

# R32C/117 群

用户手册 硬件篇

瑞萨单片机  
M16C 族 / R32C/100 系列

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

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
  - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
  - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

## 产品使用时的注意事项

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

### 1. 未使用的引脚的处理

**【注意】**将未使用的引脚按照正文的“未使用引脚的处理”进行处理。

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

### 2. 通电时的处理

**【注意】**通电时产品处于不定状态。

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

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

### 3. 禁止存取保留地址（保留区）

**【注意】**禁止存取保留地址（保留区）

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

### 4. 关于时钟

**【注意】**复位时，请在时钟稳定后解除复位。

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

### 5. 关于产品间的差异

**【注意】**在变更不同型号的产品时，请对每一个产品型号进行系统评价测试。

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

# 本手册的使用方法

## 1 目的和对象

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

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

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

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

R32C/117 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
数据表	硬件的概要和电特性	R32C/117 Group Datasheet	REJ03B0254
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 ※外围功能的使用方法必须参照应用注意事项。	R32C/117 群硬件手册	本硬件手册
软件手册	CPU 指令设定的说明	R32C/100 Series Software Manual	REJ09B0267
应用注意事项	外围功能的使用方法、应用例子 参考程序 应用汇编语言、C 语言编成方法	刊登在瑞萨电子的主页上	
RENESAS TECHNICAL UPDATE	相关产品说明、文献等的快速公告		

## 2 数字、符号的表示

本手册使用的寄存器名或位名、数字或符号的表示范例如下所示。

1. 寄存器名、位名、引脚名

在正文中用符号表示。符号后面带有寄存器、位、引脚字样加以区别。

(例) PM0 寄存器的 PM03 位

P3\_5 引脚、VCC 引脚

2. 数字的表示

2 进制数的后面带有“b”，但是在只有 1 位时数字后面什么也没有；16 进制数后面带有“h”；十进制数后面什么也没有。

(例) 2 进制数: 11b

16 进制数: EEA0h

10 进制数: 1234

### 3 寄存器图表的阅读方法

说明在寄存器图中使用的符号和用语

位符号	位名	功能	RW
... 0	... 位	b1 b0 0 0: ... 0 1: ... 1 0: 不能设定 1 1: ...	RW
... 1		RW	
— (b2)		什么也不指定。只能写“0”，读取值为不定值。	—
— (b3)		保留位	必须置“1”。
— (b4)	保留位	必须置“0”。读取值为不定值。	RW
... 5	... 位	功能因运行模式而不同。	WO
... 6			WO
... 7	... 标志	0: ... 1: ...	RO

**\* 1**  
 空白：必须根据用途置“0”或者置“1”。  
 0：必须置“0”。  
 1：必须置“1”。  
 —：什么也不指定的位。

**\* 2**  
 RW：读时能读位的值。写入值为有效数据。  
 RO：读时能读位的值。写入值为无效数据。  
 WO：写入值为有效数据。不能读位的值（读取值为不定值）。  
 —：什么也不指定的位。

**\* 3**  
 • 保留位  
 保留位，必须写指定值。有关RW位，在没有特殊记载时能读写入值。

**\* 4**  
 • 什么也不指定  
 对相应的位什么也不指定。根据将来外围功能的展开，有可能出现新的功能。因此，只能写“0”。  
 • 不能设定  
 不能保证设定后的运行。  
 • 功能因运行模式而不同  
 位功能随外围功能的模式而变。请参照各模式的寄存器图。

#### 4 省略语及简称的说明

省略 / 简称	全称	备注
ACIA	Asynchronous Communication Interface Adapter	异步通信接口适配器
bps	bits per second	位 / 秒; 每秒传送位数
CRC	Cyclic Redundancy Check	循环冗余校验
DMA	Direct Memory Access	直接存储器存取
DMAC	Direct Memory Access Controller	DMA 控制器
GSM	Global System for Mobile Communications	全球数字移动电话系统
Hi-Z	High Impedance	高阻抗
IEBus	Inter Equipment bus	-
I/O	Input/Output	输入 / 输出
IrDA	Infrared Data Association	红外线数据协会
LSB	Least Significant Bit	最低有效位
MSB	Most Significant Bit	最高有效位
NC	Non-Connection	未连接引脚
PLL	Phase Locked Loop	锁相环路
PWM	Pulse Width Modulation	脉宽调制
SFR	Special Function Registers	特殊功能寄存器
SIM	Subscriber Identity Module	ISO-7816 规定的 IC 卡
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发器
VCO	Voltage Controlled Oscillator	电压控制振荡器

所有商标及注册商标分别归属于其所有者。

# 目 录

1. 概要 .....	1
1.1 特点 .....	1
1.1.1 用途 .....	1
1.1.2 规格概要 .....	2
1.2 产品一览表 .....	6
1.3 框图 .....	8
1.4 引脚连接图 .....	9
1.5 引脚功能的说明 .....	18
2. 中央处理器（CPU） .....	23
2.1 基本寄存器 .....	24
2.1.1 数据寄存器（R2R0、R3R1、R6R4、R7R5） .....	24
2.1.2 地址寄存器（A0、A1、A2、A3） .....	24
2.1.3 静态基址寄存器（SB） .....	24
2.1.4 帧基址寄存器（FB） .....	24
2.1.5 程序计数器（PC） .....	24
2.1.6 中断向量表基址寄存器（INTB） .....	24
2.1.7 用户堆栈指针（USP）和中断堆栈指针（ISP） .....	24
2.1.8 标志寄存器（FLG） .....	24
2.2 高速中断寄存器 .....	26
2.2.1 标志保存寄存器（SVF） .....	26
2.2.2 PC 保存寄存器（SVP） .....	26
2.2.3 向量寄存器（VCT） .....	26
2.3 DMAC 相关寄存器 .....	26
2.3.1 DMA 模式寄存器（DMD0、DMD1、DMD2、DMD3） .....	26
2.3.2 DMA 终端计数寄存器（DCT0、DCT1、DCT2、DCT3） .....	26
2.3.3 DMA 终端计数重加载寄存器（DCR0、DCR1、DCR2、DCR3） .....	27
2.3.4 DMA 源地址寄存器（DSA0、DSA1、DSA2、DSA3） .....	27
2.3.5 DMA 源地址重加载寄存器（DSR0、DSR1、DSR2、DSR3） .....	27
2.3.6 DMA 目标地址寄存器（DDA0、DDA1、DDA2、DDA3） .....	27
2.3.7 DMA 目标地址重加载寄存器（DDR0、DDR1、DDR2、DDR3） .....	27
3. 存储器 .....	28
4. SFR .....	29
5. 复位 .....	68
5.1 硬件复位 .....	68
5.2 软件复位 .....	71
5.3 看门狗定时器复位 .....	71
5.4 复位向量 .....	71
6. 电源管理 .....	72
6.1 电压稳压器 .....	72
6.1.1 去耦电容器 .....	73
6.2 低电压检测电路 .....	73
6.2.1 低电压检测电路的运行 .....	75
6.2.2 低电压检测中断 .....	75
6.2.3 低电压检测电路的应用例子 .....	76



7.	处理器模式 .....	77
7.1	处理器模式的种类 .....	77
7.2	处理器模式的设定 .....	77
8.	时钟发生电路 .....	80
8.1	时钟发生电路的种类 .....	80
8.1.1	主时钟 .....	87
8.1.2	副时钟 (fC) .....	87
8.1.3	PLL 时钟 .....	88
8.1.4	内部振荡器时钟 .....	91
8.2	振荡停止检测功能 .....	91
8.2.1	振荡停止检测功能的使用方法 .....	91
8.3	基本时钟 .....	91
8.4	CPU 时钟和外围总线时钟 .....	92
8.5	外围功能时钟 .....	92
8.6	时钟输出功能 .....	93
8.7	功率控制 .....	94
8.7.1	正常运行模式 .....	98
8.7.2	等待模式 .....	99
8.7.3	停止模式 .....	102
8.8	系统时钟的保护功能 .....	103
8.9	使用时钟发生电路时的注意事项 .....	104
8.9.1	副时钟 .....	104
8.9.2	功率控制 .....	104
9.	总线 .....	105
9.1	总线的设定 .....	105
9.2	外围总线时序的设定 .....	106
9.3	外部总线的设定 .....	107
9.3.1	外部地址空间的设定 .....	107
9.3.2	外部数据总线宽度的选择 .....	113
9.3.3	分离总线和多路复用总线的选择 .....	114
9.3.4	读写信号 .....	116
9.3.5	外部总线时序 .....	118
9.3.6	<u>ALE</u> 信号 .....	122
9.3.7	<u>RDY</u> 信号 .....	123
9.3.8	<u>HOLD</u> 信号 .....	126
9.3.9	<u>BCLK</u> 的输出 .....	126
9.4	存取内部区域时的外部总线状态 .....	126
9.5	使用总线时的注意事项 .....	127
9.5.1	系统设计时的注意事项 .....	127
9.5.2	设定寄存器时的注意事项 .....	127
10.	保护 .....	128
10.1	PRCR 寄存器 .....	128
10.2	PRCR2 寄存器 .....	129
10.3	PRCR3 寄存器 .....	129
10.4	PRR 寄存器 .....	130
11.	中断 .....	131
11.1	中断的分类 .....	131
11.2	软件中断 .....	131

11.3	硬件中断	132
11.3.1	特殊中断	132
11.3.2	外围功能中断	133
11.4	高速中断	133
11.5	中断向量	133
11.5.1	固定向量表	134
11.5.2	可向量表	134
11.6	中断请求的接受	139
11.6.1	I 标志和 IPL	139
11.6.2	中断控制寄存器	140
11.6.3	用于返回的中断优先级设定寄存器	142
11.6.4	中断响应顺序	143
11.6.5	中断响应时间	143
11.6.6	接受中断请求时的 IPL 变化	144
11.6.7	寄存器的压栈	144
11.7	从中断处理程序的返回	145
11.8	中断优先级	145
11.9	中断优先级的判断电路	146
11.10	外部中断	147
11.11	NMI	148
11.12	键输入中断	149
11.13	智能 I/O 中断	150
11.14	使用中断时的注意事项	153
11.14.1	ISP 的设定	153
11.14.2	NMI	153
11.14.3	外部中断	153
12.	看门狗定时器	154
13.	DMAC	156
13.1	传送周期	163
13.1.1	传送地址和数据总线宽度的影响	163
13.1.2	总线时序的影响	164
13.1.3	RDY 信号的影响	165
13.2	DMA 的传送周期数	166
13.3	通道优先级和 DMA 传送时序	166
13.4	使用 DMA 控制器时的注意事项	167
13.4.1	DMAC 相关寄存器的设定	167
13.4.2	读 DMAC 相关寄存器	167
14.	DMAC II	168
14.1	DMAC II 的设定	169
14.1.1	RIPL1 寄存器和 RIPL2 寄存器	169
14.1.2	DMAC II 索引	170
14.1.3	外围功能的中断控制寄存器	173
14.1.4	外围功能的可向量表	173
14.1.5	IIOiE 寄存器 (i=0 ~ 11) 的 IRLT 位	173
14.2	DMAC II 的运行	173
14.3	传送对象	173
14.3.1	存储器之间的传送	173
14.3.2	立即数传送	174
14.3.3	运算传送	174

14.4	传送方式 .....	174
14.4.1	单次传送 .....	174
14.4.2	突发传送 .....	174
14.4.3	多次传送 .....	174
14.5	链传送 .....	175
14.6	传送结束中断 .....	175
14.7	执行时间 .....	176
15.	可编程输入 / 输出端口 .....	177
15.1	端口 Pi 寄存器 (Pi 寄存器, i=0 ~ 15) .....	178
16.	定时器 .....	179
16.1	定时器 A .....	181
16.1.1	定时器模式 .....	185
16.1.2	事件计数器模式 .....	187
16.1.3	单触发定时器模式 .....	191
16.1.4	脉宽调制模式 .....	193
16.2	定时器 B .....	196
16.2.1	定时器模式 .....	198
16.2.2	事件计数器模式 .....	199
16.2.3	脉冲周期测量模式和脉宽测量模式 .....	200
16.3	使用定时器时的注意事项 .....	203
16.3.1	定时器 A 和定时器 B 的共同事项 .....	203
16.3.2	定时器 A .....	203
16.3.3	定时器 B .....	204
17.	三相马达控制定时器功能 .....	205
17.1	三相马达控制定时器的模式 .....	210
17.2	定时器 B2 .....	211
17.3	定时器 A4,A1,A2 .....	212
17.4	上下同时导通输出禁止功能和死区时间定时器 .....	214
17.5	三相马达控制定时器的运行例子 .....	216
17.6	使用三相马达控制定时器功能时的注意事项 .....	218
17.6.1	关机功能 .....	218
17.6.2	寄存器的设定 .....	218
18.	串行接口 .....	219
18.1	时钟同步串行接口模式 .....	233
18.1.1	发生通信错误时的处理方法 .....	238
18.1.2	CLK 极性选择 .....	238
18.1.3	LSB first 或者 MSB first 的选择 .....	239
18.1.4	连续接收模式 .....	239
18.1.5	串行数据的逻辑转换 .....	240
18.1.6	CTS/RTS 功能 .....	240
18.2	异步串行接口模式 (UART 模式) .....	241
18.2.1	位速率 .....	246
18.2.2	发生通信错误时的处理方法 .....	247
18.2.3	LSB first 或者 MSB first 的选择 .....	247
18.2.4	串行数据的逻辑转换 .....	248
18.2.5	TXD 和 RXD 的输入 / 输出极性转换 .....	249
18.2.6	CTS/RTS 功能 .....	249

18.3	特殊模式 1 (I <sup>2</sup> C 模式)	250
18.3.1	开始条件和停止条件的检测	255
18.3.2	开始条件和停止条件的生成	255
18.3.3	仲裁	256
18.3.4	SCL 控制和时钟同步	257
18.3.5	SDA 输出	258
18.3.6	SDA 输入	259
18.3.7	应答	259
18.3.8	发送和接收的初始化	259
18.4	特殊模式 2	260
18.4.1	SSi 输入引脚的功能 (i=0 ~ 6)	262
18.4.2	时钟相位的设定功能	263
18.5	使用串行接口时的注意事项	265
18.5.1	UiBRG 寄存器 (i=0 ~ 8) 的变更	265
18.5.2	时钟同步模式	265
18.5.3	特殊模式 1 (I <sup>2</sup> C 模式)	265
19.	A/D 转换器	266
19.1	模式的说明	272
19.1.1	单次模式	272
19.1.2	重复模式	273
19.1.3	单次扫描模式	274
19.1.4	重复扫描模式 0	275
19.1.5	重复扫描模式 1	276
19.1.6	多端口单次扫描模式	277
19.1.7	多端口重复扫描模式 0	278
19.2	功能	279
19.2.1	分辨率的选择功能	279
19.2.2	采样 & 保持	279
19.2.3	触发的选择功能	279
19.2.4	DMAC 利用模式	279
19.2.5	扩展模拟输入引脚	280
19.2.6	外部运算放大器的连接模式	280
19.2.7	降低消耗电流的功能	281
19.2.8	传感器的输出阻抗	281
19.3	使用 A/D 转换器时的注意事项	283
19.3.1	设计电路板时的注意事项	283
19.3.2	编程时的注意事项	284
20.	D/A 转换器	285
21.	CRC 运算电路	287
22.	X-Y 转换电路	289
22.1	读时的数据转换	290
22.2	写时的数据转换	292
23.	智能 I/O	293
23.1	基本定时器 (组 0 ~ 组 2)	306
23.2	时间测量功能 (组 0 和组 1)	311
23.3	波形生成功能 (组 0 ~ 组 2)	315
23.3.1	单相波形输出模式 (组 0 ~ 组 2)	316

23.3.2	反相波形输出模式（组 0 ~ 组 2）	318
23.3.3	置位 - 复位波形输出（SR 波形输出）模式（组 0 ~ 组 2）	320
23.3.4	位调制 PWM 输出模式（组 2）	323
23.3.5	实时端口（RTP）输出模式（组 2）	325
23.3.6	并行实时端口（RTP）输出模式（组 2）	327
23.4	组 2 的通信功能	328
23.4.1	可变长时钟同步串行 I/O 模式（组 2）	333
<b>24.</b>	<b>多主控 I<sup>2</sup>C 总线接口</b>	<b>336</b>
24.1	多主控 I <sup>2</sup> C 总线接口电路的相关寄存器	338
24.1.1	多主控 I <sup>2</sup> C 总线发送 / 接收移位寄存器（I2CTRSR）	338
24.1.2	I <sup>2</sup> C 总线从属地址寄存器（I2CSAR）	339
24.1.3	I <sup>2</sup> C 总线控制寄存器 0（I2CCR0）	340
24.1.4	I <sup>2</sup> C 总线时钟控制寄存器（I2CCCR）	342
24.1.5	I <sup>2</sup> C 总线开始条件 / 停止条件的控制寄存器（I2CSSCR）	344
24.1.6	I <sup>2</sup> C 总线控制寄存器 1（I2CCR1）	345
24.1.7	I <sup>2</sup> C 总线控制寄存器 2（I2CCR2）	348
24.1.8	I <sup>2</sup> C 总线状态寄存器（I2CSR）	350
24.1.9	I <sup>2</sup> C 总线模式寄存器（I2CMR）	353
24.2	开始条件的生成方法	354
24.3	停止条件的生成方法	355
24.4	开始条件重复防止功能	356
24.5	开始条件 / 停止条件的检测	357
24.6	数据的发送 / 接收	359
24.6.1	主控发送的例子	360
24.6.2	从属接收的例子	361
24.7	使用时的注意事项	362
24.7.1	I <sup>2</sup> C 总线接口电路的各寄存器的存取	362
24.7.2	重新开始条件的生成	363
<b>25.</b>	<b>CAN 模块</b>	<b>364</b>
25.1	CAN SFR	366
25.1.1	CAN0 控制寄存器（COCTL 寄存器）	367
25.1.2	CAN0 时钟选择寄存器（COCLKR 寄存器）	371
25.1.3	CAN0 位结构寄存器（COBCR 寄存器）	372
25.1.4	CAN0 屏蔽寄存器 k（COMKRk 寄存器）（k=0 ~ 7）	374
25.1.5	CAN0 FIFO 接收 ID 比较寄存器 n（COFIDCR0、COFIDCR1 寄存器）（n=0,1）	375
25.1.6	CAN0 屏蔽无效寄存器（COMKIVLR 寄存器）	376
25.1.7	CAN0 邮箱（COMBj 寄存器）（j=0 ~ 31）	377
25.1.8	CAN0 邮箱中断允许寄存器（COMIER 寄存器）	380
25.1.9	CAN0 消息控制寄存器 j（COMCTLj 寄存器）（j=0 ~ 31）	381
25.1.10	CAN0 接收 FIFO 控制寄存器（CORFCR 寄存器）	384
25.1.11	CAN0 接收 FIFO 指针控制寄存器（CORFPCR 寄存器）	386
25.1.12	CAN0 发送 FIFO 控制寄存器（COTFCR 寄存器）	387
25.1.13	CAN0 发送 FIFO 指针控制寄存器（COTFPCR 寄存器）	388
25.1.14	CAN0 状态寄存器（COSTR 寄存器）	389
25.1.15	CAN0 邮箱检索模式寄存器（COMSMR 寄存器）	392
25.1.16	CAN0 邮箱检索状态寄存器（COMSSR 寄存器）	393
25.1.17	CAN0 通道检索支持寄存器（COCSSR 寄存器）	394
25.1.18	CAN0 接收滤波器支持寄存器（COAFSR 寄存器）	395
25.1.19	CAN0 错误中断允许寄存器（COEIER 寄存器）	396
25.1.20	CAN0 错误中断源判断寄存器（COEIFR 寄存器）	398

25.1.21	CAN0 接收错误计数寄存器 (C0RECR 寄存器)	400
25.1.22	CAN0 发送错误计数寄存器 (C0TECR 寄存器)	400
25.1.23	CAN0 错误码保存寄存器 (C0ECSR 寄存器)	401
25.1.24	CAN0 时戳寄存器 (C0TSR 寄存器)	402
25.1.25	CAN0 测试控制寄存器 (C0TCR 寄存器)	403
25.2	运行模式	405
25.2.1	CAN 复位模式	406
25.2.2	CAN Halt 模式	407
25.2.3	CAN 睡眠模式	407
25.2.4	CAN 操作模式 (总线关闭状态除外)	408
25.2.5	CAN 操作模式 (总线关闭状态)	409
25.3	CAN 通信速度的设定	410
25.3.1	CAN 时钟的设定	410
25.3.2	位时序的设定	410
25.3.3	位速率	411
25.4	邮箱和屏蔽寄存器的结构	412
25.5	接收滤波器的处理和屏蔽功能	413
25.6	接收和发送	415
25.6.1	接收	416
25.6.2	发送	418
25.7	CAN 中断	419
26.	输入 / 输出引脚	420
26.1	端口 Pi 方向寄存器 (PDi 寄存器, i=0 ~ 15)	421
26.2	输出功能选择寄存器	422
26.3	输入功能选择寄存器	439
26.4	上拉控制寄存器 0 ~ 4 (PUR0 ~ PUR4 寄存器)	443
26.5	端口控制寄存器 (PCR 寄存器)	446
26.6	未使用引脚的处理	447
27.	闪存	449
27.1	概要	449
27.2	闪存保护	451
27.2.1	锁定位保护	451
27.2.2	ROM 码保护	451
27.2.3	ID 码保护	452
27.3	CPU 改写模式	453
27.3.1	CPU 运行模式和闪存的改写	459
27.3.2	改写闪存的总线时序设定	460
27.3.3	软件命令	464
27.3.4	模式的转移	464
27.3.5	命令的发行步骤	465
27.3.6	状态检查	470
27.4	标准串行输入 / 输出模式	471
27.5	并行输入 / 输出模式	474
27.6	有关改写闪存的注意事项	474
27.6.1	有关电源电压的注意事项	474
27.6.2	有关硬件复位的注意事项	474
27.6.3	有关闪存保护的注意事项	474
27.6.4	编程时的注意事项	474
27.6.5	使用中断时的注意事项	474
27.6.6	有关改写控制程序的改写注意事项	475

27.6.7	编程 / 擦除次数和软件命令的执行时间 .....	475
27.6.8	其他注意事项 .....	475
28.	电特性 .....	476
29.	使用时的注意事项 .....	511
29.1	有关电路板设计的注意事项 .....	511
29.1.1	电源引脚 .....	511
29.1.2	电源电压 .....	511
29.2	设定寄存器时的注意事项 .....	511
29.2.1	含有只写位的寄存器 .....	511
29.3	使用时钟发生电路时的注意事项 .....	513
29.3.1	副时钟 .....	513
29.3.2	功率控制 .....	513
29.4	使用总线时的注意事项 .....	514
29.4.1	系统设计时的注意事项 .....	514
29.4.2	设定寄存器时的注意事项 .....	514
29.5	使用中断时的注意事项 .....	515
29.5.1	ISP 的设定 .....	515
29.5.2	NMI .....	515
29.5.3	外部中断 .....	515
29.6	使用 DMA 控制器时的注意事项 .....	516
29.6.1	DMAC 相关寄存器的设定 .....	516
29.6.2	读 DMAC 相关寄存器 .....	516
29.7	使用定时器时的注意事项 .....	516
29.7.1	定时器 A 和定时器 B 的共同事项 .....	516
29.7.2	定时器 A .....	516
29.7.3	定时器 B .....	518
29.8	使用三相马达控制定时器功能时的注意事项 .....	518
29.8.1	关机功能 .....	518
29.8.2	寄存器的设定 .....	518
29.9	使用串行接口时的注意事项 .....	519
29.9.1	UiBRG 寄存器 (i=0 ~ 8) 的变更 .....	519
29.9.2	时钟同步模式 .....	519
29.9.3	特殊模式 1 (I <sup>2</sup> C 模式) .....	519
29.10	使用 A/D 转换器时的注意事项 .....	520
29.10.1	设计电路板时的注意事项 .....	520
29.10.2	编程时的注意事项 .....	520
29.11	有关改写闪存的注意事项 .....	521
29.11.1	有关电源电压的注意事项 .....	521
29.11.2	有关硬件复位的注意事项 .....	521
29.11.3	有关闪存保护的注意事项 .....	521
29.11.4	编程时的注意事项 .....	521
29.11.5	使用中断时的注意事项 .....	522
29.11.6	有关改写控制程序的改写注意事项 .....	522
29.11.7	编程 / 擦除次数和软件命令的执行时间 .....	522
29.11.8	其他注意事项 .....	522
附录	.....	523
附录 1.	封装尺寸图 .....	523
索引	.....	524

## 1. 概要

### 1.1 特点

M16C 族是 32 位 /16 位 CISC 单片机，具有高 ROM 效率、低噪声特性、超低功耗、实际应用中的高处理能力以及各种内部外围功能等特点。在相同的体系结构上保持引脚排列的兼容以及外围功能的向上兼容，从高端到低端机种的系列展开来对应广泛的应用领域。

R32C/100 系列是 M16C 族的最高端产品，采用了 32 位 CISC 体系结构，具备 4G 字节的地址空间。另外，通过装载乘法器、乘加运算器和单精度浮点运算器，达到高指令效率和高处理能力。R32C/100 系列装载了串行接口、CRC 运算电路、DMAC、A/D 转换器、D/A 转换器、定时器、I<sup>2</sup>C 和看门狗定时器等各种外围功能。

R32C/117 群是 R32C/100 系列的标准产品，采用了 144 个引脚的 LQFP 封装和 100 个引脚的 LQFP 封装，内置了 9 个通道的串行接口、1 个通道的多主控 I<sup>2</sup>C 总线接口和 1 个通道的 CAN 模块。

#### 1.1.1 用途

汽车音响、音响、打印机、办公设备、工业设备等。



## 1.1.2 规格概要

R32C/117 群的规格概要如表 1.1 ~ 表 1.4 所示。

表 1.1 规格概要 (144 引脚版) (1/2)

分类	功能	说明
CPU	中央处理器	R32C/100 系列 CPU 内核 <ul style="list-style-type: none"> <li>基本指令数: 108 条</li> <li>指令的最短执行时间: 20ns (f(CPU)=50MHz)</li> <li>乘法器: 32 位 × 32 位 → 64 位</li> <li>乘加运算指令: 32 位 × 32 位 + 64 位 → 64 位</li> <li>FPU: 单精度 (符合 IEEE-754 标准)</li> <li>桶式移位器: 32 位</li> <li>运行模式: 单芯片模式、存储器扩展模式、微处理器模式 (选项 (注 1))</li> </ul>
存储器		闪存: 384K ~ 1M 字节 RAM: 40K/48K/63K 字节 数据闪存: 4K 字节 × 2 块 有关各机种的存储器容量, 请参照表 1.5。
电压检测	低电压检测电路	选项 (注 1) 低电压检测中断
时钟	时钟发生电路	<ul style="list-style-type: none"> <li>4 个电路 (主时钟、副时钟、PLL、内部振荡器)</li> <li>振荡停止检测: 主时钟振荡停止 / 再振荡的检测功能</li> <li>分频电路: 可选择 2 ~ 24 分频</li> <li>低功耗结构: 等待模式、停止模式</li> </ul>
外部总线扩展	总线 / 存储器扩展功能	<ul style="list-style-type: none"> <li>地址空间: 4G 字节 (其中, 用户最多能使用 64M 字节)</li> <li>外部总线接口: 可插入等待, 有 4 个片选输出。</li> <li>总线模式: 可选择分离总线 / 多路复用总线。</li> <li>可选择数据总线的宽度 (8 位 / 16 位 / 32 位)。</li> </ul>
中断		中断向量数: 261 个 外部中断输入: $\overline{\text{NMI}}$ 、 $\overline{\text{INT}} \times 9$ 个、键输入 × 4 个 中断优先级: 7 级
看门狗定时器		15 位 × 1 个 (带预分频器)
DMA	DMAC	4 个通道 <ul style="list-style-type: none"> <li>周期挪用方式</li> <li>启动源的个数: 57 个</li> <li>传送模式: 2 种 (单次传送、重复传送)</li> </ul>
	DMAC II	<ul style="list-style-type: none"> <li>能通过全部的外围功能中断源来启动。</li> <li>立即数传送功能、运算传送功能、链传送功能</li> </ul>
I/O 端口	可编程输入 / 输出端口	<ul style="list-style-type: none"> <li>输入专用: 2 个</li> <li>CMOS 输入 / 输出: 120 个</li> <li>其中 5V 容许端口: 32 个</li> <li>每 4 个引脚可设定上拉电阻 (5V 容许端口除外)。</li> </ul>

注 1. 如果要使用选项功能, 请向本公司的营业窗口询问。

表 1.2 规格概要 (144 引脚版) (2/2)

分类	功能	说明
定时器	定时器 A	16 位定时器 ×5 个 定时器模式、事件计数器模式、单触发定时器模式、脉宽调制 (PWM) 模式 事件计数器二相脉冲信号处理 (二相编码器输入) ×3 个
	定时器 B	16 位定时器 ×6 个 定时器模式、事件计数器模式、脉冲周期测量模式、脉宽测量模式
	三相马达控制定时器功能	三相马达控制定时器 ×1 个 (使用定时器 A1,A2,A4,B2) 内置死区时间定时器
串行接口	UART0 ~ UART8	时钟同步 / 异步兼用 ×9 个通道 <ul style="list-style-type: none"> <li>• I<sup>2</sup>C 总线 (UART0 ~ UART6)</li> <li>• 特殊模式 2 (UART0 ~ UART6)</li> <li>• IEBus (注 1) (UART0 ~ UART6) (选项 (注 2))</li> </ul>
A/D 转换器		10 位分辨率 ×34 个通道 有采样 & 保持
D/A 转换器		8 位分辨率 ×2 个电路
CRC 运算电路		CRC-CCITT ( $X^{16}+X^{12}+X^5+1$ )
X-Y 转换电路		16 位 ×16 位
智能 I/O		时间测量功能 (输入捕捉): 16 位 ×16 个 波形成功能 (输出比较): 16 位 ×24 个 通信功能: 可变速时钟同步串行 I/O、IEBus (注 1) (选项 (注 2))
多主控 I <sup>2</sup> C 总线接口		1 个通道
CAN 模块		1 个通道 符合 ISO11898-1 规格 32 个邮箱
闪存		编程 / 擦除电压: VCC=3.0 ~ 5.5V 编程 / 擦除次数: 1000 次 编程保护: ROM 码保护、ID 码保护 调试功能: on-chip 调试功能、板上闪存改写功能
工作频率 / 电源电压		50MHz/VCC=3.0 ~ 5.5V
工作环境温度		-20°C ~ 85°C (N 版) -40°C ~ 85°C (D 版) -40°C ~ 85°C (P 版)
消耗电流		35mA (VCC=5.0V, f(CPU)=50MHz) 8μA (VCC=3.3V, f(XCIN)=32.768kHz, 等待模式)
封装		144 个引脚塑模 LQFP (PLQP0144KA-A)

注 1. IEBus 是瑞萨电子株式会社的注册商标。

注 2. 如果要使用选项功能, 请向本公司的营业窗口询问。

表 1.3 规格概要 (100 引脚版) (1/2)

分类	功能	说明
CPU	中央处理器	R32C/100 系列 CPU 内核 <ul style="list-style-type: none"> <li>基本指令数: 108 条</li> <li>指令的最短执行时间: 20ns (f(CPU)=50MHz)</li> <li>乘法器: 32 位 × 32 位 → 64 位</li> <li>乘加运算指令: 32 位 × 32 位 + 64 位 → 64 位</li> <li>FPU: 单精度 (符合 IEEE-754 标准)</li> <li>桶式移位器: 32 位</li> <li>运行模式: 单芯片模式、存储器扩展模式、微处理器模式 (选项 (注 1))</li> </ul>
存储器		闪存: 384K ~ 1M 字节 RAM: 40K/48K/63K 字节 数据闪存: 4K 字节 × 2 块 有关各机种的存储器容量, 请参照表 1.5。
电压检测	低电压检测电路	选项 (注 1) 低电压检测中断
时钟	时钟发生电路	<ul style="list-style-type: none"> <li>4 个电路 (主时钟、副时钟、PLL、内部振荡器)</li> <li>振荡停止检测: 主时钟振荡停止 / 再振荡的检测功能</li> <li>分频电路: 可选择 2 ~ 24 分频</li> <li>低功耗结构: 等待模式、停止模式</li> </ul>
外部总线扩展	总线 / 存储器扩展功能	<ul style="list-style-type: none"> <li>地址空间: 4G 字节 (其中, 最多能使用 64M 字节)</li> <li>外部总线接口: 可插入等待, 有 4 个片选输出。</li> <li>总线模式: 可选择分离总线 / 多路复用总线, 可选择数据总线的宽度 (8 位 / 16 位)。</li> </ul>
中断		中断向量数: 261 个 外部中断输入: $\overline{\text{NMI}}$ 、 $\overline{\text{INT}} \times 6$ 个、键输入 × 4 个 中断优先级: 7 级
看门狗定时器		15 位 × 1 个 (带预分频器)
DMA	DMAC	4 个通道 <ul style="list-style-type: none"> <li>周期挪用方式</li> <li>启动源的个数: 51 个</li> <li>传送模式: 2 种 (单次传送、重复传送)</li> </ul>
	DMAC II	<ul style="list-style-type: none"> <li>能通过全部的外围功能中断源来启动。</li> <li>立即数传送功能、运算传送功能、链传送功能</li> </ul>
I/O 端口	可编程输入 / 输出端口	<ul style="list-style-type: none"> <li>输入专用: 2 个</li> <li>CMOS 输入 / 输出: 84 个</li> <li>其中 5V 容许端口: 32 个</li> <li>每 4 个引脚可设定上拉电阻 (5V 容许端口除外)。</li> </ul>

注 1. 如果要使用选项功能, 请向本公司的营业窗口询问。

表 1.4 规格概要（100 引脚版）（2/2）

分类	功能	说明
定时器	定时器 A	16 位定时器 ×5 个 定时器模式、事件计数器模式、单触发定时器模式、脉宽调制（PWM）模式 事件计数器二相脉冲信号处理（二相编码器输入）×3 个
	定时器 B	16 位定时器 ×6 个 定时器模式、事件计数器模式、脉冲周期测量模式、脉宽测量模式
	三相马达控制定时器功能	三相马达控制定时器 ×1 个（使用定时器 A1,A2,A4,B2） 内置死区时间定时器
串行接口	UART0 ~ UART8	时钟同步 / 异步兼用 ×9 个通道 <ul style="list-style-type: none"> <li>• I<sup>2</sup>C 总线（UART0 ~ UART6）</li> <li>• 特殊模式 2（UART0 ~ UART6）</li> <li>• IEBus（注 1）（UART0 ~ UART6）（选项（注 2））</li> </ul>
A/D 转换器		10 位分辨率 ×26 个通道 有采样 & 保持
D/A 转换器		8 位分辨率 ×2 个电路
CRC 运算电路		CRC-CCITT（ $X^{16}+X^{12}+X^5+1$ ）
X-Y 转换电路		16 位 ×16 位
智能 I/O		时间测量功能（输入捕捉）：16 位 ×16 个 波形成功能（输出比较）：16 位 ×19 个 通信功能：可变速时钟同步串行 I/O、IEBus（注 1）（选项（注 2））
多主控 I <sup>2</sup> C 总线接口		1 个通道
CAN 模块		1 个通道 符合 ISO11898-1 规格 32 个邮箱
闪存		编程 / 擦除电压：VCC=3.0 ~ 5.5V 编程 / 擦除次数：1000 次 编程保护：ROM 码保护、ID 码保护 调试功能：on-chip 调试功能、板上闪存改写功能
工作频率 / 电源电压		50MHz/VCC=3.0 ~ 5.5V
工作环境温度		-20°C ~ 85°C（N 版） -40°C ~ 85°C（D 版） -40°C ~ 85°C（P 版）
消耗电流		35mA（VCC=5.0V, f(CPU)=50MHz） 8μA（VCC=3.3V, f(XCIN)=32.768kHz, 等待模式）
封装		100 个引脚塑模 LQFP（PLQP0100KB-A）

注 1. IEBus 是瑞萨电子株式会社的注册商标。

注 2. 如果要使用选项功能，请向本公司的营业窗口询问。

## 1.2 产品一览表

产品一览表如表 1.5 所示，产品型号、存储器容量和封装如图 1.1 所示。

表 1.5 产品一览表

2009 年 11 月

产品型号	封装 (注 1)	ROM 容量 (注 2)	RAM 容量	备注	
R5F64175NFD (计)	PLQP0144KA-A	384K 字节 +8K 字节	40K 字节	-20°C ~ 85°C (N 版)	
R5F64175DFD				-40°C ~ 85°C (D 版)	
R5F64175PFD				-40°C ~ 85°C (P 版)	
R5F64175NFB (计)	PLQP0100KB-A			-20°C ~ 85°C (N 版)	
R5F64175DFB				-40°C ~ 85°C (D 版)	
R5F64175PFB				-40°C ~ 85°C (P 版)	
R5F64176NFD (计)	PLQP0144KA-A	512K 字节 +8K 字节		-20°C ~ 85°C (N 版)	
R5F64176DFD				-40°C ~ 85°C (D 版)	
R5F64176PFD				-40°C ~ 85°C (P 版)	
R5F64176NFB (计)	PLQP0100KB-A			-20°C ~ 85°C (N 版)	
R5F64176DFB				-40°C ~ 85°C (D 版)	
R5F64176PFB				-40°C ~ 85°C (P 版)	
R5F64177NFD (计)	PLQP0144KA-A	640K 字节 +8K 字节	48K 字节	-20°C ~ 85°C (N 版)	
R5F64177DFD				-40°C ~ 85°C (D 版)	
R5F64177PFD				-40°C ~ 85°C (P 版)	
R5F64177NFB (计)	PLQP0100KB-A			-20°C ~ 85°C (N 版)	
R5F64177DFB				-40°C ~ 85°C (D 版)	
R5F64177PFB				-40°C ~ 85°C (P 版)	
R5F64178NFD (计)	PLQP0144KA-A	768K 字节 +8K 字节		63K 字节	-20°C ~ 85°C (N 版)
R5F64178DFD					-40°C ~ 85°C (D 版)
R5F64178PFD					-40°C ~ 85°C (P 版)
R5F64178NFB (计)	PLQP0100KB-A				-20°C ~ 85°C (N 版)
R5F64178DFB					-40°C ~ 85°C (D 版)
R5F64178PFB					-40°C ~ 85°C (P 版)
R5F64179NFD (计)	PLQP0144KA-A	1M 字节 +8K 字节	-20°C ~ 85°C (N 版)		
R5F64179DFD			-40°C ~ 85°C (D 版)		
R5F64179PFD			-40°C ~ 85°C (P 版)		
R5F64179NFB (计)	PLQP0100KB-A		-20°C ~ 85°C (N 版)		
R5F64179DFB			-40°C ~ 85°C (D 版)		
R5F64179PFB			-40°C ~ 85°C (P 版)		

(开)：开发中 (计)：计划中

注 1. 旧封装代码如下：

PLQP0100KB-A: 100P6Q-A, PLQP0144KA-A: 144P6Q-A

注 2. ROM 容量的“+8K 字节”是数据闪存的容量。

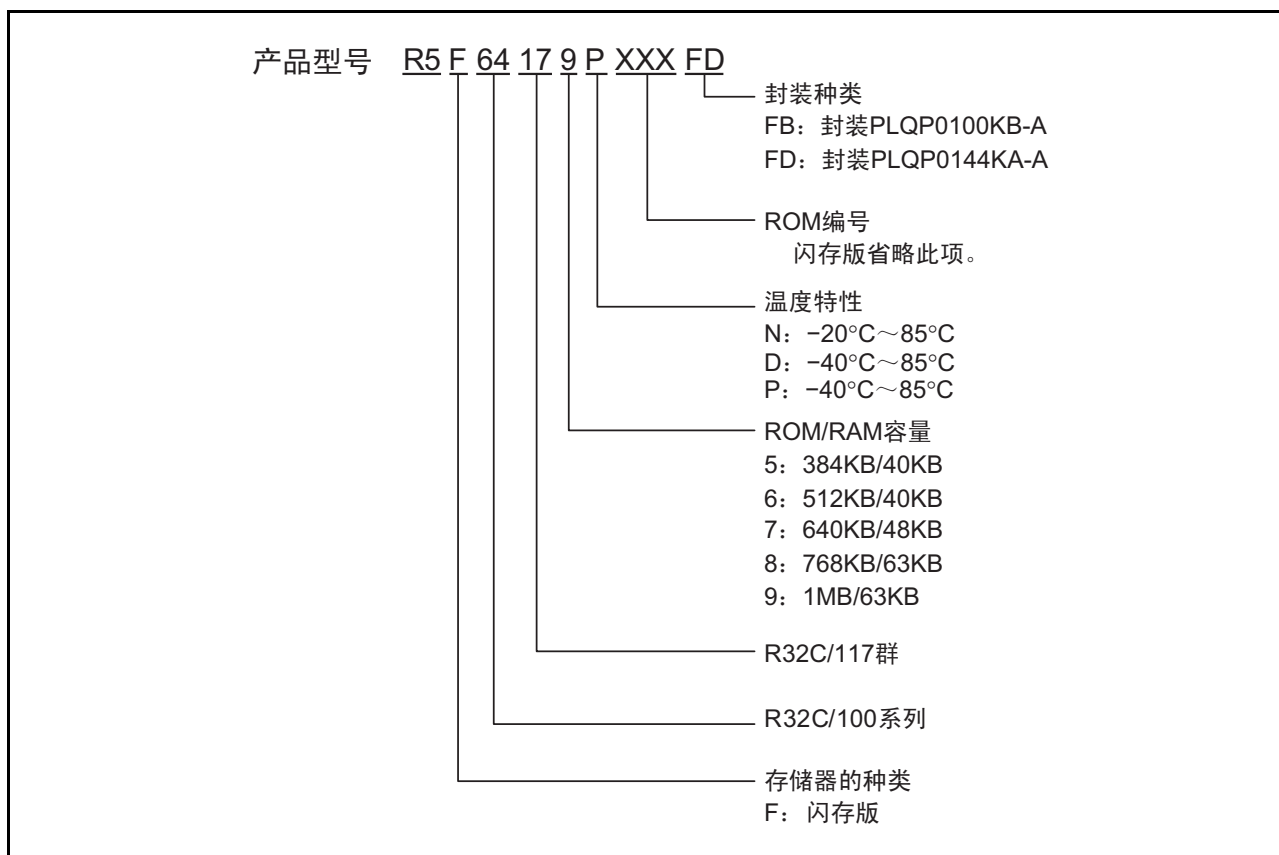


图 1.1 产品型号、存储器容量和封装

1.3 框图

R32C/117 群的框图如图 1.2 所示。

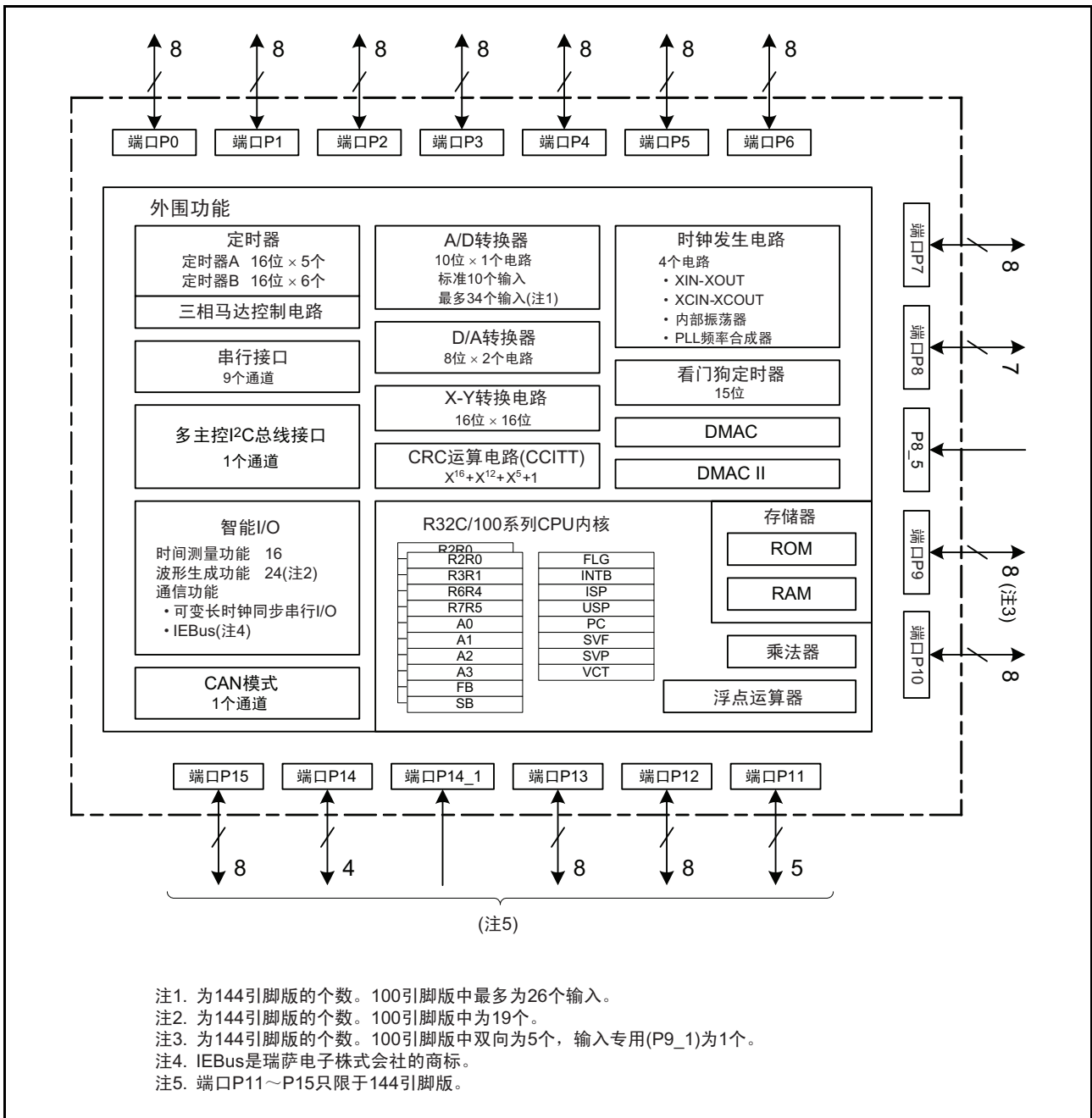


图 1.2 R32C/117 群的框图

### 1.4 引脚连接图

引脚连接图（俯视图）如图 1.3、图 1.4 所示。

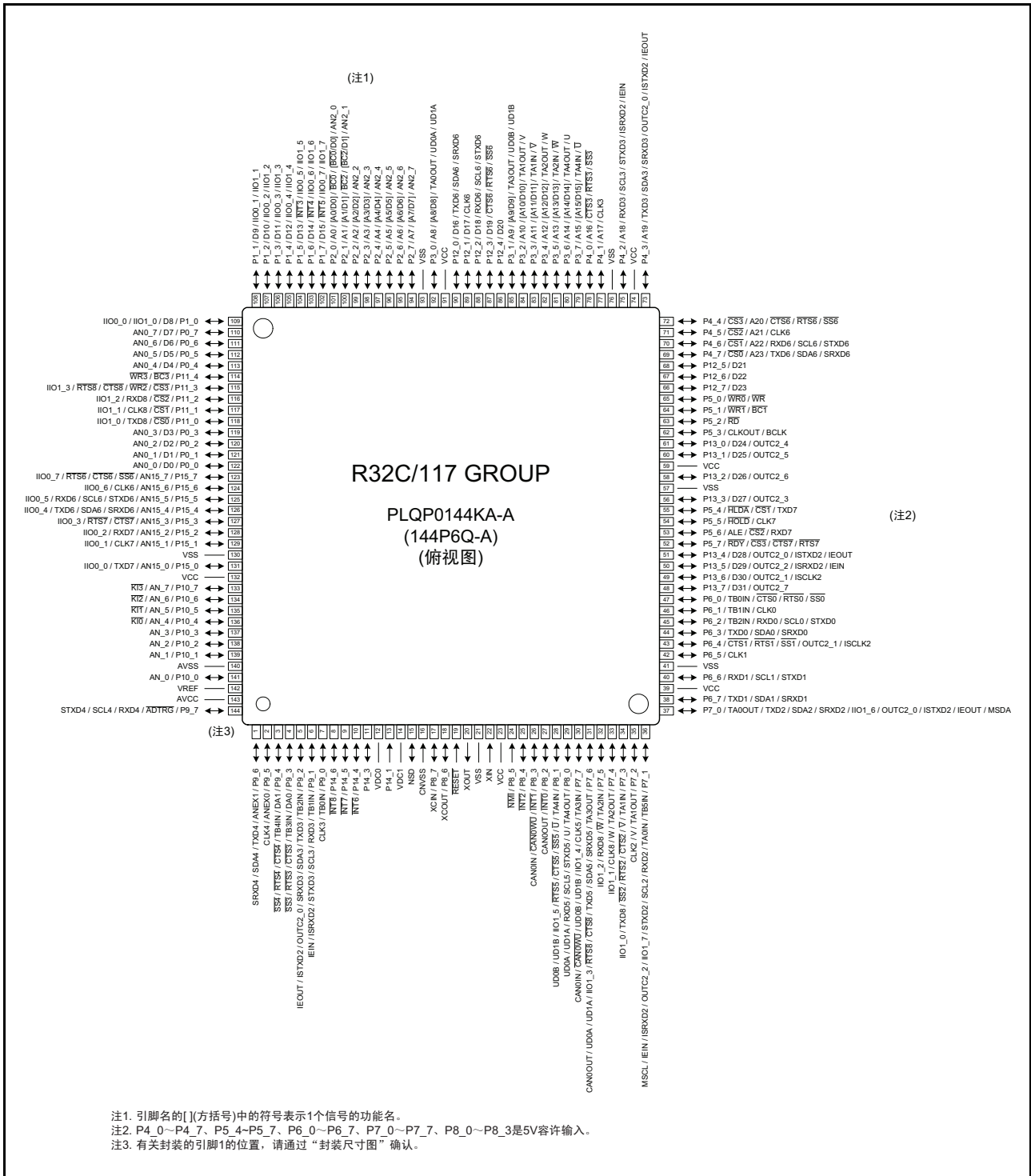


图 1.3 144 引脚版的引脚连接图（俯视图）



表 1.6 144 引脚版的引脚名一览表 (1/4)

Pin No.	控制引脚	端口	中断引脚	定时器引脚	UART/CAN 引脚	智能 I/O 引脚	模拟引脚	总线控制引脚
1		P9_6			TXD4/SDA4/SRXD4		ANEX1	
2		P9_5			CLK4		ANEX0	
3		P9_4		TB4IN	CTS4/RTS4/SS4		DA1	
4		P9_3		TB3IN	CTS3/RTS3/SS3		DA0	
5		P9_2		TB2IN	TXD3/SDA3/SRXD3	OUTC2_0/ISTXD2/ IEOUT		
6		P9_1		TB1IN	RXD3/SCL3/STXD3	ISRXD2/IEIN		
7		P9_0		TB0IN	CLK3			
8		P14_6	$\overline{\text{INT8}}$					
9		P14_5	$\overline{\text{INT7}}$					
10		P14_4	$\overline{\text{INT6}}$					
11		P14_3						
12	VDC0							
13		P14_1						
14	VDC1							
15	NSD							
16	CNVSS							
17	XCIN	P8_7						
18	XCOU	P8_6						
19	$\overline{\text{RESET}}$							
20	XOUT							
21	VSS							
22	XIN							
23	VCC							
24		P8_5	$\overline{\text{NMI}}$					
25		P8_4	$\overline{\text{INT2}}$					
26		P8_3	$\overline{\text{INT1}}$		CAN0IN/CAN0WU			
27		P8_2	$\overline{\text{INT0}}$		CAN0OUT			
28		P8_1		TA4IN/ $\overline{\text{U}}$	CTS5/RTS5/SS5	IIO1_5/UD0B/UD1B		
29		P8_0		TA4OUT/U	RXD5/SCL5/STXD5	UD0A/UD1A		
30		P7_7		TA3IN	CLK5/CAN0IN/CAN0WU	IIO1_4/UD0B/UD1B		
31		P7_6		TA3OUT	TXD5/SDA5/SRXD5/CTS8/ RTS8/CAN0OUT	IIO1_3/UD0A/UD1A		
32		P7_5		TA2IN/ $\overline{\text{W}}$	RXD8	IIO1_2		
33		P7_4		TA2OUT/W	CLK8	IIO1_1		
34		P7_3		TA1IN/ $\overline{\text{V}}$	CTS2/RTS2/SS2/TXD8	IIO1_0		
35		P7_2		TA1OUT/V	CLK2			
36		P7_1		TB5IN/ TA0IN	RXD2/SCL2/STXD2/MSCL	IIO1_7/OUTC2_2/ ISRXD2/IEIN		

表 1.7 144 引脚版的引脚名一览表 (2/4)

Pin No.	控制引脚	端口	中断引脚	定时器引脚	UART/CAN 引脚	智能 I/O 引脚	模拟引脚	总线控制引脚
37		P7_0		TA0OUT	TXD2/SDA2/SRXD2/ MSDA	IIO1_6/OUTC2_0/ ISTXD2/IEOUT		
38		P6_7			TXD1/SDA1/SRXD1			
39	VCC							
40		P6_6			RXD1/SCL1/STXD1			
41	VSS							
42		P6_5			CLK1			
43		P6_4			CTS1/RTS1/SS1	OUTC2_1/ISCLK2		
44		P6_3			TXD0/SDA0/SRXD0			
45		P6_2		TB2IN	RXD0/SCL0/STXD0			
46		P6_1		TB1IN	CLK0			
47		P6_0		TB0IN	CTS0/RTS0/SS0			
48		P13_7				OUTC2_7		D31
49		P13_6				OUTC2_1/ISCLK2		D30
50		P13_5				OUTC2_2/ISRXD2/IEIN		D29
51		P13_4				OUTC2_0/ISTXD2/IEOUT		D28
52		P5_7			CTS7/RTS7			RDY/CS3
53		P5_6			RXD7			ALE/CS2
54		P5_5			CLK7			HOLD
55		P5_4			TXD7			HLD/CS1
56		P13_3				OUTC2_3		D27
57	VSS							
58		P13_2				OUTC2_6		D26
59	VCC							
60		P13_1				OUTC2_5		D25
61		P13_0				OUTC2_4		D24
62		P5_3						CLKOUT /BCLK
63		P5_2						RD
64		P5_1						WR1/ BC1
65		P5_0						WR0/ WR
66		P12_7						D23
67		P12_6						D22
68		P12_5						D21
69		P4_7			TXD6/SDA6/SRXD6			CS0/A23
70		P4_6			TXD6/SCL6/STXD6			CS1/A22
71		P4_5			CLK6			CS2/A21
72		P4_4			CTS6/TRS6/SS6			CS3/A20

表 1.8 144 引脚版的引脚名一览表 (3/4)

Pin No.	控制引脚	端口	中断引脚	定时器引脚	UART/CAN 引脚	智能 I/O 引脚	模拟引脚	总线控制引脚
73		P4_3			TXD3/SDA3/SRXD3	OUTC2_0/ISTXD2/IEOUT		A19
74	VCC							
75		P4_2			RXD3/SCL3/STXD3	ISRXD2/IEIN		A18
76	VSS							
77		P4_1			CLK3			A17
78		P4_0			CTS3/RTS3/SS3			A16
79		P3_7		TA4IN $\bar{U}$				A15(/D15)
80		P3_6		TA4OUT/U				A14(/D14)
81		P3_5		TA2IN $\bar{W}$				A13(/D13)
82		P3_4		TA2OUT/W				A12(/D12)
83		P3_3		TA1IN $\bar{V}$				A11(/D11)
84		P3_2		TA1OUT/V				A10(/D10)
85		P3_1		TA3OUT		UDB/UD1B		A9(/D9)
86		P12_4						D20
87		P12_3			CTS6/RTS6/SS6			D19
88		P12_2			RXD6/SCL6/STXD6			D18
89		P12_1			CLK6			D17
90		P12_0			TXD6/SDA6/SRXD6			D16
91	VCC							
92		P3_0		TA0OUT		UD0A/UD1A		A8(/D8)
93	VSS							
94		P2_7					AN2_7	A7(/D7)
95		P2_6					AN2_6	A6(/D6)
96		P2_5					AN2_5	A5(/D5)
97		P2_4					AN2_4	A4(/D4)
98		P2_3					AN2_3	A3(/D3)
99		P2_2					AN2_2	A2(/D2)
100		P2_1					AN2_1	A1(/D1)/BC2(/D1)
101		P2_0					AN2_0	A0(/D0)/BC0(/D0)
102		P1_7	$\bar{INT5}$			IIO0_7/IIO1_7		D15
103		P1_6	$\bar{INT4}$			IIO0_6/IIO1_6		D14
104		P1_5	$\bar{INT3}$			IIO0_5/IIO1_5		D13
105		P1_4				IIO0_4/IIO1_4		D12
106		P1_3				IIO0_3/IIO1_3		D11
107		P1_2				IIO0_2/IIO1_2		D10
108		P1_1				IIO0_1/IIO1_1		D9

表 1.9 144 引脚版的引脚名一览表 (4/4)

Pin No.	控制引脚	端口	中断引脚	定时器引脚	UART/CAN 引脚	智能 I/O 引脚	模拟引脚	总线控制引脚
109		P1_0				IIO0_0/IIO1_0		D8
110		P0_7					AN0_7	D7
111		P0_6					AN0_6	D6
112		P0_5					AN0_5	D5
113		P0_4					AN0_4	D4
114		P11_4						BC3/WR3
115		P11_3			CTS8/RTS8	IIO1_3		CS3/WR2
116		P11_2			RXD8	IIO1_2		CS2
117		P11_1			CLK8	IIO1_1		CS1
118		P11_0			TXD8	IIO1_0		CS0
119		P0_3					AN0_3	D3
120		P0_2					AN0_2	D2
121		P0_1					AN0_1	D1
122		P0_0					AN0_0	D0
123		P15_7			CTS6/TRS6/SS6	IIO0_7	AN15_7	
124		P15_6			CLK6	IIO0_6	AN15_6	
125		P15_5			RXD6/SCL6/STXD6	IIO0_5	AN15_5	
126		P15_4			TXD6/SDA6/SRXD6	IIO0_4	AN15_4	
127		P15_3			CTS7/RTS7	IIO0_3	AN15_3	
128		P15_2			RXD7	IIO0_2	AN15_2	
129		P15_1			CLK7	IIO0_1	AN15_1	
130	VSS							
131		P15_0			TXD7	IIO0_0	AN15_0	
132	VCC							
133		P10_7	$\overline{KI3}$				AN_7	
134		P10_6	$\overline{KI2}$				AN_6	
135		P10_5	$\overline{KI1}$				AN_5	
136		P10_4	$\overline{KI0}$				AN_4	
137		P10_3					AN_3	
138		P10_2					AN_2	
139		P10_1					AN_1	
140	AVSS							
141		P10_0					AN_0	
142	VREF							
143	AVCC							
144		P9_7			RXD4/SCL4/STXD4		ADTRG	

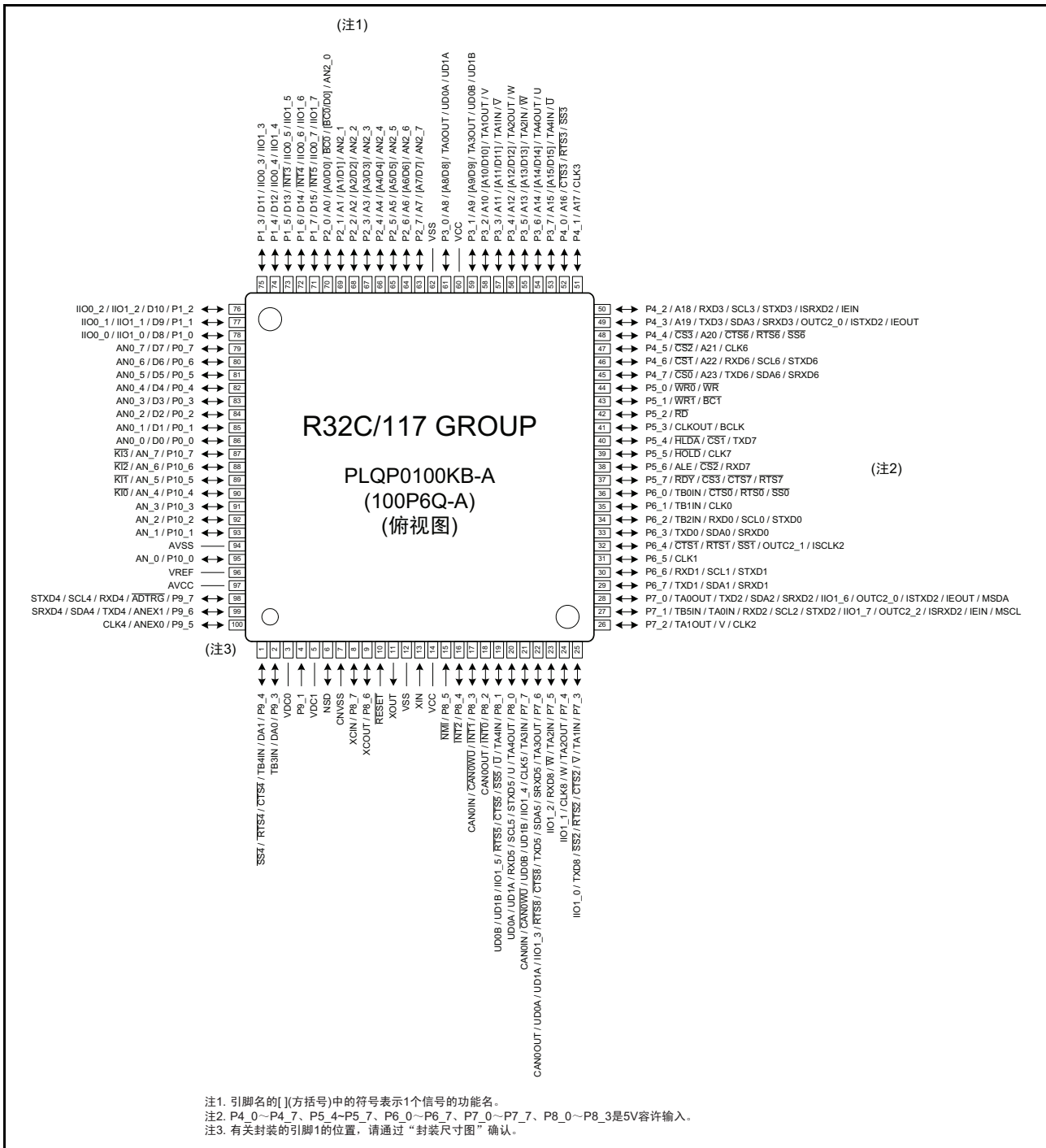


图 1.4 100 引脚版的引脚连接图 (俯视图)

表 1.10 100 引脚版的引脚名一览表 (1/3)

Pin No.	控制引脚	端口	中断引脚	定时器引脚	UART/CAN 引脚	智能 I/O 引脚	模拟引脚	总线控制引脚
1		P9_4		TB4IN	CTS4/RTS4/SS4		DA1	
2		P9_3		TB3IN			DA0	
3	VDC0							
4		P9_1						
5	VDC1							
6	NSD							
7	CNVSS							
8	XCIN	P8_7						
9	XCOUT	P8_6						
10	RESET							
11	XOUT							
12	VSS							
13	XIN							
14	VCC							
15		P8_5	NMI					
16		P8_4	INT2					
17		P8_3	INT1		CAN0IN/CAN0WU			
18		P8_2	INT0		CAN0OUT			
19		P8_1		TA4IN/U	CTS5/RTS5/SS5	IIO1_5/UD0B/UD1B		
20		P8_0		TA4OUT/U	RXD5/SCL5/STXD5	UD0A/UD1A		
21		P7_7		TA3IN	CLK5/CAN0IN/CAN0WU	IIO1_4/UD0B/UD1B		
22		P7_6		TA3OUT	TXD5/SDA5/SRXD5/CTS8/ RTS8/CAN0OUT	IIO1_3/UD0A/UD1A		
23		P7_5		TA2IN/W	RXD8	IIO1_2		
24		P7_4		TA2OUT/W	CLK8	IIO1_1		
25		P7_3		TA1IN/V	CTS2/RTS2/SS2/TXD8	IIO1_0		
26		P7_2		TA1OUT/V	CLK2			
27		P7_1		TB5IN/TA0IN	RXD2/SCL2/STXD2/MSCL	IIO1_7/OUTC2_2/ ISRXD2/IEIN		
28		P7_0		TA0OUT	TXD2/SDA2/SRXD2/MSDA	IIO1_6/OUTC2_0/ ISTXD2/IEOUT		
29		P6_7			TXD1/SDA1/SRXD1			
30		P6_6			RXD1/SCL1/STXD1			
31		P6_5			CLK1			
32		P6_4			CTS1/RTS1/SS1	OUTC2_1/ISCLK2		
33		P6_3			TXD0/SDA0/SRXD0			
34		P6_2		TB2IN	RXD0/SCL0/STXD0			
35		P6_1		TB1IN	CLK0			
36		P6_0		TB0IN	CTS0/RTS0/SS0			
37		P5_7			CTS7/RTS7			RDY/CS3

表 1.11 100 引脚版的引脚名一览表 (2/3)

Pin No.	控制引脚	端口	中断引脚	定时器引脚	UART/CAN 引脚	智能 I/O 引脚	模拟引脚	总线控制引脚
38		P5_6			RXD7			ALE/ $\overline{\text{CS}}2$
39		P5_5			CLK7			$\overline{\text{HOLD}}$
40		P5_4			TXD7			$\overline{\text{HLDA}}/\overline{\text{CS}}1$
41		P5_3						CLKOUT/BCLK
42		P5_2						$\overline{\text{RD}}$
43		P5_1						$\overline{\text{WR}}1/\overline{\text{BC}}1$
44		P5_0						$\overline{\text{WR}}0/\overline{\text{WR}}$
45		P4_7			TXD6/SDA6/SRXD6			$\overline{\text{CS}}0/\text{A}23$
46		P4_6			RXD6/SCL6/STXD6			$\overline{\text{CS}}1/\text{A}22$
47		P4_5			CLK6			$\overline{\text{CS}}2/\text{A}21$
48		P4_4			$\overline{\text{CTS}}6/\overline{\text{RTS}}6/\overline{\text{SS}}6$			$\overline{\text{CS}}3/\text{A}20$
49		P4_3			TXD3/SDA3/SRXD3	OUTC2_0/ISTXD2/IEOUT		A19
50		P4_2			RXD3/SCL3/STXD3	ISRXD2/IEIN		A18
51		P4_1			CLK3			A17
52		P4_0			$\overline{\text{CTS}}3/\overline{\text{RTS}}3/\overline{\text{SS}}3$			A16
53		P3_7		TA4IN/ $\overline{\text{U}}$				A15(/D15)
54		P3_6		TA4OUT/ $\overline{\text{U}}$				A14(/D14)
55		P3_5		TA2IN/ $\overline{\text{W}}$				A13(/D13)
56		P3_4		TA2OUT/ $\overline{\text{W}}$				A12(/D12)
57		P3_3		TA1IN/ $\overline{\text{V}}$				A11(/D11)
58		P3_2		TA1OUT/ $\overline{\text{V}}$				A10(/D10)
59		P3_1		TA3OUT		UD0B/UD1B		A9(/D9)
60	VCC							
61		P3_0		TA0OUT		UD0A/UD1A		A8(/D8)
62	VSS							
63		P2_7					AN2_7	A7(/D7)
64		P2_6					AN2_6	A6(/D6)
65		P2_5					AN2_5	A5(/D5)
66		P2_4					AN2_4	A4(/D4)
67		P2_3					AN2_3	A3(/D3)
68		P2_2					AN2_2	A2(/D2)
69		P2_1					AN2_1	A1(/D1)
70		P2_0					AN2_0	A0(/D0)/ $\overline{\text{BC}}0$ (/D0)
71		P1_7	$\overline{\text{INT}}5$			IIO0_7/IIO1_7		D15
72		P1_6	$\overline{\text{INT}}4$			IIO0_6/IIO1_6		D14
73		P1_5	$\overline{\text{INT}}3$			IIO0_5/IIO1_5		D13
74		P1_4				IIO0_4/IIO1_4		D12
75		P1_3				IIO0_3/IIO1_3		D11
76		P1_2				IIO0_2/IIO1_2		D10

表 1.12 100 引脚版的引脚名一览表 (3/3)

Pin No.	控制引脚	端口	中断引脚	定时器引脚	UART/CAN 引脚	智能 I/O 引脚	模拟引脚	总线控制引脚
77		P1_1				IIO0_1/IIO1_1		D9
78		P1_0				IIO0_0/IIO1_0		D8
79		P0_7					AN0_7	D7
80		P0_6					AN0_6	D6
81		P0_5					AN0_5	D5
82		P0_4					AN0_4	D4
83		P0_3					AN0_3	D3
84		P0_2					AN0_2	D2
85		P0_1					AN0_1	D1
86		P0_0					AN0_0	D0
87		P10_7	$\overline{KI3}$				AN_7	
88		P10_6	$\overline{KI2}$				AN_6	
89		P10_5	$\overline{KI1}$				AN_5	
90		P10_4	$\overline{KI0}$				AN_4	
91		P10_3					AN_3	
92		P10_2					AN_2	
93		P10_1					AN_1	
94	AVSS							
95		P10_0					AN_0	
96	VREF							
97	AVCC							
98		P9_7			RXD4/SCL4/STXD4		$\overline{ADTRG}$	
99		P9_6			TXD4/SDA4/SRXD4		ANEX1	
100		P9_5			CLK4		ANEX0	



## 1.5 引脚功能的说明

表 1.13 引脚功能的说明 (1/4)

分类	引脚名	输入 / 输出	功能
电源输入	VCC、VSS	输入	必须将 3.0 ~ 5.5V 输入到 VCC 引脚，将 VSS 引脚接地。
去耦电容器的连接引脚	VDC0、VDC1	—	必须在两个引脚之间连接用于稳定内部逻辑电压的去耦电容器。
模拟电源输入	AVCC、AVSS	输入	A/D 转换器的电源输入 必须将 AVCC 连接到 VCC，将 AVSS 连接到 VSS。
复位输入	$\overline{\text{RESET}}$	输入	如果将“L”电平输入到此引脚，单片机就进入复位状态。
CNVSS	CNVSS	输入	必须通过电阻下拉到 VSS。
调试端口	NSD	输入 / 输出	用于和调试器的通信。必须通过 1k ~ 4.7k $\Omega$ 的电阻上拉到 VCC。
主时钟输入	XIN	输入	主时钟振荡电路的输入 / 输出。必须在 XIN 和 XOUT 之间连接陶瓷谐振器或者晶体振荡器。如果输入外部生成的时钟，就必须从 XIN 输入时钟，并将 XOUT 置为开路。
主时钟输出	XOUT	输出	
副时钟输入	XCIN	输入	副时钟振荡电路的输入 / 输出。必须在 XCIN 和 XCOU 之间连接晶体振荡器。如果输入外部生成的时钟，就必须从 XCIN 输入时钟，并将 XCOU 置为开路。
副时钟输出	XCOU	输出	
BCLK 输出	BCLK	输出	输出总线时钟信号。
时钟输出	CLKOUT	输出	输出和 fC、f8 或者 f32 相同周期的时钟。
外部中断输入	$\overline{\text{INT0}} \sim \overline{\text{INT8}}$ (注 1)	输入	外部中断的输入
NMI 输入	$\overline{\text{P8\_5/NMI}}$	输入	NMI 的输入
键输入中断	$\overline{\text{KI0}} \sim \overline{\text{KI3}}$	输入	键输入中断的输入
总线控制引脚	D0 ~ D7	输入 / 输出	在存取了选择分离总线的区域时，进行数据 (D0 ~ D7) 的输入 / 输出。
	D8 ~ D15	输入 / 输出	在外部数据总线为 16 位或者 32 位并且存取了选择分离总线的区域时，进行数据 (D8 ~ D15) 的输入 / 输出。
	D16 ~ D31 (注 2)	输入 / 输出	在存取了选择 32 位外部数据总线的区域时，进行数据 (D16 ~ D31) 的输入 / 输出。
	A0 ~ A23	输出	输出地址 A0 ~ A23。

注 1.  $\overline{\text{INT6}} \sim \overline{\text{INT8}}$  只限于 144 引脚版。

注 2. D16 ~ D31 只限于 144 引脚版。

表 1.14 引脚功能的说明 (2/4)

分类	引脚名	输入 / 输出	功能
总线控制引脚	A0/D0 ~ A7/D7	输入 / 输出	在存取了选择多路复用总线的区域时，分时进行地址 (A0 ~ A7) 的输出和数据 (D0 ~ D7) 的输入 / 输出。
	A8/D8 ~ A15/D15	输入 / 输出	在外部数据总线为 16 位或者 32 位并且存取了选择多路复用总线的区域时，分时进行地址 (A8 ~ A15) 的输出和数据 (D8 ~ D15) 的输入 / 输出。
	$\overline{BC0}/D0$ 、 $\overline{BC2}/D1$ (注 1)	输入 / 输出	在存取了选择多路复用总线的区域时，分时进行字节控制 ( $\overline{BC0}$ 、 $\overline{BC2}$ ) 的输出和数据 (D0、D1) 的输入 / 输出。
	$\overline{CS0} \sim \overline{CS3}$	输出	输出片选信号。
	$\overline{WR0}/\overline{WR1}/\overline{WR2}/\overline{WR3}$ 、 $\overline{WR}/\overline{BC0}/\overline{BC1}/\overline{BC2}/\overline{BC3}$ 、 $\overline{RD}$ (注 1)	输出	输出写信号、字节控制信号和读信号。能通过程序选择是使用 $\overline{WRx}$ 还是使用 $\overline{WR}$ 、 $\overline{BCx}$ 。 <b>■选择 <math>\overline{WR0}</math>、<math>\overline{WR1}</math>、<math>\overline{WR2}</math>、<math>\overline{WR3}</math>、<math>\overline{RD}</math> 时</b> 在外部数据总线为 32 位的情况下，当 $\overline{WR0}$ 信号为“L”电平时写地址 $4n+0$ ；当 $\overline{WR1}$ 信号为“L”电平时写地址 $4n+1$ ；当 $\overline{WR2}$ 信号为“L”电平时写地址 $4n+2$ ；当 $\overline{WR3}$ 信号为“L”电平时写地址 $4n+3$ 。 在外部数据总线为 16 位的情况下，当 $\overline{WR0}$ 信号为“L”电平时写偶数地址；当 $\overline{WR1}$ 信号为“L”电平时写奇数地址。 当 $\overline{RD}$ 信号为“L”电平时进行读操作。 <b>■选择 <math>\overline{WR}</math>、<math>\overline{BC0}</math>、<math>\overline{BC1}</math>、<math>\overline{BC2}</math>、<math>\overline{BC3}</math>、<math>\overline{RD}</math> 时</b> 当 $\overline{WR}$ 信号为“L”电平时进行写操作；当 $\overline{RD}$ 信号为“L”电平时进行读操作。 在外部数据总线为 32 位的情况下，当 $\overline{BC0}$ 信号为“L”电平时存取地址 $4n+0$ ；当 $\overline{BC1}$ 信号为“L”电平时存取地址 $4n+1$ ；当 $\overline{BC2}$ 信号为“L”电平时存取地址 $4n+2$ ；当 $\overline{BC3}$ 信号为“L”电平时存取地址 $4n+3$ 。 在外部数据总线为 16 位的情况下，当 $\overline{BC0}$ 信号为“L”电平时存取偶数地址；当 $\overline{BC1}$ 信号为“L”电平时存取奇数地址。
	ALE	输出	在选择多路复用总线时，是用于锁存地址的信号。
	$\overline{HOLD}$	输入	在此引脚为“L”电平时，单片机处于保持状态。
	$\overline{HLDA}$	输出	在单片机为保持状态期间，输出“L”电平。
$\overline{RDY}$	输入	在 BCLK 下降时，如果将“L”电平输入到此引脚，CPU 就延长总线周期。	

注 1.  $\overline{BC2}/D1$ 、 $\overline{WR2}$ 、 $\overline{WR3}$ 、 $\overline{BC2}$ 、 $\overline{BC3}$  只限于 144 引脚版。

表 1.15 引脚功能的说明 (3/4)

分类	引脚名	输入 / 输出	功能
输入 / 输出端口 (注 1、注 2)	P0_0 ~ P0_7、 P1_0 ~ P1_7、 P2_0 ~ P2_7、 P3_0 ~ P3_7、 P4_0 ~ P4_7、 P5_0 ~ P5_7、 P6_0 ~ P6_7、 P7_0 ~ P7_7、 P8_0 ~ P8_4、 P8_6、P8_7、 P9_0 ~ P9_7、 P10_0 ~ P10_7、 P11_0 ~ P11_4、 P12_0 ~ P12_7、 P13_0 ~ P13_7、 P14_3 ~ P14_6、 P15_0 ~ P15_7	输入 / 输出	CMOS 的输入 / 输出端口 每个引脚能设定为输入端口或者输出端口。 部分端口为 5V 容许输入。另外，根据端口，可将上拉电阻、N 沟道漏极开路输出的设定置为有效。详细内容请参照“表 1.17 引脚的功能一览”。
输入端口 (注 2)	P9_1 (100 引脚版) P14_1 (144 引脚版)	输入	CMOS 的输入端口 能选择有无上拉电阻。 详细内容请参照“表 1.17 引脚的功能一览”。
定时器 A	TA0OUT ~ TA4OUT	输入 / 输出	定时器 A0 ~ A4 的输入 / 输出
	TA0IN ~ TA4IN	输入	定时器 A0 ~ A4 的输入
定时器 B	TB0IN ~ TB5IN	输入	定时器 B0 ~ B5 的输入
三相马达控制定 时器的输出	U、 $\bar{U}$ 、V、 $\bar{V}$ 、W、 $\bar{W}$	输出	三相马达控制定时器的输出
串行接口	$\overline{CTS0} \sim \overline{CTS8}$	输入	握手信号的输入
	$\overline{RTS0} \sim \overline{RTS8}$	输出	握手信号的输出
	CLK0 ~ CLK8	输入 / 输出	发送 / 接收时钟的输入 / 输出
	RXD0 ~ RXD8	输入	串行数据的输入
	TXD0 ~ TXD8	输出	串行数据的输出
简易型 I <sup>2</sup> C 总线	SDA0 ~ SDA6	输入 / 输出	串行数据的输入 / 输出
	SCL0 ~ SCL6	输入 / 输出	发送 / 接收时钟的输入 / 输出
串行接口特殊功 能	STXD0 ~ STXD6	输出	选择从属模式时的串行数据输出
	SRXD0 ~ SRXD6	输入	选择从属模式时的串行数据输入
	$\overline{SS0} \sim \overline{SS6}$	输入	串行接口特殊功能的控制输入

注 1. 100 引脚版的 P9\_1 是输入专用端口。

注 2. P9\_0、P9\_2、P11 ~ P15 只限于 144 引脚版。

表 1.16 引脚功能的说明 (4/4)

分类	引脚名	输入 / 输出	功能
A/D 转换器	AN_0 ~ AN_7、 AN0_0 ~ AN0_7、 AN2_0 ~ AN2_7、 AN15_0 ~ AN15_7 (注 1)	输入	A/D 转换器的模拟输入
	ADTRG	输入	A/D 转换器的外部触发输入
	ANEX0	输入 / 输出	A/D 转换器的扩展模拟输入和外部运算放大器连接模式中的输出
	ANEX1	输入	A/D 转换器的扩展模拟输入
D/A 转换器	DA0、DA1	输出	D/A 转换器的输出
基准电压输入	VREF	输入	A/D 转换器和 D/A 转换器的基准电压输入
智能 I/O	IIO0_0 ~ IIO0_7	输入 / 输出	智能 I/O 组 0 的输入 / 输出 能选择输入捕捉的输入或者输出比较的输出。
	IIO1_0 ~ IIO1_7	输入 / 输出	智能 I/O 组 1 的输入 / 输出 能选择输入捕捉的输入或者输出比较的输出。
	UD0A、UD0B、 UD1A、UD1B	输入	用于 2 相编码器的输入
	OUTC2_0 ~ OUTC2_7 (注 2)	输出	智能 I/O 组 2 的输出比较的输出
	ISCLK2	输入 / 输出	通信功能部的时钟输入 / 输出
	ISRXD2	输入	通信功能部的接收数据输入
	ISTXD2	输出	通信功能部的发送数据输出
	IEIN	输入	通信功能部的接收数据输入
	IEOUT	输出	通信功能部的发送数据输出
多主控 I <sup>2</sup> C 总线	MSDA	输入 / 输出	串行数据输入 / 输出
	MSCL	输入 / 输出	发送 / 接收时钟输入 / 输出
CAN 模式	CAN0IN	输入	CAN 通信功能的接收数据输入
	CAN0OUT	输出	CAN 通信功能的发送数据输出
	CAN0WU	输入	用于 CAN 唤醒的中断输入

注 1. AN15\_0 ~ AN15\_7 只限于 144 引脚版。

注 2. OUTC2\_3 ~ OUTC2\_7 只限于 144 引脚版。

表 1.17 引脚的功能一览

引脚名	封装		可选择的功能		5V 容许输入 (注 3)
	144 引脚版	100 引脚版	上拉电阻 (注 1)	N 沟道漏极开路输出 (注 2)	
P0_0 ~ P0_7	○	○	○		
P1_0 ~ P1_7	○	○	○		
P2_0 ~ P2_7	○	○	○		
P3_0 ~ P3_7	○	○	○		
P4_0 ~ P4_7	○	○		○	○
P5_0 ~ P5_3	○	○	○		
P5_4 ~ P5_7	○	○		○	○
P6_0 ~ P6_7	○	○		○	○
P7_0 ~ P7_7	○	○		○	○
P8_0 ~ P8_3	○	○		○	○
P8_4、P8_6、P8_7	○	○	○		
P9_0 ~ P9_3 (144 引脚版)	○		○	○	
P9_1、P9_3 (100 引脚版)		○	○		
P9_4 ~ P9_7	○	○	○	○	
P10_0 ~ P10_7	○	○	○		
P11_0 ~ P11_3	○		○	○	
P11_4	○		○		
P12_0 ~ P12_3	○		○	○	
P12_4 ~ P12_7	○		○		
P13_0 ~ P13_7	○		○		
P14_1	○		○		
P14_3 ~ P14_6	○		○		
P15_0 ~ P15_7	○		○	○	

注 1. 能以每 4 个引脚为单位选择有无上拉电阻。设定为输入端口的引脚有效。

注 2. 能给每个引脚设定 N 沟道漏极开路输出。

注 3. 5V 容许对设定为输入端口的引脚有效。使 5V 容许在输入 / 输出端口有效时，必须设定为 N 沟道漏极开路输出。

## 2. 中央处理器 (CPU)

CPU 的寄存器如图 2.1 所示。其中，R2R0、R3R1、R6R4、R7R5、A0、A1、A2、A3、SB 和 FB 共 10 个寄存器有 2 组。

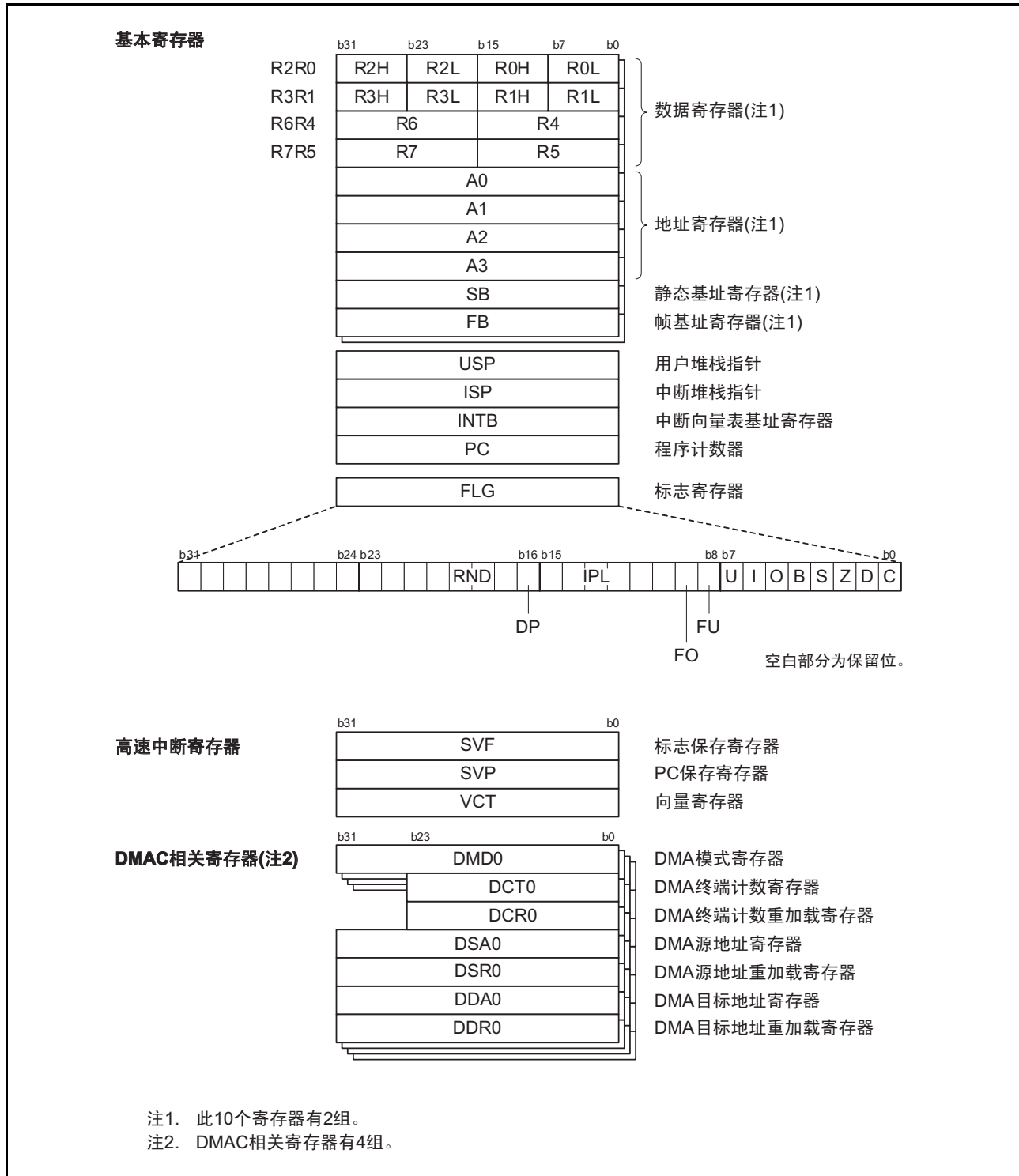


图 2.1 中央处理器的寄存器结构

## 2.1 基本寄存器

### 2.1.1 数据寄存器 (R2R0、R3R1、R6R4、R7R5)

R2R0、R3R1、R6R4、R7R5 由 32 位构成，主要用于传送、算术运算和逻辑运算。

能将 R2R0 的高位 (R2) 和低位 (R0) 分别用作 16 位数据寄存器。同样，也能将 R3R1、R6R4、R7R5 分别用作 2 个 16 位寄存器。

能将 R2R0 的高位 (R2H)、中高位 (R2L)、中低位 (R0H) 和低位 (R0L) 分别用作 8 位数据寄存器。同样，也能将 R3R1 分别用作 4 个 8 位寄存器。

### 2.1.2 地址寄存器 (A0、A1、A2、A3)

A0、A1、A2、A3 由 32 位构成，用于地址寄存器间接寻址和地址寄存器相对寻址，也能用于数据寄存器之间的传送、算术运算和逻辑运算。

### 2.1.3 静态基址寄存器 (SB)

SB 由 32 位构成，用于 SB 相对寻址。

### 2.1.4 帧基址寄存器 (FB)

FB 由 32 位构成，用于 FB 相对寻址。

### 2.1.5 程序计数器 (PC)

PC 由 32 位构成，表示下次执行的指令地址。

### 2.1.6 中断向量表基址寄存器 (INTB)

INTB 由 32 位构成，表示可变向量表的起始地址。

### 2.1.7 用户堆栈指针 (USP) 和中断堆栈指针 (ISP)

堆栈指针 (SP) 有 USP 和 ISP 两种，都由 32 位构成。

能通过 U 标志进行 USP 和 ISP 的转换。有关 U 标志，请参照“2.1.8 标志寄存器 (FLG)”。

能通过堆栈指针指定标志 (U 标志)，转换要使用的堆栈指针 (USP/ISP)。堆栈指针指定标志 (U 标志) 是标志寄存器 (FLG) 的 bit7。

必须给 USP 和 ISP 设定 4 的倍数。设定 4 的倍数能减少存储器的存取次数并且加快中断响应顺序的执行速度。

### 2.1.8 标志寄存器 (FLG)

FLG 由 32 位构成，表示 CPU 状态。

#### 2.1.8.1 进位标志 (C 标志)

保存由算术逻辑运算器产生的进位、借位和移出位等。

### 2.1.8.2 调试标志 (D 标志)

这是调试专用的标志，只能写“0”。

### 2.1.8.3 零标志 (Z 标志)

当运算结果是“0”时为“1”，否则为“0”。

### 2.1.8.4 符号标志 (S 标志)

当运算结果为负时为“1”，否则为“0”。

### 2.1.8.5 寄存器组指定标志 (B 标志)

选择寄存器组。当 B 标志为“0”时，指定寄存器组 0；当 B 标志为“1”时，指定寄存器组 1。

### 2.1.8.6 上溢标志 (O 标志)

当运算结果上溢时为“1”，否则为“0”。

### 2.1.8.7 中断允许标志 (I 标志)

这是允许可屏蔽中断的标志。当 I 标志为“0”时，禁止中断；当 I 标志为“1”时，允许中断。如果接受中断，I 标志就变为“0”。

### 2.1.8.8 堆栈指针指定标志 (U 标志)

当 U 标志为“0”时，指定中断堆栈指针 (ISP)；当 U 标志为“1”时，指定用户堆栈指针 (USP)。当接受硬件中断或者执行软件中断序号 0 ~ 127 的 INT 指令时，U 标志变为“0”。

### 2.1.8.9 浮点下溢标志 (FU 标志)

当浮点运算结果小于最小的规格化数 (下溢) 时为“1”，否则为“0”。  
在操作数的数据既不是规格化数也不是“0” (非法输入值) 的情况下为“1”。

### 2.1.8.10 浮点上溢标志 (FO 标志)

当浮点运算结果大于最大的规格化数 (上溢) 时为“1”，否则为“0”。  
在操作数的数据既不是规格化数也不是“0” (非法输入值) 的情况下为“1”。

### 2.1.8.11 处理器中断优先级 (IPL)

IPL 由 3 位构成，指定 0 级 ~ 7 级的 8 个级别的处理器中断优先级。如果产生请求的中断请求级高于处理器中断优先级 (IPL)，就允许该中断。

如果将处理器中断优先级 (IPL) 设定为 7 级 (111b)，就禁止全部的中断。



#### 2.1.8.12 定点数位置的指定位 (DP 位)

这是指定定点数的小数点位置的位，也是指定从定点数的乘法结果中取出哪个部分作为最终运算结果的位。使用 MULX 指令。

#### 2.1.8.13 浮点舍入运算模式 (RND)

浮点舍入运算模式 (RND) 由 2 位构成，指定舍入浮点运算结果的方式。

#### 2.1.8.14 保留位

只能写“0”，读取值为不定值。

### 2.2 高速中断寄存器

高速中断寄存器是高速执行中断响应顺序的专用寄存器。

高速中断寄存器有以下 3 个寄存器。详细内容请参照“11.4 高速中断”。

#### 2.2.1 标志保存寄存器 (SVF)

标志保存寄存器 (SVF) 由 32 位构成，用于发生高速中断时保存标志寄存器。

#### 2.2.2 PC 保存寄存器 (SVP)

PC 保存寄存器 (SVP) 由 32 位构成，用于发生高速中断时保存程序计数器。

#### 2.2.3 向量寄存器 (VCT)

向量寄存器 (VCT) 由 32 位构成，表示发生高速中断时的转移目标地址。

### 2.3 DMAC 相关寄存器

DMAC 相关寄存器有以下 7 种寄存器。详细内容请参照“13. DMAC”。

#### 2.3.1 DMA 模式寄存器 (DMD0、DMD1、DMD2、DMD3)

DMA 模式寄存器 (DMD0、DMD1、DMD2、DMD3) 由 32 位构成，是设定 DMA 的传送模式等的寄存器。

#### 2.3.2 DMA 终端计数寄存器 (DCT0、DCT1、DCT2、DCT3)

DMA 终端计数寄存器 (DCT0、DCT1、DCT2、DCT3) 由 24 位构成，是设定 DMA 的传送次数的寄存器。

### 2.3.3 DMA 终端计数重加载寄存器 (DCR0、DCR1、DCR2、DCR3)

DMA 终端计数重加载寄存器 (DCR0、DCR1、DCR2、DCR3) 由 24 位构成，是设定 DMA 终端计数寄存器的重加载值的寄存器。

### 2.3.4 DMA 源地址寄存器 (DSA0、DSA1、DSA2、DSA3)

DMA 源地址寄存器 (DSA0、DSA1、DSA2、DSA3) 由 32 位构成，是设定 DMA 的传送源地址的寄存器。

### 2.3.5 DMA 源地址重加载寄存器 (DSR0、DSR1、DSR2、DSR3)

DMA 源地址重加载寄存器 (DSR0、DSR1、DSR2、DSR3) 由 32 位构成，是设定 DMA 源地址寄存器的重加载值的寄存器。

### 2.3.6 DMA 目标地址寄存器 (DDA0、DDA1、DDA2、DDA3)

DMA 目标地址寄存器 (DDA0、DDA1、DDA2、DDA3) 由 32 位构成，是设定 DMA 的传送目标地址的寄存器。

### 2.3.7 DMA 目标地址重加载寄存器 (DDR0、DDR1、DDR2、DDR3)

DMA 目标地址重加载寄存器 (DDR0、DDR1、DDR2、DDR3) 由 32 位构成，是设定 DMA 目标地址寄存器的重加载值的寄存器。

### 3. 存储器

R32C/117 群的存储器分配图如图 3.1 所示。

地址空间为地址 00000000h ~地址 FFFFFFFFh 的 4G 字节。

内部 ROM 分配在从地址 FFFFFFFFh 向低位地址方向延伸的区域。例如，1M 字节的内部 ROM 分配在地址 FFF00000h ~地址 FFFFFFFFh。

固定中断向量分配在地址 FFFFFFFDCh ~地址 FFFFFFFFh，此区域保存中断程序的起始地址。

内部 RAM 分配在从地址 00000400h 向高位地址方向延伸的区域。例如，63K 字节的内部 RAM 分配在地址 00000400h ~地址 0000FFFFh。内部 RAM 除了保存数据以外，还用作子程序调用和中断时的堆栈。

SFR（Special Function Register）分配在地址 00000000h ~地址 000003FFh 和地址 00040000h ~地址 0004FFFFh，此区域分配了外围功能的控制寄存器。SFR 区域中未被分配的地址全部为保留区，所以不能存取。

在存储器扩展模式或者微处理器模式中，一部分区域为内部保留区，不能使用。

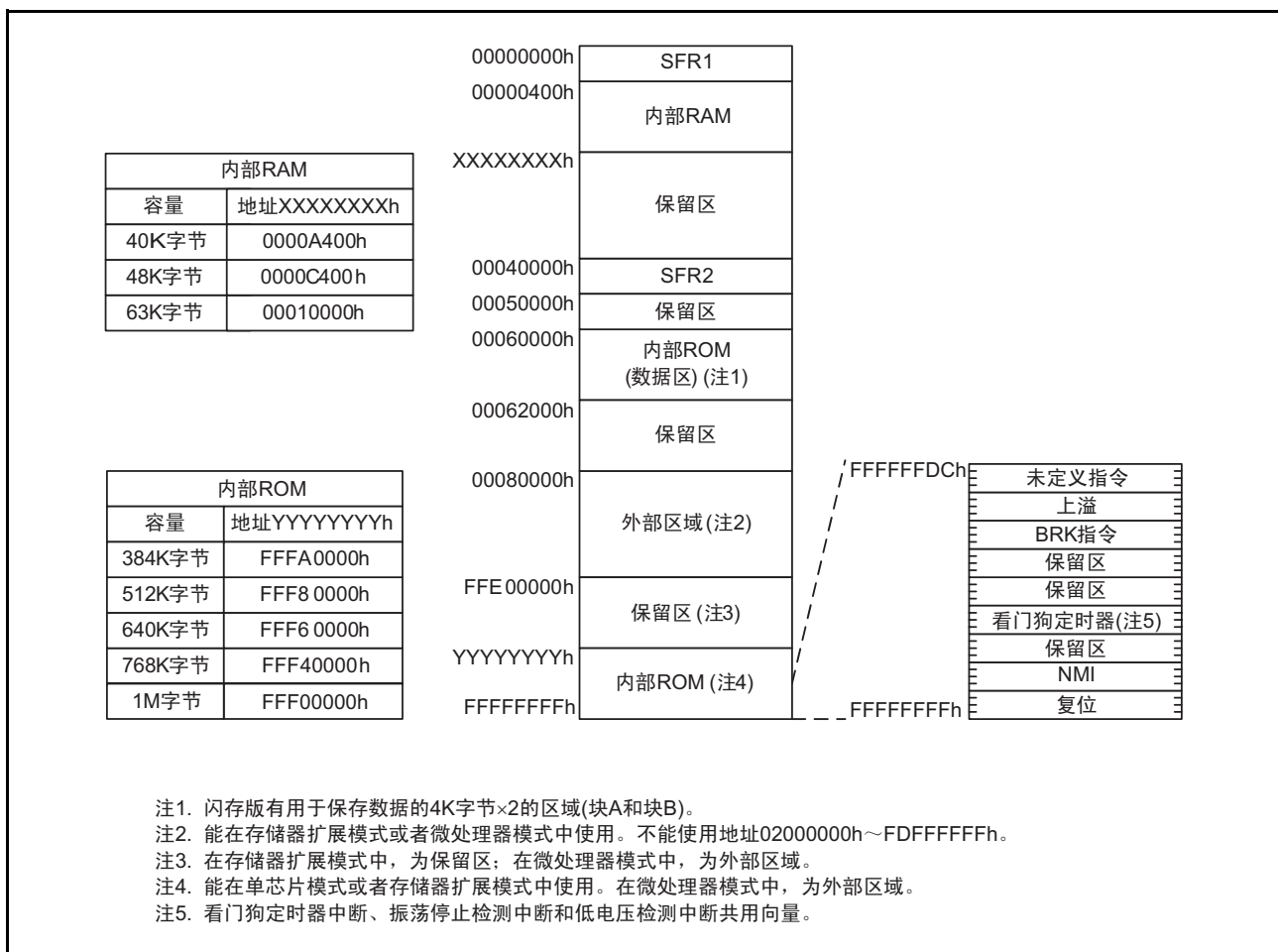


图 3.1 存储器分配图

## 4. SFR

SFR (Special Function Register) 是控制外围功能以及获取状态的寄存器。

SFR 一览表如表 4.1 SFR 一览表 (1) ~ 表 4.39 SFR 一览表 (39) 所示。

表 4.1 SFR 一览表 (1)

地址	寄存器	符号	复位后的值
000000h			
000001h			
000002h			
000003h			
000004h	时钟控制寄存器	CCR	0001 1000b
000005h			
000006h	闪存控制寄存器	FMCR	0000 0001b
000007h	保护解除寄存器	PRR	00h
000008h			
000009h			
00000Ah			
00000Bh			
00000Ch			
00000Dh			
00000Eh			
00000Fh			
000010h	外部总线控制寄存器 3/ 改写闪存的总线控制寄存器 3	EBC3/FEBC3	0000h
000011h			
000012h	片选 2-3 边界设定寄存器	CB23	00h
000013h			
000014h	外部总线控制寄存器 2	EBC2	0000h
000015h			
000016h	片选 1-2 边界设定寄存器	CB12	00h
000017h			
000018h	外部总线控制寄存器 1	EBC1	0000h
000019h			
00001Ah	片选 0-1 边界设定寄存器	CB01	00h
00001Bh			
00001Ch	外部总线控制寄存器 0/ 改写闪存的总线控制寄存器 0	EBC0/FEBC0	0000h
00001Dh			
00001Eh	外围总线控制寄存器	PBC	0504h
00001Fh			
000020h ~ 00005Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.2 SFR 一览表 (2)

地址	寄存器	符号	复位后的值
000060h			
000061h	定时器 B5 的中断控制寄存器	TB5IC	XXXX X000b
000062h	UART5 的发送 /NACK 中断控制寄存器	S5TIC	XXXX X000b
000063h	UART2 的接收 /ACK 中断控制寄存器 /I <sup>2</sup> C 总线的线中断控制寄存器	S2RIC/I2CLIC	XXXX X000b
000064h	UART6 的发送 /NACK 中断控制寄存器	S6TIC	XXXX X000b
000065h	UART3 的接收 /ACK 中断控制寄存器	S3RIC	XXXX X000b
000066h	UART5/6 总线冲突、开始条件 / 停止条件检测的中断控制寄存器	BCN5IC/BCN6IC	XXXX X000b
000067h	UART4 的接收 /ACK 中断控制寄存器	S4RIC	XXXX X000b
000068h	DMA0 传送结束的中断控制寄存器	DM0IC	XXXX X000b
000069h	UART0/3 总线冲突、开始条件 / 停止条件检测的中断控制寄存器	BCN0IC/BCN3IC	XXXX X000b
00006Ah	DMA2 传送结束的中断控制寄存器	DM2IC	XXXX X000b
00006Bh	A/D 转换器 0 转换结束的中断控制寄存器	AD0IC	XXXX X000b
00006Ch	定时器 A0 的中断控制寄存器	TA0IC	XXXX X000b
00006Dh	智能 I/O 中断控制寄存器 0	IIO0IC	XXXX X000b
00006Eh	定时器 A2 的中断控制寄存器	TA2IC	XXXX X000b
00006Fh	智能 I/O 中断控制寄存器 2	IIO2IC	XXXX X000b
000070h	定时器 A4 的中断控制寄存器	TA4IC	XXXX X000b
000071h	智能 I/O 中断控制寄存器 4	IIO4IC	XXXX X000b
000072h	UART0 的接收 /ACK 中断控制寄存器	S0RIC	XXXX X000b
000073h	智能 I/O 中断控制寄存器 6	IIO6IC	XXXX X000b
000074h	UART1 的接收 /ACK 中断控制寄存器	S1RIC	XXXX X000b
000075h	智能 I/O 中断控制寄存器 8	IIO8IC	XXXX X000b
000076h	定时器 B1 的中断控制寄存器	TB1IC	XXXX X000b
000077h	智能 I/O 中断控制寄存器 10	IIO10IC	XXXX X000b
000078h	定时器 B3 的中断控制寄存器	TB3IC	XXXX X000b
000079h			
00007Ah	INT5 中断控制寄存器	INT5IC	XX00 X000b
00007Bh	CAN0 唤醒中断控制寄存器	C0WIC	XXXX X000b
00007Ch	INT3 中断控制寄存器	INT3IC	XX00 X000b
00007Dh			
00007Eh	INT1 中断控制寄存器	INT1IC	XX00 X000b
00007Fh			
000080h			
000081h	UART2 的发送 /NACK 中断控制寄存器 /I <sup>2</sup> C 总线接口中断控制寄存器	S2TIC/I2CIC	XXXX X000b
000082h	UART5 的接收 /ACK 中断控制寄存器	S5RIC	XXXX X000b
000083h	UART3 的发送 /NACK 中断控制寄存器	S3TIC	XXXX X000b
000084h	UART6 的接收 /ACK 中断控制寄存器	S6RIC	XXXX X000b
000085h	UART4 的发送 /NACK 中断控制寄存器	S4TIC	XXXX X000b
000086h			
000087h	UART2 总线冲突、开始条件 / 停止条件检测的中断控制寄存器	BCN2IC	XXXX X000b

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.3 SFR 一览表 (3)

地址	寄存器	符号	复位后的值
000088h	DMA1 传送结束的中断控制寄存器	DM1IC	XXXX X000b
000089h	UART1/4 总线冲突、开始条件 / 停止条件检测的中断控制寄存器	BCN1IC/BCN4IC	XXXX X000b
00008Ah	DMA3 传送结束的中断控制寄存器	DM3IC	XXXX X000b
00008Bh	键输入中断控制寄存器	KUPIC	XXXX X000b
00008Ch	定时器 A1 的中断控制寄存器	TA1IC	XXXX X000b
00008Dh	智能 I/O 中断控制寄存器 1	IIO1IC	XXXX X000b
00008Eh	定时器 A3 的中断控制寄存器	TA3IC	XXXX X000b
00008Fh	智能 I/O 中断控制寄存器 3	IIO3IC	XXXX X000b
000090h	UART0 的发送 /NACK 中断控制寄存器	S0TIC	XXXX X000b
000091h	智能 I/O 中断控制寄存器 5	IIO5IC	XXXX X000b
000092h	UART1 的发送 /NACK 中断控制寄存器	S1TIC	XXXX X000b
000093h	智能 I/O 中断控制寄存器 7	IIO7IC	XXXX X000b
000094h	定时器 B0 的中断控制寄存器	TB0IC	XXXX X000b
000095h	智能 I/O 中断控制寄存器 9	IIO9IC	XXXX X000b
000096h	定时器 B2 的中断控制寄存器	TB2IC	XXXX X000b
000097h	智能 I/O 中断控制寄存器 11	IIO11IC	XXXX X000b
000098h	定时器 B4 的中断控制寄存器	TB4IC	XXXX X000b
000099h			
00009Ah	INT4 中断控制寄存器	INT4IC	XX00 X000b
00009Bh			
00009Ch	INT2 中断控制寄存器	INT2IC	XX00 X000b
00009Dh			
00009Eh	INT0 中断控制寄存器	INT0IC	XX00 X000b
00009Fh			
0000A0h	智能 I/O 中断请求寄存器 0	IIO0IR	0000 0XX1b
0000A1h	智能 I/O 中断请求寄存器 1	IIO1IR	0000 0XX1b
0000A2h	智能 I/O 中断请求寄存器 2	IIO2IR	0000 0X01b
0000A3h	智能 I/O 中断请求寄存器 3	IIO3IR	0000 XXX1b
0000A4h	智能 I/O 中断请求寄存器 4	IIO4IR	000X 0XX1b
0000A5h	智能 I/O 中断请求寄存器 5	IIO5IR	000X 0XX1b
0000A6h	智能 I/O 中断请求寄存器 6	IIO6IR	000X 0XX1b
0000A7h	智能 I/O 中断请求寄存器 7	IIO7IR	X00X 0XX1b
0000A8h	智能 I/O 中断请求寄存器 8	IIO8IR	XX0X 0XX1b
0000A9h	智能 I/O 中断请求寄存器 9	IIO9IR	0X00 0XX1b
0000AAh	智能 I/O 中断请求寄存器 10	IIO10IR	0X00 0XX1b
0000ABh	智能 I/O 中断请求寄存器 11	IIO11IR	0X00 0XX1b
0000ACh			
0000ADh			
0000AEh			
0000AFh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.4 SFR 一览表 (4)

地址	寄存器	符号	复位后的值
0000B0h	智能 I/O 中断允许寄存器 0	IIO0IE	00h
0000B1h	智能 I/O 中断允许寄存器 1	IIO1IE	00h
0000B2h	智能 I/O 中断允许寄存器 2	IIO2IE	00h
0000B3h	智能 I/O 中断允许寄存器 3	IIO3IE	00h
0000B4h	智能 I/O 中断允许寄存器 4	IIO4IE	00h
0000B5h	智能 I/O 中断允许寄存器 5	IIO5IE	00h
0000B6h	智能 I/O 中断允许寄存器 6	IIO6IE	00h
0000B7h	智能 I/O 中断允许寄存器 7	IIO7IE	00h
0000B8h	智能 I/O 中断允许寄存器 8	IIO8IE	00h
0000B9h	智能 I/O 中断允许寄存器 9	IIO9IE	00h
0000BAh	智能 I/O 中断允许寄存器 10	IIO10IE	00h
0000BBh	智能 I/O 中断允许寄存器 11	IIO11IE	00h
0000BCh			
0000BDh			
0000BEh			
0000BFh			
0000C0h			
0000C1h	CAN0 发送中断控制寄存器	C0TIC	XXXX X000b
0000C2h			
0000C3h	CAN0 错误中断控制寄存器	C0EIC	XXXX X000b
0000C4h			
0000C5h			
0000C6h			
0000C7h			
0000C8h			
0000C9h			
0000CAh			
0000CBh			
0000CCh			
0000CDh			
0000CEh			
0000CFh			
0000D0h	CAN0 发送 FIFO 中断控制寄存器	C0FTIC	XXXX X000b
0000D1h			
0000D2h			
0000D3h			
0000D4h			
0000D5h			
0000D6h			
0000D7h			
0000D8h			
0000D9h			
0000DAh			
0000DBh			
0000DCh			
0000DDh	UART7 发送中断控制寄存器	S7TIC	XXXX X000b
0000DEh	INT7 中断控制寄存器	INT7IC	XX00 X000b
0000DFh	UART8 发送中断控制寄存器	S8TIC	XXXX X000b

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.5 SFR 一览表 (5)

地址	寄存器	符号	复位后的值
0000E0h			
0000E1h	CAN0 接收中断控制寄存器	C0RIC	XXXX X000b
0000E2h			
0000E3h			
0000E4h			
0000E5h			
0000E6h			
0000E7h			
0000E8h			
0000E9h			
0000EAh			
0000EBh			
0000ECh			
0000EDh			
0000EEh			
0000EFh			
0000F0h	CAN0 接收 FIFO 中断控制寄存器	C0FRIC	XXXX X000b
0000F1h			
0000F2h			
0000F3h			
0000F4h			
0000F5h			
0000F6h			
0000F7h			
0000F8h			
0000F9h			
000FAh			
000FBh			
000FCh	INT8 中断控制寄存器	INT8IC	XX00 X000b
000FDh	UART7 接收中断控制寄存器	S7RIC	XXXX X000b
000FEh	INT6 中断控制寄存器	INT6IC	XX00 X000b
000FFh	UART8 接收中断控制寄存器	S8RIC	XXXX X000b
000100h	组 1 的时间测量 / 波形生成寄存器 0	G1TM0/G1PO0	XXXXh
000101h			
000102h	组 1 的时间测量 / 波形生成寄存器 1	G1TM1/G1PO1	XXXXh
000103h			
000104h	组 1 的时间测量 / 波形生成寄存器 2	G1TM2/G1PO2	XXXXh
000105h			
000106h	组 1 的时间测量 / 波形生成寄存器 3	G1TM3/G1PO3	XXXXh
000107h			

X: 不定值

空白部分全部为保留区，不能存取。



表 4.6 SFR 一览表 (6)

地址	寄存器	符号	复位后的值
000108h	组 1 的时间测量 / 波形生成寄存器 4	G1TM4/G1PO4	XXXXh
000109h			
00010Ah	组 1 的时间测量 / 波形生成寄存器 5	G1TM5/G1PO5	XXXXh
00010Bh			
00010Ch	组 1 的时间测量 / 波形生成寄存器 6	G1TM6/G1PO6	XXXXh
00010Dh			
00010Eh	组 1 的时间测量 / 波形生成寄存器 7	G1TM7/G1PO7	XXXXh
00010Fh			
000110h	组 1 的波形生成控制寄存器 0	G1POCR0	0000 X000b
000111h	组 1 的波形生成控制寄存器 1	G1POCR1	0X00 X000b
000112h	组 1 的波形生成控制寄存器 2	G1POCR2	0X00 X000b
000113h	组 1 的波形生成控制寄存器 3	G1POCR3	0X00 X000b
000114h	组 1 的波形生成控制寄存器 4	G1POCR4	0X00 X000b
000115h	组 1 的波形生成控制寄存器 5	G1POCR5	0X00 X000b
000116h	组 1 的波形生成控制寄存器 6	G1POCR6	0X00 X000b
000117h	组 1 的波形生成控制寄存器 7	G1POCR7	0X00 X000b
000118h	组 1 的时间测量控制寄存器 0	G1TMCR0	00h
000119h	组 1 的时间测量控制寄存器 1	G1TMCR1	00h
00011Ah	组 1 的时间测量控制寄存器 2	G1TMCR2	00h
00011Bh	组 1 的时间测量控制寄存器 3	G1TMCR3	00h
00011Ch	组 1 的时间测量控制寄存器 4	G1TMCR4	00h
00011Dh	组 1 的时间测量控制寄存器 5	G1TMCR5	00h
00011Eh	组 1 的时间测量控制寄存器 6	G1TMCR6	00h
00011Fh	组 1 的时间测量控制寄存器 7	G1TMCR7	00h
000120h	组 1 的基本定时器的寄存器	G1BT	XXXXh
000121h			
000122h	组 1 的基本定时器的控制寄存器 0	G1BCR0	00h
000123h	组 1 的基本定时器的控制寄存器 1	G1BCR1	0000 0000b
000124h	组 1 的时间测量预分频寄存器 6	G1TPR6	00h
000125h	组 1 的时间测量预分频寄存器 7	G1TPR7	00h
000126h	组 1 的功能允许寄存器	G1FE	00h
000127h	组 1 的功能选择寄存器	G1FS	00h
000128h			
000129h			
00012Ah			
00012Bh			
00012Ch			
00012Dh			
00012Eh			
00012Fh			
000130h ~ 00013Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.7 SFR 一览表 (7)

地址	寄存器	符号	复位后的值
000140h	组 2 的波形生成寄存器 0	G2PO0	XXXXh
000141h			
000142h	组 2 的波形生成寄存器 1	G2PO1	XXXXh
000143h			
000144h	组 2 的波形生成寄存器 2	G2PO2	XXXXh
000145h			
000146h	组 2 的波形生成寄存器 3	G2PO3	XXXXh
000147h			
000148h	组 2 的波形生成寄存器 4	G2PO4	XXXXh
000149h			
00014Ah	组 2 的波形生成寄存器 5	G2PO5	XXXXh
00014Bh			
00014Ch	组 2 的波形生成寄存器 6	G2PO6	XXXXh
00014Dh			
00014Eh	组 2 的波形生成寄存器 7	G2PO7	XXXXh
00014Fh			
000150h	组 2 的波形生成控制寄存器 0	G2POCR0	0000 0000b
000151h	组 2 的波形生成控制寄存器 1	G2POCR1	0000 0000b
000152h	组 2 的波形生成控制寄存器 2	G2POCR2	0000 0000b
000153h	组 2 的波形生成控制寄存器 3	G2POCR3	0000 0000b
000154h	组 2 的波形生成控制寄存器 4	G2POCR4	0000 0000b
000155h	组 2 的波形生成控制寄存器 5	G2POCR5	0000 0000b
000156h	组 2 的波形生成控制寄存器 6	G2POCR6	0000 0000b
000157h	组 2 的波形生成控制寄存器 7	G2POCR7	0000 0000b
000158h			
000159h			
00015Ah			
00015Bh			
00015Ch			
00015Dh			
00015Eh			
00015Fh			
000160h	组 2 的基本定时器的寄存器	G2BT	XXXXh
000161h			
000162h	组 2 的基本定时器的控制寄存器 0	G2BCR0	00h
000163h	组 2 的基本定时器的控制寄存器 1	G2BCR1	0000 0000b
000164h	基本定时器的开始寄存器	BTSR	XXXX 0000b
000165h			
000166h	组 2 的功能允许寄存器	G2FE	00h
000167h	组 2 的 RTP 输出缓冲寄存器	G2RTP	00h
000168h			
000169h			
00016Ah	组 2 的 SI/O 通信模式寄存器	G2MR	00XX X000b
00016Bh	组 2 的 SI/O 通信控制寄存器	G2CR	0000 X110b
00016Ch	组 2 的 SI/O 发送缓冲寄存器	G2TB	XXXXh
00016Dh			
00016Eh	组 2 的 SI/O 接收缓冲寄存器	G2RB	XXXXh
00016Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.8 SFR 一览表 (8)

地址	寄存器	符号	复位后的值
000170h	组 2 的 IE Bus 地址寄存器	IEAR	XXXXh
000171h			
000172h	组 2 的 IE Bus 控制寄存器	IECR	00XX X000b
000173h	组 2 的 IE Bus 发送中断源判断寄存器	IETIF	XXX0 0000b
000174h	组 2 的 IE Bus 接收中断源判断寄存器	IERIF	XXX0 0000b
000175h			
000176h			
000177h			
000178h			
000179h			
00017Ah			
00017Bh			
00017Ch			
00017Dh			
00017Eh			
00017Fh			
000180h	组 0 的时间测量 / 波形生成寄存器 0	G0TM0/G0PO0	XXXXh
000181h			
000182h	组 0 的时间测量 / 波形生成寄存器 1	G0TM1/G0PO1	XXXXh
000183h			
000184h	组 0 的时间测量 / 波形生成寄存器 2	G0TM2/G0PO2	XXXXh
000185h			
000186h	组 0 的时间测量 / 波形生成寄存器 3	G0TM3/G0PO3	XXXXh
000187h			
000188h	组 0 的时间测量 / 波形生成寄存器 4	G0TM4/G0PO4	XXXXh
000189h			
00018Ah	组 0 的时间测量 / 波形生成寄存器 5	G0TM5/G0PO5	XXXXh
00018Bh			
00018Ch	组 0 的时间测量 / 波形生成寄存器 6	G0TM6/G0PO6	XXXXh
00018Dh			
00018Eh	组 0 的时间测量 / 波形生成寄存器 7	G0TM7/G0PO7	XXXXh
00018Fh			
000190h	组 0 的波形生成控制寄存器 0	G0POCR0	0000 X000b
000191h	组 0 的波形生成控制寄存器 1	G0POCR1	0X00 X000b
000192h	组 0 的波形生成控制寄存器 2	G0POCR2	0X00 X000b
000193h	组 0 的波形生成控制寄存器 3	G0POCR3	0X00 X000b
000194h	组 0 的波形生成控制寄存器 4	G0POCR4	0X00 X000b
000195h	组 0 的波形生成控制寄存器 5	G0POCR5	0X00 X000b
000196h	组 0 的波形生成控制寄存器 6	G0POCR6	0X00 X000b
000197h	组 0 的波形生成控制寄存器 7	G0POCR7	0X00 X000b
000198h	组 0 的时间测量控制寄存器 0	G0TMCR0	00h
000199h	组 0 的时间测量控制寄存器 1	G0TMCR1	00h
00019Ah	组 0 的时间测量控制寄存器 2	G0TMCR2	00h
00019Bh	组 0 的时间测量控制寄存器 3	G0TMCR3	00h
00019Ch	组 0 的时间测量控制寄存器 4	G0TMCR4	00h
00019Dh	组 0 的时间测量控制寄存器 5	G0TMCR5	00h
00019Eh	组 0 的时间测量控制寄存器 6	G0TMCR6	00h
00019Fh	组 0 的时间测量控制寄存器 7	G0TMCR7	00h

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.9 SFR 一览表 (9)

地址	寄存器	符号	复位后的值
0001A0h	组 0 的基本定时器的寄存器	G0BT	XXXXh
0001A1h			
0001A2h	组 0 的基本定时器的控制寄存器 0	G0BCR0	00h
0001A3h	组 0 的基本定时器的控制寄存器 1	G0BCR1	0000 0000b
0001A4h	组 0 的时间测量预分频寄存器 6	G0TPR6	00h
0001A5h	组 0 的时间测量预分频寄存器 7	G0TPR7	00h
0001A6h	组 0 的功能允许寄存器	G0FE	00h
0001A7h	组 0 的功能选择寄存器	G0FS	00h
0001A8h			
0001A9h			
0001AAh			
0001ABh			
0001ACh			
0001ADh			
0001AEh			
0001AFh			
0001B0h			
0001B1h			
0001B2h			
0001B3h			
0001B4h			
0001B5h			
0001B6h			
0001B7h			
0001B8h			
0001B9h			
0001BAh			
0001BBh			
0001BCh			
0001BDh			
0001BEh			
0001BFh			
0001C0h			
0001C1h			
0001C2h			
0001C3h			
0001C4h	UART5 特殊模式寄存器 4	U5SMR4	00h
0001C5h	UART5 特殊模式寄存器 3	U5SMR3	00h
0001C6h	UART5 特殊模式寄存器 2	U5SMR2	00h
0001C7h	UART5 特殊模式寄存器	U5SMR	00h
0001C8h	UART5 发送 / 接收模式寄存器	U5MR	00h
0001C9h	UART5 位速率寄存器	U5BRG	XXh
0001CAh	UART5 发送缓冲寄存器	U5TB	XXXXh
0001CBh			
0001CCh	UART5 发送 / 接收控制寄存器 0	U5C0	0000 1000b
0001CDh	UART5 发送 / 接收控制寄存器 1	U5C1	0000 0010b
0001CEh	UART5 接收缓冲寄存器	U5RB	XXXXh
0001CFh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.10 SFR 一览表 (10)

地址	寄存器	符号	复位后的值
0001D0h			
0001D1h			
0001D2h			
0001D3h			
0001D4h	UART6 特殊模式寄存器 4	U6SMR4	00h
0001D5h	UART6 特殊模式寄存器 3	U6SMR3	00h
0001D6h	UART6 特殊模式寄存器 2	U6SMR2	00h
0001D7h	UART6 特殊模式寄存器	U6SMR	00h
0001D8h	UART6 发送 / 接收模式寄存器	U6MR	00h
0001D9h	UART6 位速率寄存器	U6BRG	XXh
0001DAh	UART6 发送缓冲寄存器	U6TB	XXXXh
0001DBh			
0001DCh	UART6 发送 / 接收控制寄存器 0	U6C0	0000 1000b
0001DDh	UART6 发送 / 接收控制寄存器 1	U6C1	0000 0010b
0001DEh	UART6 接收缓冲寄存器	U6RB	XXXXh
0001DFh			
0001E0h	UART7 发送 / 接收模式寄存器	U7MR	00h
0001E1h	UART7 位速率寄存器	U7BRG	XXh
0001E2h	UART7 发送缓冲寄存器	U7TB	XXXXh
0001E3h			
0001E4h	UART7 发送 / 接收控制寄存器 0	U7C0	00X0 1000b
0001E5h	UART7 发送 / 接收控制寄存器 1	U7C1	XXXX 0010b
0001E6h	UART7 接收缓冲寄存器	U7RB	XXXXh
0001E7h			
0001E8h	UART8 发送 / 接收模式寄存器	U8MR	00h
0001E9h	UART8 位速率寄存器	U8BRG	XXh
0001EAh	UART8 发送缓冲寄存器	U8TB	XXXXh
0001EBh			
0001ECh	UART8 发送 / 接收控制寄存器 0	U8C0	00X0 1000b
0001EDh	UART8 发送 / 接收控制寄存器 1	U8C1	XXXX 0010b
0001EEh	UART8 接收缓冲寄存器	U8RB	XXXXh
0001EFh			
0001F0h	UART7,8 发送 / 接收控制寄存器 2	U78CON	X000 0000b
0001F1h			
0001F2h			
0001F3h			
0001F4h			
0001F5h			
0001F6h			
0001F7h			
0001F8h			
0001F9h			
0001FAh			
0001FBh			
0001FCh			
0001FDh			
0001FEh			
0001FFh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.11 SFR 一览表 (11)

地址	寄存器	符号	复位后的值
000200h ~ 0002BFh			
0002C0h 0002C1h	X0 寄存器 /Y0 寄存器	X0R/Y0R	XXXXh
0002C2h 0002C3h	X1 寄存器 /Y1 寄存器	X1R/Y1R	XXXXh
0002C4h 0002C5h	X2 寄存器 /Y2 寄存器	X2R/Y2R	XXXXh
0002C6h 0002C7h	X3 寄存器 /Y3 寄存器	X3R/Y3R	XXXXh
0002C8h 0002C9h	X4 寄存器 /Y4 寄存器	X4R/Y4R	XXXXh
0002CAh 0002CBh	X5 寄存器 /Y5 寄存器	X5R/Y5R	XXXXh
0002CCh 0002CDh	X6 寄存器 /Y6 寄存器	X6R/Y6R	XXXXh
0002CEh 0002CFh	X7 寄存器 /Y7 寄存器	X7R/Y7R	XXXXh
0002D0h 0002D1h	X8 寄存器 /Y8 寄存器	X8R/Y8R	XXXXh
0002D2h 0002D3h	X9 寄存器 /Y9 寄存器	X9R/Y9R	XXXXh
0002D4h 0002D5h	X10 寄存器 /Y10 寄存器	X10R/Y10R	XXXXh
0002D6h 0002D7h	X11 寄存器 /Y11 寄存器	X11R/Y11R	XXXXh
0002D8h 0002D9h	X12 寄存器 /Y12 寄存器	X12R/Y12R	XXXXh
0002DAh 0002DBh	X13 寄存器 /Y13 寄存器	X13R/Y13R	XXXXh
0002DCh 0002DDh	X14 寄存器 /Y14 寄存器	X14R/Y14R	XXXXh
0002DEh 0002DFh	X15 寄存器 /Y15 寄存器	X15R/Y15R	XXXXh
0002E0h	XY 控制寄存器	XYC	XXXX XX00b
0002E1h			
0002E2h			
0002E3h			
0002E4h	UART1 特殊模式寄存器 4	U1SMR4	00h
0002E5h	UART1 特殊模式寄存器 3	U1SMR3	00h
0002E6h	UART1 特殊模式寄存器 2	U1SMR2	00h
0002E7h	UART1 特殊模式寄存器	U1SMR	00h
0002E8h	UART1 发送 / 接收模式寄存器	U1MR	00h
0002E9h	UART1 位速率寄存器	U1BRG	XXh
0002EAh 0002EBh	UART1 发送缓冲寄存器	U1TB	XXXXh
0002ECh	UART1 发送 / 接收控制寄存器 0	U1C0	0000 1000b
0002EDh	UART1 发送 / 接收控制寄存器 1	U1C1	0000 0010b
0002EEh 0002EFh	UART1 接收缓冲寄存器	U1RB	XXXXh

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.12 SFR 一览表 (12)

地址	寄存器	符号	复位后的值
0002F0h			
0002F1h			
0002F2h			
0002F3h			
0002F4h	UART4 特殊模式寄存器 4	U4SMR4	00h
0002F5h	UART4 特殊模式寄存器 3	U4SMR3	00h
0002F6h	UART4 特殊模式寄存器 2	U4SMR2	00h
0002F7h	UART4 特殊模式寄存器	U4SMR	00h
0002F8h	UART4 发送 / 接收模式寄存器	U4MR	00h
0002F9h	UART4 位速率寄存器	U4BRG	XXh
0002FAh	UART4 发送缓冲寄存器	U4TB	XXXXh
0002FBh			
0002FCh	UART4 发送 / 接收控制寄存器 0	U4C0	0000 1000b
0002FDh	UART4 发送 / 接收控制寄存器 1	U4C1	0000 0010b
0002FEh	UART4 接收缓冲寄存器	U4RB	XXXXh
0002FFh			
000300h	定时器 B3,B4,B5 计数开始标志	TBSR	000X XXXXb
000301h			
000302h	定时器 A1-1 寄存器	TA11	XXXXh
000303h			
000304h	定时器 A2-1 寄存器	TA21	XXXXh
000305h			
000306h	定时器 A4-1 寄存器	TA41	XXXXh
000307h			
000308h	三相 PWM 控制寄存器 0	INVC0	00h
000309h	三相 PWM 控制寄存器 1	INVC1	00h
00030Ah	三相输出缓冲寄存器 0	IDB0	XX11 1111b
00030Bh	三相输出缓冲寄存器 1	IDB1	XX11 1111b
00030Ch	死区时间定时器	DTT	XXh
00030Dh	定时器 B2 的中断发生频率设定计数器	ICTB2	XXh
00030Eh			
00030Fh			
000310h	定时器 B3 的寄存器	TB3	XXXXh
000311h			
000312h	定时器 B4 的寄存器	TB4	XXXXh
000313h			
000314h	定时器 B5 的寄存器	TB5	XXXXh
000315h			
000316h			
000317h			
000318h			
000319h			
00031Ah			
00031Bh	定时器 B3 的模式寄存器	TB3MR	00XX 0000b
00031Ch	定时器 B4 的模式寄存器	TB4MR	00XX 0000b
00031Dh	定时器 B5 的模式寄存器	TB5MR	00XX 0000b
00031Eh			
00031Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.13 SFR 一览表 (13)

地址	寄存器	符号	复位后的值
000320h			
000321h			
000322h			
000323h			
000324h	UART3 特殊模式寄存器 4	U3SMR4	00h
000325h	UART3 特殊模式寄存器 3	U3SMR3	00h
000326h	UART3 特殊模式寄存器 2	U3SMR2	00h
000327h	UART3 特殊模式寄存器	U3SMR	00h
000328h	UART3 发送 / 接收模式寄存器	U3MR	00h
000329h	UART3 位速率寄存器	U3BRG	XXh
00032Ah	UART3 发送缓冲寄存器	U3TB	XXXXh
00032Bh			
00032Ch	UART3 发送 / 接收控制寄存器 0	U3C0	0000 1000b
00032Dh	UART3 发送 / 接收控制寄存器 1	U3C1	0000 0010b
00032Eh	UART3 接收缓冲寄存器	U3RB	XXXXh
00032Fh			
000330h			
000331h			
000332h			
000333h			
000334h	UART2 特殊模式寄存器 4	U2SMR4	00h
000335h	UART2 特殊模式寄存器 3	U2SMR3	00h
000336h	UART2 特殊模式寄存器 2	U2SMR2	00h
000337h	UART2 特殊模式寄存器	U2SMR	00h
000338h	UART2 发送 / 接收模式寄存器	U2MR	00h
000339h	UART2 位速率寄存器	U2BRG	XXh
00033Ah	UART2 发送缓冲寄存器	U2TB	XXXXh
00033Bh			
00033Ch	UART2 发送 / 接收控制寄存器 0	U2C0	0000 1000b
00033Dh	UART2 发送 / 接收控制寄存器 1	U2C1	0000 0010b
00033Eh	UART2 接收缓冲寄存器	U2RB	XXXXh
00033Fh			
000340h	计数开始寄存器	TABSR	00h
000341h	时钟预分频器的复位寄存器	CPSRF	0XXX XXXXb
000342h	单触发开始寄存器	ONSF	00h
000343h	触发选择寄存器	TRGSR	00h
000344h	递增 / 递减选择寄存器	UDF	0000 0000b
000345h			
000346h	定时器 A0 的寄存器	TA0	XXXXh
000347h			
000348h	定时器 A1 的寄存器	TA1	XXXXh
000349h			
00034Ah	定时器 A2 的寄存器	TA2	XXXXh
00034Bh			
00034Ch	定时器 A3 的寄存器	TA3	XXXXh
00034Dh			
00034Eh	定时器 A4 的寄存器	TA4	XXXXh
00034Fh			

X: 不定值

空白部分全部为保留区, 不能存取。



表 4.14 SFR 一览表 (14)

地址	寄存器	符号	复位后的值
000350h	定时器 B0 的寄存器	TB0	XXXXh
000351h			
000352h	定时器 B1 的寄存器	TB1	XXXXh
000353h			
000354h	定时器 B2 的寄存器	TB2	XXXXh
000355h			
000356h	定时器 A0 的模式寄存器	TA0MR	0000 0000b
000357h	定时器 A1 的模式寄存器	TA1MR	0000 0000b
000358h	定时器 A2 的模式寄存器	TA2MR	0000 0000b
000359h	定时器 A3 的模式寄存器	TA3MR	0000 0000b
00035Ah	定时器 A4 的模式寄存器	TA4MR	0000 0000b
00035Bh	定时器 B0 的模式寄存器	TB0MR	00XX 0000b
00035Ch	定时器 B1 的模式寄存器	TB1MR	00XX 0000b
00035Dh	定时器 B2 的模式寄存器	TB2MR	00XX 0000b
00035Eh	定时器 B2 的特殊模式寄存器	TB2SC	XXXX XXX0b
00035Fh	计数源预分频寄存器	TCSPR	0000 0000b
000360h			
000361h			
000362h			
000363h			
000364h	UART0 特殊模式寄存器 4	U0SMR4	00h
000365h	UART0 特殊模式寄存器 3	U0SMR3	00h
000366h	UART0 特殊模式寄存器 2	U0SMR2	00h
000367h	UART0 特殊模式寄存器	U0SMR	00h
000368h	UART0 发送 / 接收模式寄存器	U0MR	00h
000369h	UART0 位速率寄存器	U0BRG	XXh
00036Ah	UART0 发送缓冲寄存器	U0TB	XXXXh
00036Bh			
00036Ch	UART0 发送 / 接收控制寄存器 0	U0C0	0000 1000b
00036Dh	UART0 发送 / 接收控制寄存器 1	U0C1	0000 0010b
00036Eh	UART0 接收缓冲寄存器	U0RB	XXXXh
00036Fh			
000370h			
000371h			
000372h			
000373h			
000374h			
000375h			
000376h			
000377h			
000378h			
000379h			
00037Ah			
00037Bh			
00037Ch	CRC 数据寄存器	CRCD	XXXXh
00037Dh			
00037Eh	CRC 输入寄存器	CRCIN	XXh
00037Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.15 SFR 一览表 (15)

地址	寄存器	符号	复位后的值
000380h	A/D0 寄存器 0	AD00	00XXh
000381h			
000382h	A/D0 寄存器 1	AD01	00XXh
000383h			
000384h	A/D0 寄存器 2	AD02	00XXh
000385h			
000386h	A/D0 寄存器 3	AD03	00XXh
000387h			
000388h	A/D0 寄存器 4	AD04	00XXh
000389h			
00038Ah	A/D0 寄存器 5	AD05	00XXh
00038Bh			
00038Ch	A/D0 寄存器 6	AD06	00XXh
00038Dh			
00038Eh	A/D0 寄存器 7	AD07	00XXh
00038Fh			
000390h			
000391h			
000392h	A/D0 控制寄存器 4	AD0CON4	XXXX 00XXb
000393h			
000394h	A/D0 控制寄存器 2	AD0CON2	X00X X000b
000395h	A/D0 控制寄存器 3	AD0CON3	XXXX X000b
000396h	A/D0 控制寄存器 0	AD0CON0	00h
000397h	A/D0 控制寄存器 1	AD0CON1	00h
000398h	D/A 寄存器 0	DA0	XXh
000399h			
00039Ah	D/A 寄存器 1	DA1	XXh
00039Bh			
00039Ch	D/A 控制寄存器	DACON	XXXX XX00b
00039Dh			
00039Eh			
00039Fh			
0003A0h			
0003A1h			
0003A2h			
0003A3h			
0003A4h			
0003A5h			
0003A6h			
0003A7h			
0003A8h			
0003A9h			
0003AAh			
0003ABh			
0003ACh			
0003ADh			
0003AEh			
0003AFh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.16 SFR 一览表 (16)

地址	寄存器	符号	复位后的值
0003B0h			
0003B1h			
0003B2h			
0003B3h			
0003B4h			
0003B5h			
0003B6h			
0003B7h			
0003B8h			
0003B9h			
0003BAh			
0003BBh			
0003BCh			
0003BDh			
0003BEh			
0003BFh			
0003C0h	端口 P0 寄存器	P0	XXh
0003C1h	端口 P1 寄存器	P1	XXh
0003C2h	端口 P0 方向寄存器	PD0	0000 0000b
0003C3h	端口 P1 方向寄存器	PD1	0000 0000b
0003C4h	端口 P2 寄存器	P2	XXh
0003C5h	端口 P3 寄存器	P3	XXh
0003C6h	端口 P2 方向寄存器	PD2	0000 0000b
0003C7h	端口 P3 方向寄存器	PD3	0000 0000b
0003C8h	端口 P4 寄存器	P4	XXh
0003C9h	端口 P5 寄存器	P5	XXh
0003CAh	端口 P4 方向寄存器	PD4	0000 0000b
0003CBh	端口 P5 方向寄存器	PD5	0000 0000b
0003CCh	端口 P6 寄存器	P6	XXh
0003CDh	端口 P7 寄存器	P7	XXh
0003CEh	端口 P6 方向寄存器	PD6	0000 0000b
0003CFh	端口 P7 方向寄存器	PD7	0000 0000b
0003D0h	端口 P8 寄存器	P8	XXh
0003D1h	端口 P9 寄存器	P9	XXh
0003D2h	端口 P8 方向寄存器	PD8	00X0 0000b
0003D3h	端口 P9 方向寄存器	PD9	0000 0000b
0003D4h	端口 P10 寄存器	P10	XXh
0003D5h	端口 P11 寄存器	P11	XXh
0003D6h	端口 P10 方向寄存器	PD10	0000 0000b
0003D7h	端口 P11 方向寄存器	PD11	XXX0 0000b
0003D8h	端口 P12 寄存器	P12	XXh
0003D9h	端口 P13 寄存器	P13	XXh
0003DAh	端口 P12 方向寄存器	PD12	0000 0000b
0003DBh	端口 P13 方向寄存器	PD13	0000 0000b
0003DCh	端口 P14 寄存器	P14	XXh
0003DDh	端口 P15 寄存器	P15	XXh
0003DEh	端口 P14 方向寄存器	PD14	X000 0000b
0003DFh	端口 P15 方向寄存器	PD15	0000 0000b

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.17 SFR 一览表 (17)

地址	寄存器	符号	复位后的值
0003E0h			
0003E1h			
0003E2h			
0003E3h			
0003E4h			
0003E5h			
0003E6h			
0003E7h			
0003E8h			
0003E9h			
0003EAh			
0003EBh			
0003ECh			
0003EDh			
0003EEh			
0003EFh			
0003F0h	上拉控制寄存器 0	PUR0	0000 0000b
0003F1h	上拉控制寄存器 1	PUR1	XXXX X0XXb
0003F2h	上拉控制寄存器 2	PUR2	000X XXXXb
0003F3h	上拉控制寄存器 3	PUR3	0000 0000b
0003F4h	上拉控制寄存器 4	PUR4	XXXX 0000b
0003F5h			
0003F6h			
0003F7h			
0003F8h			
0003F9h			
0003FAh			
0003FBh			
0003FCh			
0003FDh			
0003FEh			
0003FFh	端口控制寄存器	PCR	0XXX XXX0b

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.18 SFR 一览表 (18)

地址	寄存器	符号	复位后的值
04000h	闪存控制寄存器 0	FMR0	0X01 XX00b
040001h	闪存状态寄存器 0	FMSR0	1000 0000b
040002h			
040003h			
040004h			
040005h			
040006h			
040007h			
040008h	闪存寄存器的保护解除寄存器 0	FPR0	00h
040009h	闪存控制寄存器 1	FMR1	0000 0010b
04000Ah	块保护位监视寄存器 0	FBPM0	??X? ???b (注 1)
04000Bh	块保护位监视寄存器 1	FBPM1	XXX? ???b (注 1)
04000Ch			
04000Dh			
04000Eh			
04000Fh			
040010h			
040011h	块保护位监视寄存器 2	FBPM2	??? ????b (注 1)
040012h			
040013h			
040014h			
040015h			
040016h			
040017h			
040018h			
040019h			
04001Ah			
04001Bh			
04001Ch			
04001Dh			
04001Eh			
04001Fh			
040020h	PLL 控制寄存器 0	PLC0	0000 0001b
040021h	PLL 控制寄存器 1	PLC1	0001 1111b
040022h			
040023h			
040024h			
040025h			
040026h			
040027h			
040028h			
040029h			
04002Ah			
04002Bh			
04002Ch			
04002Dh			
04002Eh			
04002Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

注 1. 反映闪存的各块保护位的状态。

表 4.19 SFR 一览表 (19)

地址	寄存器	符号	复位后的值
040030h ~ 04003Fh			
040040h			
040041h			
040042h			
040043h			
040044h	处理器模式寄存器 0 (注 1)	PM0	1000 0000b (CNVSS 引脚为“L”电平) 0000 0011b (CNVSS 引脚为“H”电平)
040045h			
040046h	系统时钟控制寄存器 0	CM0	0000 1000b
040047h	系统时钟控制寄存器 1	CM1	0010 0000b
040048h	处理器模式寄存器 3	PM3	00h
040049h			
04004Ah	保护寄存器	PRCR	XXXX X000b
04004Bh			
04004Ch	保护寄存器 3	PRCR3	0000 0000b
04004Dh	振荡停止检测寄存器	CM2	00h
04004Eh			
04004Fh			
040050h			
040051h			
040052h			
040053h	处理器模式寄存器 2	PM2	00h
040054h	片选输出引脚设定寄存器 0	CSOP0	1000 XXXXb
040055h	片选输出引脚设定寄存器 1	CSOP1	01X0 XXXXb
040056h	片选输出引脚设定寄存器 2	CSOP2	XXXX 0000b
040057h			
040058h			
040059h			
04005Ah	低速模式时钟控制寄存器	CM3	XXXX XX00b
04005Bh			
04005Ch			
04005Dh			
04005Eh			
04005Fh			
040060h	电压稳压器的控制寄存器	VRRCR	0000 0000b
040061h			
040062h	低电压检测电路的控制寄存器	LVDC	0000 XX00b
040063h			
040064h	检测电压设定寄存器	DVCR	0000 XXXXb
040065h			
040066h			
040067h			
040068h ~ 040093h			

X: 不定值

空白部分全部为保留区, 不能存取。

注 1. 即使进行软件复位或者看门狗定时器复位, PM0 寄存器也保持复位前的值。

表 4.20 SFR 一览表 (20)

地址	寄存器	符号	复位后的值
040094h			
040095h			
040096h			
040097h	三相输出缓冲控制寄存器	IOBC	0XXX XXXXb
040098h	输入功能选择寄存器 0	IFS0	X000 0000b
040099h	输入功能选择寄存器 1	IFS1	XXXX X0X0b
04009Ah	输入功能选择寄存器 2	IFS2	0000 00X0b
04009Bh	输入功能选择寄存器 3	IFS3	XXXX XX00b
04009Ch			
04009Dh			
04009Eh			
04009Fh			
0400A0h	端口 P0_0 的功能选择寄存器	P0_0S	0XXX X000b
0400A1h	端口 P1_0 的功能选择寄存器	P1_0S	XXXX X000b
0400A2h	端口 P0_1 的功能选择寄存器	P0_1S	0XXX X000b
0400A3h	端口 P1_1 的功能选择寄存器	P1_1S	XXXX X000b
0400A4h	端口 P0_2 的功能选择寄存器	P0_2S	0XXX X000b
0400A5h	端口 P1_2 的功能选择寄存器	P1_2S	XXXX X000b
0400A6h	端口 P0_3 的功能选择寄存器	P0_3S	0XXX X000b
0400A7h	端口 P1_3 的功能选择寄存器	P1_3S	XXXX X000b
0400A8h	端口 P0_4 的功能选择寄存器	P0_4S	0XXX X000b
0400A9h	端口 P1_4 的功能选择寄存器	P1_4S	XXXX X000b
0400AAh	端口 P0_5 的功能选择寄存器	P0_5S	0XXX X000b
0400ABh	端口 P1_5 的功能选择寄存器	P1_5S	XXXX X000b
0400ACh	端口 P0_6 的功能选择寄存器	P0_6S	0XXX X000b
0400ADh	端口 P1_6 的功能选择寄存器	P1_6S	XXXX X000b
0400AEh	端口 P0_7 的功能选择寄存器	P0_7S	0XXX X000b
0400AFh	端口 P1_7 的功能选择寄存器	P1_7S	XXXX X000b
0400B0h	端口 P2_0 的功能选择寄存器	P2_0S	0XXX X000b
0400B1h	端口 P3_0 的功能选择寄存器	P3_0S	XXXX X000b
0400B2h	端口 P2_1 的功能选择寄存器	P2_1S	0XXX X000b
0400B3h	端口 P3_1 的功能选择寄存器	P3_1S	XXXX X000b
0400B4h	端口 P2_2 的功能选择寄存器	P2_2S	0XXX X000b
0400B5h	端口 P3_2 的功能选择寄存器	P3_2S	XXXX X000b
0400B6h	端口 P2_3 的功能选择寄存器	P2_3S	0XXX X000b
0400B7h	端口 P3_3 的功能选择寄存器	P3_3S	XXXX X000b
0400B8h	端口 P2_4 的功能选择寄存器	P2_4S	0XXX X000b
0400B9h	端口 P3_4 的功能选择寄存器	P3_4S	XXXX X000b
0400BAh	端口 P2_5 的功能选择寄存器	P2_5S	0XXX X000b
0400BBh	端口 P3_5 的功能选择寄存器	P3_5S	XXXX X000b
0400BCh	端口 P2_6 的功能选择寄存器	P2_6S	0XXX X000b
0400BDh	端口 P3_6 的功能选择寄存器	P3_6S	XXXX X000b
0400BEh	端口 P2_7 的功能选择寄存器	P2_7S	0XXX X000b
0400BFh	端口 P3_7 的功能选择寄存器	P3_7S	XXXX X000b

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.21 SFR 一览表 (21)

地址	寄存器	符号	复位后的值
0400C0h	端口 P4_0 的功能选择寄存器	P4_0S	X0XX X000b
0400C1h	端口 P5_0 的功能选择寄存器	P5_0S	XXXX X000b
0400C2h	端口 P4_1 的功能选择寄存器	P4_1S	X0XX X000b
0400C3h	端口 P5_1 的功能选择寄存器	P5_1S	XXXX X000b
0400C4h	端口 P4_2 的功能选择寄存器	P4_2S	X0XX X000b
0400C5h	端口 P5_2 的功能选择寄存器	P5_2S	XXXX X000b
0400C6h	端口 P4_3 的功能选择寄存器	P4_3S	X0XX X000b
0400C7h	端口 P5_3 的功能选择寄存器	P5_3S	XXXX X000b
0400C8h	端口 P4_4 的功能选择寄存器	P4_4S	X0XX X000b
0400C9h	端口 P5_4 的功能选择寄存器	P5_4S	X0XX X000b
0400CAh	端口 P4_5 的功能选择寄存器	P4_5S	X0XX X000b
0400CBh	端口 P5_5 的功能选择寄存器	P5_5S	X0XX X000b
0400CCh	端口 P4_6 的功能选择寄存器	P4_6S	X0XX X000b
0400CDh	端口 P5_6 的功能选择寄存器	P5_6S	X0XX X000b
0400CEh	端口 P4_7 的功能选择寄存器	P4_7S	X0XX X000b
0400CFh	端口 P5_7 的功能选择寄存器	P5_7S	X0XX X000b
0400D0h	端口 P6_0 的功能选择寄存器	P6_0S	X0XX X000b
0400D1h	端口 P7_0 的功能选择寄存器	P7_0S	X0XX X000b
0400D2h	端口 P6_1 的功能选择寄存器	P6_1S	X0XX X000b
0400D3h	端口 P7_1 的功能选择寄存器	P7_1S	X0XX X000b
0400D4h	端口 P6_2 的功能选择寄存器	P6_2S	X0XX X000b
0400D5h	端口 P7_2 的功能选择寄存器	P7_2S	X0XX X000b
0400D6h	端口 P6_3 的功能选择寄存器	P6_3S	X0XX X000b
0400D7h	端口 P7_3 的功能选择寄存器	P7_3S	X0XX X000b
0400D8h	端口 P6_4 的功能选择寄存器	P6_4S	X0XX X000b
0400D9h	端口 P7_4 的功能选择寄存器	P7_4S	X0XX X000b
0400DAh	端口 P6_5 的功能选择寄存器	P6_5S	X0XX X000b
0400DBh	端口 P7_5 的功能选择寄存器	P7_5S	X0XX X000b
0400DCh	端口 P6_6 的功能选择寄存器	P6_6S	X0XX X000b
0400DDh	端口 P7_6 的功能选择寄存器	P7_6S	X0XX X000b
0400DEh	端口 P6_7 的功能选择寄存器	P6_7S	X0XX X000b
0400DFh	端口 P7_7 的功能选择寄存器	P7_7S	X0XX X000b
0400E0h	端口 P8_0 的功能选择寄存器	P8_0S	X0XX X000b
0400E1h	端口 P9_0 的功能选择寄存器	P9_0S	X0XX X000b
0400E2h	端口 P8_1 的功能选择寄存器	P8_1S	X0XX X000b
0400E3h	端口 P9_1 的功能选择寄存器	P9_1S	X0XX X000b
0400E4h	端口 P8_2 的功能选择寄存器	P8_2S	X0XX X000b
0400E5h	端口 P9_2 的功能选择寄存器	P9_2S	X0XX X000b
0400E6h	端口 P8_3 的功能选择寄存器	P8_3S	X0XX X000b
0400E7h	端口 P9_3 的功能选择寄存器	P9_3S	00XX X000b
0400E8h	端口 P8_4 的功能选择寄存器	P8_4S	XXXX X000b
0400E9h	端口 P9_4 的功能选择寄存器	P9_4S	00XX X000b
0400EAh			
0400EBh	端口 P9_5 的功能选择寄存器	P9_5S	00XX X000b
0400ECh	端口 P8_6 的功能选择寄存器	P8_6S	XXXX X000b
0400EDh	端口 P9_6 的功能选择寄存器	P9_6S	00XX X000b
0400EEh	端口 P8_7 的功能选择寄存器	P8_7S	XXXX X000b
0400EFh	端口 P9_7 的功能选择寄存器	P9_7S	X0XX X000b

X: 不定值

空白部分全部为保留区, 不能存取。



表 4.22 SFR 一览表 (22)

地址	寄存器	符号	复位后的值
0400F0h	端口 P10_0 的功能选择寄存器	P10_0S	0XXX X000b
0400F1h	端口 P11_0 的功能选择寄存器	P11_0S	X0XX X000b
0400F2h	端口 P10_1 的功能选择寄存器	P10_1S	0XXX X000b
0400F3h	端口 P11_1 的功能选择寄存器	P11_1S	X0XX X000b
0400F4h	端口 P10_2 的功能选择寄存器	P10_2S	0XXX X000b
0400F5h	端口 P11_2 的功能选择寄存器	P11_2S	X0XX X000b
0400F6h	端口 P10_3 的功能选择寄存器	P10_3S	0XXX X000b
0400F7h	端口 P11_3 的功能选择寄存器	P11_3S	X0XX X000b
0400F8h	端口 P10_4 的功能选择寄存器	P10_4S	0XXX X000b
0400F9h	端口 P11_4 的功能选择寄存器	P11_4S	XXXX X000b
0400FAh	端口 P10_5 的功能选择寄存器	P10_5S	0XXX X000b
0400FBh			
0400FCh	端口 P10_6 的功能选择寄存器	P10_6S	0XXX X000b
0400FDh			
0400FEh	端口 P10_7 的功能选择寄存器	P10_7S	0XXX X000b
0400FFh			
040100h	端口 P12_0 的功能选择寄存器	P12_0S	X0XX X000b
040101h	端口 P13_0 的功能选择寄存器	P13_0S	XXXX X000b
040102h	端口 P12_1 的功能选择寄存器	P12_1S	X0XX X000b
040103h	端口 P13_1 的功能选择寄存器	P13_1S	XXXX X000b
040104h	端口 P12_2 的功能选择寄存器	P12_2S	X0XX X000b
040105h	端口 P13_2 的功能选择寄存器	P13_2S	XXXX X000b
040106h	端口 P12_3 的功能选择寄存器	P12_3S	X0XX X000b
040107h	端口 P13_3 的功能选择寄存器	P13_3S	XXXX X000b
040108h	端口 P12_4 的功能选择寄存器	P12_4S	XXXX X000b
040109h	端口 P13_4 的功能选择寄存器	P13_4S	XXXX X000b
04010Ah	端口 P12_5 的功能选择寄存器	P12_5S	XXXX X000b
04010Bh	端口 P13_5 的功能选择寄存器	P13_5S	XXXX X000b
04010Ch	端口 P12_6 的功能选择寄存器	P12_6S	XXXX X000b
04010Dh	端口 P13_6 的功能选择寄存器	P13_6S	XXXX X000b
04010Eh	端口 P12_7 的功能选择寄存器	P12_7S	XXXX X000b
04010Fh	端口 P13_7 的功能选择寄存器	P13_7S	XXXX X000b
040110h			
040111h	端口 P15_0 的功能选择寄存器	P15_0S	00XX X000b
040112h			
040113h	端口 P15_1 的功能选择寄存器	P15_1S	00XX X000b
040114h			
040115h	端口 P15_2 的功能选择寄存器	P15_2S	00XX X000b
040116h	端口 P14_3 的功能选择寄存器	P14_3S	XXXX X000b
040117h	端口 P15_3 的功能选择寄存器	P15_3S	00XX X000b
040118h	端口 P14_4 的功能选择寄存器	P14_4S	XXXX X000b
040119h	端口 P15_4 的功能选择寄存器	P15_4S	00XX X000b
04011Ah	端口 P14_5 的功能选择寄存器	P14_5S	XXXX X000b
04011Bh	端口 P15_5 的功能选择寄存器	P15_5S	00XX X000b
04011Ch	端口 P14_6 的功能选择寄存器	P14_6S	XXXX X000b
04011Dh	端口 P15_6 的功能选择寄存器	P15_6S	00XX X000b
04011Eh			
04011Fh	端口 P15_7 的功能选择寄存器	P15_7S	00XX X000b

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.23 SFR 一览表 (23)

地址	寄存器	符号	复位后的值
040120h ~ 04403Fh			
044040h			
044041h			
044042h			
044043h			
044044h			
044045h			
044046h			
044047h			
044048h			
044049h			
04404Ah			
04404Bh			
04404Ch			
04404Dh			
04404Eh	看门狗定时器的开始寄存器	WDTS	XXXX XXXXb
04404Fh	看门狗定时器的控制寄存器	WDC	000X XXXXb
044050h			
044051h			
044052h			
044053h			
044054h			
044055h			
044056h			
044057h			
044058h			
044059h			
04405Ah			
04405Bh			
04405Ch			
04405Dh			
04405Eh			
04405Fh	保护寄存器 2	PRCR2	0XXX XXXXb

X: 不定值

空白部分全部为保留区，不能存取。

表 4.24 SFR 一览表 (24)

地址	寄存器	符号	复位后的值
044060h			
044061h			
044062h			
044063h			
044064h			
044065h			
044066h			
044067h			
044068h			
044069h			
04406Ah			
04406Bh			
04406Ch			
04406Dh	外部中断源选择寄存器 1	IFSR1	X0XX X000b
04406Eh			
04406Fh	外部中断源选择寄存器 0	IFSR0	0000 0000b
044070h	DMA0 启动源选择寄存器 2	DM0SL2	XX00 0000b
044071h	DMA1 启动源选择寄存器 2	DM1SL2	XX00 0000b
044072h	DMA2 启动源选择寄存器 2	DM2SL2	XX00 0000b
044073h	DMA3 启动源选择寄存器 2	DM3SL2	XX00 0000b
044074h			
044075h			
044076h			
044077h			
044078h	DMA0 启动源选择寄存器	DM0SL	XXX0 0000b
044079h	DMA1 启动源选择寄存器	DM1SL	XXX0 0000b
04407Ah	DMA2 启动源选择寄存器	DM2SL	XXX0 0000b
04407Bh	DMA3 启动源选择寄存器	DM3SL	XXX0 0000b
04407Ch			
04407Dh	用于返回的中断优先级设定寄存器 2	RIPL2	XX0X 0000b
04407Eh			
04407Fh	用于返回的中断优先级设定寄存器 1	RIPL1	XX0X 0000b
044080h			
044081h			
044082h			
044083h			
044084h			
044085h			
044086h			
044087h			
044088h			
044089h			
04408Ah			
04408Bh			
04408Ch			
04408Dh			
04408Eh			
04408Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.25 SFR 一览表 (25)

地址	寄存器	符号	复位后的值
044090h ~ 0443FFh			
044400h	I <sup>2</sup> C 总线发送 / 接收移位寄存器	I2CTRSR	XXh
044401h			
044402h	I <sup>2</sup> C 总线从属地址寄存器	I2CSAR	00h
044403h	I <sup>2</sup> C 总线控制寄存器 0	I2CCR0	0000 0000b
044404h	I <sup>2</sup> C 总线时钟控制寄存器	I2CCCR	0000 0000b
044405h	I <sup>2</sup> C 总线开始条件 / 停止条件的控制寄存器	I2CSSCR	0000 0000b
044406h	I <sup>2</sup> C 总线控制寄存器 1	I2CCR1	0000 0000b
044407h	I <sup>2</sup> C 总线控制寄存器 2	I2CCR2	0000 0000b
044408h	I <sup>2</sup> C 总线状态寄存器	I2CSR	0000 0000b
044409h			
04440Ah			
04440Bh			
04440Ch			
04440Dh			
04440Eh			
04440Fh			
044410h	I <sup>2</sup> C 总线模式寄存器	I2CMR	0000 0000b
044411h			
044412h			
044413h			
044414h			
044415h			
044416h			
044417h			
044418h			
044419h			
04441Ah			
04441Bh			
04441Ch			
04441Dh			
04441Eh			
04441Fh			
044420h ~ 0457FFh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.26 SFR 一览表 (26)

地址	寄存器	符号	复位后的值		
046800h ~ 047BFFh					
047C00h 047C01h 047C02h 047C03h 047C04h	CAN0 邮箱 0: 消息标识符	COMB0	XXXX XXXXh		
047C05h	CAN0 邮箱 0: 数据长		XXh		
047C06h 047C07h 047C08h 047C09h 047C0Ah 047C0Bh 047C0Ch 047C0Dh	CAN0 邮箱 0: 数据区		XXXX XXXX XXXX XXXXh		
047C0Eh 047C0Fh	CAN0 邮箱 0: 时戳		XXXXh		
047C10h 047C11h 047C12h 047C13h 047C14h	CAN0 邮箱 1: 消息标识符		COMB1	XXXX XXXXh	
047C15h	CAN0 邮箱 1: 数据长			XXh	
047C16h 047C17h 047C18h 047C19h 047C1Ah 047C1Bh 047C1Ch 047C1Dh	CAN0 邮箱 1: 数据区			XXXX XXXX XXXX XXXXh	
047C1Eh 047C1Fh	CAN0 邮箱 1: 时戳			XXXXh	
047C20h 047C21h 047C22h 047C23h 047C24h	CAN0 邮箱 2: 消息标识符			COMB2	XXXX XXXXh
047C25h	CAN0 邮箱 2: 数据长				XXh
047C26h 047C27h 047C28h 047C29h 047C2Ah 047C2Bh 047C2Ch 047C2Dh	CAN0 邮箱 2: 数据区				XXXX XXXX XXXX XXXXh
047C2Eh 047C2Fh	CAN0 邮箱 2: 时戳				XXXXh

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.27 SFR 一览表 (27)

地址	寄存器	符号	复位后的值
047C30h	CAN0 邮箱 3: 消息标识符	COMB3	XXXX XXXXh
047C31h			
047C32h			
047C33h			
047C34h			
047C35h	CAN0 邮箱 3: 数据长		XXh
047C36h	CAN0 邮箱 3: 数据区		XXXX XXXX
047C37h			XXXX XXXXh
047C38h			
047C39h			
047C3Ah			
047C3Bh			
047C3Ch			
047C3Dh			
047C3Eh	CAN0 邮箱 3: 时戳		XXXXh
047C3Fh			
047C40h	CAN0 邮箱 4: 消息标识符	COMB4	XXXX XXXXh
047C41h			
047C42h			
047C43h			
047C44h			
047C45h	CAN0 邮箱 4: 数据长		XXh
047C46h	CAN0 邮箱 4: 数据区		XXXX XXXX
047C47h			XXXX XXXXh
047C48h			
047C49h			
047C4Ah			
047C4Bh			
047C4Ch			
047C4Dh			
047C4Eh	CAN0 邮箱 4: 时戳		XXXXh
047C4Fh			
047C50h	CAN0 邮箱 5: 消息标识符	COMB5	XXXX XXXXh
047C51h			
047C52h			
047C53h			
047C54h			
047C55h	CAN0 邮箱 5: 数据长		XXh
047C56h	CAN0 邮箱 5: 数据区		XXXX XXXX
047C57h			XXXX XXXXh
047C58h			
047C59h			
047C5Ah			
047C5Bh			
047C5Ch			
047C5Dh			
047C5Eh	CAN0 邮箱 5: 时戳		XXXXh
047C5Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.28 SFR 一览表 (28)

地址	寄存器	符号	复位后的值
047C60h	CAN0 邮箱 6: 消息标识符	COMB6	XXXX XXXXh
047C61h			
047C62h			
047C63h			
047C64h			
047C65h	CAN0 邮箱 6: 数据长		XXh
047C66h	CAN0 邮箱 6: 数据区		XXXX XXXX
047C67h			XXXX XXXXh
047C68h			
047C69h			
047C6Ah			
047C6Bh			
047C6Ch			
047C6Dh			
047C6Eh	CAN0 邮箱 6: 时戳		XXXXh
047C6Fh			
047C70h	CAN0 邮箱 7: 消息标识符	COMB7	XXXX XXXXh
047C71h			
047C72h			
047C73h			
047C74h			
047C75h	CAN0 邮箱 7: 数据长		XXh
047C76h	CAN0 邮箱 7: 数据区		XXXX XXXX
047C77h			XXXX XXXXh
047C78h			
047C79h			
047C7Ah			
047C7Bh			
047C7Ch			
047C7Dh			
047C7Eh	CAN0 邮箱 7: 时戳		XXXXh
047C7Fh			
047C80h	CAN0 邮箱 8: 消息标识符	COMB8	XXXX XXXXh
047C81h			
047C82h			
047C83h			
047C84h			
047C85h	CAN0 邮箱 8: 数据长		XXh
047C86h	CAN0 邮箱 8: 数据区		XXXX XXXX
047C87h			XXXX XXXXh
047C88h			
047C89h			
047C8Ah			
047C8Bh			
047C8Ch			
047C8Dh			
047C8Eh	CAN0 邮箱 8: 时戳		XXXXh
047C8Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.29 SFR 一览表 (29)

地址	寄存器	符号	复位后的值
047C90h	CAN0 邮箱 9: 消息标识符	COMB9	XXXX XXXXh
047C91h			
047C92h			
047C93h			
047C94h			
047C95h	CAN0 邮箱 9: 数据长		XXh
047C96h	CAN0 邮箱 9: 数据区		XXXX XXXX
047C97h			XXXX XXXXh
047C98h			
047C99h			
047C9Ah			
047C9Bh			
047C9Ch			
047C9Dh			
047C9Eh	CAN0 邮箱 9: 时戳		XXXXh
047C9Fh			
047CA0h	CAN0 邮箱 10: 消息标识符	COMB10	XXXX XXXXh
047CA1h			
047CA2h			
047CA3h			
047CA4h			
047CA5h	CAN0 邮箱 10: 数据长		XXh
047CA6h	CAN0 邮箱 10: 数据区		XXXX XXXX
047CA7h			XXXX XXXXh
047CA8h			
047CA9h			
047CAAh			
047CABh			
047CACh			
047CADh			
047CAEh	CAN0 邮箱 10: 时戳		XXXXh
047CAFh			
047CB0h	CAN0 邮箱 11: 消息标识符	COMB11	XXXX XXXXh
047CB1h			
047CB2h			
047CB3h			
047CB4h			
047CB5h	CAN0 邮箱 11: 数据长		XXh
047CB6h	CAN0 邮箱 11: 数据区		XXXX XXXX
047CB7h			XXXX XXXXh
047CB8h			
047CB9h			
047CBAh			
047CBBh			
047CBCh			
047CBDh			
047CBEh	CAN0 邮箱 11: 时戳		XXXXh
047CBFh			

X: 不定值

空白部分全部为保留区, 不能存取。



表 4.30 SFR 一览表 (30)

地址	寄存器	符号	复位后的值
047CC0h	CAN0 邮箱 12: 消息标识符	COMB12	XXXX XXXXh
047CC1h			
047CC2h			
047CC3h			
047CC4h			
047CC5h	CAN0 邮箱 12: 数据长		XXh
047CC6h	CAN0 邮箱 12: 数据区		XXXX XXXX
047CC7h			XXXX XXXXh
047CC8h			
047CC9h			
047CCAh			
047CCBh			
047CCCh			
047CCDh			
047CCEh	CAN0 邮箱 12: 时戳		XXXXh
047CCFh			
047CD0h	CAN0 邮箱 13: 消息标识符	COMB13	XXXX XXXXh
047CD1h			
047CD2h			
047CD3h			
047CD4h			
047CD5h	CAN0 邮箱 13: 数据长		XXh
047CD6h	CAN0 邮箱 13: 数据区		XXXX XXXX
047CD7h			XXXX XXXXh
047CD8h			
047CD9h			
047CDAh			
047CDBh			
047CDCh			
047CDDh			
047CDEh	CAN0 邮箱 13: 时戳		XXXXh
047CDFh			
047CE0h	CAN0 邮箱 14: 消息标识符	COMB14	XXXX XXXXh
047CE1h			
047CE2h			
047CE3h			
047CE4h			
047CE5h	CAN0 邮箱 14: 数据长		XXh
047CE6h	CAN0 邮箱 14: 数据区		XXXX XXXX
047CE7h			XXXX XXXXh
047CE8h			
047CE9h			
047CEAh			
047CEBh			
047CECh			
047CEDh			
047CEEh	CAN0 邮箱 14: 时戳		XXXXh
047CEFh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.31 SFR 一览表 (31)

地址	寄存器	符号	复位后的值
047CF0h	CAN0 邮箱 15: 消息标识符	COMB15	XXXX XXXXh
047CF1h			
047CF2h			
047CF3h			
047CF4h			
047CF5h	CAN0 邮箱 15: 数据长		XXh
047CF6h	CAN0 邮箱 15: 数据区		XXXX XXXX
047CF7h			XXXX XXXXh
047CF8h			
047CF9h			
047CFAh			
047CFBh			
047CFC			
047CFDh			
047CFEh	CAN0 邮箱 15: 时戳		XXXXh
047CFFh			
047D00h	CAN0 邮箱 16: 消息标识符	COMB16	XXXX XXXXh
047D01h			
047D02h			
047D03h			
047D04h			
047D05h	CAN0 邮箱 16: 数据长		XXh
047D06h	CAN0 邮箱 16: 数据区		XXXX XXXX
047D07h			XXXX XXXXh
047D08h			
047D09h			
047D0Ah			
047D0Bh			
047D0Ch			
047D0Dh			
047D0Eh	CAN0 邮箱 16: 时戳		XXXXh
047D0Fh			
047D10h	CAN0 邮箱 17: 消息标识符	COMB17	XXXX XXXXh
047D11h			
047D12h			
047D13h			
047D14h			
047D15h	CAN0 邮箱 17: 数据长		XXh
047D16h	CAN0 邮箱 17: 数据区		XXXX XXXX
047D17h			XXXX XXXXh
047D18h			
047D19h			
047D1Ah			
047D1Bh			
047D1Ch			
047D1Dh			
047D1Eh	CAN0 邮箱 17: 时戳		XXXXh
047D1Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.32 SFR 一览表 (32)

地址	寄存器	符号	复位后的值
047D20h	CAN0 邮箱 18: 消息标识符	COMB18	XXXX XXXXh
047D21h			
047D22h			
047D23h			
047D24h			
047D25h	CAN0 邮箱 18: 数据长		XXh
047D26h	CAN0 邮箱 18: 数据区		XXXX XXXX
047D27h			
047D28h			
047D29h			
047D2Ah			
047D2Bh			
047D2Ch			
047D2Dh			
047D2Eh	CAN0 邮箱 18: 时戳		XXXXh
047D2Fh			
047D30h	CAN0 邮箱 19: 消息标识符	COMB19	XXXX XXXXh
047D31h			
047D32h			
047D33h			
047D34h			
047D35h	CAN0 邮箱 19: 数据长		XXh
047D36h	CAN0 邮箱 19: 数据区		XXXX XXXX
047D37h			
047D38h			
047D39h			
047D3Ah			
047D3Bh			
047D3Ch			
047D3Dh			
047D3Eh	CAN0 邮箱 19: 时戳		XXXXh
047D3Fh			
047D40h	CAN0 邮箱 20: 消息标识符	COMB20	XXXX XXXXh
047D41h			
047D42h			
047D43h			
047D44h			
047D45h	CAN0 邮箱 20: 数据长		XXh
047D46h	CAN0 邮箱 20: 数据区		XXXX XXXX
047D47h			
047D48h			
047D49h			
047D4Ah			
047D4Bh			
047D4Ch			
047D4Dh			
047D4Eh	CAN0 邮箱 20: 时戳		XXXXh
047D4Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.33 SFR 一览表 (33)

地址	寄存器	符号	复位后的值
047D50h	CAN0 邮箱 21: 消息标识符	COMB21	XXXX XXXXh
047D51h			
047D52h			
047D53h			
047D54h			
047D55h	CAN0 邮箱 21: 数据长		XXh
047D56h	CAN0 邮箱 21: 数据区		XXXX XXXX
047D57h			XXXX XXXXh
047D58h			
047D59h			
047D5Ah			
047D5Bh			
047D5Ch			
047D5Dh			
047D5Eh	CAN0 邮箱 21: 时戳		XXXXh
047D5Fh			
047D60h	CAN0 邮箱 22: 消息标识符	COMB22	XXXX XXXXh
047D61h			
047D62h			
047D63h			
047D64h			
047D65h	CAN0 邮箱 22: 数据长		XXh
047D66h	CAN0 邮箱 22: 数据区		XXXX XXXX
047D67h			XXXX XXXXh
047D68h			
047D69h			
047D6Ah			
047D6Bh			
047D6Ch			
047D6Dh			
047D6Eh	CAN0 邮箱 22: 时戳		XXXXh
047D6Fh			
047D70h	CAN0 邮箱 23: 消息标识符	COMB23	XXXX XXXXh
047D71h			
047D72h			
047D73h			
047D74h			
047D75h	CAN0 邮箱 23: 数据长		XXh
047D76h	CAN0 邮箱 23: 数据区		XXXX XXXX
047D77h			XXXX XXXXh
047D78h			
047D79h			
047D7Ah			
047D7Bh			
047D7Ch			
047D7Dh			
047D7Eh	CAN0 邮箱 23: 时戳		XXXXh
047D7Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.34 SFR 一览表 (34)

地址	寄存器	符号	复位后的值
047D80h	CAN0 邮箱 24: 消息标识符	COMB24	XXXX XXXXh
047D81h			
047D82h			
047D83h			
047D84h			
047D85h	CAN0 邮箱 24: 数据长		XXh
047D86h	CAN0 邮箱 24: 数据区		XXXX XXXX
047D87h			XXXX XXXXh
047D88h			
047D89h			
047D8Ah			
047D8Bh			
047D8Ch			
047D8Dh			
047D8Eh	CAN0 邮箱 24: 时戳		XXXXh
047D8Fh			
047D90h	CAN0 邮箱 25: 消息标识符	COMB25	XXXX XXXXh
047D91h			
047D92h			
047D93h			
047D94h			
047D95h	CAN0 邮箱 25: 数据长		XXh
047D96h	CAN0 邮箱 25: 数据区		XXXX XXXX
047D97h			XXXX XXXXh
047D98h			
047D99h			
047D9Ah			
047D9Bh			
047D9Ch			
047D9Dh			
047D9Eh	CAN0 邮箱 25: 时戳		XXXXh
047D9Fh			
047DA0h	CAN0 邮箱 26: 消息标识符	COMB26	XXXX XXXXh
047DA1h			
047DA2h			
047DA3h			
047DA4h			
047DA5h	CAN0 邮箱 26: 数据长		XXh
047DA6h	CAN0 邮箱 26: 数据区		XXXX XXXX
047DA7h			XXXX XXXXh
047DA8h			
047DA9h			
047DAAh			
047DABh			
047DACH			
047DADh			
047DAEh	CAN0 邮箱 26: 时戳		XXXXh
047DAFh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.35 SFR 一览表 (35)

地址	寄存器	符号	复位后的值
047DB0h	CAN0 邮箱 27: 消息标识符	COMB27	XXXX XXXXh
047DB1h			
047DB2h			
047DB3h			
047DB4h			
047DB5h	CAN0 邮箱 27: 数据长		XXh
047DB6h	CAN0 邮箱 27: 数据区		XXXX XXXX XXXX XXXXh
047DB7h			
047DB8h			
047DB9h			
047DBAh			
047DBBh			
047DBCh			
047DBDh			
047DBEh	CAN0 邮箱 27: 时戳		XXXXh
047DBFh			
047DC0h	CAN0 邮箱 28: 消息标识符	COMB28	XXXX XXXXh
047DC1h			
047DC2h			
047DC3h			
047DC4h			
047DC5h	CAN0 邮箱 28: 数据长		XXh
047DC6h	CAN0 邮箱 28: 数据区		XXXX XXXX XXXX XXXXh
047DC7h			
047DC8h			
047DC9h			
047DCAh			
047DCBh			
047DCCCh			
047DCDh			
047DCEh	CAN0 邮箱 28: 时戳		XXXXh
047DCFh			
047DD0h	CAN0 邮箱 29: 消息标识符	COMB29	XXXX XXXXh
047DD1h			
047DD2h			
047DD3h			
047DD4h			
047DD5h	CAN0 邮箱 29: 数据长		XXh
047DD6h	CAN0 邮箱 29: 数据区		XXXX XXXX XXXX XXXXh
047DD7h			
047DD8h			
047DD9h			
047DDAh			
047DDBh			
047DDCh			
047DDDh			
047DDEh	CAN0 邮箱 29: 时戳		XXXXh
047DDFh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.36 SFR 一览表 (36)

地址	寄存器	符号	复位后的值
047DE0h	CAN0 邮箱 30: 消息标识符	COMB30	XXXX XXXXh
047DE1h			
047DE2h			
047DE3h			
047DE4h			
047DE5h	CAN0 邮箱 30: 数据长		XXh
047DE6h	CAN0 邮箱 30: 数据区		XXXX XXXX XXXX XXXXh
047DE7h			
047DE8h			
047DE9h			
047DEAh			
047DEBh			
047DECb			
047DEDh			
047DEEh	CAN0 邮箱 30: 时戳		XXXXh
047DEFh			
047DF0h	CAN0 邮箱 31: 消息标识符	COMB31	XXXX XXXXh
047DF1h			
047DF2h			
047DF3h			
047DF4h			
047DF5h	CAN0 邮箱 31: 数据长		XXh
047DF6h	CAN0 邮箱 31: 数据区		XXXX XXXX XXXX XXXXh
047DF7h			
047DF8h			
047DF9h			
047DFAh			
047DFBh			
047DFCh			
047DFDh			
047DFEh	CAN0 邮箱 31: 时戳		XXXXh
047DFFh			
047E00h	CAN0 接收屏蔽寄存器 0	COMKR0	XXXX XXXXh
047E01h			
047E02h			
047E03h			
047E04h	CAN0 接收屏蔽寄存器 1	COMKR1	XXXX XXXXh
047E05h			
047E06h			
047E07h			
047E08h	CAN0 接收屏蔽寄存器 2	COMKR2	XXXX XXXXh
047E09h			
047E0Ah			
047E0Bh			
047E0Ch	CAN0 接收屏蔽寄存器 3	COMKR3	XXXX XXXXh
047E0Dh			
047E0Eh			
047E0Fh			

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.37 SFR 一览表 (37)

地址	寄存器	符号	复位后的值
047E10h	CAN0 接收屏蔽寄存器 4	COMKR4	XXXX XXXXh
047E11h			
047E12h			
047E13h			
047E14h	CAN0 接收屏蔽寄存器 5	COMKR5	XXXX XXXXh
047E15h			
047E16h			
047E17h			
047E18h	CAN0 接收屏蔽寄存器 6	COMKR6	XXXX XXXXh
047E19h			
047E1Ah			
047E1Bh			
047E1Ch	CAN0 接收屏蔽寄存器 7	COMKR7	XXXX XXXXh
047E1Dh			
047E1Eh			
047E1Fh			
047E20h	CAN0FIFO 接收 ID 比较寄存器 0	C0FIDCR0	XXXX XXXXh
047E21h			
047E22h			
047E23h			
047E24h	CAN0FIFO 接收 ID 比较寄存器 1	C0FIDCR1	XXXX XXXXh
047E25h			
047E26h			
047E27h			
047E28h	CAN0 屏蔽无效寄存器	COMKIVLR	XXXX XXXXh
047E29h			
047E2Ah			
047E2Bh			
047E2Ch	CAN0 邮箱中断允许寄存器	COMIER	XXXX XXXXh
047E2Dh			
047E2Eh			
047C4Fh			
047E30h			
047E31h			
047E32h			
047E33h			
047E34h			
047E35h			
047E36h			
047E37h			
047E38h			
047E39h			
047E3Ah			
047E3Bh			
047E3Ch			
047E3Dh			
047E3Eh			
047E3Fh			
047E40h ~ 047F1Fh			

X: 不定值

空白部分全部为保留区, 不能存取。



表 4.38 SFR 一览表 (38)

地址	寄存器	符号	复位后的值
047F20h	CAN0 消息控制寄存器 0	COMCTL0	00h
047F21h	CAN0 消息控制寄存器 1	COMCTL1	00h
047F22h	CAN0 消息控制寄存器 2	COMCTL2	00h
047F23h	CAN0 消息控制寄存器 3	COMCTL3	00h
047F24h	CAN0 消息控制寄存器 4	COMCTL4	00h
047F25h	CAN0 消息控制寄存器 5	COMCTL5	00h
047F26h	CAN0 消息控制寄存器 6	COMCTL6	00h
047F27h	CAN0 消息控制寄存器 7	COMCTL7	00h
047F28h	CAN0 消息控制寄存器 8	COMCTL8	00h
047F29h	CAN0 消息控制寄存器 9	COMCTL9	00h
047F2Ah	CAN0 消息控制寄存器 10	COMCTL10	00h
047F2Bh	CAN0 消息控制寄存器 11	COMCTL11	00h
047F2Ch	CAN0 消息控制寄存器 12	COMCTL12	00h
047F2Dh	CAN0 消息控制寄存器 13	COMCTL13	00h
047F2Eh	CAN0 消息控制寄存器 14	COMCTL14	00h
047F2Fh	CAN0 消息控制寄存器 15	COMCTL15	00h
047F30h	CAN0 消息控制寄存器 16	COMCTL16	00h
047F31h	CAN0 消息控制寄存器 17	COMCTL17	00h
047F32h	CAN0 消息控制寄存器 18	COMCTL18	00h
047F33h	CAN0 消息控制寄存器 19	COMCTL19	00h
047F34h	CAN0 消息控制寄存器 20	COMCTL20	00h
047F35h	CAN0 消息控制寄存器 21	COMCTL21	00h
047F36h	CAN0 消息控制寄存器 22	COMCTL22	00h
047F37h	CAN0 消息控制寄存器 23	COMCTL23	00h
047F38h	CAN0 消息控制寄存器 24	COMCTL24	00h
047F39h	CAN0 消息控制寄存器 25	COMCTL25	00h
047F3Ah	CAN0 消息控制寄存器 26	COMCTL26	00h
047F3Bh	CAN0 消息控制寄存器 27	COMCTL27	00h
047F3Ch	CAN0 消息控制寄存器 28	COMCTL28	00h
047F3Dh	CAN0 消息控制寄存器 29	COMCTL29	00h
047F3Eh	CAN0 消息控制寄存器 30	COMCTL30	00h
047F3Fh	CAN0 消息控制寄存器 31	COMCTL31	00h

X: 不定值

空白部分全部为保留区, 不能存取。

表 4.39 SFR 一览表 (39)

地址	寄存器	符号	复位后的值
047F40h	CAN0 控制寄存器	C0CTLR	0000 0101b
047F41h			0000 0000b
047F42h	CAN0 状态寄存器	C0STR	0000 0101b
047F43h			0000 0000b
047F44h	CAN0 位结构寄存器	C0BCR	00 0000h
047F45h			
047F46h			
047F47h	CAN0 时钟选择寄存器	C0CLKR	000X 0000b
047F48h	CAN0 接收 FIFO 控制寄存器	C0RFCR	1000 0000b
047F49h	CAN0 接收 FIFO 指针控制寄存器	C0RFPCR	XXh
047F4Ah	CAN0 发送 FIFO 控制寄存器	C0TFCR	1000 0000b
047F4Bh	CAN0 发送 FIFO 指针控制寄存器	C0TFPCR	XXh
047F4Ch	CAN0 错误中断允许寄存器	C0EIER	00h
047F4Dh	CAN0 错误中断源判断寄存器	C0EIFR	00h
047F4Eh	CAN0 接收错误计数寄存器	C0RECR	00h
047F4Fh	CAN0 发送错误计数寄存器	C0TECR	00h
047F50h	CAN0 错误码保存寄存器	C0ECSR	00h
047F51h	CAN0 通道搜索支持寄存器	C0CSSR	XXh
047F52h	CAN0 邮箱搜索状态寄存器	C0MSSR	1000 0000b
047F53h	CAN0 邮箱搜索模式寄存器	C0MSMR	XXXX XX00b
047F54h	CAN0 时戳寄存器	C0TSR	0000h
047F55h			
047F56h	CAN0 接收滤波器支持寄存器	C0AFSR	XXXXh
047F57h			
047F58h	CAN0 测试控制寄存器	C0TCR	00h
047F59h			
047F5Ah			
047F5Bh			
047F5Ch			
047F5Dh			
047F5Eh			
047F60h ~ 047FFFh			
048000h ~ 04FFFFh			

X: 不定值

空白部分全部为保留区, 不能存取。

## 5. 复位

复位有硬件复位、软件复位和看门狗定时器复位。

### 5.1 硬件复位

硬件复位是由  $\overline{\text{RESET}}$  引脚控制的复位。当电源电压满足推荐的工作条件时，如果将“L”电平输入到  $\overline{\text{RESET}}$  引脚，引脚就被初始化（参照表 5.1），并且振荡电路也被初始化，主时钟开始振荡。如果将  $\overline{\text{RESET}}$  引脚的输入电平从“L”电平变为“H”电平，就对 CPU 和 SFR 进行初始化，并从复位向量指向的地址开始执行程序。但是不对内部 RAM 进行初始化。如果在写内部 RAM 的过程中  $\overline{\text{RESET}}$  引脚变为“L”电平，内部 RAM 就为不定值。

复位电路的例子、复位顺序以及复位后的 CPU 寄存器状态分别如图 5.1、图 5.2 和图 5.3 所示， $\overline{\text{RESET}}$  引脚的电平为“L”电平期间的引脚状态如表 5.1 所示。复位后的 SFR 状态请参照“4. SFR”。

#### A. 电源稳定的情况

- (1) 将“L”电平输入到  $\overline{\text{RESET}}$  引脚。
- (2) 至少将 20 个周期的时钟输入到 XIN 引脚。
- (3) 将“H”电平输入到  $\overline{\text{RESET}}$  引脚。

#### B. 接通电源的情况

- (1) 将“L”电平输入到  $\overline{\text{RESET}}$  引脚。
- (2) 使电源电压上升到满足推荐工作条件的电平。
- (3) 等待  $t_d(\text{P-R})$  直到内部电源稳定。
- (4) 至少将 20 个周期的时钟输入到 XIN 引脚。
- (5) 将“H”电平输入到  $\overline{\text{RESET}}$  引脚。

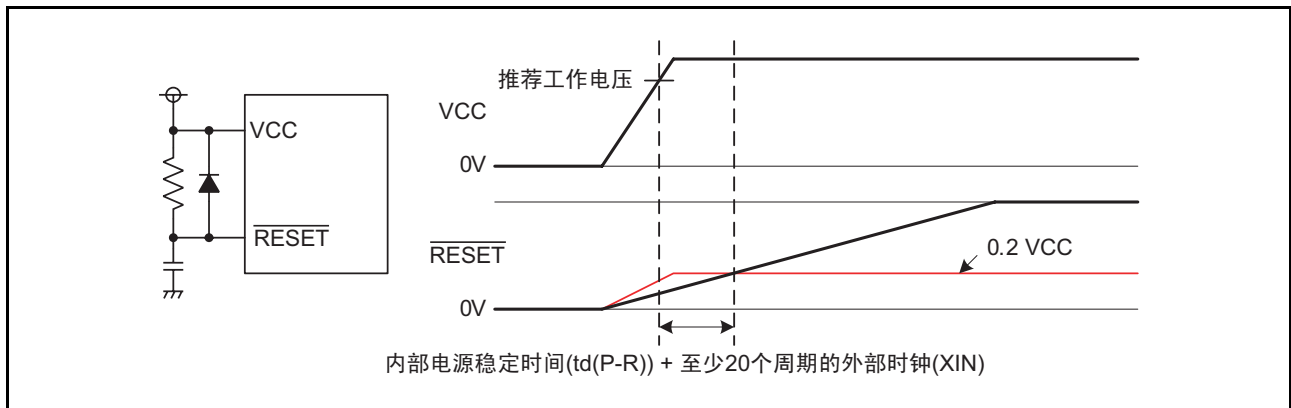


图 5.1 复位电路的例子

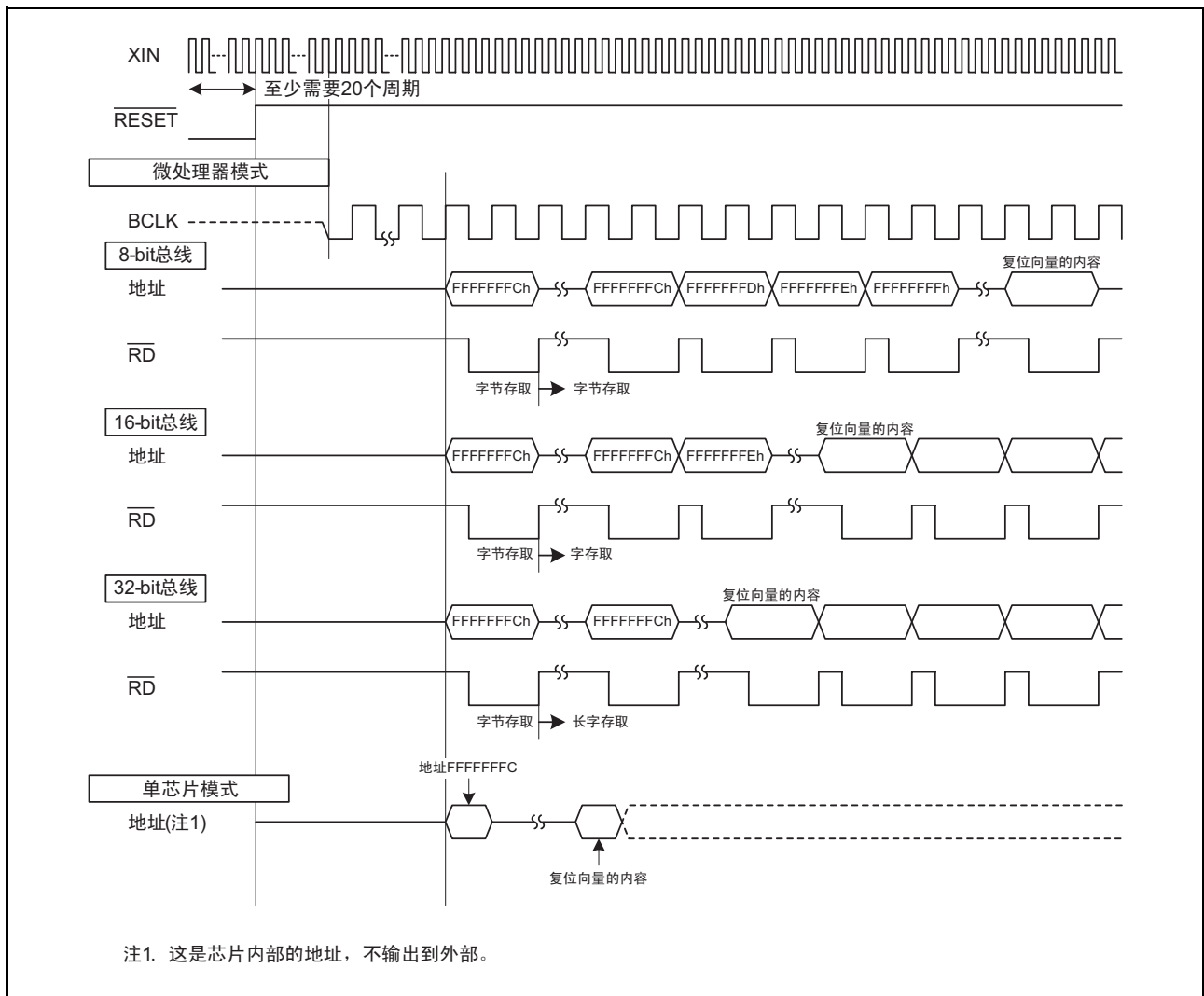


图 5.2 复位顺序

表 5.1 RESET 引脚的电平为“L”电平期间的引脚状态（注1）

引脚名	引脚状态	
	CNVSS=VSS	CNVSS=VCC
P0	输入端口（高阻抗）	数据输入
P1	输入端口（高阻抗）	输入端口（高阻抗）
P2、P3	输入端口（高阻抗）	地址输出（不确定）
P4_0 ~ P4_6	输入端口（高阻抗）	地址输出（不确定）
P4_7	输入端口（高阻抗）	$\overline{CS0}$ 输出（输出“H”电平）
P5_0	输入端口（高阻抗）	$\overline{WR}$ 输出（输出“H”电平）
P5_1	输入端口（高阻抗）	$\overline{BC1}$ 输出（不确定）
P5_2	输入端口（高阻抗）	$\overline{RD}$ 输出（输出“H”电平）
P5_3	输入端口（高阻抗）	BCLK 输出（注2）
P5_4	输入端口（高阻抗）	$\overline{HLDA}$ 输出（输出值取决于 $\overline{HOLD}$ 引脚的输入值）（注2）
P5_5	输入端口（高阻抗）	$\overline{HOLD}$ 输入（高阻抗）
P5_6	输入端口（高阻抗）	$\overline{CS2}$ 输出（输出“H”电平）
P5_7	输入端口（高阻抗）	$\overline{RDY}$ 输入（高阻抗）
P6 ~ P10	输入端口（高阻抗）	输入端口（高阻抗）
P11 ~ P15（注3）	输入端口（高阻抗）	输入端口（高阻抗）

注1. 在内部电源电压稳定前，不能确定是否有上拉电阻。

注2. 这是在接通电源后内部电源电压稳定后的状态。在内部电压稳定前，引脚状态不确定。

注3. 端口 P11 ~ P15 只限于 144 引脚版。

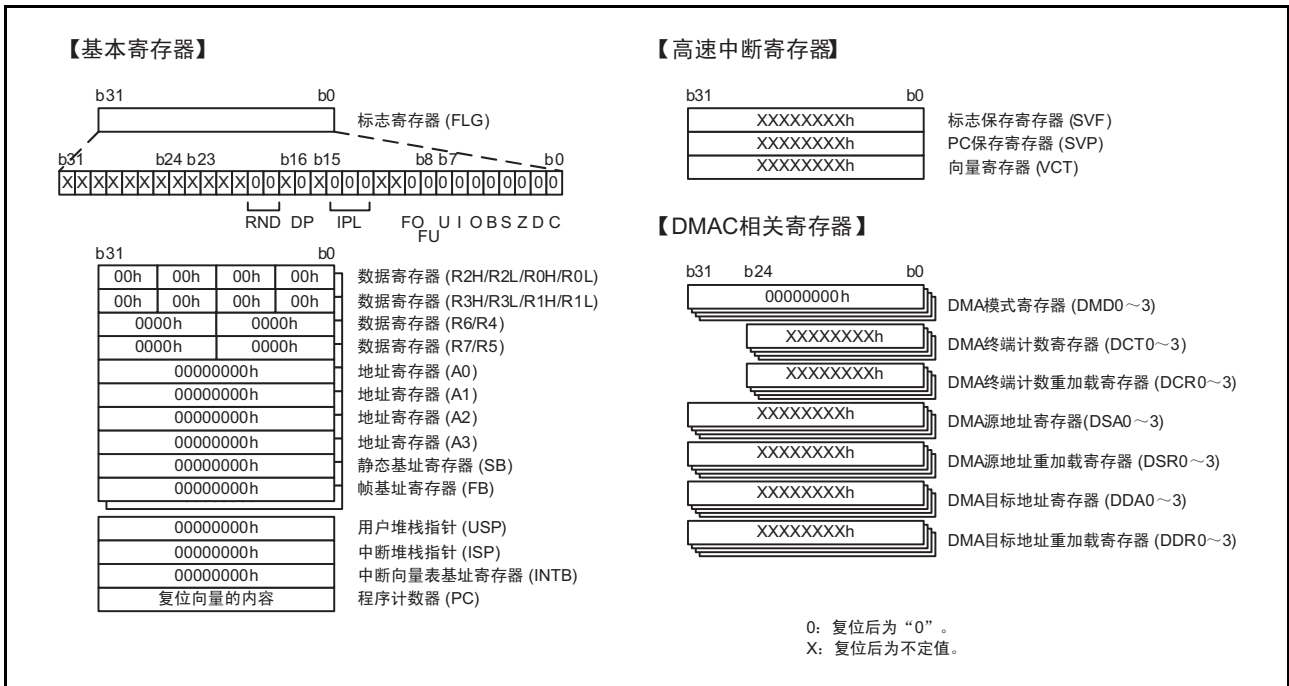


图 5.3 复位后的 CPU 寄存器状态

## 5.2 软件复位

如果将 PM0 寄存器的 PM03 位置“1”（将单片机复位），单片机就对 CPU、SFR 和引脚进行初始化。然后，从复位向量指向的地址开始执行程序。

必须选择 PLL 时钟作为 CPU 时钟源，并且在主时钟振荡充分稳定的状态下将 PM03 位置“1”。

在软件复位时，不对 PM0 寄存器的 PM01 ~ PM00 位进行初始化，所以微处理器模式不变。

## 5.3 看门狗定时器复位

当 CM0 寄存器的 CM06 位为“1”（在看门狗定时器下溢时复位）时，如果看门狗定时器发生下溢，单片机就对 CPU、SFR 和引脚进行初始化。然后，从复位向量指向的地址执行程序。

在看门狗定时器复位时，不对 PM0 寄存器的 PM01 ~ PM00 位进行初始化，所以微处理器模式不变。

## 5.4 复位向量

R32C/100 系列的复位向量的结构如图 5.4 所示。

将复位向量内容的低 2 位置“00b”的地址作为程序的起始地址，低 2 位是微处理器模式的外部总线宽度指定位。因此，为了使低 2 位为“00b”，需要将程序的起始地址对准 4 字节分配。

在单芯片模式中，必须将外部总线宽度指定位“00b”。

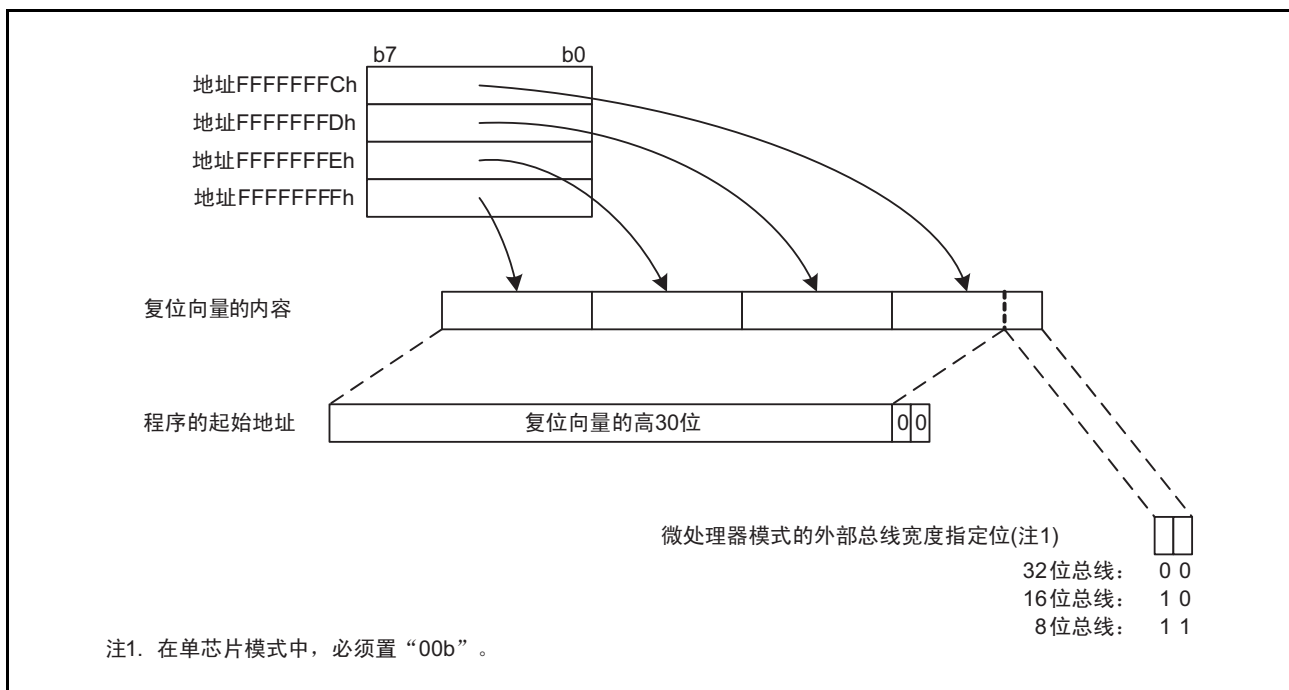


图 5.4 复位向量的结构

## 6. 电源管理

### 6.1 电压稳压器

通过内部电压稳压器降低 VCC 引脚的输入电压，生成内部逻辑电压。

内部电压稳压器的框图和电压稳压器的控制寄存器分别如图 6.1 和图 6.2 所示。

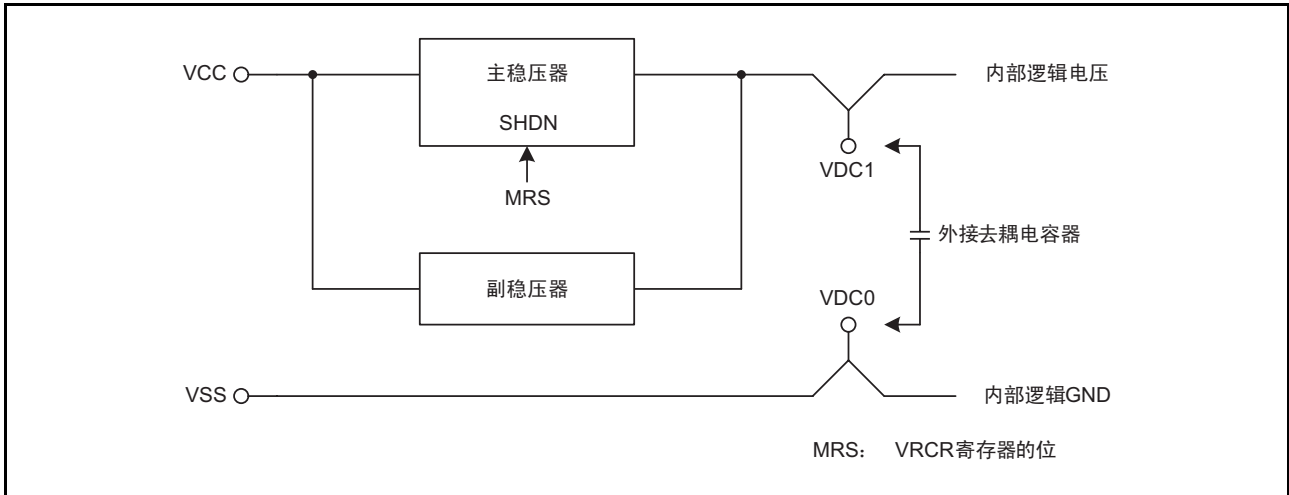


图 6.1 内部电压稳压器的框图

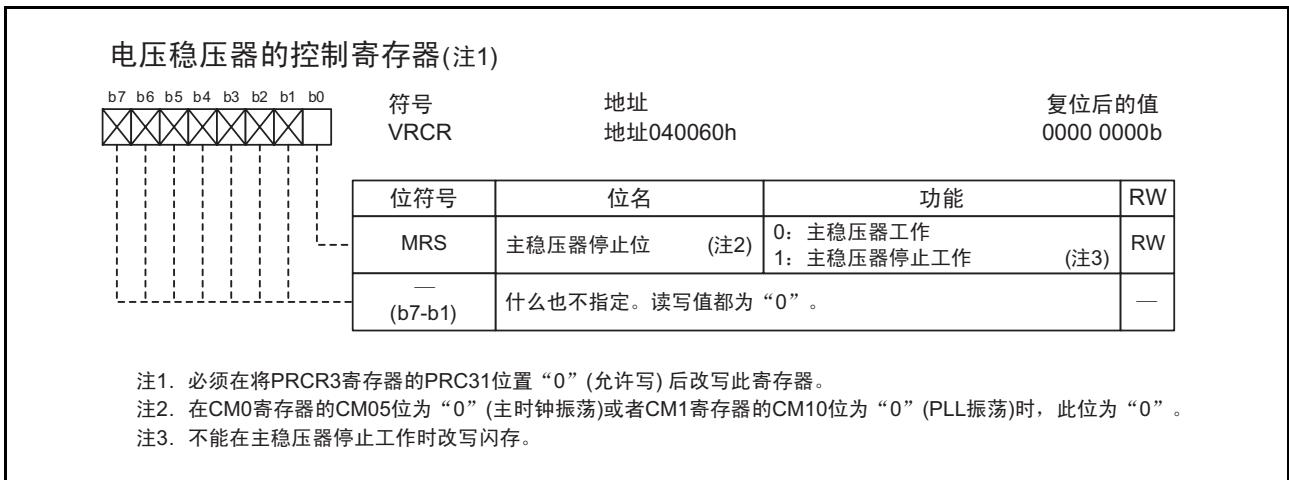


图 6.2 VRCCR 寄存器

### 6.1.1 去耦电容器

为了使内部电压稳定，芯片的外部需要去耦电容器。必须选择高频特性和静电电容温度特性都良好的电容器，一般推荐陶瓷电容器。电容器的电容因使用的温度、两个引脚之间的直流电压和老化等条件而发生变化。必须在考虑这些条件之后，选择满足电特性记载的电容值的电容器。

推荐的电容器一览表如表 6.1 所示。

在 VDC1/VDC0 引脚和去耦电容器之间，必须尽量使用粗而短的布线。

表 6.1 推荐的电容器一览表

温度特性			额定电压	静电电容	电容误差
特性符号	温度范围	静电电容变化率			
B	JIS	-25°C ~ 85°C	≥ 6.3V	4.7μF	≤ ±20%
R	JIS	-55°C ~ 125°C	≥ 6.3V	4.7μF	≤ ±20%
X5R	EIA	-55°C ~ 85°C	≥ 6.3V	4.7μF	≤ ±20%
X7R	EIA	-55°C ~ 125°C	≥ 6.3V	4.7μF	≤ ±20%
X8R	EIA	-55°C ~ 150°C	≥ 6.3V	4.7μF	≤ ±20%
X6S	EIA	-55°C ~ 105°C	≥ 6.3V	4.7μF	≤ ±20%
X7S	EIA	-55°C ~ 125°C	≥ 6.3V	4.7μF	≤ ±20%

### 6.2 低电压检测电路

低电压检测电路是监视向 VCC 引脚输入的电源电压的电路。

能监视内部电压稳压器输入的 VCC 电源电压，并且在内部逻辑电压下降前发出警告。在 CPU 接受此警告后，到实际内部逻辑电压下降前的期间，能将重要的数据保存到闪存，并安全关机。

低电压检测电路的框图和相关寄存器分别如图 6.3 和图 6.4 ~ 图 6.5 所示。

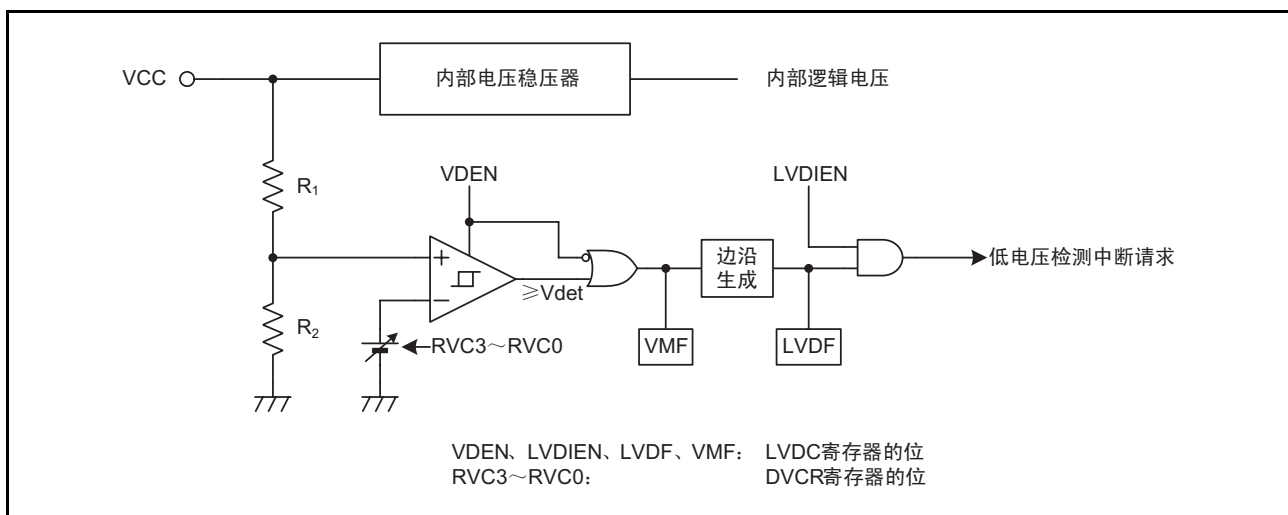


图 6.3 低电压检测电路的框图



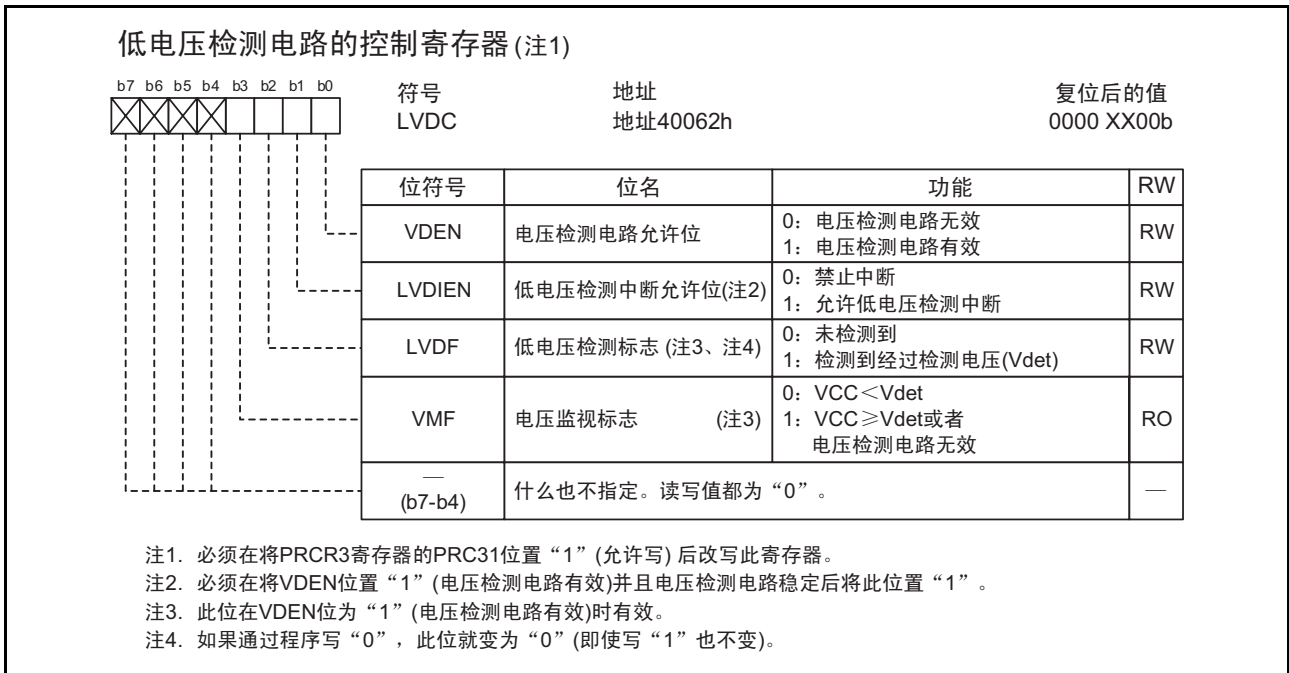


图 6.4 LVDC 寄存器

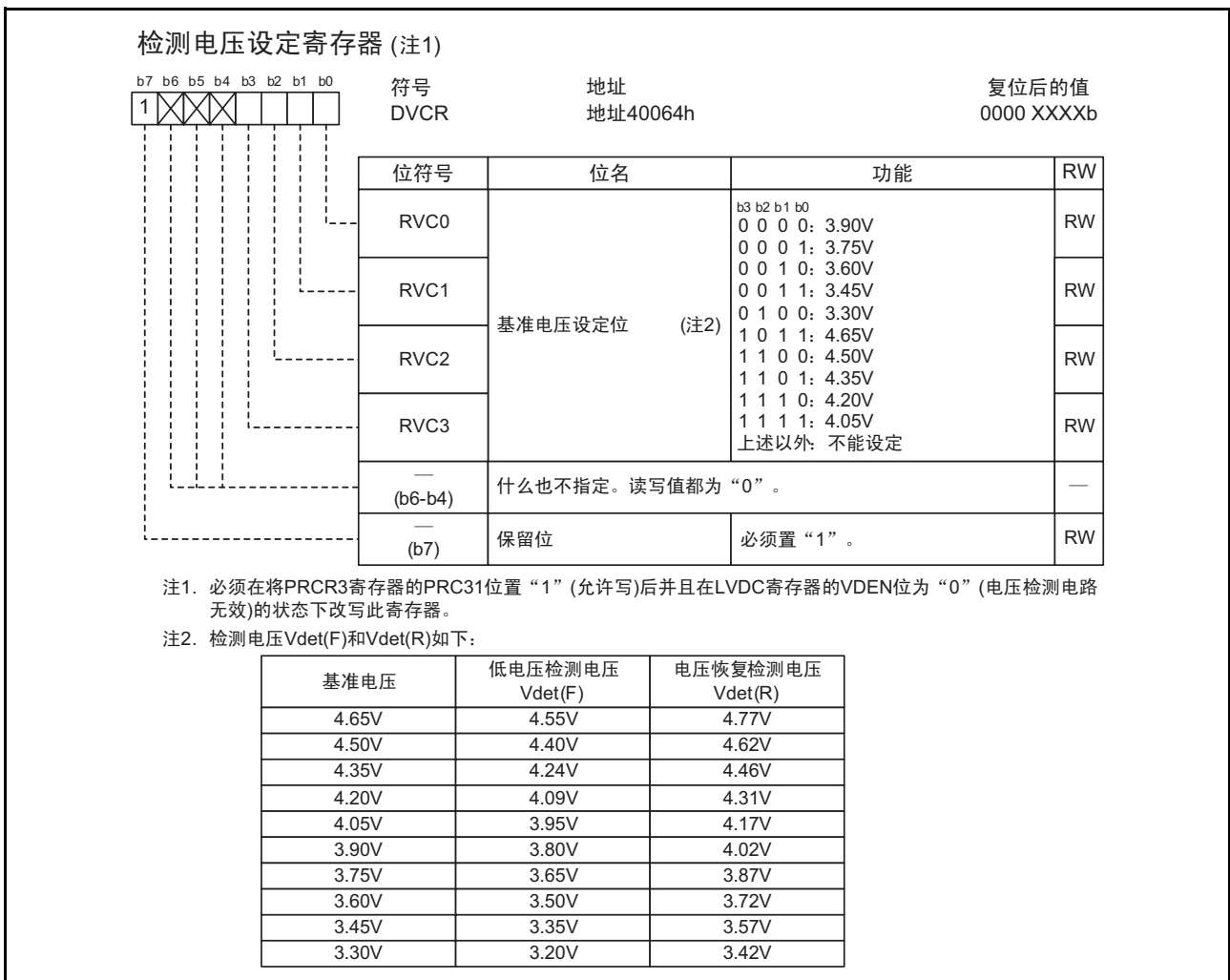


图 6.5 DVCR 寄存器

### 6.2.1 低电压检测电路的运行

如果将 LVDC 寄存器的 VDEN 位置 “1”（电压检测电路有效），电压检测电路就在经过  $t_{d(E-A)}$  后开始稳定运行。

如果 VCC 引脚的输入电压下降到低于  $V_{det(F)}$ ，VMF 位就变为 “0”（ $V_{CC} < V_{det}$ ），LVDF 位就变为 “1”（检测到经过检测电压（ $V_{det}$ ））。此时，如果 LVDIEN 位为 “1”（允许低电压检测中断），就产生中断请求。必须通过程序将 LVDF 位置 “0”（未检测到）。

如果 VCC 引脚的输入电压至少重新上升到  $V_{det(R)}$ ，VMF 位就变为 “1”（ $V_{CC} \geq V_{det}$ ），LVDF 位就变为 “1”（检测到经过检测电压（ $V_{det}$ ））。此时，如果 LVDIEN 位为 “1”（允许低电压检测中断），就产生中断请求。

低电压检测电路的运行如图 6.6 所示。

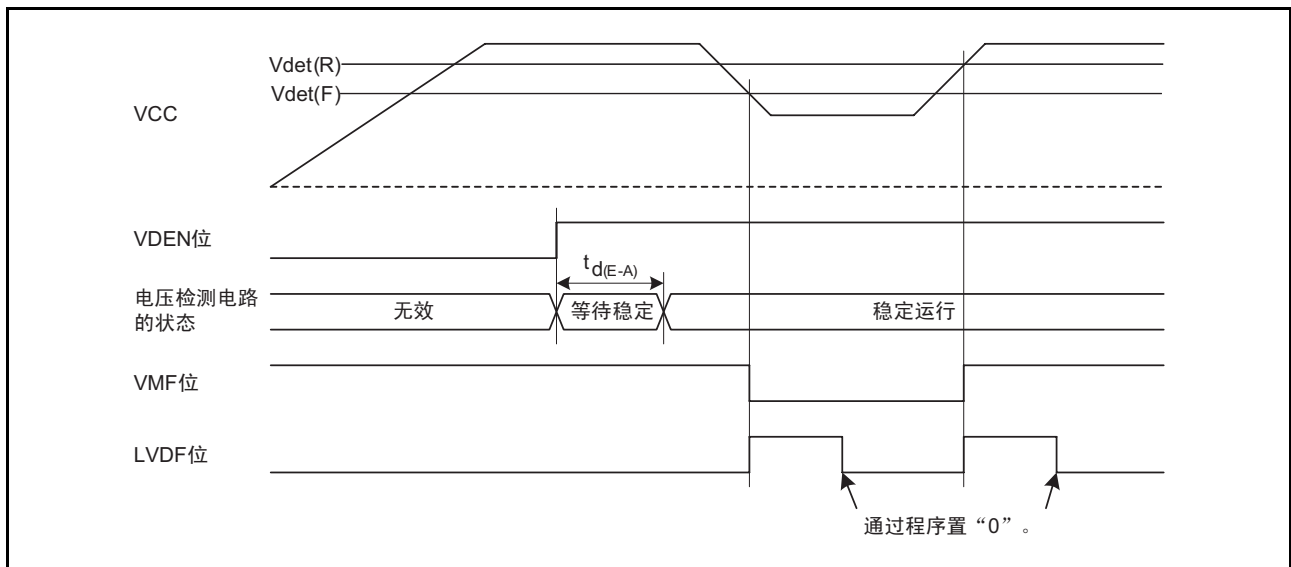


图 6.6 低电压检测电路的运行

### 6.2.2 低电压检测中断

当 LVDC 寄存器的 LVDIEN 位为 “1”（允许低电压检测中断）时，如果 VCC 引脚的输入电压下降到低于  $V_{det(F)}$  或者至少上升到  $V_{det(R)}$ ，就产生低电压检测中断请求。

低电压检测中断与看门狗定时器中断、振荡停止检测中断共用中断向量。在和这些中断同时使用低电压检测中断时，必须通过中断处理程序读 LVDC 寄存器的 LVDF 位，确认已发生低电压检测中断。

在检测到 VCC 引脚的输入电压上升或者下降并经过  $V_{det}$  时，LVDF 位变为 “1”。如果 LVDF 位从 “0” 变为 “1”，就产生低电压检测中断请求。必须通过程序将 LVDF 位置 “0”（未检测到）。

### 6.2.3 低电压检测电路的应用例子

低电压检测中断的应用例子如图 6.7 所示。

通过内部电压稳压器降低 VCC 引脚的输入电压，生成内部逻辑电压。因此，在 VCC 电压开始下降时，内部逻辑电压稳定，但是如果 VCC 电压还继续下降，内部逻辑电压就会开始下降，所以无法保障 MCU 的运行。因此，VCC 电压开始下降后到内部逻辑电压开始下降前的时间为系统能安全关机的时间。低电压检测中断能用于检测到此 VCC 电压开始下降。

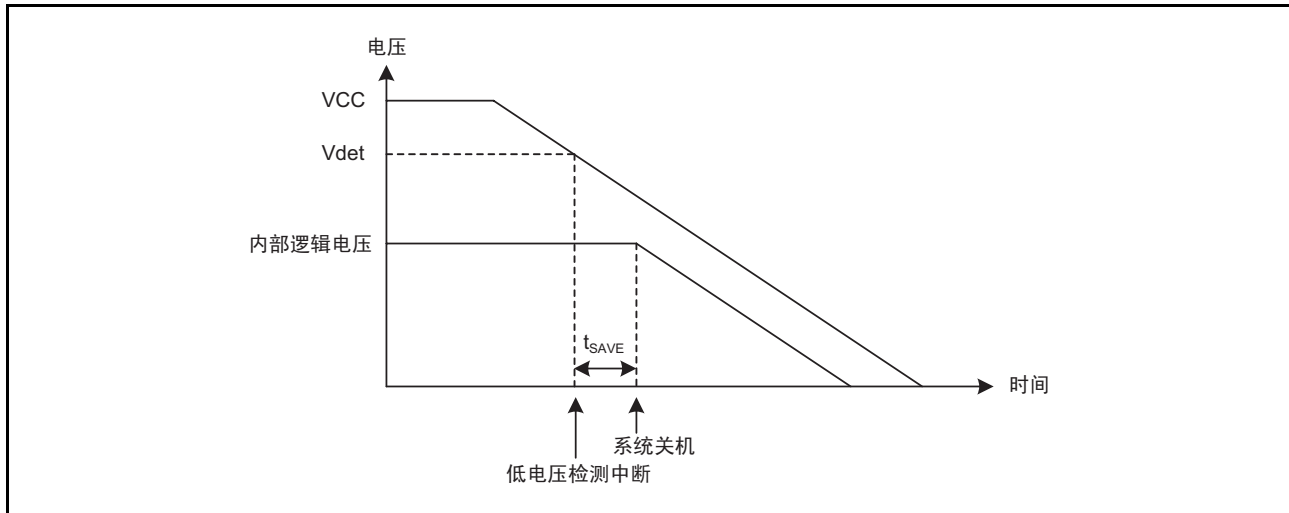


图 6.7 低电压检测中断的应用例子

## 7. 处理器模式

### 7.1 处理器模式的种类

R32C/100 系列的处理器模式有单芯片模式、存储器扩展模式和微处理器模式共 3 种。各处理器模式的特点如表 7.1 所示。

表 7.1 各处理器模式的特点

处理器模式	能存取的区域	分配输入 / 输出端口的引脚
单芯片模式	SFR、内部 RAM、内部 ROM	全部引脚能用作输入 / 输出端口或者外围功能输入 / 输出。
存储器扩展模式	SFR、内部 RAM、内部 ROM、外部区域	将部分引脚用作总线控制引脚（注 1）。
微处理器模式	SFR、内部 RAM、外部区域	将部分引脚用作总线控制引脚（注 1）。

注 1. 详细内容请参照“9. 总线”。

对于 R32C/117 群，标准上支持上述中的单芯片模式和存储器扩展模式，微处理器模式是选项。在使用时，请向本公司的营业窗口询问。

### 7.2 处理器模式的设定

通过 CNVSS 引脚和 PM0 寄存器的 PM01 ~ PM00 位设定处理器模式。

如表 7.2 所示，在进行硬件复位后，根据 CNVSS 引脚的状态，以单芯片模式或者微处理器模式开始运行。

表 7.2 进行硬件复位后的处理器模式

CNVSS 引脚的输入电平（注 1）	处理器模式
“L”电平	以单芯片模式开始运行。
“H”电平	以微处理器模式开始运行。

注 1. 必须通过电阻将 CNVSS 引脚连接到 VCC 或者 VSS。

在以单芯片模式开始运行后，如果将 PM01 ~ PM00 位改写为“01b”（存储器扩展模式），就转换为存储器扩展模式。

在以任意一个处理器模式开始运行后，如果改写 PM01 ~ PM00 位，就转换为对应设定值的模式，但是如果以微处理器模式开始运行，就与 PM01 ~ PM00 位的设定无关，不能存取内部 ROM。

在更改处理器模式时，必须注意以下几点：

- 注 1. 在将 PM01 ~ PM00 位改写为“01b”（存储器扩展模式）或者“11b”（微处理器模式）时，不能同时改写 PM07 ~ PM02 位。
- 注 2. 在改写 PM02 ~ PM07 位时，不能改变 PM01 ~ PM00 位。
- 注 3. 在执行内部 ROM 的程序时，不能转移到微处理器模式。
- 注 4. 在执行外部区域的程序时，不能转移到单芯片模式。
- 注 5. 在和内部 ROM 重叠区域执行程序时，不能转移到存储器扩展模式。

PM0 寄存器和各处理器模式的存储器分配图分别如图 7.1 和图 7.2 所示。

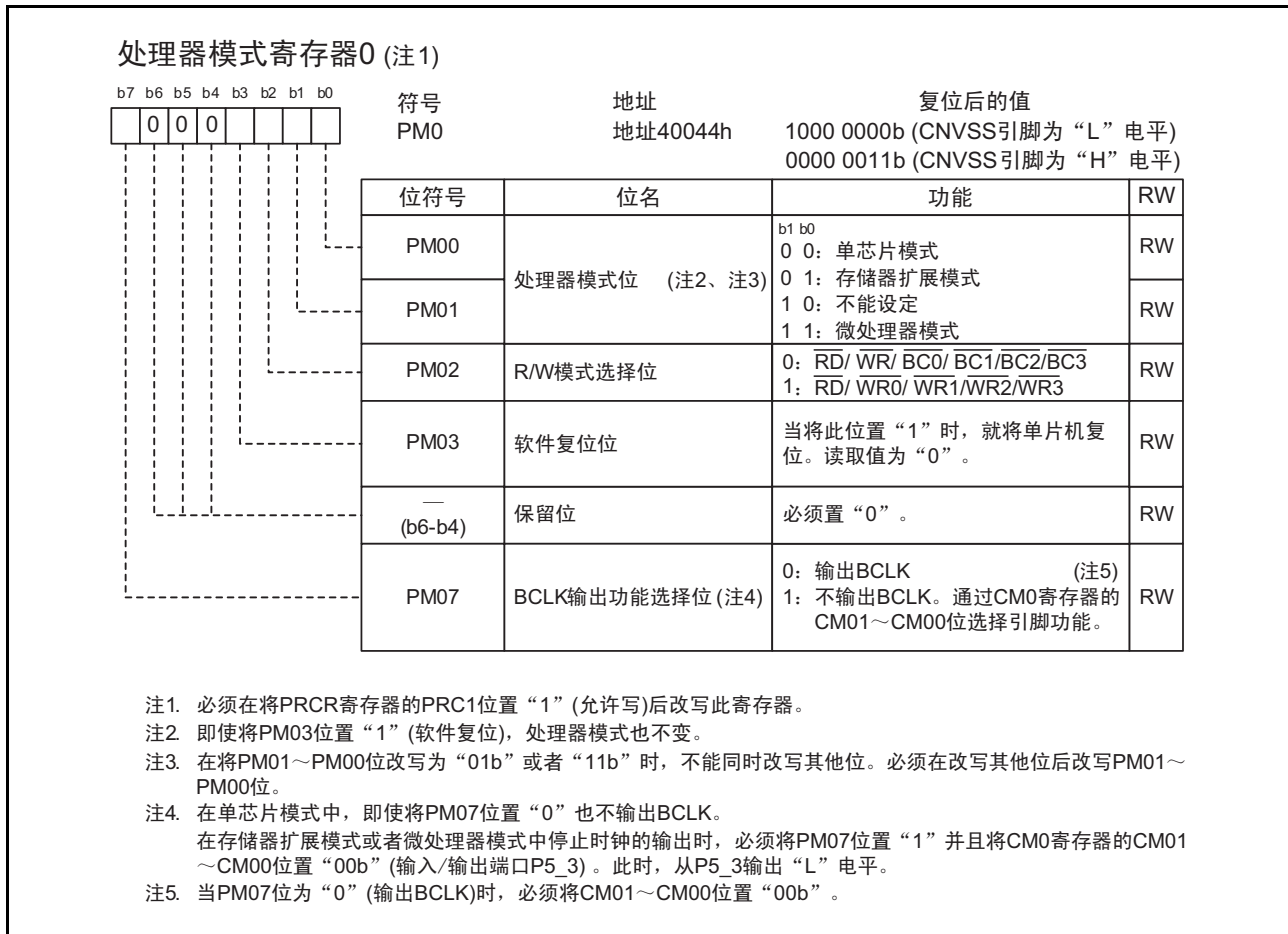


图 7.1 PM0 寄存器

	单芯片模式	存储器扩展模式	微处理器模式
00000000h	SFR	SFR	SFR
00000400h	内部RAM	内部RAM	内部RAM
00008000h	保留区 (内部RAM)	保留区 (内部RAM)	保留区 (内部RAM)
00040000h	SFR2	SFR2	SFR2
00050000h	保留区	保留区	保留区
00060000h	数据ROM	数据ROM	数据ROM
00062000h	保留区 (内部ROM)	保留区 (内部ROM)	保留区 (内部ROM)
00080000h	不可使用 (注1)	外部区域 31.5MB	外部区域 31.5MB
02000000h		不可使用 (注2)	不可使用 (注2)
FE000000h		外部区域 30MB	外部区域 32MB
FFE00000h	保留区 (内部ROM)		
FFF80000h	内部ROM	保留区 (内部ROM)	
FFFFFFFFh		内部ROM	

注1. 在单芯片模式中，不能进行外部扩展。  
注2. 与处理器模式无关，禁止使用。

图 7.2 各处理器模式的存储器分配

## 8. 时钟发生电路

### 8.1 时钟发生电路的种类

R32C/117 群内置了 4 种时钟发生电路：

- 主时钟振荡电路
- 副时钟振荡电路
- PLL 频率合成器
- 内部振荡器

时钟发生电路的概略规格如表 8.1 所示，时钟发生电路的框图和时钟控制的相关寄存器分别如图 8.1 和图 8.2 ~ 图 8.10 所示。

表 8.1 时钟发生电路的概略规格

项目	主时钟 振荡电路	副时钟 振荡电路	PLL 频率 合成器	内部振荡器
用途	PLL 基准时钟源 外围功能的时钟源	CPU 的时钟源 定时器 A 和定时器 B 的时钟源	CPU 的时钟源 外围功能的时钟源	CPU 的时钟源 定时器 A 和定时器 B 的时钟源
时钟频率	4MHz ~ 16MHz	32.768kHz	$f_{SO(PLL)}$ 或者 $f_{(PLL)}$	约 125kHz
能连接的振荡器或者附加电路	陶瓷谐振器 晶体振荡器	晶体振荡器	—	—
振荡器或者附加电路的连接引脚	XIN、XOUT	XCIN、XCOUT	—	—
振荡停止 / 再振荡功能	有	有	有	有
复位后的状态	振荡	停止	振荡	停止
其他	能输入外部生成的时钟。	能输入外部生成的时钟。	在主时钟停止振荡时，以 PLL 频率合成器的自振荡频率 $f_{SO(PLL)}$ 振荡。	

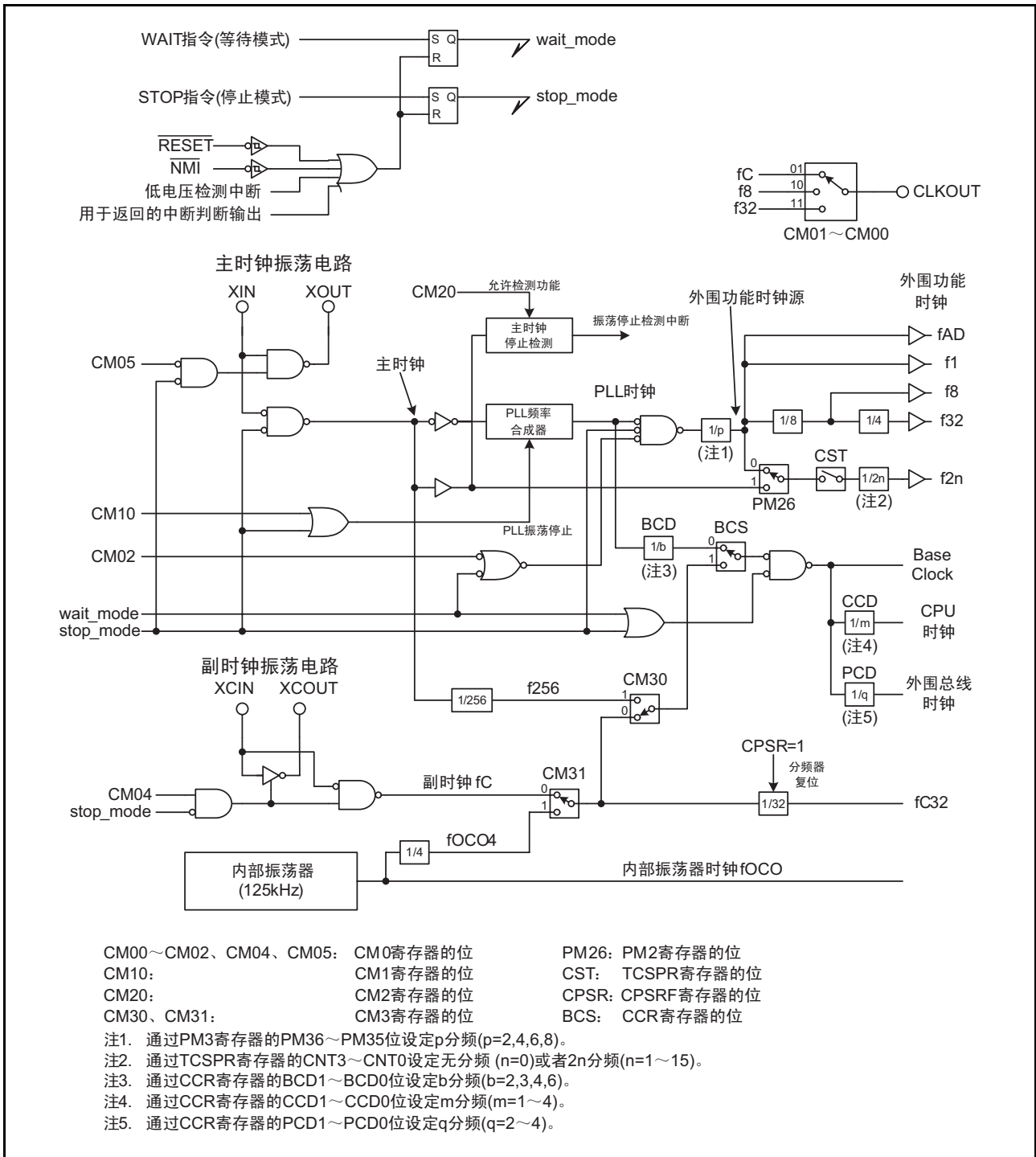


图 8.1 时钟发生电路的框图



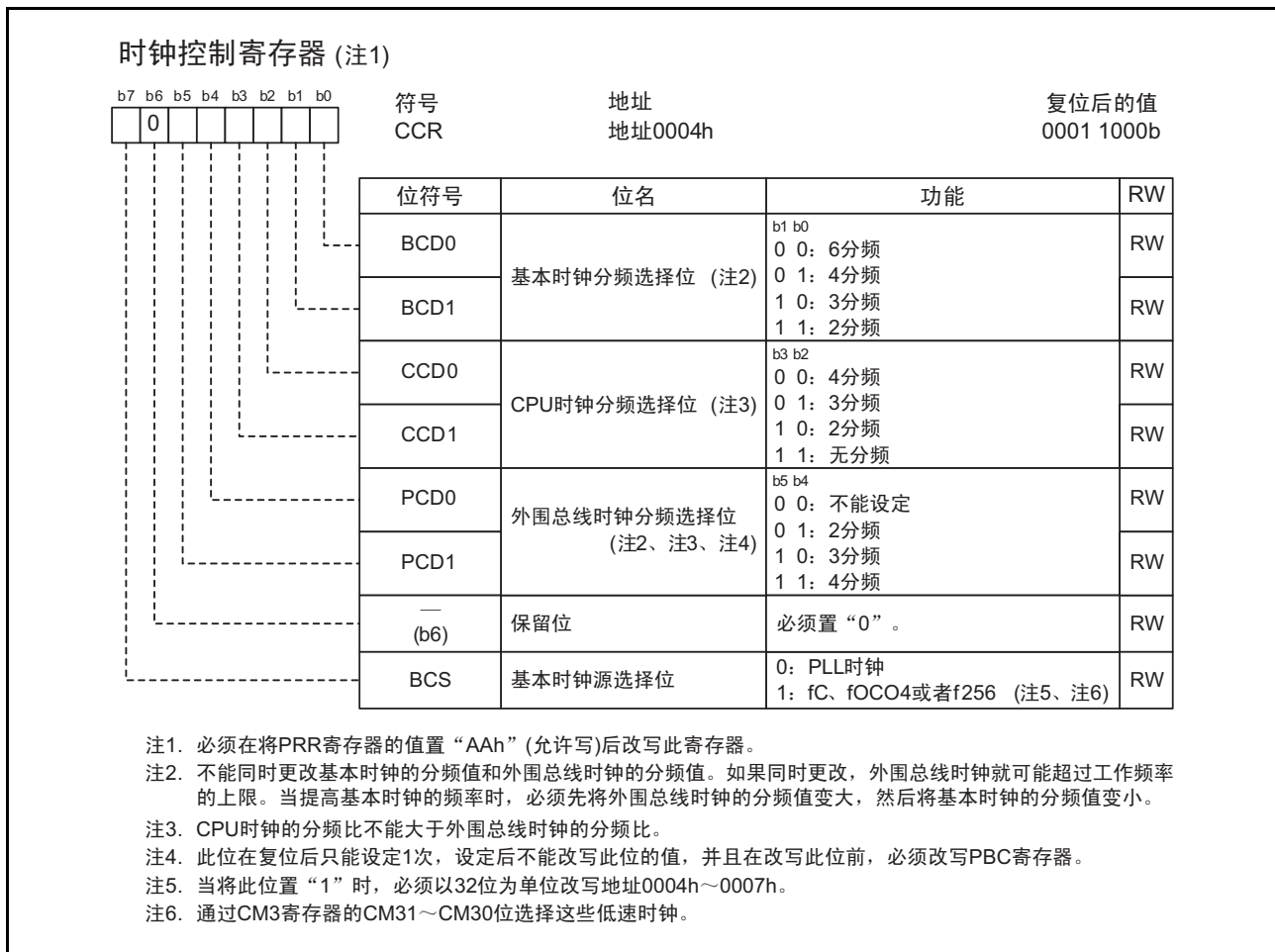


图 8.2 CCR 寄存器

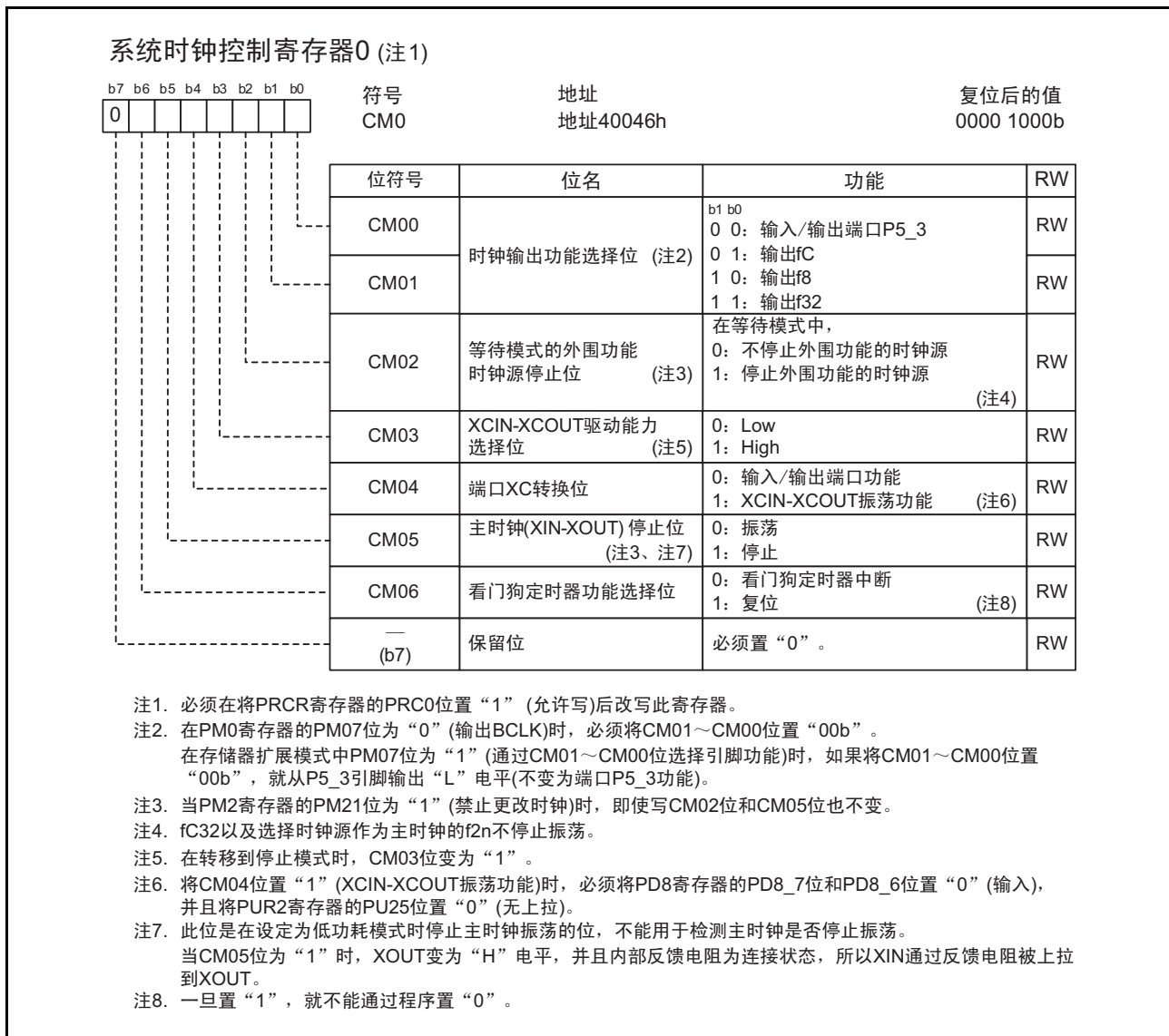


图 8.3 CM0 寄存器

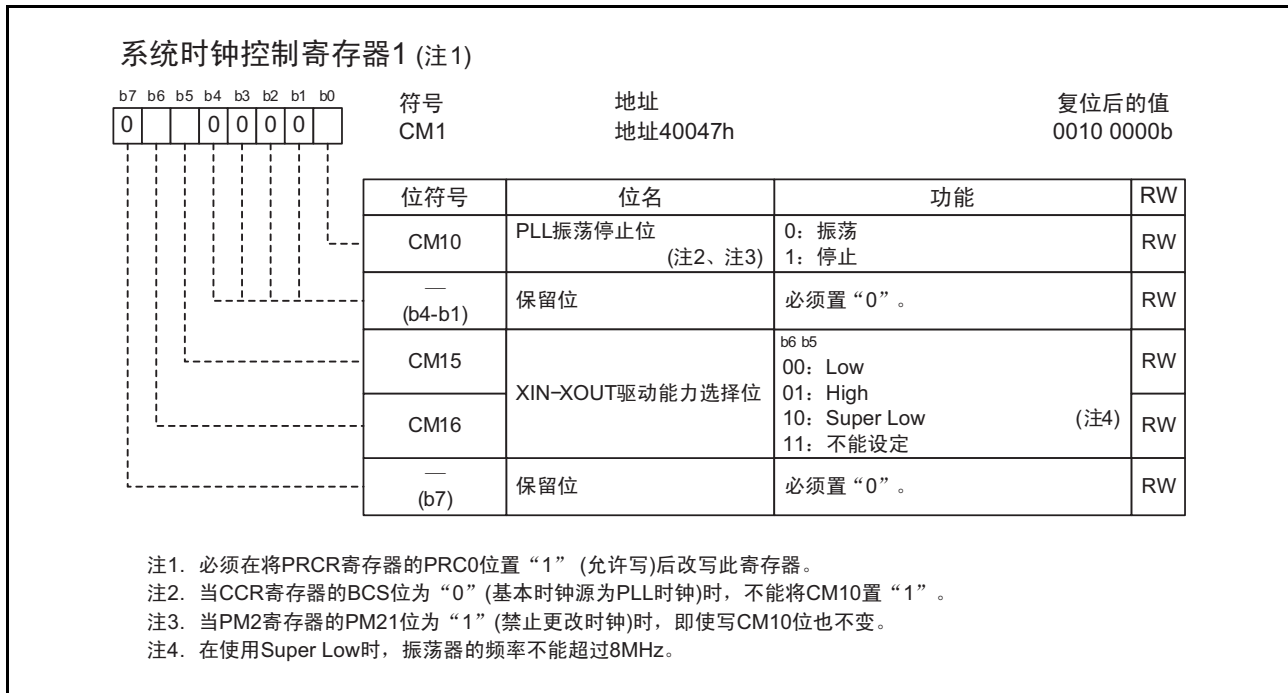


图 8.4 CM1 寄存器

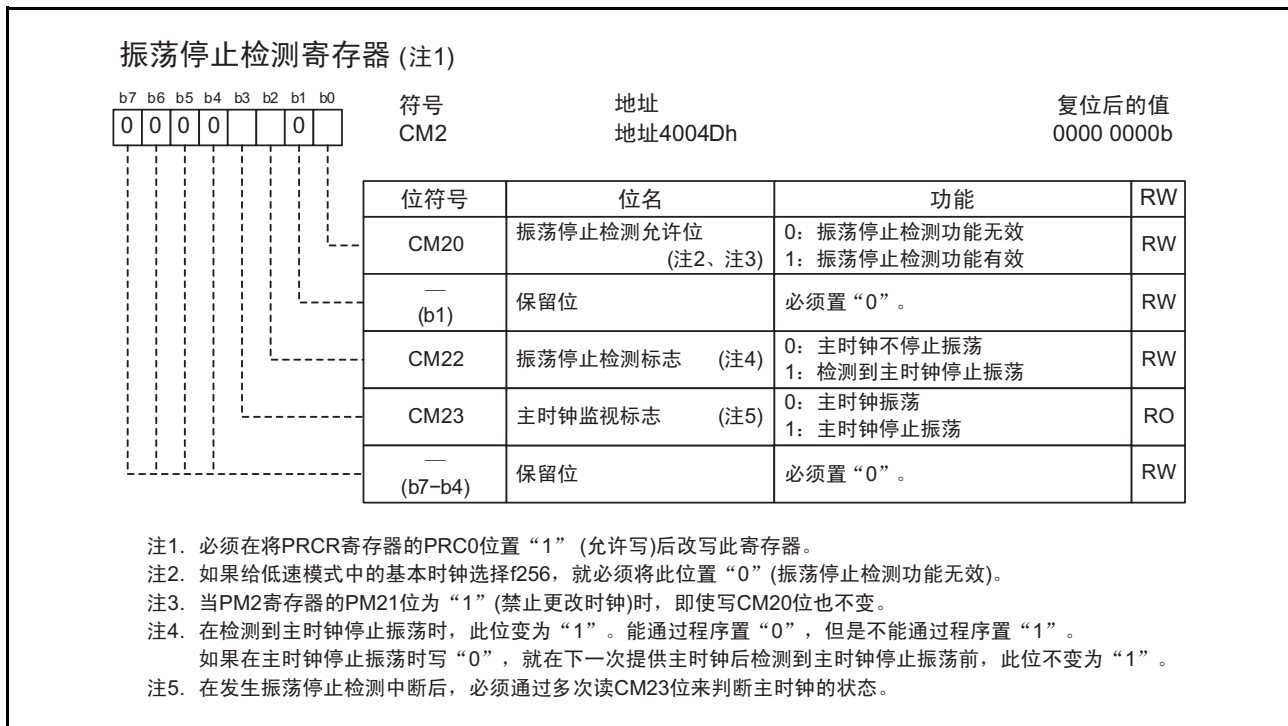


图 8.5 CM2 寄存器

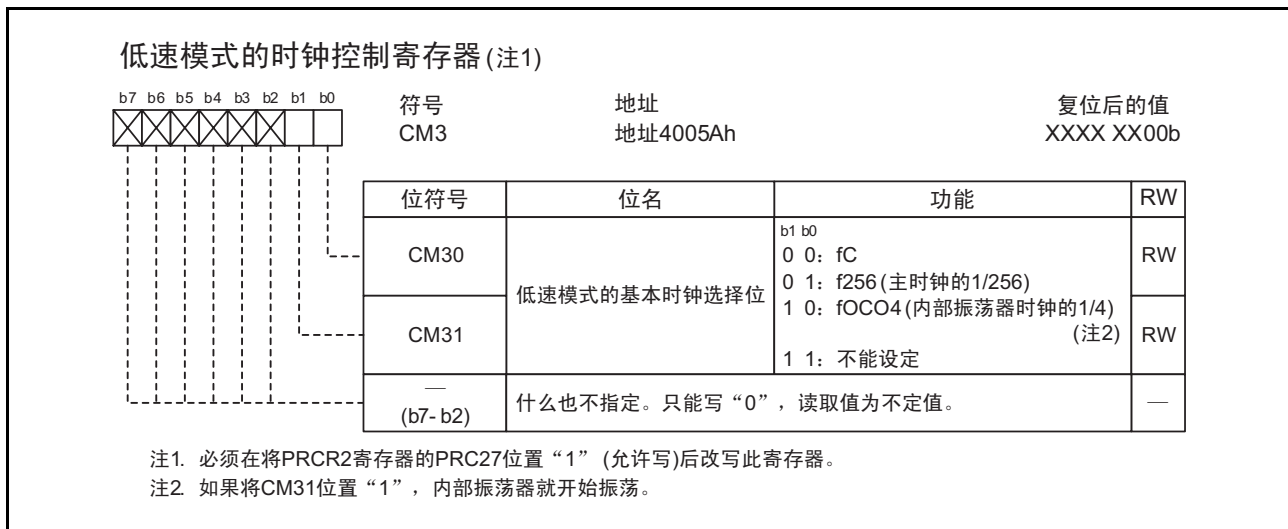


图 8.6 CM3 寄存器

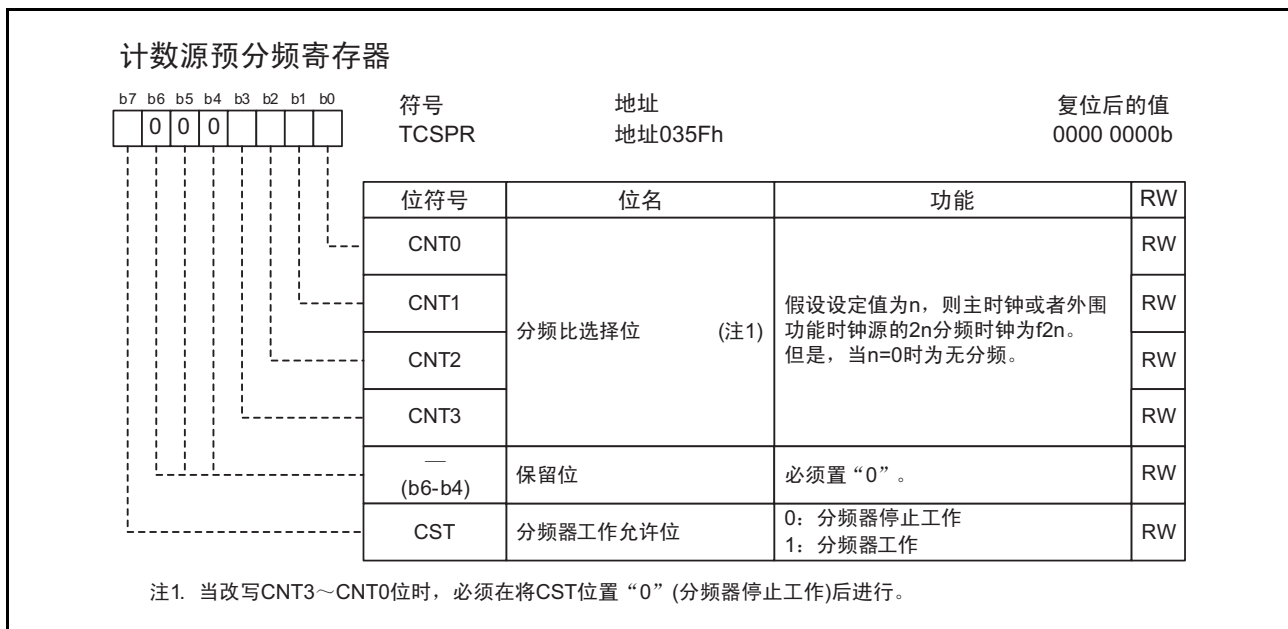


图 8.7 TCSPR 寄存器

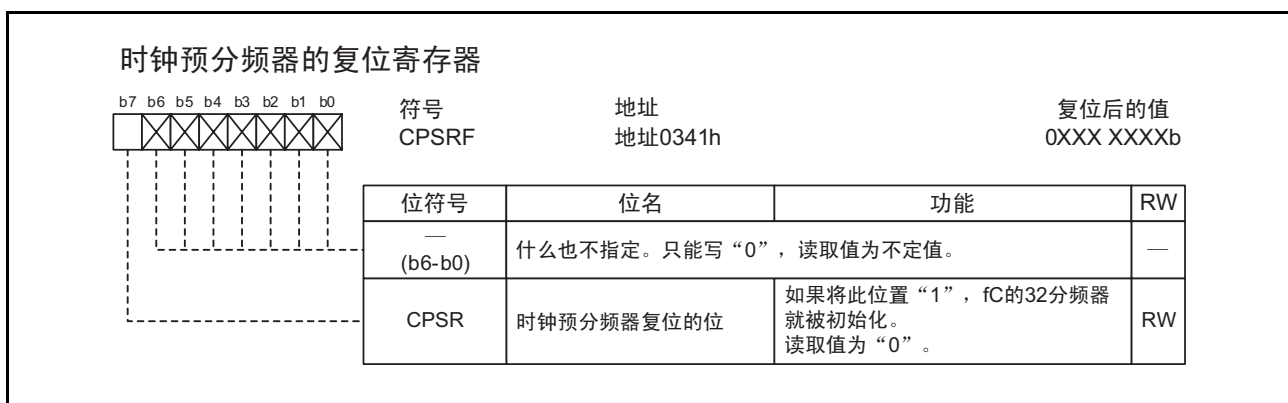


图 8.8 CPSRF 寄存器

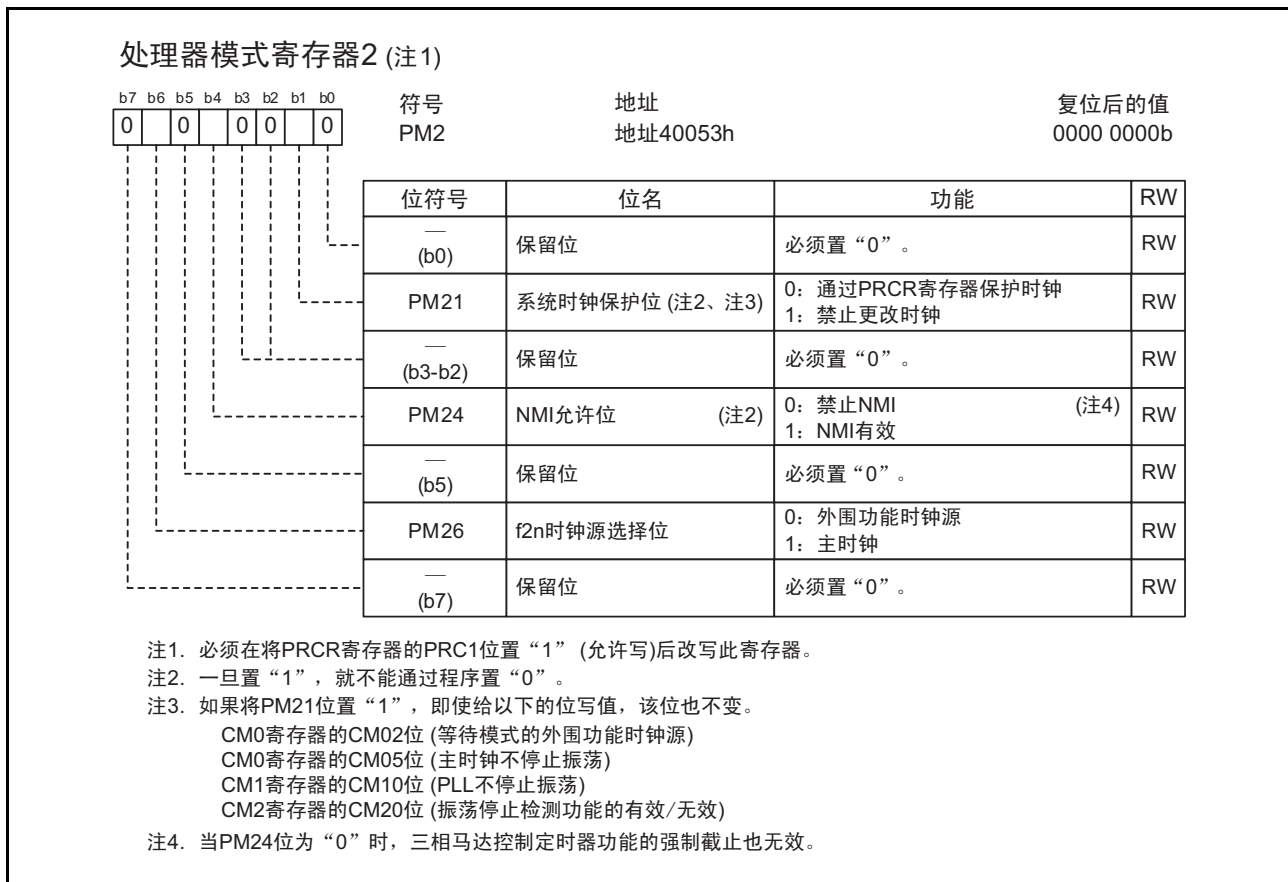


图 8.9 PM2 寄存器

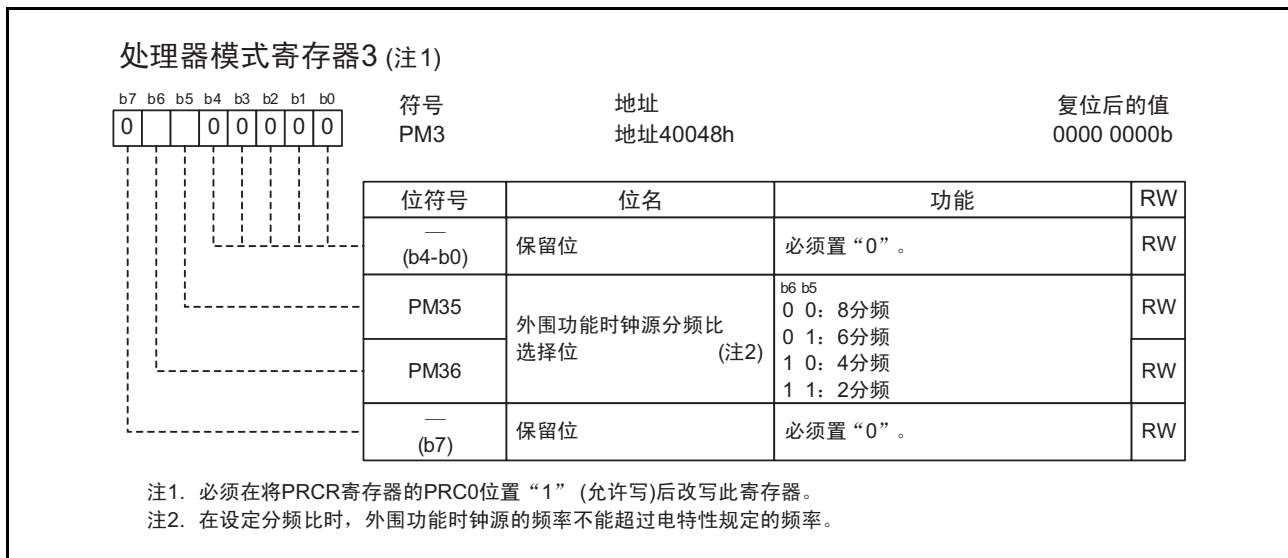


图 8.10 PM3 寄存器

以下说明时钟发生电路生成的时钟。

### 8.1.1 主时钟

主时钟是主时钟振荡电路提供的时钟，是 PLL 基准时钟和外围功能时钟的时钟源。另外，也能用作 CAN 模块的运行时钟。

主时钟振荡电路通过在 XIN-XOUT 引脚之间连接振荡器而构成。主时钟振荡电路内置反馈电阻，为了在停止模式中降低功耗，将反馈电阻从振荡电路分开。对于主时钟振荡电路，也能将外部生成的时钟输入到 XIN 引脚。主时钟的连接电路例子如图 8.11 所示。因为各振荡器的电路常数不同，所以必须设定振荡器厂商的推荐值。

在复位后，虽然主时钟振荡电路在工作，但是因 PLL 频率合成器的输入而被切断，所以给 CPU 提供 PLL 频率合成器自振荡频率的 12 分频时钟。

如果将 CM0 寄存器的 CM05 位置“1”（主时钟振荡电路停止振荡），就能降低功耗。此时，XOUT 引脚变为“H”电平。因为 XIN 引脚和 XOUT 引脚通过芯片内部的反馈电阻连接，所以 XIN 引脚通过反馈电阻被上拉到 XOUT 引脚。另外，在将外部生成的时钟输入到 XIN 引脚时，不能将 CM05 位置“1”。

在停止模式中，包括主时钟在内的全部时钟都停止振荡，详细内容请参照“8.7 功率控制”。

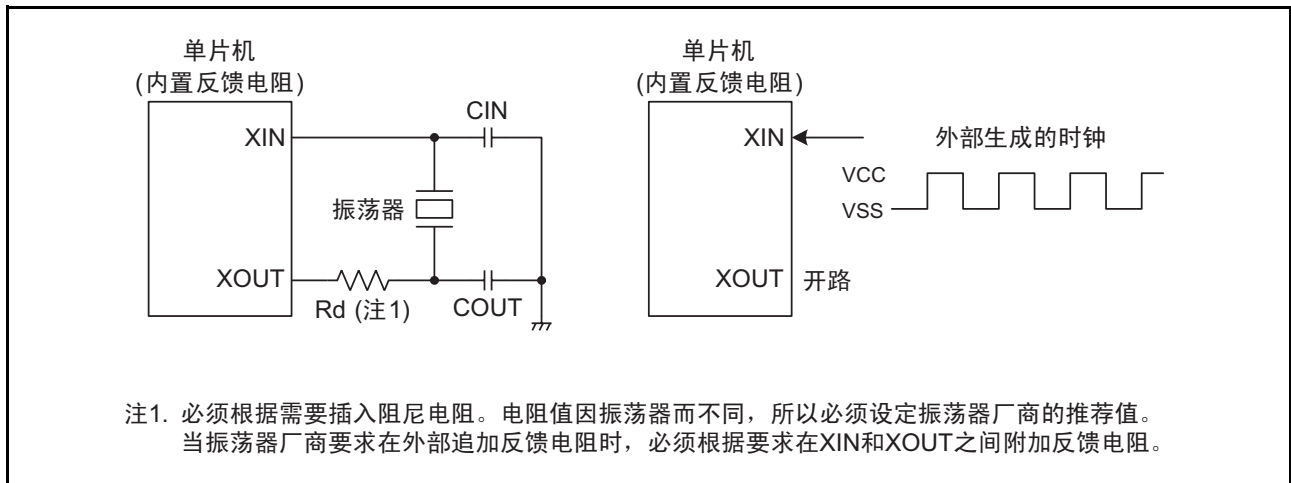


图 8.11 主时钟的连接电路例子

### 8.1.2 副时钟 (fC)

副时钟是副时钟振荡电路提供的时钟，是 CPU 时钟、定时器 A 和定时器 B 计数源的时钟源，而且能从 CLKOUT 引脚输出到外部。

副时钟振荡电路是通过在 XCIN-XCOUT 引脚之间连接晶体振荡器而构成。副时钟振荡电路内置反馈电阻，为了在停止模式中降低功耗，将反馈电阻从振荡电路分开。对于副时钟振荡电路，能将外部生成的时钟输入到 XCIN 引脚。副时钟的电路连接例子如图 8.12 所示。因为各振荡器的电路常数不同，所以必须设定振荡器厂商的推荐值。

在复位后，副时钟停止振荡。此时，反馈电阻已从振荡电路分开。在将 PD8 寄存器的 PD8\_6 位和 PD8\_7 位置“0”（输入模式）并且将 PUR2 寄存器的 PU25 位置“0”（不上拉）后，如果将 CM0 寄存器的 CM04 位置“1”（XCIN-XCOUT 振荡功能），副时钟振荡电路就开始振荡。在将外部生成的时钟输入到 XCIN 引脚时，如果在将 PD8\_7 位和 PU25 位置“0”后将 CM04 位置“1”，输入到 XCIN 引脚的时钟就变为副时钟源。

在副时钟振荡稳定后，如果给 CM3 寄存器写“00h”（fC）并且将 CCR 寄存器的 BCS 位置“1”（fC、fCO04 或者 f256），副时钟就变为 CPU 和外围总线的基本时钟。

在停止模式中，包括副时钟在内的全部时钟都停止振荡，详细内容请参照“8.7 功率控制”。

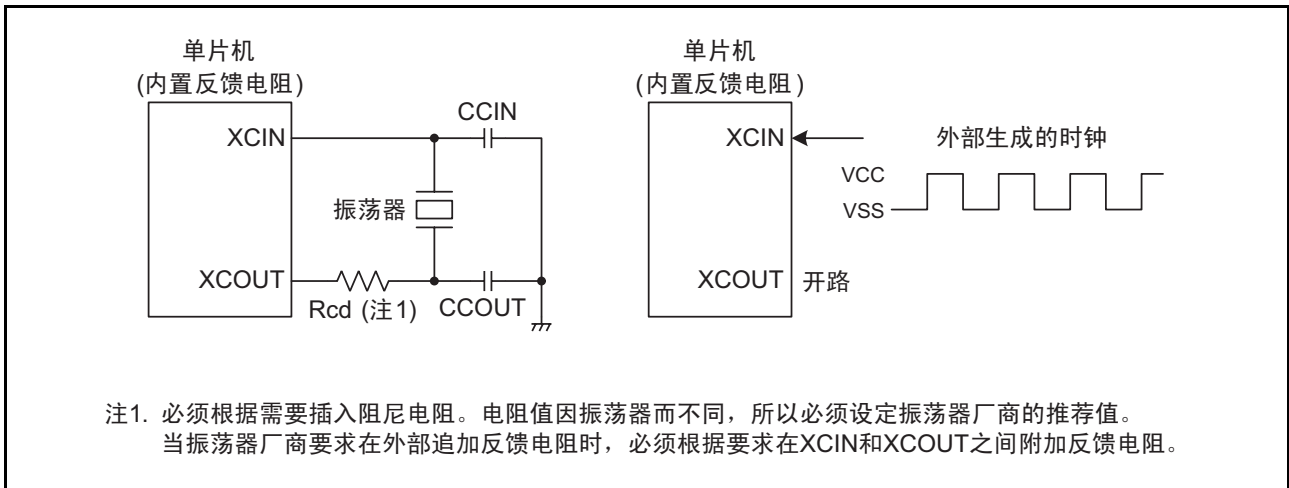


图 8.12 副时钟的连接电路例子

### 8.1.3 PLL 时钟

PLL 时钟是 PLL 频率合成器以主时钟为基准生成的时钟, 用作 CPU 时钟和外围功能时钟等全部时钟的时钟源。

PLL 频率合成器的框图、PLC0 寄存器和 PLC1 寄存器分别如图 8.13、图 8.14 和图 8.15 所示。

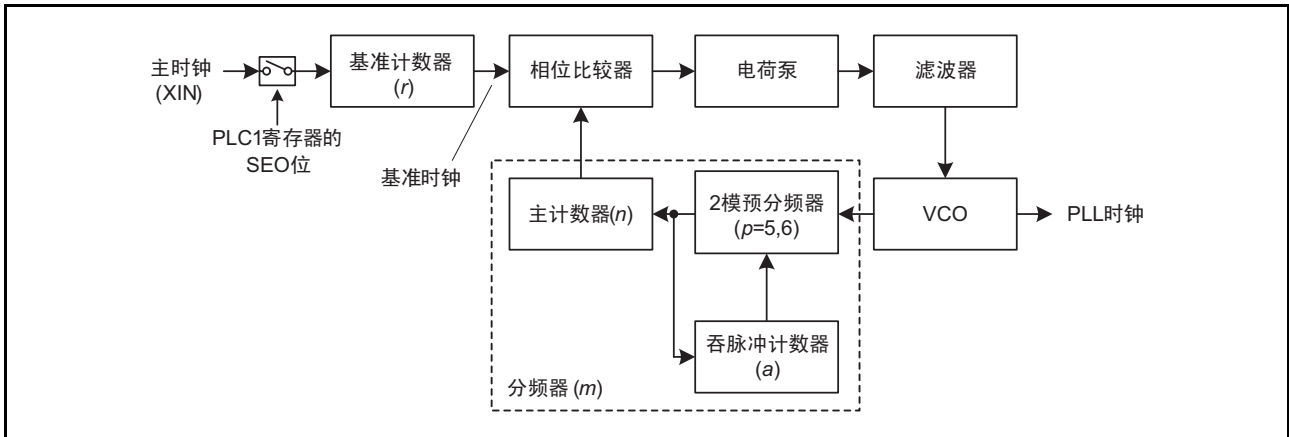


图 8.13 PLL 频率合成器的框图

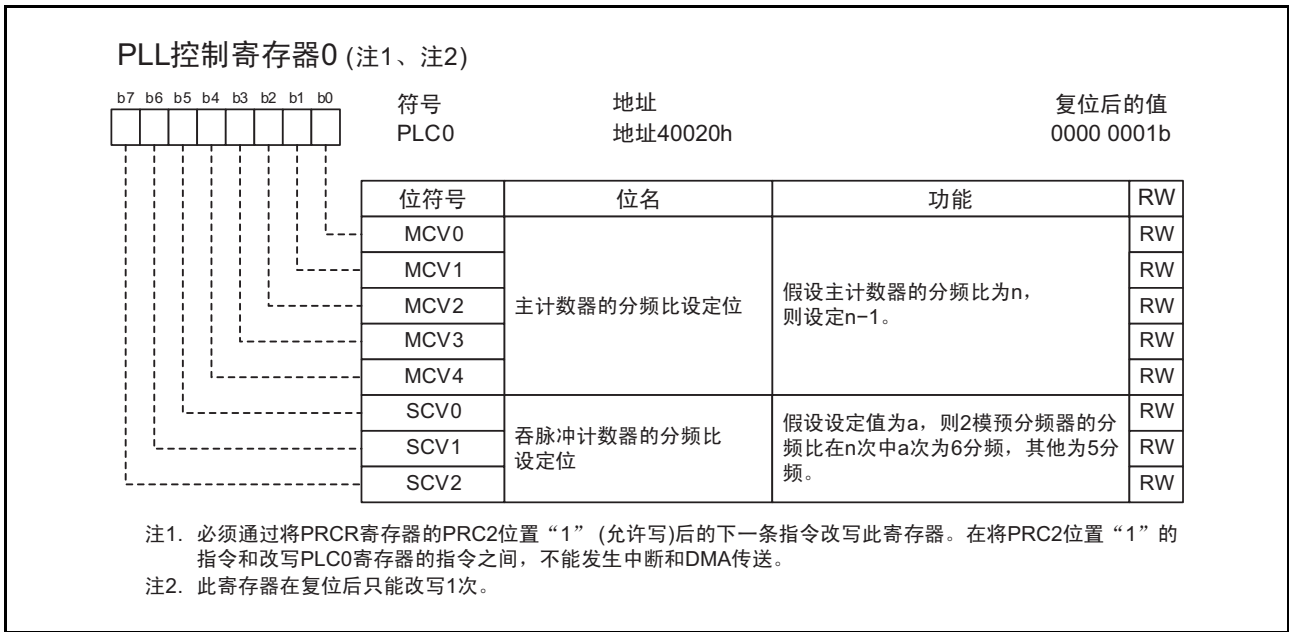


图 8.14 PLC0 寄存器

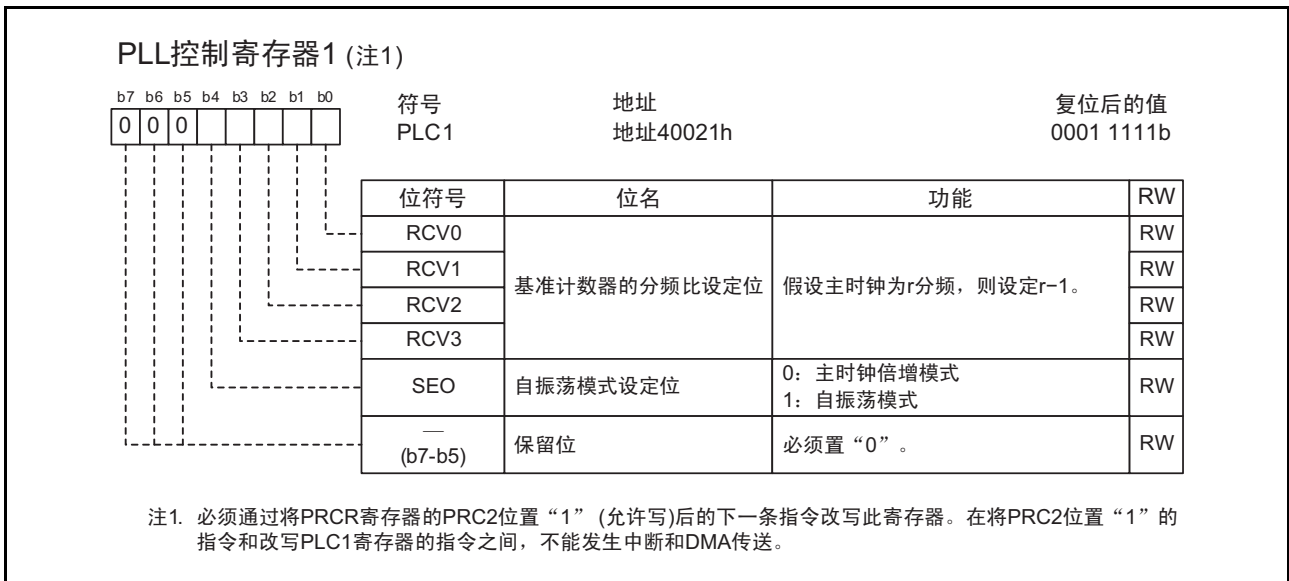


图 8.15 PLC1 寄存器

PLL 频率合成器为吞脉冲方式, 分频比  $m$  能表示为  $n \times p$ 。但是通过吞脉冲计数器的功能,  $p$  的分频比在  $n$  次中  $a$  次为 6 分频, 其他为 5 分频。因此, 实际的分频比  $m$  能表示为:

$$\begin{aligned}
 m &= n \times p \\
 &= n \times \left( \frac{a}{n} \cdot 6 + \frac{n-a}{n} \cdot 5 \right) \\
 &= 5n + a
 \end{aligned}$$

但是,  $a$  能设定的范围为  $0 \leq a < 5$ ,  $0 \leq a \leq n$ 。

对基准计数器进行  $r$  分频, 则相应的 PLL 时钟频率为主时钟 (XIN) 频率的  $m/r$  倍。



$$\begin{aligned} \text{PLL时钟频率}(PLL) &= \frac{m}{r} \cdot \text{主时钟频率} \\ &= \frac{5n+a}{r} \cdot \text{主时钟频率} \end{aligned}$$

在复位后，基准计数器为 16 分频，PLL 频率合成器增加 10 倍。因为没有连接作为基准时钟的主时钟，所以以其固有的频率  $f_{\text{SO(PLL)}}$  自振荡。

为了满足  $2\text{MHz} \leq$  将主时钟  $r$  分频后的基准时钟的频率  $\leq 4\text{MHz}$ ，并且使分频比  $m$  为 25 ~ 100，需要设定各寄存器的值。

PLC1 ~ PLC0 寄存器的设定值如表 8.2 所示，请使用此表中记载的设定值。在主时钟振荡稳定的状态下，从更改 PLC1 ~ PLC0 寄存器的设定后到 PLL 时钟振荡稳定为止，需要的等待时间为  $t_{\text{LOCK(PLL)}}$ 。

表 8.2 PLC1 ~ PLC0 寄存器的设定值（注 1）

主时钟	$r$	基准时钟	$n$	$a$	$m$	PLC1 设定值	PLC0 设定值	$m/r$	PLL 时钟
4MHz	2	2MHz	9	3	48	01h	68h	24	96MHz
6MHz	2	3MHz	6	2	32	01h	45h	16	96MHz
8MHz	3	2.6667MHz	7	1	36	02h	26h	12	96MHz
10MHz	5	2MHz	9	3	48	04h	68h	9.6	96MHz
12MHz	4	3MHz	6	2	32	03h	45h	8	96MHz
16MHz	5	3.2MHz	6	0	30	04h	05h	6	96MHz
4MHz	1	4MHz	5	0	25	00h	04h	25	100MHz
6MHz	3	2MHz	10	0	50	02h	09h	16.6667	100MHz
8MHz	2	4MHz	5	0	25	01h	04h	12.5	100MHz
10MHz	3	3.3333MHz	6	0	30	02h	05h	10	100MHz
12MHz	3	4MHz	5	0	25	02h	04h	8.3333	100MHz
16MHz	4	4MHz	5	0	25	03h	04h	6.25	100MHz
4MHz	1	4MHz	6	0	30	00h	05h	30	120MHz
6MHz	2	3MHz	8	0	40	01h	07h	20	120MHz
8MHz	2	4MHz	6	0	30	01h	05h	15	120MHz
10MHz	3	3.3333MHz	7	1	36	02h	26h	12	120MHz
12MHz	3	4MHz	6	0	30	02h	05h	10	120MHz
16MHz	4	4MHz	6	0	30	03h	05h	7.5	120MHz
4MHz	1	4MHz	6	2	32	00h	45h	32	128MHz
6MHz	3	2MHz	12	4	64	02h	8Bh	21.3333	128MHz
8MHz	2	4MHz	6	2	32	01h	45h	16	128MHz
10MHz	5	2MHz	12	4	64	04h	8Bh	12.8	128MHz
12MHz	3	4MHz	6	2	32	02h	45h	10.6667	128MHz
16MHz	4	4MHz	6	2	32	03h	45h	8	128MHz

注 1. 请使用此表中记载的设定值。

### 8.1.4 内部振荡器时钟

内部振荡器时钟是内部振荡器（OCO）提供的时钟，是 CPU 时钟、定时器 A 和定时器 B 计数源的时钟源。振荡频率约为 125kHz，频率为此振荡频率的 1/4 的时钟能用作 CPU 和外围总线的基本时钟。

在复位后，内部振荡器停止振荡。如果将 CM3 寄存器的 CM31 位置“1”，内部振荡器就开始振荡。因为内部振荡器几乎瞬间就开始振荡，所以不需要振荡稳定的等待时间。

## 8.2 振荡停止检测功能

振荡停止检测功能是在主时钟因外部因素而停止振荡时，检测时钟停止的功能。

当 CM2 寄存器的 CM20 位为“1”（振荡停止检测功能有效）时，如果主时钟停止振荡，就产生振荡停止检测中断请求。此时，因为 PLL 频率合成器以固有的振荡频率自振荡，所以在 PLL 时钟用作 CPU 时钟和外围功能时钟的时钟源的情况下，这些时钟不停止振荡。

在检测到振荡停止时，CM2 寄存器的以下的位发生变化：

- CM22 位为“1”（检测到主时钟停止振荡）。
- CM23 位为“1”（主时钟停止振荡）（参照“图 8.17 状态转移图（使用副时钟）”）。

### 8.2.1 振荡停止检测功能的使用方法

振荡停止检测中断和看门狗定时器中断、低电压检测中断共用向量。在同时使用振荡停止检测中断和这些中断时，必须在中断处理程序内读 CM22 位，确认已产生振荡停止检测的中断请求。

在检测到振荡停止后，如果主时钟重新开始振荡，PLL 时钟频率就有可能在 PLL 频率合成器振荡稳定前暂时超过设定频率，所以在检测到振荡停止后，必须通过程序迅速提高基本时钟的分频比（通过 CCR 寄存器的 BCD1 ~ BCD0 位设定）和外围功能时钟源的分频比（通过 PM3 寄存器的 PM36 ~ PM35 位设定）。

在低速模式中，当 CM20 位为“1”（振荡停止检测功能有效）时，如果主时钟停止振荡，就产生振荡停止检测的中断请求。此时，CPU 时钟仍为低速时钟。另外，在低速模式中将 f256（主时钟的 256 分频）时钟用作基本时钟时，不能使用振荡停止检测功能。

此功能是针对外部因素导致主时钟停止振荡而设置的功能，因此在通过程序使主时钟停止振荡，即进入停止模式或者将 CM05 位置“1”（主时钟停止振荡）时，必须将 CM20 位置“0”（振荡停止检测功能无效）。另外，在等待模式中也必须将 CM20 位置“0”。

振荡停止检测功能通过观察电容器的充电电压是否在某个值以下进行判断。实际上，如果主时钟的振荡频率大约低于 500kHz，就认为是振荡停止。因此，如果在中断处理程序内记述了将 CM22 位置“0”的程序，并且主时钟的振荡频率在 500kHz 左右反复上升或者下降，就有可能产生多重中断以及堆栈上溢。

## 8.3 基本时钟

基本时钟是 CPU 时钟和外围总线时钟的基准时钟。在复位后，基本时钟是 PLL 时钟的 6 分频时钟。

能选择 PLL 时钟或者低速时钟作为基本时钟源，选择副时钟（fC）、内部振荡器时钟的 4 分频（fOCO4）时钟或者主时钟的 256 分频（f256）时钟作为低速时钟。

如果选择 PLL 时钟作为基本时钟的时钟源，对所选时钟进行 2,3,4,6 分频后的时钟就为基本时钟；如果选择低速时钟作为基本时钟的时钟源，低速时钟本身就为基本时钟。

能通过 CCR 寄存器的 BCS 位选择基本时钟源，通过 CCR 寄存器的 BCD1 ~ BCD0 位选择 PLL 时钟的分频比，通过 CM3 寄存器的 CM31 ~ CM30 位选择低速时钟。

## 8.4 CPU 时钟和外围总线时钟

CPU 时钟是 CPU 的运行时钟。在复位后，CPU 时钟为基本时钟的 2 分频时钟。

CPU 时钟源是基本时钟，能通过 CCR 寄存器的 CCD1 ~ CCD0 位选择分频比；对基本时钟进行 2 ~ 4 分频后的时钟为外围总线时钟，能通过 CCR 寄存器的 PCD1 ~ PCD0 位选择分频比。

外围总线时钟也用作看门狗定时器的计数源和 CAN 模块的运行时钟。

在存储器扩展模式或者微处理器模式中，作为用于单片机外部时序生成的基准时钟，能从 BCLK 引脚输出外围总线时钟（作为 BCLK），详细内容请参照“8.6 时钟输出功能”。

为了在系统失控时使 CPU 时钟（PLL 时钟作为时钟源）不停止振荡，必须在 CM0 寄存器的 CM05 位为“0”（主时钟振荡）并且 CCR 寄存器的 BCS 位为“0”（基本时钟源为 PLL 时钟）的状态下，进行以下的处理：

1. 将 PRCR 寄存器的 PRC1 位置“1”（允许写 PM2 寄存器）。
2. 将 PM2 寄存器的 PM21 位置“1”（禁止更改时钟）。

## 8.5 外围功能时钟

外围功能时钟是除看门狗定时器和 CAN 模块以外的外围功能的运行时钟或者计数源。通过设定 PM3 寄存器的 PM36 ~ PM35 位，对和 PLL 时钟相同频率的时钟进行 2,4,6,8 分频后的时钟是作为外围功能时钟基准的外围功能时钟源。

外围功能时钟大致能分为以下 3 种：

### (1) f1、f8、f32、f2n

f1、f8、f32 是对外围功能时钟源分别进行 1,8,32 分频后的时钟。能通过 PM2 寄存器的 PM26 位，选择外围功能时钟源或者主时钟作为 f2n 的时钟源；能通过 TCSPPR 寄存器的 CNT3 ~ CNT0 位，设定 f2n 的分频比（n=1 ~ 15，当 n=0 时为无分频）。

如果在将 CM02 位置“1”（在等待模式中停止外围功能的时钟源）后转移到等待模式，或者在低功耗模式中，f1、f8、f32 以及选择外围功能时钟源作为时钟源的 f2n 就停止振荡。

f1、f8、f2n 用作定时器 A 和定时器 B 的计数源以及串行通信接口的运行时钟，f1 还用作智能 I/O 的运行时钟。

能从 CLKOUT 引脚输出 f8 和 f32，详细内容请参照“8.6 时钟输出功能”。

### (2) fAD

fAD 的频率和外围功能时钟源的频率相同，是 A/D 转换器的运行时钟。

如果在将 CM02 位置“1”（在等待模式中停止外围功能的时钟源）后转移到等待模式，或者在低功耗模式中，此时钟就停止振荡。

### (3) fC32

fC32 是副时钟的 32 分频时钟或者内部振荡器时钟的 128 分频时钟，用作定时器 A 和定时器 B 的计数源，能在提供副时钟或者内部振荡器时钟时使用。

## 8.6 时钟输出功能

能从 CLKOUT 引脚输出 fC、f8 或者 f32。

在存储器扩展模式或者微处理器模式中，能从 BCLK 引脚输出基本时钟 2 ~ 4 分频后的外围总线时钟（作为 BCLK）。

单芯片模式的 CLKOUT 引脚功能、存储器扩展模式或者微处理器模式的 CLKOUT 引脚功能分别如表 8.3 和表 8.4 所示。

表 8.3 单芯片模式的 CLKOUT 引脚功能

PM0 寄存器（注 1）	CM0 寄存器（注 2）		CLKOUT 引脚功能
	PM07	CM01	
0 或者 1	0	0	输入 / 输出端口 P5_3
1	0	1	输出 fC
1	1	0	输出 f8
1	1	1	输出 f32

注 1. 必须在将 PRCR 寄存器的 PRC1 位置“1”（允许写）后改写此寄存器。

注 2. 必须在将 PRCR 寄存器的 PRC0 位置“1”（允许写）后改写此寄存器。

表 8.4 存储器扩展模式或者微处理器模式的 CLKOUT 引脚功能

PM0 寄存器（注 1）	CM0 寄存器（注 2）		CLKOUT 引脚功能
	PM07	CM01	
0	0（注 3）	0（注 3）	输出 BCLK
1	0	0	输出“L”电平（不为 P5_3）
1	0	1	输出 fC
1	1	0	输出 f8
1	1	1	输出 f32

注 1. 必须在将 PRCR 寄存器的 PRC1 位置“1”（允许写）后改写此寄存器。

注 2. 必须在将 PRCR 寄存器的 PRC0 位置“1”（允许写）后改写此寄存器。

注 3. 当 PM07 位为“0”（输出 BCLK）时，必须将 CM01 ~ CM00 位置“00b”（输入 / 输出端口 P5\_3）。

## 8.7 功率控制

功率控制有 3 种模式。

在此章节中，将等待模式、停止模式以外的状态称为正常运行模式。等待模式和停止模式的状态转移图、使用副时钟时的状态转移图、使用主时钟的 256 分频时钟时的状态转移图、使用内部振荡器时的状态转移图分别如图 8.16、图 8.17、图 8.18 和图 8.19 所示。

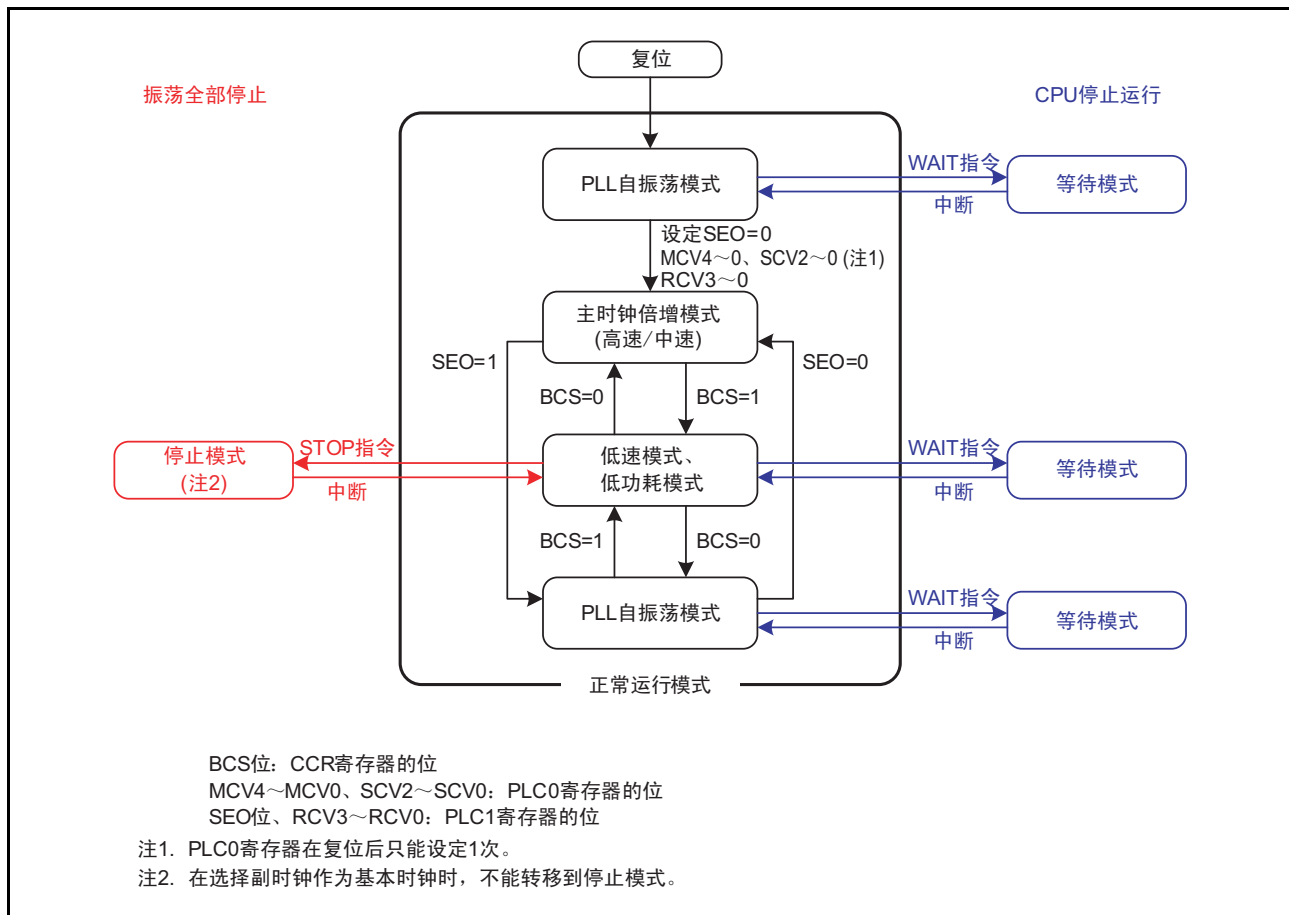


图 8.16 等待模式和停止模式的状态转移图

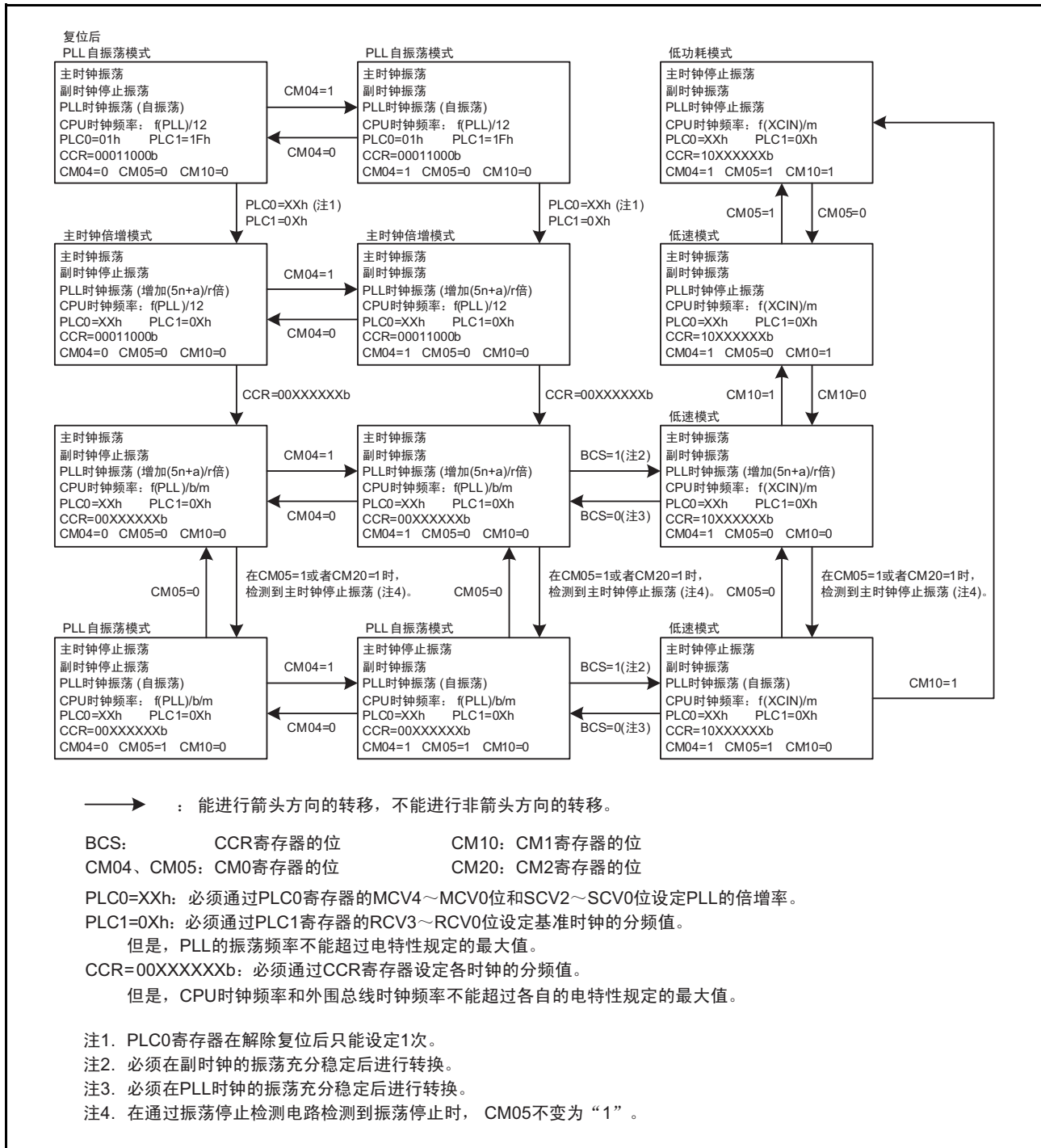


图 8.17 状态转移图 (使用副时钟)

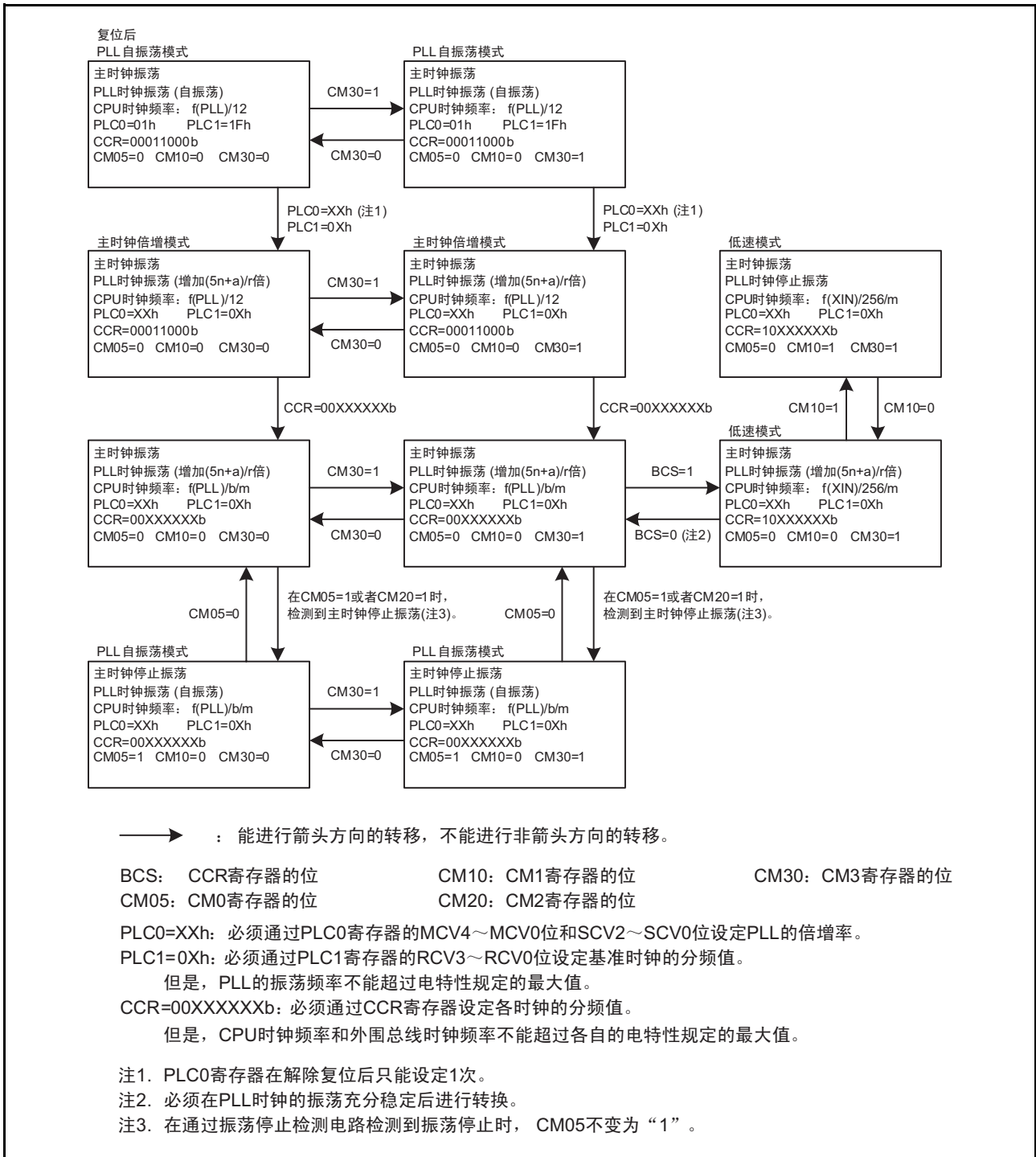


图 8.18 状态转移图 (使用主时钟的 256 分频时钟)

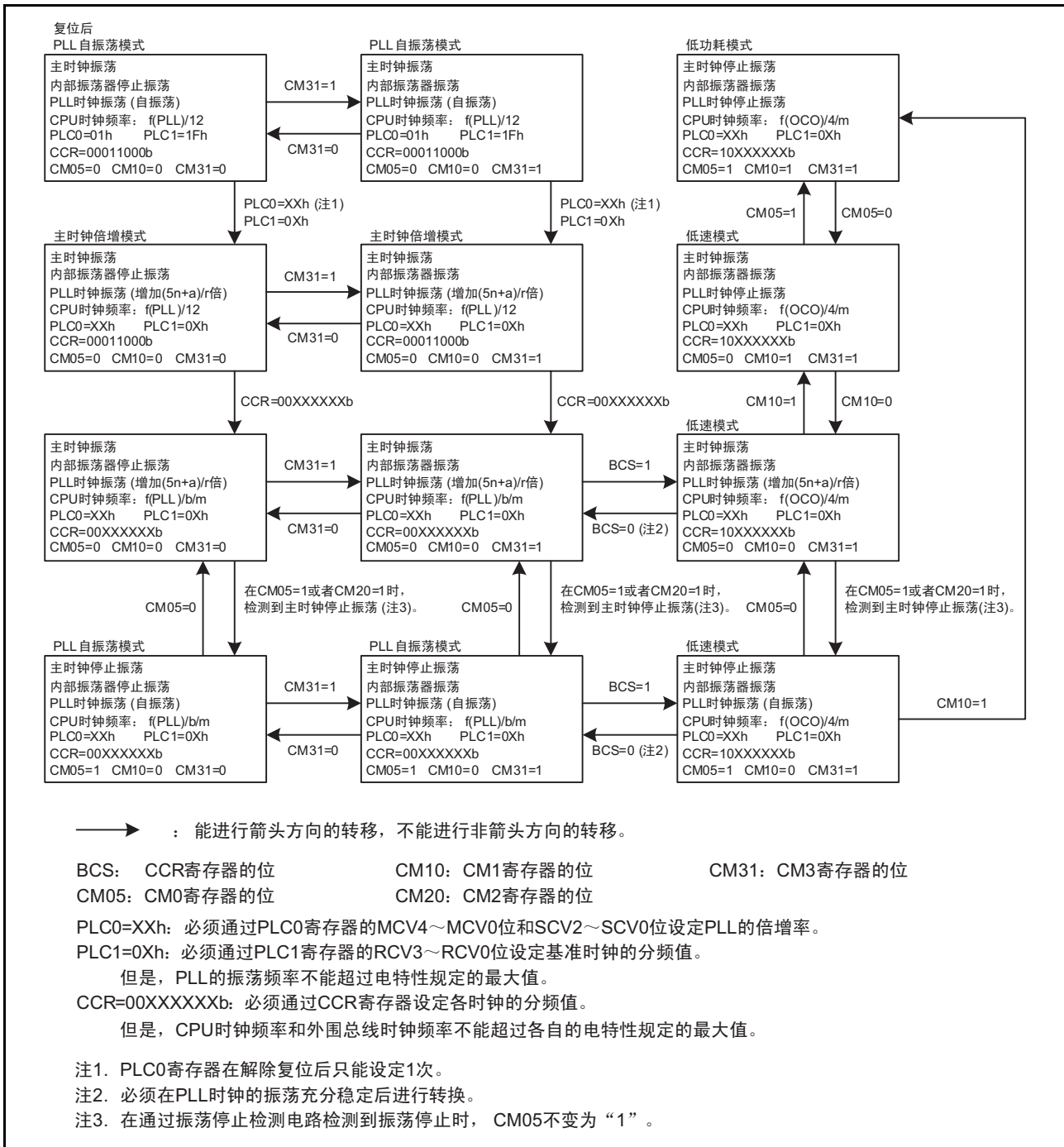


图 8.19 状态转移图 (使用内部振荡器)



### 8.7.1 正常运行模式

正常运行模式又分为以下 5 种模式。

在正常运行模式中，因为提供 CPU 时钟和外围功能时钟，所以 CPU 和外围功能都运行。通过控制 CPU 时钟的频率进行功率控制。CPU 时钟的频率越高，处理能力就越强；频率越低，功耗就越小。如果停止不需要的振荡电路，就能进一步降低功耗。

#### (1) 主时钟倍增模式（高速模式）

这是 CPU 以最高工作频率运行的模式。PLL 时钟的 2 分频时钟为基本时钟，基本时钟的频率和 CPU 时钟的频率相同。能选择 PLL 时钟或者主时钟作为外围功能的时钟源。在提供副时钟或者内部振荡器时钟的情况下，fC32 能用作定时器 A 和定时器 B 的计数源。

#### (2) 主时钟倍增模式（中速模式）

PLL 时钟的 2,3,4,6 分频时钟为基本时钟，基本时钟的 1 ~ 4 分频时钟为 CPU 时钟。能选择 PLL 时钟或者主时钟作为外围功能的时钟源。在提供副时钟或者内部振荡器时钟的情况下，fC32 能用作定时器 A 和定时器 B 的计数源。

#### (3) 低速模式

低速时钟为基本时钟，基本时钟的 1 ~ 4 分频时钟为 CPU 时钟。能选择 PLL 时钟或者主时钟作为外围功能的时钟源。在提供副时钟或者内部振荡器时钟的情况下，fC32 能用作定时器 A 和定时器 B 的计数源。

#### (4) 低功耗模式

这是在进入低速模式后，使主时钟和 PLL 频率合成器处于停止状态的模式。副时钟或者内部振荡器时钟的 4 分频时钟为基本时钟，基本时钟的 1 ~ 4 分频时钟为 CPU 时钟。fC32 能用作定时器 A 和定时器 B 的计数源，外围功能时钟只有 fC32。在此模式中，如果将 VRCR 寄存器的 MRS 位置“1”（主稳压器停止工作），就能进一步减小消耗电流。

#### (5) PLL 自振荡模式

这是使作为 PLL 频率合成器的基准输入的主时钟处于停止状态的模式。PLL 频率合成器以固有的振荡频率自振荡。PLL 时钟的 2,3,4,6 分频时钟为基本时钟，基本时钟的 1 ~ 4 分频时钟为 CPU 时钟。能选择 PLL 时钟或者主时钟作为外围功能的时钟源。在提供副时钟或者内部振荡器时钟的情况下，fC32 能用作定时器 A 和定时器 B 的计数源。

必须在转换目标的时钟振荡稳定后转换 CPU 时钟。尤其对于副时钟，因为振荡稳定需要时间（注 1），所以必须在接通电源后或者从停止模式返回后，通过程序获取等待时间后再进行转换。

注 1. 有关振荡稳定时间，请向各振荡器厂商询问。

## 8.7.2 等待模式

在等待模式中，因为基本时钟停止振荡，所以由基本时钟生成的 CPU 时钟和外围总线时钟也停止振荡，通过这些时钟运行的 CPU 和看门狗定时器也停止运行。因为主时钟、副时钟、PLL 时钟、内部振荡器时钟不停止振荡，所以使用这些时钟的外围功能保持运行状态。

### 8.7.2.1 外围功能时钟源的停止功能

当 CM0 寄存器的 CM02 位为“1”（在等待模式中停止外围功能的时钟源）时，因为在等待模式中，f1、f8、f32、fAD 以及选择外围功能时钟作为时钟源的 f2n 停止振荡，所以能降低功耗。fC32 以及选择主时钟作为时钟源的 f2n 不停止振荡。

### 8.7.2.2 向等待模式的转移

一旦执行 WAIT 指令，就转移到等待模式。

在使用等待模式时，必须在进行以下的设定后执行 WAIT 指令：

- 初始设定中的处理

先将用于返回的中断优先级（RIPL1 寄存器和 RIPL2 寄存器的 RLVL2～RLVL0 位）置“7”，然后设定各中断请求级。

- 转移到等待模式前的处理

(1) 将 I 标志置“0”。

(2) 对于用于从等待模式返回的中断，设定中断请求级。

(3) 对于不用于从等待模式返回的中断，将中断请求级设定为“0”。

(4) 设定标志寄存器的 IPL。

(5) 给用于返回的中断优先级设定和 IPL 相同的值。

用于返回的中断的中断请求级 > IPL = 用于返回的中断优先级。

(6) 转移到 PLL 自振荡模式、低速模式或者低功耗模式。

(7) 将 I 标志置“1”。

(8) 执行 WAIT 指令。

- 返回等待模式后的处理

在返回等待模式后，立即将用于返回的中断优先级置“7”。

## 8.7.2.3 等待模式的引脚状态

等待模式的引脚状态如表 8.5 所示。

表 8.5 等待模式的引脚状态

引脚		存储器扩展模式 微处理器模式	单芯片模式
地址总线、数据总线、 $\overline{CS0} \sim \overline{CS3}$ 、 $\overline{BC0} \sim \overline{BC3}$		保持进入等待模式前的状态。	—
$\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{WR0} \sim \overline{WR3}$		“H”电平	—
$\overline{HLDA}$ 、 $\overline{BCLK}$		“H”电平	—
ALE		“H”电平	—
端口		保持进入等待模式前的状态。	
DA0、DA1		保持进入等待模式前的状态。	
CLKOUT	选择 fC	输出时钟。	
	选择 f8 或者 f32	当 CM0 寄存器的 CM02 位为 “0”（在等待模式中不停止外围功能的时钟源）时，输出时钟。 当 CM02 位为 “1”（在等待模式中停止外围功能的时钟源）时，保持进入等待模式前的状态。	

## 8.7.2.4 从等待模式的返回

通过硬件复位、NMI 或者中断序号 0 ~ 63 的外围功能中断从等待模式返回。

在不使用外围功能中断而通过硬件复位或者 NMI 返回时，必须在将外围功能中断的 ILVL2 ~ ILVL0 位置 “000b”（禁止中断）后执行 WAIT 指令。

外围功能中断受 CM0 寄存器的 CM02 位的影响。当 CM02 位为 “0”（在等待模式中不停止外围功能的时钟源）时，中断序号 0 ~ 63 的外围功能中断能用于从等待模式的返回；当 CM02 位为 “1”（在等待模式中停止外围功能的时钟源）时，使用 f1、f8、f32、fAD 以及 f2n（选择外围功能的时钟源作为时钟源的 f2n）的外围功能停止振荡，并且不能用于从等待模式的返回。但是，通过 fC32、外部时钟或者 f2n（选择主时钟作为时钟源的 f2n）运行的中断序号 0 ~ 63 的外围功能中断，能用于从等待模式的返回。

通过外围功能中断或者 NMI 从等待模式返回时的 CPU 时钟和执行 WAIT 指令的 CPU 时钟相同。

能用于从等待模式返回的中断和使用条件如表 8.6 所示。

表 8.6 能用于从等待模式返回的中断和使用条件

中断	CM02=0	CM02=1
NMI	可使用	可使用
外部中断	可使用	可使用
键输入中断	可使用	可使用
低电压检测中断	可使用	可使用
定时器 A 中断 定时器 B 中断	可用于所有模式。	可在事件计数器模式或者计数源为 fC32 或者 f2n（选择主时钟作为时钟源）时使用。
串行接口中断（注 1）	可用于内部时钟和外部时钟。	可在使用外部时钟或者 f2n（选择主时钟作为时钟源）时使用。
A/D 转换中断	可用于单次模式或者单次扫描模式。	不可使用
智能 I/O 中断	可使用	不可使用
I <sup>2</sup> C 总线接口中断	可使用	不可使用
I <sup>2</sup> C 总线的线中断	可使用	可使用
CAN 唤醒中断	可使用	可使用

注 1. UART7 和 UART8 除外。

### 8.7.3 停止模式

在停止模式中，因为全部的振荡都停止，所以 CPU 时钟和外围功能时钟也停止振荡，通过这些时钟运行的 CPU 和外围功能也停止运行。停止模式是功耗最小的模式。

#### 8.7.3.1 向停止模式的转移

一旦执行 STOP 指令，就转移到停止模式。

在使用停止模式时，必须在进行以下的设定后执行 STOP 指令：

- 初始设定中的处理

先将用于返回的中断优先级（RIPL1 寄存器和 RIPL2 寄存器的 RLVL2～RLVL0 位）置“7”，然后设定各中断请求级。

- 转移到停止模式前的处理

(1) 将 I 标志置“0”。

(2) 对于用于从停止模式返回的中断，设定中断请求级。

(3) 对于不用于从停止模式返回的中断，将中断请求级设定为“0”。

(4) 设定标志寄存器的 IPL。

(5) 给用于返回的中断优先级设定和 IPL 相同的值。

用于返回的中断的中断请求级 > IPL = 用于返回的中断优先级

(6) 将基本时钟更改为主时钟的 256 分频时钟（f256）或者内部振荡器时钟的 4 分频时钟（fOCO4）。

(7) 在使用振荡停止检测功能时，将 CM2 寄存器的 CM20 位置“0”（振荡停止检测功能无效）。

(8) 将 I 标志置“1”。

(9) 执行 STOP 指令。

- 从停止模式返回后的处理

在从停止模式返回后，立即将用于返回的中断优先级置“7”。

#### 8.7.3.2 停止模式的引脚状态

停止模式的引脚状态如表 8.7 所示。

表 8.7 停止模式的引脚状态

引脚	存储器扩展模式 微处理器模式	单芯片模式
地址总线、数据总线、 $\overline{CS0} \sim \overline{CS3}$ 、 $\overline{BC0} \sim \overline{BC3}$	保持进入停止模式前的状态。	—
$\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{WR0} \sim \overline{WR3}$	“H”电平	—
$\overline{HLDA}$ 、BCLK	“H”电平	—
ALE	“H”电平	—
端口	保持进入停止模式前的状态。	
DA0、DA1	保持进入停止模式前的状态。	
CLKOUT	选择 fC	“H”电平
	选择 f8 或者 f32	保持进入停止模式的前状态。
XIN	高阻抗	
XOUT	“H”电平	
XCIN、XCOUT	高阻抗	

### 8.7.3.3 从停止模式的返回

通过硬件复位、NMI、低电压检测中断或者外围功能中断（参照表 8.8）从停止模式返回。

在不使用外围功能中断而通过硬件复位或者 NMI 返回时，必须在将外围功能中断的 ILVL2 ~ ILVL0 位全部置“000b”（禁止中断）后执行 STOP 指令。

通过外围功能中断或者 NMI 从停止模式返回时的 CPU 时钟和执行 STOP 指令时的 CPU 时钟相同。

能用于从停止模式返回的中断和使用条件如表 8.8 所示。

表 8.8 能用于从停止模式返回的中断和使用条件

中断	条件
NMI	
低电压检测中断	
外部中断	
键输入中断	
定时器 A 中断 定时器 B 中断	在事件计数器模式中对频率不超过 100Hz 的外部脉冲进行计数的情况下。
串行接口中断（注 1）	在使用外部时钟的情况下。
I <sup>2</sup> C 总线的线中断	
CAN 唤醒中断	

注 1. UART7 和 UART8 除外。

## 8.8 系统时钟的保护功能

这是在选择 PLL 时钟作为基本时钟的时钟源时，为了在程序失控时使 CPU 时钟不停止振荡而禁止更改时钟的功能。

如果将 PM2 寄存器的 PM21 位置“1”（禁止更改时钟），就无法写以下的位：

- CM0 寄存器的 CM02 位和 CM05 位
- CM1 寄存器的 CM10 位
- CM2 寄存器的 CM20 位
- PM2 寄存器的 PM27 位

在使用系统时钟的保护功能时，必须在 CM0 寄存器的 CM05 位为“0”（主时钟振荡）并且 CCR 寄存器的 BCS 位为“0”（基本时钟的时钟源为 PLL 时钟）的状态下进行以下的处理：

1. 将 PRCR 寄存器的 PRC1 位置“1”（允许写 PM2 寄存器）。
2. 将 PM2 寄存器的 PM21 位置“1”（禁止更改时钟）。
3. 将 PRCR 寄存器的 PRC1 位置“0”（禁止写 PM2 寄存器）。

## 8.9 使用时钟发生电路时的注意事项

### 8.9.1 副时钟

#### 8.9.1.1 振荡电路常数匹配の確認

必须确认驱动能力 High 和驱动能力 Low 时的副时钟振荡电路的振荡电路常数的匹配情况。  
有关振荡电路常数的匹配，请向振荡器厂商询问。

### 8.9.2 功率控制

只有在要使用的时钟振荡稳定后，才能转换基本时钟源和时钟分频比。一旦将 CM3 寄存器的 CM31 位置“1”，内部振荡器瞬间就开始振荡，因此不需要振荡稳定的等待时间。

如果将基本时钟源从 PLL 时钟转换为低速时钟（将 CCR 寄存器的 BCS 位置“1”），就必须使用 MOVL 指令或者 OR.L 指令。

- 使用汇编语言时的例子  
OR.L                    #80h,0004h
- 使用C语言时的例子  
asm("OR.L            #80h,0004h");

#### 8.9.2.1 停止模式

- 在通过复位从停止模式返回时，必须在主时钟振荡充分稳定前将“L”电平输入到RESET引脚。

#### 8.9.2.2 降低功耗的要点

请在进行系统设计和编程时参考。

- 引脚处理  
如果将输入引脚置为开路，就可能有穿透电流流过。因此必须将未使用引脚设定为输入并且各引脚通过电阻连接到VSS（下拉），或者将未使用引脚设定为输出并将引脚置为开路。
- A/D转换器  
在不进行A/D转换时，必须将AD0CON1寄存器的VCUT位置“0”（未连接VREF）；在进行A/D转换时，必须先将VCUT位置“1”（连接VREF），再至少等待1μs，然后开始A/D转换。
- D/A转换器  
在不进行D/A转换时，必须将DACON寄存器的DAiE位（i=0,1）置“0”（禁止输出）并将DAi寄存器置“00h”。
- 外围功能的停止  
在向等待模式转移时，能通过CM0寄存器的CM02位停止外围功能的时钟源，从而降低功耗。但是，fC32不停止振荡。

## 9. 总线

总线包括内部总线和外部总线，内部总线包括高速运行的 CPU 总线和低速外围总线。总线的框图如图 9.1 所示。

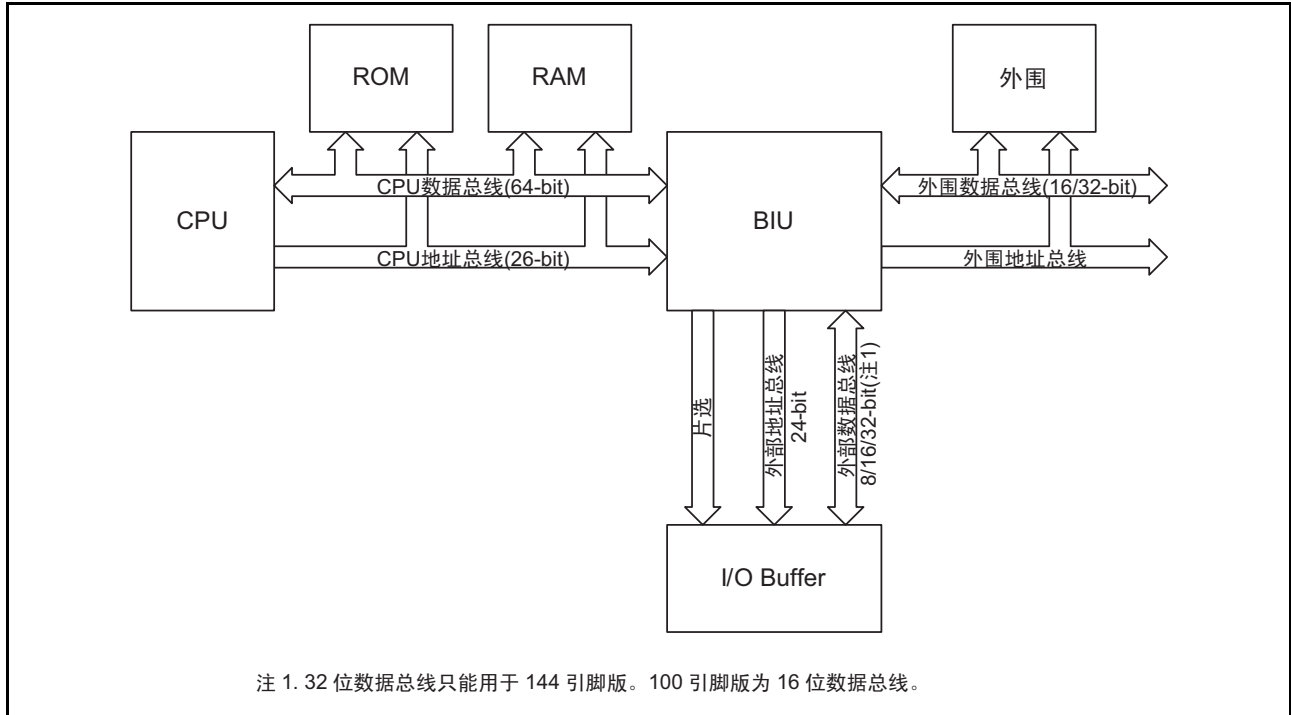


图 9.1 总线的框图

在存储器扩展模式或者微处理器模式中，部分引脚为控制地址总线 and 数据总线的总线控制引脚。总线控制引脚有  $A0 \sim A23$ 、 $D0 \sim D31$ 、 $\overline{CS0} \sim \overline{CS3}$ 、 $\overline{WR0}/\overline{WR}$ 、 $\overline{BC0}$ 、 $\overline{WR1}/\overline{BC1}$ 、 $\overline{WR2}/\overline{BC2}$ 、 $\overline{WR3}/\overline{BC3}$ 、 $\overline{RD}$ 、 $\overline{BCLK}$ 、 $\overline{HLDA}$ 、 $\overline{HOLD}$ 、 $\overline{ALE}$ 、 $\overline{RDY}$ 。

### 9.1 总线的设定

能通过复位向量的低 2 位、PBC 寄存器、EBC0 ~ EBC3 寄存器、CSOP0 ~ CSOP2 寄存器转换总线的设定。

总线的设定和转换源如表 9.1 所示。

表 9.1 总线的设定和转换源

总线的设定	转换源
内部 SFR 总线时序	PBC 寄存器
外部总线时序	EBC0 ~ EBC3 寄存器
外部数据总线宽度	PBC 寄存器、EBC0 ~ EBC3 寄存器
复位后的外部数据总线宽度	复位向量的低 2 位
分离总线 / 多路复用总线的选择	PBC 寄存器、EBC0 ~ EBC3 寄存器
片选的输出引脚	CSOP0 ~ CSOP2 寄存器



## 9.2 外围总线时序的设定

外围总线是以最大 32MHz（理论值，各产品的最大值是“28. 电特性”中规定的 f(BCLK) 的值）运行的 16 位 /32 位总线，通过总线接口单元（BIU）进行与高速运行的 64 位 CPU 总线的时序调整和总线宽度转换。

设定外围总线时序的 PBC 寄存器如图 9.2 所示。

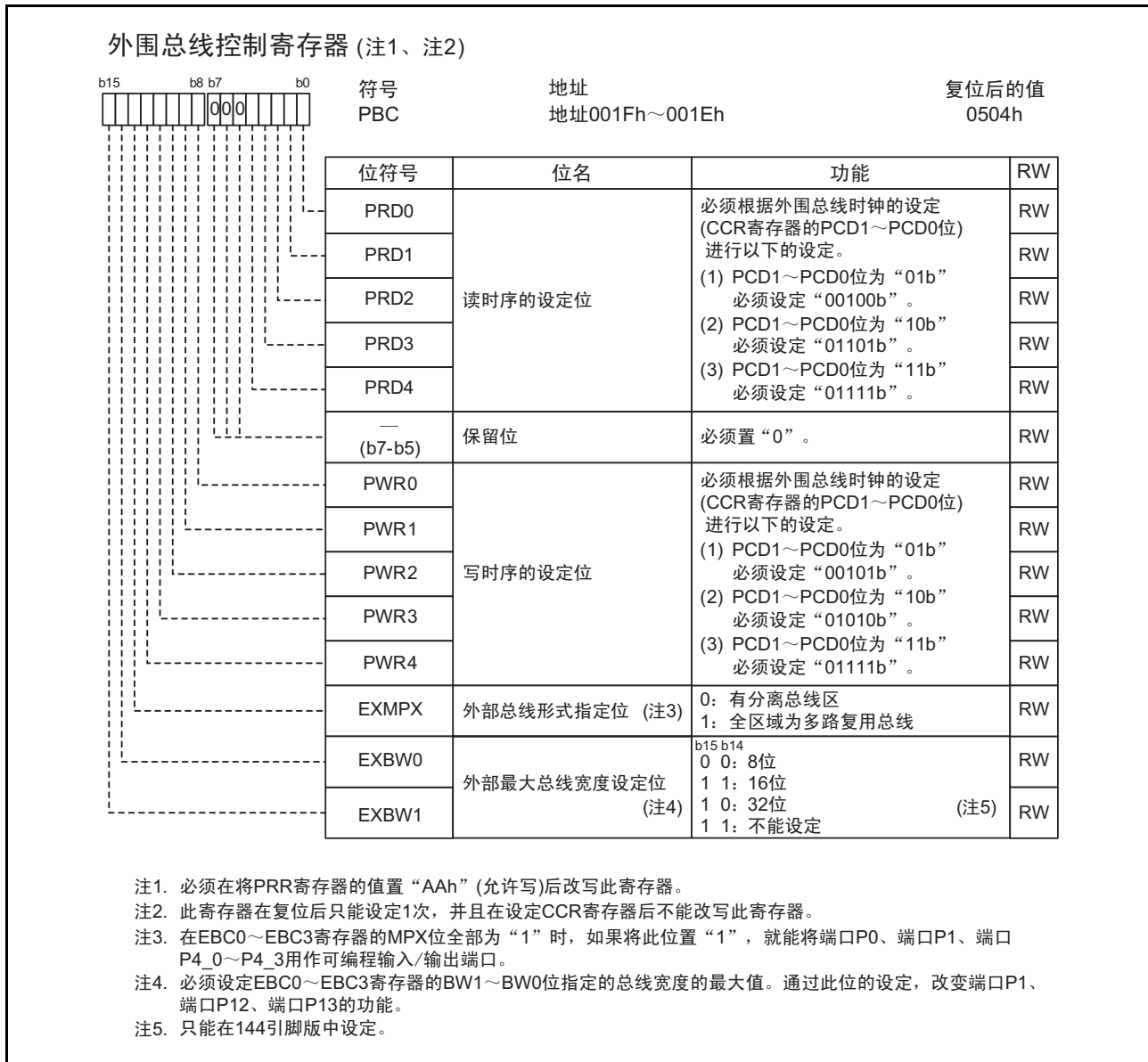


图 9.2 PBC 寄存器

### 9.3 外部总线的设定

外部总线是以最大 32MHz（理论值，各产品的最大值和“28. 电特性”中规定的 f(BCLK) 的值相同）运行的 8 位 /16 位 /32 位总线。通过总线接口单元（BIU）进行与高速运行的 64 位 CPU 总线的时序调整和总线宽度转换。

#### 9.3.1 外部地址空间的设定

R32C/100 系列的 CPU 有 26 条 MCU 内部地址总线 A0 ~ A25。因为 A26 ~ A31 对 A25 进行符号扩展，所以能存取的区域为地址 00000000h ~ 01FFFFFFh 和地址 FE000000h ~ FFFFFFFFh，共 64MB。

其中，输出到外部的地址总线为 A0 ~ A23，最多 24 条，并且对 A18 ~ A25 解码后生成的片选信号是  $\overline{CS3}$  ~  $\overline{CS0}$ ，共 4 个。通过给全部片选信号分配 16MB 空间，能将最大 63.5MB 的空间分配给外部。在从单芯片模式变为存储器扩展模式时，在存取外部区域前地址总线的状态不确定。

片选信号  $\overline{CS3}$  ~  $\overline{CS0}$  分别和 A20 ~ A23 引脚复用。另外，也能从 P11\_0 ~ P11\_3 输出  $\overline{CS0}$  ~  $\overline{CS3}$ ，从 P5\_4、P5\_6 和 P5\_7 输出  $\overline{CS1}$  ~  $\overline{CS3}$ 。

在微处理器模式中，复位后从端口 P4\_7 输出  $\overline{CS0}$  信号。因为不能使用 A23，所以每个片选的最大区域为 8MB。另外，在设定前不能输出  $\overline{CS1}$  ~  $\overline{CS3}$ 。

$\overline{CSi}$ （i=0 ~ 3）在存取外部区域 i 的过程中输出“L”电平。如果存取其他外部区域，就输出“H”电平。地址总线和片选信号的输出例子如图 9.3 所示。

能通过 CSOP0 ~ CSOP2 寄存器选择要使用的片选信号的种类和输出引脚，通过 CB01、CB12、CB23 寄存器选择各片选区域。

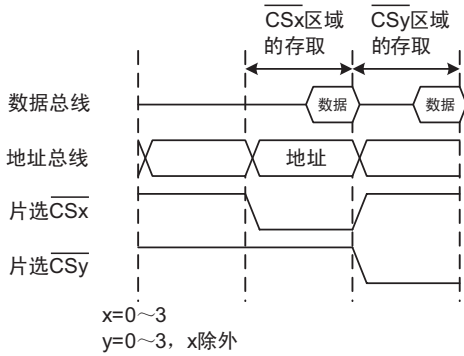
CSOP0 ~ CSOP2 寄存器分别如图 9.4 ~ 图 9.6 所示，CB01、CB12、CB23 寄存器分别如图 9.7、图 9.8 和图 9.9 所示，片选区域如图 9.10 和图 9.11 所示。

对于 CSOP0 ~ CSOP2 寄存器，不能设定为从多个引脚输出相同的片选信号，并且 CB01、CB12、CB23 寄存器的设定必须满足以下的条件：

- 存储器扩展模式的情况  
 $0080000h < (CB23 \times 2^{18}) < (CB12 \times 2^{18}) < (CB01 \times 2^{18}) \leq 3DC0000h$
- 微处理器模式的情况  
 $0080000h < (CB23 \times 2^{18}) < (CB12 \times 2^{18}) < (CB01 \times 2^{18}) \leq 3FC0000h$

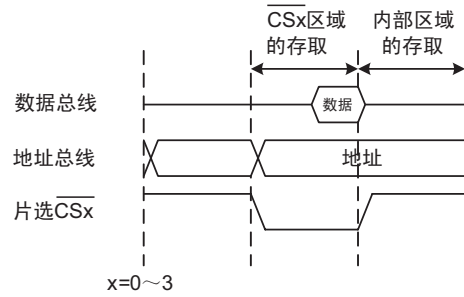
(1) 在存取外部区域后的下一个周期，地址总线和片选信号都发生变化。

在存取CSx区域后的下一个周期，如果存取CSy区域，地址总线和片选信号就都发生变化。



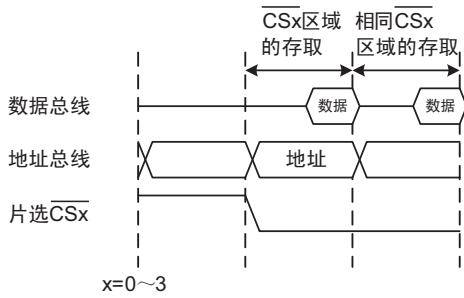
(2) 在存取外部区域后的下一个周期，只有片选信号发生变化(地址总线不变化)。

在存取CSx区域后的下一个周期，如果存取内部区域，片选信号就发生变化，而地址总线不变化。



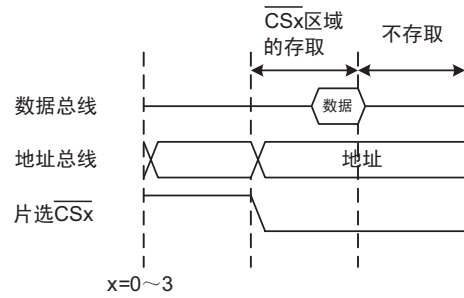
(3) 在存取外部区域后的下一个周期，只有地址总线发生变化(片选信号不变化)。

在存取CSx区域后的下一个周期，如果存取相同的CSx区域，地址总线就发生变化，而片选信号不变化。



(4) 在存取外部区域后的下一个周期，地址总线和片选信号都不变化。

在存取CSx区域后的下一个周期，如果不存取任何区域(也不发生预取指令)，地址总线和片选信号就都不变化。



注1. 此例表示连续2个周期的地址总线和片选信号。通过这些组合，片选信号有可能超过2个总线周期。

图 9.3 地址总线和片选信号的输出例子 (分离总线)

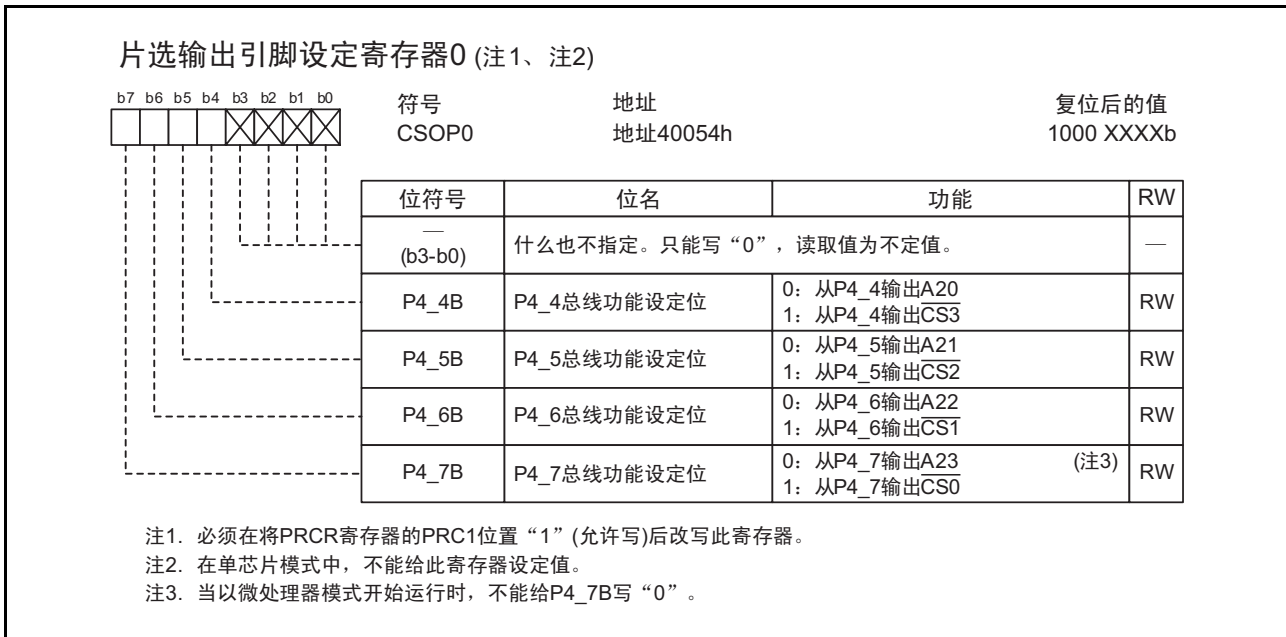


图 9.4 CSOP0 寄存器

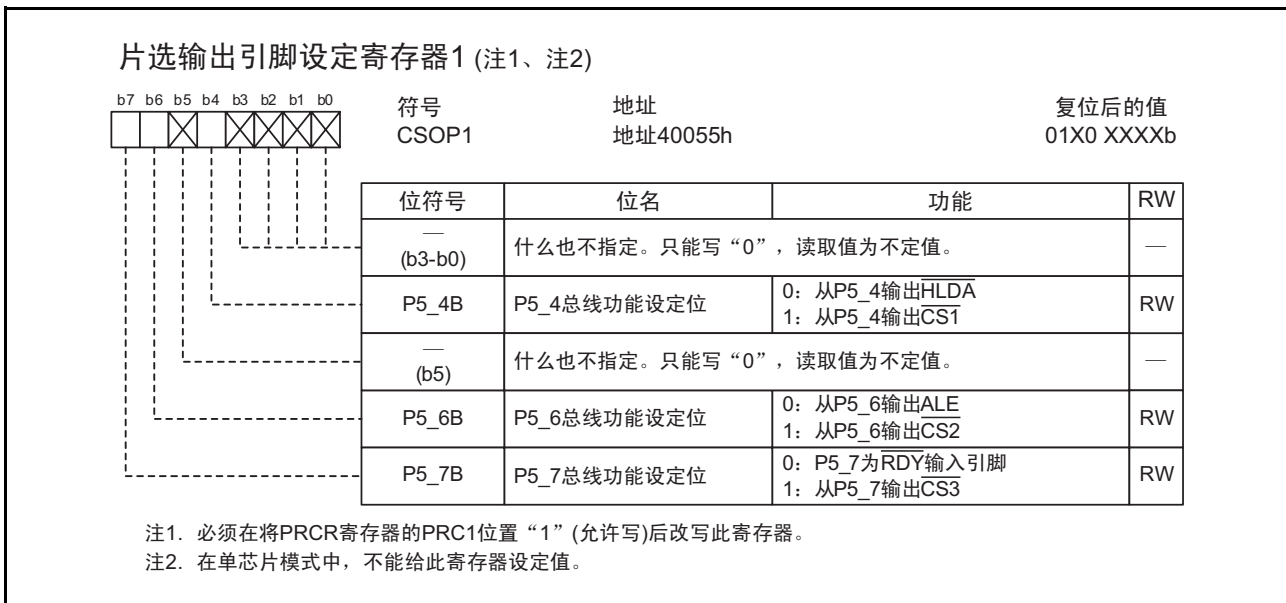


图 9.5 CSOP1 寄存器

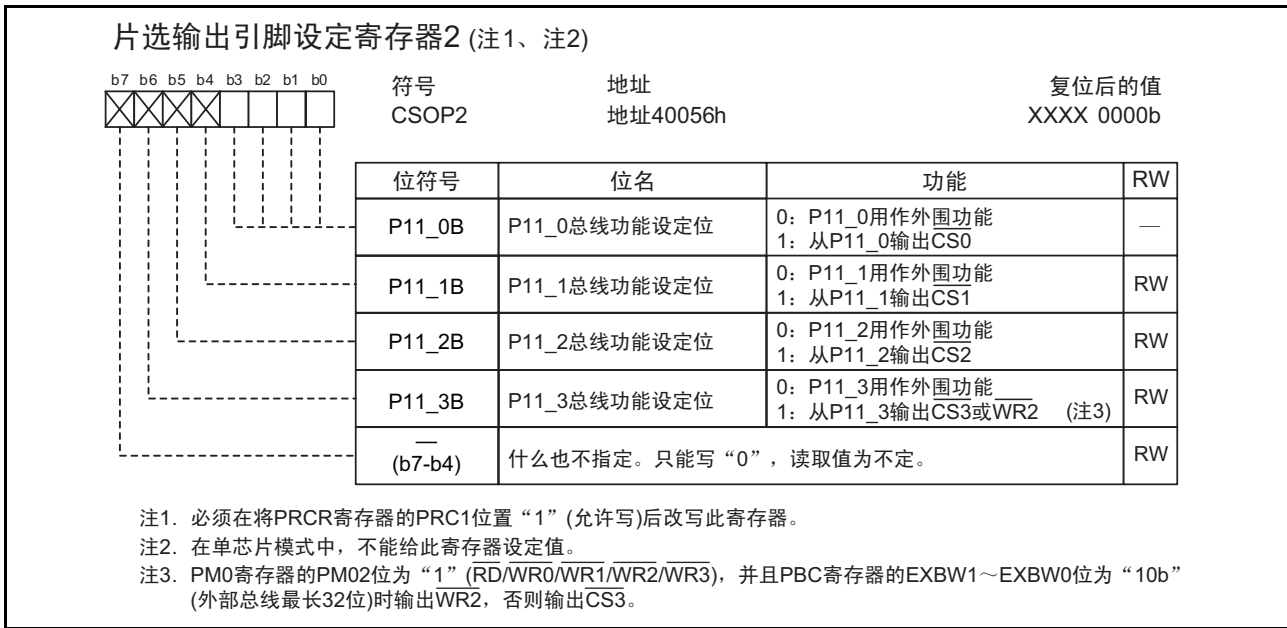


图 9.6 CSOP2 寄存器

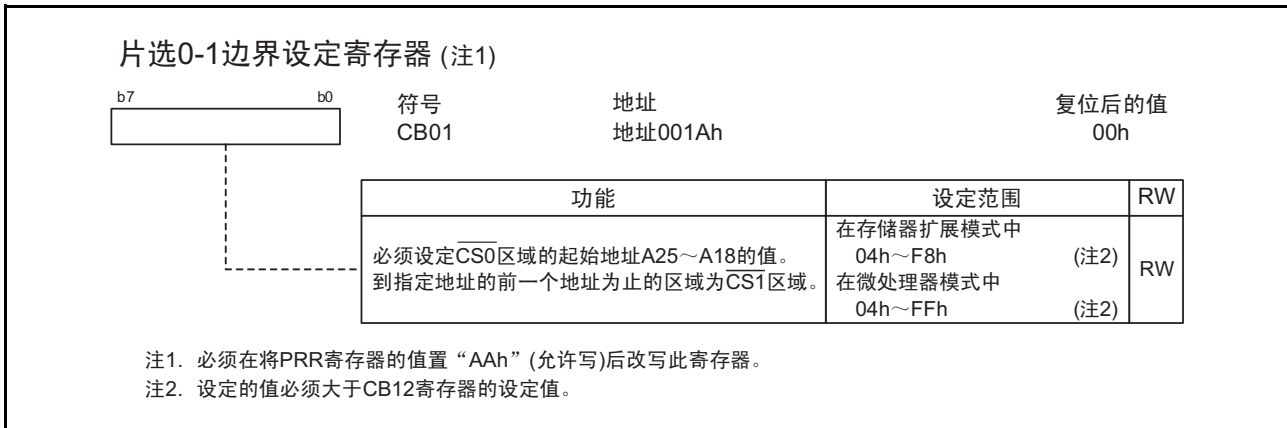


图 9.7 CB01 寄存器

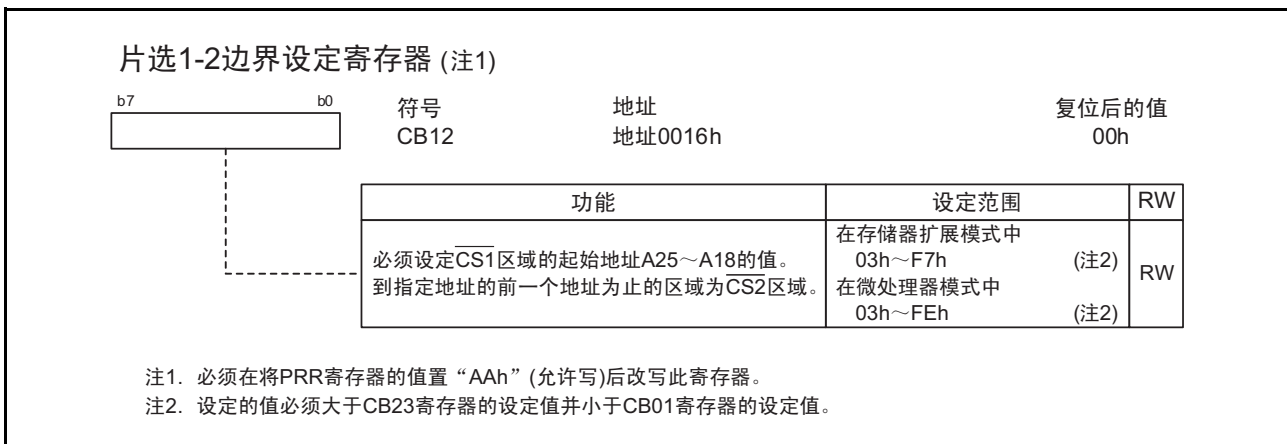


图 9.8 CB12 寄存器

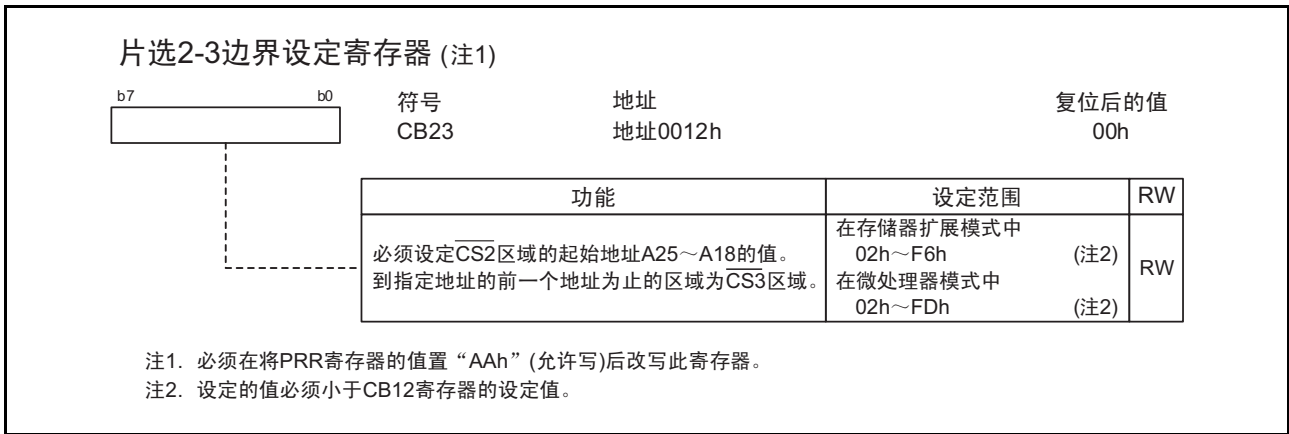


图 9.9 CB23 寄存器

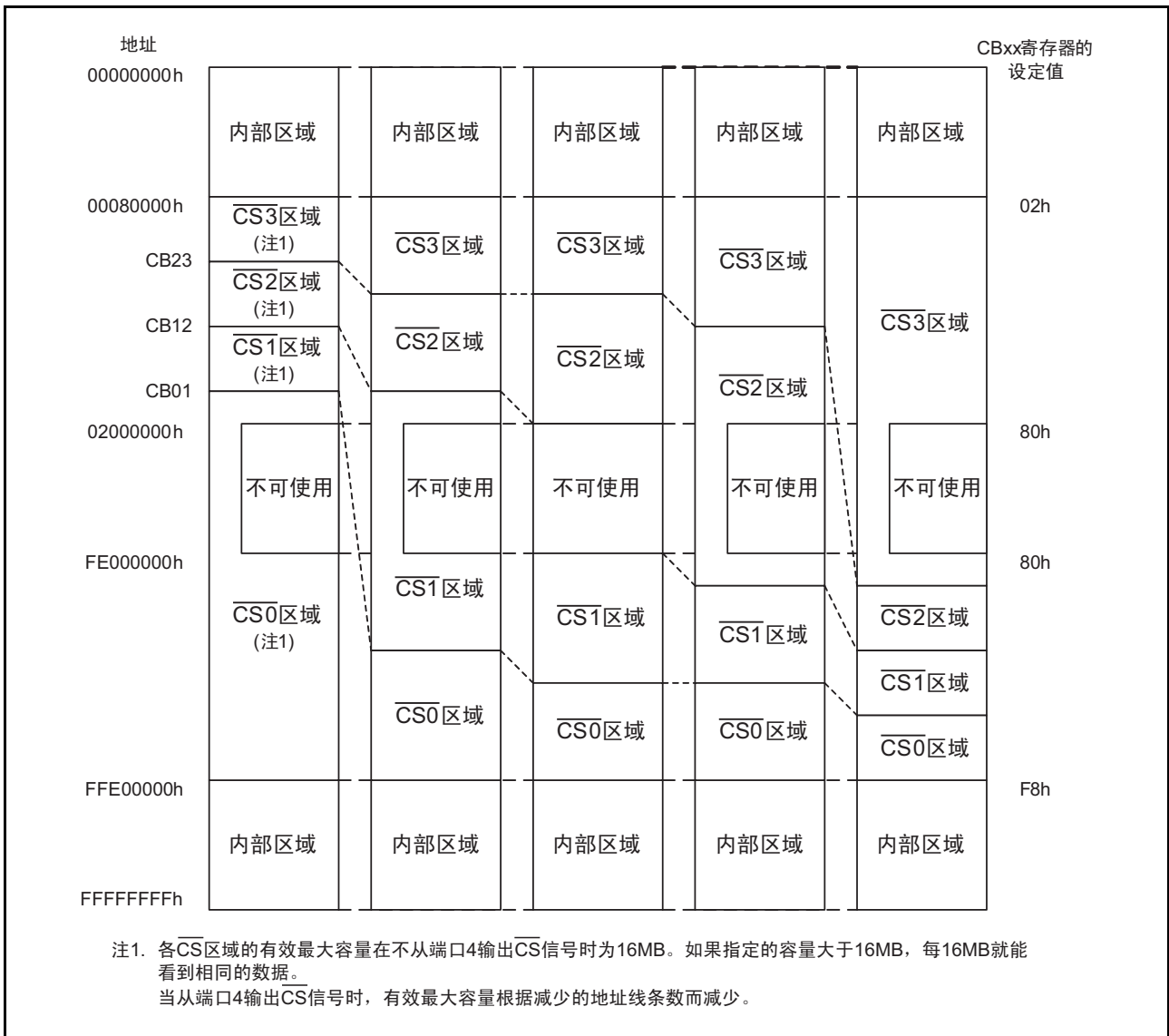


图 9.10 存储器扩展模式的片选区域

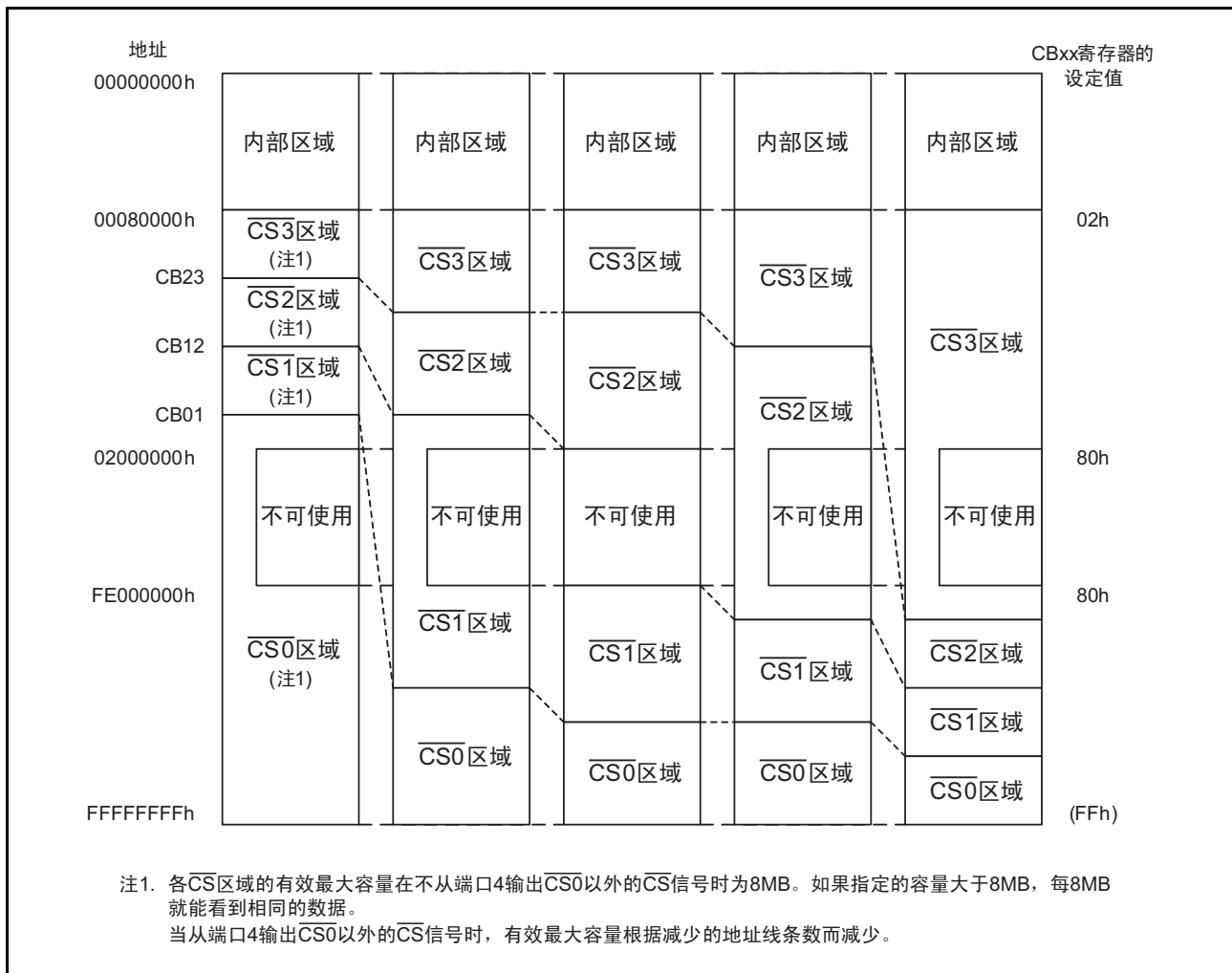


图 9.11 微处理器模式的片选区域





### 9.3.3 分离总线和多路复用总线的选择

总线的形式能选择分离总线或者多路复用总线。能通过 EBC0 ~ EBC3 寄存器的 MPX 位指定各区域的总线形式。在全区域使用多路复用总线时，如果将 PBC 寄存器的 EXMPX 位置“1”（全区域为多路复用总线），就能将端口 P0、P1、P4\_0 ~ P4\_3 用作可编程输入 / 输出端口。

#### (1) 分离总线

分离总线是从不同引脚分别输入或者输出数据和地址的总线形式。

如果将 EBC0 ~ EBC3 寄存器的 MPX 位置“0”，该区域就为分离总线。

能通过 EBC0 ~ EBC3 寄存器的 BW1 ~ BW0 位选择 8 位、16 位或者 32 位的数据总线。

当 PBC 寄存器的 EXBW1 ~ EXBW0 位为“00b”（8 位数据总线）时，端口 P0 为数据总线，端口 P1、端口 P12、以及端口 P13 为可编程输入 / 输出端口。

当 EXBW1 ~ EXBW0 位为“01b”（16 位数据总线）时，端口 P0 和端口 P1 为数据总线，端口 P12 和端口 P13 为可编程输入 / 输出端口。但是，在存取 BW1 ~ BW0 位为“00b”（8 位数据总线）的区域时，端口 P1（D8 ~ D15）为不定值。

当 EXBW1 ~ EXBW0 位为“10b”（32 位数据总线）时，端口 P0、端口 P1、端口 P12、以及端口 P13 为数据总线。但是，在存取 BW1 ~ BW0 位为“00b”（8 位数据总线）的区域时，端口 P1、端口 P12、以及 P13（D8 ~ D31）为不定值；在存取 BW1 ~ BW0 位为“01b”（16 位数据总线）的区域时，端口 P12 和端口 P13（D16~D31）为不定值。

#### (2) 多路复用总线

多路复用总线是从同一个引脚分时输入或者输出数据和地址的总线形式。

如果将 EBC0 ~ EBC3 寄存器的 MPX 位置“1”，该区域就为多路复用总线。

对于将 EBC0 ~ EBC3 寄存器的 BW1 ~ BW0 位置“00b”（8 位数据总线）的区域，D0 ~ D7 和 A0 ~ A7 被多路复用；对于将 BW1 ~ BW0 位置“01b”（16 位数据总线）或者置“10b”（32 位数据总线）的区域，D0 ~ D15 和  $\overline{BC0}$ 、A1/ $\overline{BC2}$ 、A2 ~ A15 被多路复用。

在微处理器模式中，因为复位后通过分离总线运行，所以能将  $\overline{CS1}$  ~  $\overline{CS3}$  区域设定为多路复用总线，但是不能将  $\overline{CS0}$  区域设定为多路复用总线。

各处理器模式的引脚功能如表 9.2、各总线形式的引脚功能如表 9.3 所示。

表 9.2 处理器模式和引脚功能 (注 1)

处理器模式	单芯片模式	微处理器模式 / 存储器扩展模式						存储器扩展模式		
总线形式	—	分离总线			分离总线和多路复用总线			多路复用总线		
数据总线宽度	—	8 位总线	8 位 /16 位总线	8 位 /16 位 /32 位总线	8 位总线	8 位 /16 位总线	8 位 /16 位 /32 位总线	8 位总线	8 位 /16 位总线	8 位 /16 位 /32 位总线
P0_0 ~ P0_7	输入 / 输出端口	D0 ~ D7						输入 / 输出端口		
P1_0 ~ P1_7	输入 / 输出端口	输入 / 输出端口	D8 ~ D15		输入 / 输出端口	D8 ~ D15		输入 / 输出端口		
P2_0	输入 / 输出端口	A0	A0 或者 $\overline{BC0}$		A0 或者 A0/D0	A0、A0/D0、 $\overline{BC0}$ 或者 $\overline{BC0/D0}$		A0/D0	A0/D0 或者 $\overline{BC0/D0}$	
P2_1	输入 / 输出端口	A1	A1 或者 $\overline{BC2}$	A1 或者 A1/D1		A1 或者 A1/D1 或者 $\overline{BC2}$ 或者 $\overline{BC2/D1}$		A1/D1	A1/D1 或者 $\overline{BC2/D1}$	
P2_2 ~ P2_7	输入 / 输出端口	A2 ~ A7			A2 ~ A7 或者 A2/D2 ~ A7/D7			A2/D2 ~ A7/D7		
P3_0 ~ P3_7	输入 / 输出端口	A8 ~ A15			A8 ~ A15	A8 ~ A15 或者 A8/D8 ~ A15/D15		A8 ~ A15	A8/D8 ~ A15/D15	
P4_0 ~ P4_3	输入 / 输出端口	A16 ~ A19						A16 ~ A19 或者 输入 / 输出端口		
P4_4	输入 / 输出端口	A20 或者 $\overline{CS3}$								
P4_5	输入 / 输出端口	A21 或者 $\overline{CS2}$								
P4_6	输入 / 输出端口	A22 或者 $\overline{CS1}$								
P4_7	输入 / 输出端口	A23 或者 $\overline{CS0}$								
P5_0	输入 / 输出端口	$\overline{WR}$ 或者 $\overline{WR0}$								
P5_1	输入 / 输出端口	不定值 (注 2)	$\overline{BC1}$ 或者 $\overline{WR1}$		不定值 (注 2)	$\overline{BC1}$ 或者 $\overline{WR1}$		不定值 (注 2)	$\overline{BC1}$ 或者 $\overline{WR1}$	
P5_2	输入 / 输出端口	$\overline{RD}$								
P5_3	输入 / 输出端口	BCLK								
P5_4	输入 / 输出端口	HLDA 或者 $\overline{CS1}$								
P5_5	输入 / 输出端口	HOLD								
P5_6	输入 / 输出端口	ALE 或者 $\overline{CS2}$			必须设定为 ALE					
P5_7	输入 / 输出端口	RDY 或者 $\overline{CS3}$								
P11_0 ~ P11_2	输入 / 输出端口	$\overline{CS0}$ ~ $\overline{CS2}$ 或者 输入 / 输出端口								
P11_3	输入 / 输出端口	$\overline{CS3}$ 或者 输入 / 输出端口	$\overline{CS3}$ 或者 $\overline{WR2}$	$\overline{CS3}$ 或者 输入 / 输出端口	$\overline{CS3}$ 或者 $\overline{WR2}$	$\overline{CS3}$ 或者 输入 / 输出端口	$\overline{CS3}$ 或者 $\overline{WR2}$	$\overline{CS3}$ 或者 输入 / 输出端口	$\overline{CS3}$ 或者 $\overline{WR2}$	$\overline{CS3}$ 或者 $\overline{WR2}$
P11_4	输入 / 输出端口	输入 / 输出端口	$\overline{CS3}$ 或者 $\overline{WR2}$	输入 / 输出端口	$\overline{CS3}$ 或者 $\overline{WR2}$	输入 / 输出端口	$\overline{CS3}$ 或者 $\overline{WR2}$	输入 / 输出端口	$\overline{CS3}$ 或者 $\overline{WR2}$	$\overline{CS3}$ 或者 $\overline{WR2}$
P12_0 ~ P12_7	输入 / 输出端口	输入 / 输出端口	D16 ~ D23	输入 / 输出端口	D16 ~ D23	输入 / 输出端口	D16 ~ D23	输入 / 输出端口	D16 ~ D23	D16 ~ D23
P13_0 ~ P13_7	输入 / 输出端口	输入 / 输出端口	D24 ~ D31	输入 / 输出端口	D24 ~ D31	输入 / 输出端口	D24 ~ D31	输入 / 输出端口	D24 ~ D31	D24 ~ D31

注 1. 端口 P11 ~ P15 只限于 144 引脚版。

注 2. 输出不定值。

表 9.3 总线形式和引脚功能（微处理器模式 / 存储器扩展模式）（注 1）

总线形式	分离总线			多路复用总线		
MPX 位	“0”			“1”		
总线宽度	8 位	16 位	32 位	8 位	16 位	32 位
BW1 ~ BW0 位	“00b”	“01b”	“10b”	“00b”	“01b”	“10b”
P0_0 ~ P0_7	D0 ~ D7			输入 / 输出端口		
P1_0 ~ P1_7	输入 / 输出端口	D8 ~ D15		输入 / 输出端口		
P2_0	A0	$\overline{BC0}$		A0/D0	$\overline{BC0/D0}$	
P2_1	A1		$\overline{BC2}$	A1/D1		$\overline{BC2/D1}$
P2_2 ~ P2_7	A2 ~ A7			A2/D2 ~ A7/D7		
P3_0 ~ P3_7	A8 ~ A15			A8/D8 ~ A15/D15		
P4_0 ~ P4_3	A16 ~ A19			A16 ~ A19 或者输入 / 输出端口		
P4_4	A20 或者 $\overline{CS3}$					
P4_5	A21 或者 $\overline{CS2}$					
P4_6	A22 或者 $\overline{CS1}$					
P4_7	A23 或者 $\overline{CS0}$ （在微处理器模式中固定为 $\overline{CS0}$ ）					
P5_0	$\overline{WR}$ 或者 $\overline{WR0}$					
P5_1	不定值（注 2）	$\overline{BC1}$ 或者 $\overline{WR1}$		不定值（注 2）	$\overline{BC1}$ 或者 $\overline{WR1}$	
P5_2	$\overline{RD}$					
P5_3	BCLK					
P5_4	$\overline{HLDA}$ 或者 $\overline{CS1}$					
P5_5	$\overline{HOLD}$					
P5_6	ALE 或者 $\overline{CS2}$			必须设定为 ALE		
P5_7	$\overline{RDY}$ 或者 $\overline{CS3}$					
P11_0 ~ P11_2	$\overline{CS0} \sim \overline{CS2}$ 或者输入 / 输出端口					
P11_3	$\overline{CS3}$ 或者输入 / 输出端口		$\overline{CS3}$ 或者 $\overline{WR2}$	$\overline{CS3}$ 或者输入 / 输出端口		$\overline{CS3}$ 或者 $\overline{WR2}$
P11_4	输入 / 输出端口		$\overline{BC3}$ 或者 $\overline{WR3}$	输入 / 输出端口		$\overline{BC3}$ 或者 $\overline{WR3}$
P12_0 ~ P12_7	输入 / 输出端口		D16 ~ D23	输入 / 输出端口		D16 ~ D23
P13_0 ~ P13_7	输入 / 输出端口		D24 ~ D31	输入 / 输出端口		D24 ~ D31

注 1. 端口 P11 ~ P15 只限于 144 引脚版。

注 2. 输出不定值。

### 9.3.4 读写信号

当数据总线为 16 位或者 32 位时，能通过 PM0 寄存器的 PM02 位选择  $\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{BC0}$ 、 $\overline{BC1}$ 、 $\overline{BC2}$ 、 $\overline{BC3}$  的组合或者  $\overline{RD}$ 、 $\overline{WR0}$ 、 $\overline{WR1}$ 、 $\overline{WR2}$ 、 $\overline{WR3}$  的组合作为读写信号。

当 PBC 寄存器的 EXBW1 ~ EXBW0 位为 “00b”（外部区域为 8 位数据总线）时，必须将 PM02 位置 “0”（ $\overline{RD}/\overline{WR}/\overline{BC0}/\overline{BC1}/\overline{BC2}/\overline{BC3}$ ）。当 EXBW1 ~ EXBW0 位为 “01b”（外部区域最大为 16 位宽度）或者为 “10b”（外部区域最大为 32 位宽度），并且存取 8 位宽度的区域时，与 PM02 位的值无关，读写信号为  $\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{BC0}$ 、 $\overline{BC1}$ 、 $\overline{BC2}$ 、 $\overline{BC3}$  的组合。

各信号的运行如表 9.4 和表 9.5 所示。

在复位后，读写信号为  $\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{BC0}$ 、 $\overline{BC1}$ 、 $\overline{BC2}$ 、 $\overline{BC3}$  的组合。在转换为  $\overline{RD}$ 、 $\overline{WR0}$ 、 $\overline{WR1}$ 、 $\overline{WR2}$ 、 $\overline{WR3}$  的组合时，必须在设定 PM02 位后写外部存储器。

表 9.4  $\overline{RD}$ 、 $\overline{WR0}$ 、 $\overline{WR1}$ 、 $\overline{WR2}$ 、 $\overline{WR3}$  的运行 (注 1)

数据总线宽度	$\overline{RD}$	$\overline{WR0}$	$\overline{WR1}$	$\overline{WR2}$	$\overline{WR3}$	外部数据总线的状态
32 位 (注 2)	L	H	H	H	H	读 4 字节数据。
	H	L	H	H	H	将 1 字节数据写到地址 4n+0。
	H	H	L	H	H	将 1 字节数据写到地址 4n+1。
	H	H	H	L	H	将 1 字节数据写到地址 4n+2。
	H	H	H	H	L	将 1 字节数据写到地址 4n+3。
	H	L	L	H	H	将 2 字节数据写到地址 4n+0 ~ 1。
	H	H	L	L	H	将 2 字节数据写到地址 4n+1 ~ 2。
	H	H	H	L	L	将 2 字节数据写到地址 4n+2 ~ 3。
	H	L	L	L	H	将 3 字节数据写到地址 4n+0 ~ 2。
	H	H	L	L	L	将 3 字节数据写到地址 4n+1 ~ 3。
16 位	L	H	H	H/L (A1)	—	读 2 字节数据。
	H	L	H	H/L (A1)	—	将 1 字节数据写到偶数地址。
	H	H	L	H/L (A1)	—	将 1 字节数据写到奇数地址。
	H	L	L	H/L (A1)	—	将 2 字节数据写到偶数地址和奇数地址。
8 位	L	H(WR)	—	H/L (A1)	—	读 1 字节数据。
	H	L(WR)	—	H/L (A1)	—	写 1 字节数据。

注 1.  $\overline{WR2}$ 、 $\overline{WR3}$  只限于 144 引脚版。

注 2. 只能在 144 引脚版中指定。

表 9.5  $\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{BC0}$ 、 $\overline{BC1}$ 、 $\overline{BC2}$ 、 $\overline{BC3}$  的运行 (注 1)

数据总线宽度	$\overline{RD}$	$\overline{WR}$	$\overline{BC0}$	$\overline{BC1}$	$\overline{BC2}$	$\overline{BC3}$	外部数据总线的状态
32 位 (注 2)	L	H	L	L	L	L	读 4 字节数据。
	H	L	L	H	H	H	将 1 字节数据写到地址 4n+0。
	H	L	H	L	H	H	将 1 字节数据写到地址 4n+1。
	H	L	H	H	L	H	将 1 字节数据写到地址 4n+2。
	H	L	H	H	H	L	将 1 字节数据写到地址 4n+3。
	H	L	L	L	H	H	将 2 字节数据写到地址 4n+0 ~ 1。
	H	L	H	L	L	H	将 2 字节数据写到地址 4n+1 ~ 2。
	H	L	H	H	L	L	将 2 字节数据写到地址 4n+2 ~ 3。
	H	L	L	L	L	H	将 3 字节数据写到地址 4n+0 ~ 2。
	H	L	H	L	L	L	将 3 字节数据写到地址 4n+1 ~ 3。
16 位	L	H	L	L	H/L (A1)	—	读 2 字节数据。
	H	L	L	H	H/L (A1)	—	将 1 字节数据写到偶数地址。
	H	L	H	L	H/L (A1)	—	将 1 字节数据写到奇数地址。
	H	L	L	L	H/L (A1)	—	将 2 字节数据写到偶数地址和奇数地址。
8 位	L	H	H/L(A0)	—	H/L (A1)	—	读 1 字节数据。
	H	L	H/L(A0)	—	H/L (A1)	—	写 1 字节数据。

注 1.  $\overline{BC2}$ 、 $\overline{BC3}$  只限于 144 引脚版。

注 2. 只能在 144 引脚版中指定。

### 9.3.5 外部总线时序

通过 EBC0 ~ EBC3 寄存器设定外部总线时序，基准时钟是通过 CCR 寄存器的 BCD1 ~ BCD0 位设定的基本时钟。

MPY1 ~ MPY0 位和 ESUR1 ~ ESUR0 位的设定值与  $T_{su}(A-R)$ （读操作前的地址准备周期数）的关系、MPY1 ~ MPY0 位和 EWR1 ~ EWR0 位的设定值与  $T_w(R)$ （读信号的脉宽）的关系、MPY1 ~ MPY0 位和 ESUW1 ~ ESUW0 位的设定值与  $T_{su}(A-W)$ （写操作前的地址准备周期数）的关系、MPY1 ~ MPY0 位和 EWW1 ~ EWW0 位的设定值与  $T_w(W)$ （写信号的脉宽）的关系分别如表 9.6、表 9.7、表 9.8 和表 9.9 所示。

表 9.6 MPY1 ~ MPY0 位和 ESUR1 ~ ESUR0 位的设定值与  $T_{su}(A-R)$  的关系（单位：周期）

ESUR1 ~ ESUR0 位的设定值		分离总线				多路复用总线			
		MPY1 ~ MPY0 位的设定值				MPY1 ~ MPY0 位的设定值			
		00b	01b	10b	11b	00b	01b	10b	11b
		<i>mpy</i> =1	<i>mpy</i> =2	<i>mpy</i> =3	<i>mpy</i> =4	<i>mpy</i> =1	<i>mpy</i> =2	<i>mpy</i> =3	<i>mpy</i> =4
00b	<i>sur</i> =0	0.5	0.5	0.5	0.5	1	1	1	1
01b	<i>sur</i> =1	1.5	2.5	3.5	4.5	2	3	4	5
10b	<i>sur</i> =2	2.5	4.5	6.5	8.5	3	5	7	9
11b	<i>sur</i> =3	3.5	6.5	9.5	12.5	4	7	10	13
计算式		$T_{su}(A-R)=sur \times mpy+0.5$				$T_{su}(A-R)=sur \times mpy+1$			

表 9.7 MPY1 ~ MPY0 位和 EWR1 ~ EWR0 位的设定值与  $T_w(R)$  的关系（单位：周期）

EWR1 ~ EWR0 位的设定值		分离总线				多路复用总线			
		MPY1 ~ MPY0 位的设定值				MPY1 ~ MPY0 位的设定值			
		00b	01b	10b	11b	00b	01b	10b	11b
		<i>mpy</i> =1	<i>mpy</i> =2	<i>mpy</i> =3	<i>mpy</i> =4	<i>mpy</i> =1	<i>mpy</i> =2	<i>mpy</i> =3	<i>mpy</i> =4
00b	<i>wr</i> =1	1.5	2.5	3.5	4.5	0.5 (注1)	1.5	2.5	3.5
01b	<i>wr</i> =2	2.5	4.5	6.5	8.5	1.5	3.5	5.5	7.5
10b	<i>wr</i> =3	3.5	6.5	9.5	12.5	2.5	5.5	8.5	11.5
11b	<i>wr</i> =4	4.5	8.5	12.5	16.5	3.5	7.5	11.5	15.5
计算式		$T_w(R)=wr \times mpy+0.5$				$T_w(R)=wr \times mpy-0.5$			

注 1. 不能设定。

表 9.8 MPY1 ~ MPY0 位和 ESUW1 ~ ESUW0 位的设定值与  $T_{su}(A-W)$  的关系（单位：周期）

ESUW1 ~ ESUW0 位的设定值		MPY1 ~ MPY0 位的设定值			
		00b	01b	10b	11b
		<i>mpy</i> =1	<i>mpy</i> =2	<i>mpy</i> =3	<i>mpy</i> =4
00b	<i>suw</i> =0	1	1	1	1
01b	<i>suw</i> =1	2	3	4	5
10b	<i>suw</i> =2	3	5	7	9
11b	<i>suw</i> =3	4	7	10	13
计算式		$T_{su}(A-W)=suw \times mpy+1$			

表 9.9 MPY1 ~ MPY0 位和 EWW1 ~ EWW0 位的设定值与 Tw(W) 的关系 (单位: 周期)

EWW1 ~ EWW0 位的设定值		MPY1 ~ MPY0 位的设定值			
		00b	01b	10b	11b
		<i>mpy</i> =1	<i>mpy</i> =2	<i>mpy</i> =3	<i>mpy</i> =4
00b	<i>ww</i> =1	0.5 (注 1)	1.5	2.5	3.5
01b	<i>ww</i> =2	1.5	3.5	5.5	7.5
10b	<i>ww</i> =3	2.5	5.5	8.5	11.5
11b	<i>ww</i> =4	3.5	7.5	11.5	15.5
计算式		$T_w(W) = ww \times mpy - 0.5$			

注 1. 不能设定。

MPX 位为“0”（分离总线）时的外部总线时序例子以及 MPX 位为“1”（多路复用总线）时的外部总线时序例子分别如图 9.13 和图 9.14 所示。

实际的总线周期数需要按以下情况调整为外围总线时钟周期的整数倍：

- 2 分频的情况      当总线周期数的计算值为奇数时，需要插入空闲周期，使总线周期数成为偶数。
- 3 分频的情况      当总线周期数的计算值不是 3 的倍数时，需要插入空闲周期，使总线周期数成为 3 的倍数。
- 4 分频的情况      当总线周期数的计算值不是 4 的倍数时，需要插入空闲周期，使总线周期数成为 4 的倍数。

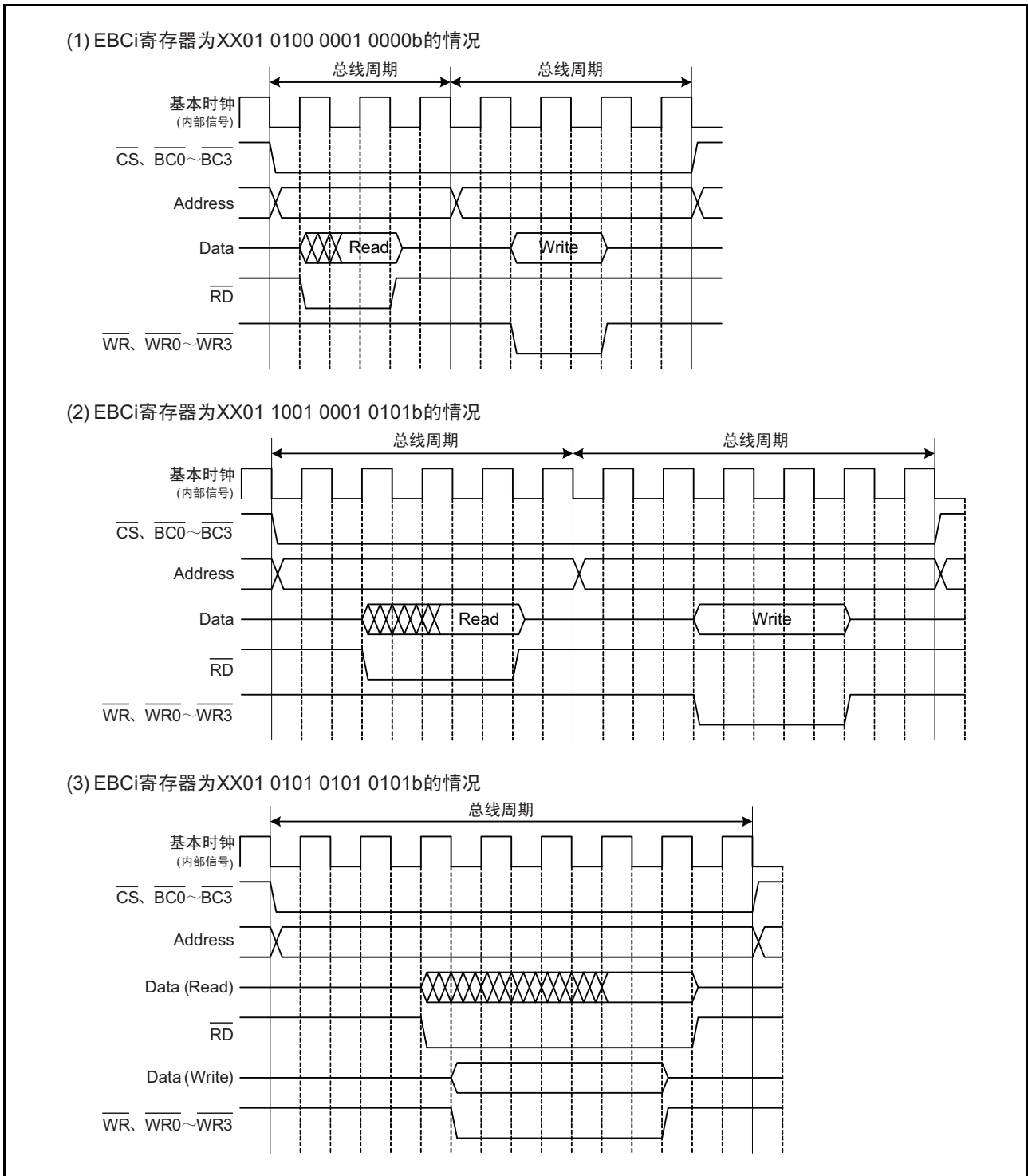


图 9.13 选择分离总线时的外部总线时序例子 (i=0 ~ 3)

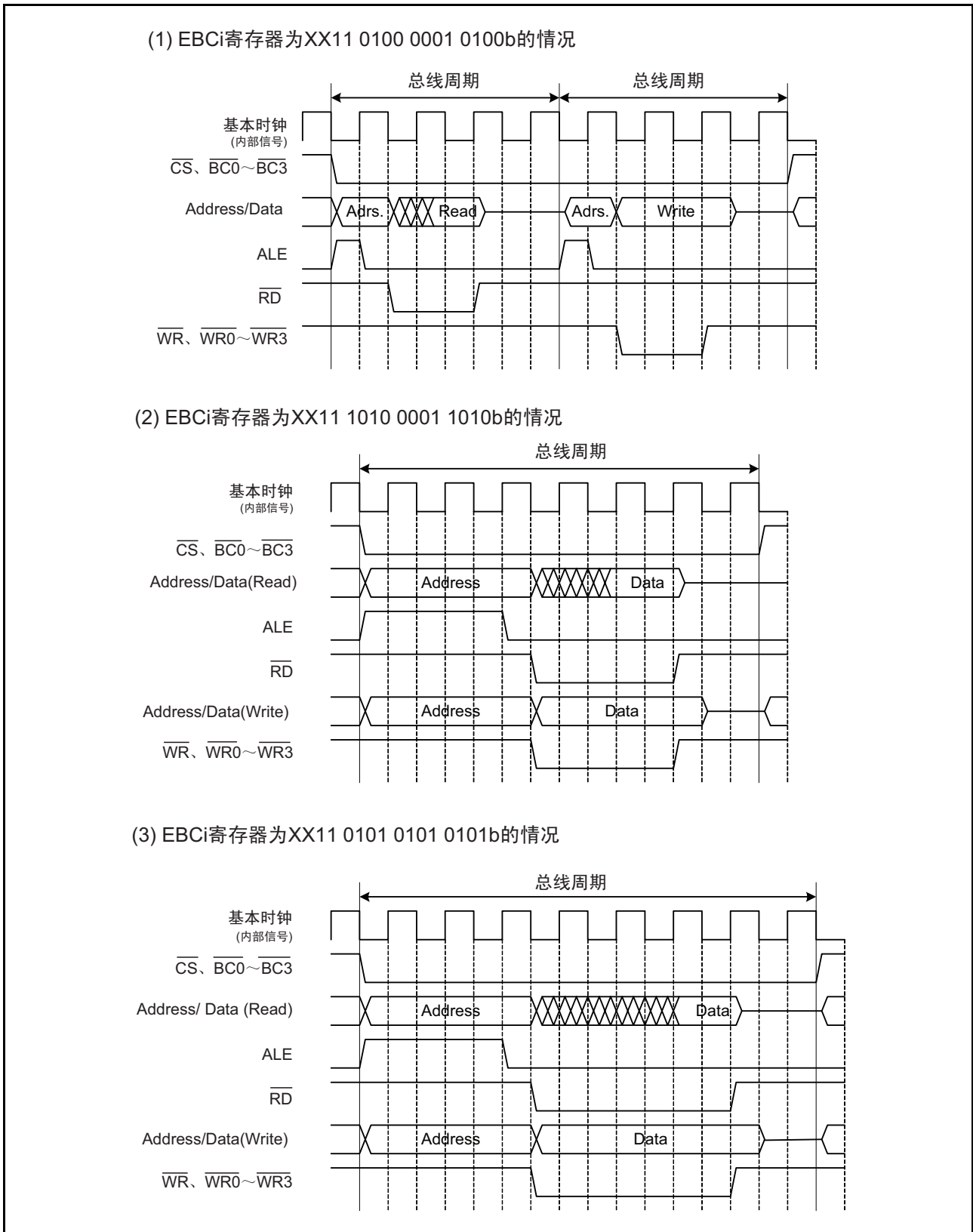


图 9.14 选择多路复用总线时的外部总线时序例子 (i=0 ~ 3)



### 9.3.6 ALE 信号

ALE 信号是用于锁存多路复用总线地址的信号，必须在 ALE 信号的下降沿锁存地址。与是存取内部区域还是存取外部区域无关，都输出 ALE 信号。

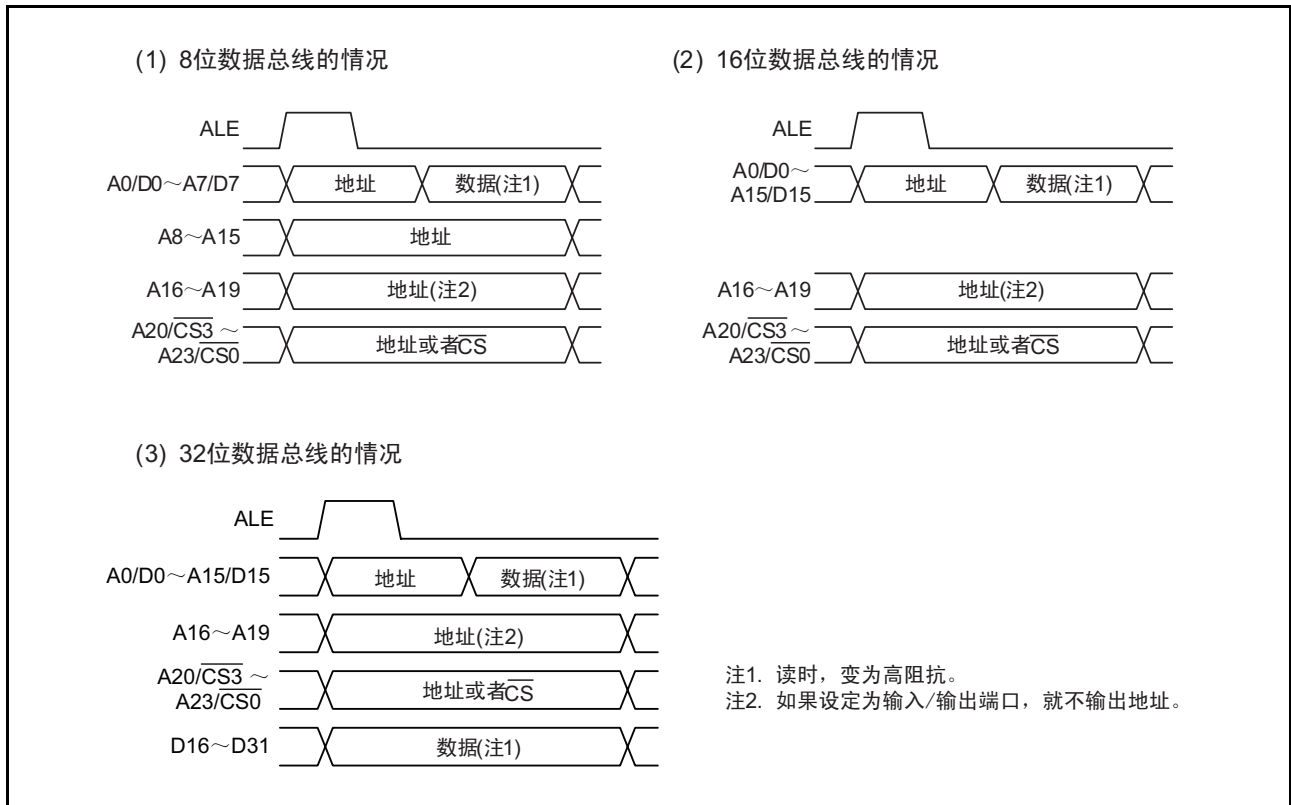


图 9.15 ALE 信号和地址总线、数据总线

与总线周期的开始同时，ALE 信号变为“H”电平，在读信号或者写信号变为“L”电平的 1/2 基本时钟前，ALE 信号变为“L”电平。

### 9.3.7 $\overline{\text{RDY}}$ 信号

在连接存取速度慢的外部器件（慢于 EBC0 ~ EBC3 寄存器设定的时序）时，或者连接 1 个  $\overline{\text{CS}}$  区域中有多个不同存取时序的设备时，外部控制器为了延长相对于 CPU 的总线周期而使用  $\overline{\text{RDY}}$  信号。

当 EBC0 ~ EBC3 寄存器的 RDY 位为“1”（使用  $\overline{\text{RDY}}$  信号）时，如果在基本时钟的下降沿每隔 *mpy* 次对  $\overline{\text{RDY}}$  引脚进行采样并输入“L”电平，就将等待插入到总线周期；如果在基本时钟的下降沿每隔 *mpy* 次确认  $\overline{\text{RDY}}$  引脚并输入“H”电平，就执行剩余的总线周期。

基本时钟没有输出到外部引脚，实际上，在  $\overline{\text{RD}}$ 、 $\overline{\text{WR}}$ 、 $\overline{\text{WR0}} \sim \overline{\text{WR3}}$  信号为“L”电平时，将  $\overline{\text{RDY}}$  信号置为“L”电平，然后与 BCLK 信号的上升沿同步，将  $\overline{\text{RDY}}$  信号置为“H”电平。

生成  $\overline{\text{RDY}}$  信号的电路例子以及使用此电路时的 EBC0 ~ EBC3 寄存器的设定条件分别如图 9.16 和表 9.10 所示，通过  $\overline{\text{RDY}}$  信号延长总线周期的例子如图 9.17 所示。

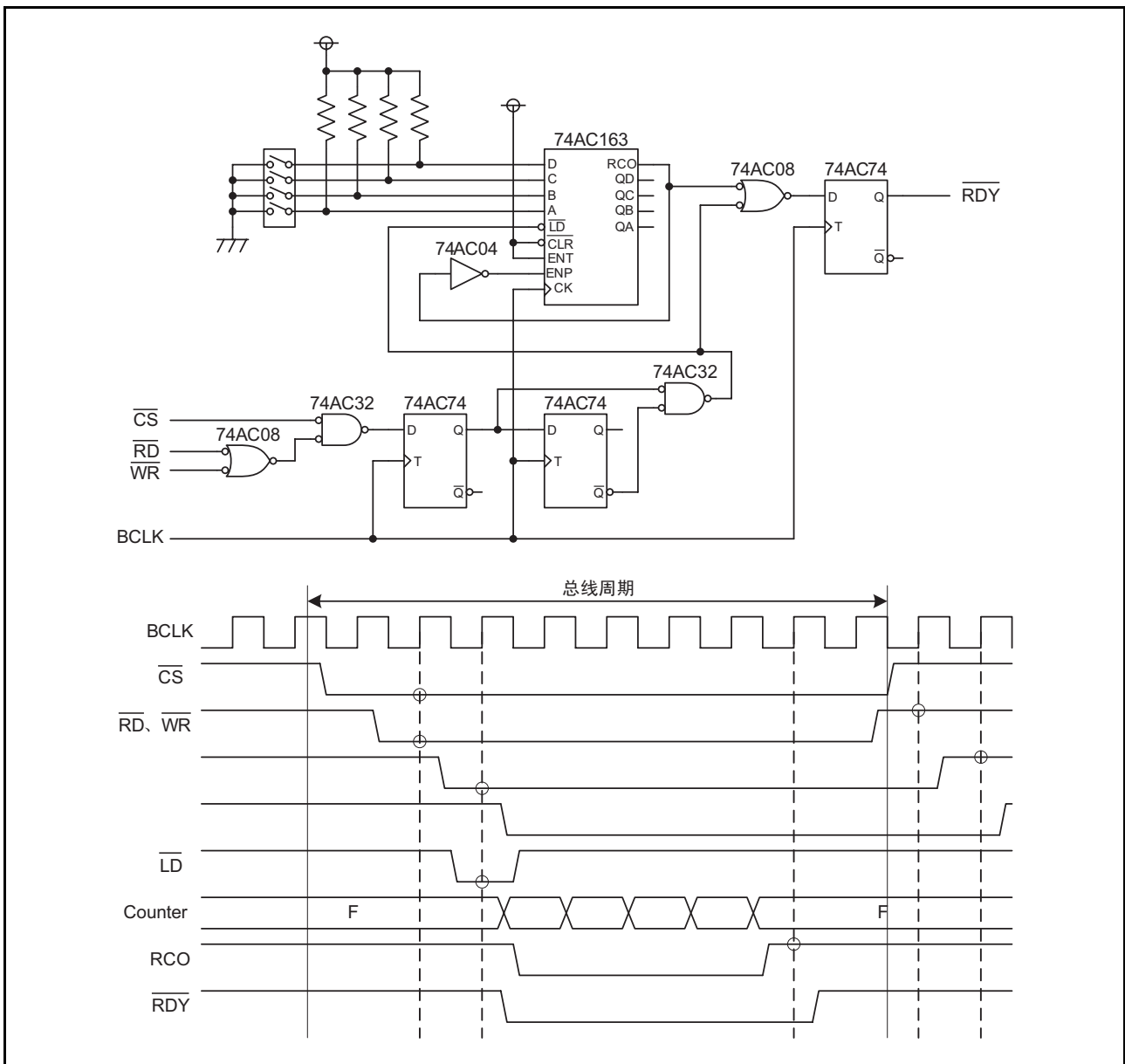


图 9.16 生成  $\overline{\text{RDY}}$  信号的电路例子

表 9.10 使用图 9.16 的电路时的 EBCi 寄存器的设定条件 (i=0 ~ 3)

外围总线时钟的频率	设定条件	设定例子
基本时钟频率的 1/2	$mpy=3$ 分离总线的情况: $\overline{RD}$ 信号的脉宽 $\geq 9.5$ $\overline{WR}$ 信号的脉宽 $\geq 11.5$ $\overline{RD}/\overline{WR}$ 信号的“H”电平宽度 $\geq 2.5$ 多路复用总线的情况: $\overline{RD}$ 信号的脉宽 $\geq 11.5$ $\overline{WR}$ 信号的脉宽 $\geq 11.5$	分离总线的情况: EBCi=XX01 1101 1011 1001b 等  多路复用总线的情况: EBCi=XX11 1101 1011 1101b 等
基本时钟频率的 1/3	$mpy=3$ 分离总线的情况: $\overline{RD}$ 信号的脉宽 $\geq 12.5$ $\overline{WR}$ 信号的脉宽 $\geq 11.5$ $\overline{RD}/\overline{WR}$ 信号的“H”电平宽度 $\geq 3.5$ 多路复用总线的情况: $\overline{RD}$ 信号的脉宽 $\geq 11.5$ $\overline{WR}$ 信号的脉宽 $\geq 11.5$	分离总线的情况: EBCi=XX01 1101 1011 1101b 等  多路复用总线的情况: EBCi=XX11 1101 1011 1101b 等
基本时钟频率的 1/4	$mpy=4$ 分离总线的情况: $\overline{RD}$ 信号的脉宽 $\geq 20.5$ $\overline{WR}$ 信号的脉宽 $\geq 19.5$ $\overline{RD}/\overline{WR}$ 信号的“H”电平宽度 $\geq 4.5$ 多路复用总线的情况: $\overline{RD}$ 信号的脉宽 $\geq 19.5$ $\overline{WR}$ 信号的脉宽 $\geq 19.5$	分离总线的情况: 不能使用  多路复用总线的情况: 不能使用

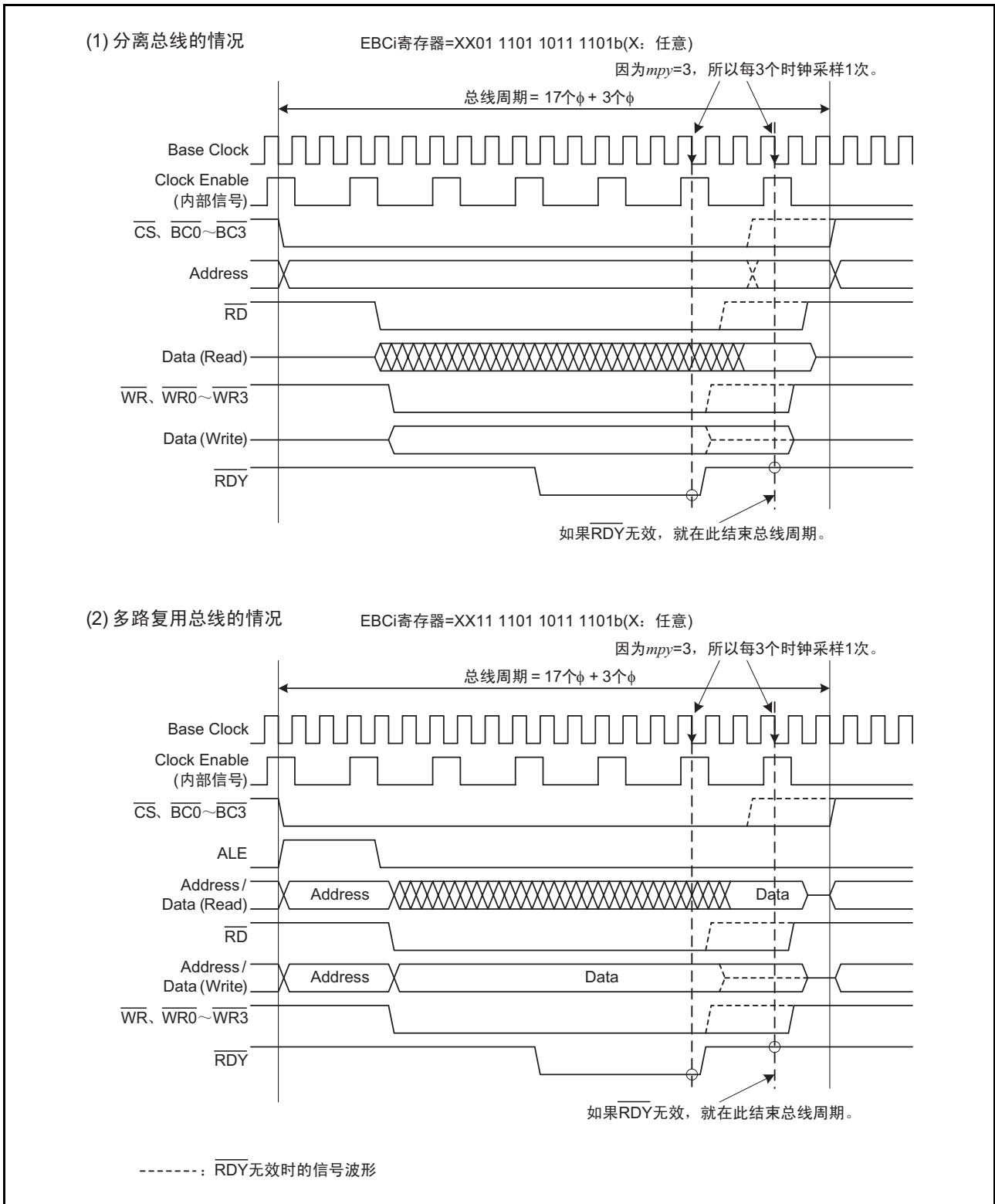


图 9.17 通过  $\overline{RDY}$  信号延长总线周期的例子 ( $f(\text{BCLK})=1/2f(\text{Base})$ ) ( $i=0 \sim 3$ )

### 9.3.8 $\overline{\text{HOLD}}$ 信号

此信号用于外部总线主控向 CPU 请求外部总线使用权。如果外部总线主控将  $\overline{\text{HOLD}}$  引脚置为“L”电平，CPU 就在结束当时的总线周期后从 HLDA 引脚输出“L”电平，并将外部总线让给外部总线主控。CPU 不在  $\overline{\text{HOLD}}$  引脚为“L”电平期间开始下一个总线周期。

在外部总线主控将外部总线使用权还给 CPU 时，将  $\overline{\text{HOLD}}$  引脚置为“H”电平。此时，必须在确认 HLDA 引脚已经变为“L”电平后将  $\overline{\text{HOLD}}$  引脚置为“H”电平。

保持中的单片机状态如表 9.11 所示。

总线使用优先权从高到低的顺序是外部总线主控、DMAC、CPU。

表 9.11 保持中的单片机状态

项目	状态
振荡	运行
地址总线、数据总线、 $\overline{\text{CS0}} \sim \overline{\text{CS3}}$ 、 $\overline{\text{BC0}} \sim \overline{\text{BC3}}$	高阻抗
$\overline{\text{RD}}$ 、 $\overline{\text{WR}}$ 、 $\overline{\text{WR0}} \sim \overline{\text{WR3}}$	高阻抗
可编程输入 / 输出端口	保持接受 $\overline{\text{HOLD}}$ 信号时的状态。
HLDA 引脚	输出“L”电平。
内部外围电路	运行（但是，看门狗定时器停止。）
ALE 引脚	输出“L”电平。

### 9.3.9 BCLK 的输出

BCLK 时钟的频率和外围总线时钟的频率相同，是对 PLL 生成的时钟进行分频后的时钟。在存储器扩展模式或者微处理器模式中，如果将 PM0 寄存器的 PM07 位置“0”（输出 BCLK）、将 CM0 寄存器的 CM01 ~ CM00 位置“00b”（输入 / 输出端口 P5\_3），就从 P5\_3 输出 BCLK。在单芯片模式中，不能输出 BCLK，详细内容请参照“8. 时钟发生电路”。

## 9.4 存取内部区域时的外部总线状态

存取内部区域时的外部总线状态如表 9.12 所示。

表 9.12 存取内部区域时的外部总线状态

引脚	存取 SFR 时的状态	存取内部存储器时的状态
地址总线	输出地址。	保持刚存取的 SFR 或者外部区域的地址。
数据总线	读	高阻抗
	写	不确定
$\overline{\text{CS0}} \sim \overline{\text{CS3}}$	输出“H”电平。	输出“H”电平。
$\overline{\text{BC0}} \sim \overline{\text{BC3}}$	输出 $\overline{\text{BC0}} \sim \overline{\text{BC3}}$ 。	保持刚存取的 SFR 或者外部区域的地址。
$\overline{\text{RD}}$ 、 $\overline{\text{WR}}$ 、 $\overline{\text{WR0}} \sim \overline{\text{WR3}}$	输出 $\overline{\text{RD}}$ 、 $\overline{\text{WR}}$ 、 $\overline{\text{WR0}} \sim \overline{\text{WR3}}$ 。	输出“H”电平。
ALE	输出 ALE 信号。	输出 ALE 信号。

## 9.5 使用总线时的注意事项

### 9.5.1 系统设计时的注意事项

在使用存储器扩展模式并在执行闪存的 CPU 改写时， $\overline{CS0}$  区域和  $\overline{CS3}$  区域有以下的限制。必须将想在 CPU 改写时存取的器件分配到  $\overline{CS1}$  区域和  $\overline{CS2}$  区域。

- 如果在 CPU 改写时设定 FEBC0 寄存器、FEBC3 寄存器，对应区域就为分离总线。如果通过多路复用总线连接外部器件，就不能存取该器件。
- 如果在 CPU 改写时设定 FEBC0 寄存器、FEBC3 寄存器，对应区域的总线时序就发生变化。根据所设定的内容，有可能不能存取外部器件。

### 9.5.2 设定寄存器时的注意事项

#### 9.5.2.1 片选边界设定寄存器

在不使用存储器扩展模式时，不能更改 CB01、CB12、CB23 寄存器的值，保持复位后的值。

在使用存储器扩展模式时，这些寄存器与是否使用各片选区域无关，必须全部设定指定范围内的值。

#### 9.5.2.2 外部总线控制寄存器

EBC0 寄存器和 EBC3 寄存器分别与 FEBC0 寄存器和 FEBC3 寄存器共用地址。如果在改写闪存时更改 FEBC0 寄存器和 FEBC3 寄存器，就必须在改写闪存结束后重新设定 EBC0 寄存器和 EBC3 寄存器。

## 10. 保护

这是为了在程序失控时保护重要的寄存器不被轻易改写的功能，有 PRCR 寄存器、PRCR2 ~ PRCR3 寄存器和 PRR 寄存器。

### 10.1 PRCR 寄存器

PRCR 寄存器及其保护的寄存器分别如图 10.1 和表 10.1 所示。

表 10.1 PRCR 寄存器保护的寄存器

位	保护的寄存器
PRC0 位	CM0 寄存器、CM1 寄存器、CM2 寄存器、PM3 寄存器
PRC1 位	PM0 寄存器、PM2 寄存器、CSOP0 寄存器、CSOP1 寄存器、CSOP2 寄存器、INVC0 寄存器、INVC1 寄存器、IOBC 寄存器、I2CMR 寄存器
PRC2 位	PLC0 寄存器、PLC1 寄存器、PD9 寄存器、P9_iS 寄存器 (i=0 ~ 7)

在将 PRC2 位置“1”（允许写）后，一旦对任意的地址进行写操作，PRC2 位就变为“0”（禁止写）。必须通过将 PRC2 位置“1”后的下一条指令更改 PD9 寄存器、P9\_iS 寄存器 (i=0 ~ 7)、PLC0 寄存器和 PLC1 寄存器，并且不能在将 PRC2 位置“1”的指令和下一条指令之间发生中断和 DMA 传送。即使对任意的地址进行写操作，PRC0 位和 PRC1 位也不变为“0”，因此必须通过程序将 PRC0 位和 PRC1 位置“0”。

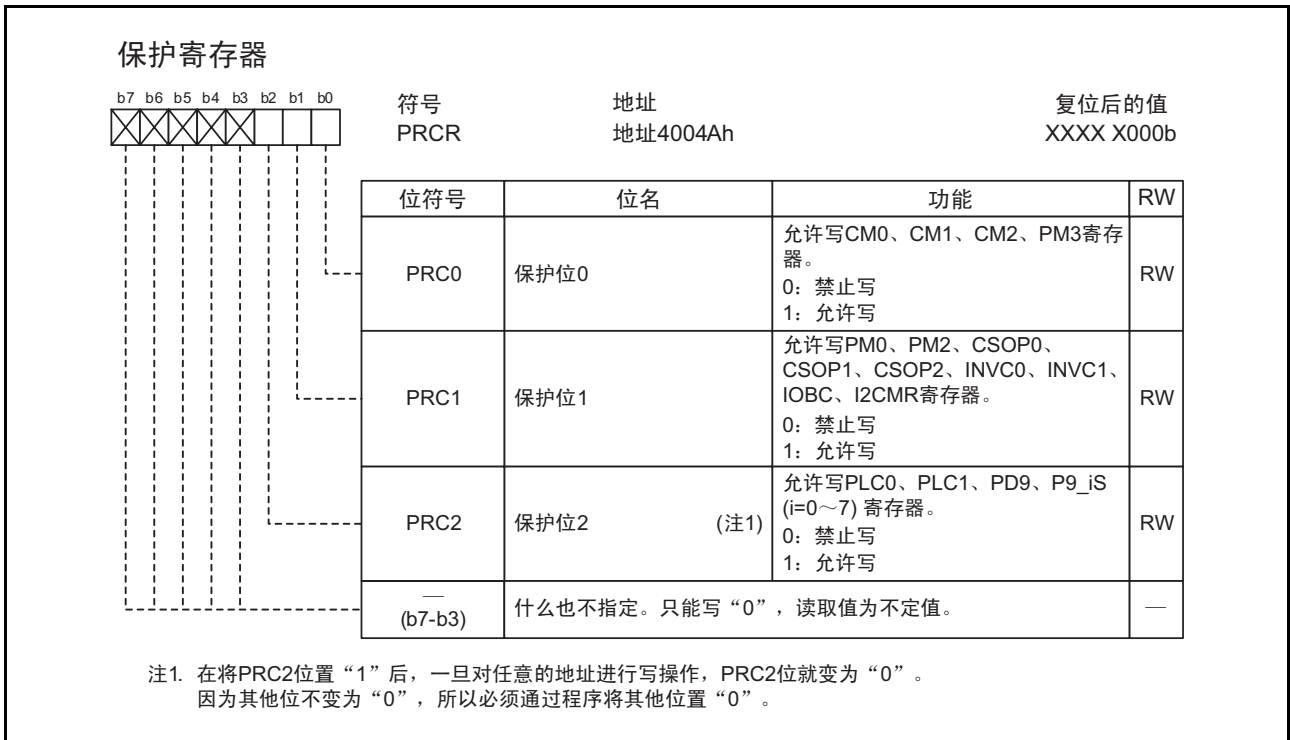


图 10.1 PRCR 寄存器

## 10.2 PRCR2 寄存器

PRCR2 寄存器如图 10.2 所示，PRCR2 寄存器保护的寄存器只有 CM3 寄存器。

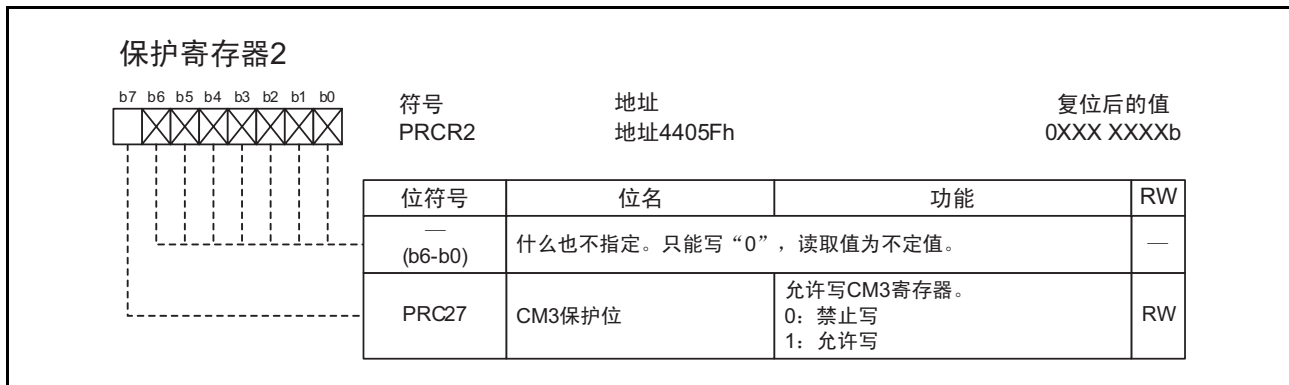


图 10.2 PRCR2 寄存器

## 10.3 PRCR3 寄存器

PRCR3 寄存器及其保护的寄存器分别如图 10.3 和表 10.2 所示。

表 10.2 PRCR3 寄存器保护的寄存器

位	保护的寄存器
PRC31 位	VRCCR 寄存器、LVDC 寄存器、DVCR 寄存器

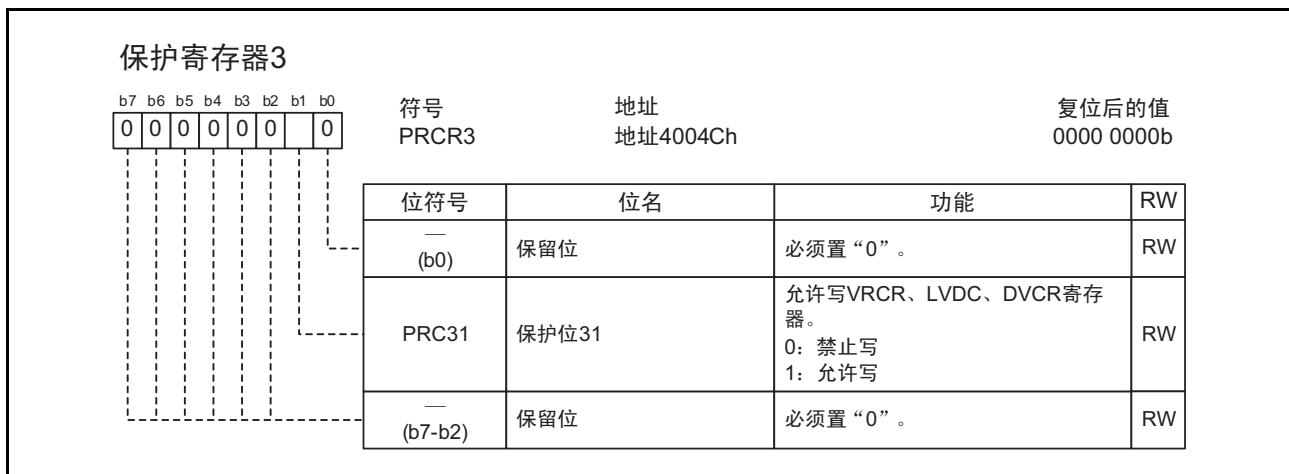


图 10.3 PRCR3 寄存器



## 10.4 PRR 寄存器

PRR 寄存器及其保护的寄存器分别如图 10.4 和表 10.3 所示。

表 10.3 PRR 寄存器保护的寄存器

CCR 寄存器、FMCR 寄存器、PBC 寄存器、FEBC0 寄存器、FEBC3 寄存器、EBC0 ~ EBC3 寄存器、CB01 寄存器、CB12 寄存器、CB23 寄存器

能在给 PRR 寄存器写 “AAh”（允许写）后写上述寄存器。在不需要给上述寄存器写数据时，为了保护上述寄存器不被意外地改写，必须给 PRR 寄存器写 “AAh” 以外的值。



图 10.4 PRR 寄存器

## 11. 中断

### 11.1 中断的分类

中断的分类如图 11.1 所示。

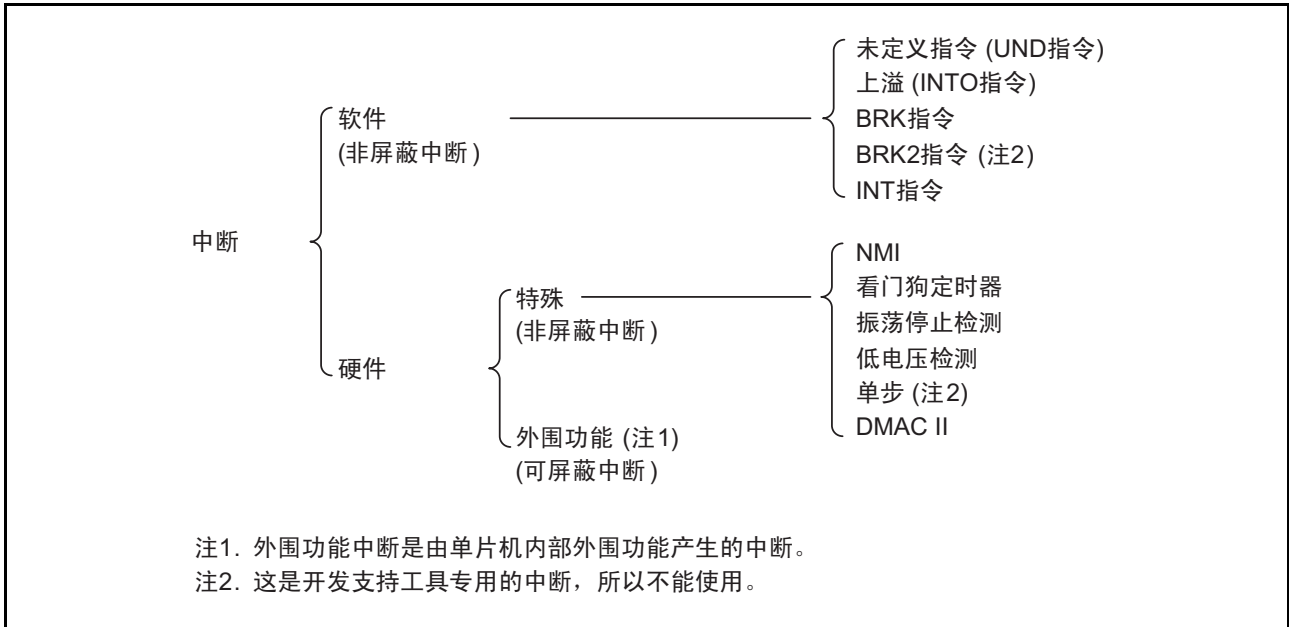


图 11.1 中断的分类

除上述分类以外，还能根据是否可以屏蔽分为可屏蔽中断和非屏蔽中断。

#### (1) 可屏蔽中断

能通过中断允许标志（I 标志）允许或者禁止中断，并且能通过中断请求级更改中断的优先级。

#### (2) 非屏蔽中断

不能通过中断允许标志（I 标志）允许或者禁止中断，也不能通过中断请求级更改中断的优先级。

### 11.2 软件中断

这是通过执行指令产生的中断，是非屏蔽中断。

软件中断有以下 5 个中断。

#### (1) 未定义指令中断

如果执行 UND 指令，就产生未定义指令中断。

#### (2) 上溢中断

当上溢标志（O 标志）为“1”时，如果执行 INTO 指令，就产生上溢中断。根据运算，O 标志发生变化的指令如下：

ABS、ADC、ADCF、ADD、ADDF、ADSF、CMP、CMPF、CNVIF、DIV、DIVF、DIVU、DIVX、EDIV、EDIVU、EDIVX、MUL、MULF、MULU、MULX、NEG、RMPA、ROUND、SBB、SCMPU、SHA、SUB、SUBF、SUNTIL、SWHILE

### (3) BRK 指令中断

如果执行 BRK 指令，就产生 BRK 指令中断。

### (4) BRK2 指令中断

如果执行 BRK2 指令，就产生 BRK2 指令中断。  
此中断是开发支持工具专用的中断，所以不能使用。

### (5) INT 指令中断

如果指定软件中断序号 0 ~ 255 并执行 INT 指令，就产生 INT 指令中断。因为软件中断序号 0 ~ 127 分配给外围功能中断，所以能通过执行 INT 指令，执行和外围功能中断相同的中断处理程序。

用于 INT 指令中断的堆栈指针（SP）因软件中断序号而不同。对于软件中断序号 0 ~ 127，在接受中断请求时将堆栈指针指定标志（U 标志）压栈，在将 U 标志置“0”并且选择中断堆栈指针（ISP）后，执行中断响应顺序。在从中断处理程序返回时，恢复接受中断请求前的 U 标志。对于软件中断序号 128 ~ 255，不能转换堆栈指针。

## 11.3 硬件中断

硬件中断包括特殊中断和外围功能中断。

只能将外围功能中断中优先级最高的 1 个中断设定为高速中断。

### 11.3.1 特殊中断

特殊中断是非屏蔽中断，有以下 5 个中断。

#### (1) NMI（Non Maskable Interrupt）

如果  $\overline{\text{NMI}}$  引脚的输入信号从“H”电平变为“L”电平，就发生 NMI。有关 NMI 的详细内容，请参照“11.11 NMI”。

#### (2) 看门狗定时器中断

这是看门狗定时器产生的中断。有关看门狗定时器中断的详细内容，请参照“12. 看门狗定时器”。

#### (3) 振荡停止检测中断

这是通过振荡停止检测功能检测到主时钟振荡停止时产生的中断。有关振荡停止检测的详细内容，请参照“8.2 振荡停止检测功能”。

#### (4) 低电压检测中断

这是通过电压检测电路功能检测到 VCC 引脚电压下降时产生的中断。有关电压检测电路的详细内容，请参照“6.2 低电压检测电路”。

#### (5) 单步中断

此中断是开发支持工具专用的中断，所以不能使用。

### 11.3.2 外围功能中断

这是单片机内部的外围功能产生的中断，中断向量表和 INT 指令使用的软件中断序号 0 ~ 127 相同，是可屏蔽中断。

有关外围功能中断的中断源，请参照表 11.2 ~ 表 11.5。有关外围功能的详细内容，请参照各功能的说明。

### 11.4 高速中断

高速中断是能高速执行中断响应的中断，只能用于外围功能中断中优先级最高的 1 个中断。

高速中断的使用步骤如下：

1. 将 RIPL1 寄存器和 RIPL2 寄存器的 FSIT 位都置 “1”（中断请求级 7 用于高速中断）。
2. 将 RIPL1 寄存器和 RIPL2 寄存器的 DMAII 位都置 “0”（中断请求级 7 用于中断）。
3. 将高速中断处理程序的起始地址设定到 VCT 寄存器。

在此状态下将中断控制寄存器的 ILVL2 ~ ILVL0 位置 “111b”（7 级）的中断为高速中断，此时，不能同时将多个中断设定为 7 级。

如果接受中断，高速中断就分别将 FLG 寄存器和 PC 保存到 SVF 寄存器和 SVP 寄存器，并且从 VCT 寄存器指向的地址开始执行程序。

必须通过执行 FREIT 指令从高速中断处理程序返回。一旦执行 FREIT 指令，保存在 SVF 寄存器和 SVP 寄存器的值就分别恢复到 FLG 寄存器和 PC。

### 11.5 中断向量

中断向量各由 4 字节构成。必须给中断向量设定中断处理程序的起始地址。如果接受中断请求，就转移到设定在中断向量的地址。中断向量的内容如图 11.2 所示。

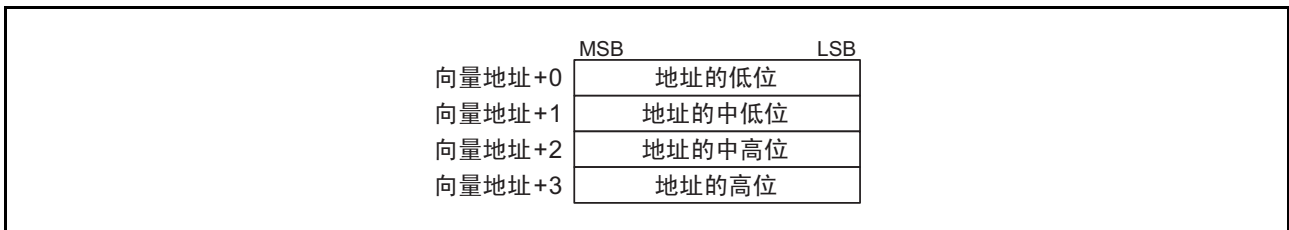


图 11.2 中断向量

### 11.5.1 固定向量表

固定向量表分配在地址 FFFFFFFDCh ~ 地址 FFFFFFFFh，如表 11.1 所示。

表 11.1 固定向量表

中断源	向量表地址 (地址 (L) ~ 地址 (H))	备注	参照
未定义指令	FFFFFFDCh ~ FFFFFFFDFh	通过 UND 指令产生中断。	R32C/100 系列软件手册
上溢	FFFFFFE0h ~ FFFFFFFE3h	通过 INTO 指令产生中断。	
BRK 指令	FFFFFFE4h ~ FFFFFFFE7h	当地址 FFFFFFFE7h 为“FFh”时，转移到可变向量表中软件中断序号为 0 的向量所指向的地址。	
—	FFFFFFE8h ~ FFFFFFFEBh	保留区	
—	FFFFFFECh ~ FFFFFFFEFh	保留区	
看门狗定时器、振荡停止检测、低电压检测	FFFFFFF0h ~ FFFFFFFF3h	看门狗定时器中断、振荡停止检测中断和低电压检测中断共用向量表地址。	12. 看门狗定时器、 8. 时钟发生电路、 6.2 低电压检测电路
—	FFFFFFF4h ~ FFFFFFFF7h	保留区	
NMI	FFFFFFF8h ~ FFFFFFFFBh	NMI 引脚产生的外部中断。	
复位	FFFFFFFCh ~ FFFFFFFFh		5. 复位

### 11.5.2 可变向量表

从设定在 INTB 寄存器的起始地址开始的 1024 字节为可变向量表区域如表 11.2 ~ 表 11.5 所示。

INTB 寄存器的设定值必须是 4 的倍数，设定 4 的倍数能提高中断响应顺序的执行速度。

表 11.2 可变向量表 (1)

中断源	向量表相对地址 地址 (L) ~ 地址 (H) (注 1)	软件中断 序号	参照
BRK 指令 (注 2)	+0 ~ +3 (0000h ~ 0003h)	0	R32C/100 系列软件手册
保留区	+4 ~ +7 (0004h ~ 0007h)	1	
UART5 发送、NACK (注 3)	+8 ~ +11 (0008h ~ 000Bh)	2	18. 串行接口
UART5 接收、ACK (注 3)	+12 ~ +15 (000Ch ~ 000Fh)	3	
UART6 发送、NACK (注 3)	+16 ~ +19 (0010h ~ 0013h)	4	
UART6 接收、ACK (注 3)	+20 ~ +23 (0014h ~ 0017h)	5	
总线冲突检测、开始条件检测或者停止条件检测 (UART5 或者 UART6) (注 3、注 4)	+24 ~ +27 (0018h ~ 001Bh)	6	
保留区	+28 ~ +31 (001Ch ~ 001Fh)	7	
DMA0 传送结束	+32 ~ +35 (0020h ~ 0023h)	8	13. DMAC
DMA1 传送结束	+36 ~ +39 (0024h ~ 0027h)	9	
DMA2 传送结束	+40 ~ +43 (0028h ~ 002Bh)	10	
DMA3 传送结束	+44 ~ +47 (002Ch ~ 002Fh)	11	
定时器 A0	+48 ~ +51 (0030h ~ 0033h)	12	
定时器 A1	+52 ~ +55 (0034h ~ 0037h)	13	
定时器 A2	+56 ~ +59 (0038h ~ 003Bh)	14	
定时器 A3	+60 ~ +63 (003Ch ~ 003Fh)	15	
定时器 A4	+64 ~ +67 (0040h ~ 0043h)	16	
UART0 发送、NACK (注 3)	+68 ~ +71 (0044h ~ 0047h)	17	18. 串行接口
UART0 接收、ACK (注 3)	+72 ~ +75 (0048h ~ 004Bh)	18	
UART1 发送、NACK (注 3)	+76 ~ +79 (004Ch ~ 004Fh)	19	
UART1 接收、ACK (注 3)	+80 ~ +83 (0050h ~ 0053h)	20	
定时器 B0	+84 ~ +87 (0054h ~ 0057h)	21	16.2 定时器 B
定时器 B1	+88 ~ +91 (0058h ~ 005Bh)	22	
定时器 B2	+92 ~ +95 (005Ch ~ 005Fh)	23	
定时器 B3	+96 ~ +99 (0060h ~ 0063h)	24	
定时器 B4	+100 ~ +103 (0064h ~ 0067h)	25	
INT5	+104 ~ +107 (0068h ~ 006Bh)	26	11.10 外部中断
INT4	+108 ~ +111 (006Ch ~ 006Fh)	27	
INT3	+112 ~ +115 (0070h ~ 0073h)	28	
INT2	+116 ~ +119 (0074h ~ 0077h)	29	
INT1	+120 ~ +123 (0078h ~ 007Bh)	30	
INT0	+124 ~ +127 (007Ch ~ 007Fh)	31	
定时器 B5	+128 ~ +131 (0080h ~ 0083h)	32	16.2 定时器 B

注 1. 这是从 INTB 寄存器指向的地址开始的相对地址。

注 2. 不能通过 I 标志禁止中断。

注 3. 在 I<sup>2</sup>C 模式中, NACK、ACK 或者开始条件检测 / 停止条件检测为中断源。

注 4. 必须通过 IFSR1 寄存器的 IFSR16 位选择是用于 UART5 还是用于 UART6 的中断。

表 11.3 可变向量表 (2)

中断源	向量表相对地址 地址 (L) ~ 地址 (H) (注 1)	软件中断 序号	参照
UART2 发送、NACK (注 2) /I <sup>2</sup> C 总线接口 (注 3)	+132 ~ +135 (0084h ~ 0087h)	33	18. 串行接口 /24. 多主控 I <sup>2</sup> C 总线接口
UART2 接收、ACK (注 2) /I <sup>2</sup> C 总线接口 (注 3)	+136 ~ +139 (0088h ~ 008Bh)	34	
UART3 发送、NACK (注 2)	+140 ~ +143 (008Ch ~ 008Fh)	35	
UART3 接收、ACK (注 2)	+144 ~ +147 (0090h ~ 0093h)	36	
UART4 发送、NACK (注 2)	+148 ~ +151 (0094h ~ 0097h)	37	
UART4 接收、ACK (注 2)	+152 ~ +155 (0098h ~ 009Bh)	38	
总线冲突检测、开始条件检测或者停止条件检测 (UART2) (注 2)	+156 ~ +159 (009Ch ~ 009Fh)	39	
总线冲突检测、开始条件检测或者停止条件检测 (UART3 或者 UART0) (注 2、注 4)	+160 ~ +163 (00A0h ~ 00A3h)	40	
总线冲突检测、开始条件检测或者停止条件检测 (UART4 或者 UART1) (注 2、注 4)	+164 ~ +167 (00A4h ~ 00A7h)	41	
A/D0	+168 ~ +171 (00A8h ~ 00ABh)	42	
键输入	+172 ~ +175 (00ACh ~ 00AFh)	43	11.12 键输入中断
智能 I/O 中断 0	+176 ~ +179 (00B0h ~ 00B3h)	44	11.13 智能 I/O 中断、 23. 智能 I/O
智能 I/O 中断 1	+180 ~ +183 (00B4h ~ 00B7h)	45	
智能 I/O 中断 2	+184 ~ +187 (00B8h ~ 00BBh)	46	
智能 I/O 中断 3	+188 ~ +191 (00BCh ~ 00BFh)	47	
智能 I/O 中断 4	+192 ~ +195 (00C0h ~ 00C3h)	48	
智能 I/O 中断 5	+196 ~ +199 (00C4h ~ 00C7h)	49	
智能 I/O 中断 6	+200 ~ +203 (00C8h ~ 00CBh)	50	
智能 I/O 中断 7	+204 ~ +207 (00CCh ~ 00CFh)	51	
智能 I/O 中断 8	+208 ~ +211 (00D0h ~ 00D3h)	52	
智能 I/O 中断 9	+212 ~ +215 (00D4h ~ 00D7h)	53	
智能 I/O 中断 10	+216 ~ +219 (00D8h ~ 00DBh)	54	
智能 I/O 中断 11	+220 ~ +223 (00DCh ~ 00DFh)	55	
保留区	+224 ~ +227 (00E0h ~ 00E3h)	56	25. CAN 模块
保留区	+228 ~ +231 (00E4h ~ 00E7h)	57	
CAN0 唤醒	+232 ~ +235 (00E8h ~ 00EBh)	58	
保留区	+236 ~ +239 (00ECh ~ 00EFh)	59	
保留区	+240 ~ +243 (00F0h ~ 00F3h)	60	
保留区	+244 ~ +247 (00F4h ~ 00F7h)	61	
保留区	+248 ~ +251 (00F8h ~ 00FBh)	62	
保留区	+252 ~ +255 (00FCh ~ 00FFh)	63	

注 1. 这是从 INTB 寄存器指向的地址开始的相对地址。

注 2. 在 I<sup>2</sup>C 模式中，NACK、ACK 或者开始条件检测 / 停止条件检测为中断源。

注 3. 由 I2CMR 寄存器的 I2CEN 位选择使用 UART2 或者 I<sup>2</sup>C 总线。

注 4. 必须通过 IFSR0 寄存器的 IFSR06 位选择是用于 UART0 还是用于 UART3 的中断，或者通过 IFSR07 位选择是用于 UART1 还是用于 UART4 的中断。

表 11.4 可变向量表 (3) (注 1)

中断源	向量表相对地址 地址 (L) ~ 地址 (H) (注 2)	软件中断 序号	参照
保留区	+256 ~ +259 (0100h ~ 0103h)	64	
保留区	+260 ~ +263 (0104h ~ 0107h)	65	
保留区	+264 ~ +267 (0108h ~ 010Bh)	66	
保留区	+268 ~ +271 (010Ch ~ 010Fh)	67	
保留区	+272 ~ +275 (0110h ~ 0113h)	68	
保留区	+276 ~ +279 (0114h ~ 0117h)	69	
保留区	+280 ~ +283 (0118h ~ 011Bh)	70	
保留区	+284 ~ +287 (011Ch ~ 011Fh)	71	
保留区	+288 ~ +291 (0120h ~ 0123h)	72	
保留区	+292 ~ +295 (0124h ~ 0127h)	73	
保留区	+296 ~ +299 (0128h ~ 012Bh)	74	
保留区	+300 ~ +303 (012Ch ~ 012Fh)	75	
保留区	+304 ~ +307 (0130h ~ 0133h)	76	
保留区	+308 ~ +311 (0134h ~ 0137h)	77	
保留区	+312 ~ +315 (0138h ~ 013Bh)	78	
保留区	+316 ~ +319 (013Ch ~ 013Fh)	79	
CAN0 发送 FIFO	+320 ~ +323 (0140h ~ 0143h)	80	25. CAN 模块
CAN0 接收 FIFO	+324 ~ +327 (0144h ~ 0147h)	81	
保留区	+328 ~ +331 (0148h ~ 014Bh)	82	
保留区	+332 ~ +335 (014Ch ~ 014Fh)	83	
保留区	+336 ~ +339 (0150h ~ 0153h)	84	
保留区	+340 ~ +343 (0154h ~ 0157h)	85	
保留区	+344 ~ +347 (0158h ~ 015Bh)	86	
保留区	+348 ~ +351 (015Ch ~ 015Fh)	87	
保留区	+352 ~ +355 (0160h ~ 0163h)	88	
保留区	+356 ~ +359 (0164h ~ 0167h)	89	
保留区	+360 ~ +363 (0168h ~ 016Bh)	90	
保留区	+364 ~ +367 (016Ch ~ 016Fh)	91	
保留区	+368 ~ +371 (0170h ~ 0173h)	92	
INT8	+372 ~ +375 (0174h ~ 0177h)	93	11.10 外部中断
INT7	+376 ~ +379 (0178h ~ 017Bh)	94	
INT6	+380 ~ +383 (017Ch ~ 017Fh)	95	

注 1. 不能用于从等待模式或者停止模式的返回。

注 2. 这是从 INTB 寄存器指向的地址开始的相对地址。



表 11.5 可变向量表 (4) (注 1)

中断源	向量表相对地址 地址 (L) ~ 地址 (H) (注 2)	软件中断 序号	参照	
CAN0 发送	+384 ~ +387 (0180h ~ 0183h)	96	25. CAN 模块	
CAN0 接收	+388 ~ +391 (0184h ~ 0187h)	97		
CAN0 错误	+392 ~ +395 (0188h ~ 018Bh)	98		
保留区	+396 ~ +399 (018Ch ~ 018Fh)	99		
保留区	+400 ~ +403 (0190h ~ 0193h)	100		
保留区	+404 ~ +407 (0194h ~ 0197h)	101		
保留区	+408 ~ +411 (0198h ~ 019Bh)	102		
保留区	+412 ~ +415 (019Ch ~ 019Fh)	103		
保留区	+416 ~ +419 (01A0h ~ 01A3h)	104		
保留区	+420 ~ +423 (01A4h ~ 01A7h)	105		
保留区	+424 ~ +427 (01A8h ~ 01ABh)	106		
保留区	+428 ~ +431 (01ACh ~ 01AFh)	107		
保留区	+432 ~ +435 (01B0h ~ 01B3h)	108		
保留区	+436 ~ +439 (01B4h ~ 01B7h)	109		
保留区	+440 ~ +443 (01B8h ~ 01BBh)	110		
保留区	+444 ~ +447 (01BCh ~ 01BFh)	111		
保留区	+448 ~ +451 (01C0h ~ 01C3h)	112		
保留区	+452 ~ +455 (01C4h ~ 01C7h)	113		
保留区	+456 ~ +459 (01C8h ~ 01CBh)	114		
保留区	+460 ~ +463 (01CCh ~ 01CFh)	115		
保留区	+464 ~ +467 (01D0h ~ 01D3h)	116		18. 串行接口
保留区	+468 ~ +471 (01D4h ~ 01D7h)	117		
保留区	+472 ~ +475 (01D8h ~ 01DBh)	118		
保留区	+476 ~ +479 (01DCh ~ 01DFh)	119		
保留区	+480 ~ +483 (01E0h ~ 01E3h)	120		
保留区	+484 ~ +487 (01E4h ~ 01E7h)	121		
保留区	+488 ~ +491 (01E8h ~ 01EBh)	122		
保留区	+492 ~ +495 (01ECh ~ 01EFh)	123		
UART7 发送	+496 ~ +499 (01F0h ~ 01F3h)	124		
UART7 接收	+500 ~ +503 (01F4h ~ 01F7h)	125	11.2 软件中断	
UART8 发送	+504 ~ +507 (01F8h ~ 01FBh)	126		
UART8 接收	+508 ~ +511 (01FCh ~ 01FFh)	127		
INT 指令 (注 3)	+0 ~ +3 (0000h ~ 0003h) ~ +1020 ~ +1023 (03FCh ~ 03FFh)	0 ~ 255		

注 1. 不能用于从等待模式或者停止模式的返回。

注 2. 这是从 INTB 寄存器指向的地址开始的相对地址。

注 3. 不能通过 I 标志禁止中断。

## 11.6 中断请求的接受

对于软件中断和特殊中断，一旦发生中断请求，就无条件接受。

在以下 3 个条件全部成立时接受外围功能中断：

- I 标志=1
- IR 位=1
- ILVL2 ~ ILVL0 位 > IPL

I 标志、IPL、IR 位和 ILVL2 ~ ILVL0 位各自独立，互不影响。I 标志和 IPL 在 FLG 寄存器中，IR 位和 ILVL2 ~ ILVL0 位在中断控制寄存器中。

以下说明这些标志和位。

### 11.6.1 I 标志和 IPL

通过 I 标志（中断允许标志）允许或者禁止可屏蔽中断。如果将 I 标志置“1”（允许），就允许全部的可屏蔽中断；如果置“0”（禁止），就禁止全部的可屏蔽中断。I 标志在复位后为“0”（禁止）。

IPL（处理器中断优先级）由 3 位构成，0 ~ 7 级的 8 个处理器中断优先级如下所示。如果请求中断的中断请求级（ILVL2 ~ ILVL0）高于 IPL，就允许该中断。

根据 IPL 的内容允许的中断请求级如表 11.6 所示。

表 11.6 处理器中断优先级（IPL）的内容和能接受的中断请求级

处理器中断优先级（IPL）			能接受的中断请求级
IPL2	IPL1	IPL0	
1	1	1	禁止全部的可屏蔽中断。
1	1	0	只允许 7 级
1	0	1	允许 6 级以上（含 6 级）
1	0	0	允许 5 级以上（含 5 级）
0	1	1	允许 4 级以上（含 4 级）
0	1	0	允许 3 级以上（含 3 级）
0	0	1	允许 2 级以上（含 2 级）
0	0	0	允许 1 级以上（含 1 级）

## 11.6.2 中断控制寄存器

中断控制寄存器控制各外围功能中断。

中断控制寄存器如图 11.3 和图 11.4 所示。

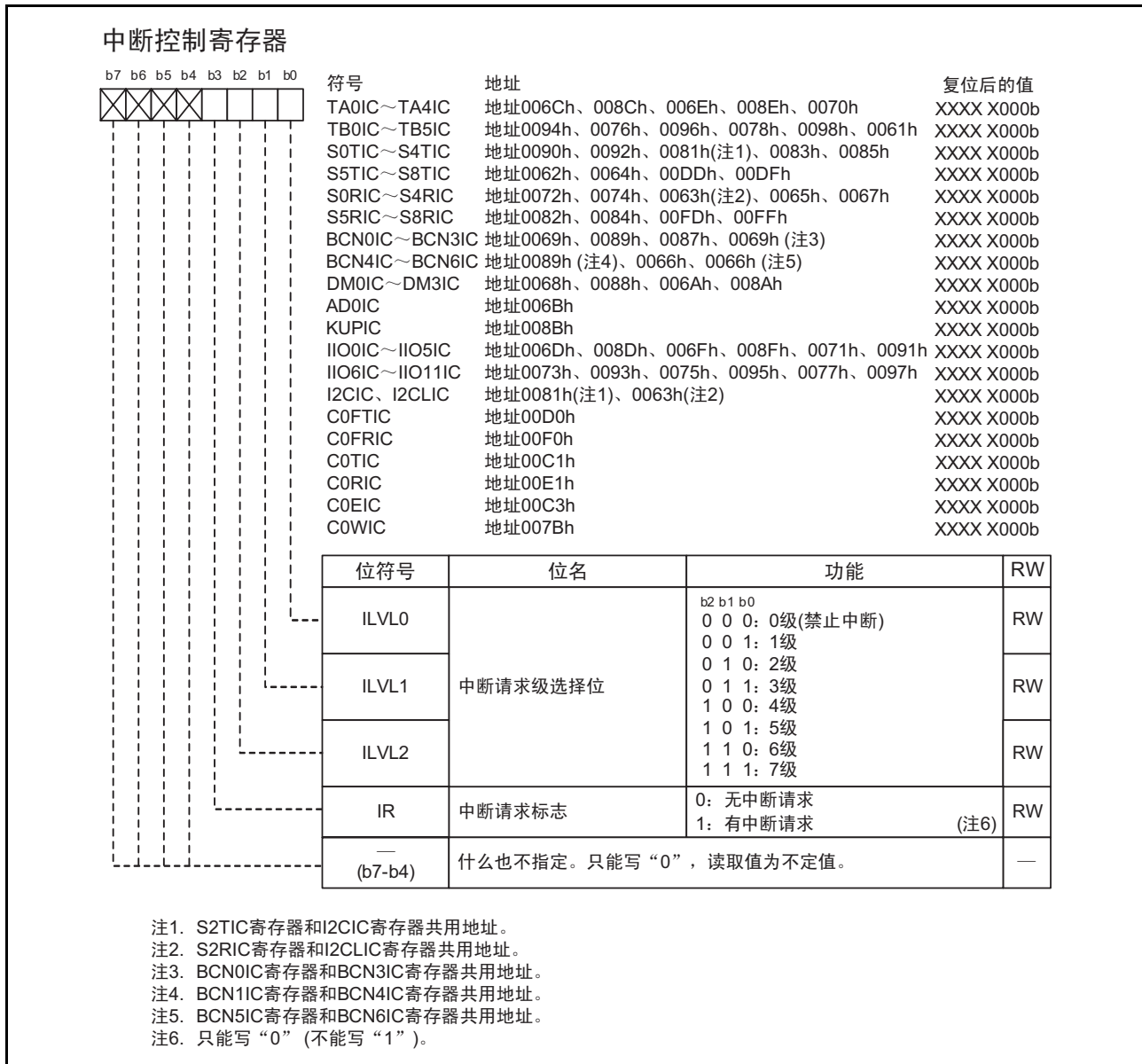


图 11.3 中断控制寄存器 (1)

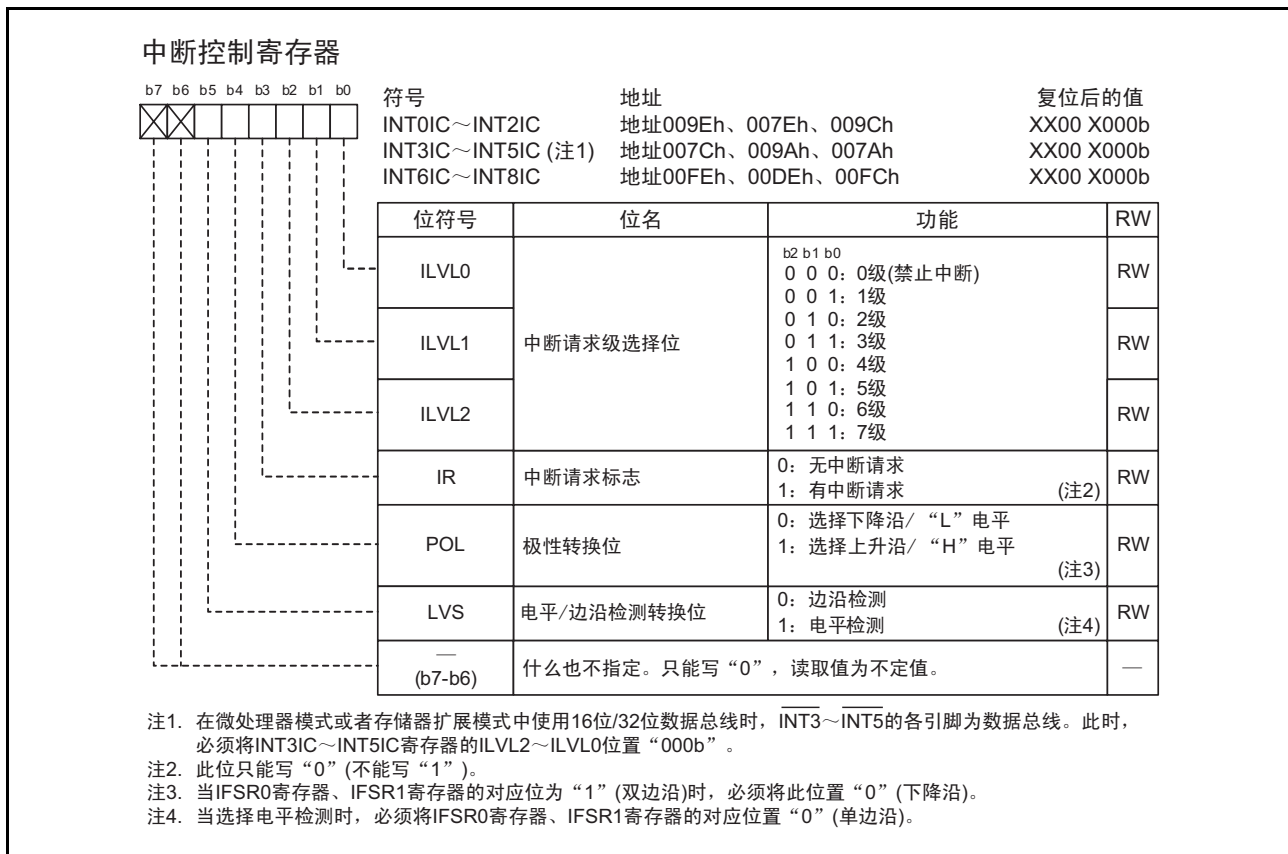


图 11.4 中断控制寄存器（2）

### ILVL2 ~ ILVL0 位

通过 ILVL2 ~ ILVL0 位选择中断请求级。中断请求级的值越大，中断优先级就越高。

在发生中断请求时，比较中断请求级和 IPL，只有在中断请求级高于 IPL 时，才允许该中断。如果将 ILVL2 ~ ILVL0 位置“000b”，就禁止该中断。

### IR 位

如果发生中断请求，IR 位就变为“1”（有中断请求），并保持到中断请求被接受为止。在接受中断请求并转移到对应的中断向量后，IR 位变为“0”（无中断请求）。

能通过程序将 IR 位置“0”（不能写“1”）。

必须在不发生与该寄存器对应的中断请求的位置，更改中断控制寄存器。如果有可能发生中断请求，就必须在禁止中断后进行更改。

在更改中断控制寄存器后立即允许中断时，为了在中断控制寄存器的写操作结束前不使中断允许标志（I 标志）变为“1”（允许中断），必须在 2 条指令之间插入 NOP 或者虚读中断控制寄存器等。

如果正在执行中断控制寄存器的改写指令时发生与该寄存器对应的中断请求，根据指令，IR 位就可能不变为“1”（有中断请求）。如果出现问题，就必须使用以下的指令改写寄存器。

- AND
- OR
- BCLR
- BSET

在将 IR 位置“0”（无中断请求）时，为了通过 AND 指令或者 BCLR 指令保持正在改写时的中断请求，IR 位有可能不变为“0”。如果因此而引起问题，就必须使用 MOV 指令改写寄存器。如果只要将 IR 位置“0”，必须先暂时将值读到存储器，然后在存储器中执行 AND 指令或者 BCLR 指令并通过 MOV 指令回写存储器的值。

### 11.6.3 用于返回的中断优先级设定寄存器

在使用中断从等待模式或者停止模式返回时或者在使用高速中断时，使用用于返回的中断优先级设定寄存器（RIPL1 寄存器和 RIPL2 寄存器）。

有关从等待模式或者停止模式的返回，请参照“8.7.2 等待模式”和“8.7.3 停止模式”。有关高速中断，请参照“11.4 高速中断”。

RIPL1 寄存器和 RIPL2 寄存器如图 11.5 所示。

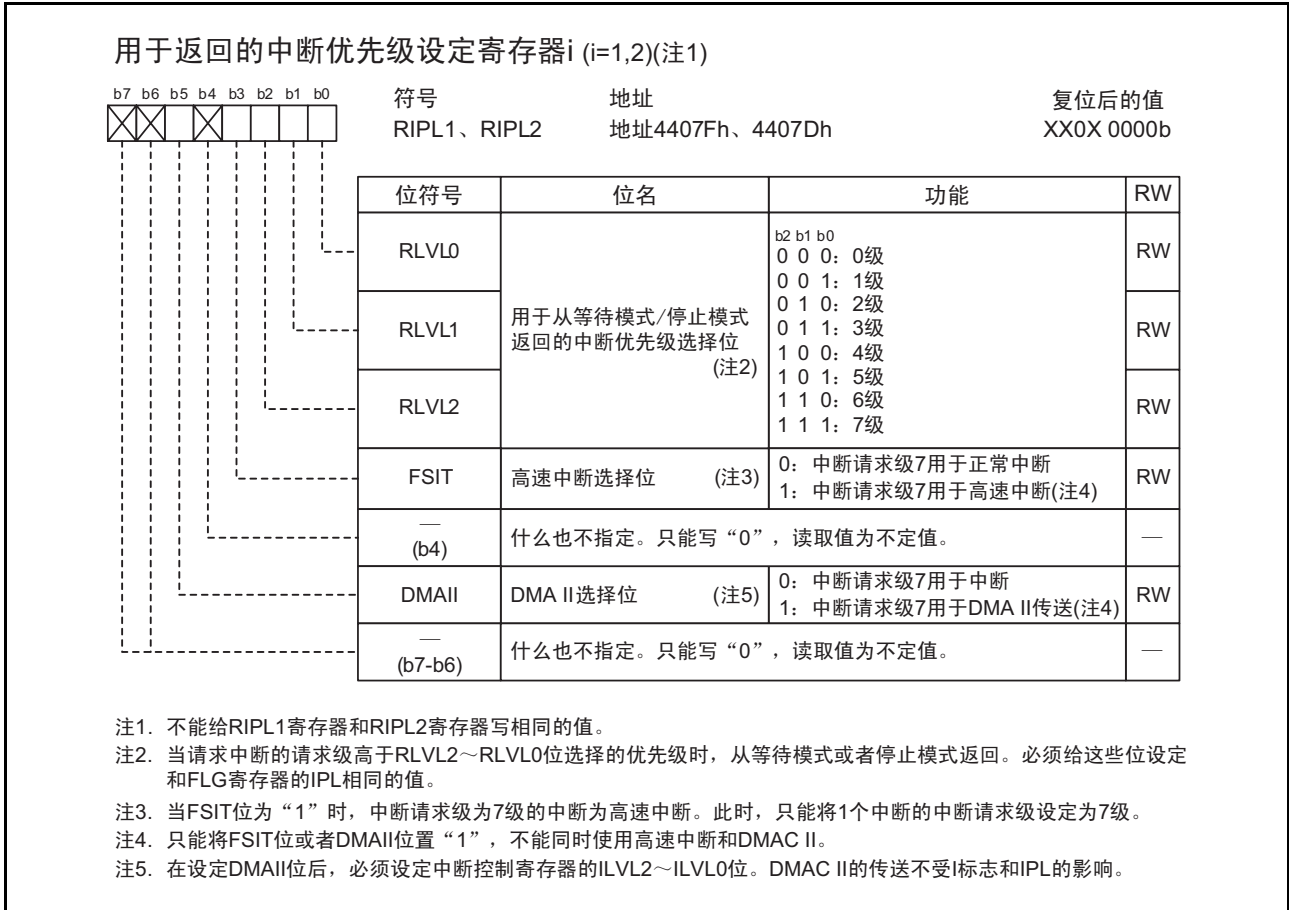


图 11.5 RIPL1 寄存器和 RIPL2 寄存器

### 11.6.4 中断响应顺序

以下说明从接受中断请求到执行中断处理程序为止的中断响应顺序。

如果在执行指令过程中发生中断请求，CPU 就在执行该指令后判断优先级，从下一个周期转移到中断响应顺序。但是，如果在执行 RMPA、SCMPU、SIN、SMOVB、SMOVF、SMOVU、SOUT、SSTR、SUNTIL、SWHILE 各指令的过程中发生中断请求，就暂时中断指令的运行，并转移到中断响应顺序。

中断响应顺序的运行如下：

1. CPU 通过返回中断应答从中断控制器取得中断信息（中断序号和中断请求级），然后对应中断的 IR 位变为“0”（无中断请求）。
2. 将中断响应顺序前的 FLG 寄存器的内容保存到 CPU 内部的临时寄存器（用户不能使用）。
3. FLG 寄存器中各位的状态如下：
  - I 标志（中断允许标志）为“0”（禁止中断）。
  - D 标志（调试标志）为“0”（禁止单步中断）。
  - U 标志（堆栈指针指定标志）为“0”（指定 ISP）。
4. 将 CPU 内部的临时寄存器（用户不能使用）的内容压栈。在高速中断的情况下，保存到 SVF（标志保存寄存器）。
5. 将 PC（程序计数器）的内容压栈。在高速中断的情况下，保存到 SVP（PC 保存寄存器）。
6. 给 IPL（处理器中断优先级）设定已接受中断的中断请求级。
7. 从中断向量表中获取已接受中断源的向量。
8. 将获取的中断向量设定到 PC（程序计数器）。

在中断响应顺序结束后，从中断处理程序的起始地址开始执行指令。

### 11.6.5 中断响应时间

中断响应时间是指从发生中断请求开始到执行中断处理程序内的第一条指令为止的时间，详细内容如图 11.6 所示，由发生中断请求开始到正在执行的指令结束为止的时间 (a) 和执行中断响应顺序的时间 (b) 构成。

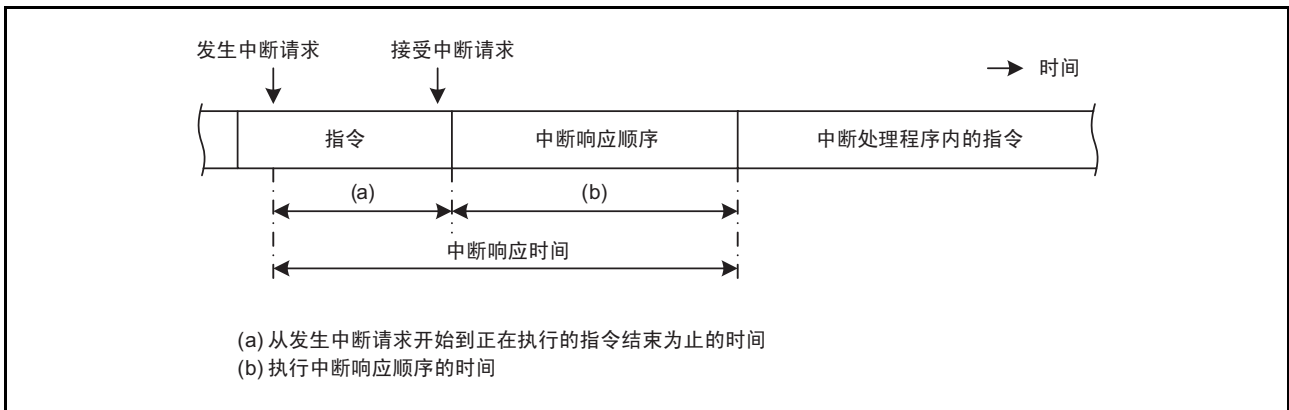


图 11.6 中断响应时间

(a) 的时间因正在执行的指令而不同。(a) 时间最长的指令是将寄存器一次性压栈或者出栈的 LDCTX、STCTX 等指令。在使用 STCTX 指令的情况下，当要压栈的寄存器有 10 个寄存器时，(a) 的时间最短是 30 个周期。如果堆栈区中有等待，(a) 的时间就会进一步增加。

(b) 的时间如表 11.7 所示。

表 11.7 中断响应顺序的执行时间（注 1）

中断	执行时间（换算为 CPU 时钟）
外围功能	13+ $\alpha$ 个周期（注 2）
INT 指令	11 个周期
NMI	10 个周期
看门狗定时器、振荡停止检测、低电压检测	11 个周期
未定义指令	12 个周期
上溢	12 个周期
BRK 指令（可变向量表）	16 个周期
BRK 指令（固定向量表）	19 个周期
BRK2 指令	19 个周期
高速中断	11 个周期

注 1. 这是将中断向量分配到 4 倍内部 ROM 地址时的值，但是高速中断除外。

注 2.  $\alpha$  为“SFR 的等待数 -2”。

### 11.6.6 接受中断请求时的 IPL 变化

如果接受外围功能中断请求，就给 IPL（处理器中断优先级）设定已接受中断的中断请求级。

软件中断和特殊中断没有中断请求级。在接受这些中断请求时，给 IPL 设定如表 11.8 所示的值。

表 11.8 没有中断请求级的中断和 IPL 的关系

没有中断请求级的中断源	IPL 的设定值
NMI、看门狗定时器、振荡停止检测、低电压检测	7
复位	0
软件	不变

### 11.6.7 寄存器的压栈

在中断响应顺序中，只将标志寄存器（FLG）和程序计数器（PC）的内容压栈。压栈的顺序是：标志寄存器—程序计数器。接受中断请求前后的堆栈状态如图 11.7 所示。

在高速中断的中断响应顺序中，分别将标志寄存器（FLG）和程序计数器（PC）保存到标志保存寄存器（SVF）和 PC 保存寄存器（SVP）。

必须在中断处理程序的开始位置通过软件将其他需要的寄存器压栈。如果使用 PUSHM 指令，就能用 1 条指令将帧基址寄存器（FB）和堆栈指针（SP）以外的全部寄存器压栈。

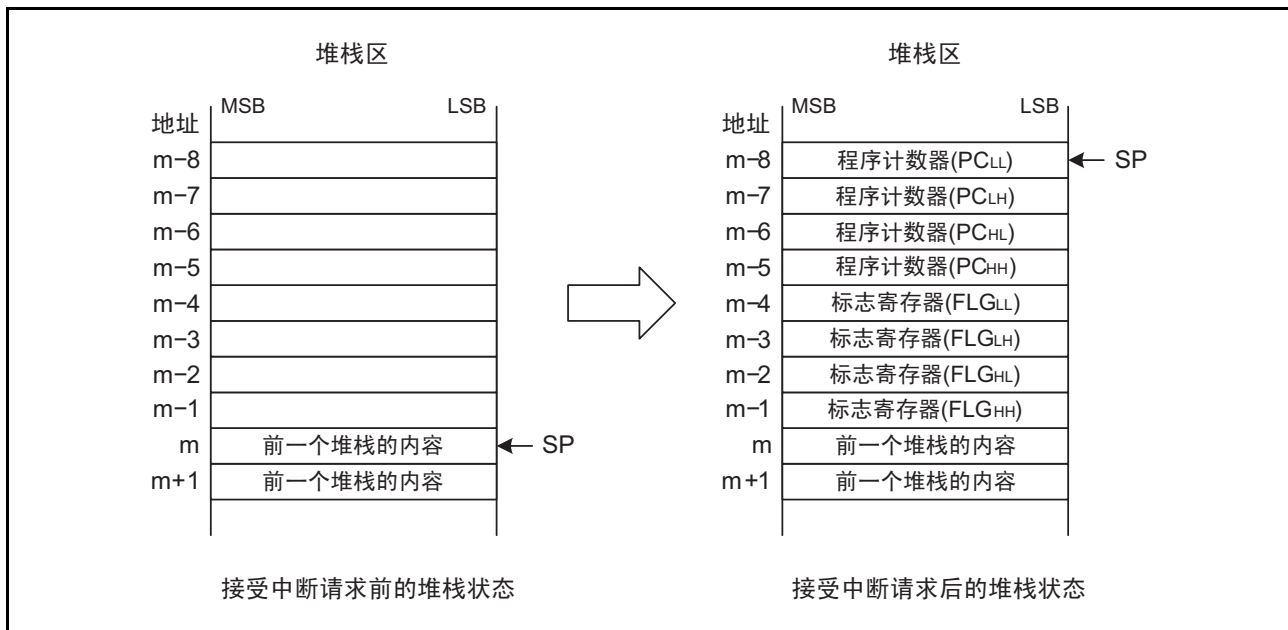


图 11.7 接受中断请求前后的堆栈状态

## 11.7 从中断处理程序的返回

如果在中断处理程序的最后执行 **REIT** 指令，就恢复被压栈的中断响应顺序前的标志寄存器（**FLG**）和程序计数器（**PC**）的内容。在高速中断的情况下，如果在中断处理程序的最后执行 **FREIT** 指令，就恢复被保存到 **SVF** 和 **SVP**（保存寄存器）的中断响应顺序前的标志寄存器（**FLG**）和程序计数器（**PC**）的内容。此后，返回到接受中断请求前正在执行的程序，继续执行被中断的处理。

必须在执行 **REIT** 指令和 **FREIT** 指令前，使用 **POPM** 等指令恢复在中断处理程序内通过软件压栈的寄存器。

在中断处理程序内转换寄存器组时，如果执行 **REIT** 指令或者 **FREIT** 指令，就自动恢复为中断响应顺序前的寄存器组。

## 11.8 中断优先级

如果在同一采样点（调查是否有中断请求时）发生 2 个或者 2 个以上的中断请求，就接受优先级高的中断。

能通过中断请求级选择位（**ILVL2** ~ **ILVL0** 位）任意设定可屏蔽中断（外围功能中断）的优先级。如果设定相同的中断请求级，就接受硬件设定的优先级高的中断。

复位（复位为优先级最高的中断）和看门狗定时器中断等特殊中断的优先级是由硬件设定的，中断优先级如下所示：

看门狗定时器  
 复位 > 振荡停止检测 > NMI > 外围功能  
 低电压检测

软件中断不受中断优先级的影响。如果执行指令，就一定转移到中断处理程序。



### 11.9 中断优先级的判断电路

中断优先级的判断电路用于从同一采样点发生请求的中断中选择优先级最高的中断。  
 中断优先级的判断电路如图 11.8 所示。

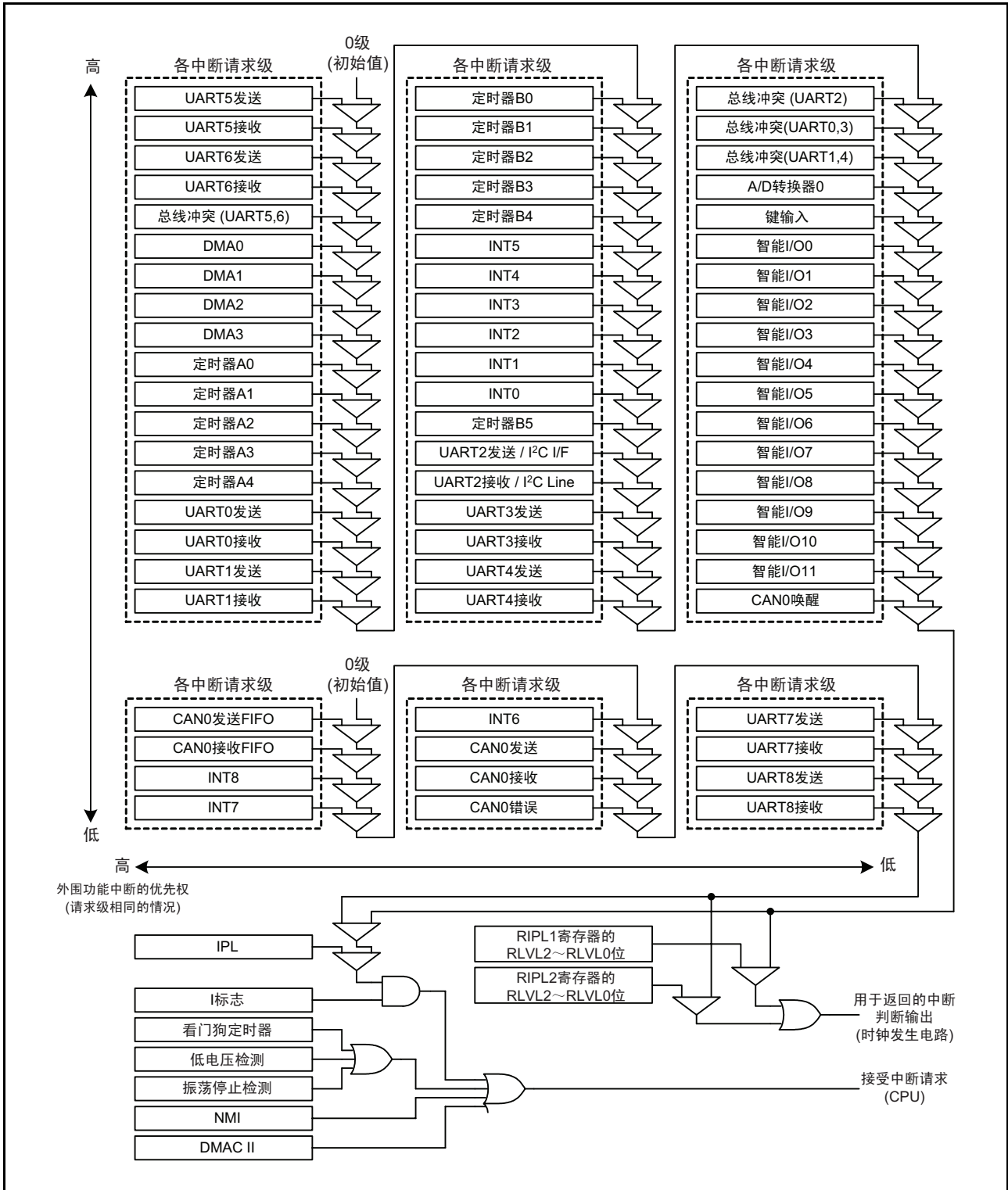


图 11.8 中断优先级的判断电路

### 11.10 外部中断

外部中断是由  $\overline{\text{INT}}_i$  引脚 ( $i=0 \sim 8$ ) 的外部输入产生的中断。能通过  $\text{INT}i\text{C}$  寄存器的 LVS 位选择电平检测 (通过输入信号的电平触发中断) 或者边沿检测 (通过边沿触发中断), 能通过  $\text{INT}i\text{C}$  寄存器的 POL 位选择输入信号的极性。

在使用边沿检测时, 如果在  $\overline{\text{INT}}_j$  ( $j=0 \sim 5$ ) 时将 IFSR0 寄存器的 IFSR0j 位、在  $\overline{\text{INT}}_m$  ( $m=6 \sim 8$ ) 时将 IFSR1 寄存器的 IFSR1n 位 ( $n=m-6$ ) 置 “1” (双边沿), 就在外部中断输入的上升沿和下降沿发生中断请求。在将 IFSR0j 位、IFSR1n 位置 “1” 时, 必须将对应的 POL 位置 “0” (下降沿)。

在使用电平检测时, 必须将 IFSR0j 位、IFSR1n 位置 “0” (单边沿)。当  $\overline{\text{INT}}_i$  引脚的输入电平为 POL 位选择的电平时,  $\text{INT}i\text{C}$  寄存器的 IR 位变为 “1”。此后, 即使  $\overline{\text{INT}}_i$  引脚发生变化, IR 位也保持 “1”。如果接受  $\text{INT}i$  中断或者给 IR 位写 “0”, IR 位就变为 “0”。

IFSR0 寄存器如图 11.9, IFSR1 寄存器如图 11.10 所示。

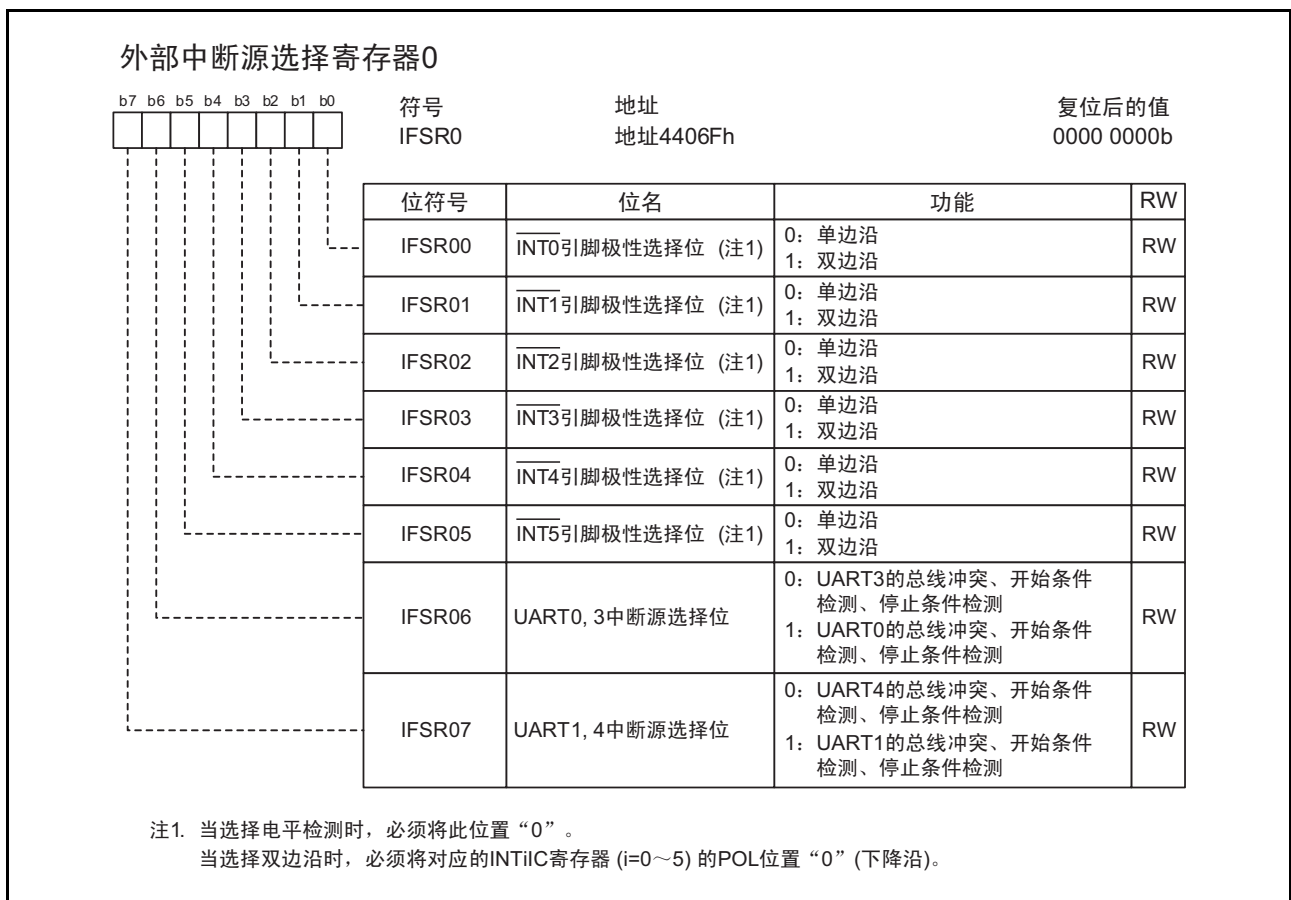


图 11.9 IFSR0 寄存器

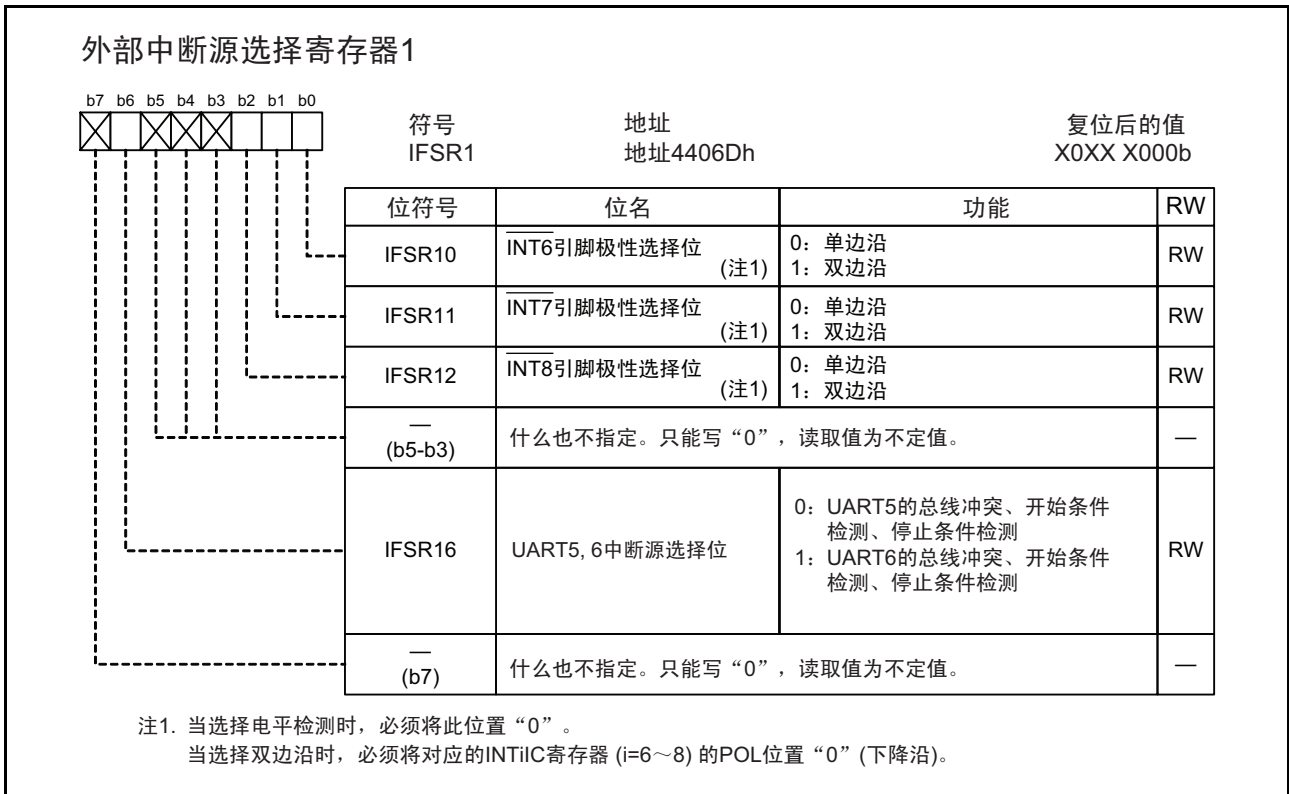


图 11.10 IFSR1 寄存器

### 11.11 NMI

如果  $\overline{\text{NMI}}$  引脚的输入电平从“H”变为“L”，就发生 NMI (Non Maskable Interrupt)。NMI 是非屏蔽中断，复位后为禁止状态。必须在程序的开始位置设定中断堆栈指针 (ISP) 后，通过将 PM2 寄存器的 PM24 位置“1”来允许 NMI。因为  $\overline{\text{NMI}}$  引脚和端口 P8\_5 引脚复用，所以能通过 P8 寄存器的 P8\_5 位读引脚的输入电平。

注 1. 在不使用 NMI 功能时，PM2 寄存器的 PM24 位必须保持“0”。

## 11.12 键输入中断

如果将 P10\_4 ~ P10\_7 全部设定为输入端口，就能使用键输入中断功能。

如果 P10\_4 ~ P10\_7 中的任意引脚从“H”电平变为“L”电平，就产生键输入中断请求。键输入中断也能用作解除等待模式或者停止模式的键唤醒功能，键输入中断的框图如图 11.11 所示。如果给任意的引脚输入“L”电平，就无法检测到其他引脚的输入中断。

在使用键输入中断功能时，必须将 P10\_4S ~ P10\_7S 寄存器全部置“00h”（端口），还必须将 PD10\_4 ~ PD10\_7 位全部置“0”（输入）。在其他设定中，不能使用键输入中断功能。

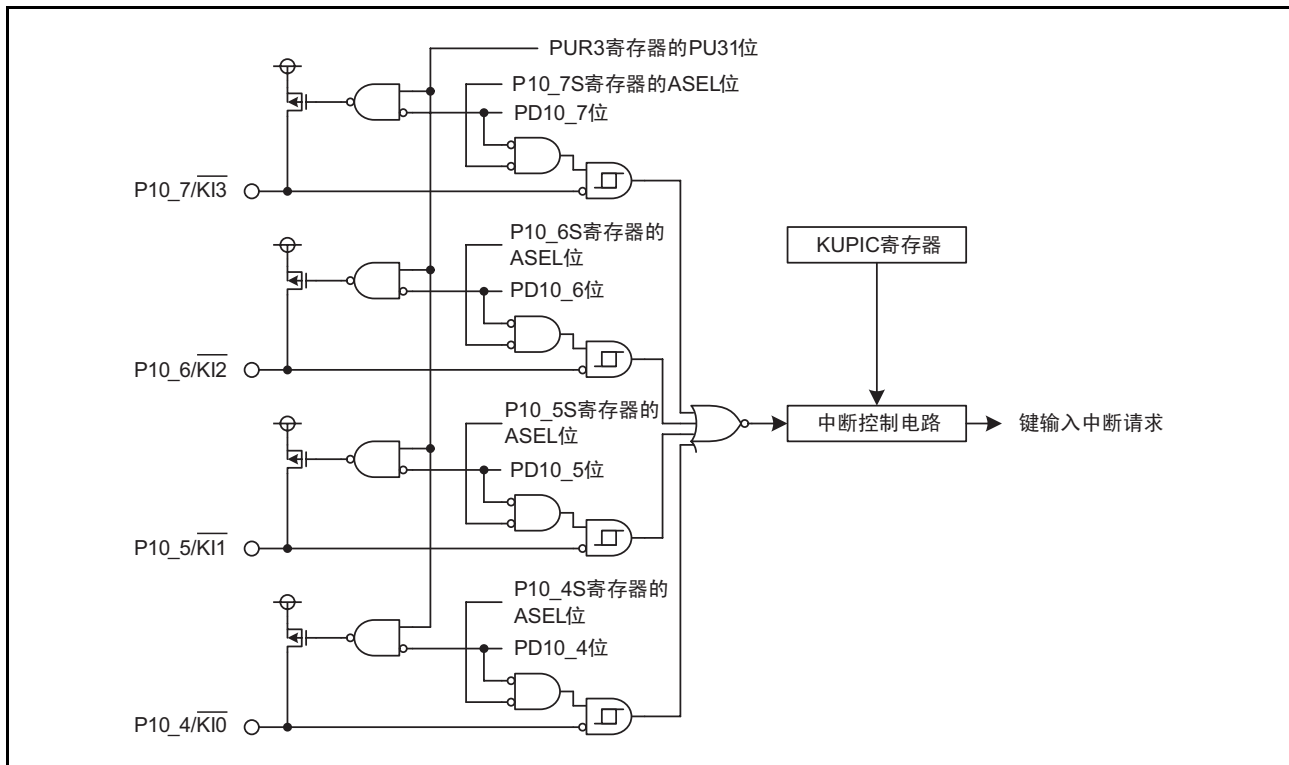


图 11.11 键输入中断的框图

### 11.13 智能 I/O 中断

智能 I/O 中断分配在软件中断序号 44 ~ 55。

智能 I/O 中断的框图如图 11.12 所示，IIOiIR 寄存器 (i=0 ~ 11) 和 IIOiIE 寄存器分别如图 11.13 和图 11.14 所示。

在使用智能 I/O 中断时，必须将 IIOiIE 寄存器的 IRLT 位置“1”（中断请求用于中断）。

智能 I/O 中断有多个中断源。如果在使用智能 I/O 的各功能时发生中断请求，IIOiIR 寄存器的对应位就变为“1”（有中断请求）。此时，如果 IIOiIE 寄存器的对应位为“1”（允许中断），对应的 IIOiC 寄存器的 IR 位就变为“1”（有中断请求）。

如果在 IR 位从“0”变为“1”后，IIOiIR 寄存器的位因其他中断源而变为“1”，并且如果 IIOiIE 寄存器的对应位为“1”，IR 位就保持“1”不变。

即使接受中断，IIOiIR 寄存器的各位也不会自动变为“0”，因此必须通过程序用 AND 指令或者 BCLR 指令将这些位置“0”。如果这些位保持“1”，以后发生的中断请求就全部无效。

在将智能 I/O 中断用作 DMA II 的启动源时，必须将 IIOiIE 寄存器的 IRLT 位置“0”（中断请求用于 DMA 和 DMA II），并且通过 IIOiIE 寄存器允许所使用的中断请求。

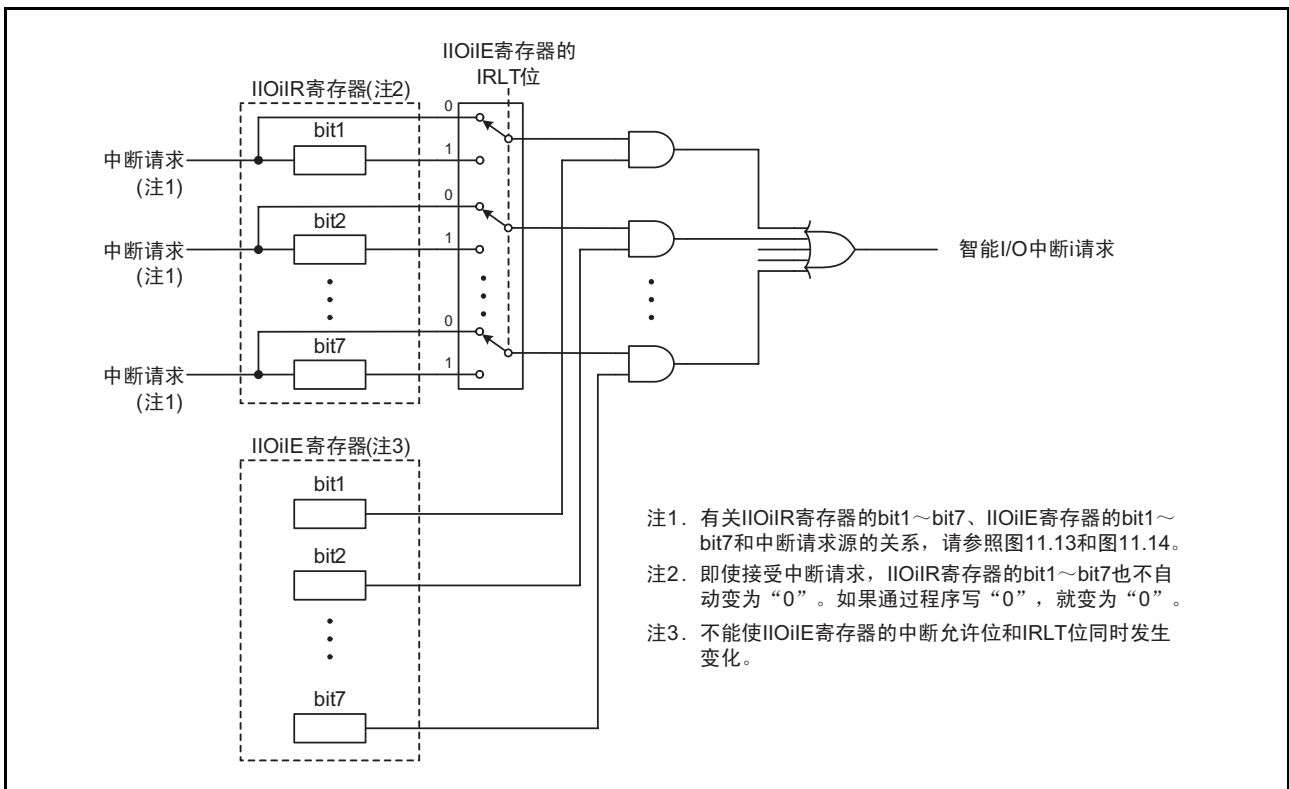


图 11.12 智能 I/O 中断的框图 (i=0 ~ 11)



图 11.13 IIO01R ~ IIO111R 寄存器

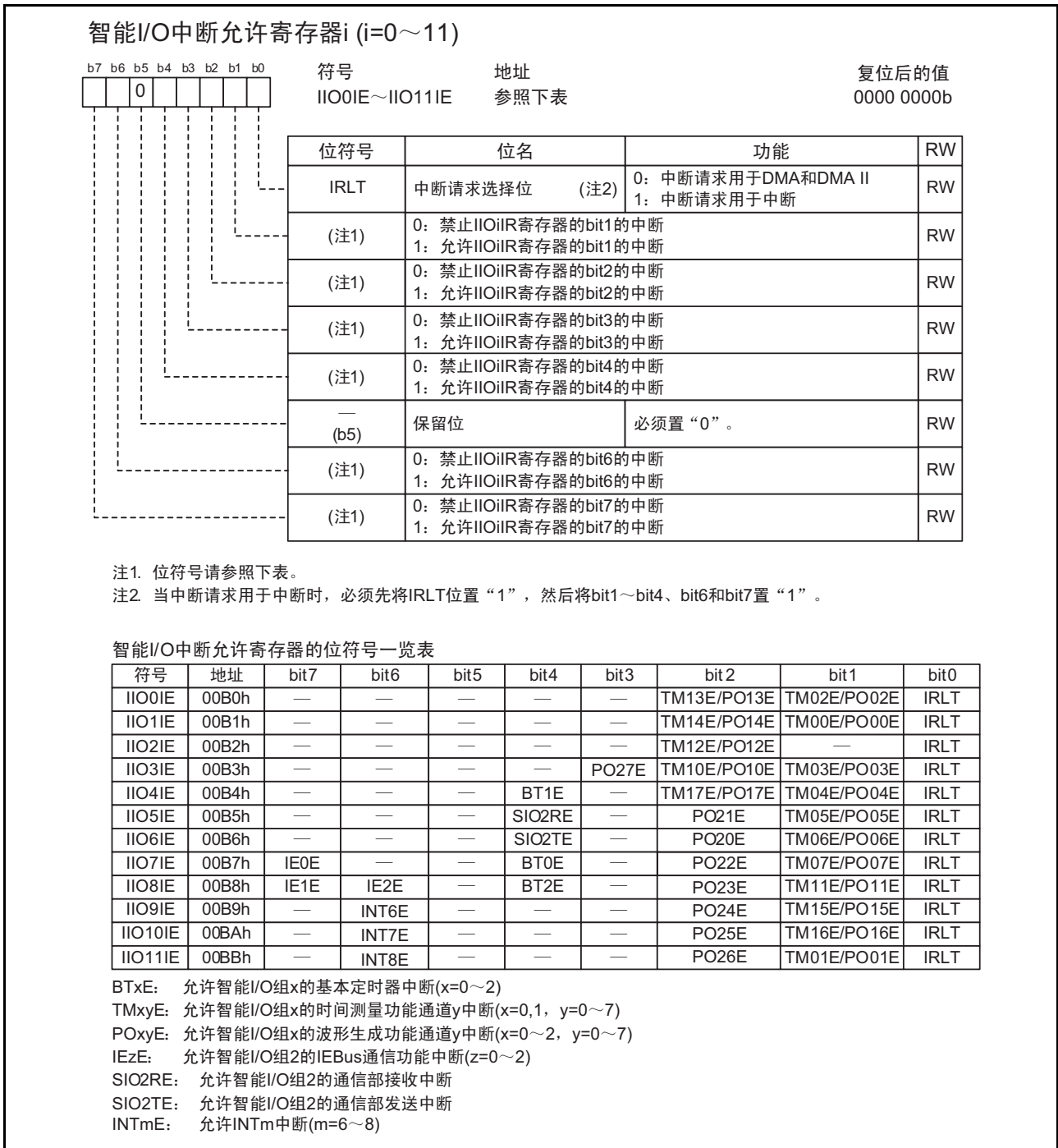


图 11.14 IIO01E ~ IIO111E 寄存器

## 11.14 使用中断时的注意事项

### 11.14.1 ISP 的设定

在复位后，ISP（中断堆栈指针）被初始化为“00000000h”。如果在给ISP设定值前接受中断，就会导致程序失控，因此必须在接受中断前给ISP设定值。ISP的设定值必须是4的倍数，设定4的倍数能减少存储器的存取次数从而加快中断响应顺序的执行速度。

尤其在使用NMI时，不能禁止中断，因此必须在程序的开始位置设定ISP后，将PM2寄存器的PM24位置“1”（NMI有效）。

### 11.14.2 NMI

- 在将PM2寄存器的PM24位置“1”（NMI有效）后，不能禁止NMI。在不使用NMI时，PM24位必须保持“0”。
- 当PM2寄存器的PM24位为“1”（NMI有效）时，P8寄存器的P8\_5位只能用于确认NMI引脚的状态，而不能用作通用端口。

### 11.14.3 外部中断

- $\overline{\text{INT}}_i$  引脚（ $i=0\sim 8$ ）的输入信号需要满足电特性规定的信号宽度。如果小于规定的最小宽度，就有可能无法接受中断。
- 在通过 $\text{INT}_i\text{IC}$ 寄存器（ $i=0\sim 8$ ）的POL位和LVS位、IFSR0寄存器的IFSR0 $i$ 位（ $i=0\sim 5$ ）、IFSR1寄存器的IFSR1 $j$ 位（ $j=i-6$ ， $i=6\sim 8$ ）转换 $\overline{\text{INT}}_i$ 引脚的有效边沿或者有效电平时，对应的IR位有可能变为“1”（有中断请求）。因此，必须在将 $\text{INT}_i\text{IC}$ 寄存器的ILVL2 $\sim$ ILVL0位置“000b”（禁止中断）后改写这些位。在改写后，必须将对应的IR位置“0”（无中断请求），然后设定ILVL2 $\sim$ ILVL0位。
- 在通过 $\text{INT}_i\text{IC}$ 寄存器（ $i=6\sim 8$ ）的POL位和LVS位、IFSR1寄存器的IFSR1 $j$ 位（ $j=i-6$ ）转换 $\overline{\text{INT}}_i$ 引脚的有效边沿或者有效电平时，IIOKIR寄存器（ $k=9\sim 11$ ）的INTIR位有可能变为“1”（有中断请求）。因此，必须将IIOKIE寄存器的INTIE位置“0”（禁止中断）后改写这些位。在改写后，必须将对应的INTIR位置“0”（无中断请求），然后将INTIE位置“1”。



## 12. 看门狗定时器

看门狗定时器用于检测程序的失控。看门狗定时器有 15 位计数器，通过预分频器分频的时钟对外围总线时钟进行递减计数。

在看门狗定时器发生下溢时，能通过 CM0 寄存器的 CM06 位选择是产生中断请求还是复位。一旦将 CM06 位置“1”（复位），就不能通过程序而只能通过复位置“0”（看门狗定时器中断）。

看门狗定时器有预分频器，对外围总线时钟进行 16 分频或者 128 分频，能通过 WDC 寄存器的 WDC7 位选择分频比。

在 MCU 为等待模式或者停止模式并且  $\overline{\text{HOLD}}$  信号为“L”电平时，看门狗定时器停止计数，如果解除这些状态，就从保持的值开始重新计数。

看门狗定时器的周期能用下式进行计算。但是，根据 WDTS 寄存器的写时序，有可能产生最大 1 个周期的预分频器输出的误差。

$$\text{看门狗定时器的周期} = \frac{\text{预分频器的分频值}(16\text{或者}128) \times 32768}{\text{外围总线时钟周期}}$$

例如，CPU 时钟的频率为 50MHz，外围总线时钟的频率为 CPU 时钟的 1/2，预分频器的分频值为 16 时，则看门狗定时器的周期约为 21ms。

在写 WDTS 寄存器或者发生看门狗定时器中断的中断请求后，对看门狗定时器进行初始化。只能在复位后对预分频器的计数值进行初始化。

在复位后，看门狗定时器和预分频器停止计数，通过给 WDTS 寄存器写值开始计数。

看门狗定时器的框图以及看门狗定时器的相关寄存器分别如图 12.1 和图 12.2 ~ 图 12.3 所示。

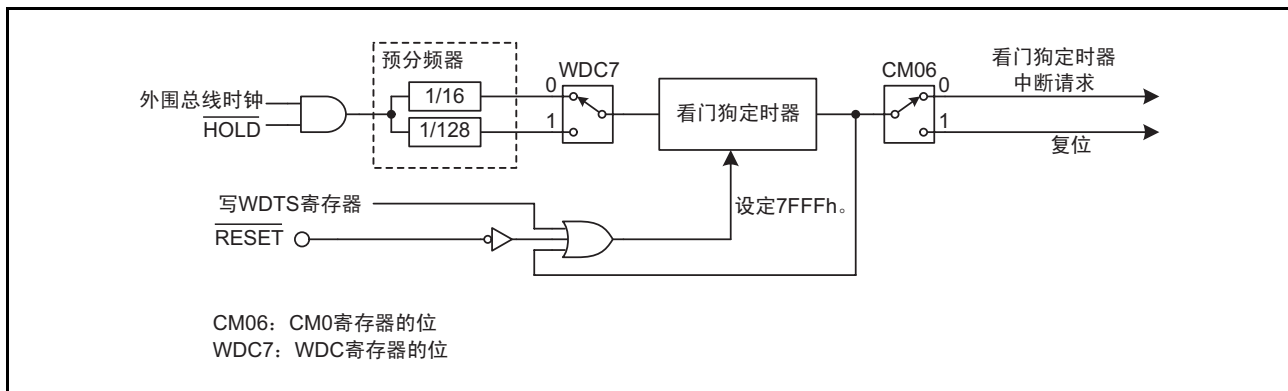


图 12.1 看门狗定时器的框图

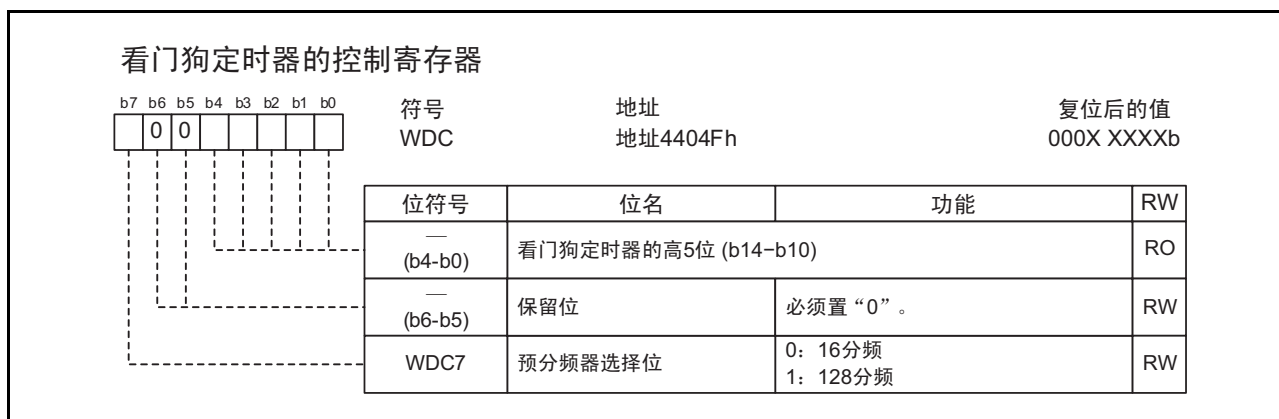


图 12.2 WDC 寄存器

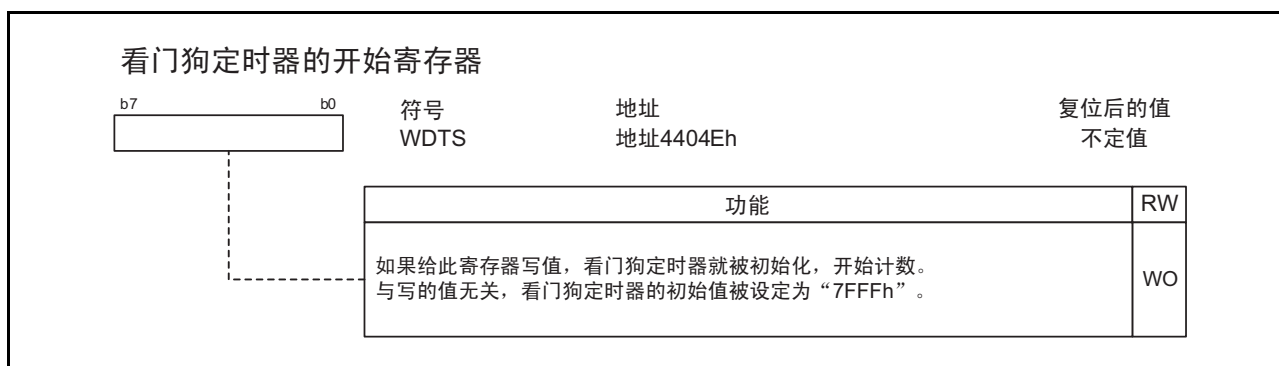


图 12.3 WDTS 寄存器

## 13. DMAC

DMA（直接存储器存取）是指不使用 CPU 的指令进行数据传送的功能，进行 DMA 的控制器称为 DMAC（DMA 控制器）。

R32C/100 系列装载了 4 个通道的周期挪用方式 DMAC。

每当发生传送请求时，DMAC 就将 8 位、16 位或者 32 位的数据从传送源地址传送 1 次到传送目标地址。虽然 CPU 和 DMAC 使用相同的总线，但是 DMAC 的总线存取优先级高于 CPU，因此能快速地进行从发生传送请求到 1 次数据传送结束为止的操作。

CPU 内部的 DMAC 相关寄存器一览表以及 DMAC 的相关寄存器分别如图 13.1 和图 13.2 ~ 图 13.10 所示，DMAC 的规格如表 13.1 所示。因为图 13.1 所示的寄存器分配在 CPU 内部，所以必须使用 LDC 指令或者 STC 指令进行存取。

DMAC相关寄存器	
DMD0	DMA0模式寄存器
DMD1	DMA1模式寄存器
DMD2	DMA2模式寄存器
DMD3	DMA3模式寄存器
DCT0	DMA0终端计数寄存器
DCT1	DMA1终端计数寄存器
DCT2	DMA2终端计数寄存器
DCT3	DMA3终端计数寄存器
DCR0	DMA0终端计数重加载寄存器(注1)
DCR1	DMA1终端计数重加载寄存器(注1)
DCR2	DMA2终端计数重加载寄存器(注1)
DCR3	DMA3终端计数重加载寄存器(注1)
DSA0	DMA0源地址寄存器
DSA1	DMA1源地址寄存器
DSA2	DMA2源地址寄存器
DSA3	DMA3源地址寄存器
DSR0	DMA0源地址重加载寄存器(注1)
DSR1	DMA1源地址重加载寄存器(注1)
DSR2	DMA2源地址重加载寄存器(注1)
DSR3	DMA3源地址重加载寄存器(注1)
DDA0	DMA0目标地址寄存器
DDA1	DMA1目标地址寄存器
DDA2	DMA2目标地址寄存器
DDA3	DMA3目标地址寄存器
DDR0	DMA0目标地址重加载寄存器(注1)
DDR1	DMA1目标地址重加载寄存器(注1)
DDR2	DMA2目标地址重加载寄存器(注1)
DDR3	DMA3目标地址重加载寄存器(注1)

注1. 这是在重复传送时使用而在单次传送时不使用的寄存器。

图 13.1 CPU 内部的 DMAC 相关寄存器一览表

表 13.1 DMAC 的规格

项目		规格
通道数		4 个通道
方式		周期挪用方式
传送空间		将数据从 64M 字节 (00000000h ~ 01FFFFFFh 和 FE000000h ~ FFFFFFFFh) 的任意空间传送到 64M 字节的任意空间。
最大传送字节数		64M 字节 (32 位传送)、32M 字节 (16 位传送)、16M 字节 (8 位传送)
DMA 启动源 (注 1)		输入到 INT0 ~ INT3、INT6 ~ INT8 引脚的下降沿或者双边沿 定时器 A0 ~ 定时器 A4 的中断请求 定时器 B0 ~ 定时器 B5 的中断请求 UART0 ~ UART8 的发送 / 接收中断请求 A/D 转换中断请求 智能 I/O 的中断请求 多主控 I <sup>2</sup> C 总线中断请求 软件触发
通道优先级		DMA0 > DMA1 > DMA2 > DMA3 (DMA0 的优先级最高)
传送大小		8 位、16 位、32 位
寻址		递增或者固定
传送模式	单次传送	如果 DCTi 寄存器 (i=0 ~ 3) 变为 “0000000h”, 就结束传送。
	重复传送	如果 DCTi 寄存器变为 “0000000h”, 就将 DCRi 寄存器的值重新加载到 DCTi 寄存器后继续进行 DMA 传送。
DMA 传送结束中断请求的发生时序		在 DCTi 寄存器从 “0000001h” 变为 “0000000h” 时。
DMA 传送开始	单次传送	在给 DCTi 寄存器设定大于等于 “0000001h” 的值并且将 DMDi 寄存器的 MDi1 ~ MDi0 位置 “01b” (单次传送) 后, 如果发生 DMA 传送请求, 就开始单次传送。
	重复传送	在给 DCTi 寄存器设定大于等于 “0000001h” 的值并且将 MDi1 ~ MDi0 位置 “11b” (重复传送) 后, 如果发生 DMA 传送请求, 就开始重复传送。
DMA 传送停止	单次传送	将 MDi1 ~ MDi0 位置 “00b” (禁止 DMA 传送)。
	重复传送	将 MDi1 ~ MDi0 位置 “00b” (禁止 DMA 传送)。
DCTi 寄存器、DSAi 寄存器、DDAi 寄存器的重新加载时序		在重复传送模式中并且 DCTi 寄存器从 “0000001h” 变为 “0000000h” 时。
DMA 传送周期数		最小 3 个周期

注 1. DMA 传送不影响各中断。

DMA 传送请求除了写 DMiSL2 寄存器 (i=0 ~ 3) 的 DSR 位所产生的软件触发以外, 还使用 DMiSL 寄存器的 DSEL4 ~ DSEL0 位和 DMiSL2 寄存器的 DSEL24 ~ DSEL20 位指定的各功能输出的中断请求。但是, DMA 传送请求与中断请求不同, 不受 I 标志和中断控制寄存器的影响, 所以在禁止中断时, 即使不能接受中断请求也能接受 DMA 传送请求。因为 DMA 传送不影响中断, 所以中断控制寄存器的 IR 位不会因 DMA 传送而发生变化。

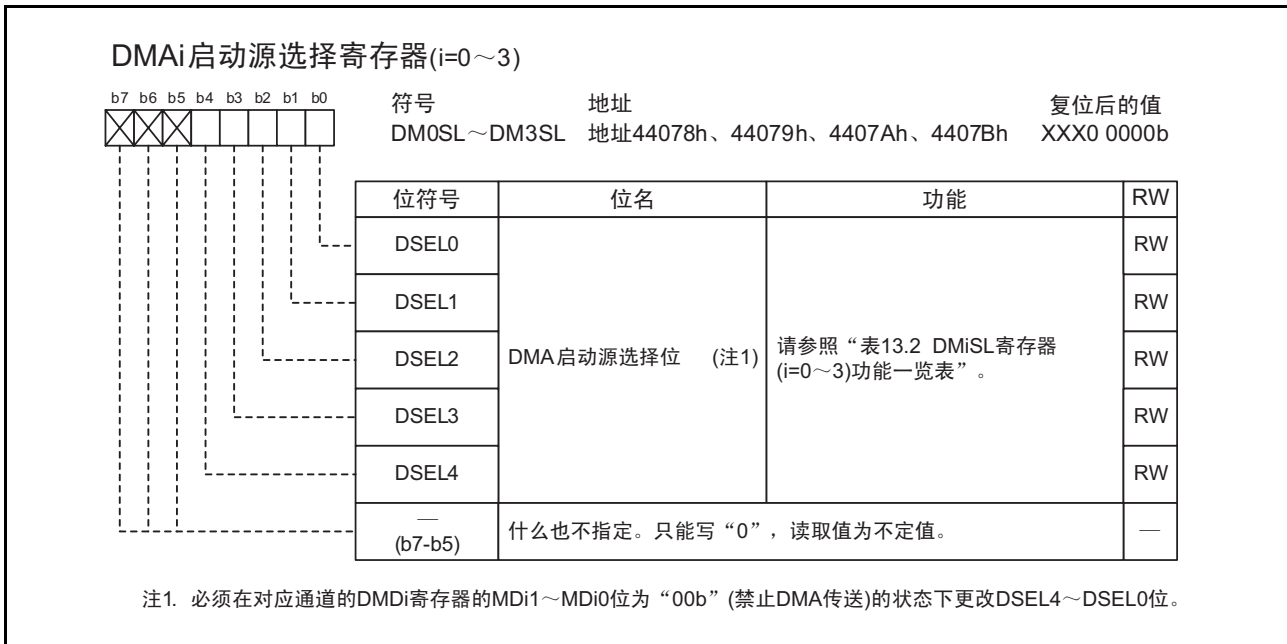


图 13.2 DM0SL ~ DM3SL 寄存器

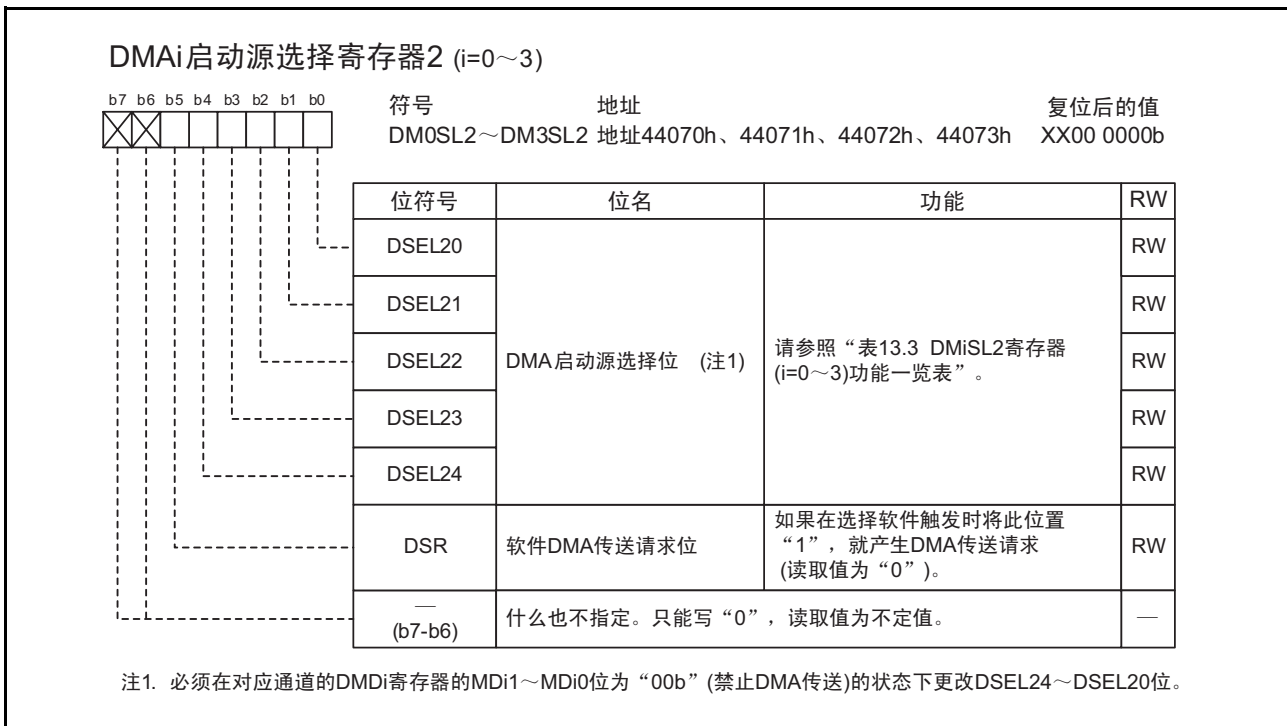


图 13.3 DM0SL2 ~ DM3SL2 寄存器

表 13.2 DMiSL 寄存器 (i=0 ~ 3) 功能一览表

设定值					DMA 启动源			
b4	b3	b2	b1	b0	DMA0	DMA1	DMA2	DMA3
0	0	0	0	0	通过启动源选择寄存器 2 来选择。			
0	0	0	0	1	$\overline{\text{INT0}}$ 的下降沿	$\overline{\text{INT1}}$ 的下降沿	$\overline{\text{INT2}}$ 的下降沿	$\overline{\text{INT3}}$ 的下降沿 (注 1) (注 2)
0	0	0	1	0	$\overline{\text{INT0}}$ 的双边沿	$\overline{\text{INT1}}$ 的双边沿	$\overline{\text{INT2}}$ 的双边沿	$\overline{\text{INT3}}$ 的双边沿 (注 1) (注 2)
0	0	0	1	1	定时器 A0 的中断请求			
0	0	1	0	0	定时器 A1 的中断请求			
0	0	1	0	1	定时器 A2 的中断请求			
0	0	1	1	0	定时器 A3 的中断请求			
0	0	1	1	1	定时器 A4 的中断请求			
0	1	0	0	0	定时器 B0 的中断请求			
0	1	0	0	1	定时器 B1 的中断请求			
0	1	0	1	0	定时器 B2 的中断请求			
0	1	0	1	1	定时器 B3 的中断请求			
0	1	1	0	0	定时器 B4 的中断请求			
0	1	1	0	1	定时器 B5 的中断请求			
0	1	1	1	0	UART0 的发送中断请求			
0	1	1	1	1	UART0 的接收或者 ACK 中断请求 (注 3)			
1	0	0	0	0	UART1 的发送中断请求			
1	0	0	0	1	UART1 的接收或者 ACK 中断请求 (注 3)			
1	0	0	1	0	UART2 的发送中断请求或者 I <sup>2</sup> C 总线接口中断请求 (注 4)			
1	0	0	1	1	UART2 的接收、ACK 中断请求 (注 3) 或者 I <sup>2</sup> C 总线的线中断请求 (注 4)			
1	0	1	0	0	UART3 的发送中断请求	UART5 的发送中断请求		
1	0	1	0	1	UART3 的接收或者 ACK 中断请求 (注 3)	UART5 的接收或者 ACK 中断请求 (注 3)		
1	0	1	1	0	UART4 的发送中断请求	UART6 的发送中断请求		
1	0	1	1	1	UART4 的接收或者 ACK 中断请求 (注 3)	UART6 的接收或者 ACK 中断请求 (注 3)		
1	1	0	0	0	A/D0 的中断请求			
1	1	0	0	1	智能 I/O 的中断 0 请求	智能 I/O 的中断 7 请求	智能 I/O 的中断 2 请求	智能 I/O 的中断 9 请求
1	1	0	1	0	智能 I/O 的中断 1 请求	智能 I/O 的中断 8 请求	智能 I/O 的中断 3 请求	智能 I/O 的中断 10 请求
1	1	0	1	1	智能 I/O 的中断 2 请求	智能 I/O 的中断 9 请求	智能 I/O 的中断 4 请求	智能 I/O 的中断 11 请求
1	1	1	0	0	智能 I/O 的中断 3 请求	智能 I/O 的中断 10 请求	智能 I/O 的中断 5 请求	智能 I/O 的中断 0 请求
1	1	1	0	1	智能 I/O 的中断 4 请求	智能 I/O 的中断 11 请求	智能 I/O 的中断 6 请求	智能 I/O 的中断 1 请求
1	1	1	1	0	智能 I/O 的中断 5 请求	智能 I/O 的中断 0 请求	智能 I/O 的中断 7 请求	智能 I/O 的中断 2 请求
1	1	1	1	1	智能 I/O 的中断 6 请求	智能 I/O 的中断 1 请求	智能 I/O 的中断 8 请求	智能 I/O 的中断 3 请求

注 1. 在存储器扩展模式或者微处理器模式中，当  $\overline{\text{INT3}}$  引脚为数据总线时， $\overline{\text{INT3}}$  引脚的输入不能用作 DMA3 的启动源。

注 2. 输入到  $\overline{\text{INTi}}$  引脚 (i=0 ~ 3) 的下降沿和双边沿为 DMA 的启动源。既不受外部中断 (INTIIC 寄存器的 POL 位和 LVS 位、IFSR0 寄存器) 的影响，也不影响外部中断。

注 3. 通过 UiSMR 寄存器和 UiSMR2 寄存器进行 UARTi 接收 (i=0 ~ 6) 和 ACK 的转换。

注 4. 通过 I2CMR 寄存器的 I2CEN 位选择使用 UART2 或者 I<sup>2</sup>C 总线接口。

表 13.3 DMiSL2 寄存器 (i=0 ~ 3) 功能一览表

设定值					DMA 启动源				
b4	b3	b2	b1	b0	DMA0	DMA1	DMA2	DMA3	
0	0	0	0	0	软件触发				
0	0	0	0	1	INT6 下降沿	$\overline{\text{INT7}}$ 下降沿	$\overline{\text{INT8}}$ 下降沿	保留	(注 1)
0	0	0	1	0	INT6 双边沿	$\overline{\text{INT7}}$ 双边沿	$\overline{\text{INT8}}$ 双边沿	保留	(注 1)
0	0	0	1	1	保留				
0	0	1	0	0	保留				
0	0	1	0	1	保留				
0	0	1	1	0	保留				
0	0	1	1	1	保留				
0	1	0	0	0	保留				
0	1	0	0	1	保留				
0	1	0	1	0	保留				
0	1	0	1	1	保留				
0	1	1	0	0	保留				
0	1	1	0	1	保留				
0	1	1	1	0	保留				
0	1	1	1	1	保留				
1	0	0	0	0	保留				
1	0	0	0	1	保留				
1	0	0	1	0	保留				
1	0	0	1	1	保留				
1	0	1	0	0	保留				
1	0	1	0	1	保留				
1	0	1	1	0	保留				
1	0	1	1	1	保留				
1	1	0	0	0	UART7 的发送中断请求				
1	1	0	0	1	UART7 的接收中断请求				
1	1	0	1	0	UART8 的发送中断请求				
1	1	0	1	1	UART8 的接收中断请求				
1	1	1	0	0	保留				
1	1	1	0	1	保留				
1	1	1	1	0	保留				
1	1	1	1	1	保留				

注 1. 输入到 INTi 引脚 (i=6 ~ 8) 的下降沿和双边沿为 DMA 的启动源, 既不受外部中断 (INTiIC 寄存器的 POL 位和 LVS 位、IFSR1 寄存器) 的影响, 也不影响外部中断。

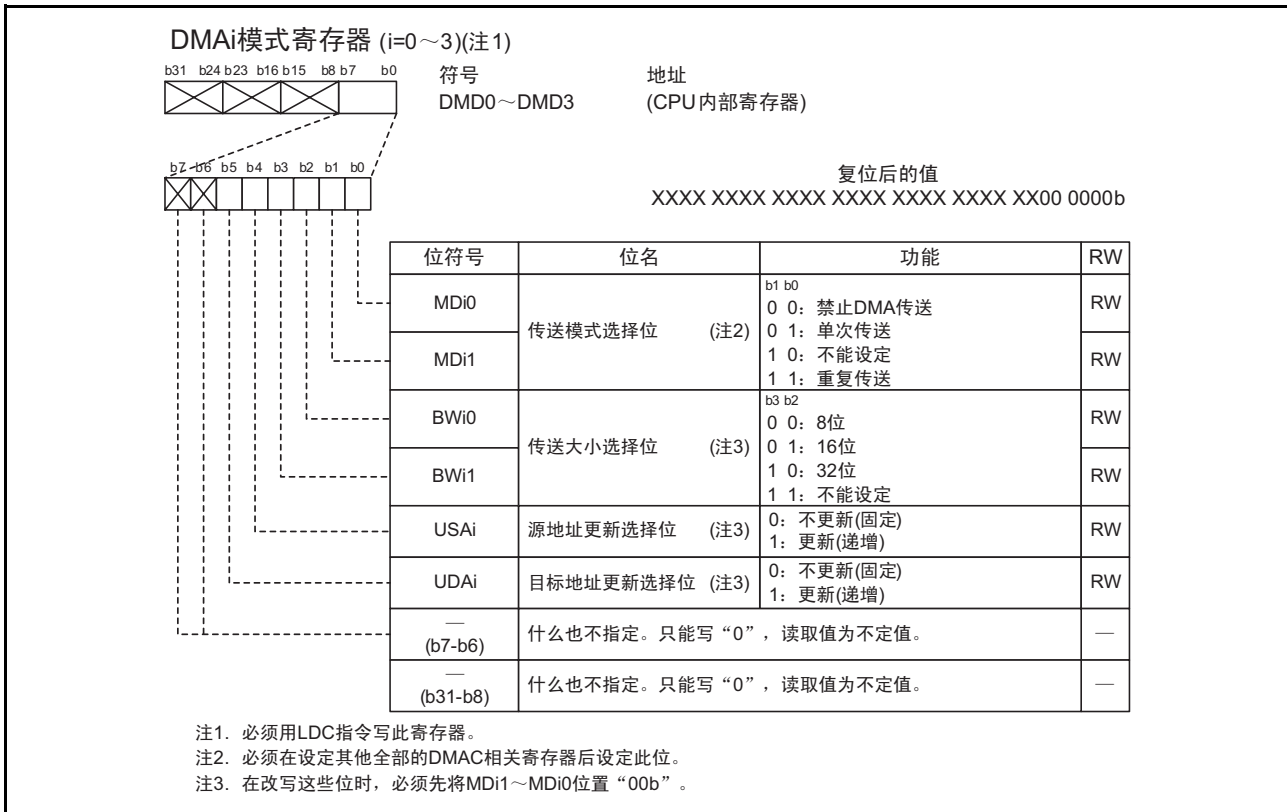


图 13.4 DMD0 ~ DMD3 寄存器

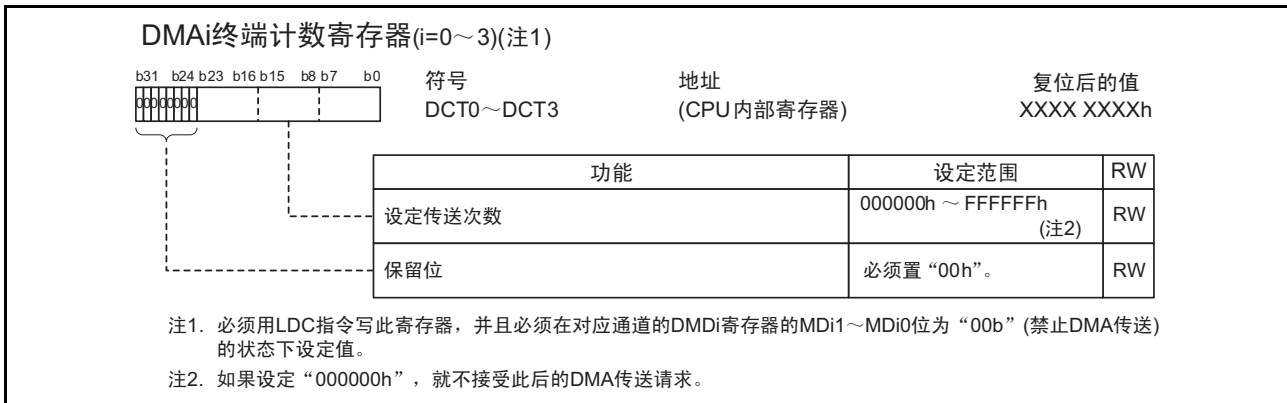


图 13.5 DCT0 ~ DCT3 寄存器

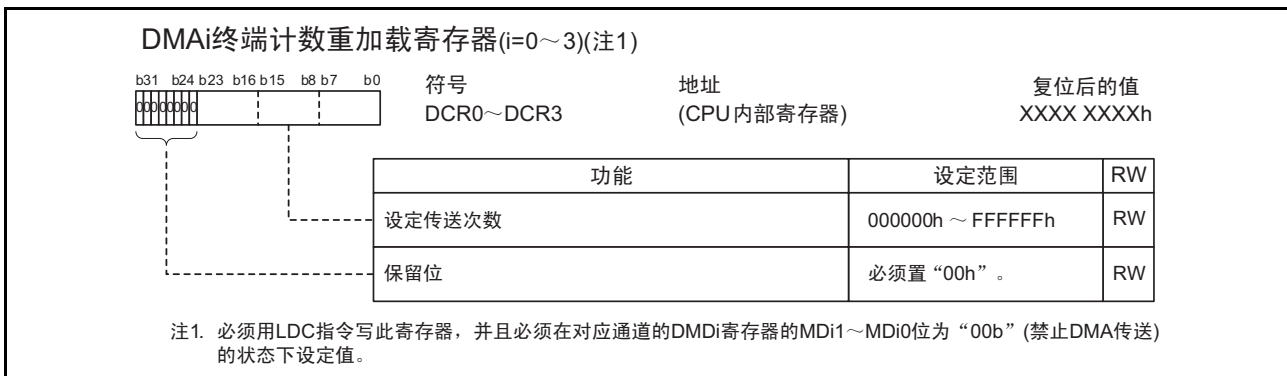


图 13.6 DCR0 ~ DCR3 寄存器



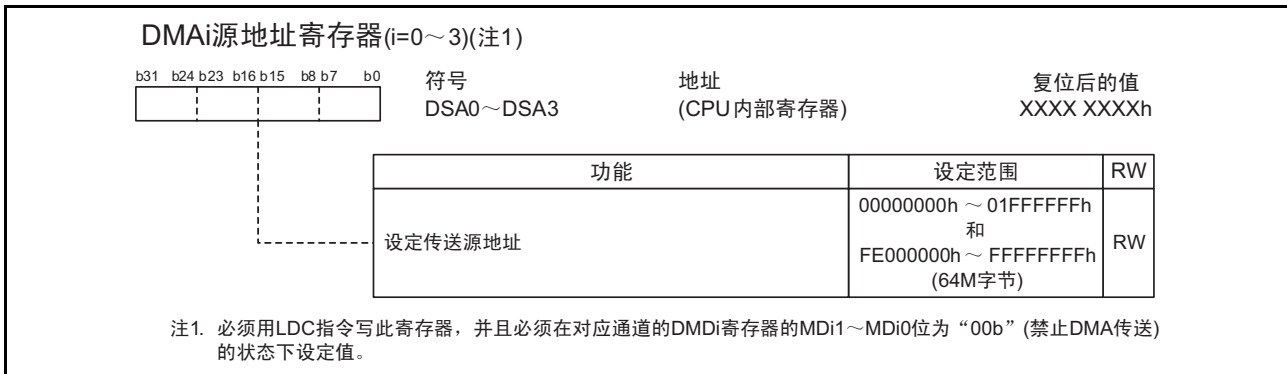


图 13.7 DSA0 ~ DSA3 寄存器

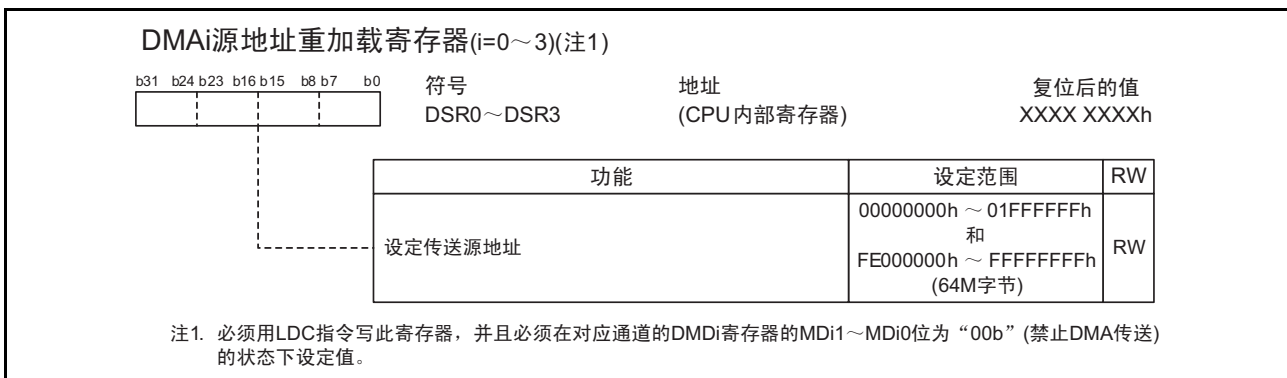


图 13.8 DSR0 ~ DSR3 寄存器

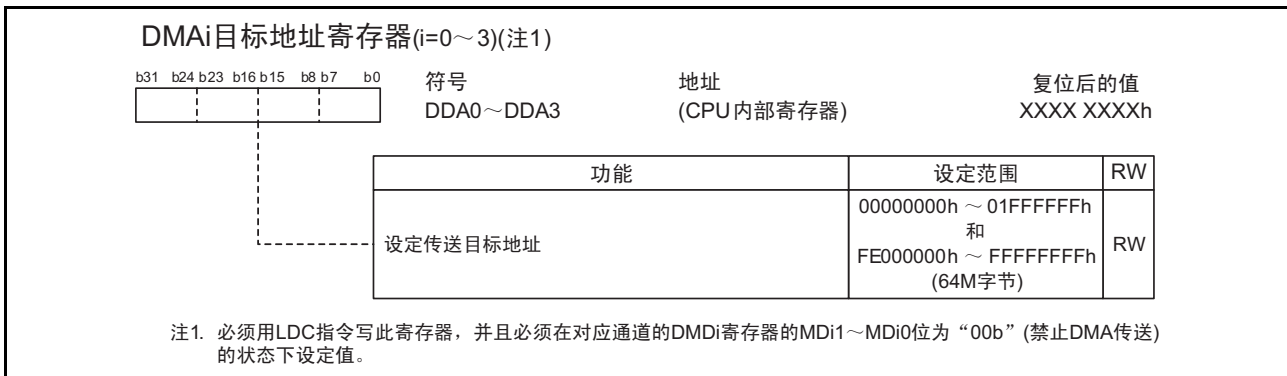


图 13.9 DDA0 ~ DDA3 寄存器

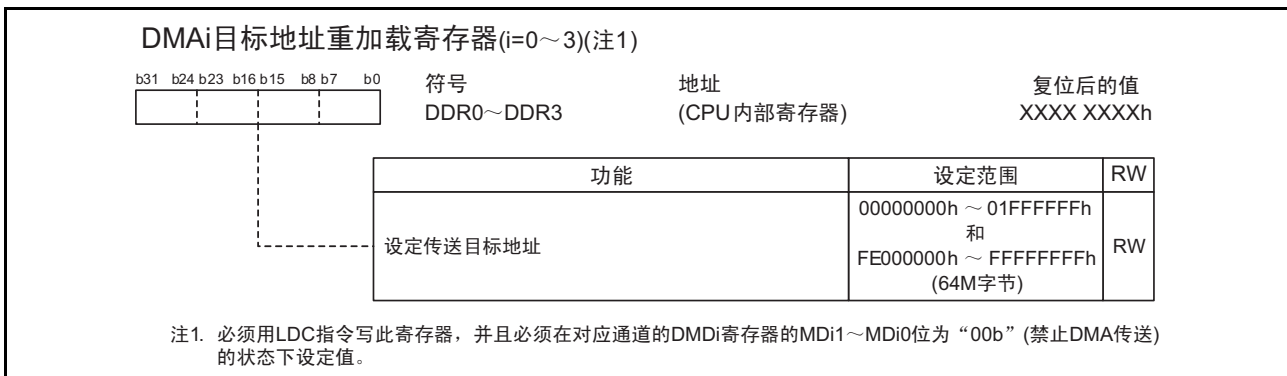


图 13.10 DDR0 ~ DDR3 寄存器

### 13.1 传送周期

传送周期由存储器或者 SFR 的读（源读）总线周期和写（目标写）总线周期构成。

读写总线周期数受 DSAi 寄存器（i=0 ~ 3）和 DDAi 寄存器的设定值、连接对象器件的数据总线宽度以及总线时序的影响。

#### 13.1.1 传送地址和数据总线宽度的影响

各传送单位对应的传送地址和数据总线宽度引起的总线周期数的增加如表 13.4 所示。

表 13.4 传送地址和数据总线宽度引起的总线周期数的增加

传送单位	数据总线宽度	传送地址	增加的总线周期数	发生的总线周期	
8 位	8 位 ~ 64 位	地址 n	0	[n]	
16 位	8 位	地址 n	+1	[n]-[n+1]	
		地址 2n	0	[2n]	
	16 位	地址 2n+1	+1	[2n+1]-[2n+2]	
		地址 4n	0	[4n]	
		地址 4n+1	0	[4n+1]	
		地址 4n+2	0	[4n+2]	
	32 位	地址 4n+3	+1	[4n+3]-[4n+4]	
		地址 8n	0	[8n]	
		地址 8n+1	0	[8n+1]	
		地址 8n+2	0	[8n+2]	
		地址 8n+3	0	[8n+3]	
		地址 8n+4	0	[8n+4]	
	64 位	地址 8n+5	0	[8n+5]	
		地址 8n+6	0	[8n+6]	
地址 8n+7		+1	[8n+7]-[8n+8]		
8 位		地址 n	+3	[n]-[n+1]-[n+2]-[n+3]	
		地址 4n	+1	[4n]-[4n+2]	
		16 位	地址 4n+1	+2	[4n+1]-[4n+2]-[4n+4]
			地址 4n+2	+1	[4n+2]-[4n+4]
地址 4n+3			+2	[4n+3]-[4n+4]-[4n+6]	
32 位	地址 4n	0	[4n]		
	地址 4n+1	+1	[4n+1]-[4n+4]		
	地址 4n+2	+1	[4n+2]-[4n+4]		
	地址 4n+3	+1	[4n+3]-[4n+4]		
	64 位	地址 8n	0	[8n]	
		地址 8n+1	0	[8n+1]	
		地址 8n+2	0	[8n+2]	
地址 8n+3		0	[8n+3]		
地址 8n+4		0	[8n+4]		
地址 8n+5	+1	[8n+5]-[8n+8]			
地址 8n+6	+1	[8n+6]-[8n+8]			
地址 8n+7	+1	[8n+7]-[8n+8]			

### 13.1.2 总线时序的影响

R32C/100 系列按各设备分配连接的总线，并且各自的总线宽度和总线时序不同。各设备的总线宽度和存取周期数如表 13.5 所示。

表 13.5 各设备的总线宽度和总线周期数

设备名称	地址 (注 1)	总线宽度	存取周期数 (注 2)	基准时钟
闪存	FFE00000h ~ FFFFFFFFh	64 位	2 或者 3 (注 3)	CPU 时钟
数据闪存	00060000h ~ 00061FFFh	64 位	5	CPU 时钟
RAM	00000400h ~ 0003FFFFh	64 位	1 或者 2 (注 4)	CPU 时钟
SFR 区域	00000000h ~ 0000001Fh	16 位	3 (注 5)	外围总线时钟
	00000020h ~ 000003FFh	16 位	2 (注 5)	外围总线时钟
SFR2 区域	00040000h ~ 00041FFFh	16 位	2 (注 5)	外围总线时钟
	00042000h ~ 00043FFFh	32 位	2 (注 5)	外围总线时钟
	00044000h ~ 000440DFh	16 位	2 (注 5、注 6)	外围总线时钟
	000440E0h ~ 000443FFh	16 位	3 (注 5、注 6)	外围总线时钟
	00044400h ~ 00045FFFh	16 位	2 (注 5、注 6)	外围总线时钟
	00046000h ~ 000467FFh	32 位	3 (注 5、注 6)	外围总线时钟
	00046800h ~ 00047FFFh	32 位	2 (注 5、注 6)	外围总线时钟
	00048000h ~ 0004FFFFh	64 位	2	CPU 时钟
外部总线	00060000h ~ 01FFFFFFh FE000000h ~ FFDFFFFFFh	8 位 /16 位 /32 位	取决于 EBCn 寄存器 (n=0 ~ 3) 的设定值 (注 5)	外围总线时钟

注 1. 包括保留区。

注 2. 存取周期数以各自的总线时钟为基准。

注 3. 如果在同一页存取，就为 2 个周期，否则就为 3 个周期。

注 4. 如果连续发生写周期，第 2 个写周期就为 2 个周期，而且，紧接在写周期后的读周期也为 2 个周期。

注 5. 如果连续进行 SFR 存取，第 2 个以后的存取就增加 1 个周期的基本时钟。

注 6. 有可能因外围总线时钟的相位而最多增加 1 个周期。

源读的传送周期例子如图 13.11 所示。此图中表示在目标地址为内部 RAM 并且目标写周期为 1 个周期时的各条件下的源读周期数。实际上，目标写周期和源读周期一样也受各条件的影 响，而且传送周期也发生变化。在计算传送周期时，各条件必须适用于目标写周期和源读周期。如图 13.11 (2) 所示，在发生 2 个总线周期时，源读周期和目标写周期各自需要 2 个总线周期。

### 13.1.3 $\overline{\text{RDY}}$ 信号的影响

在存储器扩展模式和微处理器模式中，外部区域受  $\overline{\text{RDY}}$  信号的影响，详细内容请参照“9.3.7  $\overline{\text{RDY}}$  信号”。

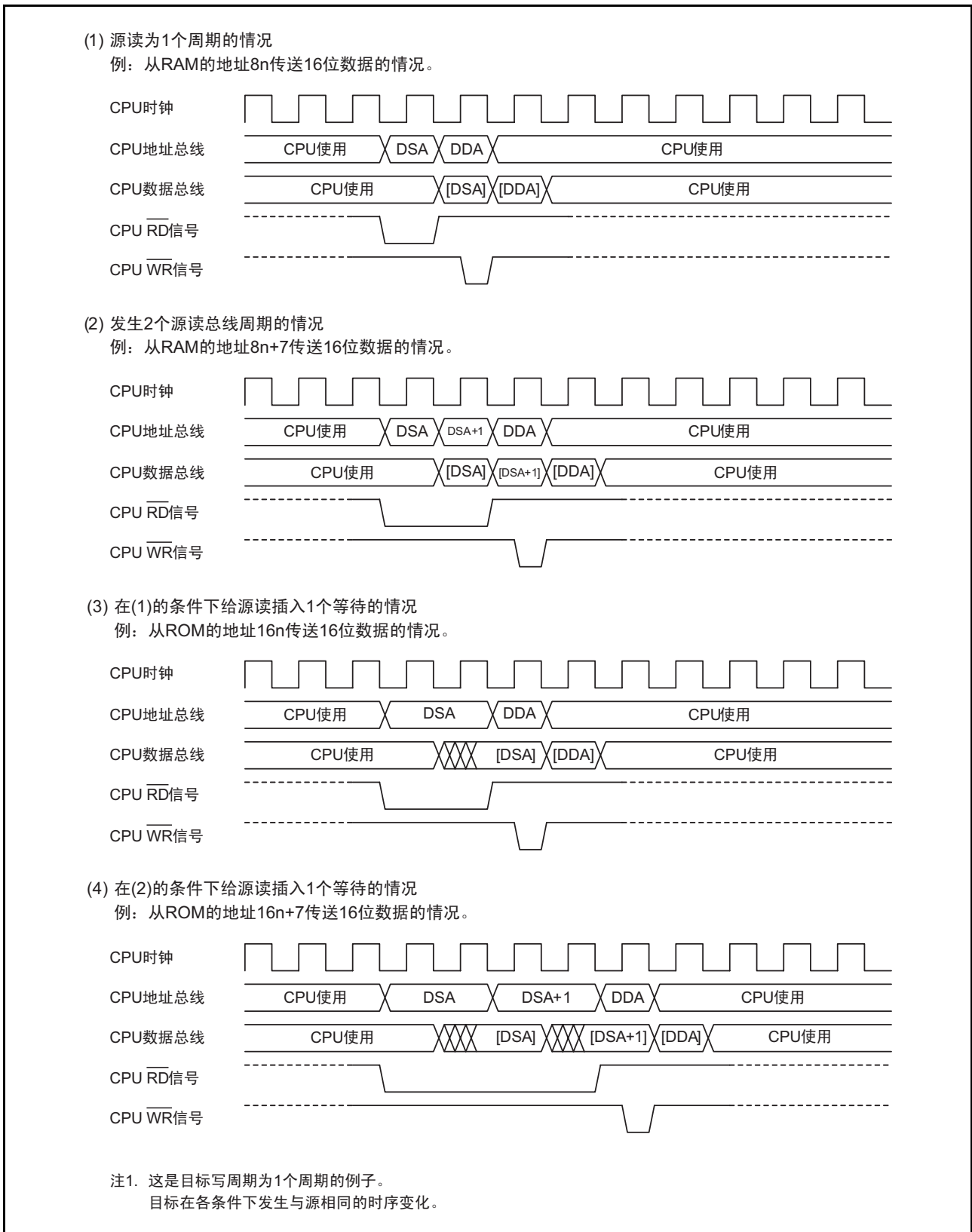


图 13.11 源读的传送周期例子

### 13.2 DMA 的传送周期数

能用下式计算 DMA 的传送周期数：

$$1 \text{ 个传送单位的传送周期数} = \text{读周期数} \times j + \text{写周期数} \times k + 1$$

读周期数是源读所需的总线周期数，写周期数是目标写所需的总线周期数，读周期数和写周期数各自最少需要 1 个周期。周期数有可能因地址而增加，因此必须根据表 13.4 求总线周期数。

$j$  为读所需的存取周期数， $k$  为写所需的存取周期数，请参照表 13.5。

最后的“+1”是 DCTi 寄存器（ $i=0 \sim 3$ ）的减法周期。

例如，通过 DMA 传送将 32 位数据从地址 400h（RAM）传送到地址 800h（RAM）时，则传送周期数为：

$$\begin{aligned} \text{传送周期数} &= 1 \times 1 + 1 \times 1 + 1 \\ &= 3 \end{aligned}$$

当外围总线时钟的频率为 CPU 时钟频率的 1/2 时，通过 DMA 传送将 16 位数据从 AD00 寄存器（地址 380h）传送到 P1 寄存器（地址 3C1h）和 P0 寄存器（地址 3C0h），则传送周期数为：

$$\begin{aligned} \text{传送周期数} &= 1 \times 2 \times 2 + 1 \times 2 \times 2 + 1 \\ &= 9 \end{aligned}$$

### 13.3 通道优先级和 DMA 传送时序

如果在同一采样期间（从 CPU 时钟的下降沿到下一个下降沿的一个周期）发生多个 DMA 传送请求，这些请求就同时被输入到 DMAC，此时的通道优先级为 DMA0 > DMA1 > DMA2 > DMA3。

以下说明在同一采样期间发生 DMA0 和 DMA1 的传送请求时的运行以及图 13.12 所示的外部源引起的 DMA 传送例子。

在图 13.12 中，因为同时发生了 DMA0 和 DMA1 的传送请求，所以通道优先级高的 DMA0 先开始传送，在 1 个传送单位的传送结束后，将总线使用权让给 CPU。当 CPU 结束 1 次总线存取时，DMA1 就接着开始传送，在 1 个传送单位的传送结束后，将总线使用权让给 CPU。

另外，因为不能对 DMA 传送请求的次数进行计数，所以如图 13.12 中的 DMA1 所示，如果在允许使用总线前发生多次 INT1 中断，传送次数也是 1。

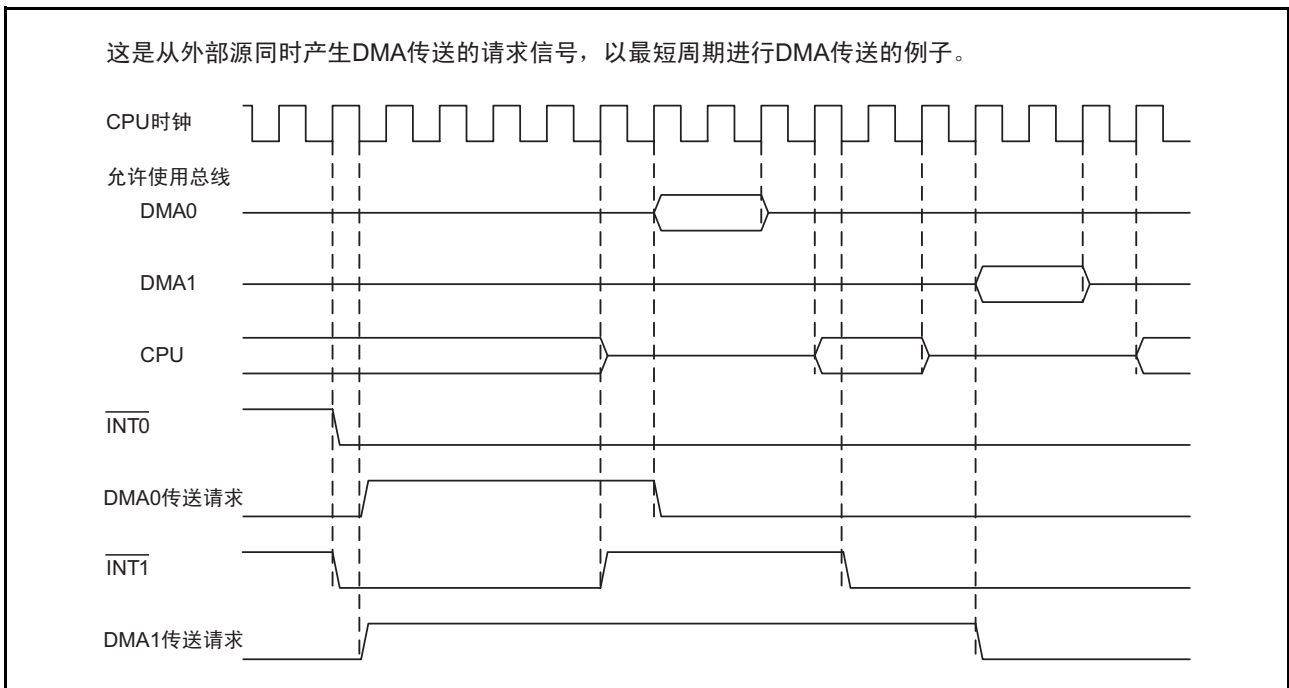


图 13.12 外部源产生的 DMA 传送例子

## 13.4 使用 DMA 控制器时的注意事项

### 13.4.1 DMAC 相关寄存器的设定

- 必须在要设定通道的DMDi寄存器 (i=0~3)的MDi1~MDi0位为“00b”(禁止DMA传送)的状态下,设定DMAC的相关寄存器,最后将MDi1~MDi0位置“01b”(单次传送)或者“11b”(重复传送)。也必须在MDi1~MDi0位为“00b”(禁止DMA传送)时改写DMDi寄存器的UDAi、USAi和BWi1~BWi0位。
- 如果在允许DMA传送后需要改写DMAC的相关寄存器,首先为了不产生DMA传送请求,必须停止作为DMA启动源的外围功能,其次将要改写的通道的DMDi寄存器的MDi1~MDi0位置“00b”(禁止DMA传送),然后进行改写。
- 一旦接受DMA传送请求,即使将DMDi寄存器的MDi1~MDi0位置“00b”(禁止DMA传送),也无法禁止DMA传送。此时,不能在DMA传送结束前更改除MDi1~MDi0位以外的DMAC相关寄存器的设定。
- 在设定DMiSL寄存器和DMiSL2寄存器后,必须先通过外围总线时钟至少等待6个时钟,然后给DMDi寄存器的MDi1~MDi0位写“01b”(单次传送)或者“11b”(重复传送)。

### 13.4.2 读 DMAC 相关寄存器

- 必须按照以下的顺序分别连续读DMiSL寄存器和DMiSL2寄存器:  
DM0SL→DM1SL→DM2SL→DM3SL  
DM0SL2→DM1SL2→DM2SL2→DM3SL2

## 14. DMAC II

DMAC II 通过外围功能的中断请求来启动，并且不使用 CPU 的指令进行数据传送，能选择存储器、立即数、存储器 + 存储器或者立即数 + 存储器作为传送对象。

DMAC II 的规格如表 14.1 所示。

表 14.1 DMAC II 的规格

项目	规格
DMAC II 的启动源	将中断控制寄存器的 ILVL2 ~ ILVL0 位置 “111b” (7 级) 后的全部外围功能的中断请求。
传送对象	<ul style="list-style-type: none"> <li>• 存储器 → 存储器 (存储器之间的传送)</li> <li>• 立即数 → 存储器 (立即数传送)</li> <li>• 存储器 + 存储器 → 存储器 (运算传送)</li> <li>• 立即数 + 存储器 → 存储器 (运算传送)</li> </ul>
传送大小	8 位、16 位
传送空间	将数据从 64M 字节 (00000000h ~ 01FFFFFFh 和 FE000000h ~ FFFFFFFFh) 的任意空间传送到 64M 字节的任意空间 (注 1)。
寻址	传送源和传送目标可分别选择以下 2 种： <ul style="list-style-type: none"> <li>• 固定：每次的地址相同。</li> <li>• 递增：在每次传送数据时，地址 +1 (传送大小 =8 位) 或者地址 +2 (传送大小 =16 位)。</li> </ul>
传送方式	<ul style="list-style-type: none"> <li>• 单次传送：1 次传送请求只进行 1 次数据传送。</li> <li>• 突发传送：通过 1 次传送请求，连续进行传送计数器所定次数的数据传送。</li> <li>• 多次传送：通过 1 次传送请求，对不同的传送源 / 传送目标进行多次存储器之间的传送。</li> </ul>
链传送功能	通过变换多个 DMAC II 索引 (传送信息) 的顺序进行数据传送。
传送结束中断请求	如果传送计数器为 “0000h”，就产生传送结束中断请求。

注 1. 但是，当传送大小为 16 位并且传送目标地址为 “FFFFFFFh” 时，将数据传送到地址 FFFFFFFFh 和地址 00000000h。传送源地址为 “FFFFFFFh” 时的操作也一样。

## 14.1 DMAC II 的设定

在使用 DMAC II 时，必须进行以下的设定：

- RIPL1 寄存器和 RIPL2 寄存器
- DMAC II 索引
- 作为 DMAC II 启动源的外围功能的中断控制寄存器
- 作为 DMAC II 启动源的外围功能的可变向量
- 在使用智能 I/O 中断时，必须设定 IIOiIE 寄存器 (i=0~11) 的 IRLT 位。有关 IIOiIE 寄存器，请参照“11. 中断”。

### 14.1.1 RIPL1 寄存器和 RIPL2 寄存器

如果将 RIPL1 寄存器和 RIPL2 寄存器的 DMAII 位都置“1”（DMA II 传送）并且将 FSIT 位置“0”（正常中断），就通过中断控制寄存器的 ILVL2 ~ ILVL0 位被置“111b”（7 级）后的全部外围功能的中断请求来启动 DMAC II。

RIPL1 寄存器和 RIPL2 寄存器如图 14.1 所示。

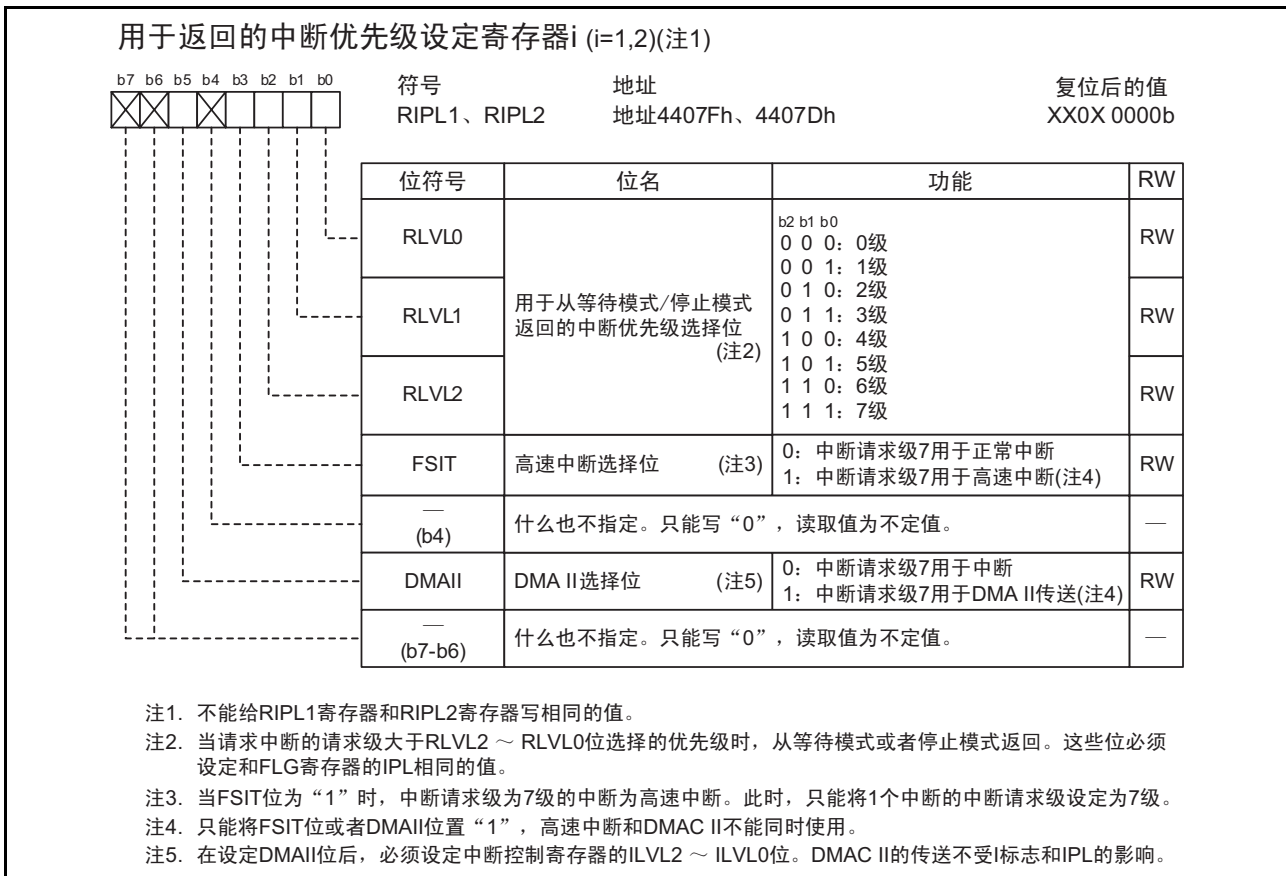


图 14.1 RIPL1 寄存器和 RIPL2 寄存器



### 14.1.2 DMAC II 索引

DMAC II 索引是由 12 ~ 60 字节构成的数据表，保存传送模式、传送计数器、传送源地址（或者立即数的数据）、运算对象的地址、传送目标地址、链传送基址、传送结束中断向量地址等参数。

必须将 DMAC II 索引分配到 RAM。

DMAC II 索引及其记述的例子分别如图 14.2 和表 14.2 所示。

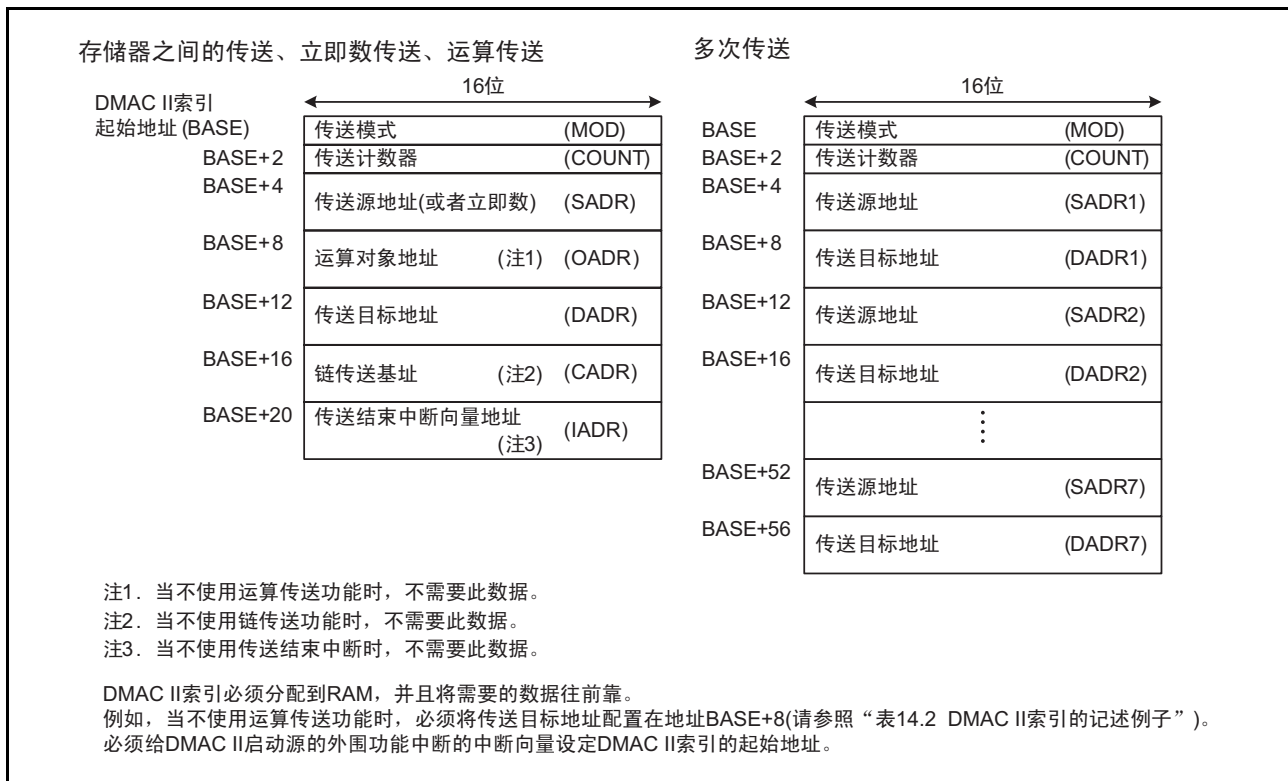


图 14.2 DMAC II 索引

以下说明 DMAC II 索引的内容。必须根据所使用的 DMAC II 的传送模式，按照表 14.2 所示的顺序来分配这些数据。

- **传送模式 (MOD)**  
这是2字节数据，必须设定传送模式，传送模式的设定内容如图 14.3 所示。
- **传送计数器 (COUNT)**  
这是2字节数据，必须设定传送次数。
- **传送源地址 (SADR)**  
这是4字节数据，必须设定传送源的存储器地址或者立即数。在设定立即数时，忽视高位2字节。
- **运算对象地址 (OADR)**  
这是4字节数据，必须设定运算对象的存储器地址。只有在使用运算传送功能时，才能设定此数据。
- **传送目标地址 (DADR)**  
这是4字节数据，必须设定传送目标的存储器地址。
- **链传送基址 (CADR)**  
这是4字节数据，必须设定下一次传送的 DMAC II 索引的起始地址 (BASE)。只有在使用链传送功能时，才能设定此数据。
- **传送结束中断向量地址 (IADR)**  
这是4字节数据，必须设定传送结束中断处理的转移地址。只有在使用传送结束中断时，才能设定此数据。

在以下的各项说明中，各参数的名称使用上述（）内的符号。

表 14.2 DMAC II 索引的记述例子

传送数据	存储器之间的传送 / 立即数传送				运算传送				多次传送	
链传送	不使用	使用	不使用	使用	不使用	使用	不使用	使用	不能使用	
传送结束 中断	不使用	不使用	使用	使用	不使用	不使用	使用	使用	不能使用	
DMAC II 索引	MOD	MOD	MOD	MOD	MOD	MOD	MOD	MOD	MOD	
	COUNT	COUNT	COUNT	COUNT	COUNT	COUNT	COUNT	COUNT	COUNT	
	SADR	SADR	SADR	SADR	SADR	SADR	SADR	SADR	SADR1	
	DADR	DADR	DADR	DADR	OADR	OADR	OADR	OADR	DADR1	
	12 字节	CADR	IADR	CADR	IADR	DADR	DADR	DADR	DADR	
		16 字节	16 字节	16 字节	20 字节	CADR	IADR	CADR	IADR	SADRi
	20 字节					20 字节	20 字节	IADR	DADRi	
								24 字节	i=1 ~ 7 最大 60 字 节 (i=7)	

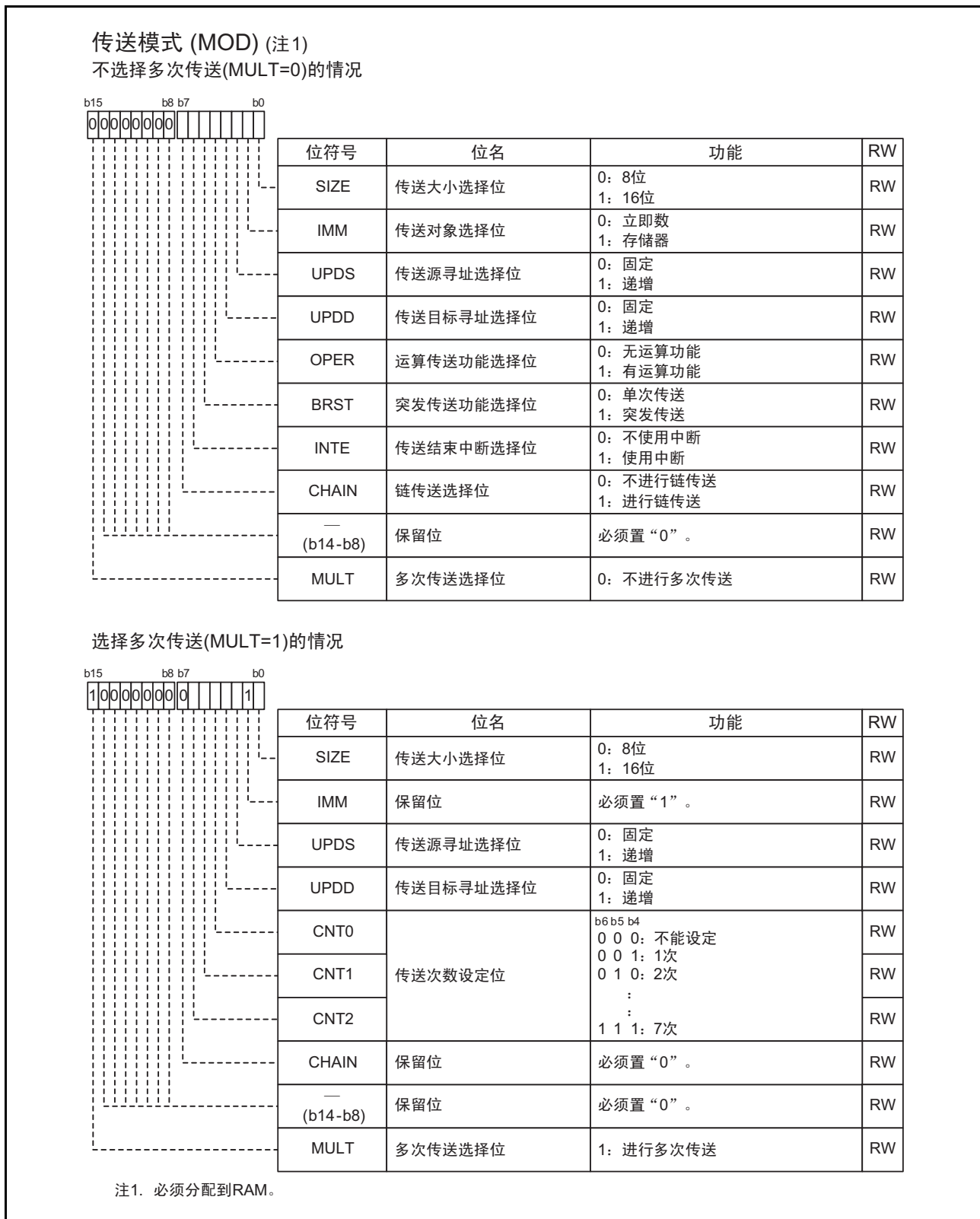


图 14.3 MOD

### 14.1.3 外围功能的中断控制寄存器

必须将用作 DMAC II 启动源的外围功能中断的中断控制寄存器的 ILVL2 ~ ILVL0 位置 “111b” (7 级)。

### 14.1.4 外围功能的可变向量表

必须给用作 DMAC II 启动源的外围功能中断的中断向量设定 DMAC II 索引的起始地址。

在使用链传送时, 必须将可变向量表分配到 RAM。

### 14.1.5 IIOiE 寄存器 (i=0 ~ 11) 的 IRLT 位

在通过智能 I/O 中断启动 DMAC II 时, 必须将用作启动源的中断的 IIOiE 寄存器 IRLT 位置 “0” (中断请求用于 DMA 和 DMA II)。

## 14.2 DMAC II 的运行

如果将 RIPL1 寄存器和 RIPL2 寄存器的 DMAII 位都置 “1” (中断请求级 7 用于 DMA II 传送), 就选择 DMA II 传送功能。中断控制寄存器的 ILVL2 ~ ILVL0 位被置 “111b” (7 级) 后的全部外围功能的中断请求作为 DMAC II 的启动源。因为这些外围功能的中断请求为 DMA II 传送请求, 所以不能用作 CPU 的中断请求。

如果发生 ILVL2 ~ ILVL0 位为 “111b” 的中断请求, DMAC II 就启动, 与 I 标志和 IPL 无关。

另外, 如果用作 DMAC II 启动源的外围功能中断请求和优先级更高的中断请求 (看门狗定时器中断、低电压检测中断、振荡停止检测中断、NMI) 同时发生, 就先接受优先级高于 DMA II 传送的中断, 在该中断响应顺序结束后开始 DMA II 传送。

## 14.3 传送对象

对于以下 3 种传送对象, DMAC II 以 8 位或者 16 位为单位进行数据传送:

- 存储器之间的传送: 将数据从 64M 字节空间 (地址 00000000h ~ 01FFFFFFh 和地址 FE000000h ~ FFFFFFFFh) 的任意存储器传送到同一空间的任意存储器。
- 立即数传送: 将立即数的数据传送到 64M 字节空间的任意存储器。
- 运算传送: 将 2 个数据相加, 并将结果传送到 64M 字节空间的任意存储器。

但是, 当传送大小为 16 位并且 DADR 为 “FFFFFFFh” 时, 将数据传送到地址 FFFFFFFFh 和地址 00000000h。SADR 为 “FFFFFFFh” 时的操作也一样。

### 14.3.1 存储器之间的传送

将数据从任意的存储器传送到任意的存储器, 传送种类如下:

- 从地址固定的存储器传送到地址固定的存储器。
- 从地址固定的存储器传送到连续的存储器区域。
- 从连续的存储器区域传送到地址固定的存储器。
- 从连续的存储器区域传送到连续的存储器区域。

如果选择 “递增” 寻址, 就在传送后为了下一次传送而递增 SADR 或者 DADR。当传送大小为 8 位时, 为 “地址 +1”; 当传送大小为 16 位时, 为 “地址 +2”。如果因地址相加而使 SADR 或者 DADR 超过 “FFFFFFFh”, 地址就返回到 “00000000h”。同样, 如果 SADR 或者 DADR 超过 “01FFFFFFh”, 地址就变为 “02000000h”, 但是实际上是对地址 FE000000h 进行传送。

### 14.3.2 立即数传送

将立即数传送到任意的存储器。传送目标的寻址能选择“固定”或者“递增”。必须将立即数保存到 SADR。在传送 8 位立即数时，必须将数据设定到 SADR 的低位 1 字节（忽视高位 3 字节）；在传送 16 位立即数时，必须将数据设定到 SADR 的低位 2 字节（忽视高位 2 字节）。

### 14.3.3 运算传送

将任意存储器的内容和任意存储器的内容、或者将立即数和任意存储器的内容相加，结果传送到任意的存储器。必须将运算对象数据的地址或者立即数设定到 SADR，并且将另一个的运算对象数据的地址设定到 OADR。在进行存储器 + 存储器的运算传送时，传送源和传送目标的寻址能选择“固定”或者“递增”。当传送源的寻址为“递增”时，运算对象的寻址也为“递增”。在进行立即数 + 存储器的运算传送时，只有传送目标的寻址能选择“固定”或者“递增”。

## 14.4 传送方式

DMAC II 能进行单次传送、突发传送和多次传送，通过 COUNT 设定传送次数。当 COUNT 为“0000h”时，不进行传送。

### 14.4.1 单次传送

如果将 MOD 的 BRST 位置“0”，就选择单次传送。

1 次传送请求只进行 1 次数据传送。

如果传送源或者传送目标的寻址选择“递增”，就在传送后为了下一次传送而递增地址。在每次数据传送时，COUNT 进行递减计数。当 MOD 的 INTE 位为“1”（使用传送结束中断）并且 COUNT 变为“0000h”时，发生传送结束中断请求。

### 14.4.2 突发传送

如果将 MOD 的 BRST 位置“1”，就选择突发传送。

通过 1 次传送请求，连续进行 COUNT 所设次数的数据传送。

在每次数据传送时，COUNT 进行递减计数。在 COUNT 变为“0000h”时，突发传送结束。当 INTE 位为“1”（使用传送结束中断）并且突发传送结束时，发生传送结束中断请求。

另外，在突发传送的过程中不接受任何中断。

### 14.4.3 多次传送

如果将 MOD 的 MULT 位置“1”，就选择多次传送。

通过 1 次传送请求，对不同的传送源 / 传送目标进行多次存储器之间的传送。能通过 MOD 的 CNT2 ~ CNT0 位选择“001b”（1 次）~“111b”（7 次）的传送次数，但是不能将 CNT2 ~ CNT0 位置“000b”。

必须将传送次数的 SDAR 和 DADR 分别交替地分配到接在 MOD 和 COUNT 之后的地址。

在选择多次传送时，不能使用运算传送、突发传送、链传送和传送结束中断的各功能。

## 14.5 链传送

如果将 MOD 的 CHAIN 位置 “1”，链传送功能就有效。

链传送的运行如下：

1. 如果发生传送请求，就根据该中断源的中断向量所指向的DMAC II索引的内容进行数据传送。当BRST位为“0”时，进行单次传送；当BRST位为“1”时，进行突发传送。
2. 如果COUNT变为“0000h”，(1)的中断向量被改写为CADR的值。如果INTE位为“1”，就同时发生传送结束中断请求。
3. 如果发生下一个DMA II传送请求，就根据(2)中被改写的中断向量所指向的DMAC II索引的内容进行数据传送。

链传送的可变向量和 DMAC II 索引如图 14.4 所示。

在使用链传送时，必须将可变向量表分配到 RAM。

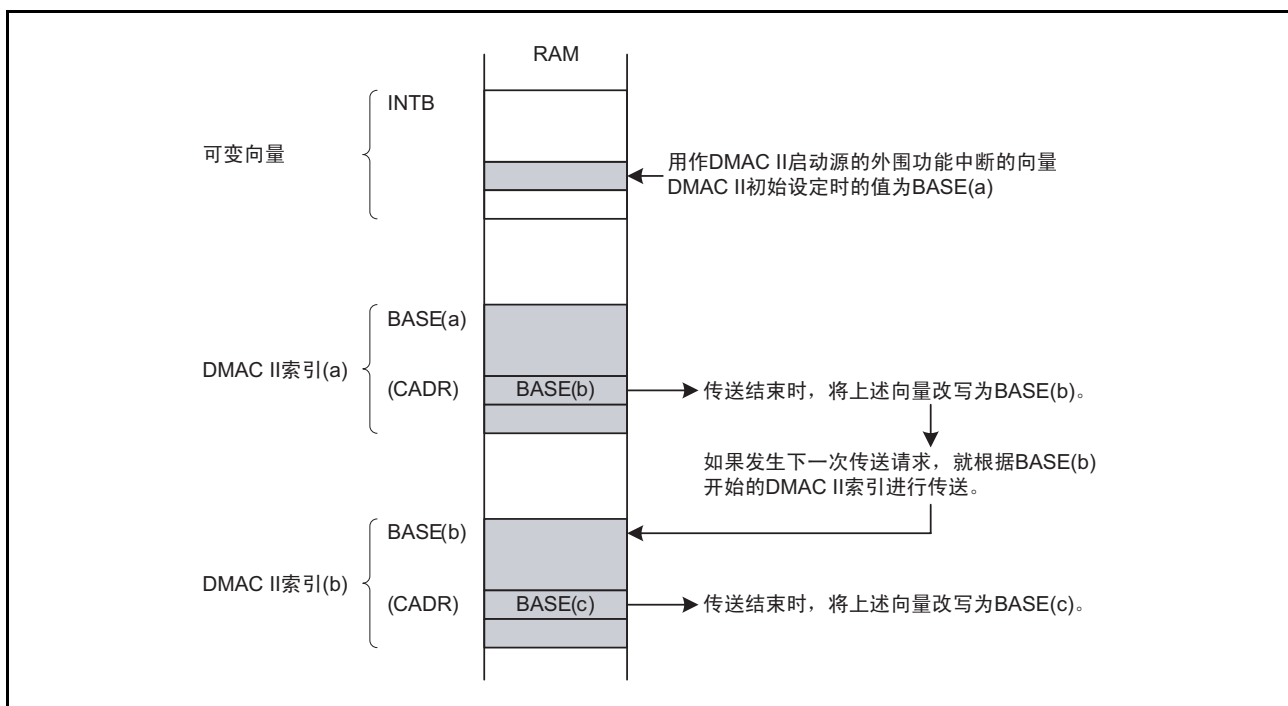


图 14.4 链传送的可变向量和 DMAC II 索引

## 14.6 传送结束中断

如果将 MOD 的 INTE 位置 “1”，就能产生传送结束中断请求。必须将传送结束中断处理程序的起始地址设定到 IADR。如果 COUNT 变为 “0000h”，就产生传送结束中断请求。

在 DMA II 传送结束后的 7 个周期后，执行传送结束中断处理程序的第 1 条指令。

## 14.7 执行时间

能用下式计算 DMAC II 的执行周期数:

多次传送以外的传送:  $t=6+(26+a+b+c+d)\times m+(4+e)\times n$  [ 周期 ]

多次传送:  $t=21+(11+b+c)\times k$  [ 周期 ]

- a: 当 IMM=0 (传送源为立即数) 时,  $a=0$  ;  
当 IMM=1 (传送源为存储器) 时,  $a=-1$ 。
- b: 当 UPDS=1 (传送源寻址为递增) 时,  $b=0$  ;  
当 UPDS=0 (传送源寻址为固定) 时,  $b=1$ 。
- c: 当 UPDD=1 (传送目标寻址为递增) 时,  $c=0$  ;  
当 UPDD=0 (传送目标寻址为固定) 时,  $c=1$ 。
- d: 当 OPER=0 (无运算功能) 时,  $d=0$  ;  
当 OPER=1 (有运算功能) 并且 UPDS=0 (传送源为立即数或者地址固定) 时,  $d=7$  ;  
当 OPER=1 (有运算功能) 并且 UPDS=1 (传送源寻址为递增) 时,  $d=8$ 。
- e: 当 CHAIN=0 (无链传送功能) 时,  $e=0$  ;  
当 CHAIN=1 (有链传送功能) 时,  $e=4$ 。
- m: 当 BRST=0 (单次传送) 时,  $m=1$  ;  
当 BRST=1 (突发传送) 时,  $m=\text{COUNT}$ 。
- n: 当 COUNT 为 “0001h” 时,  $n=0$  ;  
当 COUNT 大于等于 “0002h” 时,  $n=1$ 。
- k: CNT2 ~ CNT0 位设定的传送数

上述计算式为概算值, 周期数因 CPU 的状态、总线等待和 DMAC II 索引的分配而不同。

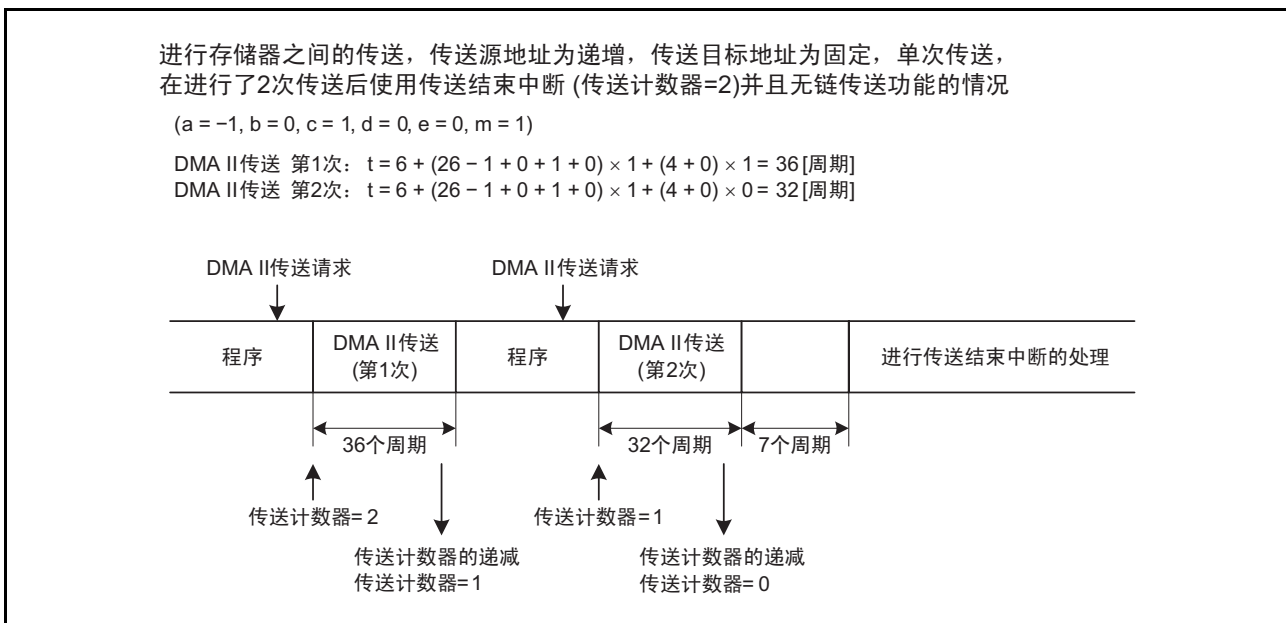


图 14.5 传送时间

### 15. 可编程输入 / 输出端口

100 引脚版有 P0 ~ P10 (P8\_5、P9\_0 ~ P9\_2 除外) 共 84 个可编程输入 / 输出端口；144 引脚版有 P0 ~ P15 (P8\_5、P14\_0 ~ P14\_2 除外) 共 120 个可编程输入 / 输出端口。

能通过方向寄存器设定各端口的输入 / 输出方向。但是，P8\_5、P9\_1 和 P14\_1 为输入专用端口。因为 P8\_5 和 NMI 引脚复用，所以能从 P8 寄存器的 P8\_5 位读 NMI 输入电平。

可编程输入 / 输出端口的结构如图 15.1 所示，输入专用端口的结构如图 15.2 ~ 图 15.4 所示。

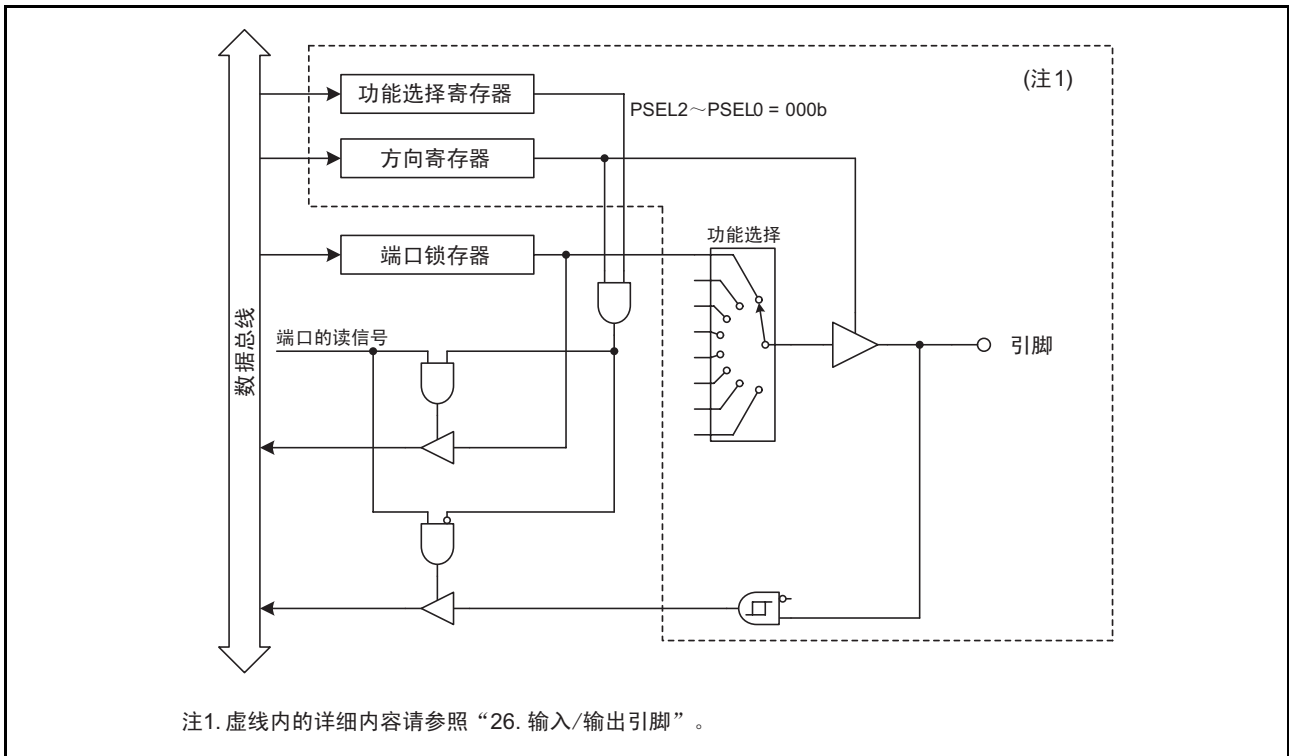


图 15.1 可编程输入 / 输出端口的结构

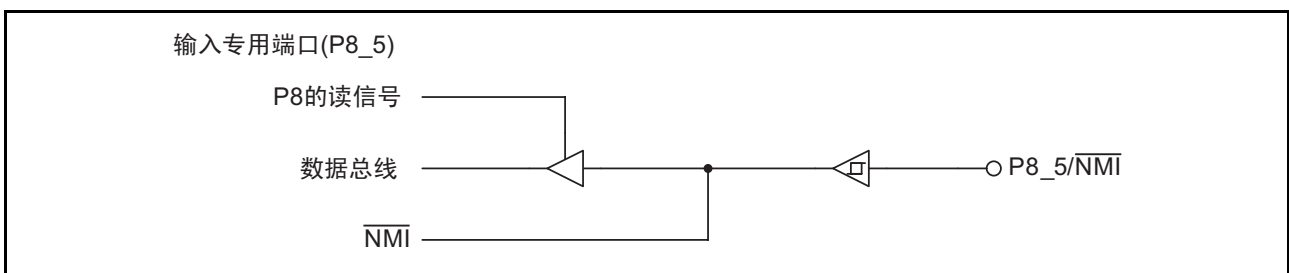


图 15.2 输入专用端口的结构 (1)

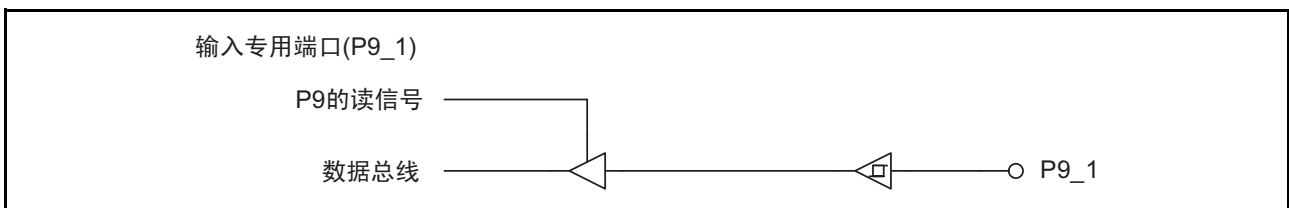


图 15.3 输入专用端口的结构 (2) (只限于 100 引脚版)



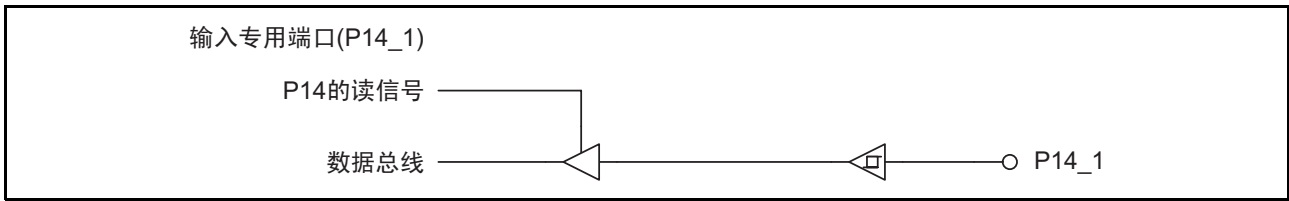


图 15.4 输入专用端口的结构 (3) (只限于 144 引脚版)

### 15.1 端口 Pi 寄存器 (Pi 寄存器, i=0 ~ 15)

通过读写 Pi 寄存器, 和外部进行数据的输入 / 输出。Pi 寄存器由保存输出数据的端口锁存器和读引脚状态的电路构成, Pi 寄存器的各位分别对应各端口。

如果通过输出功能选择寄存器选择可编程输入 / 输出端口, 就能在输出时读端口锁存器的值, 在输入时读引脚状态。

在存储器扩展模式或者微处理器模式中, 不能更改作为总线控制引脚 (A0 ~ A23、D0 ~ D31、CS0 ~ CS3、WR/WR0、BC0、BC1/WR1、BC2/WR2、BC3/WR3、RD、CLKOUT/BCLK、HLDA、HOLD、ALE、RDY) 的 Pi 寄存器。

Pi 寄存器如图 15.5 所示。

**端口Pi寄存器(i=0~15)(注1、注2)**

b7 b6 b5 b4 b3 b2 b1 b0	符号	地址	复位后的值
	P0、P1、P2、P3	地址03C0h、03C1h、03C4h、03C5h	不定值
	P4、P5、P6、P7	地址03C8h、03C9h、03CCh、03CDh	不定值
	P8 (注3)、P9 (注3、注4)	地址03D0h、03D1h	不定值
	P10、P11(注2、注5)	地址03D4h、03D5h	不定值
	P12、P13(注2)	地址03D8h、03D9h	不定值
	P14(注2、注3、注4、注5)	地址03DCh	不定值
	P15(注2)	地址03DDh	不定值

位符号	位名	功能	RW
Pi_0	端口Pi_0位 (注4)	方向位为“0”(输入)的情况 能给对应位写值, 但是因为 设定为输入, 所以不输出到引脚。 能读对应引脚的状态。	RW
Pi_1	端口Pi_1位 (注3)		RW
Pi_2	端口Pi_2位 (注4)		RW
Pi_3	端口Pi_3位		RW
Pi_4	端口Pi_4位	方向位为“1”(输出)的情况 写的值被反映为对应引脚的电平。	RW
Pi_5	端口Pi_5位 (注3、注5)		RW
Pi_6	端口Pi_6位 (注5)		RW
Pi_7	端口Pi_7位 (注5)		RW

注1. 在存储器扩展模式或者微处理器模式中, 不能更改作为总线控制引脚(A0~A23、D0~D31、CS0~CS3、WR/WR0、BC0、BC1/WR1、BC2/WR2、BC3/WR3、RD、CLKOUT/BCLK、HLDA、HOLD、ALE、RDY)的端口寄存器。

注2. P11~P15寄存器只限于144引脚版。

注3. P8寄存器的P8\_5位、P9寄存器的P9\_1位(100引脚版的情况)和P14寄存器的P14\_1位(144引脚版的情况)为只读位。

注4. P9寄存器的P9\_0位、P9\_2位(100引脚版的情况)和P14寄存器的P14\_0位、P14\_2位(144引脚版的情况)为保留位。只能写“0”, 读取值为不定值。

注5. P11寄存器的P11\_5位~P11\_7位和P14寄存器的P14\_7位什么也不指定。只能写“0”, 读取值为不定值。

图 15.5 P0 ~ P15 寄存器

## 16. 定时器

R32C/117 群有 11 个 16 位定时器。根据这些定时器所具有的功能，能分为定时器 A（5 个）和定时器 B（6 个）。全部定时器各自独立运行，各定时器的计数源为计数和重加载等的定时器运行时钟。

定时器 A 和定时器 B 的结构分别如图 16.1 和图 16.2 所示。

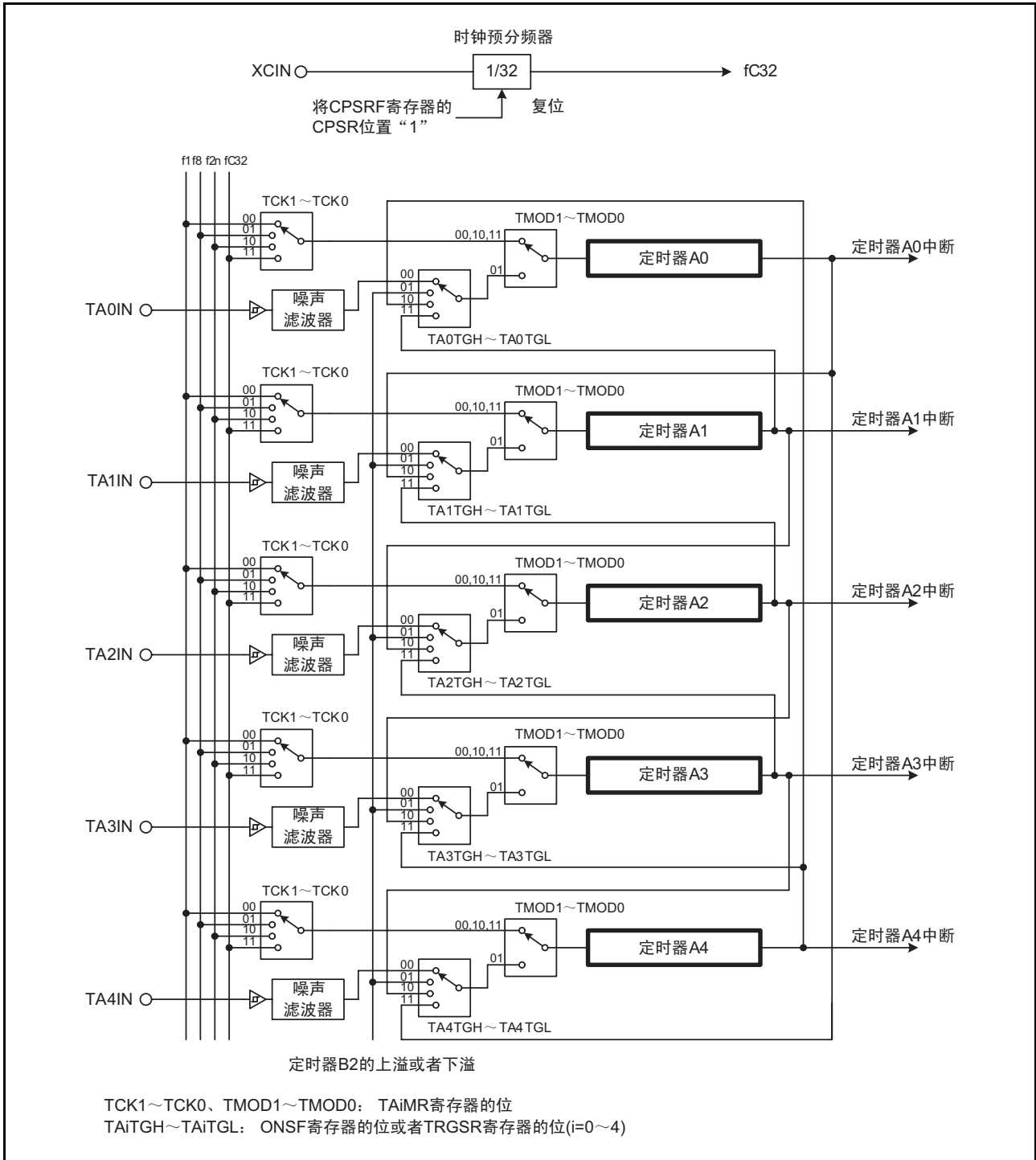


图 16.1 定时器 A 的结构

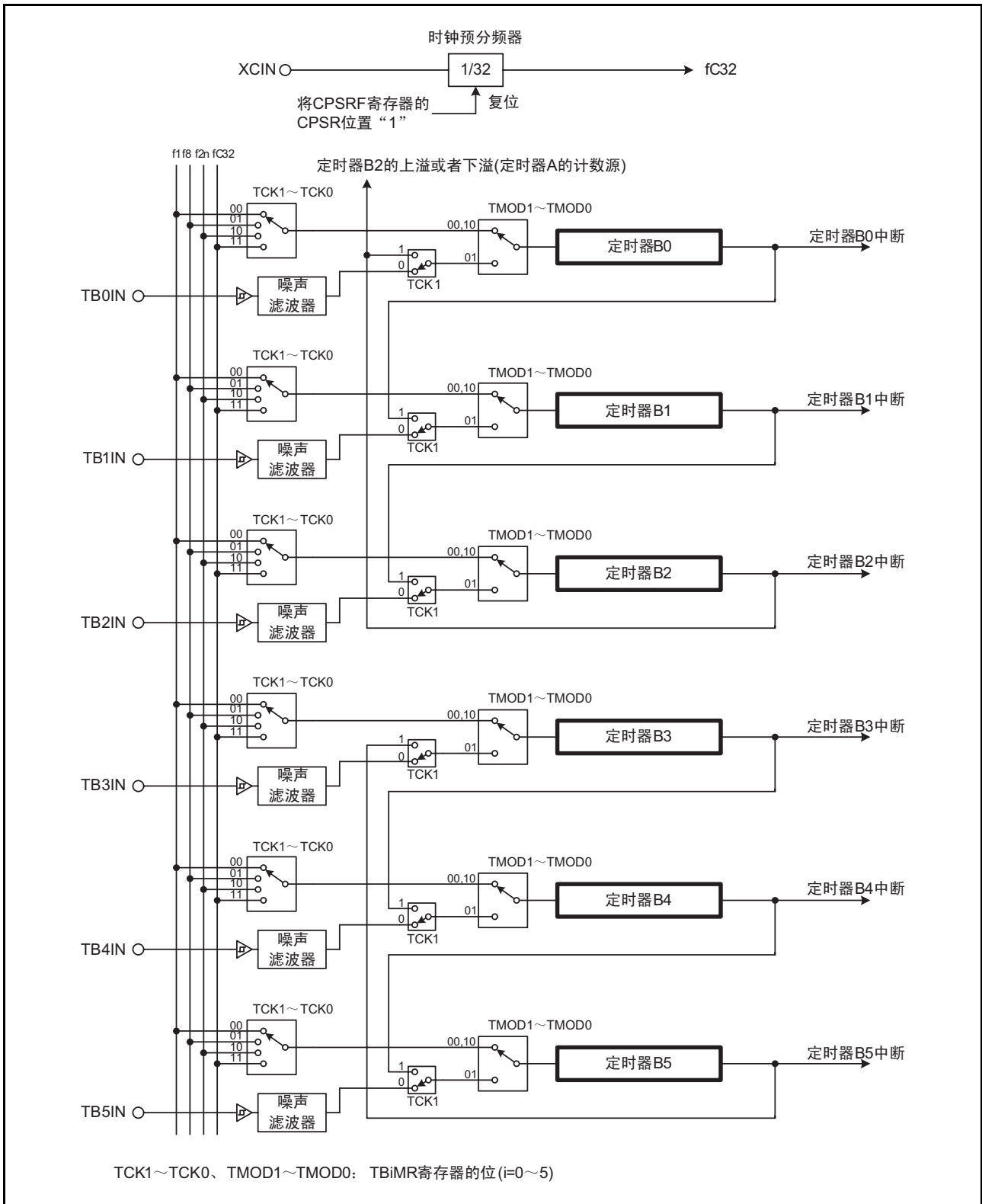


图 16.2 定时器 B 的结构

### 16.1 定时器 A

定时器 A 的框图及其相关寄存器分别如图 16.3 和图 16.4 ~ 图 16.10 所示。

定时器 A 有以下 4 种模式，除事件计数器模式以外，定时器 A0 ~ A4 具有相同的功能。能通过 TA0MR ~ TA4MR 寄存器的 TMOD1 ~ TMOD0 位选择各模式。

- 定时器模式                      这是对内部计数源进行计数的模式。
- 事件计数器模式                这是对外部脉冲或者其他定时器的上溢/下溢进行计数的模式。
- 单触发定时器模式              这是在计数值变为“0000h”前，只输出1次脉冲的模式。
- 脉宽调制模式                  这是连续输出任意宽度的脉冲的模式。

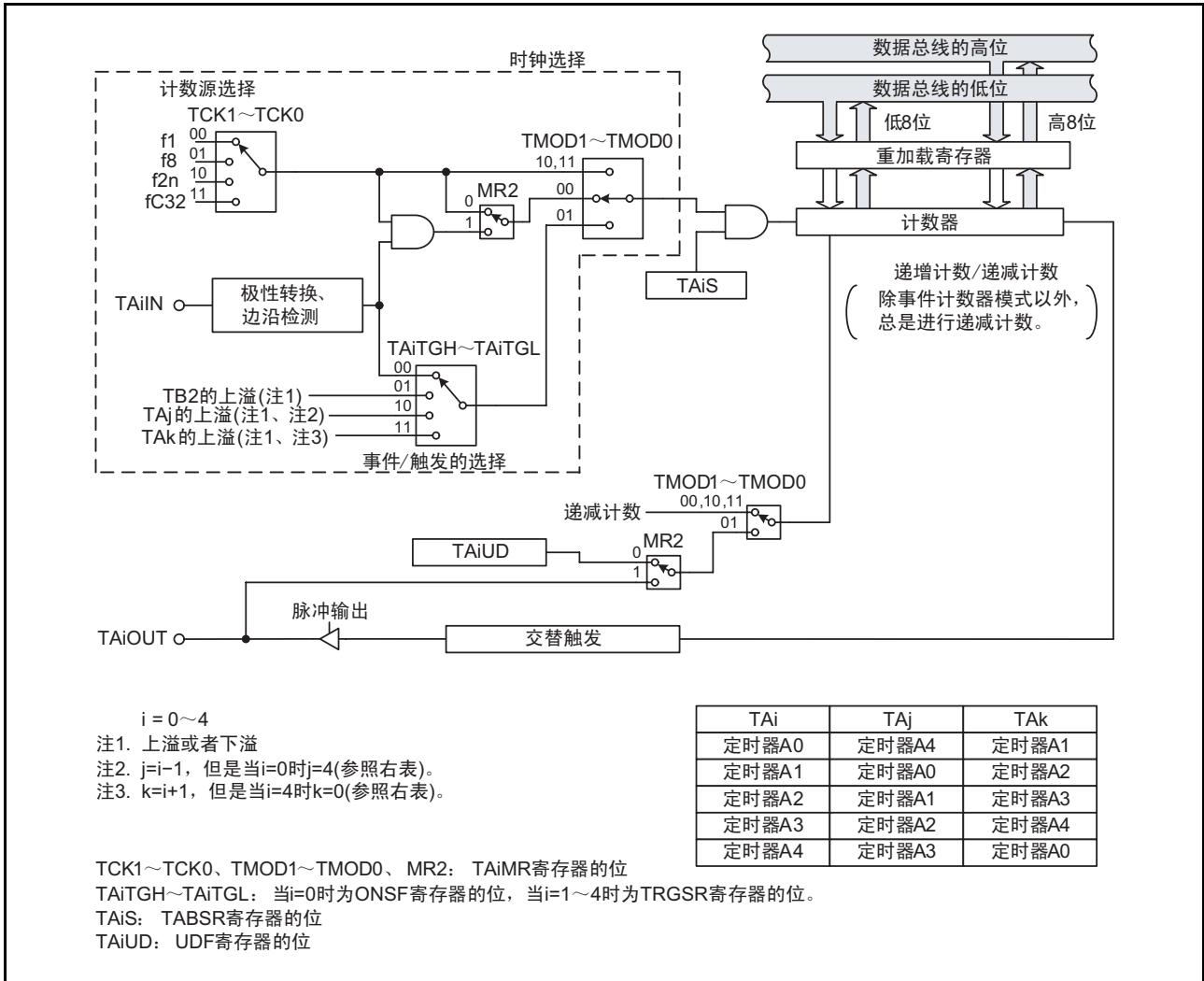


图 16.3 定时器 A 的框图

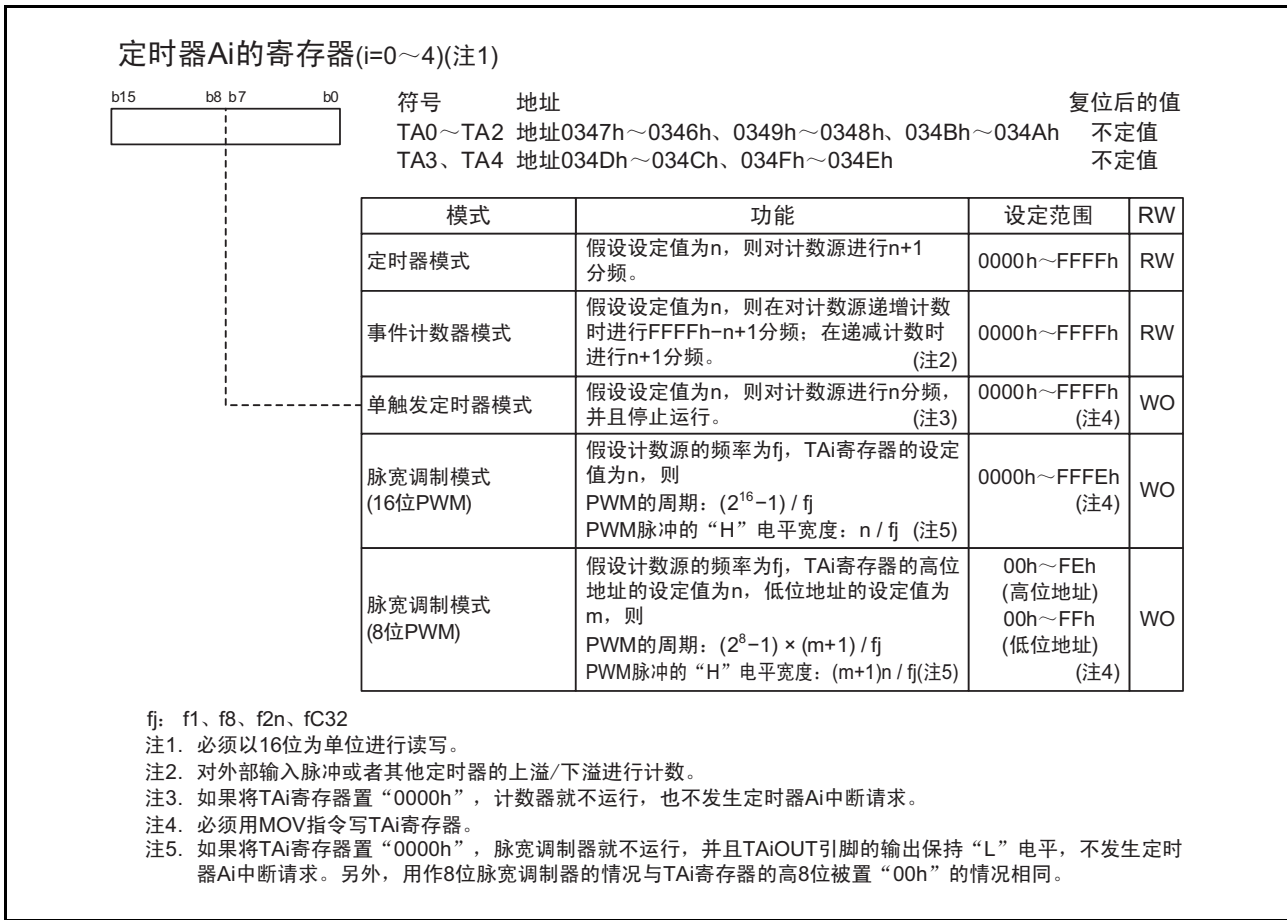


图 16.4 TA0 ~ TA4 寄存器

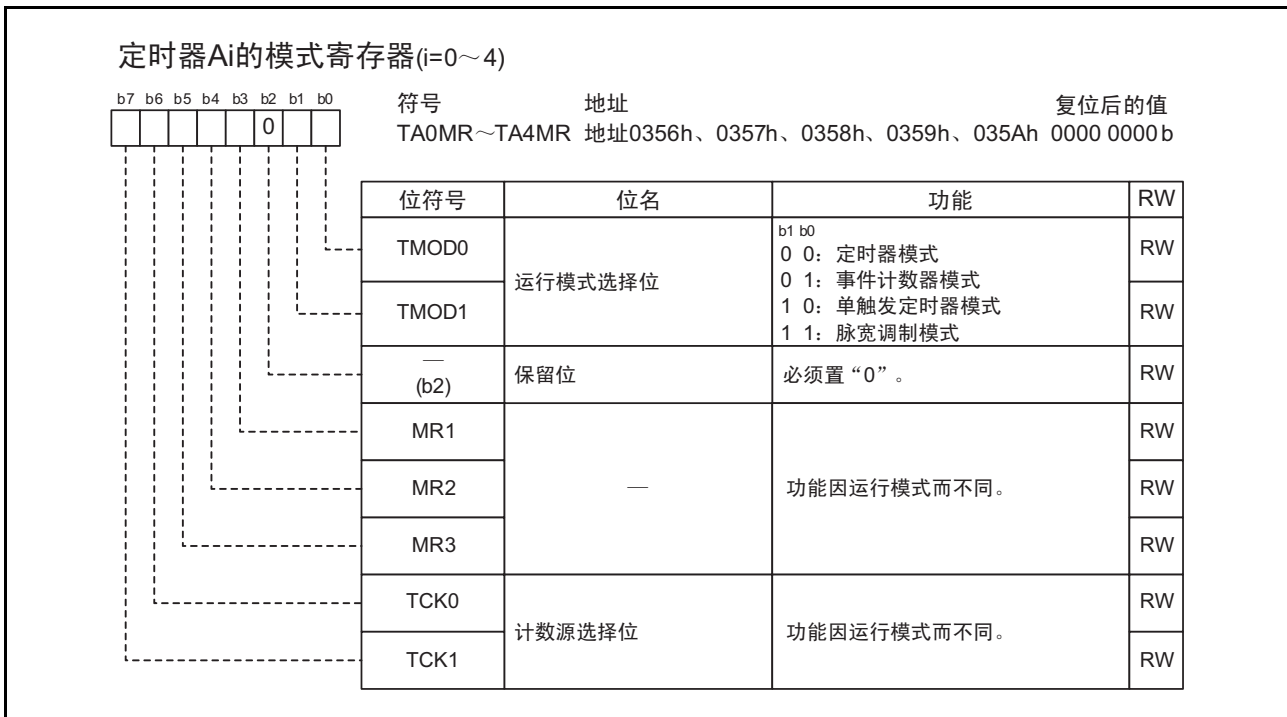


图 16.5 TA0MR ~ TA4MR 寄存器

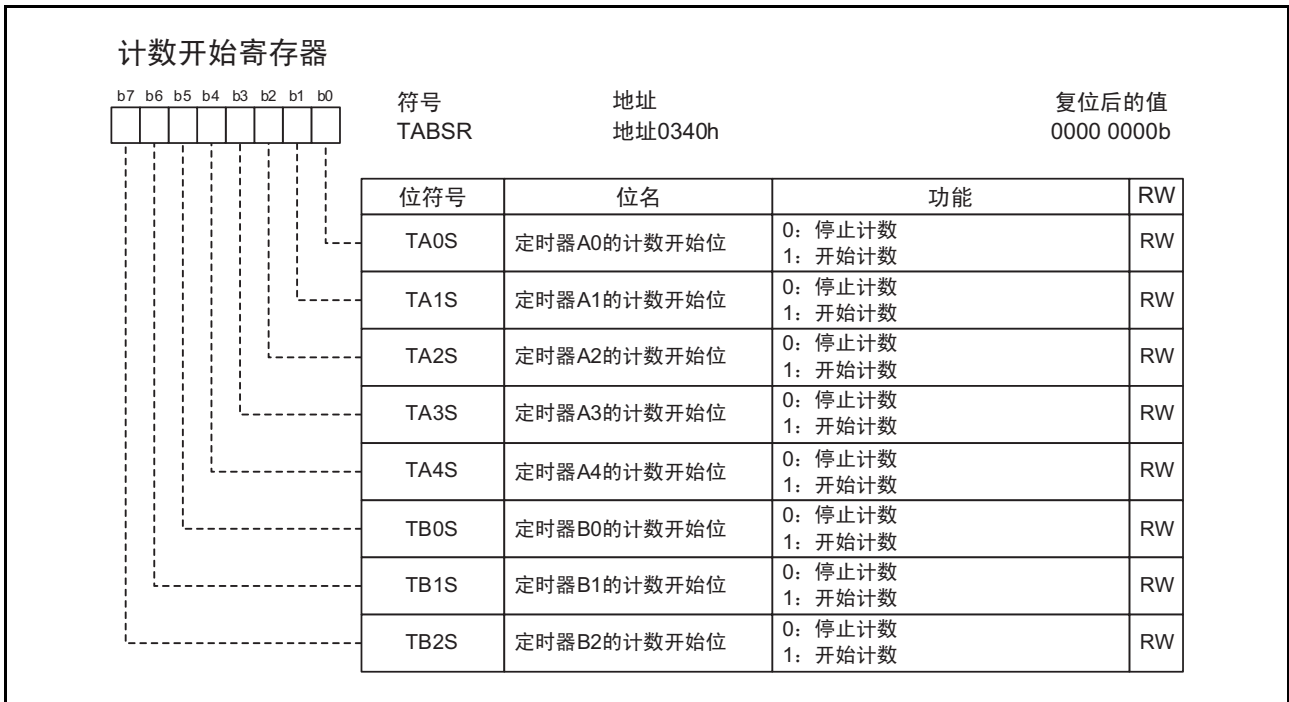


图 16.6 TABSR 寄存器

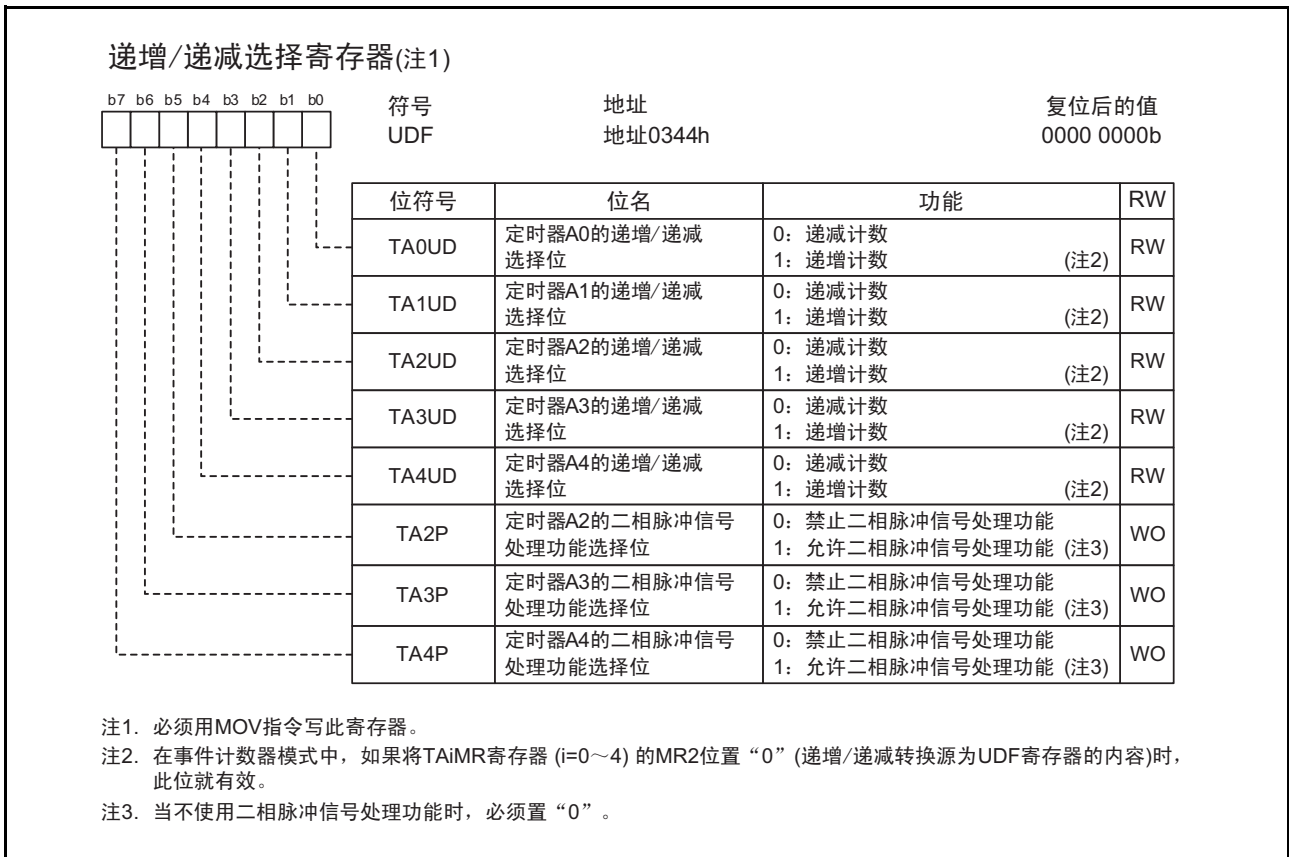


图 16.7 UDF 寄存器

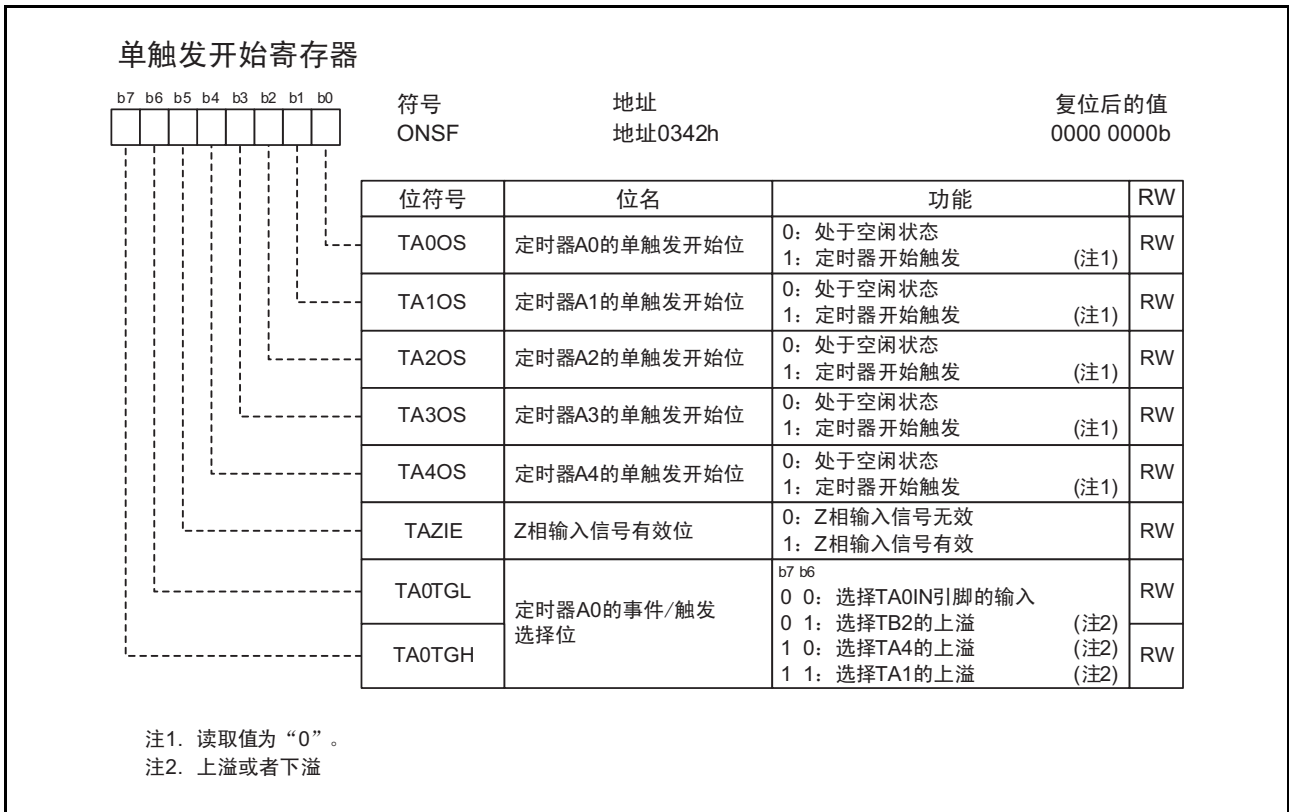


图 16.8 ONSF 寄存器

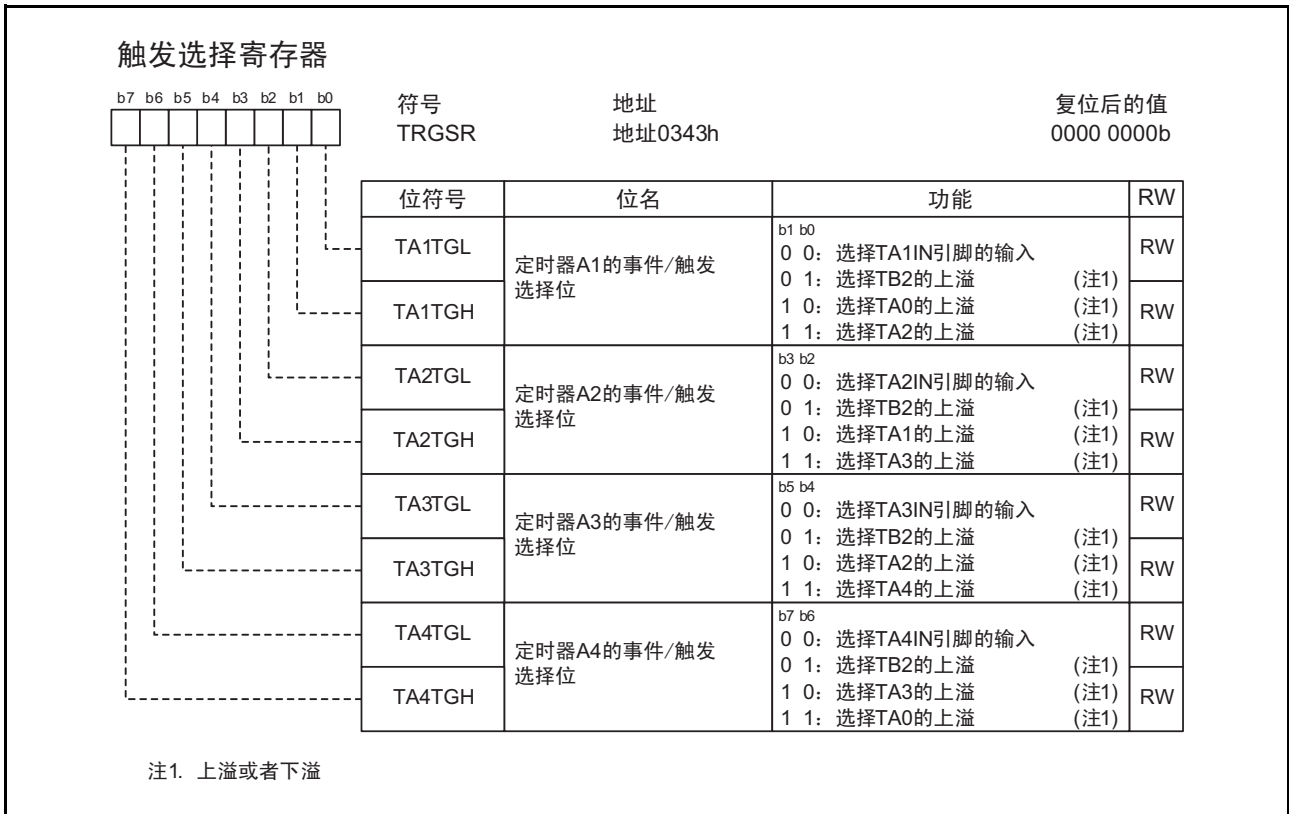


图 16.9 TRGSR 寄存器

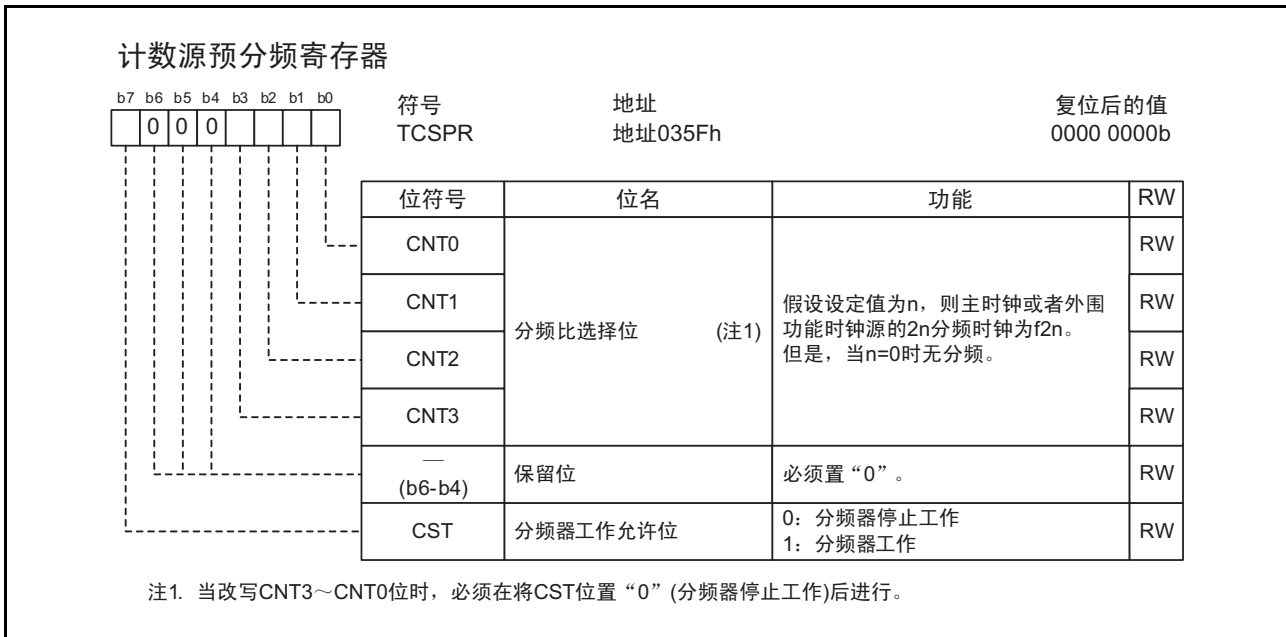


图 16.10 TCSR 寄存器

### 16.1.1 定时器模式

这是对内部生成的计数源进行计数的模式。定时器模式的规格以及定时器模式的 TA0MR ~ TA4MR 寄存器分别如表 16.1 和图 16.11 所示。

表 16.1 定时器模式的规格 (i=0 ~ 4)

项目	规格
计数源	f1、f8、f2n、fC32
计数	<ul style="list-style-type: none"> <li>• 递减计数</li> <li>• 在发生下溢时, 将重加载寄存器的内容进行重新加载后继续计数。</li> </ul>
分频比	$\frac{1}{n+1}$ n: TAI寄存器的设定值 (0000h ~ FFFFh)
计数开始条件	将 TABSR 寄存器的 TAI <sub>S</sub> 位置“1”(开始计数)。
计数停止条件	将 TABSR 寄存器的 TAI <sub>S</sub> 位置“0”(停止计数)。
中断请求的发生时序	在发生下溢时。
TAiIN 引脚功能	可编程输入 / 输出端口或者选通输入引脚
TAiOUT 引脚功能	可编程输入 / 输出端口或者脉冲输出引脚
读定时器	如果读 TAI 寄存器, 就能读到计数值。
写定时器	<ul style="list-style-type: none"> <li>• 在停止计数时或者在开始计数后到输入第 1 个计数源前, 如果写 TAI 寄存器, 数据就被写到重加载寄存器和计数器。</li> <li>• 如果在计数时写 TAI 寄存器, 数据就被写到重加载寄存器 (在下次重新加载时进行传送)。</li> </ul>
选择功能	<ul style="list-style-type: none"> <li>• 选通功能 能通过 TAIiN 引脚的输入信号开始或者停止计数。</li> <li>• 脉冲输出功能 在每次下溢时, 反转 TAIiOUT 引脚的极性。 在 TAI<sub>S</sub> 位为“0”(停止计数)期间, 输出“L”电平。</li> </ul>



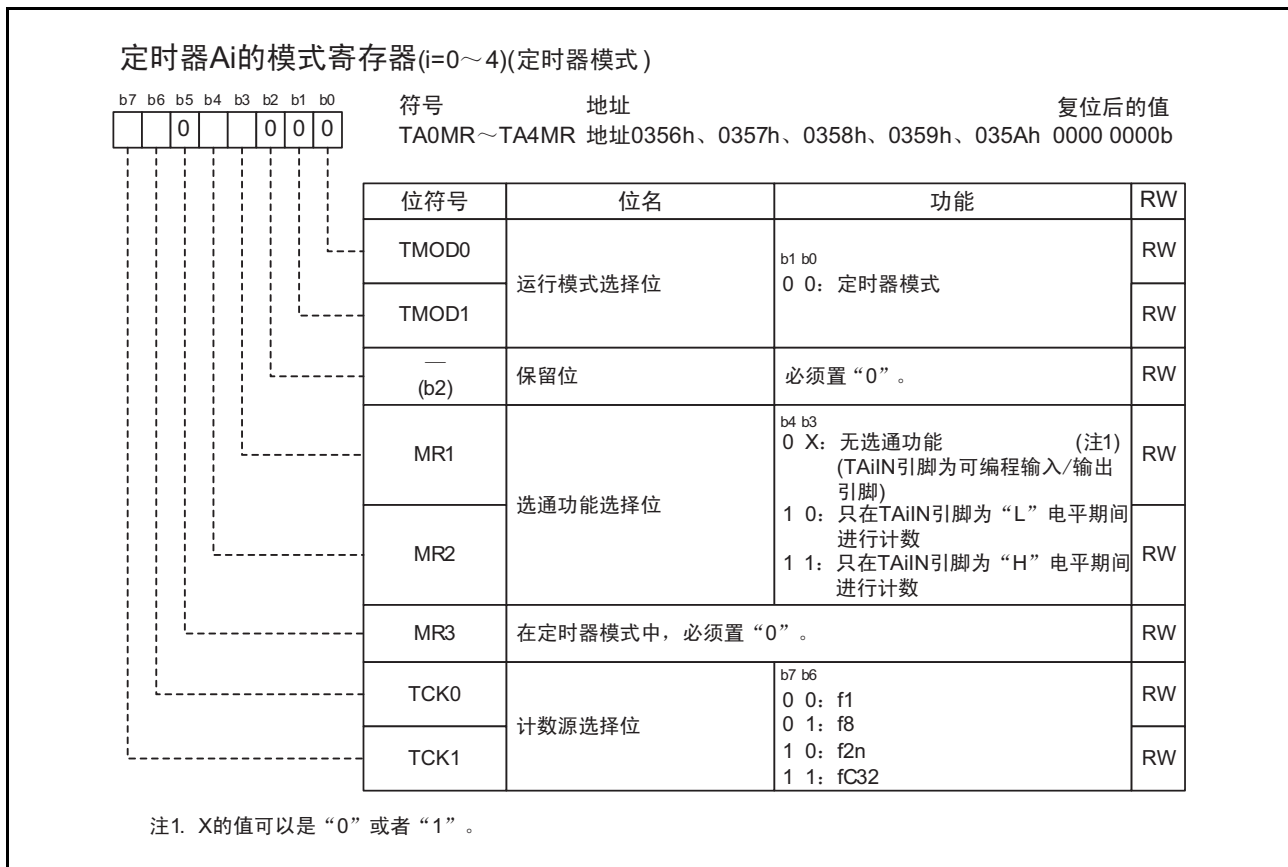


图 16.11 定时器模式的 TA0MR ~ TA4MR 寄存器

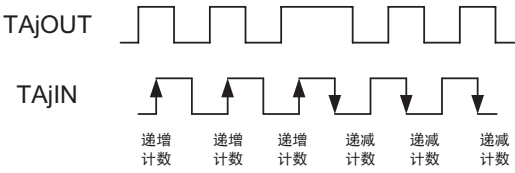
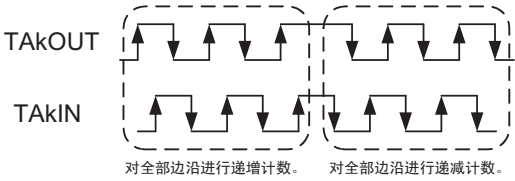
### 16.1.2 事件计数器模式

这是对外部信号或者其他定时器的上溢 / 下溢进行计数的模式。定时器 A2,A3,A4 能对二相外部信号进行计数。事件计数器模式的规格（不使用二相脉冲信号处理的情况）如表 16.2 所示，事件计数器模式的规格（定时器 A2,A3,A4 使用二相脉冲信号处理的情况）如表 16.3 所示，事件计数器模式的 TA0MR ~ TA4MR 寄存器如图 16.12 所示。

表 16.2 事件计数器模式的规格（不使用二相脉冲信号处理的情况）（i=0 ~ 4）

项目	规格
计数源	<ul style="list-style-type: none"> <li>输入到 TAIiN 引脚的外部信号（可通过程序选择有效边沿）</li> <li>定时器 B2 的上溢 / 下溢、定时器 Aj（j=i-1，但是当 i=0 时 j=4）的上溢 / 下溢、定时器 Ak（k=i+1，但是当 i=4 时 k=0）的上溢 / 下溢</li> </ul>
计数	<ul style="list-style-type: none"> <li>可通过外部信号或者程序选择递增计数或者递减计数。</li> <li>在发生上溢或者下溢时，将重加载寄存器的内容进行重新加载后继续计数。在选择自由运行功能时，不重新加载而继续计数。</li> </ul>
分频比	<ul style="list-style-type: none"> <li>在进行递增计数时，为 <math>\frac{1}{FFFFh - n + 1}</math></li> <li>在进行递减计数时，为 <math>\frac{1}{n + 1}</math></li> </ul> n: TAI 寄存器的设定值（0000h ~ FFFFh）
计数开始条件	将 TABSR 寄存器的 TAI S 位置“1”（开始计数）。
计数停止条件	将 TABSR 寄存器的 TAI S 位置“0”（停止计数）。
中断请求的发生时序	在发生上溢或者下溢时。
TAiIN 引脚功能	可编程输入 / 输出端口或者计数源输入引脚
TAiOUT 引脚功能	可编程输入 / 输出端口、脉冲输出引脚或者递增 / 递减计数的转换输入引脚
读定时器	如果读 TAI 寄存器，就能读到计数值。
写定时器	<ul style="list-style-type: none"> <li>在停止计数时或者在开始计数后到输入第 1 个计数源前，如果写 TAI 寄存器，数据就被写到重加载寄存器和计数器。</li> <li>如果在计数时写 TAI 寄存器，数据就被写到重加载寄存器（在下次重新加载时进行传送）。</li> </ul>
其他功能	<ul style="list-style-type: none"> <li>自由运行计数功能 即使发生上溢或者下溢，也不从重加载寄存器进行重新加载。</li> <li>脉冲输出功能 在每次上溢或者下溢时，反转 TAIOUT 引脚的极性。 在 TAI S 位为“0”（停止计数）期间，输出“L”电平。</li> </ul>

表 16.3 事件计数器模式的规格（定时器 A2 ~ A4 使用二相脉冲信号处理的情况）（i=2 ~ 4）

项目	规格
计数源	输入到 TAIiN 或者 TAIiOUT 引脚的二相脉冲信号
计数	<ul style="list-style-type: none"> <li>可通过二相脉冲信号进行递增计数和递减计数的转换。</li> <li>在发生上溢或者下溢时，将重加载寄存器的内容进行重新加载后继续计数。在选择自由运行功能时，不重新加载而继续计数。</li> </ul>
分频比	<ul style="list-style-type: none"> <li>在进行递增计数时，为 <math>\frac{1}{FFFFh - n + 1}</math></li> <li>在进行递减计数时，为 <math>\frac{1}{n + 1}</math></li> </ul> n: TAI 寄存器的设定值（0000h ~ FFFFh）
计数开始条件	将 TABSR 寄存器的 TAI S 位置“1”（开始计数）。
计数停止条件	将 TABSR 寄存器的 TAI S 位置“0”（停止计数）。
中断请求的发生时序	在发生上溢或者下溢时。
TAiIN 引脚功能	二相脉冲输入引脚
TAiOUT 引脚功能	二相脉冲输入引脚
读定时器	如果读 TAI 寄存器，就能读到计数值。
写定时器	<ul style="list-style-type: none"> <li>在停止计数时或者在开始计数后到输入第 1 个计数源前，如果写 TAI 寄存器，数据就被写到重加载寄存器和计数器。</li> <li>如果在计数时写 TAI 寄存器，数据就被写到重加载寄存器（在下次重新加载时进行传送）。</li> </ul>
其他功能（注 1）	<ul style="list-style-type: none"> <li>正常处理（定时器 A2 和定时器 A3） 在 TAJiOUT 引脚（j=2,3）的输入信号为“H”电平期间，对 TAJiN 引脚的上升沿和下降沿分别进行递增计数和递减计数。   </li> <li>4 倍频处理（定时器 A3 和定时器 A4） 在 TAKiOUT 引脚（k=3,4）的输入信号为“H”电平期间，当 TAKiN 引脚为上升相位关系时，对 TAKiOUT 引脚和 TAKiN 引脚的上升沿和下降沿进行递增计数；当 TAKiN 引脚为下降相位关系时，对 TAKiOUT 引脚和 TAKiN 引脚的上升沿和下降沿进行递减计数。   </li> <li>通过 Z 相输入对计数器进行初始化（定时器 A3） 通过 Z 相输入将定时器的计数值置“0”。</li> </ul>

注 1. 只有定时器 A3 能选择任意的功能。定时器 A2 固定为正常处理，定时器 A4 固定为 4 倍频处理。

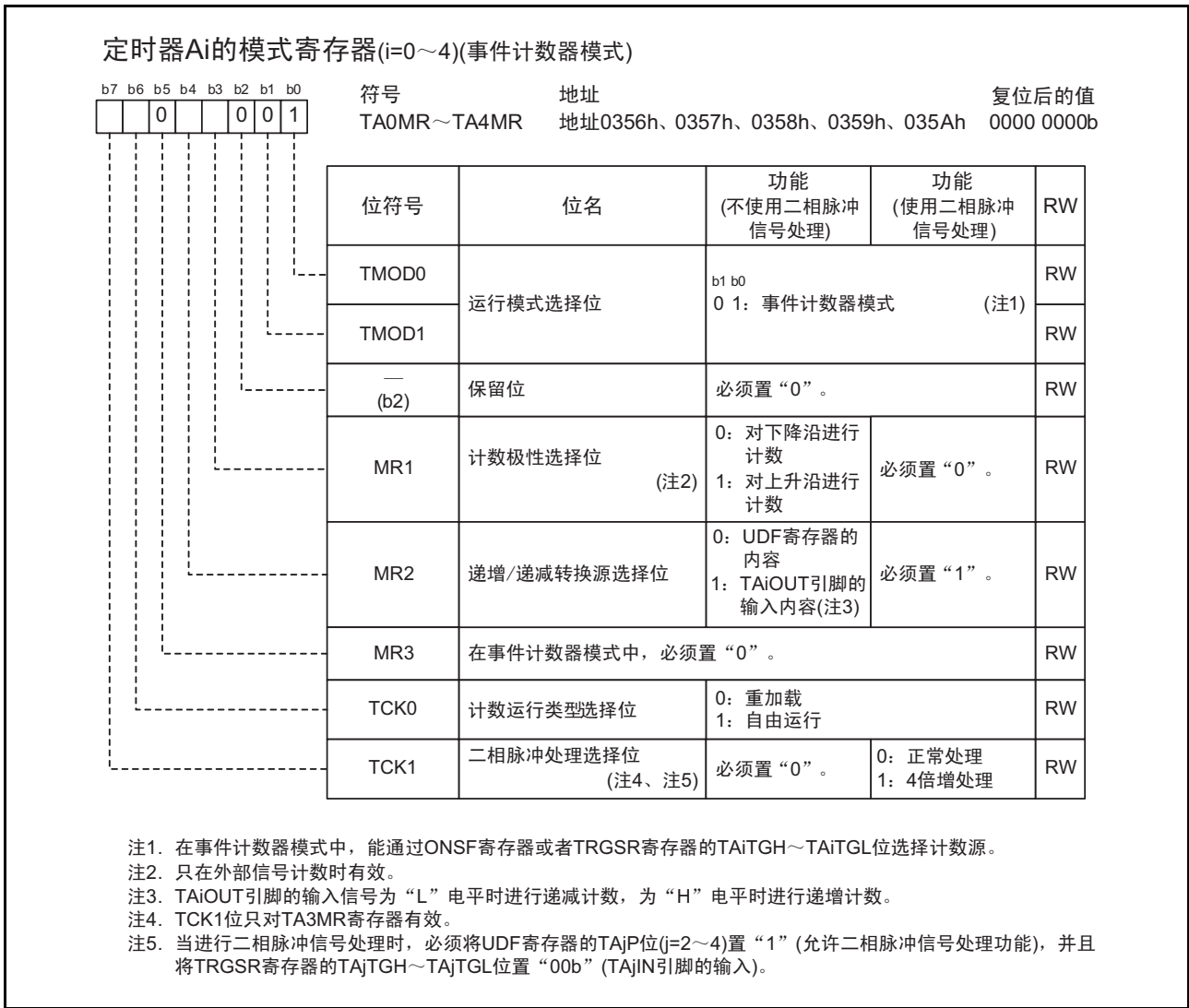


图 16.12 事件计数器模式的 TA0MR ~ TA4MR 寄存器

16.1.2.1 二相脉冲信号处理中的计数器初始化

这是在进行二相脉冲信号处理时，通过 Z 相（计数器的初始化）输入信号将定时器的计数值置“0”的功能。

此功能只能用于定时器 A3 的事件计数器模式、二相脉冲信号处理、自由运行和 4 倍频处理，从  $\overline{\text{INT2}}$  引脚输入 Z 相。

如果将 ONSF 寄存器的 TAZIE 位置“1”（Z 相输入有效），通过 Z 相输入进行的计数器初始化就有效。要通过 Z 相输入将计数器置“0”时，必须预先给 TA3 寄存器写“0000h”。

在检测  $\overline{\text{INT2}}$  的输入边沿后进行 Z 相输入。能通过 INT2IC 寄存器的 POL 位选择边沿的极性。输入的 Z 相脉宽必须至少为定时器 A3 的 1 个计数源周期。二相脉冲（A 相和 B 相）和 Z 相的关系如图 16.13 所示。

在接受到 Z 相输入的下一个计数源时，通过 Z 相输入对计数器进行初始化。计数器的初始化时序如图 16.14 所示。

如果定时器 A3 的上溢/下溢与通过  $\overline{\text{INT2}}$  输入进行的计数器初始化同时发生，就会连续产生 2 次定时器 A3 的中断请求，因此在使用此功能时，不能使用定时器 A3 的中断请求。

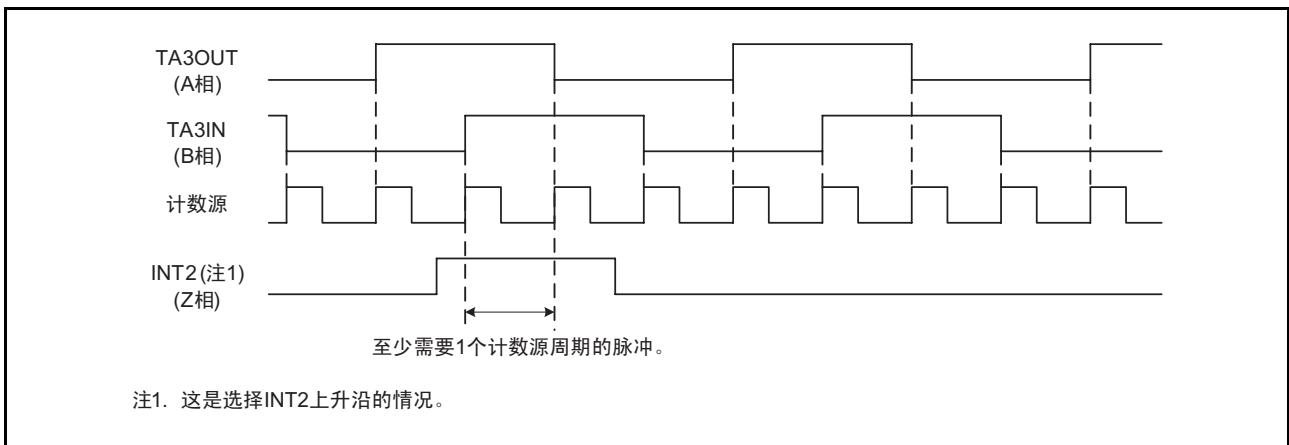


图 16.13 二相脉冲（A 相和 B 相）和 Z 相的关系

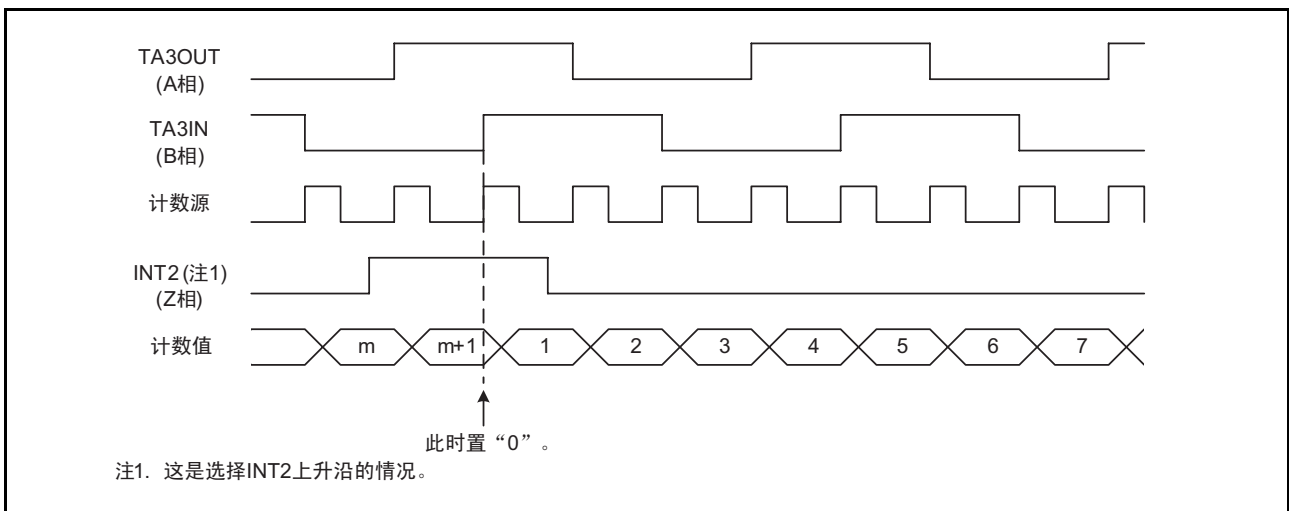


图 16.14 计数器的初始化时序

### 16.1.3 单触发定时器模式

这是对 1 次触发只运行 1 次定时器的模式。单触发定时器模式的规格如表 16.4 所示。如果发生触发，定时器就从发生触发时的任意期间开始运行。单触发定时器模式的 TA0MR ~ TA4MR 寄存器如图 16.15 所示。

表 16.4 单触发定时器模式的规格 (i=0 ~ 4)

项目	规格
计数源	f1、f8、f2n、fC32
计数	<ul style="list-style-type: none"> <li>• 递减计数</li> <li>• 在计数器值变为“0000h”时进行重新加载，然后停止计数。</li> <li>• 如果在计数时发生触发，就在进行重新加载后继续计数。</li> </ul>
分频比	$\frac{1}{n}$ n: TAI寄存器的设定值 (0000h ~ FFFFh) (当设定值为“0000h”时，计数器不进行计数。)
计数开始条件	TABSR 寄存器的 TAI <sub>S</sub> 位为“1” (开始计数) 并发生以下触发: <ul style="list-style-type: none"> <li>• 从 TAI<sub>I</sub>N 引脚输入外部触发。</li> <li>• 定时器 B2 的上溢或者下溢、定时器 A<sub>j</sub> (j=i-1, 但是当 i=0 时 j=4) 的上溢或者下溢、定时器 A<sub>k</sub> (k=i+1, 但是当 i=4 时 k=0) 的上溢或者下溢。</li> <li>• 将 ONSF 寄存器的 TAI<sub>O</sub>S 位置“1” (定时器开始计数)。</li> </ul>
计数停止条件	<ul style="list-style-type: none"> <li>• 在计数值变为“0000h”并进行重新加载后。</li> <li>• 将 TABSR 寄存器的 TAI<sub>S</sub> 位置“0” (停止计数)。</li> </ul>
中断请求的发生时序	在计数值变为“0000h”时。
TAI <sub>I</sub> N 引脚功能	可编程输入 / 输出端口或者触发输入引脚
TAI <sub>O</sub> UT 引脚功能	可编程输入 / 输出端口或者脉冲输出引脚
读定时器	如果读 TAI 寄存器，读取值就为不定值。
写定时器	<ul style="list-style-type: none"> <li>• 在计数停止时或者在开始计数后到输入第 1 个计数源前，如果写 TAI 寄存器，数据就被写到重加载寄存器和计数器。</li> <li>• 如果在计数时写 TAI 寄存器，数据就被写到重加载寄存器 (在下次重新加载时进行传送)。</li> </ul>
其他功能	<ul style="list-style-type: none"> <li>• 脉冲输出功能 在停止计数时，输出“L”电平；在计数时，输出“H”电平。</li> </ul>

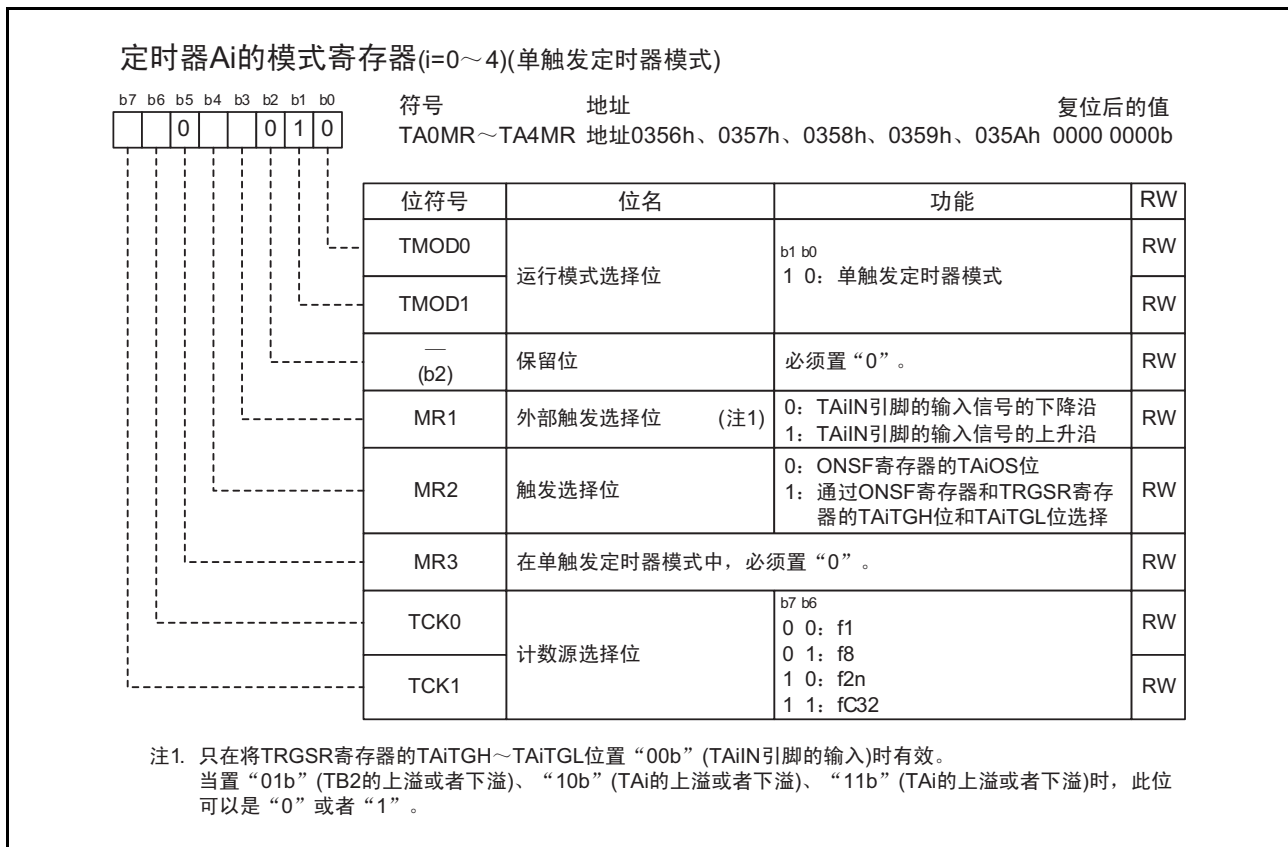


图 16.15 单触发定时器模式的 TA0MR ~ TA4MR 寄存器

### 16.1.4 脉宽调制模式

这是连续输出任意宽度的脉冲的模式，脉宽调制模式的规格如表 16.5 所示。在此模式中，计数器用作 16 位脉宽调制器或者 8 位脉宽调制器。脉宽调制模式的 TA0MR ~ TA4MR 寄存器如图 16.16 所示，16 位脉宽调制器和 8 位脉宽调制器的运行例子分别如图 16.17 和图 16.18 所示。

表 16.5 脉宽调制模式的规格 (i=0 ~ 4)

项目	规格
计数源	f1、f8、f2n、fC32
计数	<ul style="list-style-type: none"> <li>• 递减计数（用作 8 位脉宽调制器或者 16 位脉宽调制器）</li> <li>• 在 PWM 脉冲的上升沿进行重新加载后继续计数。</li> <li>• 计数时发生的触发不影响计数。</li> </ul>
16 位 PWM	<ul style="list-style-type: none"> <li>• “H” 电平宽度：<math>\frac{n}{fj}</math> n: TAI 寄存器的设定值 (0000h ~ FFFEh) fj: 计数源的频率</li> <li>• 周期：固定为 <math>\frac{2^{16}-1}{fj}</math></li> </ul>
8 位 PWM	<ul style="list-style-type: none"> <li>• “H” 电平宽度：<math>\frac{n \times (m+1)}{fj}</math> n: TAI 寄存器高位地址的设定值 (00h ~ FEh)</li> <li>• 周期：<math>\frac{(2^8-1) \times (m+1)}{fj}</math> m: TAI 寄存器低位地址的设定值 (00h ~ FFh)</li> </ul>
计数开始条件	可选择以下任意一个： <ul style="list-style-type: none"> <li>• 将 TABSR 寄存器的 TAI<sub>S</sub> 位置 “1”（开始计数）。</li> <li>• TAI<sub>S</sub> 位为 “1” 并且从 TAI<sub>I</sub>N 引脚输入外部触发。</li> <li>• TAI<sub>S</sub> 位为 “1” 并发生以下触发： 定时器 B2 的上溢或者下溢、定时器 A<sub>j</sub> (j=i-1, 但是当 i=0 时 j=4) 的上溢或者下溢、 定时器 A<sub>k</sub> (k=i+1, 但是当 i=4 时 k=0) 的上溢或者下溢</li> </ul>
计数停止条件	将 TABSR 寄存器的 TAI <sub>S</sub> 位置 “0”（停止计数）。
中断请求的发生时序	在 PWM 脉冲下降时。
TAI <sub>I</sub> N 引脚功能	可编程输入 / 输出端口或者触发输入引脚
TAI <sub>O</sub> UT 引脚功能	脉冲输出引脚
读定时器	如果读 TAI 寄存器，读取值就为不定值。
写定时器	<ul style="list-style-type: none"> <li>• 在停止计数时或者在开始计数后到输入第 1 个计数源前，如果写 TAI 寄存器，数据就被写到重加载寄存器和计数器。</li> <li>• 如果在计数时写 TAI 寄存器，数据就被写到重加载寄存器（在下次重新加载时进行传送）。</li> </ul>



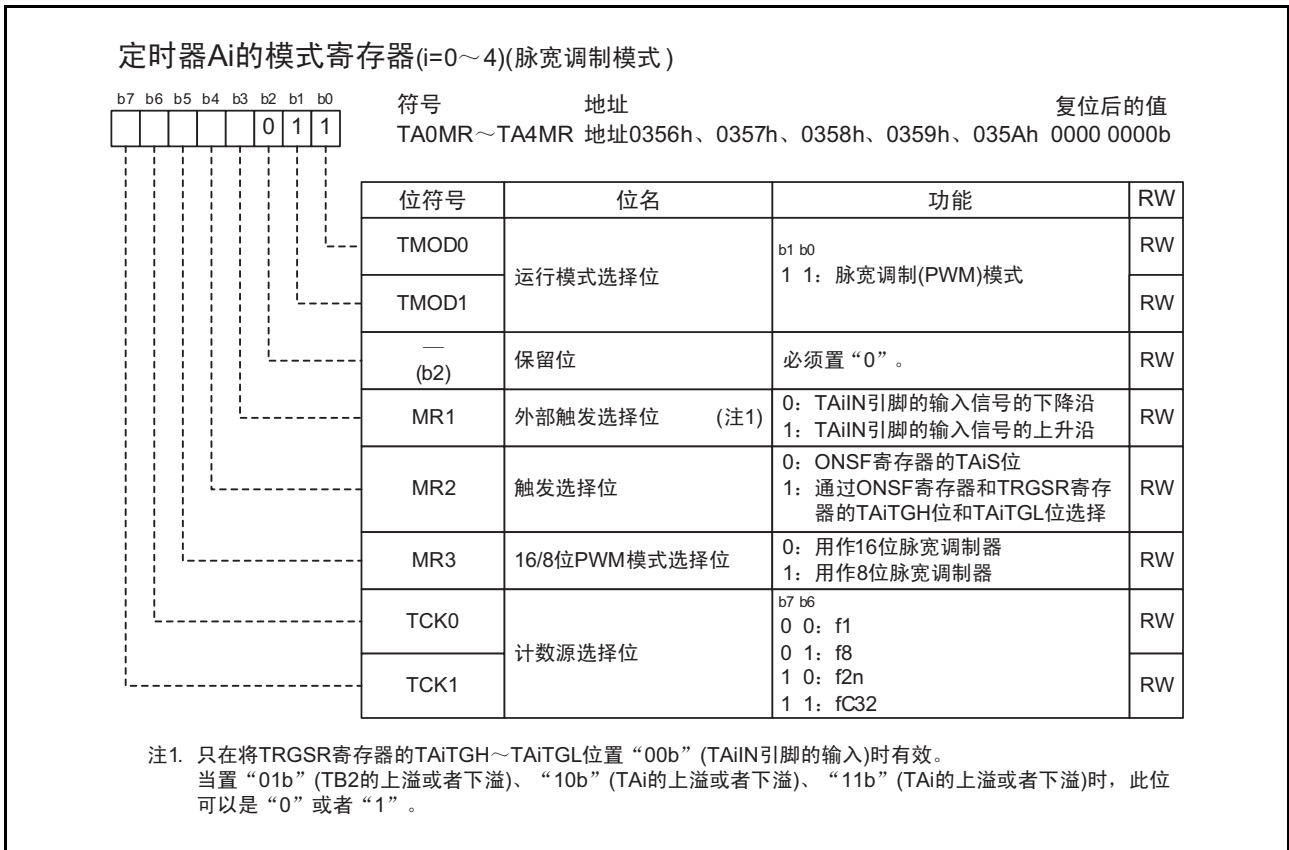


图 16.16 脉宽调制模式的 TA0MR ~ TA4MR 寄存器

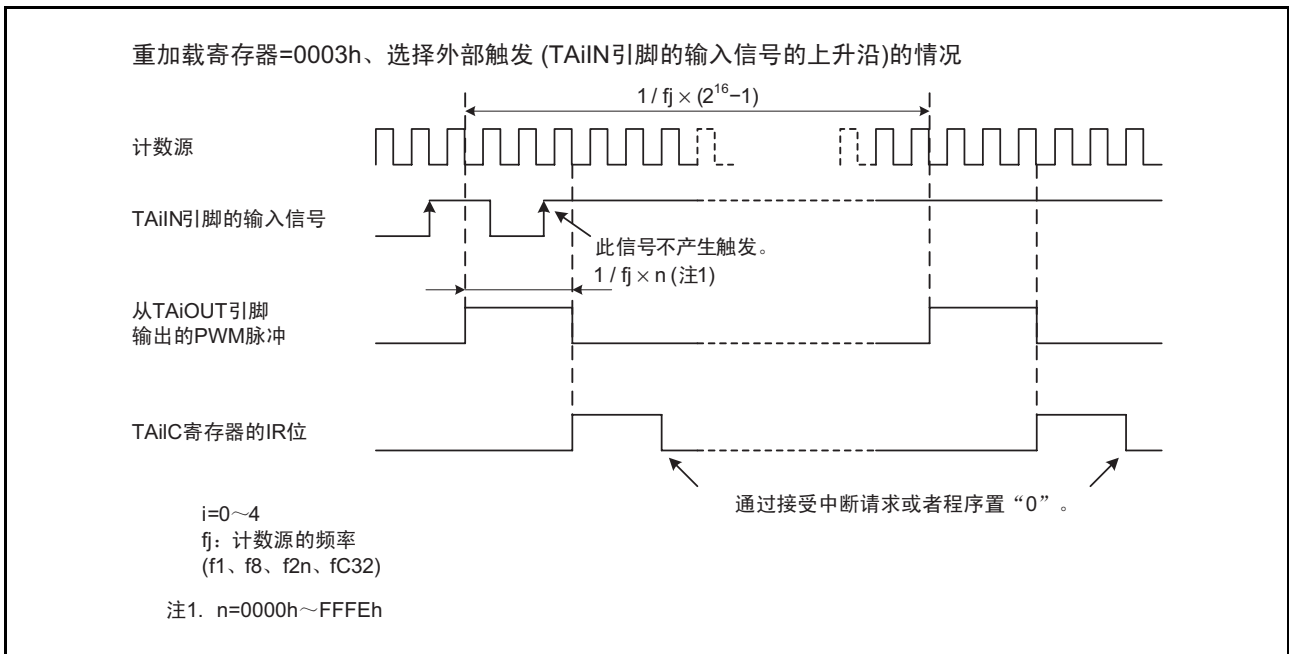


图 16.17 16 位脉宽调制器的运行例子

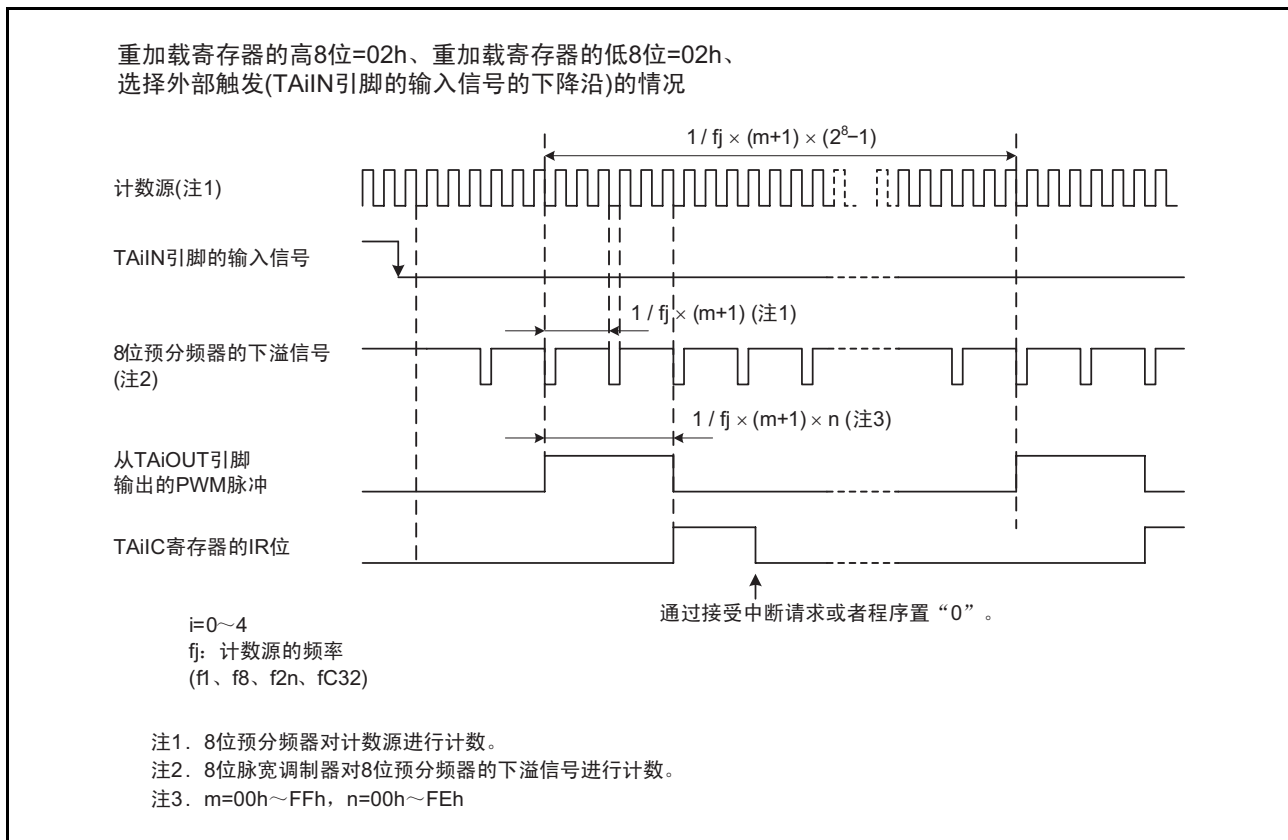


图 16.18 8 位脉宽调制器的运行例子

### 16.2 定时器 B

定时器 B 的框图及其相关寄存器分别如图 16.19 和图 16.20 ~ 图 16.23 所示。

定时器 B 有以下 3 种模式，能通过 TB0MR ~ TB5MR 寄存器的 TMOD1 ~ TMOD0 位选择各模式。

- 定时器模式 这是对内部计数源进行计数的模式。
- 事件计数器模式 这是对外部脉冲或者其他定时器的上溢/下溢进行计数的模式。
- 脉冲周期测量模式和脉宽测量模式 这是测量外部脉冲周期或者脉宽的模式。

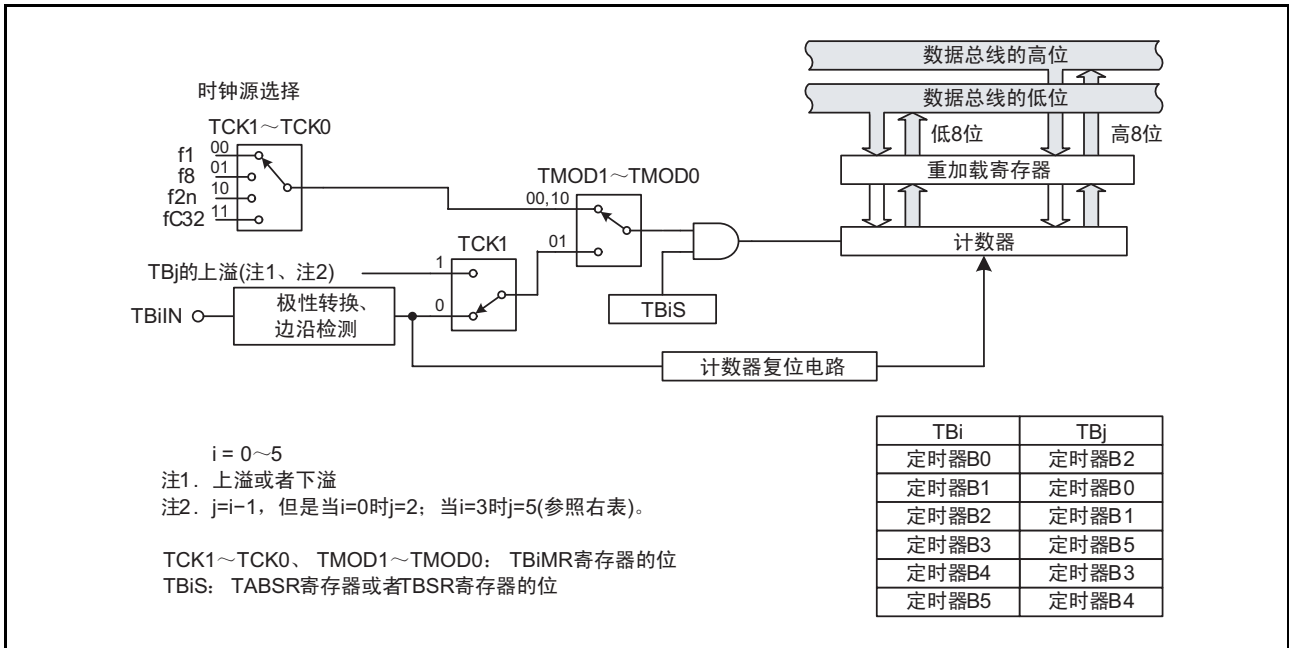


图 16.19 定时器 B 的框图

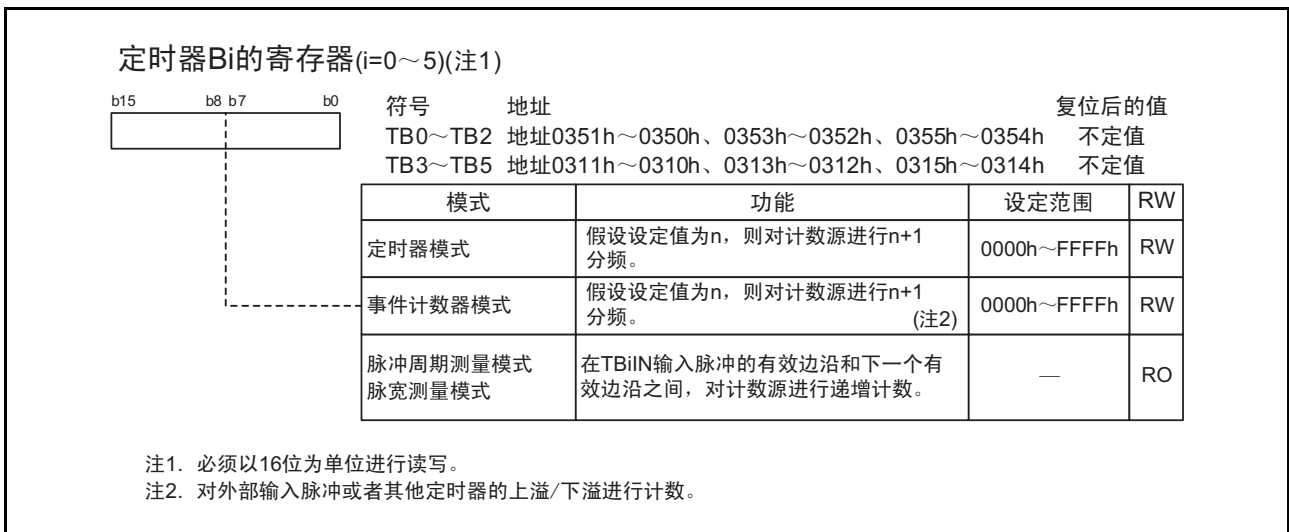


图 16.20 TB0 ~ TB5 寄存器

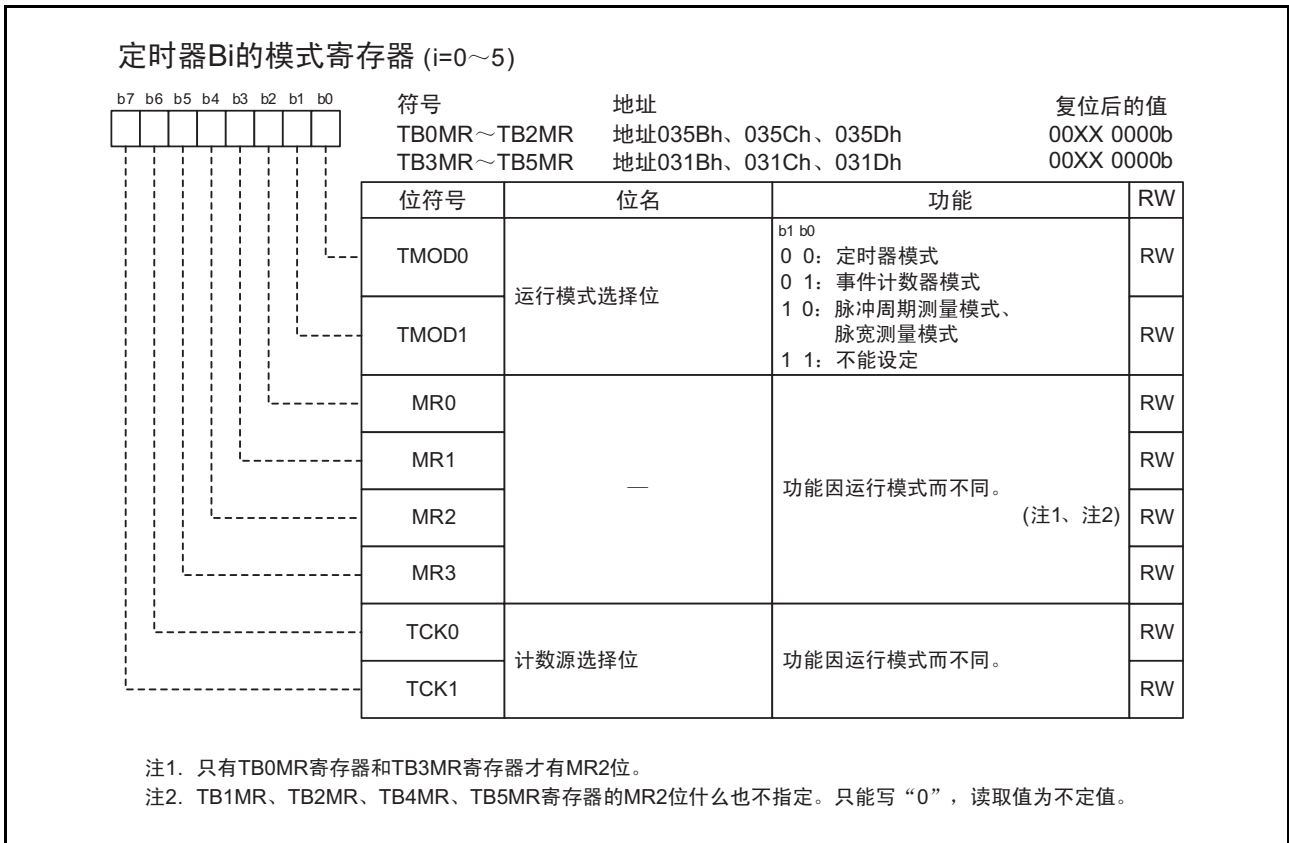


图 16.21 TB0MR ~ TB5MR 寄存器

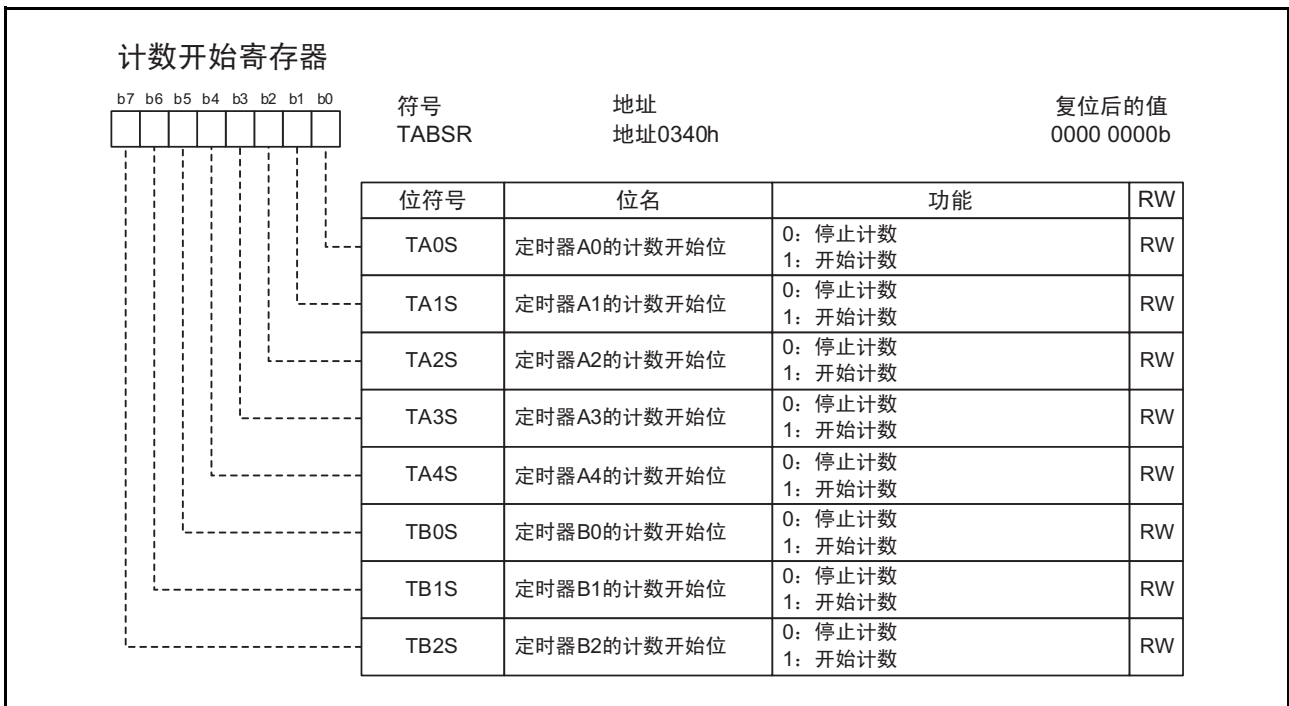


图 16.22 TABSR 寄存器

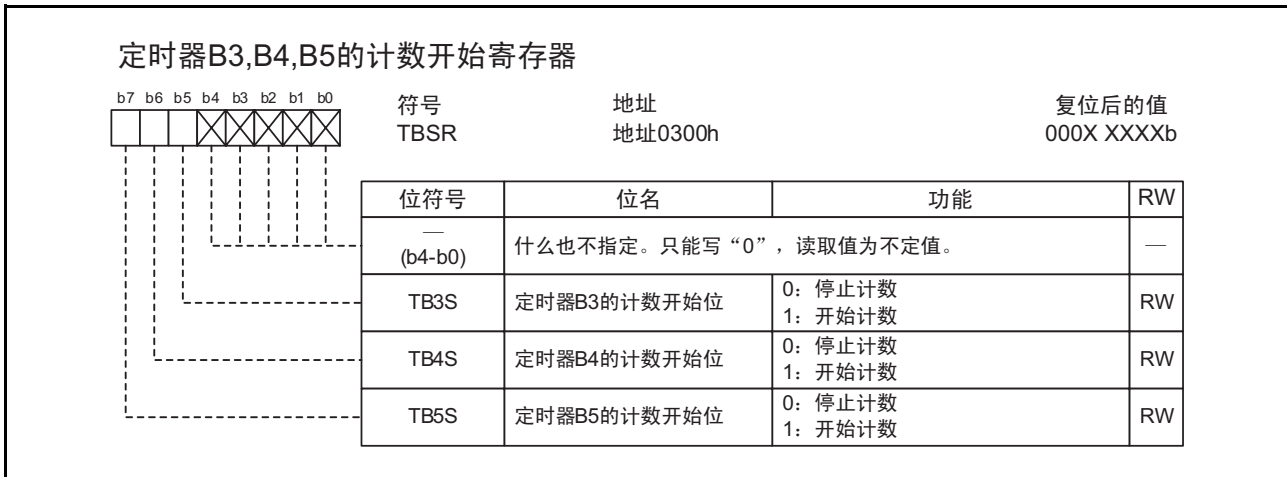


图 16.23 TBSR 寄存器

### 16.2.1 定时器模式

这是对内部生成的计数源进行计数的模式。定时器模式的规格及其 TB0MR ~ TB5MR 寄存器分别如表 16.6 和图 16.24 所示。

表 16.6 定时器模式的规格 (i=0 ~ 5)

项目	规格
计数源	f1、f8、f2n、fC32
计数	递减计数 在发生下溢时，将重加载寄存器的内容进行重新加载后继续计数。
分频比	$\frac{1}{n+1}$ n: TBi 寄存器的设定值 (0000h ~ FFFFh)
计数开始条件	将 TABSR 寄存器、TBSR 寄存器的 TBiS 位置“1” (开始计数)。
计数停止条件	将 TABSR 寄存器、TBSR 寄存器的 TBiS 位置“0” (停止计数)。
中断请求的发生时序	在发生下溢时。
TBiIN 引脚功能	可编程输入 / 输出端口
读定时器	如果读 TBi 寄存器，就能读到计数值。
写定时器	<ul style="list-style-type: none"> <li>在停止计数时或者在开始计数后到输入第 1 个计数源前，如果写 TBi 寄存器，数据就被写到重加载寄存器和计数器。</li> <li>如果在计数时写 TBi 寄存器，数据就被写到重加载寄存器 (在下次重新加载时进行传送)。</li> </ul>

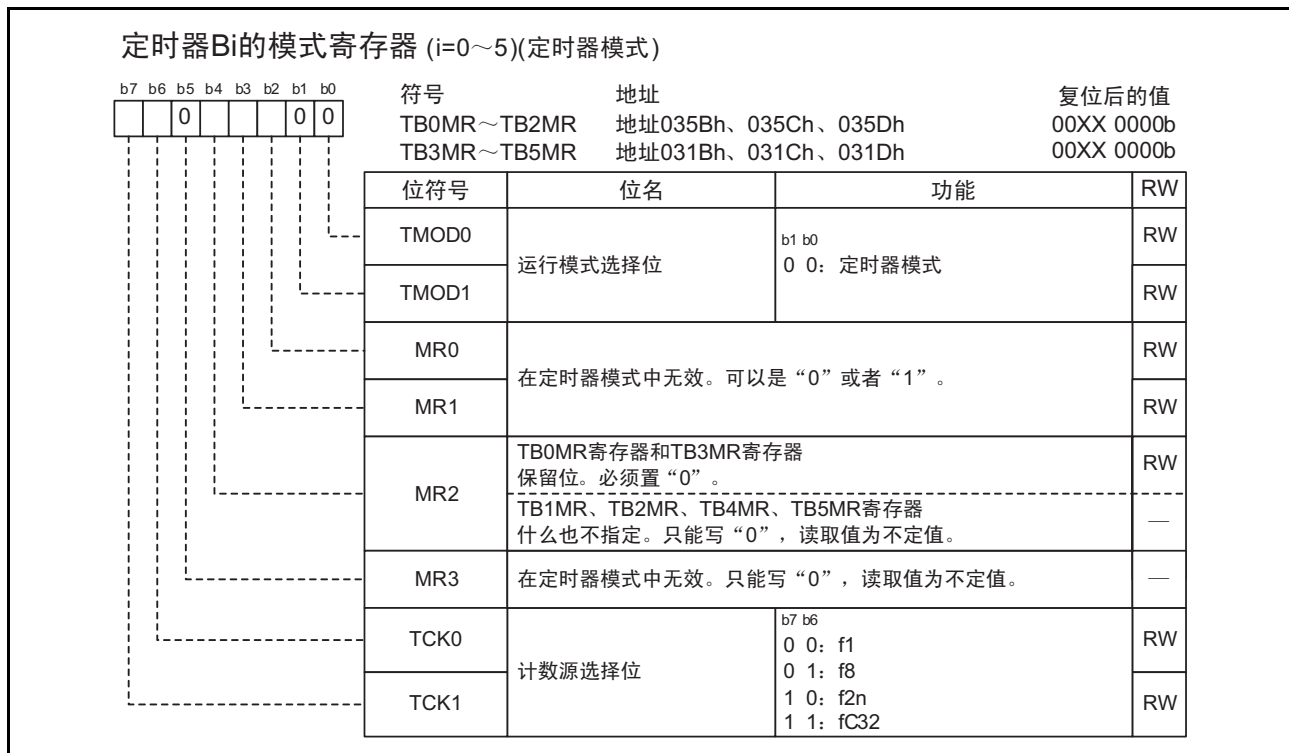


图 16.24 定时器模式的 TB0MR ~ TB5MR 寄存器

### 16.2.2 事件计数器模式

这是对外部信号或者其他定时器的上溢/下溢进行计数的模式。事件计数器模式的规格及其 TB0MR ~ TB5MR 寄存器分别如表 16.7 和图 16.25 所示。

表 16.7 事件计数器模式的规格 (i=0 ~ 5)

项目	规格
计数源	<ul style="list-style-type: none"> <li>输入到 TBiIN 引脚的外部信号 可通过程序选择上升沿、下降沿或者双边沿作为计数源的有效边沿。</li> <li>TBj 的上溢 / 下溢 (j=i-1, 但是当 i=0 时 j=2; 当 i=3 时 j=5)</li> </ul>
计数	递减计数 在发生下溢时, 将重加载寄存器的内容进行重新加载后继续计数。
分频比	$\frac{1}{n+1}$ n: TBi 寄存器的设定值 (0000h ~ FFFFh)
计数开始条件	将 TABSR 寄存器、TBSR 寄存器的 TBiS 位置“1”(开始计数)。
计数停止条件	将 TABSR 寄存器、TBSR 寄存器的 TBiS 位置“0”(停止计数)。
中断请求的发生时序	在发生下溢时。
TBiIN 引脚功能	可编程输入 / 输出端口或者计数源输入引脚
读定时器	如果读 TBi 寄存器, 就能读到计数值。
写定时器	<ul style="list-style-type: none"> <li>在停止计数时或者在开始计数后到输入第 1 个计数源前, 如果写 TBi 寄存器, 数据就被写到重加载寄存器和计数器。</li> <li>如果在计数时写 TBi 寄存器, 数据就被写到重加载寄存器 (在下次重新加载时进行传送)。</li> </ul>

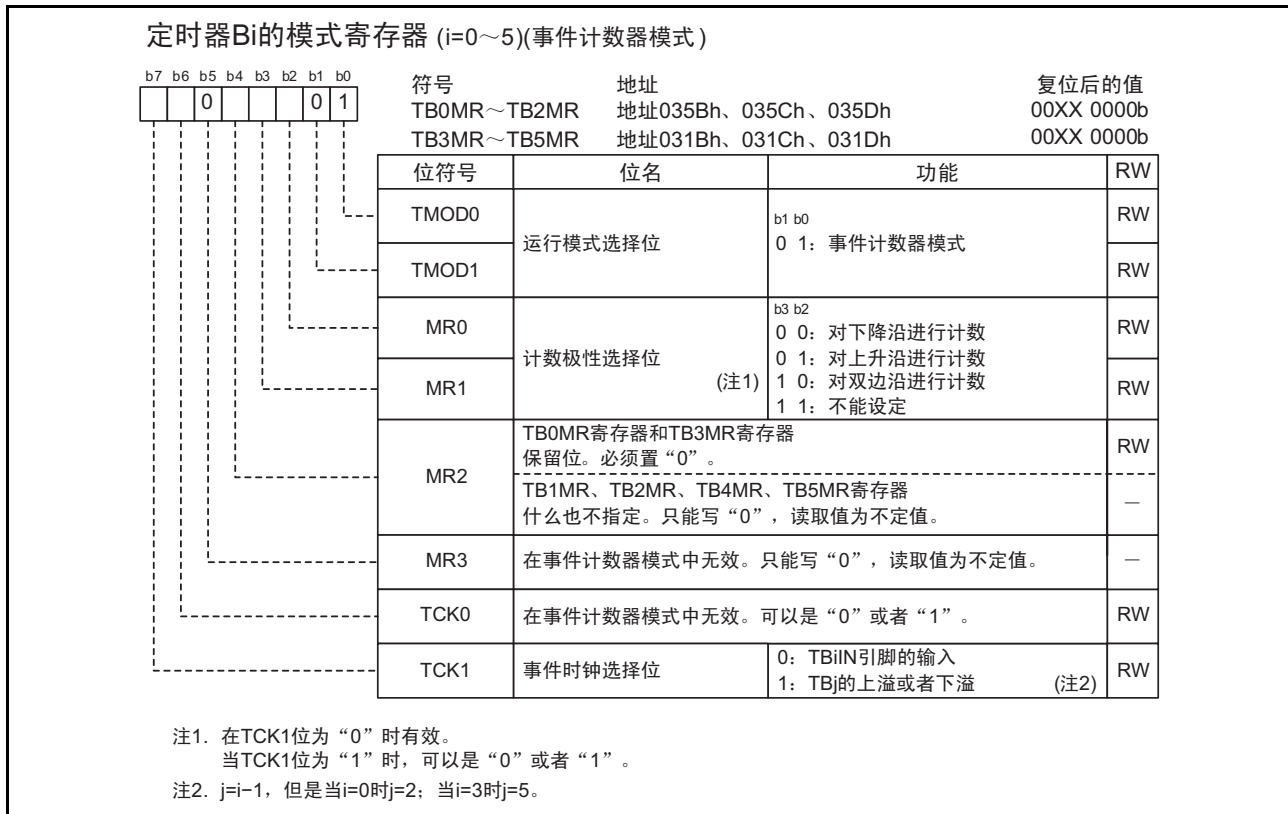


图 16.25 事件计数器模式的 TB0MR ~ TB5MR 寄存器

### 16.2.3 脉冲周期测量模式和脉宽测量模式

这是测量外部信号的脉冲周期或者脉宽的模式。脉冲周期测量模式和脉宽测量模式的规格及其 TB0MR ~ TB5MR 寄存器分别如表 16.8 和图 16.26 所示，脉冲周期测量模式和脉宽测量模式的运行例子分别如图 16.27 和图 16.28 所示。

表 16.8 脉冲周期测量模式和脉宽测量模式的规格 (i=0 ~ 5)

项目	规格
计数源	f1、f8、f2n、fC32
计数	递增计数 在被测量脉冲的有效边沿将计数器的值传送到重加载寄存器，并将计数器的值置“0000h”，然后继续计数。
计数开始条件	将 TABSR 寄存器、TBSR 寄存器的 TBIS 位置“1”（开始计数）。
计数停止条件	将 TABSR 寄存器、TBSR 寄存器的 TBIS 位置“0”（停止计数）。
中断请求的发生时序	<ul style="list-style-type: none"> <li>在输入被测量脉冲的有效边沿时（注1）。</li> <li>在发生上溢时（同时 TBiMR 寄存器的 MR3 位变为“1”（有上溢）（注2））。</li> </ul>
TBiIN 引脚功能	被测量脉冲的输入引脚
读定时器	如果读 TBi 寄存器，就能读到重加载寄存器的内容（测量结果）（注3）。
写定时器	不能将 TBi 寄存器的值写到重加载寄存器和计数器。

注1. 在开始计数后第1次输入有效边沿时，不发生中断请求。

注2. 当 TBIS 位为“1”（开始计数）时，如果在 MR3 位变为“1”（有上溢）并至少经过1个计数源时钟后，写 TBiMR 寄存器，MR3 位就变为“0”（无上溢）。

注3. 在开始计数后到第2次输入有效边沿前，TBi 寄存器的读取值为不定值。

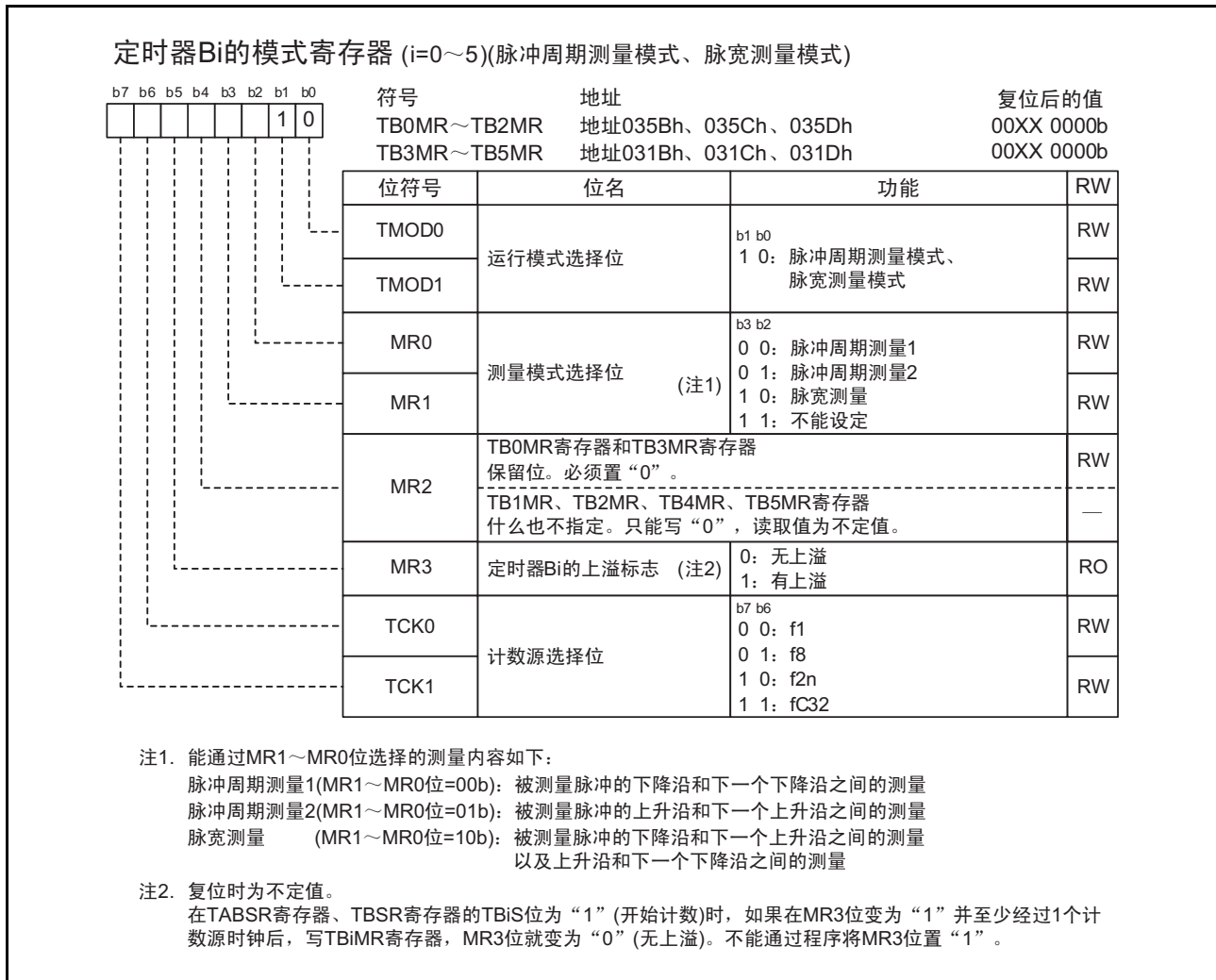


图 16.26 脉冲周期测量模式和脉宽测量模式的 TB0MR ~ TB5MR 寄存器



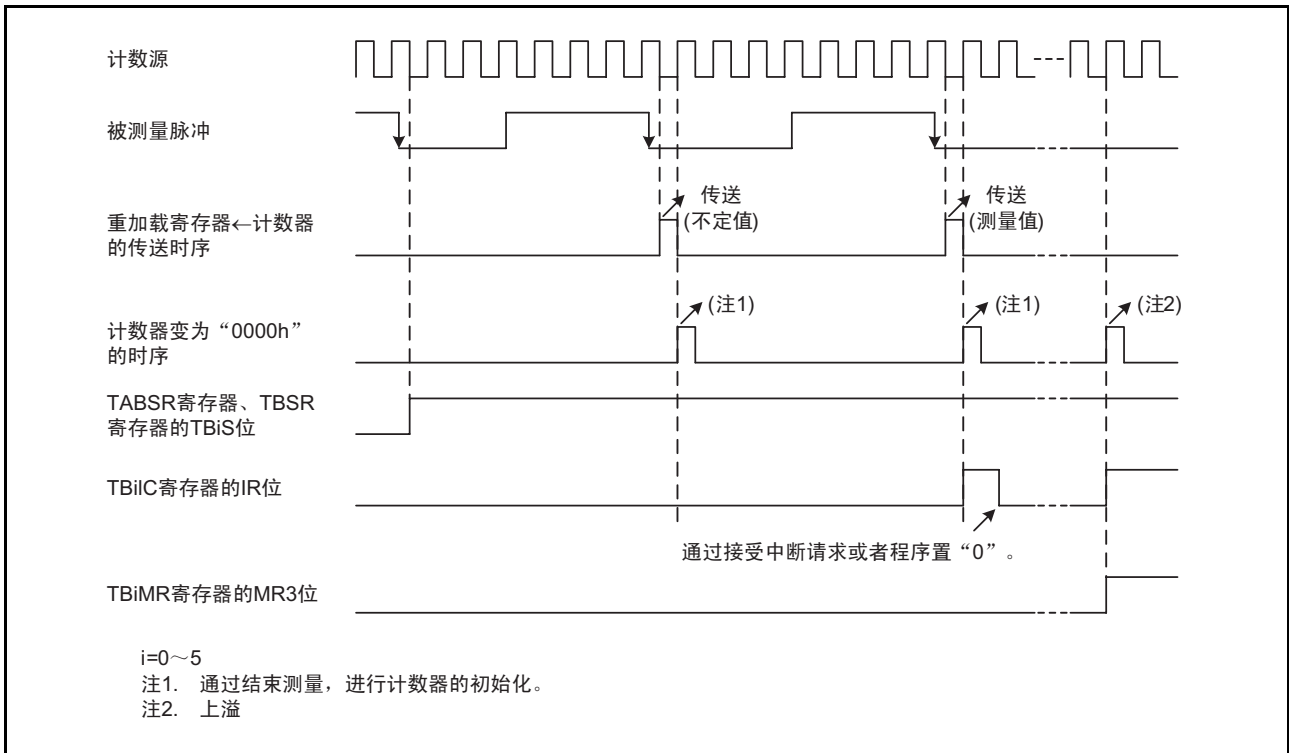


图 16.27 脉冲周期测量模式的运行图

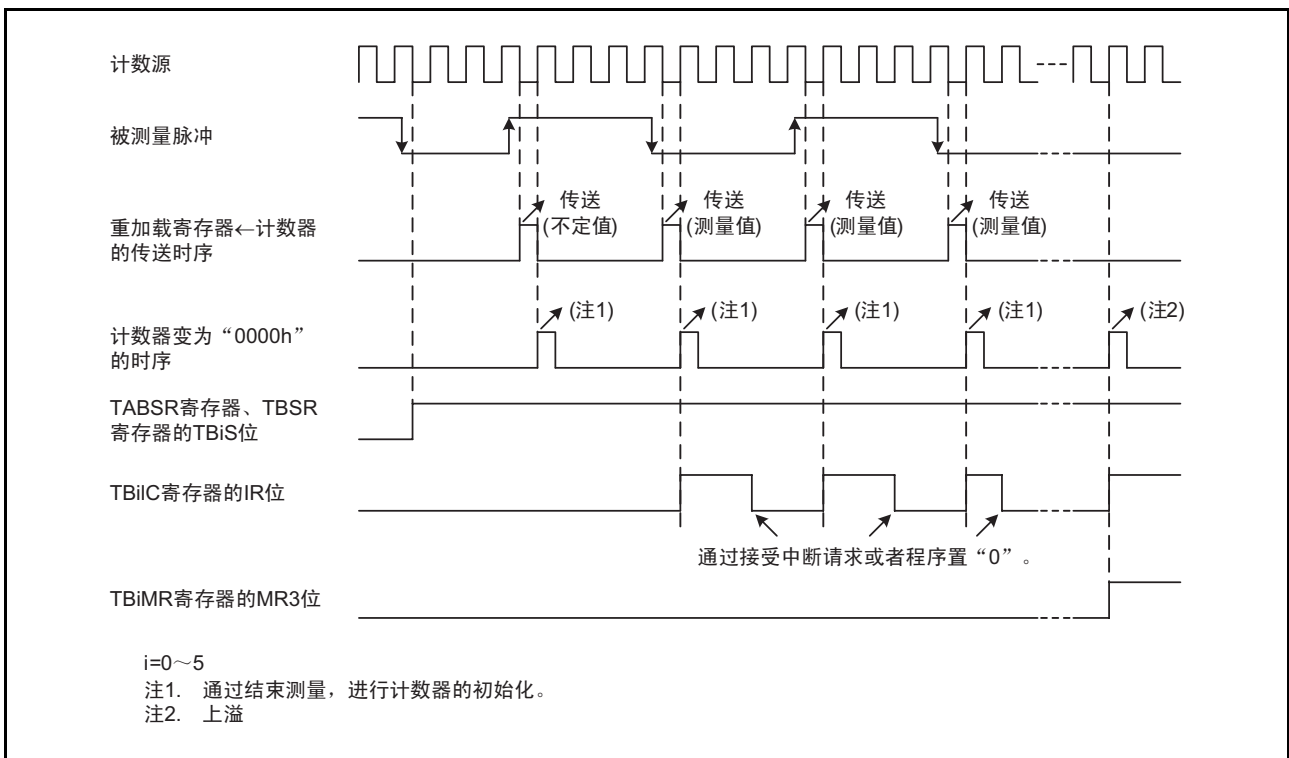


图 16.28 脉宽测量模式的运行图

## 16.3 使用定时器时的注意事项

### 16.3.1 定时器 A 和定时器 B 的共同事项

复位后，定时器停止运行。在设定模式、计数源和计数器的值后，必须将 TABSR 寄存器或者 TBSR 寄存器的 TAI<sub>S</sub> 位 (i=0 ~ 4) 或者 TB<sub>j</sub>S 位 (j=0 ~ 5) 置 “1” (开始计数)。

必须在对应的 TAI<sub>S</sub> 位或者 TB<sub>j</sub>S 位为 “0” (停止计数) 的状态下，更改以下的寄存器和位：

- TAI<sub>MR</sub> 寄存器和 TB<sub>j</sub>MR 寄存器
- UDF 寄存器
- ONSF 寄存器的 TAZIE 位、TA0TGL 位和 TA0TGH 位
- TRGSR 寄存器

### 16.3.2 定时器 A

#### 16.3.2.1 定时器模式的情况

- 通过读 TAI 寄存器，能随时知道正在计数的计数器的值。如果在进行重新加载时读 TAI 寄存器，就能读到 “FFFFh”。如果在停止计数时将值设定到 TAI 寄存器，然后在开始计数前读 TAI 寄存器，就能读到所设定的值。

#### 16.3.2.2 事件计数器模式的情况

- 通过读 TAI 寄存器，能随时知道正在计数的计数器的值。如果在进行重新加载时读 TAI 寄存器，就能在发生下溢时读到 “FFFFh”，而在发生上溢时读到 “0000h”。如果在停止计数时将值设定到 TAI 寄存器，然后在计数器开始计数前读 TAI 寄存器，就能读到所设定的值。

#### 16.3.2.3 单触发定时器模式的情况

- 如果在计数时将 TABSR 寄存器的 TAI<sub>S</sub> 位置 “0” (停止计数)，就会出现以下的状态：
  - 计数器停止计数并重新加载 TAI 寄存器的设定值。
  - TAI<sub>OUT</sub> 引脚输出 “L” 电平。
  - 在 1 个 CPU 时钟周期后，TAI<sub>IC</sub> 寄存器的 IR 位变为 “1” (有中断请求)。
- 因为单触发定时器的输出与内部生成的计数源同步，所以在选择 TAI<sub>IN</sub> 引脚的输入作为触发时，从触发输入到单触发定时器的输出期间，会产生最长 1 个计数源时钟的延迟。
- 如果在以下任意一种情况下设定定时器的运行模式，IR 位就变为 “1”。因此，在使用定时器 A<sub>i</sub> 中断 (IR 位) 时，必须在进行下述的设定后将 IR 位置 “0”。
  - 在复位后选择单触发定时器模式时。
  - 在将运行模式从定时器模式更改为单触发定时器模式时。
  - 在将运行模式从事件计数器模式更改为单触发定时器模式时。
- 如果在计数时发生再触发，计数器就先进行 1 次递减计数，接着重新加载 TAI 寄存器 (i=0 ~ 4) 的设定值，然后继续计数。如果要在计数时产生再触发，就必须在从上次触发开始至少经过 1 个定时器计数源时钟后，产生再触发。
- 如果已经选择 TAI<sub>IN</sub> 引脚的触发输入作为计数开始条件，就不能在定时器 A 的计数值变为 “0000h” 前的 300ns 期间输入再触发，否则单触发定时器就可能不继续计数而停止。

#### 16.3.2.4 脉宽调制模式的情况

- 如果在以下任意一种情况下设定定时器的运行模式，IR 位就变为“1”。因此，在使用定时器 Ai 中断（IR 位）时，必须在进行以下的设定后将 IR 位置“0”。
  - 在复位后选择 PWM 模式时。
  - 在将运行模式从定时器模式更改为 PWM 模式时。
  - 在将运行模式从事件计数器模式更改为 PWM 模式时。
- 如果在输出 PWM 脉冲时将 TAI<sub>S</sub> 位置“0”（停止计数），就会出现以下的状态：
  - 计数器停止计数。
  - 在从 TAI<sub>OUT</sub> 引脚输出“H”电平时，输出电平变为“L”并且 IR 位变为“1”。
  - 在从 TAI<sub>OUT</sub> 引脚输出“L”电平时，输出电平不变，IR 位也不变。

### 16.3.3 定时器 B

#### 16.3.3.1 定时器模式和事件计数器模式的情况

- 通过读 TB<sub>j</sub> 寄存器（j=0~5），能随时知道正在计数的计数器的值。如果在进行重新加载时读 TB<sub>j</sub> 寄存器，就能读到“FFFFh”。如果在停止计数时将值设定到 TB<sub>j</sub> 寄存器，然后在开始计数前读 TB<sub>j</sub> 寄存器，就能读到所设定的值。

#### 16.3.3.2 脉冲周期测量 / 脉宽测量模式的情况

- 要将 TB<sub>j</sub>MR 寄存器的 MR3 位置“0”（无上溢）时，必须在 TB<sub>j</sub>S 位为“1”（开始计数）时，在 MR3 位变为“1”（有上溢）并至少经过 1 个计数源时钟后，写 TB<sub>j</sub>MR 寄存器。
- 在只要检测上溢时，必须使用 TB<sub>j</sub>IC 寄存器的 IR 位。只能在通过中断处理程序判断中断源时使用 MR3 位。
- 在开始计数时，计数器的值为不定值。因此，在开始计数后第 1 次输入有效边沿前，计数器有可能发生上溢并且发生定时器 B<sub>j</sub> 的中断请求。
- 在开始计数后第 1 次输入有效边沿时，因为计数器的值为不定值，所以不定值被传送到重加载寄存器。此时，不发生定时器 B<sub>j</sub> 的中断请求。
- 如果在开始计数后更改 TB<sub>j</sub>MR 寄存器的 MR1~MR0 位，IR 位就可能变为“1”（有中断请求）。如果给 MR1~MR0 位重写相同的值，IR 位就不变。
- 在脉宽测量模式中连续测量脉宽。必须通过程序判断测量结果是“H”电平宽度还是“L”电平宽度。
- 在脉冲周期测量模式中，如果在计数器发生上溢的同时输入有效边沿，就只产生 1 次中断请求，所以无法确认是否输入了有效边沿。因此，必须在计数器不发生上溢的范围内使用。
- 在脉宽测量模式中，必须通过定时器 B<sub>j</sub> 的中断处理程序，读端口的电平来判断计数器是发生上溢还是输入了有效边沿。

## 17. 三相马达控制定时器功能

能使用定时器 A1,A2,A4,B2 输出三相马达驱动波形。如果将 INVC0 寄存器的 INV02 位置“1”，三相马达控制定时器功能就有效。通过此功能将定时器 B2 用于载波控制、将定时器 A4,A1,A2 用于三相 PWM 输出（U、 $\bar{U}$ 、V、 $\bar{V}$ 、W、 $\bar{W}$ ）控制。

三相马达控制定时器功能的规格、框图及其相关寄存器分别如表 17.1、图 17.1 和图 17.2 ~ 图 17.6 所示。

表 17.1 三相马达控制定时器功能的规格

项目	规格
三相 PWM 波形的输出引脚	6 个（U、 $\bar{U}$ 、V、 $\bar{V}$ 、W、 $\bar{W}$ ）
强制截止的输入（注 1）	给 $\overline{\text{NMI}}$ 引脚输入“L”电平。
使用的定时器	定时器 A4,A1,A2（用于单触发定时器模式） 定时器 A4：U 相和 $\bar{U}$ 相波形控制 定时器 A1：V 相和 $\bar{V}$ 相波形控制 定时器 A2：W 相和 $\bar{W}$ 相波形控制 定时器 B2（用于定时器模式） 载波周期控制 死区时间定时器（3 个 8 位定时器，共用重加载寄存器） 死区时间控制
输出波形	三角波调制和锯齿波调制 • 能输出 1 个周期全部为“H”或者“L”的电平。 • 能独立设定上侧导通信号和下侧导通信号的输出逻辑。
载波周期	三角波调制：计数源 $\times(m+1)\times 2$ 锯齿波调制：计数源 $\times(m+1)$ m：TB2 寄存器的设定值（0000h ~ FFFFh） 计数源：f1、f8、f2n、fC32
三相 PWM 输出宽度	三角波调制：计数源 $\times n \times 2$ 锯齿波调制：计数源 $\times n$ n：TA4、TA1、TA2（在 INVC1 寄存器的 INV11 位为“1”时，为 TA4、TA41、TA1、TA11、TA2、TA21）寄存器的设定值（0001h ~ FFFFh） 计数源：f1、f8、f2n、fC32
死区时间（宽度）	计数源 $\times p$ 或者无死区时间 p：DTT 寄存器的设定值（01h ~ FFh） 计数源：f1 或者 f1 的 2 分频
导通输出逻辑	可选择有效电平“H”或者“L”。
上下同时导通输出禁止功能	具有上下同时导通输出禁止功能和上下同时导通输出检测功能。
中断频率	定时器 B2 的中断频率可选择 1 ~ 15 个载波周期。

注 1. 当 PM2 寄存器的 PM24 位为“1”（NMI 有效）、INVC0 寄存器的 INV02 位为“1”（使用三相马达控制定时器功能）并且 INV03 位为“1”（允许三相马达控制定时器的输出）时，通过 NMI 输入进行的强制截止功能有效。

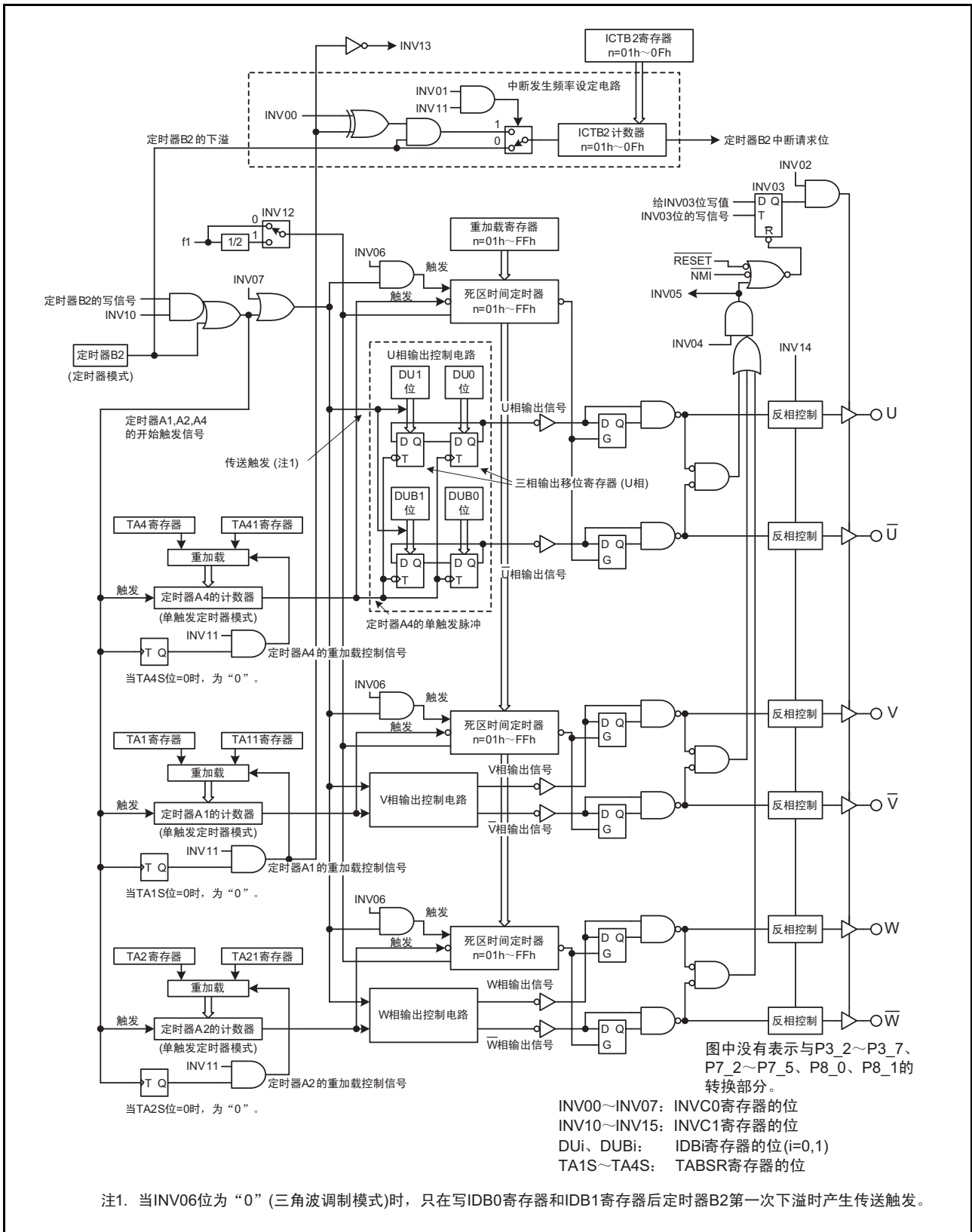


图 17.1 三相马达控制定时器功能的框图

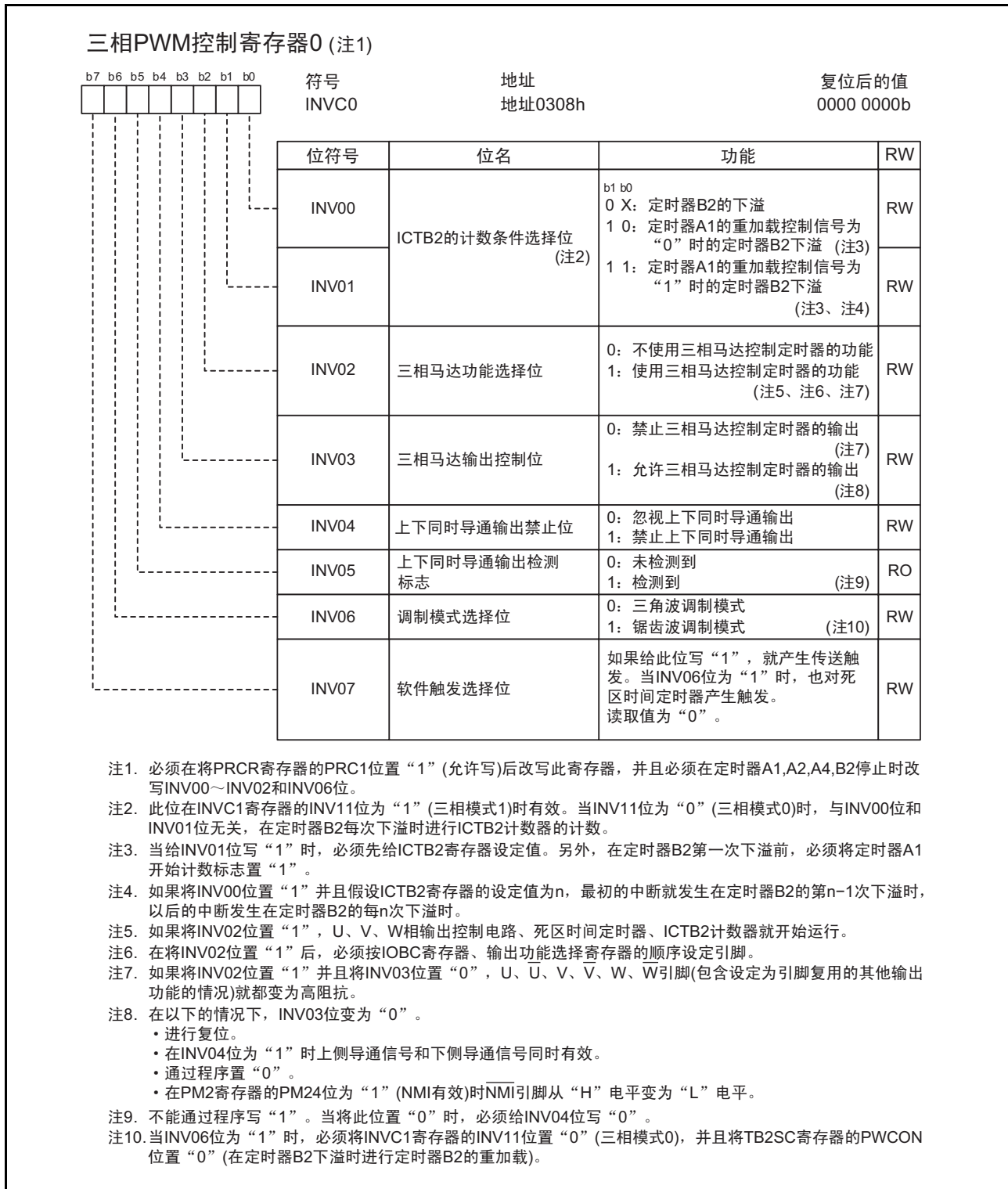


图 17.2 INVC0 寄存器

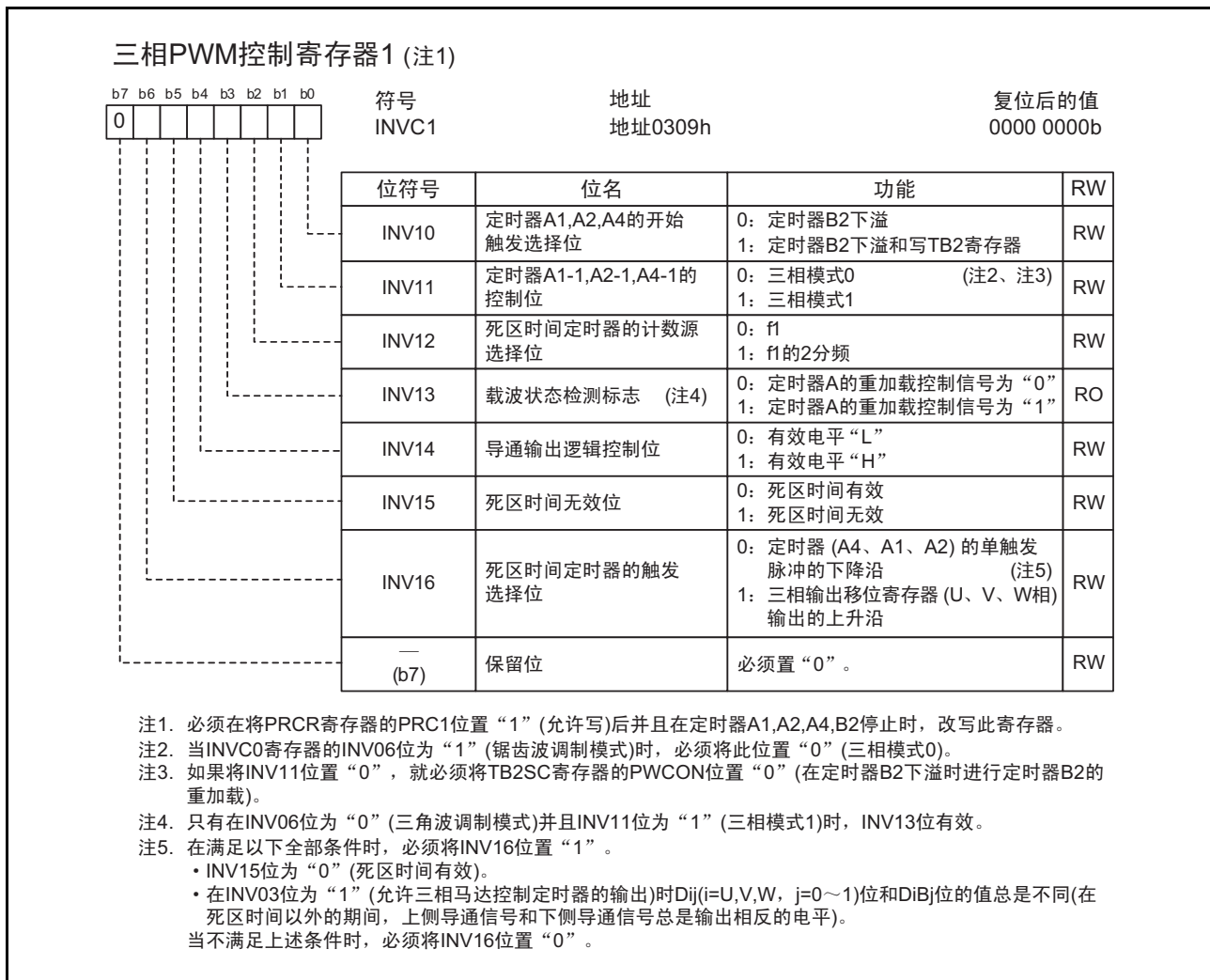


图 17.3 INVC1 寄存器

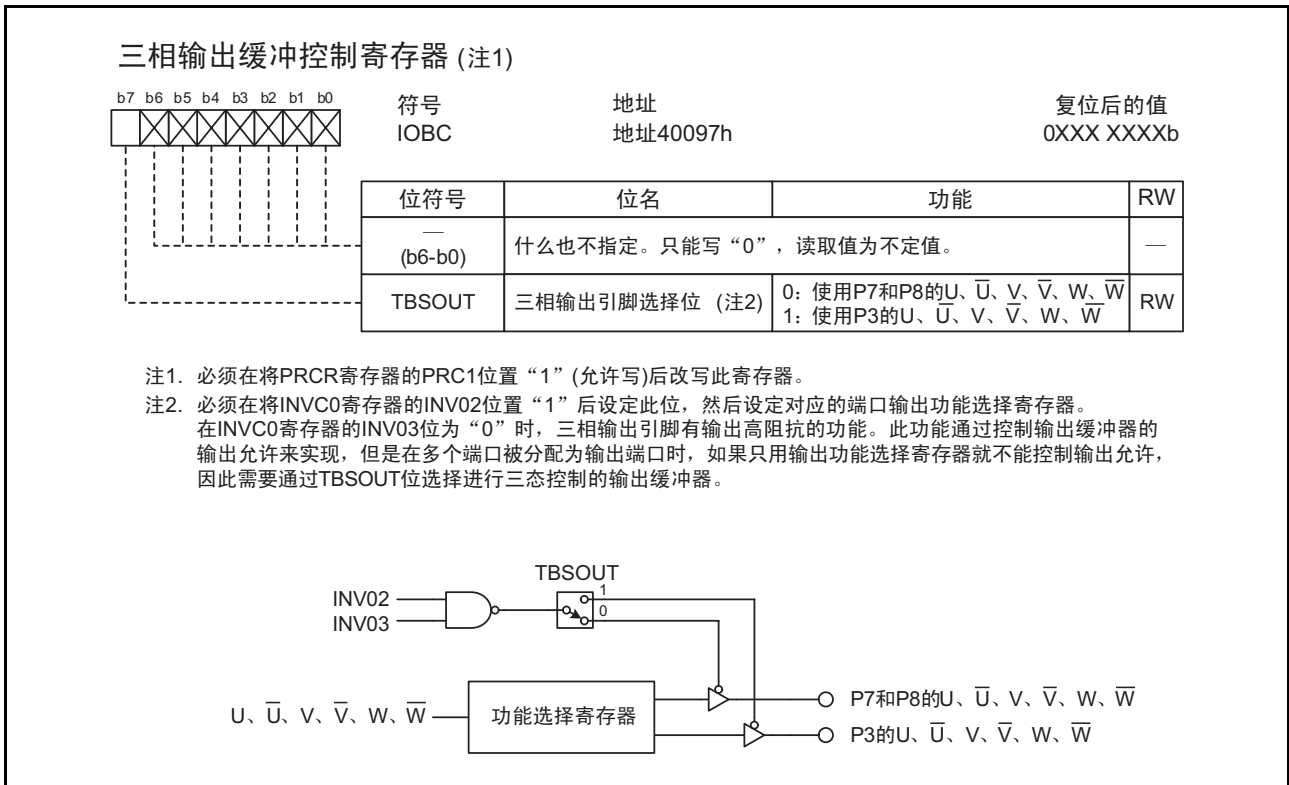


图 17.4 IOBC 寄存器

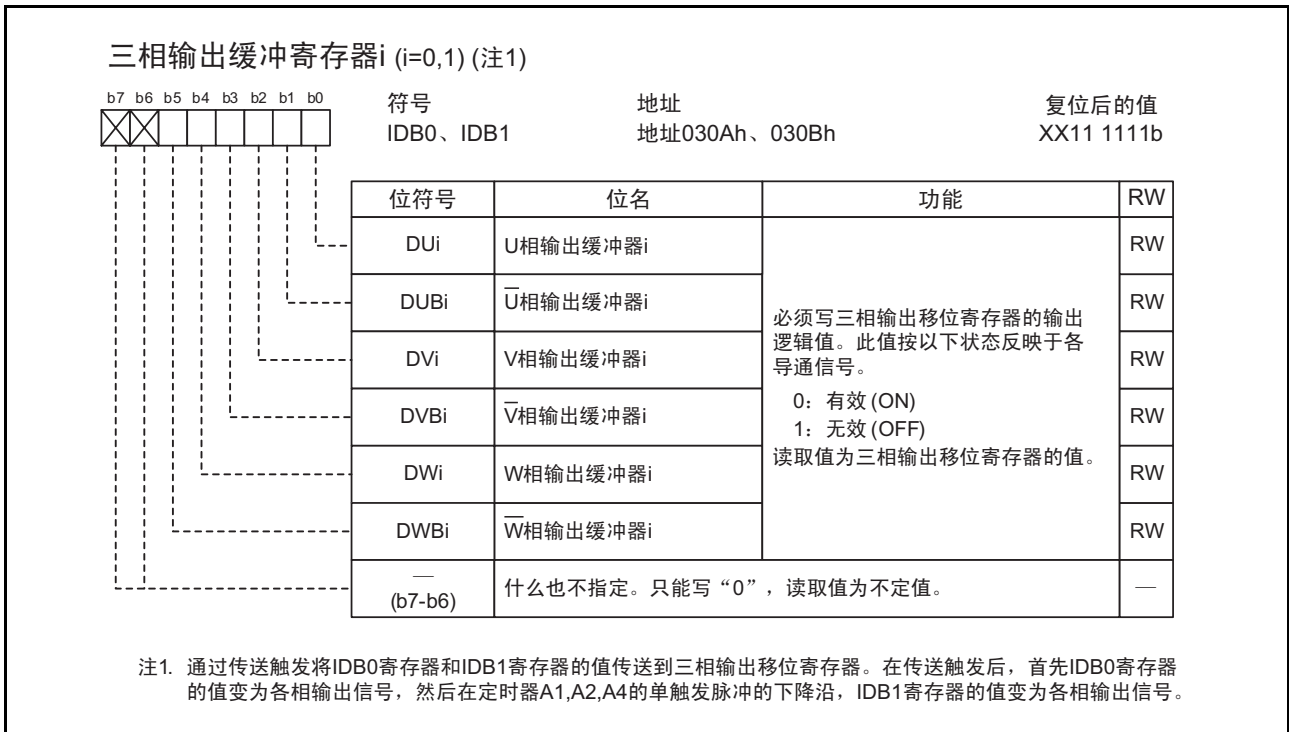


图 17.5 IDB0 寄存器和 IDB1 寄存器



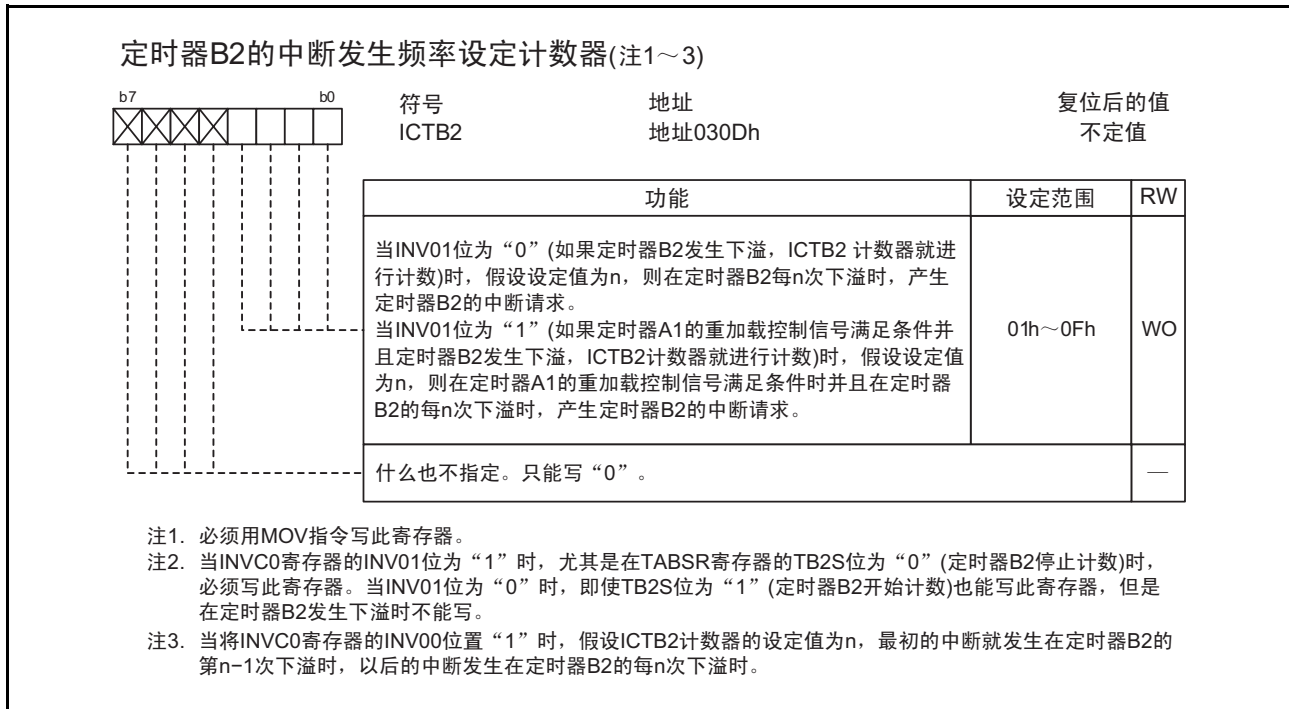


图 17.6 ICTB2 寄存器

## 17.1 三相马达控制定时器的模式

三相马达控制定时器有三角波调制模式和锯齿波调制模式, 三角波调制模式又有三相模式 0 和三相模式 1。各模式的特征和设定如表 17.2 所示。

表 17.2 模式一览表

项目	三角波调制模式		锯齿波调制模式
	三相模式 0	三相模式 1	(三相模式 0)
设定	INV06=0, INV11=0, PWCON=0	INV06=0, INV11=1	INV06=1, INV11=0, PWCON=0
载波波形	三角波		锯齿波
TA11、TA21、TA41 寄存器	不使用	使用	不使用
从 IDB0 寄存器和 IDB1 寄存器到三相输出移位寄存器的传送时序	在写 IDB0 寄存器和 IDB1 寄存器后, 与传送触发 (注 1) 同步, 只进行 1 次传送。		在每次传送触发 (注 1) 时进行传送。
INV16=0 时的死区时间定时器的触发时序	与定时器 A1,A2,A4 的单触发脉冲的下降沿同步。		与定时器 A1,A2,A4 的单触发脉冲的下降沿和传送触发同步。
INVC0 寄存器的 INV00 位和 INV01 位	无效 与 INV00 位和 INV01 位的值无关, 在定时器 B2 每次下溢时进行 ICTB2 计数。	有效	无效 与 INV00 位和 INV01 位的值无关, 在定时器 B2 每次下溢时进行 ICTB2 计数。
INV13 位	无效	有效	无效

注 1. 传送触发: 定时器 B2 下溢、写 INV07 位或者在 INV10 位为“1”时的写 TB2 寄存器。

### 17.2 定时器 B2

在使用三相马达控制定时器功能时，定时器 B2 用于载波控制。

将定时器 B2 设定为定时器模式。

三相马达控制定时器功能时的 TB2 寄存器和 TB2MR 寄存器分别如图 17.7 和图 17.8 所示。在三相模式 1 中，转换载波周期变更时序的 TB2SC 寄存器如图 17.9 所示。

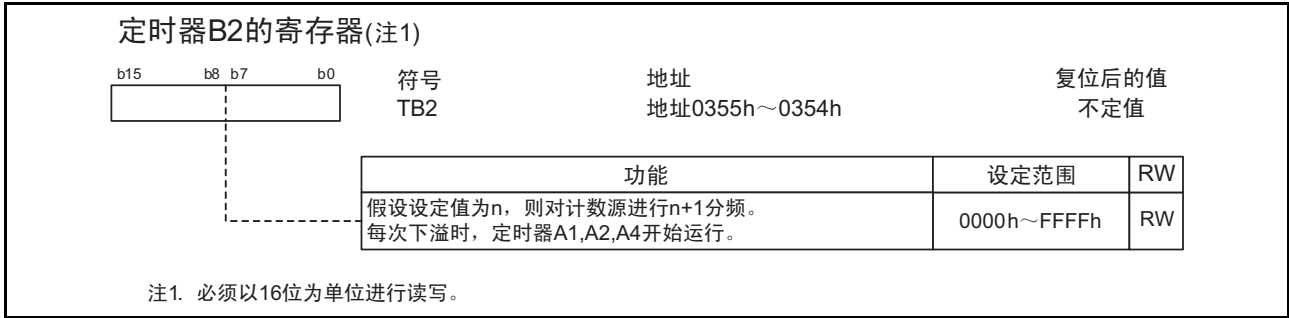


图 17.7 三相马达控制定时器功能时的 TB2 寄存器

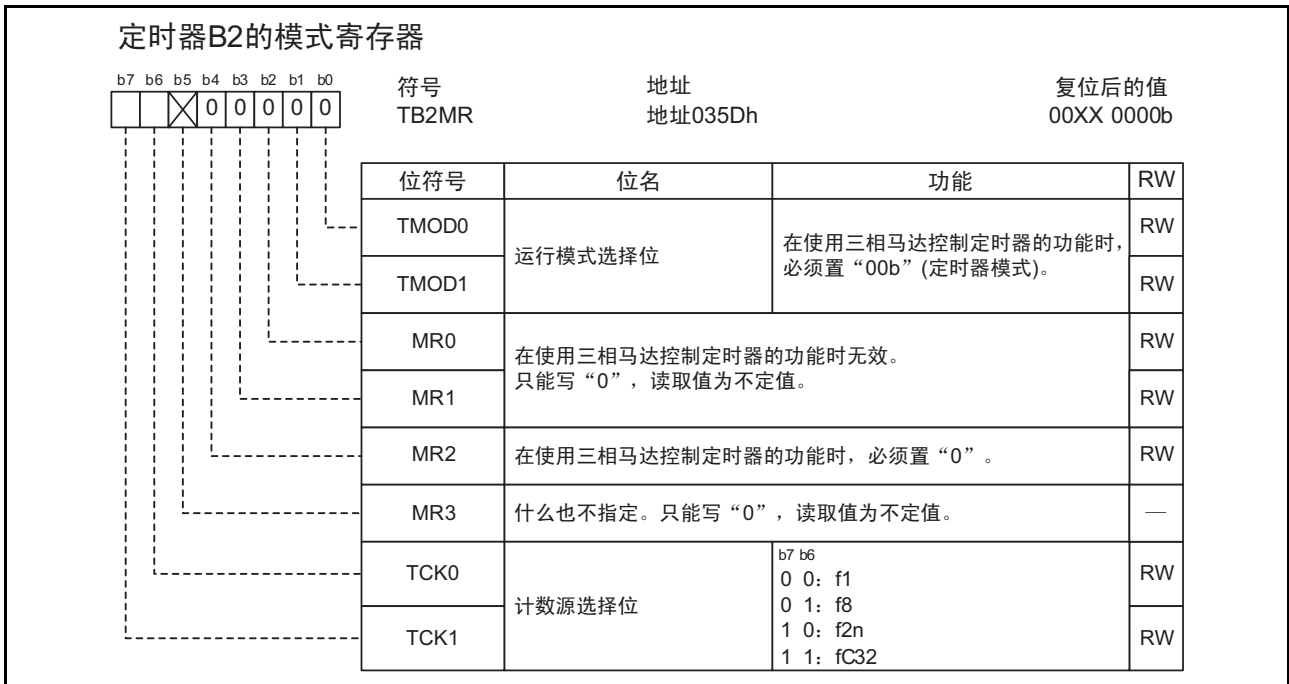


图 17.8 三相马达控制定时器功能时的 TB2MR 寄存器



图 17.9 TB2SC 寄存器



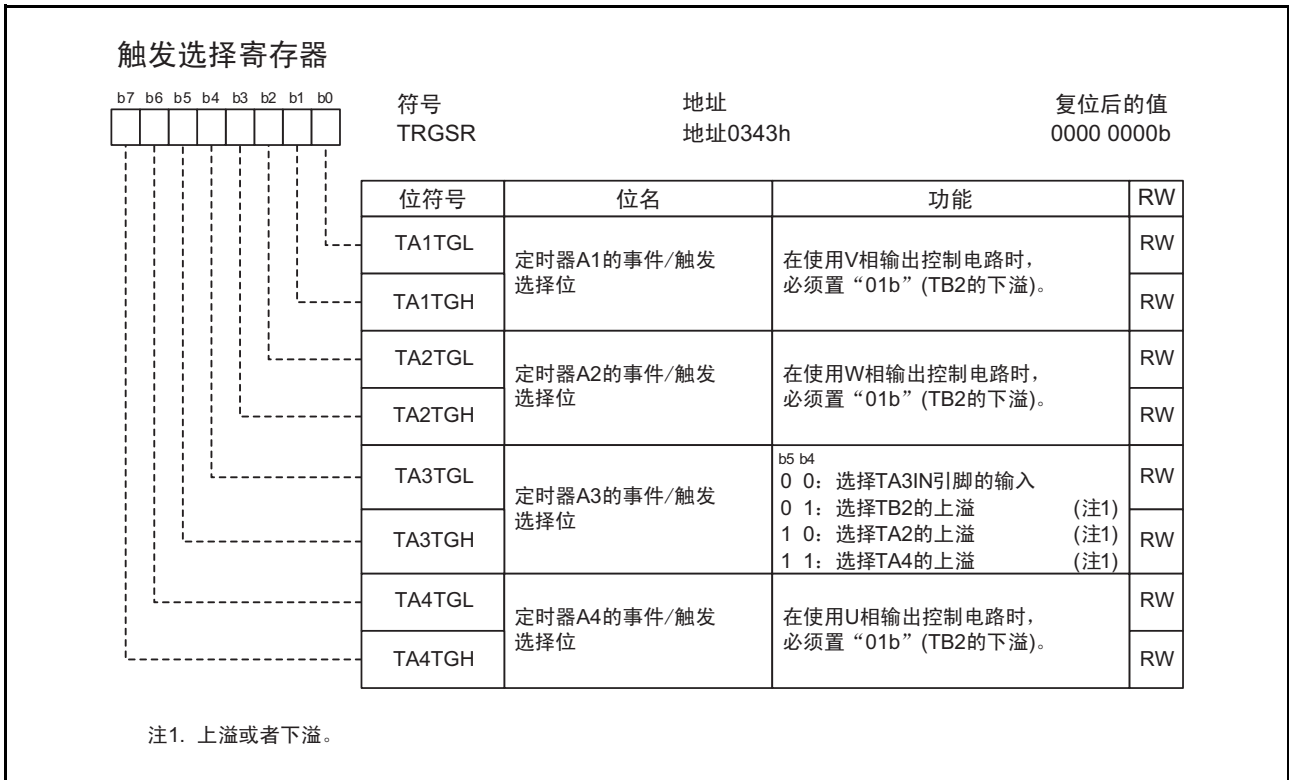


图 17.12 三相马达控制定时器功能时的 TRGSR 寄存器

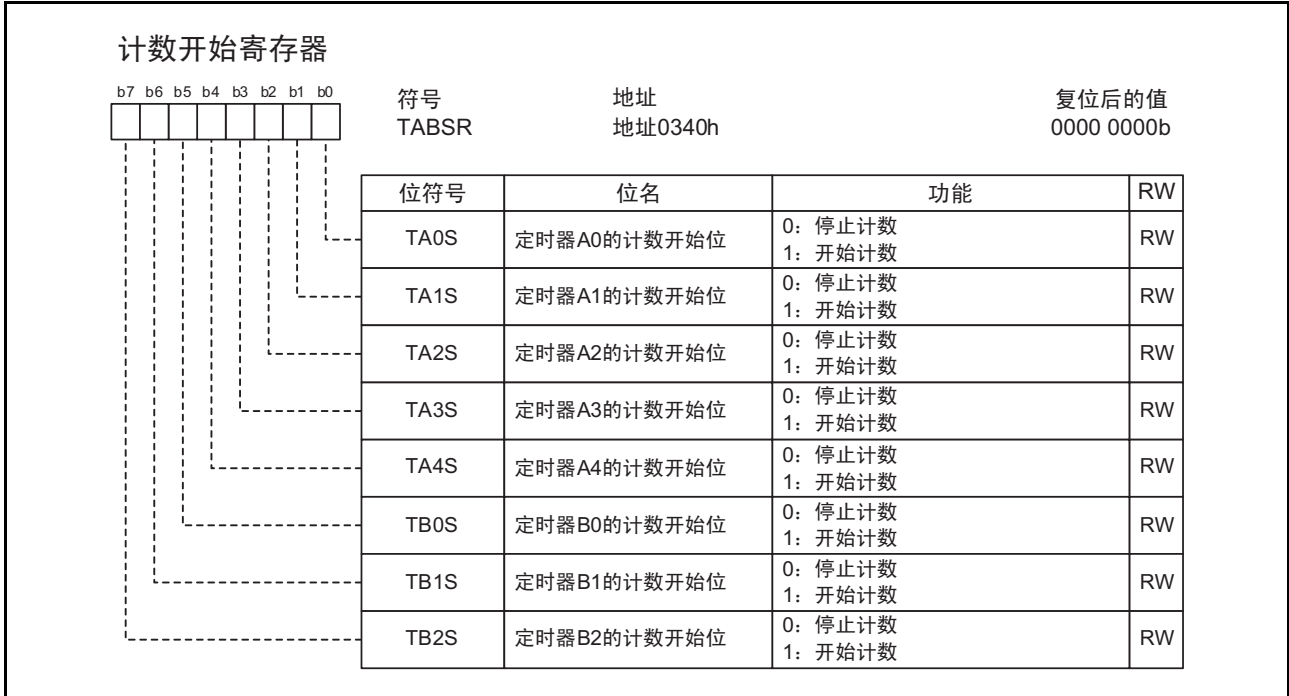


图 17.13 TABSR 寄存器

### 17.4 上下同时导通输出禁止功能和死区时间定时器

三相马达控制定时器有防止因上侧的晶体管和下侧的晶体管同时导通而引起桥臂短路的功能。一个是上下同时导通输出禁止功能，防止因程序错误等而使上下导通输出同时有效。另一个是死区时间定时器，防止因晶体管开关延迟而使上下侧的晶体管同时导通。

能通过将 INVC0 寄存器的 INV04 位置“1”，设定上下同时导通输出禁止功能。此时，如果 U 相和  $\bar{U}$  相、V 相和  $\bar{V}$  相、W 相和  $\bar{W}$  相同时有效，三相马达控制的输出引脚就全部为高阻抗。上下同时导通输出禁止功能时的输出波形例子如图 17.14 所示。

通过将 INVC1 寄存器的 INV15 位置“0”使死区时间定时器有效，通过 DTT 寄存器设定死区时间。DTT 寄存器以及使用死区时间定时器时的输出波形例子分别如图 17.15 和图 17.16 所示。

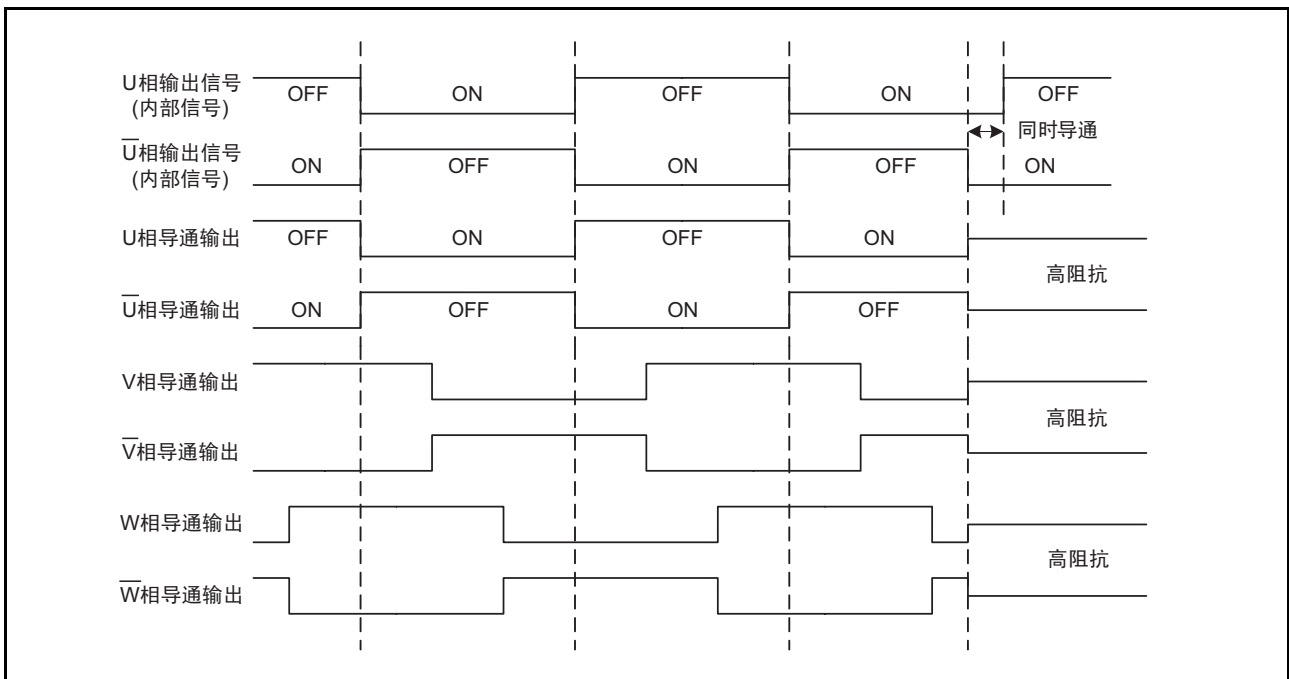


图 17.14 上下同时导通输出禁止功能时的输出波形例子

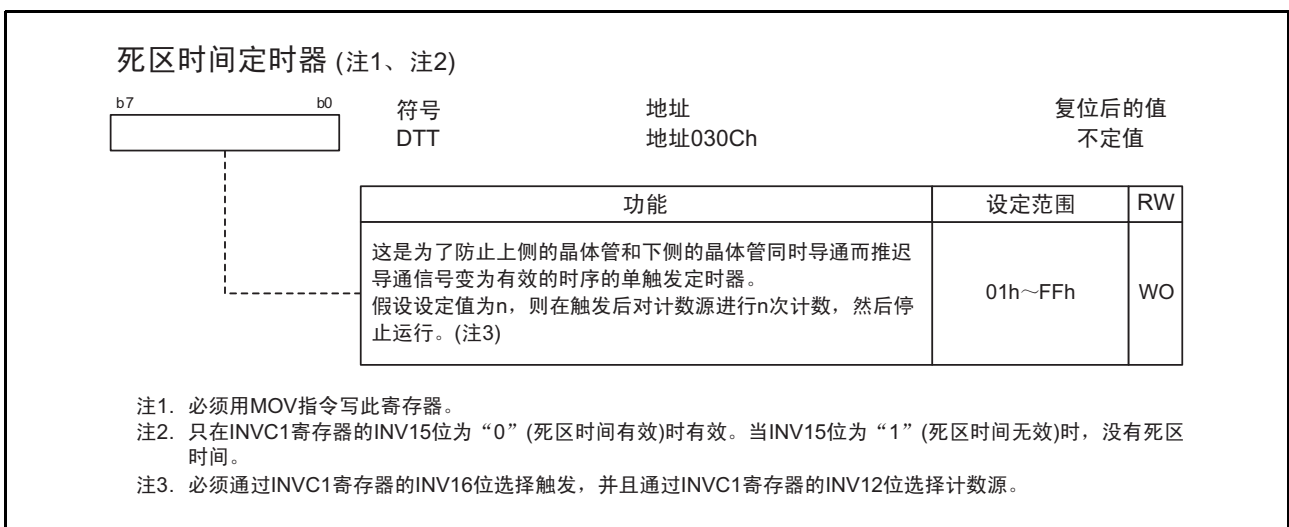


图 17.15 DTT 寄存器

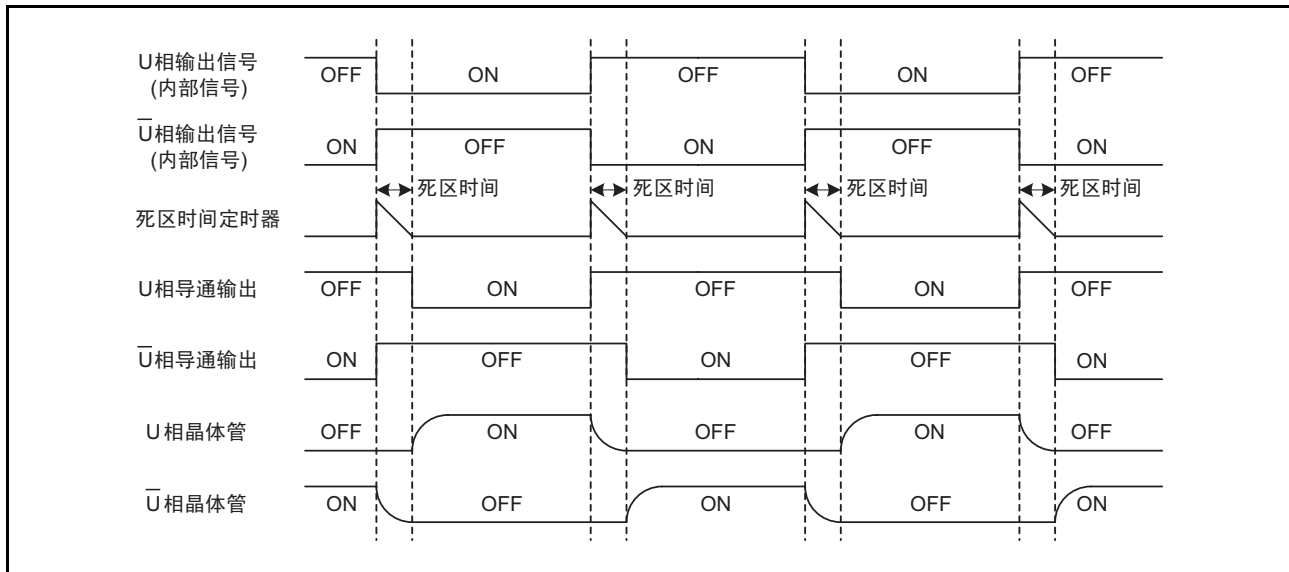


图 17.16 使用死区时间定时器时的输出波形例子

### 17.5 三相马达控制定时器的运行例子

三角波调制波形例子和锯齿波调制波形例子分别如图 17.17 和图 17.18 所示。

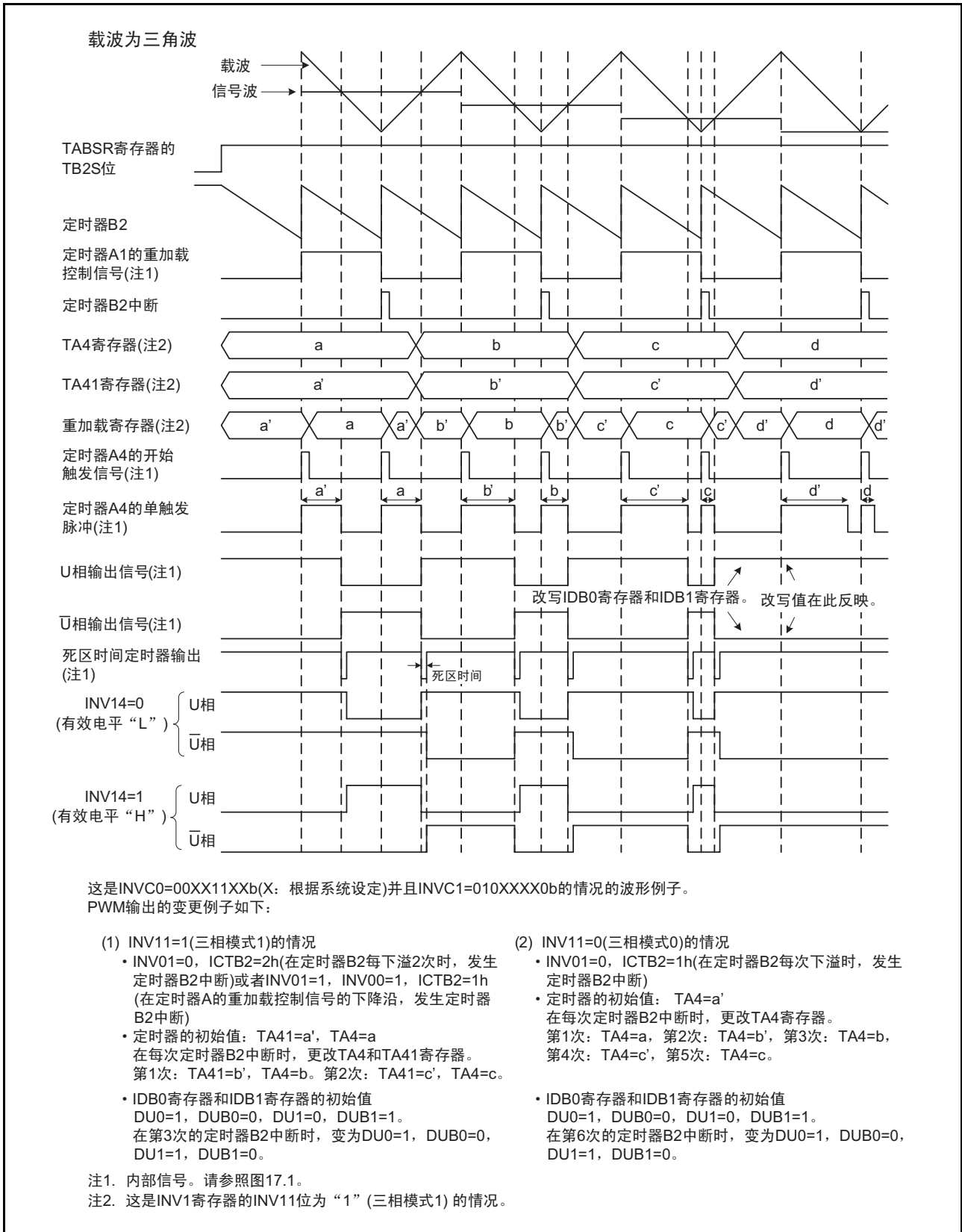


图 17.17 三角波调制的运行例子

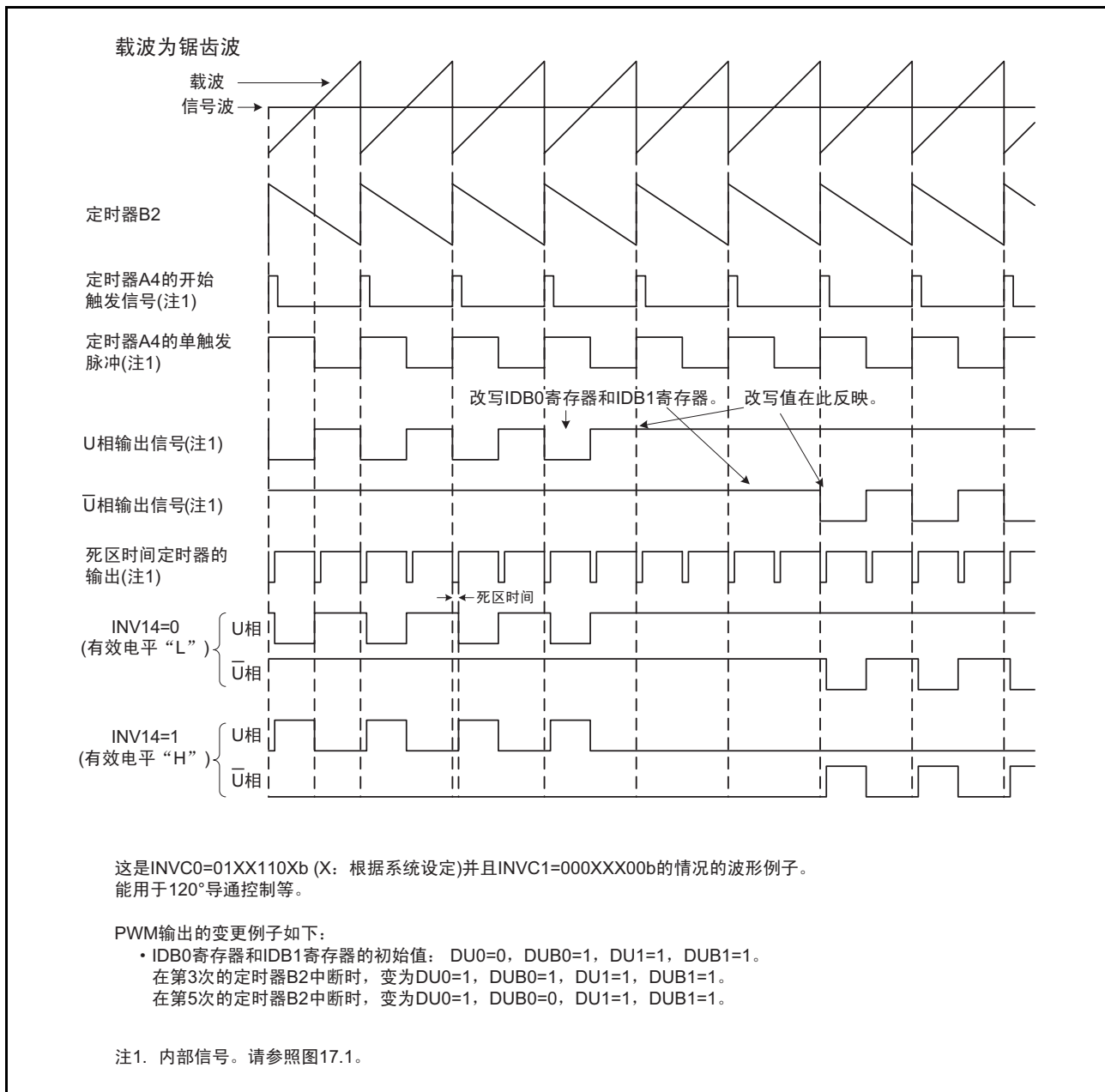


图 17.18 锯齿波调制的运行例子



## 17.6 使用三相马达控制定时器功能时的注意事项

### 17.6.1 关机功能

- 当PM2寄存器的PM24位为“1”（NMI有效）、INVC0寄存器的INV03位为“1”（允许三相马达控制定时器的输出）并且INV02位为“1”（使用三相马达控制定时器功能）时，如果给NMI引脚输入“L”电平，TA1OUT、TA2OUT、TA4OUT引脚就变为高阻抗。

### 17.6.2 寄存器的设定

- 不能在定时器B2发生上溢前后给TAi1寄存器（i=1,2,4）设定值。  
在给TAi1寄存器设定值时，必须先读TB2寄存器的值，在确认发生上溢前还有充分的时间后再进行设定。为了缩短读TB2寄存器和写TAi1寄存器之间的时间间隔，不能在此期间执行中断处理等。如果在读TB2寄存器后到发生上溢前没有充分的时间，就必须在发生上溢后重新设定TAi1寄存器。

## 18. 串行接口

串行接口有 9 个通道（UART0 ~ UART8）。

UARTi（i=0 ~ 8）有各自专用的发送 / 接收时钟发生定时器并独立运行。

UART0 ~ UART6 以及 UART7、UART8 的框图分别如图 18.1 和图 18.2 所示。

UARTi 有以下模式：

- 时钟同步串行接口模式 (UART0 ~ UART8)
- 异步串行接口模式 (UART 模式) (UART0 ~ UART8)
- 特殊模式 1 (I<sup>2</sup>C 模式) (UART0 ~ UART6)
- 特殊模式 2 (UART0 ~ UART6)
- 特殊模式 4 (总线冲突检测功能、IE 模式) (选项 (注 1)) (UART0 ~ UART6)

UARTi 的相关寄存器如图 18.3 ~ 图 18.19 所示。

寄存器和引脚的设定请参照各模式的表。

注 1. 如果要使用选项功能，请向本公司的营业窗口询问。

表 18.1 UART0 ~ UART8 的功能比较

模式 / 功能	UART0 ~ UART6	UART7、UART8
时钟同步串行接口模式	有	有
串行数据逻辑转换的选择	能选择	不能选择
UART 模式	有	有
CTS/RTS 功能的选择	能选择	能选择
TXD、RXD 输入 / 输出极性转换的选择	能选择	不能选择
特殊模式 1 (I <sup>2</sup> C 模式)	有	无
特殊模式 2	有	无
特殊模式 4 (IE 模式) (选项 (注 1))	有	无
TXD、RXD 引脚的输出形式	推挽输出 可通过功能选择寄存器，设定为 N 沟道漏极开路输出。	推挽输出 可通过功能选择寄存器，设定 N 沟道漏极开路输出。

注 1. 如果要使用选项功能，请向本公司的营业窗口询问。

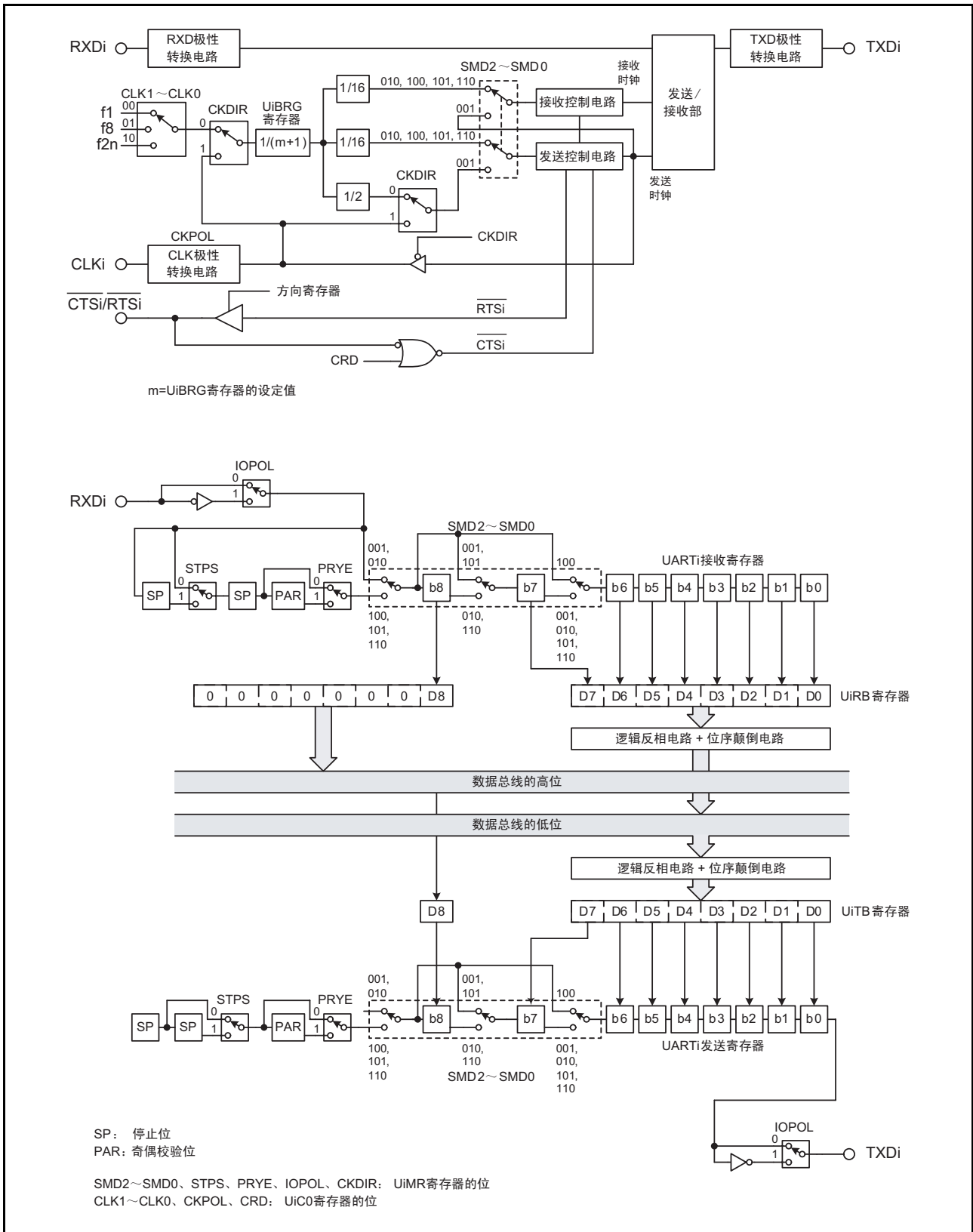


图 18.1 UARTi (i=0~6) 的框图

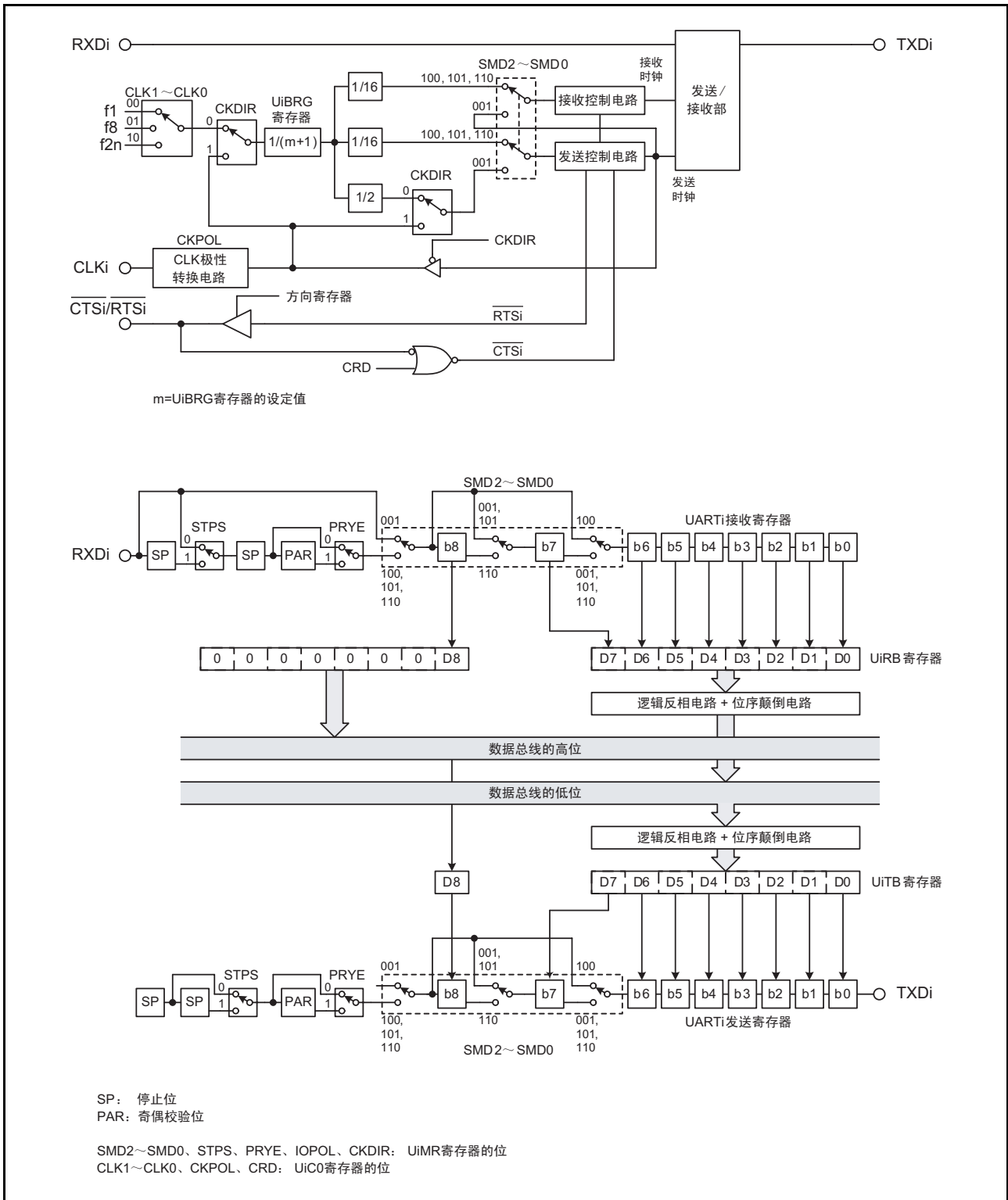


图 18.2 UARTi (i=7,8) 的框图

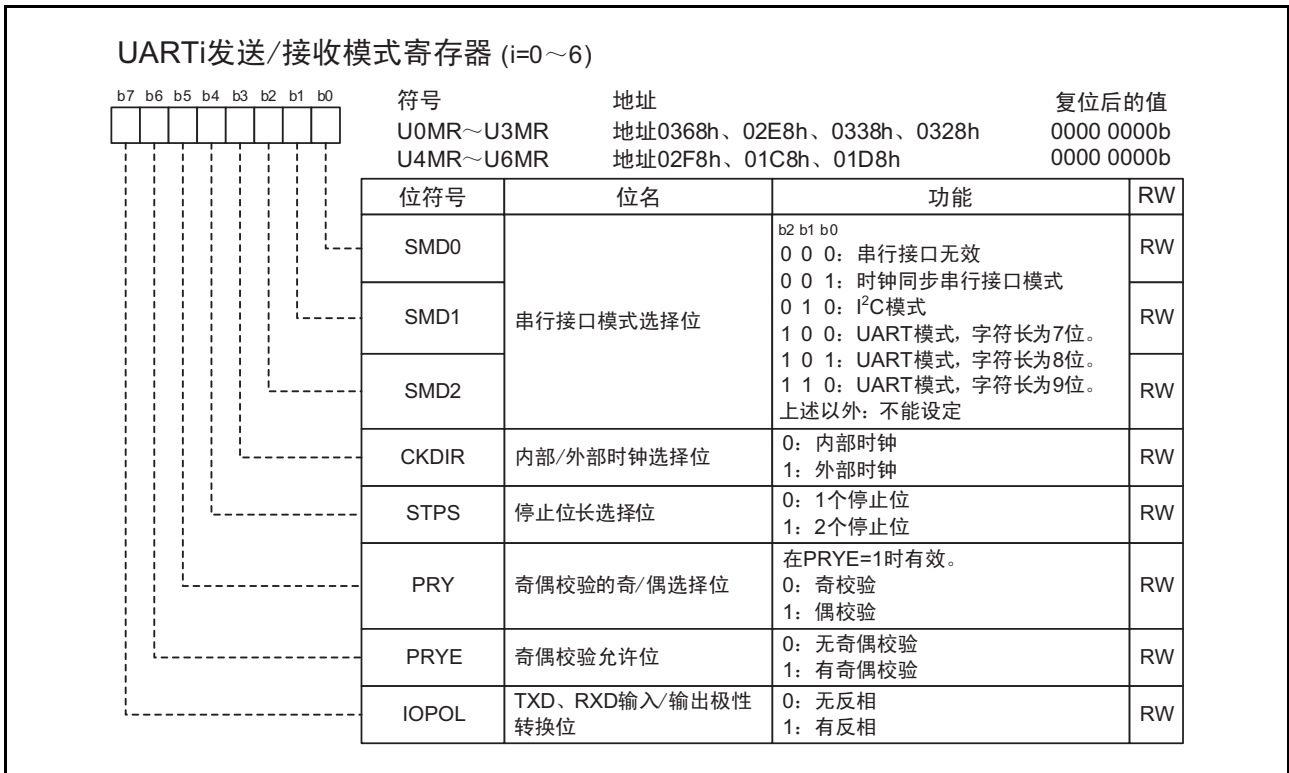


图 18.3 U0MR ~ U6MR 寄存器

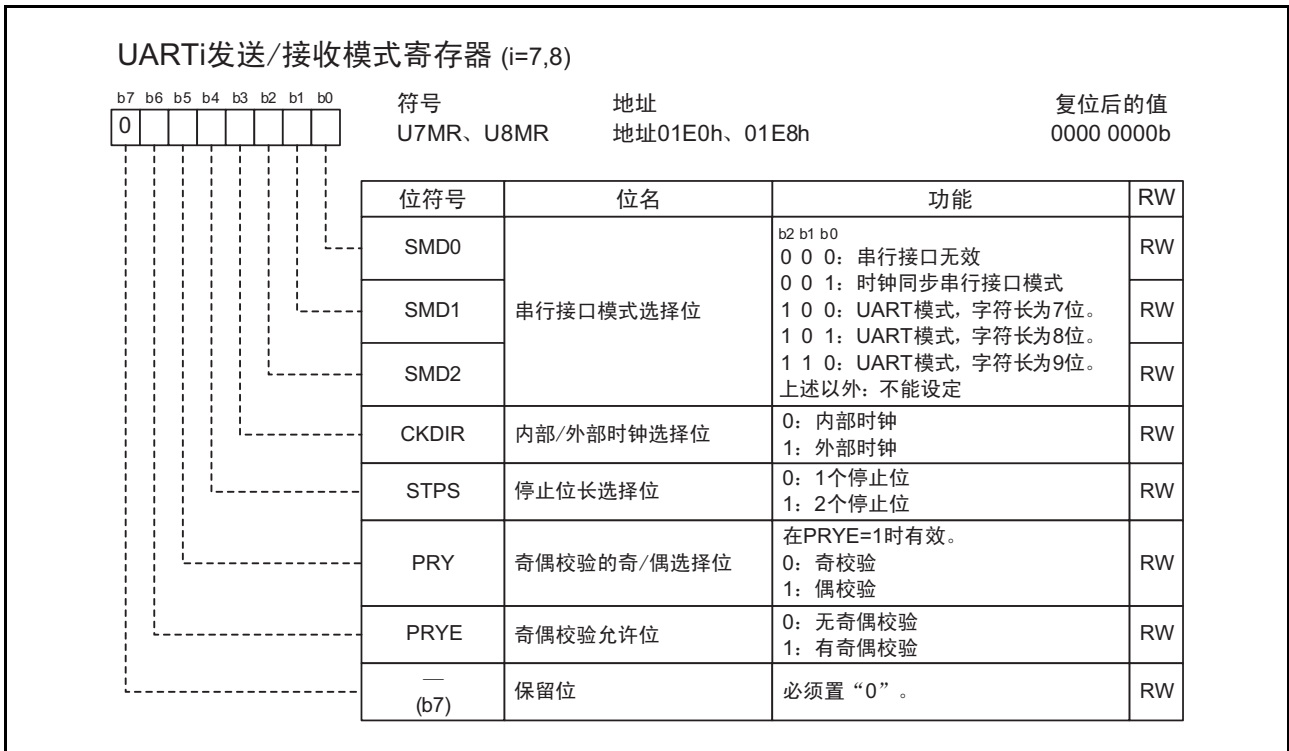


图 18.4 U7MR 寄存器和 U8MR 寄存器

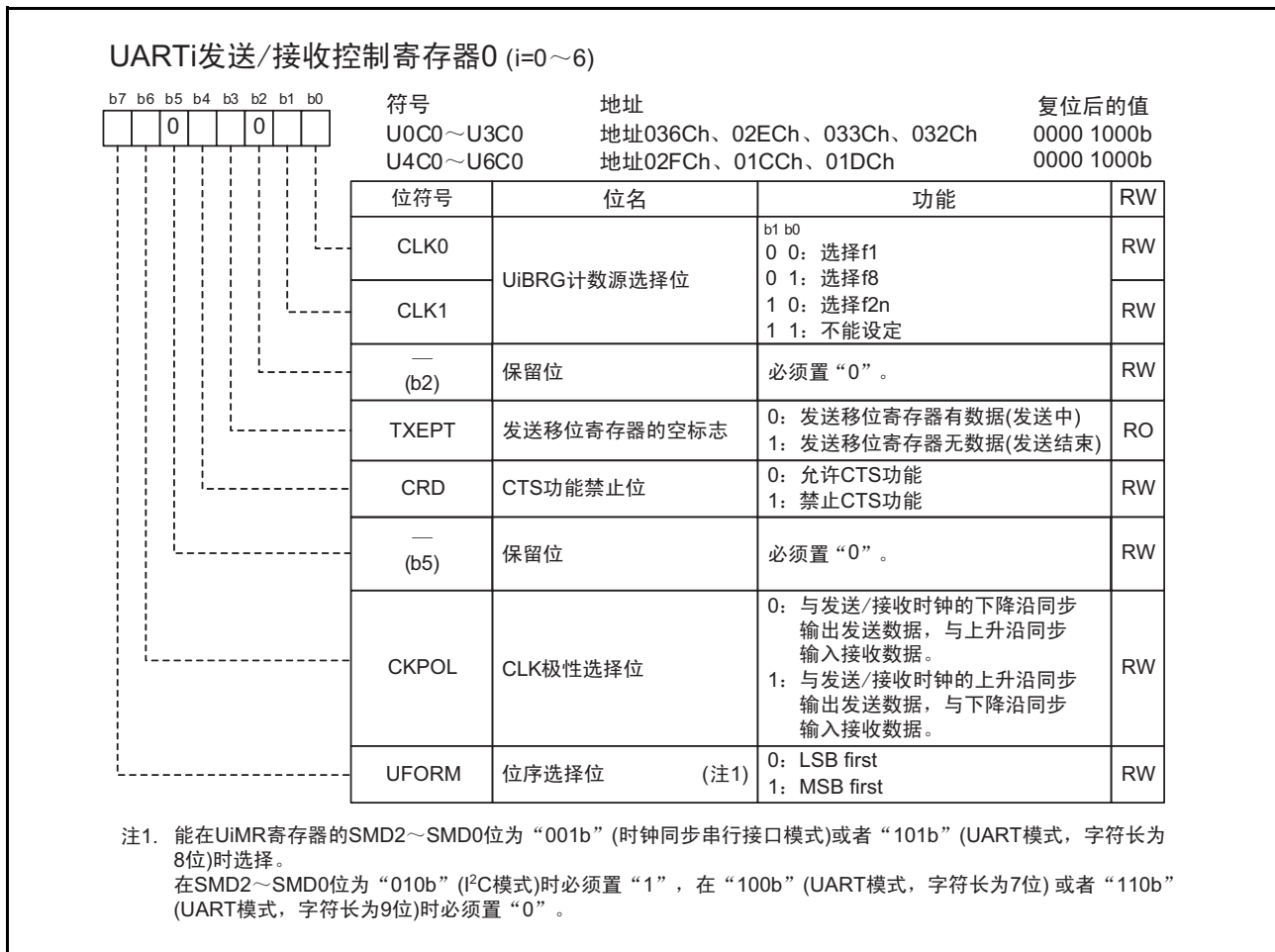


图 18.5 U0C0 ~ U6C0 寄存器

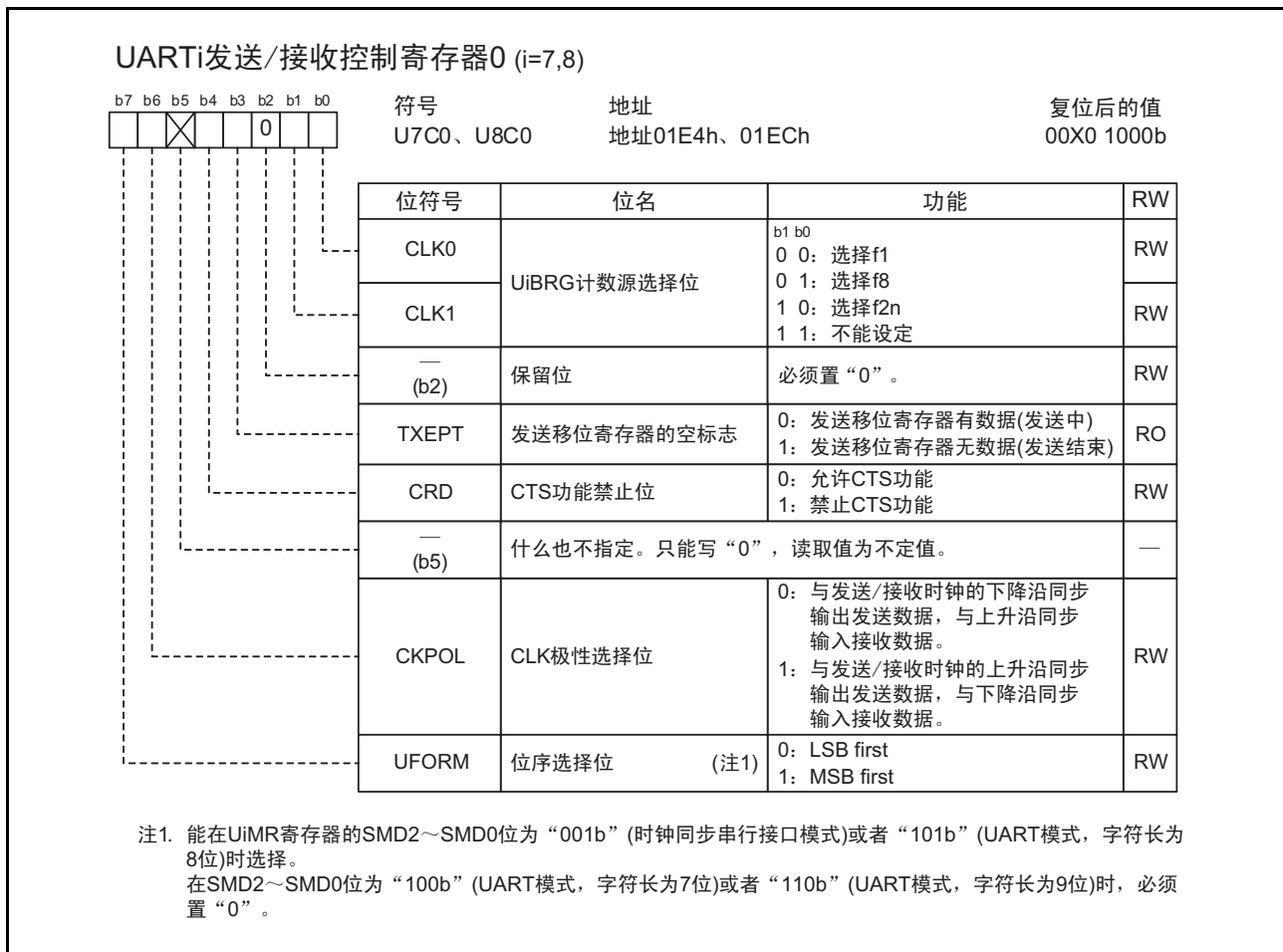


图 18.6 U7C0 寄存器和 U8C0 寄存器

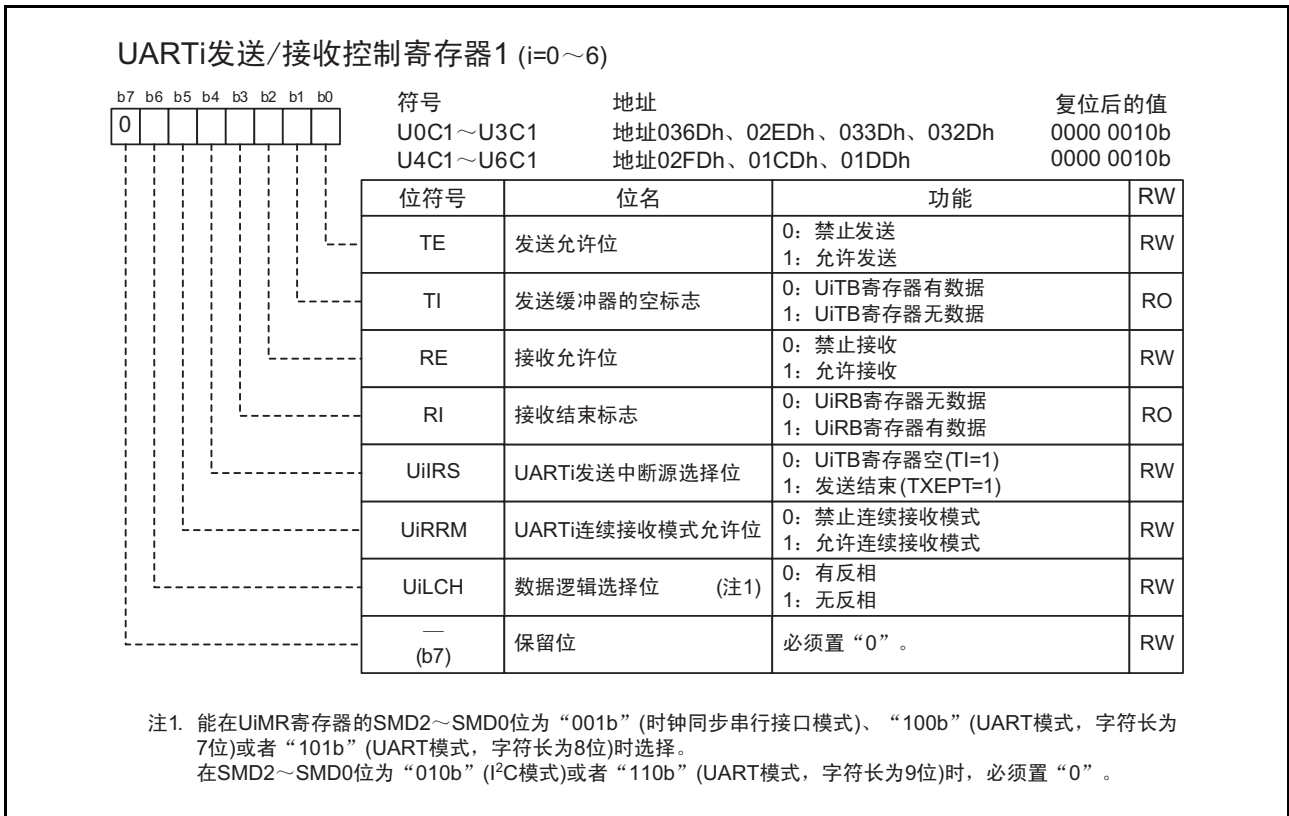


图 18.7 U0C1 ~ U6C1 寄存器

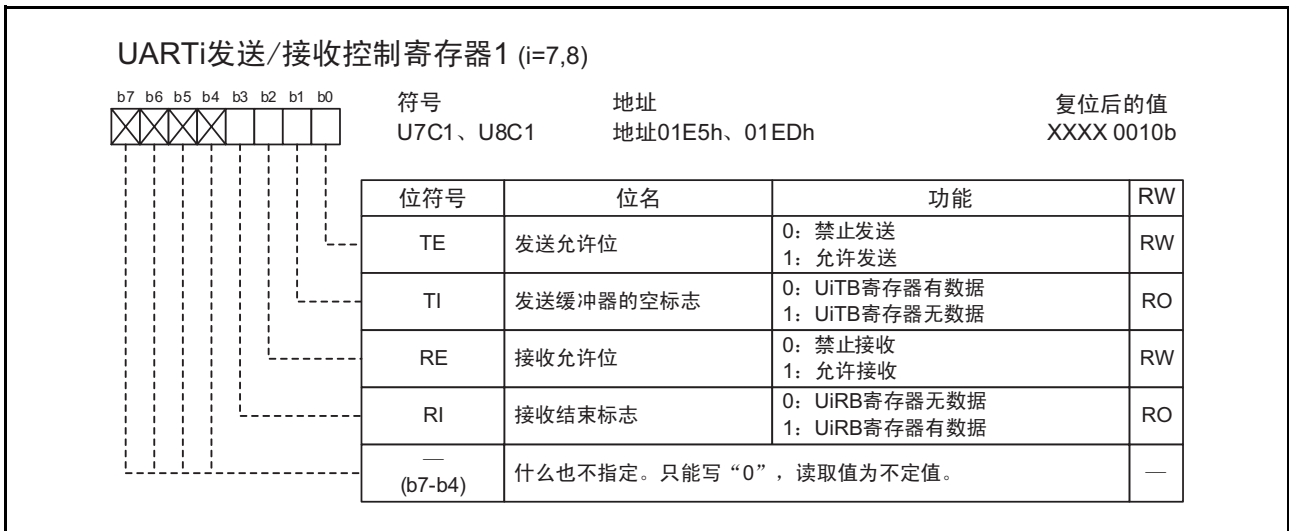


图 18.8 U7C1 寄存器和 U8C1 寄存器



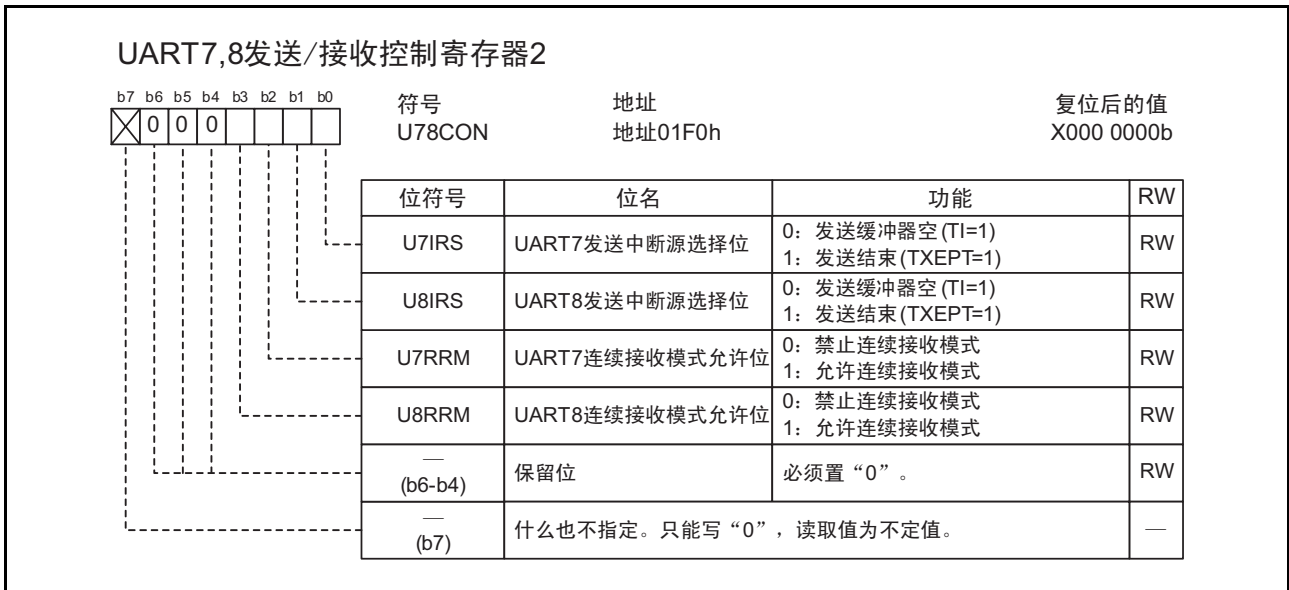


图 18.9 U78CON 寄存器

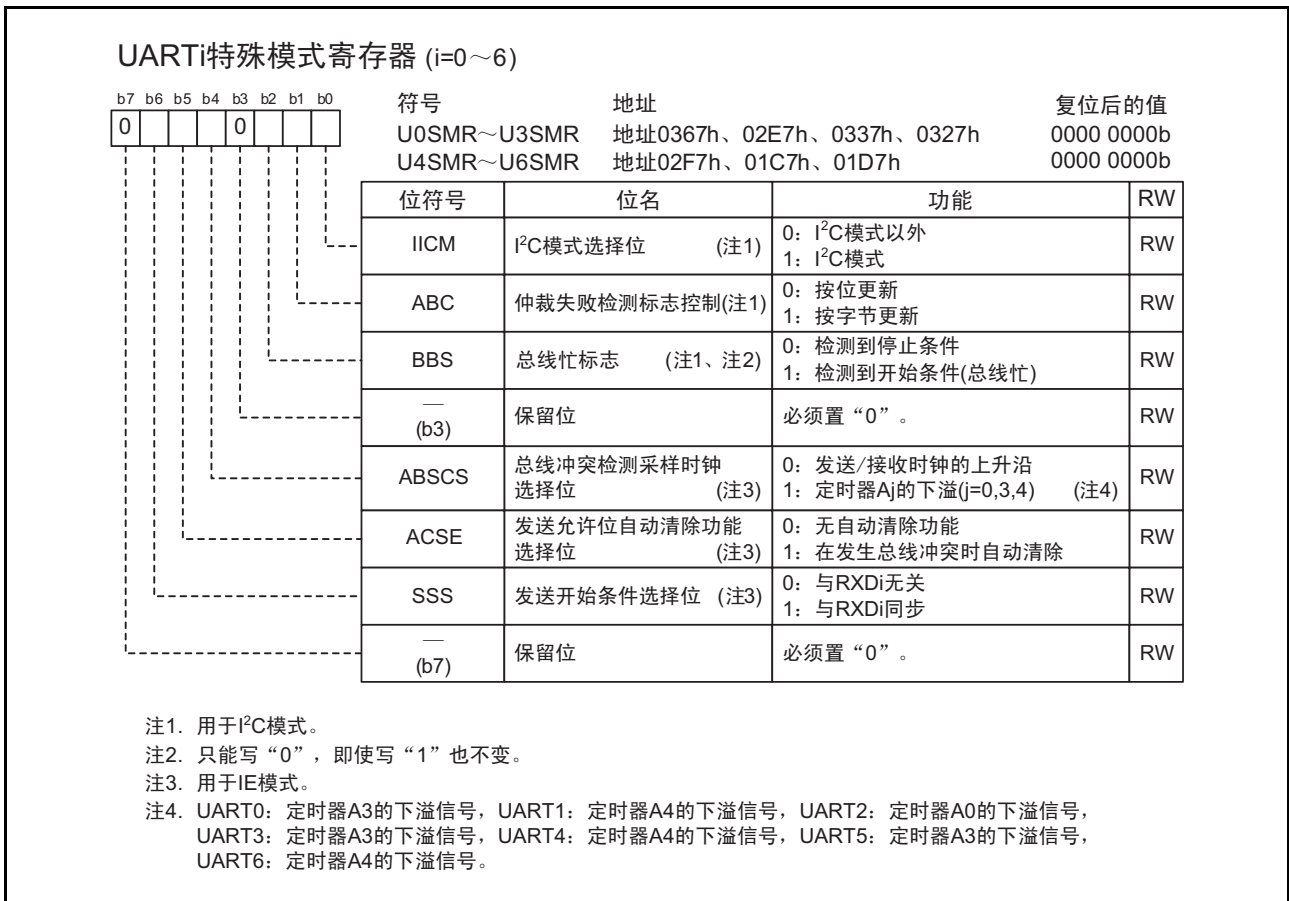


图 18.10 U0SMR ~ U6SMR 寄存器

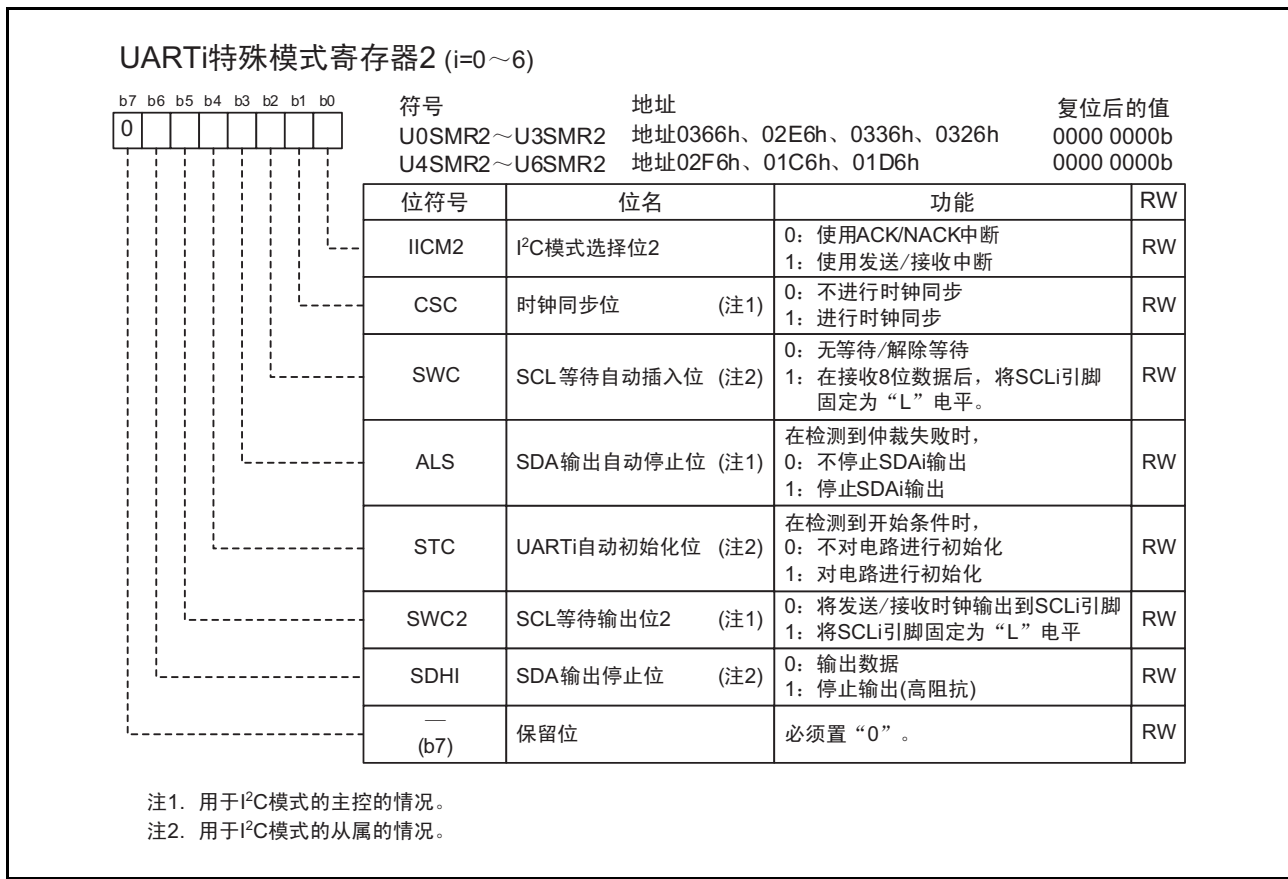


图 18.11 U0SMR2 ~ U6SMR2 寄存器



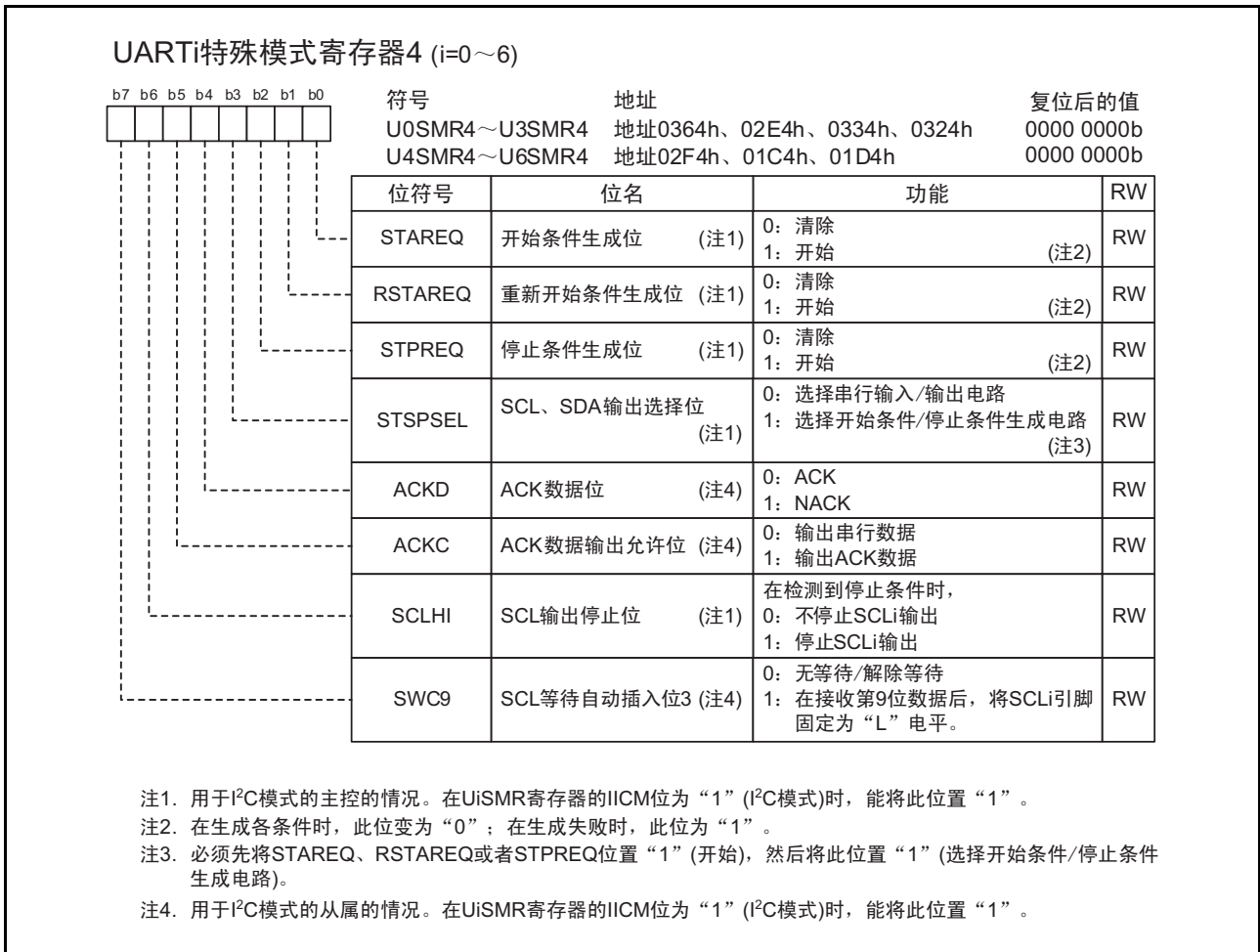


图 18.13 U0SMR4 ~ U6SMR4 寄存器

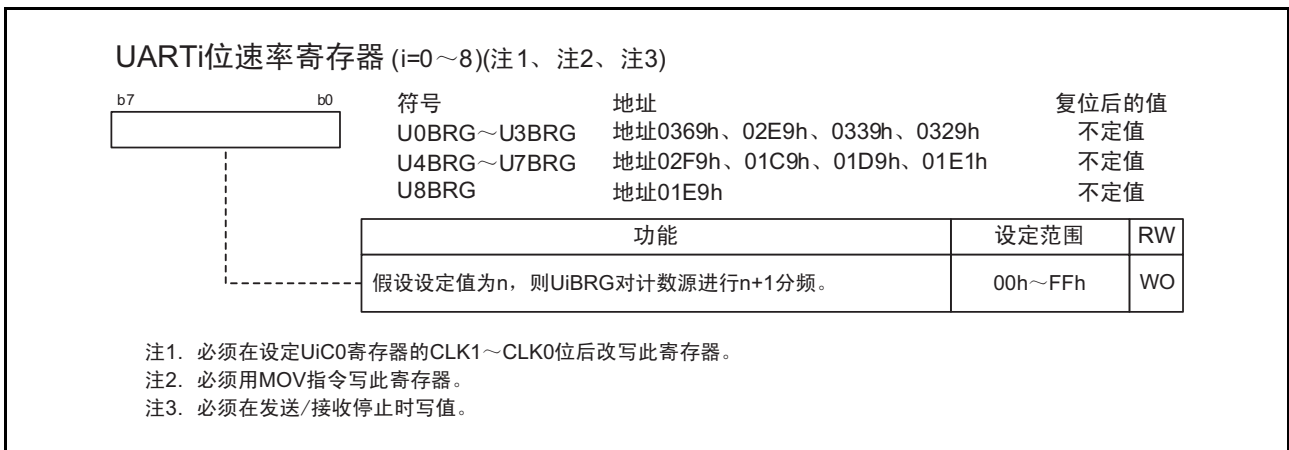


图 18.14 U0BRG ~ U8BRG 寄存器

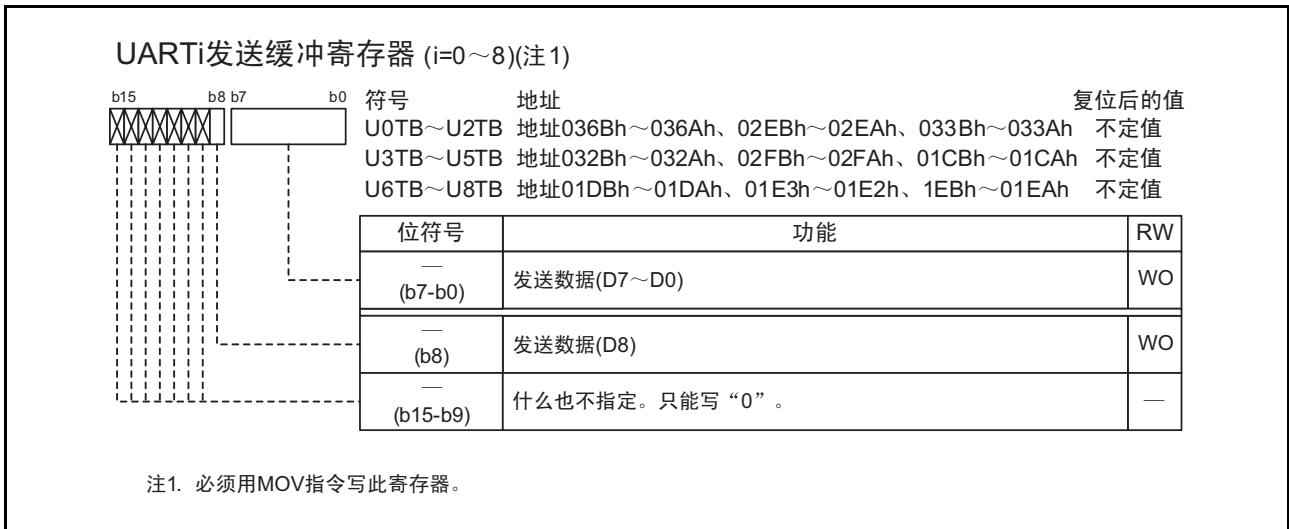


图 18.15 U0TB ~ U8TB 寄存器

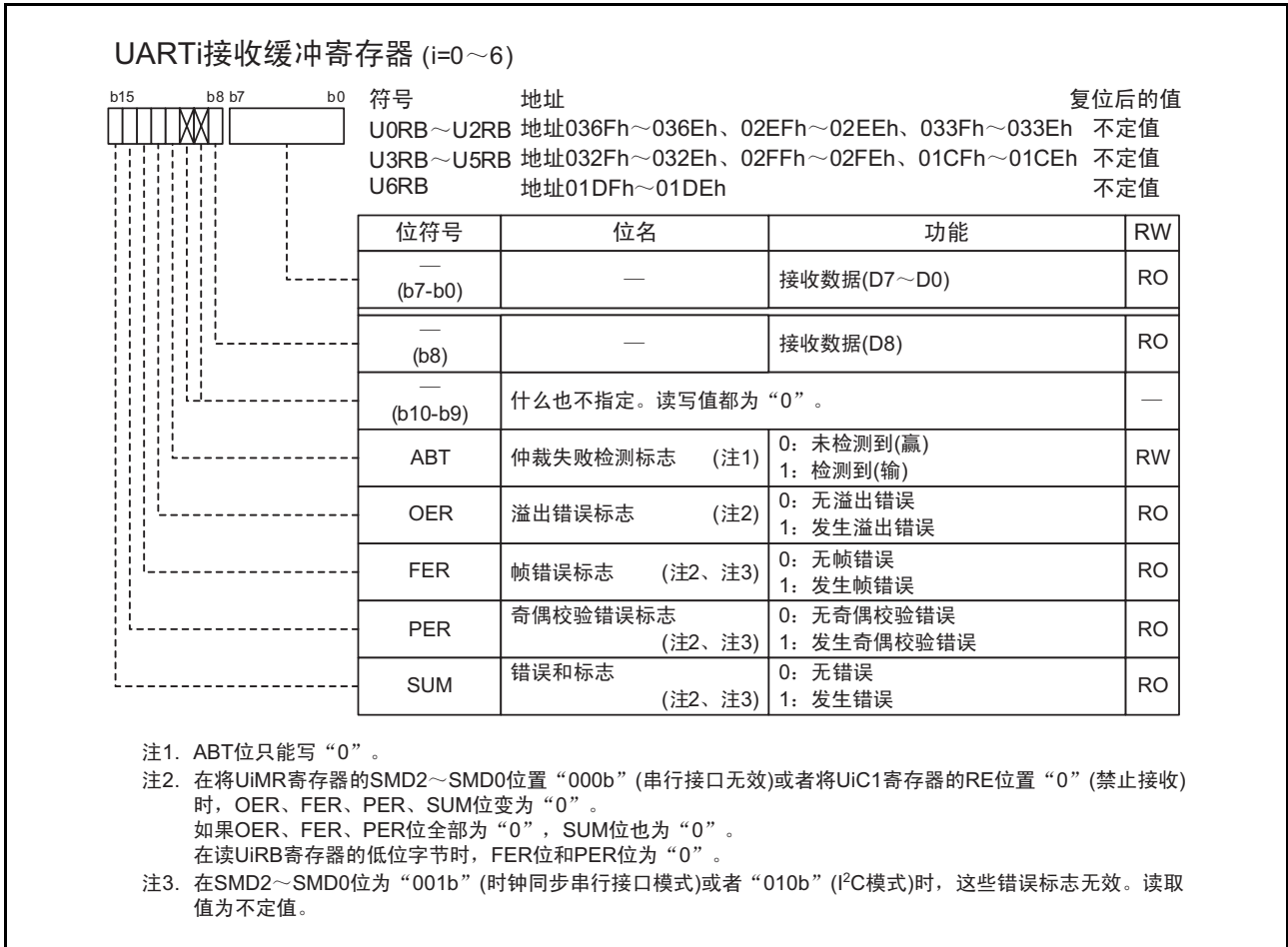


图 18.16 U0RB ~ U6RB 寄存器

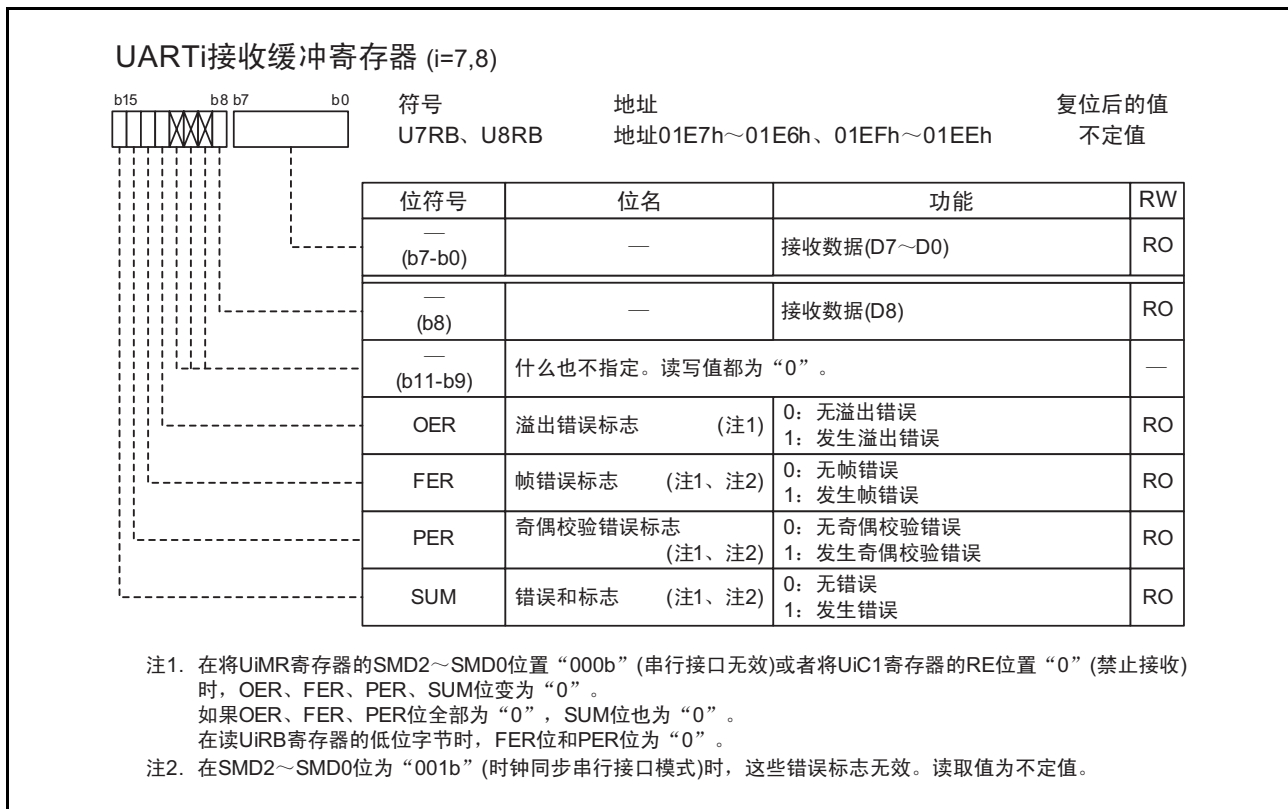


图 18.17 U7RB 寄存器和 U8RB 寄存器

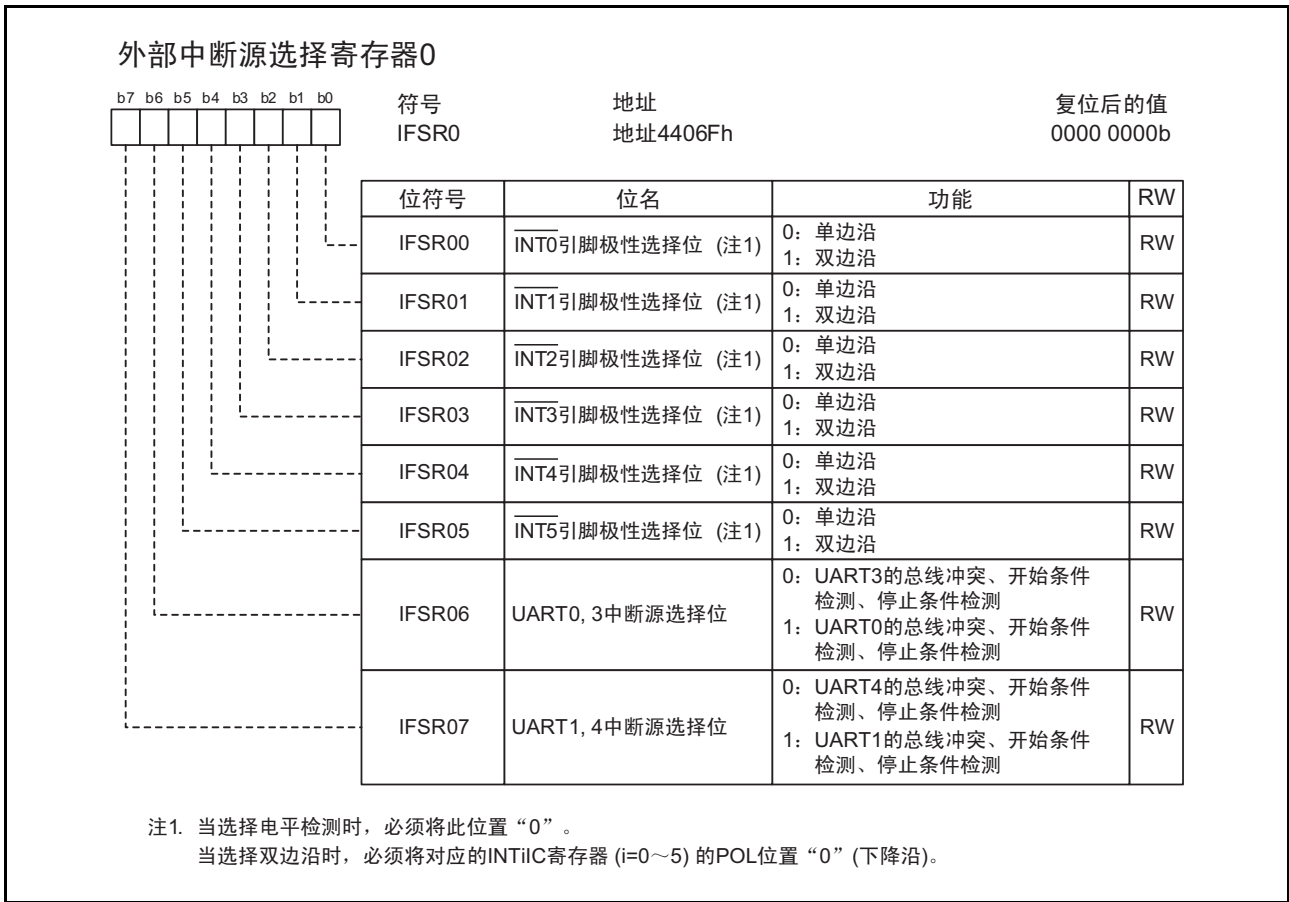


图 18.18 IFSR0 寄存器

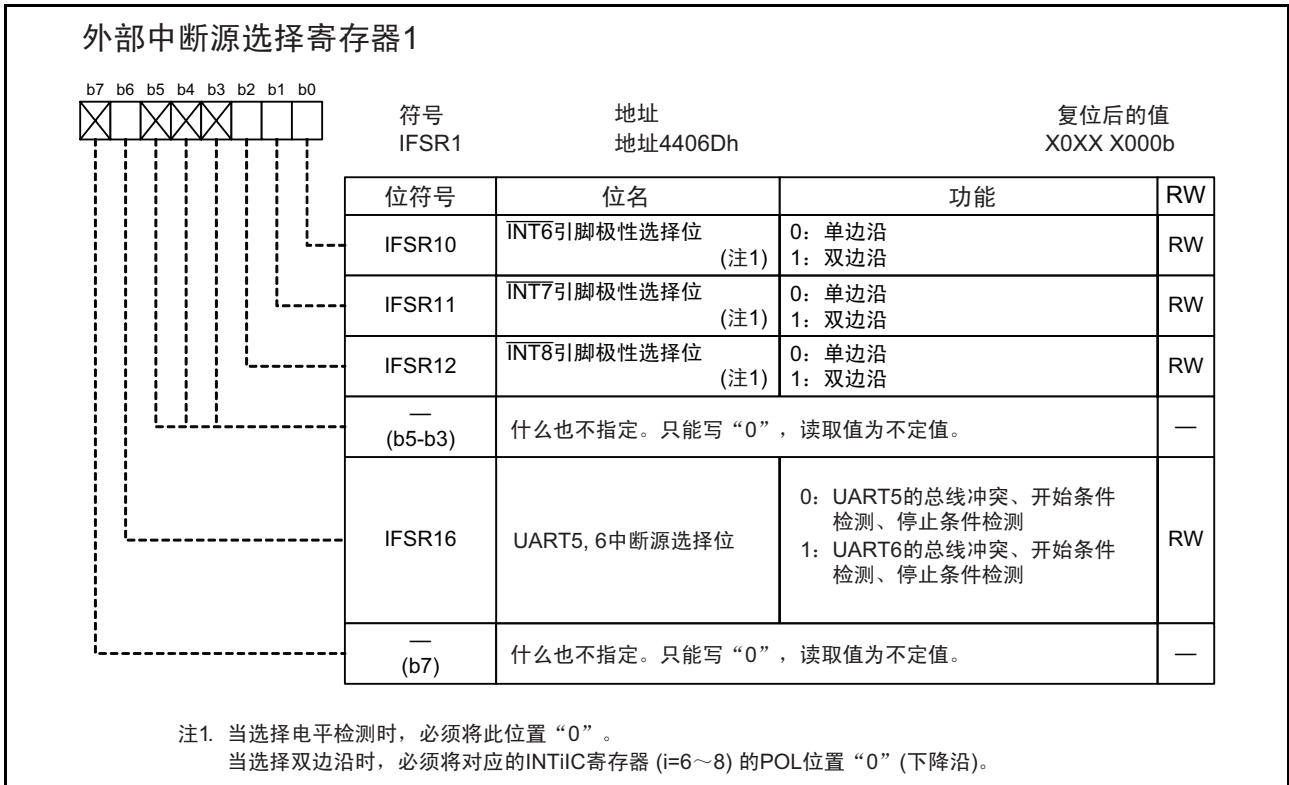


图 18.19 IFSR1 寄存器

## 18.1 时钟同步串行接口模式

时钟同步串行接口模式是与发送 / 接收时钟同步发送和接收数据的模式。时钟同步串行接口模式的规格如表 18.2 所示。

表 18.2 时钟同步串行接口模式的规格

项目	规格
数据格式	字符长: 8 位
发送 / 接收时钟	<ul style="list-style-type: none"> <li>当 UiMR 寄存器 (i=0 ~ 8) 的 CKDIR 位为 “0” (内部时钟) 时:  <math display="block">\frac{f_x}{2(m+1)}</math> <math>f_x</math>: f1、f8、f2n      <math>m</math>: UiBRG 寄存器的设定值 (00h ~ FFh)</li> <li>当 CKDIR 位为 “1” (外部时钟) 时: CLKi 引脚的输入</li> </ul>
发送 / 接收控制	选择 CTS 功能、RTS 功能或者 CTS/RTS 功能无效。
发送开始条件	开始发送时需要以下的条件 (注 1): <ul style="list-style-type: none"> <li>UiC1 寄存器的 TE 位为 “1” (允许发送)。</li> <li>UiC1 寄存器的 TI 位为 “0” (UiTB 寄存器有数据)。</li> <li>在选择 CTS 功能时, CTSi 引脚的输入信号为 “L” 电平。</li> </ul>
接收开始条件	开始接收时需要以下的条件 (注 1): <ul style="list-style-type: none"> <li>UiC1 寄存器的 RE 位为 “1” (允许接收)。</li> <li>TE 位为 “1” (允许发送)。</li> <li>TI 位为 “0” (UiTB 寄存器有数据)。</li> <li>在选择 CTS 功能时, CTSi 引脚的输入信号为 “L” 电平。</li> </ul>
中断请求的发生时序	发送中断能通过设定 U0C1 ~ U6C1 寄存器和 U78CON 寄存器的 UiIRS 位, 选择以下的任意条件: <ul style="list-style-type: none"> <li>UiIRS 位为 “0” (UiTB 寄存器空): 在将数据从 UiTB 寄存器传送到 UARTi 发送寄存器时 (开始发送时)。</li> <li>UiIRS 位为 “1” (发送结束): 在 UARTi 发送寄存器的数据发送结束时。</li> </ul> 接收中断 <ul style="list-style-type: none"> <li>在将数据从 UARTi 接收寄存器传送到 UiRB 寄存器时 (接收结束时)。</li> </ul>
错误检测	溢出错误 (注 2) 如果在读 UiRB 寄存器前接收到下一个数据的第 7 位, 就发生溢出错误。
其他选择项目	<ul style="list-style-type: none"> <li>CLK 极性选择 发送 / 接收数据的输出 / 输入时序能选择与发送 / 接收时钟的上升沿或者下降沿同步。</li> <li>位序选择 能选择 LSB first 或者 MSB first。</li> <li>连续接收模式的选择 能在读 UiRB 寄存器的同时设定为接收允许状态。</li> <li>串行数据的逻辑转换 (UART0 ~ UART6) 能将发送 / 接收数据进行逻辑取反。</li> </ul>

注 1. 在选择外部时钟时, 必须满足以下的条件:

当 UiC0 寄存器的 CKPOL 位为 “0” (与发送 / 接收时钟的下降沿同步输出发送数据, 与上升沿同步输入接收数据) 时, CLKi 引脚为 “H” 电平状态; 当 CKPOL 位为 “1” (与发送 / 接收时钟的上升沿同步输出发送数据, 与下降沿同步输入接收数据) 时, CLKi 引脚为 “L” 电平状态。

注 2. 如果发生溢出错误, UiRB 寄存器就为不定值, 而且 SiRIC 寄存器的 IR 位不变为 “1” (有中断请求)。



使用的寄存器及其设定值如表 18.3 和表 18.4 所示。另外，从选择 UART<sub>i</sub> (i=0 ~ 8) 的运行模式后到开始发送前，TXD<sub>i</sub> 引脚输出“H”电平（在选择 N 沟道漏极开路输出时，为高阻抗）。

时钟同步串行接口模式的发送和接收的运行例子分别如图 18.20 和图 18.21 所示。

表 18.3 时钟同步串行接口模式中使用的寄存器及其设定值 (UART0 ~ UART6)

寄存器	位	功能
UiMR	7 ~ 4	必须置“0000b”。
	CKDIR	必须选择内部时钟或者外部时钟。
	SMD2 ~ SMD0	必须置“001b”。
UiC0	UFORM	必须选择 LSB first 或者 MSB first。
	CKPOL	必须选择发送 / 接收时钟的极性。
	5	必须置“0”。
	CRD	必须选择允许或者禁止 CTS 功能。
	TXEPT	发送寄存器的空标志
	2	必须置“0”。
	CLK1 ~ CLK0	必须选择 UiBRG 寄存器的计数源。
UiC1	7	必须置“0”。
	UiLCH	在使用数据逻辑反相电路时，必须置“1”。
	UiRRM	在使用连续接收模式时，必须置“1”。
	UiIRS	必须选择 UART <sub>i</sub> 发送中断源。
	RI	接收结束标志
	RE	在允许接收时，必须置“1”。
	TI	发送缓冲器的空标志
	TE	在允许发送和接收时，必须置“1”。
UiSMR	7 ~ 0	必须置“00h”。
UiSMR2	7 ~ 0	必须置“00h”。
UiSMR3	7 ~ 0	必须置“00h”。
UiSMR4	7 ~ 0	必须置“00h”。
UiBRG	7 ~ 0	必须设定速率。
IFS0	IFS06	必须选择 CLK3、RXD3、 $\overline{\text{CTS3}}$ 的输入引脚。
	IFS03 ~ IFS02	必须选择 CLK6、RXD6、 $\overline{\text{CTS6}}$ 的输入引脚。
UiTB	7 ~ 0	必须设定发送数据。
UiRB	OER	溢出错误标志
	7 ~ 0	能读接收数据。

i=0 ~ 6

表 18.4 时钟同步串行接口模式中使用的寄存器及其设定值 (UART7 和 UART8)

寄存器	位	功能
UiMR	7 ~ 4	必须置“0000b”。
	CKDIR	必须选择内部时钟或者外部时钟。
	SMD2 ~ SMD0	必须置“001b”。
UiC0	UFORM	必须选择 LSB first 或者 MSB first。
	CKPOL	必须选择发送 / 接收时钟的极性。
	5	必须置“0”。
	CRD	必须选择允许或者禁止 CTS 功能。
	TXEPT	发送寄存器的空标志
	2	必须置“0”。
	CLK1 ~ CLK0	必须选择 UiBRG 寄存器的计数源。
UiC1	RI	接收结束标志
	RE	在允许接收时, 必须置“1”。
	TI	发送缓冲器的空标志
	TE	在允许发送和接收时, 必须置“1”。
U78CON	UiRRM	在使用连续接收模式时, 必须置“1”。
	UiIRS	必须选择 UARTi 发送中断源。
IFS0	IFS05	必须选择 CLK7、RXD7、 $\overline{\text{CTS7}}$ 的输入引脚。
	IFS04	必须选择 CLK8、RXD8、 $\overline{\text{CTS8}}$ 的输入引脚。
UiBRG	7 ~ 0	必须设定速率。
UiTB	7 ~ 0	必须设定发送数据。
UiRB	OER	溢出错误标志
	7 ~ 0	能读接收数据。

i=7,8

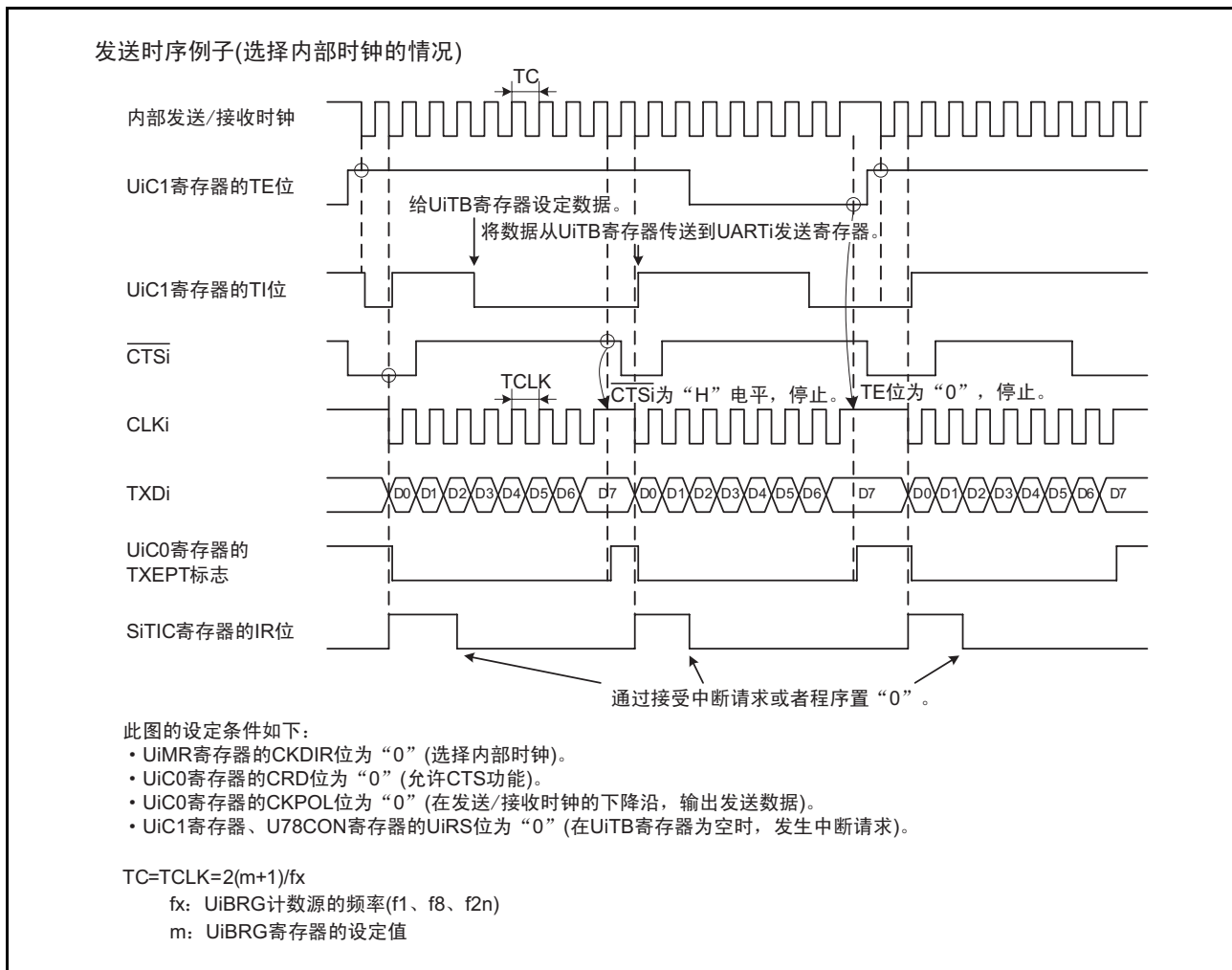


图 18.20 时钟同步串行接口模式的发送运行例子

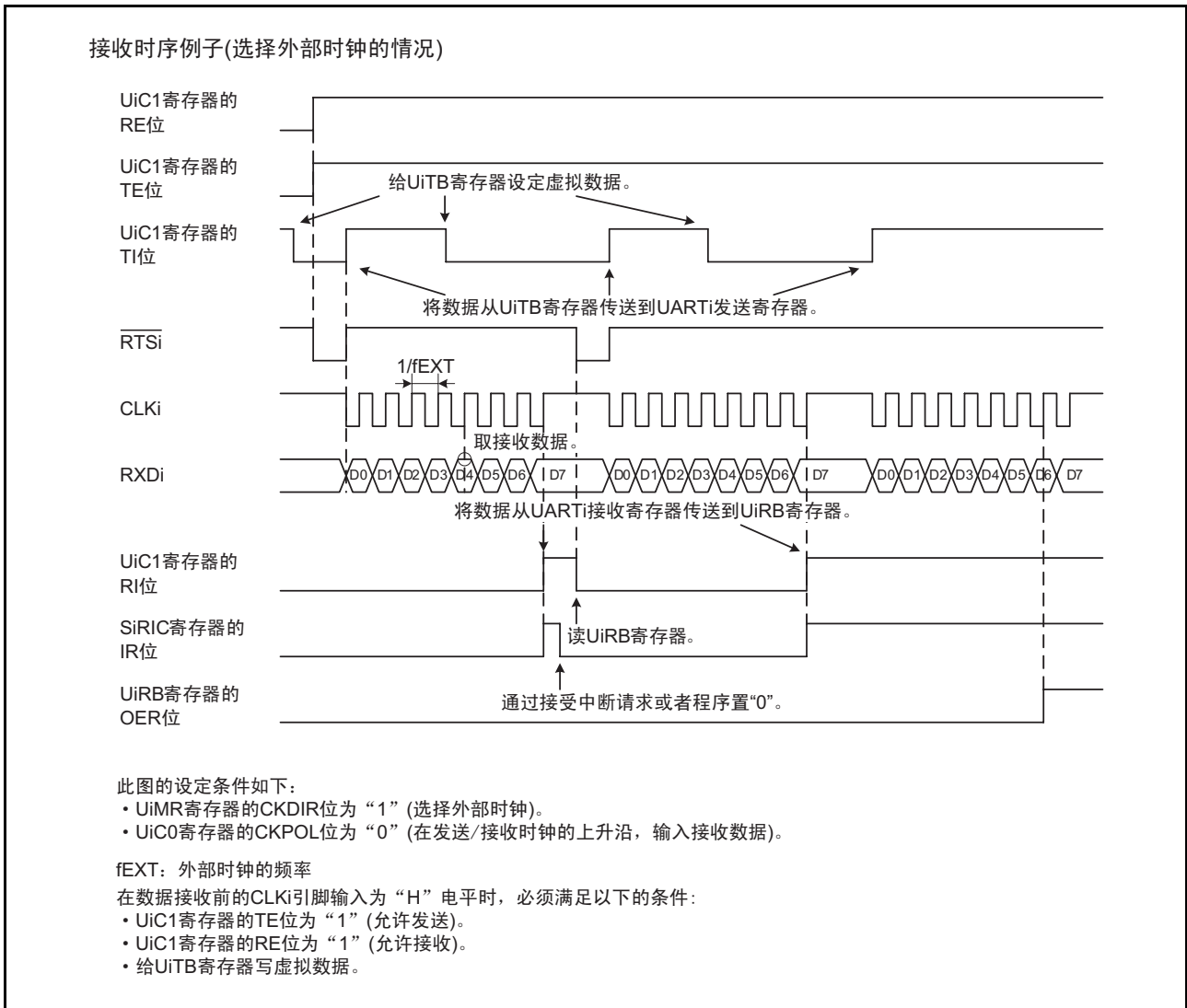


图 18.21 时钟同步串行接口模式的接收运行例子

### 18.1.1 发生通信错误时的处理方法

如果在时钟同步串行接口模式的接收或者发送时发生通信错误，就必须按照以下的步骤重新进行设定：

#### A. UiRB 寄存器 (i=0~8) 的初始化步骤

- (1) 将 UiC1 寄存器的 RE 位置 “0” (禁止接收)。
- (2) 将 UiMR 寄存器的 SMD2~SMD0 位置 “000b” (串行接口无效)。
- (3) 将 UiMR 寄存器的 SMD2~SMD0 位置 “001b” (时钟同步串行接口模式)。
- (4) 将 UiC1 寄存器的 RE 位置 “1” (允许接收)。

#### B. UiTB 寄存器的初始化步骤

- (1) 将 UiMR 寄存器的 SMD2~SMD0 位置 “000b” (串行接口无效)。
- (2) 将 UiMR 寄存器的 SMD2~SMD0 位置 “001b” (时钟同步串行接口模式)。
- (3) 与 UiC1 寄存器的 TE 位的值无关，给 TE 位写 “1” (允许发送)。

### 18.1.2 CLK 极性选择

如图 18.22 所示，能通过 UiC0 寄存器 (i=0~8) 的 CKPOL 位选择发送 / 接收时钟的极性。

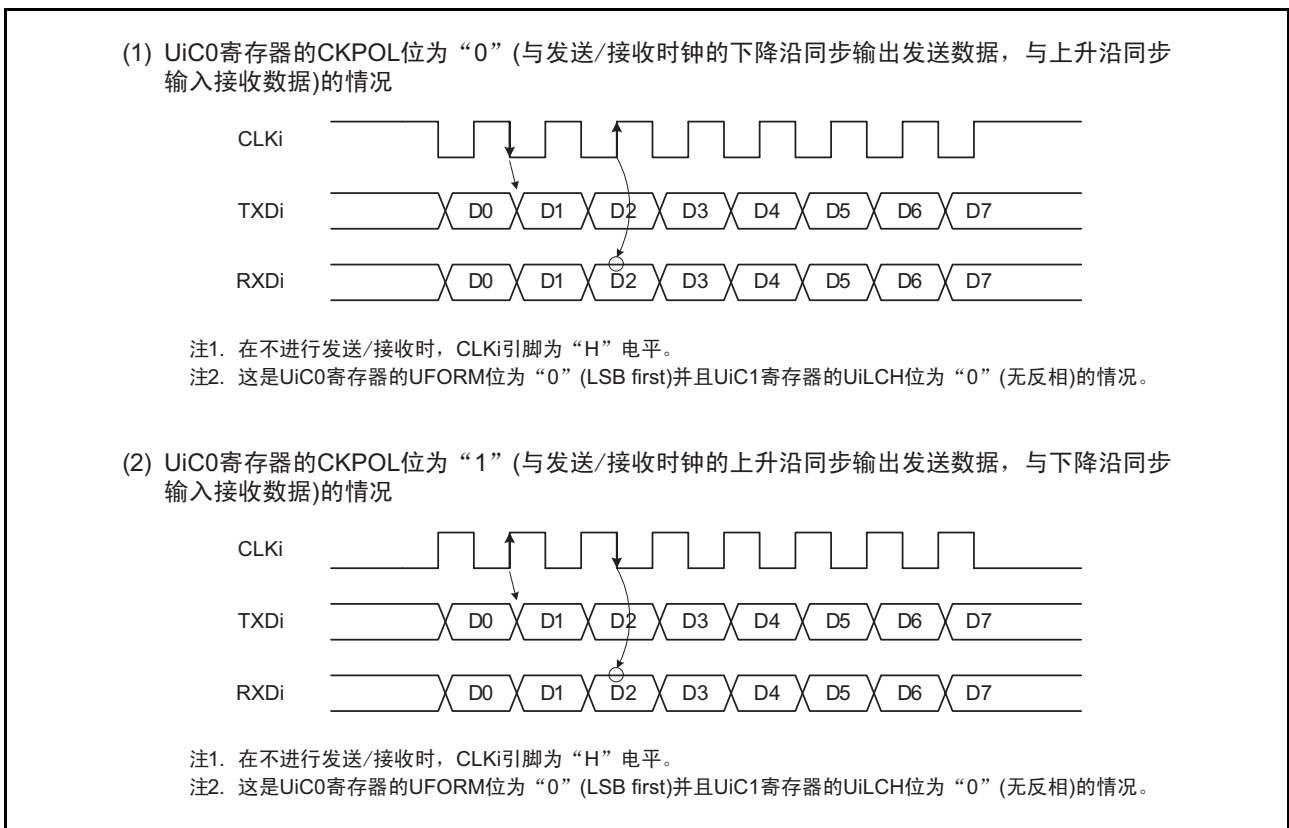


图 18.22 发送 / 接收时钟的极性 (i=0~8)

### 18.1.3 LSB first 或者 MSB first 的选择

如图 18.23 所示，能通过 UiC0 寄存器 (i=0 ~ 8) 的 UFORM 位选择位序。

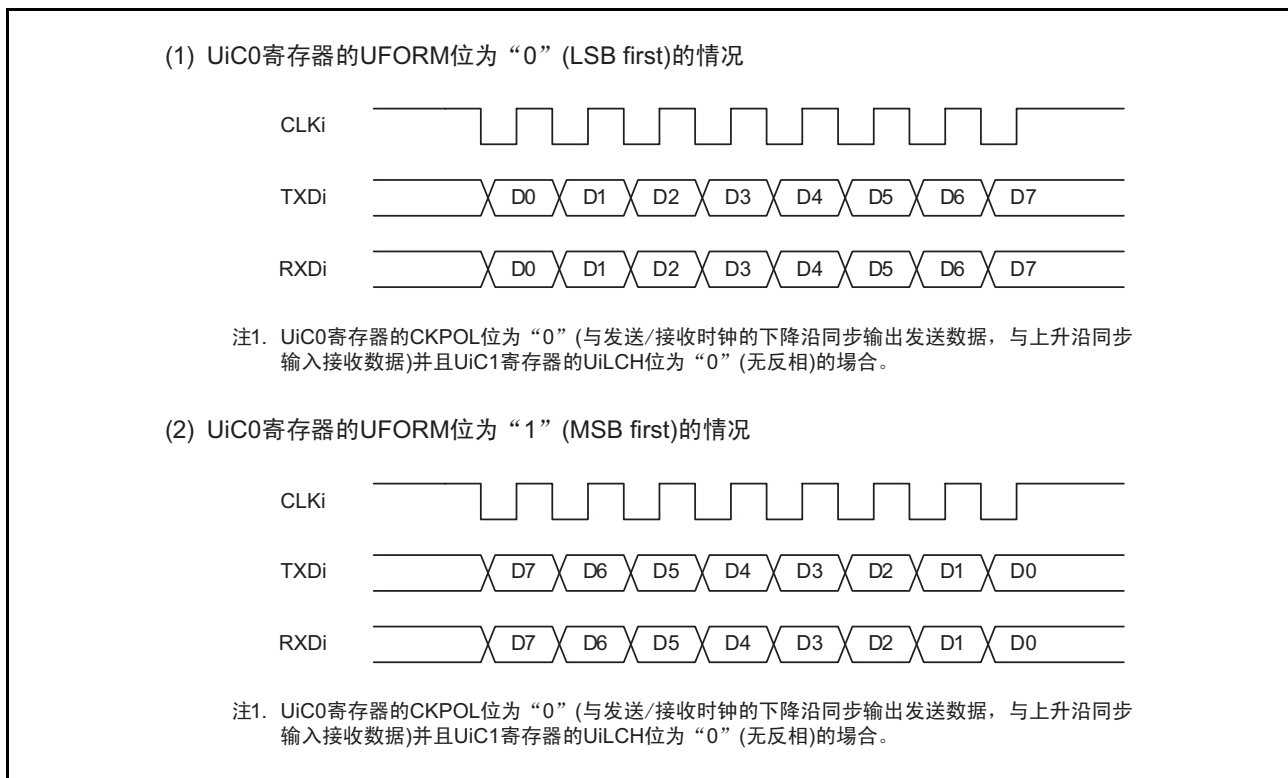


图 18.23 位序 (i=0 ~ 8)

### 18.1.4 连续接收模式

连续接收模式是通过读接收缓冲寄存器并自动进入接收允许状态的模式。如果选择此模式，就不需要为进入接收允许状态而给发送缓冲寄存器写虚拟数据。但是，在开始接收时，需要虚读接收缓冲寄存器。

如果将 U0C1 ~ U6C1 寄存器和 U78CON 寄存器的 UiRRM 位 (i=0 ~ 8) 置“1”(连续接收模式)，就在读 UiRB 寄存器时，UiC1 寄存器的 TI 位变为“0”(UiTB 寄存器有数据)。当 UiRRM 位为“1”时，不能给 UiTB 寄存器写虚拟数据。

### 18.1.5 串行数据的逻辑转换

当  $U_iC1$  寄存器 ( $i=0 \sim 6$ ) 的  $U_iLCH$  位为 “1” (有反相) 时, 在发送时将  $U_iTB$  寄存器的值进行逻辑取反后发送。如果读  $U_iRB$  寄存器, 就能读到接收数据被逻辑取反后的值。串行数据的逻辑如图 18.24 所示。

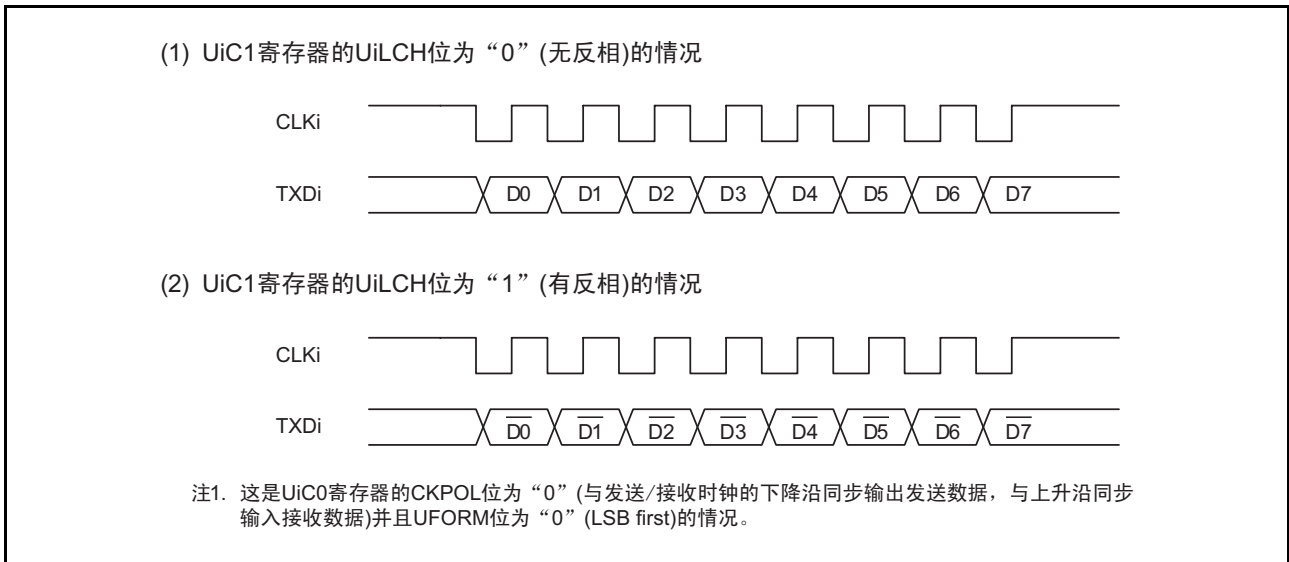


图 18.24 串行数据的逻辑 ( $i=0 \sim 6$ )

### 18.1.6 CTS/RTS 功能

CTS 功能是使用  $\overline{CTS_i/RTS_i}$  引脚 ( $i=0 \sim 8$ ) 进行发送控制的功能。如果  $\overline{CTS_i/RTS_i}$  引脚的输入电平变为 “L”, 就开始发送。如果在发送过程中将输入电平置为 “H”, 就停止下一个数据的发送。

在时钟同步串行接口模式中, 在接收时也需要让发送电路工作, 如果 CTS 功能有效, 即使在开始接收时也需要将  $\overline{CTS_i/RTS_i}$  引脚的输入电平置为 “L”。

RTS 功能是使用  $\overline{CTS_i/RTS_i}$  引脚表示接收电路状态的功能。在接收的准备结束时,  $\overline{CTS_i/RTS_i}$  引脚的输出电平变为 “L”。在 CLK<sub>i</sub> 引脚的第一个下降沿, 输出电平变为 “H”。

## 18.2 异步串行接口模式（UART 模式）

UART 模式是以开始位的下降沿为触发信号，使内部时钟与数据同步进行发送和接收的模式。UART 模式的规格如表 18.5 所示。

表 18.5 UART 模式的规格

项目	规格
数据格式	<ul style="list-style-type: none"> <li>开始位 1 位</li> <li>数据位（数据字符） 可选择 7 位、8 位或者 9 位。</li> <li>奇偶校验位 可选择奇校验、偶校验或者无奇偶校验。</li> <li>停止位 可选择 1 位或者 2 位。</li> </ul>
发送 / 接收时钟	<ul style="list-style-type: none"> <li>当 UiMR 寄存器（i=0 ~ 8）的 CKDIR 位为“0”（内部时钟）时：  <math display="block">\frac{f_x}{16(m+1)}</math> <math>f_x</math>: f1、f8、f2n      <math>m</math>: UiBRG 寄存器的设定值（00h ~ FFh）</li> <li>当 CKDIR 位为“1”（外部时钟）时：  <math display="block">\frac{f_{EXT}}{16(m+1)}</math> <math>f_{EXT}</math>: CLKi 引脚的输入时钟</li> </ul>
发送 / 接收控制	选择 CTS 功能、RTS 功能或者 CTS/RTS 功能无效。
发送开始条件	开始发送时需要以下的条件： <ul style="list-style-type: none"> <li>UiC1 寄存器的 TE 位为“1”（允许发送）。</li> <li>UiC1 寄存器的 TI 位为“0”（UiTB 寄存器有数据）。</li> <li>在选择 CTS 功能时，给 CTSi 引脚输入“L”电平。</li> </ul>
接收开始条件	开始接收时需要以下的条件： <ul style="list-style-type: none"> <li>UiC1 寄存器的 RE 位为“1”（允许接收）。</li> <li>检测到开始位。</li> </ul>
中断请求的发生时序	发送中断能通过设定 U0C1 ~ U6C1 寄存器和 U78CON 寄存器的 UiIRS 位，选择以下的任意条件： <ul style="list-style-type: none"> <li>UiIRS 位为“0”（UiTB 寄存器空）： 在将数据从 UiTB 寄存器传送到 UARTi 发送寄存器时（开始发送时）。</li> <li>UiIRS 位为“1”（发送结束）： 在 UARTi 发送寄存器的数据发送结束时。</li> </ul> 接收中断 <ul style="list-style-type: none"> <li>在将数据从 UARTi 接收寄存器传送到 UiRB 寄存器时（接收结束时）。</li> </ul>
错误检测	<ul style="list-style-type: none"> <li>溢出错误（注 1） 如果在读 UiRB 寄存器前接收到下一个数据的最后停止位的前 1 位（在选择 2 个停止位时，为第 1 个停止位），就发生溢出错误。</li> <li>帧错误 当未检测到设定个数的停止位时，发生帧错误。</li> <li>奇偶校验错误 在有奇偶校验的情况下，如果接收到的数据字符和奇偶校验位中的“1”的个数（奇 / 偶）不是设定的个数（奇 / 偶），就发生奇偶校验错误。</li> <li>错误和标志 如果发生溢出错误、帧错误或者奇偶校验错误，此标志就变为“1”。</li> </ul>
其他选择项目	<ul style="list-style-type: none"> <li>位序选择 能选择 LSB first 或者 MSB first。</li> <li>串行数据的逻辑转换 这是将发送 / 接收数据进行逻辑取反的功能，开始位和停止位不取反。</li> <li>TXD、RXD 输入 / 输出极性的转换 这是将 TXD 引脚的输出电平和 RXD 引脚的输入电平取反的功能。输入 / 输出信号的电平全部反相。</li> </ul>

注 1. 如果发生溢出错误，UiRB 寄存器就为不定值，而且 SiRIC 寄存器的 IR 位不变为“1”（有中断请求）。



使用的寄存器及其设定值如表 18.6 和表 18.7 所示。另外，从选择 UART<sub>i</sub> (i=0 ~ 8) 运行模式后到开始发送前，TXD<sub>i</sub> 引脚输出“H”电平（在选择 N 沟道漏极开路输出时，为高阻抗）。

UART 模式的发送运行例子如图 18.25 和图 18.26 所示，UART 模式的接收运行例子如图 18.27 所示。

表 18.6 UART 模式中使用的寄存器及其设定值 (UART0 ~ UART6)

寄存器	位	功能	
UiMR	IOPOL	必须选择 TXD 引脚和 RXD 引脚的输入 / 输出极性。	
	PRY、PRYE	必须选择偶校验、奇校验或者无奇偶校验。	
	STPS	必须选择停止位长。	
	CKDIR	必须选择内部时钟或者外部时钟。	
	SMD2 ~ SMD0		当字符长为 7 位时，必须置“100b”。
			当字符长为 8 位时，必须置“101b”。
		当字符长为 9 位时，必须置“110b”。	
UiC0	UFORM	当字符长为 8 位时，能选择 LSB first 或者 MSB first；当字符长为 7 位或者 9 位时，必须置“0”。	
	CKPOL	必须置“0”。	
	5	必须置“0”。	
	CRD	必须选择允许或者禁止 CTS 功能。	
	TXEPT	发送寄存器的空标志	
	2	必须置“0”。	
	CLK1 ~ CLK0	必须选择 UiBRG 寄存器的计数源。	
UiC1	7	必须置“0”。	
	UiLCH	在使用数据逻辑反相电路时，必须置“1”。	
	UiRRM	必须置“0”。	
	UiIRS	必须选择 UART <sub>i</sub> 发送中断源。	
	RI	接收结束标志	
	RE	在允许接收时，必须置“1”。	
	TI	发送缓冲器的空标志	
	TE	在允许发送时，必须置“1”。	
UiSMR	7 ~ 0	必须置“00h”。	
UiSMR2	7 ~ 0	必须置“00h”。	
UiSMR3	7 ~ 0	必须置“00h”。	
UiSMR4	7 ~ 0	必须置“00h”。	
UiBRG	7 ~ 0	必须设定速率。	
IFS0	IFS06	必须选择 CLK3、RXD3、CTS3 的输入引脚。	
	IFS03 ~ IFS02	必须选择 CLK6、RXD6、CTS6 的输入引脚。	
UiTB	8 ~ 0	必须设定发送数据（注 1）。	
UiRB	OER、FER、PER、SUM	错误标志	
	8 ~ 0	能读接收数据（注 1）。	

i=0 ~ 6

注 1. 使用的位如下：

字符长为 7 位：bit6 ~ bit0

字符长为 8 位：bit7 ~ bit0

字符长为 9 位：bit8 ~ bit0

表 18.7 UART 模式中使用的寄存器及其设定值 (UART7 和 UART8)

寄存器	位	功能
UiMR	PRY、PRYE	必须选择偶校验、奇校验或者无奇偶校验。
	STPS	必须选择停止位长。
	CKDIR	必须选择内部时钟或者外部时钟。
	SMD2 ~ SMD0	当字符长为 7 位时, 必须置 “100b”。 当字符长为 8 位时, 必须置 “101b”。 当字符长为 9 位时, 必须置 “110b”。
UiC0	UFORM	当字符长为 8 位时, 能选择 LSB first 或者 MSB first; 当字符长为 7 位或者 9 位时, 必须置 “0”。
	CKPOL	必须置 “0”。
	5	必须置 “0”。
	CRD	必须选择允许或者禁止 CTS 功能。
	TXEPT	发送寄存器的空标志
	2	必须置 “0”。
	CLK1 ~ CLK0	必须选择 UiBRG 寄存器的计数源。
UiC1	RI	接收结束标志
	RE	在允许接收时, 必须置 “1”。
	TI	发送缓冲器的空标志
	TE	在允许发送时, 必须置 “1”。
U78CON	UiRRM	必须置 “0”。
	UiIRS	必须选择 UARTi 发送中断源。
UiBRG	7 ~ 0	必须设定速率。
IFS0	IFS05	必须选择 CLK7、RXD7、 $\overline{\text{CTS7}}$ 的输入引脚。
	IFS04	必须选择 CLK8、RXD8、 $\overline{\text{CTS8}}$ 的输入引脚。
UiTB	8 ~ 0	必须设定发送数据 (注 1)。
UiRB	OER、FER、PER、SUM	错误标志
	8 ~ 0	能读接收数据 (注 1)。

i=7,8

注 1. 使用的位如下:

字符长为 7 位: bit6 ~ bit0

字符长为 8 位: bit7 ~ bit0

字符长为 9 位: bit8 ~ bit0

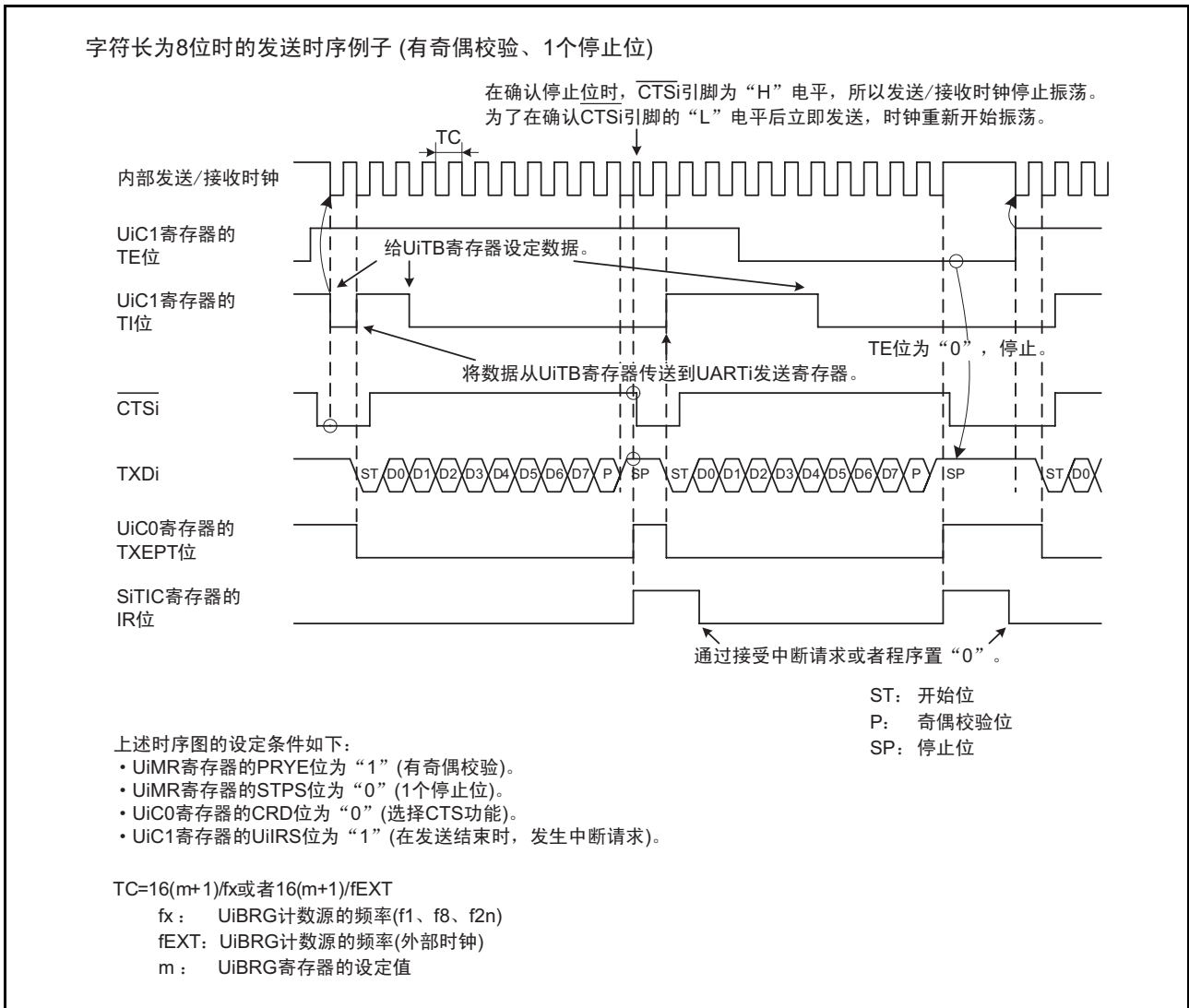


图 18.25 UART 模式的发送运行例子 (1) ( $i=0 \sim 8$ )

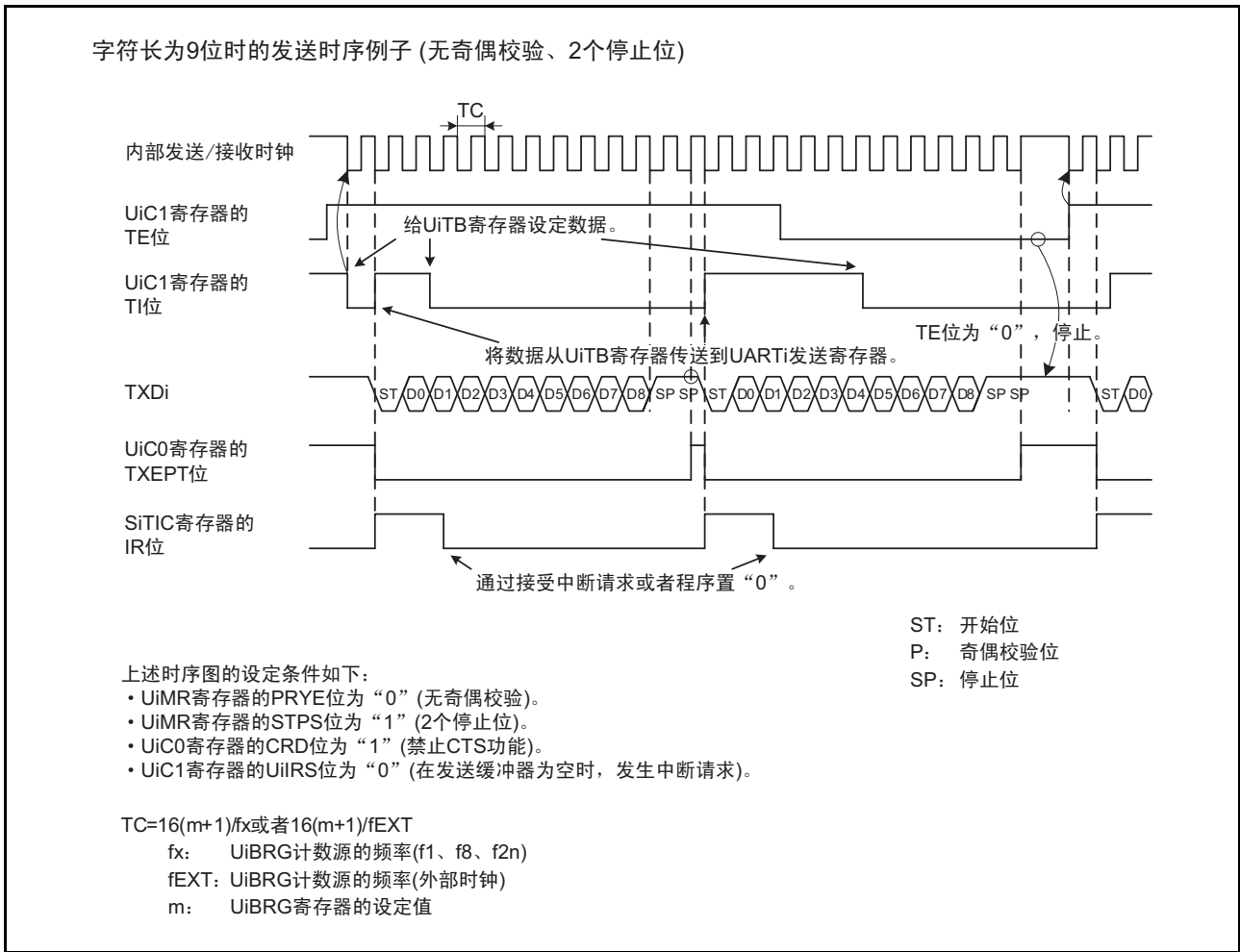


图 18.26 UART 模式的发送运行例子 (2) (i=0 ~ 8)

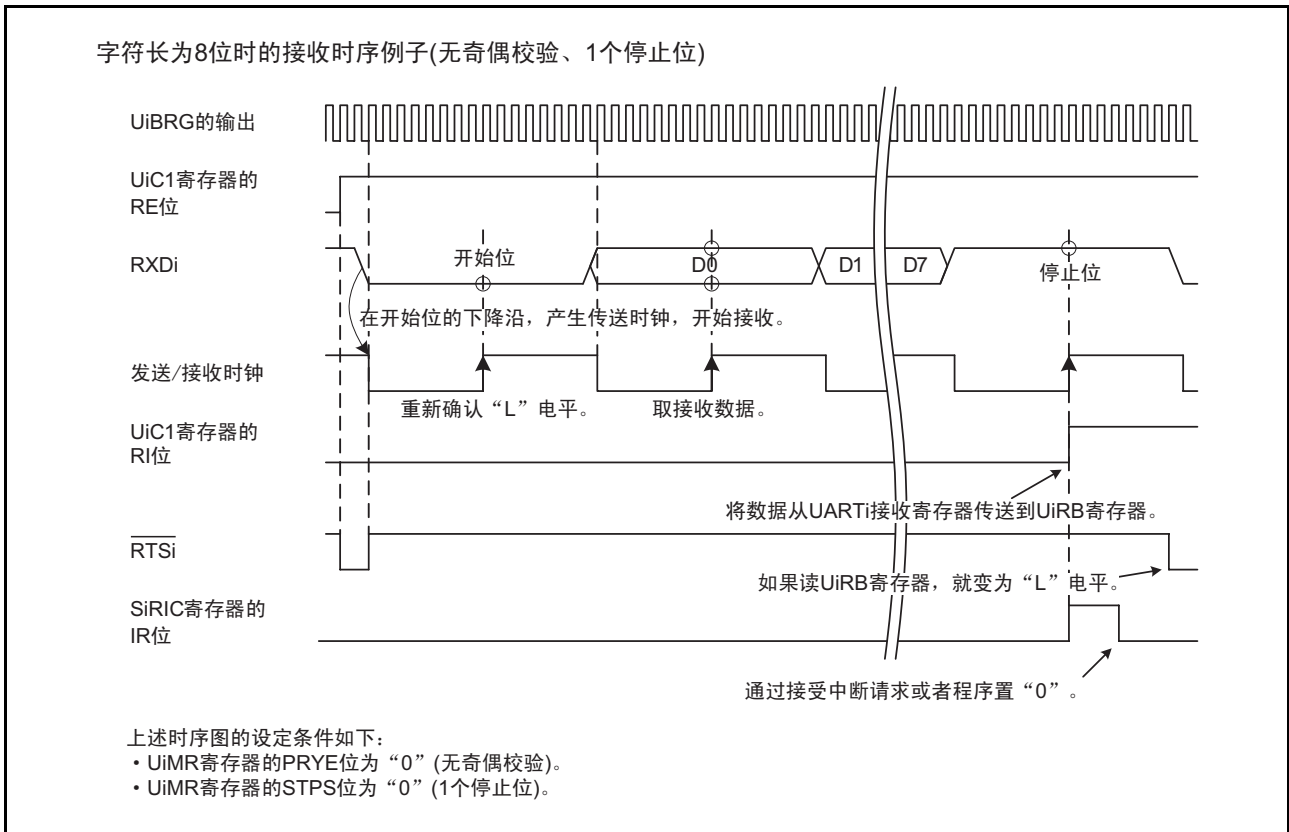


图 18.27 UART 模式的接收运行例子 (i=0 ~ 8)

### 18.2.1 位速率

在 UART 模式中，位速率是通过 UiBRG 寄存器 (i=0 ~ 8) 将计数源进行 16 分频后的频率，位速率的设定例子如表 18.8 所示。

表 18.8 位速率的设定例子

位速率 (bps)	BRG 的 计数源	外围功能时钟: 30MHz		外围功能时钟: 32MHz	
		BRG 的设定值: n	实际时间 (bps)	BRG 的设定值: n	实际时间 (bps)
1200	f8	194 (C2h)	1202	207 (CFh)	1202
2400	f8	97 (61h)	2392	103 (67h)	2404
4800	f8	48 (30h)	4783	51 (33h)	4808
9600	f1	194 (C2h)	9615	207 (CFh)	9615
14400	f1	129 (81h)	14423	138 (8Ah)	14388
19200	f1	97 (61h)	19133	103 (67h)	19231
28800	f1	64 (40h)	28846	68 (44h)	28986
31250	f1	59 (3Bh)	31250	63 (3Fh)	31250
38400	f1	48 (30h)	38265	51 (33h)	38462
51200	f1	36 (24h)	50676	38 (26h)	51282

### 18.2.2 发生通信错误时的处理方法

在 UART 模式中，如果在接收或者发送时发生通信错误，就必须按照以下的步骤进行重新设定：

#### A. UiRB 寄存器 (i=0~8) 的初始化步骤

- (1) 将 UiC1 寄存器的 RE 位置 “0” (禁止接收)。
- (2) 将 UiC1 寄存器的 RE 位置 “1” (允许接收)。

#### B. UiTB 寄存器的初始化步骤

- (1) 将 UiMR 寄存器的 SMD2~SMD0 位置 “000b” (串行接口无效)。
- (2) 重新设定 UiMR 寄存器的 SMD2~SMD0 位 (“001b”、“101b”、“110b”)。
- (3) 与 UiC1 寄存器的 TE 位的值无关，给 TE 位写 “1” (允许发送)。

### 18.2.3 LSB first 或者 MSB first 的选择

如图 18.28 所示，能通过 UiC0 寄存器 (i=0~8) 的 UFORM 位选择位序。在字符长为 8 位时使用此功能。

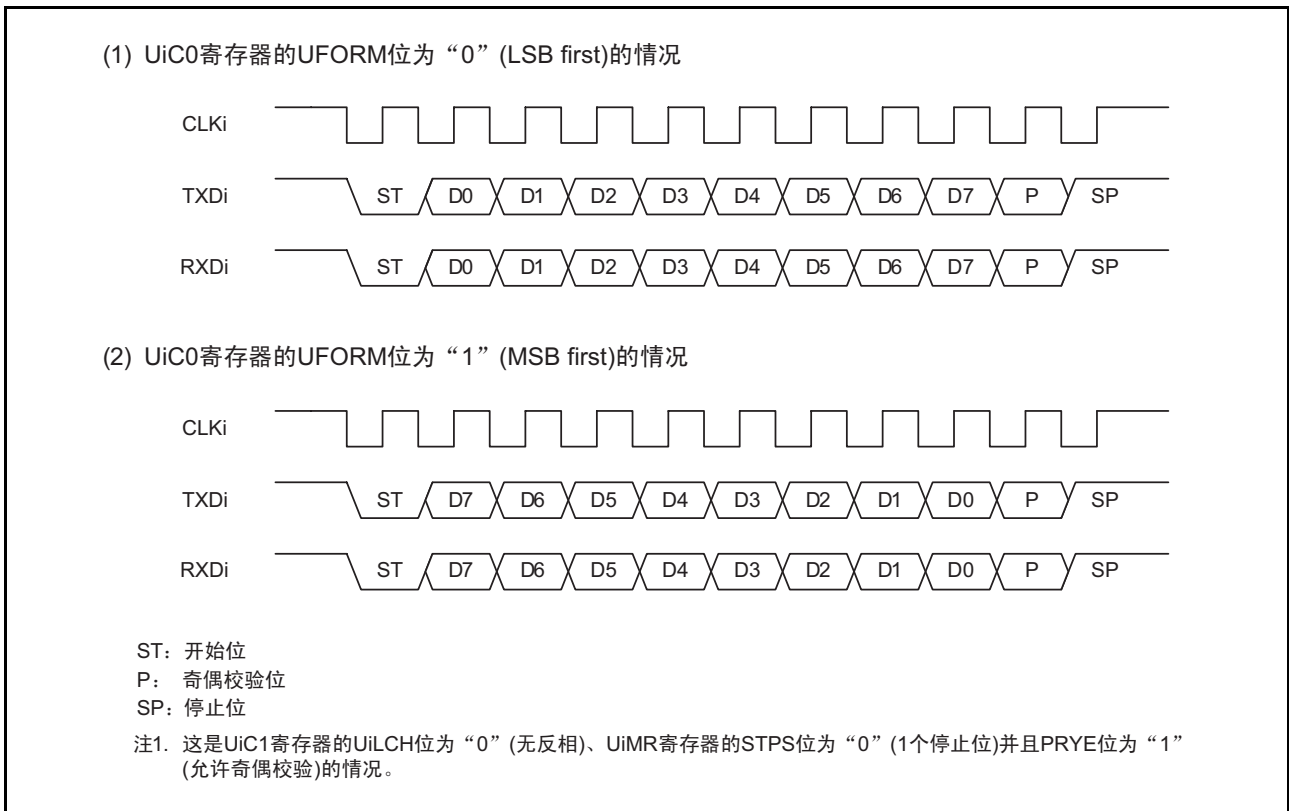


图 18.28 位序 (i=0~8)

### 18.2.4 串行数据的逻辑转换

如果将 UiC1 寄存器 (i=0 ~ 6) 的 UiLCH 位置 “1” (有反相), 就在写 UiTB 寄存器以及读 UiRB 寄存器时将数据进行逻辑取反。奇偶校验位不取反。

串行数据的逻辑如图 18.29 所示。

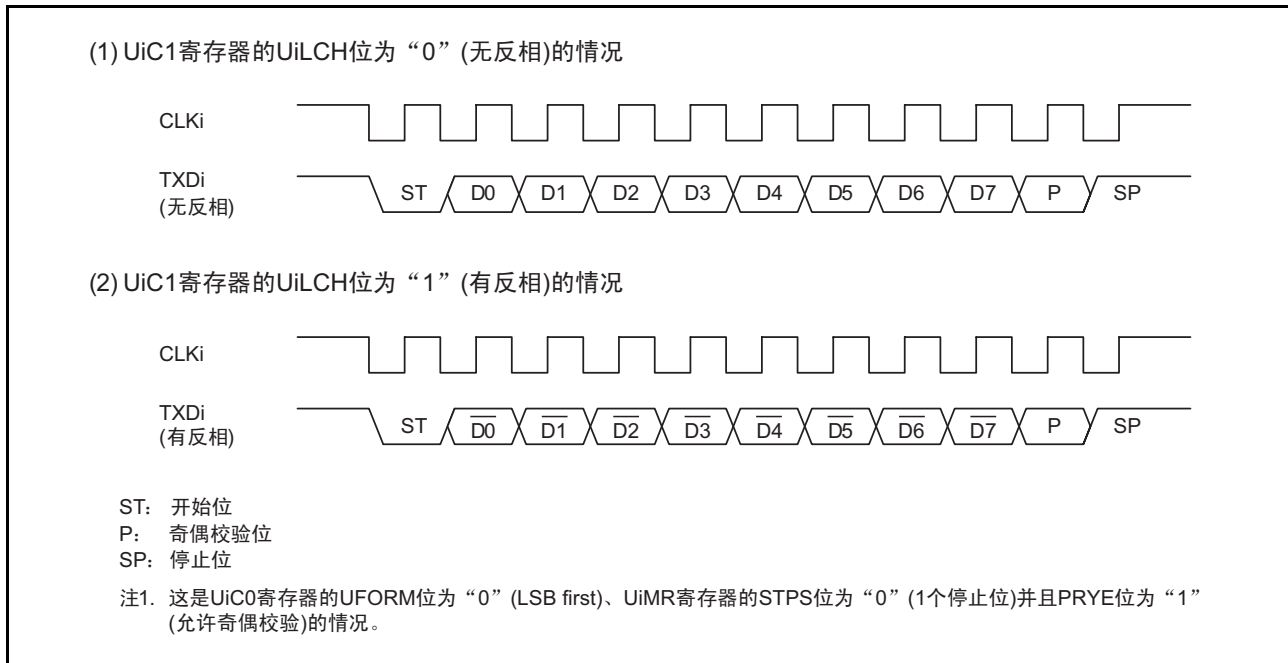


图 18.29 串行数据的逻辑 (i=0 ~ 6)

### 18.2.5 TXD 和 RXD 的输入 / 输出极性转换

这是将 TXD 引脚的输出电平和 RXD 引脚的输入电平取反的功能。如果将 UiMR 寄存器 (i=0 ~ 6) 的 IOPOL 位置 “1” (有反相), 输入 / 输出数据的电平就全部反相 (包括开始位、停止位和奇偶校验位)。TXD 和 RXD 的输入 / 输出极性转换如图 18.30 所示。

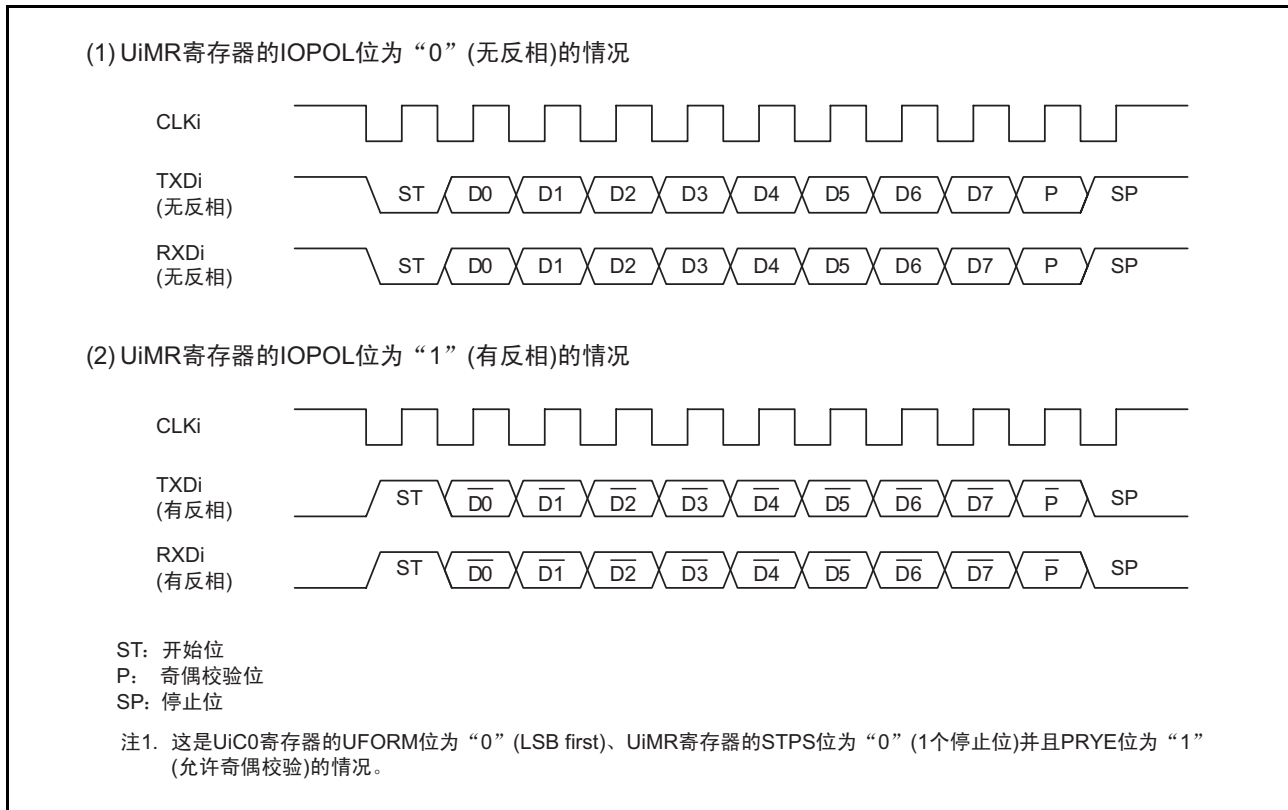


图 18.30 TXD 和 RXD 的输入 / 输出极性转换 (i=0 ~ 6)

### 18.2.6 CTS/RTS 功能

CTS 功能是使用  $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$  引脚 (i=0 ~ 8) 进行发送控制的功能。如果  $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$  引脚的输入电平变为 “L”, 就开始发送。如果在发送过程中将输入电平置为 “H”, 就停止下一个数据的发送。

RTS 功能是使用  $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$  引脚表示接收电路状态的功能。在接收的准备结束时,  $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$  引脚的输出电平变为 “L”。在 CLKi 引脚的第一个下降沿, 输出电平变为 “H”。



### 18.3 特殊模式 1 (I<sup>2</sup>C 模式)

I<sup>2</sup>C 模式是对应简易型 I<sup>2</sup>C 接口的模式，I<sup>2</sup>C 模式的规格如表 18.9 所示。

表 18.9 I<sup>2</sup>C 模式的规格

项目	规格
数据格式	字符长：8 位
发送 / 接收时钟	主控模式 <ul style="list-style-type: none"> <li>当 UiMR 寄存器 (i=0 ~ 6) 的 CKDIR 位为 “0” (内部时钟) 时：  <math display="block">\frac{f_x}{2(m+1)}</math> <math>f_x</math>: f1、f8、f2n      <math>m</math>: UiBRG 寄存器的设定值 (00h ~ FFh)</li> </ul> 从属模式 <ul style="list-style-type: none"> <li>当 CKDIR 位为 “1” (外部时钟) 时: SCLi 引脚的输入</li> </ul>
发送开始条件	开始发送时需要以下的条件 (注 1): <ul style="list-style-type: none"> <li>UiC1 寄存器的 TE 位为 “1” (允许发送)。</li> <li>UiC1 寄存器的 TI 位为 “0” (UiTB 寄存器有数据)。</li> </ul>
接收开始条件	开始接收时需要以下的条件 (注 1): <ul style="list-style-type: none"> <li>UiC1 寄存器的 RE 位为 “1” (允许接收)。</li> <li>TE 位为 “1” (允许发送)。</li> <li>TI 位为 “0” (UiTB 寄存器有数据)。</li> </ul>
中断请求的发生时序	检测到开始条件、停止条件、ACK (Acknowledge) 或者 NACK (Not-Acknowledge)。
错误检测	溢出错误 (注 2) 如果在读 UiRB 寄存器前接收到下一个数据的第 8 位, 就发生溢出错误。
其他选择项目	<ul style="list-style-type: none"> <li>仲裁失败 可选择 UiRB 寄存器的 ABT 位的更新时序。</li> <li>SDAi 数字延迟 可选择无数字延迟或者 2 ~ 8 个 UiBRG 计数源周期的延迟。</li> <li>时钟相位的设定 可选择有时钟延迟或者无时钟延迟。</li> </ul>

注 1. 在已选择外部时钟时, 必须在外部时钟为 “H” 的状态下满足条件。

注 2. 如果发生溢出错误, UiRB 寄存器就为不定值, 而且 SiRIC 寄存器的 IR 位不变。

I<sup>2</sup>C 模式中使用的寄存器及其设定值如表 18.10 所示, I<sup>2</sup>C 模式的各功能和框图分别如表 18.11 和图 18.31 所示, UiRB 寄存器 (i=0 ~ 6) 的传送和中断时序如图 18.32 所示。

如表 18.11 所示, 如果将 UiMR 寄存器 (i=0 ~ 6) 的 SMD2 ~ SMD0 位置 “010b” 并且将 UiSMR 寄存器的 IICM 位置 “1”, 就变为 I<sup>2</sup>C 模式。因为 SDAi 的发送输出附带延迟电路, 所以在 SCLi 变为 “L” 电平并稳定后, SDAi 的输出发生变化。

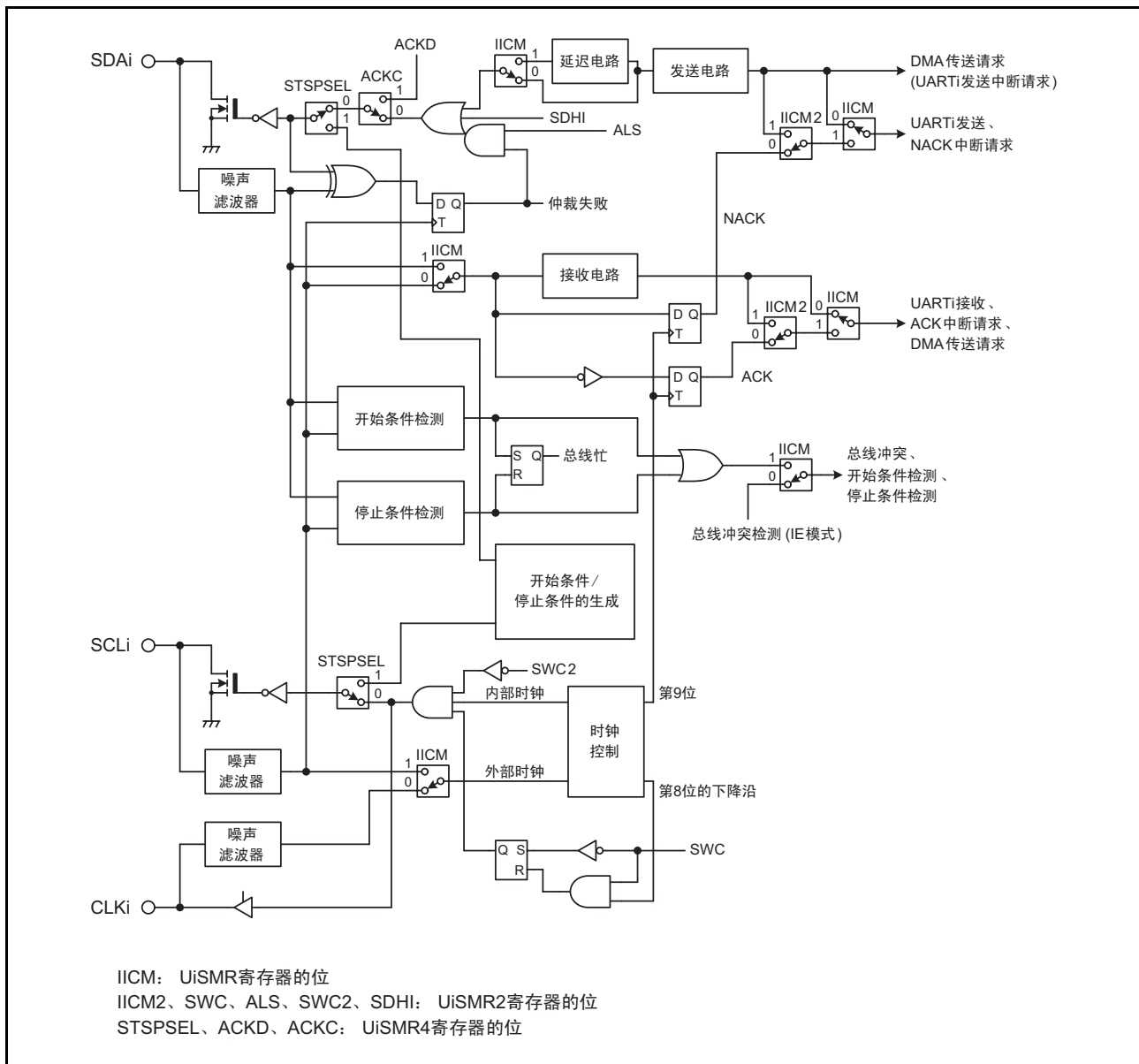


图 18.31 I<sup>2</sup>C 模式的框图 (i=0 ~ 6)

表 18.10 I<sup>2</sup>C 模式中使用的寄存器及其设定值 (i=0 ~ 6)

寄存器	位	功能	
		主控模式	从属模式
UiMR	IOPOL	必须置“0”。	
	CKDIR	必须置“0”。	必须置“1”。
	SMD2 ~ SMD0	必须置“010b”。	
UiC0	7 ~ 4	必须置“1001b”。	
	TXEPT	发送寄存器的空标志	
	2	必须置“0”。	
	CLK1 ~ CLK0	必须选择 UiBRG 的计数源。	无效
UiC1	7 ~ 5	必须置“000b”。	
	UiIRS	必须置“1”。	
	RI	接收结束标志	
	RE	在允许接收时，必须置“1”。	
	TI	发送缓冲器的空标志	
	TE	在允许发送和接收时，必须置“1”。	
	UiSMR	7 ~ 3	必须置“00000b”。
BBS		总线忙标志	
ABC		必须选择仲裁失败检测时序。	无效
IICM		必须置“1”。	
UiSMR2	7	必须置“0”。	
	SDHI	在禁止 SDA 输出时，必须置“1”。	
	SWC2	在将 SCL 输出强制设定为“L”电平时，必须置“1”。	
	STC	必须置“0”。	在检测到开始条件并对 UARTi 进行初始化时，必须置“1”。
	ALS	在检测到仲裁失败时停止 SDAi 输出的情况下，必须置“1”。	必须置“0”。
	SWC	在接收到 8 位后将 SCLi 输出固定为“L”电平时，必须置“1”。	
	CSC	在实施时钟同步时，必须置“1”。	必须置“0”。
	IICM2	参照表 18.11。	
UiSMR3	DL2 ~ DL0	必须设定 SDAi 的数字延迟值。	
	4 ~ 2	必须置“000b”。	
	CKPH	参照表 18.11。	
	SSE	必须置“0”。	
UiSMR4	SWC9	必须置“0”。	在接收到 9 位后将 SCLi 的输出固定为“L”电平时，必须置“1”。
	SCLHI	在检测到停止条件时允许 SCL 输出停止的情况下，必须置“1”。	必须置“0”。
	ACKC	在输出 ACK 数据时，必须置“1”。	
	ACKD	必须选择 ACK 或者 NACK。	
	STSPSEL	在输出各条件时，必须置“1”。	必须置“0”。
	STPREQ	在生成停止条件时，必须置“1”。	必须置“0”。
	RSTAREQ	在生成重新开始条件时，必须置“1”。	必须置“0”。
	STAREQ	在生成开始条件时，必须置“1”。	必须置“0”。
UiBRG	7 ~ 0	必须设定速率。	无效
IFSR0	IFSR06、IFSR07	必须选择使用中断的 UART。	
IFSR1	IFSR16	必须选择使用中断的 UART。	
IFS0	IFS06	必须选择 SCL3、SDA3 的输入引脚。	
	IFS03 ~ IFS02	必须选择 SCL6、SDA6 的输入引脚。	
UiTB	8	在发送时，必须设定为“1”；在接收时，必须设定 ACK 位的值。	
	7 ~ 0	在发送时，必须设定发送数据；在接收时，必须设定为“FFh”。	
UiRB	OER	溢出错误标志	
	ABT	仲裁失败检测标志	无效
	8	在发生接收中断后，保存 D0；在发生发送中断后，保存 ACK 或者 NACK。	
	7 ~ 0	在发生接收中断后，能读 D7 ~ D1；在发生发送中断后，能读 D7 ~ D0。	

表 18.11 I<sup>2</sup>C 模式的各功能 (i=0 ~ 6)

功能	时钟同步串行接口模式 (SMD2 ~ SMD0=001b, IICM=0)	I <sup>2</sup> C 模式 (SMD2 ~ SMD0=010b, IICM=1)			
		IICM2=0 (ACK/NACK 中断)		IICM2=1 (发送/接收中断)	
		CKPH=0 (无时钟延迟)	CKPH=1 (有时钟延迟)	CKPH=0 (无时钟延迟)	CKPH=1 (有时钟延迟)
中断序号 6,39 ~ 41 的源 (注 1) (参照图 18.32)	—	检测到开始条件或者停止条件 (参照图 18.12)。			
中断序号 2,4,17,19,33,35,37 的源 (注 1) (参照图 18.32)	UARTi 发送开始发送或者发送结束 (通过 UiIRS 选择)	检测到 NACK。 第 9 位的 SCLi 的上升沿	UARTi 发送第 9 位的 SCLi 的上升沿	UARTi 发送第 9 位的 SCLi 的下降沿	
中断序号 3,5,18,20,34,36,38 的源 (注 1) (参照图 18.32)	UARTi 接收在接收到第 8 位时 CKPOL=0: 上升沿 CKPOL=1: 下降沿	检测到 ACK。 第 9 位的 SCLi 的上升沿	UARTi 接收第 8 位的 SCLi 的下降沿		
从 UART 接收寄存器到 UiRB 寄存器的数据传送时序	CKPOL=0: 上升沿 CKPOL=1: 下降沿	第 9 位的 SCLi 的上升沿	第 8 位的 SCLi 的下降沿	第 8 位的 SCLi 的下降沿和第 9 位的上升沿	
UARTi 发送输出延迟	无延迟	有延迟			
P6_3、P6_7、P7_0、P7_3、P7_6、P9_2、P9_6、P11_0、P12_0、P15_0、P15_4 引脚的功能	TXDi 的输出	SDAi 的输入 / 输出			
P6_2、P6_6、P7_1、P7_5、P8_0、P9_1、P9_7、P11_2、P12_2、P15_2、P15_5 引脚的功能	RXDi 的输入	SCLi 的输入 / 输出			
P6_1、P6_5、P7_2、P7_4、P7_7、P9_0、P9_5、P11_1、P12_1、P15_1、P15_6 引脚的功能	CLKi 的输入或者输出	— (在 I <sup>2</sup> C 模式中, 不能使用)			
RXDi 引脚和 SCLi 引脚电平的读	与对应的端口方向位的内容无关, 能读。				
SDAi 输出引脚的初始值	—	H (如果通过功能选择寄存器选择端口, 就为端口寄存器的值)			
SCLi 的初始值和结束值	—	H	L	H	L
DMA 源 (参照图 18.32)	UARTi 接收	检测到 ACK。	UARTi 接收第 8 位的 SCLi 的下降沿		
接收数据的保存	将第 1 ~ 8 位保存到 UiRB 寄存器 bit0 ~ bit7。	将第 1 ~ 8 位保存到 UiRB 寄存器的 bit7 ~ bit0。	将第 1 ~ 7 位保存到 UiRB 寄存器的 bit6 ~ bit0, 将第 8 位保存到 UiRB 寄存器的 bit8。	第 1 次 (注 2) 和左边相同。第 2 次 (注 3) 将第 1 ~ 8 位保存到 UiRB 寄存器的 bit7 ~ bit0, 将第 9 位保存到 UiRB 寄存器的 bit8。	
接收数据的读	直接读 UiRB 寄存器的状态。		将 UiRB 寄存器的 bit6 ~ bit0 作为 bit7 ~ bit1, 将 bit8 作为 bit0 来读。	第 1 次 (注 2) 和左边相同。第 2 次 (注 3) 直接读 UiRB 寄存器的状态。	

注 1. 必须按照以下的步骤转换中断源:

- (1) 禁止与中断序号对应的中断。
- (2) 转换中断源。
- (3) 将对应的中断序号的 IR 位置 “0” (无中断)。
- (4) 设定与中断序号对应的 ILVL2 ~ ILVL0。

注 2. 在第 8 位的 SCLi 下降时, 第 1 次将数据传送到 UiRB 寄存器。

注 3. 在第 9 位的 SCLi 上升时, 第 2 次将数据传送到 UiRB 寄存器。

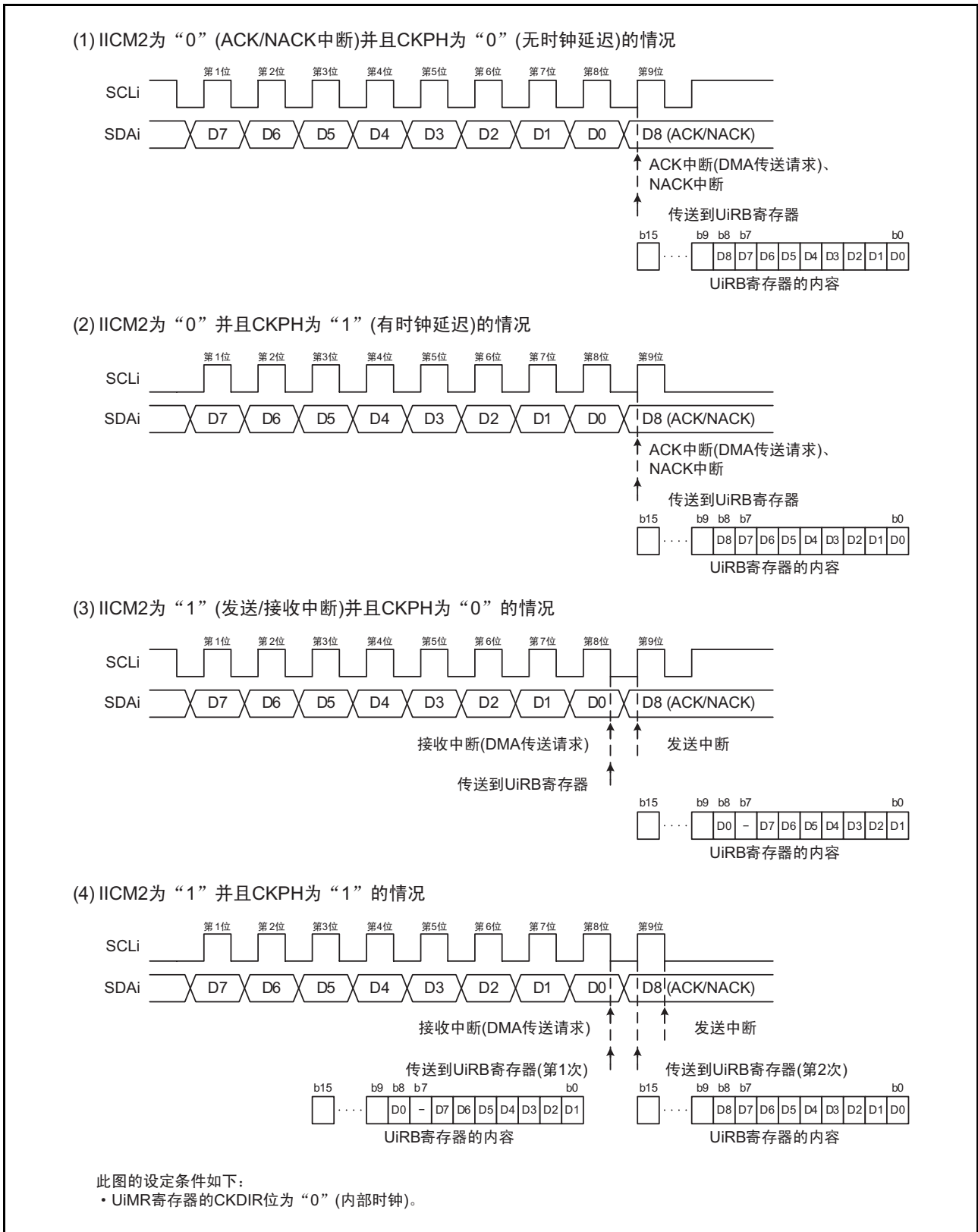


图 18.32 UiRB 寄存器的传送和中断时序 (i=0 ~ 6)

### 18.3.1 开始条件和停止条件的检测

分别通过开始条件检测电路和停止条件检测电路来检测开始条件和停止条件。

在 SCL<sub>i</sub> 引脚 (i=0 ~ 6) 为 “H” 电平的状态下, 如果 SDA<sub>i</sub> 引脚从 “H” 电平变为 “L” 电平, 就产生开始条件的检测中断请求; 如果 SDA<sub>i</sub> 引脚从 “L” 电平变为 “H” 电平, 就产生停止条件的检测中断请求。

因为开始条件的检测中断和停止条件的检测中断共用中断控制寄存器和向量, 所以必须通过 UiSMR 寄存器的 BBS 位来判断是哪个请求产生的中断。

如图 18.33 所示, 在检测开始条件和停止条件时, 准备时间和保持时间都至少需要 6 个外围功能时钟 (f<sub>1</sub>) 周期。为了满足 Fast-Mode 规格的条件, f<sub>1</sub> 不能低于 10MHz。

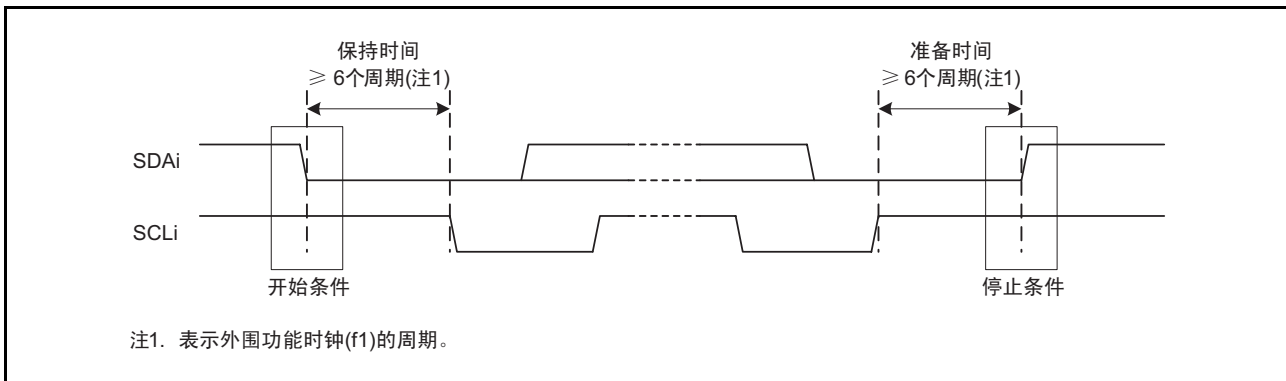


图 18.33 开始条件和停止条件的检测时序 (i=0 ~ 6)

### 18.3.2 开始条件和停止条件的生成

能使用 UiSMR4 寄存器 (i=0 ~ 6) 的 STAREQ 位、RSTAREQ 位和 STPREQ 位, 分别生成开始条件、重新开始条件和停止条件。

在将 STAREQ 位置 “1” (开始) 后, 如果将 UiSMR4 寄存器的 STSPSEL 位置 “1” (选择开始条件/停止条件的生成电路), 就输出开始条件。同样, 在将 RSTAREQ 位置 “1” (开始) 后, 如果将 STSPSEL 位置 “1”, 就输出重新开始条件。在将 STPREQ 位置 “1” (开始) 后, 如果将 STSPSEL 位置 “1”, 就输出停止条件。

STSPSEL 位的功能如表 18.12 和图 18.34 所示。

表 18.12 STSPSEL 位的功能

项目	STSPSEL=0	STSPSEL=1
开始条件和停止条件的生成	通过软件控制端口来实现开始条件和停止条件的生成 (不通过硬件自动生成)。	根据 STAREQ 位、RSTAREQ 位和 STPREQ 位, 输出开始条件和停止条件。
开始条件和停止条件的中断请求的发生时序	检测到开始条件和停止条件时。	开始条件和停止条件的生成结束时。

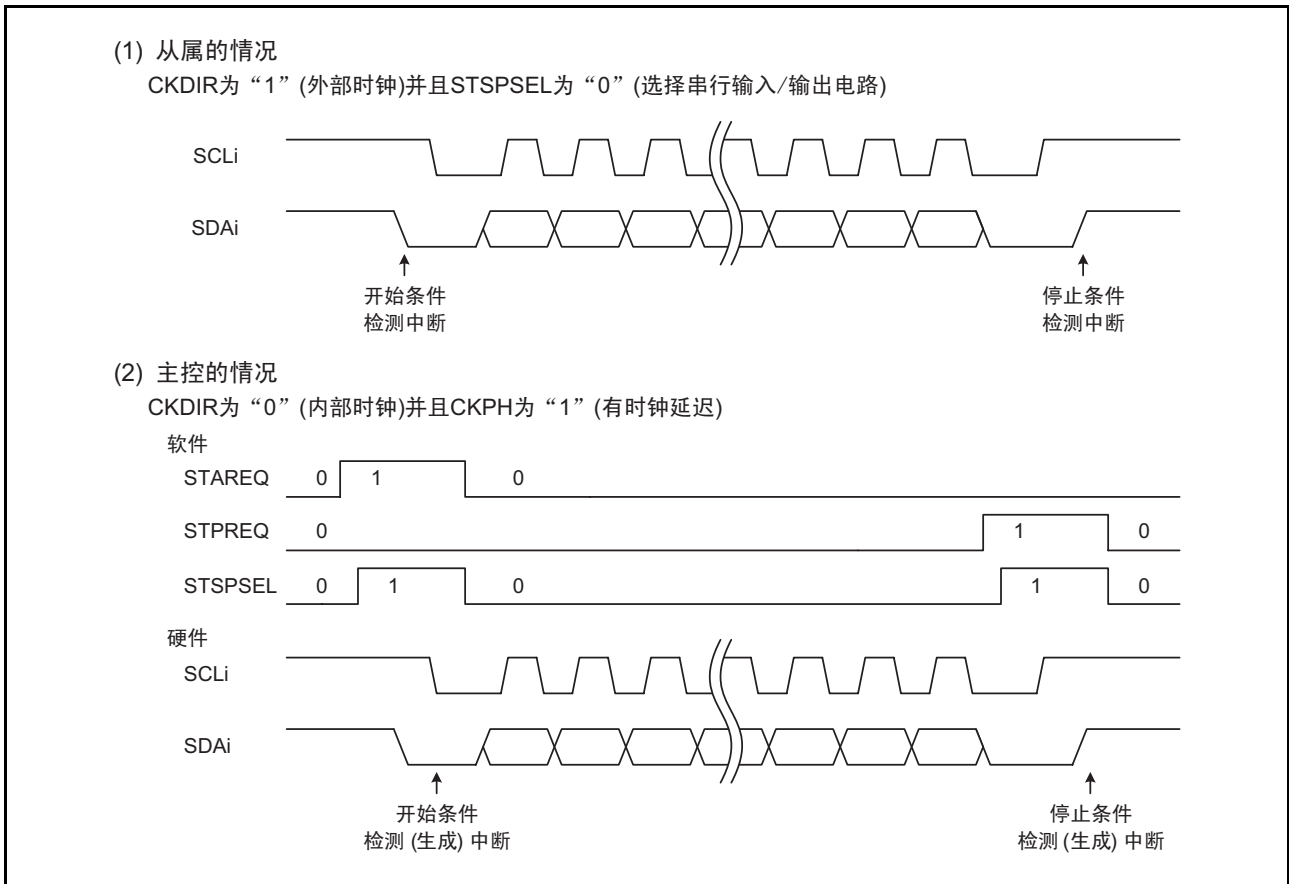


图 18.34 STSPSEL 位的功能 (i=0 ~ 6)

### 18.3.3 仲裁

在 SCLi 上升时判断自身的发送数据和 SDAi 引脚的输入数据是否相同。如果不同，就通过停止输出进行仲裁。

通过 UiSMR 寄存器 (i=0 ~ 6) 的 ABC 位，选择 UiRB 寄存器的 ABT 位的更新时序。当 ABC 位为“0”（按位更新）时，如果在判断时检测到不同，ABT 位就同时变为“1”，否则，就变为“0”。当 ABC 位为“1”（按字节更新）时，只要在判断时检测到一次不同，就在第 8 位的 SCLi 的下降沿，ABT 位变为“1”（检测到）。另外，在按字节更新时，必须在第 1 个字节的 ACK 检测结束后，先将 ABT 位置“0”（未检测到），然后开始下一个字节的发送。

如果将 UiSMR2 寄存器的 ALS 位置“1”（停止 SDA 的输出），就产生仲裁失败，并且在 ABT 位变为“1”的同时，SDAi 引脚变为高阻抗。

### 18.3.4 SCL 控制和时钟同步

通过图 18.32 所示的发送 / 接收时钟进行 I<sup>2</sup>C 模式的发送和接收。如果发送 / 接收时钟变快，就难以确保 ACK 的生成时间以及发送数据的准备时间。I<sup>2</sup>C 模式支持为确保此时间的等待插入功能，并且支持对于其他设备插入的等待进行时钟同步的功能。

在插入用于生成应答的等待时，使用 UiSMR2 寄存器 (i=0 ~ 6) 的 SWC 位。当 SWC 位为“1”（在接收 8 位后，将 SCLi 引脚固定为“L”电平）时，在第 8 位的 SCLi 的下降沿将 SCLi 引脚固定为“L”电平。如果将 SWC 位置“0”（无等待 / 等待解除），就能解除被固定的“L”电平。

如果将 UiSMR2 寄存器的 SWC2 位置“1”（将 SCLi 引脚固定为“L”电平），也能在发送和接收时将 SCLi 引脚固定为“L”电平；如果将 SWC2 位置“0”（将发送 / 接收时钟输出到 SCLi 引脚），就解除 SCLi 引脚被固定的“L”电平，并输出发送 / 接收时钟。

在插入用于判断接收到的应答位的等待时，使用 UiSMR4 寄存器的 SWC9 位。当 UiSMR3 寄存器的 CKPH 位为“1”（有时钟延迟）时，如果将 SWC9 位置“1”（在接收到 9 位后，将 SCLi 引脚固定为“L”电平），就在第 9 位的 SCLi 的下降沿将 SCLi 引脚固定为“L”电平；如果将 SWC9 位置“0”（无等待 / 等待解除），就解除被固定的“L”电平。

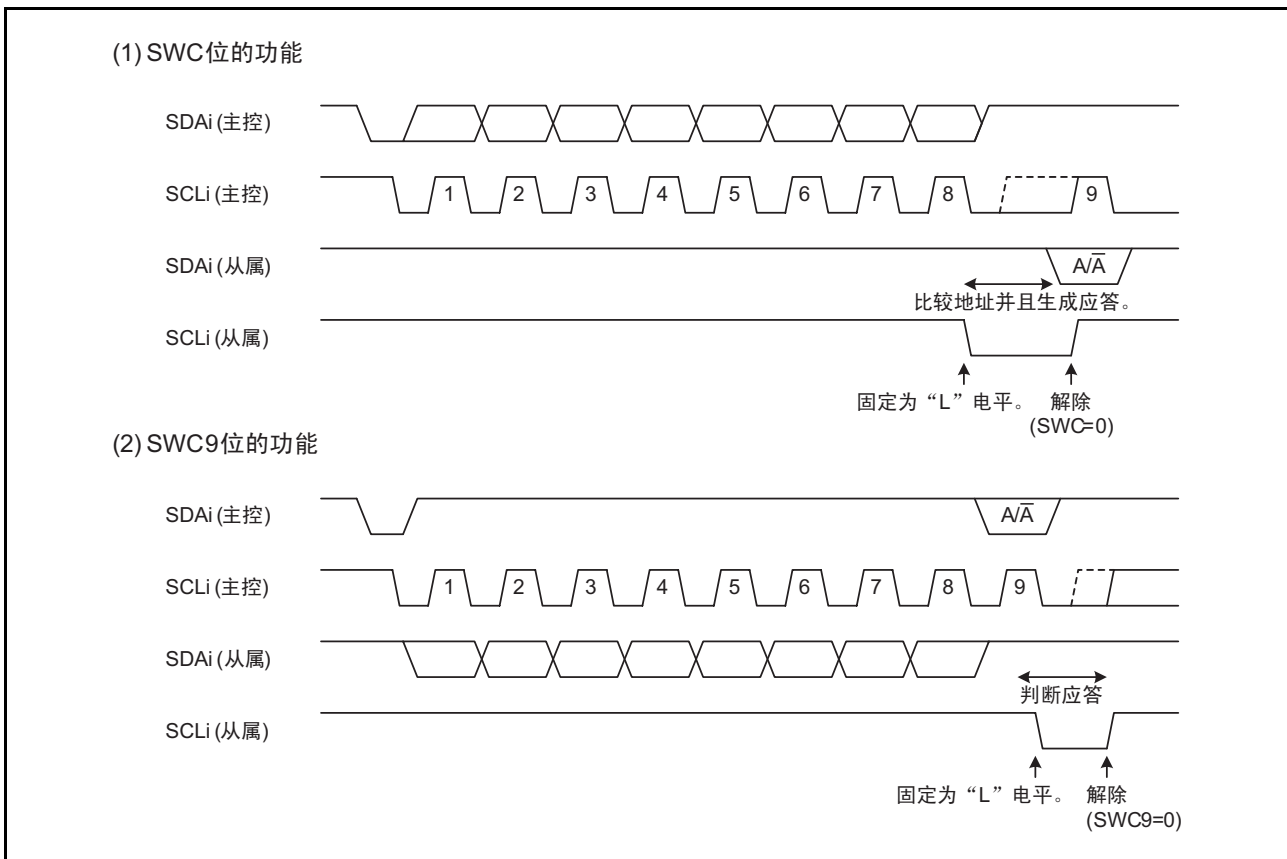


图 18.35 通过 SWC 位或者 SWC9 位插入等待 (i=0 ~ 6)



UiSMR2 寄存器的 CSC 位用于因其他设备插入了等待而在自身输出的时钟和输入到 SCLi 引脚的时钟不同时，使内部生成的时钟与输入到 SCLi 引脚的时钟同步。当 CSC 位为“1”（进行时钟同步）时，如果在内部生成时钟为“H”电平时，SCLi 引脚从“H”电平变为“L”电平，就将内部生成时钟置为“L”电平，在重新加载 UiBRG 寄存器的值后开始对“L”区间进行计数。在 SCLi 引脚为“L”电平时，如果内部生成时钟从“L”电平变为“H”电平，就停止计数。如果 SCLi 引脚变为“H”电平，就重新开始计数。因此，UARTi 的发送/接收时钟为内部生成时钟和 SCLi 引脚信号的逻辑与。在内部生成时钟的 1 个时钟前到第 9 个时钟期间，实现发送/接收时钟的同步。只能在 UiMR 寄存器的 CKDIR 位为“0”（内部时钟）时，将 CSC 位置“1”。

UiSMR4 寄存器的 SCLHI 位用于自身作为主控进行发送和接收时，在其他主控已生成停止条件的情况下，释放 SCLi 引脚。如果将 SCLHI 位置“1”（停止输出），就在检测到停止条件时释放 SCLi 引脚（高阻抗）并停止时钟的输出。

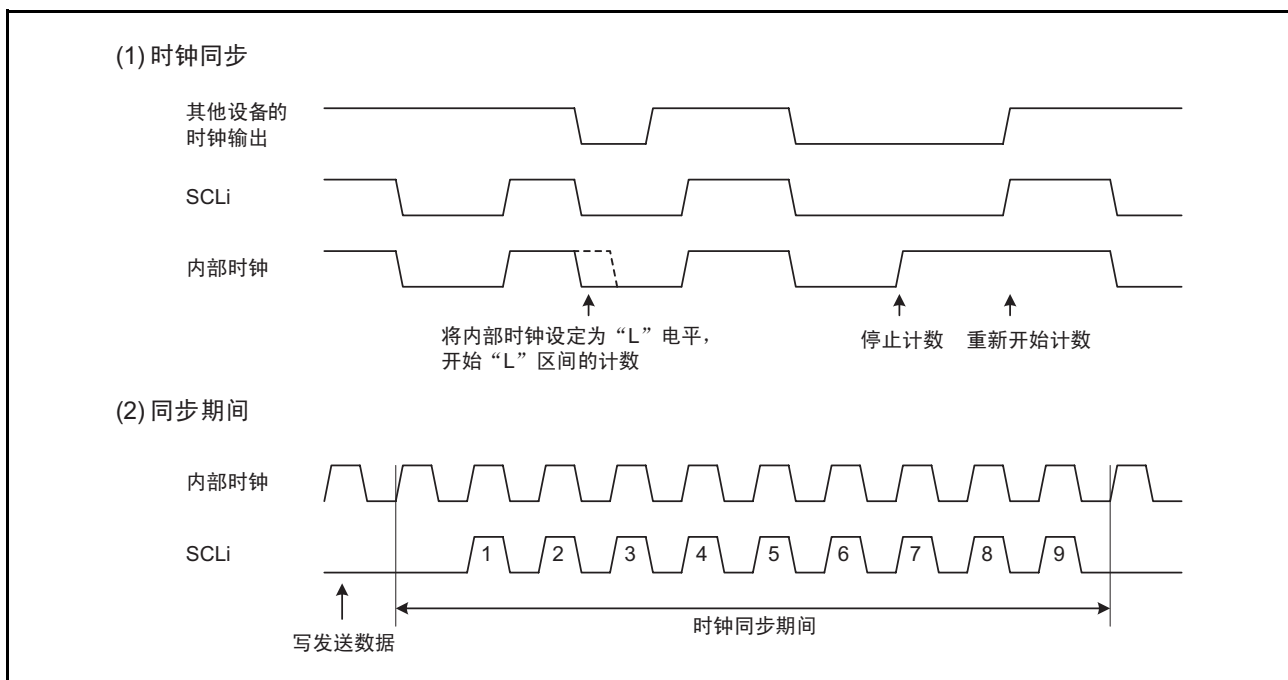


图 18.36 时钟同步 (i=0 ~ 6)

### 18.3.5 SDA 输出

按 D7→D0→D8 的顺序输出 UiTB 寄存器 (i=0 ~ 6) 的 bit8 ~ bit0 (D8 ~ D0) 的值。因为 D8 是用于应答信号的位，所以在发送时需要通过将 D8 置“1”，释放总线。另外，在接收时给 D8 设定 ACK 或者 NACK。

能通过 UiSMR3 寄存器的 DL2 ~ DL0 位，对 SCLi 的下降沿设定 SDAi 的延迟量。延迟量以 UiBRG 计数源为基准，能设定 0 个周期（无）或者 2 ~ 8 个周期。

能通过将 UiSMR2 寄存器的 SDHI 位置“1”（停止输出），随时将 SDAi 引脚置为高阻抗。在设定 I<sup>2</sup>C 模式后，如果给 SDAi 分配端口功能并将引脚设定为输出，就从 SDAi 引脚输出“L”电平。此时，如果 SDHI 位为“1”，SDAi 引脚就变为高阻抗。

如果在 SCLi 引脚为“H”电平时改写 SDHI 位，就能生成开始条件和停止条件；如果在 SCLi 上升前改写 SDHI 位，就有可能误检测到仲裁失败。如果要在 SCLi 引脚为“L”电平时使 SDAi 引脚发生变化，就必须改写 SDHI 位。

### 18.3.6 SDA 输入

当 UiSMR2 寄存器 (i=0 ~ 6) 的 IICM2 位为 “0” (使用 ACK/NACK 中断) 时, 将接收数据的第 1 ~ 8 位 (D7 ~ D0) 保存到 UiRB 寄存器的 bit7 ~ bit0, 将第 9 位 (ACK/NACK) 保存到 UiRB 寄存器的 bit8。

当 IICM2 位为 “1” (使用发送 / 接收中断) 时, 将接收数据的第 1 ~ 7 位 (D7 ~ D1) 保存到 UiRB 寄存器的 bit6 ~ bit0, 将第 8 位 (D0) 保存到 UiRB 寄存器的 bit8。即使 IICM2 位为 “1”, 只要 UiSMR3 寄存器的 CKPH 位为 “1” (有时钟延迟), 就能通过在第 9 位的 SCLi 的上升后读 UiRB 寄存器, 读到和 IICM2 位为 “0” 时的相同数据。

### 18.3.7 应答

在已确定接收数据时, 通过将 “00FFh” 作为虚拟数据设定到 UiTB 寄存器, 就在接收到 8 位后输出 ACK。

当 UiSMR4 寄存器 (i=0 ~ 6) 的 STSPSEL 位为 “0” (选择串行输入 / 输出电路) 并且 UiSMR4 寄存器的 ACKC 位为 “1” (输出 ACK 数据) 时, 从 SDAi 引脚输出 UiSMR4 寄存器的 ACKD 位的值。

当 IICM2 位为 “0” 时, 如果在第 9 位的 SCLi 上升时 SDAi 引脚为 “H” 电平, 就产生 NACK 中断请求; 如果 SDAi 引脚为 “L” 电平, 就产生 ACK 中断请求。

如果选择 “UARTi 接收或者 ACK 中断请求” 作为 DMA 启动源, 就能通过 ACK 检测来启动 DMA 传送。

### 18.3.8 发送和接收的初始化

当 UiMR 寄存器 (i=0 ~ 6) 的 CKDIR 位为 “1” (外部时钟) 并且 UiSMR2 寄存器的 STC 位为 “1” (对电路进行初始化) 时, 如果检测到开始条件, 就进行以下的运行:

- 对发送寄存器进行初始化, 将 UiTB 寄存器的内容传送到发送寄存器。将下一个 SCLi 的下降沿作为第 1 位发送时钟开始发送, 但是从 SCLi 的下降沿到输出数据前, 从 SDAi 引脚输出初始化前的发送寄存器的值。
- 对接收寄存器进行初始化, 将下一个 SCLi 的下降沿作为第 1 位接收时钟开始接收。
- UiSMR2 寄存器的 SWC 位变为 “1” (在接收 8 位后, 将 SCL 引脚固定为 “L” 电平)。

另外, 在使用此功能开始 UARTi 的发送和接收时, TI 位不变。

## 18.4 特殊模式 2

特殊模式 2 是在 1 个或者多个主控模式与多个从属模式之间，进行时钟同步串行通信的模式。使用  $\overline{\text{SSi}}$  输入引脚（ $i=0 \sim 6$ ）对串行总线的通信进行控制。特殊模式 2 的规格如表 18.13 所示。

表 18.13 特殊模式 2 的规格

项目	规格
数据格式	字符长：8 位
发送 / 接收时钟	<ul style="list-style-type: none"> <li>当 UiMR 寄存器（<math>i=0 \sim 6</math>）的 CKDIR 位为“0”（内部时钟）时：  <math display="block">\frac{f_x}{2(m+1)}</math> <math>f_x</math>: f1、f8、f2n      <math>m</math>: UiBRG 寄存器的设定值（00h ~ FFh）</li> <li>当 CKDIR 位为“1”（外部时钟）时：CLKi 引脚的输入</li> </ul>
发送 / 接收控制	SS 功能
发送开始条件	开始发送时需要以下的条件（注 1）： <ul style="list-style-type: none"> <li>UiC1 寄存器的 TE 位为“1”（允许发送）。</li> <li>UiC1 寄存器的 TI 位为“0”（UiTB 寄存器有数据）。</li> </ul>
接收开始条件	开始接收时需要以下的条件（注 1）： <ul style="list-style-type: none"> <li>UiC1 寄存器的 RE 位为“1”（允许接收）。</li> <li>TE 位为“1”（允许发送）。</li> <li>TI 位为“0”（UiTB 寄存器有数据）。</li> </ul>
中断请求的发生时序	发送中断能通过设定 U0C1 ~ U6C1 寄存器的 UiIRS 位，选择以下的任意条件： <ul style="list-style-type: none"> <li>UiIRS 位为“0”（UiTB 寄存器空）： 在将数据从 UiTB 寄存器传送到 UARTi 发送寄存器时（开始发送时）。</li> <li>UiIRS 位为“1”（发送结束）： 在 UARTi 发送寄存器的数据发送结束时。</li> </ul> 接收中断 <ul style="list-style-type: none"> <li>在将数据从 UARTi 接收寄存器传送到 UiRB 寄存器时（接收结束时）。</li> </ul>
错误检测	溢出错误（注 2） 如果在读 UiRB 寄存器前接收到下一个数据的第 7 位，就发生溢出错误。
其他选择项目	<ul style="list-style-type: none"> <li>CLK 极性选择 发送 / 接收数据的输出 / 输入时序能选择与发送 / 接收时钟上升沿或者下降沿同步。</li> <li>位序选择 能选择 LSB first 或者 MSB first。</li> <li>连续接收模式的选择 能在读 UiRB 寄存器的同时设定为接收允许状态。</li> <li>串行数据的逻辑转换 能将发送 / 接收数据进行逻辑取反。</li> <li>时钟相位的选择 能根据发送 / 接收时钟的极性和相位选择 4 种时钟。</li> <li><math>\overline{\text{SSi}}</math> 输入引脚的功能 当 <math>\overline{\text{SSi}}</math> 引脚为“H”电平时，能将输出引脚置为高阻抗。</li> </ul>

注 1. 在选择外部时钟时，必须满足以下的条件：

当 UiC0 寄存器的 CKPOL 位为“0”（与发送/接收时钟的下降沿同步输出发送数据，与上升沿同步输入接收数据）时，CLKi 引脚为“H”电平状态；当 CKPOL 位为“1”（与发送/接收时钟的上升沿同步输出发送数据，与下降沿同步输入接收数据）时，CLKi 引脚为“L”电平状态。

注 2. 如果发生溢出错误，UiRB 寄存器就为不定值，而且 SiRIC 寄存器的 IR 位不变为“1”（有中断请求）。

特殊模式 2 中使用的寄存器及其设定值如表 18.14 所示。

表 18.14 特殊模式 2 中使用的寄存器及其设定值 (i=0 ~ 6)

寄存器	位	功能
UiMR	7 ~ 4	必须置“0000b”。
	CKDIR	在主导模式中，必须置“0”；在从属模式中，必须置“1”。
	SMD2 ~ SMD0	必须置“001b”。
UiC0	UFORM	必须选择 LSB first 或者 MSB first。
	CKPOL	能根据和 UiSMR3 寄存器的 CKPH 位的组合设定时钟相位。
	5	必须置“0”。
	CRD	必须置“1”。
	TXEPT	发送寄存器的空标志
	2	必须置“0”。
	CLK1 ~ CLK0	必须选择 UiBRG 寄存器的计数源。
UiC1	7 ~ 6	必须置“00b”。
	UiRRM	在使用连续接收模式时，必须置“1”。
	UiIRS	必须选择 UARTi 发送中断源。
	RI	接收结束标志
	RE	在允许接收时，必须置“1”。
	TI	发送缓冲器的空标志
	TE	在允许发送和接收时，必须置“1”。
UiSMR	7 ~ 0	必须置“00h”。
UiSMR2	7 ~ 0	必须置“00h”。
UiSMR3	7 ~ 5	必须置“000b”。
	ERR	模式错误标志
	3	必须置“0”。
	DINC	在主导模式中，必须置“0”；在从属模式中，必须置“1”。
	CKPH	能根据和 UiC0 寄存器的 CKPOL 位的组合设定时钟相位。
	SSE	必须置“1”。
UiSMR4	7 ~ 0	必须置“00h”。
UiBRG	7 ~ 0	必须设定速率。
IFS0	IFS06	必须选择 CLK3、RXD3、SRXD3、 $\overline{SS3}$ 的输入引脚。
	IFS03 ~ IFS02	必须选择 CLK6、RXD6、SRXD6、 $\overline{SS6}$ 的输入引脚。
UiTB	7 ~ 0	必须设定发送数据。
UiRB	OER	溢出错误标志
	7 ~ 0	能读接收数据。

### 18.4.1 $\overline{\text{SS}}_i$ 输入引脚的功能 (i=0 ~ 6)

如果将 UiSMR3 寄存器的 SSE 位置 “1” (允许 SS 功能), 就选择此模式并且  $\overline{\text{CTS}}_i/\overline{\text{RTS}}_i/\overline{\text{SS}}_i$  引脚变为  $\overline{\text{SS}}_i$  输入引脚。

能通过 UiSMR3 寄存器的 DINC 位, 选择是用作主控还是用作从属。如果将多个设备用作主控 (多主控系统), 就取决于  $\overline{\text{SS}}_i$  引脚状态下的主控。

#### 18.4.1.1 从属模式的 SS 功能

当 DINC 位为 “1” (从属模式) 时, 如果从  $\overline{\text{SS}}_i$  引脚输入 “H” 电平, STXDi 引脚就为高阻抗, 忽视 CLKi 引脚的时钟输入; 如果从  $\overline{\text{SS}}_i$  引脚输入 “L” 电平, 时钟的输入就有效, 能从 STXDi 引脚输出串行数据并进行串行通信。

#### 18.4.1.2 主控模式的 SS 功能

当 DINC 位为 “0” (主控模式) 时, 如果从  $\overline{\text{SS}}_i$  引脚输入 “H” 电平, 就表明不存在其他主控或者其他主控不在进行通信, 所以能用作主控开始通信, 主控从 CLKi 引脚输出发送 / 接收时钟; 如果从  $\overline{\text{SS}}_i$  引脚输入 “L” 电平, 就表明存在其他主控, TXDi 和 CLKi 的各引脚变为高阻抗。另外, 会发生模式错误并且 UiSMR3 寄存器的 ERR 位变为 “1”。即使在通信时发生模式错误, 也不停止通信。如果要停止通信, 就必须将 UiMR 寄存器的 SMD2 ~ SMD0 位置 “000b” (串行接口无效)。

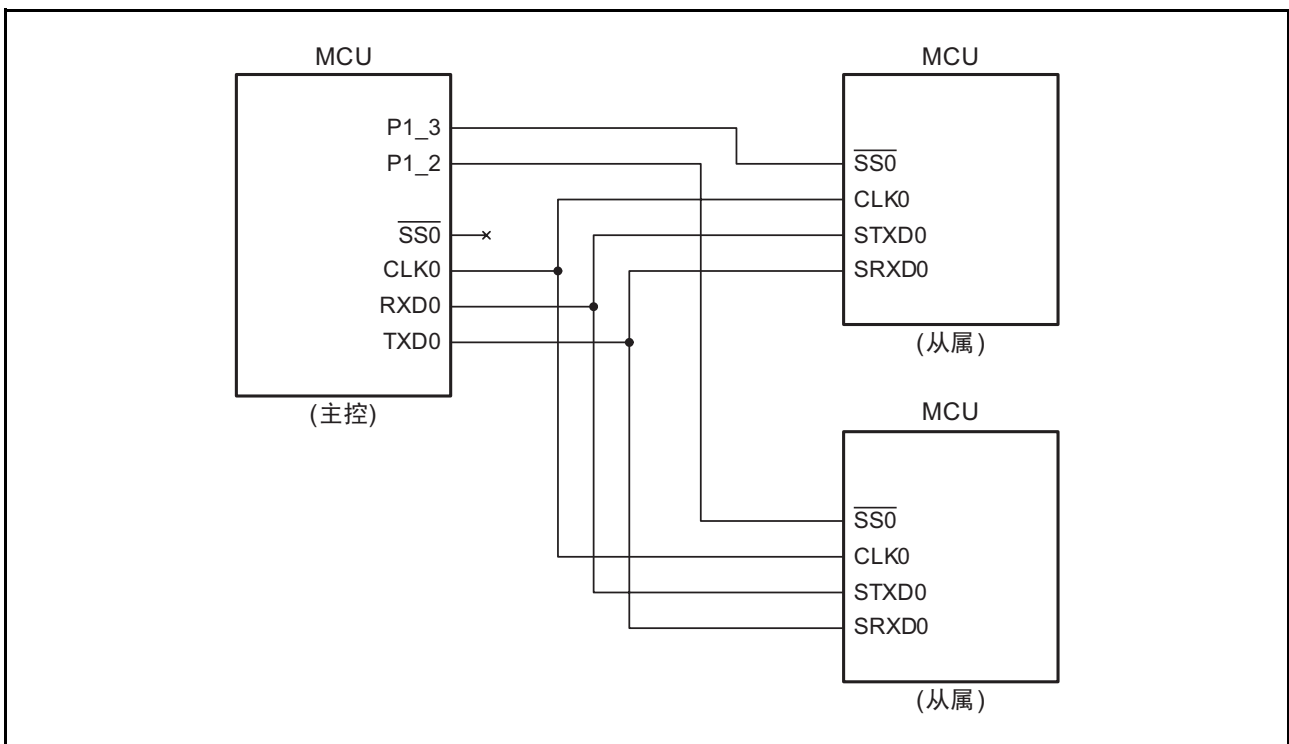


图 18.37 使用  $\overline{\text{SS}}_i$  引脚进行串行总线通信控制的例子

### 18.4.2 时钟相位的设定功能

能通过 UiC0 寄存器的 CKPOL 位和 UiSMR3 寄存器 (i=0 ~ 6) 的 CKPH 位, 选择发送 / 接收时钟的极性和相位不同组合的 4 种时钟。

主控模式的各发送 / 接收时钟的极性和相位必须和当时要通信的从属模式相同。

#### 18.4.2.1 主控模式的发送 / 接收时序

当 DINC 位为 “0” (主控模式) 时, 需要通过将 UiMR 寄存器的 CKDIR 位置 “0” (内部时钟) 并生成时钟。各时钟相位的发送 / 接收时序如图 18.38 所示。

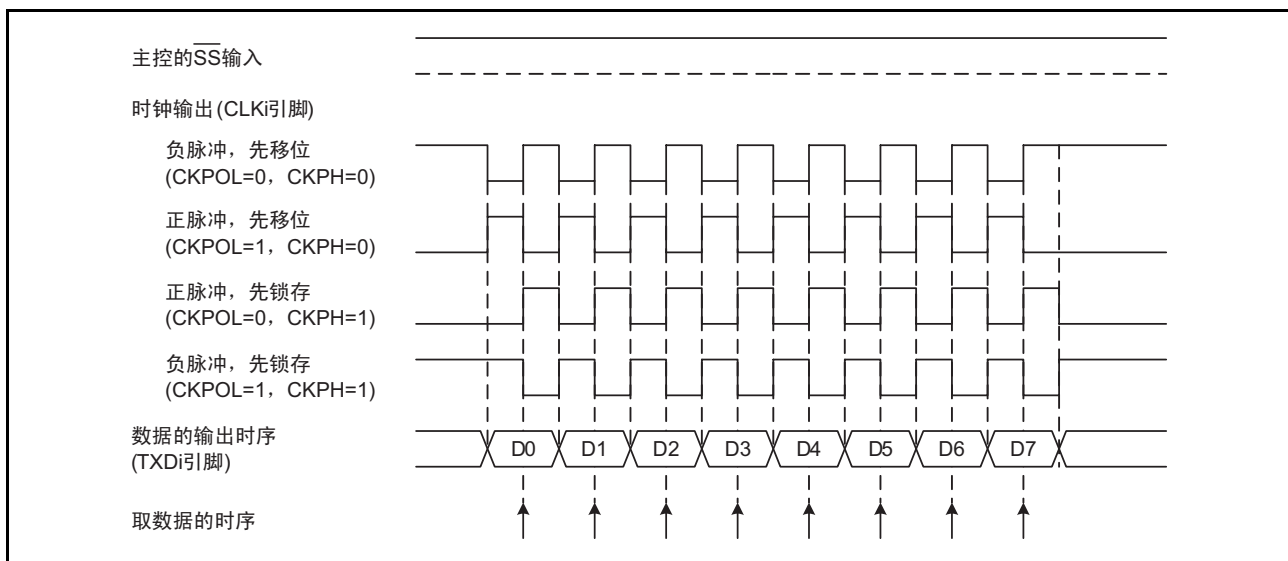


图 18.38 主控模式的发送 / 接收时序

18.4.2.2 从属模式的发送 / 接收时序

当 DINC 位为 “1”（从属模式）时，需要将 UIMR 寄存器的 CKDIR 位置 “1”（外部时钟）。

当 CKPH 位为 “0”（无时钟延迟）时，如果从 SSi 引脚输入 “H” 电平，STXDi 就为高阻抗；如果从 SSi 引脚输入 “L” 电平，就具备了数据发送的开始条件，但是输出不稳定。然后，与时钟同步进行数据发送。从属模式的发送 / 接收时序如图 18.39 所示。

当 CKPH 位为 “1”（有时钟延迟）时，如果从 SSi 引脚输入 “H” 电平，STXDi 就为高阻抗；如果从 SSi 引脚输入 “L” 电平，就输出第 1 个数据。然后，与时钟同步进行数据发送。从属模式的发送 / 接收时序如图 18.40 所示。

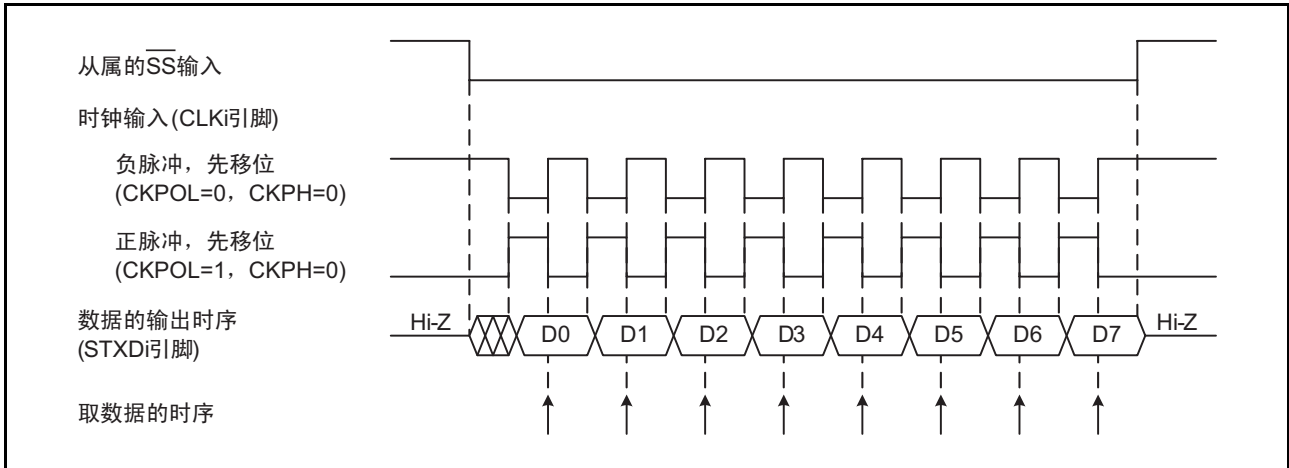


图 18.39 从属模式的发送 / 接收时序 (CKPH=0)

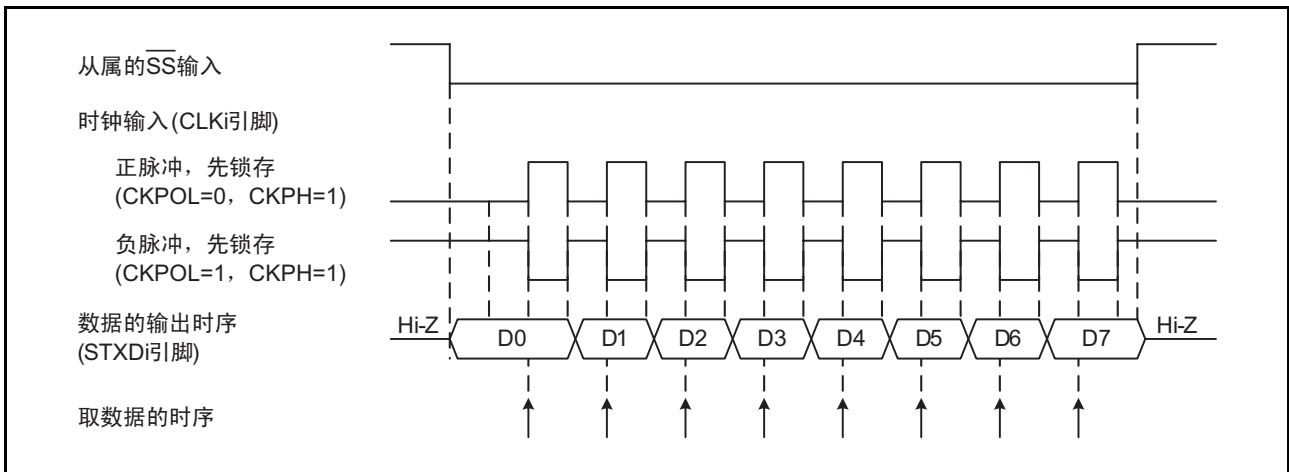


图 18.40 从属模式的发送 / 接收时序 (CKPH=1)

## 18.5 使用串行接口时的注意事项

### 18.5.1 UiBRG 寄存器 (i=0 ~ 8) 的变更

- 必须在设定 UiC0 寄存器的 CLK1 ~ CLK0 位后写 UiBRG 寄存器。如果更改 CLK1 ~ CLK0 位，就必须重新设定 UiBRG 寄存器。
- 如果给 UiBRG 寄存器写 “00h”，计数器就立即开始运行并且有可能变为 “FFh”。此时，到重新加载所设定的 “00h” 为止，需要的时间约为 256 个时钟。在重新加载 “00h” 后，根据设定，变为无分频。

### 18.5.2 时钟同步模式

#### 18.5.2.1 外部时钟的选择

- 在已选择外部时钟的情况下，必须在 UiC0 寄存器 (i=0 ~ 8) 的 CKPOL 位为 “0” (与发送/接收时钟的下降沿同步输出发送数据，与上升沿同步输入接收数据) 时并且外部时钟为 “H” 电平的状态下，或者在 CKPOL 位为 “1” (与发送/接收时钟的上升沿同步输出发送数据，与下降沿同步输入接收数据) 时并且外部时钟为 “L” 电平的状态下，满足以下的条件：
  - UiC1 寄存器的 TE 位为 “1” (允许发送)。
  - UiC1 寄存器的 RE 位为 “1” (允许接收)。
  - UiC1 寄存器的 TI 位为 “0” (UiTB 寄存器有数据)。在只进行发送时，不需要设定 RE 位。

#### 18.5.2.2 接收

- 时钟同步模式通过发送控制电路对发送/接收时钟进行控制。因此，即使在只进行接收时也必须进行发送的设定。在进行接收时，将虚拟数据从 TXDi 引脚 (i=0 ~ 8) 输出到外部。
- 在连续接收数据的情况下，如果在 UiC1 寄存器的 RI 位为 “1” (UiRB 寄存器有数据) 时接收到下一个接收数据的第 7 位，就发生溢出错误，并且 UiRB 寄存器的 OER 位变为 “1” (发生溢出错误)。此时，UiRB 寄存器为不定值。在发生溢出错误时，SiRIC 寄存器的 IR 位不变为 “1”。

### 18.5.3 特殊模式 1 (I<sup>2</sup>C 模式)

在生成开始条件、停止条件或者重新开始条件时，必须在将 UiSMR4 寄存器 (i=0 ~ 6) 的 STSPSEL 位置 “0” 后，至少等待半个发送/接收时钟，然后将各条件的生成位 (STAREQ 位、RSTAREQ 位、STPREQ 位) 从 “0” 置为 “1”。



## 19. A/D 转换器

有 1 个由电容耦合放大器构成的 10 位逐次逼近转换方式的 A/D 转换器。

将 A/D 转换结果保存到对应所选引脚的 A/D 寄存器。但是，在 DMAC 利用模式有效时，只保存到 AD00 寄存器。

在不使用 A/D 转换器时，如果将 AD0CON1 寄存器的 VCUT 位置“0”（断开 VREF），提供的电流就不从 VREF 引脚流到梯形电阻，从而能降低功耗。

A/D 转换器的规格如表 19.1 所示，A/D 转换器的框图及其相关寄存器分别如图 19.1 和图 19.2 ~ 图 19.7 所示。

表 19.1 A/D 转换器的规格

项目	性能
A/D 转换方式	逐次逼近转换方式（电容耦合放大器）
模拟输入电压（注 1）	0V ~ AVCC(VCC)
运行时钟 $\phi_{AD}$ （注 2）	fAD、fAD 的 2 分频、fAD 的 3 分频、fAD 的 4 分频、fAD 的 6 分频或者 fAD 的 8 分频
分辨率	8 位、10 位
运行模式	单次模式、重复模式、单次扫描模式、重复扫描模式 0、重复扫描模式 1、多端口单次扫描模式、多端口重复扫描模式 0
模拟输入引脚（注 3）	34 个（注 4） AN、AN0、AN2、AN15 各有 8 个（注 5），扩展输入有 2 个（ANEX0、ANEX1）。
A/D 转换的开始条件	软件触发 <ul style="list-style-type: none"> <li>将 AD0CON0 寄存器的 ADST 位置“1”（开始 A/D 转换）。</li> </ul> 外部触发（能重新触发） <ul style="list-style-type: none"> <li>在将 ADST 位置“1”后，<math>\overline{ADTRG}</math> 引脚的输入信号从“H”电平变为“L”电平。</li> </ul> 硬件触发（能重新触发） <ul style="list-style-type: none"> <li>在将 ADST 位置“1”后，发生三相马达控制定时器功能的（经过中断发生频率的设定电路后）定时器 B2 的中断请求。</li> </ul>
每个引脚的转换速度	<ul style="list-style-type: none"> <li>无采样 &amp; 保持                分辨率为 8 位：49 个 <math>\phi_{AD}</math> 周期                分辨率为 10 位：59 个 <math>\phi_{AD}</math> 周期                其中，2 个 <math>\phi_{AD}</math> 周期为采样时间。</li> <li>有采样 &amp; 保持                分辨率为 8 位：28 个 <math>\phi_{AD}</math> 周期                分辨率为 10 位：33 个 <math>\phi_{AD}</math> 周期                其中，3 个 <math>\phi_{AD}</math> 周期为采样时间。</li> </ul>

注 1. 不取决于采样 & 保持功能的有无。

注 2. 当 VCC=4.2 ~ 5.5V 时， $\phi_{AD}$  的频率不能超过 16MHz；当 VCC=3.0 ~ 4.2V 时， $\phi_{AD}$  的频率不能超过 10MHz。在没有采样 & 保持功能时， $\phi_{AD}$  的频率至少为 250kHz；在有采样 & 保持功能时， $\phi_{AD}$  的频率至少为 1MHz。

注 3. 当 AVCC=VREF=VCC 时，AN\_0 ~ AN\_7、AN0\_0 ~ AN0\_7、AN2\_0 ~ AN2\_7、AN15\_0 ~ AN15\_7、ANEX0、ANEX1 的 A/D 输入电压不能超过 VCC。

注 4. 这是 144 引脚版的数值，100 引脚版为 26 个。

注 5. 100 引脚版中没有 AN15\_0 ~ AN15\_7 引脚。

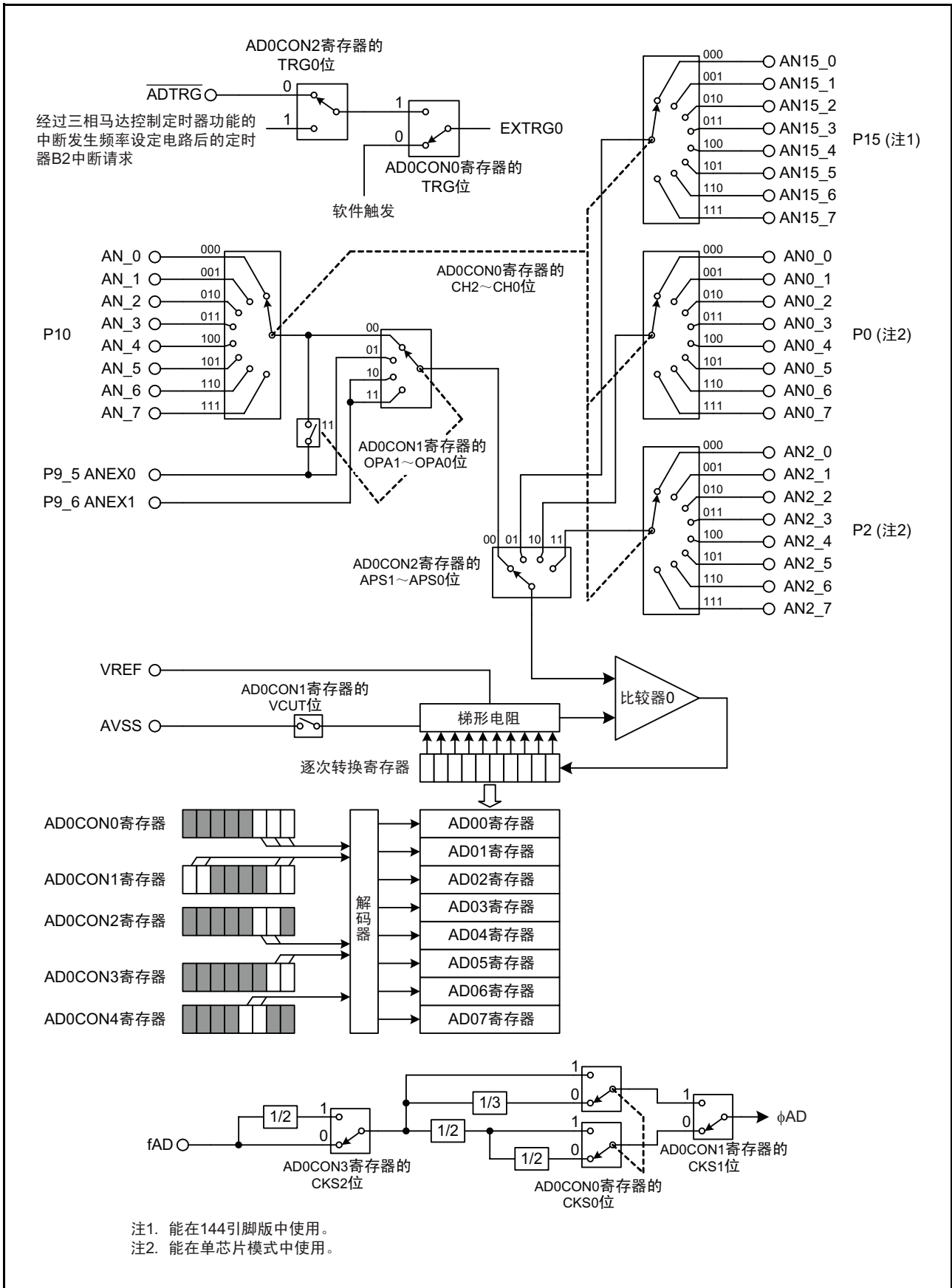


图 19.1 A/D 转换器的框图

注1. 能在144引脚版中使用。  
注2. 能在单芯片模式中使用。

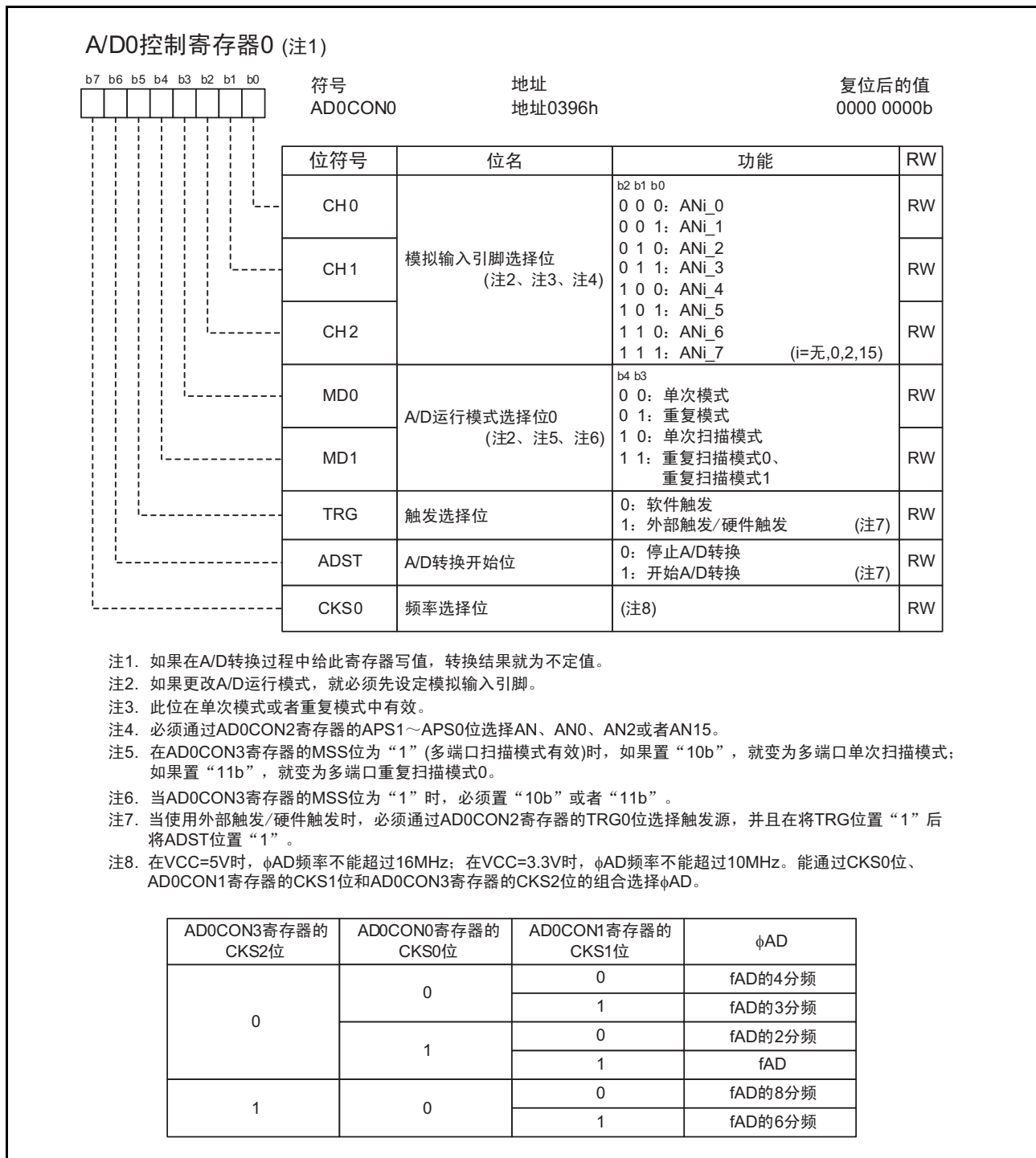


图 19.2 AD0CON0 寄存器

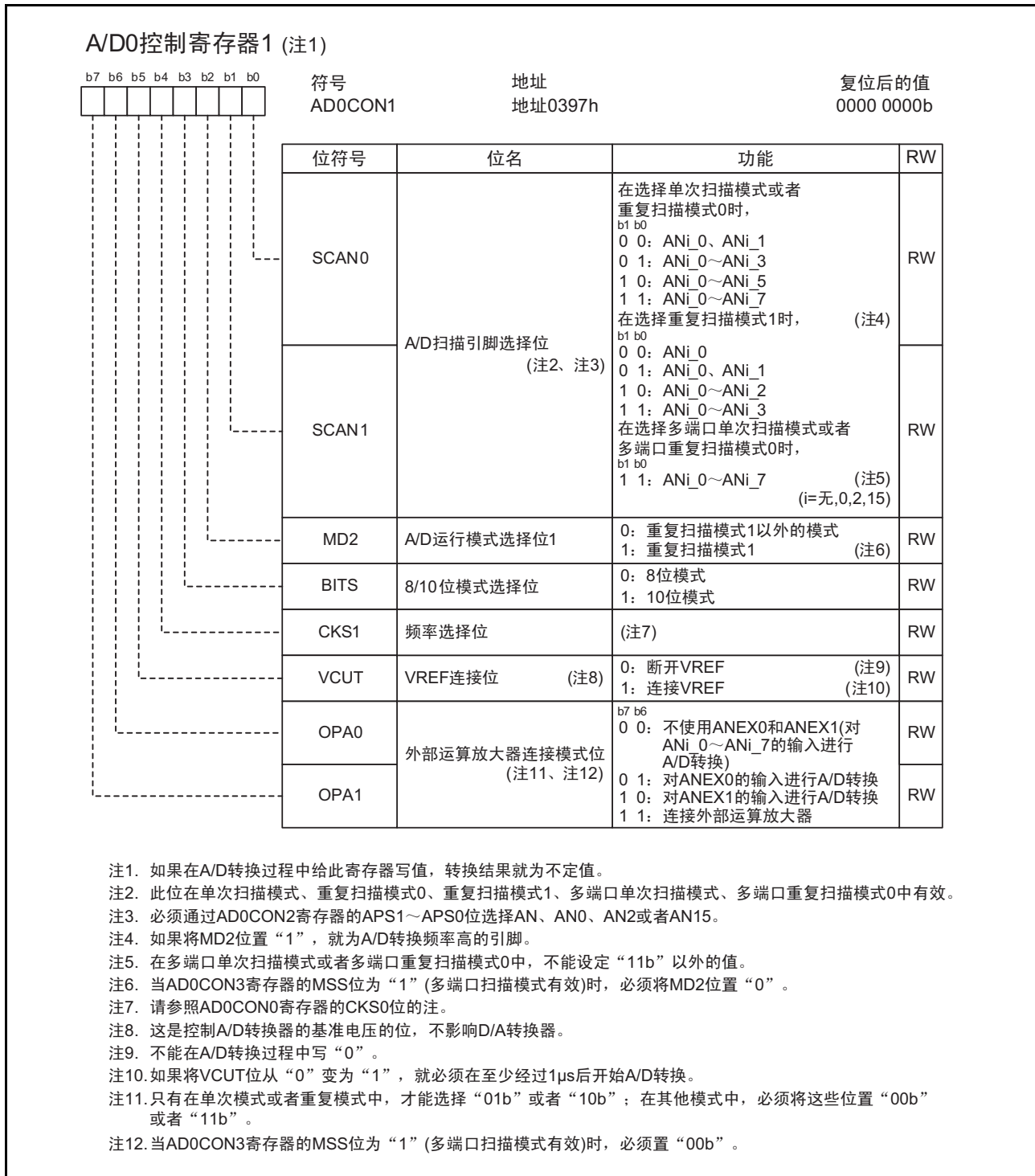


图 19.3 AD0CON1 寄存器

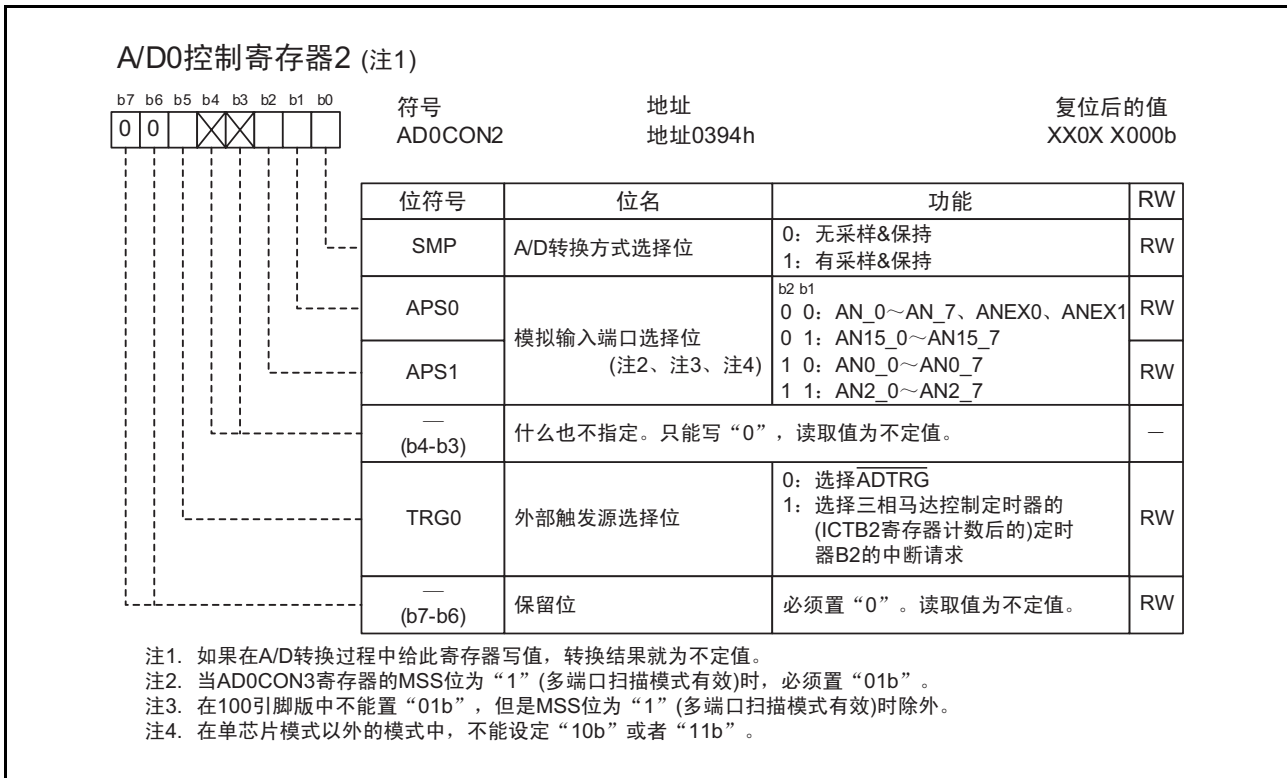


图 19.4 AD0CON2 寄存器

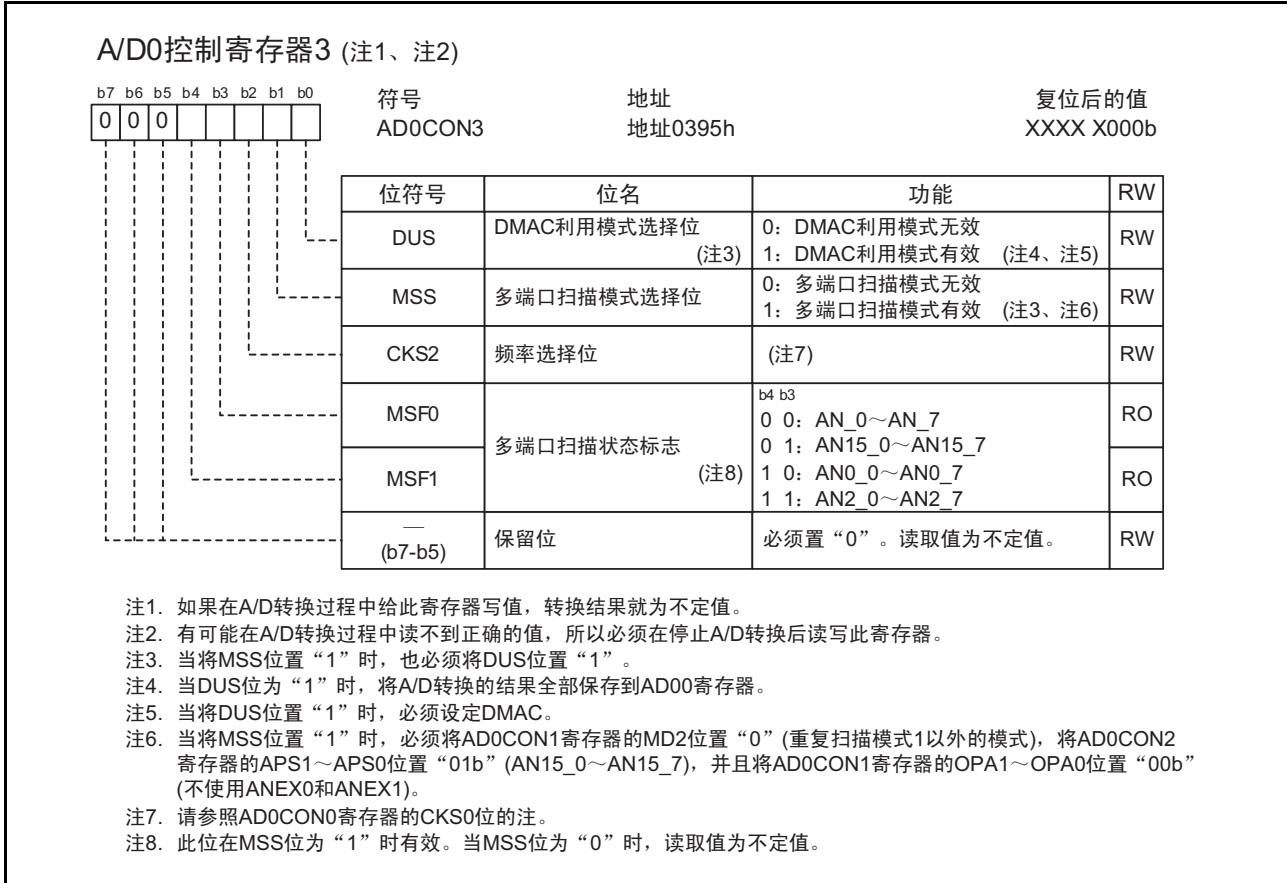


图 19.5 AD0CON3 寄存器

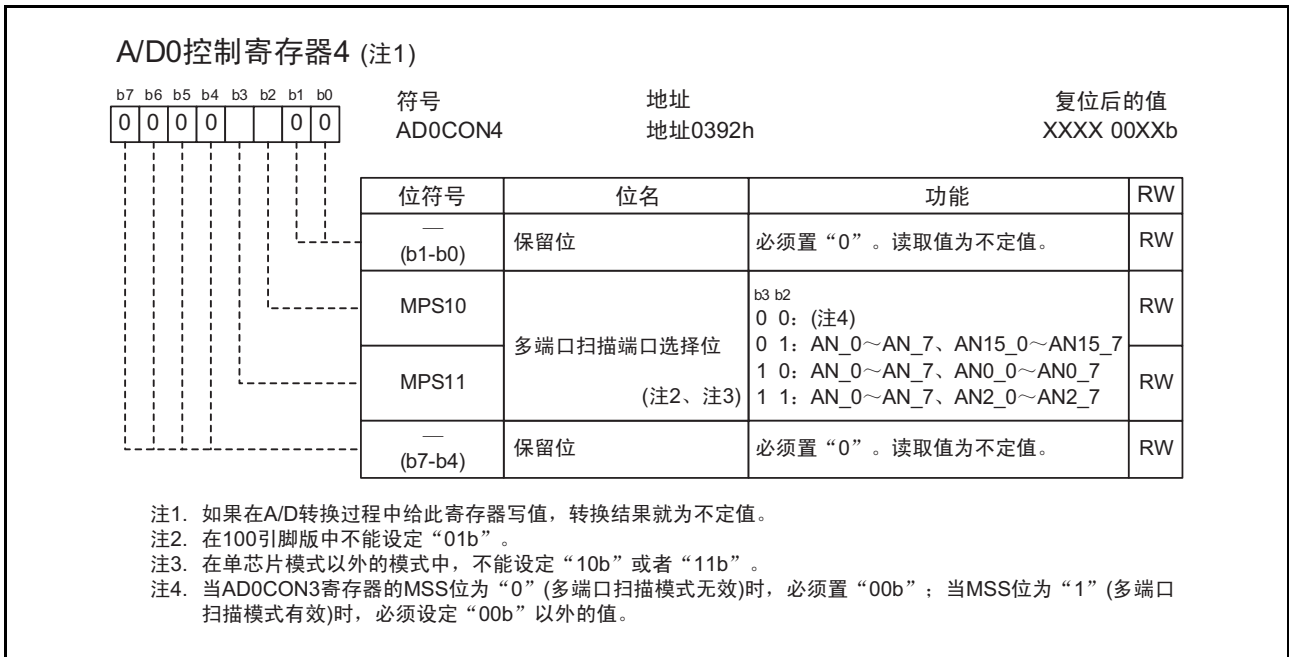


图 19.6 AD0CON4 寄存器

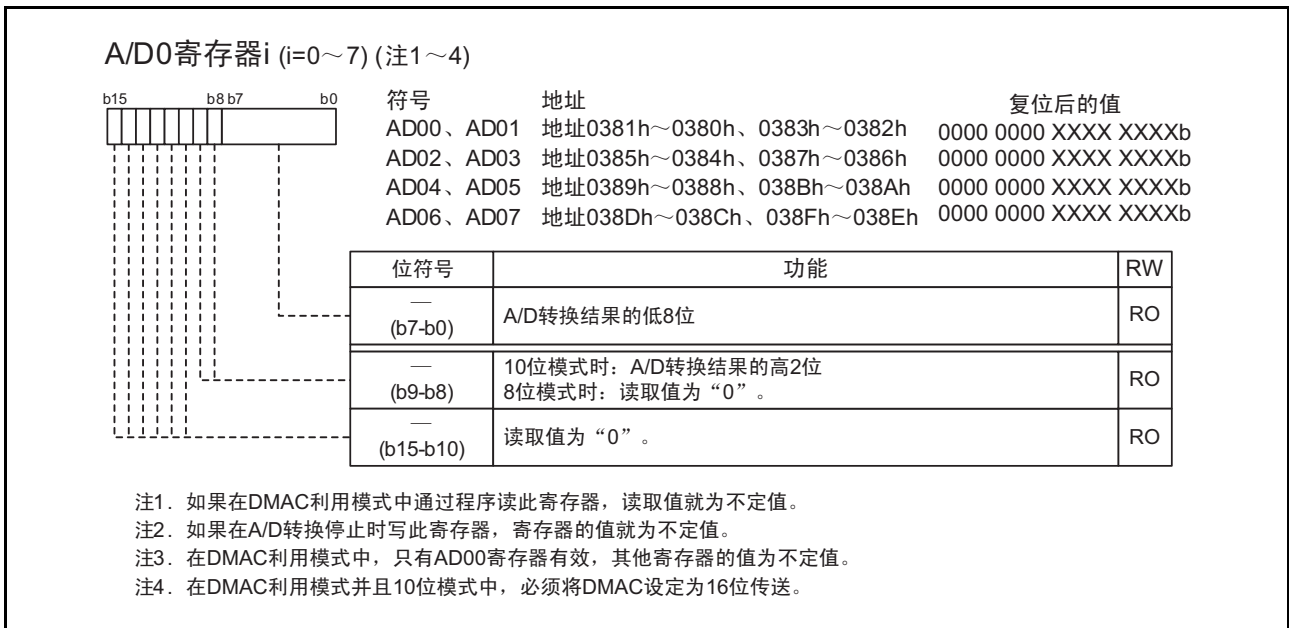


图 19.7 AD00 ~ AD07 寄存器

## 19.1 模式的说明

### 19.1.1 单次模式

这是将所选的 1 个引脚的输入电压进行 1 次 A/D 转换的模式。单次模式的规格如表 19.2 所示。

表 19.2 单次模式的规格

项目	规格
功能	通过 AD0CON0 寄存器的 CH2 ~ CH0 位、AD0CON1 寄存器的 OPA1 ~ OPA0 位、或者 AD0CON2 寄存器的 APS1 ~ APS0 位选择 1 个引脚，将该引脚的输入电压进行 1 次 A/D 转换。
开始条件	当 AD0CON0 寄存器的 TRG 位为 “0”（软件触发）时 <ul style="list-style-type: none"> <li>将 AD0CON0 寄存器的 ADST 位置 “1”（开始 A/D 转换）。</li> </ul> 当 TRG 位为 “1”（外部触发 / 硬件触发）时 <ul style="list-style-type: none"> <li>在将 ADST 位置 “1” 后，ADTRG 引脚的输入信号从 “H” 电平变为 “L” 电平。</li> <li>在将 ADST 位置 “1” 后，发生三相马达控制定时器功能的（经过中断发生频率的设定电路后）定时器 B2 的中断请求。</li> </ul>
停止条件	<ul style="list-style-type: none"> <li>在 A/D 转换结束（当选择软件触发时，ADST 位为 “0”）时。</li> <li>在将 ADST 位置 “0”（停止 A/D 转换）时。</li> </ul>
中断请求的发生时序	<ul style="list-style-type: none"> <li>在 A/D 转换结束时。</li> </ul>
输入引脚	从 AN_0 ~ AN_7、AN0_0 ~ AN0_7、AN2_0 ~ AN2_7、AN15_0 ~ AN15_7、ANEX0、ANEX1 中选择 1 个引脚。
A/D 转换值的读	当 AD0CON3 寄存器的 DUS 位为 “0”（DMAC 利用模式无效）时 读与所选引脚对应的 AD0j 寄存器（j=0 ~ 7）。 当 DUS 位为 “1”（DMAC 利用模式有效）时 不能读 AD00 寄存器。 在 A/D 转换结束后，将 A/D 转换结果保存到 AD00 寄存器，并通过 DMAC 将此结果从 AD00 寄存器传送到任意的存储空间。请参照 “13. DMAC” 设定 DMAC。

### 19.1.2 重复模式

这是将所选的 1 个引脚的输入电压重复进行 A/D 转换的模式。重复模式的规格如表 19.3 所示。

表 19.3 重复模式的规格

项目	规格
功能	通过 AD0CON0 寄存器的 CH2 ~ CH0 位、AD0CON1 寄存器的 OPA1 ~ OPA0 位、或者 AD0CON2 寄存器的 APS1 ~ APS0 位选择 1 个引脚，将该引脚的输入电压重复进行 A/D 转换。
开始条件	<p>当 AD0CON0 寄存器的 TRG 位为 “0”（软件触发）时</p> <ul style="list-style-type: none"> <li>将 AD0CON0 寄存器的 ADST 位置 “1”（开始 A/D 转换）。</li> </ul> <p>当 TRG 位为 “1”（外部触发 / 硬件触发）时</p> <ul style="list-style-type: none"> <li>在将 ADST 位置 “1” 后，ADTRG 引脚的输入信号从 “H” 电平变为 “L” 电平。</li> <li>在将 ADST 位置 “1” 后，发生三相马达控制定时器功能的（经过中断发生频率的设定电路后）定时器 B2 的中断请求。</li> </ul>
停止条件	<ul style="list-style-type: none"> <li>在将 ADST 位置 “0”（停止 A/D 转换）时。</li> </ul>
中断请求的发生时序	<ul style="list-style-type: none"> <li>在 AD0CON3 寄存器的 DUS 位为 “0”（DMAC 利用模式无效）时不发生。</li> <li>在 DUS 位为 “1”（DMAC 利用模式有效）并且各 A/D 转换结束时发生。</li> </ul>
输入引脚	从 AN_0 ~ AN_7、AN0_0 ~ AN0_7、AN2_0 ~ AN2_7、AN15_0 ~ AN15_7、ANEX0、ANEX1 中选择 1 个引脚。
A/D 转换值的读	<p>当 AD0CON3 寄存器的 DUS 位为 “0”（DMAC 利用模式无效）时</p> <p>读与所选引脚对应的 AD0j 寄存器（j=0 ~ 7）。</p> <p>当 DUS 位为 “1”（DMAC 利用模式有效）时</p> <p>不能读 AD00 寄存器。</p> <p>在 A/D 转换结束后，将 A/D 转换结果保存到 AD00 寄存器，并通过 DMAC 将此结果从 AD00 寄存器传送到任意的存储空间。请参照 “13. DMAC” 设定 DMAC。</p>



### 19.1.3 单次扫描模式

这是将所选引脚的输入电压逐次进行 A/D 转换的模式。单次扫描模式的规格如表 19.4 所示。

表 19.4 单次扫描模式的规格

项目	规格
功能	通过 AD0CON1 寄存器的 SCAN1 ~ SCAN0 位和 AD0CON2 寄存器的 APS1 ~ APS0 位选择引脚，将这些引脚的输入电压逐次进行 A/D 转换。
开始条件	<p>当 AD0CON0 寄存器的 TRG 位为 “0”（软件触发）时</p> <ul style="list-style-type: none"> <li>将 AD0CON0 寄存器的 ADST 位置 “1”（开始 A/D 转换）。</li> </ul> <p>当 TRG 位为 “1”（外部触发 / 硬件触发）时</p> <ul style="list-style-type: none"> <li>在将 ADST 位置 “1” 后，<math>\overline{\text{ADTRG}}</math> 引脚的输入信号从 “H” 电平变为 “L” 电平。</li> <li>在将 ADST 位置 “1” 后，发生三相马达控制定时器功能的（经过中断发生频率的设定电路后）定时器 B2 的中断请求。</li> </ul>
停止条件	<ul style="list-style-type: none"> <li>在 A/D 转换结束（当选择软件触发时，ADST 位为 “0”）时。</li> <li>在将 ADST 位置 “0”（停止 A/D 转换）时。</li> </ul>
中断请求的发生时序	<ul style="list-style-type: none"> <li>在 AD0CON3 寄存器的 DUS 位为 “0”（DMAC 利用模式无效）并且扫描结束时发生。</li> <li>在 DUS 位为 “1”（DMAC 利用模式有效）并且各 A/D 转换结束时发生。</li> </ul>
输入引脚	从 ANi_0 (i= 无,0,2,15) ~ ANi_1 (2 个引脚)、ANi_0 ~ ANi_3 (4 个引脚)、ANi_0 ~ ANi_5 (6 个引脚) 或者 ANi_0 ~ ANi_7 (8 个引脚) 中选择。
A/D 转换值的读	<p>当 AD0CON3 寄存器的 DUS 位为 “0”（DMAC 利用模式无效）时</p> <p>读与所选引脚对应的 AD0j 寄存器 (j=0 ~ 7)。</p> <p>当 DUS 位为 “1”（DMAC 利用模式有效）时</p> <p>不能读 AD00 寄存器。</p> <p>在 A/D 转换结束后，将 A/D 转换结果保存到 AD00 寄存器，并通过 DMAC 将此结果从 AD00 寄存器传送到任意的存储空间。请参照 “13. DMAC” 设定 DMAC。</p>

### 19.1.4 重复扫描模式 0

这是将所选引脚的输入电压重复进行 A/D 转换的模式。重复扫描模式 0 的规格如表 19.5 所示。

表 19.5 重复扫描模式 0 的规格

项目	规格
功能	通过 AD0CON1 寄存器的 SCAN1 ~ SCAN0 位和 AD0CON2 寄存器的 APS1 ~ APS0 位选择引脚，将这些引脚的输入电压重复进行 A/D 转换。
开始条件	<p>当 AD0CON0 寄存器的 TRG 位为 “0”（软件触发）时</p> <ul style="list-style-type: none"> <li>将 AD0CON0 寄存器的 ADST 位置 “1”（开始 A/D 转换）。</li> </ul> <p>当 TRG 位为 “1”（外部触发 / 硬件触发）时</p> <ul style="list-style-type: none"> <li>在将 ADST 位置 “1” 后，<math>\overline{\text{ADTRG}}</math> 引脚的输入信号从 “H” 电平变为 “L” 电平。</li> <li>在将 ADST 位置 “1” 后，发生三相马达控制定时器功能的（经过中断发生频率的设定电路后）定时器 B2 的中断请求。</li> </ul>
停止条件	<ul style="list-style-type: none"> <li>在将 ADST 位置 “0”（停止 A/D 转换）时。</li> </ul>
中断请求的发生时序	<ul style="list-style-type: none"> <li>在 AD0CON3 寄存器的 DUS 位为 “0”（DMAC 利用模式无效）时不发生。</li> <li>在 DUS 位为 “1”（DMAC 利用模式有效）并且各 A/D 转换结束时发生。</li> </ul>
输入引脚	从 ANi_0 (i= 无,0,2,15) ~ ANi_1 (2 个引脚)、ANi_0 ~ ANi_3 (4 个引脚)、ANi_0 ~ ANi_5 (6 个引脚) 或者 ANi_0 ~ ANi_7 (8 个引脚) 中选择。
A/D 转换值的读	<p>当 AD0CON3 寄存器的 DUS 位为 “0”（DMAC 利用模式无效）时 读与所选引脚对应的 AD0j 寄存器 (j=0 ~ 7)。</p> <p>当 DUS 位为 “1”（DMAC 利用模式有效）时 不能读 AD00 寄存器。 在 A/D 转换结束后，将 A/D 转换结果保存到 AD00 寄存器，并通过 DMAC 将此结果从 AD00 寄存器传送到任意的存储空间。请参照 “13. DMAC” 设定 DMAC。</p>

### 19.1.5 重复扫描模式 1

这是以所选的引脚为重点，将 8 个引脚的输入电压重复进行 A/D 转换的模式。重复扫描模式 1 的规格如表 19.6 所示。

表 19.6 重复扫描模式 1 的规格

项目	规格
功能	<p>以通过 AD0CON1 寄存器的 SCAN1 ~ SCAN0 位和 AD0CON2 寄存器的 APS1 ~ APS0 位选择的引脚为重点，将 8 个引脚的输入电压重复进行 A/D 转换。</p> <p>例：当选择 AN_0 时，按 AN_0→AN_1→AN_0→AN_2→AN_0→AN_3……的顺序进行 A/D 转换。</p>
开始条件	<p>当 AD0CON0 寄存器的 TRG 位为 “0”（软件触发）时</p> <ul style="list-style-type: none"> <li>将 AD0CON0 寄存器的 ADST 位置 “1”（开始 A/D 转换）。</li> </ul> <p>当 TRG 位为 “1”（外部触发 / 硬件触发）时（外部触发的重新触发无效）</p> <ul style="list-style-type: none"> <li>在 ADST 位置 “1” 后，ADTRG 引脚的输入信号从 “H” 电平变为 “L” 电平。</li> <li>在将 ADST 位置 “1” 后，发生三相马达控制定时器功能的（经过中断发生频率的设定电路后）定时器 B2 的中断请求。</li> </ul>
停止条件	<ul style="list-style-type: none"> <li>在将 ADST 位置 “0”（停止 A/D 转换）时。</li> </ul>
中断请求的发生时序	<ul style="list-style-type: none"> <li>在 AD0CON3 寄存器的 DUS 位为 “0”（DMAC 利用模式无效）时不发生。</li> <li>在 DUS 位为 “1”（DMAC 利用模式有效）并且各 A/D 转换结束时发生。</li> </ul>
输入引脚	ANi_0 ~ ANi_7（8 个引脚）（i= 无,0,2,15）
重点进行 A/D 转换的引脚	从 ANi_0（1 个引脚）、ANi_0 ~ ANi_1（2 个引脚）、ANi_0 ~ ANi_2（3 个引脚）、ANi_0 ~ ANi_3（4 个引脚）中选择。
A/D 转换值的读	<p>当 AD0CON3 寄存器的 DUS 位为 “0”（DMAC 利用模式无效）时</p> <p>读与所选引脚对应的 AD0j 寄存器（j=0 ~ 7）。</p> <p>当 DUS 位为 “1”（DMAC 利用模式有效）时</p> <p>不能读 AD00 寄存器。</p> <p>在 A/D 转换结束后，将 A/D 转换结果保存到 AD00 寄存器，并通过 DMAC 将此结果从 AD00 寄存器传送到任意的存储空间。请参照 “13. DMAC” 设定 DMAC。</p>

### 19.1.6 多端口单次扫描模式

这是将所选的 16 个引脚的输入电压逐次进行 A/D 转换的模式。

必须将 AD0CON3 寄存器的 DUS 位置 “1” (DMAC 利用模式有效)。

多端口单次扫描模式的规格如表 19.7 所示。

表 19.7 多端口单次扫描模式的规格

项目	规格
功能	按 AN_0 ~ AN_7 → AN <sub>i</sub> _0 ~ AN <sub>i</sub> _7 (i=0,2,15) 的顺序, 将通过 AD0CON4 寄存器的 MPS11 ~ MPS10 位选择的 16 个引脚的输入电压逐次进行 A/D 转换。 例: 当 MPS11 ~ MPS10 位为 “10b” (AN_0 ~ AN_7、AN0_0 ~ AN0_7) 时, 按 AN_0 → AN_1 → AN_2 → AN_3 → AN_4 → AN_5 → AN_6 → AN_7 → AN0_0 → …… → AN0_6 → AN0_7 的顺序进行 A/D 转换。
开始条件	当 AD0CON0 寄存器的 TRG 位为 “0” (软件触发) 时 • 将 AD0CON0 寄存器的 ADST 位置 “1” (开始 A/D 转换)。 当 TRG 位为 “1” (外部触发 / 硬件触发) 时 • 在将 ADST 位置 “1” 后, $\overline{\text{ADTRG}}$ 引脚的输入信号从 “H” 电平变为 “L” 电平。 • 在将 ADST 位置 “1” 后, 发生三相马达控制定时器功能的 (经过中断发生频率的设定电路后) 定时器 B2 的中断请求。
停止条件	• 在 A/D 转换结束 (当选择软件触发时, ADST 位为 “0”) 时。 • 在将 ADST 位置 “0” (停止 A/D 转换) 时。
中断请求的发生时序	在各 A/D 转换结束时发生 (必须将 DUS 位置 “1”)。
输入引脚	从 AN_0 ~ AN_7 → AN15_0 ~ AN15_7、AN_0 ~ AN_7 → AN0_0 ~ AN0_7、AN_0 ~ AN_7 → AN2_0 ~ AN2_7 中选择。
A/D 转换值的读	不能读 AD00 寄存器。 必须将 DUS 位置 “1”。 在 A/D 转换结束后, 将 A/D 转换结果保存到 AD00 寄存器, 并通过 DMAC 将此结果从 AD00 寄存器传送到任意的存储空间。请参照 “13. DMAC” 设定 DMAC。

### 19.1.7 多端口重复扫描模式 0

这是将所选的 16 个引脚的输入电压重复进行 A/D 转换的模式。

必须将 AD0CON3 寄存器的 DUS 位置“1”（DMAC 利用模式有效）。

多端口重复扫描模式 0 的规格如表 19.8 所示。

表 19.8 多端口重复扫描模式 0 的规格

项目	规格
功能	按 AN_0 ~ AN_7 → AN <sub>i</sub> _0 ~ AN <sub>i</sub> _7 (i=0,2,15) 的顺序, 将通过 AD0CON4 寄存器的 MPS11 ~ MPS10 位选择的 16 个引脚的输入电压重复进行 A/D 转换。 例: 当 MPS11 ~ MPS10 位为“10b”(AN_0 ~ AN_7、AN0_0 ~ AN0_7) 时, 按 AN_0 → AN_1 → AN_2 → AN_3 → AN_4 → AN_5 → AN_6 → AN_7 → AN0_0 → AN0_1 → …… → AN0_6 → AN0_7 的顺序重复进行 A/D 转换。
开始条件	当 AD0CON0 寄存器的 TRG 位为“0”(软件触发) 时 • 将 AD0CON0 寄存器的 ADST 位置“1”(开始 A/D 转换)。 当 TRG 位为“1”(外部触发/硬件触发) 时 • 在将 ADST 位置“1”后, $\overline{\text{ADTRG}}$ 引脚的输入信号从“H”电平变为“L”电平。 • 在将 ADST 位置“1”后, 发生三相马达控制定时器功能的(经过中断发生频率的设定电路后) 定时器 B2 的中断请求。
停止条件	• 在将 ADST 位置“0”(停止 A/D 转换) 时。
中断请求的发生时序	在各 A/D 转换结束时发生(必须将 DUS 位置“1”)。
输入引脚	从 AN_0 ~ AN_7 → AN15_0 ~ AN15_7、AN_0 ~ AN_7 → AN0_0 ~ AN0_7、AN_0 ~ AN_7 → AN2_0 ~ AN2_7 中选择。
A/D 转换值的读	不能读 AD00 寄存器。 必须将 DUS 位置“1”。 在 A/D 转换结束后, 将 A/D 转换结果保存到 AD00 寄存器, 并通过 DMAC 将此结果从 AD00 寄存器传送到任意的存储空间。请参照“13. DMAC”设定 DMAC。

## 19.2 功能

### 19.2.1 分辨率的选择功能

能通过 AD0CON1 寄存器的 BITS 位选择分辨率。如果将 BITS 位置“1”（转换精度为 10 位），就将 A/D 转换结果保存到 AD0i 寄存器（i=0 ~ 7）的 bit0 ~ bit9；如果将 BITS 位置“0”（转换精度为 8 位），就将 A/D 转换结果保存到 AD0i 寄存器的 bit0 ~ bit7。

### 19.2.2 采样 & 保持

如果将 AD0CON2 寄存器的 SMP 位置“1”（有采样 & 保持），就会提高每个引脚的转换速度。当分辨率是 8 位时，转换速度为 28 个  $\phi_{AD}$  周期；当分辨率是 10 位时，转换速度为 33 个  $\phi_{AD}$  周期。采样 & 保持对于全部的运行模式有效。必须在选择有无采样 & 保持后开始 A/D 转换。

### 19.2.3 触发的选择功能

能通过 AD0CON0 寄存器的 TRG 位和 AD0CON2 寄存器的 TRG0 位的组合，选择 A/D 转换的开始触发。触发选择功能的设定如表 19.9 所示。

表 19.9 触发选择功能的设定

位和设定值		触发
AD0CON0 寄存器	AD0CON2 寄存器	
TRG=0	—	软件触发 如果将 AD0CON0 寄存器的 ADST 位置“1”，就开始 A/D 转换。
TRG=1 (注 1、注 2)	TRG0=0	外部触发 在 $\overline{ADTRG}$ 输入信号的下降沿开始 A/D 转换。
	TRG0=1	硬件触发 通过三相模式控制的定时器功能的（经过中断发生频率的设定电路后）定时器 B2 的中断请求，开始 A/D 转换。

注 1. 如果在 ADST 位为“1”（开始 A/D 转换）的状态下发生触发，就开始 A/D 转换。

注 2. 如果在 A/D 转换过程中输入外部触发或者硬件触发，就停止此前进行的 A/D 转换，重新开始 A/D 转换。

### 19.2.4 DMAC 利用模式

DMAC 利用模式能用于全部模式。在多端口单次扫描模式和多端口重复扫描模式 0 的情况下，必须使用 DMAC 利用模式。如果将 AD0CON3 寄存器的 DUS 位置“1”（DMAC 利用模式有效），就将 A/D 转换结果全部保存到 AD00 寄存器。通过使用 DMAC，在每个引脚的 A/D 转换结束时进行从 AD00 寄存器到任意的存储空间的 DMA 传送。当分辨率为 8 位时，必须设定 8 位的传送；当分辨率为 10 位时，必须设定 16 位的传送。有关 DMAC 的使用方法，请参照“13. DMAC”。

### 19.2.5 扩展模拟输入引脚

在单次模式和重复模式中，能将 ANEX0 和 ANEX1 引脚用作模拟输入引脚，必须通过 AD0CON1 寄存器的 OPA1 ~ OPA0 位进行选择（参照表 19.10）。将 ANEX0 输入和 ANEX1 输入的 A/D 转换结果分别保存到 AD00 寄存器和 AD01 寄存器。但是，在 AD0CON3 寄存器的 DUS 位为“1”（DMAC 利用模式有效）时，将 A/D 转换结果保存到 AD00 寄存器。

在使用扩展模拟输入引脚时，必须将 AD0CON2 寄存器的 APS1 ~ APS0 位置“00b”（模拟输入端口为 AN\_0 ~ AN\_7、ANEX0、ANEX1）并且将 AD0CON3 寄存器的 MSS 位置“0”（多端口扫描模式无效）。

表 19.10 扩展模拟输入引脚的设置

AD0CON1 寄存器		ANEX0 的功能	ANEX1 的功能
OPA1	OPA0		
0	0	不使用	不使用
0	1	模拟输入	不使用
1	0	不使用	模拟输入
1	1	外部运算放大器的输出	外部运算放大器的输入

### 19.2.6 外部运算放大器的连接模式

能使用扩展模拟输入引脚 ANEX0 和 ANEX1 并通过 1 个外部运算放大器，将多个模拟输入的信号进行放大。

在 AD0CON1 寄存器的 OPA1 ~ OPA0 位为“11b”（连接外部运算放大器）时，从 ANEX0 引脚输出 AN\_0 ~ AN\_7 的输入信号，通过外部运算放大器将此输出信号放大，然后输入到 ANEX1 引脚。

对 ANEX1 输入进行 A/D 转换，并将 A/D 转换结果保存到对应的 AD0i 寄存器（i=0 ~ 7）。A/D 转换速度取决于外部运算放大器的响应特性。另外，不能将 ANEX0 引脚和 ANEX1 引脚直接连接。

在使用外部运算放大器连接模式时，必须将 AD0CON2 寄存器的 APS1 ~ APS0 位置“00b”。

外部运算放大器连接模式的连接例子如图 19.8 所示。

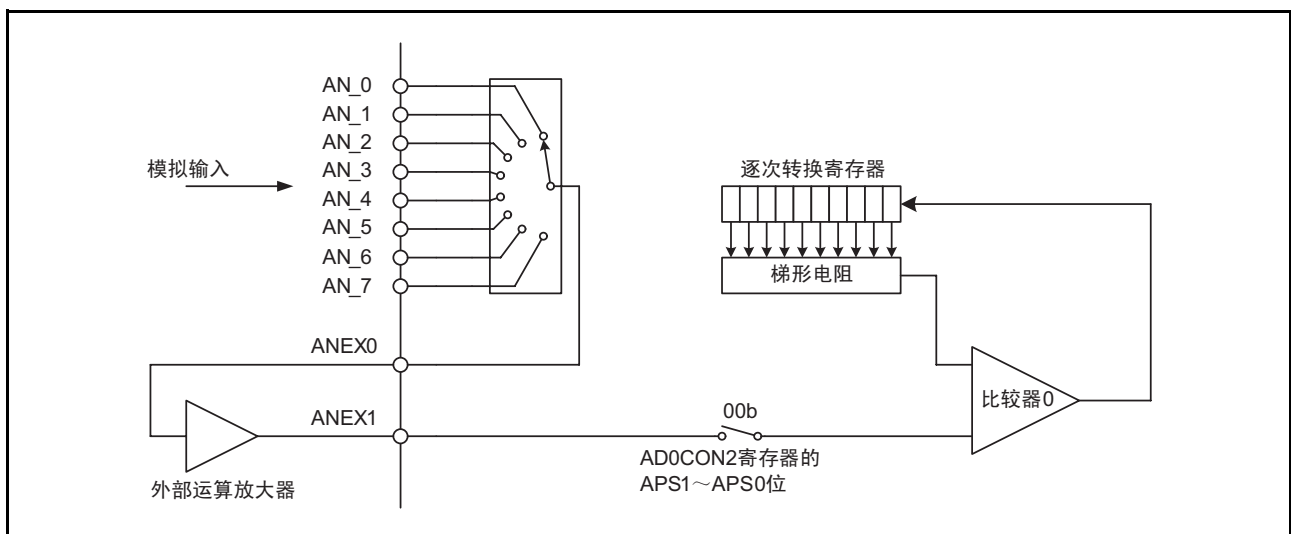


图 19.8 外部运算放大器连接模式的连接例子

### 19.2.7 降低消耗电流的功能

在不使用 A/D 转换器时，能通过将 AD0CON1 寄存器的 VCUT 位置“0”（断开 VREF），使 A/D 转换器的梯形电阻和基准电压输入引脚（VREF）断开。一旦断开，提供的电流就不从 VREF 引脚流到梯形电阻，从而能降低功耗。

在使用 A/D 转换器时，必须在将 VCUT 位置“1”（连接 VREF）并至少经过  $1\mu\text{s}$  后，将 AD0CON0 寄存器的 ADST 位置“1”（开始 A/D 转换）。不能给 ADST 位和 VCUT 位同时写“1”。

不能在 A/D 转换过程中将 VCUT 位置“0”。

另外，VCUT 位不影响 D/A 转换器的供电（参照图 19.9）。

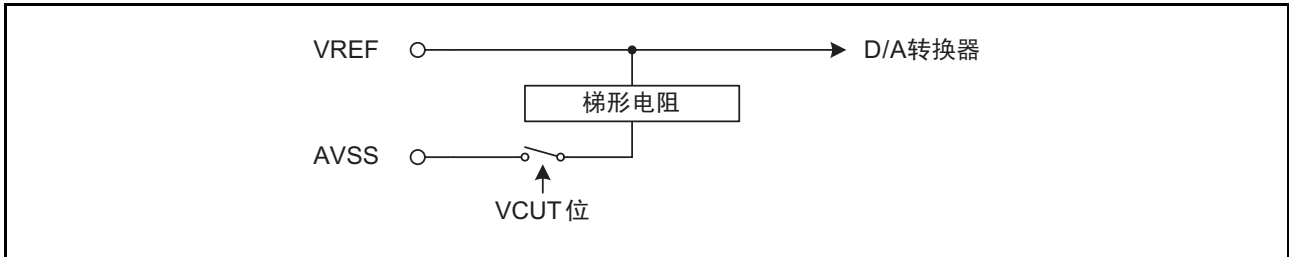


图 19.9 VCUT 位的供电

### 19.2.8 传感器的输出阻抗

模拟输入引脚和外部传感器的等效电路如图 19.10 所示。

为了正确地进行 A/D 转换，必须在规定的时间内结束对图 19.10 所示的内部电容器 C 的充电。将此规定的时间称为采样时间，在没有采样 & 保持时为 2 个  $\phi_{AD}$  周期，在有采样 & 保持时为 3 个  $\phi_{AD}$  周期。

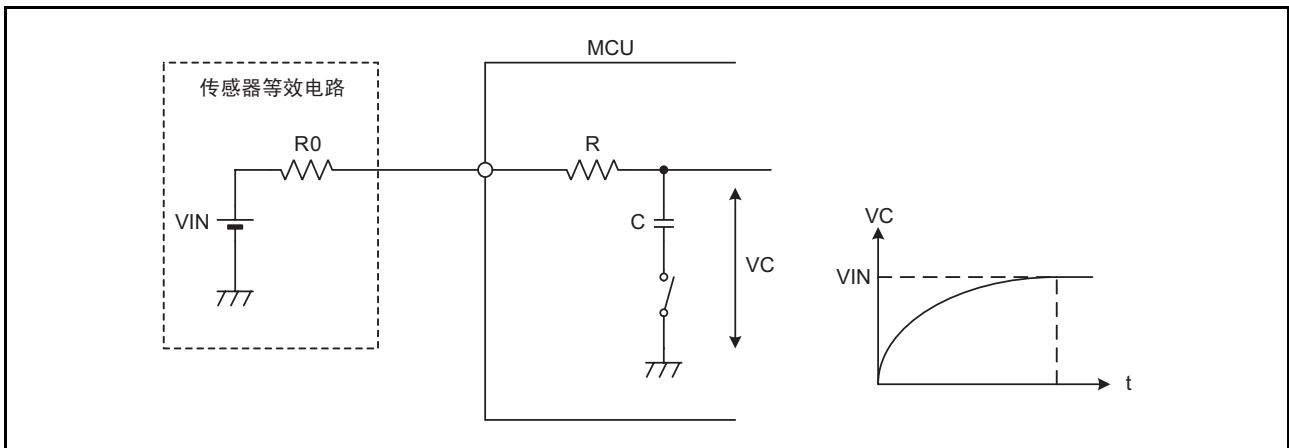


图 19.10 模拟输入引脚和外部传感器的等效电路



假设采样时间为  $T[s]$ ，传感器的输出阻抗为  $R_0[\Omega]$ ，单片机内部电阻为  $R[\Omega]$ ，A/D 转换器的精度（误差）为  $x[\text{LSB}]$ ，分辨率为  $y$  [阶梯]（在 10 位模式中  $y$  为 1024，在 8 位模式中  $y$  为 256），则电容器  $C$  两端的电位差  $V_C$  用下式表示：

$$V_C = VIN \left\{ 1 - e^{-\frac{t}{C(R_0+R)}} \right\}$$

当  $t=T$  时，为了使转换误差小于等于  $x$ ，必须是：

$$V_C = VIN - \frac{x}{y}VIN = VIN \left( 1 - \frac{x}{y} \right)$$

所以，能用下式求出：

$$e^{-\frac{T}{C(R_0+R)}} = \frac{x}{y}$$

$$-\frac{T}{C(R_0+R)} = \ln \frac{x}{y}$$

$$R_0 = -\frac{T}{C \ln \frac{x}{y}} - R$$

当  $\phi_{AD}=10\text{MHz}$ ，有采样 & 保持并且是 10 位的模式时，将  $T=0.3\mu\text{s}$ ， $x=0.1$ ， $y=1024$ ， $R=2.0\text{k}\Omega$ （参考值）， $C=6.5\text{pF}$ （参考值）代入下式，能求出误差小于等于  $0.1\text{LSB}$  的传感器的输出阻抗  $R_0$ 。

$$R_0 = -\frac{0.3 \times 10^{-6}}{6.5 \times 10^{-12} \cdot \ln \frac{0.1}{1024}} - 2.0 \times 10^3$$

$$= 2998$$

由上述得知，要使 A/D 转换器的精度（误差）小于等于  $0.1\text{LSB}$ ，传感器的输出阻抗  $R_0$  最大为  $3\text{k}\Omega$ 。另外，实际误差为上述的  $0.1\text{LSB}$  加上绝对精度后的值。

## 19.3 使用 A/D 转换器时的注意事项

### 19.3.1 设计电路板时的注意事项

- 为了防止噪声引起的误动作和闩锁以及减少转换误差，必须在 AVCC 引脚、VREF 引脚、模拟输入引脚（AN\_0~AN\_7、AN0\_0~AN0\_7、AN2\_0~AN2\_7、AN15\_0~AN15\_7）和 AVSS 引脚之间分别插入电容器。引脚的处理例子如图 19.11 所示。

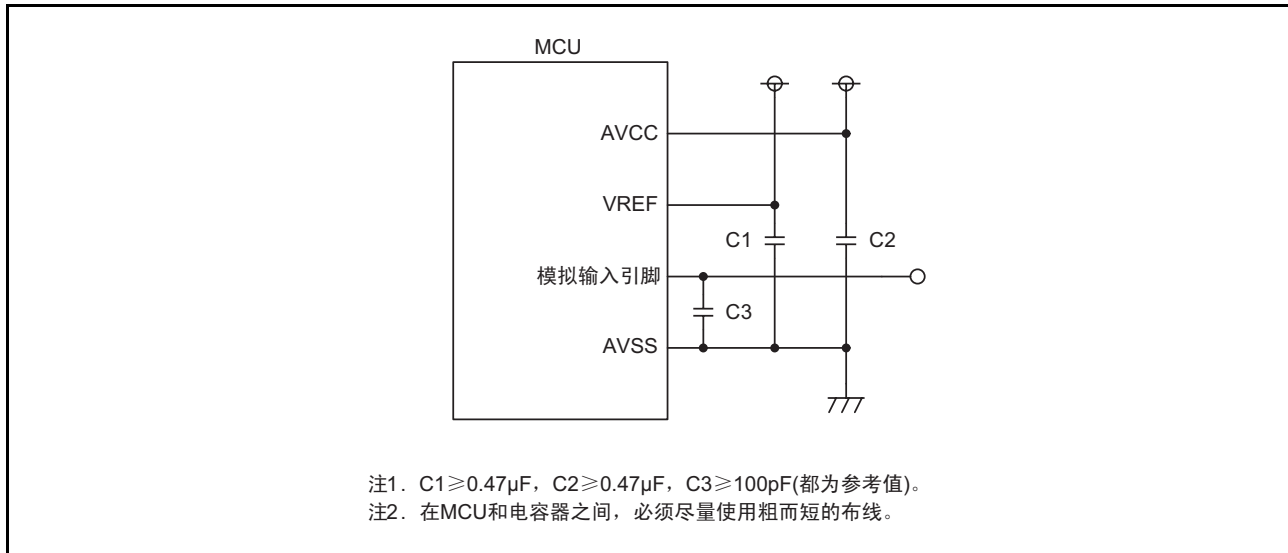


图 19.11 A/D 相关引脚的处理例子

- 在使用键输入中断时，AN\_4~AN\_7（4个引脚）都不能用作模拟输入引脚（如果A/D输入电压不超过VIL，就产生键输入中断请求）。
- 当AVCC=VREF=VCC时，AN\_0~AN\_7、AN0\_0~AN0\_7、AN2\_0~AN2\_7、AN15\_0~AN15\_7、ANEX0、ANEX1的A/D输入电压不能超过VCC。

### 19.3.2 编程时的注意事项

- 必须在停止 A/D 转换时（发生触发前）写 AD0CON0 寄存器（ADST 位除外）、AD0CON1 寄存器、AD0CON2 寄存器、AD0CON3 寄存器和 AD0CON4 寄存器。
- 在将 AD0CON1 寄存器的 VCUT 位从“0”（未连接 VREF）置为“1”（连接 VREF）时，必须在至少经过 1 $\mu$ s 后开始 A/D 转换。在不进行 A/D 转换时，为了减少消耗电流，必须将 VCUT 位从“1”置为“0”。
- 必须将用于模拟输入引脚对应的端口方向位置“0”（输入），并将对应的端口功能选择寄存器的 ASEL 位置“1”（用作 A/D 输入）。
- 当 AD0CON0 寄存器的 TRG 位为“1”（外部触发/硬件触发）时，必须将 ADTRG 引脚对应的端口方向位（PD9\_7 位）置“0”（输入）。
- 当 VCC=4.2~5.5V 时， $\phi$ AD 的频率不能超过 16MHz；当 VCC=3.0~4.2V 时， $\phi$ AD 的频率不能超过 10MHz。在没有采样 & 保持功能时， $\phi$ AD 的频率至少为 250kHz；在有采样 & 保持功能时， $\phi$ AD 的频率至少为 1MHz。
- 如果更改 A/D 运行模式（AD0CON0 寄存器的 MD1~MD0 位和 AD0CON1 寄存器的 MD2 位），就必须通过 AD0CON0 寄存器的 CH2~CH0 位或者 AD0CON1 寄存器的 SCAN1~SCAN0 位重新选择模拟输入引脚。
- 在将 A/D 转换结果保存到 AD0i 寄存器（i=0~7）时，如果 CPU 读 AD0i 寄存器，就有可能将错误值保存到 AD0i 寄存器。必须在确认 A/D 转换结束后读 AD0i 寄存器。  
在使用单次模式或者单次扫描模式时，必须在确认 AD0IC 寄存器的 IR 位为“1”（有中断请求）后读对象 AD0i 寄存器。  
在使用重复模式、重复扫描模式 0 或者重复扫描模式 1 时，如果将 AD0CON3 寄存器的 DUS 位置“1”（DMAC 利用模式有效），就能在每次转换结束时产生中断请求。同样，必须在确认 AD0IC 寄存器的 IR 位为“1”（有中断请求）后读 AD00 寄存器。
- 在 A/D 转换过程中，如果通过将 AD0CON0 寄存器的 ADST 位置“0”（停止 A/D 转换）来停止 A/D 转换，A/D 转换器的转换结果就为不定值，而且没有进行 A/D 转换的 AD0i 寄存器的值也有可能为不定值。如果停止 A/D 转换，就不能使用任何 AD0i 寄存器的值。
- 在 DMAC 利用模式中，不能使用外部触发，也不能通过程序读 AD00 寄存器。
- 在单次扫描模式中，如果在 A/D 转换过程中通过将 AD0CON0 寄存器的 ADST 位置“0”（停止 A/D 转换）来停止 A/D 转换，就有可能产生中断请求，与扫描是否结束无关。在停止 A/D 转换时，必须在禁止中断后将 ADST 位置“0”（停止 A/D 转换）。

## 20. D/A 转换器

这是 8 位 R-2R 梯形电阻方式的 D/A 转换器。有 2 个独立的 D/A 转换器。

如果给对应的 DA<sub>i</sub> 寄存器 (i=0,1) 写值, 就进行 D/A 转换。必须通过 DACON 寄存器的 DA<sub>i</sub>E 位选择是否输出转换结果。如果将 DA<sub>i</sub>E 位置 “1” (允许输出), 就从 DA<sub>i</sub> 引脚输出转换结果。此时, 禁止对应端口的上拉。

输出的模拟电压 (V) 取决于 DA<sub>i</sub> 寄存器的设定值 n (n 为 10 进制数)。

$$V = \frac{VREF \times n}{256} \quad (n=0 \sim 255)$$

VREF: 基准电压

D/A 转换器的规格如表 20.1 所示, D/A 转换器的框图及其相关寄存器分别如图 20.1 和图 20.2 ~ 图 20.3 所示, D/A 转换器的等效电路如图 20.4 所示。

在不使用 D/A 转换器时, 必须将 DA<sub>i</sub> 寄存器置 “00h” 并且将 DA<sub>i</sub>E 位置 “0” (禁止输出)。

表 20.1 D/A 转换器的规格

项目	规格
转换方式	R-2R 梯形电阻方式
分辨率	8 位
模拟输出引脚	2 个通道

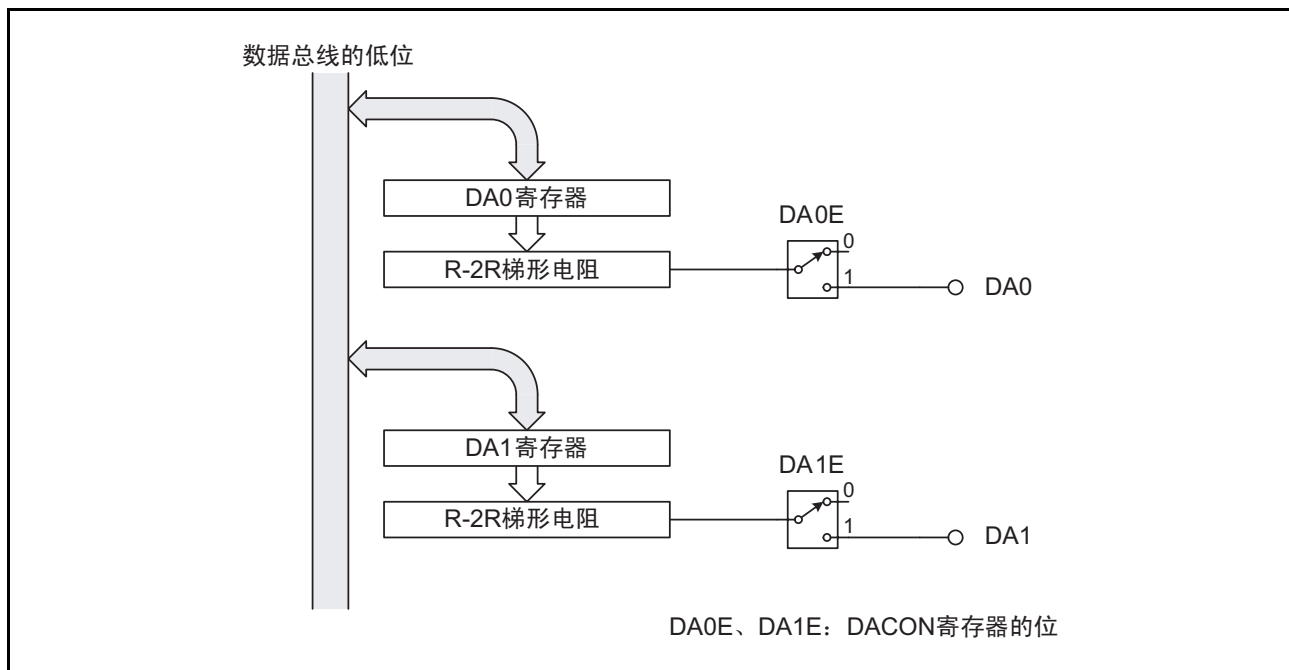


图 20.1 D/A 转换器的框图

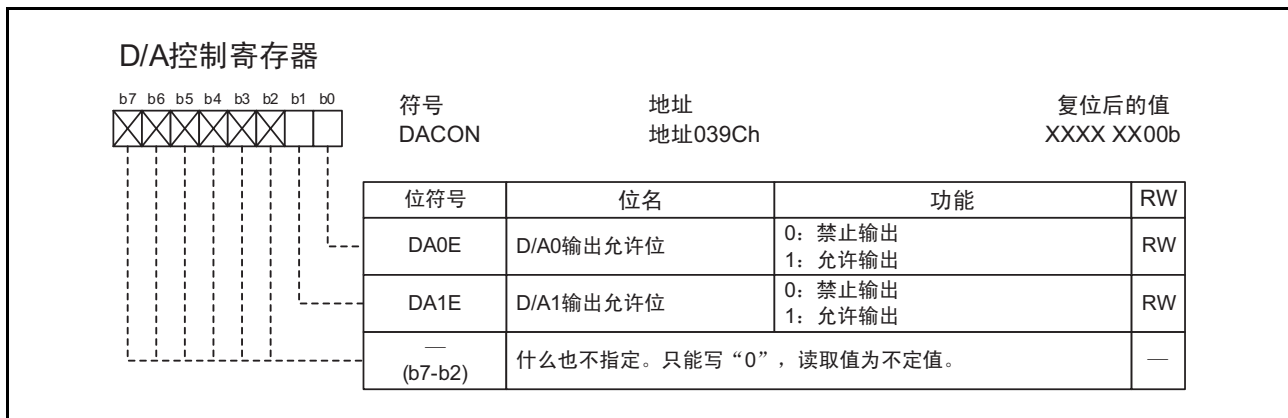


图 20.2 DACON 寄存器



图 20.3 DA0 寄存器和 DA1 寄存器

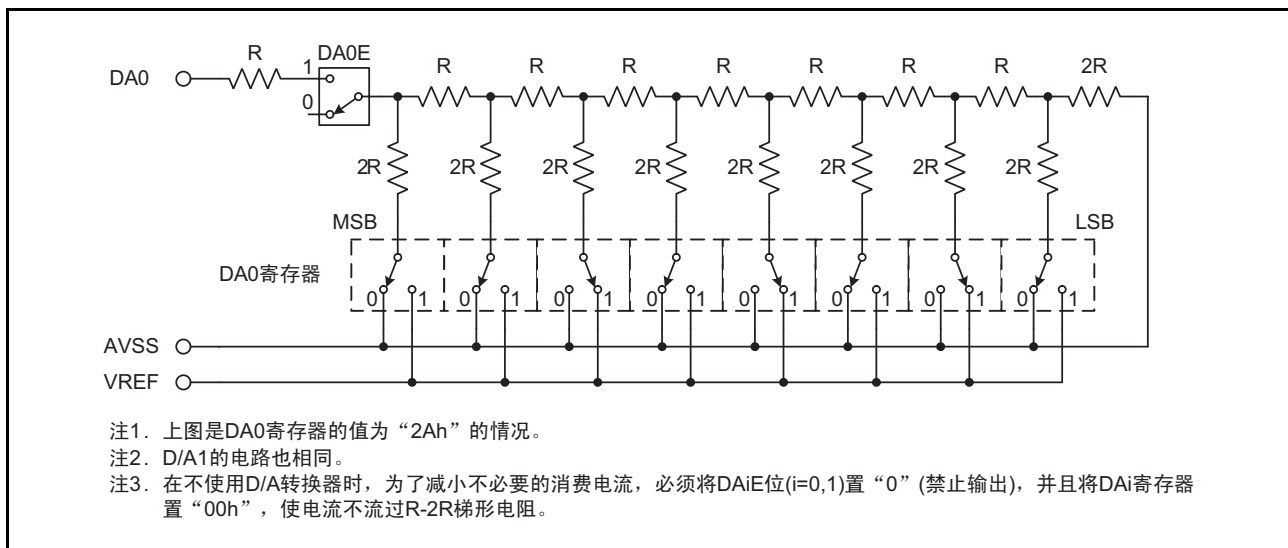


图 20.4 D/A 转换器的等效电路

## 21. CRC 运算电路

CRC (Cyclic Redundancy Check) 运算电路用于检测数据块的错误。在生成 CRC 码时，使用 CRC-CCITT ( $X^{16}+X^{12}+X^5+1$ ) 的生成多项式。

CRC 码是对以 8 位为单位的任意长度的数据块生成的 16 位代码。在给 CRCD 寄存器设定初始值后，每当将 1 字节的数据写到 CRCIN 寄存器时，CRC 码就被设定到 CRCD 寄存器。

CRC 运算电路的框图及其相关寄存器分别如图 21.1 和图 21.2 ~图 21.3 所示，CRC 运算例子如图 21.4 所示。

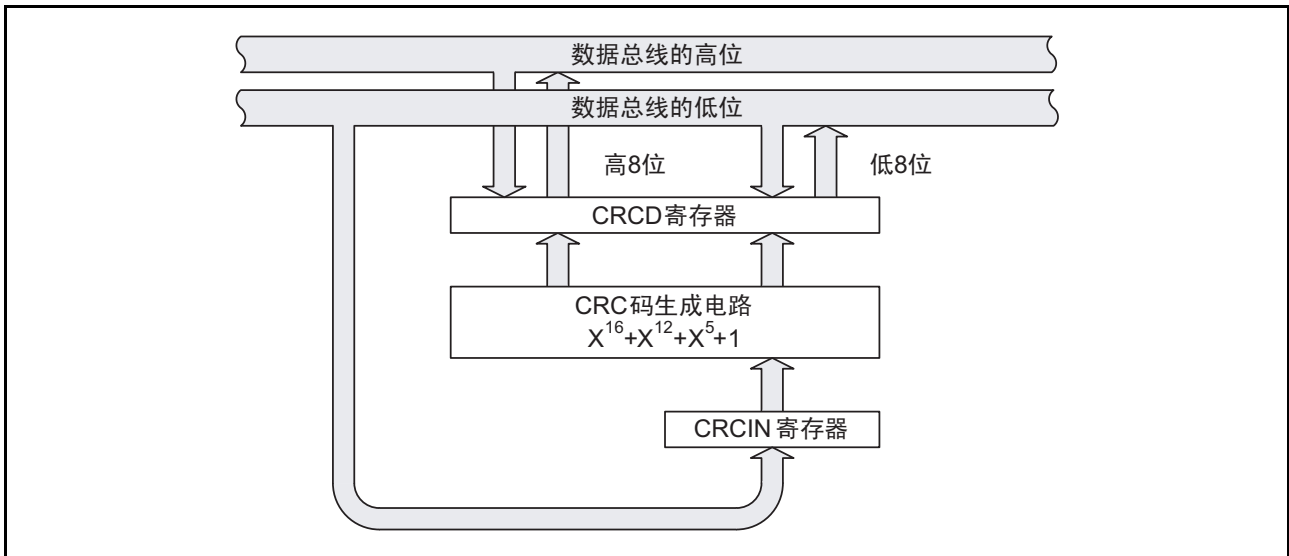


图 21.1 CRC 运算电路的框图

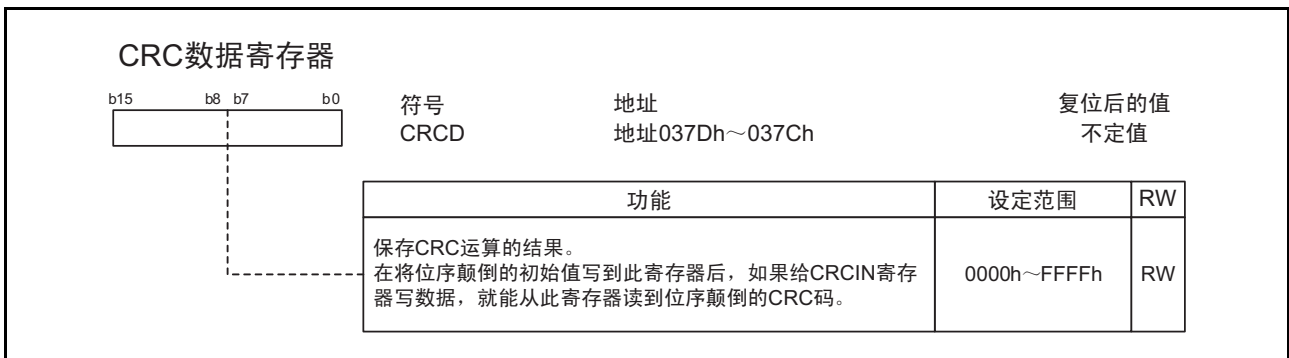


图 21.2 CRCD 寄存器



图 21.3 CRCIN 寄存器

### 生成“80C4h”CRC码的设定步骤和CRC运算

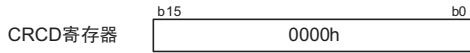
● CRC运算电路的规格

CRC码：以写到CRCIN寄存器的位序颠倒的值为被除数、生成多项式为除数的除法运算的余数  
 生成多项式： $X^{16}+X^{12}+X^5+1$  (1 0001 0000 0010 0001b)

● 设定步骤

- (1) 通过程序，以字节为单位将“80C4h”的位序颠倒。  
 “80h” → “01h”、“C4h” → “23h”

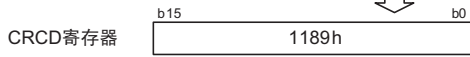
- (2) 将初始值“0000h”写到CRCD寄存器。



- (3) 将“80h”的位序颠倒值“01h”写到CRCIN寄存器。



在2个周期后，“80h”的CRC码(9188h)的位序颠倒值“1189h”被保存到CRCD寄存器。



- (4) 将“C4h”的位序颠倒值“23h”写到CRCIN寄存器。

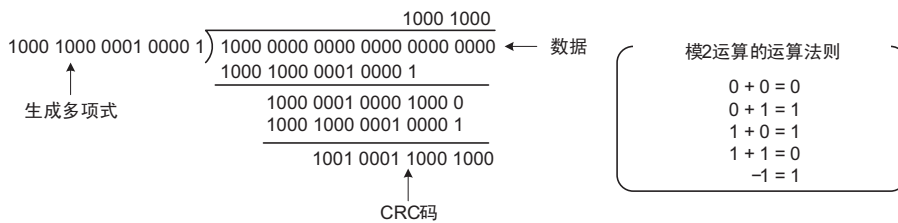


在2个周期后，“80C4h”的CRC码(8250h)的位序颠倒值“0A41h”被保存到CRCD寄存器。



● 详细的CRC运算

上述(3)的情况，将“80h(1000 0000b)”和CRCD寄存器的初始值“0000h”分别左移16位和8位，得到“1000 0000 0000 0000 0000 0000b”和“0000 0000 0000 0000 0000 0000b”，将它们的相加值进行模2的除法运算。



能从CRCD寄存器读到余数“1001 0001 1000 1000b (9188h)”的位序颠倒的“0001 0001 1000 1001b (1189h)”。

上述(4)的情况，将“C4h(1100 0100b)”和留在CRCD寄存器中的(3)的余数分别左移16位和8位，得到“1100 0100 0000 0000 0000 0000b”和“1001 0001 1000 1000 0000 0000b”，将它们的相加值进行模2的除法运算。

能从CRCD寄存器读到余数“1000 0010 0101 0000b (8250h)”的位序颠倒的“0000 1010 0100 0001b (0A41h)”。

图 21.4 CRC 运算例子

## 22. X-Y 转换电路

X-Y 转换电路能将 16×16 位的矩阵数据旋转 90 度或者颠倒 16 位数据的位序。

通过 XYC 寄存器设定 X-Y 转换电路的工作，XYC 寄存器如图 22.1 所示。

从 XiR 寄存器 (i=0 ~ 15) 写数据，从 YjR 寄存器 (j=0 ~ 15) 读被转换后的数据。XiR 寄存器和 YjR 寄存器分配在相同的地址，XiR 是只写寄存器，YjR 是只读寄存器。

XiR 寄存器和 YjR 寄存器分别如图 22.2 和图 22.3 所示。必须从偶数地址开始并以 16 位为单位存取 XiR 寄存器和 YjR 寄存器。如果以 8 位为单位进行存取，运行将处于不确定状态。



图 22.1 XYC 寄存器

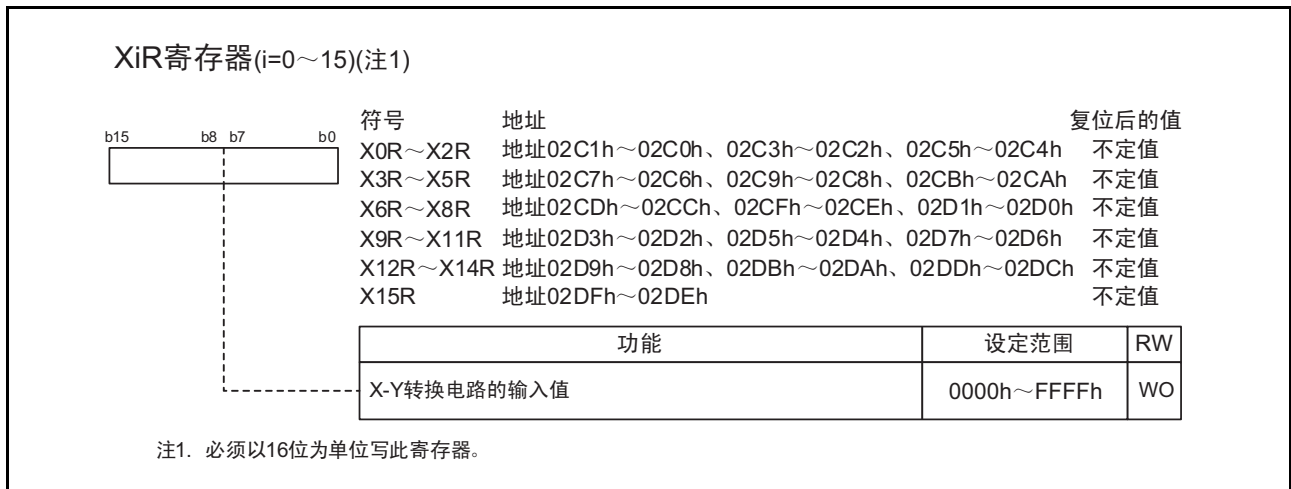


图 22.2 XiR 寄存器



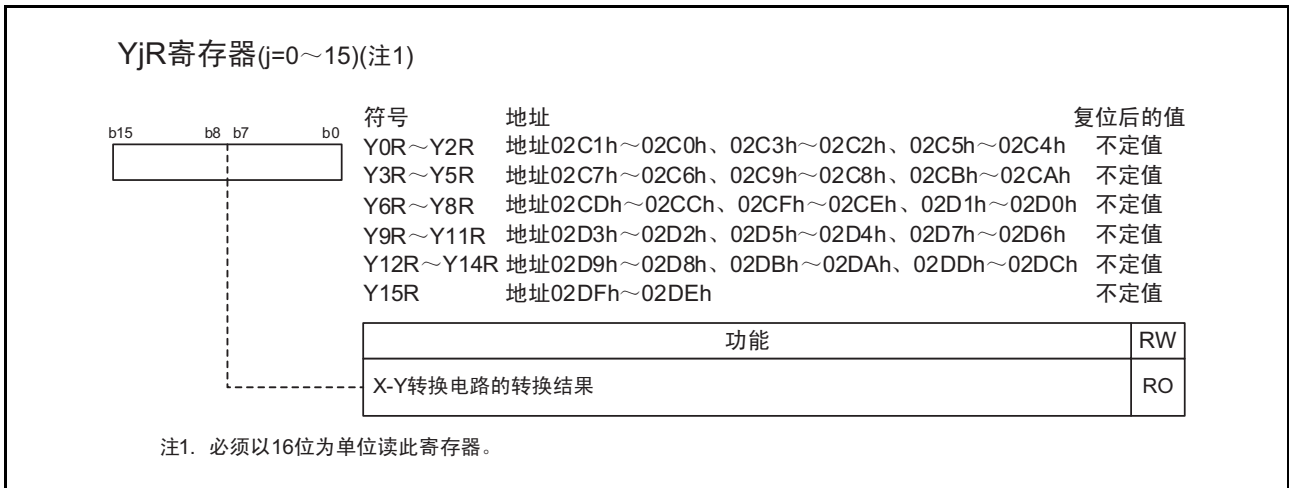


图 22.3 YjR 寄存器

### 22.1 读时的数据转换

能通过 X<sub>Y</sub>C 寄存器的 X<sub>Y</sub>C0 位选择 Y<sub>j</sub>R 寄存器的读取方法。

如果 X<sub>Y</sub>C0 位为 “0”（数据旋转）并且读 Y<sub>j</sub>R 寄存器，就能同时读 X<sub>0</sub>R ~ X<sub>15</sub>R 寄存器的位 j。

例如，如果读 Y<sub>0</sub>R 寄存器，就能通过 bit0 读 X<sub>0</sub>R 寄存器的 bit0，通过 bit1 读 X<sub>1</sub>R 寄存器的 bit0，……，通过 bit14 读 X<sub>14</sub>R 寄存器的 bit0 以及通过 bit15 读 X<sub>15</sub>R 寄存器的 bit0。同样，如果读 Y<sub>15</sub>R 寄存器，就能通过 bit0 读 X<sub>0</sub>R 寄存器的 bit15，通过 bit1 读 X<sub>1</sub>R 寄存器的 bit15，……，通过 bit14 读 X<sub>14</sub>R 寄存器的 bit15 以及通过 bit15 读 X<sub>15</sub>R 寄存器的 bit15。

X<sub>Y</sub>C0 位为 “0” 时的转换表以及 X-Y 转换例子分别如图 22.4 和图 22.5 所示。

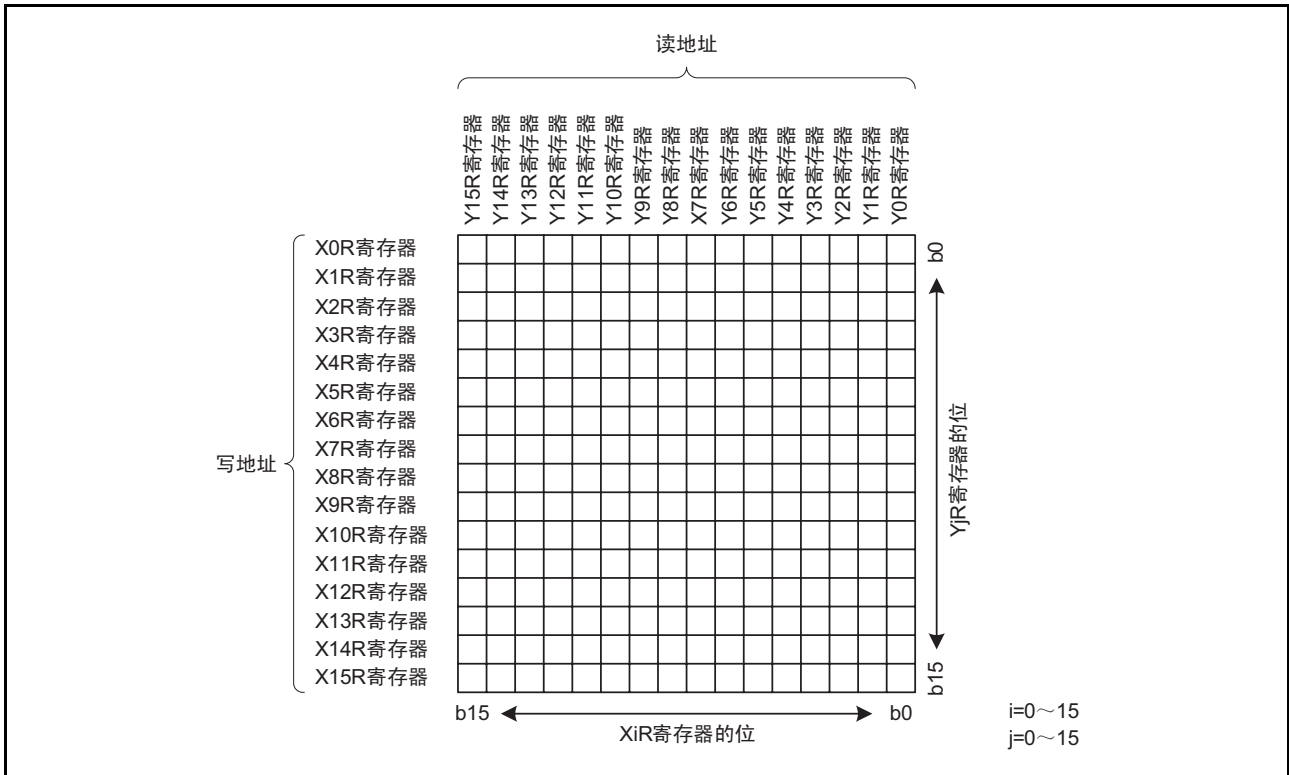


图 22.4 X<sub>Y</sub>C0 位为 “0” 时的转换表

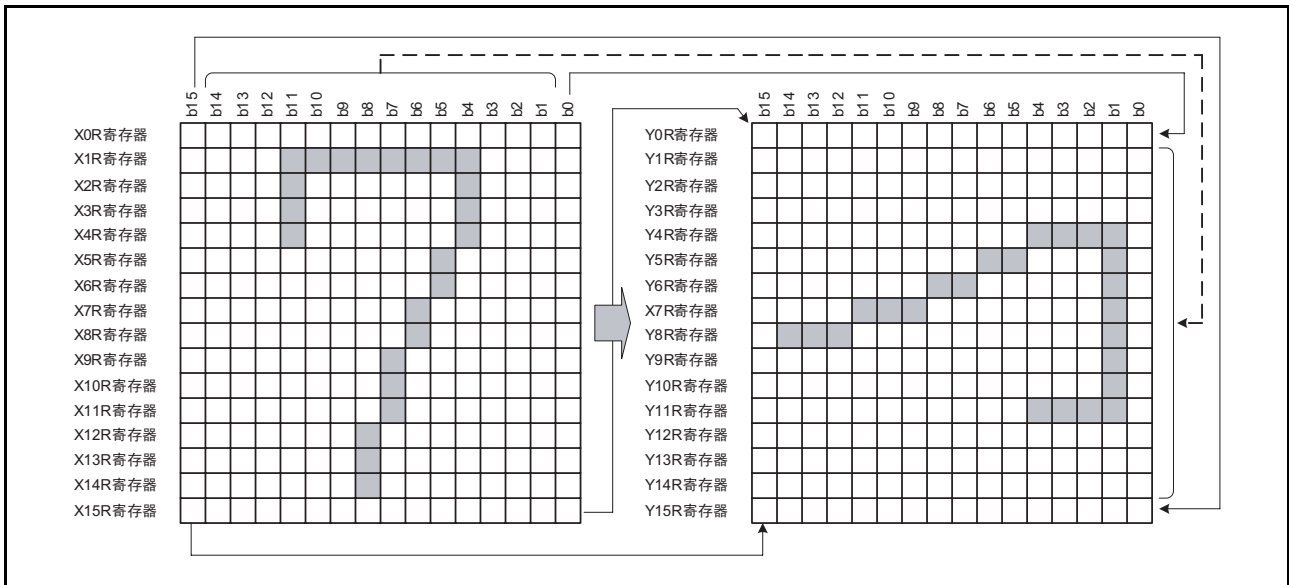


图 22.5 X-Y 转换例子

如果将 XYC 寄存器的 XYC0 位置 “1”（数据不旋转）后读 YjR 寄存器，就能直接读到被写入 XiR 寄存器的值。XYC0 位为 “1” 时的转换表如图 22.6 所示。

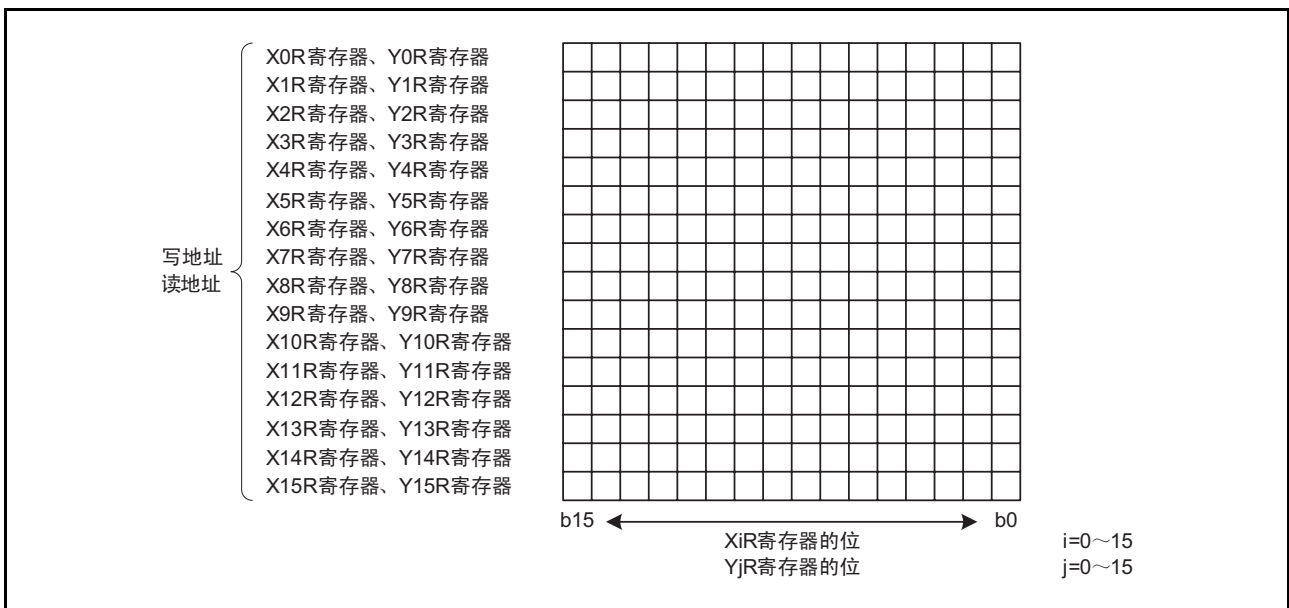


图 22.6 XYC0 位为 “1” 时的转换表

## 22.2 写时的数据转换

能通过 XYC 寄存器的 XYC1 位选择写到 XiR 寄存器的值的位序。

如果将 XYC1 位置 “0”（位序不颠倒）后写 XiR 寄存器，就按位的顺序写值。

如果将 XYC1 位置 “1”（位序颠倒）后写 XiR 寄存器，就颠倒位的顺序写值。XYC1 位为 “1” 时的转换如图 22.7 所示。

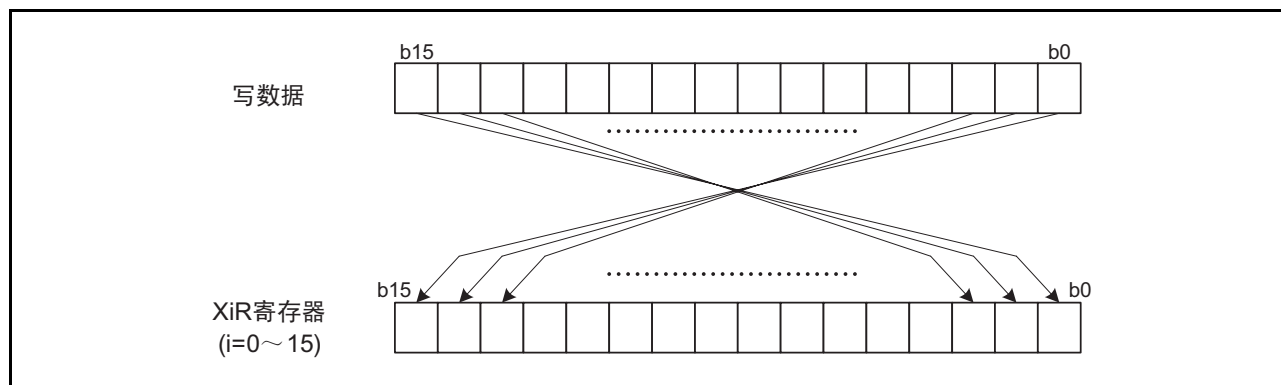


图 22.7 XYC1 位为 “1” 时的转换

## 23. 智能 I/O

智能 I/O 是实现时间测量、波形生成、可变长时钟同步串行 I/O 和 IEBus 通信（选项）的高功能输入/输出端口。

智能 I/O 有 3 组，每组有 1 个 16 位自由运行的基本定时器以及 8 个 16 位用于时间测量或者波形生成的寄存器。

智能 I/O 的功能和通道如表 23.1 所示。

表 23.1 智能 I/O 的功能和通道

功能		组 0	组 1	组 2
时间测量（注 1）	数字滤波器	8 个通道	8 个通道	无
	触发输入预分频器	2 个通道	2 个通道	
	触发输入选通	2 个通道	2 个通道	
波形生成（注 1）	单相波形输出模式	8 个通道	8 个通道	8 个通道
	反相波形输出模式	8 个通道	8 个通道	8 个通道
	SR 波形输出模式	8 个通道	8 个通道	8 个通道
	位调制 PWM 模式	无	无	8 个通道
	RTP 模式			8 个通道
	并行 RTP 模式			8 个通道
通信	可变长时钟同步串行 I/O 模式	无	无	有
	IEBus 模式（选项（注 2））			

注 1. 时间测量功能和波形生成功能引脚复用。

注 2. 如果要使用选项功能，请向本公司的营业窗口询问。

能按通道选择时间测量功能和波形生成功能。

智能 I/O 的框图如图 23.1 ~ 图 23.3 所示。

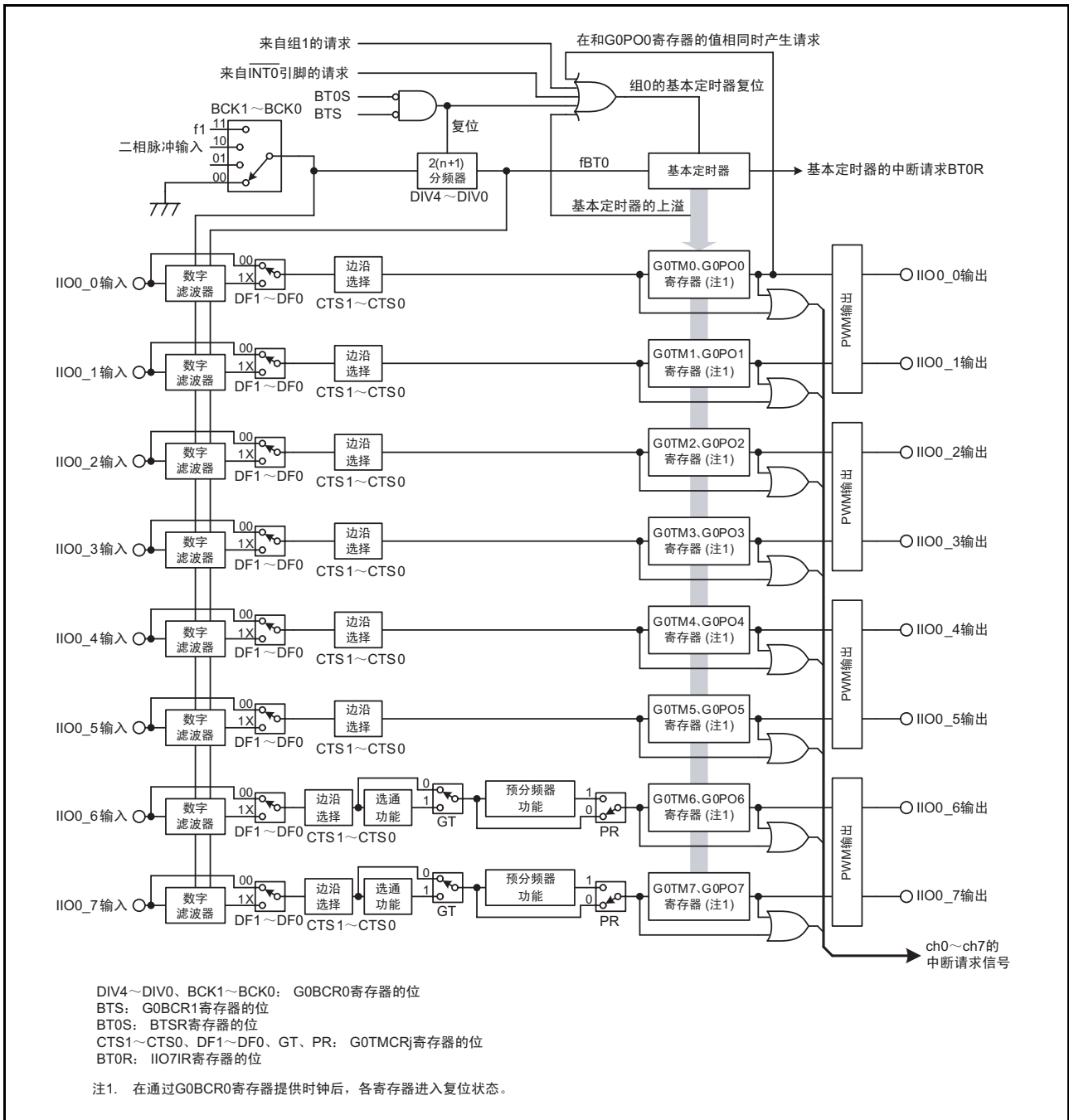


图 23.1 智能 I/O 组 0 的框图 (j=0 ~ 7)

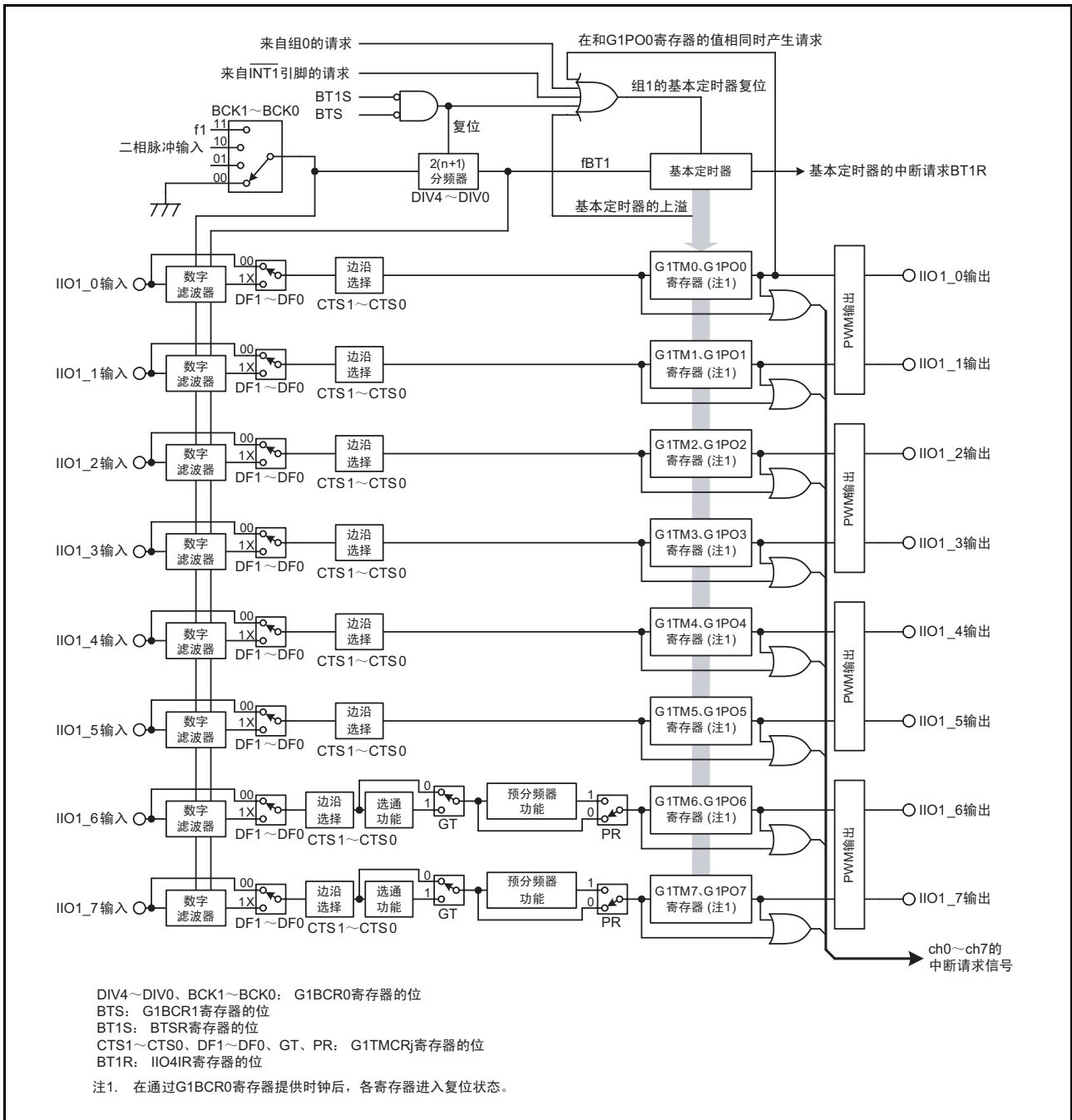


图 23.2 智能 I/O 组 1 的框图 (j=0 ~ 7)

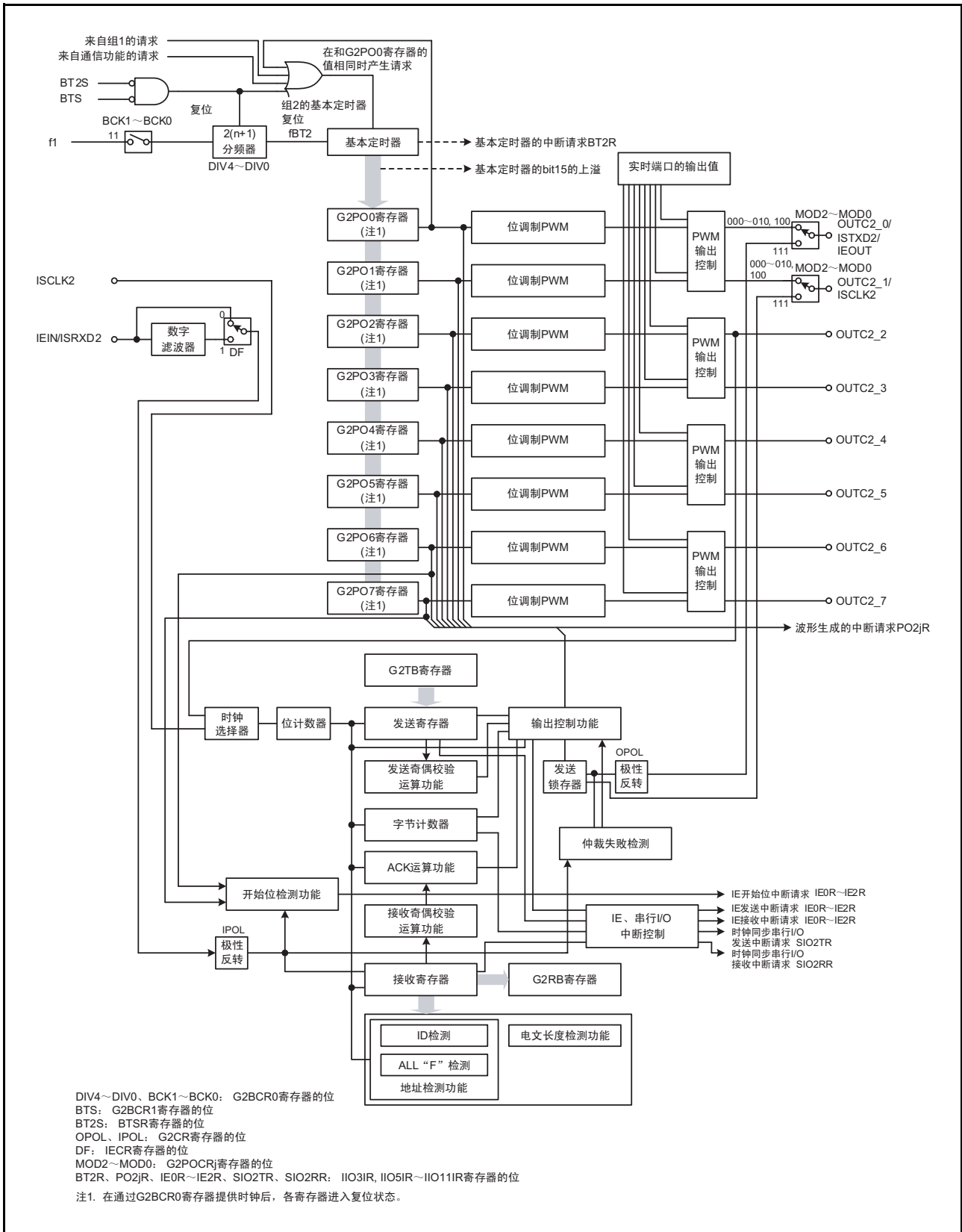


图 23.3 智能 I/O 组 2 的框图 (j=0 ~ 7)

智能 I/O 的基本定时器、时间测量功能、波形生成功能的相关寄存器如图 23.4 ~ 图 23.17 所示（通信功能的相关寄存器请参照图 23.33 ~ 图 23.40）。

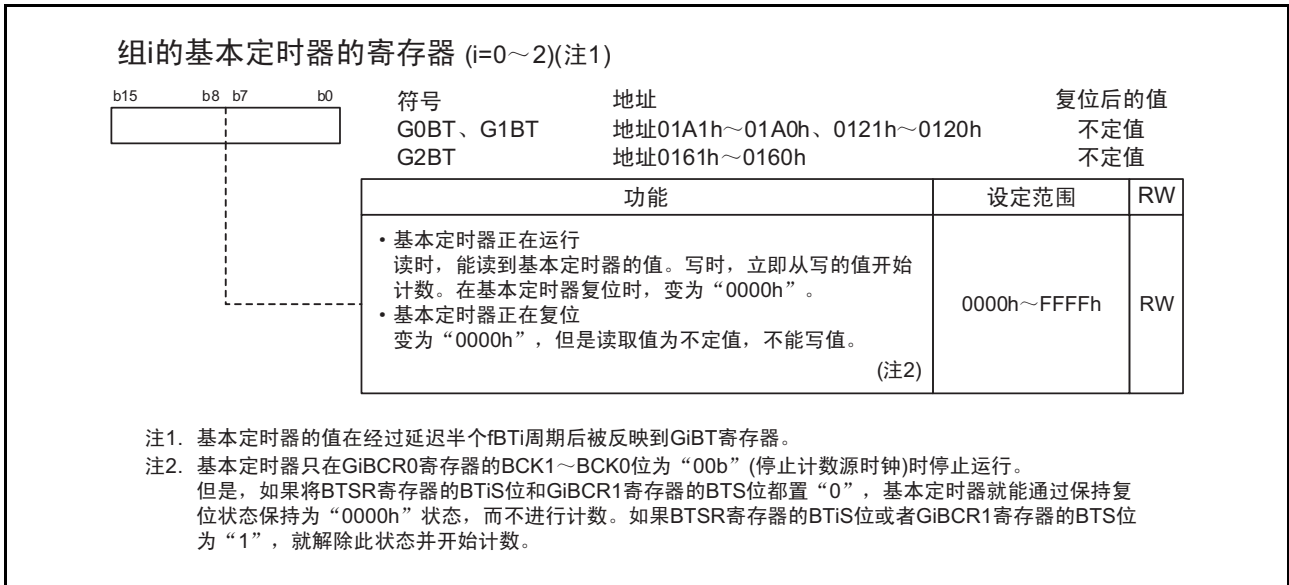


图 23.4 G0BT ~ G2BT 寄存器

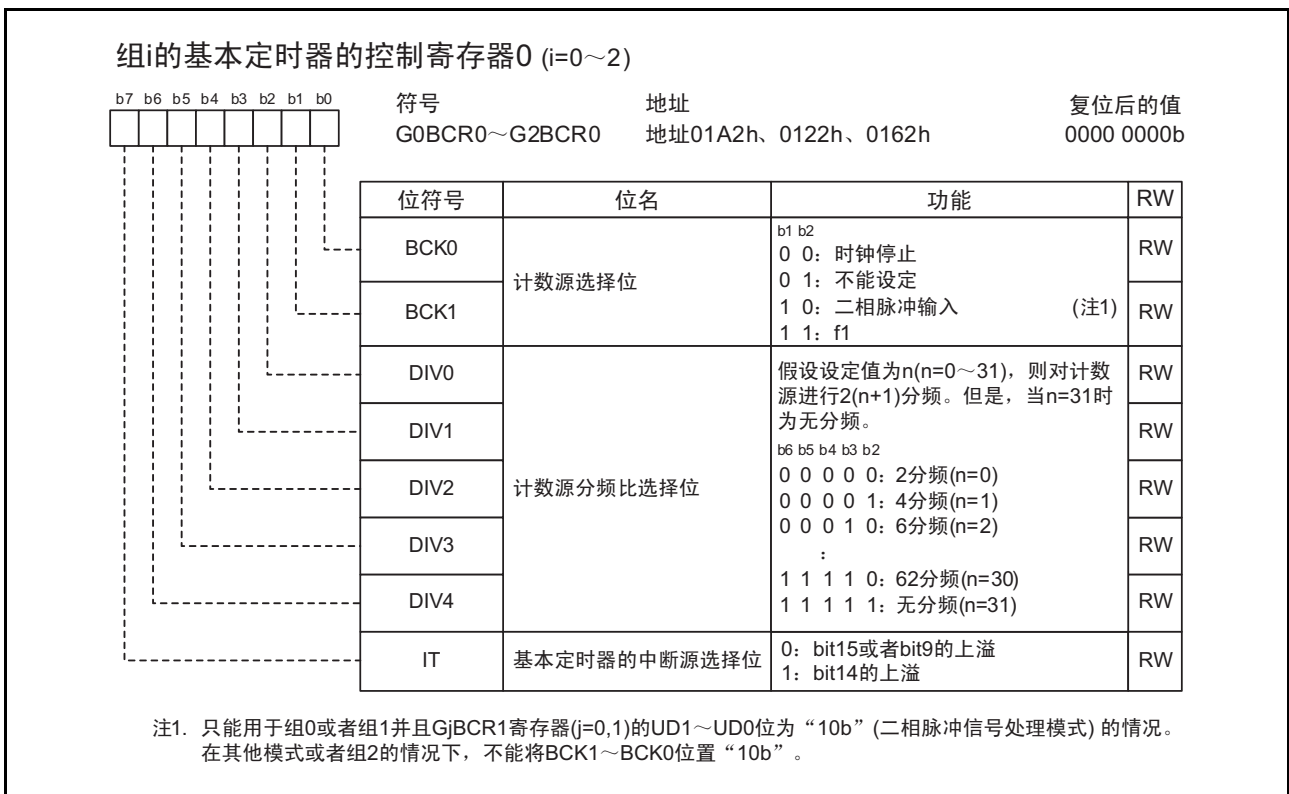


图 23.5 G0BCR0 ~ G2BCR0 寄存器



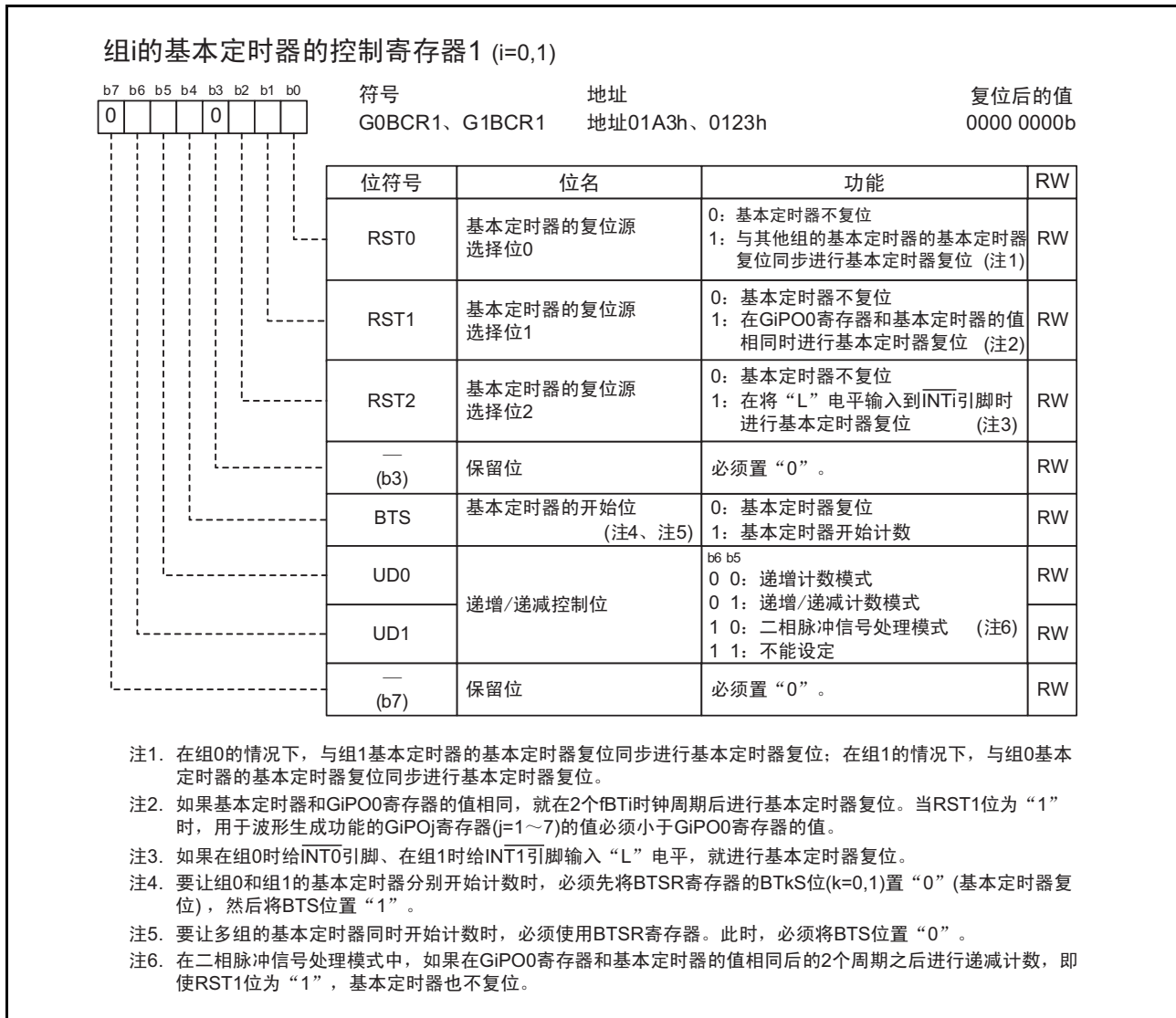


图 23.6 G0BCR1 寄存器和 G1BCR1 寄存器

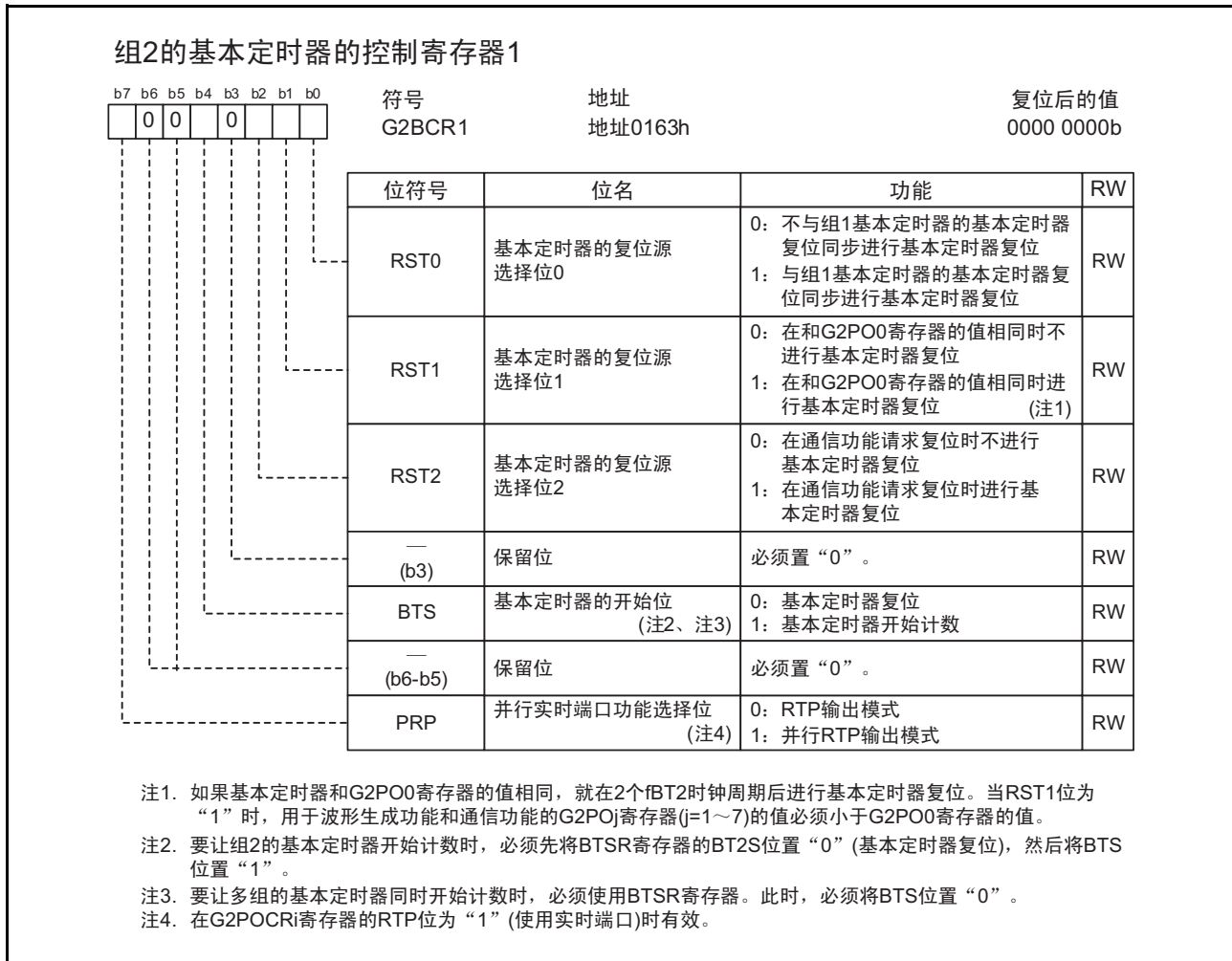


图 23.7 G2BCR1 寄存器

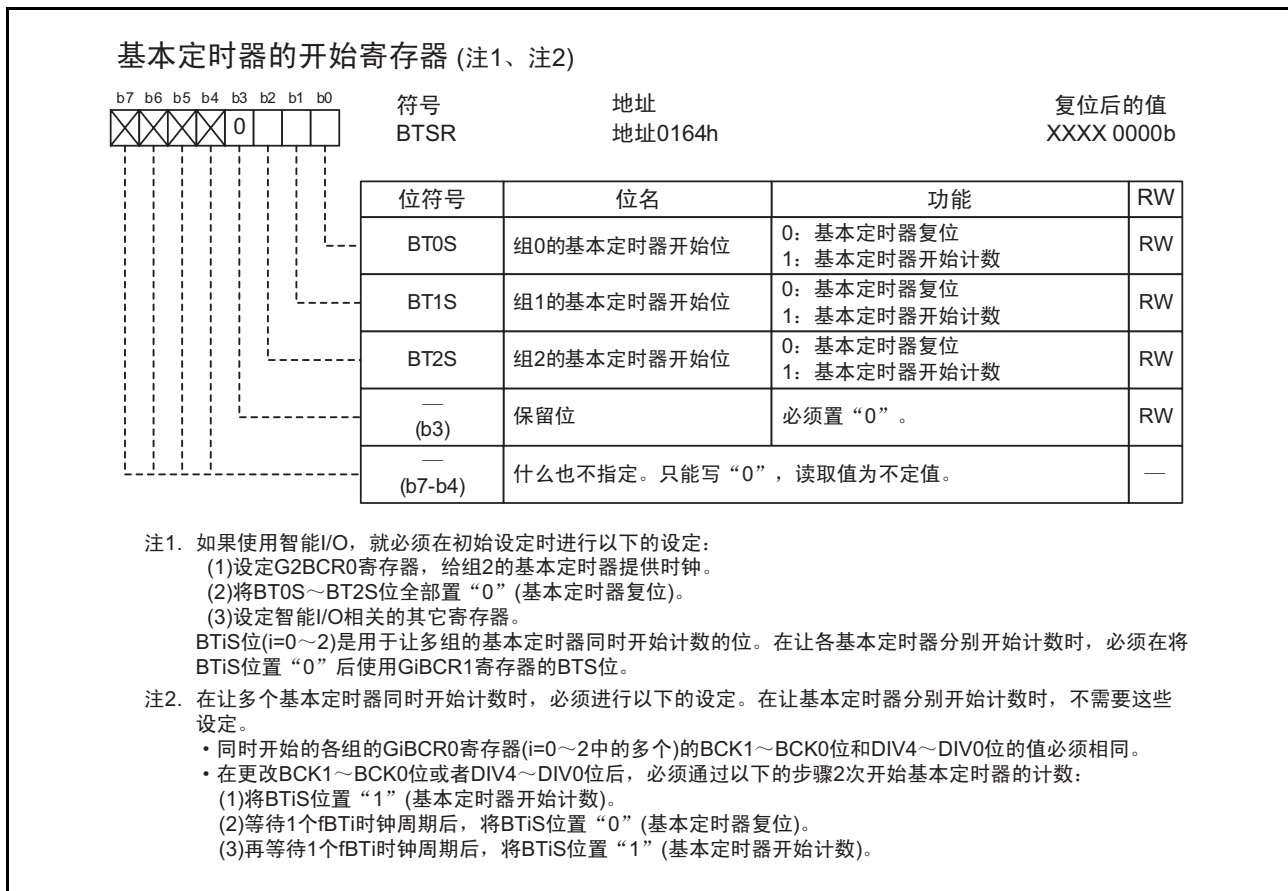


图 23.8 BTSR 寄存器

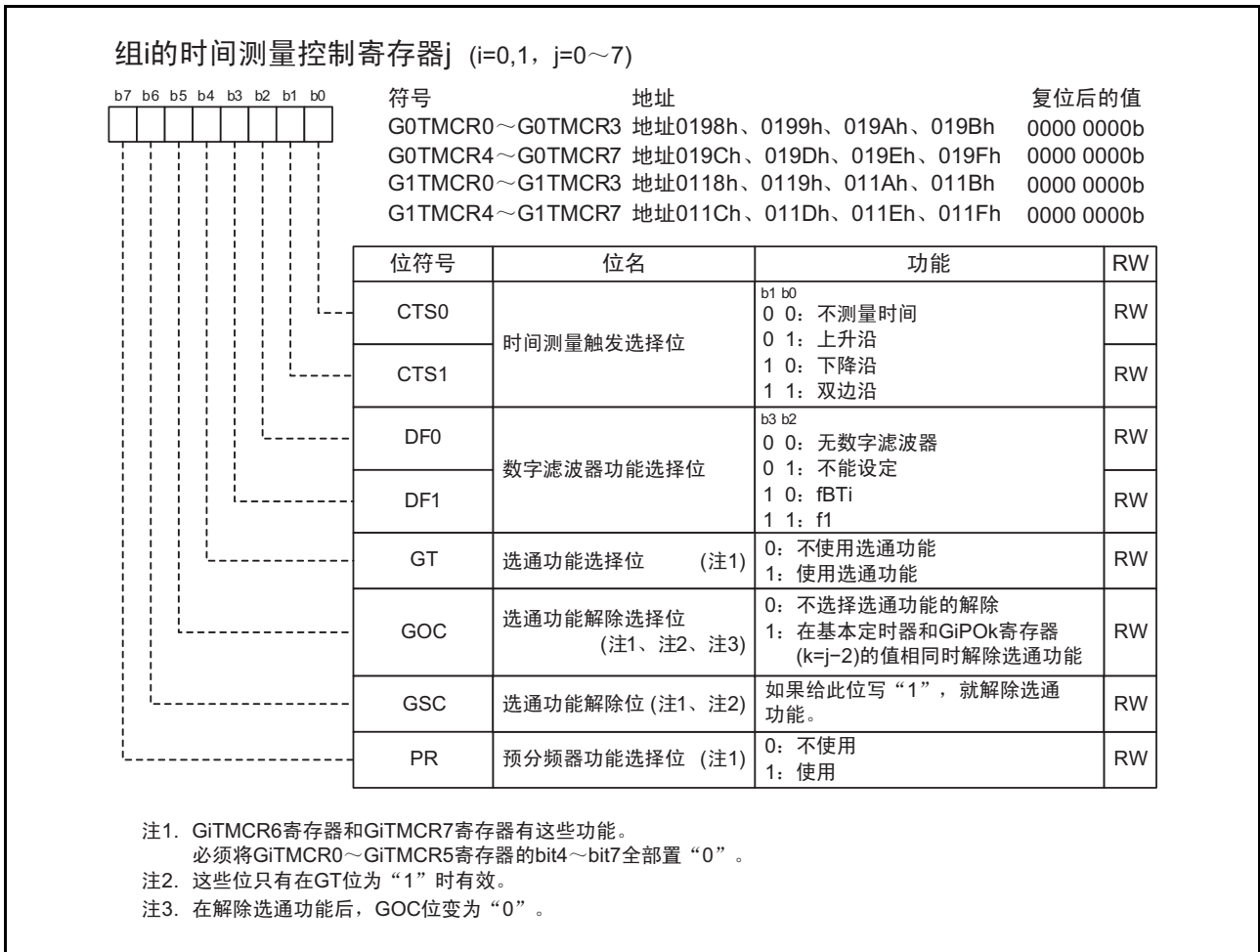


图 23.9 G0TMCR0 ~ G0TMCR7 寄存器和 G1TMCR0 ~ G1TMCR7 寄存器

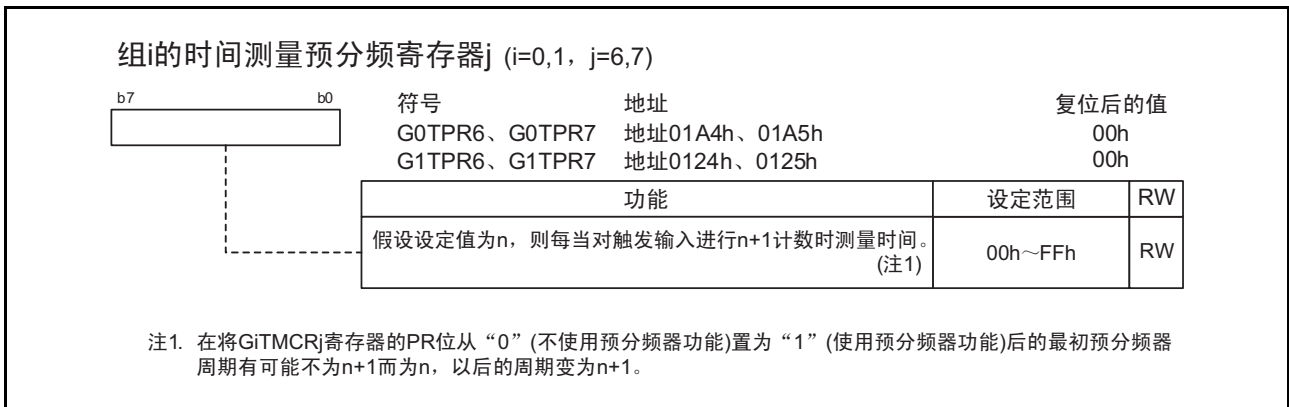


图 23.10 G0TPR6、G0TPR7、G1TPR6、G1TPR7 寄存器

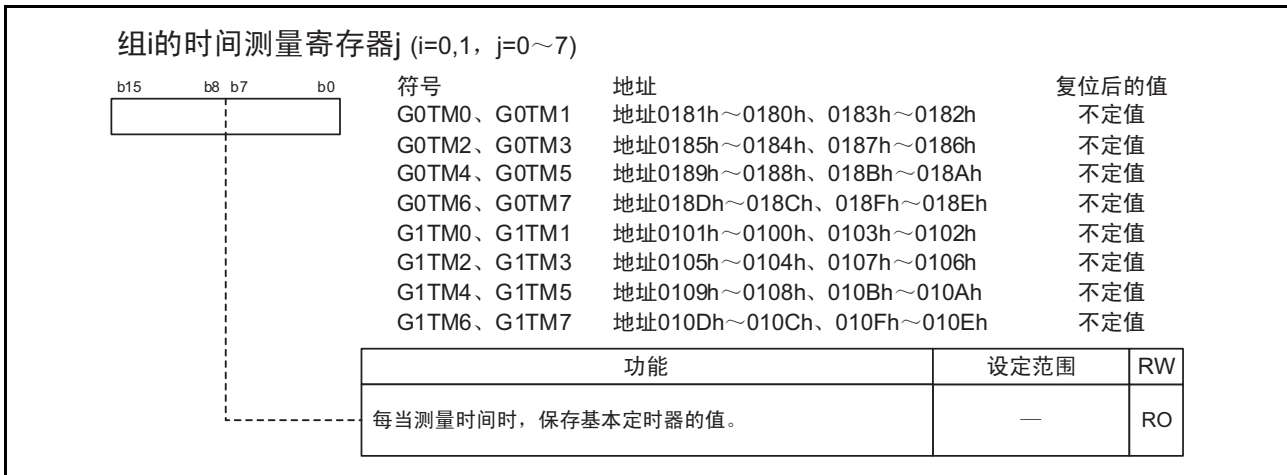


图 23.11 G0TM0 ~ G0TM7 寄存器和 G1TM0 ~ G1TM7 寄存器

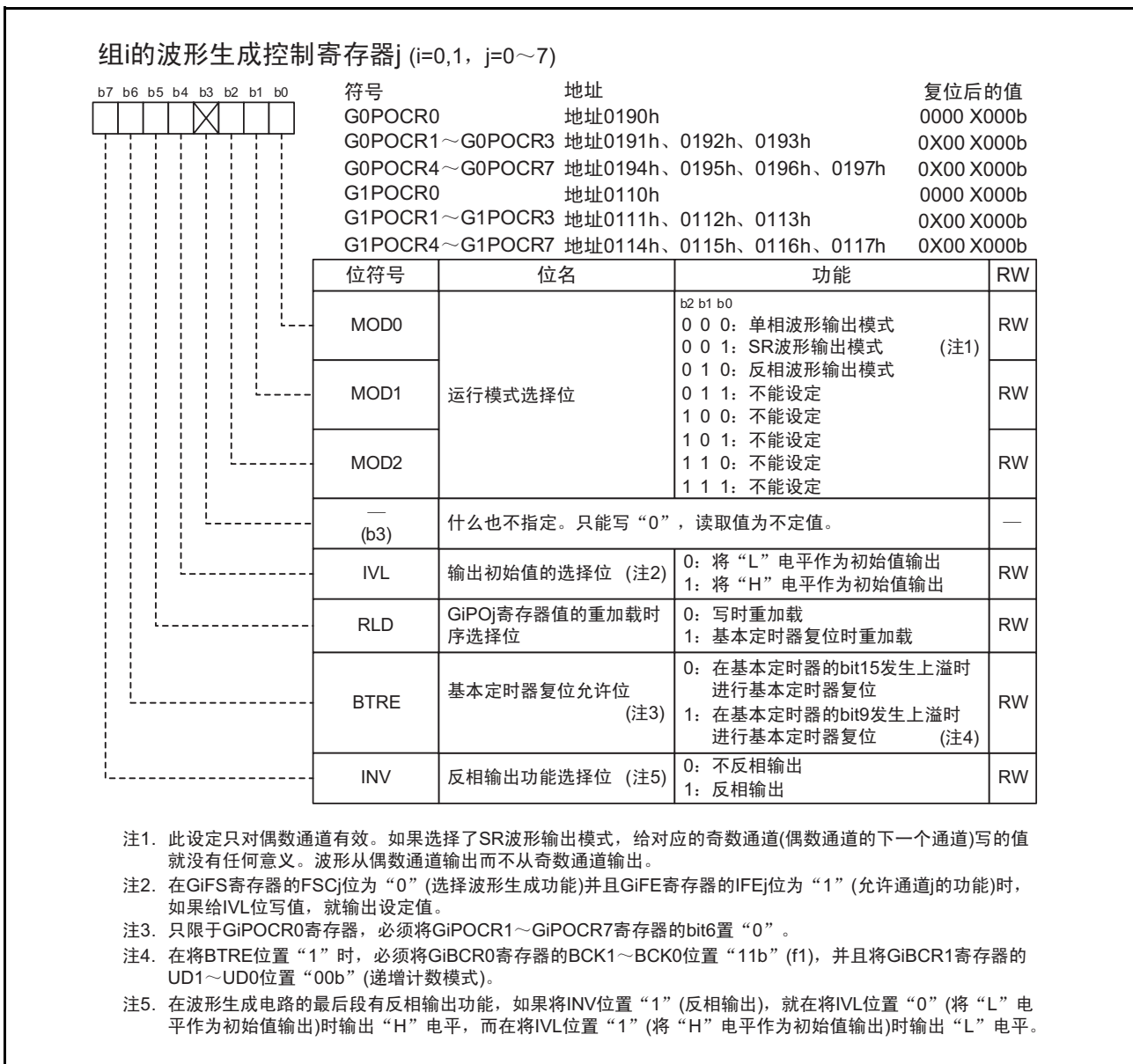


图 23.12 G0POCR0 ~ G0POCR7 寄存器和 G1POCR0 ~ G1POCR7 寄存器

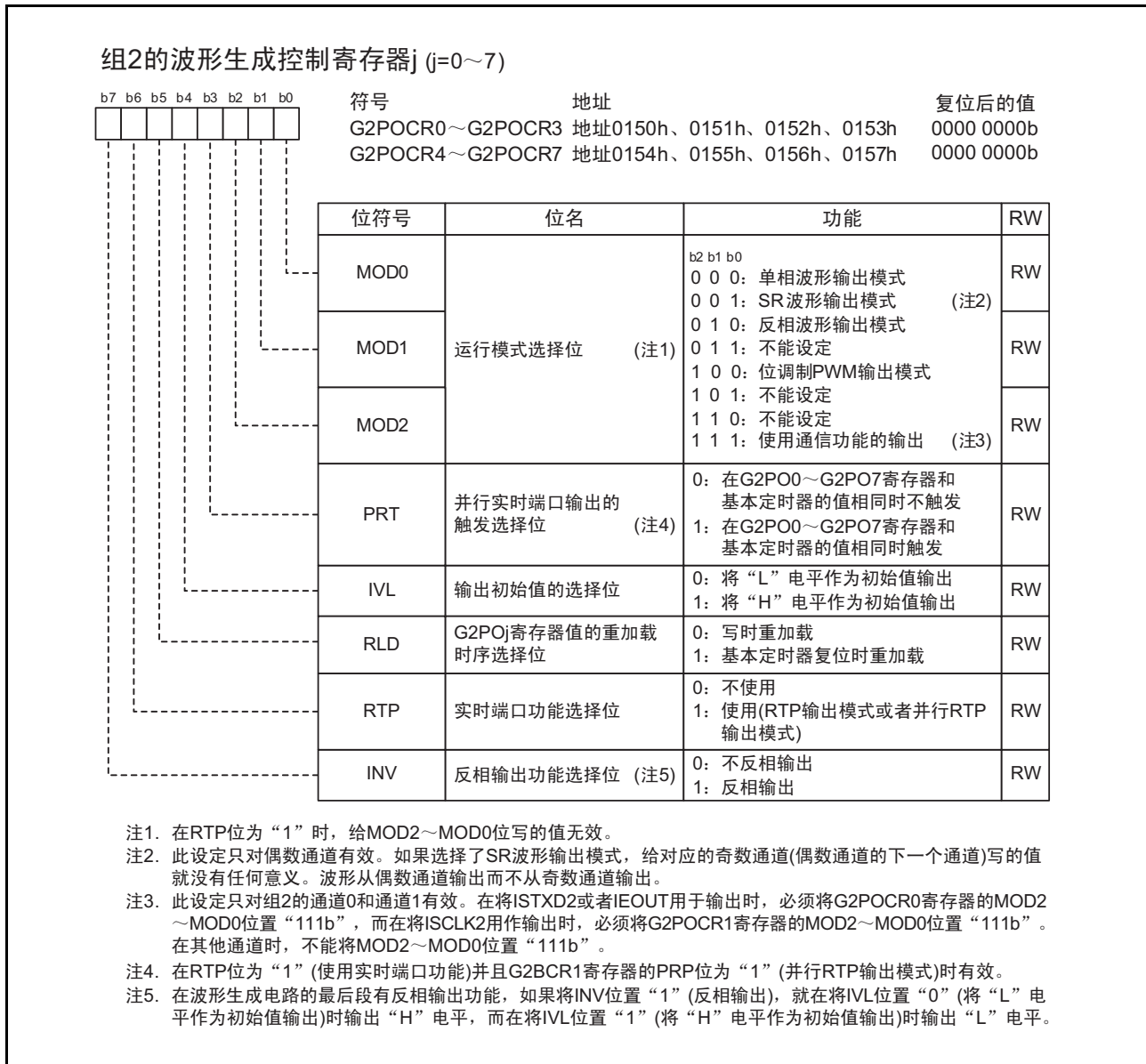


图 23.13 G2POCR0 ~ G2POCR7 寄存器



图 23.14 G0PO0 ~ G0PO7、G1PO0 ~ G1PO7、G2PO0 ~ G2PO7 寄存器

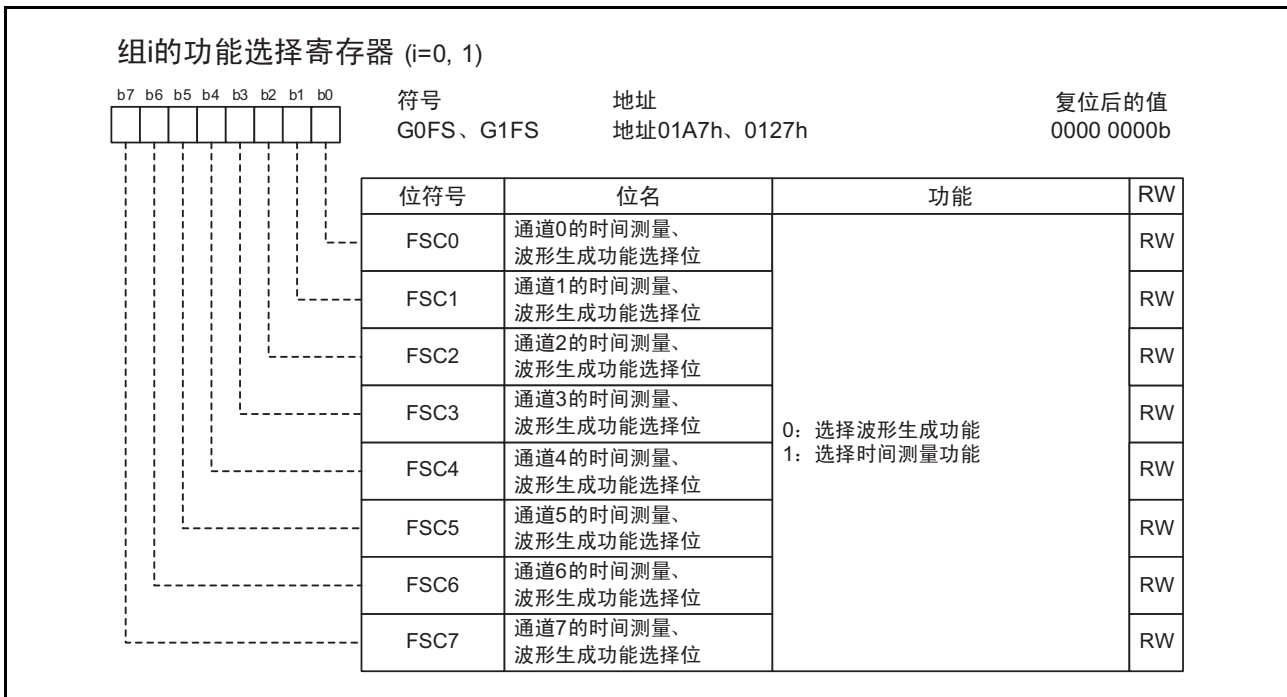


图 23.15 G0FS 寄存器和 G1FS 寄存器

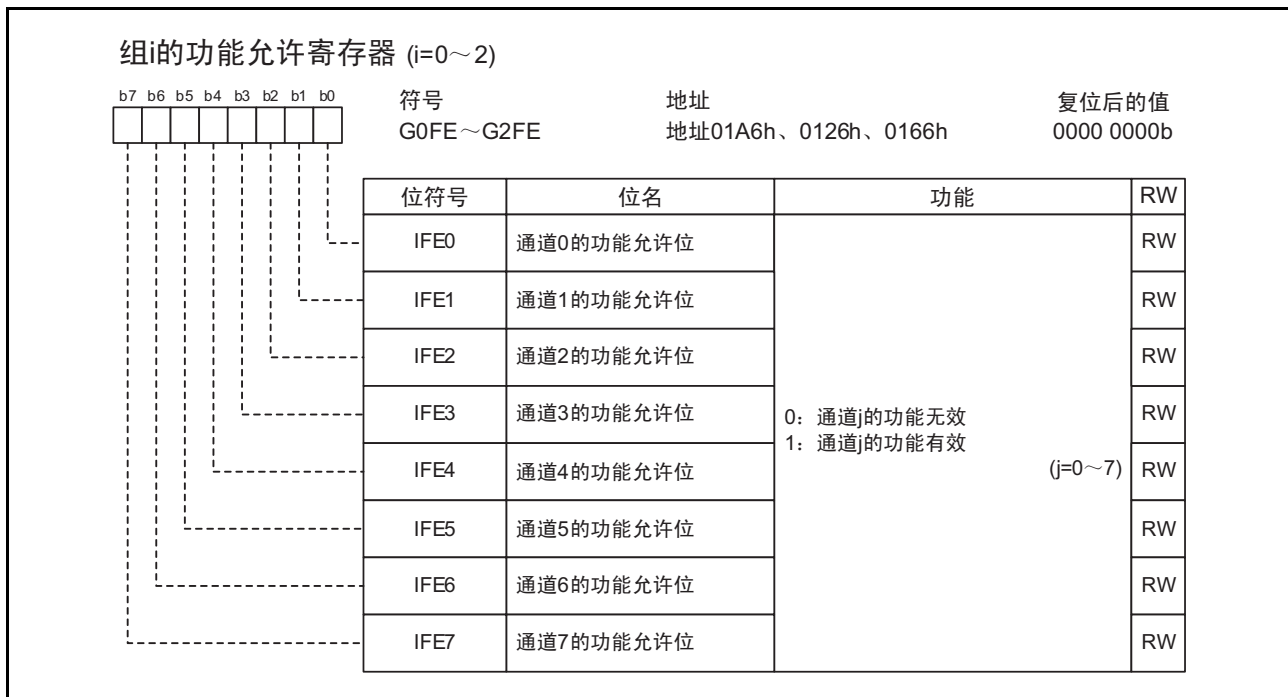


图 23.16 G0FE ~ G2FE 寄存器

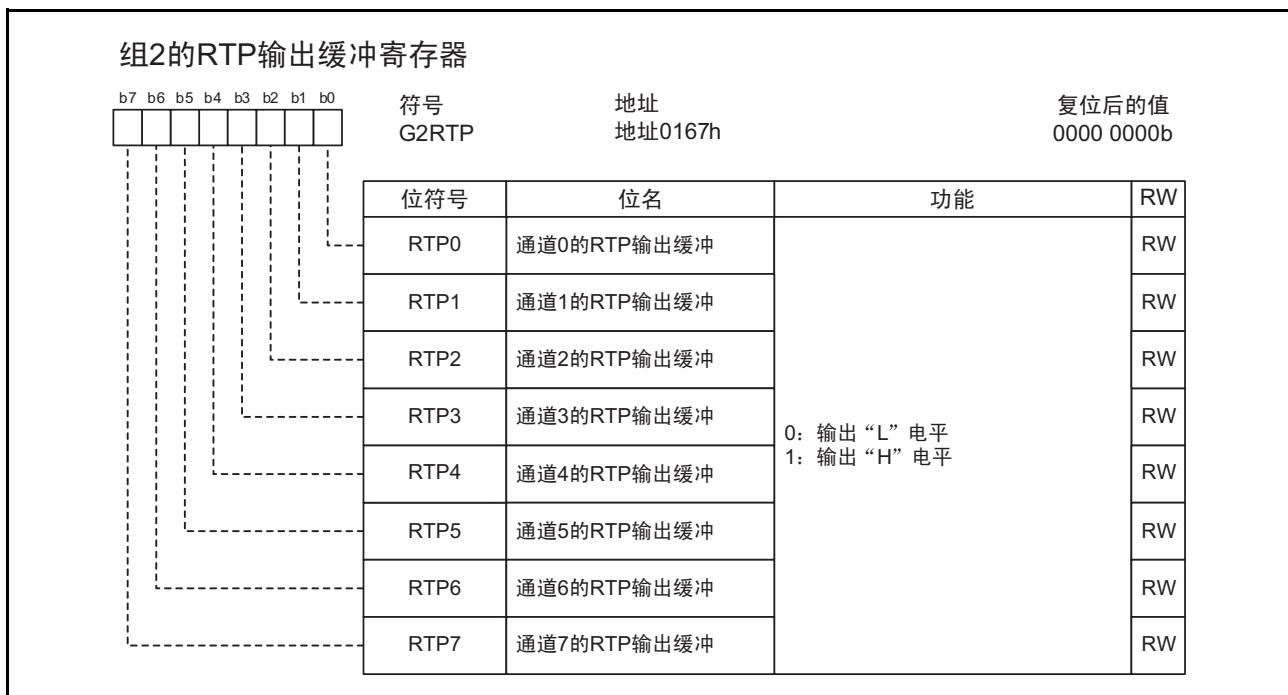


图 23.17 G2RTP 寄存器



### 23.1 基本定时器（组 0～组 2）

对内部生成的计数源进行自由计数。基本定时器的规格如表 23.2 所示，基本定时器的相关寄存器和框图如图 23.4～图 23.17、基本定时器的框图如图 23.18 所示，基本定时器（组 0 和组 1）的递增计数模式的运行例子、基本定时器的递增 / 递减计数模式的运行例子和二相脉冲信号处理模式的运行例子分别如图 23.19、图 23.20 和图 23.21 所示。

表 23.2 基本定时器的规格（i=0～2）

项目	规格
计数源（fBTi）	f1 的 2(n+1) 分频（组 0～组 2）、二相脉冲输入的 2(n+1) 分频（组 0 和组 1） n: 通过 GiBCR0 寄存器的 DIV4～DIV0 位进行设定。 n=0～31, 但是当 n=31 时, 不进行分频。
计数	<ul style="list-style-type: none"> <li>• 递增计数</li> <li>• 递增 / 递减计数</li> <li>• 二相脉冲处理</li> </ul>
计数开始条件	<ul style="list-style-type: none"> <li>• 使各组基本定时器分别开始计数的情况: 将 GiBCR1 寄存器的 BTS 位置“1”（基本定时器开始计数）。</li> <li>• 使多组基本定时器同时开始计数的情况: 将 BTRSR 寄存器的 BTIS 位置“1”（基本定时器开始计数）。</li> </ul>
计数停止条件	将 BTRSR 寄存器的 BTiS 位置“0”（基本定时器复位）并且将 GiBCR1 寄存器的 BTS 位置“0”（基本定时器复位）。
基本定时器复位的条件	<ul style="list-style-type: none"> <li>• 基本定时器和 GiPO0 寄存器的值相同。</li> <li>• 给外部中断引脚输入“L”电平。 组 0: <math>\overline{\text{INT0}}</math> 引脚 组 1: <math>\overline{\text{INT1}}</math> 引脚</li> <li>• 基本定时器的 bit15 或者 bit9 发生上溢。</li> <li>• 来自通信功能的初始化请求（组 2）。</li> </ul>
基本定时器复位时的值	“0000h”
中断请求	当基本定时器的 bit9、bit14 或者 bit15 发生上溢时, 中断请求寄存器的 BTiR 位变为“1”（有中断请求）（参照图 11.12）。
读基本定时器	<ul style="list-style-type: none"> <li>• 如果在基本定时器运行时读 GiBT 寄存器, 就能读到基本定时器的值。</li> <li>• 如果在基本定时器复位时读 GiBT 寄存器, 读取值就为不定值。</li> </ul>
写基本定时器	如果在基本定时器运行时写值, 就立即从写的值开始计数。不能在基本定时器复位时写值。
选择功能	<ul style="list-style-type: none"> <li>• 递增 / 递减计数模式（组 0 和组 1） 如果将 BTS 位或者 BTIS 位置“1”, 基本定时器就开始计数, 当计数到“FFFFh”时, 就进行递减计数。另外, 当 GiBCR1 寄存器的 RST1 位为“1”（在基本定时器和 GiPO0 寄存器的值相同时进行基本定时器复位）时, 在基本定时器和 GiPO0 寄存器的值相同时的 2 次计数后, 进行递减计数。当计数到“0000h”时, 再次进行递增计数（参照图 23.20）。</li> <li>• 二相脉冲处理模式（组 0 和组 1） 对 UDiA 和 UDiB 的二相脉冲进行计数（参照图 23.21）。</li> </ul> <div style="text-align: center;"> <p>UDiA</p> <p>UDiB</p> <p>对全部边沿进行递增计数。 对全部边沿进行递减计数。</p> </div>

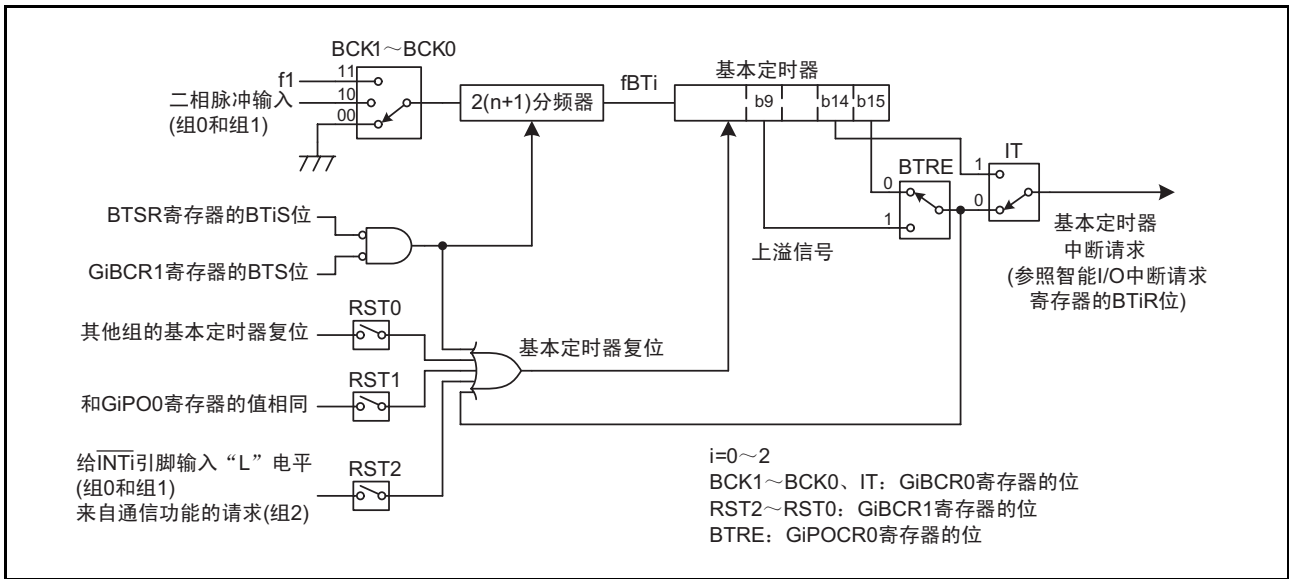


图 23.18 基本定时器的框图

表 23.3 基本定时器的相关寄存器的设定（时间测量功能、波形生成功能和通信功能通用）（i=0 ~ 2）

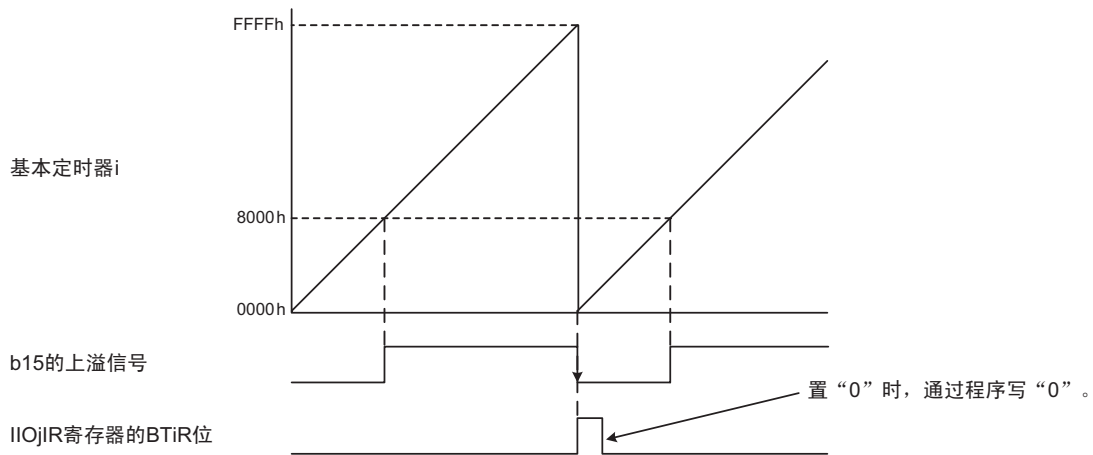
寄存器	位	功能
G2BCR0	—	给 BCSR 寄存器提供运行时钟，必须置 “0111 1111b”。
BCSR	—	必须置 “0000 0000b”。
GiBCR0	BCK1 ~ BCK0	选择计数源。
	DIV4 ~ DIV0	选择计数源的分频比。
	IT	选择基本定时器中断。
GiBCR1	RST2 ~ RST0	选择基本定时器复位的时序。
	BTS	用于基本定时器分别开始计数的情况。
	UD1 ~ UD0	选择计数方法（组 0 和组 1）。
GiPOCR0	BTRE	选择基本定时器的复位源。
GiBT	—	读写基本定时器的值。

当 RST1 位为 “1”（在基本定时器和 GiPO0 的值相同时进行基本定时器复位）时，需要对以下寄存器进行设定：

GiPOCR0	MOD2 ~ MOD0	必须置 “000b”（单相波形输出模式）。
GiPO0	—	必须设定复位周期。
GiFS	FSC0	必须置 “0”（波形生成功能）。
GiFE	IFE0	必须置 “1”（通道开始工作）。

位的构成和功能因组而不同。

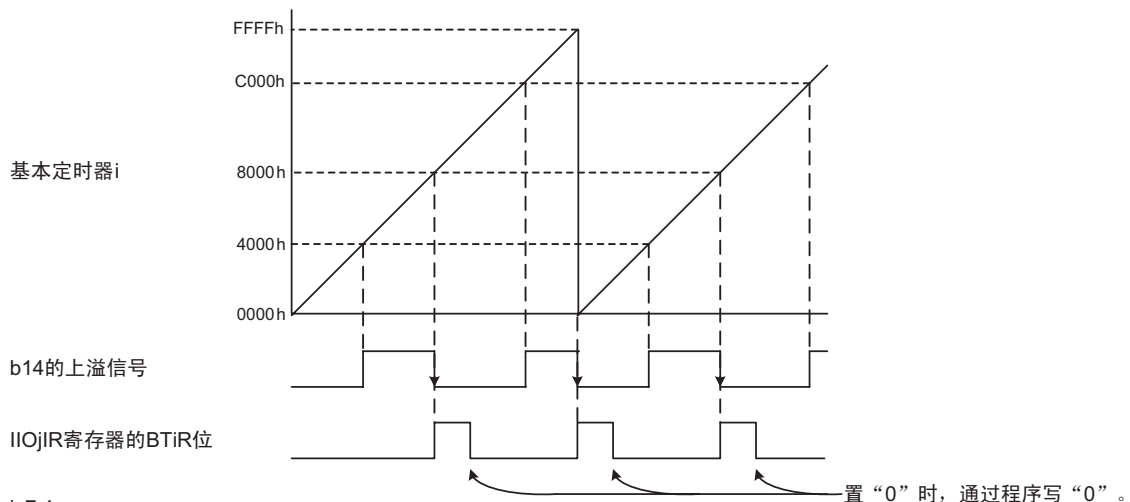
(1) GiBCR0寄存器(i=0~1)的IT位为“0”(在bit15上溢时发生基本定时器中断)的情况



此图的条件如下:

- GiBCR1寄存器的RST1位为“0”(在GiPO0寄存器和基本定时器的值相同时不进行基本定时器复位)。
- GiBCR1寄存器的UD1~UD0位为“00b”(递增计数模式)。

(2) GiBCR0寄存器(i=0~1)的IT位为“1”(在bit14上溢时发生基本定时器中断)的情况



此图的条件如下:

- GiBCR1寄存器的RST1位为“0”(在GiPO0寄存器和基本定时器的值相同时不进行基本定时器复位)。
- GiBCR1寄存器的UD1~UD0位为“00b”(递增计数模式)。

图 23.19 基本定时器的递增模式的运行例子 (组 0 和组 1)

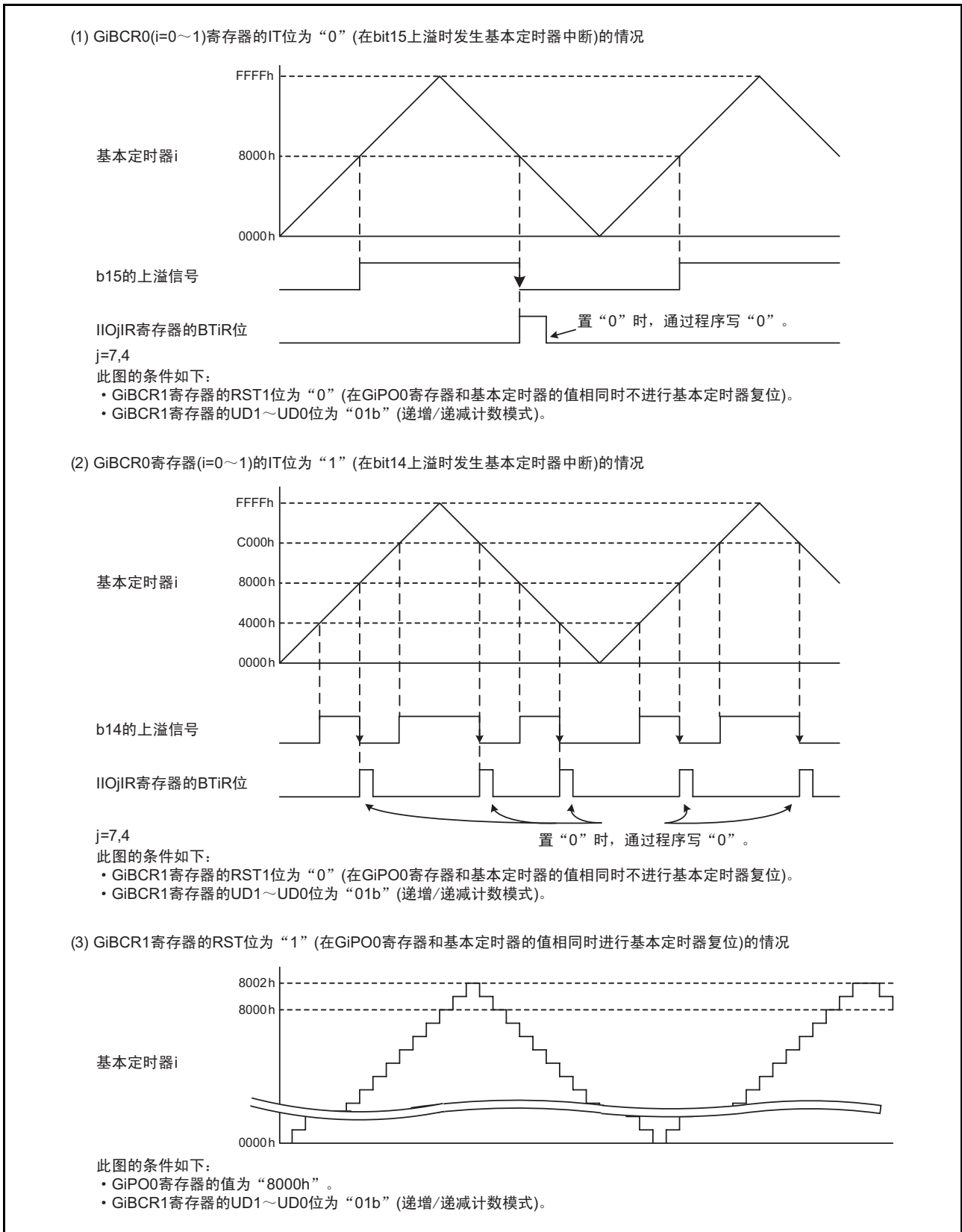
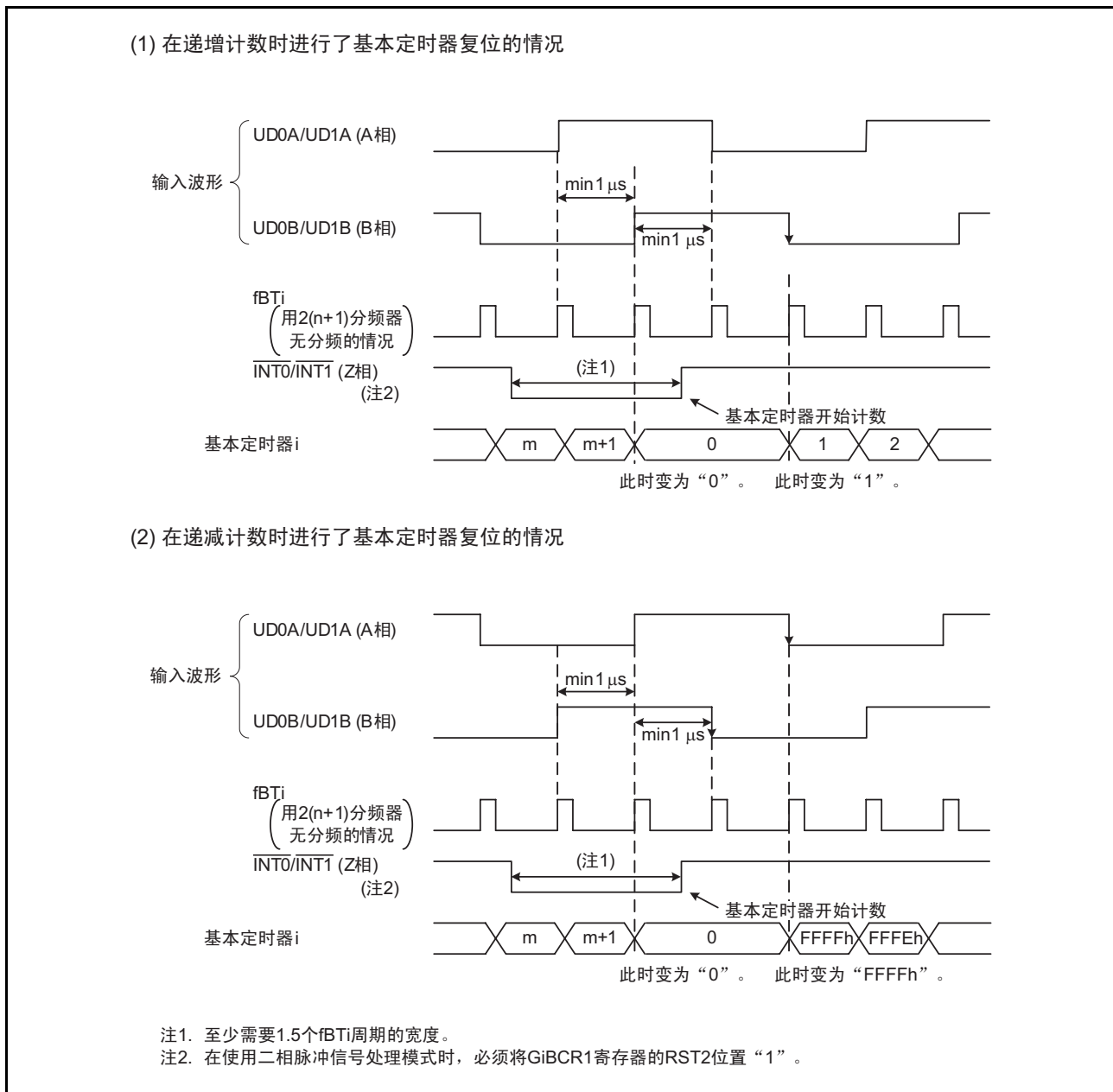


图 23.20 基本定时器的递增 / 递减模式的运行例子 (组 0 和组 1)

图 23.21 基本定时器的二相脉冲信号处理模式的运行例子 ( $i=0,1$ ) (组 0 和组 1)

## 23.2 时间测量功能（组 0 和组 1）

在每次输入外部触发时，将基本定时器的值保存到 GiTMj 寄存器（ $i=0,1$ ， $j=0 \sim 7$ ）。时间测量功能的规格及其相关寄存器的设定分别如表 23.4 和表 23.5 所示，时间测量功能的运行例子以及使用预分频器功能和选通功能时的运行例子分别图 23.22 ~ 图 23.23 和图 23.24 所示。

表 23.4 时间测量功能的规格（ $i=0,1$ ， $j=0 \sim 7$ ）

项目	规格
测量通道	组 0: 通道 0 ~ 7 组 1: 通道 0 ~ 7
触发输入极性的选择	能选择 IIOi_j 引脚的上升沿、下降沿或者双边沿。
测量开始条件	在 GiFS 寄存器的 FSCj 位为“1”（选择时间测量功能）的状态下，将 GiFE 寄存器的 IFEj 位置“1”（通道 j 的功能有效）。
测量停止条件	将 IFEj 位置“0”（通道 j 的功能无效）。
时间测量时序	<ul style="list-style-type: none"> <li>无预分频器功能：在每次输入触发时。</li> <li>有预分频器功能（通道 6 和通道 7）：在每（GiTPRk 寄存器（<math>k=6,7</math>）的值 +1）次输入触发时。</li> </ul>
中断请求	在进行时间测量时，中断请求寄存器的 TMijR 位变为“1”（有中断请求）（参照图 11.12）。
IIOi_j 引脚（输入）	触发输入
选择功能	<ul style="list-style-type: none"> <li>数字滤波器功能 在每个 f1 或者 fBTi 时判断触发输入的电平，让 3 次相同的脉冲成分通过。</li> <li>预分频器功能（通道 6 和通道 7） 对触发输入进行计数，在每（GiTPRk 寄存器值 +1）次输入触发时进行时间测量。</li> <li>选通功能（通道 6 和通道 7） 在通过第一次触发输入进行时间测量后，禁止接受触发输入。在 GiTMCRk 寄存器的 GOC 位为“1”（在和 GiPOp 寄存器（<math>p=4 \sim 5</math>。当 <math>k=6</math> 时 <math>p=4</math>；当 <math>k=7</math> 时 <math>p=5</math>）的值相同时解除选通）的状态下，如果基本定时器和 GiPOp 寄存器的值相同，或者将 GiTMCRk 寄存器的 GSC 位置“1”，就允许再次接受触发输入。</li> </ul>

表 23.5 时间测量功能（组 0 和组 1）的相关寄存器的设定（ $i=0,1$ ， $j=0 \sim 7$ ， $k=6,7$ ）

寄存器	位	功能
GiTMCRj	CTS1 ~ CTS0	选择时间测量触发。
	DF1 ~ DF0	选择数字滤波器功能。
	GT、GOC、GSC	选择选通功能。
	PR	选择预分频器功能。
GiTPRk	—	设定预分频器的值。
GiFS	FSCj	必须置“1”（时间测量功能）。
GiFE	IFEj	必须置“1”（通道 j 的功能有效）。

位的构成和功能因组和通道而不同。

必须在设定基本定时器的相关寄存器后，设定时间测量功能的相关寄存器。

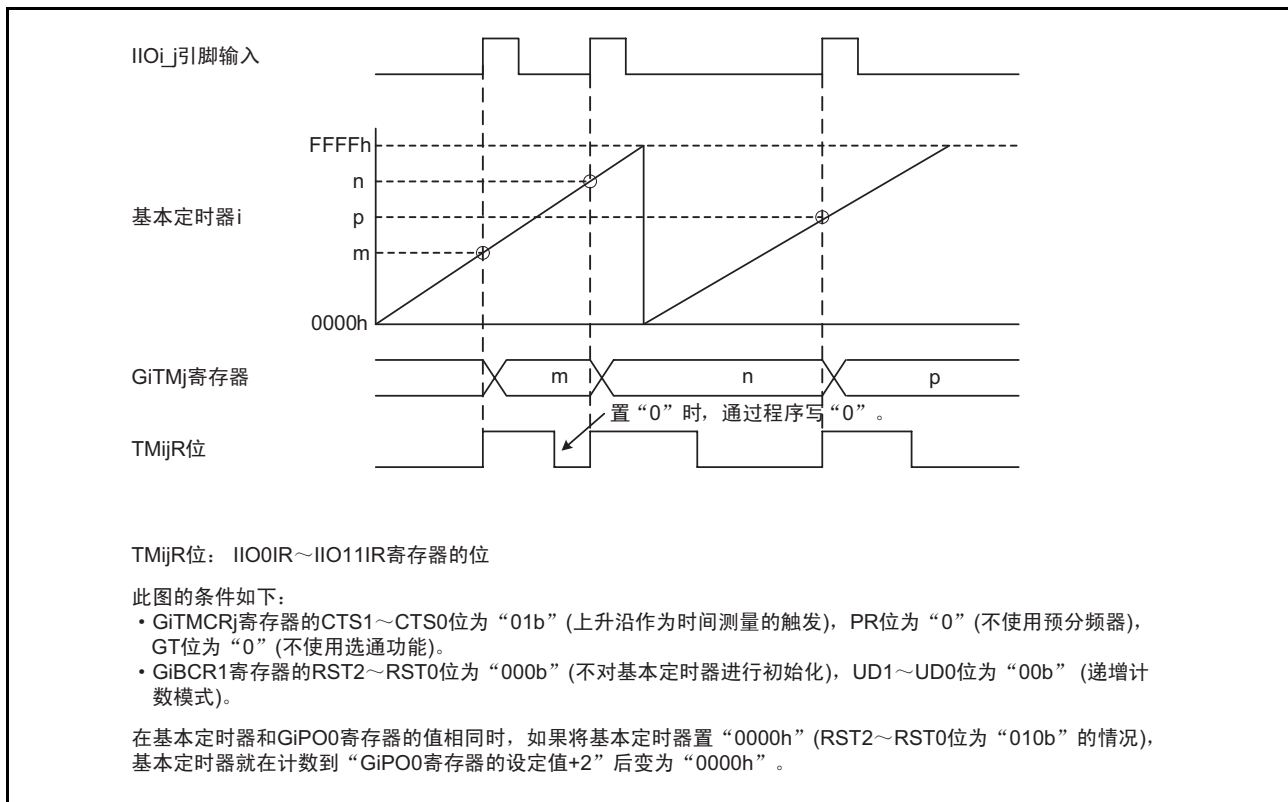


图 23.22 时间测量功能的运行例子 (1) (i=0,1, j=0~7)

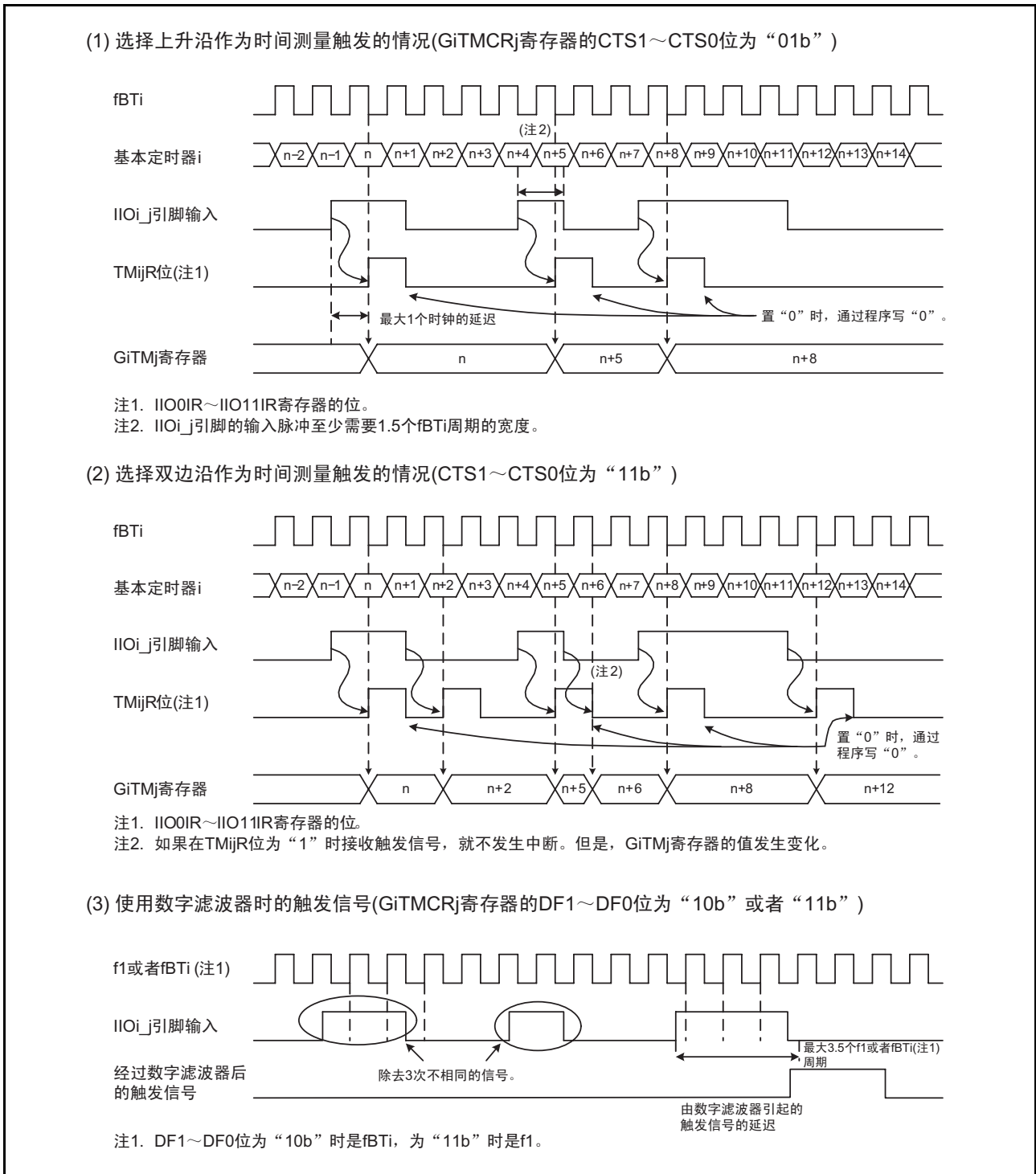
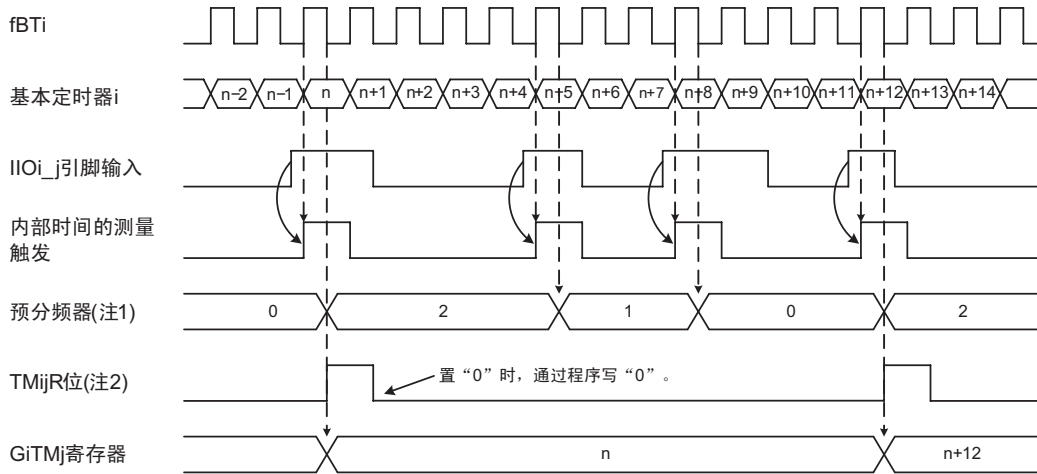


图 23.23 时间测量功能的运行例子 (2) (i=0,1, j=0 ~ 7)

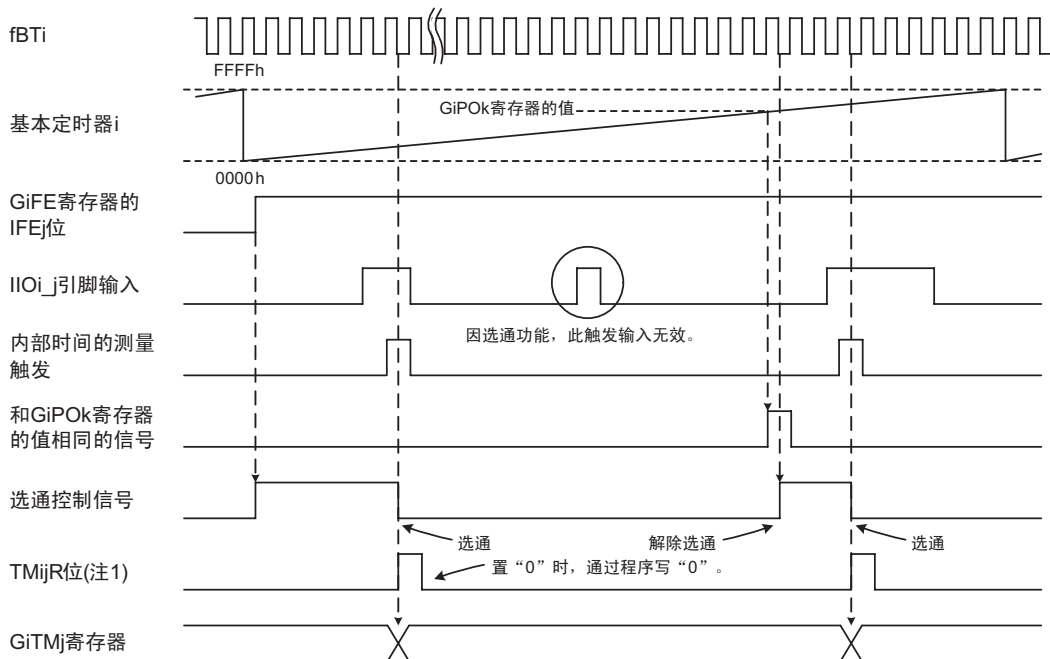


(1) 使用预分频器功能的情况  
(GiTPRj寄存器为“02h”，GiTMCRj寄存器的PR位为“1”)



注1. 这是将GiTMCRj寄存器的PR位置“1” (使用预分频器功能)后的第2次以后的预分频器周期的情况。  
注2. 这是IIO00IR~IIO111R寄存器的位。

(2) 使用选通功能的情况  
(在和GiPOk寄存器的值相同时解除选通功能，GiTMCRj寄存器的GT位为“1”，GOC位为“1”)



k=4,5  
注1. 这是IIO00IR~IIO111R寄存器的位。

图 23.24 使用预分频器功能和选通功能时的运行例子 (i=0,1, j=6,7)

### 23.3 波形生成功能（组 0～组 2）

在基本定时器和 GiPOj 寄存器（i=0～2，j=0～7）的值相同时，生成波形。

波形生成功能有以下 6 种模式：

- 单相波形输出模式（组 0～组 2）
- 反相波形输出模式（组 0～组 2）
- 置位-复位波形输出（SR 波形输出）模式（组 0～组 2）
- 位调制 PWM 输出模式（组 2）
- 实时端口（RTP）输出模式（组 2）
- 并行实时端口（RTP）输出模式（组 2）

波形生成功能的相关寄存器的设定如表 23.6 所示。

表 23.6 波形生成功能的相关寄存器的设定（i=0～2，j=0～7）

寄存器	位	功能
GiPOCRj	MOD2～MOD0	选择波形输出模式。
	PRT（注 1）	在使用并行 RTP 输出模式时，必须置“1”。
	IVL	选择输出初始值。
	RLD	选择 GiPOj 寄存器值的重加载时序。
	RTP（注 1）	在使用 RTP 输出模式或者并行 RTP 输出模式时，必须置“1”。当此位为“1”时，MOD2～MOD0 位的值无效。
	INV	选择反相输出。
G2BCR1	PRP	在使用并行 RTP 输出模式时，必须置“1”。
GiPOj	—	设定反相输出波形的时序。
GiFS	FSCj	必须置“0”（波形生成功能）（只限于组 0 和组 1）。
GiFE	IFEj	必须置“1”（通道 j 的功能有效）。
G2RTP	RTP0～RTP7	设定 RTP 输出模式、并行 RTP 输出模式中的 RTP 输出值。

位的构成和功能因组和通道而不同。

必须在设定基本定时器的相关寄存器后，设定波形生成功能的相关寄存器。

注 1. G2POCRj 寄存器有此位，而 G0POCRj 寄存器和 G1POCRj 寄存器没有此位。

### 23.3.1 单相波形输出模式（组 0～组 2）

如果基本定时器和 GiPOj 寄存器（i=0～2，j=0～7）的值相同，IIOi<sub>j</sub> 引脚（组 2 为 OUTC2<sub>j</sub> 引脚）的输出电平就变为“H”。如果基本定时器变为“0000h”，输出电平就变为“L”。如果将 GiPOCRj 寄存器的 IVL 位置“1”（将“H”电平作为初始值输出），开始输出波形时的输出电平就变为“H”。如果将 INV 位置“1”（反相输出），就将输出波形的电平反相后输出。详细内容请参照“图 23.25 单相波形输出模式的运行例子（i=0～2）”。

单相波形输出模式的规格如表 23.7 所示。

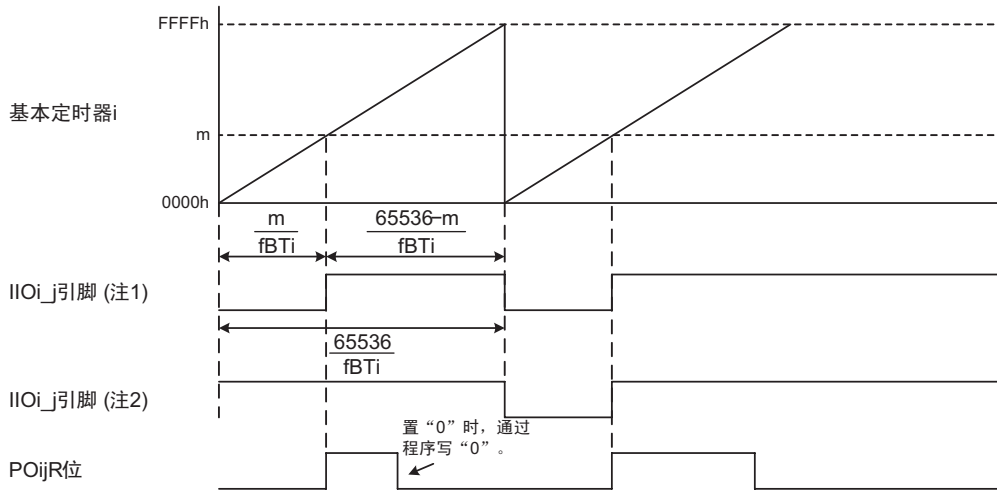
表 23.7 单相波形输出模式的规格（i=0～2）

项目	规格
输出波形（注 1）	<ul style="list-style-type: none"> <li>自由运行（GiBCR1 寄存器的 RST2～RST0 位为“000b”）的情况：           <ul style="list-style-type: none"> <li>周期：<math>\frac{65536}{fBTi}</math></li> <li>“L”电平宽度：<math>\frac{m}{fBTi}</math></li> <li>“H”电平宽度：<math>\frac{65536 - m}{fBTi}</math></li> </ul> </li> <li>m: GiPOj 寄存器（j=0～7）的设定值（0000h～FFFFh）</li> <li>在基本定时器和 GiPO0 寄存器的值相同时，将基本定时器置“0000h”（GiBCR1 寄存器的 RST2～RST0 位为“010b”）的情况：           <ul style="list-style-type: none"> <li>周期：<math>\frac{n + 2}{fBTi}</math></li> <li>“L”电平宽度：<math>\frac{m}{fBTi}</math></li> <li>“H”电平宽度：<math>\frac{n + 2 - m}{fBTi}</math></li> </ul> </li> <li>m: GiPOj 寄存器（j=1～7）的设定值（0000h～FFFFh）</li> <li>n: GiPO0 寄存器的设定值（0001h～FFFDh）</li> <li>当 <math>m \geq n + 2</math> 时，输出电平固定为“L”。</li> </ul>
波形输出开始条件（注 2）	将 GiFE 寄存器的 IFEj 位（j=0～7）置“1”（通道 j 的功能有效）。
波形输出停止条件	将 IFEj 位置“0”（通道 j 的功能无效）。
中断请求	如果基本定时器的值和 GiPOj 寄存器的值相同，智能 I/O 的中断请求寄存器的 POijR 位就变为“1”（有中断请求）（参照图 11.12）。
IIOi <sub>j</sub> 引脚（输出）（组 2 为 OUTC2 <sub>j</sub> 引脚）	脉冲输出
选择功能	<ul style="list-style-type: none"> <li>初始值的设定功能 设定开始输出波形时的输出电平。</li> <li>反相输出功能 在将输出波形的电平反相后，从 IIOi<sub>j</sub> 引脚（组 2 为 OUTC2<sub>j</sub> 引脚）输出。</li> </ul>

注 1. 当 GiPOCRj 寄存器的 INV 位为“1”（反相输出）时，“L”电平宽度和“H”电平宽度相反。

注 2. 在使用时间测量功能和波形生成功能共用的通道时，必须将 GiFS 寄存器的 FSCj 位置“0”（选择波形生成功能）。

(1) 自由运行的情况(GiBCR寄存器的RST2~RST0位为“000b”)

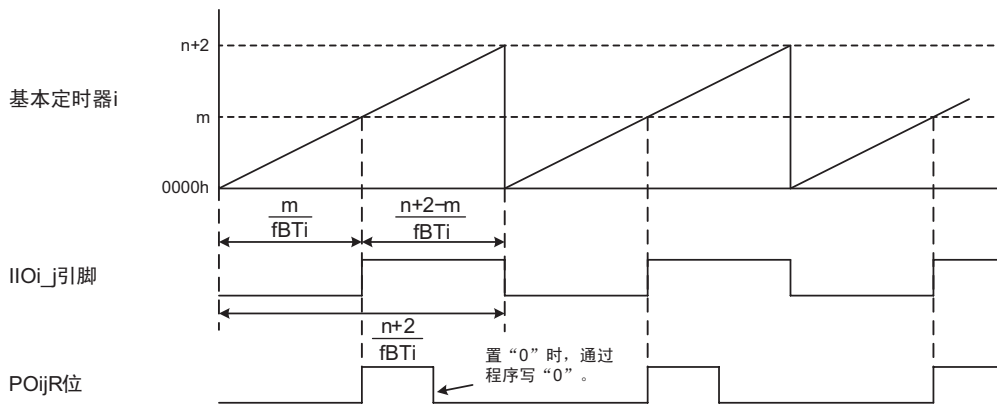


j=0~7  
 m: GiPOj寄存器的值(0000h~FFFFh)  
 POijR位: IIO0iR~IIO11iR寄存器的位

注1. 这是GiPOCRj寄存器的INV位为“0”(不反相输出)并且IVL位为“0”(将“L”电平作为初始值输出)时的波形。  
 注2. 这是INV位为“0”(不反相输出)并且IVL位为“1”(将“H”电平作为初始值输出)时的波形。

此图的条件如下:  
 • GiBCR1寄存器的UD1~UD0位为“00b”(递增计数模式)。

(2) 在基本定时器和GiPO0寄存器的值相同时进行基本定时器复位的情况(RST2~RST0位为“010b”)



j=1~7  
 m: GiPOj寄存器的值(0000h~FFFFh)  
 n: GiPO0寄存器的值(0001h~FFFDh)  
 POijR位: IIO0iR~IIO11iR寄存器的位

此图的条件如下:  
 • GiPOCRj寄存器的IVL位为“0”(将“L”电平作为初始值输出)并且INV位为“0”(不反相输出)。  
 • GiBCR1寄存器的UD1~UD0位为“00b”(递增计数模式)。  
 • m<n+2

图 23.25 单相波形输出模式的运行例子 (i=0 ~ 2)

### 23.3.2 反相波形输出模式（组 0～组 2）

在每次基本定时器和 GiPOj 寄存器（i=0～2，j=0～7）的值相同时，将 IIOi\_j 引脚（组 2 为 OUTC2\_j 引脚）的输出电平反相。

反相波形输出模式的规格和运行例子分别如表 23.8 和图 23.26 所示。

表 23.8 反相波形输出模式的规格（i=0～2）

项目	规格
输出波形	<ul style="list-style-type: none"> <li>自由运行（GiBCR1 寄存器的 RST2～RST0 位为“000b”）的情况：            周期：<math display="block">\frac{65536 \times 2}{fBTi}</math>  “H”电平宽度和“L”电平宽度：<math display="block">\frac{65536}{fBTi}</math>  GiPOj 寄存器（j=0～7）的设定值（0000h～FFFFh）         </li> <li>在基本定时器和 GiPO0 寄存器的值相同时，将基本定时器置“0000h”（RST2～RST0 位为“010b”）的情况：            周期：<math display="block">\frac{2(n+2)}{fBTi}</math>  “H”电平宽度和“L”电平宽度：<math display="block">\frac{n+2}{fBTi}</math>  n: GiPO0 寄存器的设定值（0001h～FFFDh）            GiPOj 寄存器（j=1～7）的设定值（0000h～FFFFh）            当 GiPOj 寄存器的值 <math>\geq n+2</math> 时，输出电平不反相。         </li> </ul>
波形输出开始条件（注 1）	将 GiFE 寄存器的 IFEj 位（j=0～7）置“1”（通道 j 的功能有效）。
波形输出停止条件	将 GiFE 寄存器的 IFEj 位置“0”（通道 j 的功能无效）。
中断请求	如果基本定时器和 GiPOj 寄存器的值相同，智能 I/O 的中断请求寄存器的 POijR 位就变为“1”（有中断请求）（参照图 11.12）。
IIOi_j 引脚（输出）（组 2 为 OUTC2_j 引脚）	脉冲输出
选择功能	<ul style="list-style-type: none"> <li>初始值的设定功能 设定开始输出波形时的输出电平。</li> <li>反相输出功能 在将输出波形的电平反相后，从 IIOi_j 引脚（组 2 为 OUTC2_j 引脚）输出。</li> </ul>

注 1. 在使用时间测量功能和波形生成功能共用的通道时，必须将 GIFS 寄存器的 FSCj 位置“0”（选择波形生成功能）。

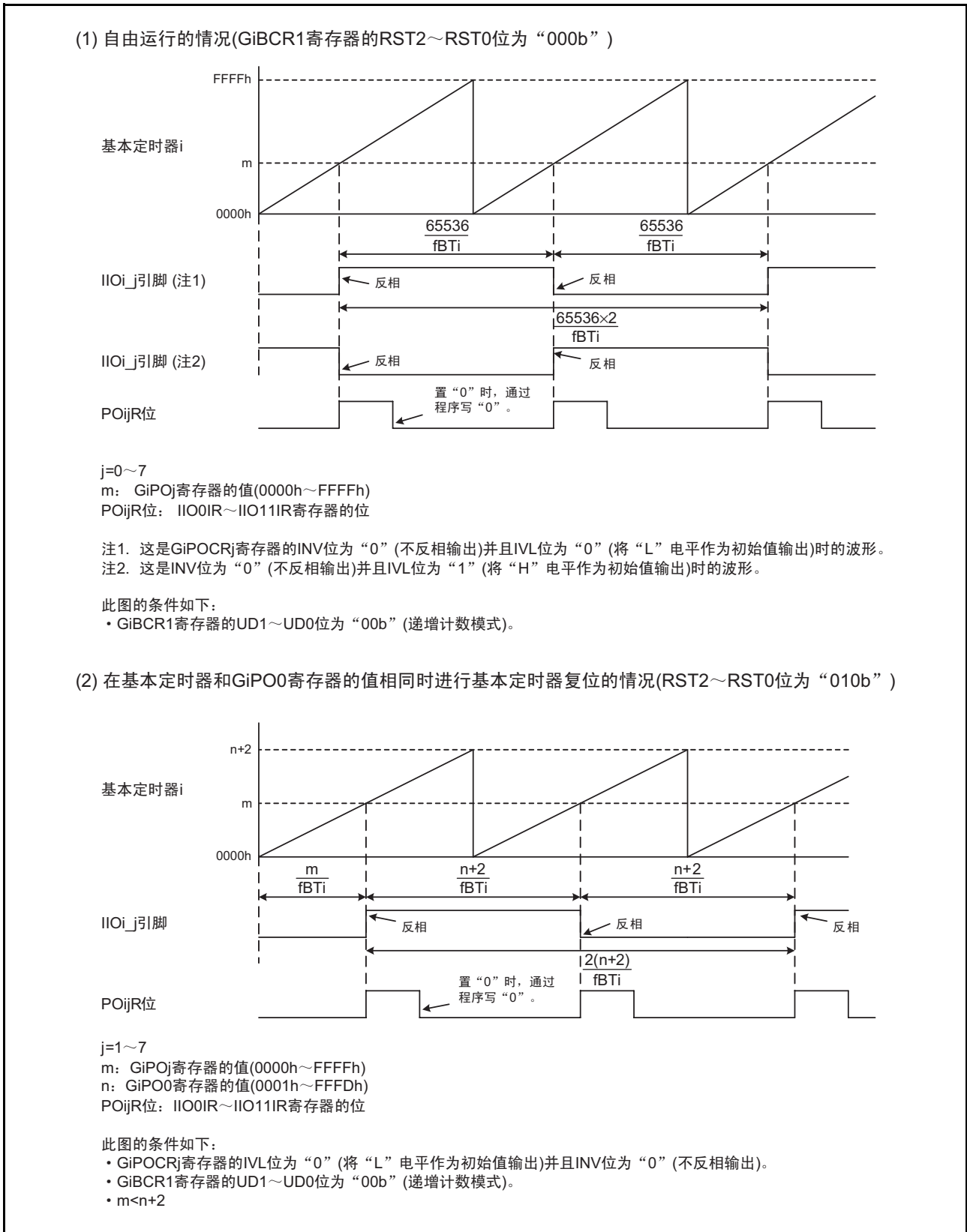


图 23.26 反相波形输出模式的运行例子 (i=0 ~ 2)

### 23.3.3 置位 - 复位波形输出（SR 波形输出）模式（组 0 ~ 组 2）

如果基本定时器和 GiPOj 寄存器（ $i=0 \sim 2$ ,  $j=0,2,4,6$ ）的值相同，IIOi<sub>j</sub> 引脚（组 2 为 OUTC2<sub>j</sub> 引脚）的输出电平就变为“H”。如果基本定时器和 GiPOk 寄存器（ $k=j+1$ ）的值相同或者基本定时器变为“0000h”，输出电平就变为“L”电平。如果将 GiPOCRj 寄存器（ $j=0 \sim 7$ ）的 IVL 位置“1”（将“H”电平作为初始值输出），开始输出波形时的输出电平就变为“H”。如果将 INV 位置“1”（反相输出），就将输出波形电平反相后输出。详细内容请参照“图 23.27 SR 波形输出模式的运行例子（ $i=0 \sim 2$ ）”。SR 波形输出模式的规格如表 23.9 所示。

表 23.9 SR 波形输出模式的规格 (i=0 ~ 2)

项目	规格
输出波形 (注 1)	<ul style="list-style-type: none"> <li>自由运行 (GiBCR1 寄存器的 RST2 ~ RST0 位为 “000b”) 的情况:               <ul style="list-style-type: none"> <li>(1) 当 <math>m &lt; n</math> 时                   <ul style="list-style-type: none"> <li>“H” 电平宽度: <math>\frac{n-m}{fBTi}</math></li> <li>“L” 电平宽度: <math>\frac{m}{fBTi}</math> (注2) + <math>\frac{65536-n}{fBTi}</math> (注3)</li> </ul> </li> <li>(2) 当 <math>m \geq n</math> 时                   <ul style="list-style-type: none"> <li>“H” 电平宽度: <math>\frac{65536-m}{fBTi}</math></li> <li>“L” 电平宽度: <math>\frac{m}{fBTi}</math></li> </ul> </li> </ul> <p>m: GiPOj 寄存器 (j=0,2,4,6) 的设定值 n: GiPOk 寄存器 (k=j+1) 的设定值 m 和 n 的值为 “0000h ~ FFFFh”。</p> </li> <li>在基本定时器和 GiPO0 寄存器的值相同时, 将基本定时器置 “0000h” (RST2 ~ RST0 位为 “010b”) 的情况 (注 4):               <ul style="list-style-type: none"> <li>(1) 当 <math>m &lt; n &lt; p+2</math> 时                   <ul style="list-style-type: none"> <li>“H” 电平宽度: <math>\frac{n+m}{fBTi}</math></li> <li>“L” 电平宽度: <math>\frac{m}{fBTi}</math> (注2) + <math>\frac{p+2-n}{fBTi}</math> (注3)</li> </ul> </li> <li>(2) 当 <math>m &lt; p+2 \leq n</math> 时                   <ul style="list-style-type: none"> <li>“H” 电平宽度: <math>\frac{p+2-m}{fBTi}</math></li> <li>“L” 电平宽度: <math>\frac{m}{fBTi}</math></li> </ul> </li> <li>(3) 当 <math>m \geq p+2</math> 时, 输出电平固定为 “L”。 p: GiPO0 寄存器的设定值 m: GiPOj 寄存器 (j=2,4,6) 的设定值 n: GiPOk 寄存器 (k=j+1) 的设定值 p 的值为 “0001h ~ FFFDh”, m 和 n 的值为 “0000h ~ FFFFh”。</li> </ul> </li> </ul>
波形输出开始条件 (注 5)	将 GiFE 寄存器的 IFEq 位 (q=0 ~ 7) 置 “1” (通道 q 的功能有效)。
波形输出停止条件	将 GiFE 寄存器的 IFEq 位置 “0” (通道 q 的功能无效)。
中断请求	如果基本定时器的值和 GiPOj 寄存器的值相同, 智能 I/O 的中断请求寄存器的 POijR 位就变为 “1”。如果基本定时器的值和 GiPOk 寄存器的值相同, POikR 位就变为 “1” (有中断请求) (参照图 11.12)。
IIOi_j 引脚 (输出) (组 2 为 OUTC2_j 引脚)	脉冲输出
选择功能	<ul style="list-style-type: none"> <li>初始值的设定功能 设定开始输出波形时的输出电平。</li> <li>反相输出功能 在将输出波形的电平反相后, 从 IIOi_j 引脚 (组 2 为 OUTC2_j 引脚) 输出。</li> </ul>

注 1. 当 GiPOCRj 寄存器的 INV 位为 “1” (反相输出) 时, “L” 电平宽度和 “H” 电平宽度相反。

注 2. 这是从基本定时器复位后到输出电平变为 “H” 前的期间。

注 3. 这是从输出电平变为 “L” 后到基本定时器复位前的期间。

注 4. 如果在 GiPO0 寄存器和基本定时器的值相同时进行基本定时器复位, 就不能使用通道 0 和通道 1 的 SR 波形生成功能。

注 5. 在使用时间测量功能和波形生成功能共用的通道时, 必须将 GiFS 寄存器的 FSCj 位置 “0” (选择波形生成功能)。



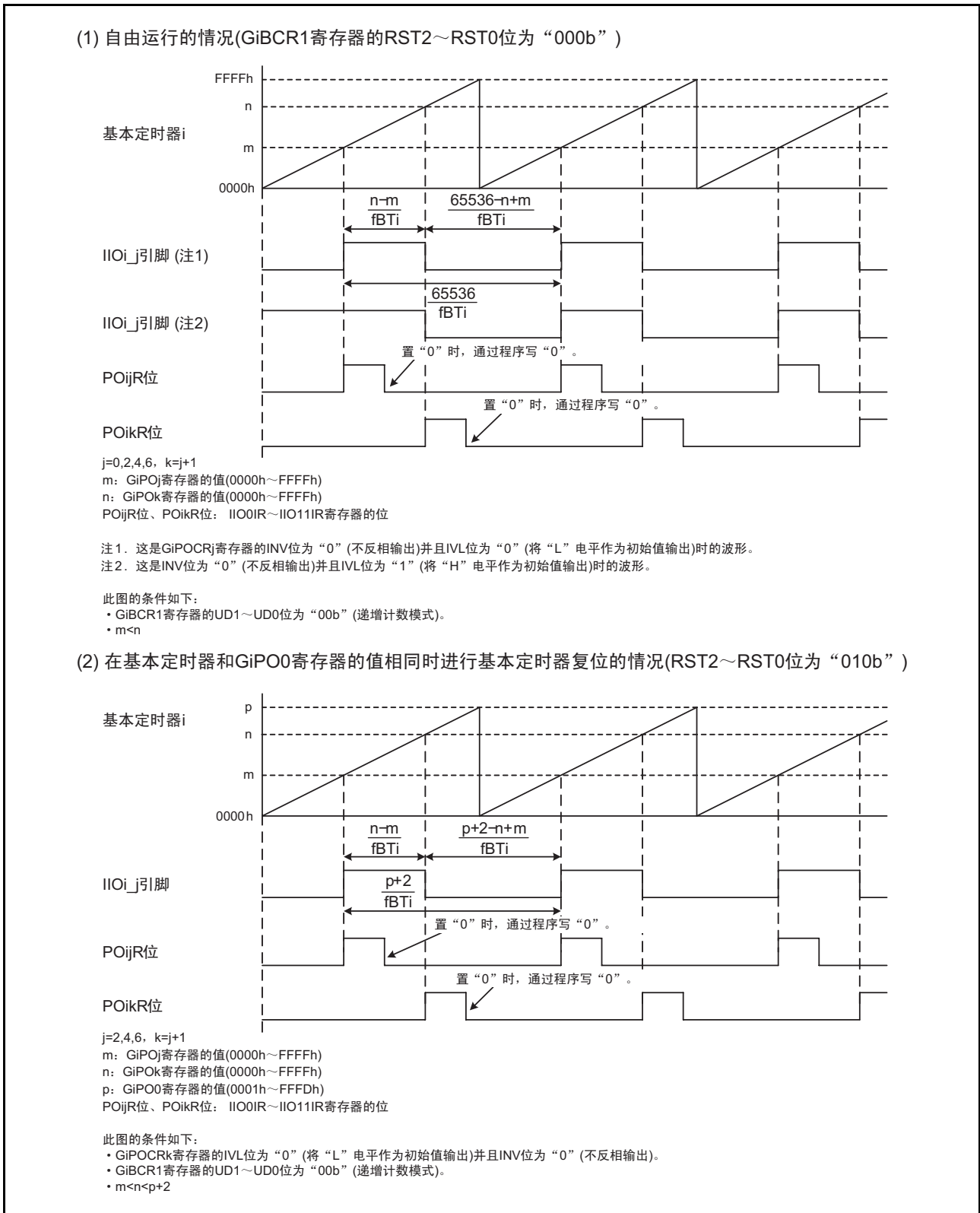


图 23.27 SR 波形输出模式的运行例子 (i=0 ~ 2)

### 23.3.4 位调制 PWM 输出模式（组 2）

在位调制 PWM 输出模式中，能进行 16 位分辨率的 PWM 输出。

重复输出以连续 1024 次的小区间  $t$  为 1 个周期的脉冲。小区间  $t$  的周期为  $\frac{64}{f_{BT2}}$ ，通过 G2POj 寄存器（ $j=0 \sim 7$ ）的高 6 位设定“L”电平的基本宽度，通过低 10 位设定小区间的个数（在附加最小分辨率位宽（1 个时钟）的“L”电平宽度的 1 个周期内）。

如果将 INV 位置“1”（反相输出），就将输出波形的电平反相后输出。

位调制 PWM 输出模式的规格和运行例子分别如表 23.10 和图 23.28 所示，附加调制的小区间个数与附加最小分辨率位宽的小区间  $t$  的关系如表 23.11 所示。

表 23.10 位调制 PWM 输出模式的规格

项目	规格
输出波形（注 1、注 2）	PWM 的重复周期 T: $\frac{65536}{f_{BT2}} \left( = \frac{64}{f_{BT2}} \times 1024 \right)$ 小区间 $t$ 的周期: $\frac{64}{f_{BT2}}$ “L”电平宽度: m 区间为 $\frac{n+1}{f_{BT2}}$ , (1024-m) 区间为 $\frac{n}{f_{BT2}}$ “L”电平的平均宽度: $\frac{1}{f_{BT2}} \times \left( n + \frac{m}{1024} \right)$ n: G2POj 寄存器（ $j=0 \sim 7$ ）的设定值（高 6 位）（00h ~ 3Fh） m: G2POj 寄存器的设定值（低 10 位）（000h ~ 3FFh）
波形输出开始条件	将 G2FE 寄存器的 IFEj 位置“1”（通道 j 的功能有效）。
波形输出停止条件	将 IFEj 位置“0”（通道 j 的功能无效）。
中断请求	如果基本定时器的低 6 位和 G2POj 寄存器的高 6 位的设定值相同，中断请求寄存器的 PO2jR 位就变为“1”（有中断请求）（参照图 11.12）。
OUTC2_j 引脚	脉冲输出
选择功能	<ul style="list-style-type: none"> <li>初始值的设定功能 设定开始输出波形时的输出电平。</li> <li>反相输出功能 在将输出波形的电平反相后，从 OUTC2_j 引脚输出。</li> </ul>

注 1. 在使用位调制 PWM 输出模式时，必须将 G2BCR1 寄存器的 RST2 ~ RST0 位置“000b”。

注 2. 当 G2POCRj 寄存器的 INV 位为“1”时，“L”电平宽度和“H”电平宽度相反。

表 23.11 附加调制的小区间个数与附加最小分辨率位宽的小区间  $t$  的关系

附加调制的小区间个数	附加最小分辨率位宽的小区间
00 0000 0000b	无
00 0000 0001b	t512
00 0000 0010b	t256、t768
00 0000 0100b	t128、t384、t640、t896
00 0000 1000b	t64、t192、t320、t448、t576、t704、t832、t960
:	:
10 0000 0000b	t1、t3、t5、t7、……、t1019、t1021、t1023

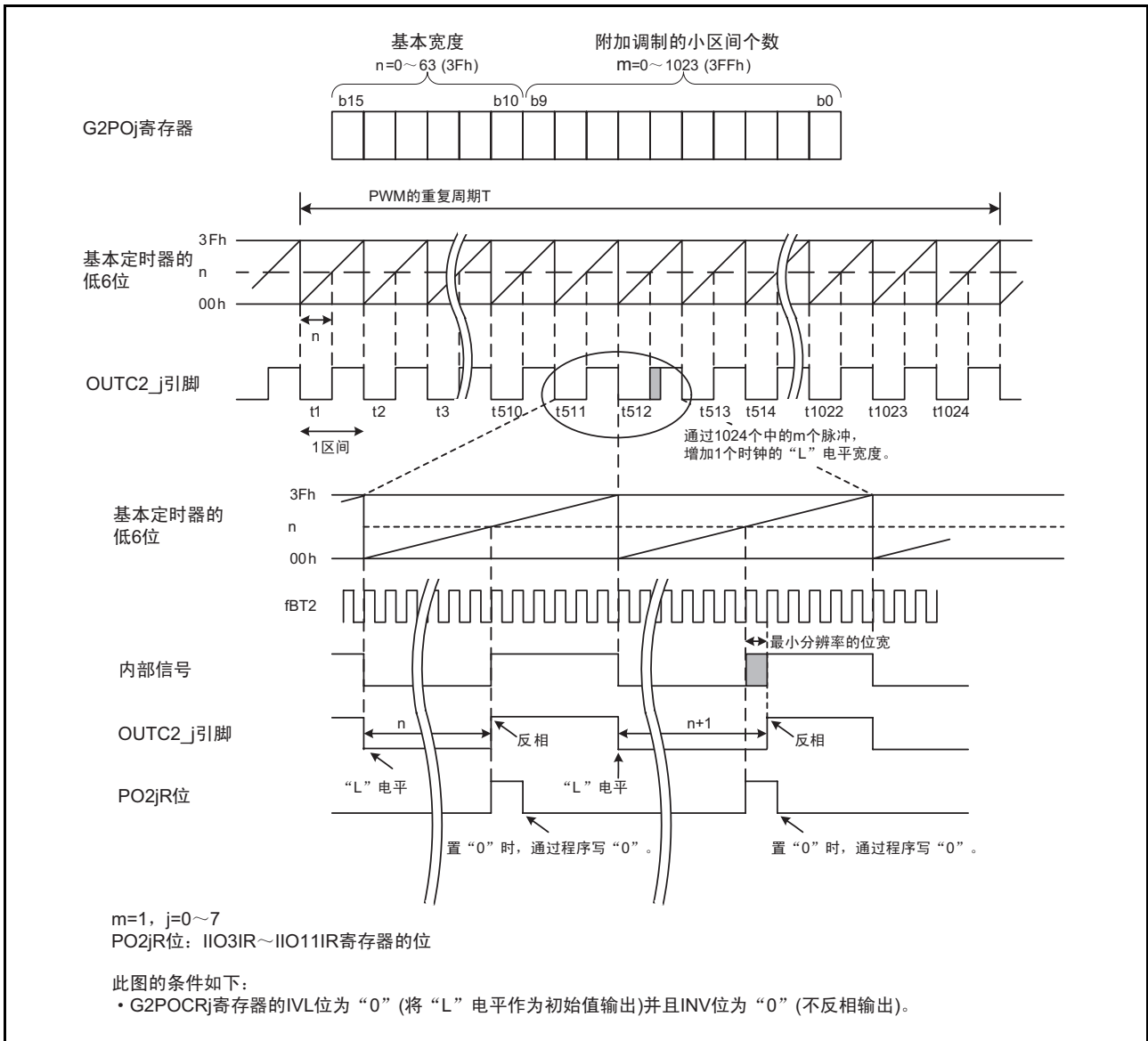


图 23.28 位调制 PWM 输出模式的运行例子

### 23.3.5 实时端口（RTP）输出模式（组 2）

如果基本定时器和 G2PO<sub>j</sub> 寄存器（j=0 ~ 7）的值相同，就以位为单位从 OUTC2<sub>j</sub> 引脚输出 G2RTP 寄存器的设定值。RTP 输出模式的规格和运行例子分别如表 23.12 和图 23.30 所示，RTP 输出功能的框图如图 23.29 所示。

表 23.12 RTP 输出模式的规格

项目	规格
波形输出开始条件	将 G2FE 寄存器的 IFE <sub>j</sub> 位（j=0 ~ 7）置“1”（通道 j 的功能有效）。
波形输出停止条件	将 G2FE 寄存器的 IFE <sub>j</sub> 位置“0”（通道 j 的功能无效）。
中断请求	如果基本定时器和 G2PO <sub>j</sub> 寄存器的值（0000h ~ FFFFh（注 1））相同，中断请求寄存器的 PO2 <sub>j</sub> R 位就变为“1”（有中断请求）（参照图 11.12）。
OUTC2 <sub>j</sub> 引脚	RTP 输出
选择功能	<ul style="list-style-type: none"> <li>初始值的设定功能 设定开始输出波形时的输出电平。</li> <li>反相输出功能 在将输出波形的电平反相后，从 OUTC2<sub>j</sub> 引脚输出。</li> </ul>

注 1. 在基本定时器和 G2PO0 寄存器的值相同时，如果要将基本定时器置“0000h”（G2BCR1 寄存器的 RST2 ~ RST0 位为“010b”），就必须将 G2PO0 寄存器的值设定为“0001h ~ FFFDh”。

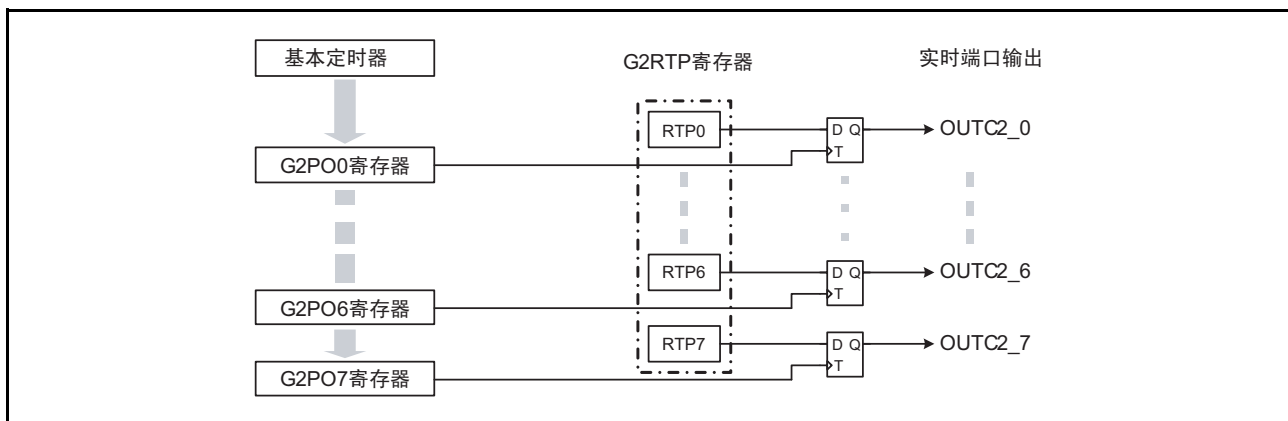
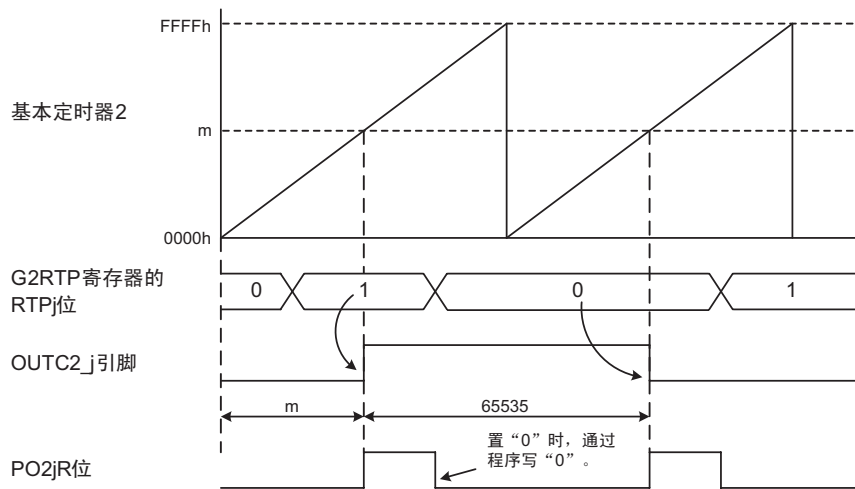


图 23.29 RTP 输出功能的框图

(1) 自由运行的情况(G2BCR1寄存器的RST2~RST0位为“000b”)



$j=0\sim7$

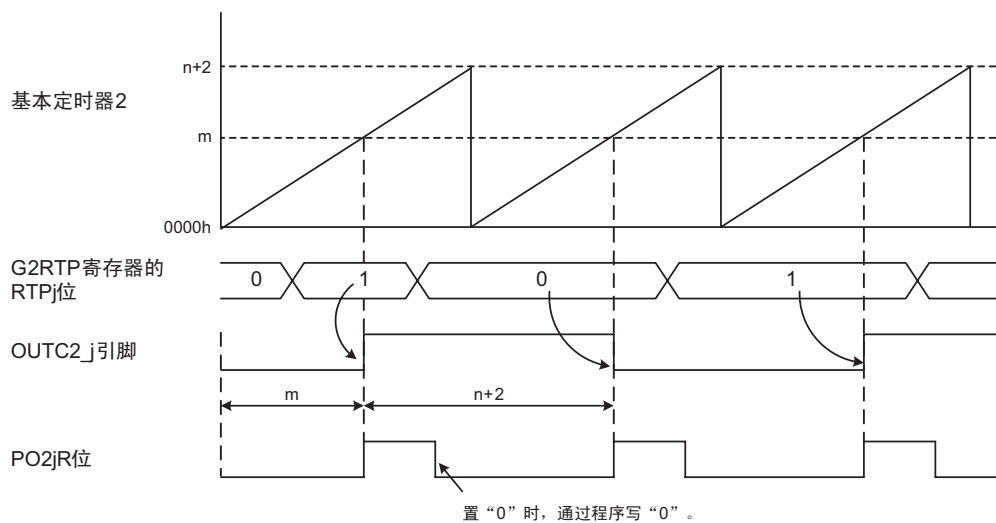
m: G2PO<sub>j</sub>寄存器的值(0000h~FFFFh)

PO2jR位: IIO03R~IIO111R寄存器的位

此图的条件如下:

- G2POCR<sub>j</sub>寄存器的IVL位为“0”(将“L”电平作为初始值输出)并且IVL位为“0”(不反相输出)。

(2) 在基本定时器和G2PO0寄存器的值相同时进行基本定时器复位的情况(RST2~RST0位为“010b”)



$j=1\sim7$

m: G2PO<sub>j</sub>寄存器的值(0000h~FFFFh)

n: G2PO<sub>0</sub>寄存器的值(0001h~FFFDh)

PO2jR位: IIO03R~IIO111R寄存器的位

此图的条件如下:

- G2POCR<sub>j</sub>寄存器的IVL位为“0”(将“L”电平作为初始值输出)并且INV位为“0”(不反相输出)。
- $m < n+2$

图 23.30 RTP 输出模式的运行例子

### 23.3.6 并行实时端口（RTP）输出模式（组 2）

如果基本定时器和 G2PO<sub>j</sub> 寄存器（j=0 ~ 7）的值相同，就以字节为单位从 OUTC2<sub>j</sub> 引脚输出 G2RTP 寄存器的全部设定值。并行 RTP 输出模式的规格和运行例子分别如表 23.13 和图 23.32 所示，G2BCR1 寄存器和并行 RTP 输出功能的框图分别如图 23.7 和图 23.31 所示。

表 23.13 并行 RTP 输出模式的规格

项目	规格
波形输出开始条件	将 G2FE 寄存器的 IFE <sub>j</sub> 位（j=0 ~ 7）置“1”（通道 j 的功能有效）。
波形输出停止条件	将 IFE <sub>j</sub> 位置“0”（通道 j 的功能无效）。
中断请求	如果基本定时器和 G2PO <sub>j</sub> 寄存器的值（0000h ~ FFFFh（注 1））相同，中断请求寄存器的 PO2 <sub>j</sub> R 位就变为“1”（有中断请求）（参照图 11.12）。
OUTC2 <sub>j</sub> 引脚	RTP 输出
选择功能	<ul style="list-style-type: none"> <li>初始值的设定功能 设定开始输出波形时的输出电平。</li> <li>反相输出功能 在将输出波形的电平反相后，从 OUTC2<sub>j</sub> 引脚输出。</li> </ul>

注 1. 在基本定时器和 G2PO0 寄存器的值相同时，如果要将基本定时器置“0000h”（G2BCR1 寄存器的 RST2 ~ RST0 位为“010b”），就必须将 G2PO0 寄存器的值设定为“0001h ~ FFFDh”。

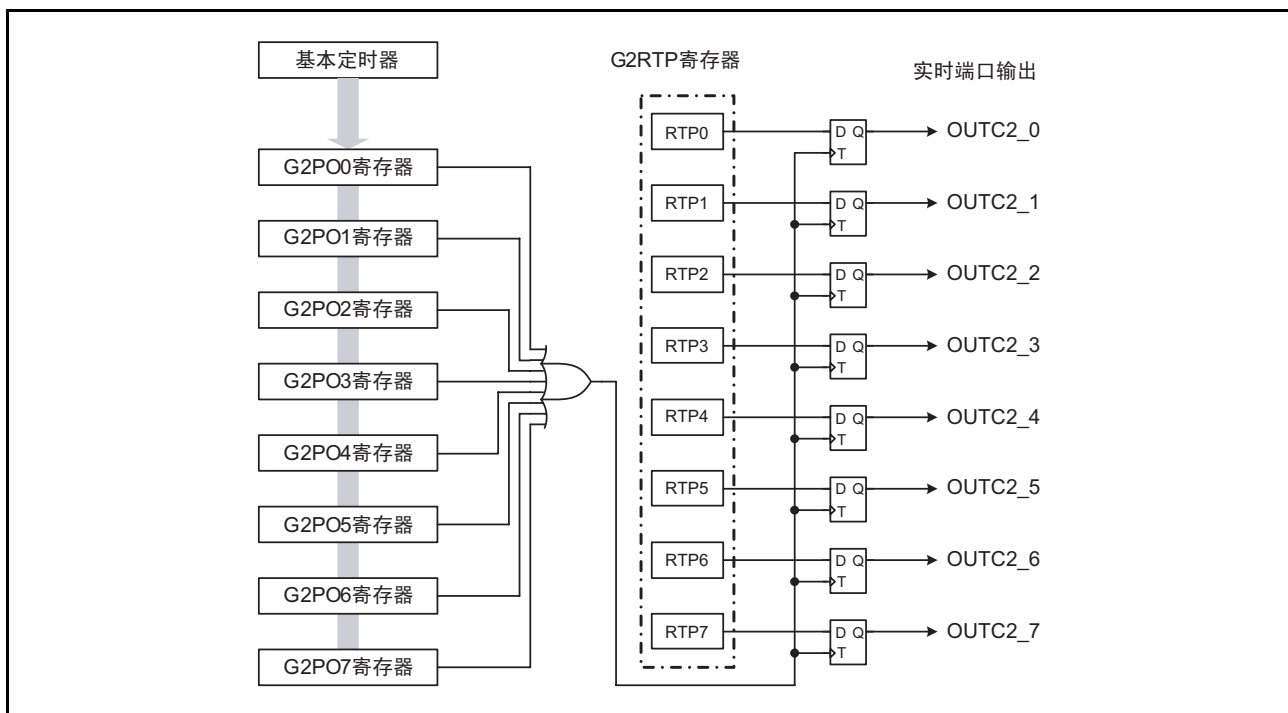


图 23.31 并行 RTP 输出功能的框图

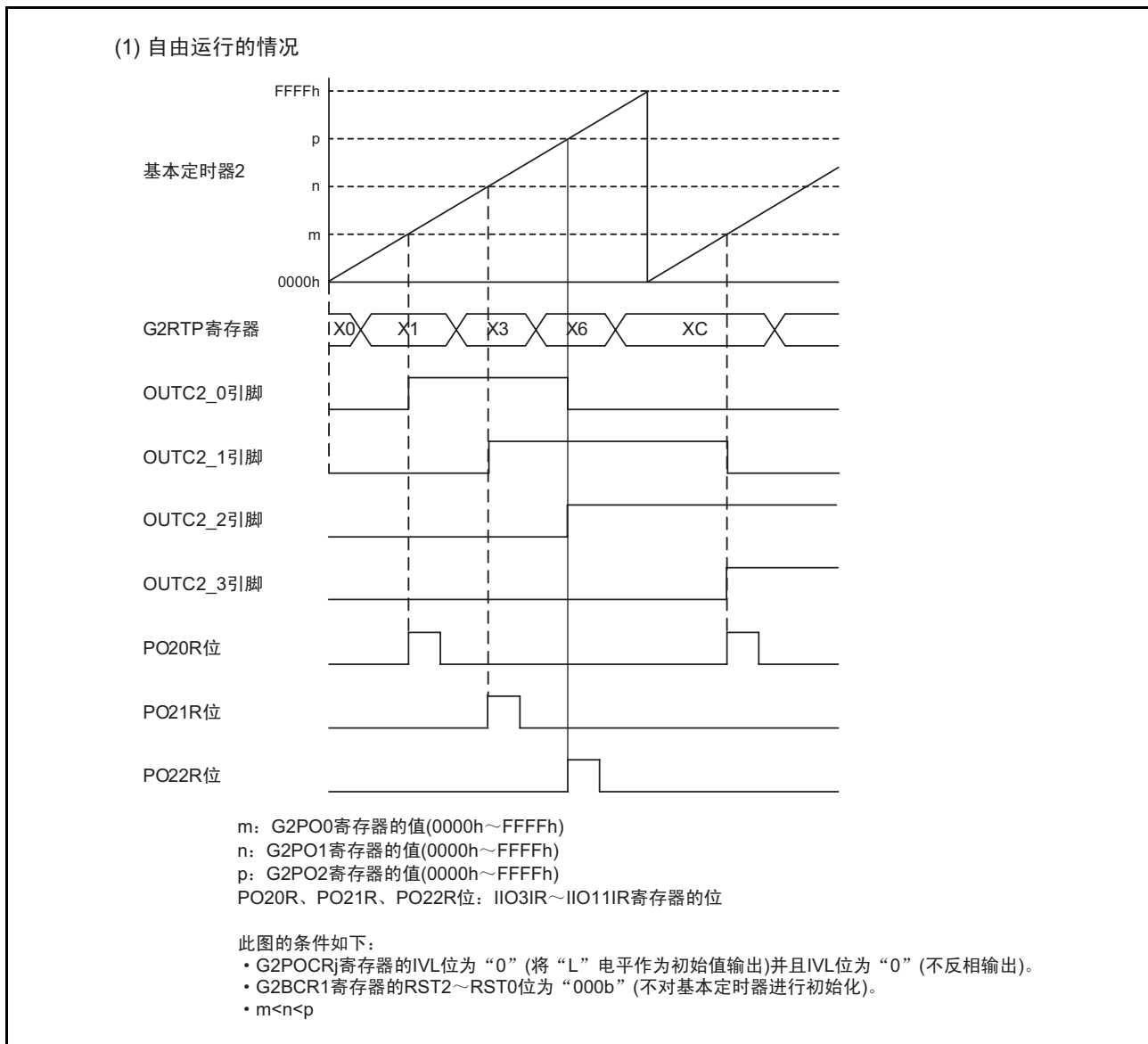


图 23.32 并行 RTP 输出模式的运行例子

## 23.4 组 2 的通信功能

使用 2 个 8 位移位寄存器和波形生成功能，使通信功能有效。

智能 I/O 的组 2 进行可变长时钟同步串行 I/O 或者 IEBus 通信（注 1）（选项（注 2））。

相关寄存器如图 23.33 ~ 图 23.40 所示。

注 1. IEBus 是瑞萨电子株式会社的注册商标。

注 2. 如果要使用选项功能，请向本公司的营业窗口询问。

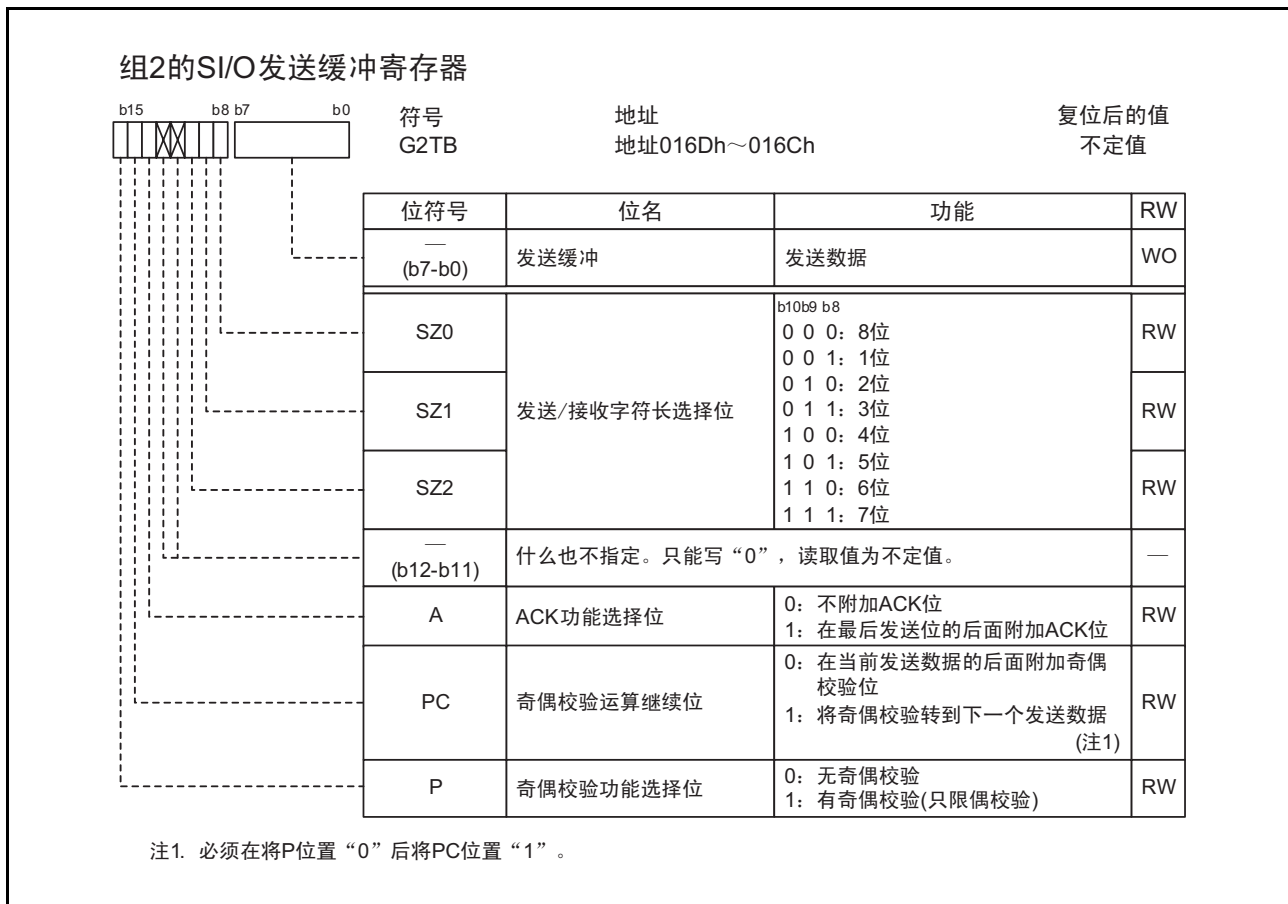


图 23.33 G2TB 寄存器

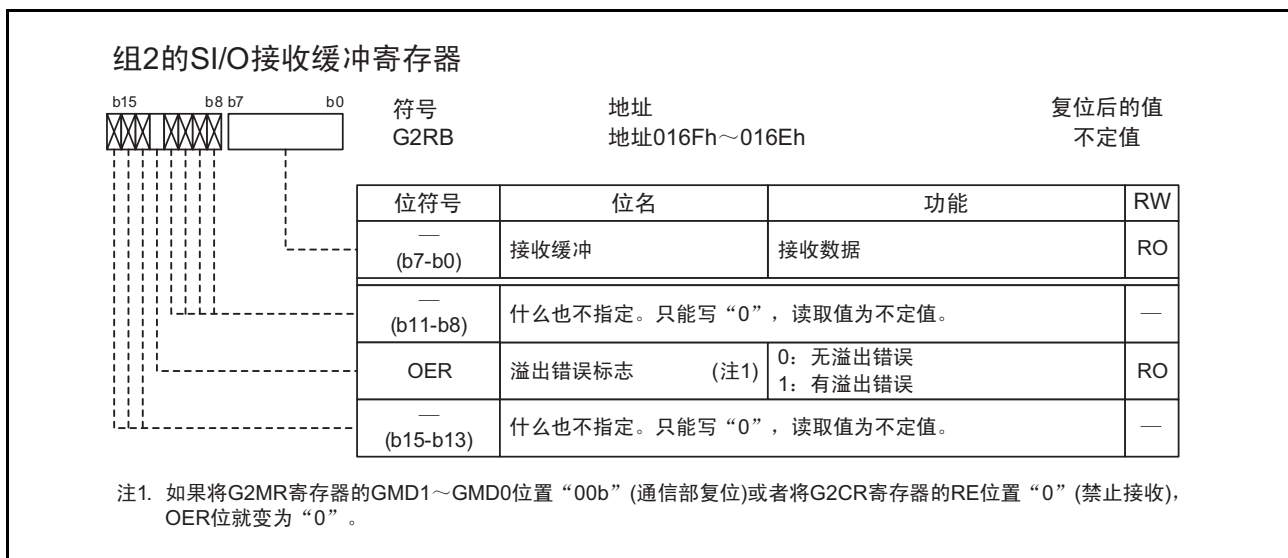


图 23.34 G2RB 寄存器



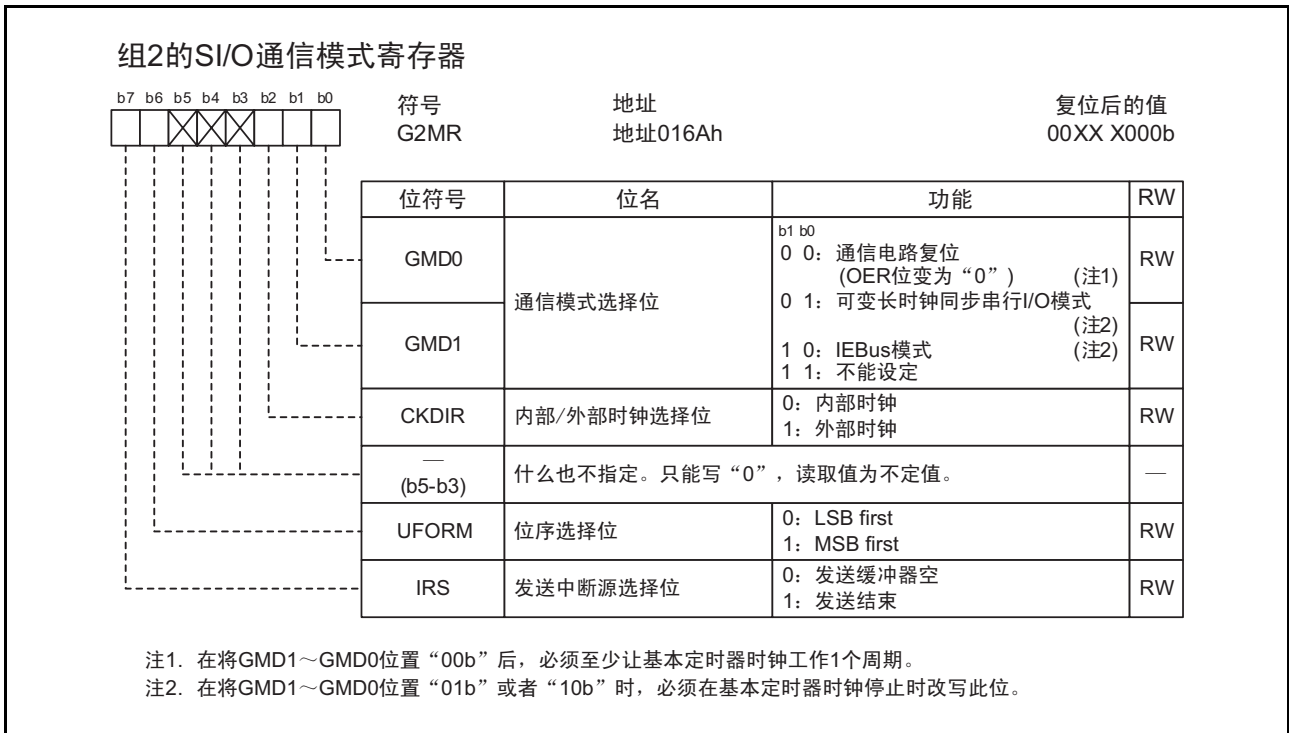


图 23.35 G2MR 寄存器

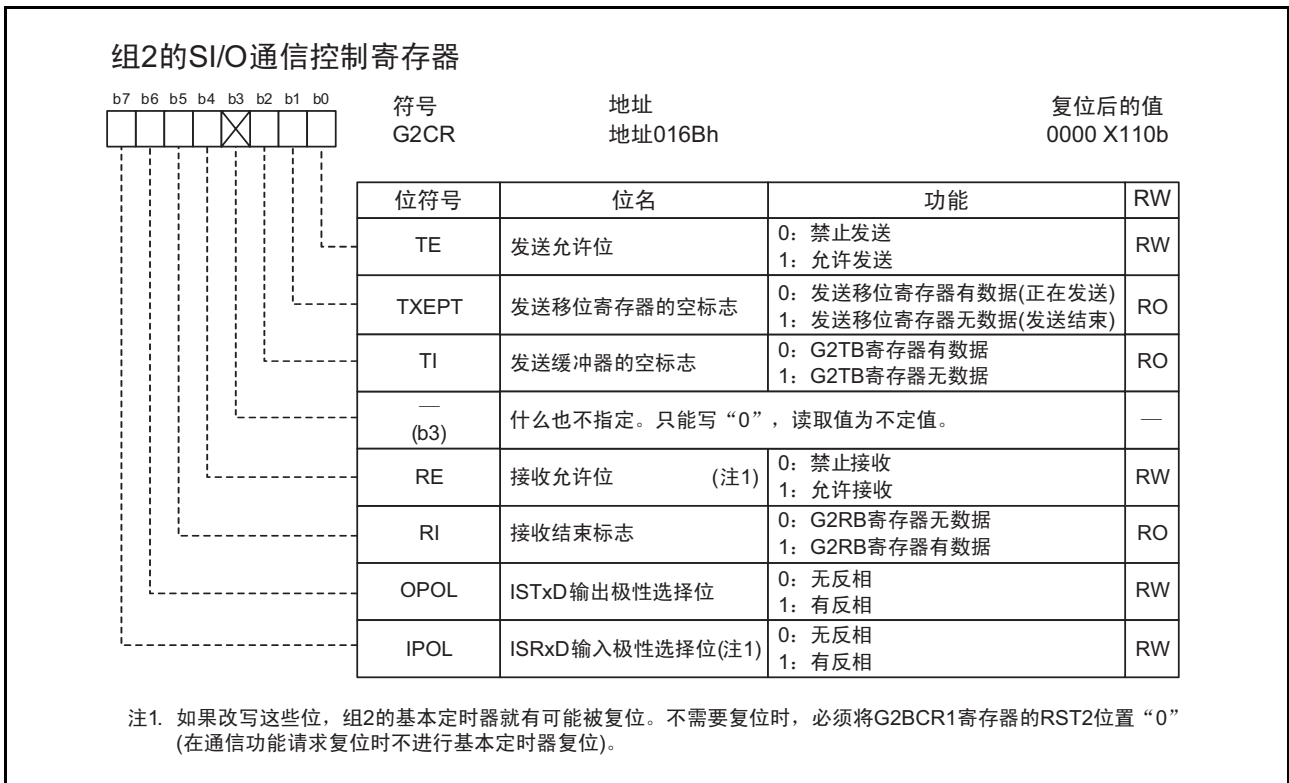


图 23.36 G2CR 寄存器

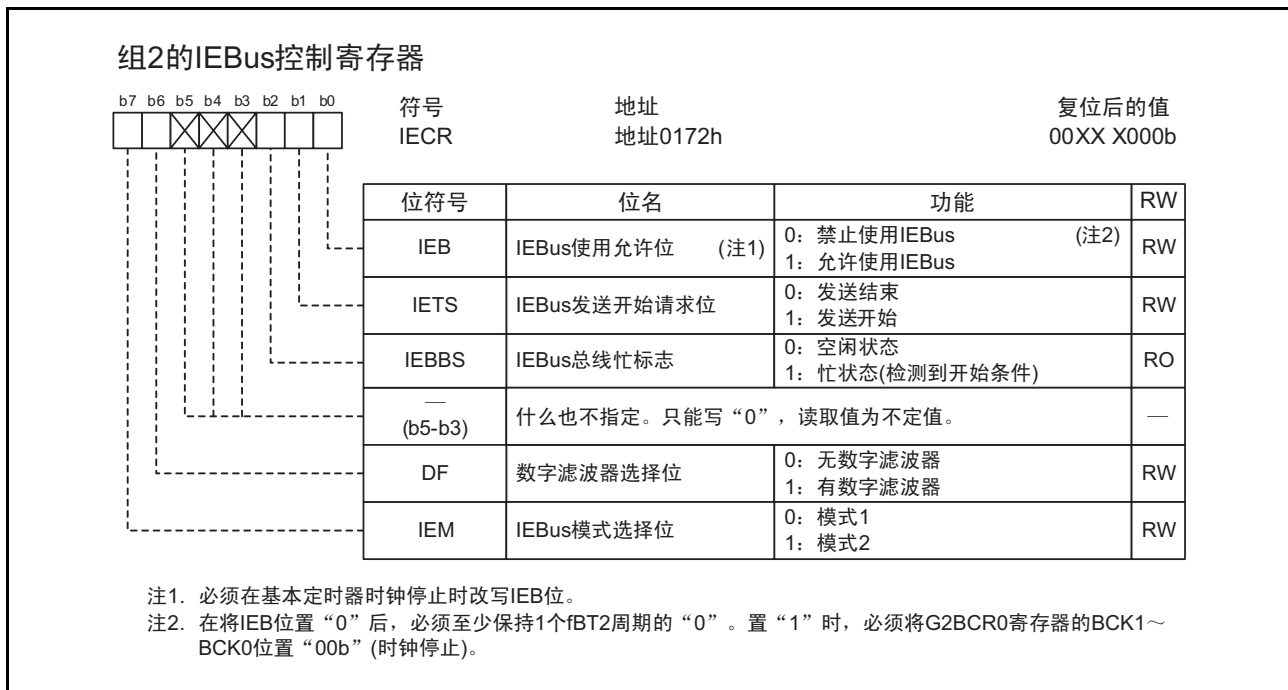


图 23.37 IECR 寄存器



图 23.38 IEAR 寄存器

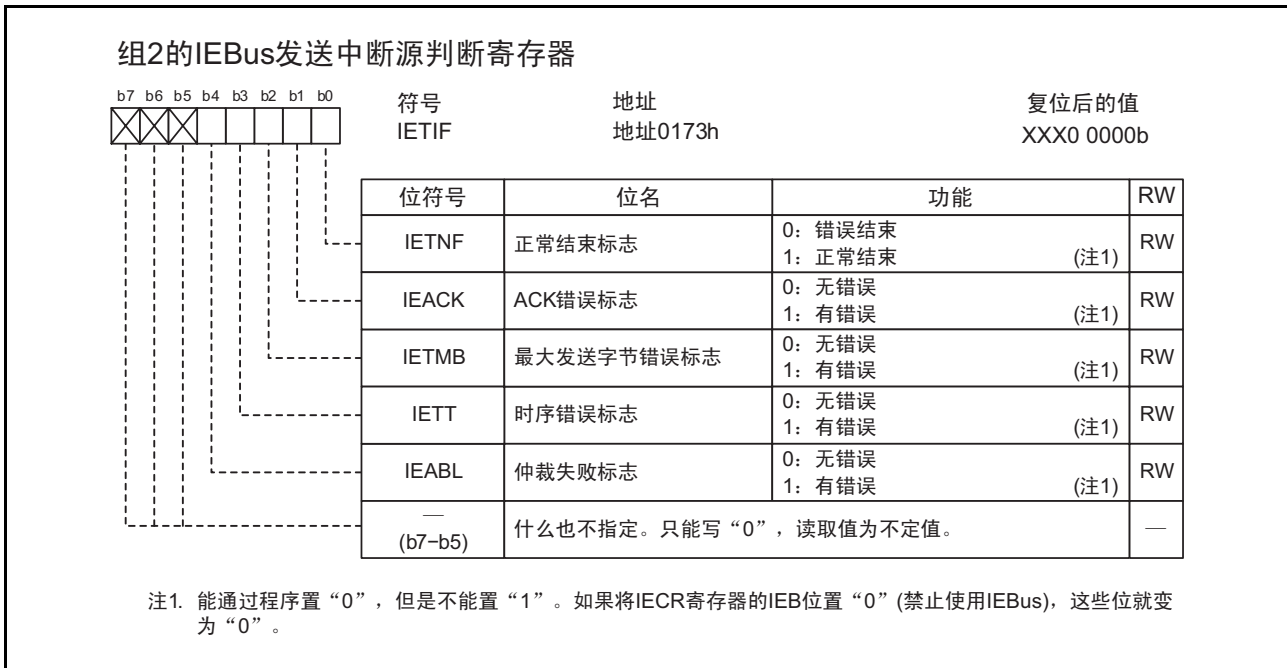


图 23.39 IETIF 寄存器

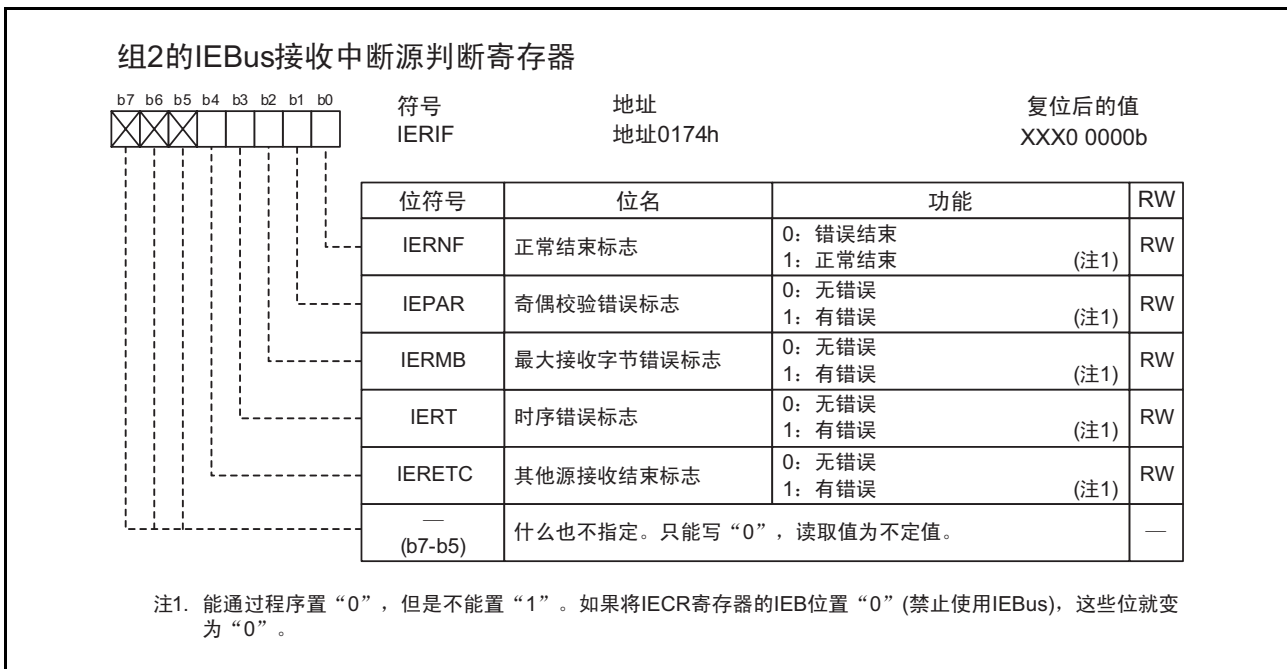


图 23.40 IERIF 寄存器

### 23.4.1 可变长时钟同步串行 I/O 模式（组 2）

这是与发送 / 接收时钟同步进行数据的发送和接收的模式，能选择 1 ~ 8 位的字符长。组 2 的可变长时钟同步串行 I/O 模式的规格以及发送 / 接收的运行例子分别如表 23.14 和图 23.41 所示，使用的寄存器及其设定值如表 23.15 所示。

表 23.14 组 2 的可变长时钟同步串行 I/O 模式的规格

项目	规格
数据格式	字符长：1 ~ 8 位
发送 / 接收时钟	<ul style="list-style-type: none"> <li>当 G2MR 寄存器的 CKDIR 位为 “0”（选择内部时钟）时：  <math display="block">\frac{f_{BT2}}{2(n+2)}</math>           n: G2PO0 寄存器的设定值（0000h ~ FFFFh）（注 1）            由 G2PO0 寄存器决定位速率，通过通道 2 波形生成功能和反相波形输出模式生成时钟。         </li> <li>当 CKDIR 位为 “1”（选择外部时钟）时：向 ISCLK2 引脚输入（注 2）</li> </ul>
发送开始条件	开始发送时需要以下的条件： <ul style="list-style-type: none"> <li>G2CR 寄存器的 TE 位为 “1”（允许发送）。</li> <li>G2CR 寄存器的 TI 位为 “0”（G2TB 寄存器有数据）。</li> </ul>
接收开始条件	开始接收时需要以下的条件： <ul style="list-style-type: none"> <li>G2CR 寄存器的 RE 位为 “1”（允许接收）。</li> <li>G2CR 寄存器的 TE 位为 “1”（允许发送）。</li> <li>G2CR 寄存器的 TI 位为 “0”（G2TB 寄存器有数据）</li> </ul>
中断请求	在发送时 能选择以下的任意条件。如果条件成立，IIO6IR 寄存器的 SIO2TR 位就变为 “1”（有中断请求）（参照图 11.12）。 <ul style="list-style-type: none"> <li>当 G2MR 寄存器的 IRS 位为 “0”（在 G2TB 寄存器为空时产生中断）时，将数据从 G2TB 寄存器传送到发送移位寄存器时（发送开始时）。</li> <li>当 IRS 位为 “1”（在发送结束时产生中断）时，发送移位寄存器的数据发送结束时。</li> </ul> 在接收时 在将数据从接收移位寄存器传送到 G2RB 寄存器时（接收结束时），IIO5IR 寄存器的 SIO2RR 位变为 “1”（有中断请求）（参照图 11.12）。
错误检测	溢出错误（注 3） 如果在读 G2RB 寄存器前开始接收下一个数据，并且在接收到下一个数据的最后位时发生溢出错误。
其他选择项目	<ul style="list-style-type: none"> <li>位序选择 能选择 LSB first 或者 MSB first。</li> <li>串行数据的极性转换 能将 ISTXD2 引脚和 ISRXD2 引脚的发送 / 接收数据的极性进行反转。</li> <li>发送 / 接收字符长的选择 能设定 1 ~ 8 位的发送 / 接收字符长。</li> </ul>

注 1. 在使用通信功能时，G2PO0 寄存器的设定值至少为 1。

注 2. 发送 / 接收时钟的频率不能超过 fBT2 的 20 分频的频率。

注 3. 如果发生溢出错误，G2RB 寄存器就为不定值。

表 23.15 组 2 的可变长时钟同步串行 I/O 模式中使用的寄存器及其设定值

寄存器	位	功能
G2BCR0	BCK1 ~ BCK0	必须置“11b”。
	DIV4 ~ DIV0	必须选择计数源的分频比。
	IT	必须置“0”。
G2BCR1	7 ~ 0	必须置“0001 0010b”。
G2POCR0	7 ~ 0	必须置“0000 0111b”。
G2POCR1	7 ~ 0	必须置“0000 0111b”。
G2POCR2	7 ~ 0	必须置“0000 0010b”。
G2PO0	15 ~ 0	必须设定用于生成波形的比较值。 $\frac{f_{BT2}}{2 \times (\text{设定值} + 2)}$ 为发送 / 接收时钟的频率。
G2PO2	15 ~ 0	必须设定小于 G2PO0 寄存器的值。
G2FE	IFE2 ~ IFE0	必须置“111b”。
G2MR	GMD1 ~ GMD0	必须置“01b”。
	CKDIR	必须选择内部时钟或者外部时钟。
	UFORM	必须选择 LSB first 或者 MSB first。
	IRS	必须选择发送中断源。
G2CR	TE	在允许发送 / 接收时，必须置“1”。
	TXEPT	发送移位寄存器的空标志
	TI	发送缓冲器的空标志
	RE	在允许接收时，必须置“1”。
	RI	接收结束标志
	OPOL	ISTXD2 输出极性的转换（通常必须置“0”）
	IPOL	ISRXD2 输入极性的转换（通常必须置“0”）
G2TB	15 ~ 0	必须写发送 / 接收字符长和发送数据。
G2RB	15 ~ 0	保存接收数据和错误标志。

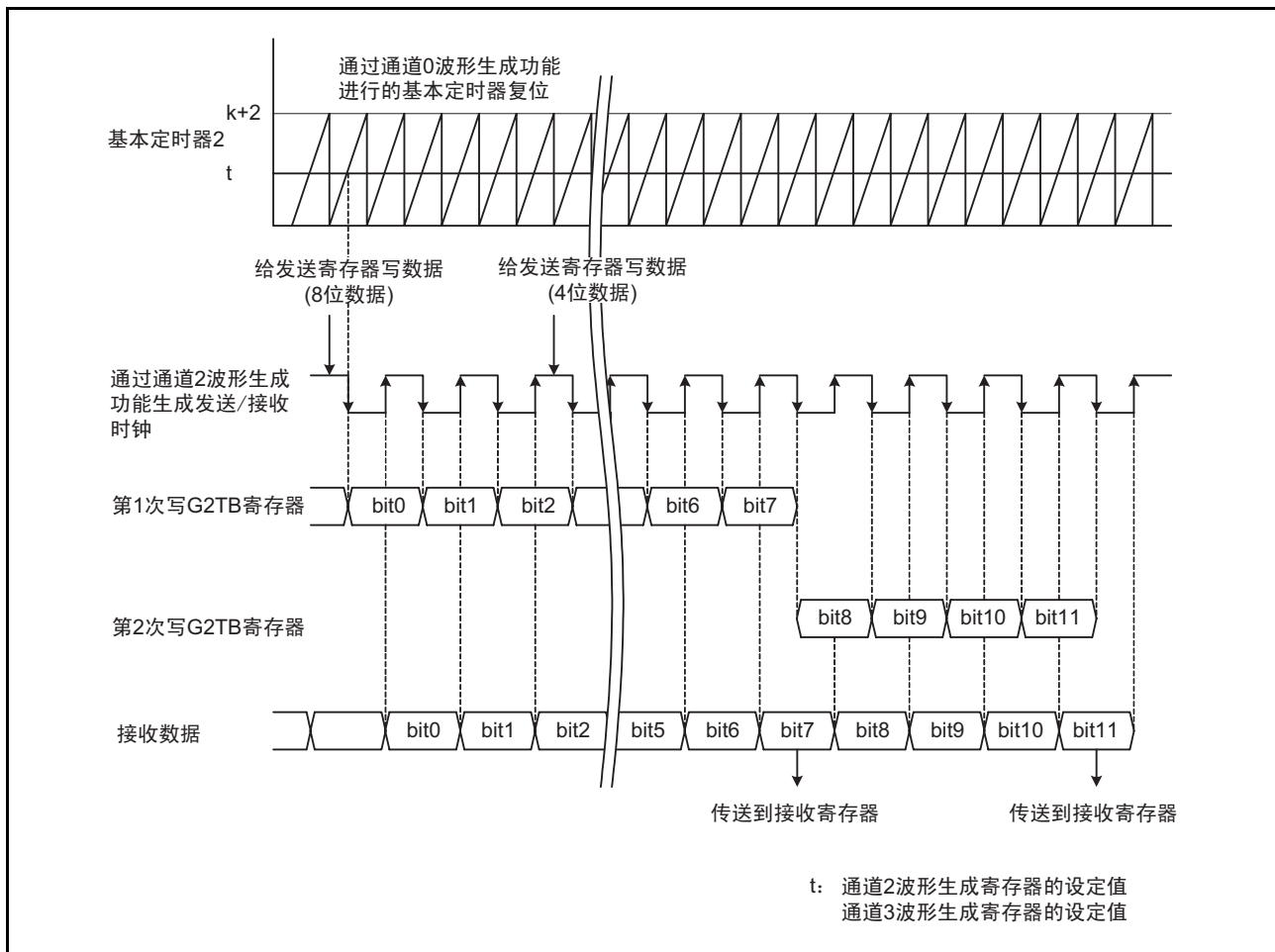


图 23.41 组 2 的可变长时钟同步串行 I/O 模式的发送 / 接收运行例子

## 24. 多主控 I<sup>2</sup>C 总线接口

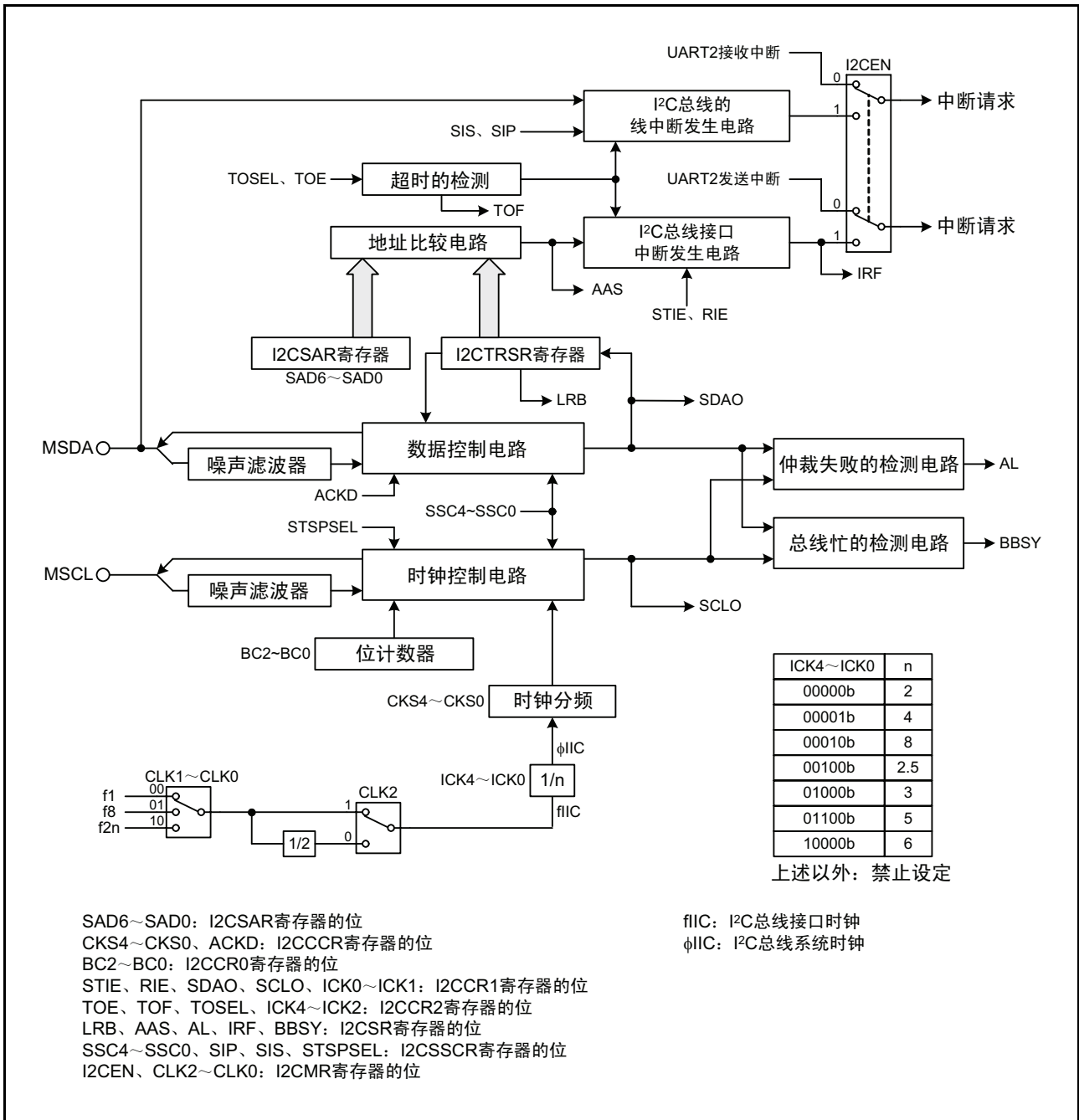
多主控 I<sup>2</sup>C 总线接口 (MMI2C) 是基于 I<sup>2</sup>C 总线的数据发送 / 接收格式进行串行发送和接收的电路, 具有仲裁失败检测功能和时钟同步功能。多主控 I<sup>2</sup>C 总线接口电路的规格、多主控 I<sup>2</sup>C 总线接口电路的检测功能, 以及多主控 I<sup>2</sup>C 总线接口电路的框图分别如表 24.1、表 24.2、和图 24.1 所示。

表 24.1 多主控 I<sup>2</sup>C 总线接口电路的规格

项目	功能
数据格式	符合 I <sup>2</sup> C 总线规格 <ul style="list-style-type: none"> <li>• 7 位寻址格式</li> <li>• 高速模式</li> <li>• 标准模式</li> </ul>
主控 / 从属器件	可选择
输入 / 输出引脚	串行数据线: MSDA (SDA) 串行时钟线: MSCL (SCL)
发送 / 接收时钟	16.1kbps ~ 400kbps ( $\phi_{IIC}=4\text{MHz}$ )  $\phi_{IIC}$ : I <sup>2</sup> C 总线系统时钟
发送 / 接收模式	符合 I <sup>2</sup> C 总线规格 <ul style="list-style-type: none"> <li>• 主控发送</li> <li>• 主控接收</li> <li>• 从属发送</li> <li>• 从属接收</li> </ul>
中断请求	<ul style="list-style-type: none"> <li>• I<sup>2</sup>C 总线接口中断: 6 种 (发送结束、接收结束、从属地址匹配的检测、全呼的检测、停止条件的检测、超时的检测)</li> <li>• I<sup>2</sup>C 总线的线中断: 2 种 (MSDA 引脚和 MSCL 引脚的上升沿或者下降沿)</li> </ul>
选择功能	<ul style="list-style-type: none"> <li>• 超时的检测 在总线忙的情况下, 检测 MSCL 引脚变为 “H” 电平的时间是否超过规定时间。</li> <li>• 自由格式的选择 与从属地址的值无关, 选择在接收到第 1 个字节时产生中断请求。</li> </ul>

表 24.2 多主控 I<sup>2</sup>C 总线接口电路的检测功能

项目	功能
从属地址匹配的检测	在进行从属发送和接收时, 检测从属地址。如果检测到地址匹配, 就自动发送 ACK; 否则就发送 NACK, 此后不进行发送和接收。
全呼的检测	在进行从属接收时, 检测全呼。
仲裁失败的检测	检测到仲裁失败后, 立刻停止 MSDA 引脚的输出。
总线忙的检测	检测到总线忙后, 置位或者复位 BBSY 标志。



SAD6~SAD0: I2CSAR寄存器的位  
 CKS4~CKS0、ACKD: I2CCCR寄存器的位  
 BC2~BC0: I2CCR0寄存器的位  
 STIE、RIE、SDAO、SCLO、ICK0~ICK1: I2CCR1寄存器的位  
 TOE、TOF、TOSEL、ICK4~ICK2: I2CCR2寄存器的位  
 LRB、AAS、AL、IRF、BBSY: I2CSR寄存器的位  
 SSC4~SSC0、SIP、SIS、STSPSEL: I2CSSCR寄存器的位  
 I2CEN、CLK2~CLK0: I2CMR寄存器的位

fIIC: I<sup>2</sup>C总线接口时钟  
 phiIIC: I<sup>2</sup>C总线系统时钟

图 24.1 多主控 I<sup>2</sup>C 总线接口电路的框图



### 24.1 多主控 I<sup>2</sup>C 总线接口电路的相关寄存器

#### 24.1.1 多主控 I<sup>2</sup>C 总线发送 / 接收移位寄存器 (I2CTRSR)

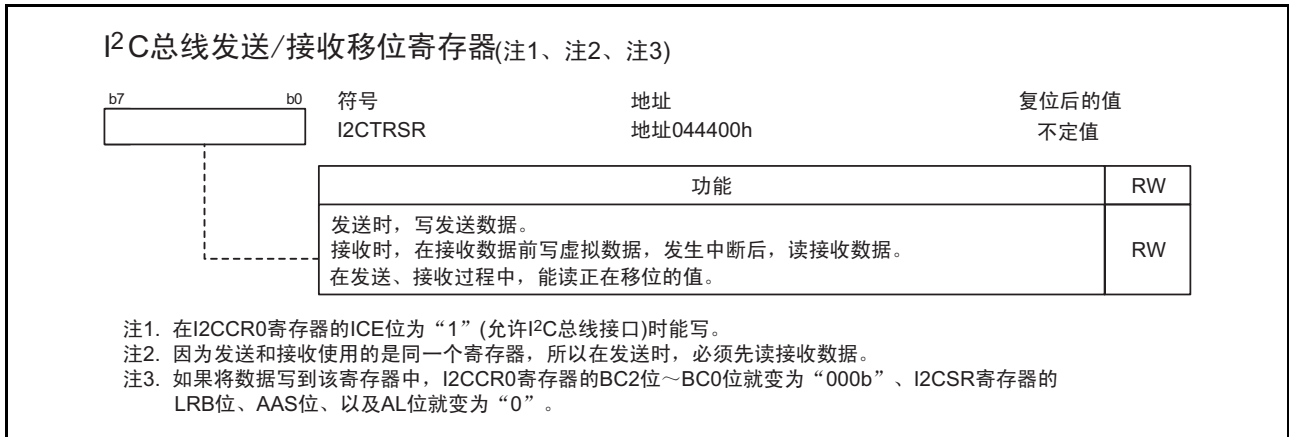


图 24.2 I2CTRSR 寄存器

I2CTRSR 寄存器是保存接收数据和写发送数据的 8 位移位寄存器。如果给 I2CTRSR 寄存器写发送数据，就与 SCL 时钟同步，从 bit7 开始按顺序向外部送出。每送出 1 位就移 1 位。数据接收时，与 SCL 时钟同步，从 bit0 开始按顺序保存，每输入 1 位就移 1 位。接收数据被保存到 I2CTRSR 寄存器的时序如图 24.3 所示。

在 I2CCR0 寄存器的 ICE 位为“1”(允许 I<sup>2</sup>C 总线接口)时，能写 I2CTRSR 寄存器。在 ICE 位为“1”、I2CSR 寄存器的 MST 位为“1”(主控模式)时，如果将数据写到 I2CTRSR 寄存器，位计数器就被复位，并且输出 SCL 时钟。

只有在生成开始条件或者向 MSCL 引脚输出“L”电平时，才能写 I2CTRSR 寄存器。可随时读 I2CTRSR 寄存器。

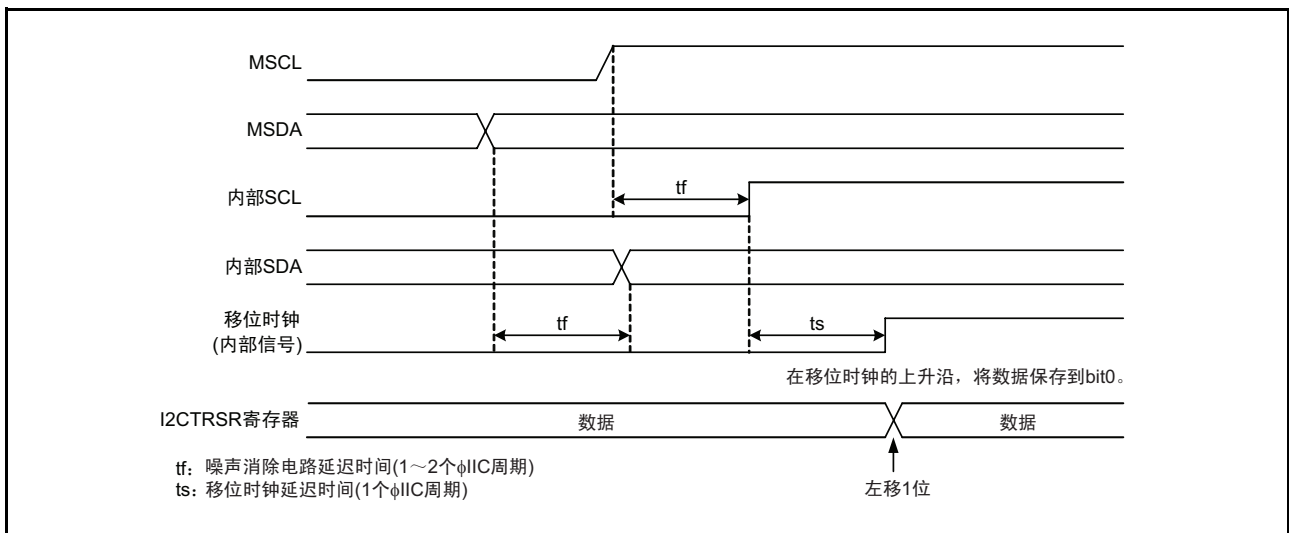


图 24.3 将接收数据保存到 I2CTRSR 寄存器的时序

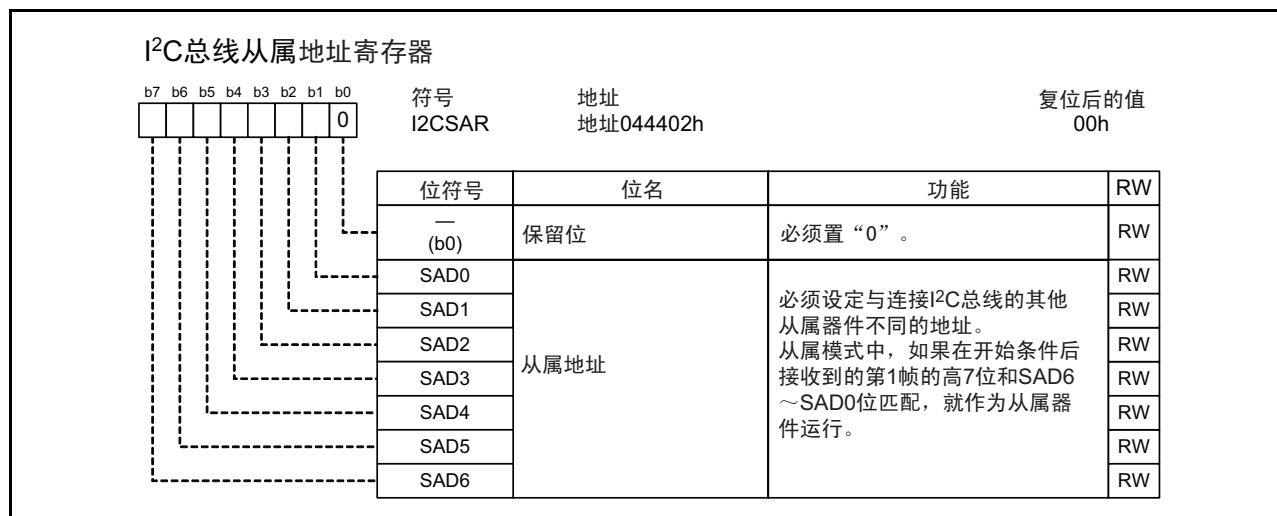
24.1.2 I<sup>2</sup>C 总线从属地址寄存器 (I2CSAR)

图 24.4 I2CSAR 寄存器

I2CSAR 寄存器是作为从属器件自动识别时保存从属地址的寄存器。如果写到 I2CSAR 寄存器的从属地址与接收到的地址匹配，就作为从属器件运行。

## 24.1.2.1 SAD6 ~ SAD0 位

SAD6 ~ SAD0 位是保存从属地址的位，由 7 位构成。在寻址格式时，比较接收到的 7 位地址数据与 SAD6 ~ SAD0 位的内容，如果检测到匹配，就作为从属器件运行。

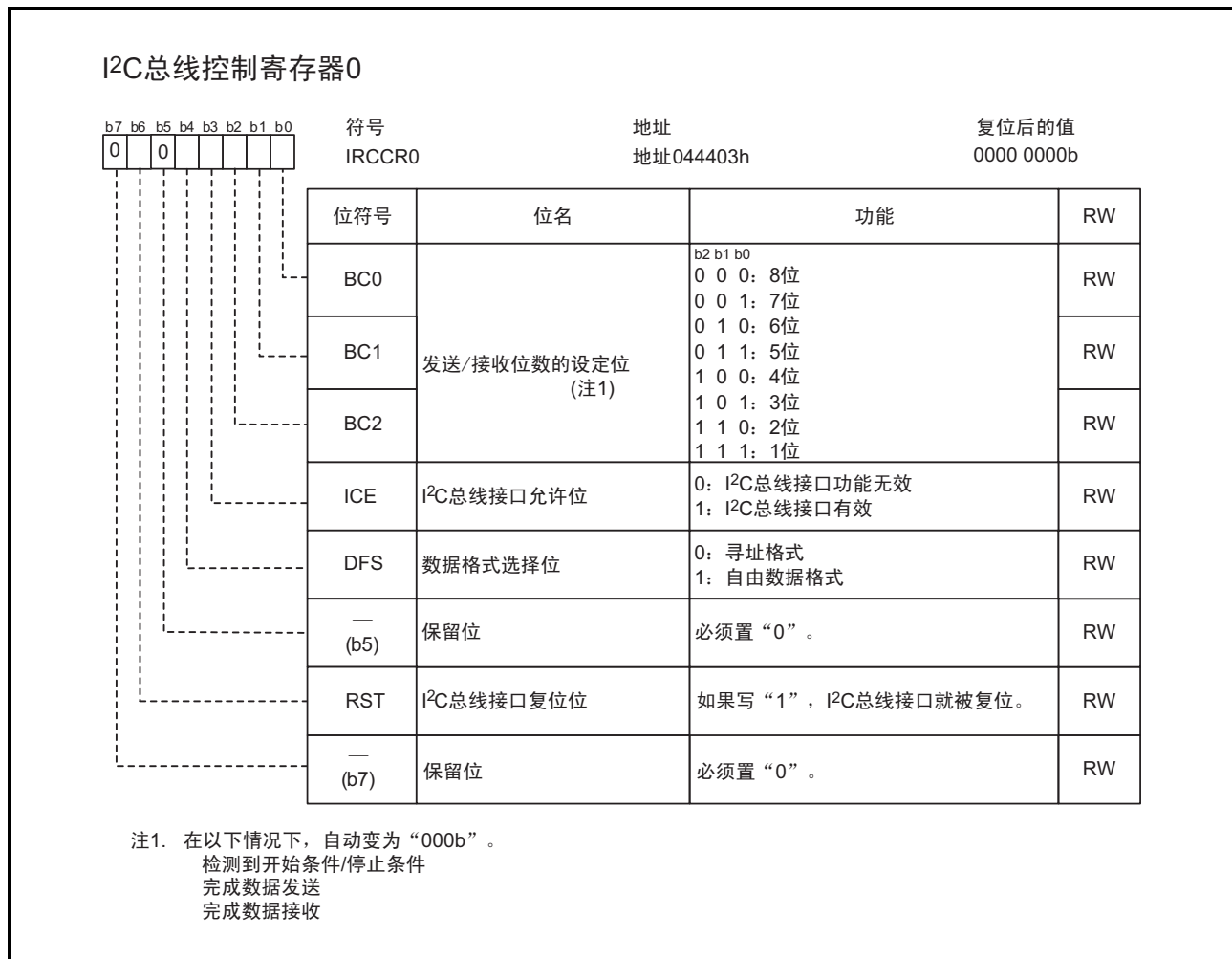
24.1.3 I<sup>2</sup>C 总线控制寄存器 0 (I2CCR0)

图 24.5 I2CCR0 寄存器

I2CCR0 寄存器是控制数据通信格式的寄存器。

## 24.1.3.1 BC2 ~ BC0 位

BC2 ~ BC0 位是设定下一个发送 / 接收数据位数的位。在完成发送 / 接收由这些位指定的位数的数据 (I2CCCR 寄存器的 ACKCLK 位为“1”时，是包括 ACK 时钟在内的位数) 后，产生 I<sup>2</sup>C 总线接口中断请求，并且 BC2 ~ BC0 位恢复为“000b”。另外，即使检测到开始条件，BC2 ~ BC0 位还是为“000b”。与 BC2 ~ BC0 位的设定值无关，以 8 位发送 / 接收地址数据。

### 24.1.3.2 ICE 位

ICE 位是允许使用 I<sup>2</sup>C 总线接口的位。如果将 ICE 位置“1”，就能使用 I<sup>2</sup>C 总线接口。如果将 ICE 位置“0”，MSDA 引脚和 MSCL 引脚就被固定为“H”电平（P7\_0S、P7\_1S 寄存器的 NOD 位为“1”时，为高阻抗），并且不能使用 I<sup>2</sup>C 总线接口。

如果将 ICE 位置“0”，就变为以下的状态。

- I2CSR 寄存器的 ADZ、AAS、AL、BBSY、TRS、MST 位为“0”，IRF 位为“1”。
- 禁止写 I2CTRSR 寄存器。
- I<sup>2</sup>C 总线系统时钟（ $\phi$ IIC）停止，内部计数器和标志复位。
- I2CCR2 寄存器的 TOF 位为“0”（未检测到超时）。

### 24.1.3.3 DFS 位

DFS 位是允许自动识别从属地址的位。该位为“0”时，选择寻址格式，并且自动识别从属地址数据。比较保存到 I2CSAR 寄存器的从属地址和接收到的地址，仅在匹配或者仅接收到全呼时，进行数据的接收处理。如果将 DFS 位置“1”，就为自由数据格式，由于不识别从属地址，因此进行所有数据的接收处理。

### 24.1.3.4 RST 位

RST 位是在发生通信异常时复位 I<sup>2</sup>C 总线接口电路的位。当 I2CCR0 寄存器的 ICE 位为“1”（允许使用 I<sup>2</sup>C 总线接口电路）时，如果给 RST 位写“1”（复位），I<sup>2</sup>C 总线接口电路就变为以下的状态。

- I2CSR 寄存器的 ADZ、AAS、AL、BBSY、TRS、MST 位为“0”，IRF 位为“1”。
- I2CCR2 寄存器的 TOF 位为“0”（未检测到超时）。
- 内部计数器和标志复位。

如果给 RST 位写“1”，最多在 2.5 个  $\phi$ IIC 时钟周期后，完成多主控 I<sup>2</sup>C 总线接口电路的复位处理，并且 RST 位自动变为“0”。

I<sup>2</sup>C 总线接口电路的复位时序如图 24.6 所示。

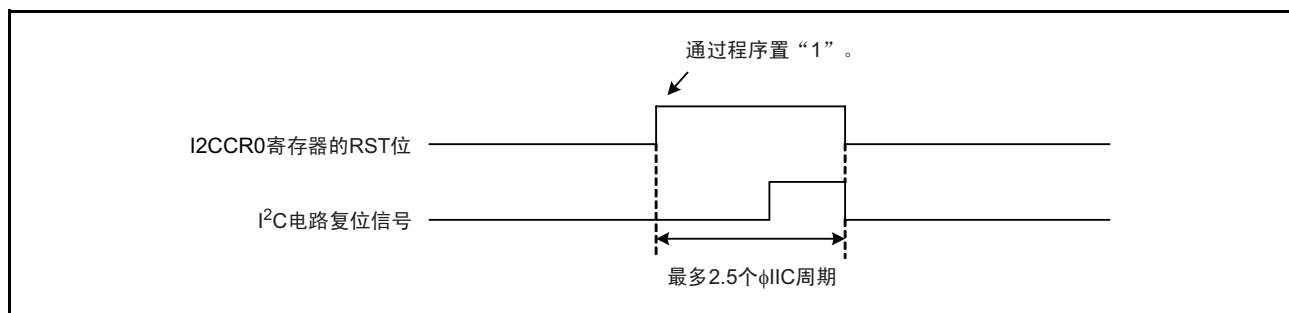


图 24.6 I<sup>2</sup>C 总线接口电路的复位时序

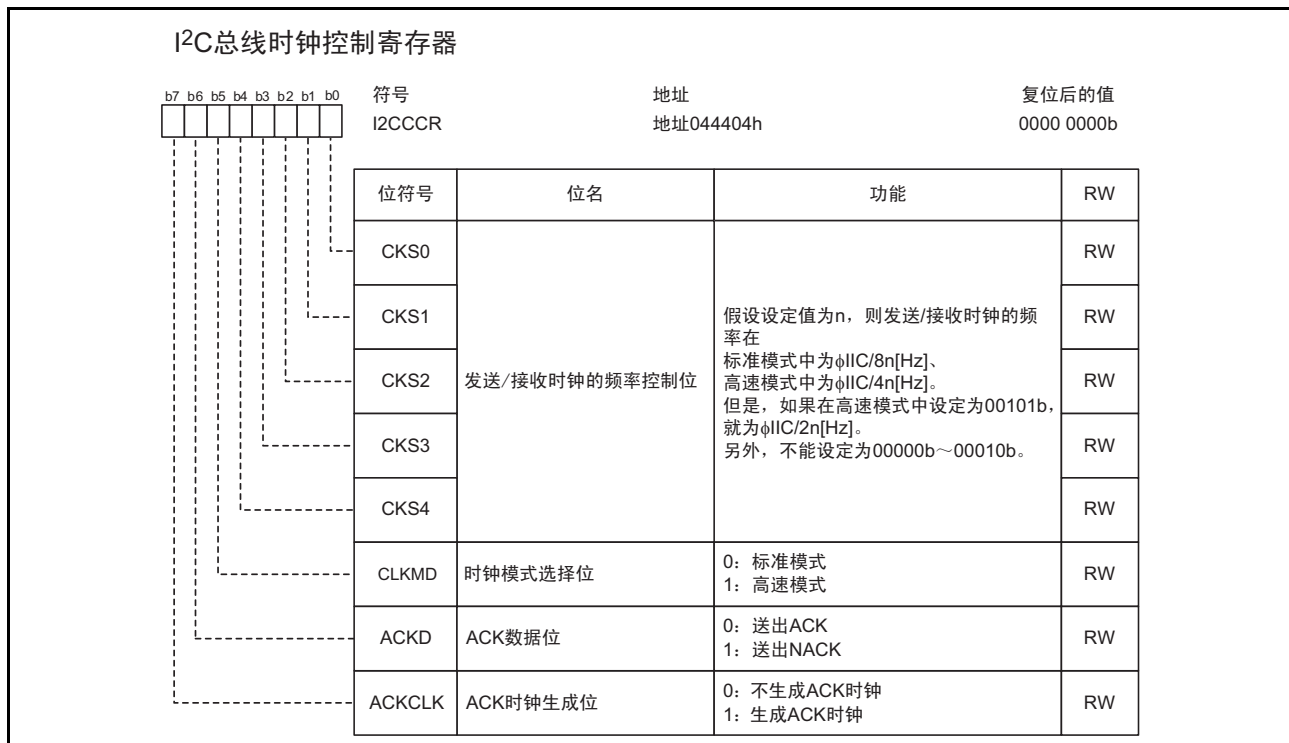
24.1.4 I<sup>2</sup>C 总线时钟控制寄存器 (I2CCCR)

图 24.7 I2CCCR 寄存器

I2CCCR 寄存器是控制 ACK、设定 SCL 模式和 SCL 时钟频率的寄存器。在发送 / 接收数据的过程中, 不能改写 I2CCCR 寄存器的 ACKD 位以外的位。

## 24.1.4.1 CKS4 ~ CKS0 位

CKS4 ~ CKS0 位是设定 SCL 时钟频率的位。假设 CKS4 ~ CKS0 位的设定值为 n (有效值 3 ~ 31), 则 SCL 时钟频率为表 24.3 中所示的值。在发送 / 接收的过程中, 不能改写 CKS4 ~ CKS0 位。

表 24.3 I2CCCR 寄存器的设定值与 SCL 时钟频率

CKS4 ~ CKS0 位的设定值 (n)	SCL 时钟频率 ( $\phi IIC=4MHz$ 的情况) (注 1)	
	标准模式	高速模式
0 ~ 2	禁止设定 (注 2)	禁止设定 (注 2)
3	禁止设定 (注 3)	333kHz ( $\phi IIC/4n$ )
4	禁止设定 (注 3)	250kHz ( $\phi IIC/4n$ )
5	100kHz ( $\phi IIC/8n$ )	400kHz ( $\phi IIC/2n$ ) (注 4)
6 ~ 31	83 ~ 16kHz ( $\phi IIC/8n$ )	166 ~ 32kHz ( $\phi IIC/4n$ )

注 1. 设定 CKS 位的值, 使 SCL 时钟频率在标准模式中小于等于 100kHz, 在高速模式中小于等于 400kHz。在 SCL 时钟的“H”电平期间, 标准模式中有  $+2 \sim -4\phi IIC$  的波动, 高速模式中有  $+2 \sim -2\phi IIC$  的波动。负值波动时, 由于缩短的“H”电平期间就是增长的“L”电平期间, 所以频率不变。

注 2. 与  $\phi IIC$  的频率无关, 不能设定为 0 ~ 2。

注 3. 当  $\phi IIC$  大于等于 4MHz 时, 由于 SCL 时钟频率在规格范围以外, 所以不能设定。

注 4. SCL 时钟的占空比通常为 50%, 但是仅在高速模式且 CKS 的值为“5”时, 占空比为 35 ~ 45%。

#### 24.1.4.2 CLKMD 位

CLKMD 位是选择 SCL 模式的位。如果将该位设定为“0”，就为标准模式，设定为“1”，就为高速模式。在高速模式 I<sup>2</sup>C 总线规格（最高 400k bit/s）下使用时，必须将  $\phi_{IIC}$  设定为大于等于 4MHz。

#### 24.1.4.3 ACKD 位

ACKD 位是设定产生 ACK 时钟时的 MSDA 引脚状态的位。如果将 ACKD 位设定为“0”，产生 ACK 时钟时的 MSDA 引脚就变为“L”电平（ACK 应答）；如果设定为“1”，产生 ACK 时钟时的 MSDA 引脚就保持“H”电平的狀態。

ACK 时钟时的 MSDA 引脚电平如表 24.4 所示。

表 24.4 ACK 时钟时的 MSDA 引脚电平

接收内容	DFS 位	ACKD 位	从属地址	MSDA 引脚电平
从属地址	0	0	匹配	“L”电平（ACK）
			不匹配	“H”电平（NACK）
	1	1	—	“H”电平（NACK）
			0	—
数据	—	0	—	“L”电平（ACK）
		1	—	“H”电平（NACK）

#### 24.1.4.4 ACKCLK 位

ACKCLK 位是设定有无 ACK 应答的位。如果将 ACKCLK 位设定为“1”（生成 ACK 时钟），就在发送 / 接收 1 字节的数据后，生成 ACK 时钟。如果将 ACKCLK 位设定为“0”（不生成 ACK 时钟），在发送 / 接收数据后，不生成 ACK 时钟。此时，在发送 / 接收数据的最后一个时钟的下降沿，I<sup>2</sup>CIC 寄存器的 IR 位变为“1”（有 I<sup>2</sup>C 总线接口中断请求）。

### 24.1.5 I<sup>2</sup>C 总线开始条件 / 停止条件的控制寄存器 (I2CSSCR)

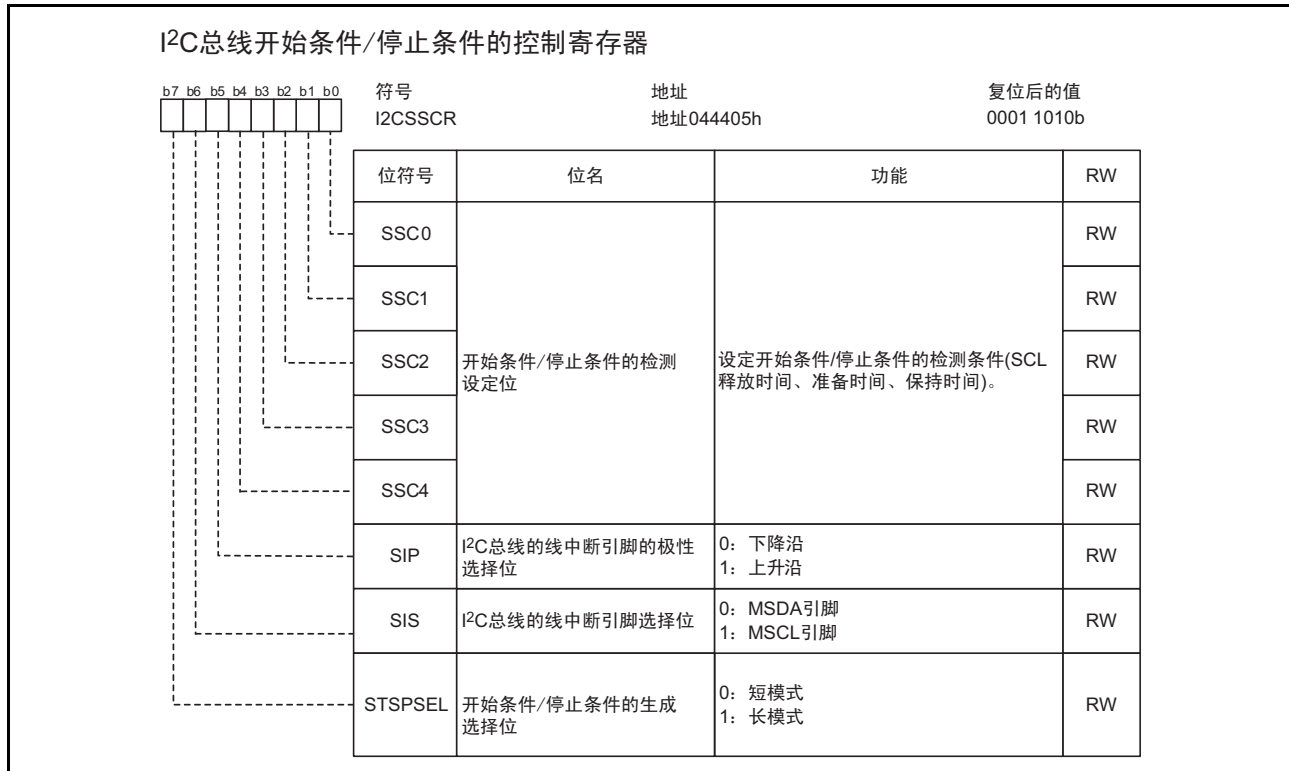


图 24.8 I2CSSCR 寄存器

I2CSSCR 寄存器是控制检测或者生成开始条件 / 停止条件的位。

#### 24.1.5.1 SSC4 ~ SSC0 位

SSC4 ~ SSC0 位是选择标准模式中开始条件 / 停止条件的检测条件 (SCL 释放时间、准备时间、保持时间) 的位。该条件以 I<sup>2</sup>C 总线系统时钟 ( $\phi_{IIC}$ ) 为基准, 所以根据 XIN 频率、I<sup>2</sup>C 总线系统时钟选择位 (I2CCR2 ~ I2CCR1 寄存器的 ICK4 ~ ICK0 位) 变化。不能给 SSC4 ~ SSC0 位设定奇数和 “00000b”。将 I2CCR0 寄存器的 ICE 位设定为 “1” 后, 立即开始进行开始条件 / 停止条件的检测。SSC4 ~ SSC0 位的推荐设定值如表 24.11 所示。

#### 24.1.5.2 SIP 位

SIP 位是选择产生 I<sup>2</sup>C 总线的线中断 (通过 MSCL、MSDA 引脚的输入信号产生) 时的边沿检测极性的位。如果给 SIP 位设定 “0”, 就检测下降沿, 设定 “1”, 就检测上升沿。

#### 24.1.5.3 SIS 位

SIS 位是选择成为 I<sup>2</sup>C 总线的线中断发生源的输入信号的位。如果给 SIS 位设定 “0”, MSDA 引脚的输入信号就成为中断发生源, 设定 “1”, MSCL 引脚的输入信号就成为中断发生源。

#### 24.1.5.4 STSPSEL 位

STSPSEL 位是选择生成开始条件 / 停止条件时的准备 / 保持时间的位。如果给 STSPSEL 位设定 “0”, 就为短模式, 设定 “1”, 就为长模式。如果  $\phi_{IIC}$  的频率超过 4MHz, 就必须将 STSPSEL 位置 “1” (长模式)。开始条件的生成时序如图 24.16 所示, 生成开始条件 / 停止条件时的准备 / 保持时间如表 24.9 所示。

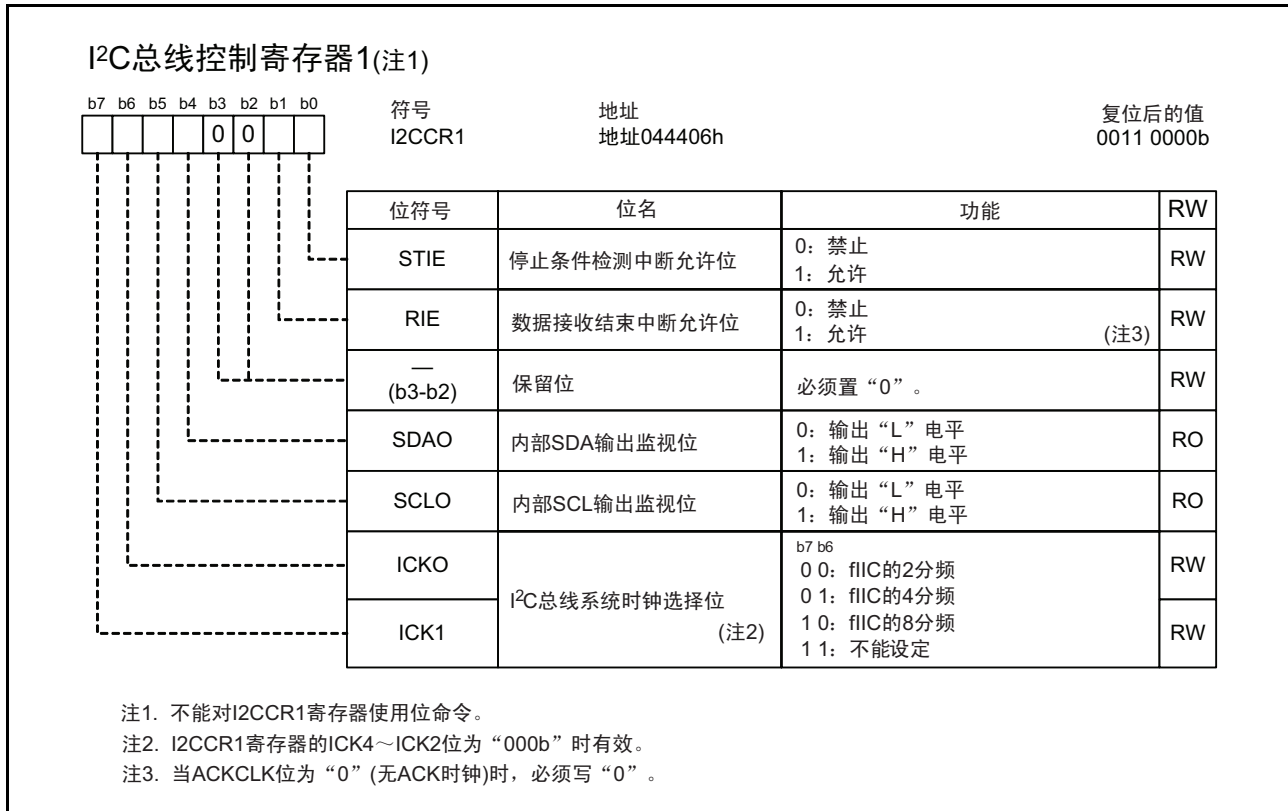
24.1.6 I<sup>2</sup>C 总线控制寄存器 1 (I2CCR1)

图 24.9 I2CCR1 寄存器

I2CCR1 寄存器是进行 I<sup>2</sup>C 总线接口电路的各种控制的寄存器。

## 24.1.6.1 STIE 位

STIE 位是允许在检测到停止条件时发生中断的位。如果给 STIE 位设定“1”，就在检测到停止条件时发生 I<sup>2</sup>C 总线接口中断，并且 I2CCR2 寄存器的 STOP 位变为“1”（有停止条件检测中断请求），I2CIC 寄存器的 IR 位变为“1”（有 I<sup>2</sup>C 总线接口中断请求）。

## 24.1.6.2 RIE 位

RIE 位是在 I2CCCR 寄存器的 ACKCLK 位为“1”（有 ACK 时钟）的情况下，允许在接收到数据的最后一位时发生中断的位。如果将 RIE 位置“1”，就在接收数据的最后一位（第 8 个时钟的 SCL 下降沿）发生 I<sup>2</sup>C 总线接口中断。

与 RIE 位的设定值无关，在 ACK 位（第 9 个时钟的 SCL 下降沿）发生 I<sup>2</sup>C 总线接口中断，因此，在 RIE 位为“1”时，每个数据发生 2 次 I<sup>2</sup>C 总线接口中断。通过读 RIE 位可判断发生的中断是由哪个中断源所引起的。RIE 位的读取值表示内部 WAIT 标志的状态，为“1”时，可判断是由数据的最后一位为中断源发生的中断，为“0”时，可判断是由 ACK 位为中断源发生的中断。

当 I2CCCR 寄存器的 ACKCLK 位为“0”（无 ACK 时钟）时，必须将 RIE 位置“0”。在发送数据和接收从属地址时，与 RIE 位的设定值无关，内部 WAIT 标志为“0”，并且仅在 ACK 位（第 9 个时钟的 SCL 下降沿）发生 I<sup>2</sup>C 总线接口中断。



表 24.5 接收数据时的 I<sup>2</sup>C 总线中断请求发生时序与重新开始发送 / 接收的方法

I <sup>2</sup> C 总线接口中断的发生时序	内部 WAIT 标志	重新开始发送 / 接收的方法
数据最后位 (第 8 个时钟)	1	写 I2CCCR 寄存器的 ACKD 位
ACK 位 (第 9 个时钟)	0	写 I2CTRSR 寄存器。

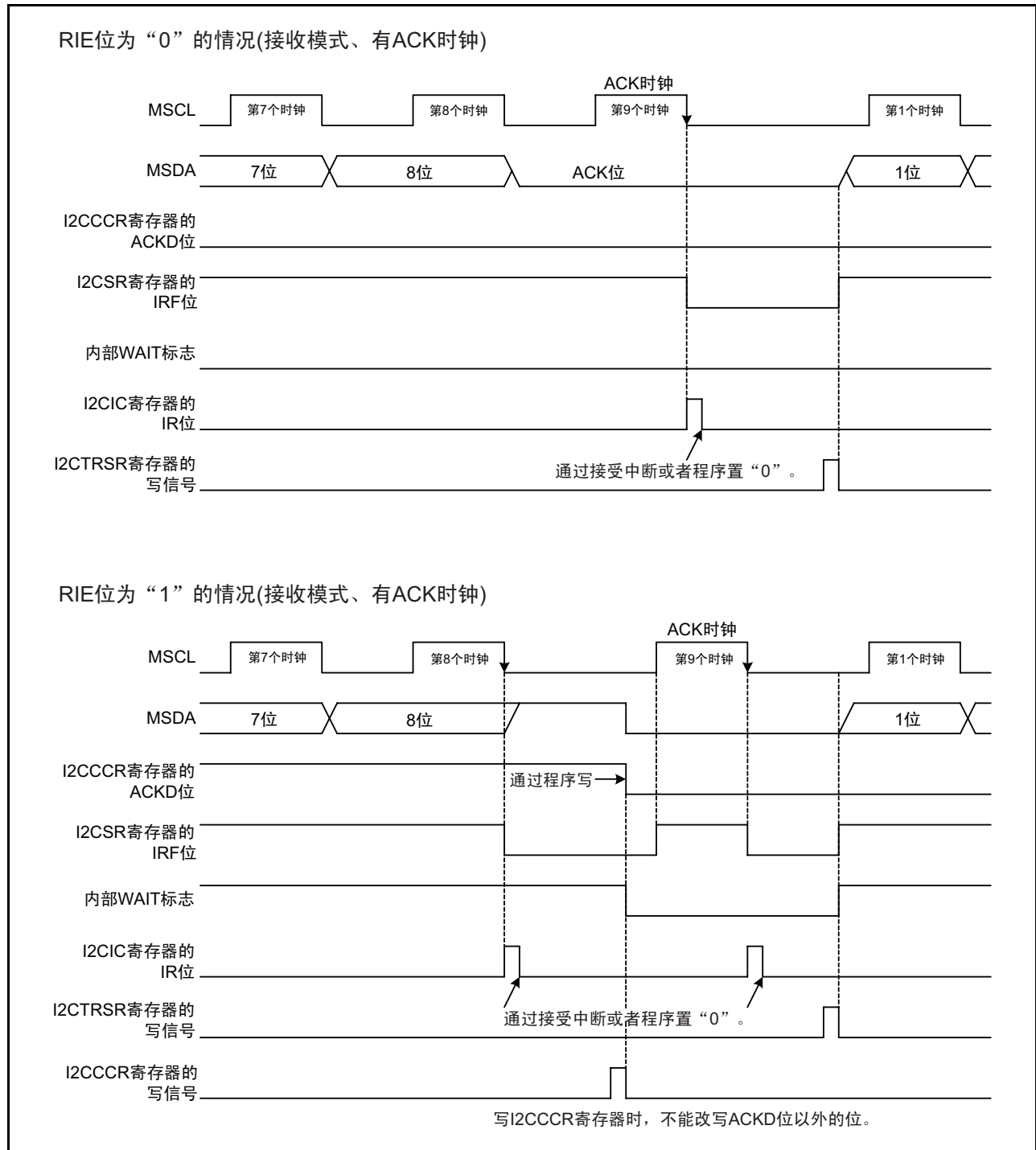


图 24.10 接收数据时的中断请求发生时序

### 24.1.6.3 SDAO/SCLO 位

SDAO/SCLO 位是分别监视内部 SDA 输出信号、内部 SCL 输出信号的逻辑值的只读位。只能写“0”。内部 SDA 输出信号和内部 SCL 输出信号是受外部器件影响前的输出电平，不表示 MSDA 引脚和 MSCL 引脚的状态。

### 24.1.6.4 ICK1 ~ ICK0 位

ICK1 ~ ICK0 位是选择 I<sup>2</sup>C 总线系统时钟 ( $\phi_{IIC}$ ) 频率的位。在 I2CCR2 寄存器的 ICK4 ~ ICK2 位为“000b”时有效。在 I2CCR0 寄存器的 ICE 位为“0” (I<sup>2</sup>C 总线接口功能无效) 时改写这些位。可选择 fIIC 的 2 分频、4 分频、8 分频作为 I<sup>2</sup>C 总线系统时钟，还可通过 I2CCR2 寄存器的 ICK4 ~ ICK2 位选择 fIIC 的 2.5 分频、3 分频、5 分频、6 分频作为 I<sup>2</sup>C 总线系统时钟，此时，ICK1 ~ ICK0 位无效。

表 24.6 I<sup>2</sup>C 总线系统时钟 ( $\phi_{IIC}$ ) 选择位

I2CCR2 寄存器			I2CCR1 寄存器		$\phi_{IIC}$
ICK4 位	ICK3 位	ICK2 位	ICK1 位	ICK0 位	
0	0	0	0	0	fIIC 的 2 分频
			0	1	fIIC 的 4 分频
			1	0	fIIC 的 8 分频
0	0	1	0	0	fIIC 的 2.5 分频
0	1	0	0	0	fIIC 的 3 分频
0	1	1	0	0	fIIC 的 5 分频
1	0	0	0	0	fIIC 的 6 分频

不能设定上述以外的组合。

24.1.7 I<sup>2</sup>C 总线控制寄存器 2 (I2CCR2)

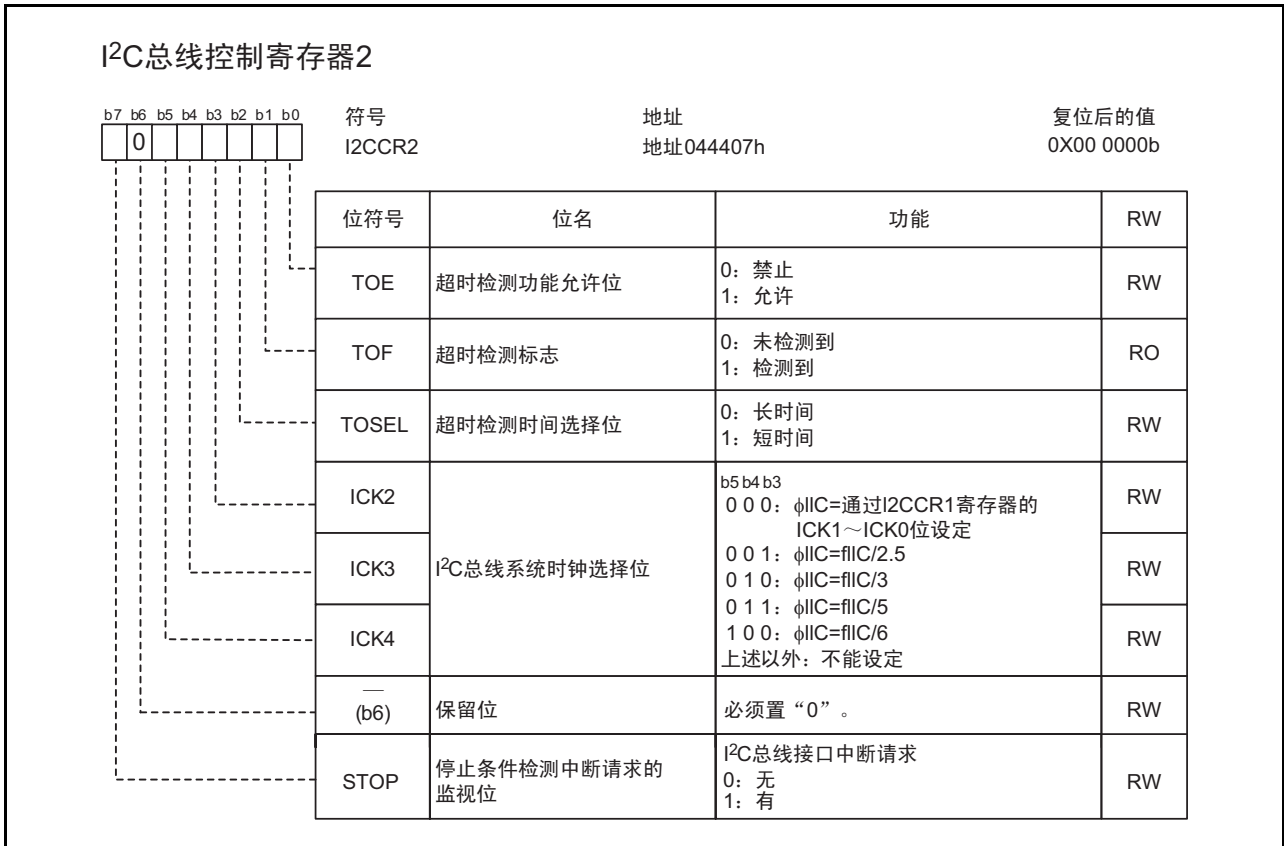


图 24.11 I2CCR2 寄存器

I2CCR2 寄存器是控制通信异常检测的位。如果在发送 / 接收过程中, SCL 时钟停止振荡, 各器件就在保持通信的状态下停止运行。为了回避此状况, 设置了发生 I<sup>2</sup>C 总线接口中断的功能, 即, 如果在发送 / 接收过程中, SCL 时钟保持“H”电平的状态并且超过了规定时间, 就发生 I<sup>2</sup>C 总线接口中断。

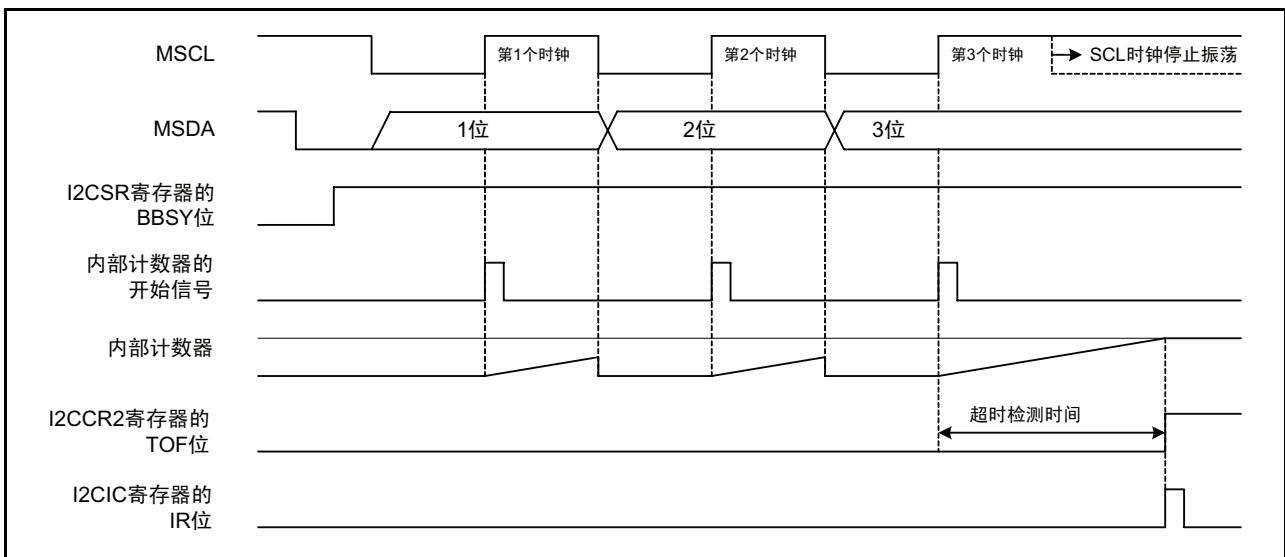


图 24.12 超时检测的时序图

### 24.1.7.1 TOE 位

TOE 位是允许超时检测功能的位。将 TOE 位设定为“1”时，允许超时检测功能，并且如果 I2CSR 寄存器的 BBSY 位保持“1”（总线忙），SCL 时钟持续“H”电平的状态并且超过了规定时间，就发生 I<sup>2</sup>C 总线接口中断。

通过以  $\phi_{IIC}$  作为计数源的内部计数器计算超时检测的时间，并且可通过 TOSEL 位选择长时间或者短时间。详细内容请参照“24.1.7.3 TOSEL 位”。

如果检测到超时，就必须将 I2CCR0 寄存器的 ICE 位置“0”（I<sup>2</sup>C 总线接口功能无效），并且将 I<sup>2</sup>C 总线接口电路初始化。

### 24.1.7.2 TOF 位

TOF 标志是表示超时检测状态的标志。在 TOE 位为“1”时有效。如果 TOF 位变为“1”（超时检测），I2CIC 寄存器的 IR 位就同时变为“1”（有 I<sup>2</sup>C 总线接口中断请求）。

### 24.1.7.3 TOSEL 位

TOSEL 位是选择超时检测时间的位。在 TOE 位为“1”（允许超时检测功能）时有效。可选择长时间或者短时间。如果将 TOSEL 位设定为“0”，就选择长时间，并且内部计数器用作 16 位计数器；如果将 TOSEL 位设定为“1”，就选择短时间，并且内部计数器用作 14 位计数器。以 I<sup>2</sup>C 总线系统时钟（ $\phi_{IIC}$ ）作为计数源的内部计数器进行递增计数。超时检测时间如表 24.7 所示。

表 24.7 超时检测时间的例子

$\phi_{IIC}$	长时间 (TOSEL=0)	短时间 (TOSEL=1)
4MHz	16.4ms	4.1ms
2MHz	32.8ms	8.2ms
1MHz	65.6ms	16.4ms

### 24.1.7.4 ICK4 ~ ICK2 位

ICK4 ~ ICK2 位是选择 I<sup>2</sup>C 总线系统时钟（ $\phi_{IIC}$ ）频率的位。在 I2CCR0 寄存器的 ICE 位为“0”（I<sup>2</sup>C 总线接口功能无效）时改写这些位。

可选择  $\phi_{IIC}$  的 2.5 分频、3 分频、5 分频、6 分频作为 I<sup>2</sup>C 总线系统时钟（ $\phi_{IIC}$ ）。如果给 ICK4 ~ ICK2 位设定“000b”，就可通过 I2CCR1 寄存器的 ICK1 ~ ICK0 位选择  $\phi_{IIC}$  的 2 分频、4 分频、8 分频。详细内容请参照表 24.6。

### 24.1.7.5 STOP 位

STOP 位是监视停止条件检测中断的位。如果发生由停止条件检测引起的 I<sup>2</sup>C 总线接口中断，此位就变为“1”。在 I2CCR1 寄存器的 STIE 位为“1”（允许由停止条件检测引起的 I<sup>2</sup>C 总线接口中断）时有效。通过程序写“0”，此位就变为“0”，即使写“1”，此位也不变。

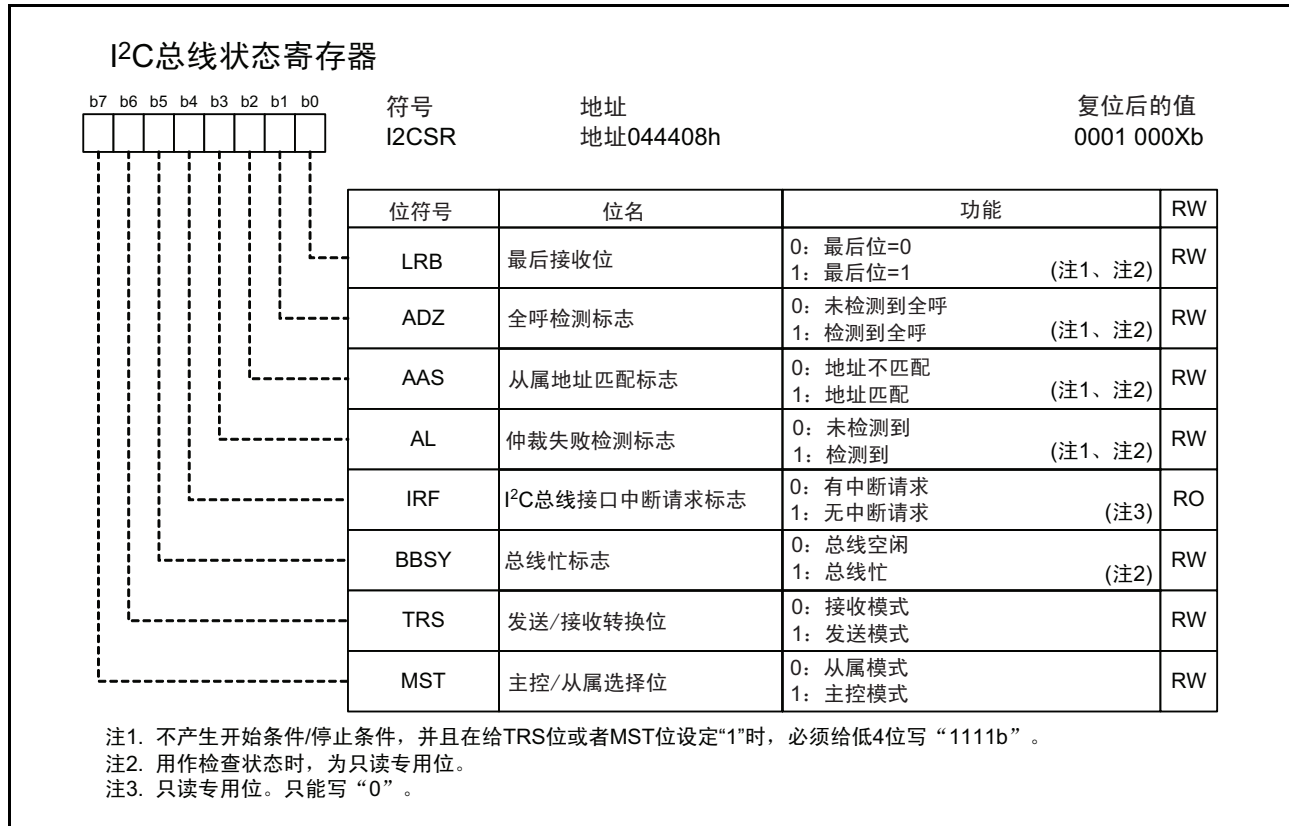
24.1.8 I<sup>2</sup>C 总线状态寄存器 (I2CSR)

图 24.13 I2CSR 寄存器

I2CSR 寄存器是监视 I<sup>2</sup>C 总线接口状态的寄存器。只有在使用如表 24.8 所示的功能时, 写此寄存器。不能给 I2CSR 寄存器写表 24.8 所示的值以外的值。即使给 I2CSR 寄存器写表 24.8 所示的值, I2CSR 寄存器的低 6 位也不变。

表 24.8 写 I2CSR 寄存器时的功能

I2CSR 寄存器的写入值								功能
MST	TRS	BBSY	IRF	AL	AAS	ADZ	LRB	
0	0	X	0	1	1	1	1	选择从属接收模式
0	1							选择从属发送模式
1	0							选择主控接收模式
1	1							选择主控发送模式
1	1	0	0	0	0	0	0	选择主控发送模式, 并且设定为停止条件待机状态。
		1						选择主控发送模式, 并且设定为停止条件待机状态。

## 24.1.8.1 LRB 位

LRB 位保存接收数据的最后位的值。用于确认接收 ACK 应答的结果。在 I2CCCR 寄存器的 ACKCLK 位为“1” (有 ACK 时钟) 时, 如果接收到 ACK 应答, LRB 位就变为“0”, 如果没有接收到 ACK 应答, LRB 位就为“1”。在 I2CCCR 寄存器的 ACKCLK 位为“0” (无 ACK 时钟) 时, LRB 位保存接收数据的最后位的值。如果给 I2CTSR 寄存器写值, LRB 位就变为“0”。

### 24.1.8.2 ADZ 位

ADZ 标志是表示接收到全呼（地址数据全部为“0”）的标志。在从属接收模式中，如果在 I2CCR0 寄存器的 DFS 位为“0”（寻址格式）时接收全呼，ADZ 标志就变为“1”。

ADZ 标志在以下的任何一种情况下变为“0”。

- 检测到开始条件/停止条件时
- 将 I2CCR0 寄存器的 ICE 位设定为“0”（I<sup>2</sup>C 总线接口功能无效）时
- 将 I2CCR0 寄存器的 RST 位设定为“1”（I<sup>2</sup>C 总线接口电路复位）时

### 24.1.8.3 AAS 位

AAS 标志是表示是否与接收到的地址数据匹配的标志。在从属接收模式中，当 I2CCR0 寄存器的 DFS 位为“0”（寻址格式）时，接收到的地址数据与 I2CSAR 寄存器的 SAD6 ~ SAD0 位（从属地址）匹配，或者在全呼时，AAS 标志变为“1”。

AAS 标志在以下的任何一种情况下变为“0”。

- 将数据写到 I2CTRSR 寄存器时
- 将 I2CCR0 寄存器的 ICE 位设定为“0”（I<sup>2</sup>C 总线接口功能无效）时
- 将 I2CCR0 寄存器的 RST 位设定为“1”（I<sup>2</sup>C 总线接口电路复位）时

### 24.1.8.4 AL 位

AL 标志是表示检测仲裁失败的标志。在主控发送时，如果由其他器件使 MSDA 引脚变为“L”电平，AL 标志就变为“1”。此时，I2CSR 寄存器的 TRS 位变为“0”（接收模式），并且在完成发送仲裁失败的数据后，I2CSR 寄存器的 MST 位变为“0”（从属模式）。

AL 标志在以下的任何一种情况下变为“0”。

- 将数据写到 I2CTRSR 寄存器时
- 将 I2CCR0 寄存器的 ICE 位设定为“0”（I<sup>2</sup>C 总线接口功能无效）时
- 将 I2CCR0 寄存器的 RST 位设定为“1”（I<sup>2</sup>C 总线接口电路复位）时

### 24.1.8.5 IRF 位

IRF 位是产生 I<sup>2</sup>C 总线接口中断的请求信号的位。如果产生 I<sup>2</sup>C 总线接口中断源，IRF 位就变为“0”，并且与 IRF 位的下降沿同步，发生 I<sup>2</sup>C 总线接口中断。时序的例子请参照图 24.10。

IRF 位在以下的任何一种情况下变为“0”。

- 在完成发送 1 字节的数据时（包括检测到仲裁失败时）
- 在完成接收 1 字节的数据时
- 从属接收时，在寻址格式检测到从属地址匹配时
- 从属接收时，在寻址格式接收到全呼地址时
- 从属接收时，在自由格式完成接收地址数据时

IRF 位在以下的任何一种情况下变为“1”。

- 将数据写到 I2CTRSR 寄存器时
- 将数据写到 I2CCCR 寄存器时（WIT=1、内部 WAIT 标志=1）
- 将 I2CCR0 寄存器的 ICE 位设定为“0”（I<sup>2</sup>C 总线接口功能无效）时
- 将 I2CCR0 寄存器的 RST 位设定为“1”（I<sup>2</sup>C 总线接口电路复位）时

#### 24.1.8.6 BBSY 位

BBSY 标志是表示 I<sup>2</sup>C 总线使用状况的标志。在检测到开始条件时，BBSY 标志变为“1”，在检测到停止条件时，BBSY 标志变为“0”。BBSY 标志为“0”时，是未使用 I<sup>2</sup>C 总线的状态，因此可生成开始条件。

开始条件 / 停止条件的检测条件取决于 I2CSSCR 寄存器的 SSC4 ~ SSC0 位的设定。

BBSY 位在以下的任何一种情况下变为“0”。

- 在检测到停止条件时
- 将 I2CCR0 寄存器的 ICE 位设定为“0”（I<sup>2</sup>C 总线接口功能无效）时
- 将 I2CCR0 寄存器的 RST 位设定为“1”（I<sup>2</sup>C 总线接口电路复位）时

#### 24.1.8.7 TRS 位

TRS 位是指定数据通信方向的位。如果给 TRS 位设定“0”，就为接收模式，等待来自其他器件的数据。如果给 TRS 位设定“1”，就为发送模式，与 SCL 时钟同步向 SDA 输出地址或者数据。

如果在从属接收模式的寻址格式检测到地址匹配，并且接收到的 R/W 位为“1”（请求数据），TRS 位就自动变为“1”（发送模式）。

TRS 位在以下的任何一种情况下变为“0”。

- 通过程序将“0”写到 TRS 位时
- 在检测到仲裁失败时
- 在检测到停止条件时
- 在开始条件重复防止功能有效时
- 在从属模式中检测到开始条件时
- 在从属模式中检测到 NACK 应答时
- 将 I2CCR0 寄存器的 ICE 位设定为“0”（I<sup>2</sup>C 总线接口功能无效）时
- 将 I2CCR0 寄存器的 RST 位设定为“1”（I<sup>2</sup>C 总线接口电路复位）时

#### 24.1.8.8 MST 位

MST 位是指定主控 / 从属模式的位。将 MST 位置“0”时，为从属模式，并且与主控器件产生的 SCL 时钟同步进行通信。将 MST 位置“1”时，为主控模式，并且在产生 SCL 时钟后进行通信。

MST 位在以下的任何一种情况下变为“0”。

- 通过程序将“0”写到 MST 位时
- 在检测到仲裁失败，并且完成该仲裁失败的 1 字节数据的通信时
- 在检测到停止条件时
- 在检测到开始条件时
- 在开始条件重复防止功能有效时
- 将 I2CCR0 寄存器的 ICE 位设定为“0”（I<sup>2</sup>C 总线接口功能无效）时
- 将 I2CCR0 寄存器的 RST 位设定为“1”（I<sup>2</sup>C 总线接口电路复位）时

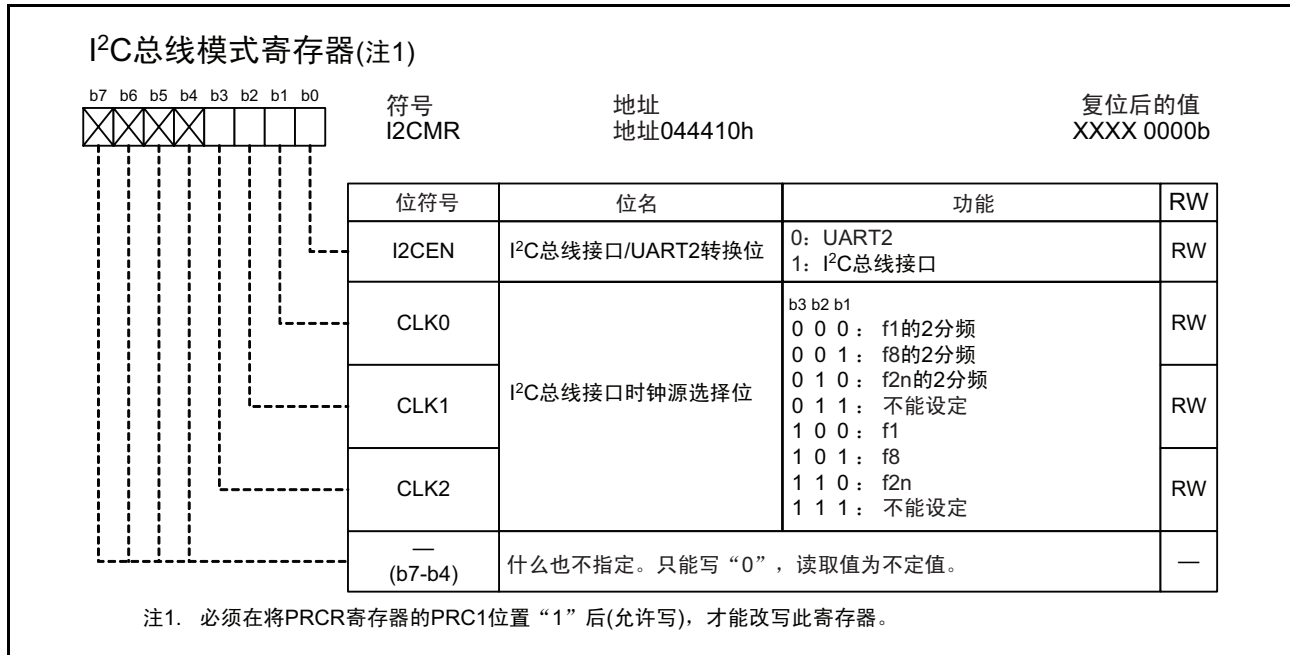
24.1.9 I<sup>2</sup>C 总线模式寄存器 (I2CMR)

图 24.14 I2CMR 寄存器

I2CMR 寄存器是转换 I<sup>2</sup>C 总线接口信号、选择时钟源的寄存器。改写此寄存器时，必须在将 PRCR 寄存器的 PRC1 位置“1”（允许写）后才能进行。

## 24.1.9.1 I2CEN 位

I2CEN 位是转换 UART2 信号和 I<sup>2</sup>C 总线接口信号的位。如果将 I2CEN 位置“1”，就选择 I<sup>2</sup>C 总线接口，并且 MSDA 信号、MSCL 信号、I<sup>2</sup>C 总线接口中断、I<sup>2</sup>C 总线的线中断有效。如果将 I2CEN 位置“0”，MSDA 信号、MSCL 信号、I<sup>2</sup>C 总线接口中断、I<sup>2</sup>C 总线的线中断无效，UART2 信号有效。

## 24.1.9.2 CLK2 ~ CLK0 位

CLK2 ~ CLK0 位是选择 I<sup>2</sup>C 总线接口时钟 (f<sub>IIC</sub>) 的时钟源的位。可选择 f1、f8、f2n、以及 f1、f8、f2n 的 2 分频。在此选择的 I<sup>2</sup>C 总线接口时钟 (f<sub>IIC</sub>) 还可用作 I<sup>2</sup>C 总线系统时钟 (φ<sub>IIC</sub>) 的时钟源。



## 24.2 开始条件的生成方法

在 I2CCR0 寄存器的 ICE 位为“1”（I<sup>2</sup>C 总线接口有效）、I2CSR 寄存器的 BBSY 标志为“0”（总线空闲）的状态下，给 I2CSR 寄存器写“E0h”，就进入开始条件待机状态。接着，给 I2CTRSR 寄存器写从属地址，就生成开始条件。之后，位计数器变为“000b”，并且输出 1 字节的 SCL 时钟和发送从属地址。开始条件的生成步骤如图 24.15 所示。

另外，在生成停止条件后，从 BBSY 标志变为“0”（总线空闲）到 1.5 个  $\phi_{IIC}$  周期期间，不能给 I2CSR 寄存器写值。在生成停止条件后立即生成开始条件时，必须在给 I2CSR 寄存器写“E0h”后，确认 I2CSR 寄存器的 TRS 位、MST 位变为“1”，然后给 I2CTRSR 寄存器写从属地址。

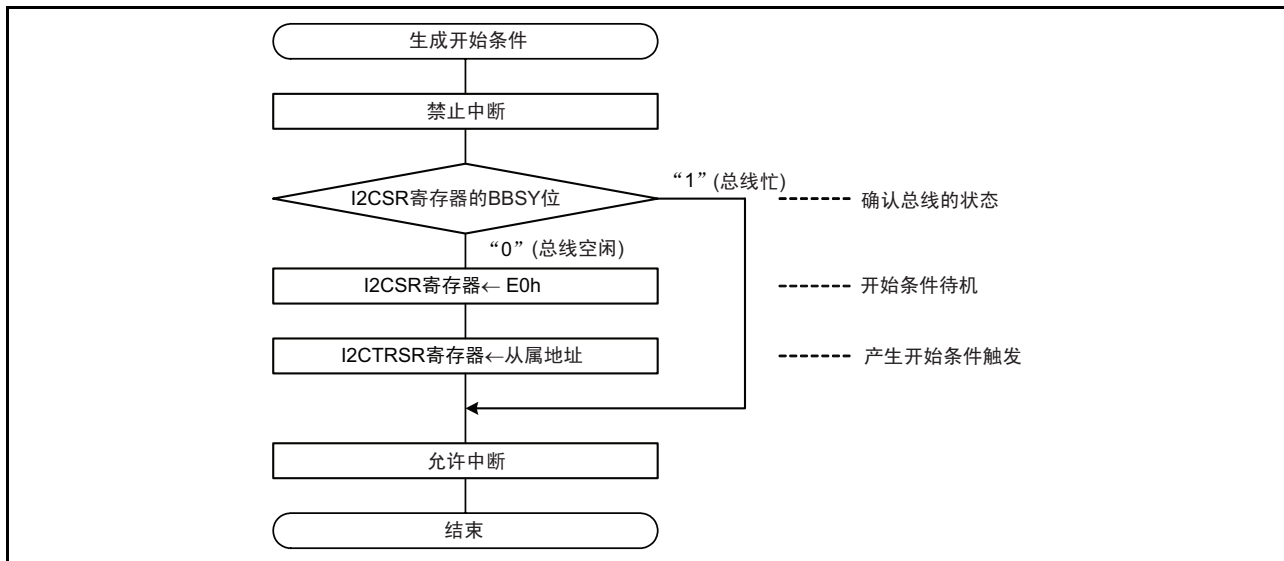


图 24.15 开始条件的生成步骤

标准模式和高速模式中的开始条件的生成时序不同，开始条件的生成时序如图 24.16 所示，生成开始条件 / 停止条件时的准备 / 保持时间如表 24.9 所示。

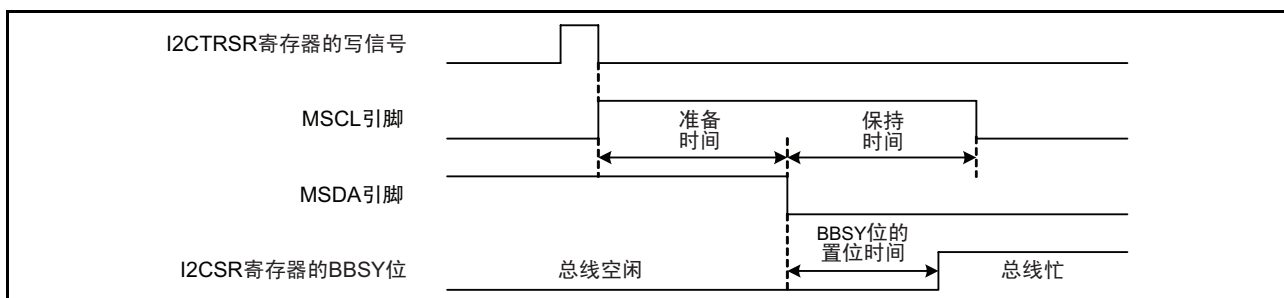


图 24.16 开始条件的生成时序

表 24.9 生成开始条件 / 停止条件时的准备 / 保持时间

项目	SCL 模式	短模式 (STSPSEL=0)	短模式 (STSPSEL=1)
准备时间	标准模式 (CLKMD=0)	5.0 $\mu$ s(20)	13.0 $\mu$ s(52)
	高速模式 (CLKMD=1)	2.5 $\mu$ s(10)	6.5 $\mu$ s(26)
保持时间	标准模式 (CLKMD=0)	5.0 $\mu$ s(20)	13.0 $\mu$ s(52)
	高速模式 (CLKMD=1)	2.5 $\mu$ s(10)	6.5 $\mu$ s(26)

CLKMD: I2CCCR 寄存器的位  
STSPSEL: I2CSSCR 寄存器的位  
( ) 内为  $\phi_{IIC}$  的周期数

### 24.3 停止条件的生成方法

在 I2CCR0 寄存器的 ICE 位为“1”（I<sup>2</sup>C 总线接口有效）的状态下，给 I2CSR 寄存器写“C0h”，就进入停止条件待机状态，并且 MSDA 引脚变为“L”电平。接着，给 I2CTRSR 寄存器写虚拟数据，就生成停止条件。停止条件的生成步骤如图 24.17 所示。

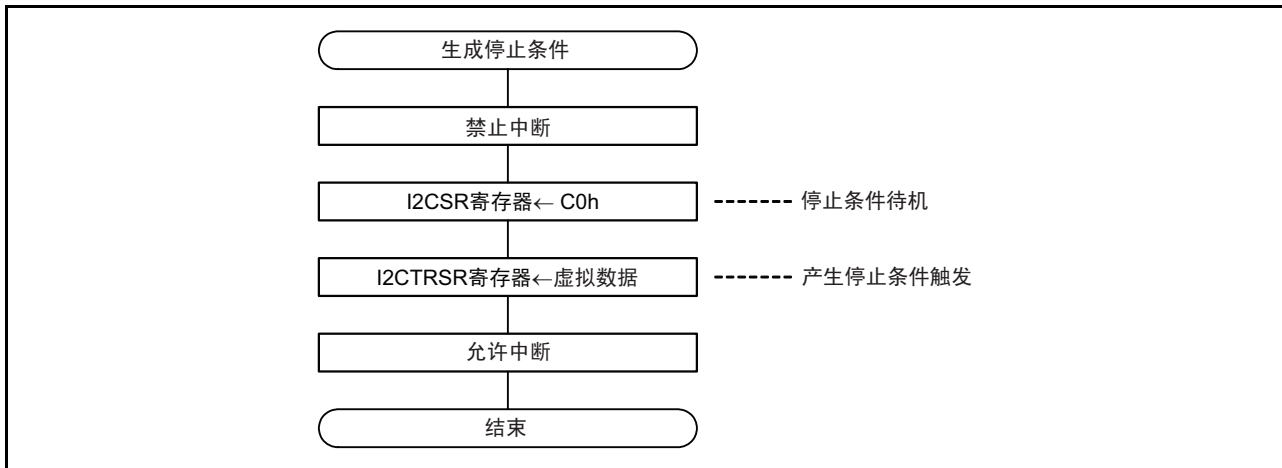


图 24.17 停止条件的生成步骤

标准模式和高速模式中的停止条件的生成时序不同，停止条件的生成时序如图 24.18 所示，生成开始条件 / 停止条件时的准备 / 保持时间如表 24.9 所示。

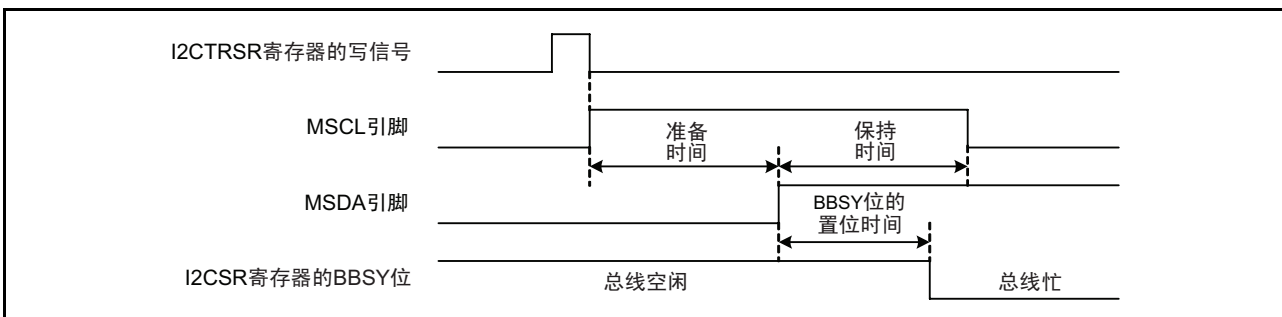


图 24.18 停止条件的生成时序

另外，从指示生成停止条件到 I2CSR 寄存器的 BBSY 标志变为“0”（总线空闲）的期间，不能写 I2CSR 寄存器和 I2CTRSR 寄存器。可能有无法正常生成停止条件的情况。

从指示生成停止条件，到 MSCL 引脚的电平变为“H”电平，再到 I2CSR 寄存器的 BBSY 标志变为“0”（总线空闲）的期间，如果 MSCL 引脚的输入信号变为“L”电平，内部 SCL 输出就为“L”电平。此时，如果执行生成停止条件、给 I2CCR0 寄存器的 ICE 位写“0”（I<sup>2</sup>C 总线接口有效）、给 RST 位写“1”（I<sup>2</sup>C 总线接口电路复位）中的任意一个，就停止（释放）MSCL 引脚的“L”电平输出。

### 24.4 开始条件重复防止功能

在生成开始条件时，先通过 I2CSR 寄存器的 BBSY 标志确认总线没有被使用后才能生成。在确认 BBSY 标志后到生成开始条件期间，可能有其他主控器件生成开始条件，并且 BBSY 标志变为“1”的可能性。此时，开始条件重复防止功能运行，中止开始条件的生成。

开始条件重复防止功能如下运行。

- 禁止设定开始条件待机（解除待机状态）
- 禁止写 I2CTRSR 寄存器（禁止产生开始条件触发）
- I2CSR 寄存器的 MST、TRS 位变为“0”（转移到从属接收模式）
- I2CSR 寄存器的 AL 标志变为“1”（检测到仲裁失败）

开始条件重复防止功能的运行例子如图 24.19 所示。

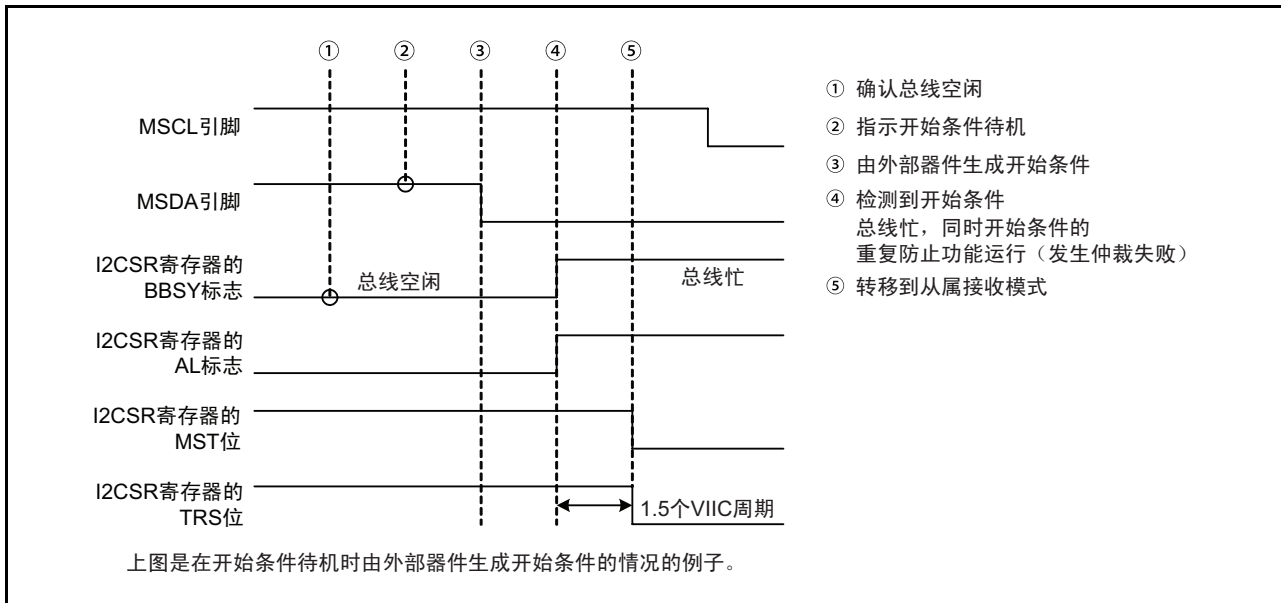


图 24.19 开始条件重复防止功能的运行例子

从开始条件的 SDA 下降沿开始，到从属地址的接收结束为止是开始条件重复防止功能的有效期间。即，如果在此期间写 I2CSR 寄存器和 I2CTRSR 寄存器，就进行前述的运行。开始条件重复防止功能的有效期间如图 24.20 所示。

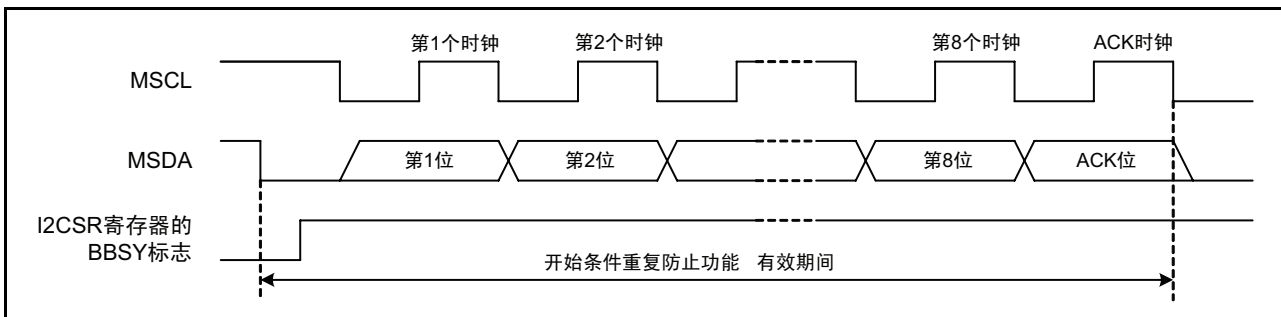


图 24.20 开始条件重复防止功能的有效期间

### 24.5 开始条件 / 停止条件的检测

开始条件的检测、停止条件的检测以及开始条件 / 停止条件的检测条件分别如图 24.21、图 24.22 和表 24.10 所示。通过 I2CSSCR 寄存器的 SSC4 ~ SSC0 位设定开始条件 / 停止条件的检测条件，并且只有在 MSCL 引脚和 MSDA 引脚的输入信号满足表 24.10 的 MSCL 释放时间、准备时间和保持时间这 3 个条件的情况下，才能检测开始条件 / 停止条件。

如果检测到开始条件，IRCSR 寄存器的 BBSY 标志就变为“1”，如果检测到停止条件，IRCSR 寄存器的 BBSY 标志就变为“0”。标准模式和高速模式中的 BBSY 标志的置位 / 复位时序不同，详细内容请参照表 24.11 的 BBSY 标志的置位 / 复位时间。另外，标准模式中的 SSC4 ~ SSC0 位的推荐值如表 24.11 所示。

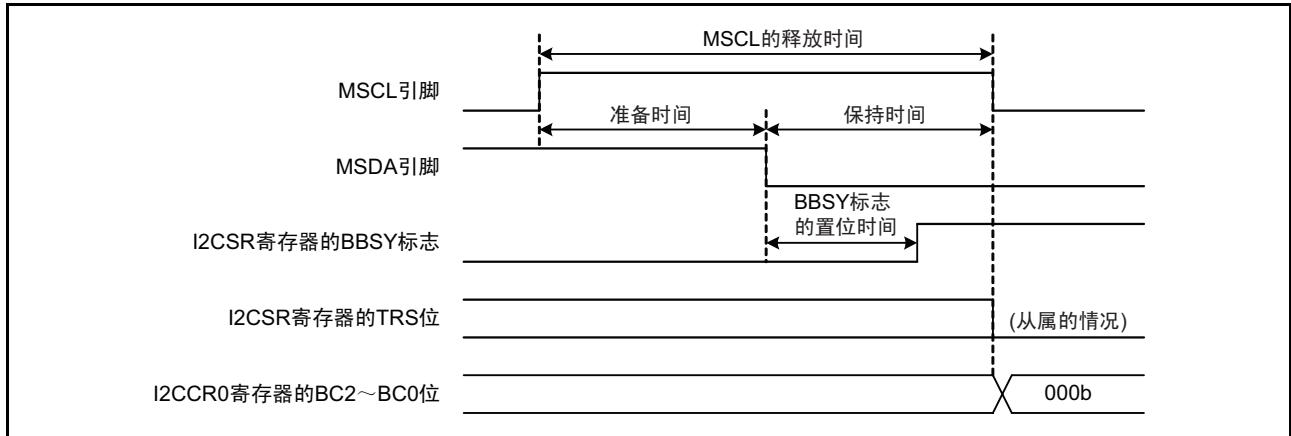


图 24.21 开始条件的检测

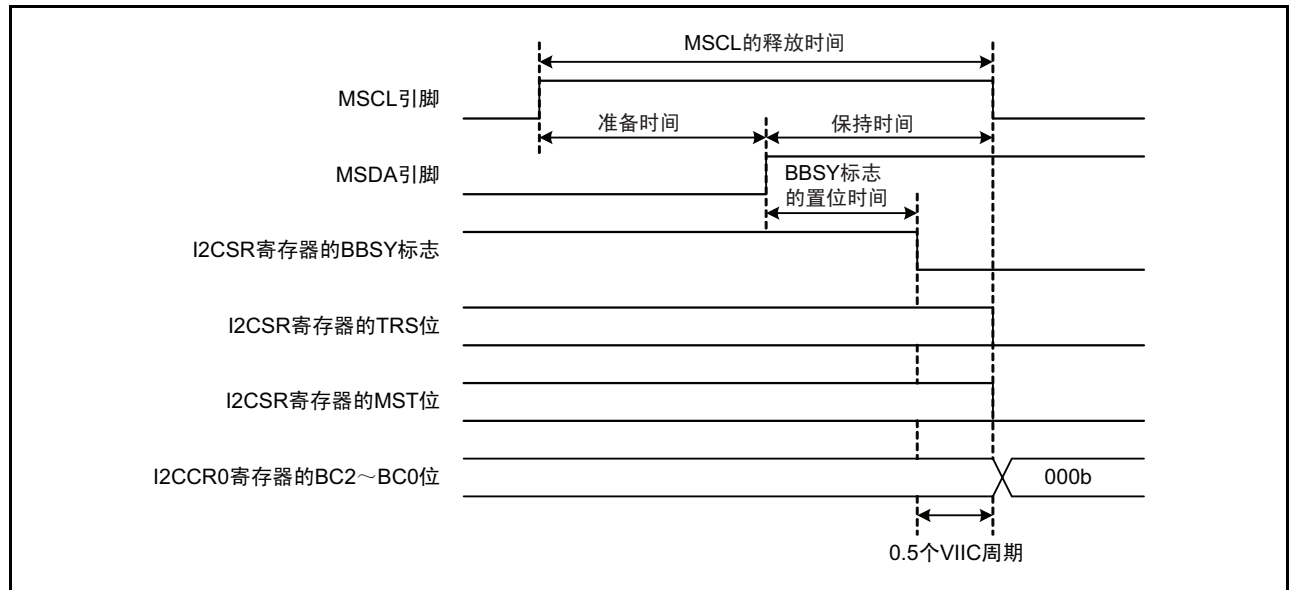


图 24.22 停止条件的检测

表 24.10 开始条件 / 停止条件的检测条件

	标准模式	高速模式
MSCL 引脚的释放时间	SSC 值 +1 个周期 (6.25μs)	4 个周期 (1.0μs)
准备时间	$\frac{\text{SSC值}}{2} + 1$ 个周期 < 4.0μs(3.25μs)	2 个周期 (0.5μs)
保持时间	$\frac{\text{SSC值}}{2}$ 个周期 < 4.0μs(3.0μs)	2 个周期 (0.5μs)
BBSY 标志的置位 / 复位时间	$\frac{\text{SSC值}-1}{2} + 2$ 个周期(3.375μs)	3.5 个周期 (0.875μs)

单位: φIIC 的周期数

SSC 值: I2CSSCR 寄存器的 SSC4 ~ SSC0 位的值, 并且禁止设定 0 以及奇数。

( ) 内为 φIIC=4MHz 时, I2CSSCR 寄存器 =18h 的时间例子。

表 24.11 标准模式中的 SSC4 ~ SSC0 位的推荐值

φIIC	SSC 值 (推荐值)	开始条件 / 停止条件的检测条件			BBSY 标志的 置位 / 复位时间
		MSCL 引脚的释放时间	准备时间	保持时间	
5MHz	30	6.2μs(31)	3.2μs(16)	3.0μs(15)	4.125μs(16.5)
4MHz	26	6.75μs(27)	3.5μs(14)	3.25μs(13)	3.625μs(14.5)
	24	6.25μs(25)	3.25μs(13)	3.0μs(12)	3.375μs(13.5)
2MHz	12	6.5μs(13)	3.5μs(7)	3.0μs(6)	3.75μs(7.5)
	10	5.5μs(11)	3.0μs(6)	2.5μs(5)	3.25μs(6.5)
1MHz	4	5.0μs(5)	3.0μs(3)	2.0μs(2)	3.5μs(3.5)

( ) 内为 φIIC 的周期数

SSC 推荐值: I2CSSCR 寄存器的 SSC4 ~ SSC0 位的 10 进制值

## 24.6 数据的发送 / 接收

用 7 位地址格式进行主控发送或者从属接收时的数据发送 / 接收格式例子如 “24.6.1 主控发送的例子～24.6.2 从属接收的例子” 所示，这些例子是通过表 24.12 的条件进行初始设定后开始通信的例子。

表 24.12 初始设定的例子

寄存器	设定值	设定项目	设定内容
I2CSAR	02h	自从属地址	1
I2CCCR	85h	SCL 频率	100kHz( $\phi$ IIC=4MHz)
		时钟模式	标准模式
		ACK 时钟	有
I2CCR2	00h	超时的检测	禁止
I2CCR1	13h	停止条件检测的中断	允许
		数据接收结束的中断	允许
		$\phi$ IIC 的选择	IIC 的 2 分频
I2CSR	0Fh	通信模式	从属接收模式
I2CSSCR	98h	SSC 值 (参照表 24.11)	24
		选择生成开始条件 / 停止条件	长模式
I2CCR0	08h	发送 / 接收的位数	8 位
		I <sup>2</sup> C 总线接口功能	有效 (允许通信)
		数据格式	寻址格式
I2CMR	09h	I <sup>2</sup> C 总线接口 / UART2 的转换	选择 I <sup>2</sup> C 总线接口
		I <sup>2</sup> C 总线接口的时钟源	IIC=f2n

### 24.6.1 主控发送的例子

以下说明主控发送的步骤和运行。主控发送的运行例子如图 24.23 所示。下述的 (A) ~ (C) 分别表示图 24.23 中对应的执行步骤，其中的 (1) ~ (3) 表示由程序进行的指示，由 “→” 开始的行表示由单片机进行的自动处理。

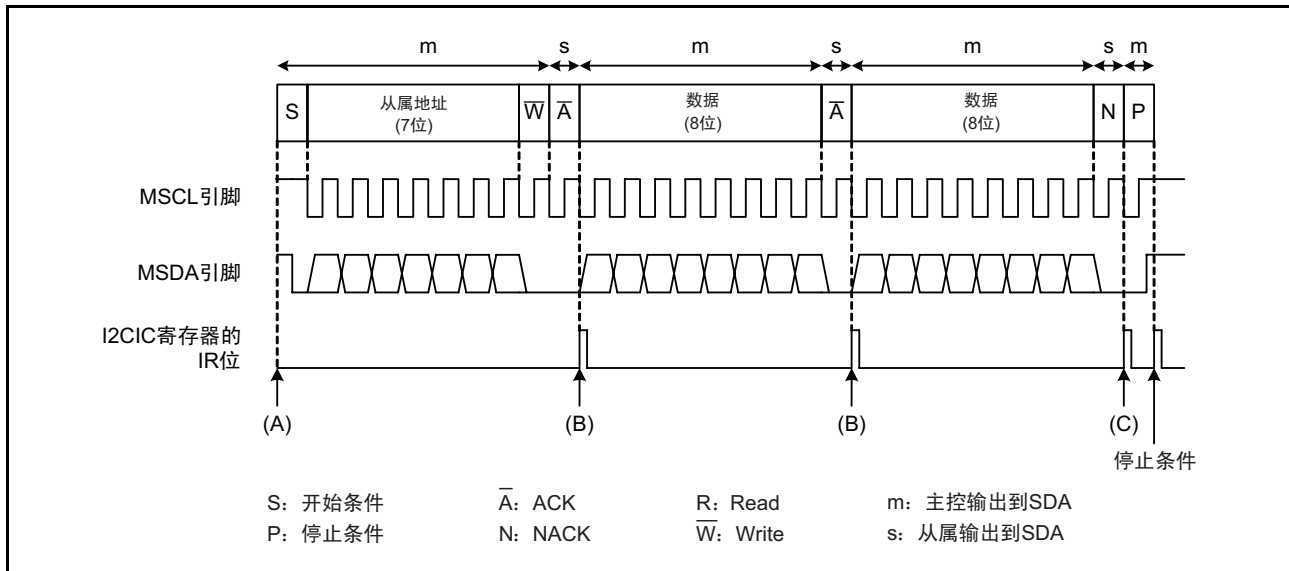


图 24.23 主控发送的运行例子

#### (A) 发送从属地址

- (1) 确认 I2CSR 寄存器的 BBSY 标志为 “0” (总线空闲)
- (2) 给 I2CSR 寄存器写 “E0h”  
→ 进入开始条件待机状态
- (3) 给 I2CTRSR 寄存器的高 7 位写发送目标 (从属) 地址  
→ 生成开始条件  
→ 发送从属地址

#### (B) 发送数据 (在 I<sup>2</sup>C 总线中断程序中处理)

- (1) 给 I2CTRSR 寄存器写发送数据  
→ 发送数据  
发送多个字节的数据时，连续给 I2CTRSR 寄存器写数据。

#### (C) 主控发送结束 (在 I<sup>2</sup>C 总线中断程序中处理)

- (1) 给 I2CSR 寄存器写 “C0h”  
→ 进入停止条件待机状态
- (2) 给 I2CTRSR 寄存器写虚拟数据  
→ 生成停止条件

除发送结束的情况外，在没有来自从属器件的 ACK 应答时 (如图 24.23 所示的 NACK 返回的情况)，也必须进行主控发送的结束处理。

### 24.6.2 从属接收的例子

以下说明从属接收的步骤和运行。从属接收的运行例子如图 24.24 所示。下述的 (A) ~ (D) 分别表示图 24.24 中对应的执行步骤，其中的 (1) ~ (3) 表示由程序进行的指示，由 “→” 开始的行表示由单片机进行的自动处理。

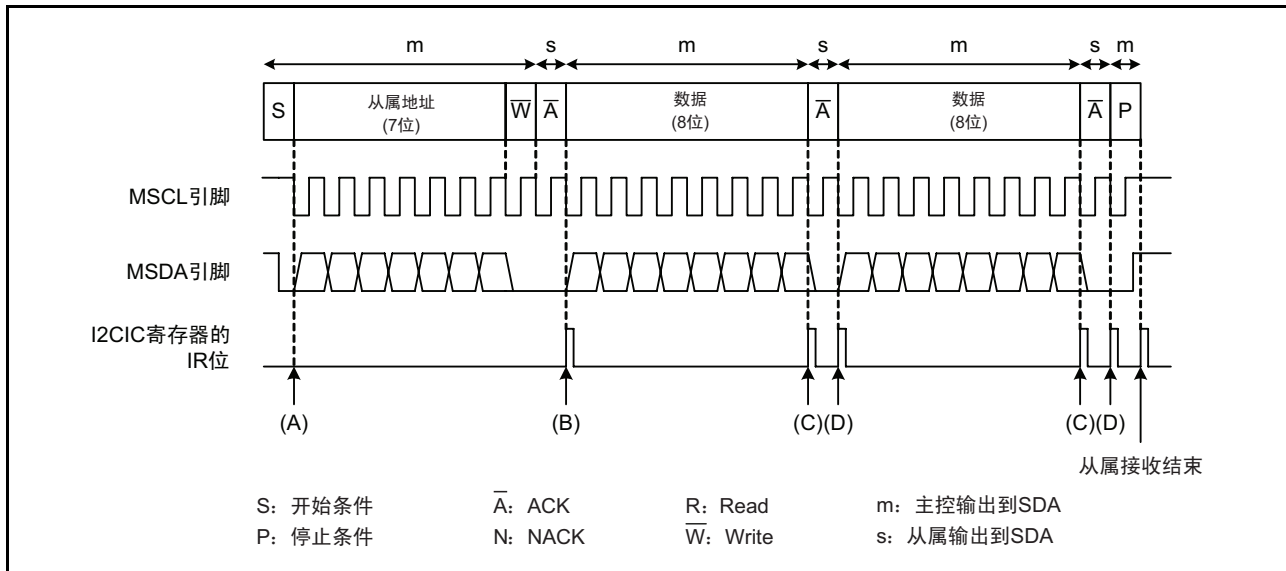


图 24.24 从属接收的运行例子

- (A) 接收从属地址（由单片机进行的自动处理）
- 检测到开始条件
  - 接收从属地址
  - 在以下的情况下发送 ACK，并且发生 I<sup>2</sup>C 总线接口中断。
    - 接收全呼地址（I2CSR 寄存器的 ADZ 标志为 “1”）
    - 检测到地址匹配（I2CSR 寄存器的 AAS 标志为 “1”）
- (B) 从属接收开始（在 I<sup>2</sup>C 总线中断程序中处理）
- (1) 确认 I2CSR 寄存器的内容，如果 TRS 位为 “0”，就开始从属接收。
  - (2) 给 I2CTRSR 寄存器写虚拟数据。
    - 开始接收数据
- (C) 数据接收结束（在 I<sup>2</sup>C 总线中断程序中处理）
- (1) 从 I2CTRSR 寄存器读接收数据。
  - (2) 为最终数据时，将 I2CCCR 寄存器的 ACKD 位置 “1”（NACK）。
  - (3) 为最终数据以外的数据时，将 I2CCCR 寄存器的 ACKD 位置 “0”（ACK）。
    - 发送 ACK/NACK，并且发生 I<sup>2</sup>C 总线接口中断。
- (D) ACK 发送结束（在 I<sup>2</sup>C 总线中断程序中处理）
- (1) 给 I2CTRSR 寄存器写虚拟数据。
    - 为最终数据时，检测停止条件。
    - 为最终数据以外的数据时，重新开始接收数据。



## 24.7 使用时的注意事项

### 24.7.1 I<sup>2</sup>C 总线接口电路的各寄存器的存取

以下是读写 I<sup>2</sup>C 总线接口电路的各控制寄存器时的注意事项。

- I2CTRSR 寄存器**  
 在数据的发送/接收过程中不能写此寄存器。否则，用于发送/接收的位计数器复位，并且无法正常的进行数据的发送/接收。
- I2CCR0 寄存器**  
 在检测到开始条件和完成1字节的发送/接收时，BC2~BC0位变为“000b”。因此，如果在此时序读写此寄存器，就有可能读写到预想不到的值。位计数器的复位时序如图24.26、图24.27所示。
- I2CCCR 寄存器**  
 在发送/接收的过程中，不能改写ACKD位以外的位。否则I<sup>2</sup>C总线时钟电路复位，并且无法正常的进行发送/接收。
- I2CCR1 寄存器**  
 必须在I2CCR0寄存器的ICE位设定为“0”（I<sup>2</sup>C总线接口功能无效）的状态下，才能改写ICK4~ICK0位。如果读RIE位，读出值就为内部WAIT标志。因此，不能对本寄存器使用位处理指令（读-修改-写指令）。
- I2CSR 寄存器**  
 所有的位因通信状态而变化，因此，不能使用位处理指令（读-修改-写指令）。另外，在设定通信模式的MST位和TRS位的变化时序，不能读写此寄存器。否则，就有可能读写到意想不到的值。MST位和TRS位的变化时序如图24.25~图24.27所示。

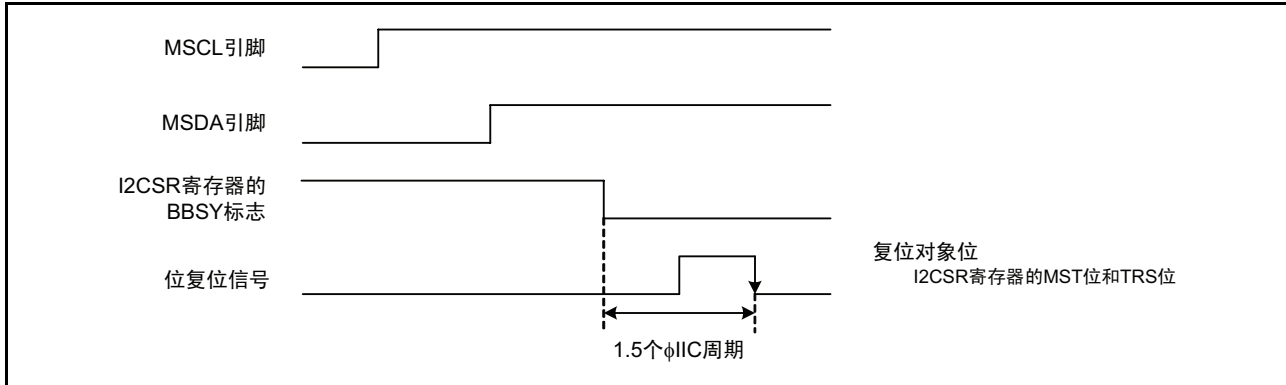


图 24.25 位复位的时序（检测到停止条件的情况）

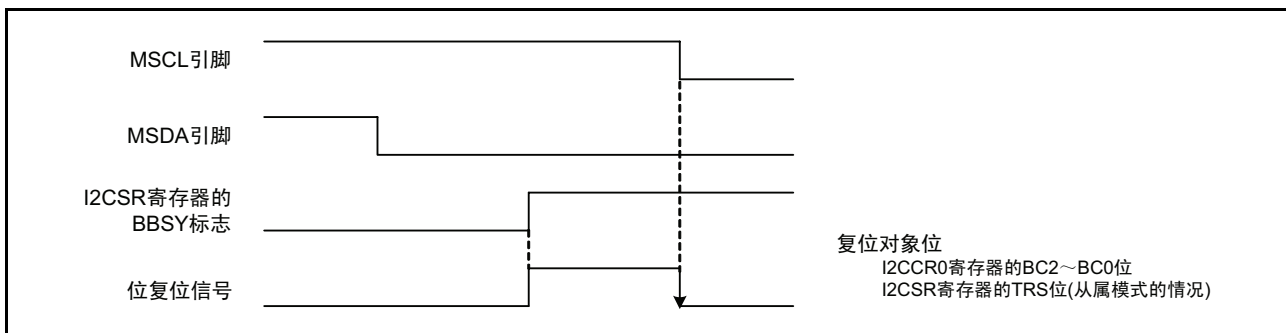


图 24.26 位复位的时序（检测到开始条件的情况）

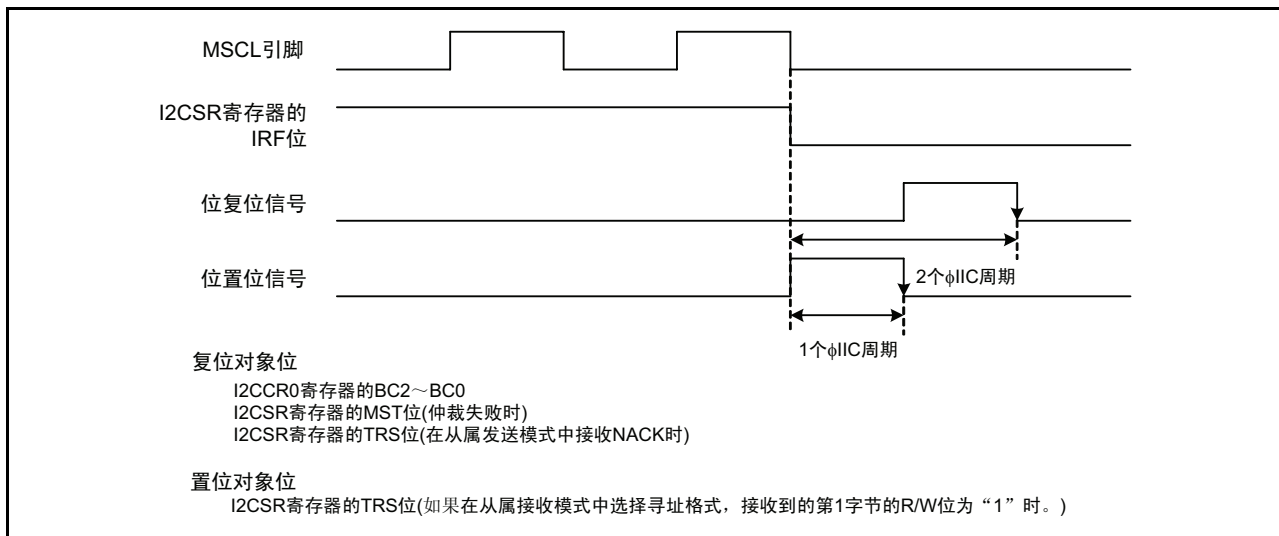


图 24.27 位置位 / 复位的时序 (数据发送 / 接收结束的情况)

### 24.7.2 重新开始条件的生成

在发送完 1 字节的数据后,生成重新开始条件时,按以下的步骤执行。

1. 给 I2CSR 寄存器写 “E0h” (开始条件待机状态、MSDA 引脚释放)
2. 等待 MSDA 引脚变为 “H” 电平
3. 给 I2CTRSR 寄存器写从属数据 (生成开始条件触发)

重新开始条件的生成时序如图 24.28 所示。

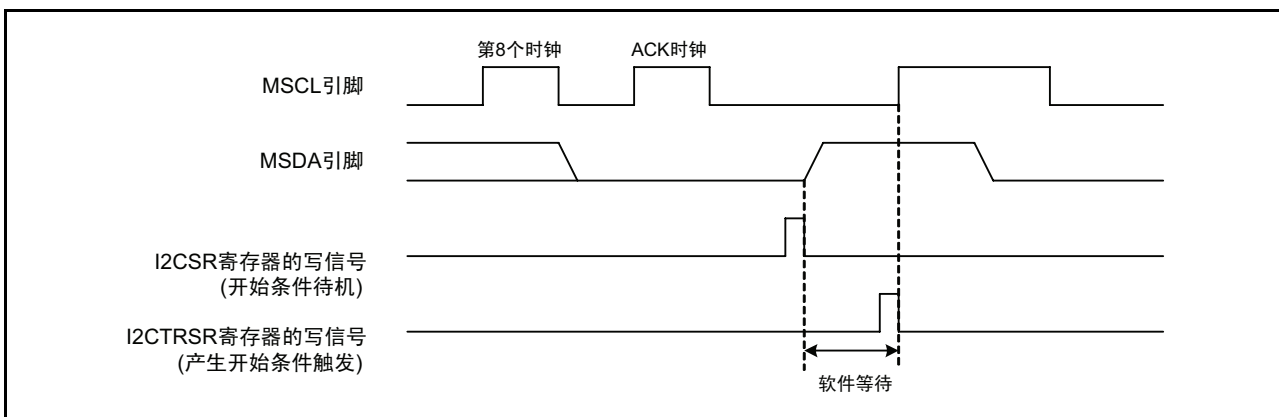


图 24.28 重新开始条件的生成时序

## 25. CAN 模块

R32C/117 群内置了具有一个通道并且符合 ISO11898-1 规格的 CAN（Controller Area Network）模块（CAN0）。CAN 模块可发送 / 接送标准（11 位）IDentifier（以下简称为 ID）格式和扩展（29 位）ID 格式的消息。

CAN 模块的规格如表 25.1、表 25.2、CAN 模块的框图如图 25.1 所示。

另外，必须在外外部连接 CAN 总线收发器。

表 25.1 CAN 模块的规格（1）

项目	规格
协议	符合 ISO11898-1 规格
位速率	最大 1Mbps
消息框	32 个邮箱 可选择两种邮箱模式 <ul style="list-style-type: none"> <li>• 通常邮箱模式 可将 32 个邮箱全部设定为用于发送或者接收的邮箱</li> <li>• FIFO 邮箱模式 可将 24 个邮箱设定为用于发送或者接收的邮箱 可将剩下的邮箱设定为用于发送的 4 段 FIFO 邮箱和用于接收的 4 段 FIFO 邮箱</li> </ul>
接收	<ul style="list-style-type: none"> <li>• 可接受数据帧和远程帧</li> <li>• 可选择接收的 ID 格式（仅标准 ID、仅扩展 ID、或者标准 ID 和扩展 ID）</li> <li>• 可选择单触发接收功能</li> <li>• 可选择改写模式（保存消息）或者溢出模式（废弃消息）</li> <li>• 可给每个邮箱设定允许 / 禁止接收结束的中断</li> </ul>
接收滤波器	8 个接收屏蔽（每 4 个邮箱有一个屏蔽） 可给每个邮箱设定屏蔽有效 / 无效
发送	<ul style="list-style-type: none"> <li>• 可发送数据帧和远程帧</li> <li>• 可选择发送的 ID 格式（仅标准 ID、仅扩展 ID、或者标准 ID 和扩展 ID）</li> <li>• 可选择单触发发送功能</li> <li>• 可选择 ID 优先发送模式或者邮箱号优先发送模式</li> <li>• 可中止发送请求（可通过标志确认中止结束）</li> <li>• 可给每个邮箱设定允许 / 禁止发送结束的中断</li> </ul>
总线关闭返回模式的转移	可选择转移到从总线关闭状态返回的模式 <ul style="list-style-type: none"> <li>• 符合 ISO11898-1 规格</li> <li>• 在开始总线关闭时自动转移到 CAN Halt 模式</li> <li>• 在总线关闭结束时自动转移到 CAN Halt 模式</li> <li>• 通过程序转移到 CAN Halt 模式</li> <li>• 通过程序转移到主动错误状态</li> </ul>
错误状态的监视	<ul style="list-style-type: none"> <li>• 可监视 CAN 总线错误（填充错误、形式错误、ACK 错误、CRC 错误、位错误、ACK 定界符错误）</li> <li>• 可检测错误状态的转移（错误报警、被动错误、总线关闭开始、总线关闭返回）</li> <li>• 可读错误计数器</li> </ul>
时戳功能	16 位计数器的时戳功能 基准时钟可从 1、2、4、8 位时间中选择

表 25.2 CAN 模块的规格 (2)

项目	规格
中断源	6 种 <ul style="list-style-type: none"> <li>接收结束</li> <li>发送结束</li> <li>接收 FIFO</li> <li>发送 FIFO</li> <li>错误</li> <li>唤醒</li> </ul>
CAN 睡眠模式	可通过停止 CAN 时钟减低消耗电流
软件支持单元	3 个软件的支持单元 <ul style="list-style-type: none"> <li>支持接收滤波器</li> <li>支持邮箱检索 (接收邮箱检索、发送邮箱检索、消息丢失检索)</li> <li>支持通道检索</li> </ul>
CAN 时钟源	可选择外围总线时钟或者主时钟
测试模式	具有 3 个用户鉴定用的测试模式 <ul style="list-style-type: none"> <li>只监听模式</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>

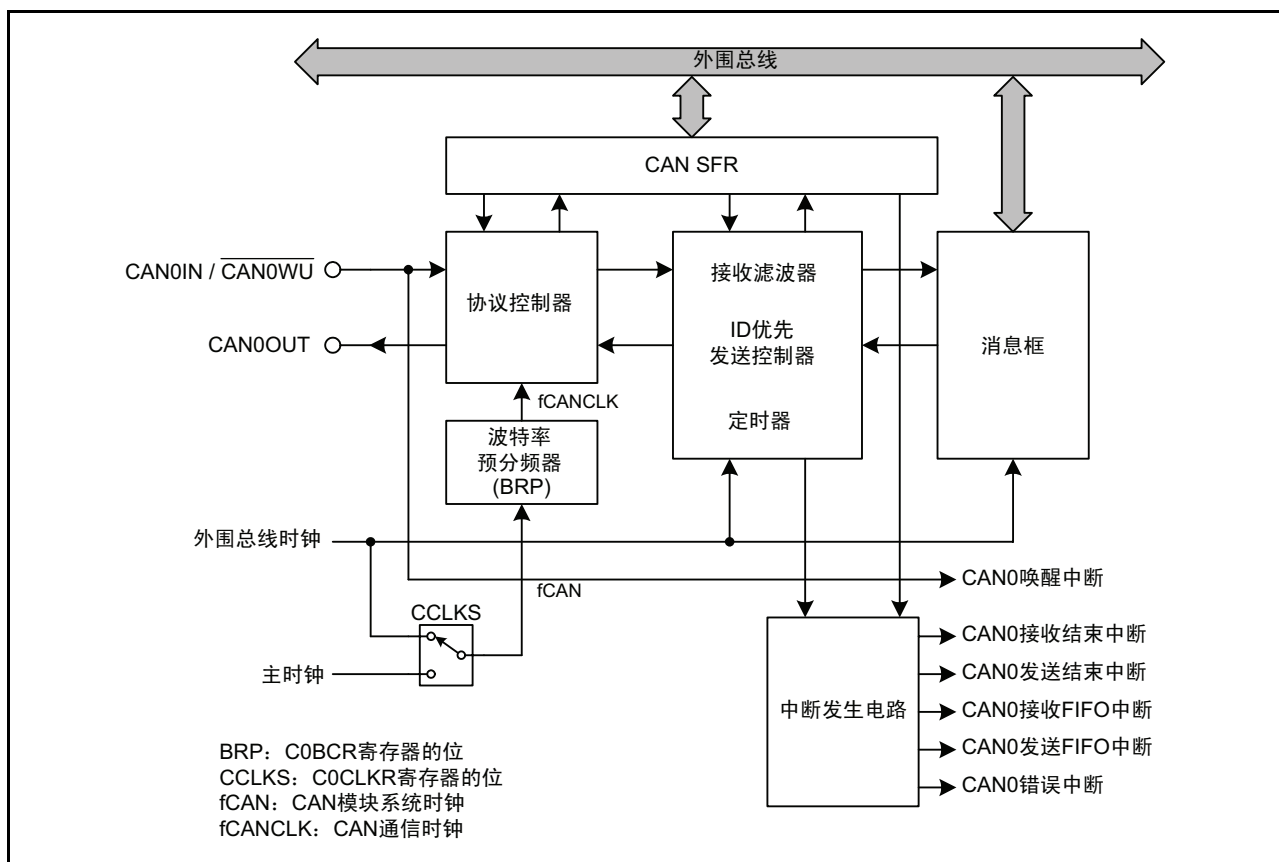


图 25.1 CAN 模块的框图

- CAN0IN/CAN0OUT: CAN的输入/输出引脚。
- 协议控制器: 进行总线仲裁或者发送/接收时的位时序、填充处理、错误处理等的CAN协议处理。
- 消息框: 由可用作发送或者接收的32个邮箱构成。各邮箱有固有的ID、数据长度码、8字节的数据区以及时戳。
- 接收滤波器: 进行接收消息的滤波器处理。该滤波器处理使用C0MKR0~C0MKR7寄存器。
- 定时器: 用于时戳功能。写消息保存到邮箱时的定时器值, 并且该值被作为时戳值。
- 唤醒: 如果在CAN总线上检测到消息, 就产生CAN0唤醒中断请求。
- 中断发生电路: 可产生以下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 SFR: CAN的相关寄存器。详细内容请参照“25.1 CAN SFR”。

## 25.1 CAN SFR

CAN的相关寄存器如图 25.2 ~图 25.11、图 25.13、图 25.14、图 25.16 ~图 25.20、图 25.22、图 25.24 ~图 25.30。

## 25.1.1 CAN0 控制寄存器 (COCTLR 寄存器)

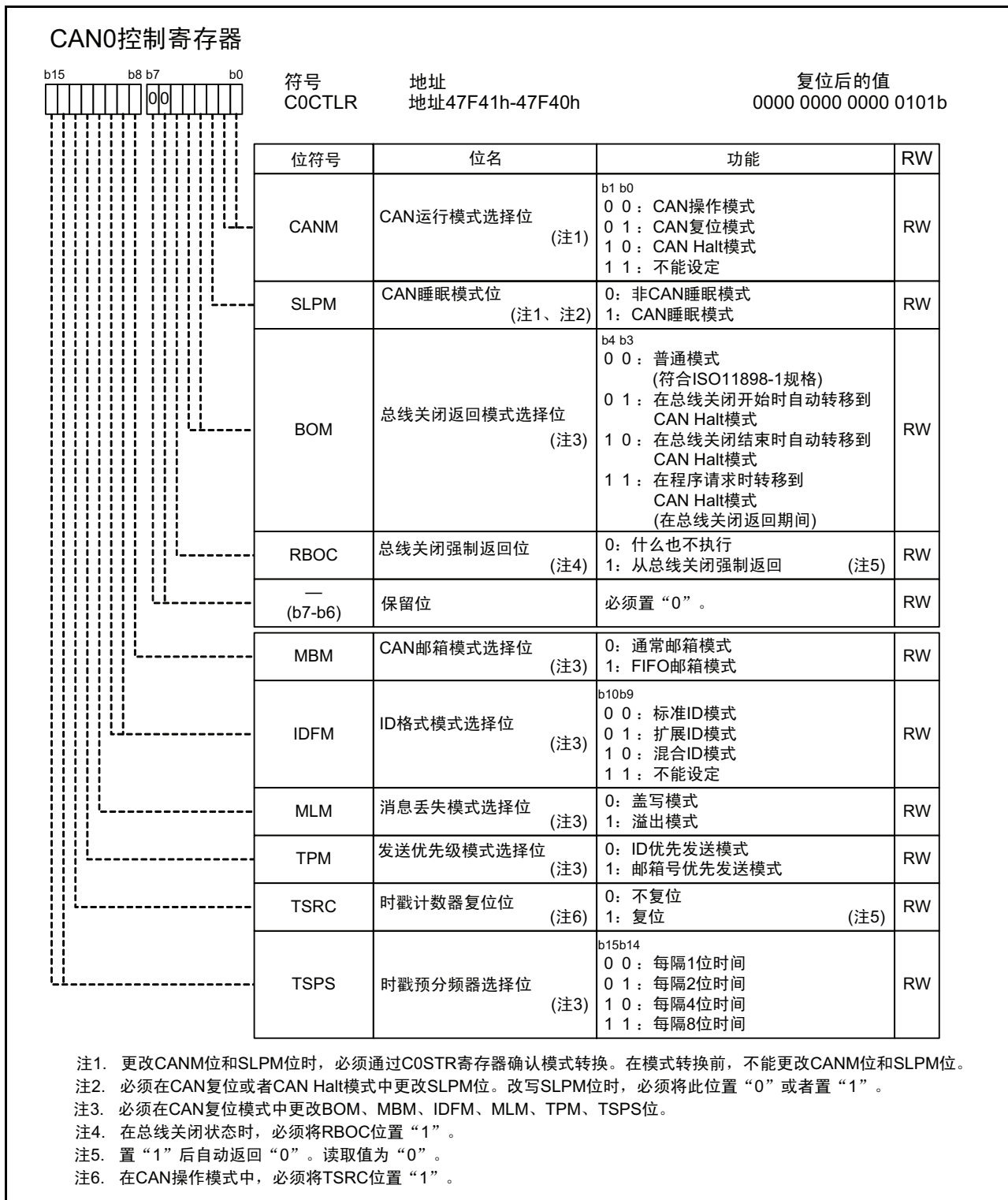


图 25.2 COCTLR 寄存器

### 25.1.1.1 CANM 位

CANM 位是选择 CAN 模块模式（CAN 操作模式、CAN 复位模式、CAN Halt 模式）的位。详细内容请参照“25.2 运行模式”。

通过 SLPM 位设定 CAN 睡眠模式。

不能设定为“11b”。

根据 BOM 位的设定转移到 CAN Halt 模式时，CANM 位自动变为“10b”。

### 25.1.1.2 SLPM 位

将此位置“1”，就进入 CAN 睡眠模式。

将此位置“0”，就解除 CAN 睡眠模式。

详细内容请参照“25.2 运行模式”。

### 25.1.1.3 BOM 位

BOM 位用于选择 CAN 模块的总线关闭返回模式。

此位为“00b”时，从总线关闭的返回符合 ISO11898-1 规格。即，CAN 模块在检测到 128 次 11 个连续的隐性位后，再次进入 CAN 通信（主动错误状态）。从总线关闭返回时，产生总线关闭返回中断请求。

此位为“01b”时，如果 CAN 模块进入总线关闭状态，就在 C0CTRL 寄存器的 CANM 位变为“10b”（CAN Halt 模式）后，转移到 CAN Halt 模式。从总线关闭返回时，不产生总线关闭返回中断请求，C0TECR 寄存器和 CORECR 寄存器变为“00h”。

此位为“10b”时，如果 CAN 模块进入总线关闭状态，CANM 位就变为“10b”，并且在从总线关闭状态返回（检测到 128 次 11 个连续的隐性位）后，转移到 CAN Halt 模式。从总线关闭返回时，产生总线关闭返回中断请求，并且 C0TECR 寄存器和 CORECR 寄存器变为“00h”。

此位为“11b”时，如果在 CAN 模块还处于总线关闭状态时，将 CANM 位置“10b”，就转移到 CAN Halt 模式。从总线关闭返回时，不产生总线关闭返回中断请求，C0TECR 寄存器和 CORECR 寄存器变为“00h”。但是，如果在将 CANM 位置“10b”前，检测到 128 次 11 个连续的隐性位，就在从总线关闭返回时，产生总线关闭返回中断请求。

如果与 CAN 模块转移到 CAN Halt 模式同时（BOM 位为“01b”时：总线关闭开始，或者 BOM 位为“10b”时：总线关闭结束），CPU 请求转移到 CAN 复位模式，则优先 CPU 的请求。

### 25.1.1.4 RBOC 位

在总线关闭时，如果将此位置“1”（从总线关闭强制返回），就从总线关闭状态强制返回。此位自动变为“0”。错误状态是从总线关闭状态变到主动错误状态。

如果将此位置“1”，C0RECR 寄存器和 C0TECR 寄存器就变为“00h”，并且 C0STR 寄存器的 BOST 位变为“0”（CAN 模块不处于总线关闭状态）。其他寄存器不变，而且从总线关闭返回时，不产生总线关闭返回中断请求。

只能在 BOM 位为“00b”（普通模式）时使用。

### 25.1.1.5 MBM 位

此位为“0”（通常邮箱模式）时，邮箱 [0] ~ [31] 被设定为发送或者接收邮箱。

此位为“1”（FIFO 邮箱模式）时，邮箱 [0] ~ [23] 被设定为发送或者接收邮箱，邮箱 [24] ~ [27] 被设定为发送 FIFO，邮箱 [28] ~ [31] 被设定为接收 FIFO。

发送数据写到邮箱 [24]（邮箱 [24] 是发送 FIFO 的窗口邮箱），接收数据从邮箱 [28] 读出（邮箱 [28] 是接收 FIFO 的窗口邮箱）。

邮箱的设定如表 25.3 所示。

表 25.3 邮箱的设定

邮箱	MBM=0 (通常邮箱模式)	MBM=1 (注 1) (FIFO 邮箱模式)
邮箱 [0] ~ [23]	通常邮箱	通常邮箱
邮箱 [24] ~ [27]		发送 FIFO
邮箱 [28] ~ [31]		接收 FIFO

注 1. 在 MBM 位为“1”时，必须注意以下几点：

- 通过 C0TFCR 寄存器控制发送 FIFO。  
邮箱 [24] ~ [27] 的 C0MCTLj 寄存器 (j=0 ~ 31) 无效。  
不能使用 C0MCTL24 ~ C0MCTL27 寄存器。
- 通过 C0RFCR 寄存器控制接收 FIFO。  
邮箱 [28] ~ [31] 的 C0MCTLj 寄存器无效。  
不能使用 C0MCTL28 ~ C0MCTL31 寄存器。
- 有关 FIFO 中断，请参照 COMIER 寄存器。
- COMKIVLR 寄存器的邮箱 [24] ~ [31] 的对应位无效。必须将这些位设定为“0”。
- 发送/接收 FIFO 即可以使用数据帧也可以使用远程帧。

### 25.1.1.6 IDFM 位

IDFM 位指定 ID 格式。

此位为“00b”时，全部邮箱（包括 FIFO 邮箱）仅支持标准 ID。

此位为“01b”时，全部邮箱（包括 FIFO 邮箱）仅支持扩展 ID。

此位为“10b”时，全部邮箱（包括 FIFO 邮箱）即支持标准 ID 也支持扩展 ID。选择标准 ID 和扩展 ID 时，在通常邮箱模式中，通过对应邮箱的 IDE 位指定；在 FIFO 邮箱模式中，邮箱 [0] ~ [23] 通过对应邮箱的 IDE 位指定，接收 FIFO 通过 C0FIDCR0 寄存器和 C0FIDCR1 寄存器的 IDE 位指定，发送 FIFO 通过邮箱 [24] 的 IDE 位指定。

不能将此位设定为“11b”。

### 25.1.1.7 MLM 位

MLM 位指定向未读邮箱保存新消息时的运行模式。可选择盖写模式或者溢出模式。全部邮箱（包括接收 FIFO）变为盖写模式或者溢出模式。

此位为“0”时，全部邮箱为盖写模式，在邮箱的旧消息上盖写新消息。

此位为“1”时，全部邮箱为溢出模式，废弃新消息。



#### 25.1.1.8 TPM 位

TPM 位指定发送消息时的优先级模式。可选择 ID 优先模式或者邮箱号优先模式。

全部邮箱变为 ID 优先发送或者邮箱号优先发送。

此位为“0”时，为 ID 优先发送模式，发送优先级符合 CAN 总线仲裁标准（ISO11898-1 规格）。ID 优先发送模式在通常邮箱模式中比较邮箱 [0] ~ [31] 的 ID 和被设定为发送 FIFO 的发送邮箱的 ID，在 FIFO 邮箱模式中比较邮箱 [0] ~ [23] 的 ID 和被设定为发送 FIFO 的发送邮箱的 ID。在至少有 2 个邮箱的 ID 相同时，就优先邮箱号小的邮箱。

只有预定的下一个从发送 FIFO 发送的消息被作为发送仲裁的对象。当发送 FIFO 中的消息正在发送时，发送 FIFO 中的下一个待机消息就成为发送仲裁对象。

此位为“1”时，为邮箱号优先发送模式，被设定为发送的邮箱号最小的邮箱优先发送。在 FIFO 邮箱模式中，发送 FIFO 的优先级比通常邮箱（邮箱 [0] ~ [23]）低。

#### 25.1.1.9 TSRC 位

TSRC 位用于复位时戳计数器。

如果将此位置“1”，C0TSR 寄存器就变为“0000h”。此位自动变为“0”。

#### 25.1.1.10 TSPS 位

TSPS 位选择用于时戳的预分频器。

时戳的基准时钟可从 1、2、4、或者 8 位时间中选择。

## 25.1.2 CAN0 时钟选择寄存器 (C0CLKR 寄存器)

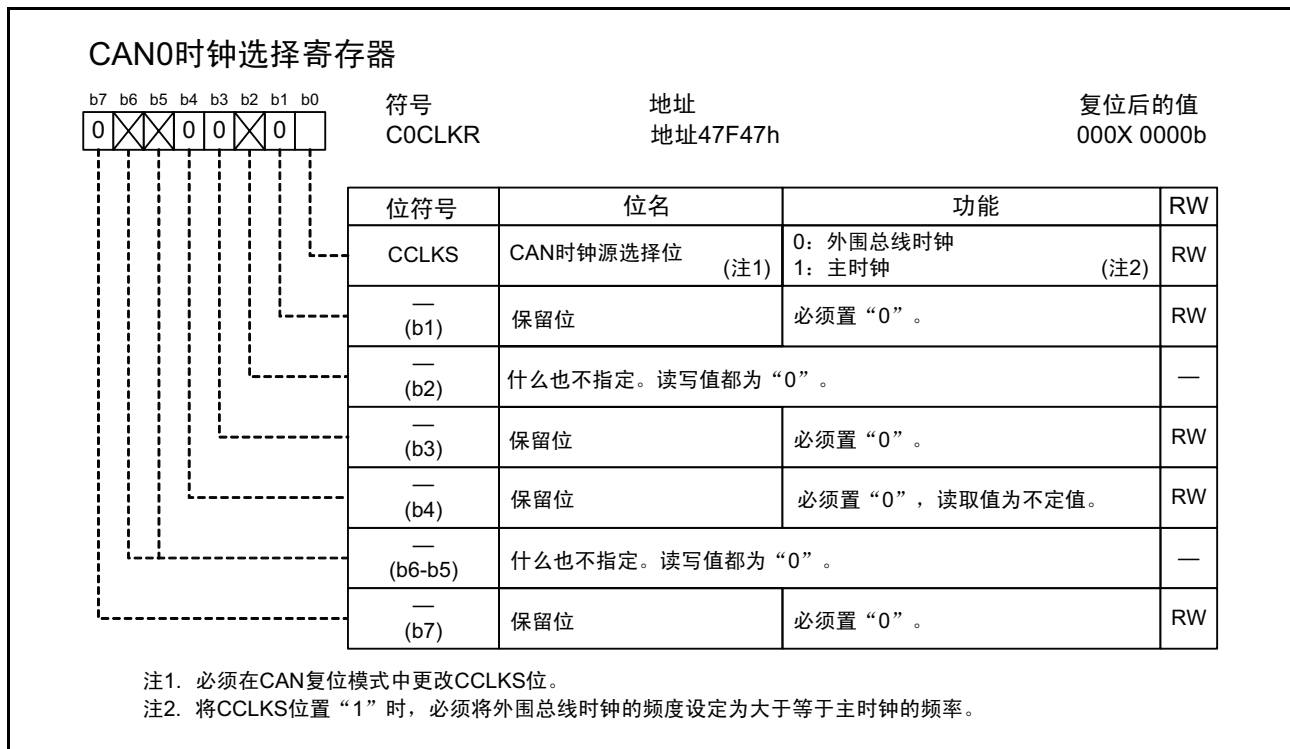


图 25.3 C0CLKR 寄存器

## 25.1.2.1 CCLKS 位

此位为“0”时，使用由 PLL 频率合成器生成的外围总线时钟作为 CAN 时钟源 (fCAN)。

此位为“1”时，不使用 PLL 频率合成器，使用从外部 XIN 引脚输入的主时钟作为 CAN 时钟源 (fCAN)。

25.1.3 CAN0 位结构寄存器 (C0BCR 寄存器)

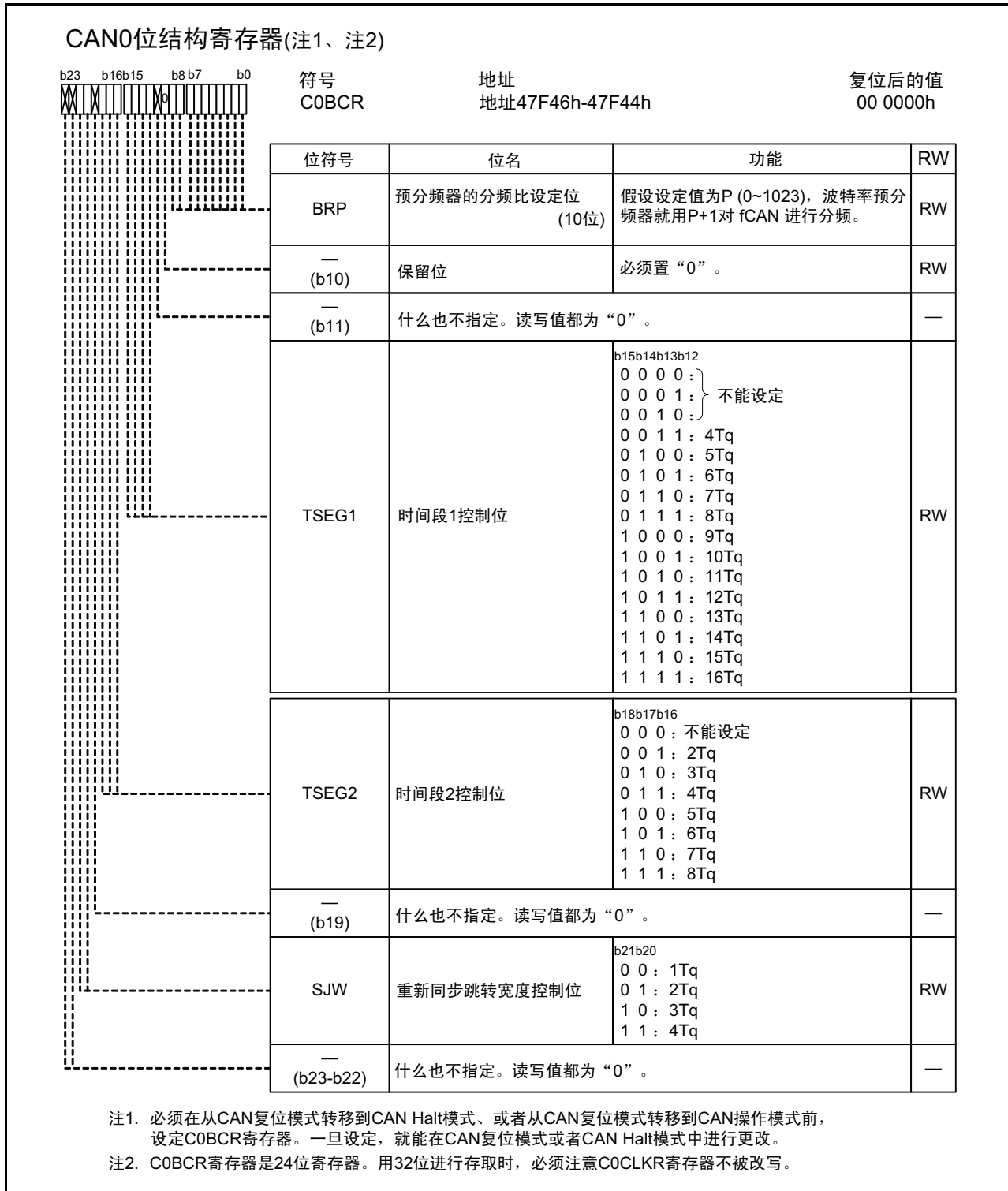


图 25.4 C0BCR 寄存器

有关位时序的设定, 请参照“25.3 CAN 通信速度的设定”。

#### 25.1.3.1 BRP 位

BRP 位用于设定 CAN 通信时钟 (fCANCLK) 的频率。  
fCANCLK 的频率为 1 Time Quantum (Tq)。

#### 25.1.3.2 TSEG1 位

TSEG 位用 Tq 值指定传播时间段 (PROP\_SEG) 和相位缓冲段 1 (PHASE\_SEG1) 的合计长度。  
可设定 4 ~ 16Tq 的值。

#### 25.1.3.3 TSEG2 位

TSEG2 位用 Tq 值指定相位缓冲段 2 (PHASE\_SEG2) 的长度。  
可设定 2 ~ 8Tq 的值。  
必须设定比 TSEG1 位小的值。

#### 25.1.3.4 SJW 位

SJW 位通过用 Tq 值指定重新同步跳转宽度 (Resynchronization Jump Width)。  
可设定 1 ~ 4Tq 的值。  
必须设定小于等于 TSEG2 位的值。

## 25.1.4 CAN0 屏蔽寄存器 k (C0MKRk 寄存器) (k=0 ~ 7)



图 25.5 C0MKR0 ~ C0MKR7 寄存器

有关 FIFO 邮箱模式中的屏蔽功能，请参照“25.5 接收滤波器的处理和屏蔽功能”。

## 25.1.4.1 EID 位

EID 位是对应 CAN 扩展 ID 位的滤波器屏蔽位。在接收扩展 ID 的消息时使用此位。

此位为“0”时，对应的 EID 位不比较接收到的 ID 和邮箱的 ID。

此位为“1”时，对应的 EID 位比较接收到的 ID 和邮箱的 ID。

## 25.1.4.2 SID 位

SID 位是对应 CAN 标准 ID 位的滤波器屏蔽位。在接收标准 ID 和扩展 ID 的消息时使用此位。

此位为“0”时，对应的 SID 位不比较接收到的 ID 和邮箱的 ID。

此位为“1”时，对应的 SID 位比较接收到的 ID 和邮箱的 ID。

## 25.1.5 CAN0 FIFO 接收 ID 比较寄存器 n (C0FIDCR0、C0FIDCR1 寄存器) (n=0,1)



图 25.6 C0FIDCR0、C0FIDCR1 寄存器

在 COCTRL 寄存器的 MBM 位为“1”(FIFO 邮箱模式)时有效。COMB28 ~ COMB31 寄存器的 EID、SID、RTR、IDE 位无效。

有关使用方法, 请参照“25.5 接收滤波器的处理和屏蔽功能”。

## 25.1.5.1 EID 位

EID 位设定数据帧和远程帧的扩展 ID。在接收扩展 ID 的消息时使用此位。

## 25.1.5.2 SID 位

SID 位设定数据帧和远程帧的标准 ID。在接收标准 ID 和扩展 ID 的消息时使用此位。

## 25.1.5.3 RTR 位

RTR 位设定数据帧或者远程帧的帧格式。

指定以下的运行。

- 在 C0FIDCR0、C0FIDCR1 寄存器的 RTR 位都为“0”时, 只能接收数据帧。
- 在 C0FIDCR0、C0FIDCR1 寄存器的 RTR 位都为“1”时, 只能接收远程帧。
- 在 C0FIDCR0、C0FIDCR1 寄存器的 RTR 位分别为“0”和“1”时, 能接受数据帧和远程帧。

## 25.1.5.4 IDE 位

IDE 设定标准 ID 或者扩展 ID 的 ID 格式。

在 COCTRL 寄存器的 IDFM 位为“10b”(混合 ID 模式)时有效。

在 IDFM 位为“10b”时, 指定以下的运行。

- 在 C0FIDCR0、C0FIDCR1 寄存器的 IDE 位都为“0”时, 只能接收标准 ID 帧。
- 在 C0FIDCR0、C0FIDCR1 寄存器的 IDE 位都为“1”时, 只能接收扩展 ID 帧。
- 在 C0FIDCR0、C0FIDCR1 寄存器的 IDE 位分别为“0”和“1”时, 能接受标准 ID 帧和扩展 ID 帧。

## 25.1.6 CAN0 屏蔽无效寄存器 (COMKIVLR 寄存器)

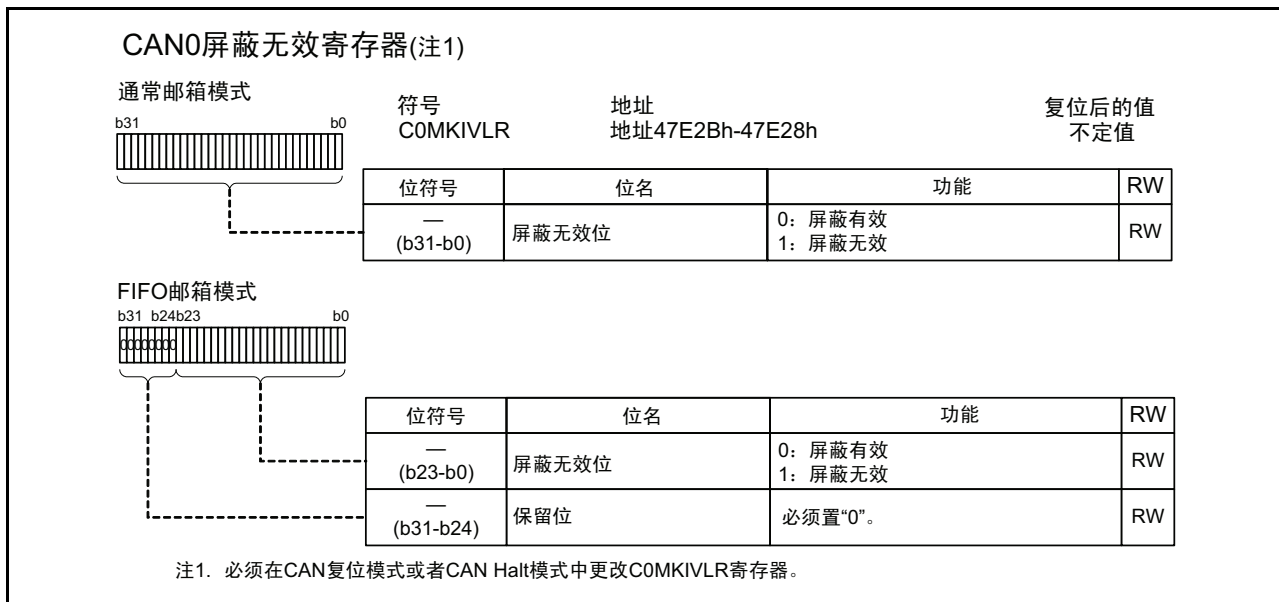


图 25.7 COMKIVLR 寄存器

各个位对应相同号的邮箱。为“1”时，位号的对应邮箱的接收屏蔽无效。此时，邮箱仅在接收消息的 ID 和 COMB<sub>j</sub> 寄存器 (j=0 ~ 31) 的 SID、EID 位匹配时进行接收。

### 25.1.7 CAN0 邮箱（COMBj 寄存器）（j=0 ~ 31）

CAN 邮箱的存储器分配如表 25.4、CAN 数据帧的结构如表 25.5 所示。

CAN0 邮箱复位后的值为不定值。

表 25.4 CAN0 邮箱的存储器分配

地址	消息内容
CAN0	存储器分配
47C00h+j×16+0	EID7 ~ EID0
47C00h+j×16+1	EID15 ~ EID8
47C00h+j×16+2	SID5 ~ SID0、EID17、EID16
47C00h+j×16+3	IDE、RTR、SID10 ~ SID6
47C00h+j×16+4	—
47C00h+j×16+5	数据长度代码（DLC）
47C00h+j×16+6	数据字节 0
47C00h+j×16+7	数据字节 1
⋮	⋮
⋮	⋮
⋮	⋮
47C00h+j×16+13	数据字节 7
47C00h+j×16+14	时戳低位字节
47C00h+j×16+15	时戳高位字节

j: 邮箱号（j=0 ~ 31）

表 25.5 CAN 数据帧的结构

SID10 ~ SID6	SID5 ~ SID0	EID17 ~ EID16	EID15 ~ EID8	EID7 ~ EID0	DLC3 ~ DLC0	DATA0	DATA1	...	DATA7
-----------------	----------------	------------------	-----------------	----------------	----------------	-------	-------	-----	-------



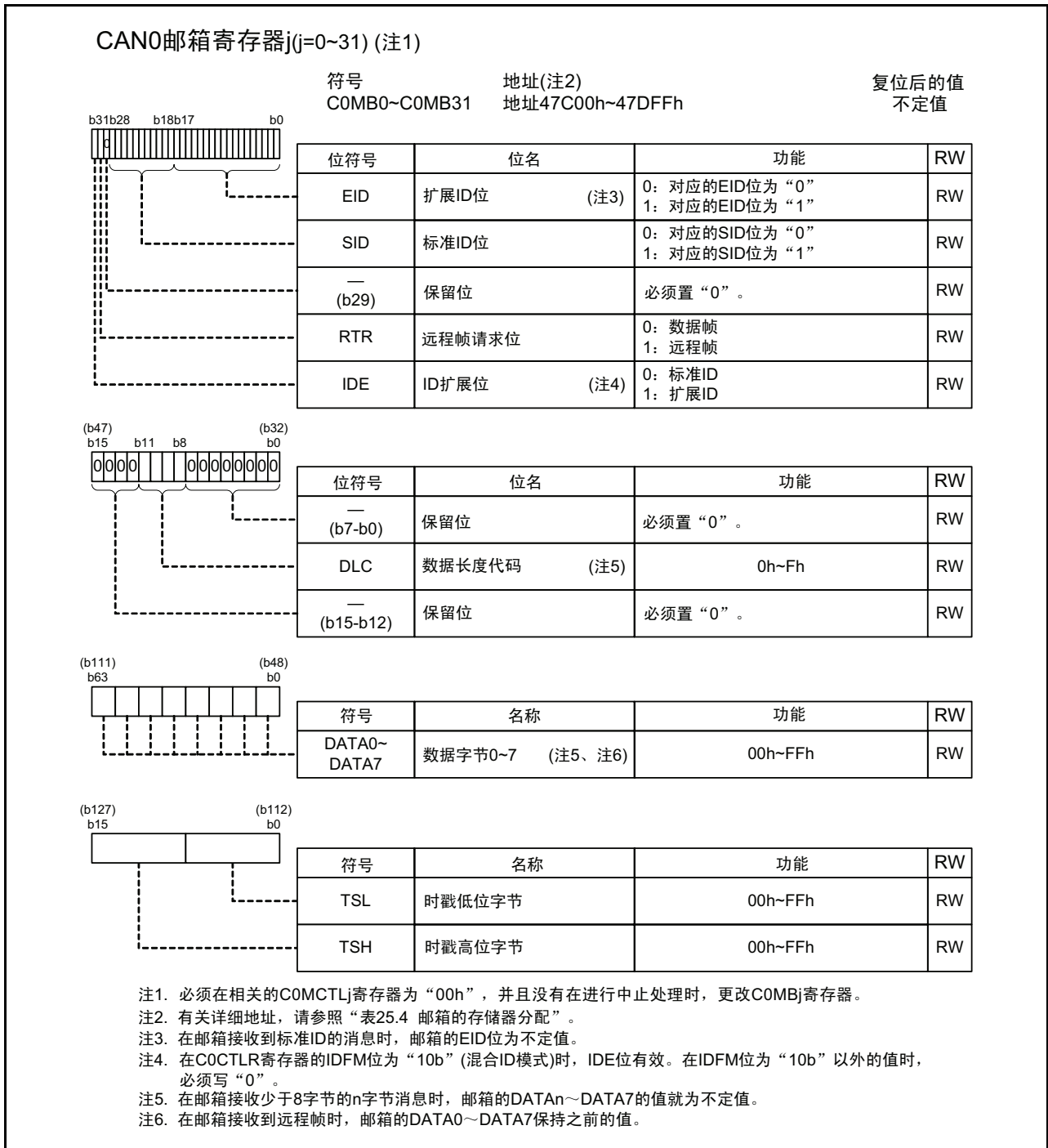


图 25.8 C0MBj 寄存器

只要不接受新消息，各邮箱的内容就保持之前的值。

#### 25.1.7.1 EID 位

EID 位设定数据帧和远程帧的扩展 ID。在发送 / 接收扩展 ID 的消息时使用此位。

#### 25.1.7.2 SID 位

SID 位设定数据帧和远程帧的标准 ID。在发送 / 接收标准 ID 和扩展 ID 的消息时使用此位。

### 25.1.7.3 RTR 位

RTR 位设定数据帧或者远程帧的帧格式。指定以下的运行。

- 接收邮箱只接受通过 RTR 位选择的帧格式。
- 发送邮箱用通过 RTR 位选择的帧格式进行发送。
- 接收 FIFO 邮箱接收通过 C0FIDCR0、C0FIDCR1 寄存器的 RTR 位选择的数据帧、远程帧或者两种帧。
- 发送 FIFO 邮箱发送通过相关的发送消息的 RTR 位选择的数据帧或者远程帧。

### 25.1.7.4 IDE 位

IDE 位设定标准 ID 或者扩展 ID 的 ID 格式。

在 COCTLR 寄存器的 IDFM 位为 “10b”（混合 ID 模式）时有效。

在 IDFM 位为 “10b” 时，IDE 位指定以下的运行。

- 接收邮箱只接收通过 IDE 位选择的 ID 格式。
- 发送邮箱用通过 IDE 位选择的 ID 格式进行发送。
- 接收 FIFO 邮箱接收通过 C0FIDCR0、C0FIDCR1 寄存器的 IDE 位选择的标准 ID、扩展 ID 或者两者的消息。
- 发送 FIFO 邮箱发送通过相关的发送消息的 IDE 位选择的标准 ID 或者扩展 ID 的消息。

### 25.1.7.5 DLC (Data Length Code)

DLC 设定使用数据帧发送数据时发送数据的字节数，以及使用远程帧请求数据时请求数据的字节数。

在接收到数据帧时，保存接收到的数据的字节数。在接收到远程帧时，保存请求的数据的字节数。

与 DLC 对应的数据长度如表 25.6 所示。

表 25.6 与 DLC 对应的数据长度

DLC[3]	DLC[2]	DLC[1]	DLC[0]	数据长度
0	0	0	0	0 字节
0	0	0	1	1 字节
0	0	1	0	2 字节
0	0	1	1	3 字节
0	1	0	0	4 字节
0	1	0	1	5 字节
0	1	1	0	6 字节
0	1	1	1	7 字节
1	X	X	X	8 字节

X: 任意值

### 25.1.7.6 DATA0 ~ DATA7

DATA0 ~ DATA7 保存已发送或者接收的 CAN 消息数据。从 DATA0 开始发送或者接收。在 CAN 总线上的位序是 MSB first，并且从 bit7 开始发送或者接收。

### 25.1.7.7 TSL、TSH

TSL、TSH 保存接收消息被保存到邮箱时的时戳计数器值。

## 25.1.8 CAN0 邮箱中断允许寄存器 (COMIER 寄存器)

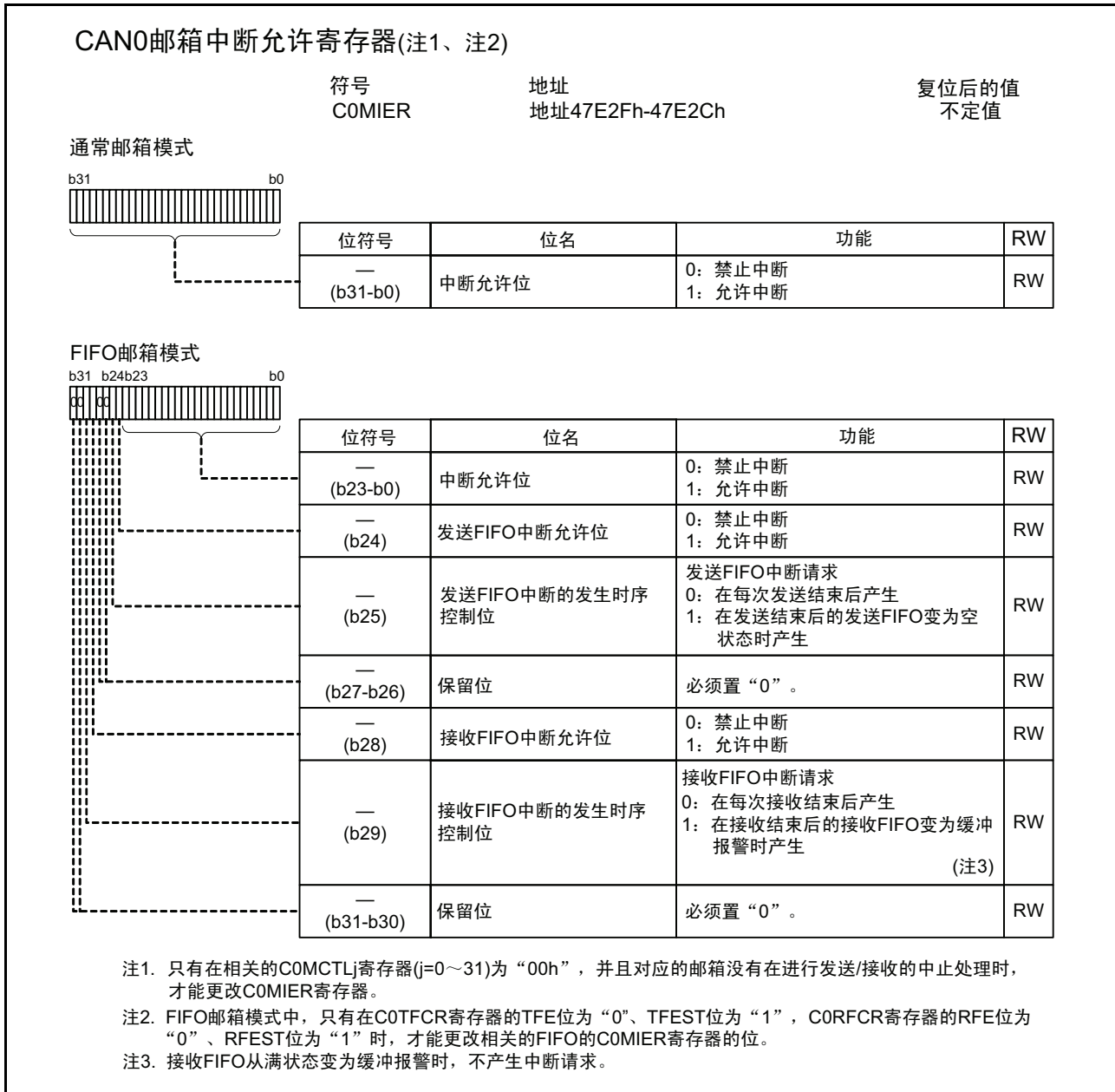


图 25.9 COMIER 寄存器

可对邮箱各别设定允许中断。

在通常邮箱模式 (bit0 ~ 31) 和 FIFO 邮箱模式 (bit0 ~ 23) 中，各个位对应相同号的邮箱。这些位允许或者禁止对应的邮箱的发送结束 / 接收结束中断。

FIFO 邮箱模式中的 bit24、bit25、bit28、bit29 位允许或者禁止发送 / 接收 FIFO 中断，以及指定中断请求的产生时序。

缓冲报警是指第 3 个消息已保存到接收 FIFO 的状态。

## 25.1.9 CAN0 消息控制寄存器 j (C0MCTLj 寄存器) (j=0 ~ 31)

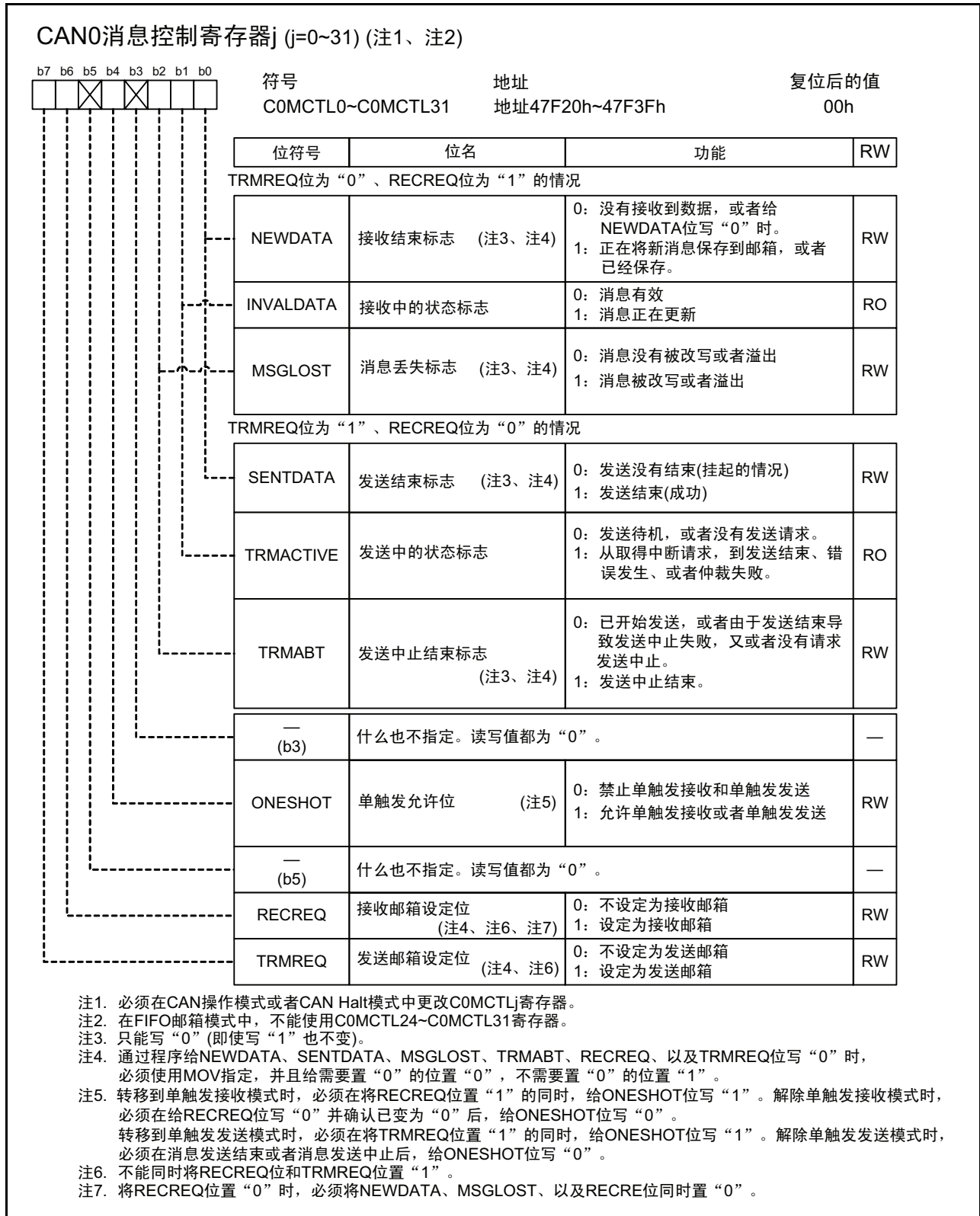


图 25.10 C0MCTLj 寄存器

### 25.1.9.1 NEWDATA 位

正在将新消息保存到邮箱或者已将新消息保存到邮箱时，此位变为“1”。变为“1”的时序，与 INVALIDDATA 位相同。

如果通过程序写“0”，此位就变为“0”。

在相关的 INVALIDDATA 位为“1”的期间，即使通过程序写“0”，NEWDATA 位也不变为“0”。

### 25.1.9.2 SENTDATA 位

在完成发送来自对应邮箱的数据时，此位变为“1”。

如果通过程序写“0”，此位就变为“0”。

将此位置“0”时，必须先将 TRMREQ 位置“0”，再将 SENTDATA 位置“0”。SENTDATA 位和 TRMREQ 位不能同时变为“0”。

从邮箱发送新消息时，必须将 SENTDATA 位置“0”。

### 25.1.9.3 INVALIDDATA 位

消息接收结束后，在邮箱中更新接收到的消息时，此位变为“1”。

在完成消息的保存时，此位变为“0”。在 INVALIDDATA 位为“1”的期间读邮箱时，数据为不定值。

### 25.1.9.4 TRMACTIVE 位

从 CAN 模块开行发送对应邮箱的消息时，此位变为“1”

在以下的情况下，此位变为“0”。

- CAN 模块在 CAN 总线仲裁中失败
- 发生 CAN 总线错误
- 数据发送结束

### 25.1.9.5 MSGLOST 位

在 NEWDATA 位为“1”的期间，邮箱中的消息由于新消息而被改写或者被废弃时，此位变为“1”。在 EOF 的第 6 位结束时，此位变为“1”。

如果通过程序写“0”，此位就变为“0”。

在盖写模式和溢出模式中，在接着 EOF 的第 6 位的 5 个 fCAN（CAN 系统时钟）周期间，即使通过程序写“0”，MSGLOST 位也不变为“0”。

### 25.1.9.6 TRMABT 位

在以下的情况下，此位变为“1”。

- 在产生发送中止请求后，发送中止在发送开始前就结束时。
- 在产生发送中止请求后，CAN 模块在 CAN 总线仲裁中失败或者检测到 CAN 总线错误时。
- 在单触发发送模式（RECREQ 位为“0”、TRMREQ 位为“1”、ONESHOT 位为“1”）中，CAN 模块在 CAN 总线仲裁中失败或者检测到 CAN 总线错误时。

在完成数据发送时，此位不变为“1”，SENTDATA 位变为“1”。

如果通过程序写“0”，此位就变为“0”。

### 25.1.9.7 ONESHOT 位

ONESHOT 位有接收模式和发送模式的 2 种使用方法。

#### 1. 单触发接收模式

如果在接收模式（RECREQ 位为“1”、TRMREQ 位为“0”）中将 ONESHOT 位置“1”，邮箱只接收 1 次消息（在接收 1 次消息结束后，就不作为接收邮箱运行）。NEWDATA 位和 INVALIDDATA 位的运行与通常的接收模式相同。在此模式中，MSGLOST 位不变为“1”。

将 ONESHOT 位置“0”时，必须先给 RECREQ 位写“0”后，并且在确认 RECREQ 位变为“0”后才能进行。

#### 2. 单触发发送模式

如果在发送模式（RECREQ 位为“0”、TRMREQ 位为“1”）中将 ONESHOT 位置“1”，CAN 模块只发送 1 次消息（即使在 CAN 总线错误或者 CAN 总线仲裁失败时，也不重新发送消息）。在发送结束时，SENTDATA 位变为“1”。由于 CAN 总线错误或者 CAN 总线仲裁失败而不能完成发送时，TRMABT 位变为“1”。

将 ONESHOT 位置“0”时，必须在 SENTDATA 位或者 TRMABT 位变为“1”后才能进行。

### 25.1.9.8 RECREQ 位

RECREQ 位选择表 25.11 所示的接收模式。

如果将此位置“1”，对应的邮箱就被设定为接收数据帧或者远程帧。

如果将此位置“0”，对应的邮箱就不被设定为接收数据帧或者远程帧。

在以下期间，硬件保护有效，因此，即使通过程序写“0”，此位也不变为“0”。

硬件保护的开始

- 开始执行接收滤波器的处理（CRC 区的起始）。

硬件保护的解除

- 被指定为接收消息的邮箱，在接收消息被保存到邮箱后，或者发生 CAN 总线错误后（即硬件保护的 maximum 期间：从 CRC 区的起始到 EOF 的第 7 位结束）。
- 其他邮箱在执行接收滤波器的处理后。
- 没有接收邮箱时，在执行接收滤波器的处理后。

在将 RECREQ 位置“1”时，不能将 TRMREQ 位置“1”。

将邮箱的设定从发送更改为接收时，必须在更改前，先中止发送，再将 SENTDATA 位和 TRMABT 位置“0”。

### 25.1.9.9 TRMREQ 位

TRMREQ 位选择表 25.11 所示的发送模式。

如果将此位置“1”，对应的邮箱就被设定为发送数据帧或者远程帧。

如果将此位置“0”，对应的邮箱就不被设定为发送数据帧或者远程帧。

如果为了取消发送请求，将 TRMREQ 位从“1”更改为“0”，TRMABT 位或者 SENTDATA 位就变为“1”。

将 TRMREQ 位置“1”时，不能将 RECREQ 位置“1”。

将邮箱的设定从接收更改为发送时，必须在更改前，先中止接收，再将 NEWDATA 位和 MSGLOST 位置“0”。

## 25.1.10 CAN0 接收 FIFO 控制寄存器 (C0RFCR 寄存器)

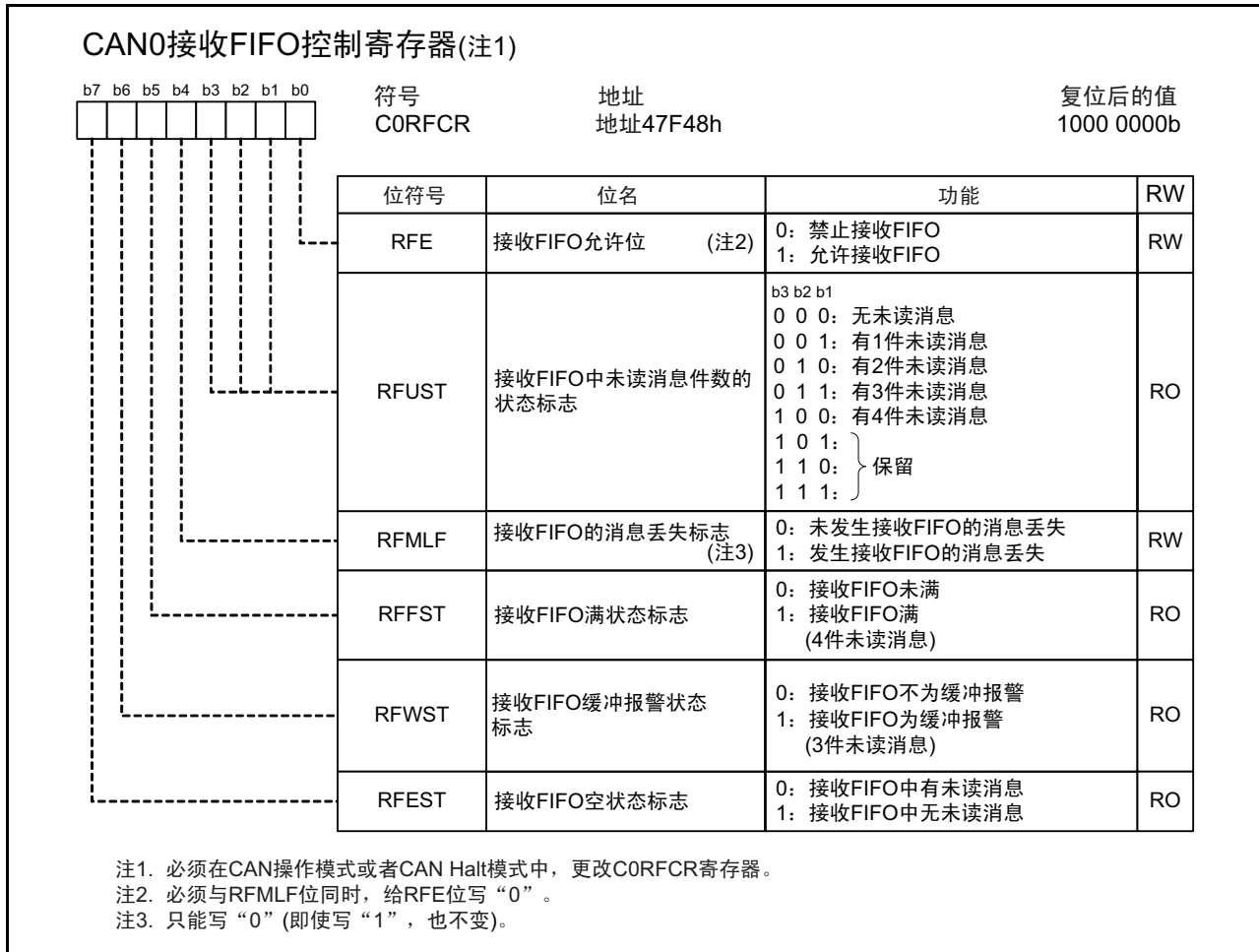


图 25.11 C0RFCR 寄存器

## 25.1.10.1 RFE 位

如果将此位置“1”, 就允许接受FIFO进行接收。

如果将此位置“0”, 就禁止接收FIFO进行接收, 并且变为空状态 (RFEST位为“1”)。

在通常邮箱模式 (C0CTLR寄存器的MBM位为“0”)中, 不能将此位置“1”。

在以下期间, 硬件保护有效, 因此, 即使通过程序写“0”, 此位也不变为“0”。

硬件保护的开始

- 开始执行接收滤波器的处理 (CRC区的起始)。

硬件保护的解除

- 当接收FIFO被指定为接收消息时, 在接收消息被保存到接收FIFO后, 或者发生CAN总线错误后 (即硬件保护的最大期间: 从CRC区的起始到EOF的第7位结束)。
- 接收FIFO没有被指定为接收消息时, 在执行接收滤波器的处理后。

## 25.1.10.2 RFUST 位

RFUST位表示接收FIFO中未读消息的件数。

如果将RFE位置“0”, RFUST位就被初始化为“000b”。

### 25.1.10.3 RFMLF 位

如果在接收 FIFO 为满状态时接收新消息，此位就变为“1”（发生接收 FIFO 的消息丢失）。在 EOF 的第 6 位结束时，变为“1”。

如果通过程序写“0”，此位就变为“0”。

在盖写模式和溢出模式中，如果接收 FIFO 为满状态并且决定接收消息时，受硬件保护，此位在接着 EOF 的第 6 位的 5 个 fCAN（CAN 系统时钟）周期内，不能通过程序变为“0”（未发生接收 FIFO 的消息丢失）。

### 25.1.10.4 RFFST 位

如果接收 FIFO 中的未读消息为 4 件，此位就变为“1”（接收 FIFO 满）。如果接收 FIFO 中的未读消息少于 4 件，此位就变为“0”（接收 FIFO 未满）。如果将 RFE 位置“0”，此位就变为“0”。

### 25.1.10.5 RFWST 位

如果接收 FIFO 中的未读消息为 3 件，此位就变为“1”（接收 FIFO 为缓冲报警）。如果接收 FIFO 中的未读消息少于 3 件或者为 4 件，此位就变为“0”（接收 FIFO 不为缓冲报警）。如果将 RFE 位置“0”，此位就变为“0”。

### 25.1.10.6 RFEST 位

如果接收 FIFO 中没有未读消息，此位就变为“1”（接收 FIFO 中无未读消息）。如果将 RFE 位置“0”，此位就变为“1”。接收 FIFO 中至少有 1 件未读消息时，此位就变为“0”（接收 FIFO 中有未读消息）。

接收 FIFO 邮箱的运行如图 25.12 所示。



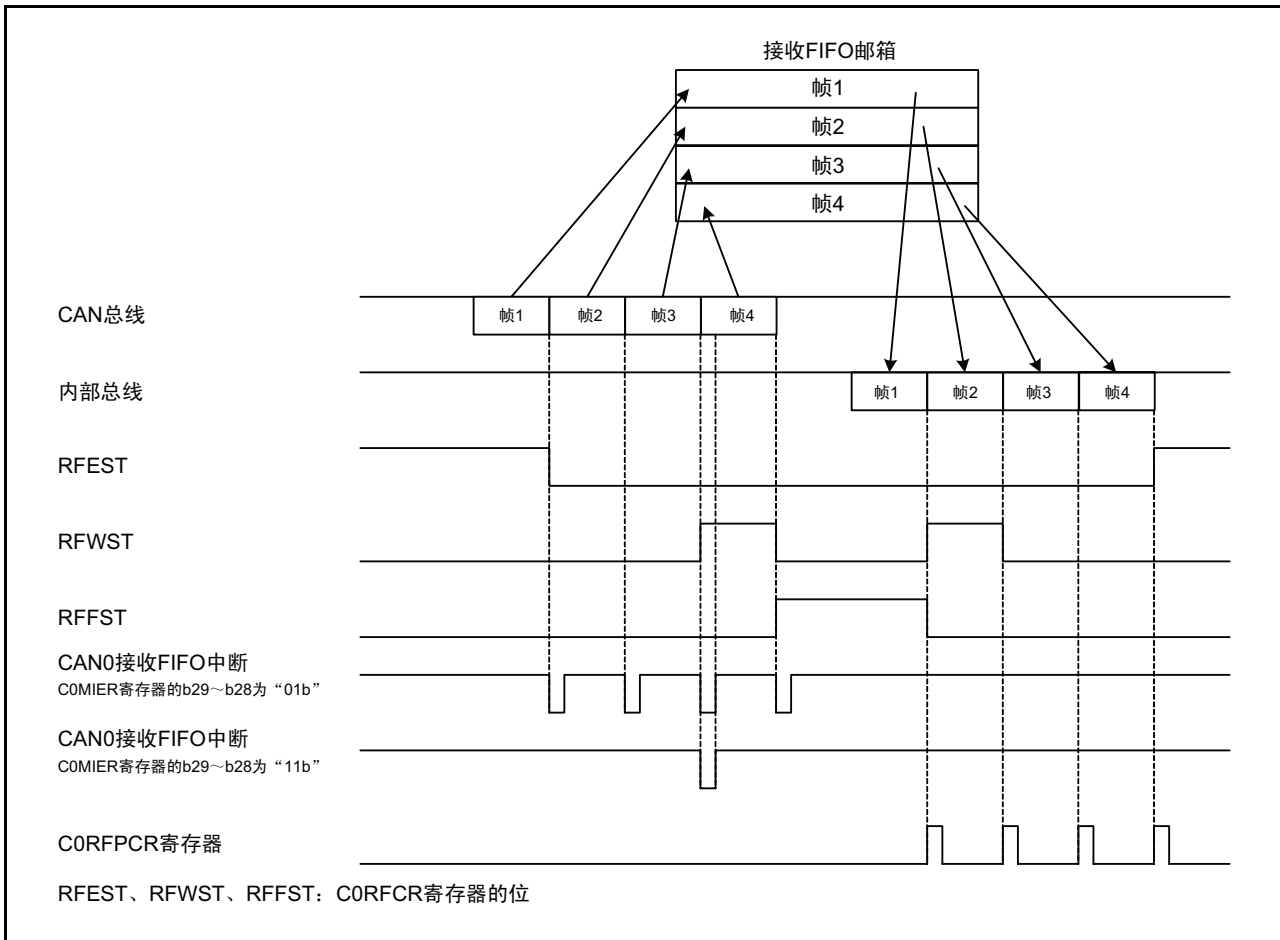


图 25.12 接收 FIFO 邮箱的运行 (C0MIER 寄存器的 b29 ~ b28 为 “01b” 或者 “11b”)

### 25.1.11 CAN0 接收 FIFO 指针控制寄存器 (C0RFPCR 寄存器)



图 25.13 C0RFPCR 寄存器

接收 FIFO 不为空状态时，为了使接收 FIFO 的 CPU 指针移动到下一个邮箱，必须通过程序写 “FFh”。C0RFPCR 寄存器的 RFE 位为 “0”（禁止接收 FIFO）时，不能写此寄存器。

在接收盖写模式中，如果在 RFFST 位为 “1”（接收 FIFO 满）时接收新消息，CAN 指针和 CPU 指针就都移动。在此状态下，如果 RFMLF 位为 “1”，即使通过程序写 C0RFPCR 寄存器，CPU 指针也不移动。

## 25.1.12 CAN0 发送 FIFO 控制寄存器 (C0TFCR 寄存器)

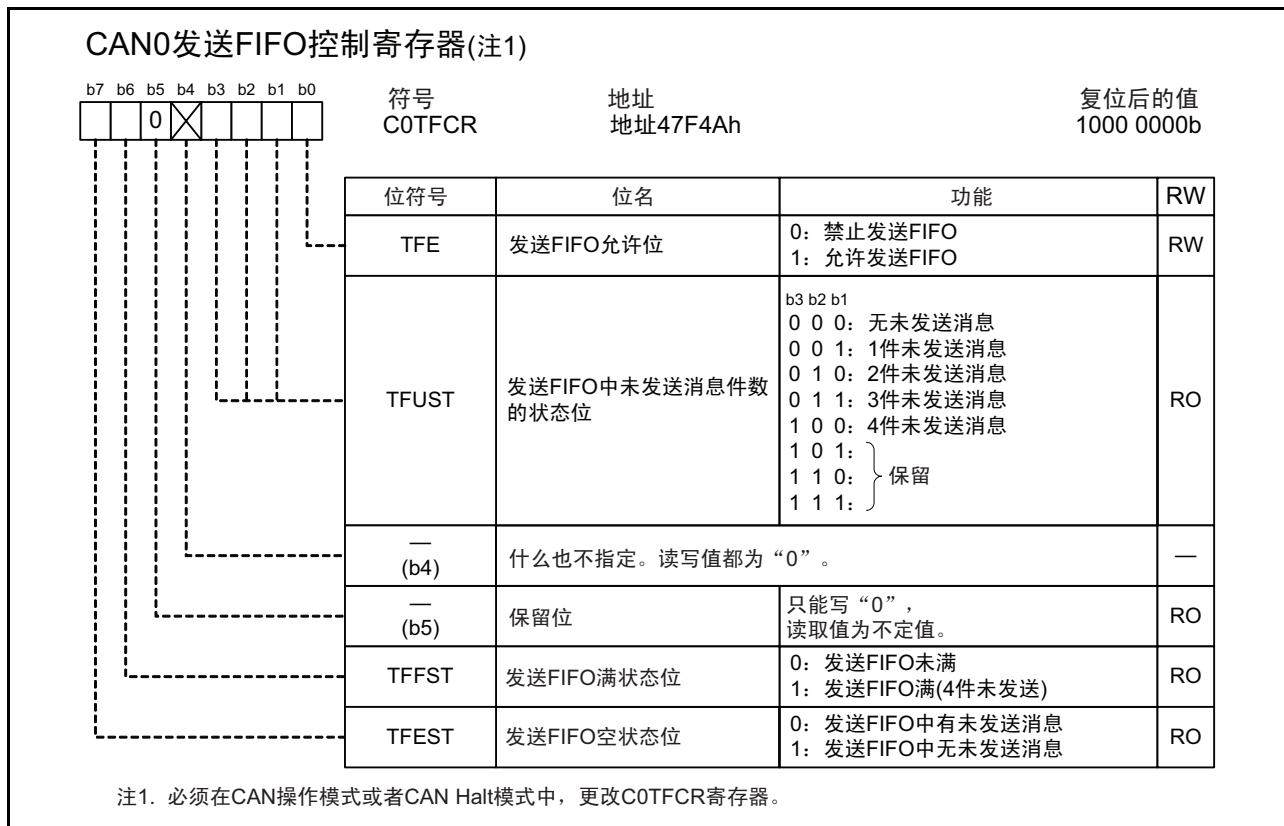


图 25.14 C0TFCR 寄存器

## 25.1.12.1 TFE 位

如果将此位置“1”，就允许发送 FIFO 进行发送。

如果将此位置“0”，发送 FIFO 为空状态（TFEST 位为“1”），并且丢失来自发送 FIFO 的未发送消息，如下所示。

- 没有预定来自发送 FIFO 的下一个发送，并且也没有在进行发送时，为空状态。
- 有预定来自发送 FIFO 的下一个发送，或者已经在进行发送时，在发送结束、CAN 总线错误、仲裁失败、或者转移到 CAN Halt 模式后，变为空状态。

将此位重新置“1”前，必须确认 TFEST 位变为“1”。

将此位置“1”后，必须给 COMB24 寄存器写发送数据。

在通常邮箱模式（C0CTLR 寄存器的 MBM 位为“0”）中，不能将此位置“1”。

## 25.1.12.2 TFUST 位

TFUST 位表示发送 FIFO 中的未发送消息的件数。

将 TFE 位置“0”后，如果发送中止结束或者发送结束，此位就变为“000b”。

## 25.1.12.3 TFFST 位

如果发送 FIFO 中的未发送消息为 4 件，此位就变为“1”（发送 FIFO 满）。如果发送 FIFO 中的未发送消息少于 4 件，此位就变为“0”（发送 FIFO 未满）。如果发送 FIFO 的发送中止结束，此位就变为“0”。

25.1.12.4 TFEST 位

如果发送 FIFO 中没有未发送消息，此位就变为 “1”（发送 FIFO 中无消息）。如果发送 FIFO 的发送中止结束，此位就变为 “1”。

发送 FIFO 中至少有 1 件未发送消息时，此位就变为 “0”（发送 FIFO 中有消息）。

发送 FIFO 邮箱的运行如图 25.15 所示。

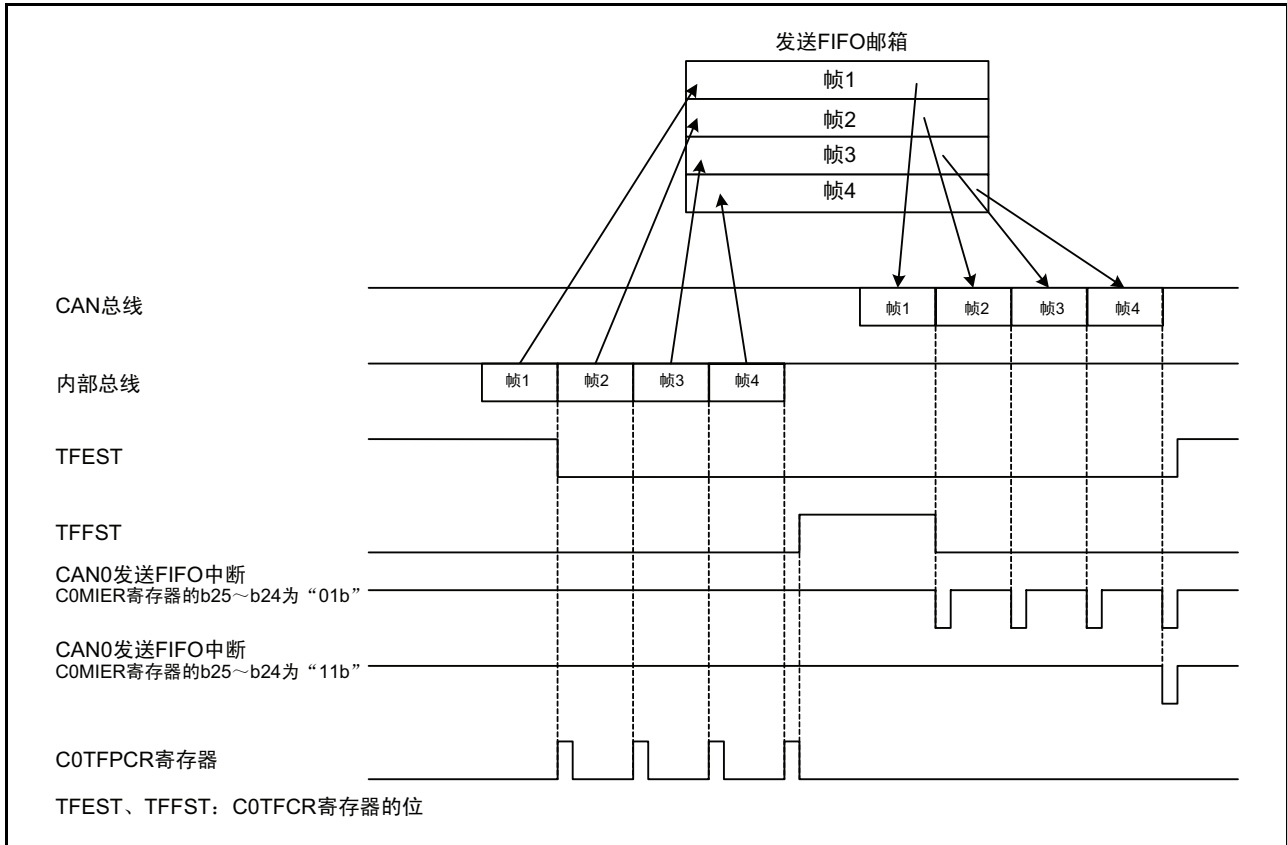


图 25.15 发送 FIFO 邮箱的运行（C0MIER 寄存器的 b25 ~ b24 位为 “01b” 或者 “11b”）

25.1.13 CAN0 发送 FIFO 指针控制寄存器（C0TFPCR 寄存器）

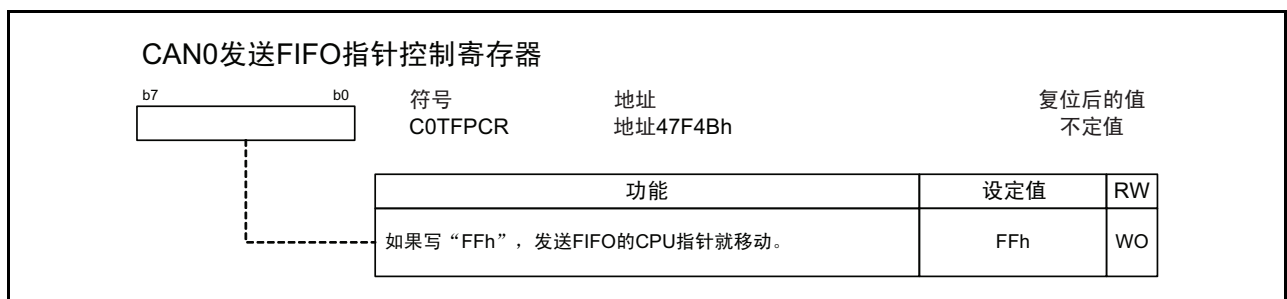


图 25.16 C0TFPCR 寄存器

发送 FIFO 未满载时，为了使发送 FIFO 的 CPU 指针移动到下一个邮箱，必须通过程序写 “FFh”。

C0TFPCR 寄存器的 TFE 位为 “0”（禁止发送 FIFO）时，不能写。

## 25.1.14 CAN0 状态寄存器 (C0STR 寄存器)

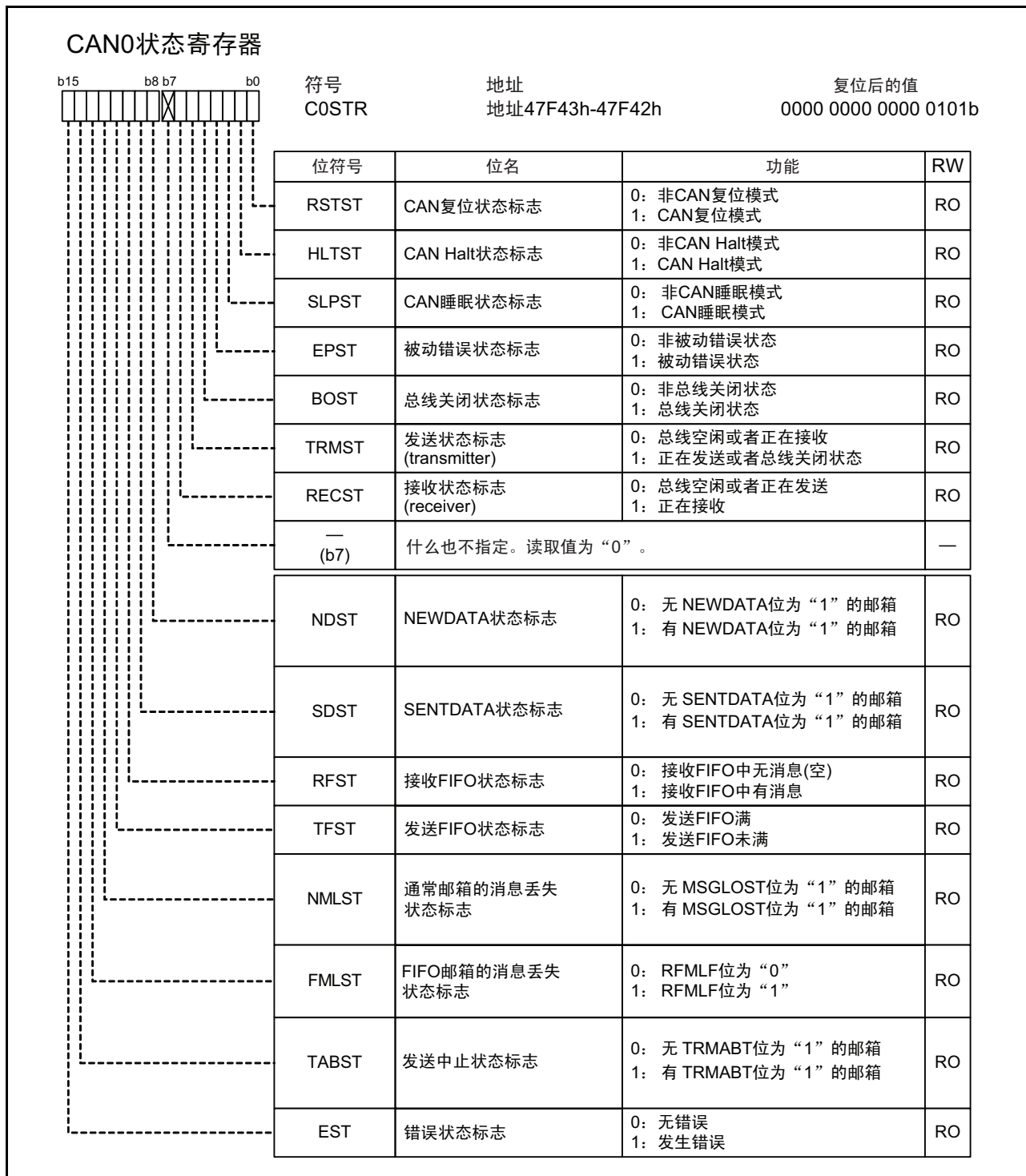


图 25.17 C0STR 寄存器

#### 25.1.14.1 RSTST 位

如果为 CAN 复位模式，此位就变为“1”。

如果为 CAN 复位模式以外的模式，此位就变为“0”。

即使从 CAN 复位模式转移到 CAN 睡眠模式，此位也保持“1”。

#### 25.1.14.2 HLTST 位

如果为 CAN Halt 模式，此位就变为“1”。

如果为 CAN Halt 模式以外的模式，此位就变为“0”。

即使从 CAN Halt 模式转移到 CAN 睡眠模式，此位也保持“1”。

#### 25.1.14.3 SLPST 位

如果为 CAN 睡眠模式，此位就变为“1”。

如果为 CAN 睡眠模式以外的模式，此位就变为“0”。

#### 25.1.14.4 EPST 位

如果 C0TECR 寄存器或者 C0RECR 寄存器的值超过 127，并且 CAN 模块为被动错误状态 ( $128 \leq \text{TEC} < 256$  或者  $128 \leq \text{REC} < 256$ )，此位就变为“1”。如果为被动状态以外的状态，此位就变为“0”。

TEC 是发送错误计数器 (C0TECR 寄存器) 的值，REC 是接收错误计数器 (C0RECR 寄存器) 的值。

#### 25.1.14.5 BOST 位

如果 C0TECR 寄存器的值超过 255，并且 CAN 模块为总线关闭状态 ( $\text{TEC} \geq 256$ )，此位就变为“1”。  
如果为总线关闭状态以外的状态，此位就变为“0”。

#### 25.1.14.6 TRMST 位

如果 CAN 模块为发送节点或者总线关闭状态，此位就变为“1”。如果 CAN 模块为接收节点或者总线空闲状态，此位就变为“0”。

#### 25.1.14.7 RECST 位

如果 CAN 模块为接收节点，此位就变为“1”。如果 CAN 模块为发送节点或者总线空闲状态，此位就为“0”。

#### 25.1.14.8 NDST 位

只要 COMCTLj 寄存器 (j=0 ~ 31) 中的 NEWDATA 位有一个变为“1”，与 COMIER 寄存器的值无关，此位就变为“1”。

如果 NEWDATA 位全部变为“0”，此位就变为“0”。

#### 25.1.14.9 SDST 位

只要 COMCTLj 寄存器 (j=0 ~ 31) 中的 SENTDATA 位有一个变为 “1”，与 COMIER 寄存器的值无关，此位就变为 “1”。

如果 SENTDATA 位全部变为 “0”，此位就变为 “0”。

#### 25.1.14.10 RFST 位

如果接收 FIFO 为空状态以外的状态，此位就变为 “1”。

如果接收 FIFO 为空状态，此位就变为 “0”。

如果为通常邮箱模式，此位就变为 “0”。

#### 25.1.14.11 TFST 位

如果发送 FIFO 为满状态以外的状态，此位就变为 “1”。

如果发送 FIFO 为满状态，此位就变为 “0”。

如果为通常邮箱模式，此位就变为 “0”。

#### 25.1.14.12 NMLST 位

只要 COMCTLj 寄存器 (j=0 ~ 31) 中的 MSGLOST 位有一个变为 “1”，与 COMIER 寄存器的值无关，此位就变为 “1”。

如果 MSGLOST 位全部变为 “0”，此位就变为 “0”。

#### 25.1.14.13 FMLST 位

如果 CORFCR 寄存器的 RFMLF 位变为 “1”，与 COMIER 寄存器的值无关，此位就变为 “1”。

如果 RFMLF 位变为 “0”，此位就变为 “0”。

#### 25.1.14.14 TABST 位

只要 COMCTLj 寄存器 (j=0 ~ 31) 中的 TRMABT 位有一个变为 “1”，与 COMIER 寄存器的值无关，此位就变为 “1”。

如果 TRMABT 位全部变为 “0”，此位就变为 “0”。

#### 25.1.14.15 EST 位

只要通过 COEIFR 寄存器检测到一个错误，与 COEIER 寄存器的值无关，此位就变为 “1”。

如果通过 COEIFR 寄存器没有检测到一个错误，此位就变为 “0”。

## 25.1.15 CAN0 邮箱检索模式寄存器 (COMSMR 寄存器)

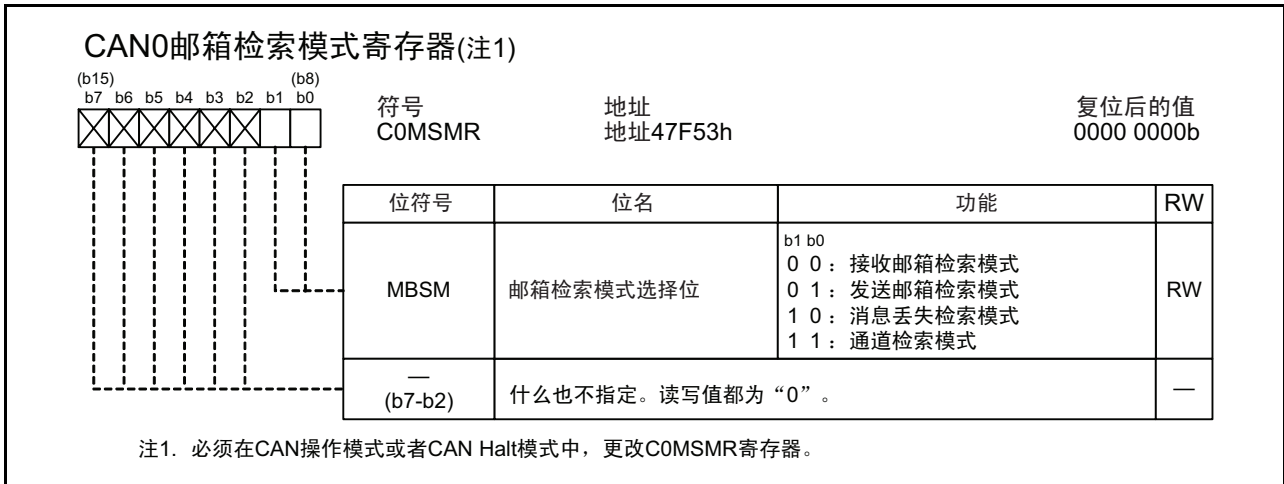


图 25.18 COMSMR 寄存器

## 25.1.15.1 MBSM 位

MBSM 位选择用于邮箱检索功能的检索模式。

此位为“00b”时，为接收邮箱检索模式。此模式中的检索对象位是，COMCTLj 寄存器 (j=0 ~ 31) 的通常邮箱中的 NEWDATA 位和 CORFCR 寄存器的 RFEST 位。

此位为“01b”时，为发送邮箱检索模式。此模式中的检索对象位是 COMCTLj 寄存器的 SENTDATA 位。

此位为“10b”时，为消息丢失检索模式。此模式中的检索对象位是，COMCTLj 寄存器的通常邮箱中的 MSGLOST 位和 CORFCR 寄存器的 RFMLF 位。

此位为“11b”时，为通道检索模式。此模式中的检索对象寄存器是 COCSSR 寄存器。详细内容请参照“25.1.17 CAN0 通道检索支持寄存器 (COCSSR 寄存器)”。

## 25.1.16 CAN0 邮箱检索状态寄存器 (COMSSR 寄存器)

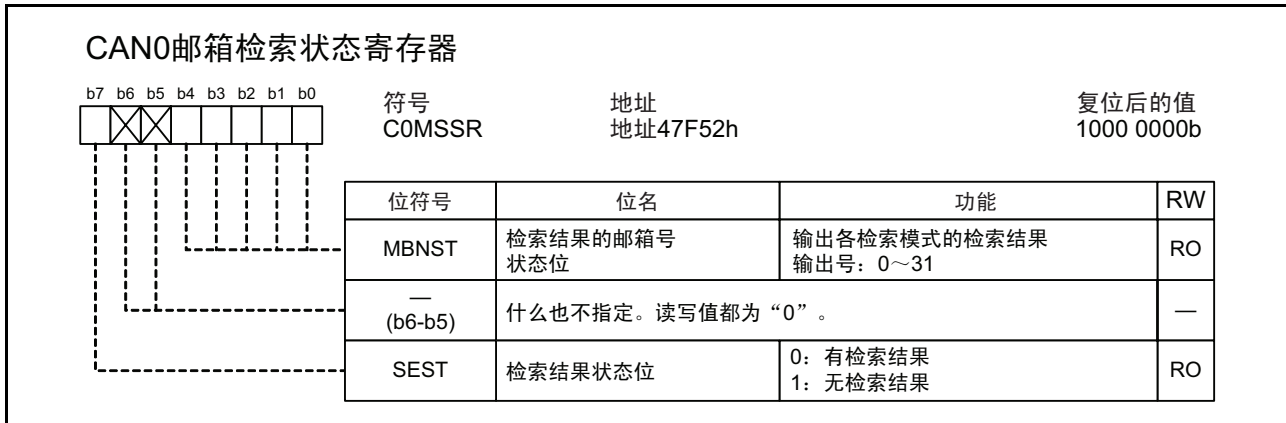


图 25.19 COMSSR 寄存器

## 25.1.16.1 MBNST 位

MBNST 位输出在 COMSMR 寄存器的各模式中检索到的最小邮箱号。

通过接收邮箱检索模式、发送邮箱检索模式、以及消息丢失检索模式输出的检索结果，即邮箱的值，在以下情况下更新。

- 输出的邮箱的 NEWDATA、SENTDATA、或者 MSGLOST 位变为“0”。
- 优先级越高的邮箱的 NEWDATA、SENTDATA、或者 MSGLOST 位变为“1”。

在接收邮箱检索模式和消息丢失检索模式中，如果接收 FIFO 不为空状态，并且在所有的通常邮箱（邮箱 [0] ~ [23] 中，没有未读的接收消息，也没有丢失消息时，就输出接收 FIFO（邮箱 [28]）。

在发送邮箱检索模式中，不输出发送 FIFO（邮箱 [24]）。

FIFO 邮箱模式中的 MBNST 位的运行如表 25.7 所示。

表 25.7 FIFO 邮箱中的 MBNST 位的运行

MBSM 位	邮箱 [24]（发送 FIFO）	邮箱 [28]（接收 FIFO）
00b	不输出邮箱 [24]。	通常邮箱的任意一个 NEWDATA 位都不变为“1”，并且在接收 FIFO 不为空时，输出邮箱 [28]。
01b		不输出邮箱 [28]。
10b		通常邮箱的任意一个 MSGLOST 位都不变为“1”，并且在接收 FIFO 中的 RFMLF 位变为“1”时，输出邮箱 [28]。
11b		不输出邮箱 [28]。

在通道检索模式中，输出通道号。通过程序读 COMSSR 寄存器后，输出下一个目标通道号。

## 25.1.16.2 SEST 位

通过检索全部邮箱也没有对应的邮箱时，此位变为“1”（无检索结果）。

例如，在发送邮箱检索模式中，没有一个 SENTDATA 位为“1”的邮箱时，此位就变为“1”。只要有一个 SENTDATA 位为“1”的邮箱，此位就变为“0”。

SEST 位为“1”时，MBNST 位的值为不定值。



25.1.17 CAN0 通道检索支持寄存器 (C0CSSR 寄存器)



图 25.20 C0CSSR 寄存器

通过 8/3 编码器 (从 LSB 开始按顺序) 编码被设定为 “1” 的 C0CSSR 寄存器的位, 并且向 C0MSSR 寄存器的 MBNST 位输出。

每当通过程序读 C0MSSR 寄存器时, C0MSSR 寄存器就输出被更新的值。

C0CSSR 寄存器和 C0MSSR 寄存器的读写如图 25.21 所示。

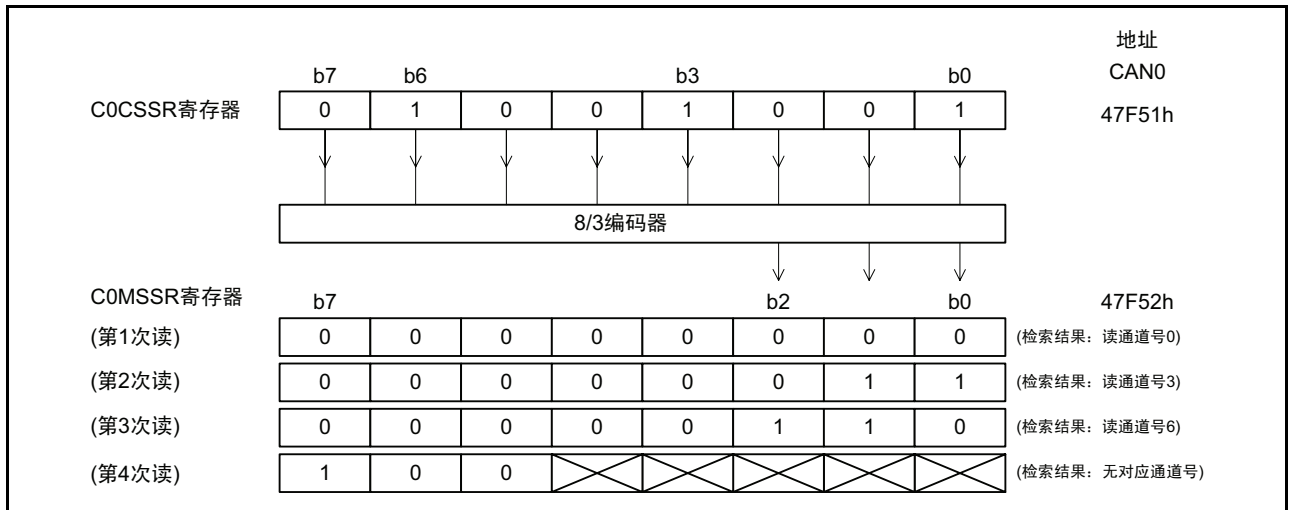


图 25.21 C0CSSR 寄存器和 C0MSSR 寄存器的读写

每当读 C0MSSR 寄存器时, C0CSSR 寄存器的值也被更新。读取值为 8/3 编码器转换前的值。



## 25.1.19 CAN0 错误中断允许寄存器 (C0EIER 寄存器)

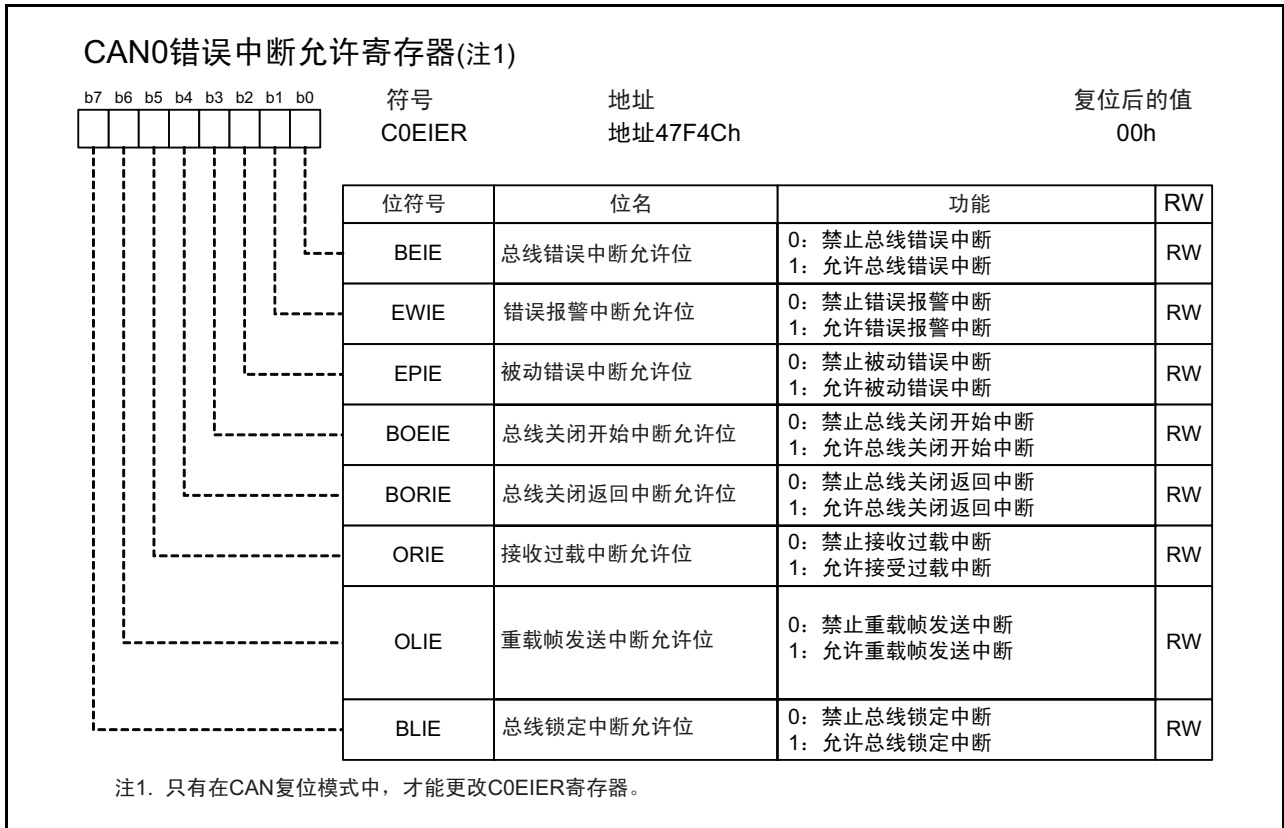


图 25.24 C0EIER 寄存器

C0EIER 寄存器可对 C0EIFR 寄存器的错误中断源各别设定禁止 / 允许的错误中断。

## 25.1.19.1 BEIE 位

如果将此位置“0”，即使在 C0EIFR 寄存器的 BEIF 位变为“1”时，也不产生错误中断请求。  
如果将此位置“1”，就在 BEIF 位变为“1”时，产生错误中断请求。

## 25.1.19.2 EWIE 位

如果将此位置“0”，即使在 C0EIFR 寄存器的 EWIF 位变为“1”时，也不产生错误中断请求。  
如果将此位置“1”，就在 EWIF 位变为“1”时，产生错误中断请求。

## 25.1.19.3 EPIE 位

如果将此位置“0”，即使在 C0EIFR 寄存器的 EPIF 位变为“1”时，也不产生错误中断请求。  
如果将此位置“1”，就在 EPIF 位变为“1”时，产生错误中断请求。

#### 25.1.19.4 BOEIE 位

如果将此位置“0”，即使在 COEIFR 寄存器的 BOEIF 位变为“1”时，也不产生错误中断请求。  
如果将此位置“1”，就在 BOEIF 位变为“1”时，产生错误中断请求。

#### 25.1.19.5 BORIE 位

如果将此位置“0”，即使在 COEIFR 寄存器的 BORIF 位变为“1”时，也不产生错误中断请求。  
如果将此位置“1”，就在 BORIF 位变为“1”时，产生错误中断请求。

#### 25.1.19.6 ORIE 位

如果将此位置“0”，即使在 COEIFR 寄存器的 ORIF 位变为“1”时，也不产生错误中断请求。  
如果将此位置“1”，就在 ORIF 位变为“1”时，产生错误中断请求。

#### 25.1.19.7 OLIE

如果将此位置“0”，即使在 COEIFR 寄存器的 OLIF 位变为“1”时，也不产生错误中断请求。  
如果将此位置“1”，就在 OLIF 位变为“1”时，产生错误中断请求。

#### 25.1.19.8 BLIE

如果将此位置“0”，即使在 COEIFR 寄存器的 BLIF 位变为“1”时，也不产生错误中断请求。  
如果将此位置“1”，就在 BLIF 位变为“1”时，产生错误中断请求。

## 25.1.20 CAN0 错误中断源判断寄存器 (C0EIFR 寄存器)

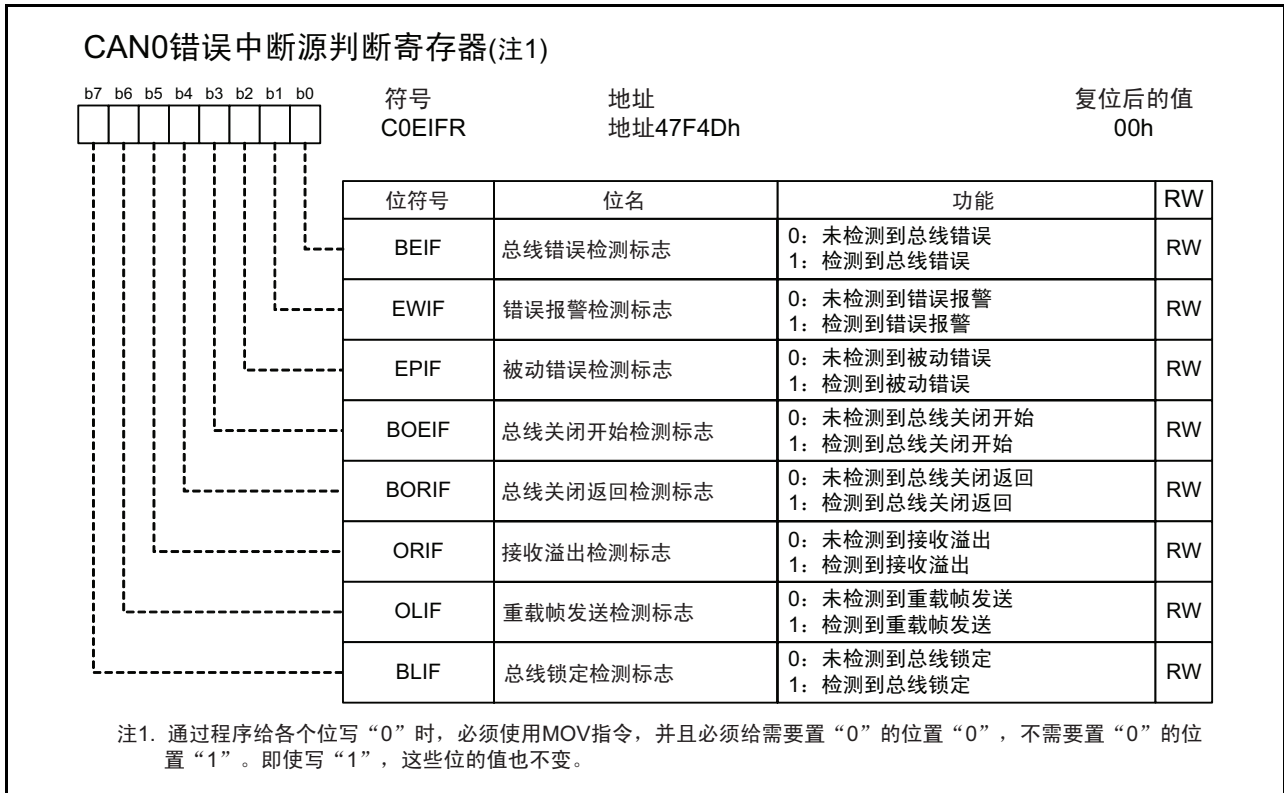


图 25.25 C0EIFR 寄存器

E0EIFR 寄存器在产生各个位对应的现象时, 与 C0EIER 寄存器的设定无关, 对应的位就变为“1”。

将各个位置“0”时, 必须通过程序写“0”。如果变为“1”的时序与通过程序写“0”的时序相同, 该位就变为“1”。

## 25.1.20.1 BEIF 位

如果检测到总线错误, 此位就变为“1”。

## 25.1.20.2 EWIF 位

如果 REC (接收错误计数器) 或者 TEC (发送错误计数器) 的值超过 95, 此位就变为“1”。

只有在 REC 或者 TEC 的值第一次超过 95 时变为“1”。因此, 在 REC 或者 TEC 的值超过 95 的状态下, 通过程序写“0”时, 一旦 REC 或者 TEC 的值低于 95, 直到 REC 或者 TEC 的值重新超过 95 为止, 此位不变为“1”。

## 25.1.20.3 EPIF 位

如果 CAN 错误状态为被动错误状态 (REC 或者 TEC 的值超过 127), 此位就变为“1”。

只有在 REC 或者 TEC 的值第一次超过 127 时变为“1”。因此, 在 REC 或者 TEC 的值超过 127 的状态下, 通过程序写“0”时, 一旦 REC 或者 TEC 的值低于 127, 直到 REC 或者 TEC 的值重新超过 127, 此位不变为“1”。

#### 25.1.20.4 BOEIF 位

如果 CAN 错误状态为总线关闭状态（TEC 的值超过 255），此位就变为“1”。

在 COCTLR 寄存器的 BOM 位为“01b”（在总线关闭开始时自动转移到 CAN Halt 模式），并且 CAN 模块为总线关闭状态时，此位也变为“1”。

#### 25.1.20.5 BORIF 位

如果 CAN 模块在以下的条件下从总线关闭状态正常返回（检测到 128 次 11 个连续的隐性位），此位就变为“1”。

1. COCTLR 寄存器的 BOM 位为“00b”时
2. BOM 位为“10b”时
3. BOM 位为“11b”时

另外，如果 CAN 模块在以下的条件下从总线关闭状态返回，此位不变为“1”。

1. 将 COCTLR 寄存器的 CANM 位置“01b”（CAN 复位模式）时
2. 将 COCTLR 寄存器的 RBOC 位置“1”（从总线关闭强制返回）时
3. BOM 位为“01b”时
4. BOM 位为“11b”并且在发生正常返回前将 CANM 位置“10b”（CAN Halt 模式）时

取决于 BOM 位设定值的 BOEIF 位和 BORIF 位的运行如表 25.8 所示。

表 25.8 取决于 BOM 位设定值的 BOEIF 位和 BORIF 位的运行

BOM 位	BOEIF 位	BORIF 位
00b	向总线关闭状态转移时，变为“1”。	从总线关闭状态返回时，变为“1”。
01b		不变为“1”。
10b		从总线关闭状态返回时，变为“1”。
11b		在 CANM 位变为“10b”（CAN Halt 模式）前，已从总线关闭状态正常返回，就变为“1”。

#### 25.1.20.6 ORIF 位

如果发生接收溢出，此位就变为“1”。

在盖写模式中，此位不变为“1”。在盖写模式中，如果产生盖写条件，就产生接收结束中断请求，并且此位不变为“1”。

在通常邮箱模式的溢出模式中，如果在邮箱 [0] ~ [31] 中的任意一个邮箱发生溢出，此位就变为“1”。

在 FIFO 邮箱模式的溢出模式中，如果在邮箱 [0] ~ [23] 中的任意一个邮箱或者在接收 FIFO 发生溢出，此位就变为“1”。

#### 25.1.20.7 OLIF 位

如果在 CAN 模块进行发送或者接收时检测到重载帧的发送条件，此位就变为“1”。

#### 25.1.20.8 BLIF 位

在 CAN 模块为 CAN 操作模式期间，如果在 CAN 总线上检测到 32 个连续的显性位，此位就变为“1”。变为“1”后，在以下任意一种情况下重新进行检测。

- 将此位从“1”置“0”后，检测到隐性位。
- 将此位从“1”置“0”后，先转移到 CAN 复位模式或者 CAN Halt 模式，再转移到 CAN 操作模式。

## 25.1.21 CAN0 接收错误计数寄存器 (C0RECR 寄存器)

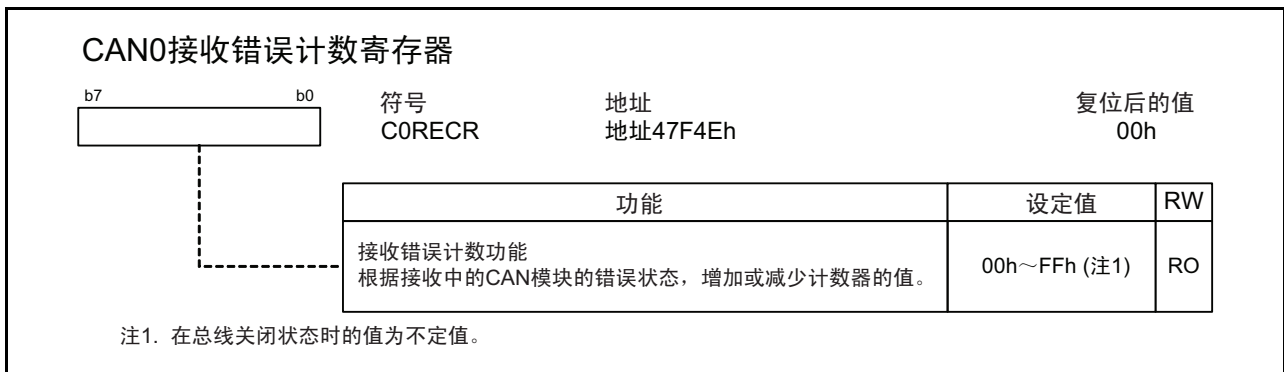


图 25.26 C0RECR 寄存器

C0RECR 寄存器表示接受错误计数器的值。

有关接收错误计数器的增减条件，请参照“CAN 规格 (ISO11898-1)”。

## 25.1.22 CAN0 发送错误计数寄存器 (C0TECR 寄存器)



图 25.27 C0TECR 寄存器

C0TECR 寄存器表示发送错误计数器的值。

有关发送错误计数器的增减条件，请参照“CAN 规格 (ISO11898-1)”。

## 25.1.23 CAN0 错误码保存寄存器 (C0ECSR 寄存器)

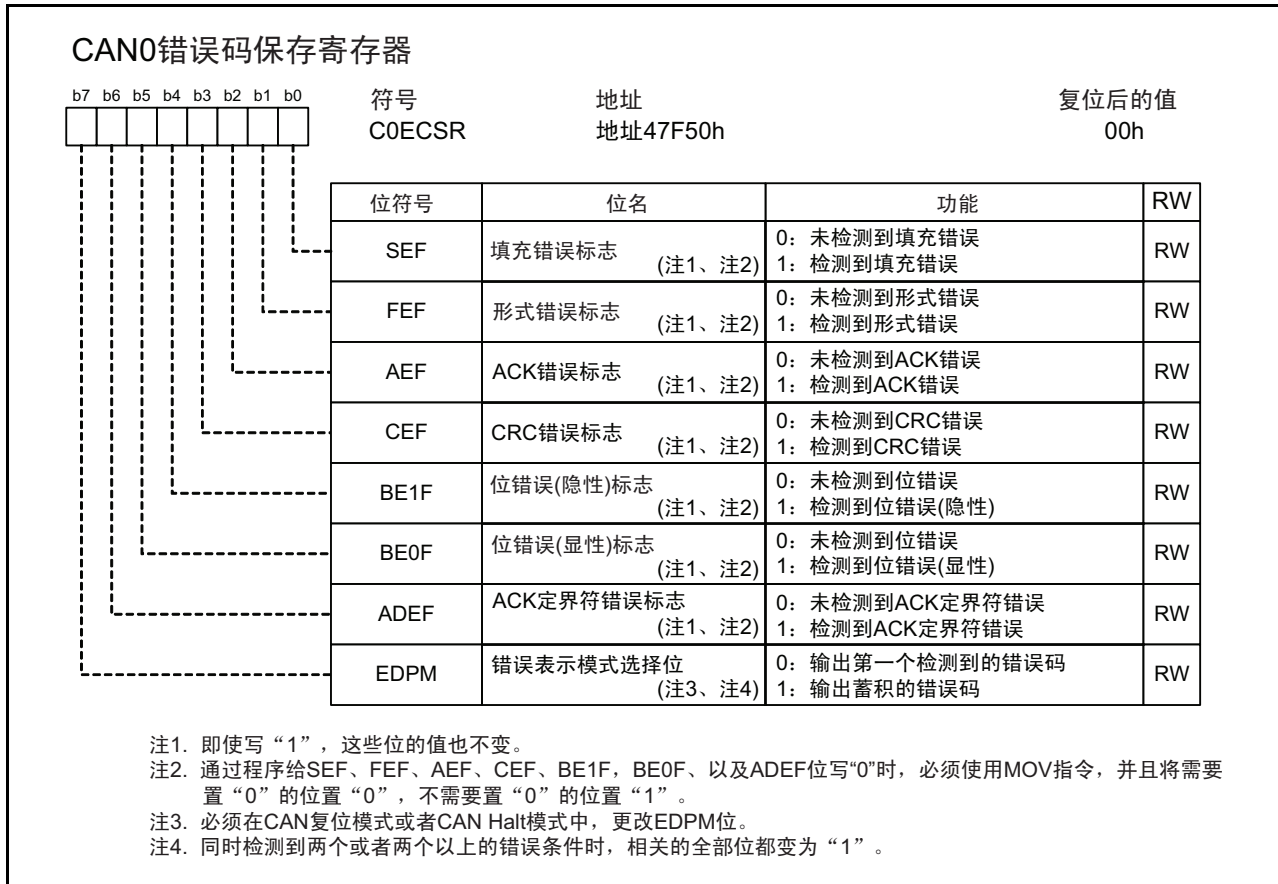


图 25.28 C0ECSR 寄存器

C0ECSR 寄存器可用于监视在 CAN 总线上发生的错误。

确认各错误的发生条件时，请参照“CAN 规格 (ISO11898-1)”。

将 EDPM 位以外的各个位置“0”时，必须通过程序写“0”。如果各个位变为“1”的时序与通过程序写“0”的时序相同，该位就变为“1”。

## 25.1.23.1 SEF 位

如果检测到填充错误，此位就变为“1”。

## 25.1.23.2 FEF 位

如果检测到形式错误，此位就变为“1”。

## 25.1.23.3 AEF 位

如果检测到 ACK 错误，此位就变为“1”。

## 25.1.23.4 CEF 位

如果检测到 CRC 错误，此位就变为“1”。



## 25.1.23.5 BE1F 位

如果检测到隐性位错误，此位就变为“1”。

## 25.1.23.6 BE0F 位错误

如果检测到显性位错误，此位就变为“1”。

## 25.1.23.7 ADEF 位

如果通过发送中的定界符检测到形式错误，此位就变为“1”。

## 25.1.23.8 EDPM 位

EDPM 位选择 COECSR 寄存器的输出模式。

如果将此位置“0”，COECSR 寄存器就输出第一个错误码。

如果将此位置“1”，COECSR 寄存器就输出蓄积的错误码。

## 25.1.24 CAN0 时戳寄存器（C0TSR 寄存器）



图 25.29 C0TSR 寄存器

如果读 C0TSR 寄存器，就能读取当时的时戳计数器（16 位自由运行计数器）的值。

时戳计数器的基准时钟值为倍增 1 位时间后的值，通过 COCTLR 寄存器的 TSPS 位设定。

时戳计数器在 CAN 睡眠模式和 CAN Halt 模式中停止，在 CAN 复位模式被初始化。

接收消息被保存到接收邮箱时的时戳计数器的值，被保存到 COMBj 寄存器（j=0~31）的 TSL、TSH。

## 25.1.25 CAN0 测试控制寄存器 (C0TCR 寄存器)

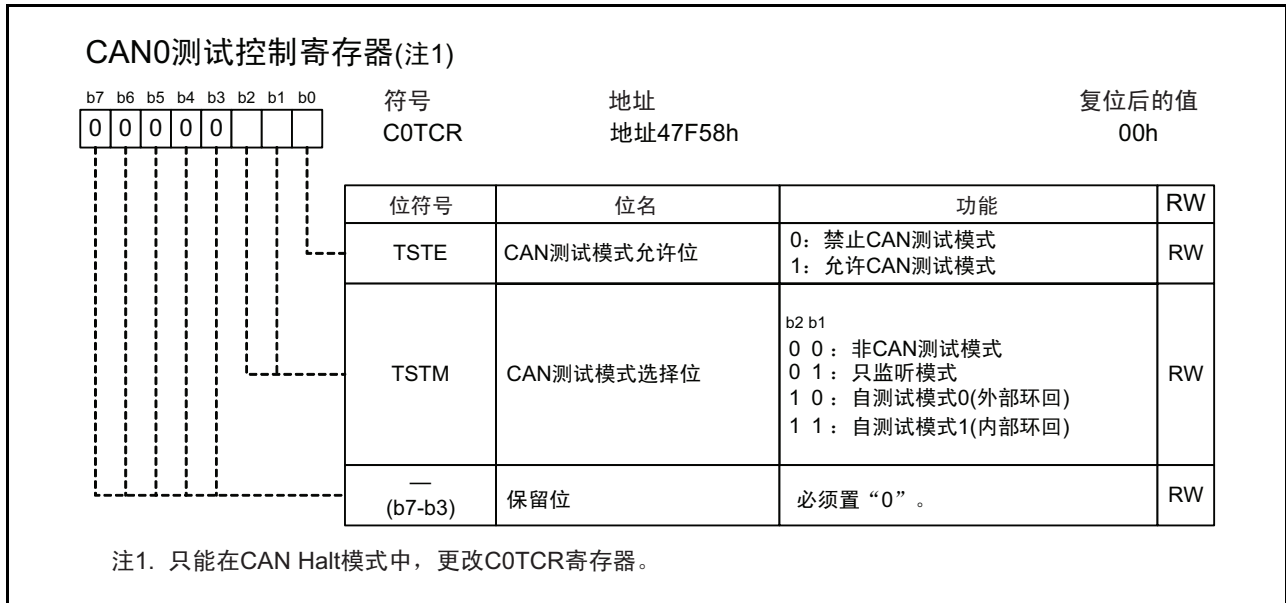


图 25.30 C0TCR 寄存器

## 25.1.25.1 TSTE 位

如果将此位置“0”, 就禁止 CAN 测试模式。  
如果将此位置“1”, 就允许 CAN 测试模式。

## 25.1.25.2 TSTM 位

TSTM 位选择 CAN 测试模式。  
以下说明各 CAN 测试模式的详细内容。

## 25.1.25.3 只监听模式

在 CAN 规格 (ISO11898-1) 中, 推荐选项的总线监视模式。在只监听模式中, 能接收有效数据帧和有效远程帧。只向 CAN 总线上发送隐性位, 不发送 ACK 位、重载标志、以及主动错误标志。

只监听模式能在波特率检测中使用。  
在只监听模式中, 任何一个邮箱都不能请求发送。  
选择只监听模式时的连接如图 25.31 所示。

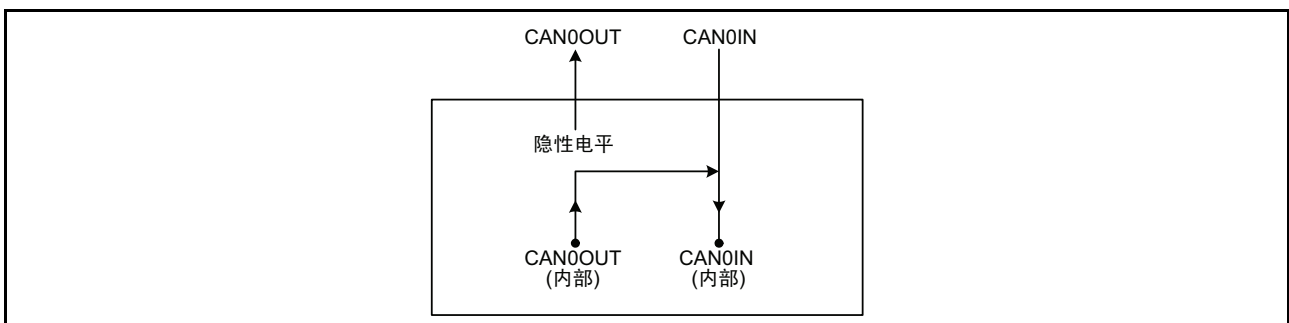


图 25.31 选择只监听模式时的连接

#### 25.1.25.4 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。

在自测试模式 0 中，经由 CAN 收发器就能将已发送的消息作为接受到的消息处理，并且将已发送的消息保存到接收邮箱。由于是在外部独立执行的功能，因此生成 ACK 位。

必须将 CAN0OUT/CAN0IN 引脚连接到 CAN 收发器。

选择自测试模式 0 时的连接如图 25.32 所示。

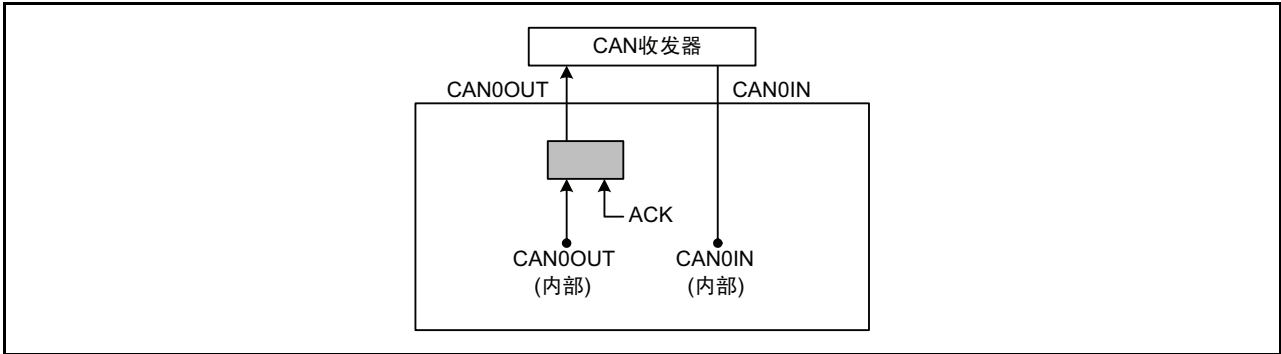


图 25.32 选择自测试模式 0 时的连接

#### 25.1.25.5 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，将已发送的消息作为接收到的消息处理，并且将已发送的消息保存到接收邮箱。由于是在外部独立执行的功能，因此生成 ACK 位。

在自测试模式 1 中，从内部 CAN0OUT 引脚向 CAN0IN 引脚进行内部反馈。忽视外部 CAN0IN 引脚的输入值。外部 CAN0OUT 引脚只输出隐性位。CAN0OUT/CAN0IN 不需要连接到 CAN 总线或者其他外部器件。

选择自测试模式 1 时的连接如图 25.33 所示。

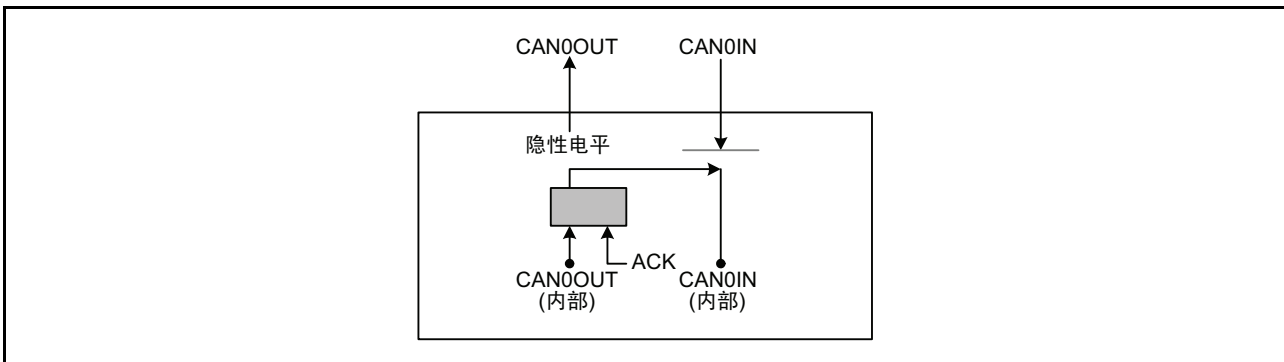


图 25.33 选择自测试模式 1 时的连接

## 25.2 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 25.34 所示。

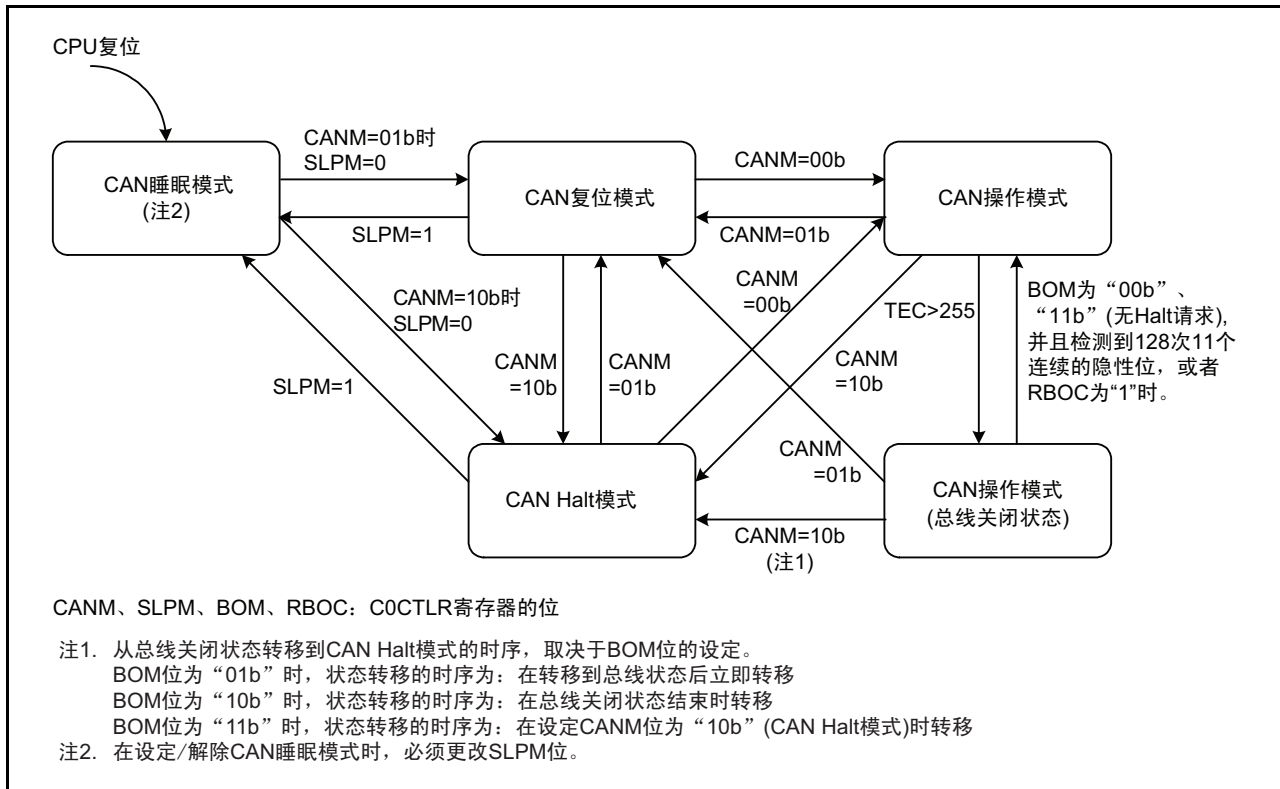


图 25.34 CAN 运行模式之间的转移

### 25.2.1 CAN 复位模式

CAN 复位模式进行 CAN 通信的设定。

如果将 COCTRL 寄存器的 CANM 位置 “01b”，就为 CAN 复位模式。此时，COSTR 寄存器的 RSTST 位变为 “1”。到 RSTST 位变为 “1” 前，不能更改 CANM 位。

从 CAN 复位模式转移到其他模式前，必须设定 COBCR 寄存器。

在转移到 CAN 复位模式后，以下各寄存器被初始化为各复位后的值，并且在 CAN 复位模式中保持初始值。

- COMCTLj 寄存器 (j=0~31)
- COSTR 寄存器 (SLPST 位和 TFST 位除外)
- COEIFR 寄存器
- CORECR 寄存器
- COTEER 寄存器
- COTSR 寄存器
- COMSSR 寄存器
- COMSMR 寄存器
- CORFCR 寄存器
- COTFCR 寄存器
- COTCR 寄存器
- COECSR 寄存器 (EDPM 位除外)

在转移到 CAN 复位模式后，以下的寄存器保持之前的值。

- COCLKR 寄存器
- COCTRL 寄存器
- COSTR 寄存器 (SLPST 位和 TFST 位)
- COMIER 寄存器
- COEIER 寄存器
- COBCR 寄存器
- COCSSR 寄存器
- COECSR 寄存器 (仅 EDPM 位)
- COMBj 寄存器
- COMKR0~COMKR7 寄存器
- COFIDCR0、COFIDCR1 寄存器
- COMKIVLR 寄存器
- COAFSR 寄存器
- CORFPCR 寄存器
- COTFPCR 寄存器

### 25.2.2 CAN Halt 模式

CAN Halt 模式设定邮箱和测试模式。

如果将 COCTLR 寄存器的 CANM 位置 “10b”，就为 CAN Halt 模式。此时， COSTR 寄存器的 HLTST 位变为 “1”。到 HLTST 位变为 “1” 前，不能更改 CANM 位。

发送或者接收时的状态转移条件，请参照 “表 25.9 在 CAN 复位模式和 CAN Halt 模式中的运行”。

在向 CAN Halt 模式转移时，除 COSTR 寄存器的 RSTST 位、HLTST 位、SLPST 位以外的位和其他寄存器都不变。

在 CAN Halt 模式中，不能更改 COCLKR 寄存器、COCTLR 寄存器（CANM 位和 SLPM 位除外）、以及 COEIER 寄存器。在 CAN 测试模式中，只有在为了作为自动波特率检测使用而选择了只监听模式时，才能在 CAN Halt 模式中更改 COBCR 寄存器。

表 25.9 在 CAN 复位模式和 CAN Halt 模式中的运行

模式	接收	发送	总线关闭
CAN 复位模式	不等到消息接收结束，就转移到 CAN 复位模式。	等到消息发送结束后，转移到 CAN 复位模式（注 1、注 4）	不等到总线关闭返回结束，就转移到 CAN 复位模式。
CAN Halt 模式	等到消息接收结束后，转移到 CAN Halt 模式（注 2、注 3）	等到消息发送结束后，转移到 CAN Halt 模式（注 1、注 4）	<p><b>【BOM 位为 “00b” 的情况】</b> 只在总线关闭返回后，接受程序的 Halt 请求。</p> <p><b>【BOM 位为 “01b” 的情况】</b> 不等到总线关闭返回结束，就自动转移到 CAN Halt 模式（与程序的 Halt 请求无关）。</p> <p><b>【BOM 位为 “10b” 的情况】</b> 等到总线关闭返回结束后，自动转移到 CAN Halt 模式（与程序的 Halt 请求无关）。</p> <p><b>【BOM 位为 “11b” 的情况】</b> 如果在总线关闭期间有程序的 Halt 请求，就转移到 CAN Halt 模式（不等到总线关闭返回结束）。</p>

BOM 位：COCTLR 寄存器的位

- 注 1. 请求发送多个消息时，就在第一个发送结束后进行模式转移。如果在挂起接收期间请求 CAN 复位模式，就在总线空闲时、或者在下一个发送结束时、又或者在 CAN 模块变为接收时，进行模式转移。
- 注 2. CAN 总线在隐性电平被锁定时，如果监视 COEIFR 寄存器的 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，如果在接收中发生 CAN 总线错误，就转移到 CAN Halt 模式。
- 注 4. 在请求 CAN 复位模式或者 CAN Halt 模式后，如果在发送中发生 CAN 总线错误或者 CAN 仲裁失败，就转移到请求的 CAN 模式。

### 25.2.3 CAN 睡眠模式

CAN 睡眠模式通过停止向 CAN 模块供给时钟来降低消耗电流。在执行 MCU 的硬件复位或者软件复位后，从 CAN 睡眠模式开始运行。

如果将 COCTLR 寄存器的 SLPM 位置 “1”，就为 CAN 睡眠模式。此时， COSTR 寄存器的 SLPST 位就变为 “1”。到 SLPST 位变为 “1” 前，不能更改 SLPM 位的值。向 CAN 睡眠模式转移时，其他寄存器不变。

必须在 CAN 复位模式或者 CAN Halt 模式中更改 SLPM 位。不能在 CAN 睡眠模式中更改除 SLPM 位以外的其他寄存器。允许读操作。

如果将 SLPM 位置 “0”，就解除 CAN 睡眠模式。从 CAN 睡眠模式返回时，其他的寄存器不变。

### 25.2.4 CAN 操作模式（总线关闭状态除外）

CAN 操作模式执行 CAN 通信。

如果将 COCTLR 寄存器的 CANM 位置“00b”，就为 CAN 操作模式。此时，COSTR 寄存器的 RSTST 位和 HLTST 位变为“0”。到 RSTST 位和 HLTST 位变为“0”前，不能更改 CANM 位的值。

在转移到 CAN 操作模式后，如果检测到 11 个连续的隐性位，CAN 模块就变为以下的状态。

- CAN 模块成为可通信的网络上的主动节点，并且可接收/发送 CAN 消息。
- 接收错误计数器、发送错误计数器等执行 CAN 总线的错误监视处理。

根据 CAN 总线的状态，CAN 操作模式变为以下 3 个子模式中的一种。

- 空闲模式：没有进行发送/接收的状态。
- 接收模式：其他节点正在接收已发送的 CAN 消息。
- 发送模式：发送 CAN 消息。在选择自测试模式 0（COTCR 寄存器的 TSTM 位为“10b”）或者自测试模式 1（TSTM 位为“11b”）时，同时接收自节点发送的消息。

CAN 操作模式的子模式如图 25.35 所示。

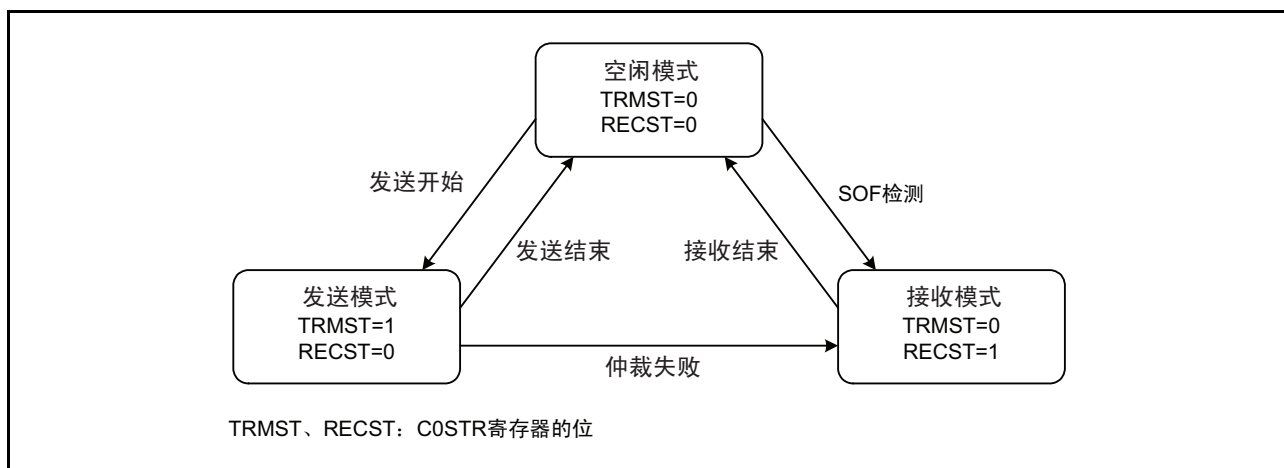


图 25.35 CAN 操作模式的子模式

### 25.2.5 CAN 操作模式（总线关闭状态）

根据 CAN 规格的发送 / 接收错误计数器的增减规则，转移到总线关闭状态。

从总线关闭状态返回有以下几种情况。另外，在总线关闭状态下，除 C0STR、C0EIFR、C0RECR、C0TECR 以及 C0TSR 寄存器以外，相关寄存器的值不变。

#### (1) C0CTLR 寄存器的 BOM 位为 “00b” 的情况（普通模式）

在从总线关闭状态返回结束后，转移到主动错误状态，并且能够进行 CAN 通信。此时，C0EIFR 寄存器的 BORIF 位变为 “1”（检测到总线关闭返回）。

#### (2) 将 C0CTLR 寄存器的 RBOC 位置 “1” 的情况（从总线关闭强制返回）

如果变为总线关闭状态、并且 RBOC 位变为 “1”，就转移到主动错误状态，而且在检测到 11 个连续的隐性位后，又能够进行 CAN 通信。此时，BORIF 位不变为 “1”。

#### (3) BOM 位为 “01b” 的情况（在总线关闭开始时自动转移到 CAN Halt 模式）

如果进入总线关闭状态，就变为 CAN Halt 模式。此时，BORIF 位不变为 “1”。

#### (4) BOM 位为 “10b” 的情况（在总线关闭结束时自动转移到 CAN Halt 模式）

在从总线关闭返回结束后，就变为 CAN Halt 模式。此时，BORIF 位变为 “1”。

#### (5) 在 BOM 位为 “11b” 时（通过程序转移到 CAN Halt 模式）

如果在总线关闭状态期间，将 CANM 位设定为 “10b”（CAN Halt 模式），就为 CAN Halt 模式。此时，BORIF 位不变为 “1”。

如果在总线关闭状态期间，没有将 CANM 位设定为 “10b”，就进行与 (1) 相同的运行。



### 25.3 CAN 通信速度的设定

以下对 CAN 通信速度的设定进行说明。

#### 25.3.1 CAN 时钟的设定

R32C/117 群中有 CAN 时钟选择电路。

可通过 C0CLKR 寄存器的 CCLKS 位和 C0BCR 寄存器的 BPR 位设定 CAN 时钟。

CAN 时钟发生电路的框图如图 25.36 所示。

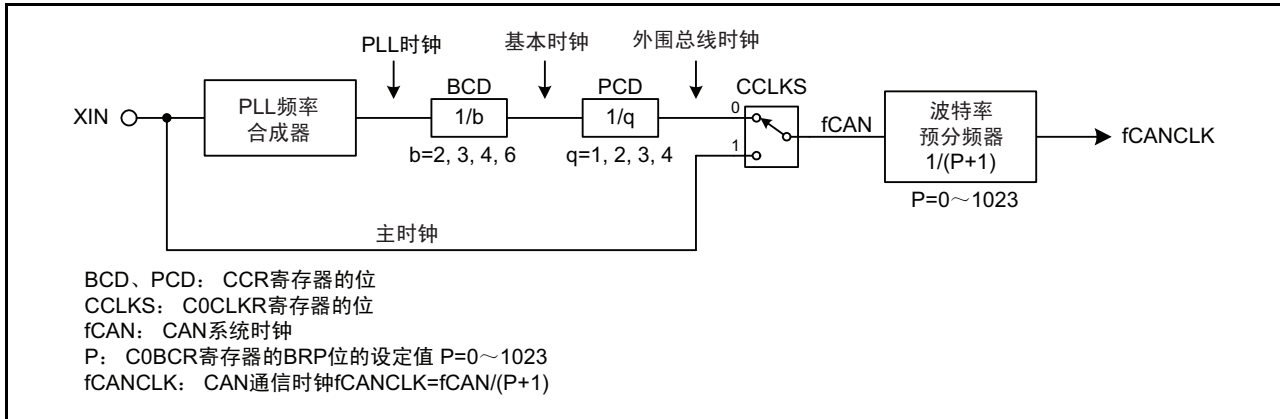


图 25.36 CAN 时钟发生电路的框图

#### 25.3.2 位时序的设定

位时间是发送或者接收 1 位消息的时间，由以下 3 段构成。

位时序图如图 25.37 所示。

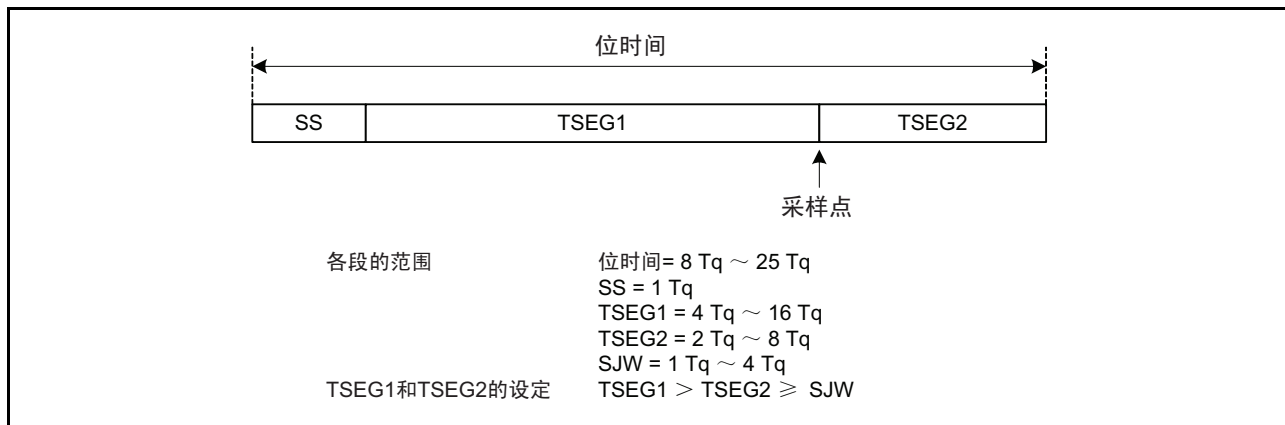


图 25.37 位时序图

### 25.3.3 位速率

位速率取决于 fCAN（CAN 时钟）、波特率预分频器的分频值、以及 1 位时间的 Tq 数。

$$\text{位速率[bps]} = \frac{f_{CAN}}{\text{波特率预分频器的分频值(注1)} \times 1 \text{位时间的Tq数}} = \frac{f_{CANCLK}}{1 \text{位时间的Tq数}}$$

注1. 波特率预分频器的预分频值= P+1(P=0~1023)

P: COBCR寄存器的BRP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	32MHz		24MHz		20MHz		16MHz		8MHz	
位速率	Tq 数	P+1	Tq 数	P+1	Tq 数	P+1	Tq 数	P+1	Tq 数	P+1
1Mbps	8Tq	4	8Tq	3	10Tq	2	8Tq	2	8Tq	1
	16Tq	2			20Tq	1	16Tq	1		
500kbps	8Tq	8	8Tq	6	10Tq	4	8Tq	4	8Tq	2
	16Tq	4	16Tq	3	20Tq	2	16Tq	2	16Tq	1
250kbps	8Tq	16	8Tq	12	10Tq	8	8Tq	8	8Tq	4
	16Tq	8	16Tq	6	20Tq	4	16Tq	4	16Tq	2
83.3kbps	8Tq	48	8Tq	36	8Tq	30	8Tq	24	8Tq	12
	16Tq	24	16Tq	18	10Tq	24	16Tq	12	16Tq	6
					16Tq	15				
					20Tq	12				
33.3kbps	8Tq	120	8Tq	90	8Tq	75	8Tq	60	8Tq	30
	10Tq	96	10Tq	72	10Tq	60	10Tq	48	10Tq	24
	16Tq	60	16Tq	45	20Tq	30	16Tq	30	16Tq	15
	20Tq	48	20Tq	36			20Tq	24	20Tq	12

## 25.4 邮箱和屏蔽寄存器的结构

具有 32 个相同结构的邮箱。

COMBj (j=0 ~ 31) 的结构如图 25.38 所示。

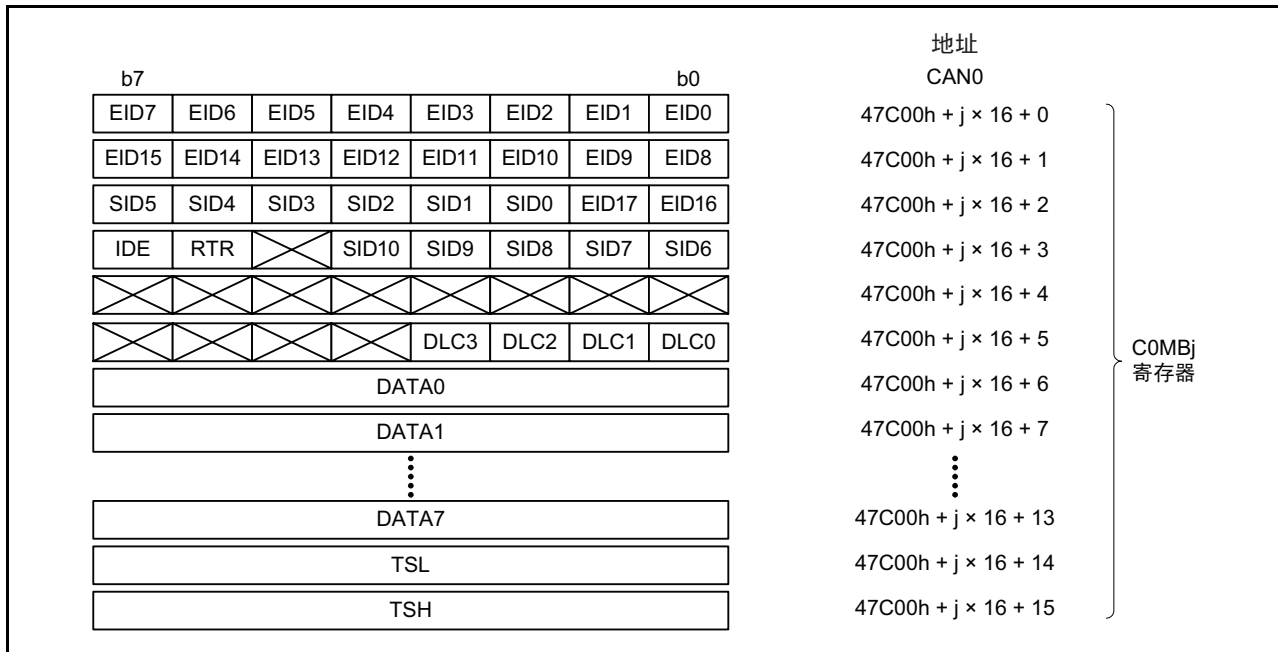


图 25.38 COMBj 寄存器的结构 (j=0 ~ 31)

具有 8 个相同结构的屏蔽寄存器。

COMKRk 寄存器 (k=0 ~ 7) 的结构如图 25.39 所示。

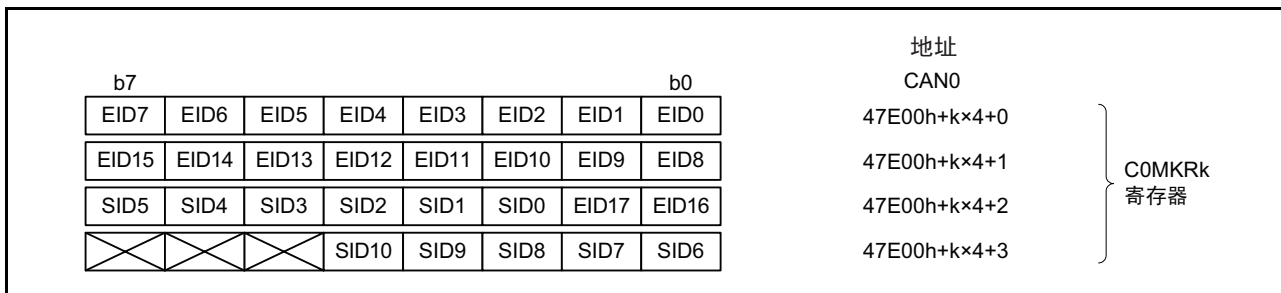


图 25.39 COMKRk 寄存器的结构 (k=0 ~ 7)

具有 2 个相同结构的 FIFO 接收 ID 比较寄存器。

C0FIDCRn 寄存器 (n=0,1) 的结构如图 25.40 所示。

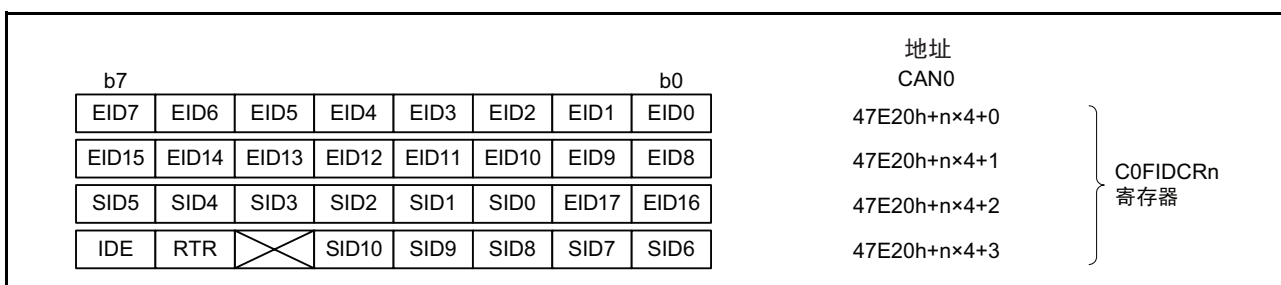


图 25.40 C0FIDCRn 寄存器的结构 (n=0,1)

## 25.5 接收滤波器的处理和屏蔽功能

接收滤波器允许邮箱接受一定范围的多个 ID 消息。

COMKR0 ~ COMKR7 寄存器能屏蔽标准 ID 和 29 位的扩展 ID。

- COMKR0 寄存器对应邮箱 [0] ~ [3]
- COMKR1 寄存器对应邮箱 [4] ~ [7]
- COMKR2 寄存器对应邮箱 [8] ~ [11]
- COMKR3 寄存器对应邮箱 [12] ~ [15]
- COMKR4 寄存器对应邮箱 [16] ~ [19]
- COMKR5 寄存器对应邮箱 [20] ~ [23]
- COMKR6 寄存器在通常邮箱模式中对邮箱 [24] ~ [27]，在 FIFO 邮箱模式中对接收 FIFO 邮箱 [28] ~ [31]
- COMKR7 寄存器在通常邮箱模式中对邮箱 [28] ~ [31]，在 FIFO 邮箱模式中对接收 FIFO 邮箱 [28] ~ [31]

COMKIVLR 寄存器禁止对各个邮箱进行单独的接收滤波器处理。

COMBj 寄存器 (j=0 ~ 31) 的 IDE 位在 COCTLR 寄存器的 IDFM 位为 “10b” 时 (混合 ID 模式) 有效。

COMBj 寄存器的 RTR 位选择数据帧或者远程帧。

在 FIFO 邮箱模式中，通常邮箱 (邮箱 [0] ~ [23]) 使用 COMKR0 ~ COMKR5 寄存器中对应的 1 个寄存器进行接收滤波器处理，而接收 FIFO 邮箱 (邮箱 [28] ~ [31]) 使用 COMKR6、COMKR7 这两个寄存器进行接收滤波器处理。

另外，接收 FIFO 使用 C0FIDCR0、C0FIDCR1 这两个寄存器进行 ID 比较。接收 FIFO 的 COMB28 ~ COMB31 寄存器的 EID、SID、RTR、IDE 位无效。分别通过 2 个逻辑和的结果进行接收滤波器的处理，因此，接收 FIFO 可接收 2 个范围的 ID。

COMKIVLR 寄存器对接收 FIFO 无效。

将 C0FIDCR0 寄存器和 C0FIDCR1 寄存器的 IDE 位分别设定为标准 ID 和扩展 ID 时，能接收标准 ID 和扩展 ID 这两种 ID 格式。

将 C0FIDCR0 寄存器和 C0FIDCR1 寄存器的 RTR 位分别设定为数据帧和远程帧时，能接收数据帧和远程帧这两种帧。

不需要组合 2 个范围的 ID 时，必须将 FIFO ID 和屏蔽寄存器设定为相同的屏蔽值和相同的 ID。

屏蔽寄存器和邮箱的对应关系如图 25.41、接收滤波器的处理如图 25.42 所示。

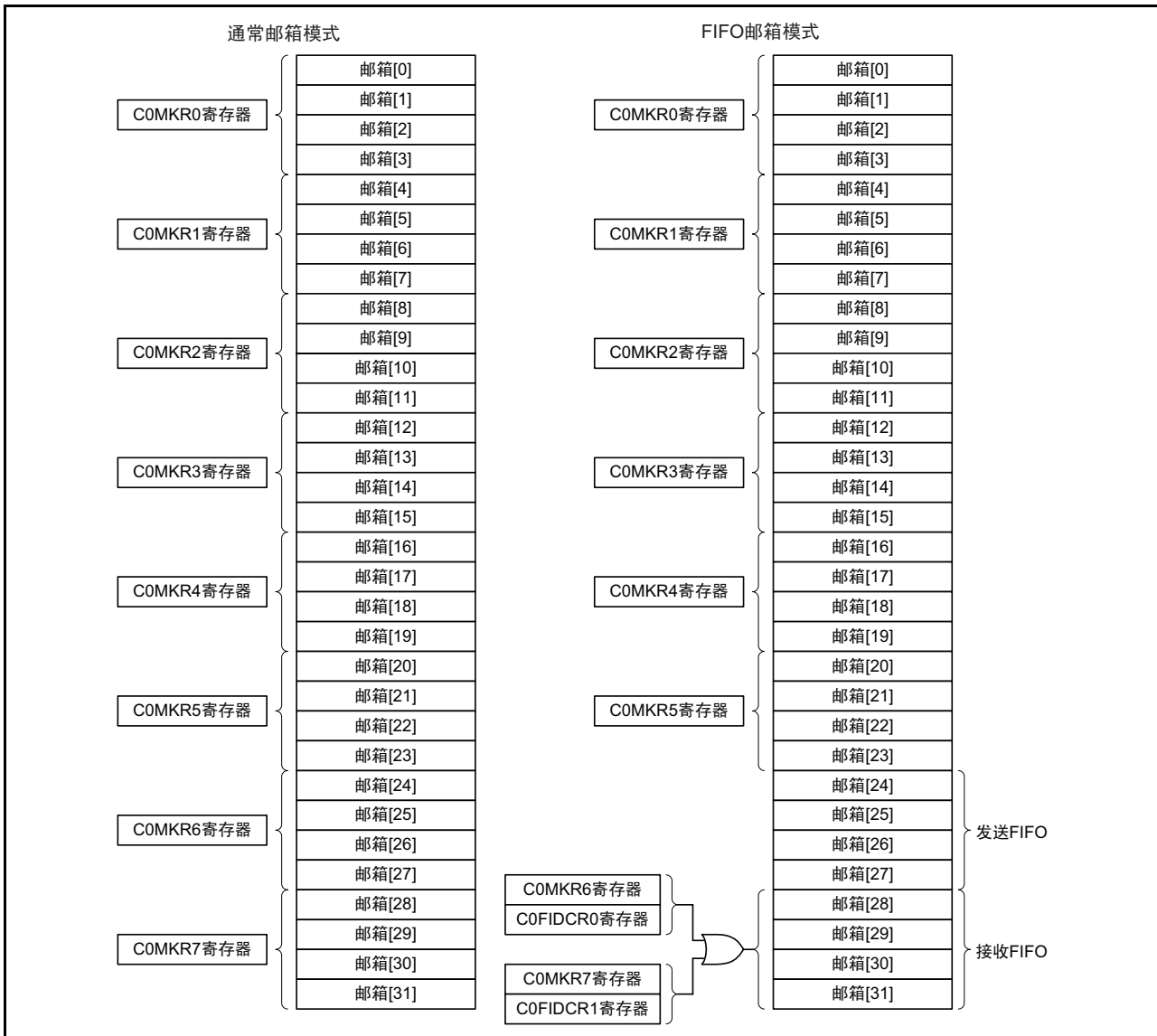


图 25.41 屏蔽寄存器和邮箱的对应关系

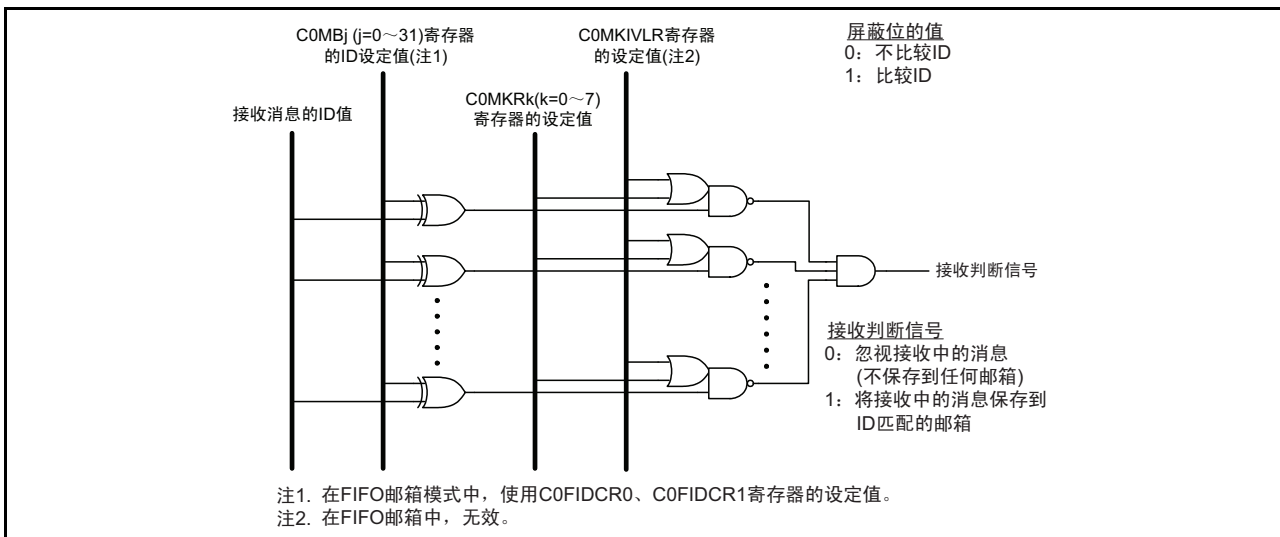


图 25.42 接收滤波器的处理

## 25.6 接收和发送

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

TRMREQ	RECREQ	ONESHOT	邮箱的通信模式
0	0	0	不能使用邮箱，或者正在中止发送。
0	0	1	只有在单触发模式中，中止了被编程的邮箱的发送或者接收时，才能设定。
0	1	0	作为数据帧或者远程帧的接收邮箱设定。
0	1	1	作为数据帧或者远程帧的单触发接收邮箱设定。
1	0	0	作为数据帧或者远程帧的发送邮箱设定。
1	0	1	作为数据帧或者远程帧的单触发发送邮箱设定。
1	1	0	不能设定。
1	1	1	不能设定。

TRMREQ、RECREQ、ONESHOT: COMCTLj 寄存器的位 (j=0 ~ 31)

将邮箱作为接收邮箱或者单触发接收邮箱设定时，必须注意以下几点。

1. 在将邮箱作为接收邮箱或者单触发接收邮箱设定前，必须将 COMCTLj 寄存器 (j=0 ~ 31) 置“00h”。
2. 根据设定的接收模式和接收滤波器的处理结果，将接收消息保存到第一个与条件匹配的邮箱。保存接收到的消息的邮箱，邮箱号越小优先级越高。
3. 在 CAN 操作模式中，如果发送了与邮箱（被设定为接收消息）的 ID/屏蔽组合匹配的消息，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。

将邮箱设定为发送邮箱或者单触发发送邮箱时，必须注意以下一点。

1. 在将邮箱作为发送邮箱或者单触发发送邮箱设定前，必须将 COMCTLj 寄存器置“00h”，并且确认没有在进行中止处理。

### 25.6.1 接收

接收数据帧时的运行例子（盖写模式的情况）如图 25.43 所示。

该例子是在接收 2 个连续的与 COMCTL0 寄存器的接收条件匹配的 CAN 消息时，盖写第 1 个消息的运行例子。

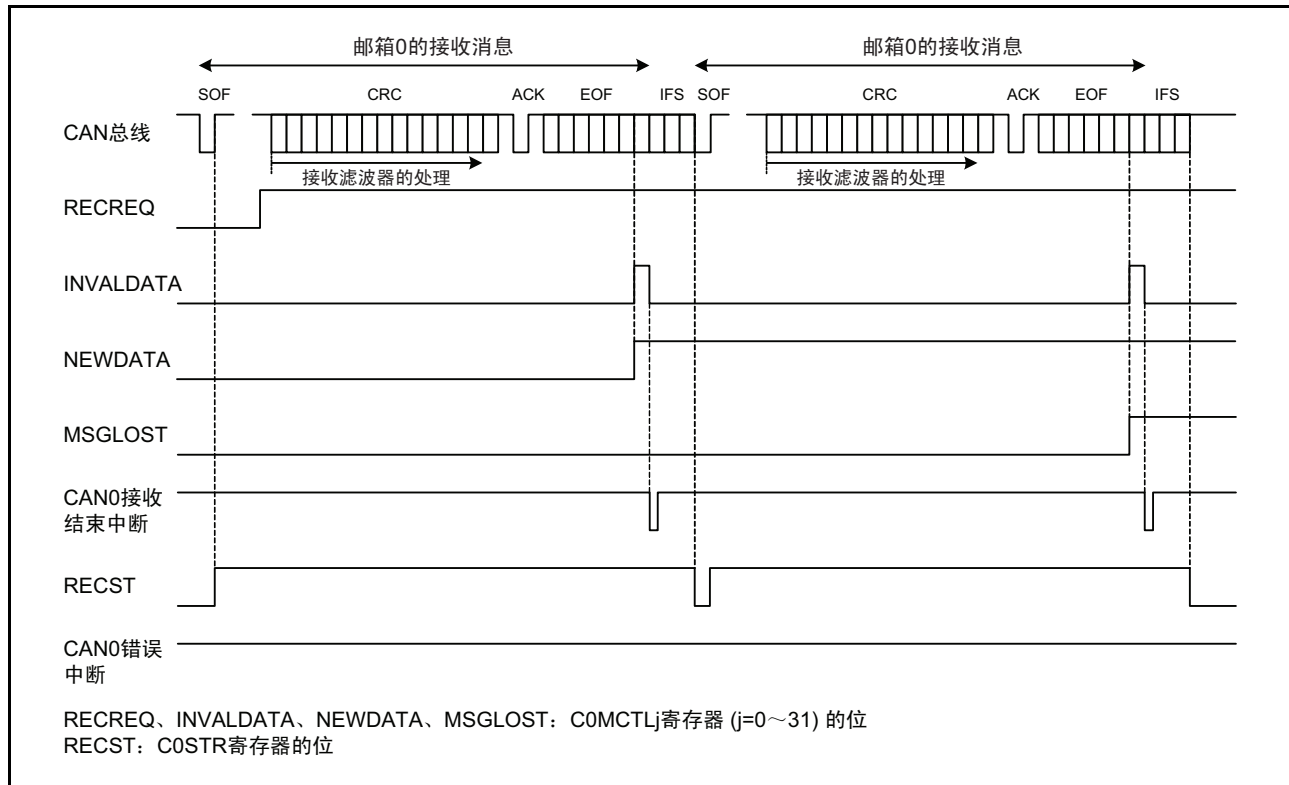


图 25.43 接收数据帧时的运行例子（盖写模式的情况）

1. 在CAN总线上检测到SOF时，如果没有开始向CAN模块发送的消息，C0STR寄存器的RECST位就变为“1”（接收中）。
2. 为了选择接收邮箱，从CRC区的起始开始进行接收滤波器的处理。
3. 如果消息接收结束，接收邮箱的COMCTLj寄存器（j=0~31）的NEWDATA位变为“1”（正在更新新消息，或者已保存到邮箱）。同时，COMCTLj寄存器的INVALIDDATA位变为“1”（正在更新消息），并且在全部消息被传送到该邮箱后，INVALIDDATA位变为“0”（消息有效）。
4. 在接收邮箱的COMIER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INVALIDDATA位变为“0”，就发生该中断（CAN0接收结束中断）。
5. 从邮箱读取消息后，必须通过程序将NEWDATA位置“0”。
6. 在盖写模式中，如果NEWDATA位仍然被设定为“1”的邮箱完成下一个CAN消息的接收，COMCTLj寄存器的MSGLOST位就变为“1”（消息被盖写）。新接收到的消息被传送到邮箱。CAN0接收结束中断请求的产生与4.相同。

接收数据帧时的运行例子（溢出模式的情况）如图 25.44 所示。

该例子是在接收 2 个连续的与 COMCTL0 寄存器的接收条件匹配的 CAN 消息时，废弃第 2 个消息的运行例子。

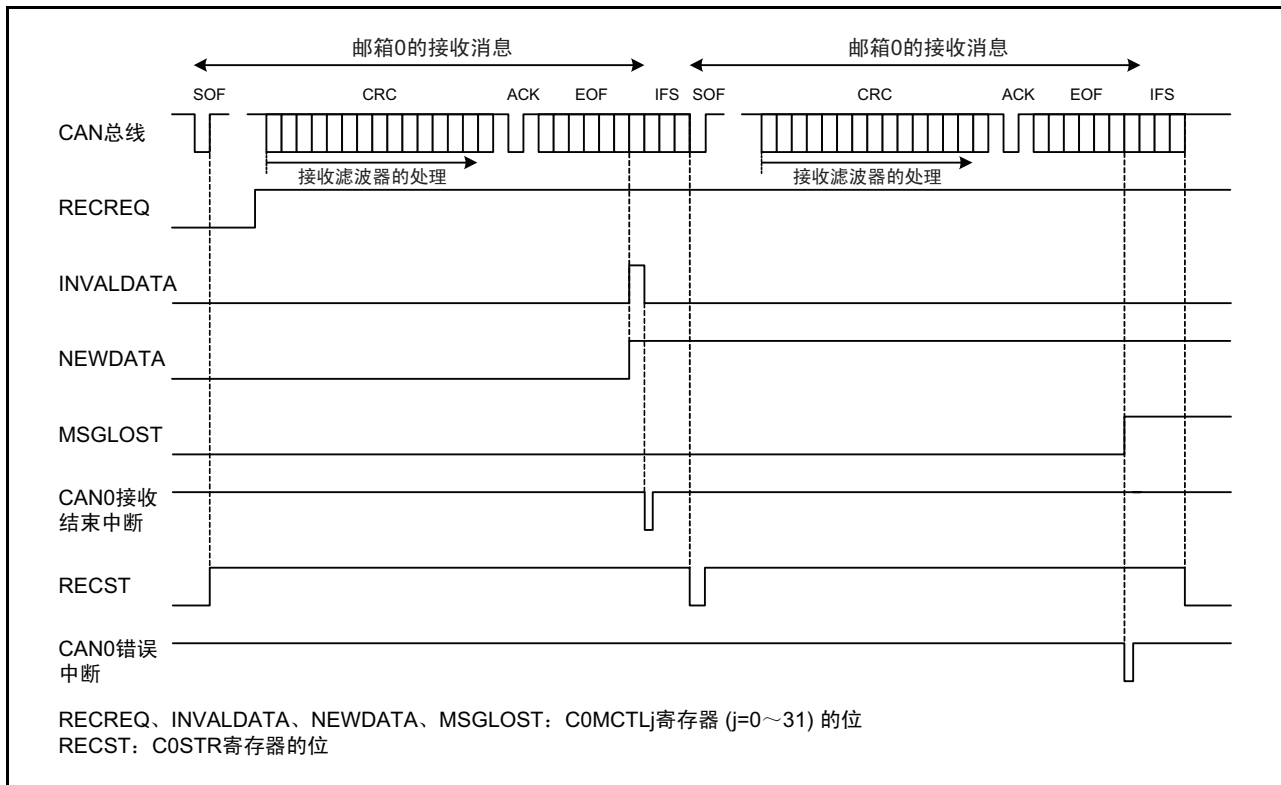


图 25.44 接收数据帧时的运行例子（溢出模式的情况）

1. ~ 5. 与盖写模式相同。
6. 在溢出模式中，如果在 NEWDATA 位被设定为“0”前，就结束下一个 CAN 消息的接收时，COMCTLj 寄存器 (j=0~31) 的 MSGLOST 位就变为“1”（消息溢出）。新接收到的消息被废弃，并且在 COEIER 寄存器的对应中断允许位为“1”（允许中断）时，产生 CAN0 错误中断请求。



## 25.6.2 发送

发送数据帧时的运行例子如图 25.45 所示。

此例子是将已设定的消息发送到 COMCTL0、COMCTL1 寄存器时的运行例子。

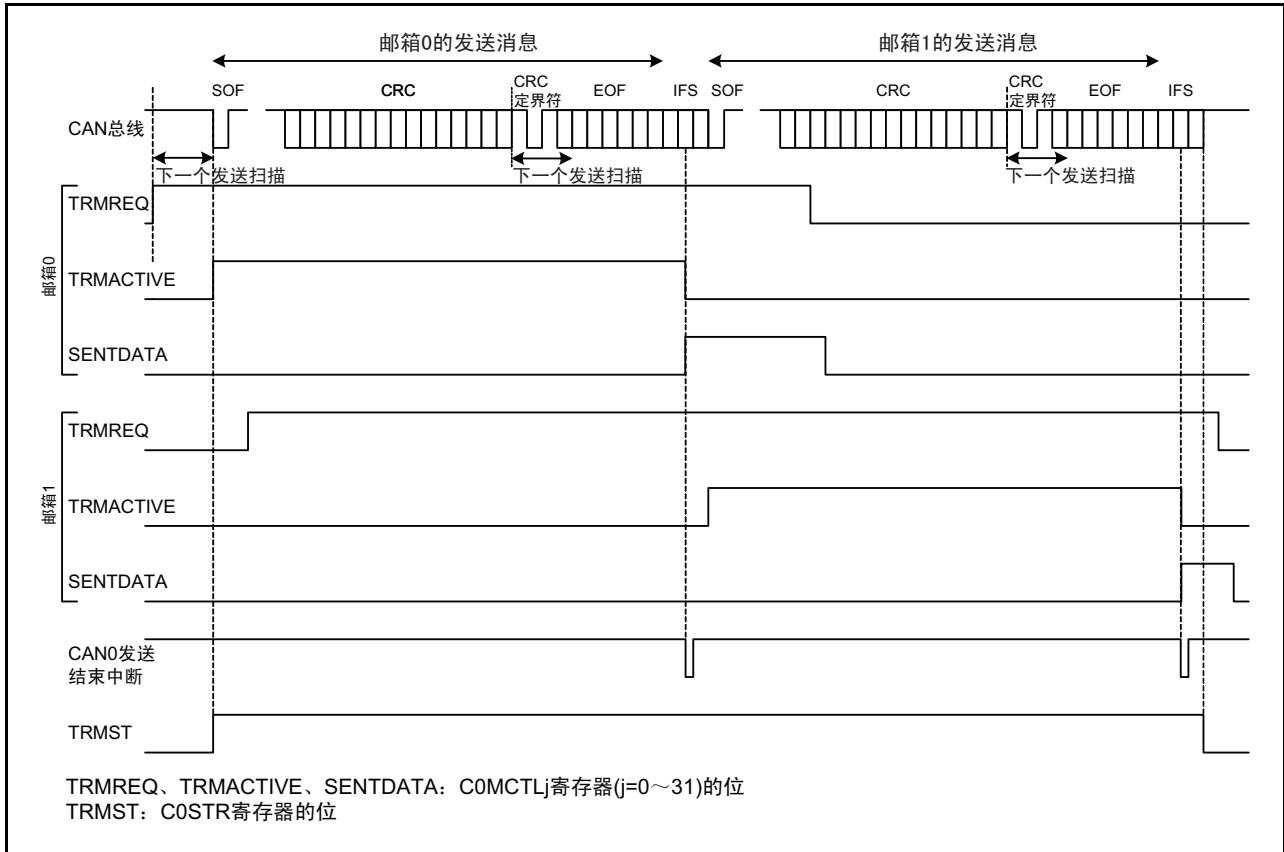


图 25.45 发送数据帧时的运行例子

1. 在总线空闲状态下，如果将COMCTLj寄存器（j=0~31）的TRMREQ位置“1”（发送邮箱），为了决定优先级最高的发送邮箱，开始邮箱扫描处理。在决定发送邮箱后，COMCTLj寄存器的TRMACTIVE位变为“1”（从读取发送请求到发送结束、发生错误、或者发生仲裁失败）、C0STR寄存器的TRMST位变为“1”（发送中），并且CAN模块开始发送（注1）。
2. 设定了其他的TRMREQ位时，从CRC定界符开始执行为了下一个发送的发送扫描处理。
3. 如果没有发生仲裁失败就完成了发送，COMCTLj寄存器的SENTDATA位就变为“1”（发送结束）、TRMACTIVE位就变为“0”（在发送待机期间，或者无发送请求）。而且在COMIER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. 从相同邮箱请求下一个发送时，必须将SENTDATA位和TRMREQ位置“0”，并且在确认SENTDATA位和TRMREQ位变为“0”后，将TRMREQ位置“1”。

注 1. 如果CAN模块在发送开始后发生仲裁失败，TRMACTIVE位就变为“0”，为了从CRC定界符的起始开始检测优先级最高的发送邮箱，必须重新进行发送扫描处理。在发送中或者在紧接着仲裁失败发生错误时，为了从错误定界符的起始开始检索优先级最高的发送邮箱，必须重新进行发送扫描处理。

## 25.7 CAN 中断

CAN 模块有以下的 CAN 中断。

- CAN0 唤醒中断
- 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

CAN0 错误中断有 8 个中断源，可通过检查 COEIFR 寄存器确认这些中断源。

- 总线错误
- 总线报警
- 被动错误
- 总线关闭开始
- 总线关闭返回
- 接收溢出
- 重载帧发送
- 总线锁定

## 26. 输入 / 输出引脚

本单片机的各引脚能用作可编程输入 / 输出端口、内部外围功能的输入 / 输出或者总线控制引脚，能通过功能选择寄存器和处理器模式寄存器选择这些功能。在此对功能选择寄存器进行说明，有关用作总线控制引脚的情况，请参照“7. 处理器模式”和“9. 总线”。

每 4 个引脚能选择有无上拉电阻。如果是输出引脚或者模拟输入 / 输出引脚，与寄存器的设定内容无关，将上拉电阻断开。

典型的输入 / 输出引脚的框图如图 26.1 所示。

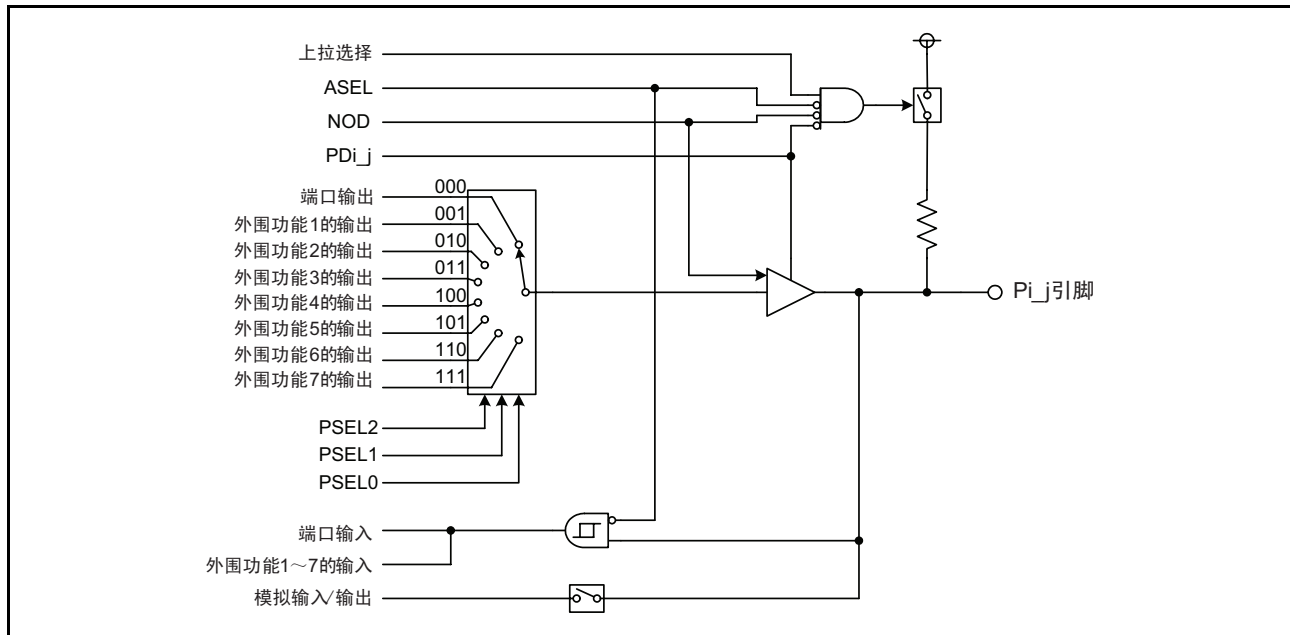


图 26.1 输入 / 输出引脚的框图 (典型例子) ( $i=0 \sim 15$ ,  $j=0 \sim 7$ )

对于输入 / 输出引脚，有决定引脚输入 / 输出方向的方向寄存器、决定引脚功能的输出功能选择寄存器以及控制上拉电阻的上拉控制寄存器。输出功能选择寄存器有 PSEL2 ~ PSEL0 位（用于选择是使用可编程输入 / 输出端口还是使用外围功能输出（模拟输出除外））、NOD 位（用于将引脚输出设定为 N 沟道漏极开路）和 ASEL 位（用于防止模拟输入 / 输出时产生的中间电位使输入缓冲器的消耗电流增大）。

在用作模拟输入 / 输出引脚时，必须将 PDi\_j 设定为“0”（输入），并且将 PSEL2 ~ PSEL0 位和 ASEL 位分别设定为“000b”和“1”。

端口 P8\_5 是和 NMI 引脚复用的输入专用引脚，没有功能选择寄存器和方向寄存器的 bit5。端口 P14\_1（在 100 引脚版中为 P9\_1）也是输入专用引脚，功能选择寄存器和方向寄存器的 bit1 都为保留位。

通过 PRCR 寄存器的 PRC2 位，保护端口 9 不被意外地改写（参照“10. 保护”）。

## 26.1 端口 Pi 方向寄存器 (PDi 寄存器, i=0 ~ 15)

这是选择引脚输入 / 输出方向的寄存器，此寄存器的各位与各引脚一一对应。

在存储器扩展模式或者微处理器模式中，不能更改作为总线控制引脚 (A0 ~ A23、D0 ~ D31、CS0 ~ CS3、WR/WR0、BC0、BC1/WR1、BC2/WR2、BC3/WR3、RD、CLKOUT/BCLK、HLDA、HOLD、ALE、RDY) 的 PDi 寄存器。

PDi 寄存器如图 26.2 所示。

没有与 P8\_5 对应的方向寄存器的位，与 P14\_1 (在 100 引脚版中为 P9\_1) 对应的方向寄存器的位为保留位。

通过 PRCR 寄存器的 PRC2 位，保护 PD9 寄存器不被意外地改写 (参照“10. 保护”)。

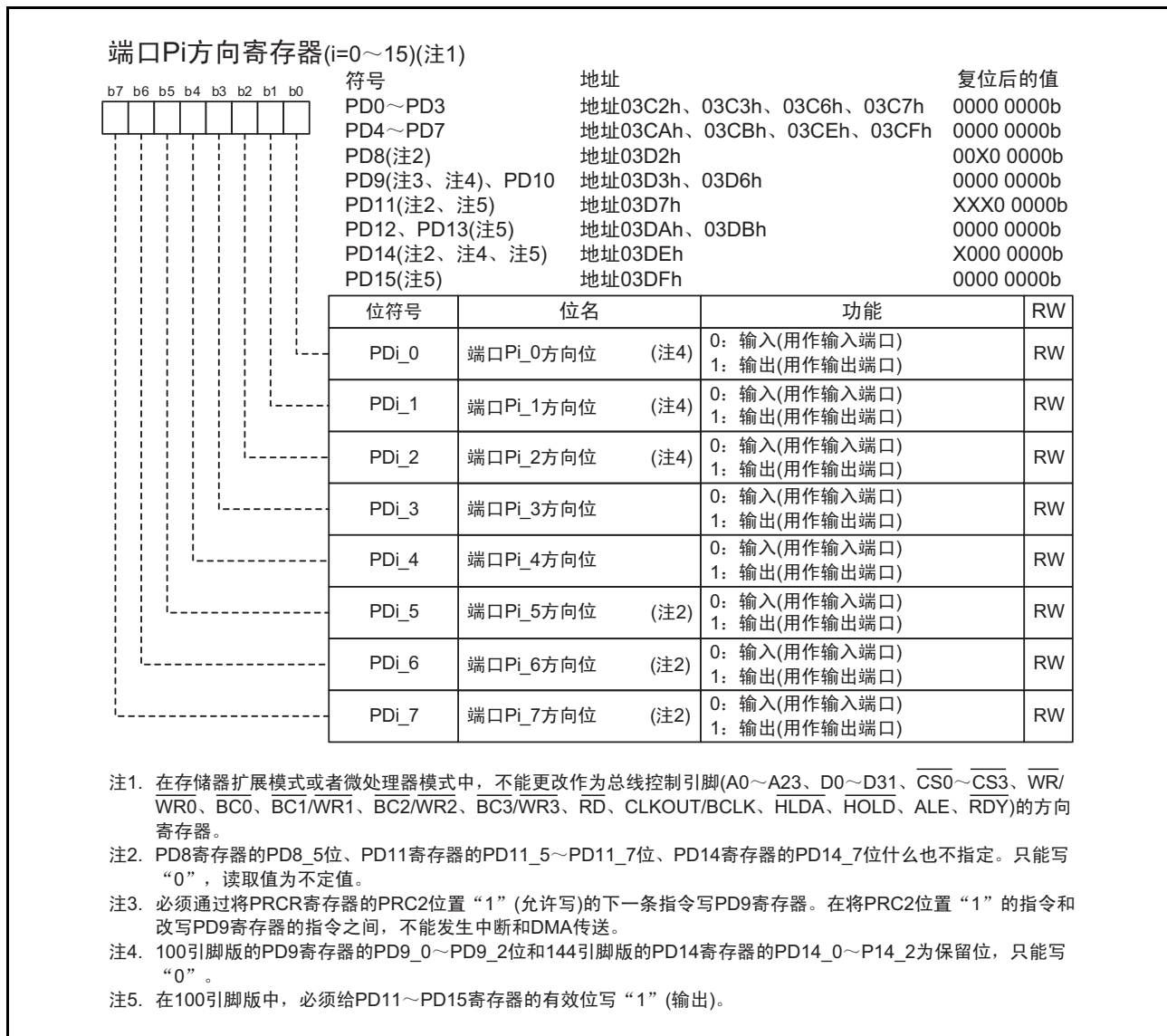


图 26.2 PD0 ~ PD15 寄存器

## 26.2 输出功能选择寄存器

这是在可编程输入 / 输出端口和外围功能输出引脚复用时，决定该引脚输出功能的寄存器。有关输入功能，与此寄存器的设定无关，输入到被连接的全部外围功能。

输出功能选择寄存器有 PSEL2 ~ PSEL0 位（用于选择是使用可编程输入 / 输出端口还是使用外围功能输出（模拟输出除外））、NOD 位（用于将引脚输出设定为 N 沟道漏极开路）和 ASEL 位（用于防止模拟输入 / 输出时产生的中间电位使消耗电流增大）。

PSEL2 ~ PSEL0 位的值和外围功能的关系如表 26.1 所示，功能选择寄存器如图 26.3 ~ 图 26.19 所示。

因为端口 P8\_5 和端口 P14\_1（在 100 引脚版中为 P9\_1）是输入专用引脚，所以没有输出功能选择寄存器。

通过 PRCR 寄存器的 PRC2 位，保护 P9\_iS 寄存器不被意外地改写（参照“10. 保护”）。

表 26.1 外围功能的分配

PSEL2 ~ PSEL0 位	外围功能
001b	定时器
010b	三相马达控制功能
011b	UART
100b	UART 特殊功能
101b	智能 I/O 的组 0 和组 2、CAN 通道 0
110b	智能 I/O 的组 1
111b	UART8

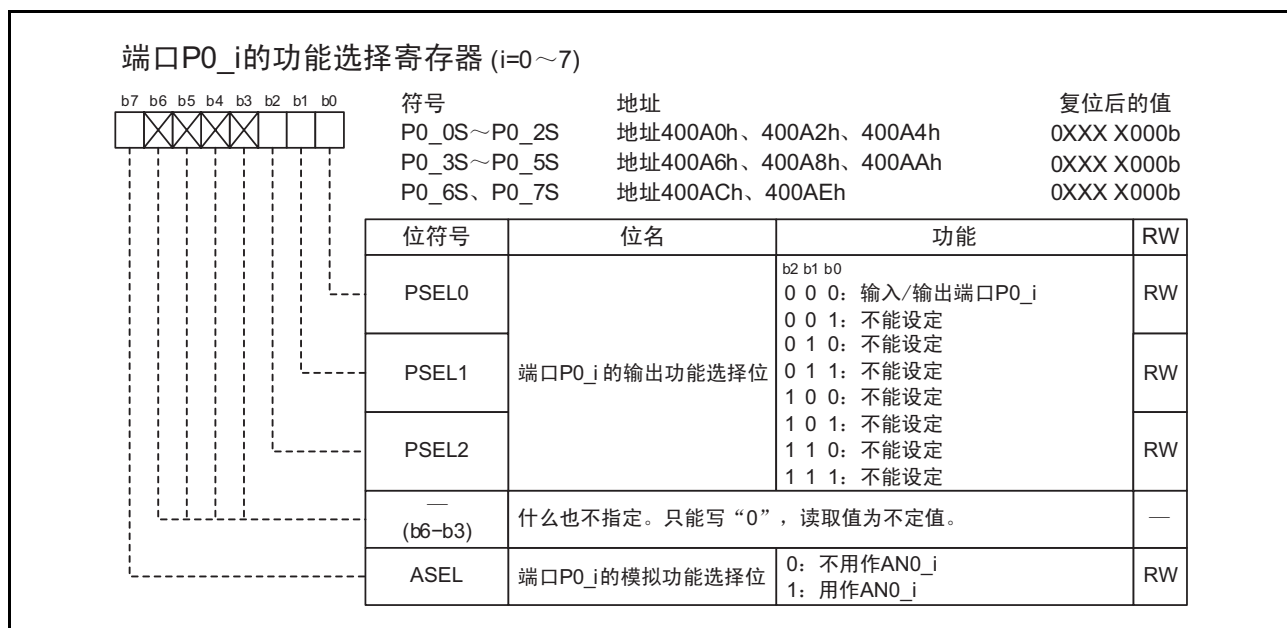


图 26.3 P0\_0S ~ P0\_7S 寄存器

端口 P0\_i (i=0~7) 和 A/D 转换器的输入 AN0\_i 引脚复用。

在用作可编程输入 / 输出端口时，必须给 P0\_iS 寄存器设定“00h”；在用作 A/D 转换器的输入时，必须给 P0\_iS 寄存器设定“80h”，并且将 PD0\_i 设定为“0”（P0\_i 用作输入端口）。

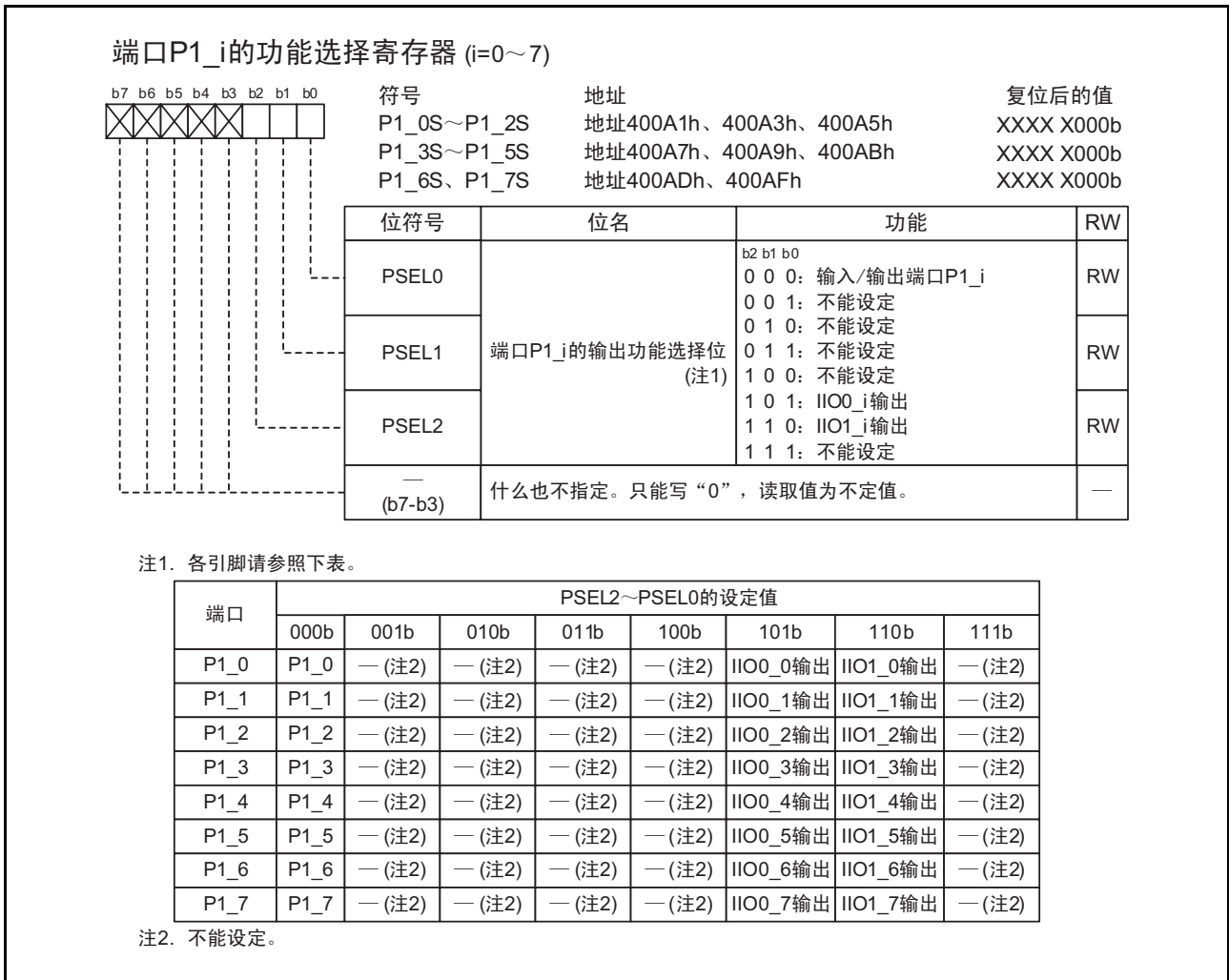


图 26.4 P1\_0S ~ P1\_7S 寄存器

端口 P1<sub>i</sub> (i=0 ~ 7) 和智能 I/O 的组 0、组 1 (IIO0,1) 以及外部中断输入引脚复用。

在用作输出引脚时，必须将 PD1<sub>i</sub> 设定为 “1” (P1<sub>i</sub> 用作输出端口)，并根据图 26.4 选择功能；在用作输入引脚时，必须将 PD1<sub>i</sub> 设定为 “0” (P1<sub>i</sub> 用作输入端口)。

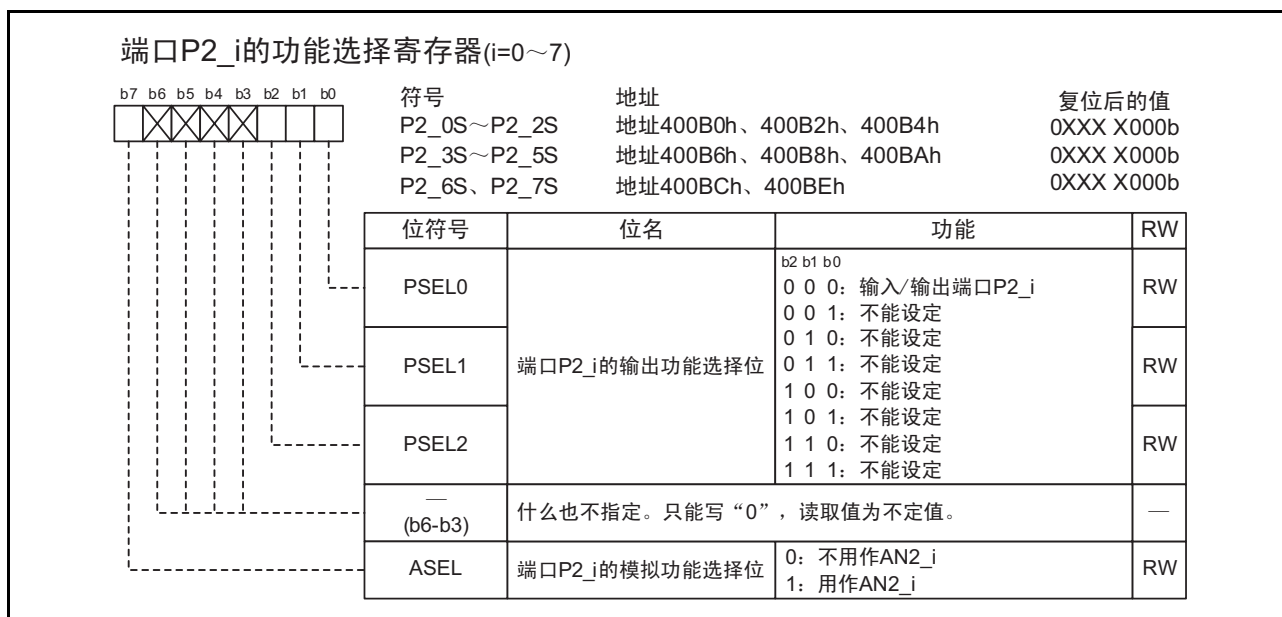


图 26.5 P2\_0S ~ P2\_7S 寄存器

端口 P2\_i (i=0 ~ 7) 和 A/D 转换器的输入 AN2\_i 引脚复用。

在用作可编程输入 / 输出端口时，必须给 P2\_iS 寄存器设定 “00h”；在用作 A/D 转换器的输入时，必须给 P2\_iS 寄存器设定 “80h”，并且将 PD2\_i 设定为 “0” (P2\_i 用作输入端口)。

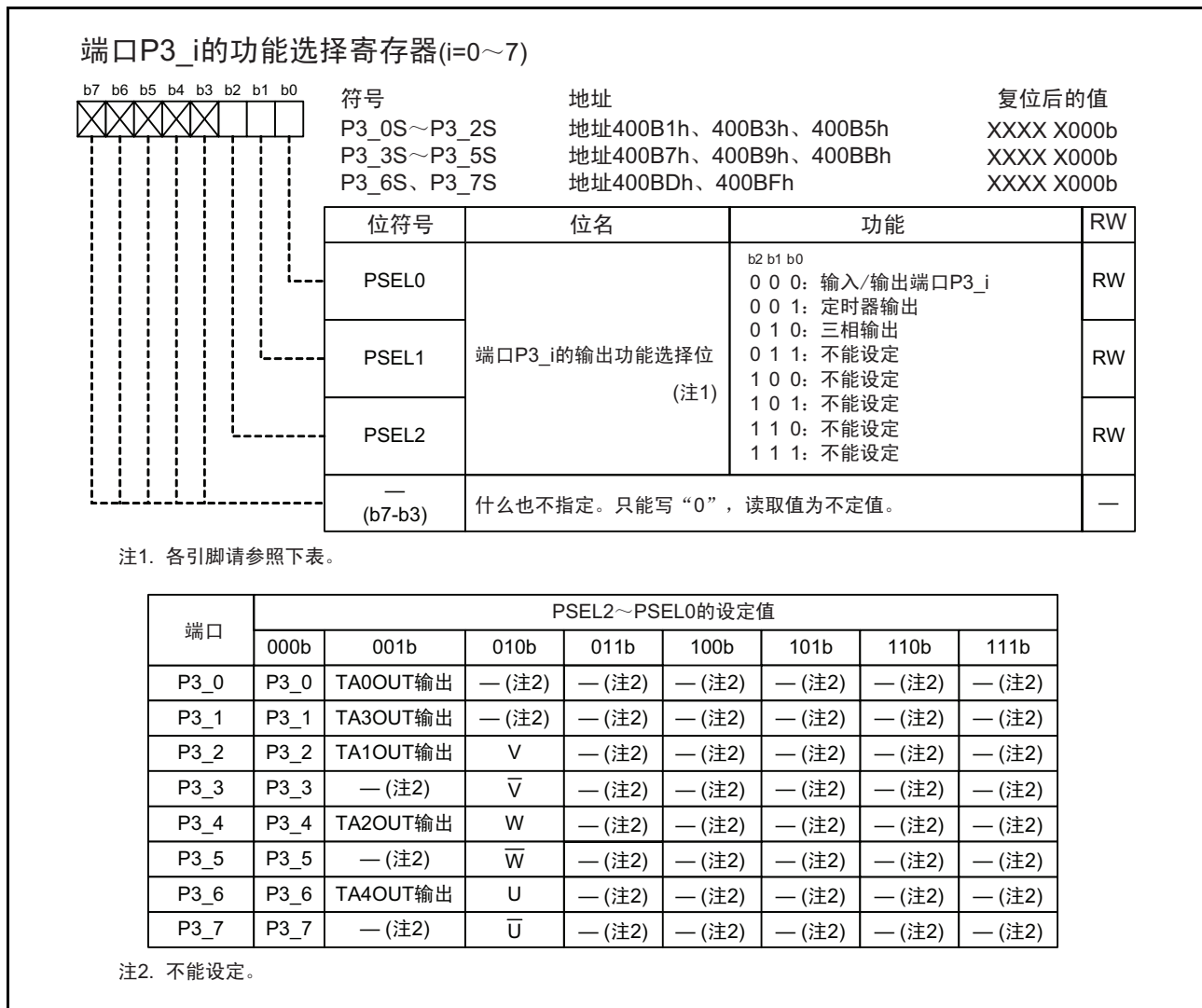


图 26.6 P3\_0S ~ P3\_7S 寄存器

端口 P3<sub>i</sub> (i=0 ~ 7) 和定时器输出、三相马达控制输出引脚复用。

在用作输出引脚时，必须将 PD3<sub>i</sub> 设定为“1”（P3<sub>i</sub> 用作输出端口），并根据图 26.6 选择功能；在用作输入引脚时，必须将 PD3<sub>i</sub> 设定为“0”（P3<sub>i</sub> 用作输入端口）。



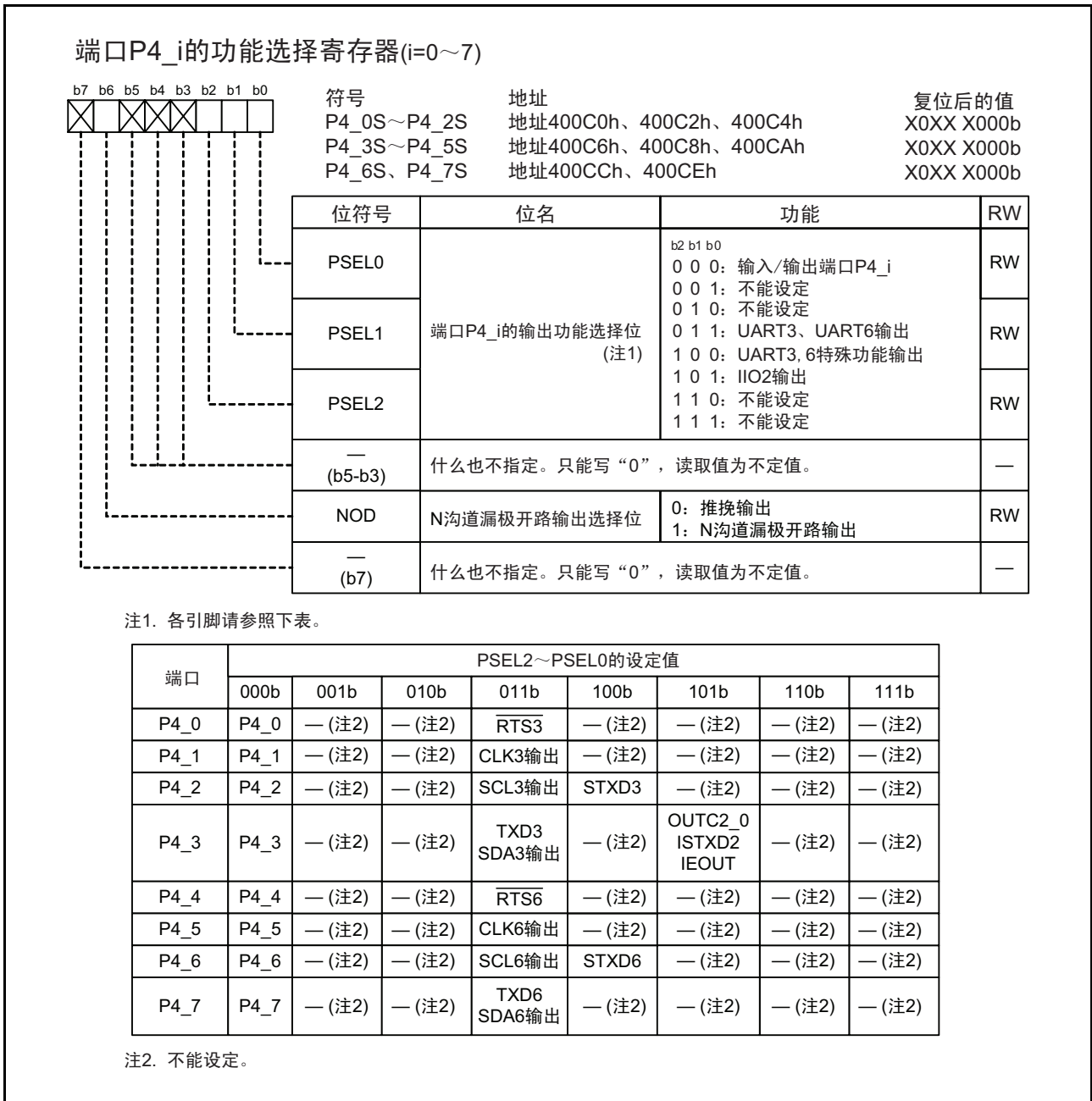


图 26.7 P4\_0S ~ P4\_7S 寄存器

端口 P4\_i (i=0 ~ 7) 和串行接口 (UART3、UART6)、智能 I/O 的组 2 (IIO2) 引脚复用。

在用作输出引脚时，必须将 PD4\_i 设定为 “1” (P4\_i 用作输出端口)，并根据图 26.7 选择功能；在用作输入引脚时，必须将 PD4\_i 设定为 “0” (P4\_i 用作输入端口)。

端口 P4\_0 ~ P4\_7 为 5V 容许输入。在用作输入 / 输出引脚时，为了将 5V 容许输入设定为有效，必须将 NOD 位置 “1”，并且将对应的引脚设定为 N 沟道漏极开路输出。

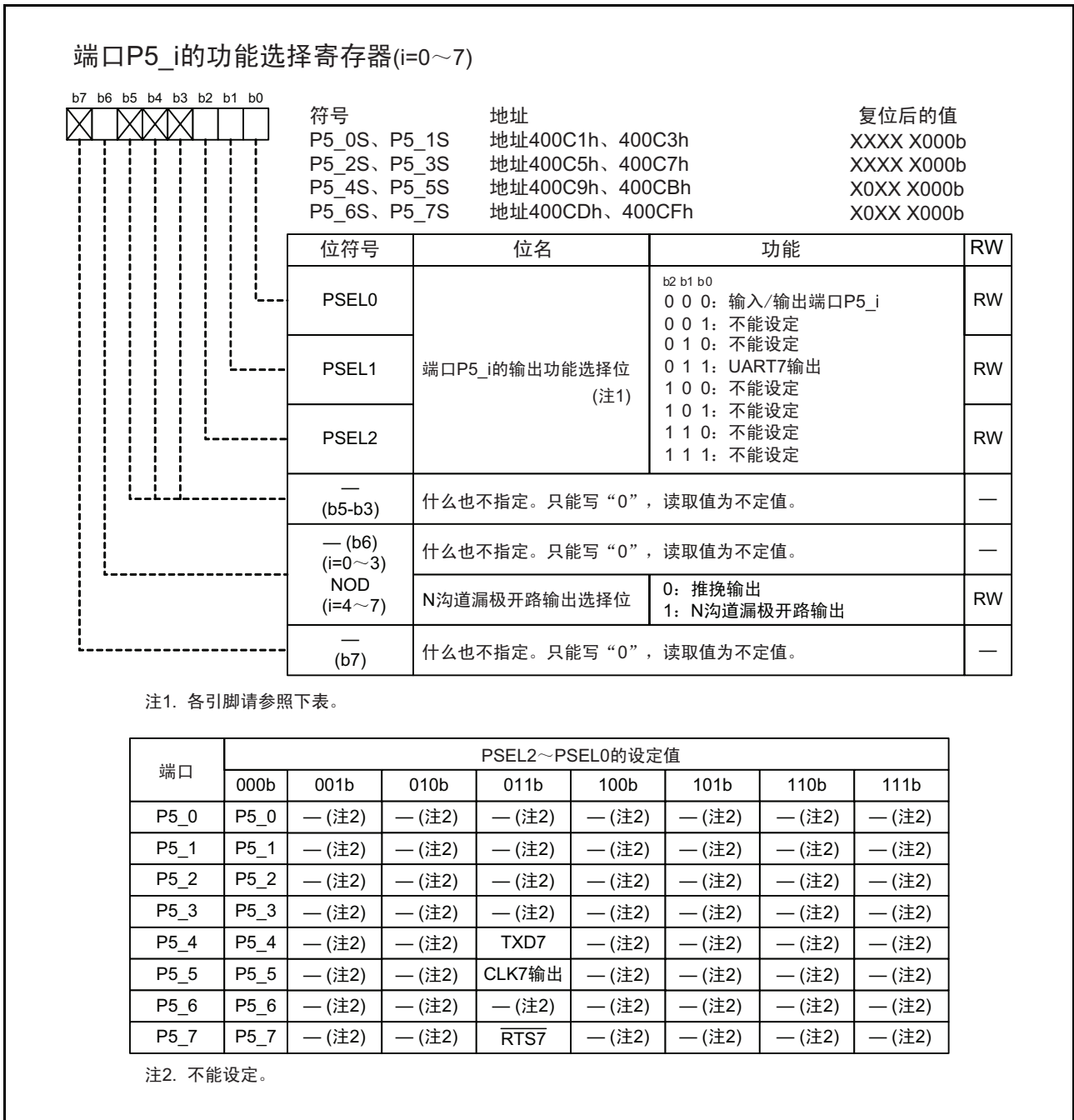


图 26.8 P5\_0S ~ P5\_7S 寄存器

端口 P5\_i (i=0 ~ 7) 和串行接口 (UART7) 引脚复用。

在用作输出引脚时，必须将 PD5\_i 设定为 “1” (P5\_i 用作输出端口)，并根据图 26.8 选择功能；在用作输入引脚时，必须将 PD5\_i 设定为 “0” (P5\_i 用作输入端口)。

端口 P5\_4 ~ P5\_7 为 5V 容许输入。在用作输入 / 输出引脚时，为了将 5V 容许输入设定为有效，必须将 NOD 位置 “1”，并且将对应的引脚设定为 N 沟道漏极开路输出。

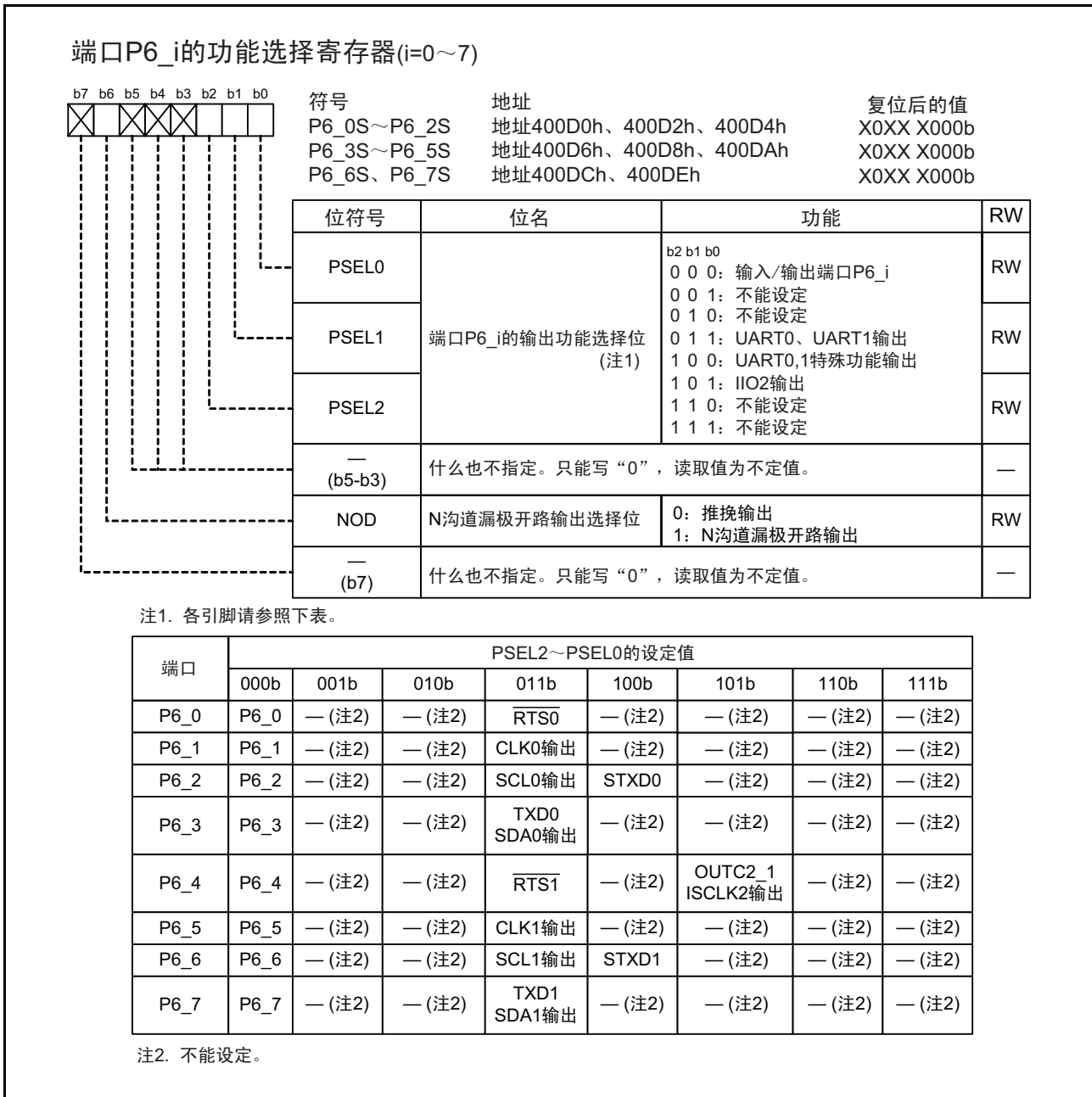


图 26.9 P6\_0S ~ P6\_7S 寄存器

端口 P6\_i (i=0 ~ 7) 和串行接口 (UART0、UART1)、智能 I/O 的组 2 (IIO2) 引脚复用。

在用作输出引脚时，必须将 PD6\_i 设定为 “1” (P6\_i 用作输出端口)，并根据图 26.9 选择功能；在用作输入引脚时，必须将 PD6\_i 设定为 “0” (P6\_i 用作输入端口)。

端口 P6\_0 ~ P6\_7 为 5V 容许输入。在用作输入 / 输出引脚时，为了将 5V 容许输入设定为有效，必须将 NOD 位置 “1”，并且将对应的引脚设定为 N 沟道漏极开路输出。



图 26.10 P7\_0S ~ P7\_7S 寄存器

端口 P7\_i (i=0 ~ 7) 和定时器、三相马达控制、串行接口 (UART2、UART5、UART8)、多主控 I<sup>2</sup>C 总线接口 (MMI2C)、智能 I/O 的组 1、组 2 (IIO1、IIO2)、CAN 模块引脚复用。

在用作输出引脚时，必须将 PD7\_i 设定为 “1” (P7\_i 用作输出端口)，并根据图 26.10 选择功能；在用作输入引脚时，必须将 PD7\_i 设定为 “0” (P7\_i 用作输入端口)。

端口 P7\_0 ~ P7\_7 为 5V 容许输入。在用作输入 / 输出引脚时，为了将 5V 容许输入设定为有效，必须将 NOD 位置 “1”，并且将对应的引脚设定为 N 沟道漏极开路输出。

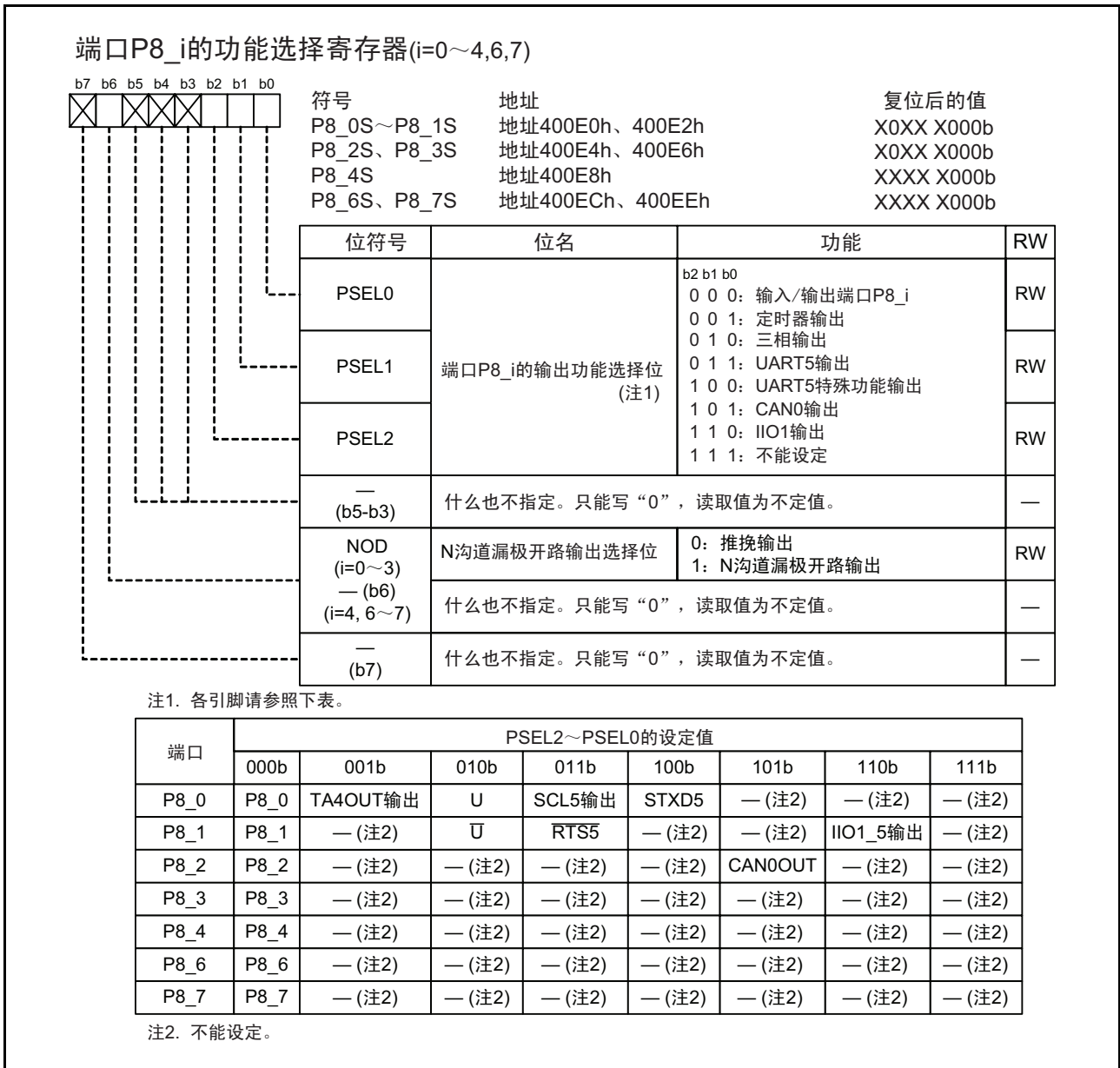


图 26.11 P8\_0S ~ P8\_4S、P8\_6S、P8\_7S 寄存器

端口 P8\_i (i=0 ~ 4,6,7) 和定时器、三相马达控制、串行接口 (UART5)、智能 I/O 的组 1 (IIO1)、CAN 模块、外部中断输入引脚复用。

在用作输出引脚时，必须将 PD8\_i 设定为 “1” (P8\_i 用作输出端口)，并根据图 26.11 选择功能；在用作输入引脚时，必须将 PD8\_i 设定为 “0” (P8\_i 用作输入端口)。

端口 P8\_0 ~ P8\_3 为 5V 容许输入。在用作输入 / 输出引脚时，为了将 5V 容许输入设定为有效，必须将 NOD 位置 “1”，并且将对应的引脚设定为 N 沟道漏极开路输出。

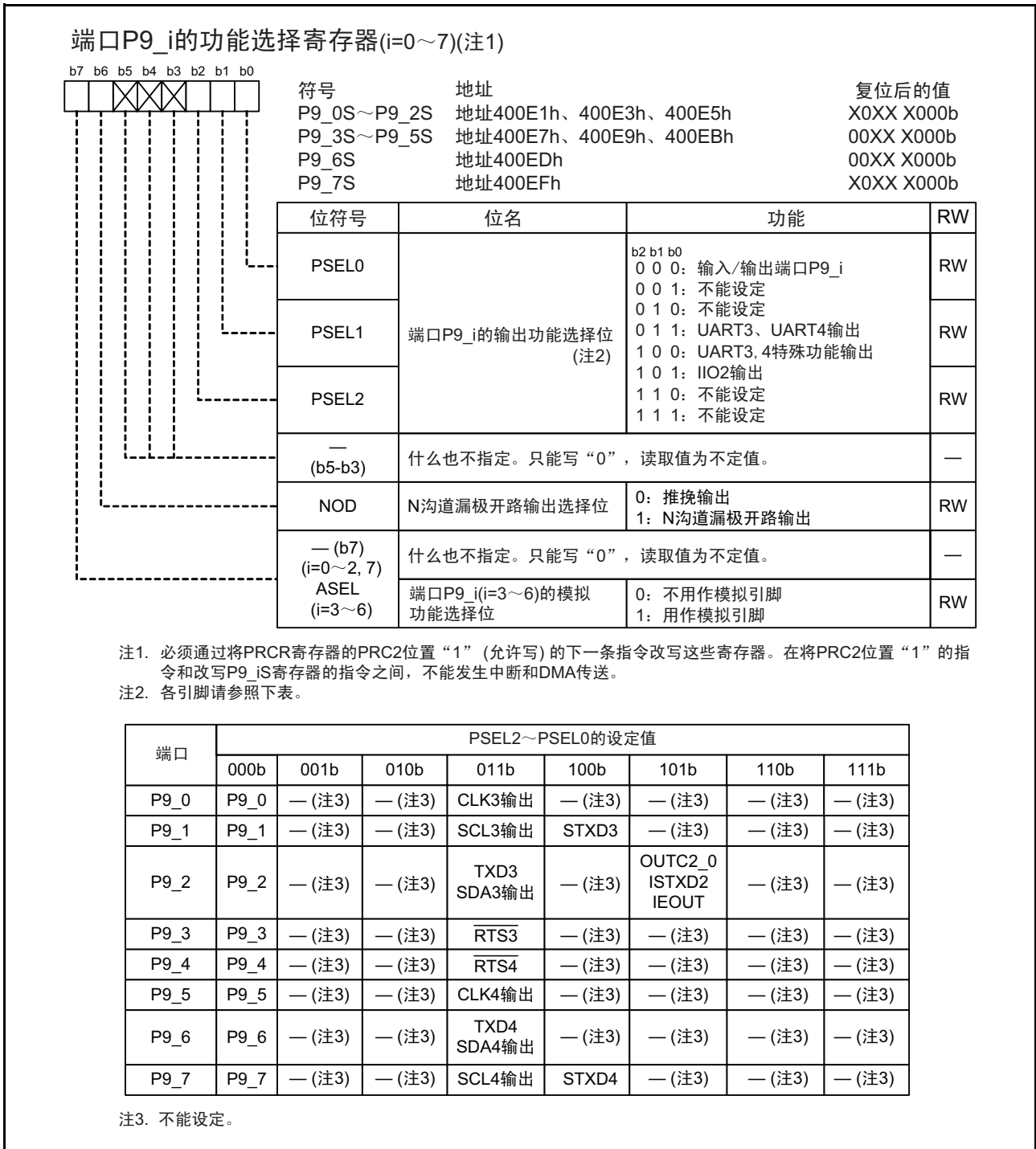


图 26.12 P9\_0S ~ P9\_7S 寄存器（144 引脚版）

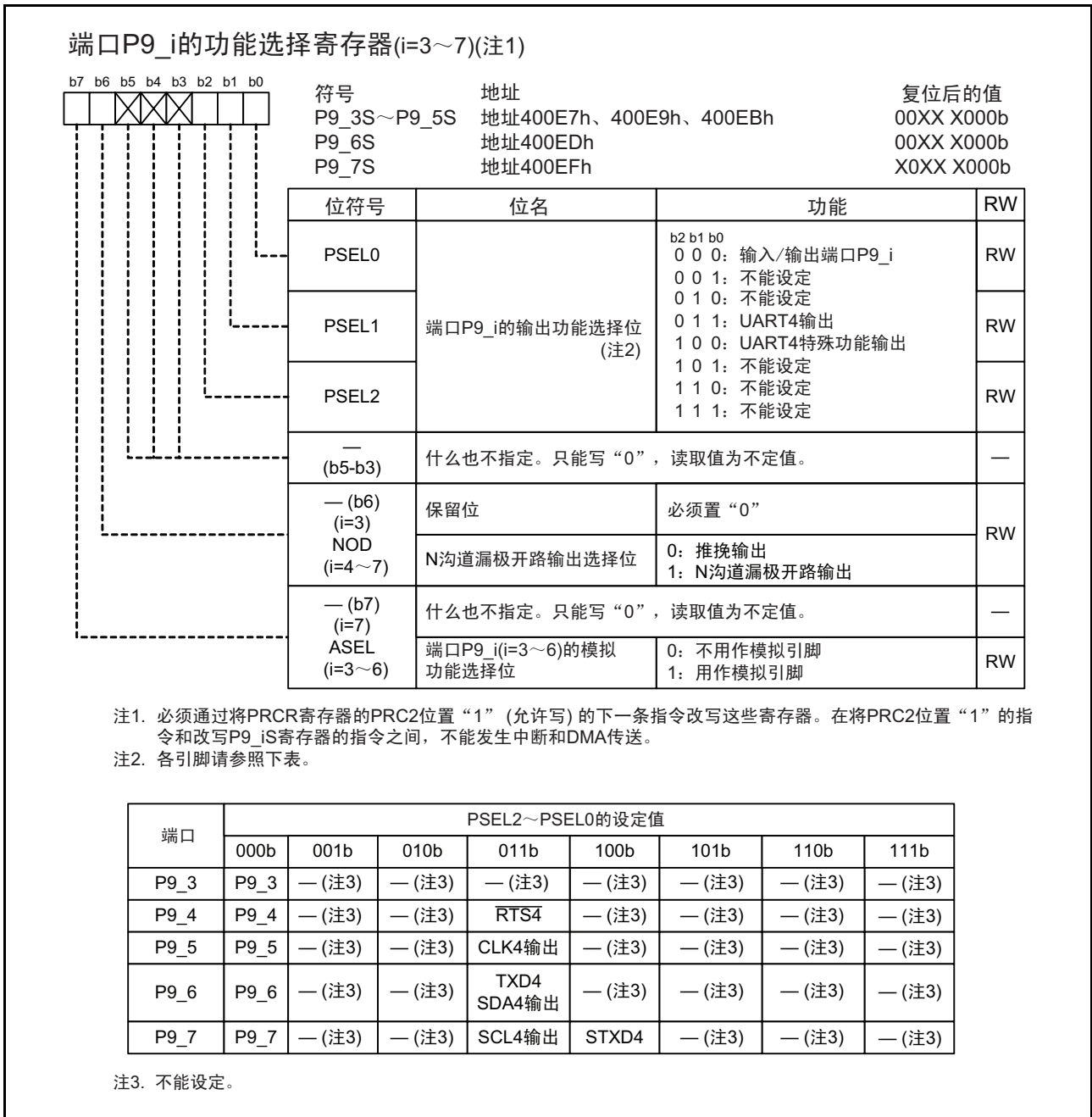


图 26.13 P9\_3S ~ P9\_7S 寄存器（100 引脚版）

端口 P9<sub>i</sub> (i=0 ~ 7) 和串行接口 (UART3、UART4)、智能 I/O 的组 2 (IIO2) 引脚复用。其中，端口 P9<sub>i</sub> (i=3 ~ 6) 又和 A/D 转换器的输入 / 输出 (ANEX0 和 ANEX1)、D/A 转换器的输出引脚复用。

在用作 A/D 转换器、D/A 转换器以外的输出引脚时，必须将 PD9<sub>i</sub> 设定为“1” (P9<sub>i</sub> 用作输出端口)，并根据图 26.12 选择功能；在用作 A/D 转换器、D/A 转换器以外的输入引脚时，必须将 PD9<sub>i</sub> 设定为“0” (P9<sub>i</sub> 用作输入端口)；在用作 A/D 转换器、D/A 转换器时，必须将 P9<sub>i</sub>S 寄存器设定为“80h”，并将 PD9<sub>i</sub> 设定为“0” (P9<sub>i</sub> 用作输入端口)，与输入 / 输出方向无关。

如果将 NOD 位置“1”，对应的引脚就为 N 沟道漏极开路输出。

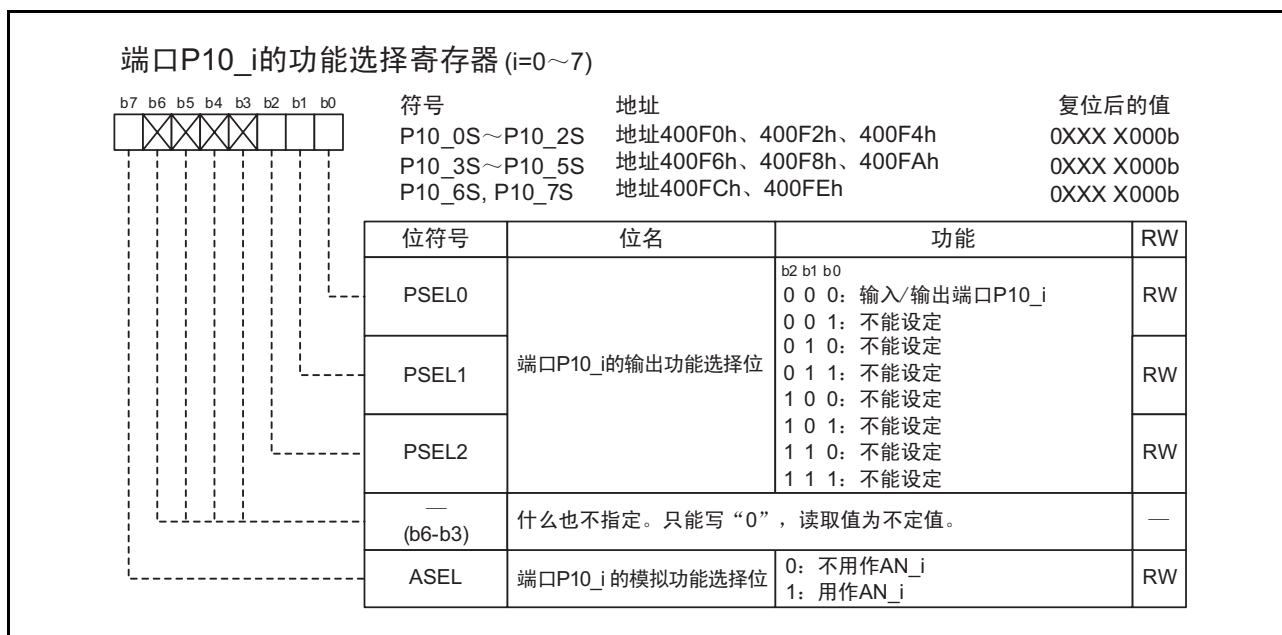


图 26.14 P10\_0S ~ P10\_7S 寄存器

端口 P10<sub>i</sub> (i=0 ~ 7) 和 A/D 转换器的输入 AN<sub>i</sub>、键输入中断引脚复用。

在用作可编程输入 / 输出端口时，必须给 P10<sub>i</sub>S 寄存器设定 “00h”；在用作 A/D 转换器以外的输入引脚时，必须将 PD10<sub>i</sub> 设定为 “0” (P10<sub>i</sub> 用作输入端口)；在用作 A/D 转换器的输入引脚时，必须给 P10<sub>i</sub>S 寄存器设定 “80h”，并将 PD10<sub>i</sub> 设定为 “0” (P10<sub>i</sub> 用作输入端口)。



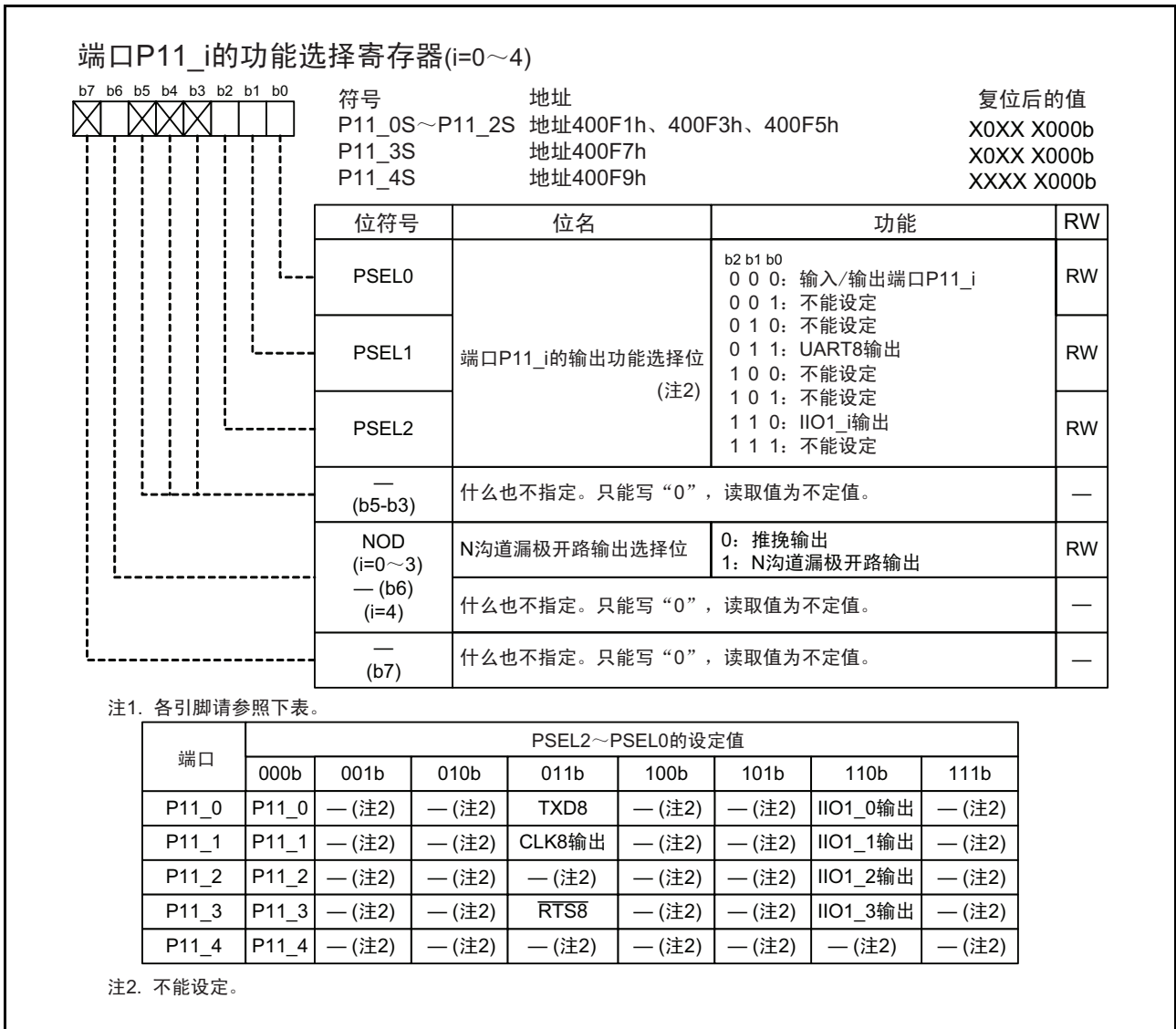


图 26.15 P11\_0S ~ P11\_4S 寄存器

端口 P11\_i (i=0 ~ 4) 和串行接口 (UART8)、智能 I/O 的组 1 (IIO1) 引脚复用。

在用作输出端口时，必须将 PD11\_i 设定为 “1” (P11\_i 用作输出端口)，并根据图 26.15 选择功能；在用作输入引脚时，必须将 PD11\_i 设定为 “0” (P11\_i 用作输入端口)。

如果将 NOD 位置 “1”，对应的引脚就为 N 沟道漏极开路输出。

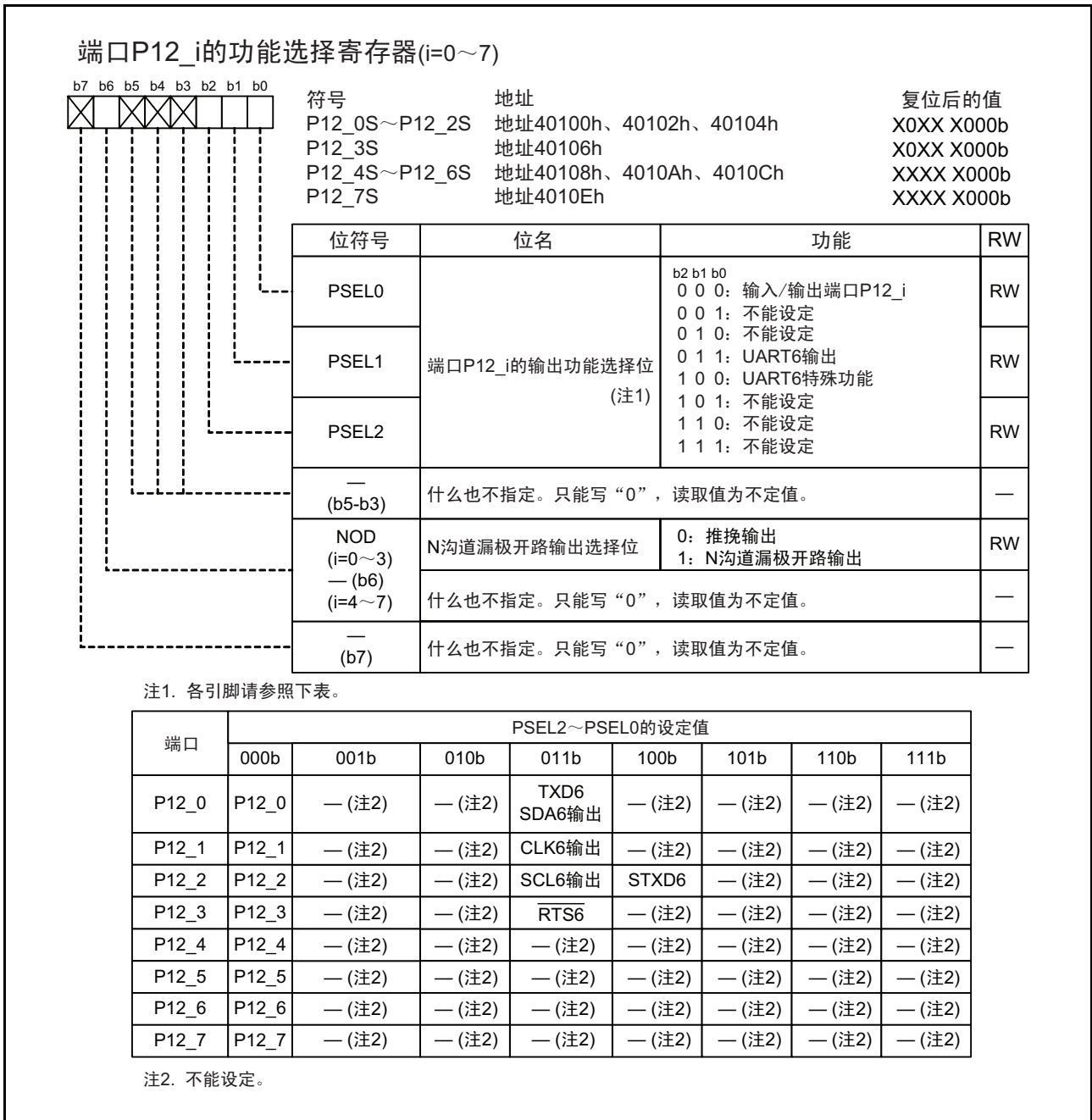


图 26.16 P12\_0S ~ P12\_7S 寄存器

端口 P12\_i (i=0 ~ 7) 和串行接口 (UART6) 引脚复用。

在用作输出端口时，必须将 PD12\_i 设定为 “1” (P12\_i 用作输出端口)，并根据图 26.16 选择功能；在用作输入引脚时，必须将 PD12\_i 设定为 “0” (P12\_i 用作输入端口)。

如果将 NOD 位置 “1”，对应的引脚就为 N 沟道漏极开路输出。

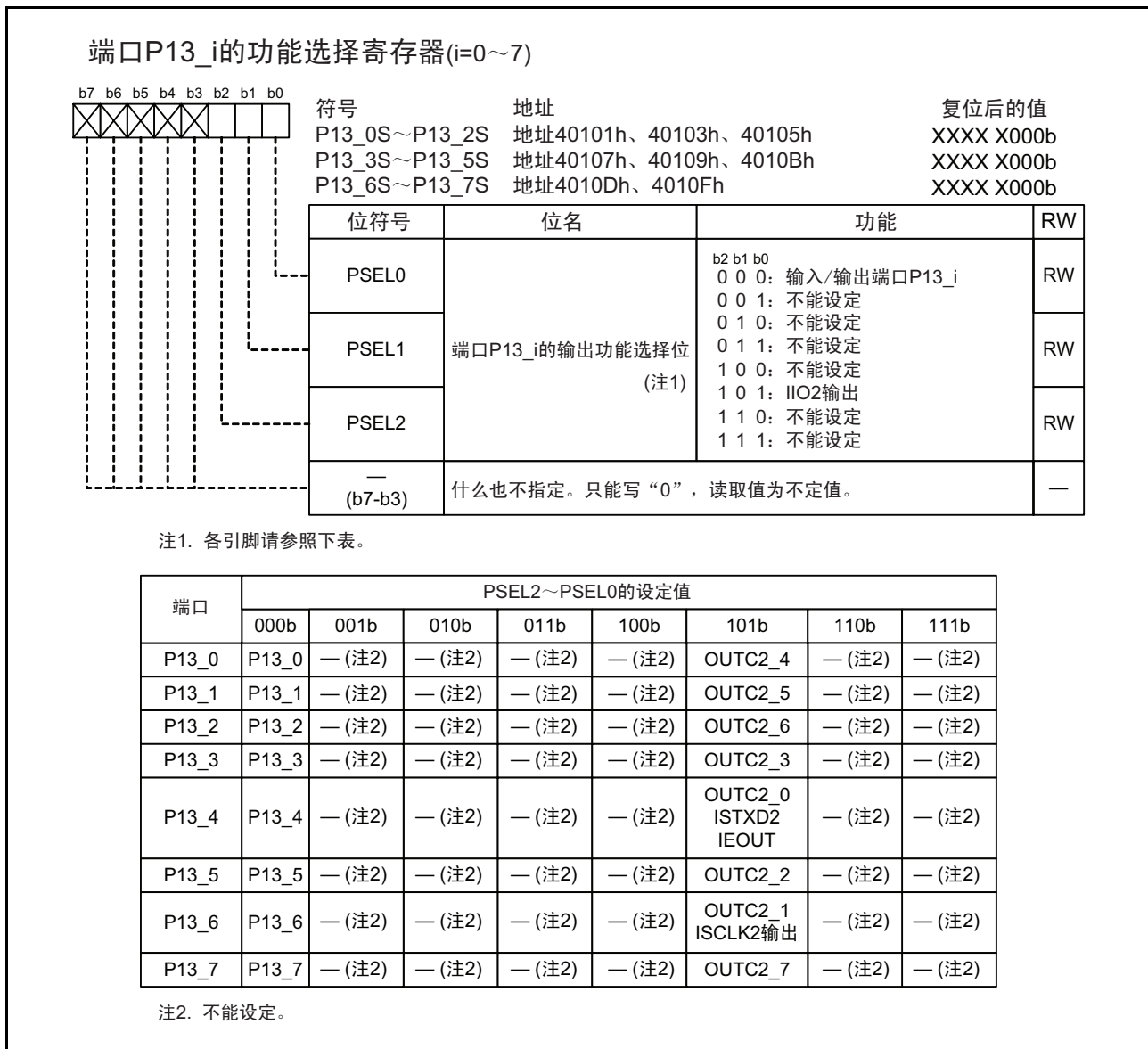


图 26.17 P13\_0S ~ P13\_7S 寄存器

端口 P13<sub>i</sub> (i=0 ~ 7) 和智能 I/O 的组 2 (IIO2) 引脚复用。

在用作输出端口时，必须将 PD13<sub>i</sub> 设定为 “1” (P13<sub>i</sub> 用作输出端口)，并根据图 26.17 选择功能；在用作输入引脚时，必须将 PD13<sub>i</sub> 设定为 “0” (P13<sub>i</sub> 用作输入端口)。

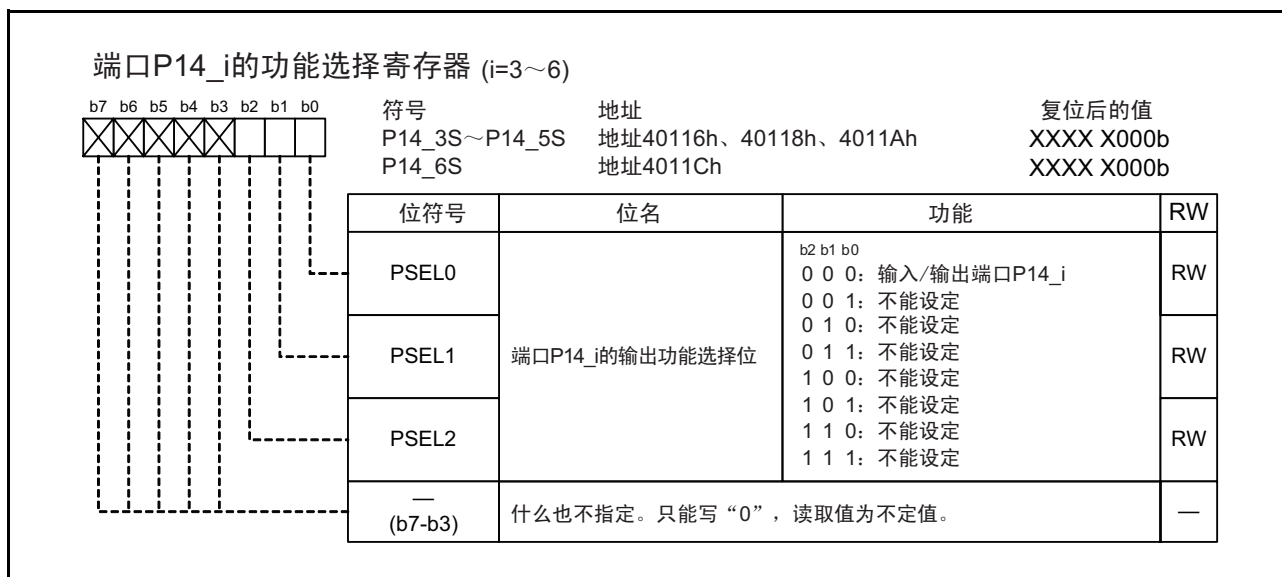


图 26.18 P14\_3S ~ P14\_6S 寄存器

端口 P14<sub>i</sub> (i=3 ~ 6) 和外部中断输入引脚复用。必须将 P14<sub>iS</sub> 设定为 “00h” (输入 / 输出端口)。

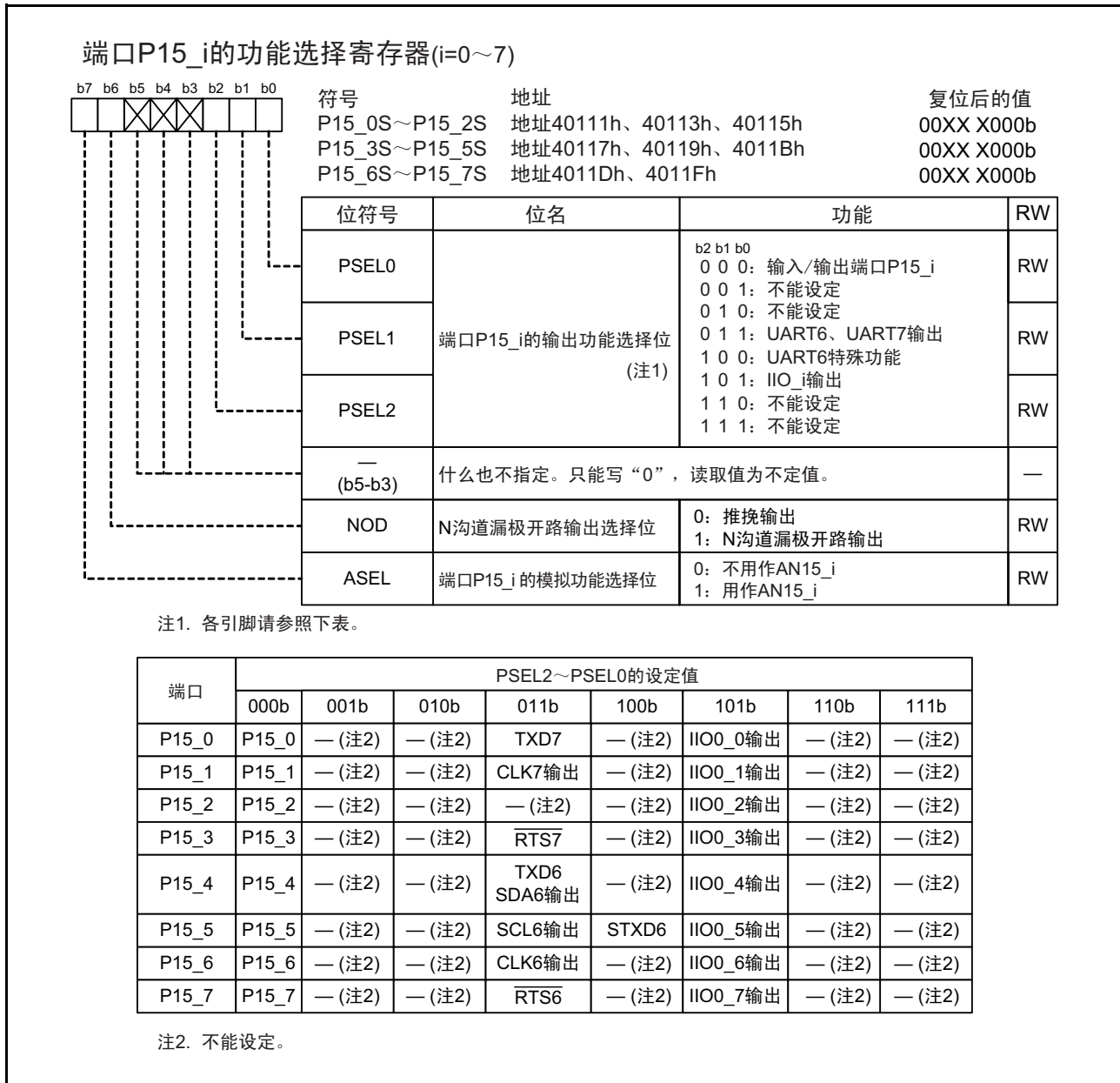


图 26.19 P15\_0S ~ P15\_7S 寄存器

端口 P15<sub>i</sub> (i=0~7) 和串行接口 (UART6、UART7)、智能 I/O 的组 0 (IIO0)、A/D 转换器的输入 AN15<sub>i</sub> 引脚复用。

在用作输出端口时，必须将 PD15<sub>i</sub> 设定为“1” (P15<sub>i</sub> 用作输出端口)，并根据图 26.19 选择功能；在用作 A/D 转换器以外的输入引脚时，必须将 PD15<sub>i</sub> 设定为“0” (P15<sub>i</sub> 用作输入端口)；在用作 A/D 转换器的输入引脚时，必须给 P15<sub>i</sub>S 寄存器设定“80h”，并将 PD15<sub>i</sub> 设定为“0” (P15<sub>i</sub> 用作输入端口)。

如果将 NOD 位置“1”，对应的引脚就为 N 沟道漏极开路输出。

### 26.3 输入功能选择寄存器

这是在给多个引脚分配了外围功能输入时，决定将哪个引脚的输入连接到外围功能的寄存器。  
输入功能选择寄存器如图 26.20 ~ 图 26.23 所示。

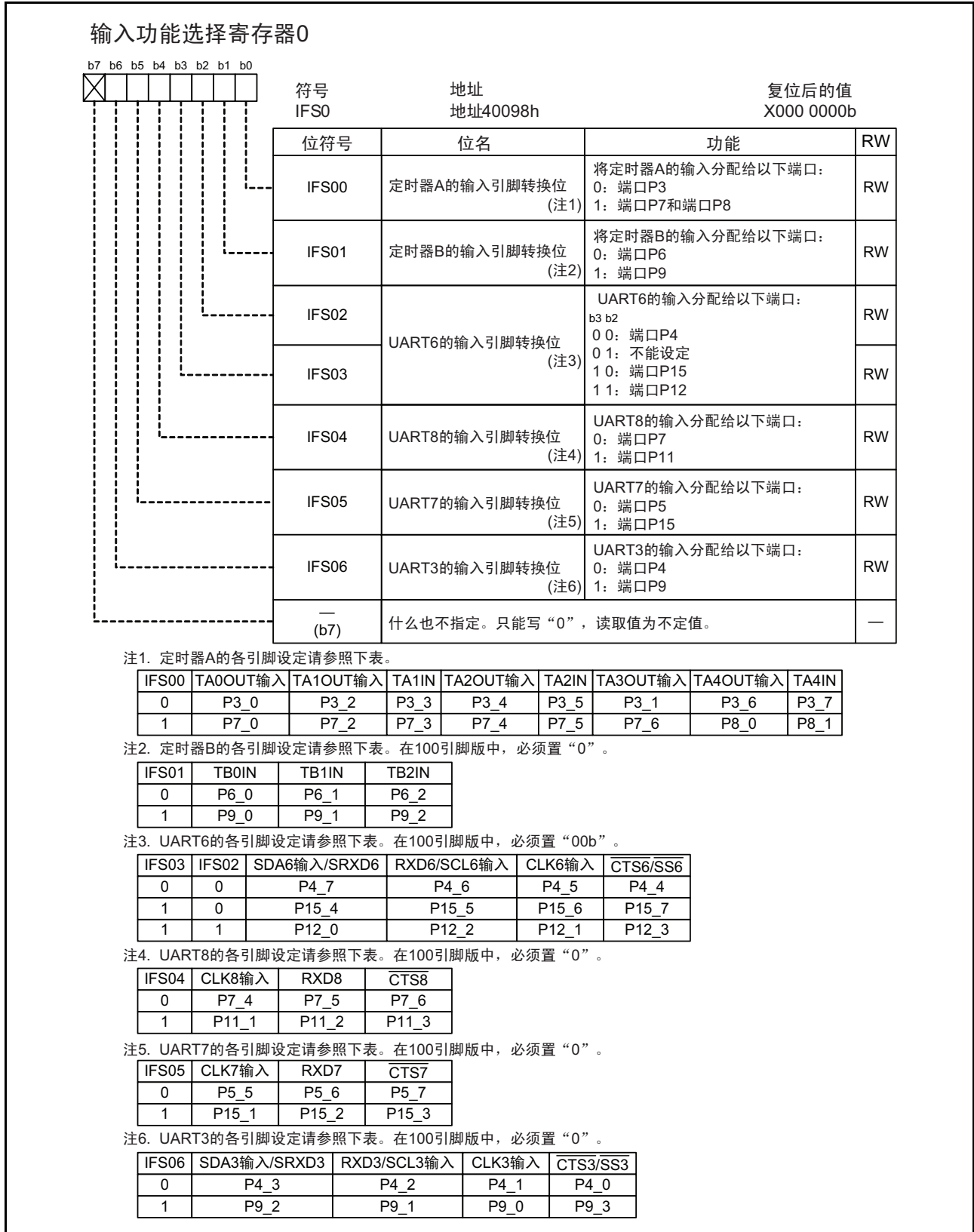


图 26.20 IFS0 寄存器

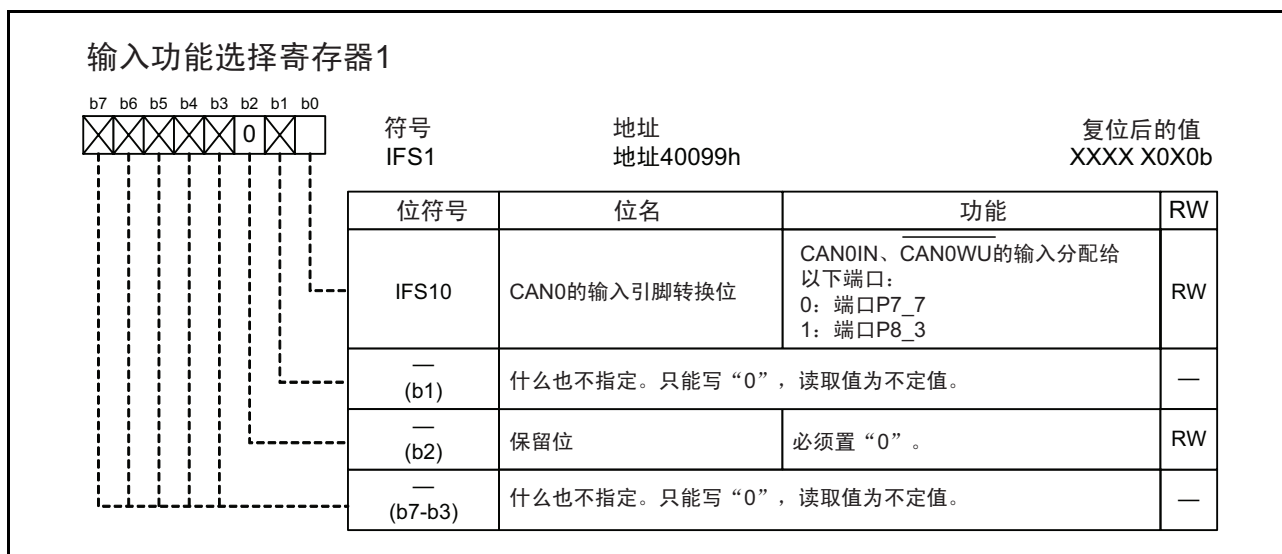


图 26.21 IFS1 寄存器

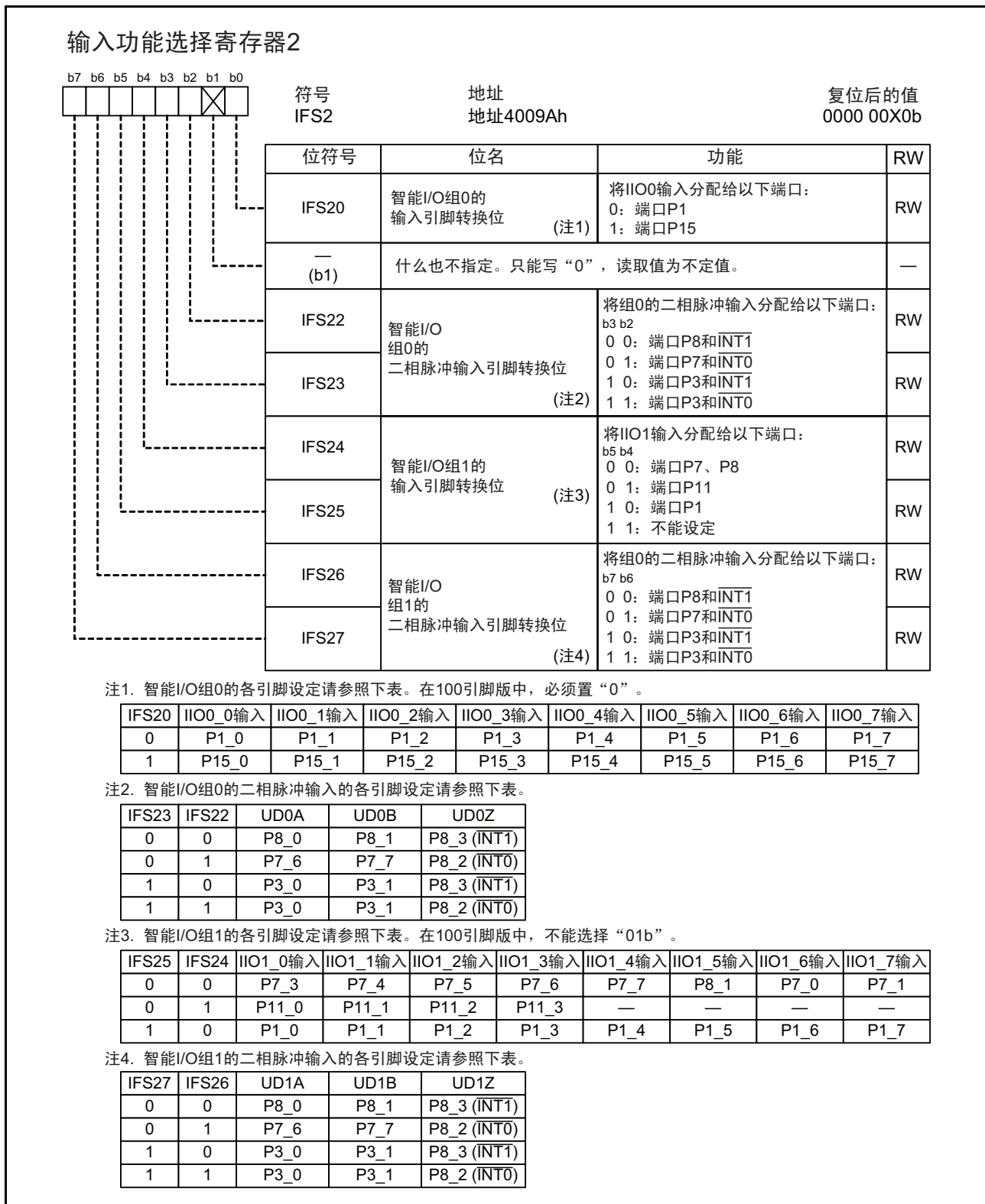


图 26.22 IFS2 寄存器



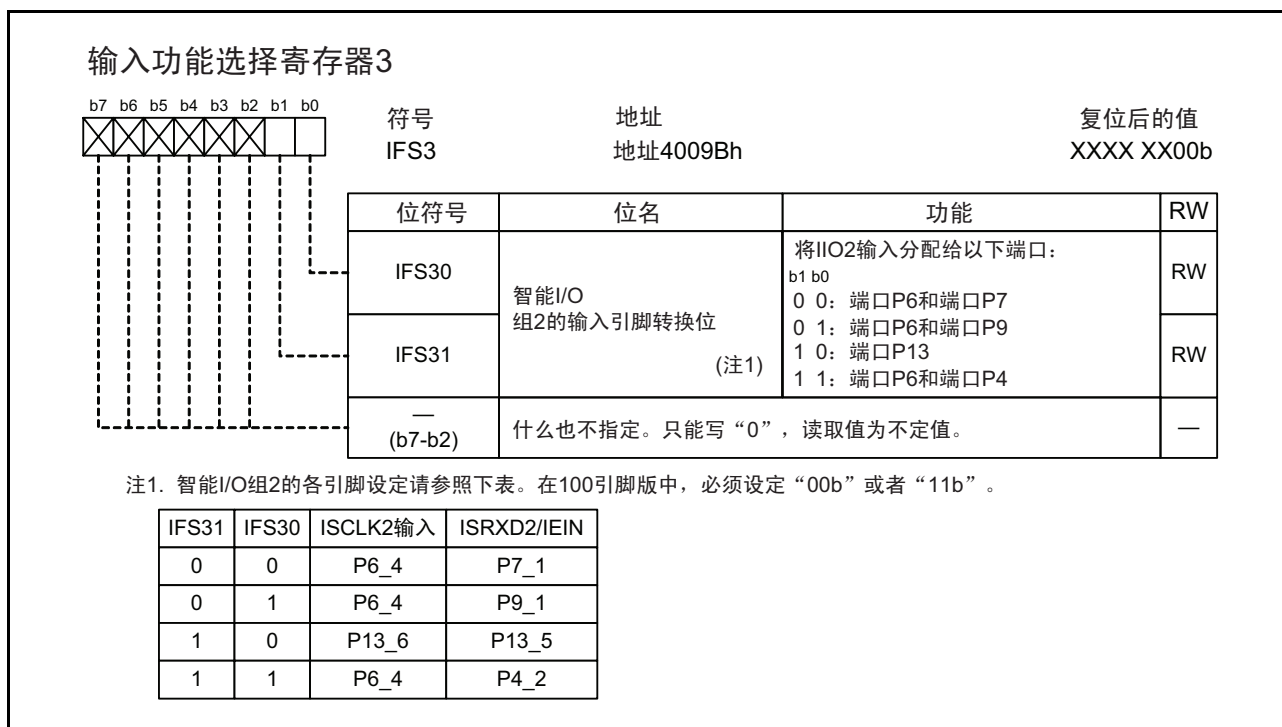


图 26.23 IFS3 寄存器

### 26.4 上拉控制寄存器 0 ~ 4 (PUR0 ~ PUR4 寄存器)

PUR0 ~ PUR4 寄存器如图 26.24 ~ 图 26.28 所示。

能通过 PUR0 ~ PUR3 寄存器，对每 4 个引脚设定是否上拉，将这些寄存器的位设定为“1”（上拉）并将方向寄存器设定为“0”（输入模式）的端口被上拉。

在存储器扩展模式或者微处理器模式中，必须将用作总线的 P0 ~ P5、P11 ~ P13 的上拉控制位置“0”（不上拉），而且在将 P0、P1、P11 ~ P13 用作输入端口时，这些端口能被上拉。

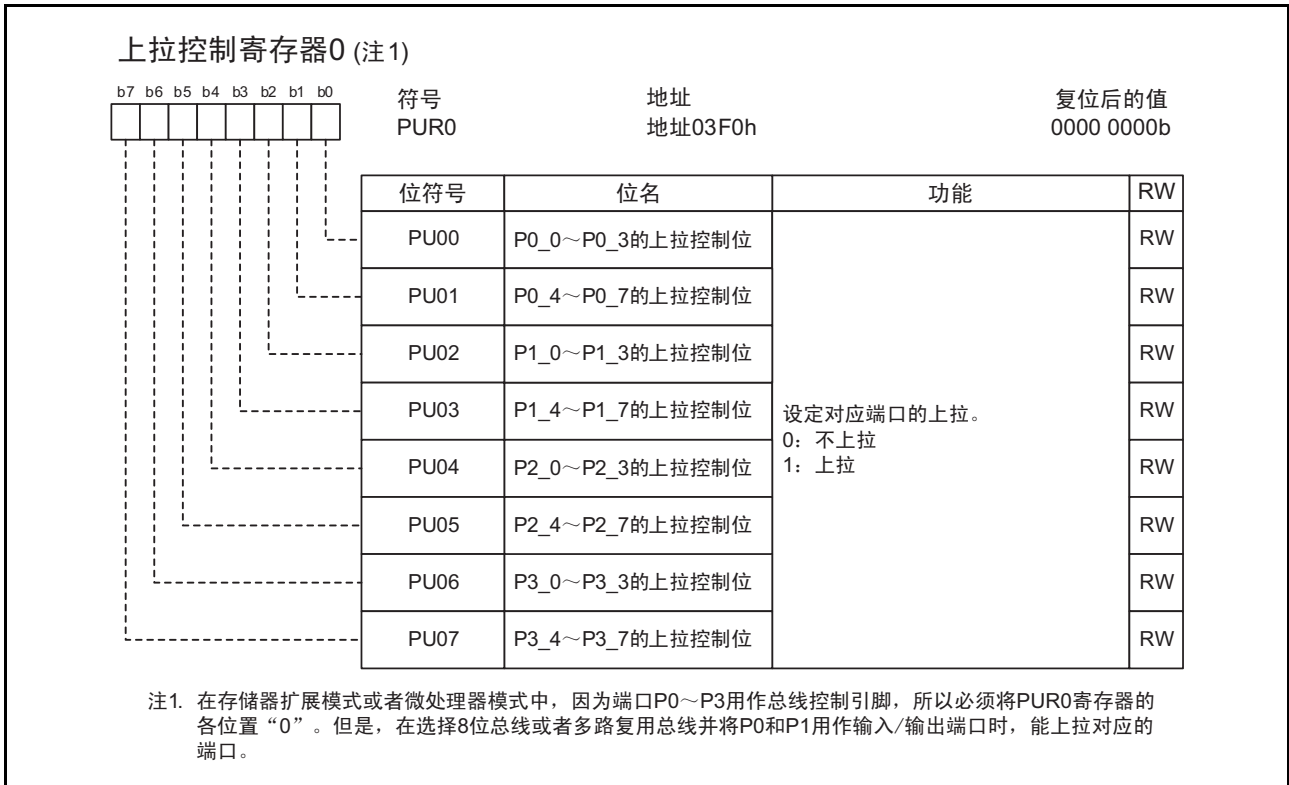


图 26.24 PUR0 寄存器

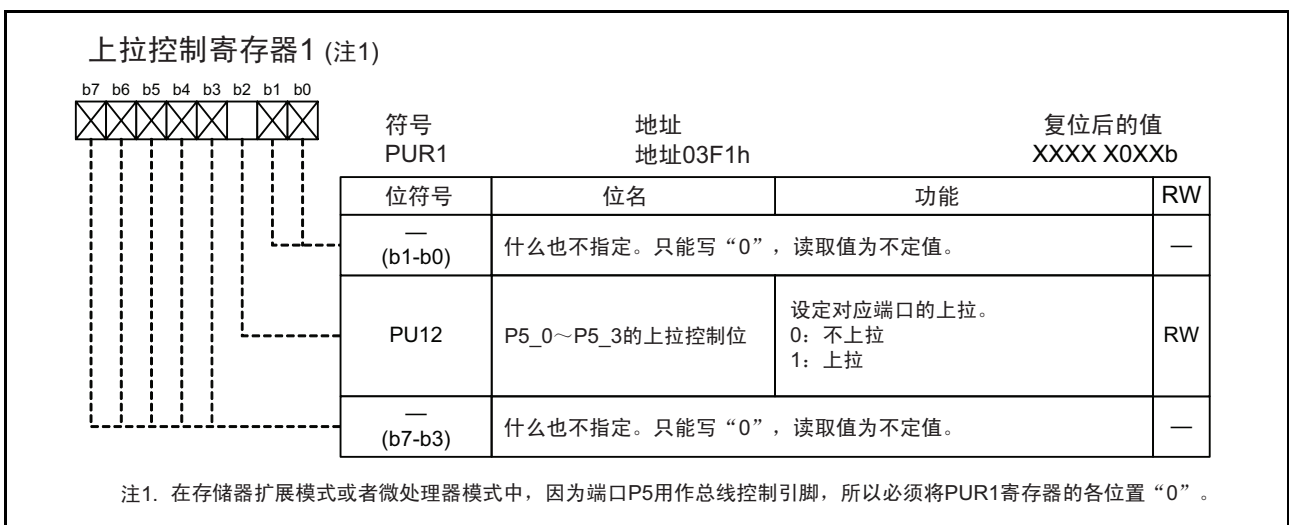


图 26.25 PUR1 寄存器

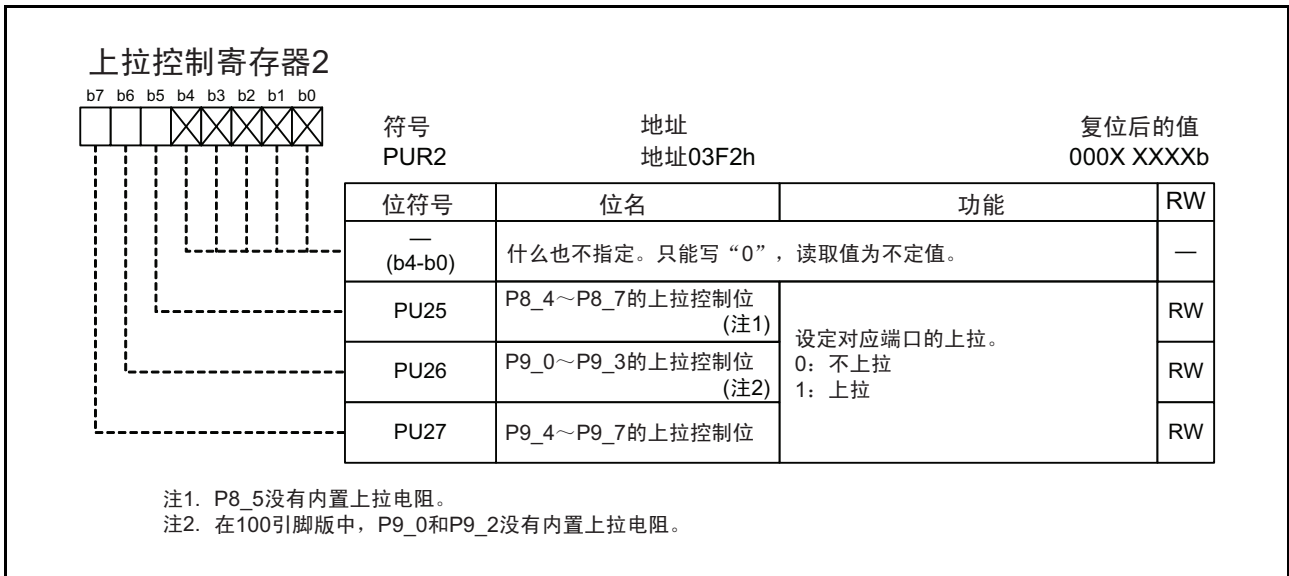


图 26.26 PUR2 寄存器

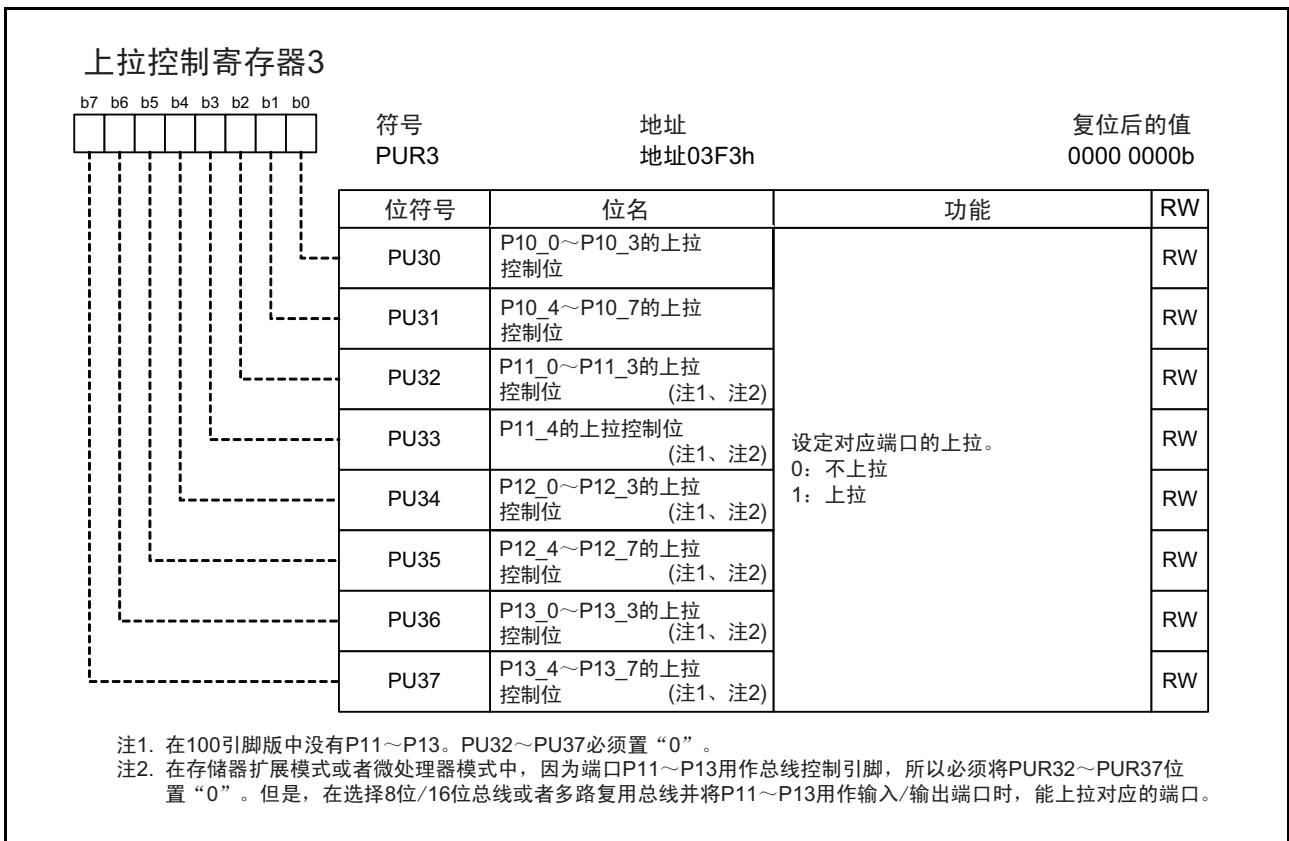


图 26.27 PUR3 寄存器

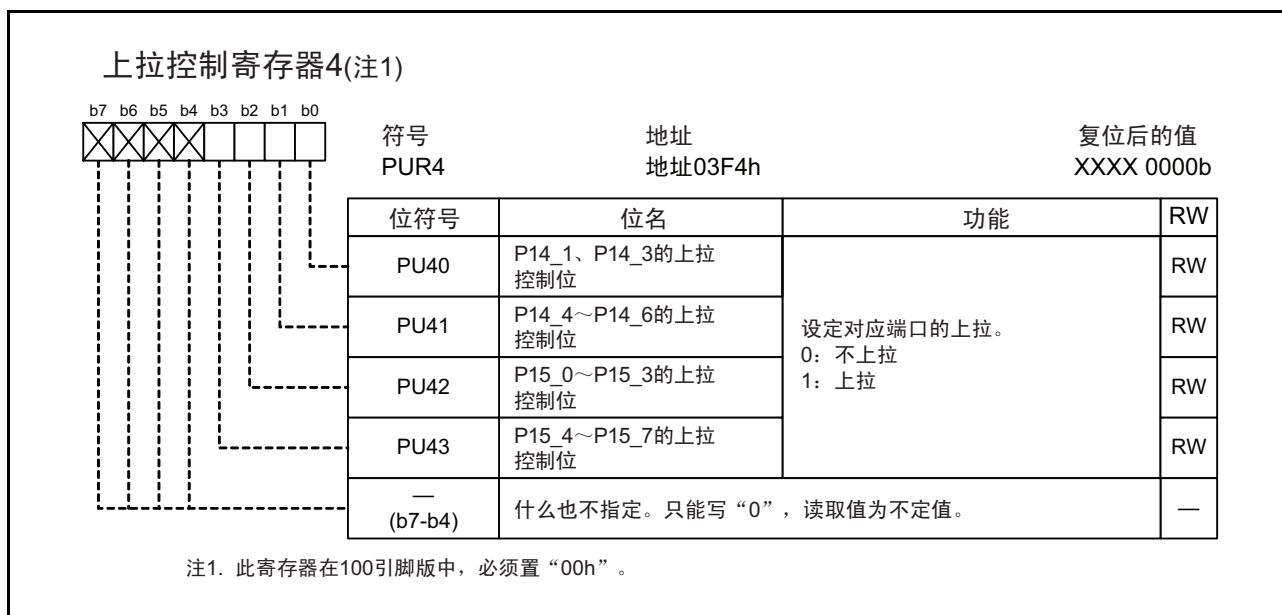


图 26.28 PUR4 寄存器

### 26.5 端口控制寄存器（PCR 寄存器）

PCR 寄存器如图 26.29 所示。

这是选择将端口 P1 的输出形式设定为推挽输出还是设定为虚拟 N 沟道漏极开路输出的寄存器。如果将 PCR0 位置“1”，就总是将输出缓冲器的 P 沟道晶体管截止。因为残留寄生二极管，端口 P1 无法完全地变为漏极开路，所以输入电压的绝对最大额定值为“-0.3V ~ VCC+0.3V”（参照图 26.30）。

在存储器扩展模式或者微处理器模式中，如果将端口 P1 用作数据总线，就必须将 PCR0 位置“0”；如果将端口 P1 用作可编程输入 / 输出端口或者外围功能输入 / 输出，就能通过 PCR0 位选择输出形式。

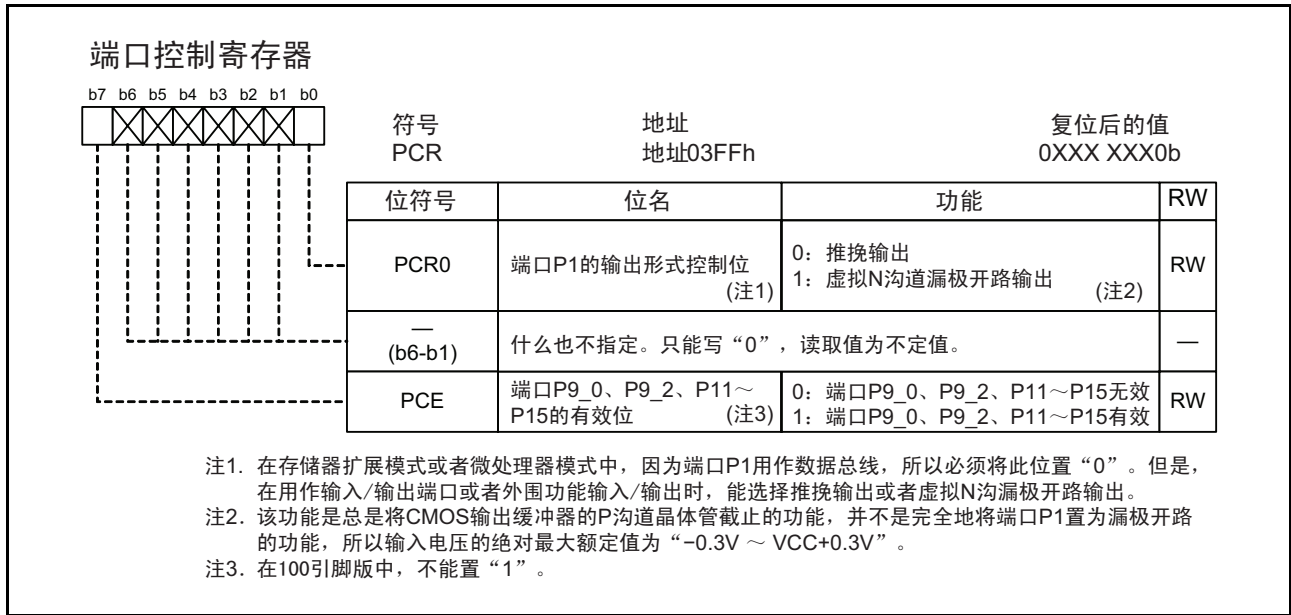


图 26.29 PCR 寄存器

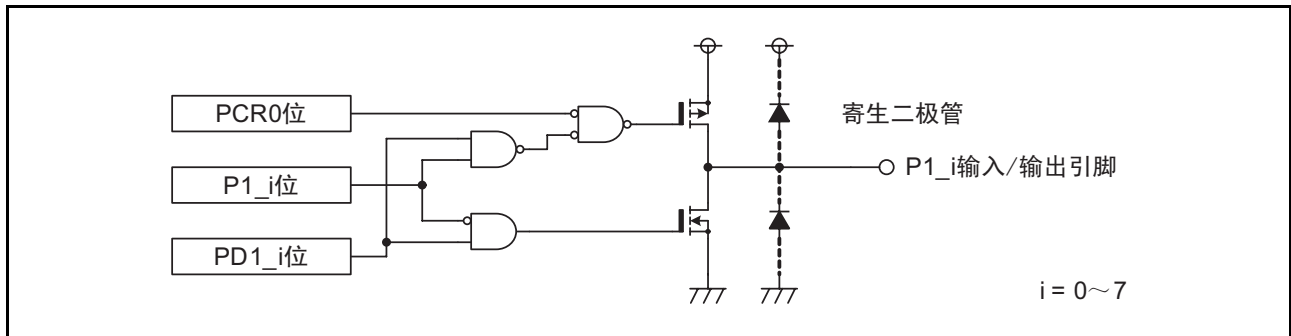


图 26.30 端口 P1 的输出缓冲器的结构

## 26.6 未使用引脚的处理

电路板上不使用的引脚的处理例子如表 26.2、表 26.3 和图 26.32 所示。

表 26.2 单芯片模式的未使用引脚的处理例子（注 1）

引脚名	处理内容
端口 P0 ~ P15 (P8_5、P9_1 (100 引脚版)、P14_1 (144 引脚版) 除外) (注 2、注 3、注 4)	设定为输入并且过电阻 (注 5) 将每个引脚通连接到 VSS (下拉)、或者设定为输出并且将引脚置为开路。
P9_1 (100 引脚版)	通过电阻 (注 5) 连接到 VSS (下拉)。
P14_1 (144 引脚版)	通过电阻 (注 5) 连接到 VSS (下拉)。
XOUT (注 6)	开路
$\overline{\text{NMI}}$ (P8_5)	通过电阻 (注 5) 连接到 VCC (上拉)。
AVCC	连接到 VCC。
AVSS、VREF	连接到 VSS。
NSD	通过 1k ~ 4.7k $\Omega$ 的电阻连接到 VCC (上拉)。

注 1. 必须尽量用较短的布线 (不超过 2cm) 处理单片机的未使用引脚。

注 2. 在设定为输出并且置为开路的情况下, 在从解除复位到端口变为输出期间, 端口为输入。因此, 在端口为输入期间, 引脚的电压电平不稳定, 电源电流有可能增加。考虑到因噪声或者噪声引起的失控而使方向寄存器的内容发生变化的情况, 建议通过软件定期地对方向寄存器的内容进行重新设定, 以提高程序的可靠性。

注 3. 端口 P11 ~ P15 只限于 144 引脚版。

注 4. 在 100 引脚版中, 必须给地址 03D7h、03DAh、03DBh、03DEh 和 03DFh 设定 “FFh”。

注 5. 必须根据系统选择适当的电阻值, 推荐值为 “10k ~ 100k $\Omega$ ”。

注 6. 这是给 XIN 引脚输入外部时钟的情况。

表 26.3 存储器扩展模式和微处理器模式的未使用引脚的处理例子（注 1）

引脚名	处理内容
端口 P1、P6 ~ P15 (P8_5、P9_1 (100 引脚版)、P14_1 (144 引脚版) 除外) (注 2、注 3、注 4)	设定为输入并且通过电阻 (注 5) 将每个引脚连接到 VSS (下拉)、或者设定为输出并且将引脚置为开路。
P9_1 (100 引脚版)	通过电阻 (注 5) 连接到 VSS (下拉)。
P14_1 (144 引脚版)	通过电阻 (注 5) 连接到 VSS (下拉)。
$\overline{\text{BC0}} \sim \overline{\text{BC3}}$ 、 $\overline{\text{WR0}} \sim \overline{\text{WR3}}$ 、ALE、HLDA、XOUT (注 6)、BCLK	开路
$\overline{\text{HOLD}}$ 、 $\overline{\text{RDY}}$	通过电阻 (注 5) 连接到 VCC (上拉)。
$\overline{\text{NMI}}$ (P8_5)	通过电阻 (注 5) 连接到 VCC (上拉)。
AVCC	连接到 VCC。
AVSS、VREF	连接到 VSS。
NSD	通过 1k ~ 4.7k $\Omega$ 的电阻连接到 VCC (上拉)。

注 1. 必须尽量用较短的布线 (不超过 2cm) 处理单片机的未使用引脚。

注 2. 在设定为输出并且置为开路的情况下, 在从解除复位到端口变为输出期间, 端口为输入。因此, 在端口为输入期间, 引脚的电压电平不稳定, 电源电压有可能增加。考虑到噪声或者噪声引起的失控等而使方向寄存器的内容发生变化的情况, 建议通过软件定期地对方向寄存器的内容进行重新设定, 以提高程序的可靠性。

注 3. 端口 P11 ~ P15 只限于 144 引脚版。

注 4. 在 100 引脚版中, 必须给地址 03D7h、03DAh、03DBh、03DEh 和 03DFh 设定 “FFh”。

注 5. 必须根据系统选择适当的电阻值, 推荐值为 “10k ~ 100k $\Omega$ ”。

注 6. 这是给 XIN 引脚输入外部时钟的情况。

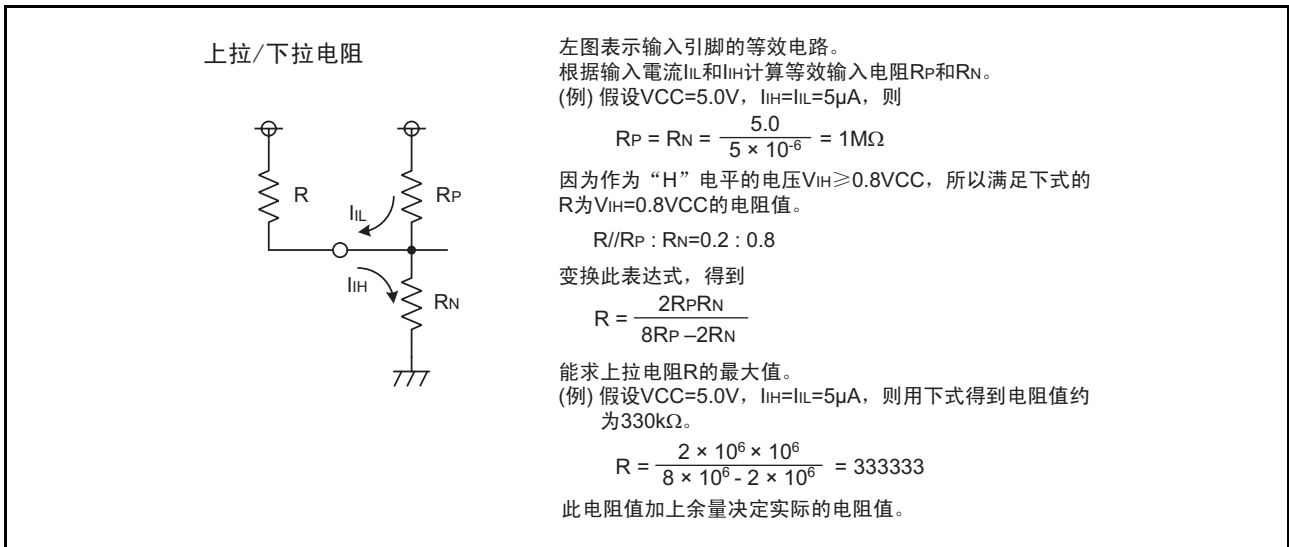


图 26.31 上拉 / 下拉电阻

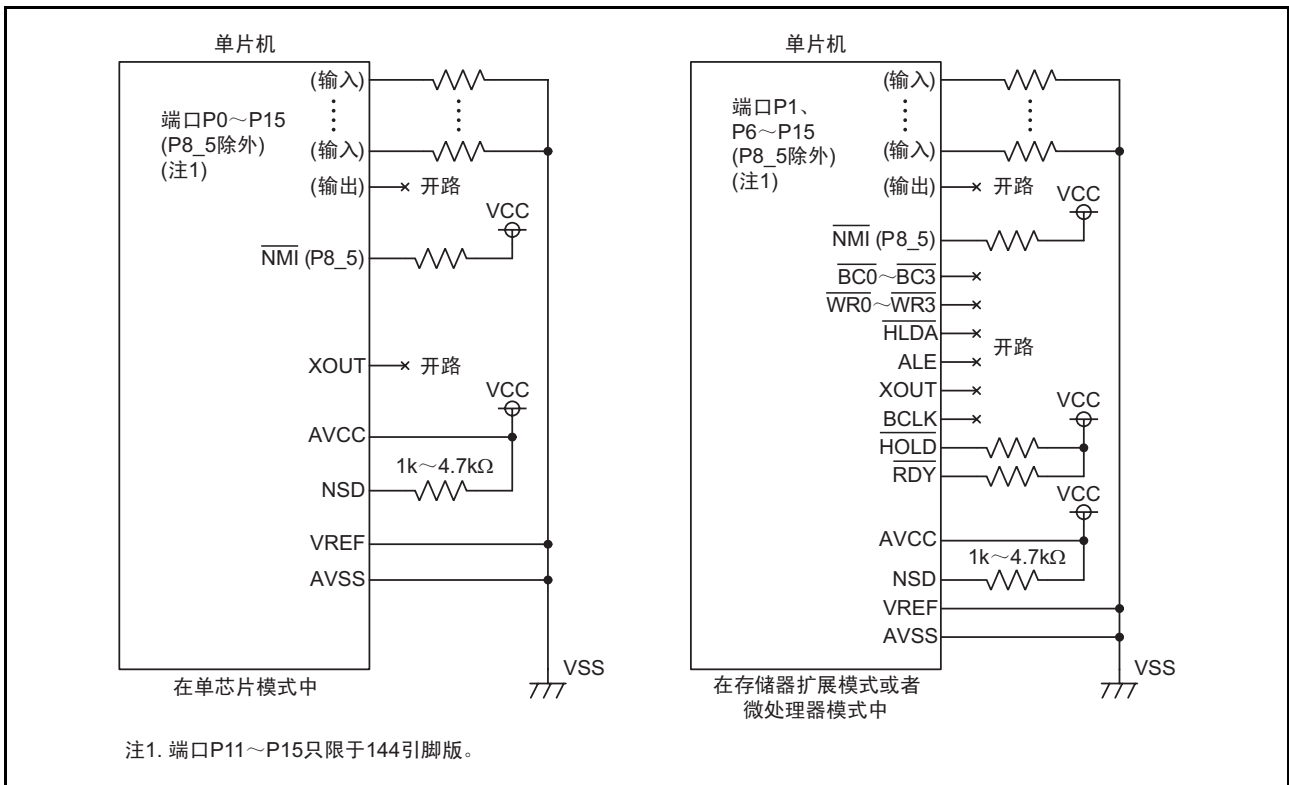


图 26.32 未使用引脚的处理例子

## 27. 闪存

### 27.1 概要

能在 CPU 改写模式、标准串行输入 / 输出模式和并行输入 / 输出模式共 3 种改写模式中改写闪存。  
闪存的规格以及各改写模式的概要分别如表 27.1 或者表 27.2 所示。

表 27.1 闪存的规格

项目	规格
闪存的改写模式	CPU 改写模式、标准串行输入 / 输出模式、并行输入 / 输出模式
块分割	请参照图 27.1。
编程单位	以 8 字节为单位。
擦除单位	以块为单位。
编程 / 擦除的控制方式	通过软件命令进行控制。
保护的种类	锁定位保护、ROM 码保护、ID 码保护
软件命令数	9 个

表 27.2 闪存的改写模式概要

改写模式	CPU 改写模式	标准串行输入 / 输出模式	并行输入 / 输出模式
概要	能通过 CPU 执行软件命令改写闪存。 EW0 模式： 能通过内部闪存以外的区域中的程序进行改写。 EW1 模式： 能通过改写对象块以外的区域中的程序进行改写。	使用专用串行编程器改写闪存。 标准串行输入 / 输出模式 1： 使用时钟同步串行接口 标准串行输入 / 输出模式 2： 使用异步串行接口	使用专用并行编程器改写闪存。
CPU 运行模式	单芯片模式、存储器扩展模式（EW0 模式）	标准串行输入 / 输出模式	并行输入 / 输出模式
ROM 编程器	—	串行编程器	并行编程器
板上编程	能	能	不能

内部闪存的框图如图 27.1 所示。

内部闪存有用于保存用户程序的块 0 ~ 17（程序区）以及用于保存用户程序运行结果数据的块 A 和块 B（数据区 / 数据闪存）。

各块独立，能通过设定锁定位禁止对各块进行改写和擦除。



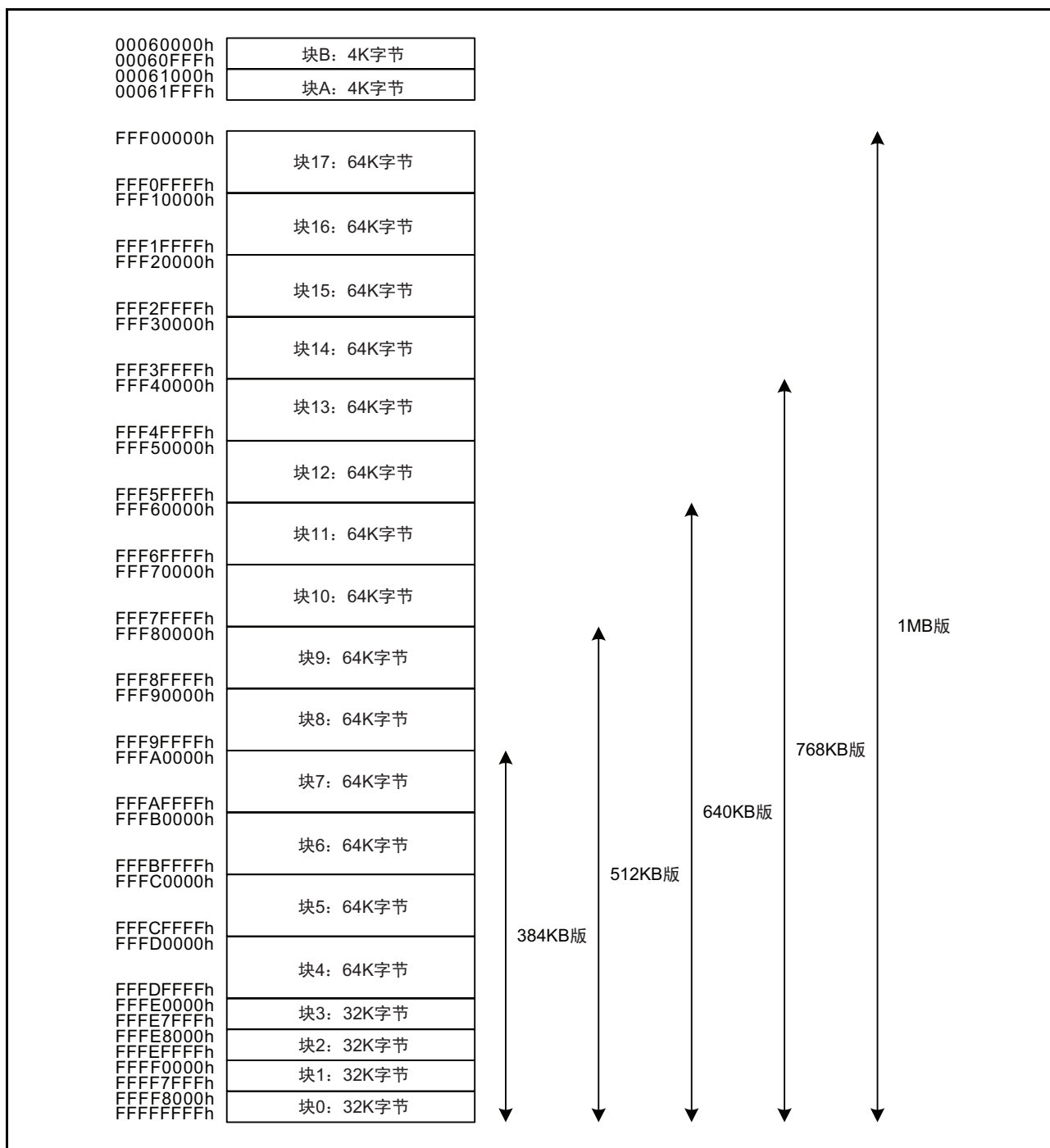


图 27.1 内部闪存的框图

## 27.2 闪存保护

保护的种类有 2 种，一种是保护闪存不会因程序失控而被意外地编程或者擦除，另一种是保护闪存不会被第三方读写。前者通过锁定位实现，后者通过保护位和 ID 码实现。

各种保护的特点如表 27.3 所示。

表 27.3 保护的种类和特点

保护	锁定位保护	ROM 码保护	ID 码保护
保护对象	擦除、编程	读、擦除、编程	读、编程
对象的改写模式	CPU 改写模式、 标准串行输入 / 输出模式、 并行输入 / 输出模式	并行输入 / 输出模式	标准串行输入 / 输出模式
保护范围	各块	闪存的全区域	闪存的全区域
保护方法	改写锁定位。	改写任意块的保护位。	写被设定 ID 码的程序。
解除方法	将 FMR 寄存器的 LBD 位置 “1” (锁定位无效)。 通常，在进行上述操作后擦除该块。	使用串行编程器擦除保护位被 置位的全部块。	从串行编程器输入正确的 ID 码。

### 27.2.1 锁定位保护

锁定位保护对全部的改写模式有效。在锁定位保护有效时，不能改写和擦除锁定位为 “0” (锁定) 的块。

要将锁定位 “0” 时，必须发行软件命令的锁定位编程命令。如果将 FMR1 寄存器的 LBD 位置 “1” (锁定位无效)，锁定位保护就无效，因而能改写和擦除全部的块。如果擦除锁定位为 “0” (锁定) 的块，该块的锁定位也被擦除而变为 “1” (非锁定)。

### 27.2.2 ROM 码保护

ROM 码保护对并行输入 / 输出模式有效。在 ROM 码保护有效时，并行编程器无法读写任意区域的内容。要解除 ROM 码保护时，必须使用串行编程器擦除保护位为 “0” (保护) 的全部块。

闪存各块各有 2 个保护位。通过软件命令指定的保护位的地址如表 27.4 所示。只要将这些保护位中的任意 1 位置 “0” (保护)，就能保护全区域。

表 27.4 保护位的地址一览表

块	保护位 0	保护位 1
BlockB	00060100h	00060300h
BlockA	00061100h	00061300h
Block17	FFF00100h	FFF00300h
Block16	FFF10100h	FFF10300h
Block15	FFF20100h	FFF20300h
Block14	FFF30100h	FFF30300h
Block13	FFF40100h	FFF40300h
Block12	FFF50100h	FFF50300h
Block11	FFF60100h	FFF60300h
Block10	FFF70100h	FFF70300h
Block9	FFF80100h	FFF80300h
Block8	FFF90100h	FFF90300h
Block7	FFFA0100h	FFFA0300h
Block6	FFFB0100h	FFFB0300h
Block5	FFFC0100h	FFFC0300h
Block4	FFFD0100h	FFFD0300h
Block3	FFFE0100h	FFFE0300h
Block2	FFFE8100h	FFFE8300h
Block1	FFFF0100h	FFFF0300h
Block0	FFFF8100h	FFFF8300h

### 27.2.3 ID 码保护

ID 码保护对标准串行输入 / 输出模式有效。如果串行编程器送来的 7 字节的 ID 码和闪存中的 ID 码相同，就接受串行编程器的命令。但是，当复位向量为“FFFFFFFFh”时，就判断为已经擦除闪存而不进行 ID 码检查。在 ROM 码保护有效并且复位向量为“FFFFFFFFh”时，只接受擦除命令。

按照 ID1、ID2、……、ID7 的顺序，从串行编程器送出 ID 码。ID 码如图 27.2 所示，从 ID1 开始按顺序将 ID 码分配到地址 FFFFFFFE8h、地址 FFFFFFFE9h、……、地址 FFFFFFFEh。在将给这些地址设定 ID 码的程序写到闪存时，ID 码保护有效。

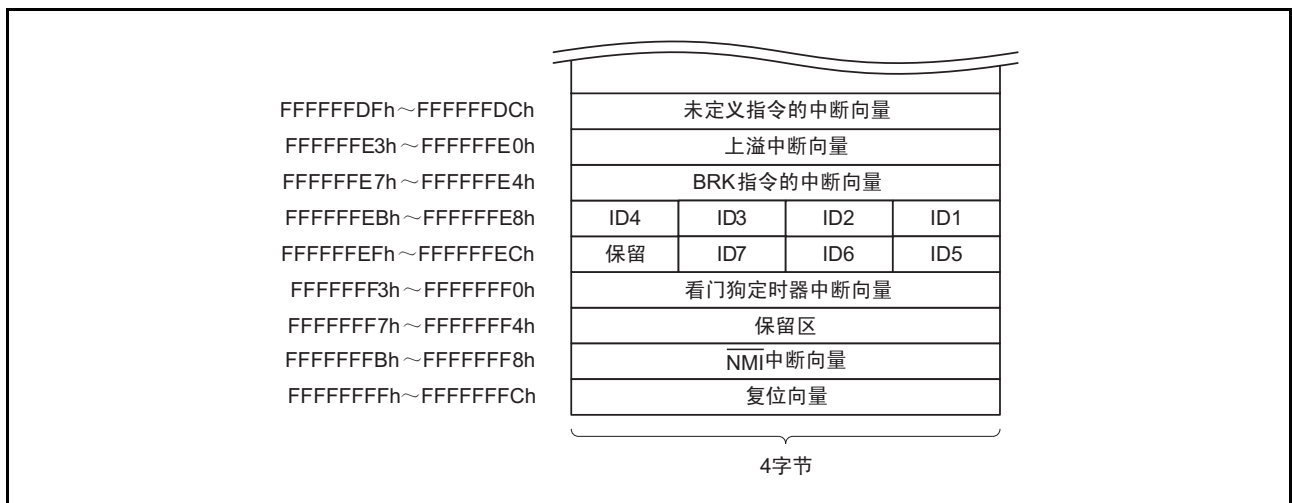


图 27.2 ID 码的保存地址

### 27.3 CPU 改写模式

CPU 改写模式通过 CPU 执行软件命令改写闪存。在 CPU 改写模式中，不从 CPU 总线直接存取闪存，而经过改写闪存的专用总线存取闪存（参照图 27.3）。

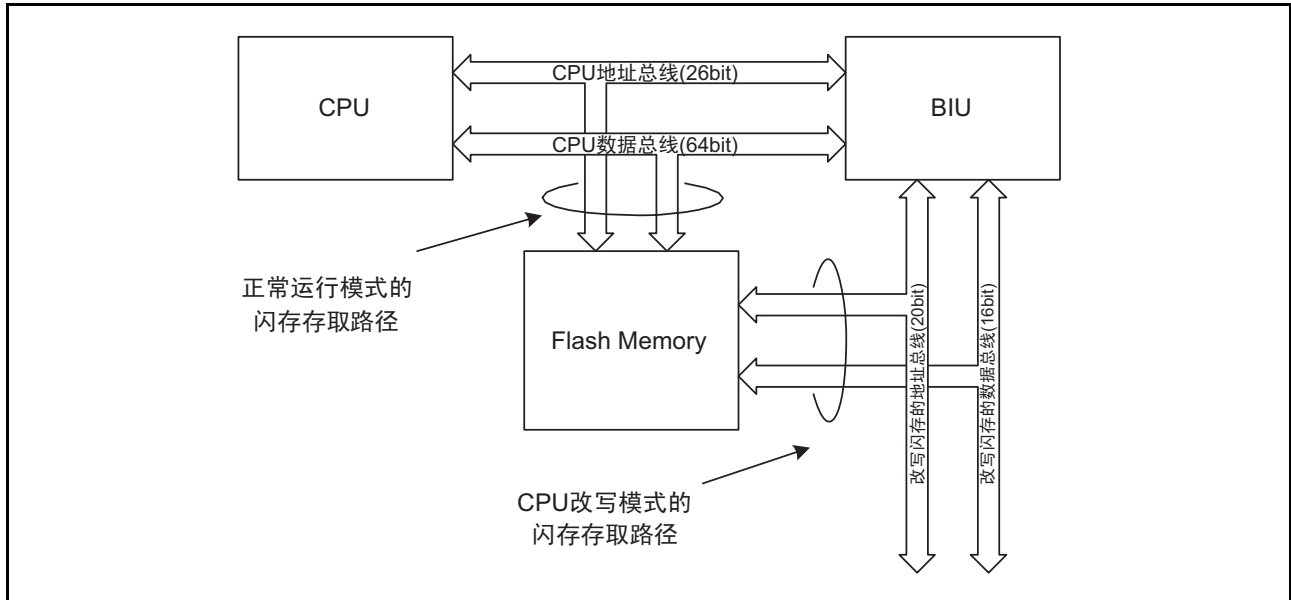


图 27.3 CPU 改写模式的闪存存取路径

通过 FEBC0 寄存器和 FEBC3 寄存器对改写闪存的专用总线进行设定。必须参照“27.3.2 改写闪存的总线时序设定”和“28. 电特性”，对 FEBC0 寄存器和 FEBC3 寄存器进行设定以满足改写条件。FEBC0 寄存器和 EBC0 寄存器、FEBC3 寄存器和 EBC3 寄存器分别共用地址，如果改写 FEBC0 寄存器和 FEBC3 寄存器，外部总线的设定就会发生变化。必须在改写闪存后，重新设定 EBC0 寄存器和 EBC3 寄存器。

CPU 改写模式有 EW0 模式和 EW1 模式，EW0 模式和 EW1 的差异如表 27.5 所示。

表 27.5 EW0 模式和 EW1 模式的差异

项目	EW0 模式	EW1 模式
CPU 运行模式	单芯片模式、 存储器扩展模式（注 1）	单芯片模式
能执行改写程序的区域	内部闪存以外的区域	改写对象块以外的内部闪存、内部 RAM
软件命令的限制	无	<ul style="list-style-type: none"> <li>编程 / 擦除命令 禁止对保存改写程序的块执行。</li> <li>读状态寄存器命令 禁止执行。</li> <li>读锁定位状态模式的转移命令 必须在 RAM 中执行。</li> <li>读保护位状态模式的转移命令 必须在 RAM 中执行。</li> </ul>
编程或者擦除后的模式	读状态寄存器模式	读阵列模式
编程或者擦除时的 CPU 状态	运行	保持（输入 / 输出端口保持发行命令前的状态）
闪存的状态检测	<ul style="list-style-type: none"> <li>通过程序读 FMSR0 寄存器。</li> <li>执行读状态寄存器命令并读数据。</li> </ul>	<ul style="list-style-type: none"> <li>通过程序读 FMSR0 寄存器。</li> </ul>
其他限制	无	<ul style="list-style-type: none"> <li>必须在编程或者擦除时禁止中断（NMI 除外）和 DMA。</li> </ul>

注 1. 在使用存储器扩展模式时， $\overline{CS0}$  区域和  $\overline{CS3}$  区域有使用限制。详细内容请参照“27.3.1 CPU 运行模式和闪存的改写”。

如果将 FMCR 寄存器的 FEW 位置“1”，就进入 CPU 改写模式。此后，能根据 FMR0 寄存器的 EWM 位的设定值选择 EW0 模式或者 EW1 模式。

FMCR 寄存器和 FMR0 寄存器分别由 PRR 寄存器和 FPR0 寄存器保护。

相关寄存器如图 27.4 ~ 图 27.12 所示。

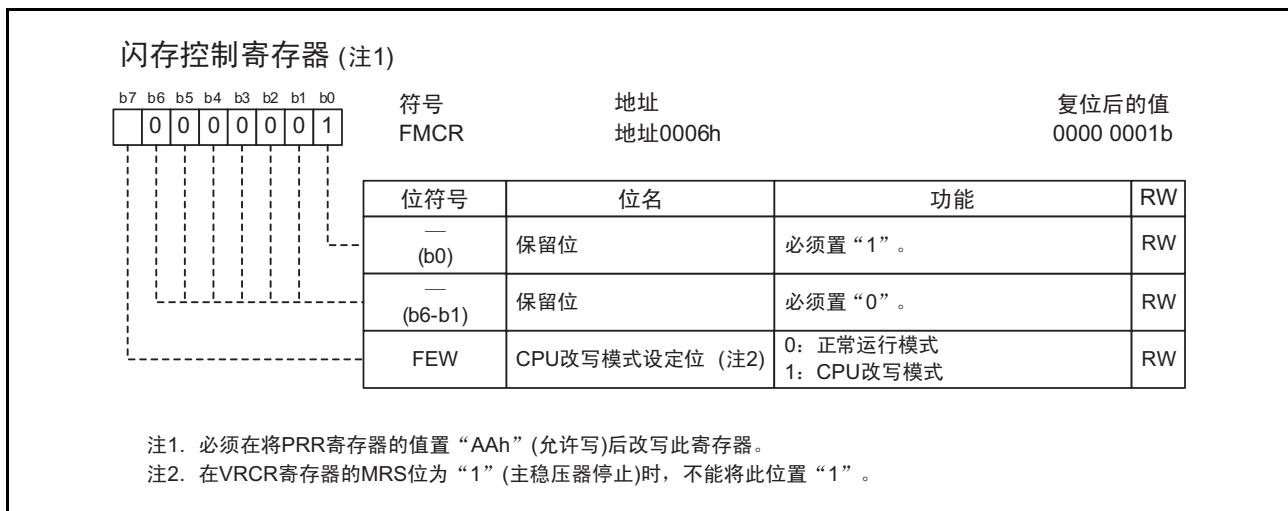


图 27.4 FMCR 寄存器

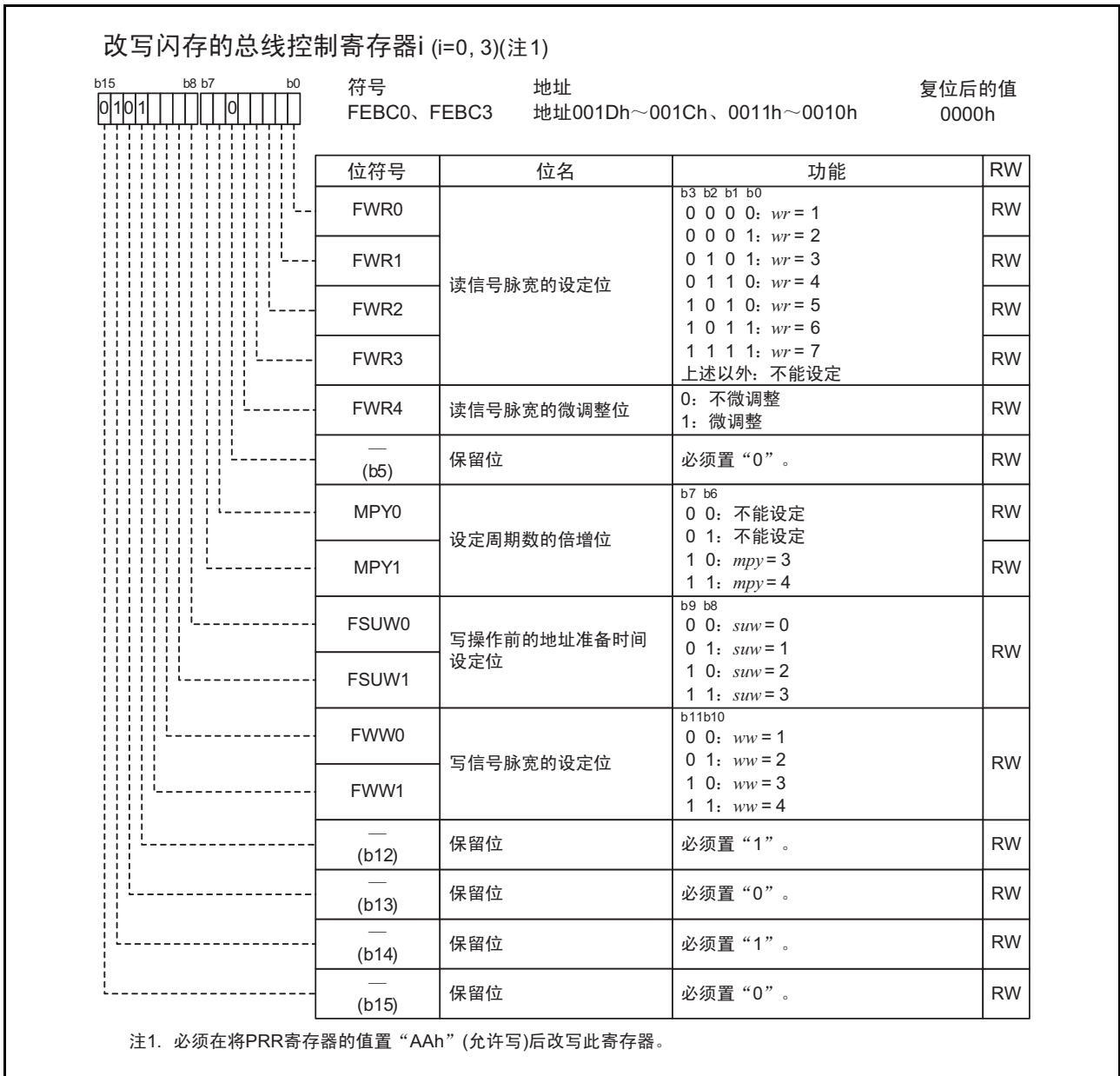


图 27.5 FEBC0 寄存器和 FEBC3 寄存器

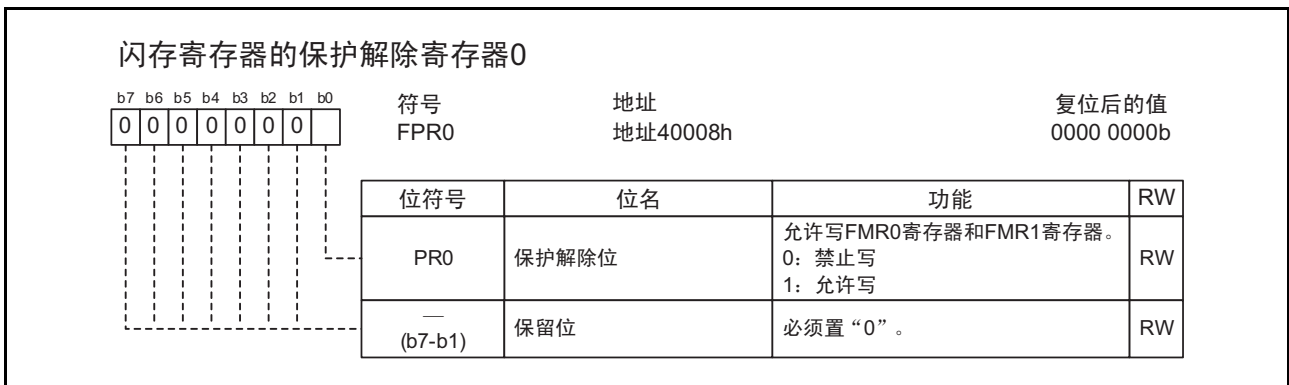


图 27.6 FPR0 寄存器

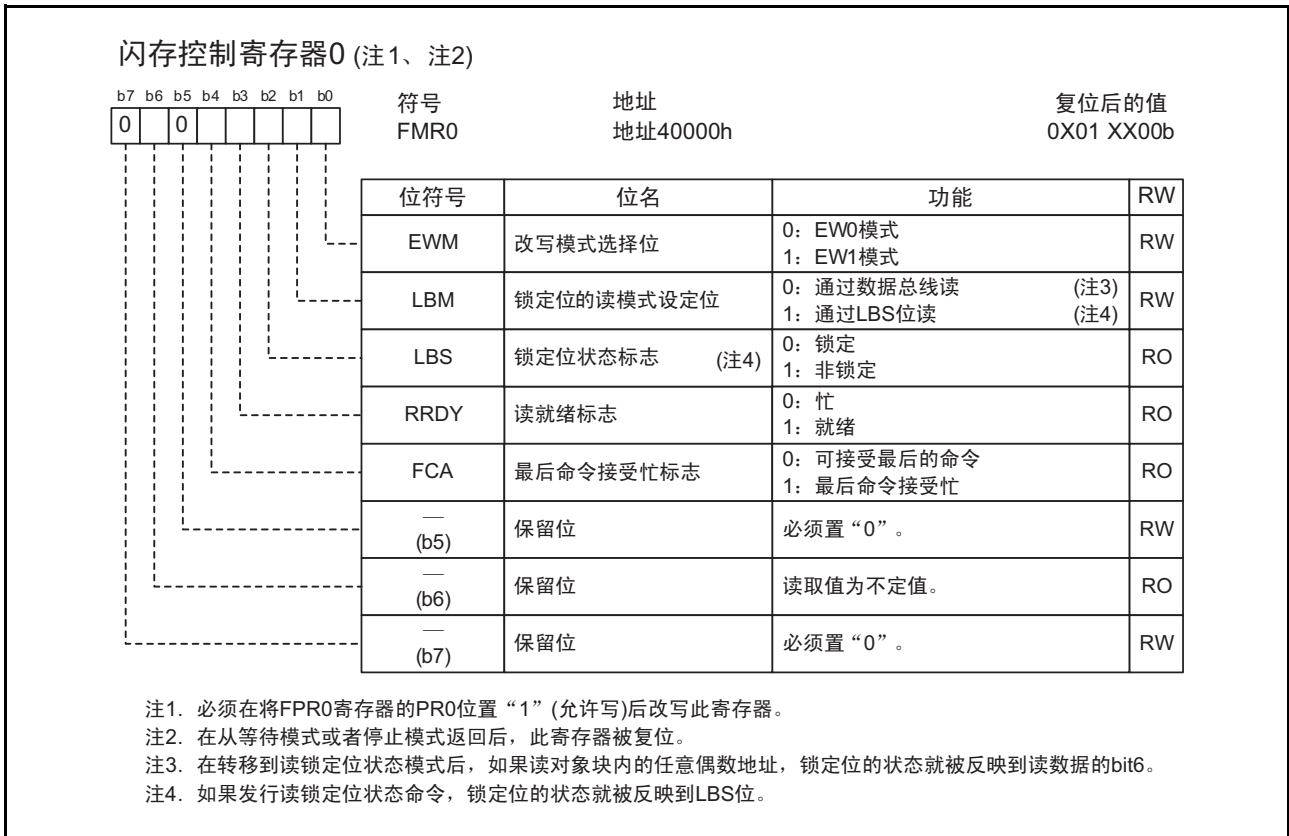


图 27.7 FMR0 寄存器

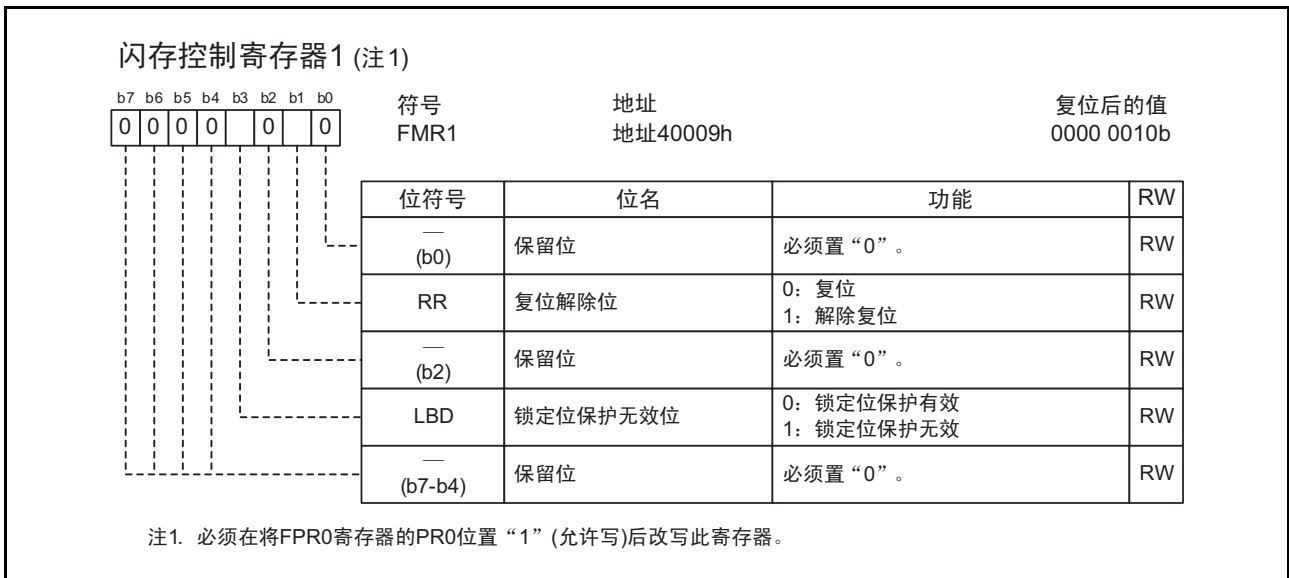


图 27.8 FMR1 寄存器



图 27.9 FMSR0 寄存器

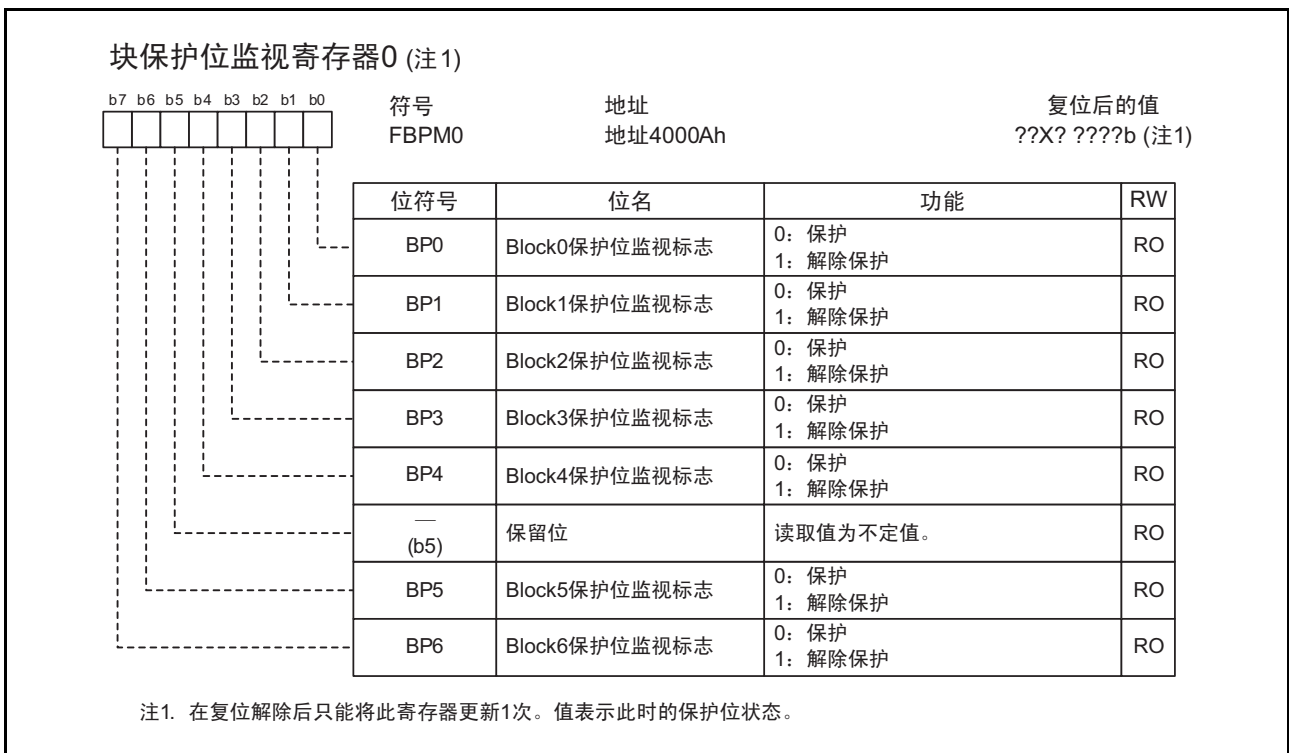


图 27.10 FBPM0 寄存器



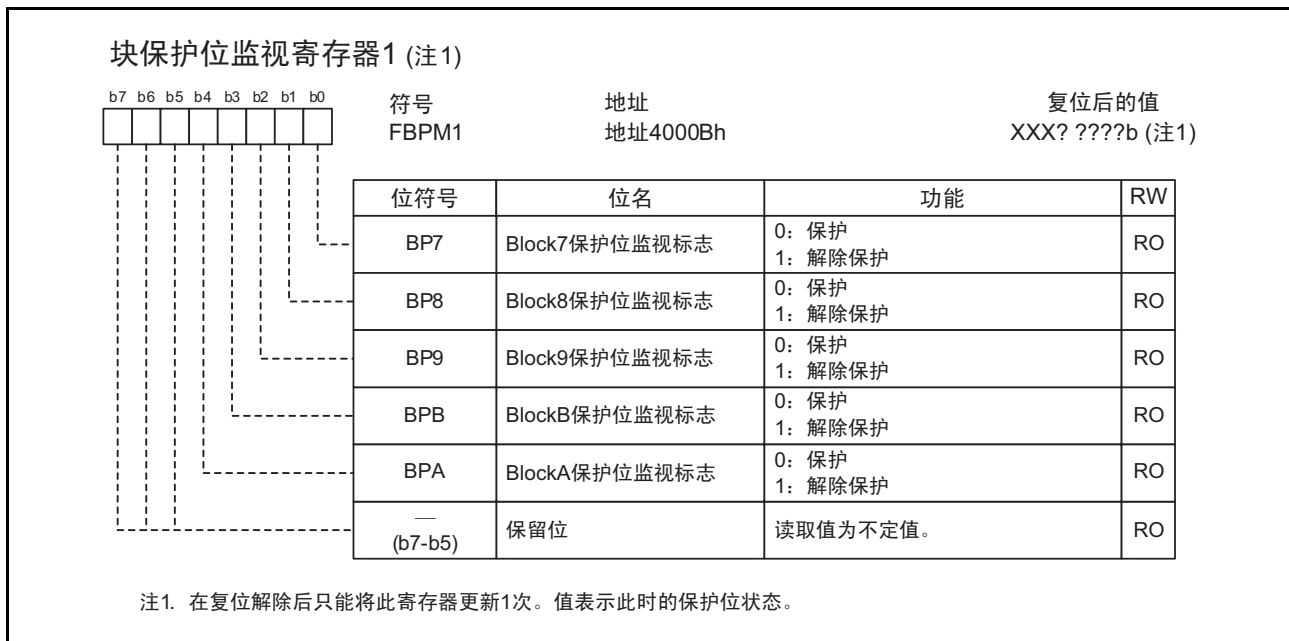


图 27.11 FBPM1 寄存器

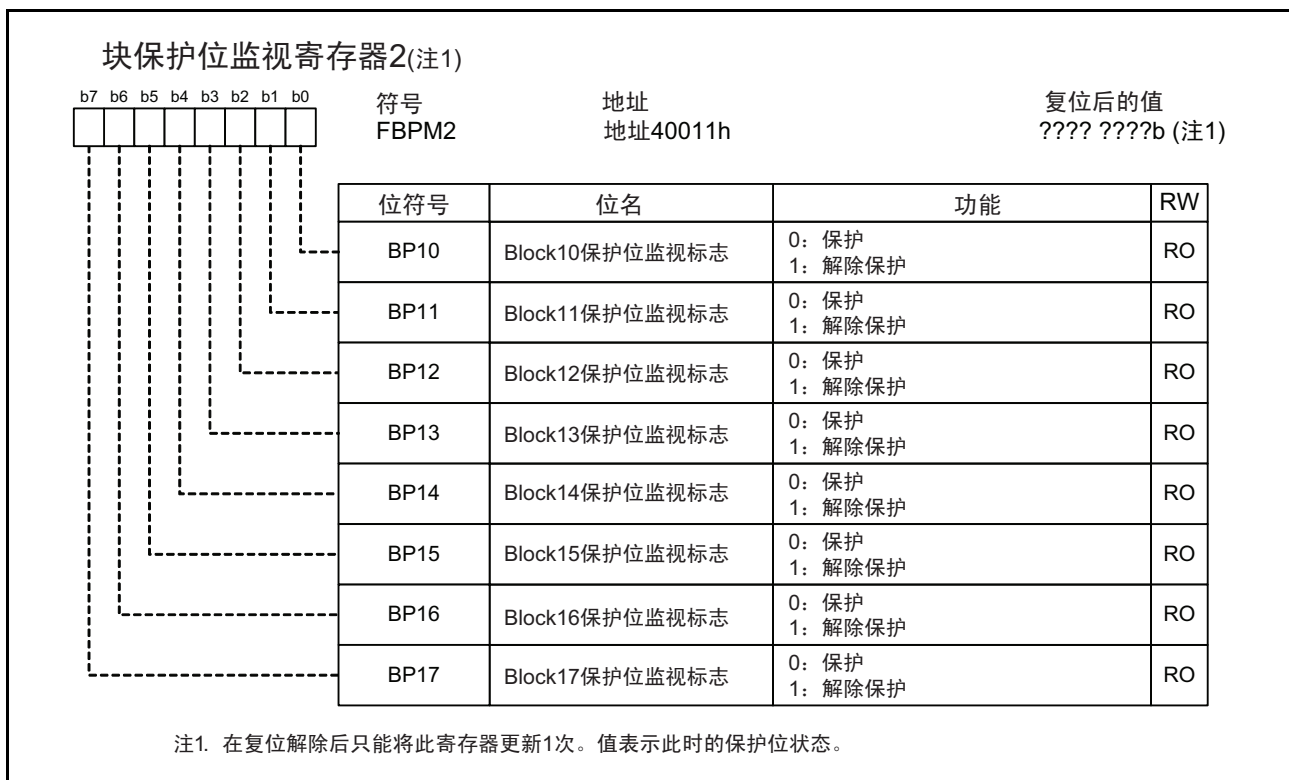


图 27.12 FBPM2 寄存器

### 27.3.1 CPU 运行模式和闪存的改写

改写闪存的总线时序设定寄存器因 CPU 运行模式而不同。

在不使用存储器扩展模式而只使用单芯片模式时，CB01、CB12、CB23 寄存器必须保持复位后的值（“00h”）而不能更改。此时，程序区和数据区都能通过 FEBC0 寄存器进行总线的设定。

在使用存储器扩展模式时，必须根据各寄存器的设定范围更改 CB01、CB12、CB23 寄存器的值。此时，程序区和数据区能分别通过 FEBC0 寄存器和 FEBC3 寄存器进行总线的设定。

存储器扩展模式中使用的 EBC0 寄存器、EBC3 寄存器分别和 FEBC0 寄存器、FEBC3 寄存器共用相同的寄存器，如果为了改写闪存而更改 FEBC<sub>i</sub> 寄存器（i=0,3），EBC<sub>i</sub> 寄存器的设定值也发生变化。因此在 CPU 改写模式中，有可能不能存取分配在 CS0 区域和 CS3 区域的外部器件。

有关 CPU 运行模式和闪存的改写的详细内容和限制事项请参照表 27.6。

表 27.6 CPU 运行模式和闪存的改写

项目	CPU 运行模式	
	单芯片模式	存储器扩展模式
CB01 寄存器	保持“00h”而不能更改。	设定值必须在“04h ~ F8h”范围内并且大于等于 CB12 寄存器的值。
CB12 寄存器	保持“00h”而不能更改。	设定值必须在“03h ~ F7h”范围内，并且大于等于 CB23 寄存器的值、小于等于 CB01 寄存器的值。
CB23 寄存器	保持“00h”而不能更改。	设定值必须在“02h ~ F6h”范围内并且小于等于 CB12 寄存器的值。
程序区的总线设定	FEBC0 寄存器	FEBC0 寄存器
数据区的总线设定	FEBC0 寄存器	FEBC3 寄存器
设定 FEBC <sub>i</sub> 寄存器后的 CS0 区域和 CS3 区域的状态	—	<ul style="list-style-type: none"> <li>分离总线</li> <li>16 位宽</li> <li>忽视 RDY 信号。</li> </ul>
限制事项	无	<ul style="list-style-type: none"> <li>忽视 HOLD 信号。</li> <li>如果在多路复用总线上使用 CS0 区域或者 CS3 区域，就不能在 CPU 改写模式中存取外部器件。</li> <li>因为 CS0 区域和 CS3 区域的总线时序会发生变化，所以有可能不能存取外部器件。</li> </ul>

### 27.3.2 改写闪存的总线时序设定

通过 FEBC0 寄存器和 FEBC3 寄存器对改写闪存的专用总线进行设定。在此对 FEBC0 寄存器和 FEBC3 寄存器的设定进行详细说明。

总线时序的基准时钟是通过 CCR 寄存器的 BCD1 ~ BCD0 位设定的基本时钟。用基本时钟的周期数规定  $t_{su}$ 、 $t_w$ 、 $t_c$ 、 $t_h$  等时间。

闪存的读时序和写时序分别如图 27.13 和图 27.14 所示，各外围总线时钟分频值的 MPY1 ~ MPY0 位和 FWR4 ~ FWR0 位的设定值与读周期的关系如表 27.7 ~ 表 27.9 所示，各外围总线时钟分频值的 MPY1 ~ MPY0 位、FSUW1 ~ FSUW0 位和 FWW1 ~ FWW0 位的设定值与写周期的关系如表 27.10 ~ 表 27.12 所示。

从这些表中选择符合电特性所示的 CPU 改写模式时序条件的读周期时序和写周期时序。

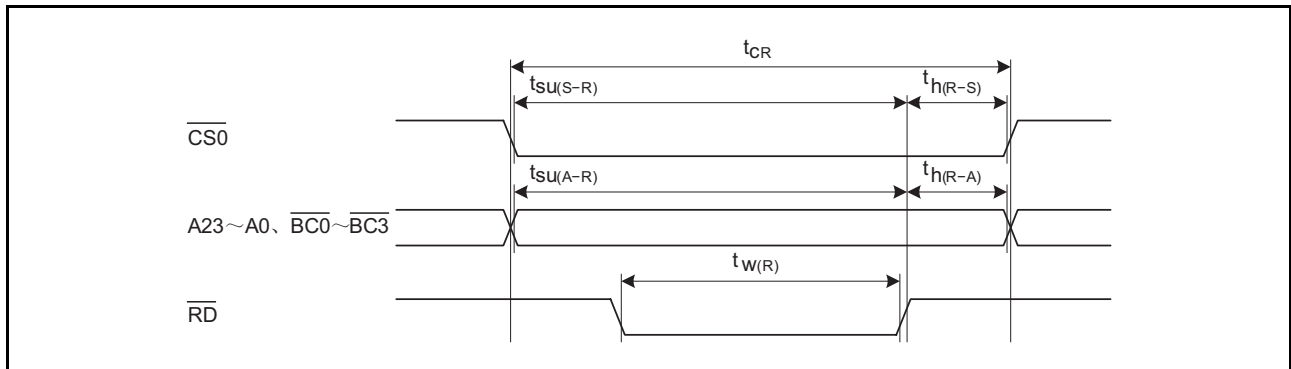


图 27.13 读时序

表 27.7 外围总线时钟 2 分频时的 MPY1 ~ MPY0 位和 FWR4 ~ FWR0 位的设定值与读周期的关系（单位：周期）

FWR3 ~ FWR0 位的设定值		FWR4 位的设定值	MPY1 ~ MPY0 位的设定值							
			10b				11b			
			$mpy=3$				$mpy=4$			
			$t_{su(S-R)}$ 、 $t_{su(A-R)}$	$t_w(R)$	$t_{cR}$	$t_h(R-S)$ 、 $t_h(R-A)$	$t_{su(S-R)}$ 、 $t_{su(A-R)}$	$t_w(R)$	$t_{cR}$	$t_h(R-S)$ 、 $t_h(R-A)$
0000b	$wr=1$	0	4	3	4	0	6	5	6	0
		1	6	5	6	0	6	5	6	0
0001b	$wr=2$	0	8	7	8	0	10	9	10	0
		1	8	7	8	0	10	9	10	0
0101b	$wr=3$	0	10	9	10	0	14	13	14	0
		1	12	11	12	0	14	13	14	0
0110b	$wr=4$	0	14	13	14	0	18	17	18	0
		1	14	13	14	0	18	17	18	0
1010b	$wr=5$	0	16	15	16	0	22	21	22	0
		1	18	17	18	0	22	21	22	0
1011b	$wr=6$	0	20	19	20	0	26	25	26	0
		1	20	19	20	0	26	25	26	0
1111b	$wr=7$	0	22	21	22	0	30	29	30	0
		1	24	23	24	0	30	29	30	0

表 27.8 外围总线时钟 3 分频时的 MPY1 ~ MPY0 位和 FWR4 ~ FWR0 位的设定值与读周期的关系（单位：周期）

FWR3 ~ FWR0 位的设定值		FWR4 位的设定值	MPY1 ~ MPY0 位的设定值								
			10b				11b				
			mpy=3				mpy=4				
			$t_{su(S-R)}$ 、 $t_{su(A-R)}$	$t_{w(R)}$	$t_{cR}$	$t_{h(R-S)}$ 、 $t_{h(R-A)}$	$t_{su(S-R)}$ 、 $t_{su(A-R)}$	$t_{w(R)}$	$t_{cR}$	$t_{h(R-S)}$ 、 $t_{h(R-A)}$	
0000b	wr=1	0	6	4.5	6	0	6	4.5	6	0	
		1	6	4.5	6	0	6	4.5	6	0	
0001b	wr=2	0	9	7.5	9	0	9	7.5	9	0	
		1	9	7.5	9	0	12	10.5	12	0	
0101b	wr=3	0	12	10.5	12	0	15	13.5	15	0	
		1	12	10.5	12	0	15	13.5	15	0	
0110b	wr=4	0	15	13.5	15	0	18	16.5	18	0	
		1	15	13.5	15	0	18	16.5	18	0	
1010b	wr=5	0	18	16.5	18	0	21	19.5	21	0	
		1	18	16.5	18	0	24	22.5	24	0	
1011b	wr=6	0	21	19.5	21	0	27	25.5	27	0	
		1	21	19.5	21	0	27	25.5	27	0	
1111b	wr=7	0	24	22.5	24	0	30	28.5	30	0	
		1	24	22.5	24	0	30	28.5	30	0	

表 27.9 外围总线时钟 4 分频时的 MPY1 ~ MPY0 位和 FWR4 ~ FWR0 位的设定值与读周期的关系（单位：周期）

FWR3 ~ FWR0 位的设定值		FWR4 位的设定值	MPY1 ~ MPY0 位的设定值								
			10b				11b				
			mpy=3				mpy=4				
			$t_{su(S-R)}$ 、 $t_{su(A-R)}$	$t_{w(R)}$	$t_{cR}$	$t_{h(R-S)}$ 、 $t_{h(R-A)}$	$t_{su(S-R)}$ 、 $t_{su(A-R)}$	$t_{w(R)}$	$t_{cR}$	$t_{h(R-S)}$ 、 $t_{h(R-A)}$	
0000b	wr=1	0	4	2	4	0	8	6	8	0	
		1	8	6	8	0	8	6	8	0	
0001b	wr=2	0	8	6	8	0	12	10	12	0	
		1	8	6	8	0	12	10	12	0	
0101b	wr=3	0	12	10	12	0	16	14	16	0	
		1	12	10	12	0	16	14	16	0	
0110b	wr=4	0	16	14	16	0	20	18	20	0	
		1	16	14	16	0	20	18	20	0	
1010b	wr=5	0	16	14	16	0	24	22	24	0	
		1	20	18	20	0	24	22	24	0	
1011b	wr=6	0	20	18	20	0	28	26	28	0	
		1	20	18	20	0	28	26	28	0	
1111b	wr=7	0	24	22	24	0	32	30	32	0	
		1	24	22	24	0	32	30	32	0	

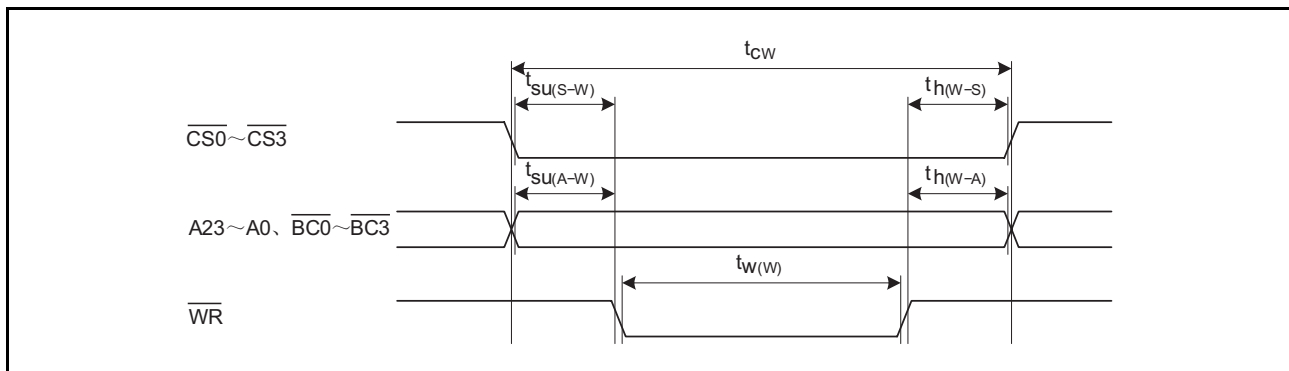


图 27.14 写时序

表 27.10 外围总线时钟 2 分频时的 MPY1 ~ MPY0 位、FSUW1 ~ FSUW0 位和 FWW1 ~ FWW0 位的设定值与写周期的关系（单位：周期）

FSUW1 ~ FSUW0 位的设定值		FWW1 ~ FWW0 位的设定值		MPY1 ~ MPY0 位的设定值							
				10b				11b			
				<i>mpy=3</i>				<i>mpy=4</i>			
				$t_{su(S-W)}$ 、 $t_{su(A-W)}$	$t_{w(W)}$	$t_{cW}$	$t_{h(W-S)}$ 、 $t_{h(W-A)}$	$t_{su(S-W)}$ 、 $t_{su(A-W)}$	$t_{w(W)}$	$t_{cW}$	$t_{h(W-S)}$ 、 $t_{h(W-A)}$
00b	<i>suw=0</i>	00b	<i>ww=1</i>	1	3	6	2	1	4	6	1
		01b	<i>ww=2</i>	1	6	8	1	1	8	10	1
		10b	<i>ww=3</i>	1	9	12	2	1	12	14	1
		11b	<i>ww=4</i>	1	12	14	1	1	16	18	1
01b	<i>suw=1</i>	00b	<i>ww=1</i>	4	3	8	1	5	4	10	1
		01b	<i>ww=2</i>	4	6	12	2	5	8	14	1
		10b	<i>ww=3</i>	4	9	14	1	5	12	18	1
		11b	<i>ww=4</i>	4	12	18	2	5	16	22	1
10b	<i>suw=2</i>	00b	<i>ww=1</i>	7	3	12	2	9	4	14	1
		01b	<i>ww=2</i>	7	6	14	1	9	8	18	1
		10b	<i>ww=3</i>	7	9	18	2	9	12	22	1
		11b	<i>ww=4</i>	7	12	20	1	9	16	26	1
11b	<i>suw=3</i>	00b	<i>ww=1</i>	10	3	14	1	13	4	18	1
		01b	<i>ww=2</i>	10	6	18	2	13	8	22	1
		10b	<i>ww=3</i>	10	9	20	1	13	12	26	1
		11b	<i>ww=4</i>	10	12	24	2	13	16	30	1

表 27.11 外围总线时钟 3 分频时的 MPY1 ~ MPY0 位、FSUW1 ~ FSUW0 位和 FWW1 ~ FWW0 位的设定值与写周期的关系（单位：周期）

FSUW1 ~ FSUW0 位的设定值		FWW1 ~ FWW0 位的设定值		MPY1 ~ MPY0 位的设定值							
				10b				11b			
				mpy=3				mpy=4			
				$t_{su(S-W)}$ 、 $t_{su(A-W)}$	$t_w(W)$	$t_{cW}$	$t_h(W-S)$ 、 $t_h(W-A)$	$t_{su(S-W)}$ 、 $t_{su(A-W)}$	$t_w(W)$	$t_{cW}$	$t_h(W-S)$ 、 $t_h(W-A)$
00b	suw=0	00b	ww=1	1	3	6	2	1	4	6	1
		01b	ww=2	1	6	9	2	1	8	12	3
		10b	ww=3	1	9	12	2	1	12	15	2
		11b	ww=4	1	12	15	2	1	16	18	1
01b	suw=1	00b	ww=1	4	3	9	2	6	3	12	3
		01b	ww=2	4	6	12	2	6	7	15	2
		10b	ww=3	4	9	15	2	6	11	18	1
		11b	ww=4	4	12	18	2	6	15	24	3
10b	suw=2	00b	ww=1	7	3	12	2	9	4	15	2
		01b	ww=2	7	6	15	2	9	8	18	1
		10b	ww=3	7	9	18	2	9	12	24	3
		11b	ww=4	7	12	21	2	9	16	27	2
11b	suw=3	00b	ww=1	10	3	15	2	13	4	18	1
		01b	ww=2	10	6	18	2	13	8	24	3
		10b	ww=3	10	9	21	2	13	12	27	2
		11b	ww=4	10	12	24	2	13	16	30	1

表 27.12 外围总线时钟 4 分频时的 MPY1 ~ MPY0 位、FSUW1 ~ FSUW0 位和 FWW1 ~ FWW0 位的设定值与写周期的关系（单位：周期）

FSUW1 ~ FSUW0 位的设定值		FWW1 ~ FWW0 位的设定值		MPY1 ~ MPY0 位的设定值							
				10b				11b			
				mpy=3				mpy=4			
				$t_{su(S-W)}$ 、 $t_{su(A-W)}$	$t_w(W)$	$t_{cW}$	$t_h(W-S)$ 、 $t_h(W-A)$	$t_{su(S-W)}$ 、 $t_{su(A-W)}$	$t_w(W)$	$t_{cW}$	$t_h(W-S)$ 、 $t_h(W-A)$
00b	suw=0	00b	ww=1	1	3	8	4	1	4	8	3
		01b	ww=2	1	6	8	1	1	8	12	3
		10b	ww=3	1	9	12	2	1	12	16	3
		11b	ww=4	1	12	16	3	1	16	20	3
01b	suw=1	00b	ww=1	4	3	8	1	5	4	12	3
		01b	ww=2	4	6	12	2	5	8	16	3
		10b	ww=3	4	9	16	3	5	12	20	3
		11b	ww=4	4	12	20	4	5	16	24	3
10b	suw=2	00b	ww=1	8	2	12	2	9	4	16	3
		01b	ww=2	8	5	16	3	9	8	20	3
		10b	ww=3	8	8	20	4	9	12	24	3
		11b	ww=4	8	11	20	1	9	16	28	3
11b	suw=3	00b	ww=1	10	3	16	3	13	4	20	3
		01b	ww=2	10	6	20	4	13	8	24	3
		10b	ww=3	10	9	20	1	13	12	28	3
		11b	ww=4	10	12	24	2	13	16	32	3

### 27.3.3 软件命令

在 CPU 改写模式中，如果对闪存发行软件命令，就能改写和擦除闪存。

必须以 16 位为单位写命令和读数据。

软件命令一览表如表 27.13 所示。

表 27.13 软件命令一览表

软件命令	第一个命令		第二个命令	
	地址	数据	地址	数据
读阵列模式的转移	FFFFFF800h	00FFh	—	—
读状态寄存器模式的转移（注 1）	FFFFFF800h	0070h	—	—
清除状态寄存器	FFFFFF800h	0050h	—	—
编程（注 2）	FFFFFF800h	0043h	WA	WD
块擦除	FFFFFF800h	0020h	BA	00D0h
锁定位编程	FFFFFF800h	0077h	BA	00D0h
读锁定位状态	FFFFFF800h	0071h	BA	00D0h
读锁定位状态模式的转移（注 3）	FFFFFF800h	0071h	—	—
保护位编程	FFFFFF800h	0067h	PBA	00D0h
读保护位状态模式的转移（注 3）	FFFFFF800h	0061h	—	—

WA: 编程地址（偶数）

WD: 编程数据（16 位）

BA: 对象块内的任意偶数地址

PBA: 保护位的地址（参照表 27.4）

注 1. 不能在 EW1 模式中使用。

注 2. 以 64 位（4 个字）为单位进行编程。第二个命令到第五个命令为止是一串命令。地址（WA）的高 29 位固定，低 3 位必须从第二个命令开始，按顺序指定 000b-010b-100b-110b（0h-2h-4h-6h 或者 8h-Ah-Ch-Eh）。

注 3. 必须从 RAM 中的程序发行。

### 27.3.4 模式的转移

在 CPU 改写模式中，闪存的运行模式有以下 4 种：

- 读阵列模式
- 读状态寄存器模式
- 读锁定位状态模式
- 读保护位状态模式

如果在这些模式中读闪存，就能分别读到存储器的内容、状态寄存器的内容、所读块的锁定位的状态以及保护位的状态。这些内容分别如表 27.14 ~ 表 27.16 所示。

表 27.14 状态寄存器的内容

位	位符号	位名	内容	
			“0”	“1”
b15 ~ b8	—	无效位	—	—
b7	SR7	顺序状态	忙	就绪
b6	—	保留位	—	—
b5	SR5	擦除状态	正常结束	异常结束
b4	SR4	编程状态	正常结束	异常结束
b3	—	保留位	—	—
b2	—	保留位	—	—
b1	—	保留位	—	—
b0	—	保留位	—	—

表 27.15 锁定位状态的内容

位	位符号	位名	内容	
			“0”	“1”
b15 ~ b7	—	无效位	—	—
b6	LBS	锁定位状态	锁定	非锁定
b5 ~ b0	—	无效位	—	—

表 27.16 保护位状态的内容

位	位符号	位名	内容	
			“0”	“1”
b15 ~ b7	—	无效位	—	—
b6	PBS	保护位状态	保护	非保护
b5 ~ b0	—	无效位	—	—

能通过从这些模式发行软件命令，对闪存进行编程和擦除。在编程和擦除结束后，自动返回到读阵列模式（EW1 模式的情况）或者读状态寄存器模式（EW0 模式的情况）。

### 27.3.5 命令的发行步骤

在此说明软件命令的发行步骤。必须在 FMSR0 寄存器的 RDY 位为 “1”（就绪）时发行命令。

#### 27.3.5.1 读阵列模式的转移命令

这是向读阵列模式转移的命令。

如果将 “00FFh” 写到地址 FFFFF800h，闪存就进入读阵列模式，并且能读闪存内任意地址的内容。

如果进入 EW1 模式，闪存总是为读阵列模式。

#### 27.3.5.2 读状态寄存器模式的转移命令

这是向读状态寄存器模式转移的命令。

如果将 “0070h” 写到地址 FFFFF800h，无论读闪存的哪个地址，都能读状态寄存器的内容。

在 EW1 模式中，不能发行此命令。



### 27.3.5.3 清除状态寄存器命令

这是将闪存内部的状态寄存器进行复位的命令。

如果将“0050h”写到地址 FFFF800h，状态寄存器的 SR5 位和 SR4 位就变为“0”（正常结束）（参照表 27.14），而且 FMSR0 寄存器的 EERR 位和 WERR 位也变为“0”。

### 27.3.5.4 编程命令

这是以 4 个字（8 字节）为单位将数据写到闪存的命令。

如果先将“0043h”写到地址 FFFF800h，然后将数据写到地址  $8n+0$  ~ 地址  $8n+6$ ，就开始自动编程（数据的编程和验证）。必须在发行最后的命令前，确认 FMR0 寄存器的 FCA 位是否为“0”。

能通过 FMSR0 寄存器的 RDY 位确认自动编程的结束。RDY 位在自动编程时为“0”（忙），在编程结束后为“1”（就绪）。

能通过 FMSR0 寄存器的 WERR 位确认自动编程的结果（参照“27.3.6 状态检查”）。

另外，不能对已编程的地址进行追加写。

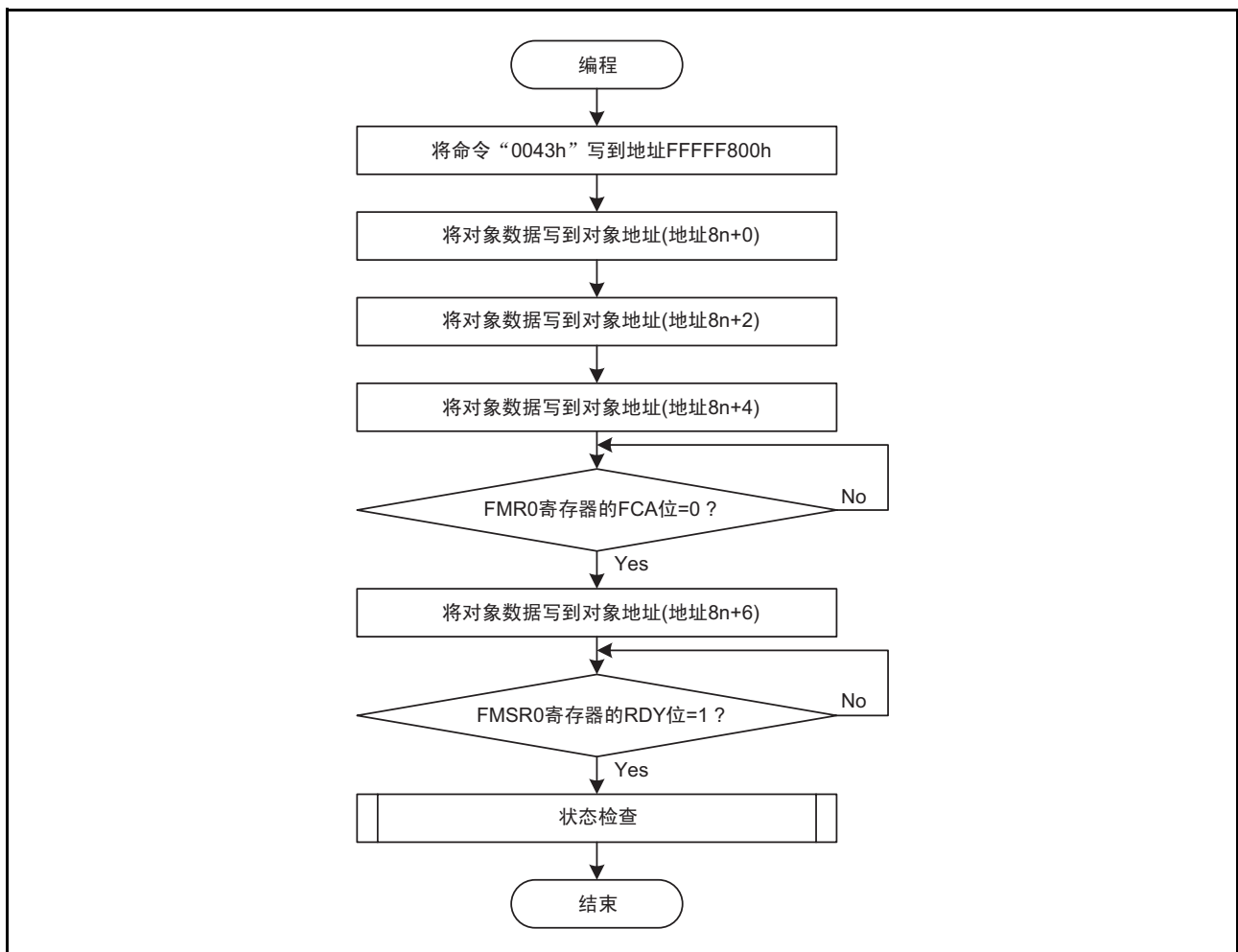


图 27.15 编程命令的发行流程图

### 27.3.5.5 块擦除命令

这是擦除闪存指定块的命令。

在将“0020h”写到地址 FFFFF800h 并且确认 FMR0 寄存器的 FCA 位为“0”后，如果将“00D0h”写到对象块的任意偶数地址，就开始自动擦除（擦除和验证）指定的块。

能通过 FMSR0 寄存器的 RDY 位确认自动擦除的结束。RDY 位在自动擦除时为“0”（忙），在擦除结束后为“1”（就绪）。

能通过 FMSR0 寄存器的 EERR 位确认自动擦除的结果（参照“27.3.6 状态检查”）。

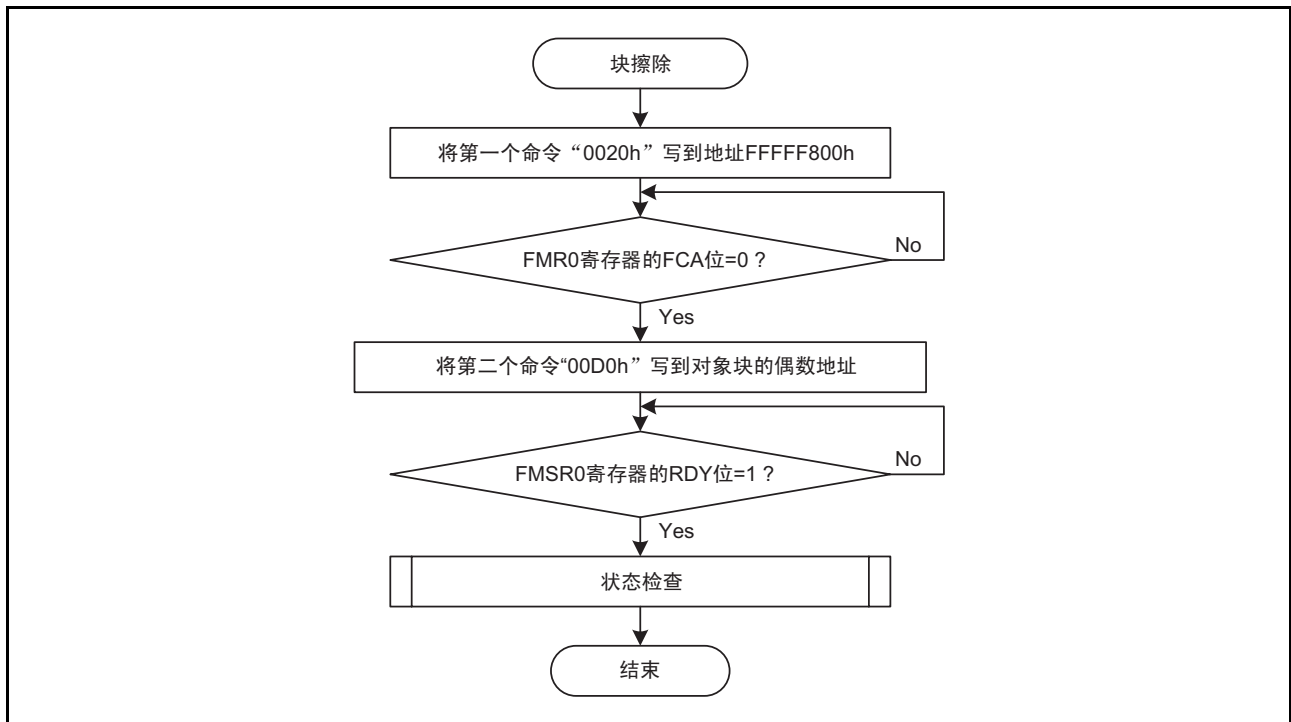


图 27.16 块擦除命令的发行流程图

### 27.3.5.6 锁定位编程命令

这是锁定闪存指定块的命令。

在将“0077h”写到地址 FFFFF800h 并确认 FMR0 寄存器的 FCA 位为“0”后，如果将“00D0h”写到对象块的任意偶数地址，就给指定块的锁定位写“0”（锁定）。

能通过 FMSR0 寄存器的 RDY 位确认锁定位编程的结束。RDY 位在进行锁定位编程时为“0”（忙），在编程结束后为“1”（就绪）。

当 FMR0 寄存器的 LBM 位为“1”（通过 LBS 位读）时，能通过读锁定位状态的命令确认锁定位的状态（参照“27.3.5.7 读锁定位状态命令”）；当 LBM 位为“0”（通过数据总线读）时，必须转移到读锁定位状态模式（参照“27.3.5.8 读锁定位状态模式的转移命令”）。

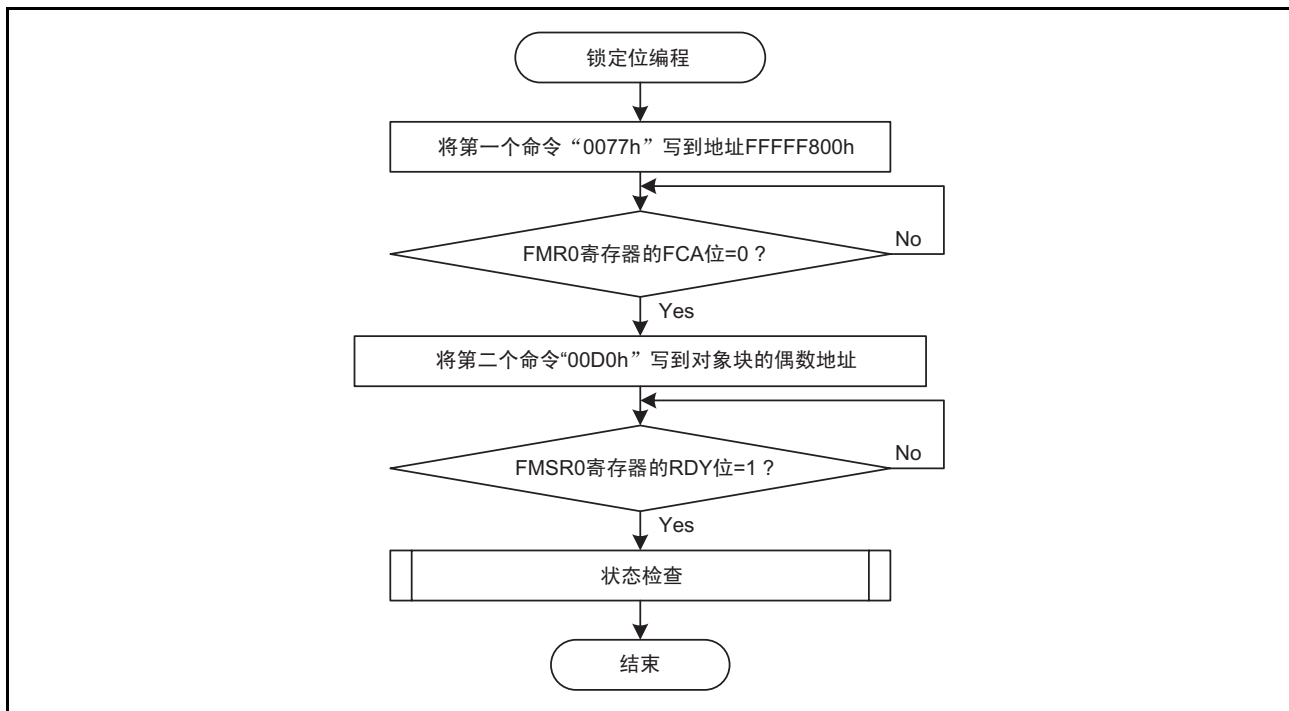


图 27.17 锁定位编程命令的发行流程图

### 27.3.5.7 读锁定位状态命令

这是确认闪存指定块的锁定状态的命令，能在 FMR0 寄存器的 LBM 位为“1”（通过 LBS 位读）时使用。

在将“0071h”写到地址 FFFF800h 地址并确认 FMR0 寄存器的 FCA 位为“0”后，如果将“00D0h”写到对象块的任意偶数地址，指定块的锁定位状态就被反映在 FMR0 寄存器的 LBS 位。

必须在 FMSR0 寄存器的 RDY 位变为“1”（就绪）后读 LBS 位。

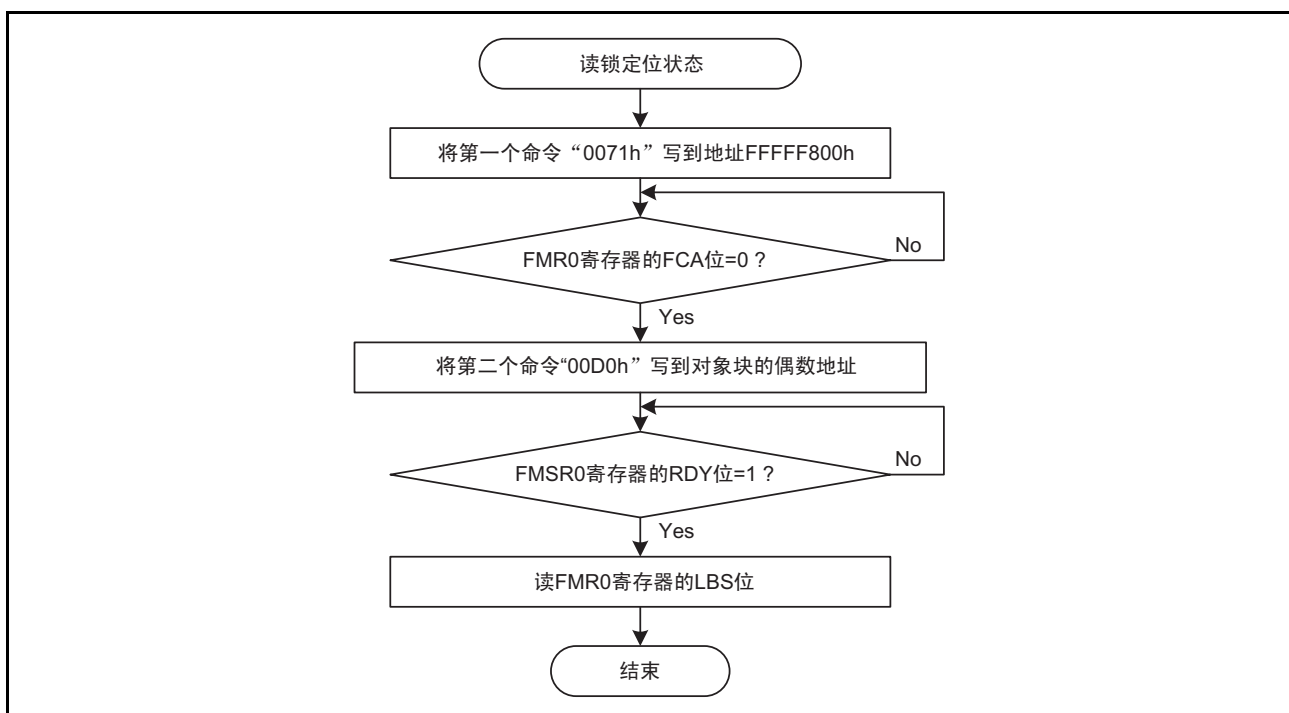


图 27.18 读锁定位状态命令的发行流程图

### 27.3.5.8 读锁定位状态模式的转移命令

这是向读锁定位状态模式转移的命令，能在 FMR0 寄存器的 LBM 位为“0”（通过数据总线读）时使用。如果将“0071h”写到地址 FFFF800h，无论读闪存的哪个地址，都能读块的锁定位状态（参照表 27.15）。必须在 RAM 中执行发行此命令的程序。

### 27.3.5.9 保护位编程命令

这是保护闪存指定块的命令。如果将任意块的保护位置“0”，ROM 码保护就有效。

在将“0067h”写到地址 FFFF800h 并确认 FMR0 寄存器的 FCA 位为“0”后，如果给对象块的保护位（参照表 27.4）写“00D0h”，就给指定块的保护位写“0”（保护）。

能通过 FMSR0 寄存器的 RDY 位确认保护位编程的结束。RDY 位在进行保护位编程时为“0”（忙），在编程结束后为“1”（就绪）。

能在转移到读保护位状态模式（参照“27.3.5.10 读保护位状态模式的转移命令”）后，通过读闪存确认保护位的状态。

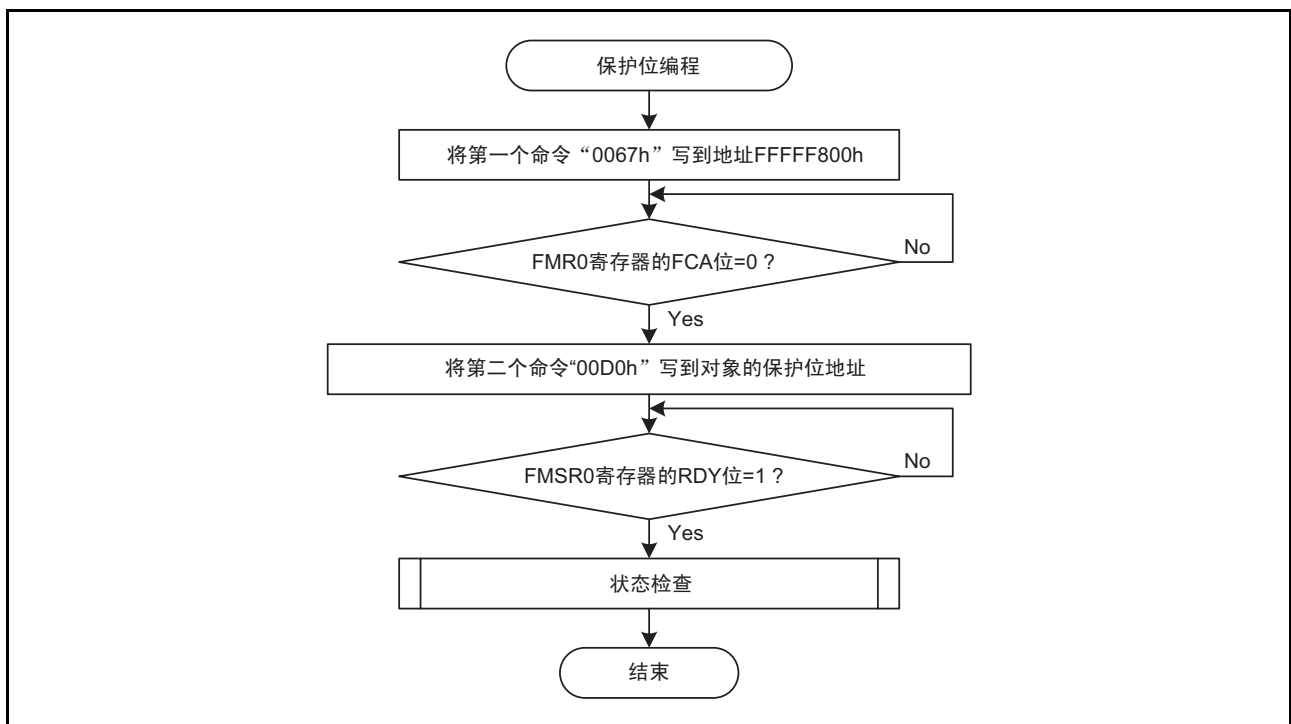


图 27.19 保护位编程命令的发行流程图

### 27.3.5.10 读保护位状态模式的转移命令

这是向读保护位状态模式转移的命令。

如果将“0061h”写到地址 FFFF800h，无论读闪存的哪个地址，都能读块的保护位状态（参照表 27.16）。必须在 RAM 中执行发行此命令的程序。

### 27.3.6 状态检查

能通过读 FMSR0 寄存器的 EERR 位和 WERR 位或者状态寄存器的 SR5 位和 SR4 位，确认是否正常地执行了软件命令。

这些位的含义和状态检查的流程图例子分别如表 27.17 和图 27.20 所示。

表 27.17 状态和错误内容

FMSR0 寄存器 (状态寄存器)		错误内容	发生错误的原因
EERR 位 (SR5 位)	WERR 位 (SR4 位)		
1	1	命令顺序错误	<ul style="list-style-type: none"> <li>对于有第二个命令的软件命令，给最后命令写了“00D0h”或者“00FFh”（命令取消）以外的值。</li> <li>给指定地址的命令指定了实际上不存在的地址。</li> </ul>
1	0	擦除错误	<ul style="list-style-type: none"> <li>试图擦除被锁定的块。</li> <li>无法正确地擦除对象块。</li> </ul>
0	1	编程错误	<ul style="list-style-type: none"> <li>试图对被锁定的块进行编程。</li> <li>无法正确地对数据进行编程。</li> <li>无法正确地对锁定位进行编程。</li> <li>无法正确地对保护位进行编程。</li> </ul>
0	0	正常结束	

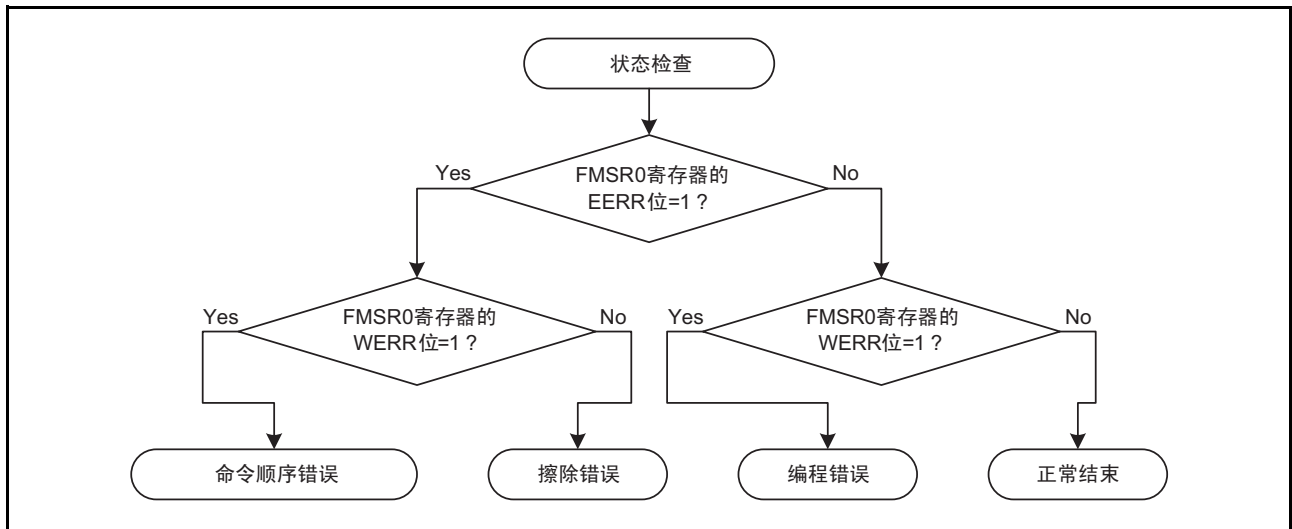


图 27.20 状态检查的流程图

在发生错误时，必须在发行清除状态寄存器命令后进行错误处理。

与编程是否正确无关，在频繁发生擦除错误或者编程错误时，有可能不能使用该块。

## 27.4 标准串行输入 / 输出模式

在标准串行输入 / 输出模式中，能使用支持 R32C/117 群的串行编程器，在将单片机安装到电路板的状态下改写闪存的内容。

有关串行编程器，请向各厂商询问；有关串行编程器的操作方法，请参照串行编程器的用户使用手册。

标准串行输入 / 输出模式有使用时钟同步串行接口的标准串行输入 / 输出模式 1 以及使用异步串行接口的标准串行输入 / 输出模式 2，各模式的特点如表 27.18 所示。

表 27.18 标准串行输入 / 输出模式的种类和特性

项目		标准串行输入 / 输出模式 1	标准串行输入 / 输出模式 2
通信方式		时钟同步串行通信	异步串行通信
通信速度		高速	低速
使用串行接口		UART1	UART1
引脚的设定	CNVSS	“H”	“H”
	$\overline{\text{CE}}$ (P5_0)	“H”	“H”
	$\overline{\text{EPM}}$ (P5_5)	“L”	“L”
	SCLK (P6_5)	在复位时，为“L”电平； 在通信时，为发送 / 接收时钟。	在复位时，为“L”电平； 在通信时，不使用。
引脚的功能	BUSY (P6_4)	BUSY 信号	监视程序运行的检查。
	RXD (P6_6)	串行数据输入	串行数据输入
	TXD (P6_7)	串行数据输出	串行数据输出

标准串行输入 / 输出模式的引脚功能如表 27.19 所示。使用标准串行输入 / 输出模式 1 时的引脚处理例子如图 27.21 所示，使用标准串行输入 / 输出模式 2 时的引脚处理例子如图 27.22 所示。控制引脚因串行编程器而不同，详细内容请参照串行编程器的用户使用手册。

表 27.19 标准串行输入 / 输出模式的引脚功能

引脚名	名称	输入 / 输出	功能
VCC、VSS	电源输入	输入	必须给 VCC 引脚输入编程 / 擦除的保证电压，给 VSS 引脚输入 0V。
VDC1、VDC0	去耦电容器的连接引脚	—	必须在 VDC1 引脚和 VDC0 引脚之间连接用于稳定内部逻辑电压的去耦电容器。
CNVSS	CNVSS	输入	必须上拉到 VCC。
RESET	复位输入	输入	这是复位输入引脚。必须在 RESET 引脚为“L”电平期间，给 XIN 引脚至少输入 20 个时钟周期。
XIN	主时钟输入	输入	必须在 XIN 和 XOUT 之间连接陶瓷谐振器或者晶体振荡器。在输入外部生成的时钟时，必须从 XIN 输入并将 XOUT 置为开路。
XOUT	主时钟输出	输出	
NSD	调试端口	输入 / 输出	必须通过 1k ~ 4.7kΩ 的电阻上拉到 VCC。
AVCC、AVSS	模拟电源输入	输入	必须将 AVCC 连接到 VCC，将 AVSS 连接到 VSS。
VREF	基准电压输入	输入	这是 A/D 转换器和 D/A 转换器的基准电压输入引脚。
P0_0 ~ P0_7、 P1_0 ~ P1_7、 P2_0 ~ P2_7、 P3_0 ~ P3_7、 P4_0 ~ P4_7	输入端口	输入	必须输入“H”电平或者“L”电平，或者将这些引脚置为开路。
P5_0	CE 输入	输入	必须输入“H”电平。
P5_1 ~ P5_4	输入端口	输入	必须输入“H”电平或者“L”电平，或者将这些引脚置为开路。
P5_5	EPM 输入	输入	必须输入“L”电平。
P5_6、P5_7、 P6_0 ~ P6_3	输入端口	输入	必须输入“H”电平或者“L”电平，或者将这些引脚置为开路。
P6_4	BUSY 输出	输出	在标准串行输入 / 输出模式 1 中，为 BUSY 信号的输出引脚； 在标准串行输入 / 输出模式 2 中，监视程序运行的检查。
P6_5	SCLK 输入	输入	在标准串行输入 / 输出模式 1 中，为串行时钟的输入引脚； 在标准串行输入 / 输出模式 2 中，必须输入“L”电平。
P6_6	数据输入 RXD	输入	这是串行数据的输入引脚。
P6_7	数据输出 TXD	输出	这是串行数据的输出引脚。
P7_0 ~ P7_7、 P8_0 ~ P8_4	输入端口	输入	必须输入“H”电平或者“L”电平，或者将这些引脚置为开路。
P8_5	NMI 输入	输入	必须上拉到 VCC。
P8_6、P8_7、 P9_0 ~ P9_7、 P10_0 ~ P10_7 (注 1)	输入端口	输入	必须输入“H”电平或者“L”电平，或者将这些引脚置为开路。
P11_0 ~ P11_4、 P12_0 ~ P12_7、 P13_0 ~ P13_7 (注 1)	输入端口	输入	必须输入“H”电平或者“L”电平，或者将这些引脚置为开路。
P14_1、 P14_3 ~ P14_6、 P15_0 ~ P15_7 (注 1)	输入端口	输入	必须输入“H”电平或者“L”电平，或者将这些引脚置为开路。

注 1. 端口 P9\_0、P9\_2、P11 ~ P15 只限于 144 引脚版。

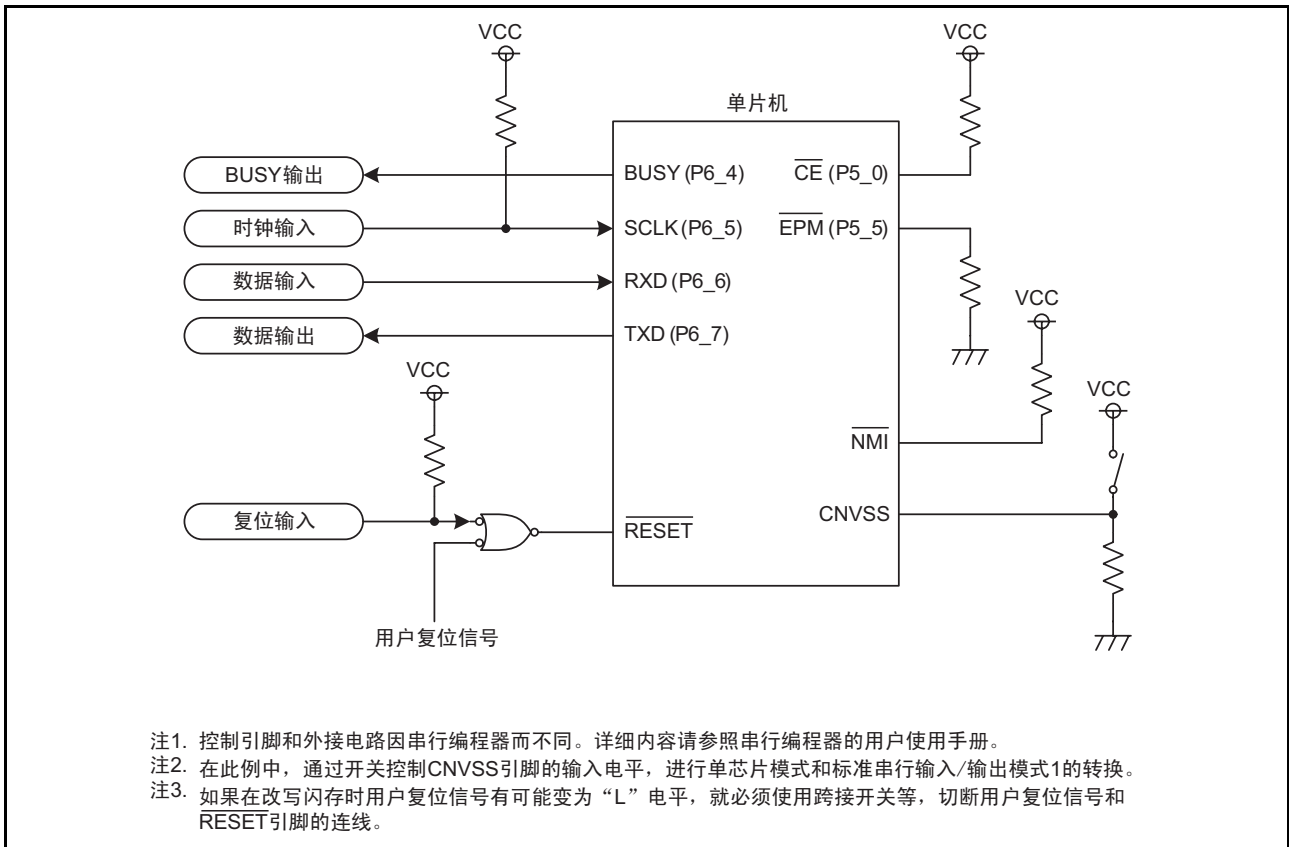


图 27.21 标准串行输入 / 输出模式 1 的引脚处理例子

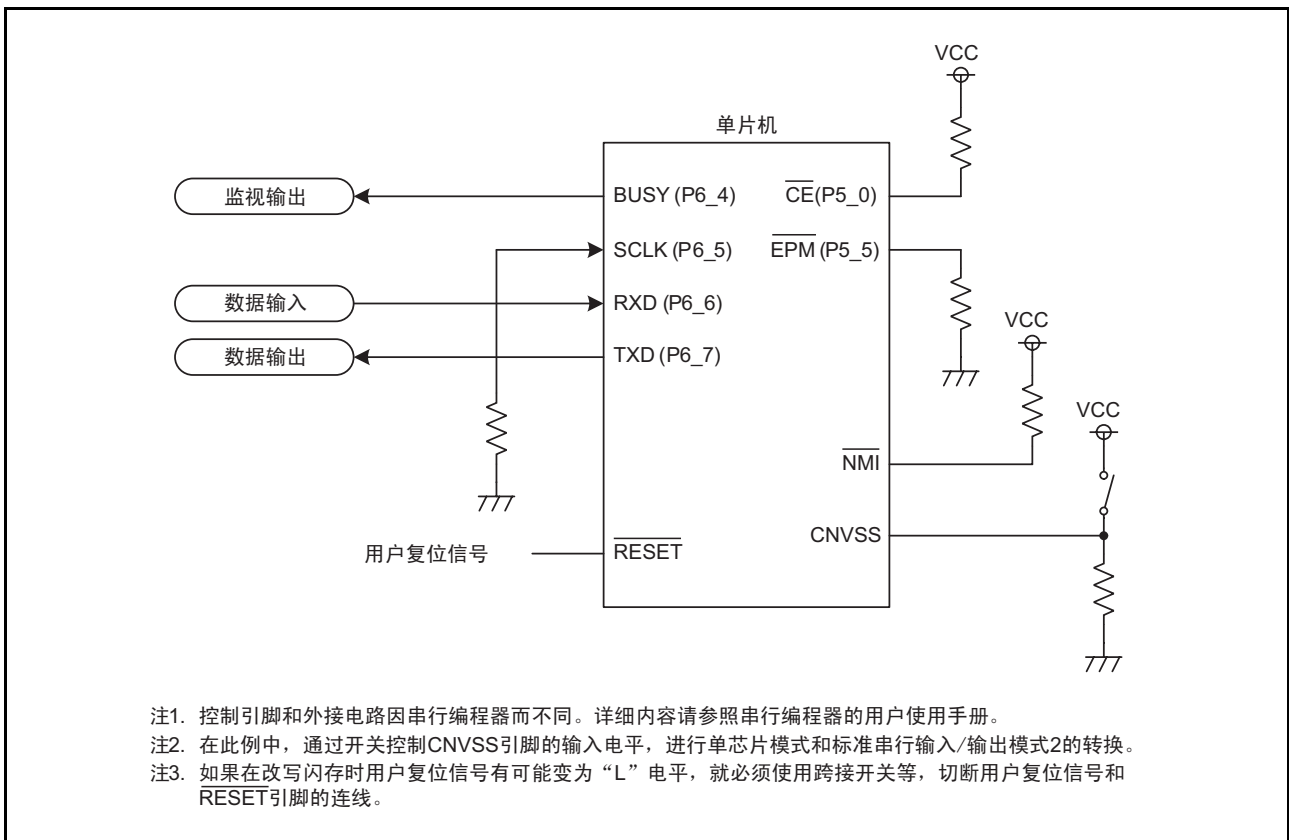


图 27.22 标准串行输入 / 输出模式 2 的引脚处理例子



## 27.5 并行输入 / 输出模式

在并行输入 / 输出模式中，能使用支持 R32C/117 的并行编程器改写闪存的内容。

有关并行编程器，请向各厂商询问；有关并行编程器的操作方法，请参照并行编程器的用户使用手册。

## 27.6 有关改写闪存的注意事项

### 27.6.1 有关电源电压的注意事项

- 改写闪存时的电源电压必须为电特性规定的电压范围内的电压。如果在改写闪存时电压的波动超过保证值，就无法保证闪存。

### 27.6.2 有关硬件复位的注意事项

- 在改写闪存时，不能进行硬件复位。

### 27.6.3 有关闪存保护的注意事项

- 如果将错误的的数据写到 ID 码的保存地址，就无法在标准串行输入 / 输出模式中读写闪存。

### 27.6.4 编程时的注意事项

- 在低速模式或者低功耗模式中，不能将 FMCR 寄存器的 FEW 位置 “1”（CPU 改写模式）。
- 通过 NMI、看门狗定时器中断、振荡停止检测中断或者低电压检测中断，将编程、块擦除、锁定位编程和保护位编程中断。如果这些软件命令被中断，就必须在擦除该块后再次执行相同的命令。尤其在块擦除被中断时，因为锁定位和保护位的值不确定，所以必须在解除锁定后再次实施块擦除。

### 27.6.5 使用中断时的注意事项

- EW0 模式
  - 能通过将可变量表中的向量写到 RAM 区，使用该向量的中断。
  - 如果发生 NMI、看门狗定时器中断、振荡停止检测中断或者低电压检测中断，就自动进入读阵列模式，因此也能在改写闪存时使用这些中断。在发生中断时，闪存的改写被中断，并且 FMR0 寄存器和 FMSR0 寄存器被复位。必须在中断处理结束后，先将 FMR1 寄存器的 LBD 位置 “1”（锁定位保护无效），然后再次执行改写程序。
  - 因为 BRK 指令、INTO 指令、UND 指令参照闪存中的数据，所以不能使用这些指令。
- EW1 模式
  - 在进行编程或者块擦除时，不能接受可变量表中有向量的中断。
  - 不能发生看门狗定时器中断。
  - 如果发生 NMI、振荡停止检测中断或者低电压检测中断，就自动进入读阵列模式，因此也能在改写闪存时使用这些中断。在发生中断时，闪存的改写被中断，并且 FMR0 寄存器和 FMSR0 寄存器被复位。必须在中断处理结束后，先将 FMR0 寄存器的 EWM 位置 “1”（EW1 模式），并且将 FMR1 寄存器的 LBD 位置 “1”（锁定位保护无效），然后再次执行改写程序。

### 27.6.6 有关改写控制程序的改写注意事项

- EW0 模式
  - 如果在对保存改写控制程序的块进行改写时电源电压降低，就无法正常地对改写控制程序进行改写，因此以后有可能无法改写闪存。如果无法改写，就必须使用串行编程器或者并行编程器进行改写。
- EW1 模式
  - 不能对保存改写控制程序的块进行改写。

### 27.6.7 编程 / 擦除次数和软件命令的执行时间

- 软件命令（编程、块擦除、锁定位编程、保护位编程）的执行时间随着编程/擦除次数的增加而变长。如果编程/擦除次数超过电特性规定的编程/擦除次数，软件命令的执行时间就会明显变长，因此软件命令的等待时间必须至少设定为电特性规定的最大时间。

### 27.6.8 其他注意事项

- 电特性规定的编程/擦除次数是能保证原始性能的编程/擦除次数的最大值。如果超过这个次数，并非立即就无法进行编程和擦除。
- 在批量生产时，不能使用调试中重复编程、擦除过的芯片。

## 28. 电特性

表 28.1 绝对最大额定值（注 1）

符号	项目		条件	额定值（注 2）	单位
$V_{CC}$	电源电压		$V_{CC}=AV_{CC}$	-0.3 ~ 6.0	V
$AV_{CC}$	模拟电源电压		$V_{CC}=AV_{CC}$	-0.3 ~ 6.0	V
$V_I$	输入电压	XIN、 $\overline{RESET}$ 、 $\overline{CNVSS}$ 、NSD、 $V_{REF}$ 、 P0_0 ~ P0_7、P1_0 ~ P1_7、 P2_0 ~ P2_7、P3_0 ~ P3_7、 P5_0 ~ P5_3、P8_4 ~ P8_7、 P9_0 ~ P9_7、P10_0 ~ P10_7 P11_0 ~ P11_4、P12_0 ~ P12_7 P13_0 ~ P13_7、P14_1 P14_3 ~ P14_6、P15_0 ~ P15_7（注 2）		-0.3 ~ $V_{CC}+0.3$	V
		P4_0 ~ P4_7、P5_4 ~ P5_7、 P6_0 ~ P6_7、P7_0 ~ P7_7、 P8_0 ~ P8_3		-0.3 ~ 6.0	V
$V_O$	输出电压	XOUT、P0_0 ~ P0_7、P1_0 ~ P1_7、 P2_0 ~ P2_7、P3_0 ~ P3_7、 P4_0 ~ P4_7、P5_0 ~ P5_7、 P6_0 ~ P6_7、P7_0 ~ P7_7、 P8_0 ~ P8_4、P8_6、P8_7、 P9_0 ~ P9_7、P10_0 ~ P10_7、 P11_0 ~ P11_4、P12_0 ~ P12_7、 P13_0 ~ P13_7、P14_3 ~ P14_6、 P15_0 ~ P15_7（注 3）		-0.3 ~ $V_{CC}+0.3$	V
$P_d$	功耗		$T_a=25^\circ\text{C}$	500	mW
—	工作环境温度			-40 ~ 85	$^\circ\text{C}$
$T_{stg}$	保存温度			-65 ~ 150	$^\circ\text{C}$

注 1. 绝对最大额定值是即使在瞬间也不能超过的极限值。如果超过此值，就有可能明显降低或者破坏器件的可靠性。另外，在长期置于绝对最大额定值的条件下，也有可能影响器件的可靠性。

注 2. 端口 P9\_0、P9\_2、P11 ~ P15 只限于 144 引脚版。端口 P9\_1 在 100 引脚版中为输入专用引脚。

表 28.2 推荐的工作条件 (1) (注 1)

符号	项目	规格值 (注 2)			单位	
		最小	典型	最大		
$V_{CC}$	电源电压	3.0	5.0	5.5	V	
$AV_{CC}$	模拟电源电压		$V_{CC}$		V	
$V_{REF}$	基准电压	3.0		$V_{CC}$	V	
$V_{SS}$	电源电压		0		V	
$AV_{SS}$	模拟电源电压		0		V	
$dV_{CC}/dt$	$V_{CC}$ 电源上升斜率 ( $V_{CC} < 2.0V$ )	0.05			V/ms	
$V_{IH}$	“H”电平输入电压	XIN、 $\overline{RESET}$ 、CNVSS、NSD、 P2_0 ~ P2_7、P3_0 ~ P3_7、 P5_0 ~ P5_3、P8_4 ~ P8_7 (注 2)、 P9_0 ~ P9_7、P10_0 ~ P10_7、 P11_0 ~ P11_4、P14_1、 P14_3 ~ P14_6、P15_0 ~ P15_7 (注 3)、	$0.8 \times V_{CC}$		$V_{CC}$	V
		P4_0 ~ P4_7、P5_4 ~ P5_7、P6_0 ~ P6_7、 P7_0 ~ P7_7、P8_0 ~ P8_3	$0.8 \times V_{CC}$		6.0	V
		P0_0 ~ P0_7、P1_0 ~ P1_7、 P12_0 ~ P12_7、P13_0 ~ P13_7 (单芯片模式中) (注 3)	$0.8 \times V_{CC}$		$V_{CC}$	V
		P0_0 ~ P0_7、P1_0 ~ P1_7、 P12_0 ~ P12_7、P13_0 ~ P13_7、 (存储器扩展模式和微处理器模式中) (注 3)	$0.5 \times V_{CC}$		$V_{CC}$	V
$V_{IL}$	“L”电平输入电压	XIN、 $\overline{RESET}$ 、CNVSS、NSD、 P2_0 ~ P2_7、P3_0 ~ P3_7、P4_0 ~ P4_7、 P5_0 ~ P5_7、P6_0 ~ P6_7、 P7_0 ~ P7_7、P8_0 ~ P8_7 (注 2)、 P9_0 ~ P9_7、P10_0 ~ P10_7、 P11_0 ~ P11_4、P14_1、 P14_3 ~ P14_6、P15_0 ~ P15_7 (注 3)	0		$0.2 \times V_{CC}$	V
		P0_0 ~ P0_7、P1_0 ~ P1_7、 P12_0 ~ P12_7、P13_0 ~ P13_7 (单芯片模式中) (注 3)	0		$0.2 \times V_{CC}$	V
		P0_0 ~ P0_7、P1_0 ~ P1_7、 P12_0 ~ P12_7、P13_0 ~ P13_7 (存储器扩展模式和微处理器模式中) (注 3)	0		$0.16 \times V_{CC}$	V
$T_{opr}$	工作环境温度	N 版	-20		85	°C
		D 版	-40		85	°C
		P 版	-40		85	°C

注 1. 推荐的工作条件是保证器件工作的范围。如果超出此范围,即使在最大额定值内也无法保证工作。

注 2. P8\_7 的  $V_{IH}$  和  $V_{IL}$  是将 P8\_7 用作可编程输入端口时的规格,而不是用作 XCIN 时的规格。

注 3. 端口 P9\_0、P9\_2、P11 ~ P15 只限于 144 引脚版。端口 P9\_1 在 100 引脚版中为输入专用引脚。

表 28.3 推荐的工作条件 (2) (在没有指定时,  $V_{CC}=3.0 \sim 5.5V$ ,  $V_{SS}=0V$ ,  $T_a=T_{opr}$ ) (注 1)

符号	项目	规格值 (注 2)			单位
		最小	典型	最大	
$C_{VDC}$	电压稳压器的去耦电容器电容 引脚之间的电压 1.5V	2.4		10.0	$\mu F$

注 1. 推荐的工作条件是保证器件工作的范围。如果超出此范围, 即使在最大额定值内也无法保证工作。

注 2. 必须在考虑电容器的使用温度、两个引脚之间的直流电压和老化等这些条件之后满足规格值。

表 28.4 推荐的工作条件 (3) (在没有指定时,  $V_{CC}=3.0 \sim 5.5V$ ,  $V_{SS}=0V$ ,  $T_a=T_{opr}$ ) (注 1)

符号	项目	规格值			单位	
		最小	典型	最大		
$I_{OH(peak)}$	“H”电平峰值输出电流 (注 2)	P0_0 ~ P0_7, P1_0 ~ P1_7, P2_0 ~ P2_7, P3_0 ~ P3_7, P4_0 ~ P4_7, P5_0 ~ P5_7, P6_0 ~ P6_7, P7_0 ~ P7_7, P8_0 ~ P8_4, P8_6, P8_7, P9_0 ~ P9_7, P10_0 ~ P10_7, P11_0 ~ P11_4, P12_0 ~ P12_7, P13_0 ~ P13_7, P14_3 ~ P14_6, P15_0 ~ P15_7 (注 3)			-10.0	mA
$I_{OH(avg)}$	“H”电平平均输出电流 (注 4)	P0_0 ~ P0_7, P1_0 ~ P1_7, P2_0 ~ P2_7, P3_0 ~ P3_7, P4_0 ~ P4_7, P5_0 ~ P5_7, P6_0 ~ P6_7, P7_0 ~ P7_7, P8_0 ~ P8_4, P8_6, P8_7, P9_0 ~ P9_7, P10_0 ~ P10_7, P11_0 ~ P11_4, P12_0 ~ P12_7, P13_0 ~ P13_7, P14_3 ~ P14_6, P15_0 ~ P15_7 (注 3)			-5.0	mA
$I_{OL(peak)}$	“L”电平峰值输出电流 (注 2)	P0_0 ~ P0_7, P1_0 ~ P1_7, P2_0 ~ P2_7, P3_0 ~ P3_7, P4_0 ~ P4_7, P5_0 ~ P5_7, P6_0 ~ P6_7, P7_0 ~ P7_7, P8_0 ~ P8_4, P8_6, P8_7, P9_0 ~ P9_7, P10_0 ~ P10_7, P11_0 ~ P11_4, P12_0 ~ P12_7, P13_0 ~ P13_7, P14_3 ~ P14_6, P15_0 ~ P15_7 (注 3)			10.0	mA
$I_{OL(avg)}$	“L”电平平均输出电流 (注 4)	P0_0 ~ P0_7, P1_0 ~ P1_7, P2_0 ~ P2_7, P3_0 ~ P3_7, P4_0 ~ P4_7, P5_0 ~ P5_7, P6_0 ~ P6_7, P7_0 ~ P7_7, P8_0 ~ P8_4, P8_6, P8_7, P9_0 ~ P9_7, P10_0 ~ P10_7, P11_0 ~ P11_4, P12_0 ~ P12_7, P13_0 ~ P13_7, P14_3 ~ P14_6, P15_0 ~ P15_7 (注 3)			5.0	mA

注 1. 推荐的工作条件是保证器件工作的范围。如果超出此范围, 即使在最大额定值内也无法保证工作。

注 2. 端口 P0、P1、P2、P8\_6、P8\_7、P9、P10、P11、P14、P15 的  $I_{OL(peak)}$  总电流值不能超过 80mA, 端口 P3、P4、P5、P6、P7、P8\_0 ~ P8\_4、P12、P13 的  $I_{OL(peak)}$  总电流值不能超过 80mA, 端口 P0、P1、P2、P11 的  $I_{OH(peak)}$  总电流值不能超过 -40mA, 端口 P8\_6、P8\_7、P9、P10、P14、P15 的  $I_{OH(peak)}$  总电流值不能超过 -40mA, 端口 P3、P4、P5、P12、P13 的  $I_{OH(peak)}$  总电流值不超过 -40mA, 端口 P6、P7、P8\_0 ~ P8\_4 的  $I_{OH(peak)}$  总电流值不能超过 -40mA。

注 3. 端口 P9\_0、P9\_2、P11 ~ P15 只限于 144 引脚版。端口 P9\_1 在 100 引脚版中为输入专用引脚。

注 4. 平均输出电流是 100ms 期间的平均值。

表 28.5 推荐的工作条件 (4) (在没有指定时,  $V_{CC}=3.0 \sim 5.5V$ ,  $V_{SS}=0V$ ,  $T_a=T_{opr}$ ) (注 1)

符号	项目	规格值			单位
		最小	典型	最大	
$f_{(XIN)}$	主时钟输入的振荡频率	4		16	MHz
$f_{(XRef)}$	基准时钟频率	2		4	MHz
$f_{(PLL)}$	PLL 时钟振荡频率	96		128	MHz
$f_{(Base)}$	基本时钟频率			50	MHz
$t_{c(Base)}$	基本时钟的周期时间	20			ns
$f_{(CPU)}$	CPU 工作频率			50	MHz
$t_{c(CPU)}$	CPU 时钟的周期时间	20			ns
$f_{(BCLK)}$	外围总线时钟频率			25	MHz
$t_{c(BCLK)}$	外围总线时钟的周期时间	40			ns
$f_{(PER)}$	外围功能时钟源频率			32	MHz
$f_{(XCIN)}$	副时钟振荡频率		32.768	62.5	kHz

注 1. 推荐的工作条件是保证器件工作的范围。如果超出此范围, 即使在最大额定值内也无法保证工作。

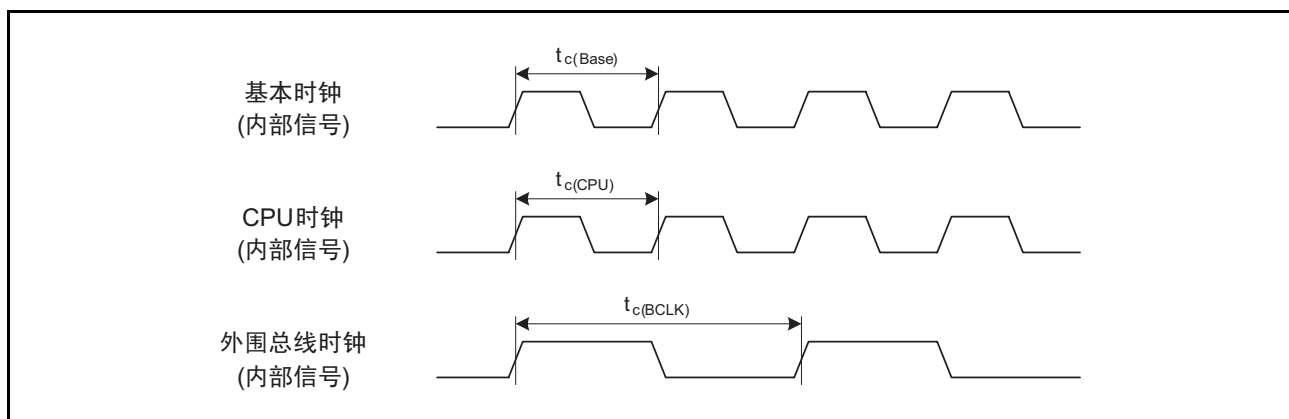


图 28.1 时钟的周期时间

表 28.6 推荐的工作条件 (5) (在没有指定时,  $V_{CC}=3.0 \sim 5.5V$ ,  $V_{SS}=0V$ ,  $T_a=T_{opr}$ ) (注 1)

符号	项目	规格值			单位
		最小	典型	最大	
$V_{r(VCC)}$	电源纹波的容许电压	$V_{CC}=5.0V$		0.5	Vp-p
		$V_{CC}=3.0V$		0.3	Vp-p
$dV_{r(VCC)}/dt$	电源纹波的上升 / 下降斜率	$V_{CC}=5.0V$		$\pm 0.3$	V/ms
		$V_{CC}=3.0V$		$\pm 0.3$	V/ms
$f_{r(VCC)}$	电源纹波的容许频率			10	kHz

注 1. 推荐的工作条件是保证器件工作的范围。如果超出此范围, 即使在最大额定值内也无法保证工作。

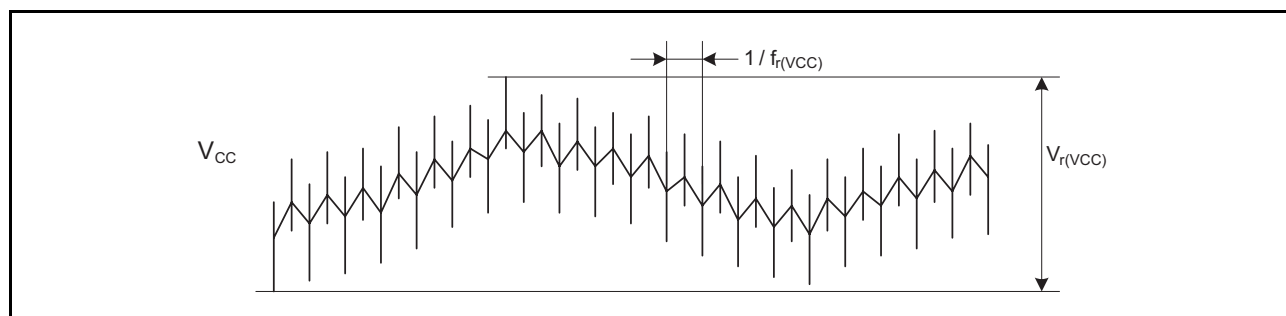


图 28.2 电源纹波的波形

表 28.7 RAM 的电特性（在没有指定时， $V_{CC}=3.0 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

符号	项目	测量条件	规格值			单位
			最小	典型	最大	
$V_{RDR}$	RAM 数据的保持电压	停止模式中	2.0			V

表 28.8 闪存电特性（在没有指定时， $V_{CC}=3.0 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

符号	项目		规格值			单位
			最小	典型	最大	
—	编程 / 擦除次数（注 1）	程序区	1000			次
		数据区	10000			次
—	4 个字的编程时间	程序区		150	900	$\mu s$
		数据区		300	1700	$\mu s$
—	锁定位的编程时间	程序区		70	500	$\mu s$
		数据区		140	1000	$\mu s$
—	块擦除时间	4K 字节块		0.12	3.0	s
		32K 字节块		0.17	3.0	s
		64K 字节块		0.20	3.0	s
—	数据保持时间（注 2）	$T_a=55^\circ C$ （注 3）	10			年

注 1. 编程 / 擦除次数的定义：

编程 / 擦除次数是指每块的擦除次数。如果编程 / 擦除次数为 n 次，每块就能擦除 n 次。

例如，对于 4K 字节的块 A，如果在将 4 个字的数据分 512 次写到不同的地址后擦除此块，编程 / 擦除次数就计为 1 次。

但是，对于 1 次擦除，不能对相同地址进行多次编程（禁止重写）。

注 2. 规格值包括没有外加电源电压和没有提供时钟的时间。

注 3. 有关此条件以外的数据保持时间，请向本公司的营业窗口询问。



表 28.9 电源电路的时序特性（在没有指定时， $V_{CC}=3.0 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

符号	项目	测量条件	规格值			单位
			最小	典型	最大	
$t_{d(P-R)}$	接通电源时的内部电源稳定时间				2	ms

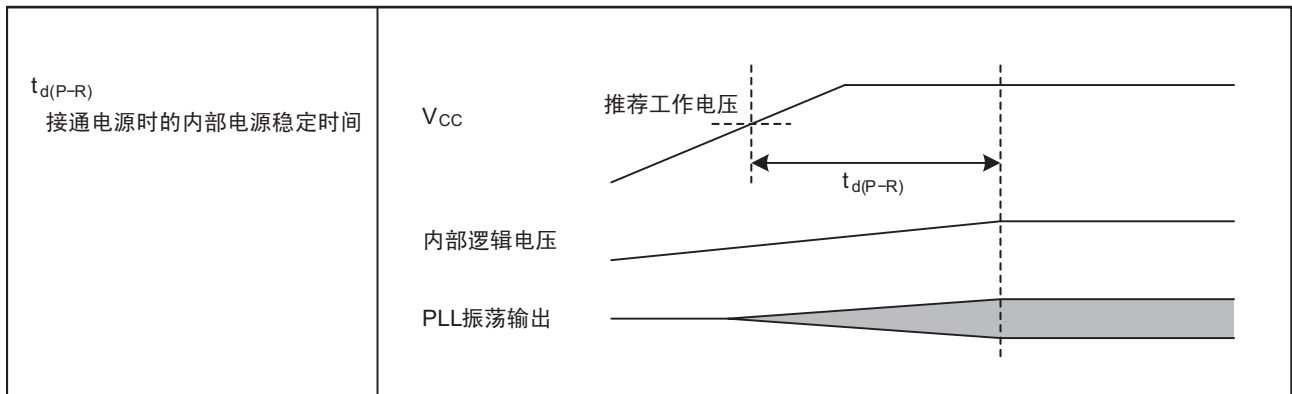


图 28.3 电源电路的时序

表 28.10 内部电压稳压器的电特性（在没有指定时， $V_{CC}=3.0 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

符号	项目	测量条件	规格值			单位
			最小	典型	最大	
$V_{VDC1}$	稳压器的输出电压			1.5		V

表 28.11 低电压检测电路的电特性（在没有指定时， $V_{CC}=4.2 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

符号	项目	测量条件	规格值			单位
			最小	典型	最大	
$\Delta V_{det}$	检测电压的误差				$\pm 0.3$	V
$V_{det(R)}-V_{det(F)}$	滞后宽度		0			V
—	自消耗电流	$V_{CC}=5.0V$ ，电压检测电路有效。		4		$\mu A$
$t_{d(E-A)}$	低电压检测电路的工作开始时间				150	$\mu s$

表 28.12 振荡电路的电特性（在没有指定时， $V_{CC}=3.0 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

符号	项目	测量条件	规格值			单位
			最小	典型	最大	
$f_{SO(PLL)}$	PLL 时钟的自振荡频率		35	50	65	MHz
$t_{LOCK(PLL)}$	PLL 振荡稳定时间（注 1）				1	ms
$t_{jitter(p-p)}$	PLL 振荡周期（p-p）				2.0	ns
$f_{(OCO)}$	内部振荡器的振荡频率		62.5	125	250	kHz

注 1. 条件是主时钟振荡稳定。

表 28.13 时钟电路的电特性（在没有指定时， $V_{CC}=3.0 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

符号	项目	测量条件	规格值			单位
			最小	典型	最大	
$t_{rec(WAIT)}$	等待模式 → 低功耗模式的恢复时间				225	$\mu s$
$t_{rec(STOP)}$	停止模式的恢复时间（注 1）				225	$\mu s$

注 1. 不包括主时钟的振荡稳定时间。在振荡稳定前，CPU 开始工作。

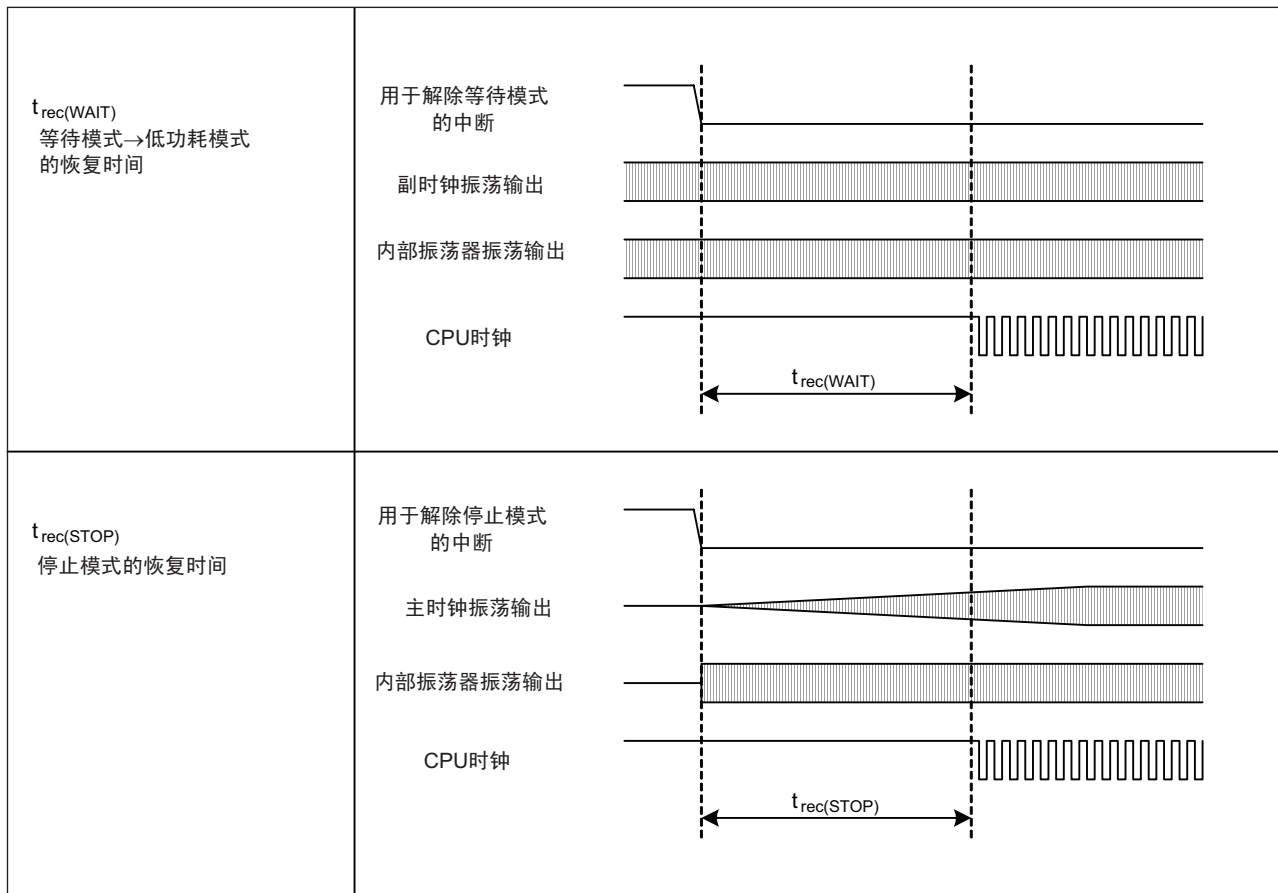


图 28.4 时钟电路的时序图

时序必要条件（在没有指定时， $V_{CC}=3.0 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.14 闪存的 CPU 改写模式时序

符号	项目	规格值		单位
		最小	最大	
$t_{cR}$	读周期时间	200		ns
$t_{su(S-R)}$	读操作前的片选准备时间	200		ns
$t_{h(R-S)}$	读操作后的片选保持时间	0		ns
$t_{su(A-R)}$	读操作前的地址准备时间	200		ns
$t_{h(R-A)}$	读操作后的地址保持时间	0		ns
$t_{w(R)}$	读脉宽	100		ns
$t_{cW}$	写周期时间	200		ns
$t_{su(S-W)}$	写操作前的片选准备时间	0		ns
$t_{h(W-S)}$	写操作后的片选保持时间	30		ns
$t_{su(A-W)}$	写操作前的地址准备时间	0		ns
$t_{h(W-A)}$	写操作后的地址保持时间	30		ns
$t_{w(W)}$	写脉宽	50		ns

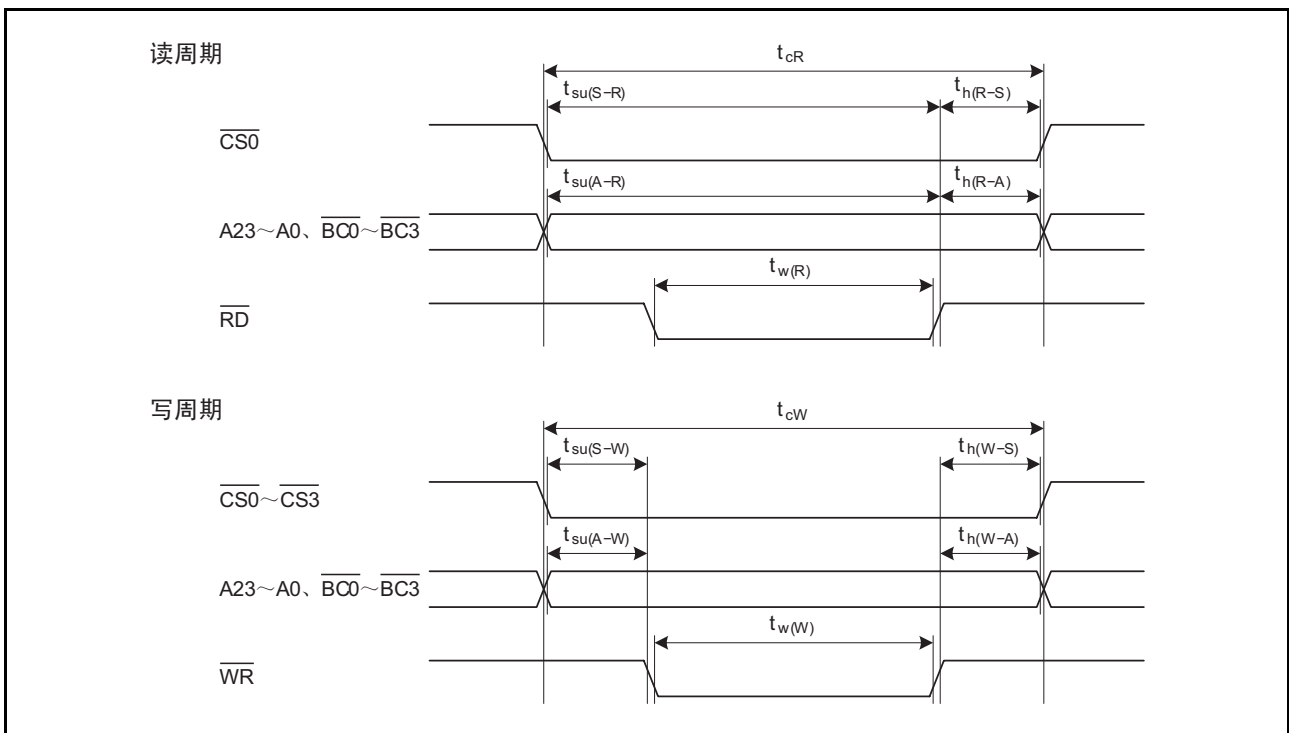


图 28.5 闪存的 CPU 改写模式时序

$V_{CC}=5V$ 表 28.15 电特性 (1) (在没有指定时,  $V_{CC}=4.2 \sim 5.5V$ ,  $V_{SS}=0V$ ,  $T_a=T_{opr}$ ,  $f_{(CPU)}=50MHz$ )

符号	项目	测量条件	规格值			单位
			最小	典型	最大	
$V_{OH}$	“H”电平输出电压 P0_0 ~ P0_7、P1_0 ~ P1_7、P2_0 ~ P2_7、 P3_0 ~ P3_7、P4_0 ~ P4_7、P5_0 ~ P5_7、 P6_0 ~ P6_7、P7_0 ~ P7_7、P8_0 ~ P8_4、 P8_6、P8_7、P9_0 ~ P9_7、 P10_0 ~ P10_7、P11_0 ~ P11_4、 P12_0 ~ P12_7、P13_0 ~ P13_7、 P14_3 ~ P14_6、P15_0 ~ P15_7 (注1)	$I_{OH}=-5mA$	$V_{CC}-2.0$		$V_{CC}$	V
		$I_{OH}=-200\mu A$	$V_{CC}-0.3$		$V_{CC}$	V
$V_{OL}$	“L”电平输出电压 P0_0 ~ P0_7、P1_0 ~ P1_7、P2_0 ~ P2_7、 P3_0 ~ P3_7、P4_0 ~ P4_7、P5_0 ~ P5_7、 P6_0 ~ P6_7、P7_0 ~ P7_7、P8_0 ~ P8_4、 P8_6、P8_7、P9_0 ~ P9_7、 P10_0 ~ P10_7、P11_0 ~ P11_4、 P12_0 ~ P12_7、P13_0 ~ P13_7、 P14_3 ~ P14_6、P15_0 ~ P15_7 (注1)	$I_{OL}=5mA$			2.0	V
		$I_{OL}=200\mu A$			0.45	V

注1. 端口 P9\_0、P9\_2、P11 ~ P15 只限于 144 引脚版。端口 P9\_1 在 100 引脚版中为输入专用引脚。

$V_{CC}=5V$ 表 28.16 电特性 (2) (在没有指定时,  $V_{CC}=4.2 \sim 5.5V$ ,  $V_{SS}=0V$ ,  $T_a=T_{opr}$ ,  $f_{(CPU)}=50MHz$ )

符号	项目	测量条件	规格值			单位	
			最小	典型	最大		
$V_{T+}-V_{T-}$	滞后	HOLD、RDY、NMI、INT0 ~ INT8、K10 ~ K13、 TA0IN ~ TA4IN、TA0OUT ~ TA4OUT、 TB0IN ~ TB5IN、CTS0 ~ CTS8、 CLK0 ~ CLK8、RXD0 ~ RXD8、 SCL0 ~ SCL6、SDA0 ~ SDA6、SS0 ~ SS6、 SRXD0 ~ SRXD6、ADTRG、 IIO0_0 ~ IIO0_7、IIO1_0 ~ IIO1_7、 UD0A、UD0B、UD1A、UD1B、ISCLK2、 ISRXD2、IEIN、CAN0IN、CAN0WU (注1)		0.2		1.0	V
		RESET		0.2		1.8	V
$I_{IH}$	“H”电 平输入电 流	XIN、RESET、CNVSS、NSD、 P0_0 ~ P0_7、P1_0 ~ P1_7、P2_0 ~ P2_7、 P3_0 ~ P3_7、P4_0 ~ P4_7、P5_0 ~ P5_7、 P6_0 ~ P6_7、P7_0 ~ P7_7、P8_0 ~ P8_7、 P9_0 ~ P9_7、P10_0 ~ P10_7、 P11_0 ~ P11_4、P12_0 ~ P12_7、 P13_0 ~ P13_7、P14_1、 P14_3 ~ P14_6、P15_0 ~ P15_7 (注2)	$V_I=5V$			5.0	$\mu A$
$I_{IL}$	“L”电 平输入电 流	XIN、RESET、CNVSS、NSD、 P0_0 ~ P0_7、P1_0 ~ P1_7、P2_0 ~ P2_7、 P3_0 ~ P3_7、P4_0 ~ P4_7、P5_0 ~ P5_7、 P6_0 ~ P6_7、P7_0 ~ P7_7、P8_0 ~ P8_7、 P9_0 ~ P9_7、P10_0 ~ P10_7、 P11_0 ~ P11_4、P12_0 ~ P12_7、 P13_0 ~ P13_7、P14_1、 P14_3 ~ P14_6、P15_0 ~ P15_7 (注2)	$V_I=0V$			-5.0	$\mu A$
$R_{PULLUP}$	上拉电阻	P0_0 ~ P0_7、P1_0 ~ P1_7、P2_0 ~ P2_7、 P3_0 ~ P3_7、P5_0 ~ P5_3、P8_4、P8_6、 P8_7、P9_0 ~ P9_7、P10_0 ~ P10_7、 P11_0 ~ P11_4、P12_0 ~ P12_7、 P13_0 ~ P13_7、P14_1、 P14_3 ~ P14_6、P15_0 ~ P15_7 (注2)	$V_I=0V$	30	50	170	k $\Omega$
$R_{fXIN}$	反馈电阻	XIN			1.5		M $\Omega$
$R_{fXCIN}$	反馈电阻	XCIN			15		M $\Omega$

注1. INT6 ~ INT8 只限于 144 引脚版。

注2. 端口 P9\_0、P9\_2、P11 ~ P15 只限于 144 引脚版。端口 P9\_1 在 100 引脚版中为输入专用引脚。

$V_{CC}=5V$ 表 28.17 电特性 (3) (在没有指定时,  $V_{CC}=4.2 \sim 5.5V$ ,  $V_{SS}=0V$ ,  $T_a=T_{opr}$ )

符号	项目	测量条件	规格值			单位	
			最小	典型	最大		
$I_{CC}$	电源电流	在单芯片模式中, 输出引脚为开路, 其他引脚连接到 $V_{SS}$ 。	$f_{(CPU)}=50MHz$ 、 $f_{(BCLK)}=25MHz$ 、 $f_{(XIN)}=8MHz$ 、PLL 振荡、XCIN 停止、OCO 停止		35	50	mA
		XIN-XOUT 驱动能力: Low	$f_{(CPU)}=f_{SO(PLL)}/24MHz$ 、XIN 停止、PLL 自振荡、XCIN 停止、OCO 停止		12		mA
		XCIN-XCOUT 驱动能力: Low	$f_{(CPU)}=f_{(BCLK)}=f_{(XIN)}/256MHz$ 、 $f_{(XIN)}=8MHz$ 、PLL 停止、XCIN 停止、OCO 停止		1.2		mA
			$f_{(CPU)}=f_{(BCLK)}=32.768kHz$ 、XIN 停止、PLL 停止、XCIN 振荡、OCO 停止、主稳压器停止		220		$\mu A$
			$f_{(CPU)}=f_{(BCLK)}=f_{(OCO)}/4kHz$ 、XIN 停止、PLL 停止、XCIN 停止、OCO 振荡、主稳压器停止		230		$\mu A$
			$f_{(CPU)}=f_{(BCLK)}=f_{(XIN)}/256MHz$ 、 $f_{(XIN)}=8MHz$ 、PLL 停止、XCIN 停止、OCO 停止、等待模式、 $T_a=25^\circ C$		960	1600	$\mu A$
			$f_{(CPU)}=f_{(BCLK)}=32.768kHz$ 、XIN 停止、PLL 停止、XCIN 振荡、OCO 停止、主稳压器停止、等待模式、 $T_a=25^\circ C$		8	140	$\mu A$
			$f_{(CPU)}=f_{(BCLK)}=f_{(OCO)}/4kHz$ 、XIN 停止、PLL 停止、XCIN 停止、OCO 振荡、主稳压器停止、等待模式、 $T_a=25^\circ C$		10	150	$\mu A$
			时钟停止、主稳压器停止、 $T_a=25^\circ C$		5	70	$\mu A$

$V_{CC}=5V$ 表 28.18 A/D 转换特性（在没有指定时， $V_{CC}=AV_{CC}=V_{REF}=4.2 \sim 5.5V$ ， $V_{SS}=AV_{SS}=0V$ ， $T_a=T_{opr}$ ， $f_{(BCLK)}=25MHz$ ）

符号	项目	测量条件	规格值			单位	
			最小	典型	最大		
—	分辨率	$V_{REF}=V_{CC}$			10	Bits	
—	绝对误差	$V_{REF}=V_{CC}=5V$	AN_0 ~ AN_7、 AN0_0 ~ AN0_7、 AN2_0 ~ AN2_7、 AN15_0 ~ AN15_7、 ANEX0、ANEX1（注1）			$\pm 3$	LSB
			外部运算放大器连接模式			$\pm 7$	LSB
INL	积分非线性误差	$V_{REF}=V_{CC}=5V$	AN_0 ~ AN_7、 AN0_0 ~ AN0_7、 AN2_0 ~ AN2_7、 AN15_0 ~ AN15_7、 ANEX0、ANEX1（注1）			$\pm 3$	LSB
			外部运算放大器连接模式			$\pm 7$	LSB
DNL	微分非线性误差				$\pm 1$	LSB	
—	偏移误差				$\pm 3$	LSB	
—	增益误差				$\pm 3$	LSB	
$R_{LADDER}$	梯形电阻	$V_{REF}=V_{CC}$	4		20	k $\Omega$	
$t_{CONV}$	转换时间 (10bit)	$\phi_{AD}=16MHz$ ，有采样 & 保持。	2.06			$\mu s$	
		$\phi_{AD}=16MHz$ ，无采样 & 保持。	3.69			$\mu s$	
$t_{CONV}$	转换时间 (8bit)	$\phi_{AD}=16MHz$ ，有采样 & 保持。	1.75			$\mu s$	
		$\phi_{AD}=16MHz$ ，无采样 & 保持。	3.06			$\mu s$	
$t_{SAMP}$	采样时间	$\phi_{AD}=16MHz$	0.188			$\mu s$	
$V_{IA}$	模拟输入电压		0		$V_{REF}$	V	
$\phi_{AD}$	工作时钟频率	无采样 & 保持	0.25		16	MHz	
		有采样 & 保持	1		16	MHz	

注 1. AN15\_0 ~ AN15\_7 只限于 144 引脚版。

表 28.19 D/A 转换特性（在没有指定时， $V_{CC}=AV_{CC}=V_{REF}=4.2 \sim 5.5V$ ， $V_{SS}=AV_{SS}=0V$ ， $T_a=T_{opr}$ ）

符号	项目	测量条件	规格值			单位
			最小	典型	最大	
—	分辨率				8	Bits
—	绝对精度				1.0	%
$t_s$	设定时间				3	$\mu s$
$R_O$	输出电阻		4	10	20	k $\Omega$
$I_{VREF}$	基准电源输入电流	（注1）			1.5	mA

注 1. 这是使用 1 个 D/A 转换器并且未使用的 D/A 转换器的 DA<sub>i</sub> 寄存器（ $i=0,1$ ）的值为“00h”的情况，A/D 转换器的梯形电阻部分除外。在将 ADCON1 寄存器的 VCUT 位置“0”（未连接  $V_{REF}$ ）时，也有  $I_{VREF}$  流过。

$$V_{CC}=5V$$

时序必要条件（在没有指定时， $V_{CC}=4.2 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.20 外部时钟输入

符号	项目	规格值		单位
		最小	最大	
$t_{c(X)}$	外部时钟输入的周期时间	62.5	250	ns
$t_{w(XH)}$	外部时钟输入的“H”电平脉宽	25		ns
$t_{w(XL)}$	外部时钟输入的“L”电平脉宽	25		ns
$t_{r(X)}$	外部时钟输入的上升时间		5	ns
$t_{f(X)}$	外部时钟输入的下降时间		5	ns
$t_w/t_c$	外部时钟输入的占空比	40	60	%

表 28.21 外部总线时序

符号	项目	规格值		单位
		最小	最大	
$t_{su(D-R)}$	读操作前的数据准备时间	40		ns
$t_h(R-D)$	读操作后的数据保持时间	0		ns
$t_{dis(R-D)}$	读操作后的数据禁止时间		$0.5 \times t_{c(Base)} + 10$	ns

表 28.22 定时器 A 的输入（事件计数器模式的计数输入）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TA)}$	TAiIN 输入的周期时间	200		ns
$t_{w(TAH)}$	TAiIN 输入的“H”电平脉宽	80		ns
$t_{w(TAL)}$	TAiIN 输入的“L”电平脉宽	80		ns

表 28.23 定时器 A 的输入（定时器模式的选通输入）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TA)}$	TAiIN 输入的周期时间	400		ns
$t_{w(TAH)}$	TAiIN 输入的“H”电平脉宽	180		ns
$t_{w(TAL)}$	TAiIN 输入的“L”电平脉宽	180		ns

表 28.24 定时器 A 的输入（单触发定时器模式的外部触发输入）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TA)}$	TAiIN 输入的周期时间	200		ns
$t_{w(TAH)}$	TAiIN 输入的“H”电平脉宽	80		ns
$t_{w(TAL)}$	TAiIN 输入的“L”电平脉宽	80		ns



$$V_{CC}=5V$$

时序必要条件（在没有指定时， $V_{CC}=4.2 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.25 定时器 A 的输入（脉宽调制模式的外部触发输入）

符号	项目	规格值		单位
		最小	最大	
$t_{w(TAH)}$	TAiIN 输入的“H”电平脉宽	80		ns
$t_{w(TAL)}$	TAiIN 输入的“L”电平脉宽	80		ns

表 28.26 定时器 A 的输入（事件计数器模式的递增 / 递减输入）

符号	项目	规格值		单位
		最小	最大	
$t_{c(UP)}$	TAiOUT 输入的周期时间	2000		ns
$t_{w(UPH)}$	TAiOUT 输入的“H”电平脉宽	1000		ns
$t_{w(UPL)}$	TAiOUT 输入的“L”电平脉宽	1000		ns
$t_{su(UP-TIN)}$	TAiOUT 输入的准备时间	400		ns
$t_{h(TIN-UP)}$	TAiOUT 输入的保持时间	400		ns

表 28.27 定时器 B 的输入（事件计数器模式的计数输入）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TB)}$	TBiIN 输入的周期时间（单边沿计数）	200		ns
$t_{w(TBH)}$	TBiIN 输入的“H”电平脉宽（单边沿计数）	80		ns
$t_{w(TBL)}$	TBiIN 输入的“L”电平脉宽（单边沿计数）	80		ns
$t_{c(TB)}$	TBiIN 输入的周期时间（双边沿计数）	200		ns
$t_{w(TBH)}$	TBiIN 输入的“H”电平脉宽（双边沿计数）	80		ns
$t_{w(TBL)}$	TBiIN 输入的“L”电平脉宽（双边沿计数）	80		ns

表 28.28 定时器 B 的输入（脉冲周期测量模式）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TB)}$	TBiIN 输入的周期时间	400		ns
$t_{w(TBH)}$	TBiIN 输入的“H”电平脉宽	180		ns
$t_{w(TBL)}$	TBiIN 输入的“L”电平脉宽	180		ns

表 28.29 定时器 B 的输入（脉宽测量模式）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TB)}$	TBiIN 输入的周期时间	400		ns
$t_{w(TBH)}$	TBiIN 输入的“H”电平脉宽	180		ns
$t_{w(TBL)}$	TBiIN 输入的“L”电平脉宽	180		ns

$$V_{CC}=5V$$

时序必要条件（在没有指定时， $V_{CC}=4.2 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.30 串行接口

符号	项目	规格值		单位
		最小	最大	
$t_{c(CK)}$	CLKi 输入的周期时间	200		ns
$t_{w(CKH)}$	CLKi 输入的“H”电平脉宽	80		ns
$t_{w(CKL)}$	CLKi 输入的“L”电平脉宽	80		ns
$t_{su(D-C)}$	RXDi 输入的准备时间	80		ns
$t_h(C-D)$	RXDi 输入的保持时间	90		ns

表 28.31 A/D 触发输入

符号	项目	规格值		单位
		最小	最大	
$t_{w(ADH)}$	ADTRG 输入的“H”电平脉宽 硬件触发输入的“H”电平脉宽	$\frac{3}{\phi_{AD}}$		ns
$t_{w(ADL)}$	ADTRG 输入的“L”电平脉宽 硬件触发输入的“L”电平脉宽	125		ns

表 28.32 外部中断  $\overline{INTi}$  输入

符号	项目	规格值		单位
		最小	最大	
$t_{w(INH)}$	$\overline{INTi}$ 输入的“H”电平脉宽	边沿检测	250	ns
		电平检测	$t_{c(CPU)}+200$	ns
$t_{w(INL)}$	$\overline{INTi}$ 输入的“L”电平脉宽	边沿检测	250	ns
		电平检测	$t_{c(CPU)}+200$	ns

表 28.33 智能 I/O 的通信功能

符号	项目	规格值		单位
		最小	最大	
$t_{c(ISCLK2)}$	ISCLK2 输入的周期时间	600		ns
$t_{w(ISCLK2H)}$	ISCLK2 输入的“H”电平脉宽	270		ns
$t_{w(ISCLK2L)}$	ISCLK2 输入的“L”电平脉宽	270		ns
$t_{su(RXD-ISCLK2)}$	ISRXD2 输入的准备时间	150		ns
$t_h(ISCLK2-RXD)$	ISRXD2 输入的保持时间	100		ns

$V_{CC}=5V$ 时序必要条件（在没有指定时， $V_{CC}=4.2 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）表 28.34 多主控 I<sup>2</sup>C 总线接口

符号	项目	规格值				单位
		标准模式		高速模式		
		最小	最大	最小	最大	
$t_{w(SCLH)}$	MSCL 输入的“H”电平脉宽	600		600		ns
$t_{w(SCLL)}$	MSCL 输入的“L”电平脉宽	600		600		ns
$t_r(SCL)$	MSCL 输入的上升时间		1000		300	ns
$t_f(SCL)$	MSCL 输入的下降时间		300		300	ns
$t_r(SDA)$	MSDA 输入的上升时间		1000		300	ns
$t_f(SDA)$	MSDA 输入的下降时间		300		300	ns
$t_{h(SDA-SCL)S}$	开始条件 / 重新开始条件后的 MSCL “H” 电平保持时间	(注 1)		$2 \times t_{c(\phi IIC)} + 40$		ns
$t_{su(SCL-SDA)P}$	重新开始条件 / 开始条件前的 MSCL “H” 电平准备时间	(注 1)		$2 \times t_{c(\phi IIC)} + 40$		ns
$t_{W(SDAH)P}$	停止条件后的 MSDA “H” 电平脉宽	(注 1)		$4 \times t_{c(\phi IIC)} + 40$		ns
$t_{SU(SDA-SCL)}$	MSDA 输入的准备时间	100		100		ns
$t_h(SCL-SDA)$	MSDA 输入的保持时间	0		0		ns

注 1. 根据 I2CSSCR 寄存器的 SSC4 ~ SSC0 位设定的值 (SSC)，用下式计算。

$$t_{h(SDA-SCL)S} = SSC \div 2 \times t_{c(\phi IIC)} + 40 [ns]$$

$$t_{su(SCL-SDA)P} = (SSC \div 2 + 1) \times t_{c(\phi IIC)} + 40 [ns]$$

$$t_{W(SDAH)P} = (SSC + 1) \times t_{c(\phi IIC)} + 40 [ns]$$

$V_{CC}=5V$ 开关特性（在没有指定时， $V_{CC}=4.2 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.35 外部总线时序（分离总线）

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_{su(S-R)}$	读操作前的片选准备时间	参照图 28.6	(注 1)		ns
$t_{h(R-S)}$	读操作后的片选保持时间		$t_{c(Base)}-15$		ns
$t_{su(A-R)}$	读操作前的地址准备时间		(注 1)		ns
$t_{h(R-A)}$	读操作后的地址保持时间		$t_{c(Base)}-15$		ns
$t_{w(R)}$	读脉宽		(注 1)		ns
$t_{su(S-W)}$	写操作前的片选准备时间		(注 1)		ns
$t_{h(W-S)}$	写操作后的片选保持时间		$1.5 \times t_{c(Base)}-15$		ns
$t_{su(A-W)}$	写操作前的地址准备时间		(注 1)		ns
$t_{h(W-A)}$	写操作后的地址保持时间		$1.5 \times t_{c(Base)}-15$		ns
$t_{w(W)}$	写脉宽		(注 1)		ns
$t_{su(D-W)}$	写操作前的数据准备时间		(注 1)		ns
$t_{h(W-D)}$	写操作后的数据保持时间		0		ns

注 1. 根据基本时钟的周期时间  $t_{c(Base)}$  和 EBC0 ~ EBC3 寄存器设定的周期数 ( $T_{su(A-R)}$ 、 $T_{w(R)}$ 、 $T_{su(A-W)}$ 、 $T_{w(W)}$ )，用下式计算。在计算出的数值为负值时，必须检查设定值。有关周期数的设定方法，请参照“9.3.5 外部总线时序”。

$$t_{su(S-R)}=t_{su(A-R)}=T_{su(A-R)} \times t_{c(Base)}-15[ns]$$

$$t_{w(R)}=T_{w(R)} \times t_{c(Base)}-10[ns]$$

$$t_{su(S-W)}=t_{su(A-W)}=T_{su(A-W)} \times t_{c(Base)}-15[ns]$$

$$t_{w(W)}=t_{su(D-W)}=T_{w(W)} \times t_{c(Base)}-10[ns]$$

$V_{CC}=5V$ 开关特性（在没有指定时， $V_{CC}=4.2 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.36 外部总线时序（多路复用总线）

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_{su}(S-ALE)$	ALE 前的片选准备时间	参照图 28.6	(注 1)		ns
$t_h(R-S)$	读操作后的片选保持时间		$1.5 \times t_{c(Base)} - 15$		ns
$t_{su}(A-ALE)$	ALE 前的地址准备时间		(注 1)		ns
$t_h(ALE-A)$	ALE 后的地址保持时间		$0.5 \times t_{c(Base)} - 5$		ns
$t_h(R-A)$	读操作后的地址保持时间		$1.5 \times t_{c(Base)} - 15$		ns
$t_d(ALE-R)$	ALE-读操作之间的延迟时间		$0.5 \times t_{c(Base)} - 5$	$0.5 \times t_{c(Base)} + 10$	ns
$t_w(ALE)$	ALE 脉宽		(注 1)		ns
$t_{dis}(R-A)$	读操作后的地址禁止时间			8	ns
$t_w(R)$	读脉宽		(注 1)		ns
$t_h(W-S)$	写操作后的片选保持时间		$1.5 \times t_{c(Base)} - 15$		ns
$t_h(W-A)$	写操作后的地址保持时间		$1.5 \times t_{c(Base)} - 15$		ns
$t_d(ALE-W)$	ALE-写操作之间的延迟时间		$0.5 \times t_{c(Base)} - 5$	$0.5 \times t_{c(Base)} + 10$	ns
$t_w(W)$	写脉宽		(注 1)		ns
$t_{su}(D-W)$	写操作前的数据准备时间		(注 1)		ns
$t_h(W-D)$	写操作后的数据保持时间		$0.5 \times t_{c(Base)}$		ns

注 1. 根据基本时钟的周期时间  $t_{c(Base)}$  和 EBC0 ~ EBC3 寄存器设定的周期数 ( $T_{su}(A-R)$ 、 $T_w(R)$ 、 $T_{su}(A-W)$ 、 $T_w(W)$ )，用下式计算。在计算出的数值为负值时，必须检查设定值。有关周期数的设定方法，请参照“9.3.5 外部总线时序”。

$$t_{su}(S-ALE) = t_{su}(A-ALE) = t_w(ALE) = (T_{su}(A-R) - 0.5) \times t_{c(Base)} - 15[\text{ns}]$$

$$t_w(R) = T_w(R) \times t_{c(Base)} - 10[\text{ns}]$$

$$t_w(W) = t_{su}(D-W) = T_w(W) \times t_{c(Base)} - 10[\text{ns}]$$

表 28.37 串行接口

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_d(C-Q)$	TXDi 输出的延迟时间	参照图 28.6		80	ns
$t_h(C-Q)$	TXDi 输出的保持时间		0		ns

表 28.38 智能 I/O 的通信功能

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_d(ISCLK2-TXD)$	ISTXD2 输出的延迟时间	参照图 28.6		180	ns
$t_h(ISCLK2-RXD)$	ISTXD2 输出的保持时间		0		ns

$$V_{CC}=5V$$

开关特性（在没有指定时， $V_{CC}=4.2 \sim 5.5V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.39 多主控 I<sup>2</sup>C 总线接口（标准模式）

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_f(SCL)$	MSCL 输出的下降时间	参照图 28.6	2		ns
$t_f(SDA)$	MSDA 输出的下降时间		2		ns
$t_d(SDA-SCL)S$	开始条件 / 重新开始条件后的 MSCL 输出的延迟时间		$20 \times t_{C(\phi IIC)} - 120$	$50 \times t_{C(\phi IIC)} - 40$	ns
$t_d(SCL-SDA)P$	MSCL “H” 电平后的重新开始条件 / 停止条件输出的延迟时间		$20 \times t_{C(\phi IIC)} + 40$	$52 \times t_{C(\phi IIC)} + 120$	ns
$t_d(SCL-SDA)$	MSDA 输出的延迟时间		$2 \times t_{C(\phi IIC)} + 40$	$3 \times t_{C(\phi IIC)} + 120$	ns

表 28.40 多主控 I<sup>2</sup>C 总线接口（高速模式）

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_f(SCL)$	MSCL 输出的下降时间	参照图 28.6	2（注 1）		ns
$t_f(SDA)$	MSDA 输出的下降时间		2（注 1）		ns
$t_d(SDA-SCL)S$	开始条件 / 重新开始条件后的 MSCL 输出的延迟时间		$10 \times t_{C(\phi IIC)} - 120$	$26 \times t_{C(\phi IIC)} - 40$	ns
$t_d(SCL-SDA)P$	MSCL “H” 电平后的重新开始条件 / 停止条件输出的延迟时间		$10 \times t_{C(\phi IIC)} + 40$	$26 \times t_{C(\phi IIC)} + 120$	ns
$t_d(SCL-SDA)$	MSDA 输出的延迟时间		$2 \times t_{C(\phi IIC)} + 40$	$3 \times t_{C(\phi IIC)} + 120$	ns

注 1. 需要用外部电路来满足 I<sup>2</sup>C 总线规格。

$V_{CC}=3.3V$ 表 28.41 电特性 (1) (在没有指定时,  $V_{CC}=3.0 \sim 3.6V$ ,  $V_{SS}=0V$ ,  $T_a=T_{opr}$ ,  $f_{(CPU)}=50MHz$ )

符号	项目		测量条件	规格值			单位
				最小	典型	最大	
$V_{OH}$	“H”电平输出电压	P0_0 ~ P0_7、P1_0 ~ P1_7、P2_0 ~ P2_7、 P3_0 ~ P3_7、P4_0 ~ P4_7、P5_0 ~ P5_7、 P6_0 ~ P6_7、P7_0 ~ P7_7、P8_0 ~ P8_4、 P8_6、P8_7、P9_0 ~ P9_7、 P10_0 ~ P10_7、P11_0 ~ P11_4、 P12_0 ~ P12_7、P13_0 ~ P13_7、 P14_3 ~ P14_6、P15_0 ~ P15_7 (注1)	$I_{OH}=-1mA$	$V_{CC}-0.6$		$V_{CC}$	V
$V_{OL}$	“L”电平输出电压	P0_0 ~ P0_7、P1_0 ~ P1_7、P2_0 ~ P2_7、 P3_0 ~ P3_7、P4_0 ~ P4_7、P5_0 ~ P5_7、 P6_0 ~ P6_7、P7_0 ~ P7_7、P8_0 ~ P8_4、 P8_6、P8_7、P9_0 ~ P9_7、 P10_0 ~ P10_7、P11_0 ~ P11_4、 P12_0 ~ P12_7、P13_0 ~ P13_7、 P14_3 ~ P14_6、P15_0 ~ P15_7 (注1)	$I_{OL}=1mA$			0.5	V

注1. 端口 P9\_0、P9\_2、P11 ~ P15 只限于 144 引脚版。端口 P9\_1 在 100 引脚版中为输入专用引脚。

$V_{CC}=3.3V$ 表 28.42 电特性 (2) (在没有指定时,  $V_{CC}=3.0 \sim 3.6V$ ,  $V_{SS}=0V$ ,  $T_a=T_{opr}$ ,  $f_{(CPU)}=50MHz$ )

符号	项目	测量条件	规格值			单位	
			最小	典型	最大		
$V_{T+}-V_{T-}$	滞后	HOLD、RDY、NMI、INT0 ~ INT8、K10 ~ K13、 TA0IN ~ TA4IN、TA0OUT ~ TA4OUT、 TB0IN ~ TB5IN、CTS0 ~ CTS8、 CLK0 ~ CLK8、RXD0 ~ RXD8、 SCL0 ~ SCL6、SDA0 ~ SDA6、SS0 ~ SS6、 SRXD0 ~ SRXD6、ADTRG、 IIO0_0 ~ IIO0_7、IIO1_0 ~ IIO1_7、UD0A、 UD0B、UD1A、UD1B、ISCLK2、ISRXD2、 IEIN、CAN0IN、CAN0WU (注1)		0.2		1.0	V
		RESET		0.2		1.8	V
$I_{IH}$	“H”电 平输入电 流	XIN、RESET、CNVSS、NSD、 P0_0 ~ P0_7、P1_0 ~ P1_7、P2_0 ~ P2_7、 P3_0 ~ P3_7、P4_0 ~ P4_7、P5_0 ~ P5_7、 P6_0 ~ P6_7、P7_0 ~ P7_7、P8_0 ~ P8_7、 P9_0 ~ P9_7、P10_0 ~ P10_7、 P11_0 ~ P11_4、P12_0 ~ P12_7、 P13_0 ~ P13_7、P14_1、P14_3 ~ P14_6、 P15_0 ~ P15_7 (注2)	$V_I=3.3V$			4.0	$\mu A$
$I_{IL}$	“L”电 平输入电 流	XIN、RESET、CNVSS、NSD、 P0_0 ~ P0_7、P1_0 ~ P1_7、P2_0 ~ P2_7、 P3_0 ~ P3_7、P4_0 ~ P4_7、P5_0 ~ P5_7、 P6_0 ~ P6_7、P7_0 ~ P7_7、P8_0 ~ P8_7、 P9_0 ~ P9_7、P10_0 ~ P10_7、 P11_0 ~ P11_4、P12_0 ~ P12_7、 P13_0 ~ P13_7、P14_1、P14_3 ~ P14_6、 P15_0 ~ P15_7 (注2)	$V_I=0V$			-4.0	$\mu A$
$R_{PULLUP}$	上拉电阻	P0_0 ~ P0_7、P1_0 ~ P1_7、P2_0 ~ P2_7、 P3_0 ~ P3_7、P5_0 ~ P5_3、P8_4、P8_6、 P8_7、P9_0 ~ P9_7、P10_0 ~ P10_7、 P11_0 ~ P11_4、P12_0 ~ P12_7、 P13_0 ~ P13_7、P14_1、P14_3 ~ P14_6、 P15_0 ~ P15_7 (注2)	$V_I=0V$	50	100	500	k $\Omega$
$R_{FXIN}$	反馈电阻	XIN			3		M $\Omega$
$R_{FXCIN}$	反馈电阻	XCIN			25		M $\Omega$

注 1. INT6 ~ INT8 只限于 144 引脚版。

注 2. 端口 P9\_0、P9\_2、P11 ~ P15 只限于 144 引脚版。端口 P9\_1 在 100 引脚版中为输入专用引脚。



$V_{CC}=3.3V$ 表 28.43 电特性 (3) (在没有指定时,  $V_{CC}=3.0 \sim 3.6V$ ,  $V_{SS}=0V$ ,  $T_a=T_{opr}$ )

符号	项目	测量条件	规格值			单位	
			最小	典型	最大		
$I_{CC}$	电源电流	在单芯片模式中, 输出引脚为开路, 其他引脚连接到 $V_{SS}$ 。	$f_{(CPU)}=50MHz$ 、 $f_{(BCLK)}=25MHz$ 、 $f_{(XIN)}=8MHz$ 、PLL 振荡、XCIN 停止、OCO 停止		32	45	mA
		XIN-XOUT 驱动能力: Low	$f_{(CPU)}=f_{SO(PLL)}/24MHz$ 、XIN 停止、PLL 自振荡、XCIN 停止、OCO 停止		9		mA
		XCIN-XCOUT 驱动能力: Low	$f_{(CPU)}=f_{(BCLK)}=f_{(XIN)}/256MHz$ 、 $f_{(XIN)}=8MHz$ 、PLL 停止、XCIN 停止、OCO 停止		670		$\mu A$
			$f_{(CPU)}=f_{(BCLK)}=32.768kHz$ 、XIN 停止、PLL 停止、XCIN 振荡、OCO 停止、主稳压器停止		180		$\mu A$
			$f_{(CPU)}=f_{(BCLK)}=f_{(OCO)}/4kHz$ 、XIN 停止、PLL 停止、XCIN 停止、OCO 振荡、主稳压器停止		190		$\mu A$
			$f_{(CPU)}=f_{(BCLK)}=f_{(XIN)}/256MHz$ 、 $f_{(XIN)}=8MHz$ 、PLL 停止、XCIN 停止、OCO 停止、等待模式、 $T_a=25^\circ C$		500	900	$\mu A$
			$f_{(CPU)}=f_{(BCLK)}=32.768kHz$ 、XIN 停止、PLL 停止、XCIN 振荡、OCO 停止、主稳压器停止、等待模式、 $T_a=25^\circ C$		8	140	$\mu A$
			$f_{(CPU)}=f_{(BCLK)}=f_{(OCO)}/4kHz$ 、XIN 停止、PLL 停止、XCIN 停止、OCO 振荡、主稳压器停止、等待模式、 $T_a=25^\circ C$		10	150	$\mu A$
			时钟停止、主稳压器停止、 $T_a=25^\circ C$		5	70	$\mu A$

$$V_{CC}=3.3V$$

表 28.44 A/D 转换特性（在没有指定时， $V_{CC}=AV_{CC}=V_{REF}=3.0 \sim 3.6V$ ， $V_{SS}=AV_{SS}=0V$ ， $T_a=T_{opr}$ ， $f_{(BCLK)}=25MHz$ ）

符号	项目	测量条件		规格值			单位
				最小	典型	最大	
—	分辨率	$V_{REF}=V_{CC}$				10	Bits
—	绝对误差	$V_{REF}=V_{CC}=3.3V$	AN_0 ~ AN_7、 AN0_0 ~ AN0_7、 AN2_0 ~ AN2_7、 AN15_0 ~ AN15_7、 ANEX0、ANEX1（注1）			$\pm 5$	LSB
			外部运算放大器连接模式			$\pm 7$	LSB
INL	积分非线性误差	$V_{REF}=V_{CC}=3.3V$	AN_0 ~ AN_7、 AN0_0 ~ AN0_7、 AN2_0 ~ AN2_7、 AN15_0 ~ AN15_7、 ANEX0、ANEX1（注1）			$\pm 5$	LSB
			外部运算放大器连接模式			$\pm 7$	LSB
DNL	微分非线性误差	$V_{REF}=V_{CC}=3.3V$				$\pm 1$	LSB
—	偏移误差					$\pm 3$	LSB
—	增益误差					$\pm 3$	LSB
$R_{LADDER}$	梯形电阻	$V_{REF}=V_{CC}$		4		20	k $\Omega$
$t_{CONV}$	转换时间 (10bit)	$\phi_{AD}=10MHz$ ，有采样 & 保持。		3.3			$\mu s$
$t_{CONV}$	转换时间 (8bit)	$\phi_{AD}=10MHz$ ，有采样 & 保持。		2.8			$\mu s$
$t_{SAMP}$	采样时间	$\phi_{AD}=10MHz$		0.3			$\mu s$
$V_{IA}$	模拟输入电压			0		$V_{REF}$	V
$\phi_{AD}$	工作时钟频率	无采样 & 保持		0.25		10	MHz
		有采样 & 保持		1		10	MHz

注 1. AN15\_0 ~ AN15\_7 只限于 144 引脚版。

表 28.45 D/A 转换特性（在没有指定时， $V_{CC}=AV_{CC}=V_{REF}=3.0 \sim 3.6V$ ， $V_{SS}=AV_{SS}=0V$ ， $T_a=T_{opr}$ ）

符号	项目	测量条件		规格值			单位
				最小	典型	最大	
—	分辨率					8	Bits
—	绝对精度					1.0	%
$t_S$	设定时间					3	$\mu s$
$R_O$	输出电阻			4	10	20	k $\Omega$
$I_{VREF}$	基准电源输入电流	(注 1)				1.0	mA

注 1 这是使用 1 个 D/A 转换器并且未使用的 D/A 转换器的  $DA_i$  寄存器 ( $i=0,1$ ) 的值为 “00h” 的情况，A/D 转换器的梯形电阻部分除外。

在将 ADOCON1 寄存器的 VCUT 位置 “0”（未连接  $V_{REF}$ ）时，也有  $I_{VREF}$  流过。

$$V_{CC}=3.3V$$

时序必要条件（在没有指定时， $V_{CC}=3.0 \sim 3.6V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.46 外部时钟输入

符号	项目	规格值		单位
		最小	最大	
$t_{c(X)}$	外部时钟输入的周期时间	62.5	250	ns
$t_{w(XH)}$	外部时钟输入的“H”电平脉宽	25		ns
$t_{w(XL)}$	外部时钟输入的“L”电平脉宽	25		ns
$t_{r(X)}$	外部时钟输入的上升时间		5	ns
$t_{f(X)}$	外部时钟输入的下降时间		5	ns
$t_w/t_c$	外部时钟输入的占空比	40	60	%

表 28.47 外部总线时序

符号	项目	规格值		单位
		最小	最大	
$t_{su(D-R)}$	读操作前的数据准备时间	40		ns
$t_h(R-D)$	读操作后的数据保持时间	0		ns
$t_{dis(R-D)}$	读操作后的数据禁止时间		$0.5 \times t_{c(Base)} + 10$	ns

表 28.48 定时器 A 的输入（事件计数器模式的计数输入）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TA)}$	TAiIN 输入的周期时间	200		ns
$t_{w(TAH)}$	TAiIN 输入的“H”电平脉宽	80		ns
$t_{w(TAL)}$	TAiIN 输入的“L”电平脉宽	80		ns

表 28.49 定时器 A 的输入（定时器模式的选通输入）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TA)}$	TAiIN 输入的周期时间	400		ns
$t_{w(TAH)}$	TAiIN 输入的“H”电平脉宽	180		ns
$t_{w(TAL)}$	TAiIN 输入的“L”电平脉宽	180		ns

表 28.50 定时器 A 的输入（单触发定时器模式的外部触发输入）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TA)}$	TAiIN 输入的周期时间	200		ns
$t_{w(TAH)}$	TAiIN 输入的“H”电平脉宽	80		ns
$t_{w(TAL)}$	TAiIN 输入的“L”电平脉宽	80		ns

$$V_{CC}=3.3V$$

时序必要条件（在没有指定时， $V_{CC}=3.0 \sim 3.6V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.51 定时器 A 的输入（脉宽调制模式的外部触发输入）

符号	项目	规格值		单位
		最小	最大	
$t_{w(TAH)}$	TAiIN 输入的“H”电平脉宽	80		ns
$t_{w(TAL)}$	TAiIN 输入的“L”电平脉宽	80		ns

表 28.52 定时器 A 的输入（事件计数器模式的递增 / 递减输入）

符号	项目	规格值		单位
		最小	最大	
$t_{c(UP)}$	TAiOUT 输入的周期时间	2000		ns
$t_{w(UPH)}$	TAiOUT 输入的“H”电平脉宽	1000		ns
$t_{w(UPL)}$	TAiOUT 输入的“L”电平脉宽	1000		ns
$t_{su(UP-TIN)}$	TAiOUT 输入的准备时间	400		ns
$t_{h(TIN-UP)}$	TAiOUT 输入的保持时间	400		ns

表 28.53 定时器 B 的输入（事件计数器模式的计数输入）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TB)}$	TBiIN 输入的周期时间（单边沿计数）	200		ns
$t_{w(TBH)}$	TBiIN 输入的“H”电平脉宽（单边沿计数）	80		ns
$t_{w(TBL)}$	TBiIN 输入的“L”电平脉宽（单边沿计数）	80		ns
$t_{c(TB)}$	TBiIN 输入的周期时间（双边沿计数）	200		ns
$t_{w(TBH)}$	TBiIN 输入的“H”电平脉宽（双边沿计数）	80		ns
$t_{w(TBL)}$	TBiIN 输入的“L”电平脉宽（双边沿计数）	80		ns

表 28.54 定时器 B 的输入（脉冲周期测量模式）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TB)}$	TBiIN 输入的周期时间	400		ns
$t_{w(TBH)}$	TBiIN 输入的“H”电平脉宽	180		ns
$t_{w(TBL)}$	TBiIN 输入的“L”电平脉宽	180		ns

表 28.55 定时器 B 的输入（脉宽测量模式）

符号	项目	规格值		单位
		最小	最大	
$t_{c(TB)}$	TBiIN 输入的周期时间	400		ns
$t_{w(TBH)}$	TBiIN 输入的“H”电平脉宽	180		ns
$t_{w(TBL)}$	TBiIN 输入的“L”电平脉宽	180		ns

$$V_{CC}=3.3V$$

时序必要条件（在没有指定时， $V_{CC}=3.0 \sim 3.6V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.56 串行接口

符号	项目	规格值		单位
		最小	最大	
$t_{c(CK)}$	CLKi 输入的周期时间	200		ns
$t_{w(CKH)}$	CLKi 输入的“H”电平脉宽	80		ns
$t_{w(CKL)}$	CLKi 输入的“L”电平脉宽	80		ns
$t_{su(D-C)}$	RXDi 输入的准备时间	80		ns
$t_h(C-D)$	RXDi 输入的保持时间	90		ns

表 28.57 A/D 触发输入

符号	项目	规格值		单位
		最小	最大	
$t_{w(ADH)}$	ADTRG 输入的“H”电平脉宽 硬件触发输入的“H”电平脉宽	$\frac{3}{\phi_{AD}}$		ns
$t_{w(ADL)}$	ADTRG 输入的“L”电平脉宽 硬件触发输入的“L”电平脉宽	125		ns

表 28.58 外部中断  $\overline{INTi}$  输入

符号	项目	规格值		单位
		最小	最大	
$t_{w(INH)}$	$\overline{INTi}$ 输入的“H”电平脉宽	边沿检测	250	ns
		电平检测	$t_{c(CPU)}+200$	ns
$t_{w(INL)}$	$\overline{INTi}$ 输入的“L”电平脉宽	边沿检测	250	ns
		电平检测	$t_{c(CPU)}+200$	ns

表 28.59 智能 I/O 的通信功能

符号	项目	规格值		单位
		最小	最大	
$t_{c(ISCLK2)}$	ISCLK2 输入的周期时间	600		ns
$t_{w(ISCLK2H)}$	ISCLK2 输入的“H”电平脉宽	270		ns
$t_{w(ISCLK2L)}$	ISCLK2 输入的“L”电平脉宽	270		ns
$t_{su(RXD-ISCLK2)}$	ISRXD2 输入的准备时间	150		ns
$t_h(ISCLK2-RXD)$	ISRXD2 输入的保持时间	100		ns

$$V_{CC}=3.3V$$

时序必要条件（在没有指定时， $V_{CC}=3.0 \sim 3.6V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.60 多主控 I<sup>2</sup>C 总线接口

符号	项目	规格值				单位
		标准模式		高速模式		
		最小	最大	最小	最大	
$t_{W(SCLH)}$	MSCL 输入的“H”电平脉宽	600		600		ns
$t_{W(SCLL)}$	MSCL 输入的“L”电平脉宽	600		600		ns
$t_{r(SCL)}$	MSCL 输入的上升时间		1000		300	ns
$t_{f(SCL)}$	MSCL 输入的下降时间		300		300	ns
$t_{r(SDA)}$	MSDA 输入的上升时间		1000		300	ns
$t_{f(SDA)}$	MSDA 输入的下降时间		300		300	ns
$t_{h(SDA-SCL)S}$	开始条件 / 重新开始条件后的 MSCL “H” 电平保持时间	(注 1)		$2 \times t_{c(\phi IIC)} + 40$		ns
$t_{su(SCL-SDA)P}$	重新开始条件 / 停止条件前的 MSCL “H” 电平准备时间	(注 1)		$2 \times t_{c(\phi IIC)} + 40$		ns
$t_{w(SDAH)P}$	停止条件后的 MSDA “H” 电平脉宽	(注 1)		$4 \times t_{c(\phi IIC)} + 40$		ns
$t_{su(SDA-SCL)}$	MSDA 输入的准备时间	100		100		ns
$t_{h(SCL-SDA)}$	MSDA 输入的保持时间	0		0		ns

注 1. 根据 I2CSSCR 寄存器的 SSC4 ~ SSC0 位设置的值 (SSC)，用下式计算。

$$t_{h(SDA-SCL)S} = SSC \div 2 \times t_{c(\phi IIC)} + 40 [ns]$$

$$t_{su(SCL-SDA)P} = (SSC \div 2 + 1) \times t_{c(\phi IIC)} + 40 [ns]$$

$$t_{w(SDAH)P} = (SSC + 1) \times t_{c(\phi IIC)} + 40 [ns]$$

$V_{CC}=3.3V$ 开关特性（在没有指定时， $V_{CC}=3.0 \sim 3.6V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.61 外部总线时序（分离总线）

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_{su(S-R)}$	读操作前的片选准备时间	参照图 28.6	(注 1)		ns
$t_{h(R-S)}$	读操作后的片选保持时间		$t_{c(Base)}-15$		ns
$t_{su(A-R)}$	读操作前的地址准备时间		(注 1)		ns
$t_{h(R-A)}$	读操作后的地址保持时间		$t_{c(Base)}-15$		ns
$t_{w(R)}$	读脉宽		(注 1)		ns
$t_{su(S-W)}$	写操作前的片选准备时间		(注 1)		ns
$t_{h(W-S)}$	写操作后的片选保持时间		$1.5 \times t_{c(Base)}-15$		ns
$t_{su(A-W)}$	写操作前的地址准备时间		(注 1)		ns
$t_{h(W-A)}$	写操作后的地址保持时间		$1.5 \times t_{c(Base)}-15$		ns
$t_{w(W)}$	写脉宽		(注 1)		ns
$t_{su(D-W)}$	写操作前的数据准备时间		(注 1)		ns
$t_{h(W-D)}$	写操作后的数据保持时间		0		ns

注 1. 根据基本时钟的周期时间  $t_{c(Base)}$  和 EBC0 ~ EBC3 寄存器设定的周期数 ( $T_{su(A-R)}$ 、 $T_{w(R)}$ 、 $T_{su(A-W)}$ 、 $T_{w(W)}$ )，用下式计算。在计算出的数值为负值时，必须检查设定值。有关周期数的设定方法，请参照“9.3.5 外部总线时序”。

$$t_{su(S-R)} = t_{su(A-R)} = T_{su(A-R)} \times t_{c(Base)} - 15 [ns]$$

$$t_{w(R)} = T_{w(R)} \times t_{c(Base)} - 10 [ns]$$

$$t_{su(S-W)} = t_{su(A-W)} = T_{su(A-W)} \times t_{c(Base)} - 15 [ns]$$

$$t_{w(W)} = t_{su(D-W)} = T_{w(W)} \times t_{c(Base)} - 10 [ns]$$

$$V_{CC}=3.3V$$

开关特性（在没有指定时， $V_{CC}=3.0 \sim 3.6V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.62 外部总线时序（多路复用总线）

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_{su}(S-ALE)$	ALE 前的片选准备时间	参照图 28.6	(注 1)		ns
$t_h(R-S)$	读操作后的片选保持时间		$1.5 \times t_{c(Base)} - 15$		ns
$t_{su}(A-ALE)$	ALE 前的地址准备时间		(注 1)		ns
$t_h(ALE-A)$	ALE 后的地址保持时间		$0.5 \times t_{c(Base)} - 5$		ns
$t_h(R-A)$	读操作后的地址保持时间		$1.5 \times t_{c(Base)} - 15$		ns
$t_d(ALE-R)$	ALE-读操作之间的延迟时间		$0.5 \times t_{c(Base)} - 5$	$0.5 \times t_{c(Base)} + 10$	ns
$t_w(ALE)$	ALE 脉宽		(注 1)		ns
$t_{dis}(R-A)$	读操作后的地址禁止时间			8	ns
$t_w(R)$	读脉宽		(注 1)		ns
$t_h(W-S)$	写操作后的片选保持时间		$1.5 \times t_{c(Base)} - 15$		ns
$t_h(W-A)$	写操作后的地址保持时间		$1.5 \times t_{c(Base)} - 15$		ns
$t_d(ALE-W)$	ALE-写操作之间的延迟时间		$0.5 \times t_{c(Base)} - 5$	$0.5 \times t_{c(Base)} + 10$	ns
$t_w(W)$	写脉宽		(注 1)		ns
$t_{su}(D-W)$	写操作前的数据准备时间		(注 1)		ns
$t_h(W-D)$	写操作后的数据保持时间		$0.5 \times t_{c(Base)}$		ns

注 1. 根据基本时钟的周期时间  $t_{c(Base)}$  和 EBC0 ~ EBC3 寄存器设定的周期数 ( $T_{su}(A-R)$ 、 $T_w(R)$ 、 $T_{su}(A-W)$ 、 $T_w(W)$ )，用下式计算。在计算出的数值为负值时，必须检查设定值。有关采样数的设定方法，请参照“9.3.5 外部总线时序”。

$$t_{su}(S-ALE) = t_{su}(A-ALE) = (T_{su}(A-R) - 0.5) \times t_{c(Base)} - 15 [ns]$$

$$t_w(ALE) = (T_{su}(A-R) - 0.5) \times t_{c(Base)} - 20 [ns]$$

$$t_w(R) = T_w(R) \times t_{c(Base)} - 10 [ns]$$

$$t_w(W) = t_{su}(D-W) = T_w(W) \times t_{c(Base)} - 10 [ns]$$

表 28.63 串行接口

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_d(C-Q)$	TXDi 输出的延迟时间	参照图 28.6		80	ns
$t_h(C-Q)$	TXDi 输出的保持时间		0		ns

表 28.64 智能 I/O 的通信功能

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_d(ISTXD2)$	ISTXD2 输出的延迟时间	参照图 28.6		180	ns
$t_h(ISTXD2)$	ISTXD2 输出的保持时间		0		ns



$V_{CC}=3.3V$

开关特性（在没有指定时， $V_{CC}=3.0 \sim 3.6V$ ， $V_{SS}=0V$ ， $T_a=T_{opr}$ ）

表 28.65 多主控 I<sup>2</sup>C 总线接口（标准模式）

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_f(SCL)$	MSCL 输出的下降时间	参照图 28.6	2		ns
$t_f(SDA)$	MSDA 输出的下降时间		2		ns
$t_d(SDA-SCL)S$	开始条件 / 重新开始条件后的 MSCL 输出的延迟时间		$20 \times t_{c(\phi IIC)} - 120$	$50 \times t_{c(\phi IIC)} - 40$	ns
$t_d(SCL-SDA)P$	MSCL “H” 电平后的重新开始条件 / 停止条件输出的延迟时间		$20 \times t_{c(\phi IIC)} + 40$	$52 \times t_{c(\phi IIC)} + 120$	ns
$t_d(SCL-SDA)$	MSDA 输出的延迟时间		$2 \times t_{c(\phi IIC)} + 40$	$3 \times t_{c(\phi IIC)} + 120$	ns

表 28.66 多主控 I<sup>2</sup>C 总线接口（高速模式）

符号	项目	测量条件	规格值		单位
			最小	最大	
$t_f(SCL)$	MSCL 输出的下降时间	参照图 28.6	2（注 1）		ns
$t_f(SDA)$	MSDA 输出的下降时间		2（注 1）		ns
$t_d(SDA-SCL)S$	开始条件 / 重新开始条件后的 MSCL 输出的延迟时间		$10 \times t_{c(\phi IIC)} - 120$	$26 \times t_{c(\phi IIC)} - 40$	ns
$t_d(SCL-SDA)P$	MSCL “H” 电平后的重新开始条件 / 停止条件输出的延迟时间		$10 \times t_{c(\phi IIC)} + 40$	$26 \times t_{c(\phi IIC)} + 120$	ns
$t_d(SCL-SDA)$	MSDA 输出的延迟时间		$2 \times t_{c(\phi IIC)} + 40$	$3 \times t_{c(\phi IIC)} + 120$	ns

注 1. 需要用外部电路来满足 I<sup>2</sup>C 总线规格。

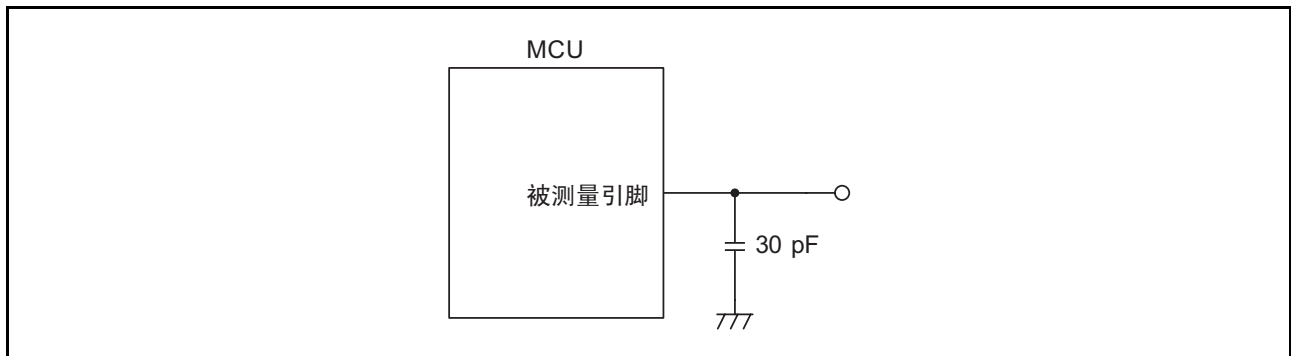


图 28.6 开关特性的测量电路

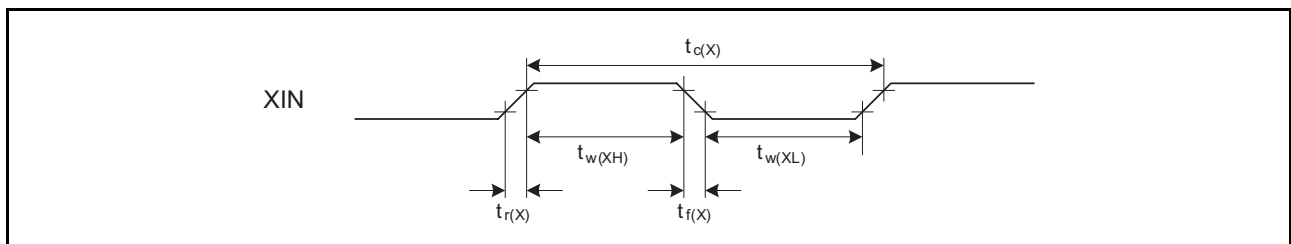


图 28.7 外部时钟输入的时序图

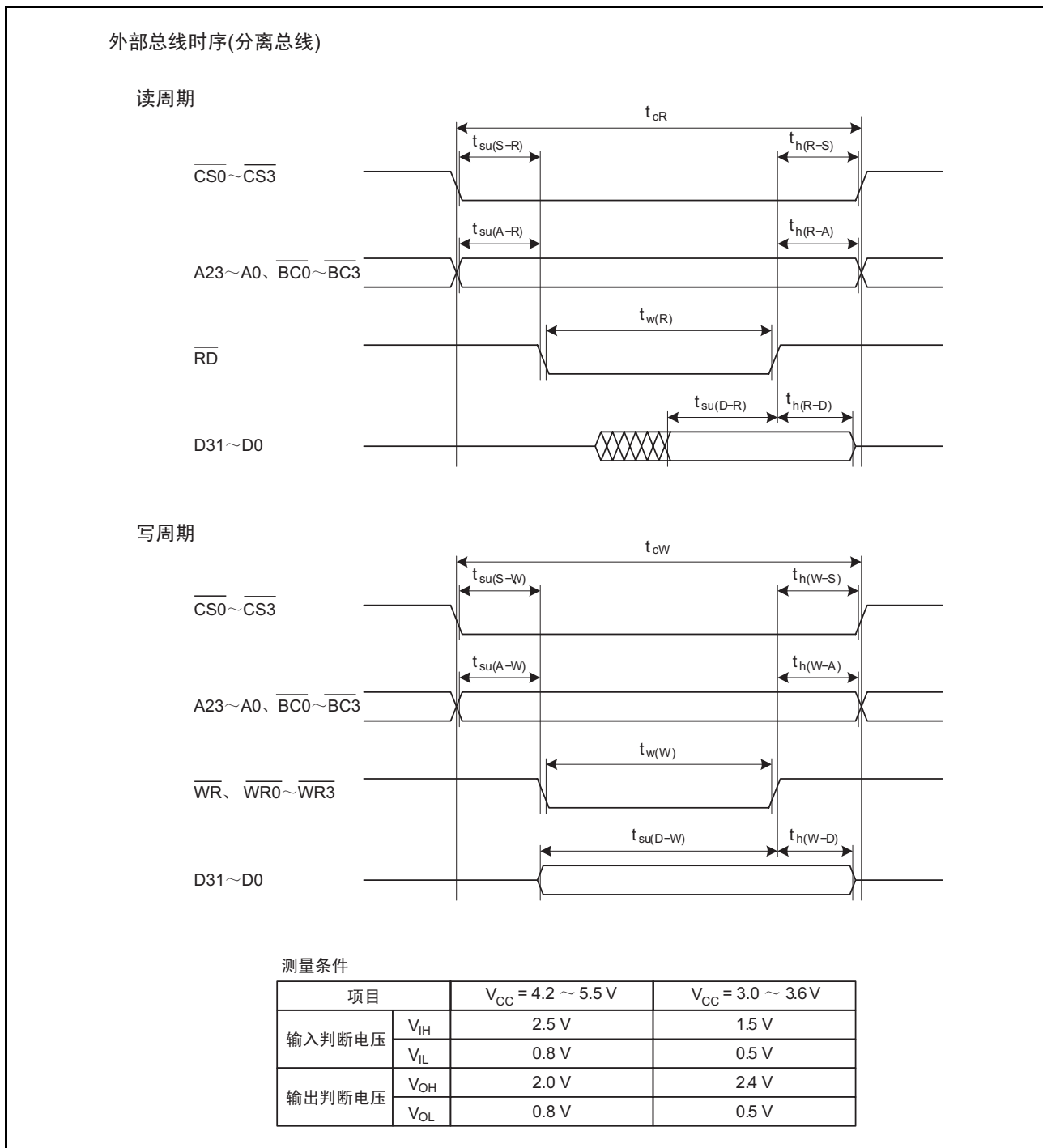


图 28.8 外部总线的时序图（分离总线）

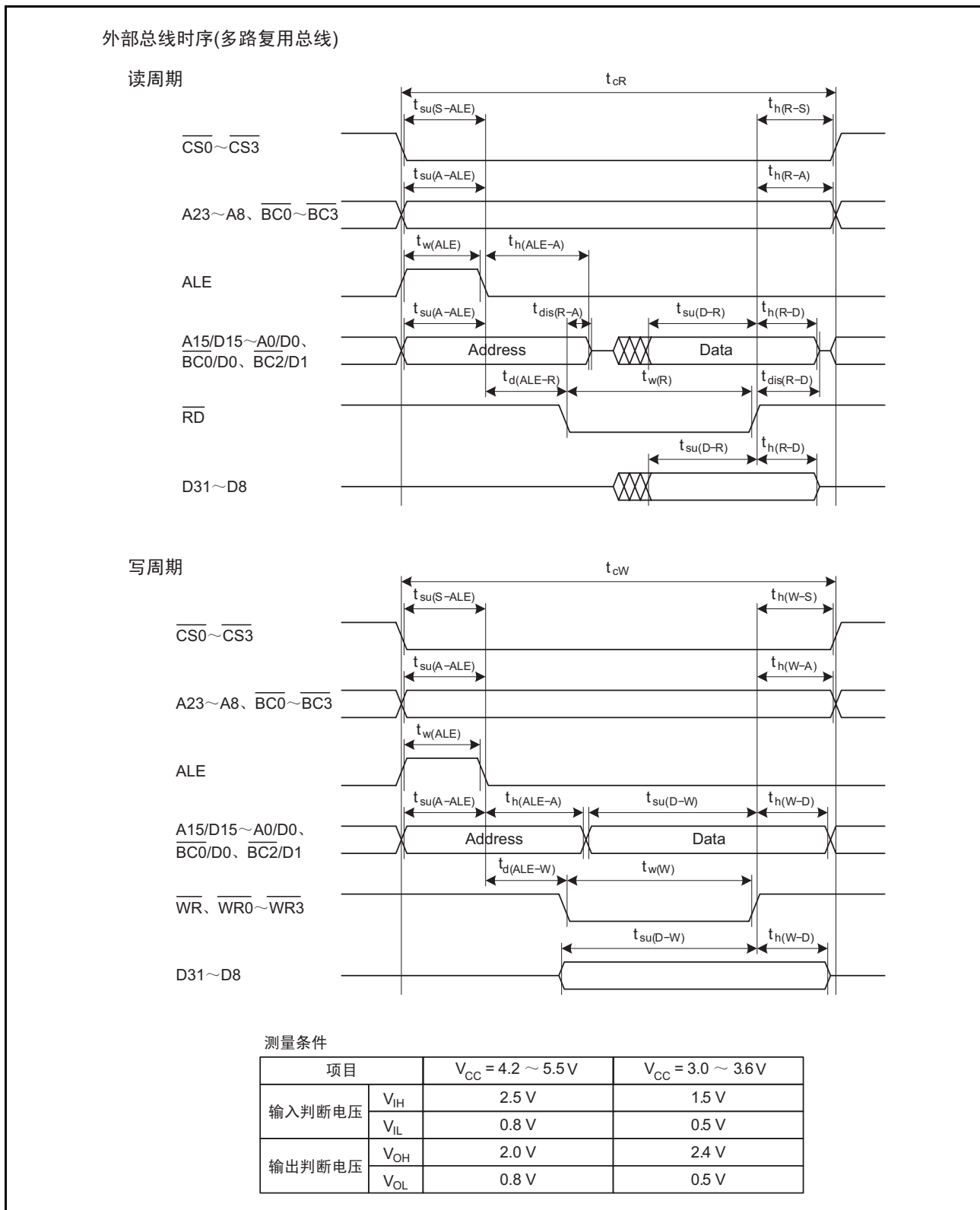


图 28.9 外部总线的时序图 (多路复用总线)

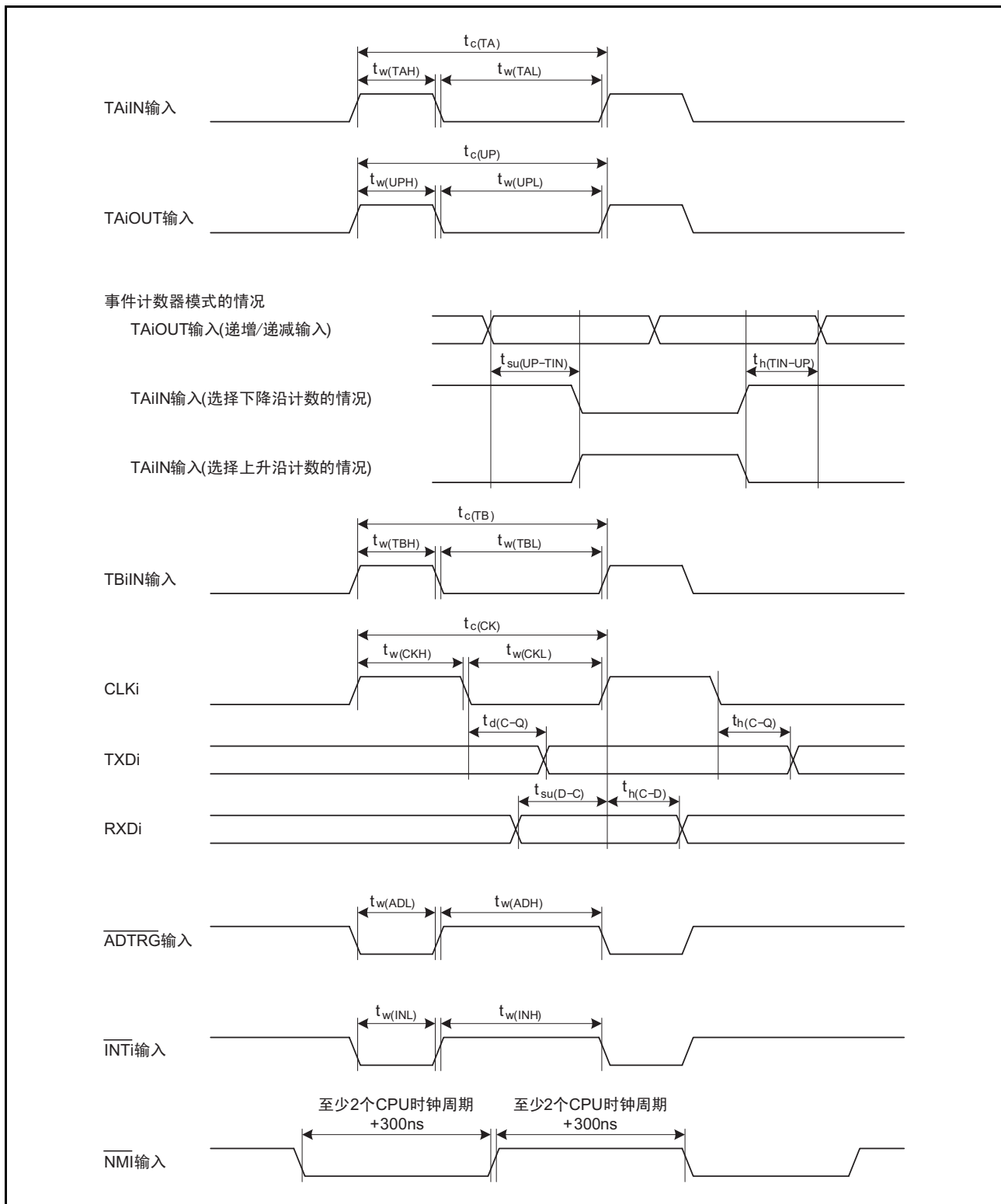
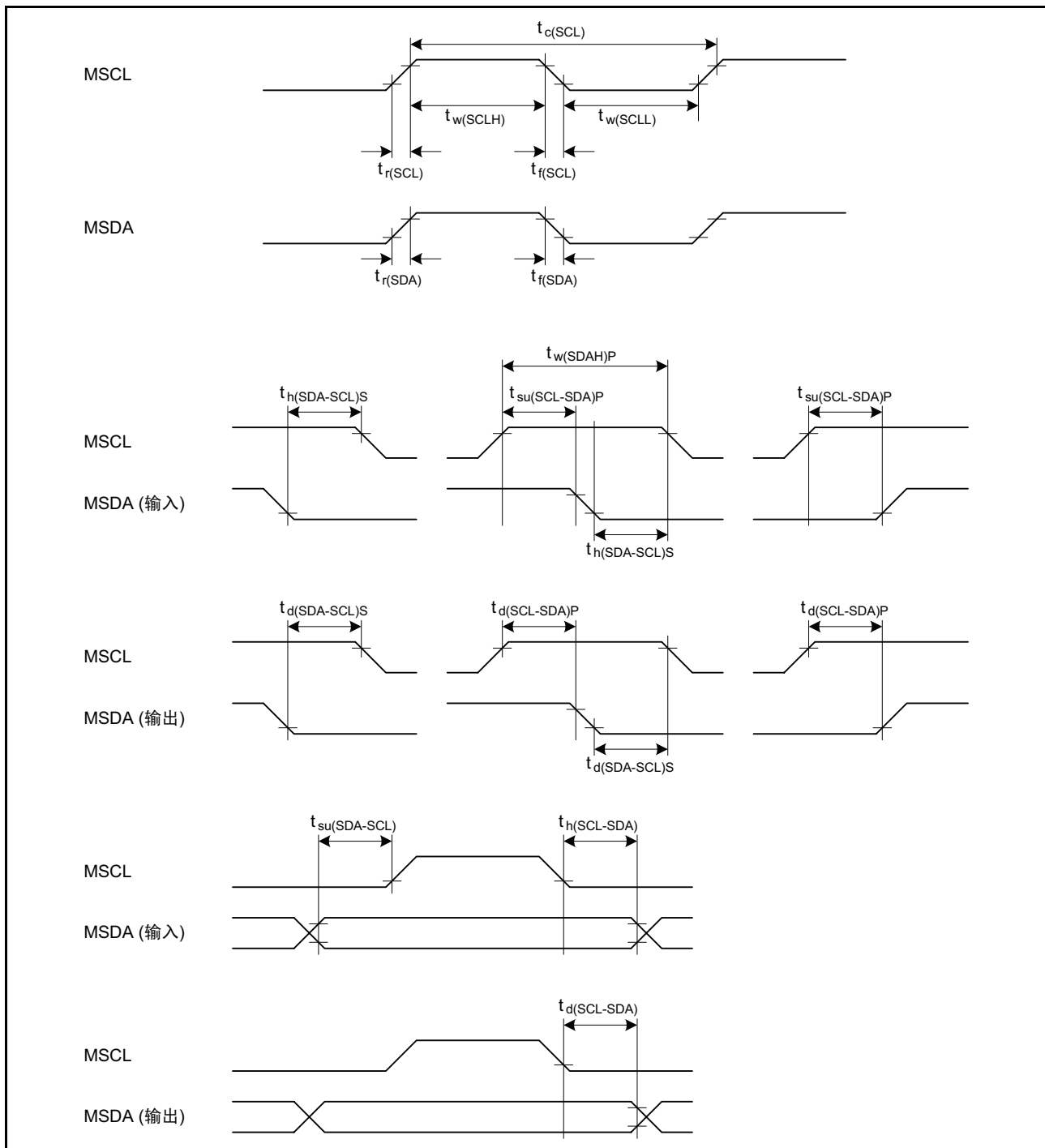


图 28.10 外围功能的时序图

图 28.11 多主控 I<sup>2</sup>C 总线接口的时序图

## 29. 使用时的注意事项

### 29.1 有关电路板设计的注意事项

#### 29.1.1 电源引脚

在设计电路板时，为了避免在引脚名相同的电源引脚之间发生电位差，必须注意：

- 所有VSS引脚都必须连接相同的GND，为了避免在引脚之间发生电位差，尽可能使用粗的布线连接各引脚。
- 所有VCC引脚都必须连接相同的电源，为了避免在引脚之间发生电位差，尽可能使用粗的布线连接各引脚。

为了防止噪声引起的误动作，各VCC引脚和VSS引脚之间必须插入频率特性良好的电容器。必须尽可能靠近电源引脚，使用最短并且较粗的布线连接0.1 $\mu$ F左右电容器。

#### 29.1.2 电源电压

如果电源电压超出电特性规定的推荐工作电压范围，就不能保证单片机的运行。必须在电源电压低于推荐的工作电压前将RESET引脚置为“L”电平。

### 29.2 设定寄存器时的注意事项

#### 29.2.1 含有只写位的寄存器

当给含有只写（WO）位的寄存器设定值时，不能使用读 - 修改 - 写指令。读 - 修改 - 写指令是先读对象地址的值，然后在更改此值后进行回写的指令。读 - 修改 - 写指令以及含有只写（WO）位的寄存器如表 29.1 和表 29.2 所示。需要在加工前一个值后决定下一个值时，必须将写到寄存器的值也写到RAM，然后在更改RAM的内容后，用MOV指令将下一个值传送到寄存器。

表 29.1 读 - 修改 - 写指令

功能	助记符
传送	MOV <i>Dir</i>
位处理	BCLR、BMC <i>nd</i> 、BNOT、BSET、BTSTC、BTSTS
移位	ROL <i>C</i> 、ROR <i>C</i> 、ROT、SHA、SHL
算术运算	ABS、ADC、ADCF、ADD、ADSF、DEC、DIV、DIVU、DIVX、EXTS、EXTZ、INC、MUL、MULU、NEG、SBB、SUB
10 进制运算	DADC、DADD、DSBB、DSUB
浮点运算	ADDF、DIVF、MULF、SUBF
逻辑运算	AND、NOT、OR、XOR

表 29.2 含有只写 (WO) 位的寄存器一览表

模块	寄存器	符号	地址	
看门狗定时器	看门狗定时器的开始寄存器	WDTS	04404Eh	
定时器 A	定时器 A0 的寄存器 (注 1)	TA0	0347h ~ 0346h	
	定时器 A1 的寄存器 (注 1)	TA1	0349h ~ 0348h	
	定时器 A2 的寄存器 (注 1)	TA2	034Bh ~ 034Ah	
	定时器 A3 的寄存器 (注 1)	TA3	034Dh ~ 034Ch	
	定时器 A4 的寄存器 (注 1)	TA4	034Fh ~ 034Eh	
	递增 / 递减选择寄存器	UDF	0344h	
三相马达控制定时器功能	定时器 B2 的中断发生频率设定计数器	ICTB2	030Dh	
	定时器 A1-1 的寄存器	TA11	0303h ~ 0302h	
	定时器 A2-1 的寄存器	TA21	0305h ~ 0304h	
	定时器 A4-1 的寄存器	TA41	0307h ~ 0306h	
	死区时间定时器	DTT	030Ch	
串行接口	UART0 位速率寄存器	U0BRG	0369h	
	UART1 位速率寄存器	U1BRG	02E9h	
	UART2 位速率寄存器	U2BRG	0339h	
	UART3 位速率寄存器	U3BRG	0329h	
	UART4 位速率寄存器	U4BRG	02F9h	
	UART5 位速率寄存器	U5BRG	01C9h	
	UART6 位速率寄存器	U6BRG	01D9h	
	UART7 位速率寄存器	U7BRG	01E1h	
	UART8 位速率寄存器	U8BRG	01E9h	
	UART0 发送缓冲寄存器	U0TB	036Bh ~ 036Ah	
	UART1 发送缓冲寄存器	U1TB	02EBh ~ 02EAh	
	UART2 发送缓冲寄存器	U2TB	033Bh ~ 033Ah	
	UART3 发送缓冲寄存器	U3TB	032Bh ~ 032Ah	
	UART4 发送缓冲寄存器	U4TB	02FBh ~ 02FAh	
	UART5 发送缓冲寄存器	U5TB	01CBh ~ 01CAh	
	UART6 发送缓冲寄存器	U6TB	01DBh ~ 01DAh	
	UART7 发送缓冲寄存器	U7TB	01E3h ~ 01E2h	
	UART8 发送缓冲寄存器	U8TB	01EBh ~ 01EAh	
	智能 I/O	组 2 的 SIO 发送缓冲寄存器	G2TB	016Dh ~ 016Ch
	CAN 模块	CAN0 接收 FIFO 指针控制寄存器	C0RFPCR	047F49h
CAN0 发送 FIFO 指针控制寄存器		C0TFPCR	047F4Bh	

注 1. 只限于单触发定时器模式和脉宽调制模式。

## 29.3 使用时钟发生电路时的注意事项

### 29.3.1 副时钟

#### 29.3.1.1 振荡电路常数匹配の確認

必须确认驱动能力 High 和驱动能力 Low 时的副时钟振荡电路的振荡电路常数的匹配情况。  
有关振荡电路常数的匹配，请向振荡器厂商询问。

### 29.3.2 功率控制

只有在要使用的时钟振荡稳定后，才能转换基本时钟源和时钟的分频比。一旦将 CM3 寄存器的 CM31 位置“1”，内部振荡器瞬间就开始振荡，因此不需要振荡稳定的等待时间。

如果将基本时钟源从 PLL 时钟转换为低速时钟（将 CCR 寄存器的 BCS 位置“1”），就必须使用 MOVL 指令或者 OR.L 指令。

- 使用汇编语言时的例子  
OR.L                    #80h,0004h
- 使用C语言时的例子  
asm("OR.L            #80h,0004h");

#### 29.3.2.1 停止模式

- 在通过复位从停止模式返回时，必须在主时钟振荡充分稳定前将“L”电平输入到RESET引脚。

#### 29.3.2.2 降低功耗的要点

请在进行系统设计和编程时参考。

- 引脚处理  
如果将输入引脚置为开路，就可能有穿透电流流过。因此必须将未使用引脚设定为输入并且各引脚通过电阻连接到VSS（下拉），或者将未使用引脚设定为输出并将引脚置为开路。
- A/D转换器  
在不进行A/D转换时，必须将AD0CON1寄存器的VCUT位置“0”（未连接VREF）；在进行A/D转换时，必须先将VCUT位置“1”（连接VREF），再至少等待1μs，然后开始A/D转换。
- D/A转换器  
在不进行D/A转换时，必须将DACON寄存器的DAiE位（i=0,1）置“0”（禁止输出）并将DAi寄存器置“00h”。
- 外围功能的停止  
在向等待模式转移时，能通过CM0寄存器的CM02位停止外围功能的时钟源，从而降低功耗。但是，fC32不停止振荡。



## 29.4 使用总线时的注意事项

### 29.4.1 系统设计时的注意事项

在使用存储器扩展模式并在执行闪存的 CPU 改写时， $\overline{CS0}$  区域和  $\overline{CS3}$  区域有以下的限制。必须将想在 CPU 改写时存取的器件分配到  $\overline{CS1}$  区域和  $\overline{CS2}$  区域。

- 如果在 CPU 改写时设定 FEBC0 寄存器、FEBC3 寄存器，对应区域就为分离总线。如果通过多路复用总线连接外部器件，就不能存取该器件。
- 如果在 CPU 改写时设定 FEBC0 寄存器、FEBC3 寄存器，对应区域的总线时序就发生变化。根据所设定的内容，有可能不能存取外部器件。

### 29.4.2 设定寄存器时的注意事项

#### 29.4.2.1 片选边界设定寄存器

在不使用存储器扩展模式时，不能更改 CB01、CB12、CB23 寄存器的值，保持复位后的值。

在使用存储器扩展模式时，这些寄存器与是否使用各片选区域无关，必须全部设定指定范围内的值。

#### 29.4.2.2 外部总线控制寄存器

EBC0 寄存器和 EBC3 寄存器分别与 FEBC0 寄存器和 FEBC3 寄存器共用地址。如果在改写闪存时更改 FEBC0 寄存器和 FEBC3 寄存器，就必须在改写闪存结束后重新设定 EBC0 寄存器和 EBC3 寄存器。

## 29.5 使用中断时的注意事项

### 29.5.1 ISP 的设定

在复位后，ISP（中断堆栈指针）被初始化为“00000000h”。如果在给ISP设定值前接受中断，就会导致程序失控，因此必须在接受中断前给ISP设定值。ISP的设定值必须是4的倍数，设定4的倍数能减少存储器的存取次数从而加快中断响应顺序的执行速度。

尤其在使用NMI时，不能禁止中断，因此必须在程序的开始位置设定ISP后，将PM2寄存器的PM24位置“1”（NMI有效）。

### 29.5.2 NMI

- 在将PM2寄存器的PM24位置“1”（NMI有效）后，不能禁止NMI。在不使用NMI时，PM24位必须保持“0”。
- 当PM2寄存器的PM24位为“1”（NMI有效）时，P8寄存器的P8\_5位只能用于确认NMI引脚的状态，而不能用作通用端口。

### 29.5.3 外部中断

- $\overline{\text{INT}}_i$  引脚（ $i=0\sim 8$ ）的输入信号需要满足电特性规定的信号宽度。如果小于规定的最小宽度，就有可能无法接受中断。
- 在通过 $\text{INT}_i\text{IC}$ 寄存器（ $i=0\sim 8$ ）的POL位和LVS位、IFSR0寄存器的IFSR0 $i$ 位（ $i=0\sim 5$ ）、IFSR1寄存器的IFSR1 $j$ 位（ $j=i-6\sim 6\sim 8$ ）转换 $\overline{\text{INT}}_i$ 引脚的有效边沿或者有效电平时，对应的IR位有可能变为“1”（有中断请求）。因此，必须在将 $\text{INT}_i\text{IC}$ 寄存器的ILVL2 $\sim$ ILVL0位置“000b”（禁止中断）后改写这些位。在改写后，必须将对应的IR位置“0”（无中断请求），然后设定ILVL2 $\sim$ ILVL0位。
- 在通过 $\text{INT}_i\text{IC}$ 寄存器（ $i=6\sim 8$ ）的POL位和LVS位、IFSR1寄存器的IFSR1 $j$ 位（ $j=i-6$ ）转换 $\overline{\text{INT}}_i$ 引脚的有效边沿或者有效电平时，IIOKIR寄存器（ $k=9\sim 11$ ）的INTIR位有可能变为“1”（有中断请求）。因此，必须在将IIOKIE寄存器的INTIE位置“0”（禁止中断）后改写这些位。在改写后，必须将对应的INTIR位置“0”（无中断请求），然后将INTIE位置“1”。

## 29.6 使用 DMA 控制器时的注意事项

### 29.6.1 DMAC 相关寄存器的设定

- 必须在要设定通道的DMDi寄存器 (i=0~3)的MDi1~MDi0位为“00b”(禁止DMA传送)的状态下,设定DMAC的相关寄存器,最后将MDi1~MDi0位置“01b”(单次传送)或者“11b”(重复传送)。也必须在MDi1~MDi0位为“00b”(禁止DMA传送)时改写DMDi寄存器的UDAi、USAi和BWi1~BWi0位。
- 如果在允许DMA传送后需要改写DMAC的相关寄存器,首先为了不产生DMA传送请求,必须停止作为DMA启动源的外围功能,其次将要改写的通道的DMDi寄存器的MDi1~MDi0位置“00b”(禁止DMA传送),然后进行改写。
- 一旦接受DMA传送请求,即使将DMDi寄存器的MDi1~MDi0位置“00b”(禁止DMA传送),也无法禁止DMA传送。此时,不能在DMA传送结束前更改除MDi1~MDi0位以外的DMAC相关寄存器的设定。
- 在设定DMiSL寄存器和DMiSL2寄存器后,必须先通过外围总线时钟至少等待6个时钟,然后给DMDi寄存器的MDi1~MDi0位写“01b”(单次传送)或者“11b”(重复传送)。

### 29.6.2 读 DMAC 相关寄存器

- 必须按照以下的顺序分别连续读DMiSL寄存器和DMiSL2寄存器:  
DM0SL→DM1SL→DM2SL→DM3SL  
DM0SL2→DM1SL2→DM2SL2→DM3SL2

## 29.7 使用定时器时的注意事项

### 29.7.1 定时器 A 和定时器 B 的共同事项

复位后,定时器停止运行。在设定模式、计数源和计数器的值后,必须将 TABSR 寄存器或者 TBSR 寄存器的 TAI*S* 位 (i=0~4) 或者 TB*jS* 位 (j=0~5) 置“1”(开始计数)。

必须在对应的 TAI*S* 位或者 TB*jS* 位为“0”(停止计数)的状态下,更改以下的寄存器和位:

- TAI*MR* 寄存器和 TB*jMR* 寄存器
- UDF 寄存器
- ONSF 寄存器的 TAZIE 位、TAOTGL 位和 TAOTGH 位
- TRGSR 寄存器

### 29.7.2 定时器 A

#### 29.7.2.1 定时器模式的情况

- 通过读 TAI 寄存器,能随时知道正在计数的计数器的值。如果在进行重新加载时读 TAI 寄存器,就能读到“FFFFh”。如果在停止计数时将值设定到 TAI 寄存器,然后在开始计数前读 TAI 寄存器,就能读到所设定的值。

#### 29.7.2.2 事件计数器模式的情况

- 通过读 TAI 寄存器,能随时知道正在计数的计数器的值。如果在进行重新加载时读 TAI 寄存器,就能在发生下溢时读到“FFFFh”,而在发生上溢时读到“0000h”。如果在停止计数时将值设定到 TAI 寄存器,然后在计数器开始计数前读 TAI 寄存器,就能读到所设定的值。

### 29.7.2.3 单触发定时器模式的情况

- 如果在计数时将 TABSR 寄存器的 TAI<sub>S</sub> 位置 “0”（停止计数），就会出现以下的状态：
  - 计数器停止计数并重新加载 TAI 寄存器的设定值。
  - TAI<sub>OUT</sub> 引脚输出 “L” 电平。
  - 在 1 个 CPU 时钟周期后，TAI<sub>IC</sub> 寄存器的 IR 位变为 “1”（有中断请求）。
- 因为单触发定时器的输出与内部生成的计数源同步，所以在选择 TAI<sub>IN</sub> 引脚的输入作为触发时，从触发输入到单触发定时器的输出期间，会产生最长 1 个计数源时钟的延迟。
- 如果在以下任意一种情况下设定定时器的运行模式，IR 位就变为 “1”。因此，在使用定时器 A<sub>i</sub> 中断（IR 位）时，必须在进行下述的设定后将 IR 位置 “0”。
  - 在复位后选择单触发定时器模式时。
  - 在将运行模式从定时器模式更改为单触发定时器模式时。
  - 在将运行模式从事件计数器模式更改为单触发定时器模式时。
- 如果在计数时发生再触发，计数器就先进行 1 次递减计数，接着重新加载 TAI 寄存器（i=0~4）的设定值，然后继续计数。如果要在计数时产生再触发，就必须在从上次触发开始至少经过 1 个定时器计数源时钟后，产生再触发。
- 如果已经选择 TAI<sub>IN</sub> 引脚的触发输入作为计数开始条件，就不能在定时器 A 的计数值变为 “0000h” 前的 300ns 期间输入再触发，否则单触发定时器就可能不继续计数而停止。

### 29.7.2.4 脉宽调制模式的情况

- 如果在以下任意一种情况下设定定时器的运行模式，IR 位就变为 “1”。因此，在使用定时器 A<sub>i</sub> 中断（IR 位）时，必须在进行以下的设定后将 IR 位置 “0”。
  - 在复位后选择 PWM 模式时。
  - 在将运行模式从定时器模式更改为 PWM 模式时。
  - 在将运行模式从事件计数器模式更改为 PWM 模式时。
- 如果在输出 PWM 脉冲时将 TAI<sub>S</sub> 位置 “0”（停止计数），就会出现以下的状态：
  - 计数器停止计数。
  - 在从 TAI<sub>OUT</sub> 引脚输出 “H” 电平时，输出电平变为 “L” 并且 IR 位变为 “1”。
  - 在从 TAI<sub>OUT</sub> 引脚输出 “L” 电平时，输出电平不变，IR 位也不变。

### 29.7.3 定时器 B

#### 29.7.3.1 定时器模式和事件计数器模式的情况

- 通过读TBj寄存器（j=0~5），能随时知道正在计数的计数器的值。如果在进行重新加载时读TBj寄存器，就能读到“FFFFh”。如果在停止计数时将值设定到TBj寄存器，然后在开始计数前读TBj寄存器，就能读到所设定的值。

#### 29.7.3.2 脉冲周期测量 / 脉宽测量模式的情况

- 要将TBjMR寄存器的MR3位置“0”（无上溢）时，必须在TBjS位为“1”（开始计数）时，在MR3位变为“1”（有上溢）并至少经过1个计数源时钟后，写TBjMR寄存器。
- 在只要检测上溢时，必须使用TBjIC寄存器的IR位。只能在通过中断处理程序判断中断源时使用MR3位。
- 在开始计数时，计数器的值为不定值。因此，在开始计数后第1次输入有效边沿前，计数器有可能发生上溢并且发生定时器Bj的中断请求。
- 在开始计数后第1次输入有效边沿时，因为计数器的值为不定值，所以不定值被传送到重加载寄存器。此时，不发生定时器Bj的中断请求。
- 如果在开始计数后更改TBjMR寄存器的MR1~MR0位，IR位就可能变为“1”（有中断请求）。如果给MR1~MR0位重写相同的值，IR位就不变。
- 在脉宽测量模式中连续测量脉宽。必须通过程序判断测量结果是“H”电平宽度还是“L”电平宽度。
- 在脉冲周期测量模式中，如果在计数器发生上溢的同时输入有效边沿，就只产生1次中断请求，所以无法确认是否输入了有效边沿。因此，必须在计数器不发生上溢的范围内使用。
- 在脉宽测量模式中，必须通过定时器Bj的中断处理程序，读端口的电平来判断计数器是发生上溢还是输入了有效边沿。

## 29.8 使用三相马达控制定时器功能时的注意事项

### 29.8.1 关机功能

- 当PM2寄存器的PM24位为“1”（NMI有效）、INVC0寄存器的INV03位为“1”（允许三相马达控制定时器的输出）并且INV02位为“1”（使用三相马达控制定时器功能）时，如果给NMI引脚输入“L”电平，TA1OUT、TA2OUT、TA4OUT引脚就变为高阻抗。

### 29.8.2 寄存器的设定

- 不能在定时器B2发生上溢前后给TAi1寄存器（i=1,2,4）设定值。  
在给TAi1寄存器设定值时，必须先读TB2寄存器的值，在确认发生上溢前还有充分的时间后再进行设定。为了缩短读TB2寄存器和写TAi1寄存器之间的时间间隔，不能在此期间执行中断处理等。如果在读TB2寄存器后到发生上溢前没有充分的时间，就必须在发生上溢后重新设定TAi1寄存器。

## 29.9 使用串行接口时的注意事项

### 29.9.1 UiBRG 寄存器 (i=0 ~ 8) 的变更

- 必须在设定 UiC0 寄存器的 CLK1 ~ CLK0 位后写 UiBRG 寄存器。如果更改 CLK1 ~ CLK0 位，就必须重新设定 UiBRG 寄存器。
- 如果给 UiBRG 寄存器写 “00h”，计数器就立即开始运行并且有可能变为 “FFh”。此时，到重新加载所设定的 “00h” 为止，需要的时间约为 256 个时钟。在重新加载 “00h” 后，根据设定，变为无分频。

### 29.9.2 时钟同步模式

#### 29.9.2.1 外部时钟的选择

- 在已选择外部时钟的情况下，必须在 UiC0 寄存器 (i=0 ~ 8) 的 CKPOL 位为 “0” (与发送/接收时钟的下降沿同步输出发送数据，与上升沿同步输入接收数据) 时并且外部时钟为 “H” 电平的状态下，或者在 CKPOL 位为 “1” (与发送/接收时钟的上升沿同步输出发送数据，与下降沿同步输入接收数据) 时并且外部时钟为 “L” 电平的状态下，满足以下的条件：
  - UiC1 寄存器的 TE 位为 “1” (允许发送)。
  - UiC1 寄存器的 RE 位为 “1” (允许接收)。
  - UiC1 寄存器的 TI 位为 “0” (UiTB 寄存器有数据)。在只进行发送时，不需要设定 RE 位。

#### 29.9.2.2 接收

- 时钟同步模式通过发送控制电路对发送/接收时钟进行控制。因此，即使在只进行接收时也必须进行发送的设定。在进行接收时，将虚拟数据从 TXDi 引脚 (i=0 ~ 8) 输出到外部。
- 在连续接收数据的情况下，如果在 UiC1 寄存器的 RI 位为 “1” (UiRB 寄存器有数据) 时接收到下一个接收数据的第 7 位，就发生溢出错误，并且 UiRB 寄存器的 OER 位变为 “1” (发生溢出错误)。此时，UiRB 寄存器为不定值。在发生溢出错误时，SiRIC 寄存器的 IR 位不变为 “1”。

### 29.9.3 特殊模式 1 (I<sup>2</sup>C 模式)

在生成开始条件、停止条件或者重新开始条件时，必须在将 UiSMR4 寄存器 (i=0 ~ 6) 的 STSPSEL 位置 “0” 后，至少等待半个发送/接收时钟，然后将各条件的生成位 (STAREQ 位、RSTAREQ 位、STPREQ 位) 从 “0” 置为 “1”。

## 29.10 使用 A/D 转换器时的注意事项

### 29.10.1 设计电路板时的注意事项

- 为了防止噪声引起的误动作和闩锁以及减少转换误差，必须在 AVCC 引脚、VREF 引脚、模拟输入引脚（AN\_0~AN\_7、AN0\_0~AN0\_7、AN2\_0~AN2\_7、AN15\_0~AN15\_7）和 AVSS 引脚之间分别插入电容器。引脚的处理例子如图 29.1 所示。

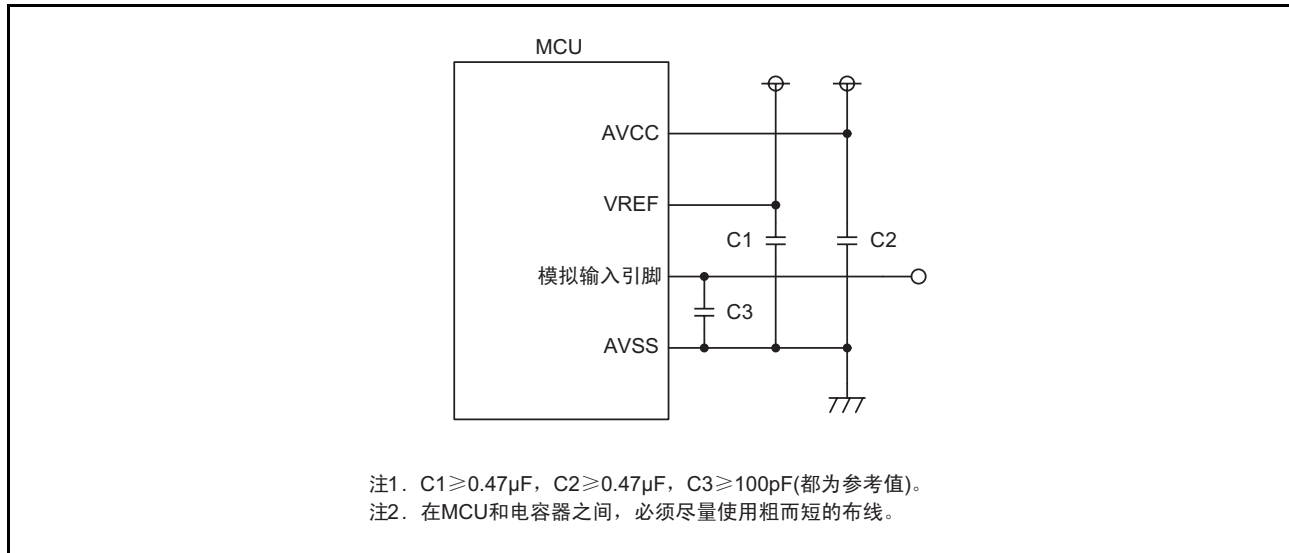


图 29.1 A/D 相关引脚的处理例子

- 在使用键输入中断时，AN\_4~AN\_7（4个引脚）都不能用作模拟输入引脚（如果 A/D 输入电压不超过  $V_{IL}$ ，就产生键输入中断请求）。
- 当  $AVCC=VREF=VCC$  时，AN\_0~AN\_7、AN0\_0~AN0\_7、AN2\_0~AN2\_7、AN15\_0~AN15\_7、ANEX0、ANEX1 的 A/D 输入电压不能超过 VCC。

### 29.10.2 编程时的注意事项

- 必须在停止 A/D 转换时（发生触发前）写 AD0CON0 寄存器（ADST 位除外）、AD0CON1 寄存器、AD0CON2 寄存器、AD0CON3 寄存器和 AD0CON4 寄存器。
- 在将 AD0CON1 寄存器的 VCUT 位从“0”（未连接 VREF）置为“1”（连接 VREF）时，必须在至少经过  $1\mu\text{s}$  后开始 A/D 转换。在不进行 A/D 转换时，为了减少消耗电流，必须将 VCUT 位从“1”置为“0”。
- 必须将用于模拟输入引脚对应的端口方向位置“0”（输入），并将对应的端口功能选择寄存器的 ASEL 位置“1”（用作 A/D 输入）。
- 当 AD0CON0 寄存器的 TRG 位为“1”（外部触发、硬件触发）时，必须将 ADTRG 引脚对应的端口方向位（PD9\_7 位）置“0”（输入）。
- 当  $VCC=4.2\sim 5.5\text{V}$  时， $\phi_{AD}$  的频率不能超过 16MHz；当  $VCC=3.0\sim 4.2\text{V}$  时， $\phi_{AD}$  的频率不能超过 10MHz。在没有采样 & 保持功能时， $\phi_{AD}$  的频率至少为 250kHz；在有采样 & 保持功能时， $\phi_{AD}$  的频率至少为 1MHz。
- 如果更改 A/D 运行模式（AD0CON0 寄存器的 MD1~MD0 位和 AD0CON1 寄存器的 MD2 位），就必须通过 AD0CON0 寄存器的 CH2~CH0 位或者 AD0CON1 寄存器的 SCAN1~SCAN0 位重新选择模拟输入引脚。

- 在将A/D转换结果保存到AD0i寄存器 (i=0~7) 时, 如果CPU读AD0i寄存器, 就有可能将错误值保存到AD0i寄存器。必须在确认A/D转换结束后读AD0i寄存器。  
在使用单次模式或者单次扫描模式时, 必须在确认AD0IC寄存器的IR位为“1”(有中断请求)后读对象AD0i寄存器。  
在使用重复模式、重复扫描模式0或者重复扫描模式1时, 如果将AD0CON3寄存器的DUS位置“1”(DMAC利用模式有效), 就能在每次转换结束时产生中断请求。同样, 必须在确认AD0IC寄存器的IR位为“1”(有中断请求)后读AD00寄存器。
- 在A/D转换过程中, 如果通过将AD0CON0寄存器的ADST位置“0”(停止A/D转换)来停止A/D转换, A/D转换器的转换结果就为不定值, 而且没有进行A/D转换的AD0i寄存器的值也有可能为不定值。如果停止A/D转换, 就不能使用任何AD0i寄存器的值。
- 在DMAC利用模式中, 不能使用外部触发, 也不能通过程序读AD00寄存器。
- 在单次扫描模式中, 如果在A/D转换过程中通过将AD0CON0寄存器的ADST位置“0”(停止A/D转换)来停止A/D转换, 就有可能产生中断请求, 与扫描是否结束无关。在停止A/D转换时, 必须在禁止中断后将ADST位置“0”(停止A/D转换)。

## 29.11 有关改写闪存的注意事项

### 29.11.1 有关电源电压的注意事项

- 改写闪存时的电源电压必须为电特性规定的电压范围内的电压。如果在改写闪存时电压的波动超过保证值, 就无法保证闪存。

### 29.11.2 有关硬件复位的注意事项

- 在改写闪存时, 不能进行硬件复位。

### 29.11.3 有关闪存保护的注意事项

- 如果将错误的的数据写到ID码的保存地址, 就无法在标准串行输入/输出模式中读写闪存。

### 29.11.4 编程时的注意事项

- 在低速模式或者低功耗模式中, 不能将FMCR寄存器的FEW位置“1”(CPU改写模式)。
- 通过NMI、看门狗定时器中断、振荡停止检测中断或者低电压检测中断, 将编程、块擦除、锁定位编程和保护位编程中断。如果这些软件命令被中断, 就必须在擦除该块后再次执行相同的命令。尤其在块擦除被中断时, 因为锁定位和保护位的值不确定, 所以必须在解除锁定后再次实施块擦除。



### 29.11.5 使用中断时的注意事项

- EW0 模式
  - 能通过将可变向量表中的向量写到RAM区，使用该向量的中断。
  - 如果发生NMI、看门狗定时器中断、振荡停止检测中断或者低电压检测中断，就自动进入读阵列模式，因此也能在改写闪存时使用这些中断。在发生中断时，闪存的改写被中断，并且FMR0寄存器和FMSR0寄存器被复位。必须在中断处理结束后，先将FMR1寄存器的LBD位置“1”（锁定位保护无效），然后再次执行改写程序。
  - 因为BRK指令、INTO指令、UND指令参照闪存中的数据，所以不能使用这些指令。
- EW1 模式
  - 在进行编程或者块擦除时，不能接受可变向量表中有向量的中断。
  - 不能发生看门狗定时器中断。
  - 如果发生NMI、振荡停止检测中断或者低电压检测中断，就自动进入读阵列模式，因此也能在改写闪存时使用这些中断。在发生中断时，闪存的改写被中断，并且FMR0寄存器和FMSR0寄存器被复位。必须在中断处理结束后，先将FMR0寄存器的EWM位置“1”（EW1模式），并且将FMR1寄存器的LBD位置“1”（锁定位保护无效），然后再次执行改写程序。

### 29.11.6 有关改写控制程序的改写注意事项

- EW0 模式
  - 如果在对保存改写控制程序的块进行改写时电源电压降低，就无法正常地对改写控制程序进行改写，因此以后有可能无法改写闪存。如果无法改写，就必须使用串行编程器或者并行编程器进行改写。
- EW1 模式
  - 不能对保存改写控制程序的块进行改写。

### 29.11.7 编程 / 擦除次数和软件命令的执行时间

- 软件命令（编程、块擦除、锁定位编程、保护位编程）的执行时间随着编程/擦除次数的增加而变长。如果编程/擦除次数超过电特性规定的编程/擦除次数，软件命令的执行时间就会明显变长，因此软件命令的等待时间必须至少设定为电特性规定的最大时间。

### 29.11.8 其他注意事项

- 电特性规定的编程/擦除次数是能保证原始性能的编程/擦除次数的最大值。如果超过这个次数，并非立即就无法进行编程和擦除。
- 在批量生产时，不能使用调试中重复编程、擦除过的芯片。

# 附录

## 附录 1. 封装尺寸图

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP144-20x20-0.50	PLQP0144KA-A	144P6Q-A / FP-144L / FP-144LV	1.2g

NOTE)  
 1. DIMENSIONS \*\*1\* AND \*\*2\* DO NOT INCLUDE MOLD FLASH.  
 2. DIMENSION \*\*3\* DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	19.9	20.0	20.1
A <sub>2</sub>	—	1.4	—
H <sub>D</sub>	21.8	22.0	22.2
H <sub>E</sub>	21.8	22.0	22.2
A	—	—	1.7
A <sub>1</sub>	0.05	0.1	0.15
b <sub>p</sub>	0.17	0.22	0.27
b <sub>1</sub>	—	0.20	—
c	0.09	0.145	0.20
c <sub>1</sub>	—	0.125	—
θ	0°	—	8°
Ⓜ	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z <sub>D</sub>	—	1.25	—
Z <sub>E</sub>	—	1.25	—
L	0.35	0.5	0.65
L <sub>1</sub>	—	1.0	—

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP100-14x14-0.50	PLQP0100KB-A	100P6Q-A / FP-100U / FP-100UV	0.6g

NOTE)  
 1. DIMENSIONS \*\*1\* AND \*\*2\* DO NOT INCLUDE MOLD FLASH.  
 2. DIMENSION \*\*3\* DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A <sub>2</sub>	—	1.4	—
H <sub>D</sub>	15.8	16.0	16.2
H <sub>E</sub>	15.8	16.0	16.2
A	—	—	1.7
A <sub>1</sub>	0.05	0.1	0.15
b <sub>p</sub>	0.15	0.20	0.25
b <sub>1</sub>	—	0.18	—
c	0.09	0.145	0.20
c <sub>1</sub>	—	0.125	—
θ	0°	—	8°
Ⓜ	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z <sub>D</sub>	—	1.0	—
Z <sub>E</sub>	—	1.0	—
L	0.35	0.5	0.65
L <sub>1</sub>	—	1.0	—

## 索引

## A

A0 .....	24
A1 .....	24
A2 .....	24
A3 .....	24
AD00 ~ AD07 .....	271
AD0CON0 .....	268
AD0CON1 .....	269
AD0CON2 .....	270
AD0CON3 .....	270
AD0CON4 .....	271
AD0IC .....	140

## B

B 标志 .....	25
BCN0IC ~ BCN6IC .....	140
BRK2 指令中断 .....	132
BRK 指令中断 .....	132
BTSR .....	300
标志保存寄存器 .....	26
标志寄存器 .....	24

## C

C0AFSR .....	395
C0BCR .....	372
C0CLKR .....	371
C0CSSR .....	394
C0CTLR .....	367
C0ECSR .....	401
C0EIC .....	140
C0EIER .....	396
C0EIFR .....	398
C0FIDCR0、C0FIDCR1 .....	375
C0FRIC .....	140
C0FTIC .....	140
COMBj (j=0 ~ 31) .....	378
COMCTLj (j=0 ~ 31) .....	381
COMIER .....	380
COMKIVLR .....	376
COMKR0 ~ COMKR7 .....	374
COMSMR .....	392
COMSSR .....	393
CORECR .....	400
CORFCR .....	384
CORFPCR .....	386
CORIC .....	140
COSTR .....	389
COTCR .....	403
COTECR .....	400
COTFCR .....	387
COTFPCR .....	388
COTIC .....	140

C0TSR .....	402
C0WIC .....	140
CB01 .....	110
CB12 .....	110
CB23 .....	111
C 标志 .....	24
CCR .....	82
CM0 .....	83
CM1 .....	84
CM2 .....	84
CM3 .....	85
CPSRF .....	85
CRCD .....	287
CRCIN .....	287
CSOP0 .....	109
CSOP1 .....	109
CSOP2 .....	110
程序计数器 .....	24
处理器中断优先级 .....	25, 139
从属接收的例子 .....	361

## D

DA0、DA1 .....	286
DACON .....	286
D 标志 .....	25
DCR0 ~ DCR3 .....	27, 161
DCT0 ~ DCT3 .....	26, 161
DDA0 ~ DDA3 .....	27, 162
DDR0 ~ DDR3 .....	27, 162
DM0IC ~ DM3IC .....	140
DM0SL2 ~ DM3SL2 .....	158
DM0SL ~ DM3SL .....	158
DMA 模式寄存器 .....	26
DMA 目标地址寄存器 .....	27
DMA 目标地址重加载寄存器 .....	27
DMA 源地址寄存器 .....	27
DMA 源地址重加载寄存器 .....	27
DMA 终端计数寄存器 .....	26
DMA 终端计数重加载寄存器 .....	27
DMD0 ~ DMD3 .....	26, 161
DP 位 .....	26
DSA0 ~ DSA3 .....	27, 162
DSR0 ~ DSR3 .....	27, 162
DTT .....	214
DVCR .....	74
单步中断 .....	132
低电压检测电路 .....	73
低电压检测中断 .....	75, 132
地址寄存器 .....	24
定点数位置的指定位 .....	26
堆栈指针 .....	24
堆栈指针指定标志 .....	25

<b>E</b>	
EBC0 ~ EBC3 .....	113

<b>F</b>	
FB .....	24
FBPM0 .....	457
FBPM1 .....	458
FEBC0、FEBC3 .....	455
FLG .....	24
FMCR .....	454
FMR0 .....	456
FMR1 .....	456
FMSR0 .....	457
FO 标志 .....	25
FPR0 .....	455
FU 标志 .....	25
非屏蔽中断 .....	131
浮点上溢标志 .....	25
浮点舍入运算模式 .....	26
浮点下溢标志 .....	25
符号标志 .....	25

<b>G</b>	
G0BCR0 ~ G2BCR0 .....	297
G0BCR1、G1BCR1 .....	298
G0BT ~ G2BT .....	297
G0FE ~ G2FE .....	305
G0FS、G1FS .....	304
G0PO0 ~ G0PO7 .....	304
G0POCR0 ~ G0POCR7 .....	302
G0TM0 ~ G0TM7 .....	302
G0TMCR0 ~ G0TMCR7 .....	301
G0TPR6、G0TPR7 .....	301
G1PO0 ~ G1PO7 .....	304
G1POCR0 ~ G1POCR7 .....	302
G1TM0 ~ G1TM7 .....	302
G1TMCR0 ~ G1TMCR7 .....	301
G1TPR6、G1TPR7 .....	301
G2BCR1 .....	299
G2CR .....	330
G2MR .....	330
G2PO0 ~ G2PO7 .....	304
G2POCR0 ~ G2POCR7 .....	303
G2RB .....	329
G2RTP .....	305
G2TB .....	329
高速中断 .....	133

<b>I</b>	
I2CCCR .....	342
I2CCR0 .....	340
I2CCR1 .....	345
I2CCR2 .....	348
I2CMR .....	353

I2CSAR .....	339
I2CSR .....	350
I2CSSCR .....	344
I2CTRSR .....	338
I 标志 .....	25
ICTB2 .....	210
IDB0、IDB1 .....	209
IEAR .....	331
IECR .....	331
IERIF .....	332
IETIF .....	332
IFS0 .....	440
IFS1 .....	440
IFS2 .....	441
IFS3 .....	442
IFSR0 .....	147, 232
IFSR1 .....	148, 232
IIO0IC ~ IIO11IC .....	140
IIO0IE ~ IIO11IE .....	152
IIO0IR ~ IIO11IR .....	151
INT0IC ~ INT5IC .....	141
INTB .....	24
INT 指令中断 .....	132
INVC0 .....	207
INVC1 .....	208
IOBC .....	209
IPL .....	25, 139
ISP .....	24

<b>J</b>	
寄存器组指定标志 .....	25
进位标志 .....	24
静态基址寄存器 .....	24

<b>K</b>	
KUPIC .....	140
看门狗定时器中断 .....	132
可屏蔽中断 .....	131

<b>L</b>	
LVDC .....	74
零标志 .....	25

<b>M</b>	
MOD .....	172

<b>N</b>	
NMI .....	132



**V**

VCT.....	26
VRCR.....	72

**W**

WDC.....	155
WDTS.....	155
外围功能中断.....	133
未定义指令中断.....	131

**X**

X0R ~ X15R.....	289
XYC.....	289
向量寄存器.....	26

**Y**

Y0R ~ Y15R.....	290
硬件中断.....	132
用户堆栈指针.....	24

**Z**

Z 标志.....	25
振荡停止检测中断.....	132
帧基址寄存器.....	24
中断	
分类.....	131
寄存器的压栈.....	144
中断堆栈指针.....	24
中断控制寄存器.....	140
中断请求级.....	141
中断向量表基址寄存器.....	24
中断响应时间.....	143
中断响应顺序.....	143
中断允许标志.....	25
主控发送的例子.....	360

修订记录	R32C/117 群 用户手册 硬件篇
------	---------------------

Rev.	发行日	修订内容	
		页	修订处
1.00	2010.07.30	—	初版发行

---

R32C/117 群  
用户手册 硬件篇

Publication Date: Rev1.00 Jul 30, 2010

Published by: Renesas Electronics Corporation

---



**SALES OFFICES**

Renesas Electronics Corporation

<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.

**Renesas Electronics America Inc.**  
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130

**Renesas Electronics Canada Limited**  
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220

**Renesas Electronics Europe Limited**  
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K  
Tel: +44-1628-585-100, Fax: +44-1628-585-900

**Renesas Electronics Europe GmbH**  
Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

**Renesas Electronics (China) Co., Ltd.**  
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

**Renesas Electronics (Shanghai) Co., Ltd.**  
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China  
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

**Renesas Electronics Hong Kong Limited**  
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

**Renesas Electronics Taiwan Co., Ltd.**  
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

**Renesas Electronics Singapore Pte. Ltd.**  
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632  
Tel: +65-6213-0200, Fax: +65-6278-8001

**Renesas Electronics Malaysia Sdn.Bhd.**  
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

**Renesas Electronics Korea Co., Ltd.**  
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

R32C/117群



瑞萨电子株式会社

RCJ09B0082-0100