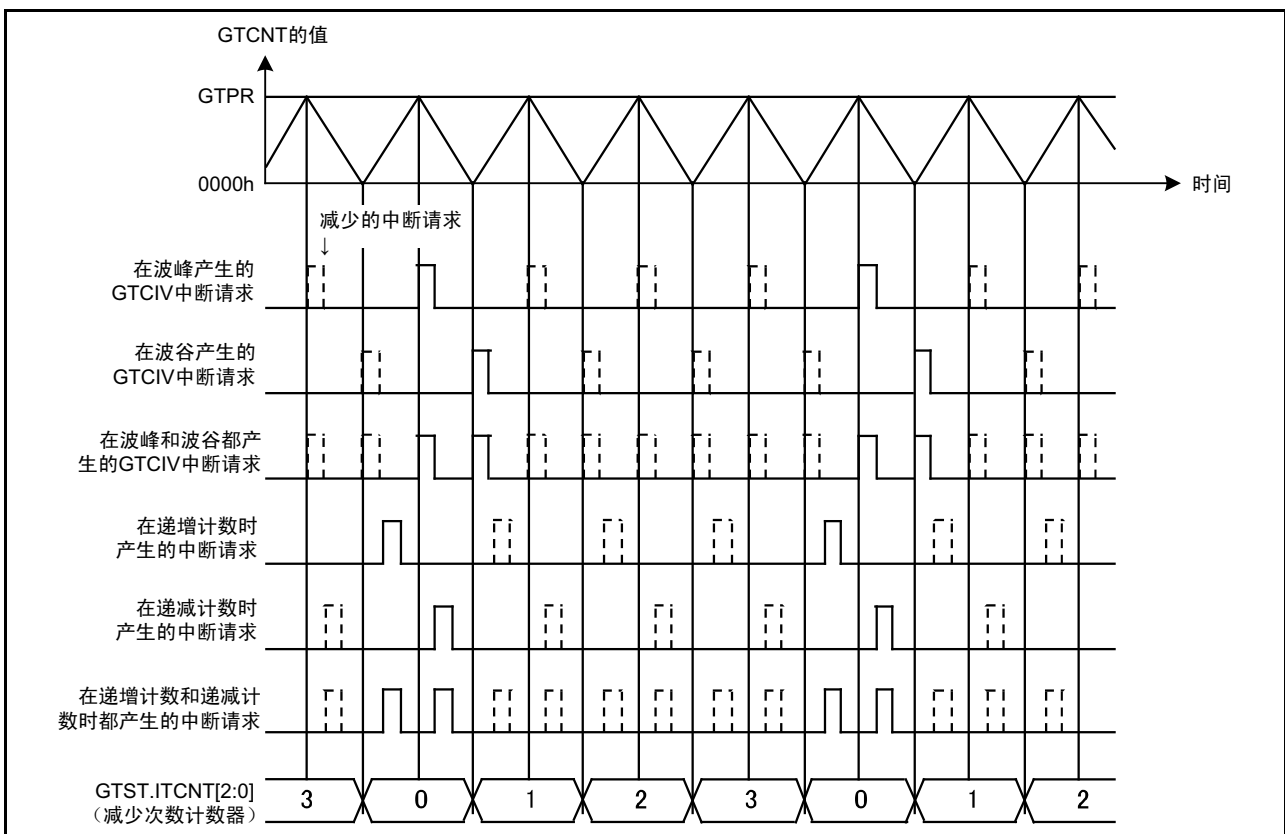


图 18.70 中断减少功能的运行例子 (三角波、对波谷进行计数并且减少功能有效、减少次数为 3 次)

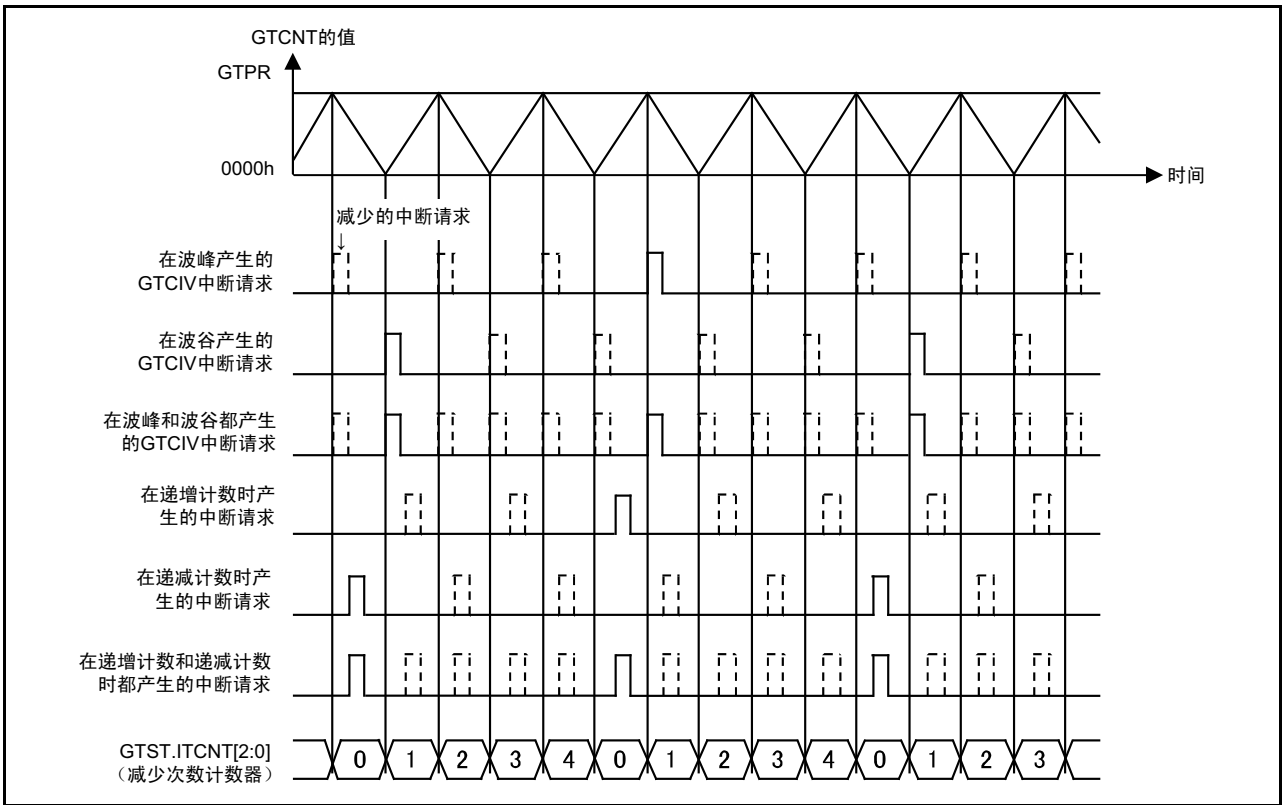


图 18.71 中断减少功能的运行例子  
(三角波、对波谷和波峰进行计数并且减少功能有效、减少次数为 4 次)

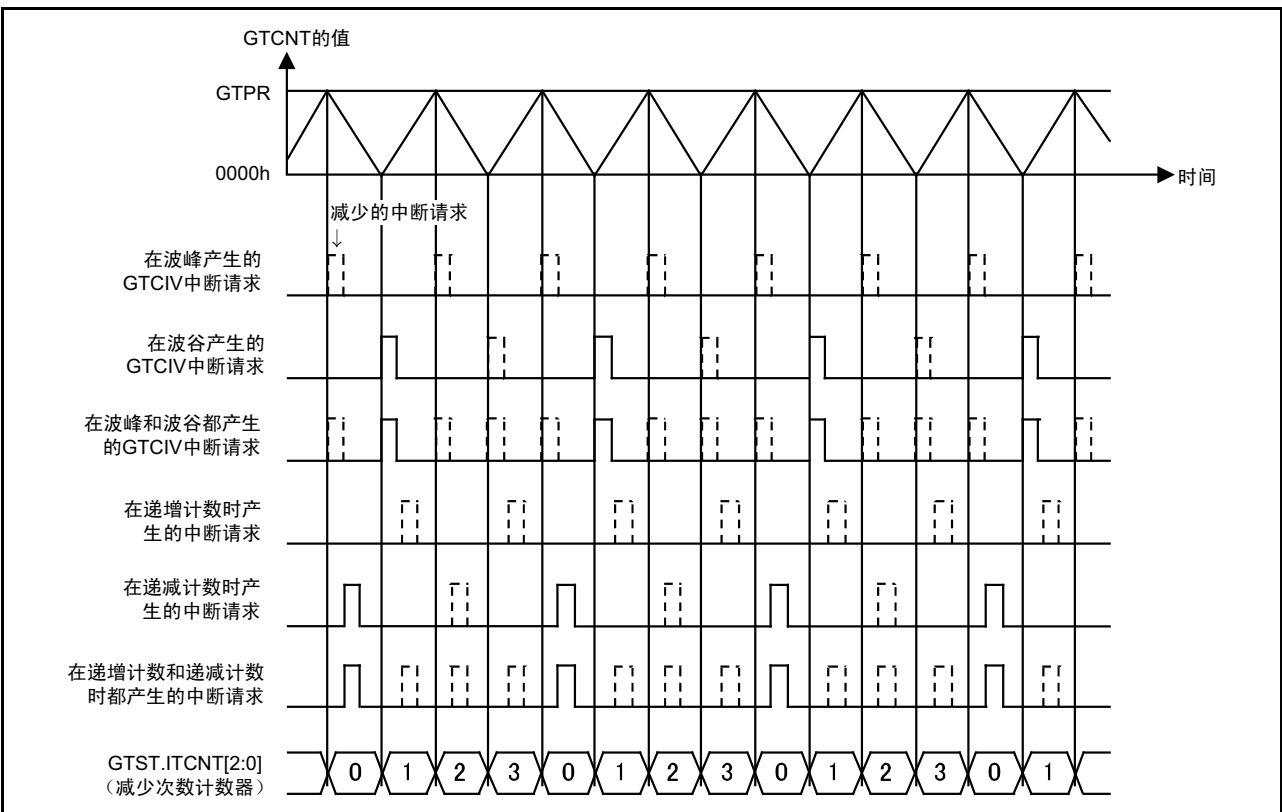


图 18.72 中断减少功能的运行例子  
(三角波、对波谷和波峰进行计数并且减少功能有效、减少次数为 3 次、在递增计数时开始减少)

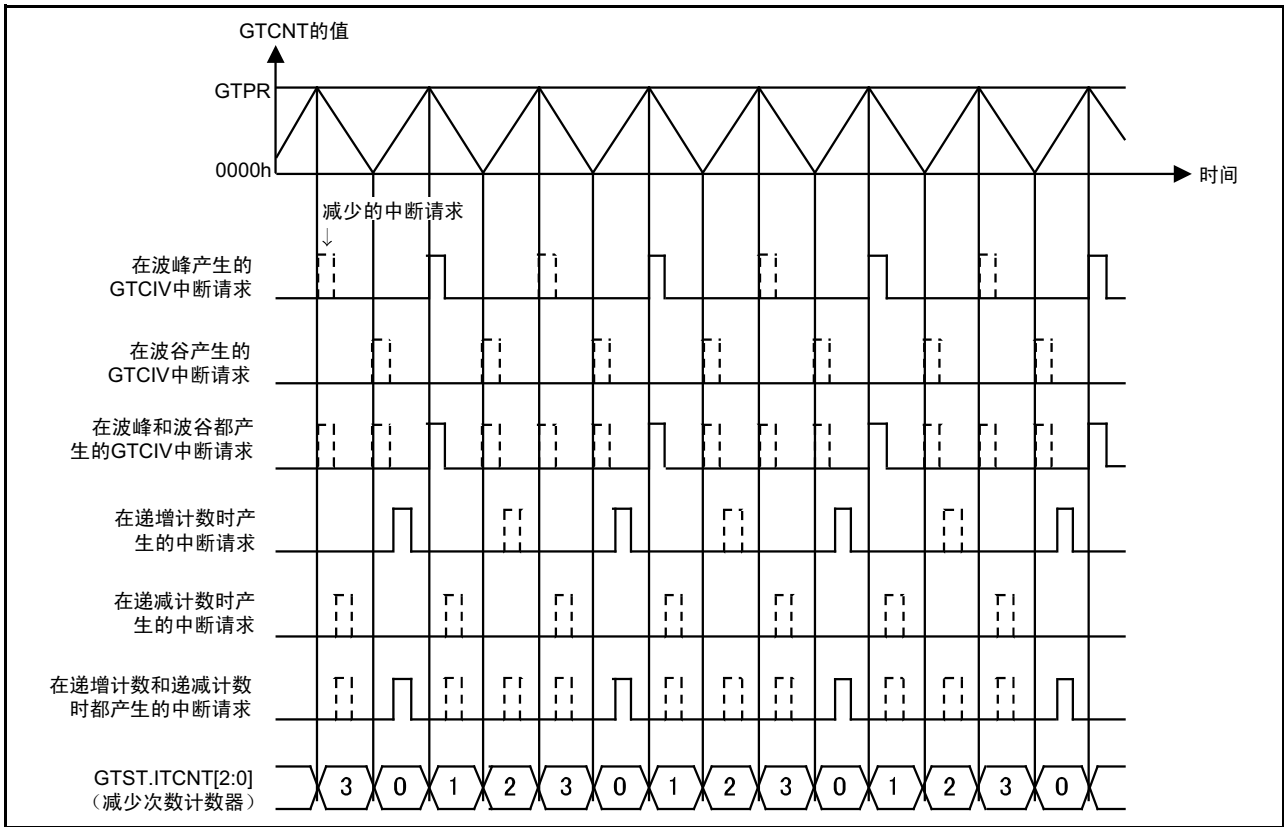


图 18.73 中断减少功能的运行例子  
 (三角波、对波谷和波峰进行计数并且减少功能有效、减少次数为 3 次、在递减计数时开始减少)

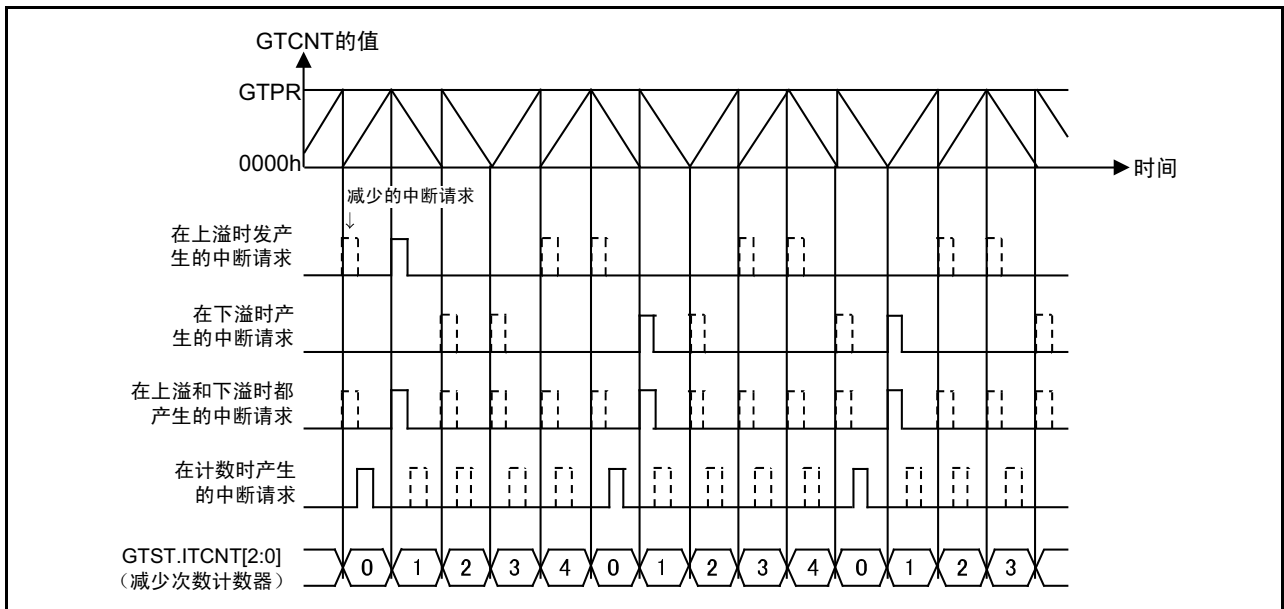


图 18.74 中断减少功能的运行例子  
 (锯齿波、一边转换计数方向一边对上溢和下溢进行计数并且减少功能有效、减少次数为 4 次)

### 18.5 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 18.75，A/D 转换开始请求运行的设定例子如图 18.76 所示。

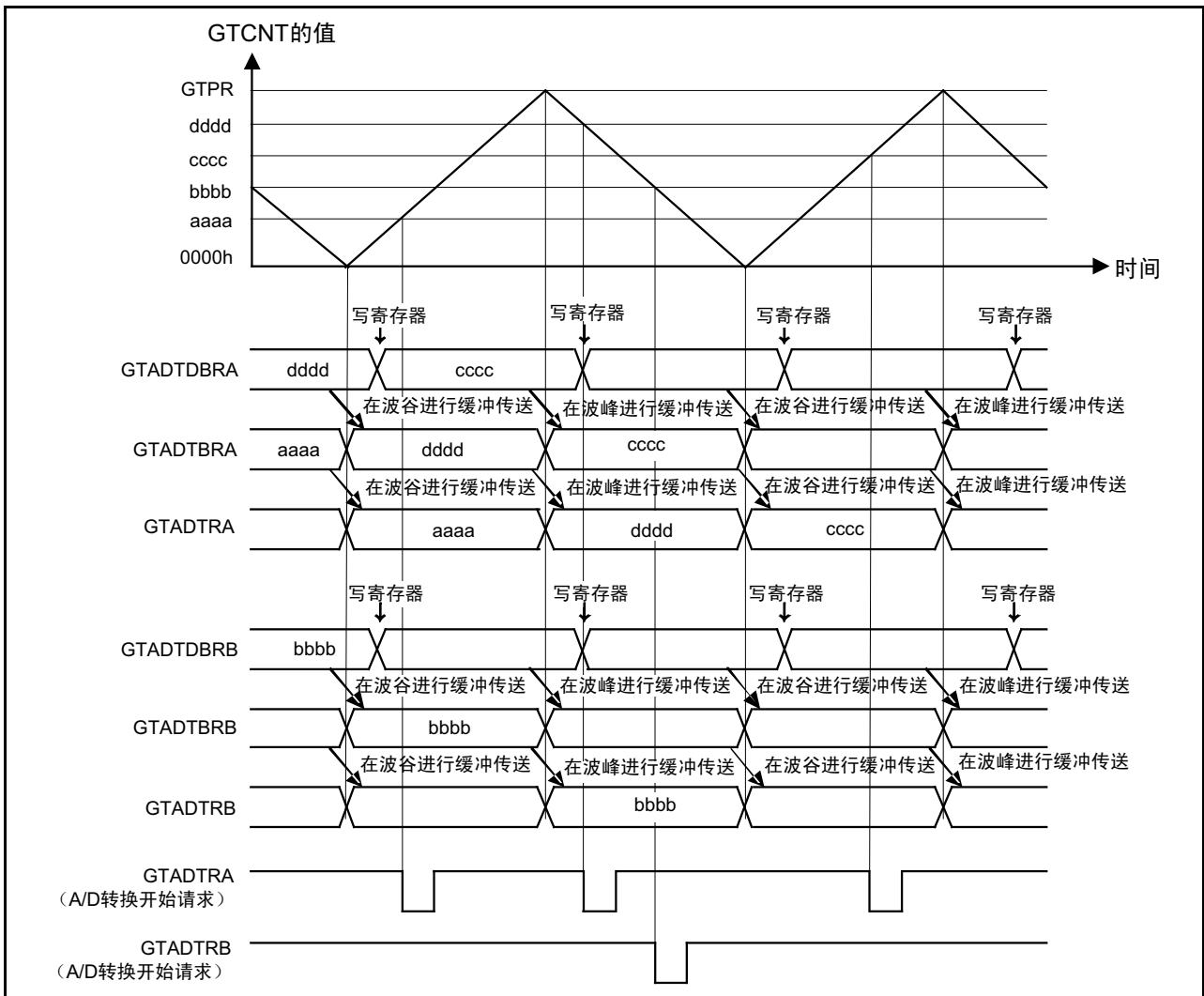


图 18.75 A/D 转换开始请求的运行例子

(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)

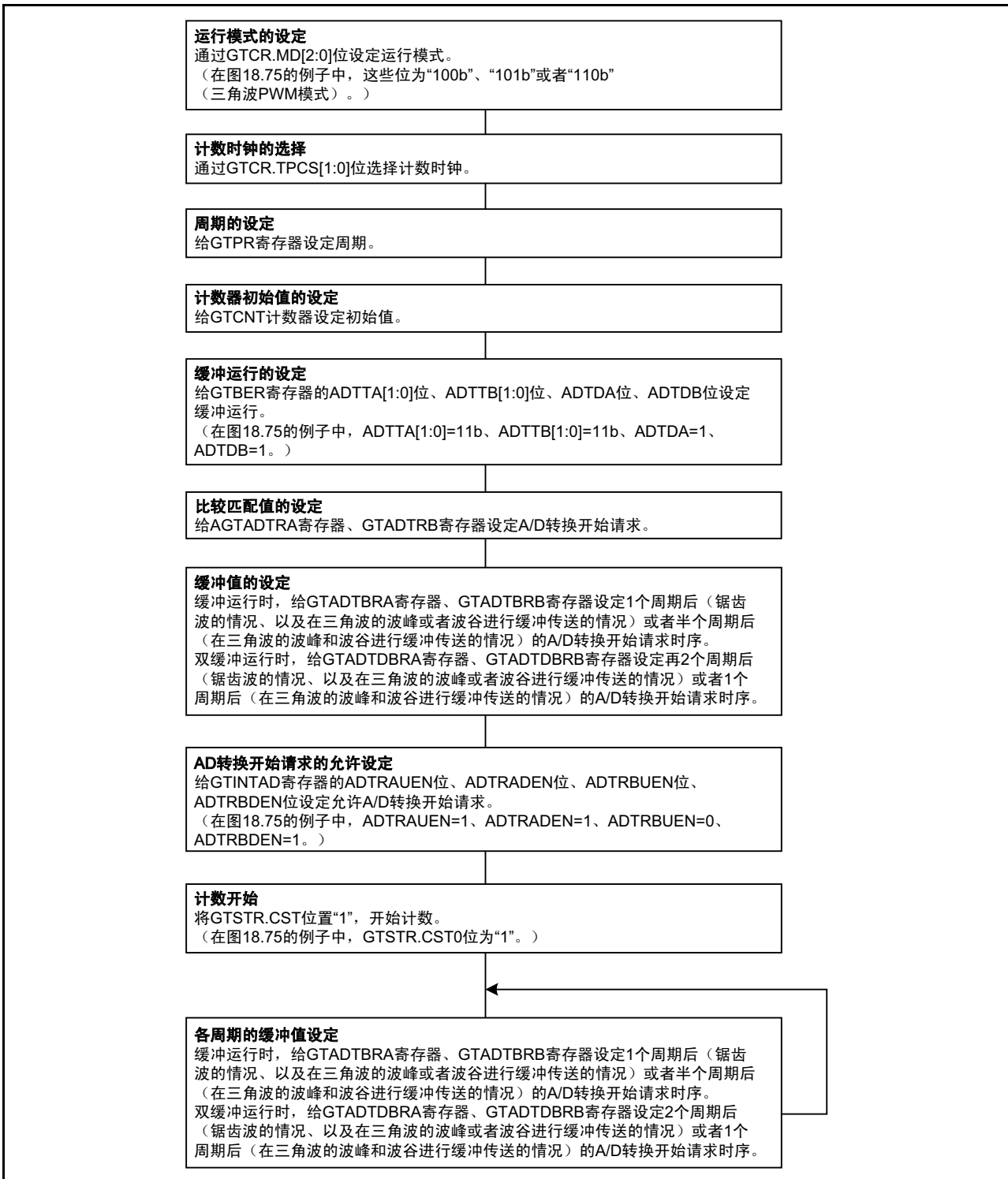


图 18.76 A/D 转换开始请求运行的设定例子



## 18.6 LOCO 计数功能

LOCO 计数功能测量 RX62T 内置的独立看门狗定时器 (IWDT) 的专用低速内部振荡器 (LOCO) 周期。通过此功能还能检测出主时钟振荡频率的异常。

能从 LOCO 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 LOCO 时钟 (LOCO 分频时钟)。能从 ICLK 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 LOCO 分频时钟的计数时钟。

LCNT 计数器通过计数时钟对 LOCO 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 LOCO 分频时钟的上升沿时, 产生 LOCO 分频时钟上升沿中断请求。能减少 LOCO 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 LOCO 偏差超中断请求。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 LOCO 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 LOCO 计数功能时, 必须使独立看门狗定时器 (IWDT) 运行。

LOCO 计数功能的周期设定例子如表 18.8 所示。

LOCO 计数功能的运行例子和设定例子分别如图 18.77、图 18.78 所示。

表 18.8 LOCO 计数功能的周期设定例子

LOCO 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 ICLK=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

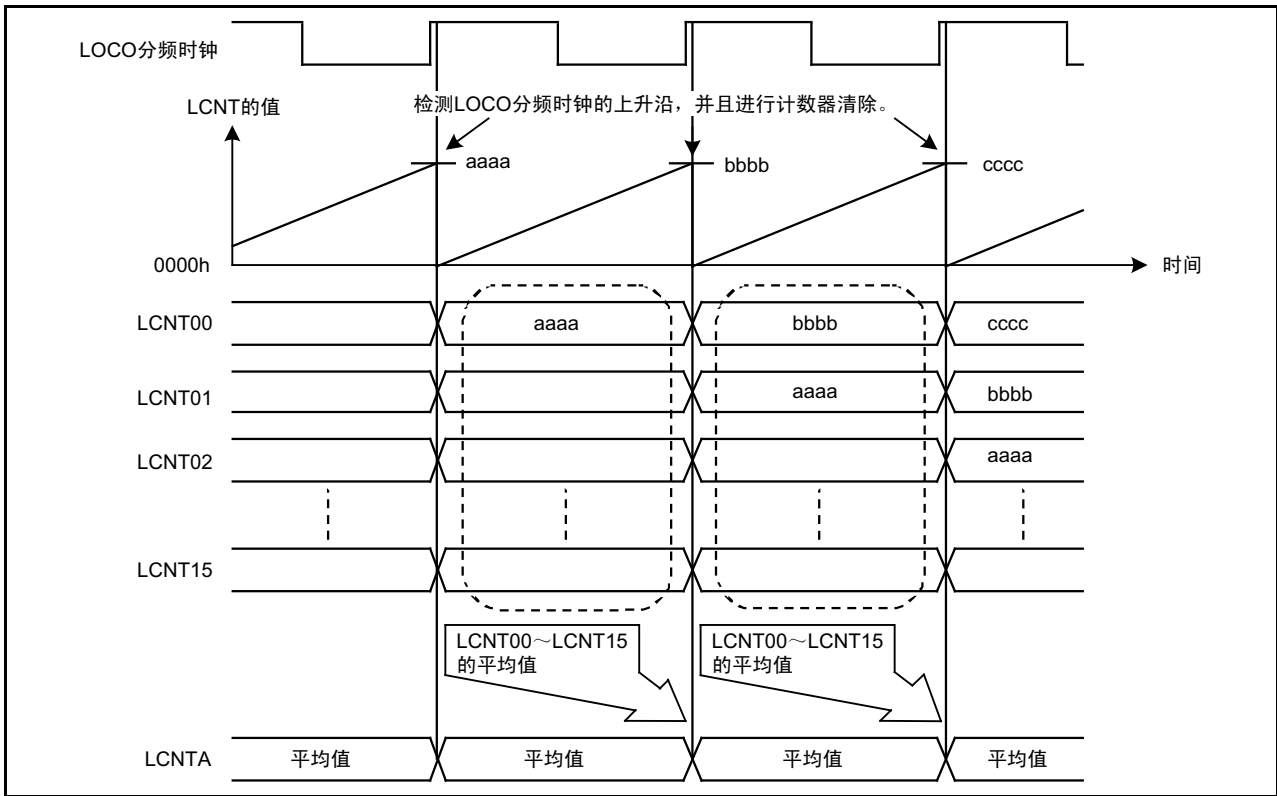


图 18.77 LOCO 计数功能的运行例子

<p><b>计数时钟的选择</b> 通过LCCR.LPSC[1:0]位选择LOCO分频时钟。 通过LCCR.TPSC[1:0]位选择计数时钟。</p>
<p><b>IWDT的设置</b> 使IWDT运行。(详细内容请参照“21. 独立看门狗定时器 (IWDT)”) 给LCCR.LPSC[1:0]位设定“00b”以外的值时， 必须将IWDT.IWDTCCR.CKS[3:0]位设定为“00xx”以外的值。</p>
<p><b>计数结果寄存器的初始化</b> 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><b>计数开始</b> 将LCCR.LCNTE位置“1”，如果检测到LOCO分频时钟的上升沿，就开始计数。</p>

图 18.78 LOCO 计数功能的设定例子

能减少 LOCO 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 18.79、减少计数结果时的运行例子如图 18.80 所示。

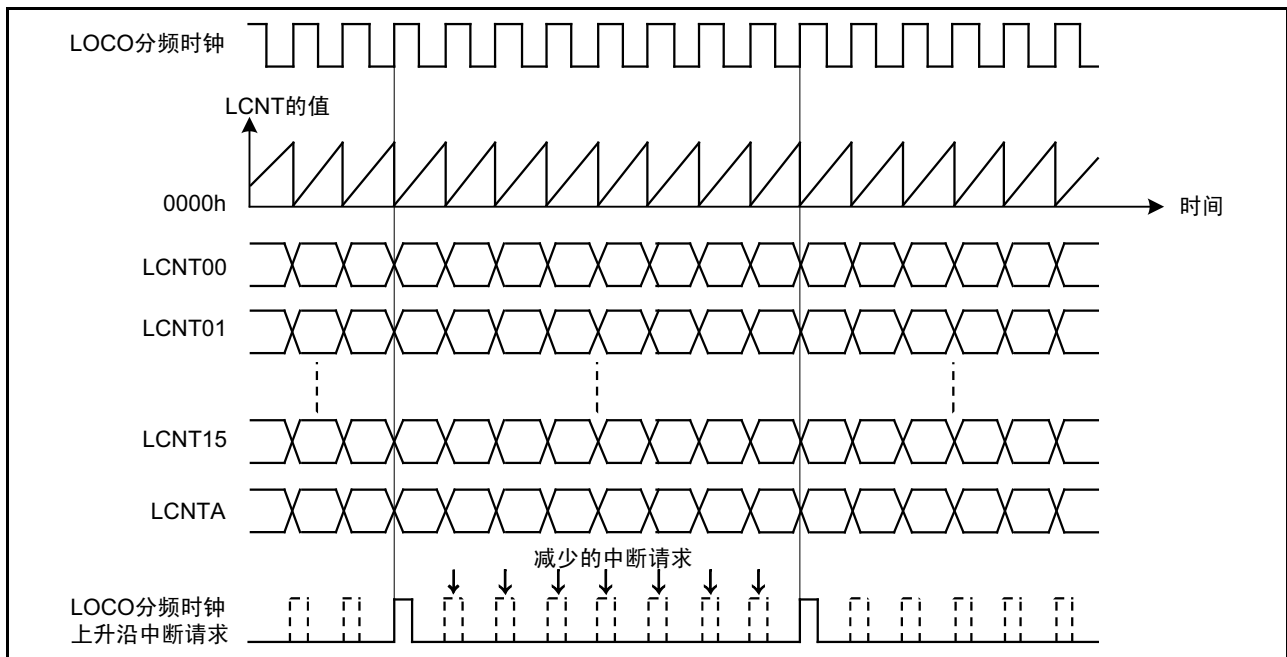


图 18.79 LOCO 计数减少功能的运行例子 (减少次数为 7 次、不减少计数结果)

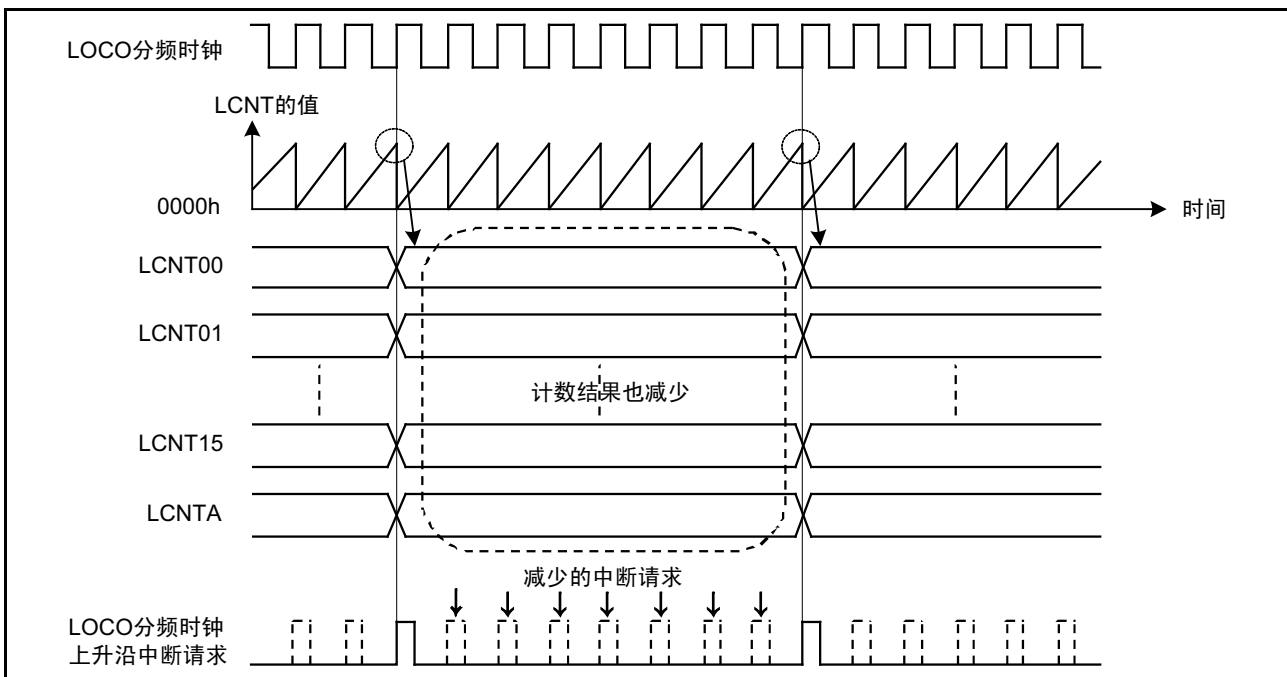


图 18.80 LOCO 计数减少功能的运行例子 (减少次数为 7 次、减少计数结果)

## 18.7 保护功能

### 18.7.1 计数器的写保护

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

### 18.7.2 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 18.81 所示。

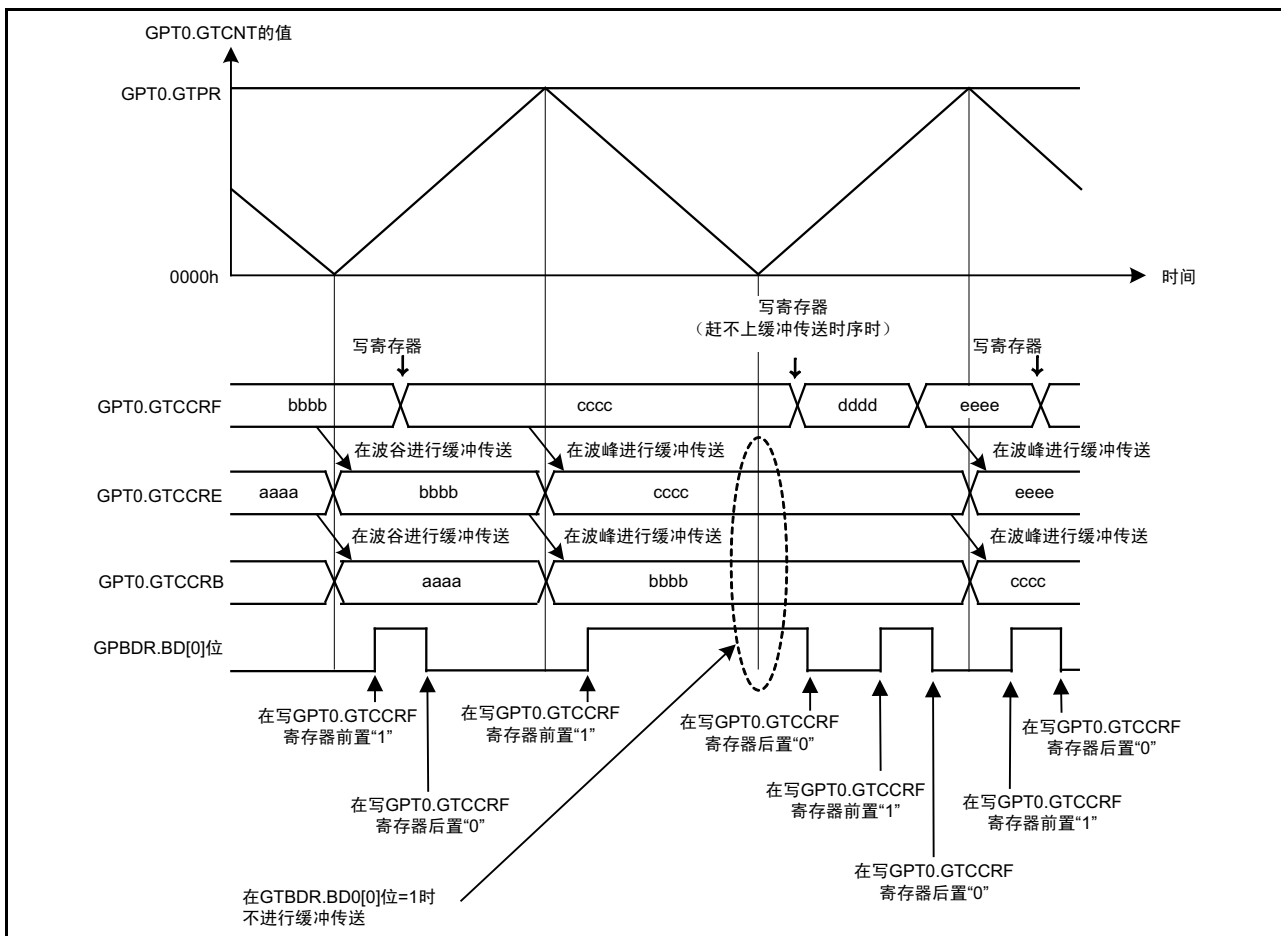


图 18.81 缓冲运行的抑制例子（三角波、双缓冲运行、在波谷和波峰都进行缓冲传送）

### 18.7.3 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 18.82 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。

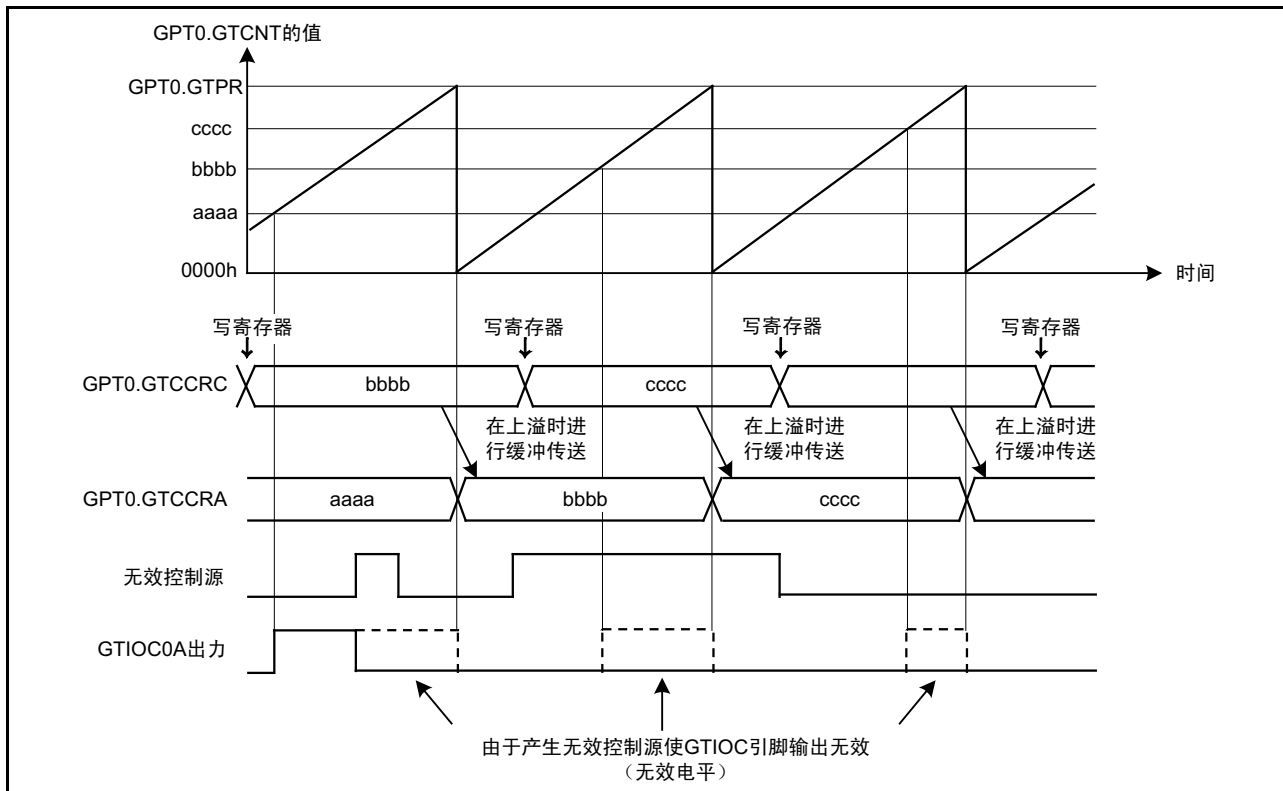


图 18.82 GTIOC 引脚输出的无效控制运行例子  
 (在锯齿波进行递增计数、缓冲运行、有效电平为 High 电平、通过 GTCCRA 的比较匹配进行 High 电平输出、在周期结束时进行 Low 电平输出)

### 18.7.4 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 (GTDTCCR.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 18.83 所示。

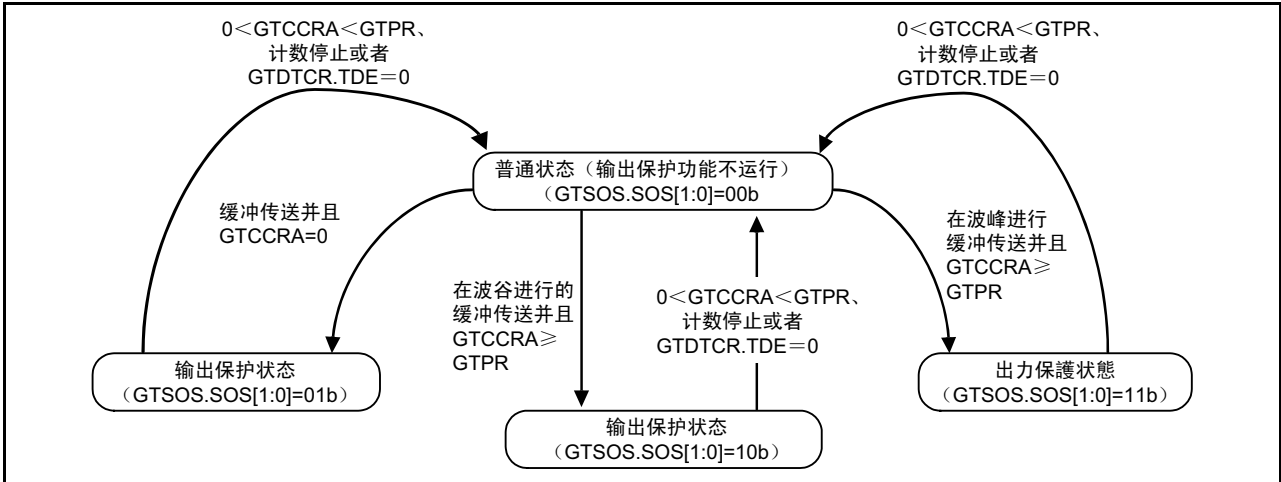


图 18.83 输出保护功能

#### (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 18.84、图 18.85 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 18.86、图 18.87 所示。

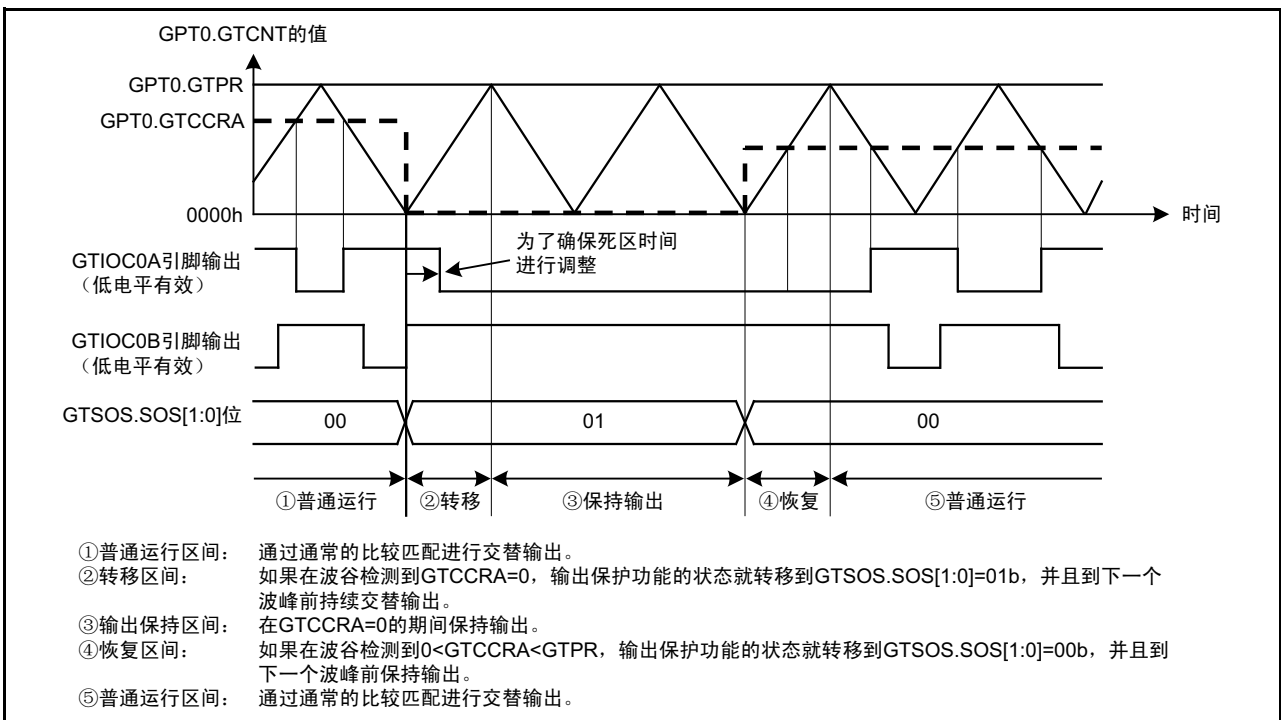


图 18.84 在波谷进行缓冲传送并且 GTCCRA 为“0”时的输出保护功能的运行例子 (在波谷进行缓冲传送时恢复  $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

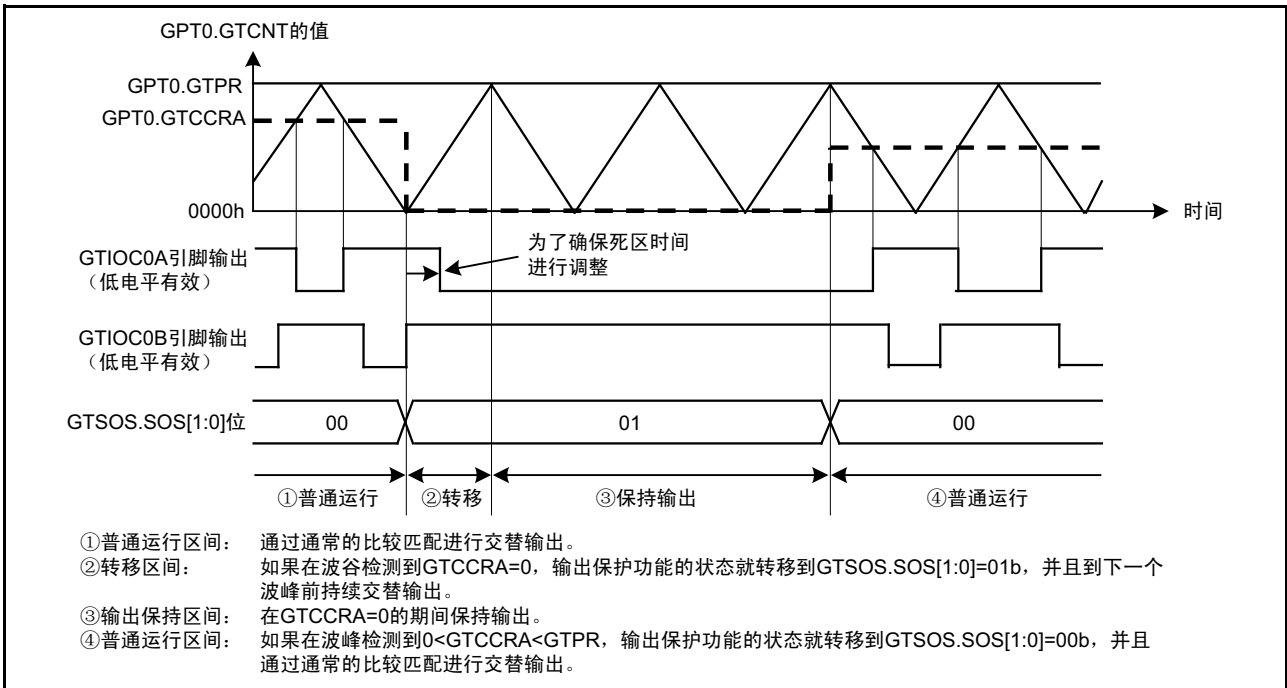


图 18.85 在波谷进行缓冲传送并且 GTCCRA 为 “0” 时的输出保护功能的运行例子  
(在波峰进行缓冲传送时恢复  $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

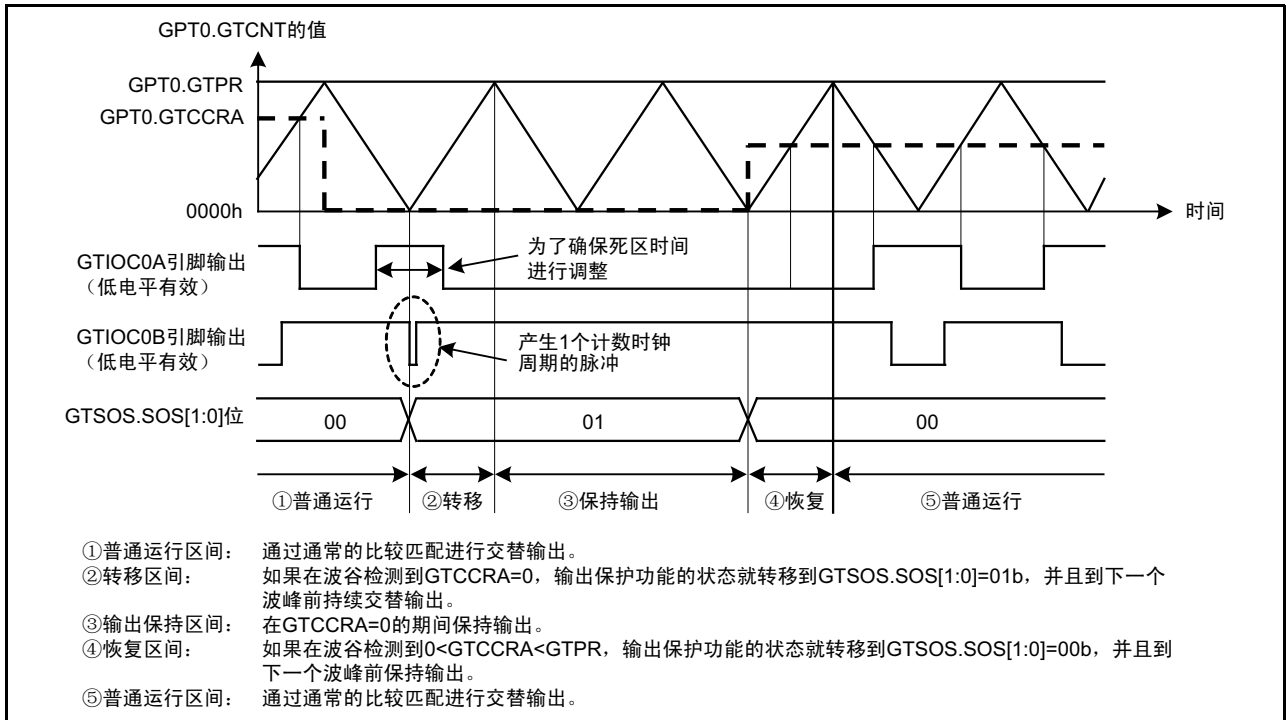


图 18.86 在波峰进行缓冲传送并且 GTCCRA 为 “0” 时的输出保护功能的运行例子  
(在波谷进行缓冲传送时恢复  $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

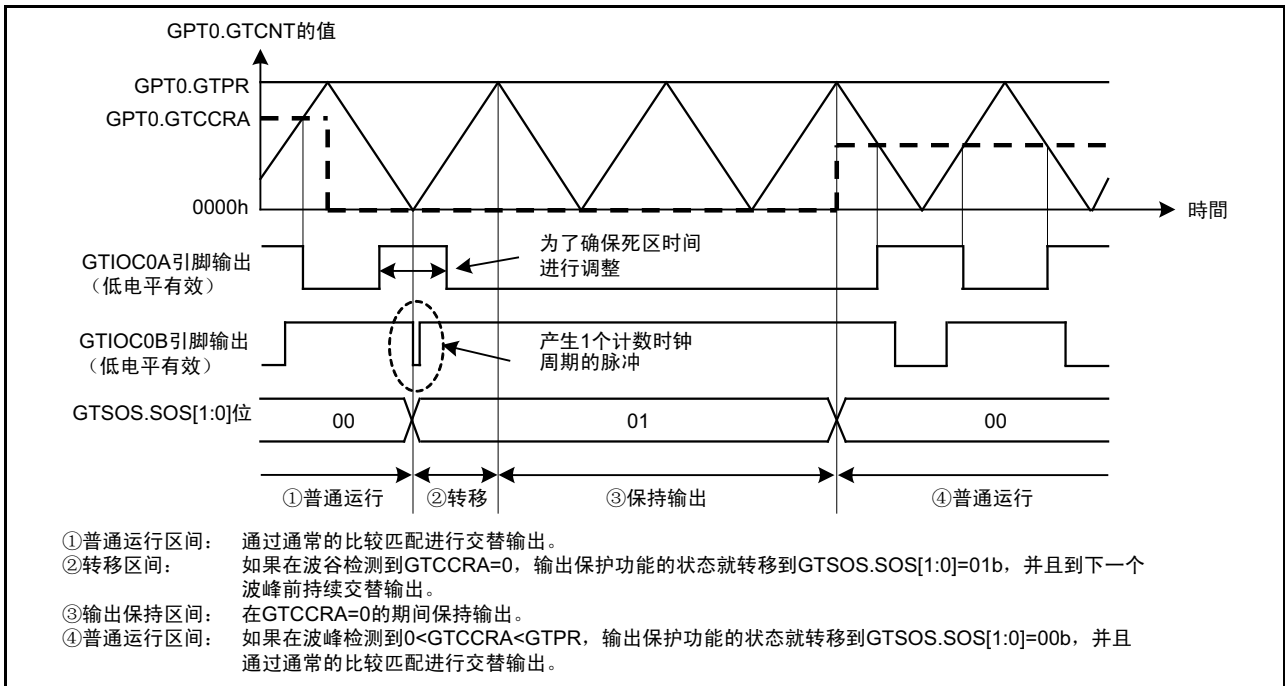


图 18.87 在波峰进行缓冲传送并且 GTCCRA 为 “0” 时的输出保护功能的运行例子  
 （在波峰进行缓冲传送时恢复  $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平）

(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 18.88、图 18.89 所示。

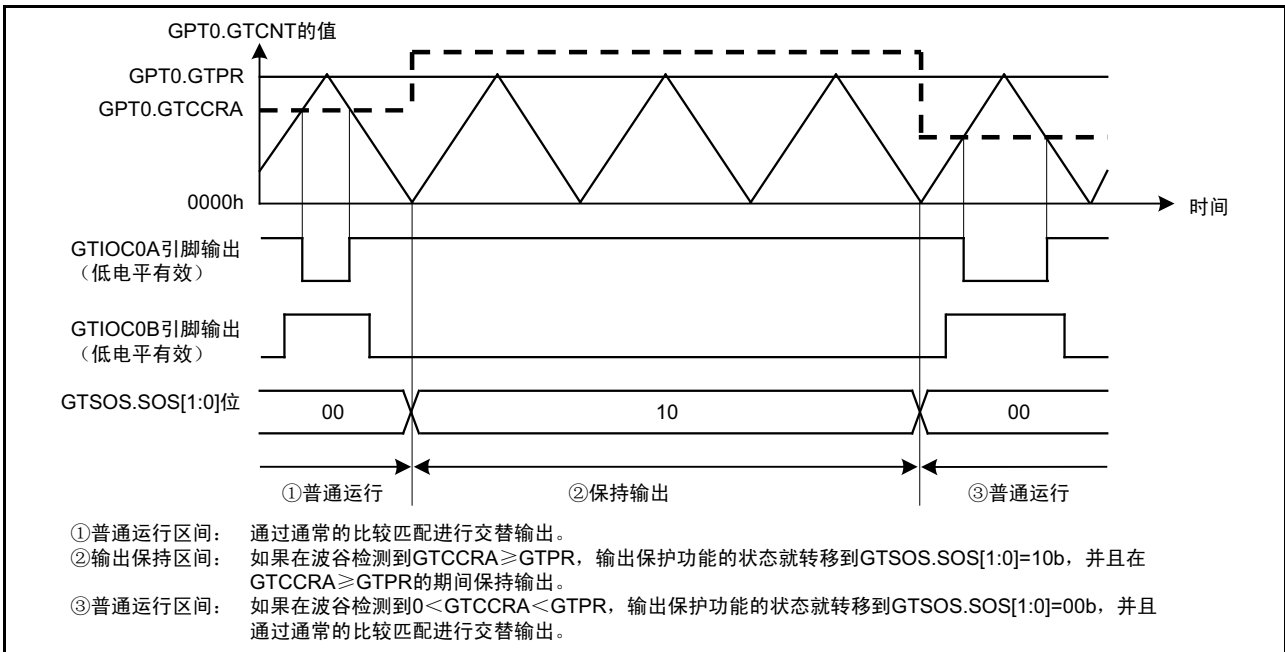


图 18.88 在波谷进行缓冲传送并且设定为  $GTCCRA \geq GTPR$  时的输出保护功能的运行例子  
 （在波谷进行缓冲传送时恢复  $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平）



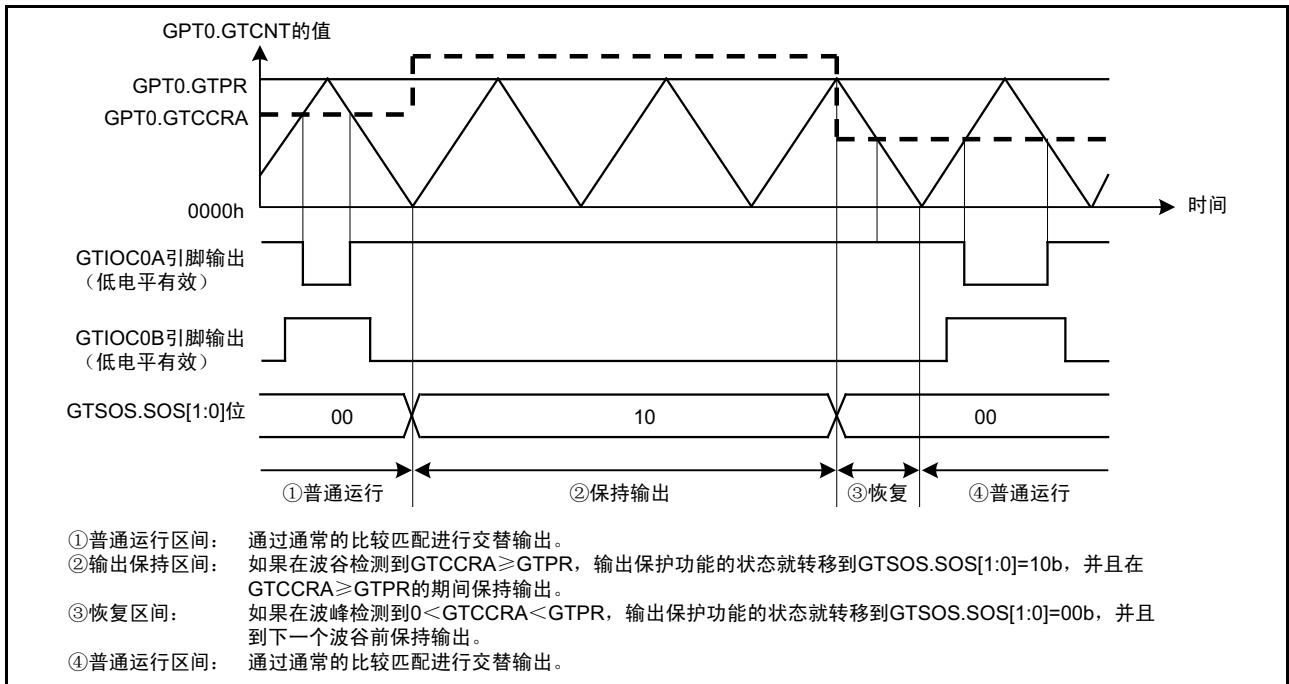


图 18.89 在波谷进行缓冲传送并且设定为  $GTCCRA \geq GTPR$  时的输出保护功能的运行例子 (在波峰进行缓冲传送时恢复  $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

(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 18.90、图 18.91 所示。

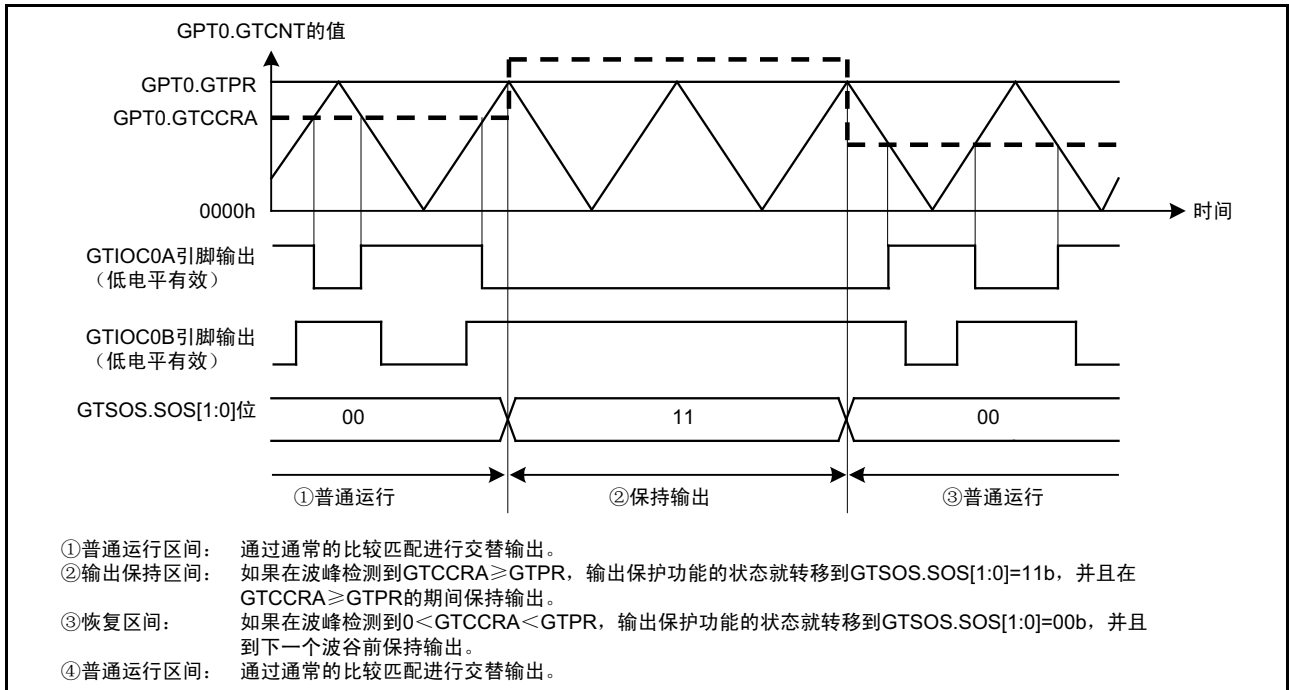


图 18.90 在波峰进行缓冲传送并且设定为  $GTCCRA \geq GTPR$  时的输出保护功能的运行例子 (在波峰进行缓冲传送时恢复  $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

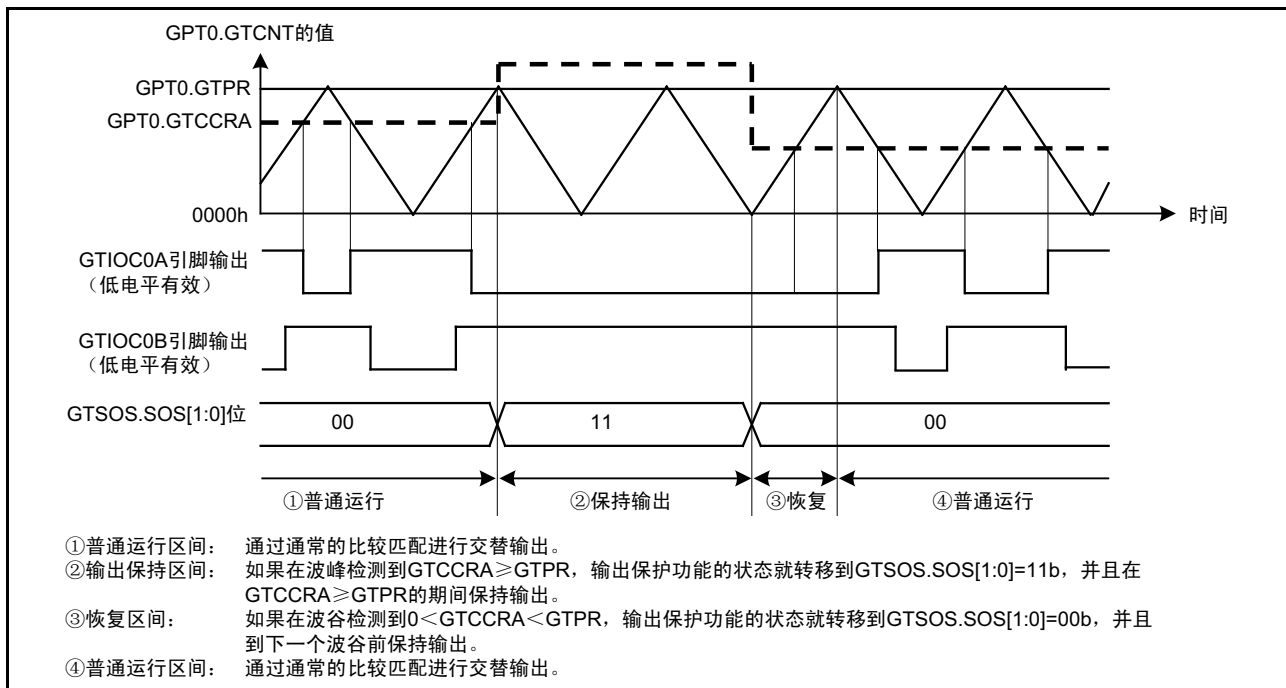


图 18.91 在波谷进行缓冲传送并且设定为  $GTCCRA \geq GTPR$  时的输出保护功能的运行例子  
 (在波谷进行缓冲传送时恢复  $0 < GTCCRA < GTPR$ 、有效电平为 Low 电平)

#### (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。

### 18.7.5 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“17. 端口输出允许 3（POE3）”。

## 18.8 输出引脚的初始化方法

### 18.8.1 复位后的引脚设定

GPT 的寄存器在复位时被初始化。设定 GTIOR 寄存器、GTONCR.OAE 位和 GTONCR.OBE 位，并且在将 GPT 功能输出到外部引脚后，开始计数。

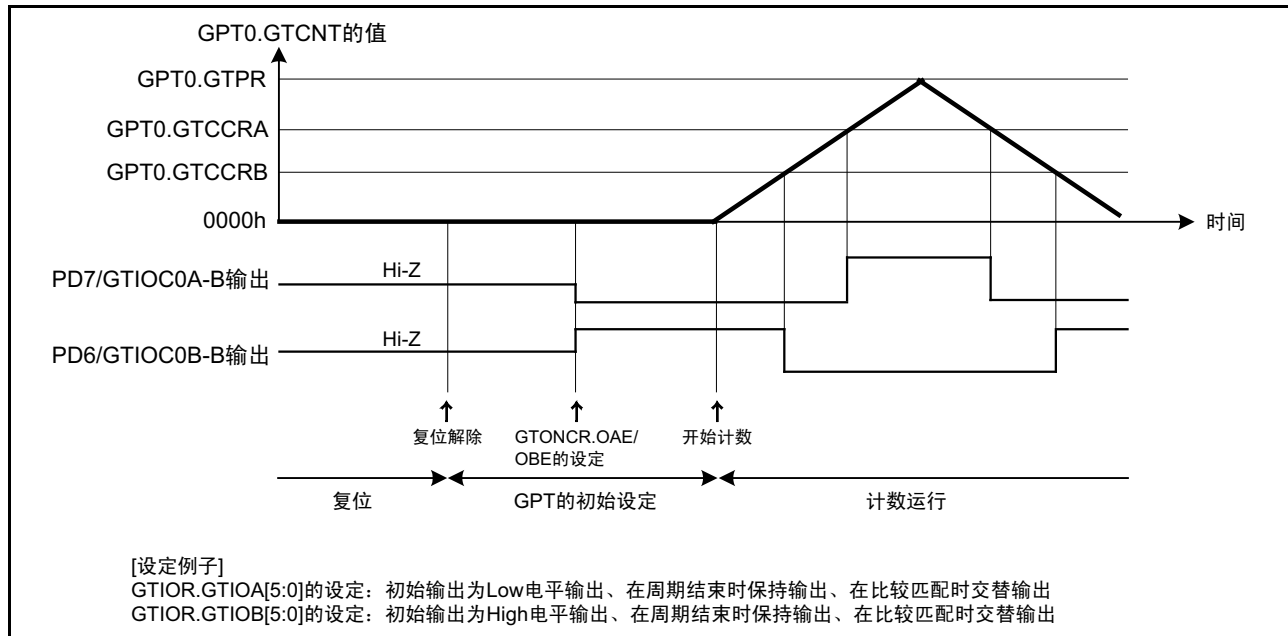


图 18.92 复位后的引脚设定例子

### 18.8.2 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 DDR 寄存器和 DR 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”，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。

## 18.9 使用时的注意事项

### 18.9.1 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“9. 低功耗功能”。

### 18.9.2 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“18.7.4 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。

### 18.9.3 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”。

### 18.9.4 使用 LOCO 计数功能时的低功耗功能的设定

在 LOCO 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 LOCO 计数。在 GPT 重新开始运行后设定  $LCCR.LCNTE=1$ ，重新开始 LOCO 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“9. 低功耗功能”。

## 19. 比较匹配定时器 (CMT)

RX62T 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。

### 19.1 概要

CMT 的规格如表 19.1 所示。

CMT (单元 0) 的框图如图 19.1 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 19.1 CMT 的规格

项目	功能
计数时钟	<ul style="list-style-type: none"> <li>4 种内部时钟</li> <li>各通道可分别从 PCLK/8、PCLK/32、PCLK/128、PCLK/512 中选择。</li> </ul>
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。

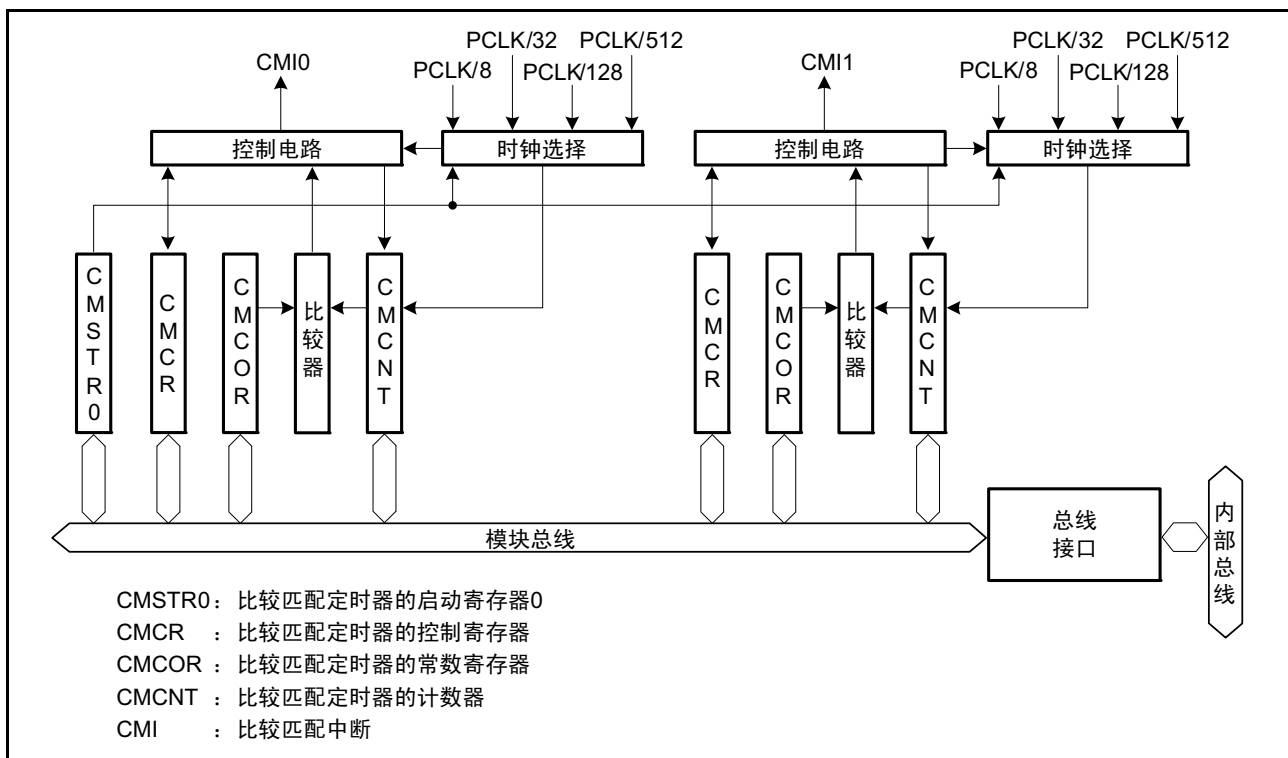


图 19.1 CMT (单元 0) 的框图

## 19.2 寄存器说明

CMT 的寄存器一览表如表 19.2 所示。

表 19.2 CMT 的寄存器一览表

单元	通道	寄存器名	符号	复位后的值	地址	存取长度
单元 0	CMT	比较匹配定时器的启动寄存器 0	CMSTR0	0000h	0008 8000h	16
	CMT0	比较匹配定时器的控制寄存器	CMCR	00x0h	0008 8002h	16
		比较匹配定时器的计数器	CMCNT	0000h	0008 8004h	16
		比较匹配定时器的常数寄存器	CMCOR	FFFFh	0008 8006h	16
	CMT1	比较匹配定时器的控制寄存器	CMCR	00x0h	0008 8008h	16
		比较匹配定时器的计数器	CMCNT	0000h	0008 800Ah	16
比较匹配定时器的常数寄存器		CMCOR	FFFFh	0008 800Ch	16	
单元 1	CMT	比较匹配定时器的启动寄存器 1	CMSTR1	0000h	0008 8010h	16
	CMT2	比较匹配定时器的控制寄存器	CMCR	00x0h	0008 8012h	16
		比较匹配定时器的计数器	CMCNT	0000h	0008 8014h	16
		比较匹配定时器的常数寄存器	CMCOR	FFFFh	0008 8016h	16
	CMT3	比较匹配定时器的控制寄存器	CMCR	00x0h	0008 8018h	16
		比较匹配定时器的计数器	CMCNT	0000h	0008 801Ah	16
比较匹配定时器的常数寄存器		CMCOR	FFFFh	0008 801Ch	16	

### 19.2.1 比较匹配定时器的启动寄存器 0 (CMSTR0)

地址 0008 8000h

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

位	符号	位名	功能	R/W
b0	STR0	计数开始 0 位	0: CMT0.CMCNT 计数器停止计数 1: CMT0.CMCNT 计数器开始计数	R/W
b1	STR1	计数开始 1 位	0: CMT1.CMCNT 计数器停止计数 1: CMT1.CMCNT 计数器开始计数	R/W
b15-b2	—	保留位	读写值都为“0”。	R/W

CMSTR0 寄存器设定开始或者停止 CMT0.CMCNT 计数器和 CMT1.CMCNT 计数器的计数。

#### STR0 位 (计数开始 0 位)

此位选择开始或者停止 CMT0.CMCNT 计数器的计数。

#### STR1 位 (计数开始 1 位)

此位选择开始或者停止 CMT1.CMCNT 计数器的计数。

## 19.2.2 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

CMSTR1 寄存器设定开始或者停止 CMT2.CMCNT 计数器和 CMT3.CMCNT 计数器的计数。

## STR2 位 (计数开始 2 位)

此位选择开始或者停止 CMT2.CMCNT 计数器的计数。

## STR3 位 (计数开始 3 位)

此位选择开始或者停止 CMT3.CMCNT 计数器的计数。



### 19.2.3 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（CMI <sub>n</sub> ） 1: 允许比较匹配中断（CMI <sub>n</sub> ）	R/W
b7	—	保留位	读取值为不定值，只能写“1”。	R/W
b15-b8	—	保留位	读写值都为“0”。	R/W

CMCR 寄存器是设定递增计数时钟的寄存器。

如果 CMCR 寄存器的改写与比较匹配发生竞争，就忽视 CMCR 寄存器的写操作，详细内容请参照“19.5.4 改写比较匹配定时器的控制寄存器（CMCR）时的注意事项”。

#### CKS[1:0] 位（时钟选择位）

这些位从外围模块时钟（PCLK）分频后得到的 4 种内部时钟中选择输入到 CMCNT 计数器的计数时钟。

如果将 CMSTR<sub>m</sub>.STR<sub>n</sub> 位（m=0、1，n=0～3）置“1”，对应的 CMCNT 计数器就通过 CKS[1:0] 位选择的计数时钟开始递增计数。

#### 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（CMI<sub>n</sub>）（n=0～3）的发生。

### 19.2.4 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] 位选择计数时钟并且将 CMSTRm.STRn 位 (m=0、1, n=0 ~ 3) 置“1”，CMCNT 计数器就通过计数时钟开始递增计数。

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<sub>n</sub>) (n=0 ~ 3)。

不能在 CMCNT 计数器停止计数的状态下给 CMCNT 寄存器和 CMCOR 寄存器设定相同的值，详细内容请参照“19.5.5 比较匹配定时器的计数器 (CMCNT) 和比较匹配常数寄存器 (CMCOR) 的注意事项”。

### 19.2.5 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寄存器。

不能在 CMCNT 计数器停止计数的状态下给 CMCNT 寄存器和 CMCOR 寄存器设定相同的值，详细内容请参照“19.5.5 比较匹配定时器的计数器 (CMCNT) 和比较匹配常数寄存器 (CMCOR) 的注意事项”。

## 19.3 运行说明

### 19.3.1 周期计数

如果通过 `CMCR.CKS[1:0]` 位选择计数时钟并且将 `CMSTRm.STRn` 位 ( $m=0, 1$ ,  $n=0 \sim 3$ ) 置“1”，`CMCNT` 计数器就通过所选的计数时钟开始递增计数。

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`) ( $n=0 \sim 3$ )。`CMCNT` 计数器从“0000h”重新开始递增计数，`CMCNT` 计数器的运行如图 19.2 所示。

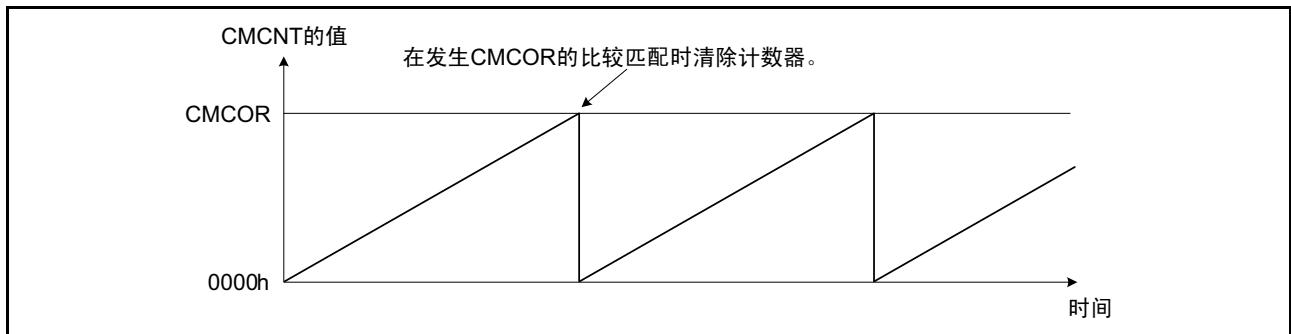


图 19.2 CMCNT 计数器的运行

### 19.3.2 CMCNT 计数器的计数时序

能通过 `CMCR.CKS[1:0]` 位从外围模块时钟 (`PCLK`) 分频后的 4 种内部时钟 (`PCLK/8`、`PCLK/32`、`PCLK/128`、`PCLK/512`) 中选择计数时钟，此时的 `CMCNT` 计数器的计数时序如图 19.3 所示。

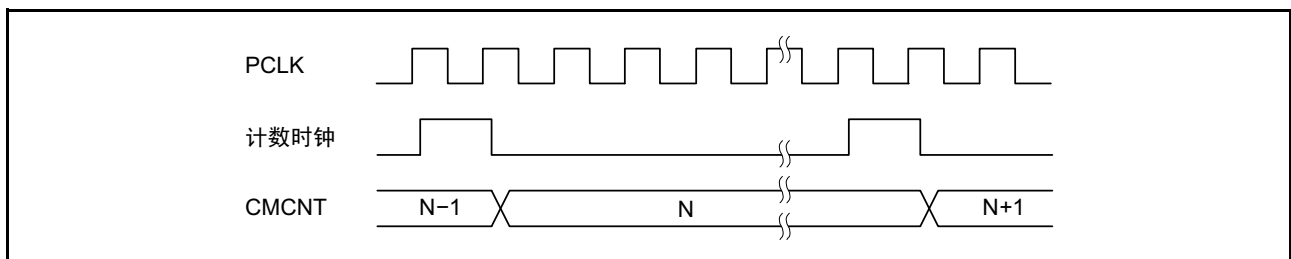


图 19.3 CMCNT 计数器的计数时序

## 19.4 中断

### 19.4.1 中断源

CMT 的各通道有比较匹配中断 (CMI<sub>n</sub>) (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“11. 中断控制器 (ICU) ”。

表 19.3 CMT 的中断源

名称	中断源	中断状态标志	DTC 的启动
CMI0	CMT0.CMCNT 和 CMT0.CMCOR 的比较匹配	ICU.IR028.IR	能
CMI1	CMT1.CMCNT 和 CMT1.CMCOR 的比较匹配	ICU.IR029.IR	能
CMI2	CMT2.CMCNT 和 CMT2.CMCOR 的比较匹配	ICU.IR030.IR	能
CMI3	CMT3.CMCNT 和 CMT3.CMCOR 的比较匹配	ICU.IR031.IR	能

### 19.4.2 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<sub>n</sub>) (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计数时钟前，不产生比较匹配信号。

比较匹配中断的发生时序时序如图 19.4 所示。

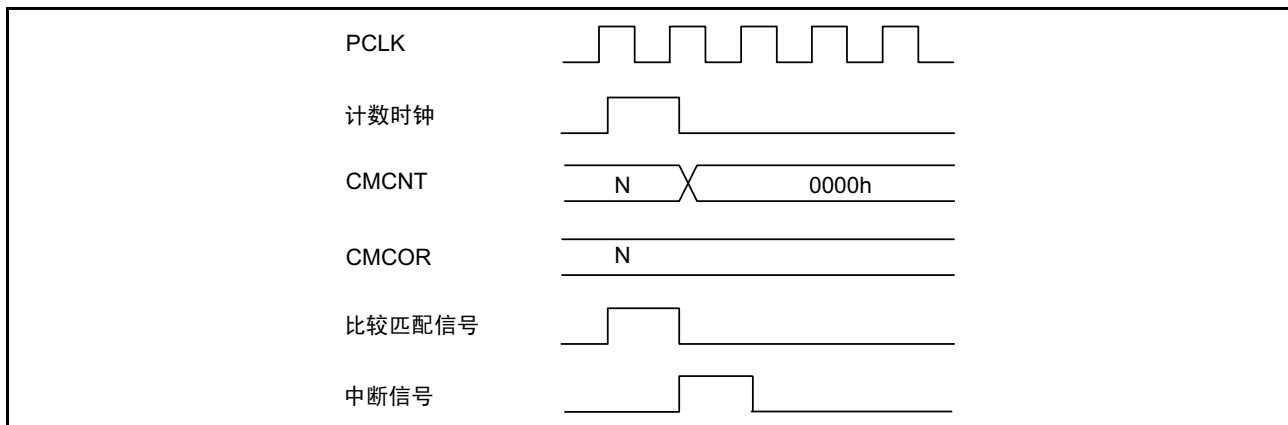


图 19.4 比较匹配中断的发生时序

### 19.5 使用时的注意事项

#### 19.5.1 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“9. 低功耗功能”。

#### 19.5.2 比较匹配定时器的计数器 (CMCNT) 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 19.5 所示。

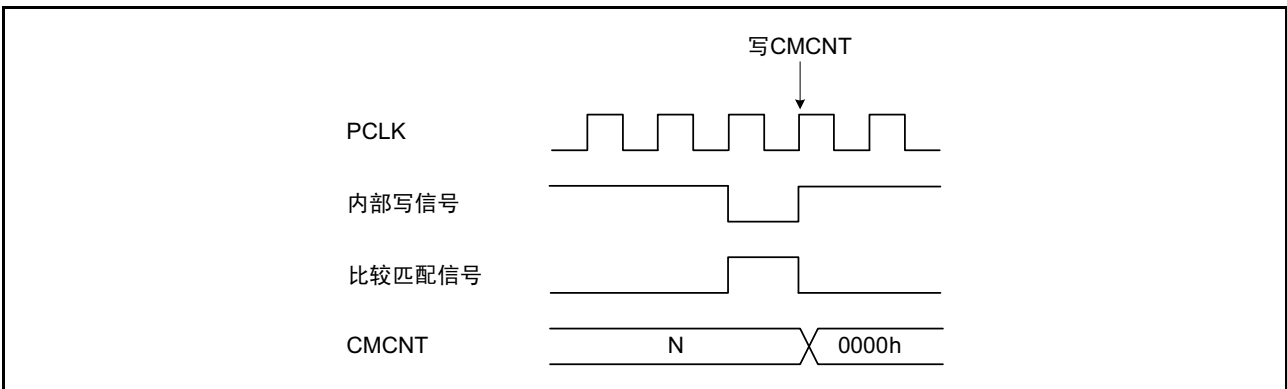


图 19.5 CMCNT 计数器的写和比较匹配的竞争

#### 19.5.3 比较匹配定时器的计数器 (CMCNT) 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 19.6 所示。

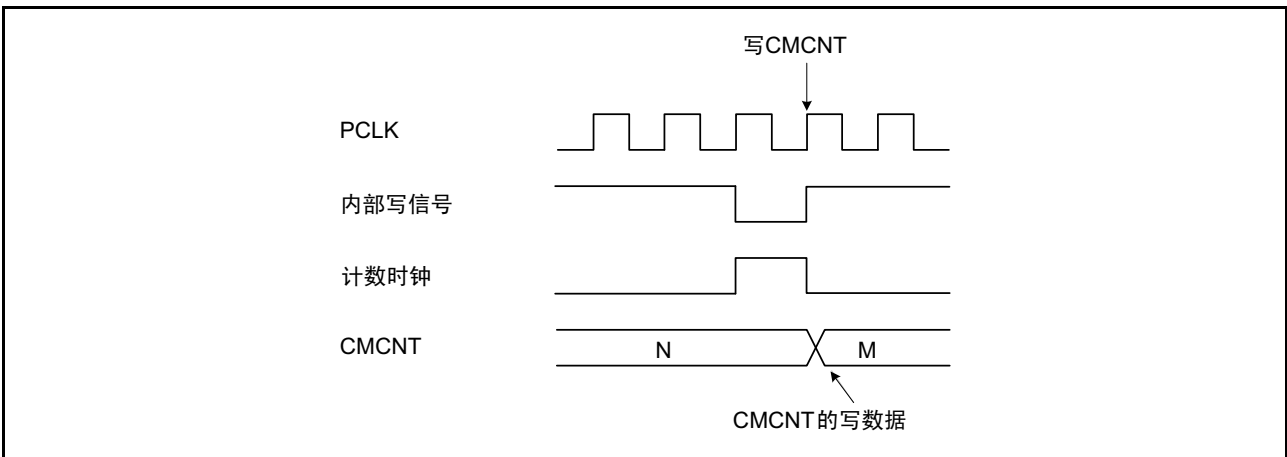


图 19.6 CMCNT 计数器的写和递增计数的竞争

#### 19.5.4 改写比较匹配定时器的控制寄存器 (CMCR) 时的注意事项

如果 CMCR 寄存器的改写与比较匹配发生竞争，就忽视 CMCR 寄存器的写操作。因此，必须在写 CMCR 寄存器后读 CMCR 寄存器，确认写数据是否被正确写入。当写数据没有被正确写入时，必须再次写 CMCR 寄存器。

因为 CMCR 寄存器的 bit7 的读取值为不定值，所以在比较写数据时必须注意。

#### 19.5.5 比较匹配定时器的计数器 (CMCNT) 和比较匹配常数寄存器 (CMCOR) 的注意事项

不能在 CMCNT 计数器停止计数的状态下给 CMCNT 计数器和 CMCOR 寄存器设定相同的值。

如果在 CMCNT 计数器停止计数的状态下给 CMCNT 计数器和 CMCOR 寄存器设定相同的值，就发生比较匹配，与计数停止状态无关。此时，如果比较匹配中断允许位 (CMCR.CMIE 位) 为“1” (允许)，就发生比较匹配中断。

与比较匹配中断是否禁止无关，如果因与 CMCOR 寄存器的值相同而发生比较匹配，CMCNT 计数器就自动清为“0000h”。

## 20. 看门狗定时器 (WDT)

看门狗定时器 (WDT) 是 8 位定时器, 如果因系统失控等而不改写计数器的值导致发生上溢, 就将上溢信号 (WDTOVF#) 输出到外部, 同时能对 LSI 的内部进行复位。

在不用作看门狗定时器时, 也能用作间隔定时器。在用作间隔定时器的情况下, 每当计数器发生上溢时, 就发生间隔定时器中断。

### 20.1 概要

看门狗定时器的规格和框图分别如表 20.1 和图 20.1 所示。

表 20.1 WDT 的规格

项目	内容
计数时钟	PCLK/4、PCLK/64、PCLK/128、PCLK/512、PCLK/2048、PCLK/8192、PCLK/32768、PCLK/131072
通道数	8 位 × 1 个通道
计数器清除	写 TCNT。
运行模式	看门狗定时器模式和间隔定时器模式的转换
看门狗定时器模式	如果计数器发生上溢, 就将 WDTOVF# 信号输出到外部, 并且能选择是否同时对 LSI 内部进行复位。
间隔定时器模式	如果计数器发生上溢, 就产生间隔定时器中断 (WOVI)。

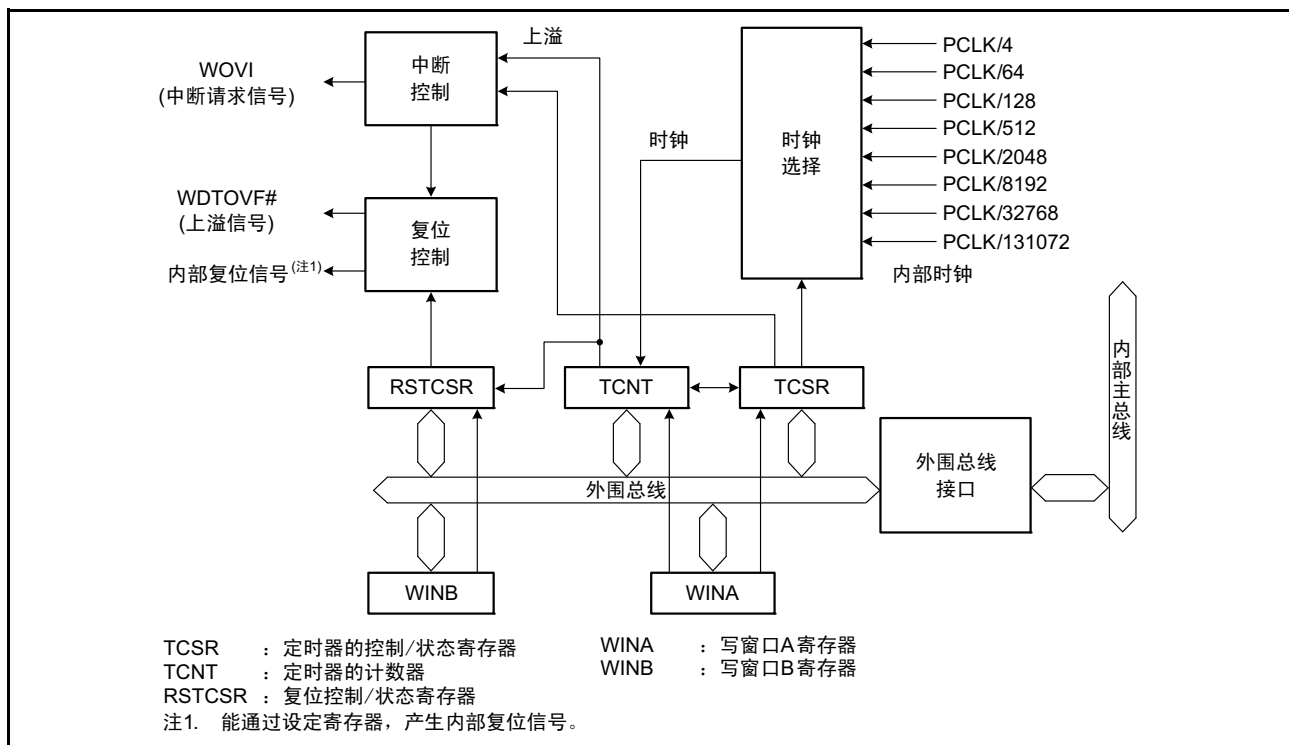


图 20.1 WDT 的框图

WDT 使用的输入 / 输出引脚如表 20.2 所示。

表 20.2 输入 / 输出引脚

引脚名	输入 / 输出	功能
WDTOVF#	输出	输出看门狗定时器模式中的计数器上溢信号。

## 20.2 寄存器说明

WDT 的寄存器一览表如表 20.3 所示。

表 20.3 WDT 的寄存器一览表

寄存器名	符号	复位后的值	地址	存取长度
定时器的控制 / 状态寄存器	TCSR	x8h	0008 8028h (注 1)	8
定时器的计数器	TCNT	00h	0008 8029h (注 1)	8
复位控制 / 状态寄存器	RSTCSR	1Fh	0008 802Bh (注 1)	8
写窗口 A 寄存器	WINA	—	0008 8028h (注 2)	16
写窗口 B 寄存器	WINB	—	0008 802Ah (注 2)	16

注 1. 这是只读寄存器。

注 2. 这是只写寄存器。

### 20.2.1 定时器的计数器 (TCNT)

地址 0008 8029h



TCNT 计数器是对内部时钟进行计数的 8 位递增计数器。

如果将 TCSR.TME 位置“0”，TCNT 计数器就被初始化为“00h”。

必须以 8 位为单位读 TCNT。

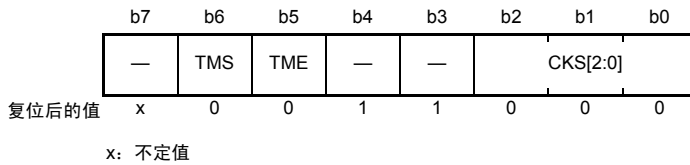
在写 TCNT 寄存器时，必须以 16 位为单位写 WINA 寄存器。

详细内容请参照“20.5.1 存取寄存器时的注意事项”。



## 20.2.2 定时器的控制 / 状态寄存器 (TCSR)

地址 0008 8028h



位	符号	位名	功能	R/W
b2-b0	CKS[2:0]	时钟选择位	b2 b0 0 0 0: PCLK/4 (周期为 20.4 $\mu$ s) 0 0 1: PCLK/64 (周期为 326.4 $\mu$ s) 0 1 0: PCLK/128 (周期为 652.8 $\mu$ s) 0 1 1: PCLK/512 (周期为 2.6ms) 1 0 0: PCLK/2048 (周期为 10.4ms) 1 0 1: PCLK/8192 (周期为 41.8ms) 1 1 0: PCLK/32768 (周期为 167.1ms) 1 1 1: PCLK/131072 (周期为 668.5ms) 注: ( ) 内表示 PCLK 为 50MHz 时的上溢周期。	R/W
b4-b3	—	保留位	读写值都为“1”。	R/W
b5	TME	定时器允许位	0: TCNT 计数器停止计数并且被初始化为“00h” 1: TCNT 计数器开始计数	R/W
b6	TMS	定时器模式选择位	0: 间隔定时器模式 在 TCNT 计数器发生上溢时, 请求间隔定时器中断 (WOVI)。 1: 看门狗定时器模式 在 TCNT 计数器发生上溢时, 将 WDTOVF# 输出到外部。	R/W
b7	—	保留位	读取值为不定值, 只能写“1”。	R/W

TCSR 寄存器是选择 TCNT 计数器的输入时钟和模式的寄存器。

必须以 8 位为单位读 TCSR。

在写 TCSR 寄存器时, 必须以 16 位为单位写 WINA 寄存器。

详细内容请参照“20.5.1 存取寄存器时的注意事项”。

## CKS[2:0] 位 (时钟选择位)

这些位选择 TCNT 计数器的输入时钟。

## TME 位 (定时器允许位)

此位选择开始或者停止 TCNT 计数器的计数。

如果将此位置“1”, TCNT 计数器就开始计数; 如果置“0”, TCNT 计数器就停止计数并且被初始化为“00h”。

## TMS 位 (定时器模式选择位)

此位选择是用作看门狗定时器还是用作间隔定时器。

### 20.2.3 复位控制 / 状态寄存器 (RSTCSR)

地址 0008 802Bh

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

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

注 1. 为了清除标志, 只能写“0”。

RSTCSR 寄存器控制因 TCNT 计数器的上溢而产生的内部复位信号以及选择内部复位信号的种类。

通过 RES# 引脚的复位信号和深度软件待机复位将 RSTCSR 寄存器初始化为“1Fh”, 而不通过因看门狗定时器的上溢而产生的内部复位信号进行初始化。

必须以 8 位为单位读 RSTCSR 寄存器。

写 RSTCSR 寄存器时, 必须以 16 位为单位写 WINB 寄存器。

详细内容请“20.5.1 存取寄存器时的注意事项”。

#### RSTE 位 (复位允许位)

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

#### WOVF 标志 (看门狗定时器上溢标志)

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

[ 为“1”的条件 ]

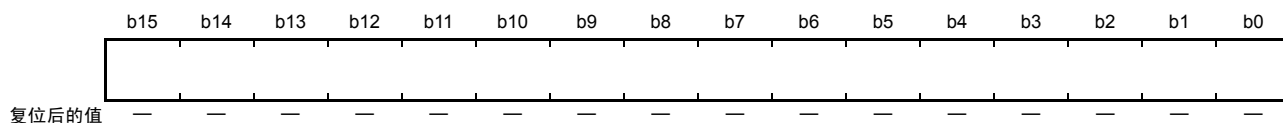
- 在看门狗定时器模式中, TCNT 计数器发生上溢 (“FFh”→“00h”) 时

[ 为“0”的条件 ]

- 读“1”后写“0”时

### 20.2.4 写窗口 A 寄存器 (WINA)

地址 0008 8028h



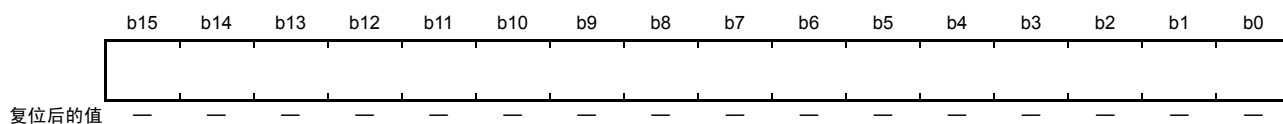
WINA 寄存器是改写 TCNT 计数器值和 TCSR 寄存器值的只写寄存器。

TCNT 计数器和 TCSR 寄存器的写法不同，详细内容请参照“20.5.1 存取寄存器时的注意事项”。

必须以 16 位为单位写 WINA 寄存器。

### 20.2.5 写窗口 B 寄存器 (WINB)

地址 0008 802Ah



WINB 寄存器是改写 RSTCSR 寄存器值的只写寄存器。

给 RSTCSR.WOVF 标志写“0”的方法不同于 RSTCSR.RSTE 位的写法，详细内容请参照“20.5.1 存取寄存器时的注意事项”。

必须以 16 位为单位写 WINB 寄存器。

## 20.3 运行说明

### 20.3.1 看门狗定时器模式

在用作看门狗定时器模式时，必须将 TCSR.TMS 位置“1”（看门狗定时器模式）并且将 TCSR.TME 位置“1”（TCNT 计数器开始计数）。

如果在用作看门狗定时器时因系统失控等而不改写 TCNT 计数器的值导致发生上溢，就输出 WDTOVF# 信号。在系统正常运行期间，TCNT 计数器不发生上溢。为了避免上溢的发生，必须编写在 TCNT 计数器发生上溢前改写 TCNT 计数器的值（通常写“00h”）的程序。而且，能在看门狗定时器模式中通过 WDTOVF# 信号对 LSI 内部进行复位。

如果在看门狗定时器模式中 TCNT 计数器发生上溢，RSTCSR.WOVF 标志就变为“1”。另外，如果将 RSTCSR.RSTE 位置“1”，就在 TCNT 计数器发生上溢时输出 WDTOVF# 信号，同时产生对 LSI 内部进行复位的信号。如果 RES# 引脚输入信号的复位和看门狗定时器上溢的复位同时发生，就优先进行 RES# 引脚的复位，并且 RSTCSR.WOVF 标志变为“0”。

当 RSTE 位为“1”时，在 PCLK 的 257 个状态期间输出 WDTOVF# 信号；当 RSTE 位为“0”时，在 PCLK 的 256 个状态期间输出 WDTOVF# 信号。在 PCLK 的 1027 个状态期间输出内部复位信号。

因为在 RSTE 位为“1”时产生内部复位信号并且系统时钟的控制寄存器（SCKCR）被复位，所以 PCLK 输入时钟的倍率变为初始值。

因为在 RSTE 位为“0”时不产生内部复位信号并且保持 SCKCR 寄存器的设定，所以 PCLK 输入时钟的倍率不变。

在看门狗定时器模式中，如果 TCNT 计数器发生上溢，WOVF 标志就被置“1”。如果事先将 RSTE 位置“1”，就在 TCNT 计数器发生上溢时，对整个 LSI 产生内部复位信号。

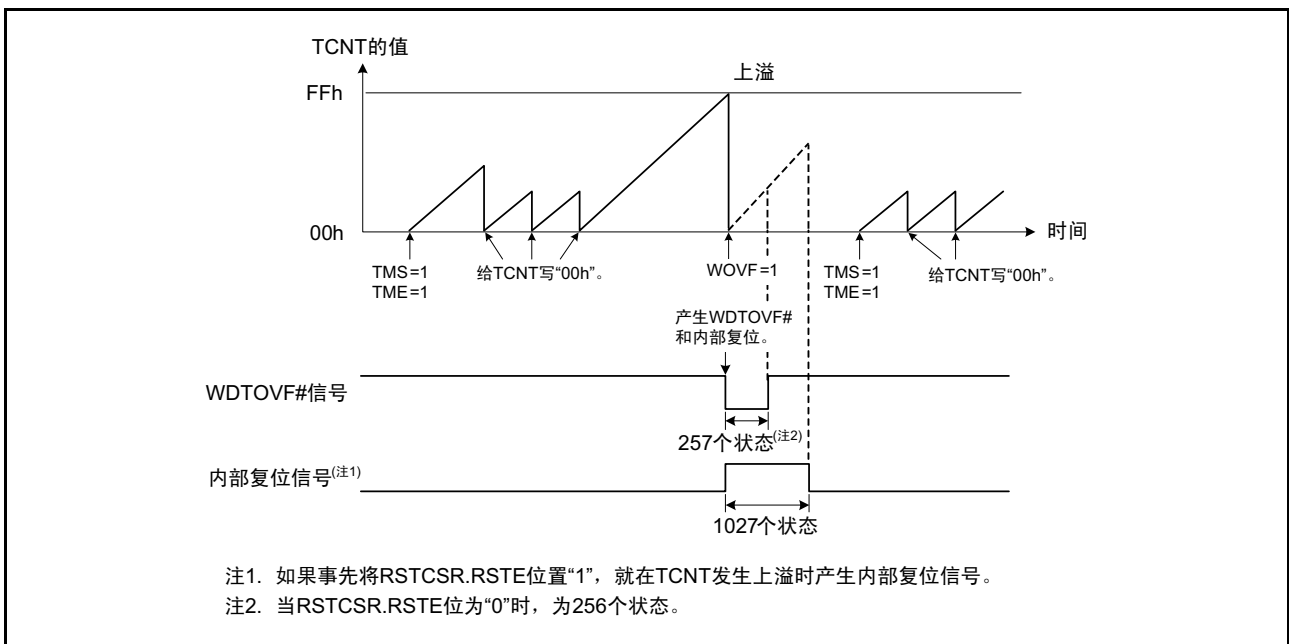


图 20.2 看门狗定时器模式的运行

### 20.3.2 间隔定时器模式

在用作间隔定时器时，必须将 TCSR.TMS 位置“0”（间隔定时器模式）并且将 TCSR.TME 位置“1”（TCNT 计数器开始计数）。

在用作间隔定时器的情况下，每当 TCNT 计数器发生上溢时，就发生间隔定时器中断（WOVI）。因此，能按固定的时间发生中断。

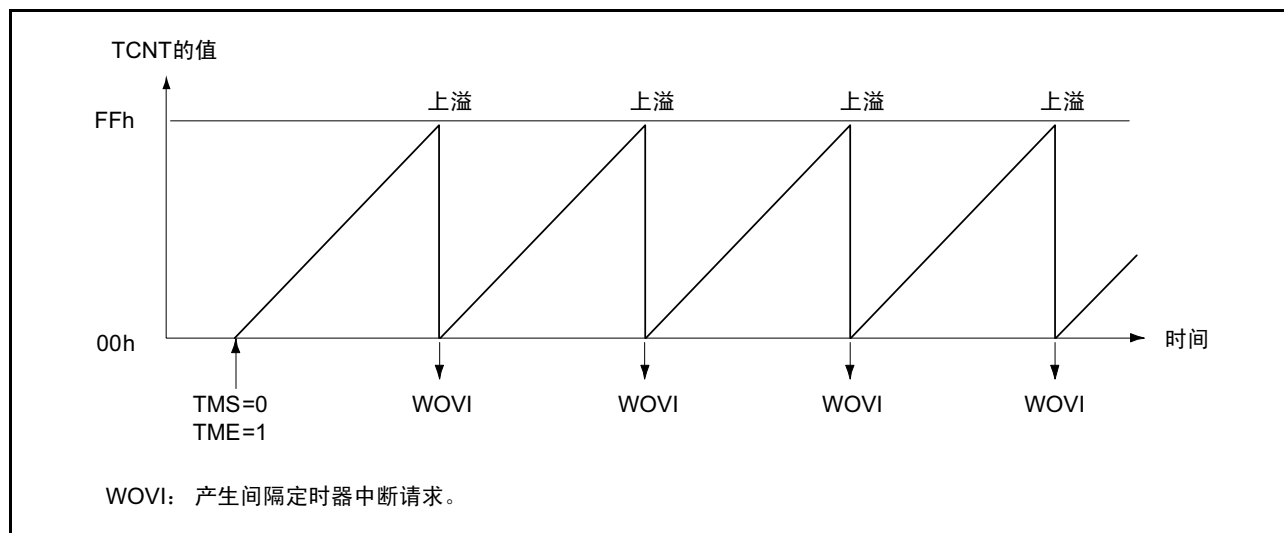


图 20.3 间隔定时器模式的运行

## 20.4 中断源

在间隔定时器模式中，通过 TCNT 计数器的上溢发生间隔定时器中断（WOVI），详细内容请参照“11. 中断控制器（ICU）”。

表 20.4 WDT 的中断源

名称	中断源	DTC 的启动	中断状态标志
WOVI	TCNT 的上溢	不能	ICU.IR096.IR

## 20.5 使用时的注意事项

### 20.5.1 存取寄存器时的注意事项

为了 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器不被轻易改写，其写法不同于一般寄存器。

#### (1) 写 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器

在写 TCNT 计数器和 TCSR 寄存器时，必须对写窗口 A 寄存器 (WINA) (0008 8028h) 使用字传送指令。TCNT 计数器和 TCSR 寄存器的写操作被分配在相同的地址，因此必须如图 20.4 进行设定。

在写 TCNT 计数器时，必须在给高位字节设定“5Ah”并且给低位字节设定 TCNT 计数器的写数据后进行传送。

在写 TCSR 寄存器时，必须在给高位字节设定“A5h”并且给低位字节设定 TCSR 寄存器的写数据后进行传送。

在写 RSTCSR 寄存器时，必须对写窗口 B 寄存器 (WINB) (0008 802Ah) 使用字传送指令。

给 RSTCSR.WOVF 标志写“0”的方法不同于 RSTCSR.RSTE 位的写法。因此，必须按图 20.4 所示的内容传送数据。

在给 WOVF 标志写“0”时，如图 20.4 所示，必须在将高位字节置“5Ah”，将低位字节置“00h”后，以 16 位为单位写数据。此时，不影响 RSTE 位。

在写 RSTE 位时，如图 20.4 所示，必须在将高位字节置“5Ah”并且将低位字节设定为 RSTCSR 寄存器的写数据后，以 16 位为单位写数据。此时，不影响 WOVF 标志。

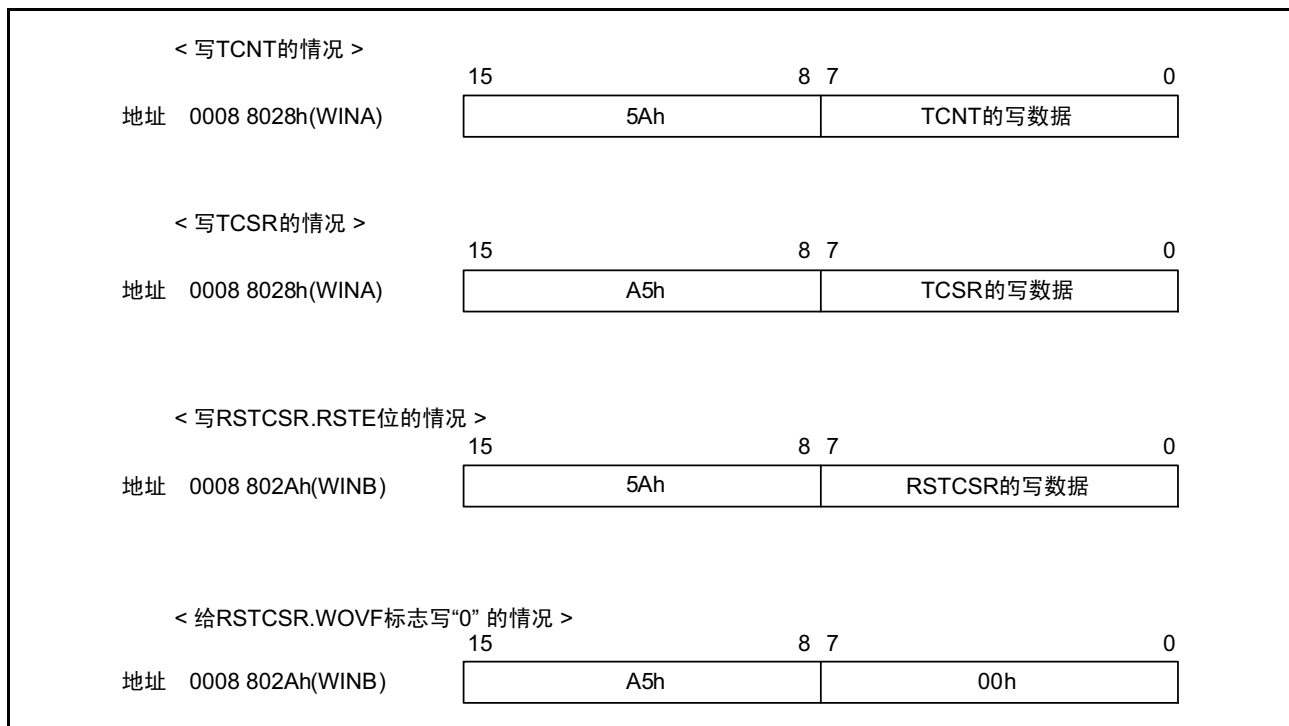


图 20.4 写 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器

(2) 读 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器

TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器的读法和一般寄存器相同。

TCSR 寄存器必须以 8 位为单位存取地址 0008 8028h，TCNT 计数器必须以 8 位为单位存取地址 0008 8029h，RSTCSR 寄存器必须以 8 位为单位存取地址 0008 802Bh。

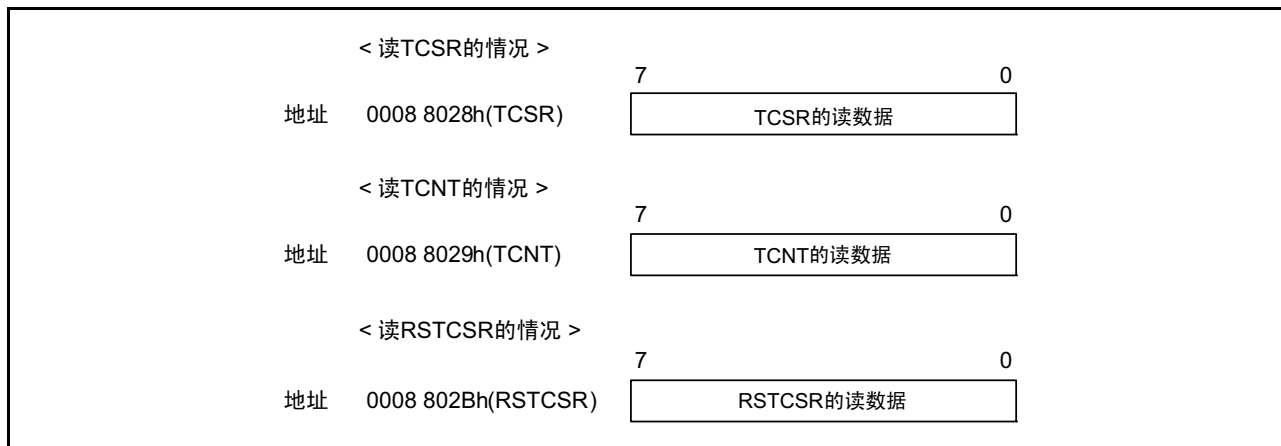


图 20.5 读 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器

20.5.2 定时器的计数器 (TCNT) 的写和递增计数的竞争

即使在写 TCNT 计数器的过程中输入递增计数的时钟，也不进行递增计数而优先写 TCNT 计数器，如图 20.6 所示。

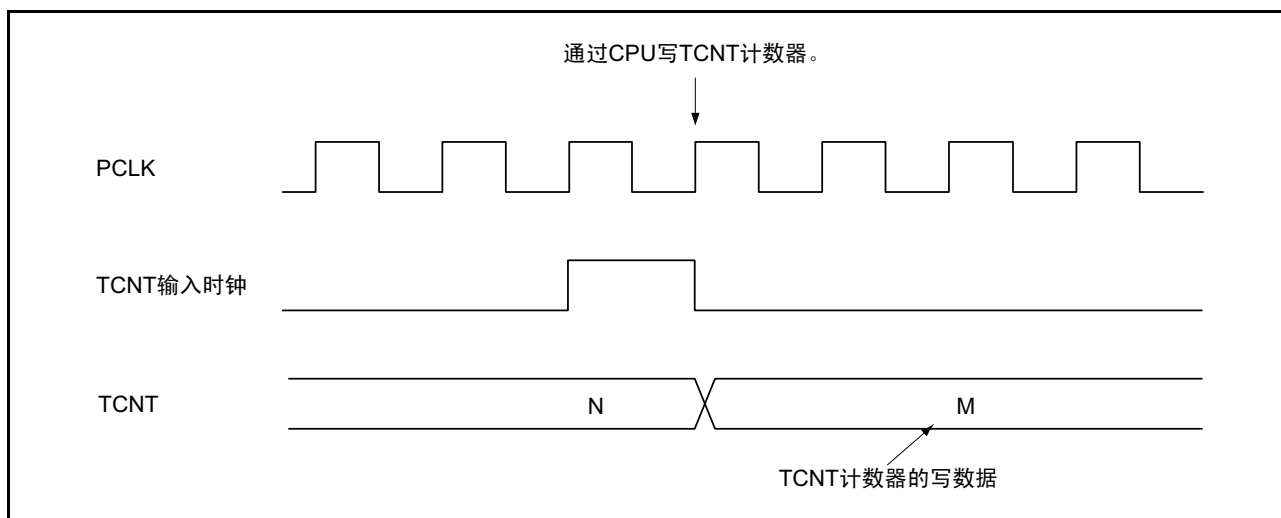


图 20.6 TCNT 计数器的写和递增计数的竞争

20.5.3 CKS[2:0] 位的改写

如果在看门狗定时器运行过程中改写 TCSR.CKS[2:0] 位，就可能无法正常进行递增计数。因此，必须在停止看门狗定时器（将 TCSR.TME 位置“0”）后改写 CKS[2:0] 位。

### 20.5.4 看门狗定时器模式和间隔定时器模式的转换

如果在看门狗定时器运行过程中进行看门狗定时器模式和间隔定时器模式的转换，就可能无法正常运行。因此，必须在停止看门狗定时器（将 TCSR.TME 位置“0”）后转换定时器模式。

### 20.5.5 看门狗定时器模式的内部复位

如果在看门狗定时器模式中将 RSTCSR.RSTE 位置“0”，即使 TCNT 计数器发生上溢，LSI 内部也不被复位，但是看门狗定时器的 TCNT 计数器和 TCSR 寄存器被复位。

在 WDTOVF# 信号输出 Low 电平期间，不能写 TCNT 计数器、TCSR 寄存器和 RSTCSR 寄存器，也无法识别 RSTCSR.WOVF 标志的读操作。因此，要清除 WOVF 标志时，必须在 WDTOVF# 信号变为 High 电平后，在读 RSTCSR 寄存器后给 WOVF 标志写“0”。

### 20.5.6 通过 WDTOVF# 信号进行系统复位

如果将 WDTOVF# 信号输入到 RES# 引脚，就不能对 LSI 进行正常的初始化。不能将 WDTOVF# 信号输入到 RES# 引脚。在通过 WDTOVF# 信号对整个系统进行复位时，必须在如图 20.7 所示的电路上进行。

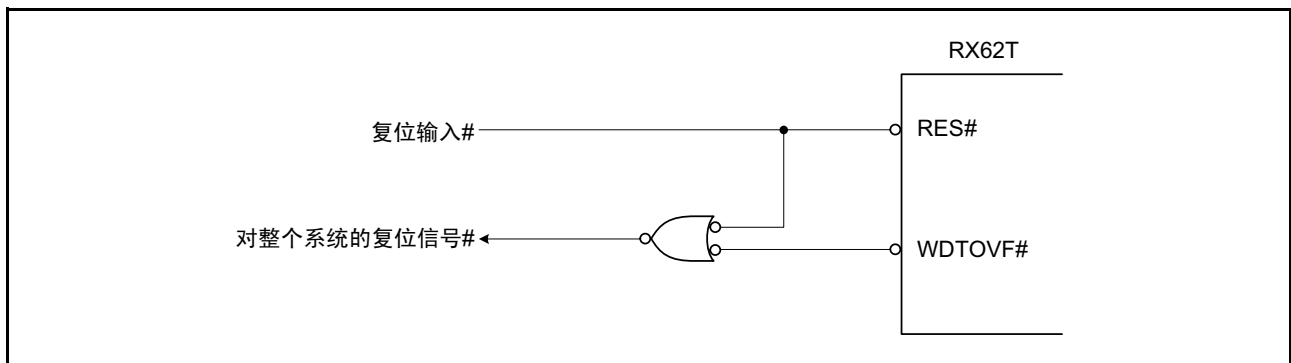


图 20.7 通过 WDTOVF# 信号进行系统复位的电路例子

### 20.5.7 向看门狗定时器模式和软件待机模式的转移

在用作看门狗定时器模式时，即使在将待机控制寄存器的软件待机位（SBYCR.SSBY）置“1”（在执行 WAIT 指令后转移到软件待机模式）的状态下执行 WAIT 指令，也不转移到软件待机模式，而转移到睡眠模式或者全模块时钟停止模式。

如果要转移到软件待机模式，就必须在停止看门狗定时器（将 TCSR.TME 位置“0”）后执行 WAIT 指令。在用作间隔定时器模式时，如果在将 SSBY 位置“1”的状态下执行 WAIT 指令，就转移到软件待机模式。详细内容请参照“9. 低功耗功能”。



## 21. 独立看门狗定时器 (IWDT)

独立看门狗定时器 (IWDT) 是为了检测程序失控而与看门狗定时器 (WDT) 独立使用的看门狗定时器。

IWDT 内置 14 位递减计数器，如果递减计数器的计数值发生下溢，就对系统进行复位。IWDT 还有刷新功能。

注. 在使用 IWDT 时，必须在发生下溢前进行刷新，详细内容请参照“21.3.3 刷新”。

### 21.1 概要

IWDT 的规格和框图分别如表 21.1 和图 21.1 所示。

表 21.1 IWDT 的规格

项目	内容
计数时钟	IWDTCLK、IWDTCLK/16、IWDTCLK/32、IWDTCLK/64、IWDTCLK/128、IWDTCLK/256
计数	通过 14 位递减计数器进行递减计数。
计数开始条件	通过刷新递减计数器（在给 IWDTRR 寄存器写“00h”后写“FFh”）开始计数。
计数停止条件	复位（递减计数器或者寄存器返回到初始值） 发生下溢时
IWDT 复位输出源	递减计数器发生下溢时
读 IWDT 计数器	能通过读 IWDTSR 寄存器，读递减计数器的计数值。

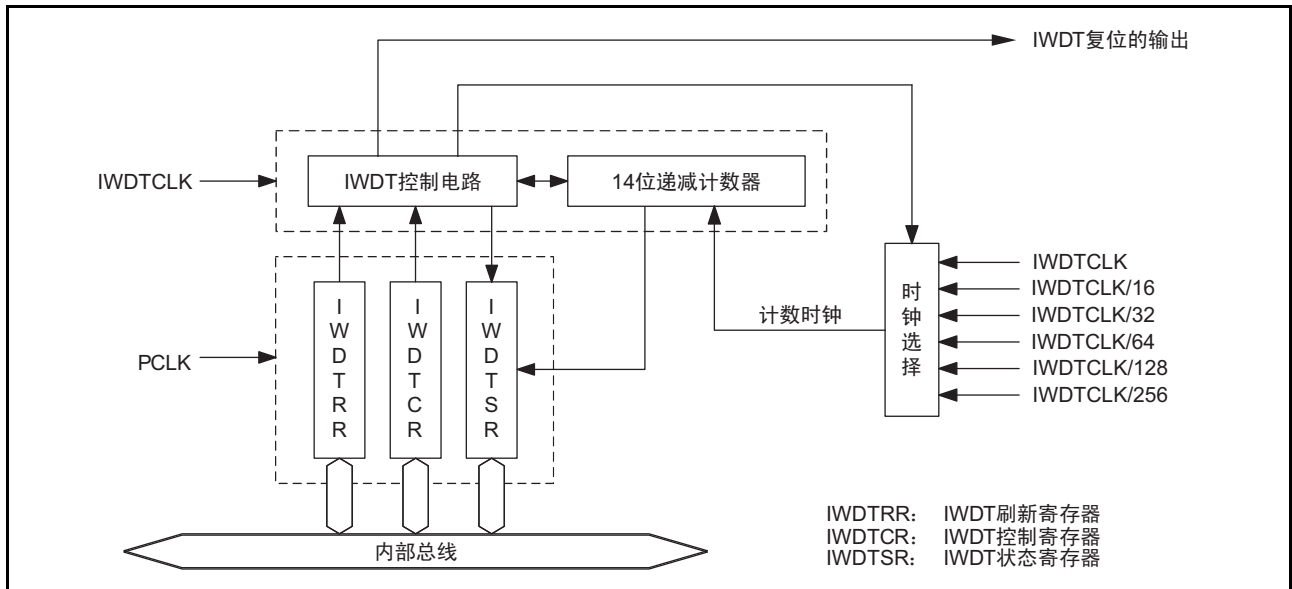


图 21.1 IWDT 的框图

## 21.2 寄存器说明

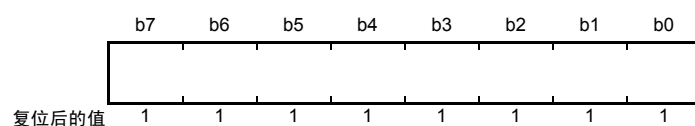
IWDT 的寄存器一览表如表 21.2 所示。

表 21.2 IWDT 的寄存器一览表

寄存器名	符号	复位后的值	地址	存取长度
IWDT 刷新寄存器	IWDTRR	FFh	0008 8030h	8
IWDT 控制寄存器	IWDTCR	3303h	0008 8032h	16
IWDT 状态寄存器	IWDTSR	0000h	0008 8034h	16

### 21.2.1 IWDT 刷新寄存器 (IWDTRR)

地址 0008 8030h



IWDTRR 寄存器是刷新 IWDT 递减计数器的寄存器。

通过在给 IWDTRR 寄存器写“00h”后写“FFh”（刷新），刷新 IWDT 的递减计数器。一旦递减计数器被刷新，就从 IWDTCR 控制寄存器（IWDTCR）的 TOPS[1:0] 位设定的值开始递减计数。

通过复位解除后的第 1 次刷新，从 IWDTCR.TOPS[1:0] 位设定的值开始递减计数。

如果在写“00h”后写“FFh”以外的值，写“00h”就无效。要将刷新设定为有效时，必须再次在写“00h”后写“FFh”。

如果写“00h”，读取值就为“00h”；如果写“00h”以外的值，读取值就总是为“FFh”。

## 21.2.2 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
—	—	—	—	—	—	—	—	CKS[3:0]			—	—	TOPS[1:0]		
0	0	1	1	0	0	1	1	0	0	0	0	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	读写值都为“0”。	R/W
b7-b4	CKS[3:0]	时钟选择位	b7 b4 0 0 — : IWDTCLK 0 1 0 0 : IWDTCLK/16 0 1 0 1 : IWDTCLK/32 0 1 1 0 : IWDTCLK/64 0 1 1 1 : IWDTCLK/128 1 — — : IWDTCLK/256	R/W
b9-b8	—	保留位	读写值都为“1”。	R/W
b11-b10	—	保留位	读写值都为“0”。	R/W
b13-b12	—	保留位	读写值都为“1”。	R/W
b15-b14	—	保留位	读写值都为“0”。	R/W

IWDTCR 寄存器是设定递减计数器发生下溢前的超时以及计数时钟的寄存器。

在从解除复位后到第 1 次刷新的期间，只能写 1 次 IWDTCR 寄存器。因为 IWDTCR 寄存器的写操作被锁定，所以刷新（开始计数）后的写操作和第 2 次以后的 IWDTCR 寄存器写操作无效。

通过 IWDT 的复位源而不能通过其他复位源来解除 IWDTCR 寄存器的锁定，详细内容请参照“6. 复位”。

## TOPS[1:0] 位（超时选择位）

以 CKS[3:0] 位设定的计数时钟为 1 个周期，这些位从 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前的超时。

刷新后到下溢前的时间（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、IWDTCLK 数的关系如表 21.3 所示。

表 21.3 超时设定表

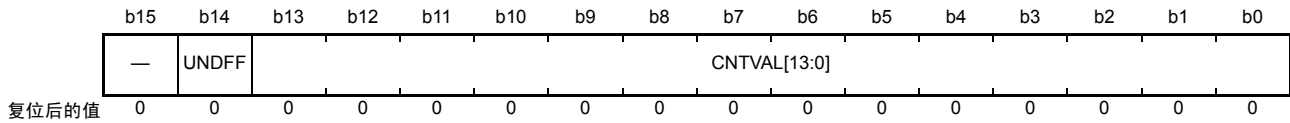
CKS[3:0]				TOPS[1:0]		计数时钟	超时 (周期数)	IWDTCLK 数
0	0	—	—	0	0	IWDTCLK	1024	1024
				0	1		4096	4096
				1	0		8192	8192
				1	1		16384	16384
0	1	0	0	0	0	IWDTCLK/16	1024	16384
				0	1		4096	65536
				1	0		8192	131072
				1	1		16384	262144
0	1	0	1	0	0	IWDTCLK/32	1024	32768
				0	1		4096	131072
				1	0		8192	262144
				1	1		16384	524288
0	1	1	0	0	0	IWDTCLK/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
0	1	1	1	0	0	IWDTCLK/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
1	—	—	—	0	0	IWDTCLK/256	1024	262144
				0	1		4096	1048576
				1	0		8192	2097152
				1	1		16384	4194304

#### CKS[3:0] 位 (时钟选择位)

从 IWDTCLK、IWDTCLK/16、IWDTCLK/32、IWDTCLK/64、IWDTCLK/128 和 IWDTCLK/256 中选择递减计数器的计数时钟。能通过和 TOPS[1:0] 位的设定组合，将 IWDT 的计数时间设定在 IWDTCLK 的 1024 ~ 4194304 个时钟之间。

### 21.2.3 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]	递减计数器位	递减计数器的计数值	R
b14	UNDFE	下溢标志	1: 下溢 0: 无下溢	R/W
b15	—	保留位	读写值都为“0”。	R/W

IWDTSR 寄存器是确认递减计数器的计数值和下溢发生状态的寄存器。

通过 IWDT 的复位源而不能通过其他复位源对 IWDTSR 寄存器进行初始化，详细内容请参照“6. 复位”。

#### CNTVAL[13:0] 位 (递减计数器位)

这些位能确认递减计数器的计数值。

#### 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，表示未发生下溢。

要清除 UNDFE 标志的值时，必须给此标志写“0”，写“1”无效。

### 21.3 运行说明

#### 21.3.1 递减计数器的计数

在复位解除后，给 IWDTCR 寄存器设定计数时钟和超时，然后通过刷新将 IWDTCR.TOPS[1:0] 位设定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递减计数器被刷新，就在每次刷新时重新设定计数器的值并且继续递减计数。在此期间，IWDT 不输出复位。但是，如果因程序失控而无法刷新递减计数器并且递减计数器发生下溢，IWDT 就输出复位。

在输出复位后，递减计数器保持初始状态 (ALL“0”)。在从复位返回后，通过刷新开始递减计数。递减计数器的计数运行例子如图 21.2 所示。

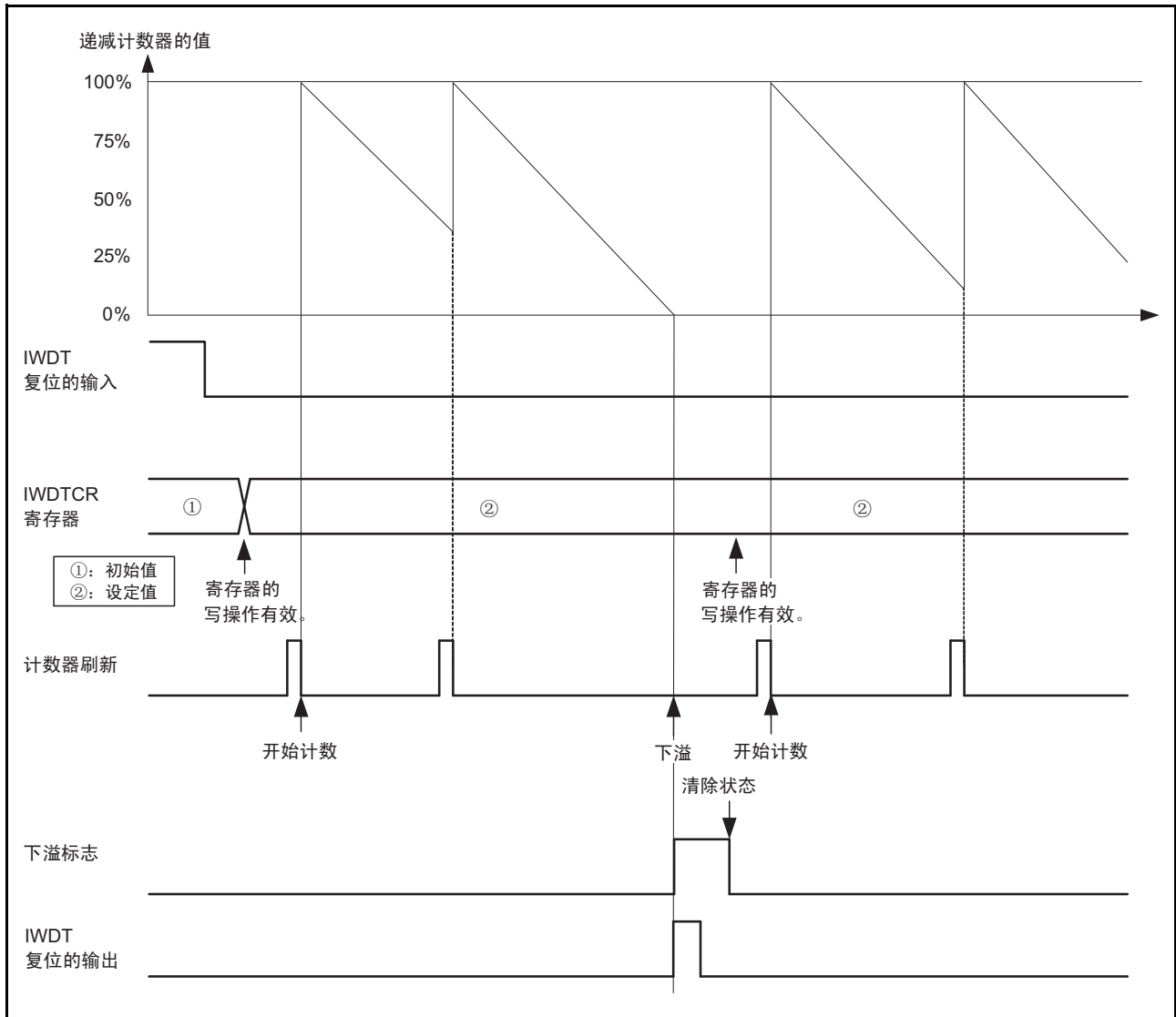


图 21.2 递减计数器的计数运行例子

### 21.3.2 IWDT 控制寄存器的写控制

只能在复位解除后写 1 次 IWDT 控制寄存器 (IWDTCR)。

如果写 IWDTCR 寄存器，IWDT 内部的寄存器锁定信号就变为“1”，锁定以后的 IWDTCR 寄存器的写操作。通过 IWDT 的复位源而不能通过其他复位源来解除锁定，详细内容请参照“6. 复位”。

IWDTCR 寄存器的写控制波形如图 21.3 所示。

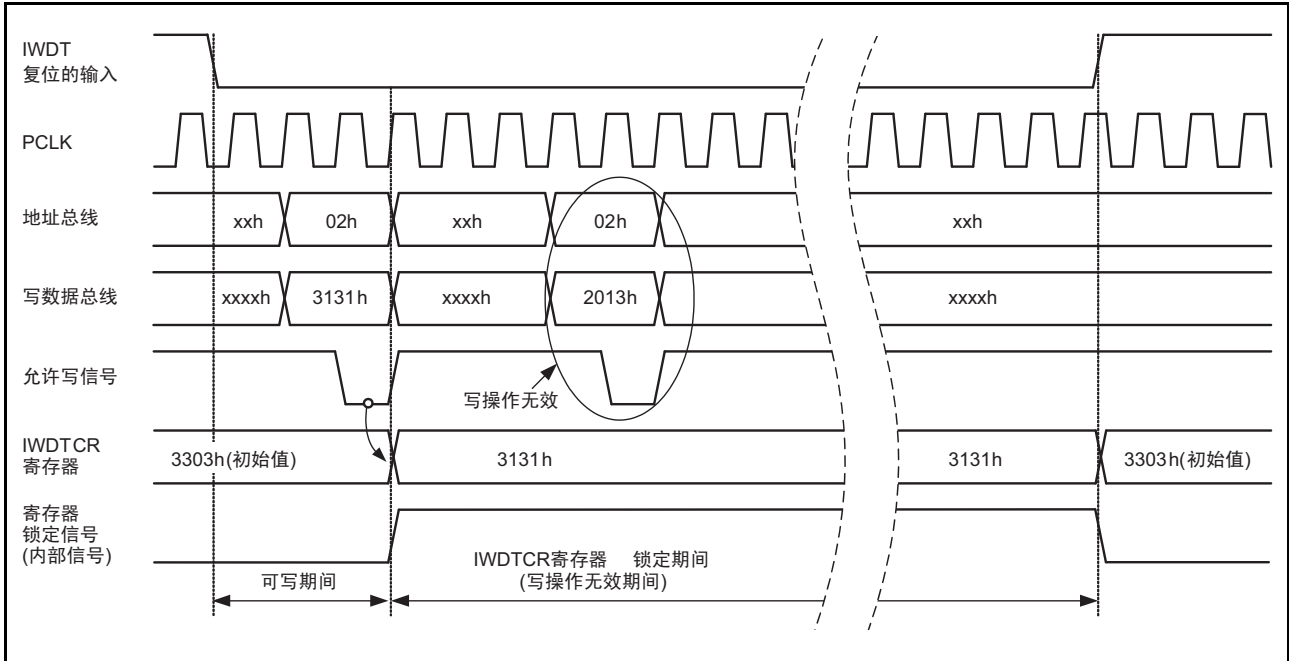


图 21.3 IWDTCR 寄存器的写控制波形

### 21.3.3 刷新

要使 IWDT 开始运行 (开始递减计数) 以及刷新递减计数器时，必须按照 00h→FFh 的顺序写 IWDT 的刷新寄存器 (IWDTRR)，否则写操作无效。因此，为了能正常地进行刷新，必须再次按照 00h→FFh 的顺序写 IWDTRR 寄存器。

虽然 00h→00h 的写操作无效，但是之后通过写“FFh”能使 00h→FFh 的写操作变为有效，因此 00h→00h→FFh 的写操作也有效。即使第 1 次写“00h”以外的值，之后 00h→FFh 的写操作也同样有效。

**【刷新无效的写操作例子】**

- 23h (不是“00h”) →FFh
- 00h→54h (不是“FFh”)
- 00h→AAh (不是“FFh”) →FFh

在给 IWDTRR 寄存器写“FFh”后，递减计数器的刷新最多需要 4 个计数周期 (1 个周期的 IWDTCLK 数因时钟选择位 (IWDTCR.CKS[3:0]) 的设定值而不同)。因此，在计数器发生下溢的 4 个计数时钟前，必须给 IWDTRR 寄存器写完“FFh”。能通过递减计数器的位 (IWDSR.CNTVAL[13:0]) 确认计数器的值。

计数时钟为 IWDTCLK 时的刷新波形如图 21.4 所示。

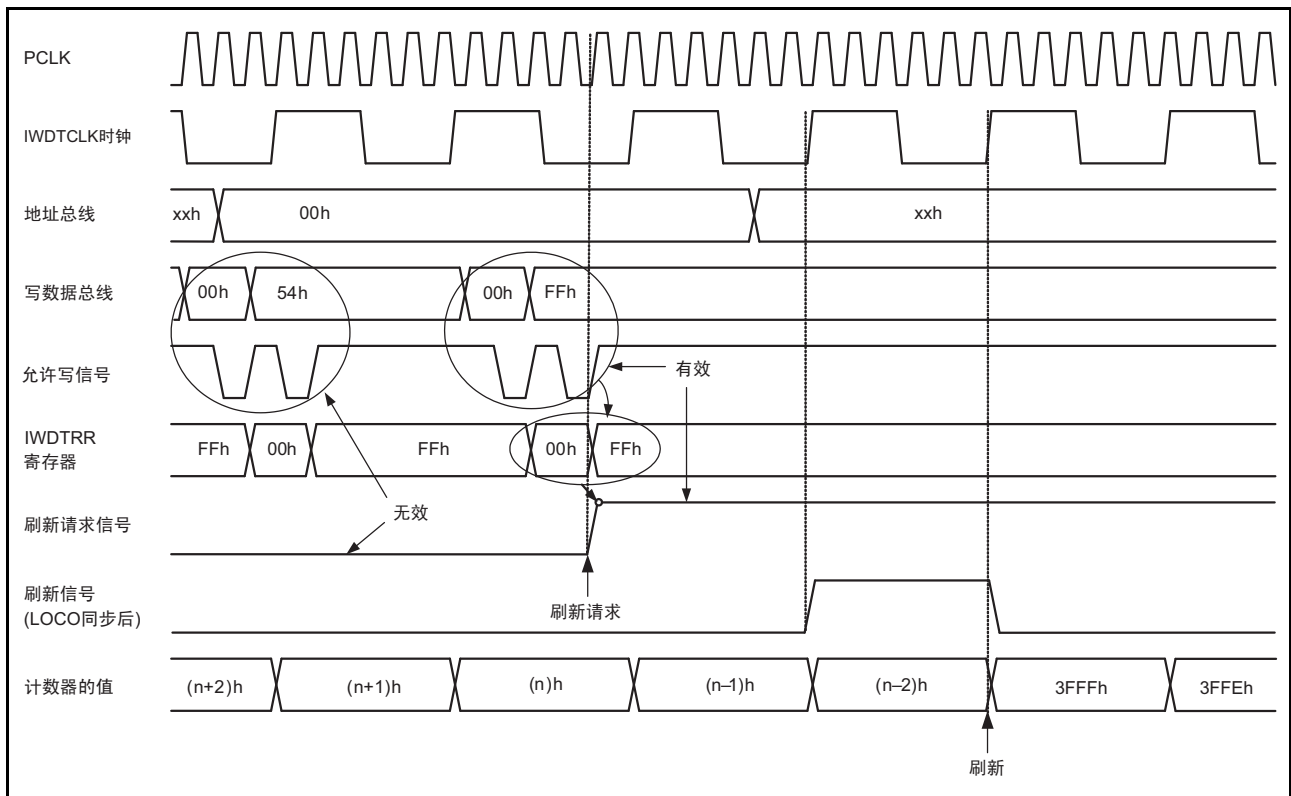


图 21.4 刷新波形 (IWDTCR.CKS[3:0]=0000b, IWDTCR.TOPS[1:0]=11b)

### 21.3.4 状态标志

下溢标志 (IWDTSR.UNDF) 保持 IWDT 输出复位时的复位源。

在解除复位后, 能通过读 IWDTSR.UNDF 标志, 确认复位源的发生状态。

要清除 UNDF 标志的值时, 必须写“0”。写“1”无效。

即使不将 UNDF 标志置“0”, 也不影响运行。在不将 UNDF 标志置“0”的情况下, 在下次 IWDT 输出复位时清除旧的复位源并且写新的复位源。

## 21.4 使用时的注意事项

### 21.4.1 向低功耗功能转移时的限制事项

通过写 IWDTCR 寄存器或者刷新 IWDTCR 寄存器, IWDT 进入使用状态。

当 IWDT 处于使用状态时, 即使在将 SBYCR.SSBY 位置“1”后执行 WAIT 指令, 也不转移到软件待机模式而转移到睡眠模式或者全模块时钟停止模式。

通过 IWDT 的复位源而不能通过其他复位源来解除 IWDT 的使用状态, 详细内容请参照“6. 复位”。



## 22. 串行通信接口 (SC1b)

RX62T 群内置 3 个独立通道的串行通信接口 (SCI: Serial Communication Interface)。

SCI 能进行异步串行通信和时钟同步串行通信。

在异步模式中, 能和 Universal Asynchronous Receiver/Transmitter (UART) 或者 Asynchronous Communication Interface Adapter (ACIA) 等标准异步通信 LSI 进行串行通信。

另外, 作为异步模式的扩展功能, 对应符合 ISO/IEC 7816-3 (Identification Card) 规格的智能卡 (IC 卡) 接口 (SMCI)。

### 22.1 概要

SCI 的规格如表 22.1 所示。

SCI0 ~ SCI2 的框图如图 22.1 所示所示。

表 22.1 SCI 的规格

项目		内容
串行通信方式		<ul style="list-style-type: none"> <li>• 异步</li> <li>• 时钟同步</li> <li>• 智能卡接口</li> </ul>
传送率		能通过内部波特率发生器设定任意的位速率。
全双工通信		发送部: 能通过双缓冲结构进行连续的发送。 接收部: 能通过双缓冲结构进行连续的接收。
输入 / 输出引脚		参照表 22.2。
数据传送		可选择 LSB first 或者 MSB first。
中断源		发送结束、发送数据空、接收数据满、接收错误
低功耗功能		各通道能设定为模块停止状态。
异步模式	数据长度	7 位 / 8 位
	发送停止位	1 位 / 2 位
	奇偶校验功能	偶校验 / 奇校验 / 无奇偶校验
	接收错误检测功能	奇偶校验错误、溢出错误、帧错误
	中止的检测	在发生帧错误时, 能通过直接读 RXDn (n=0 ~ 2) 引脚的电平检测中止。
	时钟源	可选择内部时钟或者外部时钟。
	多处理器通信功能	多个处理器之间的串行通信功能
	噪声消除功能	能消除 RXDn (n=0 ~ 2) 引脚输入的噪声。
时钟同步模式	数据长度	8 位
	接收错误的检测	溢出错误
智能卡接口模式	错误处理	如果在接收时检测到奇偶校验错误, 就自动发送错误信号。
		如果在发送时接收到错误信号, 就自动重新发送数据。
	数据类型	支持正向协议 / 反向协议

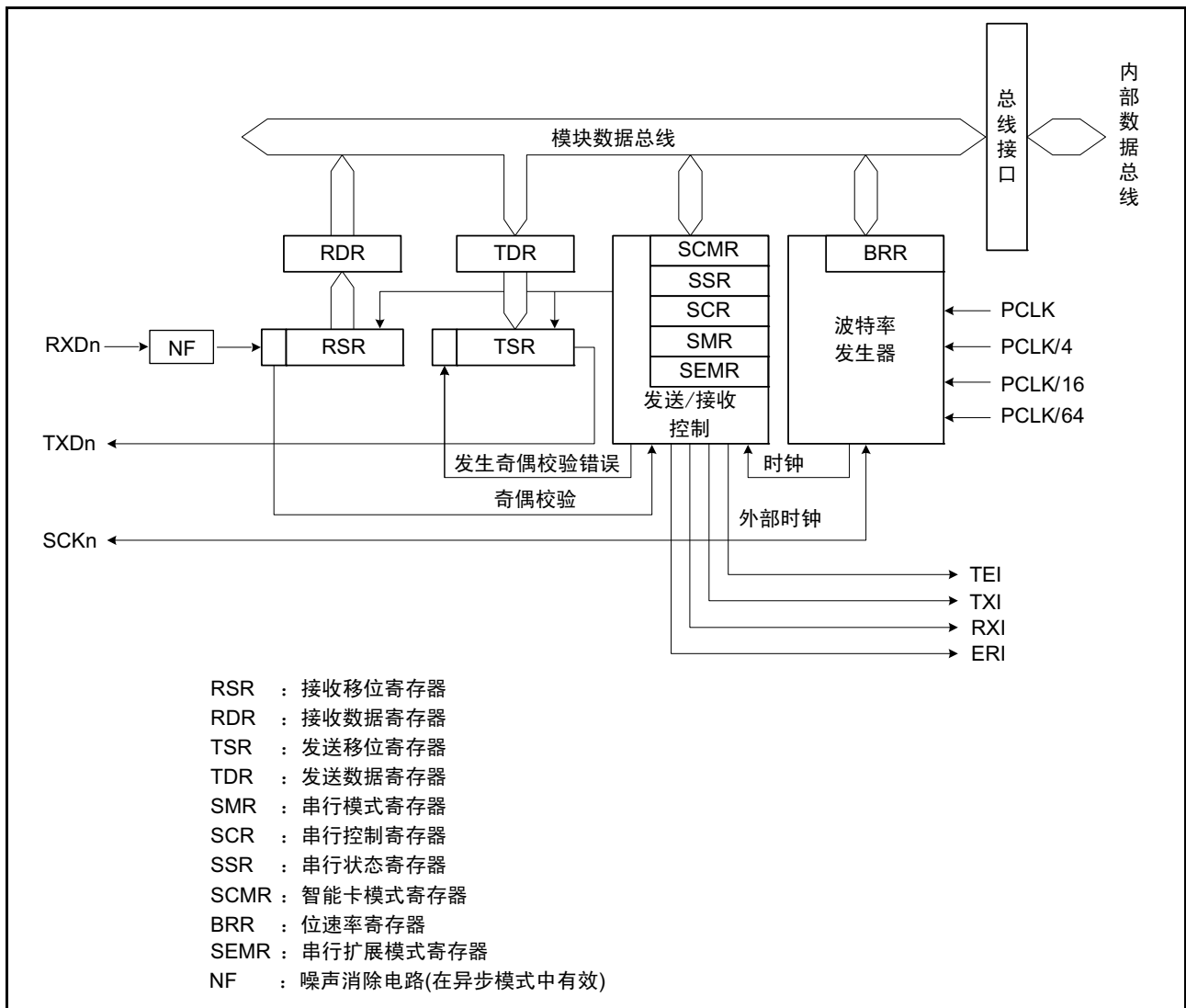


图 22.1 SCI0 ~ SCI2 的框图

SCI/SMCI 使用的输入 / 输出引脚如表 22.2 所示。

表 22.2 SCI/SMCI 的输入 / 输出引脚

通道	引脚名	输入 / 输出	功能
SCI0/SMCI0	SCK0	输入 / 输出	SCI0/SMCI0 的时钟输入 / 输出引脚
	RXD0	输入	SCI0/SMCI0 的接收数据输入引脚
	TXD0	输出	SCI0/SMCI0 的发送数据输出引脚
SCI1/SMCI1	SCK1	输入 / 输出	SCI1/SMCI1 的时钟输入 / 输出引脚
	RXD1	输入	SCI1/SMCI1 的接收数据输入引脚
	TXD1	输出	SCI1/SMCI1 的发送数据输出引脚
SCI2/SMCI2	SCK2	输入 / 输出	SCI2/SMCI2 的时钟输入 / 输出引脚
	RXD2	输入	SCI2/SMCI2 的接收数据输入引脚
	TXD2	输出	SCI2/SMCI2 的发送数据输出引脚

## 22.2 串行通信接口模式

### 22.2.1 寄存器说明

SCI 的寄存器一览表如表 22.3 所示。

表 22.3 SCI 的寄存器一览表

通道	寄存器名	符号	复位后的值	地址	存取长度
SCI0	串行模式寄存器	SMR	00h	0008 8240h	8
	位速率寄存器	BRR	FFh	0008 8241h	8
	串行控制寄存器	SCR	00h	0008 8242h	8
	发送数据寄存器	TDR	FFh	0008 8243h	8
	串行状态寄存器	SSR	84h	0008 8244h	8
	接收数据寄存器	RDR	00h	0008 8245h	8
	智能卡模式寄存器	SCMR	F2h	0008 8246h	8
	串行扩展模式寄存器	SEMR	00h	0008 8247h	8
SCI1	串行模式寄存器	SMR	00h	0008 8248h	8
	位速率寄存器	BRR	FFh	0008 8249h	8
	串行控制寄存器	SCR	00h	0008 824Ah	8
	发送数据寄存器	TDR	FFh	0008 824Bh	8
	串行状态寄存器	SSR	84h	0008 824Ch	8
	接收数据寄存器	RDR	00h	0008 824Dh	8
	智能卡模式寄存器	SCMR	F2h	0008 824Eh	8
	串行扩展模式寄存器	SEMR	00h	0008 824Fh	8
SCI2	串行模式寄存器	SMR	00h	0008 8250h	8
	位速率寄存器	BRR	FFh	0008 8251h	8
	串行控制寄存器	SCR	00h	0008 8252h	8
	发送数据寄存器	TDR	FFh	0008 8253h	8
	串行状态寄存器	SSR	84h	0008 8254h	8
	接收数据寄存器	RDR	00h	0008 8255h	8
	智能卡模式寄存器	SCMR	F2h	0008 8256h	8
	串行扩展模式寄存器	SEMR	00h	0008 8257h	8

#### 22.2.1.1 接收移位寄存器 (RSR)

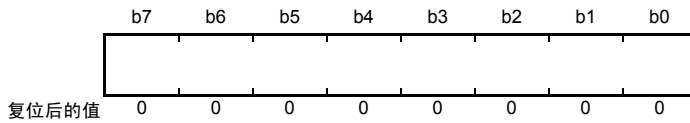
RSR 寄存器是将 RXDn 引脚输入的串行数据转换为并行数据的接收移位寄存器。

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。

### 22.2.1.2 接收数据寄存器 (RDR)

地址 SCI0.RDR 0008 8245h、SCI1.RDR 0008 824Dh、SCI2.RDR 0008 8255h



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。

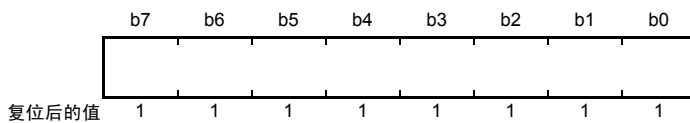
因为 RSR 寄存器和 RDR 寄存器是双缓冲结构，所以能连续接收。

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。

### 22.2.1.3 发送数据寄存器 (TDR)

地址 SCI0.TDR 0008 8243h、SCI1.TDR 0008 824Bh、SCI2.TDR 0008 8253h



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。

### 22.2.1.4 发送移位寄存器 (TSR)

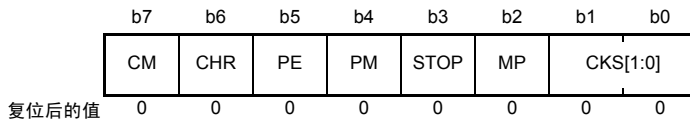
TSR 寄存器是发送串行数据的移位寄存器。

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。

## 22.2.1.5 串行模式寄存器 (SMR)

地址 SCI0.SMR 0008 8240h、SCI1.SMR 0008 8248h、SCI2.SMR 0008 8250h



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	停止位长选择位	(只在异步模式中有效) 0: 1 个停止位 1: 2 个停止位	R/W (注4)
b4	PM	奇偶校验模式位	(只在异步模式中并且 PE 位为“1”时有效) 0: 偶校验 1: 奇校验	R/W (注4)
b5	PE	奇偶校验允许位	(只在异步模式中有效) • 发送时 0: 无奇偶校验位 1: 附加奇偶校验位 • 接收时 0: 不进行奇偶校验 1: 进行奇偶校验	R/W (注4)
b6	CHR	字符长位	(只在异步模式中有效) 0: 数据长度为 8 位 (注2) 1: 数据长度为 7 位 (注3)	R/W (注4)
b7	CM	通信模式位	0: 异步通信模式 1: 时钟同步通信模式	R/W (注4)

注 1. n 为设定值的 10 进制数, 表示“22.2.1.9 位速率寄存器 (BRR)”中的 n 的值。

注 2. 在时钟同步模式中, 与设定值无关, 数据长度为 8 位。

注 3. 固定为 LSB first, 在发送时不发送 TDR 寄存器的 MSB (b7)。

注 4. 只能在 SCR.TE 位和 SCR.RE 位都为“0” (禁止串行发送和串行接收) 时写这些位。

SMR 寄存器选择通信格式以及内部波特率发生器的时钟源。

## CKS[1:0] 位 (时钟选择位)

这些位选择内部波特率发生器的时钟源。

有关 CKS[1:0] 位的设定值和波特率的关系, 请参照“22.2.1.9 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 (停止位长选择位)**

此位选择发送数据的停止位长。

在接收时，与此位的设定无关，只检查停止位的第 1 位，当第 2 位为“0”时，将第 2 位视为下一个发送帧的起始位。

**PM 位 (奇偶校验模式位)**

此位选择发送和接收时的奇偶校验 (偶校验或者奇校验)。

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 (字符长位)**

此位选择发送 / 接收数据的数据长度。

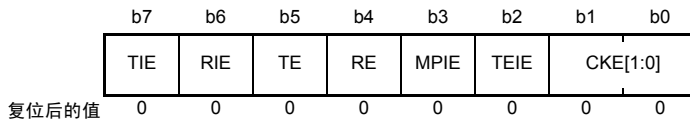
在时钟同步模式中，数据长度为 8 位。

**CM 位 (通信模式位)**

此位选择异步模式或者时钟同步模式。

## 22.2.1.6 串行控制寄存器 (SCR)

地址 SCI0.SCR 0008 8242h、SCI1.SCR 0008 824Ah、SCI2.SCR 0008 8252h



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"> <li>• 异步模式</li> <li>b1 b0</li> <li>0 0: 内部波特率发生器 SCKn 引脚能用作输入 / 输出端口。</li> <li>0 1: 内部波特率发生器 从 SCKn 引脚输出频率与位速率相同的时钟。</li> <li>1 0: 外部时钟 必须从 SCKn 引脚输入频率为 16 倍位速率的时钟。当 SEMR.ABCS 位为“1”时，必须输入频率为 8 倍位速率的时钟。</li> <li>1 1: 外部时钟 必须从 SCKn 引脚输入频率为 16 倍位速率的时钟。当 SEMR.ABCS 位为“1”时，必须输入频率为 8 倍位速率的时钟。</li> </ul> <ul style="list-style-type: none"> <li>• 时钟同步模式</li> <li>b1 b0</li> <li>0 0: 内部时钟 SCKn 引脚为时钟的输出引脚。</li> <li>0 1: 内部时钟 SCKn 引脚为时钟的输出引脚。</li> <li>1 0: 外部时钟 SCKn 引脚为时钟的输入引脚。</li> <li>1 1: 外部时钟 SCKn 引脚为时钟的输入引脚。</li> </ul>	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: 正常的接收运行 1: 多处理器接收运行	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

注 1. 只能在 TE 位和 RE 位都为“0”时写这些位。

注 2. 只能在 SMR.CM 位为“1”，TE 位和 RE 位都为“0”时写“1”。一旦将 TE 位或者 RE 位置“1”，就只能在 TE 位和 RE 位都为“0”时写此位。在 SMR.CM 位为“0”时，能以任意的时序进行写操作。

SCR 寄存器是控制发送 / 接收以及选择发送 / 接收时钟源的寄存器。

#### CKE[1:0] 位 (时钟允许位)

这些位选择时钟源和 SCKn 引脚的功能。

#### TEIE 位 (发送结束中断允许位)

此位允许或者禁止 TEI 中断。

要禁止 TEI 中断时, 将 TEIE 位置“0”。

#### MPIE 位 (多处理器中断允许位)

如果将 MPIE 位置“1”, 就不将多处理器位为“0” (SSR.MPB 位 =0) 的接收数据从 RSR 寄存器传送到 RDR 寄存器, 并且不进行接收错误的检测, 以及 SSR.ORER、SSR.FFR 的各状态标志也不变为“1”。如果接收多处理器位为“1”的数据, SSR.MPB 位就变为“1”, MPIE 位自动变为“0”并且返回到通常的接收运行。允许 RXI、ERI 中断请求 (SCR.RIE 位为“1”时) 以及允许 SSR..ORER 标志和 SSR.FER 标志的置位 (“1”)。详细内容请参照“22.2.3 多处理器通信功能”。

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 中断和 ERI 中断。

要禁止 RXI 中断时, 将 RIE 位置“0”。

如果在从 SSR.ORER 标志、SSR.FER 标志和 SSR.PER 标志读“1”后将这些标志位置“0”, 或者将 RIE 位置“0”, ERI 中断请求信号就消失。

#### TIE 位 (发送中断允许位)

此位允许或者禁止通知 TXI 中断。

要禁止 TXI 中断时, 将 TIE 位置“0”。



## 22.2.1.7 串行状态寄存器 (SSR)

地址 SCI0.SSR 0008 8244h、SCI1.SSR 0008 824Ch、SCI2.SSR 0008 8254h

b7	b6	b5	b4	b3	b2	b1	b0
TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
1	0	0	0	0	1	0	0

复位后的值

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)
b6	RDRF	接收数据满标志	0: 当从 RDR 寄存器传送数据时 1: 当接收正常结束并且将数据从 RSR 寄存器传送到 RDR 寄存器时	R/(W) (注 2)
b7	TDRE	发送数据空标志	0: 当将数据传送到 TDR 寄存器时 1: 当将数据从 TDR 寄存器传送到 TSR 寄存器时	R/(W) (注 2)

注 1. 为了清除标志, 只能在读“1”后写“0”。

注 2. 只能写“1”。

SSR 寄存器由 SCI 的状态标志以及发送 / 接收多处理器位构成。

## MPBT 位 (多处理器位的传送位)

此位设定给发送帧附加的多处理器位的值。

## 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 ]

- SCR.TE 位为“0” (禁止串行发送) 时
- 在发送字符的最后一位时未更新 TDR 寄存器时

[ 为“0”的条件 ]

- 给 TDR 寄存器写发送数据时

在通过给 TDR 寄存器写发送数据将 TEND 标志置“0”时, 必须读 TEND 标志, 确认 TEND 标志已变为“0”。

**PER 标志 (奇偶校验错误标志)**

此标志表示在异步模式中接收的数据发生了奇偶校验错误。

[为“1”的条件]

- 在接收时检测到奇偶校验错误时  
将发生奇偶校验错误时的接收数据传送到RDR寄存器，但是不产生RXI中断请求。在PER标志为“1”的状态下，以后的接收数据不传送到RDR寄存器。

[为“0”的条件]

- 读“1”后写“0”（必须在写“0”后确认PER标志已变为“0”）时  
即使将SCR.RE位置“0”（禁止串行接收），PER标志也不受影响而保持原来的状态。

**FER 标志 (帧错误标志)**

此标志表示在异步模式中接收的数据发生了帧错误并且异常结束。

[为“1”的条件]

- 停止位为“0”时  
在2个停止位模式中，只判断第1个停止位是否为“1”而不检查第2个停止位。将发生帧错误时的接收数据传送到RDR寄存器，但是不产生RXI中断请求。在FER标志为“1”的状态下，以后的接收数据不传送到RDR寄存器。

[为“0”的条件]

- 读“1”后写“0”（必须在写“0”后确认FER标志已变为“0”）时  
即使将SCR.RE位置“0”，FER标志也不受影响而保持原来的状态。

**ORER 标志 (溢出错误标志)**

此标志表示接收的数据发生了溢出错误。

[为“1”的条件]

- 不读RDR寄存器的接收数据就接收到下一个数据时  
RDR寄存器保持发生溢出错误前的接收数据而丢失后面的接收数据。在ORER标志为“1”的状态下，不能继续进行以后的串行接收。在时钟同步模式中，也不能继续进行串行发送。

[为“0”的条件]

- 读“1”后写“0”（必须在写“0”后确认ORER标志已变为“0”）时  
即使将SCR.RE位置“0”，ORER标志也不受影响而保持原来的状态。

**RDRF 标志 (接收数据满标志)**

此标志表示在RDR寄存器中是否有接收数据。

[为“1”的条件]

- 接收正常结束并且将数据从RSR寄存器传送到RDR寄存器时

[为“0”的条件]

- 从RDR寄存器传送数据时

**TDRE 标志 (发送数据空标志)**

此标志表示在TDR寄存器中是否有发送数据。

[为“1”的条件]

- 将数据从TDR寄存器传送到TSR寄存器时

[为“0”的条件]

- 将数据传送到TDR寄存器时

## 22.2.1.8 智能卡模式寄存器 (SCMR)

地址 SCI0.SCMR 0008 8246h、SCI1.SCMR 0008 824Eh、SCI2.SCMR 0008 8256h

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	智能卡数据取反位	0: 原样发送 TDR 寄存器的内容, 并且将接收数据原样保存到 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	位序选择位	0: 以 LSB first 进行发送和接收 1: 以 MSB first 进行发送和接收	R/W (注1)
b6-b4	—	保留位	读写值都为“1”。	R/W
b7	BCP2	基本时钟脉冲位 2	通过和 SMR.BCP[1:0] 位组合进行选择。 SCMR.BCP2 位和 SMR.BCP[1:0] 位的设定值 BCP2 BCP1 BCP0 0 0 0: 93 个时钟 (S=93) (注2) 0 0 1: 128 个时钟 (S=128) (注2) 0 1 0: 186 个时钟 (S=186) (注2) 0 1 1: 512 个时钟 (S=512) (注2) 1 0 0: 32 个时钟 (S=32) (注2) (初始值) 1 0 1: 64 个时钟 (S=64) (注2) 1 1 0: 372 个时钟 (S=372) (注2) 1 1 1: 256 个时钟 (S=256) (注2)	R/W (注1)

注1. 只能在 SCR.TE 位为“0”、SCR.RE 位为“0”（禁止串行发送和串行接收）时写此位。

注2. S 表示“22.2.1.9 位速率寄存器 (BRR)”中的 S 的值。

SCMR 寄存器是选择智能卡接口模式及其格式的寄存器。

## SMIF 位 (智能卡接口模式选择位)

在以智能卡接口模式运行时, 将此位置“1”。

在以异步模式或者时钟同步模式运行时, 将此位置“0”。

## SINV 位 (智能卡数据取反位)

将发送 / 接收数据的逻辑电平取反。SINV 位不影响奇偶校验位的逻辑电平。如果要将奇偶校验位取反, 就必须将 SMR.PM 位取反。

## SDIR 位 (位序选择位)

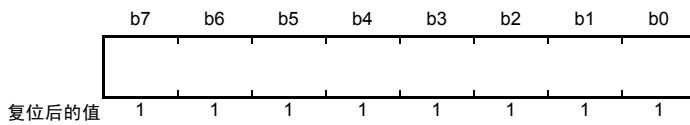
此位选择串行 / 并行转换的方向。

## BCP2 位 (基本时钟脉冲位 2)

在智能卡接口模式中, 通过和 SMR.BCP[1:0] 位组合, 选择 1 位传送期间的基本时钟数。

## 22.2.1.9 位速率寄存器 (BRR)

地址 SCI0.BRR 0008 8241h、SCI1.BRR 0008 8249h、SCI2.BRR 0008 8251h



BRR 寄存器是调整位速率的 8 位寄存器。

因为 SCI 各通道的波特率发生器独立运行，所以能设定不同的位速率。在一般的异步模式、时钟同步模式和智能卡接口模式中，BRR 寄存器的设定值 N 和位速率 B 的关系如表 22.4 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。

异步模式中的 BRR 寄存器值 N 的设定例子如表 22.5 所示，在各工作频率下能设定的最大位速率如表 22.6 所示；时钟同步模式中的 BRR 寄存器值 N 的设定例子如表 22.8 所示。输入外部时钟时的最大位速率如表 22.7 和表 22.9 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 22.5 所示的位速率的 2 倍。

表 22.4 BRR 寄存器的设定值 N 和位速率 B 的关系

模式	SEMR.ABCS位	BRR寄存器的设定值	误差
异步	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	误差(%) = $\left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	误差(%) = $\left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
时钟同步		$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	

B : 位速率 (bps)

N : 波特率发生器的 BRR 设定值 ( $0 \leq N \leq 255$ )

PCLK : 工作频率 (MHz)

n、S : 取决于下表中 SEMR 的设定值。

SEMR 寄存器的设定值	时钟源	n
CKS[1:0] 位		
0 0	PCLK 时钟	0
0 1	PCLK/4 时钟	1
1 0	PCLK/16 时钟	2
1 1	PCLK/64 时钟	3

SCMR 寄存器的设定值	SMR 寄存器的设定值	基本时钟	S
BCP2 位	BCP[1:0] 位		
0	0 0	93 个时钟	93
0	0 1	128 个时钟	128
0	1 0	186 个时钟	186
0	1 1	512 个时钟	512
1	0 0	32 个时钟	32
1	0 1	64 个时钟	64
1	1 0	372 个时钟	372
1	1 1	256 个时钟	256

表 22.5 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

位速率 (bps)	工作频率 PCLK (MHz)								
	12.288			14			16		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	2	217	0.08	2	248	-0.17	3	70	0.03
150	2	159	0.00	2	181	0.16	2	207	0.16
300	2	79	0.00	2	90	0.16	2	103	0.16
600	1	159	0.00	1	181	0.16	1	207	0.16
1200	1	79	0.00	1	90	0.16	1	103	0.16
2400	0	159	0.00	0	181	0.16	0	207	0.16
4800	0	79	0.00	0	90	0.16	0	103	0.16
9600	0	39	0.00	0	45	-0.93	0	51	0.16
19200	0	19	0.00	0	22	-0.93	0	25	0.16
31250	0	11	2.40	0	13	0.00	0	15	0.00
38400	0	9	0.00	—	—	—	0	12	0.16

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 22.5 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)											
	17.2032			18			19.6608			20		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	3	75	0.48	3	79	-0.12	3	86	0.31	3	88	-0.25
150	2	223	0.00	2	233	0.16	2	255	0.00	3	64	0.16
300	2	111	0.00	2	116	0.16	2	127	0.00	2	129	0.16
600	1	223	0.00	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	111	0.00	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	111	0.00	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	55	0.00	0	58	-0.69	0	63	0.00	0	64	0.16
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32	-1.36
31250	0	16	1.20	0	17	0.00	0	19	-1.70	0	19	0.00
38400	0	13	0.00	0	14	-2.34	0	15	0.00	0	15	1.73

位速率 (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 22.6 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	PCLK (MHz)	最大位速率 (bps)	n	N
8	250000	0	0	18	562500	0	0
9.8304	307200	0	0	19.6608	614400	0	0
10	312500	0	0	20	625000	0	0
12	375000	0	0	25	781250	0	0
12.288	384000	0	0	30	937500	0	0
14	437500	0	0	33	1031250	0	0
16	500000	0	0	50	1562500	0	0
17.2032	537600	0	0				

注. 如果将 SEMR.ABCS 位置“1”，位速率就变为原来的 2 倍。

表 22.7 输入外部时钟时的最大位速率 (异步模式) (1)

PCLK (MHz)	外部输入时钟 (MHz)	最大位速率 (bps)	PCLK (MHz)	外部输入时钟 (MHz)	最大位速率 (bps)
8	2.0000	125000	18	4.5000	281250
9.8304	2.4576	153600	19.6608	4.9152	307200
10	2.5000	156250	20	5.0000	312500
12	3.0000	187500	25	6.2500	390625
12.288	3.0720	192000	30	7.5000	468750
14	3.5000	218750	33	8.2500	515625
16	4.0000	250000	50	12.5000	781250
17.2032	4.3008	268800			

注. 这是 SEMR.ABCS 位为“0”时的例子。

表 22.7 输入外部时钟时的最大位速率 (异步模式) (2)

PCLK (MHz)	外部输入时钟 (MHz)	最大位速率 (bps)	PCLK (MHz)	外部输入时钟 (MHz)	最大位速率 (bps)
8	2.0000	250000	18	4.5000	562500
9.8304	2.4576	307200	19.6608	4.9152	614400
10	2.5000	312500	20	5.0000	625000
12	3.0000	375000	25	6.2500	781250
12.288	3.0720	384000	30	7.5000	937500
14	3.5000	437500	33	8.2500	1031250
16	4.0000	500000	50	12.5000	1562500
17.2032	4.3008	537600			

注. 这是 SEMR.ABCS 位为“1”时的例子。

表 22.8 对应位速率的 BRR 设定例子 (时钟同步模式)

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	—	—	—	—	—	—	—	—
2.5M			0	0 (注1)			0	1	—	—	0	2	—	—	0	4
5M							0	0 (注1)	—	—	—	—	—	—	—	—

空栏：不能设定。

—：能设定，但是会产生误差。

注 1. 不能进行连续的发送和接收。

表 22.9 输入外部时钟时的最大位速率 (时钟同步模式)

PCLK (MHz)	外部输入时钟 (MHz)	最大位速率 (bps)	PCLK (MHz)	外部输入时钟 (MHz)	最大位速率 (bps)
8	1.3333	1333333.3	20	3.3333	3333333.3
10	1.6667	1666666.7	25	4.1667	4166666.7
12	2.0000	2000000.0	30	5.0000	5000000.0
14	2.3333	2333333.3	33	5.5000	5500000.0
16	2.6667	2666666.7	50	8.3333	8333333.3
18	3.0000	3000000.0			



表 22.10 对应位速率的 BRR 设定例子 (在智能卡接口模式中, n 为“0”并且 S 为“372”的情况)

位速率 (bps)	工作频率 PCLK (MHz)											
	7.1424			10.00			10.7136			13.00		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
9600	0	0	0.00	0	1	30	0	1	25	0	1	8.99

位速率 (bps)	工作频率 PCLK (MHz)											
	14.2848			16.00			18.00			20.00		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2	6.66

位速率 (bps)	工作频率 PCLK (MHz)											
	25.00			30.00			33.00			50.00		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
9600	0	3	12.49	0	3	5.01	0	4	7.59	0	6	0.01

表 22.11 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	PCLK (MHz)	最大位速率 (bps)	n	N
10.00	13441	0	0	20.00	26882	0	0
10.7136	14400	0	0	25.00	33602	0	0
13.00	17473	0	0	30.00	40323	0	0
16.00	21505	0	0	33.00	44355	0	0
18.00	24194	0	0	50.00	67205	0	0

## 22.2.1.10 串行扩展模式寄存器 (SEMR)

地址 SCI0.SEMR 0008 8247h、SCI1.SEMR 0008 824Fh、SCI2.SEMR 0008 8257h

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	噪声消除功能选择位	(只在异步模式中有效) 0: RXDn 引脚输入的噪声消除功能无效 1: RXDn 引脚输入的噪声消除功能有效	R/W (注 1)
b7-b5	—	保留位	读写值都为“0”。	R/W

注. 只能在 SCR.TE 位为“0”、SCR.RE 位为“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 位时间的时钟以及 RXDn 引脚输入噪声消除功能的有效或者无效。

## ABCS 位（异步基本时钟选择位）

此位选择 1 位时间的基本时钟的脉冲数。

## NFEN 位（噪声消除功能选择位）

此位选择噪声消除功能的有效或者无效。如果设定为有效，就消除 RXDn 引脚输入的噪声。  
详细内容请操作“22.4 噪声消除功能”。

### 22.2.2 异步模式的运行

异步串行通信的一般数据格式如图 22.2 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, 如果检测到空闲 (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。

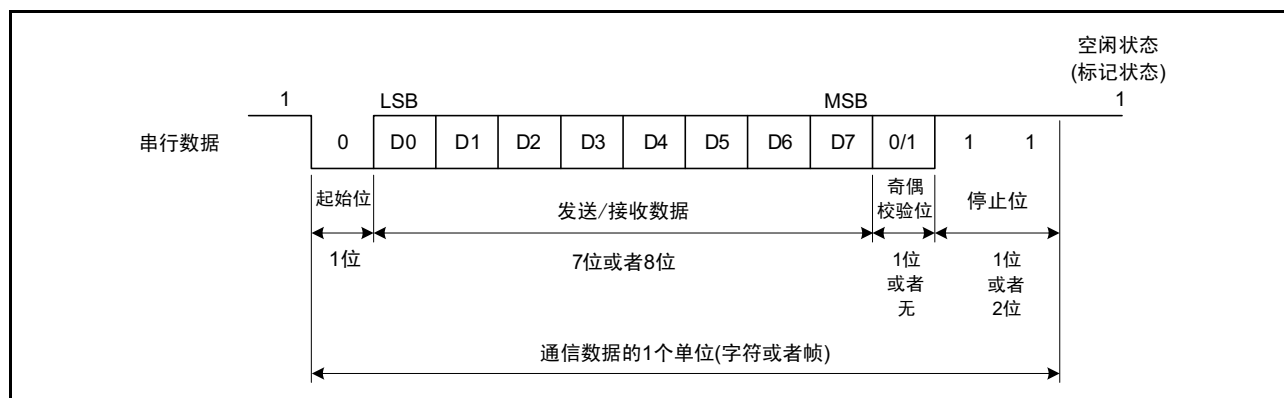


图 22.2 异步串行通信的数据格式 (8 位数据 / 有奇偶校验 / 2 个停止位的例子)

22.2.2.1 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 22.12 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“22.2.3 多处理器通信功能”。

表 22.12 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

## 22.2.2.2 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 22.3 所示，通过在第 8 个时钟（注 1）的上升沿对接收数据进行采样，在各位的中央取数据。因此，能用表达式（1）表示异步模式的接收容限。

$$M = \left| \left( 0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100[\%] \quad \dots \text{表达式(1)}$$

M : 接收容限

N : 对应时钟的位速率比

（当 SEMR.ABCS 位为“0”时，N=16；当 ABCS 位为“1”时，N=8）

D : 时钟的占空比（D=0.5 ~ 1.0）

L : 帧长（L=9 ~ 12）

F : 时钟频率的偏差绝对值

假设表达式（1）中的 F（时钟频率的偏差绝对值）为“0”，D（时钟的占空比）为“0.5”，则

$$M = \{ 0.5 - 1/(2 \times 16) \} \times 100[\%] = 46.875\%$$

但是，此值毕竟是计算值，在进行系统设计时，必须留出 20 ~ 30% 的容限。

注 1. 这是 SEMR.ABCS 位为“0”时的例子。当 ABCS 位为“1”时，基本时钟的频率为位速率的 8 倍，在基本时钟的第 4 个上升沿对接收数据进行采样。

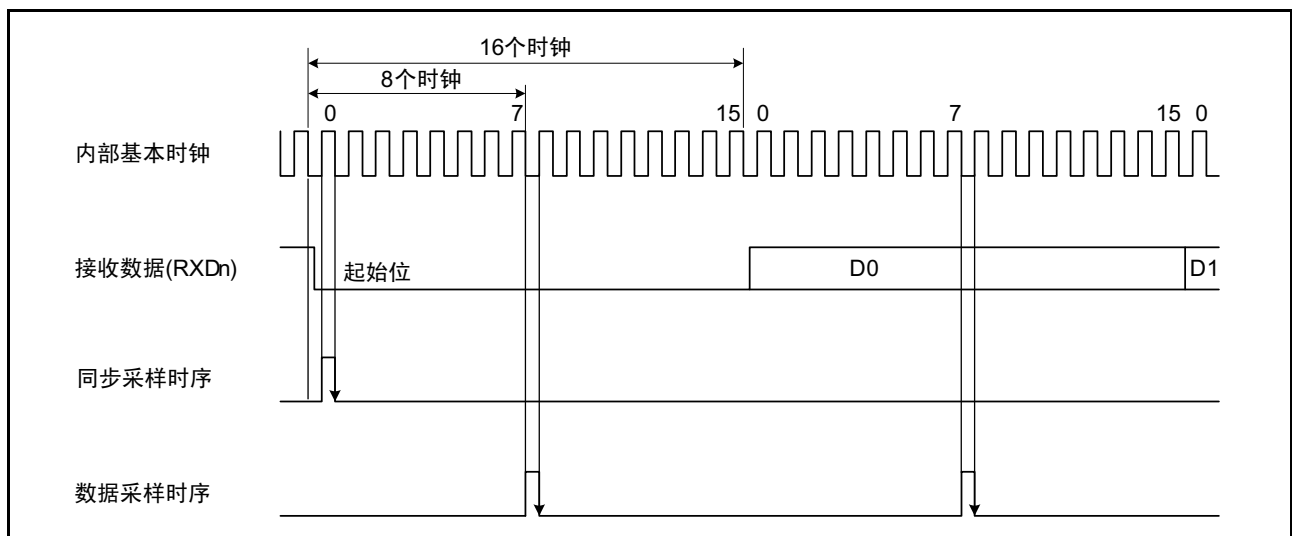


图 22.3 异步模式的接收数据采样时序

22.2.2.3 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 22.4 所示，时钟在发送数据的中央上升。

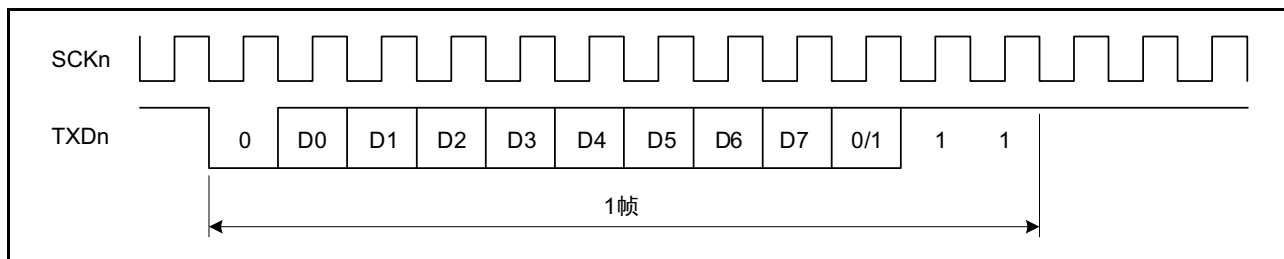


图 22.4 输出时钟和发送数据的相位关系 (异步模式)

22.2.2.4 SCI 的初始化 (异步模式)

必须在发送和接收数据前给 SCR 寄存器写初始值“00h”，并且根据图 22.5 的流程图例子对 SCI 进行初始化。必须在对 SCR 寄存器进行初始化后更改运行模式或者通信格式。

在异步模式中使用外部时钟的情况下，必须提供时钟（包括初始化期间）。

必须注意：即使将 SCR.RE 位置“0”，也不对 SSR.ORER 标志、SSR.FER 标志、SSR.PER 标志和 RDR 寄存器进行初始化。

必须注意：如果将 SCR.TE 位从“1”变为“0”或者从“0”变为“1”，就在 SCR.TIE 位为“1”时产生 TXI 中断请求。

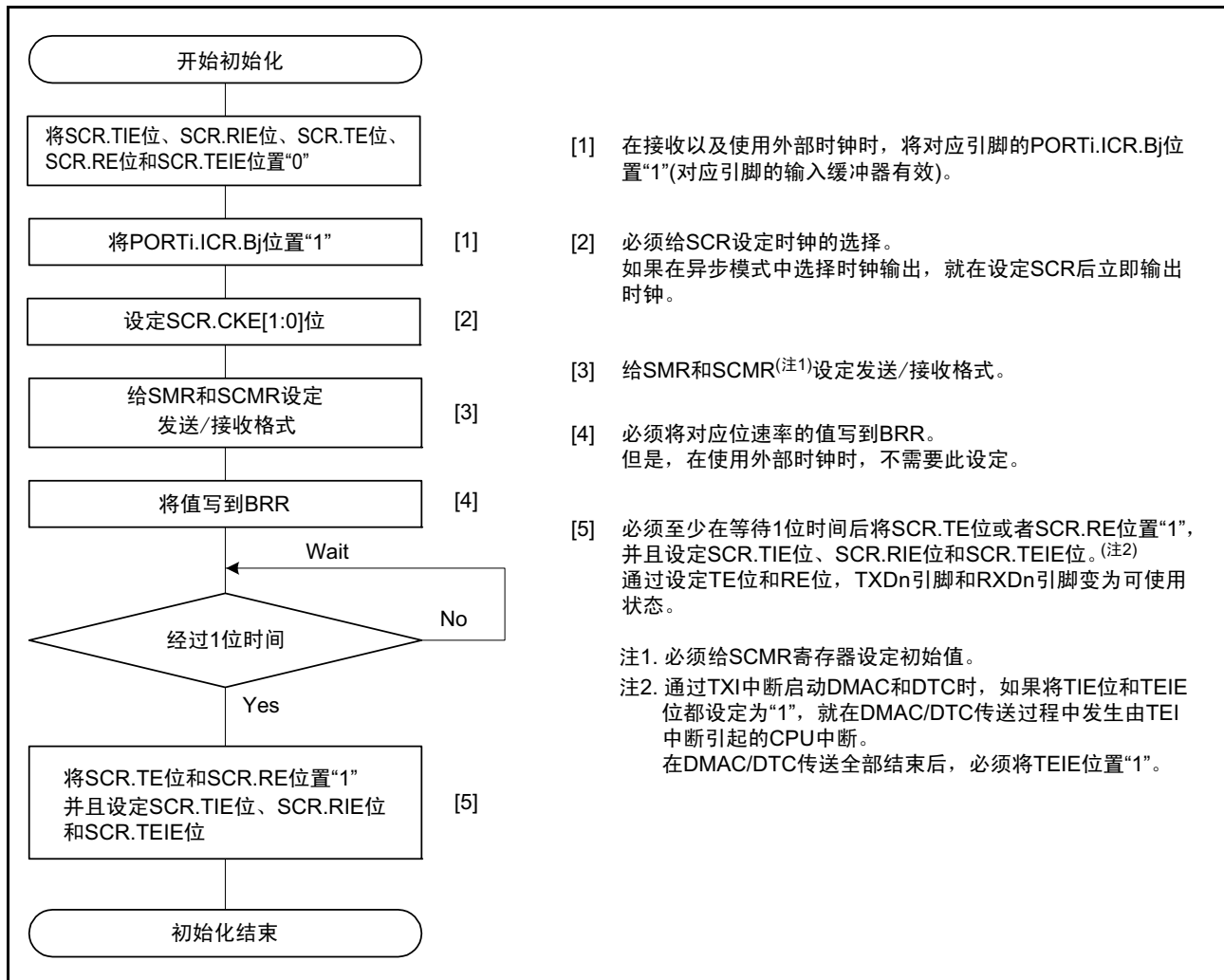


图 22.5 SCI 的初始化流程图例子 (异步模式)

22.2.2.5 串行数据的发送 (异步模式)

异步模式的串行发送例子如图 22.6 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TXI中断处理程序，在上次传送的数据发送结束前给TDR寄存器写下一个发送数据，进行连续的发送。
3. 从TXDn引脚依次发送起始位、发送数据、奇偶校验位或者多处理器位（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，就将下一个发送数据从TDR寄存器传送到TSR寄存器，并且在发送停止位后开始下一帧的发送。
6. 如果TDR寄存器未被更新，就在将SSR.TEND标志置“1”并且发送停止位后，通过输出High电平进入标记状态。此时，如果SCR.TEIE位为“1”，SSR.TEND标志就变为“1”并且产生TEI中断请求。

串行发送的流程图例子如图 22.7 所示。

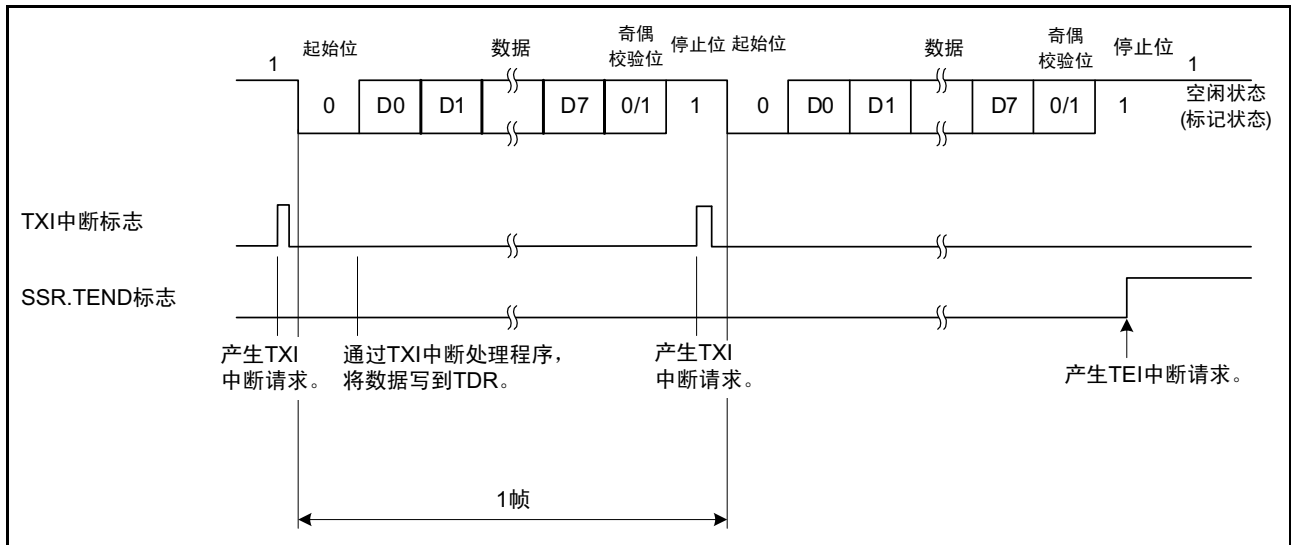


图 22.6 异步模式的串行发送例子 (8 位数据 / 有奇偶校验 / 1 个停止位的例子)



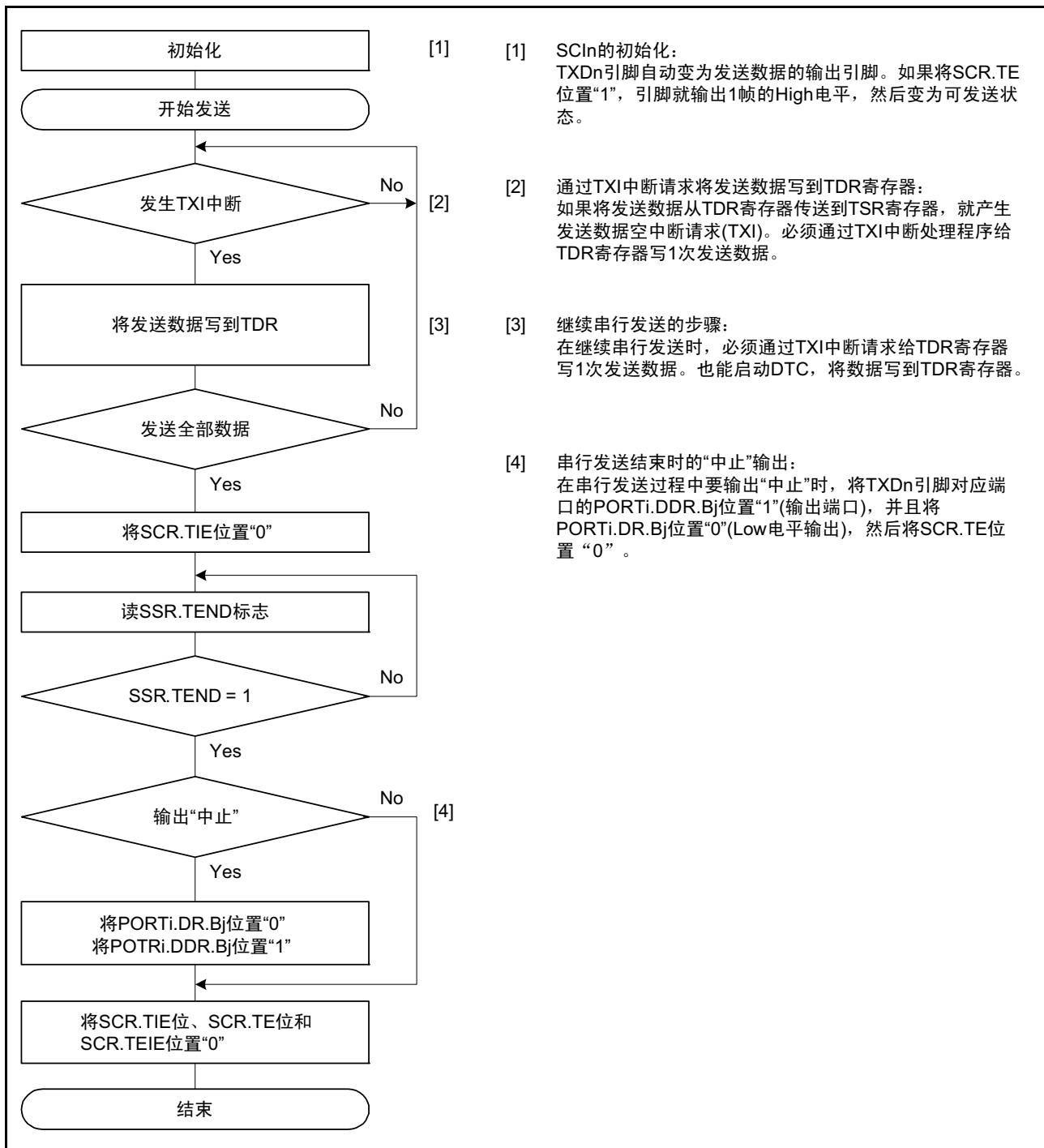


图 22.7 异步模式的串行发送的流程图例子

## 22.2.2.6 串行数据的接收 (异步模式)

异步模式的串行接收例子如图 22.8 所示。

在接收串行数据时，SCI 的运行如下：

1. 监视通信线路，如果检测到起始位，就与内部同步将接收数据取到 RSR 寄存器，并且检查奇偶校验位和停止位。
2. 如果发生溢出错误，SSR.ORER 标志就变为“1”。此时，如果 SCR.RIE 位为“1”，就产生 ERI 中断请求。不将接收数据传送到 RDR 寄存器。
3. 如果检测到奇偶校验错误，SSR.PER 标志就变为“1”，并且将接收数据传送到 RDR 寄存器。此时，如果 RIE 位为“1”，就产生 ERI 中断请求。
4. 如果检测到帧错误（停止位为“0”），SSR.FER 标志就变为“1”，并且将接收数据传送到 RDR 寄存器。此时，如果 RIE 位为“1”，就产生 ERI 中断请求。
5. 如果正常接收，就将接收数据传送到 RDR 寄存器。此时，如果 RIE 位为“1”，就产生 RXI 中断请求。能通过此 RXI 中断处理程序，在下一个数据接收结束前读已被传送到 RDR 寄存器的接收数据，进行连续接收。

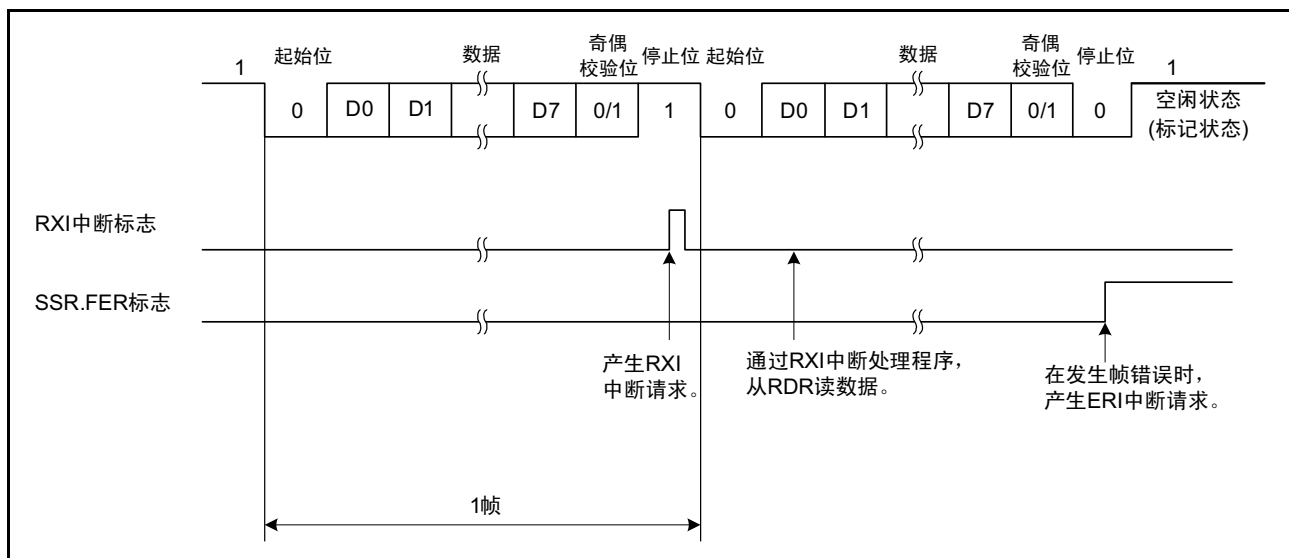


图 22.8 异步模式的串行接收例子 (8 位数据 / 有奇偶校验 / 1 个停止位的例子)

检测到接收错误时的 SSR 寄存器各状态标志的状态和接收数据的处理如表 22.13 所示。

一旦检测到接收错误，就产生 ERI 中断请求而不产生 RXI 中断请求。在接收错误标志为“1”的状态下不能进行以后的接收。因此，必须在继续接收前将 ORER 标志、FER 标志和 PER 标志置“0”。另外，在进行溢出错误处理时，必须读 RDR 寄存器。

串行接收的流程图例子如图 22.9 和图 22.10 所示。

表 22.13 SSR 寄存器状态标志的状态和接收数据的处理

SSR 寄存器的状态标志			接收数据	接收错误的状态
ORER	FER	PER		
1	0	0	消失	溢出错误
0	1	0	传送到 RDR	帧错误
0	0	1	传送到 RDR	奇偶校验错误
1	1	0	消失	溢出错误 + 帧错误
1	0	1	消失	溢出错误 + 奇偶校验错误
0	1	1	传送到 RDR	帧错误 + 奇偶校验错误
1	1	1	消失	溢出错误 + 帧错误 + 奇偶校验错误

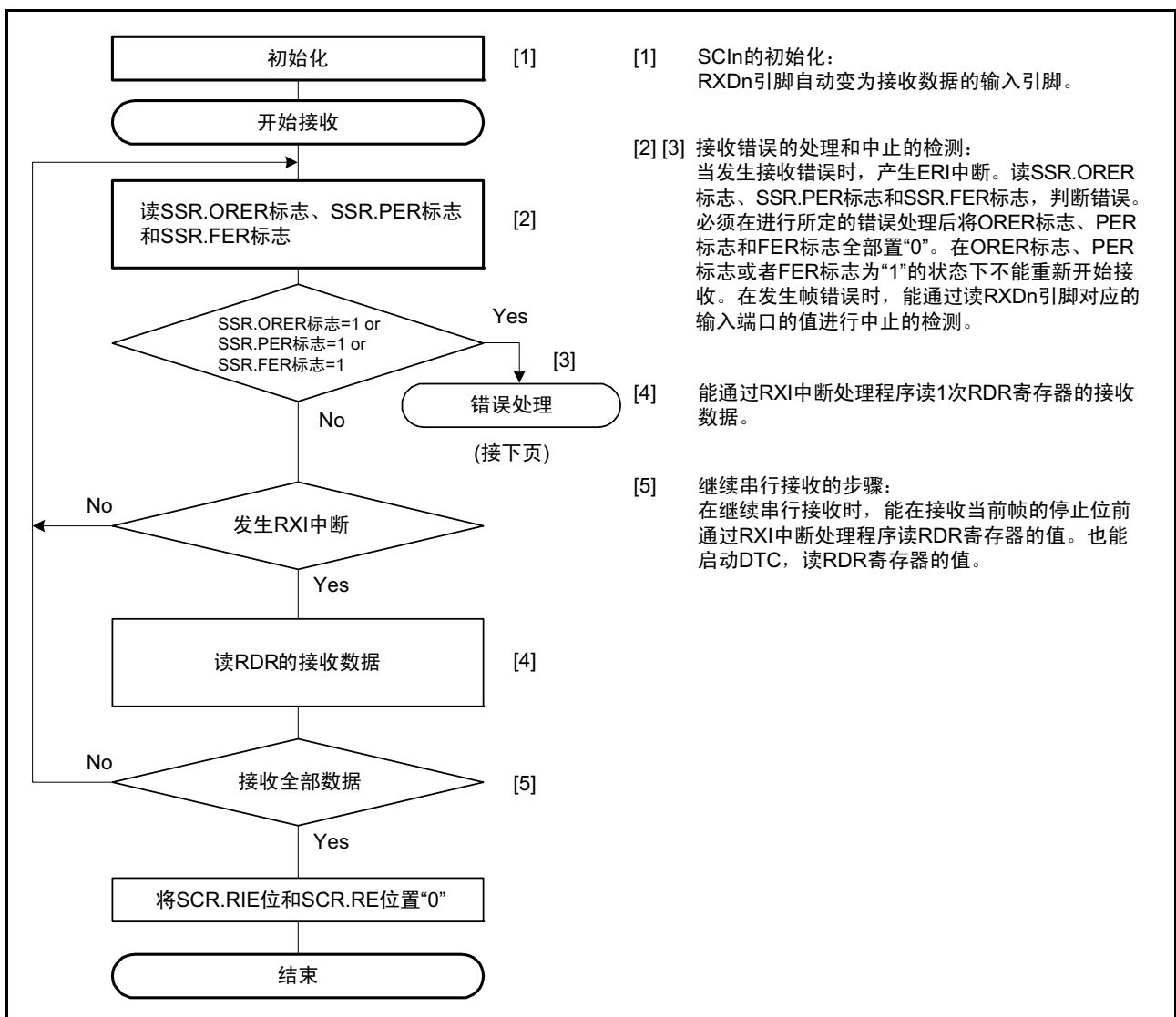


图 22.9 异步模式的串行接收的流程图例子 (1)

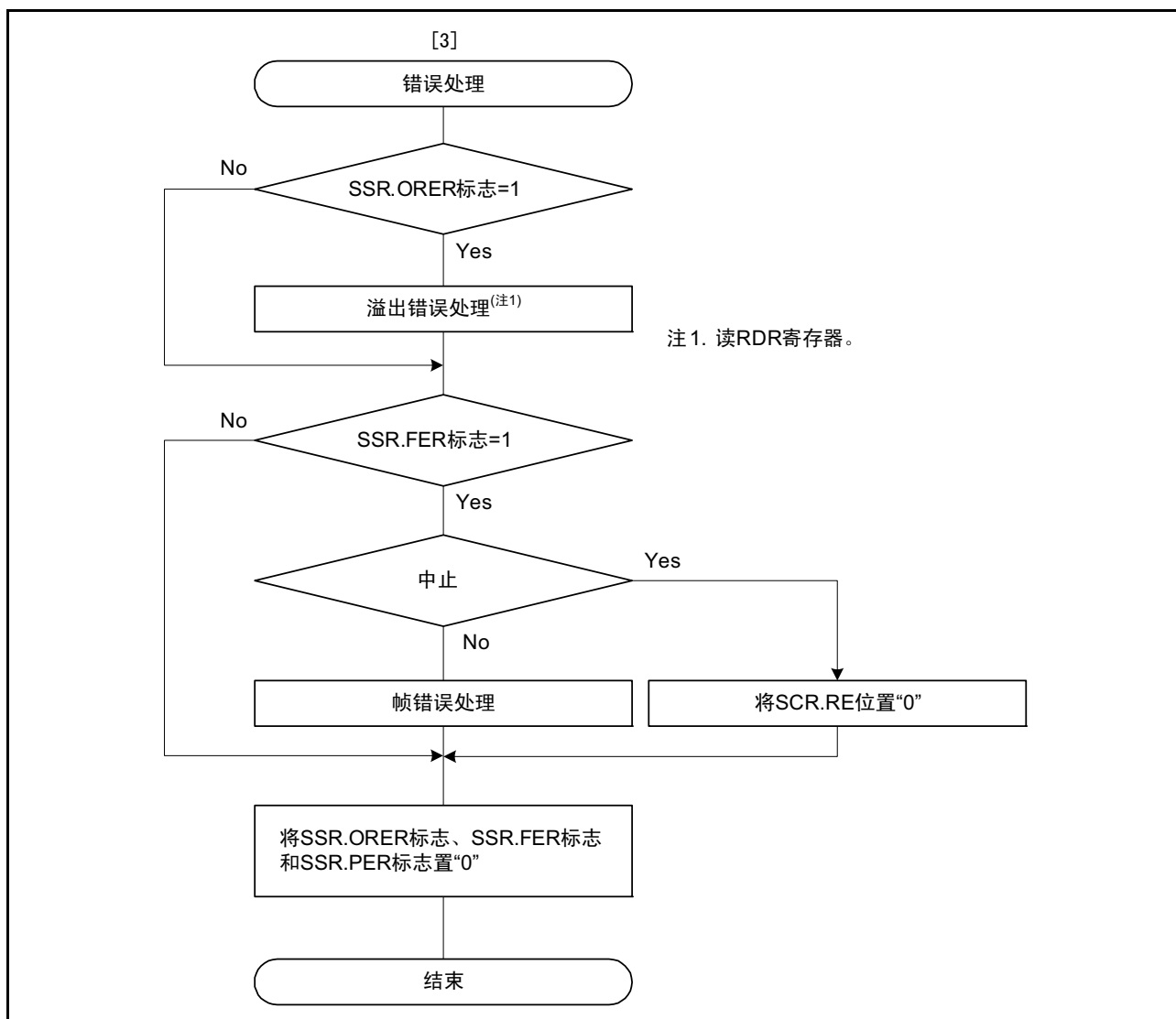


图 22.10 异步模式的串行接收的流程图例子 (2)

### 22.2.3 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 发送周期和对应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 22.11 所示。发送站首先发送给接收站 ID 码附加了多处理器位为“1”的通信数据，接着发送给接收站附加了多处理器位为“0”的通信数据。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，就在接收到下一个多处理器位为“1”的通信数据之前不进行一般的接收运行。

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，并且禁止检测接收错误和禁止将 SSR.ORER 和 SSR.FER 状态标志置位。如果接收到多处理器位为“1”的接收字符，就在 SSR.MPB 位变为“1”的同时 SCR.MPIE 自动清除，然后返回到通常的接收运行。此时，如果 SCR.RIE 位置位，就产生 RXI 中断。

在 ID 发送周期时，必须在将 SSR.MPBT 位置“1”后进行多处理器模式的发送；在数据发送周期时，必须在将 SSR.MPBT 位置“0”后进行多处理器模式的发送。

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。

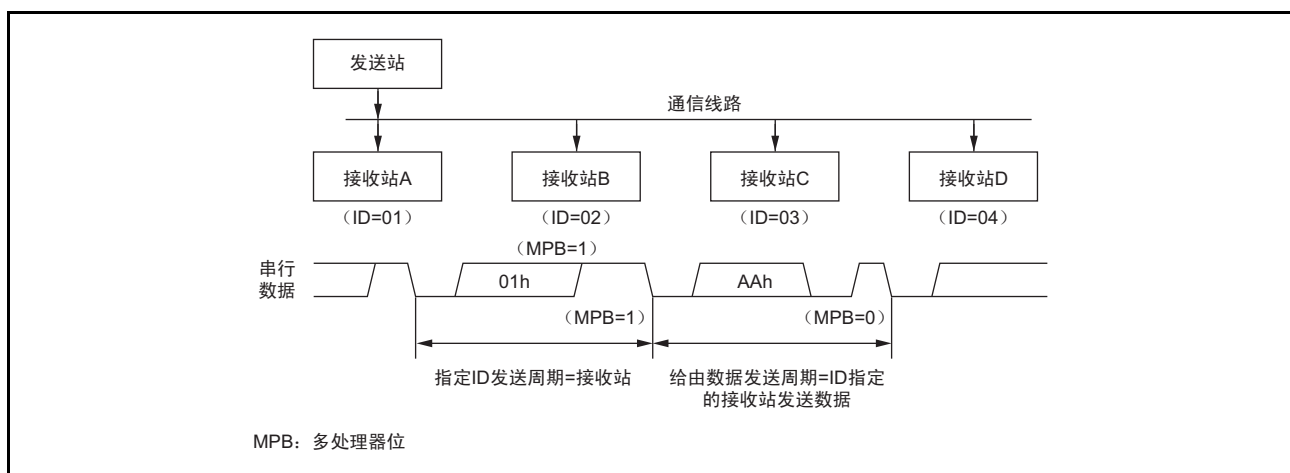


图 22.11 使用多处理器格式的通信的例子（将数据“AAh”发送到接收站 A 的例子）

22.2.3.1 多处理器串行数据的发送

多处理器串行发送的流程图例子如图 22.12 所示。在 ID 发送周期中必须将 SSR.MPBT 置“1”，然后进行发送；在数据发送周期中必须将 SSR.MPBT 置“0”，然后进行发送。其他运行和异步模式的运行相同。

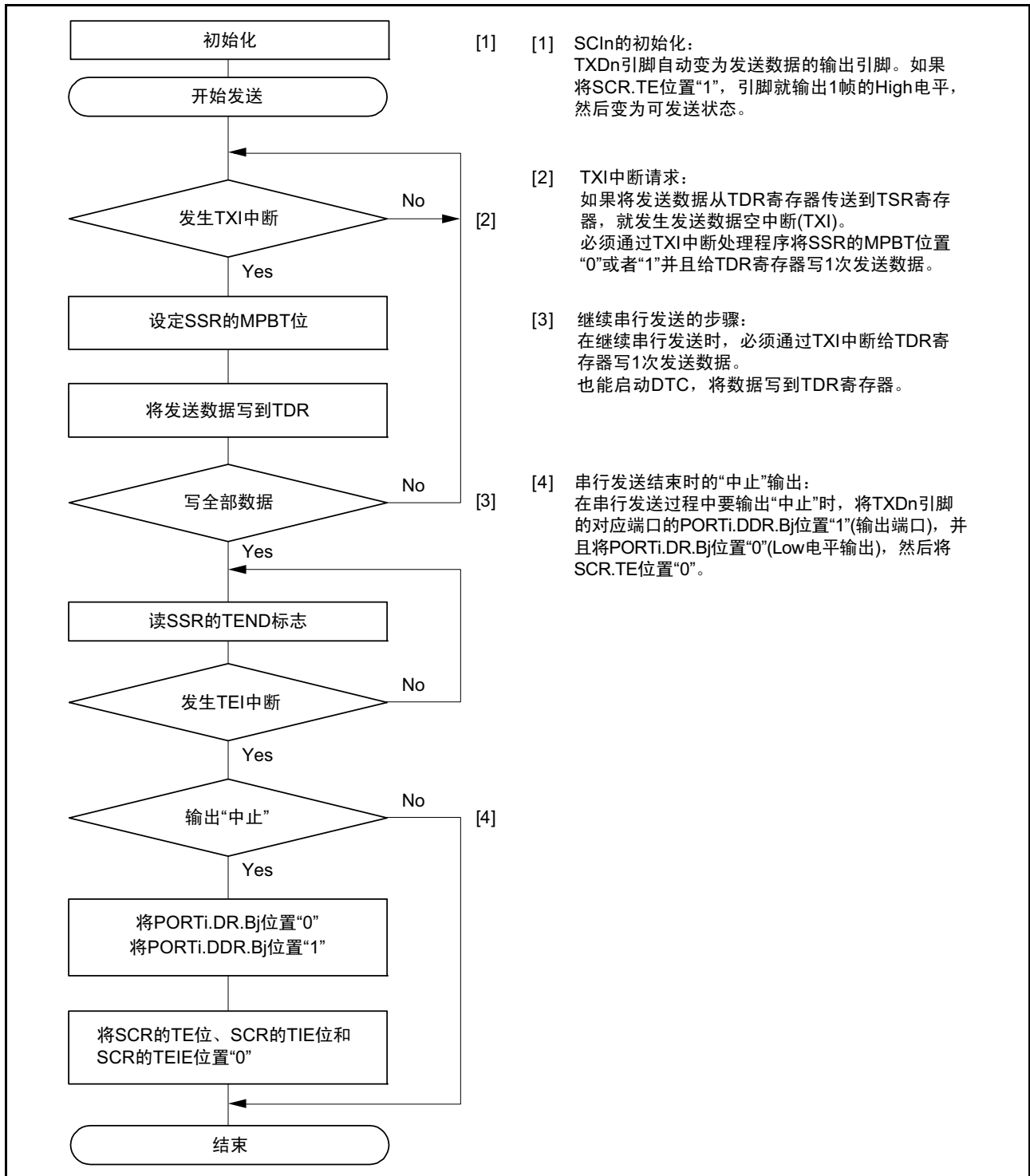


图 22.12 多处理器串行发送的流程图例子

22.2.3.2 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 22.14 和图 22.15 所示。如果将 SCR.MPIE 位置“1”，就在接收到多处理器位为“1”的通信数据之前不进行从 RSR 寄存器到 RDR 寄存器的接收数据的传送。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 22.13 所示。

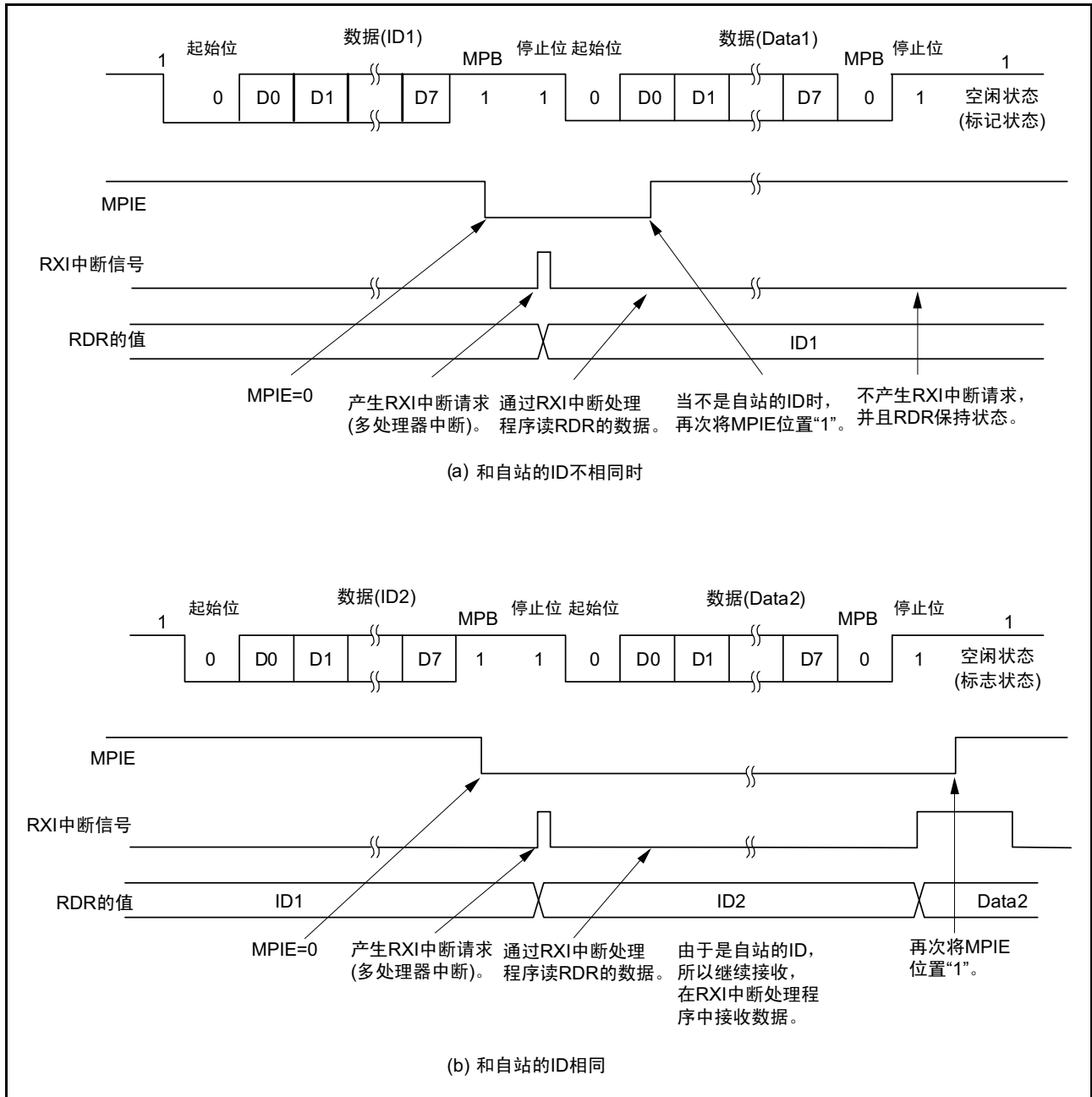


图 22.13 SCI 接收时的运行例子 (8 位数据 / 有多处理器位 / 1 个停止位的例子)

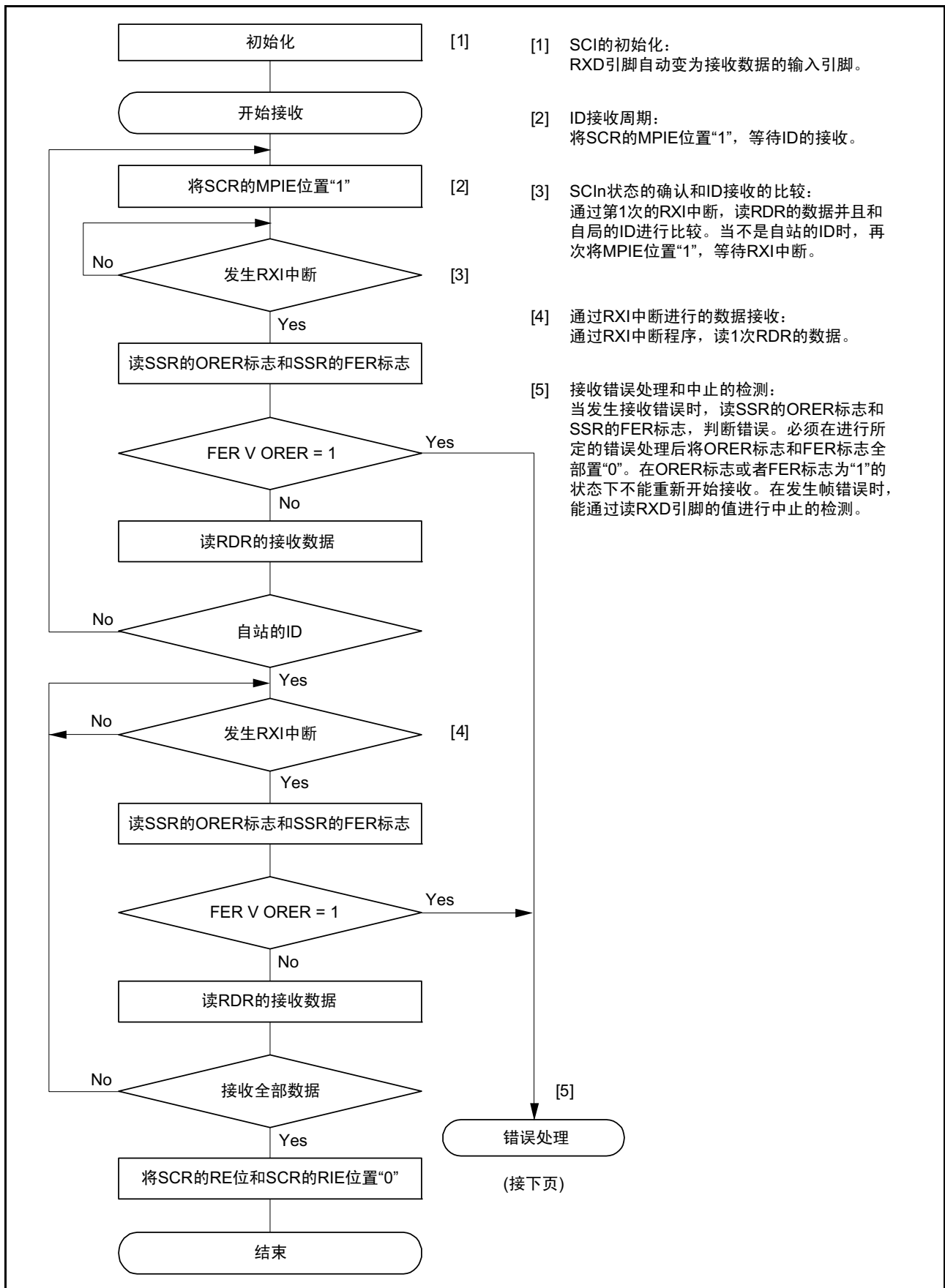


图 22.14 多处理器串行接收的流程图例子 (1)



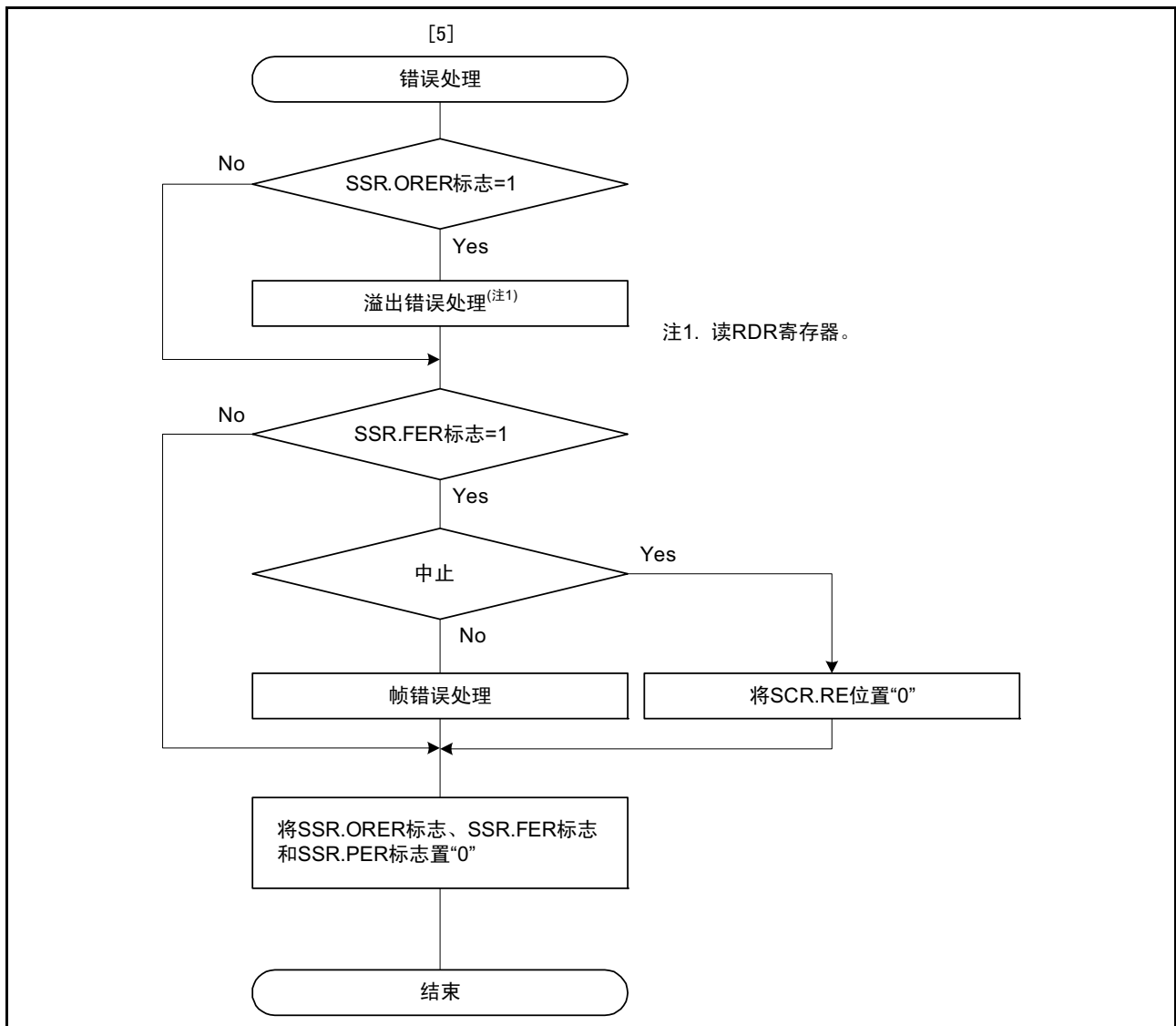


图 22.15 多处理器串行接收的流程图例子 (2)

### 22.2.4 时钟同步模式的运行

时钟同步串行通信的数据格式如图 22.16 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。

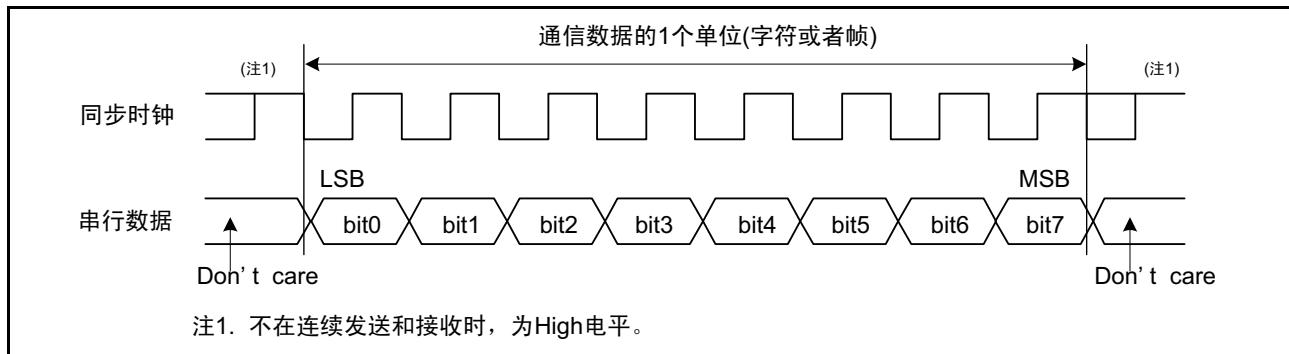


图 22.16 时钟同步串行通信的数据格式 (LSB first)

#### 22.2.4.1 时钟

能通过设定 `SCR.CKE[1:0]` 位，选择内部波特率发生器生成的内部时钟或者 `SCKn` 引脚输入的外部同步时钟。

在通过内部时钟运行时，从 `SCKn` 引脚输出同步时钟。同步时钟在进行 1 个字符的发送和接收时输出 8 个脉冲，而在不进行发送和接收时固定为 High 电平。但是，在只进行接收时，在发生溢出错误或者在将 `SCR.RE` 位置“0”前输出同步时钟。

22.2.4.2 SCI 的初始化 (时钟同步模式)

必须在发送和接收数据前给 SCR 寄存器写初始值“00h”，并且根据图 22.17 的流程图例子进行初始化。必须在对 SCR 寄存器进行初始化后更改运行模式或者通信格式。

必须注意：即使将 SCR.RE 位置“0”，也不对 SSR.ORER 标志、SSR.FER 标志、SSR.PER 标志或者 RDR 寄存器进行初始化。

必须注意：如果将 SCR.TE 位从“1”变为“0”或者从“0”变为“1”，就在 SCR.TIE 位为“1”时发生 TXI 中断。

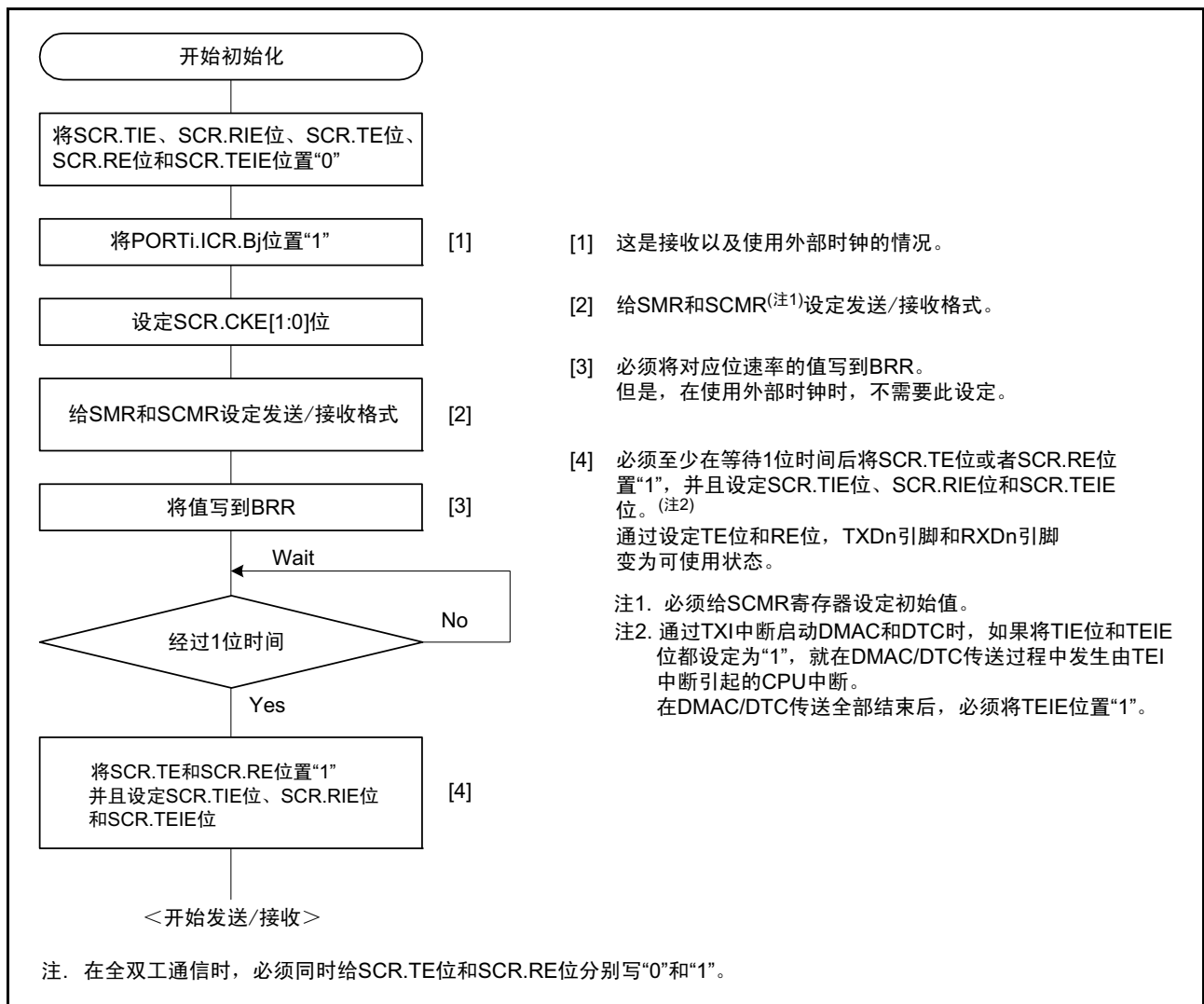


图 22.17 SCI 的初始化流程图例子 (时钟同步模式)

## 22.2.4.3 串行数据的发送 (时钟同步模式)

时钟同步模式的串行发送例子如图 22.18 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 TIE 位为“1”，就产生 TXI 中断请求。能通过此 TXI 中断处理程序，在上次传送的数据发送结束前给 TDR 寄存器写下一个发送数据，进行连续的发送。
3. 如果设定为时钟输出模式，就与输出时钟同步从 TXDn 引脚输出 8 位数据。如果设定为外部时钟，就与输入时钟同步从 TXDn 引脚输出 8 位数据。
4. 在发送最后 1 位数据时检查 TDR 寄存器的更新（写）。
5. 如果 TDR 寄存器已被更新，就将数据从 TDR 寄存器传送到 TSR 寄存器，开始下一帧的发送。
6. 如果 TDR 寄存器未被更新，SSR.TEND 标志就变为“1”，并且保持最后 1 位的输出状态。此时，如果 SCR.TEIE 位为“1”，就产生 TEI 中断请求。SCKn 引脚被固定为 High 电平。

串行发送的流程图例子如图 22.19 所示。

不能在接收错误标志 (SSR.ORER、SSR.FER、SSR.PER) 为“1”的状态下开始发送，必须在开始发送前将接收错误标志置“0”。必须注意：即使只将 SCR.RE 位置“0”，接收错误标志也不变为“0”。

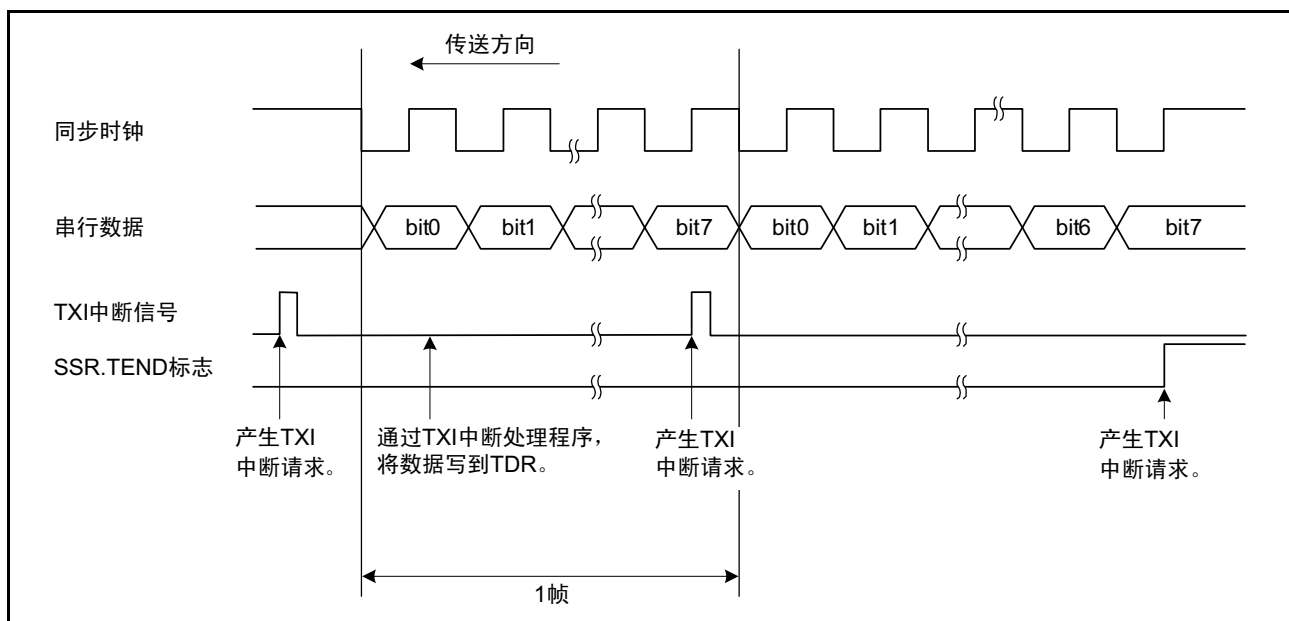


图 22.18 时钟同步模式的串行发送例子

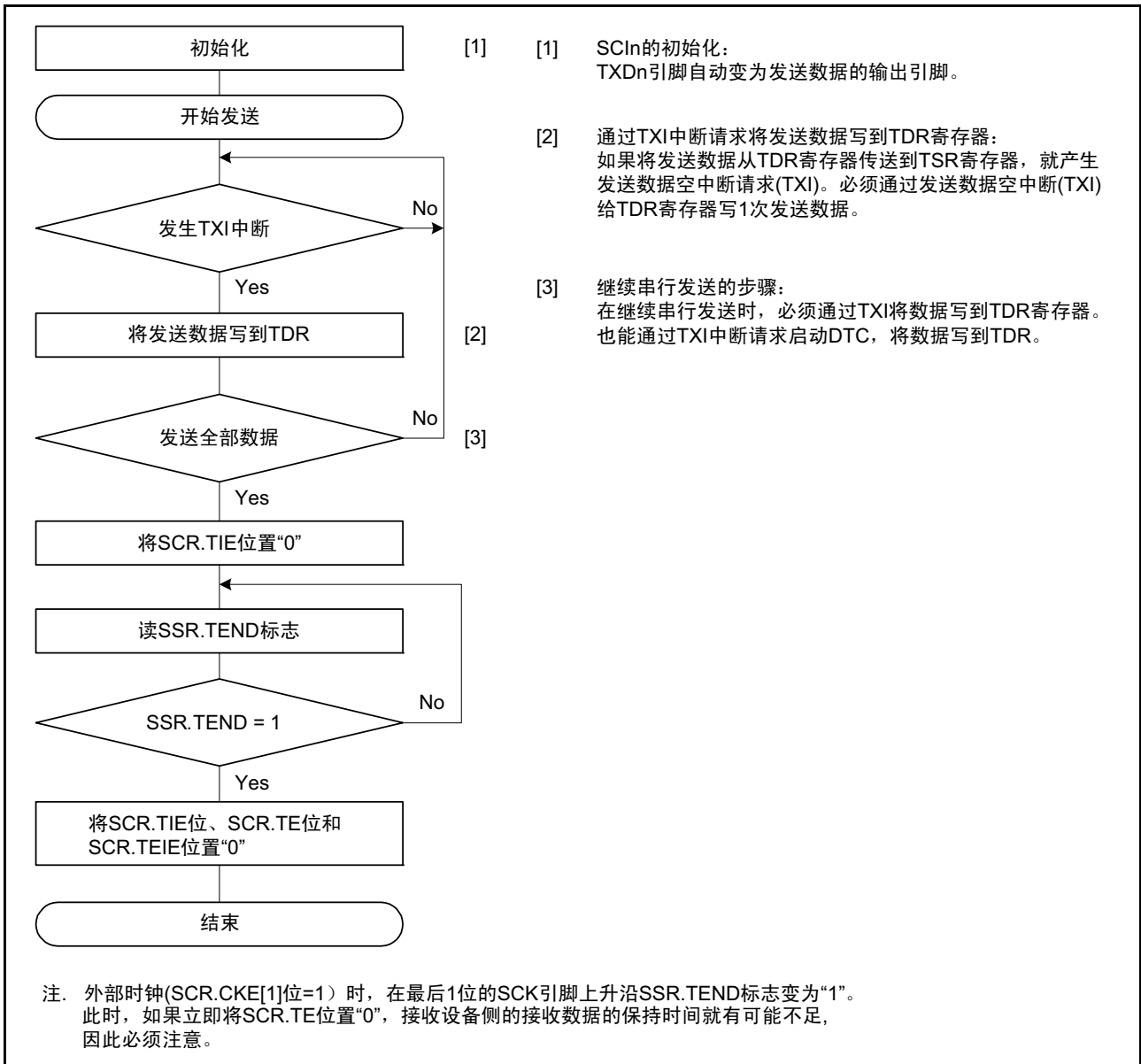


图 22.19 时钟同步模式的串行发送的流程图例子

## 22.2.4.4 串行数据的接收 (时钟同步模式)

时钟同步模式的串行接收例子如图 22.20 所示。

在接收串行数据时，SCI 的运行如下：

1. SCI与同步时钟的输入同步或者与输出同步将内部进行初始化后开始接收，并且将接收数据取到RSR寄存器。
2. 如果发生溢出错误，SSR.ORER标志就变为“1”。此时，如果SCR.RIE位为“1”，就产生ERI中断请求。不将接收数据传送到RDR寄存器。
3. 如果正常接收，就将接收数据传送到RDR寄存器。此时，如果RIE位为“1”，就产生RXI中断请求。能通过此RXI中断处理程序，在下一个数据接收结束前读已传送到RDR寄存器的接收数据，进行连续接收。

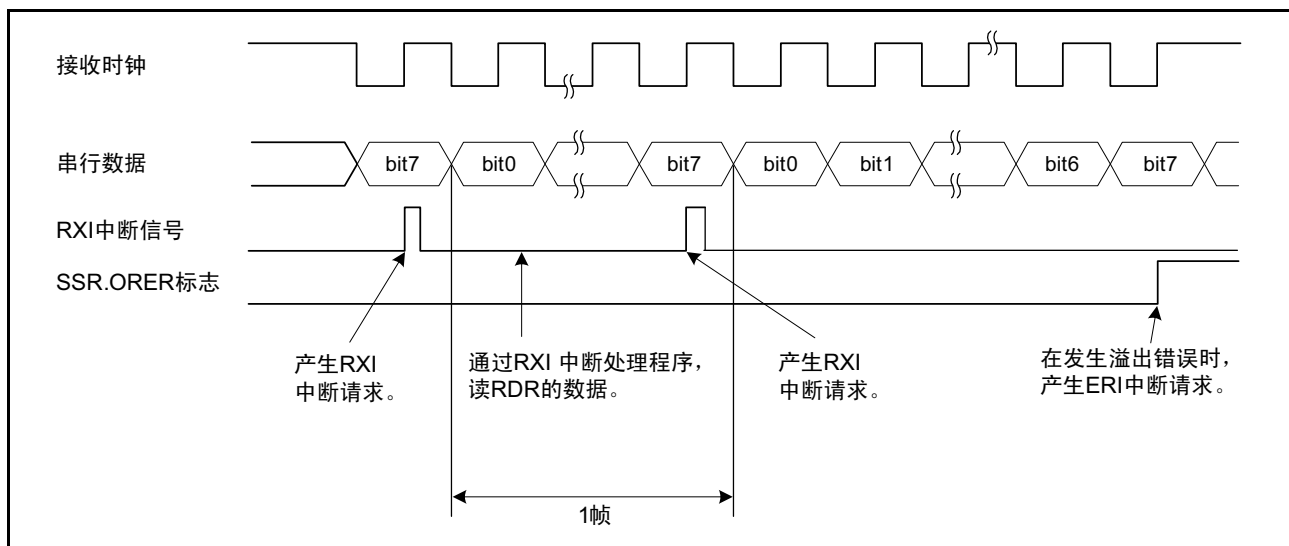


图 22.20 时钟同步模式的串行接收例子

在接收错误标志为“1”的状态下，不能进行以后的发送和接收。因此，必须在继续接收前将 SSR.ORER 标志、SSR.FER 标志和 SSR.PER 标志置“0”。另外，必须在进行溢出错误处理时读 RDR 寄存器。

串行接收的流程图例子如图 22.21 所示。

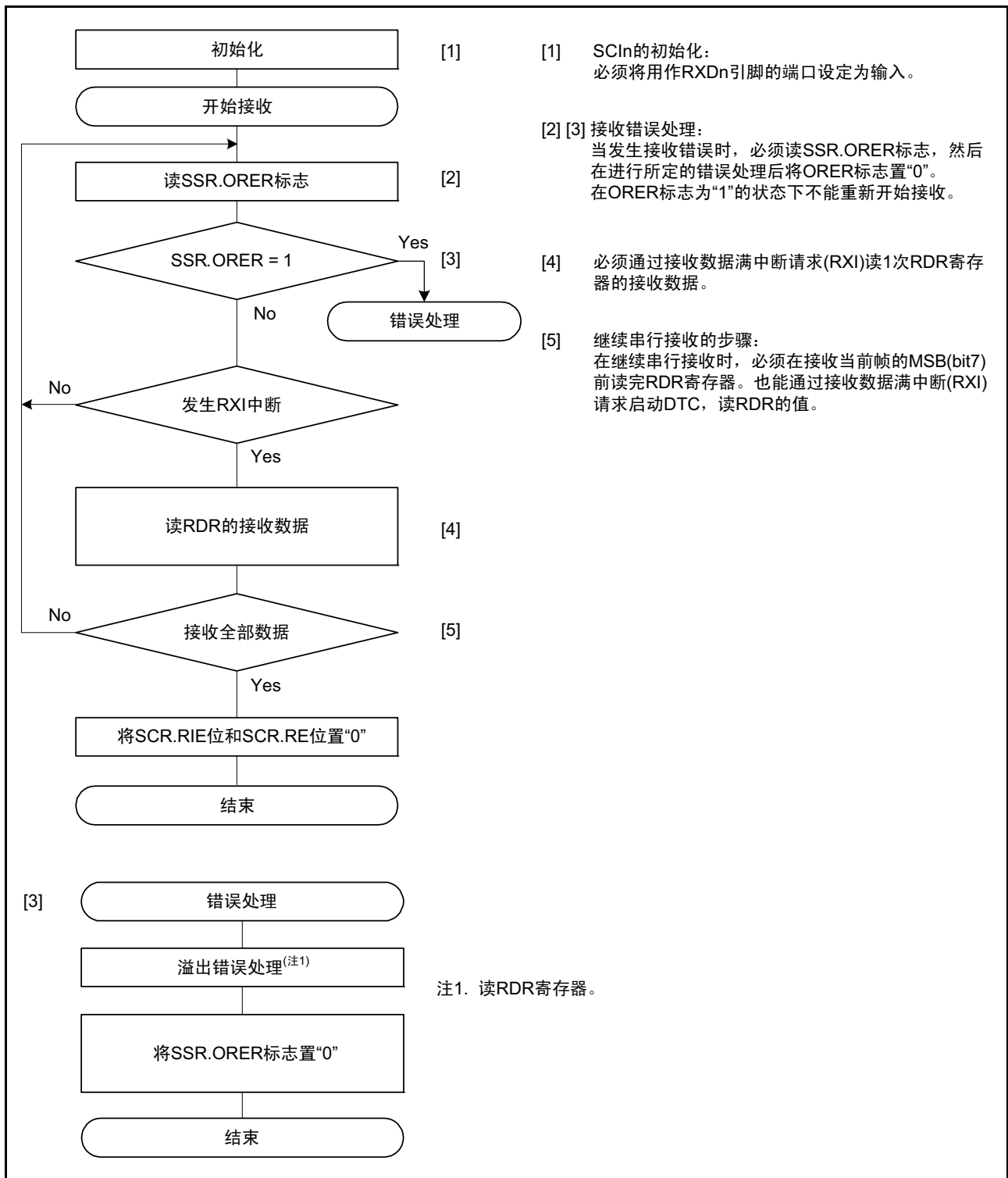


图 22.21 时钟同步模式的串行接收的流程图例子

22.2.4.5 串行数据的全双工通信 (时钟同步模式)

时钟同步模式的串行全双工通信的流程图例子如图 22.22 所示。

在对 SCI 进行初始化后, 必须按照以下步骤进行串行全双工通信。

要从发送转换为全双工通信时, 必须通过读已被置“1”的 SSR.TEND 标志, 确认 SCI 处于发送结束状态。然后, 必须在对 SCR 寄存器进行初始化后, 用 1 条指令将 SCR.TIE 位、SCR.RIE 位、SCR.TE 位、SCR.RE 位和 SCR.TEIE 位同时置“1”。

要从接收转换为全双工通信时, 必须在确认 SCI 处于接收结束状态后, 先将 SCR.RIE 位和 SCR.RE 位置“0”, 然后确认错误标志 (SSR.ORER、SSR.FER 和 SSR.PER) 为“0”, 最后用 1 条指令将 SCR.TIE 位、SCR.RIE 位、SCR.TE 位、SCR.RE 位和 SCR.TEIE 位同时置“1”。

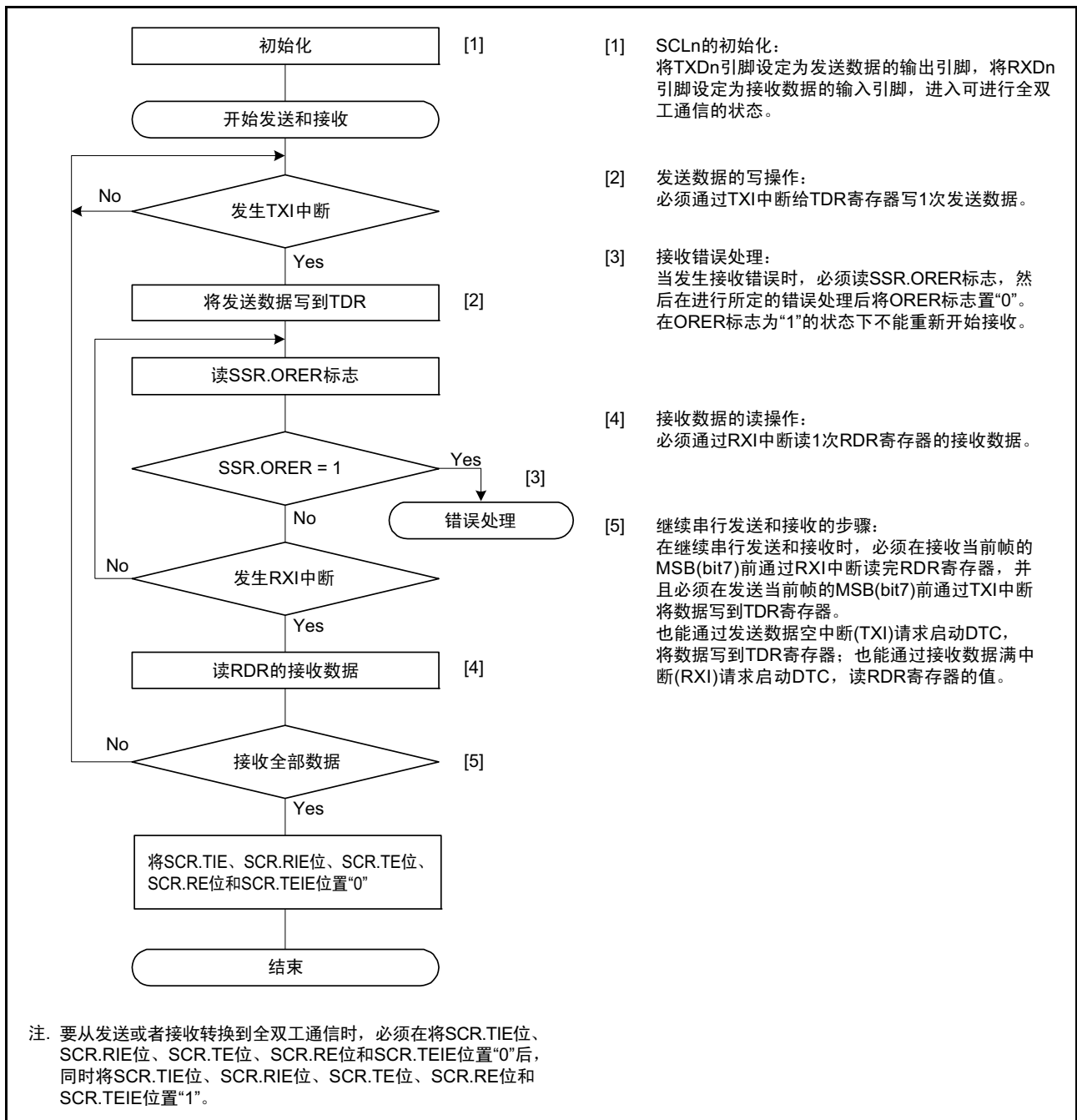


图 22.22 时钟同步模式的串行全双工通信的流程图例子



## 22.3 智能卡接口模式

作为 SCI 的扩展功能，对支持符合 ISO/IEC 7816-3 (Identification Card) 规格的智能卡 (IC 卡) 接口 (SMCI)。

在 SCMR.SMIF 位为“1”时，SCI 为智能卡接口模式。

### 22.3.1 寄存器说明

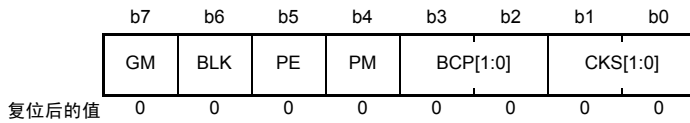
SMCI 的寄存器一览表如表 22.14 所示，部分寄存器 (TDR、RDR、SCMR 寄存器) 的功能和串行通信接口模式相同。有关 TDR、RDR、SCMR 寄存器的详细内容，请参照“22.2.1 寄存器说明”。

表 22.14 SMCI 的寄存器一览表

通道	寄存器名	符号	复位后的值	地址	存取长度
SMCI0	串行模式寄存器	SMR	00h	0008 8240h	8
	位速率寄存器	BRR	FFh	0008 8241h	8
	串行控制寄存器	SCR	00h	0008 8242h	8
	发送数据寄存器	TDR	FFh	0008 8243h	8
	串行状态寄存器	SSR	84h	0008 8244h	8
	接收数据寄存器	RDR	00h	0008 8245h	8
	智能卡模式寄存器	SCMR	F2h	0008 8246h	8
	串行扩展模式寄存器	SEMR	00h	0008 8247h	8
SMCI1	串行模式寄存器	SMR	00h	0008 8248h	8
	位速率寄存器	BRR	FFh	0008 8249h	8
	串行控制寄存器	SCR	00h	0008 824Ah	8
	发送数据寄存器	TDR	FFh	0008 824Bh	8
	串行状态寄存器	SSR	84h	0008 824Ch	8
	接收数据寄存器	RDR	00h	0008 824Dh	8
	智能卡模式寄存器	SCMR	F2h	0008 824Eh	8
	串行扩展模式寄存器	SEMR	00h	0008 824Fh	8
SMCI2	串行模式寄存器	SMR	00h	0008 8250h	8
	位速率寄存器	BRR	FFh	0008 8251h	8
	串行控制寄存器	SCR	00h	0008 8252h	8
	发送数据寄存器	TDR	FFh	0008 8253h	8
	串行状态寄存器	SSR	84h	0008 8254h	8
	接收数据寄存器	RDR	00h	0008 8255h	8
	智能卡模式寄存器	SCMR	F2h	0008 8256h	8
	串行扩展模式寄存器	SEMR	00h	0008 8257h	8

## 22.3.1.1 串行模式寄存器 (SMR)

地址 SMC10.SMR 0008 8240h、SMC11.SMR 0008 8248h、SMC12.SMR 0008 8250h



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	组合选择 SCMR.BCP2 位 SCMR.BCP2 位、SMR.BCP[1:0] 的设定值 BCP2 b3 b2 0 0 0: 93 (S=93) (注2) 0 0 1: 128 (S=128) (注2) 0 1 0: 186 (S=186) (注2) 0 1 1: 512 (S=512) (注2) 1 0 0: 32 (S=32) (注2) (初始值) 1 0 1: 64 (S=64) (注2) 1 1 0: 372 (S=372) (注2) 1 1 1: 256 (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: 在偶校验时接收 / 发送 1: 在奇校验时接收 / 发送	R/W (注3)
b5	PE	奇偶校验允许位	(只在异步模式中有效) 当 PE 位为“1”时, 在发送时附加奇偶校验位, 在接收时进行奇偶校验。在智能卡接口模式中, 必须将 PE 位置“1”。	R/W (注3)
b6	BLK	块传送模式位	0: 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“22.2.1.9 位速率寄存器 (BRR)”中的 n 的值。

注 2. S 表示“22.3.1.4 位速率寄存器 (BRR)”中的 S 的值。

注 3. 只能在 SCR.TE 位和 SCR.RE 位都为“0” (禁止串行发送和串行接收) 时写这些位。

SMR 寄存器选择通信格式以及内部波特率发生器的时钟源。

## CKS[1:0] 位 (时钟选择位)

这些位选择内部波特率发生器的时钟源。

有关 CKS[1:0] 位的设定值和波特率的关系, 请参照“22.3.1.4 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“22.3.3.2 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“22.3.3 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“22.3.3.1 块传送模式”。

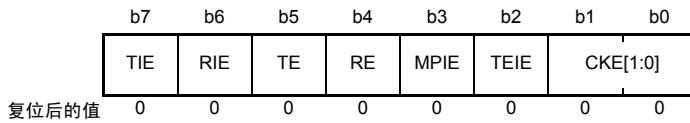
**GM 位 (GSM 模式位)**

如果将 GM 位置“1”, 就以 GSM 模式运行。

在 GSM 模式中, SSR.TEND 标志变为“1”的时序为开始发送后的 11.0etu (etu: Elementary Time Unit, 1 位传送时间), 并且追加时钟输出控制功能, 详细内容请参照“22.3.3.4 串行数据的发送 (块传送模式除外)”和“22.3.3.6 时钟的输出控制”。

## 22.3.1.2 串行控制寄存器 (SCR)

地址 SMC10.SCR 0008 8242h、SMC11.SCR 0008 824Ah、SMC12.SCR 0008 8252h



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"> <li>• SMR.GM 位 =0 的情况</li> <li style="margin-left: 20px;">b1 b0</li> <li style="margin-left: 20px;">0 0: 禁止输出 (SCKn 引脚能用作输入 / 输出端口。)</li> <li style="margin-left: 20px;">0 1: 输出时钟</li> <li style="margin-left: 20px;">1 0: 不能设定</li> <li style="margin-left: 20px;">1 1: 不能设定</li> </ul> <ul style="list-style-type: none"> <li>• SMR.GM 位 =1 的情况</li> <li style="margin-left: 20px;">b1 b0</li> <li style="margin-left: 20px;">0 0: 固定为 Low 电平输出</li> <li style="margin-left: 20px;">0 1: 输出时钟</li> <li style="margin-left: 20px;">1 0: 固定为 High 电平输出</li> <li style="margin-left: 20px;">1 1: 输出时钟</li> </ul>	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

注 1. 只能在 TE 位和 RE 位都为“0”时写这些位。

注 2. 当 SMR.CM 位为“1”时。只能在 TE 位和 RE 位都为“0”时写“1”。一旦将 TE 位或者 RE 位置“1”，就只能在 TE 位和 RE 位都为“0”时写此位。

SCR 寄存器是控制发送 / 接收和中断以及选择发送 / 接收时钟源的寄存器。  
有关各中断源，请参照“22.5 中断源”。

## CKE[1:0] 位 (时钟允许位)

这些位控制 SCKn 引脚的时钟输出。

能在 GSM 模式中对时钟输出进行动态转换，详细内容请参照“22.3.3.6 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 中断和 ERI 中断。

要禁止 RXI 中断时, 将 RIE 位置“0”。

如果在从 SSR.ORER 标志、SSR.FER 标志和 SSR.PER 标志读“1”后将这些标志位置“0”, 或者将 RIE 位置“0”, ERI 中断请求信号就消失。

**TIE 位 (发送中断允许位)**

此位允许或者禁止通知 TXI 中断。

要禁止 TXI 中断时, 将 TIE 位置“0”。

## 22.3.1.3 串行状态寄存器 (SSR)

地址 SMC10.SSR 0008 8244h、SMC11.SSR 0008 824Ch、SMC12.SSR 0008 8254h

b7	b6	b5	b4	b3	b2	b1	b0
TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
1	0	0	0	0	1	0	0

复位后的值

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)
b6	RDRF	接收数据满标志	0: 当从 RDR 寄存器传送数据时 1: 当接收正常结束并且将数据从 RSR 寄存器传送到 RDR 寄存器时	R/(W) (注 2)
b7	TDRE	发送数据空标志	0: 当将数据传送到 TDR 寄存器时 1: 当将数据从 TDR 寄存器传送到 TSR 寄存器时	R/(W) (注 2)

注 1. 为了清除标志，只能在读“1”后写“0”。

注 2. 只能写“1”。

SSR 寄存器由 SCI 的状态标志构成。

## 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，此标志变为“1”。

[ 为“1”的条件 ]

- SCR.TE 位为“0” (禁止串行发送) 时
- 在发送 1 字节数据后的一定时间后，ERS 标志变为“0”并且未更新 TDR 寄存器时  
根据寄存器的设定，变为“1”的时序如下：

当 SMR.GM 位和 SMR.BLK 位都为“0”时，在开始发送后的 12.5etu。

当 SMR.GM 位为“0”并且 SMR.BLK 位为“1”时，在开始发送后的 11.5etu。

当 SMR.GM 位为“1”并且 SMR.BLK 位为“0”时，在开始发送后的 11.0etu。

当 SMR.GM 位和 SMR.BLK 位都为“1”时，在开始发送后的 11.0etu。

[ 为“0”的条件 ]

- 给 TDR 寄存器写发送数据时

**PER 标志 (奇偶校验错误标志)**

此标志表示在异步模式中接收的数据发生了奇偶校验错误。

[为“1”的条件]

- 在接收时检测到奇偶校验错误时

将发生奇偶校验错误时的接收数据传送到RDR寄存器，但是不产生RXI中断请求。在PER标志为“1”的状态下，不能继续进行以后的串行接收。以后的接收数据不传送到RDR寄存器。

[为“0”的条件]

- 读“1”后写“0”（必须在写“0”后确认PER标志已变为“0”）时
- 即使将SCR.RE位置“0”（禁止串行接收），PER标志也不受影响而保持原来的状态。

**ERS 标志 (错误信号状态标志)**

[为“1”的条件]

- 对错误信号的Low电平进行采样时

[为“0”的条件]

- 读“1”后写“0”时

**ORER 标志 (溢出错误标志)**

此标志表示接收的数据发生了溢出错误。

[为“1”的条件]

- 不读RDR寄存器的接收数据就接收到下一个数据时

RDR寄存器保持发生溢出错误前的接收数据而放弃后面的接收数据。在ORER标志为“1”的状态下，不能继续进行以后的串行接收。

[为“0”的条件]

- 读“1”后写“0”（必须在写“0”后确认ORER标志已变为“0”）时
- 即使将SCR.RE位置“0”，ORER标志也不受影响而保持原来的状态。

**RDRF 标志 (接收数据满标志)**

此标志表示在RDR寄存器中是否有接收数据。

[为“1”的条件]

- 接收正常结束并且将数据从RSR寄存器传送到RDR寄存器时

[为“0”的条件]

- 从RDR寄存器传送数据时

**TDRE 标志 (发送数据空标志)**

此标志表示在TDR寄存器中是否有发送数据。

[为“1”的条件]

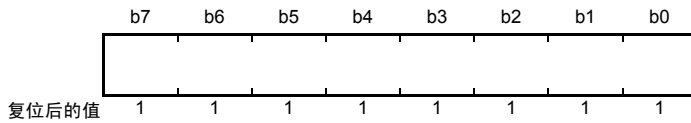
- 将数据从TDR寄存器传送到TSR寄存器时

[为“0”的条件]

- 将数据传送到TDR寄存器时

22.3.1.4 位速率寄存器 (BRR)

地址 SMCIO.BRR 0008 8241h、SMCI1.BRR 0008 8249h、SMCI2.BRR 0008 8251h



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。

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。

在智能卡接口模式中，BRR 寄存器的设定值 N 和位速率 B 的关系如表 22.15 所示。在智能卡接口模式中 能选择 1 位传送期间的基本时钟数 S。详细内容请参照“22.3.3.2 接收数据的采样时序和接收容限”。

表 22.15 BRR 寄存器的设定值 N 和位速率 B 的关系

模式	BRR寄存器的设定值	误差
智能卡接口	$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$\text{误差}(\%) = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$

B : 位速率 (bps)

N : 波特率发生器的 BRR 设定值 (0 ≤ N ≤ 255)

PCLK : 工作频率 (MHz)

n、S : 取决于下表中 SMR 的设定值。

表 22.16 对应位速率的 BRR 设定例子 (在智能卡接口模式中, n 为“0”并且 S 为“372”的情况)

位速率 (bps)	工作频率 PCLK (MHz)											
	7.1424			10.00			10.7136			13.00		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
9600	0	0	0.00	0	1	30	0	1	25	0	1	8.99

位速率 (bps)	工作频率 PCLK (MHz)											
	14.2848			16.00			18.00			20.00		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
9600	0	1	0.00	0	1	12.01	0	2	15.99	0	2	6.66

位速率 (bps)	工作频率 PCLK (MHz)											
	25.00			30.00			33.00			50.00		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
9600	0	3	12.49	0	3	5.01	0	4	7.59	0	27	0.00

表 22.17 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	PCLK (MHz)	最大位速率 (bps)	n	N
10.00	13441	0	0	20.00	26882	0	0
10.7136	14400	0	0	25.00	33602	0	0
13.00	17473	0	0	30.00	40323	0	0
16.00	21505	0	0	33.00	44355	0	0
18.00	24194	0	0	50.00	67205	0	0



### 22.3.2 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。

#### 22.3.2.1 连接例子

智能卡 (IC 卡) 的连接例子如图 22.23 所示。

IC 卡是指通过 1 条数据传送线进行发送和接收，因此必须连接 TXDn 引脚和 RXDn 引脚并且通过电阻将数据传送线上拉到电源 VCC 侧。

如果在不连接 IC 卡的状态下将 SCR.TE 位和 SCR.RE 位都置“1”，就能进行自诊断 (自己能接收自己发送的数据)。

如果给 IC 卡提供由 SCI 生成的时钟，就必须将 SCKn 引脚的输出信号输入到 IC 卡的 CLK 引脚。

能将 LSI 的输出端口用于复位信号的输出。

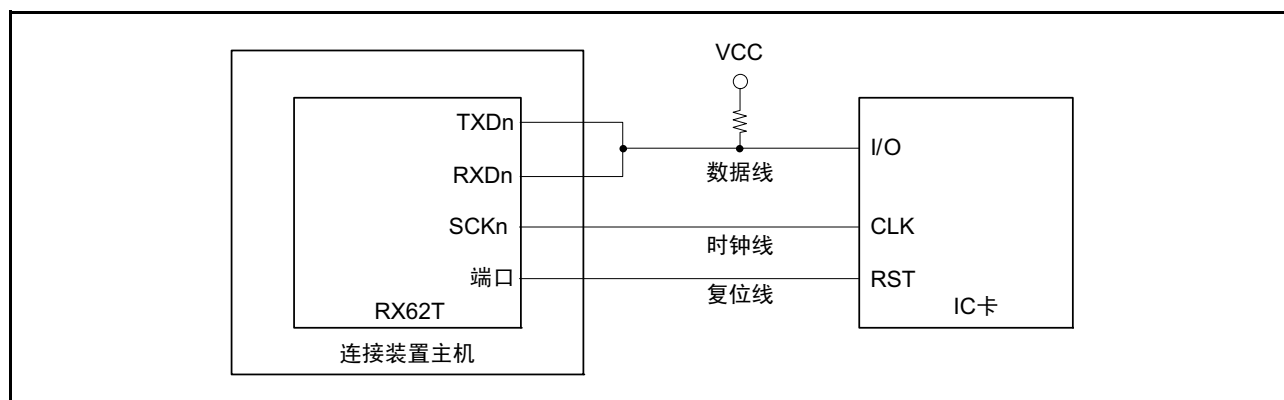


图 22.23 智能卡 (IC 卡) 的连接例子

### 22.3.3 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 22.24 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 22.24 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 22.25 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，并且以 LSB first 进行发送和接收。图 22.25 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”。

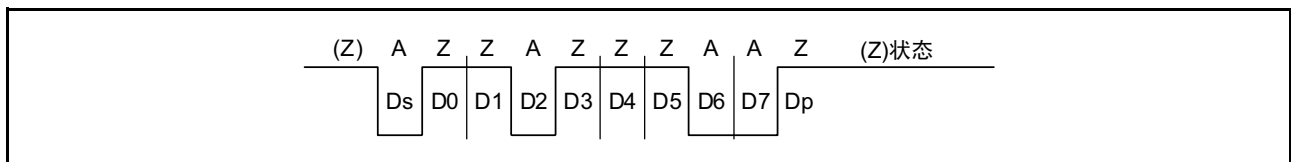


图 22.25 正向协议 (SCMR.SDIR 位 = 0, SCMR.SINV 位 = 0, SMR.PM 位 = 0)

## (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 22.26 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 RX62T 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。

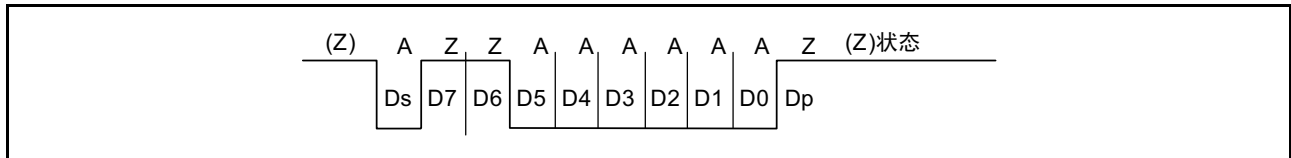


图 22.26 反向协议 (SCMR.SDIR 位 =1, SCMR.SINV 位 =1, SMR.PM 位 =1)

### 22.3.3.1 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 标志变为“1”，所以必须在接收下一帧的奇偶校验位前将此位置“0”。
- 在发送时，从奇偶校验位结束到下一帧的开始至少需要 1etu 的保护时间。
- 因为不重新进行发送，所以在开始发送后的 11.5etu 后，SSR.TEND 标志变为“1”。
- 和一般的智能卡接口模式一样，SSR.ERS 标志表示错误信号的状态。因为不发送和接收错误信号，所以 SSR.ERS 标志总是为“0”。

### 22.3.3.2 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 22.27 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 个时钟的上升沿分别对接收数据进行采样，在各位的中央取数据。能用以下表达式表示此时的接收容限：

$$M = \left| \left( 0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100[\%]$$

M : 接收容限 (%)

N : 对于时钟的位速率比 (N=32、64、372、256)

D : 时钟占空比 (D=0 ~ 1.0)

L : 帧长 (L=10)

F : 时钟频率的偏差绝对值

假设在上述的表达式中，F=0，D=0.5，N=372，则接收容限如下所示：

$$M = (0.5 - 1/2 \times 372) \times 100\% = 49.866\%$$

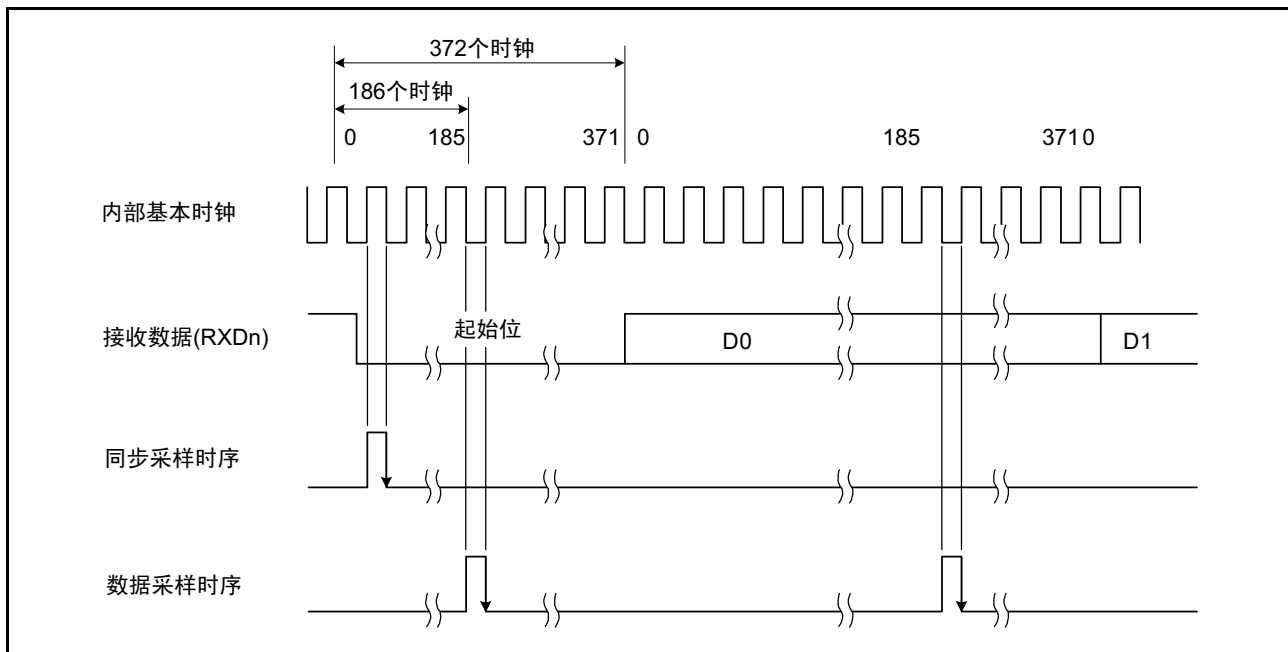


图 22.27 智能卡接口模式的接收数据的采样时序（使用 372 倍的时钟）

### 22.3.3.3 智能卡接口的初始化

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. 必须将对应引脚的 PORTi.ICR.Bj 位（n=1~9、A~G，j=0~7）置“1”。
3. 必须将 SSR 寄存器的错误标志（ORER、ERS、PER）置“0”。
4. 必须设定 SMR.GM 位、SMR.BLK 位、SMR.PM 位、SMR.BCP[1:0] 位、SMR.CKS[1:0] 位和 SCMR.BCP2 位。此时，必须将 SMR.PE 位置“1”。
5. 必须设定 SCMR.SDIR 位、SCMR.SINV 位和 SCMR.SMIF 位，并且将 TXDn 引脚对应的 PORTi.DDR.Bj 位置“0”。因此，TXDn 引脚和 RXDn 引脚都从端口转换为 SCI 的引脚，并且为高阻抗状态。
6. 给 BRR 寄存器设定与位速率对应的值。
7. 必须设定 SCR.CKE[1:0] 位。此时，必须将 SCR.TIE 位、SCR.RIE 位、SCR.TE 位、SCR.RE 位和 SCR.TEIE 位置“0”。

如果将 CKE0 位置“1”，就从 SCKn 引脚输出时钟。

8. 必须在至少等待 1 位时间后设定 SCR.TIE 位、SCR.RIE 位、SCR.TE 位和 SCR.RE 位。除了进行自诊断以外，必须将 TE 位和 RE 位同时置“1”。

要从接收模式转换为发送模式时，必须在确认接收结束后从初始化开始进行，并且将 TE 位置“1”、RE 位置“0”。能通过 RXI 中断请求、SSR.ORER 标志或者 SSR.PER 标志确认接收的结束。

要从发送模式转换为接收模式时，必须在确认发送结束后从初始化开始进行，并且将 TE 位置“0”、RE 位置“1”。能通过 SSR.TEND 标志确认发送的结束。

22.3.3.4 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 22.28 所示。

1. 如果在结束 1 帧的发送后采样到接收侧的错误信号, SSR.ERS 标志就变为“1”。此时, 如果 SCR.RIE 位为“1”, 就产生 ERI 中断请求。必须在对下一个奇偶校验位进行采样前将 ERS 标志置“0”。
2. 在接收到错误信号的帧中, SSR.TEND 标志不变为“1”。再次将数据从 TDR 寄存器传送到 TSR 寄存器, 自动进行重新发送。
3. 如果接收侧没有返回错误信号, ERS 标志就不变为“1”。
4. 在判断出包括重新传送的 1 帧的发送已结束后, SSR.TEND 标志被置位。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 22.30 所示。能通过 TXI 中断源启动 DTC, 自动进行上述的一系列处理。

在发送时, 如果在 SCR.TIE 位为“1”时 SSR.TEND 标志变为“1”, 就产生 TXI 中断请求。如果预先将 TXI 中断请求设定为 DTC 的启动源, 就在通过 TXI 中断请求启动 DTC 后, 进行发送数据的传送。在通过 DTC 进行数据传送时, TEND 标志自动变为“0”。

如果发生错误, SCI 就自动重新发送相同的数据。在此期间, TEND 标志保持“0”并且不启动 DTC。因此, 自动发送 SCI 以及 DTC 指定的字节数, 包括发生错误时的重新发送。但是, 在发生错误时 ERS 标志不自动变为“0”, 因此必须预先将 RIE 位置“1”, 使在发生错误时产生 ERI 中断请求, 并且将 ERS 标志置“0”。

在使用 DTC 进行发送和接收时, 必须先设定 DTC, 在设定为允许状态后进行 SCI 的设定。

有关 DTC 的设定方法, 请参照“14. 数据传送控制器 (DTC) ”。

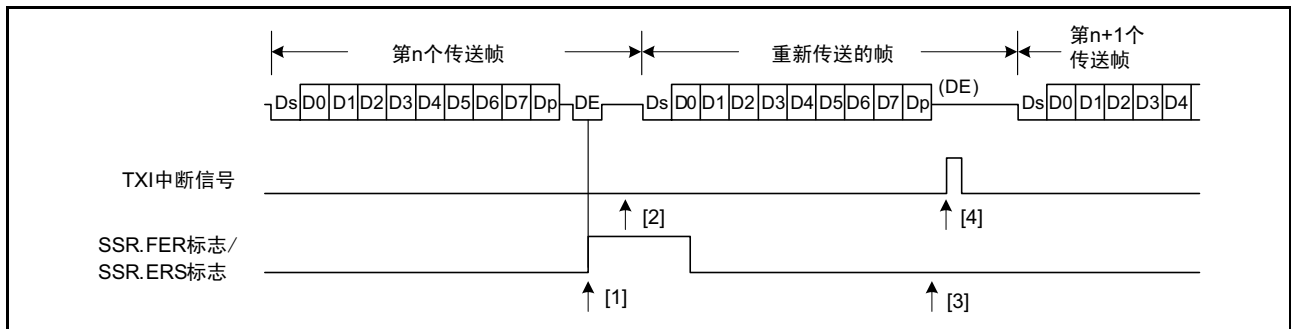


图 22.28 SMC1 发送模式的重新传送 (发送时的重新传送)

SSR.TEND 标志变为“1”的时序取决于 SMR.GM 位的设定, TEND 标志的产生时序如图 22.29 所示。

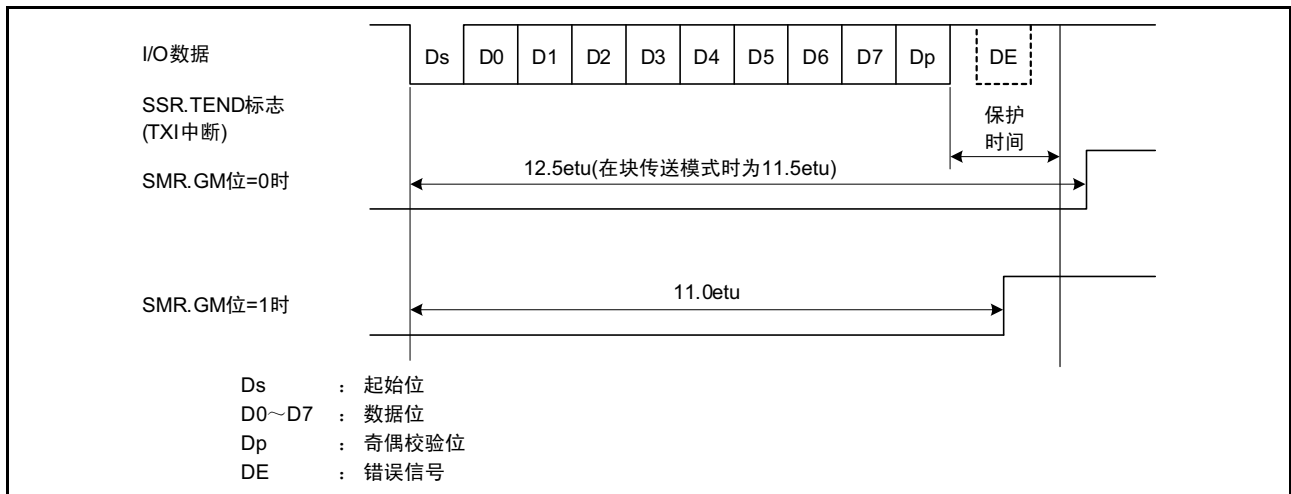


图 22.29 发送时的 SSR.TEND 标志的产生时序

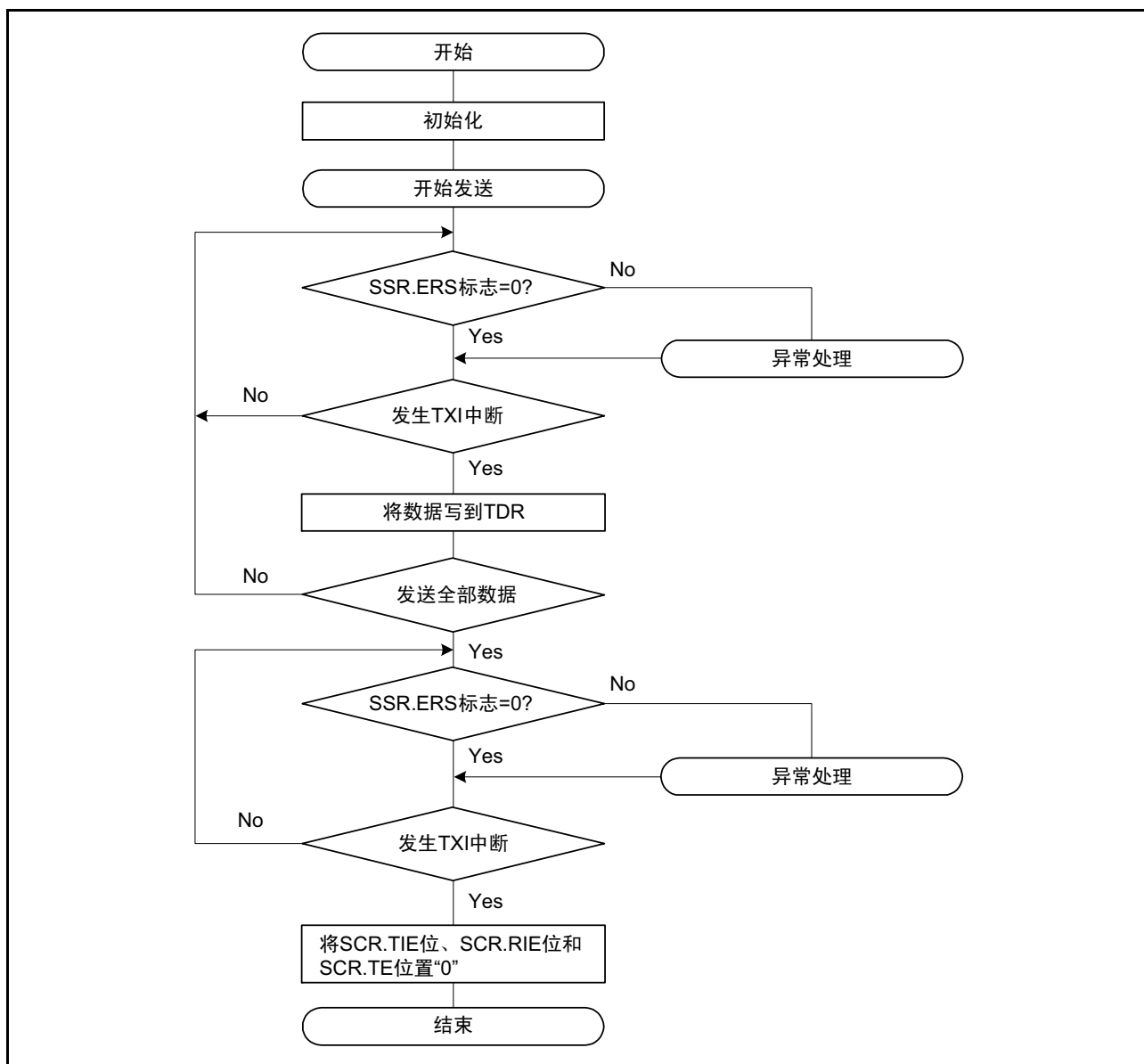


图 22.30 串行发送的流程图例子

22.3.3.5 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 22.31 所示。

1. 如果在接收数据时检测到奇偶校验错误, SSR.PER 标志就变为“1”。此时, 如果 SCR.RIE 位为“1”, 就产生 ERI 中断请求。必须在下一个奇偶校验位的采样时序前将 PER 标志置“0”。
2. 对于检测到奇偶校验错误的帧, 不发生 RXI 中断。
3. 如果检测不到奇偶校验错误, SSR.PER 标志就不变为“1”。
4. 如果在判断出接收正常结束后 RIE 位变为“1”, 就产生 RXI 中断请求。

串行接收的流程图例子如图 22.32 所示。能通过 RXI 中断请求启动 DTC, 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 中断请求设定为 DTC 的启动源, 就在通过 RXI 中断请求启动 DTC 后, 进行接收数据的传送。

如果在接收时发生错误, 并且 SSR.ORER 标志或者 SSR.PER 标志变为“1”, 就产生接收错误中断 (ERI) 请求, 因此必须将错误标志置“0”。因为在发生错误时不启动 DTC 而省略接收数据, 所以只传送由 DTC 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 标志为“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“22.2.2 异步模式的运行”。

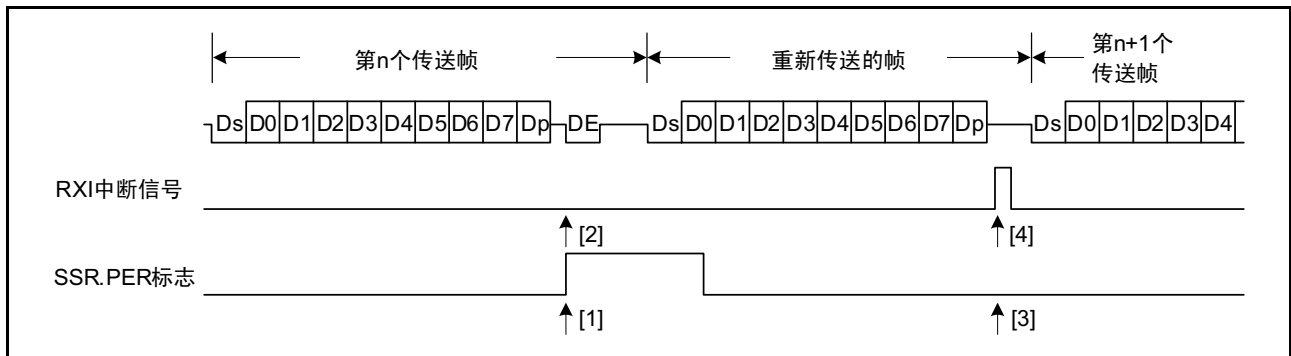


图 22.31 SMCI 接收模式的重新传送 (接收时的重新传送)

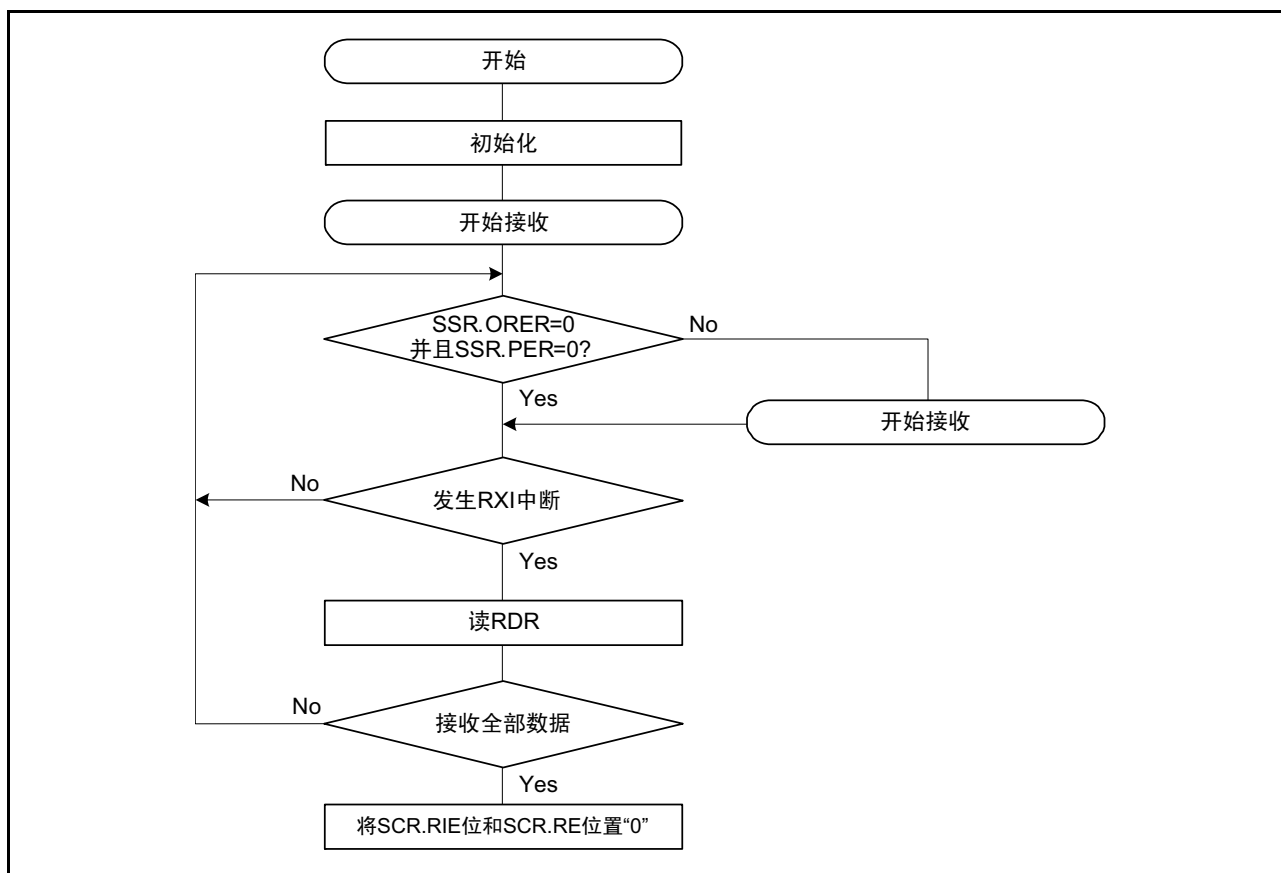


图 22.32 串行接收的流程图例子

22.3.3.6 时钟的输出控制

当 SMR.GM 位为“1”时，能通过 SCR.CKE[1:0] 位停止时钟输出。此时，能将时钟脉冲的最小宽度设定为指定的宽度。

时钟输出的停止时序如图 22.33 所示，这是将 GM 位置“1”、将 CKE1 位置“0”并且控制 CKE0 位时的例子。

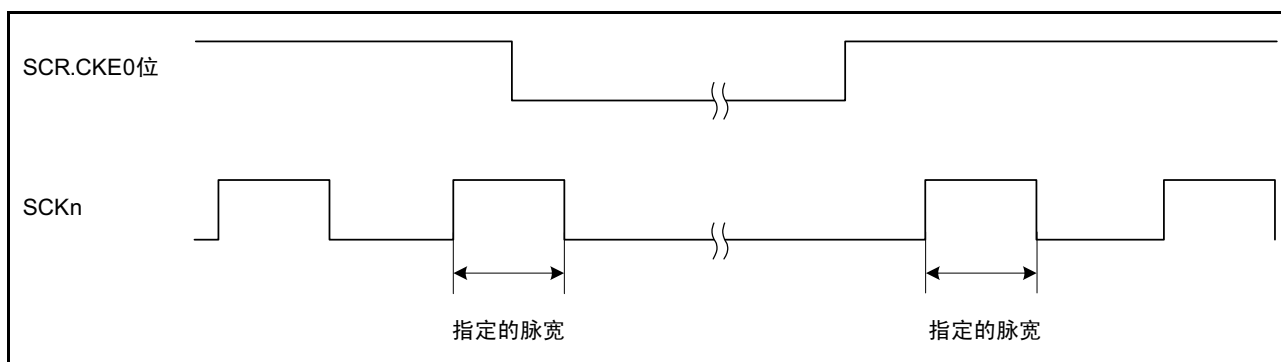


图 22.33 时钟输出的停止时序



在接通电源、向软件待机模式转移或者从软件待机模式返回时，为了确保时钟的占空比，必须按照以下步骤进行处理：

### (1) 电源接通时

为了从接通电源开始确保时钟的占空比的转换步骤如下所示：

1. 初始状态为端口输入的高阻抗状态。要固定电位时，必须使用上拉电阻或者下拉电阻。
2. 必须通过SCR.CKE1位将SCKn引脚固定为指定的输出。
3. 必须将SMR寄存器和SCMR寄存器置位并且转换为智能卡接口模式的运行，并且将SCR.CKE0位置“1”，开始时钟的输出。

### (2) 模式转换时

#### (a) 从智能卡接口模式转移到软件待机模式时

1. 必须给SCKn引脚对应的数据寄存器 (PORTi.DR) 和数据方向寄存器 (PORTi.DDR) 设定软件待机模式时的输出固定状态的值。
2. 必须给SCR.TE位和SCR.RE位写“0”，停止发送和接收。  
同时，必须给SCR.CKE1位设定软件待机时的输出固定状态的值。
3. 必须给SCR.CKE0位写“0”，停止时钟。
4. 必须等待1个串行时钟周期。在此期间，在输出指定的High电平宽度后，以Low电平停止时钟输出。
5. 必须转移到软件待机状态。

#### (b) 从软件待机模式返回到智能卡接口模式时

1. 必须解除软件待机状态。
2. 必须将SCR.CKE0位置“1”，以指定的频率重新开始时钟的输出。

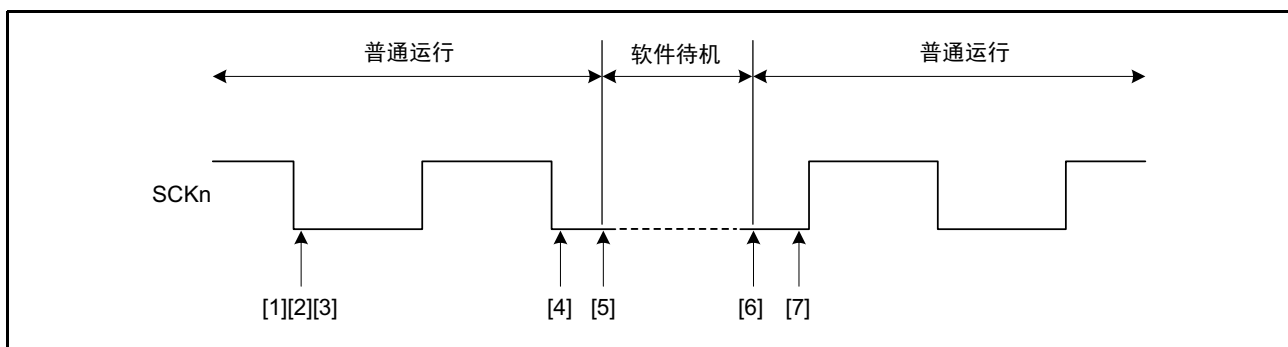


图 22.34 时钟的停止和重新启动的步骤

## 22.4 噪声消除功能

噪声消除功能的结构如图 22.35 所示。在异步模式中，能对 RXD 的输入信号使用噪声消除功能。在噪声消除功能有效时，经由噪声消除电路将接收信号 RXD 取到内部。噪声消除电路由 3 段锁存电路和匹配检测电路构成。在 SEMR.ABCS=0 时，通过频率为 16 倍传送率的时钟将 RXD 的接收电平取到 3 段锁存电路，如果 3 个锁存结果匹配，就将该电平传到后段。如果不匹配，就保持上一次匹配时的电平。

在至少保持了 3 个时钟的相同电平时，噪声消除功能将 RXD 视为有效的接收信号。但是，在未满 3 个时钟的脉冲状变化时，噪声消除功能不将 RXD 视为接收信号，而是判断为噪声。

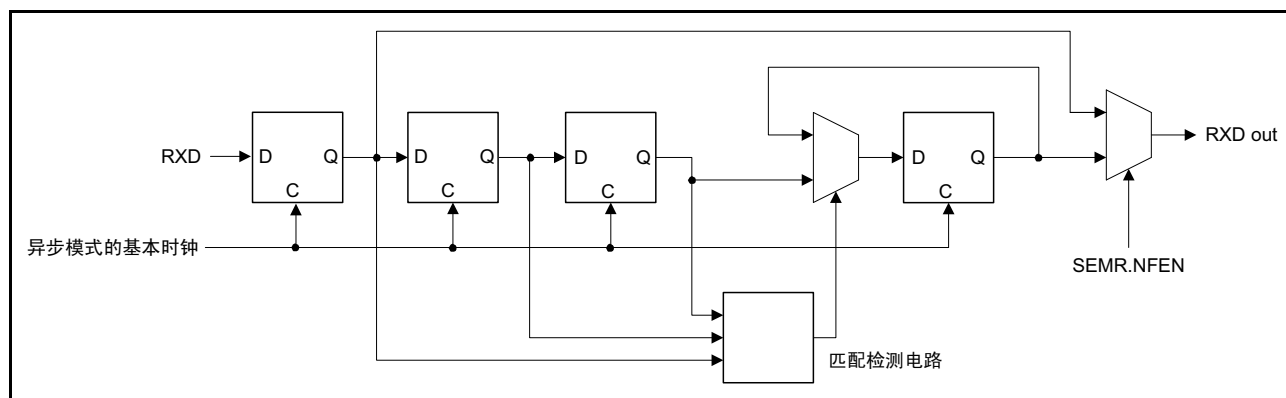


图 22.35 噪声消除功能的结构

## 22.5 中断源

### 22.5.1 串行通信接口模式的中断

串行通信接口模式的中断源如表 22.18 所示。各中断源分配有不同的中断向量，能通过 SCR 寄存器的允许位分别允许各中断源。

如果将发送数据从 TDR 寄存器（发送数据寄存器）传送到 TSR 寄存器，就在 SCR.TIE 位为“1”时产生 TXI 中断请求。通过在将 SCR.TIE 位置“1”后将 SCR.TE 位置“1”，或者用 1 条指令将 SCR.TIE 位和 SCR.TE 位同时置“1”，产生 TXI 中断请求。能通过 TXI 中断请求，在启动 DTC 后进行数据传送。

如果给 RDR 寄存器设定接收数据，就在 SCR.RIE 位为“1”时产生 RXI 中断请求。能通过 RXI 中断请求，在启动 DTC 后进行数据传送。

在 SSR.ORER 标志、SSR.FER 标志或者 SSR.PER 标志为“1”并且 SCR.RIE 位为“1”的情况下，产生 ERI 中断请求。此时，不产生 RXI 中断请求。

如果在发送数据的最后 1 位时未更新 TDR 寄存器，SSR.TEND 标志就变为“1”，并且在 SCR.TEIE 位为“1”时产生 TEI 中断请求。如果通过 TXI 中断处理给 TDR 寄存器写数据，SSR.TEND 标志就变为“0”并且 TEI 中断请求被取消。在通过给 TDR 寄存器写发送数据，将 SSR.TEND 标志置“0”时，必须读 SSR.TEND 标志，确认该标志已变为“0”。

通过在将 SCR.TIE 位置“1”后将 SCR.TE 位置“1”，或者用 1 条指令将 SCR.TIE 位和 SCR.TE 位同时置“1”，产生 TXI 中断请求。即使在 SCR.TIE 位为“0”的状态下将 SCR.TE 位或者 SCR.TIE 位置“1”，也不产生 TXI 中断请求。因此，在发送最后的数据时，必须暂时禁止 TXI 中断，在进行发送结束中断处理后要重新开始数据发送时，不通过 SCR.TIE 位而通过 TXI 中断对应的 ICU.IERm.IENj 位控制中断的禁止和允许。

表 22.18 SCI 中断源

名称	中断源	中断标志	DTC 的启动	优先级
ERI	接收错误	ORER、FER、PER	不能	高 ↑ 低
RXI	接收数据满	—	能	
TXI	发送数据空	—	能	
TEI	发送结束	TEND	不能	

### 22.5.2 智能卡接口模式的中断

智能卡接口模式的中断源如表 22.19 所示，不能使用发送结束中断 (TEI)。

表 22.19 SCI 中断源

名称	中断源	中断标志	DTC 的启动	优先级
ERI	接收错误、错误信号检测	ORER、PER、ERS	不能	高
RXI	接收数据满	—	能	↑
TXI	发送数据空	TEND	能	低

智能卡接口模式和一般的 SCI 相同，能通过 DTC 进行发送和接收。如果在发送时 SSR.TEND 标志变为“1”，就产生 TXI 中断请求。如果预先将 TXI 中断请求设定为 DTC 的启动源，就通过 TXI 中断请求，在启动 DTC 后传送发送数据。在通过 DTC 进行数据传送时，TEND 标志自动变为“0”。

如果发生错误，SCI 就自动重新发送相同的数据。在此期间，TEND 标志保持“0”，不启动 DTC。因此，SCI 和 DTC 自动发送指定字节数的数据，包括发生错误时的重新发送。但是，在发生错误时，SSR.ERS 标志不自动变为“0”，因此必须先将 SCR.RIE 位置“1”，然后在发生错误时产生 ERI 中断请求，并且将 ERS 标志置“0”。

在通过 DTC 进行发送和接收时，必须先将 DTC 设定为允许状态后进行 SCI 的设定。有关 DTC 的设定方法，请参照“14. 数据传送控制器 (DTC)”。

如果在接收时将接收数据设定到 RDR 寄存器，就产生 RXI 中断请求。如果预先将 RXI 中断请求设定为 DTC 的启动源，就通过 RXI 中断请求，在启动 DTC 后传送接收数据。如果发生错误，错误标志就被置位。因此，不启动 DTC 而向 CPU 请求 ERI 中断，所以必须将错误标志置“0”。

## 22.6 使用时的注意事项

### 22.6.1 模块停止功能的设定

能通过模块停止控制寄存器 B (MSTPCRB) 禁止或者允许 SCI 的运行, 初始值为停止 SCI 的运行。能通过解除模块停止状态, 使寄存器变为可存取的状态。详细内容请参照“9. 低功耗功能”。

### 22.6.2 有关中止的检测和处理

在检测到帧错误时, 能通过直接读 RXDn 引脚的值检测中止。因为在中止时 RXDn 引脚的输入全部为“0”, 所以 SSR.FER 标志变为“1” (发生帧错误), SSR.PER 标志也有可能变为“1” (发生奇偶校验错误)。SCI 在接收到中止后继续进行接收。因此必须注意: 即使将 FER 标志置“0” (未发生帧错误), FER 标志也会再次变为“1”。

### 22.6.3 标记状态和中止的发送

当 SCR.TE 位为“0” (禁止串行发送) 时, TXDn 引脚变为由 PORTi.DR.Bj 位和 PORTi.DDR.Bj 位决定的输入 / 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 TE 位置“1” (允许串行发送) 前将通信线路置为标记状态 (“1” 的状态), 必须将 POTRi.DDR.Bj 位和 PORTi.DR.Bj 位置“1”。当 TE 位为“0” 时, 因为 TXDn 引脚变为 I/O 端口, 所以输出“1”。

另外, 要在发送数据时发送中止的情况下, 必须在将 PORTi.DDR.Bj 位置“1” 并且将 PORTi.DR.Bj 位置“0” 后, 将 TE 位置“0”。如果将 TE 位置“0”, 就对发送部进行初始化, 与当前的发送状态无关, TXDn 引脚变为 I/O 端口, 并且从 TXDn 引脚输出“0”。

### 22.6.4 有关接收错误标志和发送 (只限于时钟同步模式)

在接收错误标志 (SSR.ORER、SSR.FER、SSR.PER) 为“1” 的状态下, 即使给 TDR 寄存器写数据, 也不能开始发送。在开始发送时, 必须将接收错误标志置“0”。必须注意: 即使将 SCR.RE 位置“0” (禁止串行接收), 接收错误标志也不能变为“0”。

### 22.6.5 有关写 TDR

能随时给 TDR 寄存器写数据。但是, 如果在 TDR 寄存器留有发送数据的状态下给 TDR 寄存器写新数据, 保存在 TDR 寄存器的数据就可能没有被传送到 TSR 寄存器而丢失。因此, 必须通过 TXI 中断请求给 TDR 寄存器写发送数据。

### 22.6.6 时钟同步发送时的限制事项

在将外部时钟源用于同步时钟时, 必须在通过 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。

### 22.6.7 使用 DTC 时的限制事项

在通过 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。

### 22.6.8 有关低功耗状态时的运行

#### (1) 发送

必须在停止运行 (SCR.TIE 位 =0, TE 位 =0, TEIE 位 =0) 的状态下设定为模块停止状态或者向软件待机模式转移。通过将 TE 位置“0”，TSR 寄存器和 SSR.TEND 标志变为“0”。模块停止状态以及软件待机模式时的输出引脚的状态取决于端口的设定，在解除模块停止状态或者软件待机模式后，为 High 电平输出。如果在发送时转移到这些模式，发送就被中止。

如果在解除低功耗状态后不更改发送模式而进行发送，就必须按照将 TE 位置“1”以及读 SSR 寄存器 → 写 TDR 寄存器的顺序开始发送。如果在更改发送模式后进行发送，就必须在进行初始设定后开始发送。

发送时的模式转移的流程图例子如图 22.36 所示，模式转移时的端口的引脚状态如图 22.37 和图 22.38 所示。

必须在停止运行的状态下，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。

如果在解除低功耗状态后不更改接收模式而进行接收，就必须通过将 RE 位置“1”，开始接收。如果在更改接收模式后进行接收，就必须在进行初始设定后开始接收。

接收时的模式转移的流程图例子如图 22.39 所示。

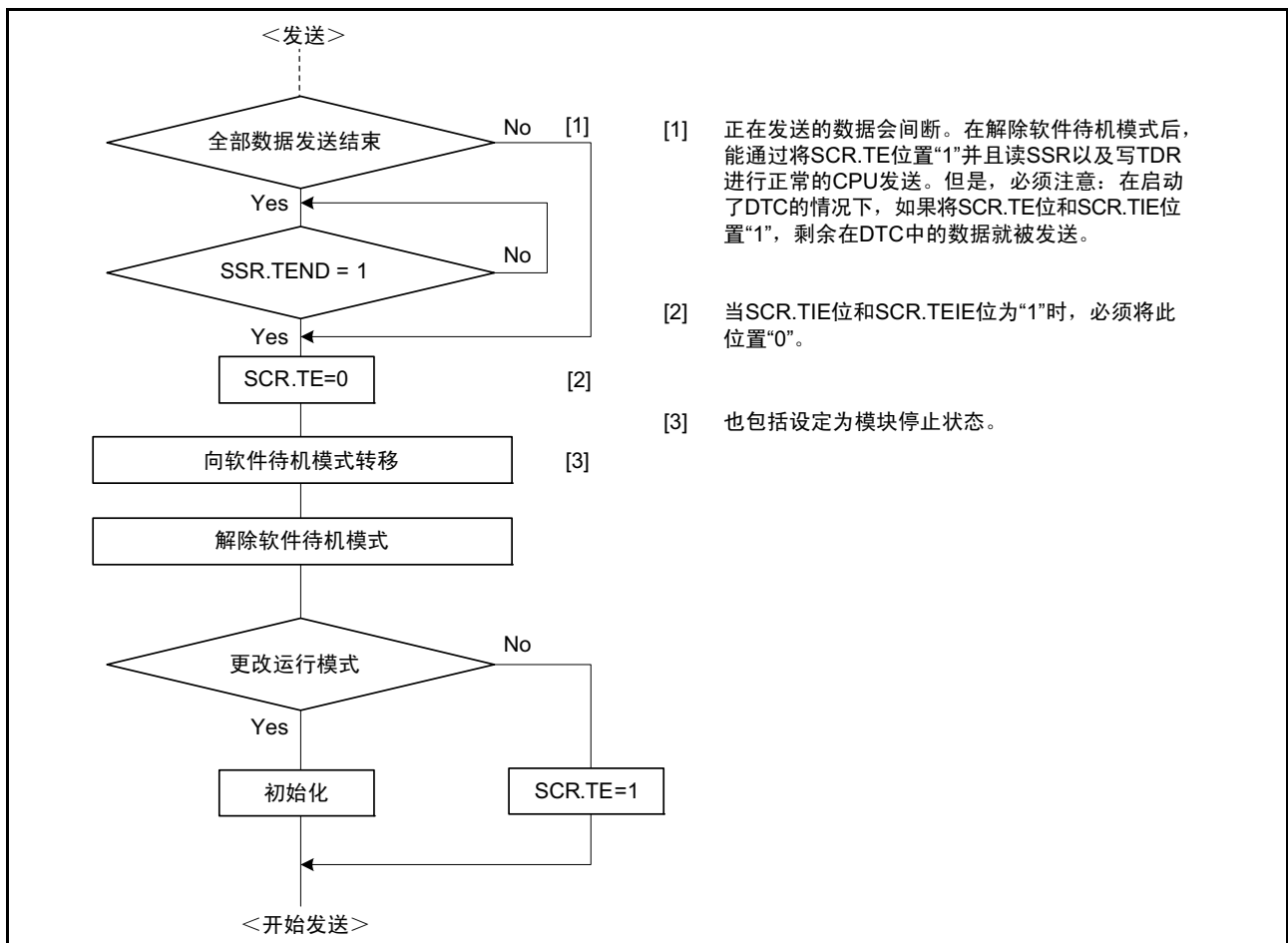


图 22.36 发送时向软件待机模式转移的流程图例子

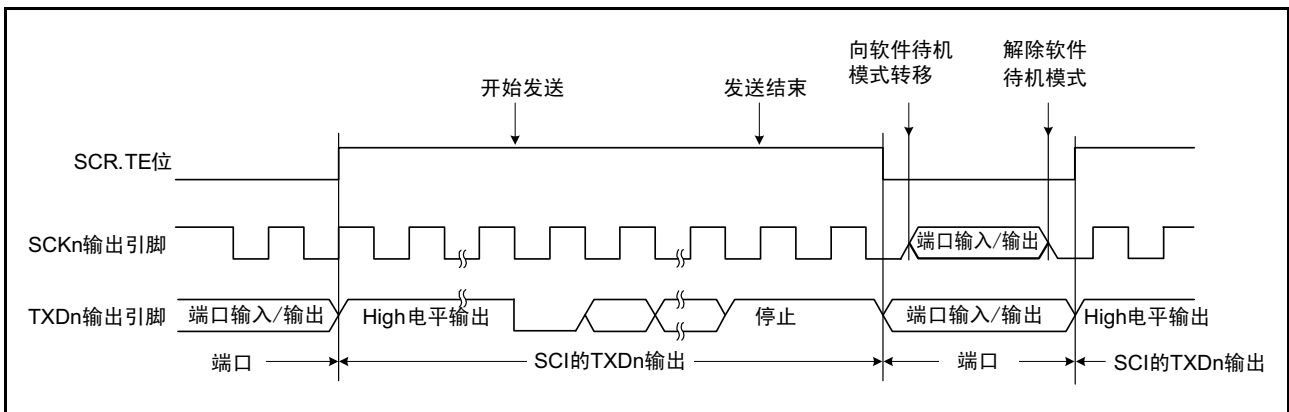


图 22.37 向软件待机模式转移时的端口引脚状态 (内部时钟、异步发送)

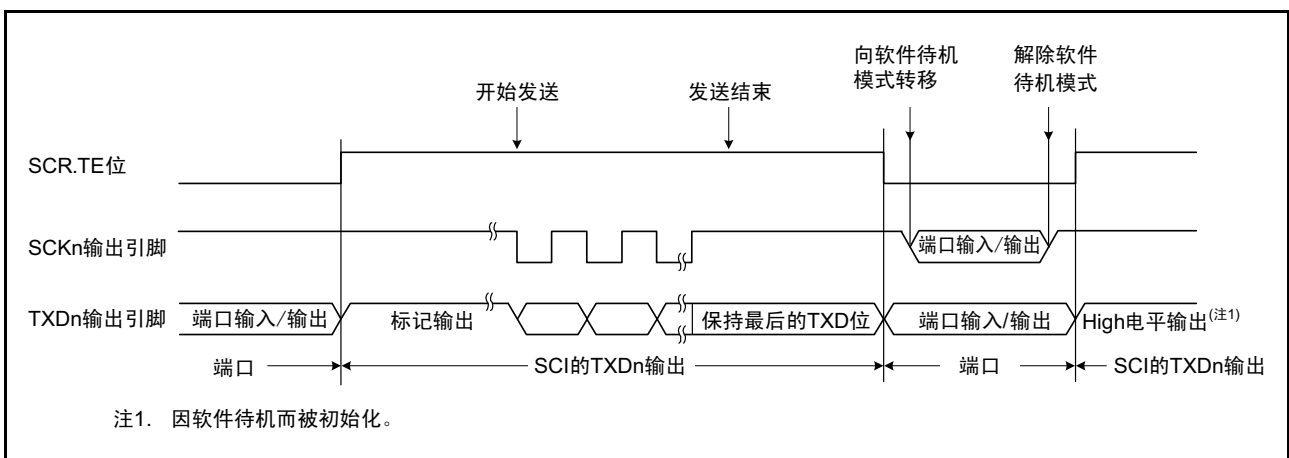


图 22.38 向软件待机模式转移时的端口引脚状态 (内部时钟、时钟同步发送)

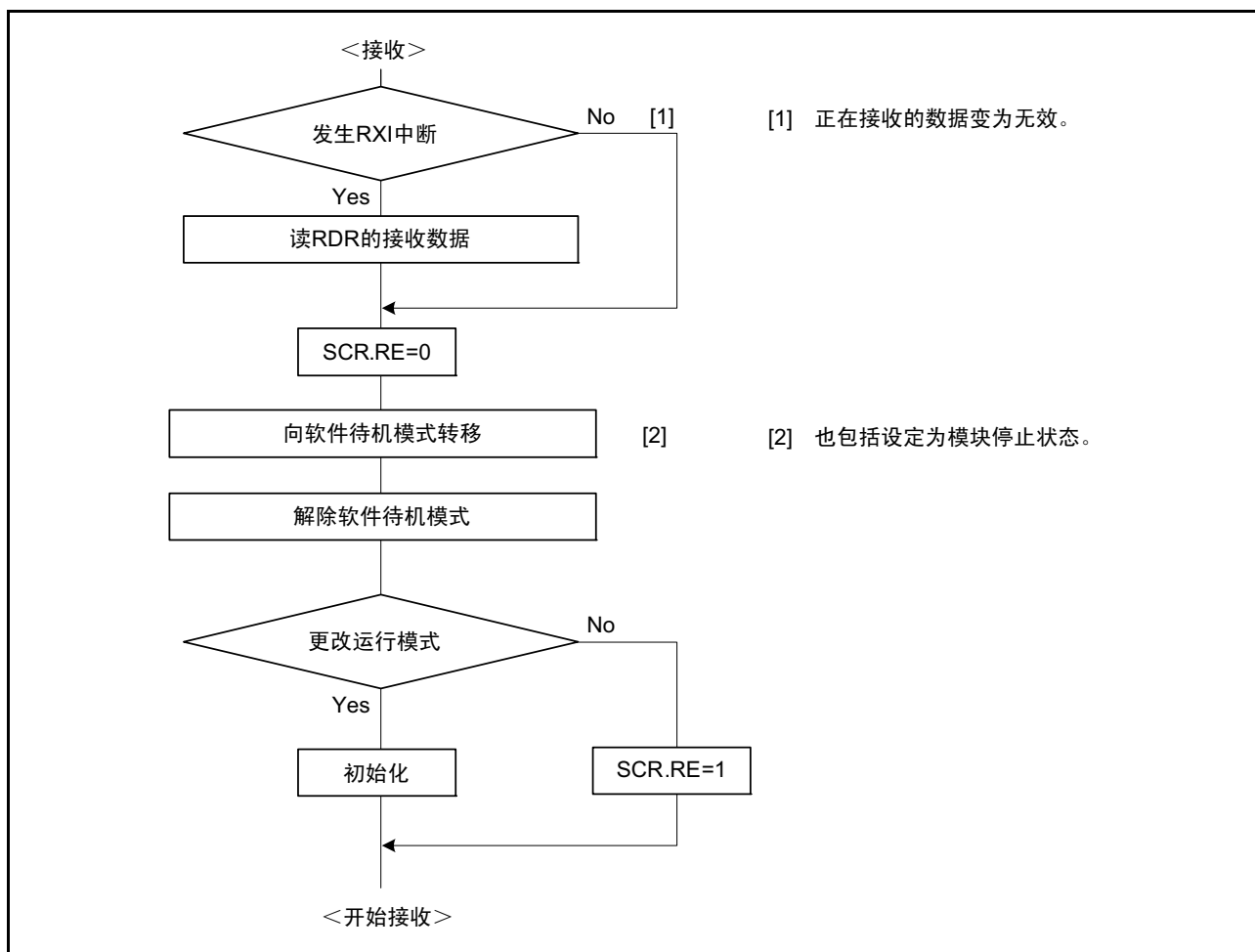


图 22.39 接收时向软件待机模式转移的流程图例子

### 22.6.9 时钟同步模式的外部时钟输入

在时钟同步模式中，外部时钟的 SCKn 输入必须至少为 2 个 PCLK 的 High 电平脉冲期间和 Low 电平脉冲期间，周期至少为 6 个 PCLK。



## 23. CRC 运算器 (CRC)

CRC (Cyclic Redundancy Check) 运算器生成数据块的 CRC 码。

### 23.1 概要

CRC 运算器的规格和框图分别如表 23.1 和图 23.1 所示。

表 23.1 CRC 运算器的规格

项目	内容
CRC 运算的对象数据 (注 1)	对应 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"> <li>• 8 位 CRC <math>X^8+X^2+X+1</math></li> <li>• 16 位 CRC <math>X^{16}+X^{15}+X^2+1</math> <math>X^{16}+X^{12}+X^5+1</math></li> </ul>
CRC 运算的转换	能选择生成用于 LSB first 通信的 CRC 码或者生成用于 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。

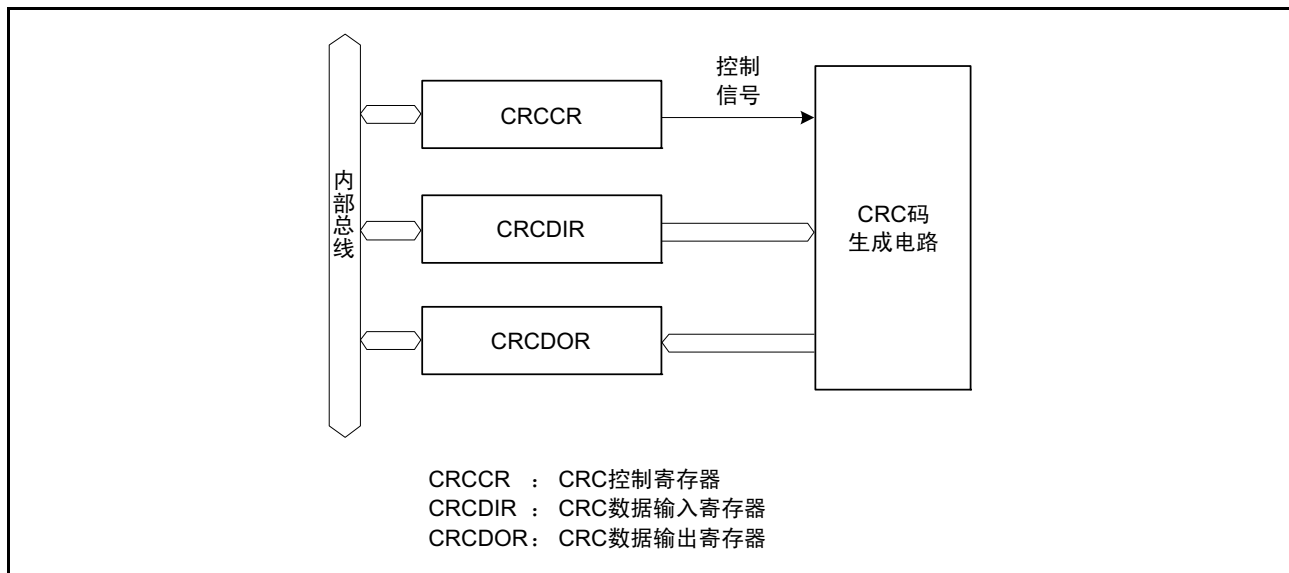


图 23.1 CRC 运算器的框图

## 23.2 寄存器说明

CRC 运算器的寄存器一览表如表 23.2 所示。

表 23.2 CRC 运算器的寄存器一览表

寄存器名	符号	复位后的值	地址	存取长度
CRC 控制寄存器	CRCCR	00h	0008 8280h	8
CRC 数据输入寄存器	CRCDIR	00h	0008 8281h	8
CRC 数据输出寄存器	CRCDOR	0000h	0008 8282h	16

### 23.2.1 CRC 控制寄存器 (CRCCR)

地址 0008 8280h

b7	b6	b5	b4	b3	b2	b1	b0
DORCLR	—	—	—	—	LMS	GPS[1:0]	
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b1-b0	GPS[1:0]	CRC 生成多项式转换位	$b1\ b0$ 0 0: 不运算 (注 1) 0 1: $X^8+X^2+X+1$ 1 0: $X^{16}+X^{15}+X^2+1$ 1 1: $X^{16}+X^{12}+X^5+1$	R/W
b2	LMS	CRC 运算转换位	0: 进行 LSB first 通信时的 CRC 运算 在将 CRCDOR 寄存器的值 (CRC 码) 以字节为单 外分开发送时, 必须先发送低位字节 (b7 ~ b0)。 1: 进行 MSB first 通信时的 CRC 运算 在将 CRCDOR 寄存器的值 (CRC 码) 以字节为单 外分开发送时, 必须先发送高位字节 (b15 ~ b8)。	R/W
b6-b3	—	保留位	读写值都为“0”。	R/W
b7	DORCLR	CRCDOR 寄存器清除位	0: 不影响运行 1: 清除 CRCDOR 寄存器 读取值为“0”。	R/W

注 1. CRC 数据输出寄存器 (CRCDOR) 的值总是为“0000h”。

CRCCR 寄存器是选择 CRC 运算器的初始化、运算转换和生成多项式的寄存器。

#### GPS[1:0] 位 (CRC 生成多项式转换位)

这些位选择 CRC 码的生成多项式。

#### LMS 位 (CRC 运算转换位)

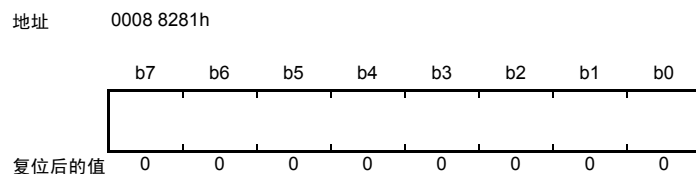
此位选择是生成 LSB first 通信的 CRC 码, 还是生成 MSB first 通信的 CRC 码。

#### DORCLR 位 (CRCDOR 寄存器清除位)

如果将 DORCLR 位置“1”, CRCDOR 寄存器就变为“0000h”。

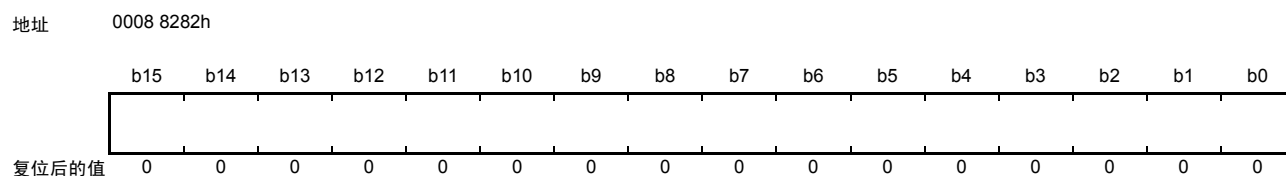
读取值为“0”。

### 23.2.2 CRC 数据输入寄存器 (CRCDIR)



CRCDIR 寄存器是设定 CRC 运算对象数据块的 8 位可读写寄存器。

### 23.2.3 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。不更新高位字节 (b15-b8)。

### 23.3 CRC 运算器的运行说明

CRC 运算器生成用于 LSB first/MSB first 通信的 CRC 码。

假设 CRCCR.GPS[1:0] 位为 “11b”，使用 16 位 CRC ( $X^{16}+X^{12}+X^5+1$  的多项式)，将数据 “F0h” 生成 CRC 码的使用例子如下所示。

如果使用 8 位 CRC ( $X^8+X^2+X+1$  的多项式)，CRCDOR 寄存器的低位字节就能得到有效的 CRC 码。

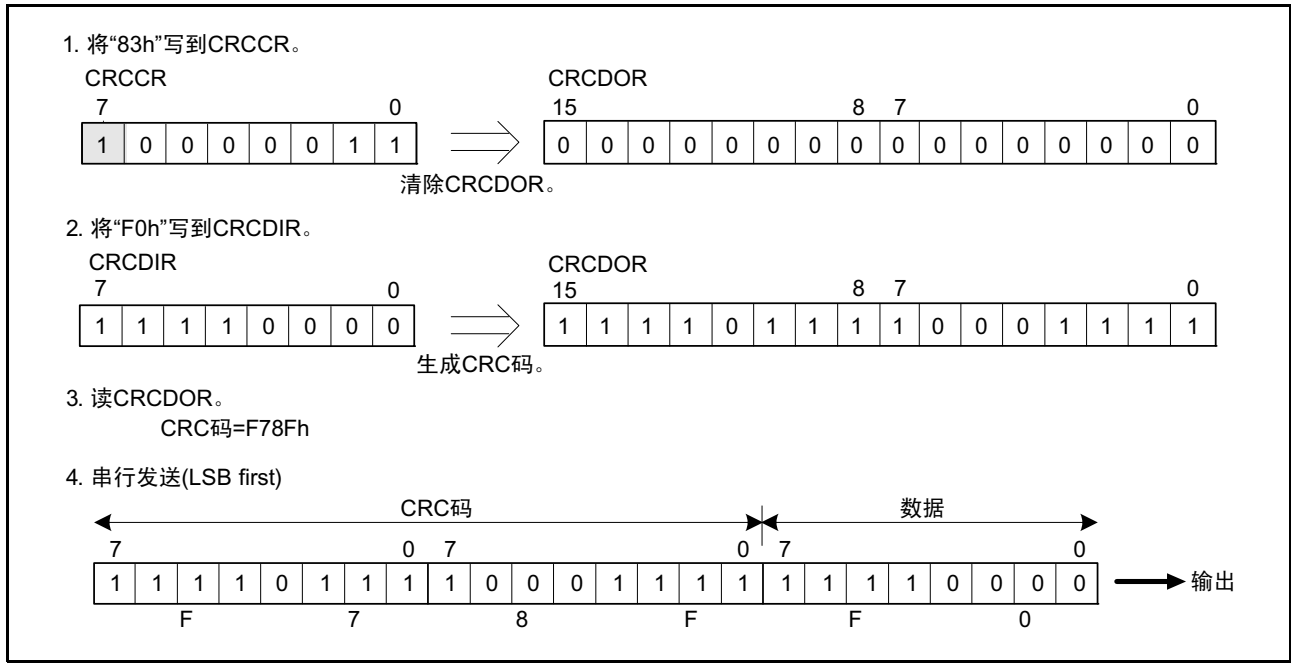


图 23.2 LSB first 的数据发送

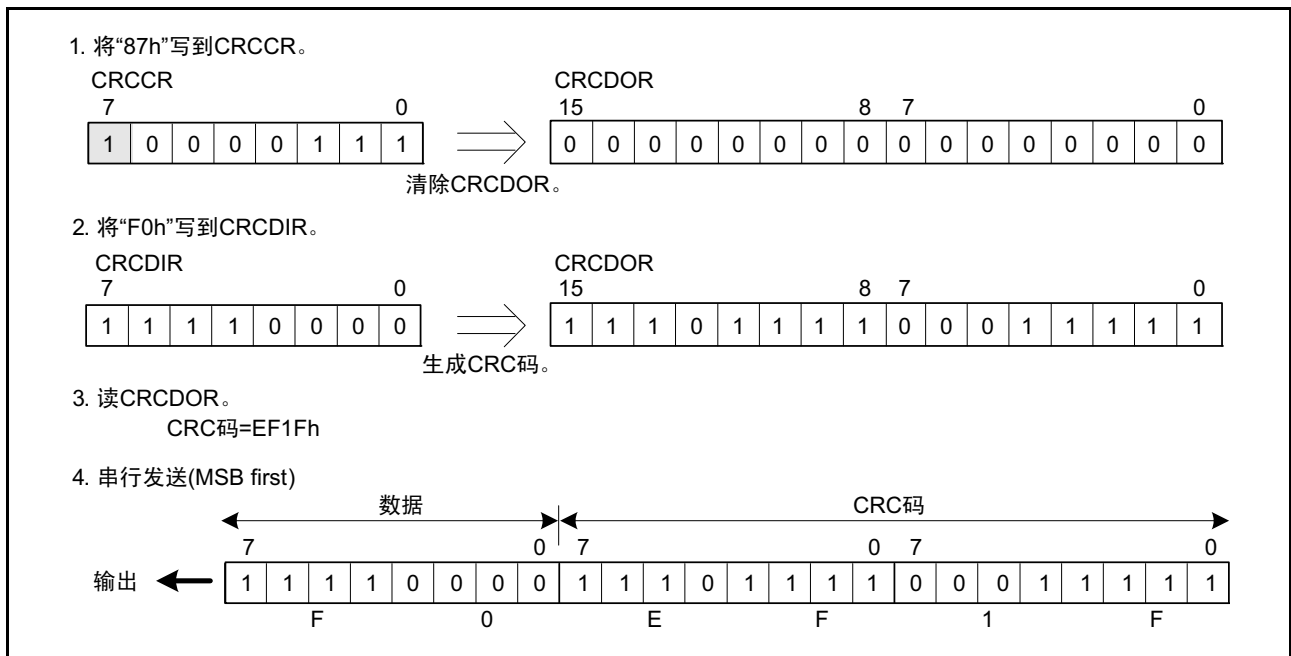


图 23.3 MSB first 的数据发送

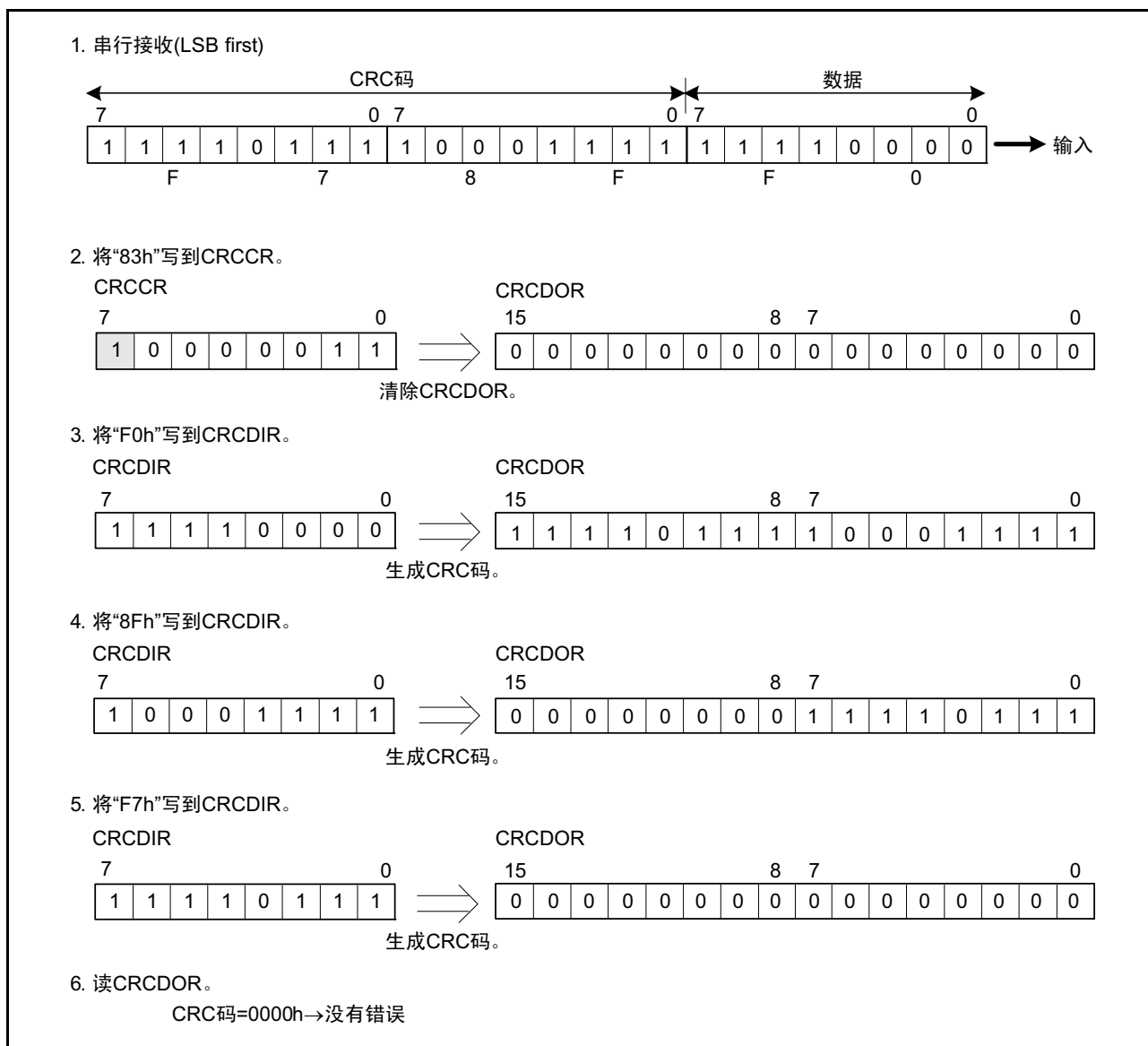


图 23.4 LSB first 的数据接收

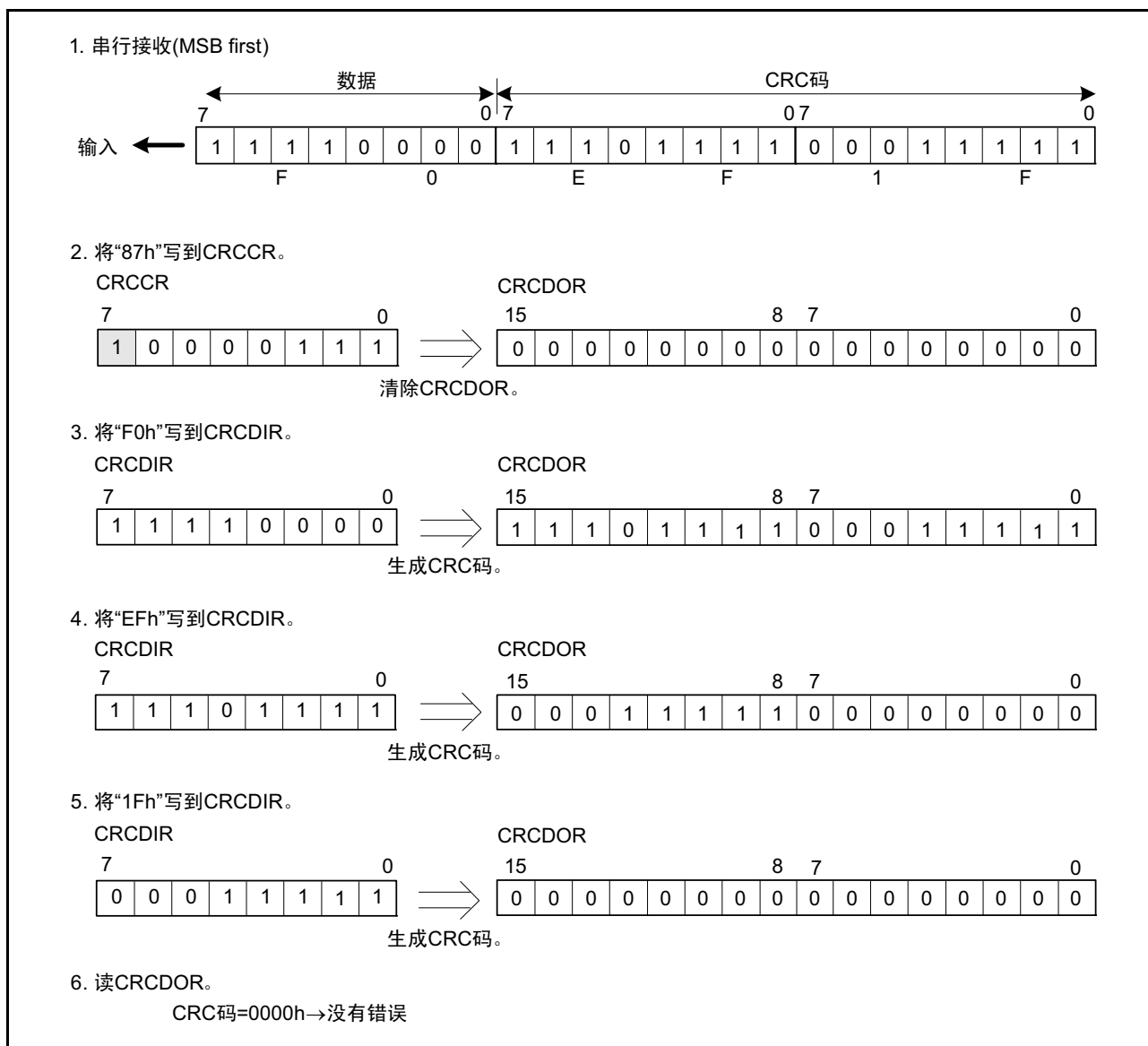


图 23.5 MSB first 的数据接收

## 23.4 使用时的注意事项

### 23.4.1 模块待机功能的设定

能通过模块待机控制寄存器 B (MSTPCRB) 禁止或者允许 CRC 运算器的运行, 初始值为停止 CRC 运算器的运行。能通过解除模块停止状态, 使寄存器变为可存取的状态。详细内容请参照“9. 低功耗功能”。

## 23.5 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。

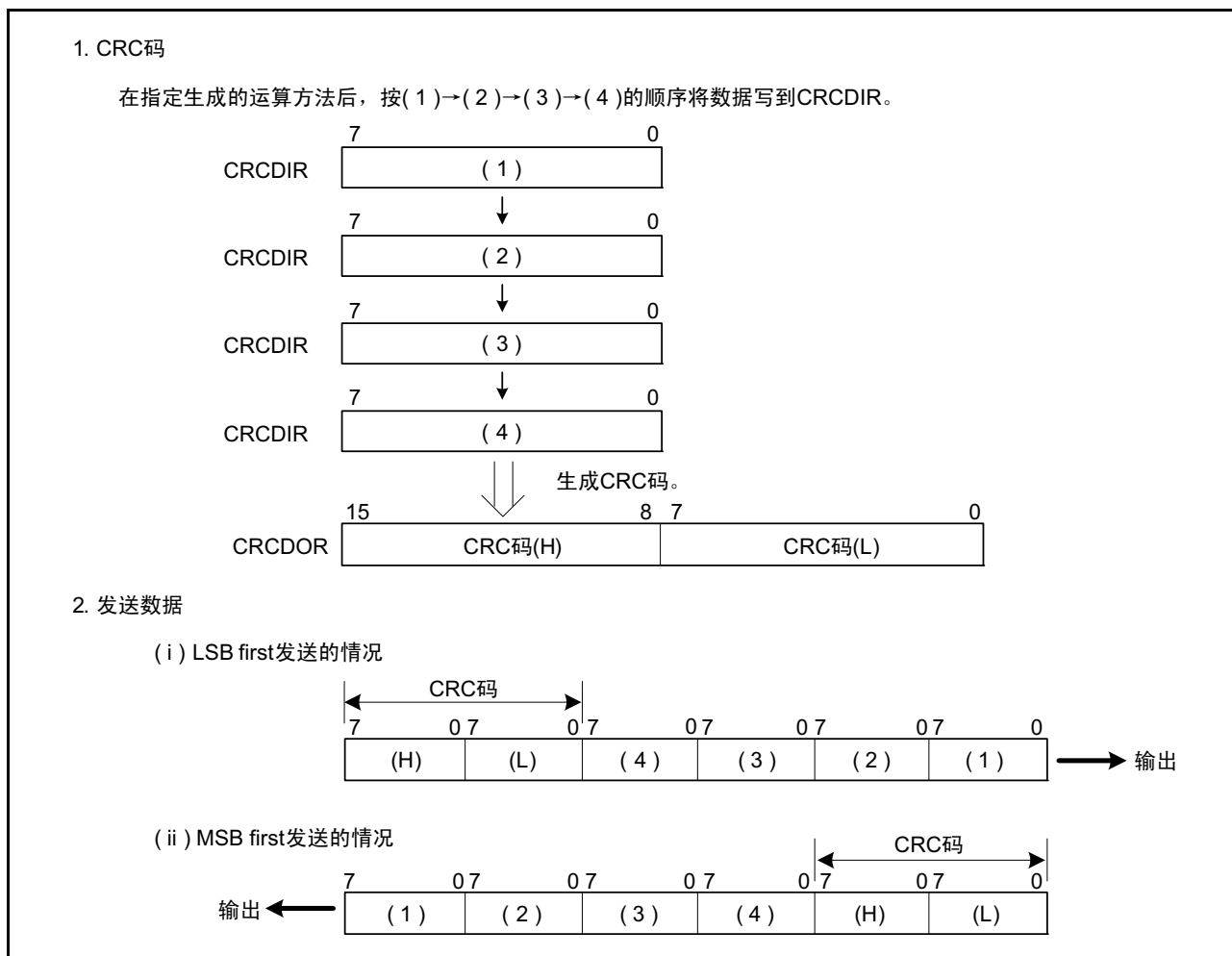


图 23.6 LSB first 和 MSB first 的发送数据

## 24. I<sup>2</sup>C 总线接口 (RIIC)

RX62T 群内置 1 个通道的 I<sup>2</sup>C 总线接口 (RIIC)。

RIIC 以 NXP 公司提倡的 I<sup>2</sup>C 总线 (Inter-IC-Bus) 接口方式为基准, 装载了子集功能。

### 24.1 概要

RIIC 的规格和框图分别如表 24.1 和图 24.1 所示, 输入 / 输出引脚的外部电路连接例子 (I<sup>2</sup>C 总线的结构例子) 如图 24.2 所示, RIIC 使用的输入 / 输出引脚如表 24.2 所示。

表 24.1 RIIC 的规格

项目	内容
通信格式	<ul style="list-style-type: none"> <li>• I<sup>2</sup>C 总线格式或者 SMBus 格式</li> <li>• 能选择主控模式或者从属模式。</li> <li>• 自动确保与所设的传送率对应的各种准备时间、保持时间和总线空闲时间。</li> </ul>
传送率	~ 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。
条件的发行和检测	自动生成开始条件、重新开始条件和停止条件, 并且能检测到开始条件 (包括重新开始条件) 和停止条件。
从属地址	<ul style="list-style-type: none"> <li>• 能设定 3 组从属地址。</li> <li>• 对应 7 位或者 10 位的地址格式 (能同时存在)。</li> <li>• 能检测到全呼地址、设备 ID 地址和 SMBus 的主机地址。</li> </ul>
应答	<ul style="list-style-type: none"> <li>• 在发送时自动加载应答位。</li> <li>能在接收 NACK 时自动中止下一个发送数据的传送。</li> <li>• 在接收时自动发送应答位。</li> <li>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。</li> </ul>
等待功能	<ul style="list-style-type: none"> <li>• 在接收时, 能通过保持 SCL 时钟的 Low 电平进行等待。</li> <li>在第 8 个时钟和第 9 个时钟之间等待。</li> <li>在第 9 个时钟和下次传送的第 1 个时钟之间等待 (WAIT 功能)。</li> </ul>
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"> <li>• 对应多主控</li> <li>在和其他主控发生 SCL 时钟冲突时, 能与 SCL 时钟同步运行。</li> <li>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。</li> <li>在 主控模式中, 能在发送数据不同时检测到仲裁失败。</li> <li>• 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)。</li> <li>• 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。</li> <li>• 在从属发送模式中, 能在数据不同时检测到仲裁失败。</li> </ul>
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"> <li>• 4 种</li> <li>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)</li> <li>接收数据满 (包括从属地址匹配的情况)</li> <li>发送数据空 (包括从属地址匹配的情况)</li> <li>发送结束</li> </ul>



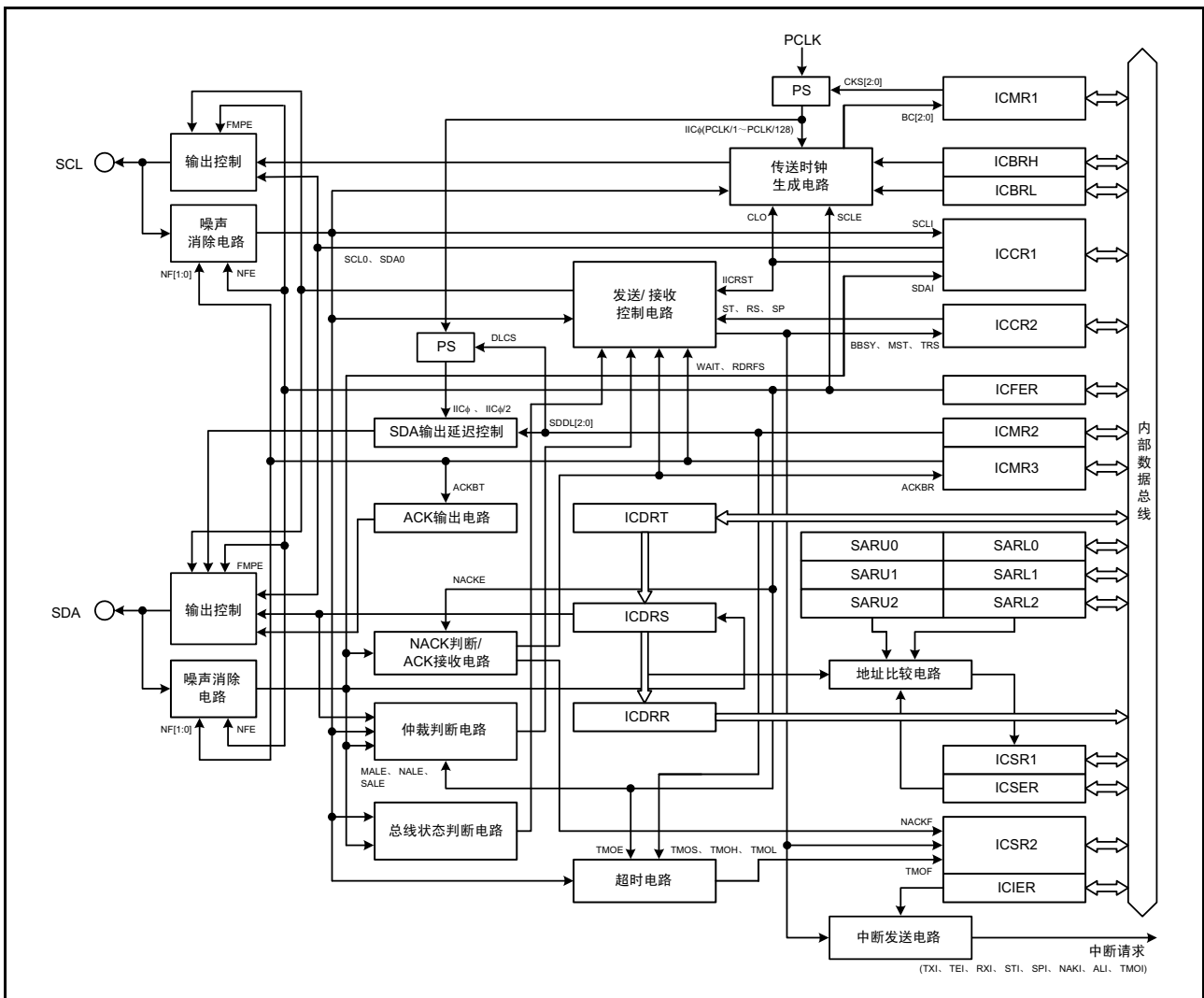


图 24.1 RIIC 的框图

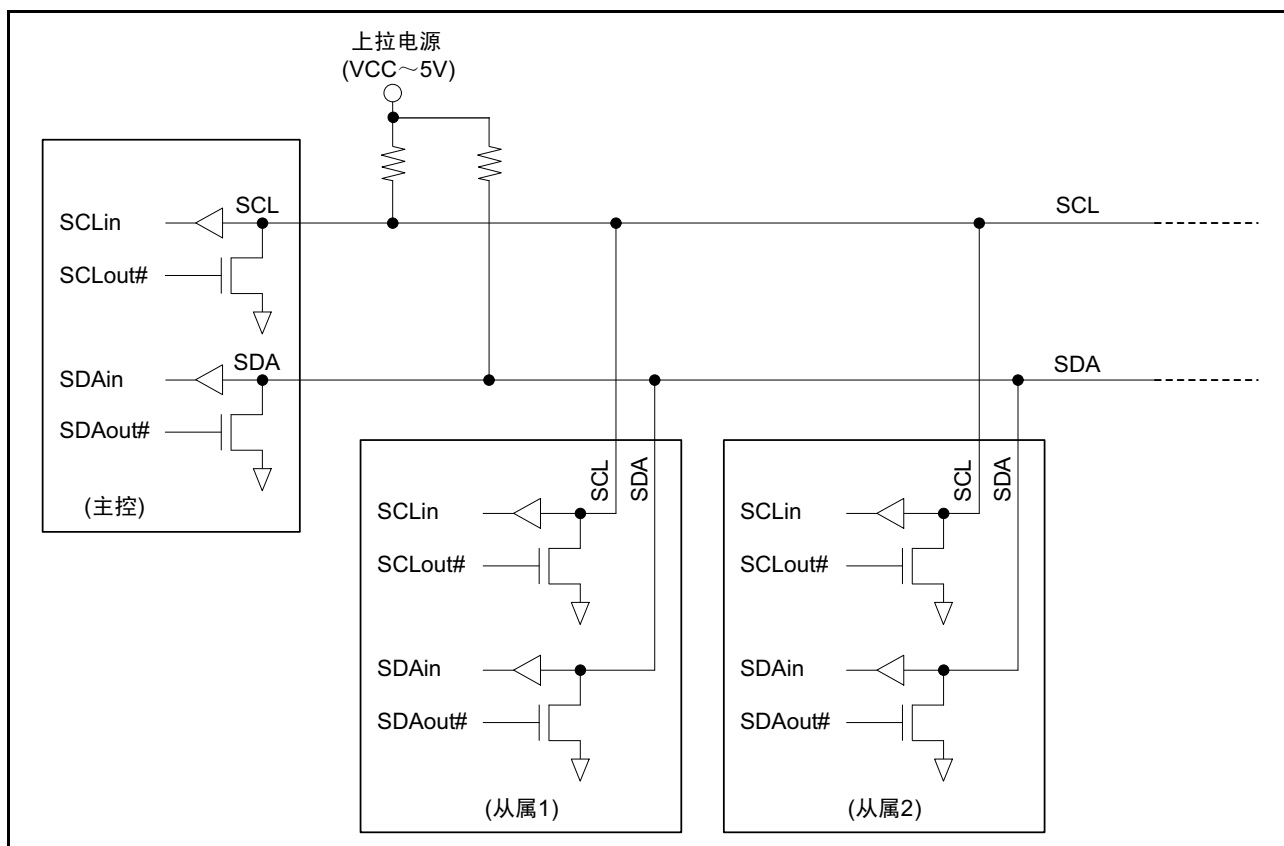


图 24.2 输入 / 输出引脚的外部电路连接例子 (I<sup>2</sup>C 总线的结构例子)

表 24.2 RIIC 的输入 / 输出引脚

引脚名	输入 / 输出	功能
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

## 24.2 寄存器说明

RIIC 的寄存器一览表如表 24.3 所示。

表 24.3 RIIC 的寄存器一览表

通道	寄存器名	符号	复位后的值	地址	存取长度
RIIC0	I <sup>2</sup> C 总线控制寄存器 1	ICCR1	1Fh	0008 8300h	8
	I <sup>2</sup> C 总线控制寄存器 2	ICCR2	00h	0008 8301h	8
	I <sup>2</sup> C 总线模式寄存器 1	ICMR1	08h	0008 8302h	8
	I <sup>2</sup> C 总线模式寄存器 2	ICMR2	06h	0008 8303h	8
	I <sup>2</sup> C 总线模式寄存器 3	ICMR3	00h	0008 8304h	8
	I <sup>2</sup> C 总线功能允许寄存器	ICFER	72h	0008 8305h	8
	I <sup>2</sup> C 总线状态允许寄存器	ICSER	09h	0008 8306h	8
	I <sup>2</sup> C 总线中断允许寄存器	ICIER	00h	0008 8307h	8
	I <sup>2</sup> C 总线状态寄存器 1	ICSR1	00h	0008 8308h	8
	I <sup>2</sup> C 总线状态寄存器 2	ICSR2	00h	0008 8309h	8
	从属地址寄存器 L0	SARL0	00h	0008 830Ah	8
	超时内部计数器 L	TMOCNTL	0000h	0008 830Ah	16
	从属地址寄存器 U0	SARU0	00h	0008 830Bh	8
	超时内部计数器 U	TMOCNTU	0000h	0008 830Bh	16
	从属地址寄存器 L1	SARL1	00h	0008 830Ch	8
	从属地址寄存器 U1	SARU1	00h	0008 830Dh	8
	从属地址寄存器 L2	SARL2	00h	0008 830Eh	8
	从属地址寄存器 U2	SARU2	00h	0008 830Fh	8
	I <sup>2</sup> C 总线位速率低电平寄存器	ICBRL	FFh	0008 8310h	8
	I <sup>2</sup> C 总线位速率高电平寄存器	ICBRH	FFh	0008 8311h	8
	I <sup>2</sup> C 总线发送数据寄存器	ICDRT	FFh	0008 8312h	8
	I <sup>2</sup> C 总线接收数据寄存器	ICDRR	00h	0008 8313h	8
	I <sup>2</sup> C 总线移位寄存器	ICDRS	—	—	8

24.2.1 I<sup>2</sup>C 总线控制寄存器 1 (ICCR1)

地址 0008 8300h

b7	b6	b5	b4	b3	b2	b1	b0
ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
0	0	0	1	1	1	1	1

复位后的值

位	符号	位名	功能	R/W
b0	SDAI	SDA 总线输入监视位	0: SDA 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"> <li>读时</li> <li>0: SDA 引脚为 Low 电平输出</li> <li>1: SDA 引脚为高阻抗</li> <li>写时</li> <li>0: 将 SDA 引脚改为 Low 电平输出</li> <li>1: 将 SDA 引脚改为高阻抗 (通过外部上拉电阻输出 High 电平)</li> </ul>	R/W (注1、注2)
b3	SCLO	SCL 输出控制位	<ul style="list-style-type: none"> <li>读时</li> <li>0: SCL 引脚为 Low 电平输出</li> <li>1: SCL 引脚为高阻抗</li> <li>写时</li> <li>0: 将 SCL 引脚改为 Low 电平输出</li> <li>1: 将 SCL 引脚改为高阻抗 (通过外部上拉电阻输出 High 电平)</li> </ul>	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 <sup>2</sup> 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 <sup>2</sup> C 总线接口允许位	0: RIIC 的功能停止 (SCL 引脚 /SDA 引脚为端口功能) 1: RIIC 能进行传送 (SCL 引脚 /SDA 引脚为总线驱动功能)	R/W

注 1. 不能在通信过程中写此位。如果在通信过程中更改此位的值, 就可能发生发送 / 接收异常或者 AL 错误。

注 2. 必须在将 SOWP 位置“0”的同时改写 SDAO 位和 SCLO 位。

ICCR1 寄存器控制 RIIC 的运行 / 停止、RIIC 内部状态的复位、SCL 时钟的追加输出、SCL 引脚 /SDA 引脚的操作以及 SCL 引脚 /SDA 引脚的监视等。

## SDAI 位 (SDA 总线输入监视位)

此位表示 SDA 引脚的输入电平。

**SCLI 位 (SCL 总线输入监视位)**

此位表示 SCL 引脚的输入电平。

**SDAO 位 (SDA 输出控制位)**

此位控制 SDA 引脚的输出电平并且表示此引脚的输出状态。

**SCLO 位 (SCL 输出控制位)**

此位控制 SCL 引脚的输出电平并且表示此引脚的输出状态。

**SOWP 位 (SCLO/SDAO 写保护位)**

此位控制 SCLO 位和 SDAO 位的写操作。

**CLO 位 (SCL 时钟的追加输出位)**

此位具有以 1 个时钟为单位追加输出 SCL 时钟的功能，用于调试或者异常处理。

在一般情况下，必须将此位“0”。如果在正常的通信过程中使用此位，就可能引起通信错误。

有关此功能的详细内容，请参照“24.11.2 SCL 时钟追加输出功能”。

**IICRST 位 (I<sup>2</sup>C 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 24.4 所示。

RIIC 复位是指对包括 ICCR2.BBSY 标志在内的全部寄存器和内部状态进行复位；内部复位是指对位计数器 (ICMR1.BC[2:0] 位)、I<sup>2</sup>C 总线移位寄存器 (ICDRS)、I<sup>2</sup>C 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“24.14 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. 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 24.4 RIIC 复位的种类

IICRST	ICE	状态	内容
1	0	RIIC 复位	对 RIIC 的全部寄存器和内部状态进行复位。
	1	内部复位	对 ICMR1.BC[2:0] 位、ICSR1、ICSR2、ICDRS 寄存器和内部状态进行复位。

### ICE 位 (I<sup>2</sup>C 总线接口允许位)

此位选择 RIIC 是能进行传送还是功能停止。

如果将 ICE 位置“0” (功能停止), SCL 引脚 /SDA 引脚就为端口功能。如果在 ICE 位为“0”时将 IICRST 位置“1”, 就进行 RIIC 复位; 如果在 ICE 位为“1”时将 IICRST 位置“1”, 就进行内部复位。

另外, 为了防止意想不到的通信开始, 必须在设定 RIIC 的寄存器时将 ICE 位置“0” (功能停止) 并且在设定完全部寄存器后将 ICE 位置“1” (能传送状态)。

注. RX62T 群的引脚功能也分配了 I<sup>2</sup>C 总线引脚以外的其他功能。在用作 I<sup>2</sup>C 总线引脚 (SCL 引脚 /SDA 引脚) 时, 必须禁止其他功能的输出。因为 I<sup>2</sup>C 总线引脚的 SCL 引脚 /SDA 引脚都为输入 / 输出引脚, 因此必须将对应的 I/O 端口的 PORTn.DDR 寄存器设定为“0” (输入), 将 PORTn.ICR 寄存器设定为“1” (输入缓冲器有效)。

24.2.2 I<sup>2</sup>C 总线控制寄存器 2 (ICCR2)

地址 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 <sup>2</sup> C 总线为释放状态 (总线空闲状态) 1: I <sup>2</sup> C 总线为占有状态 (总线忙状态或者总线处于空闲期间)	R

注 1. 当 ICMR1.MTWP 位为“1”时，能写 MST 位和 TRS 位。

ICCR2 寄存器是控制 I<sup>2</sup>C 总线接口的寄存器，表示 I<sup>2</sup>C 总线占有状态或者释放状态。

## ST 位 (开始条件发行请求位)

此位请求向主控模式的转移以及开始条件的发行。

如果 ST 位为“1”，就请求发行开始条件，并且在 BBSY 标志为“0” (总线空闲) 时发行开始条件。有关发行开始条件的详细内容，请参照“24.10 开始条件、重新开始条件和停止条件的发行功能”。

[ 为“1”的条件 ]

- 写“1”时

[ 为“0”的条件 ]

- 写“0”时
- 结束开始条件的发行时
- ICSR2.AL 标志变为“1” (仲裁失败) 时
- 给 ICCR1.IICRST 位写“1”，并且进行 RIIC 复位或者内部复位时

注. 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发生仲裁失败。

**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“24.10 开始条件、重新开始条件和停止条件的发行功能”。

[为“1”的条件]

- 在 ICCR2.BBSY 标志为“1”的状态下写“1”时

[为“0”的条件]

- 写“0”时
- 结束重新开始条件的发行或者检测到开始条件时
- ICSR2.AL 标志变为“1”（仲裁失败）时
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

注 1. 不能在 BBSY 标志为“0”（总线空闲）的状态下写此位。

注 2. 不能在发行停止条件过程中将 RS 位置“1”。

注 3. 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“24.10 开始条件、重新开始条件和停止条件的发行功能”。

[为“1”的条件]

- 在 ICCR2.BBSY 标志为“1”并且 ICCR2.MST 位为“1”的状态下写“1”时

[为“0”的条件]

- 写“0”时
- 结束停止条件的发行或者检测到停止条件时
- ICSR2.AL 标志变为“1”（仲裁失败）时
- 检测到开始条件和重新开始条件时
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

注 1. 不能在 BBSY 标志为“0”（总线空闲）时写此位。

注 2. 不能在发行重新开始条件过程中将 SP 位置“1”。

**TRS 位 (发送 / 接收模式位)**

此位是表示发送模式或者接收模式的位。

当 TRS 位为“0”时，为接收模式；当 TRS 位为“1”时，为发送模式。通过和 MST 位的组合，表示 RIIC 的运行模式。

TRS 位通过开始条件的发行和检测以及 R/W# 位等变为“1”或者“0”，并且自动转移到发送模式或者接收模式。当 ICMR1.MTWP 位为“1”时，能写 TRS 位，但是一般不需要写。

[为“1”的条件]

- 根据开始条件的发行请求，正常地发行了开始条件（在 ST 位为“1”的状态下检测到开始条件）时
- 在主控模式中，附加到从属地址的 R/W# 位为“0”时
- 在从属模式中，接收的从属地址与 ICSER 寄存器的有效地址匹配，并且在 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 ]

- 检测到停止条件时
- ICSR2.AL 标志变为“1” (仲裁失败) 时
- 在主控模式中, 附加到从属地址的R/W#位为“1”时
- 在从属模式中, 接收的从属地址与ICSER寄存器的有效地址匹配, 并且在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
- 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”, 并且自动转移到主控模式或者从属模式。当 ICMR1.MTWP 位为“1”时能写 MST 位, 但是一般不需要写。

[ 为“1”的条件 ]

- 根据开始条件的发行请求, 正常地发行了开始条件 (在 ST 位为“1”的状态下检测到开始条件) 时
- 在 ICMR1.MTWP 位为“1”的状态下写“1”时

[ 为“0”的条件 ]

- 检测到停止条件时
- ICSR2.AL 标志变为“1” (仲裁失败) 时
- 在 ICMR1.MTWP 位为“1”的状态下写“0”时
- 给 ICCR1.IICRST 位写“1”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 I<sup>2</sup>C 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 ]

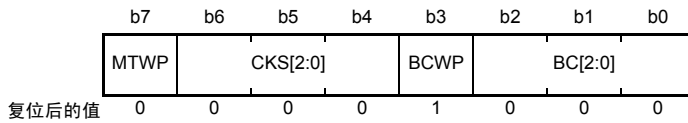
- 检测到开始条件时

[ 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.IICRST 位为“0”的状态下给 ICCR1.IICRST 位写“1” (RIIC 复位) 时

24.2.3 I<sup>2</sup>C 总线模式寄存器 1 (ICMR1)

地址 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。

ICMR1 寄存器选择 RIIC 的内部基准时钟源，取传送位数信息以及控制 ICCR2.MST 位和 ICCR2.TRS 位的写保护。

**BC[2:0] 位 (位计数器)**

这些位是在 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”。

**BCWP 位 (BC 写保护位)**

此位控制 BC[2:0] 位的写操作。

**CKS[2:0] 位 (内部基准时钟选择位)**

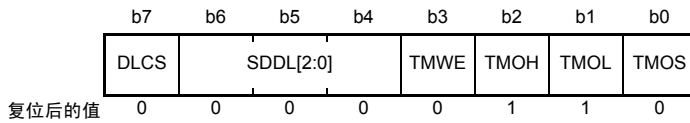
这些位选择 RIIC 的内部基准时钟源 (IIC $\phi$ )。

**MTWP 位 (MST/TRS 写保护位)**

此位控制 ICCR2.MST 位和 ICCR2.TRS 位的写操作。

24.2.4 I<sup>2</sup>C 总线模式寄存器 2 (ICMR2)

地址 0008 8303h



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"> <li>当 ICMR2.DLCS=0 (IIC<math>\phi</math>) 时</li> </ul> b6 b4 0 0 0: 无输出延迟 0 0 1: 1 个 IIC $\phi$ 周期 0 1 0: 2 个 IIC $\phi$ 周期 0 1 1: 3 个 IIC $\phi$ 周期 1 0 0: 4 个 IIC $\phi$ 周期 1 0 1: 5 个 IIC $\phi$ 周期 1 1 0: 6 个 IIC $\phi$ 周期 1 1 1: 7 个 IIC $\phi$ 周期 <ul style="list-style-type: none"> <li>当 ICMR2.DLCS=1 (IIC<math>\phi</math>/2) 时</li> </ul> b6 b4 0 0 0: 无输出延迟 0 0 1: 1 ~ 2 个 IIC $\phi$ 周期 0 1 0: 3 ~ 4 个 IIC $\phi$ 周期 0 1 1: 5 ~ 6 个 IIC $\phi$ 周期 1 0 0: 7 ~ 8 个 IIC $\phi$ 周期 1 0 1: 9 ~ 10 个 IIC $\phi$ 周期 1 1 0: 11 ~ 12 个 IIC $\phi$ 周期 1 1 1: 13 ~ 14 个 IIC $\phi$ 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 $\phi$ ) 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. 只有在 SCL 为 Low 电平时, DLCS=1 (IIC $\phi$ /2) 的选择有效。当 SCL 为 High 电平时, DLCS=1 的设定无效, 为内部基准时钟 (IIC $\phi$ )。

ICMR2 寄存器是具有超时检测功能和 SDA 输出延迟功能的寄存器。

**TMOS 位 (超时检测时间选择位)**

此位是在超时检测功能有效时 (ICFER.TMOE 位 =1) 选择超时检测时间的位。如果将此位置“0”，就为长模式；如果置“1”，就为短模式。在长模式中，用于检测超时的内部计数器用作 16 位计数器；在短模式中，此内部计数器用作 14 位计数器。在 SCL 线为 TMOH 位或者 TMOL 位选择的状态时，将内部基准时钟 (IIC $\phi$ ) 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“24.11.1 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“24.5 SDA 输出延迟功能”。

注 1. 必须注意：SDA 输出延迟的设定必须符合 I<sup>2</sup>C 总线规格 (数据有效时间 / 应答有效时间 (注 2) 内) 或者 SMBus 规格 (数据保持时间：至少为 300ns，SCL 时钟的 Low 电平宽度的数据准备时间：250ns 的范围内)，否则就可能引起通信设备的通信故障，或者根据总线状态视如开始条件或者停止条件。

注 2. 数据有效时间 / 应答有效时间

3450ns (~ 100kbps : 标准模式 [Sm])

900ns (~ 400kbp : 快速模式 [fm])

**DLCS 位 (SDA 输出延迟时钟源选择位)**

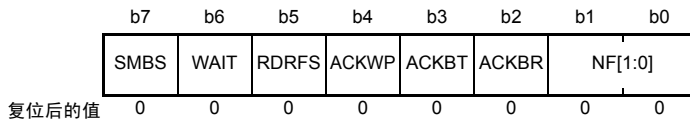
此位选择内部基准时钟 (IIC $\phi$ ) 或者内部基准时钟的 2 分频 (IIC $\phi$ /2) 时钟为 SDA 输出延迟时间的时钟源。

只有在 SCL 为 Low 电平时，DLCS=1 (IIC $\phi$ /2) 的选择有效。

当 SCL 为 High 电平时，DLCS=1 的设定无效，为内部基准时钟 (IIC $\phi$ )。

24.2.5 I<sup>2</sup>C 总线模式寄存器 3 (ICMR3)

地址 0008 8304h



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 $\phi$ 的噪声 (滤波器为 1 段) 0 1: 消除不超过 2 个 IIC $\phi$ 的噪声 (滤波器为 2 段) 1 0: 消除不超过 3 个 IIC $\phi$ 的噪声 (滤波器为 3 段) 1 1: 消除不超过 4 个 IIC $\phi$ 的噪声 (滤波器为 4 段)	R/W
b2	ACKBR	接收应答位	0: 应答位接收“0”(ACK 接收) 1: 应答位接收“1”(NACK 接收)	R
b3	ACKBT	发送应答位	0: 应答位发送“0”(ACK 发送) 1: 应答位发送“1”(NACK 发送)	R/W (注 1)
b4	ACKWP	ACKBT 写保护位	0: 禁止写 ACKBT 位 1: 允许写 ACKBT 位	W (注 1)
b5	RDRFS	RDRF 标志的置位时序选择位	0: 在 SCL 时钟的第 9 个时钟上升时, 此位为“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 <sup>2</sup> C 总线选择位	0: 选择 I <sup>2</sup> C 总线 1: 选择 SMBus	R/W

注 1. 即使在写 ACKBT 位的同时将 ACKWP 位置“1”, 也不能写 ACKBT 位。

注 2. WAIT 位和 RDRFS 位只在接收模式中有效, 在发送模式中无效。

ICMR3 寄存器控制应答发送 / 接收功能、RIIC 接收时的 RDRF 标志以及 WAIT 运行等。

## NF[1:0] 位 (噪声滤波器的段数选择位)

这些位选择数字噪声滤波器的段数。

注. 必须注意: 用噪声滤波器消除的噪声宽度必须小于 SCL 线的高/低电平宽度。

如果设定大于等于 (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 $\phi$ ) + 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.ICRST 位写“1” (RIIC 复位) 时

**ACKBT 位 (发送应答位)**

在接收模式中, 此位设定应答时要发送的位。

[ 为“1”的条件 ]

- 在 ACKWP 位为“1”的状态下写“1”时

[ 为“0”的条件 ]

- 在 ACKWP 位为“1”的状态下写“0”时
- 检测到停止条件的发行 (在 ICCR2.SP 位为“1”的状态下检测到停止条件) 时
- 在 ICCR1.ICE 位为“0”的状态下给 ICCR1.ICRST 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/I<sup>2</sup>C 总线选择位)**

如果将 SMBS 位置“1”, 就选择 SMBus 并且 IC SER.HOAE 位有效。

24.2.6 I<sup>2</sup>C 总线功能允许寄存器 (ICFER)

地址 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	超时检测功能允许位	0: 超时检测功能无效 1: 超时检测功能有效	R/W
b1	MALE	主控仲裁失败检测允许位	0: 禁止主控仲裁失败的检测 (将仲裁失败检测功能设定为无效, 在发生仲裁失败时不自动清除 ICCR2.MST 位和 ICCR2.TRS 位) 1: 允许主控仲裁失败的检测 (将仲裁失败检测功能设定为有效, 在发生仲裁失败时自动清除 ICCR2.MST 位和 ICCR2.TRS 位)	R/W
b2	NALE	NACK 发送仲裁失败检测允许位	0: 禁止 NACK 发送仲裁失败的检测 1: 允许 NACK 发送仲裁失败的检测	R/W
b3	SALE	从属仲裁失败检测允许位	0: 禁止从属仲裁失败的检测 1: 允许从属仲裁失败的检测	R/W
b4	NACKE	NACK 接收传送中止允许位	0: 在接收 NACK 时, 不中止传送 (禁止中止传送)。 1: 在接收 NACK 时, 中止传送 (允许中止传送)。	R/W
b5	NFE	数字噪声滤波器电路允许位	0: 不使用数字噪声滤波器电路 1: 使用数字噪声滤波器电路	R/W
b6	SCLE	SCL 同步电路允许位	0: SCL 同步电路无效 1: SCL 同步电路有效	R/W
b7	—	保留位	读写值都为“0”。	R/W

ICFER 寄存器控制是否使用超时检测功能、仲裁失败、接收 NACK 时的接收、SCL 同步电路和数字噪声滤波器电路。

**TMOE 位 (超时检测功能允许位)**

此位选择超时检测功能的有效或者无效。

有关超时检测功能的详细内容, 请参照“24.11.1 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“24.8.2 NACK 接收传送中止功能”。

**NFE 位 (数字噪声滤波器电路允许位)**

此位选择是否使用数字噪声滤波器电路。

**SCLE 位 (SCL 同步电路允许位)**

对于 SCL 输入时钟，此位选择是否与 SCL 时钟同步。一般必须将此位置“1”。

如果将 SCLE 位置“0”（SCL 同步电路无效），就不与时钟同步。在此设定下，与 SCL 线的状态无关，RIIC 输出 ICBRH 寄存器和 ICBRL 寄存器所设传送率的 SCL 时钟。因此必须注意：在 I<sup>2</sup>C 总线的总线负载远远大于规格值时或者多主控模式中的 SCL 时钟输出发生重叠时，有可能变为规格外的短时钟。在 SCL 同步电路无效的情况下，也影响开始条件、重新开始条件、停止条件的发行以及 SCL 时钟追加输出的连续输出。

只在确认是否输出了所设传送率时才能将 SCLE 位置“0”。



24.2.7 I<sup>2</sup>C 总线状态允许寄存器 (ICSER)

地址 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

ICSER 寄存器选择从属地址比较、全呼地址检测、设备 ID 地址检测、主机地址检测的有效或者无效。

## SARyE 位 (从属地址寄存器 y 允许位) (y=0 ~ 2)

此位选择 SARLy 寄存器和 SARUy 寄存器设定的从属地址是否有效。

如果将 SARyE 位置“1”，SARLy 寄存器和 SARUy 寄存器的设定值就有效，并且和接收的从属地址进行比较。

如果将 SARyE 位置“0”，SARLy 寄存器和 SARUy 寄存器的设定值就无效，即使与接收的从属地址匹配，也忽视此设定值。

## GCAE (全呼地址允许位)

在接收到全呼地址 (0000 000b+0[W]: All“0”) 时，此位选择是否忽视此地址。

在 GCAE 位为“1”时，如果接收的从属地址与全呼地址匹配，RIIC 就将此从属地址视为全呼地址并且进行接收，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, 并且当后续的 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“24.7.3 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。

24.2.8 I<sup>2</sup>C 总线中断允许寄存器 (ICIER)

地址 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

ICIER 寄存器是选择是否使用各种中断源的寄存器。

## TMOIE 位 (超时中断允许位)

当 ICSR2.TMOF 标志为“1”时, 此位选择允许或者禁止超时中断 (TMOI)。能通过将 TMOF 标志或者 TMOIE 位置“0”来解除 TMOI 中断。

## ALIE 位 (仲裁失败中断允许位)

当 ICSR2.AL 标志为“1”时, 此位选择允许或者禁止仲裁失败中断 (ALI)。能通过将 AL 标志或者 ALIE 位置“0”来解除 ALI 中断。

## STIE 位 (开始条件检测中断允许位)

当 ICSR2.START 标志为“1”时, 此位选择允许或者禁止开始条件检测中断 (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)。

24.2.9 I<sup>2</sup>C 总线状态寄存器 1 (ICSR1)

地址 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 匹配判断帧的 SCL 时钟的第 9 个时钟上升时，此标志变为“1”。）	R(W) (注 1)
b2	AAS2	从属地址 2 的检测标志	0: 未检测到从属地址 2 1: 检测到从属地址 2 • 在 SARU2.FS 位为“0”（选择 7 位地址格式）的情况下，接收的从属地址与 SARL2.SVA[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 • 接收的从属地址与全呼地址（All“0”）匹配时	R(W) (注 1)
b4	—	保留位	读写值都为“0”。	R/W
b5	DID	设备 ID 地址的检测标志	0: 未检测到设备 ID 地址 1: 检测到设备 ID 地址 • 紧接在开始条件后的第 1 帧与设备 ID 地址（1111 100b）+0[W] 匹配时	R(W) (注 1)
b6	—	保留位	读写值都为“0”。	R/W
b7	HOA	主机地址的检测标志	0: 未检测到主机地址 1: 检测到主机地址 • 接收的从属地址与主机地址（0001 000b）匹配时	R(W) (注 1)

注 1. 只能写“0”。

ICSR1 寄存器是确认各种地址检测状态的寄存器。

#### AASy 标志 (从属地址 y 的检测标志) (y=0 ~ 2)

[为“1”的条件]

【选择 7 位地址格式时: SARUy.FS 位 =0】

- 在 IC SER.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 IC SER.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]

- 读“1”后写“0”时
- 检测到停止条件时
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

【选择 7 位地址格式时: SARUy.FS 位 =0】

- 在 IC SER.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 IC SER.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 IC SER.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 IC SER.GCAE 位为“1” (全呼地址的检测有效) 的状态下接收的从属地址与全呼地址 (0000 000b +0[W]) 匹配时, 该帧的 SCL 时钟的第 9 个时钟上升。

[为“0”的条件]

- 读“1”后写“0”时
- 检测到停止条件时
- 在 IC SER.GCAE 位为“1” (全呼地址的检测有效) 的状态下接收的从属地址与全呼地址 (0000 000b +0[W]) 不匹配时, 该帧的 SCL 时钟的第 9 个时钟上升。
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

#### DID 标志 (设备 ID 地址的检测标志)

[为“1”的条件]

- 在 IC SER.DIDE 位为“1” (设备 ID 地址的检测有效) 的状态下检测到开始条件或者重新开始条件后的第 1 帧与设备 ID 地址 (1111 100b) +0[W] 匹配时, 该帧的 SCL 时钟的第 9 个时钟上升。

[为“0”的条件]

- 读“1”后写“0”时
- 检测到停止条件时
- 在 IC SER.DIDE 位为“1” (设备 ID 地址的检测有效) 的状态下检测到开始条件或者重新开始条件检测后的第 1 帧与设备 ID 地址 (1111 100b) 不匹配时, 该帧的 SCL 时钟的第 9 个时钟上升。
- 在 IC SER.DIDE 位为“1” (设备 ID 地址的检测有效) 的状态下检测到开始条件或者重新开始条件检测后的第 1 帧与设备的 ID 地址 (1111 100b) +0[W] 匹配, 而后续的第 2 帧与从属地址 0~2 全部不匹配时, 该帧的 SCL 时钟的第 9 个时钟上升。
- 给 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 ]

- 读“1”后写“0”时
- 检测到停止条件时
- 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。
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

24.2.10 I<sup>2</sup>C 总线状态寄存器 2 (ICSR2)

地址 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”。

ICSR2 寄存器是确认各种中断请求标志及其状态的寄存器。

## TMOF 标志 (超时检测标志)

如果 SCL 线的状态在一定期间内不发生变化, 就视为超时, 此标志变为“1”。

[ 为“1”的条件 ]

- 当 ICFER.TMOE 位为“1” (超时检测功能有效), 并且在指定为主控模式或者从属模式的状态下由 ICMR2.TMOH 位、ICMR2.TMOL 位、ICMR2.TMOS 位所选的条件的期间内, SCL 线的状态没有变化时

[ 为“0”的条件 ]

- 读“1”后写“0”时
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

## AL 标志 (仲裁失败标志)

在发行开始条件或者发送地址和数据时, 此标志表示因总线竞争等而失去了总线占有权 (仲裁失败)。RIIC 在发送过程中监视 SDA 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 =1】

- 在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 引脚为高阻抗状态）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 =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 =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 ]

- 读“1”后写“0”时
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

表 24.5 仲裁失败发生源和各仲裁失败允许功能的关系

ICFER			ICSR2	错误内容	仲裁失败发生源
MALE	NALE	SALE	AL		
1	*	*	1	开始条件发行错误	在 ICCR2.ST 位为“1”的状态下检测到开始条件时，RIIC 自身发送的 SDA 信号和 SDA 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	在发送模式中，发送数据（包括从属地址的发送）与总线状态不同。
*	1	*	1	发送的 NACK 不同	在接收模式或者从属接收模式中，在发送 NACK 时检测到 ACK。
*	*	1	1	发送数据不同	在从属发送模式中，发送数据与总线状态不同。

\*: Don't care

#### START 标志（开始条件检测标志）

[ 为“1”的条件 ]

- 检测到开始条件（包括重新开始条件）时

[ 为“0”的条件 ]

- 读“1”后写“0”时
- 检测到停止条件时
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

#### STOP 标志（停止条件检测标志）

[ 为“1”的条件 ]

- 检测到停止条件时

[ 为“0”的条件 ]

- 读“1”后写“0”时
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

**NACKF 标志 (NACK 检测标志)**

[ 为“1”的条件 ]

- 在 ICFER.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 ]

- 读“1”后写“0”时
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

注. 如果 NACKF 标志变为“1”, RIIC 就中止通信。即使在 NACKF 标志为“1”的状态下写 ICDRT 寄存器 (在发送模式中) 或者读 ICDRR 寄存器 (在接收模式中), 也不进行发送和接收。要重新开始通信时, 必须将 NACKF 标志置“0”。

**RDRF 标志 (接收数据满标志)**

[ 为“1”的条件 ]

- 在将接收数据从 ICDRS 寄存器传送到 ICDRR 寄存器时, 通过设定 ICMR3.RDRFS 位, 在 SCL 时钟的第 8 个时钟或者第 9 个时钟的上升沿 RDRF 标志变为“1”。
- 在检测到开始条件 (包括重新开始条件) 后接收的从属地址匹配并且 ICCR2.TRS 位为“0”时

[ 为“0”的条件 ]

- 读“1”后写“0”时
- 读 ICDRR 寄存器时
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

**TEND 标志 (发送结束标志)**

[ 为“1”的条件 ]

- 在 TDRE 标志为“1”的状态下 SCL 时钟的第 9 个时钟上升。

[ 为“0”的条件 ]

- 读“1”后写“0”时
- 给 ICDRT 寄存器写数据时
- 检测到停止条件时
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

**TDRE 标志 (发送数据空标志)**

[ 为“1”的条件 ]

- 将数据从 ICDRT 寄存器传送到 ICDRS 寄存器并且 ICDRT 寄存器为空时
- ICCR2.TRS 位变为“1”时
  - a. 在检测到开始条件 (包括重新开始条件) 后 ICCR2.MST 位为“1”时
  - b. 从接收模式变为发送模式时
  - c. 在 ICMR1.MTWP 位为“1”的状态下写“1”时
- 接收的从属地址匹配并且 TRS 位为“1”时

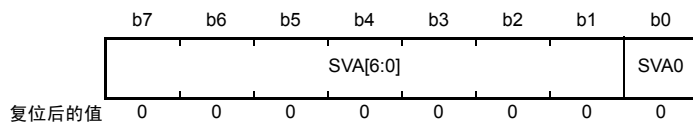
[ 为“0”的条件 ]

- 给 ICDRT 寄存器写数据时
- ICCR2.TRS 位变为“0”时
  - a. 检测到停止条件时
  - b. 从发送模式变为接收模式时
  - c. 在 ICMR1.MTWP 位为“1”的状态下写“0”时
- 给 ICCR1.IICRST 位写“1”并且进行 RIIC 复位或者内部复位时

注. 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”。

## 24.2.11 从属地址寄存器 Ly (SARLy) (y=0 ~ 2)

地址 SARL0 0008 830Ah  
SARL1 0008 830Ch  
SARL2 0008 830Eh



位	符号	位名	功能	R/W
b0	SVA0	10 位地址的最低有效位	必须设定从属地址。 • 当 SARUy.FS 位为“0”（选择 7 位地址格式）时，SVA0 位无效。 • 当 SARUy.FS 位为“1”（选择 10 位地址格式）时，SVA0 位有效。SVA0 位和 SVA[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

SARLy 寄存器是设定从属地址 y（7 位地址或者 10 位地址的低 8 位）的寄存器。

## SVA0 位（10 位地址的最低有效位）

在选择 10 位地址格式时（SARUy.FS 位=1），作为 10 位地址的最低有效位，此位和 SVA[6:0] 位合并，用作 10 位地址的低 8 位。

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。

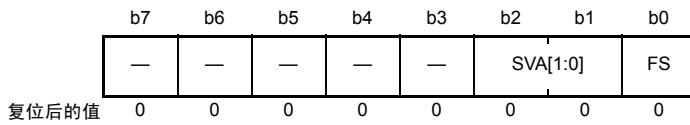
## SVA[6:0] 位（7 位地址 /10 位地址的低位）

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 位合并，用作 10 位地址的低 8 位。

当 ICSEr.SARyE 位为“0”时，忽视设定值。

## 24.2.12 从属地址寄存器 Uy (SARUy) (y=0 ~ 2)

地址 SARU0 0008 830Bh  
SARU1 0008 830Dh  
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” (7 位地址格式选择) 时, SVA[1:0] 位无效。 • 当 SARUy.FS 位为“1” (10 位地址格式选择) 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

SARUy 寄存选择 7 位 /10 位地址格式以及设定 10 位从属地址的高位。

## FS 位 (7 位 /10 位地址格式选择位)

此位选择从属地址 y (SARLy 寄存器和 SARUy 寄存器) 为 7 位地址或者 10 位地址。

当 ICSE.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 ICSE.SARyE 位为“1” (SARLy 寄存器和 SARUy 寄存器有效) 并且 SARUy.FS 位为“1” 时, 选择从属地址 m 为 10 位地址格式, SVA[1:0] 位和 SARLy 寄存器的设定值有效。

当 ICSE.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。

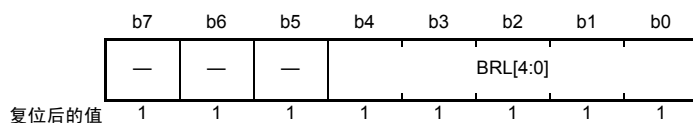
## SVA[1:0] 位 (10 位地址的高位)

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 ICSE.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。

24.2.13 I<sup>2</sup>C 总线位速率低电平寄存器 (ICBRL)

地址 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。

在发生 SCL 自动保持 Low 电平时（参照“24.8 SCL 的 Low 电平自动保持功能”），ICBRL 寄存器用作数据准备时间的确保寄存器。因此，在 RIIC 只用于从属模式的情况下，设定值不能小于数据准备时间（注 1）的值。

ICBRL 寄存器通过 ICMR1.CKS[2:0] 位选择的内部基准时钟源（IIC $\phi$ ），对 Low 电平宽度进行计数。

注 1. 数据准备时间 (t<sub>SU:DAT</sub>)

250ns (～100kbps : 标准模式[Sm])

100ns (～400kbps : 快速模式[fm])

24.2.14 I<sup>2</sup>C 总线位速率高电平寄存器 (ICBRH)

地址 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 寄存器是设定 SCL 时钟的 High 电平宽度的 5 位寄存器，在主控模式中有效。在 RIIC 总是用于从属模式的情况下，不需要设定 High 电平宽度。

ICBRH 寄存器通过 ICMR1.CKS[2:0] 位选择的内部基准时钟源 (IIC $\phi$ )，对 High 电平宽度进行计数。

用以下表达式计算 I<sup>2</sup>C 传送率和 SCL 时钟的占空比。

传送率 =  $1 / \{ [(ICBRH+1) + (ICBRL+1)] / IIC\phi \text{ (注 1)} + \text{SCL 线的上升时间 [tr]} + \text{SCL 线的下降时间 [tf]} \}$

占空比 =  $\{ \text{SCL 线的上升时间 [tr]} \text{ (注 2)} + (ICBRH+1 / IIC\phi) \} / \{ \text{SCLn 线的下降时间 [tf]} \text{ (注 2)} + (ICBRL+1 / IIC\phi) \}$

注 1.  $IIC\phi = PCLK \times 10^6 \times \text{分频比}$

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 I<sup>2</sup>C 总线规格书。

ICBRH 寄存器和 ICBRL 寄存器值的设定例子如表 24.6 所示。

表 24.6 对应传送率的 ICBRH 寄存器和 ICBRL 寄存器的设定例子

传送率 (kbps)	工作频率 PCLK (MHz)								
	8			10			12.5		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	100b	22 (F6h)	25 (F9h)	101b	13 (EDh)	15 (EFh)	101b	16 (F0h)	20 (F4h)
50	010b	16 (F0h)	19 (F3h)	010b	21 (F5h)	24 (F8h)	011b	12 (ECh)	15 (EFh)
100	001b	15 (EFh)	18 (F2h)	001b	19 (F3h)	23 (F7h)	001b	24 (F8h)	29 (FDh)
400	000b	4 (E4h)	10 (EAh)	000b	5 (E5h)	12 (ECh)	000b	7 (E7h)	16 (F0h)

传送率 (kbps)	工作频率 PCLK (MHz)								
	16			20			25		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	101b	22 (F6h)	25 (F9h)	110b	13 (EDh)	15 (EFh)	110b	16 (F0h)	20 (F4h)
50	011b	16 (F0h)	19 (F3h)	011b	21 (F5h)	24 (F8h)	100b	12 (ECh)	15 (EFh)
100	010b	15 (EFh)	18 (F2h)	010b	19 (F3h)	23 (F7h)	010b	24 (F8h)	29 (FDh)
400	000b	9 (E9h)	20 (F4h)	000b	11 (EBh)	25 (F9h)	001b	7 (E7h)	16 (F0h)

传送率 (kbps)	工作频率 PCLK (MHz)								
	30			33			50		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	26 (FAh)	111b	16 (F0h)	20 (F4h)
50	100b	15 (EFh)	18 (F2h)	100b	17 (F1h)	20 (F4h)	100b	26 (FAh)	31 (FFh)
100	010b	2 (E2h)	3 (E3h)	011b	16 (F0h)	19 (F3h)	011b	24 (F8h)	29 (FDh)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	21 (F5h)	010b	7 (E7h)	16 (F0h)

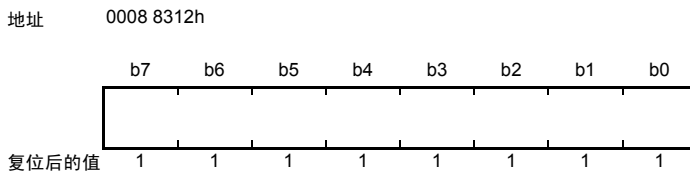
注. 计算的设定例子是假设:

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 I<sup>2</sup>C 总线规格书。

### 24.2.15 I<sup>2</sup>C 总线发送数据寄存器 (ICDRT)



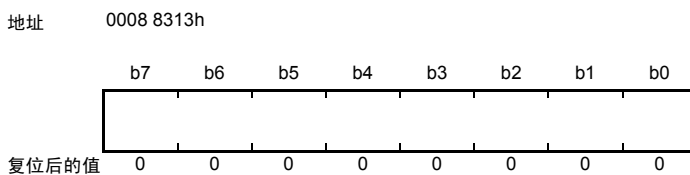
ICDRT 寄存器是保存发送数据的 8 位寄存器。

如果检测到 I<sup>2</sup>C 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。只能在产生发送数据空中断 (ICTXI) 请求时给 ICDRT 寄存器写 1 次发送数据。

### 24.2.16 I<sup>2</sup>C 总线接收数据寄存器 (ICDRR)



ICDRR 寄存器是保存接收数据的 8 位寄存器。

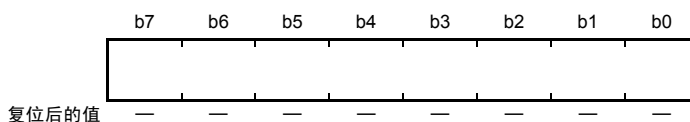
如果 1 字节数据接收结束, 就能将接收数据从 I<sup>2</sup>C 总线移位寄存器 (ICDRS) 传送到 ICDRR 寄存器, 进入能接收下一个数据的状态。

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, 就能进行连续接收。

不能写 ICDRR 寄存器。只能在产生接收数据满中断 (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。

### 24.2.17 I<sup>2</sup>C 总线移位寄存器 (ICDRS)



ICDRS 寄存器是发送和接收数据的移位寄存器。

在发送时, 将发送数据从 ICDRT 寄存器传送到 ICDRS 寄存器, 从 SDA 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。

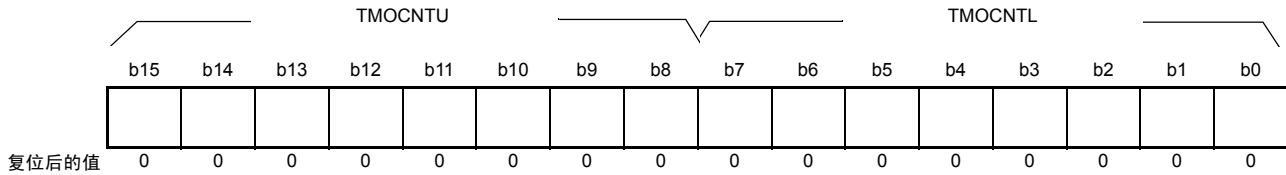
不能直接存取 ICDRS 寄存器。



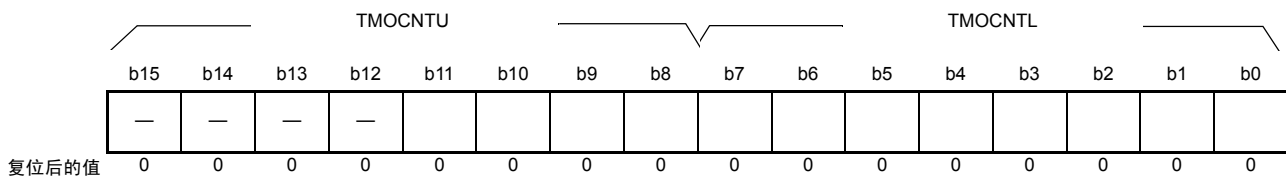
## 24.2.18 超时内部计数器 (TMOCNT)

地址 TMOCNLT 0008 830Ah、TMOCNTU 0008 830Bh

- ICMR2.TMOS=0 (长模式) 时



- ICMR2.TMOS=1 (短模式) 时



注. 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	TMOCNLT		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。虽然能写, 但是写入值无效。

注2. 不能读超时内部计数器的值。如果执行读操作, 读取值为“FFFFh”。

复位时, 如果 ICCR1.IICRST=1、或者 ICFER.TMOE=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 (TMOCNLT/TMOCNTU) 初始化为“0000h”。

## 24.3 运行说明

### 24.3.1 通信数据的格式

I<sup>2</sup>C 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。

I<sup>2</sup>C 总线的格式及其总线时序分别如图 24.3 和图 24.4 所示。

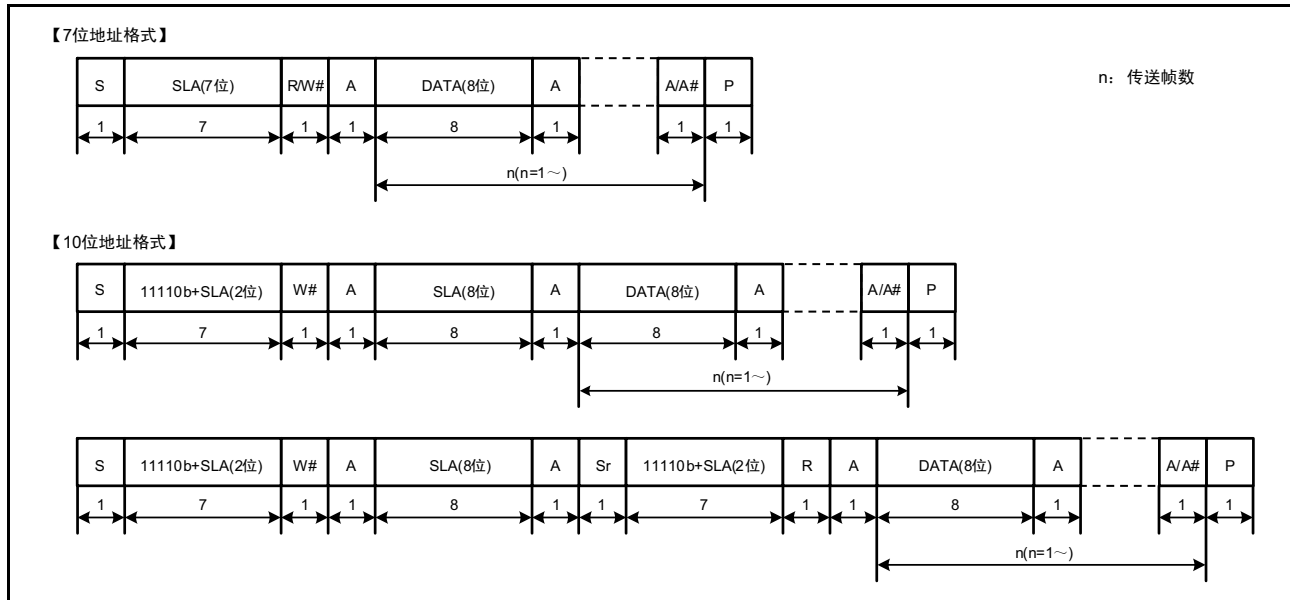


图 24.3 I<sup>2</sup>C 总线格式

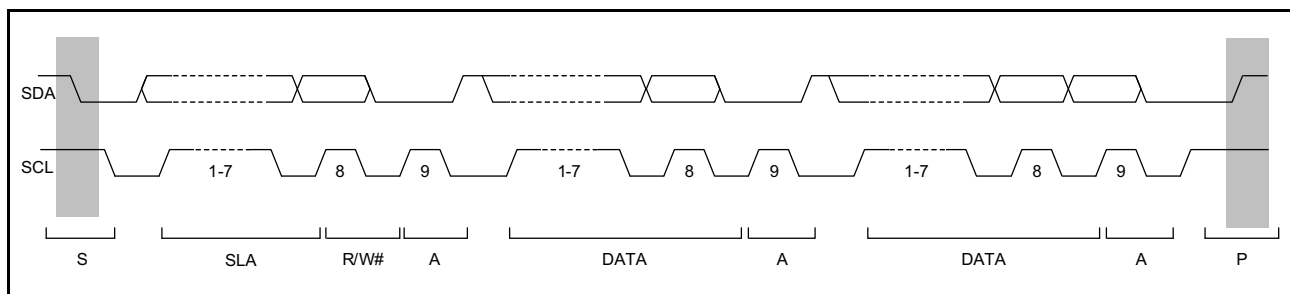


图 24.4 I<sup>2</sup>C 总线时序 (SLA 为 7 位)

#### 【符号说明】

- S : 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 : 从属地址。主控设备选择从属设备。
- R/W# : 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 : 应答。接收设备将 SDA 线置为 Low 电平（在主控发送模式中，从属设备返回应答；在接收模式中，主控设备返回应答）。
- Sr : 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 : 发送和接收的数据。
- P : 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。

### 24.3.2 初始设定

在开始发送或者接收数据时，必须按照图 24.5 所示的步骤对 RIIC 进行初始化。



图 24.5 RIIC 的初始化流程图例子

### 24.3.3 主控发送

在主控发送中，主控设备 RIIC 输出 SCL 时钟和发送数据，从属设备返回应答。主控发送的使用例子和运行时序分别如图 24.6 和图 24.7 ~ 图 24.9 所示。

主控发送的发送步骤和运行如下所示：

1. 在将 ICCR1.ICE 位置“0”（功能停止状态）的状态下，通过将 ICCR1.IICRST 位置“1”（内部复位）后又置“0”（解除复位），对 ICSR1 寄存器的各标志和内部状态进行初始化。然后，设定 SARLy、SARUy、ICSER、ICMR1、ICBRH、ICBRL 寄存器（y=0~2）并且必须根据需要设定其他寄存器（有关 RIIC 的初始设定请参照图 24.5）。一旦设定完所需的寄存器，就必须将 ICE 位置“1”（能传送状态）。在 RIIC 已初始化的情况下，不需要此步骤。
2. 读 ICCR2.BBSY 标志，在确认总线为释放状态后将 ICCR2.ST 位置“1”（请求发行开始条件）。如果 RIIC 接受开始条件的发行请求，就发行开始条件。如果 RIIC 检测到开始条件，就自动将 BBSY 标志和 ICSR2.START 标志置“1”，并且自动将 ST 位置“0”。此时，如果在 ST 位为“1”的状态下 RIIC 自身发送的 SDA 信号与 SDA 线的信号状态相同，并且检测到开始条件，RIIC 就视为通过 ST 位正确地发行了开始条件，在将 ICCR2.MST 位和 ICCR2.TRS 位自动置“1”后变为主控发送模式。另外，ICSR2.TDRE 因 TRS 位为“1”而自动变为“1”。
3. 必须在确认 ICSR2.TDRE 标志为“1”后将发送数据（从属地址和 R/W# 位）写到 ICDRT 寄存器。一旦将发送数据写到 ICDRT 寄存器，TDRE 标志就自动变为“0”，在将数据从 ICDRT 寄存器传送到 ICDRS 寄存器后，TDRE 标志再次变为“1”。一旦结束包含 R/W# 位的从属地址的发送，就根据被发送的 R/W# 位自动更改 TRS 位，并且选择发送模式或者接收模式。如果接收到为“0”的 R/W# 位，RIIC 就继续保持主控发送模式的状态。

此时，如果 ICSR2.NACKF 标志为“1”，因为没有识别到从属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 标志为“1”后将发送数据写到 ICDRT 寄存器。在准备好发送数据之前或者发行停止条件前，RIIC 自动将 SCL 线保持 Low 电平。
5. 在将要发送的全部字节写到 ICDRT 寄存器后，必须在等待 ICSR2.TEND 标志变为“1”后再给 ICCR2.SP 位写“1”（请求发行停止条件）。如果 RIIC 接受停止条件的发行请求，就发行停止条件。
6. 如果 RIIC 检测到停止条件，ICCR2.MST 位和 ICCR2.TRS 位就自动变为“00b”并且转移到从属接收模式，而且 ICSR2.TDRE 标志和 ICSR2.TEND 标志因检测到停止条件而自动变为“0”，ICSR2.STOP 标志变为“1”。
7. 在确认 ICSR2.STOP 标志为“1”后，必须将 ICSR2.NACKF 标志和 ICSR2.STOP 标志置“0”，以便进行下一次的通信。

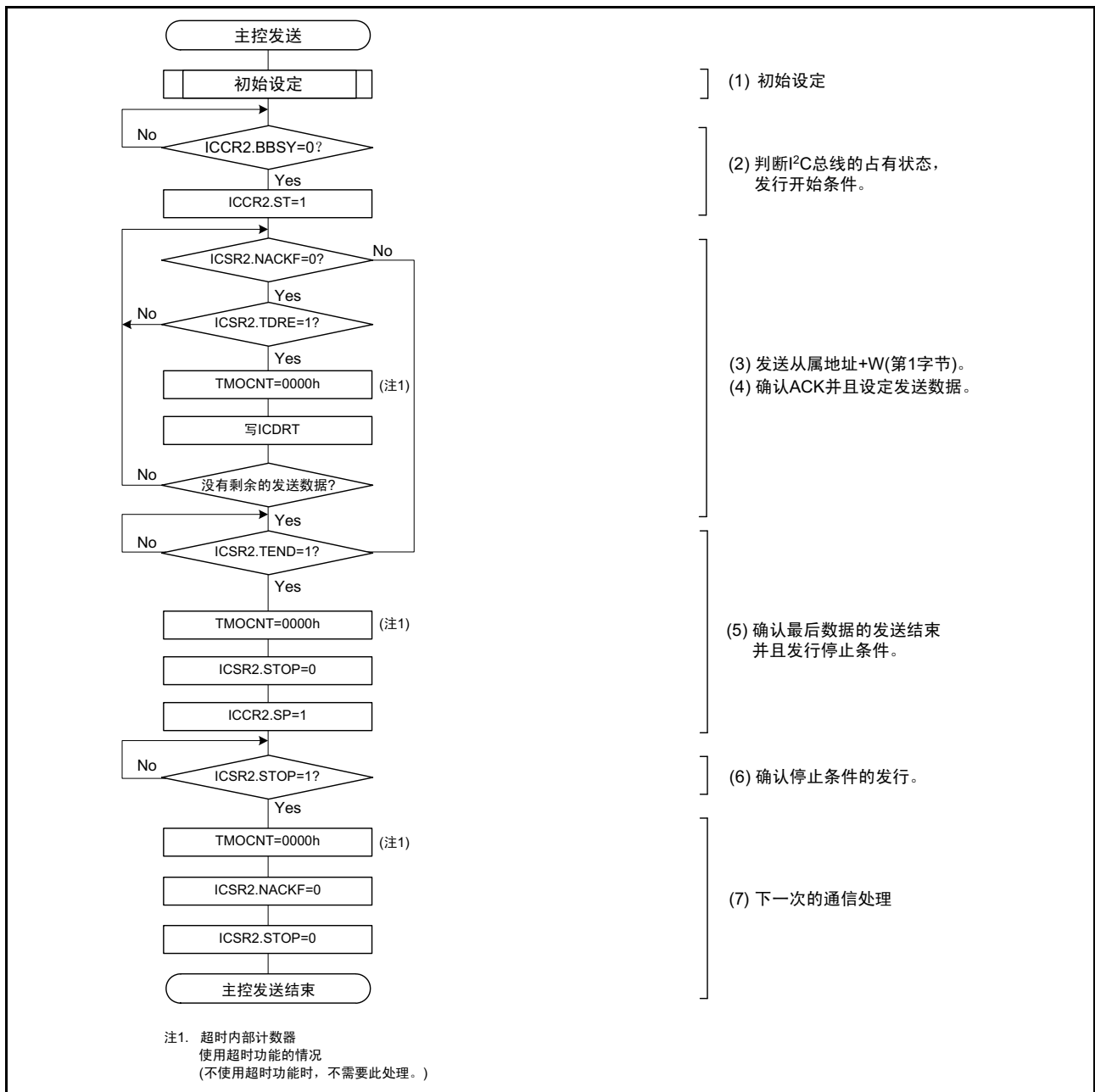


图 24.6 主控发送的流程图例子

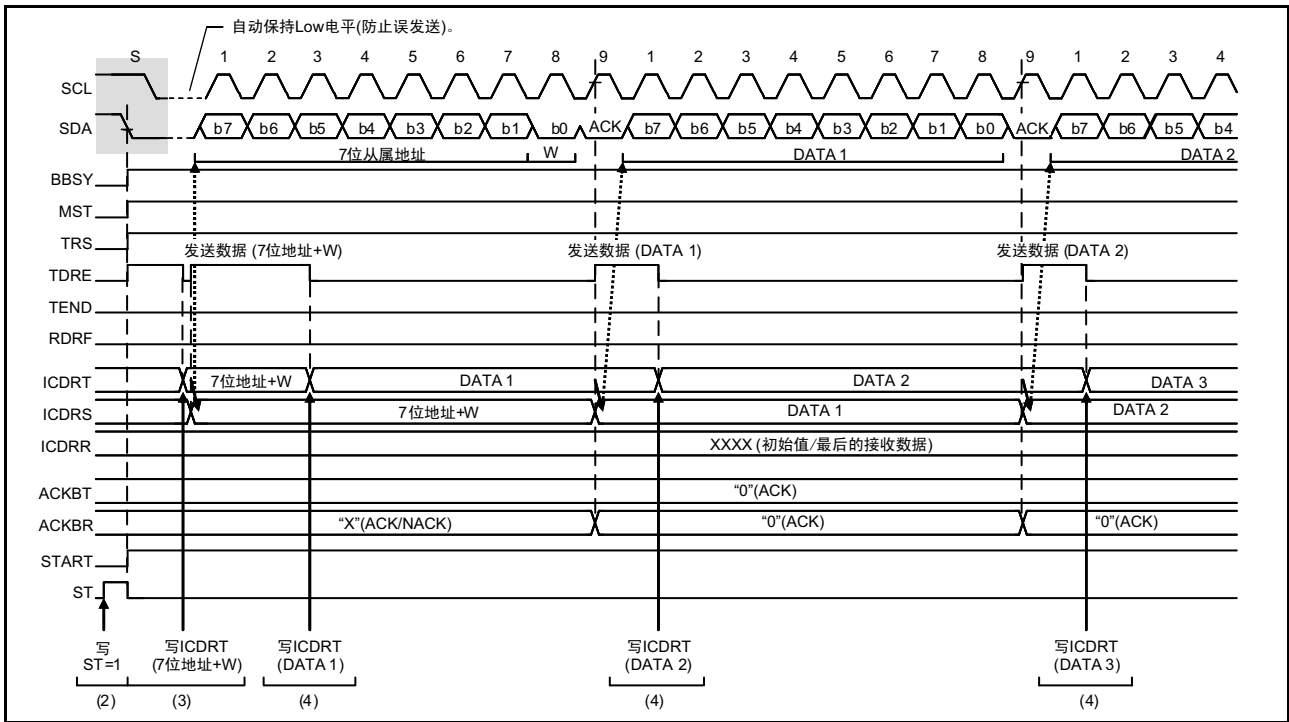


图 24.7 主控发送的运行时序 (1) (7 位地址格式的情况)

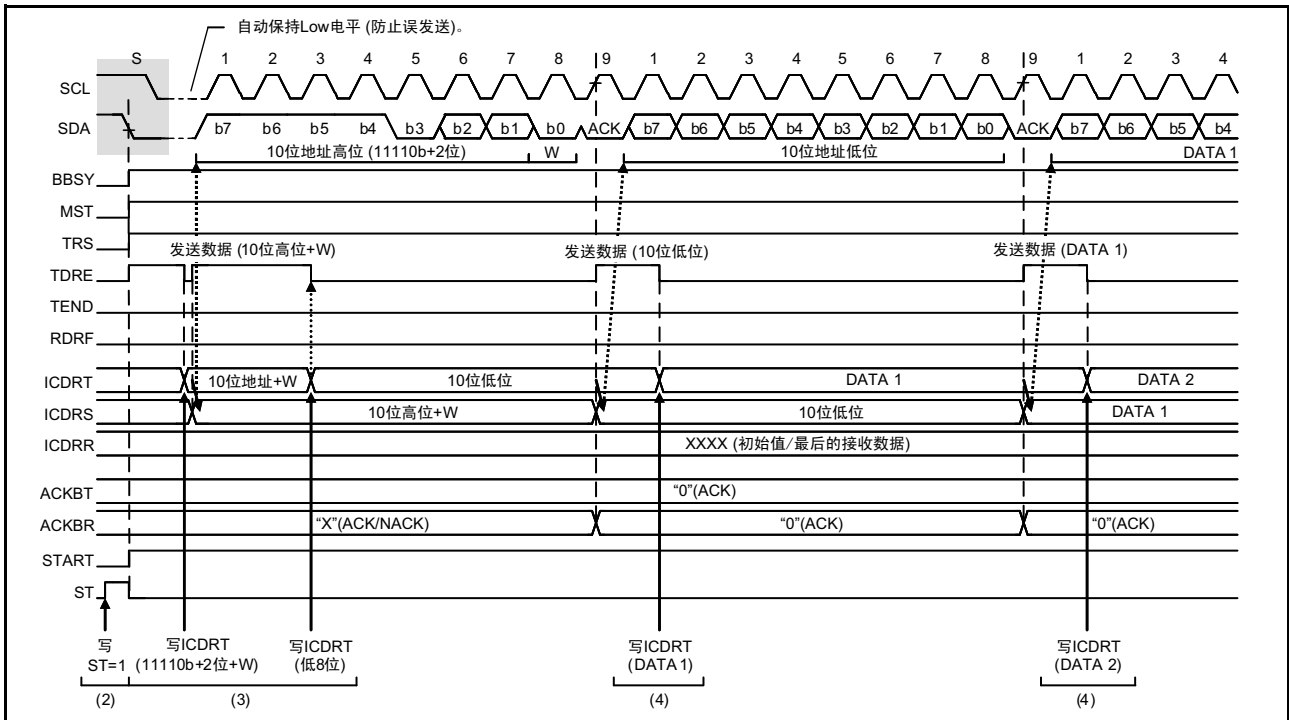


图 24.8 主控发送的运行时序 (2) (10 位地址格式的格式)

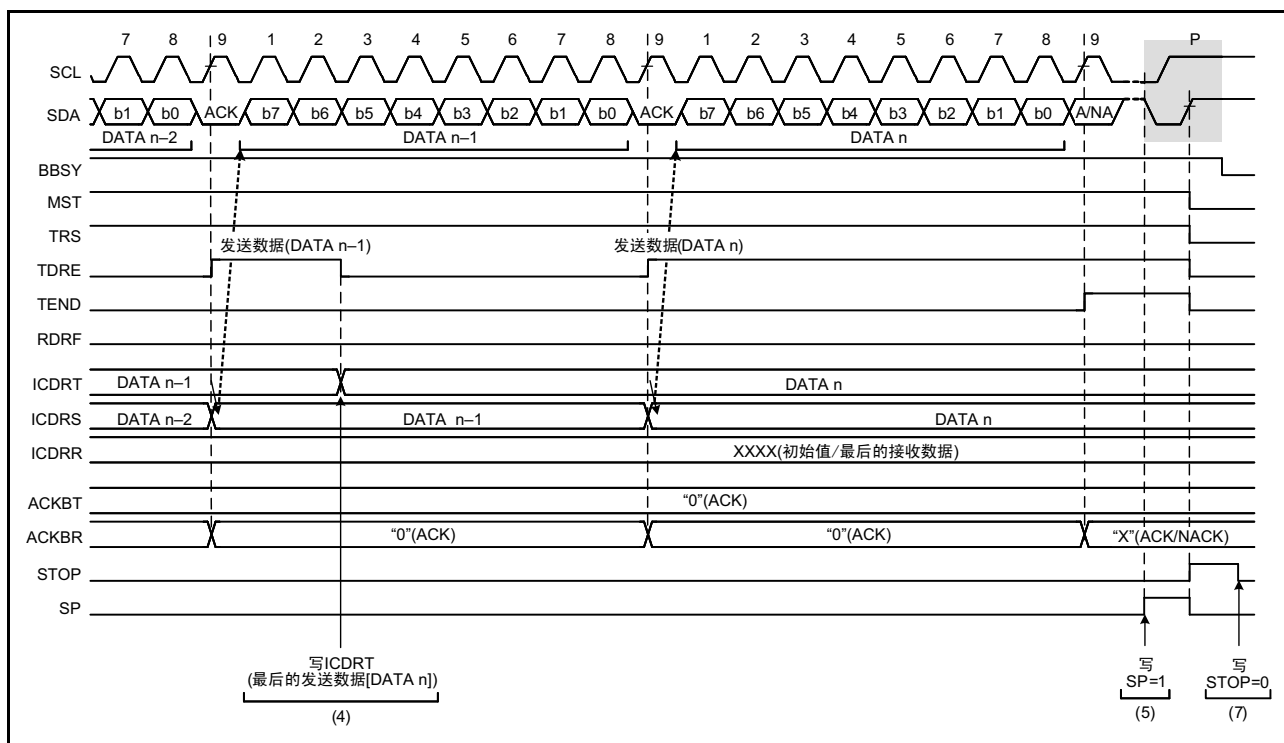


图 24.9 主控发送的运行时序 (3)

### 24.3.4 主控接收

在**主控接收**中，**主控设备 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。

主控接收的使用例子（7 位地址格式）和运行时序分别如图 24.10 和图 24.11 ~ 图 24.13 所示。

主控接收的接收步骤和运行如下所示：

1. 在将 **ICCR1.ICE** 位置“0”（功能停止状态）的状态下，通过将 **ICCR1.IICRST** 位置“1”（内部复位）后又置“0”（解除复位），对 **ICSR1** 寄存器的各标志和内部状态进行初始化。然后，设定 **SARLy**、**SARUy**、**ICSER**、**ICMR1**、**ICBRH**、**ICBRL** 寄存器（ $y=0\sim 2$ ）并且必须根据需要设定其他的寄存器（有关 **RIIC** 的初始设定请参照图 24.5）。必须在设定完所需的寄存器后将 **ICE** 位置“1”（能传送状态）。在 **RIIC** 已初始化的情况下，不需要此步骤。
2. 读 **ICCR2.BBSY** 标志，在确认总线为释放状态后将 **ICCR2.ST** 位置“1”（请求发行开始条件）。如果 **RIIC** 接受开始条件的发行请求，就发行开始条件。如果 **RIIC** 检测到开始条件，就自动将 **BBSY** 标志和 **ICSR2.START** 标志置“1”，并且自动将 **ST** 位置“0”。此时，如果在 **ST** 位为“1”的状态下 **RIIC** 自身发送的 **SDA** 信号与 **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 **TRS** 位为“1”而自动变为“1”。
3. 必须在确认 **ICSR2.TDRE** 标志为“1”后将发送数据（从属地址和 **R/W#** 位）写到 **ICDRT** 寄存器。一旦将发送数据写到 **ICDRT** 寄存器，**TDRE** 标志就自动变为“0”，在将数据从 **ICDRT** 寄存器传送到 **ICDRS** 寄存器后，**TDRE** 标志再次变为“1”。一旦结束包含 **R/W#** 位的从属地址的发送，就根据被发送的 **R/W#** 位自动更改 **ICCR2.TRS** 位，并且选择发送模式或者接收模式。如果 **RIIC** 接收到为“1”的 **R/W#** 位，就在第 9 个时钟的上升沿将 **TRS** 位置“0”后转移到**主控接收模式**，此时 **TDRE** 标志变为“0”，**ICSR2.RDRF** 标志自动变为“1”。  
此时，如果 **ICSR2.NACKF** 标志为“1”，因为没有识别到从属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** 标志为“1”后虚读 **ICDRR** 寄存器，**RIIC** 就在输出 **SCL** 时钟后开始接收。
5. 结束 1 字节数据的接收后，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，在读 **ICDRR** 寄存器（最后字节 - 第 2 个字节）前将 **ICMR3.WAIT** 位置“1”（有 **WAIT**）。由此，即使在接下来 6. 中的将 **ICMR3.ACKBT** 位置“1”（**NACK**）的处理因其他中断而发生延迟，也能在最后字节进行 **NACK** 输出，同时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，并且处于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（最后字节 - 第 1 个字节）后，必须先确认 **ICSR2.RDRF** 标志为“1”，然后给 **ICCR2.SP** 位写“1”（请求发行停止条件）并且读 **ICDRR** 寄存器（最后字节）。**RIIC** 通过读 **ICDRR** 寄存器来解除 **WAIT** 状态，在结束第 9 个时钟的 **Low** 电平输出或者解除 **SCL** 线保持的 **Low** 电平后发行停止条件。
8. 如果 **RIIC** 检测到停止条件，**ICCR2.MST** 位和 **ICCR2.TRS** 位就自动变为“00b”并且转移到从属接收模式，而且 **ICSR2.STOP** 标志因检测到停止条件而变为“1”。
9. 在确认 **ICSR2.STOP** 标志为“1”后，必须将 **ICSR2.NACKF** 和 **ICSR2.STOP** 标志置“0”，以便进行下一次的通信。



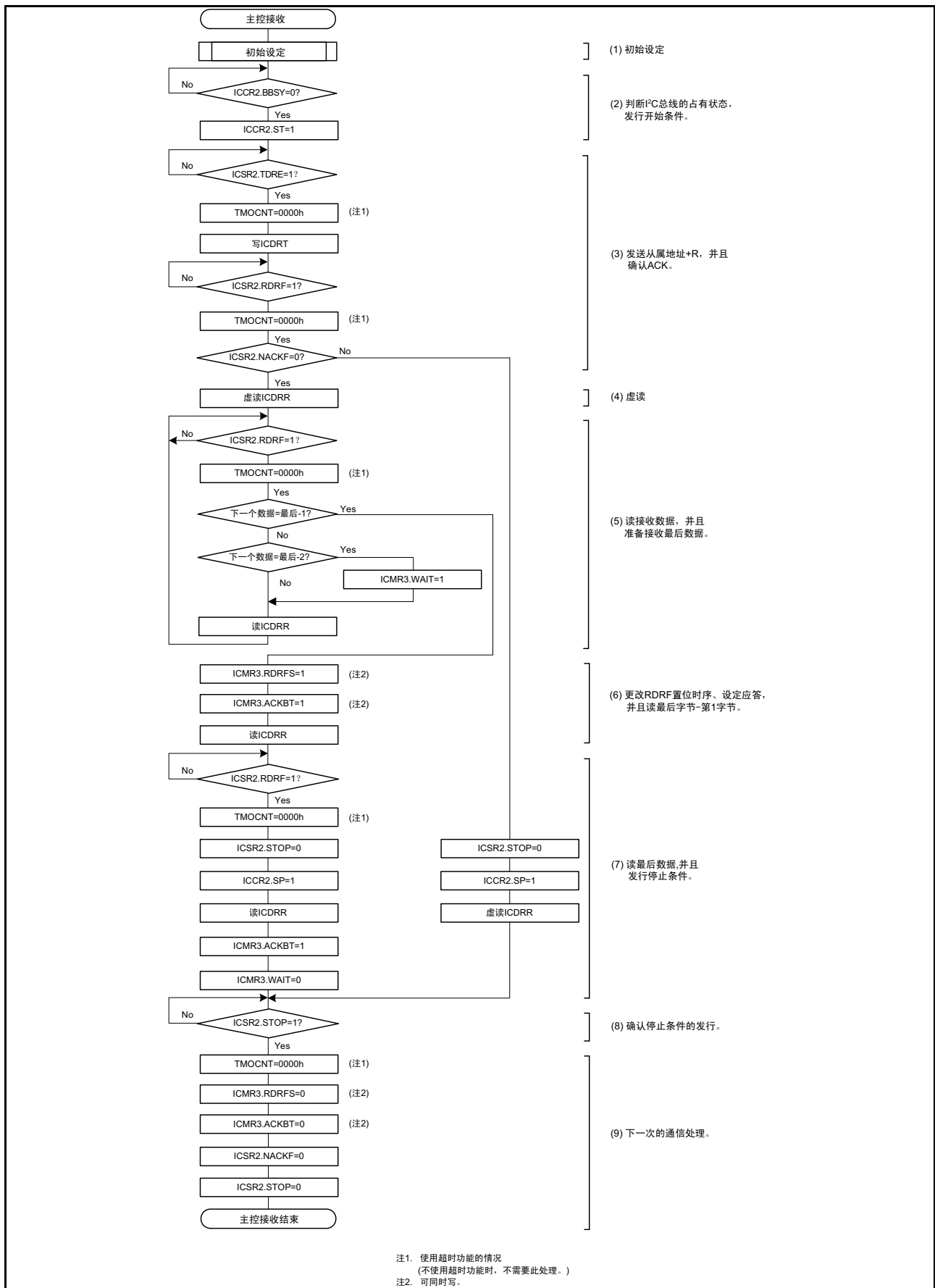


图 24.10 主控接收的流程图例子 (7 位地址格式的情况)

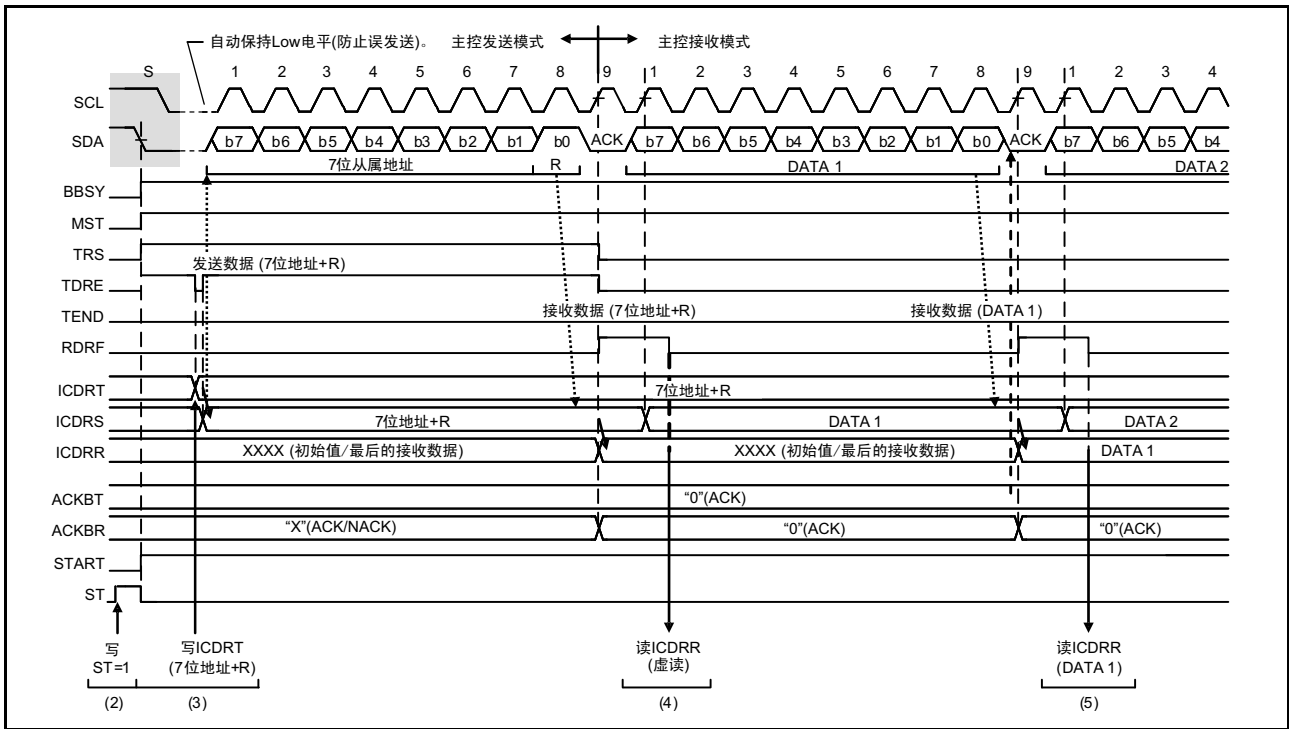


图 24.11 主控接收的运行时序 (1) (7 位地址格式, RDRFS=0 的情况)

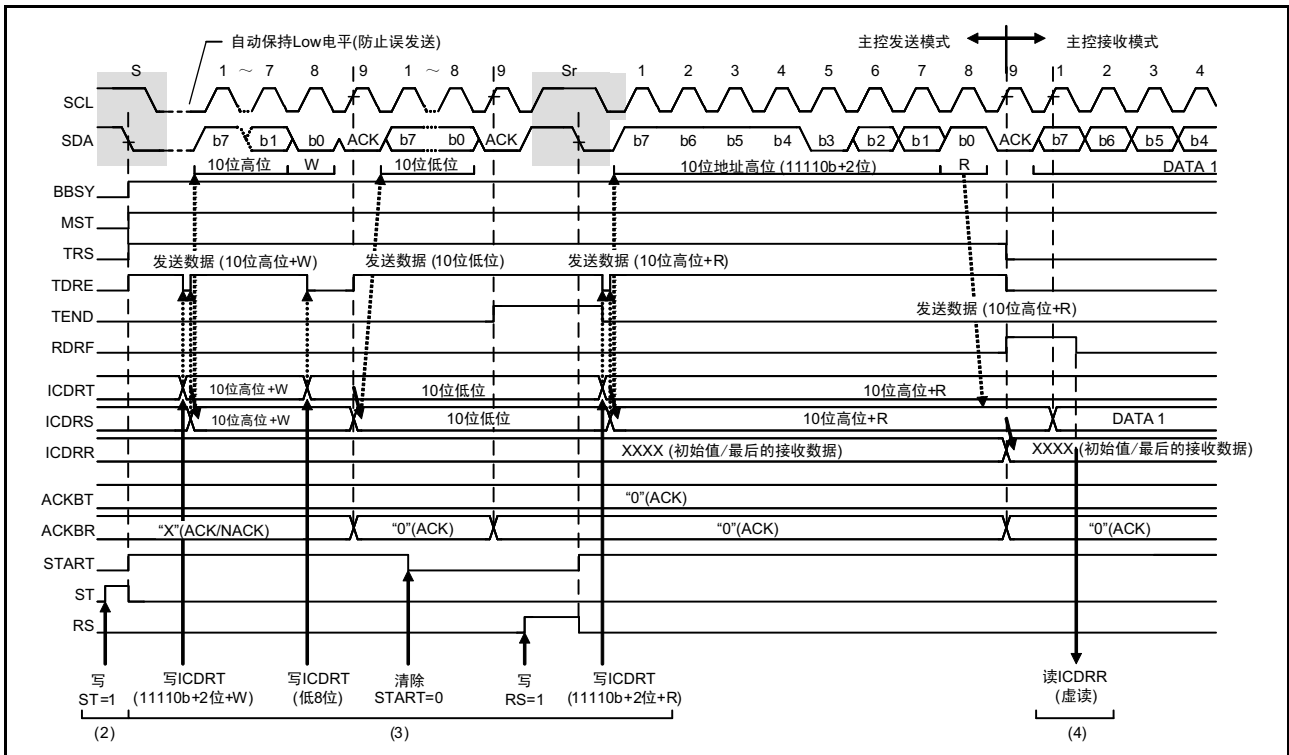


图 24.12 主控接收的运行时序 (2) (10 位地址格式, RDRFS=0 的情况)

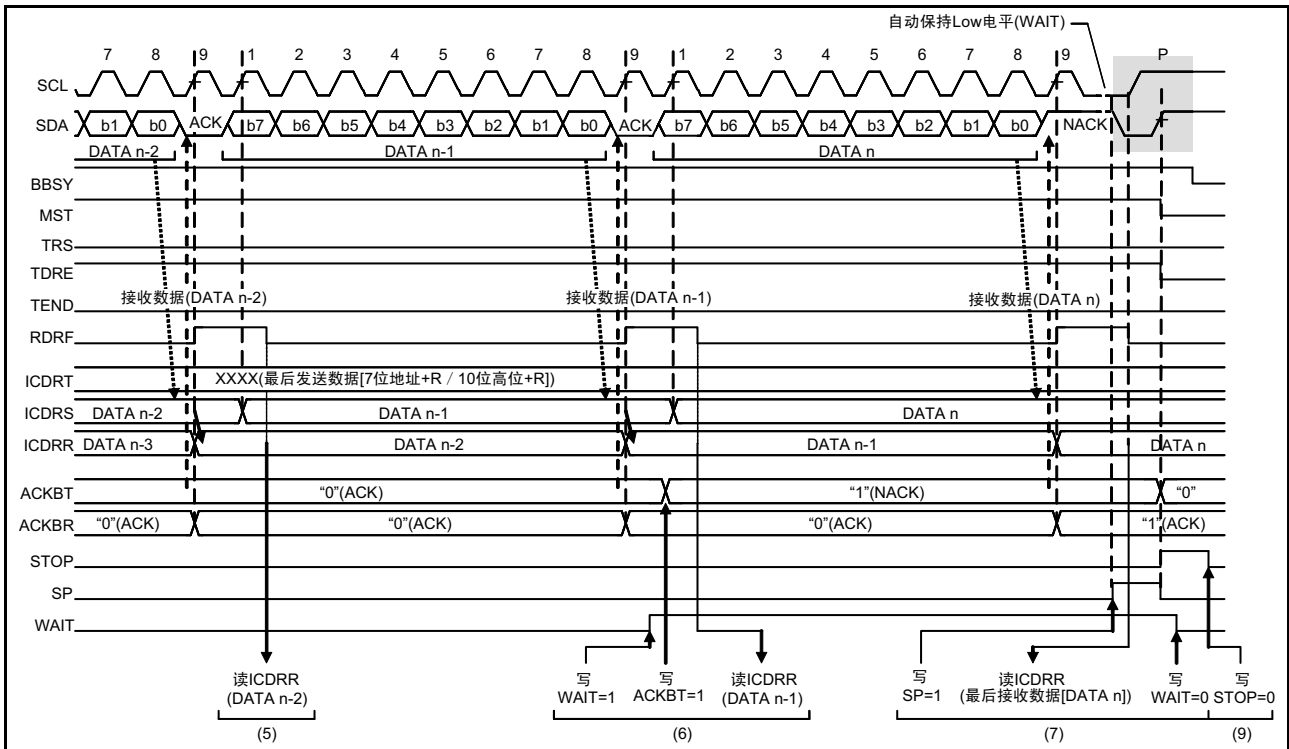


图 24.13 主控接收的运行时序 (3) (RDRFS=0 的情况)

### 24.3.5 从属发送

在从属发送中，主控设备输出 SCL 时钟，从属设备 RIIC 发送数据，并且主控设备返回应答。

从属发送的使用例子和运行时序分别如图 24.14 和图 24.15、图 24.16 所示。

从属发送的发送步骤及其运行如下所示：

1. 必须按照图 24.5 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、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“1”，就将 ICCR2.TRS 位和 ICSR2.TDRE 标志置“1”并且自动转换为从属发送模式。
3. 必须在确认 ICSR2.TDRE 标志为“1”后将发送数据写到 ICDRT 寄存器。此时，如果在 ICFER.NACK 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 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 标志为“1”后，必须将 ICSR2.NACKF 标志和 ICSR2.STOP 标志置“0”，以便进行下一次的通信。

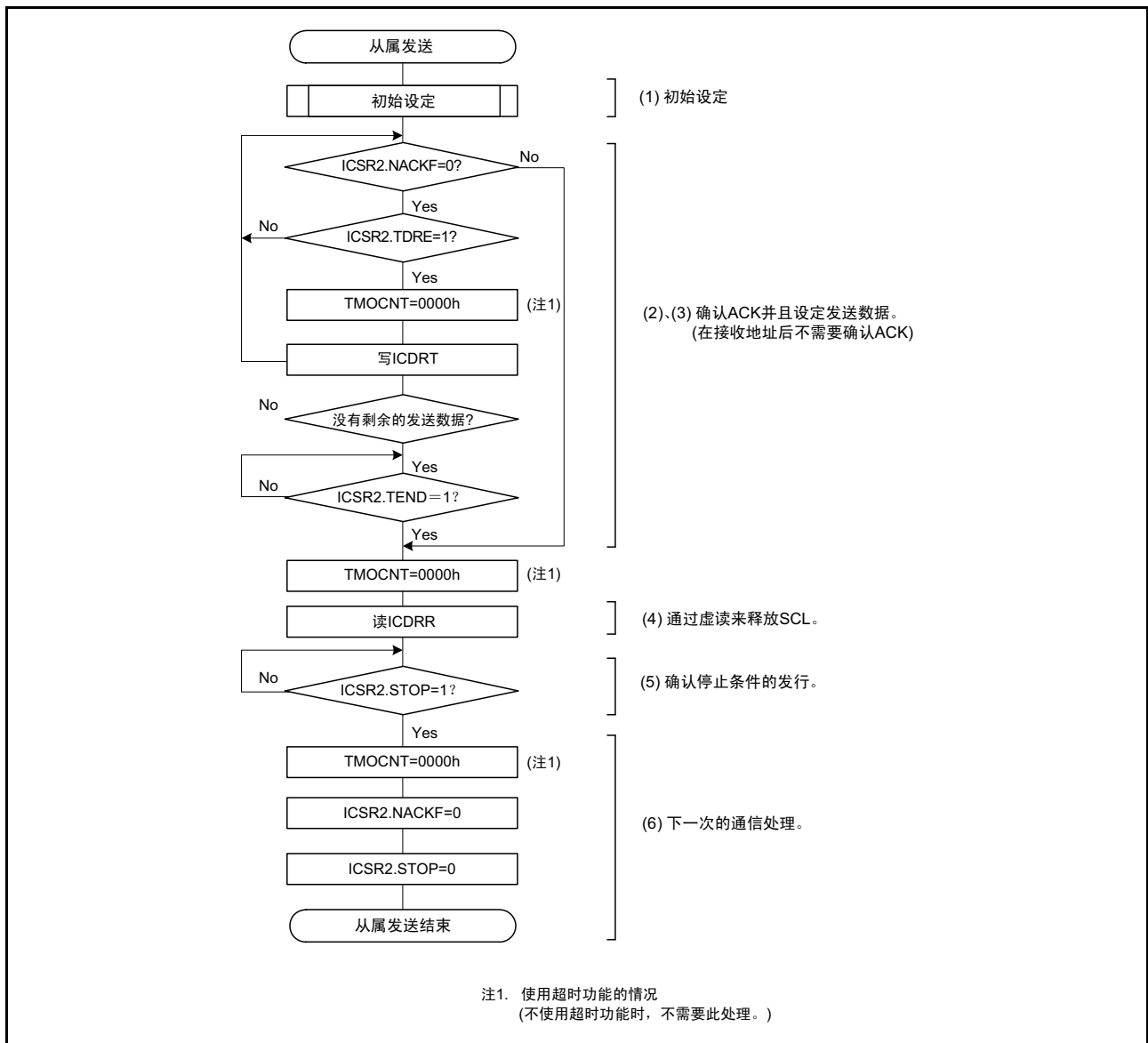


图 24.14 从属发送的流程图例子

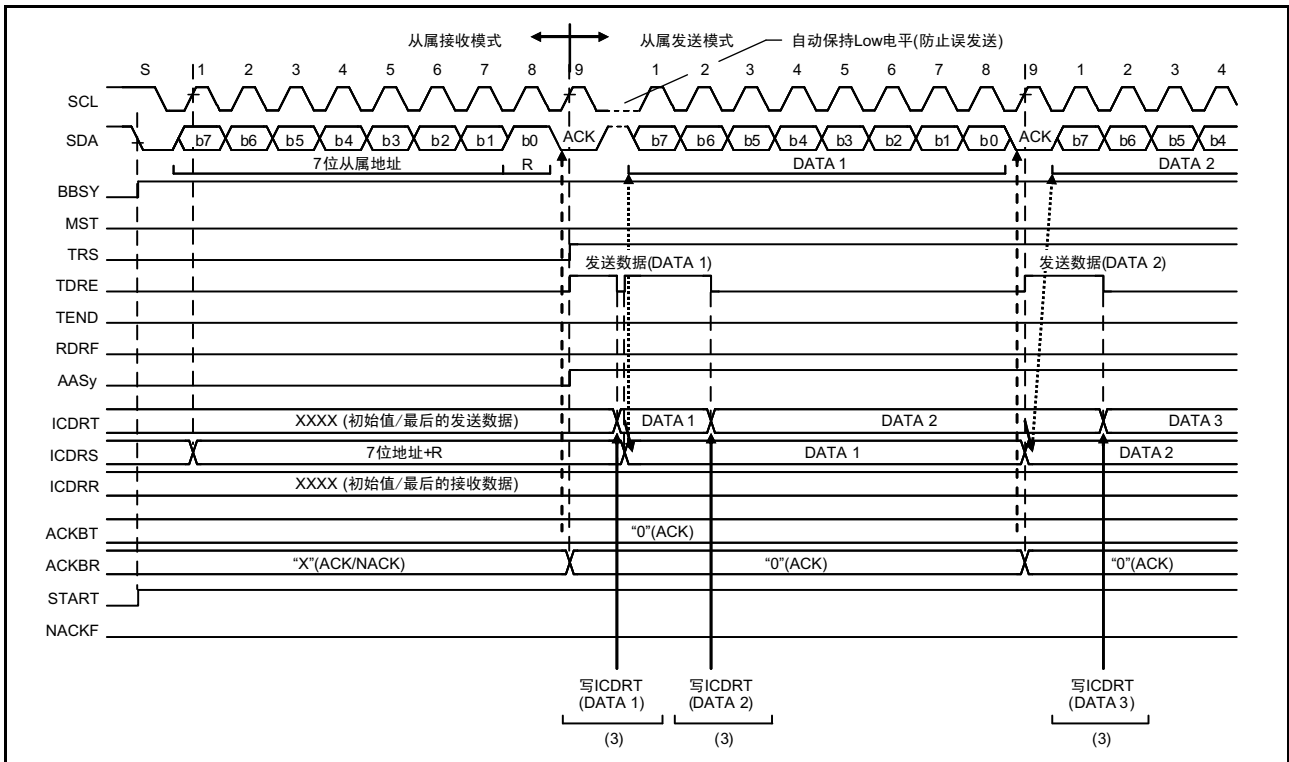


图 24.15 从属发送的运行时序 (1) (7 位地址格式的情况)

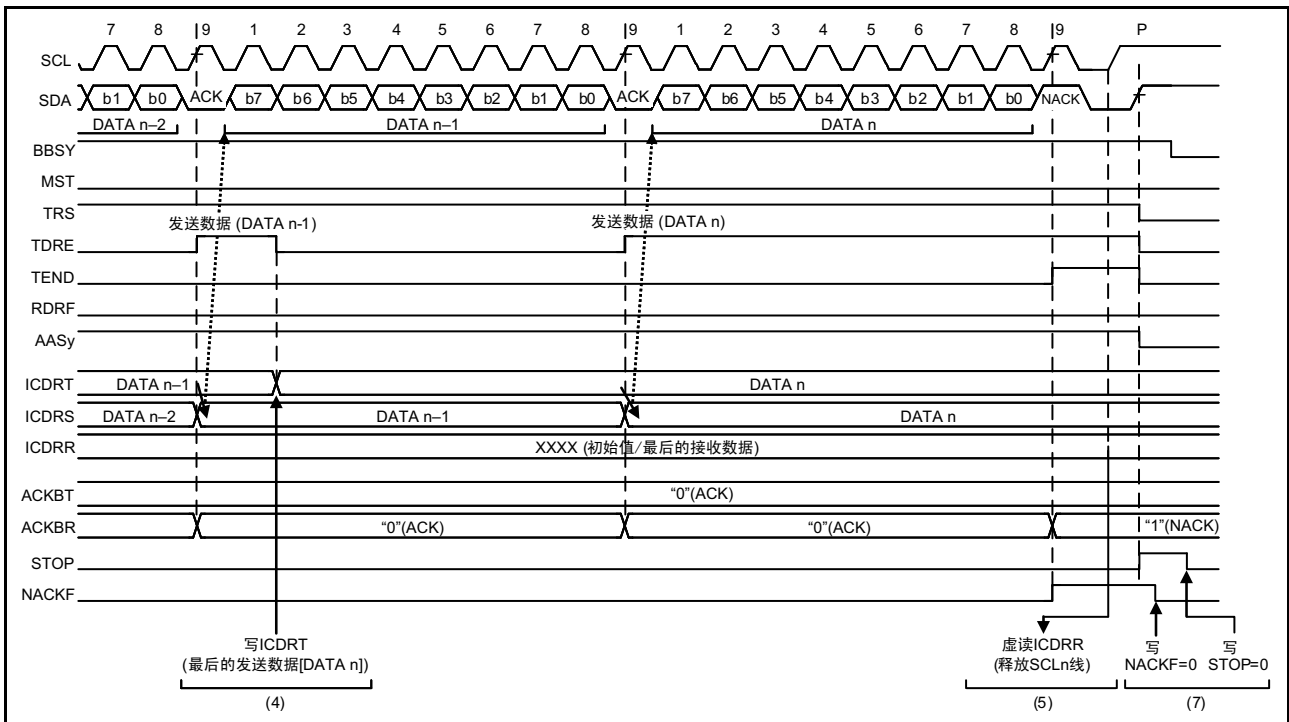


图 24.16 从属发送的运行时序 (2)

### 24.3.6 从属接收

在从属接收中，主控设备输出 SCL 时钟和发送数据，从属设备 RIIC 返回应答。

从属接收的使用例子和运行时序分别如图 24.17 和图 24.18、图 24.19 所示。

从属接收的接收步骤和运行如下所示：

1. 必须按照图 24.5 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 标志为“0”并且 ICSR2.RDRF 标志为“1”后，第 1 次必须虚读 ICDRR 寄存器（虚读的接收数据在 7 位地址格式时为从属地址+R/W# 位，在 10 位地址格式时为低 8 位地址）。
4. 如果读 ICDRR 寄存器，RIIC 就自动将 ICSR2.RDRF 标志置“0”。如果在延迟了 ICDRR 寄存器的读操作并且 RDRF 标志为“1”的状态下接收下一个数据，RIIC 就在 RDRF 标志变为“1”时的前一个 SCL 时钟下降沿将 SCL 线保持 Low 电平。通过读 ICDRR 寄存器来解除被保持的 Low 电平，RIIC 释放 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 标志为“1”后，必须将 ICSR2.STOP 标志置“0”，以便进行下一次的通信。

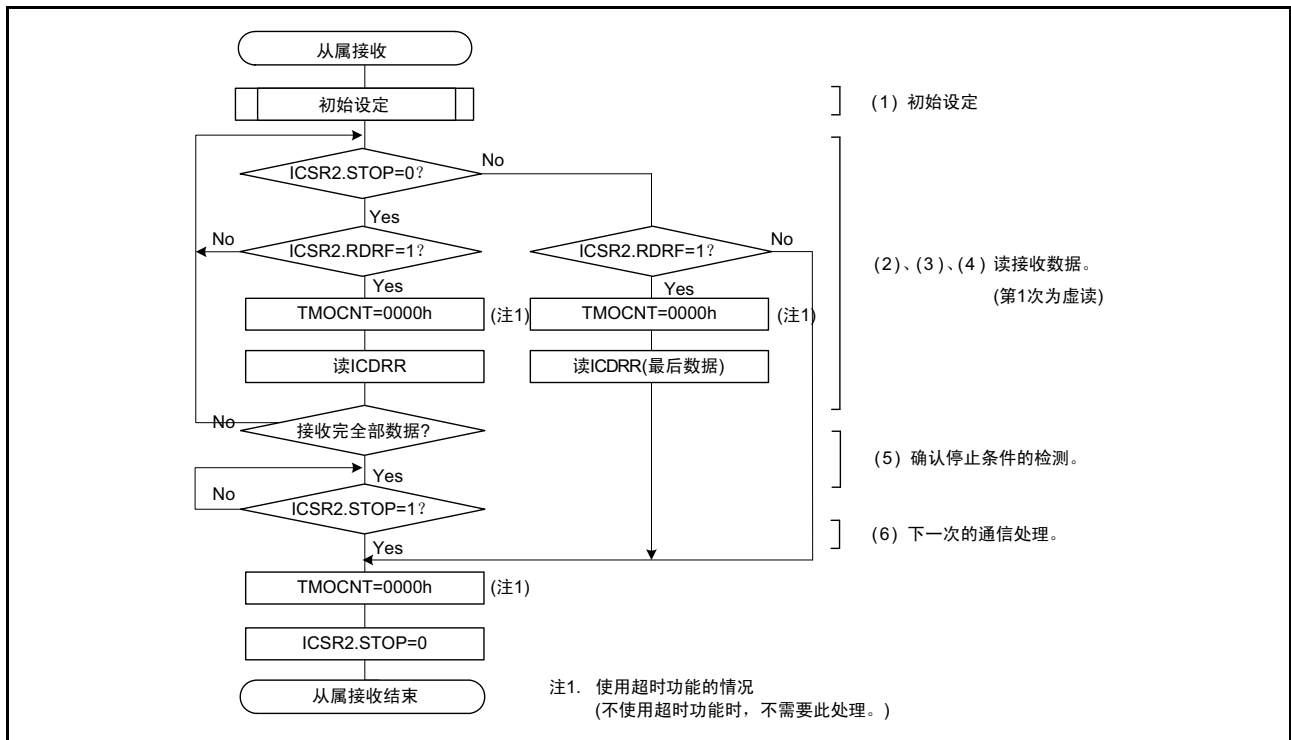


图 24.17 从属接收的流程图例子

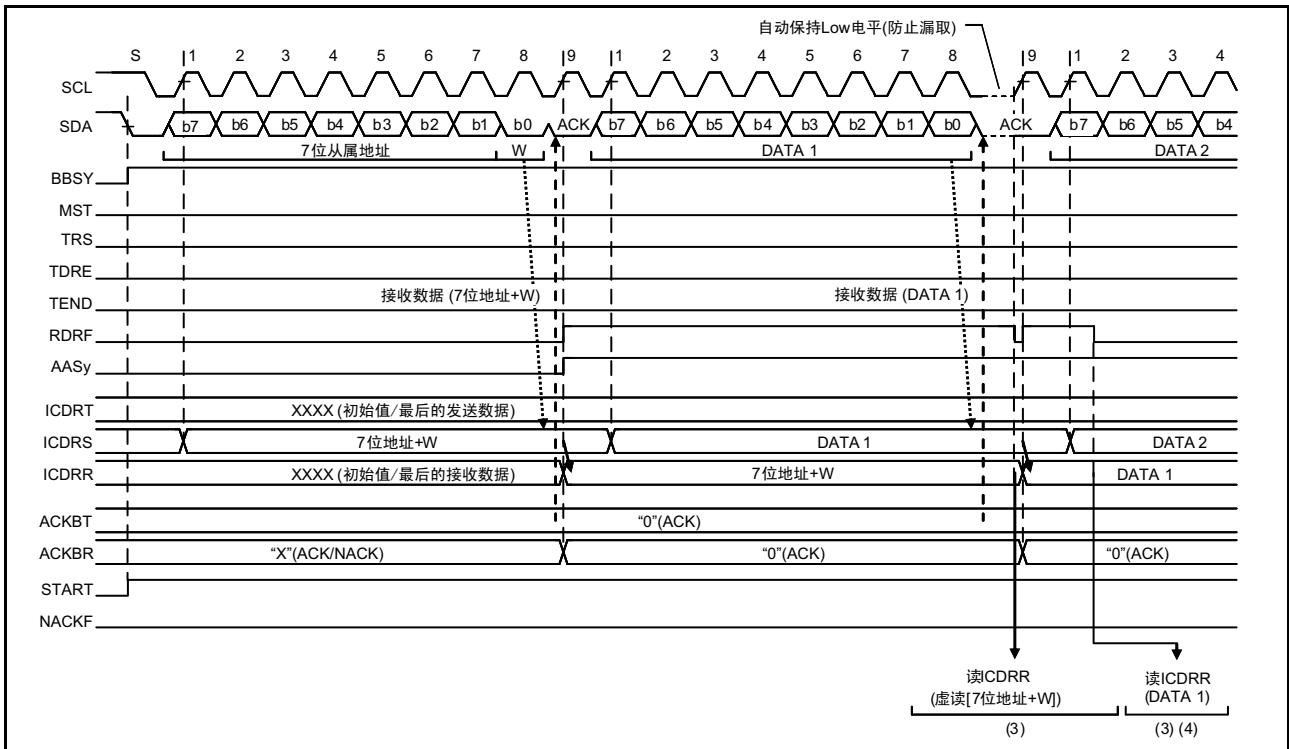


图 24.18 从属接收的运行时序 (1) (7 位地址格式, RDRFS=0 的情况)

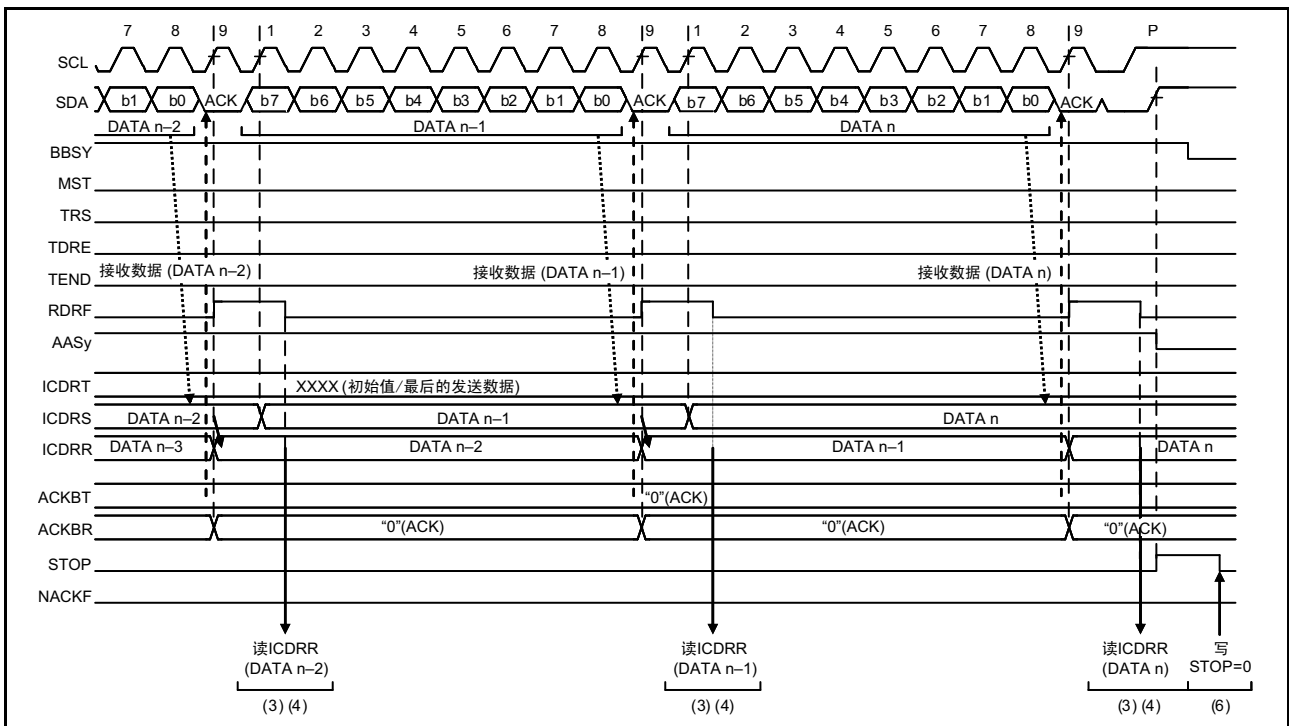


图 24.19 从属接收的运行时序 (2) (RDRFS=0 的情况)

### 24.4 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 电平并且下降。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 I<sup>2</sup>C 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, 在主控模式中监视 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。

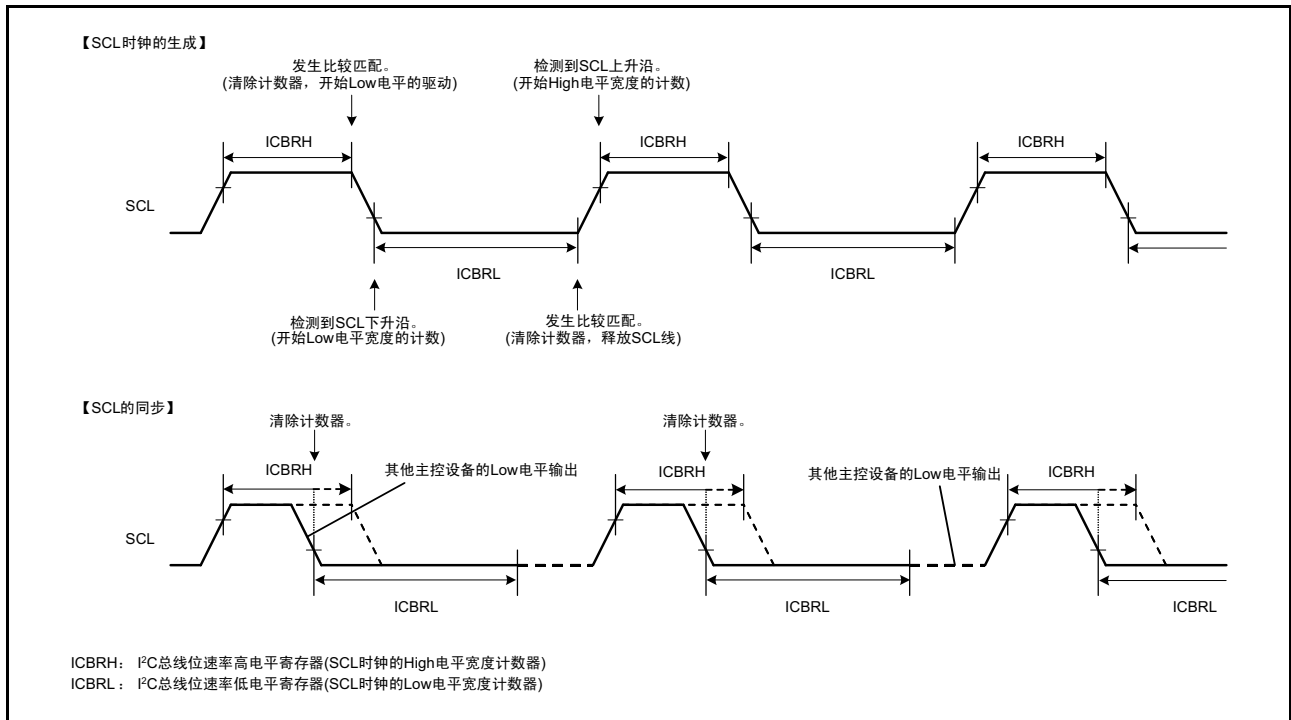


图 24.20 RIIC 的 SCL 时钟生成和 SCL 同步



### 24.5 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 $\phi$ ）或者内部基准时钟的 2 分频时钟（IIC $\phi$ /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）。

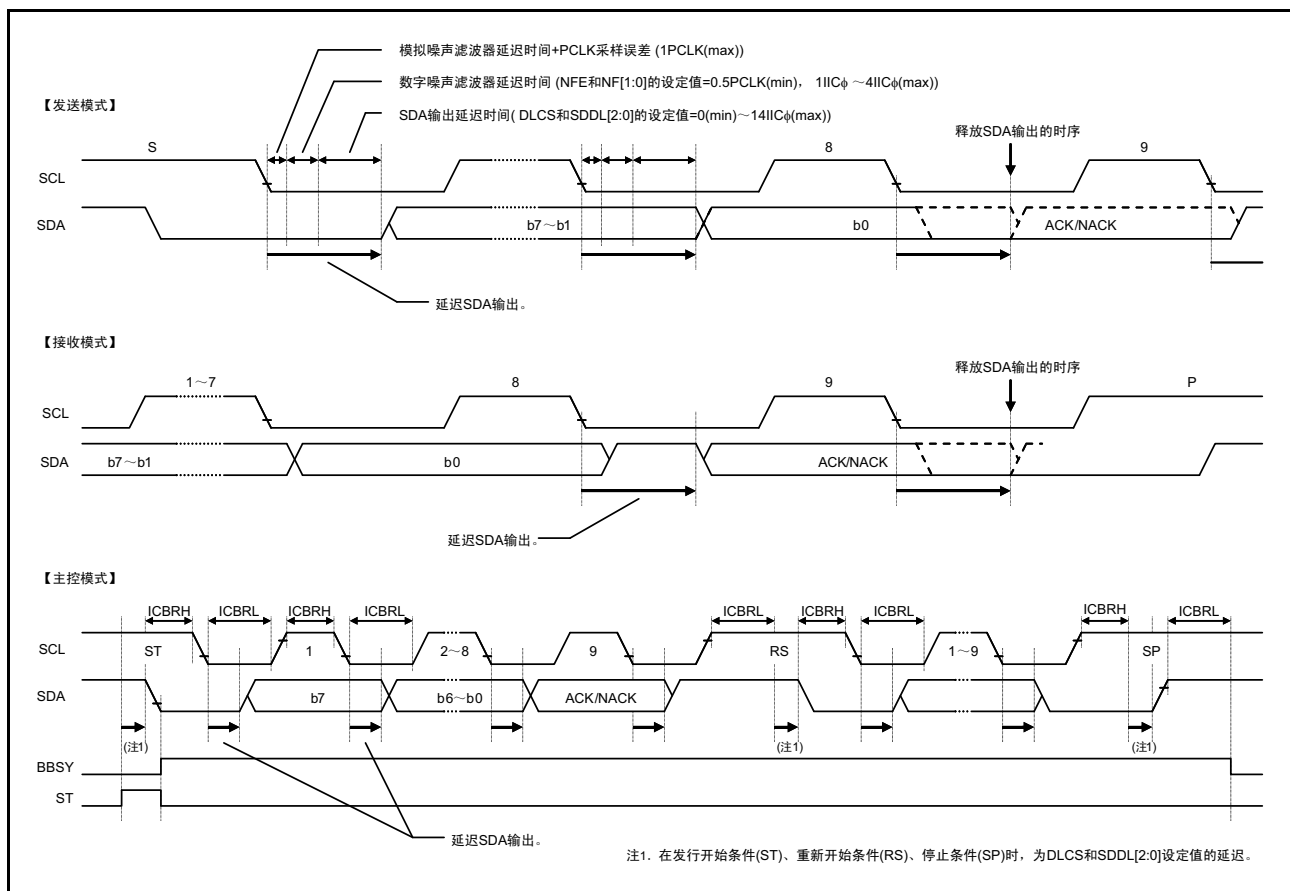


图 24.21 SDA 输出延迟时序

## 24.6 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 24.22 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 $\phi$  周期。

在 PCLK 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。

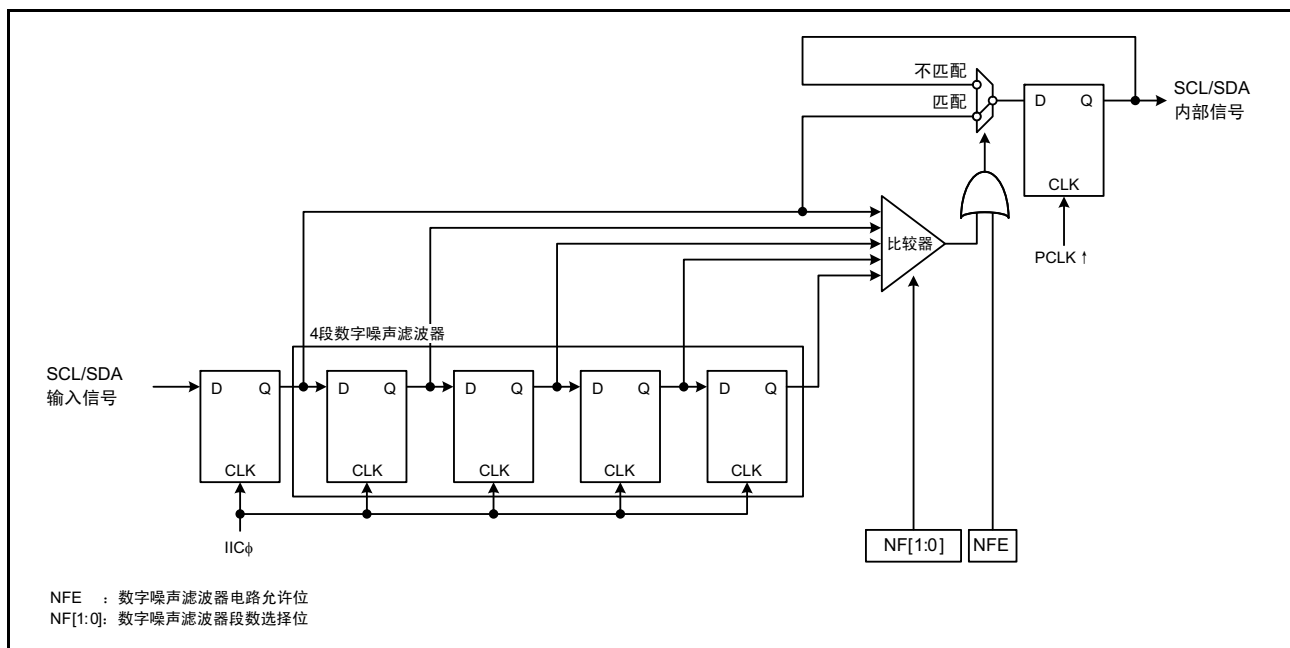


图 24.22 数字噪声滤波器电路的框图

## 24.7 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。

### 24.7.1 从属地址匹配检测功能

RIIC 能设定 3 种从属地址，有分别对应的从属地址检测功能。当 ICSER.SARyE 位 (y=0 ~ 2) 为“1”时，能检测到 SARUy/SARLy 寄存器 (y=0 ~ 2) 设定的从属地址。

如果 RIIC 检测到设定的从属地址匹配，就在 SCL 时钟的第 9 个时钟的下降沿将对应的 ICSR1.AASy 标志 (y=0 ~ 2) 置“1”，然后根据后续的 R/W# 位将 ICSR2.RDRF 标志或者 ICSR2.TDRE 标志置“1”。因此，能发生接收数据满中断 (ICRXI) 或者发送数据空中断 (ICTXI)，并且能通过 AASy 标志判断指定了哪个从属地址。

AASy (y=0 ~ 2) 标志变为“1”的时序如图 24.23 ~ 图 24.25 所示。

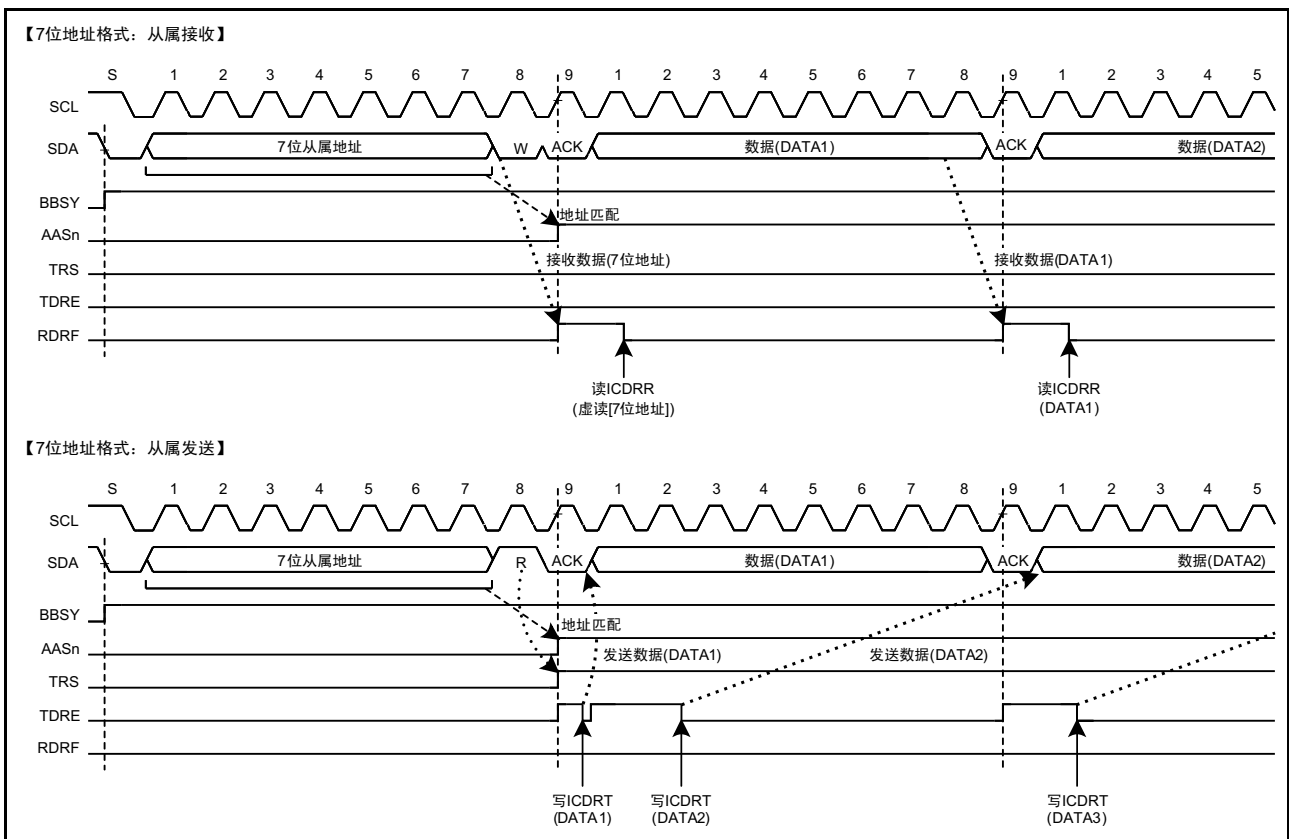


图 24.23 选择 7 位地址格式时的 AASy 标志变为“1”的时序

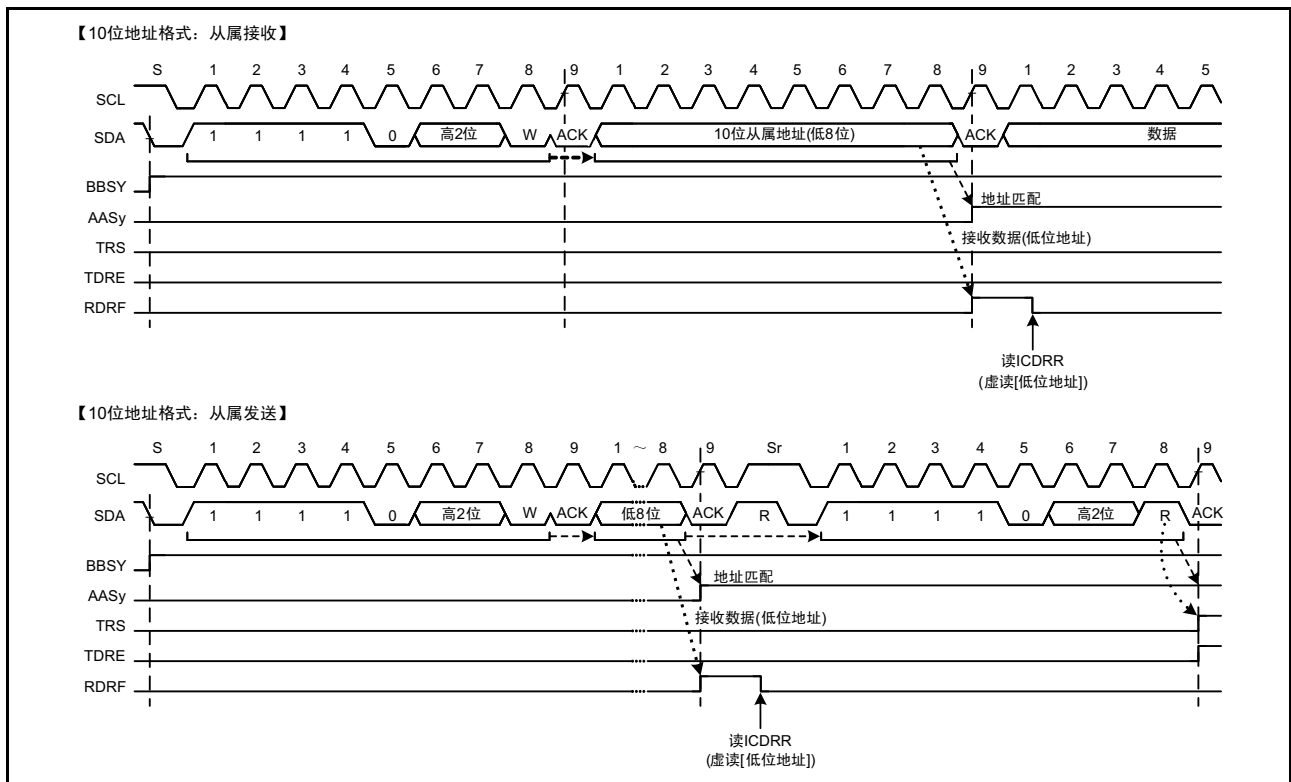


图 24.24 选择 10 位地址格式时的 AASy 标志变为“1”的时序

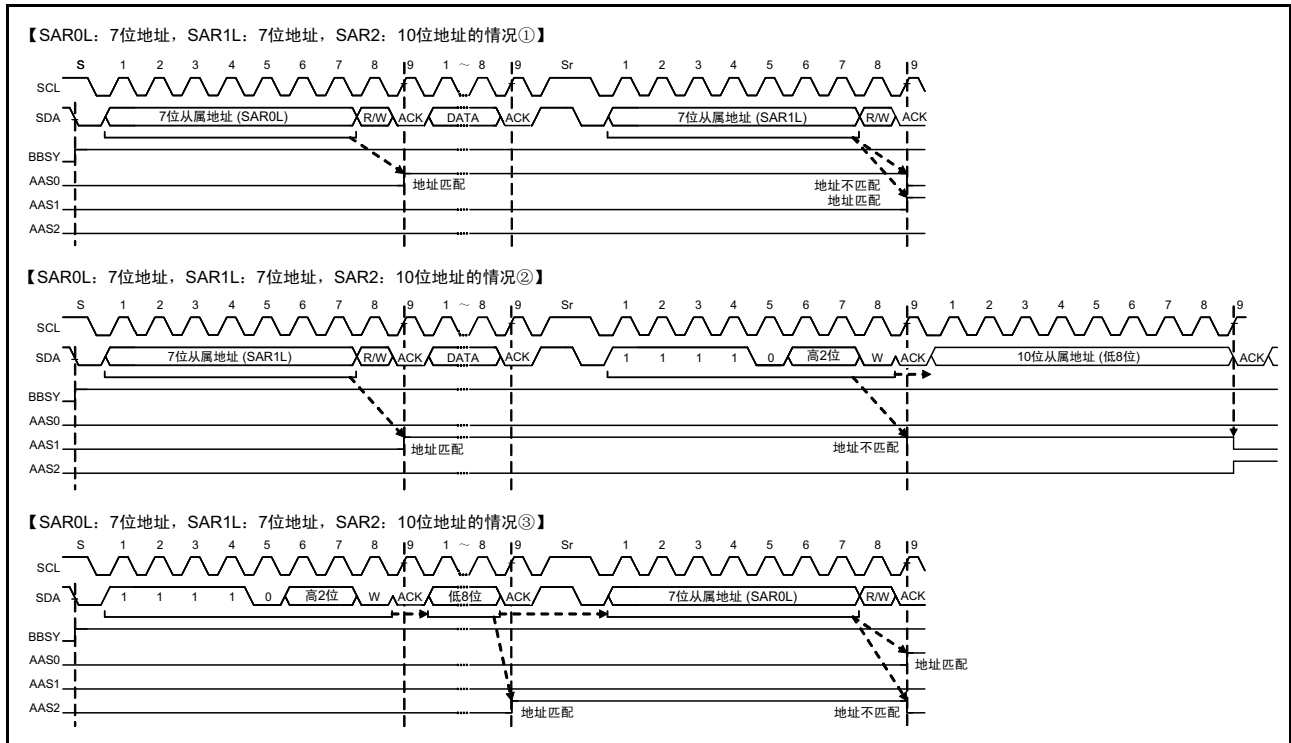


图 24.25 7 位 /10 位地址格式同时存在时的 AASy 标志变为“1”/“0”的时序

### 24.7.2 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, 能发生接收数据满中断 (ICRXI), 并且能通过 GCA 标志判断是否发送了全呼地址。

全呼地址检测后的运行和普通的从属接收运行相同。

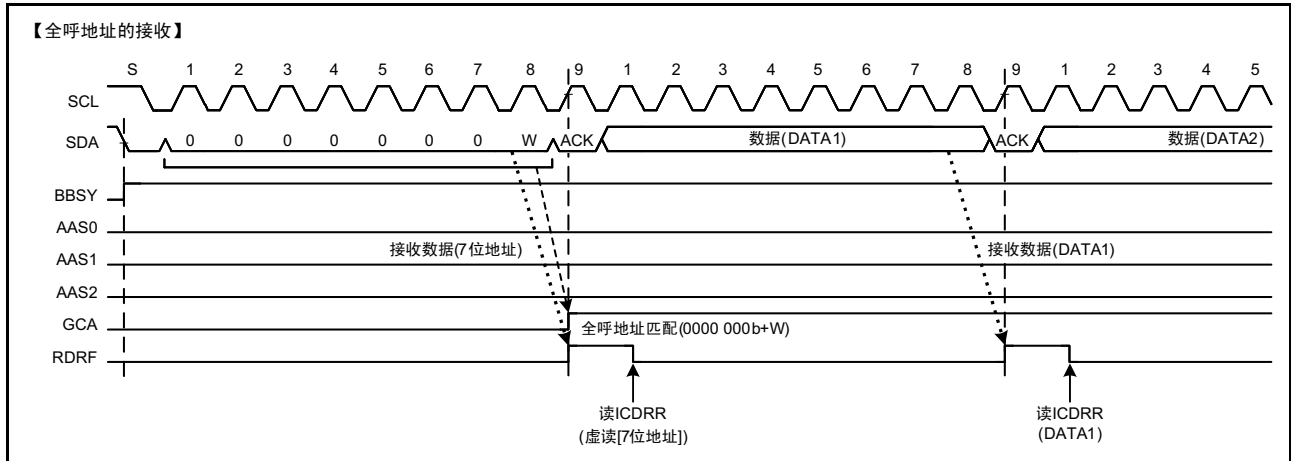


图 24.26 接收全呼地址时的 GCA 标志变为 “1” 的时序

### 24.7.3 设备 ID 地址检测功能

RIIC 具有以 I<sup>2</sup>C 总线 (Rev.03) 为基准的设备 ID 地址检测功能。在将 ICSER.DIDE 位置 “1” 的状态下, 如果开始条件或者重新开始条件之后的第 1 个字节接收到 “1111 100b”, RIIC 就将此地址视为设备 ID 地址, 当后续的 R/W# 位为 “0” 时, 就在 SCL 时钟的第 9 个时钟的上升沿将 ICSR1.DID 标志置 “1”, 然后将第 2 个字节以后的地址和自己的从属地址进行比较。如果第 2 个字节以后的地址与从属地址寄存器的值匹配, 对应的 ICSR1.AASy 标志 (y=0 ~ 2) 就变为 “1”。

此后, 如果开始条件或者重新开始条件之后的第 1 个字节再次与设备 ID 地址 (1111 100b) 匹配并且后续的 R/W# 位为 “1”, RIIC 就不比较第 2 个字节以后的地址, 而将 ICSR2.TDRE 标志置 “1”。

设备 ID 地址检测功能, 在与自己的从属地址不匹配时或者自己的从属地址匹配后的重新开始条件后面的地址与设备 ID 地址不匹配时, 将 DID 标志置 “0”。在开始条件或者重新开始条件之后的第 1 个字节与设备 ID 地址 (1111 100b) 匹配并且 R/W# 位为 “0” 时, 将 DID 标志置 “1”, 并且将第 2 个字节以后的地址和从属地址进行比较。当 R/W# 位为 “1” 时, DID 标志继续保持原来的值, 并且不比较第 2 个字节以后的从属地址。因此, 通过在确认 TDRE 为 “1” 后检查 DID 标志, 能确认接收到设备 ID。

在接收一连串的设备 ID 后, 要发送给主机的设备 ID 字段所需的信息 (3 个字节: 厂商 [12 位] + 部件识别 [9 位] + 版本 [3 位]) 和通常的发送数据一样, 必须预先准备。有关设备 ID 字段所需信息的详细内容, 请向 NXP 公司询问。

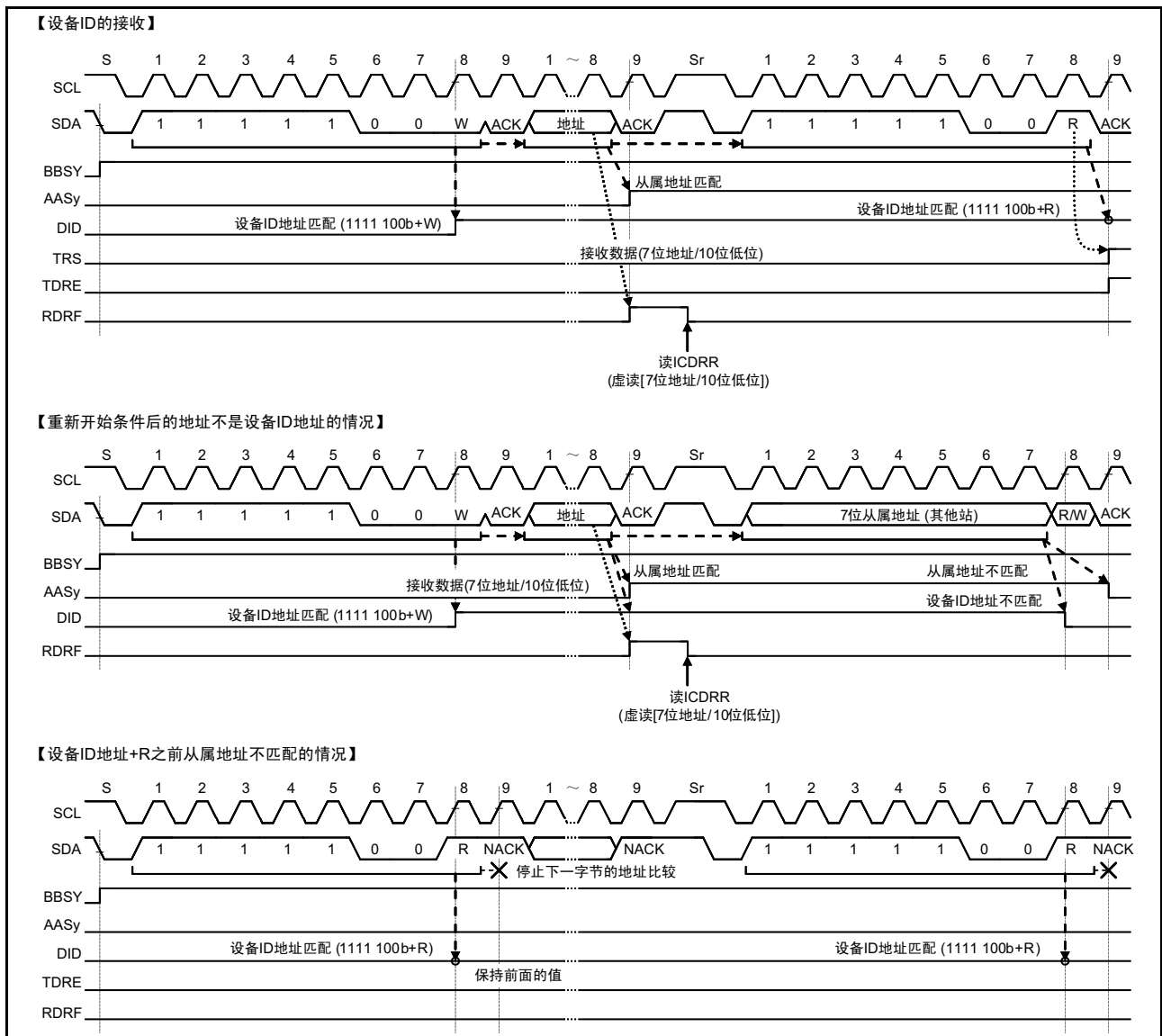


图 24.27 接收设备 ID 地址时的 AASy 标志和 DID 标志的置位 / 清除时序

### 24.7.4 主机地址检测功能

RIIC 具有 SMBus 运行时的主机地址检测功能。如果在 ICMR3.SMBS 位为“1”时将 IC SER.HOAE 位置“1”，就能在从属接收模式 (ICCR2.MST 位和 ICCR2.TRS 位为“00b”) 中检测主机地址 (0001 000b)。

如果 RIIC 检测到主机地址，就在 SCL 时钟的第 9 个时钟的下降沿将 ICSR1.HOA 标志置“1”，并且在 W<sub>r</sub> 位的情况下 (R/W# 位接收到“0”)，将 ICSR2.TDRE 标志置“1”。因此，能发生发送数据空中断 (ICTXI)，并且能通过 HOA 标志，确认从智能电池等发送了主机地址。

即使接在主机地址 (0001 000b) 后面的位是 R<sub>d</sub> 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接受运行相同。

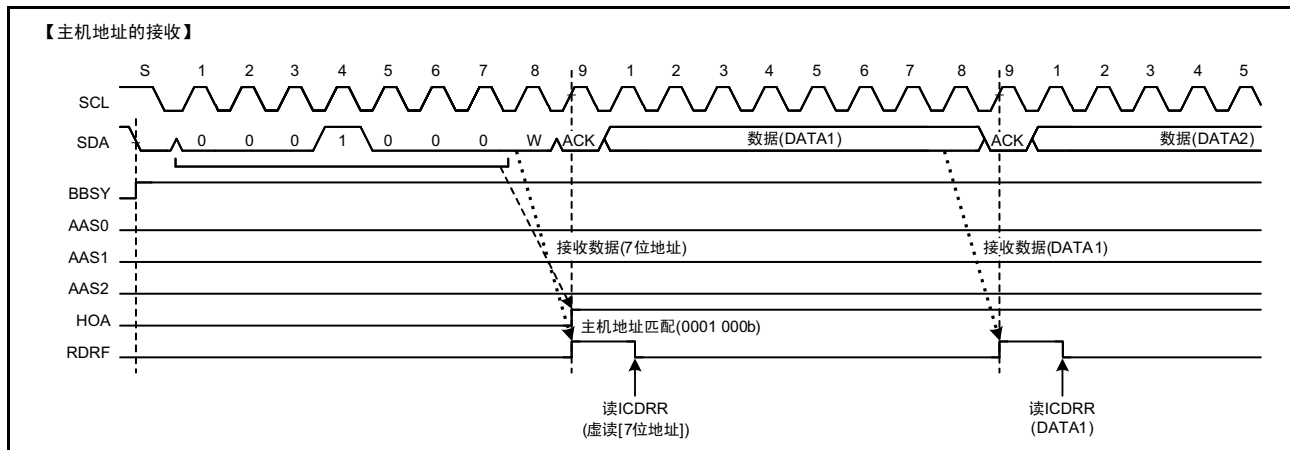


图 24.28 接收主机地址时的 HOA 标志变为“1”的时序

## 24.8 SCL 的 Low 电平自动保持功能

### 24.8.1 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

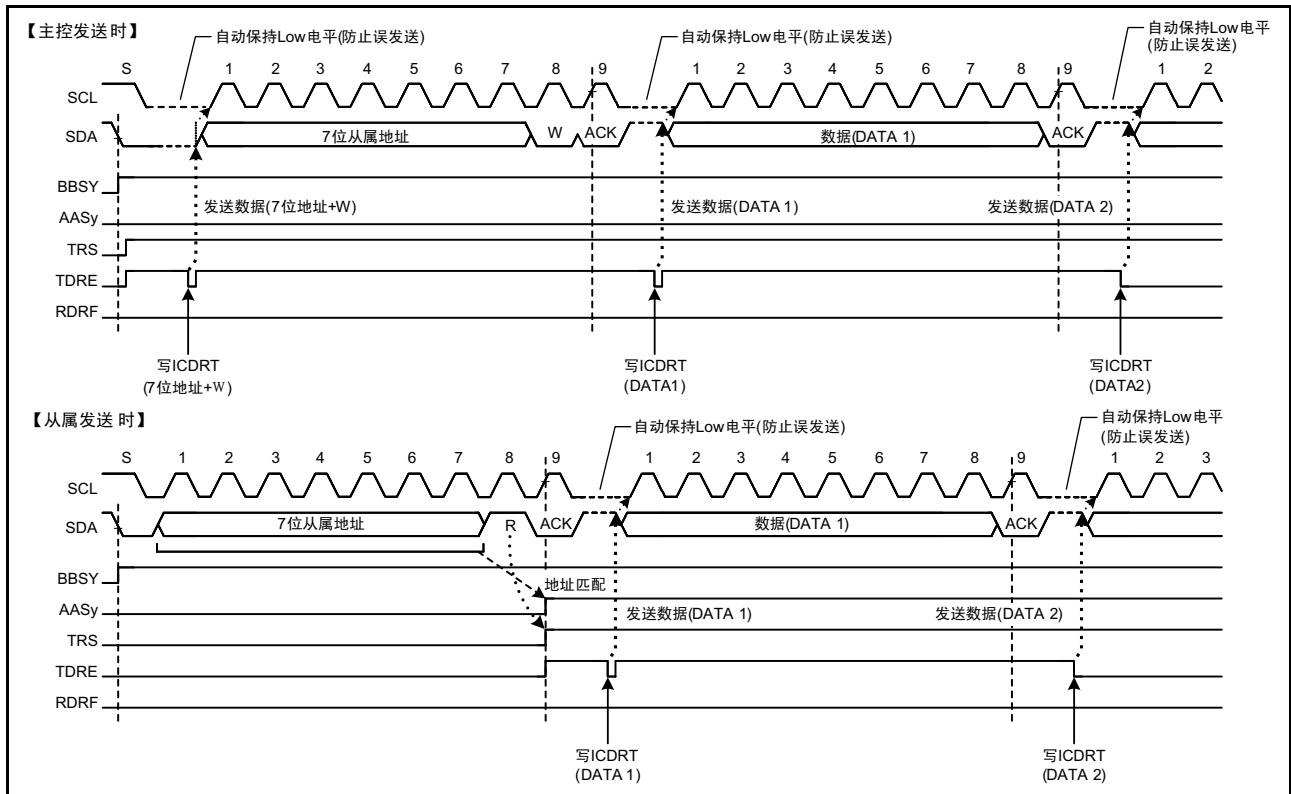


图 24.29 发送模式的 Low 电平自动保持运行



### 24.8.2 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 标志为“1”), 就不进行以后的发送和接收。要重新开始传送时, 必须将 NACKF 标志置“0”。在主机发送模式中, 必须将 NACKF 标志置“0”, 然后在发行重新开始条件或者停止条件后发行开始条件, 重新开始传送。

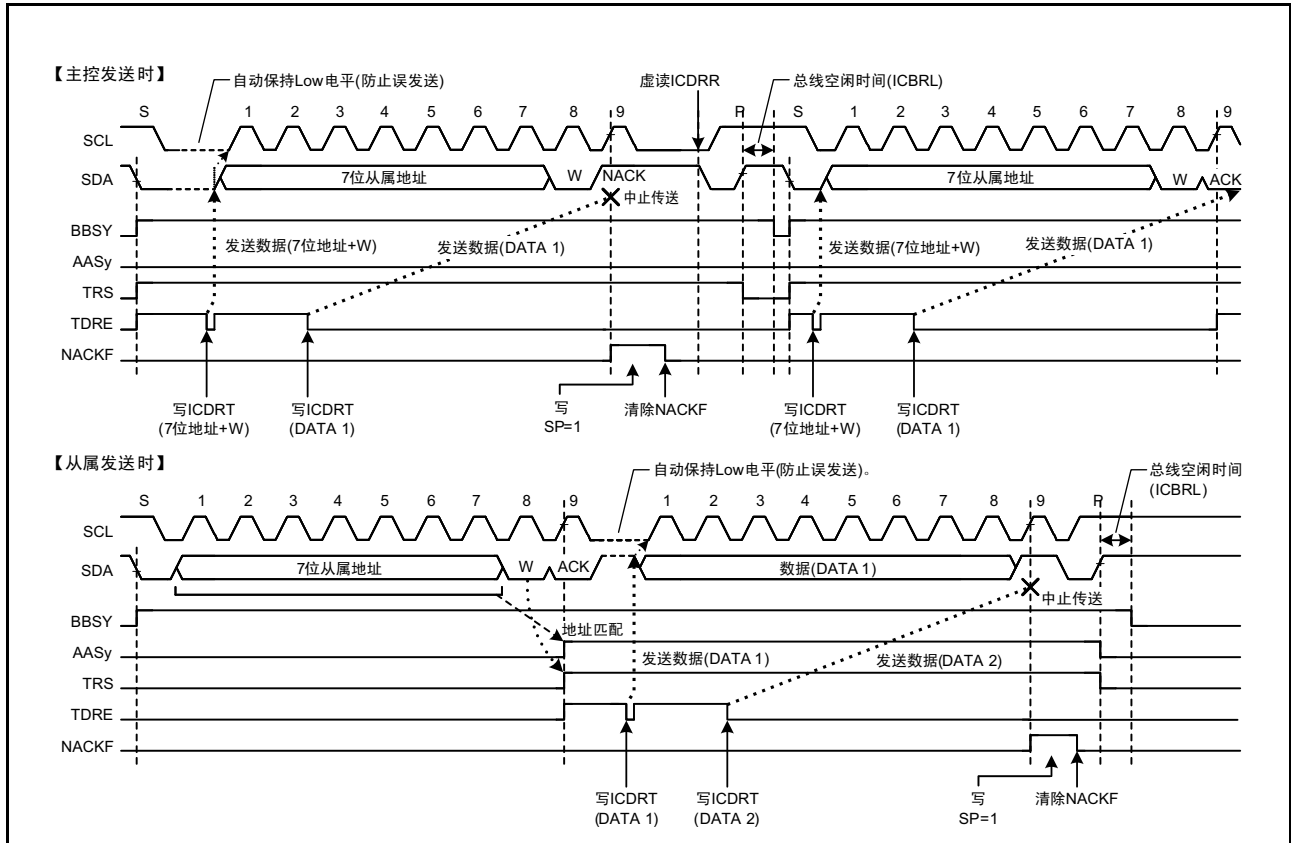


图 24.30 接收 NACK 时的传送中止 (NACKE=1)

### 24.8.3 接收数据漏取防止功能

如果在接收模式中 (ICCR2.TRS 位 =0) 并且接收数据满 (ICSR2.RDRF 标志 =1) 的状态下因至少推迟 1 个传送帧的读接收数据 (ICDRR 寄存器) 等而发生响应处理延迟, RIIC 就在开始下一个数据接收前自动保持 SCL 线的 Low 电平, 防止漏取接收数据。

在最后接收数据的读处理延迟期间, 即使在停止条件后指定了自己的从属地址, 通过自动保持 Low 电平防止漏取接收数据的功能仍然有效。因为在停止条件后自己的从属地址不匹配时不保持 Low 电平, 所以不妨碍其他通信。

RIIC 能通过 ICMR3.WAIT 位和 ICMR3.RDRFS 位的组合选择保持 Low 电平的区间。

(1) 通过 WAIT 位进行的 1 字节接收以及 Low 电平自动保持功能

如果将 ICMR3.WAIT 位置“1”，RIIC 就通过 WAIT 位的功能进行 1 字节的接收。当 ICMR3.RDRFS 位为“0”时，RIIC 就自动给从 SCL 时钟的第 8 个时钟的下降沿到第 9 个时钟的下降沿期间的应答位发送 ICMR3.ACKBT 位的内容，如果检测到第 9 个时钟的下降沿，就通过 WAIT 位的功能自动保持 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。

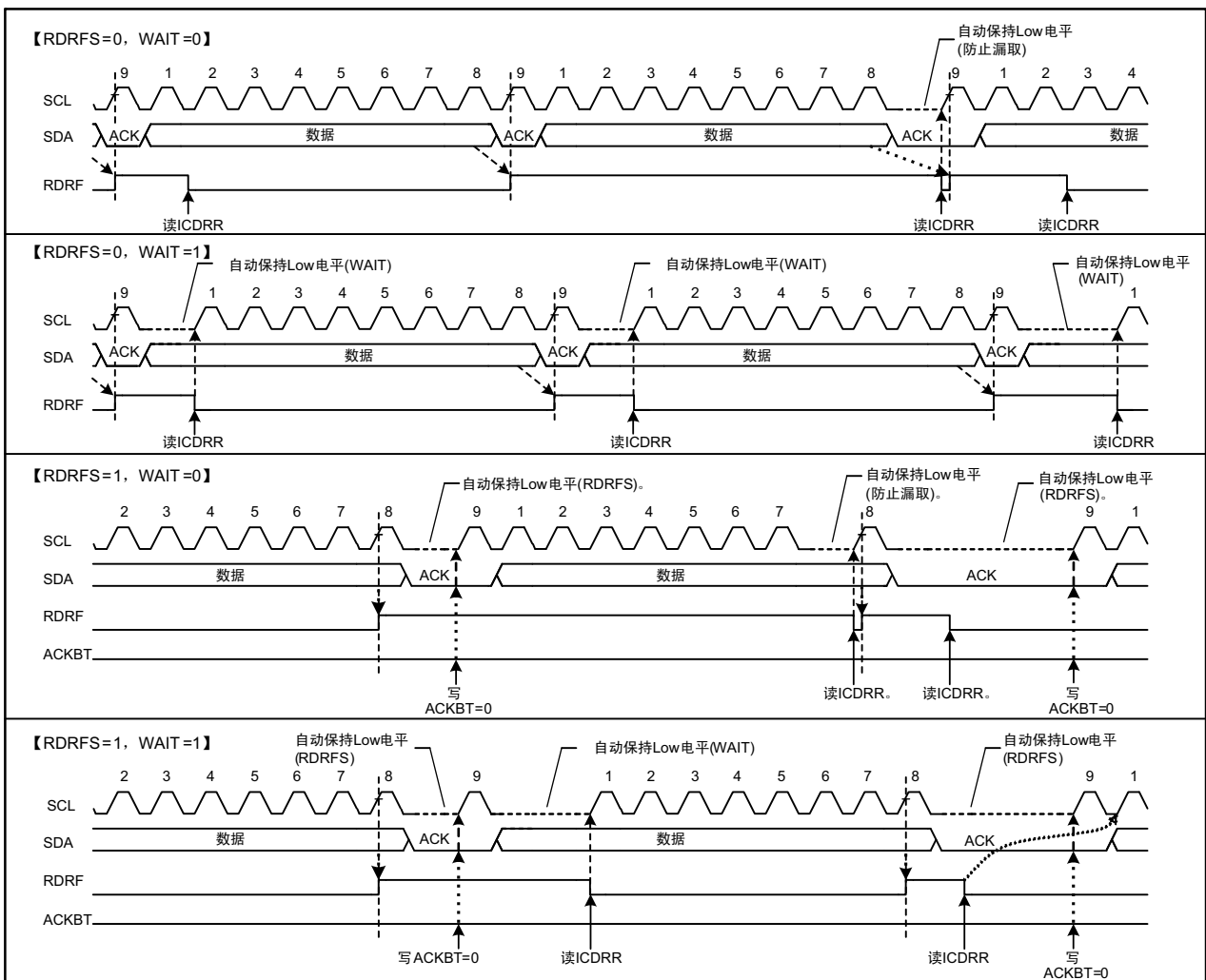


图 24.31 接收模式的 Low 电平自动保持（RDRFS 位和 WAIT 位）

## 24.9 仲裁失败检测功能

RIIC 除了 I<sup>2</sup>C 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。

### 24.9.1 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）与 SDA 线上的信号状态不同（自身发送的 SDA 输出为 High 电平输出（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 信号与 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
- 在主控发送模式中（ICCR2.MST 位和 ICCR2.TRS 位为“11b”），应答除外的发送数据（SDA 信号）与 SDA 线上的信号状态不同时

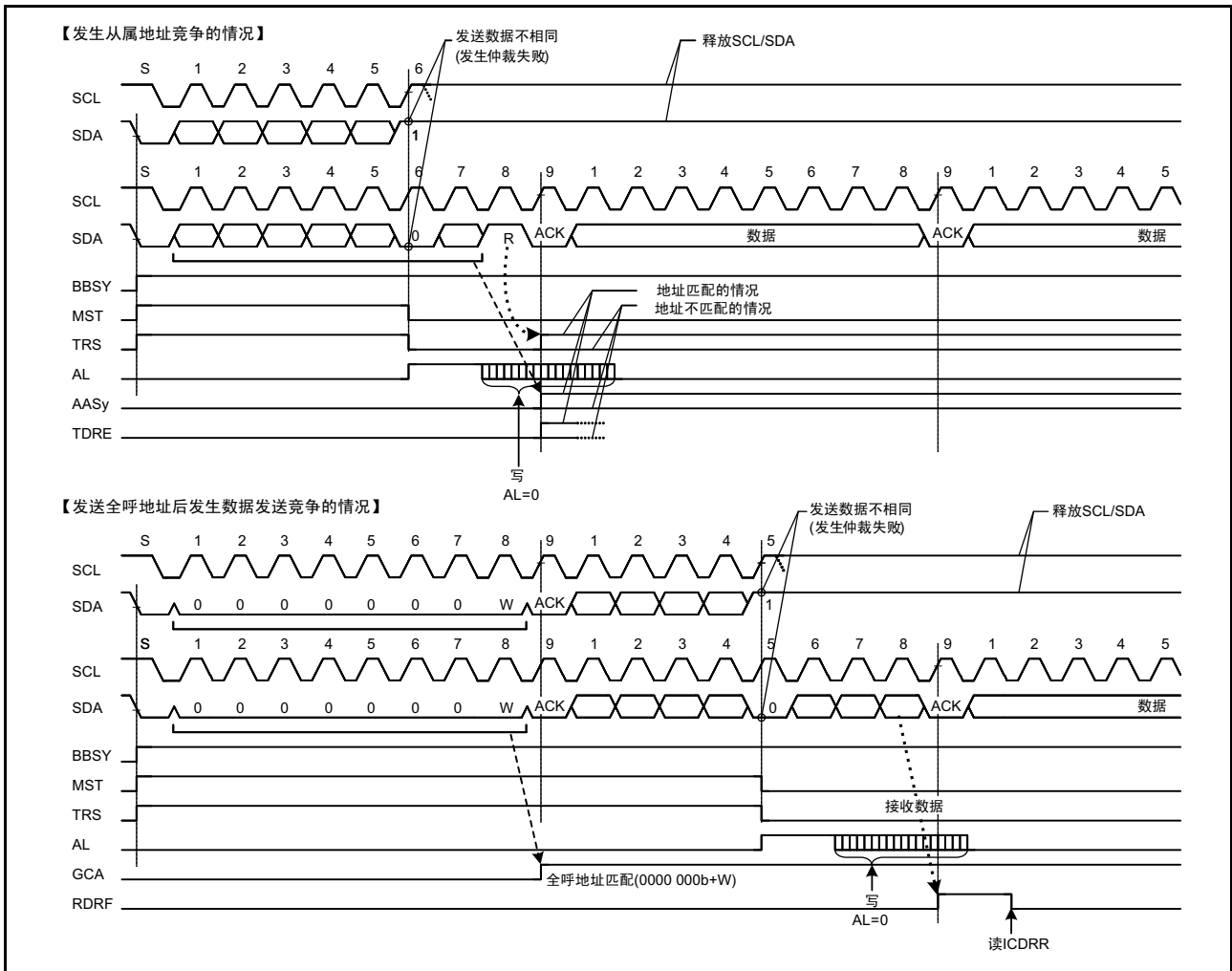


图 24.32 主控仲裁失败检测的运行例子 (MALE=1 的情况)

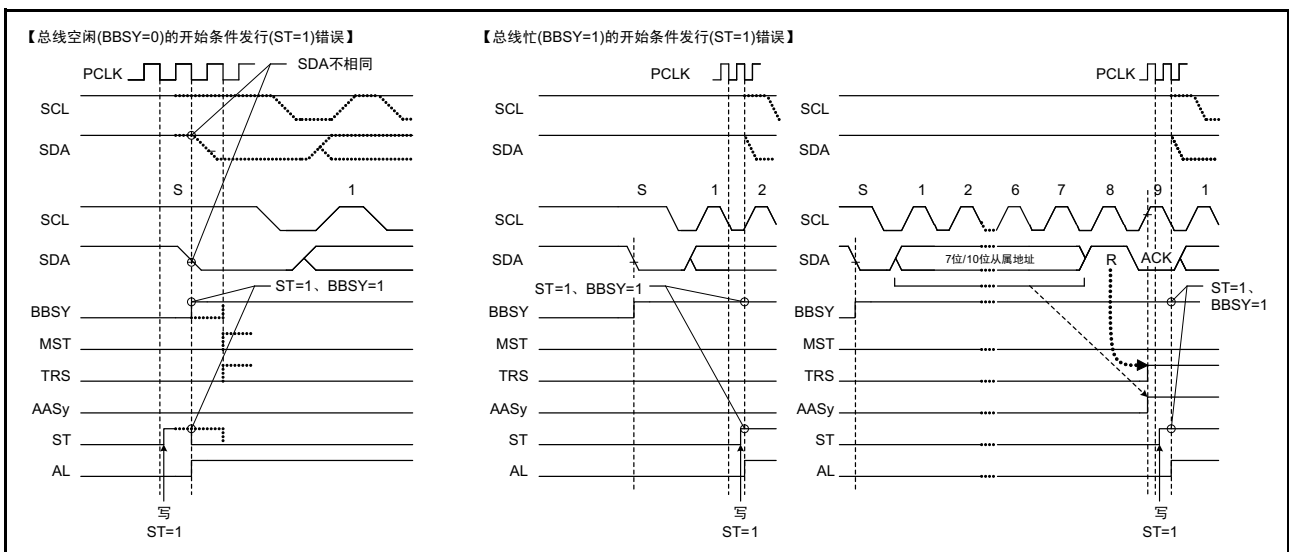


图 24.33 发行开始条件时的仲裁失败 (MALE=1 的情况)

### 24.9.2 NACK 发送仲裁失败检测功能 (NALE 位)

在接收模式中，如果在发送 NACK 时 RIIC 自身发送的 SDA 信号与 SDA 线上的信号状态不同（自身发送的 SDA 输出为 High 电平输出（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。主要在多主控系统中至少 2 个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。这种情况发生在至少 2 个的主控设备通过 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 24.34 所示。

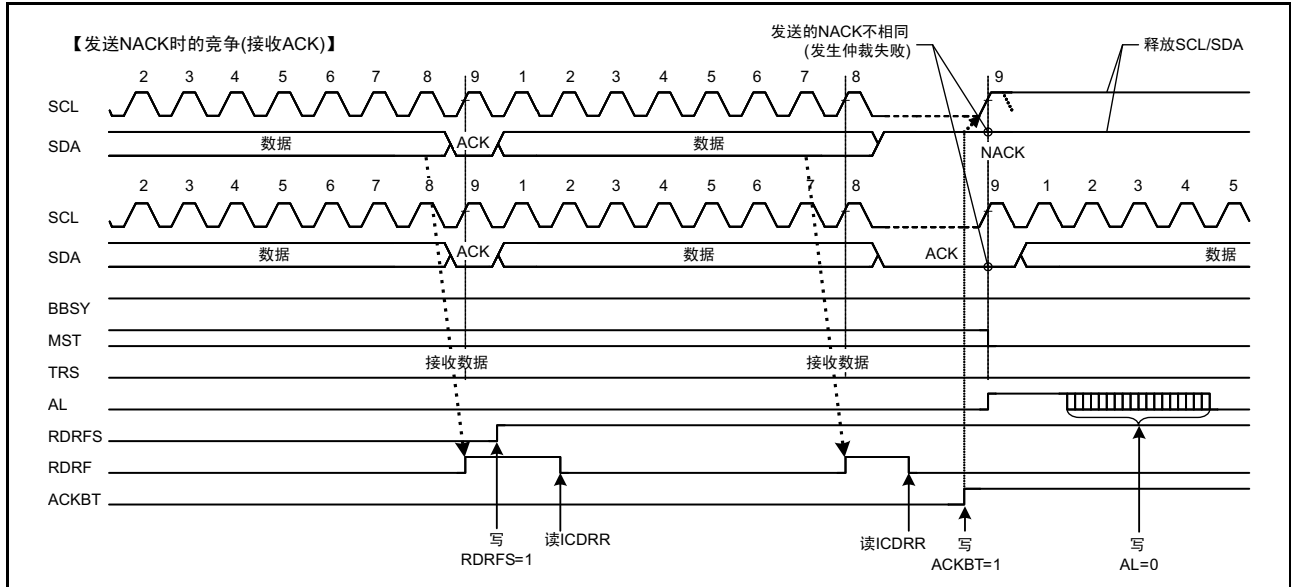


图 24.34 NACK 发送仲裁失败检测的运行例子 (NALE=1 的情况)

以 2 个主控设备（主控 A 和主控 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完全需要的 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（唯一设备标识别）不匹配时的 NACK 发送后以及确定指定地址后的 Get UDID（通用）的 NACK 发送后的剩余处理（FFh 发送处理）。

对于 NACK 发送仲裁失败检测，在 ICFER.NALE 位为“1”（允许 NACK 发送仲裁失败检测）的状态下以下条件成立时，检测到仲裁失败。

[NACK 发送仲裁失败条件]

- 在发送 NACK 时（ICMR3.ACKBT 位=1），自身发送的 SDA 信号与 SDA 线上的信号状态不同（接收到 ACK）。

### 24.9.3 从属仲裁失败检测功能 (SALE 位)

如果在从属发送时发送数据 (自身发送的 SDA 信号) 与 SDA 线上的信号状态不同 (自身发送的 SDA 输出为 High 电平输出 (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 就立即解除从属匹配状态并且转移到从属接收模式。

通过使用此功能, 能省去发送 SMBus 的 UDID 时的数据冲突检测和数据冲突后的剩余处理 (FFh 发送处理)。

对于从属仲裁失败检测, 在 ICFER.SALE 位为“1” (允许从属仲裁失败检测) 的状态下以下条件成立时, 检测到仲裁失败。

[ 从属仲裁失败条件 ]

- 在从属发送模式中 (ICCR2.MST 位和 ICCR2.TRS 位为“01b”), 应答除外的发送数据 (自身发送的 SDA 信号) 与 SDA 线上的信号状态不同。

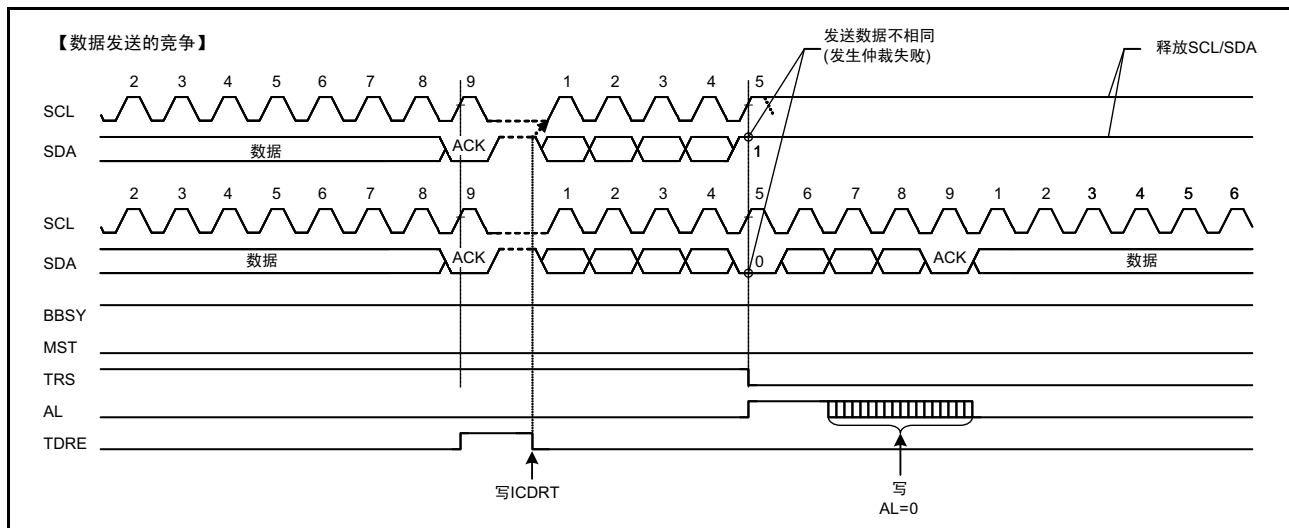


图 24.35 从属仲裁失败检测的运行例子 (SALE=1 的情况)

## 24.10 开始条件、重新开始条件和停止条件的发行功能

### 24.10.1 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。

### 24.10.2 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。

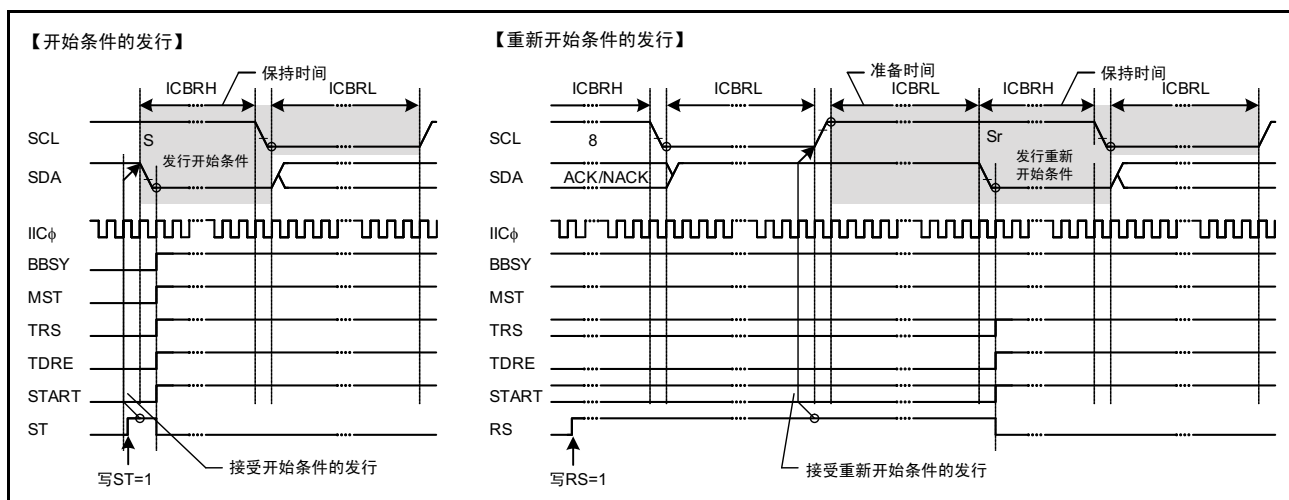


图 24.36 开始条件 / 重新开始条件的发行时序 (ST 位和 RS 位)

### 24.10.3 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）

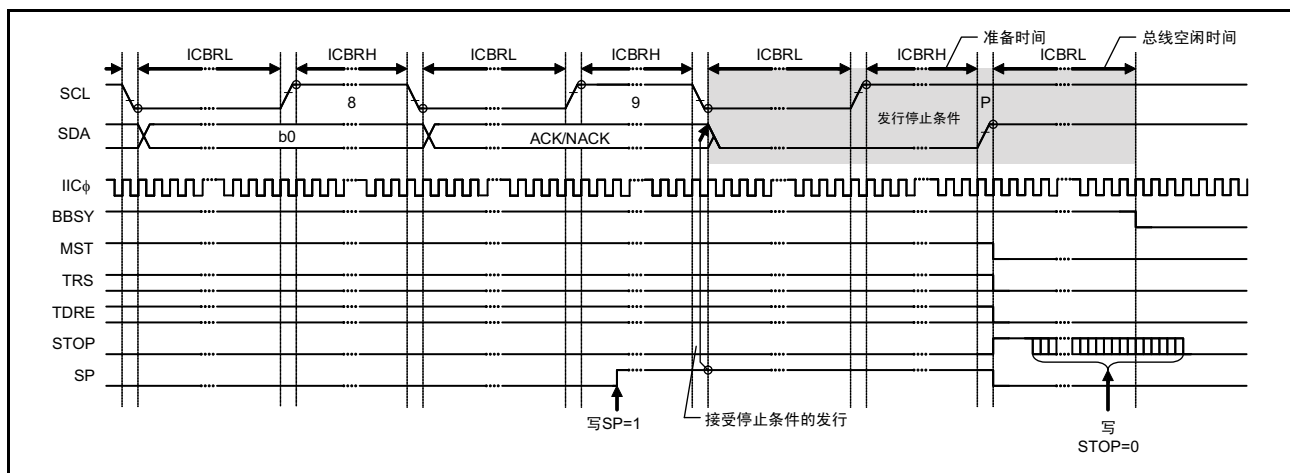


图 24.37 停止条件的发行时序（SP 位）



## 24.11 总线意外停机

如果 I<sup>2</sup>C 总线主要因噪声等的影响而在主控设备和从属设备之间失去同步，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（用于解除因失去同步引起的总线意外停机状态）和 RIIC/ 内部复位功能。

能通过 ICCR1.SCLO 位、ICCR1.SDAO 位、ICCR1.SCLI 位和 ICCR1.SDAI 位，确认是 RIIC 自身给 SCL 线 /SDA 线输出了 Low 电平还是通信设备输出了 Low 电平。

### 24.11.1 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 能通过监视到 SCL 线的 Low 电平或者 High 电平被固定了一定时间以上，并且检测到总线的异常状态。

超时检测功能监视 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 SCLn 线被固定为 Low 电平或者 High 电平的总线异常状态。

超时检测功能的内部计数器将 ICMR1.CKS[2:0] 位设定的内部基准时钟（IICφ）作为计数源，在选择长模式时（ICMR2.TMOS 位 =0），用作 16 位计数器；在选择短模式时（TMOS 位 =1），用作 14 位计数器。

对于内部计数器的计数，能通过设定 ICMR2.TMOH 位和 ICMR2.TMOL 位选择是在 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器。如果将 ICMR2.TMOH 位和 ICMR2.TMOL 位都置“0”，就不进行内部计数。

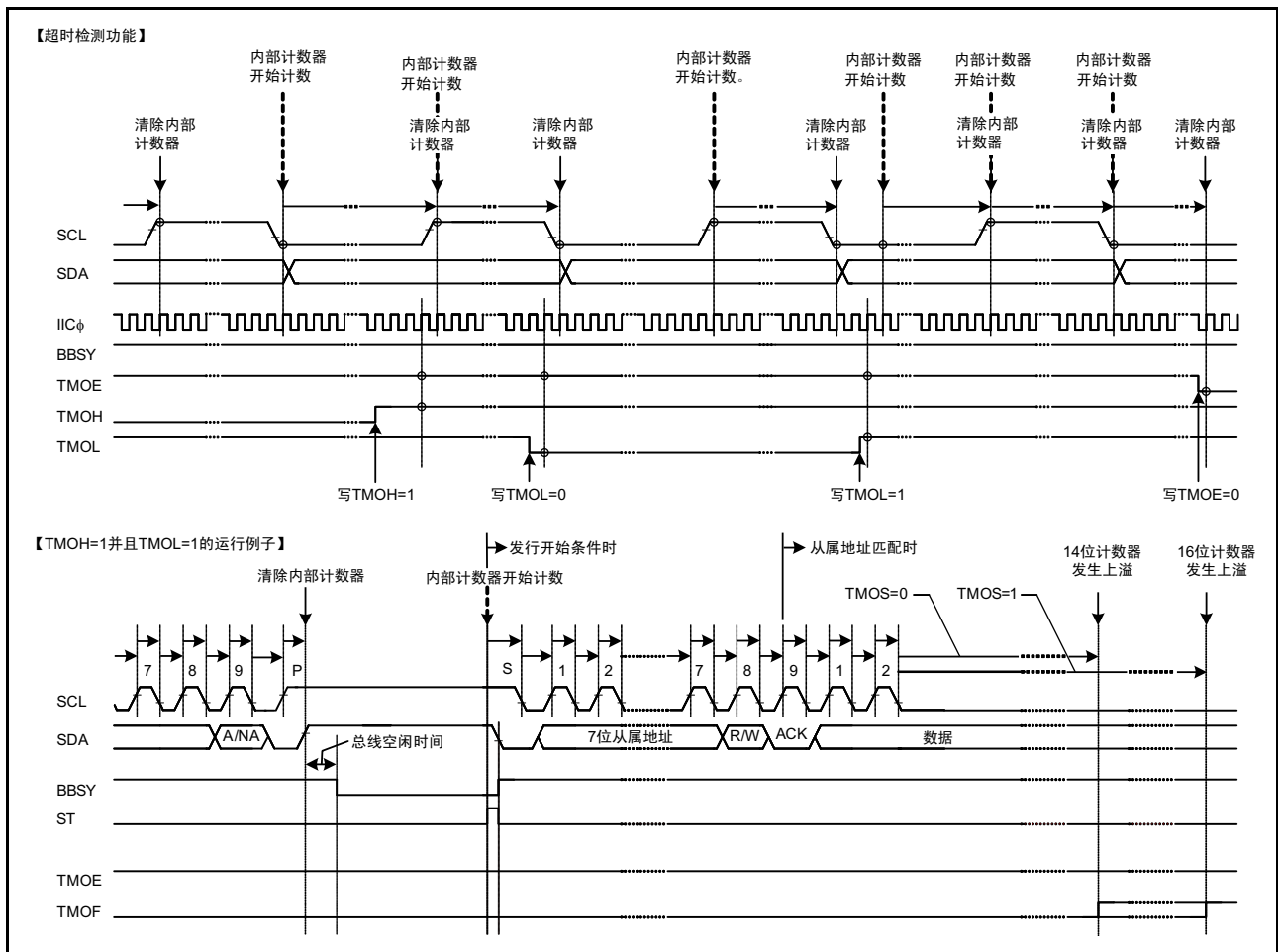


图 24.38 超时检测功能 (TMOE 位、TMOS 位、TMOH 位和 TMOL 位)

### 24.11.2 SCL 时钟追加输出功能

RIIC 具有 SCL 时钟的追加输出功能，在主导模式中用于释放因与从属设备失去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，主要用于在主导模式中从属设备在 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 寄存器所设传送率的 SCL 时钟作为 1 个时钟的追加时钟进行输出。一旦输出完 1 个时钟的追加时钟，CLO 位就自动变为“0”。因此，能通过软件在确认 CLO 位为“0”后写“1”，连续输出追加时钟。

在主导模式中，因噪声等影响引起和从属设备失去同步，导致从属设备将 SDA 线固定为 Low 电平的状态而使 RIIC 无法发行停止条件。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 位的值与 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）下或者在主导模式（ICCR2.MST 位或者 BBSY 标志为“1”的状态）中
- 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 24.39 所示。

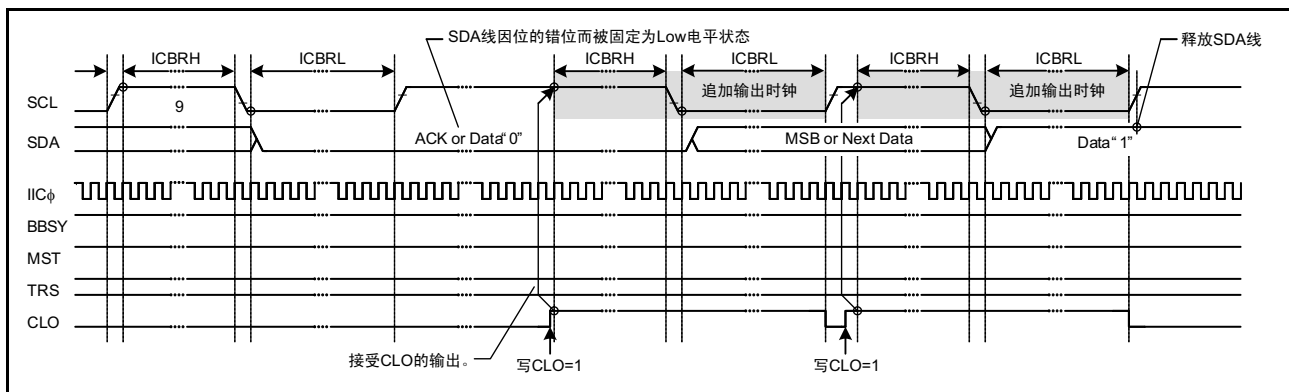


图 24.39 SCL 时钟追加输出功能（CLO 位）

### 24.11.3 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。

从属运行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“24.14 复位状况”。

## 24.12 SMBus 运行

RIIC 能进行以 SMBus (Ver.2.0) 为基准的通信。在进行 SMBus 通信时, 必须将 ICMR3.SMBS 位置“1”。必须通过设定 ICMR1.CKS[2:0] 位、ICBRH 寄存器和 ICBRL 寄存器, 将传送率设定在 SMBus 规格的 10kbps ~ 100kbps 范围内, 通过设定 ICMR2.DLCS 位和 ICMR2.SDDL[2:0] 位的值, 使数据的保持时间符合 300ns (min) 的规格。在只将 RIIC 用作从属设备时, 不需要设定传送率, 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 SARUy.FS 位 (y=0 ~ 2) (7 位 /10 位地址格式选择位) 置“0” (7 位地址格式)。

在发送 UDID (唯一设备标识符) 时, 必须将 ICFER.SALE 位置“1”, 使从属仲裁失败检测功能有效。

### 24.12.1 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 RX62T 群的内部定时器, 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]  $T_{\text{LOW:SEXT}}$ : 25ms (max) 以内。

如果通过 RX62T 群的内部定时器测量的时间超过 SMBus 规格的时钟 Low 电平检测的超时  $T_{\text{TIMEOUT}}$ : 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 RX62T 群的内部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]  $T_{\text{LOW:MEXT}}$ : 10ms (max) 以内, 从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 个时钟的上升沿); 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 RX62T 群的内部定时器测量的时间超过 SMBus 规格的时钟 Low 电平的累积时间 [主控设备]  $T_{\text{LOW:MEXT}}$ : 10ms (max), 或者各测量时间的累加结果超过 SMBus 规格的时钟 Low 电平检测的超时  $T_{\text{TIMEOUT}}$ : 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。

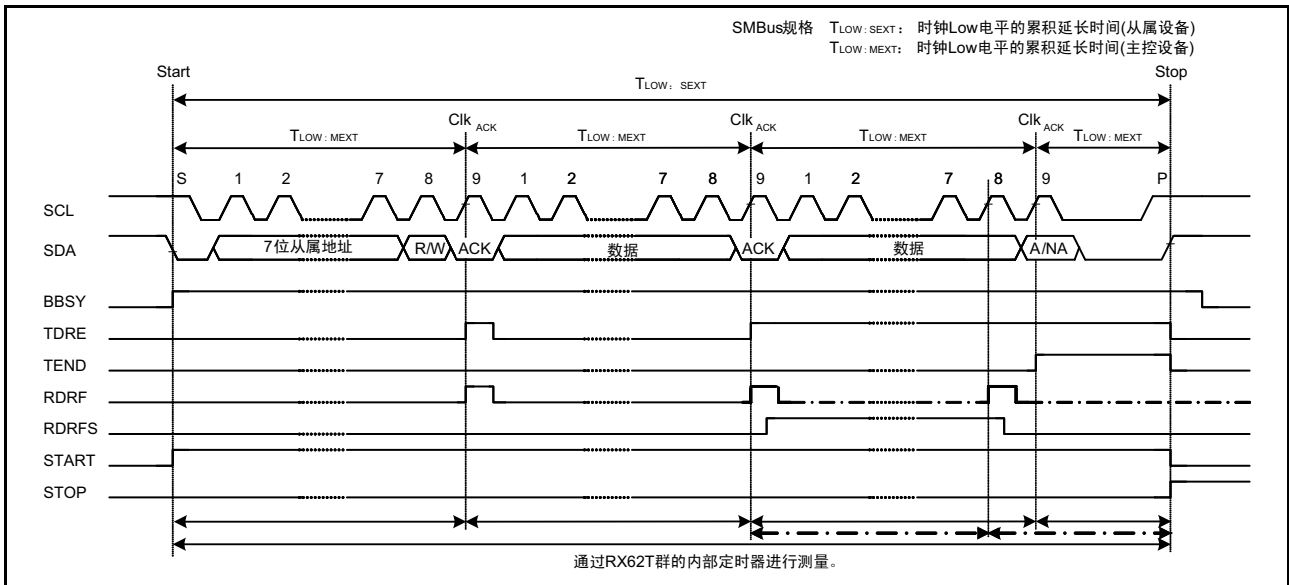


图 24.40 SMBus 超时测量

### 24.12.2 数据包错误码 (PEC)

RX62T 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“23. CRC 运算器 (CRC)”。

能通过给 CRC 运算器的 CRC 数据输入寄存器 (CRCDIR) 写全部的发送数据，生成主控发送 (主控发送器) 的 PEC 数据。

通过给 CRC 运算器的 CRCDIR 寄存器写全部的接收数据，得到 CRC 数据输出寄存器 (CRCDOR) 的值，然后将此值和接收的 PEC 数据进行比较，检查主控接收 (主控接收器) 的 REC 数据。

在检查 PEC 码的过程中接收最后的字节时，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。

### 24.12.3 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，将自己的自从属地址通知给 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 RX62T 群用作 SMBus 主机 (或者 ARP 主控) 时，需要将主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运行相同。

### 24.13 中断源

RIIC 的中断源有 4 种，分别是通信错误 / 事件的发生（仲裁失败检测、NACK 检测、超时检测、开始条件检测、停止条件检测）、接收数据满、发送数据空、发送结束。

中断一览表如表 24.7 所示。能根据接收数据满和发送数据空的中断请求，在启动 DTC 后进行数据传送。

表 24.7 中断源

名称	中断源	中断标志	DTC 的启动	优先级	中断条件
ICEEI 通信错误 / 事件的发生		AL	不能	高 ▲	AL=1 并且 ALIE=1
		NACKF			NACKF=1 并且 NAKIE=1
		TMOF			TMOF=1 并且 TMOIE=1
		START			START=1 并且 STIE=1
		STOP			STOP=1 并且 SPIE=1
ICRXI	接收数据满	—	能	低 	RDRF=1 并且 RIE=1
ICTXI	发送数据空	—	能		TDRE=1 并且 TIE=1
ICTEI	发送结束	TEND	不能		TEND=1 并且 TEIE=1

必须在中断处理中清除或者屏蔽各自的标志。

#### 【中断处理时的注意事项】

1. CPU 执行写外围模块的指令和实际写模块的时序有延时。如果清除或者屏蔽中断标志，就必须重新读中断标志，并且在确认清除或者写屏蔽位结束后从中断处理返回。如果不确认已写完模块而从中断处理返回，就可能再次产生相同的中断。
2. 因为 ICTXI 中断为边沿中断，所以不需要清除。通过给 ICDRT 寄存器写发送数据或者检测到停止条件（ICSR2.STOP 标志 =1），作为 ICTXI 中断条件的 ICSR2.TDRE 标志自动变为“0”。
3. 因为 ICRXI 中断为边沿中断，所以不需要清除。通过读 ICDRR 寄存器，作为 ICRXI 中断条件的 ICSR2.RDRF 标志自动变为“0”。
4. 在使用 ICTEI 中断时，必须在 ICTEI 中断处理中清除 ICSR2.TEND 标志。通过给 ICDRT 寄存器写发送数据或者检测到停止条件（ICSR2.STOP 标志 =1），ICSR2.TEND 标志自动变为“0”。

## 24.14 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 24.8 所示。

表 24.8 复位状况

		芯片复位	RIIC 复位 (ICE 位 =0 并且 IICRST 位 =1)	内部复位 (ICE 位 =1 并且 IICRST 位 =1)	开始条件 / 重新开始 条件的检测	停止条件的检测	
ICCR1	ICE、 IICRST	复位	保持	保持	运行 (保持)	运行 (保持)	
	SCLO、 SDAO		复位	复位			
	其他			保持			
ICCR2	BBSY	复位	复位	运行	运行	运行	
	ST			复位	复位	运行 (保持)	
	其他					复位	
ICMR1	BC[2:0]	复位	复位	复位	复位	运行 (保持)	
	其他			保持	运行 (保持)		
ICMR2		复位	复位	保持	运行 (保持)	运行 (保持)	
ICMR3		复位	复位	保持	运行 (保持)	运行 (保持)	
ICFER		复位	复位	保持	运行 (保持)	运行 (保持)	
ICSER		复位	复位	保持	运行 (保持)	运行 (保持)	
ICIER		复位	复位	保持	运行 (保持)	运行 (保持)	
ICSR1		复位	复位	复位	运行 (保持)	复位	
ICSR2	TDRE、 TEND	复位	复位	复位	运行 (保持)	复位	
	START				运行		
	STOP				运行 (保持)		运行
	其他						运行 (保持)
SARL0、SARL1、 SARL2 SARU0、SARU1、 SARU2		复位	复位	保持	运行 (保持)	运行 (保持)	
ICBRH、ICBRL		复位	复位	保持	运行 (保持)	运行 (保持)	
ICDRT		复位	复位	保持	运行 (保持)	运行 (保持)	
ICDRR		复位	复位	保持	运行 (保持)	运行 (保持)	
ICDRS		复位	复位	复位	运行 (保持)	运行 (保持)	
超时检测功能		复位	复位	运行	运行	运行	
总线空闲时间的测量		复位	复位	运行	运行	运行	

## 24.15 使用时的注意事项

### 24.15.1 模块停止功能的设定

能通过模块停止控制寄存器 B (MSTPCRB) 转移到模块停止状态或者解除模块停止状态, 初始值为 RIIC 处于模块停止状态。能通过解除模块停止状态, 使 RIIC 的寄存器变为可存取的状态。

有关模块停止控制寄存器 B 的详细内容, 请参照“9. 低功耗功能”。

### 24.15.2 输入缓冲控制寄存器的设定

能通过输入缓冲控制寄存器 (PORTn.ICR) 设定外围模块输入的有效或者无效, 初始值为 RIIC 输入无效状态。

因为 I<sup>2</sup>C 总线的 SCL 线和 SDA 线是双向线, 所以 RIIC 的 SCL 引脚和 SDA 引脚为输入 / 输出引脚。必须设定与 RIIC 的 SCL 引脚和 SDA 引脚对应的 PORTn.ICR 寄存器的输入缓冲控制位, 并且将 RIIC 的输入设定为有效。当对应引脚的输入处于无效状态时, 不能进行开始条件 (包括重新开始条件) 和停止条件的检测以及 SCL 时钟的计数。

有关输入缓冲控制寄存器的详细内容, 请参照“15. I/O 端口”。



## 25. CAN 模块 (CAN)

### 25.1 概要

RX62T 群内置 1 个通道的符合 ISO11898-1 规格的 CAN (Controller Area Network) 模块。CAN 模块能发送和接收标准 (11 位) IDentifier (以下省略为 ID) 和扩展 (29 位) ID 这两种格式的信息。

CAN 模块的规格和框图如表 25.1 和图 25.1 所示。

必须外接 CAN 总线收发器。

注. CAN 模块为选项。详细内容请参照表 1.3。

表 25.1 CAN 模块的规格 (1/2)

项目	概要
协议	<ul style="list-style-type: none"> <li>符合 ISO11898-1 规格 (标准帧 / 扩展帧)。</li> </ul>
位速率	<ul style="list-style-type: none"> <li>最大能设定 1Mbps 的位速率 (<math>f_{CAN} \geq 8\text{MHz}</math>)。</li> <li><math>f_{CAN}</math>: CAN 时钟源</li> </ul>
信息箱	<ul style="list-style-type: none"> <li>32 个信箱: 可选择 2 种信箱模式</li> <li>普通信箱模式: 能将 32 个信箱设定为发件箱或者收件箱。</li> <li>FIFO 信箱模式: 能将 24 个信箱设定为发件箱或者收件箱。</li> <li>能将剩余信箱的 4 段 FIFO 用作发件箱, 另 4 段 FIFO 用作收件箱。</li> </ul>
接收	<ul style="list-style-type: none"> <li>能接收数据帧和远程帧。</li> <li>可选择要接收的 ID 格式 (标准 ID 或者扩展 ID, 或者标准 ID 和扩展 ID)。</li> <li>可选择单次接收功能。</li> <li>可选择重写模式 (重写信息) 或者溢出模式 (放弃信息)。</li> <li>能分别将各信箱设定为允许或者禁止接收结束中断。</li> </ul>
验收滤波器	<ul style="list-style-type: none"> <li>8 个验收屏蔽 (每 4 个信箱有一个独立的屏蔽)</li> <li>能分别给各信箱设定屏蔽的有效或者无效。</li> </ul>
发送	<ul style="list-style-type: none"> <li>能发送数据帧和远程帧。</li> <li>可选择要发送的 ID 格式 (标准 ID 或者扩展 ID, 或者标准 ID 和扩展 ID)。</li> <li>可选择单次发送功能。</li> <li>可选择 ID 优先发送模式或者信箱号优先发送模式。</li> <li>能中止发送请求 (能通过标志确认中止结束)。</li> <li>能分别给各信箱设定允许或者禁止发送结束中断。</li> </ul>
总线断开恢复方法	<ul style="list-style-type: none"> <li>可选择从总线断开状态恢复的方法。</li> <li>符合 ISO11898-1 规格。</li> <li>在总线断开开始时自动转移到 CAN Halt 模式。</li> <li>在总线断开结束时自动转移到 CAN Halt 模式。</li> <li>通过程序转移到 CAN Halt 模式。</li> <li>通过程序转移到错误主动状态。</li> </ul>
错误状态的监视	<ul style="list-style-type: none"> <li>能监视 CAN 总线错误 (填充错误、格式错误、ACK 错误、CRC 错误、位错误、ACK 定界符错误)。</li> <li>能检测到错误状态的转移 (错误警告、错误被动、总线断开开始、总线断开返回)</li> <li>能读错误计数器。</li> </ul>
时戳功能	<ul style="list-style-type: none"> <li>16 位计数器的时戳功能</li> <li>基准时钟可选择 1 位、2 位、4 位或者 8 位时间。</li> </ul>
中断功能	<ul style="list-style-type: none"> <li>5 种中断源 (接收结束中断、发送结束中断、接收 FIFO 中断、发送 FIFO 中断、错误中断)</li> </ul>
CAN 睡眠模式	<ul style="list-style-type: none"> <li>能通过停止 CAN 时钟, 减少消耗电流。</li> </ul>

表 25.1 CAN 模块的规格 (2/2)

项目	概要
软件支持单元	<ul style="list-style-type: none"> <li>• 3 个软件支持单元</li> <li>支持验收滤波器。</li> <li>支持信箱的搜索 (收件箱的搜索、发件箱的搜索、信息丢失的搜索)。</li> <li>支持通道的搜索。</li> </ul>
CAN 时钟源	外围模块时钟 (PCLK)
测试模式	<ul style="list-style-type: none"> <li>• 用户评价有 3 个测试模式:</li> <li>只听模式</li> <li>自测试模式 0 (外部环回)</li> <li>自测试模式 1 (内部环回)</li> </ul>

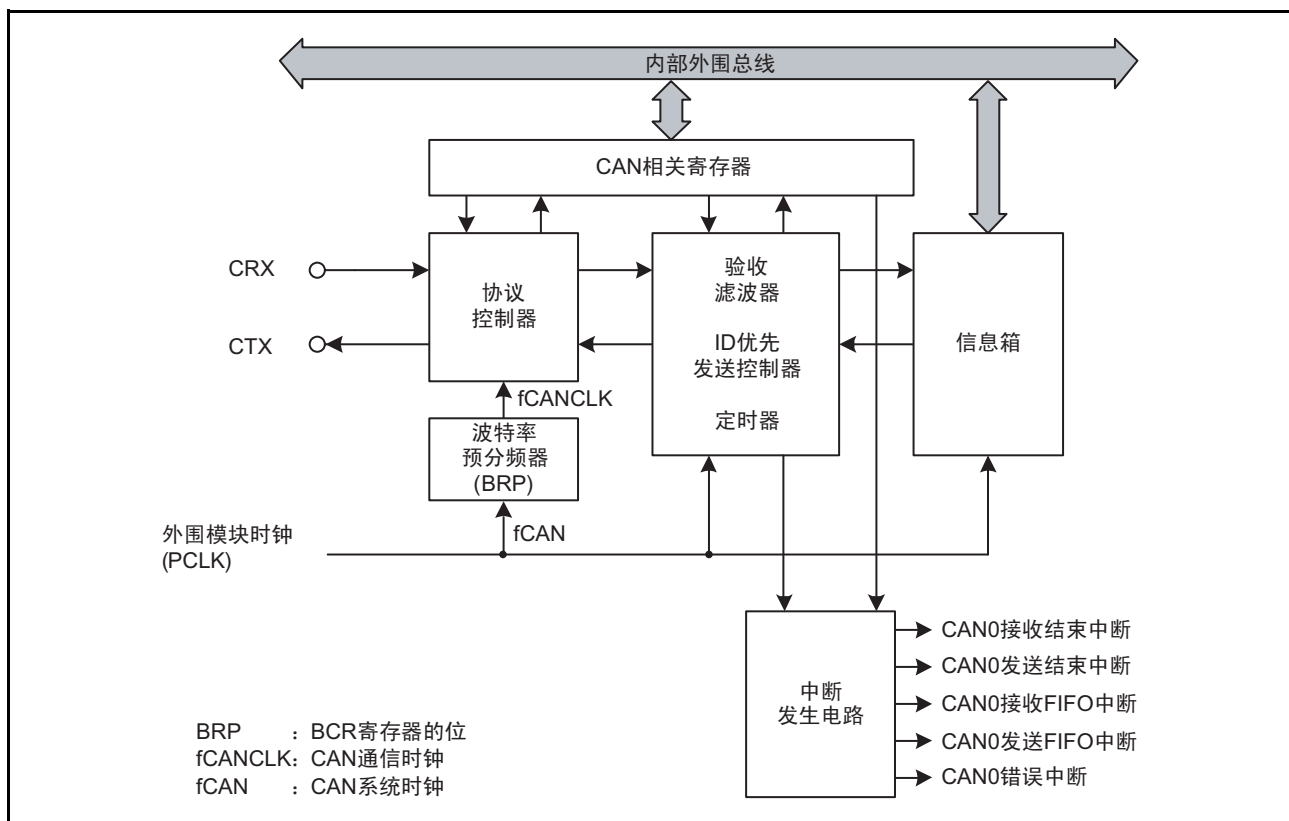


图 25.1 CAN 模块的框图

- CRX、CTX  
这是 CAN 的输入/输出引脚。
- 协议控制器  
进行总线仲裁以及发送/接收时的位时序、填充处理、错误处理等 CAN 协议处理。
- 信箱  
由能用作发件箱或者收件箱的 32 个信箱构成，有固有的 ID、数据长度代码、8 字节的数据字段和时戳。
- 验收滤波器  
对接收信息进行过滤处理。使用 MKR0~MKR7 寄存器进行过滤处理。
- 定时器  
用于时戳功能。在将信息保存到信箱时，将定时器的值作为时戳值进行写操作。

- 中断发生电路  
能产生以下5种中断请求：  
CAN0接收结束中断  
CAN0发送结束中断  
CAN0接收FIFO中断  
CAN0发送FIFO中断  
CAN0错误中断

CAN 模块使用的引脚如表 25.2 所示。

和其他功能多路复用的引脚需要进行引脚转换，详细内容请参照“15. I/O 端口”。

表 25.2 CAN 模块的引脚结构

模块符号	引脚名	输入 / 输出	功能
CAN0	CRX	输入	数据接收引脚
	CTX	输出	数据发送引脚

## 25.2 寄存器说明

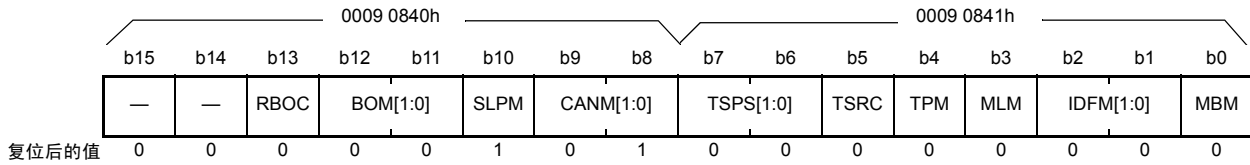
CAN 模块的寄存器结构如表 25.3 所示。

表 25.3 CAN 模块的寄存器结构

通道	寄存器名	寄存器符号	复位后的值	地址	存取长度
CAN0	控制寄存器	CTLR	0500h	0009 0840h	8、16
	位配置寄存器	BCR	0000 0000h	0009 0844h	8、16、32
	屏蔽寄存器 0	MKR0	不定值	0009 0400h	8、16、32
	屏蔽寄存器 1	MKR1	不定值	0009 0404h	8、16、32
	屏蔽寄存器 2	MKR2	不定值	0009 0408h	8、16、32
	屏蔽寄存器 3	MKR3	不定值	0009 040Ch	8、16、32
	屏蔽寄存器 4	MKR4	不定值	0009 0410h	8、16、32
	屏蔽寄存器 5	MKR5	不定值	0009 0414h	8、16、32
	屏蔽寄存器 6	MKR6	不定值	0009 0418h	8、16、32
	屏蔽寄存器 7	MKR7	不定值	0009 041Ch	8、16、32
	FIFO 接收 ID 比较寄存器 0	FIDCR0	不定值	0009 0420h	8、16、32
	FIFO 接收 ID 比较寄存器 1	FIDCR1	不定值	0009 0424h	8、16、32
	屏蔽无效寄存器	MKIVLR	不定值	0009 0428h	8、16、32
	信箱寄存器 0 ~ 31	MB0 ~ 31	不定值	0009 0200h ~ 0009 03FFh	8、16、32
	信箱中断允许寄存器	MIER	不定值	0009 042Ch	8、16、32
	信息控制寄存器 0 ~ 31	MCTL0 ~ 31	00h	0009 0820h ~ 0009 083Fh	8
	接收 FIFO 控制寄存器	RFCR	80h	0009 0848h	8
	接收 FIFO 指针控制寄存器	RFPCR	不定值	0009 0849h	8
	发送 FIFO 控制寄存器	TFCR	80h	0009 084Ah	8
	发送 FIFO 指针控制寄存器	TFPCR	不定值	0009 084Bh	8
	状态寄存器	STR	0500h	0009 0842h	8、16
	信箱搜索模式寄存器	MSMR	00h	0009 0853h	8
	信箱搜索状态寄存器	MSSR	80h	0009 0852h	8
	通道搜索支持寄存器	CSSR	不定值	0009 0851h	8
	验收滤波器支持寄存器	AFSR	不定值	0009 0856h	8、16
	错误中断允许寄存器	EIER	00h	0009 084Ch	8
	错误中断源判断寄存器	EIFR	00h	0009 084Dh	8
	接收错误计数寄存器	RECR	00h	0009 084Eh	8
	发送错误计数寄存器	TECR	00h	0009 084Fh	8
	错误代码保存寄存器	ECSR	00h	0009 0850h	8
	时戳寄存器	TSR	0000h	0009 0854h	8、16
	测试控制寄存器	TCR	00h	0009 0858h	8

## 25.2.1 控制寄存器 (CTRL)

地址 0009 0840h



位	符号	位名	功能	R/W
b0	MBM	发件箱 / 收件箱模式选择位 (注1)	0: 普通信箱模式 1: FIFO 信箱模式	R/W
b2-b1	IDFM[1:0]	ID 格式模式位 (注1)	b2 b1 00: 标准 ID 模式 全部信箱 (包含 FIFO 信箱) 只支持标准 ID。 01: 扩展 ID 模式 全部信箱 (包含 FIFO 信箱) 只支持扩展 ID。 10: 混合 ID 模式 全部信箱 (包含 FIFO 信箱) 支持标准 ID 和扩展 ID。在普通信箱模式中, 通过对应信箱的 IDE 位指定标准 ID 和扩展 ID; 在 FIFO 信箱模式中, 通过对应信箱的 IDE 位指定信箱 [0] ~ [23]; 通过 FIDCR0 寄存器和 FIDCR1 寄存器的 IDE 位指定接收 FIFO; 通过信箱 [24] 的 IDE 位指定发送 FIFO。 1 1: 不能设定	R/W
b3	MLM	信息丢失模式选择位 (注2)	0: 重写模式 1: 溢出模式	R/W
b4	TPM	发送优先级模式选择位 (注2)	0: ID 优先发送模式 1: 信箱号优先发送模式	R/W
b5	TSRC	时戳计数器复位的位 (注4)	0: 不进行复位 1: 进行复位 (注3)	R/W
b7-b6	TSPS[1:0]	时戳预分频器选择位 (注1)	b7 b6 00: 每 1 位时间 01: 每 2 位时间 10: 每 4 位时间 11: 每 8 位时间	R/W
b9-b8	CANM[1:0]	CAN 运行模式选择位 (注5)	b9 b8 00: CAN 操作模式 01: CAN 复位模式 10: CAN Halt 模式 11: CAN 复位模式 (强制转移)	R/W
b10	SLPM	CAN 睡眠模式位 (注5、注6)	0: 不是 CAN 睡眠模式 1: CAN 睡眠模式	R/W
b12-b11	BOM[1:0]	总线断开恢复模式选择位 (注1)	b12 b11 00: 正常模式 (符合 ISO11898-1 规格) 01: 在总线断开开始时自动转移到 CAN Halt 模式 10: 在总线断开结束时自动转移到 CAN Halt 模式 11: 通过程序的请求转移到 CAN Halt 模式 (在总线断开恢复期间中)	R/W
b13	RBOC	总线断开强制恢复位 (注2)	0: 不进行任何操作 1: 强制从总线断开状态恢复 (注3)	R/W
b15-b14	—	保留位	读写值都为“0”。	R/W

- 注 1. 必须在 CAN 复位模式中更改 BOM[1:0]、TSPS[1:0]、TPM、MLM、IDFM[1:0] 和 MBM 位。
- 注 2. 在总线断开状态下，必须将 RBOC 位置“1”。
- 注 3. 在置“1”后自动变为“0”，读取值为“0”。
- 注 4. 在 CAN 操作模式中，必须将 TSRC 位置“1”。
- 注 5. 如果更改 CANM[1:0] 位和 SLPM 位，就必须通过 STR 寄存器确认模式是否已被转换。在模式被转换前，不能更改 CANM[1:0] 位和 SLPM 位。
- 注 6. 必须在 CAN 复位模式或者 CAN Halt 模式中更改 SLPM 位。要改写 SLPM 位时，只能将此位置“0”或者“1”。

#### MBM 位 (发件箱 / 收件箱模式选择位)

当 MBM 位为“0” (普通信箱模式) 时，将信箱 [0] ~ [31] 设定为发件箱或者收件箱。

当 MBM 位为“1” (FIFO 信箱模式) 时，将信箱 [0] ~ [23] 设定为发件箱或者收件箱，将信箱 [24] ~ [27] 设定为发送 FIFO，将信箱 [28] ~ [31] 设定为接收 FIFO。

将发送数据写到信箱 [24] (信箱 [24] 为发送 FIFO 的窗口信箱)，从信箱 [28] 读接收数据 (信箱 [28] 为接收 FIFO 的窗口信箱)。

信箱的设定如表 25.4 所示。

#### IDFM[1:0] 位 (ID 格式模式位)

IDFM[1:0] 位决定 ID 格式。

#### MLM 位 (信息丢失模式选择位)

此位指定将新信息取到未读信箱时的运行。能选择重写模式或者溢出模式。全部信箱 (包含接收 FIFO) 可选择重写模式或者溢出模式。

当 MLM 位为“0”时，全部信箱为重写模式，新信息覆盖信箱里的旧信息。

当 MLM 位为“1”时，全部信箱为溢出模式，放弃新信息。

#### TPM 位 (发送优先级模式选择位)

此位指定发送信息的优先级模式。

TPM 位可选择 ID 优先模式或者信箱号优先模式。全部信箱为 ID 优先发送或者信箱号优先发送。

当 TPM 位为“0”时，信箱为 ID 优先发送模式，发送优先级以 CAN 总线仲裁规则 (ISO11898-1 规格) 为基准。在 ID 优先发送模式的情况下，在普通信箱模式中，将信箱 [0] ~ [31] ID 和发送 FIFO 的发件箱 ID 进行比较；在 FIFO 信箱模式中，将信箱 [0] ~ [23] ID 和发送 FIFO 的发件箱 ID 进行比较。当至少 2 个信箱 ID 相同时，信箱号小的信箱优先。

只有发送 FIFO 内预定发送的信息为发送仲裁对象。在发送 FIFO 信息的发送过程中，发送 FIFO 内的下一个待机信息为发送仲裁对象。

当 TPM 位为“1”时，为信箱号优先发送模式，发件箱中信箱号最小的信箱优先。在 FIFO 信箱模式中，发送 FIFO 的优先级低于普通信箱 (信箱 [0] ~ [23])。

#### TSRC 位 (时戳计数器复位的位)

如果将 TSRC 位置“1”，TSR 寄存器就变为“0000h”，然后此位自动变为“0”。

#### TSPS[1:0] 位 (时戳预分频器选择位)

这些位选择时戳预分频器。时戳的基准时钟能选择 1 位、2 位、4 位或者 8 位时间。

**CANM[1:0] 位 (CAN 运行模式选择位)**

CANM[1:0] 位选择 CAN 模块的模式 (CAN 操作模式、CAN 复位模式和 CAN Halt 模式)。通过 SLPM 位设定 CAN 睡眠模式, 详细内容请参照“25.3 运行模式”。

如果通过设定 BOM[1:0] 位转移到 CAN Halt 模式, CANM[1:0] 位就自动变为“10b”。

**SLPM 位 (CAN 睡眠模式位)**

如果将 SLPM 位置“1”, CAN 模块就变为 CAN 睡眠模式; 如果将 SLPM 位置“0”, 就解除 CAN 睡眠模式。详细内容请参照“25.3 运行模式”。

**BOM[1:0] 位 (总线断开返回模式选择位)**

BOM[1:0] 位用于选择 CAN 模块的总线断开返回模式。

当 BOM[1:0] 位为“00b”时, 总线断开的恢复符合 ISO11898-1 规格。即, 在对 11 个连续的隐性位进行了 128 次的检测后, CAN 模块再次进行 CAN 通信 (错误主动状态)。在从总线断开状态恢复时, 产生总线断开恢复中断请求。

当 BOM[1:0] 位为“01b”时, 一旦 CAN 模块进入总线断开状态, 就转移到 CAN Halt 模式, 并且 CANM[1:0] 位变为“10b” (CAN Halt 模式)。在从总线断开状态恢复时, 不产生总线断开恢复中断请求, 但是 TECR 寄存器和 RECR 寄存器变为“00h”。

当 BOM[1:0] 位为“10b”时, 一旦 CAN 模块进入总线断开状态, CANM[1:0] 位就变为“10b”。在从总线断开状态恢复 (对 11 个连续的隐性位进行了 128 次的检测) 后, 转移到 CAN Halt 模式。在从总线断开状态恢复时, 产生总线断开恢复中断请求, 并且 TECR 寄存器和 RECR 寄存器变为“00h”。

当 BOM[1:0] 位为“11b”时, 如果在 CAN 模块还没有进入总线断开状态时将 CANM[1:0] 位置“10b”, 就进入 CAN Halt 模式。在从总线断开状态恢复时, 不产生总线断开恢复中断请求而 TECR 寄存器和 RECR 寄存器变为“00h”。但是, 如果在将 CANM[1:0] 位置“10b”前对 11 个连续的隐性位进行了 128 次的检测, 然后从总线断开状态恢复, 就产生总线断开恢复中断请求。

如果在 CAN 模块转移到 CAN Halt 模式的同时 (当 BOM[1:0] 位为“01b”时, 总线断开开始; 当 BOM[1:0] 位为“10b”时, 总线断开结束) CPU 请求向 CAN 复位模式的转移, 就优先 CPU 的请求。

**RBOC 位 (总线断开强制恢复位)**

如果在总线断开状态下将 RBOC 位置“1” (强制从总线断开恢复), 就强制从总线断开状态恢复。此后, 此位自动变为“0”, 错误状态从总线断开状态变为错误主动状态。如果将 RBOC 位置“1”, RECR 寄存器和 TECR 寄存器就变为“00h”, STR.BOST 位变为“0” (CAN 模块不是总线断开状态)。即使将 RBOC 位置“1”, 其他寄存器也不变。在从总线断开状态恢复时, 不产生总线断开恢复中断请求。只有在 BOM[1:0] 位为“00b” (正常模式) 时, 才能使用 RBOC 位。

表 25.4 信箱的设定

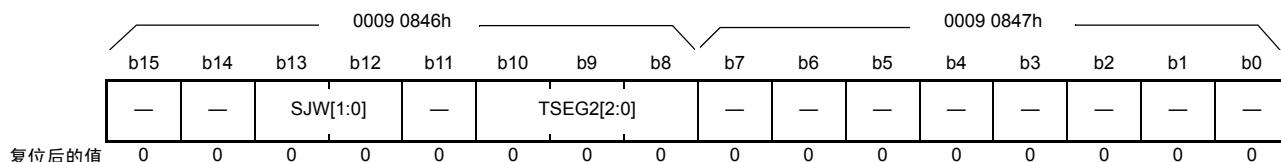
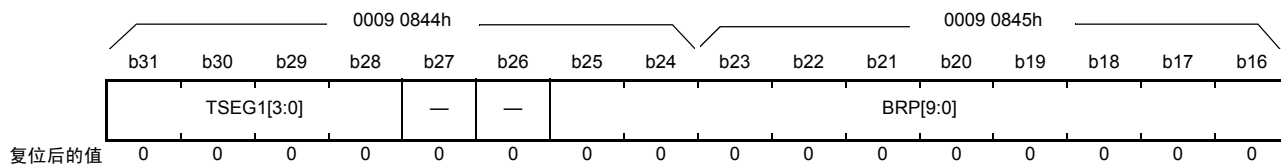
信箱	MBM 位 =0 (普通信箱模式)	MBM 位 =1 (FIFO 信箱模式)
信箱 [0] ~ [23]	普通信箱	普通信箱
信箱 [24] ~ [27]		发送 FIFO
信箱 [28] ~ [31]		接收 FIFO

注. 在 CTLR.MBM 位为“1”时, 必须注意以下 1. ~ 5. 的事项:

1. 通过 TFCR 寄存器控制发送 FIFO, 信箱 [24] ~ [27] 的 MCTLj (j=0~31) 寄存器无效。  
MCTL24 ~ MCTL27 寄存器不能用于发送 FIFO。
2. 通过 RFCR 寄存器控制接收 FIFO, 信箱 [28] ~ [31] 的 MCTLj (j=0~31) 寄存器无效。  
MCTL28 ~ MCTL31 寄存器不能用于接收 FIFO。
3. 有关 FIFO 中断, 请参照 MIER 寄存器。
4. MKIVLR 寄存器的信箱 [24] ~ [31] 的对应位无效。必须将这些位置“0”。
5. 发送/接收 FIFO 能使用数据帧和远程帧。

## 25.2.2 位配置寄存器 (BCR)

地址 0009 0844h



位	符号	位名	功能	R/W
b7-b0	—	保留位	读写值都为“0”。	R/W
b10-b8	TSEG2[2:0]	时间段 2 控制位	b10 b8 0 0 0: (不能设定) 0 0 1: 2Tq 0 1 0: 3Tq 0 1 1: 4Tq 1 0 0: 5Tq 1 0 1: 6Tq 1 1 0: 7Tq 1 1 1: 8Tq	R/W
b11	—	保留位	读写值都为“0”。	R/W
b13-b12	SJW[1:0]	再同步转移宽度控制位	b13 b12 0 0: 1Tq 0 1: 2Tq 1 0: 3Tq 1 1: 4Tq	R/W
b15-b14	—	保留位	读写值都为“0”。	R/W
b25-b16	BRP[9:0]	预分频器分频比选择位	设定 CAN 通信时钟 (fCANCLK) 的频率。	R/W
b27-b26	—	保留位	读写值都为“0”。	R/W
b31-28	TSEG1[3:0]	时间段 1 控制位	b31 b28 0 0 0 0: 不能设定 0 0 0 1: 不能设定 0 0 1 0: 不能设定 0 0 1 1: 4Tq 0 1 0 0: 5Tq 0 1 0 1: 6Tq 0 1 1 0: 7Tq 0 1 1 1: 8Tq 1 0 0 0: 9Tq 1 0 0 1: 10Tq 1 0 1 0: 11Tq 1 0 1 1: 12Tq 1 1 0 0: 13Tq 1 1 0 1: 14Tq 1 1 1 0: 15Tq 1 1 1 1: 16Tq	R/W

Tq: Time Quantum



BCR 寄存器用  $T_q$  值指定段的长度。

有关位时序的设定，请参照“25.4 CAN 通信速度的设定”。

必须在从 CAN 复位模式转移到 CAN Halt 模式或者 CAN 操作模式前设定 BCR 寄存器。一旦设定，就能在 CAN 复位模式或者 CAN Halt 模式中进行更改。

BCR 寄存器为 24 位。必须注意：在进行 32 位存取时，不能改写 b0 ~ b7。

#### TSEG2[2:0] 位 (时间段 2 控制位)

这些位用  $T_q$  值指定阶段缓冲段 2 (PHASE\_SEG2) 的长度，能设定 2 ~ 8 $T_q$  的值。必须设定小于 TSEG1[3:0] 位的值。

#### SJW[1:0] 位 (再同步转移宽度控制位)

这些位用  $T_q$  值指定再同步转移宽度 (Resynchronization Jump Width)，能设定 1 ~ 4 $T_q$  的值。必须设定不超过 TSEG2[2:0] 位的值。

#### BRP[9:0] 位 (预分频器分频比选择位)

这些位设定 CAN 通信时钟 (fCANCLK) 的频率。fCANCLK 的周期为 1 $T_q$ ，假设设定值为 P (0 ~ 1023)，波特率预分频器就对 fCAN 进行 P+1 分频。

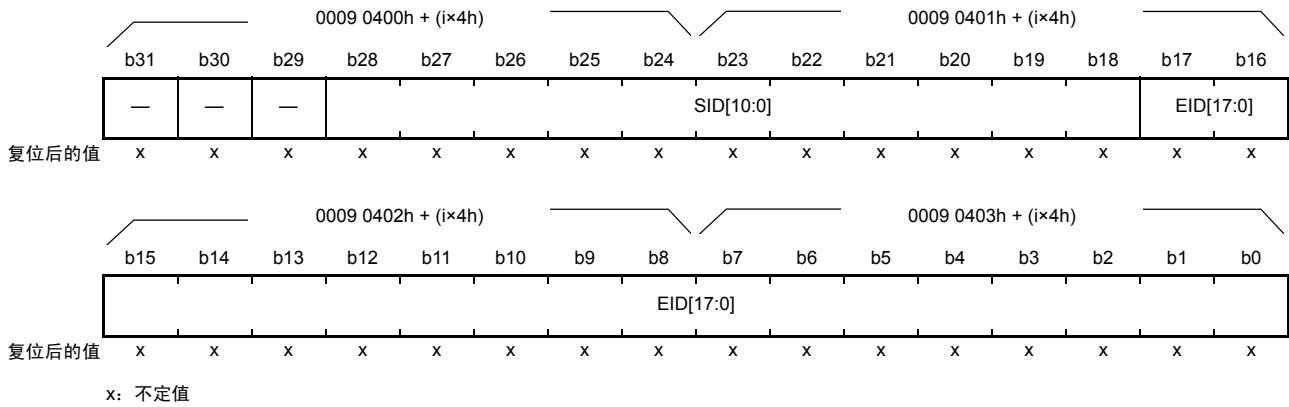
#### TSEG1[3:0] 位 (时间段 1 控制位)

这些位用 Time Quantum ( $T_q$ ) 值指定传输时间段 (PROP\_SEG) 和阶段缓冲段 1 (PHASE\_SEG1) 的总长度。

能设定 4 ~ 16 $T_q$  的值。

### 25.2.3 屏蔽寄存器 i (MKRi) (i=0 ~ 7)

地址 0009 0400h ~ 0009 041Ch



位	符号	位名	功能	R/W
b17-b0	EID[17:0]	扩展 ID 位	0: 不比较对应的 EID[17:0] 位 1: 比较对应的 EID[17:0] 位	R/W
b28-b18	SID[10:0]	标准 ID 位	0: 不比较对应的 SID[10:0] 位 1: 比较对应的 SID[10:0] 位	R/W
b31-b29	—	保留位	读取值为不定值, 只能写“0”。	R/W

有关 FIFO 信箱模式中的屏蔽功能, 请参照“25.6 验收过滤功能和屏蔽功能”。  
必须在 CAN 复位模式或者 CAN Halt 模式中更改 MKRi (i=0 ~ 7) 寄存器。

#### EID[17:0] 位 (扩展 ID 位)

EID[17:0] 位是对应 CAN 扩展 ID 位的过滤屏蔽位。

在接收扩展 ID 的信息时使用这些位。

当 EID[17:0] 位为“0”时, 对应的 EID[17:0] 位不将接收的 ID 和信箱 ID 进行比较。

当 EID[17:0] 位为“1”时, 对应的 EID[17:0] 位将接收的 ID 和信箱 ID 进行比较。

#### SID[10:0] 位 (标准 ID 位)

SID[10:0] 位是对应 CAN 标准 ID 位的过滤屏蔽位。

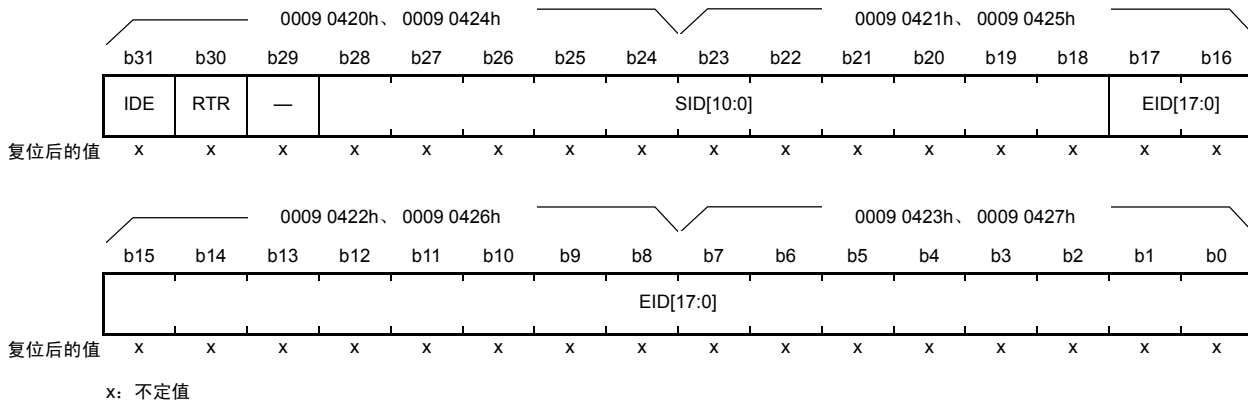
在接收标准 ID 和扩展 ID 的信息时使用这些位。

当 SID[10:0] 位为“0”时, 对应的 SID[10:0] 位不将接收的 ID 和信箱 ID 进行比较。

当 SID[10:0] 位为“1”时, 对应的 SID[10:0] 位将接收的 ID 和信箱 ID 进行比较。

## 25.2.4 FIFO 接收 ID 比较寄存器 0、1 (FIDCR0、FIDCR1)

地址 FIDCR0 0009 0420h  
FIDCR1 0009 0424h



位	符号	位名	功能	R/W
b17-b0	EID[17:0]	扩展 ID 位	0: 对应的 EID[17:0] 位为“0” 1: 对应的 EID[17:0] 位为“1”	R/W
b28-b18	SID[10:0]	标准 ID 位	0: 对应的 SID[10:0] 位为“0” 1: 对应的 SID[10:0] 位为“1”	R/W
b29	—	保留位	读取值为不定值，只能写“0”。	R/W
b30	RTR	远程发送请求位	0: 数据帧 1: 远程帧	R/W
b31	IDE	ID 扩展位 (注 1)	0: 标准 ID 1: 扩展 ID	R/W

注 1. 在 CTLR.IDFM[1:0] 位不是“10b”时，必须给 IDE 位写“0”。读取值为“0”。

在 CTLR.MBM 位为“1” (FIFO 信箱模式) 时，FIDCR0 寄存器和 FIDCR1 寄存器有效，MB28 ~ MB31 寄存器的 EID[17:0]、SID[10:0]、RTR 和 IDE 位无效。

有关 FIDCR0 寄存器和 FIDCR1 寄存器的使用方法，请参照“25.6 验收过滤功能和屏蔽功能”。

必须在 CAN 复位模式或者 CAN Halt 模式中更改 FIDCR0 寄存器和 FIDCR1 寄存器。

## EID[17:0] 位 (扩展 ID 位)

EID[17:0] 位设定数据帧和远程帧的扩展 ID。

在接收扩展 ID 的信息时使用这些位。

## SID[10:0] 位 (标准 ID 位)

SID[10:0] 位设定数据帧和远程帧的标准 ID。

在接收标准 ID 和扩展 ID 的信息时使用这些位。

### RTR 位 (远程发送请求位)

RTR 位设定被指定为数据帧或者远程帧的帧格式。

- 当 FIDCR0 寄存器和 FIDCR1 寄存器的 RTR 位都为“0”时，只能接收数据帧。
- 当 FIDCR0 寄存器和 FIDCR1 寄存器的 RTR 位都为“1”时，只能接收远程帧。
- 当 FIDCR0 寄存器的 RTR 位为“0”而 FIDCR1 寄存器的 RTR 位为“1”时，或者 FIDCR0 寄存器的 RTR 位为“1”而 FIDCR1 寄存器的 RTR 位为“0”时，能接收数据帧和远程帧。

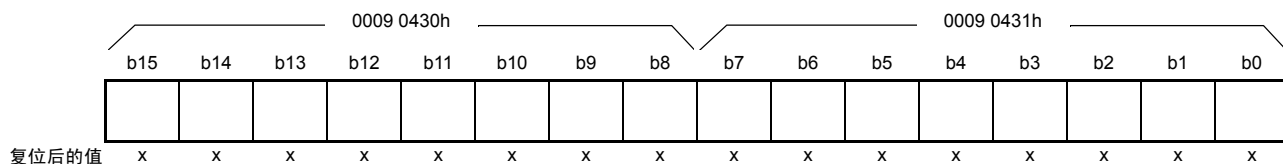
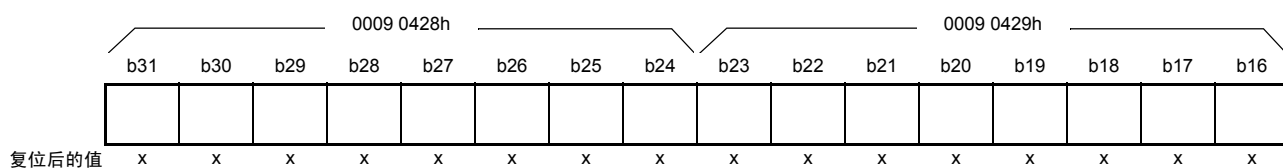
### IDE 位 (ID 扩展位)

IDE 位是设定被指定为标准 ID 或者扩展 ID 的 ID 格式。在 CTLR.IDFM[1:0] 位为“10” (混合 ID 模式) 时，IDE 位有效。

- 当 FIDCR0 寄存器和 FIDCR1 寄存器的 IDE 位都为“0”时，只能接收标准 ID 帧。
- 当 FIDCR0 寄存器和 FIDCR1 寄存器的 IDE 位都为“1”时，只能接收扩展 ID 帧。
- 当 FIDCR0 寄存器的 IDE 位为“0”而 FIDCR1 寄存器的 IDE 位为“1”时，或者 FIDCR0 寄存器的 IDE 位为“1”而 FIDCR1 寄存器的 IDE 位为“0”时，能接收标准 ID 帧和扩展 ID 帧。

## 25.2.5 屏蔽无效寄存器 (MKIVLR)

地址 0009 0428h



x: 不定值

位	符号	位名	功能	R/W
b31-b0	—	—	0: 屏蔽有效 1: 屏蔽无效	R/W

MKIVLR 寄存器包含各信箱的对应位。

各位的对应如下所示:

MKIVLR 寄存器的 bit0 对应信箱 0, bit31 对应信箱 31 (注 1)。

当该位为“1”时, 对于对应的信箱, 该验收屏蔽寄存器无效。如果将屏蔽无效位置“1”, 就只在接收信息的 ID 和信箱 ID 完全相同时对应的信箱接收信息。

必须在 CAN 复位模式或者 CAN Halt 模式中更改 MKIVLR 寄存器。

注 1. 在 FIFO 信箱模式中, 必须将 bit31 ~ bit24 置“0”。

### 25.2.6 信箱寄存器 j (MBj) (j=0 ~ 31)

CAN0 信箱的存储器分配和 CAN 数据帧的结构分别如表 25.5 和表 25.6 所示。

CAN0 信箱复位后的值为不定值。

只有在相关的 MCTLj 寄存器 (j=0 ~ 31) 为“00h”并且不在进行中止处理时, 才能更改 MBj 寄存器的设定。

有关寄存器地址的详细内容, 请参照表 25.5。

表 25.5 信箱的存储器分配

地址	寄存器符号	信息内容
CAN0	CAN0	存储器分配
0009 0200h+16×j+0	MB.ID	IDE、RTR、SID10 ~ SID6
0009 0200h+16×j+1		SID5 ~ SID0、EID17、EID16
0009 0200h+16×j+2		EID15 ~ EID8
0009 0200h+16×j+3		EID7 ~ EID0
0009 0200h+16×j+4	MB.DLC	—
0009 0200h+16×j+5		数据长度代码 (DLC[3:0])
0009 0200h+16×j+6	MB.DATA0	数据字节 0
0009 0200h+16×j+7		数据字节 1
:		:
0009 0200h+16×j+13		数据字节 7
0009 0200h+16×j+14	MB.TS	时戳高位字节
0009 0200h+16×j+15		时戳低位字节

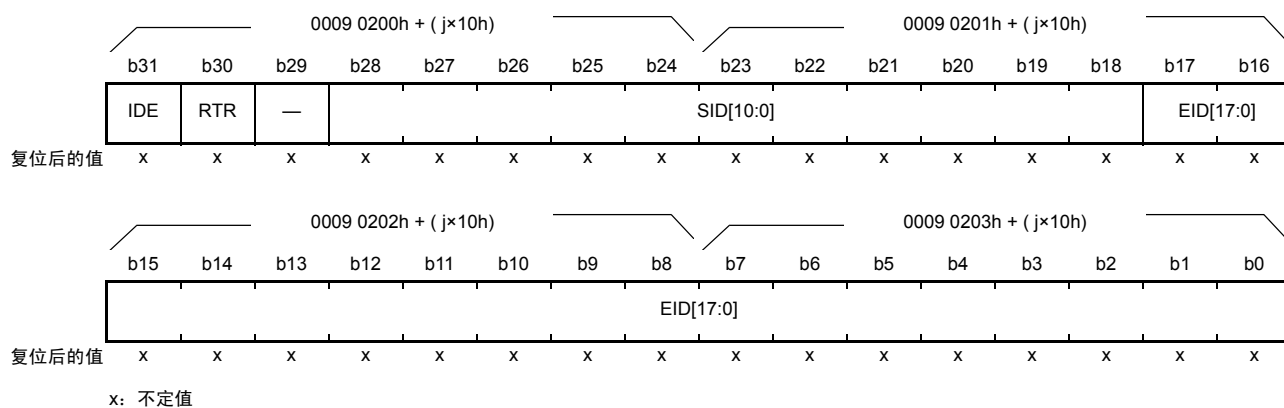
表 25.6 CAN 数据帧的结构

SID10 ~ SID6	SID5 ~ SID0	EID17 ~ EID16	EID15 ~ EID8	EID7 ~ EID0	DLC3 ~ DLC0	DATA0	DATA1	...	DATA7
--------------	-------------	---------------	--------------	-------------	-------------	-------	-------	-----	-------

只要不接收新信息, 各信箱的内容就保持以前的值。

(a) MB.ID

地址 0009 0200h ~ 0009 03FFh



位	符号	位名	功能	R/W
b17-b0	EID[17:0]	扩展 ID 位 (注 1)	0: 对应的 EID[17:0] 位为“0” 1: 对应的 EID[17:0] 位为“1”	R/W
b28-b18	SID[10:0]	标准 ID 位	0: 对应的 SID[10:0] 位为“0” 1: 对应的 SID[10:0] 位为“1”	R/W
b29	—	保留位	读取值为不定值, 只能写“0”。	R/W
b30	RTR	远程发送请求位	0: 数据帧 1: 远程帧	R/W
b31	IDE	ID 扩展位 (注 2)	0: 标准 ID 1: 扩展 ID	R/W

注 1. 如果信箱接收到标准 ID 的信息, 信箱 EID[17:0] 位的值就为不定值。

注 2. 在 CTLR.IDFM[1:0] 位为“10b” (混合 ID 模式) 时, IDE 位有效。在 CTLR.IDFM[1:0] 位不是“10b”时, 必须给 IDE 位写“0”。读取值为“0”。

#### EID[17:0] 位 (扩展 ID 位)

EID[17:0] 位设定数据帧和远程帧的扩展 ID。

在接收扩展 ID 的信息时使用这些位。

#### SID[10:0] 位 (标准 ID 位)

SID[10:0] 位设定数据帧和远程帧的标准 ID。

在接收标准 ID 和扩展 ID 的信息时使用这些位。

#### RTR 位 (远程发送请求位)

RTR 位设定被指定为数据帧或者远程帧的帧格式。

- 收件箱只接收由 RTR 位指定格式的帧。
- 发件箱根据 RTR 位指定的帧格式进行发送。
- 接收 FIFO 信箱接收由 FIDCR0 寄存器和 FIDCR1 寄存器的 RTR 位指定的数据帧和远程帧。
- 发送 FIFO 信箱发送由相关发送信息的 RTR 位选择的数据帧或者远程帧。

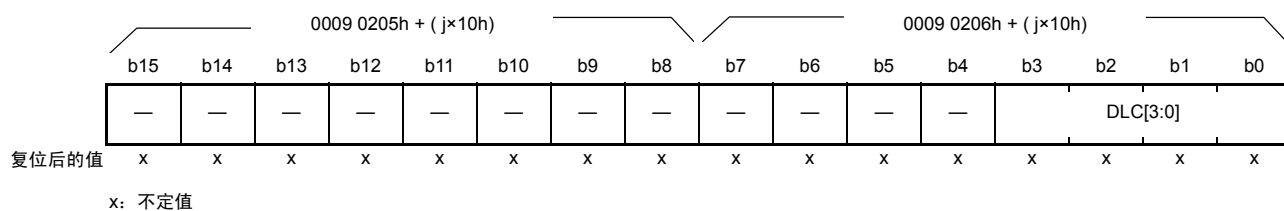
#### IDE 位 (ID 扩展位)

IDE 位设定被指定为标准 ID 或者扩展 ID 的 ID 格式。在 CTLR.IDFM[1:0] 位为“10b” (混合 ID 模式) 时, IDE 位有效。

- 收件箱只接收由 IDE 位指定的 ID 格式。
- 发件箱根据 IDE 位指定的 ID 格式进行发送。
- 接收 FIFO 信箱接收由 FIDCR0 寄存器和 FIDCR1 寄存器的 IDE 位指定的标准 ID 和扩展 ID。
- 发送 FIFO 信箱发送由相关发送信息的 IDE 位选择的标准 ID 或者扩展 ID 的信息。

## (b) MB.DLC

地址 0009 0205h ~ 0009 03F5h



位	符号	位名	功能	R/W
b3-b0	DLC[3:0]	数据长度代码位 (注1)	b3 b0 0 0 0 0: 数据长度为 0 字节 0 0 0 1: 数据长度为 1 字节 0 0 1 0: 数据长度为 2 字节 0 0 1 1: 数据长度为 3 字节 0 1 0 0: 数据长度为 4 字节 0 1 0 1: 数据长度为 5 字节 0 1 1 0: 数据长度为 6 字节 0 1 1 1: 数据长度为 7 字节 1 x x x: 数据长度为 8 字节 注. x: 任意值	R/W
b15-b4	—	保留位	读取值为不定值, 只能写“0”。	R/W

注 1. 如果信箱接收到小于 8 字节的 DLC[3:0] 的信息, 大于信箱 DLC[3:0] 的 DATA 就为不定值。

## DLC[3:0] 位 (数据长度代码位)

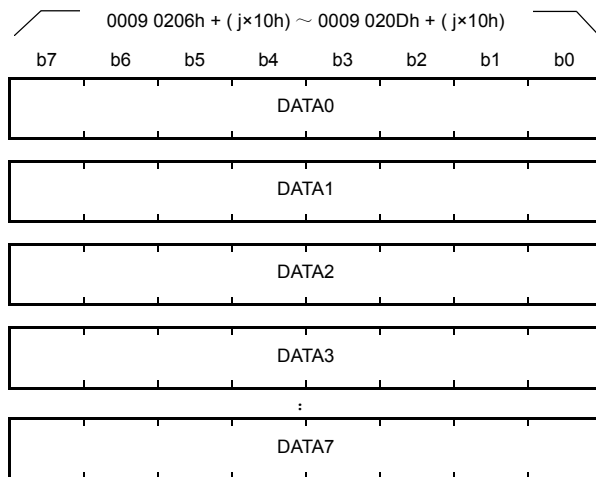
DLC[3:0] 位设定在数据帧中要发送的数据字节数。在使用远程帧请求数据时, 这些位设定要请求的数据字节数。

如果接收到数据帧, 就保存接收的数据字节数; 如果接收到远程帧, 就保存请求的数据字节数。



(c) MB.DATA0 ~ 7

地址 0009 0206h ~ 0009 03FDh



复位后的值 x x x x x x x x

x: 不定值

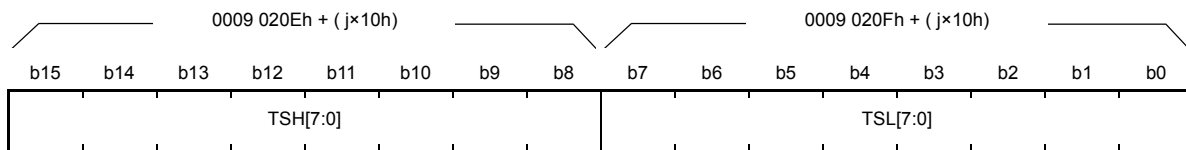
位	符号	位名	功能	R/W
b7-b0	DATA0- DATA7	数据长度代码位 (注1、注2)	DATA0 ~ 7 保存发送或者接收的 CAN 信息数据。从 DATA0 发送或者接收 CAN 信息数据。以 MSB first 方式从 bit7 开始发送或者接收 CAN 总线上的位序。	R/W

注 1. 如果信箱接收到小于 8 字节的 n 字节的信息, 信箱的 DATA0 ~ DATA7 的值就为不定值。

注 2. 如果信箱接收到远程帧, 信箱的 DATA0 ~ DATA7 就保持以前的值。

(d) MB.TS

地址 0009 020Eh ~ 0009 03FFh



复位后的值 x x x x x x x x x x x x x x x x

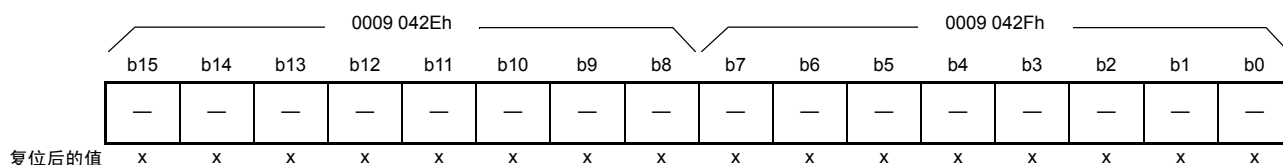
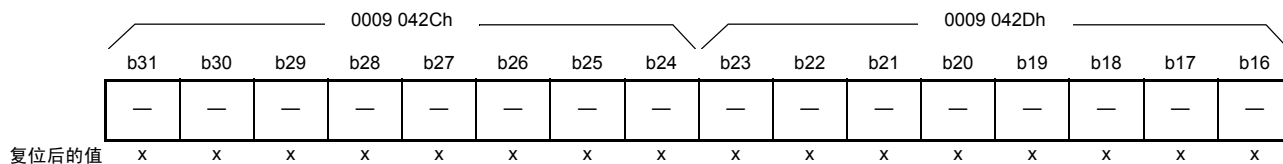
x: 不定值

位	符号	位名	功能	R/W
b7-b0	TSL[7:0]	时戳的低位字节	TSH[7:0] 和 TSL[7:0] 保存接收信息被取到信箱时的时戳计数器的值。	R/W
b15-b8	TSH[7:0]	时戳的高位字节		R/W

### 25.2.7 信箱中断允许寄存器 (MIER)

- 普通信箱模式

地址 0009 042Ch

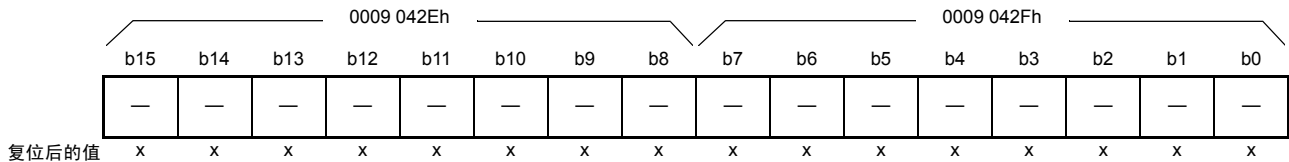
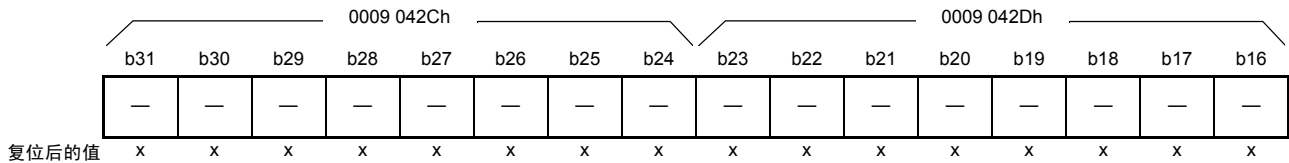


x: 不定值

位	符号	位名	功能	R/W
b31-b0	—	中断允许位	0: 禁止中断 1: 允许中断 bit31 对应信箱 31, bit0 对应信箱 0。	R/W

- FIFO 信箱模式

地址 0009 042Ch



x: 不定值

位	符号	位名	功能	R/W
b23-b0	—	中断允许位	0: 禁止中断 1: 允许中断 bit23 对应信箱 23, bit0 对应信箱 0。	R/W
b24	—	发送 FIFO 中断允许位	0: 禁止中断 1: 允许中断	R/W
b25	—	发送 FIFO 中断发生时序控制位	0: 在每次发送结束后, 产生发送 FIFO 中断请求。 1: 在因发送结束而使发送 FIFO 变为空状态时, 产生发送 FIFO 中断请求。	R/W
b27-b26	—	保留位	读取值为不定值, 只能写“0”。	R/W
b28	—	接收 FIFO 中断允许位	0: 禁止中断 1: 允许中断	R/W
b29	—	接收 FIFO 中断发生时序控制位 (注 1)	0: 在每次接收结束后, 产生接收 FIFO 中断请求。 1: 在因接收结束而使接收 FIFO 变为缓冲警告时, 产生接收 FIFO 中断请求。	R/W
b31-b30	—	保留位	读取值为不定值, 只能写“0”。	R/W

注 1. 如果接收 FIFO 从满变为缓冲警告, 就不产生中断请求。缓冲警告是指接收 FIFO 保存了第 3 条信息的状态。

MIER 寄存器能将各信箱分别设定为中断允许状态。

在普通信箱模式 (全部位) 和 FIFO 信箱模式 (MIER 寄存器的 bit23 ~ 0) 中, 各位分别对应各信箱。这些位允许或者禁止对应信箱的发送结束中断和接收结束中断。

- MIER 寄存器的 bit0 对应信箱 0。
- MIER 寄存器的 bit31 对应信箱 31。

FIFO 信箱模式的 MIER 寄存器的 bit29、28、25、24 允许或者禁止发送 / 接收 FIFO 中断以及指定中断请求的产生时序。

只有在相关的 MCTL<sub>j</sub> 寄存器 (j=0 ~ 31) 为“00h”并且对应的信箱还没有进行发送 / 接收中止处理时, 才能更改 MIER 寄存器。在 FIFO 运行模式中, 只有在 TFCR.TFE 位为“0”并且 TFCR.TFEST 位为“1”以及 RFCR.RFE 位为“0”并且 RFCR.RFEST 位为“1”时, 才能更改相关的 FIFO 的 MIER 寄存器位。

## 25.2.8 信息控制寄存器 j (MCTLj) (j=0 ~ 31)

地址 0009 0820h ~ 0009 083F

## • MCTL.TX

发送模式 (TRMREQ 位为“1”, RECREQ 位为“0”的情况)

b7	b6	b5	b4	b3	b2	b1	b0
TRMREQ	RECREQ	—	ONESHOT	—	TRMABT	TRMACTIVE	SENTDATA

复位后的值 0 0 0 0 0 0 0 0

## • MCTL.RX

接收模式 (TRMERQ 位为“0”, RECREQ 位为“1”的情况)

b7	b6	b5	b4	b3	b2	b1	b0
TRMREQ	RECREQ	—	ONESHOT	—	MSGLOST	NVALDATA	NEWDATA

复位后的值 0 0 0 0 0 0 0 0

位	符号	位名	功能	R/W
b0	SENTDATA	发送结束标志 (注1、注2)	0: 发送未结束 1: 发送结束	R/(W)
	NEWDATA	接收结束标志 (注1、注2)	0: 未接收到数据或者给 NEWDATA 位写“0” 1: 正在或者已经将新信息保存到信箱	R/(W)
b1	TRMACTIVE	发送过程中的状态位	(此位在设定为发件箱时有效。) 0: 正处于发送待机或者无发送请求 1: 从取发送请求到发送结束、发生错误或者仲裁失败为止	R
	INVALDATA	接收过程中的状态位	(此位在设定为收件箱时有效) 0: 信息有效 1: 正在更新信息	R
b2	TRMABT	发送中止结束标志 (注1、注2)	(此位在设定为发件箱时有效) 0: 开始发送, 或者发送中止因发送结束而失败, 或者未请求发送中止。 1: 发送中止结束	R/(W)
	MSGLOST	信息丢失标志 (注1、注2)	(此位在设定为收件箱时有效) 0: 未重写信息或者信息未溢出 1: 重写信息或者信息溢出	R/(W)
b3	—	保留位	读写值都为“0”。	R/W
b4	ONESHOT	单次允许位 (注3)	0: 禁止单次接收或者单次发送 1: 允许单次接收或者单次发送	R/W
b5	—	保留位	读写值都为“0”。	R/W
b6	RECREQ	收件箱设定位 (注2、注3、注4、注5)	0: 不设定为收件箱 1: 设定为收件箱	R/W
b7	TRMREQ	发件箱设定位 (注2、注4)	0: 不设定为发件箱 1: 设定为发件箱	R/W

注1. 只能写“0”(即使写“1”, 值也不变)。

注2. 当通过程序给 NEWDATA、SENTDATA、MSGLOST、TRMABT、RECREQ 和 TRMREQ 位写“0”时, 不能使用逻辑运算 (AND.B) 指令。必须使用传送 (MOV) 指令将想置“0”的位置“0”而将其他位置“1”。

注3. 要转移到单次接收模式时, 必须在将 RECREQ 位置“1”的同时给 ONESHOT 位写“1”; 要解除单次接收模式时, 必须给 RECREQ 位写“0”, 然后在确认 RECREQ 位为“0”后给 ONESHOT 位写“0”。要转移到单次发送模式时, 必须在将 TRMREQ 位置“1”的同时给 ONESHOT 位写“1”; 要解除单次发送模式时, 必须在发送或者中止信息后给 ONESHOT 位写“0”。

注4. 不能将 RECREQ 位和 TRMREQ 位都置“1”。

注5. 要将 RECREQ 位置“0”时, 必须同时将 NEWDATA、MSGLOST 标志和 RECREQ 位同时置“0”。

必须在 CAN 操作模式或者 CAN Halt 模式中更改 MCTLj 寄存器。  
在 FIFO 信箱模式中, 不能使用 MCTL24 ~ MCTL31 寄存器。

#### SENTDATA 标志 (发送结束标志)

当对应信箱的数据发送结束时, SENTDATA 标志变为“1”。如果通过程序给 SENTDATA 标志写“0”, 此标志就变为“0”。

要将 SENTDATA 标志置“0”时, 必须在将 TRMREQ 位置“0”后将 SENTDATA 标志置“0”。SENTDATA 标志和 TRMREQ 位不同时变为“0”。要从信箱发送新信息时, 必须将 SENTDATA 标志置“0”。

#### NEWDATA 标志 (接收结束标志)

当正在或者已经将新信息保存到信箱时, NEWDATA 标志变为“1”。NEWDATA 标志和 INVALIDDATA 位同时变为“1”。如果通过程序给 NEWDATA 标志写“0”, NEWDATA 标志就变为“0”。在相关的 INVALIDDATA 位为“1”期间, 即使通过程序给 NEWDATA 标志写“0”, 此标志也不会变为“0”。

#### TRMACTIVE 位 (发送过程中的状态位)

一旦从 CAN 模块开始发送对应信箱的信息, TRMACTIVE 位就变为“1”。如果 CAN 模块发生 CAN 总线仲裁失败、CAN 总线错误或者数据发送结束, TRMACTIVE 位就变为“0”。

#### INVALIDDATA 位 (接收过程中的状态位)

在接收完信息后或者正在更新信箱中的接收信息时, INVALIDDATA 位变为“1”。在保存完信息时, INVALIDDATA 位变为“0”。如果在 INVALIDDATA 位为“1”期间读信箱, 数据就为不定值。

#### TRMABT 标志 (发送中止结束标志)

在以下情况下, TRMABT 标志变为“1” (发送中止结束):

- 在发送中止请求之后并且在开始发送前, 发送中止结束。
- 在发送中止请求之后, 检测到 CAN 模块发生 CAN 总线仲裁失败或者 CAN 总线错误。
- 在单次发送模式 (RECREQ 位为“0”, TRMREQ 位为“1”, ONESHOT 位为“1”) 中, 检测到 CAN 模块发生 CAN 总线仲裁失败或者 CAN 总线错误。

即使数据发送结束, TRMABT 标志也不变为“1”。当数据发送结束时, SENTDATA 标志变为“1”。如果给 TRMABT 标志写“0”, 此标志就变为“0”。

#### MSGLOST 标志 (信息丢失标志)

在 NEWDATA 标志为“1”期间, 如果因新的接收信息而覆盖或者放弃信箱里的旧信息, MSGLOST 标志就变为“1”。在 EOF 的第 6 位结束时, 此标志为“1”。如果通过程序给 MSGLOST 标志写“0”, 此标志就变为“0”。

在重写模式和溢出模式中, 在 EOF 的第 6 位之后的 5 个外围模块时钟 (PCLK) 期间, 即使通过程序给 MSGLOST 标志写“0”, 此标志也不变为“0”。

### ONESHOT 位 (单次允许位)

ONESHOT 位有接收模式和发送模式两种使用方法。

- 单次接收模式

如果在接收模式 (RECREQ 位为“1”并且 TRMREQ 位为“0”) 中将 ONESHOT 位置“1”，信箱就只接收 1 次信息 (在接收 1 次信息后，不用作收件箱)。NEWDATA 标志和 INVALIDDATA 位的变化和普通接收模式相同。在此模式中，MSGLOST 标志不变为“1”。要将 ONESHOT 位置“0”时，必须给 RECREQ 位写“0”，在确认 RECREQ 位变为“0”后进行设定。

- 单次发送模式

如果在发送模式 (RECREQ 位为“0”并且 TRMREQ 位为“1”) 中将 ONESHOT 位置“1”，CAN 模块就只发送 1 次信息 (即使发生 CAN 总线错误或者 CAN 总线仲裁失败，也不重新发送信息)。当发送结束时，SENTDATA 标志变为“1”。在因 CAN 总线错误或者 CAN 总线仲裁失败而不结束发送时，TRMABT 标志为“1”。要将 ONESHOT 位置“0”时，必须在 SENTDATA 标志变为“1”或者 TRMABT 标志变为“1”后进行设定。

### RECREQ 位 (收件箱设定位)

RECREQ 位选择表 25.11 所示的接收模式。

如果将 RECREQ 位置“1”，就将对应的信箱设定为接收数据帧或者远程帧。

如果将 RECREQ 位置“0”，就不将对应的信箱设定为接收数据帧或者远程帧。

在以下期间内进行硬件保护，即使通过程序给 RECREQ 位写“0”，此位也不变为“0”。

- 硬件保护的开始

开始验收过滤处理 (CRC 字段的开始)。

- 硬件保护的解除

- 被指定为接收信息的信箱在将接收信息保存到信箱后或者发生 CAN 总线错误后 (即，硬件保护的  
最大期间是从 CRC 字段开始到 EOF 的第 7 位结束)

- 在其他信箱进行验收过滤处理后

- 在没有收件箱的情况下进行验收过滤处理后

在将 RECREQ 位置“1”时，不能将 TRMREQ 位置“1”。在将信箱的设定从发送改为接收时，必须在改为接收前先中止发送，然后将 SENTDATA 标志和 TRMABT 标志置“0”。

### TRMREQ 位 (发件箱设定位)

TRMREQ 位选择表 25.11 所示的发送模式。

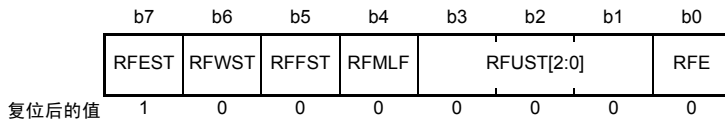
如果将 TRMREQ 位置“1”，就将对应的信箱设定为发送数据帧或者远程帧。

如果将 TRMREQ 位置“0”，就不将对应的信箱设定为发送数据帧或者远程帧。

如果为了取消发送请求而将 TRMREQ 位从“1”改为“0”，因为要确认发送中止请求是否结束，所以就在最多延迟 1 帧后 TRMABT 标志或者 SENTDATA 标志变为“1”。在将 TRMREQ 位置“1”时，不能将 RECREQ 位置“1”。要将信箱的设定从接收改为发送时，必须在改为发送前先中止接收，然后将 NEWDATA 标志和 MSGLOST 标志置“0”。

## 25.2.9 接收 FIFO 控制寄存器 (RFCR)

地址 0009 0848h



位	符号	位名	功能	R/W
b0	RFE	接收 FIFO 允许位	0: 禁止接收 FIFO 1: 允许接收 FIFO	R/W
b3-b1	RFUST[2:0]	接收 FIFO 未读信息数状态位	b3 b1 000: 没有未读信息 001: 有 1 条未读信息 010: 有 2 条未读信息 011: 有 3 条未读信息 100: 有 4 条未读信息 101: 保留 110: 保留 111: 保留	R
b4	RFMLF	接收 FIFO 信息丢失标志	0: 未发生接收 FIFO 的信息丢失 1: 发生接收 FIFO 的信息丢失	R/W
b5	RFFST	接收 FIFO 满状态位	0: 接收 FIFO 未滿 1: 接收 FIFO 满 (有 4 条未读信息)	R
b6	RFWST	接收 FIFO 缓冲警告状态标志	0: 接收 FIFO 不是缓冲警告状态 1: 接收 FIFO 是缓冲警告状态 (有 3 条未读信息)	R
b7	RFEST	接收 FIFO 空状态标志	0: 接收 FIFO 有未读信息 1: 接收 FIFO 没有未读信息	R

必须在 CAN 操作模式或者 CAN Halt 模式中更改 RFCR 寄存器。

## RFE 位 (接收 FIFO 允许位)

如果将 RFE 位置“1”，接收 FIFO 就为接收允许状态。

如果将 RFE 位置“0”，接收 FIFO 就为接收禁止状态或者空状态 (RFEST 位为“1”)。必须同时给 RFMLF 和 RFE 位写“0”。

在普通信箱模式 (CTRL 寄存器的 MBM 位为“0”)中，不能将 RFE 位置“1”。在以下期间内进行硬件保护，即使通过程序给 RFE 位写“0”，此位也不变为“0”。

- 硬件保护的开始
  - 开始接收过滤处理 (CRC 字段的开始)
- 硬件保护的解除
  - 在将接收 FIFO 指定为接收信息的情况下将接收信息保存到接收 FIFO 后或者发生 CAN 总线错误后 (即，硬件保护的期间是从 CRC 字段开始到 EOF 的第 7 位结束)
  - 在不指定接收 FIFO 的情况下进行接收过滤处理后

**RFUST[2:0] 位 (接收 FIFO 未读信息数状态位)**

RFUST[2:0] 位表示接收 FIFO 内未读信息的数量。

如果将 RFE 位置“0”，RFUST[2:0] 位的值就变为“000b”。

**RFMLF 标志 (接收 FIFO 信息丢失标志)**

如果在接收 FIFO 满时接收到新信息，RFMLF 位就变为“1” (发生接收 FIFO 的信息丢失)。在 EOF 的第 6 位结束时，此标志变为“1”。

如果通过程序给 RFMLF 位写“0”，此位就变为“0”。即使写“1”，值也不变。在溢出模式或者重写模式中，如果接收 FIFO 满并且已决定接收信息，就在 EOF 的第 6 位之后的 5 个外围模块时钟 (PCLK) 周期期间，因硬件保护而使 RFMLF 位不变为“0” (未发生接收 FIFO 的信息丢失)。

**RFFST 位 (接收 FIFO 满状态位)**

当接收 FIFO 内有 4 条未读信息时，RFFST 位为“1” (接收 FIFO 满状态)；当接收 FIFO 内不足 4 条未读信息，RFFST 位为“0” (接收 FIFO 未读)。如果将 RFE 位置“0”，RFFST 位就变为“0”。

**RFWST 标志 (接收 FIFO 缓冲警告状态标志)**

当接收 FIFO 内有 3 条未读信息时，RFWST 位为“1” (接收 FIFO 为缓冲警告状态)；当接收 FIFO 内的未读信息不足 3 条或者有 4 条，RFWST 位为“0” (接收 FIFO 不是缓冲警告状态)。如果将 RFE 位置“0”，RFWST 位就变为“0”。

**RFEST 标志 (接收 FIFO 空状态标志)**

当接收 FIFO 内没有未读信息时，RFEST 位为“1” (接收 FIFO 没有未读信息)。如果将 RFE 位置“0”，RFEST 位就变为“1”。当接收 FIFO 内有 1 条或者 1 条以上的未读信息时，RFEST 位为“0” (接收 FIFO 有未读信息)。

接收 FIFO 信箱的运行如图 25.2 所示。

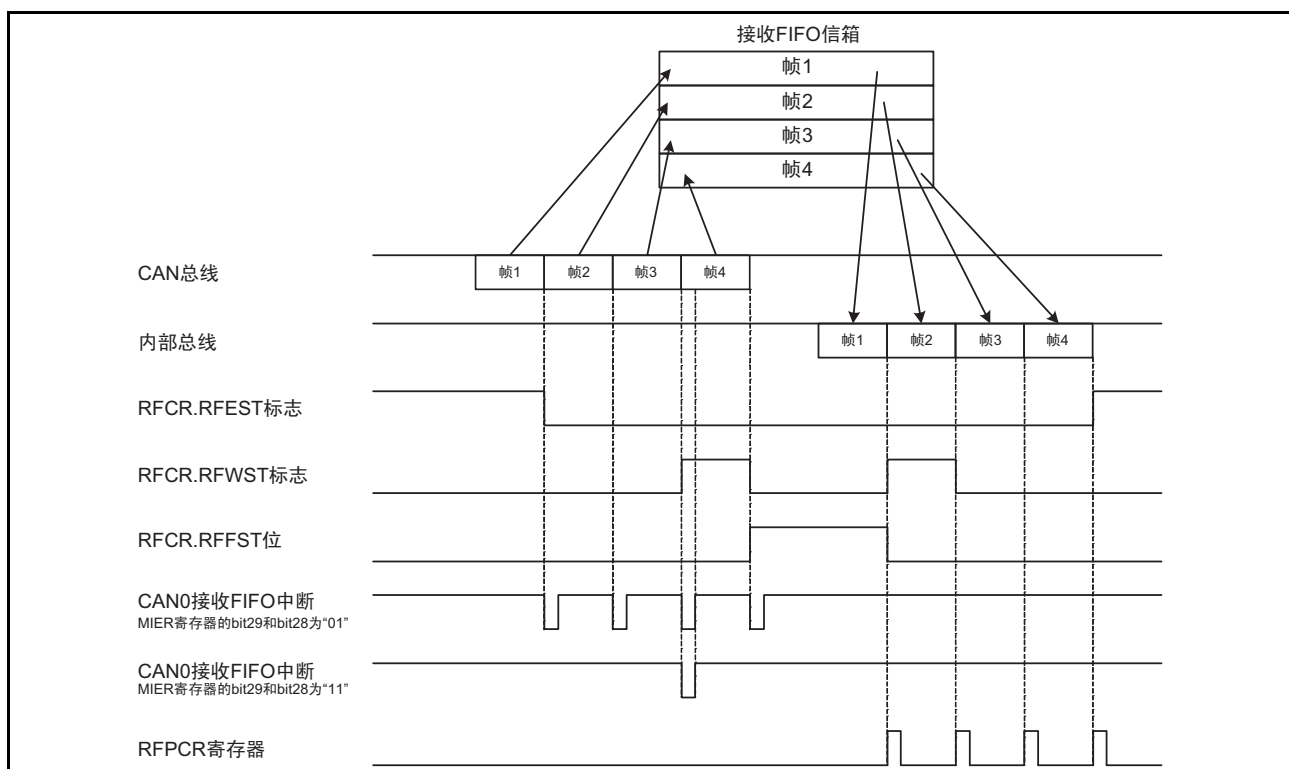


图 25.2 接收 FIFO 信箱的运行 (MIER 寄存器的 bit29 和 bit28 为“01”或者“11”)



## 25.2.10 接收 FIFO 指针控制寄存器 (RFPCR)

地址 0009 0849h

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

x: 不定值

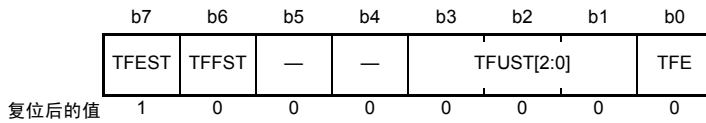
位	符号	位名	功能	R/W
b7-b0	—	—	如果给 RFPCR 寄存器写“FFh”，CPU 侧的接收 FIFO 指针就移动。	W

在接收 FIFO 不为空的状态下，要将 CPU 侧的接收 FIFO 指针移到下一个信箱时，必须通过程序给 RFPCR 寄存器写“FFh”。

在 RFCR.RFE 位为“0”（禁止接收 FIFO）时，不能写 RFPCR 寄存器。在接收重写模式中，如果在 RFCR.RFFST 位为“1”（接收 FIFO 满）时接收到新信息，CAN 侧的指针和 CPU 侧的指针都移动。在此状态下，当 RFCR.RFMLF 位为“1”时，即使通过程序写 RFPCR 寄存器，CPU 侧的指针也不移动。

## 25.2.11 发送 FIFO 控制寄存器 (TFCR)

地址 0009 084Ah



位	符号	位名	功能	R/W
b0	TFE	发送 FIFO 允许位	0: 禁止发送 FIFO 1: 允许发送 FIFO	R/W
b3-b1	TFUST[2:0]	发送 FIFO 未发送信息数状态位	b3 b1 0 0 0: 没有未发送信息 0 0 1: 有 1 条未发送信息 0 1 0: 有 2 条未发送信息 0 1 1: 有 3 条未发送信息 1 0 0: 有 4 条未发送信息 1 0 1: 保留 1 1 0: 保留 1 1 1: 保留	R
b5-b4	—	保留位	读写值都为“0”。	R/W
b6	TFFST	发送 FIFO 满状态位	0: 发送 FIFO 未满 1: 发送 FIFO 满 (有 4 条未发送信息)	R
b7	TFEST	接收 FIFO 空状态位	0: 发送 FIFO 有未发送信息 1: 发送 FIFO 没有未发送信息	R

必须在 CAN 操作模式或者 CAN Halt 模式中更改 TFCR 寄存器。

## TFE 位 (发送 FIFO 允许位)

如果将 TFE 位置“1”，发送 FIFO 就为发送允许状态。

如果将 TFE 位置“0”，发送 FIFO 就为空状态 (TFEST 位为“1”)，并且因以下情况而使发送 FIFO 的未发送信息丢失：

- 当发送 FIFO 没有下一个发送预定并且没有在发送时，直接进入空状态。
- 当发送 FIFO 有下一个发送预定或者已经在发送时，就在转移到发送结束、CAN 总线错误、仲裁失败或者 CAN Halt 模式后进入空状态。

在重新将 TFE 位置“1”前，必须确认 TFEST 位是否为“1”。必须在将 TFE 位置“1”后，将发送数据写到 MB24 寄存器。

在普通信箱模式 (CTRL.MBM 位为“0”)中，不能将 TFE 位置“1”。

## TFUST[2:0] 位 (发送 FIFO 未发送信息数状态位)

TFUST[2:0] 位表示发送 FIFO 内未发送信息的数量。

一旦发送 FIFO 的发送中止结束，TFUST[2:0] 位的值就变为“000b”。

**TFFST 位 (发送 FIFO 满状态位)**

当发送 FIFO 内有 4 条未发送信息的数量时, TFFST 位为“1”(发送 FIFO 满); 当发送 FIFO 内未发送信息数量不足 4 条时, TFFST 位为“0”(发送 FIFO 未滿)。一旦发送 FIFO 的发送中止结束, TFFST 位就变为“0”。

**TFEST 位 (接收 FIFO 空状态位)**

当发送 FIFO 内没有未发送信息时, TFEST 位为“1”(发送 FIFO 没有信息)。一旦发送 FIFO 的发送中止结束, TFEST 位就变为“1”。当发送 FIFO 内至少有 1 条未发送信息时, TFEST 位为“0”(发送 FIFO 有信息)。

发送 FIFO 信箱的运行如图 25.3 所示。

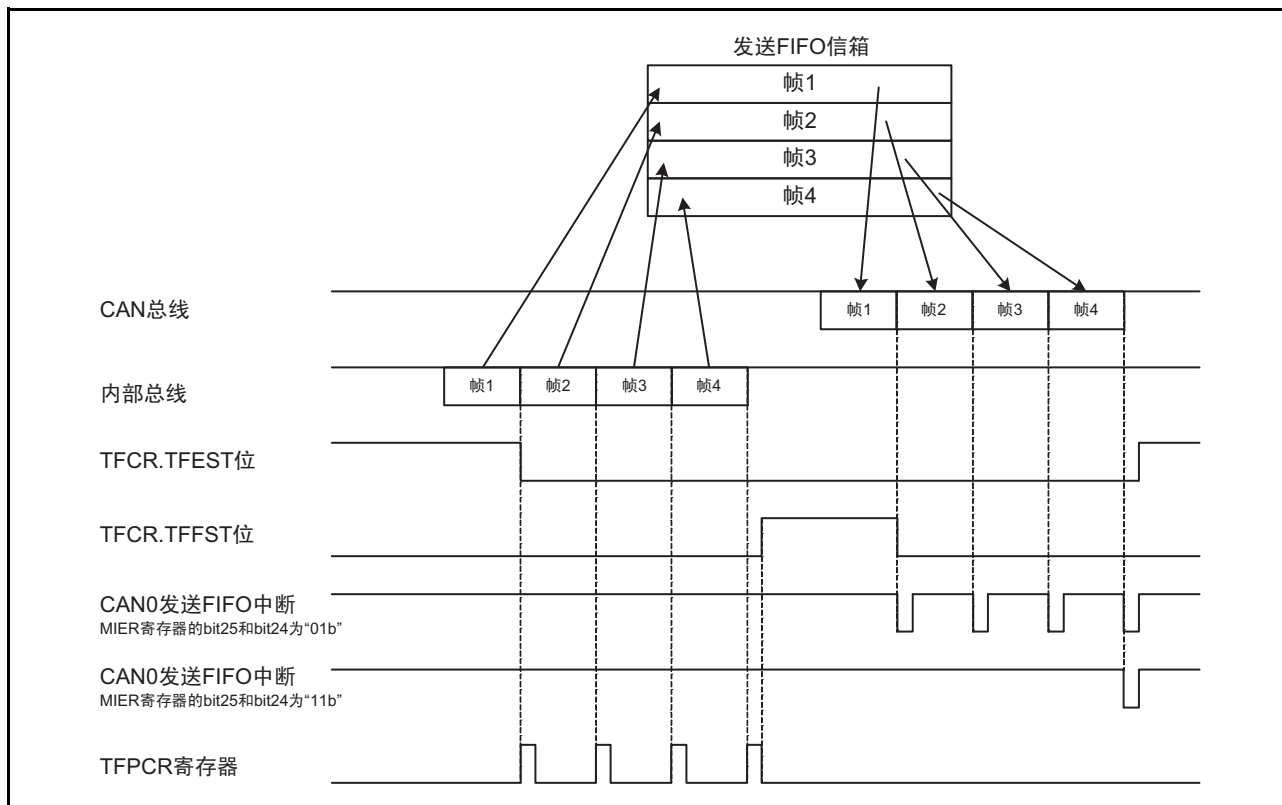
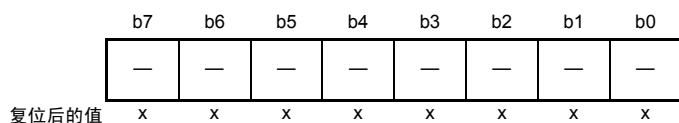


图 25.3 发送 FIFO 信箱的运行 (MIER 寄存器的 bit25 和 bit24 为“01b”或者“11b”)

## 25.2.12 发送 FIFO 指针控制寄存器 (TFPCR)

地址 0009 084Bh



复位后的值

x: 不定值

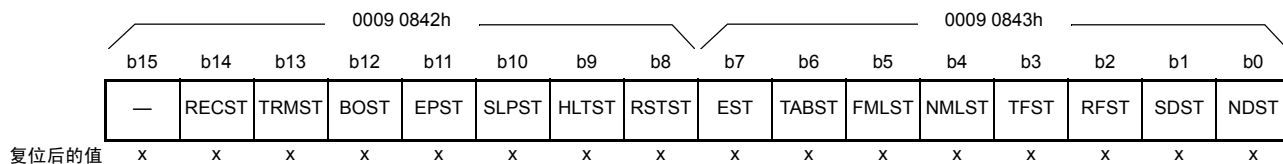
位	符号	位名	功能	R/W
b7-b0	—	—	如果给 TFPCR 寄存器写 “FFh”，CPU 侧的发送 FIFO 指针就移动。	W

在发送 FIFO 未满的情况下，要将 CPU 侧的发送 FIFO 指针移到下一个信箱时，必须通过程序给 TFPCR 寄存器写 “FFh”。

在 TFCR.TFE 位为 “0”（禁止发送 FIFO）时，不能写 TFPCR 寄存器。

## 25.2.13 状态寄存器 (STR)

地址 0009 0842h



x: 不定值

位	符号	位名	功能	R/W
b0	NDST	NEWDATA 状态标志	0: 没有 NEWDATA 标志为“1”的信箱 1: 有 NEWDATA 标志为“1”的信箱	R
b1	SDST	SENTDATA 状态标志	0: 没有 SENTDATA 标志为“1”的信箱 1: 有 SENTDATA 标志为“1”的信箱	R
b2	RFST	接收 FIFO 状态标志	0: 接收 FIFO 没有信息 (空) 1: 接收 FIFO 有信息	R
b3	TFST	发送 FIFO 状态标志	0: 发送 FIFO 满 1: 发送 FIFO 未滿	R
b4	NMLST	普通信息丢失状态标志	0: 没有 MSGLOST 标志为“1”的信箱 1: 有 MSGLOST 标志为“1”的信箱	R
b5	FMLST	FIFO 信息丢失状态标志	0: RFMLF 位为“0” 1: RFMLF 位为“1”	R
b6	TABST	发送中止状态标志	0: 没有 TRMABT 位为“1”的信箱 1: 有 TRMABT 位为“1”的信箱	R
b7	EST	错误状态标志	0: 未发生错误 1: 发生错误	R
b8	RSTST	CAN 复位状态标志	0: 不是 CAN 复位模式 1: CAN 复位模式	R
b9	HLTST	CAN Halt 状态标志	0: 不是 CAN Halt 模式 1: CAN Halt 模式	R
b10	SLPST	CAN 睡眠状态标志	0: 不是 CAN 睡眠模式 1: CAN 睡眠模式	R
b11	EPST	错误被动状态标志	0: 不是错误被动状态 1: 错误被动状态	R
b12	BOST	总线断开状态标志	0: 不是总线断开状态 1: 总线断开状态	R
b13	TRMST	发送状态标志 (transmitter)	0: 总线空闲或者正在接收 1: 正在发送或者总线断开状态	R
b14	RECST	接收状态标志 (receiver)	0: 总线空闲或者正在发送 1: 正在接收	R
b15	—	保留位	读取值为不定值, 写操作无效。	R

**NDST 标志 (NEWDATA 状态标志)**

只要 1 个 MCTLj.NEWDATA 标志 (j=0 ~ 31) 为“1”，NDST 标志就为“1”，与 MIER 寄存器的值无关。当 MCTLj.NEWDATA 标志全部为“0”时，NDST 标志为“0”。

**SDST 标志 (SENTDATA 状态标志)**

只要 1 个 MCTLj.SENTDATA 标志 (j=0 ~ 31) 为“1”，SDST 标志就为“1”，与 MIER 寄存器的值无关。当 MCTLj.SENTDATA 标志全部为“0”时，SDST 标志为“0”。

**RFST 标志 (接收 FIFO 状态标志)**

当接收 FIFO 不为空状态时，RFST 标志为“1”；当接收 FIFO 为空状态或者普通信箱模式时，此标志为“0”。

**TFST 标志 (发送 FIFO 状态位)**

当发送 FIFO 不是满状态时，TFST 标志为“1”；当发送 FIFO 是满状态或者普通信箱模式时，此标志为“0”。

**NMLST 标志 (普通信息丢失状态标志)**

只要 1 个 MCTLj.MSGLOST 标志 (j=0 ~ 31) 为“1”，NMLST 标志就为“1”，与 MIER 寄存器的值无关。当 MCTLj.MSGLOST 标志全部为“0”时，NMLST 标志为“0”。

**FMLST 标志 (FIFO 信息丢失状态标志)**

当 RFCR.RFMLF 位为“1”时，FMLST 标志为“1”，与 MIER 寄存器的值无关。当 RFCR.RFMLF 标志为“0”时，FMLST 标志为“0”。

**TABST 标志 (发送中止状态标志)**

只要 1 个 MCTLj.TRMABT 标志 (j=0 ~ 31) 为“1”，TABST 位就为“1”，与 MCTLj.MIER 寄存器的值无关。当 MCTLj.TRMABT 标志都不为“1”时，TABST 标志为“0”。

**EST 标志 (错误状态标志)**

只要通过 EIFR 寄存器检测到一个错误，EST 标志就为“1”，与 EIER 寄存器的值无关。如果通过 EIFR 寄存器一个错误也没有检测到，EST 标志就为“0”。

**RSTST 标志 (CAN 复位状态标志)**

当进入 CAN 复位模式时，RSTST 标志为“1”；当不进入 CAN 复位模式时，此标志为“0”。即使从 CAN 复位模式转移到 CAN 睡眠模式，RSTST 标志也保持“1”。

**HLTST 标志 (CAN Halt 状态标志)**

当进入 CAN Halt 模式时，HLTST 标志为“1”；当不进入 CAN Halt 模式时，此标志为“0”。即使从 CAN Halt 模式转移到 CAN 睡眠模式，HLTST 标志也保持“1”。

**SLPST 标志 (CAN 睡眠状态标志)**

当进入 CAN 睡眠模式，SLPST 标志为“1”。当不进入 CAN 睡眠模式时，此标志为“0”。

**EPST 标志 (错误被动状态标志)**

当 **TECR** 寄存器或者 **RECR** 寄存器的值超过 127 并且 CAN 模块为错误被动状态 ( $128 \leq \text{TEC} < 256$  或者  $128 \leq \text{REC} < 256$ ) 时, **EPST** 标志为“1”; 当不是错误被动状态时, **EPST** 标志为“0”。

**BOST 标志 (总线断开状态标志)**

当 **TECR** 寄存器的值超过 255 并且 CAN 模块为总线断开状态 ( $\text{TEC} \geq 256$ ) 时, **BOST** 标志为“1”; 当不是总线断开状态, **BOST** 标志为“0”。

**TRMST 标志 (发送状态标志) (transmitter)**

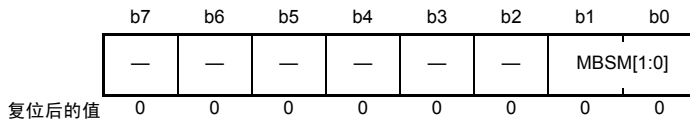
当 CAN 模块为发送节点或者总线断开状态时, **TRMST** 标志为“1”; 当 CAN 模块为接收节点或者总线空闲状态时, **TRMST** 标志为“0”。

**RECST 标志 (接收状态标志) (receiver)**

当 CAN 模块为接收节点时, **RECST** 标志为“1”; 当 CAN 模块为发送节点或者总线空闲状态时, **RECST** 标志为“0”。

## 25.2.14 信箱搜索模式寄存器 (MSMR)

地址 0009 0853h



位	符号	位名	功能	R/W
b1-b0	MBSM[1:0]	信箱搜索模式选择位	b1 b0 0 0: 收件箱搜索模式 0 1: 发件箱搜索模式 1 0: 信息丢失搜索模式 1 1: 通道搜索模式	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

必须在 CAN 操作模式或者 CAN Halt 模式中更改 MSMR 寄存器。

## MBSM[1:0] 位 (信箱搜索模式选择位)

MBSM[1:0] 位选择信箱搜索功能的搜索模式。

当 MBSM[1:0] 位为“00b”时，为收件箱搜索模式。在此模式中，搜索对象位是 MCTLj 寄存器 (j=0 ~ 31) 的普通信箱中的 MCTLj.NEWDATA 标志和 RFCR.RFEST 位。

当 MBSM[1:0] 位为“01b”时，为发件箱搜索模式。在此模式中，搜索对象位是 MCTLj.SENTDATA 标志。

当 MBSM[1:0] 位为“10b”时，为信息丢失搜索模式。在此模式中，搜索对象位是 MCTLj 寄存器的普通信箱中的 MCTLj.MSGLOST 标志和 RFCR.RFMLF 标志。

当 MBSM[1:0] 位为“11b”时，为通道搜索模式。在此模式中，搜索对象寄存器是 CSSR 寄存器，请参照“25.2.16 通道搜索支持寄存器 (CSSR)”。



## 25.2.15 信箱搜索状态寄存器 (MSSR)

地址 0009 0852h



位	符号	位名	功能	R/W
b4-b0	MBNST[4:0]	搜索结果信箱号状态位	表示在 MSMR 寄存器的各模式中搜索到的最小信箱号。	R
b6-b5	—	保留位	读写值都为“0”。	R/W
b7	SEST	搜索结果空状态位	0: 有搜索结果 1: 没有搜索结果	R

## MBNST[4:0] 位 (搜索结果信箱号状态位)

MBNST[4:0] 位表示在 MSMR 寄存器的各模式中搜索到的最小信箱号。在以下情况下，通过收件箱搜索模式、发件箱搜索模式和信息丢失搜索模式，更新被输出的搜索结果信箱的值：

- 被输出信箱的 MCTj.NEWDATA、MCTj.SENTDATA 或者 MCTj.MSGLOST 标志为“0”。
- 更高优先级信箱的 MCTj.NEWDATA、MCTj.SENTDATA 或者 MCTj.MSGLOST 标志为“1”。

在 MSMR.MBSM[1:0] 位为“00b”（收件箱搜索模式）或者“10b”（信息丢失搜索模式）时，如果接收 FIFO 不是空状态并且全部普通信箱（信箱 [0] ~ [23]）既没有未读接收信息也没有丢失信息，就输出接收 FIFO（信箱 [28]）。在 MSMR.MBSM[1:0] 位为“01b”（发件箱搜索模式）时，不输出发送 FIFO（信箱 [24]）。FIFO 信箱模式中的 MBNST[4:0] 位的变化如表 25.7 所示。

在通道搜索模式中，MBNST[4:0] 位输出信箱号。在通过程序读 MSSR 寄存器后，MBNST[4:0] 位输出下一个目标通道号。

## SEST 位 (搜索结果空状态位)

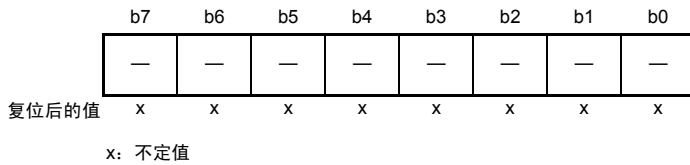
如果在全部信箱搜索中没有对应的信箱，SEST 位就为“1”（没有搜索结果）。例如，如果在发件箱搜索模式中 MCTj.SENTDATA 标志为“1”的信箱一个也没有，SEST 位就为“1”。只要有一个 MCTj.SENTDATA 标志为“1”的信箱，此位就为“0”。在 SEST 位为“1”时，MBNST[4:0] 位的值为不定值。

表 25.7 FIFO 信箱模式中 MBNST[4:0] 位的变化

MSMR.MBSM[1:0] 位	信箱 [24] (发送 FIFO)	信箱 [28] (接收 FIFO)
00b	不输出信箱 [24]。	如果普通信箱的没有一个 MCTj.NEWDATA 标志为“1”（正在或者已经将新信息保存到信箱）并且接收 FIFO 不为空，就输出信箱 [28]。 (j=0 ~ 23)
01b		不输出信箱 [28]。
10b		如果普通信箱的没有一个 MCTj.MSGLOST 标志为“1”（重写信息或者信息溢出）而且接收 FIFO 内的 RFCR.RFMLF 标志为“1”（发生接收 FIFO 信息丢失），就输出信箱 [28]。(j=0 ~ 23)
11b		不输出信箱 [28]。

### 25.2.16 通道搜索支持寄存器 (CSSR)

地址 0009 0851h



位	符号	位名	功能	R/W
b7-b0	—	—	如果输入通道搜索的值，就将通道号输出到 MSSR 寄存器。	R/W

通过 8/3 编码器 (最低位的优先级最高) 将为“1”的 CSSR 寄存器的位进行编码并且输出到 MSSR.MBNST[4:0] 位。

在每次通过程序读 MSSR 寄存器时，MSSR 寄存器表示更新后的值。

只能在 MSMR.MBSM[1:0] 位为“11b” (通道搜索模式) 时更改 CSSR 寄存器。不能在 CAN 复位模式中设定 CSSR 寄存器。

CSSR 寄存器和 MSSR 寄存器的读写操作如图 25.4 所示。

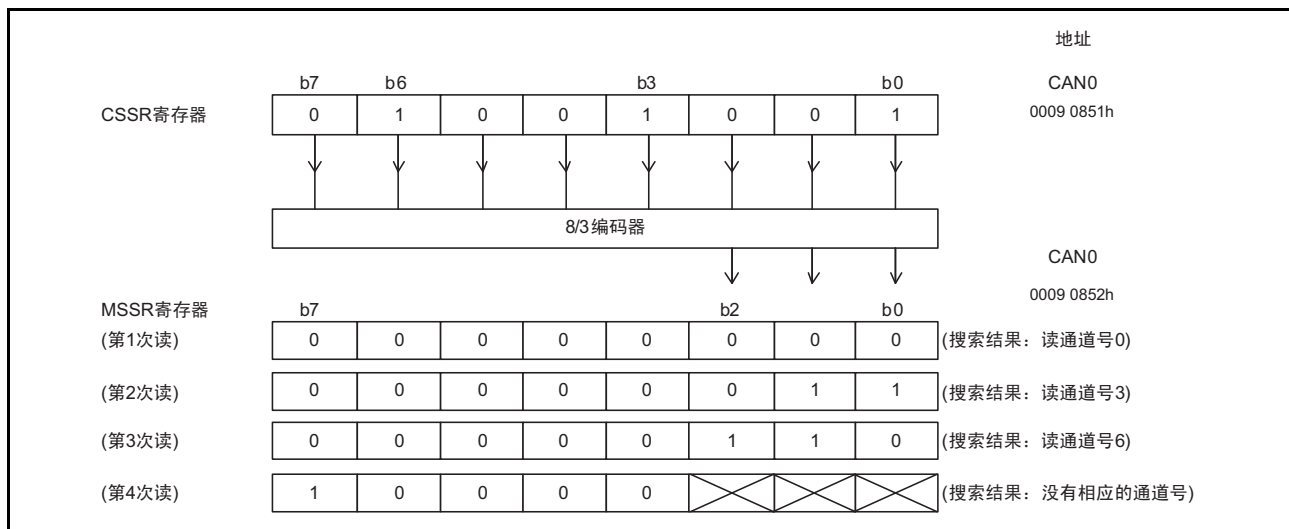
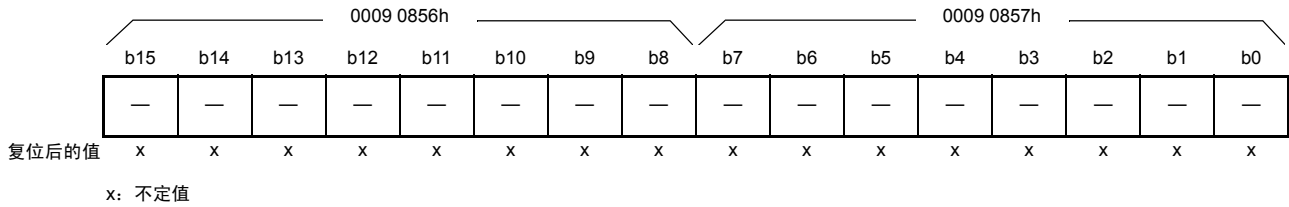


图 25.4 CSSR 寄存器和 MSSR 寄存器的读写操作

在每次读 MSSR 寄存器时，也更新 CSSR 寄存器的值。在进行读操作时，能读到 8/3 编码器转换前的值。

### 25.2.17 验收滤波器支持寄存器 (AFSR)

地址 0009 0856h



位	符号	位名	功能	R/W
b15-b0	—	—	在写接收信息的标准 ID 后, 能读为数据表搜索而转换的值。	R/W

验收滤波器支持单元 (ASU) 能用于数据表 (8 位 × 256) 的搜索, 此数据表是以 1 位为单位对用户预先建立的全部标准 ID 设定了有效或者无效的数据表。如果将包含保存接收标准 ID 的 MBj.SID 位 (j=0 ~ 31) 在内的 16 位数据写到 AFSR 寄存器, 就能读到被解码后的用于数据表搜索的行 (字节偏移) 位置和列 (位) 位置。ASU 只能用于标准 (11 位) ID。

ASU 在以下情况下有效:

- 不能通过验收滤波器屏蔽接收的 ID 的情况  
(例) 接收的 ID: 078h、087h、111h
  - 因接收的 ID 过多而想通过软件缩短过滤处理时间的情况
- 不能在 CAN 复位模式中设定 AFSR 寄存器。

AFSR 寄存器的读写操作如图 25.5 所示。

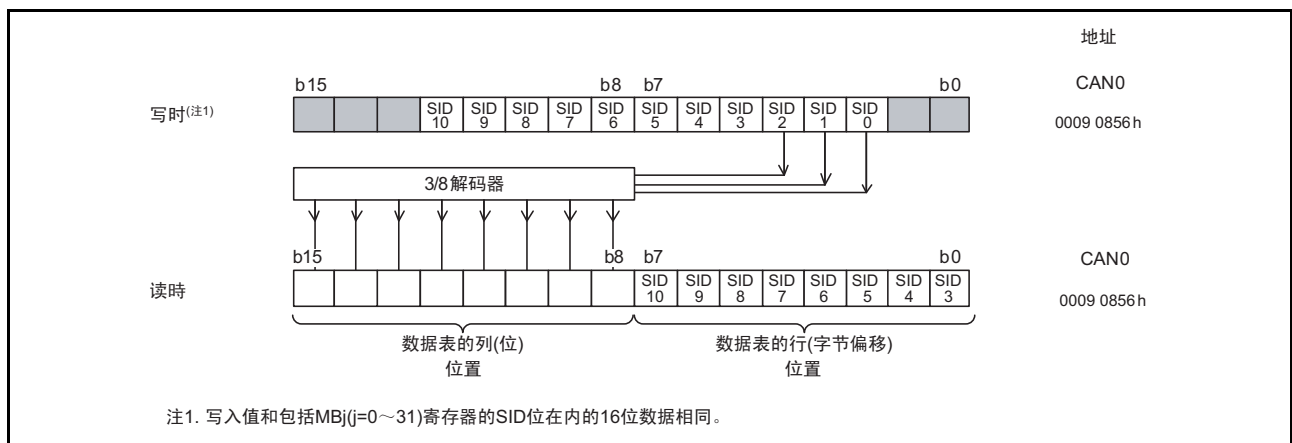


图 25.5 AFSR 寄存器的读写操作

## 25.2.18 错误中断允许寄存器 (EIER)

地址 0009 084Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	BLIE	OLIE	ORIE	BORIE	BOEIE	EPIE	EWIE	BEIE
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	BEIE	总线错误中断允许位	0: 禁止总线错误中断 1: 允许总线错误中断	R/W
b1	EWIE	错误警告中断允许位	0: 禁止错误警告中断 1: 允许错误警告中断	R/W
b2	EPIE	错误被动入口中断允许位	0: 禁止错误被动中断 1: 允许错误被动中断	R/W
b3	BOEIE	总线断开开始中断允许位	0: 禁止总线断开开始中断 1: 允许总线断开开始中断	R/W
b4	BORIE	总线断开恢复中断允许位	0: 禁止总线断开恢复中断 1: 允许总线断开恢复中断	R/W
b5	ORIE	溢出中断允许位	0: 禁止接收溢出中断 1: 允许接收溢出中断	R/W
b6	OLIE	超载帧发送中断允许位	0: 禁止超载帧发送中断 1: 允许超载帧发送中断	R/W
b7	BLIE	总线锁定中断允许位	0: 禁止总线锁定中断 1: 允许总线锁定中断	R/W

EIER 寄存器允许或者禁止 EIFR 寄存器的各错误中断源对应的各错误中断。  
只能在 CAN 复位模式中更改 EIER 寄存器。

**BEIE 位 (总线错误中断允许位)**

如果将 BEIE 位置“0”，即使 EIFR.BEIF 标志为“1”，也不产生错误中断请求。如果将 BEIE 位置“1”，就在 EIFR.BEIF 标志变为“1”时产生错误中断请求。

**EWIE 位 (错误警告中断允许位)**

如果将 EWIE 位置“0”，即使 EIFR.EWIF 标志为“1”，也不产生错误中断请求。如果将 EWIE 位置“1”，就在 EIFR.EWIF 标志变为“1”时产生错误中断请求。

**EPIE 位 (错误被动入口中断允许位)**

如果将 EPIE 位置“0”，即使 EIFR.EPIF 标志为“1”，也不产生错误中断请求。如果将 EPIE 位置“1”，就在 EIFR.EPIF 标志变为“1”时产生错误中断请求。

**BOEIE 位 (总线断开开始中断允许位)**

如果将 BOEIE 位置“0”，即使 EIFR.BOEIF 标志为“1”，也不产生错误中断请求。如果将 BOEIE 位置“1”，就在 EIFR.BOEIF 标志变为“1”时产生错误中断请求。

**BORIE 位 (总线断开恢复中断允许位)**

如果将 BORIE 位置“0”，即使 EIFR.BORIF 标志为“1”，也不产生错误中断请求。如果将 BORIE 位置“1”，就在 EIFR.BORIF 标志变为“1”时产生错误中断请求。

**ORIE 位 (溢出中断允许位)**

如果将 ORIE 位置“0”，即使 EIFR.ORIF 标志为“1”，也不产生错误中断请求。如果将 ORIE 位置“1”，就在 EIFR.ORIF 标志变为“1”时产生错误中断请求。

**OLIE 位 (超载帧发送中断允许位)**

如果将 OLIE 位置“0”，即使 EIFR.OLIF 标志为“1”，也不产生错误中断请求。如果将 OLIE 位置“1”，就在 EIFR.OLIF 标志的设定条件变为“1”时产生错误中断请求。

**BLIE 位 (总线锁定中断允许位)**

如果将 BLIE 位置“0”，即使 EIFR.BLIF 标志为“1”，也不产生错误中断请求。如果将 BLIE 位置“1”，就在 EIFR.BLIF 标志变为“1”时产生错误中断请求。

## 25.2.19 错误中断源判断寄存器 (EIFR)

地址 0009 084Dh

b7	b6	b5	b4	b3	b2	b1	b0
BLIF	OLIF	ORIF	BORIF	BOEIF	EPIF	EWIF	BEIF
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	BEIF	总线错误检测标志	0: 未检测到总线错误 1: 检测到总线错误	R/W
b1	EWIF	错误警告检测标志	0: 未检测到错误警告 1: 检测到错误警告	R/W
b2	EPIF	错误被动检测标志	0: 未检测到错误被动 1: 检测到错误被动	R/W
b3	BOEIF	总线断开开始检测标志	0: 未检测到总线断开开始 1: 检测到总线断开开始	R/W
b4	BORIF	总线断开恢复检测标志	0: 未检测到总线断开恢复 1: 检测到总线断开恢复	R/W
b5	ORIF	接收溢出检测标志	0: 未检测到接收溢出 1: 检测到接收溢出	R/W
b6	OLIF	超载帧发送检测标志	0: 未检测到超载帧发送 1: 检测到超载帧发送	R/W
b7	BLIF	总线锁定检测标志	0: 未检测到总线锁定 1: 检测到总线锁定	R/W

如果发生对应各标志的现象，EIFR 寄存器的对应标志就变为“1”，与 EIER 寄存器的设定无关。

要将各标志置“0”时，必须写“0”。在通过程序写“0”的同时，对应的标志变为“1”。

在将各标志置“0”时，不能使用逻辑运算 (AND.B) 指令而必须使用传送 (MOV) 指令，并且给对应的标志写“0”而给其他标志写“1”。即使写“1”，这些标志的值也不变。

## BEIF 标志 (总线错误检测标志)

如果检测到总线错误，BEIF 标志就为“1”。

## EWIF 标志 (错误警告检测标志)

如果 REC 或者 TEC 的值超过 95，EWIF 标志就为“1”。

只有在 REC 或者 TEC 首次超过 95 时，EWIF 标志才为“1”。在 REC 或者 TEC 超过 95 的情况下，如果通过程序给 EWIF 标志写“0”，REC 或者 TEC 就暂时小于等于 95，此标志在超过 95 前不变为“1”。

## EPIF 标志 (错误被动检测标志)

如果 CAN 错误状态为错误被动状态 (REC (接收错误计数器) 或者 TEC 的值超过 127)，EPIF 标志就为“1”。

只有在 REC 或者 TEC 首次超过 127 时，EPIF 标志才为“1”。在 REC 或者 TEC 超过 127 的情况下，如果通过程序给 EPIF 标志写“0”，REC 或者 TEC 就暂时小于等于 127，此标志在 REC 或者 TEC 再次超过 127 前不变为“1”。

**BOEIF 标志 (总线断开开始检测标志)**

如果 CAN 错误状态为总线断开状态 (TEC (发送错误计数器) 的值超过 255), BOEIF 标志就为“1”。即使在 CTLR.BOM[1:0] 位为“01b” (在总线断开开始时自动转移到 CAN Halt 模式) 并且 CAN 模块为总线断开状态时, BOEIF 标志也为“1”。

**BORIF 标志 (总线断开恢复检测标志)**

如果 CAN 模块在以下条件下从总线断开状态正常恢复 (对 11 个连续的隐性位进行了 128 次的检测), BORIF 标志就为“1”:

- CTLR.BOM[1:0] 位为“00b”。
- CTLR.BOM[1:0] 位为“10b”。
- CTLR.BOM[1:0] 位为“11b”。

如果 CAN 模块在以下条件下从总线断开状态恢复时, BORIF 标志就不变为“1”:

- 将 CTLR.CANM[1:0] 位置“01b”或者“11b” (CAN 复位模式) 时
- 将 CTLR.RBOC 位置“1” (强制从总线断开状态恢复) 时
- CTLR.BOM[1:0] 位为“01b”时
- CTLR.BOM[1:0] 位为“11b”并且在正常恢复前将 CTLR.CANM[1:0] 位置“10b” (CAN Halt 模式) 时

**ORIF 标志 (接收溢出检测标志)**

如果发生接收溢出, ORIF 标志就为“1”。在重写模式中, ORIF 标志不为“1”。

如果在重写模式中产生重写条件, 就产生接收结束中断请求并且 ORIF 标志不变为“1”。

在普通信箱模式中, 如果在溢出模式中信箱 [0] ~ [31] 的任意一个信箱发生溢出, ORIF 标志就为“1”。在 FIFO 信箱模式中, 如果在溢出模式中信箱 [0] ~ [23] 的任意一个信箱或者接收 FIFO 发生溢出, ORIF 标志就为“1”。

**OLIF 标志 (超载帧发送检测标志)**

如果在 CAN 模块进行接收或者发送时检测到超载帧的发送条件, OLIF 标志就为“1”。

**BLIF 标志 (总线锁定检测标志)**

如果在 CAN 模块为 CAN 操作模式的期间检测到 CAN 总线上有 32 个连续的显性位, BLIF 位就为“1”。

在此标志变为“1”后, 在以下任意一种情况下重新进行检测:

- 在将此位从“1”改为“0”后, 检测到隐性位。
- 在将此位从“1”改为“0”后, 转移到 CAN 复位模式或者 CAN Halt 模式, 并且再次转移到 CAN 操作模式。

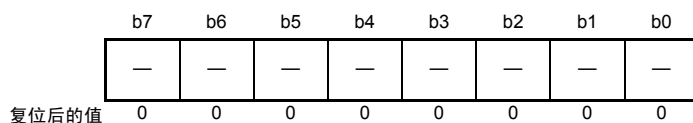
通过设定 CTLR.BOM[1:0] 位引起的 BOEIF 标志和 BORIF 标志的变化如表 25.8 所示。

表 25.8 通过设定 CTLR.BOM[1:0] 位引起的 BOEIF 标志和 BORIF 标志的变化

CTLR.BOM[1:0] 位	BOEIF 标志	BORIF 标志
00b	在转移到总线断开状态移时, 此标志为“1”。	在从总线断开状态恢复时, 此标志为“1”。
01b		此标志不变为“1”。
10b		在从总线断开状态恢复时, 此标志为“1”。
11b		如果在 CTLR.CANM[1:0] 位变为“10b” (CAN Halt 模式) 前从通常的总线断开状态恢复, 此标志就为“1”。

## 25.2.20 接收错误计数寄存器 (RECR)

地址 0009 084Eh



位	符号	位名	功能	R/W
b7-b0	—	接收错误计数功能	根据正在接收的 CAN 模块的错误状态, 增加或者减小计数器的值。	R

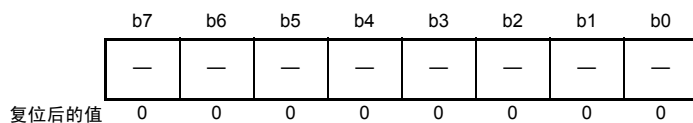
RECR 寄存器是表示接收错误计数器值的寄存器。

有关接收错误计数器的增减条件, 请参照 CAN 规格 (ISO11898-1)。

在总线断开状态下, RECR 寄存器的值为不定值。

## 25.2.21 发送错误计数寄存器 (TECR)

地址 0009 084Fh



位	符号	位名	功能	R/W
b7-b0	—	发送错误计数功能	根据正在发送的 CAN 模块的错误状态, 增加或者减小计数器的值。	R

TECR 寄存器是表示发送错误计数器值的寄存器。

有关发送错误计数器的增减条件, 请参照 CAN 规格 (ISO11898-1)。

在总线断开状态下, TECR 寄存器的值为不定值。



## 25.2.22 错误代码保存寄存器 (ECSR)

地址 0000 0850h

b7	b6	b5	b4	b3	b2	b1	b0
EDPM	ADEF	BE0F	BE1F	CEF	AEF	FEF	SEF
复位后的值	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	SEF	填充错误标志 (注1、注2)	0: 未检测到填充错误 1: 检测到填充错误	R/W
b1	FEF	格式错误标志 (注1、注2)	0: 未检测到格式错误 1: 检测到格式错误	R/W
b2	AEF	ACK 错误标志 (注1、注2)	0: 未检测到 ACK 错误 1: 检测到 ACK 错误	R/W
b3	CEF	CRC 错误标志 (注1、注2)	0: 未检测到 CRC 错误 1: 检测到 CRC 错误	R/W
b4	BE1F	位错误 (隐性) 标志 (注1、注2)	0: 未检测到位错误 1: 检测到位错误 (隐性)	R/W
b5	BE0F	位错误 (显性) 标志 (注1、注2)	0: 未检测到位错误 1: 检测到位错误 (显性)	R/W
b6	ADEF	ACK 定界符错误标志 (注1、注2)	0: 未检测到 ACK 定界符错误 1: 检测到 ACK 定界符错误	R/W
b7	EDPM	错误表示模式选择位 (注3、注4)	0: 输出最初检测到的错误代码 1: 输出累积的错误代码	R/W

注 1. 即使写“1”，这些位的值也不变。

注 2. 在给 SEF、FEF、AEF、CEF、BE1F、BE0F 或者 ADEF 标志写“0”时，不能使用逻辑运算 (AND.B) 指令而必须使用传送 (MOV) 指令，并且必须给对应的标志写“0”而给其他标志写“1”。

注 3. 必须在 CAN 复位模式或者 CAN Halt 模式中更改 EDPM 位。

注 4. 当同时检测到 1 个或者 1 个以上的错误条件时，相关的全部标志都变为“1”。

ECSR 寄存器监视 CAN 总线上的错误发生。

要确认各错误的发生条件时，请参照 CAN 规格 (ISO11898-1)。

要将除 EDPM 位以外的各标志置“0”时，必须通过程序给这些标志写“0”。如果各标志变为“1”的时序和写“0”的时序相同，该对应标志就变为“1”。

## SEF 标志 (填充错误标志)

如果检测到填充错误，SEF 标志就为“1”。

## FEF 标志 (格式错误标志)

如果检测到格式错误，FEF 标志就为“1”。

## AEF 标志 (ACK 错误标志)

如果检测到 ACK 错误，AEF 标志就为“1”。

**CEF 标志 (CRC 错误标志)**

如果检测到 CRC 错误, CEF 标志就为“1”。

**BE1F 标志 (位错误 (隐性) 标志)**

如果检测到隐性位错误, BE1F 标志就为“1”。

**BE0F 标志 (位错误 (显性) 标志)**

如果检测到显性位错误, BE0F 标志就为“1”。

**ADEF 标志 (ACK 定界符错误标志)**

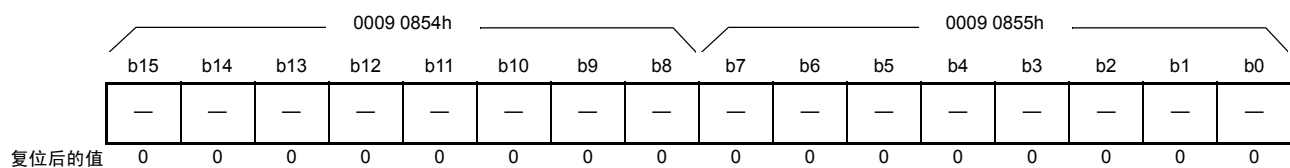
如果通过正在发送的 ACK 定界符检测到格式错误, ADEF 标志就为“1”。

**EDPM 位 (错误表示模式选择位)**

EDPM 位设定 ECSR 寄存器的输出模式。如果将 EDPM 位置“0”, ECSR 寄存器就输出最初的错误代码; 如果将 EDPM 位置“1”, ECSR 寄存器就输出累积的错误代码。

**25.2.23 时戳寄存器 (TSR)**

地址 0009 0854h



位	符号	位名	功能	R/W
b15-b0	—	时戳寄存器	时戳功能的自由运行计数器的值	R

注. 必须以 16 位为单位读 TSR 寄存器。

如果读 TSR 寄存器, 就能读到当时的时戳计数器 (16 位自由运行计数器) 的值。

时戳计数器的基准时钟的值是 1 位时间倍增后的值, 通过 CTLR.TSPS[1:0] 位进行设定。

在 CAN 睡眠模式和 CAN Halt 模式中, 时戳计数器停止计数; 在 CAN 复位模式中, 对时戳计数器进行初始化。

在将接收信息保存到收件箱时, 将当时的时戳计数器的值保存到 MBj.TSL[7:0] 或者 MBj.TSH[7:0] 位。

## 25.2.24 测试控制寄存器 (TCR)

地址 0009 0858h

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

复位后的值

位	符号	位名	功能	R/W
b0	TSTE	测试模式允许位	0: 禁止 CAN 测试模式 1: 允许 CAN 测试模式	R/W
b2-b1	TSTM	CAN 测试模式选择位	b2 b1 0 0: 不是 CAN 测试模式 0 1: 只听模式 1 0: 自测试模式 0 (外部环回) 1 1: 自测试模式 1 (内部环回)	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

TCR 寄存器是控制 CAN 测试模式的寄存器。只能在 CAN Halt 模式中更改 TCR 寄存器。

## (1) 只听模式

在 CAN 规格 (ISO11898-1) 中, 建议使用总线监视模式选项。在只听模式中, 能接收有效的数据帧和远程帧, 但是只在 CAN 总线上发送隐性位, 而不发送 ACK 位、超载标志和有效错误标志。

只听模式能用于波特率的检测。

在只听模式中, 所有信箱都不能请求发送。

选择只听模式时的连接如图 25.6 所示。

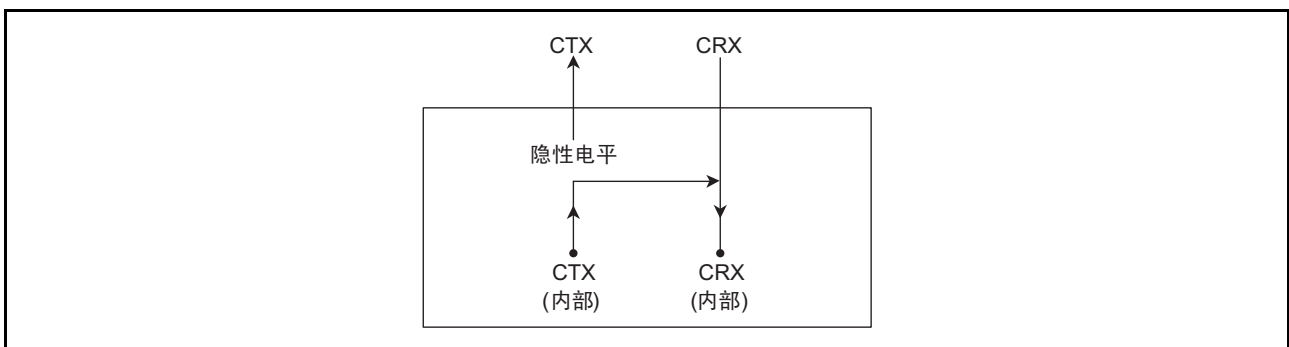


图 25.6 选择只听模式时的连接

## (2) 自测试模式 0 (外部环回)

自测试模式 0 用于 CAN 收发器的测试 (自节点的自诊断功能)。CTX/CRX 引脚必须连接 CAN 收发器。

在自测试模式 0 中, 为了在即使没有连接网络上的其他节点时也能进行自诊断测试, 必须通过从自节点发送 ACK 位来确认 CAN 的正常通信。

选择自测试模式 0 时的连接如图 25.7 所示。

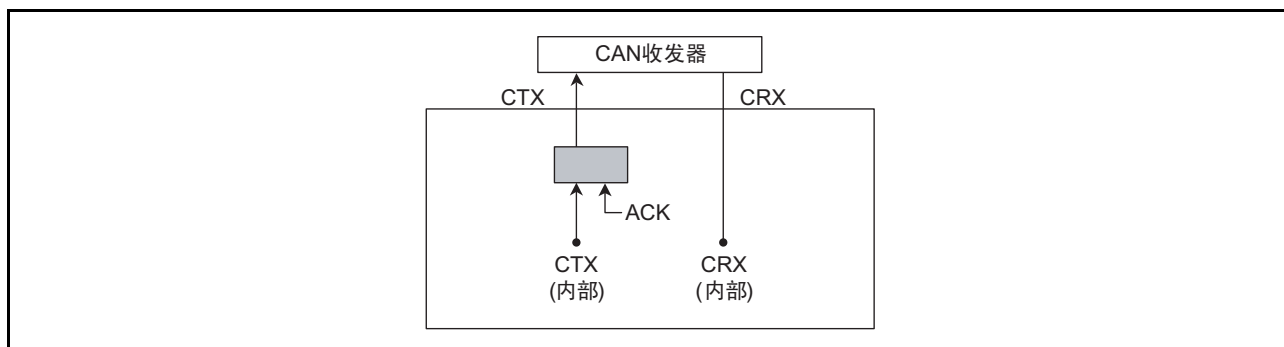


图 25.7 选择自测试模式 0 时的连接

## (3) 自测试模式 1 (内部环回)

自测试模式 1 用于自测试功能。

在自测试模式 1 中, 将发送信息作为接收信息处理, 保存到接收缓冲器。为了能从外部独立进行自测试而生成 ACK 位。

在自测试模式 1 中, 进行内部 CTX 引脚到内部 CRX 引脚的内部反馈。忽视外部 CRX 引脚的输入值, 外部 CTX 引脚只输出隐性位。CTX/CRX 引脚不需要连接 CAN 总线以及其他外部设备。

选择自测试模式 1 时的连接如图 25.8 所示。

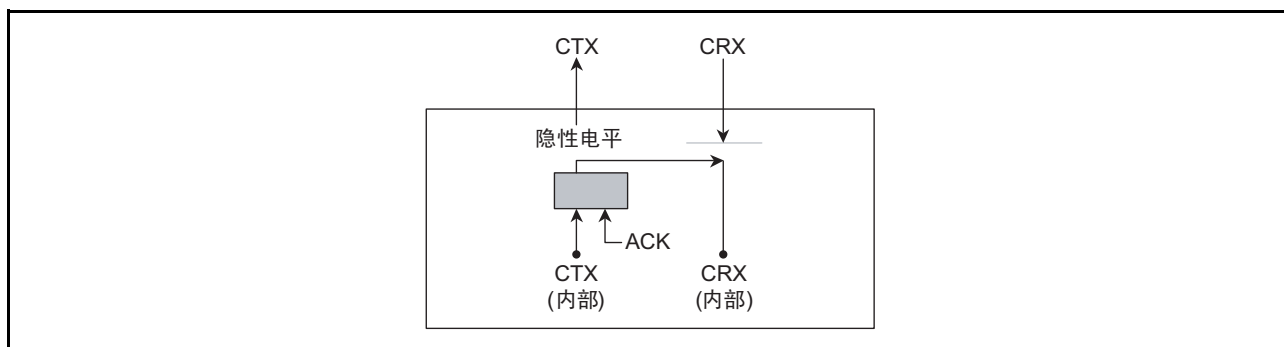


图 25.8 选择自测试模式 1 时的连接

### 25.3 运行模式

CAN 模块有以下 4 种运行模式：

- CAN 复位模式
- CAN Halt 模式
- CAN 操作模式
- CAN 睡眠模式

CAN 运行模式之间的转移如图 25.9 所示。

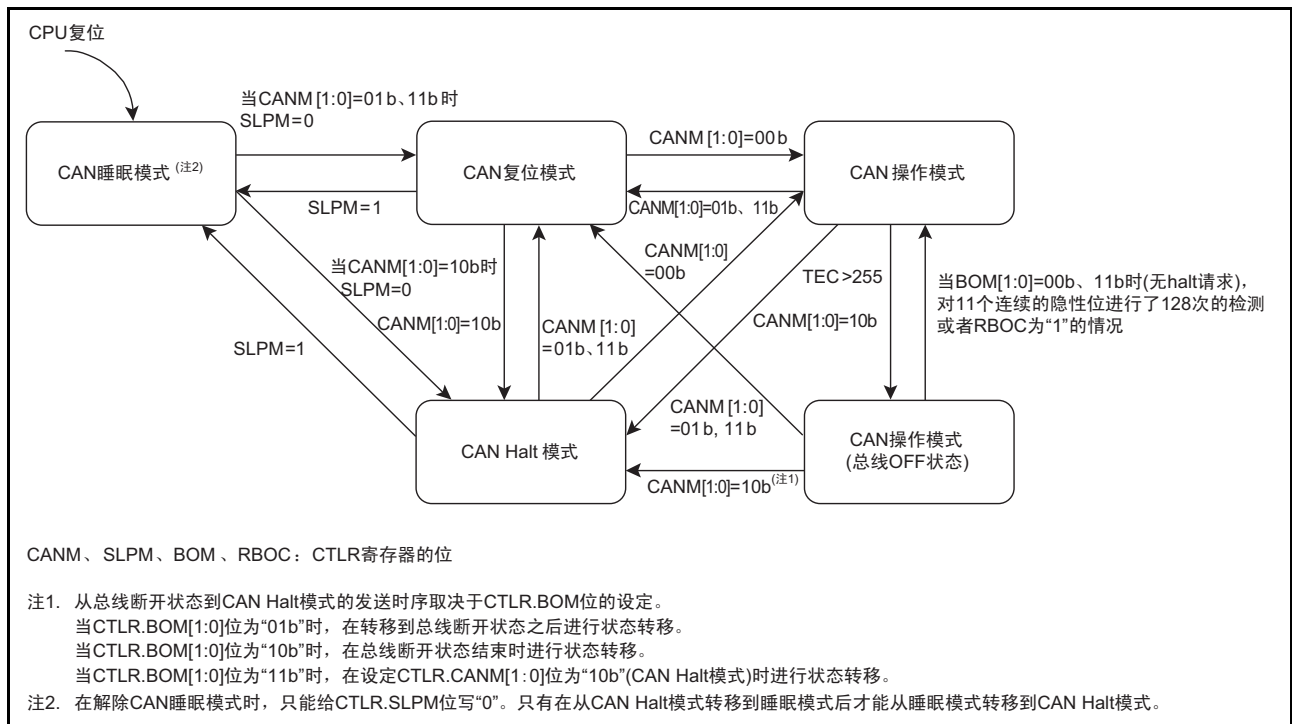


图 25.9 CAN 运行模式之间的转移

### 25.3.1 CAN 复位模式

CAN 复位模式是设定 CAN 通信的模式。

如果将 CTLR.CANM[1:0] 位置 “01b” 或者 “11b”，CAN 模块就变为 CAN 复位模式，此时 STR.RSTST 位为 “1”。不能在 RSTST 位变为 “1” 前更改 CTLR.CANM[1:0] 位。在从 CAN 复位模式转移到其他模式前，必须设定 BCR 寄存器。

以下的寄存器在转移到 CAN 复位模式后被初始化为复位后的值，而在 CAN 复位模式中保持初始值。

- MCTLj 寄存器 (j=0~31)
- STR 寄存器 (SLPST 位和 TFST 位除外)
- EIFR 寄存器
- RECR 寄存器
- TECR 寄存器
- TSR 寄存器
- MSSR 寄存器
- MSMR 寄存器
- RFCR 寄存器
- TFCR 寄存器
- TCR 寄存器
- ECSR 寄存器 (EDPM 位除外)

即使在转移到 CAN 复位模式后，以下的寄存器也保持以前的值：

- CTLR 寄存器
- STR 寄存器 (SLPST 位和 TFST 位)
- MIER 寄存器
- EIER 寄存器
- BCR 寄存器
- CSSR 寄存器
- ECSR 寄存器 (只限于 EDPM 位)
- MBj 寄存器 (j=0~31)
- MKRi 寄存器 (i=0~7)
- FIDCR0 寄存器、COFIDCR1 寄存器
- MKIVLR 寄存器
- AFSR 寄存器
- RFPCR 寄存器
- TFPCR 寄存器

### 25.3.2 CAN Halt 模式

CAN Halt 模式是设定信箱和测试模式的模式。

如果将 CTLR.CANM[1:0] 位置“10b”，就变为 CAN Halt 模式，此时 STR.HLTST 标志为“1”。不能在 STR.HLTST 标志变为“1”前更改 CTLR.CANM[1:0] 位。

有关发送或者接收时的状态转移条件，请参照表 25.9。

在向 CAN Halt 模式转移时，STR.RSTST 标志、STR.HLTST 标志和 STR.SLPST 标志以外的标志以及其他全部寄存器都不变。

在 CAN Halt 模式中，不能更改 CTLR 寄存器（CANM[1:0] 位和 SLPM 位除外）和 EIER 寄存器。在 CAN 测试模式中，只有在选择只听模式时，才能用于自动波特率的检测；在 CAN Halt 模式中，能更改 BCR 寄存器。

表 25.9 CAN 复位模式和 CAN Halt 模式中的运行

模式	接收	发送	总线断开
CAN 复位模式 (强制转移) CANM[1:0]=11b	CAN 模块不等到接收完信息就转移到 CAN 复位模式。	CAN 模块不等到发送完信息就转移到 CAN 复位模式。	CAN 模块不等到总线断开恢复结束就转移到 CAN 复位模式。
CAN 复位模式 CANM[1:0]=01b	CAN 模块不等到接收完信息就转移到 CAN 复位模式。	CAN 模块在等到发送完信息后转移到 CAN 复位模式（注 1、注 4）。	CAN 模块不等到总线断开恢复结束就转移到 CAN 复位模式。
CAN Halt 模式	CAN 模块在等到接收完信息后转移到 CAN Halt 模式（注 2、注 3）。	CAN 模块在等到发送完信息后转移到 CAN Halt 模式（注 1、注 4）。	[CTLR.BOM[1:0] 位为“00b”的情况] CAN 模块只在总线断开恢复后接收程序的 Halt 请求。 [CTLR.BOM[1:0] 位为“01b”的情况] CAN 模块不等到总线断开恢复结束就自动转移到 CAN Halt 模式（与程序的 Halt 请求无关）。 [CTLR.BOM[1:0] 位为“10b”的情况] CAN 模块在等到总线断开恢复结束后自动转移到 CAN Halt 模式（与程序的 Halt 请求无关）。 [CTLR.BOM[1:0] 位为“11b”的情况] 如果 CAN 模块在总线断开过程中有程序的 Halt 请求，就立即转移到 CAN Halt 模式（不等到总线断开恢复结束）。

- 注 1. 如果有几个信息请求发送，就在结束最初的发送后转移模式。在发送挂起过程中请求 CAN 复位模式的状态下，如果为总线空闲状态或者下次发送结束或者 CAN 模块进入接收状态，就转移模式。
- 注 2. 在 CAN 总线被锁定为显性电平的情况下，如果监视 EIFR.BLIF 位，程序就能检测到总线锁定状态。
- 注 3. 在请求 CAN Halt 模式后，如果在接收过程中发生 CAN 总线错误，就立即转移到 CAN Halt 模式。
- 注 4. 在请求 CAN 复位模式或者 CAN Halt 模式后，如果在发送过程中发生 CAN 总线错误或者 CAN 仲裁失败，CAN 模块就立即转移到所请求的 CAN 模式。

### 25.3.3 CAN 睡眠模式

CAN 睡眠模式是通过停止给 CAN 模块提供时钟而减少消耗电流的模式。在进行 MCU 的引脚复位或者软件复位后，CAN 模块从 CAN 睡眠模式开始运行。

如果将 CTLR.SLPM 位置“1”，就变为 CAN 睡眠模式，此时 STR.SLPST 标志为“1”。不能在 STR.SLPST 位标志为“1”前更改 CTLR.SLPM 位的值。在向 CAN 睡眠模式转移时，其他寄存器不变。

必须在 CAN 复位模式和 CAN Halt 模式中更改 CTLR.SLPM 位，不能在 CAN 睡眠模式中更改除 CTLR.SLPM 位以外的寄存器。允许读操作。

如果将 CTLR.SLPM 位置“0”，就解除 CAN 睡眠模式。在从 CAN 睡眠模式返回时，其他寄存器不变。

### 25.3.4 CAN 操作模式（非总线断开状态）

CAN 操作模式是进行 CAN 通信的模式。

如果将 CTLR.CANM[1:0] 位置“00b”，CAN 模块就变为 CAN 操作模式。

此时，STR.RSTST 标志和 STR.HLTST 标志为“0”。不能在 STR.RSTST 标志和 STR.HLTST 标志为“0”前更改 CTLR.CANM[1:0] 位的值。

如果在转移到 CAN 操作模式后检测到 11 个连续的隐性位，CAN 模块就变为以下的状态：

- CAN 模块为能进行通信的网络上的有效节点，能发送和接收 CAN 信息。
- 接收错误计数器和发送错误计数器等进行 CAN 总线的错误监视处理。

根据 CAN 总线的状态，CAN 模块在 CAN 操作模式中有以下 3 种子模式：

- 空闲模式：CAN 模块为未进行发送或者接收的状态。
- 接收模式：CAN 模块正在接收其他节点发送的 CAN 信息。
- 发送模式：CAN 模块正在发送 CAN 信息。在选择自测试模式 0（TCR.TSTM[1:0] 位为“10b”）或者自测试模式 1（TCR.TSTM[1:0] 位为“11b”）时，同时接收自身发送的信息。

CAN 操作模式的子模式如图 25.10 所示。

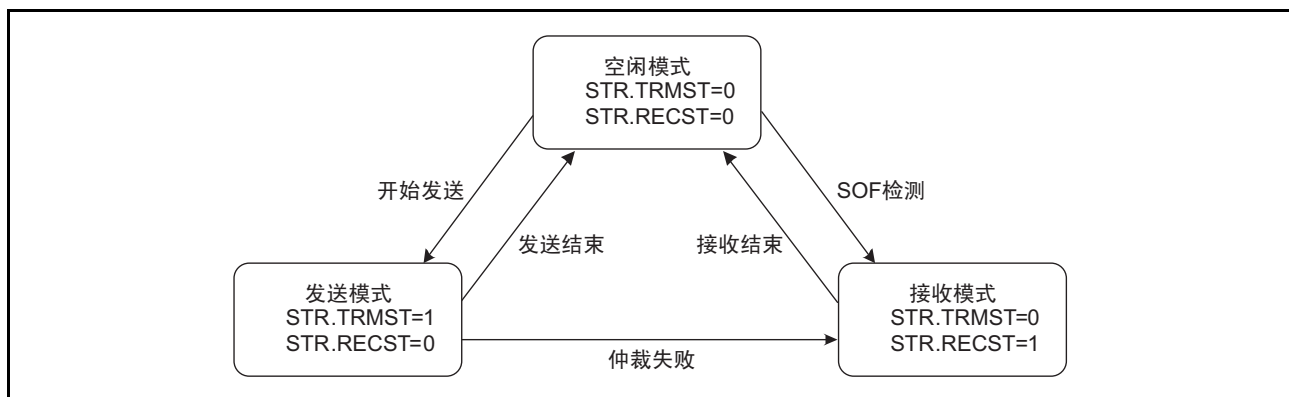


图 25.10 CAN 操作模式的子模式



### 25.3.5 CAN 操作模式 (总线断开状态)

根据 CAN 规格的发送错误计数器和接收错误计数器的增减规则, CAN 模块转移到总线断开状态。

在以下情况下, CAN 模块从总线断开状态恢复。在总线断开状态下, 除 STR、EIFR、RECR、TECR 和 TSR 寄存器以外, CAN 模块的相关寄存器的值不变。

#### (1) CTLR.BOM[1:0] 位为 “00b” 的情况 (正常模式)

在从总线断开状态恢复结束后立即转移到错误主动状态, 能进行 CAN 通信。此时, EIFR.BORIF 标志为 “1” (检测到总线断开恢复)。

#### (2) 将 CTLR.RBOC 位置 “1” 的情况 (强制从总线断开状态恢复)

如果 CAN 模块为总线断开状态并且 CTLR.RBOC 位为 “1”, CAN 模块就立即转移到错误主动状态, 并且能在检测到 11 个连续的隐性位后再次进行 CAN 通信。此时, EIFR.BORIF 标志不为 “1”。

#### (3) CTLR.BOM[1:0] 位为 “01b” 的情况 (在总线断开开始时自动转移到 CAN Halt 模式)

一旦进入总线断开状态, 就立即变为 CAN Halt 模式。此时, EIFR.BORIF 标志不为 “1”。

#### (4) CTLR.BOM[1:0] 位为 “10b” 的情况 (在总线断开结束时自动转移到 CAN Halt 模式)

一旦从总线断开恢复结束, 就变为 CAN Halt 模式。此时, EIFR.BORIF 标志为 “1”。

#### (5) CTLR.BOM[1:0] 位为 “11b” (通过程序转移到 CAN Halt 模式) 并且在总线断开状态下将 CTLR.CANM[1:0] 位置 “10b” 的情况 (CAN Halt 模式)

如果在总线断开状态下将 CTLR.CANM[1:0] 位置 “10b” (CAN Halt 模式), 就立即变为 CAN Halt 模式。此时, EIFR.BORIF 标志不为 “1”。

不在总线断开过程中将 CTLR.CANM[1:0] 位置 “10b” 时, 和 (1) 的运行相同。

## 25.4 CAN 通信速度的设定

以下说明有关 CAN 通信速度的设定。

### 25.4.1 CAN 时钟的设定

CAN 模块内置 CAN 时钟选择电路。

能通过 BCR.BRP[9:0] 位设定 CAN 时钟。

CAN 时钟发生电路的框图如图 25.11 所示。

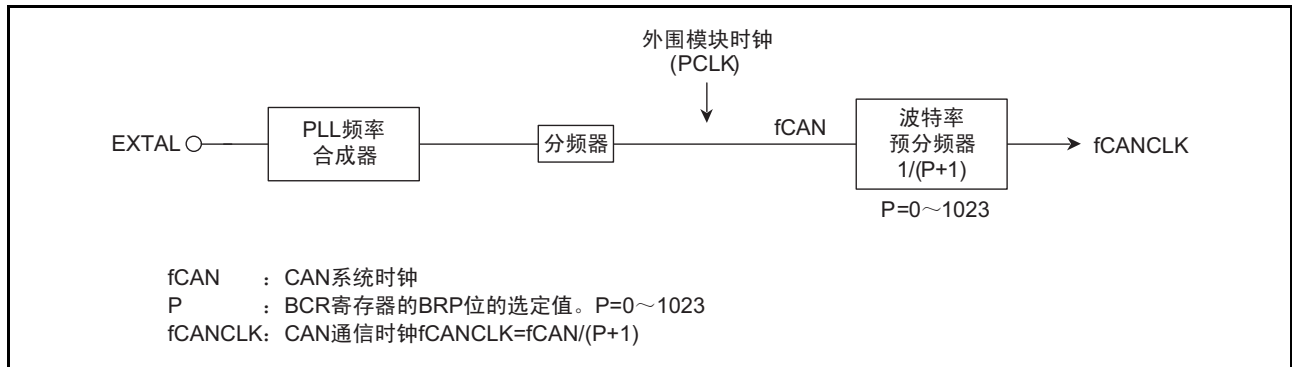


图 25.11 CAN 时钟发生电路的框图

### 25.4.2 位时序的设定

位时间由以下 3 段构成。

位时序如图 25.12 所示。

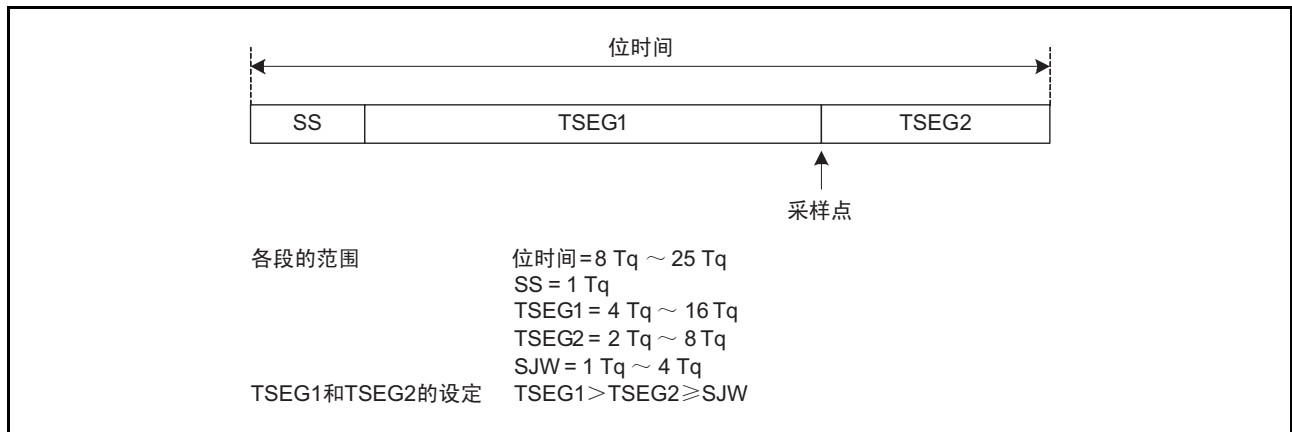


图 25.12 位时序图

### 25.4.3 位速率

位速率取决于 fCAN (CAN 系统时钟) 的分频值、波特率预分频器的分频值和 1 位的 Tq 数。

$$\text{位速率[bps]} = \frac{f_{\text{CAN}}}{\text{波特率预分频器的分频值}^{(\text{注}1)} \times \text{1位时间的Tq数}} = \frac{f_{\text{CANCLK}}}{\text{1位时间的Tq数}}$$

注 1. 波特率预分频器的分频值 = P+1 (P=0 ~ 1023)

P: BCR.BRP[9:0] 位的设定值

位速率的例子如表 25.10 所示。

表 25.10 位速率的例子

fCAN	50MHz		48MHz		40MHz		32MHz	
位速率	Tq 数	P+1	Tq 数	P+1	Tq 数	P+1	Tq 数	P+1
1Mbps	10Tq 25Tq	5 2	8Tq 12Tq 16Tq	6 4 3	10Tq 20Tq	4 2	8Tq 16Tq	4 2
500kbps	10Tq 25Tq	10 4	8Tq 12Tq 16Tq	12 8 6	10Tq 20Tq	8 4	8Tq 16Tq	8 4
250kbps	10Tq 25Tq	20 8	8Tq 12Tq 16Tq	24 16 12	10Tq 20Tq	16 8	8Tq 16Tq	16 8
125kbps	10Tq 25Tq	40 16	8Tq 12Tq 16Tq	48 32 24	10Tq 20Tq	32 16	8Tq 16Tq	32 16
83.3kbps	10Tq 25Tq	60 24	8Tq 12Tq 16Tq	72 48 36	8Tq 10Tq 16Tq 20Tq	60 48 30 24	8Tq 16Tq	48 24
33.3kbps	10Tq 25Tq	150 60	8Tq 12Tq 16Tq	180 120 90	8Tq 10Tq 20Tq	150 120 60	8Tq 10Tq 16Tq 20Tq	120 96 60 48

### 25.5 信箱和屏蔽寄存器的结构

MBj 寄存器的结构如图 25.13 所示。  
有 32 个结构相同的信箱。

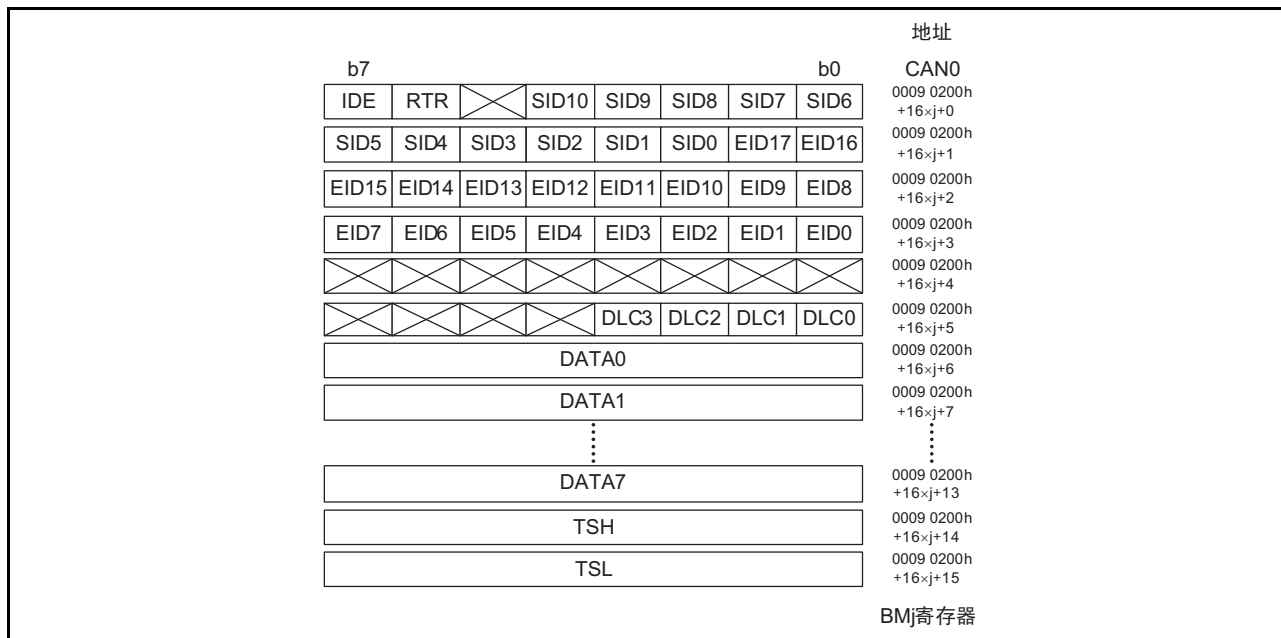


图 25.13 MBj 寄存器的结构 (j=0 ~ 31)

MKRi 寄存器的结构如图 25.14 所示。  
有 8 个结构相同的屏蔽寄存器。

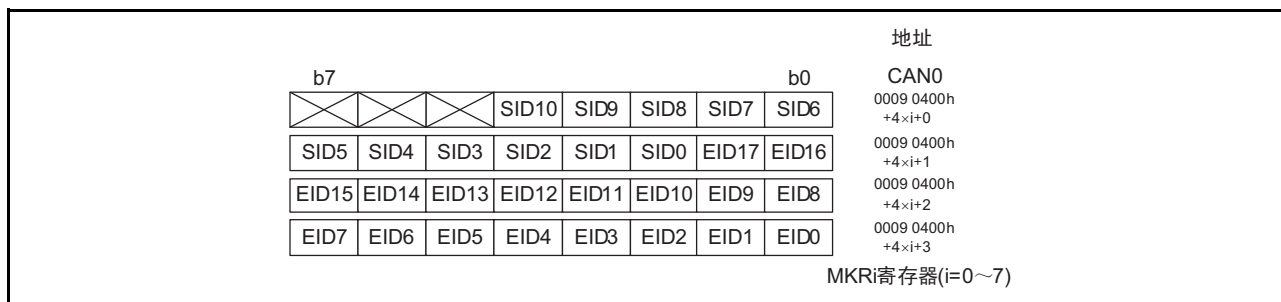


图 25.14 MKRi 寄存器的结构 (i=0 ~ 7)

FIDCR0 寄存器和 FIDCR1 寄存器的结构如图 25.15 所示。  
有 2 个结构相同的 FIFO 接收 ID 比较寄存器。

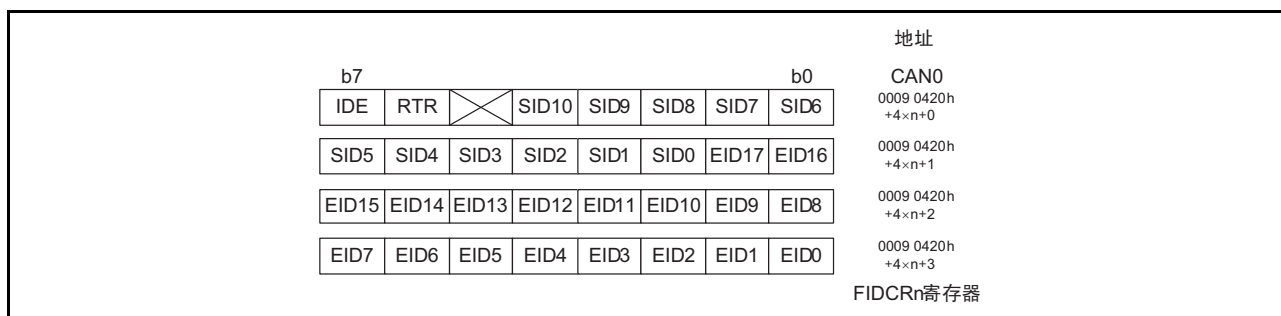


图 25.15 FIDCRn 寄存器的结构 (n=0、1)

## 25.6 验收过滤功能和屏蔽功能

验收过滤功能和屏蔽功能允许选择和接收一定范围的 ID。

MKR0 ~ MKR7 寄存器能屏蔽标准 ID 和 29 位扩展 ID。

- MKR0 寄存器对应信箱 [0] ~ [3]。
- MKR1 寄存器对应信箱 [4] ~ [7]。
- MKR2 寄存器对应信箱 [8] ~ [11]。
- MKR3 寄存器对应信箱 [12] ~ [15]。
- MKR4 寄存器对应信箱 [16] ~ [19]。
- MKR5 寄存器对应信箱 [20] ~ [23]。
- 在普通信箱模式中，MKR6 寄存器对应信箱 [24] ~ [27]；在 FIFO 信箱模式中，MKR6 寄存器对应接收 FIFO 信箱 [28] ~ [31]。
- 在普通信箱模式中，MKR7 寄存器对应信箱 [28] ~ [31]；在 FIFO 信箱模式中，MKR7 寄存器对应接收 FIFO 信箱 [28] ~ [31]。

MKIVLR 寄存器分别禁止各信箱的验收过滤功能。

在 CTLR.IDFM[1:0] 位为“10b”（混合 ID 模式）时，MBj 寄存器的 IDE 位有效。

MBj 寄存器的 RTR 位选择数据帧和远程帧。

在 FIFO 信箱模式中，普通信箱（信箱 [0] ~ [23]）使用 MKR0 ~ MKR5 寄存器中对应的 1 个寄存器进行验收过滤处理，而接收 FIFO 信箱（信箱 [28] ~ [31]）使用 MKR6 寄存器和 MKR7 寄存器进行验收过滤处理。

接收 FIFO 使用 FIDCR0 寄存器和 FIDCR1 寄存器进行 ID 的比较。接收 FIFO 的 MB28 ~ MB31 寄存器的 EID、SID、RTR 和 IDE 位无效。因为分别用 2 个逻辑或的结果进行验收过滤处理，所以接收 FIFO 能接收 2 个范围的 ID。

MKIVLR 寄存器对接收 FIFO 无效。

如果将标准 ID 和扩展 ID 分别设定到 FIDCR0.IDE 和 FIDCR1.IDE 位，就接收标准 ID 和扩展 ID 的 ID 格式。

如果将数据帧和远程帧分别设定到 FIDCR0.RTR 和 FIDCR1.RTR 位，就接收数据帧和远程帧。

在不需要 2 个范围的 ID 组合时，必须给 FIFO ID 和屏蔽寄存器设定相同的屏蔽值和 ID。

屏蔽寄存器和信箱的对应以及验收过滤处理分别如图 25.16 和图 25.17 所示。

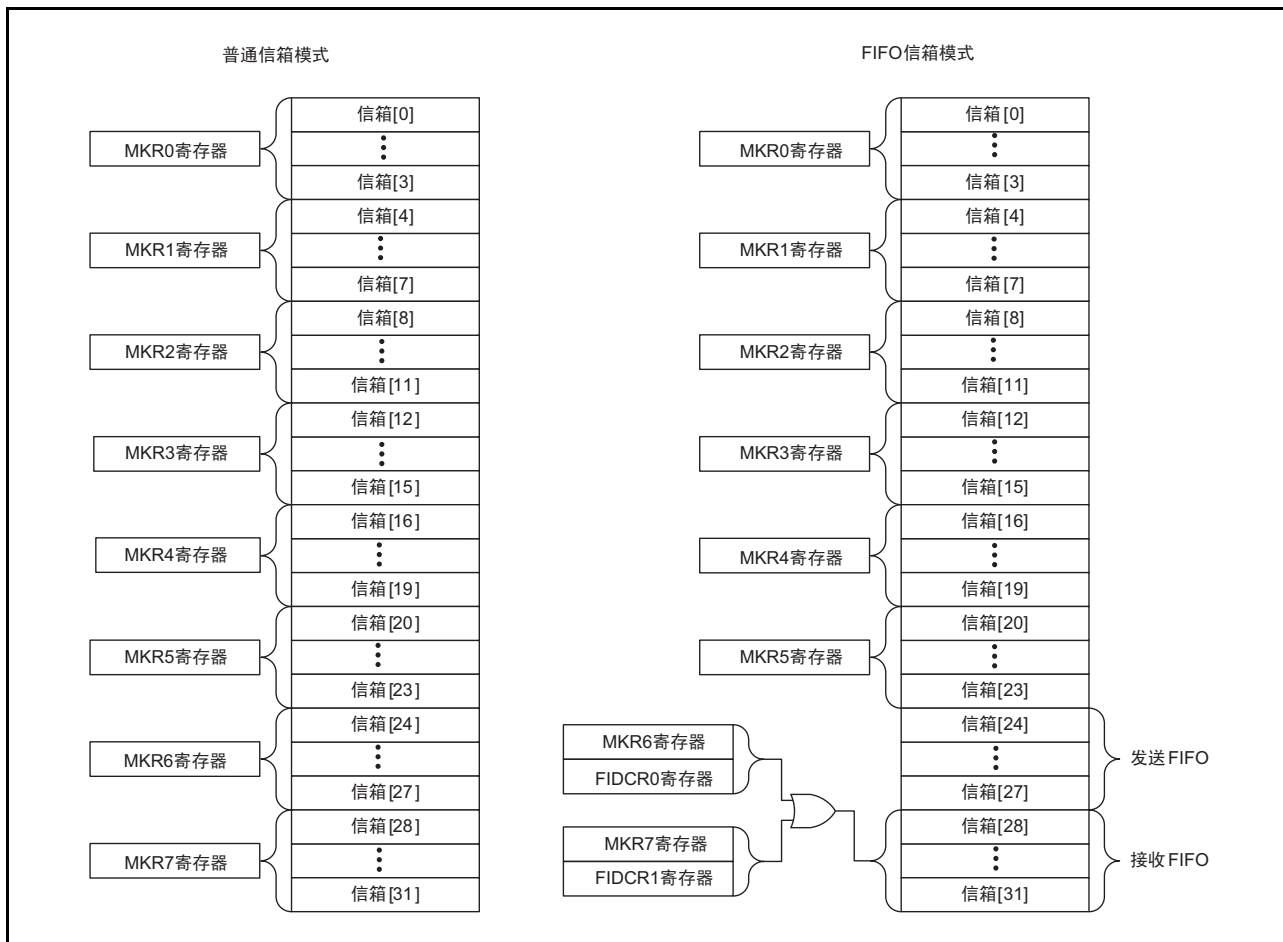


图 25.16 屏蔽寄存器和信箱的对应

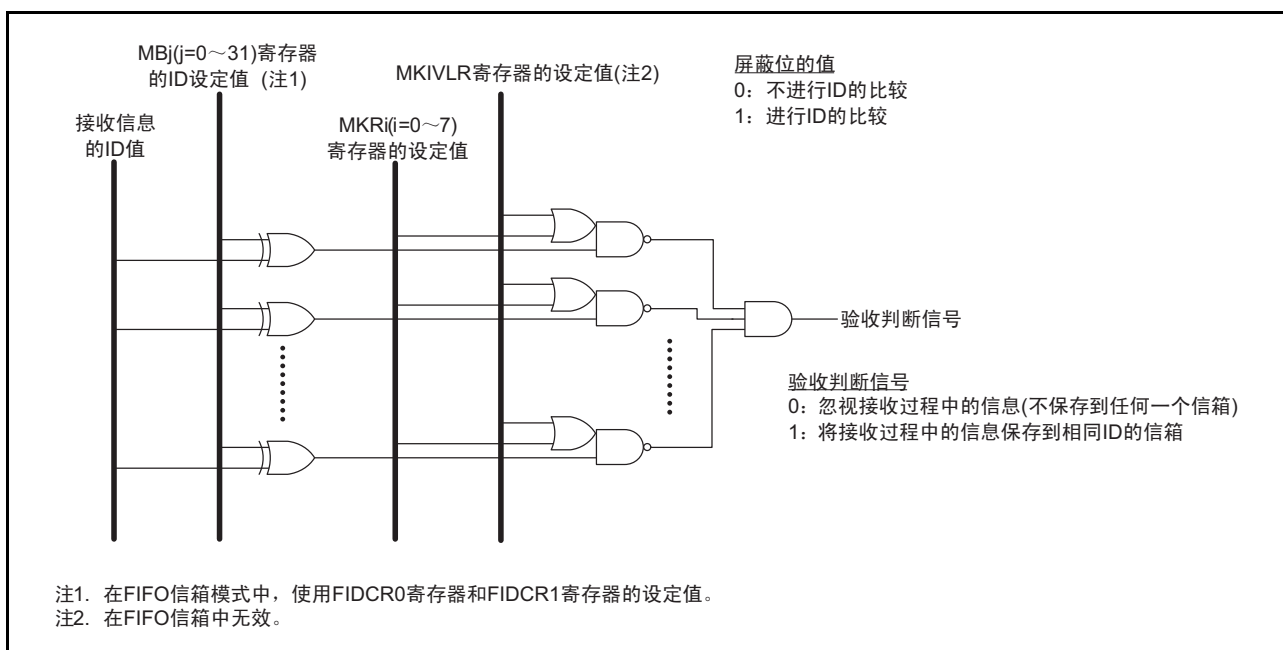


图 25.17 验收过滤处理

## 25.7 接收 / 发送

CAN 通信模式的设定方法如表 25.11 所示。

表 25.11 CAN 接收模式和 CAN 发送模式的设定方法

MCTLj. TRMREQ	MCTLj. RECREQ	MCTLj. ONESHOT	信箱的通信模式
0	0	0	不能使用信箱或者正在中止发送。
0	0	1	只有在单次模式中已中止所编程信箱的发送或者接收时才能进行设定。
0	1	0	设定为数据帧或者远程帧的收件箱。
0	1	1	设定为数据帧或者远程帧的单次收件箱。
1	0	0	设定为数据帧或者远程帧的发件箱。
1	0	1	设定为数据帧或者远程帧的单次发件箱。
1	1	0	不能设定
1	1	1	不能设定

j=0 ~ 31

在将信箱设定为收件箱或者单次收件箱时，必须注意以下事项：

1. 在将信箱设定为收件箱或者单次收件箱前，必须将MCTLj寄存器置“00h”。
2. 根据接收模式的设定和验收滤波器的处理结果，将接收信息保存到条件相同的第一个信箱。对于保存接收信息的信箱，信箱号越小，优先级越高。
3. 在CAN操作模式中，如果发送和被设定为接收信息的信箱的ID/屏蔽相同的信息，CAN模块就不接收发送数据。但是，在自测试模式中CAN模块接收发送数据，此时CAN模块返回ACK。

在将信箱设定为发件箱或者单次发件箱时，必须注意以下事项：

1. 在将信箱设定为发件箱或者单次发件箱前，必须确认MCTLj寄存器为“00h”并且不在进行中止处理。

### 25.7.1 接收

接收数据帧时的运行例子（重写模式）如图 25.18 所示。

在此例中，当接收到和 MCTLj 寄存器（j=0 ~ 31）的信箱条件相同的 2 个连续的 CAN 信息时，重写最初的信息。

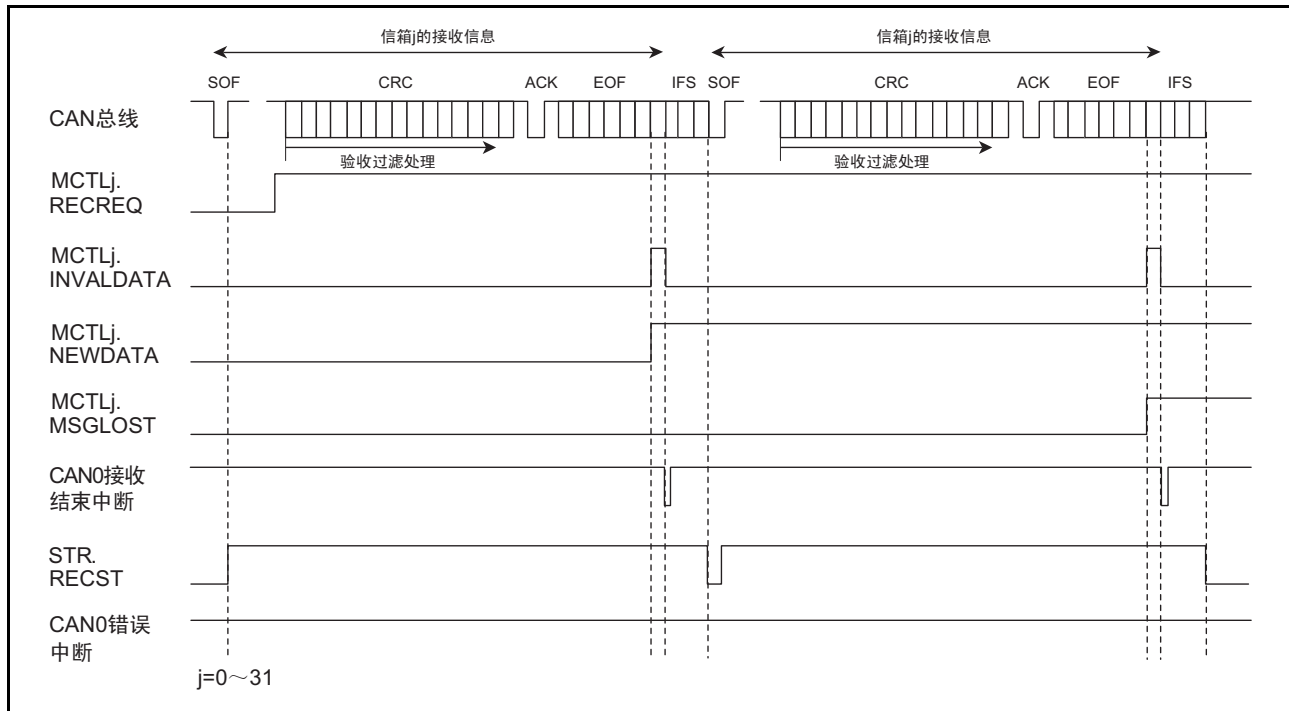


图 25.18 接收数据帧时的运行例子（重写模式）

1. 当在 CAN 总线上检测到 SOF 时，如果 CAN 模块没有要开始发送的信息，STR.RECST 位就立即变为“1”（正在接收）。
2. 要选择收件箱时，从 CRC 字段的开头开始进行验收过滤处理。
3. 一旦接收完信息，收件箱的 MCTLj.NEWDATA 位就变为“1”（正在更新新信息或者已保存到信箱），同时 MCTLj.INVALIDATA 位变为“1”（正在更新信息），在给该信箱传送完全部信息后，MCTLj.INVALIDATA 位恢复为“0”（信息有效）。
4. 当收件箱的 MIER 寄存器的中断允许位为“1”（允许中断）时，产生 CAN0 接收结束中断请求。如果 MCTLj.INVALIDATA 位为“0”，就发生此中断（CAN0 接收结束中断）。
5. 在从信箱读信息后，需要通过程序将 MCTLj.NEWDATA 标志置“0”。
6. 在重写模式中，如果 MCTLj.NEWDATA 标志再次变为“1”的信箱接收完下一条 CAN 信息，MCTLj.MSGLOST 标志就变为“1”（重写信息），并且将新的接收信息传送到信箱。CAN0 接收结束中断请求的产生条件和 4. 相同。



接收数据帧时的运行例子（溢出模式）如图 25.19 所示。

在此例中，当接收到和 MCTLj 寄存器（j=0 ~ 31）的信箱条件相同的 2 个连续的 CAN 信息时，放弃第 2 条信息。

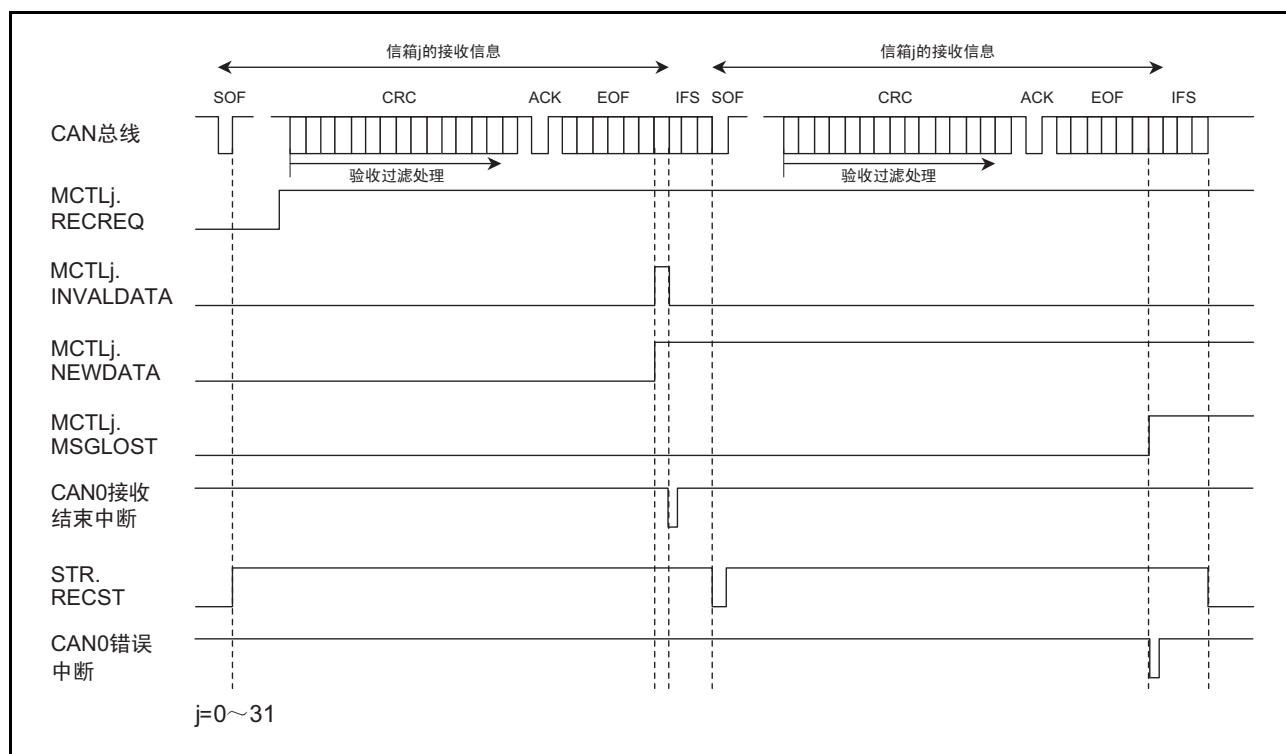


图 25.19 接收数据帧时的运行例子（溢出模式）

1. ~ 5. 和重写模式相同。
6. 在溢出模式中，如果在 MCTLj.NEWDATA 标志变为“0”前接收完下一条 CAN 信息，MCTLj.MSGLOST 标志就变为“1”（信息溢出）。当放弃新的接收信息并且 EIER 寄存器的对应中断允许位为“1”（中断允许）时，产生 CAN0 错误中断请求。

## 25.7.2 发送

发送数据帧时的运行例子如图 25.20 所示。

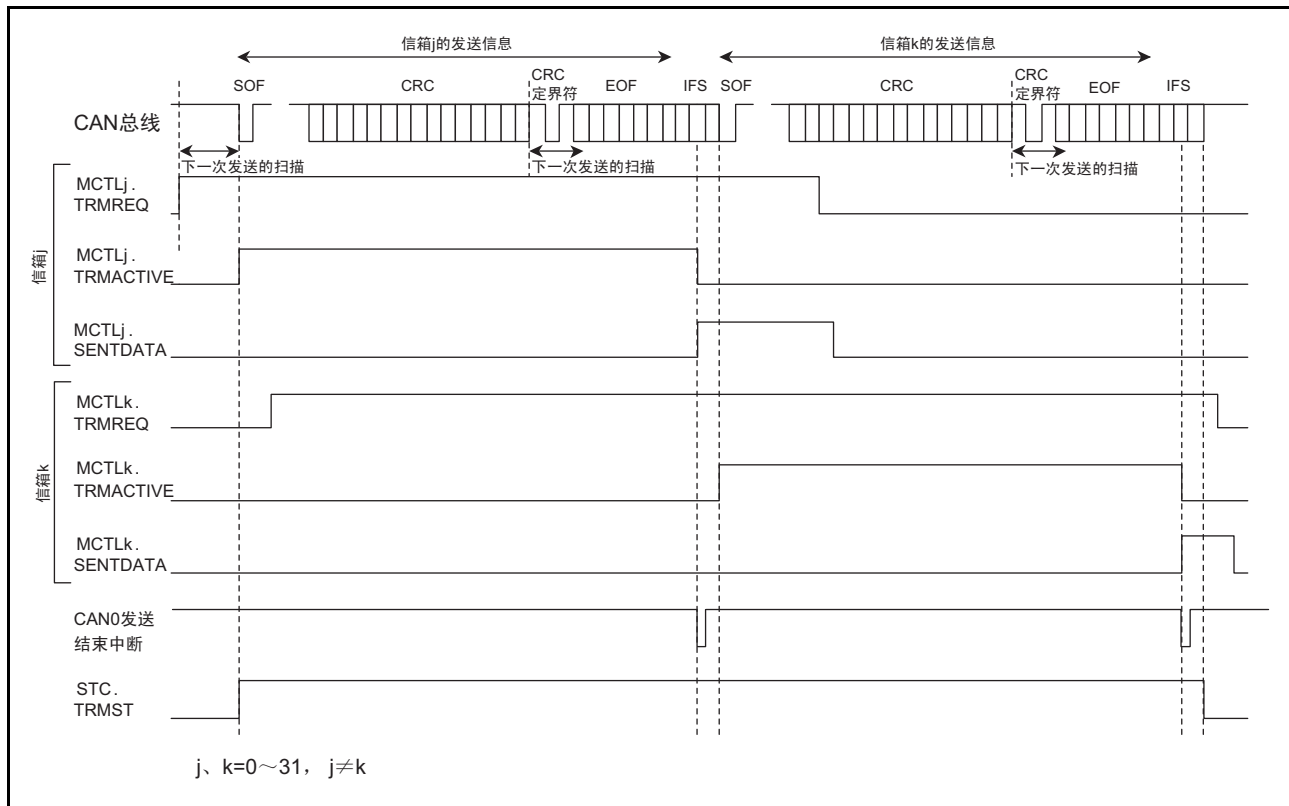


图 25.20 发送数据帧时的运行例子

1. 如果在总线空闲状态下将MCTLj.TRMREQ位 ( $j=0\sim 31$ ) 置“1” (发件箱), 为了决定优先级最高的发件箱而开始信箱取消处理。一旦决定发件箱, MCTLj.TRMACTIVE位就变为“1” (从取发送请求到发送结束、发生错误或者仲裁失败为止验收) 并且STR.TRMST位标志为“1” (正在发送), CAN模块开始发送 (注1)。
2. 在设定了其他MCTLj.TRMREQ位时, 从CRC 定义符开始下次发送的发送取消处理。
3. 当不发生仲裁失败而结束发送时, MCTLj.SENTDATA标志为“1” (发送结束) 并且MCTLj.TRMACTIVE位为“0” (正处于发送待机或者无发送请求)。当MIER寄存器的中断允许位为“1” (允许中断) 时, 产生CAN0发送结束中断请求。
4. 在从相同信箱请求下次发送时, 必须将MCTLj.SENTDATA标志和MCTLj.TRMREQ位置“0”, 在确认MCTLj.SENTDATA标志和MCTLj.TRMREQ位为“0”后将MCTLj.TRMREQ位置“1”。

注 1. 如果在CAN模块开始发送后发生仲裁失败, MCTLj.TRMACTIVE位就变为“0”。为了从CRC 定义符开始搜索优先级最高的发件箱, 需要再次进行发送取消处理。当正在进行发送或者在发生仲裁失败后发生错误时, 为了从错误定义符开始搜索优先级最高的发件箱, 需要再次进行发送取消处理。

## 25.8 CAN 中断

CAN 模块的各通道有以下的 CAN 中断，CAN 中断一览表如表 25.12 所示。

- CAN0 接收结束中断 (信箱 0~31) [RXM0]
- CAN0 发送结束中断 (信箱 0~31) [TXM0]
- CAN0 接收 FIFO 中断 [RXF0]
- CAN0 发送 FIFO 中断 [TXF0]
- CAN0 错误中断 [ERS0]

CAN0 错误中断有 8 个中断源。能通过检查 EIFR 寄存器，确认这些中断源。

- 总线错误
- 错误警告
- 错误被动
- 总线断开开始
- 总线断开恢复
- 接收溢出
- 超载帧发送
- 总线锁定

表 25.12 CAN 中断一览表

模块	中断符号	中断源	中断源标志
CAN0	ERS0	总线锁定检测	EIFR.BLIF
		超载帧发送检测	EIFR.OLIF
		溢出检测	EIFR.ORIF
		总线断开恢复检测	EIFR.BORIF
		总线断开开始检测	EIFR.BOEIF
		错误被动检测	EIFR.EPIF
		错误警告检测	EIFR.EWIF
		总线错误检测	EIFR.BEIF
	RXF0	接收 FIFO 信息接收 (MIER[29]=0)	—
		接收 FIFO 警告 (MIER[29]=1)	
	TXF0	发送 FIFO 信息发送结束 (MIER[25]=0)	—
		FIFO 最后信息发送结束 (MIER[25]=1)	
	RXM0	信箱 0 ~ 31 信息接收	MCTL0.NEWDATA ~ MCTL31.NEWDATA
	TXM0	信箱 0 ~ 31 信息发送结束	MCTL0.SENTDATA ~ MCTL31.SENTDATA

## 26. 串行外围接口 (RSPI)

### 26.1 概要

RX62T 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、外围设备进行高速串行通信的功能。

RSPI 的规格和框图分别如表 26.1 和图 26.1 所示。

表 26.1 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"> <li>• 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。</li> <li>• 只能进行发送。</li> <li>• 能在主控模式或者从属模式中进行串行通信。</li> <li>• 能更改串行传送时钟的极性。</li> <li>• 能更改串行传送时钟的相位。</li> </ul>
数据格式	<ul style="list-style-type: none"> <li>• 能转换 MSB first 和 LSB first。</li> <li>• 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。</li> <li>• 发送 / 接收缓冲器为 128 位。</li> <li>• 1 次发送 / 接收最多传送 4 帧 (1 帧最多 32 位)。</li> </ul>
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"> <li>• 模式故障错误检测</li> <li>• 溢出错检测</li> <li>• 奇偶校验错误检测</li> </ul>
SSL 控制功能	<ul style="list-style-type: none"> <li>• 每个通道有 4 个 SSL 信号 (SSL0 ~ SSL3)。</li> <li>• 在设定为单主控模式时，输出 SSL0 ~ SSL3 信号。</li> <li>• 在设定为多主控模式时，输入 SSL0 信号，输出 SSL1 ~ SSL3 信号或者将 SSL1 ~ SSL3 信号置为高阻抗状态。</li> <li>• 在设定为从属模式时，输入 SSL0 信号，将 SSL1 ~ SSL3 信号置为高阻抗状态。</li> <li>• 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</li> <li>• 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</li> <li>• 能设定下次存取的 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</li> <li>• SSL 极性变更功能</li> </ul>
主控传送时的控制方式	<ul style="list-style-type: none"> <li>• 能按顺序循环执行由最多 8 个命令构成的传送。</li> <li>• 能对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</li> <li>• 能通过写发送缓冲器来启动传送。</li> <li>• 能设定 SSL 无效时的 MOSI 信号值。</li> </ul>
中断源	<ul style="list-style-type: none"> <li>• 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、溢出错和奇偶校验错误) RSPI 空闲中断 (RSPI 空闲)</li> </ul>
其他功能	<ul style="list-style-type: none"> <li>• RSPI 禁止 (初始化) 功能</li> <li>• 环回模式功能</li> </ul>

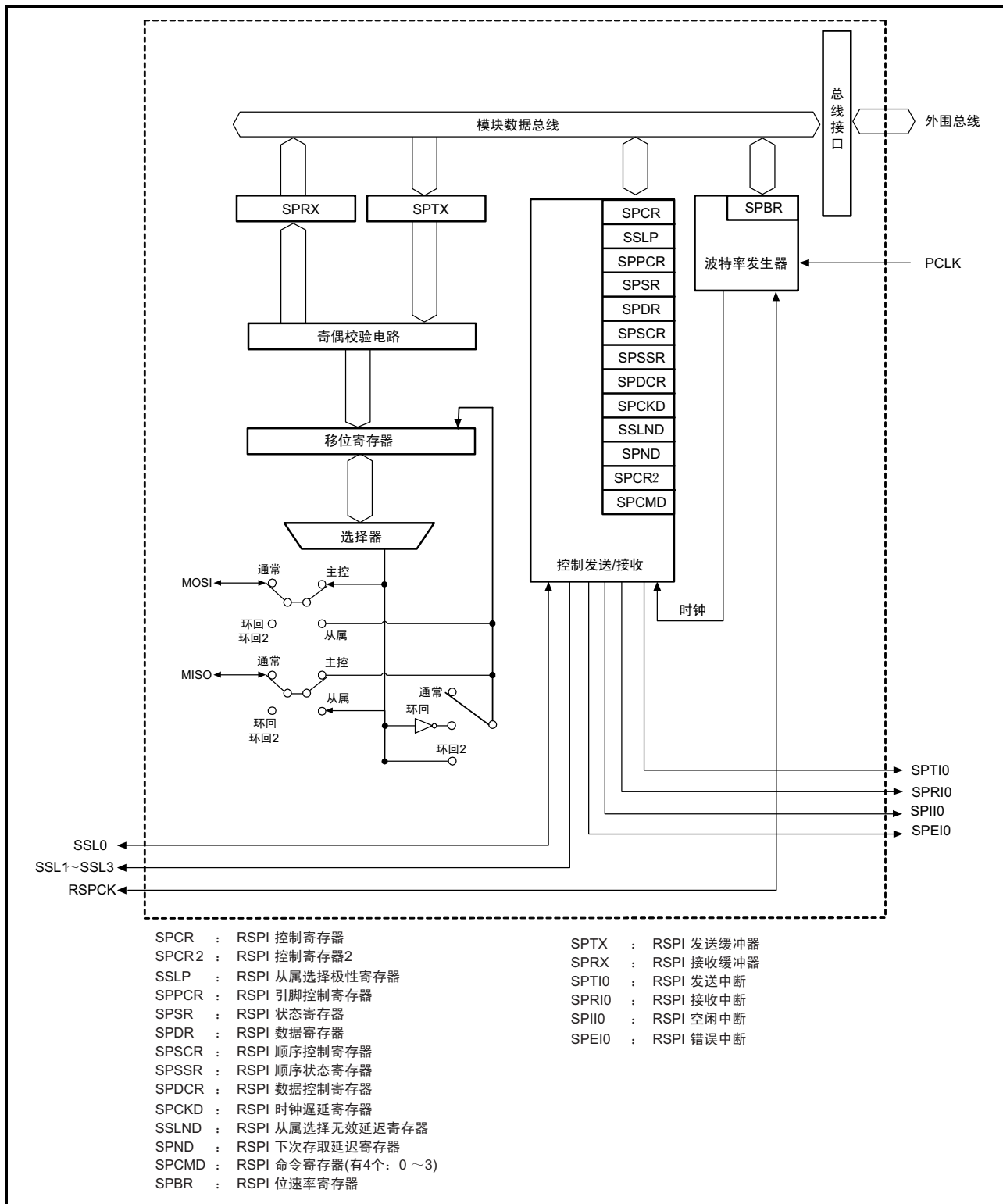


图 26.1 RSPI 的框图

RSPI 使用的输入 / 输出引脚如表 26.2 所示。

在设定为单主控模式时，RSPI 自动将 SSL0 引脚的输入 / 输出方向转换为输出；在设定为多主控模式和从属模式时，RSPI 自动将 SSL0 引脚的输入 / 输出方向转换为输入。根据主控模式 / 从属模式的设定和 SSL0 引脚的输入电平，RSPI 自动转换 RSPCK、MOSI 和 MISO 引脚的输入 / 输出方向。

详细内容请参照“26.3.2 RSPI 引脚的控制”。

表 26.2 RSPI 的输入 / 输出引脚

模块符号	引脚名	输入 / 输出	功能
RSPI0	RSPCK	输入 / 输出	时钟输入 / 输出引脚
	MOSI	输入 / 输出	主控发送数据输入 / 输出引脚
	MISO	输入 / 输出	从属发送数据输入 / 输出引脚
	SSL0	输入 / 输出	从属选择输入 / 输出引脚
	SSL1	输出	从属选择输出引脚
	SSL2	输出	从属选择输出引脚
	SSL3	输出	从属选择输出引脚

## 26.2 寄存器说明

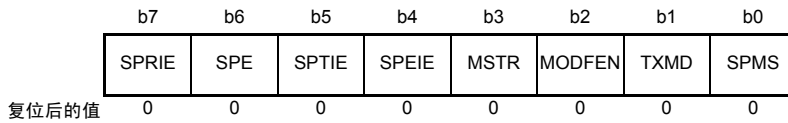
RSPI 的寄存器一览表如表 26.3 所示。能通过这些寄存器指定主控 / 从属模式和传送格式以及控制发送部 / 接收部。

表 26.3 RSPI 的寄存器一览表

模块符号	寄存器名	寄存器符号	复位后的值	地址	存取长度
RSPI0	RSPI 控制寄存器	SPCR	00h	0008 8380h	8
	RSPI 从属选择极性寄存器	SSLP	00h	0008 8381h	8
	RSPI 引脚控制寄存器	SPPCR	00h	0008 8382h	8
	RSPI 状态寄存器	SPSR	20h	0008 8383h	8
	RSPI 数据寄存器	SPDR	00000000h	0008 8384h	16、32
	RSPI 顺序控制寄存器	SPSCR	00h	0008 8388h	8
	RSPI 顺序状态寄存器	SPSSR	00h	0008 8389h	8
	RSPI 位速率寄存器	SPBR	FFh	0008 838Ah	8
	RSPI 数据控制寄存器	SPDCR	00h	0008 838Bh	8
	RSPI 时钟延迟寄存器	SPCKD	00h	0008 838Ch	8
	RSPI 从属选择无效延迟寄存器	SSLND	00h	0008 838Dh	8
	RSPI 下次存取延迟寄存器	SPND	00h	0008 838Eh	8
	RSPI 控制寄存器 2	SPCR2	00h	0008 838Fh	8
	RSPI 命令寄存器 0	SPCMD0	070Dh	0008 8390h	16
	RSPI 命令寄存器 1	SPCMD1	070Dh	0008 8392h	16
	RSPI 命令寄存器 2	SPCMD2	070Dh	0008 8394h	16
	RSPI 命令寄存器 3	SPCMD3	070Dh	0008 8396h	16
	RSPI 命令寄存器 4	SPCMD4	070Dh	0008 8398h	16
	RSPI 命令寄存器 5	SPCMD5	070Dh	0008 839Ah	16
	RSPI 命令寄存器 6	SPCMD6	070Dh	0008 839Ch	16
RSPI 命令寄存器 7	SPCMD7	070Dh	0008 839Eh	16	

## 26.2.1 RSPI 控制寄存器 (SPCR)

地址 0008 8380h



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 功能有效位	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

SPCR 寄存器是设定 RSPI 运行模式的寄存器。如果在 RSPI 功能有效的状态 (SPCR.SPE 位 =1) 下更改 SPCR.MSTR 位、SPCR.MODFEN 位和 SPCR.TXMD 位的设定值, 就无法保证以后的运行。

### 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 SSL0 ~ SSL3 引脚而使用 RSPCK 引脚、MOSI 引脚和 MISO 引脚这 3 个引脚进行通信。在从属模式中 (SPCR.MSTR=1) 进行时钟同步运行时, 能将 RSPI 命令寄存器 m (SPCMDm) (m=0 ~ 7) 的 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, 必须将 SPCMDm.CPHA 位置 “1”, 如果置 “0”, 就无法保证运行。

### 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 “26.3.6 通信运行模式”)。

如果将 TXMD 位置 “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 “26.3.8 错误检测”)。RSPI 根据 MODFEN 位和 MSTR 位的组合, 决定 SSL0 ~ SSL3 引脚的输入 / 输出方向 (参照 “26.3.2 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 RSPCK、MOSI、MISO 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 位置“1”或者在 RSPI 检测到溢出错误后将 SPSR.OVRF 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“26.3.8 错误检测”。

**SPTIE 位 (RSPI 发送中断允许位)**

在 RSPI 检测到发送缓冲器空时, 此位允许或者禁止 RSPI 发送中断请求的产生。

如果在和 SPTIE 位同时或者在 SPTIE 位之后将 SPE 位置“1”, 就产生发送开始时的发送中断请求。

必须注意: 即使转移到 RSPI 禁止状态 (SPE 位为“0”), 只要将 SPTIE 位置“1”, 也会产生 RSPI 发送中断。

**SPE 位 (RSPI 功能有效位)**

此位选择 RSPI 功能的有效或者无效。

当 SPSR.MODF 标志为“1”时, 不能将 SPE 位置“1”。详细内容请参照“26.3.8 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“26.3.9 RSPI 的初始化”。

**SPRIE 位 (RSPI 接收中断允许位)**

当 RSPI 在串行传送结束后检测到接收缓冲器的写操作时, 此位允许或者禁止 RSPI 接收中断请求的产生。



## 26.2.2 RSPI 从属选择极性寄存器 (SSLP)

地址 0008 8381h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	SSL0P	SSL0 信号极性设定位	0: SSL0 信号为“0”时有效 1: SSL0 信号为“1”时有效	R/W
b1	SSL1P	SSL1 信号极性设定位	0: SSL1 信号为“0”时有效 1: SSL1 信号为“1”时有效	R/W
b2	SSL2P	SSL2 信号极性设定位	0: SSL2 信号为“0”时有效 1: SSL2 信号为“1”时有效	R/W
b3	SSL3P	SSL3 信号极性设定位	0: SSL3 信号为“0”时有效 1: SSL3 信号为“1”时有效	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

SSLP 寄存器设定 RSPI 的 SSL0 ~ SSL3 信号的极性。

如果 CPU 在 RSPI 功能有效 (SPCR.SPE 位为“1”) 的状态下改写 SSLP 寄存器, 就无法保证以后的运行。

## SSLiP 位 (SSL 信号极性设定位) (i=0 ~ 3)

此位设定 SSLi 信号的极性。

SSLiP 的设定值表示 SSLi 信号的有效极性 (i=0 ~ 3)。

### 26.2.3 RSPI 引脚控制寄存器 (SPPCR)

地址 0008 8382h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
复位后的值	0	0	0	0	0	0	0

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 空闲固定值为“0” 1: MOSI 空闲固定值为“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

SPPCR 寄存器是设定 RSPI 引脚模式的寄存器。如果在 RSPI 功能有效 (SPCR.SPE 位为“1”) 的状态下改写 SPPCR 寄存器, 就无法保证以后的运行。

#### 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 MISO 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 就切断 MOSI 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)。详细内容请参照“26.3.13 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 MISO 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”时, RSPI 就切断 MOSI 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 2)。详细内容请参照“26.3.13 环回模式”。

#### MOIFV 位 (MOSI 空闲固定值位)

在主控模式中, 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 MOSI 引脚输出值。

#### MOIFE 位 (MOSI 空闲值固定允许位)

主控模式的 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 MOSI 的输出值。当 MOIFE 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 MOSI; 当 MOIFE 为“1”时, RSPI 将 MOIFV 位设定的固定值输出到 MOSI。

## 26.2.4 RSPI 状态寄存器 (SPSR)

地址 0008 8383h

b7	b6	b5	b4	b3	b2	b1	b0
SPRF	—	SPTEF	—	PERF	MODF	IDLNF	OVRF
0	0	1	0	0	0	0	0

复位后的值

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	SPTEF	发送缓冲器空标志	0: 将数据传送到 SPDR 寄存器时 (发送缓冲器有数据) 1: 将数据从 SPDR 寄存器传送到移位寄存器时 (发送缓冲器无数据)	R/(W) (注2)
b6	—	保留位	读写值都为“0”。	R/W
b7	SPRF	接收缓冲器满标志	0: 将数据传送到 SPDR 寄存器时 (SPDR 没有有效的接收数据) 1: 接收正常结束并且将数据从移位寄存器传送到 SPDR 寄存器时 (SPDR 有有效的接收数据)	R/(W) (注2)

注1. 为了清除标志, 只能在读“1”后写“0”。

注2. 只能写“1”。

SPSR 寄存器保存表示 RSPI 运行状态的标志。CPU 能随时读 SPSR 寄存器。CPU 写 SPSR 寄存器的操作只有在一定的条件下才有效。

## 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]

- CPU 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】**

- 下述[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】**

- SPCR.SPE 位为“1” (RSPI 功能有效) 时

[ 为“0”的条件 ]

**【主控模式】**

1. SPCR.SPE 位为“0” (RSPI 的初始化) 时
2. 发送缓冲器 (SPTX) 为空 (未设定下一个传送数据) 时
3. SPSSR.SPCP[2:0] 位为“000b” (顺序控制的开头) 时
4. 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)

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】**

- 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, SSL 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 SSL 引脚无效并且 RSPI 检测到模式故障错误时

SSL 信号的有效电平取决于 SSLP.SSLiP 位 (SSL 信号极性设定位) (i=0 ~ 3)。

[ 为“0”的条件 ]

- CPU 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 ]

- CPU 在 PERF 标志为“1”的状态下读 SPSR 寄存器后给 PERF 标志写“0”时

**SPTEF 标志 (发送缓冲器空标志)**

此标志表示 SPDR 寄存器的发送缓冲器是否有数据。

[ 为“1”的条件 ]

- 将数据从 SPDR 寄存器传送到移位寄存器时

[ 为“0”的条件 ]

- 将发送数据传送到 SPDR 寄存器时

**SPRF 标志 (接收缓冲器满标志)**

此标志表示 SPDR 寄存器的接收缓冲器是否有数据。

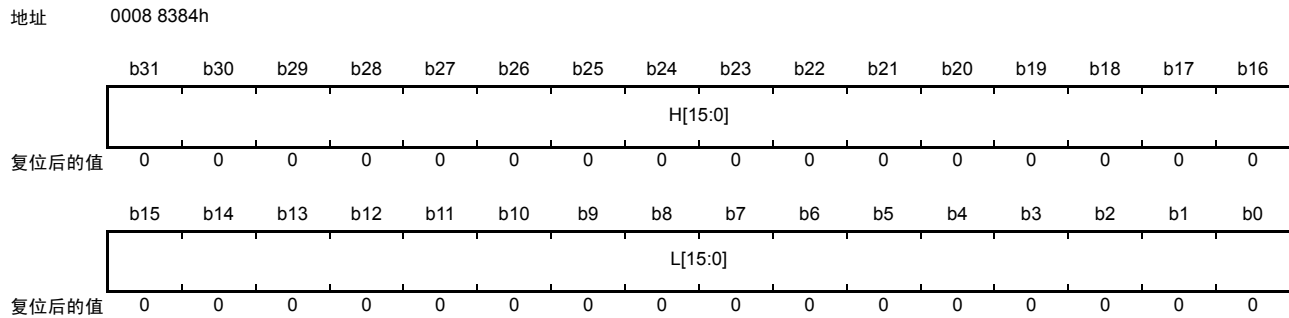
[ 为“1”的条件 ]

- 接收运行正常结束并且将接收数据从移位寄存器传送到 SPDR 寄存器时

[ 为“0”的条件 ]

- 从 SPDR 寄存器传送数据时

### 26.2.5 RSPI 数据寄存器 (SPDR)



SPDR 寄存器是 CPU 能随时读写的寄存器，是用于保存 RSPI 发送 / 接收数据的缓冲器。

发送缓冲器 (SPTX) 和接收缓冲器 (SPRX) 是独立的缓冲器，这些缓冲器被映像到 SPDR 寄存器。

必须通过设定 SPDCR.SPLW 位，用字或者长字读写 SPDR 寄存器。当 SPLW 位为“0”时，SPDR 寄存器是 64 位缓冲器，由最多 16 位的 4 帧构成；当 SPLW 位为“1”时，SPDR 寄存器是 128 位缓冲器，由最多 32 位的 4 帧构成。

帧数设定位 SPDCR.SPFC[1:0] 位决定 SPDR 寄存器所使用的帧长，SPCMDm.SPB[3:0] 位决定所使用的位长。

在写 SPDR 寄存器时，当发送缓冲器 (SPTX) 为空 (未设定下次传送的数据) 时，RSPI 将数据写到 SPDR 寄存器的发送缓冲器。在发送缓冲器有未发送数据的状态下，RSPI 不更新 SPDR 寄存器的发送缓冲器。

在读 SPDR 寄存器时，如果 SPDCR.SPRDTD 位为“0”，就读接收缓冲器；如果 SPDCR.SPRDTD 位为“1”，就读发送缓冲器。

在读发送缓冲器时，能读到刚写的值。在发送缓冲器有未发送数据的状态下，读取值全部为“0”。

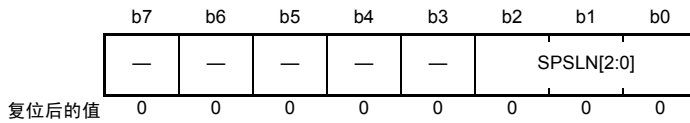
通常使用的方法是：假设 SPRDTD 位为“0”，通过接收缓冲器满中断读接收缓冲器。在接收缓冲器有未读数据或者 SPSR.OVRF 标志为“1”的状态下，RSPI 在串行传送结束时不更新 SPDR 寄存器的接收缓冲器。

在以字或者长字的存取宽度读写 SPDR 寄存器时，必须存取下述地址，否则无法保证读写时的数据。

- 长字：RSPI.SPDR 0008 8384h
- 字：RSPI.SPDR 0008 8384h

## 26.2.6 RSPI 顺序控制寄存器 (SPSCR)

地址 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 ~ SPCMD7 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 ~ SPCMD7 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 寄存器设定 RSPI 主控运行时的顺序控制方式。在 SPCR.MSTR 位和 SPCR.SPE 位为“1”并且允许主控模式的 RSPI 功能的情况下, 必须在 SPSR.IDLNF 标志为“0”的状态下改写 SPSCR.SPSLN[2:0] 位。

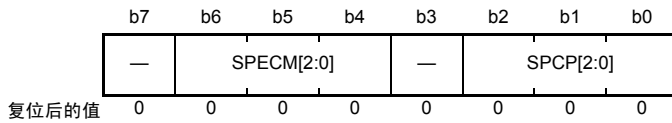
## SPSLN[2:0] 位 (RSPI 顺序长度设定位)

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 RSPI 命令寄存器 0 ~ 7 (SPCMD0 ~ SPCMD7) 和参照顺序。

从属模式的 RSPI 能随时参照 SPCMD0 寄存器。

## 26.2.7 RSPI 顺序状态寄存器 (SPSSR)

地址 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”，写操作无效。	R/W
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”，写操作无效。	R/W

SPSSR 寄存器表示 RSPI 主控运行时的顺序控制状态。

CPU 对 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 SPCMD0 ~ SPCMD7 寄存器。

有关 RSPI 的顺序控制，请参照“26.3.10.1 主控模式的运行”。

### SPECM[2:0] 位 (RSPI 错误命令位)

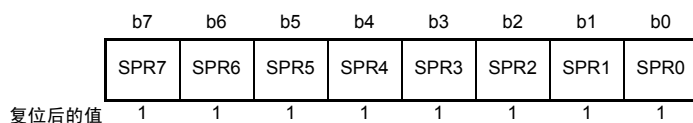
这些位通过 RSPI 顺序控制，表示在检测到错误时 SPCP[2:0] 位指定的 SPCMD0 ~ SPCMD7 寄存器。只有在检测到错误时，RSPI 才能更新 SPECM[2:0] 位。如果 SPSR.OVRF 位、SPSR.MODF 标志和 SPSR.PERF 标志都为“0”并且没有发生错误，SPECM[2:0] 位的值就没有含义。

有关 RSPI 的错误检测功能，请参照“26.3.8 错误检测”；有关 RSPI 的顺序控制，请参照“26.3.10.1 主控模式的运行”。



## 26.2.8 RSPI 位速率寄存器 (SPBR)

地址 0008 838Ah



SPBR 寄存器是 CPU 能读写的寄存器，用于设定主控模式时的位速率。在 SPCR.MSTR 位和 SPCR.SPE 位都为“1”并且允许主控模式的 RSPI 运行的状态下，如果 CPU 改写 SPBR 寄存器，就无法保证以后的运行。

在从属模式中使用 RSPI 时，与 SPBR 寄存器和 SPCMDm.BRDV[1:0] 位（位速率分频设定值）（m=0 ~ 7）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 26.4 所示。

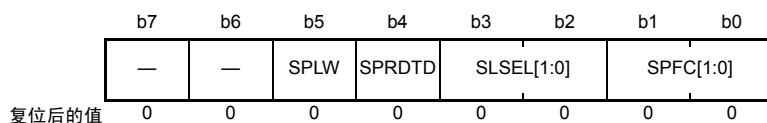
表 26.4 SPBR 寄存器、BRDV[1:0] 位的设定值和位速率

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 (注1)	18.0Mbps (注1)	20.0Mbps (注1)	25.0Mbps (注1)
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

注 1. 能设定，但是必须满足电特性的条件。

## 26.2.9 RSPI 数据控制寄存器 (SPDCR)

地址 0008 838Bh



位	符号	位名	功能	R/W																						
b1-b0	SPFC[1:0]	帧数设定位	设定 SPDR 寄存器能保存的帧数。 SPDR 寄存器能保存的帧结构以及发送 / 接收设定的组合例子如表 26.5 和图 26.2 所示。 如果设定上述组合例子中没有的组合, 就无法保证以后的运行。	R/W																						
b3-b2	SLSEL[1:0]	SSL 引脚输出选择位	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <thead> <tr> <th></th> <th>SLSEL [1:0]=00b</th> <th>SLSEL [1:0]=01b</th> <th>SLSEL [1:0]=10b</th> <th>SLSEL [1:0]=11b</th> </tr> </thead> <tbody> <tr> <td>SSL3</td> <td>输出</td> <td>IO</td> <td>IO</td> <td rowspan="4">禁止设定</td> </tr> <tr> <td>SSL2</td> <td>输出</td> <td>IO</td> <td>IO</td> </tr> <tr> <td>SSL1</td> <td>输出</td> <td>IO</td> <td>输出</td> </tr> <tr> <td>SSL0</td> <td>输出</td> <td>输出</td> <td>输出</td> </tr> </tbody> </table> <p>如果将SLSEL[1:0]置“11b”, 就无法保证以后的运行。</p>		SLSEL [1:0]=00b	SLSEL [1:0]=01b	SLSEL [1:0]=10b	SLSEL [1:0]=11b	SSL3	输出	IO	IO	禁止设定	SSL2	输出	IO	IO	SSL1	输出	IO	输出	SSL0	输出	输出	输出	R/W
	SLSEL [1:0]=00b	SLSEL [1:0]=01b	SLSEL [1:0]=10b	SLSEL [1:0]=11b																						
SSL3	输出	IO	IO	禁止设定																						
SSL2	输出	IO	IO																							
SSL1	输出	IO	输出																							
SSL0	输出	输出	输出																							
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																						

SPDCR 寄存器设定 SPDR 寄存器能保存的帧数、SSL 引脚的输出控制、SPDR 寄存器的读操作以及将 SPDR 寄存器的存取宽度设定为长字存取或者字存取。

能通过 SPCMDm.SPB[3:0] 位 (m=0~7)、SPSCR.SPSSLN[2:0] 位和 SPDCR.SPFC[1:0] 位的组合, 启动 1 次发送 / 接收进行最多 4 帧的发送 / 接收。

在 SPCR.SPE 位为“1”并且允许 RSPI 运行的状态下, 如果 CPU 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 寄存器能保存的帧数。能通过设定 SPCMDm.SPB[3:0] 位、SPSCR.SPSSLN[2:0] 位和 SPDCR.SPFC[1:0] 位, 启动 1 次发送 / 接收进行最多 4 帧的发送 / 接收。SPFC[1:0] 对发生 RSPI 接收缓冲器满中断时的接收数据量进行设定。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 26.5 和图 26.2 所示。如果设定组合例子中没有的组合, 就无法保证以后的运行。

## SLSEL[1:0] 位 (SSL 引脚输出选择位)

SLSEL[1:0] 位控制主控模式时的 SSL 引脚的输出。

**SPRDTD 位 (RSPI 接收 / 发送数据选择位)**

此位选择将 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。  
在读发送缓冲器时，能读到刚写入 SPDR 寄存器的值。

**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，就无法保证运行。

表 26.5 SPCMDm.SPB[3:0] 位能设定的数据长度

设定	SPB[3:0]	SPSLN[2:0]	SPFC[1:0]	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	N	000b	00b	1	1
1-2	N	000b	01b	2	2
1-3	N	000b	10b	3	3
1-4	N	000b	11b	4	4
2-1	N、M	001b	01b	2	2
2-2	N、M	001b	11b	4	4
3	N、M、O	010b	10b	3	3
4	N、M、O、P	011b	11b	4	4
5	N、M、O、P、Q	100b	00b	5	1
6	N、M、O、P、Q、R	101b	00b	6	1
7	N、M、O、P、Q、R、S	110b	00b	7	1
8	N、M、O、P、Q、R、S、T	111b	00b	8	1

N、M、O、P、Q、R、S、T: SPCMDm.SPB[3:0] 位能设定的数据长度

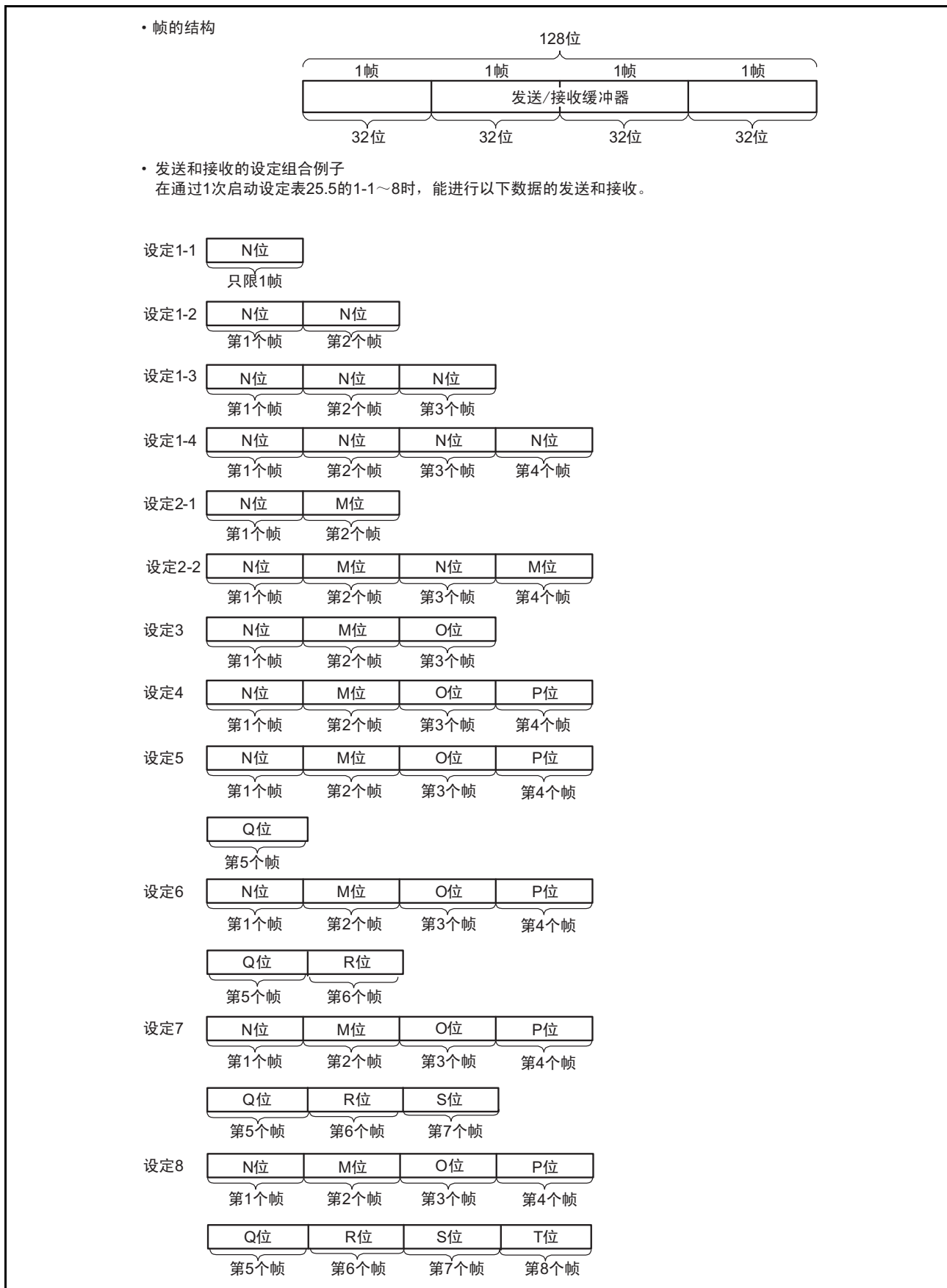
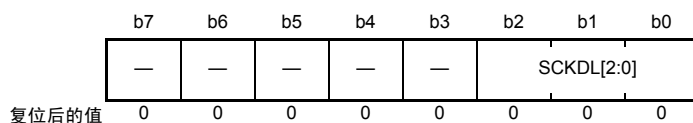


图 26.2 帧的结构和发送 / 接收设定的组合例子

## 26.2.10 RSPI 时钟延迟寄存器 (SPCKD)

地址 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 SSL 信号有效开始到 RSPCK 振荡的期间 (RSPCK 延迟)。在 SPCR.MSTR 位和 SPCR.SPE 位都为“1”并且主控模式的 RSPI 运行有效的状态下，如果改写 SPCKD 寄存器，就无法保证以后的运行。

在从属模式中使用 RSPI 时，必须将 SCKDL[2:0] 位置“000b”。

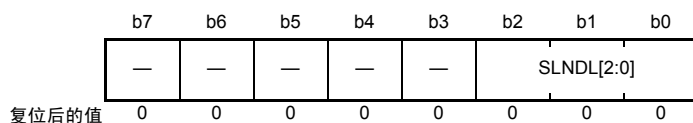
## SCKDL[2:0] 位 (RSPCK 延迟设定位)

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”。

## 26.2.11 RSPI 从属选择无效延迟寄存器 (SSLND)

地址 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 SSL 信号置为无效的期间 (SSL 无效延迟)。在 SPCR.MSTR 位和 SPCR.SPE 位都为“1”并且主控模式的 RSPI 运行有效的状态下, 如果改写 SSLND 寄存器, 就无法保证以后的运行。

在从属模式中使用 RSPI 时, 必须将 SLNDL[2:0] 位置“000b”。

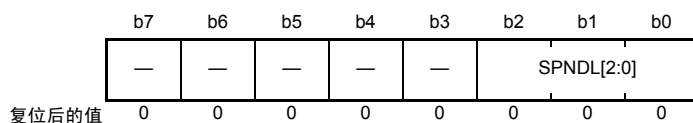
## SLNDL[2:0] 位 (SSL 无效延迟设定位)

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”。

## 26.2.12 RSPI 下次存取延迟寄存器 (SPND)

地址 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 SSL 信号的无效期间（下次存取延迟）。在 SPCR.MSTR 位和 SPCR.SPE 位都为“1”并且主控模式的 RSPI 运行有效的状态下，如果改写 SPND 寄存器，就无法保证以后的运行。

在从属模式中使用 RSPI 时，必须将 SPNDL[2:0] 位置“000b”。

## 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”。

## 26.2.13 RSPI 控制寄存器 2 (SPCR2)

地址 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	奇偶校验有效位	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

SPCR2 寄存器是设定 RSPI 运行模式的寄存器。在 SPCR.SPPE 位为“1”并且 RSPI 运行有效的状态下, 如果改写 SPCR2.SPPE 位或者 SPCR2.SPOE 位的设定值, 就无法保证以后的运行。

## SPPE 位 (奇偶校验有效位)

此位选择奇偶校验功能的有效或者无效。

当 SPCR.TXMD 为“0”并且 SPPE 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 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)

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 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是否正常而将奇偶校验电路的自诊断置为有效的位。



## 26.2.14 RSPI 命令寄存器 m (SPCMDm) (m=0 ~ 7)

地址 SPCMD0 0008 8390h、SPCMD1 0008 8392h、SPCMD2 0008 8394h、SPCMD3 0008 8396h  
SPCMD4 0008 8398h、SPCMD5 0008 839Ah、SPCMD6 0008 839Ch、SPCMD7 0008 839Eh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]			SSLKP	SSLA[2:0]		BRDV[1:0]		CPOL	CPHA		
0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1

复位后的值

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“0” 1: 空闲时的 RSPCK 为“1”	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 上述以外: 不能设定	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 0 1 0 0: 8 位 0 1 0 1: 8 位 0 1 1 0: 8 位 0 1 1 1: 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 0 0 1 0: 32 位 0 0 1 1: 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。SPCMD0 寄存器的一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。在从属模式的 RSPI 运行有效的状态下, 如果改写 SPCMDm 寄存器, 就无法保证以后的运行。

#### 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“26.2.8 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 SSL 信号的有效。SSLA[2:0] 位的设定值控制 SSL<sub>i</sub> (i=0 ~ 3) 信号的有效。SSL 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<sub>i</sub> 信号无效的状态下进行串行传送 (SSL0 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)

在从属模式的 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 SSL 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 SSL 信号无效到下次存取的 SSL 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 SSL 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 SSLi 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 SSL 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”。

## 26.3 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。

### 26.3.1 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 26.6 所示。

表 26.6 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
RSPCK 信号	输入	输出	输出 /Hi-Z	输入	输出
MOSI 信号	输入	输出	输出 /Hi-Z	输入	输出
MISO 信号	输出 /Hi-Z	输入	输入	输出	输入
SSL0 信号	输入	输出	输入	Hi-Z	Hi-Z
SSL1 ~ SSL3 信号	Hi-Z	输出 /Hi-Z	输出 /Hi-Z	Hi-Z	Hi-Z
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1)	有 (注1)	有 (注1)	有 (注1)	有 (注1)
溢出错误检测	有 (注1)	有 (注1)	有 (注1)	有 (注1)	有 (注1)
奇偶校验错误检测	有 (注1、注2)	有 (注1、注2)	有 (注1、注2)	有 (注1、注2)	有 (注1、注2)
模式故障错误检测	有 (MODFEN=1)	无	有	无	无

注1. 当 SPCR.TXMD 位为“1”时，不进行接收缓冲器满检测、溢出错误检测和奇偶校验错误检测。

注2. 当 SPCR2.SPPE 位为“0”时，不进行奇偶校验错误检测。

### 26.3.2 RSPI 引脚的控制

根据 SPCR.MSTR 位、SPCR.MODFEN 位和 SPCR.SPMS 位的设定，RSPI 自动转换引脚方向和输出模式。引脚状态和各位设定值的关系如表 26.7 所示。

表 26.7 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1)
单主控 (SPI 运行) (MSTR=1, MODFEN=0, SPMS=0)	RSPCK	输出
	SSL0 ~ 3	输出
	MOSI	输出
	MISO	输入
多主控 (SPI 运行) (MSTR=1, MODFEN=1, SPMS=0)	RSPCK (注2)	输出 /Hi-Z
	SSL0	输入
	SSL1 ~ 3 (注2)	输出 /Hi-Z
	MOSI (注2)	输出 /Hi-Z
从属 (SPI 运行) (MSTR=0, SPMS=0)	RSPCK	输入
	SSL0	输入
	SSL1 ~ 3	Hi-Z
	MOSI	输入
主控 (时钟同步运行) (MSTR=1, MODFEN=0, SPMS=1)	RSPCK	输出
	SSL0 ~ 3 (注4)	Hi-Z
	MOSI	输出
	MISO	输入
从属 (时钟同步运行) (MSTR=0, SPMS=1)	RSPCK	输入
	SSL0 ~ 3 (注4)	Hi-Z
	MOSI	输入
	MISO	输出

注 1. 未选择 RSPI 功能的多功能引脚不反映 RSPI 的设定值。

注 2. 当 SSL0 为有效电平时，引脚状态为 Hi-Z。

注 3. 当 SSL0 为无效电平或者 SPCR.SPE 位为“0”时，引脚状态为 Hi-Z。

注 4. 在时钟同步运行时，能将 SSL0 ~ 3 用作 I/O 端口。

如表 26.8 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、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 26.8 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	上次传送的最后数据
	1	
1	0	总是为“0”。
1	1	总是为“1”。

### 26.3.3 RSPI 系统结构例子

#### 26.3.3.1 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 26.3 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 SSL0 ~ SSL3 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 RSPCK 和 MOSI，RSPI 从属设备随时驱动 MISO。

注 1. 在类似 SPCMDm.CPHA 位 (m=0 ~ 7) 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 SSL 输出连接到从属设备的 SSL 输入。

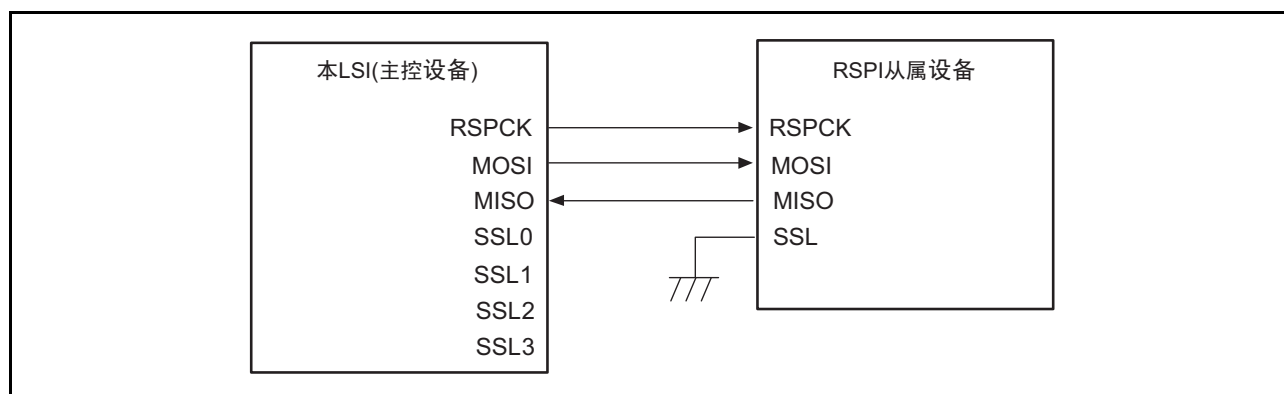


图 26.3 单主控模式 / 单从属模式的系统结构例子 (本 LSI= 主控设备)

## 26.3.3.2 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 26.4 所示。在将本 LSI 用作从属设备时，将 SSL0 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 MISO (注 1)。

在 SPCMD0.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 SSL0 输入引脚固定为低电平，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 26.5)。

注 1. 在 SSL0 为无效电平时，引脚状态为 Hi-Z。

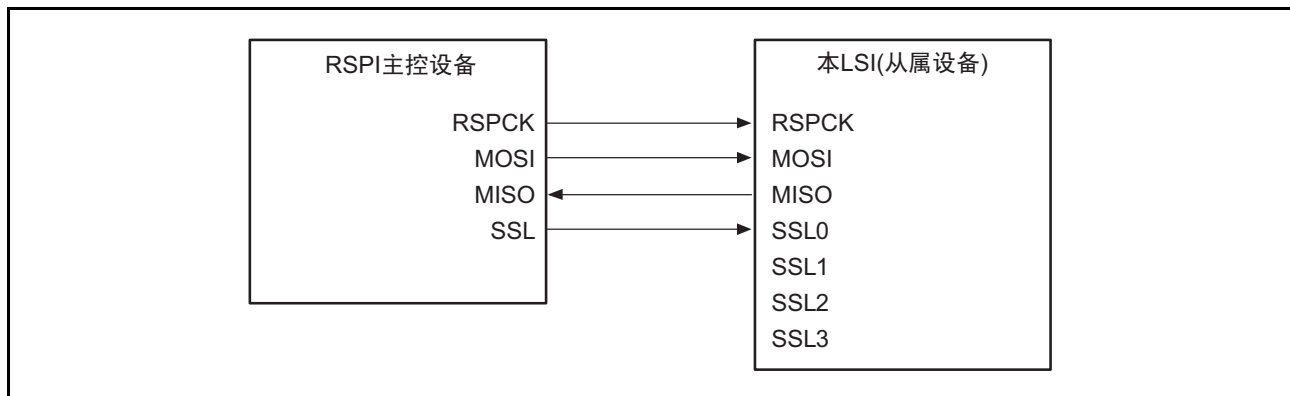


图 26.4 单主控模式 / 单从属模式的系统结构例子 (本 LSI= 从属设备)

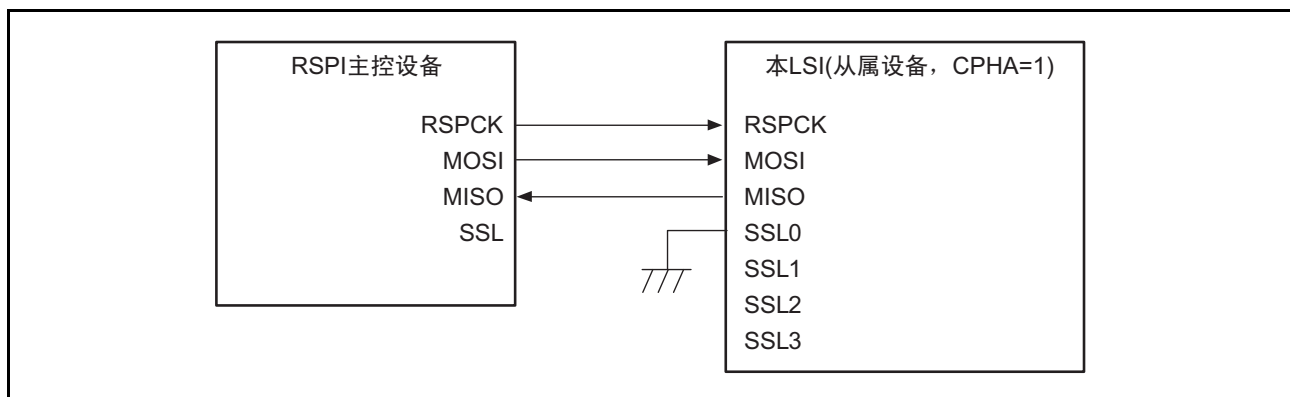


图 26.5 单主控模式 / 单从属模式的系统结构例子 (本 LSI= 从属设备, SPCMD0.CPHA=1)

## 26.3.3.3 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 26.6 所示。在图 26.6 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 RSPCK 输出和 MOSI 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 MISO 输入，本 LSI (主控设备) 的 SSL0 ~ SSL3 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 SSL0 ~ SSL3。在 RSPI 从属设备 0 ~ 从属设备 3 中，SSL 输入为低电平的从属设备驱动 MISO。

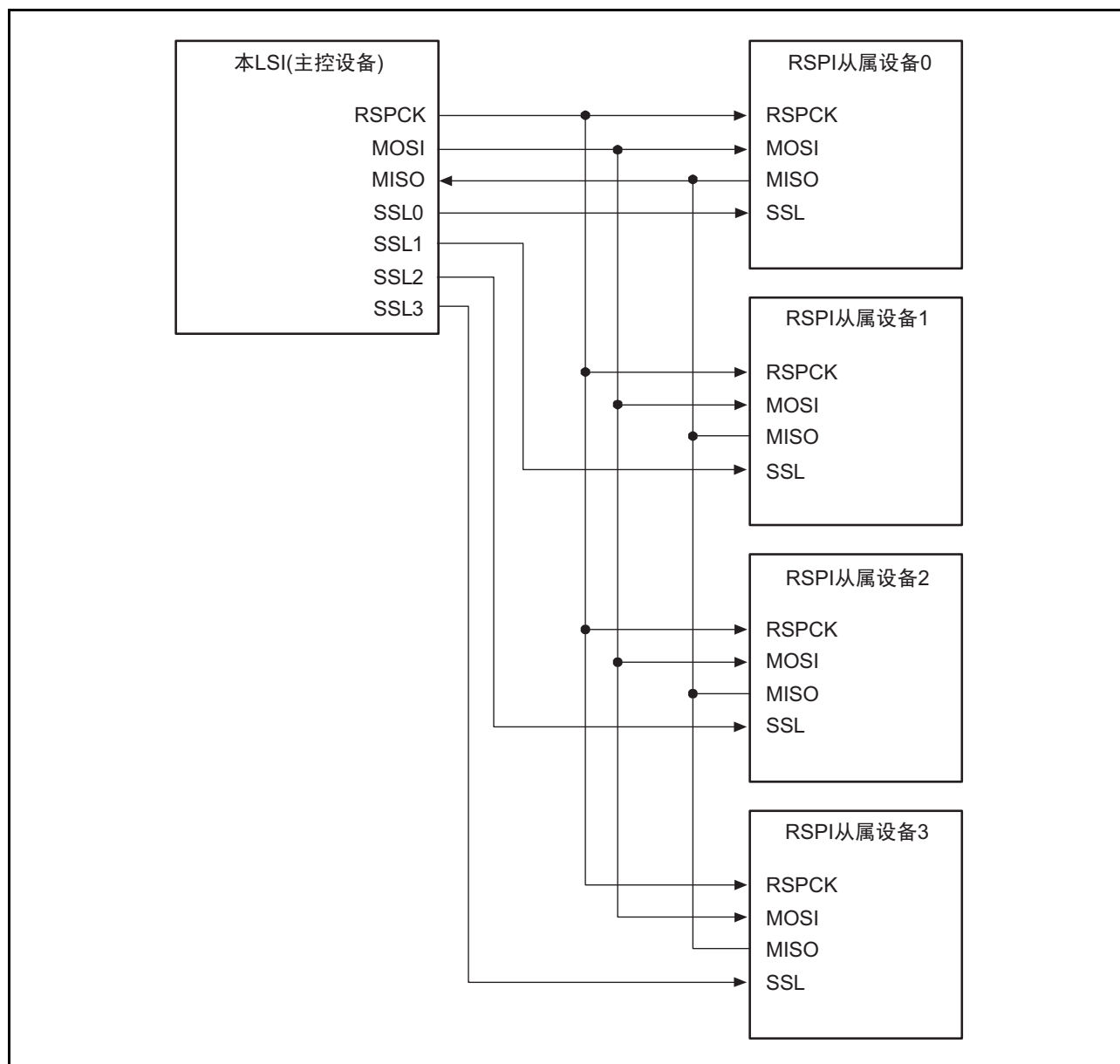


图 26.6 单主控模式 / 多从属模式的系统结构例子 (本 LSI= 主控设备)



## 26.3.3.4 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 26.7 所示。在图 26.7 的例子中，RSPI 系统由 RSPI 主控设备和 2 个本 LSI (从属设备 X 和从属设备 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 和从属设备 Y) 的 RSPCK 输入和 MOSI 输入，本 LSI (从属设备 X 和从属设备 Y) 的 MISO 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 和从属设备 Y) 的 SSL0 输入。

RSPI 主控设备随时驱动 RSPCK、MOSI、SSLX 和 SSLY。在本 LSI (从属设备 X 和从属设备 Y) 中，SSL0 输入为低电平的从属设备驱动 MISO。

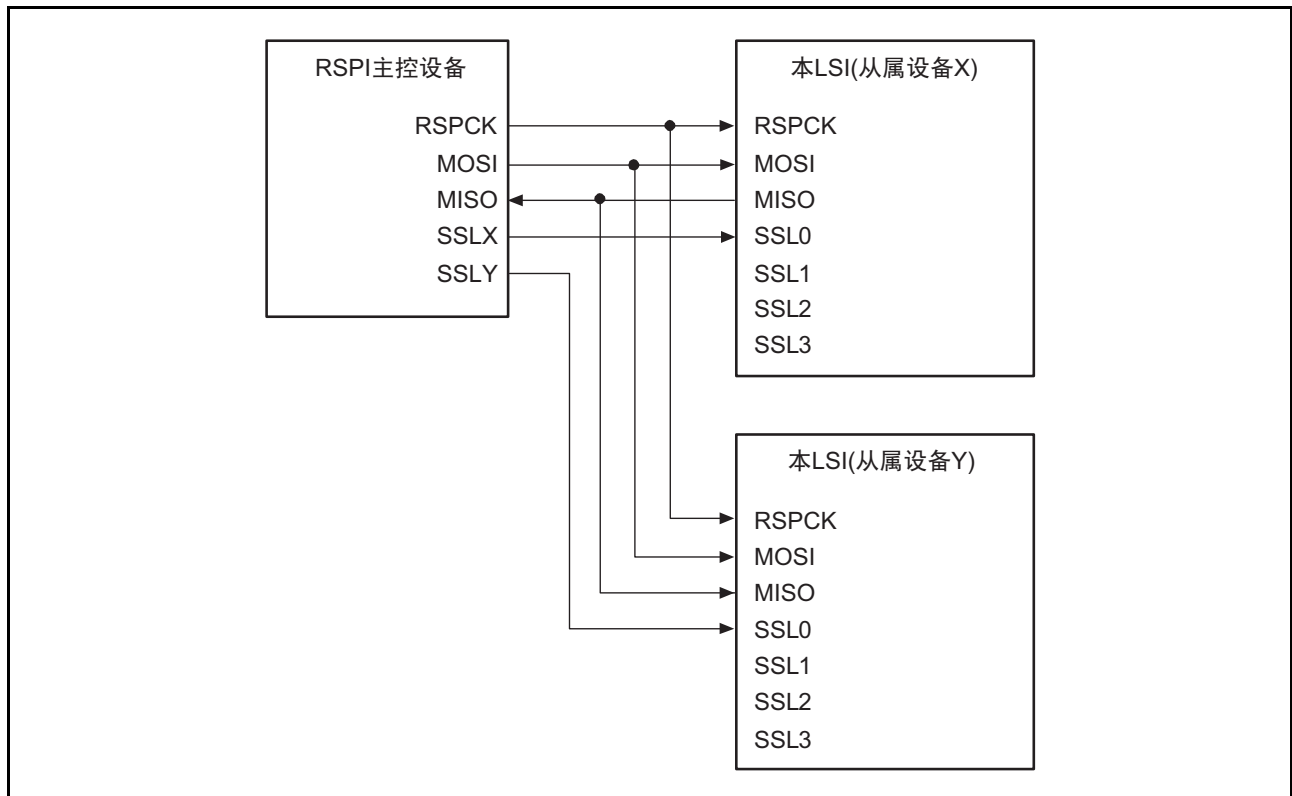


图 26.7 单主控模式 / 多从属模式的系统结构例子 (本 LSI= 从属设备)

## 26.3.3.5 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 26.8 所示。在图 26.8 的例子中，RSPI 系统由 2 个本 LSI (主控设备 X 和主控设备 Y) 和 2 个 RSPI 从属设备 (RSPI 从属设备 1 和从属设备 2) 构成。

本 LSI (主控设备 X 和主控设备 Y) 的 RSPCK 输出和 MOSI 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 和主控设备 Y) 的 MISO 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 SSL0 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 SSL0 输入，本 LSI (主控设备 X 和主控设备 Y) 的 SSL1 输出和 SSL2 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 SSL0 输入、连接从属设备的 SSL1 输出和 SSL2 输出构成，所以不使用本 LSI 的 SSL3 输出。

当 SSL0 输入电平为“1”时，本 LSI 驱动 RSPCK、MOSI、SSL1 和 SSL2；当 SSL0 输入电平为“0”时，检测模式故障错误，在将 RSPCK、MOSI、SSL1 和 SSL2 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。

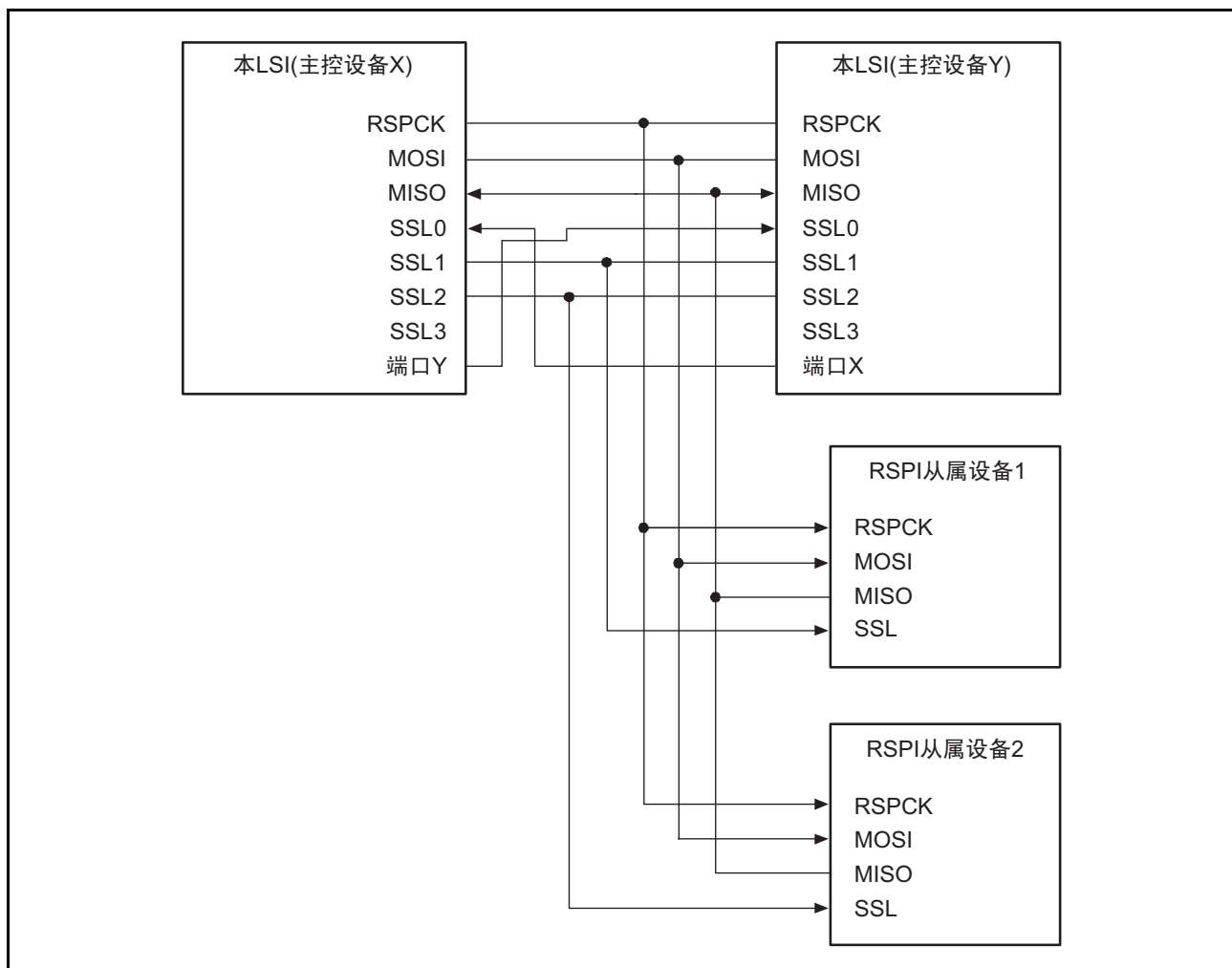


图 26.8 多主控模式 / 多从属模式的系统结构例子 (本 LSI= 主控设备)

### 26.3.3.6 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 26.9 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 SSL0 ~ SSL3。

本 LSI (主控设备) 随时启动 RSPCK 和 MOSI, RSPI 从属设备随时驱动 MISO。

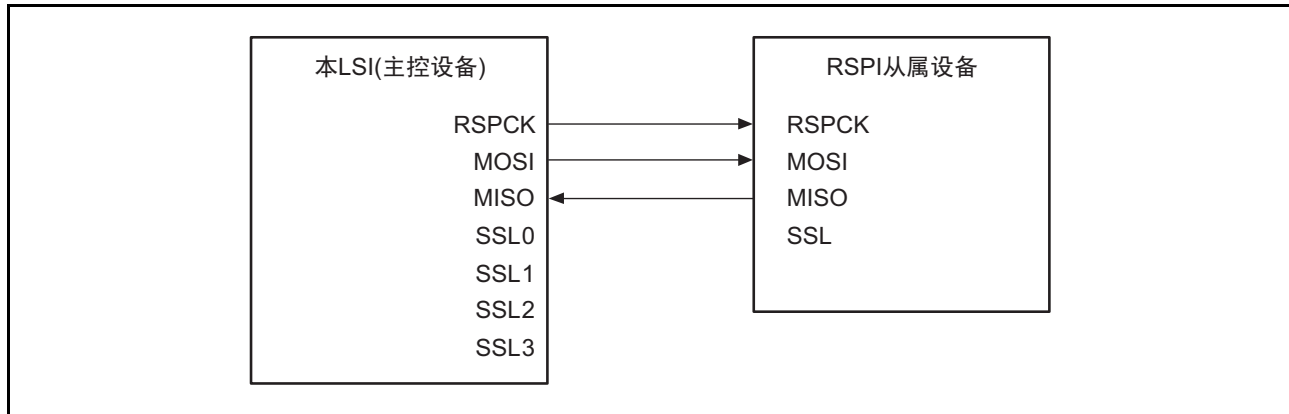


图 26.9 主控模式 (时钟同步运行) / 从属模式 (时钟同步运行) 的系统结构例子 (本 LSI= 主控设备)

### 26.3.3.7 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 26.10 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 MISO, RSPI 主控设备随时驱动 RSPCK 和 MOSI。不使用本 LSI (从属设备) 的 SSL0 ~ SSL3。

只有在将 SPCMD0.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。

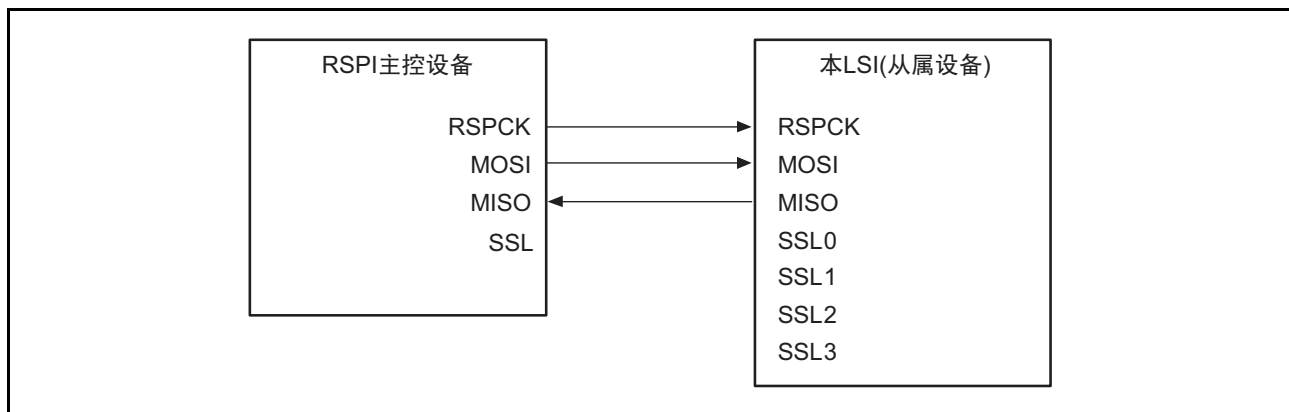


图 26.10 主控模式 (时钟同步运行) / 从属模式 (时钟同步运行) 的系统结构例子 (本 LSI= 从属设备, SPCMD0.CPHA=1)

### 26.3.4 传送格式

#### 26.3.4.1 SPCMDm.CPHA 位为“0”的情况 (m=0 ~ 7)

在 SPCMDm.CPHA 位 (m=0 ~ 7) 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 26.11 所示。但是，不保证 RSPI 为从属模式 (SPCR.MSTR=0) 并且 SPCMDm.CPHA 位为“0”时的时钟同步运行 (SPCR.SPMS 位为“1”)。在图 26.11 中，RSPCK (CPOL=0) 是 SPCMDm.CPOL 位为“0”时的 RSPCK 信号波形，RSPCK (CPOL=1) 是 SPCMDm.CPOL 位为“1”时的 RSPCK 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 的设定，详细内容请参照“26.3.2 RSPI 引脚的控制”。

当 SPCMDm.CPHA 位为“0”时，在 SSL 信号的有效时序开始驱动 MOSI 信号和 MISO 信号的有效数据。在 SSL 信号有效后发生的第一个 RSPCK 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。MOSI 信号和 MISO 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。SPCMDm.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 SSL 信号有效到 RSPCK 振荡的期间 (RSPCK 延迟)，t2 表示从 RSPCK 振荡停止到 SSL 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 SSL 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“26.3.10.1 主控模式的运行”。

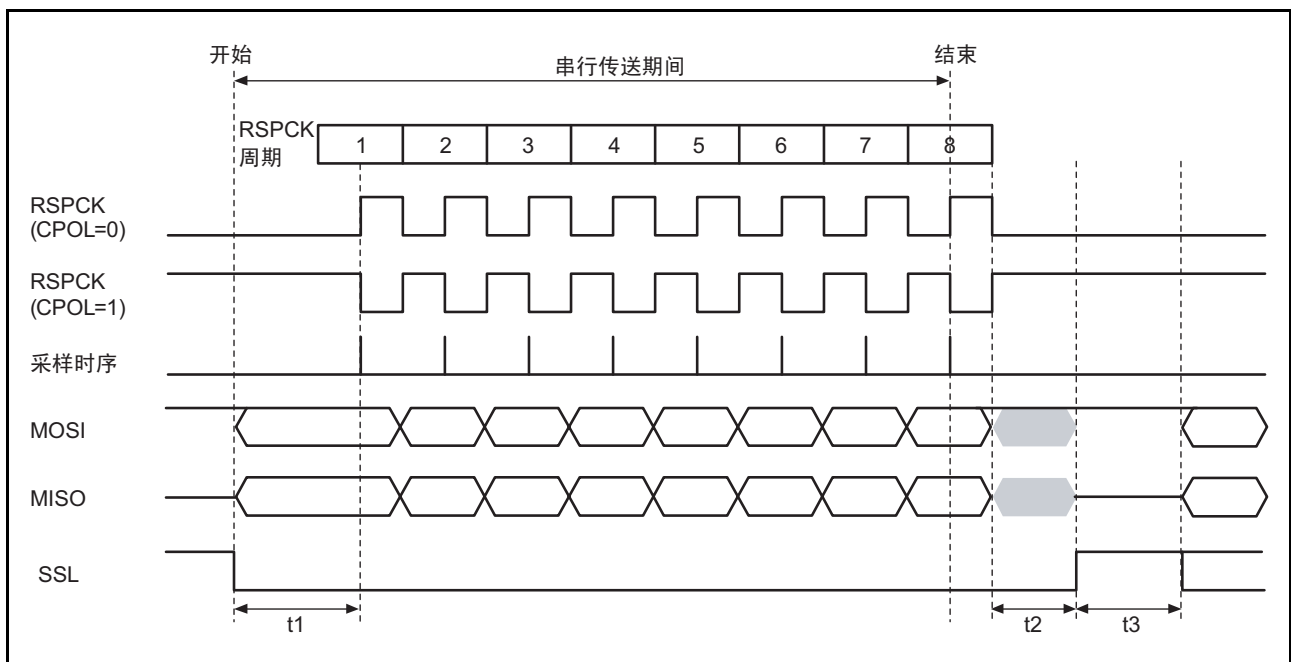


图 26.11 RSPI 传送格式 (CPCMDm.CPHA 位 = 0)

26.3.4.2 CPCMDm.CPHA 位为“1”的情况 (m=0 ~ 7)

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 26.12 所示。但是，当 SPCR.SPMS 位为“1”时，不使用 SSL 信号而只使用 RSPCK 信号、MOSI 信号和 MISO 信号进行通信。在图 26.12 中，RSPCK (CPOL=0) 是 SPCMDm.CPOL 位为“0”时的 RSPCK 信号波形，RSPCK (CPOL=1) 是 SPCMDm.CPOL 位为“1”的 RSPCK 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“26.3.2 RSPI 引脚的控制”。

当 SPCMDm.CPHA 位为“1”时，在 SSL 信号的有效时序开始驱动 MISO 信号的无效数据。在 SSL 信号有效后发生的第一个 RSPCK 信号变化时序，开始输出 MOSI 信号和 MISO 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 RSPCK 信号的运行时序而只影响信号的极性。

t1、t2、t3 的内容和 SPCMDm.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“26.3.10.1 主控模式的运行”。

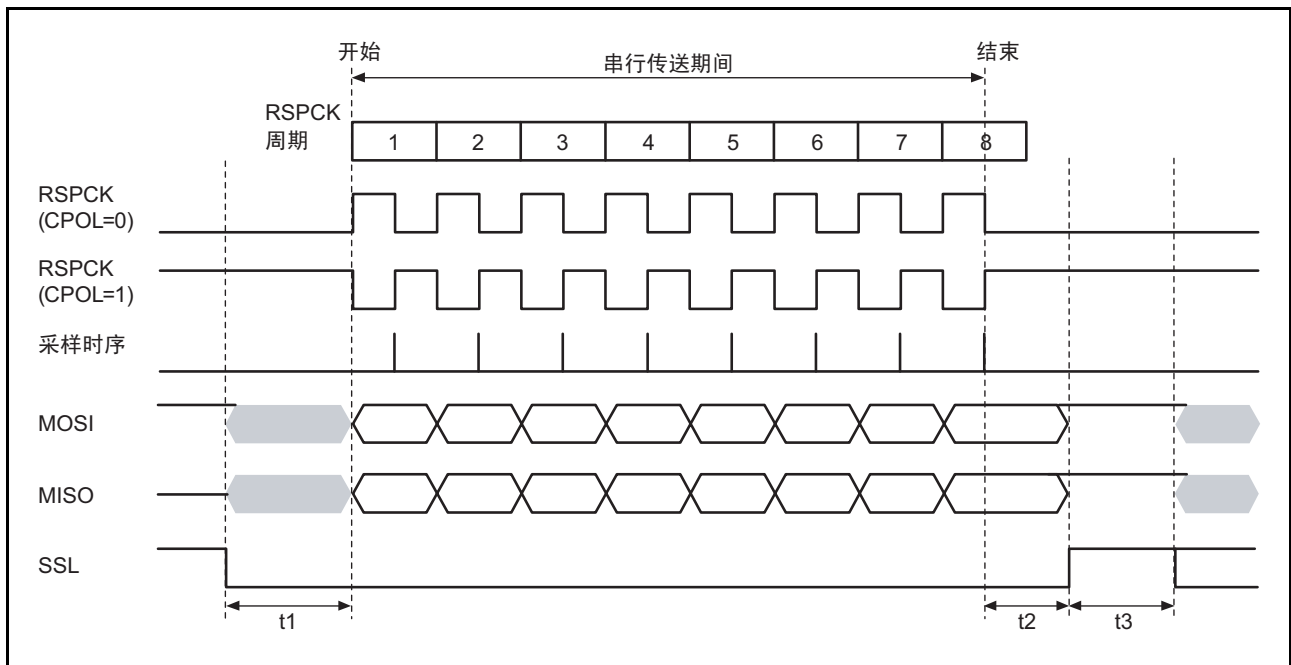


图 26.12 RSPI 传送格式 (CPCMDm.CPHA 位 =1)

### 26.3.5 数据格式

RSPI 的数据格式取决于 SPCMDm 寄存器 (m=0 ~ 7) 和 SPCR2.SPPE 位的设定值。与 MSB first 和 LSB first 无关, RSPI 将 SPDR.LSB 位开始的设定数据长度的范围作为传送数据进行处理。

#### 26.3.5.1 MSB first 传送 (32 位数据)

##### (1) 奇偶校验功能无效的情况 (SPCR2.SPPE=0)

在奇偶校验功能无效的情况下, RSPI 进行 32 位数据长度的 MSB first 传送时的 SPDR 寄存器和移位寄存器的运行内容如图 26.13 所示。

将 T31 ~ T00 写到 SPDR 寄存器的发送缓冲器。如果发送缓冲器有数据并且移位寄存器为空, RSPI 就将发送缓冲器的数据复制到移位寄存器, 使移位寄存器变满。一旦串行传送开始, RSPI 就从移位寄存器的 MSB (bit31) 开始输出数据, 从移位寄存器的 LSB (bit0) 开始移入数据。在经过 32 位串行传送所需的 RSPCK 周期后, 数据 R31 ~ R00 被保存到移位寄存器。在此状态下, 如果进行全双工同步串行通信运行 (SPCR.TXMD=0), RSPI 就将数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器, 使移位寄存器变空。

如果在写 SPDR 寄存器的发送缓冲器前启动下一次串行传送, 就从移位寄存器移出接收数据 R31 ~ R00。

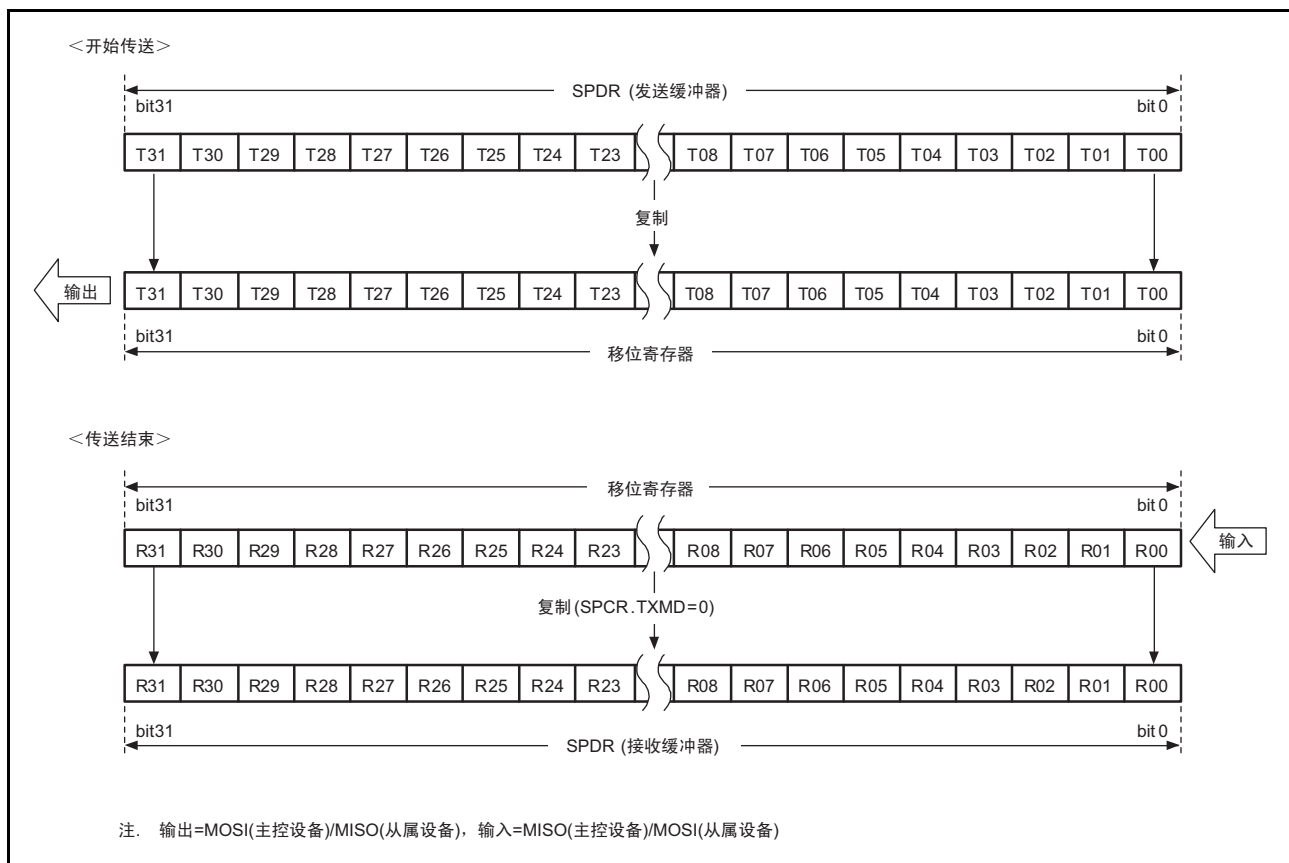


图 26.13 MSB first 传送 (1) (32 位数据 / 奇偶校验功能无效)

(2) 奇偶校验功能有效的情况 (SPCR2.SPPE=1)

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 SPDR 寄存器和移位寄存器的运行内容如图 26.14 所示。

将 T31 ~ T00 写到 SPDR 寄存器的发送缓冲器。如果发送缓冲器有数据并且移位寄存器为空，RSPI 就将 SPDR 寄存器的发送缓冲器保存的数据 T00 转换为奇偶校验位 (P)。将附加了奇偶校验位 (P) 的数据复制到移位寄存器，使移位寄存器变满。一旦串行传送开始，RSPI 就从移位寄存器的 MSB (bit31) 开始输出数据，从移位寄存器的 LSB (bit0) 开始移入数据。

在经过 32 位串行传送所需的 RSPCK 周期后，数据 R31 ~ P 被保存到移位寄存器。在此状态下，如果进行全双工同步串行通信运行 (SPCR.TXMD=0)，RSPI 就将数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器，使移位寄存器变空。

如果在写 SPDR 寄存器的发送缓冲器前启动下一次串行传送，就从移位寄存器移出接收数据 R31 ~ P。

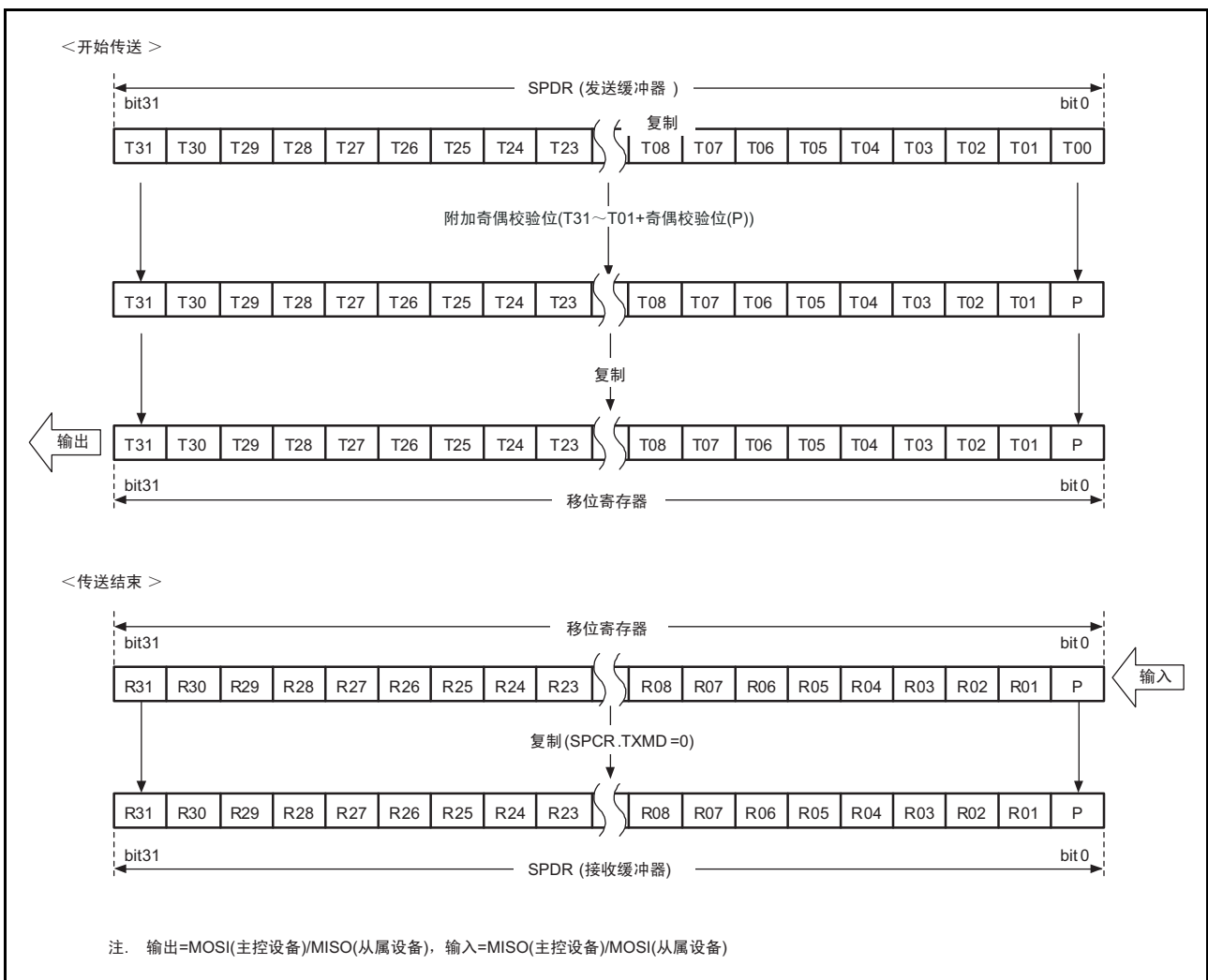


图 26.14 MSB first 传送 (2) (32 位数据 / 奇偶校验功能有效)

26.3.5.2 MSB first 传送 (24 位数据)

(1) 奇偶校验功能无效的情况 (SPCR2.SPPE=0)

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 SPDR 寄存器和移位寄存器的运行内容如图 26.15 所示。

将 T31 ~ T00 写到 SPDR 寄存器的发送缓冲器。如果发送缓冲器有数据并且移位寄存器为空，RSPI 就将 SPDR 寄存器的发送缓冲器保存的数据复制到移位寄存器，使移位寄存器变满。一旦串行传送开始，RSPI 就从移位寄存器的 bit23 开始输出数据，从移位寄存器的 LSB (bit0) 开始移入数据。在经过 24 位串行传送所需的 RSPCK 周期后，接收数据 R23 ~ R00 被保存到移位寄存器的 bit23 ~ bit0。串行传送结束后的移位寄存器的 bit31 ~ bit24 保持传送前的数据。在此状态下，如果进行全双工同步串行通信运行 (SPCR.TXMD=0)，RSPI 就将数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器，使移位寄存器变空。

如果在写 SPDR 寄存器的发送缓冲器前启动下一次串行传送，就从移位寄存器移出接收数据 R23 ~ R00。

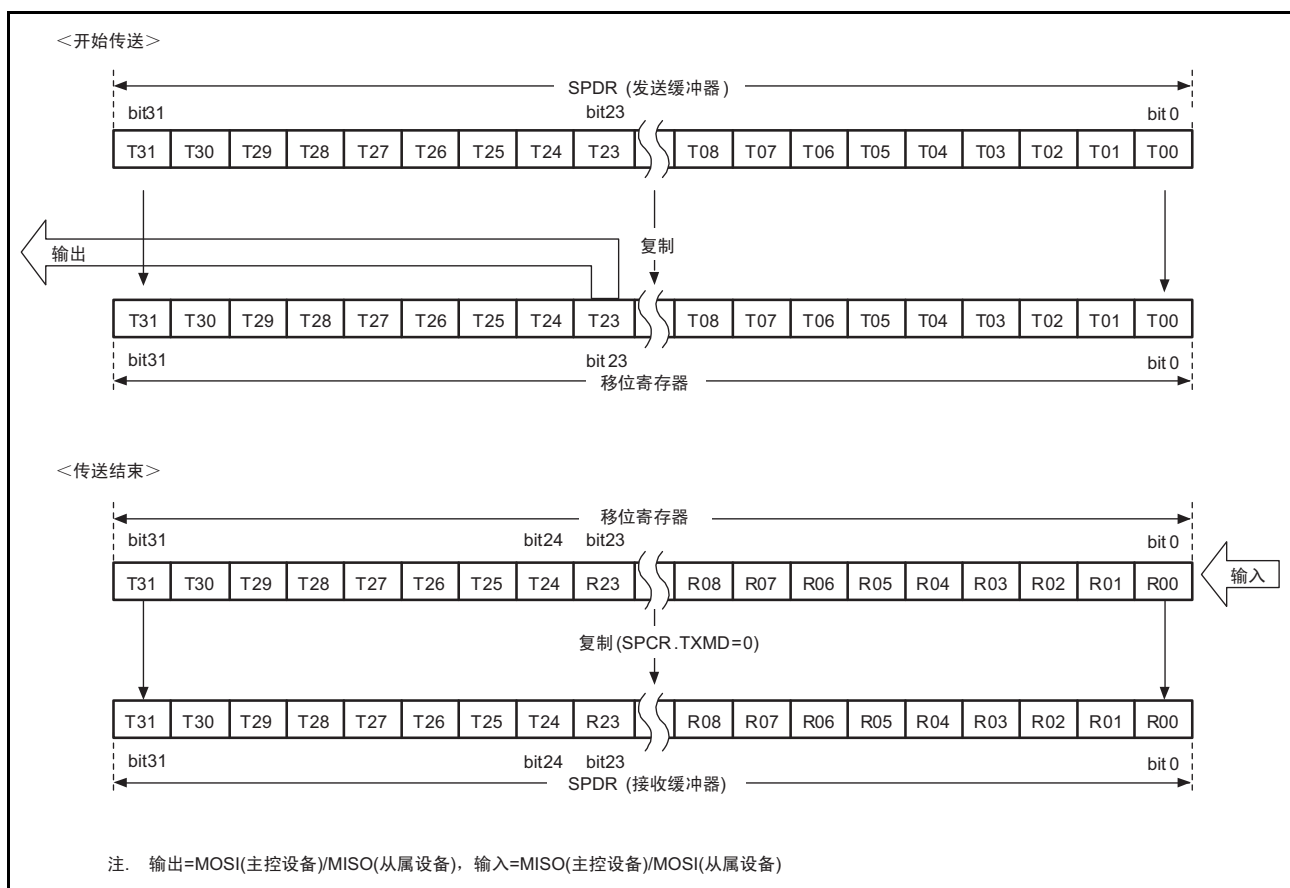


图 26.15 MSB first 传送 (1) (24 位数据 / 奇偶校验功能无效)



(2) 奇偶校验功能有效的情况 (SPCR2.SPPE=1)

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 SPDR 寄存器和移位寄存器的运行内容如图 26.16 所示。

将 T31 ~ T00 写到 SPDR 寄存器的发送缓冲器。如果发送缓冲器有数据并且移位寄存器为空，RSPI 就将 SPDR 寄存器的发送缓冲器保存的数据 T00 转换为奇偶校验位 (P)。将附加了奇偶校验位 (P) 的数据复制到移位寄存器，使移位寄存器变满。一旦串行传送开始，RSPI 就从移位寄存器的 bit23 开始输出数据，从移位寄存器的 LSB (bit0) 开始移入数据。在经过 24 位串行传送所需的 RSPCK 周期后，接收数据 R23 ~ P 被保存到移位寄存器的 bit23 ~ bit0。串行传送结束后的移位寄存器的 bit31 ~ bit24 保持传送前的数据。在此状态下，如果进行全双工同步模式的串行通信运行 (SPCR.TXMD=0)，RSPI 就将数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器，使移位寄存器变空。

如果在写 SPDR 寄存器的发送缓冲器前启动下一次串行传送，就从移位寄存器移出接收数据 R23 ~ P。

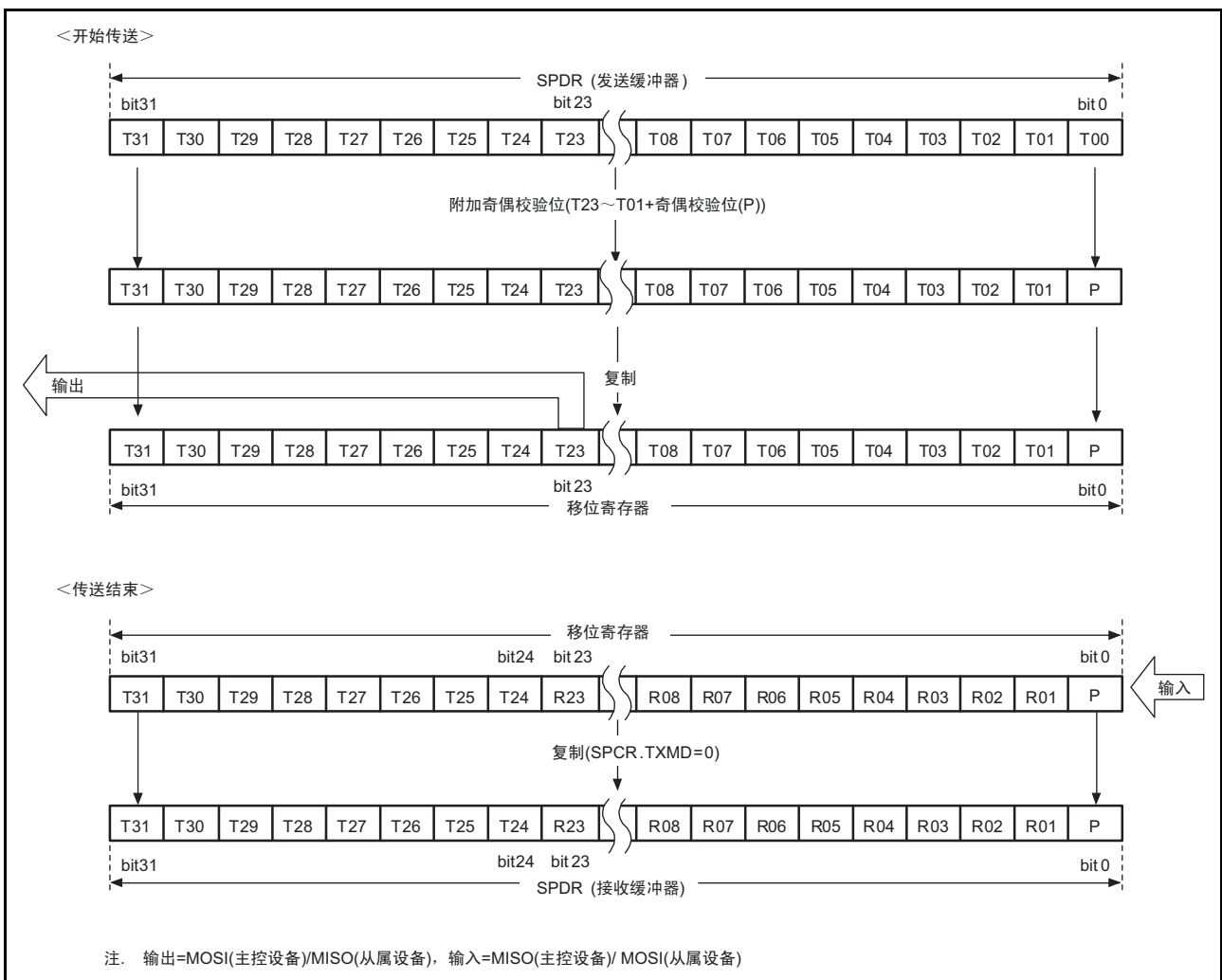


图 26.16 MSB first 传送 (2) (24 位数据 / 奇偶校验功能有效)

26.3.5.3 LSB first 传送 (32 位数据)

(1) 奇偶校验功能无效的情况 (SPCR2.SPPE=0)

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 SPDR 寄存器和移位寄存器的运行内容如图 26.17 所示。

将 T31 ~ T00 写到 SPDR 寄存器的发送缓冲器。如果发送缓冲器有数据并且移位寄存器为空，RSPI 就将 SPDR 寄存器的发送缓冲器的数据位序颠倒，然后复制到移位寄存器，使移位寄存器变满。一旦串行传送开始，RSPI 就从移位寄存器的 MSB (bit31) 开始输出数据，从移位寄存器的 LSB (bit0) 开始移入数据。在经过 32 位串行传送所需的 RSPCK 周期后，数据 R00 ~ R31 被保存到移位寄存器。在此状态下，如果进行全双工同步串行通信运行 (SPCR.TXMD=0)，RSPI 就将位序颠倒的数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器，使移位寄存器变空。

如果在写 SPDR 寄存器的发送缓冲器前启动下一次串行传送，就从移位寄存器移出接收数据 R00 ~ R31。

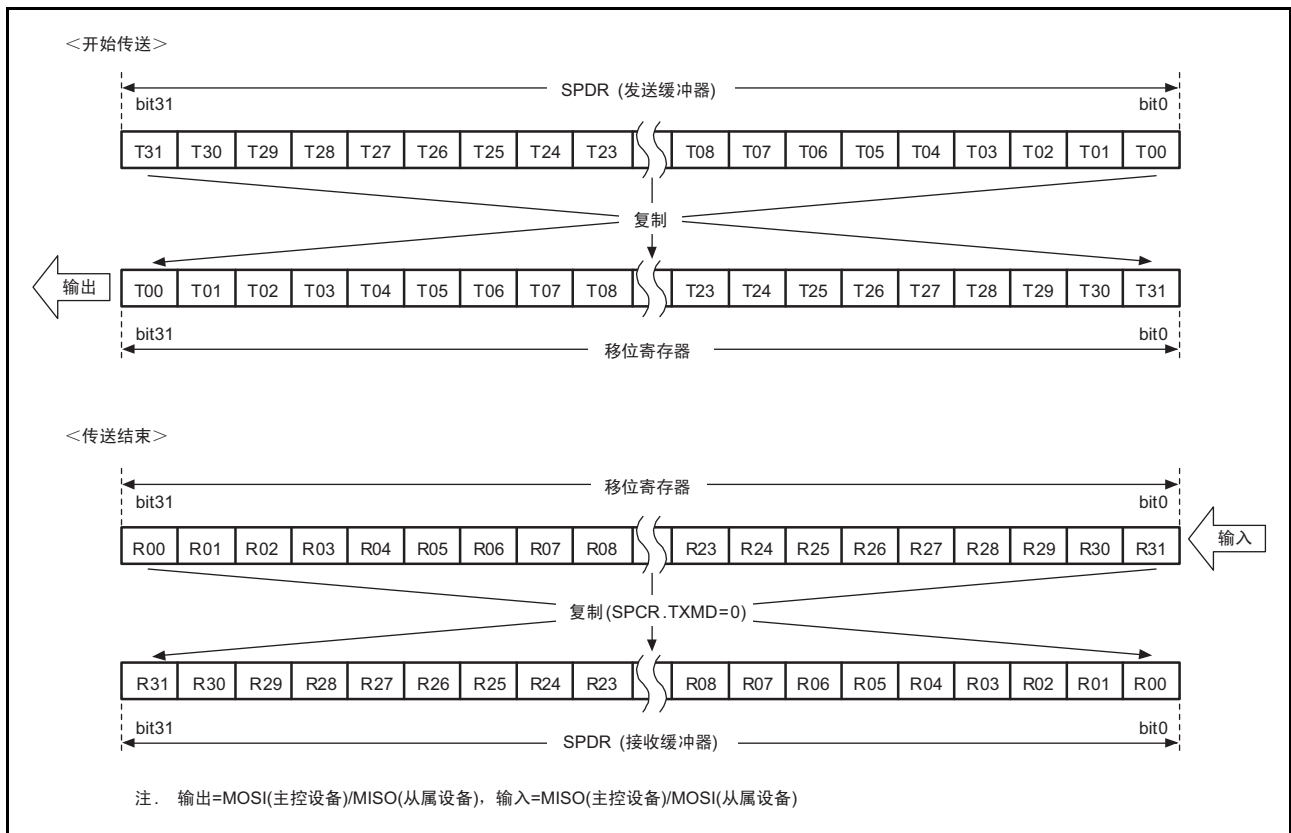


图 26.17 LSB first 传送 (1) (32 位数据 / 奇偶校验功能无效)

(2) 奇偶校验功能有效的情况 (SPCR2.SPPE=1)

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 SPDR 寄存器和移位寄存器的运行内容如图 26.18 所示。

将 T31 ~ T00 写到 SPDR 寄存器的发送缓冲器。RSPI 将 SPDR 的发送缓冲器保存的数据 T31 转换为奇偶校验位 (P)。如果发送缓冲器有数据并且移位寄存器为空，就将附加了奇偶校验位 (P) 的数据位序颠倒，然后复制到移位寄存器，使移位寄存器变满。一旦串行传送开始，RSPI 就从移位寄存器的 MSB (bit31) 开始输出数据，从移位寄存器的 LSB (bit0) 开始移入数据。在经过 32 位串行传送所需的 RSPCK 周期后，数据 R00 ~ P 被保存到移位寄存器。在此状态下，如果进行全双工同步串行通信运行 (SPCR.TXMD=0)，RSPI 就将位序颠倒的数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器，使移位寄存器变空。

如果在写 SPDR 寄存器的发送缓冲器前启动下一次串行传送，就从移位寄存器移出接收数据 R00 ~ P。

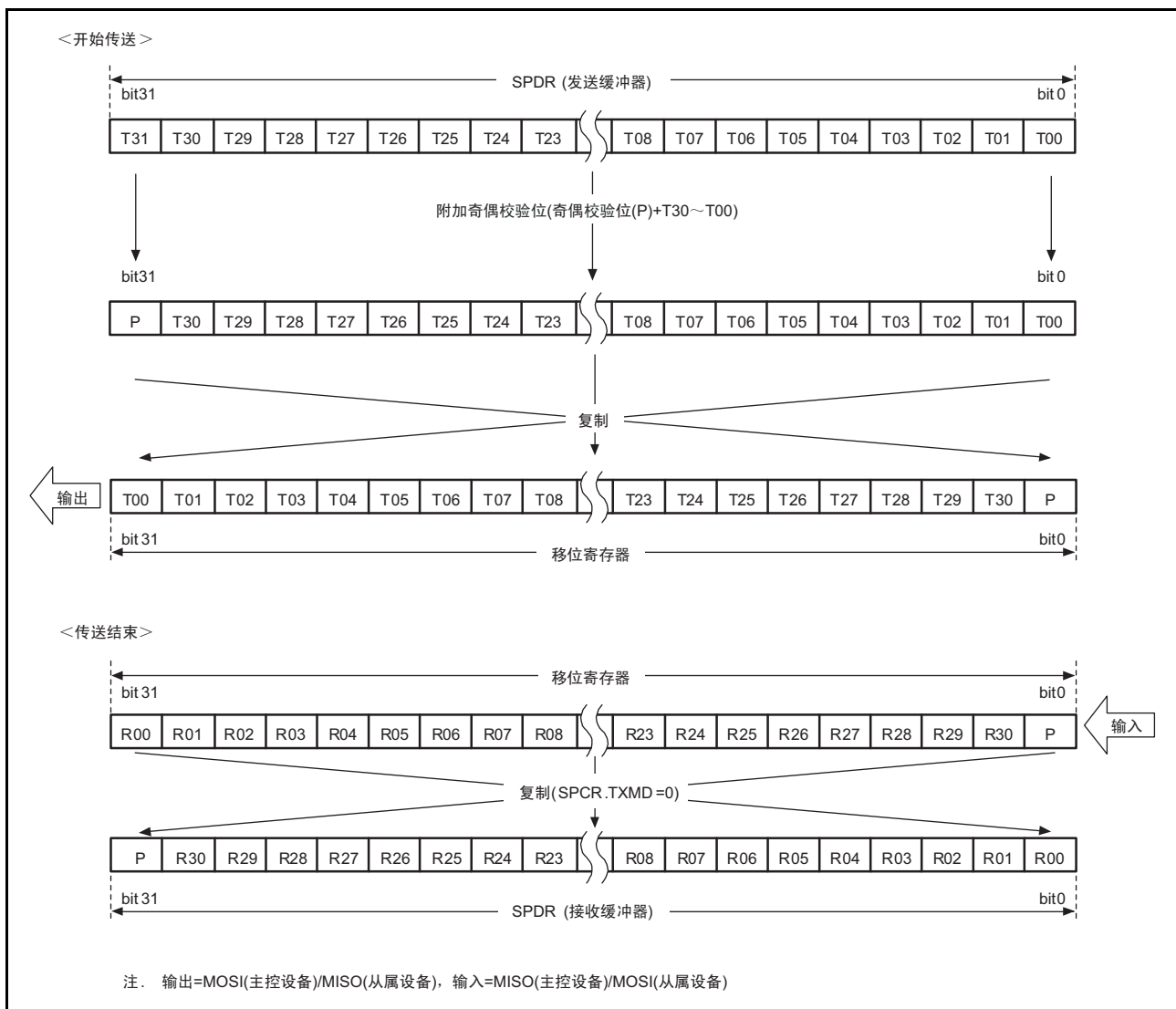


图 26.18 LSB first 传送 (2) (32 位数据 / 奇偶校验功能有效)

26.3.5.4 LSB first 传送 (24 位数据)

(1) 奇偶校验功能无效的情况 (SPCR2.SPPE=0)

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 SPDR 寄存器和移位寄存器的运行内容如图 26.19 所示。

将 T31 ~ T00 写到 SPDR 寄存器的发送缓冲器。如果发送缓冲器有数据并且移位寄存器为空，RSPI 就将 SPDR 寄存器的发送缓冲器的数据位序颠倒，然后复制到移位寄存器，使移位寄存器变满。一旦串行传送开始，RSPI 就从移位寄存器的 MSB (bit31) 开始输出数据，从移位寄存器的 bit8 开始移入数据。在经过 24 位串行传送所需的 RSPCK 周期后，接收数据 R00 ~ R23 被保存到移位寄存器的 bit31 ~ bit8。串行传送结束后的移位寄存器的 bit7 ~ bit0 保持传送前的数据。在此状态下，如果进行全双工同步串行通信运行 (SPCR.TXMD=0)，RSPI 就将位序颠倒的数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器，使移位寄存器变空。

如果在写 SPDR 寄存器的发送缓冲器前启动下一次串行传送，就从移位寄存器移出接收数据 R00 ~ R23。

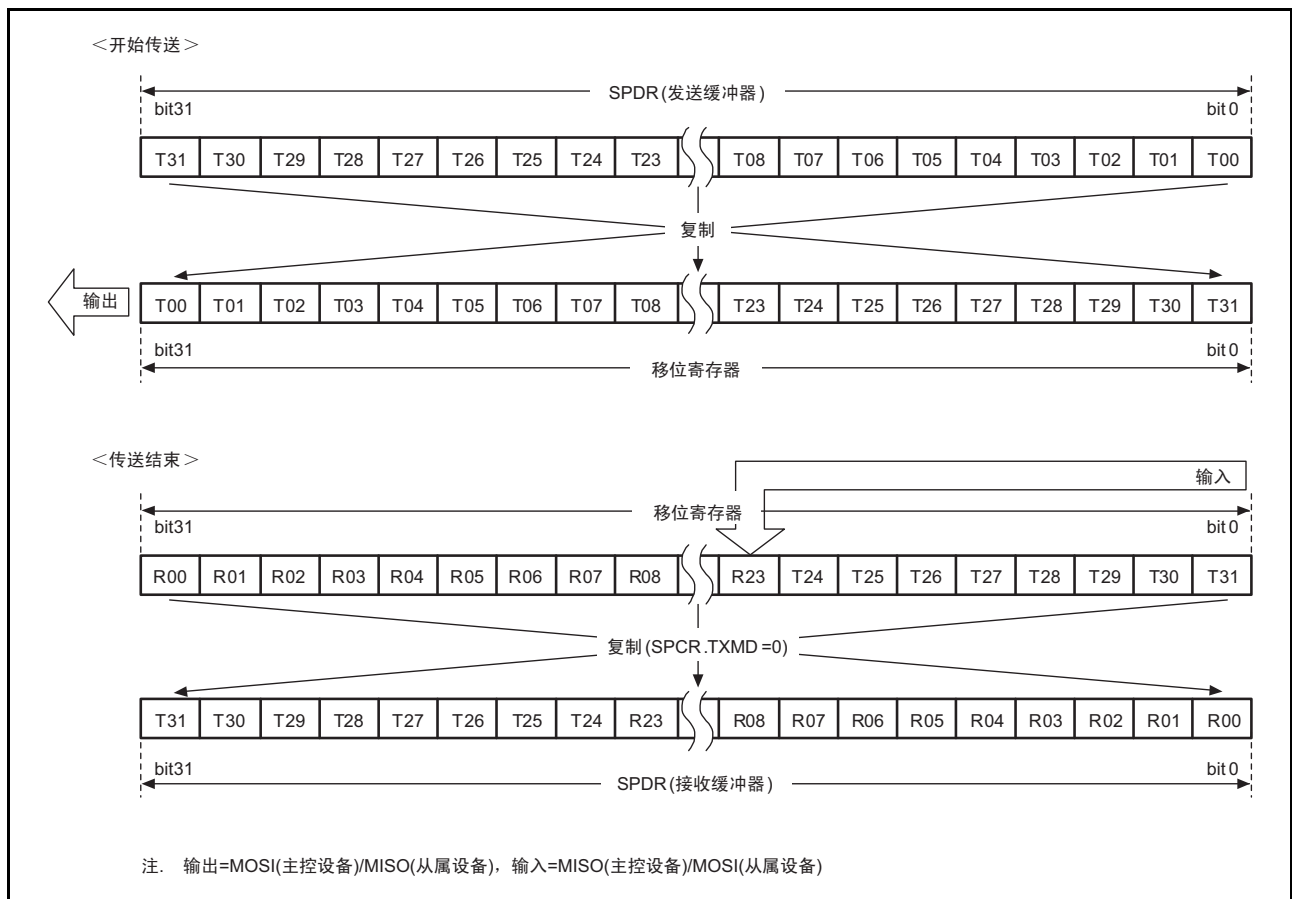


图 26.19 LSB first(1) (24 位数据 / 奇偶校验功能无效)

(2) 奇偶校验功能有效的情况 (SPCR2.SPPE=1)

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 SPDR 寄存器和移位寄存器的运行内容如图 26.20 所示。

将 T31 ~ T00 写到 SPDR 寄存器的发送缓冲器。RSPI 将 SPDR 的发送缓冲器保存的数据 T23 转换为奇偶校验位 (P)。如果发送缓冲器有数据并且移位寄存器为空，就将附加了奇偶校验位 (P) 的数据位序颠倒，然后复制到移位寄存器，使移位寄存器变满。一旦串行传送开始，RSPI 就从移位寄存器的 MSB (bit31) 开始输出数据，从移位寄存器的 bit8 开始移入数据。在经过 24 位串行传送所需的 RSPCK 周期后，接收数据 R00 ~ P 被保存到移位寄存器的 bit31 ~ bit8。串行传送结束后的移位寄存器的 bit7 ~ bit0 保持传送前的数据。在此状态下，如果进行全双工同步串行通信运行 (SPCR.TXMD=0)，RSPI 就将位序颠倒的数据从移位寄存器复制到 SPDR 寄存器的接收缓冲器，使移位寄存器变空。

如果在写 SPDR 寄存器的发送缓冲器前启动下一次串行传送，就从移位寄存器移出接收数据 R00 ~ P。

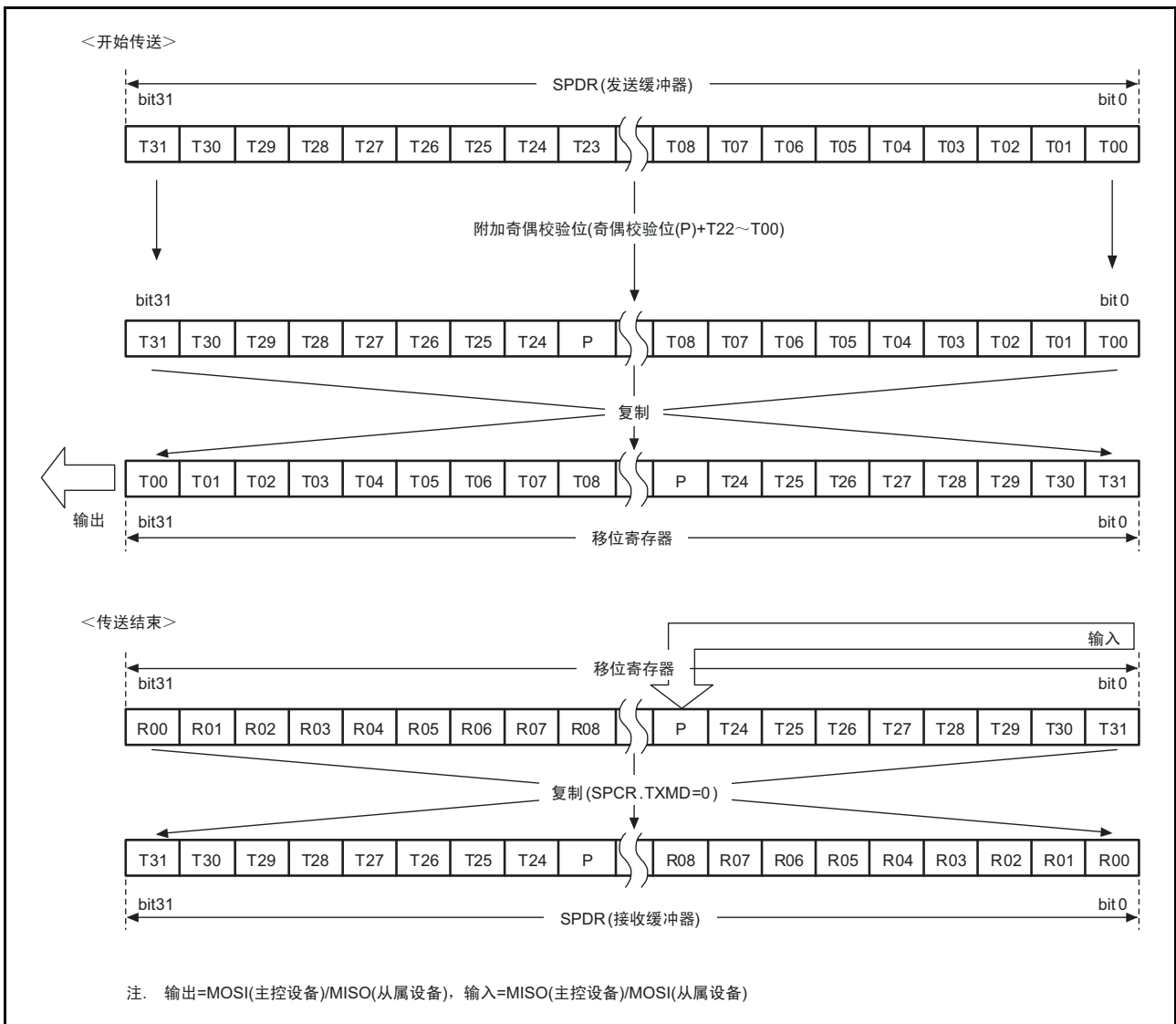


图 26.20 LSB first(2) (24 位数据 / 奇偶校验功能有效)

### 26.3.6 通信运行模式

通过设定 SPCR.TXMD，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 26.21 和图 26.22 记载的“SPDR 存取”表示 RSPI 数据寄存器 (SPDR) 的存取状况，“I”表示空闲周期，“W”表示写周期。

#### 26.3.6.1 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 26.21 所示。在图 26.21 的例子中，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。RSPCK 波形下面记载的数字表示 RSPCK 周期数 (传送位数)。(m=0~7)

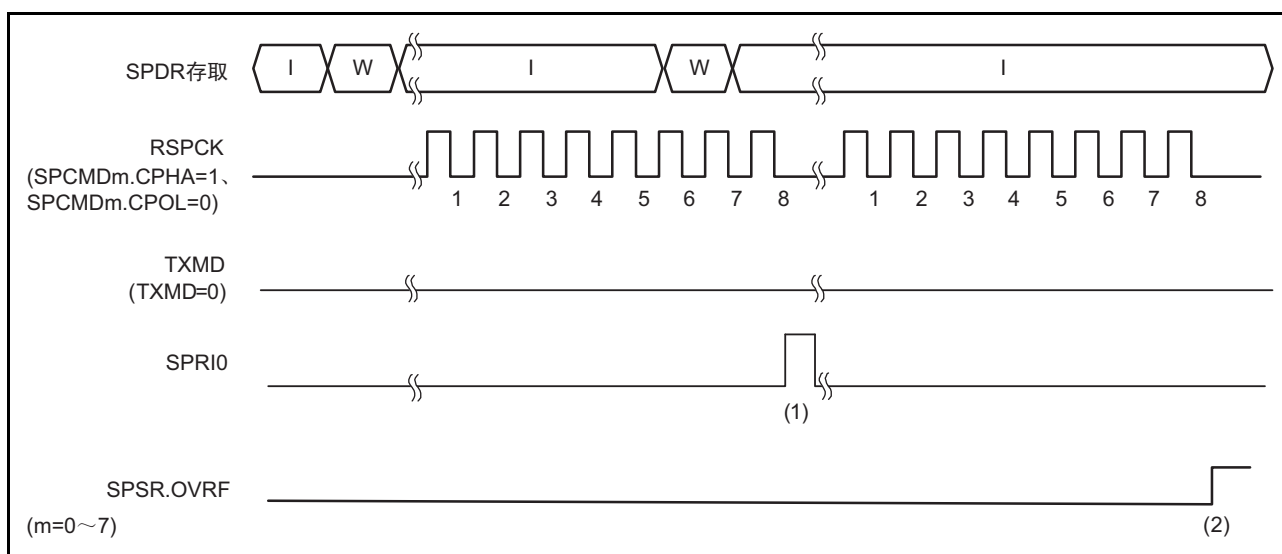


图 26.21 SPCR.TXMD 为“0”时的运行例子

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 (SPRI0) 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 的接收缓冲器有旧的接收数据的状态下结束串行传送，RSPI 就在将 SPSR.OVRF 置“1”后，放弃移位寄存器的接收数据。

在进行全双工同步串行通信时 (SPCR.TXMD=0)，发送数据以及接收数据。因此，在 (1) 和 (2) 各自的时序中，SPSR.OVRF 变为“1”。

## 26.3.6.2 只进行发送的运行 (SPCR.TXMD=1)

SPCR.TXMD 位为“1”时的运行例子如图 26.22 所示。在图 26.22 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。RSPCK 波形下面记载的数字表示 RSPCK 周期数（传送位数）。（m=0~7）

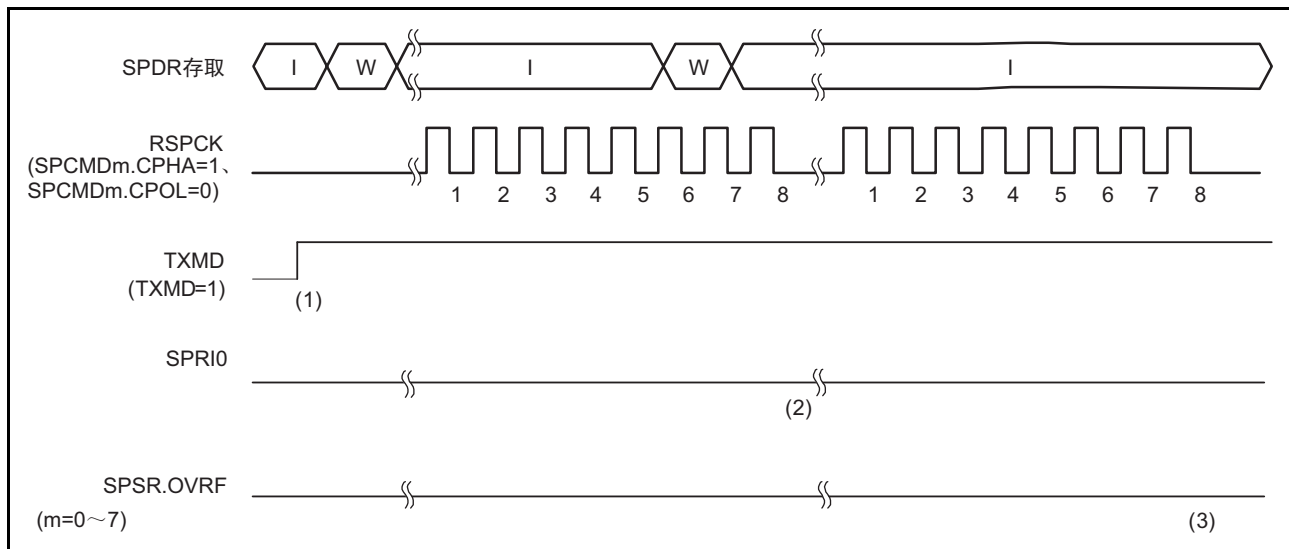


图 26.22 SPCR.TXMD 为“1”时的运行例子

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 标志为“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旧的接收数据，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 保持“0”。

### 26.3.7 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 (SPTI0) 和 RSPI 接收缓冲器满中断 (SPRI0) 的运行例子如图 26.23 所示。图 26.23 中记载的“SPDR 存取”表示 RSPI 数据寄存器 (SPDR) 的存取状况, “T”表示空闲周期, “W”表示写周期, “R”表示读周期。在图 26.23 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 位为“0”, RSPI 进行 8 位串行传送。RSPCK 波形下面记载的数字表示 RSPCK 周期数 (传送位数)。(m=0~7)

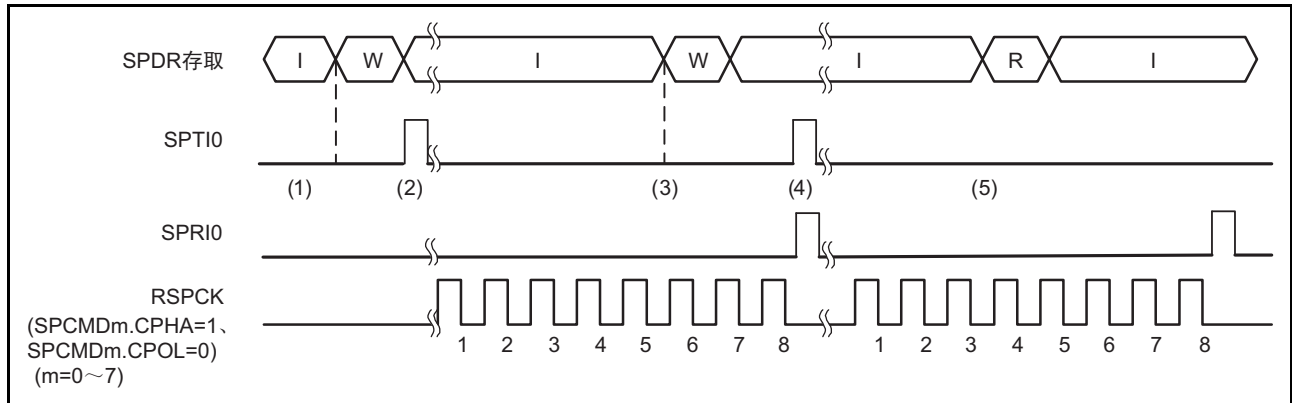


图 26.23 SPTI0 中断和 SPRI0 中断的运行例子

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 (SPTI0)。串行传送的开始方法取决于 RSPI 的模式 (参照“26.3.10 SPI 运行”和“26.3.11 时钟同步运行”)
- (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 (SPRI0)。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, 就将接收缓冲器的数据发送到芯片的内部总线。

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请求来写 SPDR 寄存器。在使用 RSPI 发送中断时, 必须将 SPCR.SPTIE 位置“1”。

在禁止 RSPI (SPCR 的 SPE 位为“0”) 状态下, 必须将 SPCR.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“26.3.8 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 IRi.IR 标志确认发送 / 接收缓冲器的状态。



### 26.3.8 错误检测

在通常的 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。普通传送以外的传送和 RSPI 错误检测功能的关系如表 26.9 所示。

表 26.9 普通传送以外的传送的产生条件和 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。	保持发送缓冲器内容并且写数据丢失。	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 SSL0 输入信号有效。	停止驱动 RSPICK、MOSI、SSL1 ~ 3 输出信号并且禁止 RSPI。	模式故障错误检测
G	在多主控模式的串行传送过程中 SSL0 输入信号有效。	中止串行传送。 发送 / 接收数据丢失。 停止驱动 RSPCK、MOSI、SSL1 ~ 3 输出信号。 RSPI 运行无效。	模式故障错误检测
H	在从属模式的串行传送过程中 SSL0 输入信号无效。	中止串行传送。 发送 / 接收数据丢失。 停止驱动 MISO 输出信号。 RSPI 运行无效。	模式故障错误检测

对于表 26.9 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。

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（如果在串行传送结束前读 SPDR 寄存器，就发生溢出错误）。

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“26.3.8.1 溢出错误”；有关 E 所示的奇偶校验错误，请参照“26.3.8.2 奇偶校验错误”；有关 F ~ H 所示的模式故障错误，请参照“26.3.8.3 模式故障错误”。

有关发送中断和接收中断，请参照“26.3.7 发送缓冲器空中断 / 接收缓冲器满中断”。

## 26.3.8.1 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 SPSR.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 SPSR.OVRF 标志置“0”时，需要在 SPSR.OVRF 标志为“1”的状态下，在读 SPSR 寄存器后给 SPSR.OVRF 标志写“0”。

SPSR.OVRF 标志的运行如图 26.24 所示。图 26.24 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，“I”表示空闲状态，“W”表示写周期，“R”表示读周期。在图 26.24 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。RSPCK 波形下面记载的数字表示 RSPCK 周期数（传送位数）。（m=0~7）

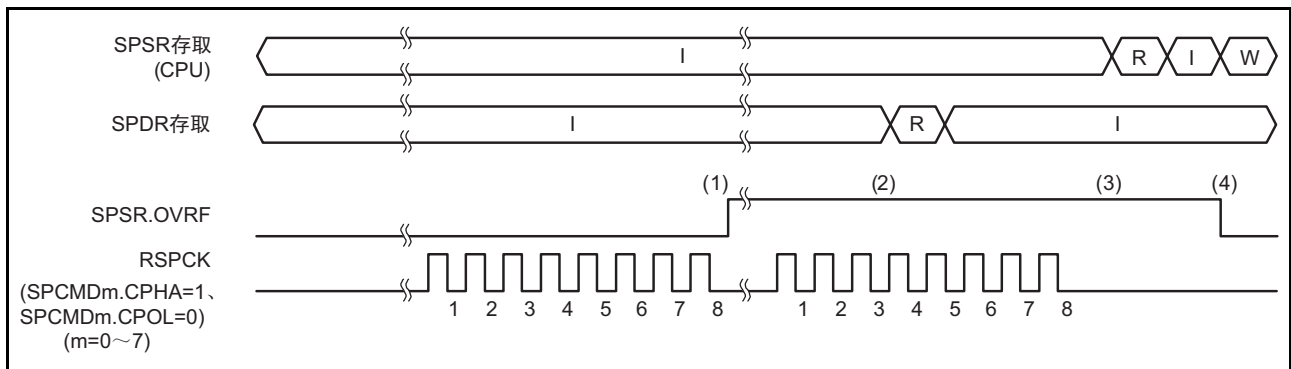


图 26.24 SPSR.OVRF 标志的运行例子

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 SPSR.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 SPCR2.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。（m=0~7）
- (2) 如果读 SPDR 寄存器，RSPI 就将接收缓冲器的数据输出到内部总线。即使接收缓冲器为空，SPSR.OVRF 标志也不变为“0”。
- (3) 如果在 SPSR.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。即使 SPCR2.SPPE 位为“1”，也不检测奇偶校验错误。在主控模式中，RSPI 不更新 SPSR.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) 在 SPSR.OVRF 标志为“1”的状态下，如果 CPU 在读 SPSR 寄存器后给 SPSR.OVRF 标志写“0”，RSPI 就将 SPSR.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。在使用 RSPI 错误中断时，必须将 SPCR.SPEIE 位置“1”。在不使用 RSPI 错误中断进行串行传送时，必须通过在读 SPDR 寄存器后立即读 SPSR 寄存器等方法，尽快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 SPSR.OVRF 标志变为“1”，就在 SPSR.OVRF 标志变为“0”前无法进行正常的接收运行。SPSR.OVRF 标志为“0”的条件如下：

- 在 SPSR.OVRF 标志为“1”的状态下，CPU 在读 SPSR 寄存器后给 SPSR.OVRF 标志写“0”。

## 26.3.8.2 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 标志为“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 SPSR.PERF 标志置“0”时，需要在 SPSR.PERF 标志为“1”的状态下，CPU 在读 SPSR 寄存器后给 SPSR.PERF 标志写“0”。

SPSR.OVRF 标志和 SPSR.PERF 标志的运行如图 26.25 所示。图 26.25 中记载的“SPSR 存取”表示 SPSR 寄存器的存取状况，“T”表示空闲状态，“W”表示写周期，“R”表示读周期。在图 26.25 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。SPCMDm.CPHA 位为“1”并且 SPCMDm.CPOL 位为“0”，RSPI 进行 8 位串行传送。RSPCK 波形下面记载的数字表示 RSPCK 周期数（传送位数）。（m=0~7）

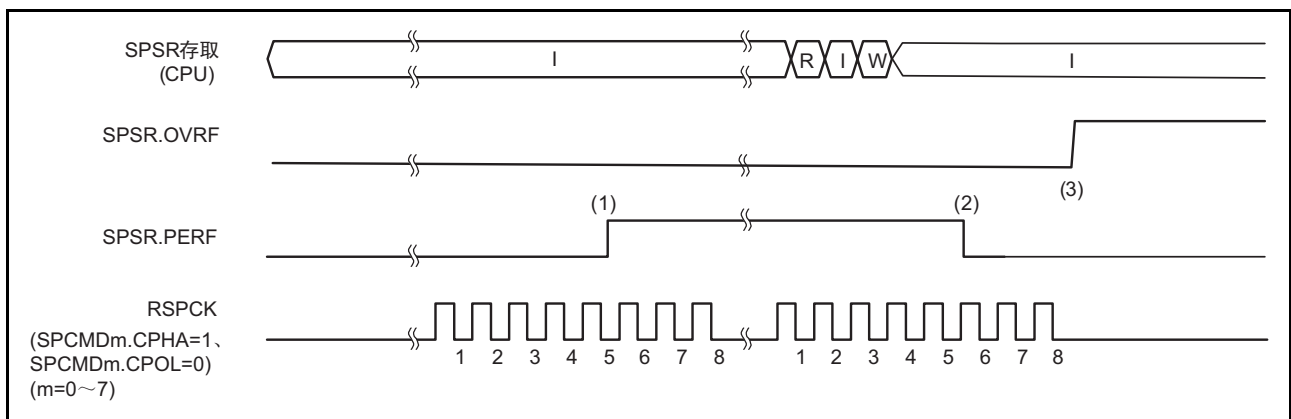


图 26.25 SPSR.PERF 标志的运行例子

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 SPSR.PERF 标志置“1”。在主控模式中，RSPI 将指向 SPCMDm 寄存器的指针值复制到 SPSSR.SPECM[2:0] 位。
- (2) 在 SPSR.PERF 标志为“1”的状态下，如果 CPU 在读 SPSR 寄存器后给 SPSR.PERF 标志写“0”，RSPI 就将 SPSR.RERF 标志置“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。在使用 RSPI 错误中断时，必须将 SPCR.SPEIE 位置“1”。在不使用 RSPI 错误中断进行串行传送时，必须通过读 SPSR 寄存器等方法，尽快检测到奇偶校验错误的发生。在可控模式中使用 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。

SPSR.PERF 标志为“0”的条件如下：

- 在 SPSR.PERF 标志为“1”的状态下，CPU 在读 SPSR 寄存器后给 SPSR.PERF 标志写“0”时

### 26.3.8.3 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 SSL0 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 的指针值复制到 SPSSR.SPECM[2:0] 位。SSL0 信号的有效电平取决于 SSLP.SSLOP 位。(m=0 ~ 7)

当 SPCR.MSTR 位为“0”时，RSPI 以从属模式运行。当从属模式的 RSPI 的 SPCR.MODFEN 位为“1”并且 SPCR.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 SSL0 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“26.3.9 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。在使用 RSPI 错误中断时，必须将 SPCR.SPEIE 位置“1”。在不使用 RSPI 错误中断来检测模式错误时，需要轮询 SPSR 寄存器。在从属模式中使用 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 SPSR.MODF 标志为“1”的状态下，RSPI 忽视通过 CPU 给 SPCR.SPE 位写“1”。为了使 RSPI 功能在检测到模式故障错误后有效，必须将 SPSR.MODF 标志置“0”。SPSR.MODF 标志为“0”的条件如下：

- 在 SPSR.MODF 标志为“1”的状态下，CPU 在读 SPSR 寄存器后给 SPSR.MODF 标志写“0”时

## 26.3.9 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 SPCR.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。

### 26.3.9.1 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 发送缓冲器置空。

在通过将 SPCR.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 CPU 重新将 SPCR.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”，就发生 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 SPCR.SPE 位写“0”的同时给 SPCR.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 SPCR.SPTIE 位写“0”。(m=0 ~ 7)

### 26.3.9.2 系统复位

在通过系统复位进行初始化时，除了“26.3.9.1 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。

## 26.3.10 SPI 运行

### 26.3.10.1 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“26.3.8 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的数据。在通过写 SPDR 寄存器将移位寄存器改为空的状态下，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。CPU 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“26.3.4 传送格式”。SSL 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 RSPCK 边沿，就结束串行传送。在接收缓冲器有空间的情况下，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，SSL 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“26.3.4 传送格式”。(m=0 ~ 7)

#### (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 SSL 引脚的输出信号值、MSB first/LSB first 方式、数据长度、一部分位速率、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 寄存器设定一部分位速率，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。CPU 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。(m=0 ~ 7)

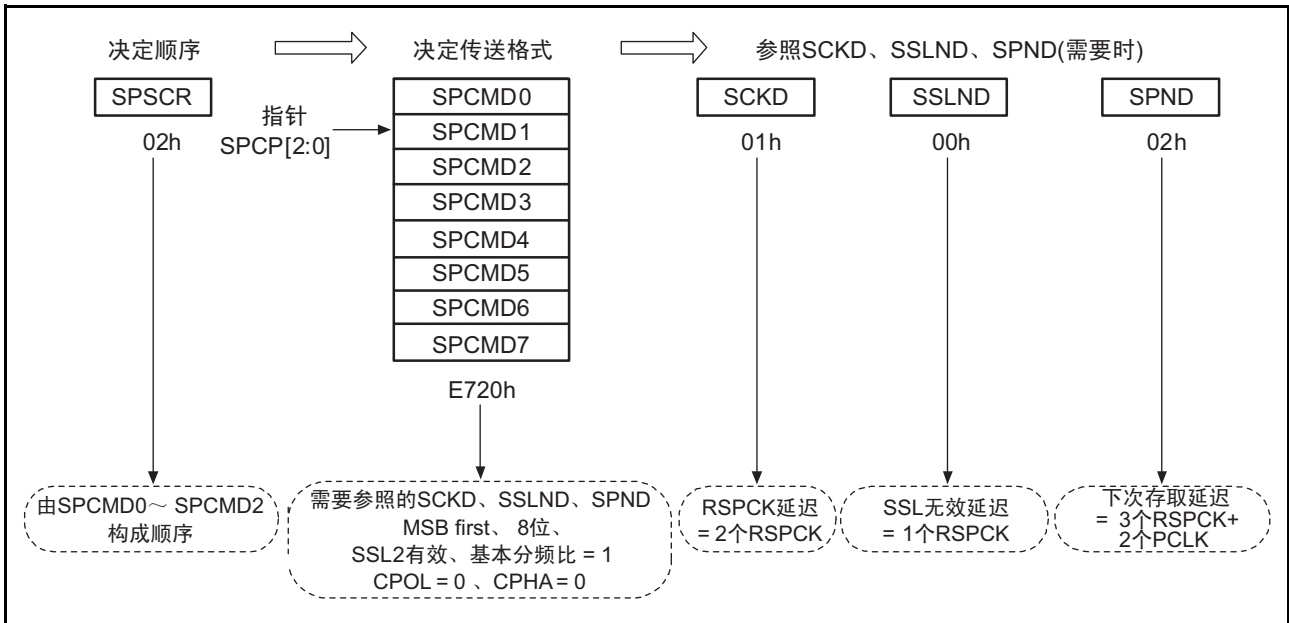


图 26.26 主控模式中的串行传送方式的决定方法 (SPI 运行)

(4) 突发传送

当 RSPI 正在进行的串行传送所参照的 SPCMDm.SSLKP 位 (m=0 ~ 7) 为“1”时, RSPI 将串行传送过程中的 SSL 信号电平保持到下次串行传送的 SSL 信号开始有效为止。如果下次串行传送的 SSL 信号电平和正在进行的串行传送的 SSL 信号电平相同, RSPI 就能在保持 SSL 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 SSL 信号运行例子如图 26.27 所示。以下说明图 26.27 中记载的 (1) ~ (7) 的 RSPI 运行内容。SSL 输出信号的极性取决于 RSPI 从属选择 SSLP 寄存器的设定值。

- (1) 根据 SPCMD0 寄存器, 使 SSL 信号有效并插入 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 SSL 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 SSL 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 SSL 信号无效。根据 SPCMD1 寄存器插入下次的存取延迟。

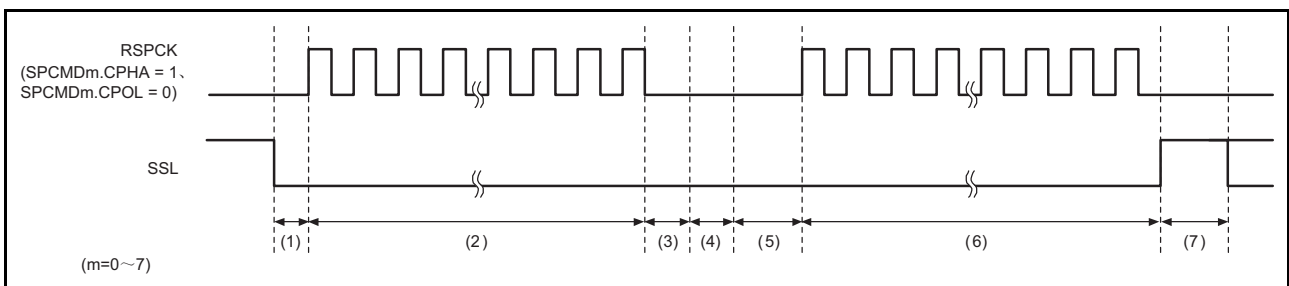


图 26.27 使用 SPCMDm.SSLKP 位的突发传送运行例子 (m=0 ~ 7)

在 SPCMDm.SSLKP 位为“1”的 SPCMDm 寄存器所设 SSL 信号输出和下次传送使用的 SPCMDm 寄存器所设 SSL 信号输出不同的情况下，RSPI 在下次传送命令所对应的 SSL 信号有效时（图 26.27 的 (5)）转换 SSL 信号状态。必须注意：如果转换 SSL 信号，就可能因驱动 MISO 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 SPCMDm.SSLKP 位时的 SSL 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 SSL 信号，正确地开始串行传送。因此，与 SPCMDm.CPHA 位的设定值无关，能进行主控模式的突发传送（参照“26.3.10.2 从属模式的运行”）。

#### (5) RSPCK 延迟 (t1)

主控模式的 RSPI 的 RSPCK 延迟值取决于 SPCMDm.SCKDEN 位 (m=0 ~ 7) 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 寄存器来决定表 26.10 所示的串行传送时的 RSPCK 延迟值。有关 RSPCK 延迟的定义，请参照“26.3.4 传送格式”。

表 26.10 SPCMDm.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	000b ~ 111b	1 个 RSPCK
1	000b	1 个 RSPCK
	001b	2 个 RSPCK
	010b	3 个 RSPCK
	011b	4 个 RSPCK
	100b	5 个 RSPCK
	101b	6 个 RSPCK
	110b	7 个 RSPCK
	111b	8 个 RSPCK

m = 0 ~ 7

#### (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 (m=0 ~ 7) 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 26.11 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“26.3.4 传送格式”。

表 26.11 SPCMDm.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	000b ~ 111b	1 个 RSPCK
1	000b	1 个 RSPCK
	001b	2 个 RSPCK
	010b	3 个 RSPCK
	011b	4 个 RSPCK
	100b	5 个 RSPCK
	101b	6 个 RSPCK
	110b	7 个 RSPCK
	111b	8 个 RSPCK

m = 0 ~ 7

## (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 (m=0 ~ 7) 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 26.12 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“26.3.4 传送格式”。

表 26.12 SPCMDm.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	000b ~ 111b	1 个 RSPCK+2 个 PCLK
1	000b	1 个 RSPCK+2 个 PCLK
	001b	2 个 RSPCK+2 个 PCLK
	010b	3 个 RSPCK+2 个 PCLK
	011b	4 个 RSPCK+2 个 PCLK
	100b	5 个 RSPCK+2 个 PCLK
	101b	6 个 RSPCK+2 个 PCLK
	110b	7 个 RSPCK+2 个 PCLK
	111b	8 个 RSPCK+2 个 PCLK

m = 0 ~ 7



## (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 26.28 所示。有关中断控制器、DTC 和输入 / 输出端口的设定方法，请参照各模块的说明。

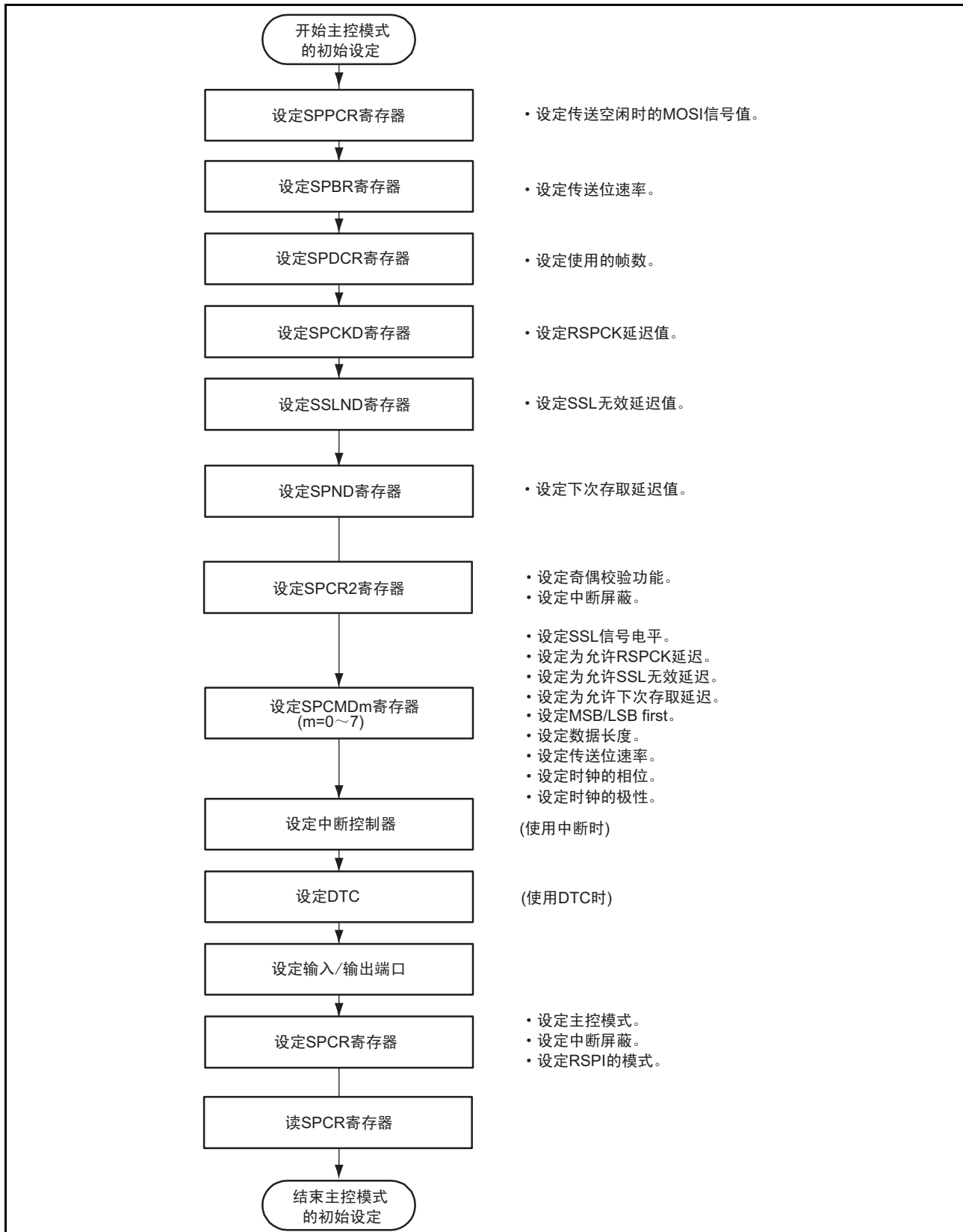
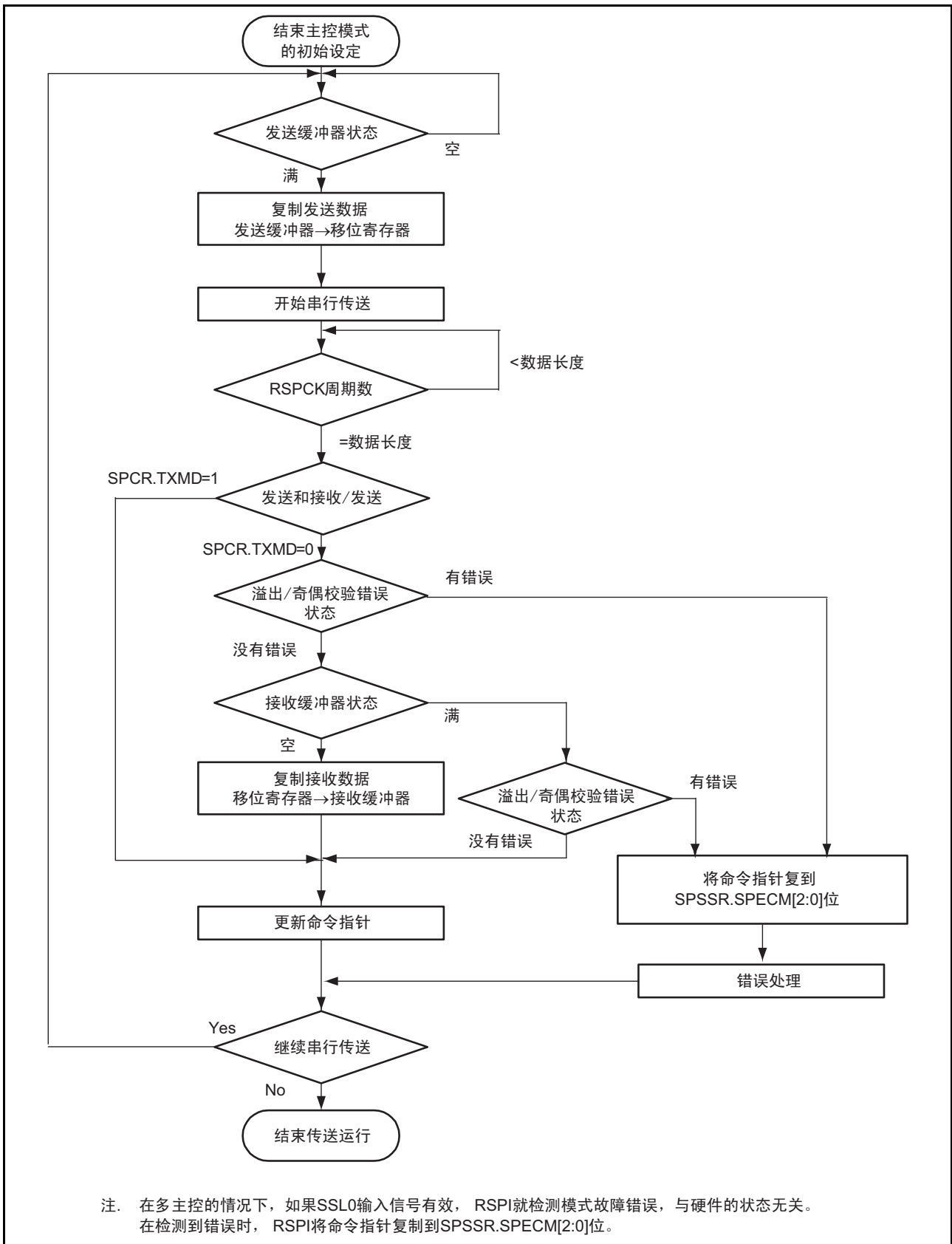


图 26.28 主控模式时的初始化流程例子 (SPI 运行)

(9) 传送运行流程

在 SPI 运行时，主控模式的 RSPI 传送运行流程如图 26.29 所示。



注. 在多主控的情况下，如果SSL0输入信号有效，RSPI就检测模式故障错误，与硬件的状态无关。在检测到错误时，RSPI将命令指针复制到SPSSR.SPECM[2:0]位。

图 26.29 主控模式时的传送运行流程 (SPI 运行)

### 26.3.10.2 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 SSL0 输入信号有效，就需要开始向 MISO 输出信号驱动有效数据。因此，在 SPCMD0.CPHA 位为“0”时，SSL0 输入信号的有效为串行传送的开始触发。

当 SPCMD0.CPHA 位为“1”时，如果在 SSL0 输入信号有效的状态下 RSPI 检测到最初的 RSPCK 边沿，就需要开始向 MISO 输出信号驱动有效数据。因此，在 SPCMD0.CPHA 位为“1”时，SSL0 信号有效状态下的最初的 RSPCK 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 SPCMD0.CPHA 位的设定无关，RSPI 开始驱动 MISO 输出信号的时序为 SSL0 信号的有效时序。RSPI 的输出数据的有效或者无效因 SPCMD0.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“26.3.4 传送格式”。SSL0 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 RSPCK 边沿，就结束串行传送。在接收缓冲器有空间的情况下，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。在串行传送开始到串行传送结束的期间，如果 RSPI 检测到 SSL0 输入信号无效，就发生模式故障错误（参照“26.3.8 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，SSL0 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“26.3.4 传送格式”。

#### (3) 单从属模式时的注意点

当 SPCMD0.CPHA 位为“0”时，如果 RSPI 检测到 SSL0 输入信号的有效边沿，就开始串行传送。在图 26.5 的例子所示的结构中，在单从属模式中使用 RSPI 时，因为 SSL0 输入信号总是被固定为有效状态，所以 SPCMD0.CPHA 位为“0”的 RSPI 无法正常开始串行传送。在 SSL0 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 SPCMD0.CPHA 位置“1”。如果需要将 SPCMD0.CPHA 位置“0”，就不能固定 SSL0 输入信号。

#### (4) 突发传送

当 SPCMD0.CPHA 位为“1”时，能在保持 SSL0 输入信号有效的状态下进行连续的串行传送（突发传送）。当 SPCMD0.CPHA 位为“1”时，从 SSL0 输入信号有效状态的最初 RSPCK 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 SSL0 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 SPCMD0.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 26.30 所示。有关中断控制器、DTC 和输入 / 输出端口的设定方法，请参照各模块的说明。

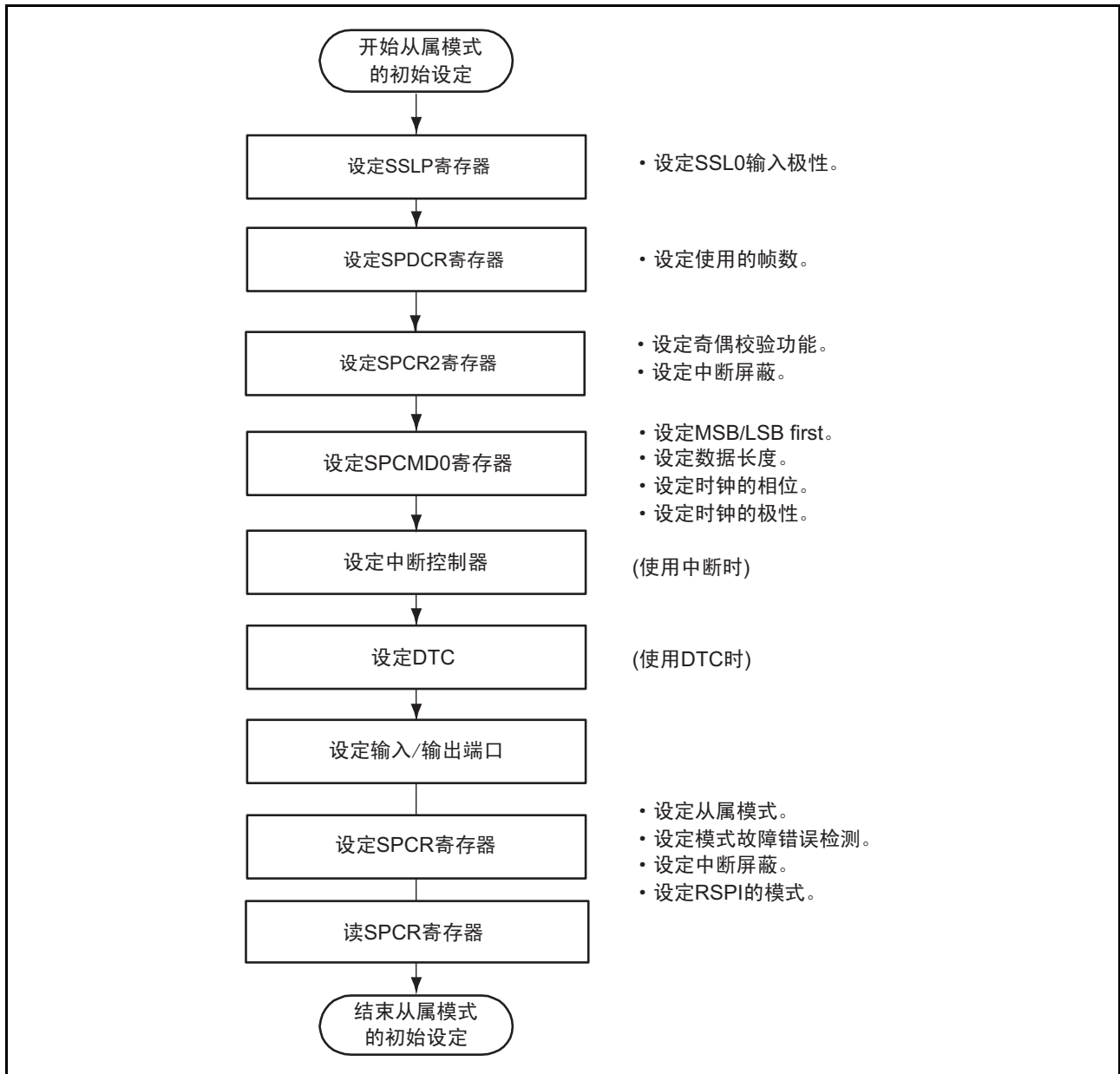


图 26.30 从属模式时的初始化流程例子 (SPI 运行)

(6) 传送运行流程 (SPCMD0.CPHA 位 = 0)

在 SPI 运行时，SPCMD0.CPHA 位为“0”的从属模式的 RSPI 传送运行流程如图 26.31 所示。

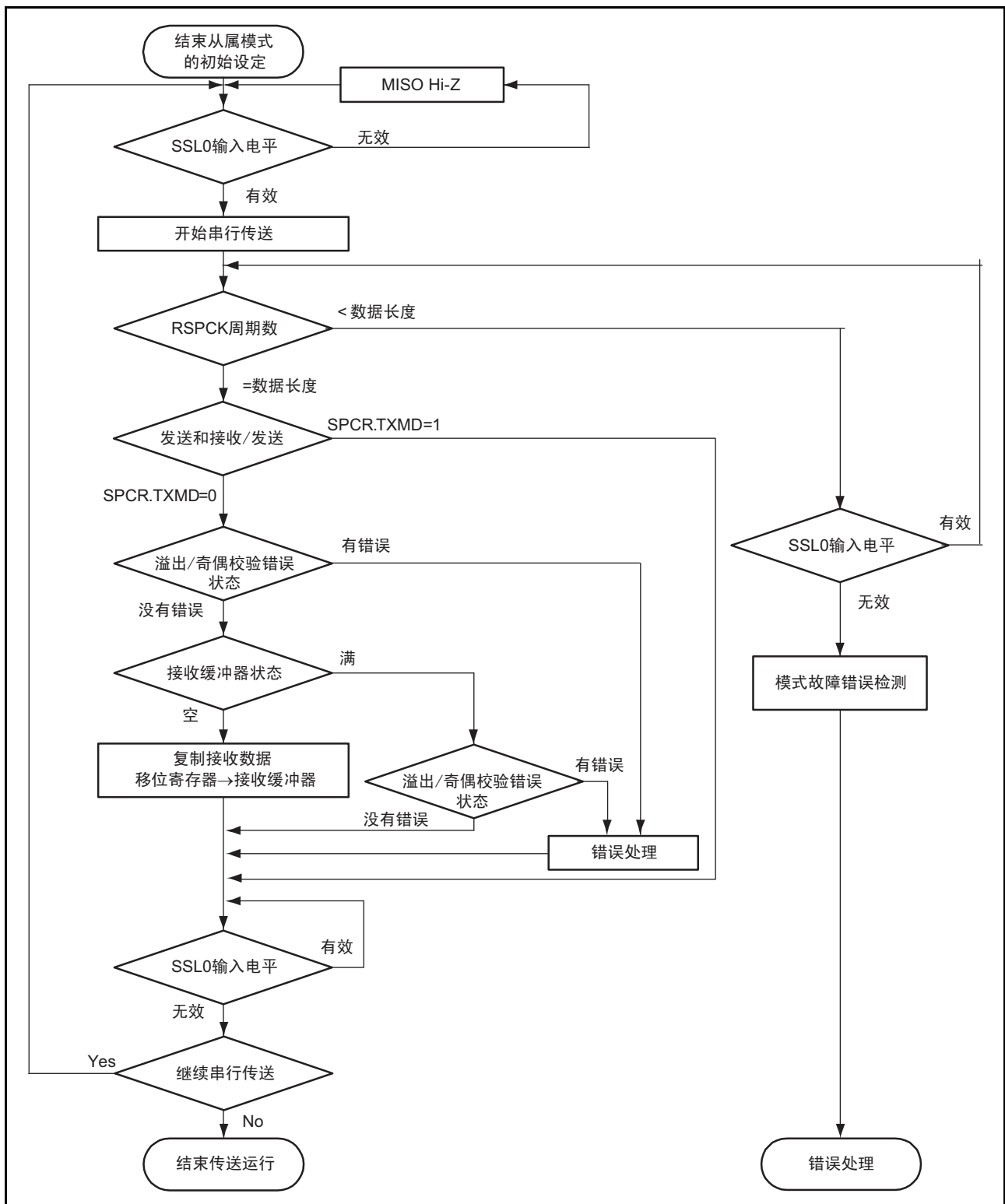


图 26.31 从属模式时的传送运行流程 (SPCMD0.CPHA 位 = 0) (SPI 运行)

(7) 传送运行流程 (SPCMD0.CPHA 位=1)

在 SPI 运行的情况下，SPCMD0.CPHA 位为“1”或者 SPCR.MODFEN 位为“1”时的从属模式的 RSPI 传送运行流程如图 26.32 所示。如果在 MODFEN 位为“0”的状态下开始串行传送，并且在 RSPCK 周期数短于数据长度的状态下将 SSL0 输入电平置为无效，就无法保证以后的运行。

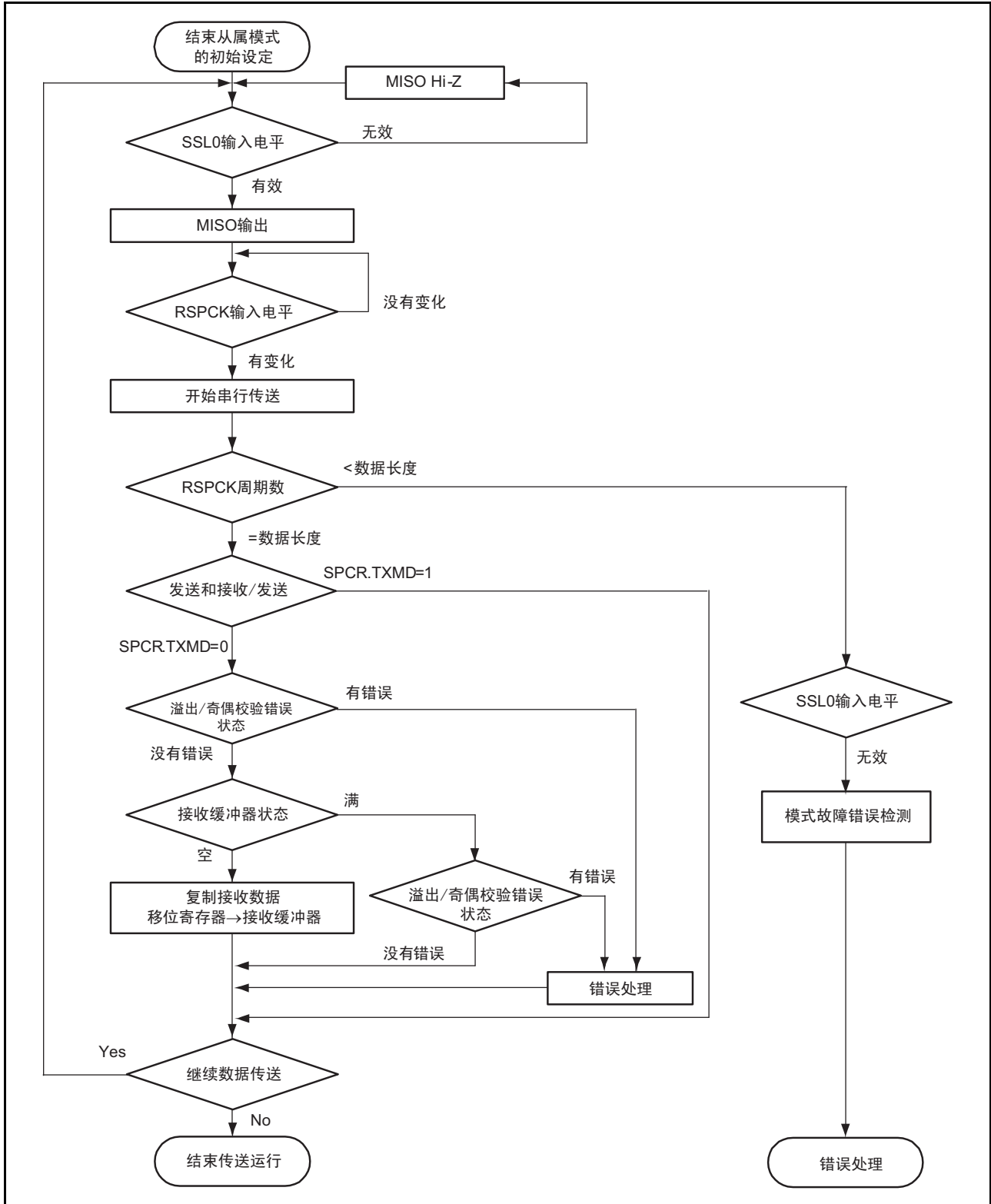


图 26.32 从属模式时的传送运行流程 (SPCMD0.CPHA=1) (SPI 运行)

### 26.3.11 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 SSL 引脚而使用 RSPCK、MOSI 和 MISO 这 3 个引脚进行通信，SSL 引脚能用作 I/O 端口。

在进行时钟同步运行时，不使用 SSL 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 SSL 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 (m=0~7) 为“0”时的运行。

#### 26.3.11.1 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的数据。在通过写 SPDR 寄存器将移位寄存器改为空的状态下，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。CPU 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“26.3.4 传送格式”。

##### (2) 串行传送的结束

如果 RSPI 发送采样时序所对应的 RSPCK 边沿，就结束串行传送。在接收缓冲器有空间的情况下，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 (m=0~7) 的设定值。

有关 RSPI 传送格式的详细内容，请参照“26.3.4 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 寄存器或者 SPND 寄存器。在进行时钟同步运行时，不输出 SSL 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 SSL 输出信号值、MSB first/LSB first 方式、数据长度、一部分位速率、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 寄存器设定一部分位速率，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。CPU 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。

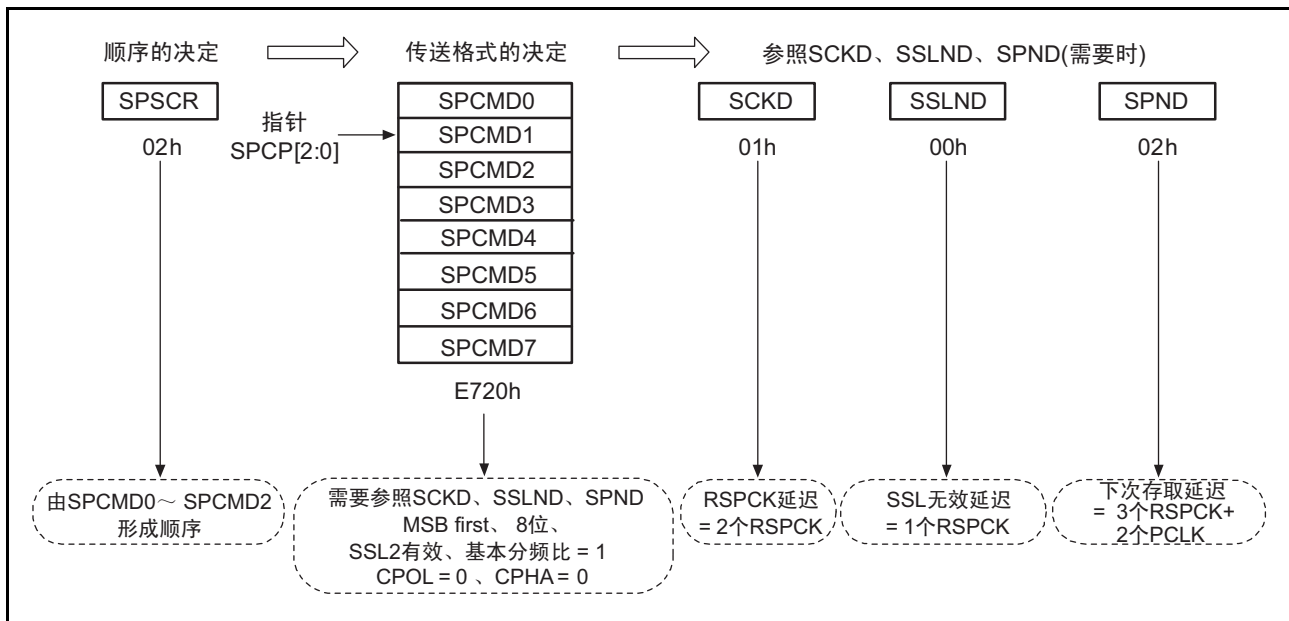


图 26.33 主控模式中的串行传送方式的决定方法 (时钟同步运行)



## (4) 初始化流程

在时钟同步运行的情况下，在主导模式中使用 RSPI 时的初始化流程例子如图 26.34 所示。有关中断控制器、DTC 和输入/输出端口的设定方法，请参照各模块的说明。

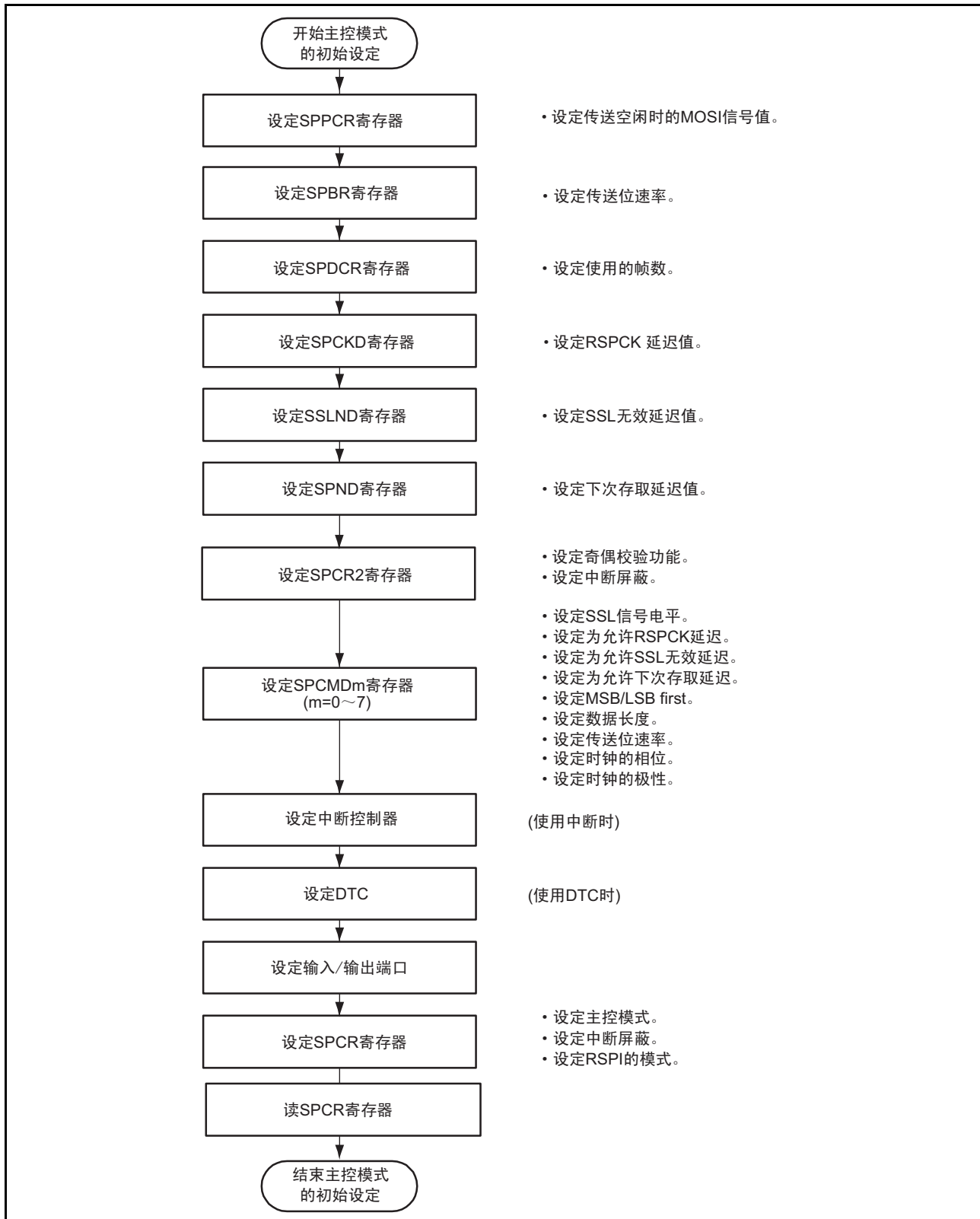


图 26.34 主导模式时的初始化流程例子 (时钟同步运行)

(5) 传送运行流程

在时钟同步运行时，主控模式的传送运行流程如图 26.35 所示。

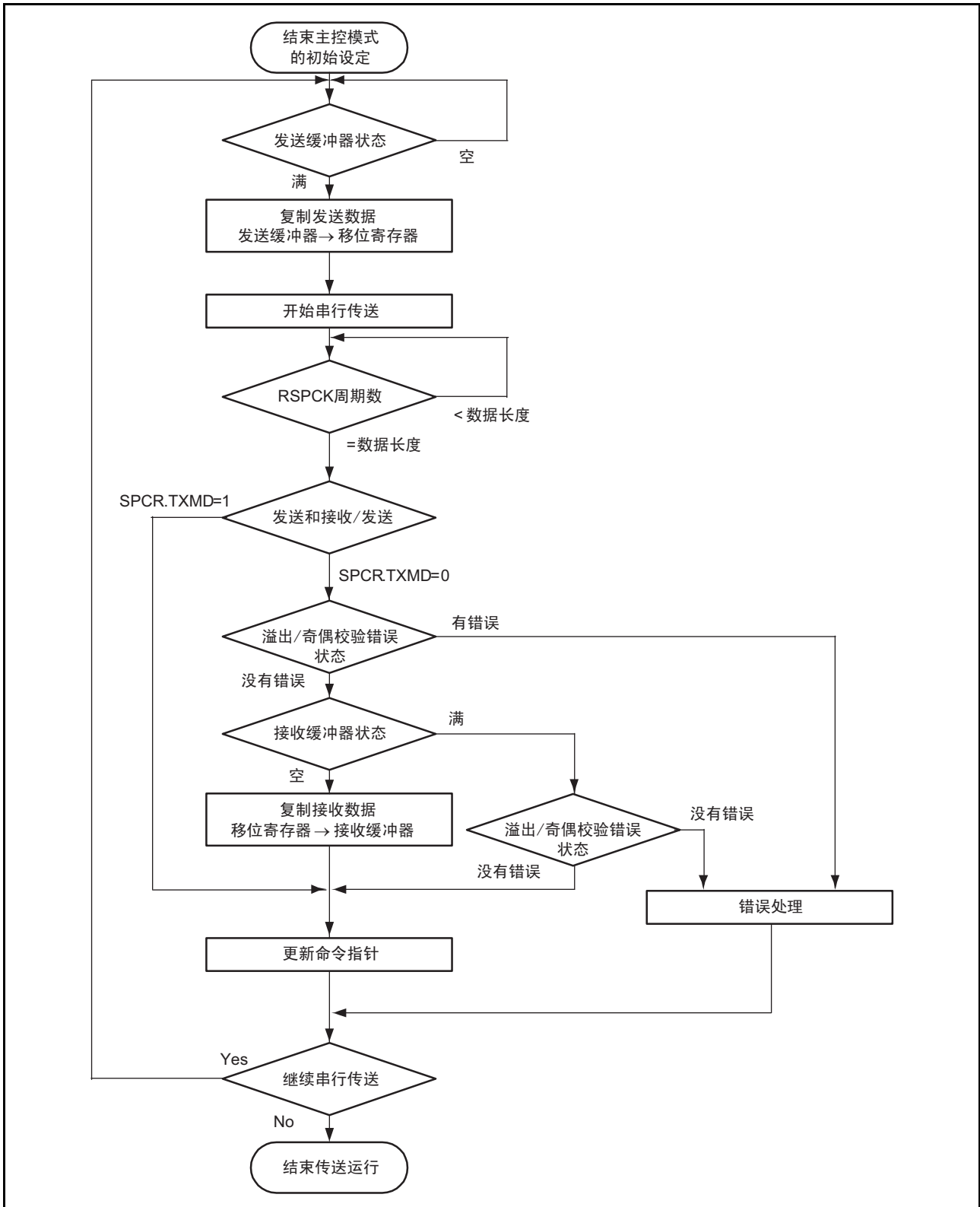


图 26.35 主控模式时的传送运行流程 (时钟同步运行)

### 26.3.11.2 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 RSPCK 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 SPCR.SPMS 位为“1”时，RSPI 随时驱动 MISO 输出信号。

有关 RSPI 传送格式的详细内容，请参照“26.3.4 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 RSPCK 边沿，就结束串行传送。在接收缓冲器有空间的情况下，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“26.3.4 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 26.36 所示。有关中断控制器、DTC 和输入/输出端口的设定方法，请参照各模块的说明。

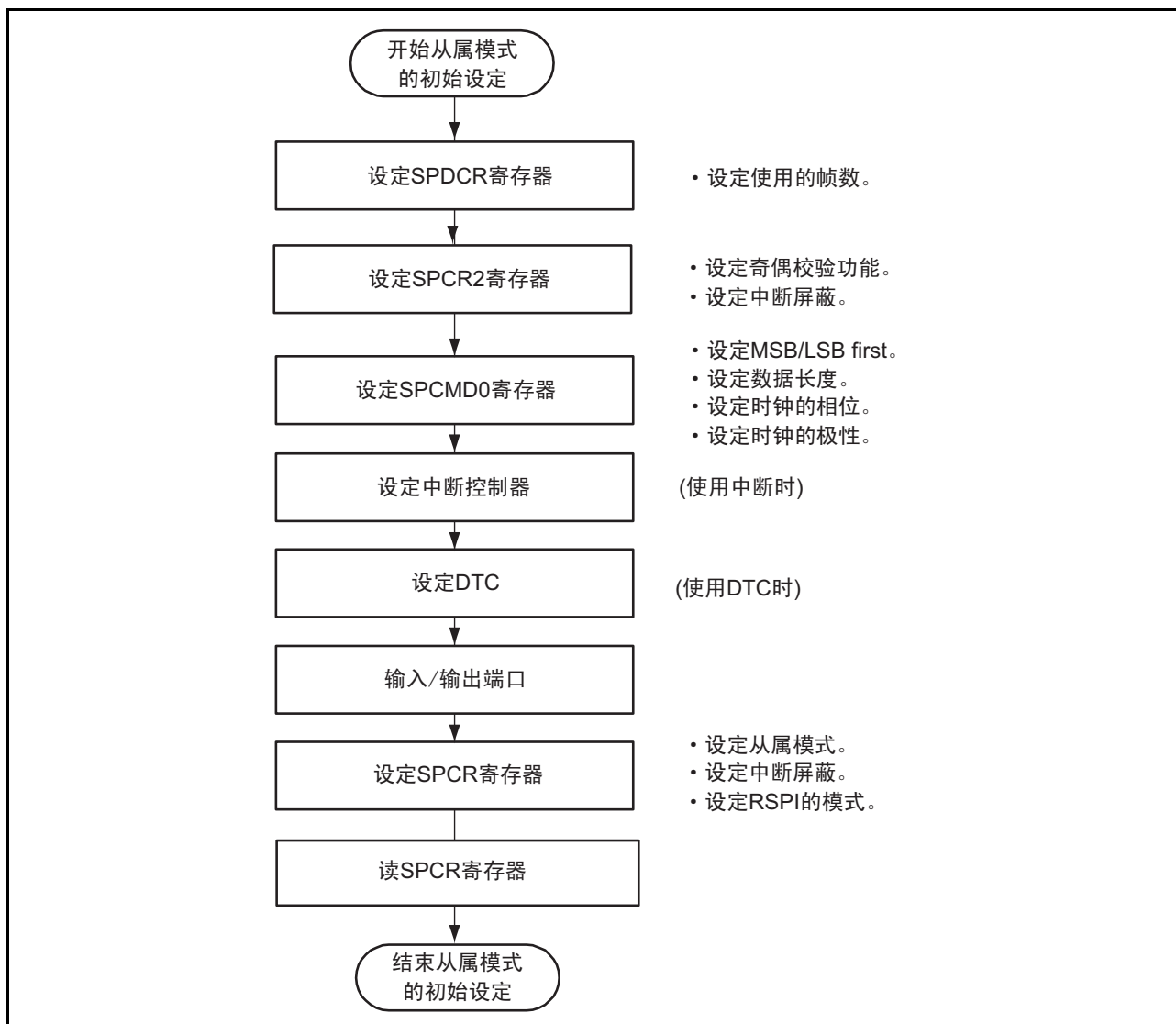


图 26.36 从属模式时的初始化流程例子 (时钟同步运行)

(4) 传送运行流程

在时钟同步运行时，RSPI 的传送运行流程如图 26.37 所示。

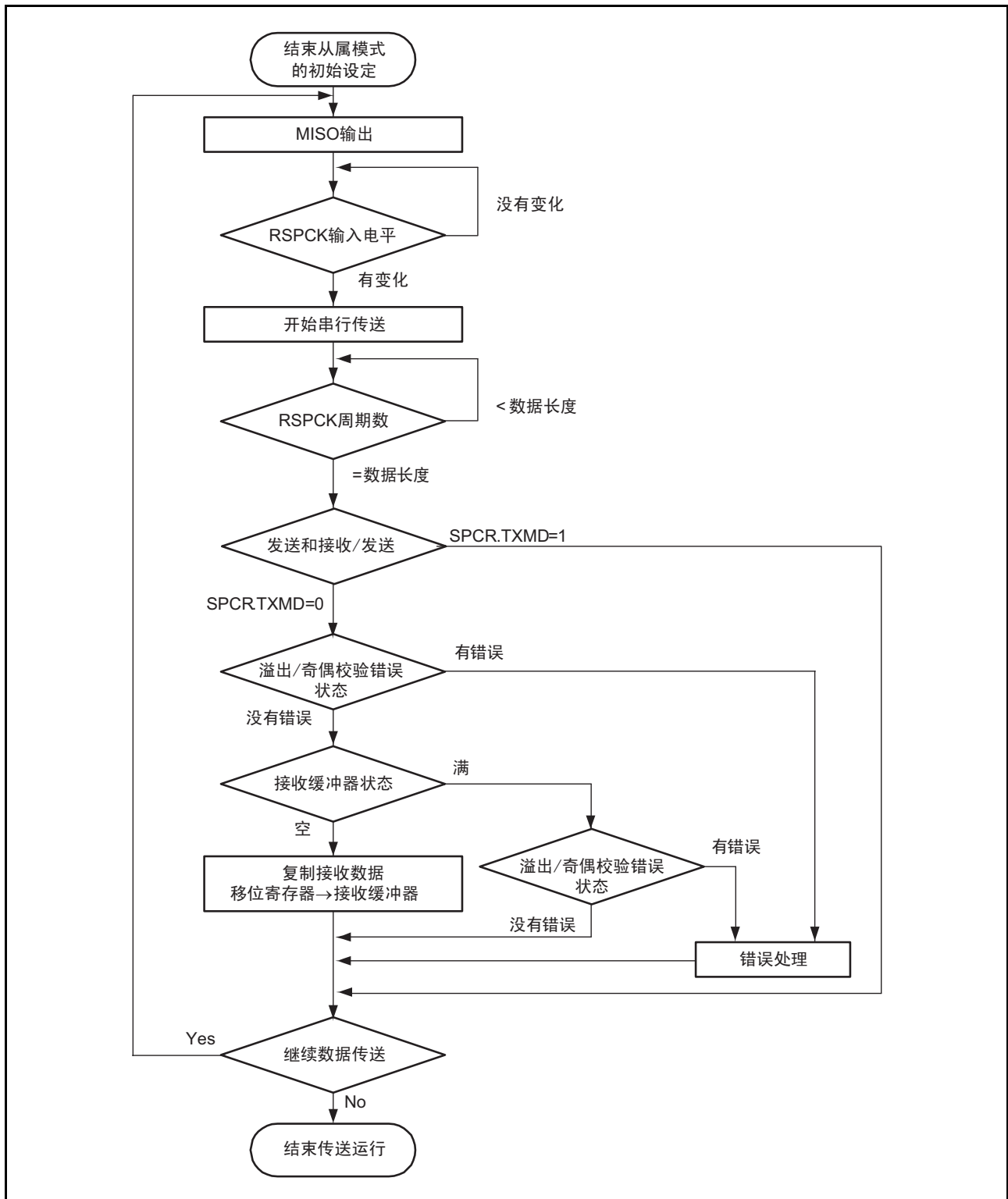


图 26.37 从属模式时的传送运行流程 (SPCMDm.CPHA=1) (时钟同步运行)

### 26.3.12 错误处理

RSPI 的错误处理如图 26.38 ~ 图 26.40 所示。对于在主控模式和从属模式中发生的错误，能通过以下的错误处理从错误状态返回。

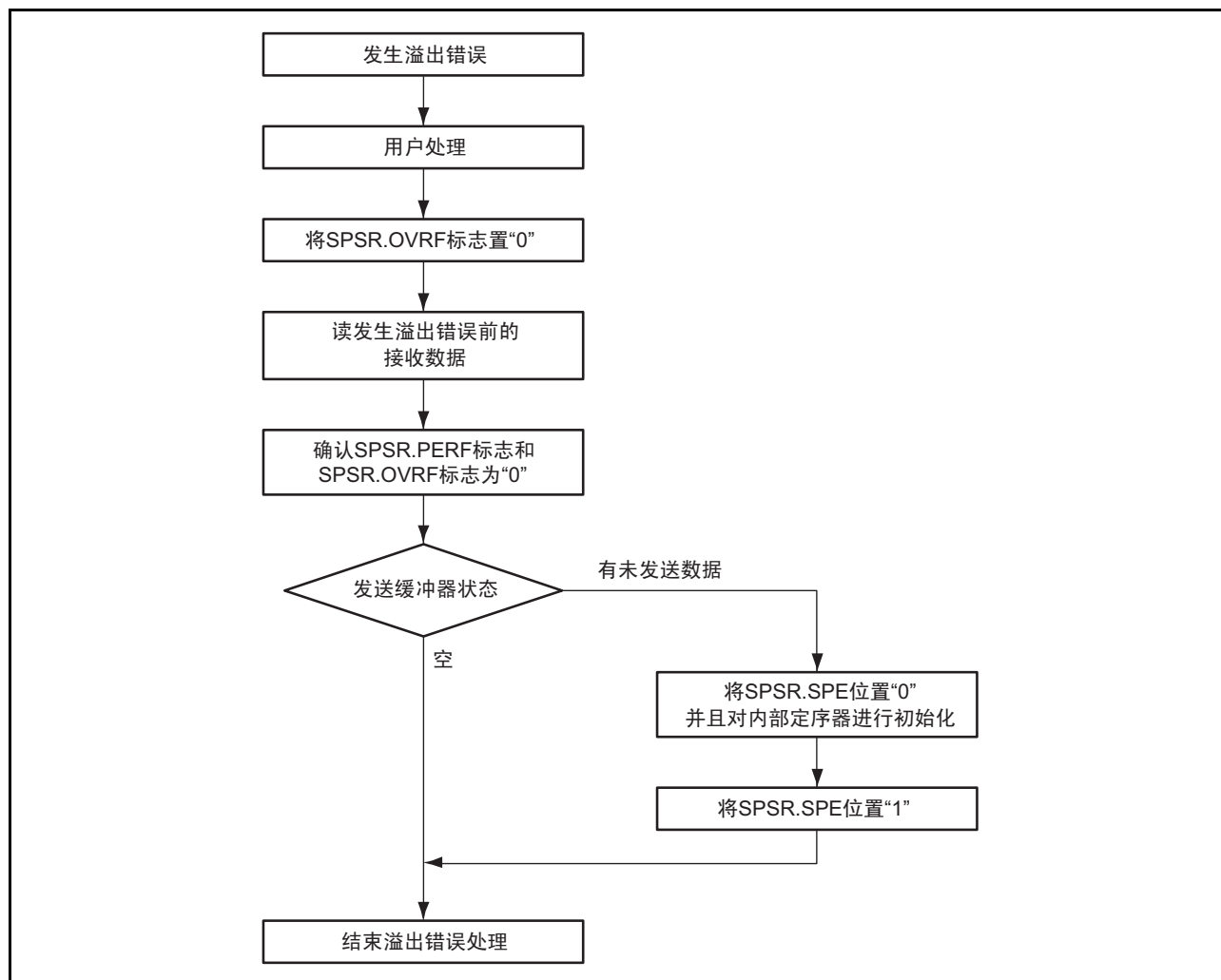


图 26.38 错误处理 (溢出错误)

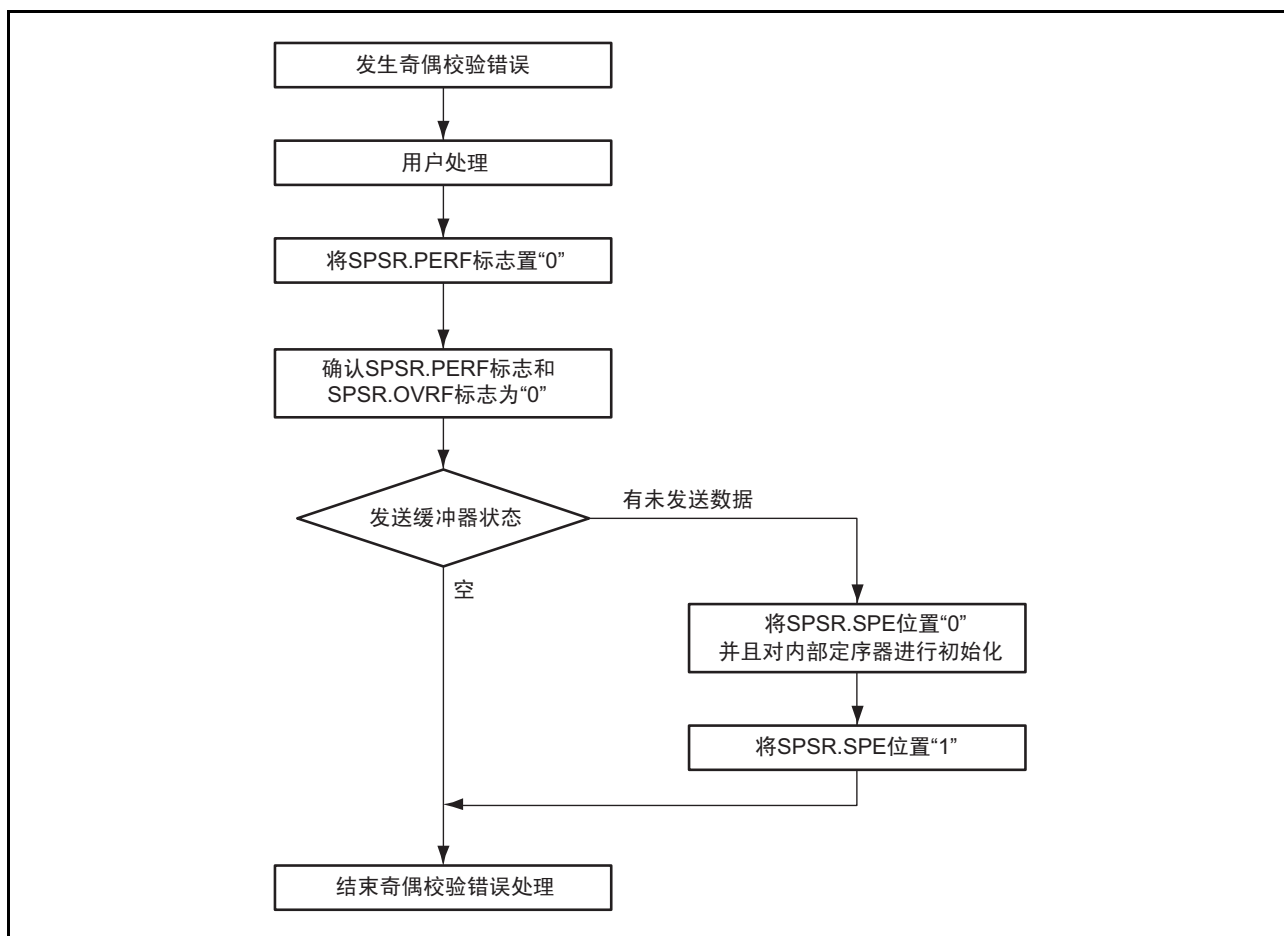


图 26.39 错误处理 (奇偶校验错误)

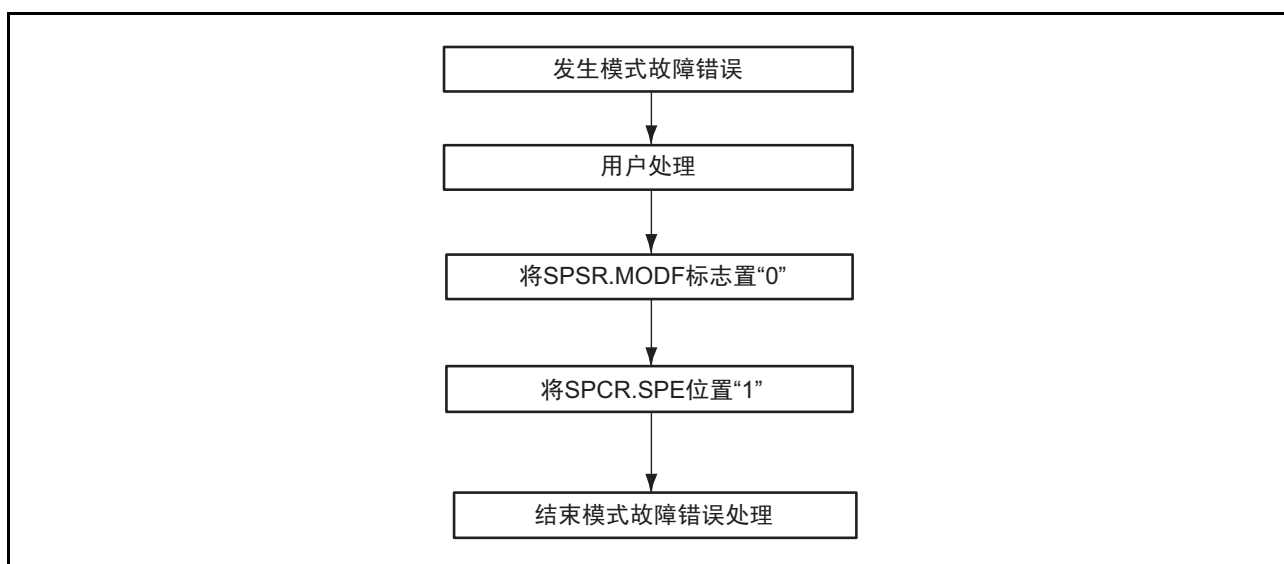


图 26.40 错误处理 (模式故障错误)

### 26.3.13 环回模式

在环回模式中 CPU 给 SPPCR.SPLP2 位或者 SPLP 位写“1”时，如果 SPCR.MSTR 位为“1”，RSPI 就切断 MISO 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 MOSI 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 MOSI 引脚和移位寄存器之间的线路；如果 SPCR.MSTR 位为“0”，就切断 MISO 引脚和移位寄存器之间的线路。

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。

SPPCR.SPLP2 位、SPPCR.SPLP 位的设定和接收数据的关系如表 26.13 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 26.41 所示。

表 26.13 SPPCR.SPLP2 位、SPPCR.SPLP 位的设定和接收数据

SPPCR.SPLP2 位	SPPCR.SPLP 位	接收数据
0	0	MOSI 引脚或者 MISO 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

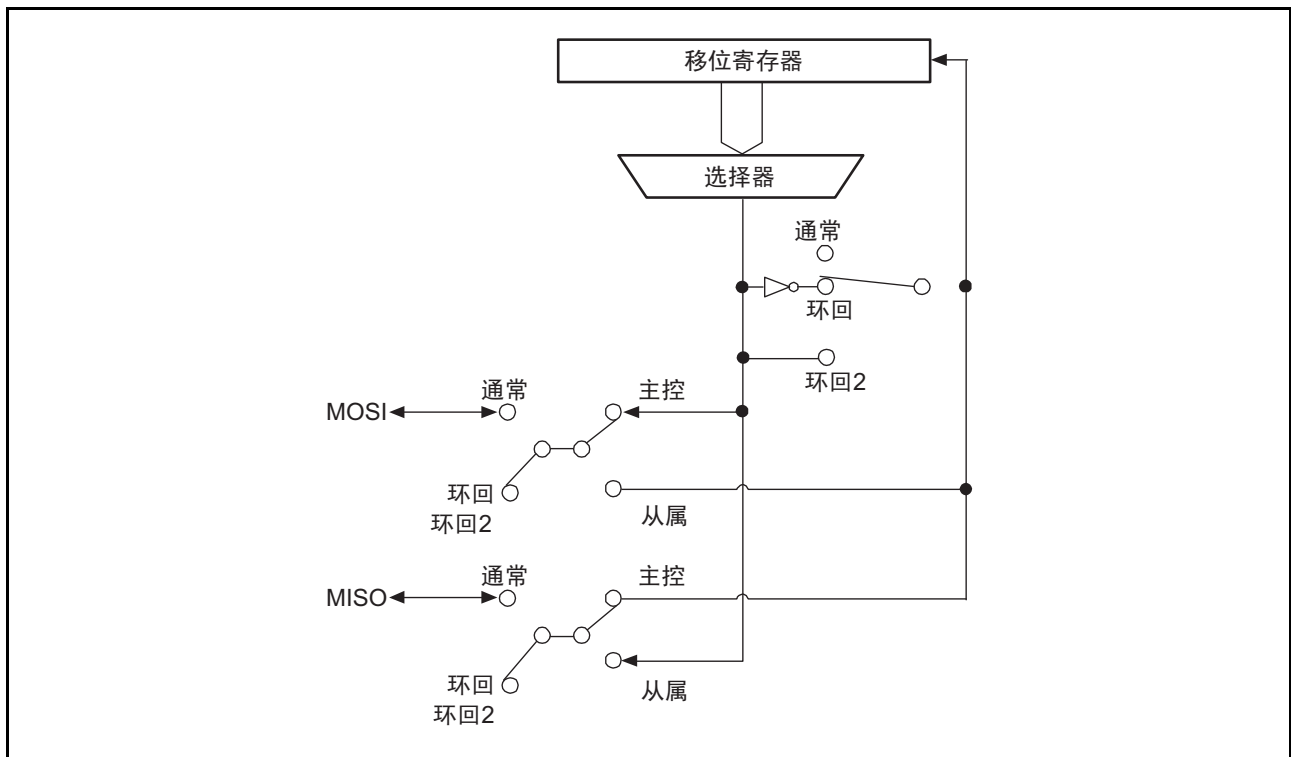


图 26.41 环回模式时的移位寄存器输入 / 输出结构 (主控模式)



### 26.3.14 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，必须按照图 26.42 所示的流程，进行奇偶校验电路的自诊断。

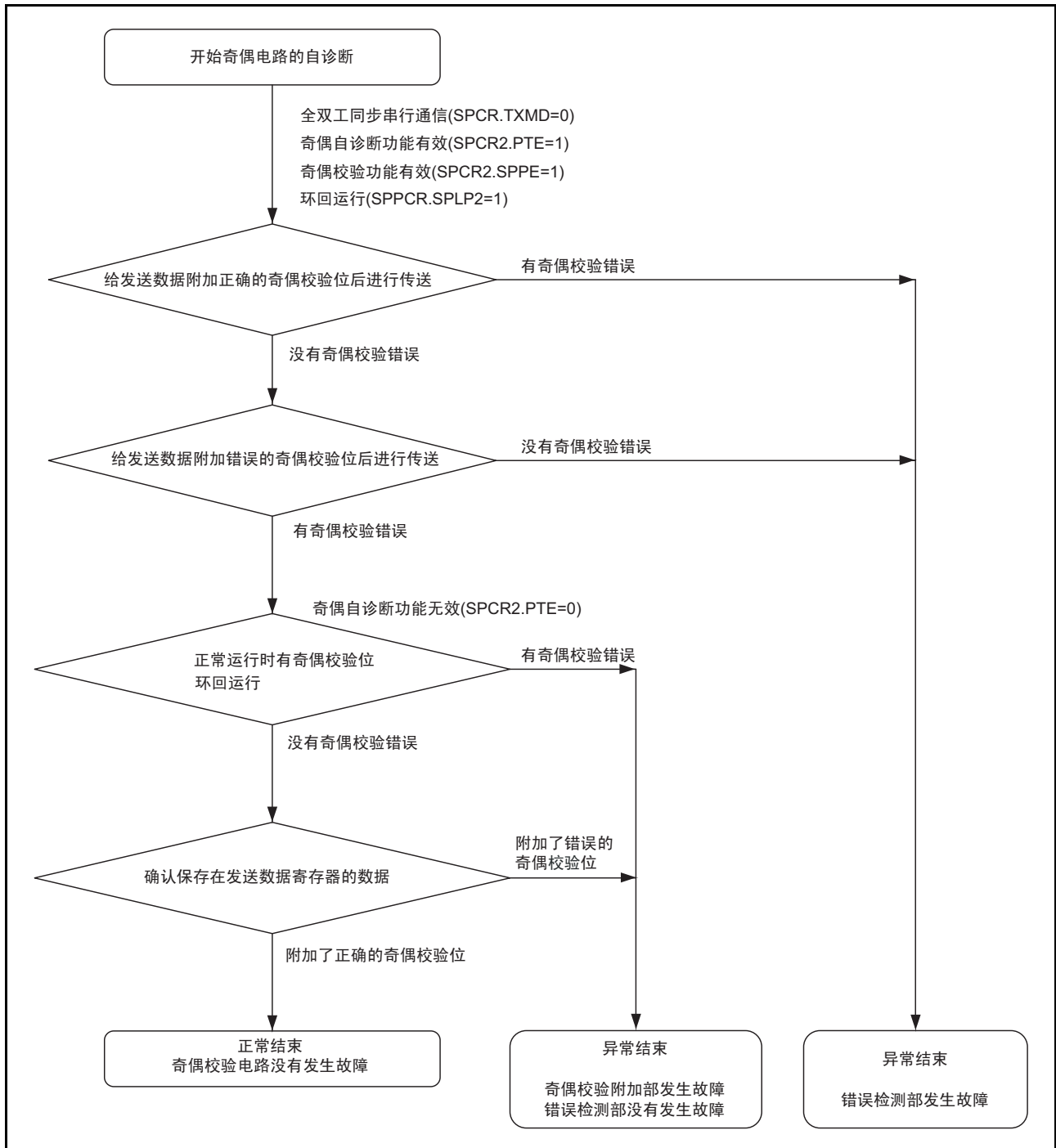


图 26.42 奇偶校验电路的自诊断流程

### 26.3.15 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，进行数据传送。

RSPI 的中断源如表 26.14 所示。一旦表 26.14 所示的中断条件成立，就发生中断。必须在数据传送过程中清除中断源。

在使用 DTC 进行发送或者接收时，必须先设定 DTC，然后在设定为允许状态后设定 RSPI。有关 DTC 的设定方法，请参照“14. 数据传送控制器 (DTC)”。

表 26.14 RSPI 的中断源

中断源	略称	中断条件	DTC 的启动
接收缓冲器满	SPRI0	(sprie=1)•(接收缓冲器满)	○
发送缓冲器空	SPTI0	(sptie=1)•(发送缓冲器空)	○
模式故障错误 溢出错误 奇偶校验错误	SPEI0	(speie=1)•{(modf=1)   (ovrf=1)   (perf=1)}	—
RSPI 空闲	SPII0	(spiie=1)•(idlnf=0)	—

## 26.4 使用时的注意事项

### 26.4.1 主控模式中奇偶校验功能有效时的发送运行

如果在主控模式中奇偶校验功能有效时进行附加的发送运行，就必须将各命令寄存器的下述内容设定为相同的值：

- 命令寄存器的传送位长的设定
- 命令寄存器的MSB first/LSB first 设定

## 27. LIN 模块 (LIN)

### 27.1 概要

LIN 模块是支持 LIN Specification Package Revision 1.3、2.0、2.1 的硬件 LIN 通信控制器，能自动进行帧通信和错误判断，并且内置 1 个通道的主控控制器。

LIN 模块的规格和框图分别如表 27.1 和图 27.1 所示。

表 27.1 LIN 模块的规格

项目	内容
协议	LIN Specification Package 1.3、2.0、2.1
通道数	1 个通道 (LIN 主控)
可变帧结构	<ul style="list-style-type: none"> <li>• 发送间隔宽度: 13 ~ 28 Tbit</li> <li>• 发送间隔定界符宽度: 1 ~ 4 Tbit</li> <li>• 字节间间隔 (报文头): 0 ~ 7 Tbit (Sync 场和 ID 场间的间隔) (注 1)</li> <li>• 响应间隔: 0 ~ 7 Tbit (注 1)</li> <li>• 字节间间隔: 0 ~ 3 Tbit (响应区域内的数据字节间的间隔)</li> <li>• 唤醒: 1 ~ 16 Tbit</li> </ul>
校验和	<ul style="list-style-type: none"> <li>• 在发送 / 接收时都进行自动运算</li> <li>• 能选择 "classic" 或者 "enhance" (每个帧都能更改)</li> </ul>
响应场数据字节数	0 ~ 8 字节
帧的发送方法	<ul style="list-style-type: none"> <li>• 通过 1 个发送开始请求发送报文头和响应的模式</li> <li>• 通过不同的发送开始请求发送报文头和响应的模式 (帧分隔模式)</li> </ul>
唤醒发送 / 接收	能用于 LIN 唤醒模式 <ul style="list-style-type: none"> <li>• 唤醒的发送功能 (1 ~ 16 Tbit)</li> <li>• 唤醒的接收</li> </ul> 对输入信号的 Low 电平宽度进行计数的功能
状态	<ul style="list-style-type: none"> <li>• 帧 / 唤醒发送结束</li> <li>• 报文头发送结束</li> <li>• 帧 / 唤醒接收结束 (注 2)</li> <li>• 数据 1 接收结束</li> <li>• 错误检测</li> <li>• 运行模式 (LIN 复位模式、LIN 唤醒模式、LIN 运行模式、LIN 自测试模式)</li> </ul>
错误状态	<ul style="list-style-type: none"> <li>• 位错误</li> <li>• 校验和错误</li> <li>• 帧超时错误</li> <li>• 物理总线错误</li> <li>• 帧错误</li> </ul>
波特率选择	能通过波特率发生器生成 LIN 规格的波特率
测试模式	用于用户评价的自测试模式
中断功能	<ul style="list-style-type: none"> <li>• 帧 / 唤醒发送结束</li> <li>• 帧 / 唤醒接收结束 (注 2)</li> <li>• 错误检测</li> </ul>

注 1. 因为是用同一寄存器进行设定，所以字节间间隔 (报文头) = 响应间隔

注 2. 接收唤醒表示对输入信号的 Low 电平宽度进行计数。

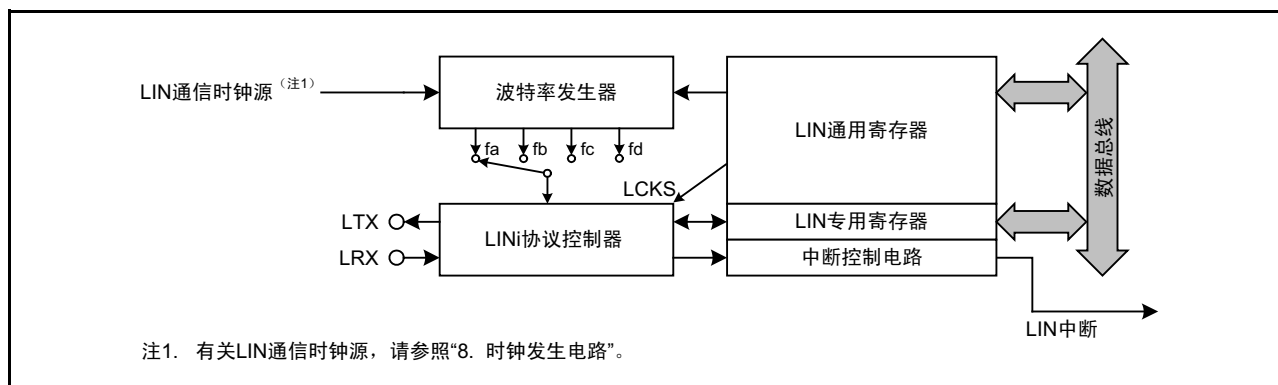


图 27.1 LIN 模块的框图

- LTX、LRX : LIN 模块的输入/输出引脚
- 波特率发生器 : 生成LIN的通信时钟
- LIN通用寄存器 : LIN 模块的通用寄存器
- 中断控制电路 : 控制通过LIN 模块生成的中断请求  
有LIN中断

LIN 模块使用的输入 / 输出引脚如表 27.2 所示。

表 27.2 LIN 模块的输入 / 输出引脚

模块符号	引脚名	输入 / 输出	功能
LIN0	LRX	输入	LIN 通信功能的输入引脚
	LTX	输出	LIN 通信功能的输出引脚

## 27.2 寄存器说明

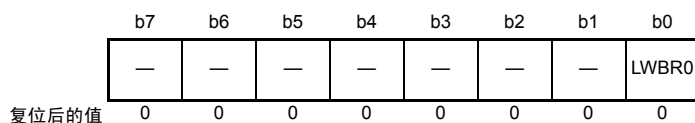
LIN 模块的寄存器一览表如表 27.3 所示。

表 27.3 LIN 模块的寄存器一览表

模块符号	寄存器名	符号	复位后的值	地址	存取长度
LIN0	LIN 唤醒波特率选择寄存器	LWBR	00h	0009 4001h	8
	LIN 波特率预分频器 0 寄存器	LBRP0	00h	0009 4002h	8、16
	LIN 波特率预分频器 1 寄存器	LBRP1	00h	0009 4003h	8、16
	LIN 自测试控制寄存器	LSTC	00h	0009 4004h	8
	模式寄存器	L0MD	00h	0009 4008h	8、16、32
	间隔场设定寄存器	L0BRK	00h	0009 4009h	8、16、32
	间隔设定寄存器	L0SPC	00h	0009 400Ah	8、16、32
	唤醒设定寄存器	L0WUP	00h	0009 400Bh	8、16、32
	中断允许寄存器	L0IE	00h	0009 400Ch	8、16
	错误检测允许寄存器	L0EDE	00h	0009 400Dh	8、16
	控制寄存器	L0C	00h	0009 400Eh	8
	发送控制寄存器	L0TC	00h	0009 4010h	8、16、32
	模式状态寄存器	L0MST	00h	0009 4011h	8、16、32
	状态寄存器	L0ST	00h	0009 4012h	8、16、32
	错误状态寄存器	L0EST	00h	0009 4013h	8、16、32
	响应场设定寄存器	L0RFC	00h	0009 4014h	8、16
	ID 缓冲寄存器	L0IDB	不定值	0009 4015h	8、16
	校验和缓冲寄存器	L0CBR	不定值	0009 4016h	8
	数据 1 缓冲寄存器	L0DB1	不定值	0009 4018h	8、16、32
	数据 2 缓冲寄存器	L0DB2	不定值	0009 4019h	8、16、32
	数据 3 缓冲寄存器	L0DB3	不定值	0009 401Ah	8、16、32
	数据 4 缓冲寄存器	L0DB4	不定值	0009 401Bh	8、16、32
	数据 5 缓冲寄存器	L0DB5	不定值	0009 401Ch	8、16、32
	数据 6 缓冲寄存器	L0DB6	不定值	0009 401Dh	8、16、32
数据 7 缓冲寄存器	L0DB7	不定值	0009 401Eh	8、16、32	
数据 8 缓冲寄存器	L0DB8	不定值	0009 401Fh	8、16、32	

### 27.2.1 LIN 唤醒波特率选择寄存器 (LWBR)

地址 0009 4001h



位	符号	位名	功能	R/W
b0	LWBR0	唤醒波特率选择模式位	0: 使用 LIN1.3 时 1: 使用 LIN2.0、2.1 时	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W

在 LIN 复位模式中设定 LWBR 寄存器。

#### LWBR0 位 (唤醒波特率选择模式位)

在使用 LIN Specification Package Revision 1.3 时, 必须将 LWBR 寄存器的 LWBR0 位置“0”, 由此最少能以 fLIN 的 2.5Tbit 对输入信号 Low 电平宽度进行测量。在使用 LIN Specification Package Revision 2.0、2.1 时, 必须将 LWBR 寄存器的 LWBR0 位置“1”, 由此最少能以 130 $\mu$ s 对输入信号 Low 电平宽度进行测量。

### 27.2.2 LIN 波特率预分频器 0 寄存器 (LBRP0)

地址 0009 4002h

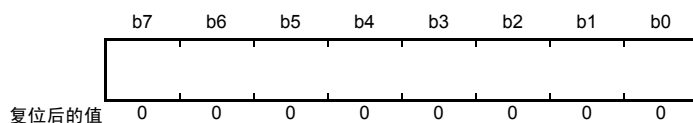


位	功能	设定范围	R/W
b7-b0	假设设定值为 N (0 ~ 255), 波特率预分频器对外围功能时钟进行 N + 1 分频。	00h ~ FFh	R/W

在 LIN 复位模式中设定 LBRP0 寄存器。

### 27.2.3 LIN 波特率预分频器 1 寄存器 (LBRP1)

地址 0009 4003h

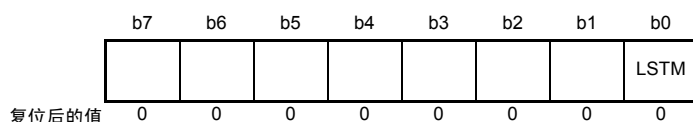


位	功能	设定范围	R/W
b7-b0	假设设定值为 M (0 ~ 255), 波特率预分频器对外围功能时钟进行 M + 1 分频。	00h ~ FFh	R/W

在 LIN 复位模式中设定 LBRP1 寄存器。

### 27.2.4 LIN 自测试控制寄存器 (LSTC)

地址 0009 4004h



位	符号	位名	功能	R/W
b7-b1			通过连续写“A7h” → “58h” → “01h”，进入 LIN 自测试模式。	R/W
b0	LSTM	自测试模式位	0: 非 LIN 自测试模式 1: LIN 自测试模式	R/W

LSTC 寄存器是用于解除 LIN 自测试模式保护的寄存器。

在 LIN 复位模式中设定 LSTC 寄存器。

通过连续写“A7h” → “58h” → “01h”，进入 LIN 自测试模式。当连续写成功并且进入 LIN 自测试模式时，LSTM 位变为“1”。在连续写期间不能进行其他的写操作。

有关向 LIN 自测试模式转移的方法，请参照“27.12 LIN 自测试模式”。

#### LSTM 位 (自测试模式位)

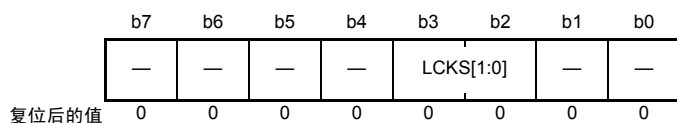
在转移到 LIN 自测试模式时，此位变为“1”。

有关 LIN 自测试模式的结束方法，请参照“27.12 LIN 自测试模式”。

除了给 LSTC 寄存器连续写“A7h” → “58h” → “01h”以外，即使给此位写“1”，此位的值也不变。

## 27.2.5 模式寄存器 (LOMD)

地址 0009 4008h



位	符号	位名	功能	R/W
b1-b0	—	保留位	读写值都为“0”。	R/W
b3-b2	LCKS[1:0]	LIN 系统时钟选择位	b3 b2 0 0: fa (波特率预分频器 0 的生成时钟) 0 1: fb (波特率预分频器 0 的生成时钟 /2) 1 0: fc (波特率预分频器 0 的生成时钟 /8) 1 1: fd (波特率预分频器 1 的生成时钟 /2)	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

在 LIN 复位模式中设定 LOMD 寄存器。

## LCKS[1:0] 位 (LIN 系统时钟选择位)

这些位选择输入到协议控制器的时钟。

当这些位为“00b”时，fa (波特率预分频器 0 的生成时钟) 被输入到协议控制器。

当这些位为“01b”时，fb (波特率预分频器 0 的生成时钟 /2) 被输入到协议控制器。

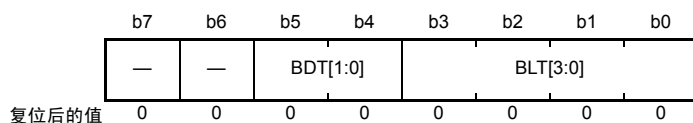
当这些位为“10b”时，fc (波特率预分频器 0 的生成时钟 /8) 被输入到协议控制器。

当这些位为“11b”时，fd (波特率预分频器 1 的生成时钟 /2) 被输入到协议控制器。



## 27.2.6 间隔场设定寄存器 (L0BRK)

地址 0009 4009h



位	符号	位名	功能	R/W
b3-b0	BLT[3:0]	发送间隔 (Low 电平) 宽度设定位	$b^3 \quad b^0$ 0 0 0 0: 13 Tbits 0 0 0 1: 14 Tbits 0 0 1 0: 15 Tbits : 1 1 1 0: 27 Tbits 1 1 1 1: 28 Tbits	R/W
b5-b4	BDT[1:0]	发送间隔定界符 (High 电平) 宽度设定位	$b^5 \quad b^4$ 0 0: 1 Tbit 0 1: 2 Tbits 1 0: 3 Tbits 1 1: 4 Tbits	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W

在 LIN 复位模式中设定 L0BRK 寄存器。

通过组合设定值，有可能使 1 帧的长度大于帧超时时间，因此必须设定适当的值。

## BLT[3:0] 位 (发送间隔 (Low 电平) 宽度设定位)

此位设定发送帧报文头部的间隔 (Low 电平) 宽度。

能设定为 13 Tbits ~ 28 Tbits。

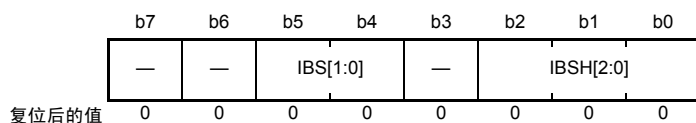
## BDT[1:0] 位 (发送间隔定界符 (High 电平) 宽度设定位)

此位设定发送帧报文头部的间隔定界符 (High 电平) 宽度。

能设定为 1 Tbit ~ 4 Tbits。

## 27.2.7 间隔设定寄存器 (L0SPC)

地址 0009 400Ah



位	符号	位名	功能	R/W
b2-b0	IBSH[2:0]	字节间间隔 (报文头) / 响应间隔 设定位	b2 b0 0 0 0: 0 Tbit 0 0 0: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits	R/W
b3	—	保留位	读写值都为“0”。	R/W
b5-b4	IBS[1:0]	字节间间隔设定位	b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W

在 LIN 复位模式中设定 L0SPC 寄存器。

只在发送 (报文头、响应) 时有效。在接受响应时无效。

通过组合设定值, 有可能使 1 帧的长度大于帧超时时间, 因此必须设定适当的值。

## IBSH[2:0] 位 (字节间间隔 / 响应间隔设定位)

此位设定发送帧报文头部的字节间间隔 (报文头) 和响应间隔的宽度。

能设定为 0 Tbit ~ 7 Tbits。

字节间间隔和响应间隔的值相同。

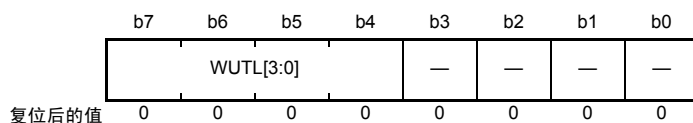
## IBS[1:0] 位 (字节间间隔设定位)

此位设定发送帧响应部的字节间间隔的宽度。

能设定为 0 Tbit ~ 3 Tbits。

## 27.2.8 唤醒设定寄存器 (LOWUP)

地址 0009 400Bh



位	符号	位名	功能	R/W
b3-b0	—	设定位	读写值都为“0”。	R/W
b7-b4	WUTL[3:0]	唤醒发送 Low 电平宽度设定位	b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits	R/W

在 LIN 复位模式中设定 LOWUP 寄存器。

## WUTL[3:0] 位 (唤醒发送 Low 电平宽度设定位)

此位设定发送唤醒帧时的 Low 电平宽度。

能设定为 1 Tbit ~ 16 Tbits

## 27.2.9 中断允许寄存器 (LOIE)

地址 0009 400Ch

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	ERRIE	FRCIE	FTCIE
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	FTCIE	帧 / 唤醒发送结束中断允许位	0: 禁止帧 / 唤醒发送结束中断 1: 允许帧 / 唤醒发送结束中断	R/W
b1	FRCIE	帧 / 唤醒接收结束中断允许位	0: 禁止帧 / 唤醒接收结束中断 1: 允许帧 / 唤醒接收结束中断	R/W
b2	ERRIE	错误检测中断允许位	0: 禁止错误检测中断 1: 允许错误检测中断	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W

在 LIN 复位模式中设定 LOIE 寄存器。

## FTCIE 位 (帧 / 唤醒发送结束中断允许位)

此位设定禁止或者允许帧发送结束或者唤醒帧发送结束时的中断。

当此位为“0”时，在 LOST.FTC 标志变为“1”时不发生 LIN 中断。

当此位为“1”时，在 LOST.FTC 标志变为“1”时发生 LIN 中断。

## FRCIE 位 (帧 / 唤醒接收结束中断允许位)

此位设定禁止或者允许帧接收结束或者唤醒帧接收 (对输入信号 Low 电平宽度进行计数) 结束时的中断。

当此位为“0”时，在 LOST.FRC 标志变为“1”时不发生 LIN 中断。

当此位为“1”时，在 LOST.FRC 标志变为“1”时发生 LIN 中断。

## ERRIE 位 (错误检测中断允许位)

此位设定禁止或者允许检测到错误时的中断。

当此位为“0”时，在 LOST.ERR 标志变为“1”时不发生 LIN 中断。

当此位为“1”时，在 LOST.ERR 标志变为“1”时发生 LIN 中断。

中断的发生源有位错误、物理总线错误、帧超时错误、帧错误以及校验和错误。

能通过 LOEDE 寄存器设定禁止或者允许检测位错误、物理总线错误、帧超时错误、帧错误以及校验和错误。

## 27.2.10 错误检测允许寄存器 (LOEDE)

地址 0009 400Dh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	FERE	FTERE	PBERE	BERE
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	BERE	位错误检测允许位	0: 禁止位错误的检测 1: 允许位错误的检测	R/W
b1	PBERE	物理总线错误检测允许位	0: 禁止物理总线错误的检测 1: 允许物理总线错误的检测	R/W
b2	FTERE	帧超时错误检测允许位	0: 禁止帧超时错误的检测 1: 允许帧超时错误的检测	R/W
b3	FERE	帧错误检测允许位	0: 禁止帧错误的检测 1: 允许帧错误的检测	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

在 LIN 复位模式中设定 LOEDE 寄存器。

## BERE 位 (位错误检测允许位)

此位设定禁止或者允许检测位错误。  
 当此位为“0”时，不检测位错误。  
 当此位为“1”时，检测位错误。  
 此位为“1”时的检测结果反映到 LOEST.BER 标志。  
 位错误的详细内容请参照“27.10 错误状态”。

## PBERE 位 (物理总线错误检测允许位)

此位设定禁止或者允许检测物理总线错误。  
 当此位为“0”时，不检测物理总线错误。  
 当此位为“1”时，检测物理总线错误。  
 此位为“1”时的检测结果反映到 LOEST.PBER 标志。  
 物理总线错误的详细内容请参照“27.10 错误状态”。

## FTERE 位 (帧超时错误检测允许位)

此位设定禁止或者允许检测帧超时错误。  
 当此位为“0”时，不检测帧超时错误。  
 当此位为“1”时，检测帧超时错误。  
 此位为“1”时的检测结果反映到 LOEST.FTER 标志。  
 帧超时错误的详细内容请参照“27.10 错误状态”。

## FERE 位 (帧错误检测允许位)

此位设定禁止或者允许检测帧错误。  
 当此位为“0”时，不检测帧错误。  
 当此位为“1”时，检测帧错误。  
 此位为“1”时的检测结果反映到 LOEST.FER 标志。  
 帧错误的详细内容请参照“27.10 错误状态”。

## 27.2.11 控制寄存器 (LOC)

地址 0009 400Eh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	OM1	OM0
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	OM0	LIN 复位的位	0: LIN 复位模式 1: 非 LIN 复位模式	R/W
b1	OM1	LIN 模式选择位	0: LIN 唤醒模式 1: LIN 运行模式	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

在解除 LIN 复位模式时，如果给 LOC 寄存器设定“01h”，就转移到 LIN 唤醒模式；如果给 LOC 寄存器设定“03h”，就转移到 LIN 运行模式。

在 LIN 自测试模式中，如果转移到 LIN 自测试模式，就必须给 LOC 寄存器设定“03h”。

## OM0 位 (LIN 复位的位)

此位选择是转移到 LIN 复位模式还是解除 LIN 复位模式。

如果将此位置“0”，就为 LIN 复位模式。

如果将此位置“1”，就解除 LIN 复位模式。

## OM1 位 (LIN 模式选择位)

此位选择解除 LIN 复位模式时的 LIN 运行模式 (LIN 唤醒模式、LIN 运行模式)。

如果将此位置“0”，就为 LIN 唤醒模式。

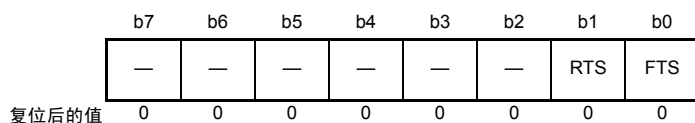
如果将此位置“1”，就为 LIN 运行模式。

此寄存器只在 LOMST.OMM0 位为“1”时有效。

在 LOTC.FTS 位为“1”期间不能写此位。

## 27.2.12 发送控制寄存器 (L0TC)

地址 0009 4010h



位	符号	位名	功能	R/W
b0	FTS	帧发送 / 唤醒发送或者接收的开始位	0: 停止帧发送 / 唤醒发送或者接收 1: 开始帧发送 / 唤醒发送或者接收	R/W
b1	RTS	响应发送的开始位	0: 在帧分隔模式中停止响应发送 1: 在帧分隔模式中开始响应发送	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

## FTS 位 (帧发送 / 唤醒发送或者接收的开始位)

在开始发送帧 / 唤醒时, 必须将此位置“1”。

在接收唤醒 (对输入信号 Low 电平宽度进行计数) 时, 必须将此位置“1”。

在进行通信时, 此位变为“1”; 在没有进行通信或者转移到 LIN 模式模式时, 此位变为“0”。

此位不能写“0”, 只能写“1”。在发送结束后自动变为“0”。

## RTS 位 (响应发送的开始位)

在开始发送响应时, 必须将此位置“1”。

在进行通信时, 此位变为“1”; 在没有进行通信或者转移到 LIN 模式模式时, 此位变为“0”。

此位不能写“0”, 只能写“1”。在发送结束后自动变为“0”。如果要给 RTS 位写“1”, 就必须通过 MOV 指令写“02h”。

只有在 LORFC.FSM 位为“1” (帧分隔模式)、FTS 位为“1” (开始帧发送 / 唤醒发送或者接收) 时才能设定此位。

## 27.2.13 模式状态寄存器 (LOMST)

地址 0009 4011h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	OMM1	OMM0
0	0	0	0	0	0	0	0

复位后的值

位	符号	位名	功能	R/W
b0	OMM0	LINO 复位状态监视位	0: LINO 复位模式 1: 非 LINO 复位模式	R/W
b1	OMM1	LINO 模式状态监视位	0: LINO 唤醒模式 1: LINO 运行模式	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

OMM0 位 (LINO 复位状态的监视位)

OMM1 位 (LINO 模式状态的监视位)

此位能确认正在进行的运行模式。



## 27.2.14 状态寄存器 (LOST)

地址 0009 4012h

b7	b6	b5	b4	b3	b2	b1	b0
HTRC	D1RC	—	—	ERR	—	FRC	FTC
复位后的值	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	FTC	帧 / 唤醒发送结束标志	0: 发送未结束 1: 帧或者唤醒的发送结束	R/W
b1	FRC	帧 / 唤醒接收结束标志	0: 接收未结束 1: 帧或者唤醒的接收结束	R/W
b2	—	保留位	读写值都为“0”。	R/W
b3	ERR	错误检测标志	0: 未检测到错误 1: 检测到错误	R
b5-b4	—	保留位	读写值都为“0”。	R/W
b6	D1RC	数据 1 接收结束标志	0: 接收未结束 1: 数据 1 的接收结束	R/W
b7	HTRC	报文头发送结束标志	0: 发送未结束 1: 报文头的发送结束	R/W

在转移到 LIN 复位模式以及开始进行下一次通信时，LOST 寄存器自动变为“00h”。

在 LIN 复位模式中保持“00h”。

在 L0TC.FTS 标志为“1”（开始帧发送 / 唤醒发送或者接收）的期间，不能写 LOST 寄存器。

## FTC 标志（帧 / 唤醒发送结束标志）

只能给此标志写“0”。如果写“1”，保持写前的值。

在结束帧发送或者唤醒发送时，此标志为“1”。此时，如果 LOIE.FTCIE 位为“1”（允许中断），就发生中断。如果要在下一次通信开始前将此标志置“0”，就必须在 LIN 运行模式中给此标志写“0”。

## FRC 标志（帧 / 唤醒接收结束标志）

只能给此标志写“0”。如果写“1”，保持写前的值。

在结束帧接收或者唤醒接收时，此标志为“1”。此时，如果 LOIE.FTCIE 位为“1”（允许中断），就发生中断。如果要在下一次通信开始前将此标志置“0”，就必须在 LIN 运行模式中给此标志写“0”。

## ERR 标志（错误检测标志）

在检测到错误时，此标志为“1”。此时，如果 LOIE.ERRIE 位为“1”（允许中断），就发生中断。如果要在下一次通信开始前将此标志置“0”，就必须在 LIN 运行模式中给 LOEST.CSER、LOEST.FER、LOEST.FTER、LOEST.PBER 和 LOEST.BER 标志写“0”。

## D1RC 标志（数据 1 接收结束标志）

只能给此标志写“0”。如果写“1”，保持写前的值。

在结束数据 1 的接收时，虽然此标志为“1”，但是不发生中断。如果要在下一次通信开始前将此标志置“0”，就必须在 LIN 运行模式中给此标志写“0”。

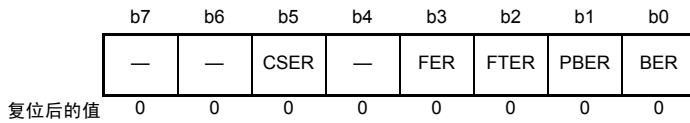
## HTRC 标志（报文头发送结束标志）

只能给此标志写“0”。如果写“1”，保持写前的值。

在结束报文头的接收时，虽然此标志为“1”，但是不发生中断。如果要在下一次通信开始前将此标志置“0”，就必须在 LIN 运行模式中给此标志写“0”。

## 27.2.15 错误状态寄存器 (LOEST)

地址 0009 4013h



位	符号	位名	功能	R/W
b0	BER	位错误标志	0: 未检测到位错误 1: 检测到位错误	R/W
b1	PBER	物理总线错误标志	0: 未检测到物理总线错误 1: 检测到物理总线错误	R/W
b2	FTER	帧超时错误标志	0: 未检测到帧超时错误 1: 检测到帧超时错误	R/W
b3	FER	帧错误标志	0: 未检测到帧错误 1: 检测到帧错误	R/W
b4	—	保留位	读写值都为“0”。	R/W
b5	CSER	校验和错误标志	0: 未检测到校验和错误 1: 检测到校验和错误	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W

在转移到 LIN 复位模式或者开始进行下一次通信时，LOEST 寄存器自动变为“00h”。

在 LIN 复位模式中保持“00h”。

在 L0TC.FTS 标志为“1”（开始帧发送 / 唤醒发送或者接收）的期间，不能写 LOEST 寄存器。

## BER 标志（位错误标志）

只能给此标志写“0”。如果写“1”，保持写前的值。

在检测到位错误时，此标志为“1”。如果要在下一次通信开始前将此标志置“0”，就必须在 LIN 运行模式中给此标志写“0”。

## PBER 标志（物理总线错误标志）

只能给此标志写“0”。如果写“1”，保持写前的值。

在检测到物理总线错误时，此标志为“1”。如果要在下一次通信开始前将此标志置“0”，就必须在 LIN 运行模式中给此标志写“0”。

## FTER 标志（帧超时错误标志）

只能给此标志写“0”。如果写“1”，保持写前的值。

在检测到帧超时错误时，此标志为“1”。如果要在下一次通信开始前将此标志置“0”，就必须在 LIN 运行模式中给此标志写“0”。

## FER 标志（帧错误标志）

只能给此标志写“0”。如果写“1”，保持写前的值。

在检测到帧错误时，此标志为“1”。如果要在下一次通信开始前将此标志置“0”，就必须在 LIN 运行模式中给此标志写“0”。

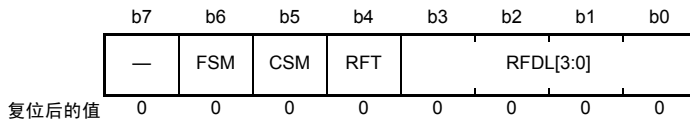
## CSER 标志（校验和错误标志）

只能给此标志写“0”。如果写“1”，保持写前的值。

在检测到校验和错误时，此标志为“1”。如果要在下一次通信开始前将此标志置“0”，就必须在 LIN 运行模式中给此标志写“0”。

## 27.2.16 响应场设定寄存器 (L0RFC)

地址 0009 4014h



位	符号	位名	功能	R/W
b3-b0	RFDL[3:0]	响应场长度设定位	b3 b0 0 0 0 0: 0 字节+校验和 0 0 0 1: 1 字节+校验和 0 0 1 0: 2 字节+校验和 : 0 1 1 1: 7 字节+校验和 1 0 0 0: 8 字节+校验和 上述以外: 不能设定	R/W
b4	RFT	响应场发送 / 接收方向设定位	0: 接收 1: 发送	R/W
b5	CSM	校验和选择位	0: classic 1: enhance	R/W
b6	FSM	帧分隔模式选择位	0: 非帧分隔模式 1: 帧分隔模式	R/W
b7	—	保留位	读写值都为“0”。	R/W

只有在 L0TC.FTS 位为“0”（停止帧发送 / 唤醒发送或者接收）时才能设定 L0RFC 寄存器。

## RFDL[3:0] 位（响应场长度设定位）

这些位设定响应场的长度。

数据长度能设定为 0 ~ 8 字节。数据长度不包括校验和的大小。

## RFT 位（响应场发送 / 接收方向设定位）

当此位为“0”时，在响应场进行接收。另外，在 LIN 唤醒模式中，进行唤醒接收（对输入信号 Low 电平宽度进行计数）。

当此位为“1”时，在响应场进行发送。另外，在 LIN 唤醒模式中，进行唤醒发送。

## CSM 位（校验和选择位）

此位设定校验和的方式。

当此位为“0”时，校验和的方式为 classic。

当此位为“1”时，校验和的方式为 enhance。

在使用帧超时错误（LOEDE.FTERE 位为“1”）时，帧超时时间因 CSM 位的设定而不同。详细内容请参照“27.10 错误状态”。

## FSM 位（帧分隔模式选择位）

当此位为“0”时，不为帧分隔模式。

当此位为“1”时，为帧分隔模式。

在进行响应接收（RFT 位为“0”）时，不受 FSM 位设定的影响。

在转移到 LIN 自测试模式前必须将此位设定为“0”。

帧分隔模式的详细内容请参照“27.7.1.1 帧分隔模式”。

## 27.2.17 ID 缓冲寄存器 (L0IDB)

地址 0009 4015h



位	符号	位名	功能	R/W
b5-b0	ID	ID 设定位	设定在 ID 场发送的 6 位 ID 值	R/W
b7-b6	IDP	奇偶校验设定位	设定通过 ID 发送的奇偶校验 (P) 位	R/W

只有在 L0TC.FTS 位为“0”（停止帧发送 / 唤醒发送或者接收）时才能设定 L0IDB 寄存器。

在 LIN 自测试模式中 L0IDB 寄存器如下所示。

- 当 LORFC.RFT 位为“1”（发送）时  
能读发送值的取反值。能写通信前的发送值。
- 当 LORFC.RFT 位为“0”时（接收）时  
能读接收值的取反值。能写通信前的接收值。

## ID 位 (ID 设定位)

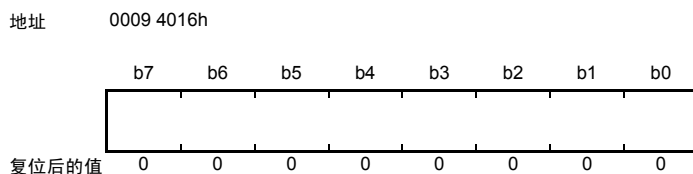
此位设定在 LIN 帧的 ID 场发送的 6 位 ID。

## IDP 位 (奇偶校验设定位)

此位设定在 LIN 帧的 ID 场发送的奇偶校验位 (P0、P1)。

如果奇偶校验不自动进行运算，就必须设定运算值。如果设定错误的运算结果就发送错误的运算结果。

## 27.2.18 校验和缓冲寄存器 (L0CBR)



位	功能	R/W
b7-b0	保存校验和的发送 / 接收数据。	R/W

只有在 L0TC.FTS 位为“0”（停止帧发送 / 唤醒发送或者接收）时才能设定 L0CBR 寄存器。

在 LIN 运行模式中 L0CBR 寄存器如下所示。

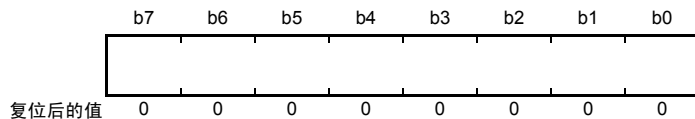
- 当 LORFC.RFT 位为“1”（发送）时  
能读发送值。写操作无效。
- 当 LORFC.RFT 位为“0”时（接收）时  
能读接收值。写操作无效。

在 LIN 自测试模式中 L0CBR 寄存器如下所示。

- 当 LORFC.RFT 位为“1”（发送）时  
能读发送值的取反值。写操作无效。
- 当 LORFC.RFT 位为“0”时（接收）时  
能读接收值的取反值。能写通信前的接收值。

## 27.2.19 数据 n 缓冲寄存器 (L0DBn) (n=1 ~ 8)

地址 L0DB1 0009 4018h、L0DB2 0009 4019h、L0DB3 0009 401Ah、L0DB4 0009 401Bh、L0DB5 0009 401Ch、L0DB6 0009 401Dh、  
L0DB7 0009 401Eh、L0DB8 0009 401Fh



位	功能	设定范围	R/W
b7-b0	设定发送数据或者读接收数据。	00h~FFh	R/W

必须在以下的状态下设定 L0DBn 寄存器。

响应发送的情况:

- LORFC.RFT 位为“1” (发送)
- LORFC.FSM 位为“0” (非帧分隔模式)
- LOTC.FTC 位为“0” (停止帧发送/唤醒发送或者接收)

或者

- LORFC.RFT 位为“1” (发送)
- LORFC.FSM 位为“1” (帧分隔模式)
- LOTC.RTS 位为“0” (停止响应发送)

响应接收的情况:

接收数据不被盖写。另外, 在检测到错误时, 只保存接收被中止前的数据。

在 LIN 自测试模式中 L0DBn 寄存器如下所示。

- 当 LORFC.RFT 位为“1” (发送) 时  
能读发送值的取反值。能写通信前的发送值。
- 当 LORFC.RFT 位为“0” 时 (接收) 时  
能读接收值的取反值。能写通信前的接收值。

### 27.3 运行模式

LIN 模块有以下 4 种运行模式。

- LIN 复位模式
- LIN 运行模式
- LIN 唤醒模式
- LIN 自测试模式

在 LIN 复位模式中，因为停止向 LIN 模块提供时钟，所以能降低功耗。

运行模式的转移如图 27.2 所示。能在各运行模式中进行的运行如表 27.4 所示。

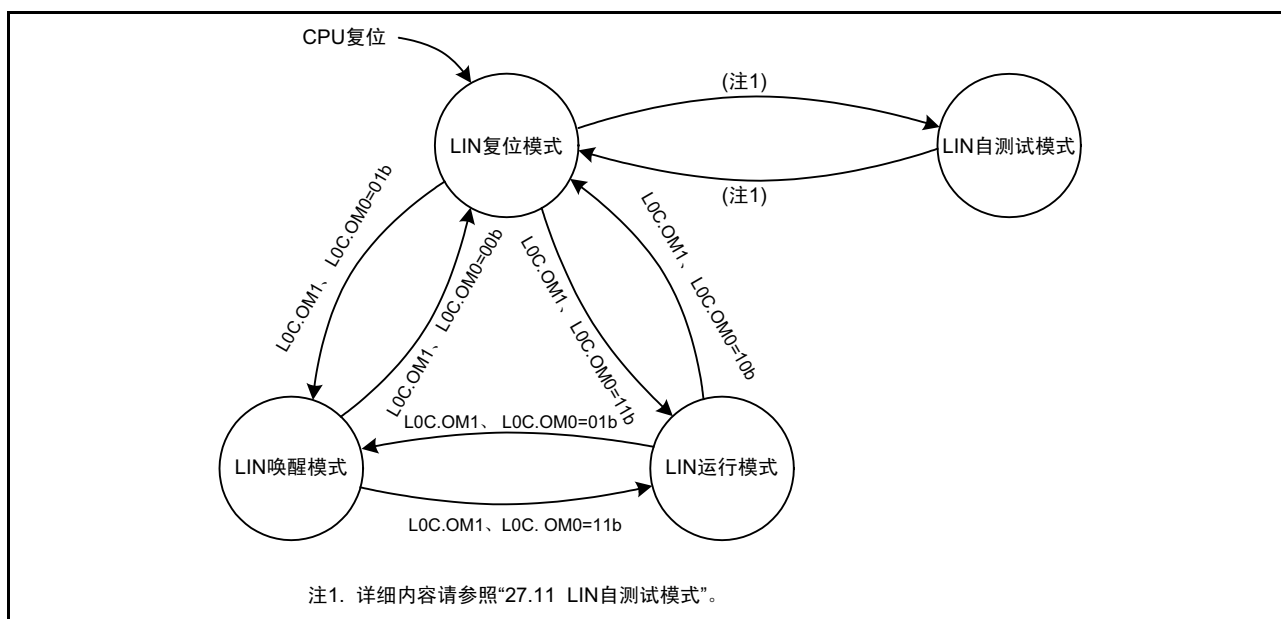


图 27.2 运行模式的转移

表 27.4 能在各运行模式中进行的运行

LIN 运行模式	LIN 唤醒模式	LIN 自测试模式
报文头的发送 响应的发送 响应的接收 错误检测	唤醒的发送 唤醒的接收 错误检测	自测试

能通过读 L0MST.OMM1 位和 L0MST.OMM0 位确认转移到 LIN 复位模式、LIN 运行模式或者 LIN 唤醒模式。

有关 LIN 自测试模式，请参照“27.12 LIN 自测试模式”。

### 27.3.1 LIN 复位模式

如果将 L0C.OM1 位和 L0C.OM0 位置“00b”或者“10b” (LIN 复位模式)，就转移到 LIN 复位模式。当 L0MST.OMM1 位和 L0MST.OMM0 位变为“00b”或者“10b” (LIN 复位模式)，就确认转移到 LIN 复位模式。在此模式中，LIN 通信功能全部停止，fLIN 也停止。

能从 LIN 复位模式转移到 LIN 运行模式、LIN 唤醒模式或者 LIN 自测试模式。

在转移到 LIN 复位模式后，以下寄存器被初始化为各自复位后的值，在 LIN 复位模式中保持初始值。

- L0TC 寄存器
- L0ST 寄存器
- L0EST 寄存器

在转移到 LIN 复位模式后，以下寄存器保持之前的值。

- LWBR 寄存器
- LBRP0 寄存器
- LBRP1 寄存器
- LSTC 寄存器 (仅 LSTC.LSTM 位)
- L0MD 寄存器
- L0BRK 寄存器
- L0SPC 寄存器
- L0WUP 寄存器
- L0IE 寄存器
- L0EDE 寄存器
- L0RFC 寄存器
- L0IDB 寄存器
- L0DBn 寄存器 (n=1~8)

### 27.3.2 LIN 运行模式

如果将 L0C.OM1 位和 L0C.OM0 位置“11b”，就为 LIN 运行模式，并且 L0MST.OMM1 位和 L0MST.OMM0 位变为“11b”。等到 L0MST 变为“11b”后，进行发送数据的设定。

### 27.3.3 LIN 唤醒模式

如果将 L0C.OM1 位和 L0C.OM0 位置“01b”，就为 LIN 唤醒模式，并且 L0MST.OMM1 位和 L0MST.OMM0 位变为“01b”。

### 27.3.4 LIN 自测试模式

通过写 LSTC 寄存器转移到 LIN 自测试模式。当 LSTC.LSTM 位变为“1”，就确认转移到 LIN 自测试模式。



## 27.4 运行概要

### 27.4.1 报文头的发送

发送 LIN 模块的报文头时的运行如图 27.3 所示，发送报文头时的处理如表 27.5 所示。

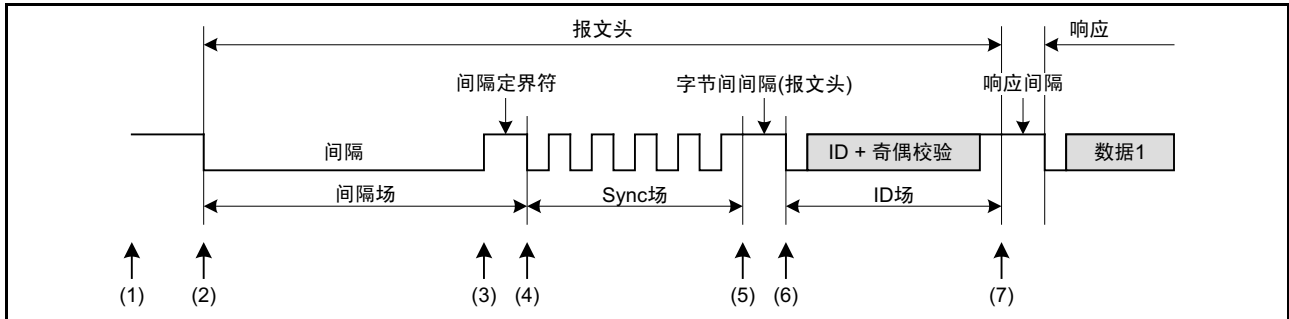


图 27.3 发送报文头时的运行

表 27.5 发送报文头时的处理

	软件处理	LIN 模块处理
(1)	<ul style="list-style-type: none"> <li>• 设定波特率 (请参照“27.5 波特率发生器”)。</li> <li>• 给 L0IE.FTCIE 位设定“1” (允许帧 / 唤醒发送结束中断)、L0IE.FRCIE 位设定“1” (允许帧 / 唤醒接收结束中断)、L0IE.ERRIE 位设定“1” (允许错误检测中断)。</li> <li>• 通过 L0C.OM1 位和 L0C.OM0 位更改 LIN 模块的运行模式。</li> <li>• 通过 L0BRK.BLT[3:0] 位设定间隔宽度 (13 ~ 28 Tbit)、通过 L0BRK.BDT[1:0] 位设定间隔定界符宽度 (1 ~ 4 Tbit)。</li> <li>• 通过 L0SPC.IBSH[2:0] 位设定字节间间隔 (报文头) / 响应间隔宽度 (0 ~ 7 Tbit)、通过 L0SPC.IBS[1:0] 位设定字节间间隔宽度 (0 ~ 3 Tbit)。</li> <li>• 给 L0IDB 寄存器设定 ID 值以及其奇偶校验值。</li> <li>• 通过 L0RFC.RFDL[3:0] 位设定数据长度、通过 L0RFC.RFT 位设定响应的发送 / 接收方向、通过 L0RFC.CSM 位设定校验和方式。</li> <li>• 设定发送数据</li> </ul>	等待通过软件开始帧 / 唤醒的发送 (空闲)。
(2)	将 L0TC.FTS 位置“1” (开始帧发送 / 唤醒发送或者接收)。	间隔的 Low 电平发送
(3)		间隔定界符的发送
(4)		Sync 场 (55h) 的发送
(5)		字节间间隔 (报文头) 的发送
(6)		ID 场的发送
(7)		报文头发送结束标志的设定、错误标志的设定或者响应间隔的发送

### 27.4.2 响应的发送

发送 LIN 模块的响应时的运行如图 27.4 所示，发送响应时的处理如表 27.6 所示。

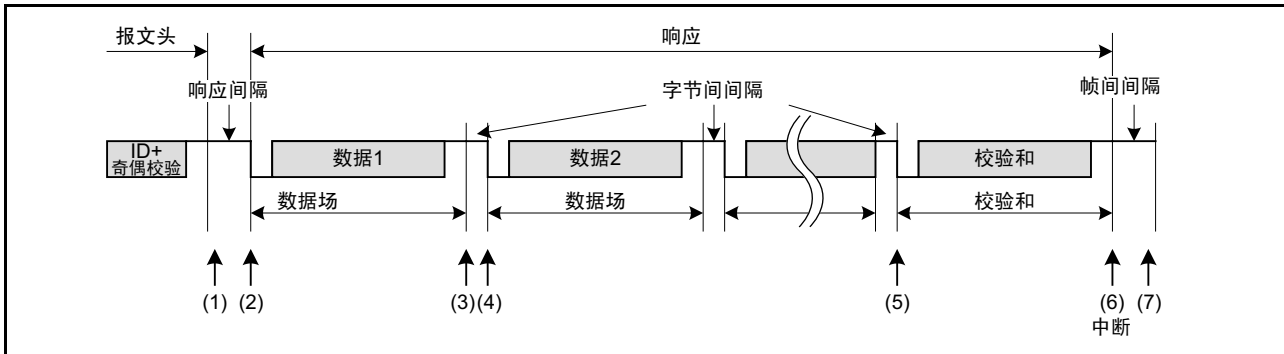


图 27.4 发送响应时的运行

表 27.6 发送响应时的处理

	软件处理	LIN 模块处理
(1)	(帧分隔模式的情况) L0TC.RTS 位 =1 (开始响应的发送) (非帧分隔模式的情况) 等待产生中断请求	(帧分隔模式的情况) 在等待开始响应的发送期间发送响应间隔 (非帧分隔模式的情况) 如果响应间隔发送结束就继续 (2)
(2)	等待产生中断请求	数据 1 的发送
(3)		字节间间隔的发送
(4)		数据 2 的发送 字节间间隔的发送 数据 3 的发送 字节间间隔的发送 (重复发送 L0RFC.RFDL[3:0] 位指定的数据长度。如果 L0EST.BER 标志 =1 (检测到位错误)，就发生中断。在发生错误时，不执行 (5) 校验和的发送。)
(5)		校验和的发送
(6)		<ul style="list-style-type: none"> <li>帧 / 唤醒发送结束标志的设定或者错误标志的设定</li> <li>将 L0TC.FTS 位置“0” (停止帧发送 / 唤醒发送或者接收)、L0TC.RTS 位置“0” (停止响应的发送)</li> </ul>
(7)	通信后的处理 L0ST 寄存器的检查、标志的清除	空闲

### 27.4.3 响应的接收

接收 LIN 模块的响应时的运行如图 27.5 所示，接收响应时的处理如表 27.7 所示。

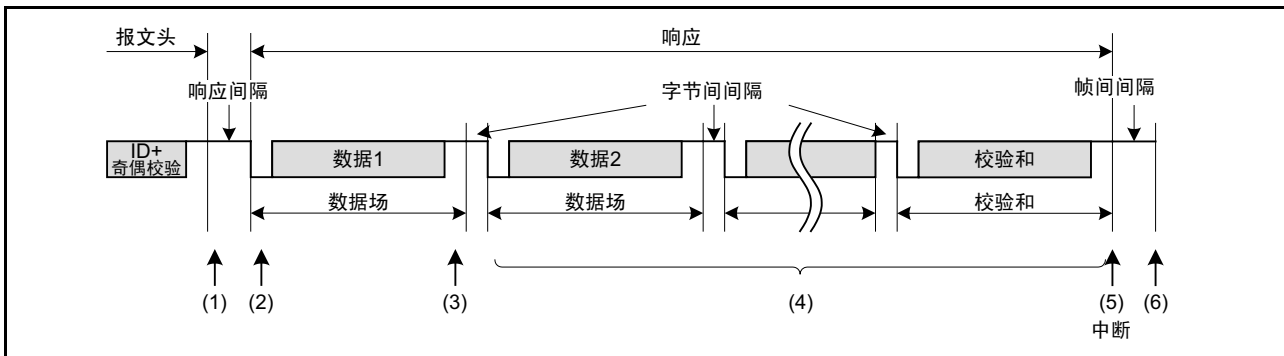


图 27.5 接收响应时的运行

表 27.7 接收响应时的处理

	软件处理	LIN 模块处理
(1)	等待产生中断请求 (无处理)	等待检测起始位
(2)	等待产生中断请求	通过检测起始位进行数据 1 的接收
(3)		数据 1 接收结束标志的设定
(4)		通过检测起始位进行数据 2 的接收 通过检测起始位进行数据 3 的接收 (重复接收 L0RFC.RFDL[3:0] 位指定的数据长度。如果 L0EST 寄存器的任意位为“1” (检测到某错误)，就发生中断。在发生错误时，不执行 (5) 校验和的判断。) : : 通过检测起始位进行校验和的接收
(5)		<ul style="list-style-type: none"> <li>• 校验和的判断</li> <li>• 帧 / 唤醒接收结束标志的设定或者错误标志的设定</li> <li>• 将 L0TC.FTS 位置“0” (停止帧发送 / 唤醒发送或者接收)</li> </ul>
(6)	通信后的处理 读接收数据 L0ST 寄存器的检查、标志的清除	<ul style="list-style-type: none"> <li>• 空闲</li> </ul>

## 27.5 波特率发生器

通过波特率发生器对 LIN 通信时钟源进行分频后的时钟为 LIN 系统时钟 (fLIN)。将 LIN 系统时钟 16 分频后的时钟为位速率。将此位速率的倒数称为位时间 (Tbit)。

设定 LBRP0 寄存器, 使 fa 为 307200Hz (=19200×16), 就能使 fa=19200×16、fb=9600×16、fc=2400×16, 因为通过位时序生成部进行 16 分频, 因此能生产 19200bps、9600bps、2400bps。另外, 通过 LBRP1 寄存器生成 10417bps。

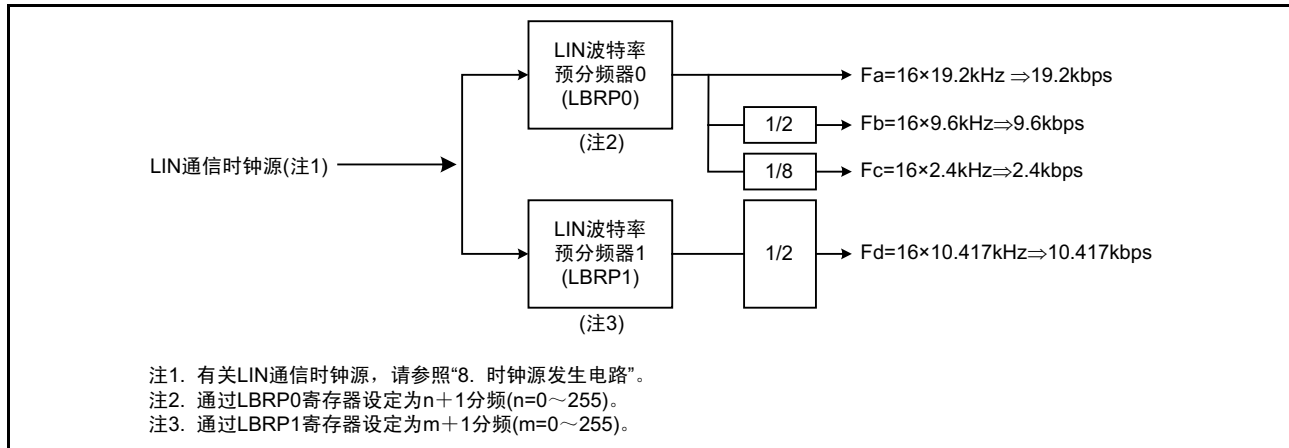


图 27.6 波特率的生成框图

各外围功能时钟频率的波特率 (19200bps、9600bps、2400bps、10417bps) 生成例子及其误差如表 27.8、表 27.9 所示。

表 27.8 波特率的生成例子 (19200bps、9600bps、2400bps)

LIN 通信 时钟源	波特率生成器 0 N + 1 分频	生成波特率						误差
		选择 fa		选择 fb		选择 fc		
50MHz	162	19290.12	(19200bps)	9645.06	(9600bps)	2411.27	(2400bps)	+0.47%
40MHz	130	19230.77	(19200bps)	9615.38	(9600bps)	2403.85	(2400bps)	+0.16%
25MHz	81	19290.12	(19200bps)	9645.06	(9600bps)	2411.27	(2400bps)	+0.47%
24MHz	78	19230.77	(19200bps)	9615.38	(9600bps)	2403.85	(2400bps)	+0.16%
20MHz	65	19230.77	(19200bps)	9615.38	(9600bps)	2403.85	(2400bps)	+0.16%
16MHz	52	19230.77	(19200bps)	9615.38	(9600bps)	2403.85	(2400bps)	+0.16%
12MHz	39	19230.77	(19200bps)	9615.38	(9600bps)	2403.85	(2400bps)	+0.16%
10MHz	65	9615.38	(9600bps)	—		—		+0.16%
8MHz	26	19230.77	(19200bps)	9615.38	(9600bps)	2403.85	(2400bps)	+0.16%
6MHz	39	9615.38	(9600bps)	—		—		+0.16%
	156	2403.85	(2400bps)	—		—		+0.16%
5MHz	130	2403.85	(2400bps)	—		—		+0.16%
4MHz	13	19230.77	(19200bps)	9615.38	(9600bps)	2403.85	(2400bps)	+0.16%
2MHz	13	9615.38	(9600bps)	—		—		+0.16%
	52	2403.85	(2400bps)	—		—		+0.16%

—: 不能生成对应的波特率

表 27.9 波特率的生成例子 (10417bps)

LIN 通信时钟源	波特率生成器 1 M + 1 分频	生成波特率 选择 fd	误差
50MHz			
40MHz	120	10416.67	-0.003%
25MHz	75	10416.67	-0.003%
24MHz	72	10416.67	-0.003%
20MHz	60	10416.67	-0.003%
16MHz	48	10416.67	-0.003%
12MHz	36	10416.67	-0.003%
10MHz	30	10416.67	-0.003%
8MHz	24	10416.67	-0.003%
6MHz	18	10416.67	-0.003%
5MHz	15	10416.67	-0.003%
4MHz	12	10416.67	-0.003%
2MHz	6	10416.67	-0.003%

## 27.6 数据的发送 / 接收

### 27.6.1 数据的发送

通过 1 Tbit 一位一位进行数据的发送。

发送的数据经由 LIN 收发器返回到接收数据的输入引脚。按位进行该接收数据和发送数据的比较，其结果保存到 LOEST.BER 标志（参照 27.10 错误状态）。以 1 Tbit=16fLIN 生成接收数据的采样点，并且成为第 13 个时钟（81.25% 位置）。

数据的发送时序如图 27.7 所示。

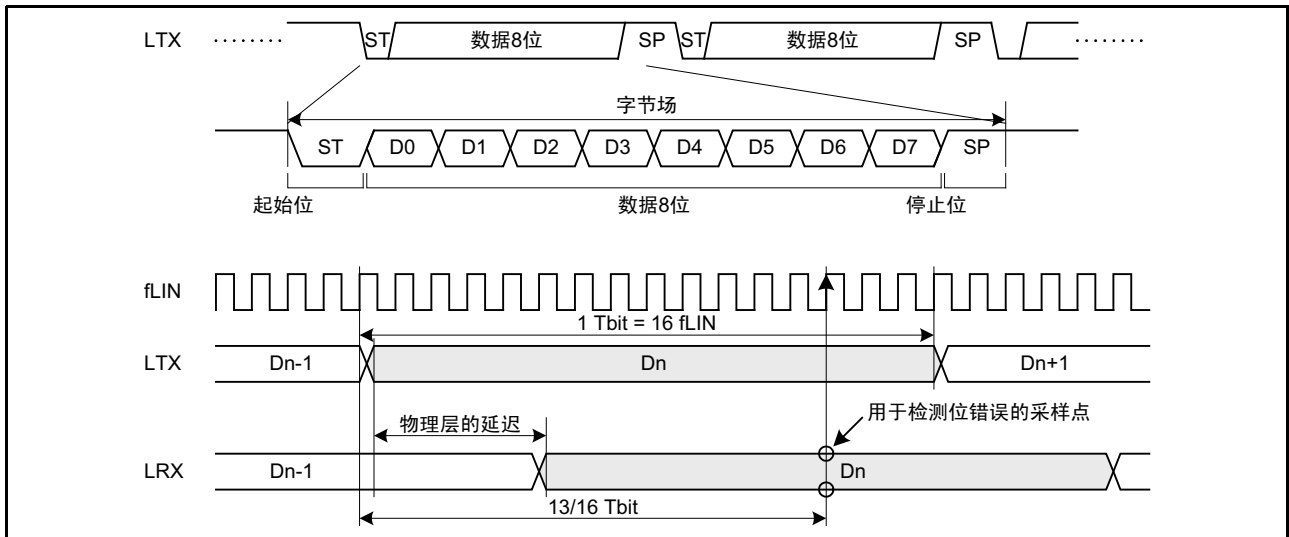


图 27.7 数据的发送时序

### 27.6.2 数据的接收

使用使 LRX 引脚输入与 fLIN 同步的同步 LRX（内部信号）进行数据的接收。

在该同步 LRX 信号的起始位下降沿与字节场同步。如果检测到下降沿，就在 0.5 Tbit 后重新进行采样，并且在同步 LRX 信号为 Low 电平时视为起始位。如果在复位解除后 LRX 信号一直保持 Low 电平，或者在重新进行采样时检测到 High 电平，就不视为起始位。

在检测到起始位后，以 1 Tbit 进行位采样。

数据的接收时序如图 27.8 所示。

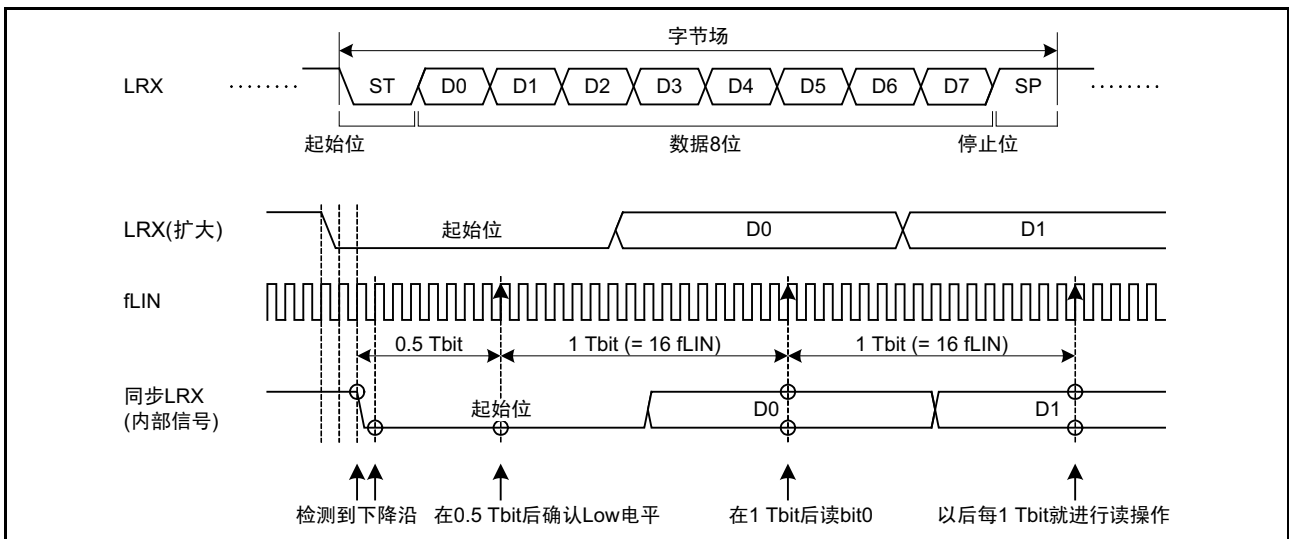


图 27.8 数据的接收时序

## 27.7 发送 / 接收数据的缓冲处理

对发送 / 接收 LIN 模块的连续数据时的缓冲处理进行说明。

### 27.7.1 LIN 帧的发送

保存在 L0DB1 ~ L0DB8 寄存器的内容在以 8 字节发送的情况下按顺序发送到 LIN 帧的数据 1 ~ 8 区域。保存在 L0DB1 ~ L0DB4 寄存器的内容在以 4 字节发送的情况下按顺序发送到 LIN 帧的数据 1 ~ 4 区域。保存在 L0DB5 ~ L0DB8 寄存器的内容不被发送。另外，L0CBR 寄存器用于保存已发送的校验和数据。

LIN 发送处理和缓冲如图 27.9 所示。

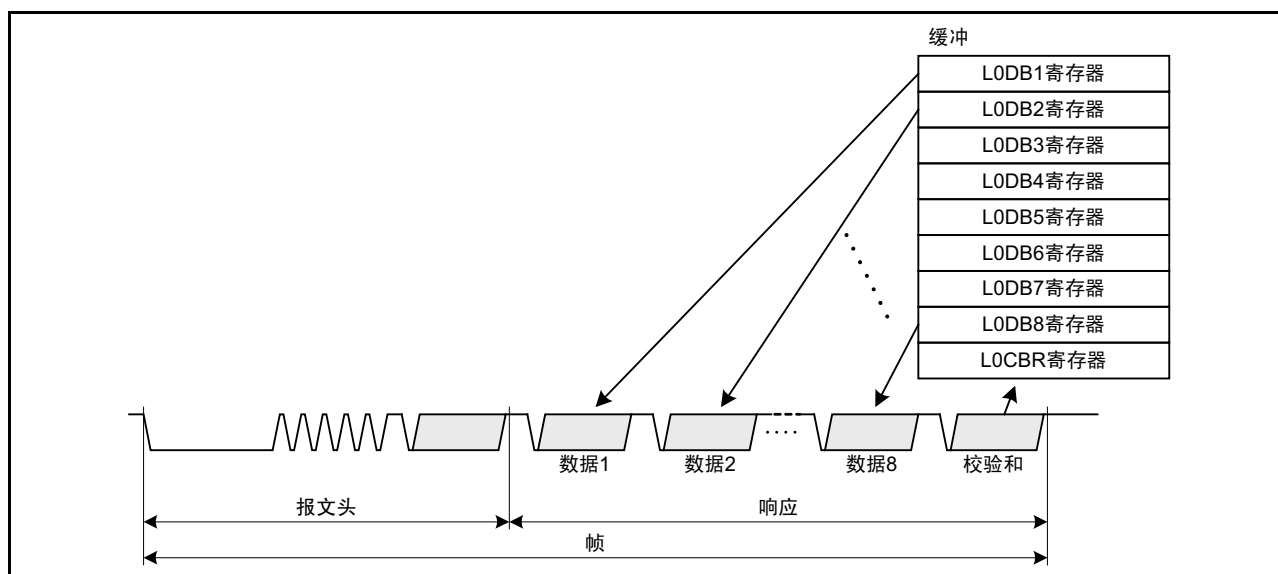


图 27.9 LIN 发送处理和缓冲

#### 27.7.1.1 帧分隔模式

通过将 L0RFC.FSM 位置“1”，变为帧分隔模式。

这是通过不同的发送开始请求发送报文头和响应的模式。

如果报文头发送结束，L0ST.HTRC 标志就变为“1”（报文头发送结束）。

## 27.7.2 LIN 帧的接收

每当接收停止位时，LIN 帧的数据 1 ~ 8 区域的内容在以 8 字节接收的情况下分别保存到 L0DB1 ~ L0DB8 寄存器；LIN 帧的数据 1 ~ 4 区域的内容在以 4 字节接收的情况下分别保存到 L0DB1 ~ L0DB4 寄存器；L0DB5 ~ L0DB8 寄存器什么都不保存。另外，L0CBR 寄存器保存已接收到的校验和数据。

LIN 接收处理和缓冲如图 27.10 所示。

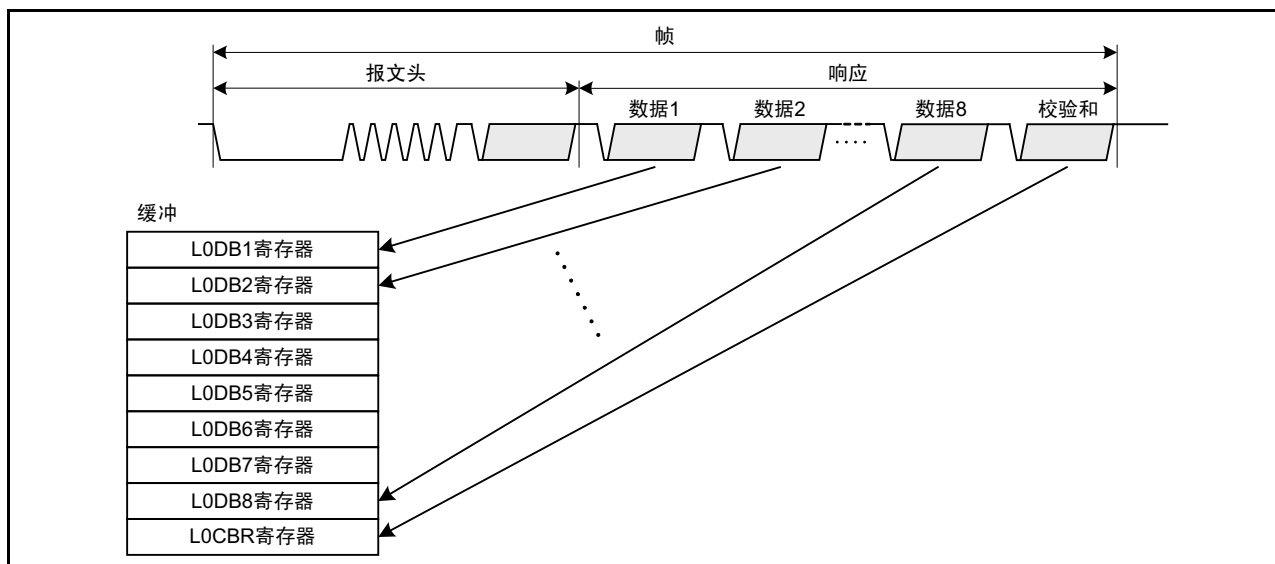


图 27.10 LIN 接收处理和缓冲

### 27.7.2.1 数据 1 的接收

如果第 1 字节的数据接收结束，LOST.DIRC 标志就变为“1”（数据 1 接收结束）。



## 27.8 唤醒的发送 / 接收

在 LIN 唤醒模式中使用唤醒的发送 / 接收。

### 27.8.1 唤醒的发送运行

在 LIN 唤醒模式中，如果将 LORFC.RFT 位置“1”（发送）、L0TC.FTS 位置“1”（开始帧发送 / 唤醒发送或者接收），就从输出引脚输出唤醒信号。通过 LOWUP.WUTL[3:0] 位设定唤醒信号的 Low 电平宽度。

在无位错误的情况下输出唤醒的 Low 电平时，LOST.FTC 标志变为“1”（帧或者唤醒发送结束），并且在 LOIE.FTCIE 位为“1”（允许帧 / 唤醒发送结束中断）时产生中断请求。

如果检测到位错误，就中止唤醒的发送，将 LOEST.BER 标志置“1”（位错误检测）。

唤醒发送的时序如图 27.11 所示。

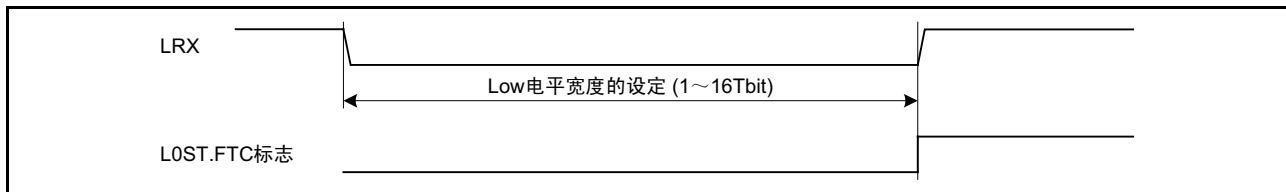


图 27.11 唤醒发送的时序

### 27.8.2 唤醒的接收运行

使用输入信号的 Low 电平宽度计数功能检测唤醒。

输入信号的 Low 电平宽度计数功能是在与接收数据相同的采样点测量 LRX 引脚输入信号的 Low 电平宽度的功能。测量的 Low 电平宽度因 LWBR.LWBR0 位的设定而不同。在使用 LIN Specification Package Revision 1.3 时，必须将 LWBR.LWBR0 位置“0”。由此能以至少 fLIN 的 2.5 Tbit 测量输入信号的 Low 电平宽度。在使用 LIN Specification Package Revision 2.0、2.1 时，必须将 LWBR.LWBR0 位置“1”。由此能至少以 130μs 测量输入信号的 Low 电平宽度。

在使用此功能时，必须在 LIN 唤醒模式中将 LORFC.RFT 位置“0”（接收）、L0TC.FTS 位置“1”（开始帧发送 / 唤醒发送或者接收）。

如果达到测量的 Low 电平宽度，LOST.FRC 标志就变为“1”（帧或者唤醒接收结束），并且在 LOIE.FRCIE 位为“1”（允许帧 / 唤醒接收结束中断）时产生中断请求。

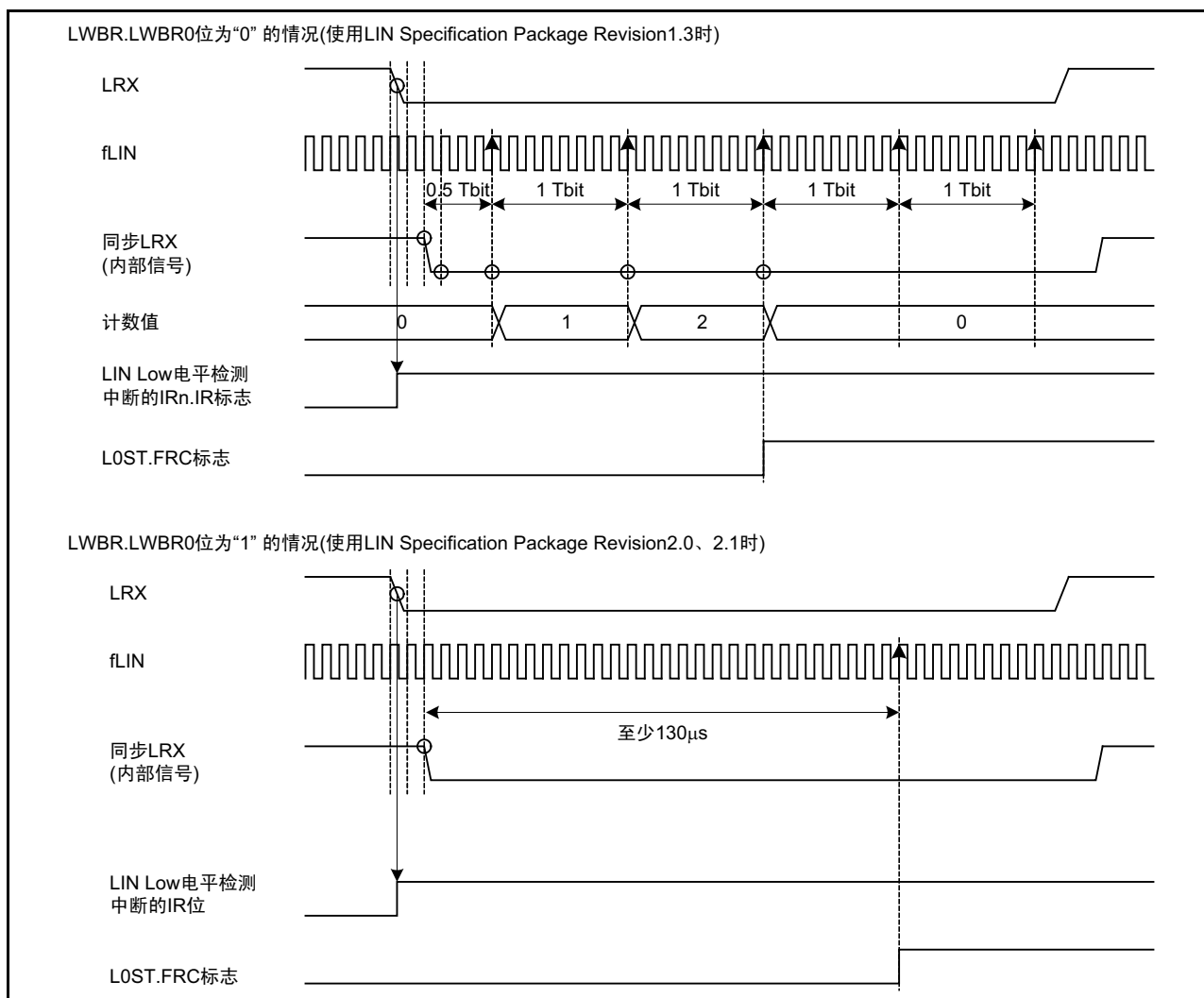


图 27.12 输入信号的 Low 电平计数功能

在唤醒的发送过程中，不能运行输入信号的 Low 电平计数功能。

### 27.8.3 唤醒冲突

如果主控节点和从属节点同时发送唤醒信号，就在 LIN 总线上发生冲突。在 LIN 模块中不检测唤醒信号的冲突。

## 27.9 状态

LIN 模块检测 7 种状态。

帧 / 唤醒发送结束、帧 / 唤醒接收结束和错误检测这 3 种状态能产生中断请求。

状态的种类如表 27.10 所示。

表 27.10 状态的种类

状态	状态复位条件	状态清除条件	能检测状态的运行模式	对应位
LIN 模式	将 L0C.OM1 位设定为 LIN 运行模式后, LIN 模块变为 LIN 运行模式时	将 L0C.OM1 位设定为 LIN 唤醒模式后, LIN 模块变为 LIN 唤醒模式时	<ul style="list-style-type: none"> <li>• LIN 运行模式</li> <li>• LIN 唤醒模式</li> </ul>	L0MST.OMM1 位
复位	将 L0C.OM0 位设定为非 LIN 运行模式后, LIN 模块解除 LIN 复位模式时	将 L0C.OM0 位设定为 LIN 复位模式后, LIN 模块变为 LIN 复位模式时	全部模式	L0MST.OMM0 位
帧 / 唤醒发送结束	正常结束响应场或者唤醒信号的发送时	<ul style="list-style-type: none"> <li>• 开始下一次通信时</li> <li>• 通过软件清除</li> <li>• 转移到 LIN 复位模式时</li> </ul>	<ul style="list-style-type: none"> <li>• LIN 运行模式</li> <li>• LIN 唤醒模式</li> </ul>	L0ST.FTC 标志
帧 / 唤醒接收结束	正常结束响应场或者唤醒信号的接收时	<ul style="list-style-type: none"> <li>• 开始下一次通信时</li> <li>• 通过软件清除</li> <li>• 转移到 LIN 复位模式时</li> </ul>	<ul style="list-style-type: none"> <li>• LIN 运行模式</li> <li>• LIN 唤醒模式</li> </ul>	L0ST.FRC 标志
错误检测	L0EST.CSER、L0EST.FER、L0EST.FTER、L0EST.PBER、L0ESR.BER 标志中的任意一个标志变为“1” (错位检测时)	<ul style="list-style-type: none"> <li>• 开始下一次通信时</li> <li>• 通过软件清除 (注 1)</li> <li>• 转移到 LIN 复位模式时</li> </ul>	<ul style="list-style-type: none"> <li>• LIN 运行模式</li> <li>• LIN 唤醒模式</li> </ul>	L0ST.ERR 标志
数据 1 接收结束	L0RFC.RFT 位为“0” (接收) 并且结束响应帧的第 1 字节接收时 (注 2)	<ul style="list-style-type: none"> <li>• 开始下一次通信时</li> <li>• 通过软件清除</li> <li>• 转移到 LIN 复位模式时</li> </ul>	LIN 运行模式	L0ST.D1RC 标志
报文头发送结束	正常结束报文头场的发送时	<ul style="list-style-type: none"> <li>• 开始下一次通信时</li> <li>• 通过软件清除</li> <li>• 转移到 LIN 复位模式时</li> </ul>	LINI 运行模式	L0ST.HTRC 标志

注 1. 在 LIN 运行模式中, 通过给 L0EST.CSER、L0EST.FER、L0EST.FTER、L0EST.PBER、L0EST.BER 标志写“0”, L0ST.ERR 标志变为“0”。

注 2. 在 L0RFC.RFDL[3:0] 位为“0000b” (0 字节+校验和) 时检测不到。

## 27.10 错误状态

### 27.10.1 错误状态的种类

LIN 模块检测 5 种错误状态。能通过 LEST 寄存器的各位确认这些错误状态。  
错误状态的种类如表 27.11 所示。

表 27.11 错误状态的种类

状态	状态检测条件 (通过软件置“0”)	能检测错误的 运行模式	通信处理	选择允许 / 禁止检测	对应位
位错误	当已发送的数据和通过接收引脚监视的 LIN 总线上的数据不相同 (注 1)	<ul style="list-style-type: none"> <li>• LIN 运行模式</li> <li>• LIN 唤醒模式</li> </ul>	中断	○	L0EST.BER 标志
物理总线错误	<ul style="list-style-type: none"> <li>• 在发送间隔场时 LIN 总线检测到 High 电平</li> <li>• 在发送间隔定界符时 LIN 总线检测到 Low 电平</li> <li>• 在发送唤醒时 LIN 总线检测到 High 电平</li> </ul>	<ul style="list-style-type: none"> <li>• LIN 运行模式</li> <li>• LIN 唤醒模式</li> </ul>	中断	○	L0EST.PBER 标志
帧超时错误	没有在一定的时间内结束帧的发送 / 接收时 (注 2)	LIN 运行模式	中断	○	L0EST.FTER 标志
帧错误	在响应帧的接收处理过程中, 各数据字节的停止位为 Low 电平时	LIN 运行模式	中断	○	L0EST.FER 标志
校验和错误	在响应帧的接收处理过程中, 校验和的判断结果发生错误时	LIN 运行模式	—	x	L0EST.CSER 标志

注 1. 在检测到位错误时, 就在发送停止位后发生中断。在字节间间隔等非数据区域检测到位错误时, 就在该区域结束后立即中止发送。在唤醒发送过程中检测到位错误时, 就在发送错误位后立即中止唤醒发送。

注 2. 超时时间取决于响应场的数据长度 (L0RFC.RFDL[3:0] 位) 和校验和的选择 (L0RFC.CSM 位), 能通过以下表达式进行计算。

选择“classic”时 (L0RFC.CSM 位为“0”的情况)  
 超时时间 =  $49 + (\text{数据字节数} + 1) \times 14 [\text{Tbit}]$   
 选择“enhance”时 (L0RFC.CSM 位为“1”的情况)  
 超时时间 =  $48 + (\text{数据字节数} + 1) \times 14 [\text{Tbit}]$

上述超时时间在选择“classic”时为超出 LIN Specification Package Revision 1.3 的 TFRAME\_MAX 的时间, 在选择“enhance”时为超出 LIN Specification Package Revision 2.0、2.1 的 TFRAME\_MAX 的时间。

### 27.10.2 LIN 错误检测的对象时间区域

为了检测错误，LIN 模块监视的时间区域如图 27.13 所示。

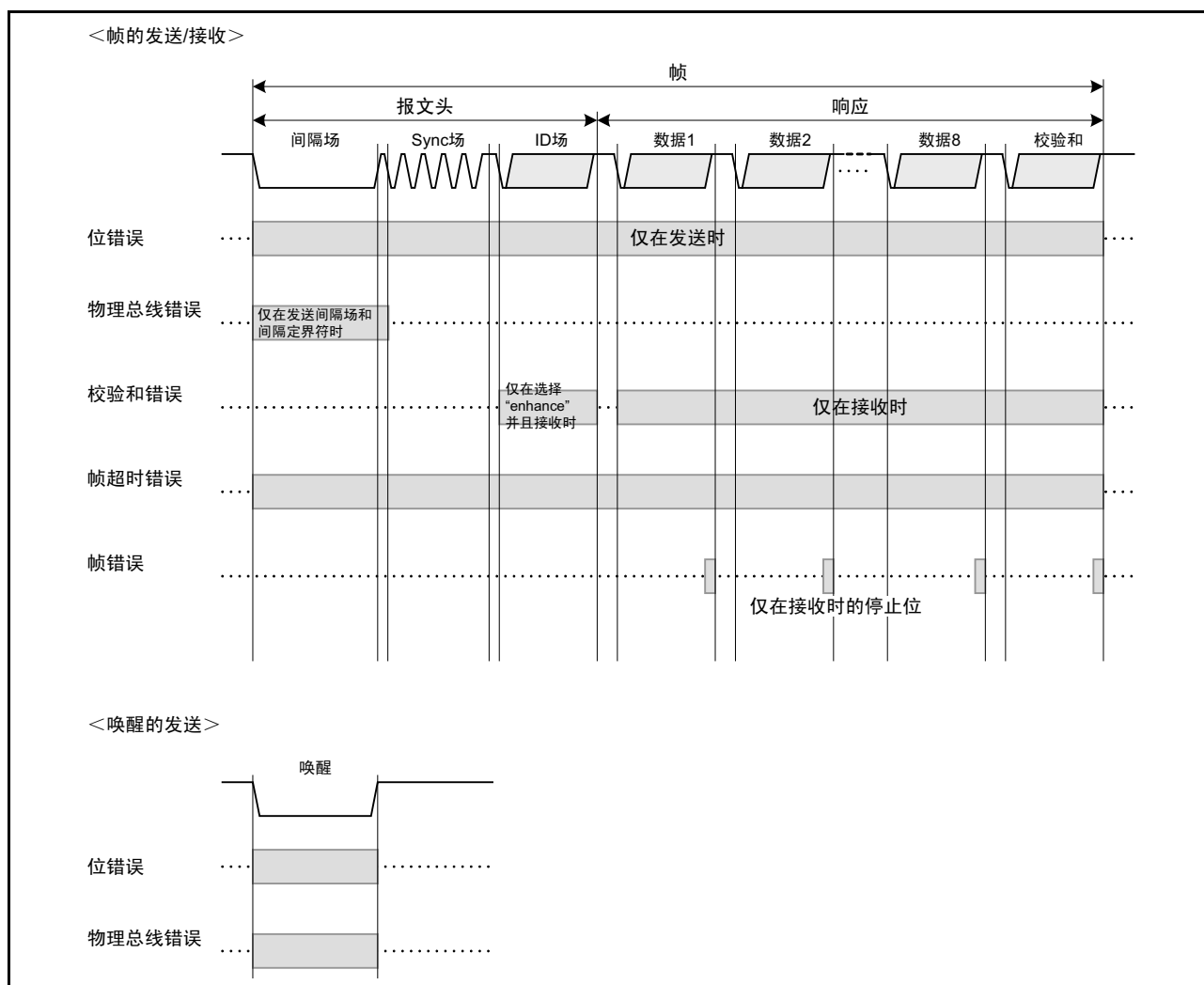


图 27.13 LIN 错误检测的对象时间区域

## 27.11 中断

LIN 模块生成的中断请求有 LIN 中断。

中断源有 3 种，分别是帧 / 唤醒发送结束、帧 / 唤醒接收结束、错误检测。

通过取逻辑或，将由帧 / 唤醒发送结束、帧 / 唤醒接收结束、错误检测这 3 种状态产生的中断请求汇总为 1 个中断请求“LIN 中断”。

在 LOIE 寄存器的对应位为“1”（允许中断）时，如果 LOST 寄存器的对应标志变为“1”就输出各中断请求。但是，因为忽视 LOST 寄存器的对应标志为“1”时的中断请求，所以为了能重新中断，必须将对应标志置“0”。

LIN 中断的框图如图 27.14 所示。

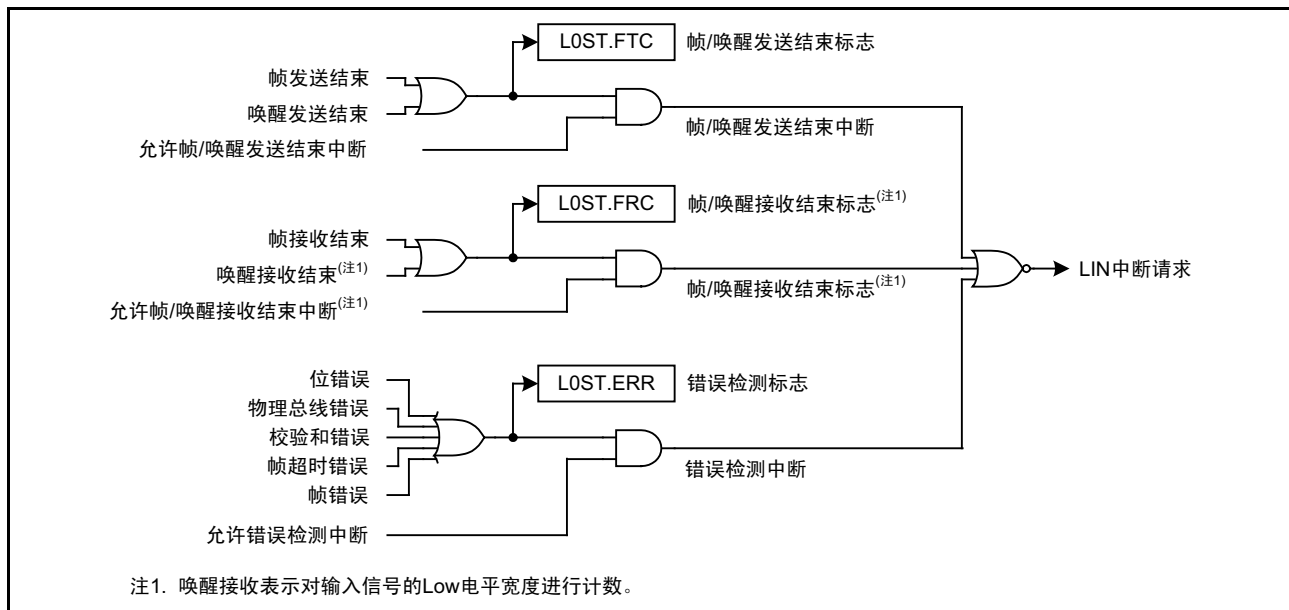


图 27.14 LIN 中断的框图

## 27.12 LIN 自测试模式

LIN 模块具有 LIN 自测试模式。一旦 LIN 模块变为 LIN 自测试模式，就从 LIN 总线被切断，并且内部 LTX 环回到内部 LRX。

LIN 自测试模式在以下的状态下运行。

- LIN 自测试模式
- 不支持唤醒功能
- 不支持帧分隔模式
- 波特率生成器为最快设定 (LBRP0 寄存器为“00h”、LBRP1 寄存器为“00h”、L0MD.LCKS[1:0] 位为“00b”)

不能转移到 LIN 唤醒模式。

在转移到 LIN 自测试模式前，必须将 LORFC.FSM 位置“0” (非帧分隔模式)。

在 LIN 自测试模式中波特率的设定为自动设定。另外，从 LIN 自测试模式转移到 LIN 复位模式时，通过自动设定返回到 LIN 自测试模式之前的设定。

在转移到 LIN 自测试模式或者从 LIN 自测试模式转移到 LIN 复位模式时，其他设定保持不变并且有效。

LOST 寄存器和 LOEST 寄存器保持运行，但是，在环回时不能检测到 LOEST.FER 标志、LOEST.PBER 标志和 LOEST.BER 标志，并且不变为“1”。

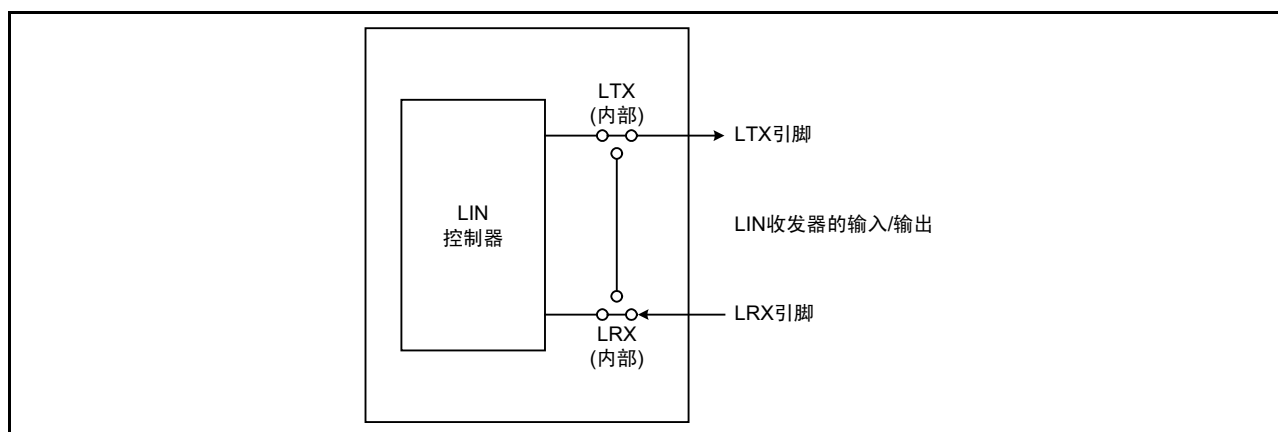


图 27.15 LIN 运行模式的连接

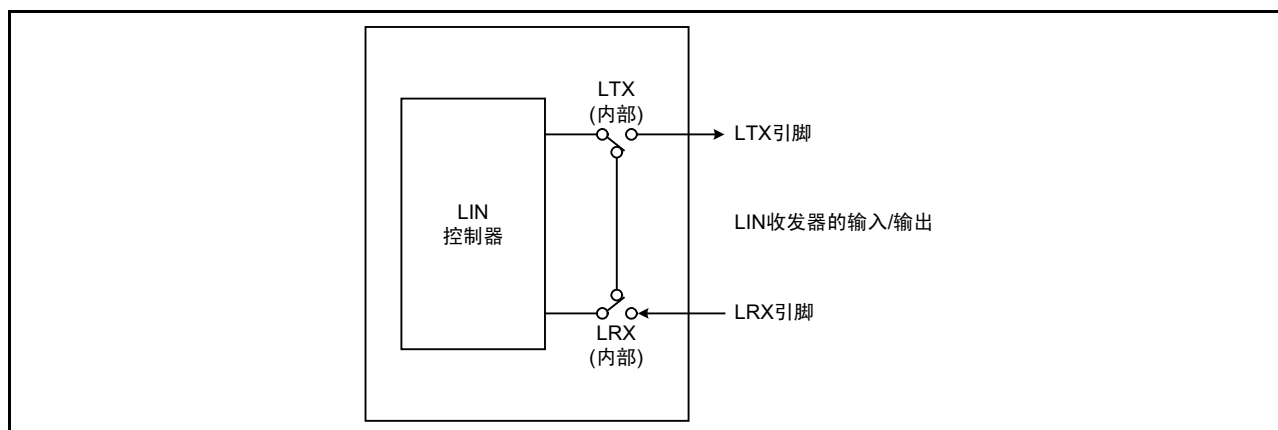


图 27.16 LIN 自测试模式的连接

### 27.12.1 向 LIN 自测试模式的转移

在向 LIN 自测试模式转移时，必须使用特定的键顺序。该键顺序需要按照以下的顺序连续写 3 次 LIN 自测试控制寄存器。

- 向 LIN 复位模式转移
- 第 1 次写操作：LSTC 寄存器=1010 0111 (A7h)
- 第 2 次写操作：LSTC 寄存器=0101 1000 (58h)
- 第 3 次写操作：LSTC 寄存器=0000 0001 (01h)

当第 1 次的键被写 2 次时，顺序就被中断，因此必须重新开始。  
由于写其他 LIN 相关寄存器而使该顺序中断时，也必须重新开始。

LIN 自测试模式不支持帧分隔模式，能进行以下 2 种测试。

- LIN 自测试模式 (发送)：报文头的发送和响应的发送
- LIN 自测试模式 (接收)：报文头的发送和响应的接收

### 27.12.2 LIN 自测试模式中的发送

按照以下的步骤执行 LIN 自测试。

- 给 LOC.OM1 和 LOC.OM0 位写“11b”，确认 L0MST.OMM1 位和 L0MST.OMM0 位变为“11b”。
- 将 LORFC.RFT 位置“1” (发送)。
- 设定要发送的帧结构。
- 将 LOTC.FTS 位置“1” (开始帧发送/唤醒发送或者接收)
- 执行 LIN 自测试模式中的发送，并且中断的发生、状态和错误状态的更新也一起执行。LIN 自动运算校验和。
- 发送结束时，已环回的帧数据的取反值被保存到 L0IDB 寄存器、L0CBR 寄存器和 L0DBn 寄存器 (因为对已发送的值和已环回的值进行比较，所以作为取反值被保存)。(n=1~8)
- 由于发生错误而未完成发送时，对应的错误标志就被设定。

### 27.12.3 LIN 自测试中的接收

按照以下的步骤执行 LIN 自测试。

- 向 LIN 自测试模式转移。
- 将 LORFC.RFT 位置“0” (接收)。
- 设定要接收的帧结构。因为不自动运算校验和，所以必须保存运算值。此时，如果给校验和设定错误的运算结果，就能测试校验和错误。
- 将 LOTC.FTS 位置“1” (开始帧发送/唤醒发送或者接收)
- 执行 LIN 自测试模式中的接收，并且中断的发生、状态和错误状态的更新也一起执行。
- 接收结束时，已环回的帧数据的取反值被保存到 L0IDB 寄存器、L0CBR 寄存器和 L0DBn 寄存器 (因为对已设定的值和已环回并且接收的值进行比较，所以作为取反值被保存)。(n=1~8)
- 由于发生错误而未完成接收时，对应的错误标志就被设定。

### 27.12.4 LIN 自测试模式的结束

按照以下的步骤结束 LIN 自测试模式。

- 向 LIN 复位模式转移  
(当 L0MST.OMM1 位和 L0MST.OMM0 位不为“11b”时，给 LOC.OM1 位和 LOC.OM0 位写“11b”，在确认 L0MST.OMM1 位和 L0MST.OMM0 位变为“11b”后，转移到 LIN 复位模式。)



## 28. 12 位 A/D 转换器 (S12ADA)

### 28.1 概要

RX62T 群内置 2 个单元的逐次逼近方式的 12 位 A/D 转换器，最多能选择 4 个通道的模拟输入。

A/D 转换器的运行模式有单次模式（对 1 个通道的模拟输入只进行 1 次转换）和扫描模式（按顺序对最多 4 个通道的模拟输入进行连续转换）。

A/D 转换器的规格、功能概要和框图分别如表 28.1、表 28.2 和图 28.1 所示。

表 28.1 A/D 转换器的规格

项目	内容
单元数	2 个单元 (S12AD0、S12AD1)
输入通道	8 个通道 (4 个通道 × 2 个单元)
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 $\mu$ s (当 A/D 转换时钟 ADCLK=50MHz、AVCC0=4.0 ~ 5.5V 时) 每个通道 2.0 $\mu$ s (当 A/D 转换时钟 ADCLK=25MHz、AVCC0=3.0 ~ 3.6V 时)
A/D 转换时钟	4 种: PCLK、PCLK/2、PCLK/4、PCLK/8
数据寄存器	10 个 将 A/D 转换结果保存到 12 位 A/D 数据寄存器。 AN000 和 AN100 输入有 2 个 A/D 数据寄存器。转换结果的保存地址通过触发类别进行转换。
运行模式	<ul style="list-style-type: none"> <li>• 单次模式: 对 1 个通道的模拟输入只进行 1 次转换。</li> <li>• 扫描模式 1 个周期扫描模式: 对最多 4 个通道的模拟输入只进行 1 次转换。 连续扫描模式: 对最多 4 个通道的模拟输入进行连续转换。 2 个通道的扫描模式: 能将单元内的通道分为 2 组, 并且能设定 2 种开始源。</li> </ul>
A/D 转换开始条件	<ul style="list-style-type: none"> <li>• 软件触发</li> <li>• 多功能定时器脉冲单元 3 (MTU3) 或者通用 PWM 定时器 (GPT) 的触发</li> <li>• 外部触发 能通过 ADTRG0# 引脚开始 S12AD0 的 A/D 转换, 通过 ADTRG1# 引脚开始 S12AD1 的 A/D 转换。</li> </ul>
功能	<ul style="list-style-type: none"> <li>• 采样 &amp; 保持功能 (3 个通道 / 1 个单元) S12AD0 的通道 0 ~ 2 (AN000 ~ AN002) 和 S12AD1 的通道 0 ~ 2 (AN100 ~ AN102) 内置专用的独立采样保持电路。由此能在每个单元对多个通道 (最多 3 个通道) 进行同时采样。</li> <li>• A/D 转换器的自诊断功能</li> <li>• 通过可编程增益放大器放大输入信号的功能 (3 个通道 / 1 个单元)</li> <li>• 窗口比较器功能 (3 个通道 / 1 个单元)</li> </ul>
中断源	<ul style="list-style-type: none"> <li>• 各单元在 A/D 转换结束时产生中断请求 (S12ADI)。</li> <li>• 能通过 S12ADI 中断来启动数据传送控制器 (DTC)。</li> <li>• 通过比较器检测产生中断请求 (CMP1) (也能用作 POE 源)。</li> </ul>
低功耗功能	<ul style="list-style-type: none"> <li>• 各单元能设定为模块停止状态。</li> </ul>

表 28.2 A/D 转换器的功能概要 (1/2)

项目			单元 0 (S12AD0)	单元 1 (S12AD1)
模拟输入通道			AN000 ~ AN003	AN100 ~ AN103
A/D 转换 开始条件	软件	软件触发	能	能
	外部触发	触发输入引脚	ADTRG0#	ADTRG1#
	MTU3 的触发	MTU0.TGRA 的比较匹配 / 输入捕捉	TRGA0N	TRGA0N
		MTU1.TGRA 的比较匹配 / 输入捕捉	TRGA1N	TRGA1N
		MTU2.TGRA 的比较匹配 / 输入捕捉	TRGA2N	TRGA2N
		MTU3.TGRA 的比较匹配 / 输入捕捉	TRGA3N	TRGA3N
		MTU4.TGRA 的比较匹配 / 输入捕捉或者互补 PWM 模式中 MTU4.TCNT 的下溢 (波谷)	TRGA4N	TRGA4N
		MTU6.TGRA 的比较匹配 / 输入捕捉	TRGA6N	TRGA6N
		MTU7.TGRA 的比较匹配 / 输入捕捉或者互补 PWM 模式中 MTU7.TCNT 的下溢 (波谷)	TRGA7N	TRGA7N
		MTU0.TGRE 的比较匹配	TRG0N	TRG0N
		MTU4.TADCORA 和 MTU4.TCNT 的比较匹配	TRG4AN	TRG4AN
		MTU4.TADCORB 和 MTU4.TCNT 的比较匹配	TRG4BN	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	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 (使用中减少功能 2 时)	TRG4ABN	TRG4ABN
		MTU7.TADCORA 和 MTU7.TCNT 的比较匹配	TRG7AN	TRG7AN
		MTU7.TADCORB 和 MTU7.TCNT 的比较匹配	TRG7BN	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	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 (使用中减少功能 2 时)	TRG7ABN	TRG7ABN		

表 28.2 A/D 转换器的功能概要 (2/2)

项目		单元 0	单元 1	
A/D 转换 开始条件	GPT 的触发	GPT0.GTADTRA 的比较匹配	GTADTRA0N	GTADTRA0N
		GPT0.GTADTRB 的比较匹配	GTADTRB0N	GTADTRB0N
		GPT1.GTADTRA 的比较匹配	GTADTRA1N	GTADTRA1N
		GPT1.GTADTRB 的比较匹配	GTADTRB1N	GTADTRB1N
		GPT2.GTADTRA 的比较匹配	GTADTRA2N	GTADTRA2N
		GPT2.GTADTRB 的比较匹配	GTADTRB2N	GTADTRB2N
		GPT3.GTADTRA 的比较匹配	GTADTRA3N	GTADTRA3N
		GPT3.GTADTRB 的比较匹配	GTADTRB3N	GTADTRB3N
		GPT0.GTADTRA 的比较匹配或者 GPT0.GTADTRB 的比较匹配	GTADTRA0N 或者 GTADTRB0N	GTADTRA0N 或者 GTADTRB0N
		GPT1.GTADTRA 的比较匹配或者 GPT1.GTADTRB 的比较匹配	GTADTRA1N 或者 GTADTRB1N	GTADTRA1N 或者 GTADTRB1N
		GPT2.GTADTRA 的比较匹配或者 GPT2.GTADTRB 的比较匹配	GTADTRA2N 或者 GTADTRB2N	GTADTRA2N 或者 GTADTRB2N
		GPT3.GTADTRA 的比较匹配或者 GPT3.GTADTRB 的比较匹配	GTADTRA3N 或者 GTADTRB3N	GTADTRA3N 或者 GTADTRB3N
通道专用的独立 采样&保持功能	对象通道	AN000 ~ AN002	AN100 ~ AN102	
可编程增益 放大器	对象通道	AN000 ~ AN002	AN100 ~ AN102	
	增益设定	2.0 倍、2.5 倍、3.077 倍、3.636 倍、4.0 倍、 4.444 倍、5.0 倍、5.714 倍、6.667 倍、10.0 倍、 13.333 倍 (共 11 级)		
窗口比较器	对象通道	AN000 ~ AN002	AN100 ~ AN102	
	基准电压的 设定基准	从外部引脚指定	CVREF: AN003、CVREFH: AN103	
		内部生成	1/8AVCC0、2/8AVCC0、3/8AVCC0、4/8AVCC0、 5/8AVCC0、6/8AVCC0、7/8AVCC0	
	噪声消除功能	用 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		S12ADI0 CMP1	S12ADI1 CMP1	
模块停止的设定		MSTPCRA.MSTPA17 位	MSTPCRA.MSTPA16 位	

注. 详细内容请参照“9. 低功耗功能”。

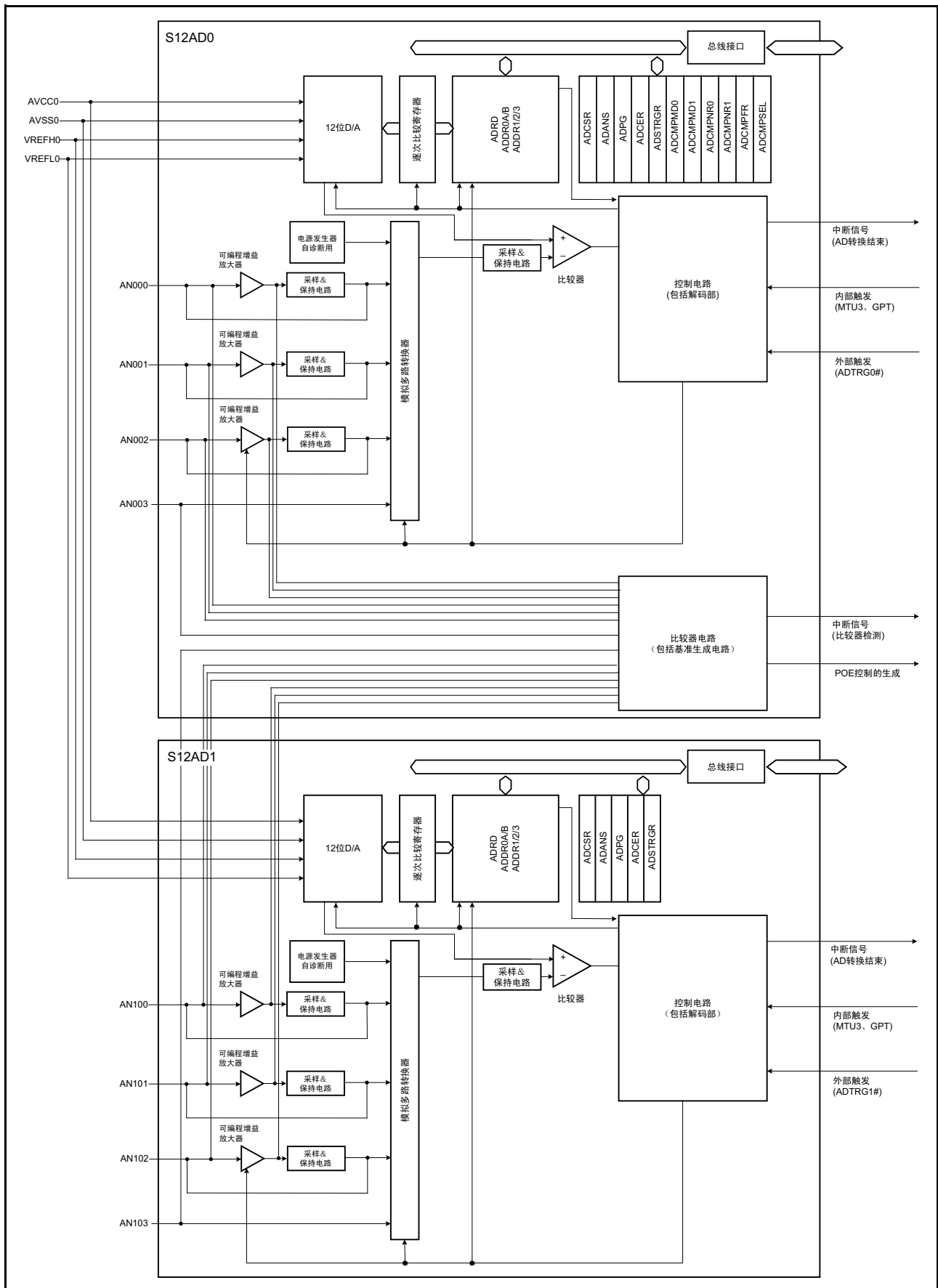


图 28.1 A/D 转换器的框图

12 位 A/D 转换器使用的输入引脚如表 28.3 所示。

12 位 A/D 转换器由 2 个单元构成。单元 0 (S12AD0) 和单元 1 (S12AD1) 能独立运行。S12AD0 和 S12AD1 的输入通道还能被分成 2 组进行运行。可编程增益放大器 (PGA) 和比较器内置于 AN000 ~ AN002、AN100 ~ AN102。

表 28.3 A/D 转换器的输入引脚

单元	引脚名	输入 / 输出	功能	内部 PGA	内部比较器
单元 0 (S12AD0)	AN000	输入	模拟输入引脚 0	内部	内部
	AN001	输入	模拟输入引脚 1	内部	内部
	AN002	输入	模拟输入引脚 2	内部	内部
	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 (当比较器运行并且选择在外部引脚外加基准电压时, 为比较器 Low 电平侧的基准电压引脚。)	—	—
	ADTRG0#	输入	用于开始 A/D 转换的外部触发输入引脚	—	—
单元 1 (S12AD1)	AN100	输入	模拟输入引脚 4	内部	内部
	AN101	输入	模拟输入引脚 5	内部	内部
	AN102	输入	模拟输入引脚 6	内部	内部
	AN103/ CVREFH	输入	模拟输入引脚 7/ 比较器 High 电平侧的基准电压引脚 (当比较器运行并且选择在外部引脚外加基准电压时, 为比较器 High 电平侧的基准电压引脚。)	—	—
	ADTRG1#	输入	用于开始 A/D 转换的外部触发输入引脚	—	—
共同	AVCC0	输入	模拟电路的电源引脚	—	—
	AVSS0	输入	模拟电路的接地引脚	—	—
	VREFH0	输入	A/D 转换器的模拟基准电源引脚	—	—
	VREFL0	输入	A/D 转换器的模拟基准电源接地引脚	—	—

注. 各引脚连接的单元不同。每个单元都有控制寄存器，必须分别进行设定。启动内部 PGA 时，必须在“33.4 A/D 转换特性”的“表 33.15 可编程增益放大器特性”中记载的模拟输入电压范围内使用。

## 28.2 寄存器说明

12 位 A/D 转换器的寄存器一览表如表 28.4 所示。

表 28.4 A/D 转换器的寄存器一览表

单元	模块符号	寄存器名	寄存器符号	复位后的值	地址	存取长度
0	S12AD0	A/D 控制寄存器	ADCSR	00h	0008 9000h	8
		A/D 通道选择寄存器	ADANS	0000h	0008 9004h	16
		A/D 可编程增益放大器寄存器	ADPG	0000h	0008 900Ah	16
		A/D 控制扩展寄存器	ADCER	0000h	0008 900Eh	16
		A/D 开始触发选择寄存器	ADSTRGR	0000h	0008 9010h	16
		A/D 数据寄存器 Diag	ADDRD	0000h	0008 901Eh	16
		A/D 数据寄存器 0A	ADDR0A	0000h	0008 9020h	16
		A/D 数据寄存器 1	ADDR1	0000h	0008 9022h	16
		A/D 数据寄存器 2	ADDR2	0000h	0008 9024h	16
		A/D 数据寄存器 3	ADDR3	0000h	0008 9026h	16
		A/D 数据寄存器 0B	ADDR0B	0000h	0008 9030h	16
		A/D 采样状态寄存器	ADSSTR	14h	0008 9060h	8
1	S12AD1	A/D 控制寄存器	ADCSR	00h	0008 9080h	8
		A/D 通道选择寄存器	ADANS	0000h	0008 9084h	16
		A/D 可编程增益放大器寄存器	ADPG	0000h	0008 908Ah	16
		A/D 控制扩展寄存器	ADCER	0000h	0008 908Eh	16
		A/D 开始触发选择寄存器	ADSTRGR	0000h	0008 9090h	16
		A/D 数据寄存器 Diag	ADDRD	0000h	0008 909Eh	16
		A/D 数据寄存器 0A	ADDR0A	0000h	0008 90A0h	16
		A/D 数据寄存器 1	ADDR1	0000h	0008 90A2h	16
		A/D 数据寄存器 2	ADDR2	0000h	0008 90A4h	16
		A/D 数据寄存器 3	ADDR3	0000h	0008 90A6h	16
		A/D 数据寄存器 0B	ADDR0B	0000h	0008 90B0h	16
		A/D 采样状态寄存器	ADSSTR	14h	0008 90E0h	8
共同	S12AD	比较器运行模式选择寄存器 0	ADCMPMD0	0000h	0008 9012h	16
		比较器运行模式选择寄存器 1	ADCMPMD1	0000h	0008 9014h	16
		比较器的滤波器模式寄存器 0	ADCMPNR0	0000h	0008 9016h	16
		比较器的滤波器模式寄存器 1	ADCMPNR1	0000h	0008 9018h	16
		比较器检测标志寄存器	ADCMPFR	00h	0008 901Ah	8
		比较器中断选择寄存器	ADCMPSEL	0000h	0008 901Ch	16

### 28.2.1 A/D 数据寄存器 n (ADDRn) (n=0A、0B、1~3)、A/D 数据寄存器 Diag (ADRD)

ADDRn 寄存器是保存 A/D 转换结果的 16 位只读寄存器。AN000 和 AN100 的数据寄存器有 2 组 (双数据寄存器)，当 A/D 转换开始源为“MTU3 的 TRGnAN 或者 TRGnBN (n=4、7)”、“GPT 的 GTADTRAnN 或者 GTADTRBnN (n=0~3)”时，AN000 和 AN100 的数据寄存器能分别保持转换结果。模拟输入通道和 ADDRn 寄存器的对应关系如表 28.5 所示。

A/D 数据寄存器 Diag (ADRD) 是通过自诊断保存 A/D 转换结果的 16 位只读寄存器。

在下述条件下，ADDRn 寄存器和 ADRD 寄存器的格式不同：

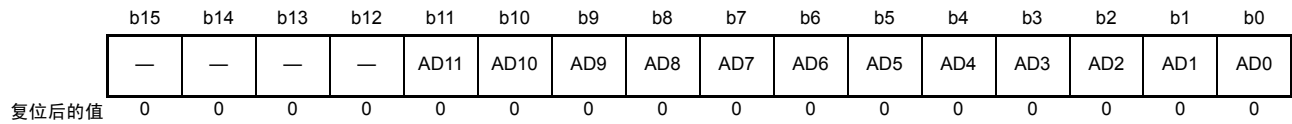
- A/D 数据寄存器格式选择位的设定值 (往右靠紧或者往左靠紧)
- A/D 数据寄存器位精度指定位的设定值 (12 位、10 位或者 8 位)

#### (1) ADDRn 寄存器 (n=0A、0B、1~3)

能通过设定 ADCER.ADRFMT 位，设定往右靠紧或者往左靠紧的格式。能通过设定 ADCER.ADPRC[1:0] 位，设定保存时的位精度。此时，ADDRn.AD11~0 位表示 12 位 A/D 转换值，其他位为保留位，读取值为“0”，写操作无效。

- 设定为往右靠紧格式、12 位精度的情况

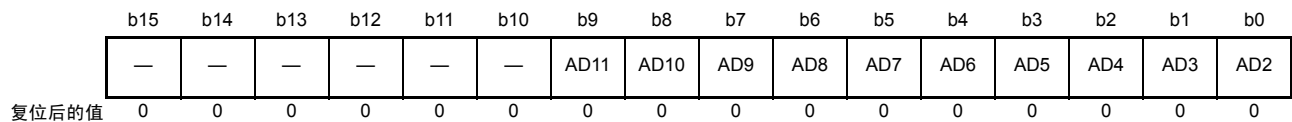
地址 S12AD0.ADDR0A 0008 9020h、S12AD0.ADDR1 0008 9022h、S12AD0.ADDR2 0008 9024h、S12AD0.ADDR3 0008 9026h、  
S12AD0.ADDR0B 0008 9030h  
S12AD1.ADDR0A 0008 90A0h、S12AD1.ADDR1 0008 90A2h、S12AD1.ADDR2 0008 90A4h、S12AD1.ADDR3 0008 90A6h、  
S12AD1.ADDR0B 0008 90B0h



位	符号	位名	功能	R/W
b11-b0	AD11 ~ AD0	转换值 11 ~ 0	12 位 A/D 转换值	R
b15-b12	—	保留位	读取值为“0”，写操作无效。	R

- 设定为往右靠紧格式、10 位精度的情况

地址 S12AD0.ADDR0A 0008 9020h、S12AD0.ADDR1 0008 9022h、S12AD0.ADDR2 0008 9024h、S12AD0.ADDR3 0008 9026h、  
S12AD0.ADDR0B 0008 9030h  
S12AD1.ADDR0A 0008 90A0h、S12AD1.ADDR1 0008 90A2h、S12AD1.ADDR2 0008 90A4h、S12AD1.ADDR3 0008 90A6h、  
S12AD1.ADDR0B 0008 90B0h



位	符号	位名	功能	R/W
b9-b0	AD11 ~ AD2	转换值 11 ~ 2	12 位 A/D 转换值的高 10 位。	R
b15-b10	—	保留位	读取值为“0”，写操作无效。	R

- 设定为往右靠紧格式、8位精度的情况

地址 S12AD0.ADDR0A 0008 9020h、S12AD0.ADDR1 0008 9022h、S12AD0.ADDR2 0008 9024h、S12AD0.ADDR3 0008 9026h、  
S12AD0.ADDR0B 0008 9030h  
S12AD1.ADDR0A 0008 90A0h、S12AD1.ADDR1 0008 90A2h、S12AD1.ADDR2 0008 90A4h、S12AD1.ADDR3 0008 90A6h、  
S12AD1.ADDR0B 0008 90B0h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	—	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4
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	AD11 ~ AD4	转换值 11 ~ 4	12 位 A/D 转换值的高 8 位。	R
b15-b8	—	保留位	读取值为“0”，写操作无效。。	R

- 设定为往左靠紧格式、12位精度的情况

地址 S12AD0.ADDR0A 0008 9020h、S12AD0.ADDR1 0008 9022h、S12AD0.ADDR2 0008 9024h、S12AD0.ADDR3 0008 9026h、  
S12AD0.ADDR0B 0008 9030h  
S12AD1.ADDR0A 0008 90A0h、S12AD1.ADDR1 0008 90A2h、S12AD1.ADDR2 0008 90A4h、S12AD1.ADDR3 0008 90A6h、  
S12AD1.ADDR0B 0008 90B0h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—
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
b3-b0	—	保留位	读取值为“0”，写操作无效。	R
b15-b4	AD11 ~ AD0	转换值 11 ~ 0	12 位 A/D 转换值	R

- 设定为往左靠紧格式、10位精度的情况

地址 S12AD0.ADDR0A 0008 9020h、S12AD0.ADDR1 0008 9022h、S12AD0.ADDR2 0008 9024h、S12AD0.ADDR3 0008 9026h、  
S12AD0.ADDR0B 0008 9030h  
S12AD1.ADDR0A 0008 90A0h、S12AD1.ADDR1 0008 90A2h、S12AD1.ADDR2 0008 90A4h、S12AD1.ADDR3 0008 90A6h、  
S12AD1.ADDR0B 0008 90B0h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	—	—	—	—	—	—
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
b5-b0	—	保留位	读取值为“0”，写操作无效。	R
b15-b6	AD11 ~ AD2	转换值 11 ~ 2	12 位 A/D 转换值的高 10 位	R



- 设定为往左靠紧格式、8 位精度的情况

地址 S12AD0.ADDR0A 0008 9020h、S12AD0.ADDR1 0008 9022h、S12AD0.ADDR2 0008 9024h、S12AD0.ADDR3 0008 9026h、  
S12AD0.ADDR0B 0008 9030h  
S12AD1.ADDR0A 0008 90A0h、S12AD1.ADDR1 0008 90A2h、S12AD1.ADDR2 0008 90A4h、S12AD1.ADDR3 0008 90A6h、  
S12AD1.ADDR0B 0008 90B0h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	—	—	—	—	—	—	—	—
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	读取值为“0”，写操作无效。	R
b15-b8	AD11 ~ AD4	转换值 11 ~ 4	12 位 A/D 转换值的高 8 位	R

## (2) ADRD 寄存器

能通过设定 ADCER.ADRFMT 位，设定往右靠紧或者往左靠紧的格式。此时，AD11 ~ AD0 位表示 12 位 A/D 转换值。另外，附加自诊断的状态位 (ADRD.DLAGST[1:0])。其他位为保留位，读取值为“0”，写操作无效。

- 设定为往右靠紧格式的情况

地址 S12AD0.ADRD 0008 901Eh、S12AD1.ADRA 0008 909Eh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DIAGST[1:0]	—	—	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—
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
b11-b0	AD11 ~ AD0	转换值 11 ~ 0	12 位 A/D 转换值。	R
b13-b12	—	保留位	读取值为“0”，写操作无效。	R
b15-b14	DIAGST[1:0]	自诊断状态位	b15 b14 0 1: 表示从上电后没有实施 1 次自诊断 0 1: 表示实施 VREFH0×0 的电压值自诊断 1 0: 表示实施 VREFH0×1/2 的电压值自诊断 1 1: 表示实施 VREFH0×1 的电压值自诊断  注. 自诊断的详细内容请参照“28.2.4 A/D 控制扩展寄存器 (ADCER)”。	R

- 设定为往左靠紧格式的情况

地址 S12AD0.ADRD 0008 901Eh、S12AD1.ADRA 0008 909Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD11	AD10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	DIAGST[1:0]	
复位后的值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

b1-b0	DIAGST[1:0]	自诊断状态位	b1 b0 0 1: 表示从上电后没有实施 1 次自诊断 0 1: 表示实施 VREFH0×0 的电压值自诊断 1 0: 表示实施 VREFH0×1/2 的电压值自诊断 1 1: 表示实施 VREFH0×1 的电压值自诊断  注. 自诊断的详细内容请参照“28.2.4 A/D 控制扩展寄存器 (ADCER)”。	R
b3-b2	—	保留位	读取值为“0”，写操作无效。	R
b15-b4	AD11 ~ AD0	转换值 11 ~ 0	12 位 A/D 转换值	R

表 28.5 模拟输入通道和 ADDRn 寄存器的对应关系

模拟输入通道	ADDRn 寄存器
AN000	S12AD0.ADDR0A/S12AD0.ADDR0B (注 1)
AN001	S12AD0.ADDR1
AN002	S12AD0.ADDR2
AN003	S12AD0.ADDR3
AN100	S12AD1.ADDR0A/S12AD1.ADDR0B (注 1)
AN101	S12AD1.ADDR1
AN102	S12AD1.ADDR2
AN103	S12AD1.ADDR3

- 注 1. • 当 A/D 转换开始源不为“MTU3 的 TRGnAN 或者 TRGnBN (n=4、7)”、“GPT 的 GTADTRAnN 或者 GTADTRBnN (n=0 ~ 3)” (ADSTRGR.ADSTRSn[4:0] 位不为“01011b”、“01111b”或者“11001b ~ 11100b”) 时, 转换结果保存到 ADDR0A 寄存器。  
• 当 A/D 转换开始源为“MTU3 的 TRGnAN 或者 TRGnBN (n=4、7)” (ADSTRGR.ADSTRSn[4:0] 位为“01011b”、“01111b”) 时, 通过 TRGnAN 开始转换的结果保存到 ADDR0A 寄存器, 通过 TRGnBN 开始转换的结果保存到 ADDR0B 寄存器。  
• 当 A/D 转换开始源为“GPT 的 GTADTRAnN 或者 GTADTRBnN (n=0 ~ 3)” (ADSTRGR.ADSTRSn[4:0] 位为“11001b ~ 11100b”) 时, 通过 GTADTRAnN 开始转换的结果保存到 ADDR0A 寄存器, 通过 GTADTRBnN 开始转换的结果保存到 ADDR0B 寄存器。

## 28.2.2 A/D 控制寄存器 (ADCSR)

地址 S12AD0.ADCSR 0008 9000h、S12AD1.ADCSR 0008 9080h



位	符号	位名	功能	R/W
b0	EXTRG	触发选择位	0: 选择由 A/D 开始触发选择寄存器 (ADSTRGR) 选择的定时器源引起的扫描转换开始 1: 选择由外部触发 (ADTRGn#) 引起的扫描转换开始 (n=0、1)	R/W
b1	TRGE	触发允许位	0: 禁止由外部触发 (ADTRGn#) 或者 MTU3、GPT 触发引起的 A/D 转换 1: 允许由外部触发 (ADTRGn#) 或者 MTU3、GPT 触发引起的 A/D 转换 (n=0、1)	R/W
b3-b2	CKS[1:0]	时钟选择位	b3 b2 0 0: PCLK/8 0 1: PCLK/4 1 0: PCLK/2 1 1: PCLK	R/W
b4	ADIE	A/D 转换结束中断允许位	0: 禁止 A/D 转换结束后的 S12ADI 中断 1: 允许 A/D 转换结束后的 A12ADI 中断	R/W
b6-b5	ADCS[1:0]	A/D 转换模式选择位	b6 b5 0 0: 单次模式 0 1: 1 个周期扫描模式 1 0: 连续扫描模式 1 1: 2 个通道扫描模式	R/W
b7	ADST	A/D 开始位	0: 停止 A/D 转换 1: 开始 A/D 转换	R/W

注. 通过外部触发开始 A/D 转换的方法:

在给外部触发引脚 (ADTRGn#) 输入 High 电平的状态下, 将 TRGE 位置“1”并且将 EXTRG 位置“1”。此后, 如果使 ADTRGn# 的信号变为 Low 电平, 就检测到 ADTRGn# 的下降沿并且开始 A/D 转换。此时, Low 电平输入的脉宽至少需要 1.5 个 PCLK 时钟。(n=0、1)

ADCSR 寄存器选择时钟、开始或者停止 A/D 转换以及设定 A/D 转换模式和 A/D 转换触发。必须在 ADST 位为“0”时设定 ADCS[1:0] 位和 CKS[1:0] 位。

## CKS[1:0] 位 (时钟选择位)

这些位设定决定 A/D 转换时间的 A/D 转换时钟 (ADCLK) 的频率。

ADCLK 的频率必须设定为大于等于 4MHz。在 AVCC0=3.0 ~ 3.6V 下使用时, ADCLK 的频率必须设定为小于等于 25MHz。

详细内容请参照“28.3.3 模拟输入的采样和 A/D 转换时间”。

**ADIE 位 (A/D 转换结束中断允许位)**

此位允许或者禁止 A/D 扫描转换结束中断 (S12ADI) 的发生。

在对象通道的 A/D 转换结束后, 如果 ADIE 位为“1”, 就发生 A/D 扫描转换结束中断 (S12ADI)。

**ADCS[1:0] 位 (A/D 转换模式选择位)**

此位选择 A/D 转换模式。

在单次模式中, 对 ADANS.CH[1:0] 位选择的 1 个通道的模拟输入进行 1 次 A/D 转换。

在 1 个周期扫描模式中, 按照从小到大的通道号顺序对 ADANS.CH[1:0] 位选择的最多 4 个通道的模拟输入只进行 1 次 A/D 转换, 一旦选择的全部通道转换结束, 就停止 A/D 转换。

在连续扫描模式中, 在 ADCSR.ADST 为“1”期间, 按照从小到大的通道号顺序对 ADANS.CH[1:0] 位选择的最多 4 个通道的模拟输入进行 A/D 转换, 一旦选择的全部通道转换结束, 就返回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 2 个通道扫描模式中, 将 4 个通道的模拟输入分为 2 组, 并且能分别选择由不同的触发引起的开始源。按照从小到大的通道号顺序对 ADANS.CH[1:0] 位选择的最多 3 个通道的模拟输入只进行 1 次 A/D 转换, 一旦选择的全部通道转换结束, 就停止 A/D 转换。

**ADST 位 (A/D 开始位)**

此位控制 A/D 转换的开始和停止。必须在将 ADST 位置“1”前设定 A/D 转换时钟和运行模式。

[ 为“1”的条件 ]

- 通过软件写“1”时
- 将 TRGE 位置“1”、EXTRG 位置“0”, 并且检测到 ADSTRGR.ADSTRSn[4:0] 位选择的 MTU3 触发或者 GPT 触发时
- 将 TRGE 位和 EXTRG 位置“1”, 将 ADSTRGR.ADSTRSn[4:0] 位置“00000b”, 并且检测到外部触发时

[ 为“0”的条件 ]

- 通过软件写“0”时
- 在单次模式中 A/D 转换结束时
- 在 1 个周期扫描模式中所选通道的 A/D 转换全部结束时
- 在 2 个通道扫描模式中每组所选通道的 A/D 转换全部结束时

## 28.2.3 A/D 通道选择寄存器 (ADANS)

## (1) S12AD0.ADANS

地址 0008 9004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CH[1:0]	—	PG002SEL	PG001SEL	PG000SEL	—	—	—	—	—	PG002EN	PG001EN	PG000EN	
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	PG000EN	AN000 的可编程增益放大器允许位	0: 禁止可编程增益放大器运行 1: 允许可编程增益放大器运行	R/W
b1	PG001EN	AN001 的可编程增益放大器允许位	0: 禁止可编程增益放大器运行 1: 允许可编程增益放大器运行	R/W
b2	PG002EN	AN002 的可编程增益放大器允许位	0: 禁止可编程增益放大器运行 1: 允许可编程增益放大器运行	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W
b8	PG000SEL	AN000 的可编程增益放大器选择位	0: 不使用可编程增益放大器 (旁路) 1: 允许可编程增益放大器	R/W
b9	PG001SEL	AN001 的可编程增益放大器选择位	0: 不使用可编程增益放大器 (旁路) 1: 允许可编程增益放大器	R/W
b10	PG002SEL	AN002 的可编程增益放大器选择位	0: 不使用可编程增益放大器 (旁路) 1: 允许可编程增益放大器	R/W
b11	—	保留位	读写值都为“0”。	R/W
b13-b12	CH[1:0]	通道设定位	通过 S12AD0 选择进行 A/D 转换的模拟输入通道。 详细内容请参照表 28.6。	R/W
b15-b14	—	保留位	读写值都为“0”。	R/W

## (2) S12AD1.ADANS

地址 0008 9084h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	CH[1:0]	—	PG102SEL	PG101SEL	PG100SEL	—	—	—	—	—	PG102EN	PG101EN	PG100EN	
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	PG100EN	AN100 的可编程增益放大器允许位	0: 禁止可编程增益放大器运行 1: 允许可编程增益放大器运行	R/W
b1	PG101EN	AN101 的可编程增益放大器允许位	0: 禁止可编程增益放大器运行 1: 允许可编程增益放大器运行	R/W
b2	PG102EN	AN102 的可编程增益放大器允许位	0: 禁止可编程增益放大器运行 1: 允许可编程增益放大器运行	R/W
b7-b3	—	保留位	读写值都为“0”。	R/W
b8	PG100SEL	AN100 的可编程增益放大器选择位	0: 不使用可编程增益放大器 (旁路) 1: 允许可编程增益放大器	R/W
b9	PG101SEL	AN101 的可编程增益放大器选择位	0: 不使用可编程增益放大器 (旁路) 1: 允许可编程增益放大器	R/W
b10	PG102SEL	AN102 的可编程增益放大器选择位	0: 不使用可编程增益放大器 (旁路) 1: 允许可编程增益放大器	R/W
b11	—	保留位	读写值都为“0”。	R/W
b13-b12	CH[1:0]	通道设定位	通过 S12AD0 选择进行 A/D 转换的模拟输入通道。 详细内容请参照表 28.6。	R/W
b15-b14	—	保留位	读写值都为“0”。	R/W

ADANS 寄存器设定要进行转换的对象通道和是否使用 AN000 ~ AN002/AN100 ~ AN102 的可编程增益放大器, 以及禁止或者允许 AN000 ~ AN002/AN100 ~ AN102 的可编程增益放大器运行。

必须在 ADCSR.ADST 位为“0”设定 ADANS 寄存器。

PGnEN 位 (ANn 的可编程增益放大器允许位) (n=000 ~ 002、100 ~ 102)

此位设定禁止或者允许可编程增益放大器运行。

PGnSEL 位 (ANn 的可编程增益放大器选择位) (n=000 ~ 002、100 ~ 102)

此位设定是否使用可编程增益放大器。

CH[1:0] 位 (通道设定位)

这些位设定要进行转换的对象通道。内容根据 ADCSR.ADCS[1:0] 位的设定而变。详细内容请参照表 28.6。

表 28.6 进行转换的对象通道的设定

ADANS.CH[1:0]		模拟输入通道			
b1	b0	单次模式		1 个周期扫描模式 / 连续扫描模式	
		S12AD0	S12AD1	S12AD0	S12AD1
0	0	AN000	AN100	AN000	AN100
0	1	AN001	AN101	AN000、AN001	AN100、AN101
1	0	AN002	AN102	AN000 ~ AN002	AN100 ~ AN102
1	1	AN003	AN103	AN000 ~ AN003	AN100 ~ AN103

ADANS.CH[1:0]		模拟输入通道	
b1	b0	2 个通道扫描模式 (通过定时器触发 / 外部触发开始)	
		S12AD0	S12AD1
0	0	组 0: AN000 组 1: AN001 ~ AN003	组 0: AN100 组 1: AN101 ~ AN103
0	1	组 0: AN000、AN001 组 1: AN002、AN003	组 0: AN100、AN101 组 1: AN102、AN103
1	0	组 0: AN000 ~ AN002 组 1: AN003	组 0: AN100 ~ AN102 组 1: AN103
1	1	不能设定 (与 00b 的设定相同)	不能设定 (与 00b 的设定相同)

ADANS.CH[1:0]		模拟输入通道	
b1	b0	2 个通道扫描模式 (通过软件开始)	
		S12AD0	S12AD1
0	0	AN000	AN100
0	1	AN000、AN001	AN100、AN101
1	0	AN000 ~ AN002	AN100 ~ AN102
1	1	AN000 ~ AN003	AN100 ~ AN103

## 28.2.4 A/D 控制扩展寄存器 (ADCER)

地址 S12AD0.ADCER 0008 900Eh、S12AD1.ADCER 0008 908Eh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADRFMT	—	ADIEW	ADIE2	DIAGM	DIAGLD	DIAGVAL[1:0]	—	—	ACE	—	—	—	ADPRC[1:0]	SHBYP	
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	SHBYP	通道专用采样 & 保持电路选择位	0: 使用采样 & 保持电路 1: 不使用采样 & 保持电路	R/W
b2-b1	ADPRC[1:0]	A/D 数据寄存器位精度指定位	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 (以 12 位精度保存)	R/W
b4-b3	—	保留位	读写值都为“0”。	R/W
b5	ACE	自动清除允许位	0: 禁止通过读 ADDRn 和 ADRD 自动清除 ADDRn 和 ADRD 1: 允许通过读 ADDRn 和 ADRD 自动清除 ADDRn 和 ADRD	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: 不能设定 0 1: 使用 VREFH0×0 的电压进行自诊断 1 0: 使用 VREFH0×1/2 的电压进行自诊断 1 1: 使用 VREFH0×1 的电压进行自诊断	R/W
b10	DIAGLD	自诊断模式选择位	0: 自诊断电压自动轮流转换 1: 自诊断电压固定为 DIAGVAL[1:0] 位的设定进行转换	R/W
b11	DIAGM	自诊断允许位	0: 不实施 A/D 转换器的自诊断 1: 实施 A/D 转换器的自诊断	R/W
b12	ADIE2	2 个通道扫描中断选择位	0: 分别在组 0 触发和组 1 触发的转换结束时发生 S12ADI 中断 1: 在组 0 触发和组 1 触发的转换都结束时发生 S12ADI 中断	R/W
b13	ADIEW	双触发中断选择位	0: 分别在双触发的转换结束时发生 S12ADI 中断 1: 在双触发的转换都结束时发生 S12ADI 中断	R/W
b14	—	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: A/D 数据寄存器的格式为往右靠紧 1: A/D 数据寄存器的格式为往左靠紧	R/W

ADCER 寄存器设定 A/D 数据寄存器的格式、自诊断模式、自动清除和中断，并且选择通道专用采样 & 保持电路。必须在 ADCSR.ADST 位为“0”时设定 ADCER 寄存器。

## SHBYP 位 (通道专用采样 &amp; 保持电路选择位)

此位选择是否使用通道专用采样 & 保持电路。

如果要使用可编程增益放大器，必须将 SHBYP 位设定为“0” (使用采样 & 保持电路)。



**ADPRC 位 (A/D 数据寄存器位精度选择位)**

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 A/D 数据寄存器。

**ACE 位 (自动清除允许位)**

此位选择是否在通过 CPU 或者 DTC 读 ADDR<sub>n</sub> 寄存器和 ADRD 寄存器后自动清除该寄存器。能通过自动清除检测到 ADDR<sub>n</sub> 寄存器和 ADRD 寄存器的未更新故障。

**DIAGVAL[1:0] 位 (自诊断转换电压选择位)**

详细内容请参照 DIAGLD 位的说明。在 DIAGVAL[1:0] 位的初始值为“00b” (禁止设定) 的状态下, 将 DIAGLD 位置“1”, 就禁止实施自诊断。

**DIAGLD 位 (自诊断模式选择位)**

此位选择是轮流转换通过自诊断转换的 3 个电压还是固定电压值。

如果将 DIAGLD 位置“0”, 就按 VREFH0×0 → VREFH0×1/2 → VREFH0×1 的顺序轮流转换。通过上电复位从 VREFH0×0 开始自诊断, 即使扫描转换结束, 也不返回 VREFH0×0。如果再次进行扫描转换, 就接着上一次进行轮流转换。

如果将 DIAGLD 位置“1”, 就固定为 DIAGVAL[1:0] 位选择的电压进行转换 (不进行自动轮流转换)。如果再次将 DIAGLD 位置“0”, 就从固定的电压值开始轮流转换 (加载功能)。

**DIAGM 位 (自诊断允许位)**

自诊断是为了检测 A/D 转换器故障的功能。转换在内部生成的 VREFH0×0、VREFH0×1/2 和 VREFH0×1 这 3 个电压值中的任意一个电压值。如果转换结束, 就将转换的电压信息和转换值保存到 A/D 数据寄存器 Diag (ADRD)。之后, 通过软件读 ADRD, 由此判断转换值是 (正常) 否 (异常) 在正常的范围内。在通过扫描转换进行最小通道号的通道转换前实施自诊断。

一旦实施自诊断, 3 个电压值中的 1 个就被转换。每当实施自诊断, 3 个电压值就自动轮流转换。自诊断的实施时间与 1 个通道的 A/D 转换时间相同。

**ADIE2 位 (2 个通道扫描中断选择位)**

此位选择在选择 2 个通道扫描模式时的 S12ADI 中断发生时序。ADIE2 位只在 2 个通道扫描模式中并且设定为通过触发开始 A/D 转换 (ADCSR.TRGE=1) 时有效。

**ADIEW 位 (双触发中断选择位)**

此位选择在选择双触发时的 S12ADI 中断发生时序。ADIEW 位只在设定为通过触发开始 A/D 转换 (ADCSR.TRGE=1) 并且选择触发源为“TRG4AN 或者 TRG4BN”、“TRG7AN 或者 TRG7BN”、“GTADTRA0N 或者 GTADTRB0N”、“GTADTRA1N 或者 GTADTRB1N”、“GTADTRA2N 或 GTADTRB2N”、“GTADTRA3N 或者 GTADTRB3N” 时有效。

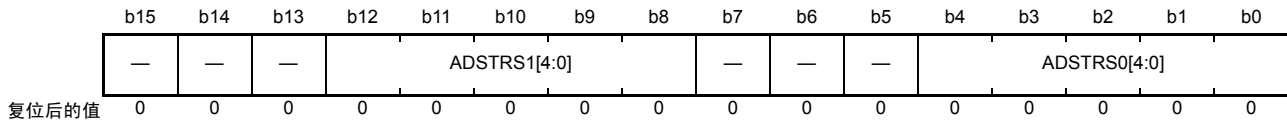
**ADRFMT 位 (A/D 数据寄存器格式选择位)**

此位选择 A/D 数据寄存器保存的数据是往右靠紧还是往左靠紧。

有关 A/D 数据寄存器的格式的详细内容, 请参照“28.2.1 A/D 数据寄存器 n (ADDR<sub>n</sub>) (n=0A、0B、1 ~ 3)、A/D 数据寄存器 Diag (ADRD)”。

## 28.2.5 A/D 开始触发选择寄存器 (ADSTRGR)

地址 S12AD0.ADSTRGR 0008 9010h、S12AD1.ADSTRGR 0008 9090h



位	符号	位名	功能	R/W
b4-b0	ADSTRS0[4:0]	A/D 开始触发组 0 选择位	通过 bit4 ~ 0 的组合选择内置外围 I/O 的 A/D 转换开始源。必须指定单次模式、1 个周期扫描模式、连续扫描模式中的开始源或者 2 个通道扫描模式中组 0 的开始源。开始源和设定值的关系如表 28.7 所示。	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W
b12-b8	ADSTRS1[4:0]	A/D 开始触发组 1 选择位	此位只用于 2 个通道扫描模式。 通过 bit4 ~ 0 的组合选择内置外围 I/O 的 A/D 转换开始源。必须指定 2 个通道扫描模式中组 1 的开始源。开始源和设定值的关系如表 28.7 所示。 在 2 个通道扫描模式中，必须给组 0 和组 1 的转换请求分别指定不同的开始源，使组 0 和组 1 的转换请求不同时产生。	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。

必须在 ADCSR.ADST 位为“0”时设定 ADSTRGR 寄存器。

- 软件触发 (ADCSR.ADST) 与 ADCSR.TRGE 位、ADCSR.EXTRG 位和 ADSTRGR 寄存器的设定值无关，总是有效。
- 在使用外部输入的 A/D 转换开始源 (ADTRGn#) 时，必须将 ADCSR.TRGE 位置“1”并且将 ADCSR.EXTRG 位置“1”。(n=0、1)
- 在使用 MTU3、GPT 的 A/D 转换开始源时，必须将 ADCSR.TRGE 位置“1”并且将 ADCSR.EXTRG 位置“0”。

表 28.7 A/D 转换开始源的选择一览表

发生源	A/D 转换开始源	开始条件	ADSTRSn[4]	ADSTRSn[3]	ADSTRSn[2]	ADSTRSn[1]	ADSTRSn[0]
软件	ADST	复位 ADCSR.ADST 位	—	—	—	—	—
外部引脚	ADTRGn#	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 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 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 的比较匹配以及 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 的比较匹配以及 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

注. 如果将开始 A/D 转换的触发设定为 ADTRGn#, 就必须将对应引脚的 PORTm.DDR.Bj 位置“0” (输入端口) 并且将 PORTm.ICR.Bj 位置“1” (对应引脚的输入缓冲器有效)。详细内容请参照“15. I/O 端口”。(n=0、1, m=A、2, j=0、1、4、5)

## 28.2.6 A/D 可编程增益放大器寄存器 (ADPG)

## (1) S12AD0.ADPG

地址 0008 900Ah

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	PG002GAIN[3:0]			PG001GAIN[3:0]			PG000GAIN[3:0]					
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
b3-b0	PG000GAIN[3:0]	AN000 的可编程增益放大器的增益选择位	b3 b0 0 0 0 0: ×2.0 0 0 0 1: ×2.5 0 0 1 0: 不能设定 0 0 1 1: 不能设定 0 1 0 0: ×3.077 0 1 0 1: 不能设定 0 1 1 0: ×3.636 0 1 1 1: ×4.0 1 0 0 0: ×4.444 1 0 0 1: ×5.0 1 0 1 0: ×5.714 1 0 1 1: ×6.667 1 1 0 0: 不能设定 1 1 0 1: ×10.0 1 1 1 0: ×13.333 1 1 1 1: 不能设定	R/W
b7-b4	PG001GAIN[3:0]	AN001 的可编程增益放大器的增益选择位	b7 b4 0 0 0 0: ×2.0 0 0 0 1: ×2.5 0 0 1 0: 不能设定 0 0 1 1: 不能设定 0 1 0 0: ×3.077 0 1 0 1: 不能设定 0 1 1 0: ×3.636 0 1 1 1: ×4.0 1 0 0 0: ×4.444 1 0 0 1: ×5.0 1 0 1 0: ×5.714 1 0 1 1: ×6.667 1 1 0 0: 不能设定 1 1 0 1: ×10.0 1 1 1 0: ×13.333 1 1 1 1: 不能设定	R/W
b11-b8	PG002GAIN[3:0]	AN002 的可编程增益放大器的选择位增益	b11 b8 0 0 0 0: ×2.0 0 0 0 1: ×2.5 0 0 1 0: 不能设定 0 0 1 1: 不能设定 0 1 0 0: ×3.077 0 1 0 1: 不能设定 0 1 1 0: ×3.636 0 1 1 1: ×4.0 1 0 0 0: ×4.444 1 0 0 1: ×5.0 1 0 1 0: ×5.714 1 0 1 1: ×6.667 1 1 0 0: 不能设定 1 1 0 1: ×10.0 1 1 1 0: ×13.333 1 1 1 1: 不能设定	R/W
b15-b12	—	保留位	读写值都为“0”。	R/W

## (2) S12AD1.ADPG

地址 0008 908Ah

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	PG102GAIN[3:0]			PG101GAIN[3:0]			PG100GAIN[3:0]					
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
b3-b0	PG100GAIN[3:0]	AN100 的可编程增益放大器的增益选择位	b3 b0 0 0 0 0: ×2.0 0 0 0 1: ×2.5 0 0 1 0: 不能设定 0 0 1 1: 不能设定 0 1 0 0: ×3.077 0 1 0 1: 不能设定 0 1 1 0: ×3.636 0 1 1 1: ×4.0 1 0 0 0: ×4.444 1 0 0 1: ×5.0 1 0 1 0: ×5.714 1 0 1 1: ×6.667 1 1 0 0: 不能设定 1 1 0 1: ×10.0 1 1 1 0: ×13.333 1 1 1 1: 不能设定	R/W
b7-b4	PG101GAIN[3:0]	AN101 的可编程增益放大器的增益选择位	b7 b4 0 0 0 0: ×2.0 0 0 0 1: ×2.5 0 0 1 0: 不能设定 0 0 1 1: 不能设定 0 1 0 0: ×3.077 0 1 0 1: 不能设定 0 1 1 0: ×3.636 0 1 1 1: ×4.0 1 0 0 0: ×4.444 1 0 0 1: ×5.0 1 0 1 0: ×5.714 1 0 1 1: ×6.667 1 1 0 0: 不能设定 1 1 0 1: ×10.0 1 1 1 0: ×13.333 1 1 1 1: 不能设定	R/W
b11-b8	PG102GAIN[3:0]	AN102 的可编程增益放大器的增益选择位	b11 b8 0 0 0 0: ×2.0 0 0 0 1: ×2.5 0 0 1 0: 不能设定 0 0 1 1: 不能设定 0 1 0 0: ×3.077 0 1 0 1: 不能设定 0 1 1 0: ×3.636 0 1 1 1: ×4.0 1 0 0 0: ×4.444 1 0 0 1: ×5.0 1 0 1 0: ×5.714 1 0 1 1: ×6.667 1 1 0 0: 不能设定 1 1 0 1: ×10.0 1 1 1 0: ×13.333 1 1 1 1: 不能设定	R/W
b15-b12	—	保留位	读写值都为“0”。	R/W

ADPG 寄存器是设定装载在 AN000 ~ AN002、AN100 ~ AN102 的可编程增益放大器增益的寄存器。必须在 ADCSR.ADST 位为“0”时设定 ADPG 寄存器。

PGnGAIN[3:0] 位 (ANn 的可编程增益放大器的增益选择位) (n=000 ~ 002、100 ~ 102)

这些位选择可编程增益放大器的增益。

## 28.2.7 比较器运行模式选择寄存器 0 (ADCMPMD0)

地址 0008 9012h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	CEN102[1:0]	CEN101[1:0]	CEN100[1:0]	—	—	CEN002[1:0]	CEN001[1:0]	CEN000[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	CEN000[1:0]	AN000 的比较器选择位	b1 b0 0 0: 不使用比较器 0 1: 作为低电平比较器使用 (检测比低电平侧基准电压还要低的输入电压) 1 0: 作为高电平比较器使用 (检测比高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 (检测在低电平侧基准电压~高电平侧基准电压范围外的输入电压)	
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: 作为低电平比较器使用 (检测比低电平侧基准电压还要低的输入电压) 1 0: 作为高电平比较器使用 (检测比高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 (检测在低电平侧基准电压~高电平侧基准电压范围外的输入电压)	
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: 作为低电平比较器使用 (检测比低电平侧基准电压还要低的输入电压) 1 0: 作为高电平比较器使用 (检测比高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 (检测在低电平侧基准电压~高电平侧基准电压范围外的输入电压)	
b7-b6	—	保留位	读写值都为"0"。	
b9-b8	CEN100[1:0]	AN100 的比较器选择位	b9 b8 0 0: 不使用比较器 0 1: 作为低电平比较器使用 (检测比低电平侧基准电压还要低的输入电压) 1 0: 作为高电平比较器使用 (检测比高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 (检测在低电平侧基准电压~高电平侧基准电压范围外的输入电压)	
b11-b10	CEN101[1:0]	AN101 的比较器选择位	b11 b10 0 0: 不使用比较器 0 1: 作为低电平比较器使用 (检测比低电平侧基准电压还要低的输入电压) 1 0: 作为高电平比较器使用 (检测比高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 (检测在低电平侧基准电压~高电平侧基准电压范围外的输入电压)	
b13-b12	CEN102[1:0]	AN102 的比较器选择位	b13 b12 0 0: 不使用比较器 0 1: 作为低电平比较器使用 (检测比低电平侧基准电压还要低的输入电压) 1 0: 作为高电平比较器使用 (检测比高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 (检测在低电平侧基准电压~高电平侧基准电压范围外的输入电压)	
b15-b14	—	保留位	读写值都为"0"。	

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、100 ~ 102)

这些位设定是否使用各比较器以及运行模式。

## 28.2.8 比较器运行模式选择寄存器 1 (ADCMPMD1)

地址 0008 9014h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	VSELL1	VSELH1	CSEL1	—	VSELL0	VSELH0	CSEL0	—	REFH[2:0]			—	REFL[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
b2-b0	REFL[2:0]	比较器低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: AVCC0×1/8 0 1 0: AVCC0×2/8 0 1 1: AVCC0×3/8 1 0 0: AVCC0×4/8 1 0 1: AVCC0×5/8 1 1 0: AVCC0×6/8 1 1 1: AVCC0×7/8	R/W
b3	—	保留位	读写值都为“0”。	R/W
b6-b4	REFH[2:0]	比较器高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: AVCC0×1/8 0 1 0: AVCC0×2/8 0 1 1: AVCC0×3/8 1 0 0: AVCC0×4/8 1 0 1: AVCC0×5/8 1 1 0: AVCC0×6/8 1 1 1: AVCC0×7/8	R/W
b7	—	保留位	读写值都为“0”。	R/W
b8	CSEL0	AN000 ~ AN002 的比较器输入选择位	0: 将可编程增益放大器放大前的信号用作比较器输入 1: 将可编程增益放大器放大后的信号用作比较器输入	R/W
b9	VSELH0	AN000 ~ AN002 的比较器高电平侧基准电压选择位	0: 从 AN103 引脚输入高电平侧基准电压 1: 将 REFH[2:0] 位选择的内部电压用作高电平侧基准电压	R/W
b10	VSELL0	AN000 ~ AN002 的比较器低电平侧基准电压选择位	0: 从 AN003 引脚输入低电平侧基准电压 1: 将 REFH[2:0] 位选择的内部电压用作低电平侧基准电压	R/W
b11	—	保留位	读写值都为“0”。	R/W
b12	CSEL1	AN100 ~ AN102 的比较器输入选择位	0: 将可编程增益放大器放大前的信号用作比较器输入 1: 将可编程增益放大器放大后的信号用作比较器输入	R/W
b13	VSELH1	AN100 ~ AN102 的比较器高电平侧基准电压选择位	0: 从 AN103 引脚输入高电平侧基准电压 1: 将 REFH[2:0] 位选择的内部电压用作高电平侧基准电压	R/W
b14	VSELL1	AN100 ~ AN102 的比较器低电平侧基准电压选择位	0: 从 AN003 引脚输入低电平侧基准电压 1: 将 REFH[2:0] 位选择的内部电压用作低电平侧基准电压	R/W
b15	—	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 (比较器低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低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“表 33.16 比较器特性”的 REFL 电压范围。



**REFH[2:0] 位 (比较器高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高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“表 33.16 比较器特性”的 REFH 电压范围。

**CSEL0 位 (AN000 ~ AN002 的比较器输入选择位)**

此位设定 AN000 ~ AN002 的比较器输入。

**VSELH0 位 (AN000 ~ AN002 的比较器高电平侧基准电压选择位)**

此位设定 AN000 ~ AN002 的比较器高电平侧基准电压的输入方法。

**VSELL0 位 (AN000 ~ AN002 的比较器低电平侧基准电压选择位)**

此位设定 AN000 ~ AN002 的比较器低电平侧基准电压的输入方法。

**CSEL1 位 (AN100 ~ AN102 的比较器输入选择位)**

此位设定 AN100 ~ AN102 的比较器输入。

**VSELH1 位 (AN100 ~ AN102 的比较器高电平侧基准电压选择位)**

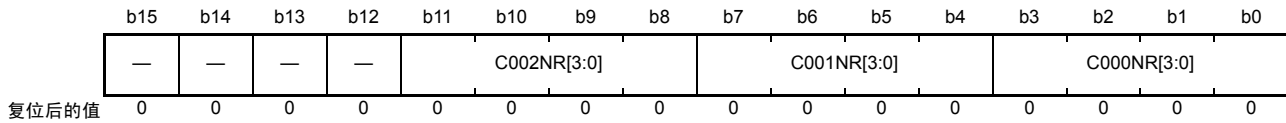
此位设定 AN100 ~ AN102 的比较器高电平侧基准电压的输入方法。

**VSELL1 位 (AN100 ~ AN102 的比较器低电平侧基准电压选择位)**

此位设定 AN100 ~ AN102 的比较器低电平侧基准电压的输入方法。

## 28.2.9 比较器的滤波器模式寄存器 0 (ADCMPNR0)

地址 0008 9016h



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

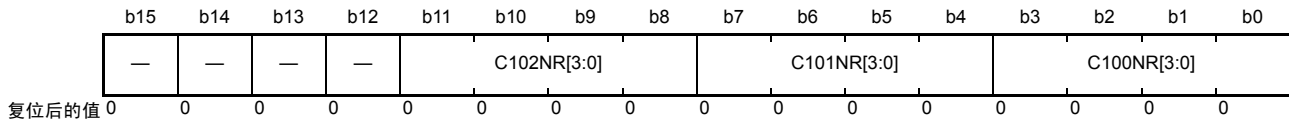
ADCMPNR0 寄存器是设定用于 AN000 ~ AN002 的比较器检测结果的噪声滤波器运行的寄存器。

## CnNR[3:0] 位 (ANn 的比较器噪声消除滤波器模式选择位) (n=000 ~ 002)

这些位设定用于 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 标志就置位。此时, 能通过设定 ADCMPSEL 寄存器, 请求比较器中断 (CMPI) 和端口输出允许 3 (POE3) 的启动。

## 28.2.10 比较器的滤波器模式寄存器 1 (ADCMPNR1)

地址 0008 9018h



位	符号	位名	功能	R/W
b3-b0	C100NR[3:0]	AN100 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	C101NR[3:0]	AN101 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	C102NR[3:0]	AN102 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

ADCMPNR1 寄存器是设定用于 AN100 ~ AN102 的比较器检测结果的噪声滤波器运行的寄存器。

## CnNR[3:0] 位 (ANn 的比较器噪声消除滤波器模式选择位) (n=100 ~ 102)

这些位设定用于 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 标志就置位。此时, 能通过设定 ADCMPSEL 寄存器, 请求比较器中断 (CMPI) 和端口输出允许 3 (POE3) 的启动。

## 28.2.11 比较器检测标志寄存器 (ADCMPFR)

地址 0008 901Ah

b7	b6	b5	b4	b3	b2	b1	b0
—	—	C102FLAG	C101FLAG	C100FLAG	C002FLAG	C001FLAG	C000FLAG
0	0	0	0	0	0	0	0

复位后的值

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)
b3	C100FLAG	AN100 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)
b4	C101FLAG	AN101 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)
b5	C102FLAG	AN102 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-b6	—	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、100 ~ 102)

这是表示各比较器的检测 / 未检测状态的标志。

[ 为“1”的条件 ]

- 用 ADCMPNRn.CmNR[3:0] (n=0、1, m=000 ~ 002、100 ~ 102) 位选择的时钟对比较器检测结果进行 16 次采样并且全部检测到时

[ 为“0”的条件 ]

- 通过软件写“0”时

## 28.2.12 比较器中断选择寄存器 (ADCMPSEL)

地址 0008 901Ch

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	POERQ	IE	—	—	SEL102	SEL101	SEL100	SEL002	SEL001	SEL000
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	SEL000	AN000 的比较器检测选择位	0: 不将比较器检测用作中断或者 POE 请求 1: 将比较器检测用作中断或者 POE 请求	R/W
b1	SEL001	AN001 的比较器检测选择位	0: 不将比较器检测用作中断或者 POE 请求 1: 将比较器检测用作中断或者 POE 请求	R/W
b2	SEL002	AN002 的比较器检测选择位	0: 不将比较器检测用作中断或者 POE 请求 1: 将比较器检测用作中断或者 POE 请求	R/W
b3	SEL100	AN100 的比较器检测选择位	0: 不将比较器检测用作中断或者 POE 请求 1: 将比较器检测用作中断或者 POE 请求	R/W
b4	SEL101	AN101 的比较器检测选择位	0: 不将比较器检测用作中断或者 POE 请求 1: 将比较器检测用作中断或者 POE 请求	R/W
b5	SEL102	AN102 的比较器检测选择位	0: 不将比较器检测用作中断或者 POE 请求 1: 将比较器检测用作中断或者 POE 请求	R/W
b7-b6	—	保留位	读写值都为“0”。	R/W
b8	IE	中断允许设定位	0: 禁止发生比较器检测时的 CMPI 中断 1: 允许发生比较器检测时的 CMPI 中断	R/W
b9	POERQ	POE 请求设定位	0: 禁止产生比较器检测时的 POE 请求 1: 允许产生比较器检测时的 POE 请求	R/W
b15-b10	—	保留位	读写值都为“0”。	R/W

ADCMPSEL 寄存器是设定将比较器检测用作中断或者 POE 请求的寄存器。

## SELn 位 (ANn 的比较器检测选择位) (n=000 ~ 002、100 ~ 102)

此位设定是否将各比较器检测用作中断或者 POE 请求。

## IE 位 (中断允许设定位)

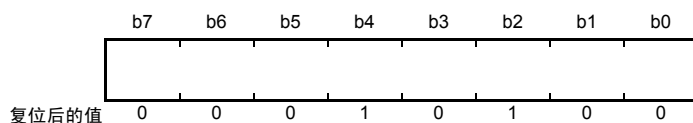
此位禁止或者允许发生比较器检测中断 (CMPI)。CMPI 中断为 SELn 位选择的比较器检测的逻辑或。

## POE 位 (POE 请求设定位)

此位禁止或者允许产生由比较器检测引起 POE 请求。POE 请求为 SELn 位选择的比较器检测的逻辑或。

### 28.2.13 A/D 采样状态寄存器 (ADSSTR)

地址 S12AD0.ADSSTR 0008 9060h、S12AD1.ADSSTR 0008 90E0h



ADSSTR 寄存器是用于设定模拟输入采样时间的 8 位可读写寄存器。

当模拟输入的信号源阻抗高并且采样时间不够或者 A/D 转换时钟 (ADCLK) 低速运行时，能调整采样时间。

必须设定为大于等于“0Dh”的值。

为了避免误动作，必须在 A/D 转换停止 (ADCSR.ADST 位 =0) 的状态下进行改写。

详细内容请参照“28.3.3 模拟输入的采样和 A/D 转换时间”。

## 28.3 运行说明

### 28.3.1 单次模式

单次扫描模式对指定的 1 个通道的模拟输入只进行 1 次 A/D 转换，如下所示。

1. 如果通过软件、MTU3、GPT 或者外部触发输入将 ADCSR.ADST 位置“1”（开始 A/D 转换），就开始已选通道的 A/D 转换。
2. 一旦 A/D 转换结束，就将 A/D 转换结果保存到对应该通道的 A/D 数据寄存器 n（ADDRn）（n=0A、0B、1~3）。
3. 在 A/D 转换结束后，如果 ADCSR.ADIE 位为“1”（允许由 A/D 转换结束引起的 S12ADI 中断），就产生 S12ADI 中断请求。
4. ADCSR.ADST 位在 A/D 转换过程中保持“1”，一旦转换结束就自动变为“0”，并且 A/D 转换器进入待机状态。
5. 如果在 A/D 转换过程中将 ADCSR.ADST 位置“0”（停止 A/D 转换），就暂停转换并且 A/D 转换器进入待机状态。

给模拟输入选择 AN001 时的运行例子如图 28.2 所示。

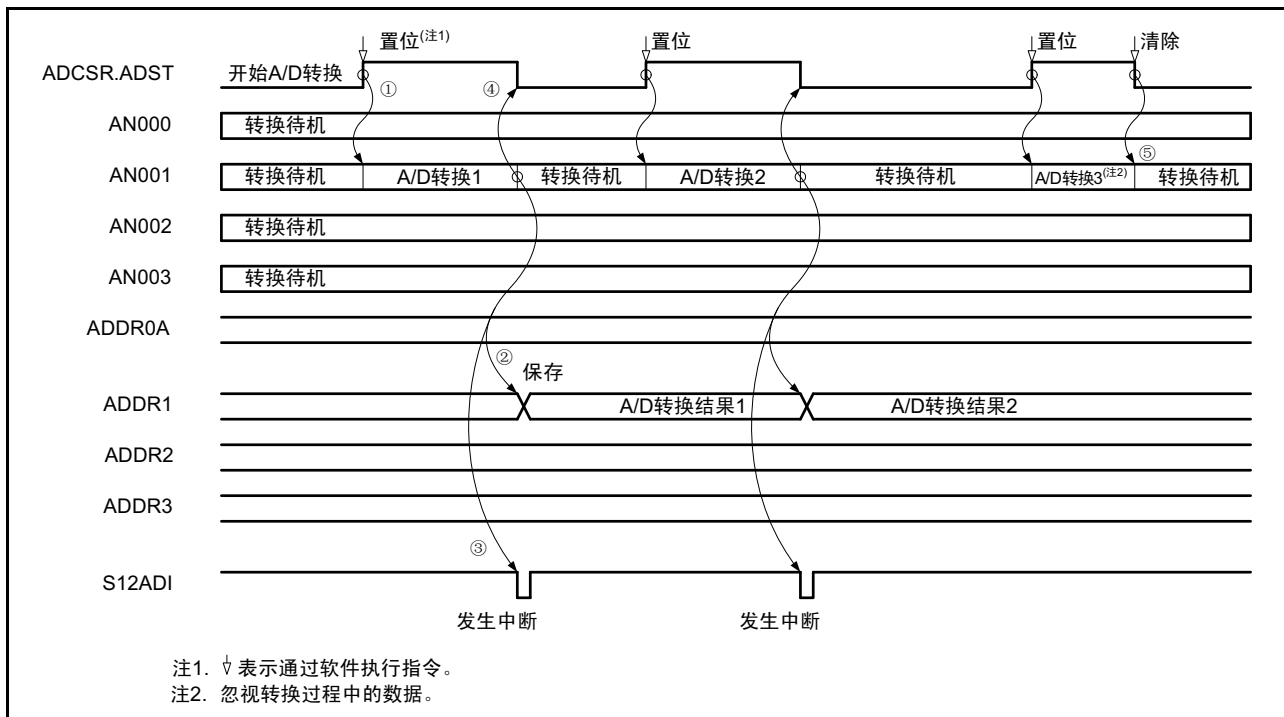


图 28.2 A/D 转换器的运行例子（单次模式）

### 28.3.2 扫描转换运行的说明

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 2 个通道扫描模式。

1 个周期扫描模式是在对指定的至少 1 个通道进行 1 次扫描后结束的模式。连续扫描模式是在通过软件将 ADCSR.ADST 位置“0”（从“1”变为“0”）前对指定的至少 1 个通道无限制的重复进行扫描的模式。2 个通道扫描模式是将 4 个通道的模拟输入分为组 0 和组 1，并且能给各组选择通过各自的触发引起的开始源的模式。组 0 和组 1 的转换运行和 1 个周期扫描模式相同（对至少 1 个通道进行 1 次扫描后结束）。

#### 28.3.2.1 1 个周期扫描模式

1 个周期扫描模式对所选通道的模拟输入只进行 1 个周期 A/D 转换，如下所示：

1. 如果通过软件、MTU3、GPT 或者外部触发输入将 ADCSR.ADST 位置“1”（开始 A/D 转换），就按照 ADANS 寄存器的 CH[1:0] 位设定的顺序开始 A/D 转换。
2. 一旦 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 n（ADDRn）（n=0A、0B、1~3）。
3. 如果在所选通道的 A/D 转换全部结束后 ADCSR.ADIE 位为“1”（允许由 A/D 转换结束引起的 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”，一旦所选通道的 A/D 转换全部结束就自动被清除，并且 A/D 转换器进入待机状态。

另外，采样运行根据 ADCER.SHBY 位的设定而不同。ADCER.SHBY=0 时的运行如图 28.3、ADCER.SHBY=1 时的运行如图 28.4 所示。

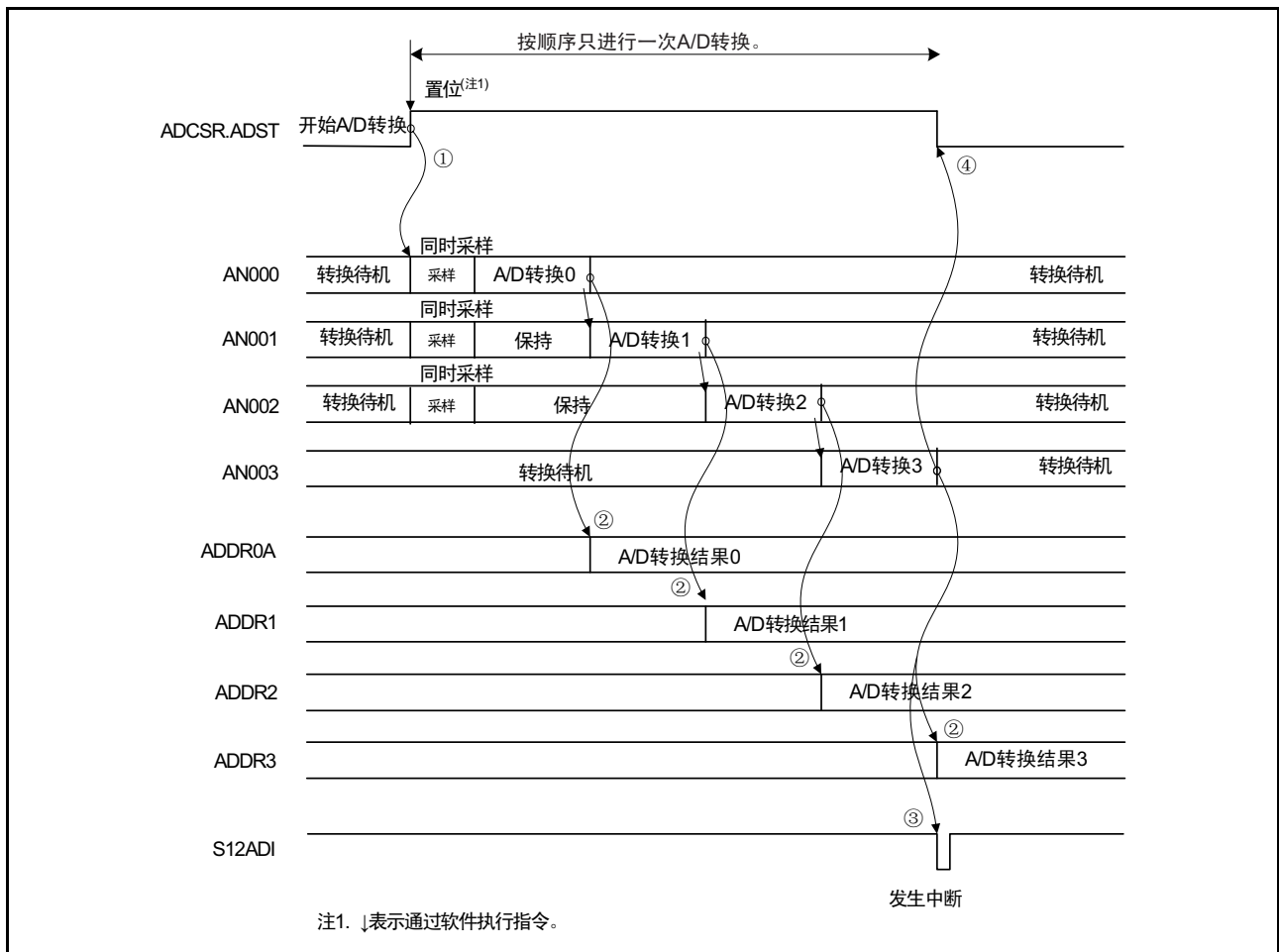


图 28.3 1 个周期扫描模式的运行例子（ADCER.SHBY=0：使用通道专用采样 & 保持电路时）



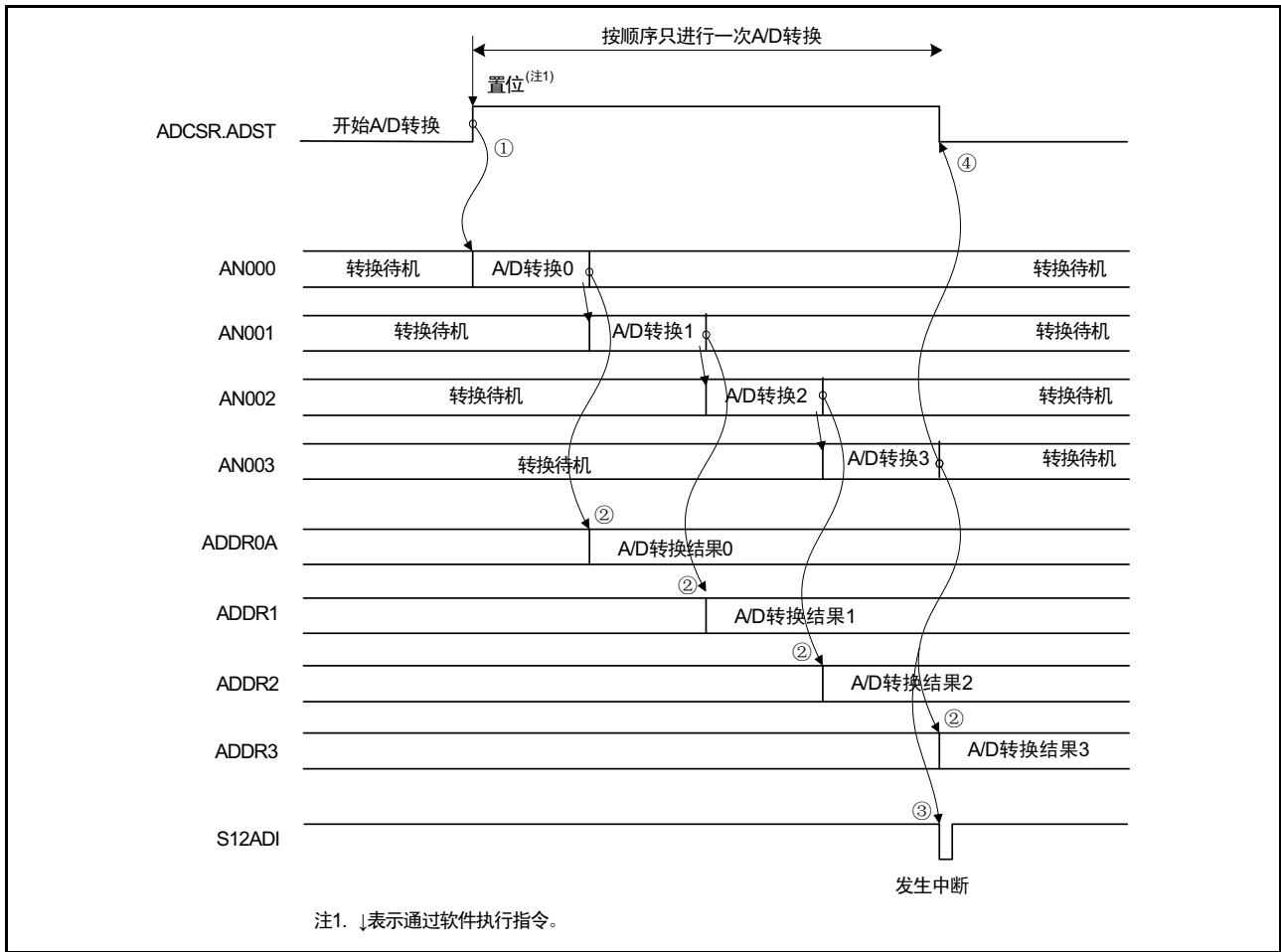


图 28.4 1 个周期扫描模式的运行例子 (ADCER.SHBY=1: 未使用通道专用采样 & 保持电路时)

28.3.2.2 连续扫描模式

连续扫描模式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、MTU3、GPT 或者外部触发输入将 ADCSR.ADST 位置“1”（开始 A/D 转换），就按照 ADANS 寄存器的 CH[1:0] 位设定的顺序开始 A/D 转换。
2. 一旦 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 n (ADDRn) (n=0A、0B、1~3)。
3. 如果在所选通道的 A/D 转换全部结束后 ADCSR.ADIE 位为“1”（允许由 A/D 转换结束引起的 S12ADI 中断），就产生 S12ADI 中断请求。A/D 转换器按照 ADANS 寄存器的 CH[1:0] 位设定的顺序开始 A/D 转换。
4. ADST 位不自动变被清除，而在为“1”的期间重复 2.~3. 的运行。如果将 ADST 位置“0”（停止 A/D 转换），就中止 A/D 转换并且 A/D 转换器进入待机状态。
5. 此后，如果将 ADST 位置“1”，就按照 ADANS 寄存器的 CH[1:0] 位设定的顺序重新开始 A/D 转换。另外，采样运行根据 ADCER.SHBY 位的设定而不同。ADCER.SHBY=0 时的运行如图 28.5、ADCER.SHBY=1 时的运行如图 28.6 所示。

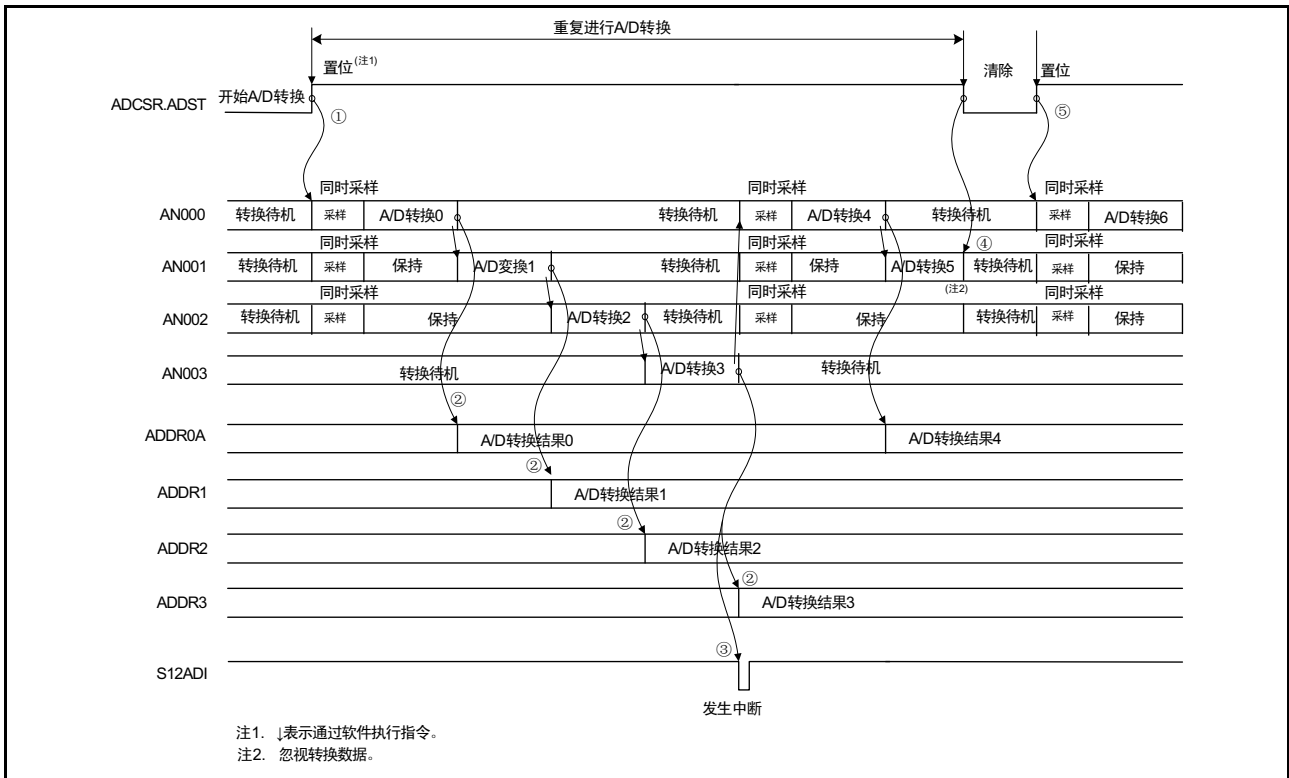


图 28.5 连续扫描模式的运行例子 (ADCER.SHBYP=0: 使用通道专用采样 & 保持电路时)

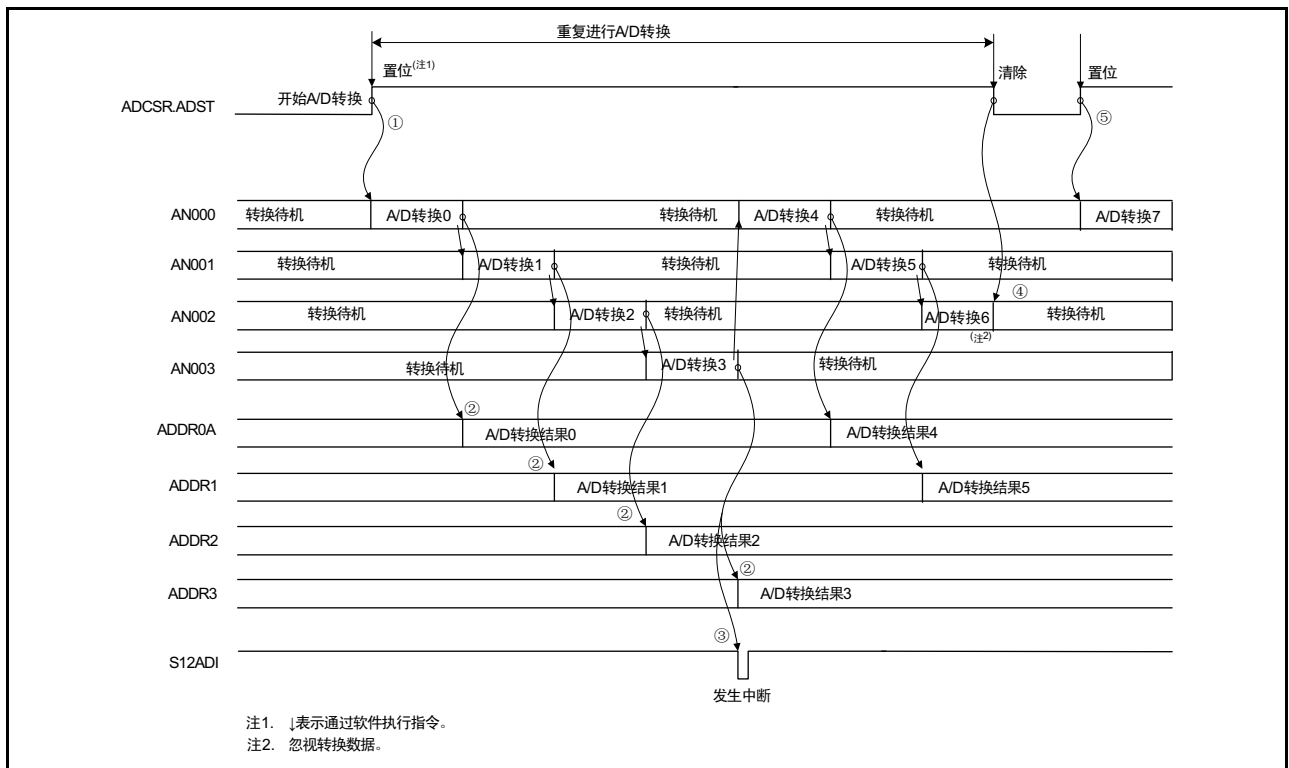


图 28.6 连续扫描模式的运行例子 (ADCER.SHBYP=1: 未使用通道专用采样 & 保持电路时)

## 28.3.2.3 2 个通道扫描模式

2 个通道扫描模式将 4 个通道的模拟输入分为组 0 和组 1，并且能给各组选择由不同触发引线的开始源。

能通过设定 ADCER.ADIE2 位选择是在组 0 或者组 1 转换结束后，还是在组 0 和组 1 转换都结束后发生 2 个通道扫描模式的转换结束中断。

在通过触发开始转换时，必须通过设定 ADSTRGR 寄存器分别给组 0 和组 1 设定不同的转换开始源。

另外，如果在组 0 的转换过程中产生组 1 的转换请求，就忽视组 1 的转换请求。

给组 0 的 A/D 转换开始请求设定 MTU4 的 TRG4AN，给组 1 的 A/D 转换开始请求设定 MTU4 的 TRG4BN，并且将 ADANS.CH[1:0] 位置 "01b" 时的运行例子如图 28.7 所示。

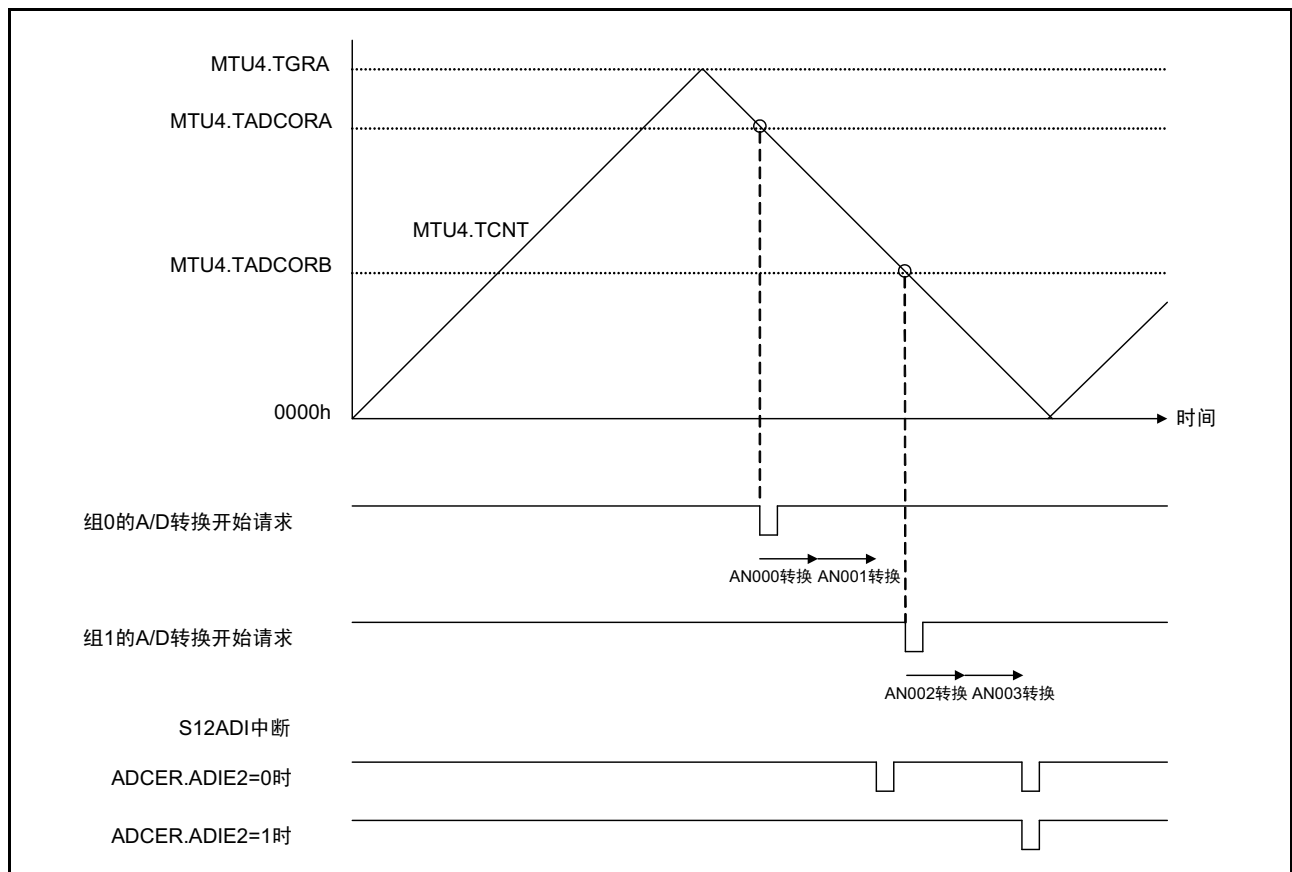


图 28.7 2 个通道扫描模式的运行例子

### 28.3.3 模拟输入的采样和 A/D 转换时间

能选择通过软件、MTU3、GPT 触发或者 ADTRGn#（外部触发）开始转换。在 A/D 转换开始延迟时间（tD）后进行模拟输入的采样和自诊断转换处理，然后开始 A/D 转换处理。

A/D 转换时间（tSCAN）包括 A/D 转换开始延迟时间（tD）、通道专用采样 & 保持电路采样时间（tSPLSH）、自诊断转换时间（tDIAG）、A/D 转换处理时间（tCONV）和 A/D 转换结束延迟时间（tED）。

A/D 转换处理时间（tCONV）是输入采样时间（tSPL）和逐次转换时间（tSAM）的总时间。

采样时间（tSPL）是为了向 A/D 转换器的采样 & 保持电路进行电荷充电的时间。在模拟输入的信号源阻抗高并且采样时间不够，或者 A/D 转换时钟（ADCLK）低速运行时，能通过 ADSSTR 寄存器调整采样时间。

逐次转换时间（tSAM）固定为 ADCLK 的 30 个状态。ADSSTR 寄存器的设定例子如表 28.8 所示。

A/D 转换时间如表 28.9 所示。

选择的通道数为 n 的 1 个周期扫描的 A/D 转换时间（tSCAN）如下所示：

1. AN000 ~ AN002、AN100 ~ AN102 的情况
  - 未使用通道专用采样 & 保持电路和自诊断时  
 $tSCAN = tD + (tCONV \times n) + tED$
  - 未使用通道专用采样 & 保持电路，但使用自诊断时  
 $tSCAN = tD + tDIAG + (tCONV \times n) + tED$
  - 使用通道专用采样 & 保持电路，但未使用自诊断时  
 $tSCAN = tD + tSPLSH + (tCONV \times n) + tED$
  - 使用通道专用采样 & 保持电路和自诊断时  
 $tSCAN = tD + tSPLSH + tDIAG + (tCONV \times n) + tED$
2. AN003、AN103 的情况
  - 未使用自诊断时  
 $tSCAN = tD + (tCONV \times n) + tED$
  - 使用自诊断时  
 $tSCAN = tD + tDIAG + (tCONV \times n) + tED$

单次模式和连续扫描模式的第 1 个周期和 1 个周期扫描模式相同。

从连续扫描的第 2 个周期起省略 1 个周期扫描模式的 tSCAN 到 tD 的时间。

表 28.8 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”	0.4μs（当 PCLK = ADCLK = 50MHz 时）
在模拟输入的信号源阻抗高并且采样时间不够时设定	“15h” ~ “FF”	例：“FFh” 5.1μs（当 PCLK = ADCLK = 50MHz 时）
当 ADCLK 小于 50MHz，将采样时间置为小于初始值时设定	“0Dh” ~ “13”	例：“10h” 0.4μs（当 PCLK = ADCLK = 40MHz 时）

注 1. 必须设定为采样时间 ≥ 0.4μs。采样时间通过以下表达式表示：

$$\text{采样时间}(\mu\text{s}) = \frac{\text{ADSSTR寄存器的设定值}}{\text{ADCLK}(\text{MHz})}$$

表 28.9 A/D 转换时间

项目		符号		ADTRGn# (外部触发)	MTU3、GPT	通过软件开始	单位
扫描转换开始延迟时间 (注2)		tD		4 个 PCLK+ 3 个 ADCLK (注1)	2 个 PCLK+ 3 个 ADCLK	2 个 PCLK+ 3 个 ADCLK	周期
通道专用独立采样 & 保持 电路处理时间	采样时间	tSPLSH	tSH	20 个 ADCLK			
	采样 -A/D 转 换等待时间		tW	10 个 ADCLK			
自诊断转换处理时间	采样时间	tDIAG	tSPL	ADSSTR 寄存器的设定值 (初始值 20) ×ADCLK			
	逐次转换时间		tSAM	30 个 ADCLK			
A/D 转换处理时间	采样时间	tCONV	tSPL	ADSSTR 寄存器的设定值 (初始值 20) ×ADCLK			
	逐次转换时间		tSAM	30 个 ADCLK			
扫描转换结束延迟时间 (注3)		tED		1 个 PCLK+5 个 ADCLK			

PCLK: 模块时钟, ADCLK: A/D 转换时钟

注 1. 有关外部触发输入时序, 请参照“33.3.3 内部外围模块的时序”。

注 2. 这是从软件写或者触发输入到开始 A/D 转换为止的最长时间。

注 3. 这是从 A/D 转换结束开始到发生 A/D 转换结束中断为止的时间。

#### 28.3.4 ADDRn、ADRD 寄存器的自动清除功能的使用例子 (n=0A、0B、1~3)

通过将 ADCER.ACE 位置“1”, 能在通过 CPU 和 DTC 读 A/D 数据寄存器 n (ADDRn 和 ADRD) 时自动将 ADDRn 寄存器和 ADRD 寄存器置“0000h”。

通过使用自动清除功能, 能检测到 ADDRn 寄存器和 ADRD 寄存器未更新的故障。

ADDRn 寄存器的自动清除功能无效和有效时的例子如下所示:

当 ADCER.ACE 位为“0” (禁止自动清除) 时, 如果因某种原因而没有将 A/D 转换结果 (0222h) 写到 ADDRn 寄存器, 旧的数据 (0111h) 就为 ADDRn 寄存器的值。如果使用 A/D 转换结束中断将此 ADDRn 寄存器的值读到通用寄存器, 就能将旧的数据 (0111h) 保存到通用寄存器等。但是, 在进行未更新的检查时, 需要边将旧的数据逐个保存到 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) 时, 在通过 CPU 和 DTC 读 ADDRn 为“0111h”时, ADDRn 寄存器自动变为“0000h”。此后, 如果因某种原因而无法将 A/D 转换结果“0222h”传送到 ADDRn 寄存器, 将被清除的数据 (0000h) 留作 ADDRn 寄存器的值。此时, 如果使用 A/D 转换结束中断将此 ADDRn 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值为“0000h”, 就能判断 ADDRn 寄存器有未更新的故障。

当选择 ADDRn 寄存器的自动清除时, 就在读取时清除转换数据。

### 28.3.5 双数据寄存器的运行 (仅 ADDR0 寄存器)

当选择 AN000/AN100, 并且通过 ADSTRGR.ADSTRSn[4:0] 位将开始源设定为“MTU3 的 TRGnAN 或者 TRGnBN (n=4、7)” (将 ADSTRGR.ADSTRSn[4:0] 位置“01011b”、“01111b”) 以及“GPT 的 GTADTRAnN 或者 GTADTRBnN (n=0 ~ 3)” (将 ADSTRGR.ADSTRSn[4:0] 位置“11001b ~ 11100b”) 后开始 A/D 转换时, AN000/AN100 的转换结果通过开始触发保存到不同的转换结果寄存器。在通过 MTU3 的 TRGnAN 和 GPT 的 GTADTRAnN 开始转换时, AN000/AN100 的转换结果保存到 ADDR0A 寄存器。在通过 MTU3 的 TRGnBN 和 GPT 的 GTADTRBnN 开始转换时, AN000/AN100 的转换结果保存到 ADDR0B 寄存器。通过设定 ADCER.ADIEW 位, 能选择是在由后一个触发引起的转换结束后, 还是分别在由各触发引起的转换结束后发生转换结束中断。选择 AN000 并且通道 MTU3 的定时器触发设定 TRG4AN 或者 TRG4BN, 使单次模式运行时的运行例子如图 28.8 所示。

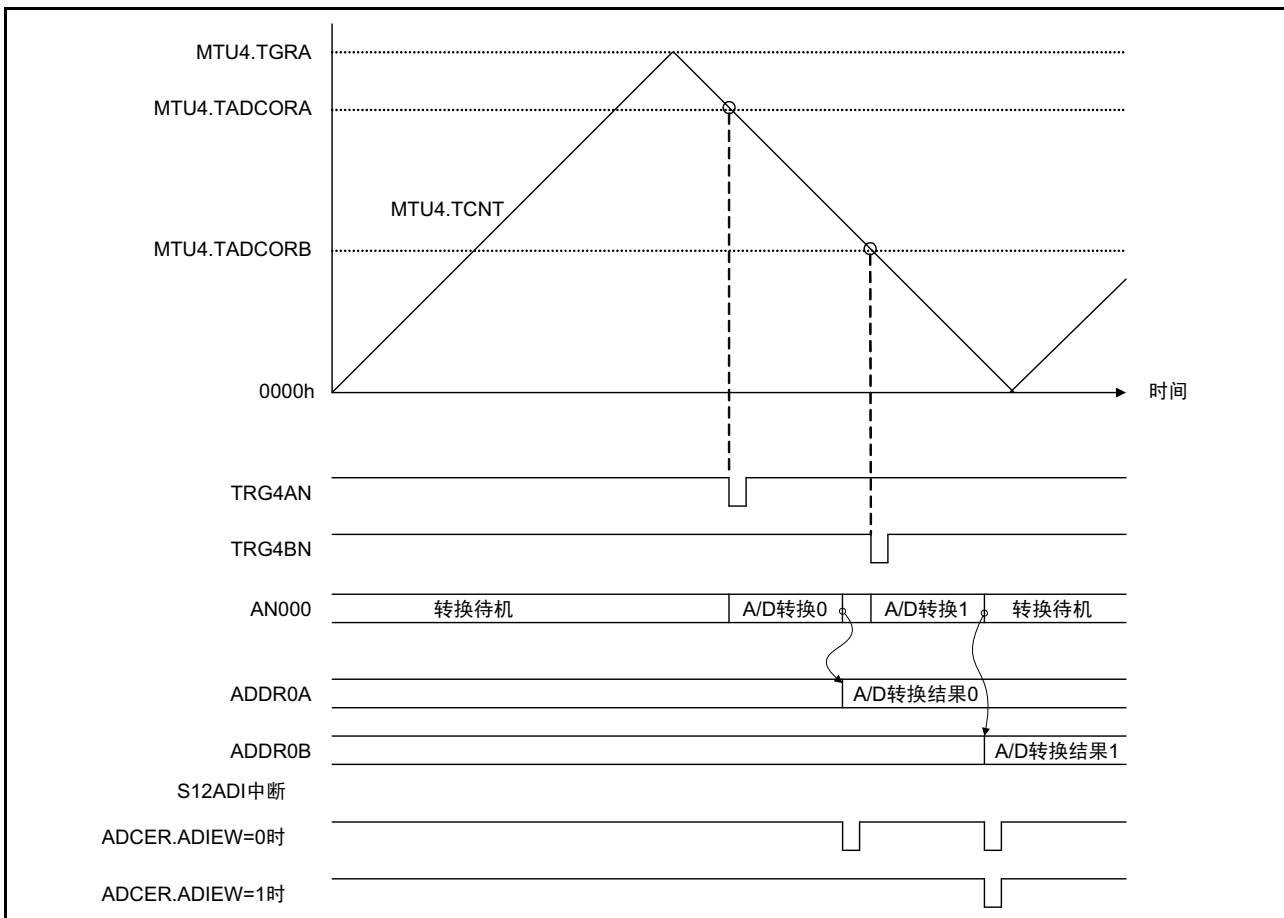


图 28.8 双数据寄存器的运行例子

### 28.3.6 可编程增益放大器

可编程增益放大器装载于 AN000 ~ AN002 引脚和 AN100 ~ AN102 引脚。通过 ADPG.PGnGAIN[3:0] 位 (n=000 ~ 002、100 ~ 102) 选择增益, 通过 ADANS.PGnEN 位和 ADANS.PGnSEL 位选择使用的放大器。

如果要使用可编程增益放大器, 必须将 ADCER.SHBY 位设定为“0” (使用采样 & 保持电路)。

### 28.3.7 比较器

比较器装载于 AN000 ~ AN002 引脚和 AN100 ~ AN102 引脚。作为比较器的运行模式能选择低电平比较器（检测比低电平侧基准电压还要低的电压被外加到输入）、高电平比较器（检测比高电平侧基准电压还要高的电压被外加到输入）和窗口比较器（检测在低电平侧基准电压到高电平侧基准电压范围外的电压被外加到输入）。检测的输入电压能选择可编程增益放大器的放大前或者放大后的电压。基准电压能选择是从引脚输入（低电平侧：AN003/CVREFL、高电平侧：AN103/CVREFH）还是使用内部电压（ $1/8 \times AVCC0 \sim 7/8 \times AVCC0$ ）。另外，比较器检测信号内置噪声消除电路，通过  $1/PCLK \sim 128/PCLK$  进行 16 次采样，只在全部检测时将标志置位，并且产生中断请求（CMPI）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）。

比较器的设定例子如下所示。

1. 通过 ADCMPMD1.CSELn 位（n=0、1）设定是检测可编程增益放大器放大前的电压，还是检测可编程增益放大器放大后的电压。
  2. 通过 ADCMPMD1.VSELLn 位（n=0、1）和 ADCMPMD1.VSELHn（n=0、1）位设定基准电压是从引脚输入还是使用内部电压。如果选择内部电压，就通过 ADCMPMD1.REFL[2:0] 位设定低电平侧基准电压，通过 ADCMPMD1.REFH[2:0] 位设定高电平侧基准电压。
  3. 通过 ADCMPNRn.CmNR[3:0] 位（n=0、1，m=000~002、100~102）给各比较器设定用于比较器检测结果的噪声消除电路。
  4. 通过 ADCMPSEL.SELm 位（m=000~002、100~102）给各比较器设定是否通过比较器检测产生中断请求（CMPI）或者 POE 请求。如果产生中断请求（CMPI）或者 POE 请求，就通过 ADCMPSEL.IE 位允许中断请求（CMPI），通过 ADCMPSEL.POE 位允许 POE 请求。
  5. 通过 ADCMPMD0.CENm[1:0] 位（m=000~002、100~102）设定使用的比较器以及运行模式。
- 比较器的运行例子如图 28.9 所示。

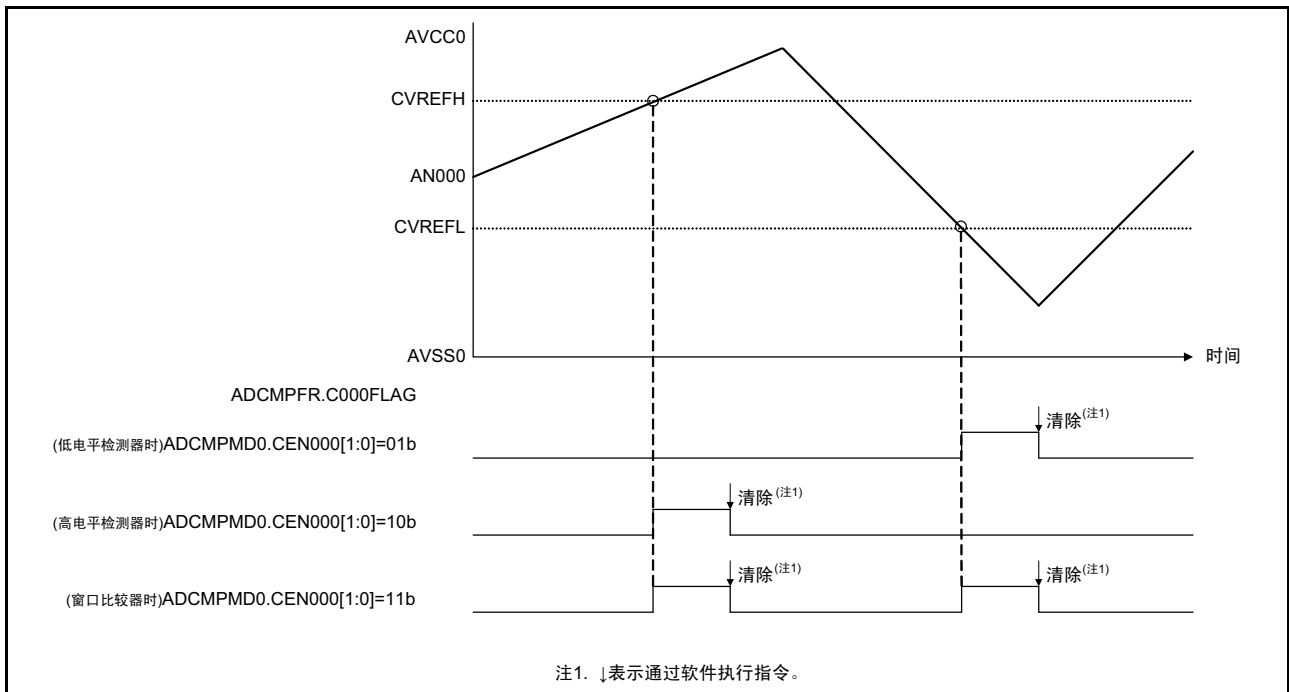


图 28.9 比较器的运行例子（从引脚输入基准电压时）

### 28.3.8 通过外部触发开始的 A/D 转换

能通过输入外部触发来开始 A/D 转换。在使用外部触发开始 A/D 转换时，通过 PFAADC 寄存器设定引脚功能，在将 A/D 开始触发选择寄存器 (ADSTRGR) 置“00h”并且给 ADTRGn# 引脚输入高电平后，将 ADCSR.TRGE 位置“1”并且将 ADCSR.EXTRG 位置“1”。

外部触发输入时序如图 28.10 所示。

引脚功能的设定请参照“15. I/O 端口”。

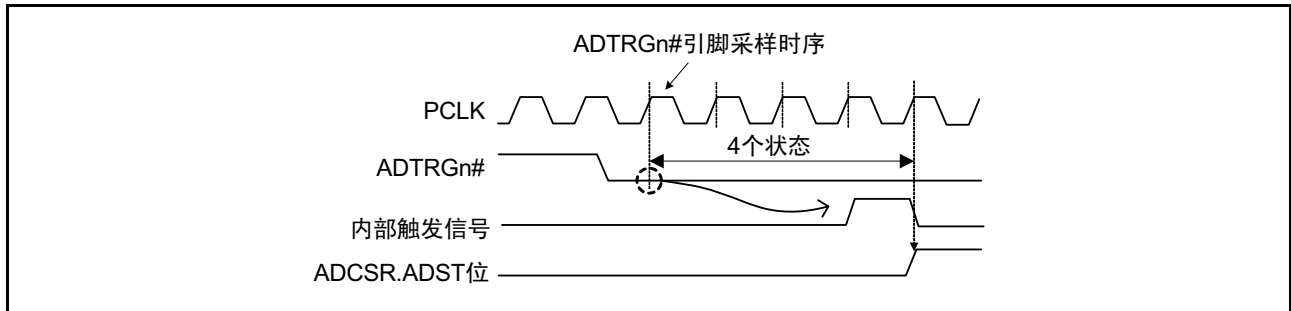


图 28.10 外部触发输入时序

### 28.3.9 通过外围模块的触发开始的 A/D 转换

能通过 MTU3 或者 GPT 的定时器触发来开始 A/D 转换。在通过定时器触发开始 A/D 转换时，将 ADCSR.TRGE 位置“1”，将 ADCSR.EXTRG 位置“0”并且通过 ADSTRGR.ADSTRS[4:0] 位设定对应的 A/D 转换开始源。

## 28.4 中断源和 DTC 传送请求

### 28.4.1 A/D 转换的各 A/D 转换结束时的中断请求

A/D 转换器能向 CPU 产生 A/D 转换结束中断请求 (S12ADI)。

如果将 ADCSR.ADIE 位置“1”，就能允许 S12ADI 中断；如果置“0”，就能禁止 S12ADI 中断。

能在发生 S12ADI 中断时启动 DTC。如果通过 DTC 读由 S12ADI 中断转换的数据，就能实现连续转换而没有软件负担。

DTC 的设定请参照“14. 数据传送控制器 (DTC)”。

### 28.4.2 比较器检测时的中断请求

比较器能向 CPU 产生比较器检测中断请求 (CMPI)。

如果在将 ADCMPSEL.SEL000 ~ 002、100 ~ 102 位中的任意位置“1”后，将 ADCMPSEL.IE 位置“1”，就能允许 CMPI 中断；如果值“0”，就能禁止 CMPI 中断。但是，忽视在比较器检测标志寄存器

(ADCMPFR) 的比较器检测标志为“1”的状态下产生的中断请求，因此为了能重新中断，必须将对应状态标志置“0”。



## 28.5 使用时的注意事项

### 28.5.1 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 A/D 转换器的运行，初始值为停止 A/D 转换器的运行。能通过解除模块停止状态，使寄存器变为可存取的状态。详细内容请参照“9. 低功耗功能”。

### 28.5.2 重新开始 A/D 转换时的注意事项

如果通过将 ADCSR.ADST 位置“0”来停止 A/D 转换，就需要 2 个 ADCLK 时钟的时间来停止 A/D 转换器的模拟电路。在将 ADCSR.ADST 位置“1”后开始 A/D 转换时，需要 3 个 ADCLK 时钟的时间。

必须在实施自诊断后才能重新开始 A/D 转换。

### 28.5.3 停止 A/D 转换时的注意事项

在选择外部触发或者定时器为 A/D 转换开始条件时，为了停止 A/D 转换，必须在将 ADCSR.TRGE 位置“0”并且在选择软件触发为 A/D 转换开始条件后，将 ADCSR.ADST 位置“0”（停止 A/D 转换）。

### 28.5.4 向低功耗状态转移时的注意事项

必须在停止 A/D 转换的状态下向模块停止模式或者软件待机模式转移。在停止 A/D 转换时，需要确保从 ADCSR.ADST 位被置“0”后到 A/D 转换器的模拟电路停止前的时间。为了确保此时间，必须按照以下步骤进行设定。

必须在将 ADCSR.TRGE 位置“0”（软件触发）并且将 ADCSR.ADST 位置“0”后，将 ADCSR.CKS[1:0] 位置“11b”（PCLK）。此后，必须在确认 A/D 转换已经停止后（停止前至少需要 6 个 PCLK 的时间）转移到模块停止状态或者软件待机模式。

另外，在向模块停止模式或者软件待机模式转移时，A/D 转换器的一部分处于运行待机状态。如果能让 A/D 转换器完全处于待机状态，就必须将 MSTPCRA.MSTPA24 位置“1”。在此状态下解除模块停止模式、软件待机模式或者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。

### 28.5.5 有关容许信号源高阻抗

为了实行  $1.0\mu\text{s}$  的高速转换，对于信号源阻抗不超过  $3.0\text{k}\Omega$  的输入信号，RX62T 群的模拟输入能保证转换精度。在单次模式中进行转换并且外接大电容的情况下，因为输入负载实际上就只有  $10\text{k}\Omega$  的内部输入电阻，因此信号源阻抗可忽略不计。但是，由于形成低通滤波器，所以有可能无法迎合急剧变化的模拟信号（例如  $5\text{mV/s}$  以上）（图 28.11）。在转换高速模拟信号或者在扫描模式模式中进行转换时，必须插入输出阻抗低的缓冲放大器。

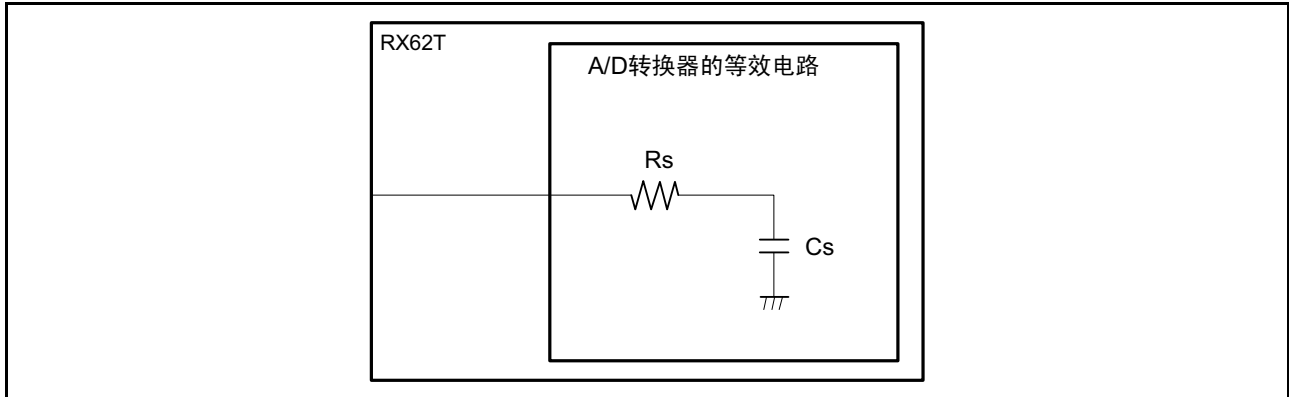


图 28.11 模拟输入引脚的内部等效电路

表 28.10 模拟引脚的规格

项目	min	max	单位
容许信号源高阻抗	—	3.0	$\text{k}\Omega$
引脚的内部等效电路	Rs	10.0	$\text{k}\Omega$
	Cs	8.0	$\text{pF}$

### 28.5.6 对绝对精度的影响

由于附加电容接会导致与 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。

另外，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。

### 28.5.7 模拟电源引脚等的设定范围

当超过以下所示电压的设定范围并且使用 LSI 时，可能对 LSI 的可靠性造成不良影响。

- 模拟输入电压的设定范围  
给模拟输入引脚  $\text{ANn}$  外加的电压必须在  $\text{VREFL0} \leq \text{VAN} \leq \text{VREFH0}$  的范围内。
- 各电源引脚（AVCC0-AVSS0、AVCC-AVSS、VCC-VSS）的关系  
各电源引脚（AVCC0-AVSS0、AVCC-AVSS、VCC-VSS）的关系必须为  $\text{VCC} \leq \text{AVCC0} = \text{AVCC}$  并且  $\text{AVSS0} = \text{AVSS} = \text{VSS}$ 。在不使用 12 位 A/D 转换器时必须为  $\text{VCC} = \text{AVCC0} = \text{AVCC}$ 、 $\text{AVSS0} = \text{AVSS} = \text{VSS}$ 。
- VREFH0、VREFL0 的设定范围  
VREFH0、VREFL0 引脚的设定范围必须为  $\text{VREFH0} \leq \text{AVCC0}$  并且  $\text{VREFL0} = \text{AVSS0}$ 。在不使用 A/D 转换器时必须为  $\text{VREFH0} = \text{AVCC0}$ 、 $\text{VREFL0} = \text{AVSS0}$ 。

### 28.5.8 设计电路板时的注意事项

在设计电路板时，必须尽量使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对 A/D 转换值造成不良影响。必须通过模拟接地 (AVSS0) 将模拟输入引脚 (AN000 ~ AN003、AN100 ~ AN103)、模拟基准电源 (VREFH0、VREFL0) 和模拟电源电压 (AVCC0) 与数字电路分开。另外，模拟接地 (AVSS0) 必须与电路板上的稳定接地 (VSS) 进行单点连接。如图 28.12 所示，必须连接 0.1μF 电容器，以在各电源间形成最短的闭环路。

### 28.5.9 噪声对策的注意事项

如图 28.12 所示，为了防止过大电涌等异常电压对模拟输入引脚 (AN000 ~ AN003、AN100 ~ AN103) 造成破坏，必须在 AVCC0-AVSS0 之间和 VREFH0-VREFL0 之间连接电容，并且以模拟输入引脚 (AN000 ~ AN003、AN100 ~ AN103) 为基准连接保护电路。

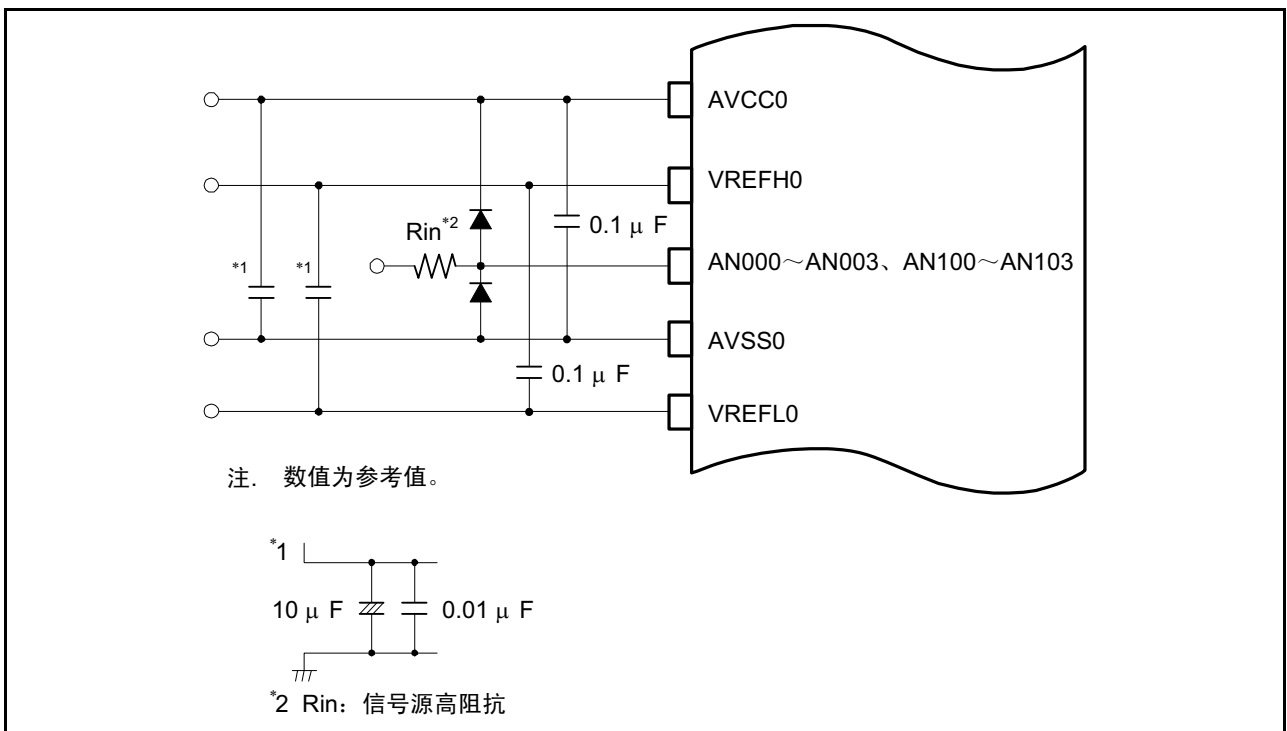


图 28.12 模拟输入保护电路的例子

## 29. 10 位 A/D 转换器 (ADA)

### 29.1 概要

RX62T 群内置 1 个单元的逐次逼近方式的 10 位 A/D 转换器，各单元最多能选择 12 个通道的模拟输入。

A/D 转换器的运行模式有对 1 个通道的模拟输入只进行 1 次转换的单次模式，以及对最多 12 个通道的模拟输入依次进行连续转换的扫描模式。

A/D 转换器的规格以及功能一览表分别如表 29.1 和表 29.2 所示，框图如图 29.1 所示。

表 29.1 A/D 转换器的规格

项目	规格
单元数	1 个单元
输入通道	12 个通道
A/D 转换方式	逐次逼近方式
分辨率	10 位
转换时间	每个通道 1.0 $\mu$ s (当 A/D 转换时钟 ADCLK=50MHz、AVCC=4.0 ~ 5.5V 时) 每个通道 2.0 $\mu$ s (当 A/D 转换时钟 ADCLK=25MHz、AVCC=3.0 ~ 3.6V 时)
A/D 转换时钟	4 种: PCLK、PCLK/2、PCLK/4、PCLK/8
运行模式	<ul style="list-style-type: none"> <li>• 单次模式: 对 1 个通道的模拟输入只进行 1 次转换。</li> <li>• 扫描模式 连续扫描模式: 重复转换最多 12 个通道的模拟输入。 1 个周期扫描模式: 对最多 12 个通道的模拟输入只进行 1 个周期的转换。</li> </ul>
A/D 转换开始条件	<ul style="list-style-type: none"> <li>• 软件触发</li> <li>• 多功能定时器脉冲单元 3 (MTU3) 或者通用 PWM 定时器 (GPT) 触发</li> <li>• 外部触发 能通过 ADTRG# 引脚开始 A/D 转换。</li> </ul>
功能	<ul style="list-style-type: none"> <li>• 采样 &amp; 保持功能</li> <li>• 采样状态数的可变功能</li> <li>• A/D 转换器的自诊断功能</li> </ul>
中断源	<ul style="list-style-type: none"> <li>• 在 A/D 转换结束时产生 A/D 中断请求 (ADI0)。</li> <li>• 能通过 ADI 中断来启动数据传送控制器 (DTC)。</li> </ul>
低功耗功能	设定为模块停止状态。

表 29.2 A/D 转换器的功能一览表

项目			功能		
模拟输入通道			AN0 ~ AN11		
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	ADTRG#		
MTU3 的触发		MTU0.TGRA 的比较匹配 / 输入捕捉	TRGA0N		
		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 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 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 的比较匹配以及 MTU4.TADCORB 和 MTU4.TCNT 的比较匹配 (使用中中断减少功能 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 的比较匹配以及 MTU7.TADCORB 和 MTU7.TCNT 的比较匹配 (使用中中断减少功能 2 时)	TRG7ABN		
		GPT 的触发		GPT0.GTADTRA 的比较匹配	GTADTRA0
				GPT0.GTADTRB 的比较匹配	GTADTRB0
				GPT1.GTADTRA 的比较匹配	GTADTRA1
				GPT1.GTADTRB 的比较匹配	GTADTRB1
GPT2.GTADTRA 的比较匹配	GTADTRA2				
GPT2.GTADTRB 的比较匹配	GTADTRB2				
GPT3.GTADTRA 的比较匹配	GTADTRA3				
GPT3.GTADTRB 的比较匹配	GTADTRB3				
GPT0.GTADTRA 的比较匹配或者 GPT0.GTADTRB 的比较匹配	GTADTRA0 或者 GTADTRB0				
GPT1.GTADTRA 的比较匹配或者 GPT1.GTADTRB 的比较匹配	GTADTRA1 或者 GTADTRB1				
GPT2.GTADTRA 的比较匹配或者 GPT2.GTADTRB 的比较匹配	GTADTRA2 或者 GTADTRB2				
GPT3.GTADTRA 的比较匹配或者 GPT3.GTADTRB 的比较匹配	GTADTRA3 或者 GTADTRB3				
中断			ADI 中断		
模块停止功能的设定 (注 1)			MSTPCRA. MSTPA23 位		

注 1. 详细内容请参照“9. 低功耗功能”。

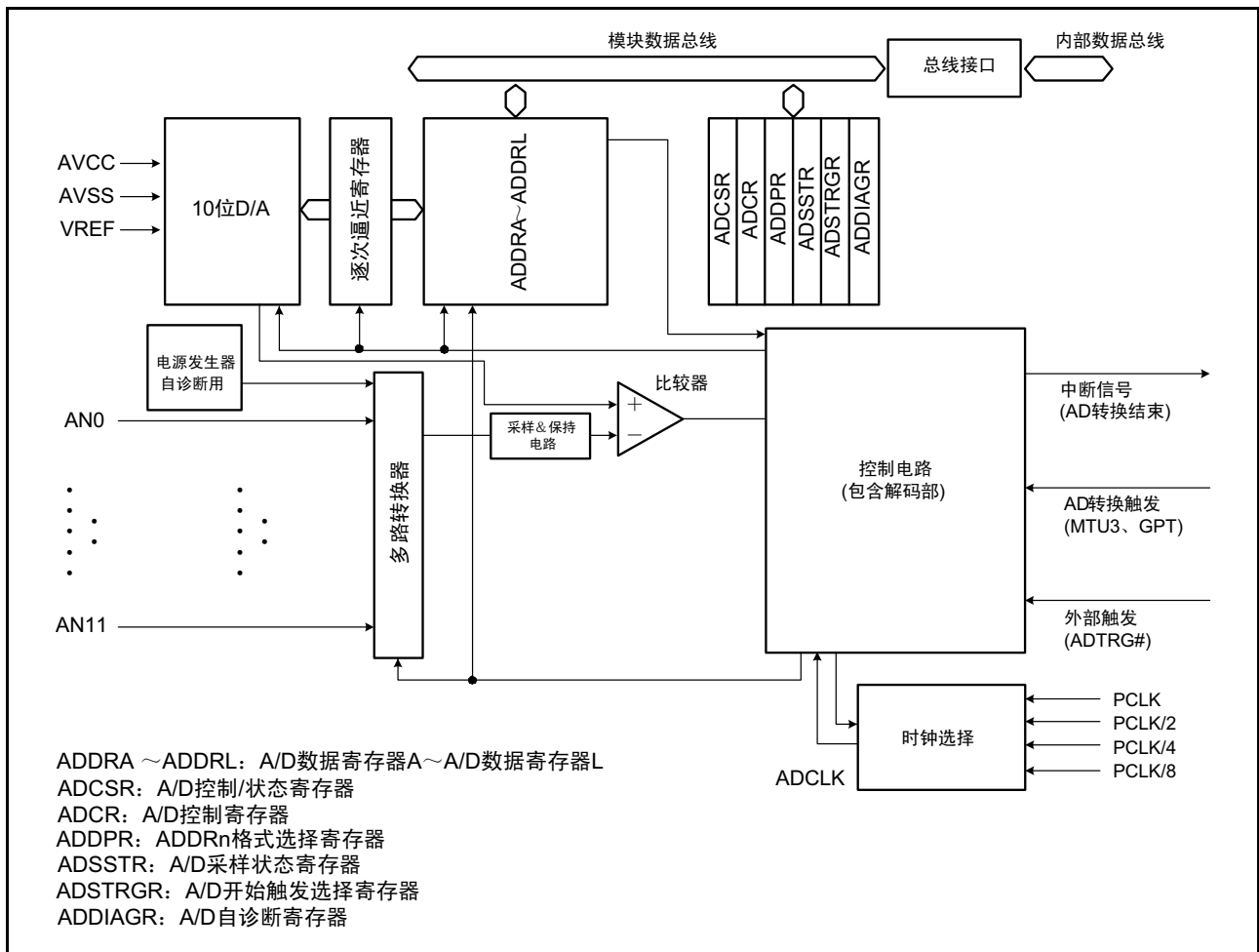


图 29.1 A/D 转换器的框图

A/D 转换器使用的输入引脚如表 29.3 所示。

表 29.3 A/D 转换器的输入引脚

模块符号	引脚名	输入	功能
AD0	AN0 ~ AN11	输入	模拟输入引脚
	ADTRG#	输入	用于开始 A/D 转换的外部触发输入引脚
	AVCC	输入	模拟电路的电源引脚
	AVSS	输入	模拟电路的接地引脚
	VREF	输入	A/D 转换器的基准电源引脚

## 29.2 寄存器说明

A/D 转换器的寄存器一览表如表 29.4 所示。

表 29.4 A/D 转换器的寄存器一览表

模块符号	寄存器名	寄存器符号	复位后的值	地址	存取长度
AD0	A/D 数据寄存器 A	ADDRA	0000h	0008 8040h	16
	A/D 数据寄存器 B	ADDRB	0000h	0008 8042h	16
	A/D 数据寄存器 C	ADDRC	0000h	0008 8044h	16
	A/D 数据寄存器 D	ADDRD	0000h	0008 8046h	16
	A/D 数据寄存器 E	ADDRE	0000h	0008 8048h	16
	A/D 数据寄存器 F	ADDRF	0000h	0008 804Ah	16
	A/D 数据寄存器 G	ADDRG	0000h	0008 804Ch	16
	A/D 数据寄存器 H	ADDRH	0000h	0008 804Eh	16
	A/D 数据寄存器 I	ADDRI	0000h	0008 8060h	16
	A/D 数据寄存器 J	ADDRJ	0000h	0008 8062h	16
	A/D 数据寄存器 K	ADDRK	0000h	0008 8064h	16
	A/D 数据寄存器 L	ADDRL	0000h	0008 8066h	16
	A/D 控制 / 状态寄存器	ADCSR	x0h	0008 8050h	8
	A/D 控制寄存器	ADCR	00h	0008 8051h	8
	A/D 采样状态寄存器	ADSSTR	19h	0008 805Bh	8
	A/D 自诊断寄存器	ADDIAGR	00h	0008 805Dh	8
	A/D 开始触发选择寄存器	ADSTRGR	00h	0008 8070h	8
	ADDRn 格式选择寄存器	ADDPR	00h	0008 8072h	8

### 29.2.1 A/D 数据寄存器 n (ADDRn) (n=A ~ L)

地址 ADDRn 0008 8040h、ADDRB 0008 8042h、ADDRC 0008 8044h、ADDRD 0008 8046h、ADDRE 0008 8048h、ADDRF 0008 804Ah、  
ADDRG 0008 804Ch、ADDRH 0008 804Eh、ADDRI 0008 8060h、ADDRJ 0008 8062h、ADDRK 0008 8064h、ADDRL 0008 8066h

- ADDPR.DPSEL 位 =1、ADDPR.DPPRC 位 =0 (数据为 10 位精度、并且往 MSB 靠紧)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
复位后的值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b5-b0	—	保留位	读取值为“0”，写操作无效。	R
b15-b6	AD9 ~ AD0	转换值 [9:0]	10 位 A/D 转换值	R

- ADDPR.DPSEL 位 =1、ADDPR.DPPRC 位 =1 (数据为 8 位精度、并且往 MSB 靠紧)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	—	—	—	—	—	—	—	—
复位后的值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b7-b0	—	保留位	读取值为“0”，写操作无效。	R
b15-b8	AD9 ~ AD2	转换值 [9:2]	10 位 A/D 转换值的高 8 位	R

- ADDPR.DPSEL 位 =0、ADDPR.DPPRC 位 =0 (数据为 10 位精度、并且往 LSB 靠紧)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
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
b9-b0	AD9 ~ AD0	转换值 [9:0]	10 位 A/D 转换值	R
b15-b10	—	保留位	读取值为“0”，写操作无效。	R

- ADDPR.DPSEL 位 =0、ADDPR.DPPRC 位 =1 (数据为 8 位精度、并且往 LSB 靠紧)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
复位后的值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b7-b0	AD9 ~ AD2	转换值 [9:2]	10 位 A/D 转换值的高 8 位	R
b15-b8	—	保留位	读取值为“0”，写操作无效。	R



ADDRn 寄存器是保存 A/D 转换结果的 16 位只读寄存器。

模拟输入通道和 ADDRn 寄存器的对应如表 29.5 所示。

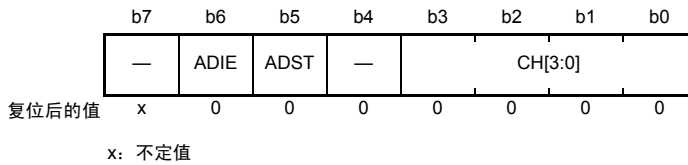
能通过设定 ADDPR.DPSEL 位, 更改 10 位数据的排列。另外, 能通过设定 ADDPR.DPPRC 位, 设定保存数据的精度。

表 29.5 模拟输入通道和 ADDRn 寄存器的对应

模拟输入通道	ADDRn 寄存器
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH
AN8	ADDRI
AN9	ADDRJ
AN10	ADDRK
AN11	ADDRL

## 29.2.2 A/D 控制 / 状态寄存器 (ADCSR)

地址 0008 8050h



位	符号	位名	功能	R/W																													
b3-b0	CH[3:0]	通道选择位 (注1)	单次模式 (ADCR.MODE[1:0] 位为 “00b”)  <table style="font-size: small; border: none;"> <tr><td style="padding-right: 10px;">b3</td><td>b0</td></tr> <tr><td>0 0 0 0:</td><td>AN0</td></tr> <tr><td>0 0 0 1:</td><td>AN1</td></tr> <tr><td>0 0 1 0:</td><td>AN2</td></tr> <tr><td>0 0 1 1:</td><td>AN3</td></tr> <tr><td style="text-align: center;">:</td><td style="text-align: center;">:</td></tr> <tr><td>1 0 1 1:</td><td>AN11</td></tr> </table> 上述以外: 不能设定	b3	b0	0 0 0 0:	AN0	0 0 0 1:	AN1	0 0 1 0:	AN2	0 0 1 1:	AN3	:	:	1 0 1 1:	AN11	扫描模式 (ADCR.MODE[1:0] 位为 “10b”或者“11b”)  <table style="font-size: small; border: none;"> <tr><td style="padding-right: 10px;">b3</td><td>b0</td></tr> <tr><td>0 0 0 0:</td><td>AN0</td></tr> <tr><td>0 0 0 1:</td><td>AN0、AN1</td></tr> <tr><td>0 0 1 0:</td><td>AN0 ~ AN2</td></tr> <tr><td>0 0 1 1:</td><td>AN0 ~ AN3</td></tr> <tr><td style="text-align: center;">:</td><td style="text-align: center;">:</td></tr> <tr><td>1 0 1 1:</td><td>AN0 ~ AN11</td></tr> </table> 上述以外: 不能设定	b3	b0	0 0 0 0:	AN0	0 0 0 1:	AN0、AN1	0 0 1 0:	AN0 ~ AN2	0 0 1 1:	AN0 ~ AN3	:	:	1 0 1 1:	AN0 ~ AN11	R/W
b3	b0																																
0 0 0 0:	AN0																																
0 0 0 1:	AN1																																
0 0 1 0:	AN2																																
0 0 1 1:	AN3																																
:	:																																
1 0 1 1:	AN11																																
b3	b0																																
0 0 0 0:	AN0																																
0 0 0 1:	AN0、AN1																																
0 0 1 0:	AN0 ~ AN2																																
0 0 1 1:	AN0 ~ AN3																																
:	:																																
1 0 1 1:	AN0 ~ AN11																																
b4	—	保留位	读写值都为“0”。	R/W																													
b5	ADST	A/D 开始位	0: 停止 A/D 转换 1: 开始 A/D 转换	R/W																													
b6	ADIE	A/D 中断允许位	0: 禁止由 A/D 转换结束引起的 ADI 中断 1: 允许由 A/D 转换结束引起的 ADI 中断	R/W																													
b7	—	保留位	读取值为不定值, 只能写“1”。	R/W																													

注 1. 必须将用作模拟输入引脚的 PORTn.DDR.Bi 位置“0” (输入端口) 并且将 PORTn.ICR.Bi 位置“0” (对应引脚的输入缓冲器无效, 输入信号固定为 High 电平)。详细内容请参照“15. I/O 端口”。(n=6, 5, i=5 ~ 0)

ADCSR 寄存器是控制 A/D 转换的寄存器。

必须在 ADST 位为“0”时更改 CH[3:0] 位和 ADIE 位的设定。

## CH[3:0] 位 (通道选择位)

这些位选择进行 A/D 转换的模拟输入通道。

- 单次模式 (ADCR.MODE[1:0] 位为“00b”)
  - 选择 1 个进行 A/D 转换的模拟输入通道。
- 扫描模式 (ADCR.MODE[1:0] 位为“10b”或者“11b”)
  - 最多选择 12 个进行 A/D 转换的模拟输入通道。

**ADST 位 (A/D 开始位)**

此位控制 A/D 转换的开始或者停止。

必须在将 ADST 位置“1”前设定 A/D 转换时钟和运行模式。

[ 为“1”的条件 ]

- 通过软件写“1”时
- 检测到 ADSTRGR.ADSTRS[4:0] 位所选的触发时

[ 为“0”的条件 ]

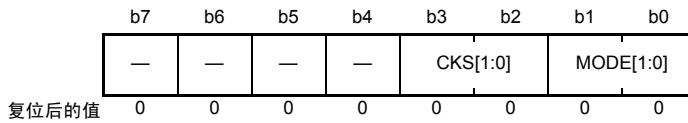
- 通过软件写“0”时
- 在单次模式中 A/D 转换结束时
- 在 1 个周期扫描模式中所选通道的 A/D 转换全部结束时

**ADIE 位 (A/D 中断允许位)**

此位允许或者禁止由 A/D 转换结束引起的 ADI 中断。

### 29.2.3 A/D 控制寄存器 (ADCR)

地址 0008 8051h



位	符号	位名	功能	R/W
b1-b0	MODE[1:0]	运行模式选择位	b1 b0 0 0: 单次模式 0 1: 不能设定 1 0: 连续扫描模式 1 1: 1 个周期扫描模式	R/W
b3-b2	CKS[1:0]	时钟选择位	b3 b2 0 0: PCLK/8 0 1: PCLK/4 1 0: PCLK/2 1 1: PCLK	R/W
b7-b4	—	保留位	读写值都为“0”。	R/W

ADCR 寄存器设定 A/D 转换的运行模式和 A/D 转换时钟。

必须在 ADCSR.ADST 位为“0”时设定 ADCR 寄存器。

#### MODE[1:0] 位 (运行模式选择位)

这些位选择 A/D 转换的运行模式。

#### CKS[1:0] 位 (时钟选择位)

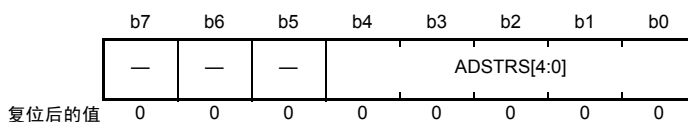
这些位设定决定 A/D 转换时间的 A/D 转换时钟 (ADCLK) 的频率。

必须将 ADCLK 的频率设定为大于等于 4MHz。另外, 在 AVCC=3.0 ~ 3.6V 下使用时, ADCLK 的频率必须设定为小于等于 25MHz。

详细内容请参照“29.3.3 输入采样和 A/D 转换时间”。

### 29.2.4 A/D 开始触发选择寄存器

地址 0008 8070h



位	符号	位名	功能	R/W
b4-b0	ADSTRS[4:0]	A/D 开始触发选择位	通过 bit4 ~ bit0 的组合选择外围模块的 A/D 转换启动源。启动源和设定值的关系如表 29.6 所示。	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W

表 29.6 A/D 启动源选择一览表

启动源	源名	开始条件	ADSTRS[4]	ADSTRS[3]	ADSTRS[2]	ADSTRS[1]	ADSTRS[0]
软件	ADST	将 ADCSR.ADST 置位	—	—	—	—	—
外部 引脚	ADTRG# (注1)	外部触发	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 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 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 的比较匹配和 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 的比较匹配和 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	GTADTRA0	GPT0.GTADTRA 的比较匹配	1	0	0	0	1
	GTADTRB0	GPT0.GTADTRB 的比较匹配	1	0	0	1	0
	GTADTRA1	GPT1.GTADTRA 的比较匹配	1	0	0	1	1
	GTADTRB1	GPT1.GTADTRB 的比较匹配	1	0	1	0	0
	GTADTRA2	GPT2.GTADTRA 的比较匹配	1	0	1	0	1
	GTADTRB2	GPT2.GTADTRB 的比较匹配	1	0	1	1	0
	GTADTRA3	GPT3.GTADTRA 的比较匹配	1	0	1	1	1
	GTADTRB3	GPT3.GTADTRB 的比较匹配	1	1	0	0	0
	GTADTRA0 或者 GTADTRB0	GPT0.GTADTRA 的比较匹配或者 GPT0.GTADTRB 的比较匹配	1	1	0	0	1
	GTADTRA1 或者 GTADTRB1	GPT1.GTADTRA 的比较匹配或者 GPT1.GTADTRB 的比较匹配	1	1	0	1	0
	GTADTRA2 或者 GTADTRB2	GPT2.GTADTRA 的比较匹配或者 GPT2.GTADTRB 的比较匹配	1	1	0	1	1
	GTADTRA3 或者 GTADTRB3	GPT3.GTADTRA 的比较匹配或者 GPT3.GTADTRB 的比较匹配	1	1	1	0	0

注 1. 给 ADTRG# 设定开始 A/D 转换的触发时, 必须将对应引脚的 PORTn.DDR.Bi 位置“0” (输入端口) 并且将 PORTn.ICR.Bi 位置“1” (对应引脚的输入缓冲器有效)。详细内容请参照“15. I/O 端口”。(n=2, i=2)

## 29.2.5 ADDRn 格式选择寄存器 (ADDPR) (n=A ~ L)

地址 0008 8072h

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

位	符号	位名	功能	R/W
b0	DPPRC	位精度指定位	0: 以 10 位精度保存到 A/D 数据寄存器 1: 以 8 位精度保存到 A/D 数据寄存器	R/W
b6-b1	—	保留位	读写值都为“0”。	R/W
b7	DPSEL	ADDRn 格式选择位	0: 数据往 LSB 靠紧 1: 数据往 MSB 靠紧	R/W

ADDPR 寄存器选择 A/D 数据寄存器的数据排列。

## DPPRC 位 (位精度指定位)

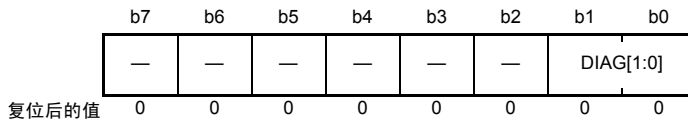
此位选择 A/D 转换结果是以 8 位精度还是 10 位精度保存到 A/D 数据寄存器。

## DPSEL 位 (ADDRn 格式选择位)

此位选择 A/D 数据寄存器 n (ADDRn) 的数据是往 LSB 靠紧还是往 MSB 靠紧。

## 29.2.6 A/D 自诊断寄存器 (ADDIAGR)

地址 0008 805Dh



位	符号	位名	功能	R/W
b1-b0	DIAG[1:0]	自诊断位	b1 b0 0 0: 普通运行 0 1: 使用 VREF×0 的电压进行自诊断 1 0: 使用 VREF×1/2 的电压进行自诊断 1 1: 使用 VREF×1 的电压进行自诊断	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

ADDIAGR 寄存器是设定自诊断功能和使用自诊断功能时的电压的寄存器。  
 必须在 ADCSR.ADST 位为“0”时设定 ADDIAGR 寄存器。

## DIAG[1:0] 位 (自诊断位)

自诊断是用于检测 A/D 转换器故障的功能。对内部生成的 VREF×0、VREF×1/2 或者 VREF×1 这 3 个电压值进行转换。

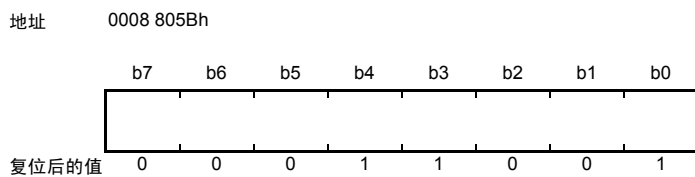
要进行自诊断时，必须通过 ADDIAGR.DIAG[1:0] 位选择电压值并且通过以下设定进行 A/D 转换：

- 单次模式 (ADCR.MODE[1:0] 位为“00b”)
- 只有模拟输入 AN0 有效 (ADCSR.CH[3:0] 位为“0000b”) (注1)
- 通过软件开始 A/D 转换 (ADCR.TRGS[2:0] 位为“000b”)

如果转换结束，就将转换结果保存到 A/D 数据寄存器 A (ADDRA)。此后，通过软件读 ADDRA，判断转换值是 (正常) 否 (异常) 在正常范围内。自诊断的执行时间和 1 个通道的 A/D 转换时间相同。

注 1. 当对 A/D 转换器进行自诊断时，必须设定 AN0 为输入通道。此设定的目的是选择保存转换结果的数据寄存器，但是 AN0 引脚为无效状态。

### 29.2.7 A/D 采样状态寄存器 (ADSSTR)



ADSSTR 寄存器是设定模拟输入的采样时间的 8 位可读写寄存器。

能在模拟输入的信号源阻抗高并且采样时间不够或者在 A/D 转换时钟 (ADCLK) 低速运行时调整采样时间。

设定值必须大于等于“02h”。

为了避免误动作，必须在 A/D 转换停止 (ADCSR.ADST 位为“0”) 的状态下进行改写。

详细内容请参照“29.3.3 输入采样和 A/D 转换时间”。



## 29.3 运行说明

A/D 转换器的运行模式有单次模式和扫描模式。

单次模式对指定的 1 个通道的模拟输入只进行 1 次转换。

扫描模式依次连续转换最多 12 个通道的模拟输入。

扫描模式有重复进行 A/D 转换的连续扫描以及对指定的通道只进行 1 个周期 A/D 转换的 1 个周期扫描。

### 29.3.1 单次模式

单次模式对指定的 1 个通道的模拟输入只进行 1 次 A/D 转换，如下所示：

1. 如果通过软件、MTU3、GPT 或者外部触发输入将 ADCSR.ADST 位置“1”（开始 A/D 转换），就开始所选通道的 A/D 转换。
2. 如果 A/D 转换结束，就将 A/D 转换结果保存到对应该通道的 A/D 数据寄存器 n（ADDRn）（n=A～L）。
3. 如果在 A/D 转换结束后 ADCSR.ADIE 位为“1”（允许由 A/D 转换结束引起的 ADI 中断），就产生 ADI 中断请求。
4. 在 A/D 转换过程中，ADCSR.ADST 位保持“1”，如果转换结束，此位就自动变为“0”，A/D 转换器进入待机状态。
5. 如果在 A/D 转换过程中将 ADCSR.ADST 位置“0”（停止 A/D 转换），就中止 A/D 转换，A/D 转换器进入待机状态。

选择 AN1 进行模拟输入时的运行例子如图 29.2 所示。

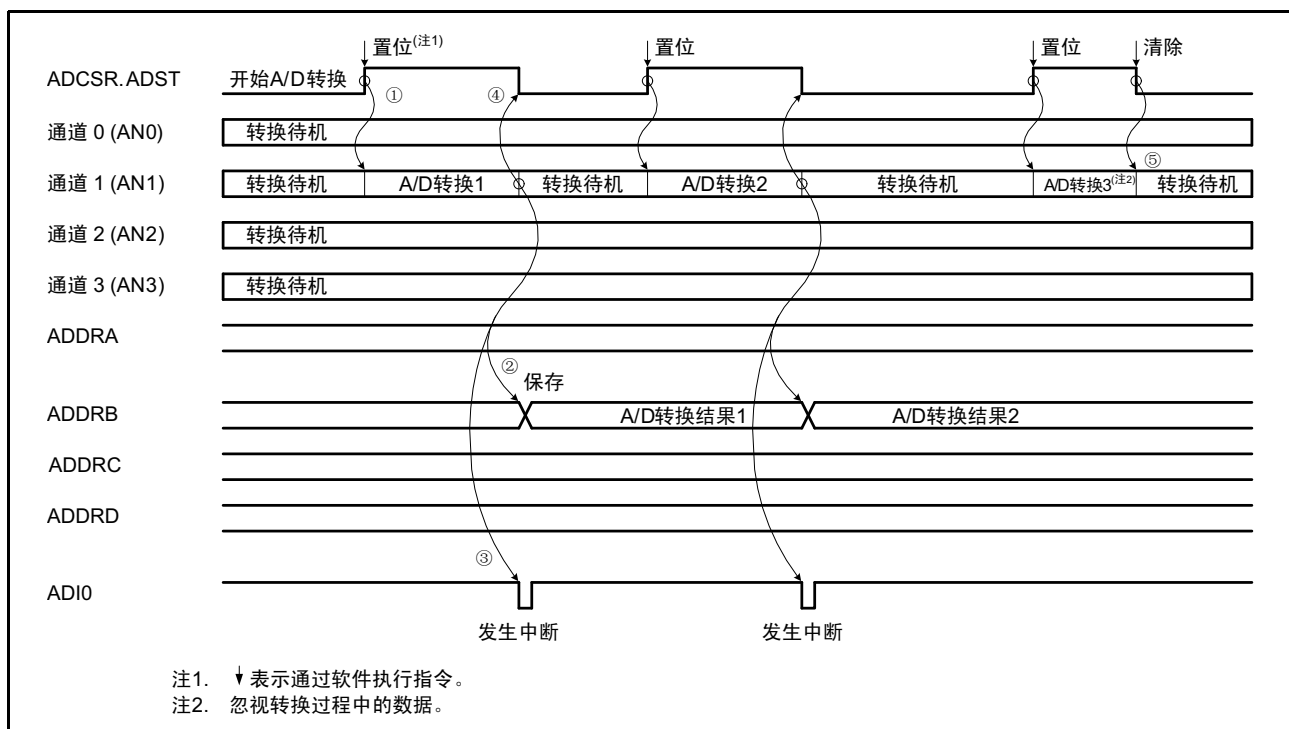


图 29.2 A/D 转换器的运行例子（单次模式）

### 29.3.2 扫描模式

扫描模式对最多 12 个通道的模拟输入依次连续进行 A/D 转换，如下所示。

扫描模式有重复进行 A/D 转换的连续扫描以及对指定的通道只进行 1 个周期 A/D 转换的 1 个周期扫描。

#### 29.3.2.1 连续扫描模式

连续扫描模式对指定通道的模拟输入重复进行 A/D 转换，如下所示：

1. 如果通过软件、MTU3、GPT 或者外部触发输入使 ADCSR.ADST 位变为“1”（开始 A/D 转换），就从所选通道中通道号小的通道开始 A/D 转换。
2. 如果 A/D 转换结束，就将 A/D 转换结果保存到对应的 A/D 数据寄存器 n（ADDRn）（n=A~L）。
3. 如果在所选通道的 A/D 转换全部结束后 ADCSR.ADIF 位为“1”（允许由 A/D 转换结束引起的 ADI 中断），就产生 ADI 中断请求。A/D 转换器就从通道号小的通道开始 A/D 转换。
4. ADCSR.ADST 位不自动变为“0”，而在为“1”的期间重复 2.~3. 的运行。如果将 ADCSR.ADST 位置“0”（停止 A/D 转换），就中止 A/D 转换并且 A/D 转换器进入待机状态。
5. 此后，如果将 ADCSR.ADST 位置“1”（开始 A/D 转换），就重新从通道号小的通道开始 A/D 转换。

选择 AN0 ~ AN2 的 3 个通道进行模拟输入时的运行例子如图 29.3 所示。

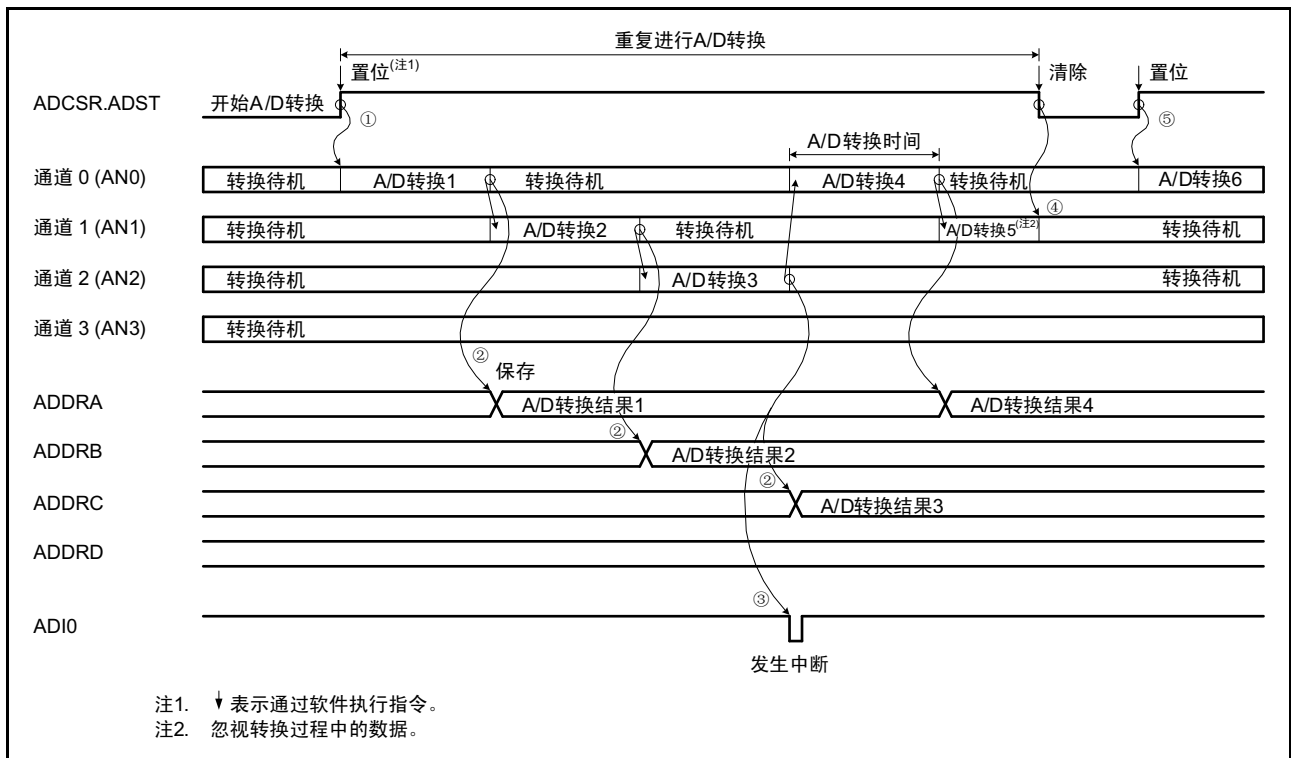


图 29.3 A/D 转换器的运行例子（连续扫描模式）

29.3.2.2 1 个周期扫描模式

1 个周期扫描模式对指定通道的模拟输入只进行 1 个周期的 A/D 转换，如下所示：

1. 如果通过软件、MTU3、GPT 或者外部触发输入使 ADCSR.ADST 位变为“1”（开始 A/D 转换），就从所选通道中通道号小的通道开始 A/D 转换。
2. 如果 A/D 转换结束，就将 A/D 转换结果保存到对应的 A/D 数据寄存器 n（ADDRn）（n=A~L）。
3. 如果在所选通道的 A/D 转换全部结束后 ADCSR.ADIE 位为“1”（允许由 A/D 转换结束引起的 ADI 中断），就产生 ADI 中断请求。
4. 在 A/D 转换过程中，如果 ADCSR.ADST 位保持“1”，并且所选通道的 A/D 转换全部结束，此位就自动变为“0”，A/D 转换器进入待机状态。

选择 AN0 ~ AN2 的 3 个通道进行模拟输入时的运行例子如图 29.4 所示。

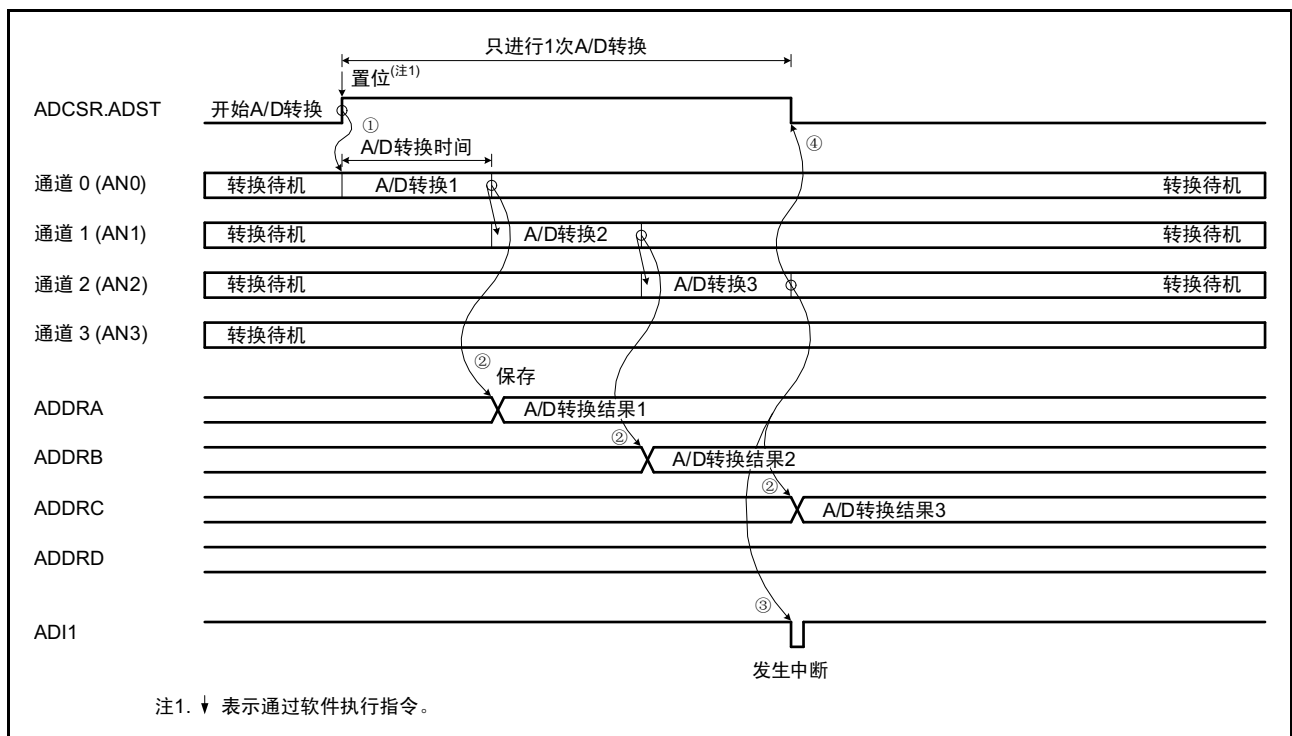


图 29.4 A/D 转换器的运行例子（1 个周期扫描模式）

### 29.3.3 输入采样和 A/D 转换时间

在通过软件、MTU3、GPT 或者外部触发产生 A/D 转换开始条件后经过 A/D 转换开始延迟时间 (tD)，A/D 转换器对模拟输入进行采样，然后开始 A/D 转换。

A/D 转换时序如图 29.5 所示。

A/D 转换开始条件产生后的 A/D 转换时间 (tCONV) 为 tD、输入采样时间 (tSPL) 和逐次转换时间 (tSAM) 的总时间。此后的 A/D 转换时间 (tCONV) 还包括 tSPL 和 tSAM 的时间。

采样时间 (tSPL) 是给 A/D 转换器的采样 & 保持电路充电的时间。在模拟输入的信号源阻抗高并且采样时间不够的情况下或者在 A/D 转换时钟 (ADCLK) 低速运行的情况下，能通过 ADSSTR 寄存器调整采样时间。

逐次转换时间 (tSAM) 总是固定为 25 个 ADCLK 周期。

ADSSTR 寄存器的设定例子以及 A/D 转换时间分别如表 29.7 和表 29.8 所示。

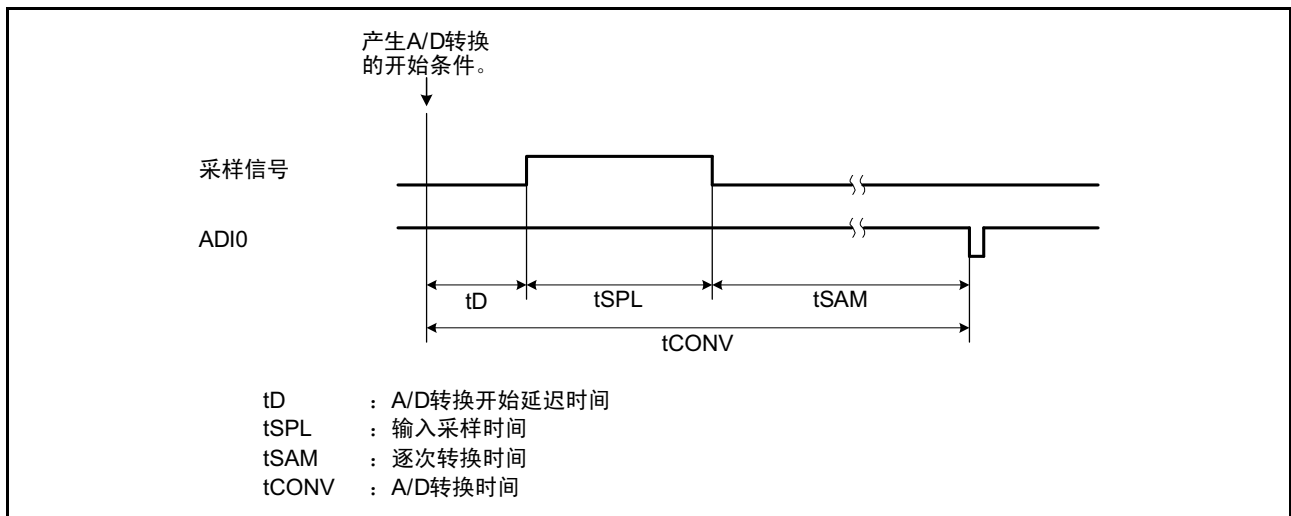


图 29.5 A/D 转换时序

表 29.7 ADSSTR 寄存器的设定例子

使用例子	设定范围	采样时间 (注)
标准 (初始值)	"19h"	0.5μs (当 PCLK=ADCLK=50MHz 时)
在模拟输入的信号源阻抗高并且采样时间不够的情况下进行设定。	"1Ah" ~ "FFh"	例: "FFh" 5.1μs (当 PCLK=ADCLK=50MHz 时)
在 ADCLK 小于 50MHz 以及采样时间比初始值短的情况下进行设定。	"02h" ~ "18h"	例: "14h" 0.5μs (当 PCLK=ADCLK=40MHz 时)

注. 必须设定为采样时间 ≥ 0.5μs, 用以下表达式计算采样时间:

$$\text{采样时间 } (\mu\text{s}) = \frac{\text{ADSSTR寄存器的设定值}}{\text{ADCLK (MHz)}}$$

表 29.8 A/D 转换时间

项目	符号	表达式	
		min	max
A/D 转换开始的延迟时间 (①)	tD	$\frac{3}{\text{PCLK (MHz)}}$	$\frac{1}{\text{ADCLK (MHz)}} + \frac{4}{\text{PCLK (MHz)}}$
输入采样时间 (②)	tSPL	$\frac{\text{ADSSTR寄存器的设定值}}{\text{ADCLK (MHz)}}$	
逐次转换时间 (③)	tSAM	$\frac{25}{\text{ADCLK (MHz)}}$	
A/D 转换时间 (注1)	tCONV	① + ② + ③	
A/D 转换时间 (注2)	tCONV	② + ③	

注1. 单次模式或者扫描模式 (第1次) 的 A/D 转换时间

注2. 扫描模式 (第2次以后) 的 A/D 转换时间

A/D 转换时间的计算例子如下所示:

当 PCLK=ADCLK=50MHz, ADSSTR=19h 并且为扫描模式 (第2次) 时

$$\begin{aligned}
 \text{A/D 转换时间 (tCONV)} &= \text{ADSSTR/ADCLK} + 25/\text{ADCLK} \\
 &= 25/50\text{MHz} + 25/50\text{MHz} \\
 &= 0.5\mu\text{s} + 0.5\mu\text{s} \\
 &= 1.0\mu\text{s}
 \end{aligned}$$

当 PCLK=ADCLK=40MHz, ADSSTR=14h 并且为扫描模式 (第1次: min) 时

$$\begin{aligned}
 \text{A/D 转换时间 (tCONV)} &= 3/\text{PCLK} + \text{ADSSTR/ADCLK} + 25/\text{ADCLK} \\
 &= 3/40\text{MHz} + 20/40\text{MHz} + 25/40\text{MHz} \\
 &= 0.075\mu\text{s} + 0.5\mu\text{s} + 0.625\mu\text{s} \\
 &= 1.2\mu\text{s}
 \end{aligned}$$

### 29.3.4 通过外部触发开始的 A/D 转换

能通过外部触发的输入 (ADTRG#) 开始 A/D 转换。

如果将 ADSTRGR.ADSTRS[4:0] 位置 “00000b” (ADTRG# 的触发), 就在 ADTRG# 的下降沿 ADCSR.ADST 位变为 “1” (开始 A/D 转换), 开始 A/D 转换, 此时序如图 29.6 所示。

必须注意: 在使用外部触发时, 如果外部触发的输入已经为 Low 电平, 内部信号就可能产生下降沿并且开始 A/D 转换。

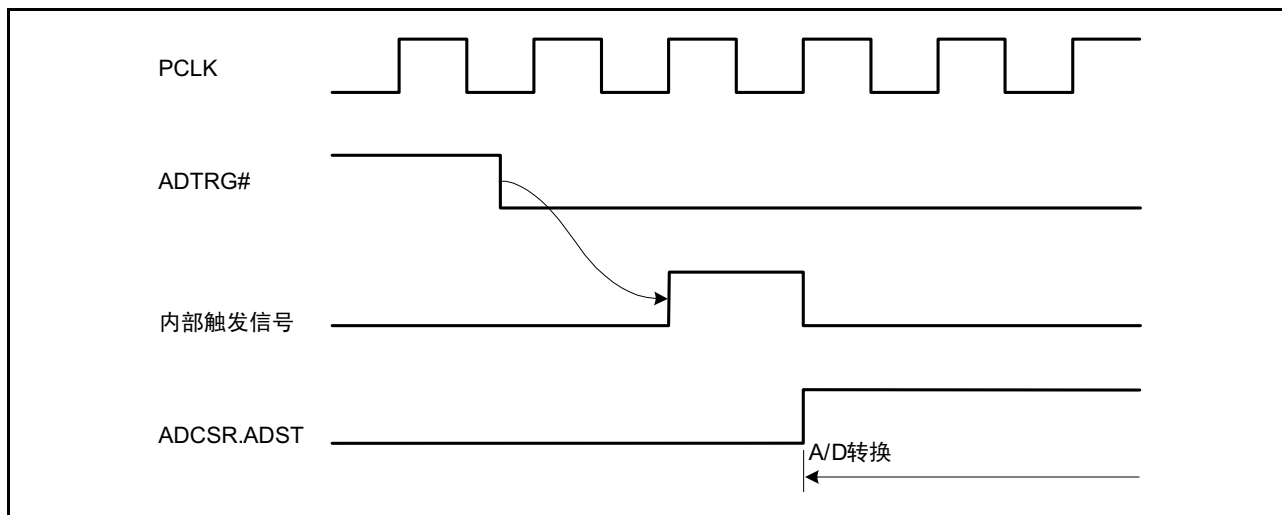


图 29.6 外部触发输入时序

### 29.3.5 通过 MTU3、GPT 开始的 A/D 转换

能通过 MTU3、GPT 的间隔定时器的 A/D 转换请求开始 A/D 转换。

通过 MTU3、GPT 启动 A/D 比较器时, 必须通过 ADSTRGR.ADSTRS[4:0] 位设定 A/D 转换启动源。如果在此状态下产生 MTU3、GPT 的间隔定时器的 A/D 转换请求, ADCSR.ADST 位就变为 “1” (开始 A/D 转换), 并且开始 A/D 转换。从 ADCSR.ADST 位变为 “1” 到开始 A/D 转换的时序与通过软件给 ADCSR.ADST 位写 “1” 时相同。

## 29.4 中断源

当 ADCSR.ADIE 位为“1”（允许由 A/D 转换结束引起的 ADI 中断）时，如果 A/D 转换结束，A/D 转换器就发生中断（ADI0）。

能使用 ADI 中断启动数据传送控制器（DTC）。如果通过 ADI 中断启动 DTC，并且读被转换的数据，就能不通过 CPU 进行连续的转换。

表 29.9 A/D 转换器的中断源

名称	中断源	中断状态标志	DTC 的启动
ADI0	A/D 转换结束	ICU.IR98.IR	能

## 29.5 A/D 转换精度的定义

RX62T 群的 A/D 转换精度的定义如下：

- 分辨率  
这是指 A/D 转换器的数字输出码数。
- 量化误差  
这是 A/D 转换器固有的偏差，为 1/2LSB（图 29.7）。
- 偏移误差  
这是在数字输出从最小电压值“000000000b（000h）”变为“000000001b（001h）”时与模拟输入电压值的理想 A/D 转换特性的偏差（图 29.8）。
- 满刻度误差  
这是在数字输出从“111111110b（3FEh）”变为“111111111b（3FFh）”时与模拟输入电压值的理想 A/D 转换特性的偏差（图 29.8）。
- 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 29.8）。
- 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。

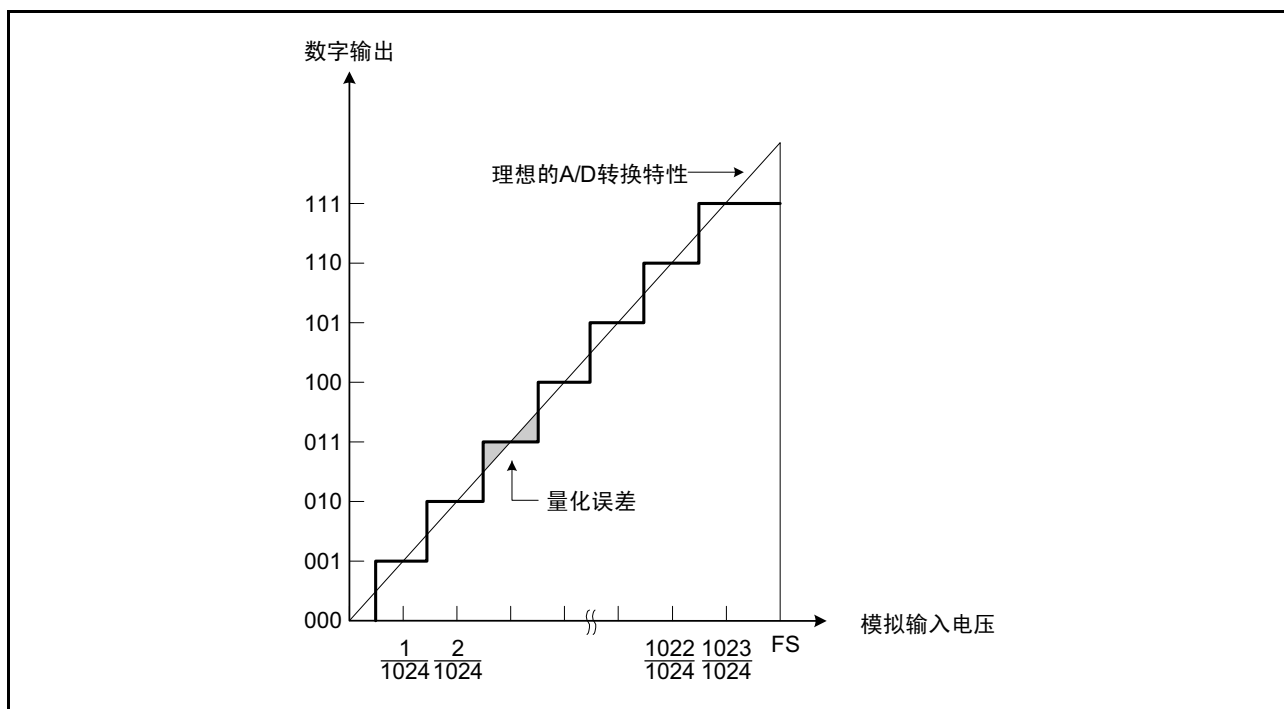


图 29.7 A/D 转换精度的定义 (1)

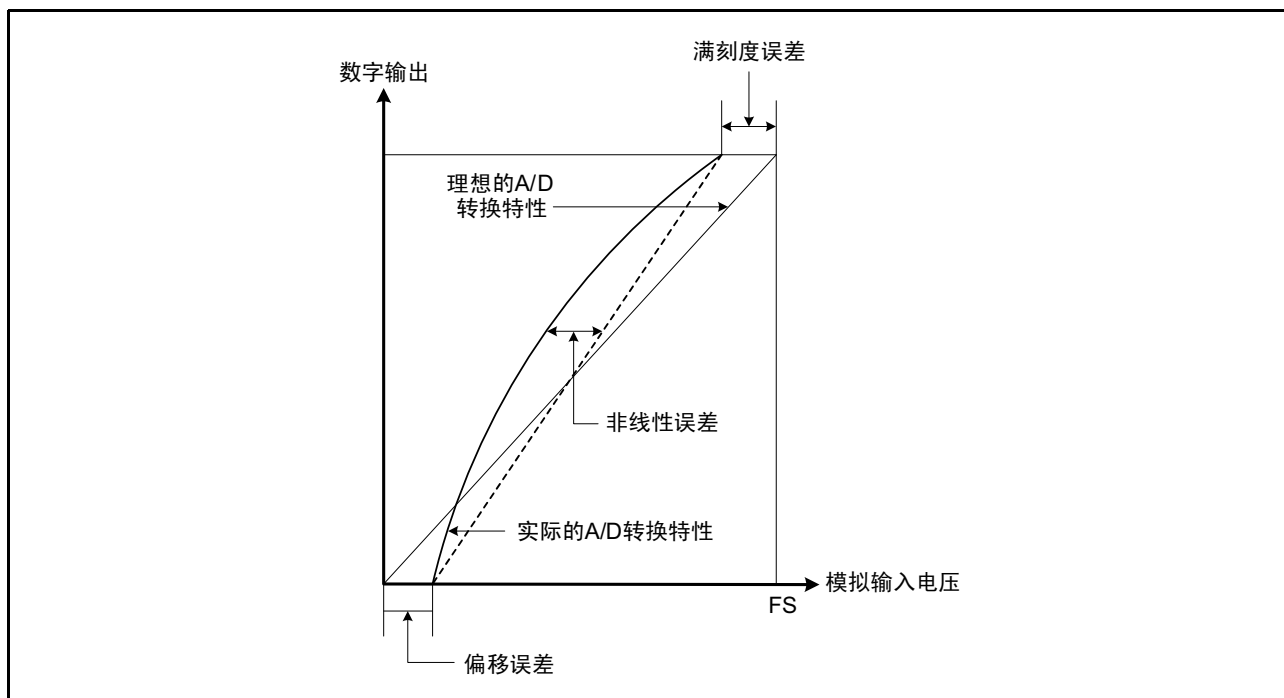


图 29.8 A/D 转换精度的定义 (2)



## 29.6 使用时的注意事项

### 29.6.1 模块停止功能的设定

能通过模块停止控制寄存器设定禁止或者允许 A/D 转换器的运行，初始值为停止 A/D 转换器的运行。能通过解除模块停止状态，使寄存器变为可存取的状态。详细内容请参照“9. 低功耗功能”。

### 29.6.2 停止 A/D 转换时的注意事项

在选择外部触发或者定时器为 A/D 转换开始条件时，为了停止 A/D 转换，必须将 ADCSR.ADST 位置“0”（停止 A/D 转换）。

### 29.6.3 重新开始 A/D 转换时的注意事项

如果通过将 ADCSR.ADST 位置“0”来停止 A/D 转换，就需要 1 个 ADCLK 时钟的时间来停止 A/D 转换器的模拟电路。

如果在通过将 ADCSR.ADST 位置“0”停止 A/D 转换后，重新开始 A/D 转换，就在经过 1 个 ADCLK 时钟的时间后，将 ADCSR.ADST 位置“1”。

### 29.6.4 向低功耗状态转移时的注意事项

在允许 A/D 转换的状态下，如果 RX62T 群向模块停止状态或者软件待机模式转移，模拟电源的电流就和 A/D 转换过程中的电流相同。如果需要在模块停止状态或者软件待机模式中减少模拟电源电流，就必须停止 A/D 转换。在停止 A/D 转换时，需要确保从 ADCSR.ADST 位被置“0”后到 A/D 转换器的模拟电路停止前的时间。为了确保此时间，必须按照以下步骤进行设定。

必须在将 ADCSR.ADST 位置“0”后，将 ADCR.CKS[1:0] 位置“11b”（PCLK）。此后，必须在确认 A/D 转换已经停止后转移到模块停止状态或者软件待机模式状态。

### 29.6.5 有关容许信号源阻抗

为了实现  $1.0\mu\text{s}$  的高速转换，对于信号源阻抗不超过  $1.0\text{k}\Omega$  的输入信号，RX62T 群的模拟输入能保证转换精度。在单次模式中进行转换并且外接大电容的情况下，因为输入负载实际上只有  $8\text{k}\Omega$  的内部输入电阻，所以信号源阻抗可忽略不计。但是，由于形成低通滤波器，所以有可能无法迎合急剧变化的模拟信号（例如变动率至少为  $5\text{mV/s}$ ）（图 29.9）。在转换高速模拟信号或者在扫描模式中进行转换时，必须插入 1 个输出阻抗低的缓冲放大器。

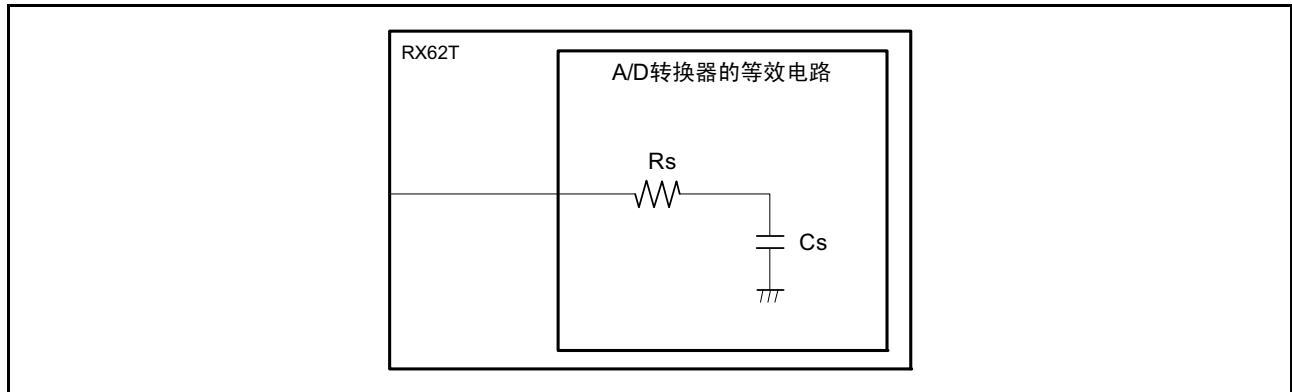


图 29.9 模拟输入引脚的内部等效电路

表 29.10 模拟引脚的规格

项目	min	max	单位
容许信号源阻抗	—	1.0	$\text{k}\Omega$
引脚的内部等效电路	Rs	8.0	$\text{k}\Omega$
	Cs	7.0	$\text{pF}$

### 29.6.6 对绝对精度的影响

由于附加电容会导致与 GND 的耦合，如果 GND 有噪声就可能降低绝对精度，所以必须连接 AVSS 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。

### 29.6.7 模拟电源引脚等的设定范围

当超过以下所示的电压设定范围并且使用 LSI，可能对 LSI 的可靠性造成不良影响。

- 模拟输入电压的设定范围  
给模拟输入引脚 AN<sub>n</sub> 外加的电压必须在  $\text{AVSS} \leq \text{VAN} \leq \text{VREF}$  的范围内。
- 各电源引脚（AVCC0-AVSS0、AVCC-AVSS、VCC-VSS）的关系  
各电源引脚（AVCC0-AVSS0、AVCC-AVSS、VCC-VSS）的关系必须为  $\text{VCC} \leq \text{AVCC0} = \text{AVCC}$  并且  $\text{AVSS0} = \text{AVSS} = \text{VSS}$ 。在不使用 10 位 A/D 转换器时，必须为  $\text{VCC} = \text{AVCC0} = \text{AVCC}$ 、 $\text{AVSS0} = \text{AVSS} = \text{VSS}$ 。
- VREF 的设定范围  
VREF 引脚的基准电压的设定范围必须为  $\text{VREF} \leq \text{AVCC}$ 。在不使用 A/D 转换器时，必须为  $\text{VREF} = \text{AVCC}$ 。

### 29.6.8 设计电路板时的注意事项

在设计电路板时，必须尽量将数字电路和模拟电路分开，而且不能使数字电路的信号线和模拟电路的信号线交叉或者靠近，否则会给模拟信号带来噪声并且对 A/D 转换值造成不良影响。必须通过模拟接地 (AVSS) 将模拟输入引脚 (AN0 ~ AN11)、模拟基准电源 (VREF) 和模拟电源电压 (AVCC) 与数字电路分开，另外，模拟接地 (AVSS) 必须与电路板上稳定的接地 (VSS) 进行单点连接。

### 29.6.9 噪声对策的注意事项

如图 29.10 所示，为了防止过大电涌等异常电压对模拟输入引脚 (AN0 ~ AN11) 造成破坏，必须在 AVCC 和 AVSS 之间以及 VREF 和 AVSS 之间连接电容器，并且以模拟输入引脚 (AN0 ~ AN11) 为基准连接保护电路。

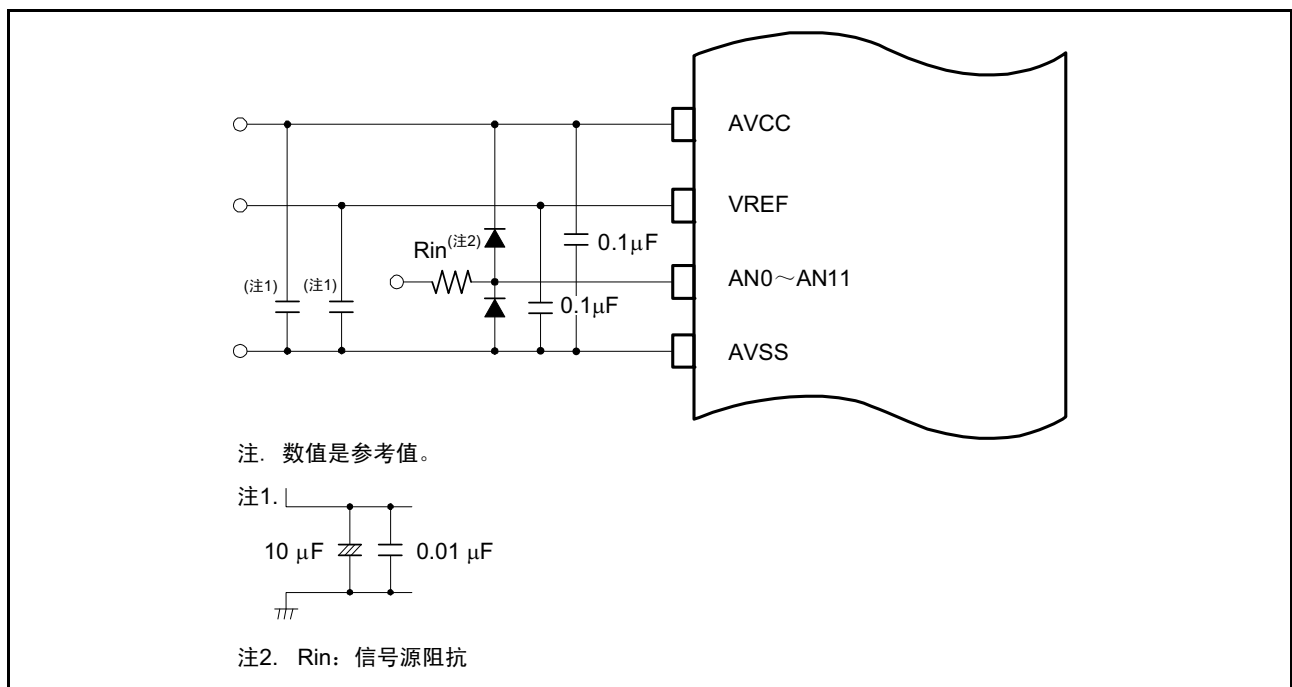


图 29.10 模拟输入的保护电路例子

## 30. RAM

RX62T 群内置高速静态 RAM。

### 30.1 概要

RAM 的规格如表 30.1 所示。

表 30.1 RAM 的规格

项目	内容
RAM 容量	16K 字节 /8K 字节 (注 1)
RAM 地址	0000 0000h ~ 0000 3FFFh (16K 字节) 0000 0000h ~ 0000 1FFFh (8K 字节) (注 1)
存取	<ul style="list-style-type: none"> <li>以 1 个周期进行读写。</li> <li>可选择内部 RAM 的有效或者无效 (注 2)。</li> </ul>
低功耗功能	能设定为模块停止状态。

注 1. RAM 容量因产品种类而不同。

产品型号	RAM 容量	RAM 地址
R5F562TAxxxx	16K 字节	0000 0000h ~ 0000 3FFFh
R5F562T7xxxx	8K 字节	0000 0000h ~ 0000 1FFFh
R5F562T6xxxx		

注 2. 能通过 SYSCR1.RAME 位进行选择。有关 SYSCR1 寄存器，请参照“3.2.4 系统控制寄存器 1 (SYSCR1)”。

## 30.2 运行说明

### 30.2.1 低功耗功能

通过设定模块停止控制寄存器 C (MSTPCRC) 停止给 RAM 提供时钟，能降低功耗。

如果将 MSTPCRC.MSTPC0 位置“1”，提供给 RAM 的时钟就停止。

通过停止提供时钟，使 RAM 变为模块停止状态。但是在复位后，RAM 运行。

一旦进入模块停止状态，就无法存取 RAM。不能在存取 RAM 的过程中转移到模块停止状态。

有关 MSTPCRC 寄存器的详细内容，请参照“9. 低功耗功能”。

## 31. ROM（保存代码的闪存）

RX62T 群内置用于保存最多 256K 字节代码的闪存（ROM）以及用于保存最多 32K 字节数据的闪存（数据闪存）。

本章说明用于保存代码的闪存，有关数据闪存请参照“32. 数据闪存（保存数据的闪存）”。

### 31.1 概要

ROM 的规格如表 31.1 所示，ROM 和数据闪存外围的框图如图 31.1 所示。

表 31.1 ROM 的规格

项目	内容	
存储空间	• 用户 MAT: 256K 字节 /128K 字节 /64K 字节 (注 1)	
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"> <li>• 内置对 ROM 进行改写的专用定序器 (FCU)。</li> <li>• 能通过给 FCU 发行命令, 进行 ROM 的编程或者擦除。</li> <li>• 如果读擦除状态的 ROM, 就能以 32 位为单位读到“FFFF FFFFh”。</li> </ul>	
BGO (后台操作) 功能	<ul style="list-style-type: none"> <li>• 在对 ROM 进行编程或者擦除的期间, CPU 能执行非 ROM 区和非数据闪存区的程序。</li> <li>• 在对数据闪存进行编程或者擦除的期间, CPU 能执行 ROM 区的程序。</li> </ul>	
挂起 / 恢复功能	<ul style="list-style-type: none"> <li>• 中止 ROM 的编程或者擦除, CPU 能执行 ROM 区的程序 (挂起)。</li> <li>• 在中止后, 能重新开始对 ROM 进行编程或者擦除 (恢复)。</li> </ul>	
编程 / 擦除单位	<ul style="list-style-type: none"> <li>• 用户 MAT 的编程单位: 256 字节</li> <li>• 用户 MAT 的擦除单位: 4K 字节 (8 块)、16K 字节 (当 ROM 容量为 256K 字节时: 14 块; 当 ROM 容量为 128K 字节时: 6 块; 当 ROM 容量为 64K 字节时: 2 块)</li> </ul>	
板上编程 (2 种)	通过引导模式改写 <ul style="list-style-type: none"> <li>• 使用异步串行接口 (SCI1)。</li> <li>• 自动调整通信速度。</li> </ul> 通过用户程序中的 ROM 改写程序进行改写 <ul style="list-style-type: none"> <li>• 不用对系统进行复位就能改写 ROM。</li> </ul>	
板外编程	能使用 PROM 编程器改写用户 MAT。	
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。
	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“33. 电特性”。	

注 1. ROM 容量因产品种类而不同。

产品型号	ROM 容量	ROM 地址	
		用于读	用于写 / 擦除
R5F562TAxxxx	256K 字节	FFFC 0000h ~ FFFF FFFFh	00FC 0000h ~ 00FF FFFFh
R5F562T7xxxx	128K 字节	FFFE 0000h ~ FFFF FFFFh	00FE 0000h ~ 00FF FFFFh
R5F562T6xxxx	64K 字节	FFFF 0000h ~ FFFF FFFFh	00FF 0000h ~ 00FF FFFFh

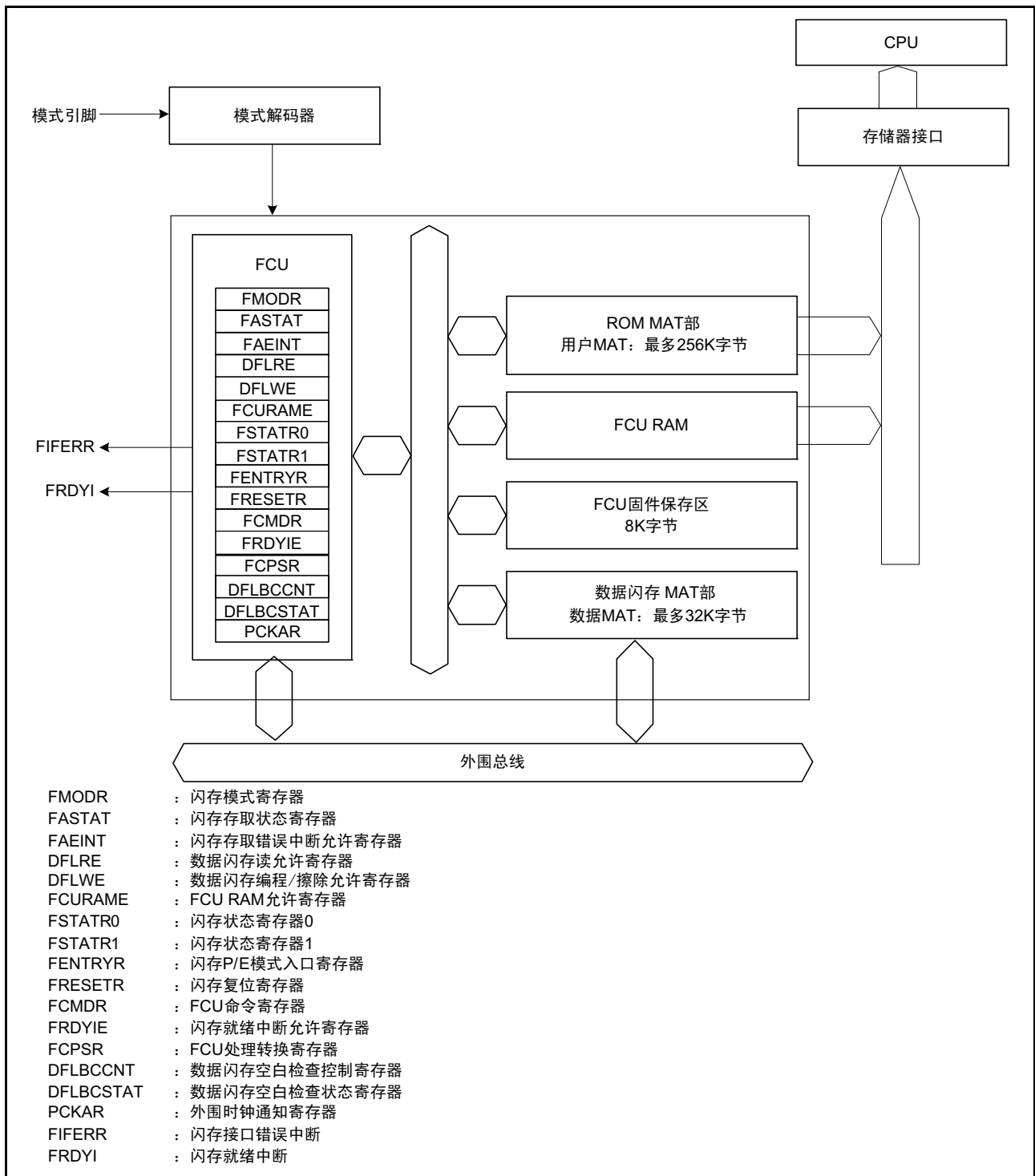


图 31.1 ROM 的框图

ROM 的相关输入 / 输出引脚如表 31.2 所示。

表 31.2 ROM 的相关输入 / 输出引脚

引脚名	输入 / 输出	功能
PD5/RxD1	输入	用于引导模式。SCI1 的接收数据（用于主机通信）。
PD3/TxD1	输出	用于引导模式。SCI1 的发送数据（用于主机通信）。
MD1、MD0	输入	设定 RX62T 群的运行模式。

## 31.2 寄存器说明

ROM 的相关寄存器一览表如表 31.3 所示。部分寄存器也有数据闪存的相关位，本章只说明 ROM 的相关位的功能。有关数据闪存相关位功能的详细内容，请参照“32. 数据闪存（保存数据的闪存）”的“32.2 寄存器说明”。

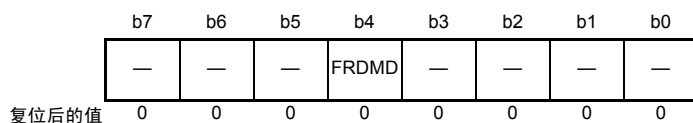
通过复位对 ROM 的相关寄存器进行初始化。

表 31.3 ROM 的相关寄存器一览表

寄存器名	符号	复位后的值	地址	存取长度
闪存模式寄存器	FMODR	00h	007F C402h	8
闪存存取状态寄存器	FASTAT	00h	007F C410h	8
闪存存取错误中断允许寄存器	FAEINT	9Bh	007F C411h	8
闪存就绪中断允许寄存器	FRDYIE	00h	007F C412h	8
FCU RAM 允许寄存器	FCURAME	0000h	007F C454h	16
闪存状态寄存器 0	FSTATR0	80h	007F FFB0h	8
闪存状态寄存器 1	FSTATR1	0xh	007F FFB1h	8
闪存 P/E 模式入口寄存器	FENTRYR	0000h	007F FFB2h	16
闪存保护寄存器	FPROTR	0000h	007F FFB4h	16
闪存复位寄存器	FRESETR	0000h	007F FFB6h	16
FCU 命令寄存器	FCMDR	FFFFh	007F FFBAh	16
FCU 处理转换寄存器	FCPSR	0000h	007F FFC8h	16
闪存 P/E 状态寄存器	FPESTAT	0000h	007F FFCCh	16
外围时钟通知寄存器	PCKAR	0000h	007F FFE8h	16
闪存编程 / 擦除保护寄存器	FWEPROR	02h	0008 C289h	8

## 31.2.1 闪存模式寄存器（FMODR）

地址 007F C402h



位	符号	位名	功能	R/W
b3-b0	—	保留位	读写值都为“0”。	R/W
b4	FRDMD	FCU 读模式选择位	0: 存储区读方式 在 ROM 锁定位读模式中读 ROM 的锁定位时进行设定。 1: 寄存器读方式 在使用锁定位读 2 命令读 ROM 的锁定位时进行设定。	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W

FMODR 寄存器是指定锁定位的读法的寄存器。

在内部 ROM 无效模式中，FMODR 寄存器的读数据为“00h”，不能写。

通过复位对 FMODR 寄存器进行初始化。

## FRDMD 位（FCU 读模式选择位）

此位指定锁定位的读法。

在使用数据闪存的空白检查命令时，需要设定为寄存器读方式。

详细内容请参照“32. 数据闪存（保存数据的闪存）”。



## 31.2.2 闪存存取状态寄存器（FASTAT）

地址 007F C410h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROMAE	—	—	CMDLK	DFLAE	—	DFLRPE	DFLWPE
复位后的值	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	DFLWPE	数据闪存的编程 / 擦除保护违反位	请参照“32. 数据闪存（保存数据的闪存）”。	R/(W) (注1)
b1	DFLRPE	数据闪存的读保护违反位	请参照“32. 数据闪存（保存数据的闪存）”。	R/(W) (注1)
b2	—	保留位	读写值都为“0”。	R/W
b3	DFLAE	数据闪存的存取违反位	请参照“32. 数据闪存（保存数据的闪存）”。	R/(W) (注1)
b4	CMDLK	FCU 命令锁定位	0: FCU 不处于命令锁定状态 1: FCU 处于命令锁定状态	R
b6-b5	—	保留位	读写值都为“0”。	R/W
b7	ROMAE	ROM 存取违反位	0: 无 ROM 存取错误 1: 有 ROM 存取错误	R/(W) (注1)

注1. 为了将标志置“0”，只能在读“1”后写“0”。

FASTAT 寄存器是确认有无 ROM/ 数据闪存的存取违反的寄存器。

在内部 ROM 无效模式中，FASTAT 寄存器的读数据为“00h”，不能写。如果 FASTAT 寄存器的某位变为“1”，FCU 就进入命令锁定状态（参照“31.8.2 错误保护”）。为了解除命令锁定状态，需要在将 FASTAT 寄存器置“10h”后，给 FCU 发行状态寄存器清除命令。

通过复位对 FASTAT 寄存器进行初始化。

## CMDLK 位（FCU 命令锁定位）

此位表示 FCU 处于命令锁定状态（参照“31.8.2 错误保护”）。

[ 为“1”的条件 ]

- FCU 检测到错误并且转移到命令锁定状态后

[ 为“0”的条件 ]

- 在 FASTAT 寄存器为“10h”的状态下给 FCU 发行状态寄存器清除命令后

## ROMAE 位（ROM 存取违反位）

此位表示有无 ROM 的存取违反。

如果 ROMAE 位变为“1”，FSTATR0.ILGLERR 位就变为“1”，FCU 进入命令锁定状态。

[ 为“1”的条件 ]

- 在 FENTRYR.FENTRY0 位为“1”并且在 ROM P/E 正常模式的状态下，对 ROM 的编程/擦除地址 00FC 0000h ~ 00FF FFFFh 发行读存取命令。
- 在 FENTRY0 位为“0”的状态下，对 ROM 的编程/擦除地址 00FC 0000h ~ 00FF FFFFh 发行存取命令。
- 在 FENTRYR 寄存器不为“000h”的状态下，对 ROM 的读地址 FFFC 0000h ~ FFFF FFFFh 发行读存取命令。

[ 为“0”的条件 ]

- 读“1”后写“0”时

## 31.2.3 闪存存取错误中断允许寄存器（FAEINT）

地址 007F C411h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROMAEIE	—	—	CMDLKIE	DFLAEIE	—	DFLRPEIE	DFLWPEIE
复位后的值	1	0	0	1	1	0	1	1

位	符号	位名	功能	R/W
b0	DFLWPEIE	数据闪存的编程 / 擦除保护违反中断允许位	请参照“32. 数据闪存（保存数据的闪存）”。	R/W
b1	DFLRPEIE	数据闪存的读保护违反中断允许位	请参照“32. 数据闪存（保存数据的闪存）”。	R/W
b2	—	保留位	读写值都为“0”。	R/W
b3	DFLAEIE	数据闪存的存取违反中断允许位	请参照“32. 数据闪存（保存数据的闪存）”。	R/W
b4	CMDLKIE	FCU 命令锁定中断允许位	0: 在 FASTAT.CMDLK 位为“1”时，不产生 FIFERR 中断请求。 1: 在 FASTAT.CMDLK 位为“1”时，产生 FIFERR 中断请求。	R/W
b6-b5	—	保留位	读写值都为“0”。	R/W
b7	ROMAEIE	ROM 存取违反中断允许位	0: 在 FASTAT.ROMAE 位为“1”时，不产生 FIFERR 中断请求。 1: 在 FASTAT.ROMAE 位为“1”时，产生 FIFERR 中断请求。	R/W

FAEINT 寄存器设定允许或者禁止闪存接口错误中断（FIFERR）的输出。

在内部 ROM 无效模式中，FAEINT 寄存器的读数据为“00h”，不能写。

通过复位对 FAEINT 寄存器进行初始化。

## CMDLKIE 位（FCU 命令锁定中断允许位）

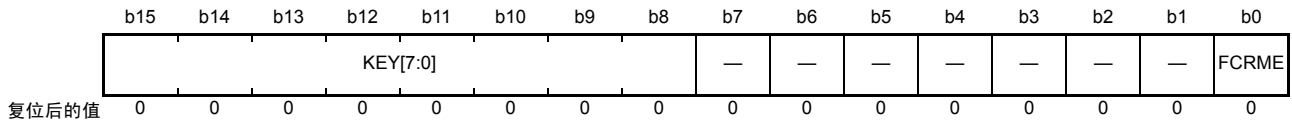
在发生 FCU 命令锁定并且 FASTAT.CMDLK 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的产生。

## ROMAEIE 位（ROM 存取违反中断允许位）

在发生 ROM 存取违反并且 FASTAT.ROMAE 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的产生。

## 31.2.4 FCU RAM 允许寄存器（FCURAME）

地址 007F C454h



位	符号	位名	功能	R/W
b0	FCRME	FCU RAM 允许位	0: 禁止存取 FCU RAM 1: 允许存取 FCU RAM	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W
b15-b8	KEY[7:0]	键码	控制能否改写 FCRME 位。	R/(W) (注1)

注 1. 不保持写数据。

FCURAME 寄存器允许或者禁止存取 FCU RAM 区。

只在给高位字节写特定值时才能通过字存取写此寄存器，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，FCURAME 寄存器的读数据为“00h”，不能写。

通过复位对 FCURAME 寄存器进行初始化。

## FCRME 位（FCU RAM 允许位）

此位允许或者禁止存取 FCU RAM。

只在 KEY[7:0] 位为“C4h”时才能通过字存取写 FCRME 位。要写 FCU RAM 时，必须将 FENTRYR 寄存器置“0000h”，使 FCU 停止。

## KEY[7:0] 位（键码）

这些位控制能否改写 FCRME 位。

不保持 KEY[7:0] 位的写数据。

## 31.2.5 闪存状态寄存器 0（FSTATR0）

地址 007F FF0h

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/ 数据闪存的非法存取 1: FCU 检测到非法命令或者 ROM/ 数据闪存的非法存取	R
b7	FRDY	闪存就绪位	0: 正在处理编程或者擦除、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、或者正在处理数据闪存的空白检查（参照“32. 数据闪存（保存数据的闪存）”）。 1: 没有执行上述处理	R

FSTATR0 寄存器是确认 FCU 状态的寄存器。

在内部 ROM 无效模式中，FSTATR0 寄存器的读数据为“00h”。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FSTATR0 寄存器进行初始化。

## PRGSPD 位（编程挂起状态位）

此位表示 FCU 正在进行编程的中止处理或者已经转移到编程挂起状态，详细内容请参照“31.7 挂起”。

[为“1”的条件]

- 开始进行编程的中止处理时

[为“0”的条件]

- 接受恢复命令时

**ERSSPD 位（擦除挂起状态位）**

此位表示 FCU 正在进行擦除的中止处理或者已经转移到擦除挂起状态，详细内容请参照“31.7 挂起”。

[为“1”的条件]

- 开始进行擦除的中止处理时

[为“0”的条件]

- 接受恢复命令时

**SUSRDY 位（挂起就绪位）**

此位表示 FCU 能否接受 P/E 挂起命令。

[为“1”的条件]

- 在开始进行编程或者擦除处理后，转移到能接受 P/E 挂起命令的状态时

[为“0”的条件]

- 接受 P/E 挂起命令时
- 在编程或者擦除处理过程中转移到命令锁定状态时

**PRGERR 位（编程错误位）**

此位表示通过 FCU 对 ROM/ 数据闪存进行编程处理的结果。

当 PRGERR 位为“1”时，FCU 进入命令锁定状态，详细内容请参照“31.8.2 错误保护”。

[为“1”的条件]

- 在编程过程中发生错误时
- 对由锁定位保护的区域发行编程命令时

[为“0”的条件]

- FCU 发行状态寄存器清除命令后

**ERSERR 位（擦除错误位）**

此位表示通过 FCU 对 ROM/ 数据闪存进行擦除处理的结果。

当 ERSERR 位为“1”时，FCU 进入命令锁定状态，详细内容请参照“31.8.2 错误保护”。

[为“1”的条件]

- 在擦除过程中发生错误时
- 对由锁定位保护的区域发行擦除命令时

[为“0”的条件]

- FCU 发行状态寄存器清除命令后

**ILGLERR 位（非法命令错误位）**

此位表示 FCU 检测到非法命令或者 ROM/ 数据闪存的非法存取。

当 ILGLERR 位为“1”时，FCU 进入命令锁定状态，详细内容请参照“31.8.2 错误保护”。

[为“1”的条件]

- FCU 检测到非法命令时
- FCU 检测到 ROM/ 数据闪存的非法存取时  
（FASTAT.ROMAE 位、FASTAT.DFLAE 位、FASTAT.DFLRPE 位或者 FASTAT.DFLWPE 位为“1”）
- FENTRYR 寄存器的设定不正确时

[为“0”的条件]

- 在 FASTAT 寄存器为“10h”的状态下，FCU 发行状态寄存器清除命令后

**FRDY 位（闪存就绪位）**

此位是确认 FCU 处理状态的位。

## 31.2.6 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	x	x

x: 不定值

位	符号	位名	功能	R/W
b1-b0	—	保留位	读取值为不定值，写操作无效。	R
b3-b2	—	保留位	读取值为“0”，写操作无效。	R
b4	FLOCKST	锁定位的状态位	0: 保护状态 1: 非保护状态	R
b6-b5	—	保留位	读取值为“0”，写操作无效。	R
b7	FCUERR	FCU 错误位	0: 在 FCU 处理中未发生错误 1: 在 FCU 处理中发生错误	R

FSTATR1 寄存器是确认 FCU 状态的寄存器。

在内部 ROM 无效模式中，FSTATR1 寄存器的读数据为“00h”。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FSTATR1 寄存器进行初始化。

## FLOCKST 位（锁定位的状态位）

当使用锁定位读 2 命令时，此位反映锁定位的读数据。

在发行锁定位读 2 命令后 FSTATR0.FR DY 位变为“1”时，将有效数据保存到 FLOCKST 位。FLOCKST 位的值保持到下一个锁定位读 2 命令结束为止。

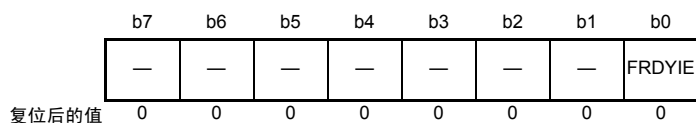
## FCUERR 位（FCU 错误位）

此位表示在 FCU 内部处理中发生错误。

当 FCUERR 位为“1”时，必须将 FRESETR.FRESET 位置“1”，对 FCU 进行初始化。必须将 FCU 固件从 FCU 固件区重新复制到 FCU 的 RAM 区。

## 31.2.7 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

FRDYIE 寄存器设定允许或者禁止闪存就绪中断（FRDYI）的输出。

在内部 ROM 无效模式中，FRDYIE 寄存器的读数据为“00h”，不能写。

通过复位对 FRDYIE 寄存器进行初始化。

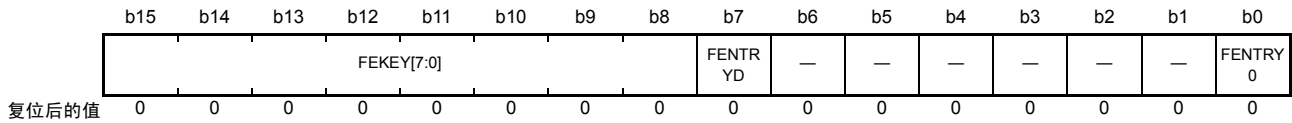
## FRDYIE 位（闪存就绪中断允许位）

此位允许或者禁止在结束编程或者擦除处理时产生 FRDYI 中断请求。

如果在将 FRDYIE 位置“1”的状态下结束 FCU 命令的执行（FSTATR0.FRDY 位从“0”变为“1”），就产生闪存就绪中断请求（FRDYI）。

## 31.2.8 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	数据闪存 P/E 模式入口位	请参照“32. 数据闪存（保存数据的闪存）”。	R/W
b15-b8	FEKEY[7:0]	键码	控制能否改写 FENTRYD 位和 FENTRY0 位。	R/(W) (注 1)

注 1. 不保持写数据。

FENTRYR 寄存器是将 ROM/ 数据闪存设定为 P/E 模式的寄存器。

为了将 ROM/ 数据闪存设定为 P/E 模式，使 FCU 能接受命令，需要将 FENTRYD 位或者 FENTRY0 位置“1”。但是，如果将这些位中的多个位置“1”，FSTATR0.ILGLERR 位就变为“1”，FCU 进入命令锁定状态。

只在给高位字节写特定值时才能通过字存取写此寄存器，而在其他情况下写此寄存器时，此寄存器就被初始化，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，FENTRYR 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FENTRYR 寄存器进行初始化。

## FENTRY0 位（ROM P/E 模式入口位 0）

此位将 ROM 设定为 P/E 模式。

[ 写操作的有效条件（满足以下全部条件时） ]

- 内部 ROM 为有效模式
- FSTATR0.FR DY 位为“1”
- 通过字存取给 FEKEY[7:0] 位写“AAh”

[ 为“1”的条件 ]

- 在满足写操作的有效条件并且 FENTRYR 寄存器为“0000h”的状态下，给 FENTRY0 位写“1”时

[ 为“0”的条件 ]

- 通过字节存取进行写操作时
- 在 FEKEY[7:0] 位不为“AAh”的状态下，通过字存取进行写操作时
- 在满足写操作的有效条件的状态下给 FENTRY0 位写“0”时
- 在满足写操作的有效条件并且 FENTRYR 寄存器不为“0000h”的状态下，写 FENTRYR 寄存器时

## FEKEY[7:0] 位（键码）

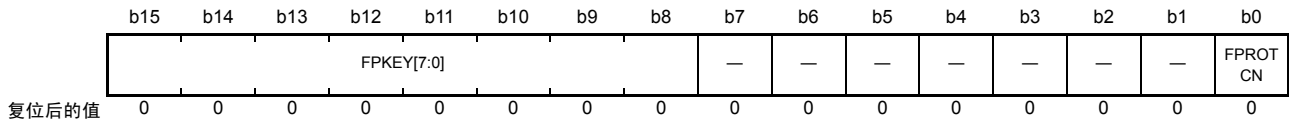
这些位控制能否改写 FENTRYD 位和 FENTRY0 位。

不保持 FEKEY[7:0] 位的写数据。



## 31.2.9 闪存保护寄存器（FPROTR）

地址 007F FFB4h



位	符号	位名	功能	R/W
b0	FPROTCN	锁定位保护取消位	0: 通过锁定位进行的保护有效 1: 通过锁定位进行的保护无效	R/W
b7-b1	—	保留位	读写值都为“0”。	R/W
b15-b8	FPKEY[7:0]	键码	控制能否改写 FPROTCN 位。	R/(W) (注1)

注 1. 不保持写数据。

FPROTR 寄存器设定通过锁定位进行的编程 / 擦除保护功能的有效或者无效。

只在给高位字节写特定值时才能通过字存取写此寄存器，而在其他情况下写此寄存器时，此寄存器就被初始化，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，FPROTR 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FPROTR 寄存器进行初始化。

## FPROTCN 位（锁定位保护取消位）

此位设定通过锁定位进行的编程 / 擦除保护的有效或者无效。

[ 为“1”的条件 ]

- 在 FENTRYR 寄存器的值不为“0000h”的状态下，通过字存取给 FPKEY[7:0]位写“55h”、给 FPROTCN 位写“1”时

[ 为“0”的条件 ]

- 通过字节存取进行写操作时
- 在 FPKEY[7:0]位不为“55h”的状态下，通过字存取进行写操作时
- 通过字存取给 FPKEY[7:0]位写“55h”、给 FPROTCN 位写“0”时
- FENTRYR 寄存器的值为“0000h”时

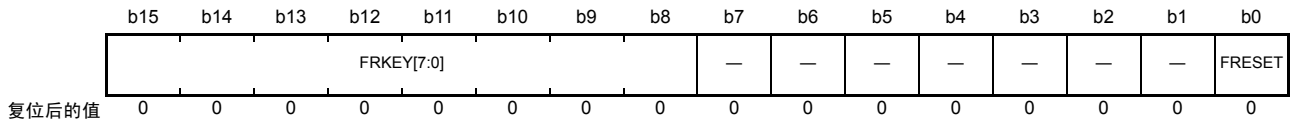
## FPKEY[7:0] 位（键码）

此位控制能否改写 FPROTCN 位。

不保持 FPKEY[7:0] 位的写数据。

## 31.2.10 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	控制能否改写 FRESET 位。	R/(W) (注1)

注 1. 不保持写数据。

FRESETR 寄存器是用于对 FCU 进行初始化的寄存器。

只在给高位字节写特定值时才能通过字存取写此寄存器，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，FRESETR 寄存器的读数据为“0000h”，不能写。

通过复位对 FRESETR 寄存器进行初始化。

## FRESET 位（闪存复位的位）

如果将 FRESET 位置“1”，ROM/数据闪存的编程或者擦除就被强制结束并且 FCU 被初始化。

给编程或者擦除过程中的 ROM/数据闪存的存储器外加高电压。为了确保存储器外加电压的下降所需的时间，在对 FCU 进行初始化时，FRESET 位为“1”的状态必须保持  $t_{RESW2}$ （参照“33. 电特性”）的时间。在 FRESET 位保持“1”的期间，必须禁止读 ROM/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。

只在 FRKEY[7:0] 位为“CCh”时才能通过字存取写 FRESET 位。

## FRKEY[7:0] 位（键码）

这些位控制能否改写 FRESET 位。

不保持 FRKEY[7:0] 位的写数据。

## 31.2.11 FCU 命令寄存器（FCMDR）

地址 007F FFBAh



位	符号	位名	功能	R/W
b7-b0	PCMDR[7:0]	预置命令	保存 FCU 接受的前一个命令。	R
b15-b8	CMDR[7:0]	命令	保存 FCU 接受的最新命令。	R

FCMDR 寄存器保存 FCU 接受的命令。

在内部 ROM 无效模式中，FCMDR 寄存器的读数据为“0000h”，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 31.4 所示，空白检查的内容请参照“32. 数据闪存（保存数据的闪存）”的“32.6 数据闪存的编程 / 擦除”。

表 31.4 接受各命令后的 FCMDR 寄存器状态

命令	CMDR	PCMDR
P/E 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/ 空白检查	D0h	71h
锁定位编程	D0h	77h

## 31.2.12 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ESUSP MD
复位后的值	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位	符号	位名	功能	R/W
b0	ESUSPMD	擦除挂起模式位	0: 挂起优先模式 1: 擦除优先模式	R/W
b15-b1	—	保留位	读写值都为“0”。	R/W

FCPSR 寄存器选择 FCU 擦除处理的挂起方法。

在内部 ROM 无效模式中，FCPSR 寄存器的读数据为“0000h”，不能写。

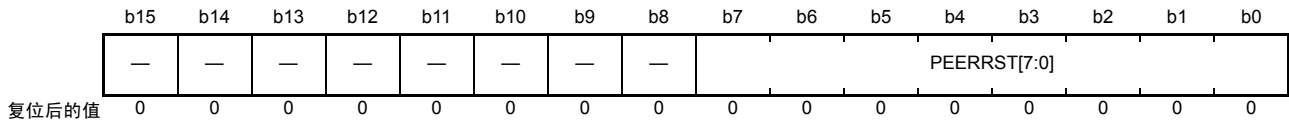
通过复位或者将 FRESETR.FRESET 位置“1”，对 FCPSR 寄存器进行初始化。

## ESUSPMD 位（擦除挂起模式位）

在 FCU 进行 ROM/ 数据闪存的擦除处理过程中发行 P/E 挂起命令时，此位选择擦除中断处理模式，详细内容请参照“31.7 挂起”。

## 31.2.13 闪存 P/E 状态寄存器（FPESTAT）

地址 007F FFCh



位	符号	位名	功能	R/W
b7-b0	PEERRST[7:0]	P/E 错误状态位	01h: 锁定位保护区的编程错误 02h: 由锁定位保护以外的原因引起的编程错误 11h: 锁定位保护区的擦除错误 12h: 由锁定位保护以外的原因引起的擦除错误 上述以外: 保留	R
b15-b8	—	保留位	读取值为“0”，写操作无效。	R

FPESTAT 寄存器是表示 ROM/ 数据闪存的编程或者擦除处理结果的寄存器。

在内部 ROM 无效模式中，FPESTAT 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 FPESTAT 寄存器进行初始化。

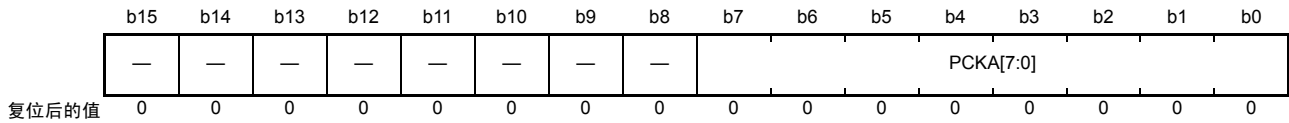
## PEERRST[7:0] 位（P/E 错误状态位）

这些位表示在 ROM/ 数据闪存的编程或者擦除处理过程中发生错误时的错误原因。

只在 FSTATR0.ERSERR 位或者 FSTATR0.PRGERR 位为“1”的状态下，PEERRST[7:0] 位的值才有效。当 ERSERR 位和 PRGERR 位为“0”时，PEERRST[7:0] 位保持过去发生的错误原因的值。

## 31.2.14 外围时钟通知寄存器（PCKAR）

地址 007F FFE8h



位	符号	位名	功能	R/W
b7-b0	PCKA[7:0]	外围时钟通知位	在对 ROM/ 数据闪存进行编程或者擦除时，这些位设定外围时钟（PCLK）。	R/W
b15-b8	—	保留位	读写值都为“0”。	R/W

在对 ROM/ 数据闪存进行编程或者擦除时，PCKAR 寄存器将外围时钟（PCLK）的频率设定信息通知定序器。此设定用于控制编程或者擦除的时间。

在内部 ROM 无效模式中，PCKAR 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESET 位置“1”，对 PCKAR 寄存器进行初始化。

## PCKA[7:0] 位（外围时钟通知位）

在对 ROM/ 数据闪存进行编程或者擦除时，这些位设定外围时钟（PCLK）。

必须在编程或者擦除前给 PCKA[7:0] 位设定 PCLK 的频率，然后发行外围时钟通知命令。不能在对 ROM/ 数据闪存进行编程或者擦除过程中更改频率。

如下进行设定值的计算：

- 将以 MHz 为单位表现的工作频率转换为 2 进制数并且写到 PCKA[7:0] 位。  
以外围时钟的工作频率为 35.9MHz 时的情况为具体例子，说明如下：
- 将 35.9 进行舍入。
- 将 36 转换为 2 进制数，给 PCKA[7:0] 位的高位设定“00h”、低位设定“24h”（0010 0100b）。

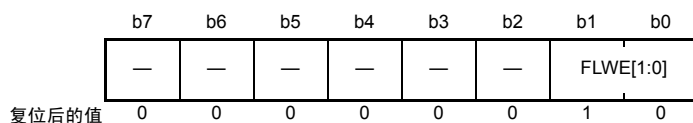
注 1. 如果将 PCKA[7:0] 位设定在 8MHz ~ 50MHz 的范围以外，就不能对 ROM/ 数据闪存发行改写命令。

注 2. 如果给 PCKA[7:0] 位设定和实际频率不同的频率，就可能破坏 ROM/ 数据闪存的数据。

注 3. 即使有效地利用 PCKA[7:0] 位，在一定程度上改写时间也取决于频率。

## 31.2.15 闪存编程 / 擦除保护寄存器（FWEPROR）

地址 0008 C289h



位	符号	位名	功能	R/W
b1-b0	FLWE[1:0]	闪存编程 / 擦除位	b1 b0 0 0: 不能进行编程和擦除 0 1: 能进行编程和擦除 1 0: 不能进行编程和擦除（初始值） 1 1: 不能进行编程和擦除	R/W
b7-b2	—	保留位	读写值都为“0”。	R/W

FWEPROR 寄存器是通过软件进行闪存编程 / 擦除保护的读写寄存器。  
在软件待机模式或者深度软件待机模式中，也对 FWEPROR 寄存器进行初始化。

## FLWE[1:0] 位（闪存编程 / 擦除位）

此位通过软件进行闪存编程 / 擦除的保护。

### 31.3 ROM 的存储器 MAT 结构

RX62T 群的 ROM 由最多 256K 字节的用户 MAT 构成，这些 MAT 的地址如图 31.2 所示。  
 必须注意：在进行读、编程或者擦除时，用户 MAT 的地址不同。

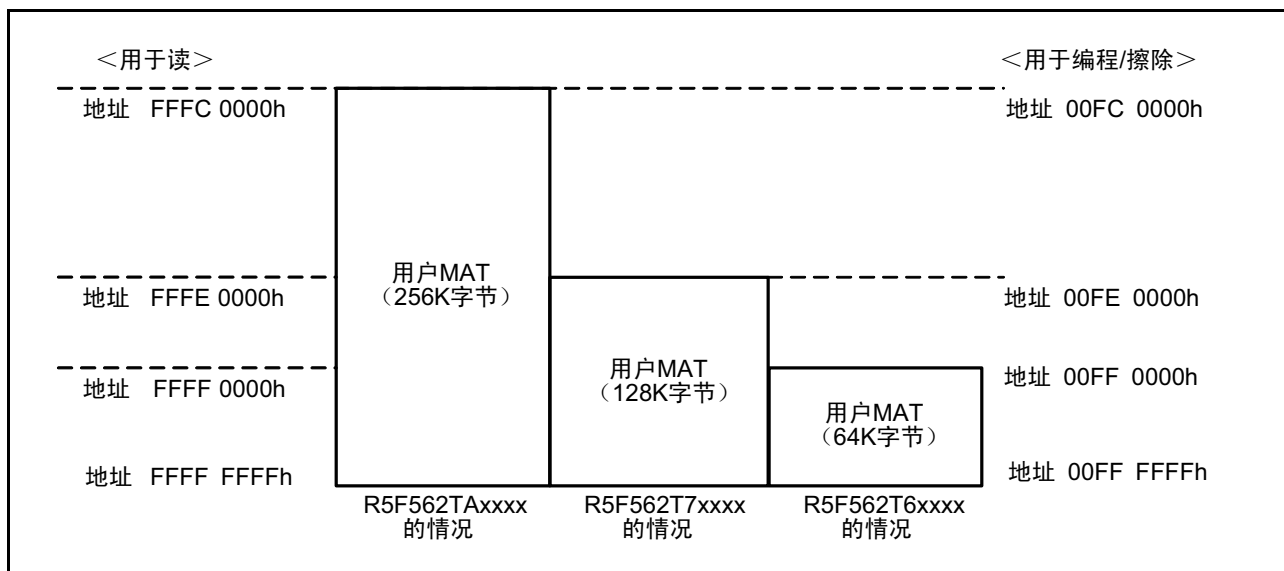


图 31.2 ROM 的存储器 MAT 结构

### 31.4 块结构

用户 MAT 的擦除块结构如图 31.3 所示。用户 MAT 被分割为 4K 字节（8 块）和 16K 字节（14 块），以块为单位进行擦除。以低位地址为“00h”开始的 256 字节为单位进行编程。

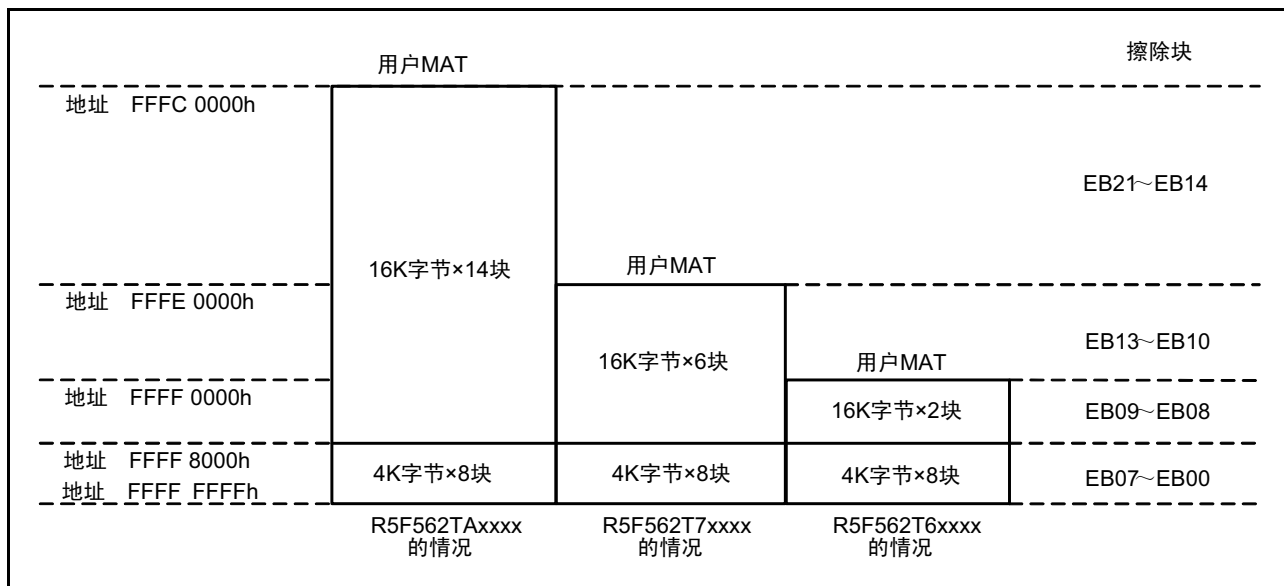


图 31.3 用户 MAT 的擦除块结构



### 31.5 ROM 相关的运行模式

RX62T 群的运行模式转移图如图 31.4 所示。

一旦设定 MD1 引脚和 MD0 引脚并且进行复位解除，就进行如图 31.4 的转移。

有关 MD1 引脚和 MD0 引脚的设定值与 RX62T 群的运行模式的关系，请参照“3. 运行模式”。

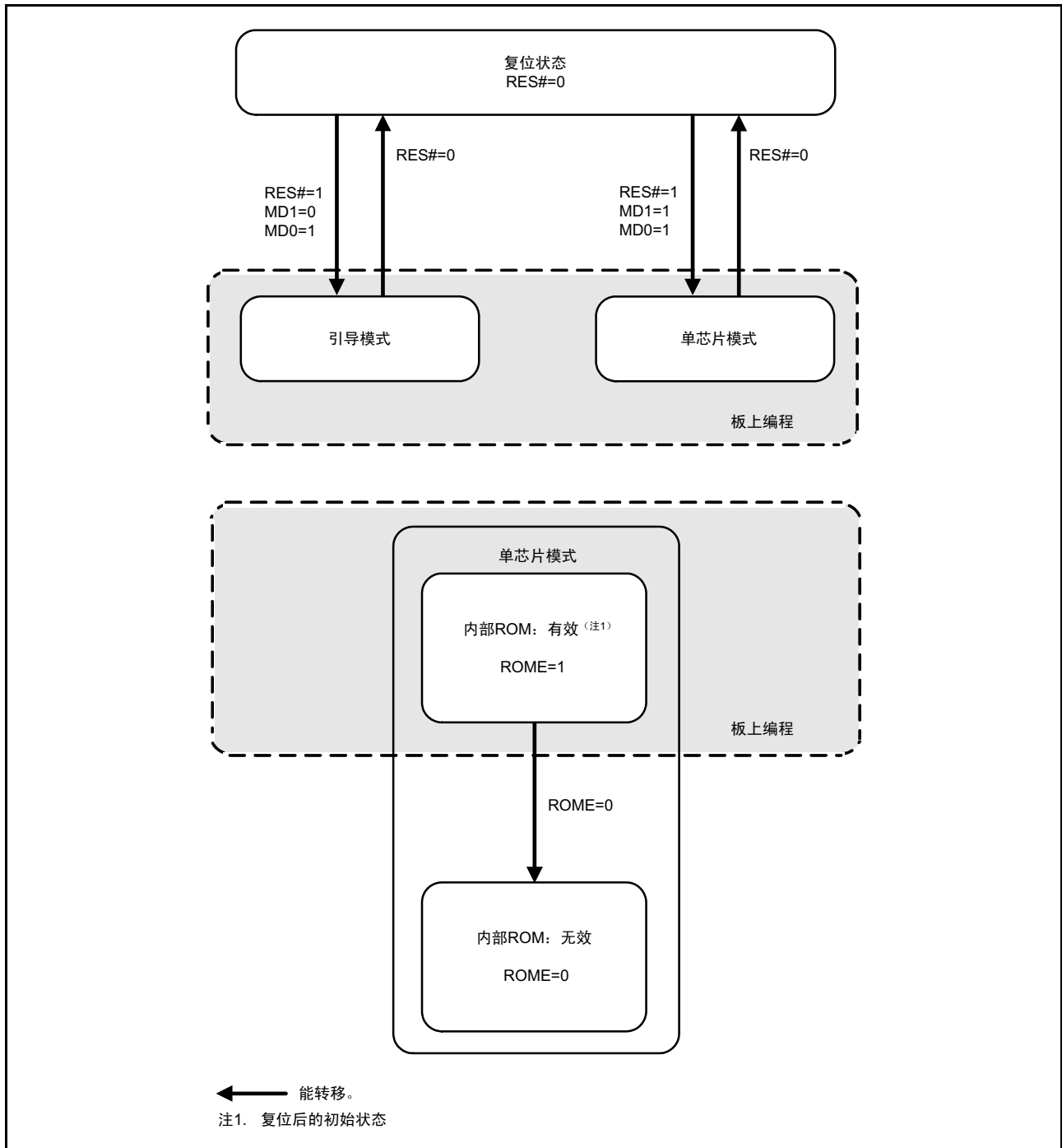


图 31.4 ROM 相关的运行模式转移图

在引导模式、单芯片模式（内部 ROM 有效）中，能进行板上 ROM 的读、编程或者擦除。

在各模式中，能编程和擦除的 MAT、复位时的启动 MAT 等不同，各模式的不同点如表 31.5 所示。

表 31.5 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 / 擦除的 MAT	用户 MAT	用户 MAT
块分割擦除	○（注 1）	○
复位时的启动 MAT	嵌入式程序存储 MAT（注 2）	用户 MAT

注 1. 有可能在启动时被全部擦除。此后能擦除特定块，详细内容请参照“31.9.2 ID 码保护”。

注 2. 用户不能使用。

- 在引导模式中，主机能经由 SCI 对用户 MAT/数据 MAT 进行编程和读。
- 在引导模式中，通过引导模式的嵌入式程序使用内部 RAM。因此，不保持内部 RAM 的数据。

### 31.6 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。

#### 31.6.1 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 31.5 所示。

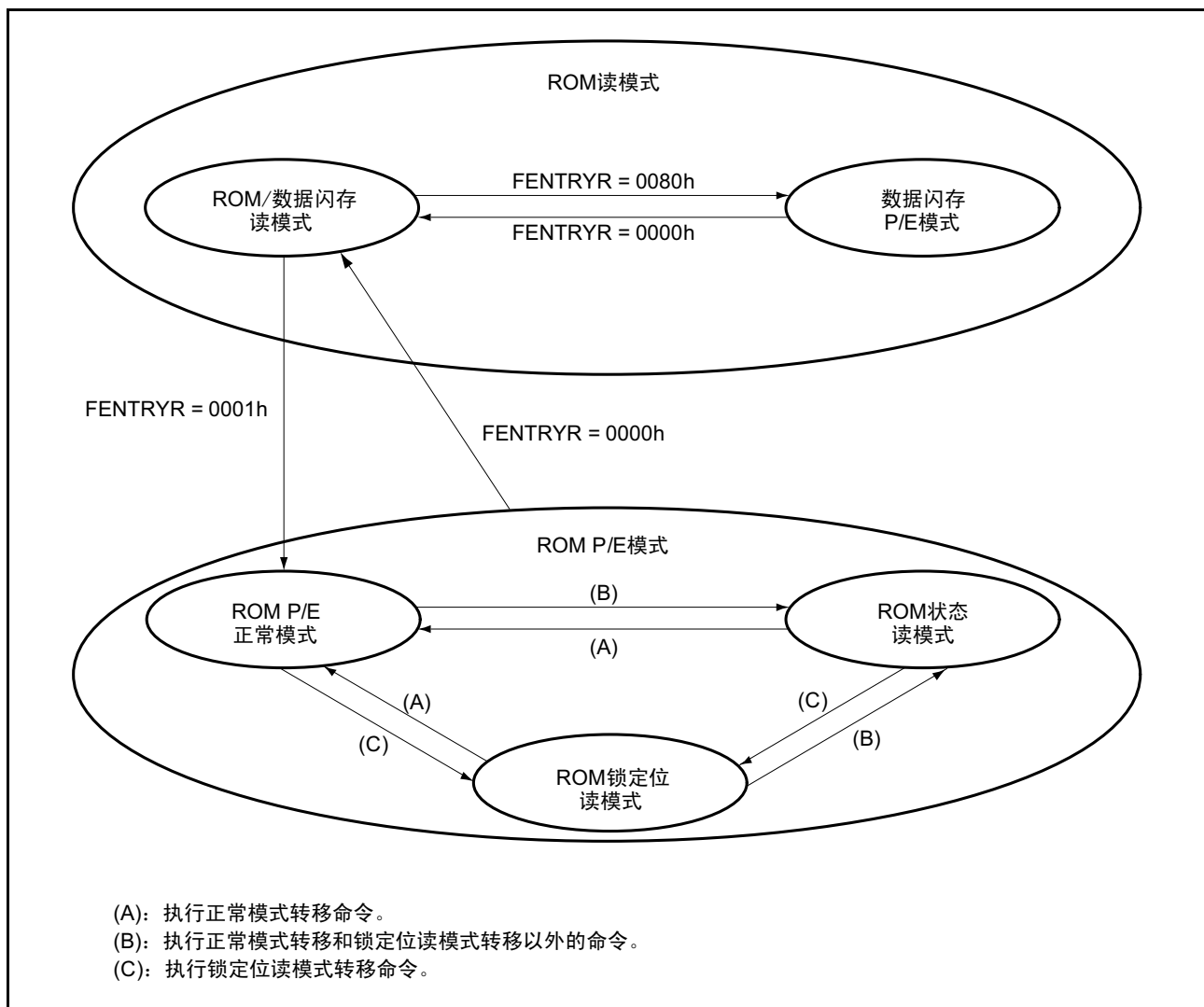


图 31.5 FCU 的模式转移图（ROM 相关）

### 31.6.1.1 ROM 读模式

ROM 读模式是高速读 ROM 的模式。如果对读地址进行读存取，就能进行 1 个 ICLK 周期的高速读取。ROM 读模式有 ROM/ 数据闪存读模式和数据闪存 P/E 模式。

#### (1) ROM/ 数据闪存读模式

ROM/ 数据闪存读模式是能读 ROM 和数据闪存的模式，不接受 FCU 命令。如果将 FENTRYR.FENTRY0 位和 FENTRYR.FENTRYD 位都置“0”，就转移到此模式。

#### (2) 数据闪存 P/E 模式

数据闪存 P/E 模式是对数据闪存进行编程和擦除的模式，能高速读 ROM。此模式接受数据闪存的 FCU 命令，但是不接受 ROM 的 FCU 命令。如果将 FENTRYR.FENTRY0 位置“0”并且将 FENTRYR.FENTRYD 位置“1”，就转移到此模式。

有关数据闪存 P/E 模式的详细内容，请参照“32. 数据闪存（保存数据的闪存）”的“32.6.1 FCU 的模式”。

### 31.6.1.2 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“31.8.2 错误保护”）。

ROM P/E 模式有 ROM P/E 正常模式、ROM 状态读模式和 ROM 锁定位读模式共 3 种。

#### (1) ROM P/E 正常模式

ROM P/E 正常模式是在对 ROM 进行编程或者擦除时最先转移的模式。如果在 ROM 读模式中将 FENTRYR.FENTRYD 位置“0”并且将 FENTRYR.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 31.8 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“31.8.2 错误保护”）。

#### (2) ROM 状态读模式

ROM 状态读模式是能读 ROM 状态的模式。如果在 ROM P/E 模式中接受正常模式转移和锁定位读模式转移以外的命令，就转移到此模式。

FSTATR0.FRDY 位为“0”的状态以及发生错误后的命令锁定状态也是 ROM 状态读模式中的状态。能接受的命令如表 31.8 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 31.8 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。

### 31.6.2 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 31.6 所示。

表 31.6 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“31.6.3 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“31.6.3 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“31.6.3 FCU 的模式和命令的关系”）。
外围时钟通知	设定外围时钟的频率。
编程	对 ROM 进行编程（以 256 字节为单位）。
块擦除	擦除 ROM（以块为单位，同时擦除锁定位）。
P/E 挂起	中止编程和擦除。
P/E 恢复	重新开始编程和擦除。
状态寄存器清除	清除 FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位和 FSTATR0.PRGERR 位以及解除命令锁定状态。
锁定位读 2/ 空白检查	读指定的擦除块的锁定位（将锁定位反映到 FSTATR1.FLOCKST 位）/ 数据闪存的空白检查。
锁定位编程	对指定的擦除块的锁定位进行编程。

锁定位读 2 命令兼用数据闪存的空白检查命令。如果对数据闪存发行锁定位读 2 命令，就进行数据闪存的空白检查（参照“32. 数据闪存（保存数据的闪存）”）。

通过写 ROM 的编程 / 擦除地址给 FCU 发行命令。FCU 命令的格式如表 31.7 所示。如果在 FCU 的特定条件下进行表 31.7 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“31.6.3 FCU 的模式和命令的关系”和“31.6.4 FCU 命令的使用方法”。

表 31.7 FCU 命令的格式

命令	总线周期数	第 1 周期		第 2 周期		第 3 周期		第 4 ~ 第 5 周期		第 6 周期		第 7 ~ 第 130 周期		第 131 周期	
		地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据
P/E 正常模式转移	1	RA	FFh	—	—	—	—	—	—	—	—	—	—	—	—
状态读模式转移	1	RA	70h	—	—	—	—	—	—	—	—	—	—	—	—
锁定位读模式转移（锁定位读 1）	1	RA	71h	—	—	—	—	—	—	—	—	—	—	—	—
外围时钟通知	6	RA	E9h	RA	03h	RA	0F0h	RA	0F0h	RA	D0h	—	—	—	—
编程	131	RA	E8h	RA	80h	WA	WDn	RA	WDn	RA	WDn	RA	WDn	RA	D0h
块擦除	2	RA	20h	BA	D0h	—	—	—	—	—	—	—	—	—	—
P/E 挂起	1	RA	B0h	—	—	—	—	—	—	—	—	—	—	—	—
P/E 恢复	1	RA	D0h	—	—	—	—	—	—	—	—	—	—	—	—
状态寄存器清除	1	RA	50h	—	—	—	—	—	—	—	—	—	—	—	—
锁定位读 2	2	RA	71h	BA	D0h	—	—	—	—	—	—	—	—	—	—
锁定位编程	2	RA	77h	BA	D0h	—	—	—	—	—	—	—	—	—	—

- 地址列 RA: ROM 的编程 / 擦除地址  
当 FENTRYR.FENTRY0 位为“1”时: 00FC 0000h ~ 00FF FFFFh 的任意地址
- WA: ROM 的编程目标地址  
编程数据 256 字节的起始地址
- BA: ROM 的擦除块地址  
擦除对象块内的任意地址（由编程 / 擦除地址指定）
- 数据列 WDn: 编程数据的第 n 个字（n=1 ~ 128）

### 31.6.3 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 31.8 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“31.8.2 错误保护”）。

必须在转移到能接受的模式后，确认 FSTATR0.FRDY 位、FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位和 FSTATR1.FCUEERR 位的值，然后发行 FCU 命令。能根据 FASTAT.CMDLK 位的值确认有无发生错误。FASTAT.CMDLK 位的值是 FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位和 FSTATR1.FCUEERR 位的值的逻辑或。

表 31.8 FCU 的模式 / 状态和能接受的命令的关系（ROM P/E 模式）

命令	P/E 正常模式			状态读模式						锁定位读模式			
	编程挂起中	擦除挂起中	其他状态	编程 / 擦除处理中	编程 / 擦除中止处理中	锁定位读 2 处理中	编程挂起中	擦除挂起中	命令锁定状态	其他状态	编程挂起中	擦除挂起中	其他状态
FSTATR0.FRDY 位	1	1	1	0	0	0	1	1	0/1	1	1	1	1
FSTATR0.SUSRDY 位	0	0	0	1	0	0	0	0	0	0	0	0	0
FSTATR0.ERSSPD 位	0	1	0	0	0/1	0	0	1	0	0	0	1	0
FSTATR0.PRGSPD 位	1	0	0	0	0/1	0	1	0	0	0	1	0	0
FASTAT.CMDLK 位	0	0	0	0	0	0	0	0	1	0	0	0	0
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	○	○	○	×	×	×	○	○	○	○	○	○	○
锁定位读 2	○	○	○	×	×	×	○	○	×	○	○	○	○
锁定位编程	×	△	○	×	×	×	×	△	×	○	×	△	○

○：能接受，△：只能接受对非擦除中止块进行的编程，×：不能接受

### 31.6.4 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“31.6.3 FCU 的模式和命令的关系”。

#### 31.6.4.1 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 31.5。

##### (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，必须以字节为单位将“01h”写到 FWEPROR 寄存器，进入能进行编程和擦除的状态（参照“31.2.15 闪存编程 / 擦除保护寄存器（FWEPROR）”）。

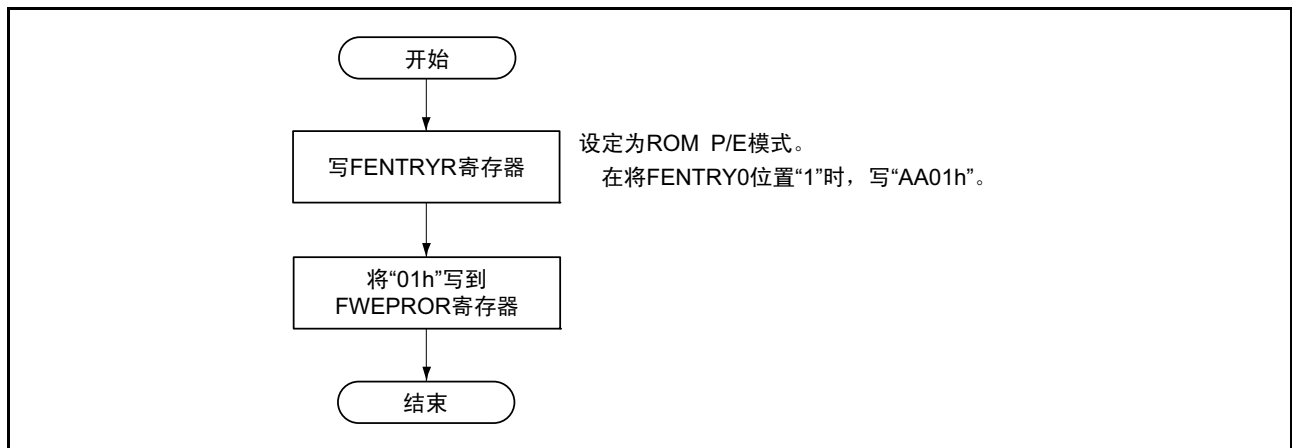


图 31.6 ROM P/E 模式的转移流程



(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。

还需要通过字节存取将“02h”写到 FWEPROR 寄存器，进入不能编程和擦除的状态（参照“31.2.15 闪存编程 / 擦除保护寄存器（FWEPROR）”）。

必须在结束 FCU 的命令处理并且 FCU 未检测到错误的状态下从 ROM P/E 模式转移到 ROM 读模式。

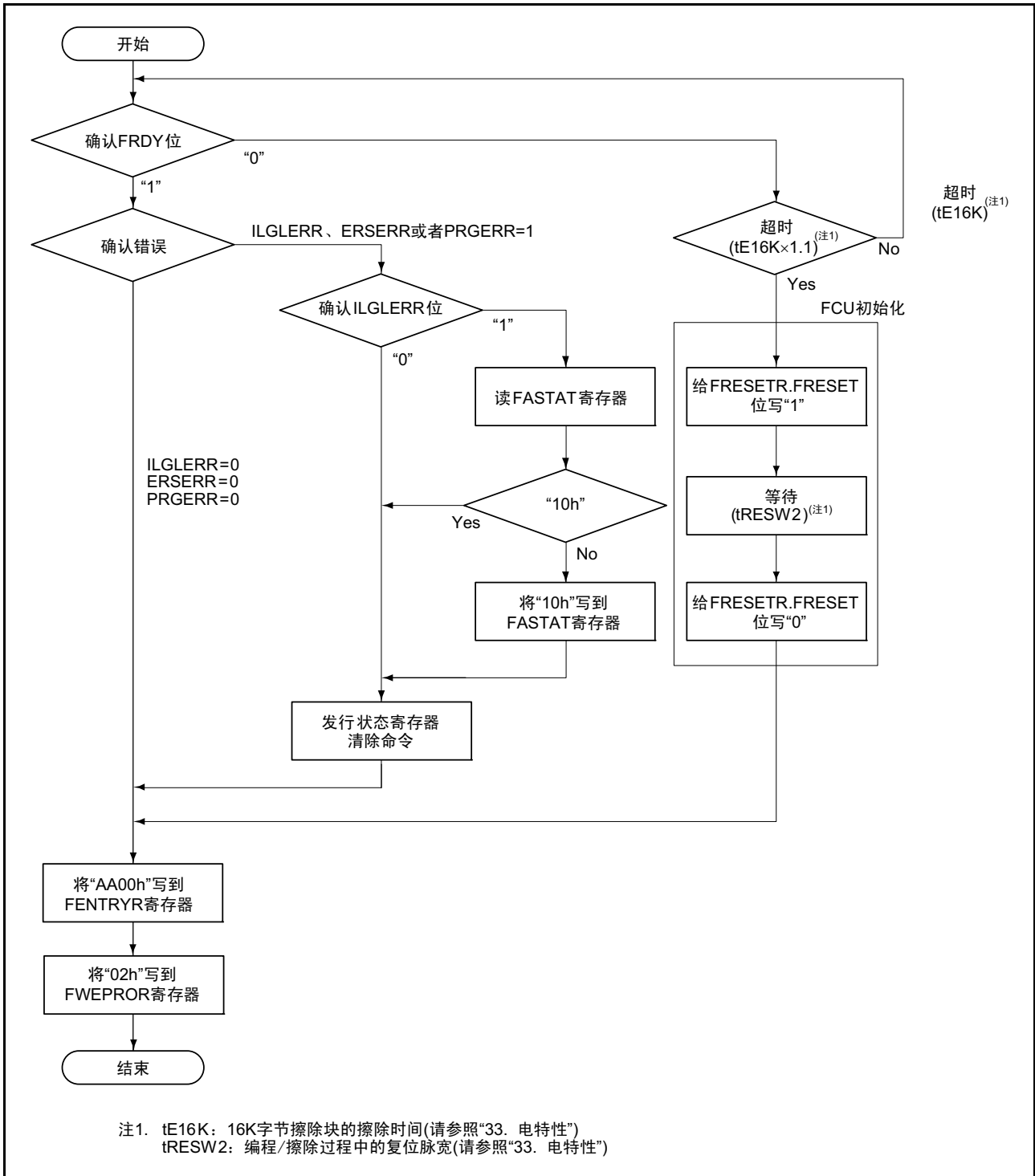


图 31.7 ROM 读模式的转移流程

### (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“31.6.1 FCU 的模式”）以及在 ROM P/E 模式中发行正常模式转移命令（图 31.8）的方法。通过字节存取将“FFh”写到 ROM 的编程 / 擦除地址，执行正常模式转移命令。

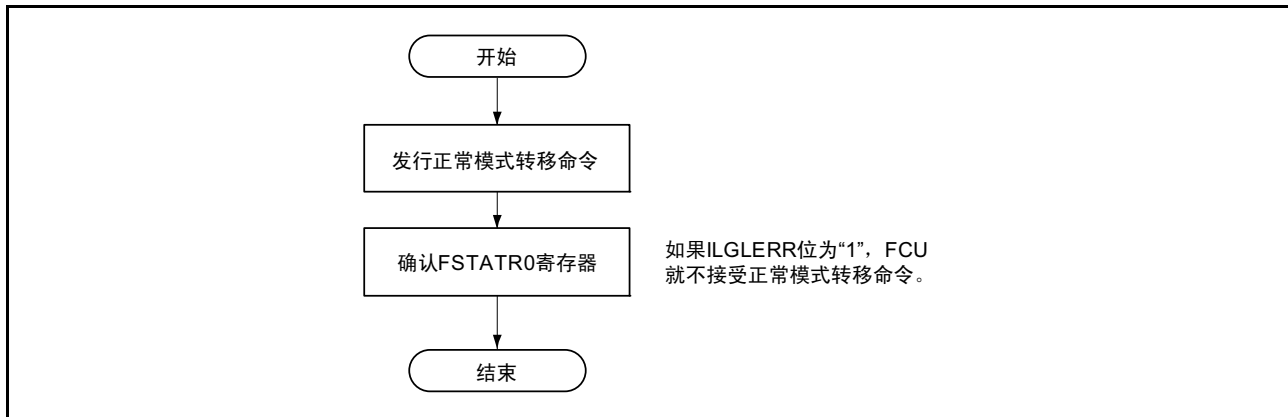


图 31.8 ROM P/E 正常模式的转移流程

### (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 31.9 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。

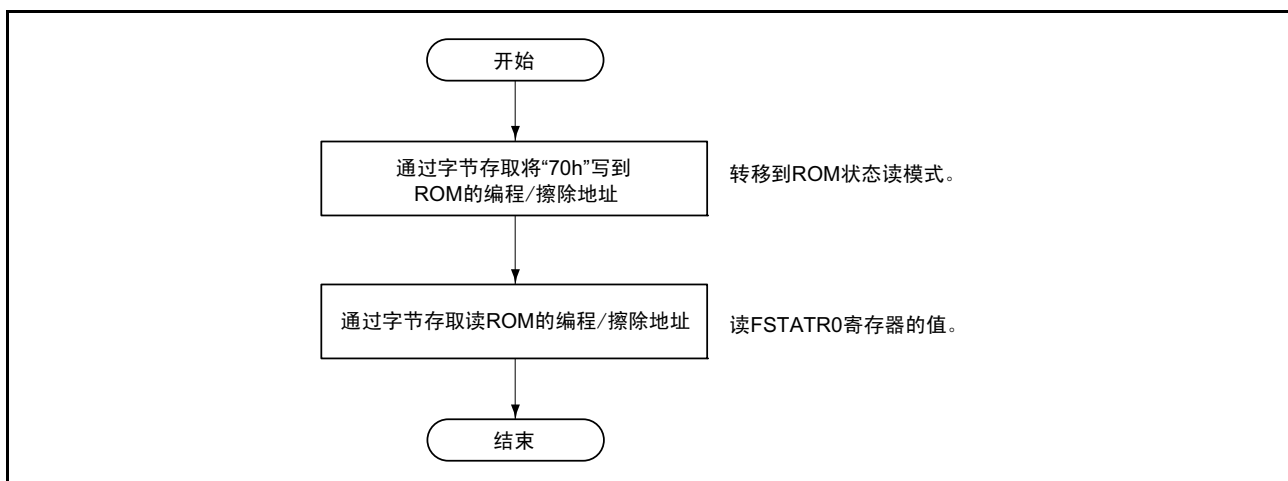


图 31.9 ROM 状态读模式的转移流程以及状态的确认方法

## (5) ROM 锁定位读模式的转移方法

在 FMODR.FRDM 位为“0”（存储区读方式）时，通过发行锁定位读模式转移命令（锁定位读 1），转移到 ROM 锁定位读模式。如果在转移到 ROM 锁定位读模式后读 ROM 的编程/擦除地址，就读与存取对象对应的擦除块的锁定位并且复制到读数据的全部位（图 31.10）。

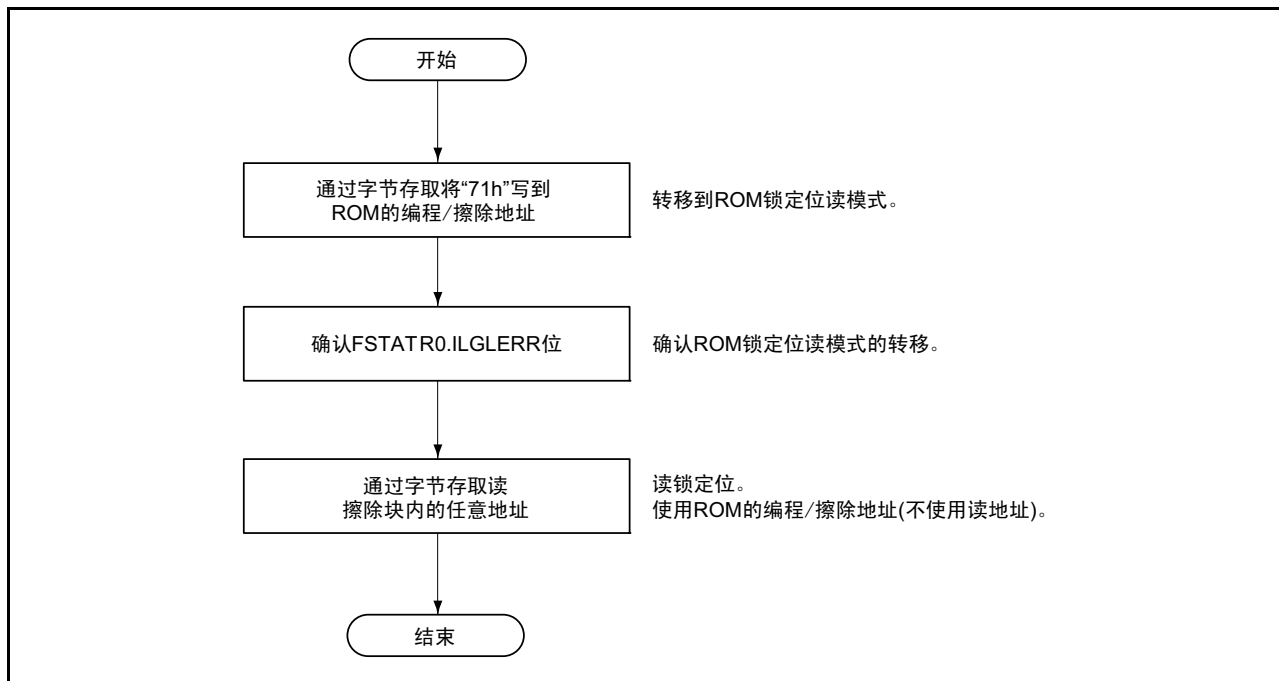


图 31.10 ROM 锁定位读模式的转移流程以及锁定位的读法

31.6.4.2 编程 / 擦除的步骤

在此说明 ROM 的编程 / 擦除流程。有关 FCU 命令的接受条件，请参照“31.6.3 FCU 的模式和命令的关系”。FCU 命令的概略流程如图 31.11 所示。

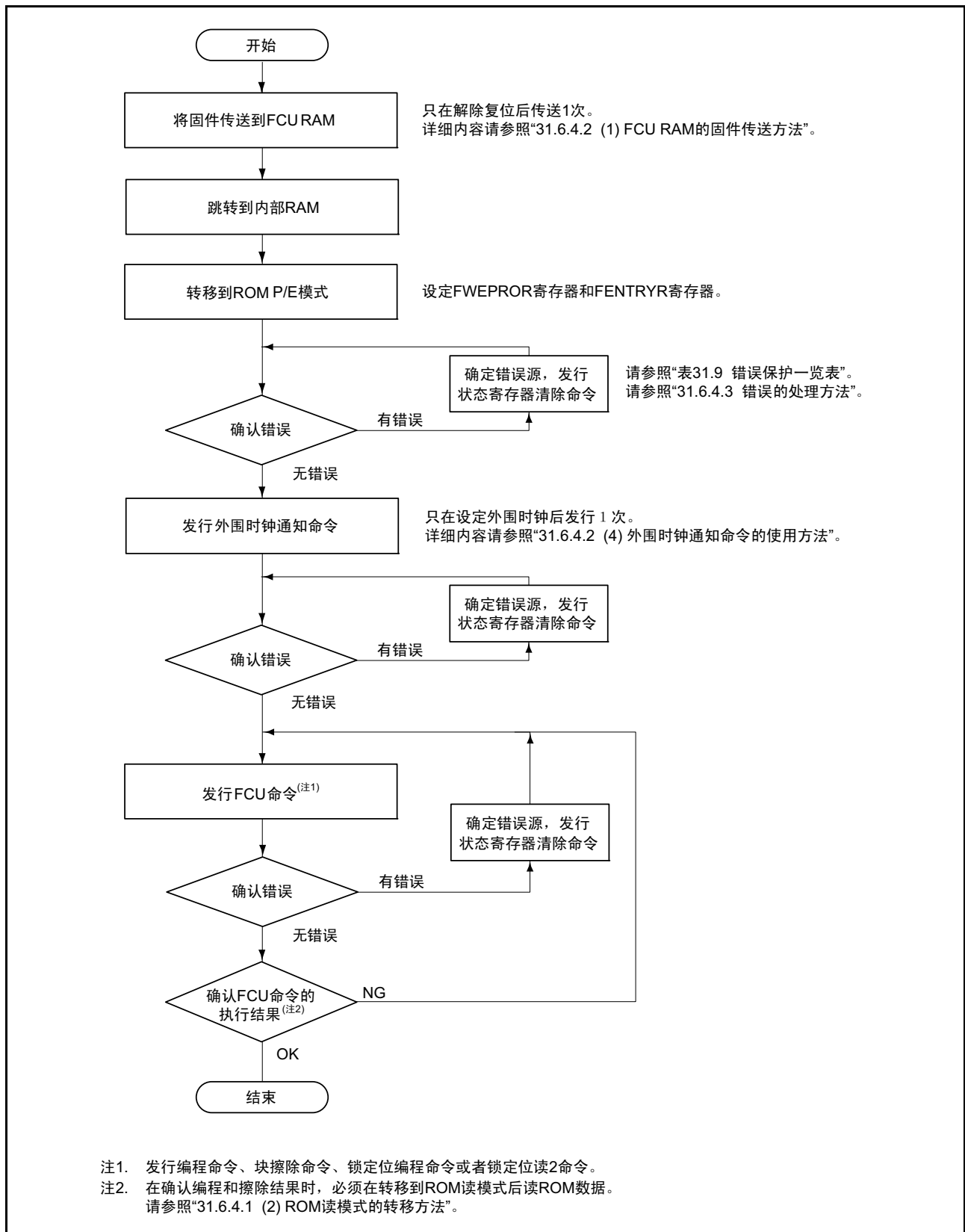


图 31.11 编程 / 擦除处理的概略流程

### (1) FCU RAM 的固件传送方法

要使用 FCU 命令时，需要将 FCU 的固件保存到 FCU RAM。因为在启动芯片时没有将 FCU 的固件保存到 FCU RAM，所以需要将在 FCU 固件区的 FCU 固件复制到 FCU RAM。当 FSTATR1.FCUEERR 位为“1”时，保存在 FCU RAM 的固件就可能被破坏，因此需要对 FCU 进行复位并且重新复制 FCU 固件。

FCU RAM 的固件传送流程如图 31.12 所示。在给 FCU RAM 写数据时，必须将 FENTRYR 寄存器置“0000h”，使 FCU 停止。

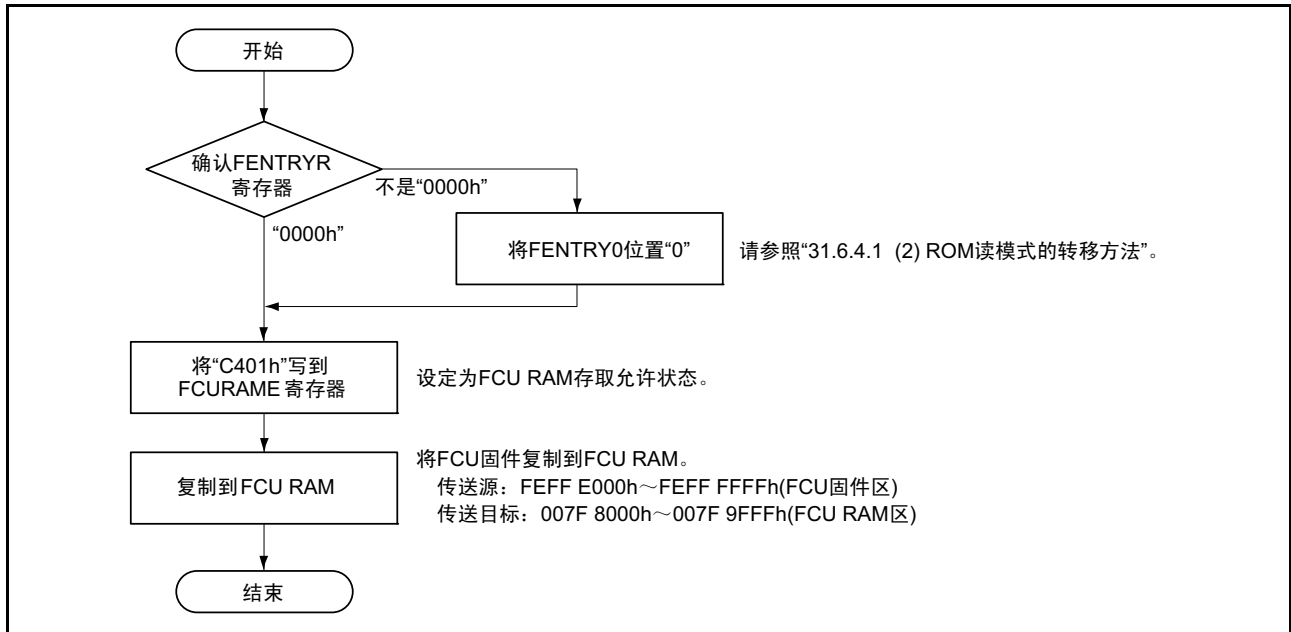


图 31.12 FCU RAM 的固件传送流程

### (2) 向内部 RAM 的跳转

在对 ROM 进行编程或者擦除时，因为不对 ROM 取指令，所以需要转移到 ROM 以外的区域。必须在将所需的指令码复制到内部 RAM 后跳转到内部 RAM。

### (3) ROM P/E 模式的转移

需要设定 FENTRYR.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“31.6.4.1 (1) ROM P/E 模式的转移方法”。

### (4) 外围时钟通知命令的使用方法

需要给 PCKAR 寄存器设定在对 ROM 进行编程或者擦除前使用的外围时钟的频率。能设定的频率范围是 8 ~ 50MHz。不能进行超出此范围的设定。

在设定 PCKAR 寄存器后使用外围时钟通知命令。在外围时钟通知命令的第 1 周期通过字节存取将“E9h”写到 ROM 的编程 / 擦除地址，在第 2 周期，通过字节存取将“03h”写到 ROM 的编程 / 擦除地址。在命令的第 3 周期~第 5 周期，通过字存取进行写操作。此时，起始地址必须使用调整为以 4 字节为边界的地址。在通过 3 次字存取将数据“0F0Fh”写到 ROM 的编程 / 擦除地址后，如果在第 6 周期通过字节存取将“D0h”写到 ROM 的编程 / 擦除地址，FCU 就开始进行外围时钟的频率设定处理。能通过 FSTATR0.FRDY 位确认设定的结束。

第 1 周期~第 6 周期中能指定的地址因 FENTRYR.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“31.8.2 错误保护”）。

如果在复位后不更改正在使用的外围时钟的设定，此设定就只需执行 1 次而且对后续的 FCU 命令有效。

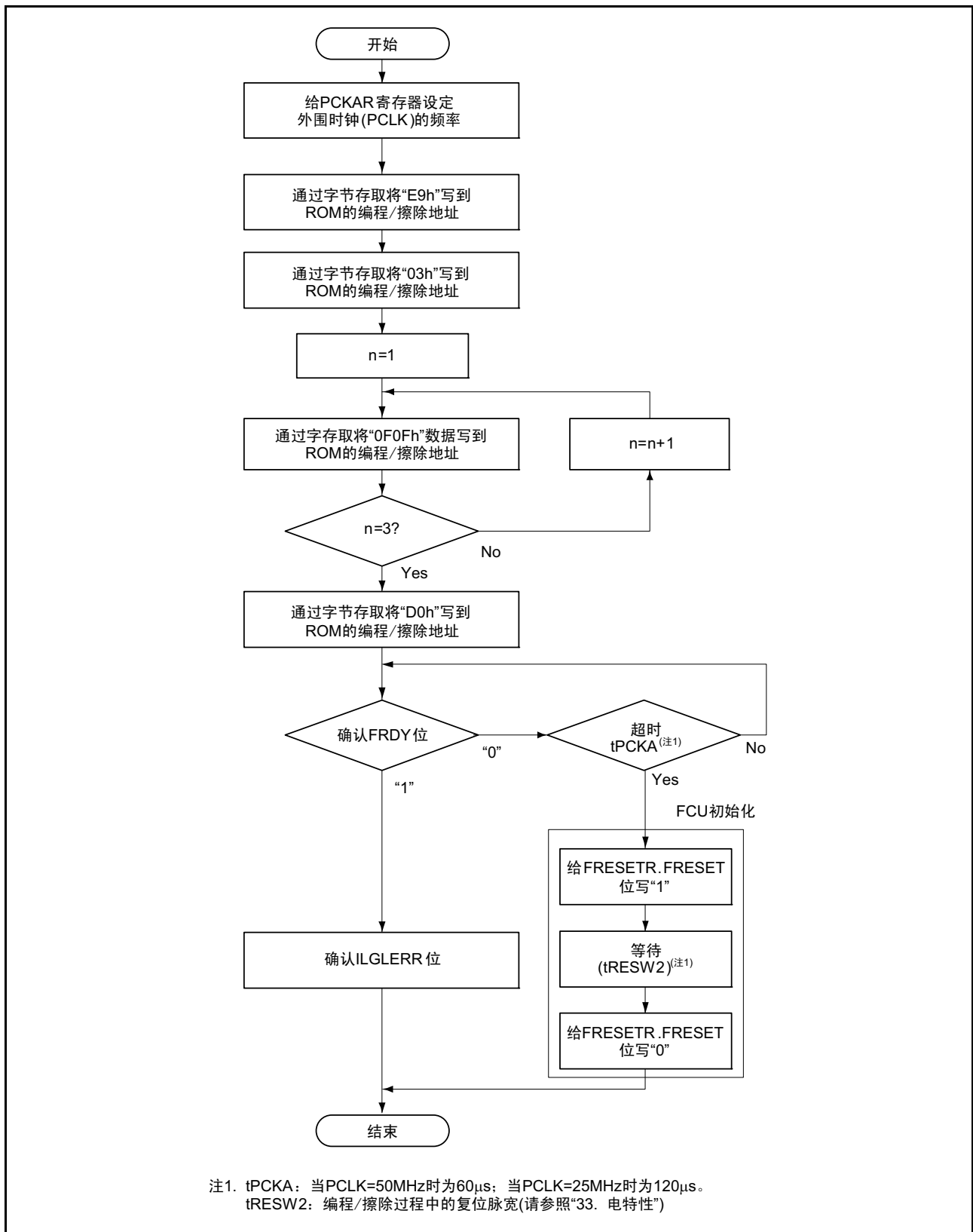


图 31.13 外围时钟通知命令的使用方法

### (5) 编程方法

使用编程命令对 ROM 进行数据编程。

在编程命令的第 1 周期，通过字节存取将“E8h”写到 ROM 的编程 / 擦除地址，在第 2 周期，通过字节存取将“80h”写到 ROM 的编程 / 擦除地址。在第 3 周期的存取中，必须通过字存取将编程数据写到编程对象区的起始地址。此时，起始地址必须使用调整为以 256 字节为边界的地址。在第 4 周期～第 130 周期，必须通过 127 次字存取将编程数据写到 ROM 的编程 / 擦除地址。如果在第 131 周期通过字节存取将“D0h”写到 ROM 的编程 / 擦除地址，FCU 就开始对 ROM 进行编程处理。能通过 FSTATR0.FRDIY 位确认编程的结束。

第 1 周期～第 131 周期中能指定的地址因 FENTRYR.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“31.8.2 错误保护”）。

如果在第 3 周期～第 130 周期中存取的区域包括不需要编程的地址，就必须将该地址的编程数据置“FFFFh”。如果要在将通过锁定位进行的保护设定为无效后进行编程，就必须将 FPROTR.FPROTCN 位置“1”。

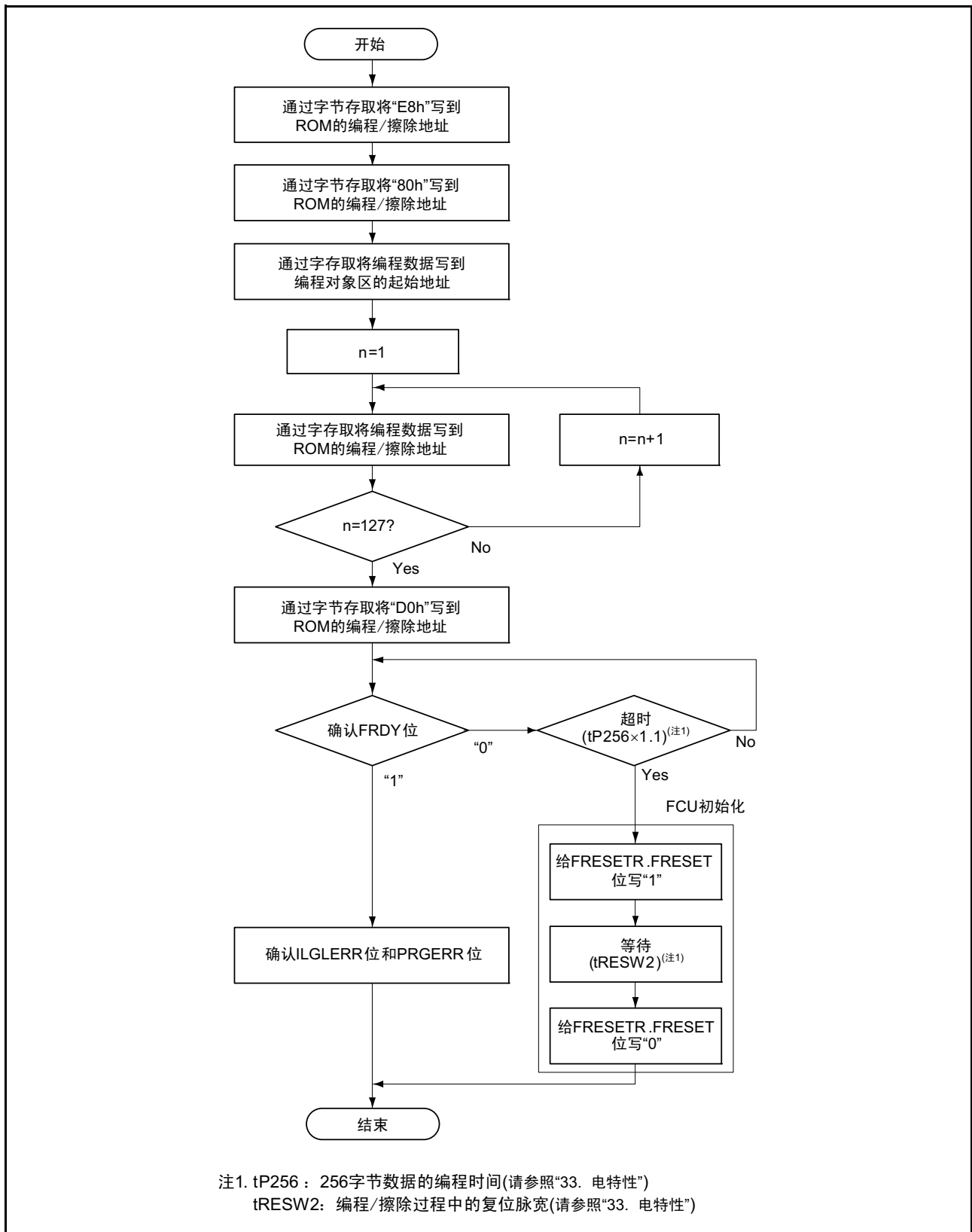


图 31.14 ROM 的编程方法



## (6) 擦除方法

使用块擦除命令来擦除 ROM。

在块擦除命令的第 1 周期，通过字节存取将“20h”写到 ROM 的编程 / 擦除地址。如果在第 2 周期通过字节存取将“D0h”写到擦除对象块内的任意地址，FCU 就开始对 ROM 进行擦除处理。能通过 FSTATR0.FRDY 位确认擦除的结束。如果通过 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，就必须将 FPROTR.FPROTCN 位置“1”。

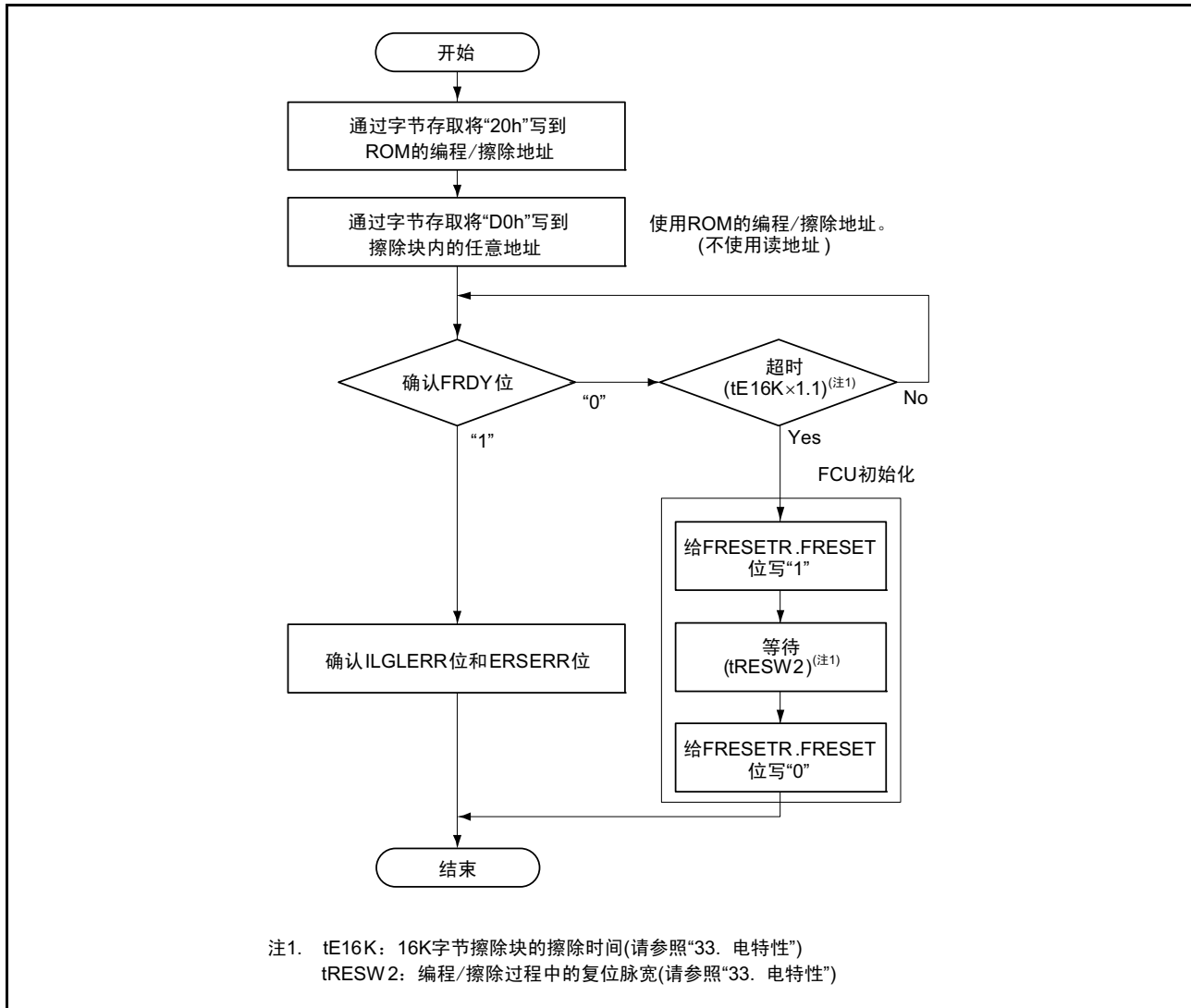


图 31.15 ROM 的擦除方法

(7) 锁定位的编程 / 擦除方法

用户 MAT 的各擦除块内置锁定位。使用锁定位编程命令对锁定位进行编程。在锁定位编程命令的第 1 周期，通过字节存取将“77h”写到 ROM 的编程 / 擦除地址。如果在第 2 周期通过字节存取将“D0h”写到要进行锁定位编程的擦除块内的任意地址，FCU 就开始进行锁定位的编程处理。能通过 FSTATR0.FRDY 位确认编程的结束。

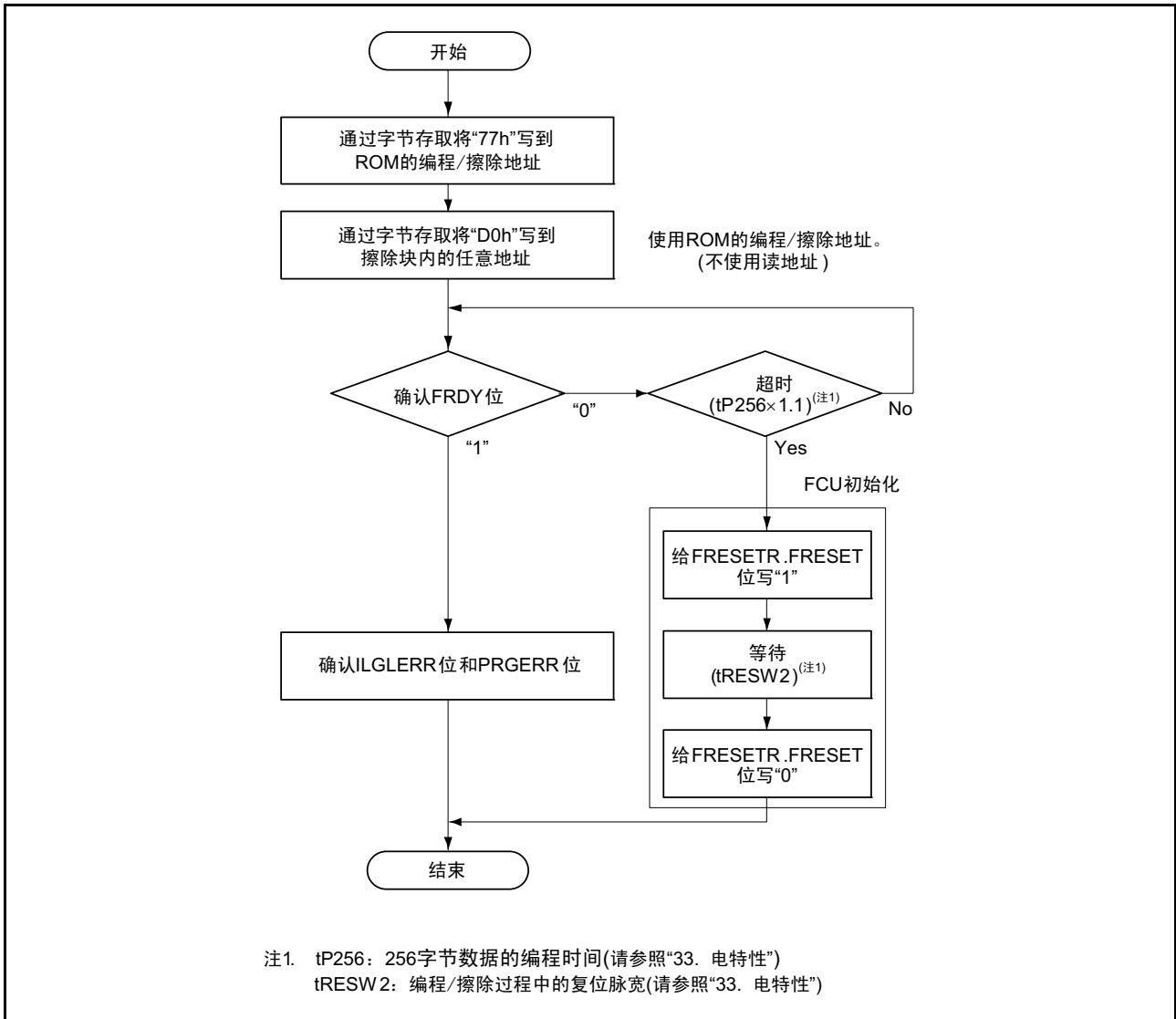


图 31.16 锁定位的编程设定方法

使用块擦除命令来擦除锁定位。

在 FPROTR.FPROTCN 位为“0”的状态下，不能擦除锁定位为“0”的擦除块。要擦除锁定位时，必须在将 FPROTCN 位置“1”的状态下发行块擦除命令。如果使用块擦除命令，擦除块内的全部数据就被擦除，而不能只擦除锁定位。

## (8) 锁定位的读法

锁定位的读法有存储区读方式和寄存器读方式。

寄存器读方式（FMODR.FRDM位为“1”）使用锁定位读2命令，对要读锁定位的擦除块的编程/擦除地址发行锁定位读2命令。如果在锁定位读2命令的第1周期，通过字节存取写“71h”，在第2周期，通过字节存取写“D0h”，就将对应的擦除块的锁定位复制到FSTATR1.FLOCKST位。

通过转移到锁定位读模式并且读ROM的编程/擦除地址来执行存储区读方式（FMODR.FRDM位为“0”）。有关详细内容请参照“31.6.4.1 (5) ROM 锁定位读模式的转移方法”。

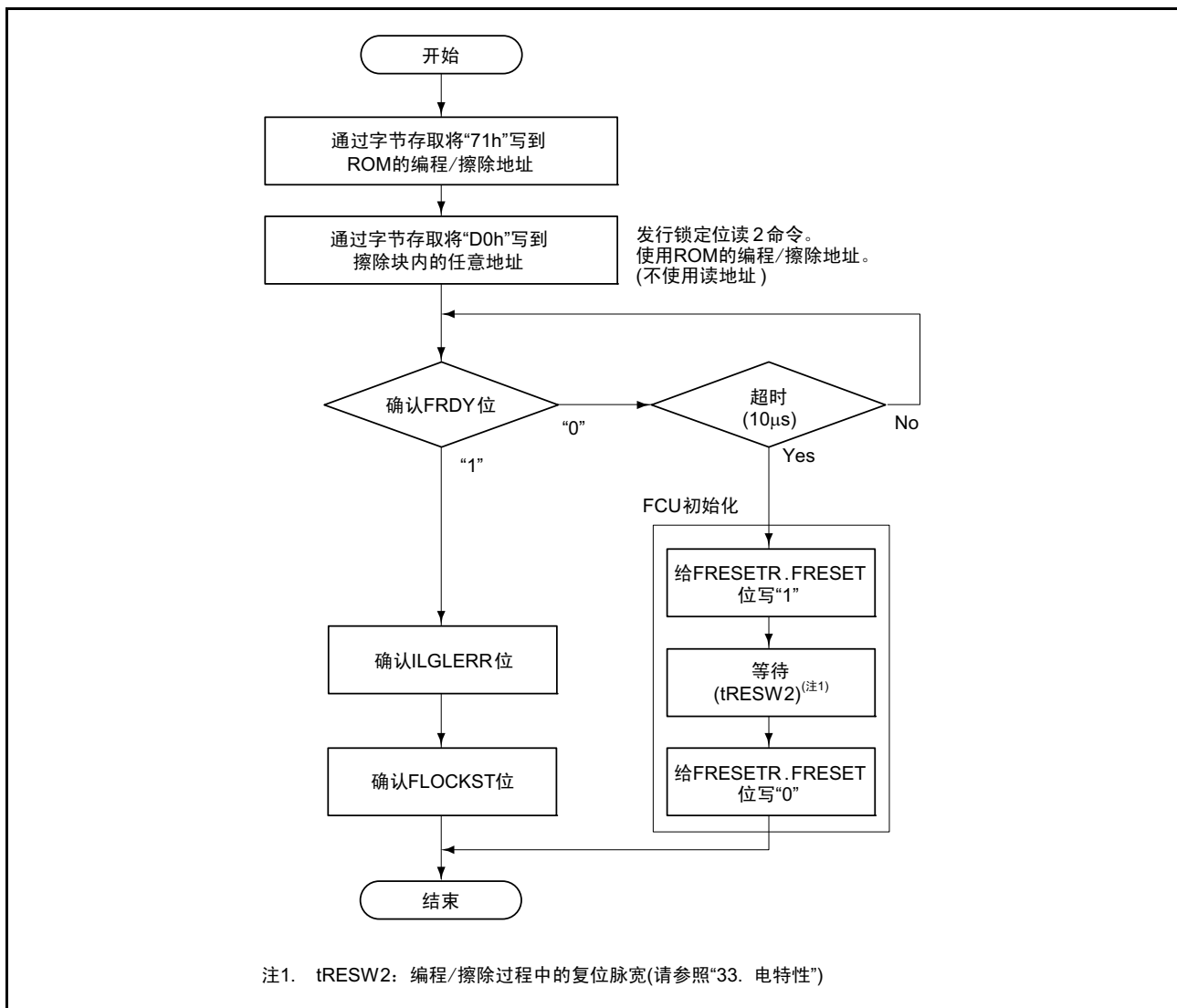


图 31.17 通过寄存器读方式读锁定位的方法

## 31.6.4.3 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“31.8 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“31.6.4.1 (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”。

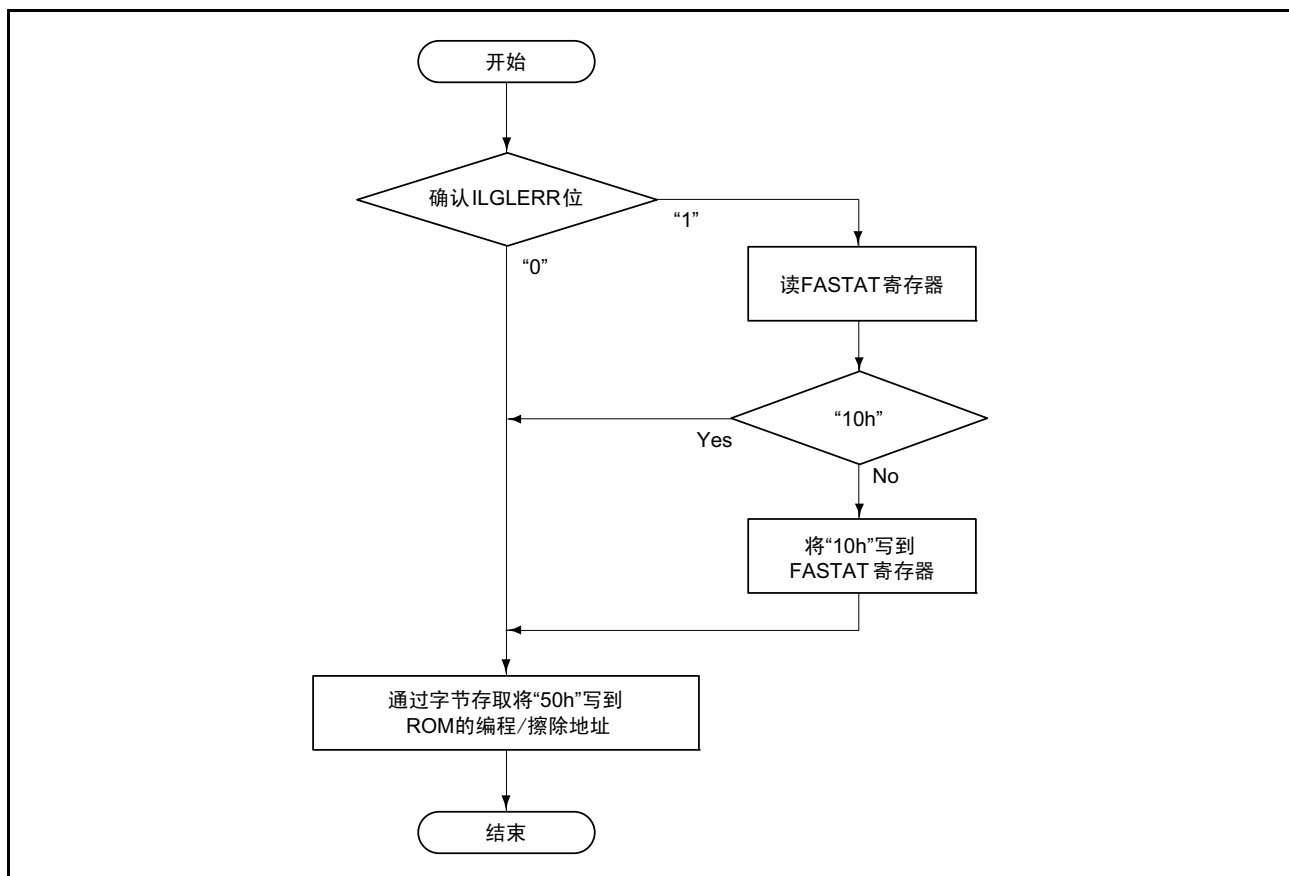


图 31.18 FSTATR0 寄存器的清除方法

## (3) FCU 的初始化方法

在发行 FCU 命令后，如果因超时而使 FSTATR0.FRDY 位不变为“1”，就需要通过 FRESETR 寄存器对 FCU 进行初始化。在 FSTATR1.FCUERR 位为“1”时，也需要通过 FRESETR 寄存器对 FCU 进行初始化。无论在何种情况下，FRESETR.FRESET 位为“1”的状态都必须保持 tRESW2 的时间（参照“33. 电特性”）。在 FRESET 位保持“1”的期间，禁止读 ROM/ 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 31.11 的处理。

#### 31.6.4.4 挂起 / 恢复

##### (1) 编程 / 擦除的挂起方法

使用 P/E 挂起命令来中止 ROM 的编程或者擦除。

要发行 P/E 挂起命令时，必须预先通过检查 FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位和 FSTATR1.FCUERR 位为“0”，确认正常进行了编程或者擦除处理。要确认能接受挂起命令时，也必须确认 FSTATR0.SUSRDY 位为“1”。在发行 P/E 挂起命令后，必须读 FSTATR0 寄存器和 FSTATR1 寄存器，确认没有发生错误。

如果在编程或者擦除处理过程中发生异常，ILGLERR 位、PRGERR 位、ERSERR 位和 FCUERR 位中至少有 1 位变为“1”。在确认 SUSRDY 位为“1”后到接受 P/E 挂起命令的期间，如果编程或者擦除处理已经结束，发行的 P/E 挂起命令就被检测为非法命令，因此 ILGLERR 位变为“1”。

如果在接受 P/E 挂起命令的同时编程 / 擦除处理结束，就不产生错误，也不转移到挂起状态（FSTATR0.FRDRY 位为“1”并且 FSTATR0.ERSSPD 位和 FSTATR0.PRGSPPD 位为“0”）。如果在接受 P/E 挂起命令并且编程 / 擦除的中止处理正常结束，FCU 就转移到挂起状态，FRDRY 位变为“1”并且 ERSSPD 位或者 PRGSPPD 位变为“1”。在发行 P/E 挂起命令后，必须通过检查 ERSSPD 位或者 PRGSPPD 位为“1”，确认转移到挂起状态，然后决定后续的流程。即使未转移到挂起状态，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“31.8.2 错误保护”）。

如果转移到擦除挂起状态，就能对擦除对象以外的块进行编程。在编程或者擦除挂起状态下，能通过清除 FENTRYR 寄存器转移到 ROM 读模式。

有关接受 P/E 挂起命令时的 FCU 运行内容，请参照“31.7 挂起”。

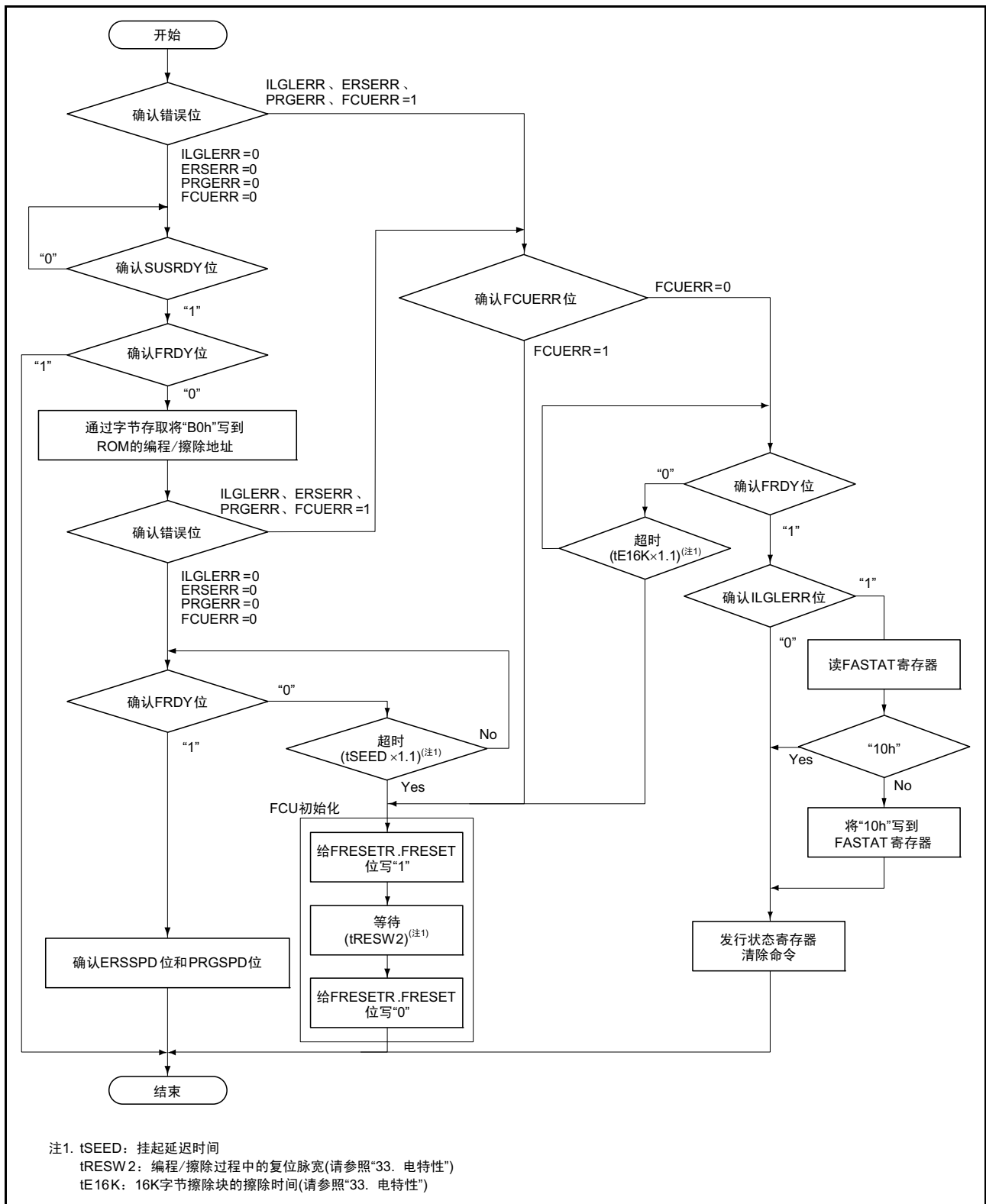


图 31.19 编程 / 擦除的挂起方法

## (2) 编程 / 擦除的恢复方法

使用 P/E 恢复命令重新开始挂起的编程或者擦除处理。如果在挂起过程中更改了 FENTRYR 寄存器的设定，就必须在发行 P/E 恢复命令前将发行 P/E 挂起命令前的值重新设定到 FENTRYR 寄存器。

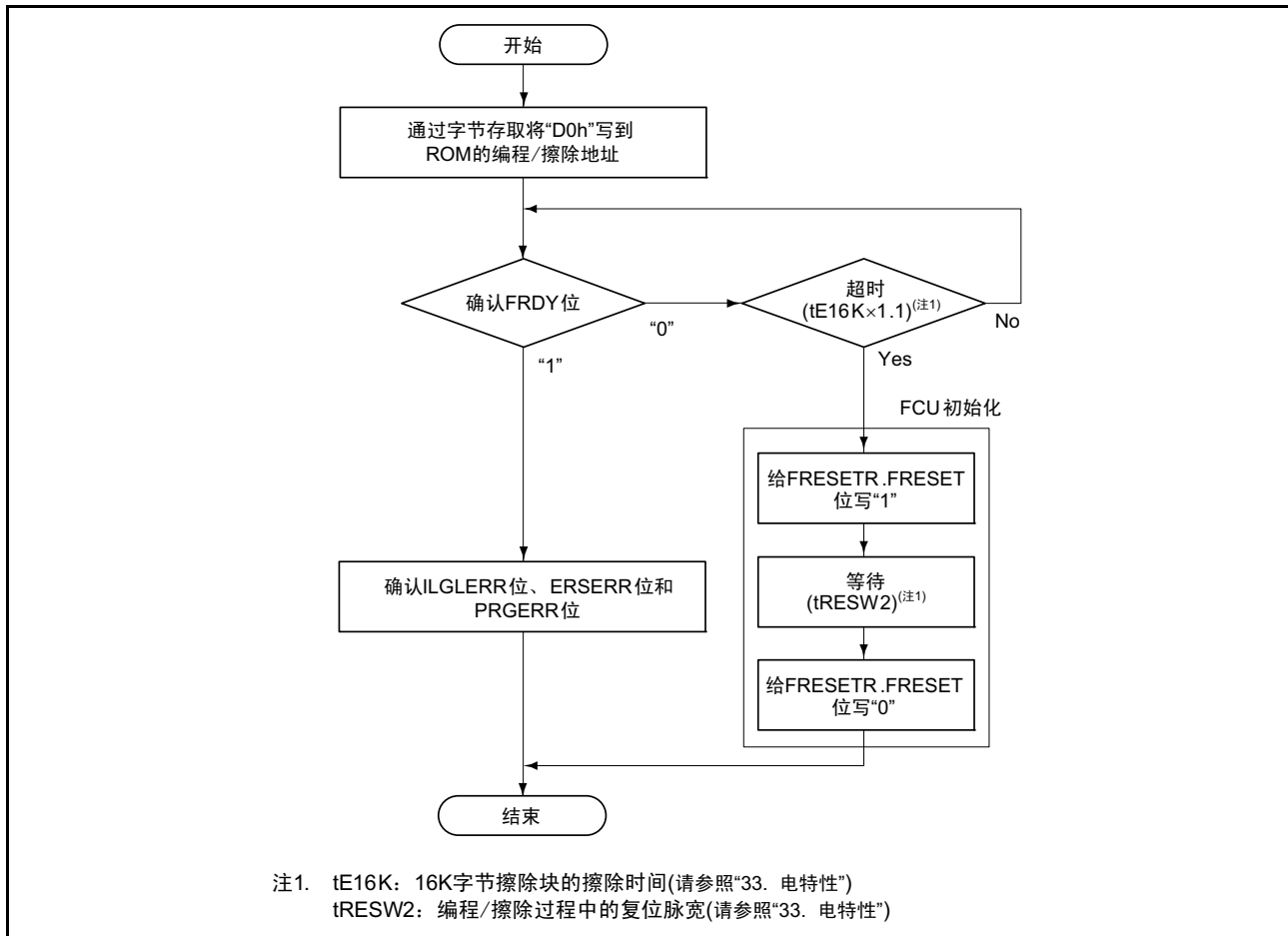


图 31.20 编程 / 擦除的恢复方法

## 31.7 挂起

不能在编程或者擦除处理过程中读 ROM。通过发行 P/E 挂起命令来中止 ROM 的编程或者擦除处理，变为能读 ROM 的状态。P/E 挂起命令有 1 种编程模式和 2 种擦除模式（挂起优先模式和擦除优先模式），还有用于重新开始被中止的编程或者擦除处理的 P/E 恢复命令。

### 31.7.1 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 31.21 所示。

如果 FCU 接受编程命令，就将 FSTATR0.FRDY 位置“0”，开始编程处理。如果在开始编程处理后 FCU 转移到能接受 P/E 挂起命令的状态，FSTATR0.SUSRDY 位就变为“1”。如果发行 P/E 挂起命令，FCU 就在接受挂起命令后将 SUSRDY 位置“0”。如果 FCU 在外加编程脉冲过程中接受 P/E 挂起命令，FCU 就继续外加脉冲。如果超过规定的脉冲外加时间，FCU 就结束脉冲的外加，并且在开始进行编程的中止处理后将 FSTATR0.PRGSPPD 位置“1”。一旦中止处理结束，FCU 就将 FRDY 位置“1”，转移到编程挂起状态。如果 FCU 在编程挂起状态下接受 P/E 恢复命令，就将 FRDY 位和 PRGSPPD 位置“0”，重新开始编程处理。

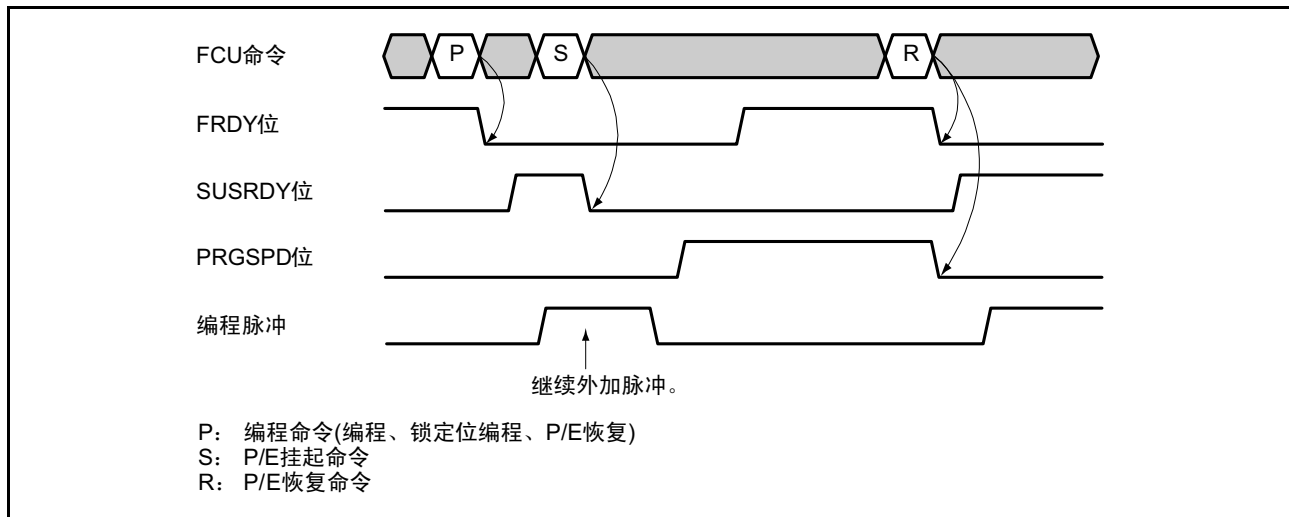


图 31.21 编程处理的中止



### 31.7.2 擦除时的挂起（挂起优先模式）

擦除挂起模式为挂起优先模式（FCPSR.ESUSPMD 位为“0”）时的擦除处理的中止如图 31.22 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，FSTATR0.SUSRDY 位就变为“1”。如果发行 P/E 挂起命令，FCU 就在接受挂起命令后将 SUSRDY 位置“0”。如果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”。一旦中断处理结束，FCU 就将 FRDY 位置“1”，转移到擦除挂起状态。如果 FCU 在擦除挂起状态下接受 P/E 恢复命令，就将 FRDY 位和 ERSSPD 位置“0”，重新开始擦除处理。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，而不取决与擦除挂起模式的设定。

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。

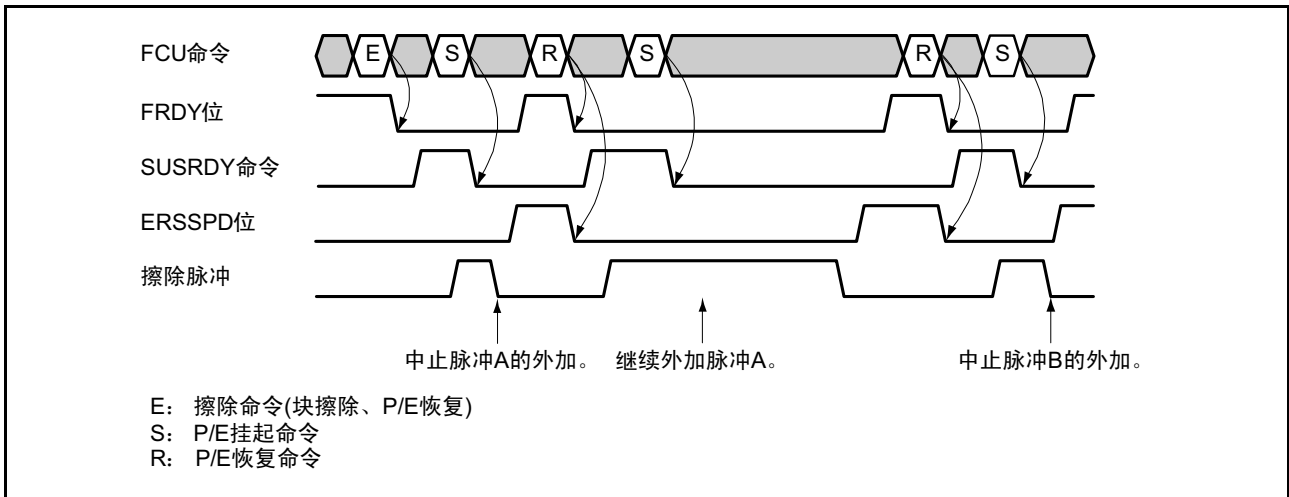


图 31.22 擦除处理的中止（挂起优先模式）

### 31.7.3 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 31.23 所示。擦除优先模式的擦除脉冲控制方式和编程中止处理的编程脉冲控制方式相同。

如果 FCU 在外加擦除脉冲过程中接受 P/E 挂起命令，就继续外加擦除脉冲。在此模式中，因为在发行 P/E 恢复命令时不会再次外加擦除脉冲，所以与挂起优先模式相比，能缩短整个擦除处理所需的时间。

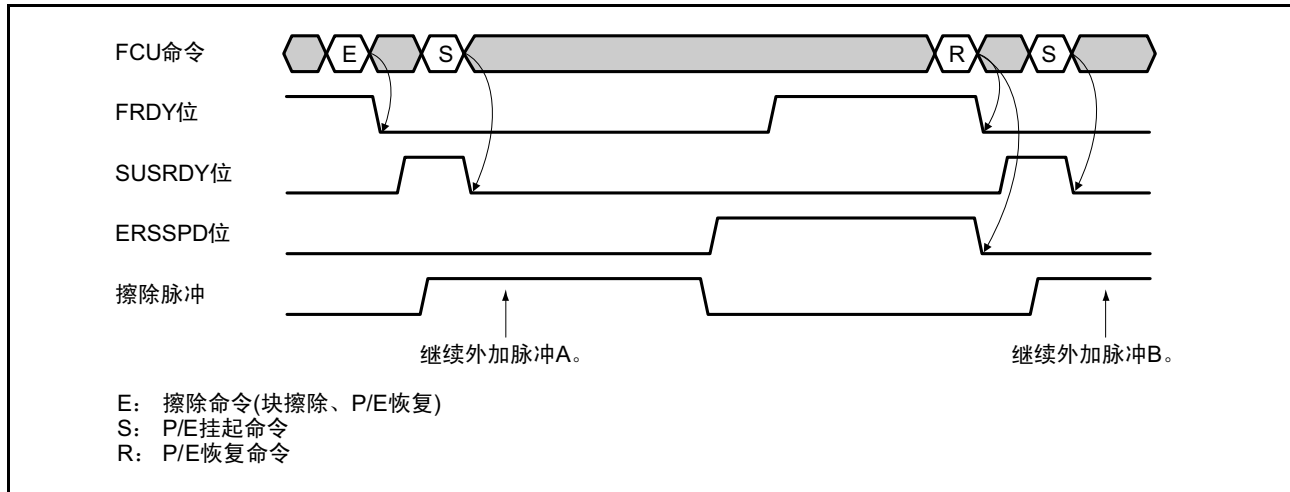


图 31.23 擦除处理的中止（擦除优先模式）

## 31.8 保护

ROM 的编程 / 擦除的保护有软件保护和错误保护。

### 31.8.1 软件保护

软件保护是通过设定控制寄存器和用户 MAT 的锁定位来禁止对 ROM 进行编程 / 擦除的状态。如果违反软件保护，发行 ROM 的编程 / 擦除命令，FCU 就检测到错误，进入命令锁定状态。

#### (1) 通过 FWEPROR 寄存器进行的保护

如果不将 FWEPROR.FLWE[1:0] 位置“01b”，在任何模式中都不能进行改写。

#### (2) 通过 FENTRYR 寄存器进行的保护

当 FENTRYR.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“31.8.2 错误保护”）。

#### (3) 通过锁定位进行的保护

用户 MAT 的各擦除块内置锁定位。当 FPROTR.FPROTCN 位为“0”时，锁定位为“0”的擦除块为禁止编程和擦除的状态。要对锁定位为“0”的擦除块进行编程或者擦除时，必须将 FPROTCN 位置“1”。如果违反通过锁定位进行的保护，发行 ROM 的编程 / 擦除命令，FCU 就检测到编程 / 擦除错误，进入命令锁定状态（参照“31.8.2 错误保护”）。

### 31.8.2 错误保护

错误保护是因误发行 FCU 命令以及发生禁止的存取而导致 FCU 在检测到误动作后禁止接受 FCU 命令的状态（命令锁定状态）。

一旦 FCU 进入命令锁定状态（FASTAT.CMDLK 位为“1”），就将状态位（FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位、FSTATR1.FCUERR 位和 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 相关的错误保护内容和错误检测时的状态位的值（FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位、FSTATR1.FCUERR 位和 FASTAT.ROMAE 位）的关系如表 31.9 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 31.9 错误保护一览表（ROM 专用 +ROM/ 数据闪存通用）

分类	内容	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE	CMDLK
FENTRYR 的设置错误	将 FENTRYR.FENTRYD 位和 FENTRYR.FENTRY0 位中至少 1 个位置“1”。	1	0	0	0	0	1
	挂起和恢复时的 FENTRYR 寄存器的设定不同。	1	0	0	0	0	1
非法命令错误	在 FCU 命令的第 1 周期指定未定义代码。	1	0	0	0	0	1
	在多个周期的 FCU 命令的最后周期指定“D0h”以外的数据。	1	0	0	0	0	1
	在编程或者擦除处理过程中发行挂起以外的命令。	1	0	0	0	0	1
	在编程和擦除以外的处理过程中发行挂起命令。	1	0	0	0	0	1
	在挂起状态下发行挂起命令。	1	0	0	0	0	1
	在挂起以外的状态下发行恢复命令。	1	0	0	0	0	1
	在编程挂起状态下发行编程 / 擦除（编程、锁定位编程、块擦除）命令。	1	0	0	0	0	1
	在擦除挂起状态下发行块擦除命令。	1	0	0	0	0	1
	在擦除挂起状态下发行擦除挂起对象区的编程命令 / 锁定位编程命令。	1	0	0	0	0	1
	在编程命令的第 2 个周期指定“80h”以外的数据。	1	0	0	0	0	1
在命令锁定状态下发行命令。	1	0/1	0/1	0/1	0/1	1	
擦除错误	在擦除处理过程中发生错误。	0	1	0	0	0	1
	在 FPROTR.FPROTCN 位为“0”时，对锁定位为“0”的擦除块发行块擦除命令。	0	1	0	0	0	1
编程错误	在编程处理过程中发生错误。	0	0	1	0	0	1
	在 FPROTR.FPROTCN 位为“0”时，对锁定位为“0”的擦除块发行编程命令或者锁定位编程命令。	0	0	1	0	0	1
FCU 错误	在 FCU 内部处理过程中发生错误。	0	0	0	1	0	1
ROM 存取违反	在 FENTRYR.FENTRY0 位为“1”时并且在 ROM P/E 正常模式中，对编程 / 擦除地址发行读存取命令。	1	0	0	0	1	1
	在 FENTRYR.FENTRY0 位为“0”时，对编程 / 擦除地址发行存取命令。	1	0	0	0	1	1
	在将 FENTRYR 寄存器不为“0000h”的状态下，对写地址发行读存取命令。	1	0	0	0	1	1

### 31.9 引导模式

#### 31.9.1 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 MAT 或者数据 MAT 进行编程或者擦除。为了进行主机和 RX62T 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 RX62T，就执行嵌入式程序存储 MAT 中的程序，此程序在自动调整 SCI 的位速率以及接受主机的控制命令后，控制编程和擦除。

引导模式的系统结构如图 31.24 所示。

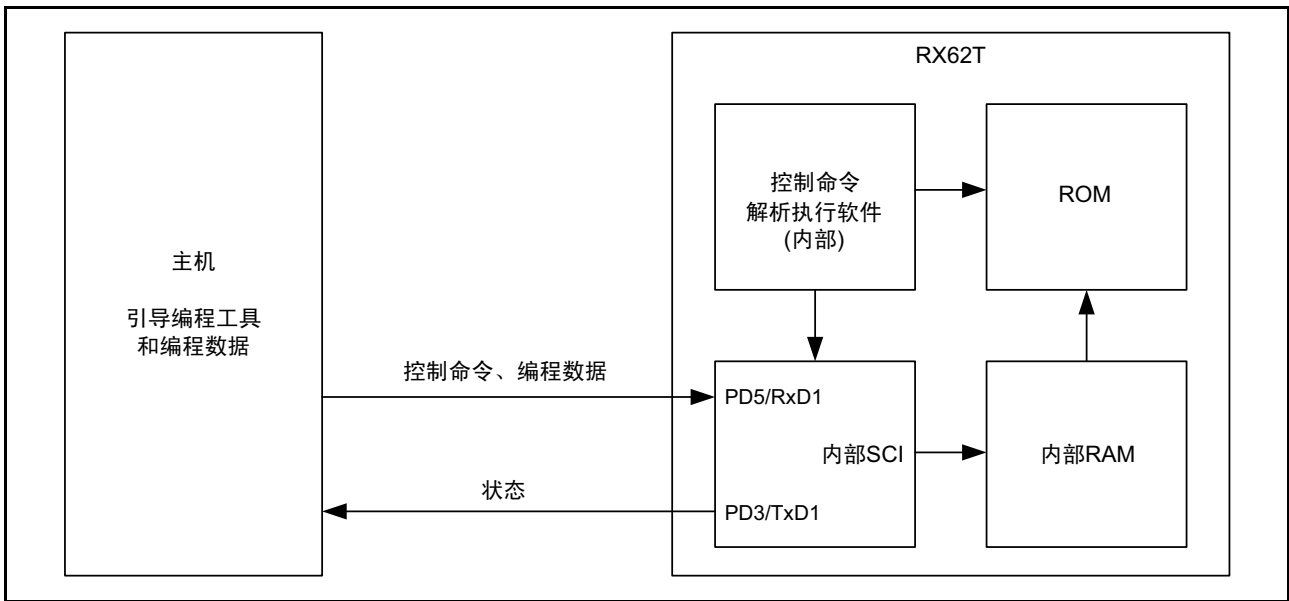


图 31.24 引导模式的系统结构

#### 31.9.2 ID 码保护

这是禁止从主机读、编程和擦除的功能。

使用写在 ROM 中的控制码和 ID 码，判断 ID 码保护的有效或者无效以及 ID 码保护。在 ID 码保护有效的情况下，只在从主机送来的代码和 ROM 中的控制码及 ID 码相同时，允许读、编程和擦除。

ROM 中的控制码和 ID 码是 32 位长的 4 字数据，控制码和 ID 码的结构如图 31.25 所示。必须以 32 位为单位设定 ID 码。

	31	24	23	16	15	8	7	0
FFFF FFA0h	控制码		ID 码1		ID 码2		ID 码3	
FFFF FFA4h	ID 码4		ID 码5		ID 码6		ID 码7	
FFFF FFA8h	ID 码8		ID 码9		ID 码10		ID 码11	
FFFF FFAC h	ID 码12		ID 码13		ID 码14		ID 码15	

图 31.25 ROM 中的控制码和 ID 码的结构

## (1) 控制码

控制码决定 ID 码保护的有效或者无效以及主机的验证方法，控制码和验证方法如表 31.10 所示。

表 31.10 ID 码保护的规格

控制码	ID 码	保护状态	连接 SCI 时的运行
45h	任意	保护有效 (验证方法 1)	ID 码相同: 结束 ID 码保护并且转移到主机命令等待状态。 ID 码不同: 再次转移到 ID 码等待状态。但是, 如果连续 3 次 ID 码不同, 就进行全部擦除。
52h	50h、72h、6Fh、74h、65h、 63h、74h、FFh、.....、FFh 除外	保护有效 (验证方法 2)	ID 码相同: 结束 ID 码保护并且转移到主机命令等待状态。 ID 码不同: 再次转移到 ID 码等待状态。
	50h、72h、6Fh、74h、65h、 63h、74h、FFh、.....、FFh	保护有效 (验证方法 3)	总是判断为 ID 码不同。
上述以外	—	保护无效	擦除全部的块。

## (2) ID 码

ID 码能设定为任意的值。如果控制码为“52h”并且 ID 码从 ID 码 1 开始按顺序设定“50h”、“72h”、“6Fh”、“74h”、“65h”、“63h”、“74h”、“FFh”、.....、“FFh”，就不判断 ID 码是否相同而总是视为不同，禁止从主机读、编程和擦除。

## (3) 设定 ID 码的编程例子

控制码为“45h”，ID 码从 ID 码 1 开始按顺序设定“01h”、“02h”、“03h”、“04h”、“05h”、“06h”、“07h”、“08h”、“0Ah”、“0Bh”、“0Ch”、“0Dh”、“0Eh”、“0Fh”时的编程例子：

```
.SECTION ID_CODE, CODE
.ORG 0FFFFFFA0h
.LWORD 45010203h
.LWORD 04050607h
.LWORD 08090A0Bh
.LWORD 0C0D0E0Fh
```

### 31.9.3 引导模式的状态转移

引导模式的状态转移图如图 31.26 所示。

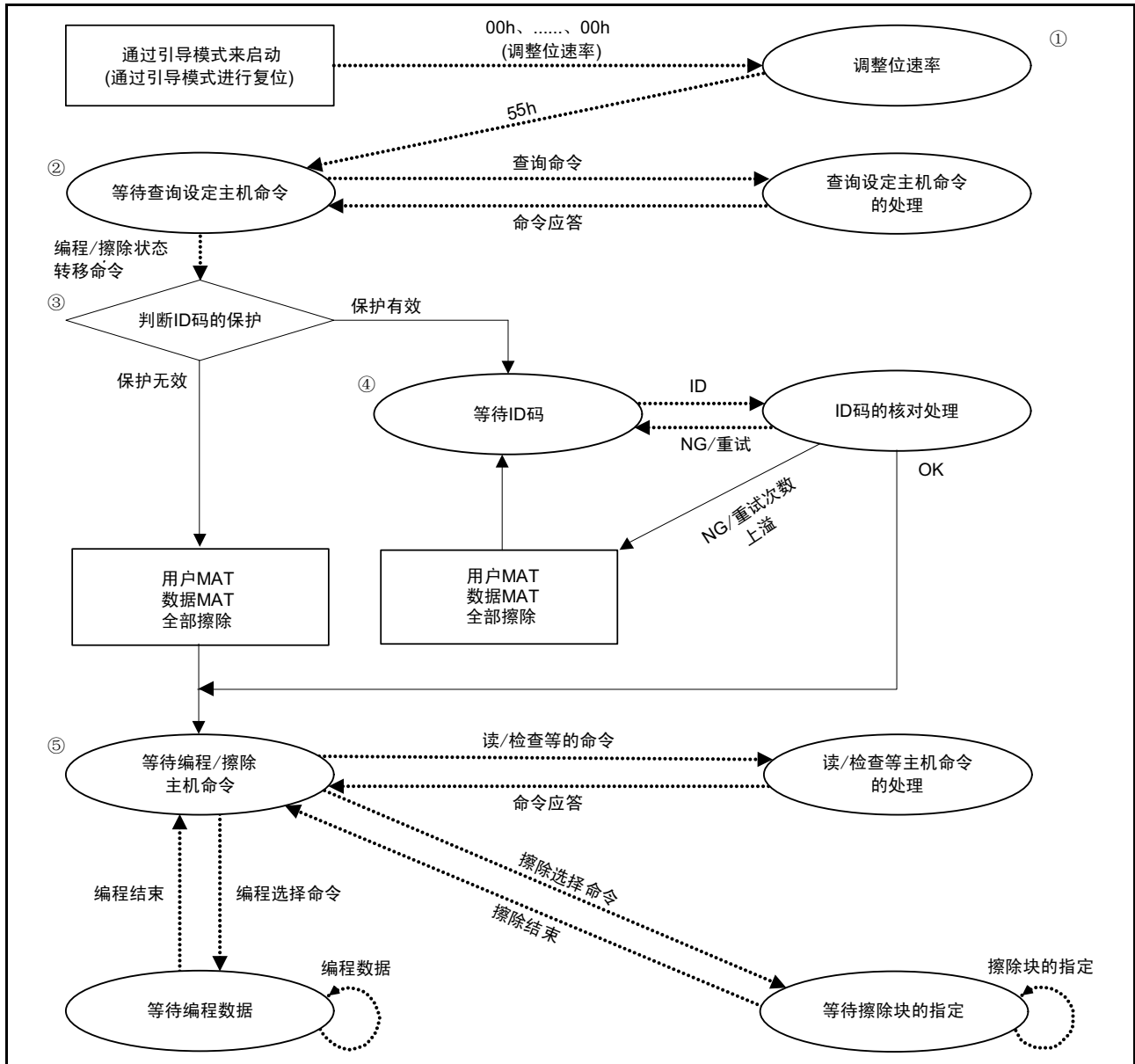


图 31.26 引导模式的状态转移图

#### ①位速率的调整

如果通过引导模式启动 RX62T，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 RX62T 发送到主机。此后，如果 RX62T 正常接受从主机送来的“55h”，就转移到查询设定主机命令等待状态。有关位速率调整的详细内容，请参照“31.9.4 位速率的自动调整”。

#### ②查询设定主机命令的等待

这是查询 MAT 大小、MAT 结构、MAT 起始地址和支持状况等以及选择器件、时钟模式和位速率的状态。如果从主机发行编程 / 擦除状态转移命令，就转移到判断 ID 码保护的有效或者无效的状态。有关查询设定主机命令的详细内容，请参照“31.9.5 查询设定主机命令的等待状态”。

### ③ ID 码保护的判断

判断 ID 码保护的有效或者无效。通过写在 ROM 中的控制码和 ID 码，判断 ID 码保护的有效或者无效。当 ID 码保护有效时，转移到 ID 码等待状态；当 ID 码保护无效时，全部擦除用户 MAT、用户引导 MAT 和数据 MAT，并且转移到编程 / 擦除主机命令等待状态。有关控制码和 ID 码的详细内容，请参照“31.9.2 ID 码保护”。

### ④ ID 码的等待

等待从主机送来的控制码和 ID 码。将主机送来的控制码、ID 码和 ROM 中的代码进行比较，如果相同，就转移到编程 / 擦除主机命令等待状态；如果不同，就返回到 ID 码等待状态。如果连续 3 次不同并且保护状态为验证方法 1，就进行全部擦除并且再次返回到 ID 码等待状态。要解除此不同的状态时，需要进行复位。有关控制码和 ID 码的详细内容，请参照“31.9.2 ID 码保护”。

### ⑤编程 / 擦除主机命令的等待

这是根据主机的命令进行编程或者擦除的状态。根据 RX62T 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 RX62T 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 RX62T 接收擦除选择命令，就转移到擦除块指定等待状态。主机在发送擦除选择命令后，必须接着发送擦除块号。如果将擦除块号设定为“FFh”，就在擦除结束后，从擦除块指定等待状态转移到编程 / 擦除命令等待状态。因为在通过引导模式启动后到转移到编程 / 擦除主机命令状态的期间，用户 MAT、数据闪存被全部擦除，所以在引导模式中除了不进行复位而要擦除新编程的数据的情况以外，不需要进行擦除。

除了编程 / 擦除命令以外，还有用户 MAT 的校验和命令、空白检查（擦除检查）命令、存储器读命令、状态信息取命令等主机命令。



### 31.9.4 位速率的自动调整

如果通过引导模式启动 RX62T，就测量从主机连续发送的异步 SCI 通信数据“00h”的 Low 电平期间。在测量 Low 电平期间时，主机的 SCI 发送 / 接收格式必须为 8 位数据、1 个停止位、无奇偶校验，位速率必须设定为 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，RX62T 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 RX62T。否则，就必须通过引导模式重新启动 RX62T，并且重新自动调整位速率。如果 RX62T 正常接收“55h”，就发送“E6h”。否则，就发送“FFh”。



图 31.27 自动调整位速率时的 SCI 发送 / 接收格式



图 31.28 主机和 RX62T 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 RX62T 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 31.11 所示的条件下进行 SCI 的通信。

表 31.11 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	8 ~ 14MHz
19200bps	8 ~ 14MHz

### 31.9.5 查询设定主机命令的等待状态

在查询设定主机命令等待状态下能使用的主机命令一览表如表 31.12 所示。在编程 / 擦除主机命令等待状态下，也能使用嵌入式程序状态查询命令。其他命令只能在查询设定主机命令等待状态下使用。

表 31.12 查询设定主机命令

主机命令名	功能
支持器件查询	查询器件码和嵌入式程序的产品型号。
器件选择	选择器件码。
时钟模式查询	查询时钟模式数和各自的值。
时钟模式选择	通知已选择的时钟模式。
倍增比查询	查询时钟的种类、倍增比 / 分频比的种类以及倍增比 / 分频比。
工作频率查询	查询时钟的种类、最大工作频率和最小工作频率。
用户 MAT 信息查询	查询用户 MAT 的个数以及起始地址和结束地址。
擦除块信息查询	查询块数以及起始地址和结束地址。
编程长度查询	查询编程时的数据长度。
新位速率选择	更改主机和 RX62T 之间的 SCI 通信的位速率。
编程 / 擦除状态转移	转移到 ID 码保护判断状态。
嵌入式程序状态查询	查询处理状态。

如果主机发送未定义的命令，RX62T 就发送命令错误应答。命令错误应答的内容如下所示。在命令中保存主机发送的命令的起始字节。

错误应答	80h	命令
------	-----	----

在查询设定主机命令等待状态下，必须参考查询命令的应答，按照器件的选择 → 时钟模式的选择 → 新位速率的选择的顺序从主机发送选择命令，设定 RX62T。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，RX62T 就发送命令错误应答。查询设定主机命令等待状态下的主机命令使用例子如图 31.29 所示。

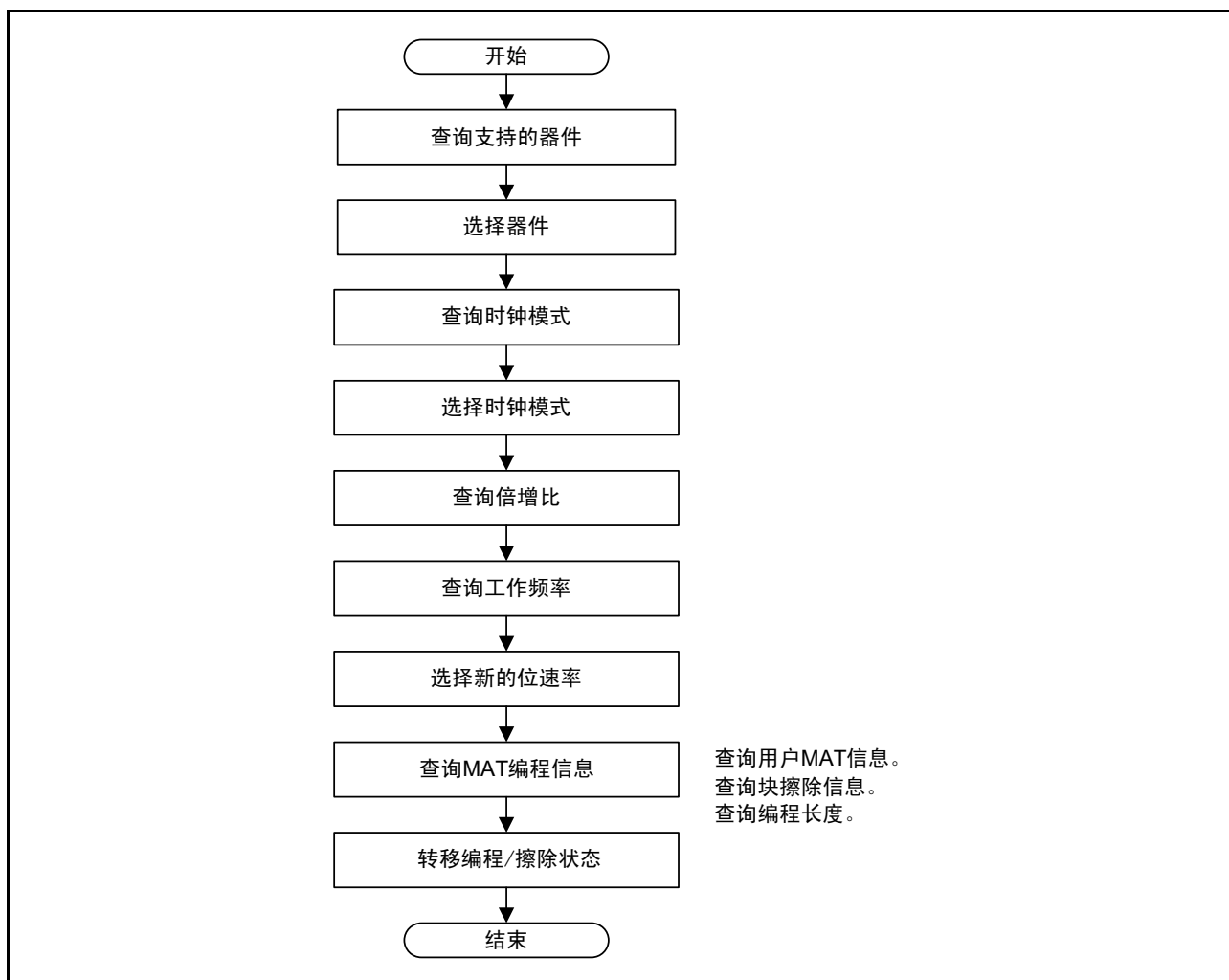


图 31.29 用户 MAT 的查询设定主机命令的使用例子

以下说明各主机命令的详细内容。说明内容中的“命令”是从主机发送到 RX62T 的命令，“应答”是从 RX62T 发送到主机的应答。“校验和”是指在合计 RX62T 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，RX62T 就通过引导模式的嵌入式程序发送能支持的器件信息。如果在主机选择器件后发送支持器件查询命令，RX62T 就只发送所选的器件信息。

命令	20h			
应答	30h	长度	器件数	
	字符数	器件码		产品名称
	字符数	器件码		产品名称
	...	...		...
	字符数	器件码		产品名称
	SUM			

【符号说明】	长度（1 字节）	: 器件数、字符数、器件码和产品名称的数据总字节数
	器件数（1 字节）	: 引导模式的嵌入式程序支持的品种数
	字符数（1 字节）	: 器件码和产品名称的字符数
	器件码（4 字节）	: 芯片产品名称的 ASCII 码
	产品名称（n 字节）	: 支持器件名称的 ASCII 码
	SUM（1 字节）	: 校验和

## (2) 器件选择

如果主机发送器件选择命令，RX62T 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，RX62T 就将支持器件更改为指定的器件并且发送应答（06h）。如果不是能支持的器件或者送来的命令为非法命令，RX62T 就发送错误应答（90h）。

即使在支持器件查询结果的器件数为“01h”时，也必须通过器件选择命令设定查询结果的器件码的值。

命令	10h	长度	器件码	SUM
应答	06h			
错误应答	90h	错误		

【符号说明】	长度（1 字节）	: 器件码的字符数（是固定值，为“4”。）
	器件码（4 字节）	: 芯片产品名称的 ASCII 码（和支持器件查询命令的应答相同的代码）
	SUM（1 字节）	: 校验和
	错误（1 字节）	: 错误码
		11h: 校验和错误（命令为非法命令）
		21h: 器件码不同

### (3) 时钟模式查询

如果主机发送时钟模式查询命令，RX62T 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，RX62T 就只发送所选的时钟模式的信息。

命令	21h			
应答	31h	长度		
	模式	模式	...	模式
	SUM			

【符号说明】	长度（1 字节）	: 模式数、模式的数据总字节数
	模式（1 字节）	: 可选择的时钟模式（例：01h 时钟模式 1）
	SUM（1 字节）	: 校验和

### (4) 时钟模式选择

如果主机发送时钟模式选择命令，RX62T 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，RX62T 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，RX62T 就发送错误应答（91h）。

必须在发送器件选择命令后发送时钟模式选择命令。即使在时钟模式查询结果的时钟模式数为“00h”或者“01h”时，也必须通过时钟模式选择命令设定查询结果的模式的值。

命令	11h	长度	模式	SUM
应答	06h			
错误应答	91h	错误		

【符号说明】	长度（1 字节）	: 模式的字符数（是固定值，为“1”。）
	模式（1 字节）	: 时钟模式（和时钟模式查询命令的应答相同的模式）
	SUM（1 字节）	: 校验和
	错误（1 字节）	: 错误码
		11h: 校验和错误（命令为非法命令）
		22h: 时钟模式不同

## (5) 倍增比查询

如果主机发送倍增比查询命令，RX62T 就发送时钟的种类、倍增比 / 分频比的种类以及倍增比 / 分频比的信息。

命令	22h			
应答	32h	长度	时钟数	
	倍增比的种类	倍增比	倍增比	...
	倍增比的种类	倍增比	倍增比	...
	...	...	...	...
	倍增比的种类	倍增比	倍增比	...
	SUM			

- 【符号说明】
- 长度（1 字节）：时钟数、倍增比的种类、倍增比的数据总字节数
  - 时钟数（1 字节）：时钟的种类（例：02h 系统时钟和外围时钟）
  - 倍增比的种类（1 字节）：可选择的倍增比 / 分频比的种类  
（例：04h 系统时钟有 1 倍增、2 倍增、4 倍增、8 倍增共 4 种）
  - 倍增比（1 字节）：倍增比（例：04h=4 4 倍增）← 用正数指定  
分频比（例：FEh=-2 2 分频）← 用负数指定
  - SUM（1 字节）：校验和

## (6) 工作频率查询

如果主机发送工作频率查询命令，RX62T 就发送各时钟的工作频率的最小值和最大值的消息。

命令	23h	
应答	33h	长度
	最小频率	最大频率
	最小频率	最大频率
	...	...
	最小频率	最大频率
	SUM	

- 【符号说明】
- 长度（1 字节）：时钟数、最小频率、最大频率的数据总字节数
  - 时钟数（1 字节）：时钟的种类（例：02h 系统时钟和外围时钟）
  - 最小频率（2 字节）：工作频率的最小值（例：07D0h 20.00MHz）  
这是将频率（MHz）的小数点后的 2 位的值扩大 100 倍的值。
  - 最大频率（2 字节）：工作频率的最大值  
格式和最小频率相同。
  - SUM（1 字节）：校验和

## (7) 用户 MAT 信息查询

如果主机查询用户 MAT 信息，RX62T 就发送用户 MAT 的区域数和地址的信息。

命令	25h		
应答	35h	长度	区域数
	区域的起始地址		
	区域的结束地址		
	区域的起始地址		
	区域的结束地址		
	...		
	区域的起始地址		
	区域的结束地址		
	SUM		

- 【符号说明】 长度（1 字节）：区域数、区域的起始地址、区域的结束地址的数据总字节数  
 区域数（1 字节）：用户 MAT 的区域数（连续区域为 1 个区域）  
 区域的起始地址（4 字节）：用户 MAT 区域的起始地址  
 区域的结束地址（4 字节）：用户 MAT 区域的结束地址  
 SUM（1 字节）：校验和

## (8) 擦除块信息查询

如果主机查询擦除块信息，RX62T 就发送用户 MAT 的擦除块数和地址的信息。

命令	26h		
应答	36h	长度	块数
	块的起始地址		
	块的结束地址		
	块的起始地址		
	块的结束地址		
	...		
	块的起始地址		
	块的结束地址		
	SUM		

- 【符号说明】 长度（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 MAT 的擦除块数  
 区域的起始地址（4 字节）：擦除块的起始地址  
 区域的结束地址（4 字节）：擦除块的结束地址  
 SUM（1 字节）：校验和

## (9) 编程长度查询

如果主机查询编程长度，RX62T 就发送编程长度的信息。

命令	27h			
应答	37h	长度	编程长度	SUM

【符号说明】 长度（1 字节）：编程长度的字符数（是固定值，为“2”。）  
 编程长度（2 字节）：编程单位（以字节数为单位）  
 SUM（1 字节）：校验和

## (10) 新位速率选择

如果主机发送新位速率选择命令，RX62T 就检查能否将内部 SCI 设定为指定的新位速率。如果能设定新为位速率，RX62T 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，RX62T 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），RX62T 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 31.30 新位速率的选择顺序

命令	3Fh	长度	位速率		输入频率
	时钟数	倍增比 1	倍增比 2		
	SUM				
应答	06h				
错误应答	BFh	错误			
确认	06h				
应答	06h				



【符号说明】	长度（1字节）	: 位速率、输入频率、时钟数、倍增比的数据总字节数
	位速率（2字节）	: 新位速率（例：00C0h 19200bps） 将位速率值设定为 1/100 的值。
	输入频率（2字节）	: RX62T 的输入频率（例：04E2h 12.50MHz） 这是将输入频率的小数点后的 2 位的值扩大 100 倍的值。
	时钟数（1字节）	: 时钟的种类（例：02h 系统时钟和外围时钟）
	倍增比 1（1字节）	: 输入频率对应的系统时钟（ICLK）的倍增比 / 分频比 倍增比（例：04h=4 4 倍增）← 用正数指定 分频比（例：FEh=-2 2 分频）← 用负数指定
	倍增比 2（1字节）	: 输入频率对应的外围时钟（PCLK）的倍增比 / 分频比 和倍增比 1 相同的格式
	SUM（1字节）	: 校验和
	错误	: 错误码 11h: 校验和错误 24h: 位速率不能选择错误 25h: 输入频率错误 26h: 倍增比错误 27h: 工作频率错误

- 位速率不能选择错误

如果因RX62T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。

$$\text{误差(\%)} = \frac{f_{EX} \times M_{P\phi} \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1$$

- 输入频率错误

如果通过新位速率选择命令指定的输入频率不在与通过时钟模式选择命令指定的时钟模式对应的输入频率的最小值和最大值的范围内，就产生输入频率错误。

- 倍增比错误

如果通过新位速率选择命令指定的倍增比不是与通过时钟模式选择命令指定的时钟模式对应的倍增比，就产生倍增比错误。必须使用倍增比查询命令确认能选择的倍增比。

- 工作频率错误

如果在通过新位速率选择命令指定的工作频率下RX62T无法运行，就产生工作频率错误。RX62T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，RX62T 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（26h）并且转移到 ID 码等待状态；当 ID 码保护无效时，全部  
擦除用户 MAT/ 数据 MAT。在全部擦除后，RX62T 发送应答（06h）并且转移到编程 / 擦除主机命令等待状  
态。如果发生错误而没有全部擦除，RX62T 就发送错误应答（C0h→51h）。

不能在选择器件、时钟模式和新位速率前发行编程 / 擦除状态转移命令。

命令	40h
应答	ACK
错误应答	C0h      51h

【符号说明】 ACK（1 字节）                   : ACK 码  
  06h: ID 码保护无效  
  26h: ID 码保护有效

## (12) 嵌入式程序状态查询

如果主机发送嵌入式程序状态查询命令，RX62T 就发送当前的状态。能在查询设定主机命令等待状态和  
编程 / 擦除主机命令等待状态下使用嵌入式程序状态查询命令。

命令	4Fh
应答	5Fh      长度      状态      错误

【符号说明】 长度（1 字节）                   : 状态、错误的总字节数（是固定值，为“2”。）  
  状态（1 字节）                               : RX62T 的状态（参照表 31.13）  
  错误（1 字节）                               : RX62T 的错误发生状况（参照表 31.14）

表 31.13 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 / 擦除主机命令等待状态的转移（位速率选择结束）。
31h	正在擦除用户 MAT。
3Fh	等待编程 / 擦除主机命令。
4Fh	等待编程数据的接收。
5Fh	等待擦除块的指定。

表 31.14 错误的内容

代码	内容
00h	无错误
11h	校验和错误
21h	器件码不同错误
22h	时钟模式不同错误
24h	位速率不能选择错误
25h	输入频率错误
26h	倍增比错误
27h	工作频率错误
29h	块号错误
2Ah	地址错误
2Bh	数据长度错误
51h	擦除错误
52h	未擦除错误
53h	编程错误
54h	选择处理错误
80h	命令错误
FFh	位速率调整确认错误

### 31.9.6 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 31.15 所示。

表 31.15 ID 码检查主机命令

主机命令名	功能
ID 码检查	进行 ID 码的检查。

如果主机发送未定义命令，RX62T 就发送命令错误应答。有关命令错误的内容，请参照“31.9.5 查询设定主机命令的等待状态”。

#### (1) ID 码检查

如果主机发送 ID 码检查命令，RX62T 就将 ROM 中的控制码、ID 码和主机送来的码进行比较，并且送回比较结果。

命令	60h	长度
	控制码 +ID 码	
	SUM	
应答	ACK	
	E0h	错误

【符号说明】	长度（1 字节）	: ID 码的字节数（是固定值，为“16”。）
	ID 码（16 字节）	: 控制码（1 字节）+ID 码（15 字节）
	SUM（1 字节）	: 校验和
	ACK（1 字节）	: ACK 码
		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。

### 31.9.7 编程 / 擦除主机命令等待状态

在编程 / 擦除主机命令等待状态下能使用的主机命令一览表如表 31.16 所示。

表 31.16 编程 / 擦除主机命令

主机命令名	功能
用户 MAT 编程选择	选择用户 MAT 的编程。
256 字节编程	256 字节的编程
擦除选择	选择擦除。
块擦除	擦除块数据。
存储器读	读存储器。
用户 MAT 校验和	用户 MAT 的校验和
用户 MAT 空白检查	用户 MAT 的空白检查
读锁定位状态	读锁定位。
锁定位编程	对锁定位进行编程。
锁定位有效	设定为锁定位保护有效。
锁定位无效	设定为锁定位保护无效。
嵌入式程序状态查询	查询 RX62T 的状态。

如果主机发送未定义命令，RX62T 就发送命令错误应答。有关命令错误的内容，请参照“31.9.5 查询设定主机命令的等待状态”

在进行 ROM 的编程时，主机在发送编程选择命令（选择用户 MAT 编程）后发送 256 字节编程命令。如果主机发送编程选择命令，RX62T 就进入编程数据等待状态（参照“31.9.3 引导模式的状态转移”）。如果在编程数据等待状态下主机发送 256 字节编程命令，RX62T 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，RX62T 就判断为编程结束，转移到编程 / 擦除主机命令等待状态。

在擦除 ROM 时，主机在发送擦除选择命令后发送块擦除命令。如果主机发送擦除选择命令，RX62T 就进入擦除块指定等待状态（参照“31.9.3 引导模式的状态转移”）。如果在擦除块指定等待状态下主机发送块擦除命令，RX62T 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，RX62T 就判断为擦除结束，转移到编程 / 擦除主机命令等待状态。

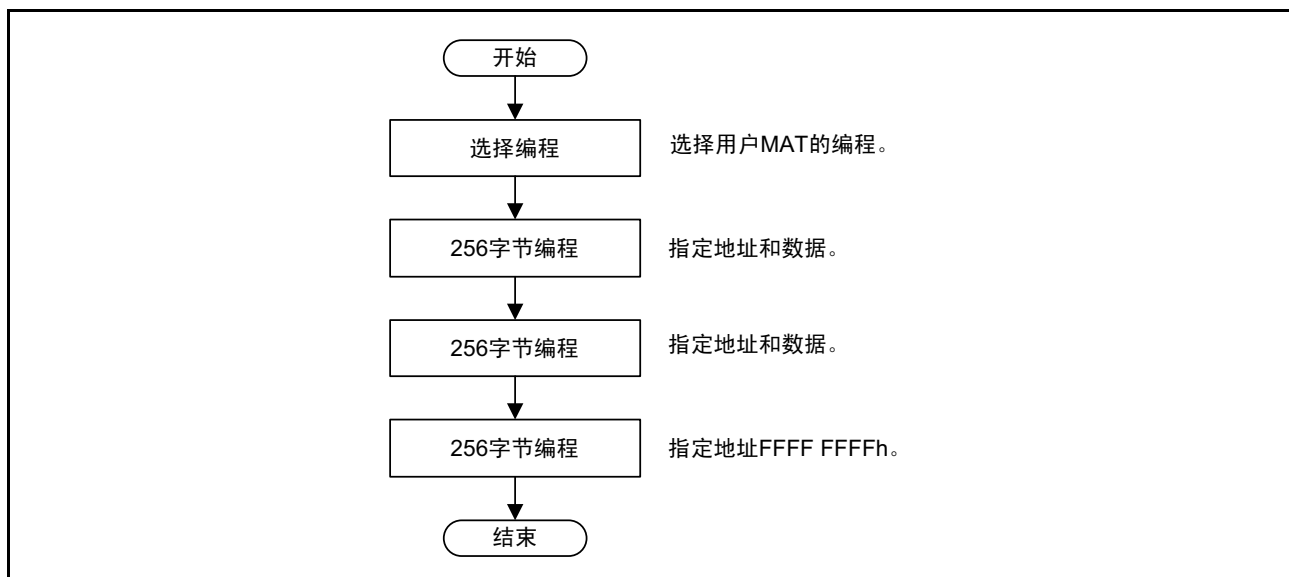


图 31.31 引导模式的 ROM 编程方法

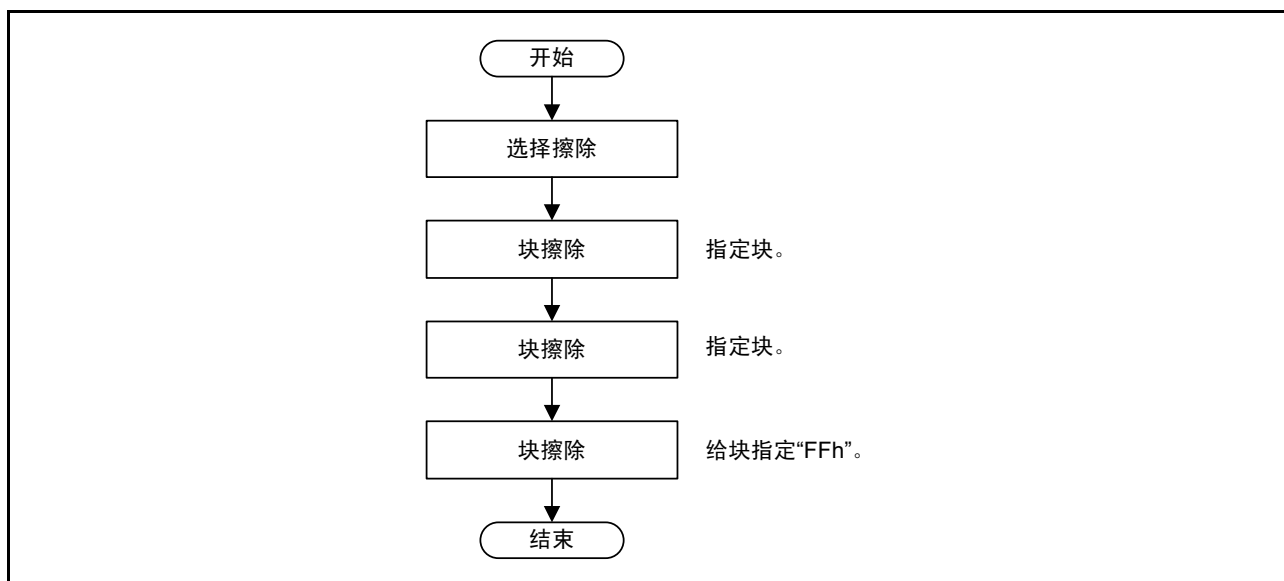


图 31.32 引导模式的 ROM 擦除方法

以下说明各主机命令的详细内容。说明内容中的“命令”是从主机发送到 RX62T 的命令，“应答”是从 RX62T 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 MAT 编程选择

如果主机发送用户 MAT 编程选择命令，RX62T 就选择用户 MAT 编程程序，进入编程数据等待状态。

命令 43h

应答 06h

## (2) 256 字节编程

如果主机发送 256 字节编程命令，RX62T 就进行 ROM 的编程。如果 ROM 的编程正常结束，RX62T 就发送应答（06h）。如果在编程处理过程中发生错误，RX62T 就发送错误应答（D0h）。

命令	50h	编程地址		
	数据	数据	...	数据
	SUM			
应答	06h			
错误应答	D0h	错误		

【符号说明】	编程地址（4 字节）	: 编程目标地址 在进行编程时调整为以 256 字节为边界的地址。 在指定编程结束时发送“FFFF FFFFh”。
	数据（256 字节）	: 编程数据 给不需要编程的字节指定“FFh”。 在指定编程结束时不需要发送数据。 （按照编程地址 →SUM 的顺序进行发送）
	SUM（1 字节）	: 校验和
	错误（1 字节）	: 错误码 11h: 校验和错误 2Ah: 地址错误（地址不在指定的 MAT 内） 53h: 发生编程错误，不能进行编程。

## (3) 擦除选择

如果主机发送擦除选择命令，RX62T 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，RX62T 就擦除 ROM。如果 ROM 擦除正常结束，RX62T 就发送应答（06h）。如果在擦除处理过程中发生错误，RX62T 就发送错误应答（D8h）。

命令	58h	长度	块	SUM
应答	06h			
错误应答	D8h	错误		

【符号说明】	长度（1 字节）	:	块的数据字节数（是固定值，为“1”。）
	块（1 字节）	:	要擦除的擦除块号 在指定擦除的结束时发送“FFh”。
	SUM（1 字节）	:	校验和
	错误（1 字节）	:	错误码 11h: 校验和错误 29h: 块号错误（块号不正确） 51h: 发生擦除错误，不能进行擦除。

## (5) 存储器读

如果主机发送存储器读命令，RX62T 就读 ROM。如果读操作正常结束，RX62T 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，RX62T 就发送错误应答（D2h）。

命令	52h	长度	区域	读起始地址	
	读长度			SUM	
应答	52h	读长度			
	数据	数据	...	数据	
	SUM				
错误应答	D2h	错误			

【符号说明】	长度（1 字节）	:	区域、读地址、读长度的数据总字节数
	区域（1 字节）	:	读对象的 MAT 01h: 用户 MAT
	读起始地址（4 字节）	:	读对象区的起始地址
	读长度（4 字节）	:	读取数据的长度（以字节为单位）
	SUM（1 字节）	:	校验和
	数据（1 字节）	:	从 ROM 读取的数据
	错误（1 字节）	:	错误码 11h: 校验和错误 2Ah: 地址错误 • 在区域的选择中指定“00h”和“01h”以外的值。 • 读起始地址不在指定的 MAT 区内。 2Bh: 长度错误 • 在读长度的选择中指定“00h”。 • 读长度超过 MAT 的长度。 • 由读起始地址和读长度计算出的地址不在 MAT 区内。



## (6) 用户 MAT 校验和

如果主机发送用户 MAT 校验和命令，RX62T 就发送以字节为单位将用户 MAT 的数据相加后的结果（校验和）。

命令	4Bh			
应答	5Bh	长度	MAT 的校验和	SUM

- 【符号说明】**
- 长度（1 字节）：MAT 的校验和的字节数（是固定值，为“4”。）
  - MAT 的校验和（4 字节）：用户 MAT 的校验和结果  
用户 MAT 也包含用于验证调试功能的键码。  
必须注意：相加的结果包含键码值。
  - SUM（1 字节）：校验和（应答数据的校验和）

## (7) 用户 MAT 空白检查

如果主机发送用户 MAT 空白检查命令，RX62T 就检查用户 MAT 是否全部为擦除状态。如果用户 MAT 全部为擦除状态，RX62T 就发送应答（06h）。如果用户 MAT 有未擦除区域，RX62T 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，RX62T 就读锁定位。如果读操作正常结束，RX62T 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，RX62T 就发送错误应答（F1h）。

命令	71h	长度	区域	中位地址	高位地址	最高位地址	SUM
应答	状态						
错误应答	F1h	错误					

【符号说明】	长度（1 字节）	:	区域、中位地址、高位地址、最高位地址的数据总字节数（RX62T 中为固定值“4”。）
	区域（1 字节）	:	读对象的 MAT 01h: 用户 MAT
	中位地址（1 字节）	:	指定块的结束地址的中位地址（8 ~ 15 位）
	高位地址（1 字节）	:	指定块的结束地址的高位地址（16 ~ 23 位）
	最高有效位地址（1 字节）	:	指定块的结束地址的最高位地址（24 ~ 31 位）
	SUM（1 字节）	:	校验和
	状态（1 字节）	:	当 bit6 为“0”时，为锁定状态。 当 bit6 为“1”时，为非锁定状态。
	错误（1 字节）	:	错误码 11h: 校验和错误 2Ah: 地址错误（地址不在指定的 MAT 内）

## (9) 锁定位编程

如果主机发送锁定位编程命令，RX62T 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，RX62T 就发送应答（06h）。否则，RX62T 就发送错误应答（F7h）。

命令	77h	长度	区域	中位地址	高位地址	最高位地址	SUM
应答	06h						
错误应答	F7h	错误					

【符号说明】	长度（1 字节）	:	区域、中位地址、高位地址、最高位地址的数据总字节数（RX62T 中为固定值“4”。）
	区域（1 字节）	:	锁定对象的 MAT 01h: 用户 MAT
	中位地址（1 字节）	:	指定块的结束地址的中位地址（8 ~ 15 位）
	高位地址（1 字节）	:	指定块的结束地址的高位地址（16 ~ 23 位）
	最高有效位地址（1 字节）	:	指定块的结束地址的最高位地址（24 ~ 31 位）
	SUM（1 字节）	:	校验和
	错误（1 字节）	:	错误码 11h: 校验和错误 2Ah: 地址错误（地址不在指定的 MAT 内） 53h: 发生编程错误，不能设定为锁定状态。

## (10) 锁定位有效

如果主机发送锁定位有效命令，RX62T 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，RX62T 就将锁定位置为无效。

命令 

75h
-----

应答 

06h
-----

## (12) 嵌入式程序状态查询

请参照“31.9.5 查询设定主机命令的等待状态”。

### 31.10 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，就将从 on-chip 调试器送来的代码和 ROM 中的控制码、ID 码进行比较。如果相同，就允许连接 on-chip 调试器；否则就不能连接 on-chip 调试器。但是，如果控制码为“52h”并且 ID 码从 ID 码 1 开始按照顺序设定“50h”、“72h”、“6Fh”、“74h”、“65h”、“63h”、“74h”、“FFh”、.....、“FFh”，就不判断 ID 码而总是视为不同，禁止连接 on-chip 调试器。当控制码和 ID 码都为“FFh”时，就不判断 ID 码而视为相同，允许连接 on-chip 调试器。闪存的 ID 码的结构和图 31.25 相同。

表 31.17 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	不判断控制码和 ID 码，总是视为和 ID 码相同，允许连接 on-chip 调试器。
52h	50h、72h、6Fh、74h、65h、63h、74h、FFh、.....、FFh	保护有效	不判断控制码和 ID 码，总是视为和 ID 码不同，禁止连接 on-chip 调试器。
上述以外	上述以外	保护有效	ID 码相同：结束 on-chip 调试器的验证，允许连接 on-chip 调试器。 ID 码不同：再次转移到 ID 码等待状态。

### 31.11 ROM 码保护

ROM 码保护是在使用 PROM 编程器时禁止读和改写闪存的功能。闪存的 ROM 码是 32 位长的数据。ROM 码的结构如图 31.33 所示，必须以 32 位为单位设定 ROM 码。

要解除 ROM 码保护时，必须在引导模式或者用户编程模式中擦除包括 ROM 码的用户 MAT 的 EB00 块。

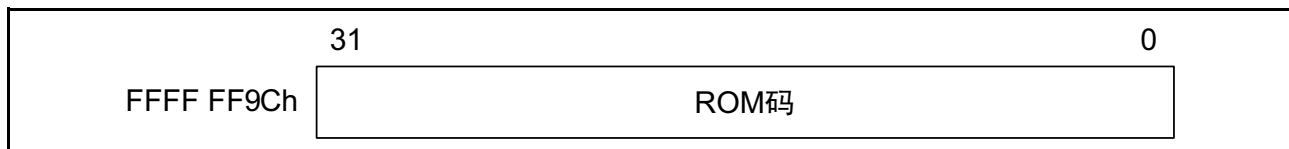


图 31.33 ROM 码的结构

表 31.18 ROM 码保护的规格

ROM 码	保护状态	连接 PROM 编程器时的运行
0000 0000h	ROM 码保护有效（ROM 码保护 1）	禁止读和改写用户 MAT。
0000 0001h	ROM 码保护有效（ROM 码保护 2）	禁止读用户 MAT。
上述以外	ROM 码保护无效	允许读或者改写用户 MAT。

### 31.12 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。在发行恢复命令后的 20 $\mu$ s 内（PCLK=50MHz 时）不能再次发行编程 / 擦除挂起命令。

#### (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，至少等待 100 $\mu$ s 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 tRESW2（参照“33. 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。

有关编程 / 擦除中的 WDT 复位和 IWDT 复位，即使不保持上述时间也能使用。

#### (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、电压监视 1 中断和电压监视 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、复位或者 FRESETR.FRESET 位引起 FCU 复位、因检测到错误而进入命令锁定状态，以及因下述 (8) 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写操作。

#### (8) 编程 / 擦除过程中的禁止事项

在编程 / 擦除过程中为了防止损坏器件，不能进行以下的操作：

- RX62T 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 位更改运行模式。
- 通过 SCKCR 寄存器更改 PCLK 的倍增比。
- 给 PCKAR 寄存器设定不同于 PCLK 的频率。
- 转移到全模块时钟停止模式、软件待机模式和深度软件待机模式。

## 32. 数据闪存（保存数据的闪存）

RX62T 群内置用于保存最多 256K 字节代码的闪存（ROM）以及用于保存 32K 字节数据的闪存（数据闪存）。

本章说明数据闪存，有关 ROM 请参照“31. ROM（保存代码的闪存）”。

### 32.1 概要

数据闪存的规格表 32.1 所示，ROM 和数据闪存外围的框图如图 32.1 所示。

表 32.1 数据闪存的规格

项目	内容	
存储空间	数据 MAT: 32K 字节 / 8K 字节（注 1）	
经由外围总线的读操作	在进行字存取或者字节存取时，以 3 个 PCLK 周期进行读操作。	
编程 / 擦除方式	<ul style="list-style-type: none"> <li>内置对数据闪存进行改写的专用定序器（FCU）。</li> <li>能通过给 FCU 发行命令，进行数据闪存的编程或者擦除。</li> </ul>	
BGO（后台操作）功能	<ul style="list-style-type: none"> <li>在对数据闪存进行编程或者擦除的期间，CPU 能执行分配在 ROM 区 / 数据闪存以外区域的程序。</li> <li>在对数据闪存进行编程或者擦除的期间，CPU 能执行分配在 ROM 区的程序。</li> </ul>	
挂起 / 恢复功能	<ul style="list-style-type: none"> <li>中止数据闪存的编程或者擦除，CPU 能执行数据闪存区的程序（挂起）。</li> <li>在中止后，能重新开始对 ROM 进行编程或者擦除（恢复）。</li> </ul>	
编程 / 擦除单位	<ul style="list-style-type: none"> <li>数据 MAT 的编程单位：8 字节或者 128 字节</li> <li>数据 MAT 的擦除单位：2K 字节 （当数据闪存容量为 32K 字节时：16 块；当数据闪存容量为 8K 字节时：4 块）</li> </ul>	
空白检查功能	<ul style="list-style-type: none"> <li>能执行用于确认数据闪存擦除状态的空白检查命令。</li> <li>能进行空白检查的区域为 8 字节或者 2K 字节。</li> </ul>	
板上编程 (2 种)	通过引导模式改写 <ul style="list-style-type: none"> <li>使用异步串行接口（SCI1）。</li> <li>自动调整通信速度。</li> </ul> 通过用户程序中的数据闪存改写程序进行改写 <ul style="list-style-type: none"> <li>不用对系统进行复位就能改写数据闪存。</li> </ul>	
保护功能	软件保护功能	能通过 FENTRYR.FENTRYD 位、FWEPROR.FLWE[1:0] 位、DFLREK 寄存器和 DFLWEK 寄存器防止意料不到的改写（k=0、1）。
	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“33. 电特性”。	

注 1 数据闪存容量因产品种类而不同。

产品型号	数据闪存容量	数据闪存地址
R5F562TAxxxx	32K 字节	0010 0000h ~ 0010 7FFFh
R5F562T7xxxx、R5F562T6xxxx	8K 字节	0010 0000h ~ 0010 1FFFh

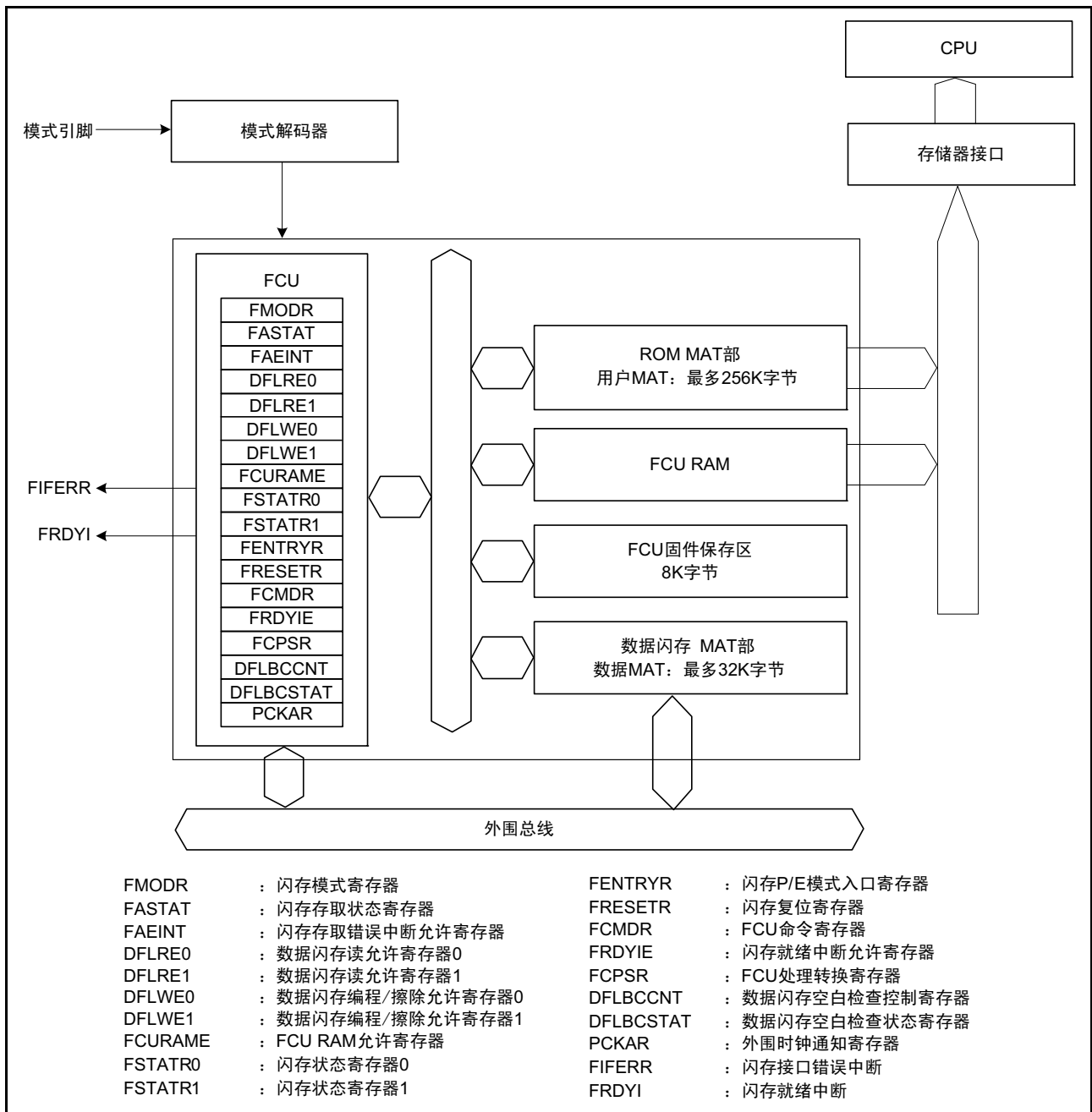


图 32.1 数据闪存的框图

数据闪存的相关输入 / 输出引脚如表 32.2 所示。

表 32.2 数据闪存的相关输入 / 输出引

引脚名	输入 / 输出	功能
PD5/RxD1	输入	用于引导模式。SCI 的接收数据（用于主机通信）。
PD3/TxD1	输出	用于引导模式。SCI 的发送数据（用于主机通信）。
MD1、MD0	输入	设定 RX62T 群的运行模式。

## 32.2 寄存器说明

数据闪存的相关寄存器一览表如表 32.3 所示。部分寄存器也有 ROM 的相关位，本章只说明数据闪存相关位的功能。有关由 ROM/ 数据闪存的通用位构成的寄存器（FRDYIE、FCURAME、FSTATR0、FSTATR1、FRESETR、FCMDR、FCPSR、PCKAR、FWEPROR）以及 ROM 专用位功能的详细内容，请参照“31. ROM（保存代码的闪存）”的“31.2 寄存器说明”。

通过复位对数据闪存的相关寄存器进行初始化。

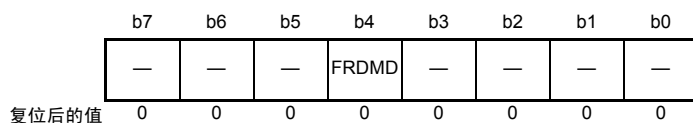
表 32.3 数据闪存的相关寄存器一览表

寄存器名	符号	复位后的值	地址	存取长度
闪存模式寄存器	FMODR	00h	007F C402h	8
闪存存取状态寄存器	FASTAT	00h	007F C410h	8
闪存存取错误中断允许寄存器	FAEINT	9Bh	007F C411h	8
闪存就绪中断允许寄存器	FRDYIE	00h	007F C412h	8
数据闪存读允许寄存器 0	DFLRE0	0000h	007F C440h	16
数据闪存读允许寄存器 1	DFLRE1	0000h	007F C442h	16
数据闪存编程 / 擦除允许寄存器 0	DFLWE0	0000h	007F C450h	16
数据闪存编程 / 擦除允许寄存器 1	DFLWE1	0000h	007F C452h	16
FCU RAM 允许寄存器	FCURAME	0000h	007F C454h	16
闪存状态寄存器 0	FSTATR0	80h	007F FFB0h	8
闪存状态寄存器 1	FSTATR1	0xh	007F FFB1h	8
闪存 P/E 模式入口寄存器	FENTRYR	0000h	007F FFB2h	16
闪存复位寄存器	FRESETR	0000h	007F FFB6h	16
FCU 命令寄存器	FCMDR	FFFFh	007F FFBAh	16
FCU 处理转换寄存器	FCPSR	0000h	007F FFC8h	16
数据闪存空白检查控制寄存器	DFLBCCNT	0000h	007F FFCAh	16
数据闪存空白检查状态寄存器	DFLBCSTAT	0000h	007F FFCEh	16
外围时钟通知寄存器	PCKAR	0000h	007F FFE8h	16
闪存编程 / 擦除保护寄存器	FWEPROR	02h	0008 C289h	8



## 32.2.1 闪存模式寄存器（FMODR）

地址 007F C402h



位	符号	位名	功能	R/W
b3-b0	—	保留位	读写值都为“0”。	R/W
b4	FRDMD	FCU 读模式选择位	0: 存储区读方式 在转移到数据闪存锁定位读模式时进行设定。 数据闪存没有锁定位，如果在转移到锁定位读模式后读数据闪存区，就会读到不确定的数据。 1: 寄存器读方式 在使用空白检查命令时进行设定。	R/W
b7-b5	—	保留位	读写值都为“0”。	R/W

FMODR 寄存器是指定锁定位的读法的寄存器。在使用空白检查命令时，必须将 FRDMD 位置“1”。在内部 ROM 无效模式中，FMODR 寄存器的读数据为“00h”，不能写。通过复位对 FMODR 寄存器进行初始化。

## FRDMD 位（FCU 读模式选择位）

此位选择是进行数据闪存锁定位读模式的转移处理还是进行空白检查处理。

在读 ROM 的锁定位时，使用 FRDMD 位选择锁定位的读法（参照“31. ROM（保存代码的闪存）”）。

## 32.2.2 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	数据闪存的编程 / 擦除保护违反位	0: 不发行数据闪存的编程 / 擦除命令（违反 DFLWEk 寄存器的设定） 1: 发行数据闪存的编程 / 擦除命令（违反 DFLWEk 寄存器的设定） (k=0、1)	R/(W) (注1)
b1	DFLRPE	数据闪存的读保护违反位	0: 不读数据闪存（违反 DFLREk 寄存器的设定） 1: 读数据闪存（违反 DFLREk 寄存器的设定） (k=0、1)	R/(W) (注1)
b2	—	保留位	读写值都为“0”。	R/W
b3	DFLAE	数据闪存的存取违反位	0: 不违反数据闪存的存取 1: 违反数据闪存的存取	R/(W) (注1)
b4	CMDLK	FCU 命令锁定位	0: FCU 不处于命令锁定状态 1: FCU 处于命令锁定状态	R
b6-b5	—	保留位	读写值都为“0”。	R/W
b7	ROMAE	ROM 存取违反位	请参照“31. ROM（保存代码的闪存）”。	R/(W) (注1)

注 1. 为了将标志置“0”，只能在读“1”后写“0”。

FASTAT 寄存器是确认有无 ROM/ 数据闪存的存取违反的寄存器。

在内部 ROM 无效模式中，FASTAT 寄存器的读数据为“00h”，不能写。如果 FASTAT 寄存器的某位变为“1”，FCU 就进入命令锁定状态（参照“32.7.2 错误保护”）。为了解除命令锁定状态，需要在将 FASTAT 寄存器置“10h”后，给 FCU 发行状态寄存器清除命令。

通过复位对 FASTAT 寄存器进行初始化。

## DFLWPE 位（数据闪存的编程 / 擦除保护违反位）

此位表示有无违反由 DFLWEk (k=0、1) 寄存器设定的编程 / 擦除保护。

[ 为“1”的条件 ]

- 对通过 DFLWEk (k=0、1) 寄存器设定为禁止编程或者擦除的数据闪存区发行编程/擦除命令。

[ 为“0”的条件 ]

- 读“1”后写“0”时

## DFLRPE 位（数据闪存的读保护违反位）

此位表示有无违反由 DFLREk (k=0、1) 寄存器设定的读保护。

[ 为“1”的条件 ]

- 对通过 DFLREk (k=0、1) 寄存器设定为禁止读的数据闪存区发行读存取命令。

[ 为“0”的条件 ]

- 读“1”后写“0”时

**DFLAE 位（数据闪存的存取违反位）**

此位表示有无数据闪存的存取违反。

如果 DFLAE 位变为“1”，FSTATR0.ILGLERR 位就变为“1”，FCU 进入命令锁定状态。

有关 FSTATR0 寄存器，请参照“31.2.5 闪存状态寄存器 0（FSTATR0）”。

[为“1”的条件]

- 在 FENTRYR.FENTRYD 位为“1”时并且在数据闪存 P/E 正常模式中，对数据闪存区发行读存取命令。
- 在 FENTRYD 位为“0”的状态下，对数据闪存区发行写存取命令。
- 在 FENTRYR.FENTRY0 位为“1”的状态下，对数据闪存区发行存取命令。

[为“0”的条件]

- 读“1”后写“0”时

**CMDLK 位（FCU 命令锁定位）**

此位表示 FCU 处于命令锁定状态（参照“32.7.2 错误保护”）。

[为“1”的条件]

- FCU 检测到错误并且转移到命令锁定状态后

[为“0”的条件]

- FCU 发行状态寄存器清除命令后

## 32.2.3 闪存存取错误中断允许寄存器（FAEINT）

地址 007F C411h

b7	b6	b5	b4	b3	b2	b1	b0
ROMAEIE	—	—	CMDLKIE	DFLAEIE	—	DFLRPEIE	DFLWPEIE
复位后的值	1	0	0	1	1	0	1

位	符号	位名	功能	R/W
b0	DFLWPEIE	数据闪存的编程 / 擦除保护违反中断允许位	0: 在 FASTAT.DFLWPE 位为“1”时，不产生 FIFERR 中断请求。 1: 在 FASTAT.DFLWPE 位为“1”时，产生 FIFERR 中断请求。	R/W
b1	DFLRPEIE	数据闪存的读保护违反中断允许位	0: 在 FASTAT.DFLRPE 位为“1”时，不产生 FIFERR 中断请求。 1: 在 FASTAT.DFLRPE 位为“1”时，产生 FIFERR 中断请求。	R/W
b2	—	保留位	读写值都为“0”。	R/W
b3	DFLAEIE	数据闪存的存取违反中断允许位	0: 在 FASTAT.DFLAE 位为“1”时，不产生 FIFERR 中断请求。 1: 在 FASTAT.DFLAE 位为“1”时，产生 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	请参照“31. ROM（保存代码的闪存）”。	R/W

FAEINT 寄存器设定允许或者禁止闪存接口错误中断（FIFERR）的输出。

在内部 ROM 无效模式中，FAEINT 寄存器的读数据为“00h”，写操作无效。

通过复位对 FAEINT 寄存器进行初始化。

## DFLWPEIE 位（数据闪存的编程 / 擦除保护违反中断允许位）

在发生数据闪存的编程 / 擦除保护违反并且 FASTAT.DFLWPE 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的产生。

## DFLRPEIE 位（数据闪存的读保护违反中断允许位）

在发生数据闪存的读保护违反并且 FASTAT.DFLRPE 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的产生。

## DFLAEIE 位（数据闪存的存取违反中断允许位）

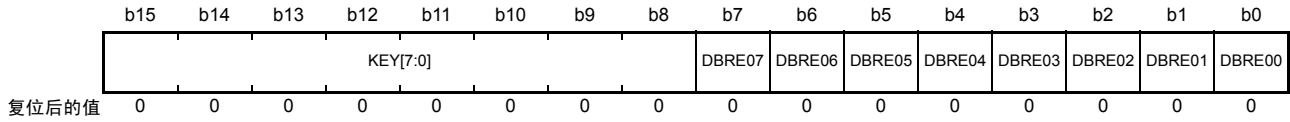
在发生数据闪存的存取违反并且 FASTAT.DFLAE 位变为“1”时，此位允许或者禁止 FIFERR 中断请求的产生。

## 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 中断请求的产生。

## 32.2.4 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	DB00 块的读允许位	0: 禁止读 1: 允许读	R/W
b1	DBRE01	DB01 块的读允许位		R/W
b2	DBRE02	DB02 块的读允许位		R/W
b3	DBRE03	DB03 块的读允许位		R/W
b4	DBRE04 (注 1)	DB04 块的读允许位		R/W
b5	DBRE05 (注 1)	DB05 块的读允许位		R/W
b6	DBRE06 (注 1)	DB06 块的读允许位		R/W
b7	DBRE07 (注 1)	DB07 块的读允许位		R/W
b15-b8	KEY[7:0]	键码	控制能否改写 DBREi 位 (i=07 ~ 00)。	R/(W) (注 2)

注 1. 在 R5F562T7xxxx、R5F562T6xxxx 中为保留位。读写值都为 "0"。

注 2. 不保持写数据。

DFLRE0 寄存器允许或者禁止读数据 MAT 的 DB07 ~ DB00 块（参照图 32.3）。

只在通过字存取给高位字节写特定值时才能写此寄存器，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，DFLRE0 寄存器的读数据为 "0000h"，不能写。

通过复位对 DFLRE0 寄存器进行初始化。

## DBREi 位（DBi 块的读允许位）（i=07 ~ 00）

此位设定允许或者禁止读数据 MAT 的 DB07 ~ DB00 块。

DBREi 位用于控制 DBi 块的读。

只在 KEY[7:0] 位为 "2Dh" 时才能通过字存取写 DBREi 位。

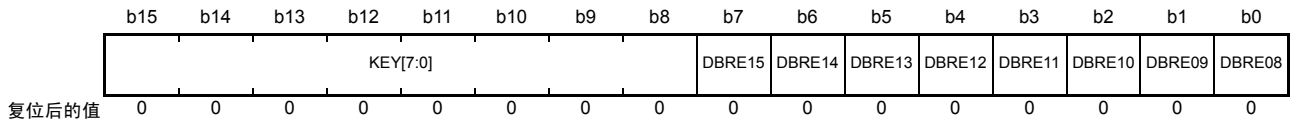
## KEY[7:0] 位（键码）

这些位控制能否改写 DBREi 位。

不保持 KEY[7:0] 位的写数据。

## 32.2.5 数据闪存读允许寄存器 1（DFLRE1）

地址 007F C442h



位	符号	位名	功能	R/W
b0	DBRE08 (注1)	DB08 块的读允许位	0: 禁止读 1: 允许读	R/W
b1	DBRE09 (注1)	DB09 块的读允许位		R/W
b2	DBRE10 (注1)	DB10 块的读允许位		R/W
b3	DBRE11 (注1)	DB11 块的读允许位		R/W
b4	DBRE12 (注1)	DB12 块的读允许位		R/W
b5	DBRE13 (注1)	DB13 块的读允许位		R/W
b6	DBRE14 (注1)	DB14 块的读允许位		R/W
b7	DBRE15 (注1)	DB15 块的读允许位		R/W
b15-b8	KEY[7:0]	键码	控制能否改写 DBREi 位 (i=15 ~ 08)。	R/(W) (注2)

注 1. 在 R5F562T7xxxx、R5F562T6xxxx 中为保留位。读写值都为“0”。

注 2. 不保持写数据。

DFLRE1 寄存器允许或者禁止读数据 MAT 的 DB15 ~ DB08 块（参照图 32.3）。

只在通过字存取给高位字节写特定值时才能写此寄存器，但是不保持高位字节的写数据。

在内部 ROM 无效模式中，DFLRE1 寄存器的读数据为“0000h”，不能写。

通过复位对 DFLRE1 寄存器进行初始化。

## DBREi 位（DBi 块的读允许位）（i=15 ~ 08）

此位设定允许或者禁止读数据 MAT 的 DB15 ~ DB08 块。

DBREi 位用于控制 DBi 块的读。

只在 KEY[7:0] 位为“D2h”时才能通过字存取写 DBREi 位。

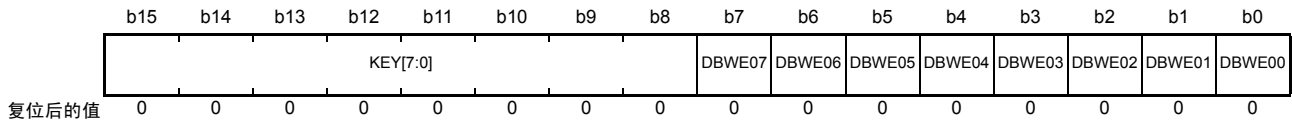
## KEY[7:0] 位（键码）

这些位控制能否改写 DBREi 位。

不保持 KEY[7:0] 位的写数据。

## 32.2.6 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	DB00 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	DB01 块的编程 / 擦除允许位		R/W
b2	DBWE02	DB02 块的编程 / 擦除允许位		R/W
b3	DBWE03	DB03 块的编程 / 擦除允许位		R/W
b4	DBWE04 (注 1)	DB04 块的编程 / 擦除允许位		R/W
b5	DBWE05 (注 1)	DB05 块的编程 / 擦除允许位		R/W
b6	DBWE06 (注 1)	DB06 块的编程 / 擦除允许位		R/W
b7	DBWE07 (注 1)	DB07 块的编程 / 擦除允许位		R/W
b15-b8	KEY[7:0]	键码	控制能否改写 DBWE <sub>i</sub> 位 (i=07 ~ 00)。	R/(W) (注 2)

注 1. 在 R5F562T7xxxx、R5F562T6xxxx 中为保留位。读写值都为“0”。

注 2. 不保持写数据。

DFLWE0 寄存器允许或者禁止对数据 MAT 的 DB07 ~ DB00 块（参照图 32.3）进行编程或者擦除。只在通过字存取给高位字节写特定值时才能写此寄存器，但是不保持高位字节的写数据。在内部 ROM 无效模式中，DFLWE0 寄存器的读数据为“0000h”，不能写。通过复位对 DFLWE0 寄存器进行初始化。

DBWE<sub>i</sub> 位（DB<sub>i</sub> 块的编程 / 擦除允许位）(i=07 ~ 00)

此位设定允许或者禁止对数据 MAT 的 DB07 ~ DB00 块进行编程或者擦除。

DBWE<sub>i</sub> 位用于控制 DB<sub>i</sub> 块的编程和擦除。

只在 KEY[7:0] 位为“1Eh”时才能通过字存取写 DBWE<sub>i</sub> 位。

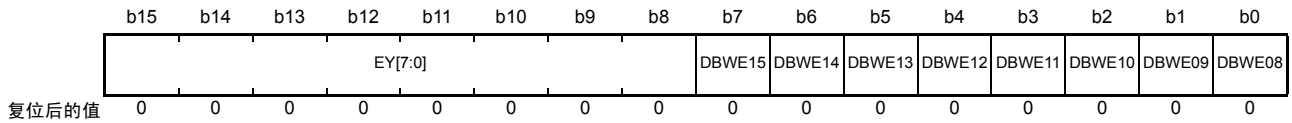
## KEY[7:0] 位（键码）

这些位控制能否改写 DBWE<sub>i</sub> 位。

不保持 KEY[7:0] 位的写数据。

## 32.2.7 数据闪存编程 / 擦除允许寄存器 1（DFLWE1）

地址 007F C452h



位	符号	位名	功能	R/W
b0	DBWE08 (注1)	DB08 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	DBWE09 (注1)	DB09 块的编程 / 擦除允许位		R/W
b2	DBWE10 (注1)	DB10 块的编程 / 擦除允许位		R/W
b3	DBWE11 (注1)	DB11 块的编程 / 擦除允许位		R/W
b4	DBWE12 (注1)	DB12 块的编程 / 擦除允许位		R/W
b5	DBWE13 (注1)	DB13 块的编程 / 擦除允许位		R/W
b6	DBWE14 (注1)	DB14 块的编程 / 擦除允许位		R/W
b7	DBWE15 (注1)	DB15 块的编程 / 擦除允许位		R/W
b15-b8	KEY[7:0]	键码	控制能否改写 DBWE <sub>i</sub> 位 (i=15 ~ 08)。	R/(W) (注2)

注 1. 在 R5F562T7xxxx、R5F562T6xxxx 中为保留位。读写值都为 "0"。

注 2. 不保持写数据。

DFLWE1 寄存器允许或者禁止对数据 MAT 的 DB15 ~ DB08 块（参照图 32.3）进行编程或者擦除。只在通过字存取给高位字节写特定值时才能写此寄存器，但是不保持高位字节的写数据。在内部 ROM 无效模式中，DFLWE1 寄存器的读数据为 "0000h"，写操作无效。通过复位对 DFLWE1 寄存器进行初始化。

DBWE<sub>i</sub> 位（DB<sub>i</sub> 块的编程 / 擦除允许位）(i=15 ~ 08)

此位设定允许或者禁止对数据 MAT 的 DB15 ~ DB08 块进行编程或者擦除。

DBWE<sub>i</sub> 位用于控制 DB<sub>i</sub> 块的编程和擦除。

只在 KEY[7:0] 位为 "E1h" 时才能通过字存取写 DBWE<sub>i</sub> 位。

## KEY[7:0] 位（键码）

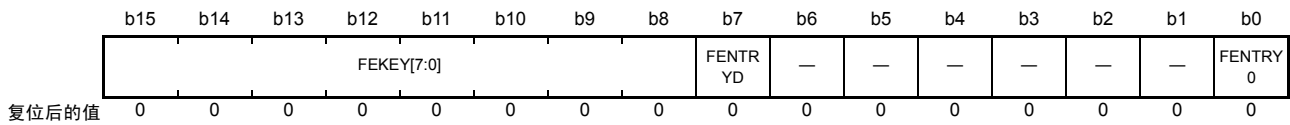
这些位控制能否改写 DBWE<sub>i</sub> 位。

不保持 KEY[7:0] 位的写数据。



## 32.2.8 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	请参照“31. 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	数据闪存 P/E 模式入口位	0: 数据闪存为读模式 1: 数据闪存为 P/E 模式	R/W
b15-b8	FEKEY[7:0]	键码	控制能否改写 FENTRYD 位和 FENTRY0 位。	R/(W) (注 1)

注 1. 不保持写数据。

FENTRYR 寄存器是将 ROM/ 数据闪存设定为 P/E 模式的寄存器。

为了将 ROM/ 数据闪存设定为 P/E 模式，使 FCU 能接收命令，需要将 FENTRYD 位或者 FENTRY0 位置“1”。但是，如果将多个位置“1”，FSTATR0.ILGLERR 位就变为“1”，FCU 进入命令锁定状态。

只在通过字存取给高位字节写特定值时才能写此寄存器，而在其他情况下写此寄存器时，此寄存器就被初始化。不保持高位字节的写数据。

在内部 ROM 无效模式中，FENTRYR 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESETR 位置“1”，对 FENTRYR 寄存器进行初始化。

有关 FSTATR0 寄存器，请参照“31.2.5 闪存状态寄存器 0（FSTATR0）”。

有关 FRESETR 寄存器，请参照“31.2.10 闪存复位寄存器（FRESETR）”。

## FENTRYD 位（数据闪存 P/E 模式入口位）

FENTRYD 位将数据闪存设定为 P/E 模式。

[ 写操作的有效条件（满足以下全部条件时） ]

- 内部 ROM 为有效模式。
- FSTATR0.FRDI 位为“1”。
- 通过字存取给 FEKEY[7:0] 位写“AAh”。

[ 为“1”的条件 ]

- 在满足写操作的有效条件并且 FENTRYR 寄存器为“0000h”的状态下，给 FENTRYD 位写“1”时

[ 为“0”的条件 ]

- 通过字节存取进行写操作时
- 在 FEKEY[7:0] 位不为“AAh”的状态下，通过字存取进行写操作时
- 在满足写操作的有效条件的状态下，给 FENTRYD 位写“0”时
- 在满足写操作的有效条件并且 FENTRYR 寄存器不为“0000h”的状态下，写 FENTRYR 寄存器时

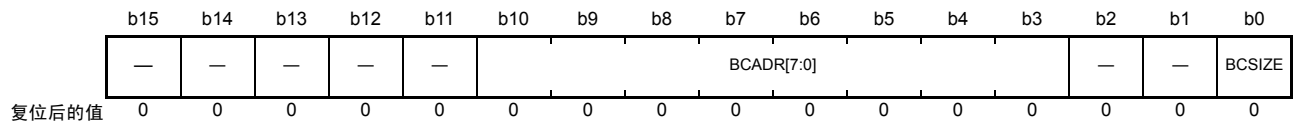
## FEKEY[7:0] 位（键码）

这些位控制能否改写 FENTRYD 位和 FENTRY0 位。

不保持 FEKEY[7:0] 位的写数据。

### 32.2.9 数据闪存空白检查控制寄存器（DFLBCCNT）

地址 007F FFCAh



位	符号	位名	功能	R/W
b0	BCSIZE	空白检查大小设定位	0: 空白检查对象区为 8 字节 1: 空白检查对象区为 2K 字节	R/W
b2-b1	—	保留位	读写值都为“0”。	R/W
b10-b3	BCADR[7:0]	空白检查地址设定位	设定检查对象区的地址。	R/W
b15-b11	—	保留位	读写值都为“0”。	R/W

DFLBCCNT 寄存器指定空白检查命令的检查对象区的地址和大小。

在内部 ROM 无效模式中，DFLBCCNT 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESETR 位置“1”，对 DFLBCCNT 寄存器进行初始化。

有关 FRESETR 寄存器，请参照“31.2.10 闪存复位寄存器（FRESETR）”。

#### BCSIZE 位（空白检查大小设定位）

此位设定空白检查命令的检查对象区的大小。

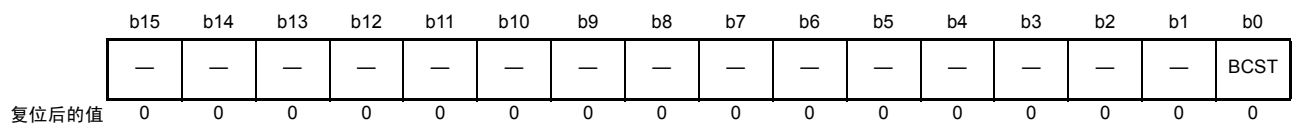
#### BCADR[7:0] 位（空白检查地址设定位）

在空白检查命令的检查对象区大小为 8 字节（BCSIZE 位为“0”）时，此位设定检查对象区的地址。

当 BCSIZE 位为“0”时，DFLBCCNT 寄存器的设定值（将 BCADR[7:0] 位的设定值向 MSB 侧移 3 位后的值）以及发行空白检查命令时指定的擦除块起始地址相加后的值为检查对象区的起始地址。

### 32.2.10 数据闪存空白检查状态寄存器（DFLBCSTAT）

地址 007F FFCEh



位	符号	位名	功能	R/W
b0	BCST	空白检查状态位	0: 空白检查对象区为擦除状态（空白） 1: 空白检查对象区为写“0”或者“1”的状态	R
b15-b1	—	保留位	读写值都为“0”。	R/W

DFLBCSTAT 寄存器保存空白检查命令的处理结果。

在内部 ROM 无效模式中，DFLBCSTAT 寄存器的读数据为“0000h”，不能写。

通过复位或者将 FRESETR.FRESETR 位置“1”，对 DFLBCSTAT 寄存器进行初始化。

有关 FRESETR 寄存器，请参照“31.2.10 闪存复位寄存器（FRESETR）”。

#### BCST 位（空白检查状态位）

此位表示空白检查的结果。

### 32.3 数据闪存的存储器 MAT 结构

RX62T 群的数据闪存由最多 32K 字节的数据 MAT 构成，数据 MAT 的地址如图 32.2 所示。在读、编程和擦除时，数据 MAT 的地址相同。

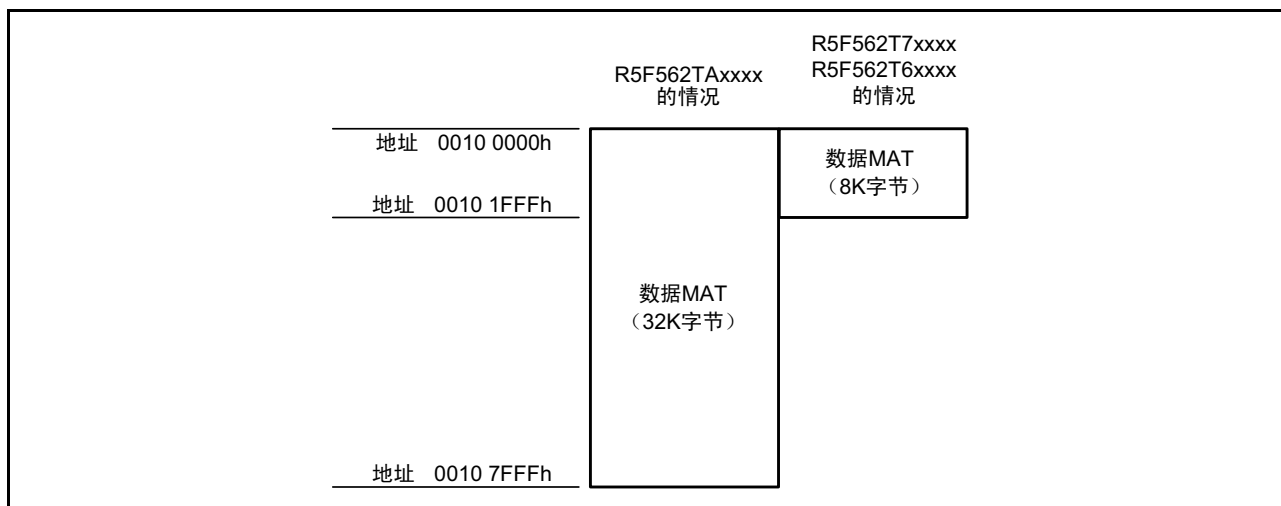


图 32.2 数据闪存的数据 MAT 结构

### 32.4 块结构

数据 MAT 的擦除块结构如图 32.3 所示。数据 MAT 被分割为 2K 字节（16 块），以块为单位进行擦除，以 8 字节或者 128 字节为单位进行编程。以低位地址的 3 位全部为“0”的 8 字节为单位进行 8 字节编程；以低位地址为“00h”或者“80h”开始的 128 字节为单位进行 128 字节编程。

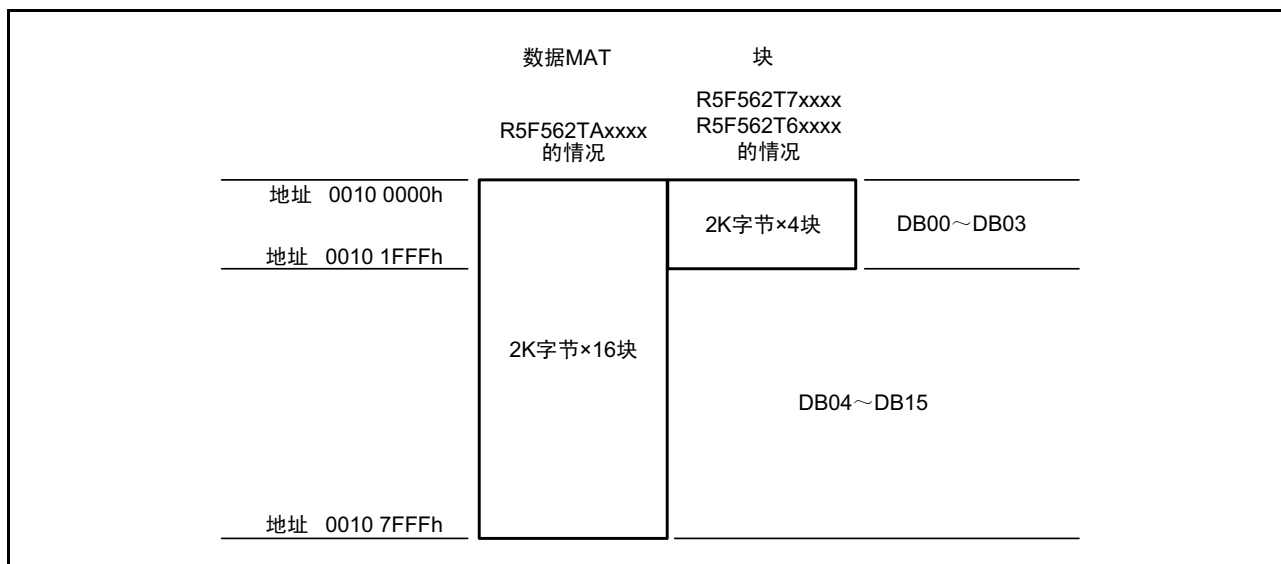


图 32.3 数据 MAT 的块分割

### 32.5 数据闪存相关的运行模式

有关运行模式的转移，请参照“31.5 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 MAT 的读、编程或者擦除。

各模式的不同点如表 32.4 所示。

表 32.4 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 MAT	数据 MAT	数据 MAT
块分割擦除	○（注 1）	○
复位时的启动 MAT	嵌入式程序存储 MAT（注 2）	用户 MAT

注 1. 有可能在启动时被全部擦除。此后能擦除特定块，详细内容请参照“31.9.2 ID 码保护”。

注 2. 用户不能使用。

- 在引导模式中，主机能经由 SCI 对数据 MAT 进行读和编程。
- 在引导模式中，通过引导模式的嵌入式程序使用内部 RAM。因此，不保持内部 RAM 的数据。

## 32.6 数据闪存的编程 / 擦除

通过给编程 / 擦除的专用定序器（FCU）发行命令（FCU 命令），对数据闪存进行编程和擦除。FCU 有 5 种模式，通过在模式转移后发行编程和擦除命令，进行编程和擦除。

以下说明数据闪存的编程和擦除所需的模式转移和命令体系。这些内容在引导模式、单芯片模式（内部 ROM 有效）中通用。

### 32.6.1 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 32.4 所示。

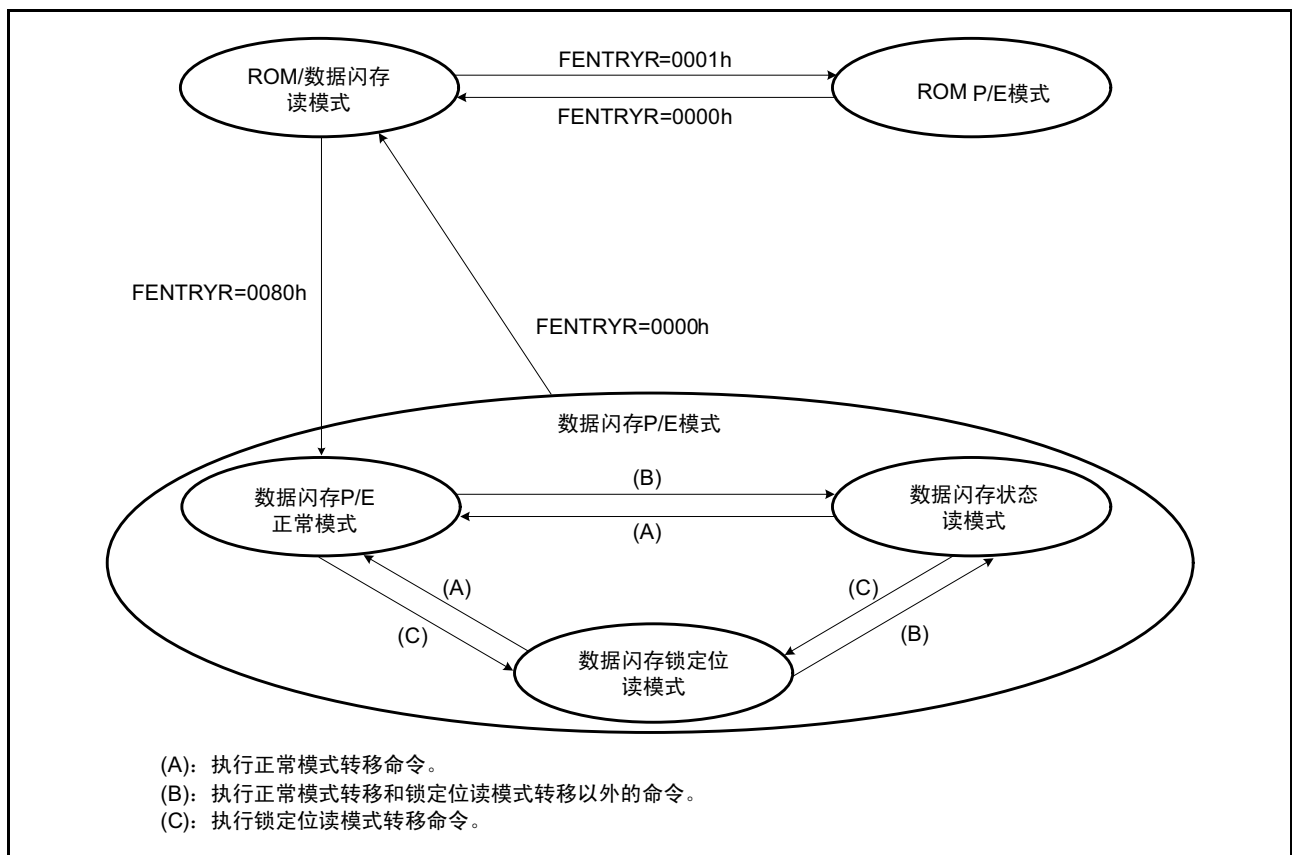


图 32.4 FCU 的模式转移图（数据闪存相关）

#### 32.6.1.1 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“31.6.1.2 ROM P/E 模式”。

#### 32.6.1.2 ROM/ 数据闪存读模式

ROM/ 数据闪存读模式是能读 ROM 和数据闪存的模式，不接受 FCU 命令。如果将 FENTRYR.FENTRYD 位置“0”并且将 FENTRYR.FENTRY0 位置“0”，就转移到此模式。

### 32.6.1.3 数据闪存 P/E 模式

数据闪存 P/E 模式是对数据闪存进行编程和擦除的模式，不能读数据闪存。

数据闪存 P/E 模式有数据闪存 P/E 正常模式、数据闪存状态读模式、数据闪存锁定位读模式共 3 种。

#### (1) 数据闪存 P/E 正常模式

数据闪存 P/E 正常模式是在对数据闪存进行编程或者擦除时最先转移的模式。如果在 ROM/ 数据闪存读模式中将 FENTRYR.FENTRYD 位置“1”并且将 FENTRYR.FENTRY0 位置“0”，或者在数据闪存 P/E 模式中接受正常模式转移命令，就转移到此模式。能接受的命令如表 32.7 所示。

如果读数据闪存区，就产生数据闪存的存取违反，FCU 进入命令锁定状态。能高速读 ROM。

#### (2) 数据闪存状态读模式

数据闪存状态读模式是能读数据闪存状态的模式。

如果在数据闪存 P/E 模式中接受正常模式转移和锁定位读模式转移以外的命令，就转移到此模式。FSTATR0.FRDY 位为“0”的状态以及发生错误后的命令锁定状态也是数据闪存状态读模式中的状态。能接受的命令如表 32.7 所示。

如果读数据闪存区，就读 FSTATR0 寄存器的值。能高速读 ROM。

#### (3) 数据闪存锁定位读模式

数据闪存锁定位读模式是能读数据闪存锁定位的模式。但是，因为数据闪存没有锁定位，所以不能读锁定位。

如果在数据闪存 P/E 模式中接受锁定位读模式转移命令，就转移到数据闪存锁定位读模式。能接受的命令如表 32.7 所示。

数据闪存没有锁定位，如果读数据闪存区，读数据就为不定值，但是不发生数据闪存的存取违反。能高速读 ROM。

### 32.6.2 FCU 命令一览表

FCU 命令有 FCU 模式转移命令和编程 / 擦除命令。对数据闪存进行编程或者擦除时能使用的 FCU 命令一览表如表 32.5 所示。

表 32.5 FCU 命令一览表（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“32.6.3 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“32.6.3 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“32.6.3 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	对数据闪存进行编程（以 8 字节或者 128 字节为单位）。
块擦除	擦除数据闪存（以块为单位）。
P/E 挂起	中止编程和擦除。
P/E 恢复	重新开始编程和擦除。
状态寄存器清除	清除 FSTATR0.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 命令兼用数据闪存的空白检查命令。如果对 ROM 发行空白检查命令，就读 ROM 的锁定位。

通过写数据闪存区给 FCU 发行命令。

编程命令和空白检查命令的格式如表 32.6 所示。有关编程命令和空白检查命令以外的 FCU 命令格式，请参照“31. ROM（保存代码的闪存）”的“31.6.2 FCU 命令一览表”。

如果在特定条件下进行如表 32.6 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“32.6.3 FCU 的模式和命令的关系”和“32.6.4 FCU 命令的使用方法”。

表 32.6 FCU 命令的格式（数据闪存专用命令）

命令	总线周期数	第 1 周期		第 2 周期		第 3 周期		第 4 ~ 第 N+2 周期		第 N+3 周期	
		地址	数据	地址	数据	地址	数据	地址	数据	地址	数据
编程 (8 字节编程: N=4)	7	EA	E8h	EA	04h	WA	WDn	EA	WDn	EA	D0h
编程 (128 字节编程: N=64)	67	EA	E8h	EA	40h	WA	WDn	EA	WDn	EA	D0h
空白检查	2	EA	71h	BA	D0h	—	—	—	—	—	—

地址列 EA: 数据闪存区的地址

0010 0000h ~ 0010 1FFFh 的任意地址

WA: 8 字节或者 128 字节的起始地址

BA: 数据闪存的擦除块地址

擦除对象块内的任意地址

数据列 WDn: 编程数据的第 n 个字 (n=1 ~ N)

### 32.6.3 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 32.7 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“32.7.2 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 位的值确认有无发生错误，FASTAT.CMDLK 位的值是 FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位和 FSTATR1.FCUERR 位的值的逻辑或。

表 32.7 FCU 的模式 / 状态和能接受的命令的关系（数据闪存 P/E 模式）

	P/E 正常模式			状态读模式						锁定位读模式			
	编程挂起中	擦除挂起中	其他状态	编程 / 擦除中止处理中	编程 / 擦除中止处理中	空白检查处理中	编程挂起中	擦除挂起中	命令锁定状态	其他状态	编程挂起中	擦除挂起中	其他状态
FSTATR0.FRDY 位	1	1	1	0	0	0	1	1	0/1	1	1	1	1
FSTATR0.SUSRDY 位	0	0	0	1	0	0	0	0	0	0	0	0	0
FSTATR0.ERSSPD 位	0	1	0	0	0/1	0	0	1	0	0	0	1	0
FSTATR0.PRGSPD 位	1	0	0	0	0/1	0	1	0	0	0	1	0	0
FASTAT.CMDLK 位	0	0	0	0	0	0	0	0	1	0	0	0	0
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



### 32.6.4 FCU 命令的使用方法

FCU 命令的使用方法有：使用编程命令和块擦除命令对数据闪存进行编程和擦除的方法以及使用空白检查命令确认数据闪存的擦除状态的方法。有关 FCU RAM 的固件传送方法以及其他 FCU 命令的使用方法，请参照“31. ROM（保存代码的闪存）”的“31.6.4 FCU 命令的使用方法”。

#### (1) 外围时钟通知命令的使用方法

通知外围时钟的频率，详细内容请参照“31. ROM（保存代码的闪存）”的“31.6.4 FCU 命令的使用方法”。必须在将 FENTRYR.FENTRYD 位置“1”后设定数据闪存区内的地址。

#### (2) 编程方法

使用编程命令对数据闪存进行数据编程。

在编程命令的第 1 周期，通过字节存取将“E8h”写到数据闪存区的地址，在第 2 周期，通过字节存取将编程字数（N）（注 1）写到数据闪存区的地址。在命令的第 3～第 N+2 周期，必须通过字存取写编程数据。在第 3 周期的存取中，必须将编程数据写到编程对象区的起始地址。在进行 8 字节编程时，必须将起始地址调整为以 8 字节为边界的地址，在进行 128 字节编程时，必须将起始地址调整为以 128 字节为边界的地址。如果在通过 N 次字存取将编程数据写到数据闪存区的地址后，在第 N+3 周期通过字节存取将“D0h”写到数据闪存区的地址，FCU 就开始对数据闪存进行编程处理。能通过 FSTATR0.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”。如果要在将通过 DFLWEk（k=0、1）寄存器控制的编程 / 擦除保护功能设定为无效后进行编程，就必须将编程对象块的编程 / 擦除允许位置“1”。

数据闪存的编程方法如图 32.5 所示。

注 1. 当进行 8 字节编程时，N 为“04h”；当进行 128 字节编程时，N 为“40h”。

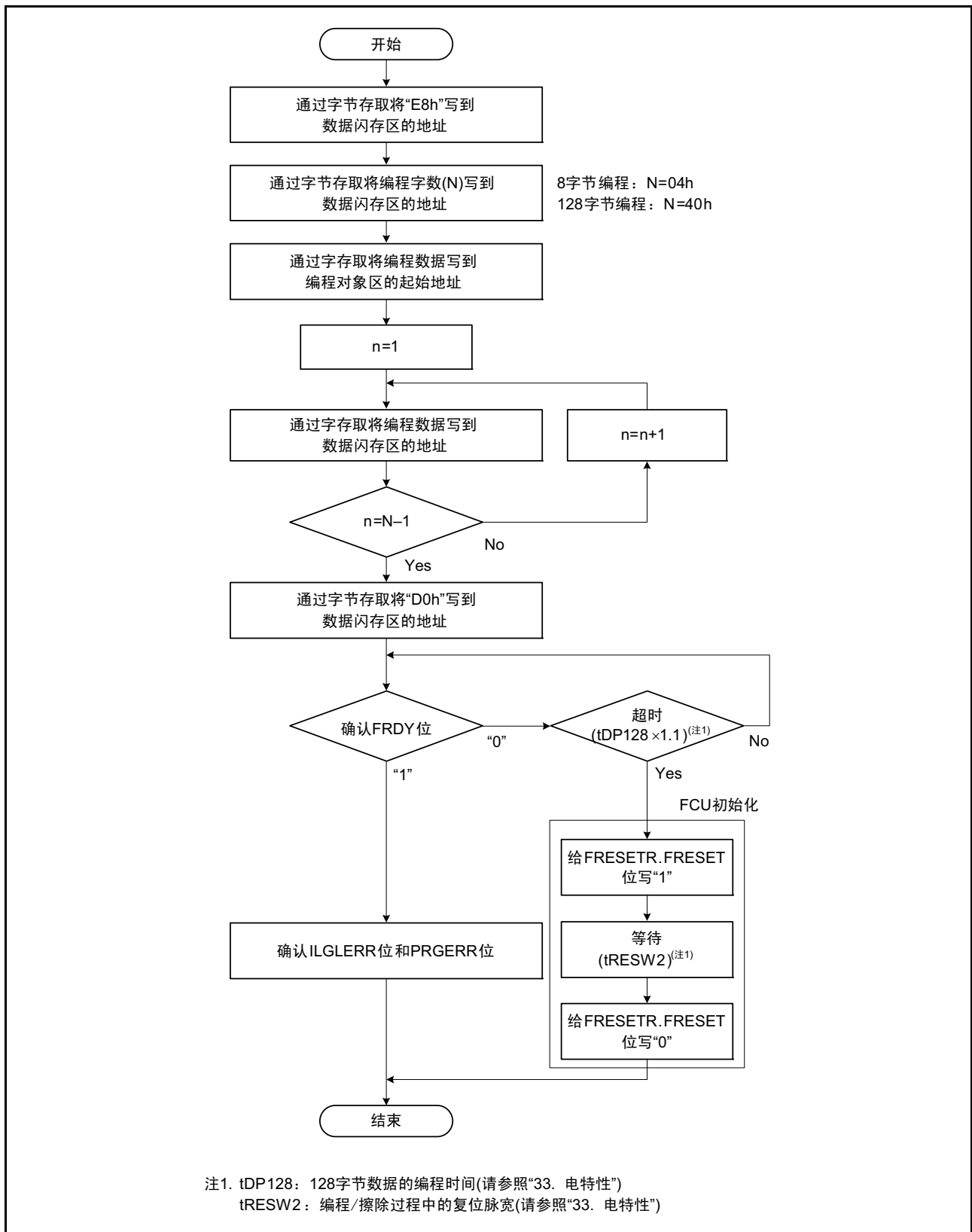


图 32.5 数据闪存的编程

### (3) 擦除方法

使用块擦除命令来擦除数据闪存。使用块擦除命令的擦除方法和 ROM 的擦除方法相同（参照“31. ROM（保存代码的闪存）”）。

必须注意：数据闪存具有 DFLWEk（k=0、1）寄存器控制的编程 / 擦除保护功能。如果要在将通过 DFLWEk 寄存器控制的保护功能设定为无效后进行擦除，就必须将擦除对象块的编程 / 擦除允许位置“1”。

### (4) 空白检查

即使通过 CPU 读擦除状态的数据闪存，读取值也为不定值，因此需要使用空白检查命令确认擦除状态。在使用空白检查命令时，必须预先将 FMODR.FRDM 位置“1”，使空白检查命令处于可使用的状态，然后给 DFLBCCNT 寄存器设定检查对象区的大小和地址。当 DFLBCCNT.BCSIZE 位为“1”时，能在空白检查命令的第 2 周期对指定的整个擦除块（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 8 字节区域进行空白检查。在空白检查命令的第 1 周期，通过字节存取将“71h”写到数据闪存区的地址。在命令的第 2 周期，如果通过字节存取将“D0h”写到包括空白检查对象区的擦除块内的任意地址，FCU 就开始进行数据闪存空白检查处理。能通过 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。

数据闪存空白检查方法如图 32.6 所示。

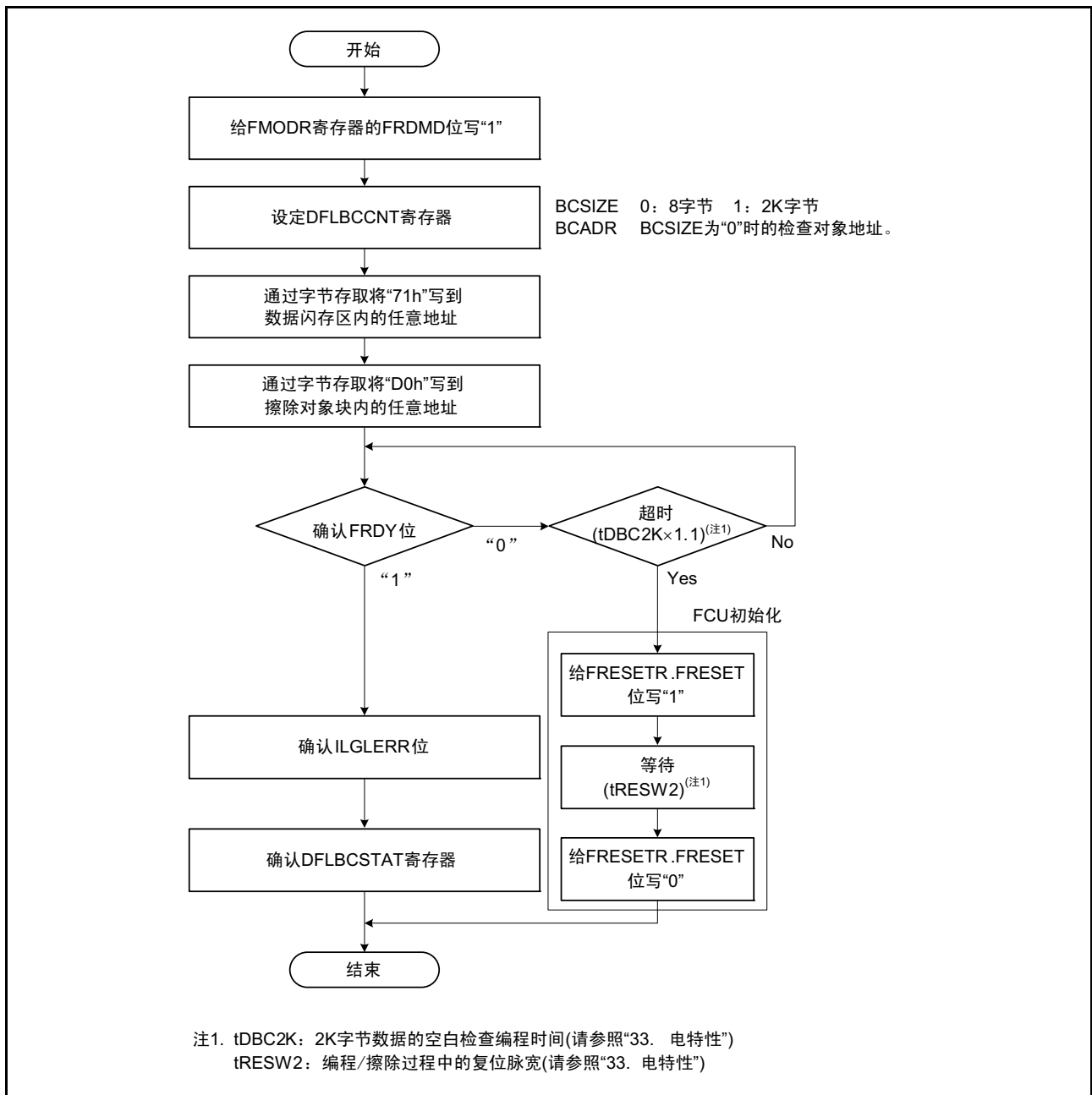


图 32.6 数据闪存的空白检查

## 32.7 保护

数据闪存的编程 / 擦除的保护有软件保护和错误保护。

### 32.7.1 软件保护

软件保护是通过设定控制寄存器禁止对数据闪存进行编程、擦除和读的状态。如果违反软件保护而对数据闪存发行编程 / 擦除命令或者读数据闪存，FCU 就检测到错误，进入命令锁定状态。

#### (1) 通过 FWEPROR 寄存器进行的保护

如果不将 FWEPROR.FLWE[1:0] 位置“01b”，就不能在任何模式中进行改写。

#### (2) 通过 FENTRYR 寄存器进行的保护

当 FENTRYR.FENTRYD 位为“0”时，FCU 进入 ROM/ 数据闪存读模式。因为在 ROM/ 数据闪存读模式中不接受 FCU 命令，所以数据闪存为禁止编程和擦除的状态。如果在 ROM/ 数据闪存读模式中对数据闪存发行 FCU 命令，FCU 就检测到非法命令错误，进入命令锁定状态（参照“32.7.2 错误保护”）。

#### (3) 通过 DFLWEk 寄存器进行的保护

当 DFLWEk.DBWEi (k=0、1, i=15 ~ 00) 位为“0”时，数据 MAT 的 DBi 块为禁止编程和擦除的状态。如果在 DBWEi 位为“0”的状态下对 DBi 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“32.7.2 错误保护”）。

#### (4) 通过 DFLREk 寄存器进行的保护

当 DFLREk.DBREi (k=0、1, i=15 ~ 00) 位为“0”时，数据 MAT 的 DBi 块为禁止读的状态。如果在 DBREi 位为“0”的状态下读 DBi 块，FCU 就检测到读保护违反，进入命令锁定状态（参照“32.7.2 错误保护”）。

### 32.7.2 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 位和 FASTAT.DFLWPE 位）中的某位或者多个位就为“1”，禁止对数据闪存进行编程和擦除。要解除命令锁定状态时，需要在 FASTAT 寄存器为“10h”的状态下发行状态寄存器清除命令。

当 FAEINT.CMDLKIE 位为“1”时，如果 FCU 进入命令锁定状态（FASTAT.CMDLK 位为“1”），就发生闪存接口错误（FIFERR）中断。当 FAEINT 寄存器的数据闪存相关的中断允许位（DFLAEIE 位、DFLRPEIE 位和 DFLWPEIE 位）为“1”时，即使 FASTAT 寄存器的对应位（DFLAE 位、DFLRPE 位和 DFLWPE 位）变为“1”，也发生 FIFERR 中断。

数据闪存相关的错误保护内容和错误检测后的状态位的值（FSTATR0.ILGLERR 位、FSTATR0.ERSERR 位、FSTATR0.PRGERR 位、FASTAT.DFLAE 位、FASTAT.DFLRPE 位和 FASTAT.DFLWPE 位）的关系如表 32.8 所示。有关 ROM/ 数据闪存通用的错误保护内容（FENTRYR 设定错误、大部分非法命令、擦除错误、编程错误和 FCU 错误），请参照“31. ROM（保存代码的闪存）”的“31.8.2 错误保护”。

如果在编程或者擦除处理过程中发行挂起以外的命令，就转移到命令锁定状态，但是 FCU 继续进行编程或者擦除处理。在此状态下无法通过发行 P/E 挂起命令来中止编程或者擦除。如果在命令锁定状态下发行命令，ILGLERR 位就变为“1”。

表 32.8 错误保护一览表（数据闪存专用）

分类	内容	ILGLERR	ERSERR	PRGERR	DFLAE	DFLRPE	DFLWPE	CMDLK
非法命令	在编程命令的第 2 周期指定“04h”和“40h”以外的数据。	1	0	0	0	0	0	1
	在 FENTRYR.FENTRYD 位为“1”的状态下，对数据闪存区发行锁定位编程命令。	1	0	0	0	0	0	1
数据闪存的存取违反	在 FENTRYD 位为“1”时并且在数据闪存 P/E 正常模式中，对数据闪存区发行读存取命令。	1	0	0	1	0	0	1
	在 FENTRYD 位为“0”的状态下，对数据闪存区发行写存取命令。	1	0	0	1	0	0	1
	在 FENTRYR.FENTRY0 位为“1”的状态下，对数据闪存区发行存取命令。	1	0	0	1	0	0	1
数据闪存的读保护违反	对通过 DFLREk 寄存器（k=0、1）设定为禁止读的数据闪存区发行读存取命令。	1	0	0	0	1	0	1
数据闪存的写保护违反	对通过 DFLWEk 寄存器（k=0、1）设定为禁止编程和擦除的数据闪存区发行编程 / 块擦除命令。	1	0	0	0	0	1	1

## 32.8 引导模式

在引导模式中，能从主机发送控制命令和编程数据，并且对数据 MAT 进行编程或者擦除。有关引导模式的系统结构和使用方法的详细内容，请参照“31.9 引导模式”。以下说明数据闪存固有的命令。

### 32.8.1 查询设定主机命令

数据闪存固有的查询设定主机命令一览表如表 32.9 所示。在“31.9.5 查询设定主机命令的等待状态”和“图 31.29 用户 MAT 的查询设定主机命令的使用例子”所示的流程中，“查询 MAT 编程信息”使用数据 MAT 有无查询命令和数据 MAT 信息查询命令。

表 32.9 查询设定主机命令（数据闪存固有）

主机命令名	功能
数据 MAT 有无查询	查询是否有数据 MAT。
数据 MAT 信息查询	查询数据 MAT 的个数、起始地址和结束地址。

以下说明各命令的详细内容。说明内容中的“命令”是从主机发送到 RX62T 的命令，“应答”是从 RX62T 发送到主机的应答。“校验和”是指在合计 RX62T 发送的各字节时，为了得到“00h”而算出的字节数据。

## (1) 数据 MAT 有无查询

如果主机发送数据 MAT 有无查询命令，RX62T 就发送表示有数据 MAT 的信息。

命令	2Ah			
应答	3Ah	长度	有无 MAT	SUM

- 【符号说明】
- 长度（1 字节）：有无 MAT 的字符数（是固定值，为“1”。）
  - MAT 有无（1 字节）：有无数据 MAT（是固定值，为“21h”。）  
21h: 有数据 MAT
  - SUM（1 字节）：校验和

## (2) 数据 MAT 信息查询

如果主机发送数据 MAT 信息查询命令，RX62T 就发送数据 MAT 的区域数和地址的信息。

命令	2Bh		
应答	3Bh	长度	区域数
	区域的起始地址		
	区域的结束地址		
	区域的起始地址		
	区域的结束地址		
	...		
	区域的起始地址		
	区域的结束地址		
	SUM		

- 【符号说明】
- 长度（1 字节）：区域数、区域的起始地址、区域的结束地址的数据总字节数
  - 区域数（1 字节）：数据 MAT 的区域数（连续区域为 1 个区域）
  - 区域的起始地址（4 字节）：数据 MAT 区域的起始地址
  - 区域的最后地址（4 字节）：数据 MAT 区域的结束地址
  - SUM（1 字节）：校验和

数据 MAT 的块结构信息包括在擦除块信息查询命令（参照“31.9.5 查询设定主机命令的等待状态”）的应答中。

### 32.8.2 编程 / 擦除主机命令

数据闪存固有的编程 / 擦除主机命令一览表如表 32.10 所示。数据闪存固有的主机命令只有数据 MAT 校验和命令以及数据 MAT 空白检查命令，而编程命令、擦除命令、读命令和 ROM 通用。

在对数据 MAT 进行编程时，在发行用户 MAT 编程选择命令后，用 256 字节编程命令给编程地址指定数据 MAT 的地址。在擦除数据 MAT 时，在发行擦除选择命令后，用块擦除命令指定数据 MAT 的擦除块。数据 MAT 的擦除块信息包括在擦除块查询命令的应答中。在读数据 MAT 时，用存储器读命令选择用户 MAT 并且给读对象地址指定数据 MAT 的地址。

有关用户 MAT 编程选择命令、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“31.9.7 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“31.9.5 查询设定主机命令的等待状态”。

表 32.10 编程 / 擦除主机命令（数据闪存固有）

主机命令名	功能
数据 MAT 校验和	数据 MAT 的校验和
数据 MAT 空白检查	数据 MAT 的空白检查

以下说明各命令的详细内容。说明内容中的“命令”是从主机发送到 RX62T 的命令，“应答”是从 RX62T 发送到主机的应答。“校验和”是指在合计 RX62T 发送的各字节时，为了得到“00h”而算出的字节数据。

#### (1) 数据 MAT 校验和

如果主机发送数据 MAT 校验和命令，RX62T 就发送以字节为单位将数据 MAT 的数据相加后的结果（校验和）。

命令

应答	<input type="text" value="71h"/>	长度	<input type="text" value="MAT 校验和"/>	SUM
----	----------------------------------	----	--------------------------------------	-----

【符号说明】 长度（1 字节）：MAT 校验和的字节数（是固定值，为“4”。）  
 MAT 校验和（4 字节）：数据 MAT 的校验和结果  
 SUM（4 字节）：校验和（应答数据的校验和）

#### (2) 数据 MAT 空白检查

如果主机发送数据 MAT 空白检查命令，RX62T 就检查数据 MAT 是否为全部擦除状态。如果数据 MAT 为全部擦除状态，RX62T 就发送应答（06h）。如果数据 MAT 有未擦除的区域，RX62T 就发送错误应答（E2h → 52h）。

命令

应答

错误应答	<input type="text" value="E2h"/>	<input type="text" value="52h"/>
------	----------------------------------	----------------------------------



## 32.9 使用时的注意事项

### (1) 启动复位后的数据 MAT 保护状态

因为 DFLREk 寄存器和 DFLWEk 寄存器（k=0、1）的初始值为“0000h”，所以启动复位后的数据 MAT 为禁止读、编程和擦除的状态。如果需要读数据 MAT，就必须在设定 DFLREk 寄存器后存取数据 MAT。如果需要数据 MAT 进行编程或者擦除，就必须在设定 DFLWEk 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“31. ROM（保存代码的闪存）”的“31.12 使用时的注意事项”。但是，在数据闪存的情况下，编程 / 擦除的注意事项里有空白检查。

将“编程 / 擦除”改读为“编程 / 擦除 / 空白检查”。

## 33. 电特性

### 33.1 绝对最大额定值

表 33.1 绝对最大额定值

项目	符号	额定值	单位
电源电压	VCC、PLLVCC	-0.3 ~ +6.5	V
输入电压（端口 4 ~ 6 除外）	V <sub>in</sub>	-0.3 ~ VCC+0.3	V
输入电压（端口 4）	V <sub>in</sub>	-0.3 ~ AVCC0+0.3	V
输入电压（端口 5、6）	V <sub>in</sub>	-0.3 ~ AVCC+0.3	V
模拟电源电压	AVCC0、AVCC（注1）	-0.3 ~ +6.5	V
基准电源电压	VREFH0（注1）	-0.3 ~ AVCC0+0.3	V
	VREF（注1）	-0.3 ~ AVCC+0.3	V
模拟输入电压（端口 4）	V <sub>AN</sub>	-0.3 ~ AVCC0+0.3	V
模拟输入电压（端口 5、6）	V <sub>AN</sub>	-0.3 ~ AVCC+0.3	V
工作温度	T <sub>opr</sub>	-40 ~ +85	°C
保存温度	T <sub>stg</sub>	-55 ~ +125	°C

【使用时的注意事项】如果在超过绝对最大额定值的状态下使用 LSI，就可能造成 LSI 的永久性破坏。

注 1. 在不使用 A/D 转换器时，不能将 AVCC0、VREFH0、VREFL0、AVSS0、AVCC、VREF、AVSS 引脚置为开路。AVCC0、VREFH0、AVCC、VREF 引脚必须连接 VCC，AVSS0、VREFL0、AVSS 引脚必须连接 VSS。

• 不使用 12 位 A/D 转换器时

AVCC0 引脚必须连接 AVCC（64 引脚版必须连接 VCC）、VREFH0 引脚必须连接 VREF（80 引脚版必须连接 AVCC、64 引脚版必须连接 VCC）、AVSS0、VREFL0 引脚必须连接 VSS。

• 不使用 10 位 A/D 转换器时

AVCC 引脚必须连接 AVCC0、VREF 引脚必须连接 VREFH0、AVSS 引脚必须连接 AVSS0。

• 不使用 12 位 A/D 转换器和 10 位 A/D 转换器时

AVCC0、VREFH0、AVCC 和 VREF 引脚必须连接 VCC、AVSS0、VREFL0 和 AVSS 引脚必须连接 VSS。

## 33.2 DC 特性

表 33.2 DC 特性 (1)

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
T<sub>a</sub>=-40 ~ +85°C。T<sub>a</sub> 在条件 1 ~ 3 中通用。

项目	符号	min	typ	max	单位	测量条件
施密特触发 输入电压	CAN 输入引脚	V <sub>IH</sub>	VCC×0.8	—	VCC+0.3	V
	IRQ 输入引脚	V <sub>IL</sub>	-0.3	—	VCC×0.2	
	MTU3 输入引脚	ΔV <sub>T</sub>	VCC×0.06	—	—	
	POE3 输入引脚					
	SCI 输入引脚					
	A/D 触发输入引脚					
	NMI 输入引脚					
	GPT 输入引脚					
	LIN 输入引脚					
	RES#					
	RIIC 输入引脚 (IICBus 运行时)					
		V <sub>IL</sub>	-0.3	—	VCC×0.3	
		ΔV <sub>T</sub>	VCC×0.05	—	—	
	端口 4 (注 1) (模拟兼用引脚)	V <sub>IH</sub>	AVCC0×0.8	—	AVCC0+0.3	
		V <sub>IL</sub>	-0.3	—	AVCC0×0.2	
		ΔV <sub>T</sub>	AVCC0× 0.06	—	—	
	端口 5、6 (注 1) (模拟兼用引脚)	V <sub>IH</sub>	AVCC×0.8	—	AVCC+0.3	
		V <sub>IL</sub>	-0.3	—	AVCC×0.2	
		ΔV <sub>T</sub>	AVCC×0.06	—	—	
	端口 1 ~ 3 (注 1) 端口 7 ~ B (注 1) 端口 D、E、G (注 1)	V <sub>IH</sub>	VCC×0.8	—	VCC+0.3	
		V <sub>IL</sub>	-0.3	—	VCC×0.2	
ΔV <sub>T</sub>		VCC×0.06	—	—		
输入 High 电平 电压 (施密特 触发输入引脚 除外)	MD 引脚、EMLE	V <sub>IH</sub>	VCC×0.9	—	VCC+0.3	V
	EXTAL RSPI 输入引脚		VCC×0.8	—	VCC+0.3	
	RIIC 输入引脚 (SMBus 运行时)		2.1	—	VCC+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 <sub>IL</sub>	-0.3	—	VCC×0.1	V
	EXTAL RSPI 输入引脚		-0.3	—	VCC×0.2	
	RIIC 输入引脚 (SMBus 运行时)		-0.3	—	0.8	

表 33.2 DC 特性 (1)

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLCC=2.7 ~ 3.6V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLCC=2.7 ~ 3.6V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLCC=4.0 ~ 5.5V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
 $T_a = -40 \sim +85^\circ\text{C}$ 。 $T_a$  在条件 1 ~ 3 中通用。

项目		符号	min	typ	max	单位	测量条件	
输出 High 电平电压	全部输出引脚 (P71 ~ P76、P90 ~ P95 除外)	$V_{OH}$	VCC-0.5	—	—	V	$I_{OH} = -1\text{mA}$	
	P71 ~ P76		VCC-0.5	—	—		$I_{OH} = -1\text{mA}$ 64 引脚 LQFP 版的 条件 3 时	
			VCC-1.0	—	—		$I_{OH} = -5\text{mA}$ 64 引脚 LQFP 版的 条件 3 以外时	
	P90 ~ P95		VCC-0.5	—	—		$I_{OH} = -1\text{mA}$ 80 引脚 LQFP 版以及 64 引脚 LQFP 版时	
			VCC-1.0	—	—		$I_{OH} = -5\text{mA}$ 112 引脚 LQFP 版以及 100 引脚 LQFP 版时	
输出 Low 电平电压	全部输出引脚 (P71 ~ P76、P90 ~ P95 和 RIIC 引脚除外)	$V_{OL}$	—	—	0.5	V	$I_{OL} = 1.0\text{mA}$	
	P71 ~ P76		—	—	0.5		$I_{OL} = 1.0\text{mA}$ 64 引脚 LQFP 版的 条件 3 时	
			—	—	1.1		$I_{OL} = 15\text{mA}$ 条件 1、2 时	
			—	—	1.4		$I_{OL} = 15\text{mA}$ 64 引脚 LQFP 版以外的 条件 3 时	
	P90 ~ P95		—	—	0.5		$I_{OL} = 1.0\text{mA}$ 80 引脚 LQFP 版以及 64 引脚 LQFP 版时	
			—	—	1.1		$I_{OL} = 15\text{mA}$ 112 引脚 LQFP 版以及 100 引脚 LQFP 版的条件 1、2 时	
			—	—	1.4		$I_{OL} = 15\text{mA}$ 112 引脚 LQFP 版以及 100 引脚 LQFP 版的条件 3 时	
	RIIC 引脚		—	—	0.4		$I_{OL} = 3\text{mA}$	
			—	—	0.6		$I_{OL} = 6\text{mA}$	
	输入泄漏电流		RES#、MD 引脚、EMLE	$I_{in}$	—		—	1.0
三态泄漏电流 (OFF 状态)	端口 1 ~ A 端口 PB0、PB3 ~ PB7 端口 D、E、G	$I_{TSIL}$	—	—	—	1.0	$\mu\text{A}$	$V_{in} = 0\text{V}$ 、 $V_{in} = \text{VCC}$
	端口 PB1、PB2		—	—	5.0			
输入电容	全部输入引脚 (端口 PB1、PB2 除外)	$C_{in}$	—	—	—	15	pF	$V_{in} = 0\text{V}$ 、 $f = 1\text{MHz}$ 、 $T_a = 25^\circ\text{C}$
	端口 PB1、PB2		—	—	30			

注 1. 包括兼用输入引脚。但是端口 PB1、PB2 作为 RIIC 输入引脚使用时，和端口 P22 ~ P24、P30、PA3 ~ PA5、PB0、PDO ~ PD2、PD6 作为 RSPI 输入引脚使用时除外。

表 33.3 DC 特性 (2)

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
T<sub>a</sub>=-40 ~ +85°C。T<sub>a</sub>在条件 1 ~ 3 中通用。

项目		符号	min	typ	max	单位	测量条件	
消耗电流 (注1)	运行时	I <sub>CC</sub> (注3)	最大运行时 (注2)	—	—	70	mA	ICLK=100MHz PCLK=50MHz
			正常运行时 (注4)	—	35	—		
			因 BGO 运行 (注5) 而增加	—	15	—		
	睡眠时		—	22	60			
	全模块时钟停止时 (注6)		—	14	28			
	待机时	软件待机时	—	0.10	3	mA		
		深度软件待机时	—	20	60	μA		
模拟电源电流	12 位 A/D 转换中 (未使用采样 & 保持电路时, 每个单元)	AI <sub>CC0</sub>	—	3	5	mA		
	12 位 A/D 转换中 (未使用采样保持电路时, 每个单元)		—	3	5	mA		
	可编程增益放大器 (每个通道)		—	1	2	mA		
	窗口比较器 (1ch 运行时)		—	0.5	1	mA		
	窗口比较器 (6ch 运行时)		—	1	2	mA		
	12 位 A/D 转换待机时 (全部单元)		—	60	90	μA		
	10 位 A/D 转换中 (每个单元)	AI <sub>CC</sub>	—	0.9	2	mA		
10 位 A/D 转换待机时 (全部单元)	—		0.3	3	μA			
基准电源电流	12 位 A/D 转换中 (每个单元)	AI <sub>REFH0</sub>	—	1.6	3	mA		
	12 位 A/D 转换待机时 (全部单元)		—	1.6	3	mA		
	10 位 A/D 转换中 (每个单元)	AI <sub>REF</sub>	—	0.1	1	mA		
	10 位 A/D 转换待机时 (全部单元)		—	0.1	3	μA		
VCC 上升斜率		SV <sub>CC</sub>	—	—	20	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. 外围功能为提供时钟的状态, BGO 运行除外。

注 3. 根据以下表达式, I<sub>CC</sub> 取决于 f (ICLK)。(ICLK:PCLK=8:4)

$$I_{CC \max} = 0.54 \times f + 16 \quad (\text{最大运行时})$$

$$I_{CC \max} = 0.14 \times f + 6 \quad (\text{正常运行时})$$

$$I_{CC \max} = 0.44 \times f + 16 \quad (\text{睡眠时})$$

注 4. 外围功能为停止提供时钟的状态, 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擦除时的增加。

注 6. 这是参考值。

表 33.4 输出容许电流

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC  
条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
 $T_a = -40 \sim +85^\circ\text{C}$ 。 $T_a$  在条件 1 ~ 3 中通用。

项目	符号	min	typ	max	单位
输出 Low 电平的容许电流 (每个引脚的平均值)	$I_{OL}$	—	—	2.0 (注 1)	mA
输出 Low 电平的容许电流 (每个引脚的最大值)	$I_{OL}$	—	—	4.0 (注 1)	mA
输出 Low 电平的容许电流 (总和)	$\Sigma I_{OL}$	—	—	110	mA
输出 High 电平的容许电流 (每个引脚的平均值)	$-I_{OH}$	—	—	2.0 (注 1)	mA
输出 High 电平的容许电流 (每个引脚的最大值)	$-I_{OH}$	—	—	4.0 (注 1)	mA
输出 High 电平的容许电流 (总和)	$\Sigma -I_{OH}$	—	—	35	mA

【使用时的注意事项】为了确保 LSI 的可靠性，输出电流值不能超过表 33.4 中的值。

注 1. P71 ~ 76、P90 ~ 95 为  $I_{OL}=15\text{mA}$  (Max.) /  $-I_{OH}=5\text{mA}$  (Max.)。但是，在 112 引脚 LQFP 版 / 100 引脚 LQFP 版中，同时流过超过 2.0mA 的  $I_{OL}/-I_{OH}$  电流的引脚必须在 6 个以内，80 引脚 LQFP 版 / 64 引脚 LQFP 版中必须在 3 个以内。

### 33.3 AC 特性

表 33.5 工作频率值

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC  
条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
 $T_a = -40 \sim +85^\circ\text{C}$ 。 $T_a$  在条件 1 ~ 3 中通用。

项目		符号	min	typ	max	单位
工作频率	系统时钟 (ICLK)	f	8	—	100	MHz
	外围模块时钟 (PCLK)		8	—	50	

## 33.3.1 时钟的时序

表 33.6 时钟的时序

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

$T_a = -40 \sim +85^\circ\text{C}$ 。 $T_a$  在条件 1 ~ 3 中通用。

项目	符号	min	max	单位	测量条件
复位振荡稳定时间 (晶体)	$t_{\text{OSC1}}$	10	—	ms	图 33.1
软件待机振荡稳定时间 (晶体)	$t_{\text{OSC2}}$	10	—	ms	图 33.2
深度软件待机振荡稳定时间 (晶体)	$t_{\text{OSC3}}$	10	—	ms	图 33.3
EXTAL 外部时钟输出的延迟稳定时间	$t_{\text{DEXT}}$	1	—	ms	图 33.1
EXTAL 外部时钟输入的 Low 电平脉宽	$t_{\text{EXL}}$	35	—	ns	图 33.4
EXTAL 外部时钟输入的 High 电平脉宽	$t_{\text{EXH}}$	35	—	ns	
EXTAL 外部时钟上升时间	$t_{\text{EXr}}$	—	5	ns	
EXTAL 外部时钟下降时间	$t_{\text{EXf}}$	—	5	ns	
内部振荡器 (IWDTCLK) 振荡频率	$f_{\text{IWDTCLK}}$	62.5	187.5	kHz	

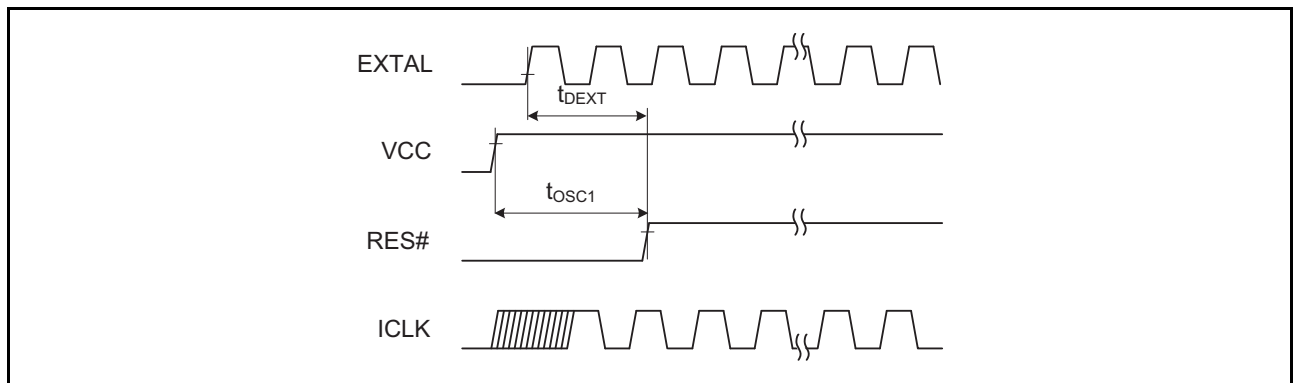


图 33.1 振荡稳定时间的时序

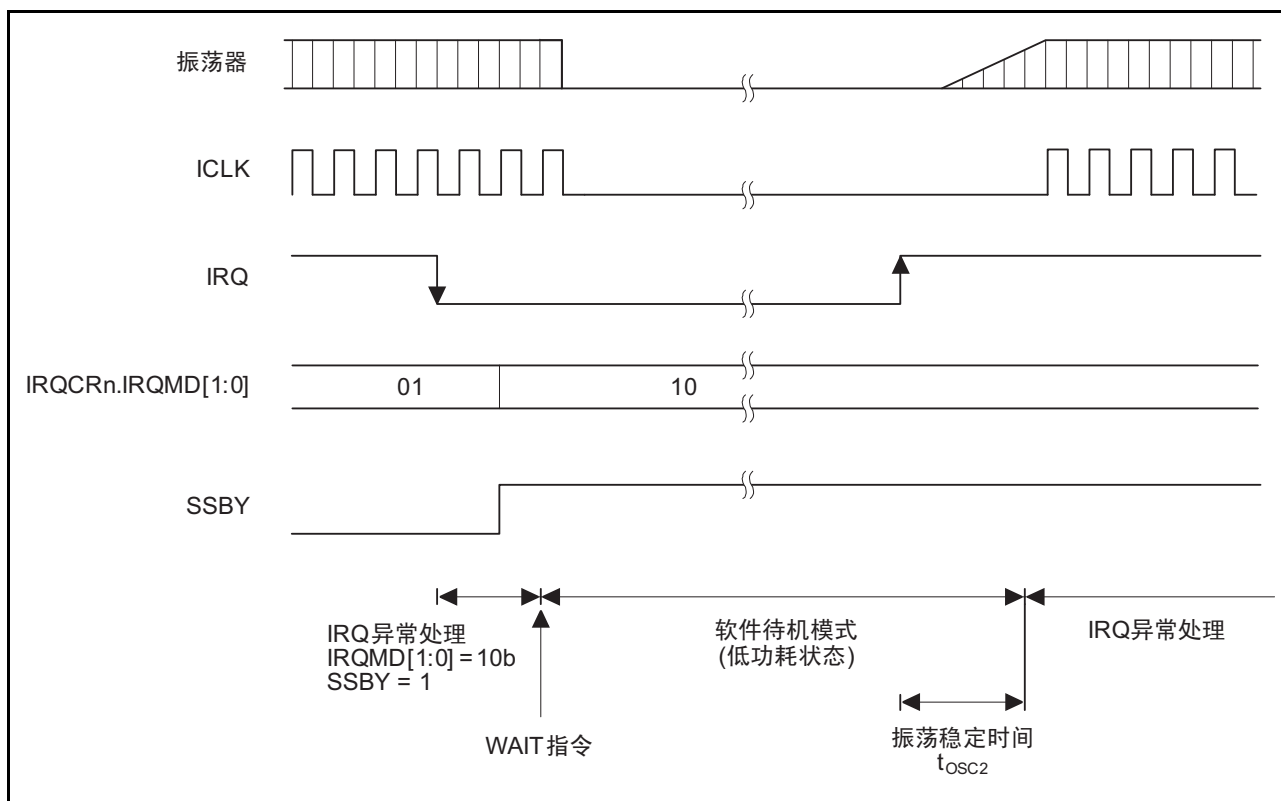


图 33.2 软件待机振荡稳定时间的时序



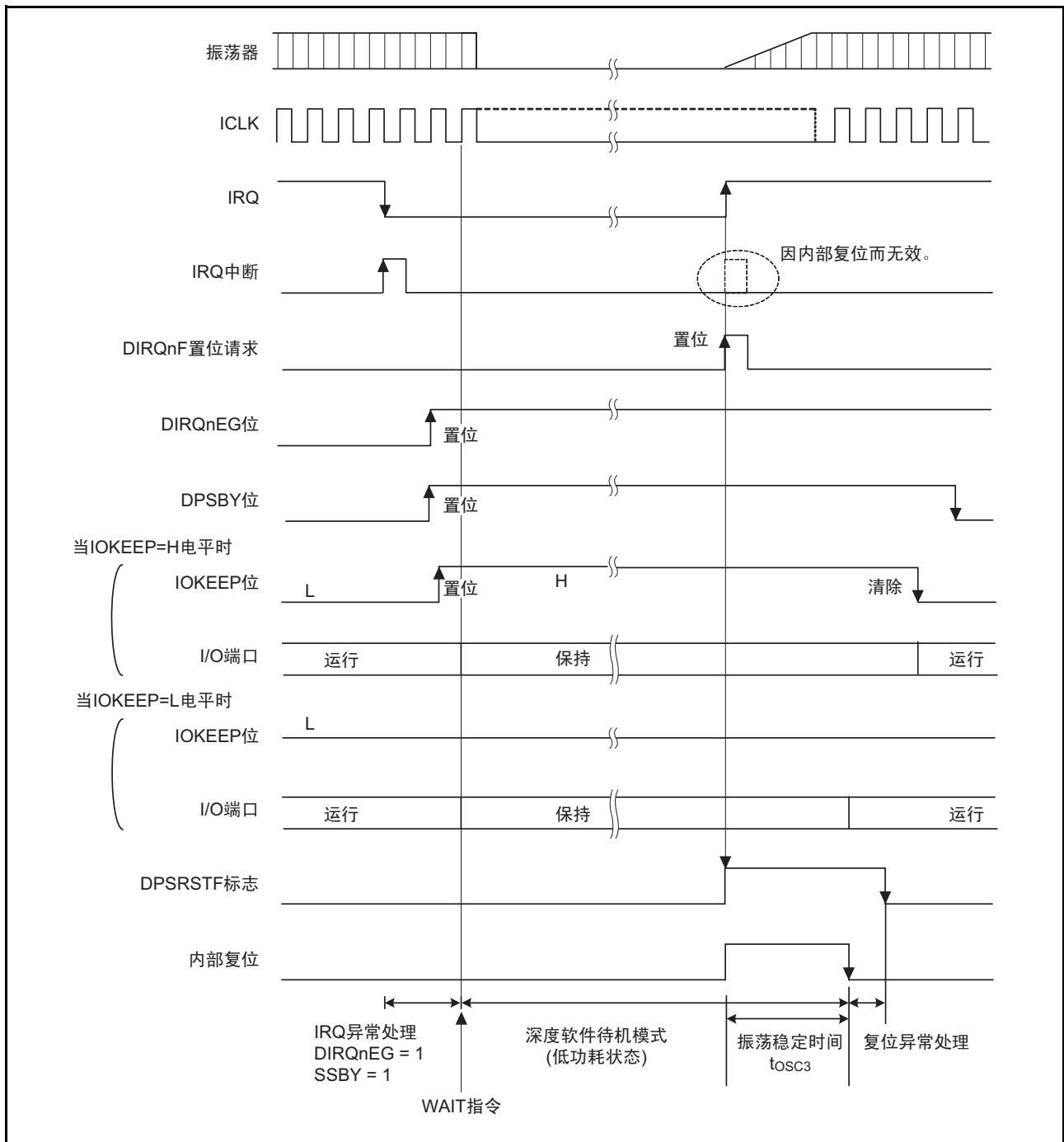


图 33.3 深度软件待机振荡稳定时间的时序

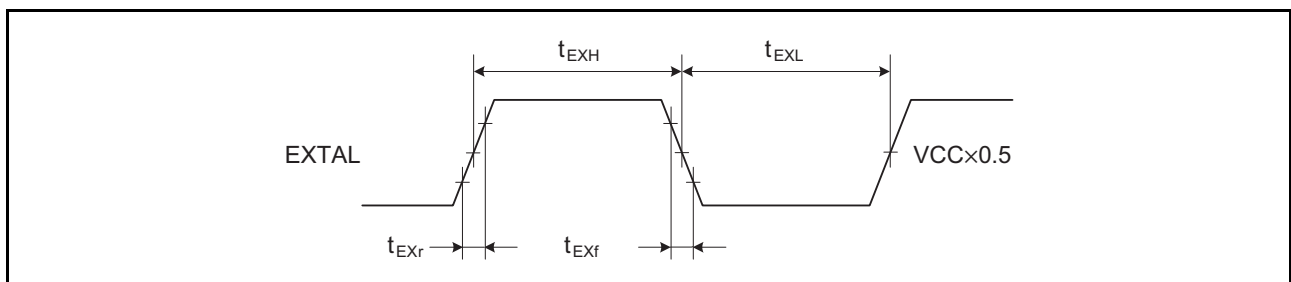


图 33.4 EXTAL 外部输入时钟的时序

### 33.3.2 控制信号的时序

表 33.7 控制信号的时序

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

- 条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC
  - 条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC
  - 条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC
- $T_a = -40 \sim +85^\circ\text{C}$ 。 $T_a$  在条件 1 ~ 3 中通用。

项目	符号	min	max	单位	测量条件
RES# 脉宽 (ROM 的编程 / 擦除、数据闪存的编程 / 擦除和数据闪存的空白检查除外 (注 1))	$t_{RESW}$ (注 2)	20	—	$t_{cyc}$ (注 4)	图 33.5
		1.5	—	$\mu\text{s}$	
内部复位时间 (注 3)	$t_{RESW2}$	35	—	$\mu\text{s}$	
NMI 脉宽	$t_{NMIW}$	200	—	ns	图 33.6
IRQ 脉宽	$t_{IRQW}$	200	—	ns	图 33.7

- 注 1. 在 ROM 的编程 / 擦除、数据闪存的编程 / 擦除和数据闪存的空白检查过程中，如果通过 RES# 引脚进行复位，请参照“31. ROM (保存代码的闪存)”的“31.12 使用时的注意事项”。
- 注 2. 需要同时满足时间和周期数的规定。
- 注 3. 本项目是对 FCU 复位的规定。
- 注 4. ICLK 的周期。

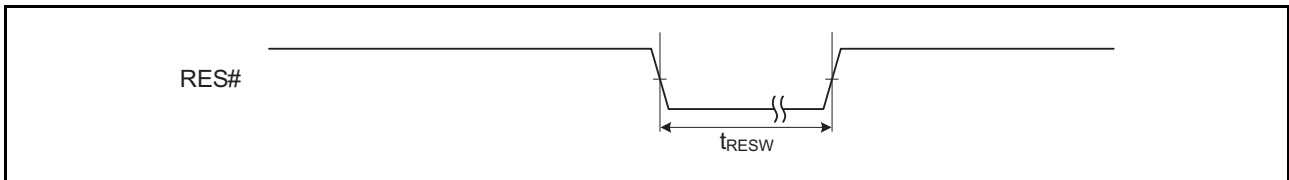


图 33.5 复位输入的时序

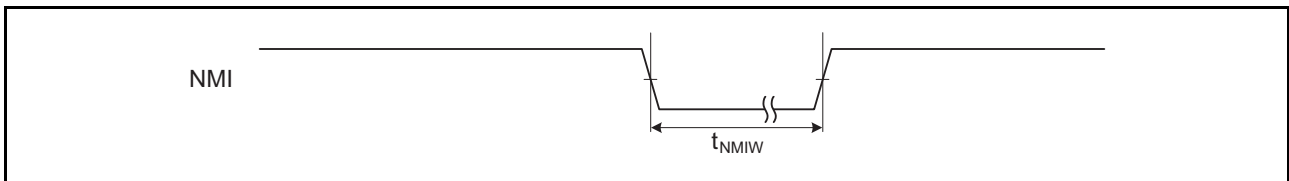


图 33.6 NMI 中断输入的时序

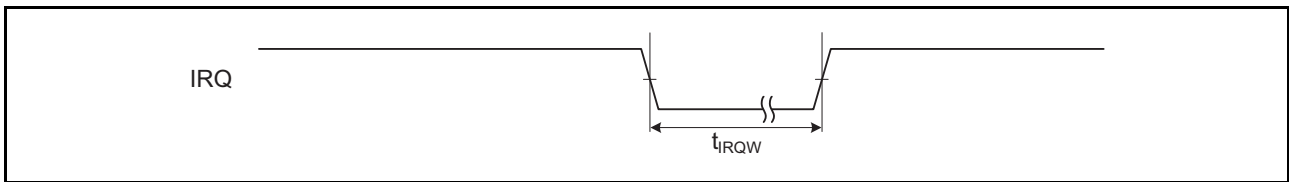


图 33.7 IRQ 中断输入的时序

## 33.3.3 内部外围模块的时序

表 33.8 内部外围模块的时序 (1)

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V

AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V

AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V

AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

$T_a = -40 \sim +85^\circ\text{C}$ 。  $T_a$  在条件 1 ~ 3 中通用。

项目		符号	min	max	单位	测量条件	
SCI	输入时钟周期	异步	$4 \times t_{Pcyc}$	—	ns	图 33.8	
		时钟同步	$6 \times t_{Pcyc}$	—			
	输入时钟脉宽		$t_{SCKW}$	$0.4 \times t_{Scyc}$	$0.6 \times t_{Scyc}$		ns
	输入时钟的上升时间		$t_{SCKr}$	—	20		ns
	输入时钟的下降时间		$t_{SCKf}$	—	20		ns
	输出时钟周期	异步	$t_{Scyc}$	$16 \times t_{Pcyc}$	—		ns
		时钟同步		$6 \times t_{Pcyc}$	—		ns
	输出时钟脉宽		$t_{SCKW}$	$0.4 \times t_{Scyc}$	$0.6 \times t_{Scyc}$		ns
	输出时钟的上升时间		$t_{SCKr}$	—	20		ns
	输出时钟的下降时间		$t_{SCKf}$	—	20		ns
发送数据的延迟时间 (时钟同步)		$t_{TXD}$	—	40	ns	图 33.9	
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		

$t_{Pcyc}$ : PCLK 的周期

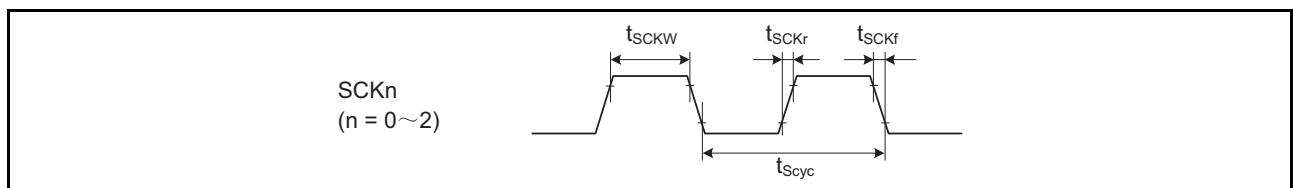


图 33.8 SCK 时钟输入的时序

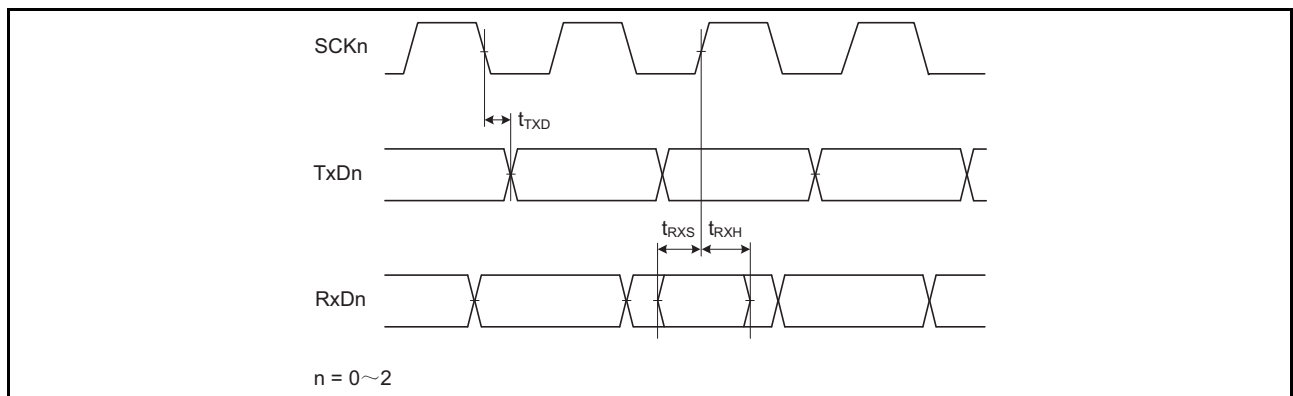


图 33.9 SCI 输入 / 输出的时序和时钟同步模式

表 33.9 内部外围模块的时序 (2)

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
 $T_a = -40 \sim +85^\circ\text{C}$ 。 $T_a$  在条件 1 ~ 3 中通用。

项目		符号	min (注 1、注 2)	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 33.10
	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 输入尖峰脉冲的消除时间	$t_{SP}$	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	图 33.10
	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 输入尖峰脉冲的消除时间	$t_{SP}$	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 $\phi$ ) 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。

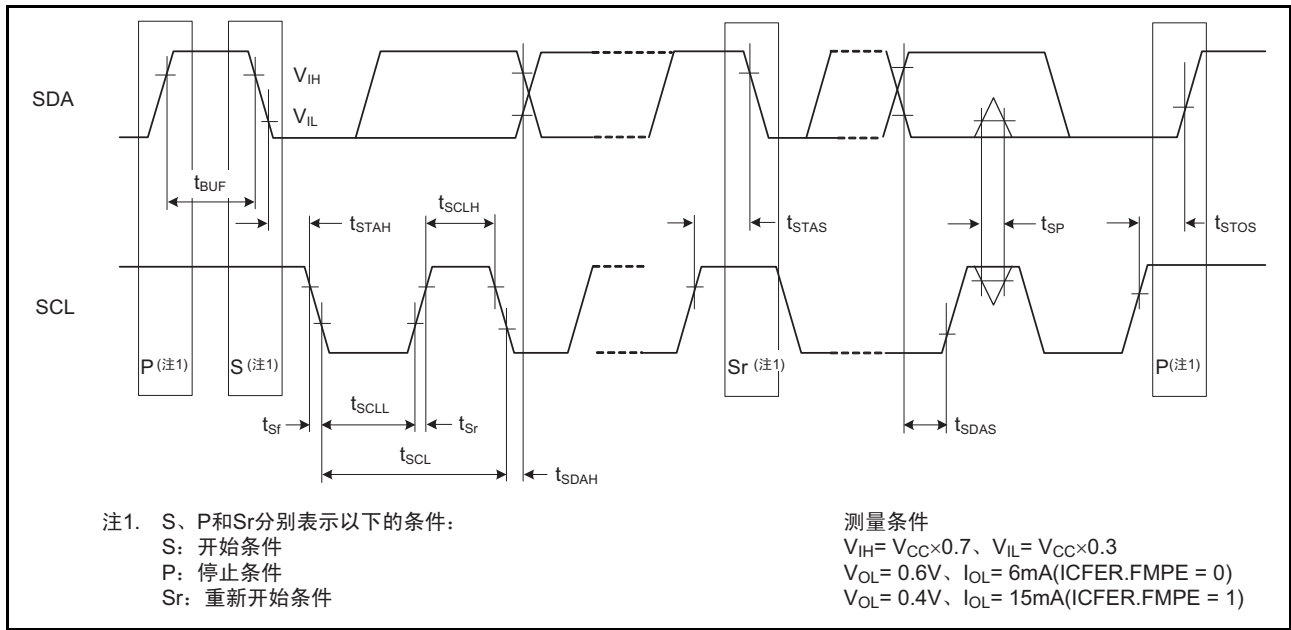


图 33.10 I<sup>2</sup>C 总线接口输入 / 输出的时序

表 33.10 内部外围模块的时序 (3)

注：表中没有记载条件的项目的规格值在条件 1～3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
 $T_a = -40 \sim +85^\circ\text{C}$ 。 $T_a$  在条件 1～3 中通用。

项目		符号	min	max	单位	测量条件		
RSPI	RSPCK 时钟周期	主控	$t_{SPCyc}$	4	4096	$t_{Pcyc}$	图 33.11	
		从属		8	4096			
	RSPCK 时钟的 High 电平脉宽	主控	$t_{SPCKWH}$	$(t_{SPCyc} - t_{SPCKR} - t_{SPCKF})/2 - 3$	—	—	ns	图 33.12 ~ 图 33.15
		从属			$(t_{SPCyc} - t_{SPCKR} - t_{SPCKF})/2$	—	—	
	RSPCK 时钟的 Low 电平脉宽	主控	$t_{SPCKWL}$	$(t_{SPCyc} - t_{SPCKR} - t_{SPCKF})/2 - 3$	—	—	ns	
		从属			$(t_{SPCyc} - t_{SPCKR} - t_{SPCKF})/2$	—	—	
	RSPCK 时钟的上升 / 下降时间	输出	$t_{SPCKR}$	—	5	ns		
		输入	$t_{SPCKF}$	—	1	$\mu\text{s}$		
	数据输入的准备时间	主控	$t_{SU}$	25	—	ns		
		从属		0	—	—		
	数据输入的保持时间	主控	$t_{H}$	0	—	ns		
		从属		$20 + 2 \times t_{Pcyc}$	—	—		
	SSL 准备时间	主控	$t_{LEAD}$	1	8	$t_{SPCyc}$		
		从属		4	—	$t_{Pcyc}$		
	SSL 保持时间	主控	$t_{LAG}$	1	8	$t_{SPCyc}$		
		从属		4	—	$t_{Pcyc}$		
	数据输出的延迟时间	主控	$t_{OD}$	—	20	ns		
		从属		—	$3 \times t_{Pcyc} + 40$	—		
	数据输出的保持时间	主控	$t_{OH}$	0	—	ns		
		从属		0	—	—		
连续发送的延迟时间	主控	$t_{TD}$	$t_{SPCyc} + 2 \times t_{Pcyc}$	$8 \times t_{SPCyc} + 2 \times t_{Pcyc}$	ns			
	从属		$4 \times t_{Pcyc}$	—	—			
MOSI 和 MISO 的上升 / 下降时间	输出	$t_{DR}$	—	15	ns	图 33.12 ~ 图 33.15		
	输入	$t_{DF}$	—	1	$\mu\text{s}$			
SSL 上升 / 下降时间	输出	$t_{SSLR}$	—	15	ns			
	输入	$t_{SSLF}$	—	1	$\mu\text{s}$			
从属存取时间		$t_{SA}$	—	4	$t_{Pcyc}$	图 33.14		
从属输出释放时间		$t_{REL}$	—	3	$t_{Pcyc}$	图 33.15		

$t_{Pcyc}$ : PCLK 的周期

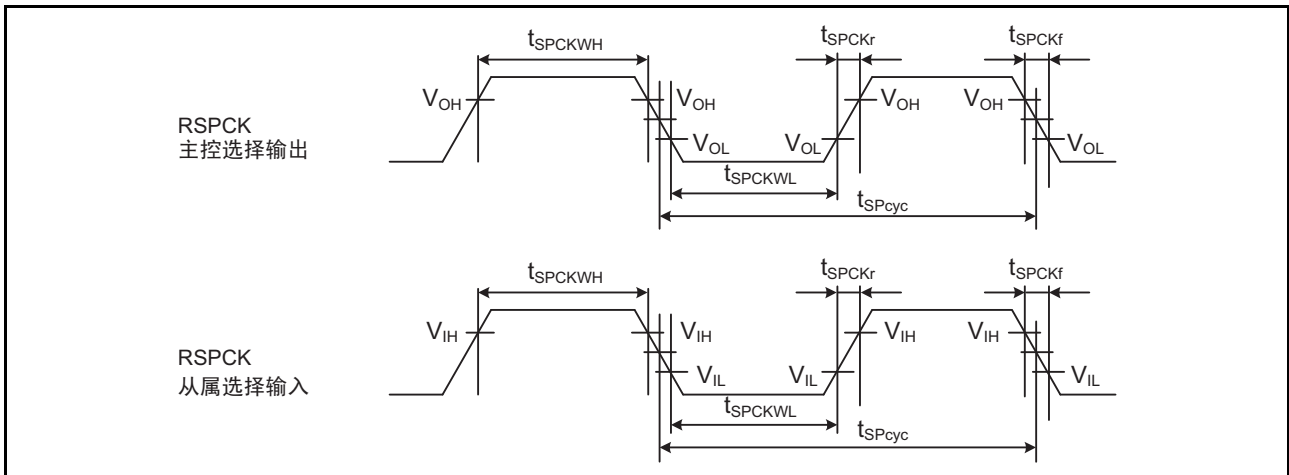


图 33.11 RSPCK 时钟的时序

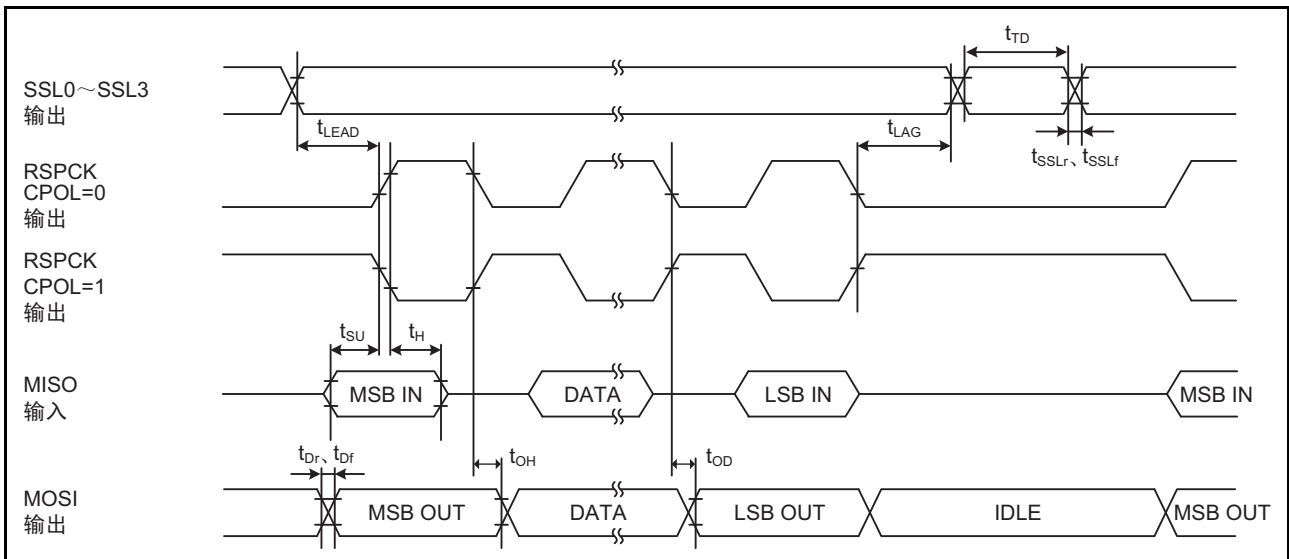


图 33.12 RSPCK 时序 (主控、CPHA=0)

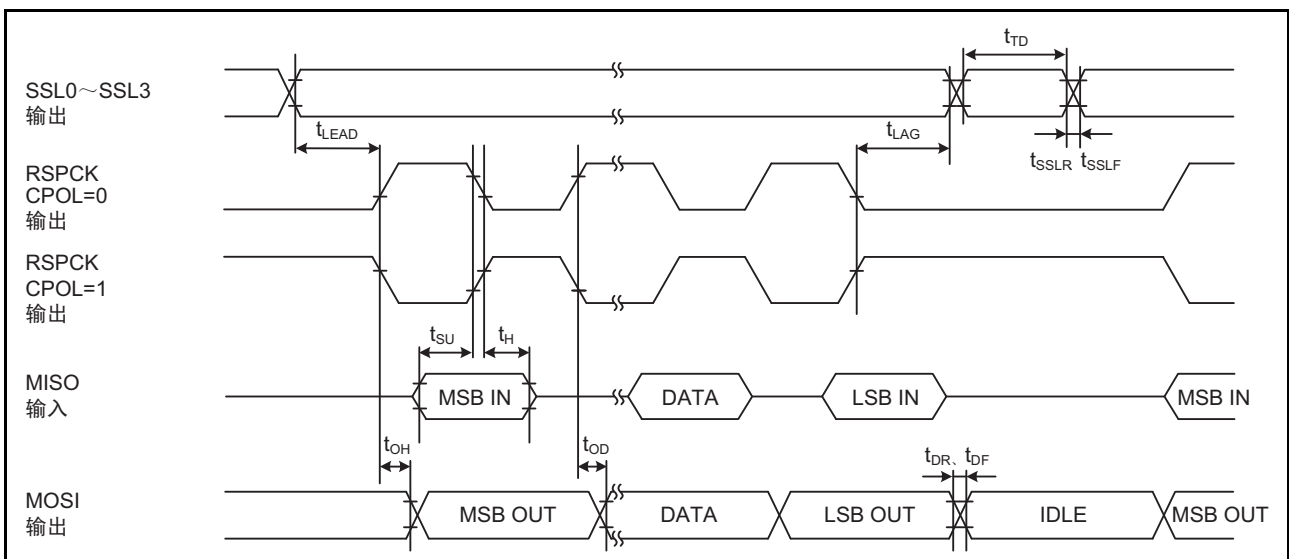


图 33.13 RSPCK 时序 (主控、CPHA=1)

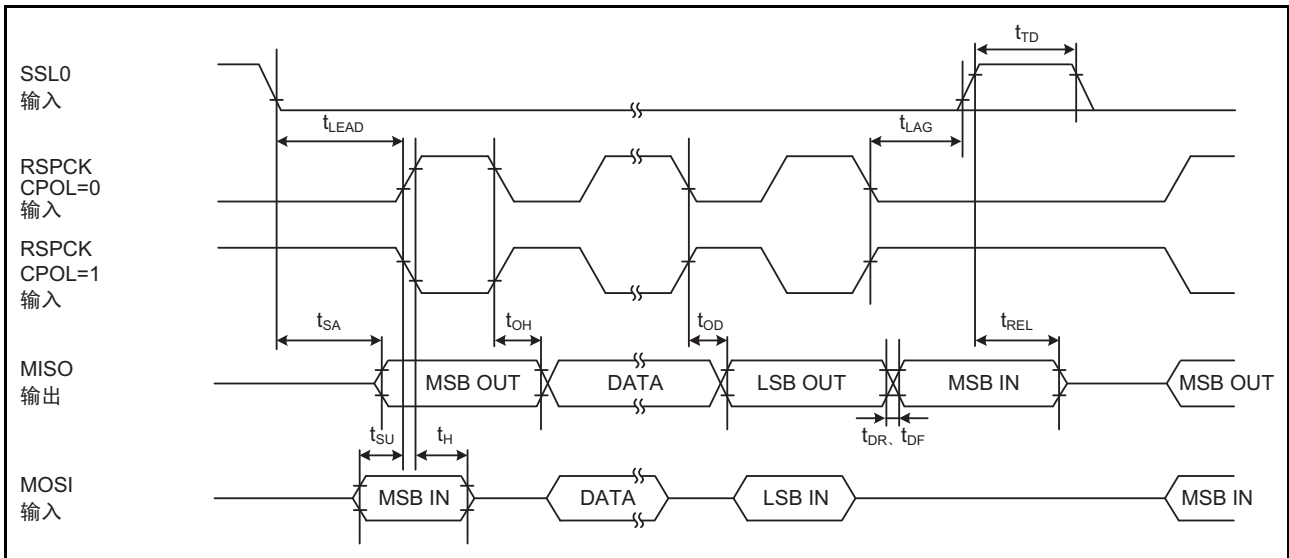


图 33.14 RSPI 时序 (从属、CPHA=0)

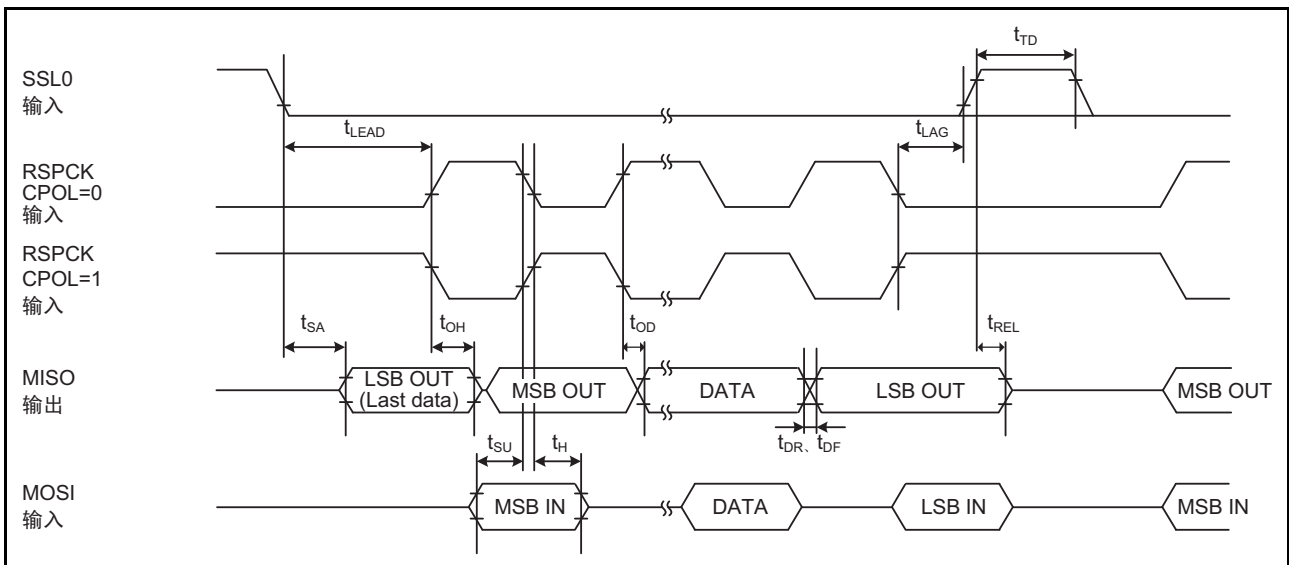


图 33.15 RSPI 时序 (从属、CPHA=1)



表 33.11 内部外围模块的时序 (4)

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
T<sub>a</sub>=-40 ~ +85°C。T<sub>a</sub> 在条件 1 ~ 3 中通用。

项目		符号	min	max	单位	测量条件
MTU3	输入捕捉的输入脉宽 (指定单边沿)	t <sub>TICW</sub>	3.0	—	t <sub>ICyc</sub>	图 33.16
	输入捕捉的输入脉宽 (指定双边沿)	t <sub>TICW</sub>	5.0	—	t <sub>ICyc</sub>	
	定时器时钟脉宽 (指定单边沿)	t <sub>TCKWH/L</sub>	3.0	—	t <sub>ICyc</sub>	图 33.17
	定时器时钟脉宽 (指定双边沿)	t <sub>TCKWH/L</sub>	5.0	—	t <sub>ICyc</sub>	
	定时器时钟脉宽 (相位系数模式)	t <sub>TCKWH/L</sub>	5.0	—	t <sub>ICyc</sub>	
GPT	输入捕捉的输入脉宽 (指定单边沿)	t <sub>GTICW</sub>	3.0	—	t <sub>ICyc</sub>	图 33.18
	输入捕捉的输入脉宽 (指定双边沿)	t <sub>GTICW</sub>	5.0	—	t <sub>ICyc</sub>	

t<sub>ICyc</sub>: ICLK 的周期

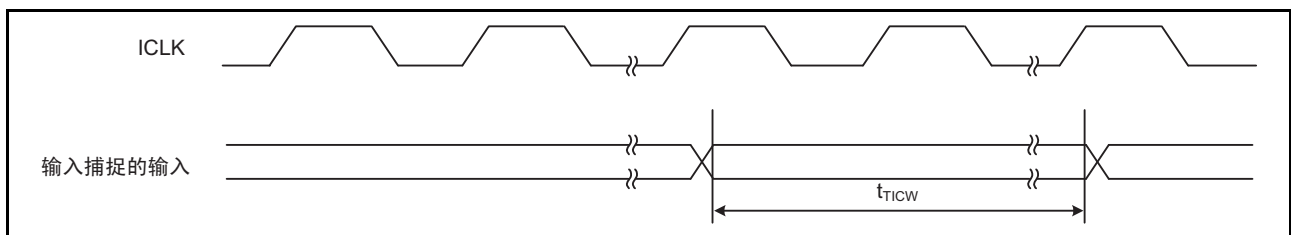


图 33.16 MTU3 输入 / 输出的时序

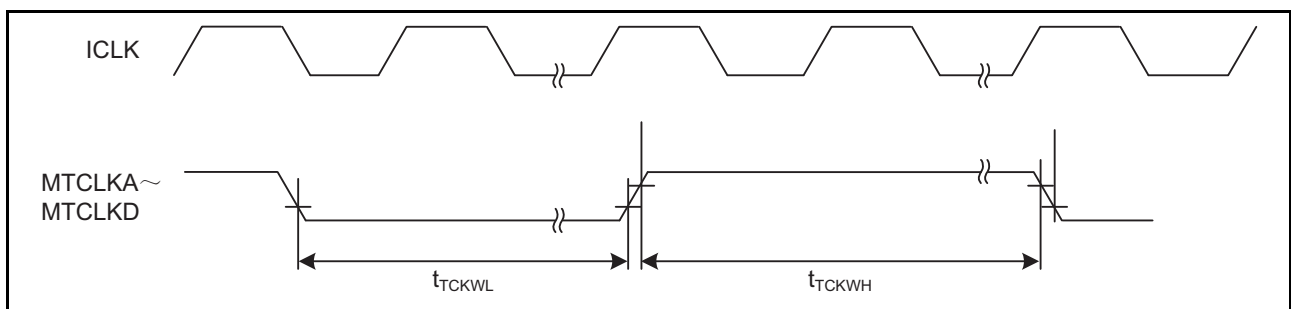


图 33.17 MTU3 时钟输入的时序

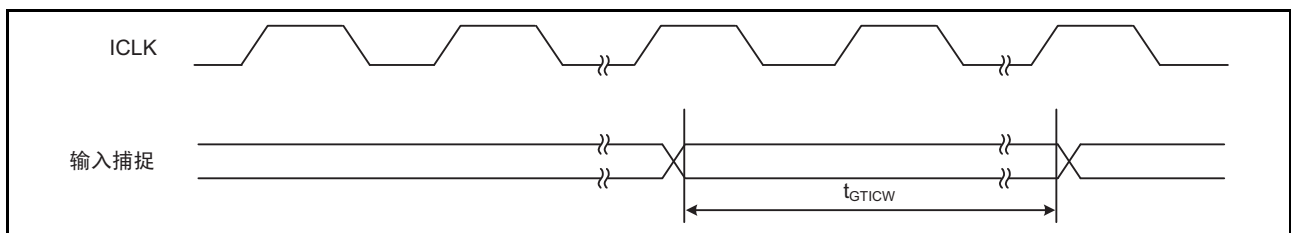


图 33.18 GPT 输入 / 输出的时序

表 33.12 内部外围模块的时序 (5)

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
T<sub>a</sub>=-40 ~ +85°C。T<sub>a</sub> 在条件 1 ~ 3 中通用。

项目		符号	min	max	单位	测量条件
POE3	POE# 输入脉宽	t <sub>POEW</sub>	1.5	—	t <sub>Pcyc</sub>	图 33.19

t<sub>Pcyc</sub>: PCLK 的周期

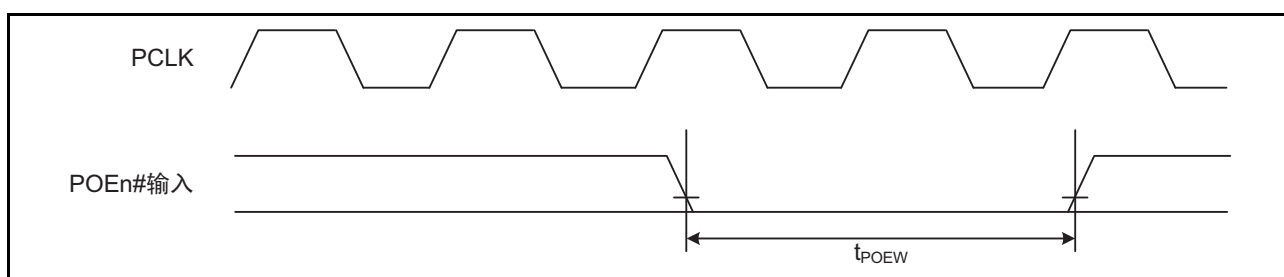


图 33.19 POE3# 输入的时序

## 33.4 A/D 转换特性

表 33.13 10 位 A/D 转换特性

注：表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLCC=2.7 ~ 3.6V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0~3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC  
T<sub>a</sub>=-40 ~ +85°C

项目	min	typ	max	单位	测量条件
分辨率	10	10	10	位	
转换时间 (注 1) (AD 时钟 =25MHz 时)	2.0	—	—	μs	采样的 25 个状态
模拟输入的电容	—	—	4	pF	
积分非线性误差	—	—	±3.0	LSB	
偏移误差	—	—	±3.0	LSB	
满刻度误差	—	—	±3.0	LSB	
量化误差	—	±0.5	—	LSB	
绝对精度	—	—	±4.0	LSB	
容许信号源阻抗	—	—	1.0	kΩ	

条件 2: VCC=PLLCC=2.7 ~ 3.6V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLCC=4.0 ~ 5.5V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
T<sub>a</sub>=-40 ~ +85°C。T<sub>a</sub> 在条件 2 和条件 3 中通用。

项目	min	typ	max	单位	测量条件
分辨率	10	10	10	位	
转换时间 (注 1) (AD 时钟 =50MHz 时)	1.0	—	—	μs	采样的 25 个状态
模拟输入的电容	—	—	4	pF	
积分非线性误差	—	—	±3.0	LSB	
偏移误差	—	—	±3.0	LSB	
满刻度误差	—	—	±3.0	LSB	
量化误差	—	±0.5	—	LSB	
绝对精度	—	—	±4.0	LSB	
容许信号源阻抗	—	—	1.0	kΩ	

注 1. 转换时间是采样时间和比较时间的总和，各项目的测量条件表示采样的状态数。

表 33.14 12 位 A/D 转换特性

注：表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1:  $V_{CC}=PLL_{VCC}=2.7 \sim 3.6V$ 、 $V_{SS}=PLL_{VSS}=AV_{SS0}=AV_{SS}=V_{REFL0}=0V$   
 $AV_{CC0}=AV_{CC}=3.0 \sim 3.6V$ 、 $V_{REFH0}=3.0V \sim AV_{CC0}$ 、 $V_{REF}=3.0V \sim AV_{CC}$   
 $T_a=-40 \sim +85^\circ C$ 、 $ICLK=8 \sim 100MHz$ 、 $PCLK=8 \sim 50MHz$

项目		min	typ	max	单位	测量条件
分辨率		12	12	12	位	
转换时间 (注 1) (AD 时钟 =25MHz 时)		2.0	—	—	$\mu s$	采样的 20 个状态
模拟输入的电容		—	—	6	pF	
积分非线性误差		—	—	$\pm 4.0$	LSB	
偏移误差		—	—	$\pm 7.5$	LSB	
满刻度误差		—	—	$\pm 7.5$	LSB	
量化误差		—	$\pm 0.5$	—	LSB	
绝对精度	使用采样 & 保持电路	—	—	$\pm 8.0$	LSB	$AV_{in}=0.25 \sim AV_{REFH}-0.25$
	未使用采样 & 保持电路	—	—	$\pm 8.0$	LSB	$AV_{in}=AV_{REFL} \sim AV_{REFH}$
容许信号源阻抗		—	—	3.0	k $\Omega$	

条件 2:  $V_{CC}=PLL_{VCC}=2.7 \sim 3.6V$ 、 $V_{SS}=PLL_{VSS}=AV_{SS0}=AV_{SS}=V_{REFL0}=0V$   
 $AV_{CC0}=AV_{CC}=4.0 \sim 5.5V$ 、 $V_{REFH0}=4.0V \sim AV_{CC0}$ 、 $V_{REF}=4.0V \sim AV_{CC}$

条件 3:  $V_{CC}=PLL_{VCC}=4.0 \sim 5.5V$ 、 $V_{SS}=PLL_{VSS}=AV_{SS0}=AV_{SS}=V_{REFL0}=0V$   
 $AV_{CC0}=AV_{CC}=4.0 \sim 5.5V$ 、 $V_{REFH0}=4.0V \sim AV_{CC0}$ 、 $V_{REF}=4.0V \sim AV_{CC}$   
 $T_a=-40 \sim +85^\circ C$ 。 $T_a$  在条件 2 和条件 3 中通用。 $ICLK=8 \sim 100MHz$ 、 $PCLK=8 \sim 50MHz$

项目		min	typ	max	单位	测量条件
分辨率		12	12	12	位	
转换时间 (注 1) (AD 时钟 =50MHz 时)		1.0	—	—	$\mu s$	采样的 20 个状态
模拟输入的电容		—	—	6	pF	
积分非线性误差		—	—	$\pm 4.0$	LSB	
偏移误差		—	—	$\pm 7.5$	LSB	
满刻度误差		—	—	$\pm 7.5$	LSB	
量化误差		—	$\pm 0.5$	—	LSB	
绝对精度	使用采样 & 保持电路	—	—	$\pm 8.0$	LSB	$AV_{in}=0.25 \sim AV_{REFH}-0.25$
	未使用采样 & 保持电路	—	—	$\pm 8.0$	LSB	$AV_{in}=AV_{REFL} \sim AV_{REFH}$
容许信号源阻抗		—	—	3.0	k $\Omega$	

注 1. 转换时间是采样时间和比较时间的总和，各项目的测量条件表示采样的状态数。

表 33.15 可编程增益放大器特性

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLCC=2.7 ~ 3.6V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLCC=2.7 ~ 3.6V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLCC=4.0 ~ 5.5V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
T<sub>a</sub>=-40 ~ +85°C。T<sub>a</sub> 在条件 1 ~ 3 中通用。

项目	符号	min	typ	max	单位	测量条件
模拟输入的电容	Cin	—	—	6	pF	
输入偏移电压	Voff	—	—	8	mV	
输入电压范围 (Vin)	增益 ×2.000	0.050×AVcc	—	0.38×AVcc	V	
	增益 ×2.500	0.047×AVcc	—	0.30×AVcc		
	增益 ×3.077	0.045×AVcc	—	0.24×AVcc		
	增益 ×3.636	0.042×AVcc	—	0.21×AVcc		
	增益 ×4.000	0.040×AVcc	—	0.19×AVcc		
	增益 ×4.444	0.036×AVcc	—	0.17×AVcc		
	增益 ×5.000	0.033×AVcc	—	0.15×AVcc		
	增益 ×5.714	0.031×AVcc	—	0.13×AVcc		
	增益 ×6.667	0.029×AVcc	—	0.11×AVcc		
	增益 ×10.000	0.025×AVcc	—	0.08×AVcc		
	增益 ×13.333	0.023×AVcc	—	0.06×AVcc		
转换速率	SR	10	—	—	V/μs	
增益误差	增益 ×2.000	—	—	1	%	
	增益 ×2.500	—	—	1		
	增益 ×3.077	—	—	1		
	增益 ×3.636	—	—	1.5		
	增益 ×4.000	—	—	1.5		
	增益 ×4.444	—	—	2		
	增益 ×5.000	—	—	2		
	增益 ×5.714	—	—	2		
	增益 ×6.667	—	—	3		
	增益 ×10.000	—	—	4		
	增益 ×13.333	—	—	4		

表 33.16 比较器特性

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLCC=2.7 ~ 3.6V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLCC=2.7 ~ 3.6V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLCC=4.0 ~ 5.5V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
T<sub>a</sub>=-40 ~ +85°C。T<sub>a</sub> 在条件 1 ~ 3 中通用。

项目	符号	min	typ	max	单位	测量条件
模拟输入的电容	Cin	—	—	6	pF	
REFH 引脚的偏移电压	Voff	—	—	5	mV	
REFL 引脚的偏移电压		—	—	5	mV	
REFH 输入电压范围	Vin	1.7	—	AVcc-0.3	V	
REFL 输入电压范围		0.3	—	AVcc-1.7	V	
REFH 应答时间	tCR	—	—	1	μs	
REFL 应答时间	tCF	—	—	1	μs	

## 33.5 上电复位电路和电压检测电路的特性

表 33.17 上电复位电路和电压检测电路的特性

注：表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLCC=2.7 ~ 3.6V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLCC=2.7 ~ 3.6V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
 $T_a=-40 \sim +85^{\circ}\text{C}$ 。 $T_a$  在条件 1、2 中通用。

项目		符号	min	typ	max	单位	测量条件
电压检测电平	上电复位 (POR)	$V_{\text{POR}}$	2.48	2.60	2.72	V	图 33.20
	电压检测电路 (LVD)	$V_{\text{det1}}$	2.68	2.80	2.92		图 33.21
		$V_{\text{det2}}$	2.98	3.10	3.22		图 33.22
内部复位时间		$t_{\text{POR}}$	20	35	50	ms	图 33.21、图 33.22
最小 VCC 降低时间 (注 1)		$t_{\text{VOFF}}$	200	—	—	us	图 33.20 ~ 图 33.22
应答延迟时间		$t_{\text{det}}$	—	—	200	us	

条件 3: VCC=PLLCC=4.0 ~ 5.5V、VSS=PLLSS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
 $T_a=-40 \sim +85^{\circ}\text{C}$ 。

项目		符号	min	typ	max	单位	测量条件
电压检测电平	上电复位 (POR)	$V_{\text{POR}}$	3.70	3.90	4.10	V	图 33.20
	电压检测电路 (LVD)	$V_{\text{det1}}$	3.95	4.15	4.35		图 33.21
		$V_{\text{det2}}$	4.40	4.60	4.80		图 33.22
内部复位时间		$t_{\text{POR}}$	20	35	50	ms	图 33.21、图 33.22
最小 VCC 降低时间 (注 1)		$t_{\text{VOFF}}$	200	—	—	us	图 33.20 ~ 图 33.22
应答延迟时间		$t_{\text{det}}$	—	—	200	us	

注 1. 这是 VCC 低于 POR/LVD 的电压检测电平  $V_{\text{POR}}$ 、 $V_{\text{det1}}$ 、 $V_{\text{det2}}$  的 min 值的时间。

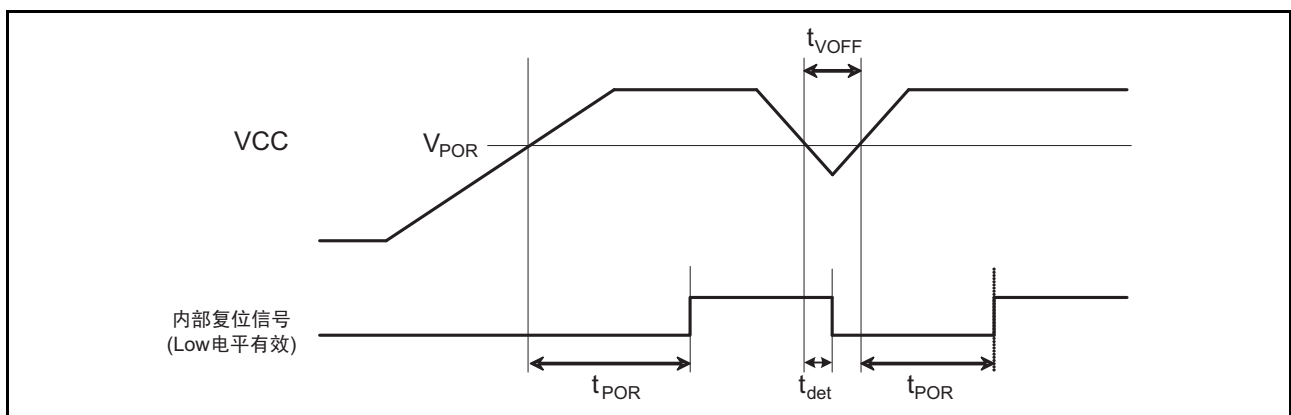


图 33.20 上电复位的时序

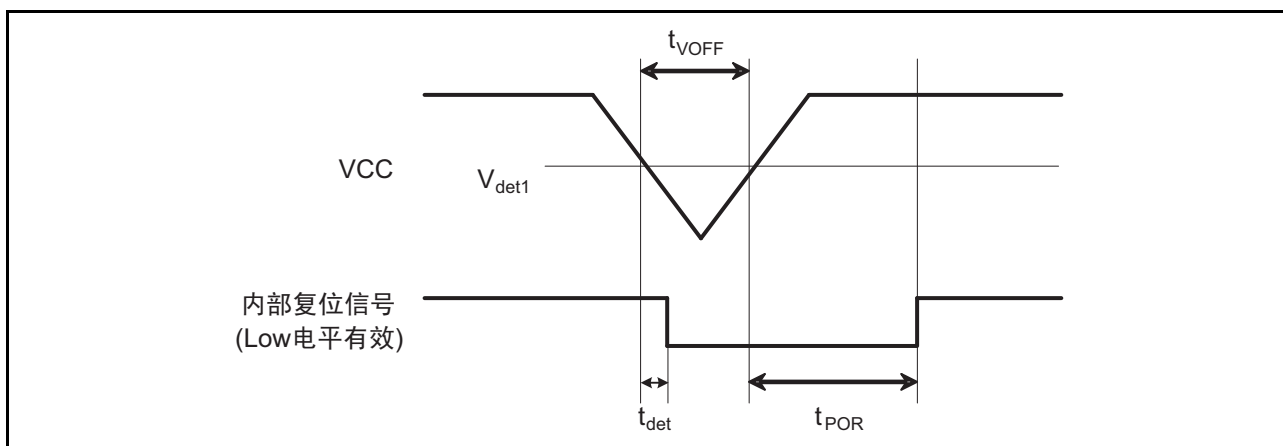


图 33.21 电压检测电路的时序 ( $V_{det1}$ )

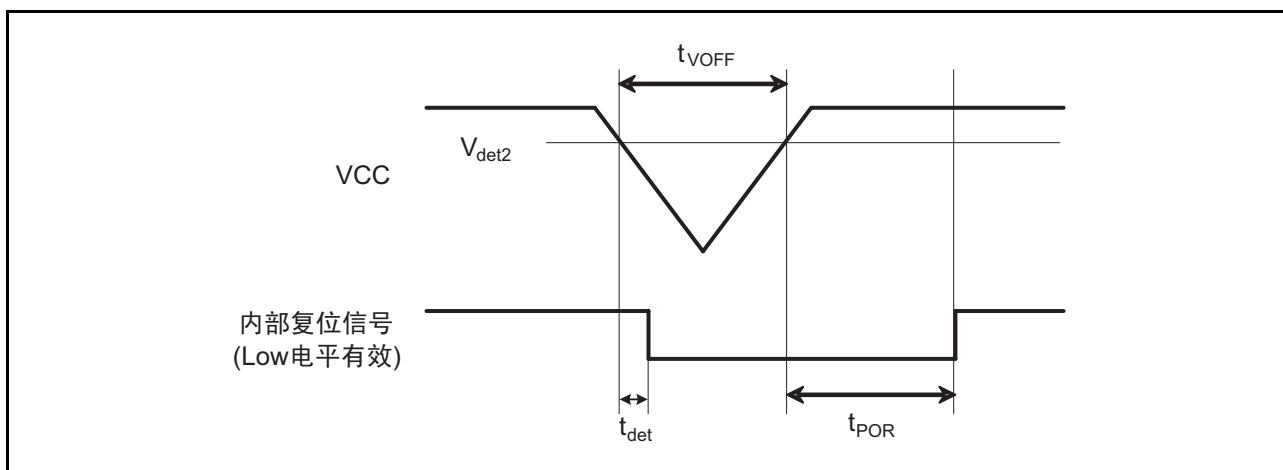


图 33.22 电压检测电路的时序 ( $V_{det2}$ )



## 33.6 振荡停止检测时序

表 33.18 振荡停止检测时序

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC  
 $T_a = -40 \sim +85^\circ\text{C}$ 。 $T_a$  在条件 1 ~ 3 中通用。

项目	符号	min	typ	max	单位	测量条件
检测时间	$t_{dr}$	—	—	1.0	ms	图 33.23
检测振荡停止时的内部振荡频率	$f_{\text{MAIN}}$	0.5	—	7.0	MHz	

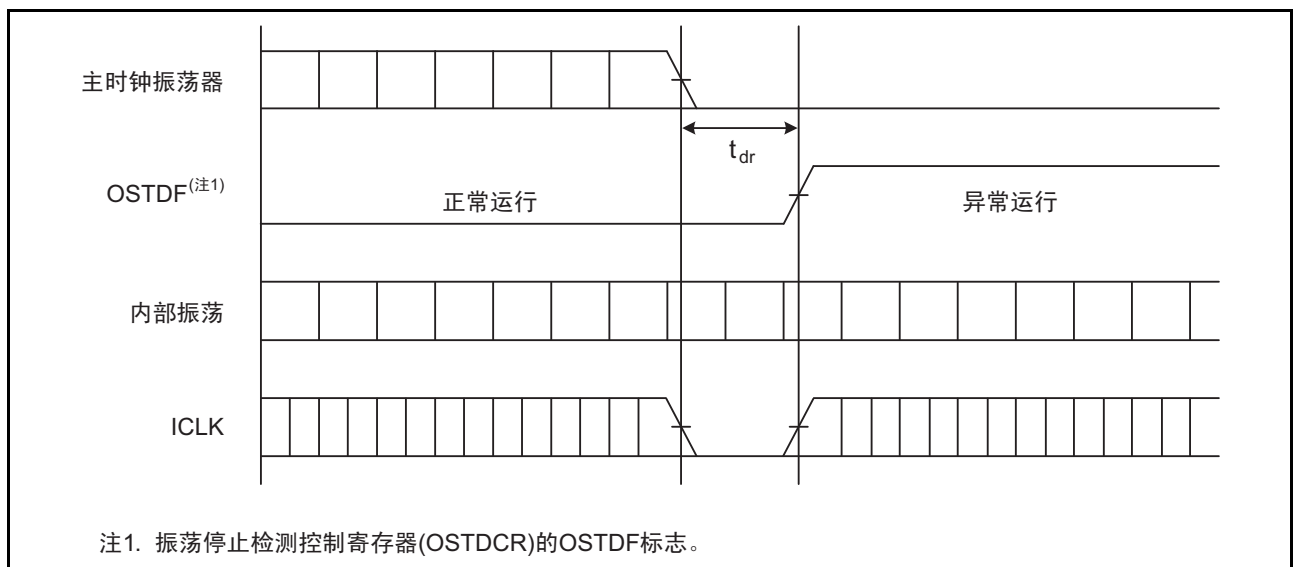


图 33.23 振荡停止检测的时序

## 33.7 ROM（保存代码的闪存）特性

表 33.19 ROM（保存代码的闪存）特性

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

编程 / 擦除时的工作温度范围:  $T_a = -40 \sim +85^\circ\text{C}$ 。  $T_a$  在条件 1 ~ 3 中通用。

项目	符号	min	typ	max	单位	测量条件		
编程时间	256 字节	$t_{P256}$	—	2	12	ms	PCLK=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	256 字节	$t_{E256}$	—	2.4	14.4	ms	PCLK=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 / 擦除周期 (注 1)	4K 字节	$t_{E4K}$	—	25	60	ms	PCLK=50MHz	
	16K 字节	$t_{E16K}$	—	100	240	ms	$N_{PEC} \leq 100$ 时	
	4K 字节	$t_{E4K}$	—	30	72	ms	PCLK=50MHz	
	16K 字节	$t_{E16K}$	—	120	288	ms	$N_{PEC} > 100$ 时	
再编程 / 擦除周期 (注 1)	$N_{PEC}$	1000 (注 2)	—	—	—	次		
编程时的挂起延迟时间	$t_{SPD}$	—	—	120	$\mu\text{s}$	图 33.24		
擦除时的第 1 次挂起延迟时间 (在挂起优先模式中)	$t_{SESD1}$	—	—	120	$\mu\text{s}$	PCLK=50MHz 时		
擦除时的第 2 次挂起延迟时间 (在挂起优先模式中)	$t_{SESD2}$	—	—	1.7	ms			
擦除时的挂起延迟时间 (在擦除优先模式中)	$t_{SEED}$	—	—	1.7	ms			
数据的保持时间 (注 3)	$t_{DRP}$	10	—	—	—	年		

注 1. 再编程 / 擦除周期的定义:

再编程 / 擦除周期是指每块的擦除次数。如果再编程 / 擦除周期为  $n$  ( $n=1000$ )，就能逐块擦除  $n$  次。例如，对于 4K 字节的块，如果将 256 字节的数据分 16 次写到不同的地址后擦除该块，再编程 / 擦除周期次数就计为 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。

## 33.8 数据闪存（保存数据的闪存）特性

表 33.20 数据闪存（保存数据的闪存）特性

注. 表中没有记载条件的项目的规格值在条件 1 ~ 3 中通用。

条件 1: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=3.0 ~ 3.6V、VREFH0=3.0V ~ AVCC0、VREF=3.0V ~ AVCC

条件 2: VCC=PLLVC=2.7 ~ 3.6V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

条件 3: VCC=PLLVC=4.0 ~ 5.5V、VSS=PLLVS=AVSS0=AVSS=VREFL0=0V  
AVCC0=AVCC=4.0 ~ 5.5V、VREFH0=4.0V ~ AVCC0、VREF=4.0V ~ AVCC

编程 / 擦除时的工作温度范围:  $T_a = -40 \sim +85^\circ\text{C}$ 。  $T_a$  在条件 1 ~ 3 中通用。

项目	符号	min	typ	max	单位	测量条件	
编程时间	8 字节	$t_{DP8}$	—	0.4	2	ms	PCLK=50MHz 时
	128 字节	$t_{DP128}$	—	1	5	ms	
擦除时间	2K 字节	$t_{DE2K}$	—	70	250	ms	PCLK=50MHz 时
空白检查时间	8 字节	$t_{DBC8}$	—	—	30	$\mu\text{s}$	PCLK=50MHz 时
	2K 字节	$t_{DBC2K}$	—	—	0.7	ms	
再编程 / 擦除周期 (注 1)	$N_{DPEC}$	30000 (注 2)	—	—	—	次	
编程时的挂起延迟时间	$t_{DSPD}$	—	—	—	120	$\mu\text{s}$	图 33.24 PCLK=50MHz 时
擦除时的第 1 次挂起延迟时间 (在挂起优先模式中)	$t_{DSESD1}$	—	—	—	120	$\mu\text{s}$	
擦除时的第 2 次挂起延迟时间 (在挂起优先模式中)	$t_{DSESD2}$	—	—	—	1.7	ms	
擦除时的挂起延迟时间 (在擦除优先模式中)	$t_{DSEED}$	—	—	—	1.7	ms	
数据的保持时间 (注 3)	$t_{DDRP}$	10	—	—	—	年	

注 1. 再编程 / 擦除周期的定义:

再编程 / 擦除周期是指每块的擦除次数。如果再编程 / 擦除周期为  $n$  ( $n=30000$ )，就能逐块擦除  $n$  次。例如，对于 2K 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 / 擦除周期次数就计为 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。

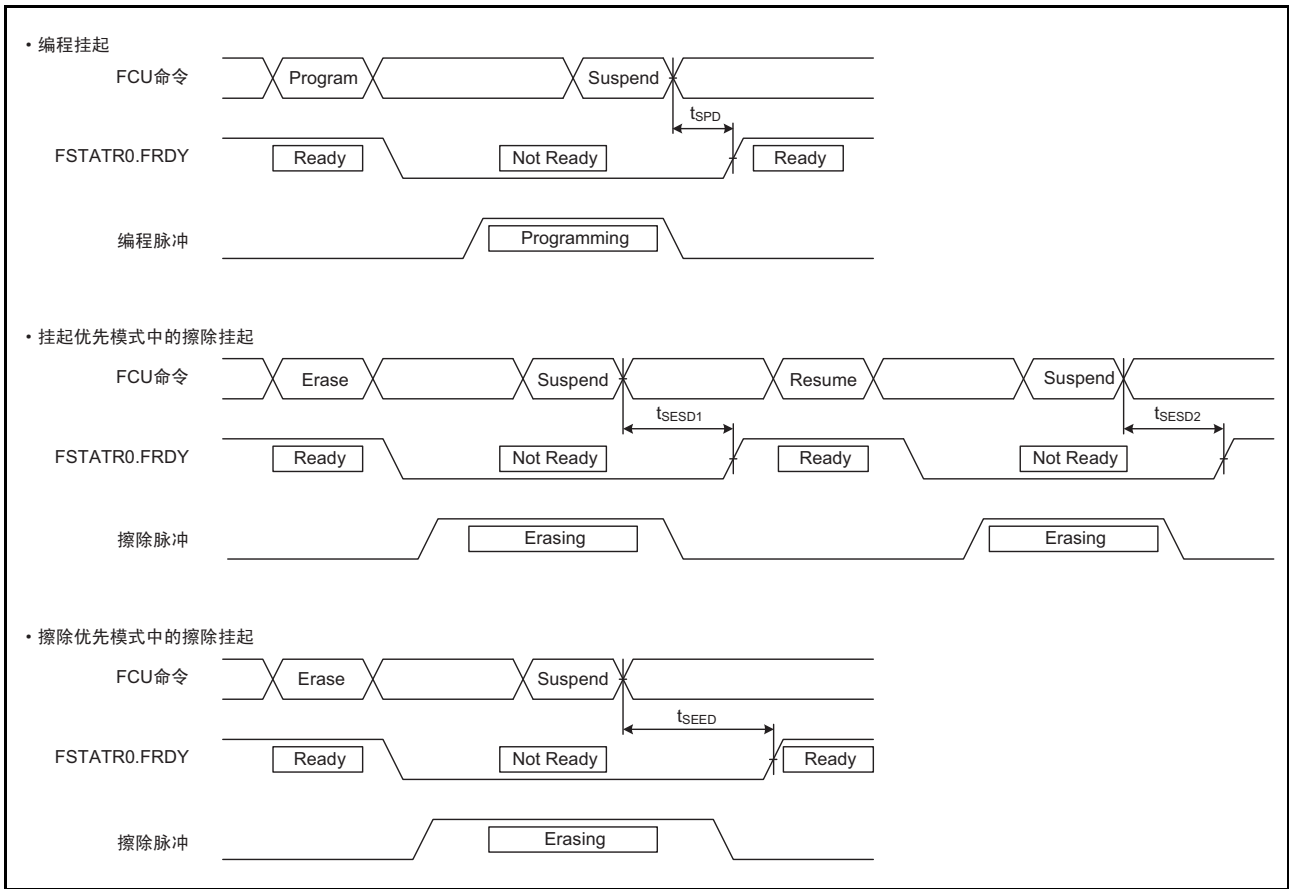


图 33.24 闪存的编程 / 擦除挂起时序

# 附录

## 附录 1. 各运行模式中的端口状态

附表 1.1 各运行模式中的端口状态

端口名 引脚名	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 *	IOKEEP = 0
端口 1	Hi-Z	Keep-O (注 1)	Keep-O (注 2)	Keep	Hi-Z
P20、P21	Hi-Z	Keep-O (注 1)	Keep	Keep	Hi-Z
P22 ~ P24	Hi-Z	Keep-O	Keep	Keep	Hi-Z
端口 3	Hi-Z	Keep-O	Keep	Keep	Hi-Z
端口 4	Hi-Z	Keep-O	Keep	Keep	Hi-Z
端口 5	Hi-Z	Keep-O	Keep	Keep	Hi-Z
端口 6	Hi-Z	Keep-O	Keep	Keep	Hi-Z
P70	Hi-Z	Keep-O (注 1)	Keep	Keep	Hi-Z
P71 ~ P76	Hi-Z	Hi-Z	Keep	Keep	Hi-Z
8	Hi-Z	Keep-O	Keep	Keep	Hi-Z
P90 ~ P95	Hi-Z	Hi-Z	Keep	Keep	Hi-Z
P96	Hi-Z	Keep-O (注 1)	Keep	Keep	Hi-Z
端口 A	Hi-Z	Keep-O	Keep	Keep	Hi-Z
PB0 ~ PB3	Hi-Z	Keep-O	Keep	Keep	Hi-Z
PB4	Hi-Z	Keep-O (注 1)	Keep	Keep	Hi-Z
PB5 ~ PB7	Hi-Z	Keep-O	Keep	Keep	Hi-Z
端口 D	Hi-Z	Keep-O	Keep	Keep	Hi-Z
PE0 ~ PE1	Hi-Z	Keep-O	Keep	Keep	Hi-Z
PE2	Hi-Z	[当设定 NMI 为解除源时] NMI [上述以外] Keep-O	[当设定 NMI 为解除源时] NMI [上述以外] Keep	[当设定 NMI 为解除源时] NMI [上述以外] Keep	[当设定 NMI 为解除源时] NMI [上述以外] Hi-Z
PE3 ~ PE5	Hi-Z	Keep-O	Keep	Keep	Hi-Z
PG0 ~ PG2	Hi-Z	Keep-O	Keep	Keep	Hi-Z
PG3 ~ PG5	Hi-Z	Keep-O	Keep	Keep	Hi-Z
WDTOVF#	WDTOVF 出力	H	H	H	

### 【符号说明】

H : High 电平

L : Low 电平

Keep-O : 当用作输出引脚时, 保持以前的值; 当用作输入引脚时, 为高阻抗。

Keep : 保持软件待机时的引脚状态。

HiZ : 高阻抗

IOKEEP=1\*: 将 DPSBYCR.IOKEEP 位清“0”前, 保持 I/O 端口的状态。

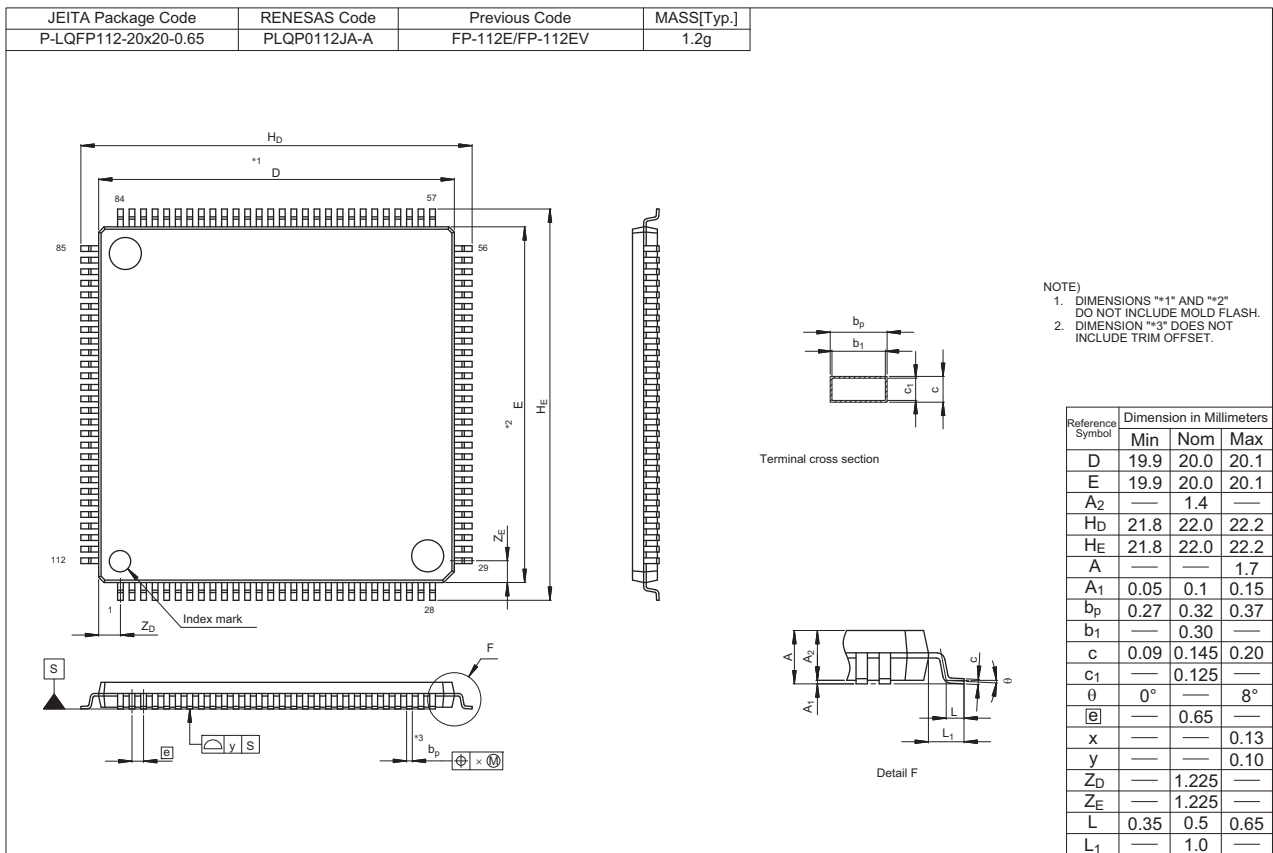
注. I/O 端口的结构因封装而不同。有关各封装的 I/O 端口, 请参照“15. I/O 端口”。

注 1. 当作为 IRQ 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。

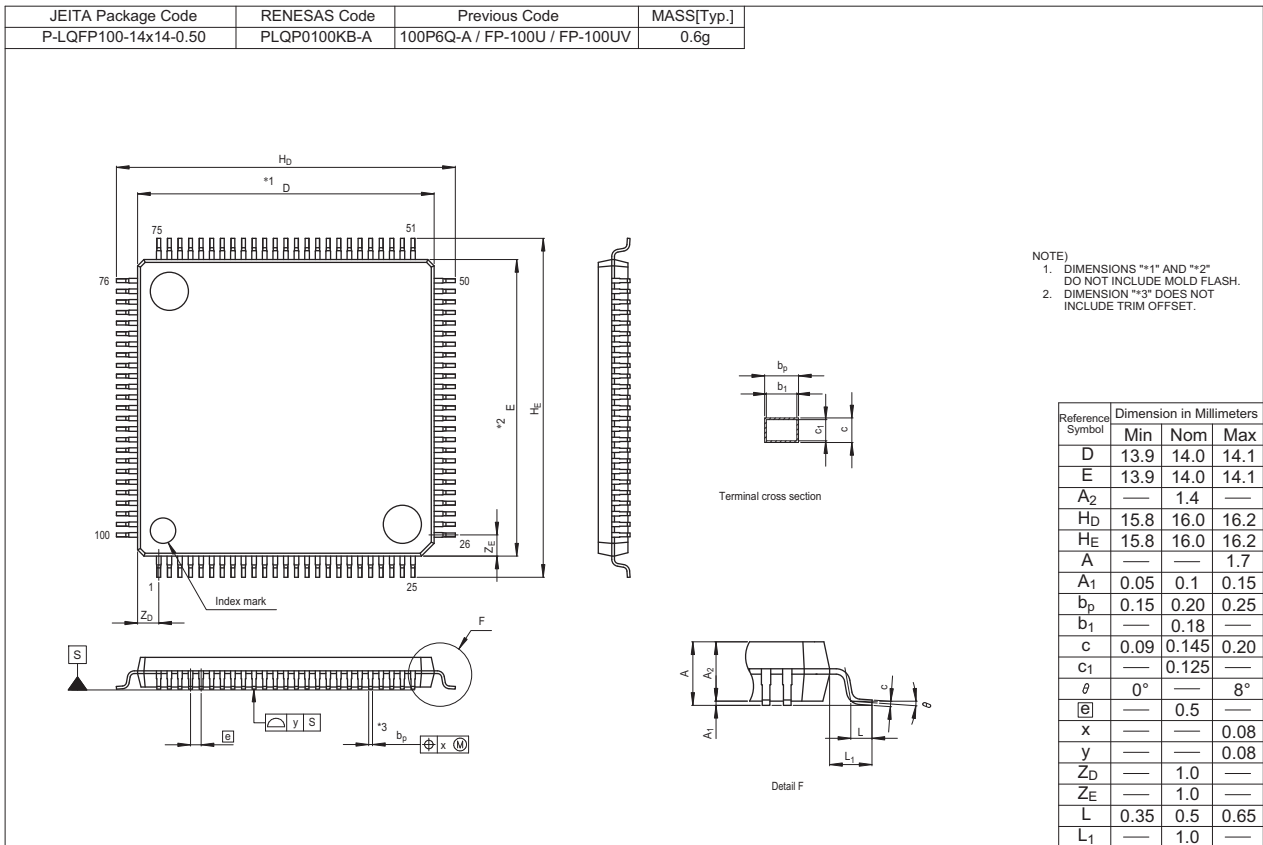
注 4. 当作为 IRQ 引脚使用时, 能在设定为深度软件待机解除源的情况下进行输入。

附录 2. 封装尺寸图

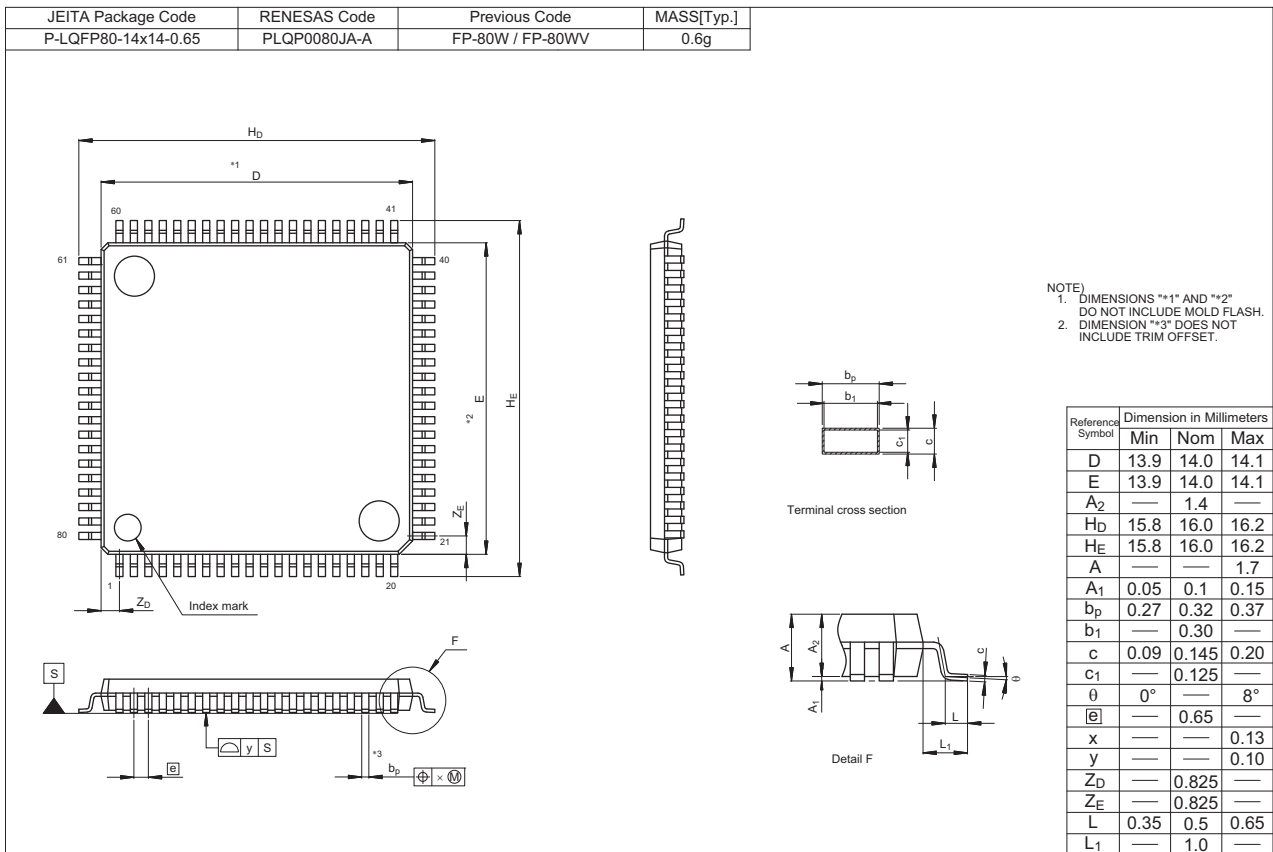
有关封装尺寸图的最新版和安装的信息，刊登在瑞萨电子的主页的“封装”中。



附图 2.1 112 引脚 LQFP (PLQP0112JA-A) 封装尺寸图

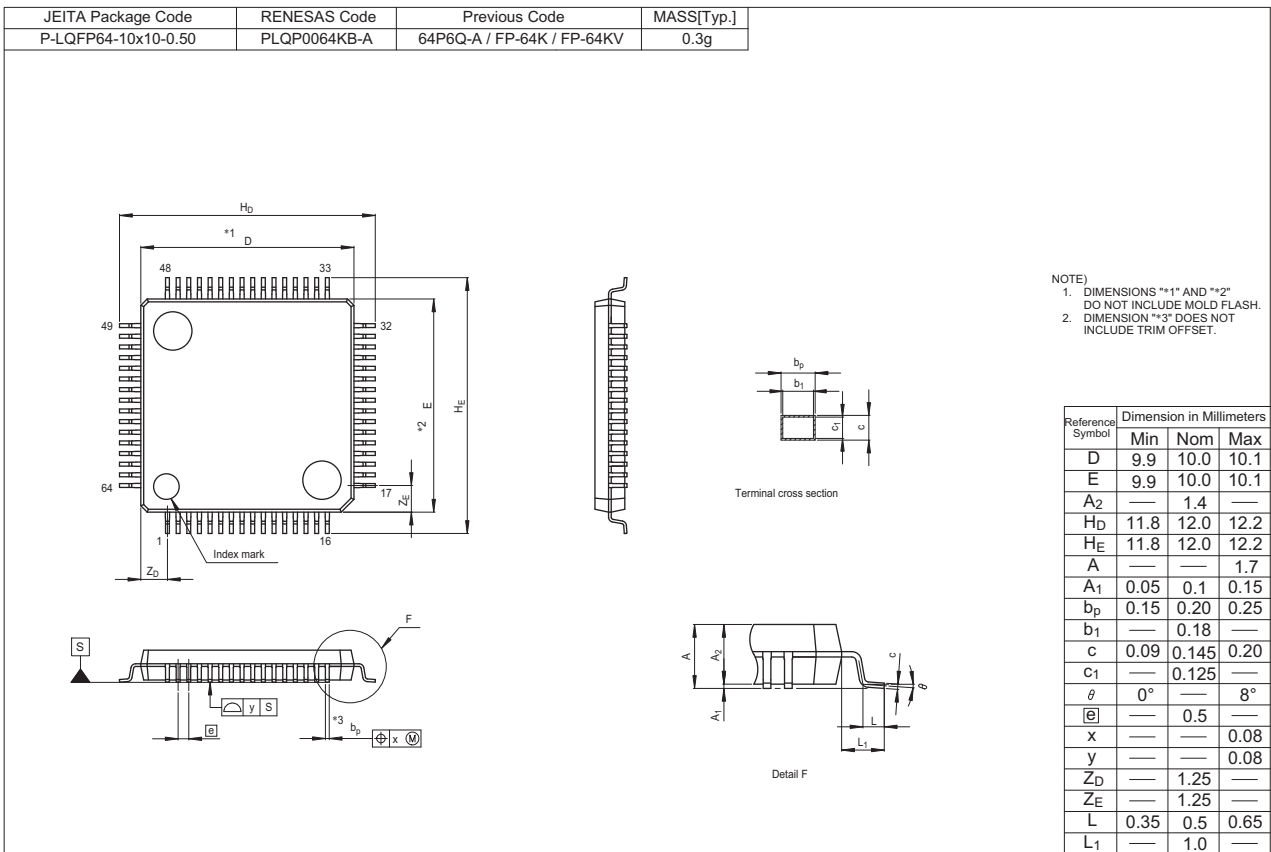


附图 2.2 100 引脚 LQFP (PLQP0100KB-A) 封装尺寸图

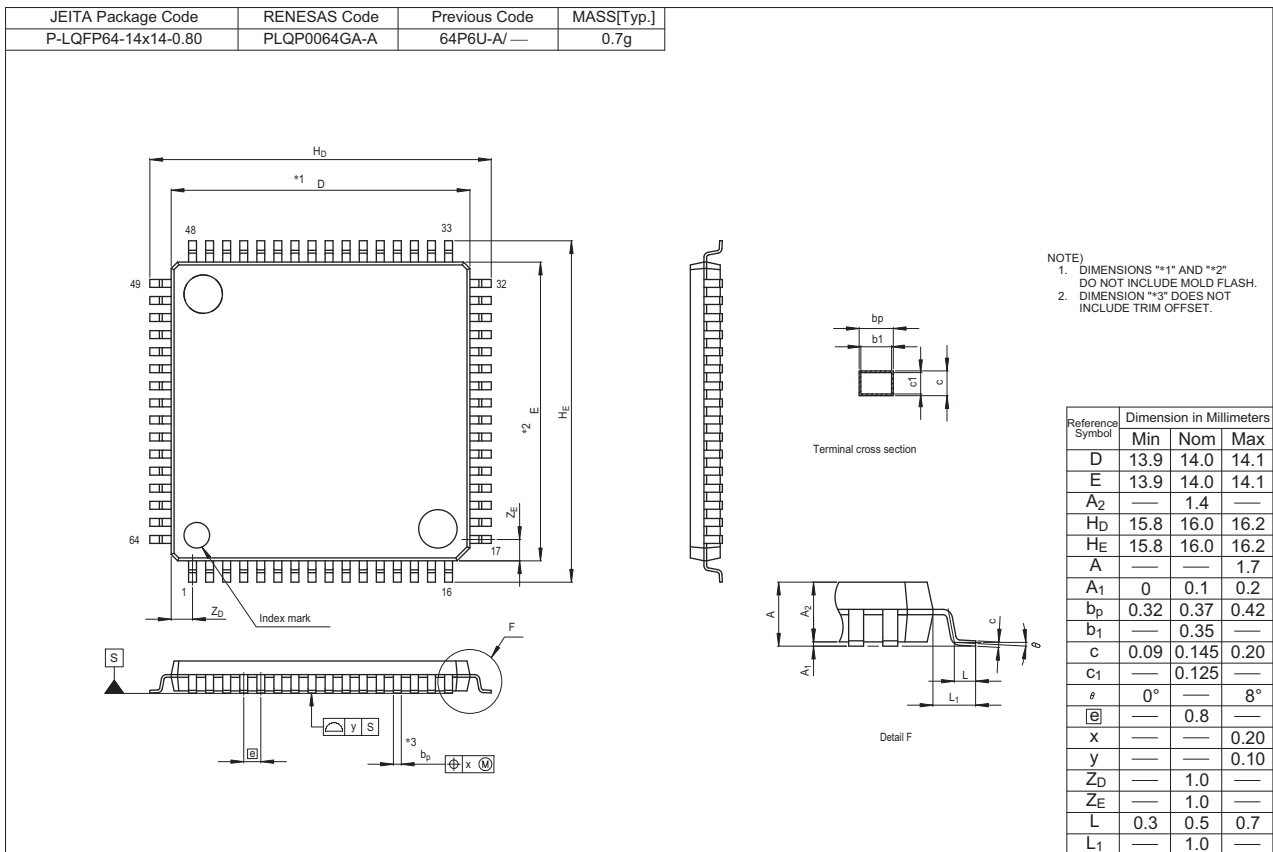


附图 2.3 80 引脚 LQFP (PLQP0080JA-A) 封装尺寸图





附图 2.4 64 引脚 LQFP (PLQP0064KB-A) 封装尺寸图



附图 2.5 64 引脚 LQFP (PLQP0064GA-A)

Rev.	发行日	修订内容	
		页	修订处
1.00	2011.09.13	—	初版发行
1.31	2012.06.28	2	在表 1.1 的中央处理器说明中追加“存储器保护单元 (MPU)”。
		6	在表 1.1 的封装中追加“64 引脚 LQFP (PLQP0064GA-A (14×14-0.8mm 节距))”。
		7	在表 1.2 的封装中追加“LQFP1414 (0.8mm 节距)”。
		9	替换表 1.3。
		10	在图 1.1 中追加“FK: LQFP64-0.80”。
		11	在图 1.2 中追加“MPU: 存储器保护单元”。
		15	在图 1.6 中追加“PLQP0064GA-A”。
		32	在 2.1 中追加“• 存储器保护单元”。
		49	在 2.6.1 中追加“存取异常”；将图 2.8 中地址 FFFFFFFD4 的“(保留区)”更改为“存取异常”。
		52	替换表 2.13 中的助记符内容。
		54	替换表 2.14 中的助记符内容；在表 2.14 中追加“算术 / 逻辑运算指令 (乘法 32bit×32bit64bit) (存储器源操作数)”。
		69	在表 5.1 中追加“区域 0 起始页码寄存器”~“数据命中寄存器”及其相关内容。
		79	在表 5.1 中追加“超时内部计数器 L”和“超时内部计数器 U”及其相关内容。
		P90	在表 5.1 中追加“通用 PWM 定时器启动的写保护寄存器”及其相关内容。
		96	更改注 6。
		97 ~ 126	替换表 5.2。
		145	删除 8.3.1 中的“晶体谐振器必须使用 AT-cut 并联谐振型”；删除图 8.3 中的“并联谐振型 cut-AT”。
		148	将 8.10.1 中的“振荡停止中断请求”更改为“中断请求”。
		155	在 9.2.2 的位表中追加“复位后的值”。
		161	更改 9.2.7 的部分说明内容。
		167	在 9.5.1.1 中追加部分说明内容。
		168	将 9.5.2.1 中的“传达目标”更改为“请求目标”；将 9.5.2.2 中的“电压检测电路”更改为“电压监视”；。
		169	将 9.5.3.1 中的“传达目标”更改为“请求目标”。
		171	更改图 9.2。
		174	更改图 9.3。
		175	更改图 9.4。
		176	更改 9.6.6 的第 2 段说明内容。
		177	在图 10.1 中追加“存取异常”；追加 10.1.3。
		180	在表 10.1 和表 10.2 中追加“存取异常”的相关内容。
		182	追加 10.5.3。
		184	在表 10.3 和表 10.4 中追加“存取异常”的相关内容。
		185	将表 11.1 中的“低电源电压检测”更改为“电压监视中断”，“振荡停止检测”更改为“振荡停止检测中断”。
		186	将图 11.1 中的“低电源电压检测”更改为“电压监视中断”，“振荡停止检测”更改为“振荡停止检测中断”；将“传送目标”更改为“请求目标”。

Rev.	发行日	修订内容	
		页	修订处
1.31	2012.06.28	200	在 11.2.7 的说明中追加“但是，在更改为 Low 电平时不需要清除 IR 标志”。
		201	将 11.2.8 中的“低电源电压检测”更改为“电压监视中断”，“振荡停止检测”更改为“振荡停止检测中断”。
		202	将 11.2.9 中的“低电源电压检测”更改为“电压监视中断”，“振荡停止检测”更改为“振荡停止检测中断”。
		207	删除表 11.7 中的 1 个“IPR52”。
		215	将 11.5 中的“低电源电压检测”更改为“电压监视中断”，“振荡停止检测”更改为“振荡停止检测中断”。
		227 ~ 247	追加第 13 章。
		274	在表 14.10 中追加注 7。
		306	将表 15.6 中的 PB0 的“MISOE”更改为“MOSIE”。
		331	将表 15.13 中的 PB0 的“MISOE”更改为“MOSIE”。
		346	将 15.3.2.8 的位表中的“选择 PD2 为 GTIOC2B-B 引脚”更改为“（不能选择 GTIOC2B-B 引脚）”。
		354	将表 15.20 中的 PB0 的“MISOE”更改为“MOSIE”。
		367	将 15.4.2.6 的位表中的“选择 PD2 为 GTIOC2B-B 引脚”更改为“（不能选择 GTIOC2B-B 引脚）”。
		374	将表 15.27 中的 PB0 的“MISOE”更改为“MOSIE”。
		377、378	追加图 15.1 和图 15.2。
		383	将图 16.1 中的“MTI2”更改为“MTU2”。
		400	删除表 16.4 注中的“n=0、6”。
		420	分别在 TGFE 标志和 TGFF 标志后追加“（比较匹配标志 E）”和“（比较匹配标志 F）”。
		427	将 16.2.14 中 TSYRA 的位表的地址“MTU.TSTRA 000C 1281h”更改为“MTU.TSYRA 000C 1281h”。
		428	将 16.2.14 中 TSYRB 的位表的地址“MTU.TSTRB 000C 1A81h”更改为“MTU.TSYRB 000C 1A81h”。
		429	更改 16.2.15 中的注 1。
		432	在 16.2.17 中追加第 4 段和第 5 段说明内容。
		434	更改 16.2.18 中的注。
		436	在 16.2.19 中追加注 2。
		443	更改 16.2.28 中的注。
		445 ~ 448	更改 16.2.29 中的注。
		468	在 16.3.4 的表 16.59 下的说明中追加部分内容。
		470	在图 16.24 中追加注 1。
		472	将 16.3.5(b) 中的“同步寄存器”更改为“周期寄存器”。
		490	将图 16.39 中 [8] 的“载波周期数据寄存器”更改为“定时器周期数据寄存器”、“载波周期缓冲寄存器”更改为“定时器周期缓冲寄存器”；在 [13] 中追加“同时”。
		499	将 16.3.8(2)(j) 中第二段的“数据寄存器”更改为“比较寄存器”，将第三段中的“实线”更改为“MTU3.TCNT”、“点线”更改为“MTU4.TCNT”，将第六段中的“反相 OFF”更改为“反相 ON”。

Rev.	发行日	修订内容	
		页	修订处
1.31	2012.06.28	501	将 16.3.8(2)(K) 中第二段的“数据寄存器”更改为“比较寄存器”。
		518	将图 16.80 中的“通用寄存器”更改为“比较寄存器”。
		519	在图 16.81 的 (1) 中追加“比较寄存器”及其相关图。
		541	在图 16.118 中追加注。
		543	在图 16.120 中追加注。
		553	将图 16.136 中的“TCDR”和“TDDR”更改为“TCDRA”和“TDDRA”。
		556	在 16.6.20 中追加第二段说明内容。
		557	在 16.6.21 中追加第三段说明内容。
		566	将图 16.152 中的“MTIOCnB”和“MTIOCnD”更改为“MTIOC3B”和“MTIOC3D”。
		571	将图 16.162 中的“MTIOCnB”和“MTIOCnD”更改为“MTIOC3B”和“MTIOC3D”。
		580	将 17.2.1 中注 1 的“上电复位”更改为“复位”。
		581	将 17.2.2 中注 1 的“上电复位”更改为“复位”。
		582	将 17.2.3 中注 1 的“上电复位”更改为“复位”。
		584	将 17.2.4 中注 1 的“上电复位”更改为“复位”。
		585	将 17.2.5 中注 1 的“上电复位”更改为“复位”。
		586	将 17.2.6 中注 1 的“上电复位”更改为“复位”。
		587	将 17.2.7 中注 1 的“上电复位”更改为“复位”。
		588	将 17.2.8 中注 1 的“上电复位”更改为“复位”。
		591	将 17.2.10 中注 1 的“上电复位”更改为“复位”。
		592	将 17.2.11 中注 1 的“上电复位”更改为“复位”。
		594	将 17.2.12 中注 1 的“上电复位”更改为“复位”。
		595	将 17.2.13 中注 1 的“上电复位”更改为“复位”。
		597	将 17.2.14 中注 1 的“上电复位”更改为“复位”。
		598	将 17.2.15 中注 1 的“上电复位”更改为“复位”。
		608	将 17.3.7 中的“上电复位”更改为“复位”。
		611	在图 18.1 中追加“GTSWP: 通用 PWM 定时器启动的写保护寄存器”。
		613	在表 18.4 的 GPT 中追加“通用 PWM 定时器启动的写保护寄存器”及其相关内容。
		614	将表 18.4 中 GPT0 的 GTSOS 的复位后的值更改为“0x00h”。
		615	将表 18.4 中 GPT1 的 GTSOS 的复位后的值更改为“0x00h”；将 GPT2 的 GTSOS 的复位后的值更改为“0x00h”。
		616	将表 18.4 中 GPT3 的 GTSOS 的复位后的值更改为“0x00h”。
		627	追加 18.2.10。
		628	调换 18.2.11 位表功能项中“b13-b12”和“b10-b8”的位置，
		638	将 18.2.19 中的“b8-b9”更改为“b9-b8”。
		641	将 18.2.21 位图中的 b0 的复位后的值更改为“1”。
		644	将 18.2.23 位表中带有“(注 1)”的“R/W”更改为“R/(W)”。
		648	更改 18.2.25 位图中的部分地址。
		650	将 18.2.32 位图中的 b8 的复位后的值更改为“1”，将位表中的“b11-b8”更改为“b11-b9”。
		693	将图 18.46 中的“通过硬件启动”更改为“通过软件启动”。
		714	在 18.4.1 中追加 (5)。

Rev.	发行日	修订内容	
		页	修订处
1.31	2012.06.28	760	删除表 22.1 中的注 1。
		766	在 22.2.1.6 的位表中追加“当 SEMR.ABCS 位为“1”时，必须输入频率为 8 倍位速率的时钟。”。
		776	更改表 22.10 中的第 3 个表的 50.00 位速率的相关值。
		780	更改图 22.3。
		782	在图 22.5 中追加注 2。
		794	在图 22.17 中追加注 2。
		825	将 23.2.1 位表中的“W”更改为“R/W”。
		826	将 23.2.3 中的“高位字节（b15 -b8）为“00h”。”更改为“不更新高位字节（b15 -b8）。”。
		834	在表 24.3 中追加“超时内部寄存器 L”和“超时内部寄存器 U”及其相关内容。
		842	更改 24.2.4 位图和位表中 b3 的内容；在位表中追加注 1。
		843	在 24.2.4 中追加 TMWE 位说明；在 DLCS 位说明中追加部分内容。
		864	追加 24.2.18。
		866	替换图 24.5。
		868	替换图 24.6。
		872	替换图 24.10。
		873	更改图 24.11 和图 24.12 中的部分内容。
		874	替换图 24.13。
		875	替换图 24.14。
		877	替换图 24.17。
		883	替换图 24.25。
		941	删除 25.219 标题中的“CAN0”。
		948	将图 25.9 中的“CANM”更改为“CANM[1:0]”。
		955	更改图 25.13 的标题。
		962	将表 25.12 中的 RXF0 和 TXF0 的中断源标志更改为“—”；将“C0EIFR.BEIF”更改为“EIFR.BEIF”。
		970	将 26.2.4 位表中带有“（注 2）”的“R/W”更改为“R/(W)”。
		988	将表 26.7 中主控和从属（时钟同步运行）的引脚“SSL1 ~ 3”更改为“SSL0 ~ 3”。
		999	将图 26.15 传送结束中的“T00”更改为“R00”。
		1010	将 26.3.8.2(2) 中的“SPSR.OVRF”更改为“SPSR.RERF”。
		1012	在 26.3.10 的 26.3.10.1(2) 中追加“（m=0 ~ 7）”。
		1034	将表 27.1 中可变帧结构内容中的“bit”更改为“Tbit”。
1046	在 27.2.12 的 RTS 位说明中追加“如果要给 RTS 位写“1”，就必须通过 MOV 指令写“02h”。”。		
1085	将 28.2.3(2) 位表中的地址“0008 9004h”更改为“0008 9084h”。		
1087	在 28.2.4 的 SHBYP 位说明中追加“如果要使用可编程增益放大器，必须将 SHBYP 位设定为“0”（使用采样 & 保持电路）。”。		
1100	将 28.2.12 中的“逻辑与”更改为“逻辑或”。		
1109	在 28.3.6 中追加“如果要使用可编程增益放大器，必须将 ADCER.SHBYP 位设定为“0”（使用采样 & 保持电路）。”。		

## 修订记录

## RX62T 群 用户手册 硬件篇

Rev.	发行日	修订内容	
		页	修订处
1.31	2012.06.28	1113	将 28.5.5 中的“0.3k”更改为“3.0k”；更改 28.5.7 中的“各电源引脚（AVCC0-AVSS0、AVCC-AVSS、VCC-VSS）的关系”。
		1119	将 29.2.1 位表地址中的“ADDR1 0008 8066Eh”更改为“ADDR1 0008 8066h”。
		1137	更改 29.6.7 中的“各电源引脚（AVCC0-AVSS0、AVCC-AVSS、VCC-VSS）的关系”。
		1140	更改表 31.1 中板上编程的内容。
		1200	更改 31.9.5(10) 中位速率误差的计算式。
		1212	将 31.12(2) 中的“编程 / 擦除命令”更改为“编程 / 擦除挂起命令”；并且更改 (4)、(5)、(7)、(8) 的内容。
		1213	更改表 32.1 中板上编程的内容。
		1240	更改 32.9(2)。
		1241	在表 33.1 的注 1 中追加部分内容。
		1249	更改表 33.7 的注 1 和注 3。
		1250	将表 33.8 中输出时钟周期的 min 更改为 $16 \times t_{Pcyc}$ 。
		1258	将表 33.13 第二个表中的“采样的 15 个状态”更改为“采样的 25 个状态”。
		1262	更改图 33.20。

---

RX62T 群  
用户手册 硬件篇

Publication Date: Rev.1.00 Sep 13, 2011  
Rev.1.31 Jun 28, 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

RX62T群



瑞萨电子株式会社

R01UH0034CJ0131