

To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

Notice

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

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

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



用户手册初稿

78K0R/KG3-L

16 位单片微控制器

μPD78F1013

μPD78F1014

文件号 U19460CA1V0UD00 (第 1 版)

发布日期 2009 年 6 月 NS

© NEC Electronics Corporation 2009

日本印刷

[备忘录]

① 输入引脚处的电压波形

输入噪声或一个反射波引起的波形失真可能导致错误发生。如果由于噪声等的影响使CMOS设备的输入电压范围保持在 V_{IL} （最大值）和 V_{IH} （最小值）之间，设备可能发生错误。在输入电平固定时以及输入电平从 V_{IL} （最大值）过渡到 V_{IH} （最小值）时的传输期间，要防止散射噪声影响设备。

② 未使用的输入引脚的处理

CMOS设备的输入端保持开路可能导致误操作。如果一个输入引脚未被连接，则由于噪声等原因可能会产生内部输入电平，从而导致误操作。CMOS设备的操作特性与Bipolar或NMOS设备不同。CMOS设备的输入电平必须借助上拉或下拉电路固定在高电平或低电平。每一个未使用引脚都应该通过附加电阻连接到 V_{DD} 或GND。如果有可能尽量定义为输出引脚。对未使用引脚的处理因设备而异，必须遵循与设备相关的规定和说明。

③ ESD 防护措施

如果MOS设备周围有强电场，将会击穿氧化栅极，从而影响设备的运行。因此必须采取措施，尽可能防止静电产生。一旦有静电，必须立即释放。对于环境必须有适当的控制。如果空气干燥，应当使用增湿器。建议避免使用容易产生静电的绝缘体。半导体设备的存放和运输必须使用抗静电容器、抗静电屏蔽袋或导电材料容器。所有的测试和测量工具包括工作台和工作面必须良好接地。操作员应当佩戴静电消除手带以保证良好接地。不能用手直接接触半导体设备。对于装配有半导体设备的PW板也应采取类似的静电防范措施。

④ 初始化之前的状态

在上电时MOS设备的初始状态是不确定的。在刚刚上电之后，具有复位功能的MOS设备并没有被初始化。因此上电不能保证输出引脚的电平，I/O设置和寄存器的内容。设备在收到复位信号后才进行初始化。具有复位功能的设备在上电后必须立即进行复位操作。

⑤ 电源开关顺序

在一个设备的内部操作和外部接口使用不同的电源的情况下，按照规定，应先在接通内部电源之后再接通外部电源。当关闭电源时，按照规定，先关闭外部电源再关闭内部电源。如果电源开关顺序颠倒，可能会导致设备的内部组件过电压，产生异常电流，从而引起内部组件的误操作和性能的退化。

对于每个设备电源的正确开关顺序必须依据设备的规范说明分别进行判断。

⑥ 电源关闭状态下的输入信号

不要向没有加电的设备输入信号或提供I/O上拉电源。因为输入信号或提供I/O上拉电源将引起电流注入，从而引起设备的误操作，并产生异常电流，从而使内部组件退化。每个设备电源关闭时的信号输入必须依据设备的规范说明分别进行判断。

Windows 和 Windows NT 是 Microsoft Corporation 在美国及其他国家和地区的注册商标或商标。

PC/AT 是 International Business Machines Corporation 的商标。

EEPROM 是 NEC Electronics Corporation 的注册商标。

SuperFlash 是 Silicon Storage Technology, Inc 在许多国家及地区的注册商标，包括美国和日本。

注意：该产品使用的SuperFlash® 技术经Silicon Storage Technology, Inc授权。

- 本文档所登载的内容有效期截止至 2009 年 6 月，信息先于产品的生产周期发布。将来可能未经预先通知而更改。在实际进行生产设计时，请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可，禁止复制或转载本文件中的内容。否则因本文档所登载内容引发的错误，本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的，对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息，应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失，本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性，但用户应同意并知晓，我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害（包括死亡）的危险，用户务必在其设计中采用必要的安全措施，如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为：“标准等级”、“专业等级”以及“特殊等级”三种质量等级。
“特殊等级”仅适用于为特定用途而根据用户指定的质量保证程序所开发的日电电子产品。另外，各种日电电子产品的推荐用途取决于其质量等级，详见如下。用户在选用本公司的产品时，请事先确认产品的质量等级。

“标准等级”： 计算机，办公自动化设备，通信设备，测试和测量设备，音频·视频设备，家电，加工机械以及产业用机器人。

“专业等级”： 运输设备（汽车、火车、船舶等），交通用信号控制设备，防灾装置，防止犯罪装置，各种安全装置以及医疗设备（不包括专门为维持生命而设计的设备）。

“特殊等级”： 航空器械，宇航设备，海底中继设备，原子能控制系统，为了维持生命的医疗设备、用于维持生命的装置或系统等。

除在本公司半导体产品的数据表或数据手册等资料中另有特别规定以外，本公司半导体产品的质量等级均为“标准等级”。如果用户希望在本公司设计意图以外使用本公司半导体产品，务必事先与本公司销售代表联系以确认本公司是否同意为该项应用提供支持。

（注）

- （1）本声明中的“本公司”是指日本电气电子株式会社（NEC Electronics Corporation）及其控股公司。
- （2）本声明中的“本公司产品”是指所有由日本电气电子株式会社开发或制造的产品或为日本电气电子株式会社（定义如上）开发或制造的产品。

M8E 02.11-1

引言

读者对象

本手册适用于那些希望了解 78K0R/KG3-L 功能，并设计开发应用系统和程序的工程师。
目标产品如下。

- 78K0R/KG3-L: μ PD78F1013, 78F1014

目的

本手册用于帮助用户了解下面**组件**中描述的功能。

组件

78K0R/KG3-L 手册主要分为两个部分：手册和指令（与 78K0R 系列通用）。

78K0R/KG3-L 用户手册 (本手册)	78K0R 微控制器 用户手册 指令

• 引脚功能 • 内部模块功能 • 中断 • 其它片上外围功能 • 电气特性（目标值）	• CPU 功能 • 指令集 • 指令扩展

手册使用方法

在阅读本手册前，读者应掌握电子工程、逻辑电路和微控制器等方面的一般知识。

- 如果读者要了解产品功能：
→ 请按**目录**顺序阅读本手册。
- 如何解释寄存器格式：
→ 括号中的位名在 RA78K0R 中被定义为保留字，并且在 CC78K0R 中直接通过 `#pragma sfr` 定义为一个 sfr 变量。
- 如果了解 78K0R 微控制器指令的详细信息：
→ 可参阅 **78K0R 系列指令用户手册(U17792E)**。

规定	数据规则:	数据的高位部分在左边，低位部分在右边
	有效低电平表示法:	xxx (在引脚和信号名称上划一条线)
	注:	文中用 注 标注的相关术语的脚注
	注意事项:	需要特别关注的信息
	备注:	补充信息
	数的表示法:	二进制 ...xxxx 或 xxxxB
		十进制 ...xxxx
		十六进制 ...xxxxH

相关文档 本手册中的相关文档可能包括基本版本。但是并未对初稿版本作特殊标注。

设备文档

文档名称	文档编号
78K0R/KG3-L 用户手册	本手册
78K0R 微控制器指令用户手册	U17792E

开发工具的相关文档 (软件) (用户手册)

文档名称	文档编号
CC78K0R Ver. 2.00 C 编译器。	操作
	语言
RA78K0R Ver 1.20 汇编程序包	操作
	语言
SM+ 系统仿真器	操作
PM+ Ver. 6.30	
ID78K0R-QB Ver. 3.20 集成调试器	操作

开发工具的相关文档 (硬件) (用户手册)

文档名称	文档编号
QB-MINI2 具有编程功能的片上调试仿真器	U18371E
QB-78K0RKX3C 在线仿真器	准备中

与 Flash 存储器编程相关文档

文档名称	文档编号
PG-FP5 Flash 存储器编程器用户手册	U18865E

注意事项 对以上列出的相关文档所做修改恕不另行通知。在设计时请使用每个文档的最新版本。

其他文档

文档名称	文档编号
半导体选择指南 –产品和封装–	X13769X
半导体设备安装手册	注
NEC 半导体设备质量等级	C11531E
NEC 半导体设备可靠性/质量控制系统	C10983E
半导体设备防静电 ESD 保护指南	C11892E

注 见“半导体器件安装手册”网站(<http://www.necel.com/pkg/en/mount/index.html>)。

注意事项 对以上列出的相关文档所做修改恕不另行通知。在设计时请使用每个文档的最新版本。

目录

第 1 章 概述	18
1.1 特性	18
1.2 应用	19
1.3 订购信息	19
1.4 引脚配置(俯视图).....	20
1.5 78K0R/Kx3-L 微控制器列表.....	23
1.6 框图	24
1.7 功能概述	25
第 2 章 引脚功能	27
2.1 引脚功能列表	27
2.2 引脚功能描述	33
2.2.1 P00 至 P06 (端口 0)	33
2.2.2 P10 至 P17 (端口 1)	34
2.2.3 P20 至 P27 (端口 2)	35
2.2.4 P30 至 P31 (端口 3)	35
2.2.5 P40 至 P47 (端口 4)	36
2.2.6 P50 至 P57 (端口 5)	37
2.2.7 P60 至 P67 (端口 6)	37
2.2.8 P70 至 P77 (端口 7)	38
2.2.9 P80 至 P87 (端口 8)	38
2.2.10 P91 (端口 9)	38
2.2.11 P110, P111 (端口 11)	38
2.2.12 P120 至 P124 (端口 12)	39
2.2.13 P130 和 P131 (端口 13).....	39
2.2.14 P140 至 P145 (端口 14)	40
2.2.15 P150 至 P157 (端口 15)	41
2.2.16 AVREF	41
2.2.17 AVSS.....	41
2.2.18 RESET	41
2.2.19 REGC.....	42
2.2.20 VDD, EVDD0, EVDD1.....	42
2.2.21 VSS, EVSS0, EVSS1.....	42
2.2.22 FLMD0	42
2.3 引脚 I/O 电路和未使用引脚的推荐连接方式	43

第 3 章 CPU 架构	48
3.1 存储器空间	48
3.1.1 内部程序存储器空间	52
3.1.2 镜像区域	54
3.1.3 内部数据存储器空间	55
3.1.4 特殊功能寄存器（SFR）区域	56
3.1.5 扩展特殊功能寄存器（第二 SFR：第二 特殊功能寄存器）区域	56
3.1.6 数据存储器寻址	57
3.2 处理器寄存器	59
3.2.1 控制寄存器	59
3.2.2 通用寄存器	61
3.2.3 ES 和 CS 寄存器	63
3.2.4 特殊功能寄存器 (SFR)	64
3.2.5 扩展特殊功能寄存器（第二 SFR：第二特殊功能寄存器）	70
3.3 指令地址寻址	77
3.3.1 相对寻址	77
3.3.2 立即寻址	77
3.3.3 表间接寻址	78
3.3.4 寄存器直接寻址	79
3.4 用于处理数据地址的寻址	80
3.4.1 隐式寻址	80
3.4.2 寄存器寻址	80
3.4.3 直接寻址	81
3.4.4 短直接寻址	82
3.4.5 SFR 寻址	83
3.4.6 寄存器间接寻址	84
3.4.7 基址寻址	85
3.4.8 基址变址寻址	88
3.4.9 堆栈寻址	89
第 4 章 端口功能	90
4.1 端口功能	90
4.2 端口配置	93
4.2.1 端口 0	94
4.2.2 端口 1	99
4.2.3 端口 2	107
4.2.4 端口 3	109
4.2.5 端口 4	110
4.2.6 端口 5	117

4.2.7 端口 6.....	119
4.2.8 端口 7.....	122
4.2.9 端口 8.....	123
4.2.10 端口 9.....	124
4.2.11 端口 11.....	125
4.2.12 端口 12.....	127
4.2.13 端口 13.....	130
4.2.14 端口 14.....	132
4.2.15 端口 15.....	136
4.3 寄存器控制端口功能.....	137
4.4 端口功能操作.....	144
4.4.1 写入 I/O 端口.....	144
4.4.2 从 I/O 端口读取.....	144
4.4.3 I/O 端口的操作.....	144
4.4.4 以不同电势(2.5 V、3 V)连接到外部设备.....	145
4.5 使用复用功能时端口模式寄存器和输出锁存的设置.....	147
4.6 端口寄存器 n (Pn)的 1 位操作指令的注意事项.....	149
第 5 章 时钟发生器.....	150
5.1 时钟发生器的功能.....	150
5.2 时钟发生器的配置.....	151
5.3 控制时钟发生器的寄存器.....	153
5.4 系统时钟振荡器.....	167
5.4.1 X1 振荡器.....	167
5.4.2 XT1 振荡器.....	167
5.4.3 内部高速振荡器 ^注	171
5.4.4 20MHz 内部高速振荡器 ^注	171
5.4.5 内部低速振荡器.....	171
5.4.6 预分频器.....	171
5.5 时钟发生器操作.....	172
5.6 控制时钟.....	178
5.6.1 控制高速系统时钟的示例.....	178
5.6.2 控制内部高速振荡时钟的示例.....	181
5.6.3 控制副系统时钟示例.....	183
5.6.4 内部低速振荡时钟的控制示例.....	185
5.6.5 CPU 时钟状态转换图.....	186
5.6.6 CPU 时钟改变之前的状况与改变之后的处理.....	193
5.6.7 CPU 时钟和主系统时钟切换所需的时间.....	195
5.6.8 时钟振荡停止前的状况.....	196

第 6 章 定时器阵列单元	197
6.1 定时器阵列单元的功能	197
6.1.1 独立操作时每个通道的功能	197
6.1.2 和其它通道组合操作时每个通道的功能	198
6.1.3 LIN 总线支持功能(仅限通道 7).....	198
6.2 定时器阵列单元的配置	199
6.3 控制定时器阵列单元的寄存器	205
6.4 通道输出 (TOmn 引脚) 控制.....	228
6.4.1 TOmn 引脚输出电路配置.....	228
6.4.2 TOmn 引脚输出设置.....	229
6.4.3 通道输出操作的注意事项.....	230
6.4.4 TOmn 位的集中操作.....	233
6.4.5 定时器中断和操作开始时的 TOmn 引脚输出	234
6.5 通道输入 (TImn 引脚) 控制.....	235
6.5.1 TImn 边沿检测电路	235
6.6 定时器阵列单元的基本功能.....	236
6.6.1 独立操作功能和组合操作功能的概述.....	236
6.6.2 组合操作功能的基本规则.....	236
6.6.3 组合操作功能的基本规则的应用范围.....	237
6.7 作为独立通道的定时器阵列单元的操作.....	238
6.7.1 作为间隔定时器/方波输出的操作	238
6.7.2 作为外部事件计数操作	245
6.7.3 作为分频器操作(仅对单元 0 的通道 0).....	248
6.7.4 作为输入脉冲间隔测量的操作	252
6.7.5 用作输入信号高/ 低电平宽度测量的操作	256
6.8 定时器阵列单元的多通道操作	260
6.8.1 作为 PWM 功能操作	260
6.8.2 作为单脉冲输出功能操作.....	267
6.8.3 作为多路 PWM 输出功能的操作	274
第 7 章 实时计数器.....	282
7.1 实时计数器的功能	282
7.2 实时计数器的配置	282
7.3 控制实时计数器的寄存器.....	284
7.4 实时计数器操作	299
7.4.1 启动实时计数器的操作	299
7.4.2 启动操作后切换至 STOP 模式.....	300
7.4.3 读/写实时计数器	301
7.4.4 设置实时计数器报警.....	303

7.4.5 实时计数器的 1 Hz 输出	304
7.4.6 实时计数器的 32.768 kHz 输出	304
7.4.7 实时计数器的 512 Hz, 16.384 kHz 输出	304
7.4.8 实时计数器的钟表错误修正示例	305
第 8 章 看门狗定时器	310
8.1 看门狗定时器的功能	310
8.2 看门狗定时器的配置	311
8.3 控制看门狗定时器的寄存器	312
8.4 看门狗定时器的操作	313
8.4.1 看门狗定时器的控制操作	313
8.4.2 看门狗定时器溢出时间的设置	314
8.4.3 看门狗定时器窗口打开周期的设置	315
8.4.4 看门狗定时器间隔中断的设置	316
第 9 章 时钟输出/蜂鸣器输出控制器	317
9.1 时钟输出/蜂鸣器输出控制器的功能	317
9.2 时钟输出/蜂鸣器输出控制器的配置	318
9.3 控制时钟输出/蜂鸣器输出控制器的寄存器	318
9.4 时钟输出/蜂鸣器输出控制器的操作	320
9.4.1 作为输出引脚的操作	320
第 10 章 A/D 转换器	321
10.1 A/D 转换器的功能	321
10.2 A/D 转换器的配置	322
10.3 A/D 转换器使用的寄存器	324
10.4 A/D 转换器操作	335
10.4.1 A/D 转换器的基本操作	335
10.4.2 输入电压和转换结果	337
10.4.3 A/D 转换器操作模式	338
10.5 A/D 转换器特征表的阅读方法	341
10.6 A/D 转换器的注意事项	343
第 11 章 串行阵列单元	347
11.1 串行阵列单元的功能	348
11.1.1 3 线串行 I/O (CSI00、CSI01、CSI10 和 CSI20)	348
11.1.2 UART (UART0 至 UART3)	349
11.1.3 简易的 I ² C (IIC10, IIC20)	350
11.2 串行阵列单元的配置	351

11.3 控制串行阵列单元的寄存器	356
11.4 操作停止模式	378
11.4.1 由部件停止操作	378
11.4.2 由通道停止操作	379
11.5 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 通信的操作	381
11.5.1 主发送	382
11.5.2 主接收	391
11.5.3 主发送/接收	397
11.5.4 从发送	405
11.5.5 从接收	414
11.5.6 从发送/接收	420
11.5.7 计算传输时钟频率	429
11.6 UART(UART0 至 UART3)通信的操作	431
11.6.1 UART 发送	432
11.6.2 UART 接收	442
11.6.3 LIN 发送	449
11.6.4 LIN 接收	452
11.6.5 计算波特率	457
11.7 简易I ² C (IIC10, IIC20)通信操作	461
11.7.1 发送地址区域	462
11.7.2 数据发送	467
11.7.3 数据接收	470
11.7.4 停止条件的产生	474
11.7.5 计算传输速率	475
11.8 错误的处理程序	478
11.9 寄存器设置和引脚之间的关系	480
第 12 章 串行接口 IICA	487
12.1 串行接口 IICA 的功能	487
12.2 串行接口 IICA 的配置	490
12.3 控制串行接口 IICA 的寄存器	493
12.4 I ² C总线模式功能	506
12.4.1 引脚配置	506
12.4.2 通过 IICWL 和 IICWH 寄存器设置传输时钟	507
12.5 I ² C总线定义和控制方法	508
12.5.1 启动条件	508
12.5.2 地址	509
12.5.3 传输方向指示	509
12.5.4 响应信号 (ACK)	510
12.5.5 停止条件	511

12.5.6 等待	512
12.5.7 取消等待	514
12.5.8 中断请求 (INTIICA)产生时序和等待控制.....	515
12.5.9 地址匹配的检测方法.....	516
12.5.10 误差检测	516
12.5.11 扩展码	516
12.5.12 仲裁	517
12.5.13 唤醒功能	519
12.5.14 通信预约	522
12.5.15 注意事项	526
12.5.16 通信操作	527
12.5.17 产生I ² C中断请求 (INTIICA) 的时序	535
12.6 时序图.....	556
第 13 章 乘法器/除法器	563
13.1 乘法器/除法器的功能.....	563
13.2 乘法器/除法器的配置.....	563
13.3 乘法器/除法器的寄存器	568
13.4 乘法器/除法器的操作.....	569
13.4.1 乘法运算	569
13.4.2 除法操作	570
第 14 章 DMA 控制器.....	572
14.1 DMA 控制器.....	572
14.2 DMA 控制器的配置.....	573
14.3 控制 DMA 控制器的寄存器	576
14.4 DMA 控制器的操作	579
14.4.1 操作过程.....	579
14.4.2 传输模式	580
14.4.3 DMA 传输的终止	580
14.5 DMA 控制器设置的实例.....	581
14.5.1 CSI 连续发送	581
14.5.2 A/D 转换结果的连续捕捉	583
14.5.3 UART 连续接收+ACK 发送.....	585
14.5.4 通过 DWAITn 保持 DMA 传输挂起.....	587
14.5.5 软件强行中止.....	588
14.6 使用 DMA 控制器的注意事项	589
第 15 章 中断功能	591

15.1 中断功能的类型	591
15.2 中断源和配置	591
15.3 控制中断功能的寄存器	597
15.4 中断服务操作	607
15.4.1 可屏蔽的中断响应	607
15.4.2 软件中断请求响应	609
15.4.3 中断嵌套	610
15.4.4 保持中断请求	613
第 16 章 按键中断功能	614
16.1 按键中断的功能	614
16.2 按键中断的配置	614
16.3 控制按键中断的寄存器	615
第 17 章 待机功能	616
17.1 待机功能及配置	616
17.1.1 待机功能	616
17.1.2 控制待机功能的寄存器	617
17.2 待机功能的操作	620
17.2.1 HALT 模式	620
17.2.2 STOP 模式	625
第 18 章 复位功能	630
18.1 确认复位源的寄存器	639
第 19 章 上电清零电路	640
19.1 上电清零电路的功能	640
19.2 上电清零电路的配置	641
19.3 上电清零电路的操作	641
19.4 上电清零电路使用注意事项	644
第 20 章 低电压检测电路	646
20.1 低电压检测电路的功能	646
20.2 低电压检测电路的配置	647
20.3 控制低电压检测电路的寄存器	647
20.4 低电压检测电路的操作	652
20.4.1 当用于复位	653
20.4.2 当作为中断使用时	659
20.5 低电压检测电路的注意事项	665

第 21 章 稳压器	669
21.1 稳压器概述	669
21.2 控制稳压器的寄存器	669
第 22 章 选项字节	671
22.1 选项字节的功能	671
22.1.1 用户选项字节 (000C0H 至 000C2H/ 010C0H 至 010C2H)	671
22.1.2 片上调试选项字节 (000C3H/ 010C3H)	672
22.2 用户选项字节的格式	672
22.3 片上调试选项字节的格式	674
22.4 选项字节的设置	675
第 23 章 FLASH 存储器	676
23.1 用 Flash 存储器编程器写入	676
23.2 编程环境	680
23.3 通信模式	680
23.4 On Board 方式的引脚连接	681
23.4.1 FLMD0 引脚	681
23.4.2 TOOL0 引脚	682
23.4.3 RESET 引脚	682
23.4.4 端口引脚	683
23.4.5 REGC 引脚	683
23.4.6 X1 和 X2 引脚	683
23.4.7 供电电压	683
23.5 控制 Flash 存储器的寄存器	683
23.6 编程方法	684
23.6.1 控制 Flash 存储器	684
23.6.2 Flash 存储器编程模式	684
23.6.3 选择通信模式	685
23.6.4 通信命令	685
23.7 安全设置	687
23.8 通过自编程进行 Flash 存储器编程	689
23.8.1 引导交换功能	691
23.8.2 Flash 保护窗口功能	693
第 24 章 片上调试功能	694
24.1 连接 QB-MINI2 到 78K0R/KG3-L	694
24.2 片上调试安全 ID	695
24.3 用户资源的确保	695

第 25 章 BCD 修正电路	697
25.1 BCD 修正电路功能.....	697
25.2 BCD 修正电路使用的寄存器	697
25.3 BCD 修正电路操作.....	698
第 26 章 指令集.....	700
26.1 操作列表使用规则	701
26.1.1 操作数标识符和指定方法	701
26.1.2 操作栏描述.....	702
26.1.3 标志操作栏的描述	703
26.1.4 PREFIX 指令.....	703
26.2 操作列表.....	704
第 27 章 电气特性（目标值）	721
第 28 章 封装图.....	765
附录 A 开发工具	767
A.1 软件包	770
A.2 语言处理软件	770
A.3 控制软件	771
A.4 Flash 存储器编程工具	771
A.4.1 当使用 Flash 存储器编程器 PG-FP5, FL-PR5	771
A.4.2 当使用具有编程功能的片上调试仿真器 QB-MINI2.....	772
A.5 调试工具 (硬件).....	772
A.5.1 当使用在线仿真器 QB-78K0RKX3C	772
A.5.2 当使用具有编程功能的片上调试仿真器 QB-MINI2.....	773
A.6 调试工具 (软件).....	773

第 1 章 概述

1.1 特性

- 指令最短执行时间可从高速(0.05 μ s: @高速系统时钟操作频率为 20 MHz)改变至超低速(61 μ s: @副系统时钟的操作频率为 32.768 kHz)
- 通用寄存器: 8 位 $\times$ 32 个寄存器(8 位 $\times$ 8 个寄存器 $\times$ 4 个 bank)
- ROM、RAM 容量

产品型号 \ 项目	程序存储器 (ROM)		数据存储器 (RAM)
μ PD78F1013 ^{注 1}	Flash 存储器	96 KB	6 KB
μ PD78F1014 ^{注 1}		128 KB	8 KB ^{注 2}

注 1. 开发中

2. 当使用自编程功能时为 7KB。

- 片上内部高速振荡时钟
 - 20 MHz 内部高速振荡时钟频率: 20 MHz $\pm$ 1 % (目标值)
 - 8 MHz 内部高速振荡时钟: 8 MHz $\pm$ 1 % (目标值)
 - 1 MHz 内部高速振荡时钟: 1 MHz $\pm$ 5 %
- 片上单电源 Flash 存储器(禁止芯片擦除/块擦除/写入功能)
- 自编程(具有引导交换功能/Flash 窗口屏蔽功能)
- 片上调试功能
- 片上上电清零电路 (POC) 和低电压检测电路 (LVI)
- 片上看门狗定时器(用专用内部低速振荡时钟操作可操作)
- 片上乘法器/除法器(16 位 $\times$ 16 位, 32 位 $\div$ 32 位)
- 片上按键中断功能
- 片上时钟输出/蜂鸣器输出控制器
- 片上 BCD 调整
- I/O 端口: 89(N 沟道漏极开路: 4)
- 定时器: 14 个通道
 - 16 位定时器: 12 通道
 - 看门狗定时器: 1 通道
 - 实时计数器: 1 通道
- 串行接口
 - CSI: 2 通道/UART: 1 通道
 - CSI: 单通道/UART: 单通道/简易 I²C: 单通道
 - CSI: 单通道/UART: 单通道/简易 I²C: 单通道
 - UART(支持 LIN 总线): 单通道
 - I²C: 单通道
- 10 位分辨率 A/D 转换器(AV_{REF} = 1.8 至 5.5 V): 16 个通道
- 电源供电电压: V_{DD} = 1.8 至 5.5 V
- 工作环境温度: T_A = -40 至 +85°C

1.2 应用

- 视听设备
- 家用电器
- 工业设备

1.3 订购信息

- Flash 存储器版本

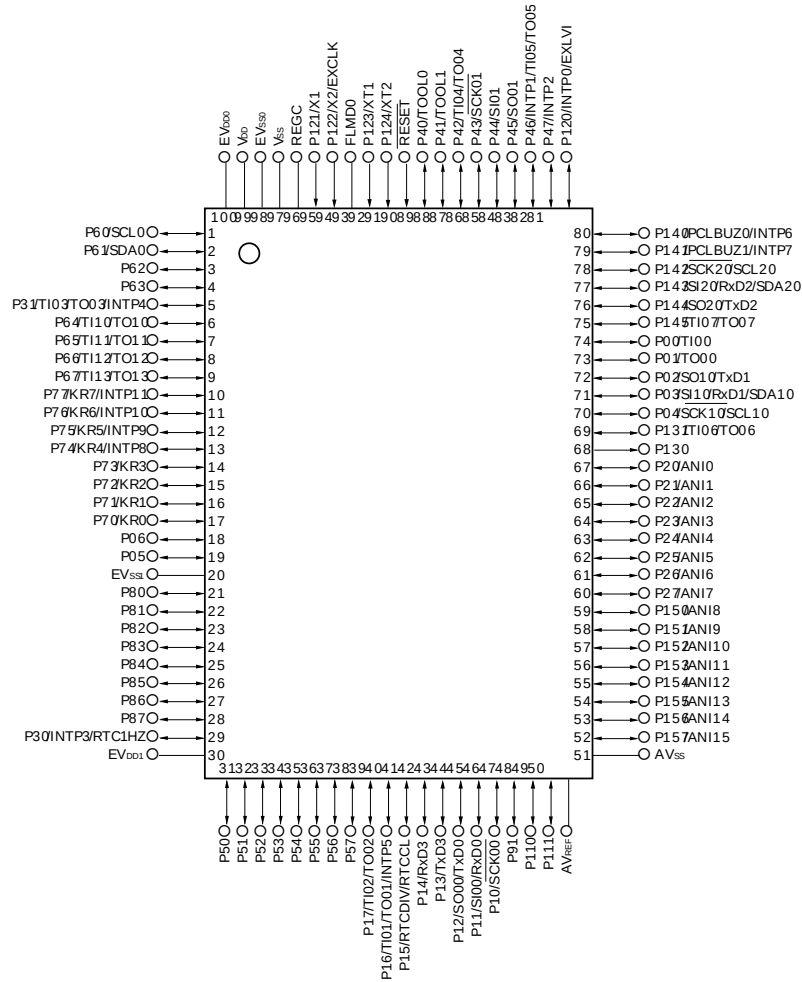
产品型号	封装
μ PD78F1013GC-UEU-AX ^注	100 引脚塑封 LQFP(密脚距)(14 × 14)
μ PD78F1013GF-GAS-AX ^注	100 引脚塑封 LQFP (14 × 20)
μ PD78F1014GC-UEU-AX ^注	100 引脚塑封 LQFP(密脚距)(14 × 14)
μ PD78F1014GF-GAS-AX ^注	100 引脚塑封 LQFP (14 × 20)

注 开发中

注意事项 78K0R/KG3-L 具有片上调试功能，可用于开发和评价。在量产时不要使用产品的片上调试功能，因为 flash 存储器中保证的可重写次数可能会超出，从而不能保证产品的可靠性。对于因使用片上调试功能而产生的问题，NEC 电子不承担任何责任。

1.4 引脚配置(俯视图)

- 100 引脚塑封 LQFP (14 × 20)[※]



注 开发中

注意事项 1. 使 AV_{SS}、EV_{SS0} 和 EV_{SS1} 与 V_{SS} 同电势。

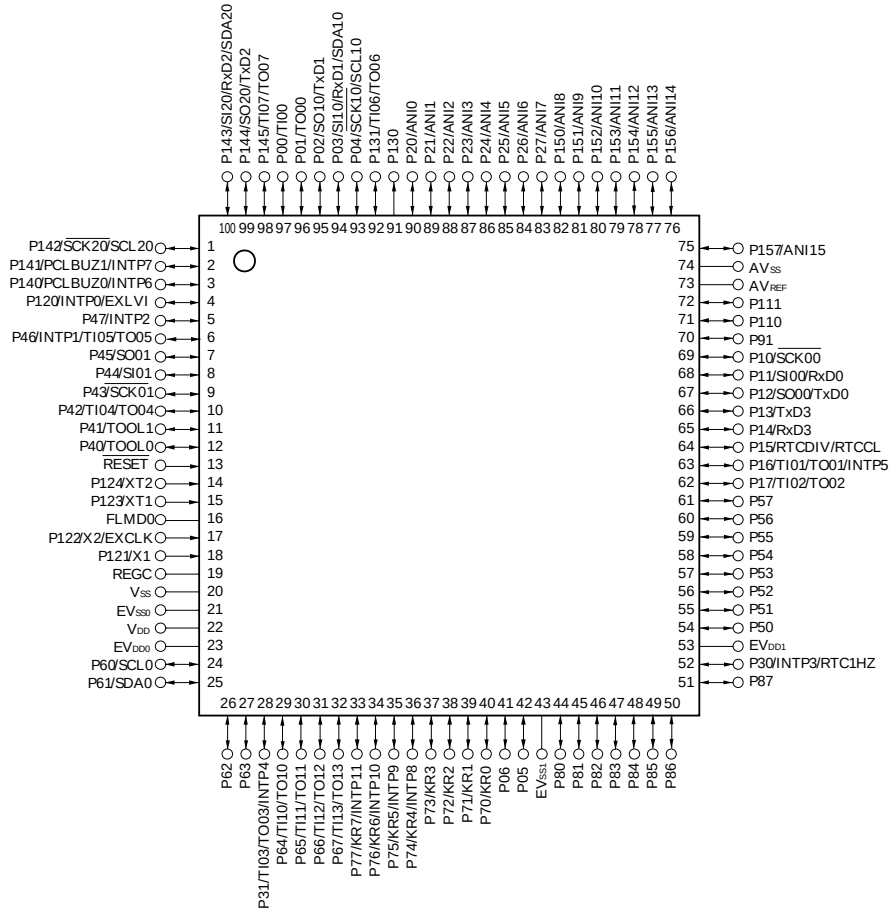
2. 使 EV_{DD0} 和 EV_{DD1} 与 V_{DD} 同电势。

3. 通过一个电容(0.47 至 1 μ F: 目标值)连接 REGC 引脚和 V_{SS}。

4. 由 A/D 端口配置寄存器(ADPC)以 P157/ANI15, ..., P150/ANI8, P27/ANI7, ..., P20/ANI0 的顺序设置 P20/ANI0 至 P27/ANI7 和 P150/ANI8 至 P157/ANI15 为模拟输入。当使用 P20/ANI0 至 P27/ANI7 和 P150/ANI8 至 P157/ANI15 作为模拟输入时, 从 P157/ANI15 开始设计(详情参见 10.3 (6) A/D 端口配置寄存器(ADPC))。

备注 当作为应用使用的微控制器必须降低由微控制器内部产生的噪音时, 推荐对 V_{DD} 和两个 EV_{DD} 引脚单独供电, 并将 V_{SS} 和两个 EV_{SS} 引脚分别接地。

- 100 引脚塑封 LQFP(密脚距)(14 × 14)[※]



注 开发中

- 注意事项**
1. 使 AV_{SS}、EV_{SS0} 和 EV_{SS1} 与 V_{SS} 同电势。
 2. 使 EV_{DD0} 和 EV_{DD1} 与 V_{DD} 同电势。
 3. 通过一个电容(0.47 至 1 μ F: 目标值)连接 REGC 引脚和 V_{SS}。
 4. 由 A/D 端口配置寄存器(ADPC)以 P157/ANI15, ..., P150/ANI8, P27/ANI7, ..., P20/ANI0 的顺序设置 P20/ANI0 至 P27/ANI7 和 P150/ANI8 至 P157/ANI15 为模拟输入。当使用 P20/ANI0 至 P27/ANI7 和 P150/ANI8 至 P157/ANI15 作为模拟输入时, 从 P157/ANI15 开始设计(详情参见 10.3 (6) A/D 端口配置寄存器(ADPC))。

备注 当作为应用使用的微控制器必须降低由微控制器内部产生的噪音时, 推荐对 V_{DD} 和两个 EV_{DD} 引脚单独供电, 并将 V_{SS} 和两个 EV_{SS} 引脚分别接地。

引脚标识

ANI0 至 ANI15:	模拟输入	REGC:	稳压器电容
AVREF:	模拟参考电压	RESET:	复位
AVss:	模拟接地	RTC1HZ:	实时计数器修正时钟 (1 Hz) 输出
EVDD0, EVDD1:	端口供电电源	RTCCL:	实时计数器时钟(32 kHz 原始振荡)
EVSS0, EVSS1:	端口接地		输出
EXCLK:	外部时钟输入(主系统时钟)	RTCDIV:	实时计数器时钟 (32 kHz 分频) 输出
EXLVI:	外部电势输入用于 低电压检测器	RxD0 至 RxD3:	接收数据
FLMD0:	Flash 编程模式	SCK00, SCK01, SCK10, SCK20:	串行时钟输入/输出
INTP0 至 INTP11:	外部中断输入	SCL0, SCL10, SCL20:	串行时钟输入/输出
KR0 至 KR7:	按键返回	SDA0, SDA10, SDA20:	串行数据输入/输出
P00 至 P06:	端口 0	SI00, SI01,	
P10 至 P17:	端口 1	SI10, SI20:	串行数据输入
P20 至 P27:	端口 2	SO00, SO01,	
P30, P31:	端口 3	SO10, SO20:	串行数据输出
P40 至 P47:	端口 4	TI00 至 TI07,	
P50 至 P57:	端口 5	TI10 至 TI13:	定时器输入
P60 至 P67:	端口 6	TO00 至 TO07,	
P70 至 P77:	端口 7	TO10 至 TO13:	定时器输出
P80 至 P87:	端口 8	TOOL0:	对工具的数据输入/输出
P91:	端口 9	TOOL1:	对工具的时钟输出
P110, P111	端口 11	TxD0 至 TxD3:	发送数据
P120 至 P124:	端口 12	VDD:	电源电压
P130, P131:	端口 13	Vss:	地
P140 至 P145:	端口 14	X1,X2:	晶体振荡器 (主系统时钟)
P150 至 P157:	端口 15	XT1,XT2:	晶体振荡器 (副系统时钟)
PCLBUZ0,PCLBUZ1:	可编程时钟输出/蜂鸣器输出		

1.5 78K0R/Kx3-L 微控制器列表

ROM	RAM	78K0R/KC3-L		78K0R/KD3-L	78K0R/KE3-L	78K0R/KF3-L	78K0R/KG3-L
		44 引脚	48 引脚	52 引脚	64 引脚	80 引脚	100 引脚
128 KB	8 KB ^{注1}	—	—	—	—	μPD78F1012 ^{注2}	μPD78F1014 ^{注2}
96 KB	6 KB	—	—	—	—	μPD78F1011 ^{注2}	μPD78F1013 ^{注2}
64 KB	4 KB	—	—	—	—	μPD78F1010 ^{注2}	—
	3 KB ^{注3}	μPD78F1003 ^{注2}	μPD78F1003 ^{注2}	μPD78F1006 ^{注2}	μPD78F1009 ^{注2}	—	—
48 KB	2 KB	μPD78F1002 ^{注2}	μPD78F1002 ^{注2}	μPD78F1005 ^{注2}	μPD78F1008 ^{注2}	—	—
32 KB	1.5 KB	μPD78F1001 ^{注2}	μPD78F1001 ^{注2}	μPD78F1004 ^{注2}	μPD78F1007 ^{注2}	—	—
16 KB	1 KB	μPD78F1000 ^{注2}	—	—	—	—	—

- 注
1. 当使用自编程功能时为 7KB。
 2. 开发中
 3. 当使用自编程功能时为 2KB。

1.7 功能概述

(1/2)

项目		μ PD78F1013 ^{注1}	μ PD78F1014 ^{注1}
内部存储器	Flash 存储器 (支持自编程)	96 KB	128 KB
	RAM	6 KB	8 KB ^{注2}
存储器空间		1 MB	
主系统时钟 (振荡频率)	高速系统时钟	X1 (晶体/陶瓷)振荡, 外部主系统时钟输入(EXCLK) 2 至 20 MHz: $V_{DD} = 2.7$ 至 5.5V, 2 至 5 MHz: $V_{DD} = 1.8$ 至 5.5 V	
	内部高速 振荡时钟	内部振荡 1 MHz $\pm 5\%$, 8 MHz $\pm 1\%$ (目标值): $V_{DD} = 1.8$ 至 5.5 V	
	20 MHz 内部高速振荡 时钟	内部振荡 20 MHz $\pm 1\%$ (目标值): $V_{DD} = 2.7$ 至 5.5 V	
副系统时钟 (振荡频率)		XT1(晶体)振荡 32.768 kHz(典型值): $V_{DD} = 1.8$ 至 5.5 V	
内部低速振荡时钟 (WDT 专用)		内部振荡 30 kHz (典型值): $V_{DD} = 1.8$ 至 5.5 V	
通用寄存器		8 位 $\times$ 32 寄存器 (8 位 $\times$ 8 个寄存器 $\times$ 4 个 bank)	
指令最短执行时间		0.05 μ s (高速系统时钟: $f_{MX} = 20$ MHz 操作)	
		61 μ s (副系统时钟: $f_{SUB} = 32.768$ kHz 操作)	
指令集		• 8 位操作、16 位操作 • 乘法 (8 位 $\times$ 8 位) • 位操作(设置、复位、测试、布尔操作)等	
I/O 端口		总计: 89 CMOS I/O: 80 CMOS 输入: 4 CMOS 输出: 1 N 沟道漏极 I/O (6 V 耐压): 4	
定时器		• 16 位定时器: 12 通道 (单元 0: 8 通道, 单元 1: 4 通道) • 看门狗定时器: 1 通道 • 实时计数器(RTC): 1 通道	
	定时器输出	12 (PWM 输出: 定时器阵列单元 0: 7, 定时器阵列单元 1: 3 ^{注3})	
	RTC 输出	2 • 1 Hz (副系统时钟: $f_{SUB} = 32.768$ kHz) • 512 Hz, 16.384 kHz 或 32.768 kHz (副系统时钟: $f_{SUB} = 32.768$ kHz)	

- 注
- 1.开发中
 - 2.当使用自编程功能时为 7KB。
 - 3.输出的个数根据设置而不同。

(2/2)

项目		μ PD78F1013 ^{註1}	μ PD78F1014 ^{註1}
时钟输出/蜂鸣器输出		2 • 2.44 kHz、4.88 kHz、9.76 kHz、1.25 MHz、2.5 MHz、5 MHz、10 MHz (外围硬件时钟: $f_{MAIN} = 20$ MHz 操作) • 256 Hz、512 Hz、1.024 kHz、2.048 kHz、4.096 kHz、8.192 kHz、16.384 kHz、32.768 kHz (副系统时钟: $f_{SUB} = 32.768$ kHz 操作)	
A/D 转换器		10 位分辨率 × 16 通道($AV_{REF} = 1.8$ 至 5.5 V)	
串行接口		• CSI: 2 通道/UART: 1 通道 • CSI: 1 通道/UART: 1 通道/简易 I²C: 1 通道 • CSI: 1 通道/UART: 1 通道/简易 I²C: 1 通道 • UART 支持 LIN 总线: 1 通道 • I²C 总线: 1 通道	
乘法器/除法器		• 16 位 × 16 位 = 32 位 (乘法) • 32 位 ÷ 32 位 = 32 位 (除法)	
DMA 控制器		2 通道	
向量中断源	内部	33	
	外部	13	
按键中断		通过检测按键输入引脚(KR0 至 KR7)的下降沿发生的按键中断(INTKR)。	
复位		• 使用 RESET 引脚复位 • 使用看门狗定时器内部复位 • 使用上电清零内部复位 • 使用低电压检测电路内部复位 • 非法指令执行引起内部复位^{註2}	
上电清零电路		• 启动复位: 1.61 ± 0.09 V • 关机复位: 1.59 ± 0.09 V	
低电压检测器		1.91 V 至 4.22 V (16 阶段)	
片上调试功能		提供	
供电电压		$V_{DD} = 1.8$ 至 5.5 V	
工作环境温度		$T_A = -40$ 至 $+85^{\circ}\text{C}$	
封装		100 引脚塑封 LQFP (14 × 20)(引脚间距: 0.65 mm) 100 引脚塑封 LQFP (密脚距)(14×14)(引脚间距: 0.5 mm)	

- 注
1. 开发中
 2. 当执行 FFH 中的指令码时, 产生非法操作。
由非法指令执行引起的复位, 不能通过在线仿真器或片上调试仿真器终止。

第 2 章 引脚功能

2.1 引脚功能列表

有 4 种类型的引脚 I/O 缓冲器供给电源：AV_{REF}、EV_{DD0}、EV_{DD1} 和 V_{DD}。下表显示了这些供电电源与引脚之间的关系。

表 2-1. 引脚 I/O 缓冲区供电电源

供电电源	对应的引脚
AV _{REF}	P20 至 P27, P150 至 P157
EV _{DD0} , EV _{DD1}	• 端口引脚，除 P20 至 P27, P121 至 P124, 以及 P150 至 P157 以外 • $\overline{\text{RESET}}$ 和 FLMD0 引脚
V _{DD}	• P121 至 P124 • 除端口引脚以外的引脚（不包括 $\overline{\text{RESET}}$ 和 FLMD0 引脚）

(1) 端口功能 (1/2)

功能名称	I/O	功能	复位后	复用功能
P00	I/O	端口 0。 7 位 I/O 端口。 P03 和 P04 输入可设置成 TTL 输入缓冲器。 P02 至 P04 的输出可设置成 N 沟道漏极开路输出 (V_{DD} 耐压)。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片内上拉电阻的使用。	输入端口	TI00
P01				TO00
P02				SO10/TxD1
P03				SI10/RxD1/SDA10
P04				SCK10/SCL10
P05, P06				—
P10	I/O	端口 1。 8 位 I/O 端口。 P10 和 P11 输入可设置成 TTL 输入缓冲器。 P10 和 P12 的输出可设置到 N 沟道漏极开路输出 (V_{DD} 耐压)。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	SCK00
P11				SI00/RxD0
P12				SO00/TxD0
P13				TxD3
P14				RxD3
P15				RTCDIV/RTCCL
P16				TI01/TO01/INTP5
P17				TI02/TO02
P20 至 P27	I/O	端口 2。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。	数字输入端口	ANI0 至 ANI7
P30	I/O	端口 3。 2 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	RTC1HZ/INTP3
P31				TI03/TO03/INTP4
P40 ^注	I/O	端口 4。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	TOOL0
P41				TOOL1
P42				TI04/TO04
P43				SCK01
P44				SI01
P45				SO01
P46				INTP1/TI05/TO05
P47				INTP2
P50 至 P57	I/O	端口 5。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	—

注 如果通过选项字节使能片上调试，则确保外部上拉 P40/TOOL0 引脚（参见 2.2.5 P40 至 P47 (端口 4) 中的注意事项）。

(1) 端口功能 (2/2)

功能名称	I/O	功能	复位后	复用功能
P60	I/O	端口 6。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。 仅对 P64 至 P67，通过软件设置，可以定义片上上拉电阻的使用。	输入端口	SCL0
P61				SDA0
P62, P63				—
P64 至 P67				TI10/TO10 至 TI13/TO13
P70 至 P73	I/O	端口 7。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	KR0 至 KR3
P74 至 P77				KR4/INTP8 至 KR7/INTP11
P80 至 P87	I/O	端口 8。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	—
P91	I/O	端口 9。 1 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	—
P110, P111	I/O	端口 11。 2 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片内上拉电阻的使用。	输入端口	—
P120	I/O	端口 12 1 位 I/O 端口和 4 位输入端口。 仅对 P120，通过软件设置，可以定义片上上拉电阻的使用。	输入端口	INTP0/EXLVI
P121	输入			X1
P122				X2/EXCLK
P123				XT1
P124				XT2
P130	输出	端口 13。 1 位输出端口和 1 位 I/O 端口。	输出端口	—
P131	I/O	仅对 P131，通过软件设置，可以定义片上上拉电阻的使用。	输入端口	TI06/TO06
P140	I/O	端口 14。 6 位 I/O 端口。 P142 和 P143 输入可设置成 TTL 输入缓冲器。 P142 至 P144 的输出可设置成 N 沟道漏极开路输出 (V _{DD} 耐压)。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	PCLBUZ0/INTP6
P141				PCLBUZ1/INTP7
P142				SCK20/SCL20
P143				SI20/RxD2/SDA20
P144				SO20/TxD2
P145				TI07/TO07
P150 至 P157	I/O	端口 15。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。	数字输入端 口	ANI8 至 ANI15

(2) 非端口功能 (1/3)

功能名称	I/O	功能	复位后	复用功能
ANI0 至 ANI7	输入	A/D 转换器模拟输入	数字输入端口	P20 至 P27
ANI8 至 ANI15				P150 至 P157
EXLVI	输入	为外部低电压检测提供的电势输入	输入端口	P120/INTP0
INTP0	输入	可指定有效沿（上升沿、下降沿、或双沿）时的外部中断请求输入	输入端口	P120/EXLVI
INTP1				P46/TI05/TO05
INTP2				P47
INTP3				P30/RTC1HZ
INTP4				P31/TI03/TO03
INTP5				P16/TI01/TO01
INTP6				P140/PCLBUZ0
INTP7				P141/PCLBUZ1
INTP8 至 INTP11				P74/KR4 至 P77/KR7
KR0 至 KR3	输入	按按键中断输入	输入端口	P70 至 P73
KR4 至 KR7				P74/INTP8 至 P77/INTP11
PCLBUZ0	输出	时钟输出/蜂鸣器输出	输入端口	P140/INTP6
PCLBUZ1				P141/INTP7
REGC	—	为内部操作，连接稳压器输出（2.4 V）稳定电容。通过电容器(0.47 至 1 μ F：目标值)连接到 V _{SS} 。	—	—
RTCDIV	输出	实时计数器时钟 (32 kHz 分频) 输出	输入端口	P15/RTCCL
RTCCL	输出	实时计数器时钟(32 kHz 原始振荡) 输出	输入端口	P15/RTCDIV
RTC1HZ	输出	实时计数器修正时钟（1 Hz）输出	输入端口	P30/INTP3
RESET	输入	系统复位输入	—	—
RxD0	输入	串行数据输入到 UART0	输入端口	P11/SI00
RxD1	输入	串行数据输入到 UART1	输入端口	P03/SI10/SDA10
RxD2	输入	串行数据输入到 UART2	输入端口	P143/SI20/SDA20
RxD3	输入	串行数据输入到 UART3	输入端口	P14
SCK00	I/O	CSI00、CSI01、CSI10 和 CSI20 的时钟输入/输出	输入端口	P10
SCK01				P43
SCK10				P04/SCL10
SCK20				P142/SCL20
SCL0	I/O	I ² C 的时钟输入/输出	输入端口	P60
SCL10	I/O	简易 I ² C 的时钟输入/输出	输入端口	P04/SCK10
SCL20	I/O	简易 I ² C 的时钟输入/输出	输入端口	P142/SCK20

(2) 非端口功能 (2/3)

功能名称	I/O	功能	复位后	复用功能
SDA0	I/O	I ² C 的串行数据 I/O	输入端口	P61
SDA10	I/O	简易 I ² C 的串行数据 I/O	输入端口	P03/SI10/RxD1
SDA20	I/O	简易 I ² C 的串行数据 I/O	输入端口	P143/SI20/RxD2
SI00	输入	串行数据输入到 CSI00、CSI01、CSI10 和 CSI20	输入端口	P11/RxD0
SI01				P44
SI10				P03/RxD1/SDA10
SI20				P143/RxD2/SDA20
SO00	输出	从 CSI00、CSI01、CSI10 和 CSI20 的串行数据输出	输入端口	P12/TxD0
SO01				P45
SO10				P02/TxD1
SO20				P144/TxD2
TI00	输入	外部计数时钟输入到 16 位定时器 00	输入端口	P00
TI01		外部计数时钟输入到 16 位定时器 01		P16/TO01/INTP5
TI02		外部计数时钟输入到 16 位定时器 02		P17/TO02
TI03		外部计数时钟输入到 16 位定时器 03		P31/TO03/INTP4
TI04		外部计数时钟输入到 16 位定时器 04		P42/TO04
TI05		外部计数时钟输入到 16 位定时器 05		P46/INTP1/TO05
TI06		外部计数时钟输入到 16 位定时器 06		P131/TO06
TI07		外部计数时钟输入到 16 位定时器 07		P145/TO07
TI10		外部计数时钟输入到 16 位定时器 10		P64/TO10
TI11		外部计数时钟输入到 16 位定时器 11		P65/TO11
TI12		外部计数时钟输入到 16 位定时器 12		P66/TO12
TI13		外部计数时钟输入到 16 位定时器 13		P67/TO13
TO00	输出	16 位定时器 00 输出	输入端口	P01
TO01		16 位定时器 01 输出		P16/TI01/INTP5
TO02		16 位定时器 02 输出		P17/TI02
TO03		16 位定时器 03 输出		P31/TI03/INTP4
TO04		16 位定时器 04 输出		P42/TI04
TO05		16 位定时器 05 输出		P46/INTP1/TI05
TO06		16 位定时器 06 输出		P131/TI06
TO07		16 位定时器 07 输出		P145/TI07
TO10		16 位定时器 10 输出		P64/TI10
TO11		16 位定时器 11 输出		P65/TI11
TO12		16 位定时器 12 输出		P66/TI12
TO13		16 位定时器 13 输出		P67/TI13
TxD0	输出	从 UART0 的串行数据输出	输入端口	P12/SO00
TxD1		从 UART1 的串行数据输出		P02/SO10
TxD2		从 UART2 输出串行数据		P144/SO20
TxD3		从 UART3 输出串行数据		P13
X1	—	主系统时钟的振荡器连接	输入端口	P121
X2	—		输入端口	P122/EXCLK

(2) 非端口功能 (3/3)

功能名称	I/O	功能	复位后	复用功能
EXCLK	输入	主系统时钟的外部时钟输入	输入端口	P122/X2
XT1	—	副系统时钟的振荡器连接	输入端口	P123
XT2	—		输入端口	P124
V _{DD}	—	正向供电（P121 至 P124 和除了端口（除 RESET 引脚和 FLMD0 引脚以外））	—	—
EV _{DD0} , EV _{DD1}	—	端口的正向供电（除 P20 至 P27、P121 至 P124, P150 至 P157 以外）以及 RESET 引脚和 FLMD0 引脚	—	—
AV _{REF}	—	- A/D 转换器参考电压输入 - P20 至 P27, P150 至 P157 的正向供电和 A / D 转换器	—	—
V _{SS}	—	地电位（P121 至 P124 和除了端口（除 RESET 引脚和 FLMD0 引脚以外））	—	—
EV _{SS0} , EV _{SS1}	—	端口的地电位（除 P20 至 P27、P121 至 P124 和 P150 至 P157 以外）以及 RESET 引脚和 FLMD0 引脚	—	—
AV _{SS}	—	A / D 转换器、P20 至 P27 以及 P150 至 P157 的地电位。使用与 EV _{SS0} 、EV _{SS1} 和 V _{SS} 同电势的引脚。	—	—
FLMD0	—	Flash 存储器编程模式设置	—	—
TOOL0	I/O	Flash 存储器编程器/调试器的数据 I/O	输入端口	P40
TOOL1	输出	调试器的时钟输出	输入端口	P41

2.2 引脚功能描述

2.2.1 P00 至 P06 (端口 0)

P00 至 P06 作为 7 位 I/O 端口的功能。这些引脚也具有定时器 I/O、串行接口数据 I/O 和时钟 I/O 的功能。

使用端口输入模式寄存器 0 (PIM0)，可以通过一个常规输入缓冲器或 TTL 输入缓冲器以位单元指定 P03 和 P04 引脚的输入。

使用端口输出模式寄存器 0 (POM0)，可以以位单元指定 P02 至 P04 引脚输出为普通 CMOS 输出或 N 沟道漏极开路输出(V_{DD} 耐压)。

可以指定下列操作模式为位单元。

(1) 端口模式

P00 至 P06 作为 7 位 I/O 端口的功能。通过端口模式寄存器 0 (PM0)，P00 至 P06 可以以位单元设置为输入或输出端口。由上拉电阻选择寄存器 0 (PU0)，指定使用片上上拉电阻。

(2) 控制模式

P00 至 P04 也具有定时器 I/O、串行接口数据 I/O 和时钟 I/O 的功能。

(a) TI00

这是用于输入外部计数时钟/捕捉触发到 16 位定时器 00 的引脚。

(b) TO00

这是 16 位定时器 00 的定时器输出引脚。

(c) SI10

这是串行接口 CSI10 的串行数据输入引脚。

(d) SO10

这是串行接口 CSI10 的串行数据输出引脚。

(e) $\overline{\text{SCK10}}$

这是串行接口 CSI10 的串行时钟 I/O 引脚。

(f) TxD1

这是串行接口 UART1 的串行数据输出引脚。

(g) RxD1

这是串行接口 UART1 的串行数据输入引脚。

(h) SDA10

这是简易 I²C 串行接口的串行数据 I/O 引脚。

(i) SCL10

这是简易 I²C 串行接口的串行时钟 I/O 引脚。

注意事项 使用 P02/SO10/TxD1 和 P04/ $\overline{\text{SCK10}}$ /SCL10 作为通用端口，设置串行通信操作设置寄存器 02(SCR02)到默认状态(0087H)。此外，清除端口输出模式寄存器 0(POM0)为 00H。

2.2.2 P10 至 P17（端口 1）

P10 至 P17 作为 8 位 I/O 端口的功能。这些引脚也有作为外部中断请求输入、串行接口数据 I/O、时钟 I/O、定时器 I/O 和实时计数器时钟功能。

使用端口输入模式寄存器 1（PIM1），可以通过一个常规输入缓冲器或 TTL 输入缓冲器以位单元指定 P10 和 P11 引脚的输入。

使用端口输出模式寄存器 1（POM1），可以以位单元指定 P10 至 P12 引脚输出为普通 CMOS 输出或 N 沟道漏极开路输出（ V_{DD} 耐压）。

可以指定下列操作模式为位单元。

(1) 端口模式

P10 至 P17 作为 8 位 I/O 端口的功能。通过端口模式寄存器 1（PM1），P10 至 P17 可以位单元设置为输入或输出端口。由上拉电阻选择寄存器 1（PU1）指定使用片上上拉电阻。

(2) 控制模式

P10 至 P17 也有作为外部中断请求输入、串行接口数据 I/O、时钟 I/O、定时器 I/O 和实时计数器时钟功能。

(a) SI00

这是串行接口 CSI00 的串行数据输入引脚。

(b) SO00

这是串行接口 CSI00 的串行数据输出引脚。

(c) $\overline{SCK00}$

这是串行接口 CSI00 的串行时钟 I/O 引脚。

(d) RxD0, RxD3

这些是串行接口 UART0 和 UART3 的串行数据输入引脚。

(e) TxD0, TxD3

这些是串行接口 UART0 和 UART3 的串行数据输出引脚。

(f) TI01, TI02

这些是用于输入一个外部计数时钟/捕捉触发到 16 位定时器 01 和 02 的引脚。

(g) TO01, TO02

这些是 16 位定时器 01 至 02 的定时器输出引脚。

(h) INTP5

可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

(i) RTCDIV

这是实时计数器时钟（32 kHz，分频）输出引脚。

(j) RTCCL

这是实时计数器时钟（32 kHz，原始振荡）输出引脚。

- 注意事项**
1. 使用 **P10/SCK00** 和 **P12/SO00/TxD0** 作为通用端口，设置串行通信操作设置寄存器 **00(SCR00)**到默认状态(**0087H**)。
 2. 不要同时使能输出 **RTCCL** 和 **RTCDIV**。

2.2.3 P20 至 P27（端口 2）

P20 至 P27 作为 8 位 I/O 端口的功能。这些引脚也具有作为 A/D 转换器模拟输入的功能。
可以指定下列操作模式为位单元。

(1) 端口模式

P20 至 P27 作为 8 位 I/O 端口的功能。通过端口模式寄存器 2 (PM2)，P20 至 P27 可以以位单元设置为输入或输出端口。

(2) 控制模式

P20 至 P27 作为 A/D 转换器模拟输入引脚（ANI0 至 ANI7）的功能。当这些引脚作为模拟输入引脚使用时，参见 **10.6 (5) ANI0/P20 至 ANI7/P27 和 ANI8/P150 至 ANI15/P157**。

注意事项 在复位释放后，**ANI0/P20 至 ANI7/P27** 设置为数字输入（通用端口）模式。

2.2.4 P30 至 P31（端口 3）

P30 和 P31 功能作为 2 位 I/O 端口。这些引脚具有外部中断请求输入，定时器 I / O 和实时计数器校正时钟输出功能。
可以指定下列操作模式为位单元。

(1) 端口模式

P30 和 P31 功能作为 2 位 I/O 端口。利用端口模式寄存器 3 (PM3)，P30 和 P31 可以以位单元指定为输入或输出端口。由上拉电阻选择寄存器 3 (PU3) 指定使用片上上拉电阻。

(2) 控制模式

P30 和 P31 具有外部中断请求输入，定时器 I / O 和实时计数器校正时钟输出功能。

(a) INTP3, INTP4

可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

(b) TI03

这是用于输入外部计数时钟/捕捉触发到 16 位定时器 03 的引脚。

(c) TO03

这是 16 位定时器 03 的定时器输出引脚。

(d) RTC1HZ

这是实时计数器校正时钟（1 Hz）输出引脚。

2.2.5 P40 至 P47（端口 4）

P40 至 P47 作为 8 位 I/O 端口的功能。这些引脚也有作为外部中断请求输入，串行接口数据 I/O，时钟 I/O，flash 编程器/调试器的数据 I/O，时钟输出，以及定时器 I/O。

可以指定下列操作模式为位单元。

(1) 端口模式

P40 至 P47 作为 8 位 I/O 端口的功能。通过端口模式寄存器 4（PM4），P40 至 P47 可以以位单元设置为输入或输出端口。由上拉电阻选择寄存器 4（PU4）指定使用片上上拉电阻。

当使能片上调试（通过选项字节）时，确保连接一个外部上拉电阻到 P40。

(2) 控制模式

P40 至 P47 有作为串行接口数据 I/O，时钟 I/O，外部中断请求输入，flash 编程器/调试器的数据 I/O，时钟输出，以及定时器 I/O。

(a) INTP1, INTP2

可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

(b) TOOL0

这是 Flash 存储器编程器/调试器的数据 I/O 引脚。

当使能片上调试时，确认此外部引脚为上拉状态(下拉禁止)。

(c) TOOL1

这是调试器的时钟输出引脚。

当使用片上调试功能时，通过调试器模式设置，P41/TOOL1 引脚可作如下用途。

1 线模式：可用作端口（P41）。

2 线模式：可用作 TOOL1 引脚，不能作为端口（P41）使用。

(d) TI04, TI05

这些是用于输入一个外部计数时钟/捕捉触发到 16 位定时器 04 和 05 的引脚。

(e) TO04, TO05

这些是 16 位定时器 04 至 05 的定时器输出引脚。

(f) $\overline{\text{SCK01}}$

这是串行接口 CSI01 的串行时钟 I/O 引脚。

(g) SI01

这是串行接口 CSI01 的串行数据输入引脚。

(h) SO01

这是串行接口 CSI01 的串行数据输出引脚。

注意事项 P40/TOOL0 引脚功能与下面(a)到(c)中描述的不同。

在(b)或(c)的情况中，确保指定的连接。

- (a) 在普通操作模式下，通过选项字节(000C3H)禁止片上调试(OCDENSET = 0)时
=> 此引脚用作端口引脚(P40)。
- (b) 在普通操作模式下，通过选项字节(000C3H)使能片上调试(OCDENSET = 1)时
=> 通过一个外部电阻将该引脚连接到 EV_{DD0} 或 EV_{DD1}，并且在复位释放前始终输入高电平到该引脚。
- (c) 当使用片上调试功能，或在 Flash 存储器编程器写入模式时
=> 此引脚作为 TOOL0 使用。
直接连接该引脚到片上调试仿真器或 Flash 存储器编程器上，或者通过一个外部电阻把它连接到 EV_{DD0} 或 EV_{DD1} 上使得该引脚上拉。

2.2.6 P50 至 P57 (端口 5)

P50 至 P57 具有 8 位 I/O 端口作用。

通过端口模式寄存器 5(PM5)，P50 至 P57 可以以位单元设置为输入或输出端口。由上拉电阻选择寄存器 5(PU5)指定使用片上上拉电阻。

2.2.7 P60 至 P67 (端口 6)

P60 至 P67 具有 8 位 I/O 端口作用。这些引脚也具有串行接口数据 I/O、时钟 I/O 和定时器 I/O 的功能。

可以指定下列操作模式为位单元。

(1) 端口模式

P60 至 P67 具有 8 位 I/O 端口作用。通过端口模式寄存器 6(PM6)，P60 至 P67 可以以位单元设置为输入端口或输出端口。只对 P64 至 P67，由上拉电阻选择寄存器 6(PU6)，指定使用片上上拉电阻。

P60 至 P63 的输出是 N 沟道漏极开路输出 (6 V 耐压)。

(2) 控制模式

P60、P61 和 P64 至 P67 也具有串行接口数据 I/O、时钟 I/O 和定时器 I/O 的功能。

(a) SDA0

这是串行接口 IICA 的一个串行数据 I/O 引脚。

(b) SCL0

这是串行接口 IICA 的一个串行时钟 I/O 引脚。

(c) TI10 至 TI13

这些是用于输入一个外部计数时钟/捕捉触发到 16 位定时器 10 至 13 的引脚。

(d) TO10 至 TO13

这些是 16 位定时器 10 至 13 的定时器输出引脚。

2.2.8 P70 至 P77 (端口 7)

P70 至 P77 是 8 位 I/O 端口。这些引脚也具有按键中断输入和外部中断请求输入的功能。

可以指定下列操作模式为位单元。

(1) 端口模式

P70 至 P77 是 8 位 I/O 端口。通过端口模式寄存器 7(PM7)，P70 至 P77 可以以位单元设置为输入或输出端口。由上拉电阻选择寄存器 7(PU7)指定使用片上上拉电阻。

(2) 控制模式

P70 至 P77 也具有按键中断输入和外部中断请求输入功能。

(a) KR0 至 KR7

这些是按键中断输入引脚。

(b) INTP8 至 INTP11

可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

2.2.9 P80 至 P87 (端口 8)

P80 至 P87 功能作为 8 位 I/O 端口。

通过端口模式寄存器 8(PM8)，P80 至 P87 可以位单元设置为输入或输出端口。由上拉电阻选择寄存器 8 (PU8)，指定使用片上拉电阻。

2.2.10 P91 (端口 9)

P91 作为 1 位 I/O 端口功能。

P91 可通过端口模式寄存器 9(PM9)按位设置为输入或输出端口。由上拉电阻选择寄存器 9(PU9)指定使用片上拉电阻。

2.2.11 P110, P111 (端口 11)

P110 和 P111 功能作为 2 位 I/O 端口。

通过端口模式寄存器 11 (PM11)，P110 和 P111 可以位单元设置为输入或输出端口。由上拉电阻选项寄存器 11(PU11)，指定使用片上拉电阻。

2.2.12 P120 至 P124 (端口 12)

P120 作为 1 位 I/O 端口功能。P121 至 P124 作为 4 位输入端口功能。这些引脚也具有作为外部中断请求输入、外部低电压检测的电势输入、用于主系统时钟的振荡器连接、用于副系统时钟的振荡器连接和用于主系统时钟的外部时钟输入功能。

可以指定下列操作模式为位单元。

(1) 端口模式

P120 作为 1 位 I/O 端口功能。通过端口模式寄存器 12(PM12)，P120 可以以位单元设置为输入或输出端口。由上拉电阻选项寄存器 12(PU12)，指定使用片内上拉电阻。

P121 至 P124 作为 4 位输入端口功能。

(2) 控制模式

P120 至 P124 作为外部中断请求输入、外部低电压检测的电势输入、用于主系统时钟的连接振荡器、用于副系统时钟的连接振荡器和用于主系统时钟的外部时钟输入功能。

(a) INTP0

可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

(b) EXLVI

这是外部低电压检测的电压输入引脚。

(c) X1, X2

这些引脚用于主系统时钟振荡器连接。

(d) EXCLK

这是用于主系统时钟的外部时钟输入引脚。

(e) XT1, XT2

这些引脚用于副系统时钟振荡器连接。

2.2.13 P130 和 P131 (端口 13)

P130 具有 1 位输出端口作用。P131 作为 1 位 I/O 端口功能。这些引脚也作为定时器 I/O 的功能。

备注 复位设备时，P130 输出低电平。因此在设备复位之前通过 P130 输出高电平，P130 的输出信号能用作 CPU 伪复位信号。(参阅 4.2.13 端口 13 图的备注)。

(1) 端口模式

P130 具有 1 位输出端口作用。

P131 作为 1 位 I/O 端口功能。通过端口模式寄存器 13(PM13)，P131 可以以位单元设置为输入或输出端口。由上拉电阻选项寄存器 13(PU13)，指定使用片上上拉电阻。

(2) 控制模式

P131 具有定时器 I/O 作用。

(a) TI06

这是用于输入外部计数时钟/捕捉触发到 16 位定时器 06 的引脚。

(b) TO06

这是 16 位定时器 06 的定时器输出引脚。

2.2.14 P140 至 P145 (端口 14)

P140 至 P145 作为 6 位 I/O 端口功能。这些引脚也有定时器 I/O，外部中断请求输入，时钟/蜂鸣器输出，串行接口数据 I/O 和时钟 I/O 功能。

通过端口输入模式寄存器 14(PIM14)，可以以位单元指定 P142 和 P143 引脚通过普通缓冲器或 TTL 输入缓冲器的输入。

使用端口输出模式寄存器 14 (POM14)，可以以位单元指定 P142 至 P144 引脚输出为普通 CMOS 输出或 N 沟道漏极开路输出(V_{DD} 耐压)。

可以指定下列操作模式为位单元。

(1) 端口模式

P140 至 P145 作为 6 位 I/O 端口功能。通过端口模式寄存器 14(PM14)，P140 至 P145 可以以位单元设置为输入或输出端口。由上拉电阻选项寄存器 14(PU14)，指定使用片上上拉电阻。

(2) 控制模式

P140 至 P145 也有定时器 I/O，外部中断请求输入，时钟/蜂鸣器输出，串行接口数据 I/O 和时钟 I/O 功能。

(a) INTP6, INTP7

可定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），用于外部中断请求输入。

(b) PCLBUZ0, PCLBUZ1

这些是时钟/蜂鸣器输出引脚。

(c) TI07

这是用于输入外部计数时钟/捕捉触发到 16 位定时器 07 的引脚。

(d) TO07

这是 16 位定时器 07 的定时器输出引脚。

(e) SI20

这是串行接口 CSI20 的串行数据输入引脚。

(f) SO20

这是串行接口 CSI20 的串行数据输出引脚。

(g) $\overline{\text{SCK20}}$

这是串行接口 CSI20 的串行时钟 I/O 引脚。

(h) TxD2

这是串行接口 UART2 的串行数据输出引脚。

(i) RxD2

这是串行接口 UART2 的串行数据输出引脚。

(j) SDA20

这是简易 I²C 串行接口的串行数据 I/O 引脚。

(k) SCL20

这是简易 I²C 串行接口的串行时钟 I/O 引脚。

2.2.15 P150 至 P157 (端口 15)

P150 至 P157 具有 8 位 I/O 端口作用。这些引脚也具有作为 A/D 转换器模拟输入的功能。可以指定下列操作模式为位单元。

(1) 端口模式

P150 至 P157 具有 8 位 I/O 端口作用。通过端口模式寄存器 15(PM15)，P150 至 P157 可以以位单元设置为输入或输出端口。

(2) 控制模式

P150 至 P157 作为 A/D 转换器模拟输入引脚(ANI8 至 ANI15)的功能。当这些引脚作为模拟输入引脚使用时，参见 10.6 (5) ANI0/P20 至 ANI7/P27 和 ANI8/P150 至 ANI15/P157。

注意事项 复位释放后，将 ANI8/P150 至 ANI15/P157 设置到数字输入（通用端口）模式。

2.2.16 AV_{REF}

这是 A/D 转换器参考电压输入引脚，也是 P20 至 P27、P150 至 P157 和 A/D 转换器的正向电源供电引脚。

当端口 2 和端口 15 的所有引脚用作模拟端口引脚时，要使 AV_{REF} 电压在 $1.8\text{ V} \leq \text{AV}_{\text{REF}} \leq \text{V}_{\text{DD}}$ 。当端口 2 和 15 的一个或多个引脚作为数字端口引脚使用时，或当 A/D 转换器未使用时，确保将 AV_{REF} 与 EV_{DD0}、EV_{DD1} 或 V_{DD} 设为同电势。

2.2.17 AV_{SS}

这是 A / D 转换器、P20 至 P27、以及 P150 至 P157 的地电位引脚。即使当 A/D 转换器未使用时，始终使该引脚与 EV_{SS0}、EV_{SS1}、和 V_{SS} 同电势。

2.2.18 RESET

这是低有效系统复位输入引脚。

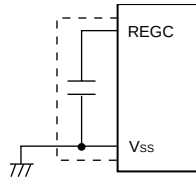
当外部复位引脚不使用时，直接连接此引脚或通过一个电阻连接到 EV_{DD0} 或 EV_{DD1}。

当使用外部复位引脚时，设计基于 V_{DD} 电路。

2.2.19 REGC

这是用于连接内部操作的稳压器输出（2.4V）稳定电容的引脚。通过一个电容（0.47 至 1 μ F：目标值）连接该引脚到 V_{SS}。

同时，使用性能良好的电容，它是用来稳定内部电压的。



注意事项 上图中虚线部分，请使用尽可能短的配线连接。

2.2.20 V_{DD}, EV_{DD0}, EV_{DD1}

V_{DD} 是 P121 至 P124 和除端口（非 RESET 引脚和 FLMD0 引脚）以外引脚的正向供电电源引脚。

EV_{DD0} 和 EV_{DD1} 是端口的正向供电引脚（除 P20 至 P27、P121 至 P124 以及 P150 至 P157 以外）以及 RESET 引脚和 FLMD0 引脚。

2.2.21 V_{SS}, EV_{SS0}, EV_{SS1}

V_{SS} 是 P121 至 P124 和除端口（非 RESET 引脚和 FLMD0 引脚）以外引脚的地电位引脚。

EV_{SS0} 和 EV_{SS1} 是端口的地电势引脚（除 P20 至 P27、P121 至 P124 以及 P150 至 P157 以外）以及 RESET 引脚和 FLMD0 引脚。

2.2.22 FLMD0

此引脚用于设置 Flash 存储器编程模式。

请执行下列操作过程中的一个。

(a) 在普通操作模式下

普通操作期间，建议此引脚开路。

复位释放前 FLMD0 引脚必须保持 V_{SS} 电位，但是不必由外部进行下拉，因为复位时会内部下拉。但是，必须保持选择下拉（如 FLMDPUP = “0”，默认值），通过使用背景事件控制寄存器（BECTL）中的位 7（FLMDPUP）（见 23.5 （1）背景事件控制寄存器）。要处于外部下拉状态，需要使用 200k Ω 或更小值的电阻。

自编程功能和使用编程器对 Flash 存储器进行重写的功能可以通过硬件来禁止，将该引脚直接连接到 V_{SS} 引脚即可。

(b) 在自编程模式下

使用自编程功能时，此引脚推荐开路。要外部下拉，使用 100k Ω 到 200k Ω 的电阻。

在自编程模式下，在自编程库中设置切换到上拉状态。

(c) 在 Flash 存储器编程模式下

当通过 Flash 存储器编程器写入数据时，将此引脚直接连接到 Flash 存储器编程器。这提供了一个 V_{DD} 电平的写入电压到 FLMD0 引脚。

FLMD0 引脚不必由外部进行下拉，因为复位时会内部下拉。要外部下拉，使用 1k Ω 到 200k Ω 的电阻。

2.3 引脚 I/O 电路和未使用引脚的推荐连接方式

表 2-2 为引脚 I/O 电路类型和未使用引脚的推荐连接。

表 2-2. 未使用引脚的连接(1/3)

引脚名称	I/O 电路类型	I/O	未使用引脚的推荐连接方式
P00/TI00	8-R	I/O	输入： 通过电阻独立连接 EV _{DD0} , EV _{DD1} , EV _{SS0} , 或 EV _{SS1} 。 输出： 悬空。
P01/TO00	5-AG		
P02/SO10/TxD1			
P03/SI10/RxD1/SDA10	5-AN		
P04/SCK10/SCL10			
P05	8-R		
P06			
P10/SCK00	5-AN		
P11/SI00/RxD0			
P12/SO00/TxD0	5-AG		
P13/TxD3			
P14/RxD3	8-R		
P15/RTCDIV/RTCCL	5-AG		
P16/TI01/TO01/INTP5	8-R		
P17/TI02/TO02			
P20/ANI0 至 P27/ANI7 [‡]	11-G		输入： 通过一个电阻独立连接到 AV _{REF} 或者 AV _{SS} 。 输出： 悬空。
P30/RTC1HZ/INTP3	8-R		输入： 通过电阻独立校正 EV _{DD0} , EV _{DD1} , EV _{SS0} , 或 EV _{SS1} 。 输出： 悬空。
P31/TI03/TO03/INTP4			
P40/TOOL0			<当片上调试使能时> 上拉此引脚(禁止下拉)。 <当片上调试禁止时> 输入： 通过电阻独立连接 EV _{DD0} , EV _{DD1} , EV _{SS0} , 或 EV _{SS1} 。 输出： 悬空。
P41/TOOL1	5-AG		输入： 通过电阻独立连接 EV _{DD0} , EV _{DD1} , EV _{SS0} , 或 EV _{SS1} 。 输出： 悬空。
P42/TI04/TO04	8-R		
P43/SCK01			
P44/SI01			
P45/SO01	5-AG		
P46/TI05/TO05/INTP1	8-R		
P47/INTP2			

注 P20/ANI0 至 P27/ANI7 复位释放后设置为数字输入端口模式。

表 2-2. 未使用引脚的连接(2/3)

引脚名称	I/O 电路类型	I/O	未使用引脚的推荐连接方式
P50, P51, P53 至 P55	8-R	I/O	输入： 通过电阻独立连接 EV _{DD0} , EV _{DD1} , EV _{SS0} , 或 EV _{SS1} 。 输出： 悬空。
P52, P56, P57	5-AG		
P60/SCL0	13-R		输入： 连接 EV _{SS0} 或 EV _{SS1} 。
P61/SDA0			输出： 设置端口输出锁为 0 且让这些引脚悬空 通过低电平输出。
P62, P63	13-P		
P64/TI10/TO10	8-R		输入： 通过电阻独立连接 EV _{DD0} , EV _{DD1} , EV _{SS0} , 或 EV _{SS1} 。 输出： 悬空。
P65/TI11/TO11			
P66/TI12/TO12			
P67/TI13/TO13			
P70/KR0 至 P73//KR3			
P74/KR4/INTP8 至 P77/KR7/INTP11			
P80 至 P87	5-AG		
P91			
P110	8-R		
P111	5-AG		
P120/INTP0/EXLVI	8-R		
P121/X1 ^{註1}	37-B	输入	通过电阻分别连接到 V _{DD} 或 V _{SS} 。
P122/X2/EXCLK ^{註1}			
P123/XT1 ^{註1}			
P124/XT2 ^{註1}			
P130	3-C	输出	悬空。
P131/TI06/TO06	8-R	I/O	输入： 通过电阻独立连接 EV _{DD0} , EV _{DD1} , EV _{SS0} , 或 EV _{SS1} 。 输出： 悬空。
P140/PCLBUZ0/ITNP6			
P141/PCLBUZ1/INTP7			
P142/SCK20/SCL20	5-AN		
P143/SI20/RxD2/SDA20			
P144/SO20/TxD2	5-AG		
P145/TI07/TO07	8-R		
P150/ANI8 至 P157/ANI15 ^{註2}	11-G		输入： 通过一个电阻单独连接到 AV _{REF} 或者 AV _{SS} 。 输出： 悬空。

- 注 1. 当这些引脚不使用时，在输入端口模式下，使用上面推荐的连接(参见 图 5-2 时钟操作模式控制寄存器(CMC)的格式)。
2. P150/ANI8 至 P157/ANI15 复位释放后设置为数字输入端口模式。

表 2-2. 未使用引脚的连接(3/3)

引脚名称	I/O 电路类型	I/O	未使用引脚的推荐连接方式
AVREF	—	—	<当 P20 至 P27 和 P150 至 P157 中一个或多个设置为数字端口时> 使该引脚与 EVDD0, EVDD1, 或 VDD 同电势。 <当 P20 至 P27 和 P150 至 P157 都设置为端口时> 使该引脚的电势为 $1.8\text{ V} \leq \text{AVREF} \leq \text{VDD}$ 。
AVss	—	—	使得该引脚与 EVSS0, EVSS1, 或 VSS 同电势。
FLMD0	2-W	—	悬空或通过一个 100kΩ或更大值的电阻连接到 VSS。
RESET	2	输入	直接或通过一个电阻连接到 EVDD0 或 EVDD1。
REGC	—	—	通过一个电容(0.47 至 1 μF: 目标值)连接到 VSS。

图 2-1. 引脚 I/O 电路列表(1/2)

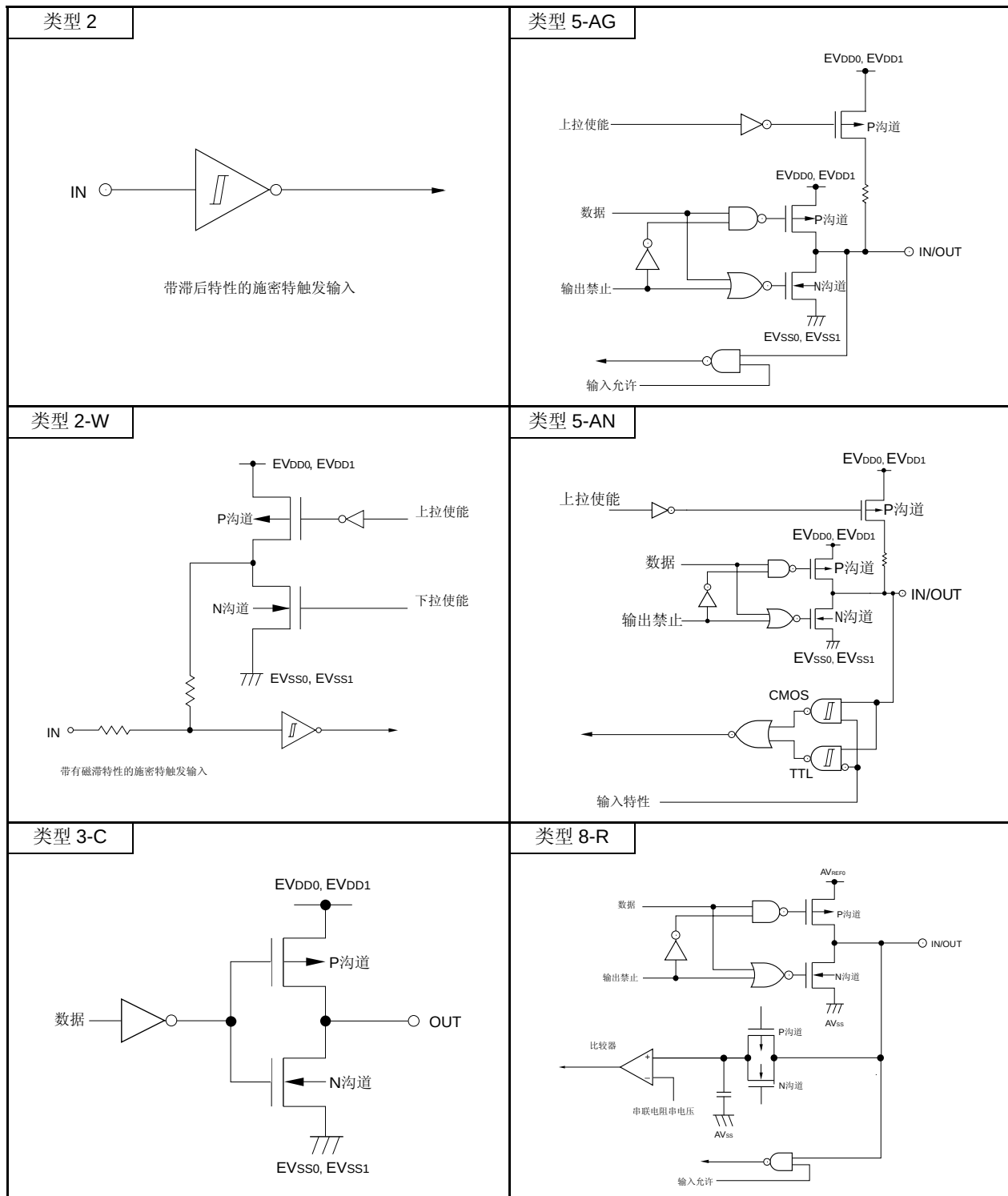
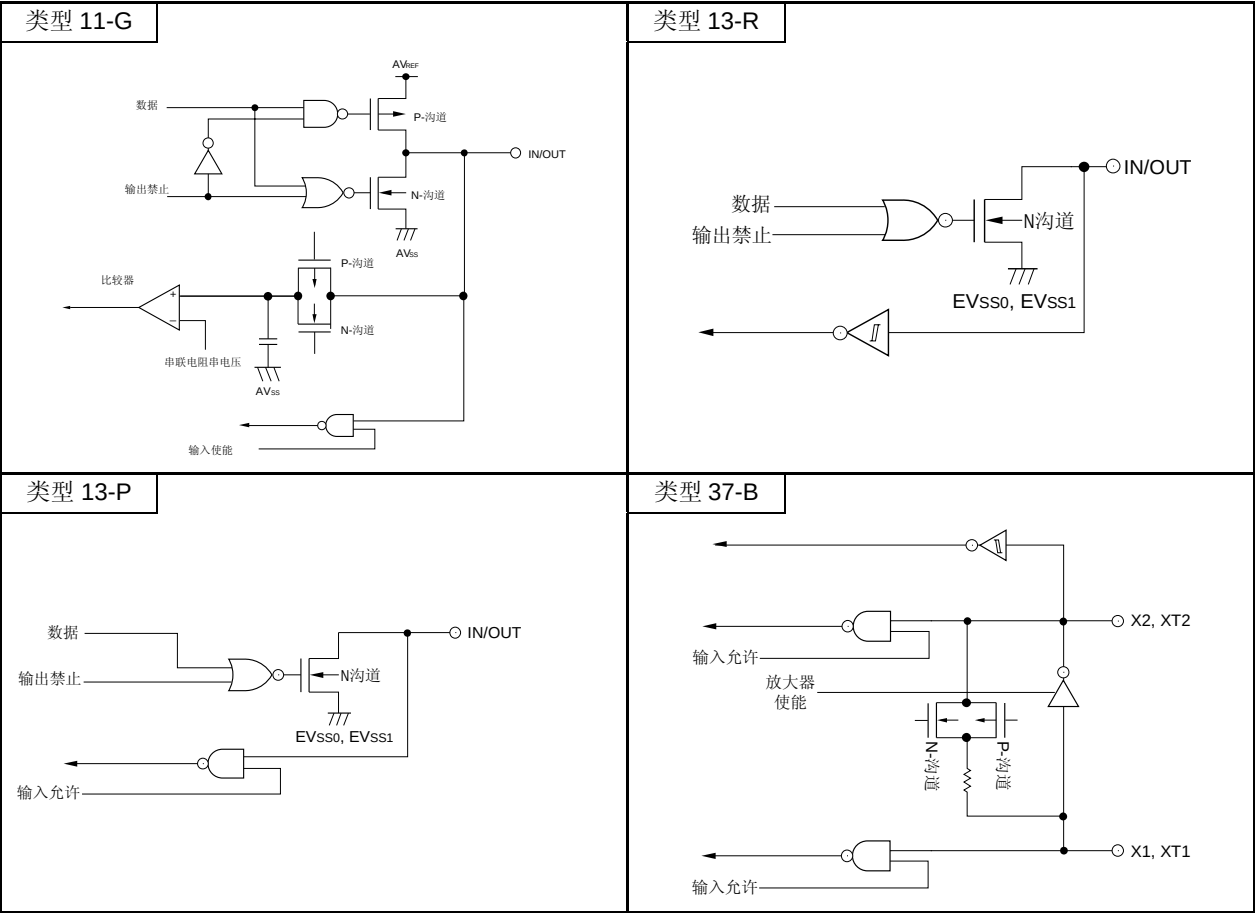


图 2-1. 引脚 I/O 电路列表(2/2)

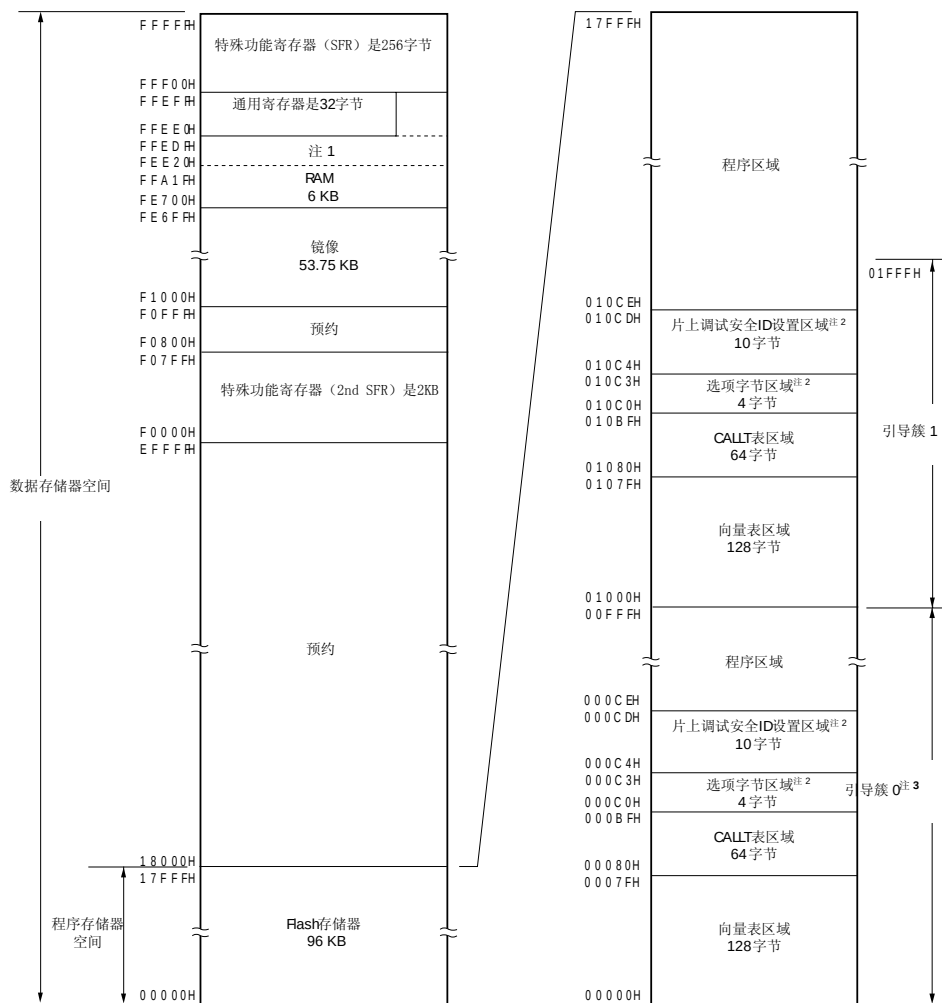


第3章 CPU 架构

3.1 存储器空间

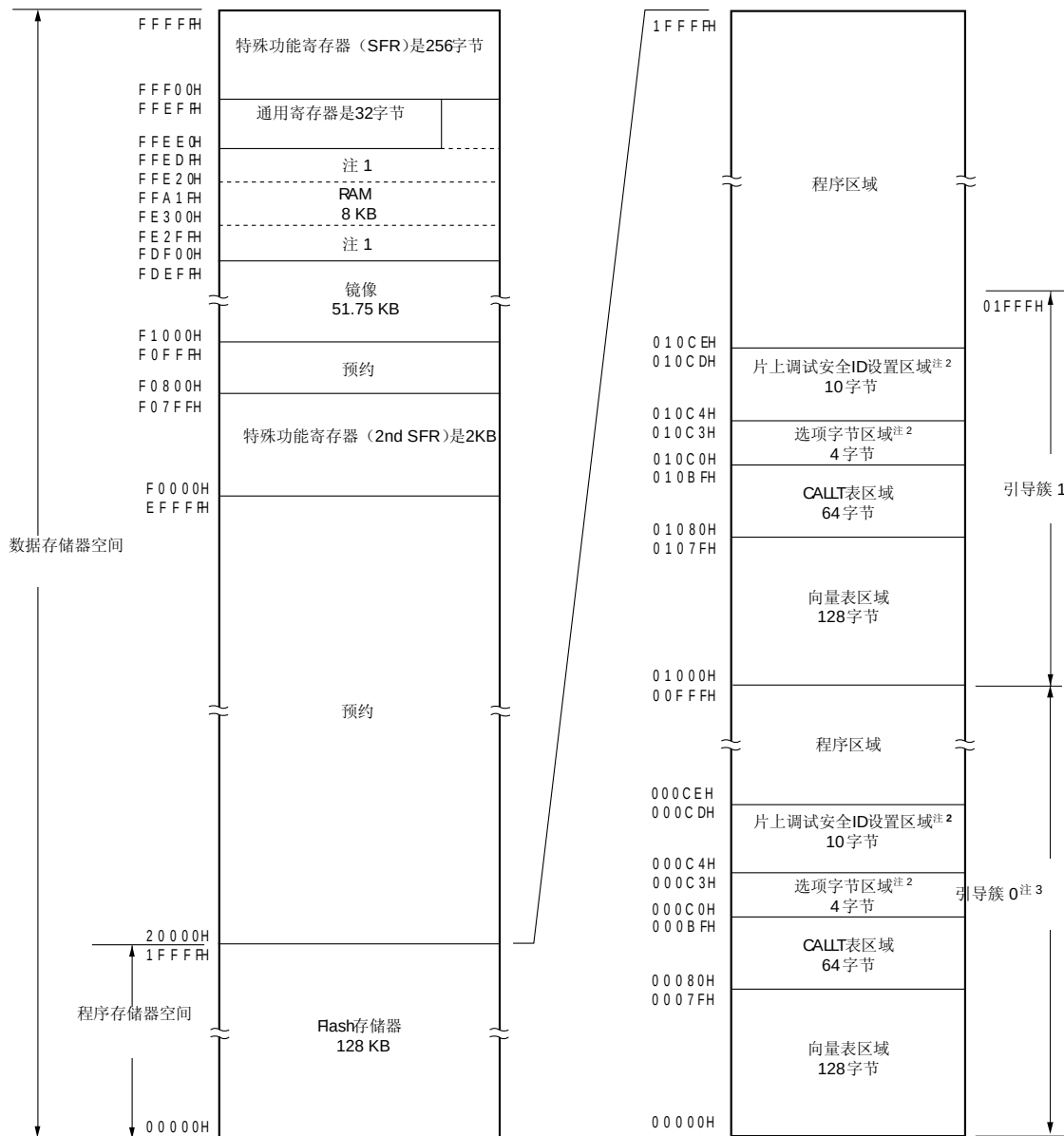
78K0R/KG3-L 的产品可访问 1MB 的空间。图 3-1 和 3-2 显示了存储空间映射图。

图 3-1. 存储器映射图 (μPD78F1013)



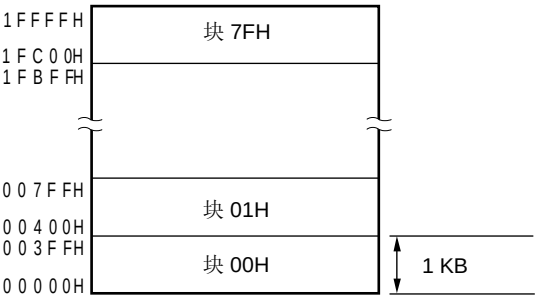
- 注
1. 当使用自编程功能时，禁止使用区域 FFE20H 至 FFEDFH，因为该区域用于自编程库。
 2. 当未使用引导交换时：设置选项字节到 000C0H 至 000C3H，及片上调试安全 ID 到 000C4H 至 000CDH。
当使用引导交换时：设置选项字节到 000C0H 至 000C3H 和 010C0H 至 010C3H，及片上调试安全 ID 到 000C4H 至 000CDH 和 010C4H 至 010CDH。
 3. 根据安全设置，可以禁止写入引导簇 0（参见 23.7 安全设置）。

图 3-2. 存储器映射图 (μPD78F1014)



- 注
1. 当使用自编程功能时，禁止使用区域 FFE20H 至 FFEDFH 和 FDF00H 至 FE2FFH，因为该区域用于自编程库。
 2. 当未使用引导交换时：设置选项字节到 000C0H 至 000C3H，及片上调试安全 ID 到 000C4H 至 000CDH。
当使用引导交换时：设置选项字节到 000C0H 至 000C3H 和 010C0H 至 010C3H，及片上调试安全 ID 到 000C4H 至 000CDH 和 010C4H 至 010CDH。
 3. 根据安全设置，可以禁止写入引导簇 0（参见 23.7 安全设置）。

备注 Flash 存储器分为许多块(1 块 = 1 KB)。如需了解地址值和块的编号，请参见表 3-1 Flash 存储器的地址值和块编号之间的对应关系。



Flash 存储器的地址值和块编号之间的关系如下所示。

表 3-1. Flash 存储器中地址值和块编号之间的对应关系

地址值	块编号	地址值	块编号	地址值	块编号	地址值	块编号
00000H至003FFH	00H	08000H至083FFH	20H	10000H至103FFH	40H	18000H至183FFH	60H
00400H至007FFH	01H	08400H至087FFH	21H	10400H至107FFH	41H	18400H至187FFH	61H
00800H至00BFFH	02H	08800H至08BFFH	22H	10800H至10BFFH	42H	18800H至18BFFH	62H
00C00H至00FFFH	03H	08C00H至08FFFH	23H	10C00H至10FFFH	43H	18C00H至18FFFH	63H
01000H至013FFH	04H	09000H至093FFH	24H	11000H至113FFH	44H	19000H至193FFH	64H
01400H至017FFH	05H	09400H至097FFH	25H	11400H至117FFH	45H	19400H至197FFH	65H
01800H至01BFFH	06H	09800H至09BFFH	26H	11800H至11BFFH	46H	19800H至19BFFH	66H
01C00H至01FFFH	07H	09C00H至09FFFH	27H	11C00H至11FFFH	47H	19C00H至19FFFH	67H
02000H至023FFH	08H	0A000H至0A3FFH	28H	12000H至123FFH	48H	1A000H至1A3FFH	68H
02400H至027FFH	09H	0A400H至0A7FFH	29H	12400H至127FFH	49H	1A400H至1A7FFH	69H
02800H至02BFFH	0AH	0A800H至0ABFFH	2AH	12800H至12BFFH	4AH	1A800H至1ABFFH	6AH
02C00H至02FFFH	0BH	0AC00H至0AFFFH	2BH	12C00H至12FFFH	4BH	1AC00H至1AFFFH	6BH
03000H至033FFH	0CH	0B000H至0B3FFH	2CH	13000H至133FFH	4CH	1B000H至1B3FFH	6CH
03400H至037FFH	0DH	0B400H至0B7FFH	2DH	13400H至137FFH	4DH	1B400H至1B7FFH	6DH
03800H至03BFFH	0EH	0B800H至0BBFFH	2EH	13800H至13BFFH	4EH	1B800H至1BBFFH	6EH
03C00H至03FFFH	0FH	0BC00H至0BFFFH	2FH	13C00H至13FFFH	4FH	1BC00H至1BFFFH	6FH
04000H至043FFH	10H	0C000H至0C3FFH	30H	14000H至143FFH	50H	1C000H至1C3FFH	70H
04400H至047FFH	11H	0C400H至0C7FFH	31H	14400H至147FFH	51H	1C400H至1C7FFH	71H
04800H至04BFFH	12H	0C800H至0CBFFH	32H	14800H至14BFFH	52H	1C800H至1CBFFH	72H
04C00H至04FFFH	13H	0CC00H至0CFFFH	33H	14C00H至14FFFH	53H	1CC00H至1CFFFH	73H
05000H至053FFH	14H	0D000H至0D3FFH	34H	15000H至153FFH	54H	1D000H至1D3FFH	74H
05400H至057FFH	15H	0D400H至0D7FFH	35H	15400H至157FFH	55H	1D400H至1D7FFH	75H
05800H至05BFFH	16H	0D800H至0DBFFH	36H	15800H至15BFFH	56H	1D800H至1DBFFH	76H
05C00H至05FFFH	17H	0DC00H至0DFFFH	37H	15C00H至15FFFH	57H	1DC00H至1DFFFH	77H
06000H至063FFH	18H	0E000H至0E3FFH	38H	16000H至163FFH	58H	1E000H至1E3FFH	78H
06400H至067FFH	19H	0E400H至0E7FFH	39H	16400H至167FFH	59H	1E400H至1E7FFH	79H
06800H至06BFFH	1AH	0E800H至0EBFFH	3AH	16800H至16BFFH	5AH	1E800H至1EBFFH	7AH
06C00H至06FFFH	1BH	0EC00H至0EFFFH	3BH	16C00H至16FFFH	5BH	1EC00H至1EFFFH	7BH
07000H至073FFH	1CH	0F000H至0F3FFH	3CH	17000H至173FFH	5CH	1F000H至1F3FFH	7CH
07400H至077FFH	1DH	0F400H至0F7FFH	3DH	17400H至177FFH	5DH	1F400H至1F7FFH	7DH
07800H至07BFFH	1EH	0F800H至0FBFFH	3EH	17800H至17BFFH	5EH	1F800H至1FBFFH	7EH
07C00H至07FFFH	1FH	0FC00H至0FFFFH	3FH	17C00H至17FFFH	5FH	1FC00H至1FFFFH	7FH

备注 μPD78F1013: 块编号 00H 至 5FH
 μPD78F1014: 块编号 00H 至 7FH

3.1.1 内部程序存储器空间

内部程序存储空间用于存储程序和表格数据。
78K0R/KG3-L 产品集成内部 ROM（Flash 存储器），如下所示。

表 3-2. 内部 ROM 容量

产品型号	内部 ROM	
	结构	容量
μPD78F1013	Flash 存储器	98304 × 8 位 (00000H 至 17FFFH)
μPD78F1014		131072 × 8 位 (00000H 至 1FFFFH)

内部程序存储空间分为以下几个区域。

(1) 向量表区域

00000H 至 0007FH 的 128 字节保留为向量表区域。在向量表区域中存放的是用于分支复位或每个中断请求的产生的程序的起始地址。此外，中断跳转地址为 00000H 至 0FFFFH 的 64K 地址，因为向量代码设置为 2 个字节。在 16 位地址中，低 8 位存于偶地址中，高 8 位存于奇地址中。

表 3-3. 向量表

向量表地址	中断源	向量表地址	中断源
00000H	RESET 输入,POC,LVI,WDI, TRAP	00032H	INTTM03
		00034H	INTAD
00004H	INTWDI	00036H	INTRTC
00006H	INTLVI	00038H	INTRTCI
00008H	INTP0	0003AH	INTKR
0000AH	INTP1	0003CH	INTST2/INTCSI20/INTIIC20
0000CH	INTP2	0003EH	INTP6
0000EH	INTP3	00040H	INTTM13
00010H	INTP4	00042H	INTTM04
00012H	INTP5	00044H	INTTM05
00014H	INTST3	00046H	INTTM06
00016H	INTSR3	00048H	INTTM07
00018H	INTSRE3	0004AH	INTSR2
0001AH	INTDMA0	0004CH	INTP7
0001CH	INTDMA1	0004EH	INTP8
0001EH	INTST0/INTCSI00	00050H	INTP9
00020H	INTSR0/INTCSI01	00052H	INTP10
00022H	INTSRE0	00054H	INTP11
00024H	INTST1/INTCSI10/INTIIC10	00056H	INTTM10
00026H	INTSR1	00058H	INTTM11
00028H	INTSRE1	0005AH	INTTM12
0002AH	INTIICA	0005CH	INTSRE2
0002CH	INTTM00	0005EH	INTMD
0002EH	INTTM01	0007EH	BRK
00030H	INTTM02		

(2) CALLT 指令表区域

00080H 至 000BFH 共 64 字节的区域，可存放 2 字节调用指令(CALLT)的子程序入口地址。设置子程序入口地址的值在 00000H 至 0FFFFH 范围内（因为地址码为 2 字节）。

要使用引导交换功能，设置 CALLT 指令表也在 01080H 至 010BFH。

(3) 选项字节区域

000C0H 至 000C3H 的 4 字节的区域可以用作选项字节区域。当使用引导交换功能时，在 010C0H 至 010C3H 设置选项字节。详细信息参考第 22 章 选项字节。

(4) 片上调试安全 ID 设置区域

000C4H 至 000CDH 和 010C4H 至 010CDH 的 10 字节区域可用作片上调试安全 ID 设置区域。不使用引导交换功能时，在 000C4H 至 000CDH 区域设置片上调试安全 ID；而当使用引导交换功能时，在 000C4H 至 000CDH 和 010C4H 至 010CDH 区域设置。详细信息参考第 24 章 片上调试功能。

3.1.2 镜像区域

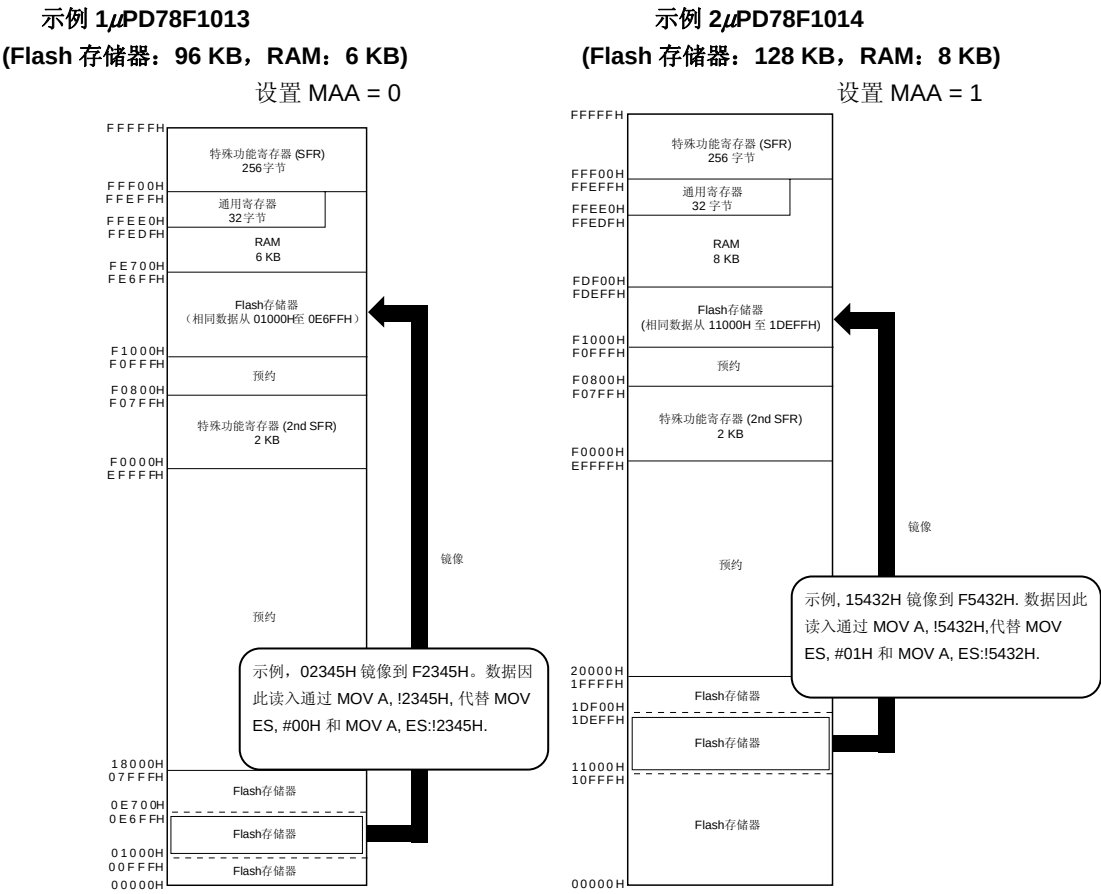
μ PD78F1013 和 78F1014 把 00000H 至 0FFFFH 或 10000H 至 1FFFFH 的数据 Flash 区域，镜像到 F0000H 至 FFFFFH（由处理器模式控制寄存器（PMC）设置）。

通过从镜像区域 F0000H 至 FFFFFH 读取数据，不具备 ES 寄存器的指令可作为操作数使用，因此数据 Flash 的内容可由较短的代码进行读取。但是，数据 Flash 区域不能镜像到 SFR、扩展 SFR、RAM 和禁止使用的区域。

参见 3.1 存储器空间，关于每个产品的镜像区域。

镜像区域只能读取，并且任何指令都不能从此区域中获取。

举例如下。



备注 MAA: 处理器模式控制寄存器(PMC)的位 0

PMC 寄存器描述如下。

● 处理器模式控制寄存器 (PMC)

此寄存器选择 Flash 存储器空间镜像区域 F0000H 至 FFFFFH。

PMC 只能由 1 位或 8 位存储器操作指令设置。

复位信号发生设置此寄存器为 00H。

图 3-3. 处理器模式控制寄存器 (PMC)的格式

地址： FFFFEH 复位后： 00H R/W

符号	7	6	5	4	3	2	1	<0>
PMC	0	0	0	0	0	0	0	MAA

MAA	从区域F0000H至FFFFFH中选择用于镜像的Flash存储器空间
0	00000H至0FFFFH镜像到F0000H至FFFFFH
1	10000H至1FFFFH镜像到F0000H至FFFFFH

- 注意事项
1. 在初始化设置期间，设置 PMC 优先于操作 DMA 控制器。在初始化设置期外，禁止重写 PMC。
 2. 设置 PMC 后，等待至少一个指令周期并访问镜像区域。

3.1.3 内部数据存储器空间

78K0R/KG3-L 产品集成了下列 RAM。

表 3-4. 内部 RAM 容量

产品型号	内部 RAM
μPD78F1013	6144 × 8 位 (FE700H 至 FFEFFH)
μPD78F1014	8192 × 8 位 (FDF00H 至 FFEFFH)

内部 RAM 可用作指令写入和执行的区域和程序区域。在四个通用寄存器区中，每个区由 8 个 8 位寄存器组成，这四个区分配到内部 RAM 区域的 FFEE0H 至 FFEFFH 的 32 位区域。但是，通用寄存器不能执行指令。

内部 RAM 用作堆栈存储器。

- 注意事项
1. 禁止使用通用寄存器 (FFEE0H 至 FFEFFH) 空间获取指令或作为堆栈区域。
 2. 当使用自编程功能时，不能把 FFE20H 至 FFEFFH 区域用作堆栈存储器。此外，FDF00H 至 FE2FFH 区域也不能作为 μPD78F1014 的堆栈存储器使用。

3.1.4 特殊功能寄存器 (SFR) 区域

片上外围硬件特殊功能寄存器 (SFR) 位于 FFF00H 至 FFFFFH 的空间 (参见 **3.2.4 特殊功能寄存器 (SFR)** 的表 3-5)。

注意事项 不要访问那些 SFR 未分配的地址区域。

3.1.5 扩展特殊功能寄存器 (第二 SFR: 第二特殊功能寄存器) 区域

片上外围硬件特殊功能寄存器 (第二 SFR) 分配在 F0000H 至 F07FFH 区域 (参见 **3.2.5 扩展特殊功能寄存器 (第二 SFR: 第二特殊功能寄存器)** 的表 3-6)

除了那些位于 SFR 区域 (FFF00H 至 FFFFFH) 以外的 SFR 均分配于该区域。访问扩展 SFR 区域的指令, 比访问 SFR 区域的指令长 1 字节。

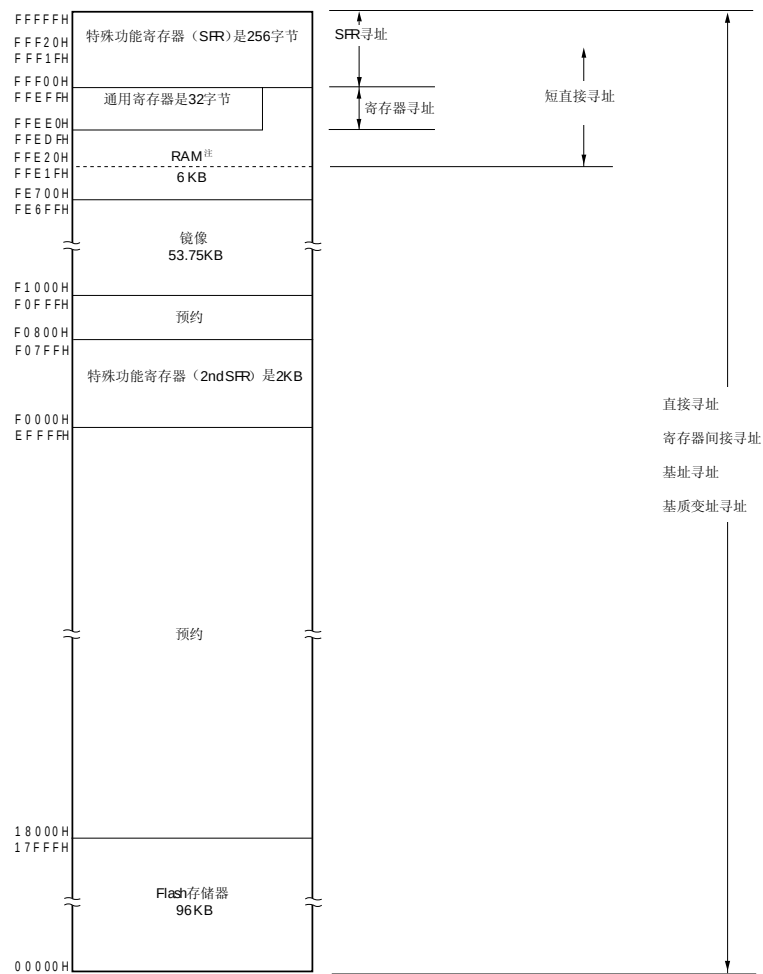
注意事项 不要访问第二 SFR 没有分配的地址。

3.1.6 数据存储器寻址

寻址是定位地址的方式，需要定位的地址包括下一条指令地址、或者与指令执行相关的存储器或寄存器地址。

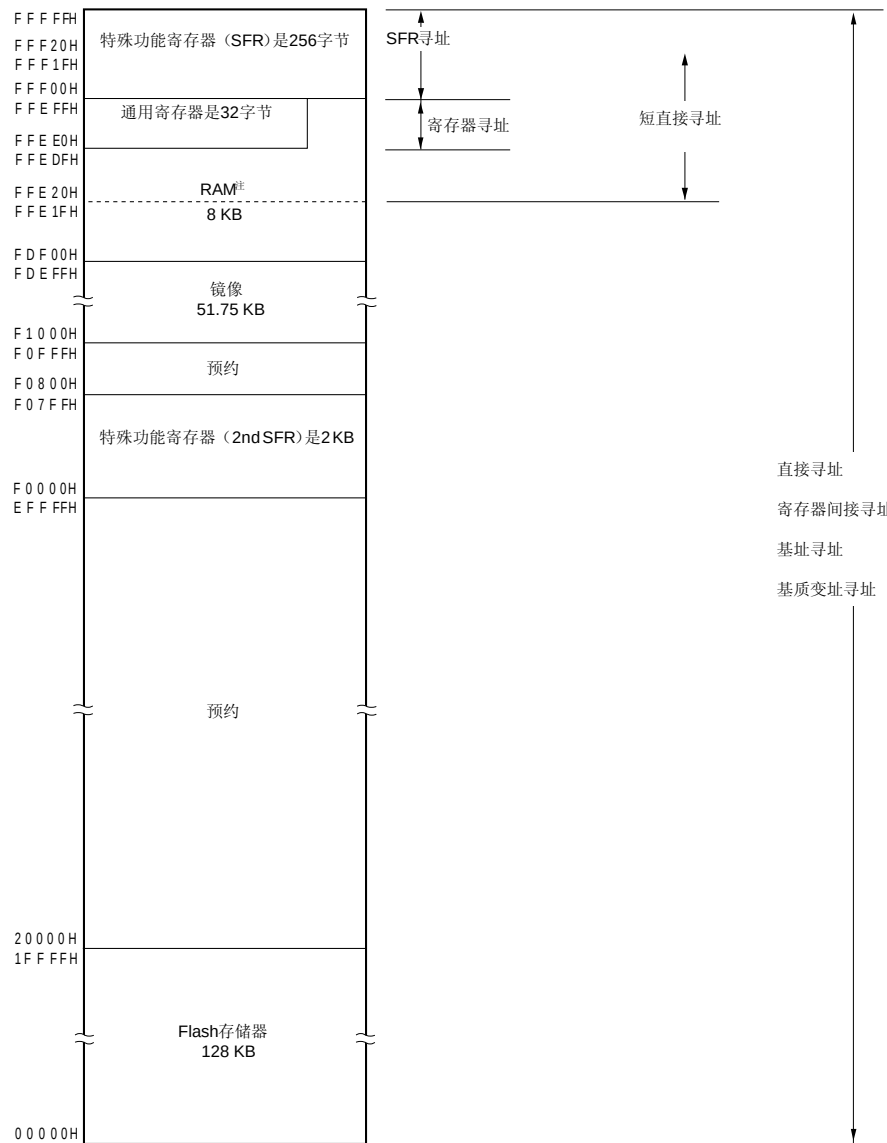
基于可操作性和其他考虑，在 78K0R/KG3-L 中提供了几种用于与指令执行相关的存储器寻址方式。因为有专门的数据存储区域，故可使用专为特殊功能寄存器(SFR)和通用寄存器功能设计的特殊寻址方式。图 3-4 和 3-5 显示了数据存储器寻址和寻址之间的关系。

图 3-4. 数据存储器寻址和选址之间的对应关系(μPD78F1013)



注 当使用自编程功能时，禁止使用区域 FFE20H 至 FFEDFH，因为该区域用于自编程库。

图 3-5. 数据存储器和寻址之间的关系 (μPD78F1014)



注 当使用自编程功能时，禁止使用区域 FFE20H 至 FFEDFH 和 FDF00H 至 FE2FFH，因为该区域用于自编程库。

3.2 处理器寄存器

78K0R/KG3-L 产品集成了下列处理器寄存器。

3.2.1 控制寄存器

控制寄存器用于控制程序执行的顺序、状态和堆栈空间。控制寄存器包含程序计数器(PC)、程序状态字(PSW)和堆栈指针(SP)。

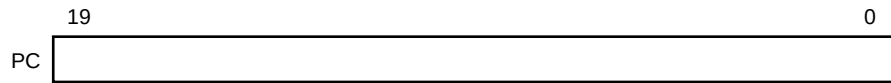
(1) 程序计数器 (PC)

程序计数器是一个 20 位的寄存器，用于存放下一条要执行程序地址信息。

在正常操作下，根据获取的指令字节数，程序计数器(PC)的值会自动累加。当执行分支指令时，则设置立即数和寄存器内容。

复位信号的产生将复位向量表中地址为 0000H 和 0001H 中的值赋给程序计数器。

图 3-6. 程序计数器的格式



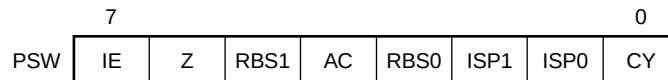
(2) 程序状态字 (PSW)

程序状态字(PSW)是一个 8 位寄存器，由各种标志位组成，通过指令执行对其进行设置或复位。

根据向量中断请求应答或 PUSH PSW 指令执行，程序状态字的内容自动入栈；通过执行 RETB、RETI 和 POP PSW 指令，程序状态字的值自动恢复。

复位信号的产生将程序状态字的内容设置为 06H。

图 3-7. 程序状态字的格式



(a) 中断使能标志 (IE)

该标志用于控制 CPU 响应中断请求操作。

当 IE 为 0 时，表示禁止中断(DI)，即禁止所有可屏蔽中断请求。

当 IE 为 1 时，表示使能中断(EI)，即通过优先服务标志(ISP1, ISP0)、用于各种中断源的中断屏蔽标志以及优先级指定标志来完成响应中断请求的控制。

当执行 DI 指令或中断响应时，该标志复位(0)；当执行 EI 指令时，该标志设置为(1)。

(b) 零标志 (Z)

当运行结果为零时，该标志设置为(1)。其它情况该标志置(0)。

(c) 寄存器 bank 选择标志 (RBS0, RBS1)

寄存器 bank 选择标志是 2 位，用于选择四个寄存器 bank 中的一个。

在这些标志中，存储了表明寄存器 bank 的 2 位信息，而 bank 由 SEL Rn 执行指令选择。

(d) 辅助进位标志 (AC)

如果操作结果中第 3 位有进位或借位，则该标志置(1)。其它情况该标志置(0)。

(e) 优先服务标志(ISP1, ISP0)

该标志用来管理可屏蔽向量中断响应的优先级。通过优先级指定标志寄存器(PRn0L、PRn0H、PRn1L、PRn1H、PRn2L、PRn2H)向量中断请求指定低于 ISP0 和 ISP1 的值（参见 15.3 (3)）不能响应。实际的请求响应由中断使能标志 (IE) 控制。

备注 n = 0, 1

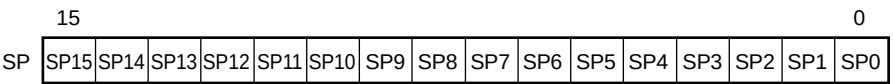
(f) 进位标志 (CY)

该标志存储的是在执行加减指令时出现的进位或借位。它也存储循环指令执行中的移出值，还可以在位操作指令执行中作为位累加器使用。

(3) 堆栈指针 (SP)

这是一个 16 位的寄存器，用来存放存储器堆栈区的起始地址。只有内部高速 RAM 区域才能设置为堆栈区。

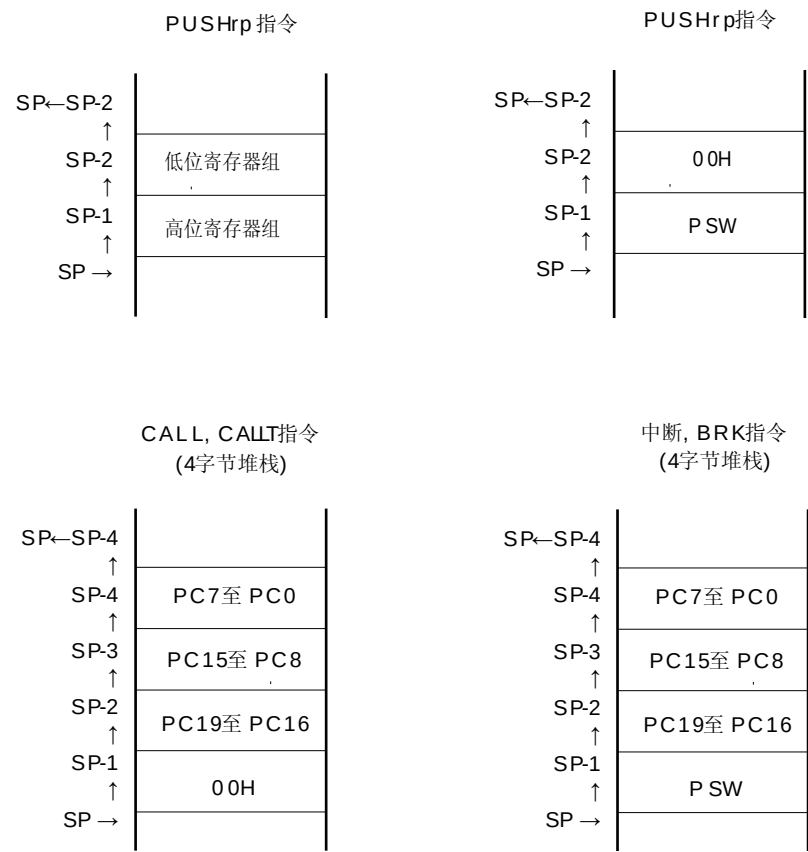
图 3-8. 堆栈指针的格式



在向堆栈写（存入）数据时，堆栈指针 SP 递减，而从堆栈中读出（恢复）数据时，堆栈指针累加。堆栈的数据存储/恢复操作过程如图 3-9 所示。

- 注意事项
- 1. 由于复位信号的产生使 SP 的内容不确定，所以在使用堆栈前必须先对 SP 初始化。
 - 2. 禁止使用通用寄存器（FFEE0H 至 FFEFFH）空间作为堆栈区域。
 - 3. 当使用自编程功能时，不能把 FFE20H 至 FFEFFH 区域用作堆栈存储器。此外，FDF00H 至 FE2FFH 区域也不能作为 μ PD78F1014 的堆栈存储器使用。

图 3-9. 将数据存入堆栈存储器



3.2.2 通用寄存器

通用寄存器映射到数据存储器特定的地址空间为 FFEE0H 至 FFEFFH。通用寄存器共有四组，每一组由 8 个 8 位寄存器（X、A、C、B、E、D、L 和 H）组成。

每个寄存器可作为一个 8 位寄存器使用，两个成对的 8 位寄存器可作为一个 16 位寄存器（AX、BC、DE 和 HL）使用。

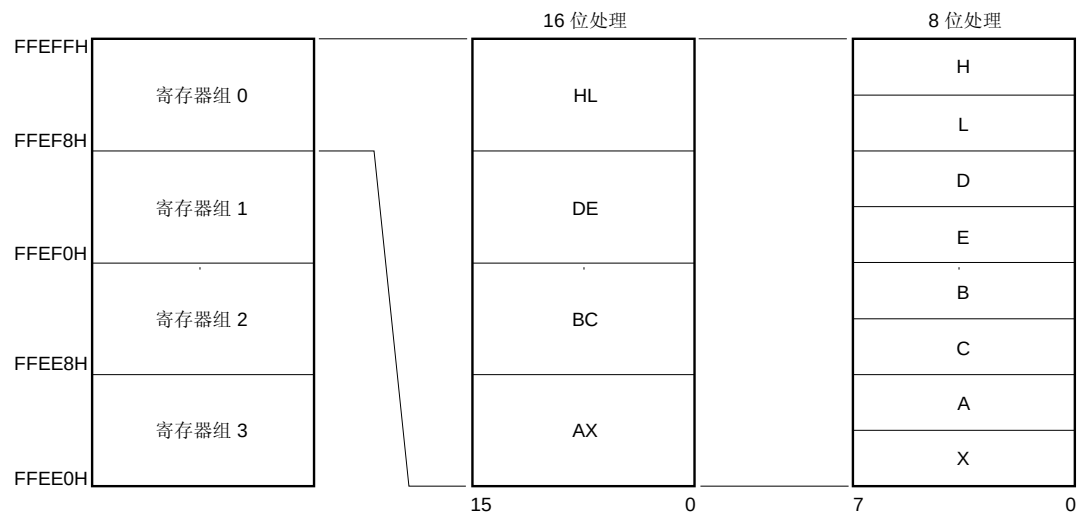
描述通用寄存器时，可以使用功能名称（X、A、C、B、E、D、L、H、AX、BC、DE 和 HL）或绝对名称（R0 至 R7 和 RP0 至 RP3）。

用于指令执行的寄存器组由 CPU 控制指令（SEL RBn）来设置。由于 4 个寄存器 bank 的配置，通过一个用于正常处理的寄存器和另一个用于每个 bank 中断处理的寄存器之间的切换，可以创建一个高效率的程序。

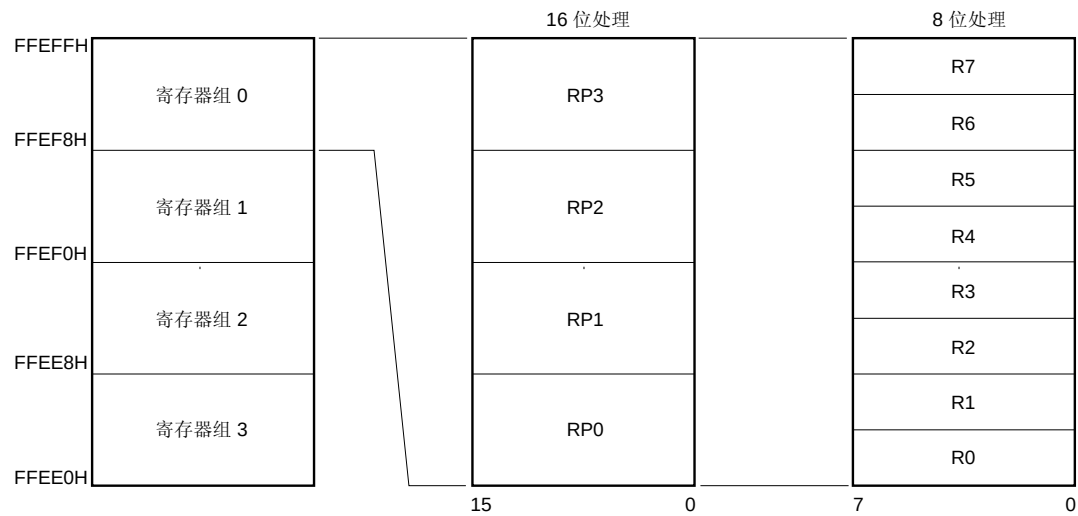
注意事项 禁止使用通用寄存器 (FFEE0H 至 FFEFFH) 空间获取指令或作为堆栈区域。

图 3-10. 通用寄存器的配置

(a) 功能名称



(b) 绝对名称



3.2.3 ES 和 CS 寄存器

ES 寄存器用于数据访问，CS 寄存器用于执行分支指令时指定高位地址。
复位后，ES 寄存器的默认值为 0FH，CS 寄存器的默认值为 00H。

图 3-11. ES 和 CS 寄存器的配置

	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0

	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CP2	CP1	CP0

3.2.4 特殊功能寄存器 (SFR)

与通用寄存器不同，每个 SFR 都有特定的功能。

SFR 分配在 FFF00H 至 FFFFFH 区域。

SFR 可像通用寄存器那样用运算指令、传输指令及位操作指令进行操作。根据特殊功能寄存器的类型不同，可操作的位单元也不同，可以是 1 位、8 位和 16 位。

每种位单元操作的描述如下。

- 1 位操作
描述 1 位操作指令的操作数 (sfr.bit) 为汇编程序的保留符号。该操作也可由一个地址来定义。
- 8 位操作
描述 8 位操作指令的操作数 (sfr) 为汇编程序的保留符号。该操作也可由一个地址来定义。
- 16 位操作
描述 16 位操作指令的操作数 (sfrp) 为汇编程序的保留符号。寻址时表示为一个偶地址。

表 3-5 为 SFR 列表。表中术语的含义如下。

- 符号
表示特殊功能寄存器地址的符号。这是在 RA78K0R 中的保留字，并且定义为 sfr 变量使用 CC78K0R 中的 #pragma sfr 指令。在使用 RA78K0R、ID78K0R-QB 和用于 78K0R 的 SM+时，可以将符号作为指令操作数执行写操作。
- R/W
表示相应的 SFR 是否可读或可写。
R/W: 使能读/写
R: 只读
W: 只写
- 可操作的位单元
“√”表示可操作位单元(1、8 或 16)。“-”表示无法进行操作的位单元。
- 复位后
表示复位信号产生后，每个寄存器的状态。

注意事项 不要访问那些 SFR 未分配的地址区域。

备注 对于扩展 SFR（第二 SFR），参见 3.2.5 扩展特殊功能寄存器（第二 SFR：第二特殊功能寄存器）。

表 3-5. SFR 列表 (1/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位范围			复位后
					1 位	8 位	16 位	
FFF00H	端口寄存器 0	P0		R/W	√	√	—	00H
FFF01H	端口寄存器 1	P1		R/W	√	√	—	00H
FFF02H	端口寄存器 2	P2		R/W	√	√	—	00H
FFF03H	端口寄存器 3	P3		R/W	√	√	—	00H
FFF04H	端口寄存器 4	P4		R/W	√	√	—	00H
FFF05H	端口寄存器 5	P5		R/W	√	√	—	00H
FFF06H	端口寄存器 6	P6		R/W	√	√	—	00H
FFF07H	端口寄存器 7	P7		R/W	√	√	—	00H
FFF08H	端口寄存器 8	P8		R/W	√	√	—	00H
FFF09H	端口寄存器 9	P9		R/W	√	√	—	00H
FFF0BH	端口寄存器 11	P11		R/W	√	√	—	00H
FFF0CH	端口寄存器 12	P12		R/W	√	√	—	未定义
FFF0DH	端口寄存器 13	P13		R/W	√	√	—	00H
FFF0EH	端口寄存器 14	P14		R/W	√	√	—	00H
FFF0FH	端口寄存器 15	P15		R/W	√	√	—	00H
FFF10H	串行数据寄存器 00	TXD0/ SIO00	SDR00	R/W	—	√	√	0000H
FFF11H		—			—	—		
FFF12H	串行数据寄存器 01	RXD0/ SIO01	SDR01	R/W	—	√	√	0000H
FFF13H		—			—	—		
FFF14H	串行数据寄存器 12	TXD3	SDR12	R/W	—	√	√	0000H
FFF15H		—			—	—		
FFF16H	串行数据寄存器 13	RXD3	SDR13	R/W	—	√	√	0000H
FFF17H		—			—	—		
FFF18H	定时器数据寄存器 00	TDR00		R/W	—	—	√	0000H
FFF19H								
FFF1AH	定时器数据寄存器 01	TDR01		R/W	—	—	√	0000H
FFF1BH								
FFF1EH	10 位 A/D 转换结果寄存器		ADCR	R	—	—	√	0000H
FFF1FH	8 位 A/D 转换结果寄存器		ADCRH	R	—	√	—	00H
FFF20H	端口模式寄存器 0	PM0		R/W	√	√	—	FFH
FFF21H	端口模式寄存器 1	PM1		R/W	√	√	—	FFH
FFF22H	端口模式寄存器 2	PM2		R/W	√	√	—	FFH
FFF23H	端口模式寄存器 3	PM3		R/W	√	√	—	FFH
FFF24H	端口模式寄存器 4	PM4		R/W	√	√	—	FFH
FFF25H	端口模式寄存器 5	PM5		R/W	√	√	—	FFH
FFF26H	端口模式寄存器 6	PM6		R/W	√	√	—	FFH
FFF27H	端口模式寄存器 7	PM7		R/W	√	√	—	FFH
FFF28H	端口模式寄存器 8	PM8		R/W	√	√	—	FFH
FFF29H	端口模式寄存器 9	PM9		R/W	√	√	—	FFH
FFF2BH	端口模式寄存器 11	PM11		R/W	√	√	—	FFH
FFF2CH	端口模式寄存器 12	PM12		R/W	√	√	—	FFH
FFF2DH	端口模式寄存器 13	PM13		R/W	√	√	—	FEH
FFF2EH	端口模式寄存器 14	PM14		R/W	√	√	—	FFH
FFF2FH	端口模式寄存器 15	PM15		R/W	√	√	—	FFH

表 3-5. SFR 列表 (2/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位范围			复位后
					1 位	8 位	16 位	
FFF30H	A/D 转换器模式寄存器	ADM		R/W	√	√	—	00H
FFF31H	模拟输入通道指定寄存器	ADS		R/W	√	√	—	00H
FFF37H	按键返回模式寄存器	KRM		R/W	√	√	—	00H
FFF38H	外部中断上升沿使能寄存器 0	EGP0		R/W	√	√	—	00H
FFF39H	外部中断下降沿使能寄存器 0	EGN0		R/W	√	√	—	00H
FFF3AH	外部中断上升沿使能寄存器 1	EGP1		R/W	√	√	—	00H
FFF3BH	外部中断下降沿使能寄存器 1	EGN1		R/W	√	√	—	00H
FFF3CH	输入切换控制寄存器	ISC		R/W	√	√	—	00H
FFF3EH	定时器输入选择寄存器 0	TIS0		R/W	√	√	—	00H
FFF3FH	定时器输入选择寄存器 1	TIS1		R/W	√	√	—	00H
FFF44H	串行数据寄存器 02	TXD1/ SIO10	SDR02	R/W	—	√	√	0000H
FFF45H		—			—	—	—	
FFF46H	串行数据寄存器 03	RXD1	SDR03	R/W	—	√	√	0000H
FFF47H		—			—	—	—	
FFF48H	串行数据寄存器 10	TXD2/ SIO20	SDR10	R/W	—	√	√	0000H
FFF49H		—			—	—	—	
FFF4AH	串行数据寄存器 11	RXD2	SDR11	R/W	—	√	√	0000H
FFF4BH		—			—	—	—	
FFF50H	IICA 移位寄存器	IICA		R/W	—	√	—	00H
FFF51H	IICA 状态寄存器	IICS		R	√	√	—	00H
FFF52H	IICA 标志寄存器	IICF		R/W	√	√	—	00H
FFF64H	定时器数据寄存器 02	TDR02		R/W	—	—	√	0000H
FFF65H								
FFF66H	定时器数据寄存器 03	TDR03		R/W	—	—	√	0000H
FFF67H								
FFF68H	定时器数据寄存器 04	TDR04		R/W	—	—	√	0000H
FFF69H								
FFF6AH	定时器数据寄存器 05	TDR05		R/W	—	—	√	0000H
FFF6BH								
FFF6CH	定时器数据寄存器 06	TDR06		R/W	—	—	√	0000H
FFF6DH								
FFF6EH	定时器数据寄存器 07	TDR07		R/W	—	—	√	0000H
FFF6FH								
FFF70H	定时器数据寄存器 10	TDR10		R/W	—	—	√	0000H
FFF71H								
FFF72H	定时器数据寄存器 11	TDR11		R/W	—	—	√	0000H
FFF73H								
FFF74H	定时器数据寄存器 12	TDR12		R/W	—	—	√	0000H
FFF75H								

表 3-5. SFR 列表 (3/5)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作位范围			复位后
				1 位	8 位	16 位	
FFF76H	定时器数据寄存器 13	TDR13	R/W	—	—	√	0000H
FFF77H							
FFF90H	副计数寄存器	RSUBC	R	—	—	√	0000H
FFF91H							
FFF92H	秒计数寄存器	SEC	R/W	—	√	—	00H
FFF93H	分钟计数寄存器	MIN	R/W	—	√	—	00H
FFF94H	小时计数寄存器	HOURL	R/W	—	√	—	12H ^{注1}
FFF95H	星期计数寄存器	WEEK	R/W	—	√	—	00H
FFF96H	日期计数寄存器	DAY	R/W	—	√	—	01H
FFF97H	月计数寄存器	MONTH	R/W	—	√	—	01H
FFF98H	年计数寄存器	YEAR	R/W	—	√	—	00H
FFF99H	钟表错误修正寄存器	SUBCUD	R/W	—	√	—	00H
FFF9AH	分钟报警寄存器	ALARMWM	R/W	—	√	—	00H
FFF9BH	小时报警寄存器	ALARMWH	R/W	—	√	—	12H
FFF9CH	星期报警寄存器	ALAMMWW	R/W	—	√	—	00H
FFF9DH	实时计数器控制寄存器 0	RTCC0	R/W	√	√	—	00H
FFF9EH	实时计数器控制寄存器 1	RTCC1	R/W	√	√	—	00H
FFF9FH	实时计数器控制寄存器 2	RTCC2	R/W	√	√	—	00H
FFFA0H	时钟操作模式控制寄存器	CMC	R/W	—	√	—	00H
FFFA1H	时钟操作状态控制寄存器	CSC	R/W	√	√	—	C0H
FFFA2H	振荡稳定时间计数器状态寄存器	OSTC	R	√	√	—	00H
FFFA3H	振荡稳定时间选择寄存器	OSTS	R/W	—	√	—	07H
FFFA4H	系统时钟控制寄存器	CKC	R/W	√	√	—	09H
FFFA5H	时钟输出选择寄存器 0	CKS0	R/W	√	√	—	00H
FFFA6H	时钟输出选择寄存器 1	CKS1	R/W	√	√	—	00H
FFFA8H	复位控制标志寄存器	RESF	R	—	√	—	未定义 ^{注2}
FFFA9H	低电压检测寄存器	LVIM	R/W	√	√	—	00H ^{注3}
FFFAAH	低电压检测电平选择寄存器	LVIS	R/W	√	√	—	0EH ^{注4}
FFFABH	看门狗定时器使能寄存器	WDTE	R/W	—	√	—	1A/9A ^{注5}
FFFB0H	DMA SFR 地址寄存器 0	DSA0	R/W	—	√	—	00H
FFFB1H	DMA SFR 地址寄存器 1	DSA1	R/W	—	√	—	00H
FFFB2H	DMA RAM 地址寄存器 0L	DRA0L	DRA0	R/W	—	√	00H
FFFB3H	DMA RAM 地址寄存器 0H	DRA0H		R/W	—	√	00H
FFFB4H	DMA RAM 地址寄存器 1L	DRA1L	DRA1	R/W	—	√	00H
FFFB5H	DMA RAM 地址寄存器 1H	DRA1H		R/W	—	√	00H

- 注
1. 如果 AMPM 位 (RTCC0 寄存器的第 3 位) 复位后设置为 1, 此寄存器的值为 00H。
 2. RESF 的复位值根据复位源不同而不同。
 3. LVIM 的复位值根据复位源和选项字节的设置的不同而不同。
 4. LVIS 的复位值根据复位源不同而不同。
 5. WDTE 的复位值由选项字节的设置决定。

表 3-5. SFR 列表 (4/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位范围			复位后
					1 位	8 位	16 位	
FFFB6H	DMA 字节计数寄存器 0L	DBC0L	DBC0	R/W	—	√	√	00H
FFFB7H	DMA 字节计数寄存器 0H	DBC0H		R/W	—	√		00H
FFFB8H	DMA 字节计数寄存器 1L	DBC1L	DBC1	R/W	—	√	√	00H
FFFB9H	DMA 字节计数寄存器 1H	DBC1H		R/W	—	√		00H
FFFBAAH	DMA 模式控制寄存器 0	DMC0		R/W	√	√	—	00H
FFFBBAH	DMA 模式控制寄存器 1	DMC1		R/W	√	√	—	00H
FFFBCH	DMA 操作控制寄存器 0	DRC0		R/W	√	√	—	00H
FFFBDAH	DMA 操作控制寄存器 1	DRC1		R/W	√	√	—	00H
FFFBEBH	背景事件控制寄存器	BECTL		R/W	√	√	—	00H
FFFC0H	—	PFCMD [※]		—	—	—	—	未定义
FFFC2H	—	PFS [※]		—	—	—	—	未定义
FFFC4H	—	FLPMC [※]		—	—	—	—	未定义
FFFD0H	中断请求标志寄存器 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	中断请求标志寄存器 2H	IF2H		R/W	√	√		00H
FFFD4H	中断屏蔽标志寄存器 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	中断屏蔽标志寄存器 2H	MK2H		R/W	√	√		FFH
FFFD8H	优先级指定标志寄存器 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	优先级指定标志寄存器 02H	PR02H		R/W	√	√		FFH
FFFDCH	优先级指定标志寄存器 12L	PR12L	PR12	R/W	√	√	√	FFH
FFDDH	优先级指定标志寄存器 12H	PR12H		R/W	√	√		FFH
FFFE0H	中断请求标志寄存器 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	中断请求标志寄存器 0H	IF0H		R/W	√	√		00H
FFFE2H	中断请求标志寄存器 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	中断请求标志寄存器 1H	IF1H		R/W	√	√		00H
FFFE4H	中断屏蔽标志寄存器 0L	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H	中断屏蔽标志寄存器 0H	MK0H		R/W	√	√		FFH
FFFE6H	中断屏蔽标志寄存器 1L	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H	中断屏蔽标志寄存器 1H	MK1H		R/W	√	√		FFH
FFFE8H	优先级指定标志寄存器 00L	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H	优先级指定标志寄存器 00H	PR00H		R/W	√	√		FFH
FFFEAH	优先级指定标志寄存器 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFEBAH	优先级指定标志寄存器 01H	PR01H		R/W	√	√		FFH
FFFECH	优先级指定标志寄存器 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFEDH	优先级指定标志寄存器 10H	PR10H		R/W	√	√		FFH
FFFEEH	优先级指定标志寄存器 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	优先级指定标志寄存器 11H	PR11H		R/W	√	√		FFH

注 不要直接对该 SFR 进行操作，因为其用于自编程库。

表 3-5. SFR 列表 (5/5)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作位范围			复位后
				1 位	8 位	16 位	
FFFF0H	乘法/除法数据寄存器 A(L)	MDAL/MULA	R/W	–	–	√	0000H
FFFF1H							
FFFF2H	乘法/除法数据寄存器 A(H)	MDAH/MULB	R/W	–	–	√	0000H
FFFF3H							
FFFF4H	乘法/除法数据寄存器 B(H)	MDBH/MULOH	R/W	–	–	√	0000H
FFFF5H							
FFFF6H	乘法/除法数据寄存器 B(L)	MDBL/MULOL	R/W	–	–	√	0000H
FFFF7H							
FFFFEH	处理器模式控制寄存器	PMC	R/W	√	√	–	00H

备注 扩展 SFR（第二 SFR），参见表 3-6 扩展 SFR（第二 SFR）列表。

3.2.5 扩展特殊功能寄存器（第二 SFR：第二特殊功能寄存器）

与通用寄存器不同，每个扩展 SFR（第二 SFR）都有特殊的功能。

扩展 SFR 分配至 F0000H 至 F07FFH 区域。除了那些位于 SFR 区域(FFF00H 至 FFFFFH)以外的 SFR 均分配于该区域。访问扩展 SFR 区域的指令，比访问 SFR 区域的指令长 1 字节。

扩展 SFR 可像通用寄存器那样用运算指令、传输指令以及位操作指令进行操作。根据特殊功能寄存器的类型不同，可操作的位单元也不同，可以是 1 位、8 位和 16 位。

每种位单元操作的描述如下。

- 1 位操作
描述 1 位操作指令的操作数 (!addr16.bit) 为汇编程序的保留符号。该操作也可由一个地址来定义。
- 8 位操作
描述 8 位操作指令的操作数 (!addr16) 为汇编程序的保留符号。该操作也可由一个地址来定义。
- 16 位操作
描述 16 位操作指令的操作数 (!addr16) 为汇编程序的保留符号。寻址时表示为一个偶地址。

表 3-6 为扩展 SFR 列表。表中术语的含义如下。

- 符号
表示扩展 SFR 地址的符号。这是在 RA78K0R 中的保留字，并且定义为 sfr 变量使用 CC78K0R 中的 #pragma sfr 指令。在使用 RA78K0R、ID78K0R-QB 和用于 78K0R 的 SM+ 时，可以将符号作为指令操作数执行写操作。
- R/W
表示相应的扩展 SFR 是否可读或可写。
R/W: 可读/写
R: 只读
W: 只写
- 可操作的位单元
“√”表示可操作位单元(1,8 或 16)。“-”表示无法进行操作的位单元。
- 复位后
表示复位信号产生后，每个寄存器的状态。

注意事项 不要访问第二 SFR 没有分配的地址。

备注 SFR 区域的 SFR，参见 3.2.4 特殊功能寄存器 (SFR)。

表 3-6. 扩展 SFR（第二 SFR）列表 (1/6)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作位范围			复位后
				1 位	8 位	16 位	
F0017H	A/D 端口配置寄存器	ADPC	R/W	—	√	—	10H
F0030H	上拉电阻选项寄存器 0	PU0	R/W	√	√	—	00H
F0031H	上拉电阻选项寄存器 1	PU1	R/W	√	√	—	00H
F0033H	上拉电阻选项寄存器 3	PU3	R/W	√	√	—	00H
F0034H	上拉电阻选项寄存器 4	PU4	R/W	√	√	—	00H
F0035H	上拉电阻选项寄存器 5	PU5	R/W	√	√	—	00H
F0036H	上拉电阻选项寄存器 6	PU6	R/W	√	√	—	00H
F0037H	上拉电阻选项寄存器 7	PU7	R/W	√	√	—	00H
F0038H	上拉电阻选项寄存器 8	PU8	R/W	√	√	—	00H
F0039H	上拉电阻选项寄存器 9	PU9	R/W	√	√	—	00H
F003BH	上拉电阻选项寄存器 11	PU11	R/W	√	√	—	00H
F003CH	上拉电阻选项寄存器 12	PU12	R/W	√	√	—	00H
F003DH	上拉电阻选项寄存器 13	PU13	R/W	√	√	—	00H
F003EH	上拉电阻选项寄存器 14	PU14	R/W	√	√	—	00H
F0040H	端口输入模式寄存器 0	PIM0	R/W	√	√	—	00H
F0041H	端口输入模式寄存器 1	PIM1	R/W	√	√	—	00H
F004EH	端口输入模式寄存器 14	PIM14	R/W	√	√	—	00H
F0050H	端口输出模式寄存器 0	POM0	R/W	√	√	—	00H
F0051H	端口输出模式寄存器 1	POM1	R/W	√	√	—	00H
F005EH	端口输出模式寄存器 14	POM14	R/W	√	√	—	00H
F0060H	噪声滤波器使能寄存器 0	NFEN0	R/W	√	√	—	00H
F0061H	噪声滤波器使能寄存器 1	NFEN1	R/W	√	√	—	00H
F0062H	噪声滤波器使能寄存器 2	NFEN2	R/W	√	√	—	00H
F00E0H	乘法/除法数据寄存器 C (L)	MDCL	R	—	—	√	0000H
F00E2H	乘法/除法数据寄存器 C (H)	MDCH	R	—	—	√	0000H
F00E8H	乘法/除法控制寄存器	MDUC	R/W	√	√	—	00H
F00F0H	外围使能寄存器 0	PER0	R/W	√	√	—	00H
F00F3H	操作速度模式控制寄存器	OSMC	R/W	—	√	—	00H
F00F4H	稳压器模式控制寄存器	RMC	R/W	—	√	—	00H
F00F6H	20 MHz 内部高速振荡控制寄存器	DSCCTL	R/W	√	√	—	00H
F00FEH	BCD 调整结果寄存器	BCDADJ	R	—	√	—	未定义
F0100H	串行状态寄存器 00	SSR00L	SSR00	R	—	√	0000H
F0101H		—			—	—	
F0102H	串行状态寄存器 01	SSR01L	SSR01	R	—	√	0000H
F0103H		—			—	—	
F0104H	串行状态寄存器 02	SSR02L	SSR02	R	—	√	0000H
F0105H		—			—	—	
F0106H	串行状态寄存器 03	SSR03L	SSR03	R	—	√	0000H
F0107H		—			—	—	
F0108H	串行标志清零触发寄存器 00	SIR00L	SIR00	R/W	—	√	0000H
F0109H		—			—	—	

表 3-6. 扩展 SFR (第二 SFR) 列表 (2/6)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位范围			复位后
					1 位	8 位	16 位	
F010AH	串行标志清零触发寄存器 01	SIR01L	SIR01	R/W	—	√	√	0000H
F010BH		—			—			
F010CH	串行标志清零触发寄存器 02	SIR02L	SIR02	R/W	—	√	√	0000H
F010DH		—			—			
F010EH	串行标志清零触发寄存器 03	SIR03L	SIR03	R/W	—	√	√	0000H
F010FH		—			—			
F0110H	串行模式寄存器 00	SMR00		R/W	—	—	√	0020H
F0111H								
F0112H	串行模式寄存器 01	SMR01		R/W	—	—	√	0020H
F0113H								
F0114H	串行模式寄存器 02	SMR02		R/W	—	—	√	0020H
F0115H								
F0116H	串行模式寄存器 03	SMR03		R/W	—	—	√	0020H
F0117H								
F0118H	串行通信操作设置寄存器 00	SCR00		R/W	—	—	√	0087H
F0119H								
F011AH	串行通信操作设置寄存器 01	SCR01		R/W	—	—	√	0087H
F011BH								
F011CH	串行通信操作设置寄存器 02	SCR02		R/W	—	—	√	0087H
F011DH								
F011EH	串行通信操作设置寄存器 03	SCR03		R/W	—	—	√	0087H
F011FH								
F0120H	串行通道使能状态寄存器 0	SE0L	SE0	R	√	√	√	0000H
F0121H		—			—			
F0122H	串行通道开始寄存器 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		—			—			
F0124H	串行通道停止寄存器 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		—			—			
F0126H	串行时钟选择寄存器 0	SPS0L	SPS0	R/W	—	√	√	0000H
F0127H		—			—			
F0128H	串行输出寄存器 0	SO0		R/W	—	—	√	0F0FH
F0129H								
F012AH	串行输出使能寄存器 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		—			—			
F0134H	串行输出电平寄存器 0	SOL0L	SOL0	R/W	—	√	√	0000H
F0135H		—			—			
F0140H	串行状态寄存器 10	SSR10L	SSR10	R	—	√	√	0000H
F0141H		—			—			
F0142H	串行状态寄存器 11	SSR11L	SSR11	R	—	√	√	0000H
F0143H		—			—			
F0144H	串行状态寄存器 12	SSR12L	SSR12	R	—	√	√	0000H
F0145H		—			—			

表 3-6. 扩展 SFR（第二 SFR）列表 (3/6)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位范围			复位后
					1 位	8 位	16 位	
F0146H	串行状态寄存器 13	SSR13L	SSR13	R	—	√	√	0000H
F0147H		—			—			
F0148H	串行标志清零触发寄存器 10	SIR10L	SIR10	R/W	—	√	√	0000H
F0149H		—			—			
F014AH	串行标志清零触发寄存器 11	SIR11L	SIR11	R/W	—	√	√	0000H
F014BH		—			—			
F014CH	串行标志清零触发寄存器 12	SIR12L	SIR12	R/W	—	√	√	0000H
F014DH		—			—			
F014EH	串行标志清零触发寄存器 13	SIR13L	SIR13	R/W	—	√	√	0000H
F014FH		—			—			
F0150H	串行模式寄存器 10	SMR10		R/W	—	—	√	0020H
F0151H								
F0152H	串行模式寄存器 11	SMR11		R/W	—	—	√	0020H
F0153H								
F0154H	串行模式寄存器 12	SMR12		R/W	—	—	√	0020H
F0155H								
F0156H	串行模式寄存器 13	SMR13		R/W	—	—	√	0020H
F0157H								
F0158H	串行通信操作设置寄存器 10	SCR10		R/W	—	—	√	0087H
F0159H								
F015AH	串行通信操作设置寄存器 11	SCR11		R/W	—	—	√	0087H
F015BH								
F015CH	串行通信操作设置寄存器 12	SCR12		R/W	—	—	√	0087H
F015DH								
F015EH	串行通信操作设置寄存器 13	SCR13		R/W	—	—	√	0087H
F015FH								
F0160H	串行通道使能状态寄存器 1	SE1L	SE1	R	√	√	√	0000H
F0161H		—			—			
F0162H	串行通道开始寄存器 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		—			—			
F0164H	串行通道停止寄存器 1	ST1L	ST1	R/W	√	√	√	0000H
F0165H		—			—			
F0166H	串行时钟选择寄存器 1	SPS1L	SPS1	R/W	—	√	√	0000H
F0167H		—			—			
F0168H	串行输出寄存器 1	SO1		R/W	—	—	√	0F0FH
F0169H								
F016AH	串行输出使能寄存器 1	SOE1L	SOE1	R/W	√	√	√	0000H
F016BH		—			—			
F0174H	串行输出电平寄存器 1	SOL1L	SOL1	R/W	—	√	√	0000H
F0175H		—			—			
F0180H	定时器计数器寄存器 00	TCR00		R	—	—	√	FFFFH
F0181H								
F0182H	定时器计数器寄存器 01	TCR01		R	—	—	√	FFFFH
F0183H								

表 3-6. 扩展 SFR (第二 SFR) 列表 (4/6)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作位范围			复位后
				1 位	8 位	16 位	
F0184H	定时器计数器寄存器 02	TCR02	R	—	—	√	FFFFH
F0185H							
F0186H	定时器计数器寄存器 03	TCR03	R	—	—	√	FFFFH
F0187H							
F0188H	定时器计数器寄存器 04	TCR04	R	—	—	√	FFFFH
F0189H							
F018AH	定时器计数器寄存器 05	TCR05	R	—	—	√	FFFFH
F018BH							
F018CH	定时器计数器寄存器 06	TCR06	R	—	—	√	FFFFH
F018DH							
F018EH	定时器计数器寄存器 07	TCR07	R	—	—	√	FFFFH
F018FH							
F0190H	定时器模式寄存器 00	TMR00	R/W	—	—	√	0000H
F0191H							
F0192H	定时器模式寄存器 01	TMR01	R/W	—	—	√	0000H
F0193H							
F0194H	定时器模式寄存器 02	TMR02	R/W	—	—	√	0000H
F0195H							
F0196H	定时器模式寄存器 03	TMR03	R/W	—	—	√	0000H
F0197H							
F0198H	定时器模式寄存器 04	TMR04	R/W	—	—	√	0000H
F0199H							
F019AH	定时器模式寄存器 05	TMR05	R/W	—	—	√	0000H
F019BH							
F019CH	定时器模式寄存器 06	TMR06	R/W	—	—	√	0000H
F019DH							
F019EH	定时器模式寄存器 07	TMR07	R/W	—	—	√	0000H
F019FH							
F01A0H	定时器状态寄存器 00	TSR00L	R	—	√	√	0000H
F01A1H		—		—	—		
F01A2H	定时器状态寄存器 01	TSR01L	R	—	√	√	0000H
F01A3H		—		—	—		
F01A4H	定时器状态寄存器 02	TSR02L	R	—	√	√	0000H
F01A5H		—		—	—		
F01A6H	定时器状态寄存器 03	TSR03L	R	—	√	√	0000H
F01A7H		—		—	—		
F01A8H	定时器状态寄存器 04	TSR04L	R	—	√	√	0000H
F01A9H		—		—	—		
F01AAH	定时器状态寄存器 05	TSR05L	R	—	√	√	0000H
F01ABH		—		—	—		
F01ACH	定时器状态寄存器 06	TSR06L	R	—	√	√	0000H
F01ADH		—		—	—		
F01AEH	定时器状态寄存器 07	TSR07L	R	—	√	√	0000H
F01AFH		—		—	—		

表 3-6. 扩展 SFR（第二 SFR）列表 (5/6)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位范围			复位后
					1 位	8 位	16 位	
F01B0H	定时器通道使能状态寄存器 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		—			—			
F01B2H	定时器通道开始寄存器 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		—			—			
F01B4H	定时器通道停止寄存器 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		—			—			
F01B6H	定时器时钟选择寄存器 0	TPS0L	TPS0	R/W	—	√	√	0000H
F01B7H		—			—			
F01B8H	定时器输出寄存器 0	TO0L	TO0	R/W	—	√	√	0000H
F01B9H		—			—			
F01BAH	定时器输出使能寄存器 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		—			—			
F01BCH	定时器输出电平寄存器 0	TOL0L	TOL0	R/W	—	√	√	0000H
F01BDH		—			—			
F01BEH	定时器输出模式寄存器 0	TOM0L	TOM0	R/W	—	√	√	0000H
F01BFH		—			—			
F01C0H	定时器计数器寄存器 10	TCR10		R	—	—	√	FFFFH
F01C1H								
F01C2H	定时器计数器寄存器 11	TCR11		R	—	—	√	FFFFH
F01C3H								
F01C4H	定时器计数器寄存器 12	TCR12		R	—	—	√	FFFFH
F01C5H								
F01C6H	定时器计数器寄存器 13	TCR13		R	—	—	√	FFFFH
F01C7H								
F01C8H	定时器模式寄存器 10	TMR10		R/W	—	—	√	0000H
F01C9H								
F01CAH	定时器模式寄存器 11	TMR11		R/W	—	—	√	0000H
F01CBH								
F01CCH	定时器模式寄存器 12	TMR12		R/W	—	—	√	0000H
F01CDH								
F01CEH	定时器模式寄存器 13	TMR13		R/W	—	—	√	0000H
F01CFH								
F01D0H	定时器状态寄存器 10	TSR10L	TSR10	R	—	√	√	0000H
F01D1H		—			—			
F01D2H	定时器状态寄存器 11	TSR11L	TSR11	R	—	√	√	0000H
F01D3H		—			—			
F01D4H	定时器状态寄存器 12	TSR12L	TSR12	R	—	√	√	0000H
F01D5H		—			—			
F01D6H	定时器状态寄存器 13	TSR13L	TSR13	R	—	√	√	0000H
F01D7H		—			—			
F01D8H	定时器通道使能状态寄存器 1	TE1L	TE1	R	√	√	√	0000H
F01D9H		—			—			
F01DAH	定时器通道开始寄存器 1	TS1L	TS1	R/W	√	√	√	0000H
F01DBH		—			—			

表 3-6. 扩展 SFR（第二 SFR）列表 (6/6)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位范围			复位后
					1 位	8 位	16 位	
F01DCH	定时器通道停止寄存器 1	TT1L	TT1	R/W	√	√	√	0000H
F01DDH		—			—			
F01DEH	定时器时钟选择寄存器 1	TPS1L	TPS1	R/W	—	√	√	0000H
F01DFH		—			—			
F01E0H	定时器输出寄存器 1	TO1L	TO1	R/W	—	√	√	0000H
F01E1H		—			—			
F01E2H	定时器输出使能寄存器 1	TOE1L	TOE1	R/W	√	√	√	0000H
F01E3H		—			—			
F01E4H	定时器输出电平寄存器 1	TOL1L	TOL1	R/W	—	√	√	0000H
F01E5H		—			—			
F01E6H	定时器输出模式寄存器 1	TOM1L	TOM1	R/W	—	√	√	0000H
F01B7H		—			—			
F0230H	IICA 控制寄存器 0	IICCTL0		R/W	√	√	—	00H
F0231H	IICA 控制寄存器 1	IICCTL1		R/W	√	√	—	00H
F0232H	IICA 低电平宽度设置寄存器	IICWL		R/W	—	√	—	FFH
F0233H	IICA 高电平宽度设置寄存器	IICWH		R/W	—	√	—	FFH
F0234H	从地址寄存器	SVA		R/W	—	√	—	00H

备注 对于 SFR 区域中的 SFR，参见表 3-5 SFR 列表。

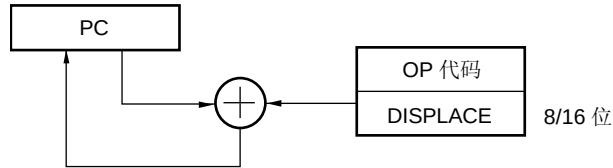
3.3 指令地址寻址

3.3.1 相对寻址

[功能]

相对寻址是把存放在程序计数器(PC)中的值，加上一个包含在指令字中的位移值(带符号数：-128 至+127 或 -32768 至+32767)的结果，再存入程序计数器(PC)中(下一指令的起始地址)，然后将该地址作为分支地址。相对寻址只应用于分支指令。

图 3-12. 相对寻址的图示



3.3.2 立即寻址

[功能]

将指令字中的立即数赋给程序计数器(PC)，然后将该地址作为分支地址。

对于立即寻址，CALL !!addr20 或 BR !!addr20 用作指定 20 位地址，CALL !addr16 或 BR !addr16 用作指定 16 位地址。当指定 16 位地址时，0000 设置到高 4 位。

图 3-13. CALL !!addr20 或 BR !!addr20 的示例

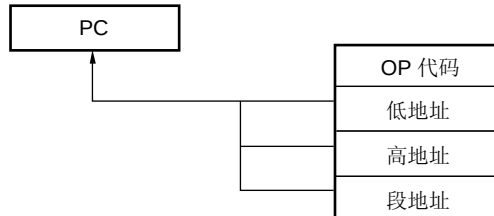
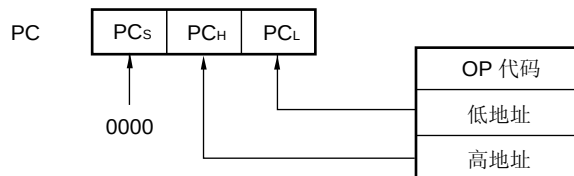


图 3-14. CALL !addr16 或 BR !addr16 的示例

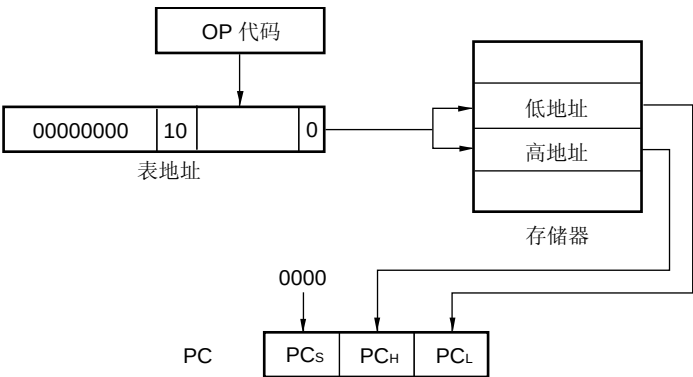


3.3.3 表间接寻址

[功能]

表间接寻址是以指令字中的 5 位立即数，指定 CALLT 表区域(0080H 至 00BFH)的表地址，将该内容和程序计数器(PC)中的下一个地址作为 16 位数据存储在表地址，并指定程序地址。表间接寻址仅适用于 CALLT 指令。
在 78K0R 微控制器中，仅使能在从 00000H 至 0FFFFH 的 64 KB 空间中分支。

图 3-15. 表间接寻址的图示

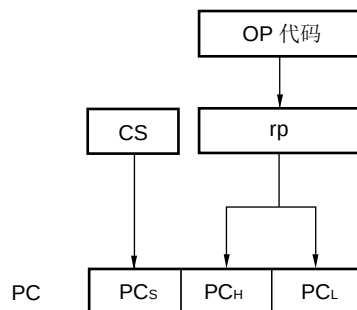


3.3.4 寄存器直接寻址

[功能]

寄存器直接寻址存放于程序计数器（PC）中，通用寄存器对(AX/BC/DE/HL)和当前寄存器区中的 CS 寄存器的内容由指令字定义为 20 位数据，并指定程序地址。寄存器直接寻址仅适用于 CALL,AX,BC,DE,HL 和 BR AX 指令。

图 3-16. 寄存器直接寻址的图示



3.4 用于处理数据地址的寻址

3.4.1 隐式寻址

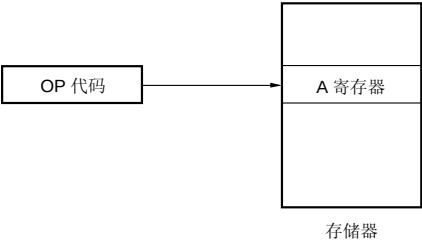
[功能]

用于访问具有特殊功能的寄存器（比如累加器）的指令，直接定义为指令字，在该指令字中不使用任何寄存器专用区域。

[操作数格式]

由于指令自动采用隐式寻址方式，所以无需特定的操作数格式。
是必须的。
隐式寻址仅适用于 MULU X。

图 3-17. 隐式寻址图示



3.4.2 寄存器寻址

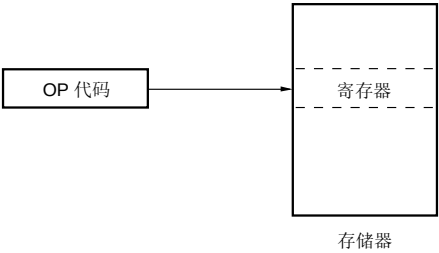
[功能]

寄存器寻址方式将通用寄存器作为操作数进行访问。3 位长的指令字用于选择 8 位寄存器，2 位长的指令字用于选择 16 位寄存器。

[操作数格式]

标识符	描述
r	X,A,C,B,E,D,L,H
rp	AX,BC,DE,HL

图 3-18. 寄存器寻址图示



3.4.3 直接寻址

[功能]
将指令字中的立即数作为操作数地址，直接指定目标地址。

[操作数格式]

标识符	描述
ADDR16	标号或16位立即数（仅F0000H至FFFFFH的空间可指定）
ES:ADDR16	标号或16位立即数（通过ES寄存器指定高4位地址）

图 3-19. ADDR16 的示例

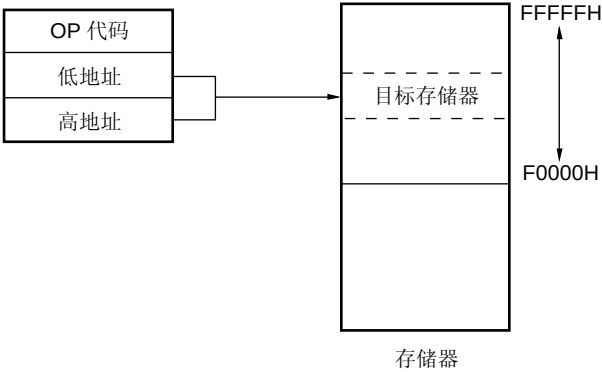
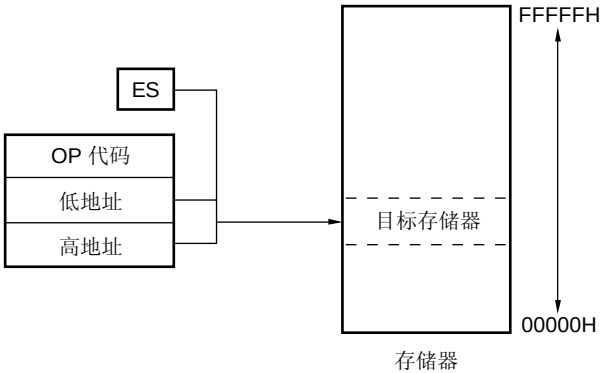


图 3-20. ES:ADDR16 的示例



3.4.4 短直接寻址

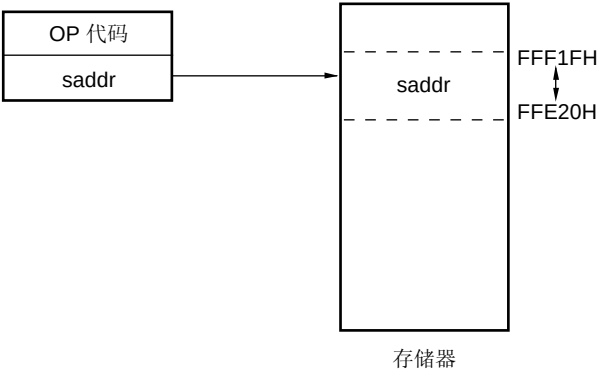
[功能]

短直接寻址通过指令字中的 8 位数据直接指定目标地址。这种寻址方式仅适用于 FFE20H 至 FFF1FH 的空间。

[操作数格式]

标识符	描述
SADDR	标号，FFE20H至FFF1FH的立即数，或0FE20H至0FF1FH的立即数 (仅 FFE20H至FFF1FH 的空间可指定)
SADDRP	标号，FFE20H至FFF1FH的立即数，或0FE20H至0FF1FH的立即数(仅偶地址) (仅 FFE20H至FFF1FH 的空间可指定)

图 3-21. 短直接寻址图示



备注 SADDR 和 SADDRP 用于以 16 位立即数(实际地址的高 4 位忽略)表示地址 FE20H 至 FF1FH 的值，以 20 位立即数表示地址 FFE20H 至 FFF1FH 的值。
不管是否使用 16 位或 20 位的立即数据，FFE20H 至 FFF1FH 空间中的地址都为存储器指定。

3.4.5 SFR 寻址

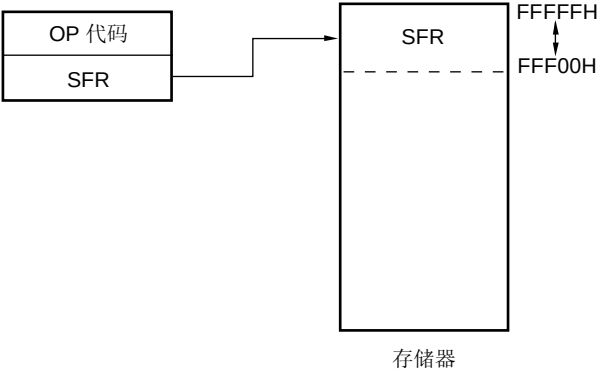
[功能]

SFR 寻址利用指令字中的 8 位数据直接指定目标 SFR 地址。这种寻址方式仅适用于 FFF00H 至 FFFFFH 的空间。

[操作数格式]

标识符	描述
SFR	SFR名称
SFRP	16位可操作SFR名称(仅使用偶地址)

图 3-22. SFR 寻址图示



3.4.6 寄存器间接寻址

[功能]

寄存器间接寻址直接指定了使用由指令字定义的寄存器对内容的目标地址作为操作数地址。

[操作数格式]

标识符	描述
–	[DE],[HL]（仅F0000H至FFFFFH的空间可指定）
–	ES:[DE],ES:[HL]（通过ES寄存器指定高4位地址）

图 3-23. [DE],[HL]的示例

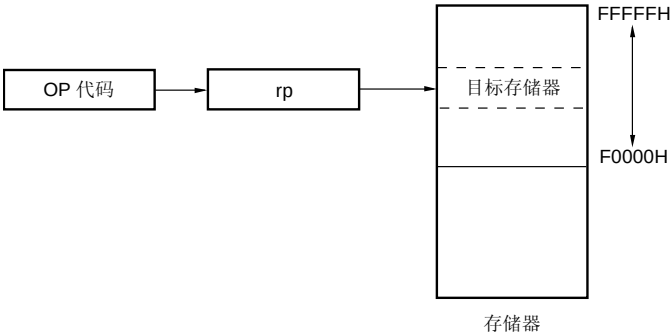
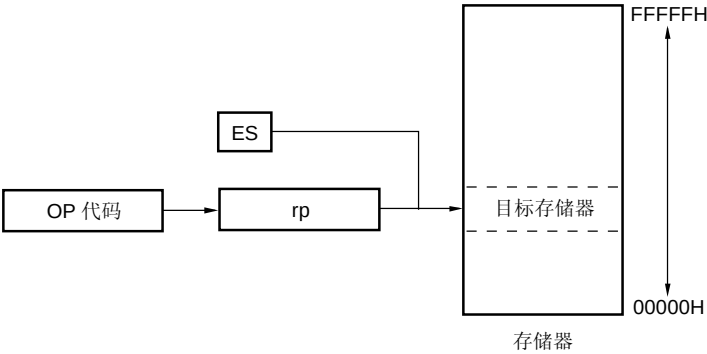


图 3-24. ES:[DE],ES:[HL]的示例



3.4.7 基址寻址

[功能]

基址地址采用由指令字定义的寄存器对内容作为基址，8 位立即数据或 16 位立即数据作为补偿数据。相加结果用于指定目标地址。

[操作数格式]

标识符	描述
–	[HL + 字节], [DE + 字节], [SP + 字节] (仅F0000H至FFFFFH的空间可指定)
–	字[B], 字[C] (仅F0000H至FFFFFH的空间可指定)
–	字[BC] (仅指定F0000H至FFFFFH的空间)
–	ES:[HL + 字节], ES:[DE + 字节] (通过ES寄存器指定高4位地址)
–	ES:字[B], ES:字[C] (通过ES寄存器指定高4位地址)
–	ES:字[BC] (通过ES寄存器指定高4位地址)

图 3-25. [SP + 字节]的示例

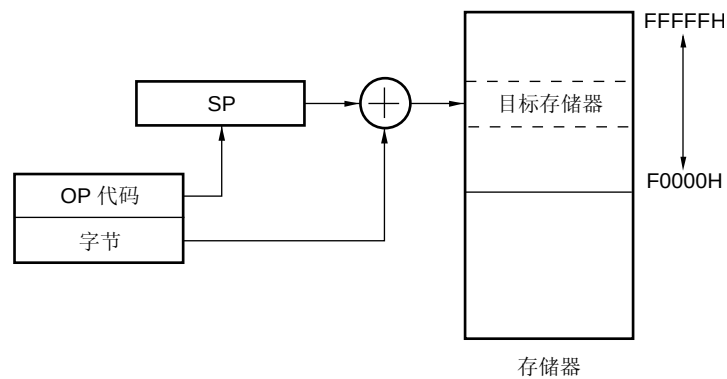


图 3-26. [HL + 字节],[DE + 字节]的示例

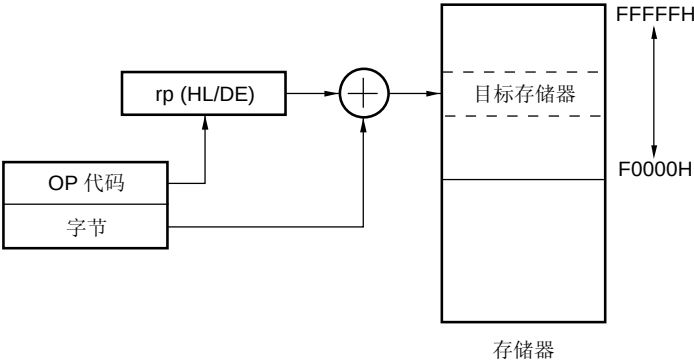


图 3-27. 字[B],字[C]的示例

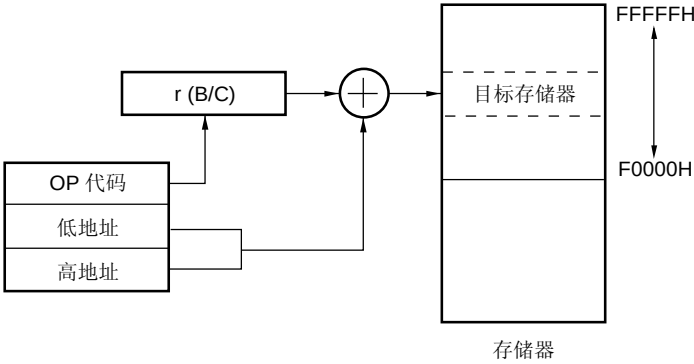


图 3-28. 字[BC]的示例

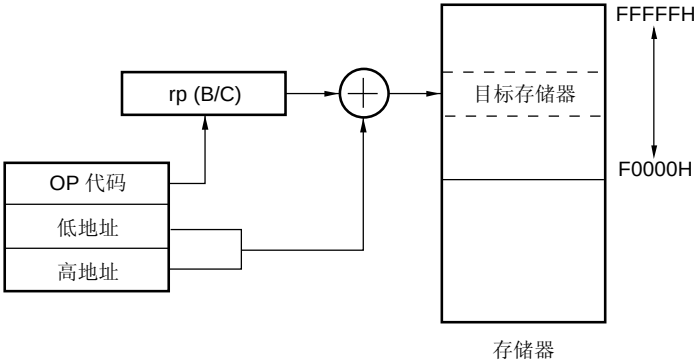


图 3-29. ES:[HL + 字节],ES:[DE + 字节]的示例

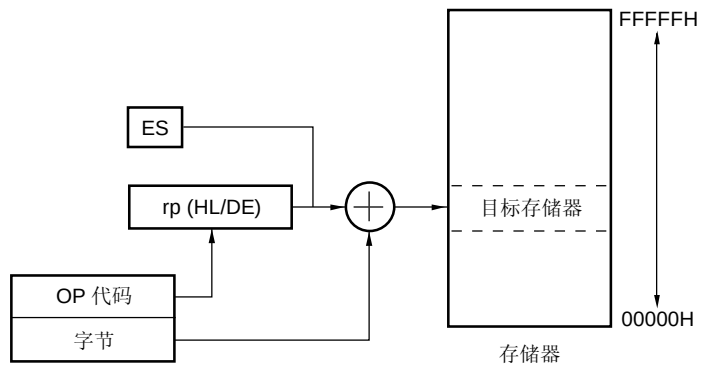


图 3-30. ES: 字[B],ES: 字[C] 的示例

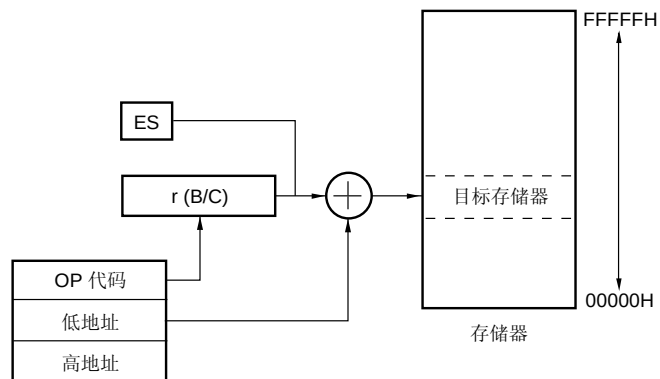
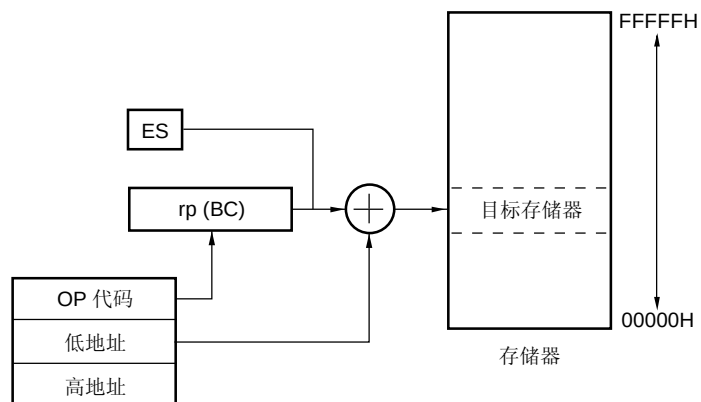


图 3-31. ES: 字[BC] 的示例



3.4.8 基址变址寻址

[功能]

相对地址寻址使用由指令字指定的寄存器对中的内容作为基址，以及由指令字类似指定的 B 寄存器或 C 寄存器的内容作为补偿地址。相加结果用于指定目标地址。

[操作数格式]

标识符	描述
—	[HL+B],[HL+C]（仅F0000H至FFFFFH的空间可指定）
—	ES: [HL+B],ES: [HL+C]（通过ES寄存器指定高4位地址）

图 3-32. [HL+B],[HL+C] 的示例

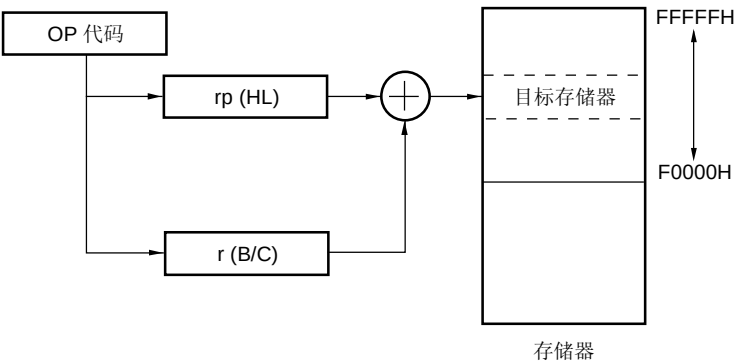
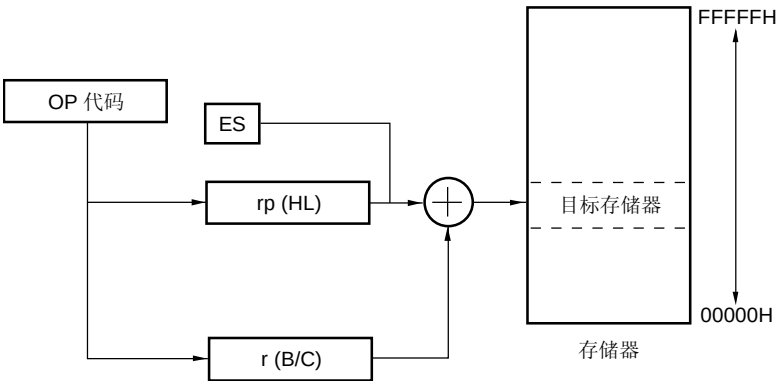


图 3-33. ES: [HL+B],ES: [HL+C] 的示例



3.4.9 堆栈寻址

[功能]

根据堆栈指针(SP)的内容对堆栈区域进行间接寻址。当执行 PUSH、POP、子程序调用和返回指令、或者产生中断请求时保存或恢复寄存器时，将自动采用这种寻址方式。

该堆栈仅应用于内部 RAM 区域。

[操作数格式]

标识符	描述
—	PUSH AX/BC/DE/HL POP AX/BC/DE/HL CALL/CALLT RET BRK RETB (产生中断请求) RETI

第 4 章 端口功能

4.1 端口功能

有 4 种类型的引脚 I/O 缓冲器供电电源：AVREF, EVDD0, EVDD1 和 VDD。下表显示了这些供电电源与引脚之间的关系。

表 4-1. 引脚 I/O 缓冲区供电电源

供电电源	对应的引脚
AVREF	P20 至 P27, P150 至 P157
EVDD0, EVDD1	•端口引脚，除 P20 至 P27, P121 至 P124, 以及 P150 至 P157 以外 •RESET 和 FLMD0 引脚
VDD	•P121 至 P124 •非端口引脚 (不包括 RESET 和 FLMD0 引脚)

78K0R/KG3-L 产品中提供的数字 I/O 端口如图 4-1 所示，这些端口可以支持多种控制操作。每个端口的功能如表 4-2 所示。

除了作为数字 I/O 端口功能，这些端口还有若干复用功能。复用功能的详细信息请见第 2 章 引脚功能。

图 4-1. 端口类型

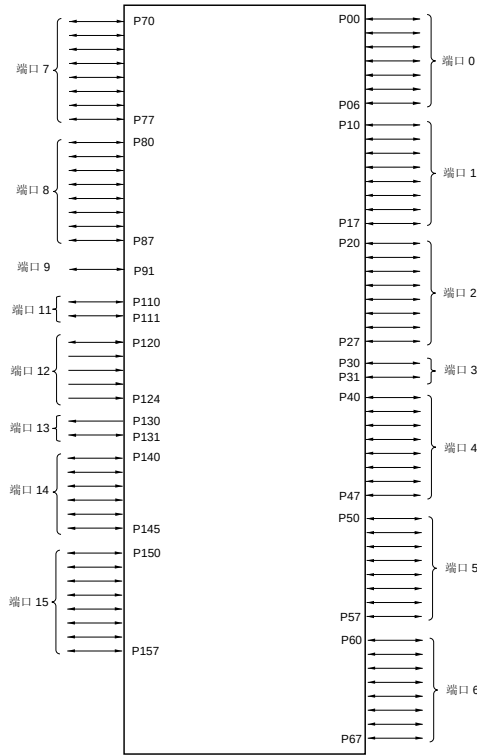


表 4-2. 端口功能 (1/2)

功能名称	I/O	功能	复位后	复用功能
P00	I/O	端口 0。 7 位 I/O 端口。 P03 和 P04 输入可设置成 TTL 输入缓冲器。 P02 至 P04 的输出可设置成 N 沟道漏极开路输出 (V _{DD} 耐压)。 可以指定输入/输出为 1 位单元模式。 通过软件设置, 可以定义片上上拉电阻的使用。	输入端口	TI00
P01				TO00
P02				SO10/TxD1
P03				SI10/RxD1/SDA10
P04				SCK10/SCL10
P05, P06				—
P10	I/O	端口 1。 8 位 I/O 端口。 P10 和 P11 输入可设置成 TTL 输入缓冲器。 P10 和 P12 的输出可设置到 N 沟道漏极开路输出 (V _{DD} 耐压)。 可以指定输入/输出为位单元模式。 通过软件设置, 可以定义片上上拉电阻的使用。	输入端口	SCK00
P11				SI00/RxD0
P12				SO00/TxD0
P13				TxD3
P14				RxD3
P15				RTCDIV/RTCCL
P16				TI01/TO01/INTP5
P17				TI02/TO02
P20 至 P27	I/O	端口 2。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。	数字输入端口	ANI0 至 ANI7
P30	I/O	端口 3。 2 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置, 可以定义片上上拉电阻的使用。	输入端口	RTC1HZ/INTP3
P31				TI03/TO03/INTP4
P40 ^注	I/O	端口 4。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置, 可以定义片上上拉电阻的使用。	输入端口	TOOL0
P41				TOOL1
P42				TI04/TO04
P43				SCK01
P44				SI01
P45				SO01
P46				INTP1/TI05/TO05
P47				INTP2
P50 至 P57	I/O	端口 5。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置, 可以定义片上上拉电阻的使用。	输入端口	—

注 如果通过选项字节允许片上调试, 则确保外部上拉 P40/TOOL0 引脚(参见 2.2.5 P40 至 P47 (端口 4) 中的注意事项)。

表 4-2. 端口功能 (2/2)

功能名称	I/O	功能	复位后	复用功能
P60	I/O	端口 6。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。 仅对 P64 至 P67，通过软件设置，可以定义片上上拉电阻的使用。	输入端口	SCL0
P61				SDA0
P62, P63				—
P64 至 P67				TI10/TO10 至 TI13/TO13
P70 至 P73	I/O	端口 7。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	KR0 至 KR3
P74 至 P77				KR4/INTP8 至 KR7/INTP11
P80 至 P87	I/O	端口 8。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	—
P91	I/O	端口 9。 1 位 I/O 端口。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	—
P110, P111	I/O	端口 11。 2 位 I/O 端口。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	—
P120	I/O	端口 12 1 位 I/O 端口和 4 位输入端口。 仅对 P120，通过软件设置，可以定义片上上拉电阻的使用。	输入端口	INTP0/EXLVI
P121				X1
P122				X2/EXCLK
P123				XT1
P124				XT2
P130	输出	端口 13。 1 位输出端口和位 I/O 端口。	输出端口	—
P131	I/O	仅对 P131，通过软件设置，可以定义片上上拉电阻的使用。	输入端口	TI06/TO06
P140	I/O	端口 14。 6 位 I/O 端口。 P142 和 P143 输入可设置成 TTL 输入缓冲器。 P142 至 P144 的输出可设置成 N 沟道漏极开路输出 (V_{DD} 耐压)。 可以指定输入/输出为位单元模式。 通过软件设置，可以定义片上上拉电阻的使用。	输入端口	PCLBUZ0/INTP6
P141				PCLBUZ1/INTP7
P142				SCK20/SCL20
P143				SI20/RxD2/SDA20
P144				SO20/TxD2
P145				TI07/TO07
P150 至 P157	I/O	端口 15。 8 位 I/O 端口。 可以指定输入/输出为位单元模式。	数字输入端口	AN18 至 AN15

4.2 端口配置

端口包括如下硬件。

表 4-3. 端口配置

项目	配置
控制寄存器	端口模式寄存器(PM0 至 PM9,PM11 至 PM15) 端口寄存器(P0 至 P9,P11 至 P15) 上拉电阻选项寄存器(PU0,PU1,PU3 至 PU9,PU11 至 PU14) 端口输入模式寄存器(PIM0,PIM1,PIM14) 端口输出模式寄存器(POM0,POM1,POM14) A/D 端口配置 寄存器(ADPC)
端口	总计: 89(CMOS I/O: 80,CMOS 输入: 4,CMOS 输出: 1,N 沟道漏极开路 I/O: 4)
上拉电阻	总计: 64

4.2.1 端口 0

端口 0 是具有输出锁存功能的 7 位 I/O 端口。通过使用端口模式寄存器 0(PM0)，可以以 1 位单元设置端口 0 为输入或输出模式。当 P0 至 P06 引脚用作输入端口时，通过上拉电阻选项寄存器 0(PU0)，可以以位单元指定片内上拉电阻的使用。

使用端口输入模式寄存器 0(PIM0)，可以通过一个常规输入缓冲器或 TTL 输入缓冲器以位单元指定 P03 和 P04 引脚的输入。

通过端口输出模式寄存器 0(POM0)，可以以位单元指定 P02 至 P04 引脚的输出为 N 沟道漏极开路输出(V_{DD} 耐压)。

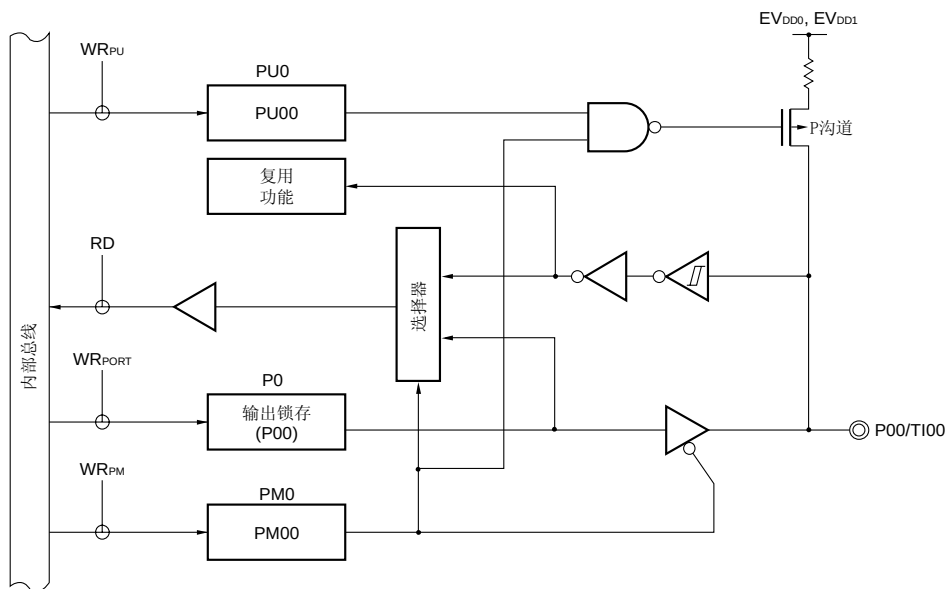
此端口也可用作定时器 I/O、串行接口数据 I/O 和时钟 I/O。

复位信号发生将端口 0 设置为输入模式。

图 4-2 至 4-6 所示为端口 0 的框图。

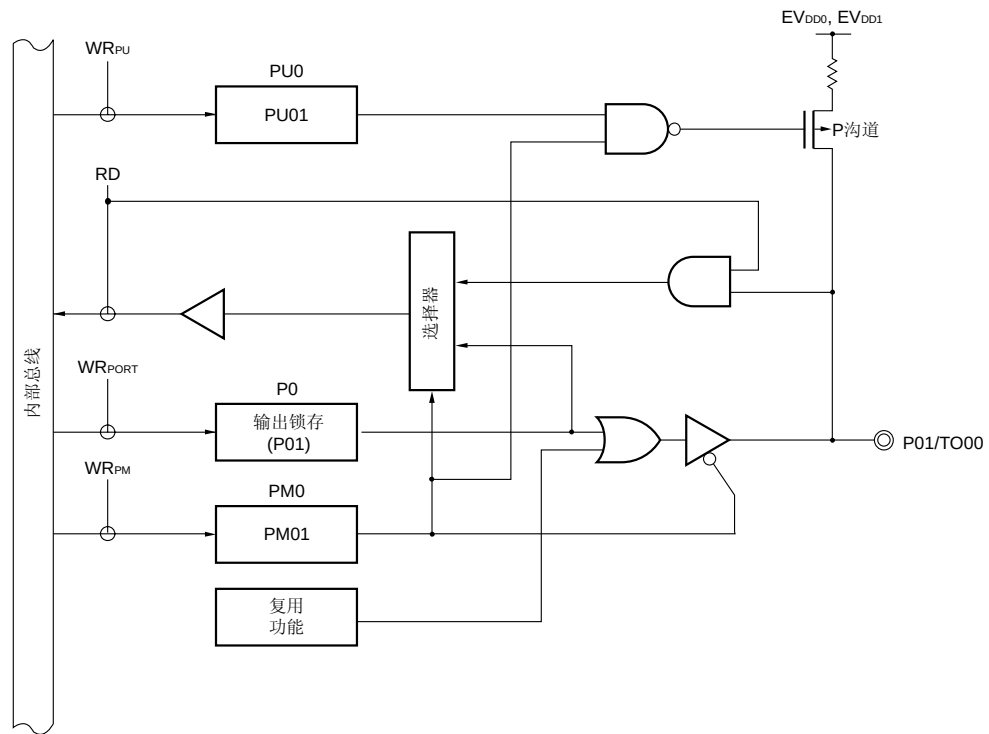
- 注意事项**
1. P01/TO00 作为通用端口时，应设置定时器输出寄存器 0(TO0)的位 0(TO00)以及定时器输出使能寄存器 0(TOE0)的位 0(TOE00)为“0”，与它们的默认状态设置相同。
 2. 要将 P02/SO10/TxD1、P03/SI10/RxD1/SDA10 或 P04/SCK10/SCL10 作为通用端口使用，注意设置串行阵列单元 0。详情参见下列表格。
 - 表 11-7 寄存器设置和引脚之间的关系(单元 0 的通道 2: CSI10, UART1 发送, IIC10)
 - 表 11-8 寄存器设置和引脚之间的关系(单元 0 的通道 3: UART1 接收)

图 4-2. P00 的框图



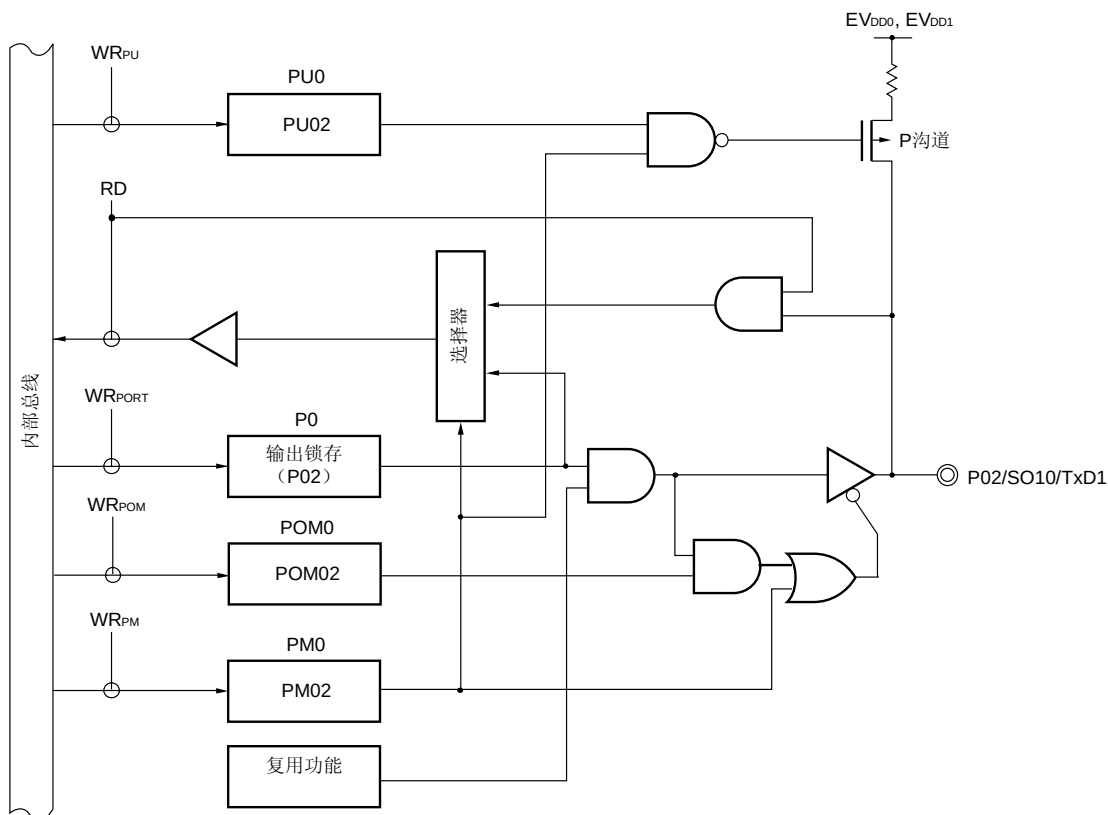
P0: 端口寄存器 0
 PU0: 上拉电阻选项寄存器 0
 PM0: 端口模式寄存器 0
 RD: 读信号
 WRxx: 写信号

图 4-3. P01 的框图



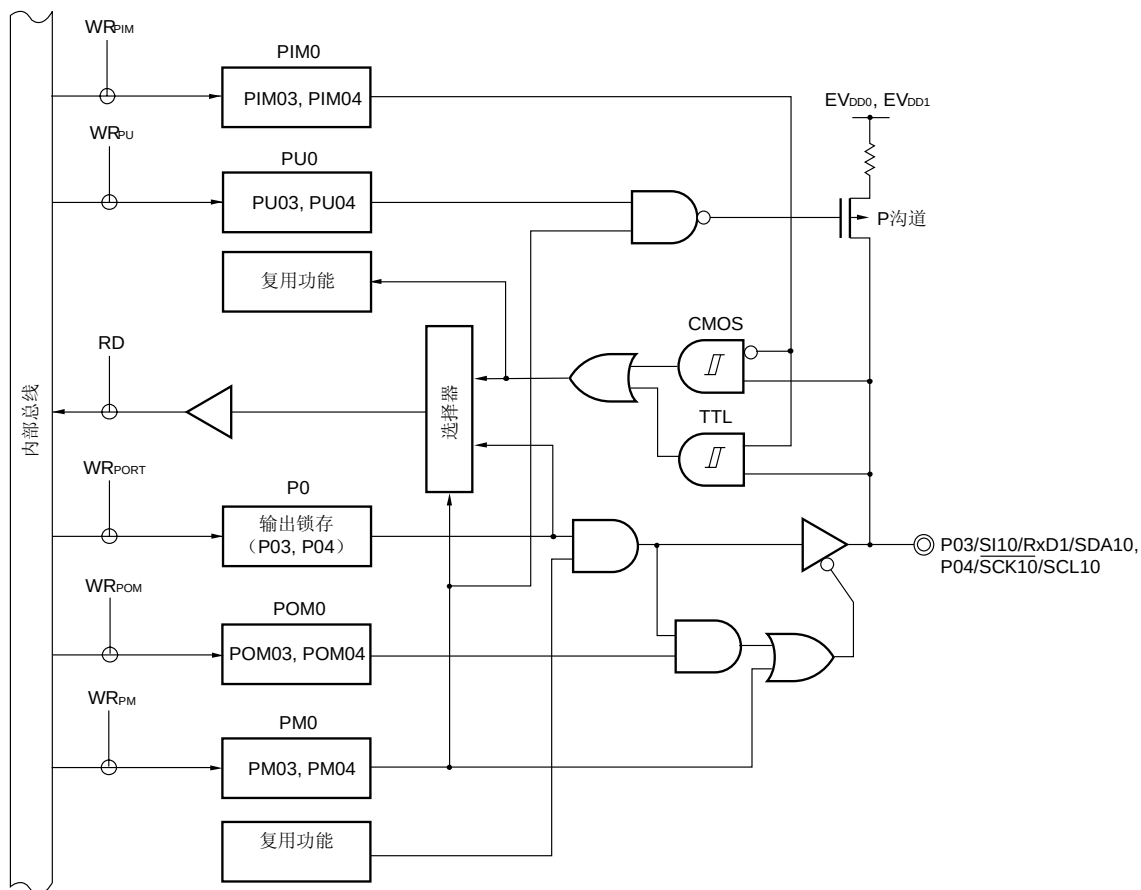
P0: 端口寄存器 0
 PU0: 上拉电阻选项寄存器 0
 PM0: 端口模式寄存器 0
 RD: 读信号
 WR_{xx}: 写信号

图 4-4. P02 的框图



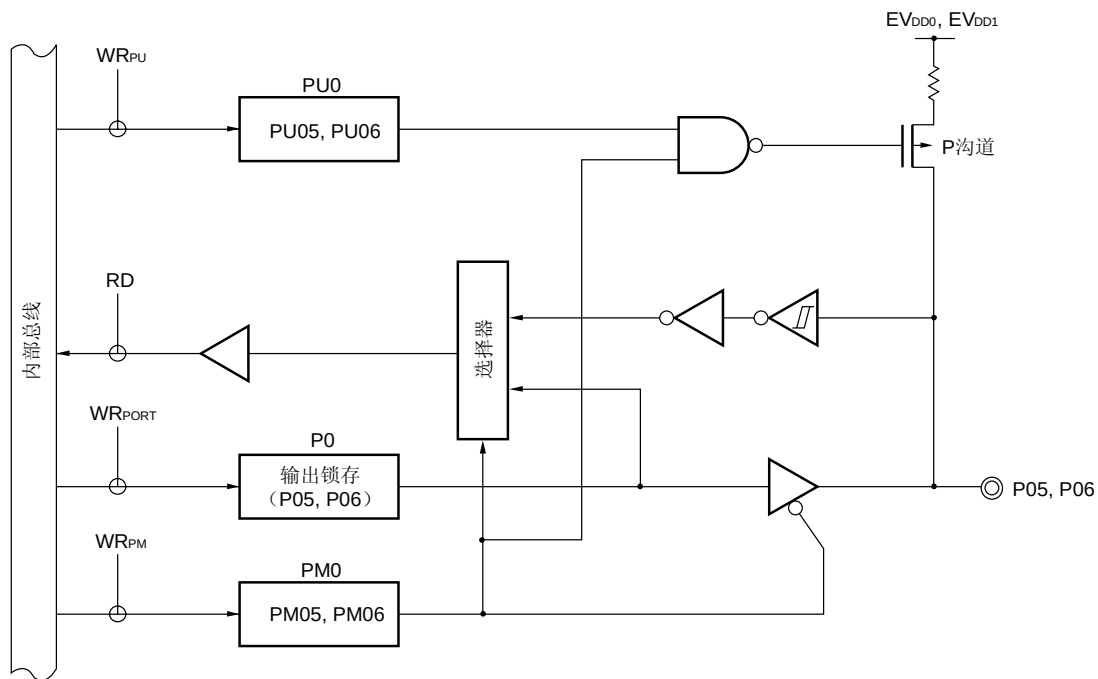
P0: 端口寄存器 0
 PU0: 上拉电阻选项寄存器 0
 PM0: 端口模式寄存器 0
 POM0: 端口输出模式寄存器 0
 RD: 读信号
 WR_{xx}: 写信号

图 4-5. P03 和 P04 的框图



P0: 端口寄存器 0
 PU0: 上拉电阻选项寄存器 0
 PM0: 端口模式寄存器 0
 PIM0: 端口输入模式寄存器 0
 POM0: 端口输出模式寄存器 0
 RD: 读信号
 WR_{xx}: 写信号

图 4-6. P05 和 P06 的框图



P0: 端口寄存器 0
 PU0: 上拉电阻选项寄存器 0
 PM0: 端口模式寄存器 0
 RD: 读信号
 WR_{xx}: 写信号

4.2.2 端口 1

端口 1 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 1(PM1)，可以以位单元设置端口 1 为输入或输出模式。当 P10 和 P17 引脚用作输入端口时，通过上拉电阻选项寄存器 1(PU1)，可以以位单元指定片上上拉电阻的使用。

使用端口输入模式寄存器 1(PIM1)，可以通过一个常规输入缓冲器或 TTL 输入缓冲器以位单元指定 P10 和 P11 引脚的输入。

通过端口输出模式寄存器 1(POM1)，可以以位单元指定 P10 至 P12 引脚的输出为 N 沟道漏极开路输出(V_{DD} 耐压)。

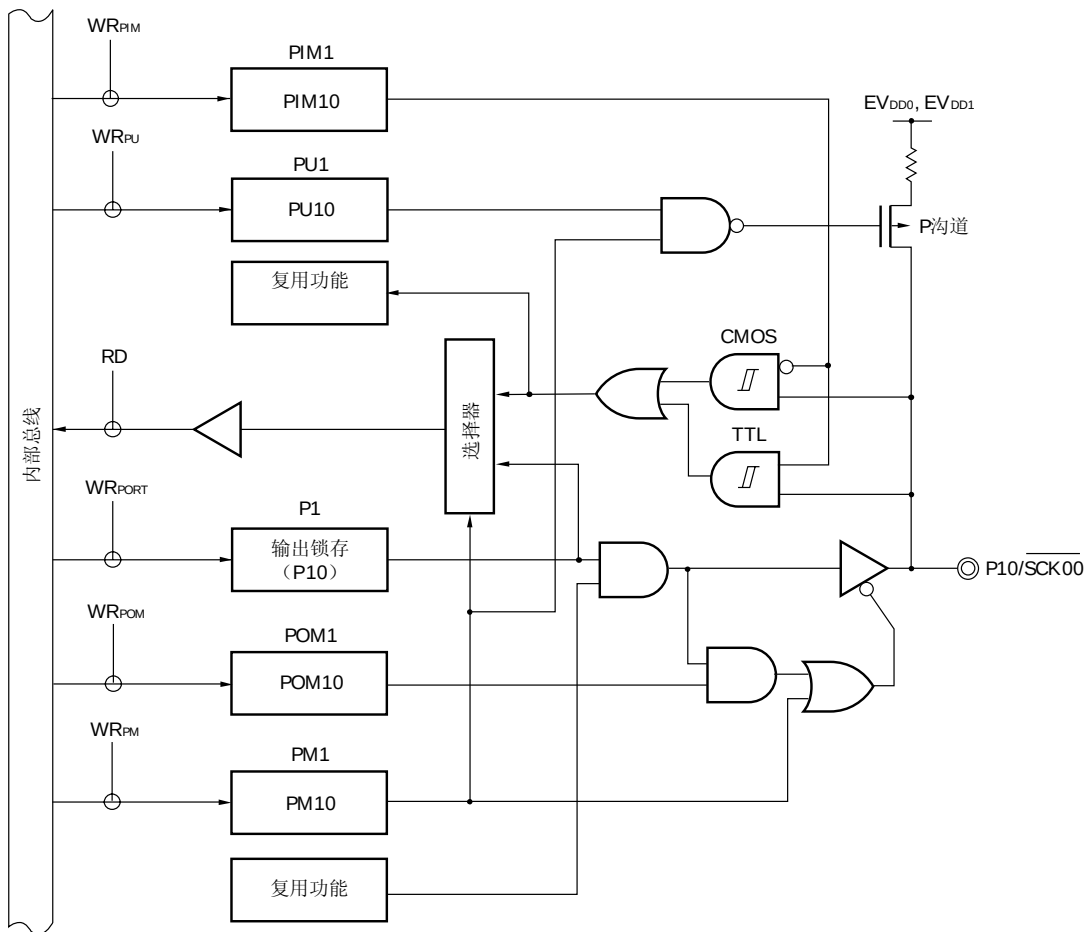
该端口也可作为外部中断请求输入、串行接口数据 I/O、时钟 I/O、定时器 I/O 和实时计数器时钟功能使用。

复位信号发生将端口 1 设置为输入模式。

图 4-7 至 4-13 所示为端口 1 的框图。

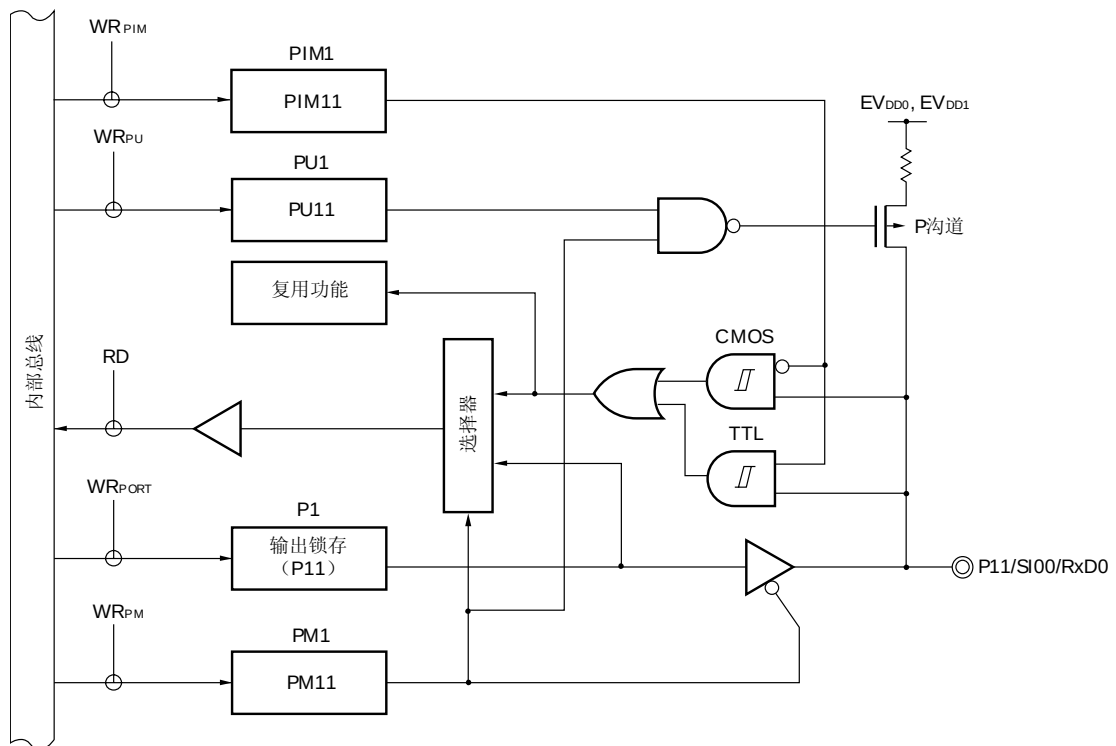
- 注意事项**
1. 要使用 P10/ $\overline{SCK00}$, P11/SI00/RxD0 或者 P12/SO00/TxD0, P13/TxD3, P14/RxD3 作为通用端口，注意串行阵列单元的设置。详情参见下列表格。
 - 表 11-5 寄存器设置和引脚之间的关系(单元 0 的通道 0: CSI00, UART0 发送)
 - 表 11-6 寄存器设置和引脚之间的关系(单元 0 的通道 1: CSI01, UART0 接收)
 - 表 11-11 寄存器设置和引脚之间的关系(单元 1 的通道 2: UART3 发送)
 - 表 11-12 寄存器设置和引脚之间的关系(单元 1 的通道 3: UART3 接收)
 2. 将 P16/TI01/TO01/INTP5 或 P17/TI02/TO02 作为通用端口时，设置定时器输出寄存器 0(TO0)的位 1 和 2(TO01, TO02)以及定时器输出使能寄存器 0(TOE0)的位 1 和 2(TOE01, TOE02)为“0”，同它们的默认状态设置相同。
 3. P15/RTCDIV/RTCCL 作为通用端口时，应设置实时计数器控制寄存器 0(RTCC0)的位 4(RCLOE0)以及实时计数器控制寄存器 2(RTCC2)的位 6(RCLOE2)为“0”与它们的默认状态设置相同。

图 4-7. P10 的框图



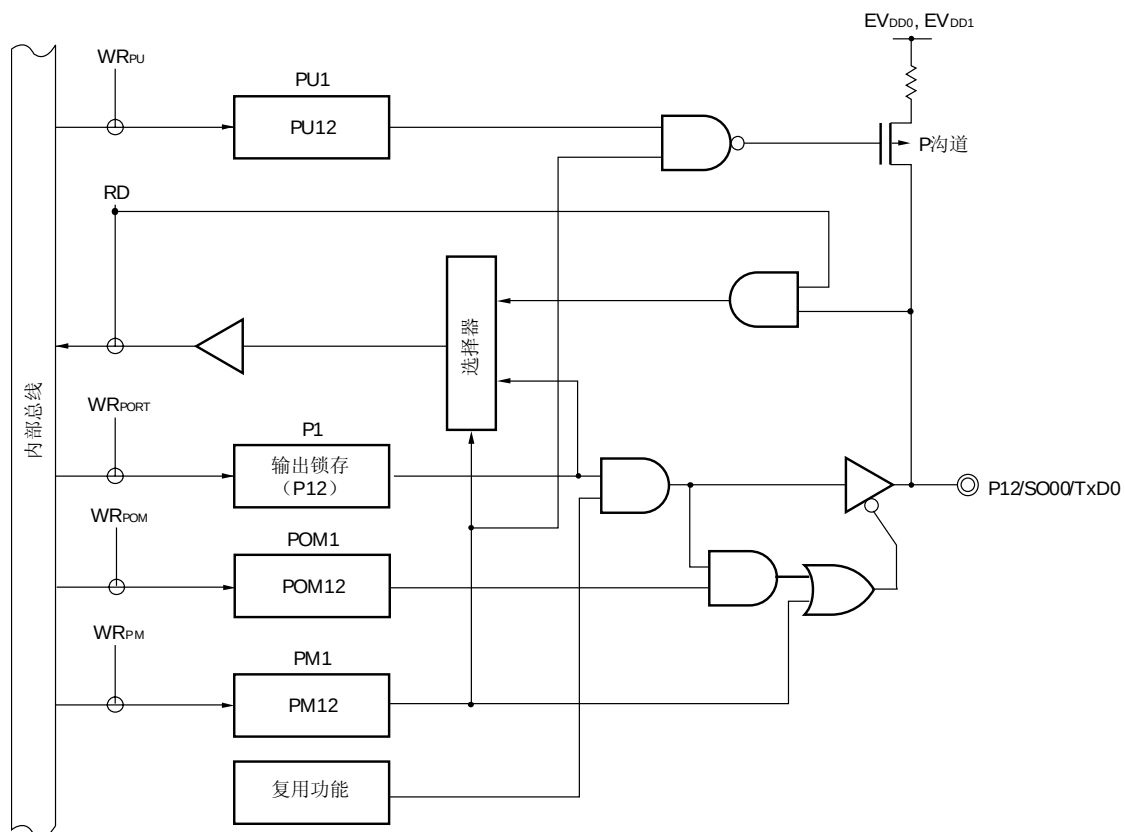
P1: 端口寄存器 1
 PU1: 上拉电阻选项寄存器 1
 PM1: 端口模式寄存器 1
 PIM1: 端口输入模式寄存器 1
 POM1: 端口输出模式寄存器 1
 RD: 读信号
 WR_{xx}: 写信号

图 4-8. P11 的框图



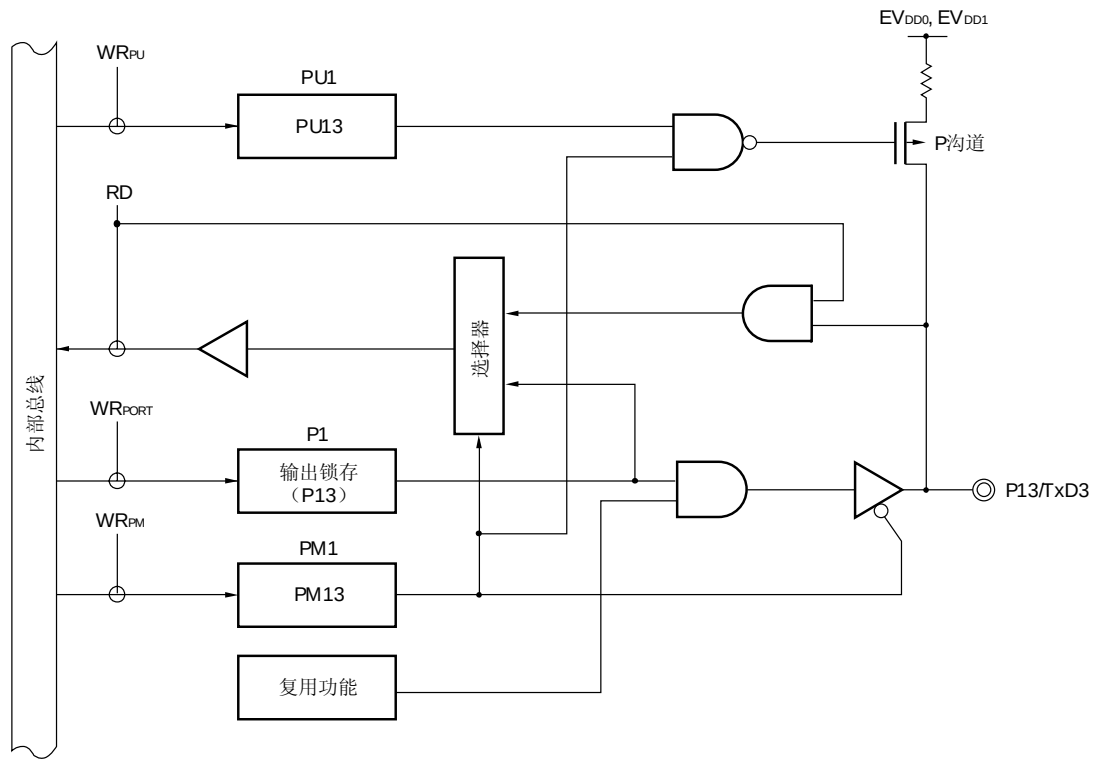
P1: 端口寄存器 1
 PU1: 上拉电阻选项寄存器 1
 PM1: 端口模式寄存器 1
 PIM1: 端口输入模式寄存器 1
 RD: 读信号
 WRxx: 写信号

图 4-9 P12 的框图



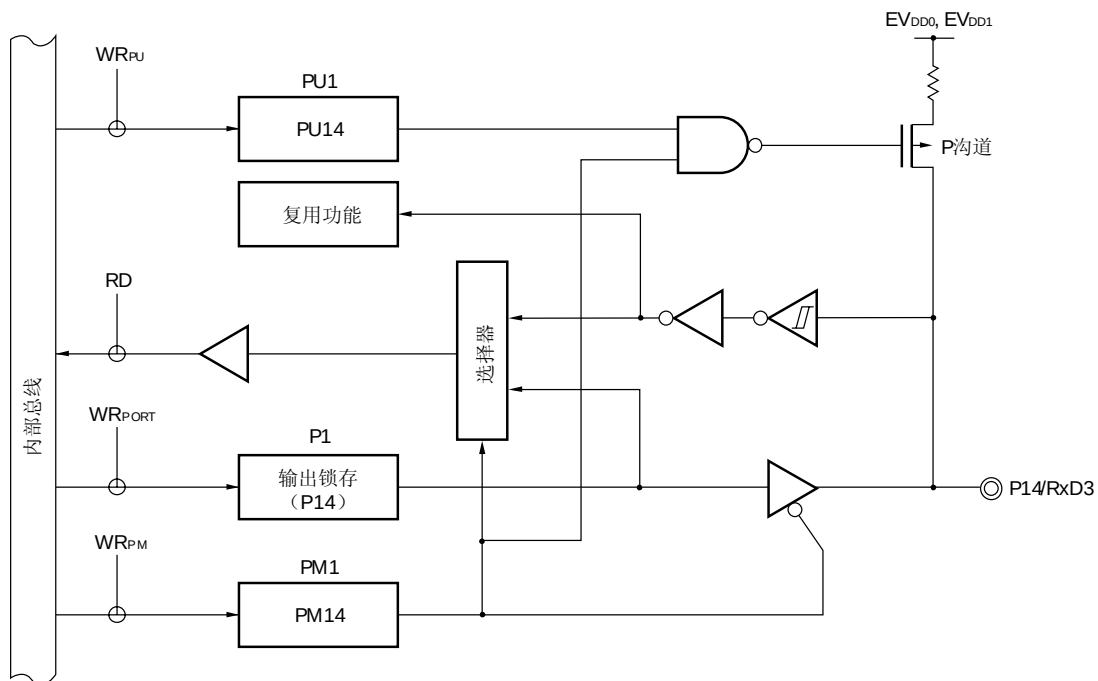
P1:	端口寄存器 1
PU1:	上拉电阻选项寄存器 1
PM1:	端口模式寄存器 1
POM1:	端口输出模式寄存器 1
RD:	读信号
WRxx:	写信号

图 4-10. P13 的框图



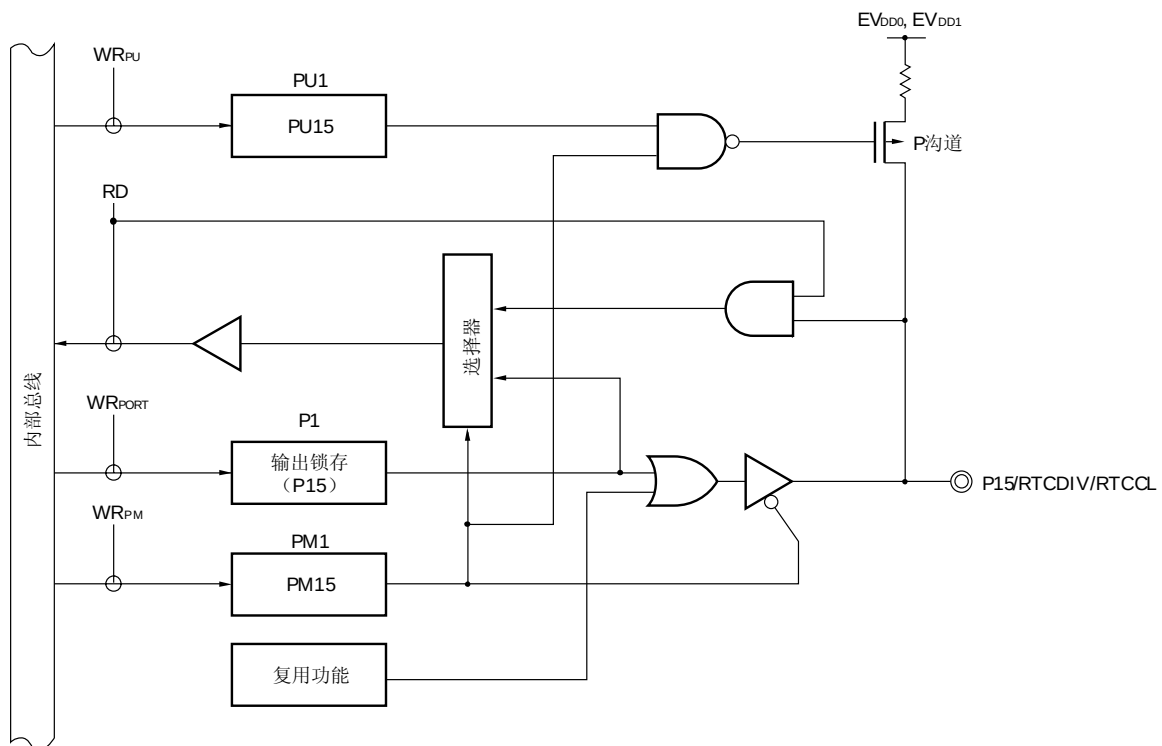
- P1: 端口寄存器 1
 PU1: 上拉电阻选项寄存器 1
 PM1: 端口模式寄存器 1
 RD: 读信号
 WR_{xx}: 写信号

图 4-11. P14 的框图



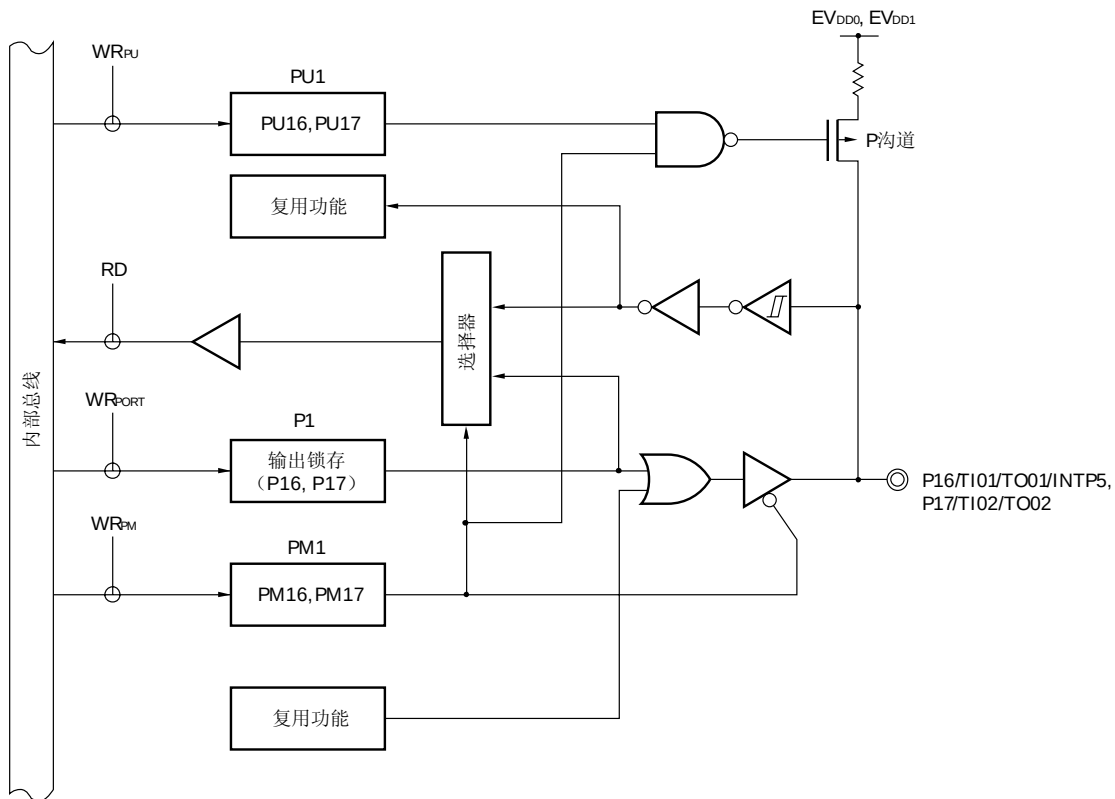
P1: 端口寄存器 1
 PU1: 上拉电阻选项寄存器 1
 PM1: 端口模式寄存器 1
 RD: 读信号
 WR_{xx}: 写信号

图 4-12. P15 的框图



- P1: 端口寄存器 1
 PU1: 上拉电阻选项寄存器 1
 PM1: 端口模式寄存器 1
 RD: 读信号
 WRxx: 写信号

图 4-13. P16, P17 的框图



P1: 端口寄存器 1
 PU1: 上拉电阻选项寄存器 1
 PM1: 端口模式寄存器 1
 RD: 读信号
 WR $\times\times$: 写信号

4.2.3 端口 2

端口 2 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 2(PM2)，可以以位单元设置端口 2 为输入或输出模式。

此端口也可用作 A/D 转换器模拟输入。

如果要将 P20/ANI0 至 P27/ANI7 作为数字输入引脚，应通过使用 A/D 端口配置寄存器(ADPC)将这些引脚设置为数字 I/O 模式，并通过使用 PM2 将这些引脚设置为输入模式。从低位开始使用这些引脚。

如果要将 P20/ANI0 至 P27/ANI7 作为数字输出引脚，应通过使用 ADPC 将这些引脚设置为数字 I/O 模式，并通过使用 PM2 将这些引脚设置成输入模式。

如果要将 P20/ANI0 至 P27/ANI7 作为模拟输入引脚，应通过 A/D 端口配置寄存器(ADPC) 将这些引脚设置为模拟输入模式，并通过使用 PM2 将这些引脚设置为输入模式。从高位开始使用这些引脚。

表 4-4. P20/ANI0 至 P27/ANI7 引脚的设置功能

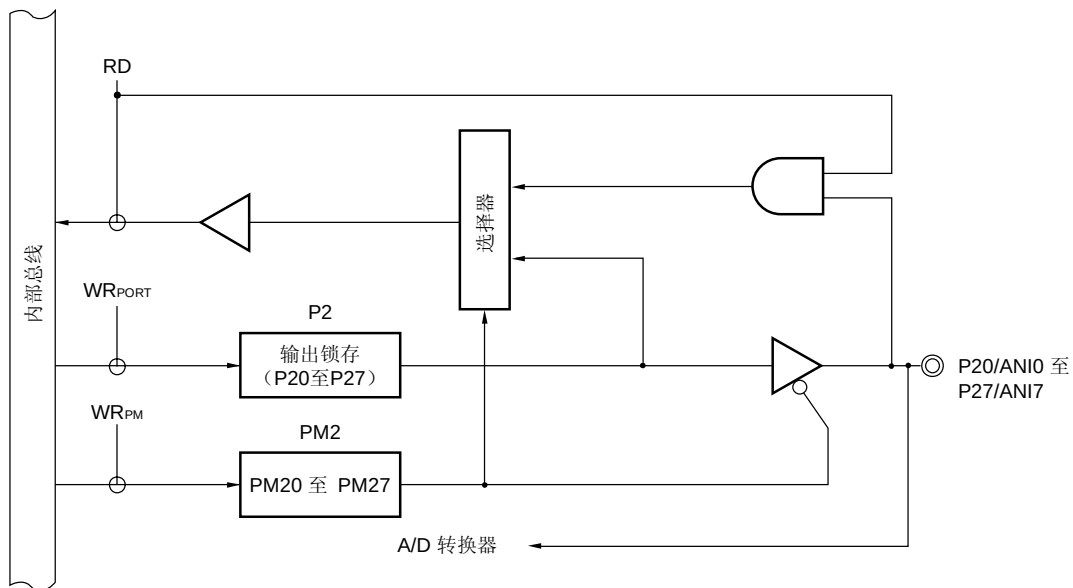
ADPC	PM2	ADS	P20/ANI0至P27/ANI7引脚
数字I/O 选择	输入模式	–	数字输入
	输出模式	–	数字输出
模拟输入选择	输入模式	选择ANI。	模拟输入(转换)
		不要选择ANI。	模拟输入(不转换)
	输出模式	选择ANI。	设置禁止
		不要选择ANI。	

当复位信号发生时，所有 P20/ANI0 至 P27/ANI7 均设置为数字输入模式。

图 4-14 所示为端口 2 的框图。

注意事项 当使用端口 2 作为数字 I/O 时，参见 2.2.16 应用的 AVREF 引脚的电压 AVREF 。

图 4-14. P20 至 P27 的框图



P2: 端口寄存器 2
 PM2: 端口模式寄存器 2
 RD: 读信号
 WR_{xx}: 写信号

4.2.4 端口 3

端口 3 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 3(PM3)，可以以 1 位单元设置端口 3 为输入或输出模式。如果 P30 和 P31 用于输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 3(PU3)按位指定。

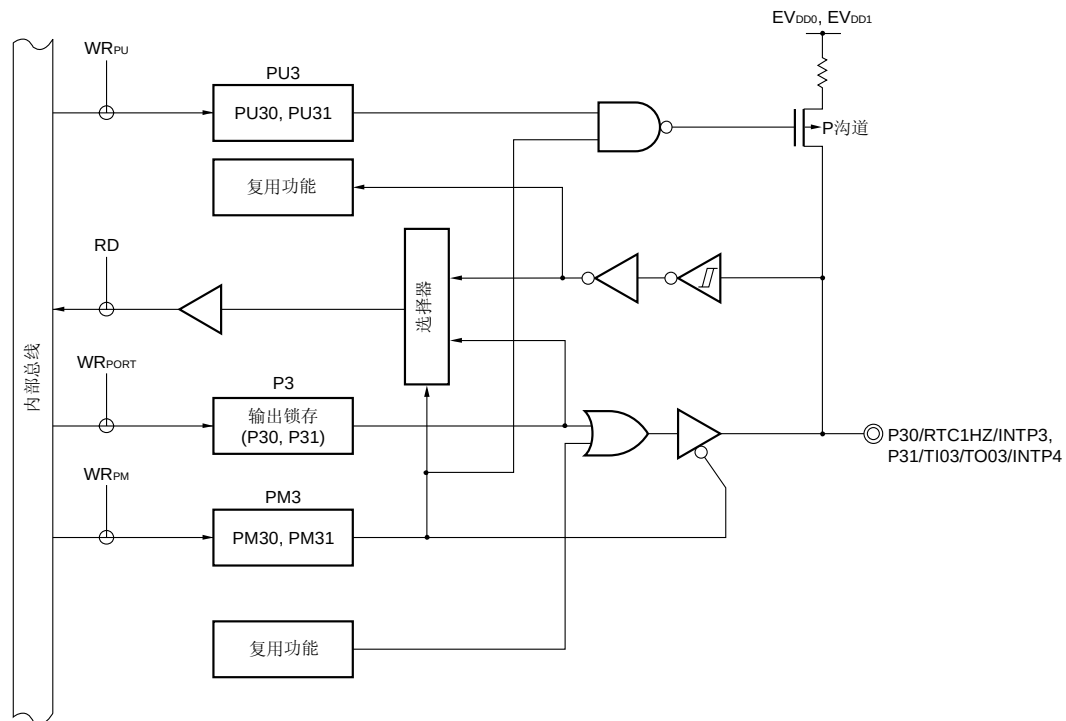
此端口也可用于外部中断请求输入，定时器 I/O 和实时计数器修正时钟输出。

复位信号发生将端口 3 设置为输入模式。

图 4-15 所示为端口 3 的框图。

- 注意事项**
1. P31/TI03/TO03/INTP4 作为通用端口时，应设置定时器输出寄存器 0(TO0)的位 3(TO03) 和定时器输出使能寄存器 0(TOE0)的位 3(TOE03) 置“0”，和它们的默认状态设置相同。
 2. P30/RTC1HZ/INTP3 作为通用端口时，应设置实时计数器控制寄存器 0(RTCC0)的位 5(RCLOE1)为“0”，与其默认状态设定一致。

图 4-15. P30 和 P31 的框图



P3: 端口寄存器 3

PU3: 上拉电阻选项寄存器 3

PM3: 端口模式寄存器 3

RD: 读信号

WRxx: 写信号

4.2.5 端口 4

端口 4 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 4(PM4)，可以以位单元设置端口 4 为输入或输出模式。当 P40 至 P47 引脚用作输入端口，则通过上拉电阻选择寄存器 4(PU4)^注，片内上拉电阻的使用可以以位单元指定。

该端口也可作为外部中断请求输入，串行接口数据 I/O，时钟 I/O，Flash 存储器编程器/调试器数据 I/O，时钟输出，以及定时器 I/O。

复位信号发生将端口 4 设置为输入模式。

图 4-16 至 4-21 所示为端口 4 的框图。

注 连接工具时，P40 和 P41 引脚不能连接上拉电阻。

注意事项 1. 当连接工具时，P40 引脚不能作为端口引脚使用。

使用片上调试功能时，P41 引脚通过模式设置可用于如下调试。

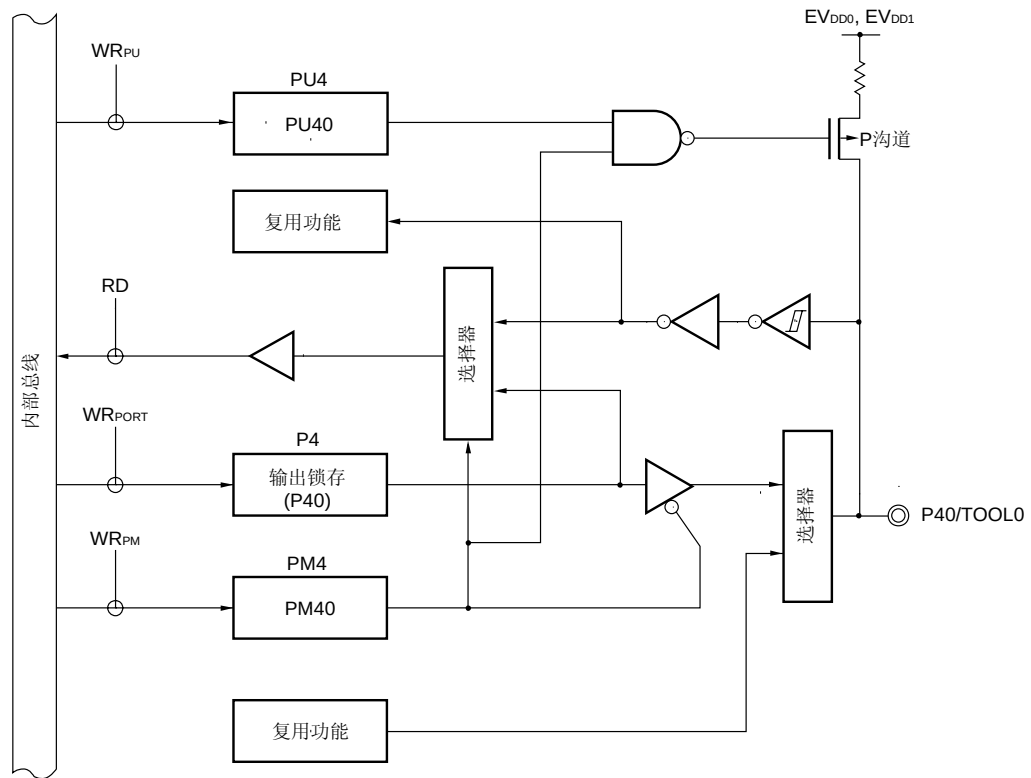
1 线模式：可用作端口(P41)。

2 线模式：可用作 TOOL1 引脚，不能作为端口(P41)使用。

2. 要使用 P43/SCK01、P44/SI01 或者 P45/SO01 作为通用端口时，注意设置串行阵列单元 0。详情参见表 11-6 寄存器设置和引脚之间的关系(单元 0 的通道 1: CSI01, UART0 接收)。

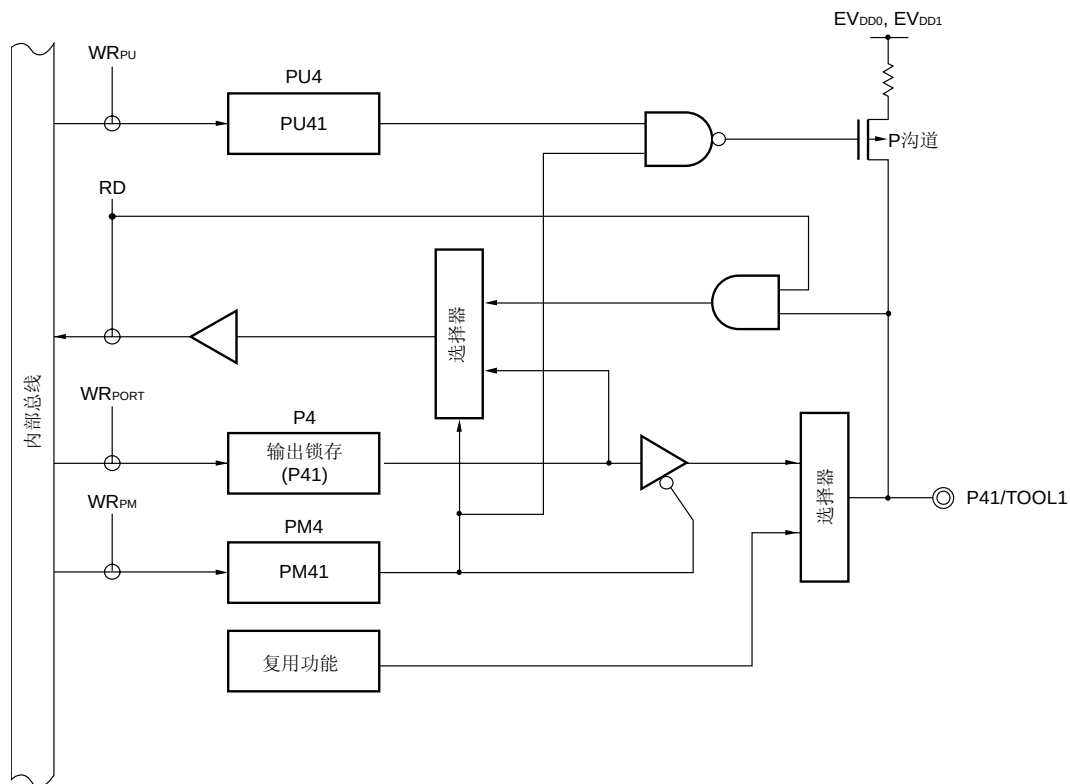
3. P42/TI04/TO04 或 P46/INTP1/TI05/TO05 作为通用端口时，应设置定时器输出寄存器 0(TO0) 的位 4 和 5(TO04, TO05)以及定时器输出使能寄存器 0(TOE0)位 4 和 5(TOE04, TOE05)置“0”，和它们的初始状态设置相同。

图 4-16. P40 的框图



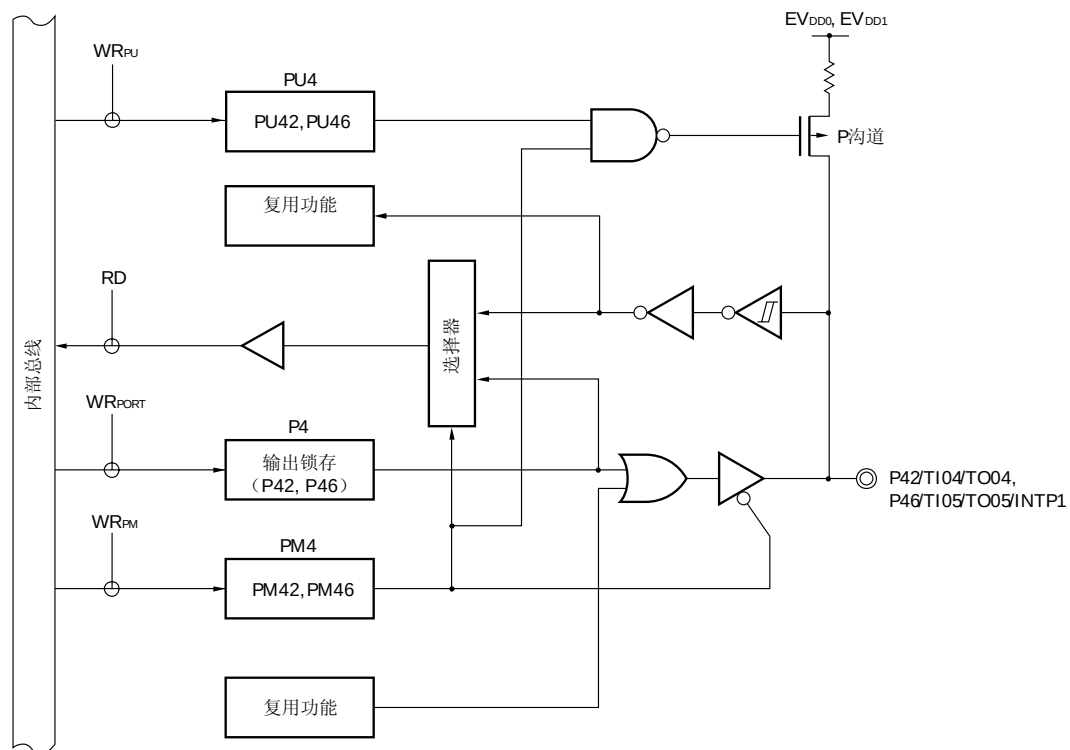
- P4: 端口寄存器 4
 PU4: 上拉电阻选项寄存器 4
 PM4: 端口模式寄存器 4
 RD: 读信号
 WR_{xx}: 写信号

图 4-17. P41 的框图



P4: 端口寄存器 4
 PU4: 上拉电阻选项寄存器 4
 PM4: 端口模式寄存器 4
 RD: 读信号
 WR_{xx}: 写信号

图 4-18. P42 和 P46 的框图



P4: 端口寄存器 4

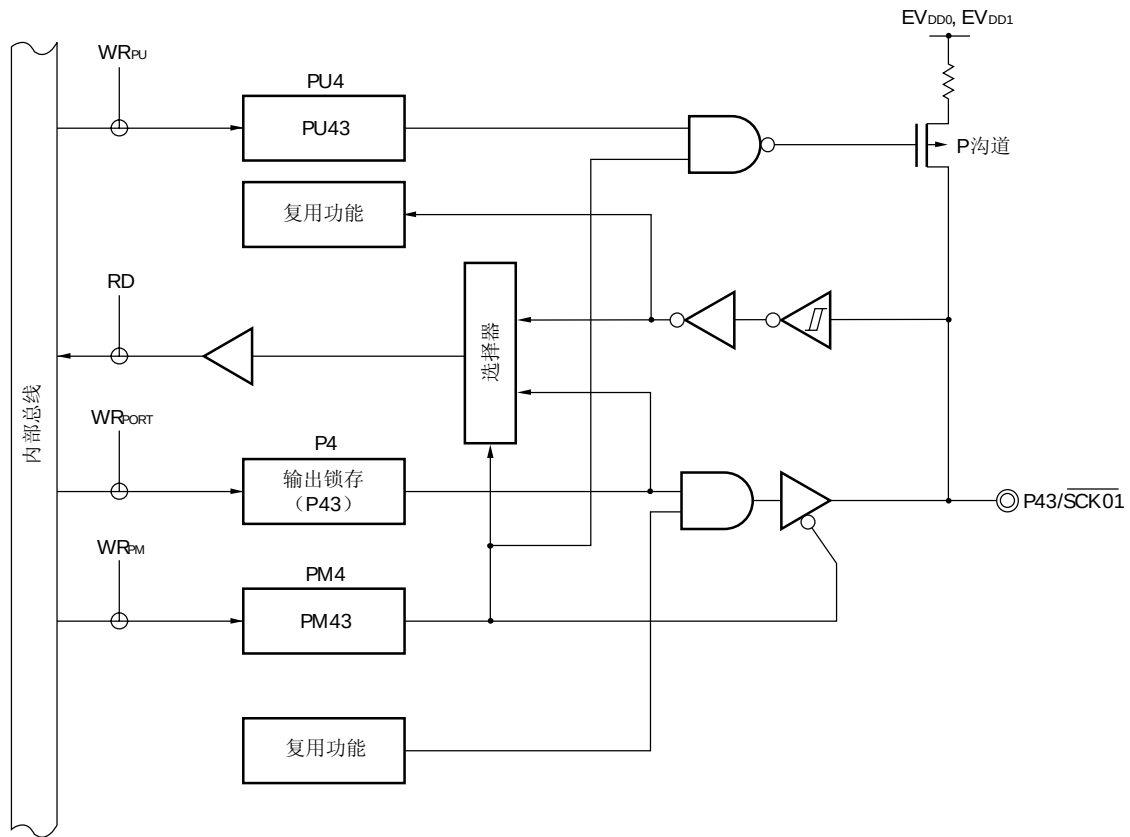
PU4: 上拉电阻选项寄存器 4

PM4: 端口模式寄存器 4

RD: 读信号

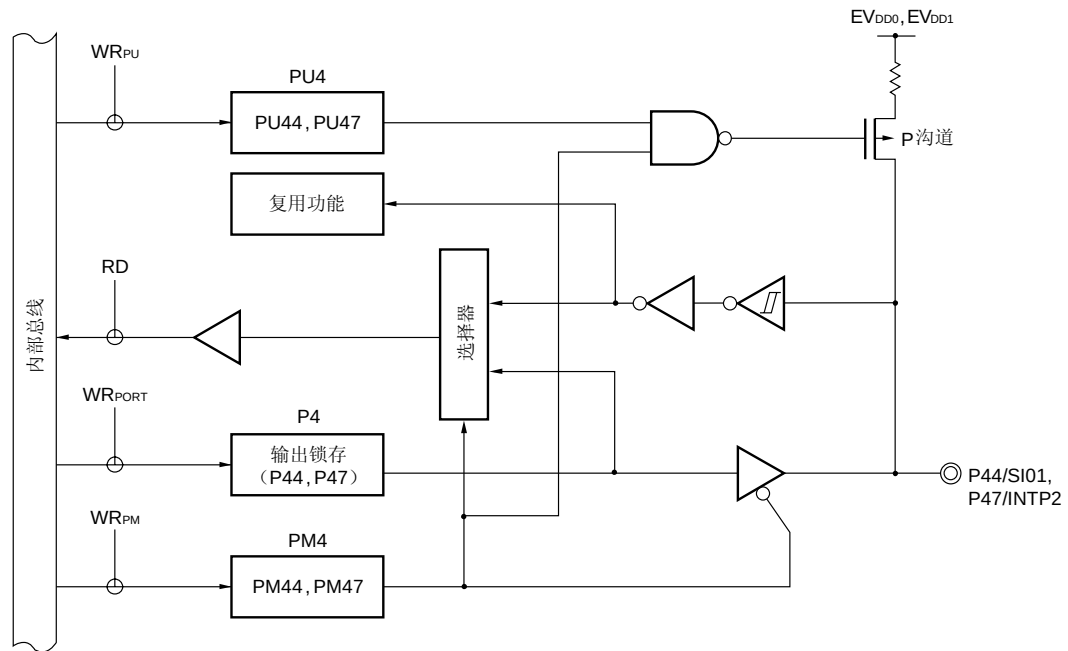
WR_{xx}: 写信号

图 4-19. P43 的框图



P4: 端口寄存器 4
 PU4: 上拉电阻选项寄存器 4
 PM4: 端口模式寄存器 4
 RD: 读信号
 WR $\times$: 写信号

图 4-20. P44 和 P47 的框图



P4: 端口寄存器 4
 PU4: 上拉电阻选项寄存器 4
 PM4: 端口模式寄存器 4
 RD: 读信号
 WR_{xx}: 写信号

PU4: 上拉电阻选项寄存器 4

PM4: 端口模式寄存器 4

RD: 读信号

WRxx: 写信号

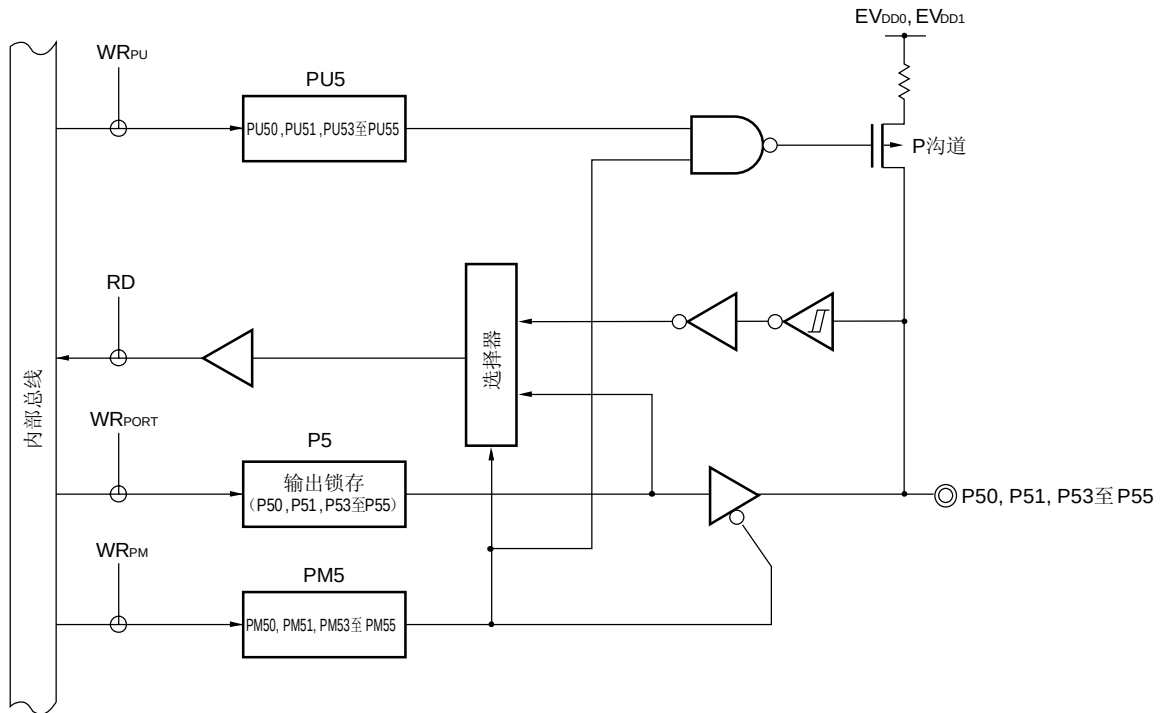
4.2.6 端口 5

端口 5 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 5(PM5)，可以以 1 位单元设置端口 5 为输入或输出模式。当 P50 至 P57 作为输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 5(PU5)以位单元的方式指定。

复位信号发生将端口 5 设置为输入模式。

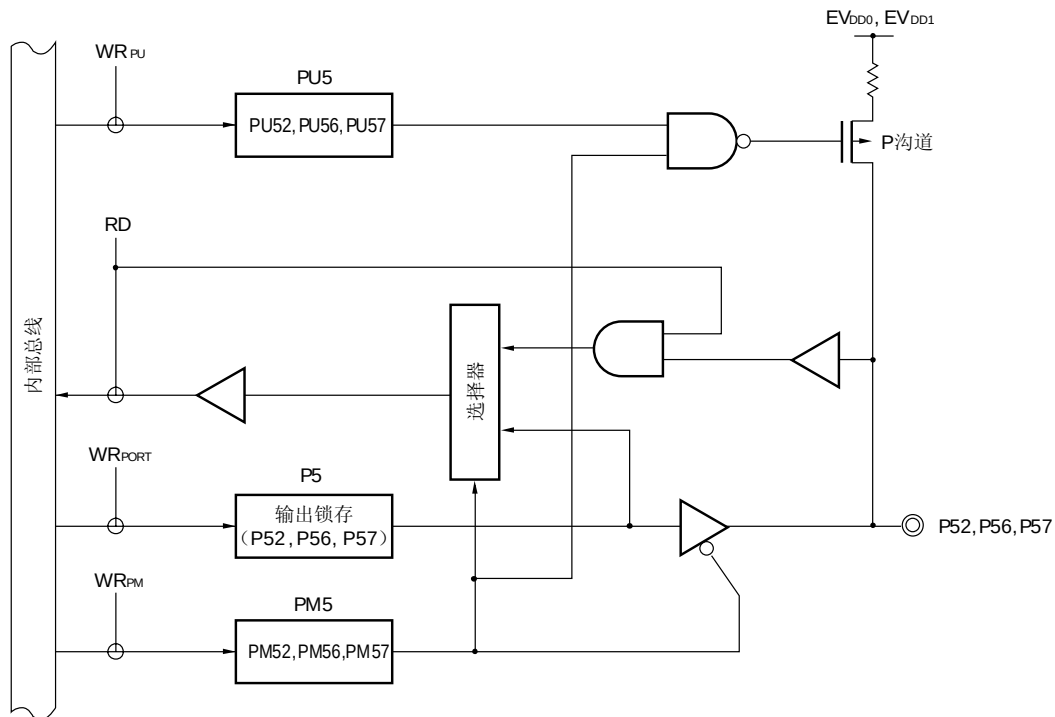
图 4-22 和 4-23 所示为端口 5 的框图。

图 4-22. P50 和 P51, P53 至 P55 的框图



- P5: 端口寄存器 5
 PU5: 上拉电阻选项寄存器 5
 PM5: 端口模式寄存器 5
 RD: 读信号
 WR_{xx}: 写信号

图 4-23. P52, P56, P57 的框图



P5: 端口寄存器 5
 PU5: 上拉电阻选项寄存器 5
 PM5: 端口模式寄存器 5
 RD: 读信号
 WR_{xx}: 写信号

4.2.7 端口 6

端口 6 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 6(PM6)，可以以 1 位单元设置端口 6 为输入或输出模式。当 P64 至 P67 用于输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 6(PU6)按位指定。

P60 至 P63 引脚的输出是 N 沟道漏极开路输出(6 V 耐压)。

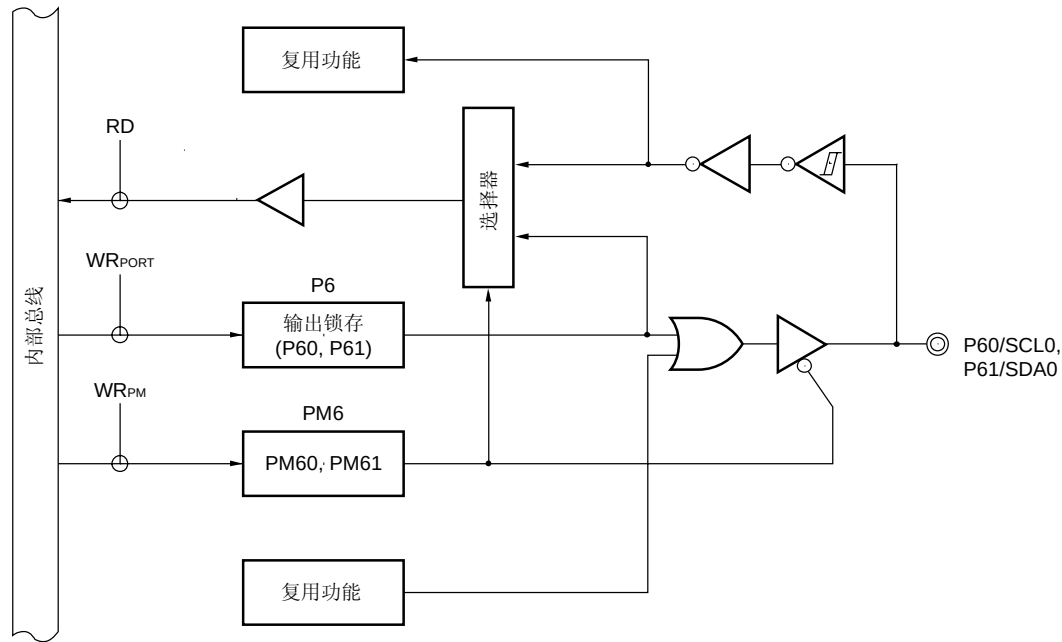
此端口也可用作串行接口数据 I/O、时钟 I/O 和定时器 I/O。

复位信号发生将端口 6 设置为输入模式。

图 4-24 至 4-26 所示为端口 6 的框图。

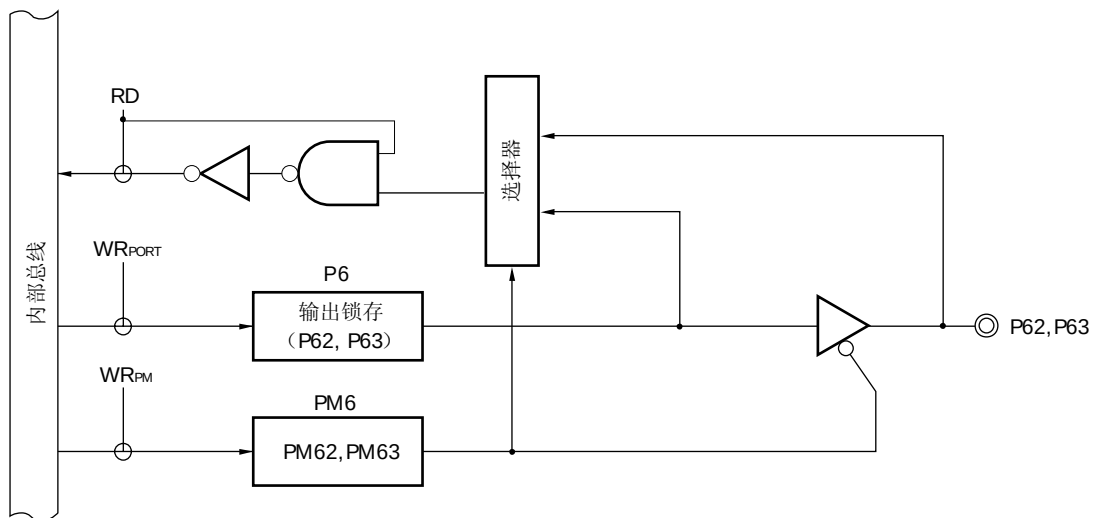
- 注意事项
- 1. 当使用 P60/SCL0 和 P61/SDA0 作为通用端口时，停止串行接口 IICA 的操作。
 - 2. P64/TI10/TO10 至 P67/TI13/TO13 作为通用端口时，应设置定时器输出寄存器 1(TO1)的位 0 至位 3(TO10 至 TO13)以及定时器输出使能寄存器 1(TOE1)的位 0 至位 3(TOE10 至 TOE13)为“0”，与它们的默认状态设置相同。

图 4-24. P60 和 P61 的框图



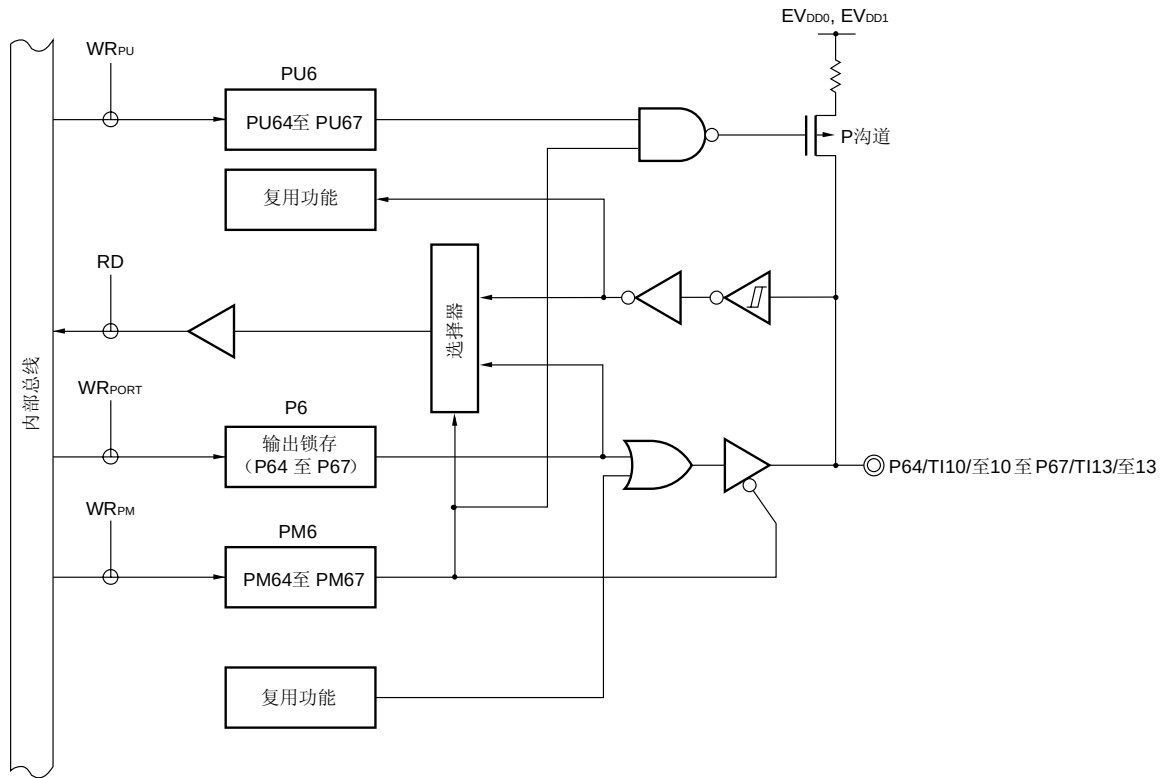
P6: 端口寄存器 6
PM6: 端口模式寄存器 6
RD: 读信号
WRxx: 写信号

图 4-25. P62, P63 的框图



P6: 端口寄存器 6
 PM6: 端口模式寄存器 6
 RD: 读信号
 WR_{xx}: 写信号

图 4-26. P64 至 P67 的框图



P6: 端口寄存器 6
 PU6: 上拉电阻选项寄存器 6
 PM6: 端口模式寄存器 6
 RD: 读信号
 WR $\times\times$: 写信号

4.2.8 端口 7

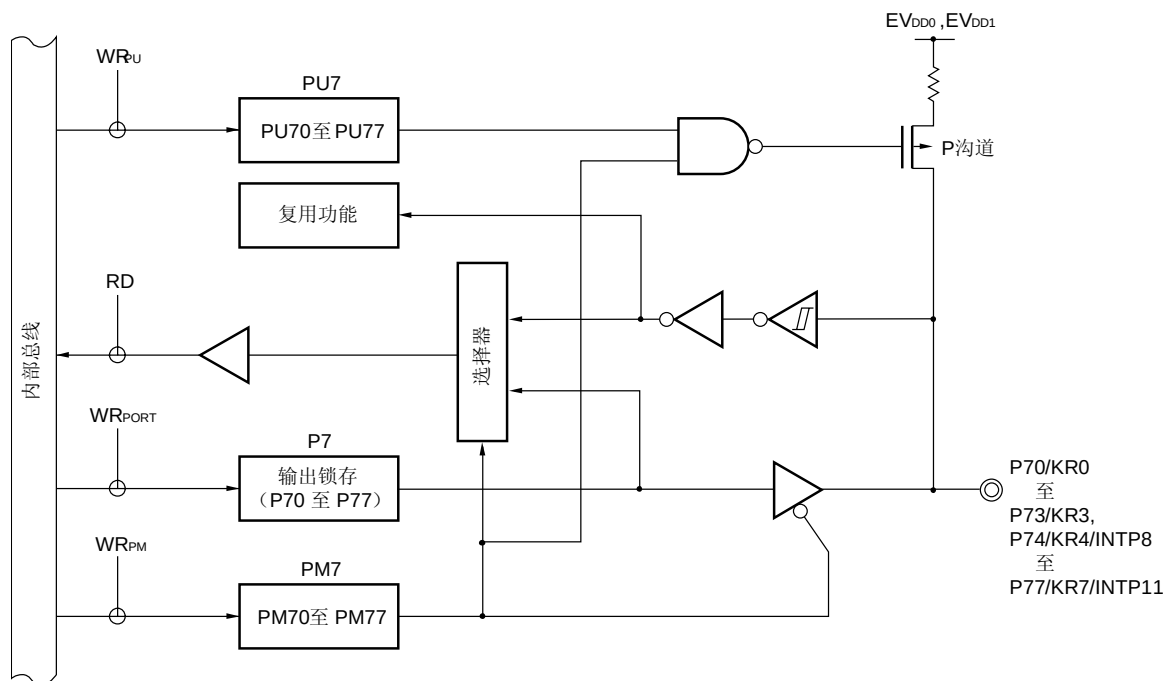
端口 7 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 7(PM7)，可以以位单元设置端口 7 为输入或输出模式。当用作输入端口时，通过上拉电阻选择寄存器 7(PU7)，片上上拉电阻的使用可以以位单元指定。

该端口也可作为按键返回输入和中断请求输入。

复位信号发生将端口 7 设置为输入模式。

图 4-27 所示为端口 7 的框图。

图 4-27. P70 至 P77 的框图



- P7: 端口寄存器 7
 PU7: 上拉电阻选项寄存器 7
 PM7: 端口模式寄存器 7
 RD: 读信号
 WR_{xx}: 写信号

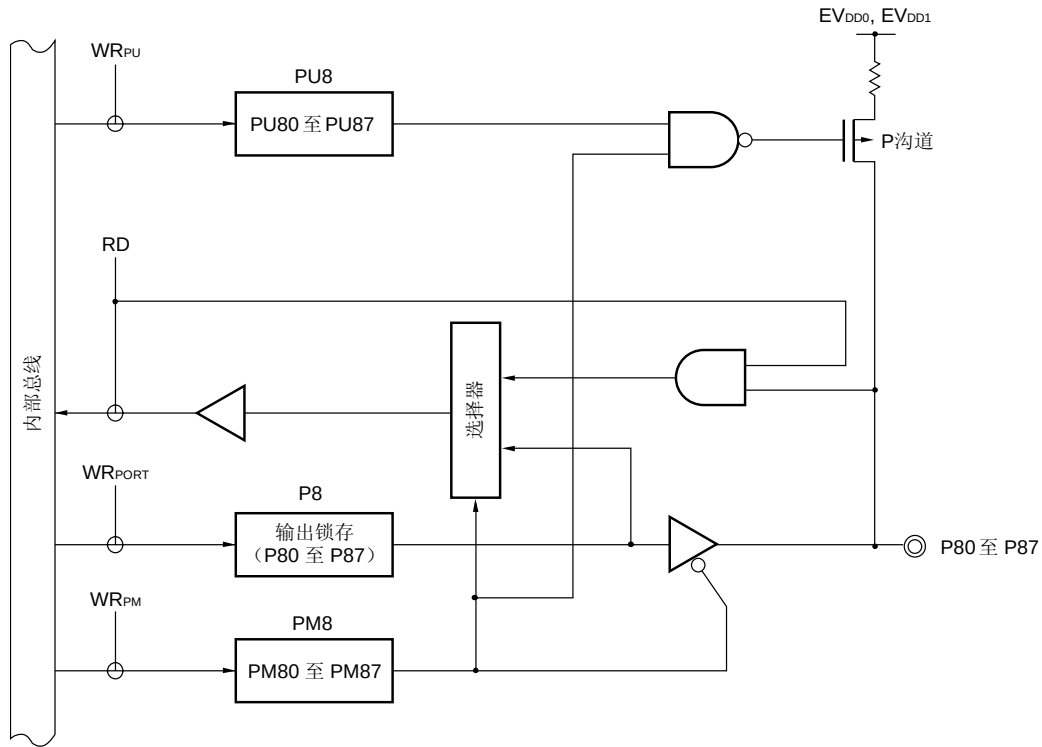
4.2.9 端口 8

端口 8 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 8(PM8)，可以以位单元设置端口 8 为输入或输出模式。当 P80 至 P87 引脚用作输入端口时，通过上拉电阻选项寄存器 8(PU8)，片上上拉电阻的使用可以以 1 位单元指定。

复位信号发生将端口 8 设置为输入模式。

图 4-28 所示为端口 8 的框图。

图 4-28. P80 至 P87 的框图



P8: 端口寄存器 8

PU8: 上拉电阻选项寄存器 8

PM8: 端口模式寄存器 8

RD: 读信号

WRxx: 写信号

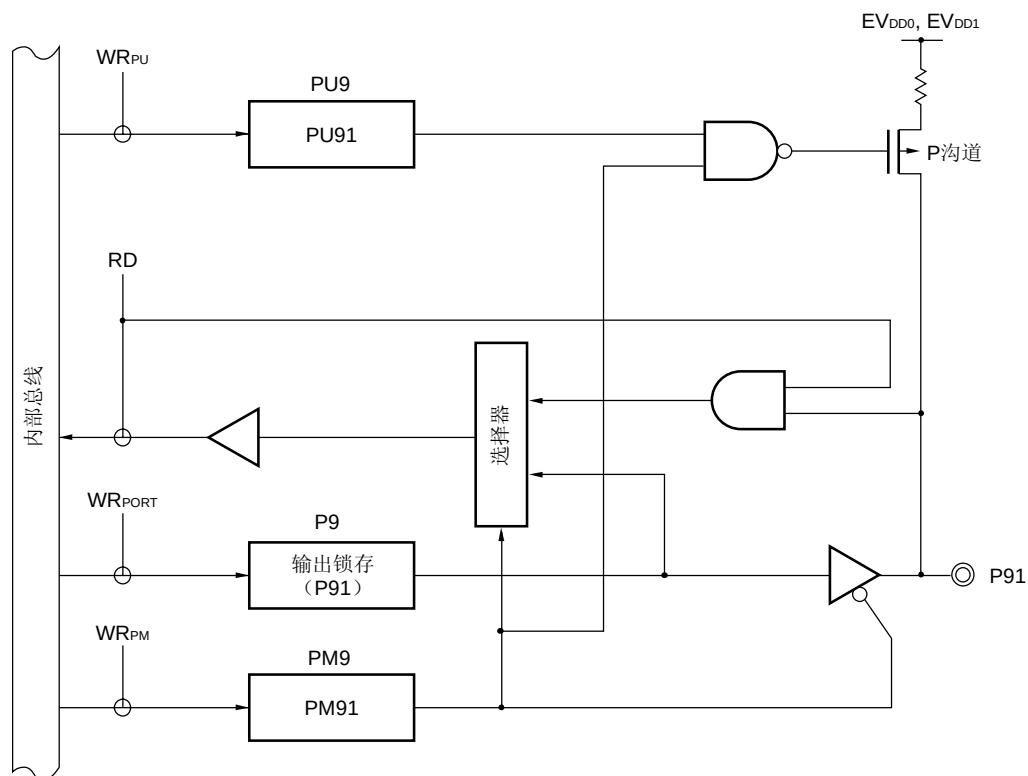
4.2.10 端口 9

P91 是一个具有输出锁存功能的 1 位 I/O 端口。通过使用端口模式寄存器 9(PM9)，可以以位单元设置端口 9 为输入或输出模式。当 P91 引脚用作输入端口时，通过上拉电阻选择寄存器 9(PU9)，片内上拉电阻的使用可以以位单元指定。

复位信号发生将 P91 设置为输入模式。

图 4-29 所示为端口 9 的框图。

图 4-29. P91 的框图



- P9: 端口寄存器 9
 PU9: 上拉电阻选项寄存器 9
 PM9: 端口模式寄存器 9
 RD: 读信号
 WR_{xx} : 写信号

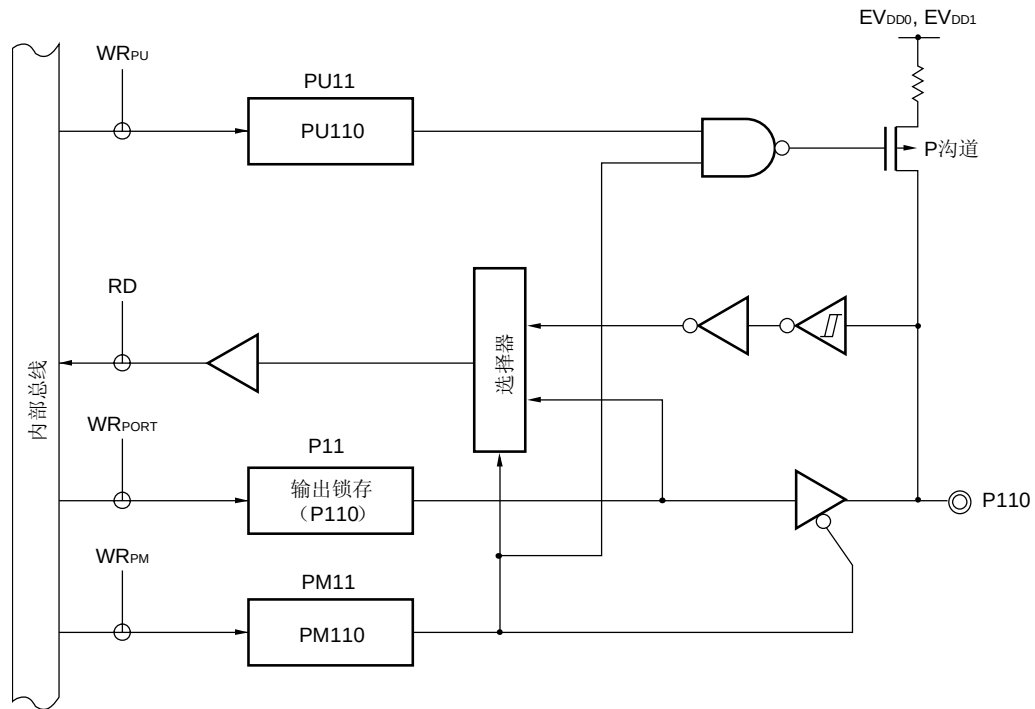
4.2.11 端口 11

端口 11 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 11(PM11)，可以以位单元设置端口 11 为输入或输出模式。当 P110 至 P111 用于输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 11(PU11)按位指定。

复位信号发生将端口 11 设置为输入模式。

图 4-30 和 4-31 所示为端口 11 的框图。

图 4-30. P110 的框图



P11: 端口寄存器 11

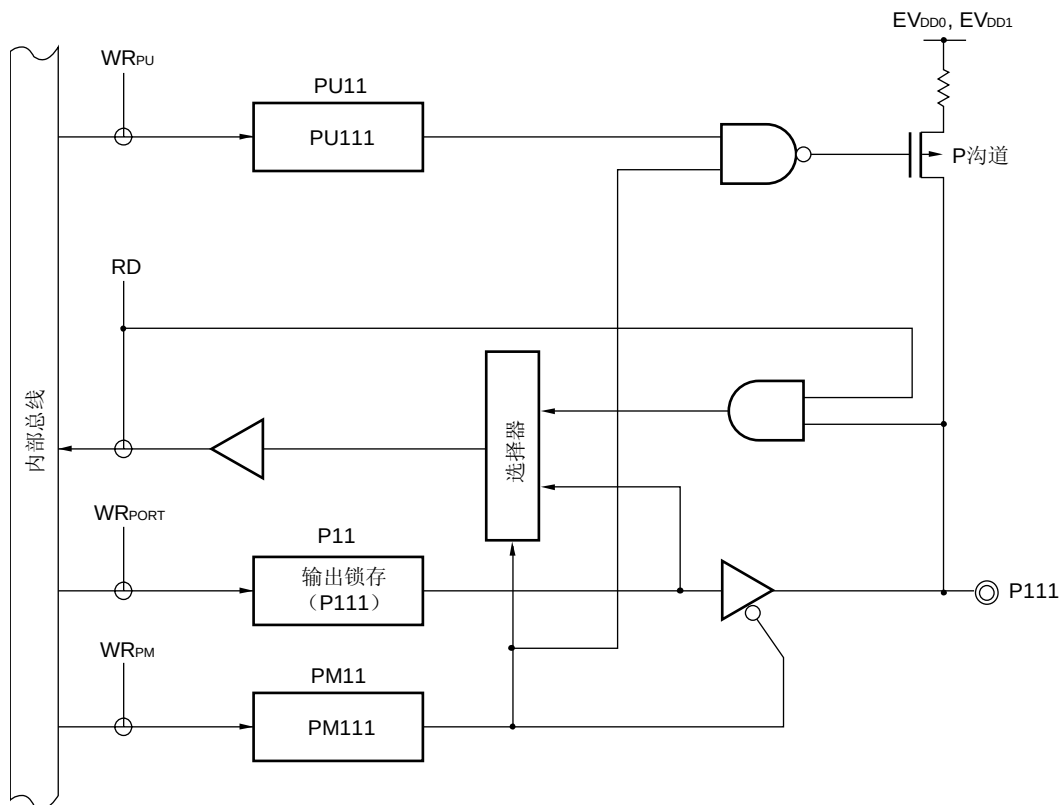
PU11: 上拉电阻选项寄存器 11

PM11: 端口模式寄存器 11

RD: 读信号

WR_{xx}: 写信号

图 4-31. P111 的框图



P11: 端口寄存器 11
 PU11: 上拉电阻选项寄存器 11
 PM11: 端口模式寄存器 11
 RD: 读信号
 WRxx: 写信号

4.2.12 端口 12

P120 是具有输出锁存功能的 1 位 I/O 端口。通过使用端口模式寄存器 12(PM12)，可以以 1 位单元设置端口 12 为输入或输出模式。当用作输入端口时，片内上拉电阻的使用可以通过上拉电阻选择寄存器 12(PU12)指定。

P121 至 P124 是 4 位输入端口。

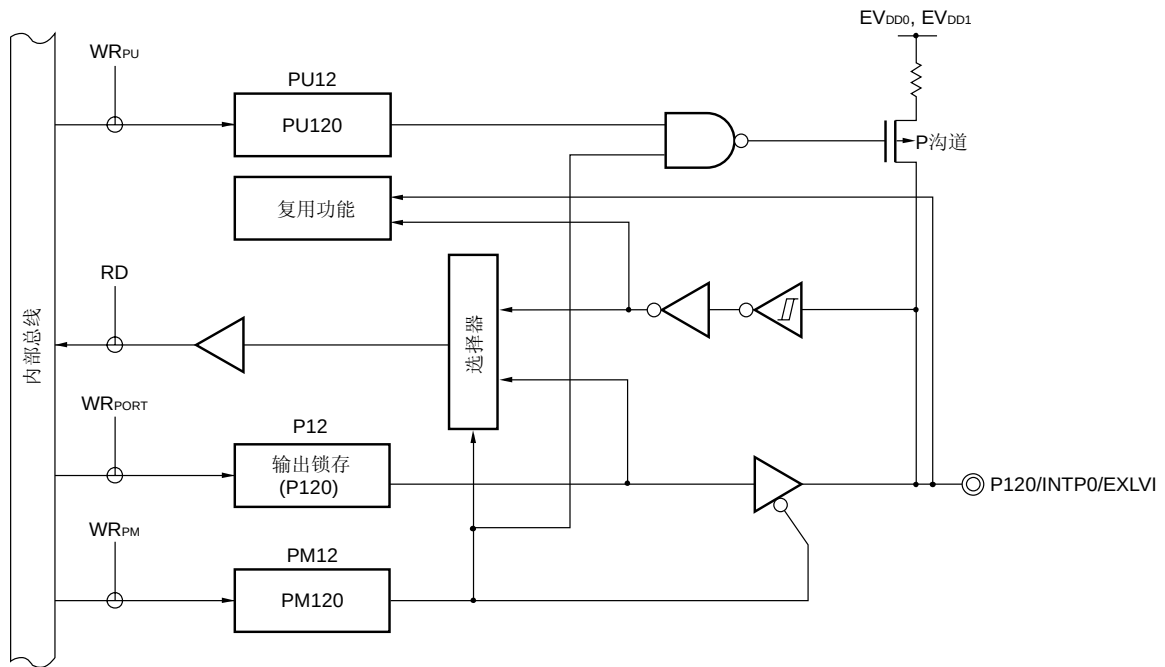
此端口也可用作外部中断请求输入、外部低电压检测的电压输入、主系统时钟振荡器连接、副系统时钟振荡器连接和主系统时钟的外部时钟输入。

复位信号发生将端口 12 设置为输入模式。

图 4-32 至 4-34 所示为端口 12 的框图。

注意事项 P121 至 P124 的功能设置只有在复位释放后可用。端口一旦设置为振荡器连接，除非执行复位，否则不能用作输入端口。

图 4-32 P120 的框图



P12: 端口寄存器 12

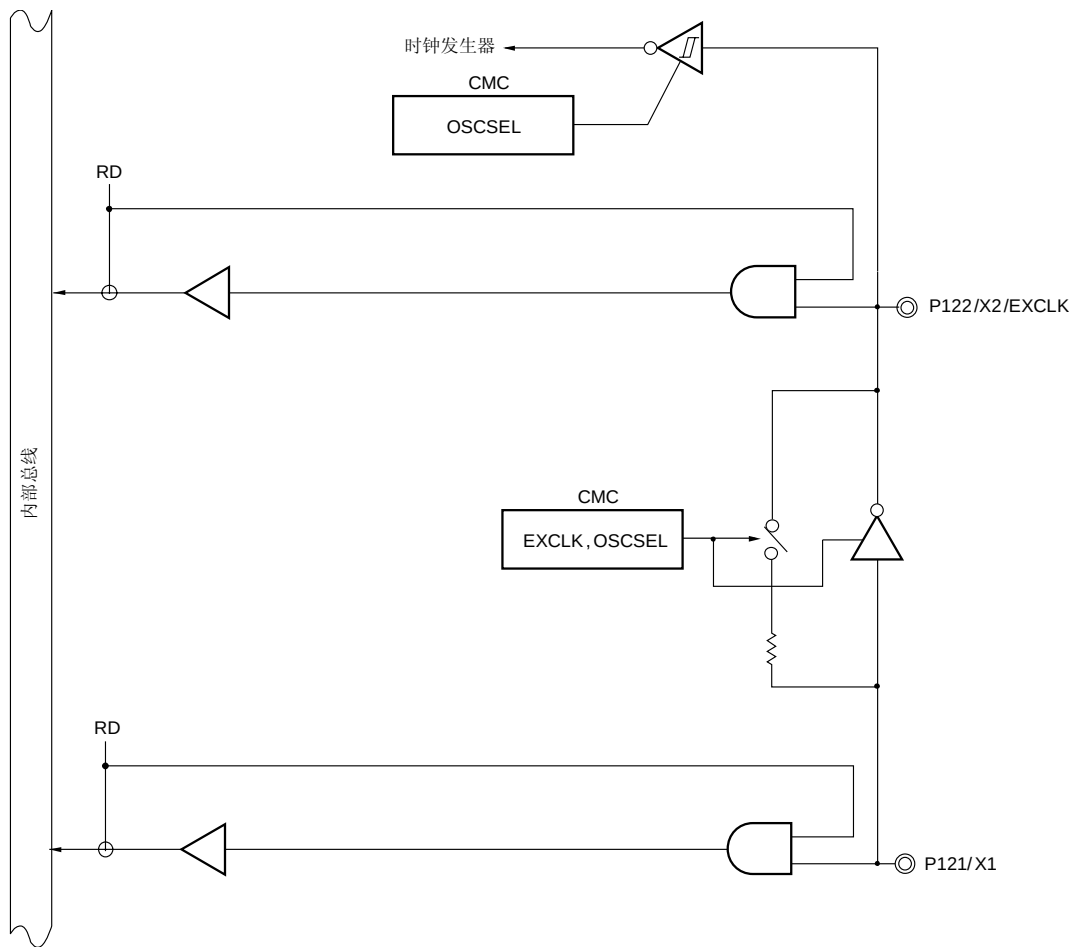
PU12: 上拉电阻选项寄存器 12

PM12: 端口模式寄存器 12

RD: 读信号

WRxx: 写信号

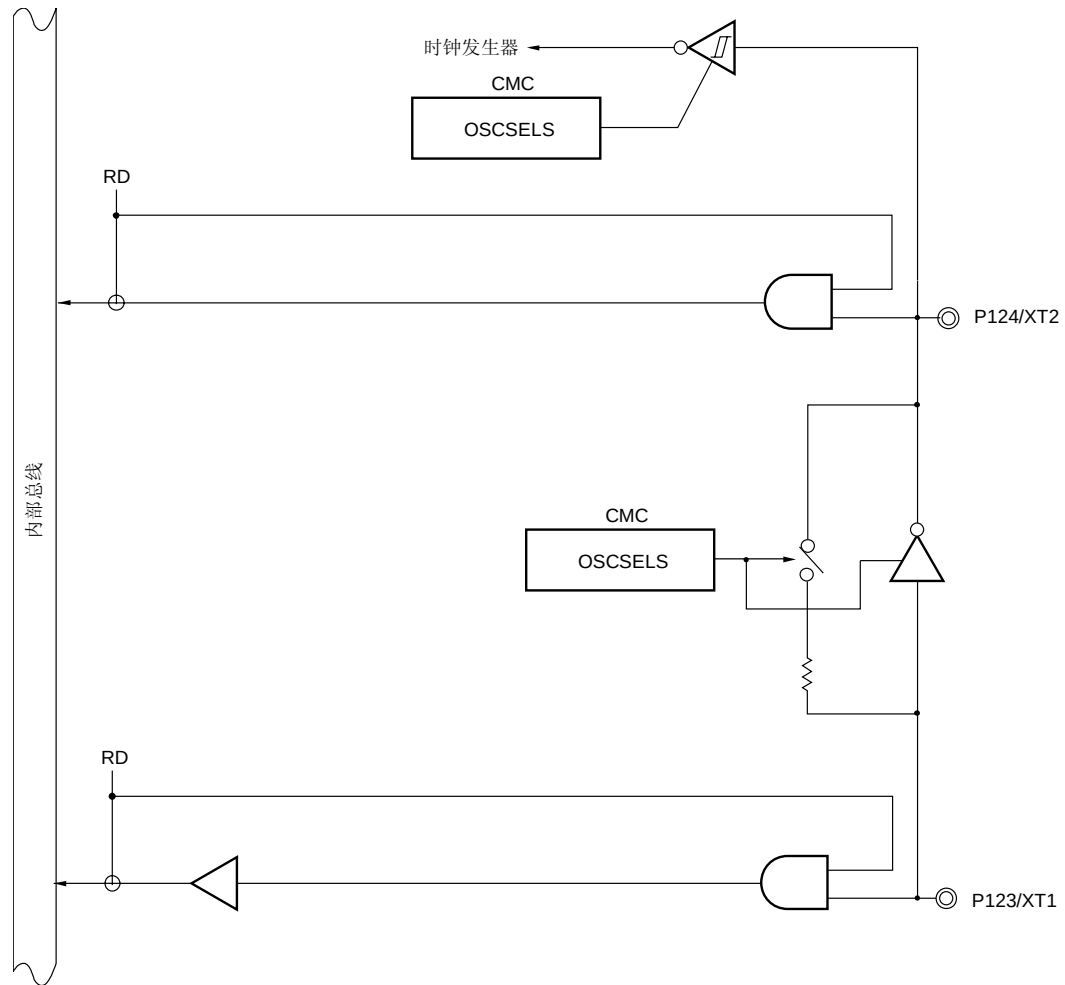
图 4-33. P121 和 P122 的框图



CMC: 时钟操作模式控制寄存器

RD: 读信号

图 4-34. P123 和 P124 的框图



CMC: 时钟操作模式控制寄存器

RD: 读信号

4.2.13 端口 13

P130 是具有输出锁存功能的 1 位单输出端口。

P131 是具有输出锁存功能的 1 位 I/O 端口。当用作输入端口时，片上上拉电阻的使用可以通过上拉电阻选择寄存器 13(PU13)指定。

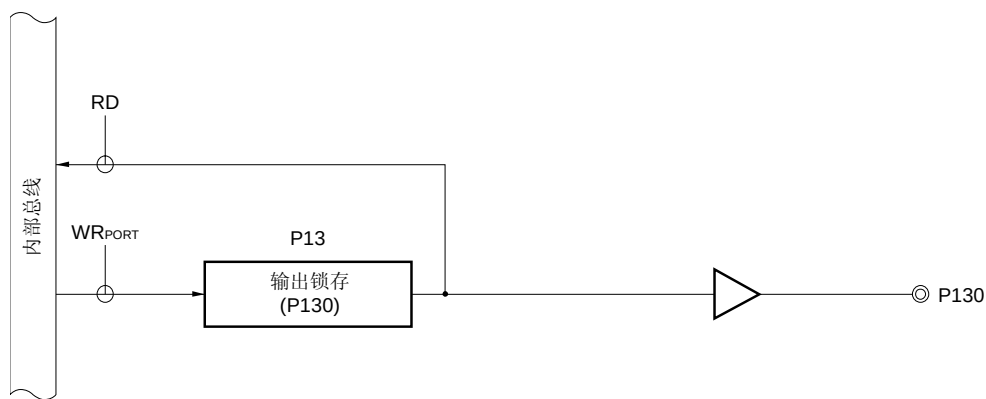
复位信号发生将端口 13 设置为输入模式。

此端口也可用于定时器 I/O。

图 4-35 和 4-36 所示为端口 13 的框图。

注意事项 P131/TI06/TO06 作为通用端口时，应设置定时器输出寄存器 0(TO0)的位 6(TO06)以及定时器输出使能寄存器 0(TE0)的位 6(TE06)为“0”，与他们默认状态设置相同。

图 4-35. P130 的框图



P13: 端口寄存器 13

RD: 读信号

WRxx: 写信号

备注 复位有效时，P130 输出低电平。如果 P130 在复位有效前设置为输出一个高电平，则 P130 的输出信号可作为 CPU 复位信号的虚拟输出。

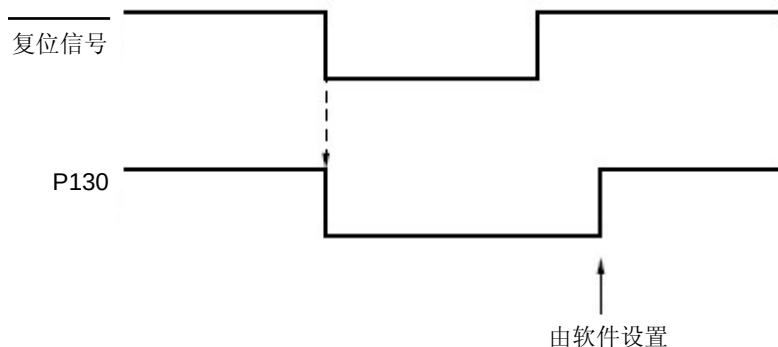
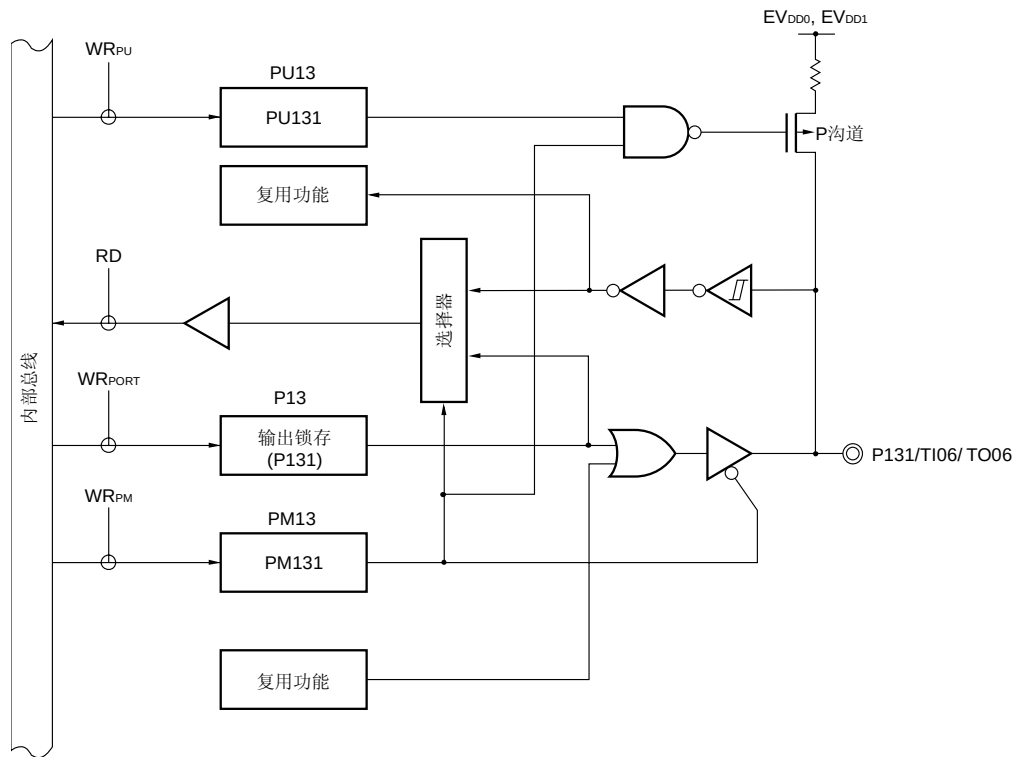


图 4-36. P131 的框图



P13: 端口寄存器 13

PU13: 上拉电阻选项寄存器 13

PM13: 端口模式寄存器 13

RD: 读信号

WRxx: 写信号

4.2.14 端口 14

端口 14 是具有输出锁存功能的 6 位 I/O 端口。通过使用端口模式寄存器 14(PM14)，可以以 1 位单元设置端口 14 为输入或输出模式。当 P140 和 P145 引脚用作输入端口时，通过上拉电阻选项寄存器 14(PU14)，可以以 14 位单元指定片内上拉电阻的使用。

通过端口输入模式寄存器 14(PIM14)，可以以 1 位单元指定 P142 和 P143 引脚通过普通缓冲器或 TTL 输入缓冲器的输入。

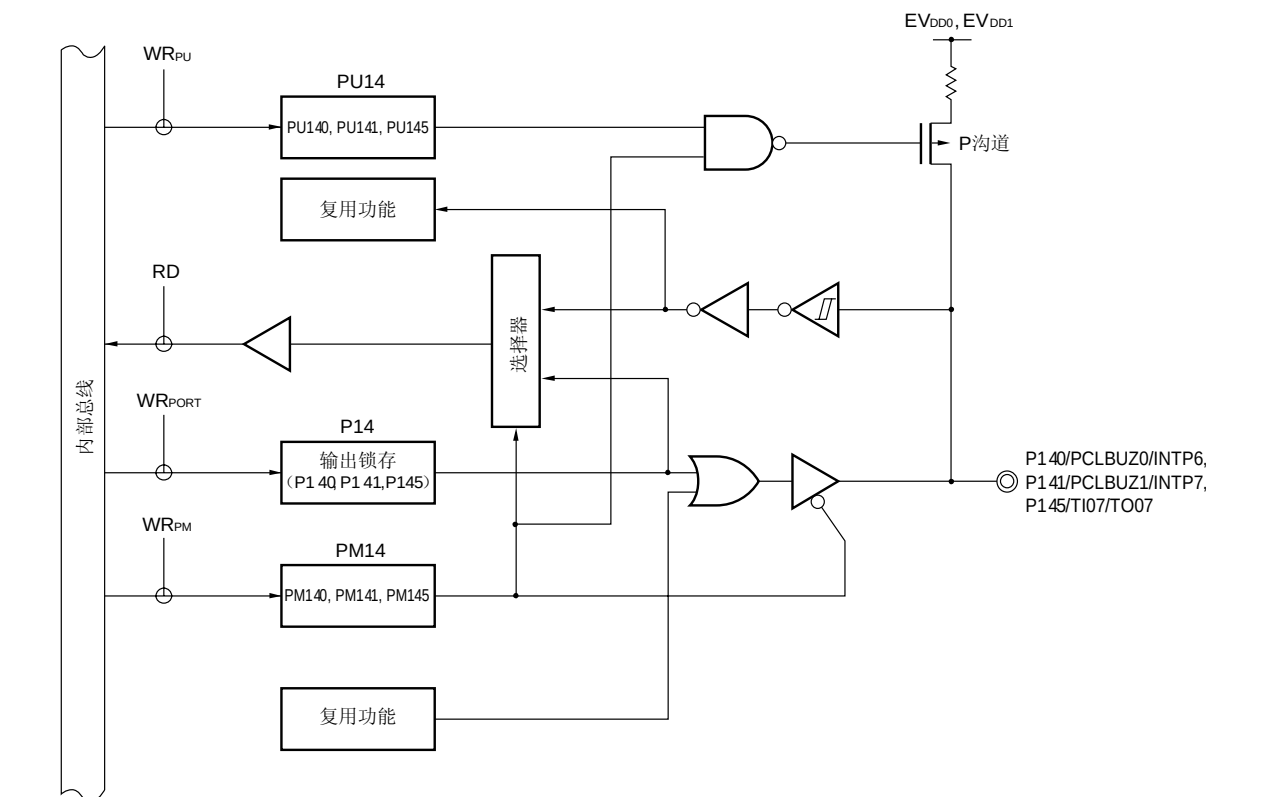
通过端口输出模式寄存器 14(POM14)，可以以 1 位单元指定 P142 至 P144 引脚的输出为 N 沟道漏极开路输出(V_{DD} 耐压)。

该端口也可作为定时器 I/O，外部中断请求输入，时钟/蜂鸣器输出，串行接口数据 I/O 和时钟 I/O 功能使用。

复位信号发生将端口 14 设置为输入模式。

图 4-37 至 4-39 所示为端口 14 的框图。

- 注意事项**
1. 要将 P142/ $\overline{SCK20}$ /SCL20、P143/SI20/RxD2/SDA20 或 P144/SO20/TxD2 作为通用端口使用，注意设置串行阵列单元 1。详情参见下列表格。
 - 表 11-9 寄存器设置和引脚之间的关系(单元 1 的通道 0: CSI20, UART2 发送, IIC20)
 - 表 11-10 寄存器设置和引脚之间的关系(单元 1 的通道 1: UART2 接收)
 2. P145/TI07/TO07 作为通用端口时，应设置定时器输出寄存器 0(TO0)的位 7(TO07) 和定时器输出使能寄存器 0(TOE0)的位 7(TOE07) 置“0”，和它们的初始状态设置相同。
 3. 要将 P140/PCLBUZ0/INTP6 或 P141/PCLBUZ1/INTP7 用作通用端口，设置时钟输出选择寄存器 0 和 1(CKS0, CKS1)的位 7 为“0”，同它们的默认状态设置相同。



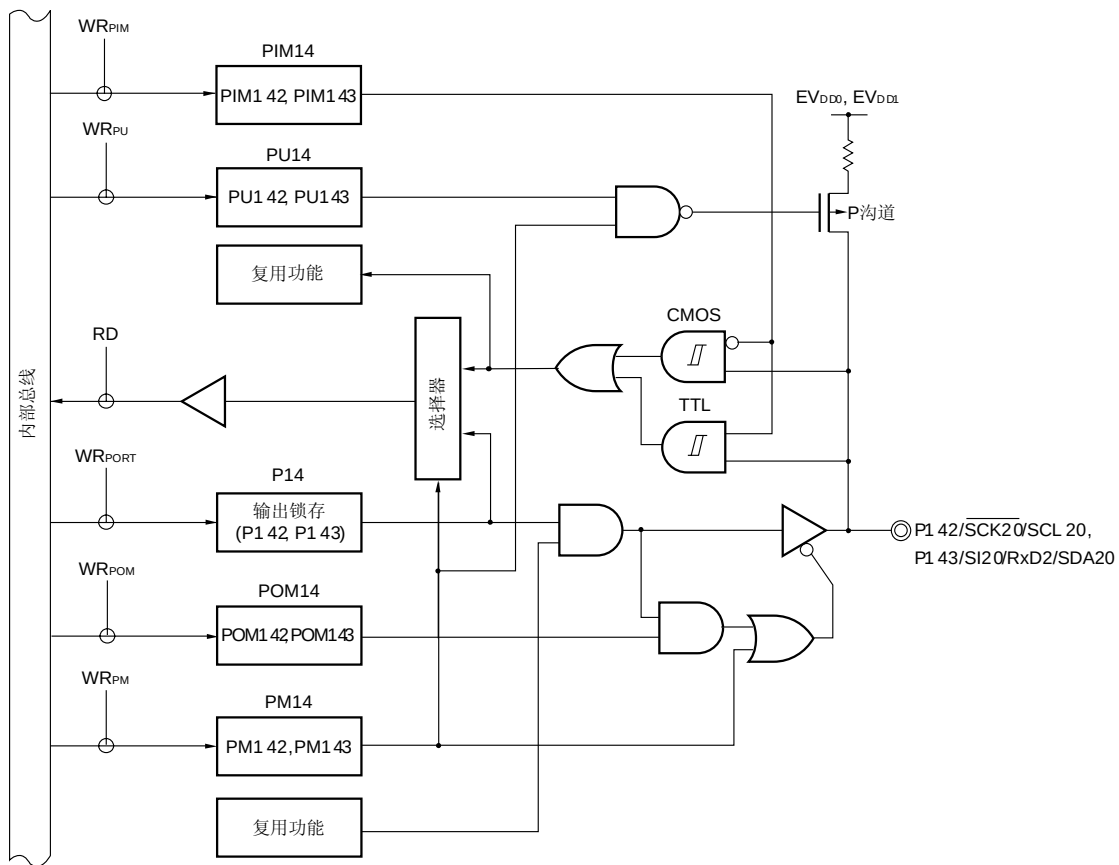
PU14: 上拉电阻选项寄存器 14

PM14: 端口模式寄存器 14

RD: 读信号

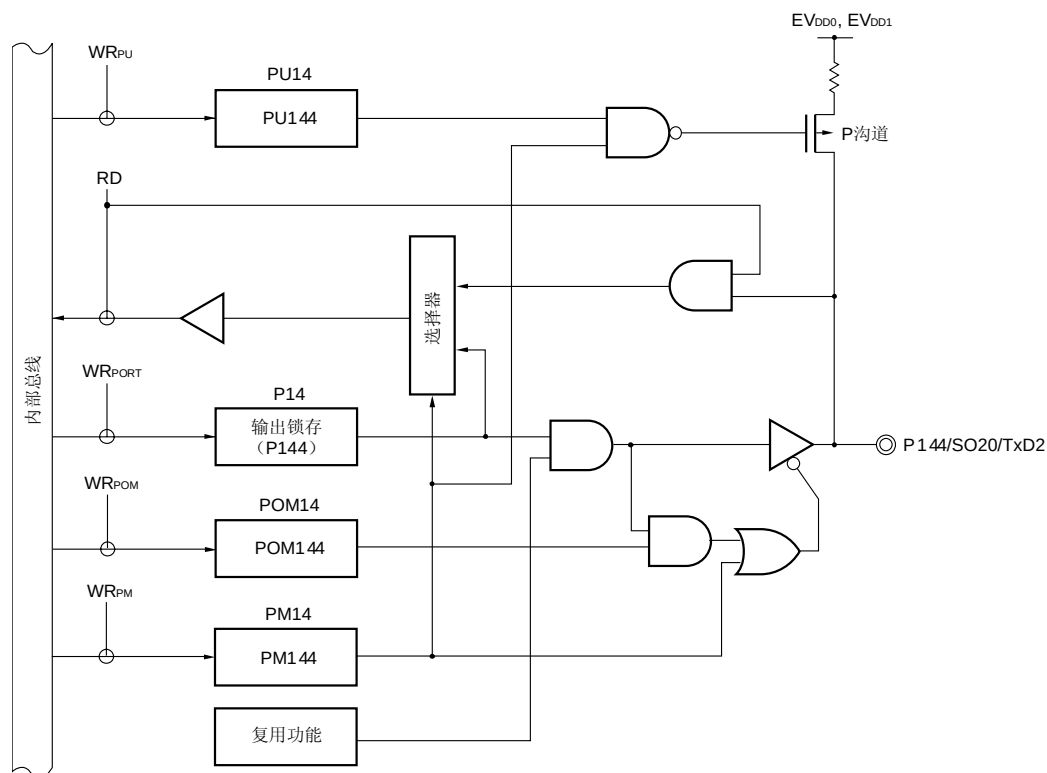
WR_{xx}: 写信号

图 4-38. P142 和 P143 的框图



- P14: 端口寄存器 14
 PU14: 上拉电阻选项寄存器 14
 PM14: 端口模式寄存器 14
 PIM14: 端口输入模式寄存器 14
 POM14: 端口输出模式寄存器 14
 RD: 读信号
 WR_{xx}: 写信号

图 4-39. P144 的框图



P14: 端口寄存器 14

PU14: 上拉电阻选项寄存器 14

PM14: 端口模式寄存器 14

POM14: 端口输出模式寄存器 14

RD: 读信号

WRxx: 写信号

4.2.15 端口 15

端口 15 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 15(PM15)，可以以 1 位单元设置端口 15 为输入或输出模式。

此端口也可用作 A/D 转换器模拟输入。

要将 P150/ANI8 至 P157/ANI15 用作数字输入引脚，可通过 A/D 端口配置寄存器(ADPC)设置其为数字 I/O 模式，通过 PM15 设置其为输入模式。从低位开始使用这些引脚。

要将 P150/ANI8 至 P157/ANI15 用作数字输出引脚，可通过 ADPC 设置其为数字 I/O 模式，通过 PM15 设置其为输出模式。

表 4-5. P150/ANI8 至 P157/ANI15 引脚的设置功能

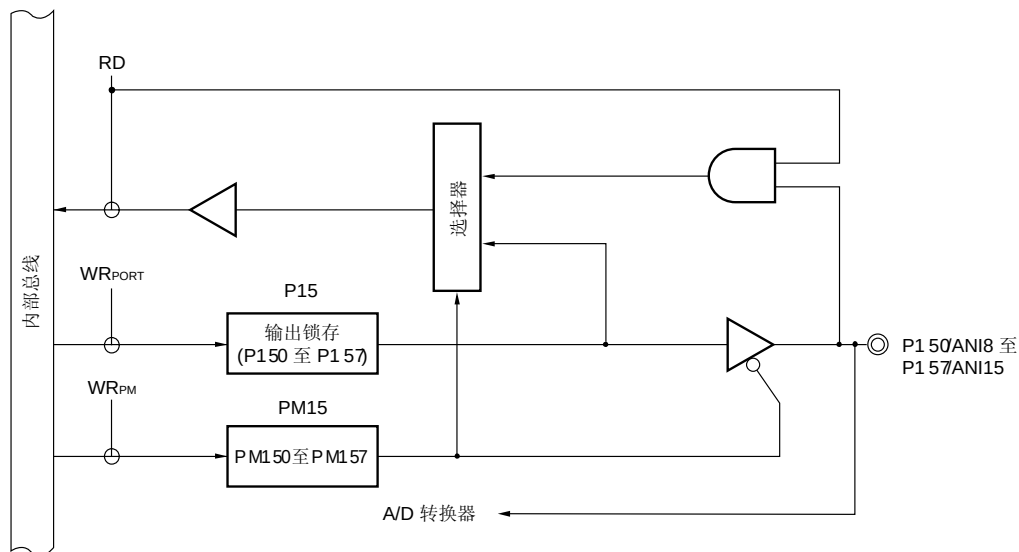
ADPC	PM15	ADS	P150/ANI8至 P157/ANI15 引脚
数字I/O 选择	输入模式	—	数字输入
	输出模式	—	数字输出
模拟输入选择	输入模式	选择ANI。	模拟输入(转换)
		不要选择ANI。	模拟输入(不转换)
	输出模式	选择ANI。	设置禁止
		不要选择ANI。	

复位信号发生时，P150/ANI8 至 P157/ANI15 所有引脚设置为数字输入模式。

图 4-40 所示为端口 15 的框图。

注意事项 当使用端口 15 作为数字 I/O 时，参见 2.2.16 应用的 AV_{REF} 引脚的电压 AV_{REF} 。

图 4-40. P150 至 P157 的框图



P15: 端口寄存器 15
 PM15: 端口模式寄存器 15
 RD: 读信号
 WRxx: 写信号

4.3 寄存器控制端口功能

端口功能由下列六种类型的寄存器控制。

- 端口模式寄存器(PM0 至 PM8、PM11 至 PM15)
- 端口寄存器 (P0 至 P9, P11 至 P15)
- 上拉电阻可选寄存器 (PU0, PU1, PU3 至 PU9, PU11 至 PU14)
- 端口输入模式寄存器 (PIM0, PIM1, PIM14)
- 端口输出模式寄存器 (POM0, POM1, POM14)
- A/D 端口配置 寄存器 (ADPC)

(1) 端口模式寄存器 (PM0 至 PM9, PM11 至 PM15)

这类寄存器以 1 位单元指定端口的输入或输出模式。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

产生复位信号设置这些寄存器为 FFH (PM13 的 FEH)。

当端口引脚用作复用功能引脚时，设置端口模式寄存器参见 4.5 使用复用功能时，设置端口模式寄存器和输出锁存器。

图 4-41. 端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FFF28H	FFH	R/W
PM9	1	1	1	1	1	1	PM91	1	FFF29H	FFH	R/W
PM11	1	1	1	1	1	1	PM111	PM110	FFF2BH	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM13	1	1	1	1	1	1	PM131	0	FFF2DH	FEH	R/W
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140	FFF2EH	FFH	R/W
PM15	PM157	PM156	PM155	PM154	PM153	PM152	PM151	PM150	FFF2FH	FFH	R/W
PMmn	Pmn 引脚 I/O 模式选择 (m = 0 至 9、11 至 15; n = 0 至 7)										
0	输出模式(输出缓冲器开)										
1	输入模式(输出缓冲器关)										

注意事项 确定设置 PM0 的位 7,PM3 的位 2 至 7,PM9 的位 0 和位 2 至 7,PM11 的位 2 至 7,PM12 的位 1 至 7,PM13 的位 2 至 7 和 PM14 的位 6 和 7 为“1”。确保将 PM13 的位 0 设置为“0”。

(2) 端口寄存器(P0 至 P9, P11 至 P15)

当数据从一个端口输出时，这类寄存器将芯片输出的数据写入。

如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值^注。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零(00H)。

注 当 P2 和 P15 设置为 A/D 转换器的模拟输入功能时，端口在输入模式期间读出的电平始终是 0，且引脚无电平输出。

图 4-42. 端口寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (输出锁存)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (输出锁存)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (输出锁存)	R/W
P3	0	0	0	0	0	0	P31	P30	FFF03H	00H (输出锁存)	R/W
P4	P47	P46	P45	P44	P43	P42	P41	P40	FFF04H	00H (输出锁存)	R/W
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFF05H	00H (输出锁存)	R/W
P6	P67	P66	P65	P64	P63	P62	P61	P60	FFF06H	00H (输出锁存)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFF07H	00H (输出锁存)	R/W
P8	P87	P86	P85	P84	P83	P82	P81	P80	FFF08H	00H (输出锁存)	R/W
P9	0	0	0	0	0	0	P91	0	FFF09H	00H (输出锁存)	R/W
P11	0	0	0	0	0	0	P111	P110	FFF0BH	00H (输出锁存)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	未定义	R/W [※]
P13	0	0	0	0	0	0	P131	P130	FFF0DH	00H (输出锁存)	R/W
P14	0	0	P145	P144	P143	P142	P141	P140	FFF0EH	00H (输出锁存)	R/W
P15	P157	P156	P155	P154	P153	P152	P151	P150	FFF0FH	00H (输出锁存)	R/W
Pmn		m = 0 至 9、11 至 15; n = 0 至 7									
		输出数据控制(输出模式)					输入数据读取(输入模式)				
0		输出 0					输入低电平				
1		输出 1					输入高电平				

注 P121 至 P124 只读。

(3) 上拉电阻选项寄存器(PU0, PU1, PU3 至 PU9, PU11 至 PU14)

这些寄存器指定是否使用 P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P64 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120, P131 或 P140 至 P145 引脚上的片内上拉电阻。可以以 1 位单元使用片内上拉电阻，仅用于那些引脚已设置为输入模式的位，并且这些引脚已通过 PU0,PU1,PU3 至 PU9 和 PU11 至 PU14 指定了片上上拉电阻的使用。不论是否设置了 PU0,PU1,PU3 至 PU9 和 PU11 至 PU14，那些设置为输出模式的位和用作复用功能的输出引脚的位都不能连接片内上拉电阻。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零 00H。

图 4-43. 上拉电阻选项寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	F0034H	00H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU6	PU67	PU66	PU65	PU64	0	0	0	0	F0036H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU8	PU87	PU86	PU85	PU84	PU83	PU82	PU81	PU80	F0038H	00H	R/W
PU9	0	0	0	0	0	0	PU91	0	F0039H	00H	R/W
PU11	0	0	0	0	0	0	PU111	PU110	F003BH	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU13	0	0	0	0	0	0	PU131	0	F003DH	00H	R/W
PU14	0	0	PU145	PU144	PU143	PU142	PU141	PU140	F003EH	00H	R/W
PUmn	Pmn 引脚片上上拉电阻选择 (m = 0,1,3 至 9,11 至 14; n = 0 至 7)										
0	片上上拉电阻不连接										
1	片上上拉电阻连接										

(4) 端口输入模式寄存器(PIM0, PIM1, PIM14)

这些寄存器以 1 位单元的方式设置 P03, P04, P10, P11, P142 或 P143 的输入缓冲。

在与不同电位的外部器件串行通讯期间，可以选择 TTL 输入缓冲器。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零(00H)。

图 4-44. 端口输入模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PIM0	0	0	0	PIM04	PIM03	0	0	0	F0040H	00H	R/W
PIM1	0	0	0	0	0	0	PIM11	PIM10	F0044H	00H	R/W
PIM14	0	0	0	0	PIM143	PIM142	0	0	F004EH	00H	R/W
PIMmn	Pmn 引脚输入缓冲选择 (m = 0,1,14; n=0 至 4)										
0	普通输入缓冲器										
1	TTL 输入缓冲器										

(5) 端口输出模式寄存器(POM0, POM1, POM14)

这些寄存器以 1 位单元的方式设置 P02 至 P04, P10, P12 或 P142 的输出模式。

在与不同电势的外部设备串行通信期间，可以选择 N 沟道漏极开路输出(V_{DD} 耐压)。与相同电势的外部设备简易 I²C 通信期间，可选择 SDA10 和 SDA20 引脚。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零 00H。

图 4-45. 端口输出模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM0	0	0	0	POM04	POM03	POM02	0	0	F0050H	00H	R/W
POM1	0	0	0	0	0	POM12	0	POM10	F0051H	00H	R/W
POM14	0	0	0	POM144	POM143	POM142	0	0	F005EH	00H	R/W
POMmn	Pmn 引脚输出模式选择 (m = 0,1,14; n = 0,2 至 4)										
0	正常输出模式.										
1	N-ch 漏极开路(V _{DD} 耐压)模式										

(6) A/D 端口配置寄存器 (ADPC)

此寄存器切换 P20/ANI0 至 P27/ANI7 和 P150/ANI8 至 P157/ANI15 引脚到端口的数字 I/O 或 A/D 转换器的模拟输入。

ADPC 可以由 1 位或 8 位存储器操作指令设置。

复位信号的发生设置此寄存器为 10H。

图 4-46. A/D 端口配置寄存器 (ADPC) 的格式

地址: F0017H 复位后: 10H R/W

符号	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	模拟输入 (A) / 数字 I/O (D) 切换															
					端口 15								端口 2							
					ANI15/ P157	ANI14/ P156	ANI13/ P155	ANI12/ P154	ANI11/ P153	ANI10/ P152	ANI9/ P151	ANI8/ P150	ANI7/ P27	ANI6/ P26	ANI5/ P25	ANI4/ P24	ANI3/ P23	ANI2/ P22	ANI1/ P21	ANI0/ P20
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	A	A	A	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	A	A	A	A	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	A	A	A	A	A	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	A	A	A	A	A	A	A	A	D	D	D	D	D	D
0	0	1	1	1	A	A	A	A	A	A	A	A	A	D	D	D	D	D	D	D
0	1	0	0	0	A	A	A	A	A	A	A	A	D	D	D	D	D	D	D	D
0	1	0	0	1	A	A	A	A	A	A	A	D	D	D	D	D	D	D	D	D
0	1	0	1	0	A	A	A	A	A	A	D	D	D	D	D	D	D	D	D	D
0	1	0	1	1	A	A	A	A	A	D	D	D	D	D	D	D	D	D	D	D
0	1	1	0	0	A	A	A	A	D	D	D	D	D	D	D	D	D	D	D	D
0	1	1	0	1	A	A	A	D	D	D	D	D	D	D	D	D	D	D	D	D
0	1	1	1	0	A	A	D	D	D	D	D	D	D	D	D	D	D	D	D	D
0	1	1	1	1	A	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
其它值					设置禁止															

- 注意事项**
1. 利用端口模式寄存器 2 和 15 (PM2, PM15)，设置输入模式下的用于 A/D 转换的通道。
 2. 不去设置那些通过模拟输入通道指定寄存器 (ADS)、由 ADPC 设置为数字 I/O 的引脚。
 3. 由 A/D 端口配置寄存器(ADPC)以 P157/ANI15, ..., P150/ANI8, P27/ANI7, ..., P20/ANI0 的顺序设置 P20/ANI0 至 P27/ANI7 和 P150/ANI8 至 P157/ANI15 为模拟输入。当使用 P20/ANI0 至 P27/ANI7 和 P150/ANI8 至 P157/ANI15 作为模拟输入时，从 P157/ANI15 开始设计。

4.4 端口功能操作

对输入模式和输出模式的端口操作是不同的，具体如下所示。

4.4.1 写入 I/O 端口

(1) 输出模式

使用传输指令对输出锁存器进行写操作，输出锁存器的内容从引脚输出。

一旦数据写入输出锁存器，它将一直保存到新数据写入。

当复位信号产生时，输出锁存器的内容清零。

(2) 输入模式

使用传输指令对输出锁存器进行写操作，但因为输出缓冲器处于关闭状态，所以引脚状态不会改变。

一旦数据写入输出锁存器，它将一直保存到新数据写入。

当复位信号产生时，输出锁存器的内容清零。

4.4.2 从 I/O 端口读取

(1) 输出模式

使用传输指令读取输出锁存器的内容。输出锁存器的内容不会改变。

(2) 输入模式

使用传输指令读取引脚状态。输出锁存器的内容不会改变。

4.4.3 I/O 端口的操作

(1) 输出模式

对输出锁存器执行一个操作时，操作结果写入输出锁存器。而输出锁存器的内容则从引脚输出。

一旦数据写入输出锁存器，它将一直保存到新数据写入。

当复位信号产生时，输出锁存器的内容清零。

(2) 输入模式

读取引脚上的电平，对其内容执行操作。虽然操作的结果写到输出锁存器上，但是由于输出缓冲器是关闭的，所以引脚状态并没有改变。

当复位信号产生时，清零输出锁存器的内容。

4.4.4 以不同电势(2.5 V、3 V)连接到外部设备

当端口 0,1 和 14 的部分工作在 $V_{DD} = 4.0V$ 至 $5.5V$ 时，与工作在 2.5V 或 3V 的供电电压的外部设备相连接的 I/O 可以实现。

关于输入，由端口输入模式寄存器(PIM0, PIM1, PIM14)可以实现 CMOS/TTL 按位切换。

另外，关于输出，利用端口输出模式寄存器(POM0, POM1, POM14)，通过将输出缓冲器切换为 N 沟道漏极开路 (V_{DD} 耐压)，就可支持不同电势。

(1) 使用 UART0 至 UART2、CSI00、CSI10 和 CSI20 功能的 I/O 引脚时的设置过程

(a) 用于 2.5V、3V 输入端口

- <1> 复位释放后，端口模式为输入模式(Hi-Z)。
- <2> 如果需要上拉，使用外部上拉引脚 (片上上拉电阻不能使用)。

UART0 的情况下:	P11
UART1 的情况下:	P03
UART2 的情况下:	P143
CSI00 的情况下:	P10, P11
CSI10 的情况下:	P03, P04
CSI20 的情况下:	P142, P143

- <3> 设置 PIMn 寄存器的相关位为 1，以切换到 TTL 输入缓冲器。
- <4> V_{IH}/V_{IL} 工作在 2.5V、3V 工作电压。

(b) 用于 2.5V,3V 输出端口

- <1> 复位释放后，端口模式改变为输入模式 (Hi-Z)。
- <2> 需要使用外部上拉引脚 (片上上拉电阻不能使用)。

UART1 的情况下:	P12
UART1 的情况下:	P02
UART2 的情况下:	P144
CSI00 的情况下:	P10, P12
CSI10 的情况下:	P02, P04
CSI20 的情况下:	P142, P144

- <3> 设置相关端口的输出锁存为 1。
- <4> 设置 POMn 寄存器的相关位为 1 以设置为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。
- <5> 通过操作 PMn 寄存器设置输出模式。
同时，输出数据为高电平，因此引脚为高阻状态。
- <6> 通过设置串行阵列单元开始通信。

注 n = 0, 1, 14

(2) 使用 IIC10 和 IIC20 功能的 I/O 引脚时的设置过程

- <1> 复位释放后，端口模式为输入模式(Hi-Z)。
- <2> 需要使用外部上拉引脚 (片上上拉电阻不能使用)。

在简易 IIC10 的情况中：P03, P04

在简易 IIC20 的情况中：P142, P143

- <3> 设置相关端口的输出锁存为 1。
- <4> 设置 POMn 寄存器的相关位为 1 以设置为 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。
- <5> 设置 PMn 寄存器的相应位为输出模式(数据 I/O 可以是输出模式)。
同时，输出数据为高电平，因此引脚为高阻状态。
- <6> 实现串行阵列单元操作，并且将此模式设置成简易 IIC 模式。

注 n = 0, 14

4.5 使用复用功能时端口模式寄存器和输出锁存的设置。

要使用端口的复用功能，按下表 4-6 设置端口模式寄存器和输出锁存器。

表 4-6. 使用复用功能时端口模式寄存器和输出锁存的设置(1/2)

引脚名称	复用功能		PMxx	Pxx	引脚名称	复用功能		PMxx	Pxx
	功能名称	I/O				功能名称	I/O		
P00	TI00	输入	1	×	P31	TI03	输入	1	×
P01	TO00	输出	0	0		TO03	输出	0	0
P02	SO10	输出	0	1		INTP4	输入	1	×
	TxD1	输出	0	1	P40	TOOL0	I/O	×	×
P03	SI10	输入	1	×	P41	TOOL1	输出	×	×
	RxD1	输入	1	×	P42	TI04	输入	1	×
	SDA10	I/O	0	1		TO04	输出	0	0
P04	SCK10	输入	1	×	P43	SCK01	输入	1	×
		输出	0	1			输出	0	1
	SCL10	I/O	0	1	P44	SI01	输入	1	×
P10	SCK00	输入	1	×	P45	SO01	输出	0	1
		输出	0	1	P46	TI05	输入	1	×
P11	SI00	输入	1	×		TO05	输出	0	0
	RxD0	输入	1	×		INTP1	输入	1	×
P12	SO00	输出	0	1	P47	INTP2	输入	1	×
	TxD0	输出	0	1	P60	SCL0	I/O	0	0
P13	TxD3	输出	0	1	P61	SDA0	I/O	0	0
P14	RxD3	输入	1	×	P64 至 P67	TI10 至 TI13	输入	1	×
P15	RTCDIV	输出	0	0		TO10 至 TO13	输出	0	0
	RTCCL	输出	0	0	P70 至 P73	KR0 至 KR3	输入	1	×
P16	TI01	输入	1	×	P74 至 P77	INTP8 至 INTP11	输入	1	×
	TO01	输出	0	0		KR4 至 KR7	输入	1	×
	INTP5	输入	1	×	P120	INTP0	输入	1	×
P17	TI02	输入	1	×		EXLVI	输入	1	×
	TO02	输出	0	0	P131	TI06	输入	1	×
P20至P27 ^注	ANIO至ANI7 ^注	输入	1	×		TO06	输出	0	0
P30	RTC1HZ	输出	0	0					
	INTP3	输入	1	×					

备注 ×: 不必考虑
 PMxx: 端口模式寄存器
 Pxx: 端口输出锁存

(注列在下页中)

表 4-6. 使用复用功能时,PF11,PF6, 端口模式寄存器和输出锁存的设置(2/2)

引脚名称	复用功能		PMxx	Pxx
	功能名称	I/O		
P140	PCLBUZ0	输出	0	0
	INTP6	输入	1	×
P141	PCLBUZ1	输出	0	0
	INTP7	输入	1	×
P142	SCK20	输入	1	×
		输出	0	1
	SCL20	I/O	0	1
P143	SI20	输入	1	×
	RxD2	输入	1	×
	SDA20	I/O	0	1
P144	SO20	输出	0	1
	TxD2	输出	0	1
P145	TI07	输入	1	×
	TO07	输出	0	0
P150 至 P157 ^注	ANI8 至 ANI15 ^注	输入	1	×

注 ANI0/P20 至 ANI7/P27 和 ANI8/P150 至 ANI15/P157 引脚的功能可通过 A/D 端口配置寄存器(ADPC),模拟输入通道选择寄存器(ADS),PM2,和 PM15 选择。

表 4-7 ANI0/P20 至 ANI7/P27 和 ANI8/P150 至 ANI15/P157 引脚的设置功能

ADPC	PM2, PM15	ADS	ANI0/P20至ANI7/P27、 ANI8/P150至ANI15/P157引脚
数字 I/O 选择	输入模式	—	数字输入
	输出模式	—	数字输出
模拟输入选择	输入模式	选择ANI。	模拟输入(转换)
		不要选择ANI。	模拟输入(不转换)
	输出模式	选择ANI。	设置禁止
		不要选择ANI。	

备注 ×: 不必考虑
PMxx: 端口模式寄存器
Pxx: 端口输出锁存

4.6 端口寄存器 n (Pn)的 1 位操作指令的注意事项

当对一个端口执行 1 位操作指令，该端口既有输入功能又有输出功能，没操作的输入端口的输出锁存值可能会写入除目标位以外的位。

所以，建议在切换一个端口从输入模式到输出模式时，要重写输出锁存。

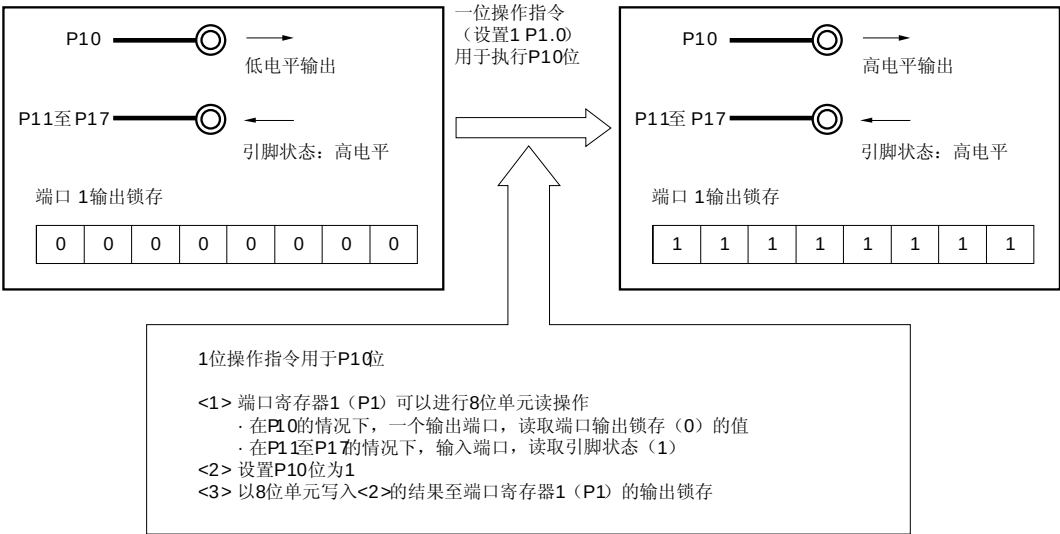
<示例> 当 P10 是输出端口时，P11 至 P17 是输入端口(所有引脚状态是高电平)，端口 1 的端口锁存值是 00H，如果通过 1 位操作指令把输出端口 P10 的输出从低电平变化到高电平，输出端口 1 的输出锁存值变为 FFH。

解释: 从 PMnm 位是 1 的端口 Pn 寄存器写入和读出的目标值分别是输出锁存和引脚状态。
在 78K0R/KG3-L 中，1 位操作指令是按照以下次序执行的。

- <1> 以 8 位单元读出 Pn 寄存器。
- <2> 一个目标位操作。
- <3> 以 8 位单元写入 Pn 寄存器。

在步骤<1>中，输出端口 P10 的输出缓存值(0)读出，同时输入端口 P11 至 P17 的引脚状态也读出。
如果 P11 至 P17 的引脚状态在此时是高电平，读出的值则为 FEH。
通过<2>中的操作把该值改变为 FFH。
FFH 通过<3>中的操作写入输出锁存。

图 4-47. 位操作指令(P10)



5.1 时钟发生器的功能

时钟发生器用于产生时钟，并提供给 CPU 和外部硬件设备。

可以使用以下三种系统时钟和时钟振荡器。

(1) 主系统时钟

<1> X1 振荡器

通过连接一个振荡器到 X1 和 X2，该振荡电路产生 $f_x = 2$ 至 20MHz 的时钟。

通过执行 STOP 指令或设置 MSTOP(时钟操作状态控制寄存器(CSC)的第 7 位)，可以停止振荡。

<2> 内部高速振荡器[※]

该振荡电路产生 $f_{IH} = 1, 8 \text{ MHz}$ (典型值)的时钟。复位释放后，CPU 总是使用这个内部高速振荡时钟进行操作。可以通过执行 STOP 指令或者设置 HIOSTOP(CSC 的第 0 位)停止其振荡。

<3> 20MHz 内部高速振荡时钟振荡器[※]

该振荡电路产生 $f_{IH20} = 20\text{MHz}$ (典型值)的时钟。当 $V_{DD} \geq 2.7 \text{ V}$ 时，可以通过设置 20MHz 内部高速振荡器控制寄存器(DSCCTL)的位 0(DSCON)为 1 开始振荡，设置 DSCON 为 0 停止振荡。

注 使用 1、8 或 20MHz 的内部高速振荡时钟，要使用选项字节预先设置频率(详细内容，参考第 22 章 选项字节)此外，复位释放后，内部高速振荡器自动开始振荡。要使用 20 MHz 内部高速振荡器来操作微控制器，通过设置 20 MHz 内部高速振荡控制寄存器(DSCCTL)的位 0(DSCON)为 1 使振荡器开始振荡。

外部主系统时钟($f_{EX} = 2$ 至 20 MHz)也可以通过 EXCLK/X2/P122 引脚提供。可以通过执行 STOP 指令或者设置 MSTOP，禁止外部主系统时钟输入。

通过设置 MCM0(系统时钟控制寄存器(CKC)的第 4 位)可以选择高速系统时钟(X1 时钟或者外部主系统时钟)或内部高速振荡时钟作为主系统时钟。

(2) 副系统时钟

• XT1 时钟振荡器

通过在 XT1 和 XT2 之间连接一个 32.768kHz 的振荡器，该电路以 $f_{SUB} = 32.768\text{kHz}$ 的频率进行振荡。通过设置 XTSTOP(CSC 的第 6 位)，可以停止振荡。

备注

- f_x : X1 时钟振荡频率
- f_{IH} : 内部高速振荡时钟频率
- f_{IH20} : 20 MHz 内部高速振荡时钟频率
- f_{EX} : 外部主系统时钟频率
- f_{SUB} : 副系统时钟频率

(3) 内部低速振荡时钟(专用于看门狗定时器的时钟)

• 内部低速振荡器

此电路振荡时钟 $f_{IL} = 30\text{ kHz}$ (典型值)。

内部低速振荡时钟不能用作 CPU 时钟。工作在内部低速振荡时钟的硬件只有看门狗定时器。

当看门狗定时器停止时，振荡停止。

- 备注
- 1. f_{IL} : 内部低速振荡时钟频率
 - 2. 以下情况看门狗定时器停止。
 - 当选项字节(000C0H)的第 4 位(WDTON) = 0
 - 当选项字节(000C0H)的第 4 位(WDTON) = 1 和第 0 位(WDSTBYON) = 0 时，如果执行 HALT 或 STOP 指令

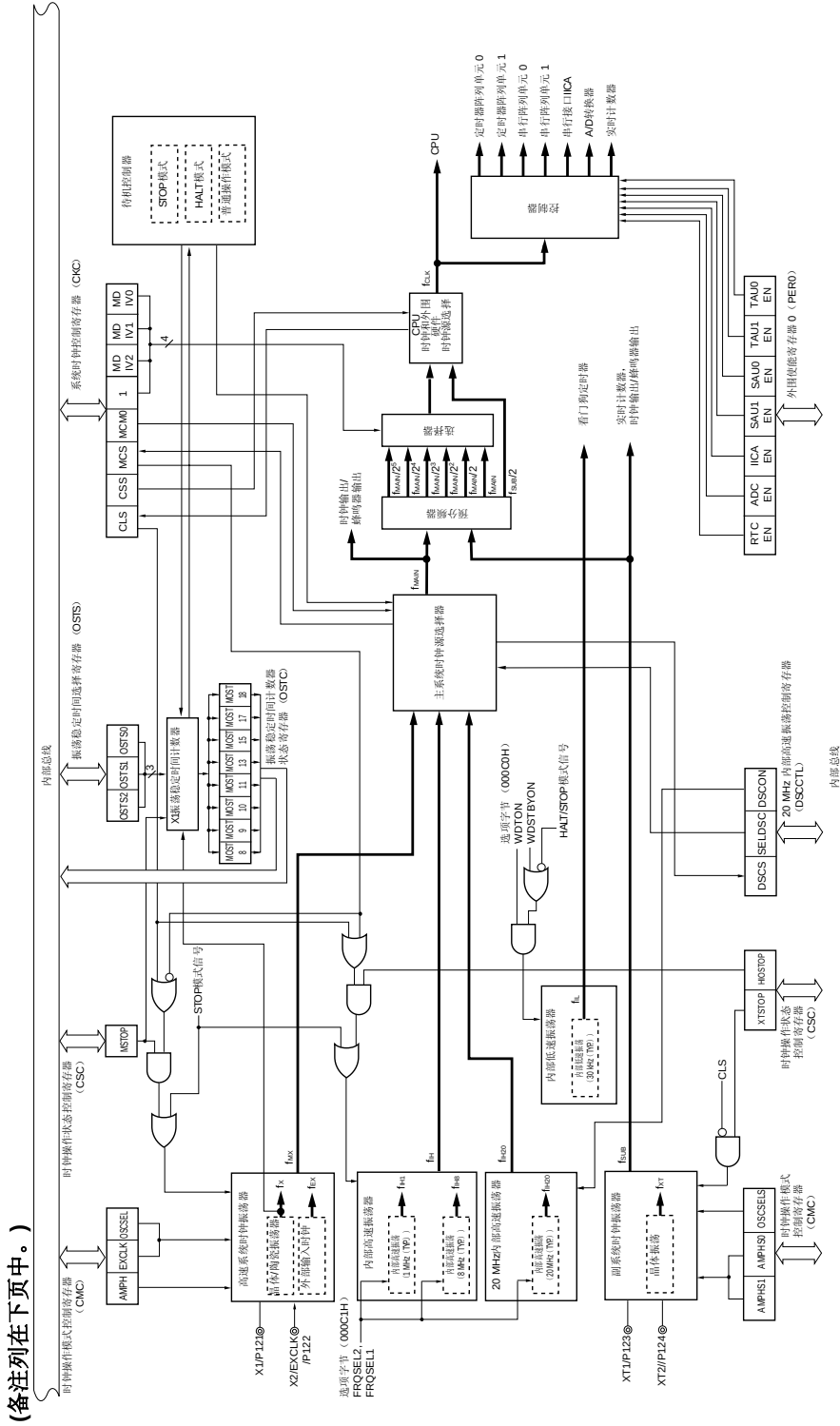
5.2 时钟发生器的配置

时钟发生器包括以下硬件。

表 5-1. 时钟发生器的配置

项目	配置
控制寄存器	时钟操作模式控制寄存器(CMC) 时钟操作状态控制寄存器(CSC) 振荡稳定时间计数器状态寄存器(OSTC) 振荡稳定时间选择寄存器(OSTS) 系统时钟控制寄存器(CKC) 20 MHz 内部高速振荡控制寄存器(DSCCTL) 外围使能寄存器 0(PER0) 操作速度模式控制寄存器(OSMC)
振荡器	X1 振荡器 XT1 振荡器 内部高速振荡器 内部低速振荡器

图 5-1. 时钟发生器的框图



备注	f _x :	X1 时钟振荡频率
	f _{IH} :	内部高速振荡时钟频率
	f _{IH20} :	20 MHz 内部高速振荡时钟频率
	f _{EX} :	外部主系统时钟频率
	f _{MX} :	高速系统时钟频率
	f _{MAIN} :	主系统时钟频率
	f _{XT} :	XT1 时钟振荡频率
	f _{SUB} :	副系统时钟频率
	f _{CLK} :	CPU/外围硬件时钟频率
	f _{IL} :	内部低速振荡时钟频率

5.3 控制时钟发生器的寄存器

以下八个寄存器用于控制时钟发生器。

- 时钟操作模式控制寄存器(CMC)
- 时钟操作状态控制寄存器(CSC)
- 振荡稳定时间计数器状态寄存器(OSTC)
- 振荡稳定时间选择寄存器(OSTS)
- 系统时钟控制寄存器(CKC)
- 20 MHz 内部高速振荡控制寄存器(DSCCTL)
- 外围使能寄存器 0 (PER0)
- 操作速度模式控制寄存器(OSMC)

(1) 时钟操作模式控制寄存器(CMC)

此寄存器用于设置 X1/P121、X2/EXCLK/P122、XT1/P123 和 XT2/P124 引脚的操作模式，并选择振荡器。

复位释放后 CMC 只能由 8 位存储器操作指令写入。此寄存器可由 1 位或 8 位存储器操作指令读取。

复位后这个寄存器的值为 00H。

图 5-2. 时钟操作模式控制寄存器(CMC)的格式

地址: FFFA0H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH

EXCLK	OSCSEL	高速系统时钟 引脚操作模式	X1/P121 引脚	X2/EXCLK/P122 引脚
0	0	输入端口模式	输入端口	
0	1	X1 振荡模式	晶体/陶瓷振荡器的连接	
1	0	输入端口模式	输入端口	
1	1	外部时钟输入模式	输入端口	外部时钟输入

OSCSELS	副系统时钟引脚操作模式	XT1/P123 引脚	XT2/P124 引脚
0	输入端口模式	输入端口	
1	XT1 振荡模式	晶体振荡器连接	

AMPHS1	AMPHS0	XT1 振荡器振荡模式选择
0	0	低功耗振荡(默认)
0	1	普通振荡
1	0	超低功耗振荡
1	1	

AMPH	高速系统时钟振荡频率的控制
0	$2\text{ MHz} \leq f_{MX} \leq 10\text{ MHz}$
1	$10\text{ MHz} < f_{MX} \leq 20\text{ MHz}$

- 注意事项**
1. CMC 只能在复位释放后通过 8 位存储器操作指令写入。
 2. 复位释放后, X1 或 XT1 振荡开始前, 通过设置时钟操作状态控制寄存器(CSC)设置 CMC。
 3. 如果 X1 时钟振荡频率超过 10MHz, 确保设置 AMPH 为 1。
 4. 当 CMC 使用的是默认值(00H)时, 为防止程序循环的错误, 确保复位释放后将该寄存器置为 00H。
 5. XT1 振荡器是低振幅电路, 从而实现低功耗。在设计电路时, 需注意以下几点。
 - 引脚和电路板包含寄生电容。因此, 在进行振荡评估时应正确使用电路板并确认一切正常。
 - 当使用超低功耗振荡(AMPHS1 = 1)作为 XT1 振荡器模式时, 使用第 27 章 电气特性(目标值)中推荐的振荡器。
 - 在 XT1、XT2 引脚和振荡器之间布线, 使布线尽可能短, 从而使寄生电容和配线电阻最小。在选择超低功耗振荡(AMPHS1 = 1)时需特别注意。

(注意事项和备注在下页中给出。)

- 使用配线电阻值小的材料来配置电路板的电路。
- 尽可能得在靠近 XT1 振荡器的位置放置一个与 V_{SS} 同电势的地模式。
- 确保信号线在 XT1 和 XT2 引脚之间，并且振荡器不与其他信号线相交。不要在流经较大变化电流的信号线周围布线。
- 当电路板由于处于潮湿环境而吸收了湿气或是板上结有露水时，XT1 和 XT2 引脚之间的阻抗可能变小并且振荡可能受到影响。当在上述环境中使用电路板时，在电路板上采取一些防潮措施，如涂层等。
- 当对电路板进行涂层时，使用不会增加 XT1 和 XT2 引脚之间电容或漏电流的材料。

备注 f_{MX}: 高速系统时钟频率

(2) 时钟操作状态控制寄存器(CSC)

此寄存器用于控制高速系统时钟、内部高速振荡时钟和副系统时钟(除 20MHz 内部高速振荡时钟和内部低速振荡时钟之外)的操作。

CSC 可由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器设置为 C0H。

图 5-3. 时钟操作状态控制寄存器(CSC)的格式

地址: FFFA1H 复位后: C0H R/W

符号	<7>	<6>	5	4	3	2	1	<0>
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP

MSTOP	高速系统时钟操作控制		
	X1 振荡模式	外部时钟输入模式	输入端口模式
0	X1 振荡器工作	EXCLK 引脚的外部时钟有效	输入端口
1	X1 振荡器停止	EXCLK 引脚的外部时钟无效	

XTSTOP	副系统时钟操作控制	
	XT1 振荡模式	输入端口模式
0	XT1 振荡器工作	输入端口
1	XT1 振荡器停止	

HIOSTOP	内部高速振荡时钟操作控制	
0	内部高速振荡器工作	
1	内部高速振荡器停止	

(注意事项在下页中列出。)

- 注意事项**
1. 复位释放后，在设定 CSC 之前，设置时钟操作模式控制寄存器(CMC)。
 2. 要使由 MSTOP 设置的 X1 振荡开始，可通过使用振荡稳定时间计数状态寄存器(OSTC)来检查 X1 时钟的振荡稳定时间。
 3. 不要停止通过 CSC 寄存器为 CPU 外围硬件时钟(f_{CLK})选择的时钟。
 4. 停止时钟振荡(外部时钟输入无效)的寄存器标志设置以及停止时钟振荡前的条件如下所示。

表 5-2. 停止时钟振荡前的条件和标志设置

时钟	停止时钟前的条件 (外部时钟输入无效)	CSC 寄存器标志 的设置
X1 时钟	CPU 和外围硬件时钟工作在除高速系统时钟以外的时钟。 (CLS = 0 和 MCS = 0, 或者 CLS = 1)	MSTOP = 1
外部主系统时钟		
副系统时钟	CPU 和外围硬件时钟工作在除副系统时钟以外的时钟。 (CLS = 0)	XTSTOP = 1
内部高速振荡时钟	CPU 和外围硬件时钟工作在除内部高速振荡器时钟以外的时钟。 (CLS = 0 和 MCS = 1, 或者 CLS = 1)	HIOSTOP = 1

(3) 振荡稳定时间计数状态寄存器(OSTC)

这是表明 X1 时钟振荡稳定时间计数器的计数状态的寄存器。

在下列情况下，可检测 X1 时钟振荡稳定时间，

- 在内部高速振荡时钟或副系统时钟用作 CPU 时钟期间，如果 X1 时钟开始振荡。
- 当内部高速振荡时钟用作 X1 时钟振荡的 CPU 时钟时，如果进入 STOP 模式然后释放。

可由 1 位或 8 位存储器操作指令读取 OSTC。

复位信号产生时，STOP 指令或 MSTOP(CSC 寄存器的位 7)= 1 将 OSTC 清为 00H。

备注 振荡稳定时间计数器在以下情况下开始计数。

- 当 X1 时钟开始振荡时(EXCLK, OSCSEL = 0, 1 → MSTOP = 0)
- 当 STOP 模式释放时

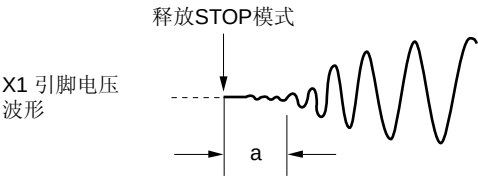
图 5-4. 振荡稳定时间计数器状态寄存器(OSTC)的格式

地址: FFFA2H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	振荡稳定时间的状态		
									$f_x = 10\text{ MHz}$	$f_x = 20\text{ MHz}$
0	0	0	0	0	0	0	0	$2^8/f_x\text{ max.}$	$25.6\text{ }\mu\text{s max.}$	$12.8\text{ }\mu\text{s max.}$
1	0	0	0	0	0	0	0	$2^8/f_x\text{ min.}$	$25.6\text{ }\mu\text{s min.}$	$12.8\text{ }\mu\text{s min.}$
1	1	0	0	0	0	0	0	$2^9/f_x\text{ min.}$	$51.2\text{ }\mu\text{s min.}$	$25.6\text{ }\mu\text{s min.}$
1	1	1	0	0	0	0	0	$2^{10}/f_x\text{ min.}$	$102.4\text{ }\mu\text{s min.}$	$51.2\text{ }\mu\text{s min.}$
1	1	1	1	0	0	0	0	$2^{11}/f_x\text{ min.}$	$204.8\text{ }\mu\text{s min.}$	$102.4\text{ }\mu\text{s min.}$
1	1	1	1	1	0	0	0	$2^{13}/f_x\text{ min.}$	$819.2\text{ }\mu\text{s min.}$	$409.6\text{ }\mu\text{s min.}$
1	1	1	1	1	1	0	0	$2^{15}/f_x\text{ min.}$	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	$2^{17}/f_x\text{ min.}$	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/f_x\text{ min.}$	26.21 ms min.	13.11 ms min.

- 注意事项
1. 在经历上述时间后，各个位从 MOST8 起按顺序被设置为 1，并保持为 1。
 2. 振荡稳定时间计数器的计数达到 OSTs 设置的振荡稳定时间。
在如下情况，设置 OSTs 的振荡稳定时间的值大于振荡开始后 OSTC 寄存器检测到的计数值。
 - 在内部高速振荡时钟或副系统时钟用作CPU时钟期间，如果X1时钟开始振荡。
 - 当内部高速振荡时钟用作X1时钟振荡的CPU时钟时，如果进入STOP模式然后释放。(注，而且，仅由OSTs设置的振荡稳定时间状态在STOP模式释放后被设置到OSTC)
 3. X1 时钟振荡稳定等待时间不包含直到时钟振荡开始的时间(下为“a”)。



备注 f_x : X1 时钟振荡频率

(4) 振荡稳定时间选择寄存器(OSTS)

此寄存器用于选择 STOP 模式释放时的 X1 时钟振荡稳定等待时间。

当选择 X1 时钟作为 CPU 时钟时，STOP 模式释放后操作自动等待一个由 OSTS 设置的时间。

当释放 STOP 模式后，且 CPU 使用内部高速振荡时钟时，可通过使用 OSTC 确认是否已经历了预期的振荡稳定时间。使用 OSTC，振荡稳定时间可被检测是否等于 OSTC 设置的时间。

可由 8 位存储器操作指令设置 OSTS。

复位信号产生将 OSTS 置为 07H。

图 5-5. 振荡稳定时间选择寄存器(OSTS)的格式

地址: FFFA3H 复位后: 07H R/W

符号	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	振荡稳定时间选择		
				$f_x = 10\text{ MHz}$	$f_x = 20\text{ MHz}$
0	0	0	$2^8/f_x$	$25.6\text{ }\mu\text{s}$	设置禁止
0	0	1	$2^9/f_x$	$51.2\text{ }\mu\text{s}$	$25.6\text{ }\mu\text{s}$
0	1	0	$2^{10}/f_x$	$102.4\text{ }\mu\text{s}$	$51.2\text{ }\mu\text{s}$
0	1	1	$2^{11}/f_x$	$204.8\text{ }\mu\text{s}$	$102.4\text{ }\mu\text{s}$
1	0	0	$2^{13}/f_x$	$819.2\text{ }\mu\text{s}$	$409.6\text{ }\mu\text{s}$
1	0	1	$2^{15}/f_x$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_x$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_x$	26.21 ms	13.11 ms

- 注意事项
1. 当 CPU 时钟使用 X1 时钟时, 若要设置 STOP 模式, 则必须在执行 STOP 指令之前设置 OSTS。

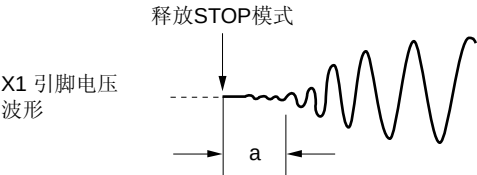
2. 禁止设置振荡稳定时间为 $20\text{ }\mu\text{s}$ 或更短。

3. 要改变 OSTS 寄存器的设置, 必须确保 OSTC 寄存器的计数操作已经完成。

4. 在 X1 时钟振荡稳定期间, 不要修改 OSTS 寄存器的值。

5. 振荡稳定时间计数器的计数达到 OSTS 设置的振荡稳定时间。
在如下情况, 设置 OSTS 的振荡稳定时间的值大于振荡开始后 OSTC 寄存器检测到的计数值。
 - 在内部高速振荡时钟或副系统时钟用作CPU时钟期间, 如果X1时钟开始振荡。
 - 当内部高速振荡时钟用作X1时钟振荡的CPU时钟时, 如果进入STOP模式然后释放。
(注, 而且, 仅由OSTS设置的振荡稳定时间状态在STOP模式释放后被设置到OSTC)

6. X1 时钟振荡稳定等待时间不包含直到时钟振荡开始的时间(下为“a”)。



备注 f_x : X1 时钟振荡频率

(5) 系统时钟控制寄存器(CKC)

此寄存器用于选择 CPU/外围硬件时钟和分频比。

CKC 可由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器置为 09H。

图 5-6. 系统时钟控制寄存器(CKC)的格式

地址: FFFA4H 复位后: 09H R/W^{注1}

符号	<7>	<6>	<5>	<4>	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	1	MDIV2	MDIV1	MDIV0

CLS	CPU/外围硬件时钟(f _{CLK})的状态
0	主系统时钟 (f _{MAIN})
1	副系统时钟 (f _{SUB})

MCS	主系统时钟(f _{MAIN})的状态
0	内部高速振荡时钟(f _{IH})
1	高速系统时钟(f _{MX})

CSS	MCM0	MDIV2	MDIV1	MDIV0	CPU/外围硬件时钟(f _{CLK})的选择
0	0	0	0	0	f _{IH}
		0	0	1	f _{IH} /2(默认)
		0	1	0	f _{IH} /2 ²
		0	1	1	f _{IH} /2 ³
		1	0	0	f _{IH} /2 ⁴
		1	0	1	f _{IH} /2 ⁵
0	1	0	0	0	f _{MX}
		0	0	1	f _{MX} /2
		0	1	0	f _{MX} /2 ²
		0	1	1	f _{MX} /2 ³
		1	0	0	f _{MX} /2 ⁴
		1	0	1	f _{MX} /2 ⁵ 注2
1注3	×注3	×	×	×	f _{SUB} /2
其它值					设置禁止

- 注
1. 第 7 位和第 5 位是只读的。
 2. f_{MX} < 4 MHz 时设置禁止。
 3. 当 CSS 设置为 1 时，禁止改变 MCM0 位的值。

- 备注
1. f_{IH}: 内部高速振荡时钟频率
f_{MX}: 高速系统时钟频率
f_{SUB}: 副系统时钟频率
 2. ×: 不必考虑

(注意事项在下页中列出。)

- 注意事项
1. 第 3 位必须设置为 1。
 2. 由 CSS、MCM0 和 MDIV2 至 MDIV0 设置的时钟供给 CPU 和外围硬件。如果 CPU 时钟改变，供给外围硬件(除实时计数器，时钟输出/蜂鸣器输出和看门狗定时器外)的时钟也同时改变。因此，当 CPU/外围硬件时钟改变时，停止每个外围的功能。
 3. 如果把外围硬件时钟用作副系统时钟，则我们不能保证 A/D 转换器和 IICA 的操作。外围硬件的操作特性，参见描述各种外围硬件的章节及第 27 章 电气特性(目标值)。

78K0R/KG3-L 中最快的指令可以在 CPU 时钟的 1 个时钟之内执行。因此，表 5-3 列出了 CPU 时钟(f_{CLK})和指令最短执行时间之间的关系。

表 5-3. CPU 时钟和最小指令执行时间之间的关系

CPU 时钟 (由 MDIV2 至 MDIV0 位设置的值)	最小指令执行时间: $1/f_{CLK}$			
	主系统时钟 (CSS = 0)			副系统时钟 (CSS = 1)
	高速系统时钟 (MCM0 = 1)		内部高速振荡时钟 (MCM0 = 0)	
	操作在 10 MHz	操作在 20 MHz	在 8 MHz (典型值) 操作	
f_{MAIN}	0.1 μs	0.05 μs	0.125 μs (典型值)	—
$f_{MAIN}/2$	0.2 μs	0.1 μs	0.25 μs (典型值) (默认)	—
$f_{MAIN}/2^2$	0.4 μs	0.2 μs	0.5 μs (典型值)	—
$f_{MAIN}/2^3$	0.8 μs	0.4 μs	1.0 μs (典型值)	—
$f_{MAIN}/2^4$	1.6 μs	0.8 μs	2.0 μs (典型值)	—
$f_{MAIN}/2^5$	3.2 μs	1.6 μs	4.0 μs (典型值)	—
$f_{SUB}/2$	—		—	61 μs

备注 f_{MAIN} : 主系统时钟频率(f_{IH} 或 f_{MX})
 f_{SUB} : 副系统时钟频率

(6) 20 MHz 内部高速振荡控制寄存器(DSCCTL)

此寄存器控制 20MHz 内部高速振荡时钟(DSC)功能。
它可用于选择是否使用 20MHz 内部高速振荡时钟(f_{IH20})作为外围硬件时钟，该外围硬件时钟支持 20MHz。
DSCCTL 只能由 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 00H。

图 5-7. 20 MHz 内部高速振荡控制寄存器(DSCCTL)的格式

地址: F00F6H 复位后: 00H R/W^注

符号	7	6	5	4	<3>	<2>	1	<0>
DSCCTL	0	0	0	0	DSCS	SELDSC	0	DSCON

DSCS	20 MHz 内部高速振荡状态标志
0	不提供
1	提供

SELDSC	选择 20 MHz 的内部高速振荡用于 CPU/外围硬件时钟(f_{CLK})
0	不选择 20 MHz 内部高速振荡(由 CKC 寄存器选择的时钟提供给 f_{CLK})。
1	选择 20 MHz 内部高速振荡(20 MHz 内部高速振荡提供给 f_{CLK})。

DSCON	20MHz 内部高速振荡时钟(f_{IH20})运行使能/禁止
0	禁止操作。
1	使能操作。

注 第 3 位只读。

- 注意事项
- 1. 仅当 $V_{DD} \geq 2.7V$ 时，可使用 20 MHz 内部振荡。
 - 2. 在 $V_{DD} \geq 2.7 V$ 下，设置 DSCON 后等待 100 μ s 后设置 SELDSC。
 - 3. 当 DSCON=1 时，内部高速振荡器必须在运行(HIOSTOP = 0)。

(7) 外围使能寄存器 0 (PER0)

该寄存器用于使能或禁止使用每个外围硬件模块。为了降低功耗和噪声，不使用的硬件时钟也被停止。
可以通过 1 位或者 8 位存储器操作指令来设置 PER0。
复位信号产生清寄存器为 00H。

图 5-8. 外围使能寄存器 0 (PER0)的格式(1/2)

地址: F00F0H 复位后: 00H R/W

符号	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

RTCEN	提供实时计数器(RTC)输入时钟的控制 ^注
0	停止输入时钟供给。 - 由实时计数器(RTC)使用的 SFR 不能写入。 - 实时计数器(RTC)在复位状态。
1	供给输入时钟 用于实时计数器(RTC)的 SFR 可读写。

ADCEN	提供 A/D 转换器输入时钟的控制
0	停止输入时钟供给。 - 被 A/D 转换器使用的 SFR 是不可重写。 - A/D 转换器在复位状态。
1	供给输入时钟 用于 A/D 转换器的 SFR 可读写。

IICAEN	提供串行接口 IICA 输入时钟的控制
0	停止输入时钟供给。 - 用于串行接口 IICA 的 SFR 不能写入。 - 串行接口 IICA 在复位状态。
1	供给输入时钟 用于串行接口 IICA 的 SFR 可读写。

注 当通过使用寄存器从CPU访问实时计数器(RTC)时，可通过RTCEN控制输入时钟。RTCEN不能控制供给RTC的操作时钟 (f_{SUB})。

注意事项 确保将位 6 清零。

图 5-8. 外围使能寄存器 0 (PER0)的格式(2/2)

地址: F00F0H 复位后: 00H R/W

符号	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

SAU1EN	提供串行阵列单元 1 输入时钟的控制
0	停止输入时钟供给。 - 不能写入串行阵列单元 1 使用的 SFR。 - 串行阵列单元 1 处于复位状态。
1	供给输入时钟 可以对串行阵列单元 1 使用的 SFR 进行读写。

SAU0EN	提供串行阵列单元 0 输入时钟的控制
0	停止输入时钟供给。 - 不能写入串行阵列单元 0 使用的 SFR。 - 串行阵列单元 0 处于复位状态。
1	供给输入时钟 可以对串行阵列单元 0 使用的 SFR 进行读写。

TAU1EN	提供定时器阵列单元 1 输入时钟的控制
0	停止输入时钟供给。 - 由定时器阵列单元 1 使用的 SFR 不能写入。 - 定时器阵列单元 1 处于复位状态。
1	供给输入时钟 可以对定时器阵列单元 1 使用的 SFR 进行读写。

TAU0EN	提供定时器阵列单元 0 输入时钟的控制
0	停止输入时钟供给。 - 由定时器阵列单元 0 使用的 SFR 不能写入。 - 定时器阵列单元 0 处于复位状态。
1	供给输入时钟 可以对定时器阵列单元 0 使用的 SFR 进行读写。

注意事项 确保将位 6 清零。

(8) 操作速度模式控制寄存器(OSMC)

FLPC 和 FSEL 用于控制 flash 存储器高速操作的递升电路。

如果微控制器在系统时钟 10MHz 或更小值的低速下操作，通过设置此寄存器到默认值(00H)可以降低功耗。此外，当运行在 1 MHz 系统时钟时，可通过设置 FLPC 为 1 来进一步降低功耗。

RTCLPC 可用来设置副系统时钟下的操作。

OSMC 可由 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 5-9. 操作速度模式控制寄存器(OSMC)的格式

地址: F00F3H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	0	0	0	FLPC	FSEL

FLPC	FSEL	f _{CLK} 频率选择
0	0	操作频率小于等于 10 MHz(默认)。
0	1	操作频率大于 10 MHz。
1	0	操作频率大于 1 MHz。
1	1	设置禁止

RTCLPC	设置副系统时钟 HALT 模式
0	使能向外围功能提供副系统时钟 (关于使能操作的外围功能，参见表 17-1。)
1	除实时计数器外，停止向外设功能提供副系统时钟

- 注意事项
1. 在执行下列两个操作之前，把“1”写入 FSEL。
 - 改变时钟优先级，除 f_{IH} 之外分频 f_{CLK} 为一个时钟。
 - 操作 DMA 控制器。
 2. 当向 FSEL 标志写入“1”时，CPU 等待。
当 f_{CLK} = f_{IH} 时等待时间为 15 μs 至 20 μs(目标值)，当 f_{CLK} = f_{IH}/2 时等待时间为 30 μs 至 40 μs (目标值)。
但是，即使当 CPU 等待时，f_x 的振荡稳定时间计数可继续。
 3. 为了增加 f_{CLK} 至 10 MHz 或更高，设置 FSEL 为“1”，然后在 2 个或更多时钟之后改变 f_{CLK}。
 4. 当时钟工作在 10 MHz 或更低时，设置 FSEL = 0。
 5. 当 V_{DD} ≤ 2.7 V 时，通过设置 FSEL = 0 转移到 STOP 模式。

(注意事项在下页列出。)

- 注意事项
6. 当使用副系统时钟时，可通过设置 **RTCLPC** 为 **1** 来降低 **HALT** 模式的电流。但是，在副系统时钟 **HALT** 模式下，除实时定时器外，不向外围功能提供时钟。在设置副系统时钟 **HALT** 模式之前，将 **PER0** 的位 **7(RTCEN)** 设为 **1**，并且将 **PER0** 的位 **0** 至 **6** 设为 **0**。
 7. 一旦 **FLPC** 由 **0** 设为 **1**，除非复位，否则禁止将其从 **1** 设回 **0**。

5.4 系统时钟振荡器

5.4.1 X1 振荡器

X1 振荡器采用将一个晶体振荡器或陶瓷振荡器(2 至 20MHz)连接到 X1 和 X2 引脚的方式振荡。

同样可以输入外部时钟。此时，把时钟信号输入到 EXCLK 引脚。

要使用 X1 振荡器，按如下方式设置时钟操作模式控制寄存器(CMC)的第 7 和第 6 位(EXCLK、OSCSEL)。

- 晶体或陶瓷振荡：EXCLK，OSCSEL = 0，1
- 外部时钟输入：EXCLK，OSCSEL = 1，1

当 X1 振荡器不使用时，设置输入端口模式(EXCLK，OSCSEL = 0，0)。

当引脚也不用作输入端口引脚时，参见表 2-2 未使用引脚的连接。

图 5-10 为 X1 振荡器的外部电路示例。

图 5-10. X1 振荡器的外部电路的示例



注意事项在下页中列出。

5.4.2 XT1 振荡器

XT1 振荡器以将一个晶体振荡器(标准：32.768 kHz)连接到 XT1 和 XT2 的方式振荡。

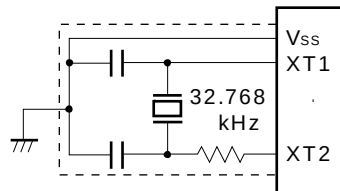
要使用 XT1 振荡器，设置时钟操作模式控制寄存器(CMC)的第 4 位(OSCSELS)为 1。

当 XT1 振荡器不使用时，设置输入端口模式(OSCSELS = 0)。

当引脚也不用作输入端口引脚时，参见表 2-2 未使用引脚的连接。

图 5-11 为 XT1 振荡器的外部电路示例。

图 5-11. XT1 振荡器(晶体振荡)的外部电路示例



注意事项在下页中列出。

注意事项 1. 在使用 X1 振荡器和 XT1 振荡器时，为防止布线电容的不利影响，图 5-10 和图 5-11 中虚线包围的部分应按照如下方法配线。

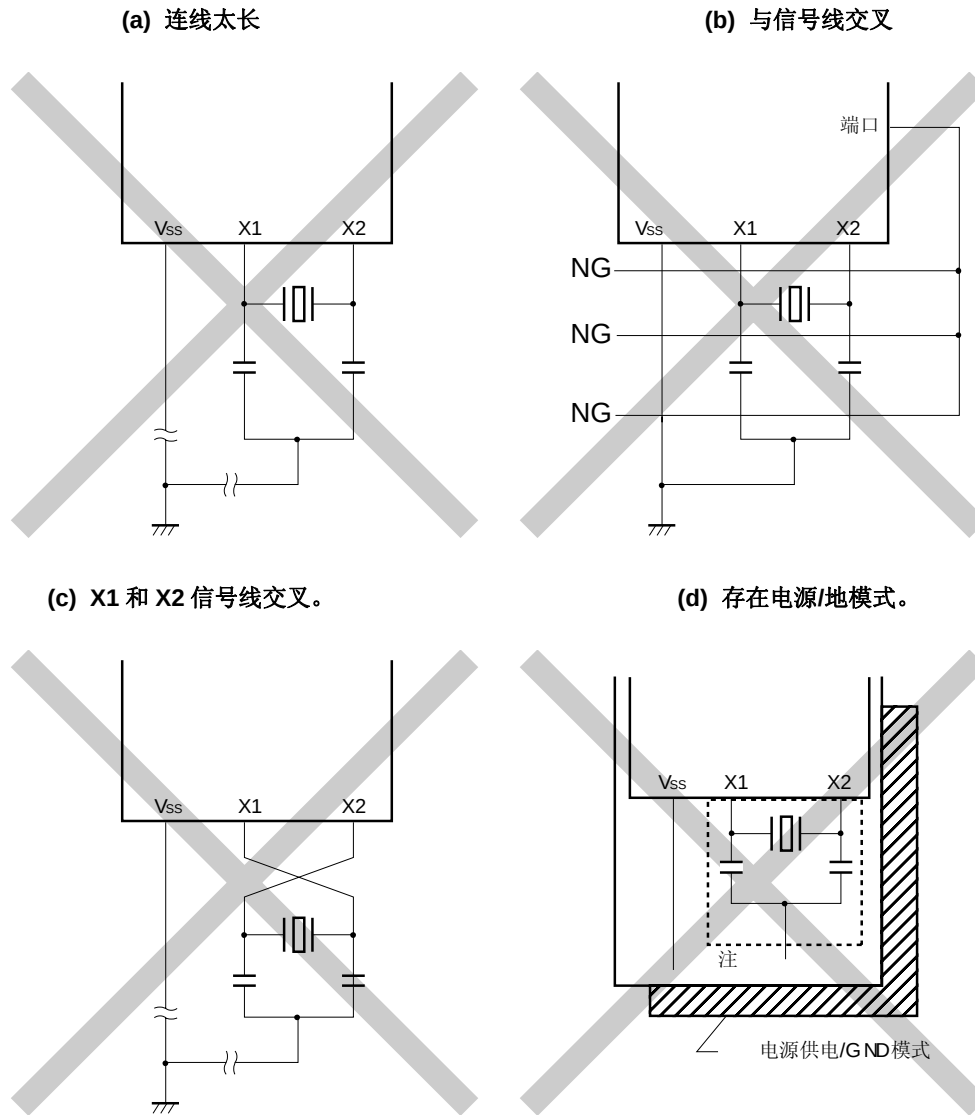
- 连接线越短越好。
- 连接线不应与其它信号线交叉。不要在流经较大变化电流的信号线周围布线。
- 始终保持振荡器电容的接地点电压与 VSS 相同。不要将电容的地模式接入大电流地。
- 不要从振荡器获取信号。

XT1 振荡器是低振幅电路，从而实现低功耗。在设计电路时，需注意以下几点。

- 引脚和电路板包含寄生电容。因此，在进行振荡评估时应正确使用电路板并确认一切正常。
- 当使用超低功耗振荡(AMPHS1 = 1)作为 XT1 振荡器模式时，使用第 27 章 电气特性(目标值)中推荐的振荡器。
- 在 XT1、XT2 引脚和振荡器之间布线，使布线尽可能短，从而使寄生电容和配线电阻最小。在选择超低功耗振荡(AMPHS1 = 1)时需特别注意。
- 使用配线电阻值小的材料来配置电路板的电路。
- 尽可能得在靠近 XT1 振荡器的位置放置一个与 VSS 同电势的地模式。
- 确保信号线在 XT1 和 XT2 引脚之间，并且振荡器不与其他信号线相交。不要在流经较大变化电流的信号线周围布线。
- 当电路板由于处于潮湿环境而吸收了湿气或是板上结有露水时，XT1 和 XT2 引脚之间的阻抗可能变小并且振荡可能受到影响。当在上述环境中使用电路板时，在电路板上采取一些防潮措施，如涂层等。
- 当对电路板进行涂层时，使用不会增加 XT1 和 XT2 引脚之间电容或漏电流的材料。

图 5-12 为不正确的振荡器连接示例。

图 5-12. 不正确的振荡器连接示例(1/2)

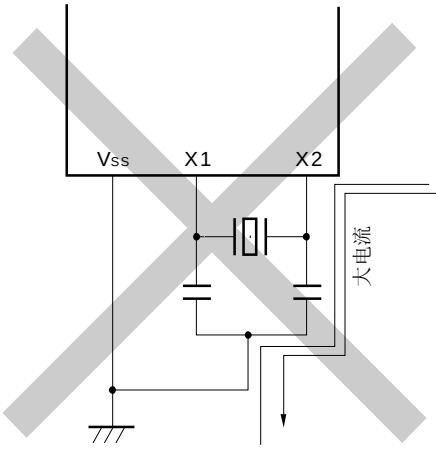
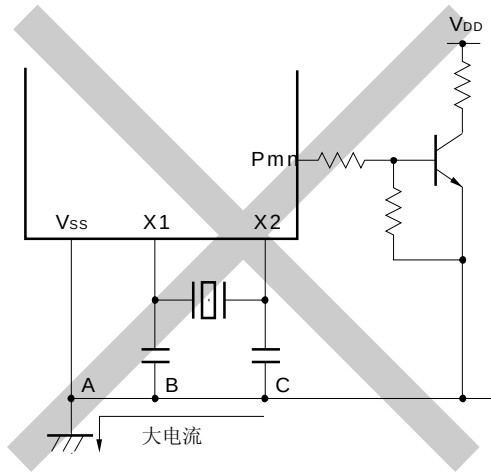


注 在多层板或双面板上，不要在 X1、X2 引脚和谐振器的布线部分(该部分在图中由虚线表示)放置电源/地模式。布局配置时不应产生电容因素及影响振荡器特性。

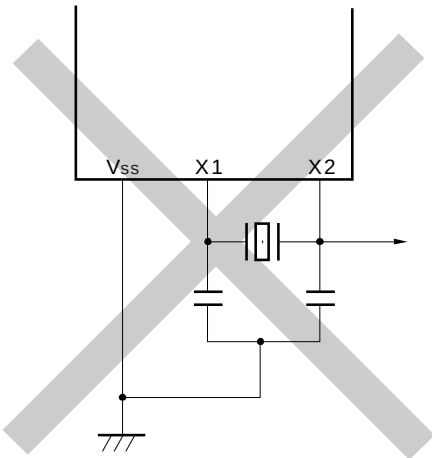
备注 在使用副系统时钟时，分别用 XT1 和 XT2 代替 X1 和 X2。串联电阻也接在 XT2 这边。

图 5-12. 不正确的振荡器连接示例(2/2)

(e) 在大交流电附近布线

(f) 电流经过振荡器的地线
(A、B、C 三点电势波动)

(g) 获取信号



备注 在使用副系统时钟时，分别用 XT1 和 XT2 代替 X1 和 X2。串联电阻也接在 XT2 这边。

注意事项 2. 当 X2 和 XT1 并行连接时，X2 的串扰噪声会叠加到 XT1，从而产生故障。

5.4.3 内部高速振荡器[‡]

78K0R/KG3-L(1.8 MHz(典型值))产品中包含内部高速振荡器。可以通过时钟操作状态控制寄存器(CSC)的第 0 位(HIOSTOP)控制振荡。

5.4.4 20MHz内部高速振荡器[‡]

78K0R/KG3-L(20MHz(典型值))产品中包含 20MHz 内部高速振荡器。当 $V_{DD} \geq 2.7\text{ V}$ 时，可以通过 20MHz 内部高速振荡器控制寄存器(DSCCTL)的位 0(DSCON)来控制振荡。

复位释放后，通过设置 DSCCTL 寄存器的位 0(DSCON)为 1 使 20MHz 内部高速振荡器开始振荡。

注 使用 1、8 或 20MHz 的内部高速振荡时钟，要使用选项字节预先设置频率(详细内容，参考第 22 章 选项字节)此外，复位释放后，内部高速振荡器自动开始振荡。要使用 20 MHz 内部高速振荡器来操作微控制器，通过设置 20 MHz 内部高速振荡控制寄存器(DSCCTL)的位 0(DSCON)为 1 启动振荡并使得 $V_{DD} \geq 2.7\text{ V}$ 。

5.4.5 内部低速振荡器

78K0R/KG3-L 产品中包含内部低速振荡器。

内部低速振荡时钟只作为看门狗定时器的时钟使用。内部低速振荡时钟不能用作 CPU 时钟。

复位释放后，内部低速振荡器自动产生振荡，同时如果使用选项字节使能看门狗定时器操作，则可以驱动看门狗定时器(30 kHz(典型值))。

除非看门狗定时器停止，否则内部低速振荡器继续振荡。当看门狗定时器工作时，即使程序循环，内部低速振荡时钟也不停止。

5.4.6 预分频器

通过主系统时钟和副系统时钟分频，预分频器产生一个 CPU/外围硬件时钟。

5.5 时钟发生器操作

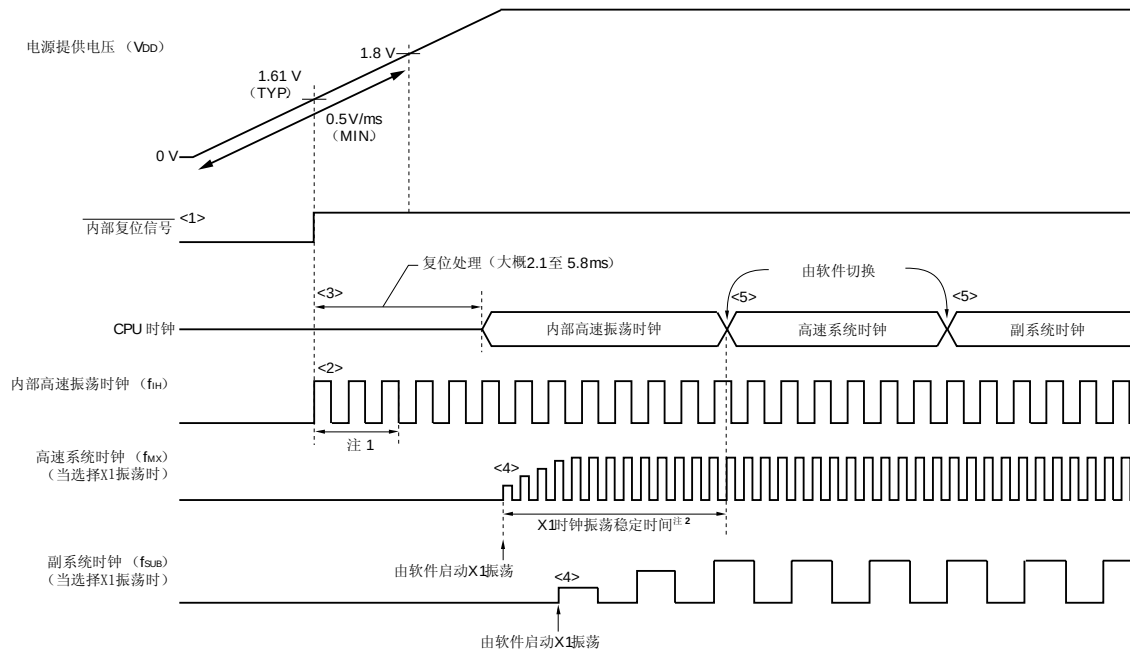
时钟发生器用于产生以下几种时钟，并控制 CPU 的操作模式，例如待机模式(参见 图 5-1)。

- 主系统时钟 f_{MAIN}
 - 高速系统时钟 f_{MX}
 - X1 时钟 f_x
 - 外部主系统时钟 f_{EX}
 - 内部高速振荡时钟 f_{IH}
 - 20 MHz 内部高速振荡时钟 f_{IH20}
- 副系统时钟 f_{SUB}
- 内部低速振荡时钟 f_{IL}
- CPU/外围硬件时钟 f_{CLK}

在 78K0R/KG3-L 中，复位释放后当内部高速振荡器开始输出时，CPU 开始操作。

图 5-13 至图 5-16 显示了供电电源打开时，时钟发生器的操作。

图 5-13. 供电电源打开时钟发生器的操作
(当设置 LVI 默认开始功能停止(选项字节: LVIOFF = 1)时)



- <1> 当电源开启时，通过上电清零 (POC) 电路产生一个内部复位信号。
- <2> 当供电电压超过 1.61 V(典型值)，复位释放并且内部高速振荡器自动开始振荡。
- <3> 当供电电压上升斜率为 0.5 V/ms (最小值)时，在复位过程(比如等待供电电压或复位释放后稳压器稳定)后，CPU 在内部高速振荡时钟上开始操作。
- <4> 通过软件设置 X1 或 XT1 时钟的振荡开始(参见 5.6.1 高速系统时钟控制示例中的(1)和 5.6.3 副系统时钟控制示例中的(1))。
- <5> 当 CPU 时钟转换到 X1 或 XT1 时钟时，等待时钟振荡稳定，然后通过软件设置转换(参见 5.6.1 高速系统时钟控制示例中的(3)和 5.6.3 副系统时钟控制示例中的(2))。

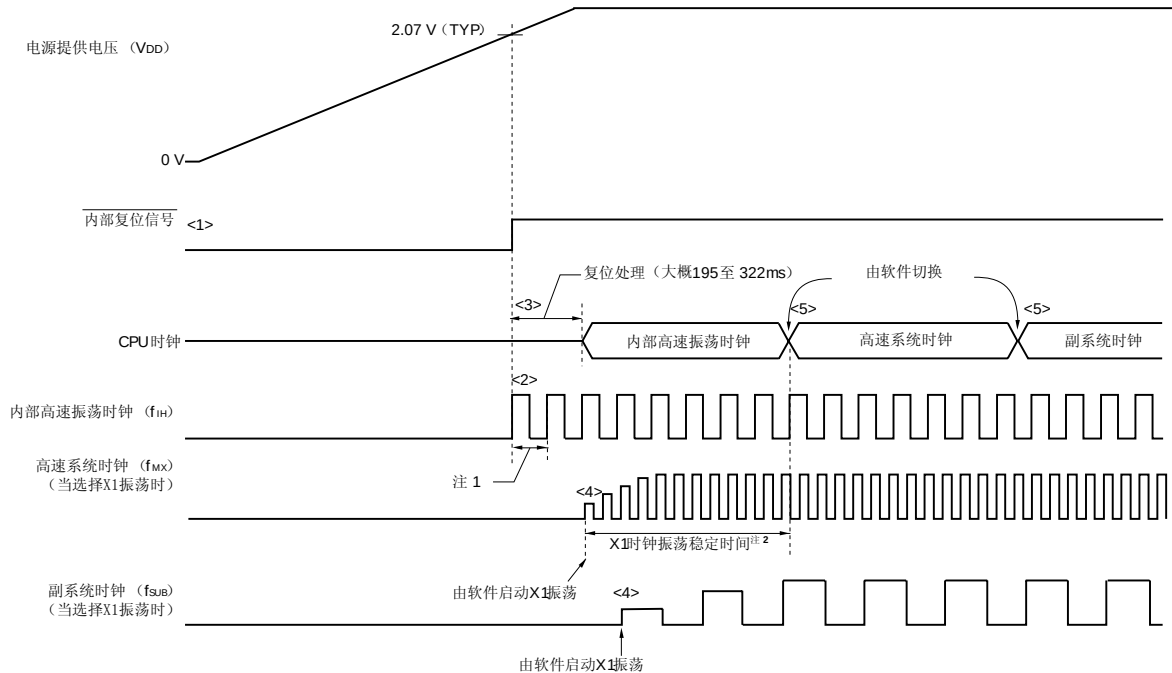
- 注
1. 内部复位处理时间包括内部高速振荡时钟的振荡精确稳定时间。
 2. 释放复位后(如上图所示)或在 CPU 使用内部高速振荡时钟时释放 STOP 模式后，使用振荡稳定时间计数器的状态寄存器(OSTC)来确认 X1 时钟的振荡稳定时间。如果 CPU 使用高速系统时钟(X1 振荡)，则可使用振荡稳定时间选择寄存器(OSTS)设置释放 STOP 模式时的振荡稳定时间。

注意事项 1. 在供电电压达到 1.8V 之前, 如果电压上升斜率小于 0.5 V/ms (MIN.), 则从供电开始至电压达到 1.8V 为止, 输入一个低电平到电源的 $\overline{\text{RESET}}$ 引脚, 或者通过使用选项字节(LVIOFF = 0)设置 LVI 默认开始功能停止(见图 5-14)。通过这种方式, CPU 的操作时序与 <2> 及由 $\overline{\text{RESET}}$ 引脚释放复位后的时序相同, 如图 5-13 所示。

2. 当使用 EXCLK 引脚的外部时钟输入时, 不需要等待振荡稳定时间。

备注 当微控制器正在操作时, 不作为 CPU 时钟使用的时钟可以通过软件设置来停止。内部高速振荡时钟和高速系统时钟可以通过执行 STOP 指令来停止(参见 5.6.1 高速系统时钟控制示例中的(4)、5.6.2 内部高速振荡时钟控制示例中的(3)和 5.6.3 副系统时钟控制示例中的(3))。

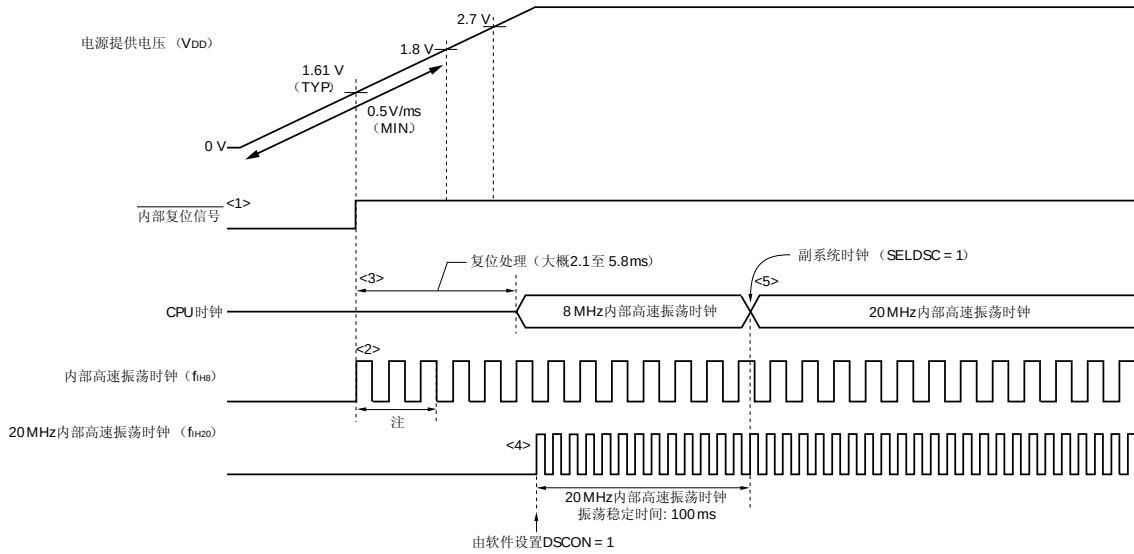
图 5-14. 供电电源打开时时钟发生器的操作
(当设置 LVI 默认开始功能使能时(选项字节: LVIOFF = 0)时)



- <1> 当电源开启时, 通过低电压检测器 (LVI) 电路产生一个内部复位信号。
- <2> 当供电电压超过 2.07 V(典型值), 复位释放并且内部高速振荡器自动开始振荡。
- <3> 复位释放并进行了复位处理后, CPU 开始使用内部高速振荡时钟操作。
- <4> 通过软件设置 X1 或 XT1 时钟的振荡开始(参见 5.6.1 高速系统时钟控制示例中的(1)和 5.6.3 副系统时钟控制示例中的(1))。
- <5> 当 CPU 时钟转换到 X1 或 XT1 时钟时, 等待时钟振荡稳定, 然后通过软件设置转换(参见 5.6.1 高速系统时钟控制示例中的(3)和 5.6.3 副系统时钟控制示例中的(2))。

- 注**
1. 内部复位处理时间包括内部高速振荡时钟的振荡精确稳定时间。
 2. 释放复位后(如上图所示)或在 CPU 使用内部高速振荡时钟时释放 STOP 模式后, 使用振荡稳定时间计数器的状态寄存器(OSTC)来确认 X1 时钟的振荡稳定时间。如果 CPU 使用高速系统时钟(X1 振荡), 则可使用振荡稳定时间选择寄存器(OSTS)设置释放 STOP 模式时的振荡稳定时间。
- 注意事项**
1. 在供电电压达到 1.61(典型值)后, 必需有一个电压稳定时间(大约为 2.0 至 5.8 ms)。如果供电电压从 1.61 V(典型值)上升到 2.07 V(典型值)所用的时间小于电压稳定时间, 那么在经过电压稳定时间之后将进入复位过程。
 2. 当使用 EXCLK 引脚的外部时钟输入时, 不需要等待振荡稳定时间。
- 备注**
- 当微控制器正在操作时, 不作为 CPU 时钟使用的时钟可以通过软件设置来停止。内部高速振荡时钟和高速系统时钟可以通过执行 STOP 指令来停止(参见 5.6.1 高速系统时钟控制示例中的(4)、5.6.2 内部高速振荡时钟控制示例中的(3)和 5.6.3 副系统时钟控制示例中的(3))。

图 5-15. 供电电源开启时时钟发生器的操作
(当设置 LVI 默认开始功能停止(选项字节: LVIOFF = 1)且改变至 20 MHz 内部高速振荡时钟时)

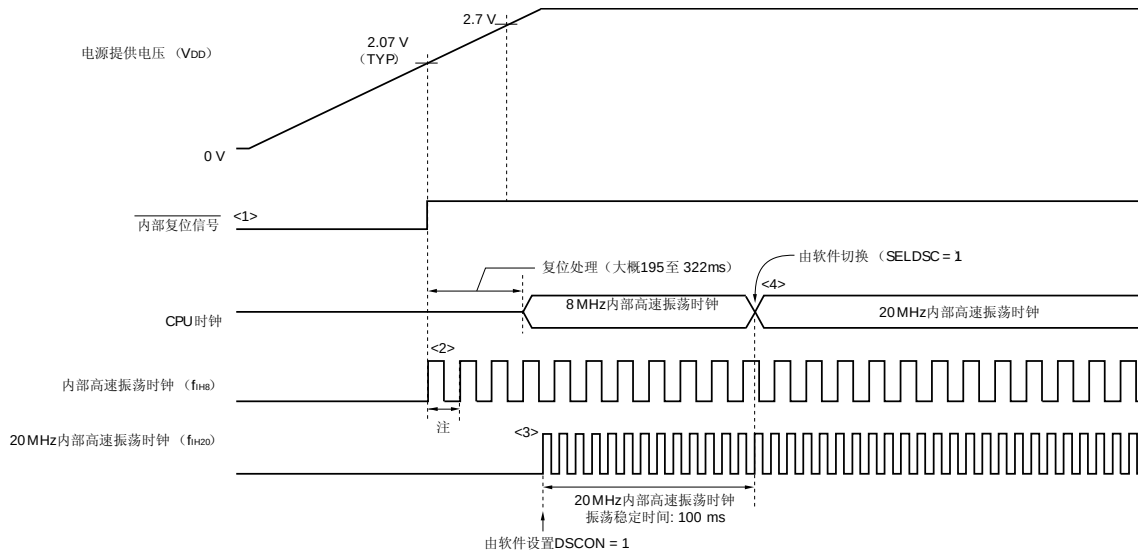


- <1> 当电源开启时，通过上电清零 (POC) 电路产生一个内部复位信号。
- <2> 当供电电压超过 1.61 V(典型值)，复位释放并且内部高速振荡器自动开始振荡。
- <3> 当供电电压上升斜率为 0.5 V/ms (最小值)时，在复位过程(比如等待供电电压或复位释放后稳压器稳定)后，CPU 在内部高速振荡时钟上开始操作。
- <4> 确认电源电压 ≥ 2.7 V，并通过软件设置 DSCON = 1。
- <5> 在等待 100 μ s 后，通过软件设置 SELDSC = 1 来转换时钟。

注 内部复位处理时间包括内部高速振荡时钟的振荡精确稳定时间。

- 注意事项 1. 使用 20MHz 的内部高速振荡时钟，要使用选项字节(000C1H)的第 2 位和第 1 位(FRQSEL2 和 FRQSEL1)预先设置频率为 20MHz(详细内容，参考第 22 章 选项字节)。
2. 在供电电压达到 1.8V 之前，如果电压上升斜率小于 0.5 V/ms (MAX.)，则从供电开始至电压达到 1.8V 为止，输入一个低电平到电源的 RESET 引脚，或者通过使用选项字节(LVIOFF = 0)设置 LVI 默认开始功能停止(见图 5-16)。通过这种方式，CPU 的操作时序与 <2> 及由 RESET 引脚释放复位后的时序相同，如图 5-15 所示。

图 5-16. 供电电源开启时时钟发生器的操作
(当设置 LVI 默认开始功能使能(选项字节: LVIOFF = 0)且更改至 20MHz 内部高速振荡时钟时)



- <1> 当电源开启时, 通过低电压检测器 (LVI) 电路产生一个内部复位信号。
- <2> 当供电电压超过 2.07 V(典型值), 复位释放并且内部高速振荡器自动开始振荡。
- <3> 复位释放并进行了复位处理后, CPU 开始使用内部高速振荡时钟操作。
- <4> 确认电源电压 ≥ 2.7 V, 并通过软件设置 DSCON = 1。
- <5> 在等待 100 μ s 后, 通过软件设置 SELDSC = 1 来转换时钟。

注 内部复位处理时间包括内部高速振荡时钟的振荡精确稳定时间。

- 注意事项
1. 使用 20MHz 的内部高速振荡时钟, 要使用选项字节(000C1H)的第 2 位和第 1 位(FRQSEL2 和 FRQSEL1)预先设置频率为 20MHz(详细内容, 参考第 22 章 选项字节)。
 2. 在供电电压达到 1.61(典型值)后, 必须有一个电压稳定时间(大约为 2.0 至 5.8 ms)。如果供电电压从 1.61 V(典型值)上升到 2.07 V(典型值)所用的时间小于电压稳定时间, 那么在经过电压稳定时间之后将进入复位过程。

5.6 控制时钟

5.6.1 控制高速系统时钟的示例

如下两种高速系统时钟可用。

- X1 时钟：晶体/陶瓷振荡器连接到 X1 和 X2 引脚。
- 外部主系统时钟：外部时钟输入到 EXCLK 引脚。

不使用高速系统时钟时，X1/P121 和 X2/EXCLK/P122 引脚可用作输入端口引脚。

注意事项 X1/P121 和 X2/EXCLK/P122 引脚复位释放后为输入端口模式。

下面描述的示例是以下情况时的设置过程示例。

- (1) 振荡 X1 时钟时
- (2) 当使用外部主系统时钟时
- (3) 当使用高速系统时钟作为 CPU/外围硬件时钟时
- (4) 停止高速系统时钟时

(1) X1 时钟振荡时的设置过程示例

<1> 设置 P121/X1 和 P122/X2/EXCLK 引脚并设置振荡频率(CMC 寄存器)

- $2\text{ MHz} \leq f_x \leq 10\text{ MHz}$

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0	1	0	0/1	0	0/1	0/1	0

- $10\text{ MHz} < f_x \leq 20\text{ MHz}$

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0	1	0	0/1	0	0/1	0/1	1

- 备注**
1. f_x : X1 时钟振荡频率
 2. P123/XT1 和 P124/XT2 引脚的设置，参见 5.6.3 控制副系统时钟的示例。

<2> 控制 X1 时钟振荡(CSC 寄存器)
若 MSTOP 被清零，则 X1 振荡器开始振荡。

<3> 等待 X1 时钟振荡稳定
检测 OSTC 寄存器并等待所需的时间。
在等待期间，其它软件处理过程的执行可以使用内部高速振荡时钟。

- 注意事项**
1. 只有当复位释放后才能通过 8 位存储器操作指令写入 CMC 寄存器。
因此，需要同时设置 OSCSELS 位的值。关于 OSCSELS 位，参见 5.6.3 副系统时钟控制的示例。
 2. 在供电电压达到时钟使用的可操作电压后，设置 X1 时钟(参见第 27 章 电气特性(目标值))。

(2) 使用外部主系统时钟时设置过程示例

<1> 设置 P121/X1 和 P122/X2/EXCLK 引脚(CMC 寄存器)

EXCLK	OSCSEL	0	OSCSLS	0	AMPHS1	AMPHS0	AMPH
1	1	0	0/1	0	0/1	0/1	0/1

备注 P123/XT1 和 P124/XT2 引脚的设置，参见 5.6.3 (1) 当副系统时钟振荡时的设置过程示例。

<2> 控制外部主系统时钟输入(CSC 寄存器)

MSTOP 清零时，使能外部主系统时钟的输入。

注意事项 1.只有当复位释放后才能通过 8 位存储器操作指令写入 CMC 寄存器。

因此，需要同时设置 OSCSLS 位的值。关于 OSCSLS 位，参见 5.6.3 副系统时钟控制的示例。

2.在供电电压达到时钟使用的可操作电压后，设置外部主系统时钟(参见第 27 章 电气特性 (目标值))。

(3) 使用高速系统时钟作为 CPU/外部硬件时钟时的设置过程示例

<1> 设置高速系统时钟振荡^注

(参见 5.6.1 (1)X1 时钟振荡时设置过程示例和(2)使用外部主系统时钟时设置过程示例。)

注 当高速系统时钟已经运行时，不需要设置<1>。

<2> 设置高速系统时钟作为 CPU/外围硬件时钟的源时钟，并设置时钟(CKC 寄存器)的分频比

MCM0	MDIV2	MDIV1	MDIV0	CPU/外围硬件时钟 (f _{clk})的选择
1	0	0	0	f _{MX}
	0	0	1	f _{MX} /2
	0	1	0	f _{MX} /2 ²
	0	1	1	f _{MX} /2 ³
	1	0	0	f _{MX} /2 ⁴
	1	0	1	f _{MX} /2 ⁵ ^注

注 f_{MX} < 4 MHz 时设置禁止。

<3> 如果某些外围硬件模块不使用，可以停止硬件模块的输入时钟供给。

(PER0 寄存器)

RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

xxxEN	输入时钟控制
0	停止输入时钟供给。
1	供给输入时钟

注意事项 确保将位 6 清零。

- 备注**
- RTCEN: 实时计数器输入时钟的控制
 - ADCEN: A/D 转换器输入时钟的控制
 - IICAEN: 控制串行接口 IICA 输入时钟
 - SAU1EN: 控制串行阵列单元 1 输入时钟
 - SAU0EN: 控制串行阵列单元 0 输入时钟
 - TAU1EN: 定时器阵列单元 1 输入时钟的控制
 - TAU0EN: 定时器阵列单元 0 输入时钟的控制

(4) 停止高速系统时钟时设置过程示例

在下列两种情况下，高速系统时钟停止(如果使用外部时钟，禁止时钟输入)。

- 执行 STOP 指令
- 设置 MSTOP 为 1

(a) 执行 STOP 指令

- <1> 设置停止外围硬件
停止那些不能在 STOP 模式下使用的外部硬件(如需了解不能在 STOP 模式下使用的外部硬件，可参见第 17 章 待机功能)。
- <2> 释放 STOP 模式后 X1 时钟振荡稳定时间的设置
如果 X1 时钟在进入 STOP 模式之前开始振荡的话，在执行 STOP 指令之前设置 OSTS 寄存器的值。
- <3> 执行 STOP 指令
执行 STOP 指令时，系统处于 STOP 模式下并且 X1 振荡停止(禁止外部时钟输入)。

(b) 通过将 MSTOP 设置为 1 来停止 X1 振荡(禁止外部时钟输入)**<1> 确定 CPU 时钟状态(CKC 寄存器)**

根据 CLS 和 MCS 确认 CPU 没有使用高速系统时钟。

当 CLS = 0 且 MCS = 1 时, CPU 采用高速系统时钟, 因此将 CPU 时钟转换为副系统时钟或者内部高速振荡时钟。

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟或 20 MHz 内部高速振荡时钟
0	1	高速系统时钟
1	×	副系统时钟

<2> X1 时钟振荡重启后设置X1 时钟振荡的稳定时间^注

优先设置 MSTOP 为“1”, 在 X1 时钟振荡重新开始之后, 向 OSTS 寄存器设置一个值大于已确认的 OSTS 寄存器的计数值。

<3> 停止高速系统时钟(CSC 寄存器)

MSTOP 设置为 1 时, X1 振荡停止(外部时钟的输入禁止)。

注 当高速系统时钟在 X1 振荡模式下, 此设置需要恢复 X1 时钟振荡。
外部时钟输入模式中不需要此设置。

注意事项 设置 MSTOP 为 1 时, 一定要确保 MCS = 0 或 CLS = 1。另外, 停止工作在高速系统时钟的外围硬件。

5.6.2 控制内部高速振荡时钟的示例

下面描述的示例是以下情况时的设置过程示例。

- (1) 当重启内部高速振荡时钟的振荡时
- (2) 当使用内部高速振荡时钟作为 CPU/外围硬件时钟时
- (3) 当停止内部高速振荡时钟时

(1) 内部高速振荡时钟重启振荡时设置过程示例^注**<1> 内部高速振荡时钟重新振荡的设置(CSC 寄存器)**

当 HIOSTOP 清为 0 时, 内部高速振荡时钟重新振荡。

注 复位释放后, 内部高速振荡器自动开始振荡, 并且内部高速振荡时钟作为 CPU/外围硬件时钟。

(2) 当内部高速振荡时钟作为 CPU/外围硬件时钟时设置过程示例**<1> 内部高速振荡时钟的重启振荡^注**

(参见 5.6.2 (1)当重启内部高速振荡时钟时设置过程示例)。

注 当内部高速振荡时钟运行时, 不需要设置<1>。

<2> 设置内部高速振荡时钟作为 CPU/外围硬件时钟的源时钟，并设置时钟(CKC 寄存器)的分频比

MCM0	MDIV2	MDIV1	MDIV0	CPU/外围硬件时钟 (f _{CLK}) 的选择
0	0	0	0	f _{IH}
	0	0	1	f _{IH} /2
	0	1	0	f _{IH} /2 ²
	0	1	1	f _{IH} /2 ³
	1	0	0	f _{IH} /2 ⁴
	1	0	1	f _{IH} /2 ⁵

注意事项 如果在重启内部高速振荡时钟后，将 CPU/外围硬件时钟从高速系统时钟转换到内部高速振荡时钟，需要等待至少 10 μs。

如果在内部高速振荡时钟重启后立即转换，则内部高速振荡的精度不能保证到 10 μs。

(3) 停止内部高速振荡时钟时设置过程示例

可以用如下两种方式停止内部高速振荡时钟。

- 执行 STOP 指令
- 设置 HIOSTOP 为 1

(a) 执行 STOP 指令

<1> 外围硬件的设置

停止那些不能在 STOP 模式下使用的外部硬件(如需了解不能在 STOP 模式下使用的外部硬件，可参见第 17 章 待机功能)。

<2> 释放 STOP 模式后 X1 时钟振荡稳定时间的设置

如果 X1 时钟在进入 STOP 模式之前开始振荡的话，在执行 STOP 指令之前设置 OSTS 寄存器的值。

<3> 执行 STOP 指令

执行 STOP 指令时，系统处于 STOP 模式下并且停止内部高速振荡时钟。

(b) 通过设置 HIOSTOP 为 1 停止内部高速振荡时钟

<1> 确定 CPU 时钟状态(CKC 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用内部高速振荡时钟。

CLS = 0 和 MCS = 0 时，CPU 采用内部高速振荡时钟，因此将 CPU 时钟转换为高速系统时钟或副系统时钟。

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟或 20 MHz 内部高速振荡时钟
0	1	高速系统时钟
1	×	副系统时钟

<2> 停止内部高速振荡时钟(CSC 寄存器)

当 HIOSTOP 设置为 1 时，内部高速振荡时钟停止。

注意事项 当 HIOSTOP = 1 时，必须确认 MCS = 1 或 CLS = 1。此外，必须停止正在使用内部高速振荡时钟的外部硬件。

5.6.3 控制副系统时钟示例

通过连接晶体振荡器到 XT1 和 XT2 引脚，副系统时钟可振荡。

当不使用副系统时钟时，XT1/P123 和 XT2/P124 引脚可用作输入端口引脚。

注意事项 XT1/P123 和 XT2/P124 引脚在复位释放后为输入端口模式。

下面描述的示例是以下情况时的设置过程示例。

- (1) 副系统时钟振荡时
- (2) 使用副系统时钟作为 CPU 时钟时
- (3) 停止副系统时钟时

注意事项 当副系统时钟作为 CPU 时钟时，副系统时钟也提供给外围硬件(除实时计数器、时钟输出/蜂鸣器输出和看门狗定时器外)。此时，不能保证 A/D 转换器和 IICA 的操作。外围硬件的操作特性，参见描述各种外围硬件的章节及第 27 章 电气特性(目标值)。

(1) 副系统时钟振荡的设置示例

<1> 设置 P123/XT1 和 P124/XT2 引脚(CMC 寄存器)

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0/1	0/1	0	1	0	0/1	0/1	0/1

备注 P121/X1 和 P122/X2 引脚的设置，参见 5.6.1 高速系统时钟控制示例。

<2> 副系统时钟(CSC 寄存器)振荡控制

如果 XTSTOP 清为 0，XT1 振荡器开始振荡。

<3> 等待副系统时钟振荡稳定

通过软件等待副系统时钟的振荡稳定时间，使用定时器功能。

注意事项 只有当复位释放后才能通过 8 位存储器操作指令写入 CMC 寄存器。

因此，需要同时设置 EXCLK 和 OSCSEL 位的值。关于 EXCLK 和 OSCSEL 位，参见 5.6.1 (1) X1 时钟振荡时设置过程示例，或 5.6.1 (2) 当使用外部主系统时钟时设置过程示例。

(2) 使用副系统时钟作为 CPU 时钟时的设置过程示例

<1> 设置副系统时钟振荡*

(参见 5.6.3 (1) 副系统时钟振荡时设置过程示例。)

注 当副系统时钟已经运行时，不需要设置<1>。

<2> 设置副系统时钟作为 CPU 时钟(CKC 寄存器)的源时钟

CSS	CPU/外围硬件时钟(f_{CLK})的选择
1	$f_{SUB}/2$

注意事项 当副系统时钟作为 CPU 时钟时，副系统时钟也提供给外围硬件(除实时计数器、时钟输出/蜂鸣器输出和看门狗定时器外)。此时，不能保证 A/D 转换器 和 IICA 的操作。外围硬件的操作特性，参见描述各种外围硬件的章节及第 27 章 电气特性(目标值)。

(3) 停止副系统时钟时的设置过程示例

<1> 确定 CPU 时钟状态(CKC 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用副系统时钟。

当 CLS = 1 时，CPU 采用副系统时钟，因此将 CPU 时钟转换为内部高速振荡时钟或高速系统时钟。

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟或 20 MHz 内部高速振荡时钟
0	1	高速系统时钟
1	×	副系统时钟

<2> 停止副系统时钟(CSC 寄存器)

XTSTOP 设置为 1 时，停止 XT1 振荡。

注意事项 1. 当设置 XTSTOP 为 1 时确保 CLS = 0。另外，如果副系统时钟工作，停止外围硬件。
2. 副系统时钟振荡不能通过使用 STOP 指令来停止。

5.6.4 内部低速振荡时钟的控制示例

内部低速振荡时钟不能用作 CPU 时钟。只能用作看门狗定时器时钟。

复位释放后内部低速振荡器自动开始振荡，并且如果通过选项字节使能看门狗定时器操作，则驱动看门狗定时器(30 kHz(典型值))。

除非看门狗定时器停止，否则内部低速振荡器继续振荡。当看门狗定时器操作时，即使程序循环的情况下，内部低速振荡时钟也不停止。

(1) 停止内部低速振荡时钟时设置过程示例

内部低速振荡时钟在下面两种情况下可被停止。

- 在 HALT/STOP 模式下，通过选项字节 (000C0H 的第 0 位(WDSTBYON) = 0)停止看门狗定时器，并且执行 HALT 或 STOP 指令。
- 通过选项字节 (000C0H 的第 4 位(WDTON) = 0)停止看门狗定时器。

(2) 内部低速振荡时钟重新振荡时设置过程示例

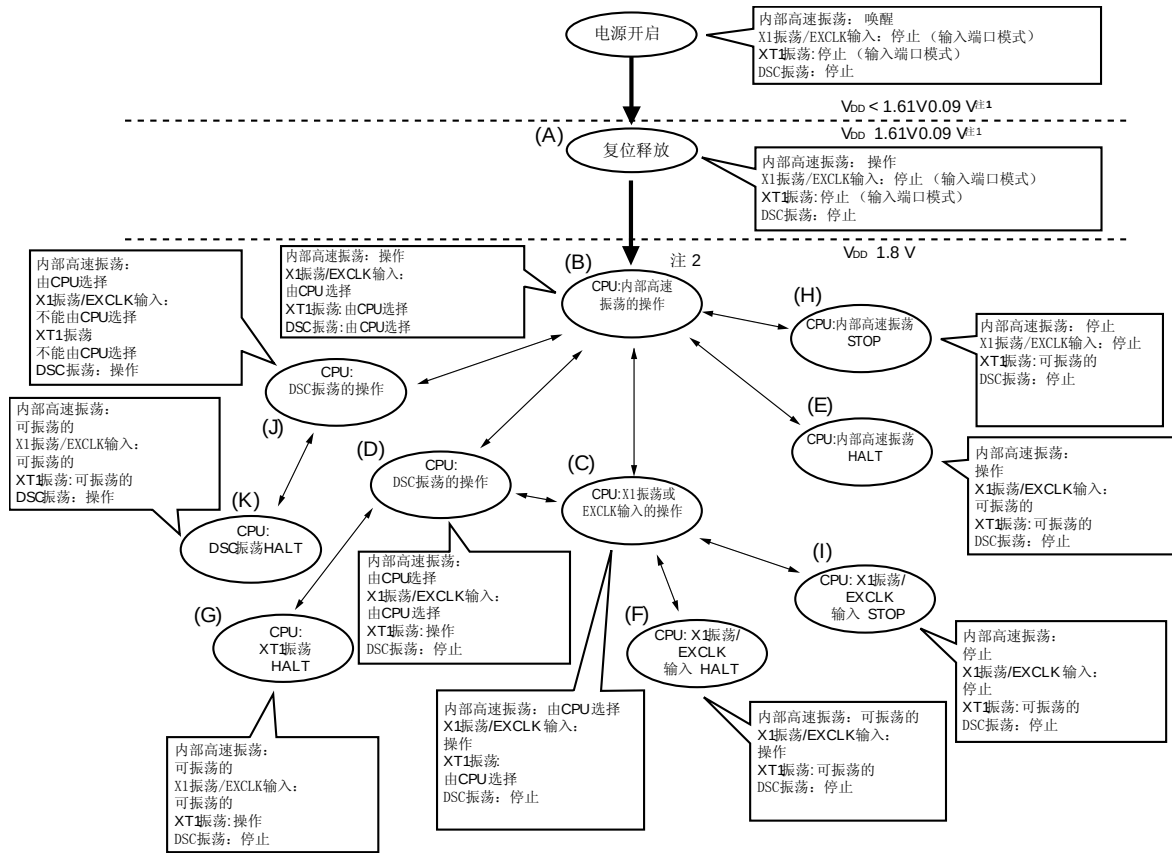
内部低速振荡时钟可重启如下。

- 释放 HALT 或 STOP 模式
(只有当通过选项字节(000C0H 的第 0 位(WDSTBYON) = 0)，在 HALT/STOP 模式下，停止看门狗定时器时和当执行 HALT 或 STOP 指令使看门狗定时器停止时)。

5.6.5 CPU 时钟状态转换图

图 5-17 显示了该产品 CPU 时钟状态转换图。

图 5-17. CPU 时钟状态转换图



- 注
1. 初值可能会改变。
 2. 在复位释放后，开始运行在 4 MHz (8 MHz/2) 上，因为通过设定系统时钟控制寄存器 (CKC) 至 09H 来选择 $f_{CLK} = f_{IH}/2$ 。
 - 当通过选项字节选择了 1MHz 时：500kHz(1 MHz/2)
 - 当通过选项字节选择了 8MHz 或 20MHz 时：4MHz(8 MHz/2)

- 备注
1. 如果低电压检测电路(LVI)通过选项字节默认设置为开启，直到电源电压(V_{DD})超过 2.07 V±0.2 V²复位信号才会被释放。
在复位处理后状态变到上图的(B)。
 2. DSC: 20 MHz 内部高速振荡时钟

表 5-4 显示了 CPU 时钟的转换过程与 SFR 寄存器设置示例。

表 5-4. CPU 时钟转换与 SFR 寄存器设置示例(1/6)

(1) 复位释放后(A)，CPU 使用内部高速振荡时钟(B)

状态转换	SFR 寄存器设置
(A) → (B)	SFR 寄存器不必设置(复位释放后的默认状态)。

(2) 复位释放后(A)，CPU 使用高速系统时钟(C)

(复位释放后，CPU 立即使用内部高速振荡时钟(B)。)

(SFR 寄存器设置序列) →							
状态转换	CMC 寄存器 ^{注1}			CSC 寄存器	OSMC 寄存器	OSTC 寄存器	CKC 寄存器
	EXCLK	OSCSEL	AMPH	MSTOP	FSEL		MCM0
(A) → (B) → (C) (X1 时钟: $2\text{ MHz} \leq f_x \leq 10\text{ MHz}$)	0	1	0	0	0	必须检查	1
(A) → (B) → (C) (X1 时钟: $10\text{ MHz} < f_x \leq 20\text{ MHz}$)	0	1	1	0	1 ^{注2}	必须检查	1
(A) → (B) → (C) (外部主时钟)	1	1	0/1	0	0/1	不能检查	1

注 1. 只有当复位释放后，才能通过 8 位存储器操作指令写入 CMC 寄存器。

2. 当 $f_{CLK} > 10\text{ MHz}$ ， $FSEL = 1$

如果选择分频时钟并且 $f_{CLK} \leq 10\text{ MHz}$ ，即使 $f_x > 10\text{ MHz}$ 也可使用 $FSEL = 0$ 。

注意事项 当供电电压达到时钟的可操作电压后，可以设置时钟(参见 第 27 章 电气特性(目标值))。

(3) 复位释放后(A)，CPU 使用副系统时钟(D)

(复位释放后，CPU 立即使用内部高速振荡时钟(B)。)

(SFR 寄存器设置序列) →						
状态转换	CMC 寄存器 ^注			CSC 寄存器	等待振荡稳定	CKC 寄存器
	OSCELS	AMPHS1	AMPHS0	XTSTOP		CSS
(A) → (B) → (C)	1	0/1	0/1	0	必须检查	1

注 只有当复位释放后，才能通过 8 位存储器操作指令写入 CMC 寄存器。

备注 表 5-4 中的(A)至(K)对应图 5-17 中的(A)至(K)。

表 5-4. CPU 时钟转换与 SFR 寄存器设置示例(2/6)

(4) 复位释放后(A)CPU 在使用 20 MHz 内部高速振荡时钟的操作(J)

(复位释放后, CPU 立即使用内部高速振荡时钟(B)。)

(SFR 寄存器设置序列)

SFR 寄存器的设置标志 状态转换	DSCCTL 寄存器 ^注	等待振荡稳定	DSCCTL 寄存器
	DSCON		SELDSC
(A) → (B) → (J)	1	必要 (100 μs)	1

注 检查 $V_{DD} \geq 2.7\text{ V}$ 并设置 $DSCON = 1$ 。

(5) CPU 时钟从内部高速振荡时钟(B)转换到高速系统时钟(C)

(SFR 寄存器设置序列)

SFR 寄存器的设置标志 状态转换	CMC 寄存器 ^{注1}			OSTS 寄存器	CSC /寄存 器	OSMC 寄存器	OSTC 寄存器	CKC 寄存 器
	EXCLK	OSCSEL	AMPH		MSTOP	FSEL		MCM0
(B) → (C) (X1 时钟: $2\text{ MHz} \leq f_X \leq 10\text{ MHz}$)	0	1	0	注 2	0	0	必须检查	1
(B) → (C) (X1 时钟: $10\text{ MHz} < f_X \leq 20\text{ MHz}$)	0	1	1	注 2	0	1 ^{注3}	必须检查	1
(B) → (C) (外部主时钟)	1	1	0/1	注 2	0	0/1	不能检查	1

如果这些寄存器已经设置则不必要 如果 CPU 使用高速系统时钟则不必要

- 注
1. CMC 寄存器只能在复位释放后改变。如果已经设置了该项, 则无需再设置。
 2. 设置振荡稳定时间如下。
 - 预期的 OSTC 振荡稳定时间 $\leq$ OSTS 设置的振荡稳定时间
 3. 当 $f_{CLK} > 10\text{ MHz}$, $FSEL = 1$
如果选择分频时钟并且 $f_{CLK} \leq 10\text{ MHz}$, 即使 $f_X > 10\text{ MHz}$ 也可使用 $FSEL = 0$ 。

注意事项 当供电电压达到时钟的可操作电压后, 可以设置时钟(参见 第 27 章 电气特性(目标值))。

备注 表 5-4 中的(A)至(K)对应图 5-17 中的(A)至(K)。

表 5-4. CPU 时钟转换与 SFR 寄存器设置示例(3/6)

(6) CPU 时钟从内部高速振荡时钟(B)转换到副系统时钟(D)

(SFR 寄存器设置序列) →				
状态转换	SFR 寄存器的设置标志	CMC 寄存器 ^注	CSC 寄存器	CKC 寄存器
		OSCSLS	XTSTOP	CSS
(B) → (D)		1	0	必要

如果 CPU 工作在副系统时钟则不需要

注 只有当复位释放后，才能通过 8 位存储器操作指令写入 CMC 寄存器。

(7) CPU 时钟从内部高速振荡时钟 (B)转换到 20 MHz 内部高速振荡时钟(J)

(SFR 寄存器设置序列) →			
状态转换	SFR 寄存器的设置标志	DSCCTL 寄存器 ^注	等待振荡稳定
		DSCON	SELDSC
(B) → (J)		1	必要 (100 μs)

如果 CPU 使用 20MHz 的内部高速振荡时钟则不必要

注 检查 $V_{DD} \geq 2.7\text{ V}$ 并设置 DSCON = 1。

(8) CPU 时钟从高速系统时钟(C)转换到内部高速振荡时钟(B)

(SFR 寄存器设置序列) →			
状态转换	SFR 寄存器的设置标志	CSC 寄存器	振荡精确稳定时间
		HIOSTOP	MCM0
(C) → (B)		0	10 μs

如果 CPU 使用内部高速振荡时钟则不必要

备注 表 5-4 中的(A)至(K)对应图 5-17 中的(A)至(K)。

表 5-4. CPU 时钟转换与 SFR 寄存器设置示例(4/6)

(9) CPU 时钟从高速系统时钟(C)转换到副系统时钟(D)

(SFR 寄存器设置序列) 

SFR 寄存器的设置标志 状态转换	CSC 寄存器	等待振荡稳定	CKC 寄存器
	XTSTOP		CSS
(C) → (D)	0	必要	1

如果 CPU 工作在副系统时钟则不必要

(10) CPU 时钟从副系统时钟(D)转换到内部高速振荡时钟(B)

(SFR 寄存器设置序列) 

SFR 寄存器的设置标志 状态转换	CSC 寄存器	CKC 寄存器	
	HIOSTOP	MCM0	CSS
(D) → (B)	0	0	0

如果 CPU 使用内部高速振荡时钟则不必要

如果这些寄存器已经设置则不必要

备注 表 5-4 中的(A)至(K)对应图 5-17 中的(A)至(K)。

表 5-4. CPU 时钟转换与 SFR 寄存器设置示例(5/6)

(11) CPU 时钟从副系统时钟(D)转换到高速系统时钟(C)

(SFR 寄存器设置序列) →

状态转换 \ SFR 寄存器的设置标志	OSTS 寄存器	CSC 寄存器	OSMC 寄存器	OSTC 寄存器	CKC 寄存器	
		MSTOP	FSEL		MCM0	CSS
(D) → (C) (X1 时钟: $2 \text{ MHz} \leq f_x \leq 10 \text{ MHz}$)	注 1	0	0	必须检查	1	0
(D) → (C) (X1 时钟: $10 \text{ MHz} < f_x \leq 20 \text{ MHz}$)	注 1	0	1 ^{※2}	必须检查	1	0
(D) → (C) (外部主时钟)	注 1	0	0/1	不能检查	1	0

如果 CPU 使用高速系统时钟则不必要

如果这些寄存器已经设置则不必要

注 1. 设置振荡稳定时间如下。

- 预期的 OSTC 振荡稳定时间 $\leq$ OSTS 设置的振荡稳定时间

2. 当 $f_{\text{CLK}} > 10 \text{ MHz}$, $\text{FSEL} = 1$

如果选择分频时钟并且 $f_{\text{CLK}} \leq 10 \text{ MHz}$, 即使 $f_x > 10 \text{ MHz}$ 也可使用 $\text{FSEL} = 0$ 。

注意事项 当供电电压达到时钟的可操作电压后, 可以设置时钟(参见 第 27 章 电气特性(目标值))。

(12) CPU 时钟从 20 MHz 内部高速振荡时钟(J)转换到内部高速振荡时钟(B)

(SFR 寄存器设置序列) →

状态转换 \ SFR 寄存器的设置标志	DSCCTL 寄存器	
	SELDSC	DSCON
(J) → (B)	0	0

备注 表 5-4 中的(A)至(K)对应图 5-17 中的(A)至(K)。

表 5-4. CPU 时钟转换与 SFR 寄存器设置示例(6/6)

- (13) • CPU 使用内部高速振荡时钟(B)时，HALT 模式(E)的设置。
- CPU 使用高速系统时钟(C)时，HALT 模式(F)的设置。
 - CPU 使用副系统时钟(D)时，HALT 模式(G)的设置。
 - 当 CPU 运行在 20 MHz 内部高速振荡时钟时(J)，HALT 模式(K)的设置。

状态转换	设置
(B) → (E) (C) → (F) (D) → (G) (J) → (K)	执行 HALT 指令

- (14) • CPU 使用内部高速振荡时钟(B)时，STOP 模式(H) 的设置。
- CPU 使用高速系统时钟(C)时，STOP 模式(I) 的设置。

(设置顺序) →

状态转换		设置		
(B) → (H)		停止那些不能在 STOP 模式下使用的外围功能	—	执行 STOP 指令
(C) → (I)	X1 停止		设置 OSTS 寄存器	
	X1 振荡		—	

备注 表 5-4 中的(A)至(K)对应图 5-17 中的(A)至(K)。

5.6.6 CPU 时钟改变之前的状况与改变之后的处理

CPU 时钟改变之前的状况与改变之后的处理显示如下。

表 5-5. 改变 CPU 时钟 (1/2)

CPU 时钟		改变前的状况	改变后的处理
改变前	改变后		
内部高速振荡时钟	X1 时钟	X1 振荡稳定 • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • 经历了振荡稳定时间	可以停止内部高速振荡器(HIOSTOP = 1)。
	外部主系统时钟	允许来自 EXCLK 引脚的外部时钟输入 • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	副系统时钟	XT1 振荡稳定 • OSCSELS = 1, XTSTOP = 0 • 经历了振荡稳定时间	
	20 MHz 内部高速振荡时钟	使用选项字节来设置 20MHz 的 DSC 振荡器的稳定性。 • $V_{DD} \geq 2.7\text{ V}$ • 经历了振荡稳定之后 设定 DSCON = 1 之后, 时间(100 μ s) • SELDSC = 1	—
X1 时钟	内部高速振荡时钟	内部高速振荡器振荡 HIOSTOP = 0	X1 振荡停止 (MSTOP = 1)。
	外部主系统时钟	转换不允许 (要转换时钟, 执行复位后再设置它。)	—
	副系统时钟	XT1 振荡稳定 • OSCSELS = 1, XTSTOP = 0 • 经历了振荡稳定时间	X1 振荡停止 (MSTOP = 1)。
	20 MHz 内部高速振荡时钟	不能进行转换, 除非时钟转换为内部高速振荡时钟。	—
外部主系统时钟	内部高速振荡时钟	内部高速振荡器振荡 HIOSTOP = 0	外部主系统时钟输入禁止(MSTOP = 1)。
	X1 时钟	转换不允许 (要改变时钟, 执行复位后再设置它。)	—
	副系统时钟	XT1 振荡稳定 • OSCSELS = 1, XTSTOP = 0 • 经历了振荡稳定时间	外部主系统时钟输入禁止(MSTOP = 1)。
	20 MHz 内部高速振荡时钟	不能进行转换 除非时钟转换为内部高速振荡时钟。	—

表 5-5. 改变 CPU 时钟 (2/2)

CPU 时钟		改变前的状况	改变后的处理
改变前	改变后		
副系统时钟	内部高速振荡时钟	内部高速振荡器振荡且选择内部高速振荡时钟作为主系统时钟 • HIOSTOP = 0, MCS = 0	XT1 振荡停止 (XTSTOP = 1)
	X1 时钟	X1 振荡稳定且选择高速系统时钟作为主系统时钟 • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • 经历了振荡稳定时间 • MCS = 1	
	外部主系统时钟	允许来自 EXCLK 引脚的外部时钟输入 并选择高速系统时钟作为主系统时钟 • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	20 MHz 内部高速振荡时钟	不能进行转换, 除非时钟转换为内部高速振荡时钟。	—
20 MHz 内部高速振荡时钟	内部高速振荡时钟	• SELDSC = 0 (当改变时钟时设置。)	可停止 20 MHz 内部高速振荡时钟 (DSCON=0)
	X1 时钟	不能进行转换 除非时钟转换为内部高速振荡时钟。	—
	外部主系统时钟	不能进行转换 除非时钟转换为内部高速振荡时钟。	—
	副系统时钟	不能进行转换 除非时钟转换为内部高速振荡时钟。	—

5.6.7 CPU 时钟和主系统时钟转换所需的时间

通过设置系统时钟控制寄存器(CKC)的第 0 至 2 位、第 4 和第 6 位(MDIV0 至 MDIV2、MCM0、CSS)，可以转换 CPU 时钟(在主系统时钟和副系统时钟之间)、转换主系统时钟(在内部高速振荡时钟和高速系统时钟之间)，同时可以改变主系统时钟的分频比。

修改 CKC 后，实际的转换操作不会立即执行；操作在予转换时钟上将继续几个时钟(见表 5-6 至表 5-9)。

可以通过 CKC 寄存器的第 7 位(CLS)来确定 CPU 使用主系统时钟还是副系统时钟。可以通过 CKC 寄存器的第 5 位(MCS)来确定主系统时钟是高速系统时钟还是内部高速振荡时钟。

当转换 CPU 时钟时，外围硬件时钟也被转换。

表 5-6. 主系统时钟转换需要的最长时间

时钟 A	转换方法	时钟 B	备注
f_{MAIN}	$\longleftrightarrow$ (改变分频比)	f_{MAIN}	请参阅表 5-7
f_{IH}	$\longleftrightarrow$	f_{MX}	请参阅表 5-8
f_{MAIN}	$\longleftrightarrow$	f_{SUB}	请参阅表 5-9

表 5-7. $f_{MAIN} \leftrightarrow f_{MAIN}$ 所需的最大时钟数(改变分频比)

转换前的设置值	转换后的设置值	
	时钟 A	时钟 B
时钟 A		$1 + f_A/f_B$ 时钟
时钟 B	$1 + f_B/f_A$ 时钟	

表 5-8. $f_{IH} \leftrightarrow f_{MX}$ 所需的最大时钟数

转换前的设置值		转换后的设置值	
MCM0		MCM0	
		0 ($f_{MAIN} = f_{IH}$)	1 ($f_{MAIN} = f_{MX}$)
0 ($f_{MAIN} = f_{IH}$)	$f_{MX} > f_{IH}$		$1 + f_{IH}/f_{MX}$ 时钟
	$f_{MX} < f_{IH}$		$2f_{IH}/f_{MX}$ 时钟
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} > f_{IH}$	$2f_{MX}/f_{IH}$ 时钟	
	$f_{MX} < f_{IH}$	$1 + f_{MX}/f_{IH}$ 时钟	

(备注在下页中列出。)

表 5-9. $f_{\text{MAIN}} \leftrightarrow f_{\text{SUB}}$ 所需的最大时钟数

转换前的设置值		转换后的设置值	
CSS		CSS	
		0 ($f_{\text{CLK}} = f_{\text{MAIN}}$)	1 ($f_{\text{CLK}} = f_{\text{SUB}}$)
0 ($f_{\text{CLK}} = f_{\text{MAIN}}$)	$f_{\text{MAIN}} < f_{\text{SUB}}$		$2 + f_{\text{MAIN}}/f_{\text{SUB}}$ 时钟
	$f_{\text{MAIN}} > f_{\text{SUB}}$		$1 + 2f_{\text{MAIN}}/f_{\text{SUB}}$ 时钟
1 ($f_{\text{CLK}} = f_{\text{SUB}}$)	$f_{\text{MAIN}} < f_{\text{SUB}}$	$1 + 2f_{\text{SUB}}/f_{\text{MAIN}}$ 时钟	
	$f_{\text{MAIN}} > f_{\text{SUB}}$	$2 + f_{\text{SUB}}/f_{\text{MAIN}}$ 时钟	

- 备注**
1. 表 5-7 至表 5-9 中列出的时钟数是切换前的 CPU 时钟数。
 2. 通过舍去小数部分，计算表 5-7 至表 5-9 的时钟数。

实例 将主系统时钟从内部高速振荡时钟切换到高速系统时钟(@振荡频率 $f_{\text{IH}} = 8 \text{ MHz}$, $f_{\text{XH}} = 10 \text{ MHz}$)

$$1 + f_{\text{IH}}/f_{\text{MX}} = 1 + 8/10 = 1 + 0.8 = 1.8 \rightarrow 2 \text{ 时钟}$$

5.6.8 时钟振荡停止前的状况

下表列出了停止时钟振荡的寄存器标志位设置(禁止外部时钟输入)和时钟振荡停止前的状况。

表 5-10. 时钟振荡停止前的状况及标志位设置

时钟	时钟振荡停止前的状况 (禁止外部时钟输入)	SFR 寄存器的标志位 设置
内部高速振荡时钟	MCS = 1 或 CLS = 1 (CPU 使用内部高速振荡时钟以外的时钟运行)	HIOSTOP = 1
X1 时钟	MCS = 0 或 CLS = 1 (CPU 使用高速系统时钟以外的时钟运行)	MSTOP = 1
外部主系统时钟		
副系统时钟	CLS = 0 (CPU 使用副系统时钟以外的时钟运行)	XTSTOP = 1
20 MHz 内部高速振荡时钟	SELDSC = 0 (主系统时钟工作在 20 MHz 内部高速振荡时钟以外的时钟下)	DSCON = 0

第 6 章 定时器阵列单元

定时器阵列单元具有两个单元。定时器阵列单元 0 具有 8 个 16 位定时器，定时器阵列单元 1 具有 4 个 16 位定时器。每个 16 位定时器叫一个通道，可以作为一个独立定时器。此外，两个或更多的“通道”可用来创建高精度定时器。

独立操作功能	组合操作功能
• 间隔定时器 • 方波输出 • 外部事件计数器 • 分频功能(仅限通道 0 的单元 0) • 输入脉冲间隔测量 • 输入信号的高/低电平宽度测量	• PWM 输出 • 单脉冲输出 • 多路 PWM 输出

通道 7 与串行阵列单元 1 的 UART3 可以组合实现 LIN 总线接收处理。

6.1 定时器阵列单元的功能

定时器阵列单元有如下功能。

6.1.1 独立操作时每个通道的功能

独立操作功能是指和其它通道的操作模式无关并可任意通道使用的功能。(详细情况，参考 6.6.1 独立操作功能和组合操作功能的概述)。

(1) 间隔定时器

每个单元的每个定时器可用作产生固定间隔中断 (INTTMmn) 的参考定时器。

(2) 方波输出

每次产生 INTTMmn 时执行反向输出操作，并且从定时器输出引脚 (TOMn) 输出占空比为 50% 的方波。

(3) 外部事件计数器

当输入到定时器输入引脚 (TIMn) 的信号的有效沿的数量达到一个指定值时，一个单元的每个定时器都可用作事件计数器来产生中断。

(4) 分频功能(仅单元 0 的通道 0)

从定时器输入引脚(TI00)输入的时钟分频，并且从输出引脚 (TO00)输出。

(5) 输入脉冲间隔测量

在脉冲信号有效沿输入到定时器输入引脚 (TIMn) 时，开始计数。定时器的计数值在下个脉冲有效沿捕捉。 这样，可以测量输入脉冲的间隔。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7)
 mn = 00 至 07, 10 至 13

(6) 输入信号高/低电平宽度测量

在信号的单沿输入到定时器输入引脚 (TImn) 时, 计数开始, 并且在其他沿捕捉计数值。这样, 可以测量输入信号的高电平或低电平宽度。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7)
 mn = 00 至 07, 10 至 13

6.1.2 和其它通道组合操作时每个通道的功能

组合操作功能是将主通道(控制周期的主要参考定时器)和从通道(随主通道操作的定时器)组合在一起进行的操作(详细信息, 请参考 **6.6.1 独立操作功能和组合操作功能的概述**)。

(1) PWM (脉冲宽度调制)输出

两个通道用作设置产生指定周期和指定占空比的脉冲。

(2) 单脉冲输出

两个通道用作设置产生指定延迟时间和指定脉冲宽度的单脉冲。

(3) 多路 PWM(脉冲宽度调制)输出

通过扩展 PWM 功能, 使用一个主通道和两个以上从通道, 可以产生七种类型指定周期和额指定占空系数的 PWM 信号。

6.1.3 LIN 总线支持功能(仅限通道 7)**(1) 唤醒信号的检测**

定时器在输入到 UART3 的串行数据输入引脚 (RxD3) 信号的下降沿开始计数, 在上升沿捕捉定时器计数值。这样, 可以测量低电平宽度。如果低电平宽度大于指定值, 则认为是一个唤醒信号。

(2) 同步中断域检测

在检测到唤醒信号后, 定时器在输入到 UART3 的串行数据输入引脚 (RxD3) 的信号下降沿开始计数, 在上升沿捕捉定时器计数值。这样, 可以测量低电平宽度。如果低电平宽度大于指定值, 则认为是同步中断域。

(3) 同步域的脉冲宽度测量

检测到同步中断区域后, 可以测量到输入至 UART3 的串行数据输入引脚 (RxD3) 信号的低电平宽度和高电平宽度。这样通过测量到的同步域的位间隔, 可计算出波特率。

6.2 定时器阵列单元的配置

定时器阵列单元包括如下硬件。

表 6-1. 定时器阵列单元的配置

项目	配置
定时器/计数器	定时计数寄存器 mn (TCRmn)
寄存器	定时数据寄存器 mn (TDRmn)
定时器输入	TI00 至 TI07, TI10 至 TI13 引脚, RxD03 引脚(对于 LIN 总线)
定时器输出	TO00 至 TO07, TO10 至 TO13 引脚, 输出控制电路
控制寄存器	<单位设置块寄存器> • 外围使能寄存器 0 (PER0) • 定时器时钟选择寄存器 m(TPSm) • 定时器通道使能状态寄存器 m (TEm) • 定时器通道启动寄存器 m (TSm) • 定时器通道停止寄存器 m (TTm) • 定时器输入选择寄存器 m (TISm) • 定时器输出使能寄存器 m (TOEm) • 定时器输出寄存器 m (TOM) • 定时器输出电平寄存器 m (TOLm) • 定时器输出模式寄存器 m (TOMm)
	<每个通道的寄存器> • 定时器模式寄存器 mn (TMRmn) • 定时器状态寄存器 mn (TSRmn) • 输入切换控制寄存器 (ISC) • 噪声滤波使能寄存器 1、2 (NFEN1、NFEN2) • 端口模式寄存器 0, 1, 3, 4, 6, 13, 14 (PM0, PM1, PM3, PM4, PM6, PM13, PM14) • 端口寄存器 0, 1, 3, 4, 6, 13, 14 (P0, P1, P3, P4, P6, P13, P14)

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7)
mn = 00 至 07, 10 至 13

图 6-1 和 6-2 显示了框图。

图 6-1. 定时器阵列单元 0 的框图

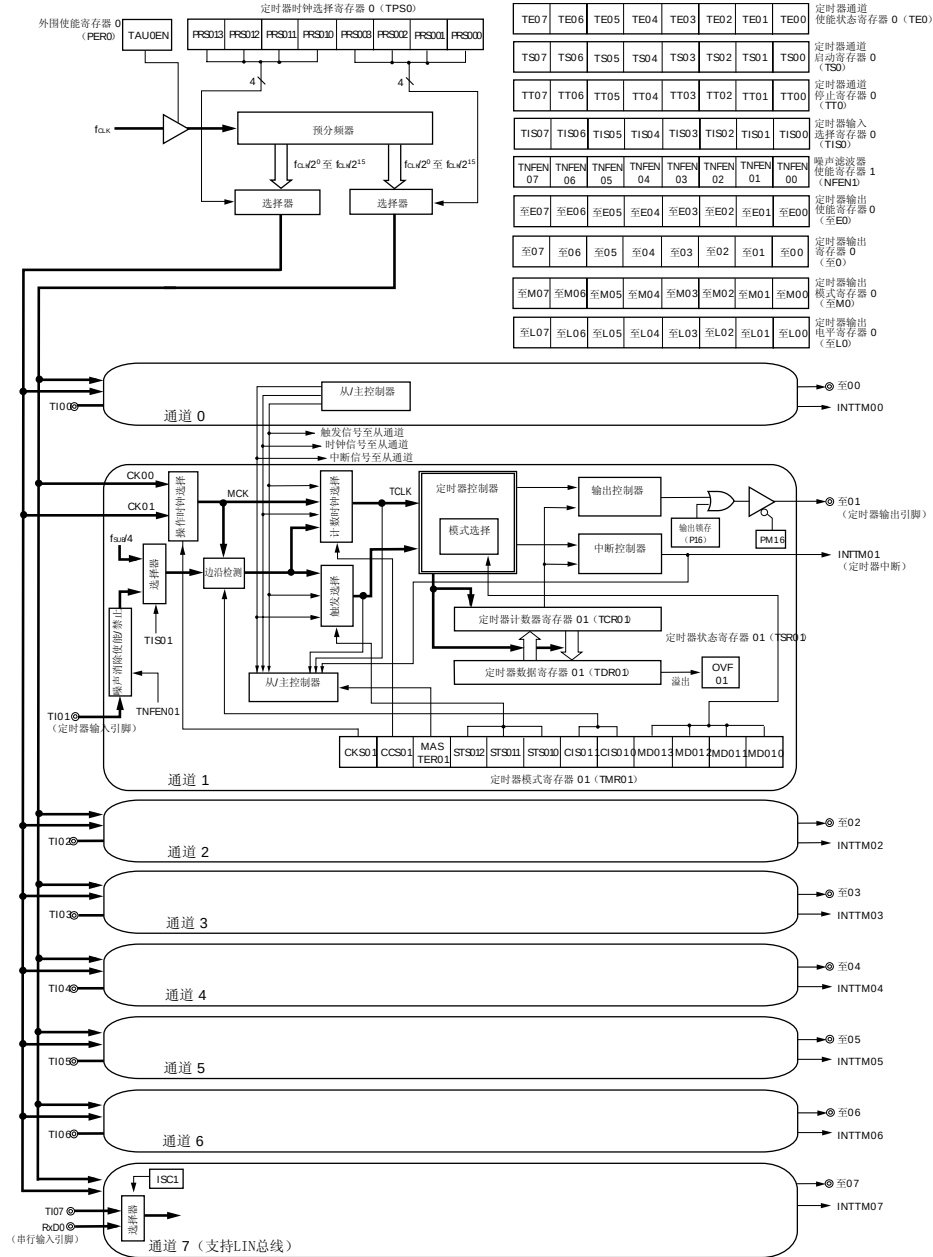
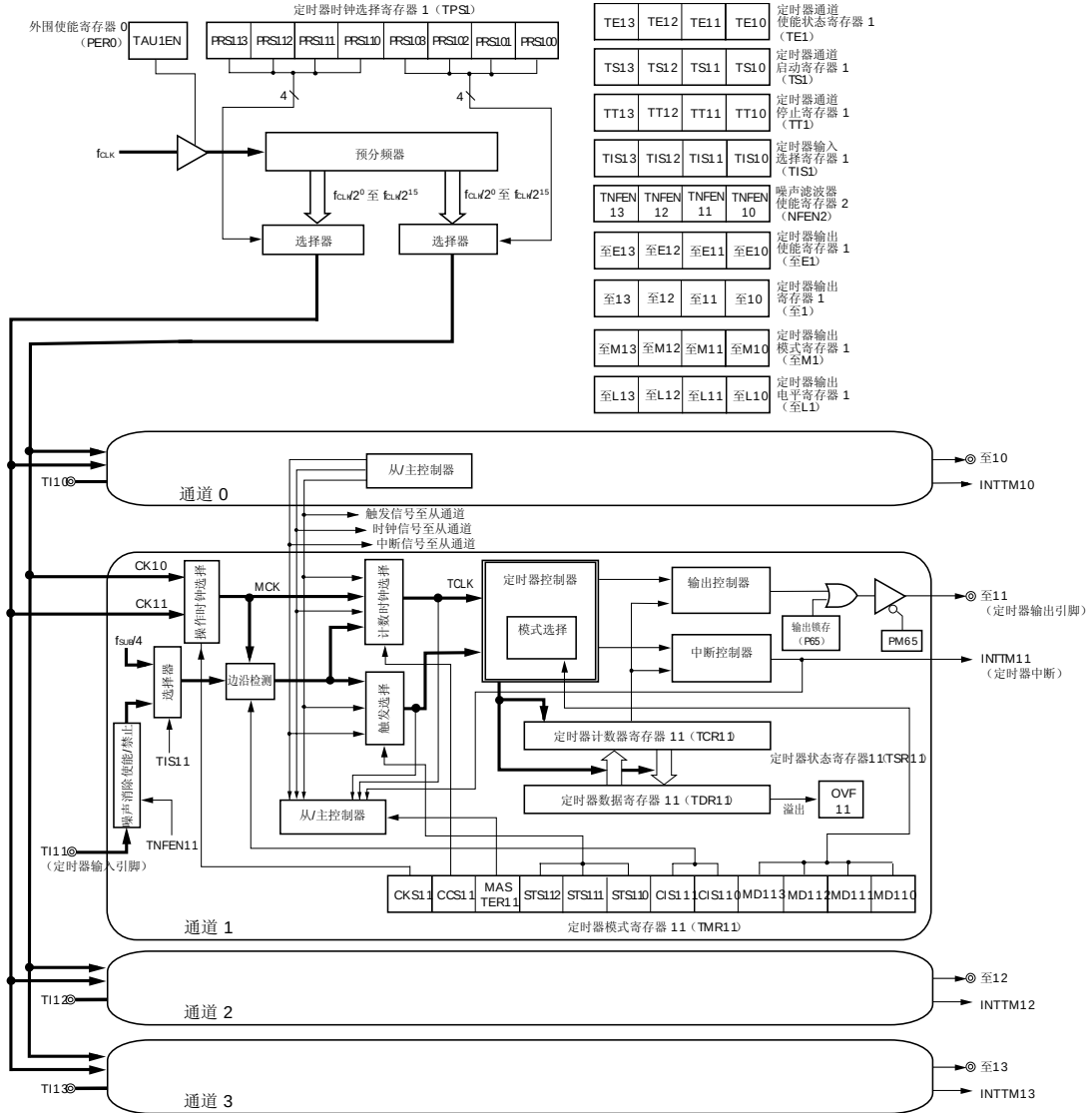


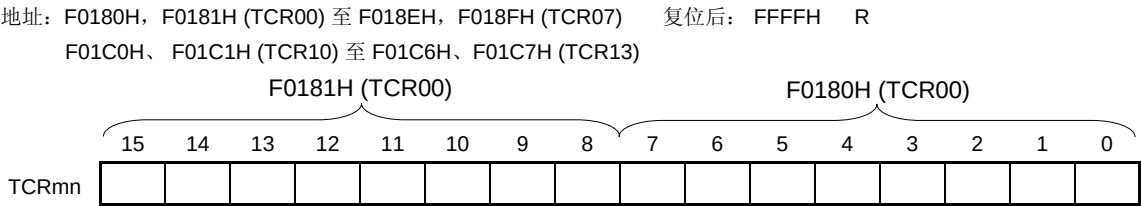
图 6-2. 定时器阵列单元 1 的框图



(1) 定时计数寄存器 mn (TCRmn)

TCRmn 是 16 位只读寄存器，用于时钟计数。
此计数器的值与计数时钟上升沿同步递增或递减。
计数器的方向是递增或递减，由 TMRmn 的 MDmn3 至 MDmn0 位选择的操作模式决定。

图 6-3. 定时器计数寄存器 mn (TCRmn)的格式



可通过读取 TCRmn 获得计数值。
如下情况下计数值设置为 FFFFH。

- 复位信号发生时
- 当外围使能寄存器 0 (PER0)的 TAU0EN 位 (TAU0 的情形) 或者 TAU1EN 位 (TAU1 的情形)清零时
- 在 PWM 输出模式下当从通道计数已经完成时
- 在单脉冲输出模式下主/从通道的计数已经完成时
- 在多路 PWM 输出模式下从通道的计数已经完成时

计数值在如下情况中清除为 0000H。

- 在捕捉模式下启动触发输入时
- 在捕捉模式下捕捉操作已完成

注意事项 即使读取 TCRmn 时，计数值也不会捕捉到 TDRmn。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
 mn = 00 至 07, 10 至 13

TCRmn 寄存器读取值根据操作模式改变和操作状态而有如下差异。

表 6-2. TCRmn 寄存器在不同操作模式下的读取值

操作模式	计数模式	TCRmn 寄存器读取值 ^注			
		复位后操作模式改变	计数操作暂停后操作模式改变 (TTmn = 1)	计数操作暂停后重启 (TTmn = 1)	一次计数后启动触发等待状态期间
间隔定时器模式	递减	FFFFH	未定义	操作终止时的值	–
捕捉模式	递增	0000H	未定义	操作终止时的值	–
事件计数器模式	递减	FFFFH	未定义	操作终止时的值	–
一次计数模式	递减	FFFFH	未定义	操作终止时的值	FFFFH
捕捉&一次计数模式	递增	0000H	未定义	操作终止时的值	TDRmn寄存器的捕捉值 + 1

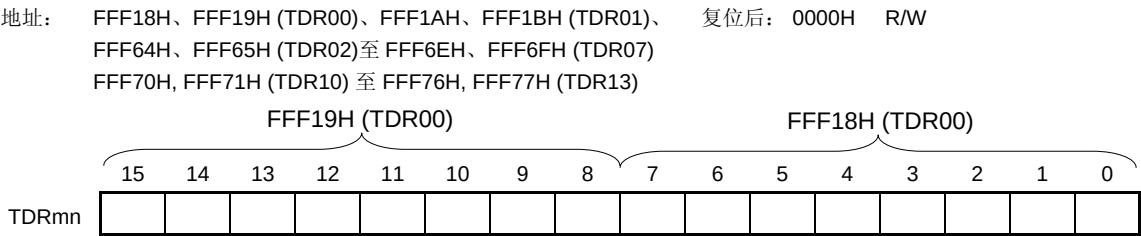
注 显示当 TEmn = 0 且 TSmn 设置为"1"时 TCRmn 寄存器的读取值。该读取值保存在 TCRmn 寄存器直到计数操作开始。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7), mn = 00 至 07, 10 至 13

(2) 定时器数据寄存器 mn (TDRmn)

此寄存器为 16 位寄存器，可选择捕捉功能和比较功能。
通过使用 TMRmn 的 MDmn3 至 MDmn0 位选择操作模式可切换捕捉功能或比较功能。
TDRmn 的值在任何时间都可改变。
此寄存器可以以 16 位单元读或写。
复位后这个寄存器的值为 0000H。

图 6-4. 定时器数据寄存器 mn (TDRmn)的格式



(i) TDRmn 用作比较寄存器时

由 TDRmn 的设置值开始向下计数。当计数值达到 0000H 时，产生一个中断信号(INTTMmn)。TDRmn 的值会一直保持，直到重写。

注意事项 当设置为比较功能时，即使有捕捉触发信号输入，TDRmn 也不会执行捕捉操作。

(ii) TDRmn 用作捕捉寄存器时

当捕捉触发输入时，TCRmn 的计数值捕捉到 TDRmn。
Tlmn 引脚的有效沿可选择作为捕捉触发。此选择由 TMRmn 进行。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

6.3 控制定时器阵列单元的寄存器

定时器阵列单元由如下寄存器控制。

- 外围使能寄存器 0 (PER0)
- 定时器时钟选择寄存器 m(TPSm)
- 定时器模式寄存器 mn (TMRmn)
- 定时器状态寄存器 mn (TSRmn)
- 定时器通道使能状态寄存器 m (TEm)
- 定时器通道启动寄存器 m (TSM)
- 定时器通道停止寄存器 m (TTm)
- 定时器输入选择寄存器 m(TISM)
- 定时器输出使能寄存器 m (TOEm)
- 定时器输出寄存器 m (TOM)
- 定时器输出电平寄存器 m (TOLm)
- 定时器输出模式寄存器 m (TOMm)
- 输入切换控制寄存器 (ISC)
- 噪声滤波器使能寄存器 1、2(NFEN1、NFEN2)
- 端口模式寄存器 0, 1, 3, 4, 6, 13, 14 (PM0, PM1, PM3, PM4, PM6, PM13, PM14)
- 端口寄存器 0, 1, 3, 4, 6, 13, 14 (P0, P1, P3, P4, P6, P13, P14)

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

(1) 外围使能寄存器 0 (PER0)

PER0 用于使能或者禁止使用每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少功耗和噪声。
使用定时器阵列单元 0 时，确保设置此寄存器的第 0 位 (TAU0EN) 为 1。
使用定时器阵列单元 1 时，确保设置此寄存器的位 1 (TAU1EN) 为 1。
可以通过 1 位或者 8 位存储器操作指令来设置 PER0。
复位后这个寄存器的值为 00H。

图 6-5. 外围使能寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

TAUmEN	定时器阵列单元m输入时钟的控制
0	停止输入时钟的供给 - 定时器阵列单元m使用的SFR不能写入。 - 定时器阵列单元m处于复位状态。
1	供给输入时钟 定时器阵列单元m使用的SFR可进行读写。

注意事项

1. 设置定时器阵列单元时，首先确保设置 TAUmEN 为 1。当 TAUmEN = 0 时，忽略写入定时器阵列单元的控制寄存器，并且所有的读数均为默认值(噪声滤波使能寄存器 1、2 (NFEN1、NFEN2)、端口模式寄存器 0, 1, 3, 4, 6, 13, 14 (PM0, PM1, PM3, PM4, PM6, PM13, PM14)和端口寄存器 0, 1, 3, 4, 6, 13, 14 (P0, P1, P3, P4, P6, P13, P14)除外)。

2. 确保将位 6 清零。

备注

m = 0, 1

(2) 定时器时钟选择寄存器 m(TPSm)

TPSm 是一个 16 位寄存器，用于选择两种类型的工作时钟(CKm0, CKm1)，这些时钟通常供给每个通道。
CKm1 由 TPSm 的位 7 至 4 选择，CKm0 由 TPSm 的位 3 至 0 选择。

定时器操作期间只有在如下情况下可以重写 TPSm。

PRSm00 至 PRSm03 位重写，只有在操作停止状态(TEmn = 0)下所有通道设置为 CKSmn = 0 时可能

PRSm10 至 PRSm13 位重写，只有在操作停止状态(TEmn = 0)下所有通道设置为 CKSmn = 1 时可能

TPSm 可由 16 位存储器操作指令设置。

TPSm 的低 8 位可通过 TPSmL 由 8 位存储操作指令设置。

复位后这个寄存器的值为 0000H。

图 6-6. 定时器时钟选择寄存器 m (TPSm)的格式

地址: F01B6H, F01B7H (TPS0) 复位后: 0000H R/W

F01DEH, F01DFH (TPS1)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0		操作时钟选择(CKmk) ^注			
					f _{CLK} = 2 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz
0	0	0	0	f _{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	f _{CLK} /2	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	f _{CLK} /2 ²	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	f _{CLK} /2 ³	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	f _{CLK} /2 ⁴	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	f _{CLK} /2 ⁵	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz
0	1	1	0	f _{CLK} /2 ⁶	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz
0	1	1	1	f _{CLK} /2 ⁷	15.62 kHz	39.1 kHz	78.1 kHz	156.2 kHz
1	0	0	0	f _{CLK} /2 ⁸	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	f _{CLK} /2 ⁹	3.91 kHz	9.76 kHz	19.5 kHz	39.1 kHz
1	0	1	0	f _{CLK} /2 ¹⁰	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz
1	0	1	1	f _{CLK} /2 ¹¹	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz
1	1	0	0	f _{CLK} /2 ¹²	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	f _{CLK} /2 ¹³	244 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	f _{CLK} /2 ¹⁴	122 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	f _{CLK} /2 ¹⁵	61 Hz	153 Hz	305 Hz	610 Hz

注 当改变 f_{CLK} (由系统时钟控制寄存器 (CKC) 的值改变) 时钟时，停止定时器阵列单元 (TTm = 00FFH)。

注意事项 确保将位 15 至 8 清“0”。

备注 1. f_{CLK}: CPU/外围硬件时钟频率

2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7), k = 0, 1, mn = 00 至 07, 10 至 13

(3) 定时器模式寄存器 mn (TMRmn)

TMRmn 设置通道 n 的操作模式。它用于选择操作时钟(MCK)、计数时钟，无论定时器作为主或从操作，开始触发和捕捉出发、定时器输入的有效沿和操作模式(间隔、捕捉、事件计数器、单计数、或捕捉&单计数)。

当寄存器操作时 (当 TEm = 1) 禁止重写 TMRmn。但是第 7 位和第 6 位 (CISmn1、CISmn0) 可以重写，即使当寄存器一些功能正在操作时 (当 TEm = 1) (详细情况，请参考 6.7 定时器阵列单元的独立通道操作和 6.8 定时器阵列单元的多通道操作)。

TMRmn 可由 16 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 6-7. 定时器模式寄存器 mn (TMRmn) 的格式(1/3)

地址: F0190H, F0191H (TMR00) 至 F019EH, F019FH (TMR07) 复位后: 0000H R/W

F01C8H、F01C9H (TMR10) 至 F01CEH、F01CFH (TMR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKS mn	0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CKS mn	通道n的工作时钟 (MCK) 选择
0	由TPSm寄存器设置时钟CKm0操作
1	由TPSm寄存器设置时钟CKm1操作
通过边沿检测器使用工作时钟MCK。根据CCSmn位的设置，产生一个计数时钟(TCLK)和一个采样时钟。	

CCS mn	通道n的计数时钟(TCLK) 选择
0	工作时钟MCK由CKSmn位指定
1	将Tlmn引脚/副系统时钟的输入信号有效沿进行四分频(fsub/4)
计数时钟TCLK用于定时器/计数器、输出控制器和中断控制器。	

MAS TER mn	选择通道n作为独立操作功能或组合操作功能中作为从通道/组合操作功能中作为主通道
0	独立操作功能或组合操作功能中作为从通道。
1	组合操作功能中作为主通道。
只有偶数通道可以设置为主通道 (MASTERmn = 1)。 奇数通道一定要作为从通道使用 (MASTERmn = 0)。 对于作为独立操作功能使用的通道MASTERmn清除为0。	

注意事项 确保将位 14, 13, 5 和 4 清为“0”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7), mn = 00 至 07, 10 至 13

图 6-7. 定时器模式寄存器 mn (TMRmn) 的格式(2/3)

地址: F0190H, F0191H (TMR00) 至 F019EH, F019FH (TMR07) 复位后: 0000H R/W

F01C8H、F01C9H (TMR10) 至 F01CEH、F01CFH (TMR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKS mn	0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

STS mn2	STS mn1	STS mn0	通道n的启动触发或捕捉触发设置
0	0	0	只有软件触发启动有效(其它触发源不选择)。
0	0	1	TImn引脚输入的有效沿用作启动触发和捕捉触发。
0	1	0	TImn引脚输入的双沿分别作为启动触发和捕捉触发
1	0	0	使用主通道的中断信号(当通道用作组合操作功能中的从通道时)。
其它值			设置禁止

CIS mn1	CIS mn0	TImn引脚输入有效沿的选择
0	0	下降沿
0	1	上升沿
1	0	双沿 (当低电平宽度时) 启动触发: 下降沿, 捕捉触发: 上升沿
1	1	双沿 (当高电平宽度时) 启动触发: 上升沿, 捕捉触发: 下降沿
如果当STSmn2至STSmn0位的值不是010B时指定双沿, 设置CISmn1至CISmn0位为10B。		

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
 mn = 00 至 07, 10 至 13

图 6-7. 定时器模式寄存器 mn (TMRmn) 的格式(3/3)

地址: F0190H, F0191H (TMR00) 至 F019EH, F019FH (TMR07) 复位后: 0000H R/W

F01C8H, F01C9H (TMR10) 至 F01CEH, F01CFH (TMR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKS mn	0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

MD mn3	MD mn2	MD mn1	MD mn0	通道n的操作模式	TCR的计数操作	独立操作
0	0	0	1/0	间隔定时器模式	递减计数	可能
0	1	0	1/0	捕捉模式	递增计数	可能
0	1	1	0	事件计数器模式	递减计数	可能
1	0	0	1/0	一次计数模式	递减计数	不可能
1	1	0	0	捕捉&一次计数模式	递增计数	可能
其它值				设置禁止		
MDmn0 位的操作根据每种操作模式的不同而不同(参见下表)。						

操作模式 (由MDmn3至MDmn1位的值设置 (参考上表))	MD mn0	开始计数和中断的设置
- 间隔定时器模式 (0, 0, 0) - 捕捉模式 (0, 1, 0)	0	当计数开始时不产生定时器中断 (定时器输出也不改变)。
	1	当计数开始时, 定时器中断产生 (定时器输出也改变)。
事件计数器模式 (0, 1, 1)	0	当计数开始时不产生定时器中断 (定时器输出也不改变)。
一次计数模式^{※1} (1, 0, 0)	0	在计数操作期间启动触发无效。 在那时, 中断也不产生。
	1	在计数操作期间启动触发有效 ^{※2} 。 此期间也会产生中断。
捕捉&一次计数模式 (1, 1, 0)	0	当计数开始时不产生定时器中断 (定时器输出也不改变)。 在计数操作期间启动触发无效。 在那时, 中断也不产生。
其它值	设置禁止	

注 1. 在一次计数模式下, 当启动一个计数操作且未控制 TOMn 输出时, 中断输出 (INTTMmn)。
2. 如果操作期间产生了启动触发 (TSmn = 1), 计数器清零, 中断操作, 重新开始计数。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

(4) 定时器状态寄存器 mn (TSRmn)

TSRmn 指示通道 n 的计数器的溢出状态。

TSRmn 只有在捕捉模式(MDmn3 至 MDmn1 = 010B)和捕捉& 单计数模式(MDmn3 至 MDmn1 = 110B)模式下有效。在其它模式下不进行设置。对于 OVF 位在每种操作模式下的操作和设置/ 清零条件参见表 6-3。

TSRmn 可由 16 位存储器操作指令读取。

TSRmn 的低 8 位由 TSRmnL 通过 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 6-8. 定时器状态寄存器 mn (TSRmn) 的格式

地址: F01A0H, F01A1H (TSR00) 至 F01AEH, F01AFH (TSR07) 复位后: 0000H R

F01D0H、F01D1H (TSR10) 至 F01D6H、F01D7H (TSR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

OVF	通道n的计数器溢出状态
0	溢出不发生。
1	溢出发生。
OVF = 1时, 如果当下一个捕捉到的值没有溢出, 此标志清零 (OVF = 0) 。	

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
 mn = 00 至 07, 10 至 13

表 6-3. 在每种操作模式下 OVF 位操作和置位/清零条件

定时器操作模式	OVF	设置/清除条件
• 捕捉模式 • 捕捉&一次计数模式	清零	当捕捉时没有溢出发生
	置位	当捕捉时溢出已经发生
• 间隔定时器模式 • 事件计数器模式 • 一次计数模式	清零	— (禁止使用, 不能置位也不能清零)
	置位	

备注 OVF 位在计数器溢出以后不立即改变, 但是后面捕捉时改变。

(5) 定时器通道使能状态寄存器 m (TEm)

TEm 用于使能或停止每个通道的定时器操作。

当定时器通道启动寄存器 m (TSm) 的一位设为 1，则该寄存器的对应位设为 1。当定时器通道停止寄存器 m (TTm) 的一位设为 1，则该寄存器的对应位清零。

TEm 可由 16 位存储器操作指令读取。

TEm 的低 8 位由 TEmL 通过 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 6-9. 定时器通道使能状态寄存器 m (TEm)的格式

地址: F01B0H, F01B1H 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	0	0	0	0	TE07	TE06	TE05	TE04	TE03	TE02	TE01	TE00

地址: F01D8H, F01D9H 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE1	0	0	0	0	0	0	0	0	0	0	0	0	TE13	TE12	TE11	TE10

TE mn	指示通道n的操作使能/停止状态
0	操作停止
1	操作使能

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
 mn = 00 至 07, 10 至 13

(6) 定时器通道启动寄存器 m (TSm)

TSm 是用于定时器计数器 (TCRmn) 清零和启动每个通道计数操作的触发寄存器。
此寄存器的一位 (TSmn) 设置为 1 时, 定时器通道使能状态寄存器 m (TEm) 的相应位 (TEmn) 也会设置为 1。
TSmn 是触发位, 当 TEmn = 1 时立即清零。
TSm 可由 16 位存储器操作指令设置。
TSm 的低 8 位由 TSmL 通过 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 0000H。

图 6-10. 定时器通道启动寄存器 m (TSm)的格式

地址: F01B2H, F01B3H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	0	0	0	0	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

地址: F01DAH, F01DBH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1	0	0	0	0	0	0	0	0	0	0	0	0	TS13	TS12	TS11	TS10

TSmn	通道n的操作使能(启动)触发
0	没有触发操作
1	把TEmn设为1且使能计数操作。 TCRmn 在计数操作使能状态下启动的计数操作根据每个操作模式的不同而不同(参考表 6-4)。

注意事项 并且确保清除 TS1 的位 15 至 8 和 15 至 4 为“0”。

- 备注
- 1. 当读取 TSm 寄存器时, 读取值始终为 0。
 - 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

表 6-4. 从计数操作使能状态到 TCRmn 计数启动的操作(1/2)

定时器操作模式	设置 TSmn = 1时的操作
• 间隔定时器模式	从启动触发检测(TS0mn=1)直到计数时钟发生没有操作执行。 第一个计数时钟加载TDRmn的值到TCRmn, 后面的计数时钟执行向下计数操作(参见 6.3 (6) (a)间隔计数器模式下的开始时序)。
• 事件计数器模式	把1写入TSmn位以把TDRmn的值加载至TCRmn。 后面的计数时钟执行向下计数操作。 由TMRmn寄存器的STSmn2至STSmn0位选择外部触发检测不启动计数操作(参考 6.3 (6) (b) 事件计数器模式下的开始时序)。
• 捕捉模式	从启动触发检测直到计数时钟发生没有操作执行。 第一个计数时钟加载0000H至TCRmn, 后面的计数时钟执行向上计数操作(参考 6.3 (6) (c)捕捉模式下的开始时序)。

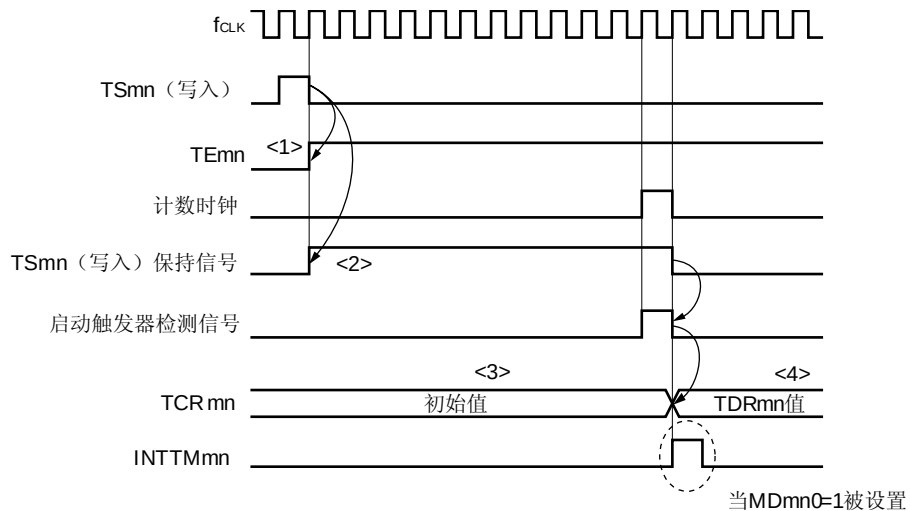
表 6-4. 从计数操作使能状态到 TCRmn 计数启动的操作(2/2)

定时器操作模式	设置 TSmn = 1时的操作
一次计数模式	当TEmn = 0时, 把1写入TSmn位以设置启动触发等待状态。 从启动触发检测直到计数时钟发生没有操作执行。 第一个计数时钟加载TDRmn的值至TCRmn, 后面的计数时钟执行向下计数操作 (参考6.3 (6) (d) 一次计数模式下的开始时序)。
捕捉&一次计数模式	当TEmn = 0时, 把1写入TSmn位以设置启动触发等待状态。 从启动触发检测直到计数时钟发生没有操作执行。 第一个计数时钟加载0000H到TCRmn, 后续计数时钟执行向上计数操作 (参见6.3 (6) (e)捕捉&一次计数 模式下的开始时序)。

(a) 间隔定时器模式下的开始时序

- <1> 向 TSmn 写入 1 以设置 TE_{mn} = 1
- <2> 写入 TSmn 的数据会一直保持, 直到产生计数时钟。
- <3> TCRmn 初始值会一直保持, 直到产生计数时钟。
- <4> 计数时钟产生时, 将“TDRmn 值”加载到 TCRmn 并且开始计数。

图 6-11. 开始时序(在间隔定时器模式下)

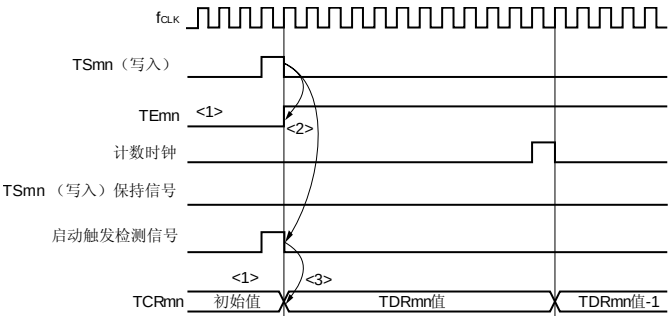


注意事项 写入 TSmn 后在计数时钟的第一个周期操作中, 由于开始计数延迟到了计数时钟产生后, 所以产生一个时钟的最大错误。当需要知道计数开始时间的时候, 通过设置 MDmn0 = 1 可在计数开始时产生一个中断。

(b) 事件计数器模式下的开始时序

- <1> TEmn 设置为 0 时，TCRmn 保持初始值。
- <2> 向 TSmn 写入 1 以设置 TEmn = 1
- <3> 一旦“1”写入 TSmn 以及 TEmn 设置为“1”，加载“TDRmn 值”到 TCRmn 上并开始计数。
- <4> 然后，TCRmn 的值根据计数时钟向下计数。

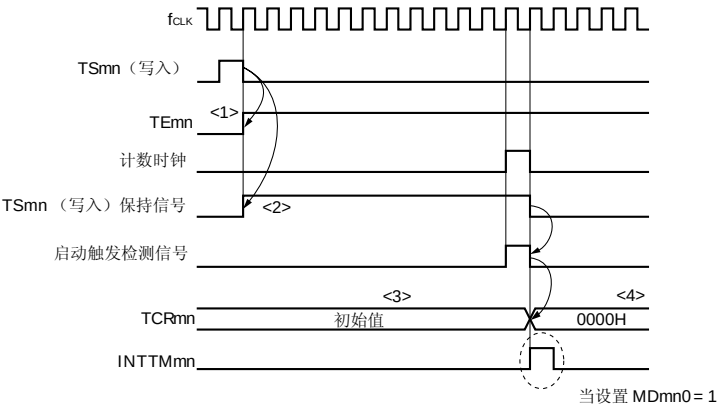
图 6-12. 开始时序(事件计数器模式下)



(c) 捕捉模式下的开始时序

- <1> 向 TSmn 写入 1 以设置 TEmn = 1
- <2> 写入 TSmn 的数据会一直保持，直到产生计数时钟。
- <3> TCRmn 初始值会一直保持，直到产生计数时钟。
- <4> 计数时钟产生时，0000H 加载到 TCRmn 并且开始计数。

图 6-13. 开始时序(捕捉模式下)

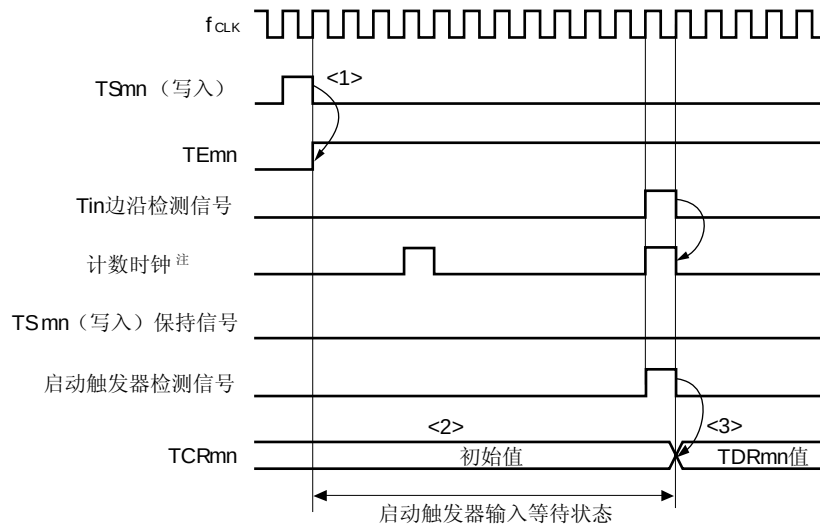


注意事项 写入 TSmn 后在计数时钟的第一个周期操作中，由于开始计数延迟到了计数时钟产生后，所以产生一个时钟的最大错误。当需要知道计数开始时间的时候，通过设置 MDmn0 = 1 可在计数开始时产生一个中断。

(d) 一次计数模式下的开始时序

- <1> 向 TSmn 写入 1 以设置 TEmn = 1
- <2> 进入启动触发输入等待状态, TCRmn 保持初始值。
- <3> 检测到启动触发, “TDRmn 值”加载到 TCRmn 并且计数开始。

图 6-14. 开始时序(一次计数模式下)



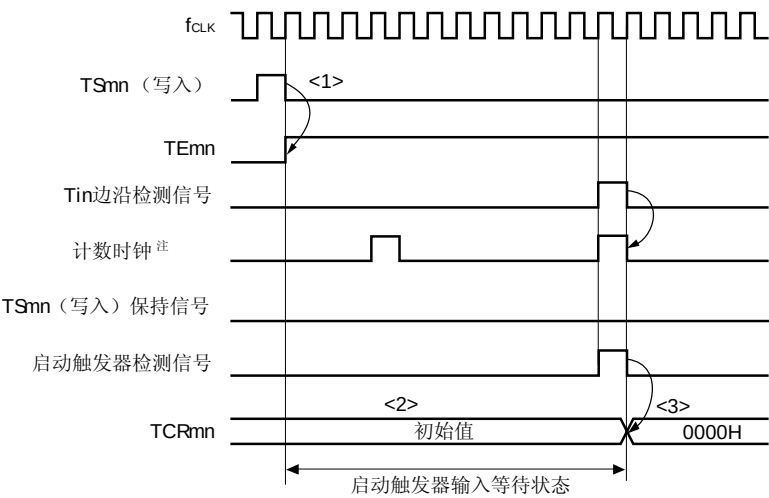
注 当设置单计数模式时, 操作时钟(MCK)选择作为计数时钟(CCSmn = 0)。

注意事项 因为需要检测启动触发才能开始操作, 会产生一个输入信号采样误差。(使用 $Tlmn$ 时, 误差为一个计数时钟)

(e) 捕捉&一次计数模式下的开始时序

- <1> 向 TSmn 写入 1 以设置 TEMn = 1
- <2> 进入启动触发输入等待状态，TCRmn 保持初始值。
- <3> 检测到启动触发，0000H 加载到 TCRmn 并且计数开始。

图 6-15. 开始时序(在捕捉和一次计数模式下)



注 当设置捕捉&单计数模式时，操作时钟(MCK)选择作为计数时钟(CCSmn = 0)。

注意事项 因为需要检测启动触发才能开始操作，会产生一个输入信号采样误差。(使用 TImn 时，误差为一个计数时钟)

(7) 定时器通道停止寄存器 m (TTm)

TTm 是用于定时器计数器 (TCRmn) 清零和启动每个通道计数操作的触发寄存器。
当此寄存器的某位 (TTmn) 设置为 1 时，定时器通道使能状态寄存器 m (TEm) 的相应位 (TEmn) 清为 0。
TTm 可由 16 位存储器操作指令设置。
TTm 低 8 位可由 TTmL 通过 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 0000H。

图 6-16. 定时器通道停止寄存器 m (TTm)的格式

地址: F01B4H, F01B5H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	0	0	0	0	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

地址: F01DCH, F01DDH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT1	0	0	0	0	0	0	0	0	0	0	0	0	TT13	TT12	TT11	TT10

TTmn	通道n的操作停止触发														
0	没有触发操作														
1	操作停止(停止触发产生)。														

注意事项 并且确保清除 TT0 的位 15 至 8 和 TT1 的位 15 至 4 为“0”。

备注 1. 当读取 TTm 寄存器时，读取值始终为 0。
2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

(8) 定时器输入选择寄存器 m(TISm)

TISm 用于选择对每个通道定时器输入引脚 (Timn) 的输入信号或 4 分频的副系统时钟 (fSUB/4) 是否有效。
TISm 可由 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 00H。

图 6-17. 定时器输入选择寄存器 m (TISm)的格式

地址: FFF3EH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TIS0	TIS07	TIS06	TIS05	TIS04	TIS03	TIS02	TIS01	TIS00

地址: FFF3FH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TIS1	0	0	0	0	TIS13	TIS12	TIS11	TIS10

TISmn	通道n定时器输入/使用的副系统时钟的选择
0	定时器输入引脚 (Timn) 的输入信号
1	副系统时钟4分频 (fSUB/4)

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
 mn = 00 至 07, 10 至 13

(9) 定时器输出使能寄存器 m (TOEm)

TOEm 用于使能或禁止每个通道的定时器输出。

定时器输出使能的通道 n 将无法用软件重写定时器输出寄存器 (TOMn) 的 TOMn 位的值，该值通过计数操作反映了定时器输出功能的设置，并从定时器输出引脚 (TOMn) 输出。

TOEm 可由 16 位存储器操作指令设置。

TOEm 可由 TOEmL 通过 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 6-18. 定时器输出使能寄存器 m (TOEm)的格式

地址: F01BAH, F01BBH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE0	0	0	0	0	0	0	0	0	TOE 07	TOE 06	TOE 05	TOE 04	TOE 03	TOE 02	TOE 01	TOE 00

地址: F01E2H, F01E3H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE1	0	0	0	0	0	0	0	0	0	0	0	0	TOE 13	TOE 12	TOE 11	TOE 10

TOE mn	通道n的定时器输出使能/禁止
0	TOMn操作计数操作 (定时器通道输出位) 停止。 使能写入TOMn位。 TOMn引脚作为数据输出功能，并且其输出电平设置为TOMn。 TOMn引脚的输出电平可由软件进行操作。
1	TOMn操作通过计数操作 (定时器通道输出位) 使能。 禁止写入TOMn位 (写入忽略)。 TOMn引脚作为定时器输出功能，且TOMn位根据定时器的运行进行置1或清零。 TOMn引脚可以根据定时器的运行输出方波或PWM。

注意事项 并且确保清除 TOE0 的位 15 至 8 和 TOE1 的位 15 至 4 为“0”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
 mn = 00 至 07, 10 至 13

(10) 定时器输出寄存器 m (TOm)

TOm 是每个通道的定时器输出的一个缓冲寄存器。

此寄存器中每个位的值从每个通道的定时器输出引脚 (TOmn) 输出。

只有当定时器输出禁止 (TOEmn = 0) 时此寄存器可以通过软件重写。当定时器输出使能时 (TOEmn = 1)，通过软件重写此寄存器忽略，只有定时器操作才会改变该值。

要使用 P01/TO00, P16/TO01, P17/TO02, P31/TO03, P42/TO04, P46/TO05, P131/TO06, P145/TO07, P64/TO10, P65/TO11, P66/TO12 或 P67/TO13 引脚作为端口功能引脚，设定相应 TOmn 位为“0”。

TOm 可由 16 位存储器操作指令设置。

TOm 可由 TOML 通过 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 6-19. 定时器输出寄存器 m (TOm)的格式

地址: F01B8H, F01B9H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	0	0	TO0 7	TO0 6	TO0 5	TO0 4	TO0 3	TO0 2	TO0 1	TO0 0

地址: F01E0H, F01E1H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO1	0	0	0	0	0	0	0	0	0	0	0	0	TO1 3	TO1 2	TO1 1	TO1 0

TO mn	通道n的定时器输出
0	定时器输出值为“0”。
1	定时器输出值为“1”。

注意事项 并且确保清除 TO0 的位 15 至 8 和 TO1 的位 15 至 4 为“0”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
 mn = 00 至 07, 10 至 13

(11) 定时器输出电平寄存器 m (TOLm)

TOLm 是控制每个通道定时器输出电平的寄存器。

当在组合操作模式 (TOMmn = 1) 中使能定时器输出 (TOEmn = 1) 时，通过此寄存器设置通道 n 反向输出，反映在对定时器输出信号的置位或复位。在反转模式 (TOMmn = 0) 下，此寄存器设置无效。

TOLm 可由 16 位存储器操作指令设置。

TOLm 可由 TOLmL 通过 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 6-20. 定时器输出电平寄存器 m (TOLm)的格式

地址: F01BCH, F01BDH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	0	0	TOL 07	TOL 06	TOL 05	TOL 04	TOL 03	TOL 02	TOL 01	TOL 00

地址: F01E4H, F01E5H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL1	0	0	0	0	0	0	0	0	0	0	0	0	TOL 13	TOL 12	TOL 11	TOL 10

TOL mn	通道n的定时器输出电平控制														
0	正向逻辑输出(高有效)														
1	反向输出(低有效)														

注意事项 并且确保清除 TOL0 的位 15 至 8 和 TOL1 的位 15 至 4 为“0”。

- 备注
- 1. 如果此寄存器的值在定时器操作期间重写，当定时器输出信号变为下一个，定时器输出反向，在寄存器值重写后立即代替。
 - 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7), mn = 00 至 07, 10 至 13

(12) 定时器输出模式寄存器 m (TOMm)

TOMm 用于每个通道的定时器输出模式控制。

当通道用于独立操作功能时，设置所使用通道的相应位为 0。

当通道用于组合操作功能(PWM 输出，单脉冲输出或多路 PWM 输出)时，设置主通道的相应位为 0，从通道的相应位为 1。

通过此寄存器对各通道 n 的设置反映在定时器输出使能 (TOEmn = 1) 时定时器输出信号的设置或复位。

TOMm 可由 16 位存储器操作指令设置。

TOMm 可由 TOMmL 通过 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 6-21. 定时器输出模式寄存器 m (TOMm)的格式

地址: F01BEH, F01BFH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	0	0	TOM07	TOM06	TOM05	TOM04	TOM03	TOM02	TOM01	TOM00

地址: F01E6, F01E7H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM1	0	0	0	0	0	0	0	0	0	0	0	0	TOM13	TOM12	TOM11	TOM10

TOMmn	通道n的定时器输出模式控制
0	反转模式 (由定时器中断请求信号 (INTTMmn) 产生反转输出)
1	组合操作模式(由主通道的定时器中断请求信号 (INTTMmn)设置，由从通道的定时器中断请求信号(INTTMmp)复位)

注意事项 并且确保清除 TOM0 的位 15 至 8 和 TOM1 的位 15 至 4 为“0”。

备注 m: 单元编号, n: 通道编号, p: 从通道编号

 当 m = 0

 n = 0 至 7(主通道 n = 0, 2, 4, 6)

 n < p ≤ 7(此处 p 是一个大于 n 的连续整数)

 当 m = 1

 n = 0 至 3(主通道 n = 0)

 n < p ≤ 3(此处 p 是一个大于 n 的连续整数)

(13) 输入切换控制寄存器 (ISC)

通过联合使用通道 7 和串行阵列单元，ISC 寄存器的 ISC1 和 ISC0 位用于实现 LIN 总线通信操作。

当 ISC1 的位 1 设置为 1 时，串行数据输入引脚 (RxD3) 的输入信号选择为定时器输入信号。

ISC 可以通过一个 1 位或者 8 位存储器操作指令来设置。

复位后这个寄存器的值为 00H。

图 6-22. 输入切换控制寄存器(ISC)的格式

地址: FFF3CH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	切换定时器阵列单元(TAU0)的通道7输入
0	使用TI07引脚的输入信号作为定时器输入(正常模式)。
1	RxD3引脚的输入信号用作定时器输入(唤醒信号检测)。

ISC0	切换外部中断 (INTP0)输入
0	使用INTP0引脚的输入信号作为外部中断(正常模式)。
1	使用RxD3引脚输入信号作为外部中断 (用来测量同步中断域和同步域的脉冲宽度)。

注意事项 确保将位 7 至 3 清“0”。

备注 当使用 LIN 总线通讯功能时，设置 ISC1=1 选择 RxD0 的输入信号。

(14) 噪声滤波器使能寄存器 1、2(NFEN1、NFEN2)

NFEN1 用于设置是否在定时器阵列单元 0 的定时器输入信号输入到通道时使用噪声滤波器。

NFEN2 用于设置是否在定时器阵列单元 1 的定时器输入信号输入到通道时使用噪声滤波器。

通过在噪声消除需要引脚上设置相应位为 1，使能噪声滤波器。

当噪声滤波器开启时，执行匹配检测和与 CPU/外围硬件时钟(f_{clk})的 2 个时钟同步。当噪声滤波器关闭时，只执行与 CPU/外围硬件时钟(f_{clk})同步操作。

NFEN1 和 NFEN2 可以由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 6-23. 噪音过滤使能寄存器 1 (NFEN1)

地址: F0061H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	使能/禁止使用噪声滤波器的TI07/TO07/P145引脚或RxD3/P14引脚输入信号 ^注
0	噪声过滤关闭
1	噪声过滤开启

TNFEN06	使能/禁止使用TI06/TO06/P131引脚输入信号的噪声滤波器
0	噪声过滤关闭
1	噪声过滤开启

TNFEN05	使能/禁止使用TI05/TO05/INTP1/P46引脚输入信号的噪声滤波器
0	噪声过滤关闭
1	噪声过滤开启

TNFEN04	使能/禁止使用TI04/TO04/P42引脚输入信号的噪声滤波器
0	噪声过滤关闭
1	噪声过滤开启

TNFEN03	使能/禁止使用TI03/TO03/INTP4/P31引脚输入信号的噪声滤波器
0	噪声过滤关闭
1	噪声过滤开启

TNFEN02	使能/禁止使用TI02/TO02/P17引脚输入信号的噪声滤波器
0	噪声过滤关闭
1	噪声过滤开启

TNFEN01	使能/禁止使用TI01/TO01/INTP5/P16引脚输入信号的噪声滤波器
0	噪声过滤关闭
1	噪声过滤开启

TNFEN00	使能/禁止使用TI00/P00引脚输入信号的噪声滤波器
0	噪声过滤关闭
1	噪声过滤开启

注 可通过设置 ISC 寄存器的 ISC1 切换可应用的引脚。

ISC1 = 0: 可选择是否使用 TI07 引脚的噪声滤波器。

ISC1 = 1: 可选择是否使用 RxD3 引脚的噪声滤波器。

图 6-24. 噪声过滤使能寄存器 2 (NFEN2)

地址: F0062H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
NFEN2	0	0	0	0	TNFEN13	TNFEN12	TNFEN11	TNFEN10

TNFEN13	使能/禁止使用TI13/TO13/P67引脚输入信号的噪声滤波器
0	噪声过滤关闭
1	噪声过滤开启

TNFEN12	使能/禁止使用TI12/TO12/P66引脚输入信号的噪声滤波器
0	噪声过滤关闭
1	噪声过滤开启

TNFEN11	使能/禁止使用TI11/TO11/P65引脚输入信号的噪声滤波器
0	噪声过滤关闭
1	噪声过滤开启

TNFEN10	使能/禁止使用TI10/TO10/P64引脚输入信号的噪声滤波器
0	噪声过滤关闭
1	噪声过滤开启

(15) 端口模式寄存器 0, 1, 3, 4, 6, 13, 14 (PM0, PM1, PM3, PM4, PM6, PM13, PM14)

这些寄存器以 1 位单元设置端口 0、1、3、4、6、13 和 14 的输入/输出。

当使用 P01/TO00, P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, P42/TO04/TI04, P46/TO05/TI05/INTP1, P131/TO06/TI06, P145/TO07/TI07 和 P64/TI10/TO10 至 P67/TI13/TO13 引脚作为定时器输出时, 设置 PM01, PM16, PM17, PM31, PM42, PM46, PM131, PM145 和 PM64 至 PM67 以及 P01, P16, P17, P31, P42, P46, P131, P145 和 P64 至 P67 的输出锁存为 0。

当使用 P00/TI00, P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, P42/TO04/TI04, P46/TO05/TI05/INTP1, P131/TO06/TI06, P145/TO07/TI07 和 P64/TI10/TO10 至 P67/TI13/TO13 引脚作为定时器输入时, 设置 PM00, PM16, PM17, PM31, PM42, PM46, PM131, PM145 和 PM64 至 PM67 为 1。此时, P00, P16, P17, P31, P42, P46, P131, P145 和 P64 至 P67 的输出锁存可能为 0 或 1。

PM0, PM1, PM3, PM4, PM6, PM13 和 PM14 可由 1 位或 8 位存储器操作指令设置。

复位信号的产生将这些寄存器设置为 FFH。

图 6-25. 端口模式寄存器 0, 1, 3, 4, 6, 13, 14 (PM0, PM1, PM3, PM4, PM6, PM13, PM14)的格式

地址: FFF20H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00

地址: FFF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

地址: FFF23H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

地址: FFF24H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

地址: FFF26H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60

地址: FFF2DH 复位后: FEH R/W

符号	7	6	5	4	3	2	1	0
PM13	1	1	1	1	1	1	PM131	0

地址: FFF2EH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140

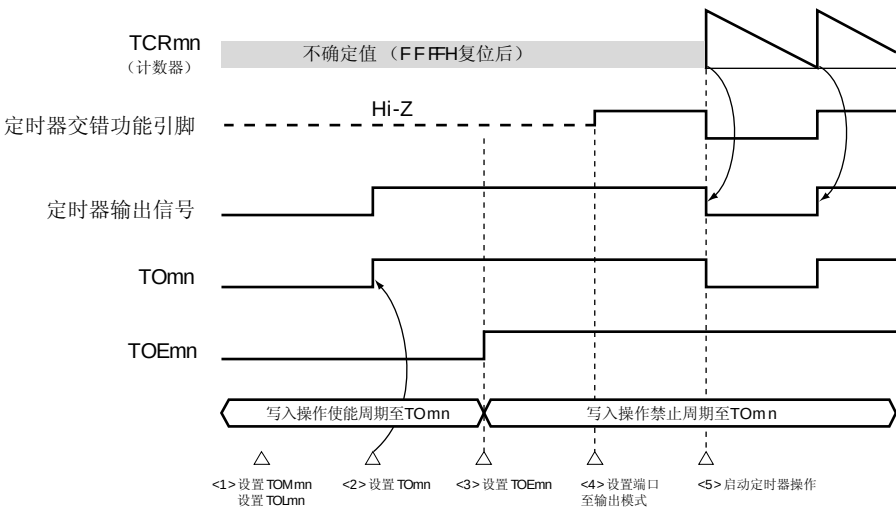
PMmn	Pmn引脚I/O模式选择 (m = 0、1、3、4、6、13、14; n = 0 至 7)						
0	输出模式(输出缓冲器开)						
1	输入模式(输出缓冲器关)						

备注 m: 单元编号, n: 通道编号, p: 从通道编号
当 m = 0
 n = 0 至 7(主通道 n = 0, 2, 4, 6)
 n < p ≤ 7(此处 p 是一个大于 n 的连续整数)
当 m = 1
 n = 0 至 3(主通道 n = 0)
 n < p ≤ 3(此处 p 是一个大于 n 的连续整数)

6.4.2 TOMn 引脚输出设置

下图所示为 TOMn 输出引脚从初始设置到定时器操作开始的过程和状态转换。

图 6-27. 从定时器输出设置到操作开始的状态转换



<1> 设置定时器输出的操作模式。

- TOMmn 位 (0: 反转模式, 1: 组合操作模式)
- TOLmn 位 (0: 正向输出, 1: 反向输出)

<2> 定时器输出信号通过设置 TOMn 恢复到初始状态。

<3> 通过写入 1 到 TOEmn 可以使能定时器输出操作 (TOMn 禁止写入)。

<4> 端口 I/O 设置为输出 (参考 6.3 (14) 端口模式寄存器 0, 1, 3, 4, 6, 13, 14 (PM0, PM1, PM3, PM4, PM6, PM13, PM14))。

<5> 使能定时器操作 (TSmn = 1)。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
 mn = 00 至 07, 10 至 13

6.4.3 通道输出操作的注意事项

(1) 定时器操作期间改变寄存器 TOM,TOEm,TOLm 和 TOMm 中的设置值

因为定时器操作 (TCRmn 和 TDRmn 的操作) 与 TOMn 输出电路无关, 并且改变设置 TOM、TOEm、TOLm 和 TOMm 值不会影响定时器操作, 所以在定时器操作期间这些值可以改变。
当 TOEm, TOLm, 和 TOMm(除 TOM 外) 中设置的值在靠近定时器中断 (INTTMmn)时改变, 根据该值是紧随定时器中断 (INTTMmn) 信号之前或之后改变, 输出到 TOMn 引脚的波形可能会不同。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
 mn = 00 至 07, 10 至 13

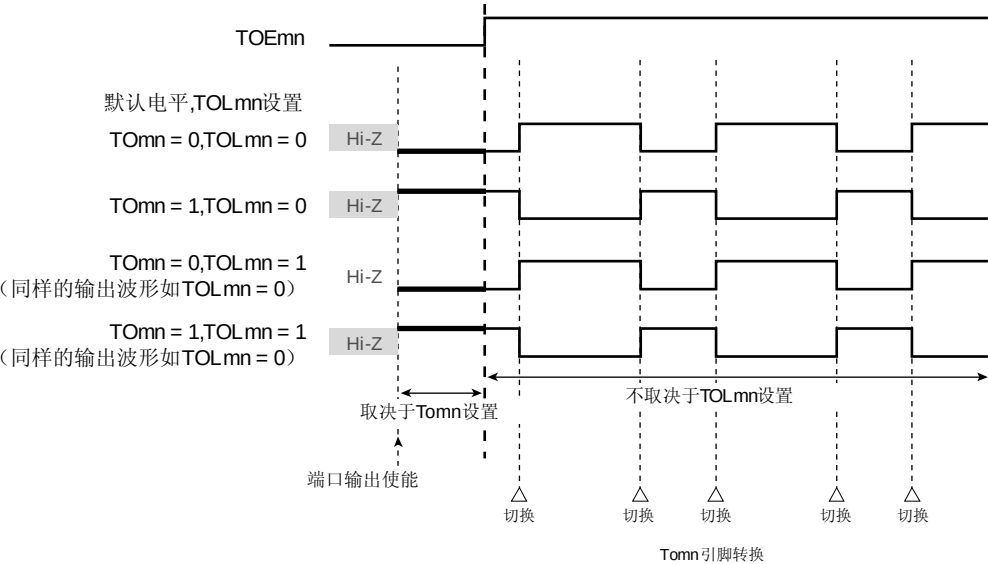
(2) TOMn 引脚的默认电平和定时器操作开始后输出电平

在改变默认电平之后, 在使能端口输出和设置 TOEmn = 1 之前, 当在 TOEmn = 0 的状态写入完成时 TOMn 引脚输出电平的转换如下图所示。

(a) 设置 TOMmn = 0 启动操作时 (反转输出)

当 TOMmn = 0 时, 对 TOLmn 的设置无效。当定时器在设置默认电平后开始运行时, 会产生反转信号并且 TOMn 引脚的输出电平会反向。

图 6-28. 反转输出时 TOMn 引脚输出状态 (TOMmn = 0)

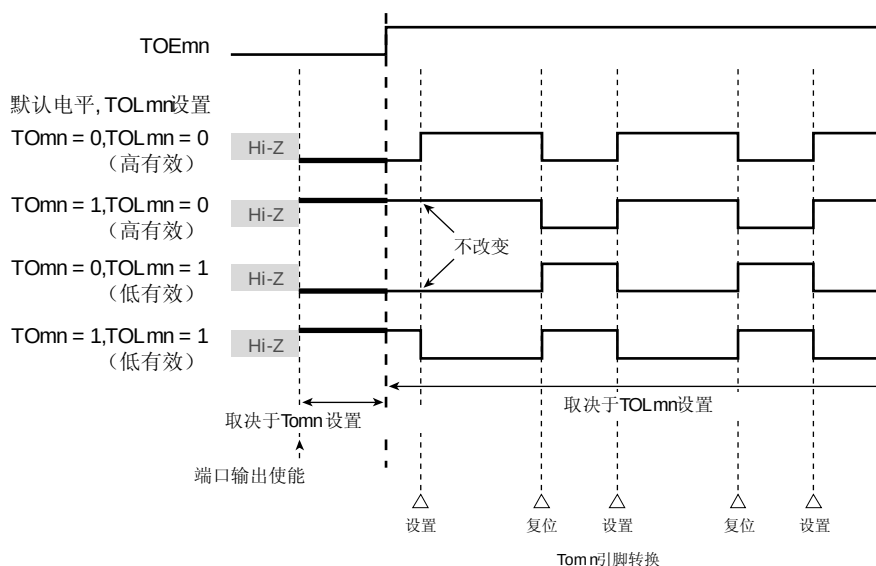


备注 1. 反转: 反转 TOMn 引脚的输出状态
2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

(b) 当操作开始时, 设置 TOMmn = 1(组合操作模式(PWM 输出))

TOMmn = 1 时，有效电平由 TOLmn 设置决定。

图 6-29. PWM 输出时 TOMn 引脚输出状态 (TOMmn = 1)



备注 1. 设置: T0mn 引脚的输出信号从无效电平变为有效电平。

复位: T0mn 引脚的输出信号从有效电平变为无效电平。

2. m: 单元编号 ($m = 0, 1$), n: 通道编号 ($n = 0$ 至 7),

mn = 00 至 07, 10 至 13

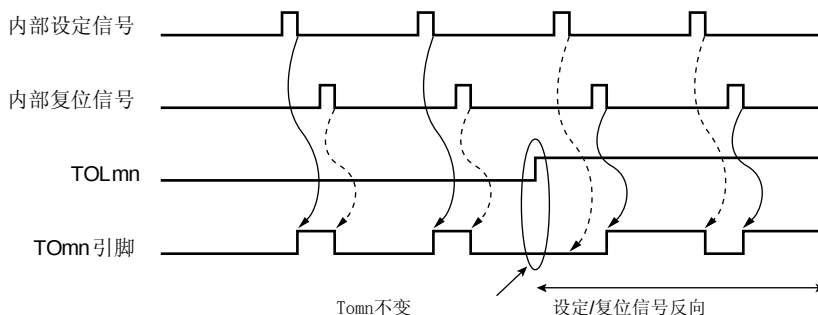
(3) T0mn 引脚在组合操作模式 (TOMmn = 1) 下的操作

(a) 当 TOLmn 设置在定时器操作期间已经改变

当在定时器操作期间改变 TOLmn 设置，当 TOMn 改变条件的时序发生时该设置变为有效。重写 TOLmn 不会改变 TOMn 的输出电平。

下图所示为 TOLmn 的值在定时器操作期间 (TOMmn = 1) 改变时的操作。

图 6-30. 当 TOLmn 在定时器操作期间改变时的操作



备注 1. 设置: TOmn 引脚的输出信号从无效电平变为有效电平。

复位: T0mn 引脚的输出信号从有效电平变为无效电平。

2. m : 单元编号 ($m = 0, 1$), n : 通道编号 ($n = 0$ 至 7),

mn = 00 至 07, 10 至 13

(b) 设置/复位的时序

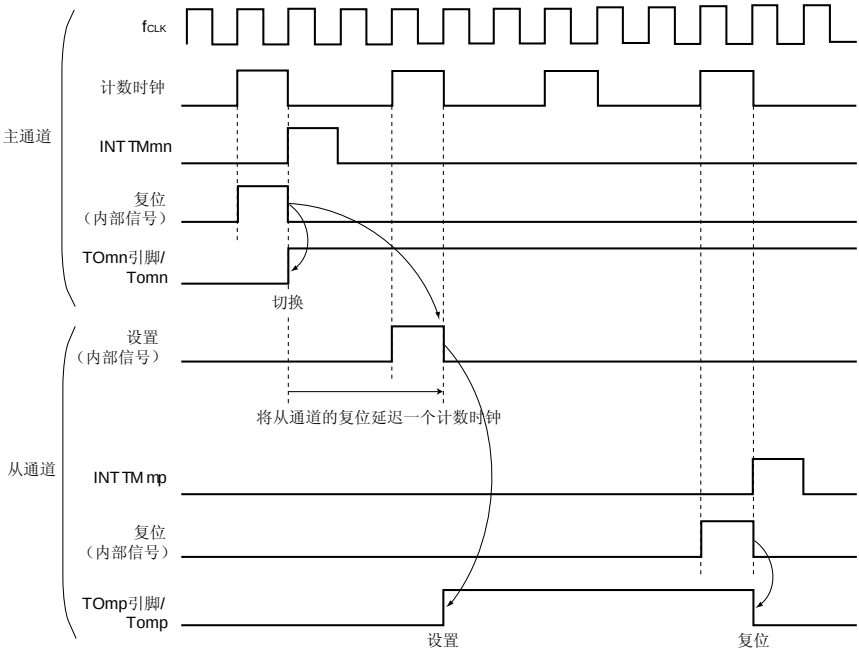
PWM 输出要实现 0%/100%输出，主通道定时器中断(INTTMmn)发生时，TOMn 引脚/TOMn 的设置时序通过从通道延迟一个计数时钟。

如果设置条件和复位条件同时满足，复位具有较高的优先级。

图 6-31 所示为主/从通道设置如下时，置位/复位操作状态。

主通道：TOEmn = 1, TOMmn = 0, TOLmn = 0
从通道：TOEmp = 1, TOMmp = 1, TOLmp = 0

图 6-31. 置位/复位时序操作状态

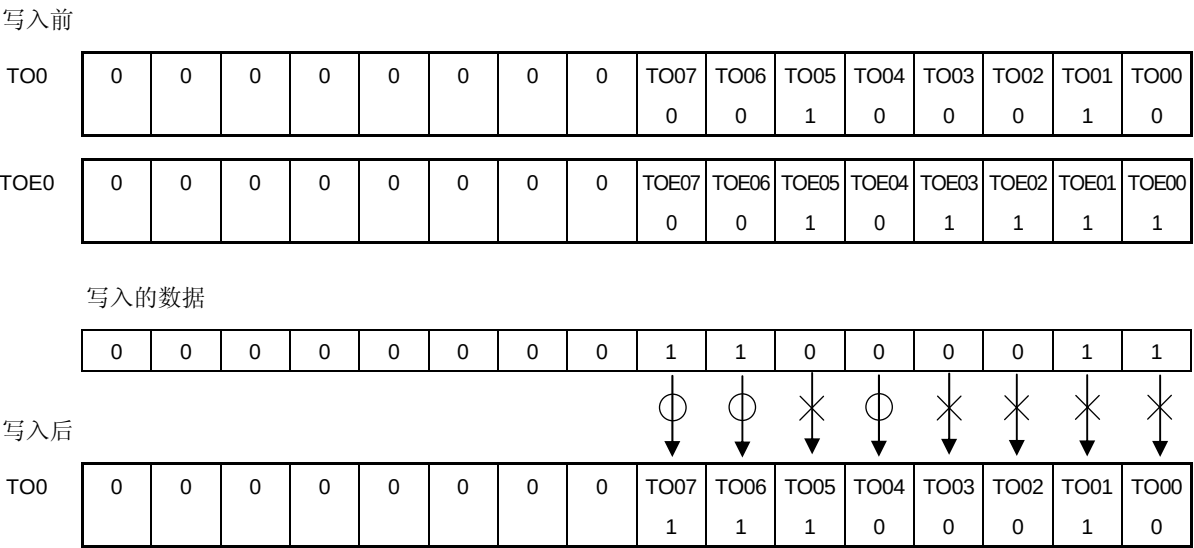


- 备注
1. 复位：TOMn 引脚复位/反转信号
置位：TOMn 引脚置位信号
 2. m: 单元编号, n: 通道编号, p: 从通道编号
当 m = 0
n = 0 至 7(主通道 n = 0, 2, 4, 6)
n < p ≤ 7(此处 p 是一个大于 n 的连续整数)
当 m = 1
n = 0 至 3(主通道 n = 0)
n < p ≤ 3(此处 p 是一个大于 n 的连续整数)

6.4.4 T0mn 位的集中操作

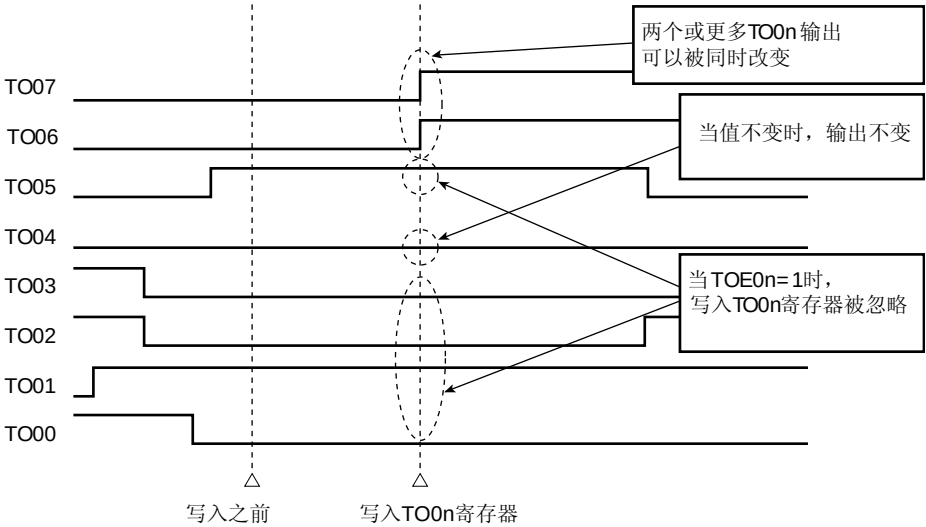
在 T0m 寄存器中，所有通道的设置位可在一个寄存器中定位，与 TSm 寄存器(通道启动触发器)相同。因此所有通道的 T0mn 可集中操作。另外只有通过设置和想要操作的 T0mn(通道输出)所对应的 TOEmn = 0 才可以进行逐位操作。

图 6-32. T00n 位集中操作的示例



只有 TOEmn = 0 可以写入 T0mn 位，忽略 TOEmn = 1 写入 T0mn。
设置为 TOEmn = 1 的 T0mn（通道输出）不受写操作影响。即使对 T0mn 进行写操作，会将其忽略并且定时器操作改变的输出正常完成。

图 6-33 T00n 位的集中操作影响 T00n 引脚状态



(注意事项和备注在下一页给出.)

注意 当 $TOEmn = 1$ 时，即使各个定时器的定时器中断 ($INTTMmn$) 输出和写入 $TOmn$ 的内容有冲突， $TOmn$ 引脚也可以正常输出。

备注 m : 单元编号 ($m = 0, 1$), n : 通道编号 ($n = 0$ 至 7),
 $mn = 00$ 至 $07, 10$ 至 13

6.4.5 定时器中断和操作开始时的 $TOmn$ 引脚输出

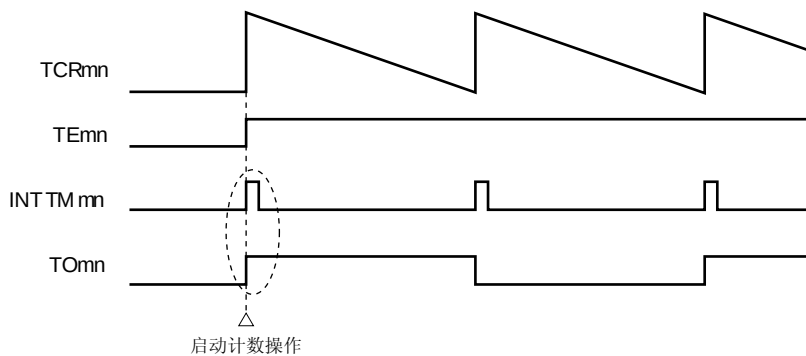
在间隔定时器模式或捕捉模式下， $TMRmn$ 寄存器中的 $MDmn0$ 位用于设置是否在计数启动时产生定时器中断。

当 $MDmn0$ 设置为 1 时，通过定时器中断 ($INTTMmn$) 的产生就可以知道计数操作开始时间。

在其它模式下，在计数操作开始的定时器中断和 $TOmn$ 输出都不进行控制。

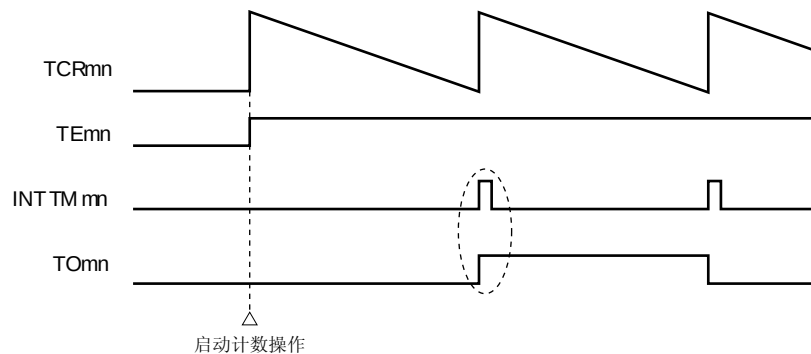
图 6-34 和 6-35 所示为使用间隔定时器模式 ($TOEmn = 1$, $TOMmn = 0$) 时的操作示例。

图 6-34. 当 $MDmn0$ 设置为 1



$MDmn0$ 设置为 1 时，定时器中断 ($INTTMmn$) 在计数操作开始时输出，并且 $TOmn$ 执行反转操作。

图 6-35. 当 $MDmn0$ 设置为 0



$MDmn0$ 设置为 0 时，定时器中断 ($INTTMmn$) 在计数操作开始时不输出，并且 $TOmn$ 也不改变。一个计数周期后， $INTTMmn$ 输出并且 $TOmn$ 执行反转操作。

备注 m : 单元编号 ($m = 0, 1$), n : 通道编号 ($n = 0$ 至 7),
 $mn = 00$ 至 $07, 10$ 至 13

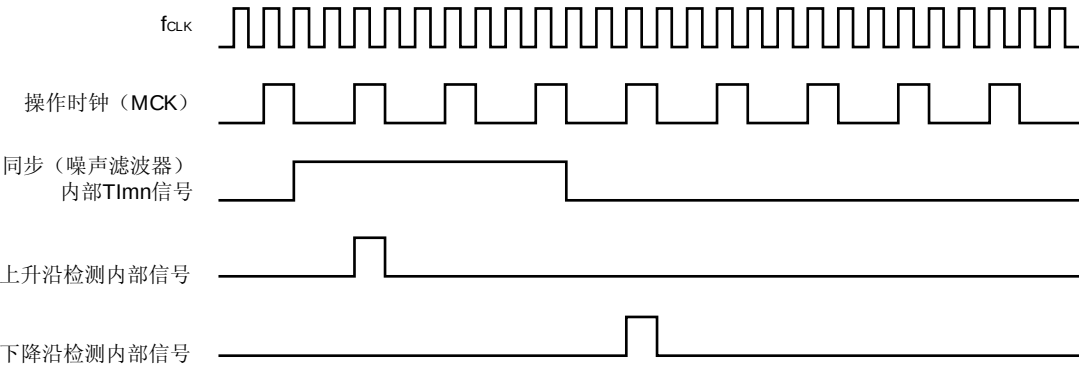
6.5 通道输入 (Tl_{mn} 引脚) 控制

6.5.1 Tl_{mn} 边沿检测电路

(1) 沿检测基本操作时序

沿检测电路采样的执行与工作时钟(MCK)一致。

图 6-36. 沿检测基本操作时序



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

6.6 定时器阵列单元的基本功能

6.6.1 独立操作功能和组合操作功能的概述

定时器阵列单元由若干个通道组成，独立操作功能使能每个通道独立操作，也可以和两个或更多通道组合在一起实现组合操作功能。

任意通道在其它通道的任何操作模式下均可使用独立操作功能。

组合操作功能通过将主通道(主要是计数周期的参考定时器)和一个从通道(与主通道操作一致的定时器)组合在一起实现，使用此功能时必须遵守下面几条规则。

6.6.2 组合操作功能的基本规则

使用组合操作功能的基本规则如下所示。

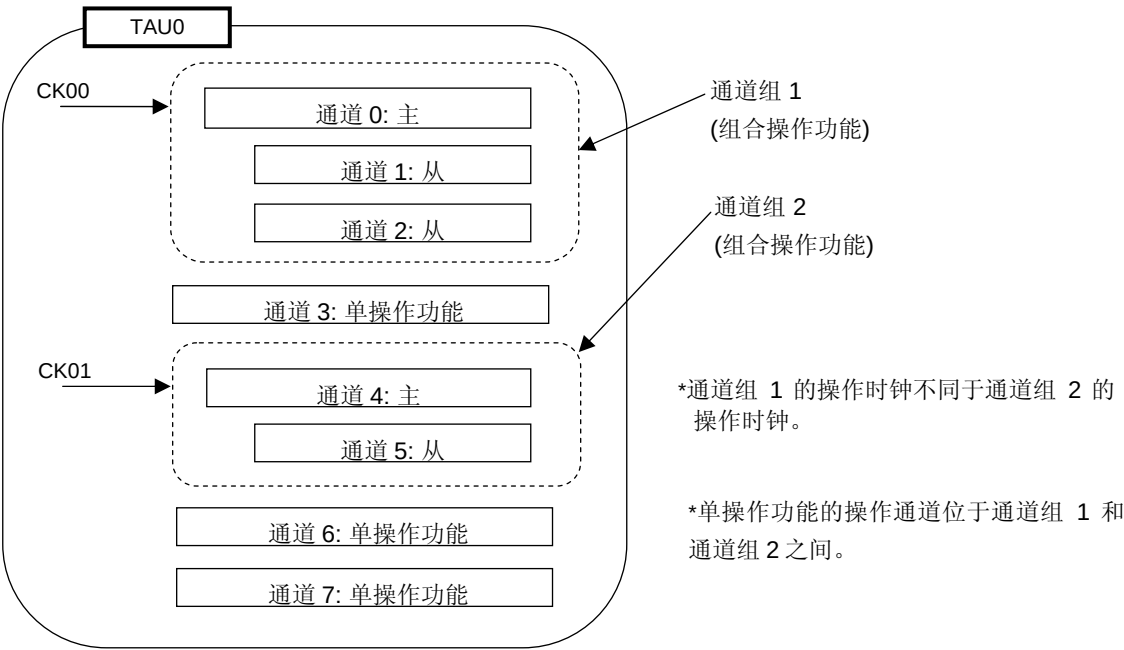
- (1) 只有偶数通道(通道 0, 2, 4 等)可设置为主通道。
- (2) 除通道 0 以外的任意通道都可设置为从通道。
- (3) 从通道必须低于主通道。
 举例：如果 TAU0 的通道 2 设置为主通道，通道 3 以后的通道 (通道 3, 4, 5, 等)可设置为从通道。
 如果 TAU1 的通道 0 设置为主通道，通道 1,2 或 3 (仅至 TAU1 上的通道 3。)可设置为从通道。
- (4) 一个主通道可以设置两个或更多从通道。
- (5) 当使用两个以上主通道时，与从通道之间有主通道的从通道不能设置。
 举例：如果 TAU0 的通道 0 和 4 设置为主通道，通道 1 到 3 可设置为主通道 0 的从通道。通道 5 至 7 不能设置作为主通道 0 的从通道。
- (6) 从通道与主通道组合时，从通道的操作时钟必须与主通道相同。与主通道一起操作的从通道的 CKS 位(TMRmn 寄存器的第 15 位)必须与主通道的相应位的值相同。
- (7) 主通道可以发送 INTTMmn(中断)，启动软件触发，计数时钟到更低通道。
- (8) 从通道可使用主通道的 INTTMmn (中断)，启动软件触发，和计数时钟，但是不能发送自己的 INTTMmn (中断)，启动软件触发，和计数时钟到更低通道。
- (9) 主通道不能使用来自其他更高主通道的 INTTMmn(中断)，启动软件触发和计数时钟。
- (10) 要同时启动组合操作的通道，组合通道的 TSmn 位必须同时设置。
- (11) 在计数操作期间，可以设置工作在主从通道或单是主通道的所有通道的 TSmn 位。仅从通道的 TSmn 不能进行设置。
- (12) 要同时停止参与组合的通道，组合通道的 TTmn 位必须同时设置。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
 mn = 00 至 07, 10 至 13

6.6.3 组合操作功能的基本规则的应用范围

组合操作功能的规则是适用于一个通道组(一个主通道和从通道组成一个组合操作功能)的规则。
如果指定两个或更多不能组合操作的通道组，则 **6.6.2 组合操作功能的基本规则**中的组合操作功能的基本规则不会应用到通道组。

示例



6.7 作为独立通道的定时器阵列单元的操作

6.7.1 作为间隔定时器/方波输出的操作

(1) 间隔定时器

定时器阵列单元可用作参考定时器，产生固定间隔的 INTTMmn (定时器中断)。
可通过下面的表达式计算出中断产生周期。

$$\text{INTTMmn (定时器中断) 的产生周期} = \text{计数时钟周期} \times (\text{TDRmn 的设置值} + 1)$$

除 CKm0 和 CKm1 之外，4 分频($f_{\text{SUB}}/4$)副系统时钟还可以选择作为计数时钟。因此，无论 f_{CLK} 频率(主系统时钟、副系统时钟)为多少，间隔定时器都可以通过计数时钟固定为 $f_{\text{SUB}}/4$ 。当改变由 f_{CLK} (改变系统时钟控制寄存器 (CKC) 的值) 选择的时钟时，但是需要先停止定时器阵列单元(TAU) (TTm = 00FFH)。

(2) 作为方波输出操作

一旦 INTTMmn 发生，TOMn 就执行反转操作，并且输出一个占空比为 50%的方波。
TOMn 输出方波的周期和频率可以通过下面的表达式计算出。

$$\bullet \text{ 从 TOMn 方波输出的周期} = \text{计数时钟周期} \times (\text{TDRmn 的设置值} + 1) \times 2$$

$$\bullet \text{ 从 TOMn 方波输出的频率} = \text{计数时钟频率} / \{(\text{TDRmn 的设置值} + 1) \times 2\}$$

TCRmn 在间隔定时器模式下作为向下计数器操作。

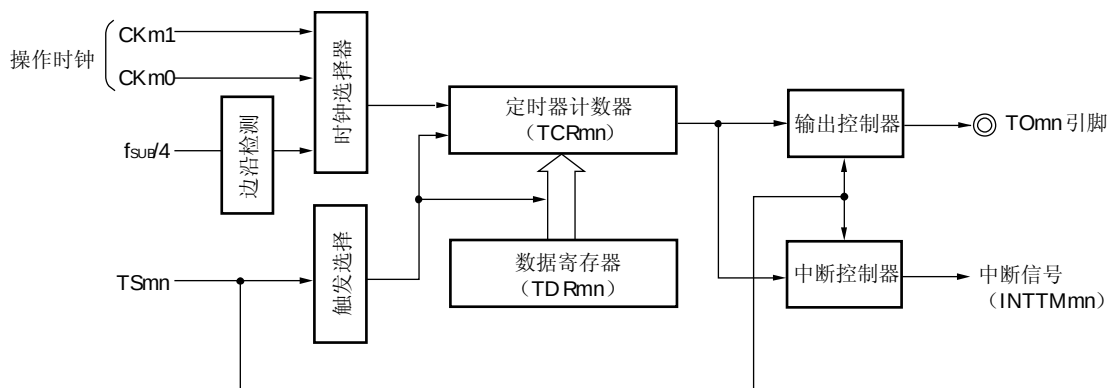
通道启动触发位 (TSmn) 设置为 1 后，在第一个计数时钟时，TCRmn 加载 TDRmn 的值。如果此时 TMRmn 的 MDmn0 = 0，则不产生 INTTMmn 并且 TOMn 不反转。如果 TMRmn 的 MDmn0 = 1，输出 INTTMmn 并且 TOMn 反转。

之后，TCRmn 随着计数时钟而向下计数。

当 TCRmn = 0000H 时，INTTMmn 输出并且 TOMn 在下一个计数时钟反转。此时 TCRmn 再次加载 TDRmn 的值。之后，重复相同的操作。

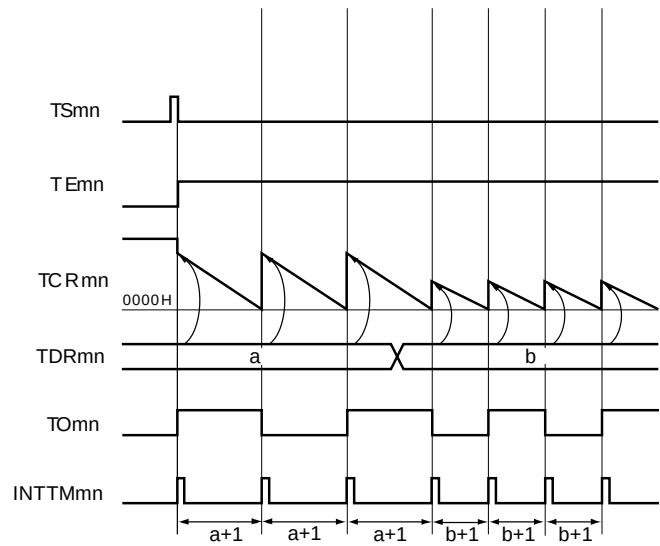
TDRmn 在任何时间可以重写。TDRmn 的新值从下个周期开始有效。

图 6-37. 作为间隔定时器/方波输出操作的框图



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-38. 作为间隔定时器/方波输出 (MDmn0= 1) 操作的基本时序示例

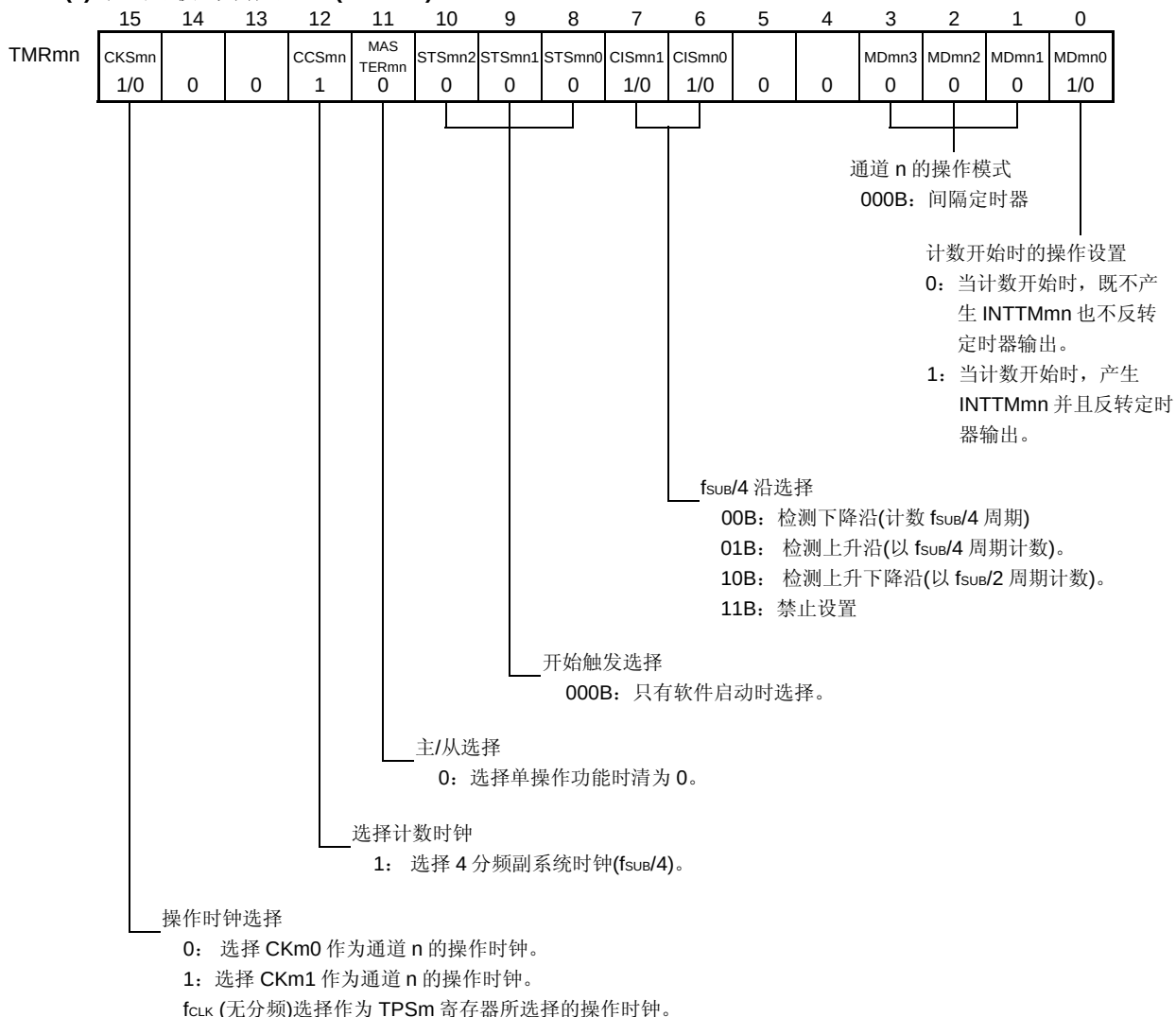


备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

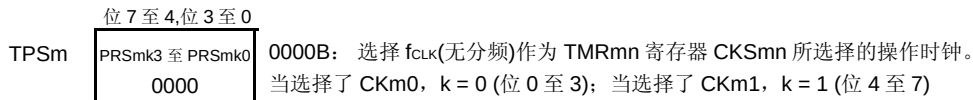
图 6-39. 作为间隔定时器/方波输出操作期间的寄存器设置内容的示例(2/3)

(2) 当选择 $f_{SUB}/4$ 作为计数时钟

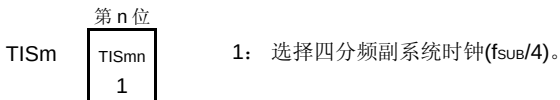
(a) 定时器模式寄存器 mn (TMRmn)



(b) 定时器时钟选择寄存器 m0(TPSm)



(c) 定时器输入选择寄存器 m(TISm)



(d) 定时器输出寄存器 m (TOM)

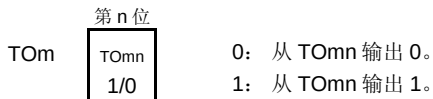


图 6-39. 作为间隔定时器/方波输出操作期间的寄存器设置内容的示例(3/3)

(2) 当选择 f_{SUB}/4 作为计数时钟(连续)

(e) 定时器输出使能寄存器 m (TOEm)

第 n 位					
TOEm	<table><tr><td>TOEmn</td><td>0: 通过计数操作, 停止 TOMn 输出操作。</td></tr><tr><td>1/0</td><td>1: 通过计数操作, 使能 TOMn 输出。</td></tr></table>	TOEmn	0: 通过计数操作, 停止 TOMn 输出操作。	1/0	1: 通过计数操作, 使能 TOMn 输出。
TOEmn	0: 通过计数操作, 停止 TOMn 输出操作。				
1/0	1: 通过计数操作, 使能 TOMn 输出。				

(f) 定时器输出电平寄存器 m (TOLm)

第 n 位					
TOLm	<table><tr><td>TOLmn</td><td>0: 当 TOMmn = 0 时 (反转模式) 清零</td></tr><tr><td>0</td><td></td></tr></table>	TOLmn	0: 当 TOMmn = 0 时 (反转模式) 清零	0	
TOLmn	0: 当 TOMmn = 0 时 (反转模式) 清零				
0					

(g) 定时器输出模式寄存器 m (TOMm)

第 n 位					
TOMm	<table><tr><td>TOMmn</td><td>0: 设置反转模式。</td></tr><tr><td>0</td><td></td></tr></table>	TOMmn	0: 设置反转模式。	0	
TOMmn	0: 设置反转模式。				
0					

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-40. 间隔定时器/方波输出功能的操作过程 (1/2)

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置PER0寄存器的TAU0EN位, TAU1EN位为1。 —————→	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器使能写入。)
	设置TPSm寄存器。 决定CKm0和CKm1的时钟频率。	
通道默认设置	设置TMRmn寄存器(决定通道的操作模式)。 当选择fsub/4作为计数时钟时, 设置TISmn位为1(fsub/4)。 设置间隔(周期)值到TDRmn寄存器。	通道停止操作。 (供给时钟并且产生一些功耗。)
	使用 TOMn 输出时 把 TOMm 寄存器的 TOMmn 位清零 (反转模式)。 把 TOLmn 位清零。 设置 TOMn 位, 并确定 TOMn 默认输出电平。 —————→	TOMn引脚为高阻输出状态。 当端口模式寄存器在输出模式下并且端口寄存器为0时, TOMn 默认设置电平为输出。
	设置 TOEmn 至 1 并使能 TOMn 操作。 —————→	因为通道停止运行, 所以TOMn 不会改变。
	将端口寄存器和端口模式寄存器清为 0。 —————→	TOMn引脚输出TOMn设置电平。
操作开始	设置TOEmn为1(只当操作恢复时)。 设置TSMn位为1。 —————→ 由于是触发位, TSMn 位自动恢复为0。	TEmn=1, 并且计数操作开始。 在计数时钟输入时TDRmn的值加载到TCRmn。如果 TMRmn寄存器的MDmn0位为1时产生INTTMmn, 并且 TOMn执行反转操作。
	操作期间	TMRmn, TOMm和TOLm寄存器的设定值不能改变。 TDRmn寄存器的设置值不能改变。 TCRmn寄存器总是可读。 不使用TSRmn寄存器。 TOM和TOEm寄存器的设置值可以改变。
	操作停止	TTmn 位设置为1。 —————→ 由于是触发位, TTmn 位自动恢复为0。 TEmn=0, 并且计数操作停止。 TCRmn保持计数值并停止。 TOMn输出不初始化而保持当前状态。
	TOEmn清除为0并且值设置到TOMn位。 —————→	TOMn引脚输出TOMn设置电平。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-40. 间隔定时器I方波输出功能的操作过程 (2/2)

	软件操作	硬件状态
TAU 停止	保持TOMn 引脚输出电平 在需要保持的值设置到端口寄存器后 清TOMn 位为0。 —————>	TOMn引脚输出电平由端口功能保持。
	不需要保持TOMn 引脚输出电平时， 将端口模式寄存器切换到输入模式。 —————>	TOMn引脚输出电平变为高阻输出状态。
	将PER0寄存器的TAU0EN位，TAU1EN位清零。 —————>	掉电状态 所有电路初始化，并且每个通道的SFR也初始化。 (TOMn位清为0，并且TOMn引脚设置为端口模式。)

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
 mn = 00 至 07, 10 至 13

6.7.2 外部事件计数器操作

定时器阵列单元可用作外部事件计数，对 TImn 引脚检测到的有效输入沿 (外部事件) 的数量进行计数。当达到指定计数值时，事件计数器产生一个中断。指定的计数号可通过下面的表达式计算得出。

指定的计数值 = TDRmn 的设置值 + 1

在事件计数模式下 TCRmn 作为向下计数器操作。
通道启动触发位 (TSmn) 设置为 1 时，TCRmn 加载 TDRmn 的值。
每次在 TImn 引脚检测到有效输入沿时，TCRmn 递减一次。当 TCRmn = 0000H，TCRmn 再次加载 TDRmn 的值，并输出 INTTMmn。
之后，重复上面的操作。
TOMn 不能使用，因为它的波形由外部事件决定，并且不规则。
TDRmn 在任何时间可以重写。在下一个计数周期 TDRmn 的新值有效。

图 6-41. 作为外部事件计数器操作的框图

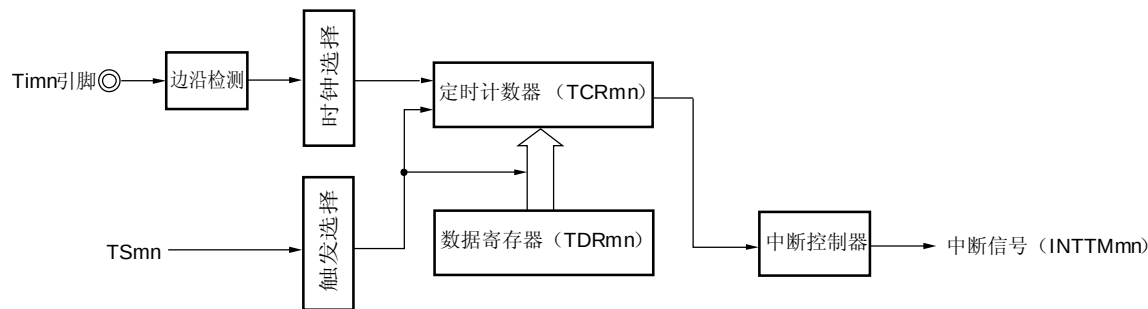
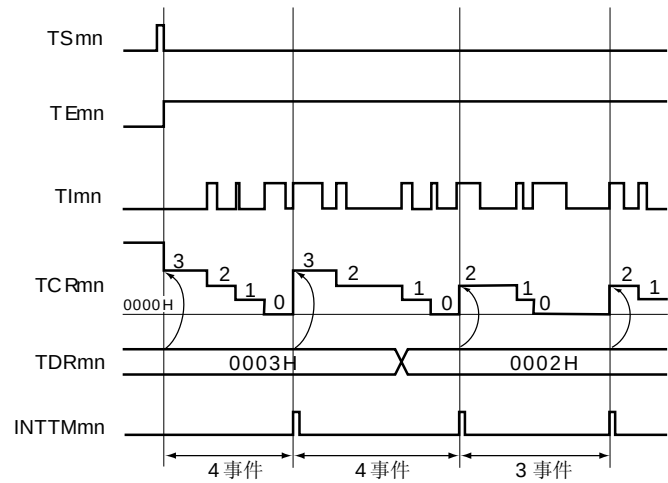
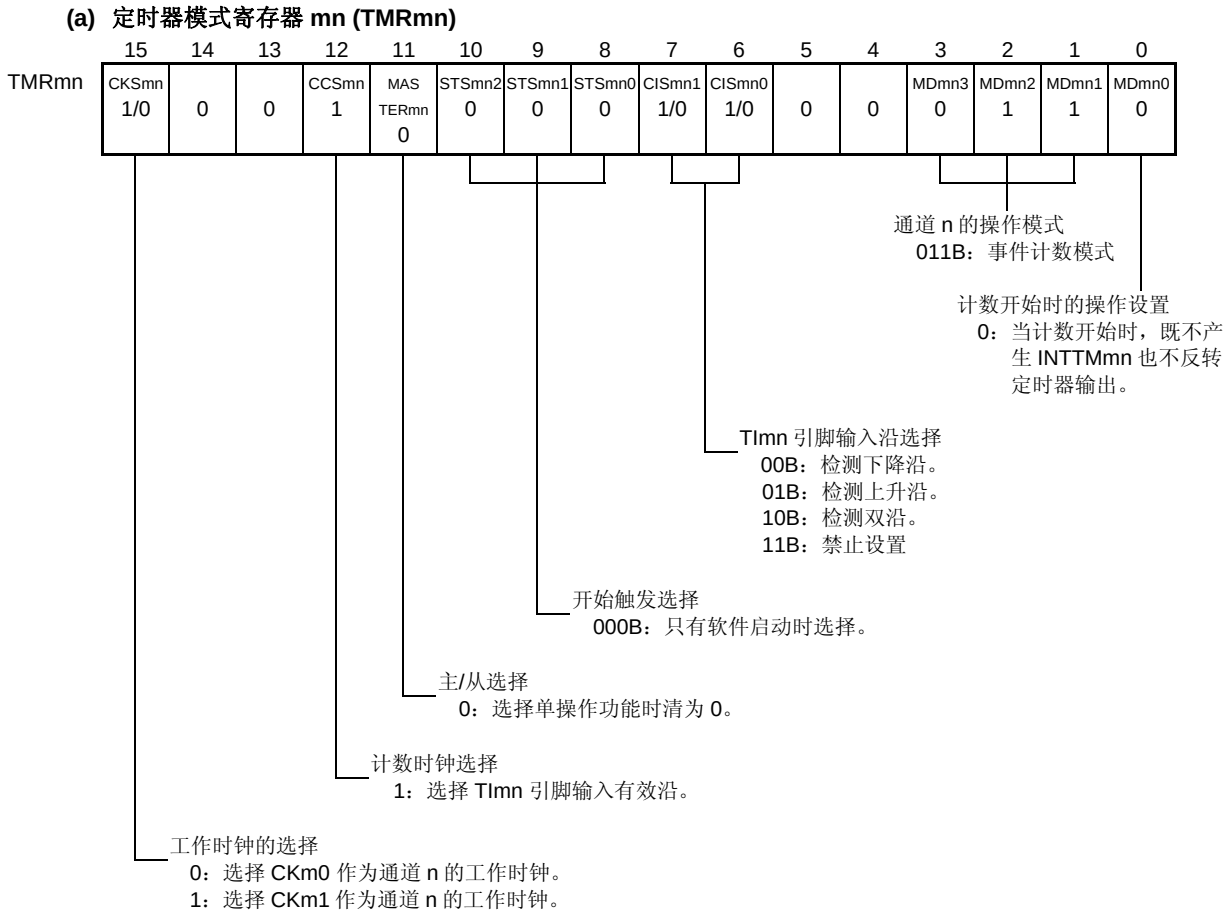


图 6-42. 作为外部事件计数器操作的基本时序示例



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-43. 在外部事件计数器模式下寄存器设置内容的示例



(b) 定时器输出寄存器 m (TOM)

第 n 位

TOM TOMn
0 0: 从 TOMn 输出 0。

(c) 定时器输出使能寄存器 m (TOEm)

第 n 位

TOEm TOEmn
0 0: 通过计数操作, 停止 TOMn 输出操作。

(d) 定时器输出电平寄存器 m (TOLm)

第 n 位

TOLm TOLmn
0 0: 当 TOMmn = 0 时 (反转模式) 清零。

(e) 定时器输出模式寄存器 m (TOMm)

第 n 位

TOMm TOMmn
0 0: 设置反转模式。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-44. 使用外部事件计数器功能时的操作过程

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置PER0寄存器的TAU0EN位, TAU1EN位为1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器使能写入。)
	设置TPSm寄存器。 决定CKm0和CKm1的时钟频率。	
通道默认设置	设置TMRmn寄存器(决定通道的操作模式)。 设置计数值到TDRmn寄存器。 把TOEm寄存器的TOEmn位清零。	通道停止操作。 (供给时钟并且产生一些功耗。)
操作开始	设置TSMn位为1。 由于是触发位, TSMn位自动恢复为0。	TEmn=1,并且计数操作开始。 TDRmn的值加载到TCRmn并且等待TImn引脚输入沿的检测。
操作期间	TDRmn寄存器的设置值不能改变。 TCRmn寄存器总是可读。 不使用TSRmn寄存器。 TMRmn, TOMm, TOLm, TOm和TOEm寄存器的设定值不能改变。	计数器 (TCRmn) 在每次检测到TImn引脚输入沿时向下计数。当计数值达到0000H时, TDRmn的值再次加载到TCRmn, 并且计数操作继续。通过检测TCRmn = 0000H, 产生INTTMmn输出。之后, 重复上面的操作。
操作停止	TTmn 位设置为1。 由于是触发位, TTmn 位自动恢复为0。	TEmn=0, 并且计数操作停止。 TCRmn保持计数值并停止。
TAU 停止	将PER0寄存器的TAU0EN位, TAU1EN位清零。	掉电状态 所有电路初始化, 并且每个通道的SFR也初始化。

操作恢复

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

6.7.3 分频器操作(仅单元 0 的通道 0)

定时器阵列单元可用作分频器，把输入到 TImn 引脚和从 TOnn 输出结果的时钟分频。
从 TOnn 输出的分频时钟频率可由下面的表达式计算。

- 当选择上升沿/下降沿时：
分频时钟频率=输入时钟频率/[(TDRmn 的设置值 + 1) × 2]
 - 当选择双沿时：
分频时钟频率≡ 输入时钟频率/(TDRmn 的设定值 + 1)

TCRmn 在间隔定时器模式下作为向下计数器操作。

通道启动触发位 (TSmn) 设置为 1 后，当检测到 TImn 有效沿时，TCRmn 加载 TDRmn 的值。如果此时 TMRmn 的 MDmn0 = 0，INTTMMn 不输出并且 TOnn 不反转。如果 TMRmn 的 MDmn0 = 1，输出 INTTMMn 并且 TOnn 反转。

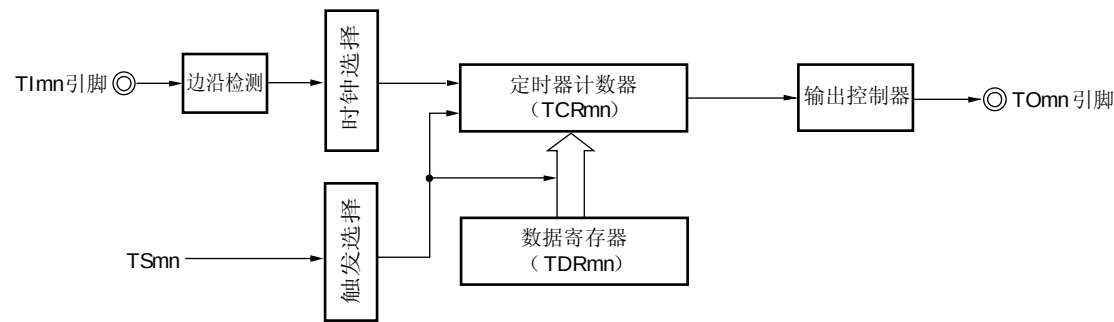
之后，TCRmn 在 TImn 有效沿向下计数。当 TCRmn = 0000H 时，触发 TOnn。同时，TCRmn 再次加载 TDRmn 的值并继续计数。

如果选择 TImn 的双沿检测，输入时钟的占空比误差将影响 TOnn 输出的分频时钟周期。
TOnn 输出时钟周期包括工作时钟的一个周期的一个采样误差。

TOnn 输出的时钟周期 = 理想的 TOnn 输出时钟周期± 工作时钟周期 (误差)

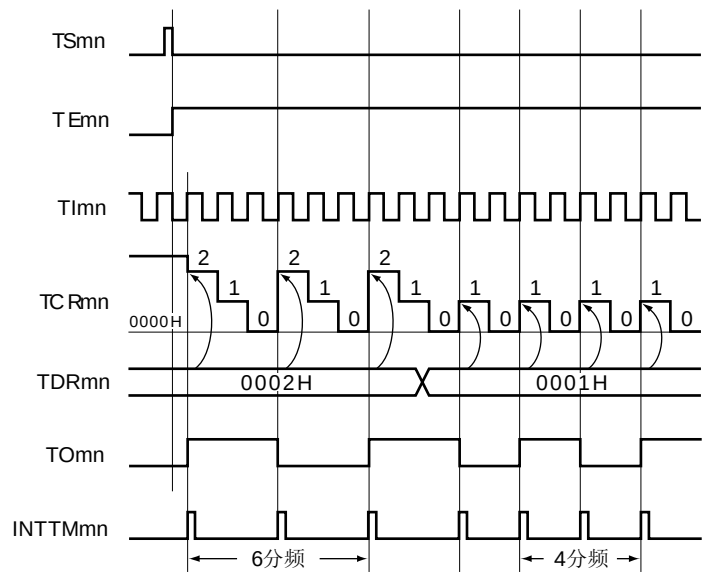
TDRmn 在任何时间可以重写。 在下一个计数周期 TDRmn 的新值有效。

图 6-45. 分频器操作的框图



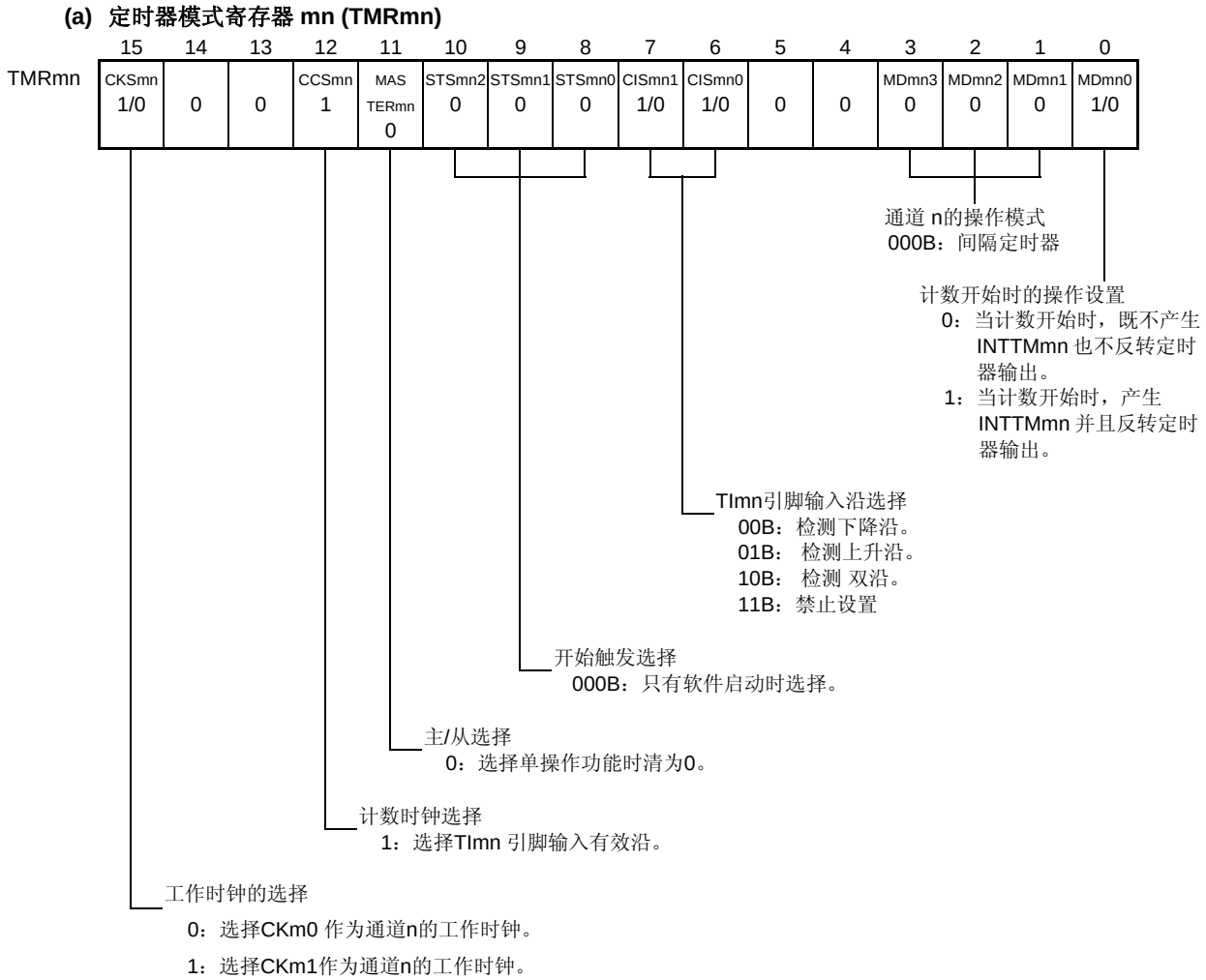
备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-46. 分频器 (MDmn0 = 1) 操作的基本时序示例



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-47. 使用分频器时寄存器设置内容的示例



(b) 定时器输出寄存器 m (TOM)

TOM	第 n 位	
	TOMn	0: 从 TOMn 输出 0。 1: 从 TOMn 输出 1。

(c) 定时器输出使能寄存器 m (TOEm)

TOEm	第 n 位	
	TOEmn	0: 通过计数操作, 停止 TOMn 输出操作。 1: 通过计数操作, 使能 TOMn 输出。

(d) 定时器输出电平寄存器 m (TOLm)

TOLm	第 n 位	
	TOLmn	0: 当 TOMmn = 0 时 (反转模式) 清零

(e) 定时器输出模式寄存器 m (TOMm)

TOMm	第 n 位	
	TOMmn	0: 设置反转模式。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-48. 使用分频功能的操作过程

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置PER0寄存器的TAU0EN位, TAU1EN位为1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器使能写入。)
	设置TPSm寄存器。 决定CKm0和CKm1的时钟频率。	
通道默认设置	设置TMRmn寄存器(决定通道的操作模式)。	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置间隔(周期)值到TDRmn寄存器。	
	将TOMmn寄存器的TOMmn位清除为0(反转模式)。 把TOLmn位清零。	TOMn引脚为高阻输出状态。
	设置TOMn 位, 并确定TOMn 默认输出电平。	当端口模式寄存器在输出模式下并且端口寄存器为0时, 输出TOMn默认设置电平。
	设置TOEmn至1并使能TOMn操作。 将端口寄存器和端口模式寄存器清为0。	因为通道停止工作, TOMn不会改变。 TOMn引脚输出TOMn设置电平。
操作开始	设置TOEmn为1(仅当操作恢复时)。 设置TSMn位为1。 由于是触发位, TSMn 位自动恢复为0。	TEmn=1, 并且计数操作开始。 在计数时钟输入时TDRmn的值加载到TCRmn。 如果TMRmn寄存器的MDmn0位为1时产生INTTMmn, 并且TOMn执行反转操作。
操作期间	TDRmn寄存器的设置值不能改变。 TCRmn寄存器总是可读。 不使用TSRmn寄存器。 TOM和TOEm寄存器的设置值可以改变。 TMRmn, TOMmn和TOLmn寄存器的设定值不能改变。	计数器(TCRmn)向下计数。当计数值达到0000H时, TDRmn的值再次加载到TCRmn, 并且计数操作继续。 通过检测TCRmn=0000H, INTTMmn产生并且TOMn执行反转操作。之后, 重复上面的操作。
操作停止	TTmn 位设置为1。 由于是触发位, TTmn 位自动恢复为0。	TEmn=0, 并且计数操作停止。 TCRmn保持计数值并停止。 TOMn输出不初始化而保持当前状态。
	TOEmn清除为0并且值设置为TOM寄存器。	TOMn引脚输出TOMn设置电平。
TAU 停止	保持TOMn 引脚输出电平 在需要保持的值设置到端口寄存器后 清TOMn 位为0。	TOMn引脚输出电平由端口功能保持。
	当不需要保持TOMn引脚输出电平时 切换端口模式寄存器到输入模式。	TOMn引脚输出电平变为高阻输出状态。
	将PER0寄存器的TAU0EN位, TAU1EN位清零。	掉电状态 所有电路初始化, 并且每个通道的SFR也初始化。 (TOMn位清为0, 并且TOMn引脚设置为端口模式)。

操作恢复

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

6.7.4 输入脉冲间隔测量的操作

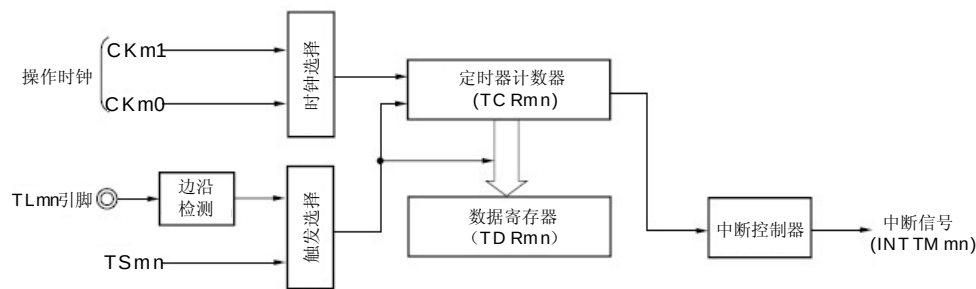
在 TImn 有效沿可以捕捉计数值，可以测量输入到 TImn 的脉冲间隔。
脉冲间隔可通过下面的公式计算。

$$\text{TImn 输入脉冲间隔} = \text{计数时钟周期} \times ((1000\text{H} \times \text{TSRmn: OVF}) + (\text{TDRmn 的捕捉值} + 1))$$

注意事项 对 TImn 引脚输入进行采样的工作时钟是由 TMRmn 寄存器的 CKSmn 位选择，因此会产生和工作时钟一致的误差。

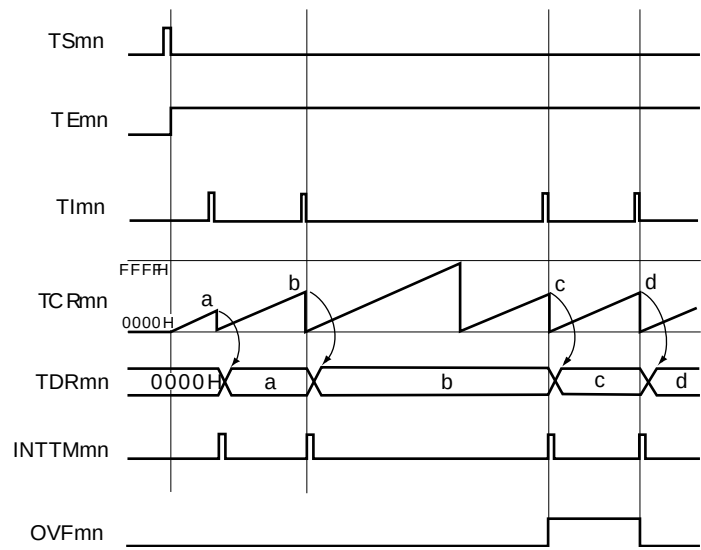
TCRmn 在捕捉模式下作为向上计数器操作。
当通道启动触发器 (TSmn) 设置为 1 时，TCRmn 从 0000H 开始与计数时钟同步向上计数。
当 TImn 引脚检测到输入的有效沿时，计数值传输(捕捉) 到 TDRmn，同时计数器 (TCRmn) 清除为 0000H，并且产生 INTTMmn。如果计数器在此时溢出，TSRmn 寄存器的 OVF 位设置为 1。如果计数器没有溢出，OVF 位清零。之后，重复上面的操作。
计数值一旦捕捉到 TDRmn 寄存器，TSRmn 寄存器的 OVF 位就根据测量期间计数器是否溢出而更新。因此，可以检测到捕捉值的溢出状态。
如果计数器在两个以上周期达到满计数，则判断为发生一个溢出，TSRmn 寄存器的 OVF 位设置为 1。因此 OVF 位作为累计标志，如果溢出发生一次以上，就不能测量到正确的间隔值。
设置 TMRmn 寄存器的 STSmn2 至 STSmn0 位为 001B，将 TImn 的有效沿作为启动触发和捕捉触发。
当 TEMn = 1，软件操作 (TSmn = 1) 可作为捕捉触发，而不是 TImn 引脚输入。

图 6-49. 作为输入脉冲间隔测量操作的框图



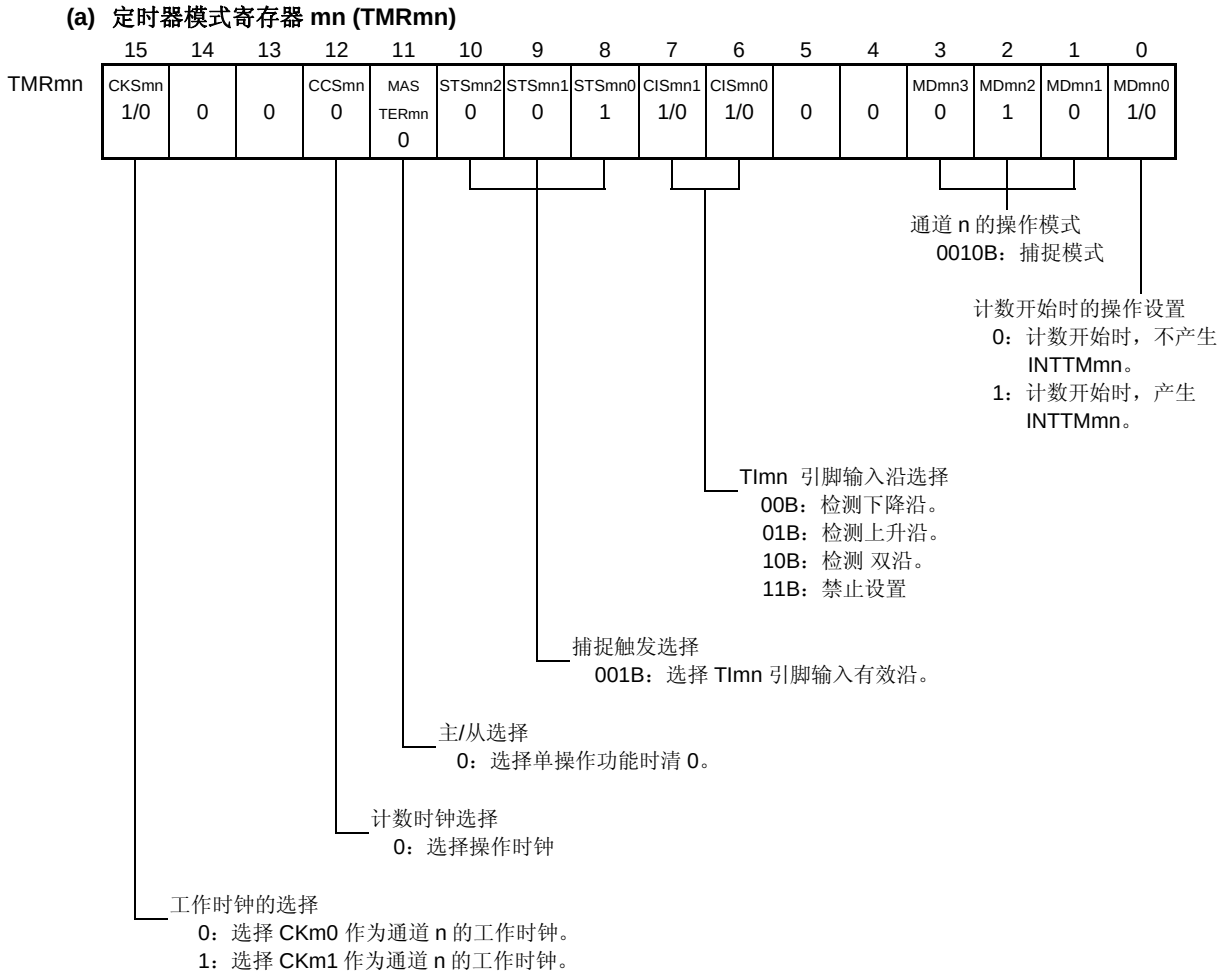
备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-50. 输入脉冲间隔测量 (MDmn0 = 0) 操作的基本时序示例



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-51. 测量输入脉冲间隔的寄存器设置内容的示例



(b) 定时器输出寄存器 m (TOM)

第 n 位

TOMn

0: 从 TOMn 输出 0。

(c) 定时器输出使能寄存器 m (TOEm)

第 n 位

TOEmn

0: 通过计数操作, 停止 TOMn 输出操作。

(d) 定时器输出电平寄存器 m (TOLm)

第 n 位

TOLmn

0: 当 TOMmn = 0 时 (反转模式) 清零。

(e) 定时器输出模式寄存器 m (TOMm)

第 n 位

TOMmn

0: 设置反转模式。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-52. 输入脉冲间隔测量功能时的操作过程



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

6.7.5 用作输入信号高/低电平宽度测量的操作

在 TImn 的一个沿开始计数，并且在另一个沿捕捉计数的数量，可以测量 TImn 的信号宽度（高电平宽度/低电平宽度）。TImn 的信号宽度可以由下面的公式计算出。

$$\text{TImn 输入的信号宽度} = \text{计数时钟周期} \times ((10000\text{H} \times \text{TSRn: OVF}) + (\text{TDRmn 的捕捉值} + 1))$$

注意事项 对 TImn 引脚输入进行采样的工作时钟是由 TMRmn 寄存器的 CKSmn 位选择，因此会产生一个和工作时钟相一致的误差。

TCRmn 在捕捉&一次计数模式下作为向上计数器。

当通道启动触发器 (TSmn) 设置为 1，TEmn 设置为 1 并且设置 TImn 引脚为开始沿检测等待状态。

当检测到 TImn 开始有效沿（当测量高电平宽度时为 TImn 的上升沿），计数器与计数时钟同步向上计数。之后当检测到有效捕捉沿（当测量高电平宽度时 TImn 的下降沿），计数值传输到 TDRmn，同时，产生 INTTMmn。如果计数器在此时溢出，TSRmn 寄存器的 OVF 位设置为 1。如果计数器没有溢出，OVF 位清零。TCRmn 在“传输至 TDRmn +1”处停止，并且设置 TImn 引脚起始沿检测等待状态。之后，重复上面的操作。

计数值一旦捕捉到 TDRmn 寄存器，TSRmn 寄存器的 OVF 位就根据测量期间计数器是否溢出而更新。因此，可以检测到捕捉值的溢出状态。

如果计数器在两个以上周期达到满计数，判断为一个溢出发生，TSRmn 寄存器的 OVF 位设置为 1。因此 OVF 位作为累计标志，如果溢出发生一次以上，不能测量到正确的间隔值。

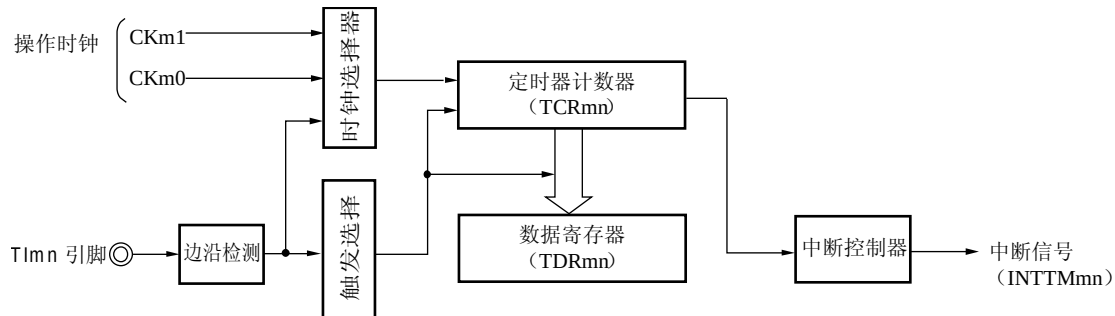
通过 TMRmn 寄存器的 CISmn1 和 CISmn0 位可以选择测量 TImn 引脚的高电平宽度或低电平宽度。

因为该功能用于测量 TImn 引脚输入的信号宽度，当 TE mn 为 1 时 TSmn 不能设置为 1。

TMRmn 的 CISmn1、CISmn0 = 10B：测量低电平宽度。

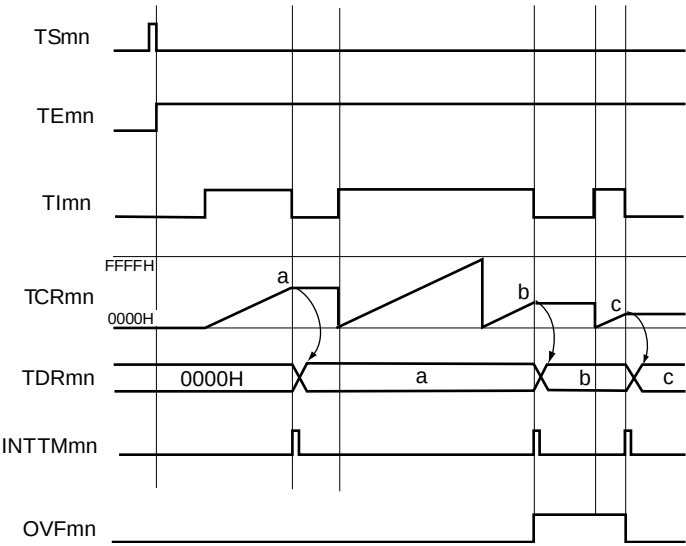
TMRmn 的 CISmn1、CISmn0 = 11B：测量高电平宽度。

图 6-53. 作为输入信号高/低电平宽度测量操作的框图



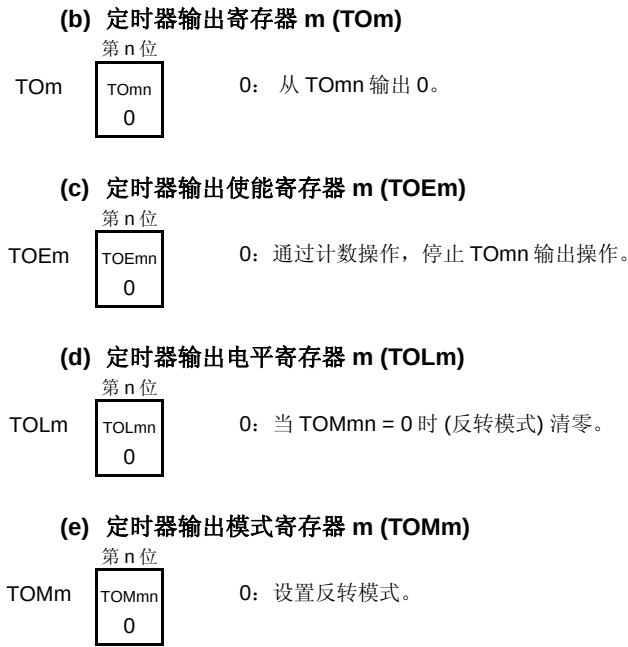
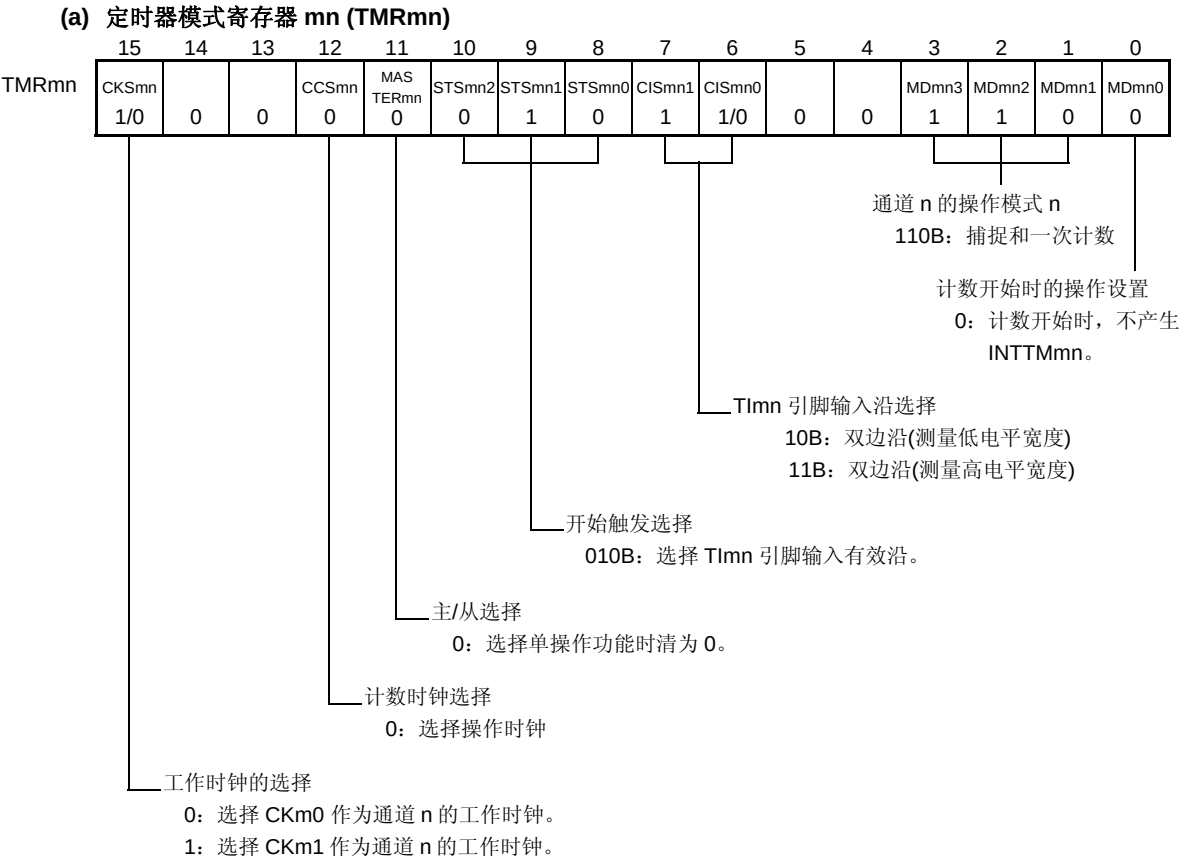
备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-54. 作为输入信号高/ 低电平宽度测量操作的基本时序示例



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-55. 测量输入信号高/ 低电平宽度的寄存器设置内容的示例



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

图 6-56. 使用输入信号高/低电平宽度测量功能的操作过程

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置PER0寄存器的TAU0EN位, TAU1EN位为1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器使能写入。)
	设置TPSm寄存器。 决定CKm0和CKm1的时钟频率。	
通道默认设置	设置TMRmn寄存器(决定通道的操作模式)。 把TOEmn清零并停止TOMn的运行。	通道停止操作。 (供给时钟并且产生一些功耗。)
操作开始	设置TSMn位为1。 由于是触发位, TSMn 位自动恢复为0。	TEmn = 1, 设置TIMn引脚开始边沿检测等待状态。
	检测TIMn引脚输入计数开始有效沿。	TCRmn清除为0000H并且开始向上计数。
操作期间	TDRmn寄存器的设置值不能改变。 TCRmn寄存器总是可读。 不使用TSRmn寄存器。 TMRmn, TOMm, TOLm, TOM和TOEm寄存器的设定值不能改变。	当TIMn引脚启动沿检测到时, 计数器 (TCRmn) 从0000H开始向上计数。如果TIMn引脚的捕捉沿检测到, 计数值就会传输到TDRmn并且产生INTTMmn。 如果计数器在此时溢出, TSRmn寄存器的OVF位设置为1。如果计数器没有溢出, OVF位清零。TCRmn停止计数操作直到下一个TIMn引脚启动沿检测到。
操作停止	TTmn 位设置为1。 由于是触发位, TTmn 位自动恢复为0。	TEmn=0, 并且计数操作停止。 TCRmn保持计数值并停止。 TSRmn寄存器的OVF位也保持。
TAU 停止	将PER0寄存器的TAU0EN位, TAU1EN位清零。	掉电状态 所有电路初始化, 并且每个通道的SFR也初始化。

操作恢复

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7),
mn = 00 至 07, 10 至 13

6.8 定时器阵列单元的多通道操作

6.8.1 作为 PWM 功能操作

两个通道可用于设置产生任意周期和占空比的脉冲。
输出脉冲的周期和占空比可通过下面的公式计算出。

脉冲周期 = { TDRmn (主通道) 的设置值 + 1} × 计数时钟周期

占空比 [%] = { TDRmp (从设备) 的设置值}/{ TDRmn (主设备) 的设置值 + 1} × 100

0% 输出: TDRmp(从设备)的设置值=0000H

100%输出: TDRmp (从设备) 的设置值≥{ TDRmn (主设备) 的设置值+ 1}

备注 如果 TDRmp(从设备) 的设置值 > (TDRmn 的设置值 (主设备) + 1)，虽然占空比会超出 100%。但是作为 100%输出。

主通道操作在间隔定时器模式并计数周期。当通道启动触发器 (TSmn) 设置为 1 时， INTTMmn 输出。TCRmn 从 TDRmn 加载的值开始与计数时钟同步向下计数。当 TCRmn=0000H 时，将输出 INTTMmn。TCRmn 再次加载 TDRmn 的值。之后，重复类似的操作。

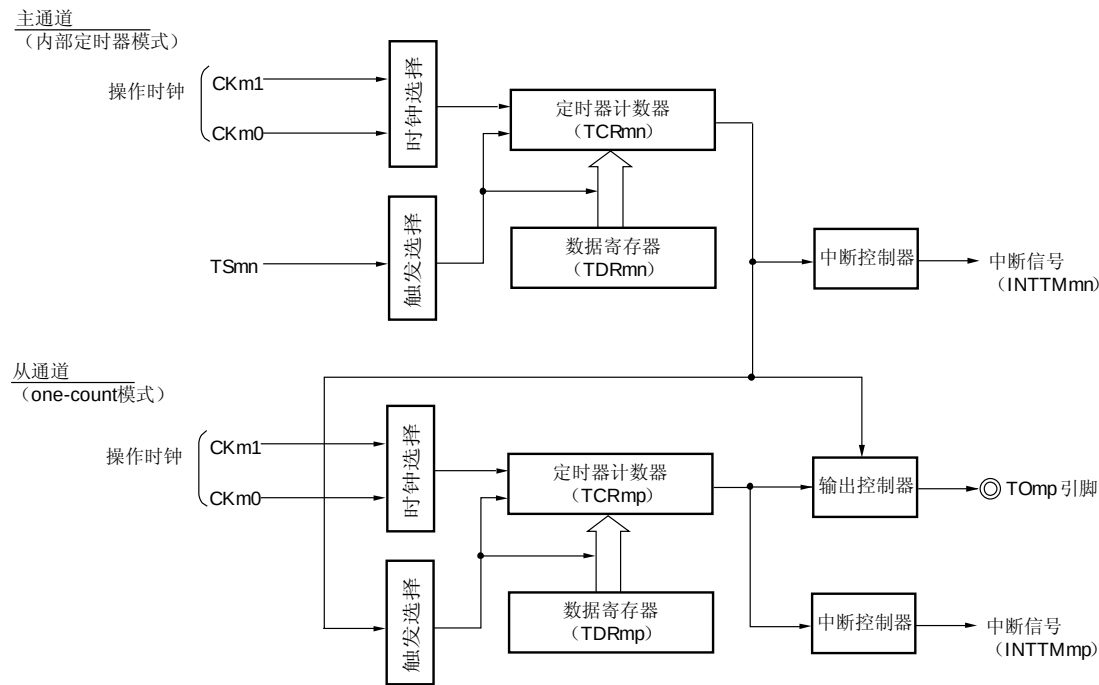
从通道的 TCRmp 运行于一次计数模式，计数占空比，并且从 TOmp 引脚输出 PWM 波形。通过使用主通道的 INTTMmn 作为启动触发，从通道的 TCRmp 加载 TDRmp 值，并且 TCRmp 从 TDRmp 的值开始向下计数。当 TCRmp = 0000H 时，TCRmp 停止计数直到下一个启动触发(主通道的 INTTMmn)开始输入。

TOmp 的输出电平在主通道产生 INTTMmn 后的一个计数时钟后开始有效，TCRmp = 0000H 时变为无效。

注意事项 要重写主通道的 TDRmn 和从通道的 TDRmp，需要两次写入访问。在主通道的 INTTMmn 发生时，TDRmn 和 TDRmp 的值加载到 TCRmn 和 TCRmp。因此，当两次重写分别在主通道的 INTTMmn 发生之前和之后执行时，TOmp 引脚不能输出预期的波形。要重写主通道的 TDRmn 和从通道的 TDRmp，必须在主通道的 INTTMmn 产生后立即重写这两个寄存器。

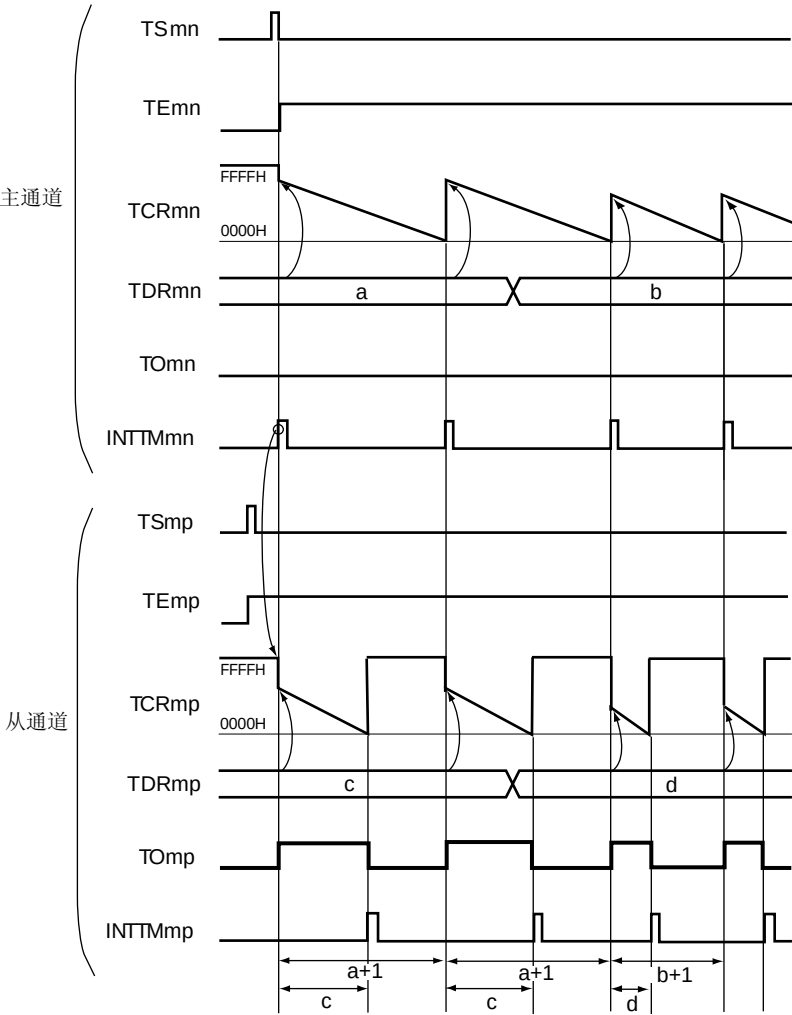
备注 m: 单元编号, n: 通道编号, p: 从通道编号(p = n+1)
当 m = 0: n = 0, 2, 4, 6
当 m = 1: n = 0, 2

图 6-57. 作为 PWM 功能操作的框图



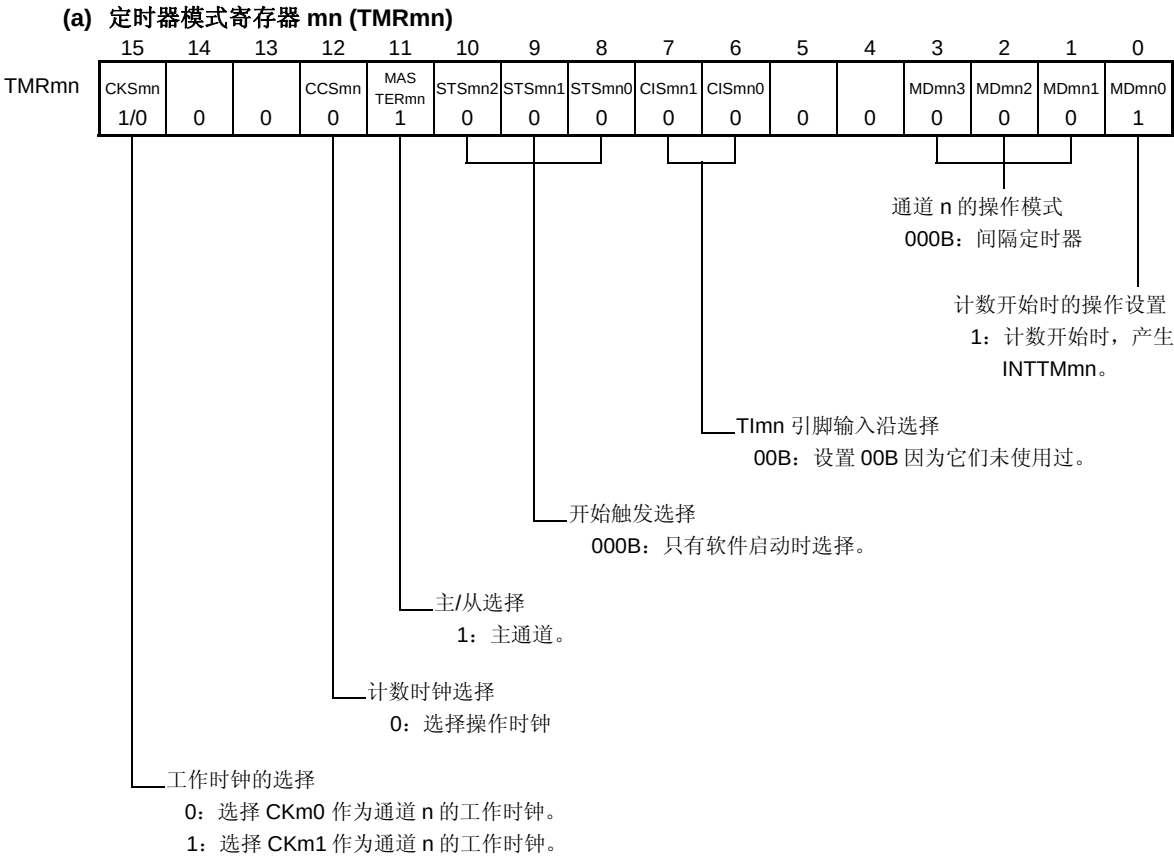
备注 m: 单元编号, n: 通道编号, p: 从通道编号(p = n+1)
当 m = 0: n = 0, 2, 4, 6
当 m = 1: n = 0, 2

图 6-58. 作为 PWM 功能操作的基本时序示例

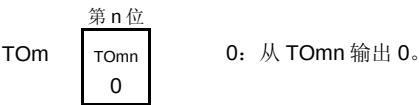


备注 m: 单元编号, n: 通道编号, p: 从通道编号(p = n+1)
当 m = 0: n = 0, 2, 4, 6
当 m = 1: n = 0, 2

图 6-59. PWM 功能(主通道)寄存器设置内容的示例



(b) 定时器输出寄存器 m (TOM)



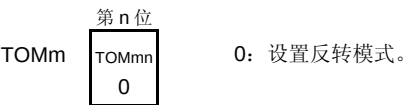
(c) 定时器输出使能寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)

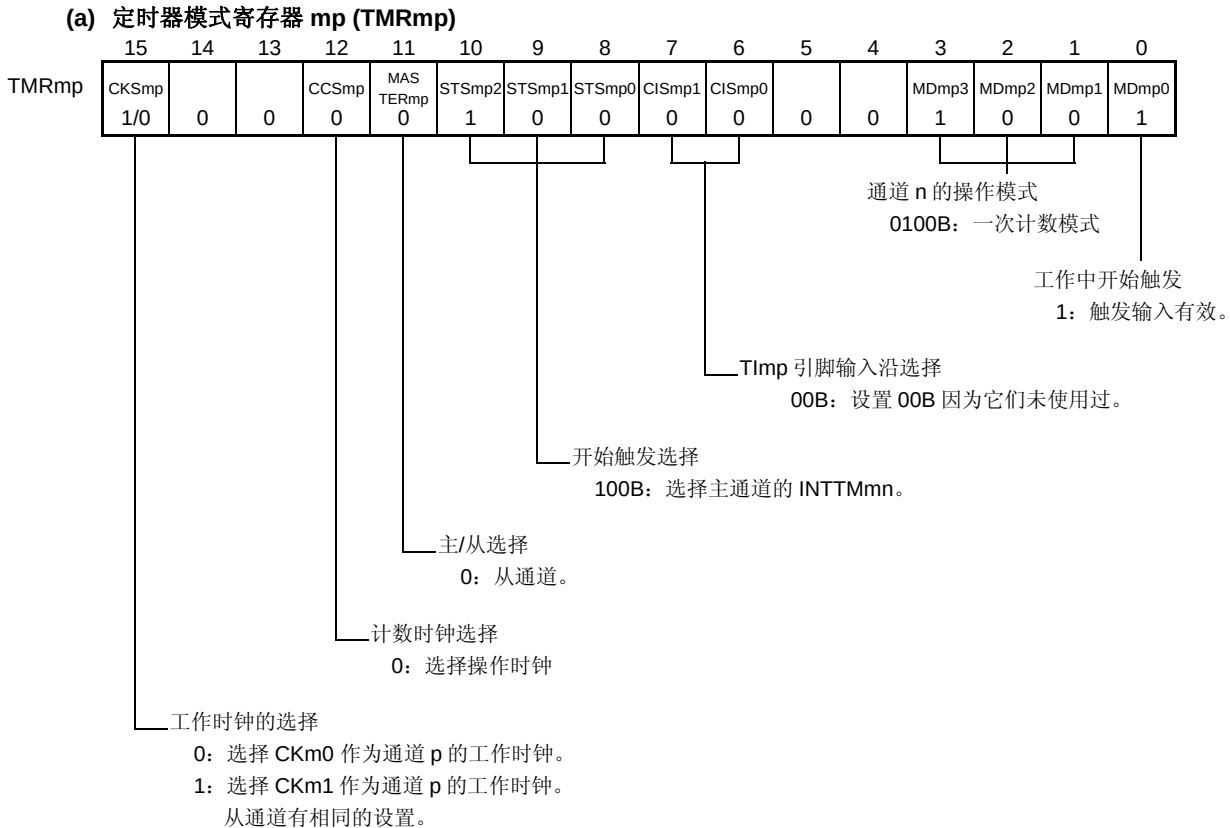


(e) 定时器输出模式寄存器 m (TOMm)



备注 m: 单元编号, n: 通道编号
当 m = 0: n = 0, 2, 4, 6
当 m = 1: n = 0, 2

图 6-60. PWM 功能(从通道)使用寄存器设置的示例



(b) 定时器输出寄存器 m (TOm)

TOm	第 p 位	
	TOmp	0: 从 TOmp 输出 0。 1: 从 TOmp 输出 1。

(c) 定时器输出使能寄存器 m (TOEm)

TOEm	第 p 位	
	TOEmp	0: 通过计数操作, 停止 TOmp 输出操作。 1: 通过计数操作, 使能 TOmp 输出。

(d) 定时器输出电平寄存器 m (TOLm)

TOLm	第 p 位	
	TOLmp	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)

(e) 定时器输出模式寄存器 m (TOMm)

TOMm	第 p 位	
	TOMmp	1: 设置组合操作模式。

备注 m: 单元编号, n: 通道编号, p: 从通道编号(p = n+1)
当 m = 0: n = 0, 2, 4, 6
当 m = 1: n = 0, 2

图 6-61. 使用 PWM 功能时的操作过程 (1/2)

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置PER0寄存器的TAU0EN位, TAU1EN位为1。 —————▶	上电状态。 每个通道停止操作。 (开始供给时钟并且每个寄存器使能写入。)
	设置TPSm寄存器。 决定CKm0和CKm1的时钟频率。	
通道默认设置	设置要使用的两个通道的TMRmn和TMRmp寄存器(决定通道的操作模式)。 间隔(周期)值设置到主通道的TDRmn寄存器, 占空比设置到从通道的TDRmp寄存器。	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置从通道。 将TOMm寄存器的TOMmp位设置为1 (组合操作模式)。 设置TOLmp位。 设置TOMp位并决定TOMp输出的默认电平。 —————▶ 设置TOEmp至1并使能TOMp操作。 —————▶ 将端口寄存器和端口模式寄存器清为0。 —————▶	TOmn引脚为高阻输出状态。 当端口模式寄存器在输出模式下并且端口寄存器为0时, 输出TOMp默认设置电平。 因为通道停止操作, TOMp 不会改变。 TOMp引脚输出TOMp的设置电平。

备注 m: 单元编号, n: 主通道编号, p: 从通道编号(p = n+1)

当 m = 0: n = 0, 2, 4, 6

当 m = 1: n = 0, 2

	软件操作	硬件状态
操作开始	设置TOEmp(从通道) 为1 (仅当操作恢复时)。 TSmn寄存器的TSmn(主通道) 和TSmp(从通道) 位同时设置为1。 因为是触发位, TSmn和TSmp位自动返回为0。	TEmn = 1, TEm = 1 当主通道开始计数, 产生INTTMmn。由此中断触发的从通道也开始计数。
操作期间	TMRmn和TMRmp寄存器的TOMmn, TOMmp, TOLmn, TOLmp位的设置值不能改变。 在主通道的INTTMmn产生后, TDRmn和TDRmp寄存器的设置值可更改。 TCRmn和TCRmp寄存器总是可读。 不使用TSRmn和TSRmp寄存器。 TOM和TOEm寄存器的设置值不能改变。	主通道的计数器加载TDRmn的值到TCRmn, 并向下计数。当计数值达到TCRmn = 0000H, 产生INTTMmn输出。同时, TDRmn寄存器的值加载到TCRmn, 计数器开始再次向下计数。 在从通道, 加载TDRmp的值到TCRmp, 由主通道的INTTMmn触发并且计数器开始向下计数。Tomp的输出电平在主通道输出INTTMmn后的一个计数时钟后开始有效。当TCRmp=0000H时变为无效, 并且计数操作停止。之后, 重复上面的操作。
操作停止	TTmn(主通道)和TTmp(从通道)位同时设置为1。 因为是触发位, TTmn和TTmp位自动返回为0。	TEmn, TEm = 0, 并且计数操作停止。 TCRmn和TCRmp保持计数值并停止。 Tomp输出不初始化而保持当前状态。
	从通道的TOEmp清除为0, 值设置到Tomp位。	Tomp引脚输出Tomp的设置电平。
TAU 停止	需要保持Tomp 引脚输出电平时 在需要保持的值设置到端口寄存器后 清Tomp 位为0。 不需要保持Tomp引脚输出电平时 切换端口模式寄存器到输入模式。 将PER0寄存器的TAU0EN位, TAU1EN位清零。	Tomp引脚输出电平由端口功能保持。 Tomp引脚输出电平变为 高阻输出状态。 掉电状态 所有电路初始化, 并且每个通道的SFR也初始化。 (Tomp位清为0, 并且Tomp引脚设置为端口模式。)

266 用户手册初稿 U19460CA1V0UD

6.8.2 作为单脉冲输出功能操作

将两个通道作为一组，通过输入到 TImn 的引脚信号可以产生任意延迟脉冲宽度的单脉冲。
延迟时间和脉冲宽度可通过下面的公式计算出。

$$\begin{aligned} \text{延迟时间} &= \{\text{TDRmn (主通道) 的设置值} + 2\} \times \text{计数时钟周期} \\ \text{脉冲宽度} &= \{\text{TDRmp (从通道) 的设置值}\} \times \text{计数时钟周期} \end{aligned}$$

主通道操作在单计数模式下，对延迟计数。主通道的 TCRmn 在检测到启动触发时开始操作，TDRmn 的值赋给 TCRmn。TCRmn 从 TDRmn 的加载值开始与计数时钟同步向下计数。当 TCRmn = 0000H 时，输出 INTTMmn 并且停止计数直到检测到下一个启动触发。

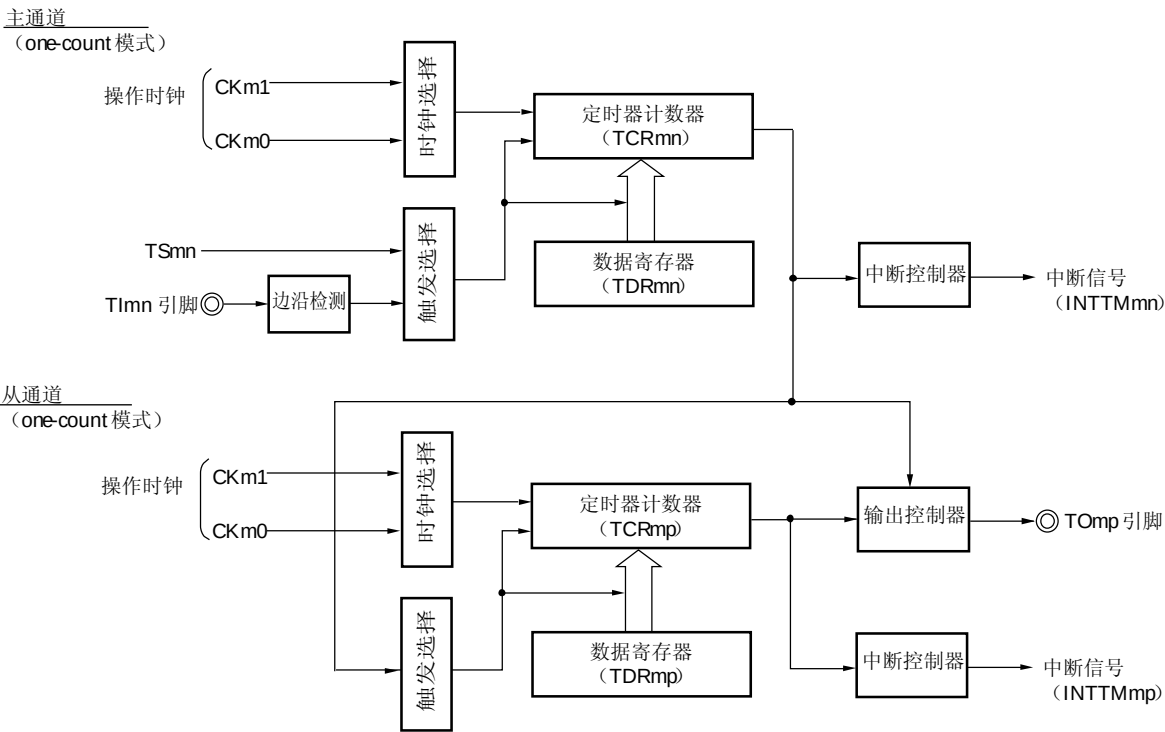
从通道操作在一次计数模式下，对脉冲宽度计数。通过主通道的 INTTMmn 作为启动触发，从通道的 TCRmp 开始计数，并加载 TDRmp 值。TCRmp 与计数值同步，由加载的 TDRmp 值开始向下计数。当 TCRmp = 0000H 时，输出 INTTMmp 并停止计数直到检测到下一个启动触发 (主通道的 INTTMmn)。TOMP 的输出电平在主通道产生 INTTMmn 后的一个计数时钟后开始有效，TCRmp = 0000H 时变为无效。

代替 TImn 引脚输入，单脉冲输出也可通过软件操作 (TSmn = 1)作为启动触发。

注意 主通道的 TDRmn 的时序与从通道的 TDRmp 的不同。如果在操作中 TDRmn 和 TDRmp 重写，则输出不规则波形。必须在将要重写通道的 INTTMmn 发生之后重写 TDRmn 和 TDRmp。

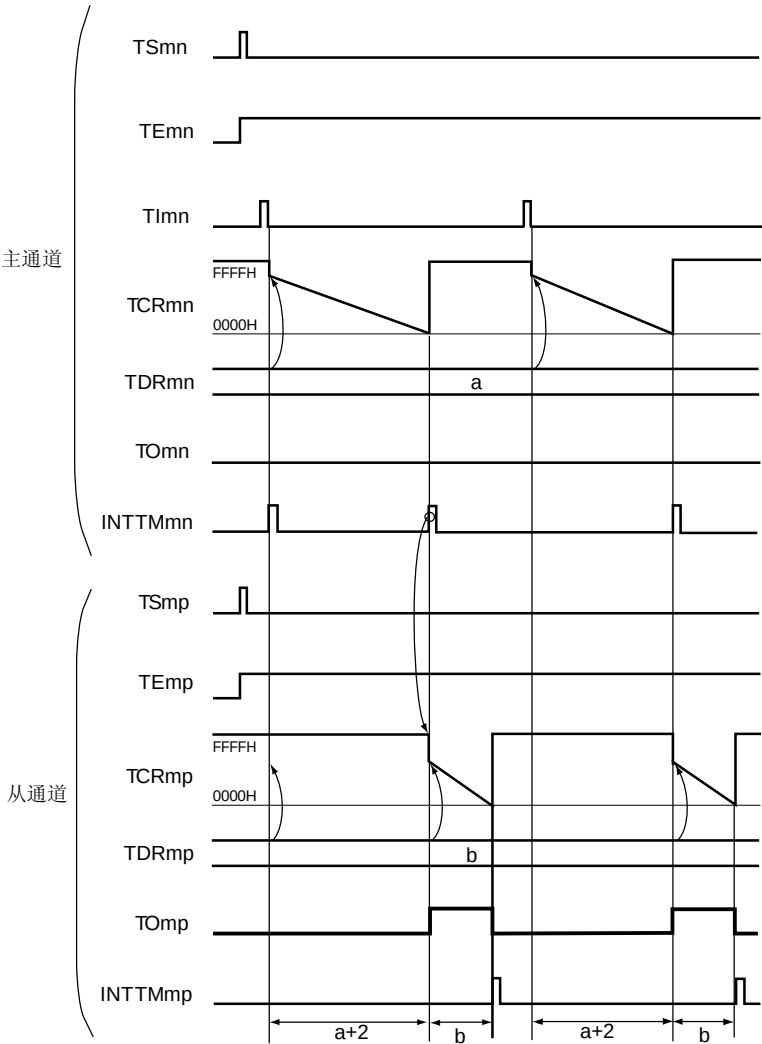
备注 m: 单元编号, n: 通道编号, p: 从通道编号(p = n+1)
当 m = 0: n = 0, 2, 4, 6
当 m = 1: n = 0, 2

图 6-62. 单脉冲输出功能操作的框图



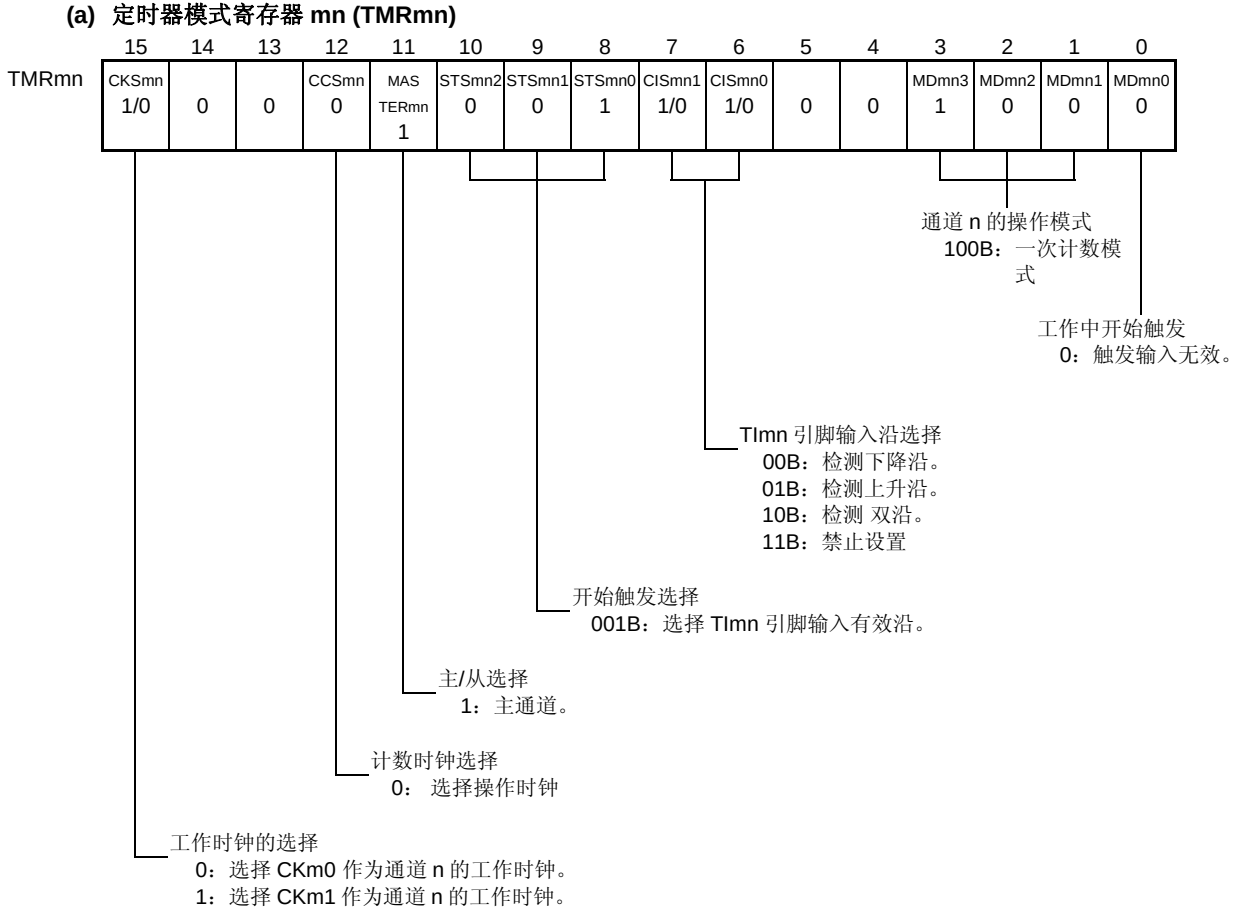
备注 m: 单元编号, n: 通道编号, p: 从通道编号(p = n+1)
当 m = 0: n = 0, 2, 4, 6
当 m = 1: n = 0, 2

图 6-63. 单脉冲输出功能操作的基本时序的示例



备注 m: 单元编号, n: 通道编号, p: 从通道编号($p = n+1$)
当 $m = 0$: $n = 0, 2, 4, 6$
当 $m = 1$: $n = 0, 2$

图 6-64. 单脉冲输出功能 (主通道) 寄存器设置的示例



(b) 定时器输出寄存器 m (TOM)

第 n 位

TOM 第 n 位: TOMn 0: 从 TOMn 输出 0。

(c) 定时器输出使能寄存器 m (TOEm)

第 n 位

TOEm 第 n 位: TOEmn 0: 通过计数操作, 停止 TOMn 输出操作。

(d) 定时器输出电平寄存器 m (TOLm)

第 n 位

TOLm 第 n 位: TOLmn 0: 当 TOMmn = 0 时 (反转模式) 清零。

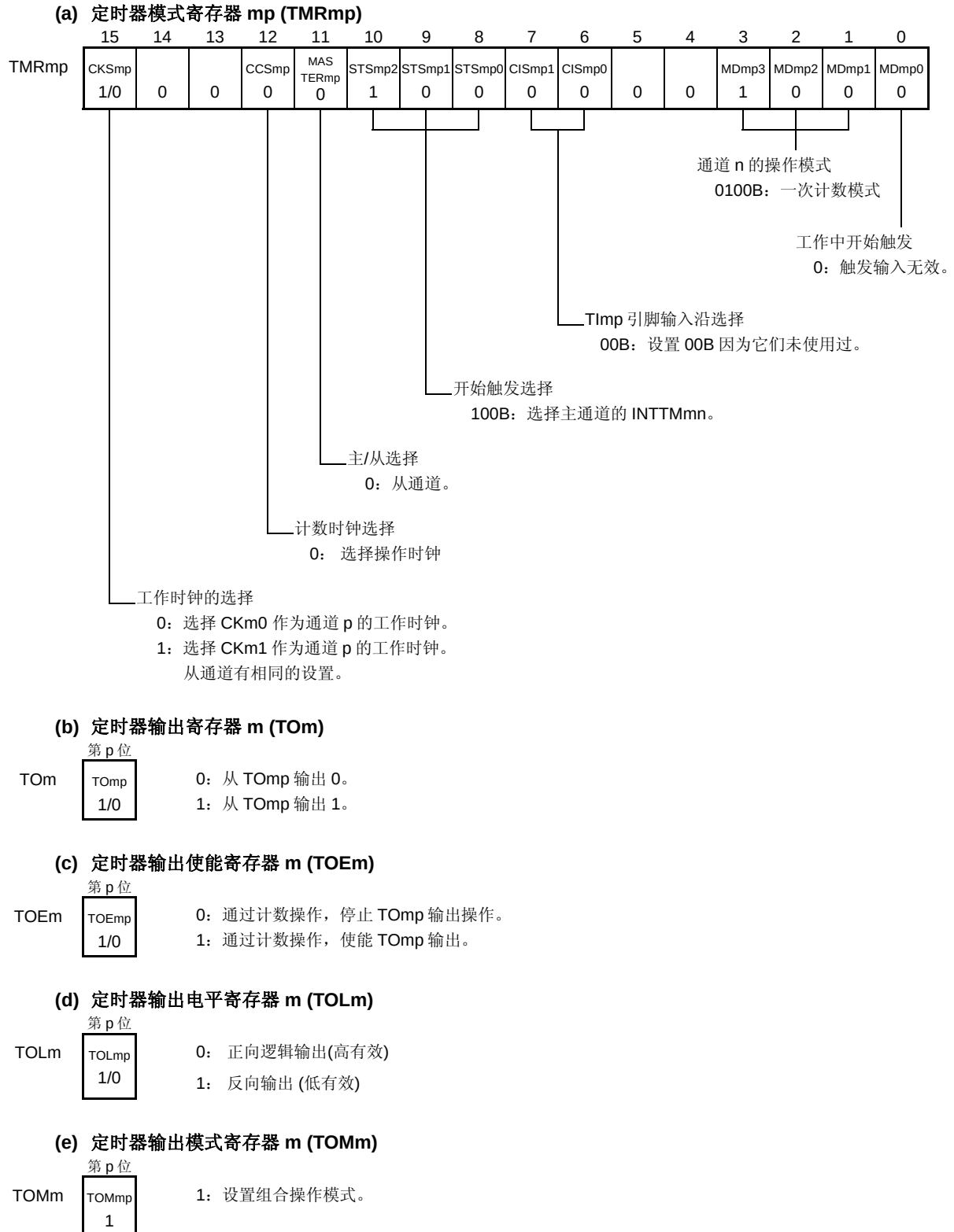
(e) 定时器输出模式寄存器 m (TOMm)

第 n 位

TOMm 第 n 位: TOMmn 0: 设置反转模式。

备注 m: 单元编号, n: 通道编号, p: 从通道编号(p = n+1),
当 m = 0: n = 0, 2, 4, 6
当 m = 1: n = 0, 2

图 6-65. 单脉冲输出功能 (从通道) 寄存器设置的示例



备注 m: 单元编号, n: 通道编号, p: 从通道编号(p = n+1),
当 m = 0: n = 0, 2, 4, 6, 当 m = 1: n = 0, 2

图 6-66. 单脉冲输出功能的操作过程(1/2)

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置PER0寄存器的TAU0EN位，TAU1EN位为1。 —————→	上电状态。 每个通道停止操作。 (开始供给时钟并且每个寄存器使能写入。)
	设置TPSm寄存器。 决定CKm0和CKm1的时钟频率。	
通道默认设置	设置要使用的两个通道的TMRmn和TMRmp寄存器(决定通道的操作模式)。 输出延迟设置到主通道的TDRmn寄存器，脉冲宽度设置到从通道的TDRmp寄存器。	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置从通道。 将TOMm寄存器的TOMmp位设置为1(组合操作模式)。 设置TOLmp位。 设置TOmp位并决定TOmp输出的默认电平。 —————→	TOMn引脚为高阻输出状态。
	设置TOEmp至1并使能TOmp操作。 —————→	当端口模式寄存器在输出模式下并且端口寄存器为0时，输出TOmp 默认设置电平。
	将端口寄存器和端口模式寄存器清为0。 —————→	因为通道停止操作，TOmp 不会改变。
		TOmp引脚输出TOmp的设置电平。

备注 m: 单元编号, n: 主通道编号, p: 从通道编号(p = n+1),
 当 m = 0: n = 0, 2, 4, 6
 当 m = 1: n = 0, 2

图 6-66. 单脉冲输出功能的操作过程(2/2)

	软件操作	硬件状态
操作恢复	操作开始 设置TOEmp(从通道) 为1 (仅当操作恢复时)。 TSm寄存器的TSmn(主通道) 和TSmp(从通道) 位同时设置为1。 因为触发位, TSmn和TSmp位自动返回为0。	TEmn和TEmp设置为1, 主通道进入Tlmn输入边沿检测等待状态。 计数器停止操作。
	检测主通道的Tlmn引脚输入有效沿。	主通道开始计数。
	操作期间 只有TMRmn寄存器的CISmn1和CISmn0位的设置值可以改变。 TMRmp, TDRmn, TDRmp寄存器的设置值, TOMmn, TOMmp, TOLmn, TOLmp位不能改变。 TCRmn和TCRmp寄存器总是可读。 不使用TSRmn和TSRmp寄存器。 TOM和TOEm寄存器的设置值可以改变。	当检测到Tlmn引脚有效输入沿, 且计数器开始向下计数时, 主通道加载TDRmn至TCRmn的值。当计数值达到TCRmn = 0000H, 产生INTTMmn输出并且计数器停止直到Tlmn引脚的下一有效沿输入。 从通道, 由主通道的INTTMmn触发, 加载TDRmp至TCRmp的值并且计数器开始向下计数。TOmp的输出电平在主通道输出INTTMmn后的一个计数时钟开始有效。当TCRmp=0000H时变为无效, 并且计数操作停止。 之后, 重复上面的操作。
	操作停止 TTmn(主通道)和TTmp(从通道)位同时设置为1。 因为触发位, TTmn和TTmp位自动返回为0。 从通道的TOEmp清除为0, 值设置到TOmp位。	TEmn, TEm = 0, 并且计数操作停止。 TCRmn和TCRmp保持计数值并停止。 TOmp输出不初始化而保持当前状态。 TOmp引脚输出TOmp的设置电平。
TAU 停止	需要保持TOmp 引脚输出电平时 在需要保持的值设置到端口寄存器后 清TOmp 位为0。	TOmp引脚输出电平由端口功能保持。
	不需要保持TOmp引脚输出电平时 切换端口模式寄存器到输入模式。	TOmp引脚输出电平变为高阻输出状态。
	将PER0寄存器的TAU0EN位, TAU1EN位清零。	掉电状态 所有电路初始化, 并且每个通道的SFR也初始化。 (TOmp位清为0, 并且TOmp引脚设置为端口模式。)

备注 m: 单元编号, n: 通道编号, p: 从通道编号($p = n+1$),
 当 $m = 0$: $n = 0, 2, 4, 6$
 当 $m = 1$: $n = 0, 2$

6.8.3 作为多路 PWM 输出功能的操作

通过执行 PWM 功能和使用两个或两个以上通道，可产生多个 PWM 输出信号。

例如，当使用两个从通道时，输出脉冲的周期和占空比可通过下面的公式计算出。

$$\begin{aligned} \text{脉冲周期} &= \{ \text{TDRmn (主通道) 的设置值} + 1 \} \times \text{计数时钟周期} \\ \text{占空比 1 [\%]} &= \{ \text{TDRmp (从通道 1) 的设置值} \} / \{ \text{TDRmn (主通道) 的设置值} + 1 \} \times 100 \\ \text{占空比 2 [\%]} &= \{ \text{TDRmq (从通道 2) 的设置值} \} / \{ \text{TDRmn (主通道) 的设置值} + 1 \} \times 100 \end{aligned}$$

备注 如果 TDRmp(从通道 1)的设置值 > (TDRmn 的设置值 (主通道) + 1)，或者如果 TDRmq(从通道 2)的设置值 > (TDRmn 的设置值 (主通道) + 1)，仍作为 100%输出。

主通道的 TCRmn 运行于间隔定时器模式，计数周期。

从通道 1 的 TCRmp 运行于一次计数模式，计数占空比，从 TOmp 引脚输出 PWM 波形。TCRmp 把 TDRmp 的值加载到 TCRmp 中，主通道的 INTTMmn 作为启动触发，向下计数开始。当 TCRmp = 0000H 时，TCRmp 输出 INTTMmp，并且停止计数直到下一个启动触发 (主通道的 INTTMmn) 输入。TOmp 的输出电平在主通道产生 INTTMmn 后的一个计数时钟后开始有效，TCRmp = 0000H 时变为无效。

与从通道 1 的 TCRmp 相同，从通道 2 的 TCRmq 运行于一次计数模式，计数占空比，且从 TOmq 引脚输出 PWM 波形。TCRmq 把 TDRmq 的值加载到 TCRmq 中，主通道的 INTTMmn 作为启动触发，向下计数开始。当 TCRmq = 0000H 时，TCRmq 输出 INTTMmq，并且停止计数直到下一个启动触发 (主通道的 INTTMmn) 输入。TOmq 的输出电平在主通道产生 INTTMmn 后的一个计数时钟后开始有效，TCRmq = 0000H 时变为无效。

当通道 0 如上述作为主通道使用时，如用作定时器阵列单元 0，最多产生 7 类 PWM 信号，如用作定时器阵列单元 1，则最多产生 3 类 PWM 信号。

注意事项 要重写主通道的 TDRmn 和从通道 1 的 TDRmp，写访问必须至少执行两次。因为在主通道的 INTTMmn 产生后 TDRmn 和 TDRmp 的值赋给 TCRmn 和 TCRmp，如果重写在主通道的 INTTMmn 发生之前和之后分别执行，TOmp 引脚可能无法输出预期波形。要重写主通道的 TDRmn 和从通道的 TDRmp，必须在主通道的 INTTMmn 发生之后立即重写这两个寄存器 (这也适用于从通道 2 的 TDRmq)。

备注 m: 单元编号, n: 通道编号, p: 从通道编号 1 (n+1), q: 从通道编号 2 (n+2)

当 m = 0

n = 0, 2, 4

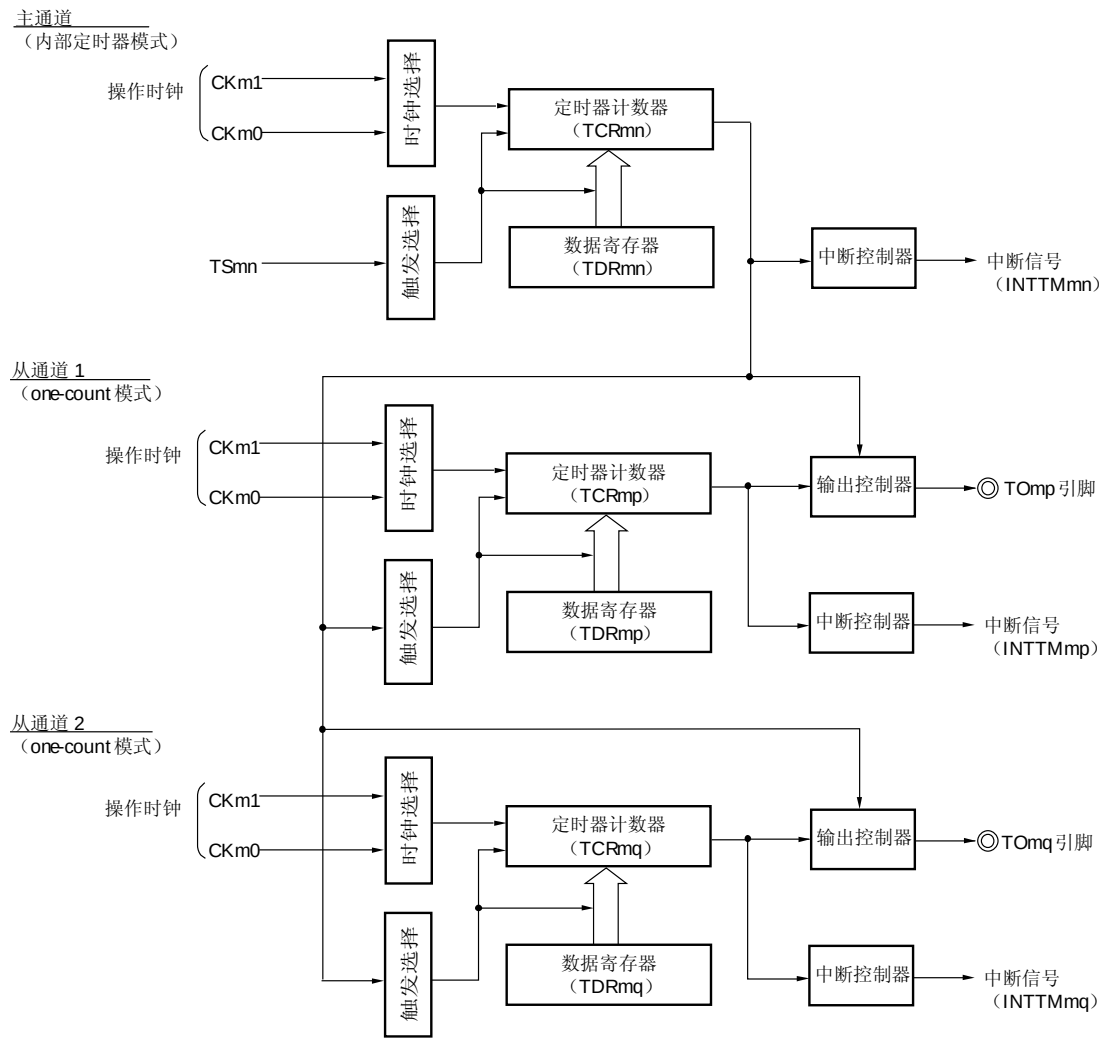
n < p < q ≤ 7 (此处 p 是一个大于 n 的连续整数)

当 m = 1

n = 0

n < p < q ≤ 3 (此处 p 是一个大于 n 的连续整数)

图 6-67. 多路 PWM 输出功能操作的框图(输出两种类型 PWM)



备注 m: 单元编号, n: 通道编号, p: 从通道编号 1 (n+1), q: 从通道编号 2 (n+2)

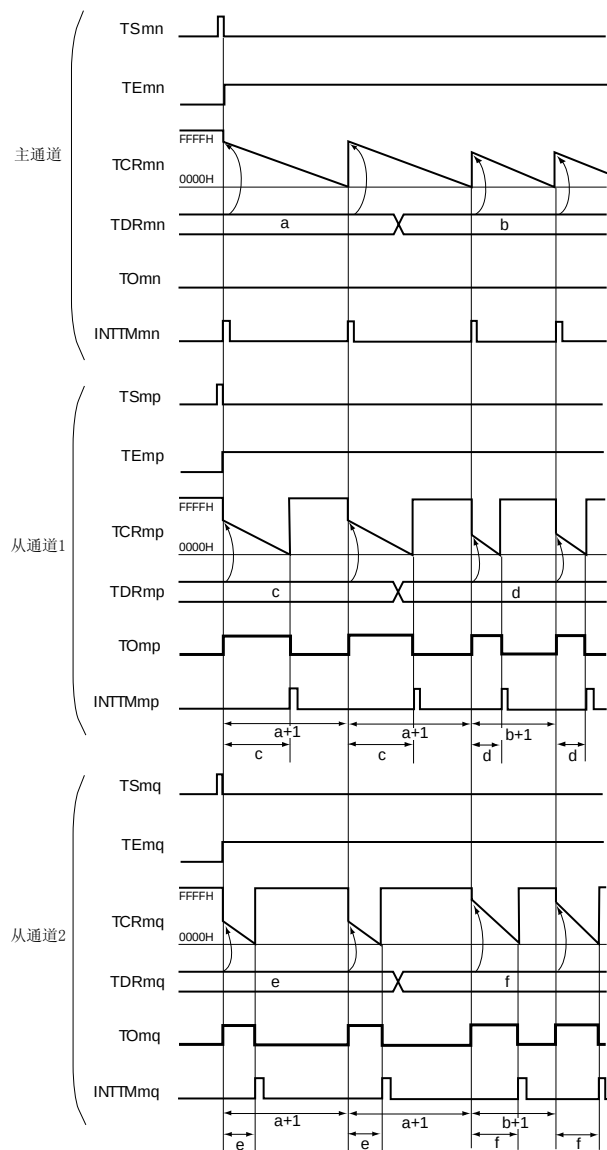
当 m = 0

- n = 0, 2, 4
- $n < p < q \leq 7$ (此处 p 是一个大于 n 的连续整数)

当 m = 1

- n = 0
- $n < p < q \leq 3$ (此处 p 是一个大于 n 的连续整数)

图 6-68. 多路 PWM 输出功能(输出 两种类型 PWM)操作的基本时序示例



备注 m: 单元编号, n: 通道编号, p: 从通道编号 1 (n+1), q: 从通道编号 2 (n+2)

当 $m = 0$

$n = 0, 2, 4$

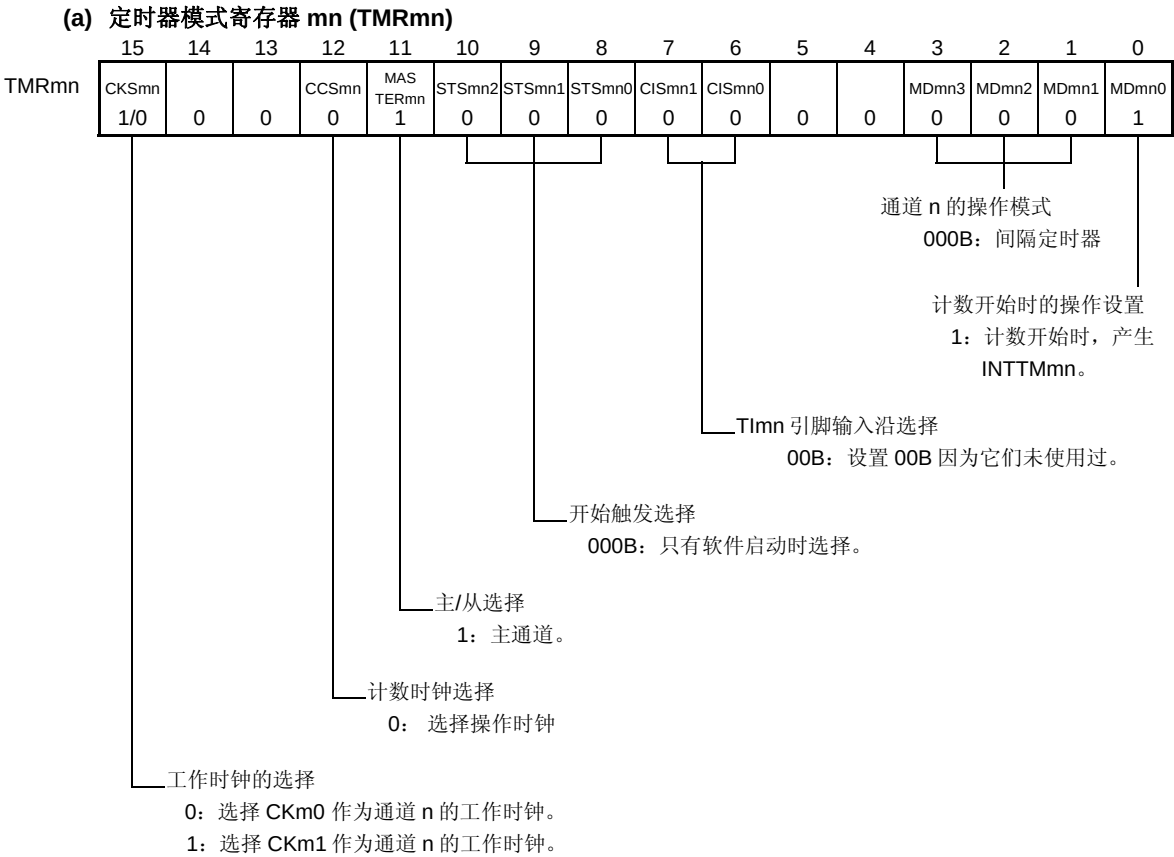
$n < p < q \leq 7$ (此处 p 是一个大于 n 的连续整数)

当 $m = 1$

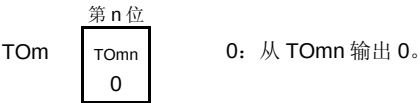
$n = 0$

$n < p < q \leq 3$ (此处 p 是一个大于 n 的连续整数)

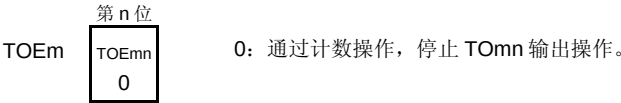
图 6-69. 多路 PWM 输出功能(主通道)时寄存器设置内容的示例



(b) 定时器输出寄存器 m (TOM)



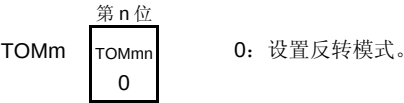
(c) 定时器输出使能寄存器 m (TOEm)



(d) 定时器输出电平寄存器 m (TOLm)

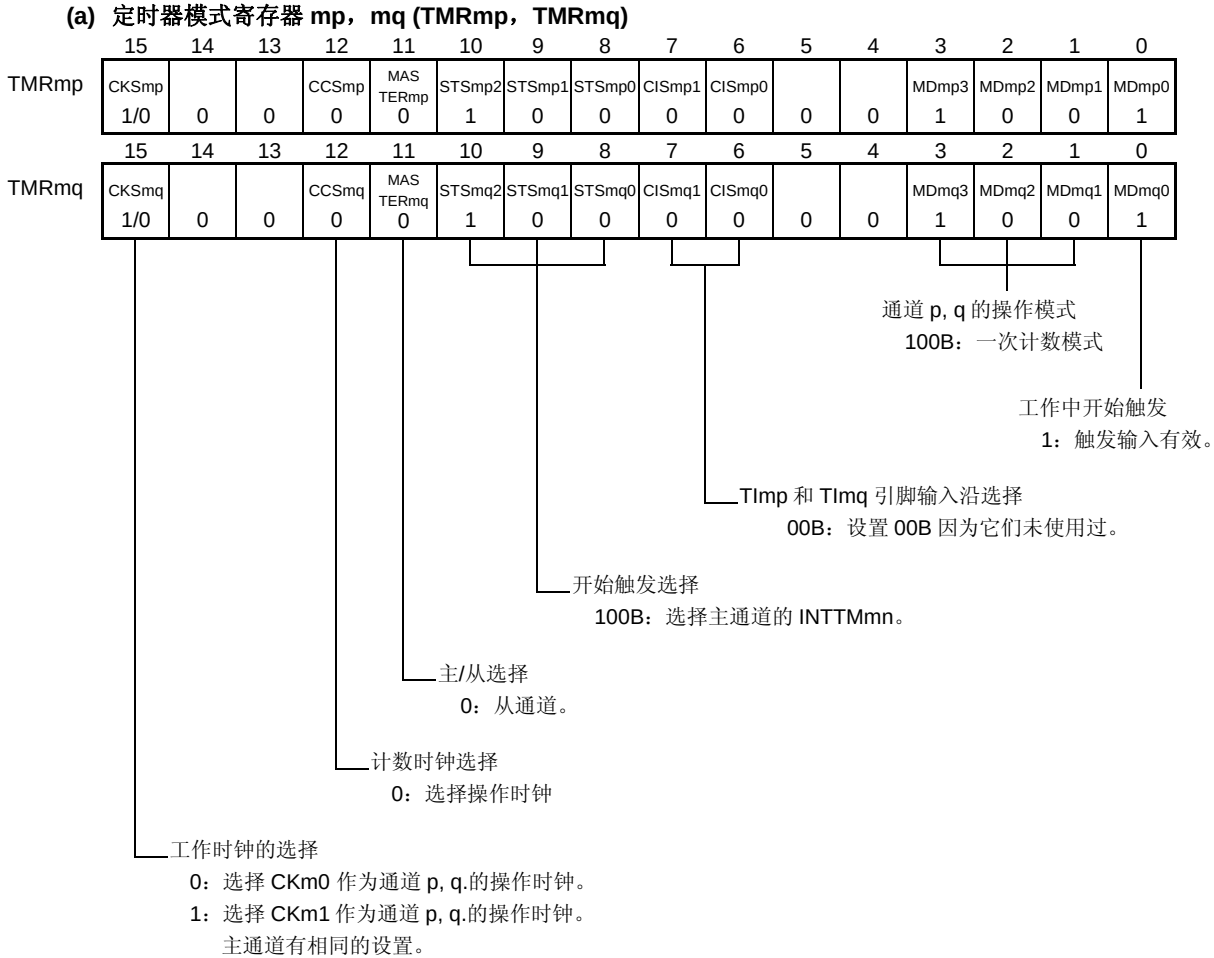


(e) 定时器输出模式寄存器 m (TOMm)



备注 m: 单元编号 (m = 0, 1), n: 通道编号
当 m = 0: n = 0, 2, 4
当 m = 1: n = 0

图 6-70. 多路 PWM 输出功能(从通道)寄存器设置内容的示例(输出两种类型 PWM)



(b) 定时器输出寄存器 m (TOM)

TOM	第 q 位 第 p 位		
	TOMq	TOMp	
	1/0	1/0	

0: 从 TOMp 或 TOMq 输出 0。
1: 从 TOMp 或 TOMq 输出 1。

(c) 定时器输出使能寄存器 m (TOEm)

TOEm	第 q 位 第 p 位		
	TOEmq	TOEmp	
	1/0	1/0	

0: 通过计数操作, 停止 TOMp 或 TOMq 的输出操作。
1: 通过计数操作, 使能 TOMp 或 TOMq 的输出操作。

(d) 定时器输出电平寄存器 m (TOLm)

TOLm	第 q 位 第 p 位		
	TOLmq	TOLmp	
	1/0	1/0	

0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)

(e) 定时器输出模式寄存器 m (TOMm)

TOMm	第 q 位 第 p 位		
	TOMmq	TOMmp	
	1	1	

1: 设置组合操作模式。

(备注在下页列出)

备注 m: 单元编号, n: 通道编号, p: 从通道编号 1 (n+1), q: 从通道编号 2 (n+2)

当 m = 0

 n = 0, 2, 4

$n < p < q \leq 7$ (此处 p 和 q 是一个大于 n 的连续整数)

当 m = 1

 n = 0

$n < p < q \leq 3$ (此处 p 和 q 是一个大于 n 的连续整数)

图 6-71. 多路 PWM 输出功能的操作过程 (1/2)

	软件操作	硬件状态
TAU 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置PER0寄存器的TAU0EN位, TAU1EN位为1。 →	上电状态。 每个通道停止操作。 (开始供给时钟并且每个寄存器使能写入。)
	设置TPSm寄存器。 决定CKm0和CKm1的时钟频率。	
通道默认设置	设置每个通道要使用的TMRmn, TMRmp, 和TMRmq寄存器通道(决定通道的操作模式)。 间隔(周期)值设置到主通道的TDRmn寄存器, 占空比设置到从通道的TDRmp和TDRmq寄存器。	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置从通道。 把 TOMm 寄存器的 TOMmp 和 TOMq 位设为 1 (组合操作模式)。 清除 TOLmp 和 TOLmq 位为 0。 设置 TOmp 和 TOMq 位, 并确定 TOmp 和 TOMq 输出默认电平。 →	TOMn引脚为高阻输出状态。 当端口模式寄存器在输出模式下并且端口寄存器为0时, 输出TOmp和TOMq 默认设置电平。
	设置 TOEmp 和 TOEmq 为 1 并使能 TOmp 和 TOMq 操作。 →	TOmp 或TOMq不会改变因为通道停止工作。
	将端口寄存器和端口模式寄存器清为 0。 →	TOmp和TOMq引脚输出TOmp和TOMq的设置值。
操作开始	设置TOEmp和TOEmq(从通道)为1 (仅当恢复操作时)。 TSm寄存器的TSmn(主通道)、TSmp和TSmq(从通道)位同时设置为1。 因为是触发位, TSmn, TSmp, 和TSmq位自动返回为0。	TEmn = 1, TEmq, TEmq = 1 当主通道开始计数, 产生INTTMmn。由此中断触发的从通道也开始计数。

操作恢复(在下一页)

图 6-71. 多路 PWM 输出功能的操作过程 (2/2)

	软件操作	硬件状态
操作期间	不能改变TMRmn, TMRmp, TMRmq寄存器中TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, 和TOLmq 位的设置值。 在主通道的INTTMmn产生后, TDRmn, TDRmp和TDRmq寄存器的设置值可更改。 TCRmn, TCRmp, 和TCRmq寄存器总是可读。 不使用TSRmn、TSRmp、和TSRmq寄存器。 TOM和TOEm寄存器的设置值可以改变。	主通道的计数器加载TDRmn的值到TCRmn, 并向下载数。当计数值达到TCRmn = 0000H, 产生INTTMmn输出。同时, TDRmn寄存器的值加载到TCRmn, 计数器开始再次向下计数。 在从通道1, 加载TDRmp的值到TCRmp, 由主通道的INTTMmn触发并且计数器开始向下计数。TOMp的输出电平在主通道输出INTTMmn后的一个计数时钟后开始有效。当TCRmp=0000H时变为无效, 并且计数操作停止。 在从通道2, TDRmq的值传输至TCRmq, 由主通道的INTTMmn触发, 并且计数器开始向下计数。TOMq的输出电平在主通道输出INTTMmn后的一个计数时钟开始有效。当TCRmq=0000H时变为无效, 并且计数操作停止。之后, 重复上面的操作。
操作停止	TTmn位(主通道)、TTmp和TTmq(从通道)位同时设置为1。 因为是触发位, TTmn、TTmp和TTmq位自动返回为0。	TEmn、TEmp和TEmq = 0, 并且计数操作停止。 TCRmn、TCRmp和TCRmq保持计数值并停止。 TOMP和TOMq输出没有初始化而是保持目前状态。
	从通道的TOEmp或TOEmq清除为0, 值设置到TOMP和TOMq 位。	TOMP和TOMq引脚输出TOMP和TOMq的设置值。
TAU 停止	需要保持TOMP和TOMq 引脚输出电平时 把需要保持的值设置到端口寄存器后清除TOMP和TOMq位为0。	TOMP和TOMq引脚输出电平通过端口功能保持。
	不需要保持TOMP和TOMq引脚输出电平时 切换端口模式寄存器到输入模式。	TOMP和TOMq引脚输出电平变为高阻输出状态。
	将PER0寄存器的TAU0EN位, TAU1EN位清零。	掉电状态 所有电路初始化, 并且每个通道的SFR也初始化。 (TOMP和TOMq位清零, 并将TOMP和TOMq引脚设置为端口模式。)

备注 m: 单元编号, n: 通道编号, p: 从通道编号 1 (n+1), q: 从通道编号 2 (n+2)

当 m = 0

n = 0, 2, 4

n < p < q ≤ 7(此处 p 是一个大于 n 的连续整数)

当 m = 1

n = 0

n < p < q ≤ 3(此处 p 是一个大于 n 的连续整数)

操作恢复(在前一页).

第 7 章 实时计数器

7.1 实时计数器的功能

实时计数器有如下特点。

- 具有年、月、星期、日、小时、分钟和秒的计数器，可以共计 99 年。
- 固定周期中断功能(周期：1 月至 0.5 秒)
- 报警中断功能(报警：星期、小时、分钟)
- 间隔中断功能
- 1 Hz 的引脚输出功能
- 512 Hz 或 16.384 kHz 或 32.768 kHz 的引脚输出功能

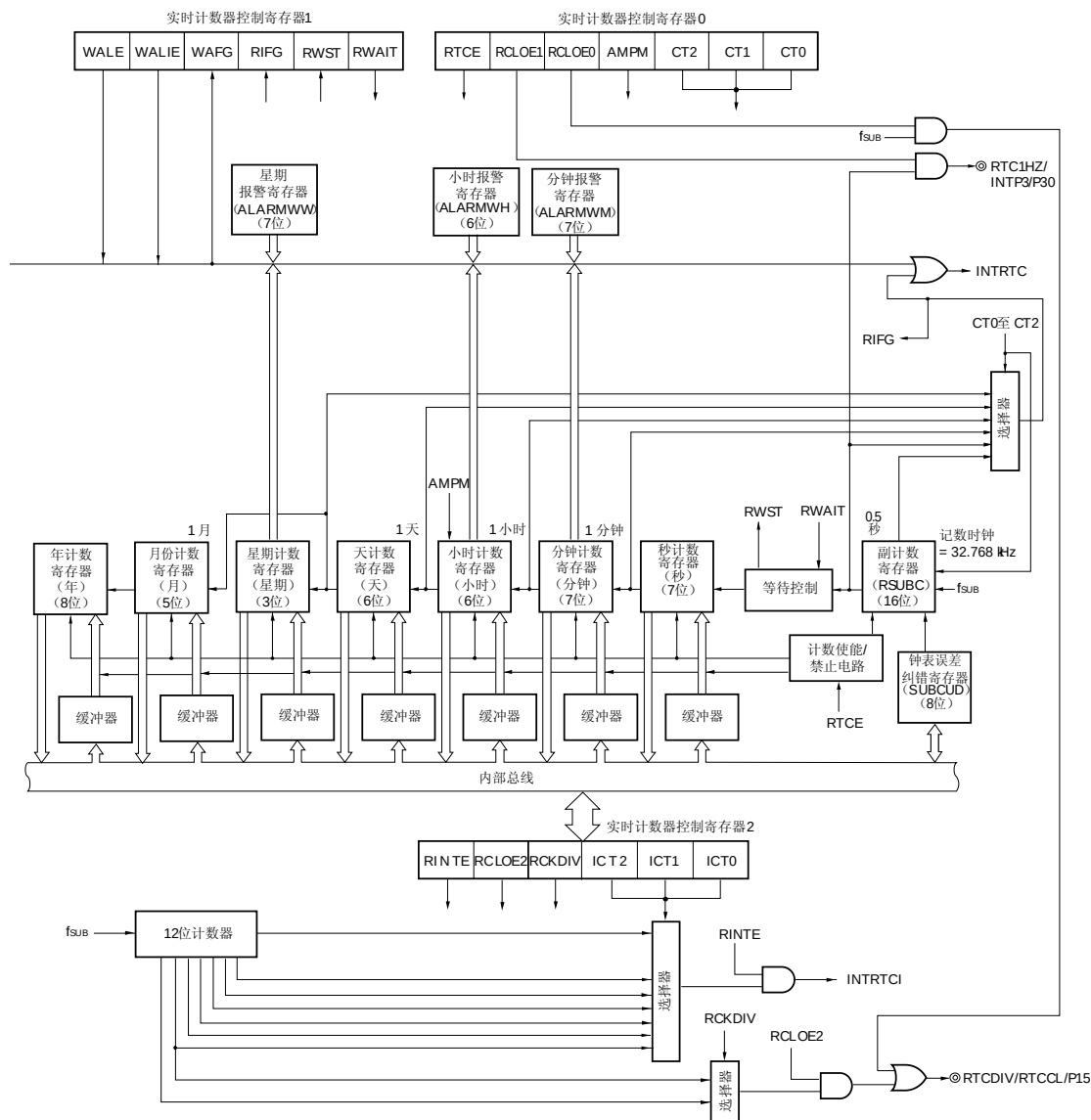
7.2 实时计数器的配置

实时计数器包括如下硬件。

表 7-1. 实时计数器的配置

项目	配置
控制寄存器	外围使能寄存器 0 (PER0)
	实时计数器控制寄存器 0 (RTCC0)
	实时计数器控制寄存器 1 (RTCC1)
	实时计数器控制寄存器 2 (RTCC2)
	副计数寄存器 (RSUBC)
	秒计数寄存器 (SEC)
	分钟计数寄存器 (MIN)
	小时计数寄存器 (HOUR)
	日计数寄存器 (DAY)
	星期计数寄存器 (WEEK)
	月计数寄存器 (MONTH)
	年计数寄存器 (YEAR)
	钟表错误修正寄存器 (SUBCUD)
	分钟报警寄存器(ALARMWM)
	小时报警寄存器(ALARMWH)
	星期报警寄存器(ALARMWW)

图 7-1. 实时计数器的框图



7.3 控制实时计数器的寄存器

以下 16 个寄存器用于控制实时计数器。

- 外围使能寄存器 0 (PER0)
- 实时计数器控制寄存器 0 (RTCC0)
- 实时计数器控制寄存器 1 (RTCC1)
- 实时计数器控制寄存器 2 (RTCC2)
- 副计数寄存器 (RSUBC)
- 秒计数寄存器 (SEC)
- 分钟计数寄存器 (MIN)
- 小时计数寄存器 (HOUR)
- 日计数寄存器 (DAY)
- 星期计数寄存器 (WEEK)
- 月计数寄存器 (MONTH)
- 年计数寄存器 (YEAR)
- 钟表错误修正寄存器 (SUBCUD)
- 分钟报警寄存器 (ALARMWM)
- 小时报警寄存器 (ALARMWH)
- 星期报警寄存器 (ALARMWW)

(1) 外围使能寄存器 0 (PER0)

PER0 用于使能或者禁止使用每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少消耗和噪声。
使用实时计数器时，务必设置此寄存器的第 7 位 (RTCEN) 为 1。
可以通过 1 位或者 8 位存储器操作指令来设置 PER0。
复位后这个寄存器的值为 00H。

图 7-2. 外围使能寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
	RTCEN	实时计数器(RTC)输入时钟供给的控制*						
	0	停止输入时钟的供给 - 由实时计数器(RTC)使用的 SFR 不能写入。 - 实时计数器(RTC)在复位状态。						
	1	供给输入时钟 由实时计数器(RTC)使用的 SFR 可读/写。						

注 当从CPU访问实时计数(RTC)寄存器时，通过RTCEN来开始或停止时钟。RTCEN不能控制供给RTC的操作时钟 (f_{SUB})。

- 注意事项
- 1

使用实时计数器时，首先设置 RTCEN 位为 1，直至副系统时钟(f_{SUB})振荡稳定。如果 RTCEN = 0，忽略写入到实时计数器的控制寄存器，并且，即使读取寄存器，只能读到默认值。
- 2

当使用副系统时钟时，在 HALT 模式下可通过设置操作速度模式控制寄存器(OSMC)的 RTCLPC 为 1 来停止对实时计数器之外的外围功能提供时钟。此时，设置 RTCEN 为 1 并将 PER0 的位 0 至 6 清零。
- 3

确保将位 6 清零。

(2) 实时计数器控制寄存器 0(RTCC0)

RTCC0 寄存器是 8 位寄存器，它用于启动或停止实时计数器操作、控制 RTCCL 和 RTC1HZ 引脚、设置 12 或 24 小时系统和固定周期中断功能。
可以由 1 位或 8 位存储器操作指令设置 RTCC0。
复位后这个寄存器的值为 00H。

图 7-3. 实时计数器控制寄存器 0 (RTCC0)的格式

地址: FFF9DH 复位后: 00H R/W

符号	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RCLOE0	AMPM	CT2	CT1	CT0

RTCE	实时计数器操作控制
0	停止计数器操作。
1	启动计数器操作。

RCLOE1	RTC1HZ 引脚输出控制
0	禁止 RTC1HZ 引脚输出 (1 Hz)。
1	使能 RTC1HZ 引脚输出 (1 Hz)。

RCLOE0 [#]	RTCCL 引脚输出控制
0	禁止 RTCCL 引脚的输出 (32.768 kHz)。
1	使能 RTCCL 引脚的输出 (32.768 kHz)。

AMPM	选择12/24小时系统
0	12 小时系统(a.m. 和 p.m. 显示)
1	24 小时系统
- 通过设置 RWAIT (RTCC1 的位 0) 为 1 来改变 AMPM 的值。 - 当 AMPM 值改变时, 小时计数寄存器(HOUR)的值将自动改变为与时间设置系统相对应的数值。 - 表 7-2 表示所显示的时间数字。	

CT2	CT1	CT0	固定周期中断(INTRTC)选择
0	0	0	不使用固定周期中断功能
0	0	1	每 0.5 秒一次(与秒累加同步)
0	1	0	每 1 秒一次(与秒累加相同)
0	1	1	每 1 分钟一次(每分钟的 00 秒)
1	0	0	每 1 小时一次(每小时的 00 分 00 秒)
1	0	1	每 1 天一次(每天的 00 小时, 00 分, 00 秒)
1	1	×	每 1 月一次(每月的第 1 天, 00 小时 a. m., 00 分, 00 秒)
在计数器工作(RTCE = 1)时改变 CT2 至 CT0 的值, 通过中断屏蔽标志寄存器在禁止中断服务 INTRTC 之后重新写入 CT2 至 CT0 数值。此外, 在对 CT2 至 CT0 重写之后, 在清除 RIFG 和 RTCIF 标志后使能中断服务。			

注 不能同时使能 RCLOE0 和 RCLOE2。

注意事项 如果在 RTCE = 1 时改变 RCLOE0 和 RCLOE1, 那么 32.768 kHz 和 1 HZ 的最后波形的输出信号可能变短。

备注 ×: 不必考虑

(3) 实时计数器控制寄存器 1 (RTCC1)

RTCC1 寄存器是 8 位寄存器，它用于控制报警中断功能和计数器等待时间。
可以由 1 位或 8 位存储器操作指令设置 RTCC1。
复位后这个寄存器的值为 00H。

图 7-4. 实时计数器控制寄存器 1 (RTCC1)的格式 (1/2)

地址: FFF9EH 复位后: 00H R/W

符号	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	报警操作控制
0	匹配操作无效
1	匹配操作有效
在计数器工作(RTCE = 1)且 WALIE = 1 时设置 WALE 位的数值时，通过中断屏蔽标志寄存器在禁止中断服务 INTRTC 之后重写 WALE 位。此外，在重写 WALE 位之后清除 WAFG 和 RTCIF 标志。当设置每个报警寄存器时 (WALIE 的标志 RTCC1 位、ALARMWM 寄存器、ALARMWH 寄存器和 ALARMWW 寄存器)，为 WALE 位设置匹配操作至无效("0")。	

WALIE	控制报警中断(INTRTC)功能操作
0	报警匹配时不产生中断。
1	报警匹配时产生中断。

WAFG	报警检测状态标志
0	报警不匹配
1	检测报警匹配
此状态标志显示报警匹配检测。仅当 WALE = 1 和检测到报警匹配后置一个时钟(32.768 kHz)为 1 时有效。当写入"0"时，清零该标志。写入"1"为无效。	

图 7-4. 实时计数器控制寄存器 1 (RTCC1)的格式 (2/2)

RIFG	固定周期中断状态标志
0	不产生固定周期中断。
1	产生固定周期中断。
此标志显示固定周期中断产生状态。当产生固定周期中断时，设置为“1”。 当写入“0”时，清零该标志。写入“1”为无效。	

RWST	实时计数器等待状态标志
0	计数器操作。
1	读或写计数器值的模式。
此状态标志表明 RWAIT 设置是否有效。 读或写计数器值之前，请确认此标志值为 1。	

RWAIT	实时计数器的等待控制
0	设置计数器操作。
1	停止 SEC 至 YEAR 计数器。读或写计数器值的模式。
此位控制计数器操作。 确保写入“1”来读取或写入计数器值。 因为 RSUBC 操作继续，在 1 秒内完成读或写，并将此位清零。当 RWAIT = 1 时，它占用 1 个时钟 (32.768kHz) 直到计数器值读出或写入。 当 RWAIT = 1 时，如果 RSUBC 溢出，在 RWAIT = 0 后它累加。如果秒计数寄存器写入，因为 RSUBC 清零，它将不能累加。	

注意事项 当 RTCC1 寄存器由 1 位操作指令写入时，RIFG 和 WAFG 标志可能清零。因此，使用 8 位操作指令写入 RTCC1 寄存器。为防止 RIFG 和 WAFG 标志在写入、空写入时清零，将相应的位设为“1”。由于未使用 RIFG 和 WAFG 标志而使数值可能重写时，可使用 1 位操作指令写入 RTCC1 寄存器。

备注 固定周期中断和报警匹配中断使用相同的中断源 (INTRTC)。当同时使用这两种类型中断时，在 INTRTC 产生的基础上，通过检查固定周期中断状态标志(RIFG)和报警检测状态标志(WAFG)，来判断产生了哪种中断。

(4) 实时计数器控制寄存器 2 (RTCC2)

RTCC2 寄存器是 8 位寄存器，它用于控制间隔中断功能和 RTCDIV 引脚。

可以由 1 位或 8 位存储器操作指令设置 RTCC2。

复位后这个寄存器的值为 00H。

图 7-5. 实时计数器控制寄存器 2 (RTCC2)的格式

地址: FFF9FH 复位后: 00H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
RTCC2	RINTE	RCLOE2	RCKDIV	0	0	ICT2	ICT1	ICT0

RINTE	ICT2	ICT1	ICT0	选择间隔中断(INTRTCI)
0	×	×	×	不产生间隔中断
1	0	0	0	$2^6/f_{XT}$ (1.953125 ms)
1	0	0	1	$2^7/f_{XT}$ (3.90625 ms)
1	0	1	0	$2^8/f_{XT}$ (7.8125 ms)
1	0	1	1	$2^9/f_{XT}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{XT}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{XT}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{XT}$ (125 ms)

RCLOE2 ^注	RTCDIV 引脚输出控制
0	禁止 RTCDIV 引脚输出。
1	使能 RTCDIV 引脚输出。

RCKDIV	选择RTCDIV 引脚输出频率
0	RTCDIV 引脚输出 512 Hz。 (1.95 ms)
1	RTCDIV 引脚输出 16.384 kHz。 (0.061 ms)

注 不能同时使能 RCLOE0 和 RCLOE2。

注意事项 1. 当 RINTE = 0 时，更改 ICT2、ICT1 和 ICT0。

2. 当 RTCDIV 引脚输出停止，在最多 2 个 f_{RTC} 时钟和进入低电平后输出继续。当输出 512Hz 时，进入高电平后，输出立刻停止时，可能产生至少 1 个 f_{XT} 时钟宽度的脉冲。

3. 在实时计数器开始工作时，RTCDIV 引脚的输出宽度可能比首个间隔周期内设置的短。

(5) 副计数寄存器 (RSUBC)

RSUBC 寄存器为 16 位寄存器，用于计算实时计数器的 1 秒的参考时间。它的取值从 0000H 至 7FFFH，且以 32.768kHz 的时钟来计 1 秒。

可以 16 位存储器操作指令设置 RSUBC。

复位后这个寄存器的值为 0000H。

- 注意事项**
1. 当使用 SUBCUD 寄存器纠正，值将变为 8000H 或更大。
 2. 写入秒计数寄存器产生复位会将该寄存器清零。
 3. 在操作期间读取该寄存器，因为读出的是变化中的值，不能保证读出值的有效。

图 7-6. 副计数寄存器(RSUBC)的格式

地址: FFF90H 复位后: 0000H R

符号	7	6	5	4	3	2	1	0
RSUBC	SUBC7	SUBC6	SUBC5	SUBC4	SUBC3	SUBC2	SUBC1	SUBC0

地址: FFF91H 复位后: 0000H R

符号	7	6	5	4	3	2	1	0
RSUBC	SUBC15	SUBC14	SUBC13	SUBC12	SUBC11	SUBC10	SUBC9	SUBC8

(6) 秒计数寄存器 (SEC)

SEC 寄存器是 8 位寄存器，为 0 至 59(十进制)值和显示计数的秒值。

当副计数器溢出，它将开始计数。

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 00 至 59 十进制数到该寄存器。

可以 8 位存储器操作指令设置 SEC。

复位后这个寄存器的值为 00H。

图 7-7. 秒计数寄存器(SEC)的格式

地址: FFF92H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

(7) 分钟计数寄存器 (MIN)

MIN 寄存器是 8 位寄存器，为 0 至 59(十进制)值和显示计数的分钟值。

当秒计数器溢出它将开始计数。

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。即使秒钟计数寄存器在写入时溢出，该寄存器将忽略溢出并设置为写入值。以 BCD 码设置 00 至 59 十进制数到该寄存器。

可以 8 位存储器操作指令设置 MIN。

复位后这个寄存器的值为 00H。

图 7-8. 分钟计数寄存器(MIN)的格式

地址: FFF93H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

(8) 小时计数寄存器 (HOUR)

HOUR 寄存器是 8 位寄存器，将 00 至 23 或 01 至 12 和 21 至 32(十进制)显示为小时的计数值。

当分钟计数器溢出，它将开始计数。

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。即使分钟计数寄存器在写入时溢出，该寄存器将忽略溢出并设置为写入值。

根据由实时计数器控制寄存器 0(RTCC0)的位 3(AMPM)设置的时间系统，通过 BCD 码以十进制的格式设置 00 至 23 或 01 至 12 和 21 至 32 的值。如果 AMPM 位的值改变的话，HOUR 值将自动改变为相应的值。

可以 8 位存储器操作指令设置 HOUR。

复位信号的产生将寄存器置为 12H。

如果当 AMPM 位置为 1，复位后该寄存器的值清零(00H)。

图 7-9. 小时计数寄存器(HOUR)的格式

地址: FFF94H 复位后: 12H R/W

符号	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

注意事项 当 AMPM = 0(如果选择 12 小时系统)，HOUR 的第 5 位 (HOUR20) 显示 AM(0)/PM(1) 。

表 7-2 显示了 AMPM 位、HOUR 寄存器值和时间之间的关系。

表 7-2. 时间数字显示

24 小时制显示(AMPM 位 = 1)		12 小时制显示(AMPM 位 = 0)	
时间	HOUR 寄存器	时间	HOUR 寄存器
0	00H	0 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	0 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

当 AMPM 位为“0”时，HOUR 寄存器的值显示为 12 小时制，当 AMPM 位为“1”时，HOUR 寄存器的值显示为 24 小时制。

在 12 小时制显示中，HOUR 寄存器的第 50 位显示 AM 的 0 和 PM 的 1。

(9) 日计数寄存器 (DAY)

DAY 寄存器是 8 位寄存器，为 1 至 31(十进制)值和显示计数的天数值。
当小时计数器溢出它将开始计数。
计数器计数如下。

- 01 至 31(一月、三月、五月、七月、八月、十月、十二月)
- 01 至 30(四月、六月、九月、十一月)
- 01 至 29(二月、闰年)
- 01 至 28(二月、正常年)

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。即使小时计数寄存器在写入时溢出，该寄存器将忽略溢出并设置为写入值。以 BCD 码设置 01 至 31 的十进制数到该寄存器。
可以 8 位存储器操作指令设置 DAY。
复位信号的产生将寄存器置为 01H。

图 7-10. 日期计数寄存器(DAY)的格式

地址: FFF96H	复位后: 01H	R/W						
符号	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

(10) 星期计数寄存器 (WEEK)

WEEK 寄存器是 8 位寄存器，为 0 至 6(十进制)值和显示计数的星期值。

当日计数器溢出它将开始计数。

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 00 至 06 的十进制数到该寄存器。

可以 8 位存储器操作指令设置 WEEK。

复位后这个寄存器的值为 00H。

图 7-11. 星期计数寄存器(WEEK)的格式

地址: FFF95H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
星期	0	0	0	0	0	WEEK4	WEEK2	WEEK1

注意事项 对应于月计数寄存器或日计数寄存器的值不会自动存储到星期计数寄存器。复位释放后，如下设置星期计数寄存器。

日	星期
星期日	00H
星期一	01H
星期二	02H
星期三	03H
星期四	04H
星期五	05H
星期六	06H

(11) 月计数寄存器 (MONTH)

MONTH 寄存器是 8 位寄存器，为 0 至 12(十进制)值和显示计数的月值。
当日计数器溢出它将开始计数。
当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768kHz)的延迟。即使日计数寄存器在写入时溢出，该寄存器将忽略溢出并设置为写入值。以 BCD 码设置 01 至 12 的十进制数到该寄存器。
可以 8 位存储器操作指令设置 MONTH。
复位信号的产生将寄存器置为 01H。

图 7-12. 月计数寄存器(MONTH)的格式

地址: FFF97H	复位后: 01H	R/W						
符号	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

(12) 年计数寄存器 (YEAR)

YEAR 寄存器是 8 位寄存器，为 0 至 99(十进制)值和显示计数的月值。
当月计数器溢出它将开始计数。
值 00、04、08、...、92 和 96 表示闰年。
当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768kHz)的延迟。即使月计数寄存器在写入时溢出，该寄存器将忽略溢出并设置为写入值。以 BCD 码设置 00 至 99 十进制数到该寄存器。
可以 8 位存储器操作指令设置 YEAR。
复位后这个寄存器的值为 00H。

图 7-13. 年计数寄存器(YEAR)的格式

地址: FFF98H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

(13) 钟表错误修正寄存器 (SUBCUD)

当从副计数寄存器(RSUBC)到秒钟计数寄存器溢出发生数值(参考值: 7FFFH)改变而使时钟变慢或变快时, 使用该寄存器对时钟进行高精度修正。

可以 8 位存储器操作指令设置 SUBCUD。

复位后这个寄存器的值为 00H。

图 7-14. 钟表错误修正寄存器(SUBCUD)的格式

地址: FFF99H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	设置钟表纠错时间
0	当秒数字为 00, 20 或 40(每 20 秒)时纠正钟表错误。
1	仅当秒数字为 00(每 60 秒)时纠正钟表错误。
下列时序禁止写入 SUBCUD 寄存器。	
- 当 DEV = 0 设置时: 周期 SEC = 00H, 20H, 40H - 当 DEV = 1 设置时: 周期 SEC = 00H	

F6	设置钟表错误修正值
0	由 $\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2$ 增大
1	由 $\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2$ 减小
当 $(F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *)$ 时, 钟表错误不能修正。* 时 0 或 1。 /F5 至 /F0 是反向值的对应位(000011 当 111100)。 修正值范围: (当 F6=0) 2, 4, 6, 8, ..., 120, 122, 124 (当 F6=1) -2, -4, -6, -8, ..., -120, -122, -124	

可由钟表错误修正寄存器(SUBCUD)修正的数值范围如下所示。

	DEV = 0 (每20秒修正)	DEV = 1 (每60秒修正)
可修正范围	-189.2 ppm 至 189.2 ppm	-63.1 ppm 至 63.1 ppm
不包括的最大 量化误差	±1.53 ppm	±0.51 ppm
最小分辨率	±3.05 ppm	±1.02 ppm

备注 当纠错范围是-63.1 ppm 或更少或 63.1 ppm 或更多时, 设置 DEV 为 0。

(14) 分钟报警寄存器(ALARMWM)

该寄存器用作设置报警分钟。
可以 8 位存储器操作指令设置 ALARMWM。
复位后这个寄存器的值为 00H。

注意事项 以 BCD 码设置 00 至 59 十进制数到该寄存器。如果设置的值超出范围，报警将不检测。

图 7-15. 分钟报警寄存器(ALARMWM)的格式

地址: FFF9AH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
报警WM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

(15) 小时报警寄存器(ALARMWH)

该寄存器用于设置报警小时。
可以 8 位存储器操作指令设置 ALARMWH。
复位信号的产生将寄存器置为 12H。
如果当 AMPM 位置为 1，复位后该寄存器的值清零(00H)。

注意事项 以 BCD 码设置 00 至 23、01 至 12 或 21 至 32 十进制数到该寄存器。如果设置的值超出范围，报警将不检测。

图 7-16. 小时报警寄存器(ALARMWH)的格式

地址: FFF9BH	复位后: 12H	R/W						
符号	7	6	5	4	3	2	1	0
报警WH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

注意事项 如果 AMPM = 0(选择 12 小时系统)，ALARMWH 的第 5 位 (WH20) 显示 AM(0)/PM(1)。

(16) 星期报警寄存器(ALARMWW)

该寄存器用于设置报警数据。
可以 8 位存储器操作指令设置 ALARMWW。
复位后这个寄存器的值为 00H。

图 7-17. 星期报警寄存器(ALARMWW)的格式

地址: FFF9CH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
报警WW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

以下是设置报警的示例。

报警的时间	日							12 小时显示				24 小时显示			
	星期日	星期一	星期二	星期三	星期四	星期五	星期六	小时	小时 1	分钟	分钟 1	小时	小时 1	分钟	分钟 1
	W W 0	W W 1	W W 2	W W 3	W W 4	W W 5	W W 6	10		10		10		10	
每天, 0: 00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
每天, 01: 30a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
每天, 11: 59a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
星期一到星期五 0: 00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
星期日 1: 30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
星期一、星期三、星期五 11: 59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

7.4 实时计数器操作

7.4.1 启动实时计数器的操作

图 7-18. 启动实时计数器的操作流程图



- 注
1. 首先设置 RTCEN 位为 1，副系统时钟(f_{sub})振荡稳定。
 2. 当在 RTCE = 1 后切换至 STOP 模式而不等待 INTRTC = 1 时，确认在 7.4.2 启动操作后切换至 STOP 模式中所描述的步骤。

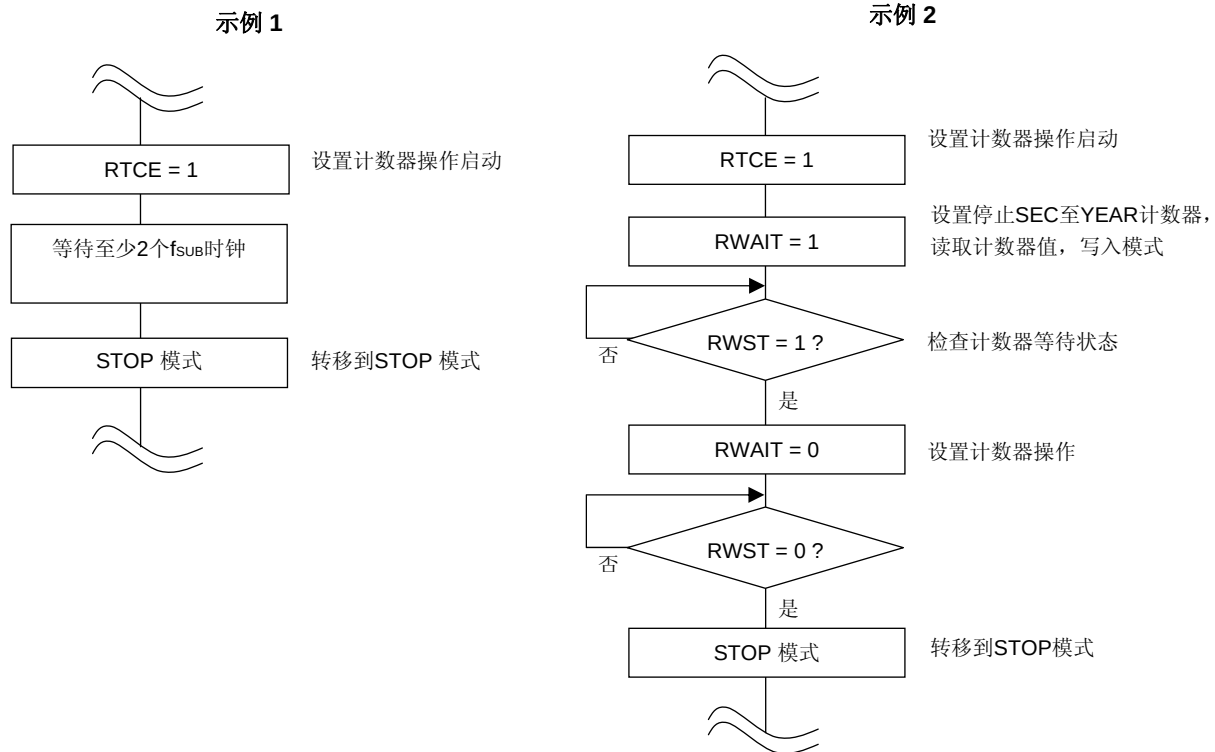
7.4.2 启动操作后切换至 STOP 模式

在设置 RTCE 为 1 后立即切换至 STOP 模式时，执行下列进程中的一个。

但是，在设置 RTCE 为 1 后，如果在第一个 INTRTC 中断发生后切换至 STOP 模式的话，则无需进行该进程。

- 在 RTCE 设置为 1 后，经历至少两个副系统时钟(f_{SUB}) (大约 $62\ \mu s$)后切换至 STOP 模式。
- 在设置 RTCE 为 1 后设置 RWAIT 为 1，通过 RWST 为 1 进行检查。此后，设置 RWAIT 为 0 并切换至 STOP 模式，之后通过 RWST 为 0 再次检查(参见图 7-19，示例 2)。

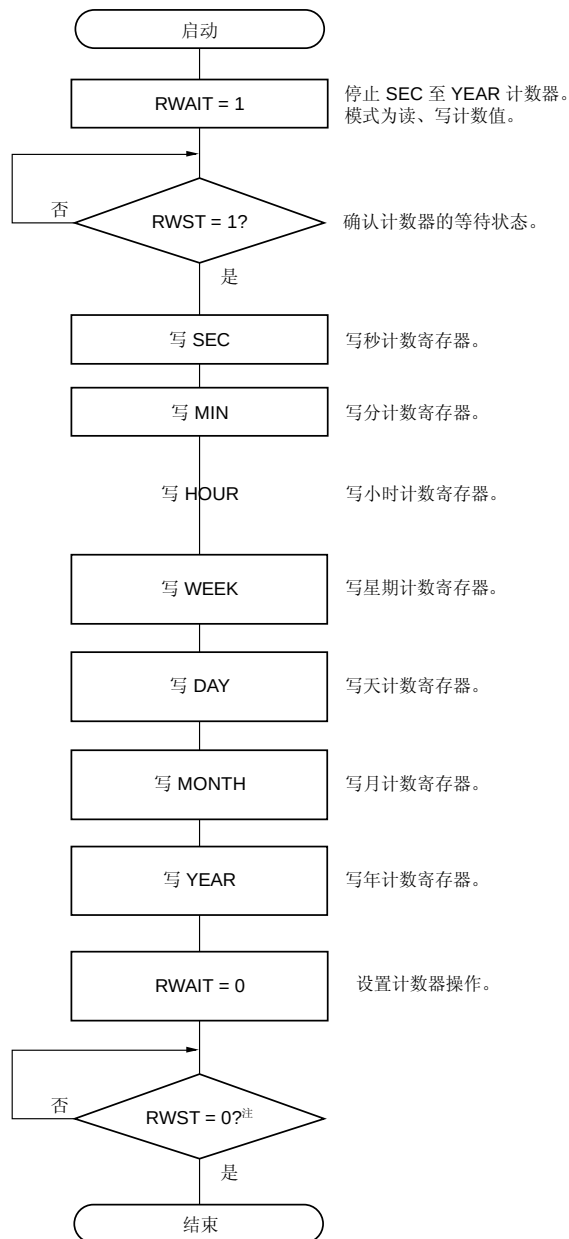
图 7-19. 在设置 RTCE 为 1 后切换至 STOP 模式的步骤



7.4.3 读/写实时计数器

在设置 RWAIT 为 1 后对计数器进行读写。

图 7-20. 读出实时计数器的操作流程

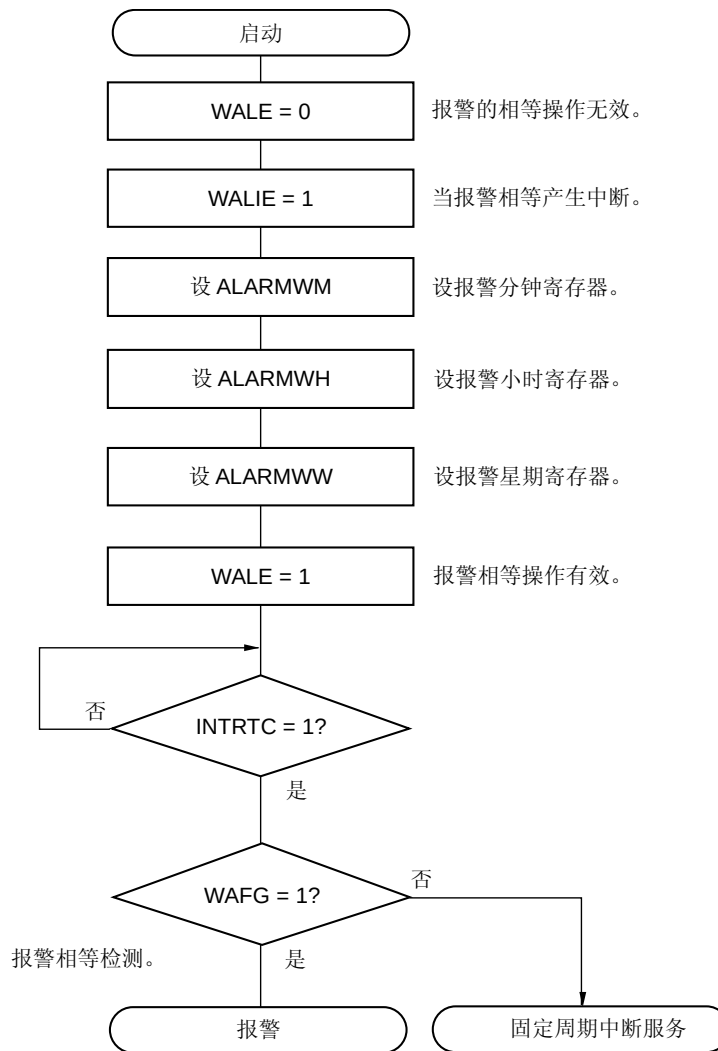


注 设置STOP 模式前必须置RWST = 0。

注意事项 在 1 秒内完成置 RWAIT 为 1 到清 RWAIT 为 0 的连续操作。

备注 SEC、MIN、HOUR、WEEK、DAY、MONTH 和 YEAR 将顺次读出。
不用设置所有的寄存器，仅部分寄存器可读出。

图 7-21. 写入实时计数器的操作流程



注 设置STOP 模式前必须置RWST = 0。

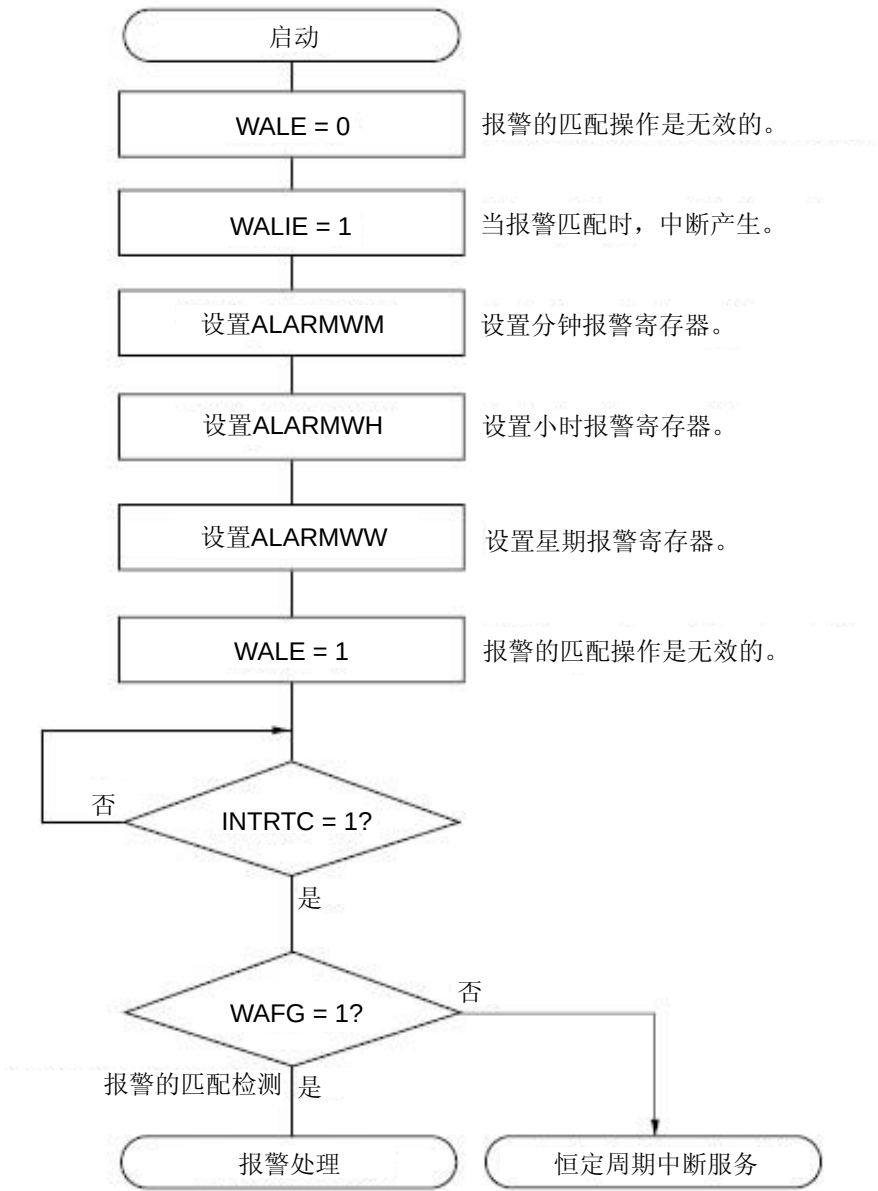
注意事项 在 1 秒内完成置 RWAIT 为 1 到清 RWAIT 为 0 的连续操作。

备注 SEC、MIN、HOUR、WEEK、DAY、MONTH 和 YEAR 将顺次写入。
不用设置所有的寄存器，仅部分寄存器可写入。

7.4.4 设置实时计数器报警

首先设置 WALE 为 0，然后设置报警时间。

图 7-22. 报警设置流程图

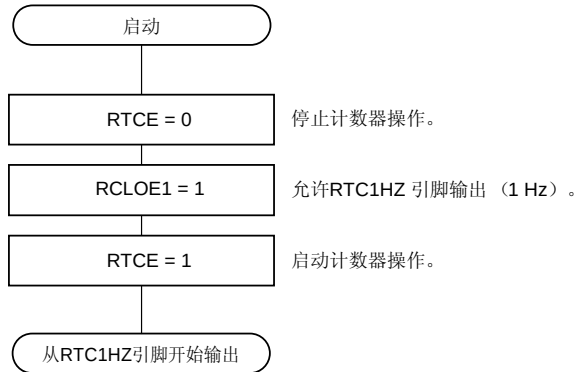


备注

1. 可以以任意顺序写入 ALARMWM、ALARMWH 和 ALARMWW。
2. 固定周期中断和报警匹配中断使用相同的中断源 (INTRTC)。当同时使用这两种类型中断时，在 INTRTC 产生的基础上，通过检查固定周期中断状态标志(RIFG)和报警检测状态标志(WAFG)，来判断产生了哪种中断。

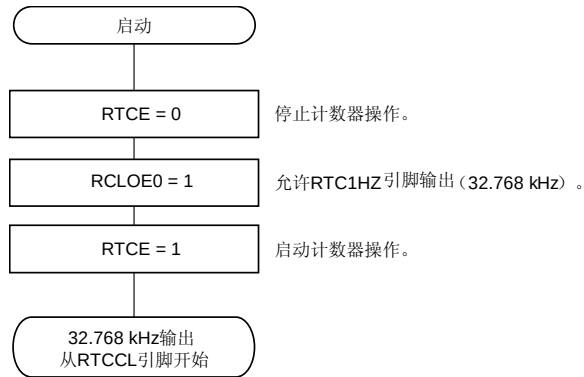
7.4.5 实时计数器的 1 Hz 输出

图 7-23. 1 Hz 输出的设置步骤



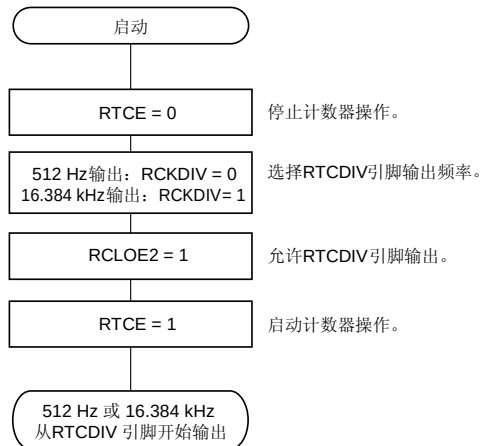
7.4.6 实时计数器的 32.768 kHz 输出

图 7-24. 32.768 kHz 输出的设置步骤



7.4.7 实时计数器的 512 Hz, 16.384 kHz 输出

图 7-25. 512 Hz, 16.384 kHz 输出的设置步骤



7.4.8 实时计数器的钟表错误修正示例

通过设置钟表错误修正寄存器的值，可在钟表变慢或变快时对其进行高精度修正。

计算修正值的示例

在修正副计数寄存器(RSUBC)的计数值时所使用的修正值通过下列表达式计算得到。

当纠错范围是-63.1 ppm 或更少或 63.1 ppm 或更多时，设置 DEV 为 0。

(当 DEV = 0)

$$\text{纠错值}^{\#} = \text{一分钟内的纠错计数数量} \div 3 = (\text{振荡频率} \div \text{目标频率} - 1) \times 32768 \times 60 \div 3$$

(当 DEV = 1)

$$\text{纠错值}^{\#} = \text{一分钟内的纠错计数数量} = (\text{振荡频率} \div \text{目标频率} - 1) \times 32768 \times 60$$

注 修正值是指由钟表错误修正寄存器 (SUBCUD)的位 6 至 0 而计算得到的钟表错误修正值。

$$(\text{当 } F6 = 0) \text{ 纠错值} = \{(F5, F4, F3, F2, F1, F0) - 1\} \times 2$$

$$(\text{当 } F6 = 1) \text{ 纠错值} = -\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2$$

当 (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *)时，不进行钟表错误修正。“*”是0或1。
/F5 至 /F0 是位取反值 (111100 时为 000011)。

备注

1. 纠错值是 2, 4, 6, 8, ... 120, 122, 124 或 -2, -4, -6, -8, ... -120, -122, -124。

2. 振荡频率为副系统时钟(f_{SUB})。

它可以从 RTCCL 引脚的 32 kHz 输出频率或 RTC1HZ 引脚× 32768 的输出频率计算得出，当时钟纠错寄存器设置为初始值时(00H)。

3. 目标频率是指钟表错误修正寄存器修正后的频率。

修正示例 <1>

从 32772.3 Hz 至 32768 Hz 的纠错示例(32772.3 Hz -131.2 ppm)

[测量振荡频率]

当钟表错误修正寄存器设置为初始值(00H)时，每个产品的振荡频率^注通过从RTCCL引脚输出约 32 kHz或从RTC1HZ引脚输出约 1 Hz进行测量。

注 参见 7.4.5 实时计数器的 1 Hz 输出，关于从 RTC1HZ 引脚输出约 1 Hz 的设定步骤；以及 7.4.6 实时计数器的 32.768 kHz 输出，关于从 RTCCL 引脚输出约 32 kHz 的设定步骤。

[计算修正值]

(当从 RTCCL 引脚的输出频率为 32772.3 Hz 时)

如果目标频率假设为 32768Hz(32772.3 Hz - 131.2 ppm)，-131.2 ppm 的纠错范围是-63.1ppm 或者更小，因此假设 DEV 为 0。

使用当 DEV 为 0 时的修正值计算表达式。

$$\begin{aligned}
 \text{纠错值} &= \text{每分钟内纠错计数的数量} \div 3 \\
 &= (\text{振荡频率} \div \text{目标频率} - 1) \times 32768 \times 60 \div 3 \\
 &= (32772.3 \div 32768 - 1) \times 32768 \times 60 \div 3 \\
 &= 86
 \end{aligned}$$

[计算设定值 (F6 至 F0)]

(当修正值为 86 时)

如果修正值大于等于 0(当变慢时)，假设 F6 为 0。

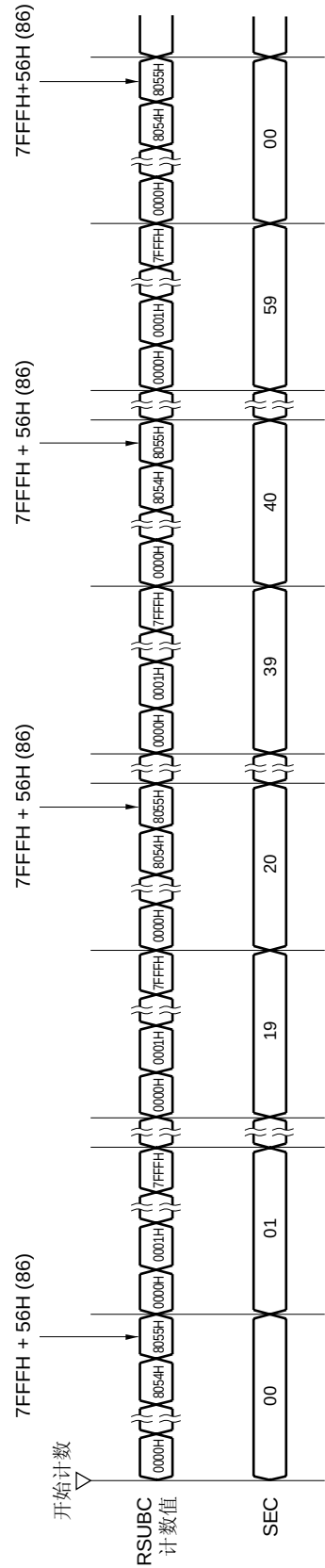
由修正值计算(F5, F4, F3, F2, F1, F0)。

$$\begin{aligned}
 \{(F5, F4, F3, F2, F1, F0) - 1\} \times 2 &= 86 \\
 (F5, F4, F3, F2, F1, F0) &= 44 \\
 (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 0, 0)
 \end{aligned}$$

从而，当纠错从 32772.3Hz 至 32768Hz(32772.3 Hz - 131.2 ppm)时，设置纠错寄存器如 DEV=0 以及纠错值是 86(SUBCUD 的位 6 至 0: 0101100)，结果是 32768 Hz (0 ppm)。

图 7-26 显示当(DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)的操作。

图 7-26 当(DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 0, 1, 1, 0, 0)时的操作



修正示例 <2>

从 32767.4 Hz 至 32768 Hz 的修正示例 (32767.4 Hz + 18.3 ppm)

[测量振荡频率]

当钟表错误修正寄存器设置为初始值(00H)时，每个产品的振荡频率^注通过从 RTCCL 引脚输出约 32 kHz 或从 RTC1HZ 引脚输出约 1 Hz 进行测量。

注 参见 7.4.5 实时计数器的 1 Hz 输出，关于从 RTC1HZ 引脚输出约 1 Hz 的设定步骤；以及 7.4.6 实时计数器的 32.768 kHz 输出，关于从 RTCCL 引脚输出约 32 kHz 的设定步骤。

[计算修正值]

(当从 RTC1HZ 引脚的输出频率为 0.9999817 Hz 时)

振荡频率 = $32768 \times 0.9999817 \approx 32767.4 \text{ Hz}$

假设目标频率为 32768 Hz (32767.4 Hz + 18.3 ppm) 且 DEV 为 1.

使用当 DEV 为 1 时的修正值计算表达式。

$$\begin{aligned} \text{修正值} &= 1 \text{ 分钟内计数的修正数量} \\ &= (\text{振荡频率} \div \text{目标频率} - 1) \times 32768 \times 60 \\ &= (32767.4 \div 32768 - 1) \times 32768 \times 60 \\ &= -36 \end{aligned}$$

[计算设定值 (F6 至 F0)]

(当纠错值是-36)

如果修正值小于等于 0(当变快时)，假设 F6 为 1。

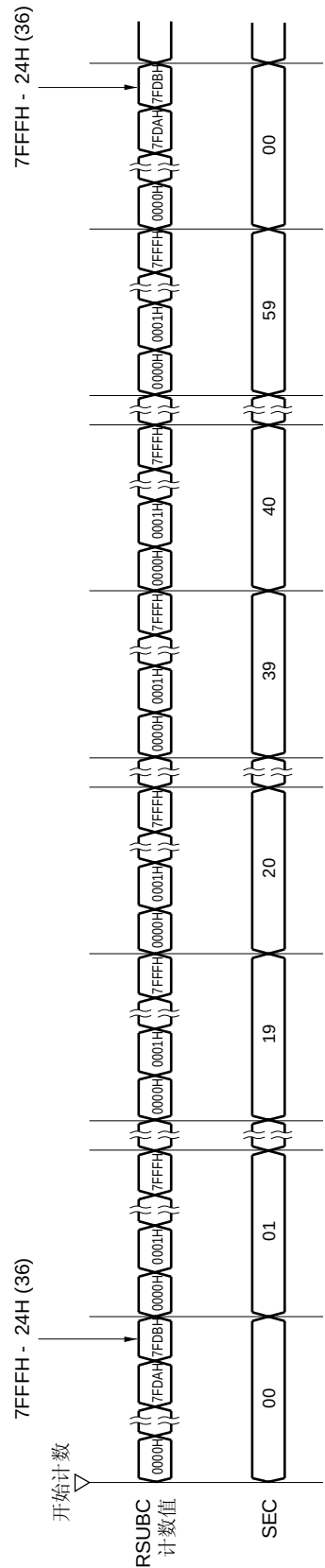
由修正值计算(F5, F4, F3, F2, F1, F0)。

$$\begin{aligned} -\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2 &= -36 \\ (/F5, /F4, /F3, /F2, /F1, /F0) &= 17 \\ (/F5, /F4, /F3, /F2, /F1, /F0) &= (0, 1, 0, 0, 0, 1) \\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 1, 0) \end{aligned}$$

从而，当纠错从 32767.4Hz 至 32768Hz(32767.4 Hz+18.3 ppm)时，设置纠错寄存器如 DEV 为 1 以及纠错值是 -36(SUBCUD 的位 6 至 1: 1101110)，结果是 32768 Hz (0 ppm)。

图 7-27 显示当(DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)的操作。

图 7-27 当(DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)时的操作



第 8 章 看门狗定时器

8.1 看门狗定时器的功能

看门狗定时器使用内部低速振荡时钟。

看门狗定时器用于检测不希望出现的程序循环。如果检测到一个程序循环，将产生一个内部复位信号。出现以下情况时检测程序循环。

- 如果看门狗定时器计数器溢出
- 如果对看门狗定时器使能寄存器(WDTE)执行位操作指令时。
- 如果“ACH”之外的数据写入 WDTE
- 如果在窗口关闭期间将数据写入 WDTE

当由看门狗定时器产生复位时，复位控制标志寄存器（RESF）的第 4 位（WDRF）置 1。RESF 的详细信息，可以参阅**第 18 章 复位功能**。

当达到溢出时间的 75%时，可产生一个间隔中断。

8.2 看门狗定时器的配置

看门狗定时器包括如下硬件。

表 8-1. 看门狗定时器的配置

项目	配置
控制寄存器	看门狗定时器使能寄存器 (WDTE)

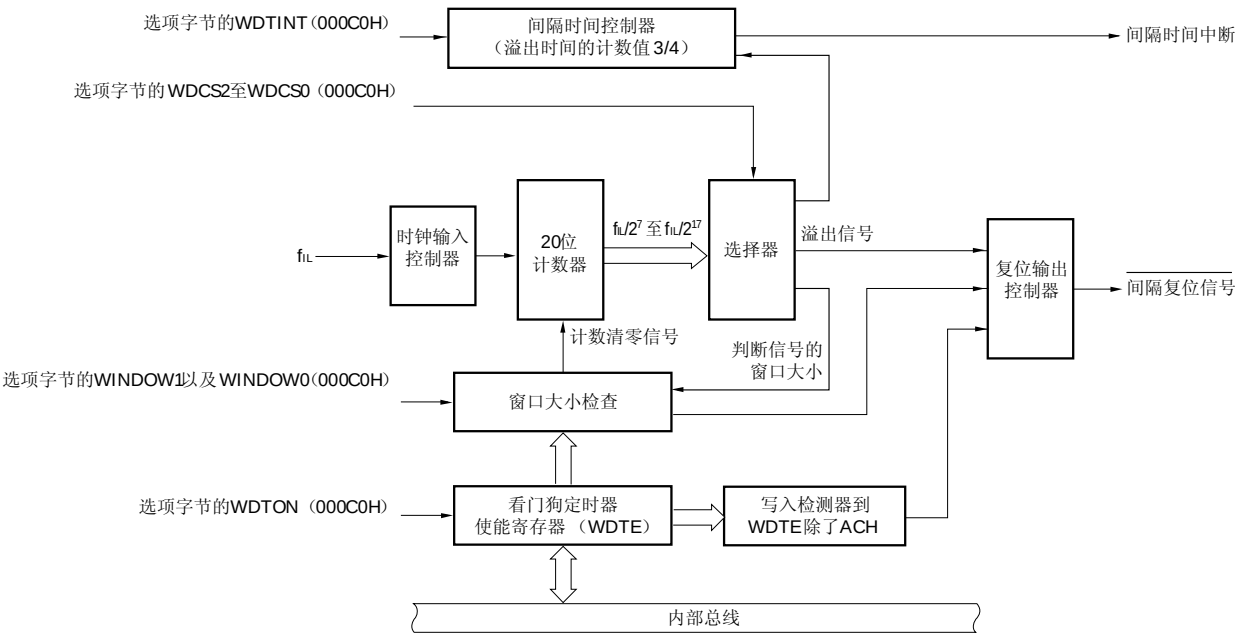
由选项字节设置如何控制计数器的操作、溢出时间和窗口打开周期。

表 8-2. 选项字节和看门狗定时器的设置

看门狗定时器的设置	选项字节 (000C0H)
看门狗定时器间隔中断	第7位(WDTINT)
窗口打开周期	第6和5位 (WINDOW1, WINDOW0)
控制看门狗定时器的计数器操作	第4位(WDTON)
看门狗定时器的溢出时间	第3 至1 位(WDCS2至WDCS0)
控制看门狗定时器的计数器操作（在HALT/STOP模式）	第0位 (WDSTBYON)

备注 关于选项字节，见第 22 章 选项字节。

图 8-1. 看门狗定时器的框图



8.3 控制看门狗定时器的寄存器

看门狗定时器由看门狗定时器使能寄存器(WDTE)控制。

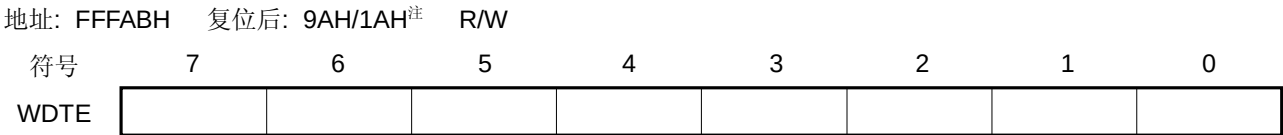
(1) 看门狗定时器使能寄存器 (WDTE)

将“ACH”写入 WDTE 可以对看门狗定时器计数器清零并再次开始计数操作。

可以由 8 位存储器操作指令设置该寄存器。

复位信号的产生将该寄存器设置为 9AH或 1AH^注。

图 8-2. 看门狗定时器使能寄存器 (WDTE)的格式



注 WDTE 的复位值随选项字节 (000C0H) 的 WDTON 位的设置值而变化。如果要对看门狗定时器进行操作，设置 WDTON = 1。

WDTON 设置值	WDTE复位值
0（禁止看门狗定时器计数的操作）	1AH
1（使能看门狗定时器计数的操作）	9AH

- 注意事项
- 1. 如果写入“WDTE”的值不是 ACH，将产生一个内部复位信号。
 - 2. 如果使用位存储器操作指令对 WDTE 进行操作，将产生一个内部复位信号。
 - 3. 从 WDTE 读取的值为 9AH/1AH（与写入值（ACH）不同）。

8.4 看门狗定时器的操作

8.4.1 看门狗定时器的控制操作

1. 当使用看门狗定时器时，由选项字节(000C0H)指定其操作。

- 通过设置选项字节（000C0H）的第 4 位（WDTON）为 1，使能看门狗定时器的计数操作（复位释放后，计数器开始计数）（如需了解详细信息，参见第 22 章）。

WDTON	看门狗定时器计数器
0	计数器操作禁止（复位后计数停止）
1	计数器操作使能（复位后计数开始）

- 通过使用选项字节（000C0H）的位 3 至 1（WDCS2 至 WDSC0），设置溢出时间（如需了解详细信息，参见 8.4.2 设置看门狗定时器的溢出时间和第 22 章）。
 - 通过使用选项字节(000C0H)的第 6 位和第 5 位(WINDOW1 和 WINDOW0)，设置窗口打开周期（如需了解详细信息，参见 8.4.3 设置看门狗定时器的窗口打开周期和第 22 章）。
2. 复位释放后，看门狗定时器开始计数。
 3. 在看门狗计时器开始计数之后和由选项字节设置的溢出时间之前写“ACH”至 WDTE，看门狗定时器清零且重新开始计数。
 4. 在上述步骤后对 WDTE 进行第二次写操作，或在复位释放后的窗口打开期间对 WDTE 进行第二次写操作。如果在窗口关闭期间对 WDTE 进行写操作，则会产生内部复位信号。
 5. 如果超出溢出时间而未写入“ACH”至 WDTE 的话，将产生一个内部复位信号。
出现以下情况时会产生内部复位信号。
 - 如果对看门狗定时器使能寄存器(WDTE)执行位操作指令时。
 - 如果“ACH”之外的数据写入 WDTE

- 注意事项**
1. 复位释放后数据第一次写入 WDTE，看门狗定时器将在任何时序下清零，与窗口打开时间无关。只要在溢出时间之前，窗口打开时间与寄存器写入时间一样长。看门狗定时器开始再次计数。
 2. 如果通过写入“ACH”至 WDTE 对看门狗定时器清零的话，实际溢出时间可能会与选项字节设置溢出时间差 $2/f_{IL}$ 秒。
 3. 在计数值溢出之前，可将看门狗定时器立即清零。

注意事项 4. 在 **HALT** 和 **STOP** 模式下看门狗定时器的操作，依据选项字节 (000C0H) 的第 0 位 (WDSTBYON) 的设置值而不同，如下所示。

	WDSTBYON = 0	WDSTBYON = 1
HALT 模式	看门狗定时器操作停止	看门狗定时器操作继续
STOP 模式		

如果 **WDSTBYON** = 0，则在 **HALT** 或 **STOP** 模式释放后，看门狗定时器恢复计数。此时，计数器清零，开始计数。

在 **STOP** 模式释放后当操作时钟为 **X1** 振荡时钟时，经过振荡稳定时间后，**CPU** 开始工作。

因此，如果从 **STOP** 模式释放到看门狗定时器溢出之间的周期比较短，在振荡稳定期间发生一个溢出，引起复位。

从而，当操作时钟为 **X1** 振荡时钟，并且看门狗定时器在由间隔中断引起的 **STOP** 模式释放后清零时，设置溢出时间要考虑振荡稳定时间。

5. 在 **Flash** 存储器自编程和 **EEPROM™** 仿真期间，看门狗定时器继续运行。处理期间，中断响应时间延迟。设置溢出时间和窗口大小时应考虑延迟。

8.4.2 看门狗定时器溢出时间的设置

通过使用选项字节(000C0H)的第 3 位至第 1 位(WDCS2 至 WDCS0)，设置看门狗定时器的溢出时间。

如果出现溢出，将产生内部复位信号。清除现有计数，看门狗定时器通过在溢出时间之前的窗口打开周期器件写入“ACH”至 **WDTE** 开始重新计数。

可设置下列溢出时间。

表 8-3. 看门狗定时器溢出时间的设置

WDCS2	WDCS1	WDCS0	看门狗定时器的溢出时间 ($f_{IL} = 33 \text{ kHz}$ (最大值))
0	0	0	$2^7/f_{IL}$ (3.88 ms)
0	0	1	$2^8/f_{IL}$ (7.76 ms)
0	1	0	$2^9/f_{IL}$ (15.52 ms)
0	1	1	$2^{10}/f_{IL}$ (31.03 ms)
1	0	0	$2^{12}/f_{IL}$ (124.12 ms)
1	0	1	$2^{14}/f_{IL}$ (496.48 ms)
1	1	0	$2^{15}/f_{IL}$ (992.97 ms)
1	1	1	$2^{17}/f_{IL}$ (3971.88 ms)

注意事项 看门狗定时器在 **Flash** 存储器自编程和 **EEPROM** 仿真期间继续操作。处理期间，中断响应时间延迟。设置溢出时间和窗口大小时应考虑延迟。

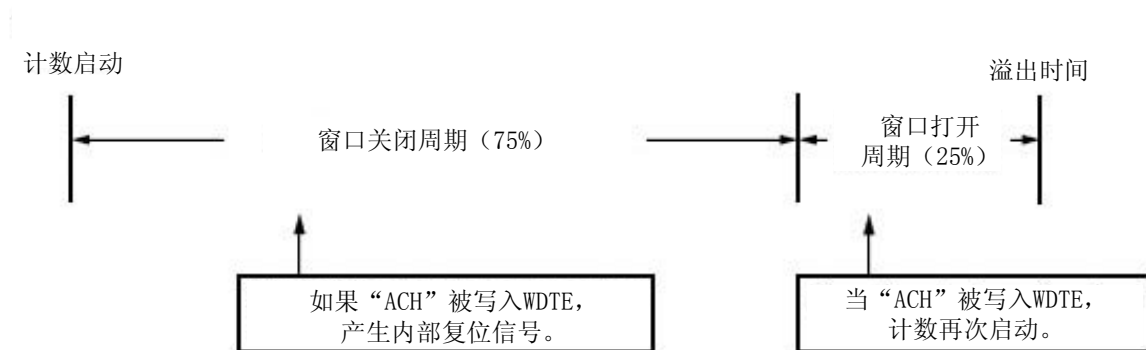
备注 f_{IL} ：内部低速振荡时钟频率

8.4.3 看门狗定时器窗口打开周期的设置

通过使用选项字节(000C0H)的第 6 位和第 5 位(WINDOW1, WINDOW0)，设置看门狗定时器的窗口打开周期。窗口使用情况描述如下。

- 如果在窗口打开周期内写入“ACH”至 WDTE，看门狗定时器清零且重新开始计数。
- 如果在窗口关闭周期内写入“ACH”至 WDTE，将检测到异常并产生内部复位信号。

示例：如果窗口打开周期变为 25%



注意事项 复位释放后数据第一次写入 WDTE，看门狗定时器将在任何时序下清零，与窗口打开时间无关。只要在溢出时间之前，窗口打开时间与寄存器写入时间一样长。看门狗定时器开始再次计数。

窗口打开周期设置如下所示。

表 8-4. 看门狗定时器窗口打开周期的设置

WINDOW1	WINDOW0	看门狗定时器的窗口打开周期
0	0	25%
0	1	50%
1	0	75%
1	1	100%

- 注意事项**
- 1.看门狗定时器在 Flash 存储器自编程和 EEPROM 仿真期间继续操作。处理期间，中断响应时间延迟。设置溢出时间和窗口大小时应考虑延迟。
 2. 当选项字节(000C0H)的第 0 位(WDSTBYON) = 0 时，无论 WINDOW1 和 WINDOW0 为何值，窗口打开周期为 100%。
 3. 如果看门狗定时器符合下面任一条件，不要设置窗口打开周期为 25%。
 - 当通过 STOP 模式或软件停止所有主系统时钟（内部高速振荡时钟、X1 时钟、和外部主系统时钟）时。
 - 低功耗模式。

备注 如果溢出时间设置为 $2^{10}/f_{IL}$ ，窗口关闭时间和打开时间如下所示。

	窗口打开周期的设置			
	25%	50%	75%	100%
窗口关闭时间	0 至 28.44 ms	0至18.96 ms	0至 9.48 ms	无
窗口打开时间	28.44至31.03 ms	18.96至31.03 ms	9.48至31.03 ms	0至31.03 ms

<当窗口打开周期为 25%>

- 溢出时间:
 $2^{10}/f_{IL}$ (最大值)= $2^{10}/33\text{ kHz}$ (最大值) = 31.03 ms
- 窗口关闭时间:
 $0\text{ 至 }2^{10}/f_{IL}$ (最小值) $\times (1 - 0.25) = 0\text{ 至 }2^{10}/27\text{ kHz}$ (最小值) $\times 0.75 = 0\text{ 至 }28.44\text{ ms}$
- 窗口打开时间:
 $2^{10}/f_{IL}$ (最小值) $\times (1 - 0.25)$ 至 $2^{10}/f_{IL}$ (最大值)= $2^{10}/27\text{ kHz}$ (最小值) $\times 0.75$ 至 $2^{10}/33\text{ kHz}$ (最大值) = 28.44 至 31.03 ms

8.4.4 看门狗定时器间隔中断的设置

根据选项字节 (000C0H)的第 7 位(WDTINT)的设置，当达到溢出时间的 75%时可产生一个间隔中断 (INTWDTI)。

表 8-5. 看门狗定时器间隔中断的设置

WDTINT	看门狗定时器间隔中断的使用
0	使用间隔中断。
1	达到溢出时间的 75% 时产生间隔中断。

注意事项 在 **STOP** 模式释放后当操作时钟为 **X1** 振荡时钟时，经过振荡稳定时间后，**CPU** 开始工作。
因此，如果从 **STOP** 模式释放到看门狗定时器溢出之间的周期比较短，在振荡稳定期间发生一个溢出，引起复位。
从而，当操作时钟为 **X1** 振荡时钟，并且看门狗定时器在由间隔中断引起的 **STOP** 模式释放后清零时，设置溢出时间要考虑振荡稳定时间。

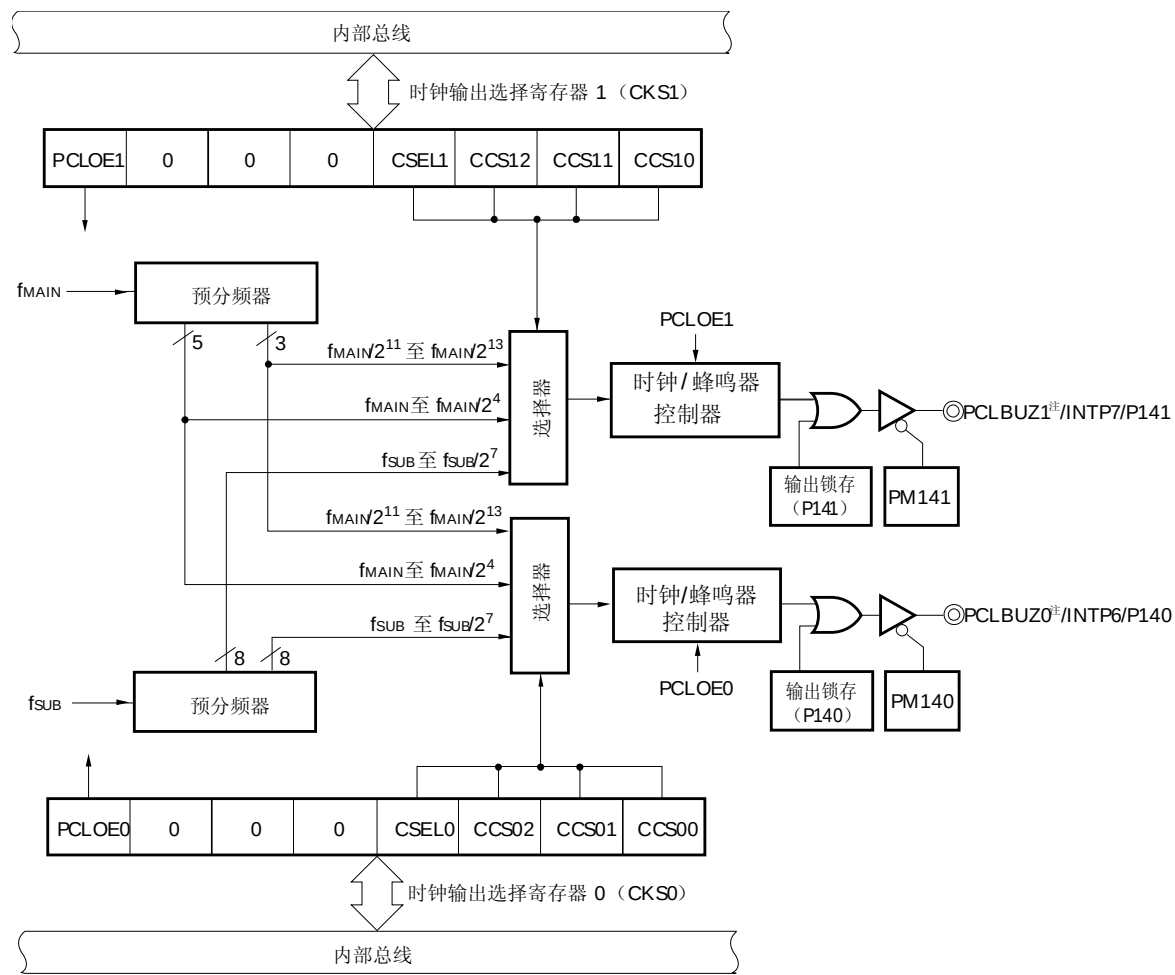
备注 即使 INTWDTI 产生后看门狗定时器继续计数（直到 ACH 写入到 WDTE 寄存器）。如果在溢出时间前 ACH 未写入到 WDTE 寄存器，产生一个内部复位信号。

第 9 章 时钟输出/蜂鸣器输出控制器

9.1 时钟输出/蜂鸣器输出控制器的功能

时钟输出控制器用于远程控制传输期间的载波输出和供给外围 ICs 的时钟输出。
蜂鸣器输出的功能是输出蜂鸣器频率的一个方波。
一个引脚可用于输出时钟或蜂鸣器声音。
两个输出引脚，PCLBUZ0 和 PCLBUZ1，可用。
PCLBUZn 输出的时钟由时钟输出选择寄存器 n(CKSn) (n = 0, 1)进行选择。
图 9-1 所示为时钟输出/ 蜂鸣器输出控制器的框图。

图 9-1. 时钟输出/蜂鸣器输出控制器的框图



注 PCLBUZ0 和 PCLBUZ1 引脚在 $2.7\text{ V} \leq V_{DD}$ 时可输出一个最高为 10 MHz 的时钟。设置时钟当 $V_{DD} < 2.7\text{ V}$ 时禁止超过 5 MHz。

备注 f_{MAIN} : 主系统时钟频率
 f_{SUB} : 副系统时钟频率

9.2 时钟输出/蜂鸣器输出控制器的配置

时钟输出/蜂鸣器输出控制器包括如下硬件。

表 9-1. 时钟输出/蜂鸣器输出控制器的配置

项目	配置
控制寄存器	时钟输出选择寄存器 0, 1 (CKS0, CKS1) 端口模式寄存器 14 (PM14) 端口寄存器 14 (P14)

9.3 控制时钟输出/蜂鸣器输出控制器的寄存器

下列两个寄存器用于控制时钟输出/蜂鸣器输出控制器。

- 时钟输出选择寄存器 0, 1 (CKS0, CSK1)
- 端口模式寄存器 14 (PM14)

(1) 时钟输出选择寄存器 0,1 (CKS0, CKS1)

这些寄存器用于对时钟输出或蜂鸣器频率输出引脚(PCLBUZ0/PCLBUZ1)设置输出使能/禁止，并设置输出时钟。
通过 CKS0 选择要从 PCLBUZ0 输出的时钟。
通过 CKS1 选择要从 PCLBUZ1 输出的时钟。
CKS0 和 CKS1 可以由 1 位或 8 位存储器操作指令设置。
复位信号的产生可将这些寄存器清零(00H)。

图 9-2. 时钟输出选择寄存器 n (CKSn) 的格式

地址: FFFA5H (CKS0), FFFA6H (CKS1) 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn 输出使能/禁止选择
0	输出禁止 (默认)
1	输出使能

CSELn	CCSn2	CCSn1	CCSn0		PCLBUZn 输出时钟选择		
					f _{MAIN} = 5 MHz	f _{MAIN} = 10 MHz	f _{MAIN} = 20 MHz
0	0	0	0	f _{MAIN}	5 MHz	10 MHz [※]	禁止设置 [※]
0	0	0	1	f _{MAIN} /2	2.5 MHz	5 MHz	10 MHz [※]
0	0	1	0	f _{MAIN} /2 ²	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	f _{MAIN} /2 ³	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	f _{MAIN} /2 ⁴	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	f _{MAIN} /2 ¹¹	2.44 kHz	4.88 kHz	9.76 kHz
0	1	1	0	f _{MAIN} /2 ¹²	1.22 kHz	2.44 kHz	4.88 kHz
0	1	1	1	f _{MAIN} /2 ¹³	610 Hz	1.22 kHz	2.44 kHz
1	0	0	0	f _{SUB}	32.768 kHz		
1	0	0	1	f _{SUB} /2	16.384 kHz		
1	0	1	0	f _{SUB} /2 ²	8.192 kHz		
1	0	1	1	f _{SUB} /2 ³	4.096 kHz		
1	1	0	0	f _{SUB} /2 ⁴	2.048 kHz		
1	1	0	1	f _{SUB} /2 ⁵	1.024 kHz		
1	1	1	0	f _{SUB} /2 ⁶	512 Hz		
1	1	1	1	f _{SUB} /2 ⁷	256 Hz		

注 在 10 MHz 范围内使用输出时钟。此外，在 V_{DD} < 2.7 V 时使用输出时钟，应在 5 MHz 范围内使用。

- 注意事项
1. 禁止时钟输出 (PCLOEn = 0) 后，改变输出时钟。
 2. 在选择主系统时钟 (CSELn = 0) 时切换至 STOP 模式的话，应在执行 STOP 指令前设置 PCLOEn = 0。当选择副系统时钟 (CSELn = 1) 时，可以设置 PCLOEn = 1，因为该时钟可在 STOP 模式下输出。

- 备注
1. n = 0, 1
 2. f_{MAIN}: 主系统时钟频率
 3. f_{SUB}: 副系统时钟频率

(2) 端口模式寄存器 14 (PM14)

此寄存器可以以 1 位单元设置端口 14 的输入/输出操作模式。
当使用 P140/INTP6/PCLBUZ0 和 P141/INTP7/PCLBUZ1 引脚作为时钟输出/ 蜂鸣器输出时，将 PM140 和 PM141 清零且 P140 和 P141 的输出锁存也清为 0。
可以由 1 位或 8 位存储器操作指令设置 PM14。
复位信号的产生将此寄存器设置为 FFH。

图 9-3. 端口模式寄存器 14 (PM14) 的格式

地址: FFF2EH	复位后: FFH	R/W						
符号	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140
PM14n	P14n 引脚 I/O 模式选择(n = 0 至 5)							
0	输出模式 (输出缓冲器开)							
1	输入模式 (输出缓冲器关)							

9.4 时钟输出/ 蜂鸣器输出控制器的操作

一个引脚可用于输出时钟或蜂鸣器声音。
两个输出引脚，PCLBUZ0 和 PCLBUZ1，可用。
PCLBUZ0 输出一个由时钟输出选择寄存器 0 (CKS0) 选择的时钟/蜂鸣器。
PCLBUZ1 输出一个由时钟输出选择寄存器 1 (CKS1) 选择的时钟/蜂鸣器。

9.4.1 作为输出引脚的操作

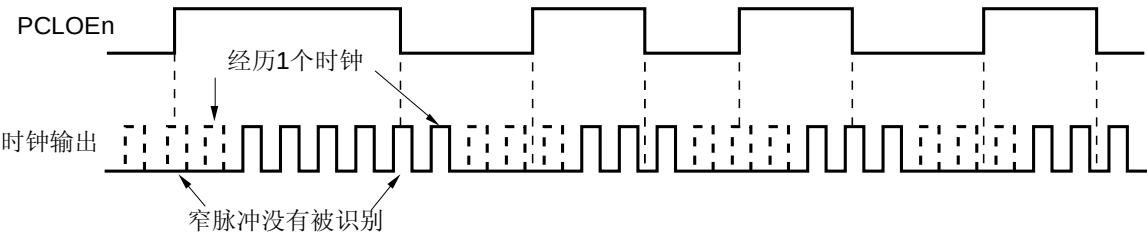
PCLBUZn 以下列过程输出。

- <1> 通过 PCLBUZn 引脚的时钟输出选择寄存器 (CKSn) 的位 0 至 3 (CCSn0 至 CCSn2, CSELn) 选择输出频率 (在禁止状态输出) 。
- <2> 设置 CKSn 的第 7 位 (PCLOEn) 为 1，使能时钟/ 蜂鸣器输出。

备注 1. 在使能或禁止时钟输出(PCLOEn)的一个时钟之后，用于输出时钟的控制权开始或停止输出时钟。此时不输出宽度窄的脉冲。图 9-4 显示了使用 PCLOEn 来使能或停止输出和输出时钟的时序。

2. n = 0, 1

图 9-4. 遥控输出应用示例



第 10 章 A/D 转换器

10.1 A/D 转换器的功能

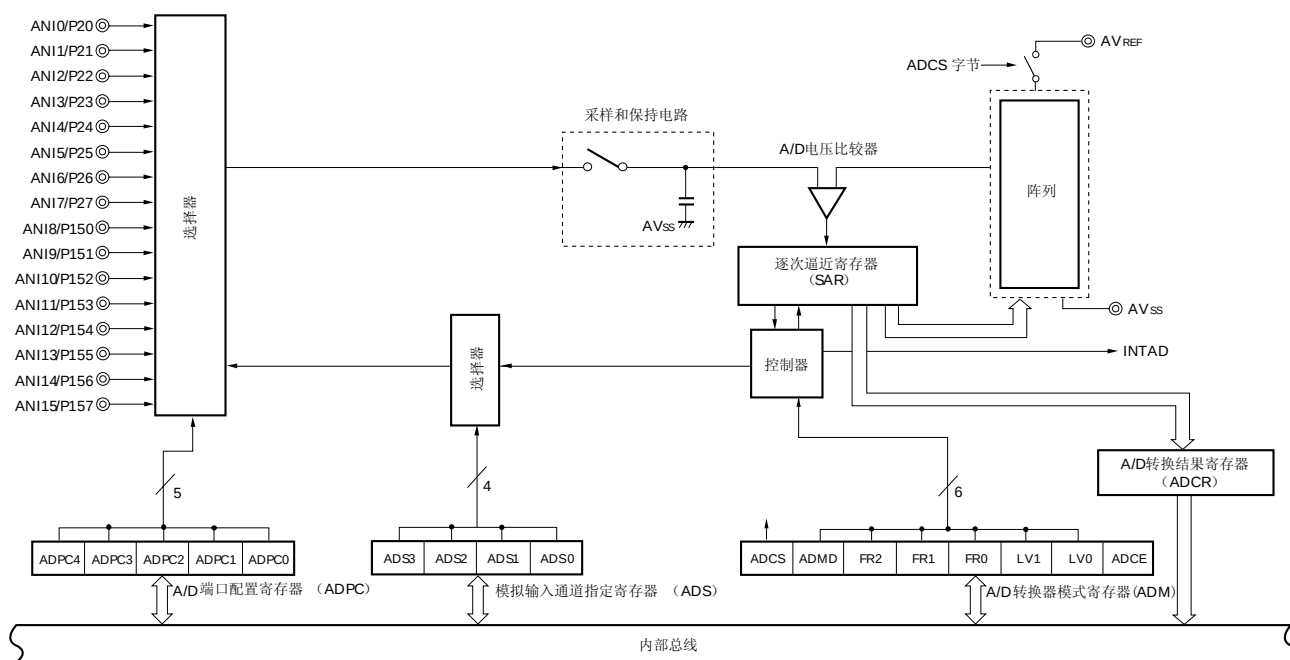
A/D 转换器将模拟输入信号转换为数值，由分辨率为 10 位的 16 个通道(ANI0 至 ANI15)组成。

A/D 转换器有如下功能。

- 10 位分辨率 A/D 转换

对于从 ANI0 至 ANI15 中选择一个模拟输入通道，重复执行 10 位分辨率 A/D 转换。每次 A/D 转换结束，都产生一个中断请求 (INTAD)。

图 10-1. A/D 转换器的框图



10.2 A/D 转换器的配置

A/D 转换器包括以下硬件。

(1) ANI0 至 ANI15 引脚

这些是 A/D 转换器的 16 个通道的模拟输入引脚。用于输入模拟信号，以便转换成数字信号。除了那些选为模拟输入的引脚外，其它引脚均可用作 I/O 端口引脚。

(2) 采样与保持电路

采样保持电路采样每个从输入电路发出的模拟输入电压，并发送到 A/D 电压比较器。此电路也在 A/D 转换期间保持已采样的模拟输入电压。

(3) A/D 电压比较器

A/D 电压比较器把从阵列中产生的电压分流和模拟输入电压进行比较。如果比较结果是模拟输入电压比参考电压 ($1/2 AV_{REF}$) 大的话，则逐次逼近寄存器(SAR)的最高有效位(MSB)置 1。如果模拟输入电压比参考电压 ($1/2 AV_{REF}$) 小的话，SAR 的 MSB 复位。

在此之后，SAR 寄存器的第 10 位自动置 1，开始下一个比较。对于已设定的结果，阵列的电压分流由第 11 位的值进行选择。

第 11 位 = 0: ($1/4 AV_{REF}$)

第 11 位 = 1: ($3/4 AV_{REF}$)

比较阵列电压分流和模拟输入电压并根据比较结果对 SAR 寄存器的第 10 位进行操作。

模拟输入电压 $\geq$ 阵列电压分流: 第 10 位 = 1

模拟输入电压 $\leq$ 阵列电压分流: 第 10 位 = 0

比较继续直到 SAR 寄存器的位 0。

(4) 阵列

该阵列产生和模拟输入引脚输入电压进行比较的比较电压。

(5) 逐次逼近寄存器 (SAR)

SAR 寄存器是 12 位寄存器，设置那些来自阵列、与模拟输入引脚的电压值相一致的电压分流数据，一次一位并从最高有效位(MSB)开始。

当电压转换成数字值,并写入最低有效位(LSB)时(A/D 转换结), SAR 寄存器(转换结果)的内容将保留在 A/D 转换结果寄存器(ADCR)中。当所有指定 A/D 转换操作完成后，产生 A/D 转换结束中断请求信号(INTAD)。

(6) 10 位 A/D 转换结果寄存器 (ADCR)

每次 A/D 转换结束，从逐次逼近寄存器中载入 A/D 转换结果，且 ADCR 寄存器在其高 10 位中保留 A/D 转换结果(低 6 位放置为 0)。

(7) 8 位 A/D 转换结果寄存器 (ADCRH)

每次 A/D 转换结束，从逐次逼近寄存器中载入 A/D 转换结果，且 ADCR 寄存器存储 A/D 转换结果的高 8 位。

(8) 控制器

该电路用于控制模拟输入信号转换成数字信号的转换时间，并启动和停止转换操作。当 A/D 转换结束时，该控制器产生中断 INTAD。

(9) AV_{REF} 引脚

该引脚输入 A/D 转换器、电源引脚和比较器的 A/D 转换器的参考电压。当端口 2 和端口 15 的所有引脚用作模拟端口引脚时，要使 AV_{REF} 电势在 $1.8\text{ V} \leq \text{AV}_{\text{REF}} \leq \text{V}_{\text{DD}}$ 。当端口 2 和端口 15 的一个或多个引脚用作数字端口引脚时，使得 AV_{REF} 和 V_{DD} 同电势。

根据 AV_{REF} 和 AV_{SS} 之间的电压，将输入到 ANI0 至 ANI10 的模拟信号转换成数字信号。

(10) AV_{SS} 引脚

这是 A/D 转换器的信号地引脚。即使不使用 A/D 转换器时，也保持此引脚的电压与 V_{SS} 引脚一致。

10.3 A/D 转换器使用的寄存器

A/D 转换器使用下面七个寄存器。

- 外围使能寄存器 0 (PER0)
- A/D 转换器模式寄存器 (ADM)
- A/D 端口配置寄存器 (ADPC)
- 模式输入通道指定寄存器(ADS)
- 端口模式寄存器 2，15(PM2 和 PM15)
- 10 位 A/D 转换结果寄存器 (ADCR)
- 8 位 A/D 转换结果寄存器 (ADCRH)

(1) 外围使能寄存器 0 (PER0)

PER0 用于使能或者禁止使用每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少消耗和噪声。
使用 A/D 转换器时，确保设置此寄存器的第 5 位 (ADCEN)为 1。
可以通过 1 位或者 8 位存储器操作指令来设置 PER0。
复位后这个寄存器的值为 00H。

图 10-2. 外围使能寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

ADCEN	A/D转换器输入时钟的控制
0	停止输入时钟的供给 • A/D转换器使用的SFR是不可重写。 • A/D转换器在复位状态。
1	供给输入时钟 • A/D转换器使用的SFR是可读/写的。

- 注意事项
1.

设置 A/D 转换器时，首先务必设置 ADCEN 为 1。如果 ADCEN = 0，忽略写入到 A/D 转换器的控制寄存器，即使读取寄存器，只能读到默认值。
2.

确保将位 6 清零。

(2) A/D 转换器模式寄存器 (ADM)

该寄存器设置模拟输入的 A/D 转换时间，并启动/停止转换。
ADM 可由 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 00H。

图 10-3. A/D 转换器模式寄存器 (ADM)的格式

地址: FFF30H 复位后: 00H R/W

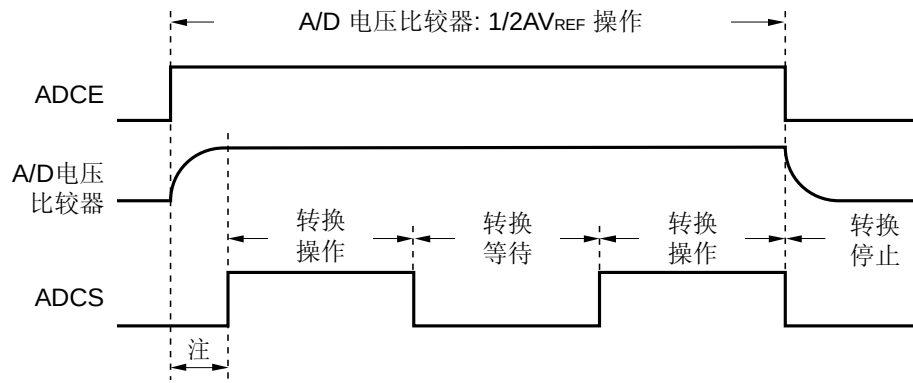
符号	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	ADMD	FR2 ^{注 1}	FR1 ^{注 1}	FR0 ^{注 1}	LV1 ^{注 1}	LV0 ^{注 1}	ADCE
ADCS		A/D转换操作控制						
0		停止转换操作						
1		允许转换操作						
ADMD		A/D转换操作模式指定						
0		选择模式						
1		扫描模式						
ADCE		A/D 电压比较器操作控制 ^{注 2}						
0		停止A/D电压比较器操作						
1		允许A/D电压比较器操作（A/D电压比较器: 1/2AV _{REF} 操作）						

- 注
- 1. FR2 至 FR0、LV1、LV0 和 A/D 转换的详细情况，参见表 10-2 A/D 转换时间选择。
 - 2. A/D 电压比较器的运行由 ADCS 和 ADCE 控制，并且从运行开始到运行稳定的时间为 1 μ s。因此，在 ADCE 设为 1 后至少等待 1 μ s 再设置 ADCS 为 1，这样从第一个转换得到的结果是有效的。否则，忽略第一次转换的数据。

表 10-1. ADCS 和 ADCE 的设置

ADCS	ADCE	A/D 转换操作
0	0	停止状态(不存在直流功耗)
0	1	转换等待模式(A/D 电压比较器: 1/2AV _{REF} 操作，只有比较器产生功耗)
1	0	设置禁止
1	1	转换模式(A/D 电压比较器: 1/2AV _{REF} 操作)

图 10-4. 使用 A/D 电压比较器时的时序图



注 从 ADCE 位上升到 ADCS 位上升所需时间必须至少为 $1\ \mu\text{s}$ ，以稳定内部电路。

注意事项 在以不同的数据重写 FR0 至 FR2 位、LV1 和 LV0 位之前，A/D 转换必须停止。

表 10-2. A/D 转换时间选择 (1/3)

(1) $4.0V \leq AV_{REF} \leq 5.5 V$

A/D转换器模式寄存器 (ADM)					模式	转换时间选择				转换时钟 (f _{AD})
FR2	FR1	FR0	LV1	LV0		f _{CLK} = 2 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz	
0	0	0	0	0	居中	设置禁止	设置禁止	34.2 μs	17.1 μs	f _{CLK} /20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f _{CLK} /10
0	1	0					27.6 μs	13.8 μs	6.9 μs	f _{CLK} /8
0	1	1				52.0 μs	20.8 μs	10.4 μs	5.2 μs	f _{CLK} /6
1	0	0				35.0 μs	14.0 μs	7.0 μs	设置禁止	f _{CLK} /4
1	0	1				26.5 μs	10.6 μs	5.3 μs		f _{CLK} /3
1	1	0				18.0 μs	7.2 μs	设置禁止		f _{CLK} /2
1	1	1				9.5 μs	设置禁止			f _{CLK}
×	×	×	0	1	增压	设置禁止				—
0	0	0	1	0	高速1	设置禁止	64.4 μs	32.2 μs	16.1 μs	f _{CLK} /20
0	0	1					32.4 μs	16.2 μs	8.1 μs	f _{CLK} /10
0	1	0				65.0 μs	26.0 μs	13.0 μs	6.5 μs	f _{CLK} /8
0	1	1				49.0 μs	19.6 μs	9.8 μs	4.9 μs	f _{CLK} /6
1	0	0				33.0 μs	13.2 μs	6.6 μs	3.3 μs	f _{CLK} /4
1	0	1				25.0 μs	10.0 μs	5.0 μs	2.5 μs	f _{CLK} /3
1	1	0				17.0 μs	6.8 μs	3.4 μs	设置禁止	f _{CLK} /2
1	1	1				9.0 μs	3.6 μs	设置禁止		f _{CLK}
0	0	0	1	1	高速2	设置禁止	设置禁止	34.2 μs	17.1 μs	f _{CLK} /20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f _{CLK} /10
0	1	0					27.6 μs	13.8 μs	6.9 μs	f _{CLK} /8
0	1	1				52.0 μs	20.8 μs	10.4 μs	5.2 μs	f _{CLK} /6
1	0	0				35.0 μs	14.0 μs	7.0 μs	3.5 μs	f _{CLK} /4
1	0	1				26.5 μs	10.6 μs	5.3 μs	设置禁止	f _{CLK} /3
1	1	0				18.0 μs	7.2 μs	3.6 μs		f _{CLK} /2
1	1	1				9.5 μs	3.8 μs	设置禁止		f _{CLK}

注意事项 1. 在用不同的数据重写 FR0 至 FR2 位、LV1 和 LV0 位之前，A/D 转换必须停止(ADCS = 0)。

2. 上述的转换时间不包括时钟频率误差。在选择转换时间时，要将时钟频率误差考虑进去。

备注 f_{CLK}: CPU/外围硬件时钟频率

表 10-2. A/D 转换时间选择 (2/3)

(2) $2.7V \leq AV_{REF} \leq 5.5 V$

A/D转换器模式寄存器 (ADM)					模式	转换时间选择				转换时钟 (f _{AD})
FR2	FR1	FR0	LV1	LV0		f _{CLK} = 2 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz	
0	0	0	0	0	标准	设置禁止	设置禁止	34.2 μs	17.1 μs	f _{CLK} /20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f _{CLK} /10
0	1	0				52.0 μs	27.6 μs	13.8 μs	设置禁止	f _{CLK} /8
0	1	1					20.8 μs	10.4 μs		f _{CLK} /6
1	0	0				35.0 μs	14.0 μs	设置禁止		f _{CLK} /4
1	0	1				26.5 μs	10.6 μs			f _{CLK} /3
1	1	0				18.0 μs	设置禁止	f _{CLK} /2		
1	1	1				9.5 μs			f _{CLK}	
×	×	×	0	1	增压	设置禁止				—
×	×	×	1	0	高速1	设置禁止				—
0	0	0	1	1	高速2	设置禁止	设置禁止	34.2 μs	17.1 μs	f _{CLK} /20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f _{CLK} /10
0	1	0				52.0 μs	27.6 μs	13.8 μs	6.9 μs	f _{CLK} /8
0	1	1					20.8 μs	10.4 μs	5.2 μs	f _{CLK} /6
1	0	0				35.0 μs	14.0 μs	7.0 μs	3.5 μs	f _{CLK} /4
1	0	1				26.5 μs	10.6 μs	5.3 μs	设置禁止	f _{CLK} /3
1	1	0				18.0 μs	7.2 μs	3.6 μs		f _{CLK} /2
1	1	1				9.5 μs	3.8 μs	设置禁止		f _{CLK}

注意事项 1. 在以不同的数据重写 FR0 至 FR2 位、LV1 和 LV0 位之前，A/D 转换必须停止(ADCS = 0)。
 2. 上述的转换时间不包括时钟频率误差。在选择转换时间时，要将时钟频率误差考虑进去。

备注 f_{CLK}: CPU/外围硬件时钟频率

表 10-2. A/D 转换时间选择 (3/3)

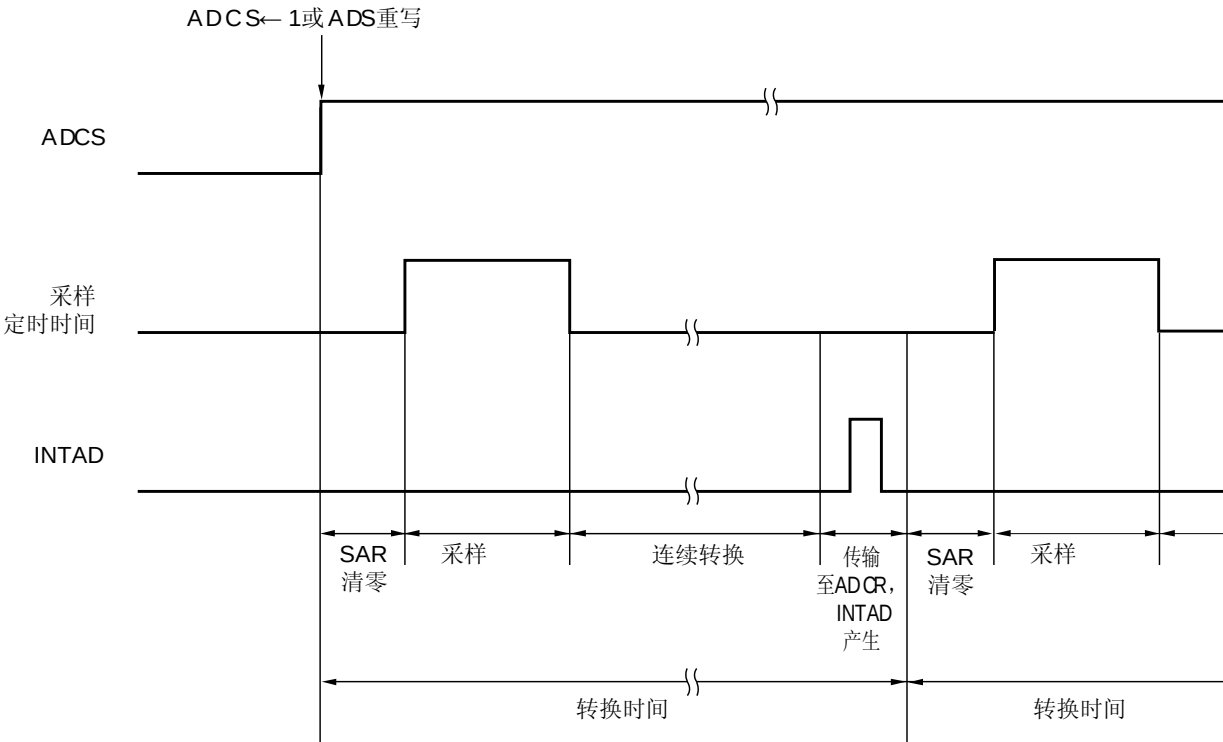
(3) 1.8 V ≤ AVREF ≤ 5.5 V

A/D转换器模式寄存器 (ADM)					模式	转换时间选择				转换时钟
FR2	FR1	FR0	LV1	LV0		f _{CLK} = 2 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz	(fAD)
×	×	×	0	0	标准	设置禁止				
0	0	0	1	1	增压	设置禁止	设置禁止	48.2 μs	24.1 μs	f _{CLK} /20
0	0	1					48.4 μs	24.2 μs	设置禁止	f _{CLK} /10
0	1	0					38.8 μs	设置禁止		f _{CLK} /8
0	1	1					29.2 μs			f _{CLK} /6
1	0	0				49.0 μs	设置禁止	f _{CLK} /4		
1	0	1				37.0 μs		f _{CLK} /3		
1	1	0				25.0 μs		f _{CLK} /2		
1	1	1				设置禁止		f _{CLK}		
×	×	×	1	0	高速1	设置禁止				—
×	×	×	1	1	高速2	设置禁止				—
其它值					设置禁止					

注意事项 1. 在以不同的数据重写 FR0 至 FR2 位、LV1 和 LV0 位之前，A/D 转换必须停止(ADCS = 0)。
2. 上述的转换时间不包括时钟频率误差。在选择转换时间时，要将时钟频率误差考虑进去。

备注 fCLK：CPU/外围硬件时钟频率

图 10-5. A/D 转换器采样和 A/D 转换时序



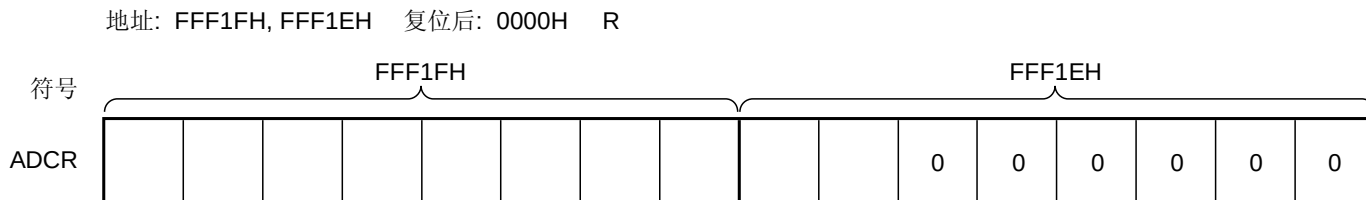
(3) 10 位 A/D 转换结果寄存器 (ADCR)

这是一个 16 位寄存器，在选择模式下用来存储 A/D 转换结果。寄存器的低 6 位恒为 0。每次 A/D 转换结束，从逐次逼近寄存器中载入转换结果。转换结果的高 8 位存储在 FFF1FH 中，而转换结果的低 2 位存储在 FFF1EH 的高 2 位中。

ADCR 可由 16 位存储器操作指令读取。

复位后这个寄存器的值为 0000H。

图 10-6. 10 位 A/D 转换结果寄存器(ADCR)的格式



注意事项 在对 A/D 转换模式寄存器(ADM)、模拟输入通道选择寄存器(ADS)和 A/D 端口配置寄存器(ADPC)赋值时，ADCR 的内容可能是不确定的。在对 ADM、ADS 和 ADPC 进行写操作前且转换结束后读取转换结果。如果没有使用上述时序，可能会读取到不正确的转换结果。

(4) 8 位 A/D 转换结果寄存器 (ADCRH)

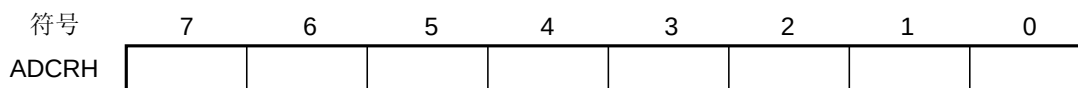
这是一个 8 位寄存器，用来存储 A/D 转换结果。可以存储 10 位分辨率的高 8 位。

ADCRH 可由 8 位存储器操作指令读取。

复位后这个寄存器的值为 00H。

图 10-7. 8 位 A/D 转换结果寄存器(ADCRH)的格式

地址: FFF1FH 复位后: 00H R



注意事项 在对 A/D 转换模式寄存器(ADM)、模拟输入通道选择寄存器(ADS)和 A/D 端口配置寄存器(ADPC)赋值时，ADCRH 的内容可能是不确定的。在对 ADM、ADS 和 ADPC 进行写操作前且转换结束后读取转换结果。如果没有使用上述时序，可能会读取到不正确的转换结果。

(5) 模式输入通道指定寄存器(ADS)

该寄存器用来选择 A/D 转换的模拟电压的输入通道。

ADS 可由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 10-8. 模拟输入通道指定寄存器 (ADS) (1/2)的格式

地址: FFF31H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

○ 选择模式 (ADMD = 0)

ADS3	ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	0	ANI0	P20/ANI0 引脚
0	0	0	1	ANI1	P21/ANI1 引脚
0	0	1	0	ANI2	P22/ANI2 引脚
0	0	1	1	ANI3	P23/ANI3 引脚
0	1	0	0	ANI4	P24/ANI4 引脚
0	1	0	1	ANI5	P25/ANI5 引脚
0	1	1	0	ANI6	P26/ANI6 引脚
0	1	1	1	ANI7	P27/ANI7 引脚
1	0	0	0	ANI8	P150/ANI8 引脚
1	0	0	1	ANI9	P151/ANI9 引脚
1	0	1	0	ANI10	P152/ANI10 引脚
1	0	1	1	ANI11	P153/ANI11 引脚
1	1	0	0	ANI12	P154/ANI12 引脚
1	1	0	1	ANI13	P155/ANI13 引脚
1	1	1	0	ANI14	P156/ANI14 引脚
1	1	1	1	ANI15	P157/ANI15 引脚

- 注意事项**
1. 确保将位 4 至 7 清“0”。
 2. 利用端口模式寄存器 2 和 15 (PM2 和 PM15)，设置一个输入模式下的用于 A/D 转换的通道。
 3. 不要通过 ADS 设置已经 ADPC 设置为数字 I/O 的引脚。

图 10-8. 模拟输入通道指定寄存器 (ADS) (2/2)的格式

地址: FFF31H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

○ 选择模式 (ADMD = 1)

ADS3	ADS2	ADS1	ADS0	模拟输入通道			
				扫描 0	扫描 1	扫描 2	扫描 3
0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	1	1	0	ANI6	ANI7	ANI8	ANI9
0	1	1	1	ANI7	ANI8	ANI9	ANI10
与上述不同的值				设置禁止			

- 注意事项**
1. 确保将位 4 至 7 清“0”。
 2. 利用端口模式寄存器 2 和 15 (PM2, PM15)，设置一个输入模式下的用于 A/D 转换的通道。
 3. 不要通过 ADS 设置已经 ADPC 设置为数字 I/O 的引脚。

(6) A/D 端口配置寄存器 (ADPC)

这个寄存器用于将 ANI0/P20 至 ANI7/P27 和 ANI8/P150 至 ANI15/P157 引脚切换为 A/D 转换器的模拟输入或是端口的数字 I/O。

ADPC 可由 8 位存储器操作指令设置。

复位信号的发生设置此寄存器为 10H。

图 10-9. A/D 端口配置寄存器 (ADPC)的格式

地址: F0017H 复位后: 10H R/W

符号	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	模拟输入 (A) / 数字 I/O (D) 切换													
					端口 15								端口 2					
					ANI15 /P157	ANI14 /P156	ANI13 /P155	ANI12 /P154	ANI11 /P153	ANI10 /P152	ANI9 /P151	ANI8 /P150	ANI7 /P27	ANI6 /P26	ANI5 /P25	ANI4 /P24	ANI3 /P23	ANI2 /P22
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	A	A	A	D
0	0	0	1	1	A	A	A	A	A	A	A	A	A	A	A	A	D	D
0	0	1	0	0	A	A	A	A	A	A	A	A	A	A	A	D	D	D
0	0	1	0	1	A	A	A	A	A	A	A	A	A	A	D	D	D	D
0	0	1	1	0	A	A	A	A	A	A	A	A	A	D	D	D	D	D
0	0	1	1	1	A	A	A	A	A	A	A	A	D	D	D	D	D	D
0	1	0	0	0	A	A	A	A	A	A	A	A	D	D	D	D	D	D
0	1	0	0	1	A	A	A	A	A	A	A	D	D	D	D	D	D	D
0	1	0	1	0	A	A	A	A	A	A	D	D	D	D	D	D	D	D
0	1	0	1	1	A	A	A	A	A	D	D	D	D	D	D	D	D	D
0	1	1	0	0	A	A	A	A	D	D	D	D	D	D	D	D	D	D
0	1	1	0	1	A	A	A	D	D	D	D	D	D	D	D	D	D	D
0	1	1	1	0	A	A	D	D	D	D	D	D	D	D	D	D	D	D
0	1	1	1	1	A	D	D	D	D	D	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D
其它值					设置禁止													

- 注意事项**
1. 利用端口模式寄存器 2 和 15 (PM2, PM15)，设置一个输入模式下的用于 A/D 转换的通道。
 2. 不要通过 ADS 设置已经 ADPC 设置为数字 I/O 的引脚。

(7) 端口模式寄存器 2 和 15 (PM2, PM15)

当使用 ANI0/P20 至 ANI7/P27 和 ANI8/P150 至 ANI15/P157 引脚作为模拟输入端口时，设置 PM20 至 PM27 和 PM150 至 PM157 为 1。此时 P20 至 P27 和 P150 至 P157 的输出锁存可为 0 或 1。

如果 PM20 至 PM27 和 PM150 至 PM157 设置为 0，则其不能用作模拟输入端口引脚。

PM2 和 PM15 可由 1 位或 8 位存储器操作指令设置。

复位信号的产生将这些寄存器设置为 FFH。

注意事项 如果引脚设置为模拟输入端口的话，读取的值不是引脚电平而是始终为“0”。

图 10-10. 端口模式寄存器 2 和 15(PM2, PM5)的格式

地址：FFF22H 复位后：FFH R/W

符号	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

地址：FFF2FH 复位后：FFH R/W

符号	7	6	5	4	3	2	1	0
PM15	PM157	PM156	PM155	PM154	PM153	PM152	PM151	PM150

PMmn	Pmn 引脚 I/O 模式选择 (mn = 20 至 27, 150 至 157)
0	输出模式(输出缓冲器开)
1	输入模式(输出缓冲器关)

根据 ADPC、ADS、PM2 和 PM15 的设置，ANI0/P20 至 ANI7/P27 和 ANI8/P150 至 ANI15/P157 引脚如下所示。

表 10-3 ANI0/P20 至 ANI7/P27 和 ANI8/P150 至 ANI15/P157 引脚的设置功能

ADPC	PM2和PM15	ADS	ANI0/P20至ANI7/P27和 ANI8/P150至ANI15/P157引脚
数字 I/O 选择	输入模式	—	数字输入
	输出模式	—	数字输出
模拟输入选择	输入模式	选择ANI。	模拟输入(转换)
		不要选择ANI。	模拟输入(不转换)
	输出模式	选择ANI。	设置禁止
		不要选择ANI。	

10.4 A/D 转换器操作

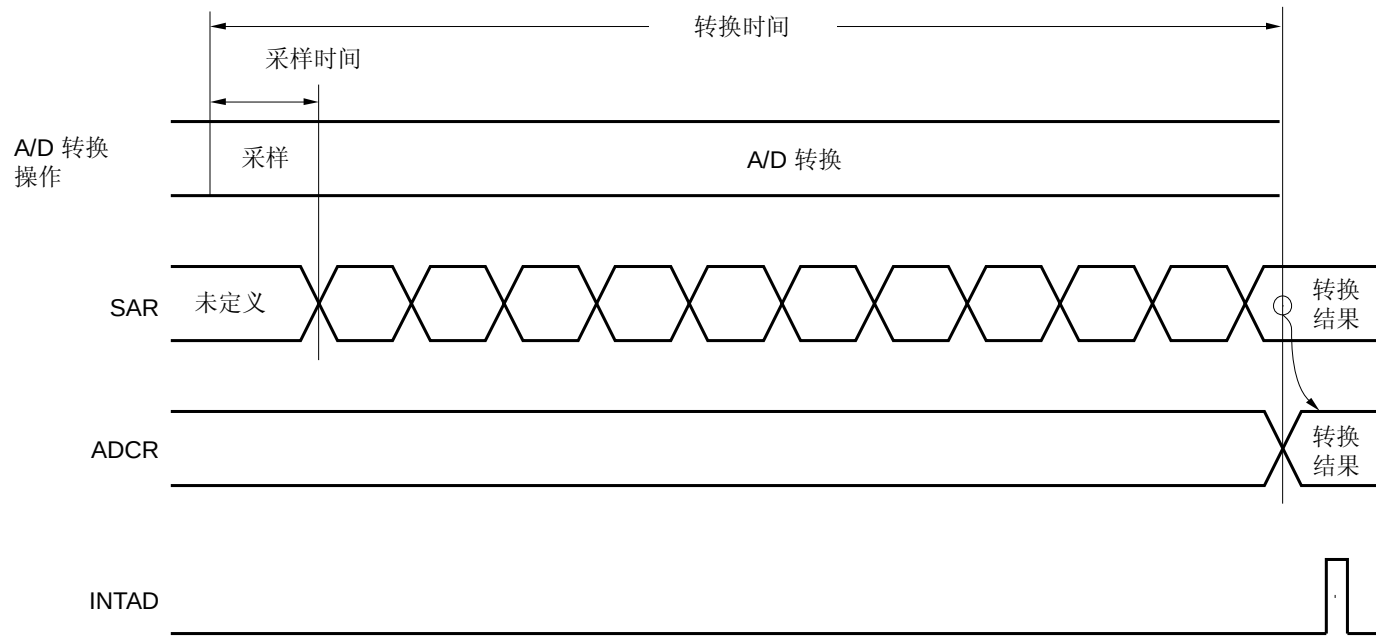
10.4.1 A/D 转换器的基本操作

- <1> 把外围使能寄存器 0 (PER0) 的第 5 位 (ADCEN) 置 1 来启动输入时钟的供给到 A/D 转换器。
- <2> 通过 ADM 的第 5 位至第 1 位 (FR2 至 FR0、LV1 和 LV0) 设置 A/D 转换时间, 以及 ADM 的第 6 位 (ADMD) 来设置运行模式。
- <3> 把 A/D 转换模式寄存器 (ADM) 的第 0 位 (ADCE) 置 1, 启动 A/D 电压比较器操作。
- <4> 通过 A/D 端口配置寄存器 (ADPC) 设置 A/D 转换到模拟输入的通道, 并且通过端口模式寄存器 (PM2 和 PM15) 设置到输入模式。
- <5> 使用模拟输入通道指定寄存器 (ADS) 为 A/D 转换选择一个通道。
- <6> 通过设置 ADM 的第 7 位 (ADCS) 为 1 开始转换操作。
如果定时器的触发模式在步骤<7>中设置时, 定时器进入触发等待状态。
(<7>至<13> 操作由硬件执行。)
- <7> 到选择的模拟输入通道的输入电压由采样&保持电路采样。
- <8> 在经过一段时间的采样后, 采样&保持电路处于保持状态, 且在 A/D 转换操作结束前一直保持采样电压。
- <9> 设置逐次逼近寄存器 (SAR) 的第 9 位。通过分接选择器将串联电阻的分接电压置为 $(1/2)AV_{REF}$ 。
- <10> 由电压比较器比较串联电阻串的分接电压与采样电压。如果模拟输入大于 $(1/2)AV_{REF}$, 则 SAR 的 MSB 保留设置为 1。如果模拟输入小于 $(1/2)AV_{REF}$, 则 MSB 复位为 0。
- <11> 接下来, SAR 的第 8 位自动置 1, 并进入下一个比较过程。根据第 9 位的预置值选择串联电阻串的分接电压, 具体描述如下。
 - 位 9 = 1: $(3/4)AV_{REF}$
 - 位 9 = 0: $(1/4)AV_{REF}$
 比较分接电压与采样电压, 并设置 SAR 的第 8 位, 如下所示。
 - 采样电压 $\geq$ 电压分流: 位 8 = 1
 - 采样电压 $<$ 电压分流: 位 8 = 0
- <12> 按此方式继续进行比较, 直至 SAR 的第 0 位。
- <13> 全部 10 位比较完成后, 在 SAR 中保留一个有效的数值结果, 然后将结果传输至 A/D 转换结果寄存器 (ADCR, ADCRH) 中, 并锁存。
同时也会产生 A/D 转换结束中断请求 (INTAD)。
- <14> 重复步骤<7>至<13>, 直到 ADCS 清为 0。
将 ADCS 清零, 以停止 A/D 转换器操作。
当 ADCE = 1 时, 若要重新启动 A/D 转换操作, 应从步骤<6>开始。当 ADCE = 0 时, 若要再次启动 A/D 转换操作, 设置 ADCE=1, 等待至少 $1\mu s$ 然后从步骤<6>开始操作。如要改变 A/D 转换的通道, 则从步骤<5>开始。

注意事项 必须确保步骤 <3>至<6>之间的操作时间至少为 $1\mu s$ 。

- 备注 有两种类型的 A/D 转换结果寄存器可以使用。
- ADCR (16 位): 存储 10 位 A/D 转换值
 - ADCRH (8 位): 存储 8 位 A/D 转换值

图 10-11. A/D 转换器的基本操作



A/D 转换操作是连续执行的，直到用软件将 A/D 转换器模式寄存器(ADM)的第 7 位(ADCS)复位(0)。
在 A/D 转换期间，如果对模拟输入通道选择寄存器(ADS)进行写操作，则转换操作初始化，且若 ADCS 设置为 1，则转换操作重新开始。
复位信号会将 A/D 转换结果寄存器(ADCR, ADCRH)的内容设置为 0000H 或 00H。

10.4.2 输入电压和转换结果

输入到模拟输入引脚(ANIO 至 ANI5)的模拟输入电压与理论 A/D 转换结果(存储在 10 位 A/D 转换结果寄存器 (ADCR)中)之间的关系如下列表达式所示。

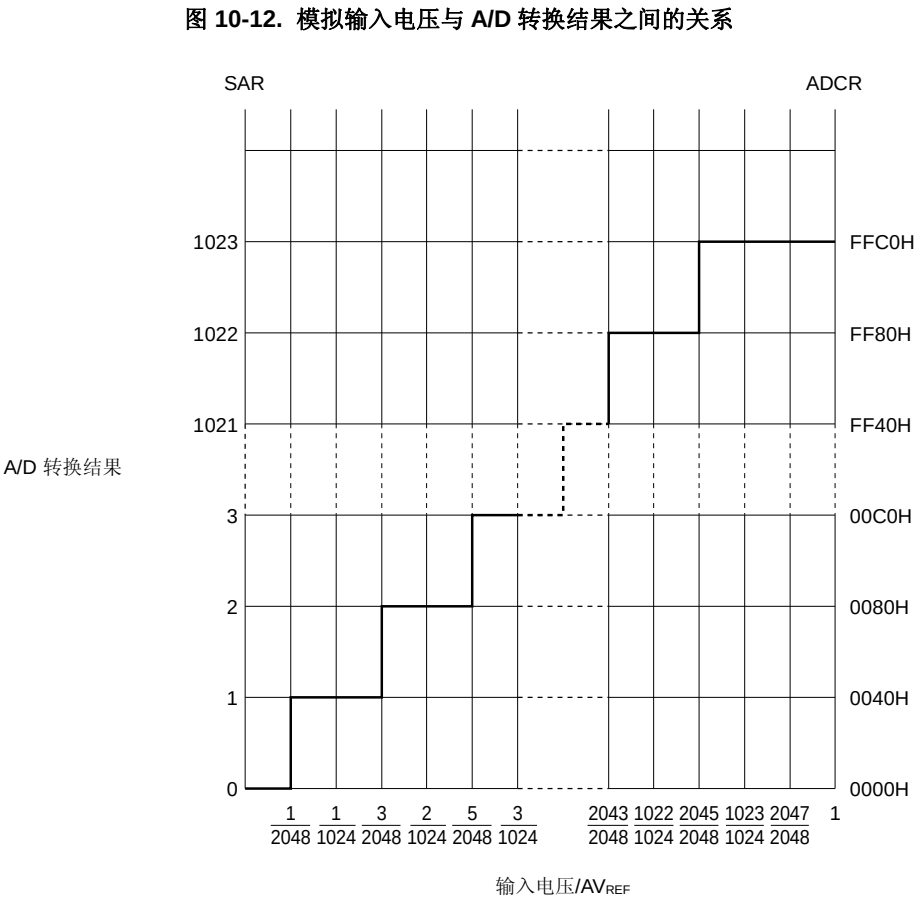
$$SAR = INT \left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5 \right)$$
$$ADCR = SAR \times 64$$

或

$$\left(\frac{ADCR}{64} - 0.5 \right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5 \right) \times \frac{AV_{REF}}{1024}$$

- 其中， INT(): 该函数返回括号中值的整数部分。
VAIN: 模拟输入电压
AVREF: AVREF 引脚电压
ADCR: A/D 转换结果寄存器(ADCR)的值
SAR: 逐次逼近寄存器

图 10-12 显示模拟输入电压与 A/D 转换结果之间的关系。



10.4.3 A/D 转换器操作模式

选择模式和扫描模式提供作为 A/D 转换器的操作模式。

(1) 选择模式

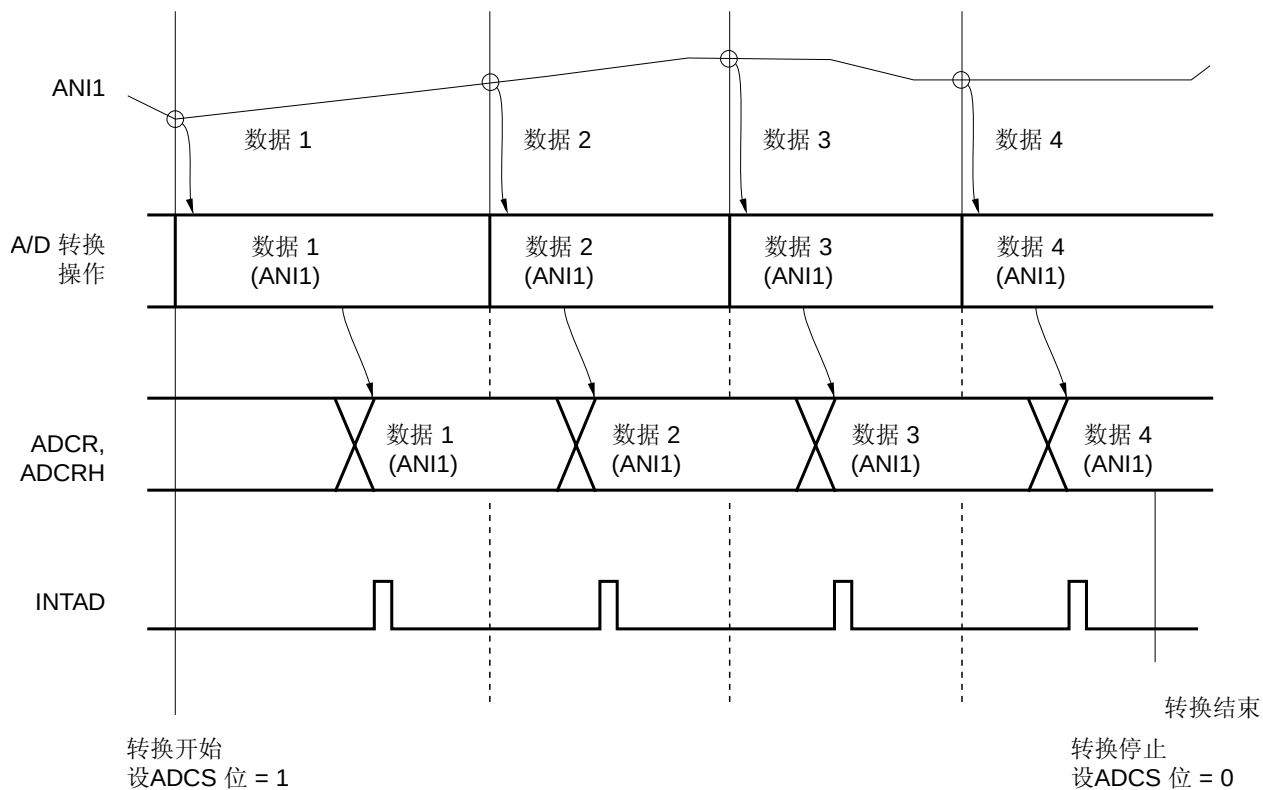
由模拟输入通道选择寄存器(ADS)指定的一个模拟输入，当 A/D 转换器模式寄存器(ADM)的 ADMD 位为 0 时，进行 A/D 转换。

当 A/D 转换完成时，转换结果存于 A/D 转换结果寄存器(ADCR)中，并产生一个 A/D 转换结束中断请求信号(INTAD)。

A/D 转换完成后，除非 ADCS 位设置为 0，否则 A/D 转换会连续的重复执行。

在转换中如果 ADM 或 ADS 寄存器写入，A/D 转换将中止。在这种情况下，A/D 转换将重新开始。

图 10-13. 选择模式运行时序的示例



(2) 扫描模式

扫描 0 至 3 的四个模拟输入通道，由模拟输入通道选择寄存器(ADS)指定。当 A/D 转换器模式寄存器(ADM)的 ADMD 位为 1 时，将先后进行 A/D 转换。A/D 转换按顺序进行，从由扫描 0 指定的模拟输入通道开始。

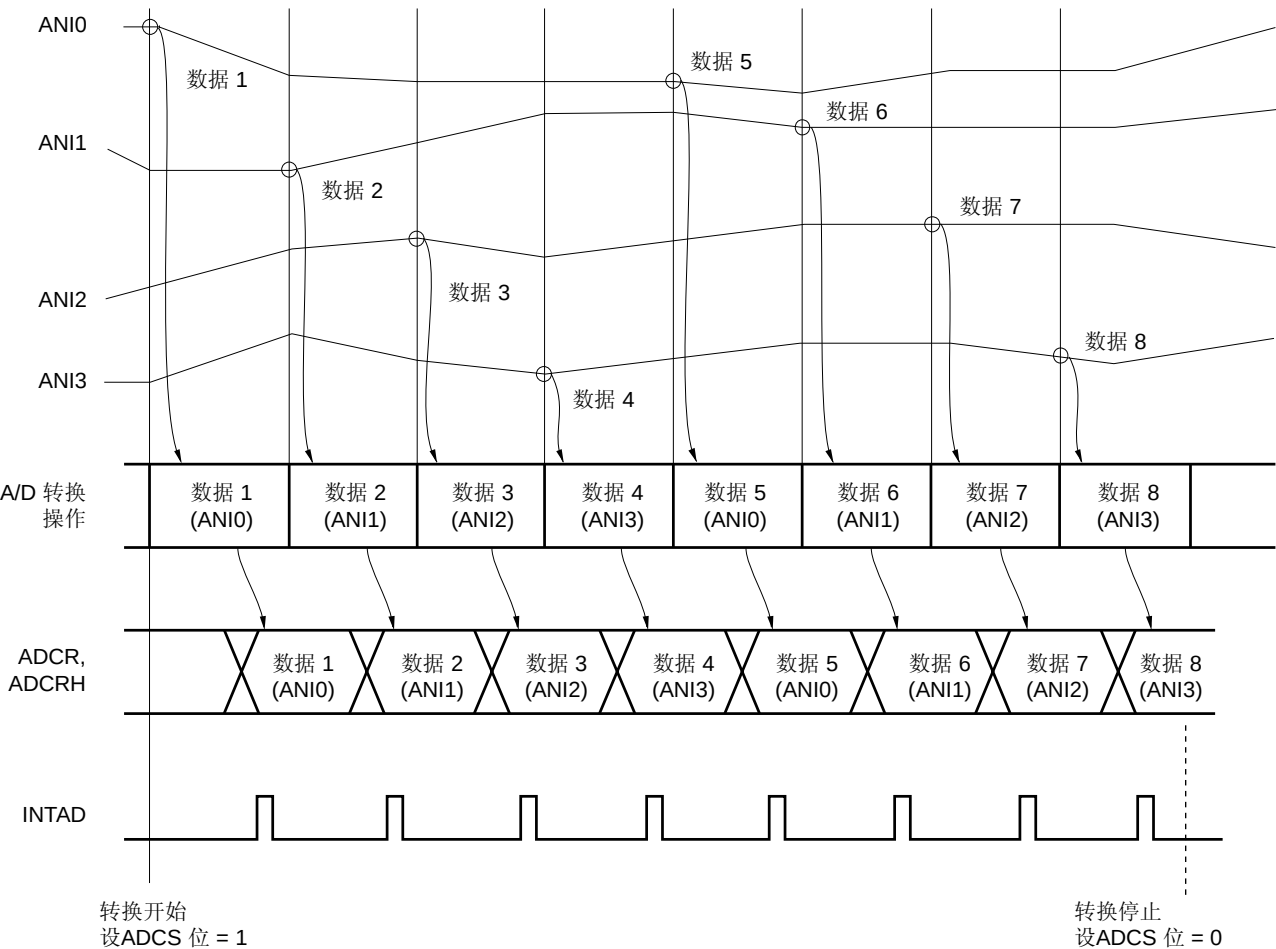
当一个模拟输入的 A/D 转换完成时，转换结果存于 A/D 转换结果寄存器(ADCR)中，并产生一个 A/D 转换结束中断请求信号(INTAD)。

所有模拟输入通道的 A/D 转换结果存放在 ADCR 中。因此建议，一旦一个模拟输入通道的 A/D 转换完成后，把 ADCR 中的内容存放到 RAM 中。

当一个 A/D 转换结束时，除非 ADCS 位设为 0，否则下一个 A/D 转换马上开始。

在转换中如果 ADM 或 ADS 寄存器写入，A/D 转换将中止。在这种情况下，A/D 从扫描 0 的模拟输入通道处重新开始。

图 10-14. 扫描模式运行时序示例



设置方法如下。

- <1> 设置外围使能寄存器 0 (PER0) 的第 5 位 (ADCEN) 为 1。
- <2> 通过 ADM 的第 5 位至第 1 位 (FR2 至 FR0、LV1 和 LV0) 选择转换时间，以及 ADM 的第 6 位 (ADMD) 来选择运行模式。
- <3> 设置 A/D 转换器模式寄存器 (ADM) 的第 0 位 (ADCE) 为 1。
- <4> 利用 A/D 端口配置寄存器 (ADPC) 的位 4 至 0 (ADPC4 至 ADPC0)、端口模式寄存器 2 (PM2) 的位 7 至 0 (PM27 至 PM20) 和端口模式寄存器 15 (PM15) 的位 7 至 0 (PM157 至 PM150)，设置用于模拟输入模式的通道。
- <5> 通过模拟输入通道指定选择寄存器 (ADS) 位 3 至 0 (ADS3 至 ADS0) 选择通道。
- <6> 将 ADM 的第 7 位 (ADCS) 置 1，启动 A/D 转换。
- <7> 当 A/D 转换结束，产生一个中断请求信号 (INTAD)。
- <8> 将 A/D 转换数据传输至 A/D 转换结果寄存器 (ADCR, ADCRH) 中。
- <改变通道>
 - <9> 通过 ADS 的位 3 至 0 (ADS3 至 ADS0) 来改变通道从而启动 A/D 转换。
 - <10> 当 A/D 转换结束，产生一个中断请求信号 (INTAD)。
 - <11> 将 A/D 转换数据传输至 A/D 转换结果寄存器 (ADCR, ADCRH) 中。
- <完成 A/D 转换>
 - <12> ADCS 清为 0。
 - <13> ADCE 清为 0。
 - <14> 将外围使能寄存器 0 (PER0) 的第 5 位 (ADCEN) 清 0。

- 注意事项**
- 1. 必须确保步骤 <3> 至 <6> 之间的操作时间至少为 $1\mu\text{s}$ 。
 - 2. <3> 可以在 <4> 和 <5> 之间进行。
 - 3. <3> 可以省略。但在这种情况下，要忽略 <7> 之后的第一个转换数据。
 - 4. 步骤 <7> 至 <10> 所经历的时间不同于通过 ADM 的位 5 至 1 (FR2 至 FR0、LV1、LV0) 设置的转换时间。步骤 <9> 至 <10> 所经历的时间为 FR2 至 FR0、LV1、LV0 设置的转换时间。

10.5 A/D 转换器特性表的阅读方法

以下介绍 A/D 转换器中的专用术语。

(1) 分辨率

这是可识别的最低模拟输入电压。这是可识别的最小的模拟输入电压，即每位数字输出的模拟输入电压的百分比，称为 1LSB(最低有效位)。对于满度的 1LSB 的百分比用%FSR(满度范围)表示。

当分辨率为 10 位时 1LSB 表示如下。

$$\begin{aligned} 1\text{LSB} &= 1/2^{10} = 1/1024 \\ &= 0.098\%\text{FSR} \end{aligned}$$

精确度与分辨率无关，而由总误差决定。

(2) 总误差

总误差是指实际测量值与理论值之间的最大误差。

零度误差、满度误差、积分线性误差和微分线性误差等组合起来表示总误差。

注意量化误差不属于特征表中总误差的范围。

(3) 量化误差

当模拟值转换成数字值时，通常会产生 $\pm 1/2\text{LSB}$ 的误差。在一个 A/D 转换器中，相差 $\pm 1/2\text{LSB}$ 的模拟输入电压转换成相同的数字代码，因此量化误差不可避免。

注意量化误差不属于特征表中总误差、零度误差、满度误差、积分线性误差和微分线性误差的范围。

图 10-15. 总误差

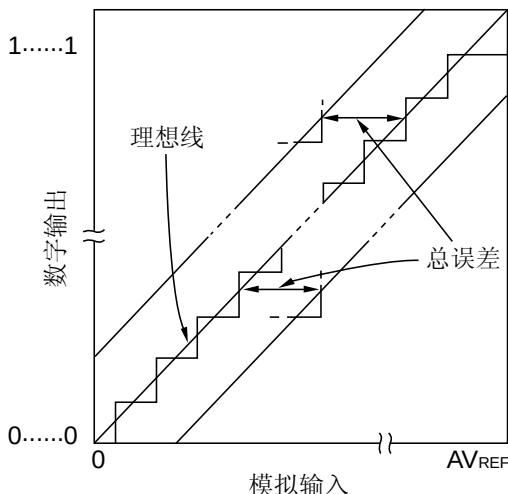
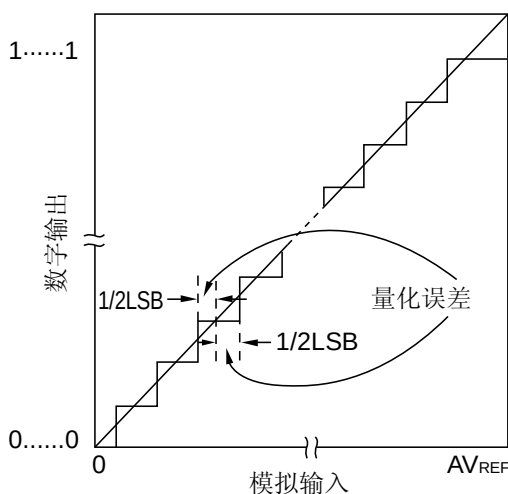


图 10-16. 量化误差



(4) 零度误差

零度误差表示当数字输出范围在 0.....000 至 0.....001 之间时，模拟输入电压的实际测量值与理论值 ($1/2\text{LSB}$) 之间的误差。

如果实际测量值大于理论值，零度误差表示当数字输出范围在 0.....001 至 0.....010 之间时，模拟输入电压的实际测量值与理论值 ($3/2\text{LSB}$) 之间的误差。

(5) 满度误差

满度误差表示当数字输出范围在 $1\ldots110$ 至 $1\ldots111$ 之间时, 模拟输入电压的实际测量值与理论值(满度 $-3/2\text{LSB}$)之间的误差。

(6) 积分线性误差

积分线性误差说明了转换特征偏离理想线性关系的程度。它表示当零度误差和满度误差均为 0 时, 实际测量值与理想直线之间误差的最大值。

(7) 微分线性误差

当代码输出的理想宽度为 1LSB 时, 微分线性误差表示实际测量值与理想值之间的差距。

图 10-17. 零度误差

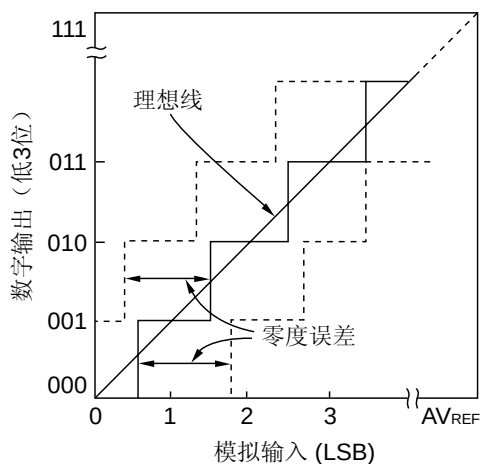


图 10-18. 满度误差

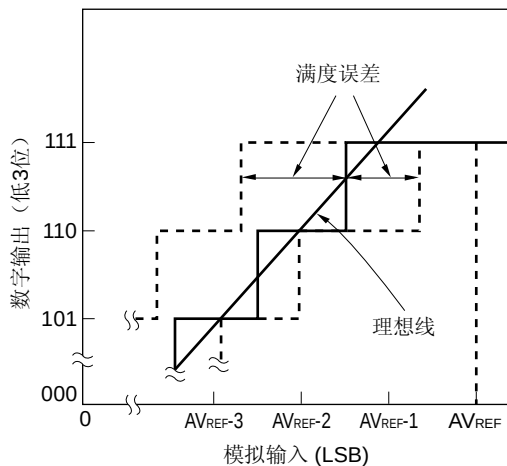


图 10-19. 积分线性误差

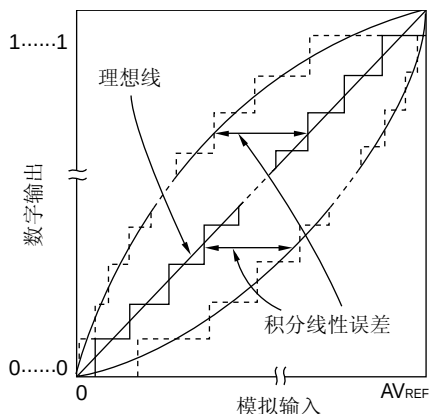
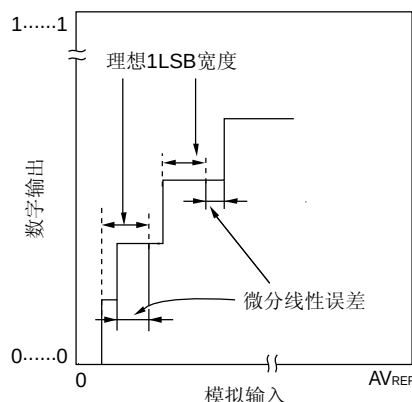


图 10-20. 微分线性误差

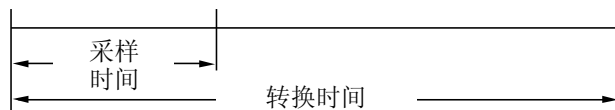
**(8) 转换时间**

转换时间表示从开始采样到获取数字输出所经历的时间。

采样时间包含在特征表中的转换时间中。

(9) 采样时间

采样时间表示模拟电压的模拟开关打开到模拟电压采样&保持电路采样所需的时间。



10.6 A/D 转换器的注意事项

(1) STOP 模式中的操作电流

在 A/D 转换器停止(通过设置 A/D 转换器模式寄存器(ADM)的位 7(ADCS)为 0)后切换至 STOP 模式。同时可通过设置 A/D 转换器模式寄存器(ADM)的位 0(ADCE)为 0 来降低工作电流。

若要从待机状态重新启动, 将中断请求标志寄存器 1L (IF1L)的第 0 位(ADIF)清零(0), 然后开始操作。

(2) ANI0 至 ANI15 的输入范围

观察 ANI0 至 ANI15 输入电压的额定范围。如果输入到模拟输入通道的电压大于等于 AV_{REF} 或者小于等于 AV_{SS} (即使在绝对最大额定范围之内), 则该通道的转换值不确定。此外, 其它通道的转换值也可能会受影响。

(3) 冲突操作

<1> A/D 转换结果寄存器 (ADCR, ADCRH) 的写操作, 与由转换结束后的指令对 ADCR 或 ADCRH 的读操作之间的冲突。

ADCR 或 ADCRH 读操作的优先级高。在执行读操作后, 才将新的转换结果写入 ADCR 或 ADCRH。

<2> 转换结束后, ADCR 或 ADCRH 的写操作与 A/D 转换器模式寄存器(ADM)的写操作、模拟输入通道选择寄存器(ADS)、或 A/D 端口配置寄存器(ADPC)的写操作之间的冲突。

ADM、ADS 或 ADPC 的写操作的优先级高。不执行 ADCR 或 ADCRH 的写操作, 也不会产生转换结束中断信号(INTAD)。

(4) 解决噪音问题的方法

为了保持 10 位分辨率, 必须注意输入到 AV_{REF} 引脚和 ANI0 至 ANI15 引脚的噪声。

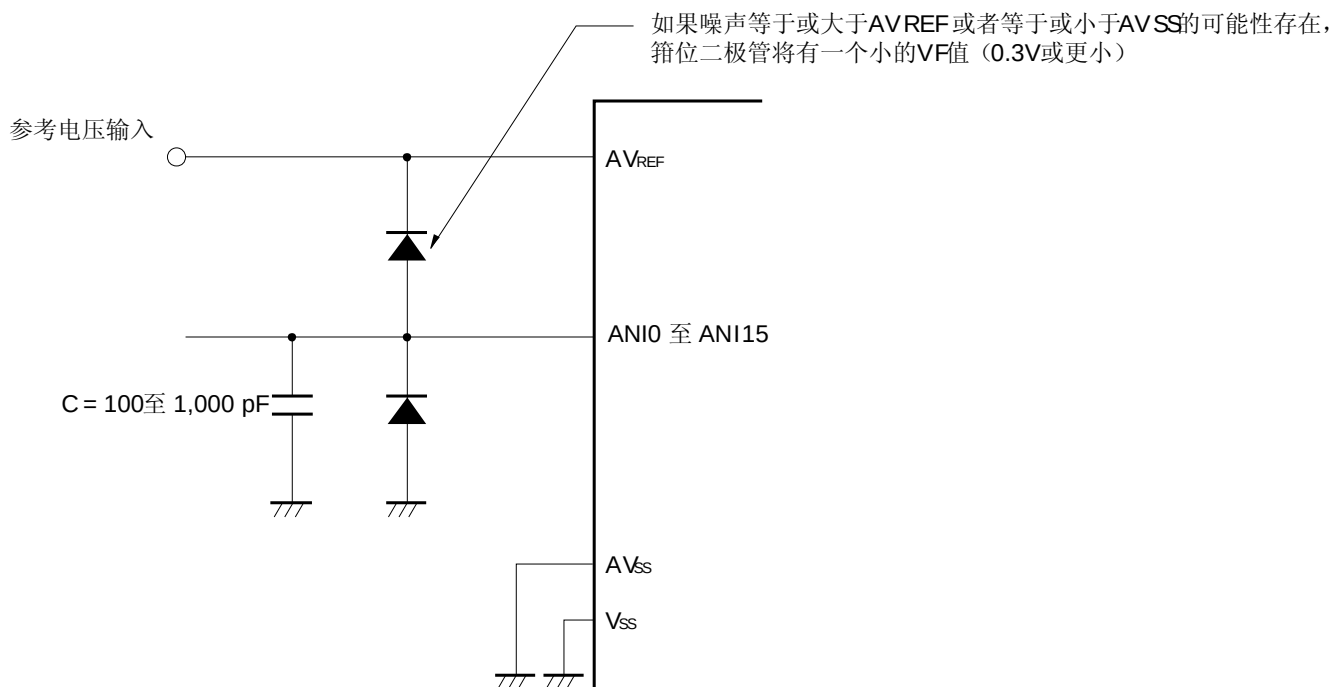
<1> 连接一个低等效电阻和优质频率响应的电容到电源上。

<2> 模拟输入源的输出阻抗越大, 干扰就越大。为了降低噪声, 建议按图 10-21 所示连接外部 C。

<3> 在转换过程中不要切换引脚。

<4> 如果在转换开始后立即设置 HALT 模式, 则可以改善精度。

图 10-21. 模拟输入引脚连接



(5) ANI0 / P20 至 ANI7 / P27 且 ANI8 / P150 至 ANI15 / P157

- <1> 模拟输入引脚 (ANI0 至 ANI7) 也可用作输入端口引脚 (P20 至 P27)。
模拟输入引脚 (ANI8 至 ANI15) 也可用作输入端口引脚 (P150 至 P157)。
当选择 ANI0 至 ANI15 中的任意一个执行 A/D 转换时，转换过程中不要访问 P20 至 P27 和 P150 至 P157；否则转换分辨率可能会降低。选择用作 P20 至 P27 和 P150 至 P157 的引脚时，推荐从离 AV_{REF} 最远的 ANI0/P20 开始。
- <2> 如果正在进行 A/D 转换的引脚的相邻引脚有数字脉冲，则由于噪音耦合，有可能得不到预期的 A/D 转换值。因此在进行 A/D 转换时不要在相邻引脚引用脉冲。

(6) ANI0 至 ANI15 引脚的输入阻抗

采样期间 A/D 转换器对采样电容充电，以便进行采样。
因此当不进行采样时仅有漏电流经过，而在采样期间则有对电容充电的电流。因此根据是正在进行采样还是在其它状态，输入阻抗会发生波动。
但是，为确保采样的有效性，建议保持模拟输入源的输出阻抗在 $1k\Omega$ 以下，并将一个约 $100pF$ 的电容连接到 ANI0 至 ANI15 引脚上(参考 图 10-21)。

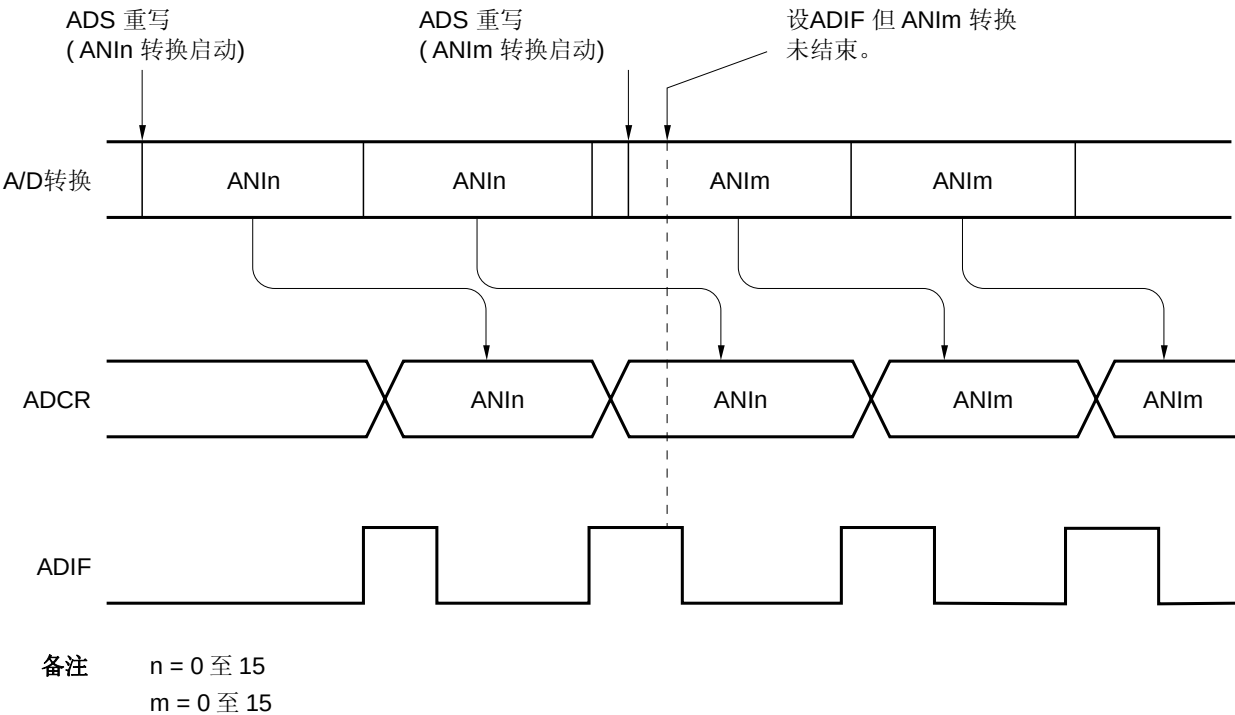
(7) AV_{REF} 引脚输入阻抗

在 AV_{REF} 与 AV_{SS} 引脚之间连接几十千欧的串联电阻串。
因此，如果参考电压源的输出阻抗很高，相当于一系列电阻连接到 AV_{REF} 与 AV_{SS} 引脚之间的串联电阻串，会导致较大的参考电压误差。

(8) 中断请求标志 (ADIF)

即使模拟输入通道指定寄存器(ADS)的值修改，中断请求标志(ADIF)也不会清零。
因此，如果在 A/D 转换期间有一个模拟输入引脚发生变化，则在 ADS 修改之前，模拟输入通道的 A/D 转换结果和改变前的模拟通道的 ADIF 可能设置。此时需要注意的是，当 ADS 修改后立即读取 ADIF 时，即使修改后的模拟输入的 A/D 转换尚未结束，也会设置 ADIF。
当 A/D 转换停止然后又继续时，在 A/D 转换继续操作前清除 ADIF。

图 10-22. A/D 转换结束中断请求产生的时序



(9) A/D 转换刚开始时的转换结果

在 A/D 转换开始后，若在 ADCE 置 1 后的 1 μ s 内对 ADCS 置 1，或者 ADCE=0 时 ADCS 置 1，那么第一次的 A/D 转换值可能不在额定范围内。可采取措施，如挂起 A/D 转换结束中断请求(INTAD)，并删除第 1 次转换结果。

(10) A/D 转换结果寄存器 (ADCR, ADCRH) 读操作

当对 A/D 转换器模式寄存器(ADM)、模拟输入通道选择寄存器(ADS)和 A/D 端口配置寄存器(ADPC)进行写操作时，ADCR 和 ADCRH 的内容可能变得不确定。在对 ADM、ADS 或 ADPC 进行写操作前，且转换结束后读取转换结果。如果采用与上述不同的时序操作可能会读取到不正确的转换结果。

(11) 内部等效电路

模拟输入模块的等效电路如下所示。

图 10-23. ANIn 引脚的内部等效电路

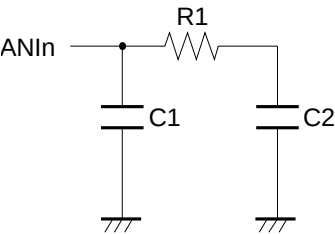


表 10-4. 等效电路的电阻和电容 (参考值)

AVREF	模式	R1	C1	C2
4.0 V ≤ VDD ≤ 5.5 V	标准	5.2 kΩ	8 pF	6.26 pF
	高速1	5.2 kΩ		
	高速2	7.8 kΩ		
2.7 V ≤ VDD < 4.0 V	标准	18.6 kΩ		
	高速2	7.8 kΩ		
1.8 V ≤ VDD < 4.0 V	增压	169.8 kΩ		

- 备注
- 1. 表 10-4 所显示的电阻和电容值不是保证值。
 - 2. n = 0 至 15

第 11 章 串行阵列单元

每个串行阵列单元有四个串行通道，可以将两个或更多不同的串行接口（3 线串行（CSI）、UART 和简易 I²C）组合起来使用。

下列显示了由 78K0R/KG3-L 支持的内部通道的功能分配（单元 1 的通道 2 和 3 专用于 UART3（支持 LIN 总线））。

单位	通道	用作 CSI	用作 UART	用作简易 I ² C
0	0	CSI00	UART0	—
	1	CSI01		—
	2	CSI10	UART1	IIC10
	3	—		—
1	0	CSI20	UART2	IIC20
	1	—		—
	2	—	UART3（支持 LIN 总线）	—
	3	—		—

（整合例子） 当在“UART0”使用单元 0 的通道 0 和 1 时，不能使用 CSI00 和 CSI01，但可使用 CSI10、UART1 或 IIC10。

11.1 串行阵列单元的功能

78K0R/KG3-L 支持的每个串行接口具有下列特性。

11.1.1 3 线串行 I/O (CSI00、CSI01、CSI10 和 CSI20)

这是使用 3 线：串行时钟 (SCK) 和串行数据 (SI 和 SO) 线的一个时钟通信功能。

[数据发送/接收]

- 数据长度 7 或 8 位
- 发送/接收数据的相位控制
- MSB/LSB 可选
- 发送/接收数据的电平设置

[时钟控制]

- 主/从选择
- I/O 时钟的相位控制
- 通过每个通道的预分频器和内部计数器设置传输周期
- 最大传输速率

主通信期间：最大值 $f_{CLK}/4$ ；从通信期间：最大值 $f_{MCK}/6$ ^注

[中断功能]

- 传输结束中断/缓冲空中断

[错误检测标志]

- 溢出错误

注 使用满足 $\overline{SCK}$ 周期时间(t_{KCY})特性范围的时钟（参见第 27 章电气特性（目标值））。

11.1.2 UART (UART0 至 UART3)

这是一个使用 2 线：串行数据发送(TxD) 和串行数据接收(RxD)线的起始停止同步功能。它和通信部分（通过使用内部波特率）同步发送或者接收数据。全双工 UART 通信可以被看作是使用两个通道，一个专门发送（偶通道），一个专门接收(奇通道)。

[数据发送/接收]

- 5、7、或者 8 位的数据长度
- 选择 MSB/LSB
- 发送/接收数据的电平设置和反向选择
- 附加校验位和校验检测功能
- 附加停止位

[中断功能]

- 传输结束中断/缓冲空中断
- 帧错误、校验错误、或者溢出错误时产生错误中断

[错误检测标志]

- 帧错误、校验错误、或者溢出错误

UART3（单元 1 的通道 2 和 3）支持 LIN 总线

[LIN 总线功能]

- 唤醒信号检测
- 同步中断域 (SBF)检测
- 同步域测量、波特率测量

} 使用外部中断(INTP0)或定时器阵列单元(TAU0)

11.1.3 简易 I²C (IIC10, IIC20)

这是一个通过使用 2 线：串行时钟(SCL)和串行数据(SDA)，用于和两个或者更多设备通信的时钟通信功能。这个简易 I²C 专为具有如 EEPROM、Flash 存储器、或者 A/D 转换器设备的单通信设计，因此，它的功能只能作为主，不具有检测等待状态的功能。

确保通过使用软件，和操作控制寄存器，可以读取起始停止条件的 AC 特性。

[数据发送/接收]

- 主发送、主接收（仅具有主功能的单个主设备）
- ACK 输出功能*和 ACK 检测功能
- 8 位数据长度（当传输一个地址时，通过高 7 位指定地址，剩下的 1 位用于 R/W 控制）
- 手动产生启动条件和停止条件

[中断功能]

- 传输结束中断

[错误检测标志]

- 校验错误（ACK 错误）

* [简易 I²C 不支持的功能]

- 从发送、从接收
- 仲裁丢失检测功能
- 等待检测功能

注 当通过写入 0 到 SOEmn（SOEm 寄存器）位接收最后数据时，ACK 不输出并且停止串行通信数据的输出。详情参见 11.7.3 (2) 处理流程。

备注 1. 要使用 I²C 总线的全部功能，参见 第 12 章 串行接口 IICA。
2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2)

11.2 串行阵列单元的配置

串行阵列单元包括下面硬件。

表 11-1. 串行阵列单元的配置

项目	配置
移位寄存器	8 位
缓冲寄存器	串行数据寄存器 mn(SDRmn) [※] 的低 8 位
串行时钟 I/O	SCK00、SCK01、SCK10 和 SCK20 引脚（用于 3 线串行 I/O），SCL10 和 SCL20 引脚（用于简易 I ² C）
串行数据输入	SI00、SI01、SI10 和 SCK20 引脚（用于 3 线串行 I/O），RxD0 至 RxD2 引脚（用于 UART），RxD3 引脚（用于支持 LIN-总线的 UART）
串行数据输出	SO00、SO01、SO10 和 SO20 引脚（用于 3 线串行 I/O），TxD0 至 RxD2 引脚（用于 UART），TxD3 引脚（用于支持 LIN 总线的 UART），输出控制器
串行数据 I/O	SDA10 和 SDA20 引脚（用于简易 I ² C）
控制寄存器	<单元设置块寄存器> • 外围使能寄存器 0 (PER0) • 串行时钟选择寄存器 m (SPSm) • 串行通道使能状态寄存器 m(SEm) • 串行通道启动寄存器 m(SSm) • 串行通道停止寄存器 m(STm) • 串行输出使能寄存器 m(SOEm) • 串行输出寄存器 m(SOM) • 串行输出电平寄存器 m(SOLm) • 输入切换控制寄存器 (ISC) • 噪声滤波器使能寄存器 0 (NFEN0)
	<每个通道的寄存器> - 串行数据寄存器 mn (SDRmn) - 串行模式寄存器 mn (SMRmn) - 串行通信操作设置寄存器 mn (SCRmn) - 串行状态寄存器 mn (SSRmn) - 串行标志清零触发寄存器 mn (SIRmn) - 端口输入模式寄存器 0, 1, 14 (PIM0, PIM1, PIM14) - 端口输出模式寄存器 0, 1, 14 (POM0, POM1, POM14) - 端口模式寄存器 0, 1, 4, 14 (PM0, PM1, PM4, PM14) - 端口寄存器 0, 1, 4, 14 (P0, P1, P4, P14)

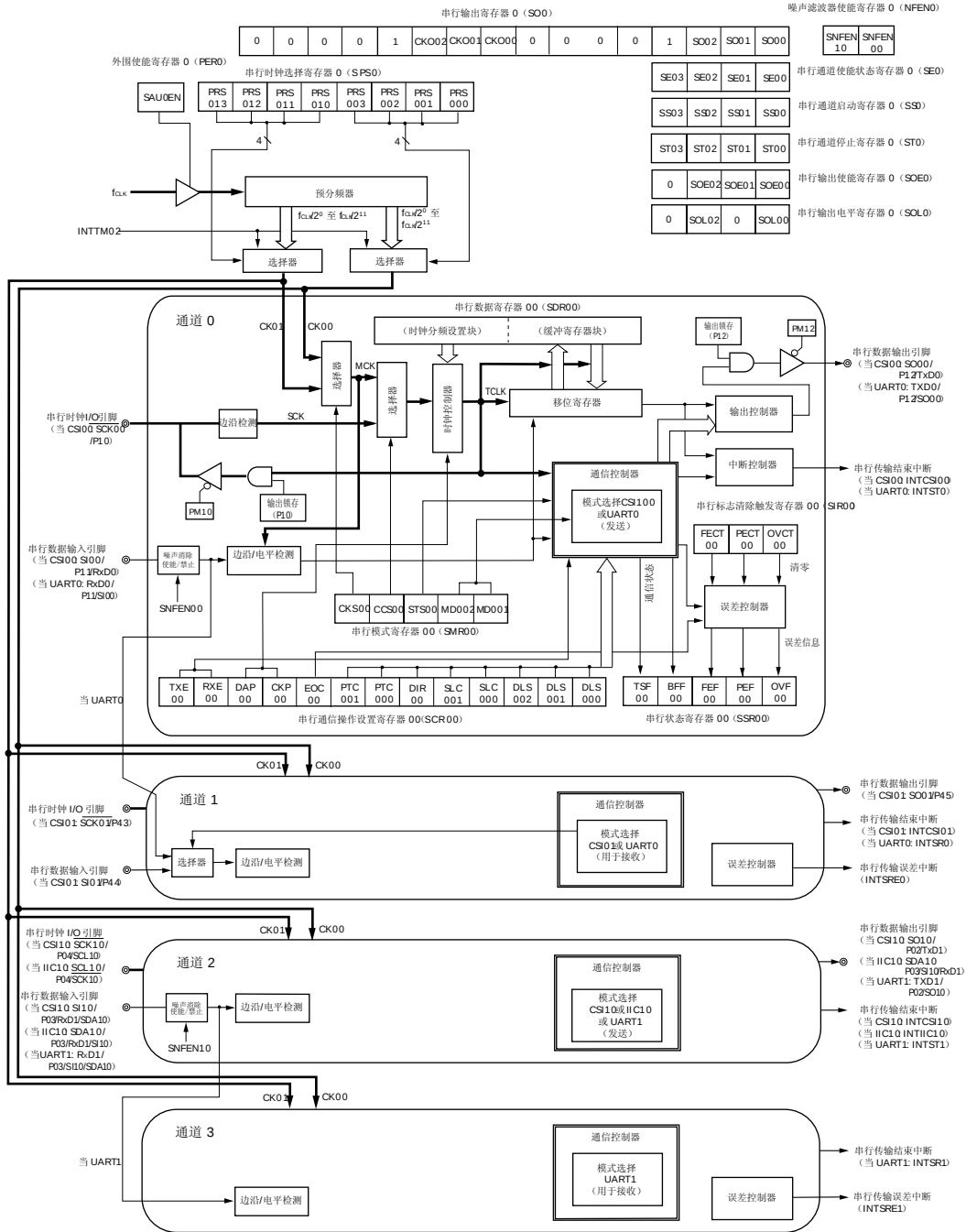
注 根据通信模式的不同,串行数据寄存器 mn (SDRmn) 的低 8 位可以使用以下 SFR 来读取或写入。

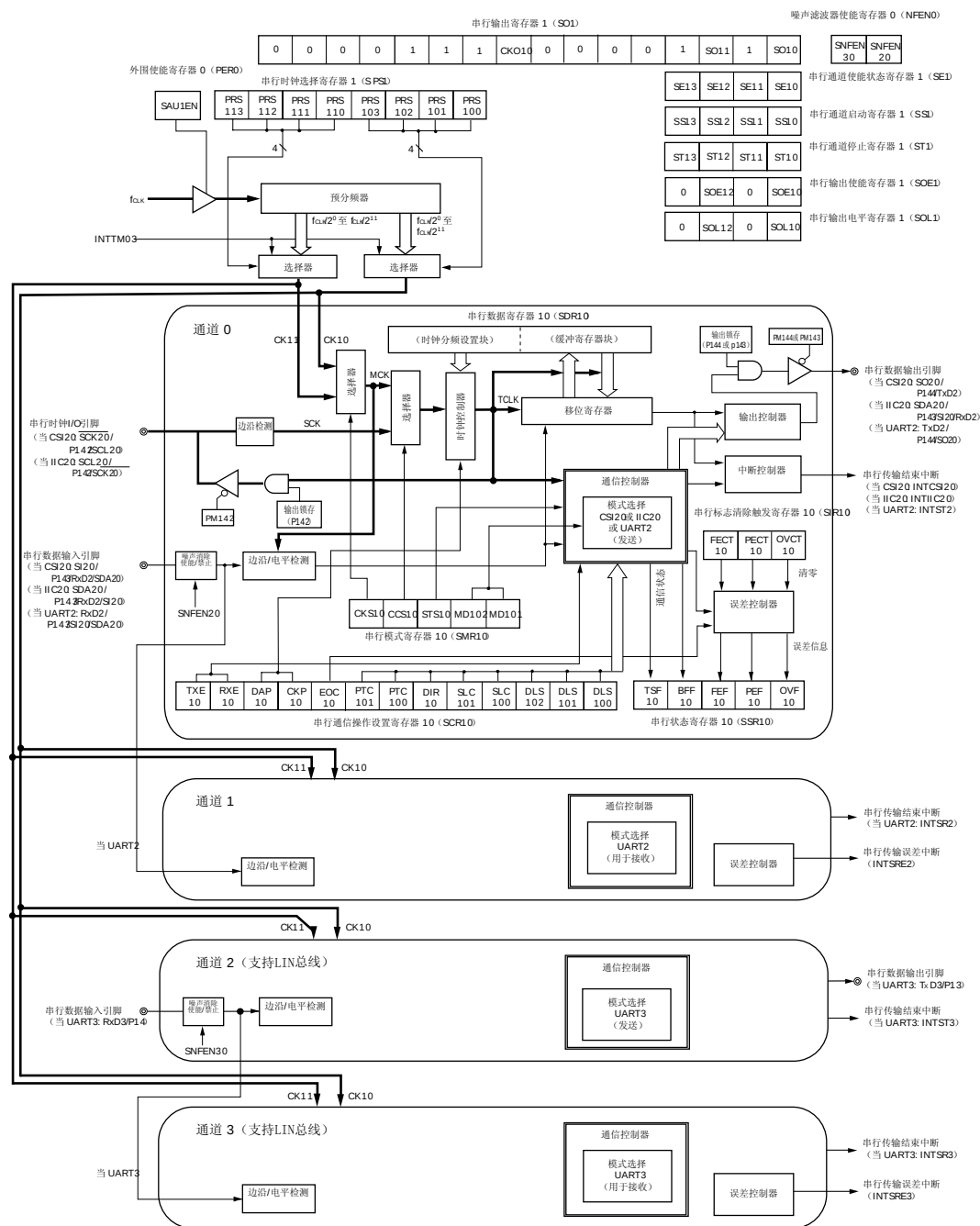
- CSIp 通信...SIOp (CSIp 数据寄存器)
- UARTq 接受...RxDq (UARTq 接收数据寄存器)
- UARTq 发送...TxDq (UARTq 发送数据寄存器)
- IICr 通信 ... SIOr (IICr 数据寄存器)

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13
p: CSI 编号 (p = 00, 01, 10, 20), q: UART 编号 (q = 0 至 3), r: IIC 编号 (r = 10, 20)

图 12-1 所示为串行阵列单元 0 的框图。

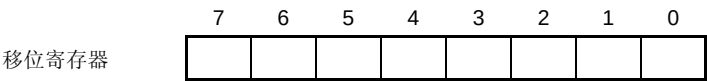
图 11-1. 串行阵列单元 0 的框图





(1) 移位寄存器

这是一个将并行数据转换到串行数据或者相反的 8 位寄存器。
接收期间，它把输入到串行引脚的数据转换为并行数据。
当数据传输时，设置到这个寄存器的值如同从串行输出引脚的串行数据一样输出。
移位寄存器不能直接通过程序操作。
要读取或写入移位寄存器，使用串行数据寄存器 mn(SDRmn)的低 8 位。



(2) 串行数据寄存器 mn(SDRmn)的低 8 位

SDRmn 是通道 n 的发送/接收数据寄存器（16 位）。第 7 位至第 0 位用发送/接收缓冲寄存器，第 15 位至第 9 位用来设置工作时钟 (MCK) 的分频比。
当接受数据时，通过保存在低 8 位的移位寄存器来转换并行数据。当发送数据时，设置要传输的数据到移位寄存器的低 8 位。
根据 SCRmn 寄存器的位 0 至 2 (DLSmn0 至 DLSmn2) 的设置，保存到该寄存器的低 8 位的数据如下所示，与数据的输出顺序无关。

- 5 位的数据长度（保存在 SDRmn 寄存器的位 0 至 4 中）（只有在 UART 模式下可以设置）
- 7 位的数据长度（保存在 SDRmn 寄存器的位 0 至 6 中）
- 8 位的数据长度（保存在 SDRmn 寄存器的位 0 至 7 中）

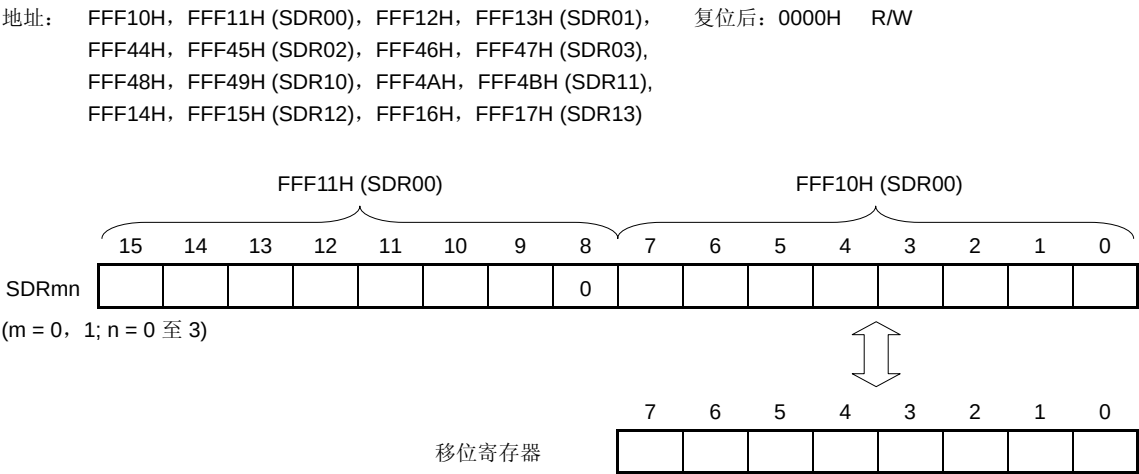
SDRmn 可以以 16 位为单元读取或写入。
根据通信模式的不同，SDRmn 的低八位可作为下列 SFR 进行读或写[※]。

- CSIp 通信...SIOp（CSIp 数据寄存器）
 - UARTq 接受...RxDq（UARTq 接收数据寄存器）
 - UARTq 发送...TxDq（UARTq 发送数据寄存器）
 - IICr 通信 ... SIOr（IICr 数据寄存器）
- 复位后这个寄存器的值为 0000H。

注 当操作停止时(SEmn=0)，禁止写入八位单元。

- 备注
1. 在接收数据后，存放在位 0 至 7 的“0”超出数据长度。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13
p: CSI 编号 (p = 00, 01, 10, 20), q: UART 编号 (q = 0 至 3), r: IIC 编号 (r = 10, 20)

图 11-3. 串行数据寄存器 mn(SDRmn)的格式



11.3 控制串行阵列单元的寄存器

通过下列寄存器控制串行阵列单元

- 外围使能寄存器 0 (PER0)
- 串行时钟选择寄存器 m (SPSm)
- 串行模式寄存器 mn (SMRmn)
- 串行通信操作设置寄存器 mn (SCRmn)
- 串行数据寄存器 mn (SDRmn)
- 串行状态寄存器 mn (SSRmn)
- 串行标志清零触发寄存器 mn (SIRmn)
- 串行通道使能状态寄存器 m (SEm)
- 串行通道启动寄存器 m (SSm)
- 串行通道停止寄存器 m (STm)
- 串行输出使能寄存器 m (SOEm)
- 串行输出电平寄存器 m (SOLm)
- 串行输出寄存器 m (SOM)
- 输入切换控制寄存器 (ISC)
- 噪声过滤使能寄存器 0 (NFEN0)
- 端口输入模式寄存器 0, 1, 14 (PIM0, PIM1, PIM14)
- 端口输出模式寄存器 0, 1, 14 (POM0, POM1, POM14)
- 端口模式寄存器 0, 1, 4, 14 (PM0, PM1, PM4, PM14)
- 端口寄存器 0, 1, 4, 14 (P0, P1, P4, P14)

备注 m: 单元编号 (m = 0, 1)
 n: 通道 (n = 0 至 3)
 mn = 00 至 03, 10 至 13

(1) 外围使能寄存器 0 (PER0)

PER0 用于使能或者禁止使用每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少消耗和噪声。

当使用串行阵列单元 0 时，确保设置该寄存器的位 2(SAU0EN)为 1。

当使用串行阵列单元 1 时，确保设置该寄存器的位 3(SAU1EN)为 1。

可以通过 1 位或者 8 位存储器操作指令来设置 PER0。

复位后这个寄存器的值为 00H。

图 11-4. 外围使能寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

SAUmEN	串行阵列单元m输入时钟的控制
0	停止输入时钟的供给 - 串行阵列单元m使用的SFR不能写入。 - 串行阵列单元m处于复位状态。
1	供给输入时钟 串行阵列单元m使用的SFR可以被读/写。

- 注意事项**
1. 当设置串行阵列单元 m 时，确保先设置 SAUmEN 为 1。如果 SAUmEN=0，向串行阵列单元 m 的控制寄存器写入被忽略，且即使读取寄存器，也只能读到默认值（噪声滤波器使能寄存器 (NFEN0)、端口输入模式寄存器(PIM0, PIM1, PIM14)、端口输出模式寄存器 (POM0, POM1, POM14)、端口模式寄存器 (PM0, PM1, PM4, PM14)和端口寄存器(P0, P1, P4, P14)外）。
 2. 在设置 SAUmEN 为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。
 3. 确保将位 6 清零。

备注 m: 单元编号 (m = 0, 1)

(2) 串行时钟选择寄存器 m(SPSm)

SPSm 是一个 16 位寄存器，用于选择两种类型的工作时钟（CKm0, CKm1），这些时钟通常供给每个通道。

CKm1 由 SPS0 的位 7 至 4 选择，CKm0 通过将位 3 设置为 0 选择。

当寄存器在操作时（当 SEMn = 1 时）禁止重写 SPSm。

SPSm 可由 16 位存储器操作指令设置。

SPSm 的低 8 位可由带 SPSmL 的 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 11-5. 串行时钟选择寄存器 m (SPSm) 的格式

地址: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mp3	PRS mp2	PRS mp1	PRS mp0	工作时钟(CKmp) ^{#1} 的选择				
					f _{CLK} = 2 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz
0	0	0	0	f _{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	f _{CLK} /2	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	f _{CLK} /2 ²	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	f _{CLK} /2 ³	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	f _{CLK} /2 ⁴	125 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	f _{CLK} /2 ⁵	62.5 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	f _{CLK} /2 ⁶	31.3 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	f _{CLK} /2 ⁷	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	f _{CLK} /2 ⁸	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	f _{CLK} /2 ⁹	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	f _{CLK} /2 ¹⁰	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	f _{CLK} /2 ¹¹	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	1	1	INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{#2}				
其它值				设置禁止				

- 注 1. 当改变用于 f_{CLK} 的时钟（通过改变系统时钟控制寄存器 (CKC) 的值）时，在停止串行阵列单元 (SAU) 的操作 (STm = 000FH) 之后，才这样做。当选择 INTTM02 和 INTTM03 为工作时钟时，同样停止定时器阵列单元 (TAU0) (TT0 = 00FFH)。
2. 无论 f_{CLK} 的频率为多少（主系统时钟，副系统时钟），通过操作间隔定时器选择 f_{SUB}/4 作为计数时钟（设定 TIS0 寄存器的 TIS02 (如果 m = 0) 或 TIS03 (如果 m = 1) 为 1），并且通过 SPSm 寄存器中 TAU0 选择 INTTM02 和 INTTM03，从而 SAU 可工作在固定的副系统时钟分频比。但是，当改变 f_{CLK}，必须按照上面注 1 描述的停止 SAU 和 TAU0。

- 注意事项 1. 确保将位 15 至 8 清“0”。
2. 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

- 备注 1. f_{CLK}: CPU/外围硬件时钟频率
f_{SUB}: 副系统时钟频率
2. m: 单元编号 (m = 0, 1), p = 0, 1

(3) 串行模式寄存器 mn (SMRmn)

SMRmn 是设置通道 n 的操作模式的一个寄存器。它也同样用作选择工作时钟 (MCK)、指定串行时钟 (SCK) 是否能够输入、设置启动触发、操作模式 (CSI, UART, 或者 I²C) 和中断源。这个寄存器同样仅在 UART 模式中用于反向接收数据的电平。

当寄存器在操作时 (当 SEMn = 1 时) 禁止重写 SMRmn。但是, MDmn0 位可以在工作期间重写。

SMRmn 可由 16 位存储器操作指令设置。

复位后这个寄存器的值为 0020H。

图 11-6. 串行模式寄存器 mn(SMRmn)(1/2)的格式

地址: F0110H, F0111H (SMR00) 至 F0116H, F0117H (SMR03), 复位后: 0020H R/W
F0150H, F0151H (SMR10) 至 F0156H, F0157H (SMR13),

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

CKS mn	通道n的工作时钟 (MCK) 选择
0	通过PRS寄存器设置的预分频器输出时钟CKm0
1	通过PRS寄存器设置的预分频器输出时钟CKm1
通过沿检测器来使用工作时钟MCK。另外, 根据CCSmn位的设置和SDRmn寄存器的高7位的设置, 产生一个传输时钟(TCLK)。	

CCS mn	通道n的传输时钟 (TCLK) 的选择
0	由CKSmn的位指定的分频工作时钟MCK
1	从SCK引脚输入时钟 (在CSI模式下从传输)
传输时钟TCLK用于切换寄存器、通信控制器、输出控制器、中断控制器和错误控制器。当CCSmn = 0时, MCK的分频比由SDRmn寄存器的高7位来设置。	

STS mn	选择一个启动触发源
0	仅软件触发有效 (为CSI、UART发送、及简易I ² C选择)。
1	RxD引脚有效沿 (为UART接收选择)
在SSm寄存器被设置为 1 后, 当以上条件满足时, 启动传输。	

注意事项 确保将位 13 至 9、7、4 和 3 清“0”。确保设置位 5 为“1”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13

图 11-6. 串行模式寄存器 mn(SMRmn)(2/2)的格式

地址: F0110H, F0111H (SMR00) 至 F0116H, F0117H (SMR03), 复位后: 0020H R/W
F0150H, F0151H (SMR10) 至 F0156H, F0157H (SMR13),

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0	控制在UART模式下，通道n接收数据的电平的反向															
0	下降沿检测作为启动位。 捕捉输入通信数据。															
1	上升沿检测作为启动位。 捕捉反向输入通信数据。															

MD mn2	MD mn1	通道n的操作模式设置													
0	0	CSI 模式													
0	1	UART 模式													
1	0	简易 I ² C 模式													
1	1	设置禁止													

MDmn0	通道n的中断源的选择															
0	传输结束中断															
1	缓冲空中断															
对于逐次发送，当SDRmn数据传输后通过设置 MDmn0为1写入下一个发送数据。																

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13

(4) 串行通信操作设置寄存器 mn (SCRmn)

SCRmn 是通道 n 的通信操作设置寄存器。它被用来设置数据发送/接收模式、数据和时钟的相位、错误信号是否被屏蔽、奇偶位、启动位、停止位和数据长度。

当寄存器在操作时（当 SE_{mn} = 1 时）禁止重写 SCRmn。

SCRmn 可由 16 位存储器操作指令设置。

复位后这个寄存器的值为 0087H。

图 11-7. 串行通信操作设置寄存器 mn (SCRmn) (1/3) 的格式

地址： F0118H, F0119H (SCR00) 至 F011EH, F011FH (SCR03), 复位后: 0087H R/W
F0158H, F0159H (SCR10) 至 F015EH, F015FH (SCR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

TXE mn	RXE mn	通道n的操作模式设置
0	0	不开始通信
0	1	仅接收
1	0	仅发送
1	1	发送/接收

DAP mn	CKP mn	在 CSI 模式下，数据和时钟相位的选择	类型
0	0		1
0	1		2
1	0		3
1	1		4
在 UART 模式和简易 I ² C 模式下，确保设置 DAPmn, CKPmn = 0, 0。			

注意事项 确保将位 3、6 和 11 清“0”。确保设置位 2 为“1”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13,
p: CSI 编号 (p = 00, 01, 10, 20)

图 11-7. 串行通信操作设置寄存器 mn (SCRmn) (2/3) 的格式

地址: F0118H, F0119H (SCR00) 至 F011EH, F011FH (SCR03), 复位后: 0087H R/W
F0158H, F0159H (SCR10) 至 F015EH, F015FH (SCR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

EOC mn	错误中断信号 (INTSREx (x = 0至3)) 屏蔽的选择
0	屏蔽错误中断 INTSREx (INTSRx 没有被屏蔽)。
1	使能产生错误中断INTSREx (如果发生错误则屏蔽INTSRx)。
在 CSI 模式、简易 I ² C 模式和 UART 发送过程中 [※] , 设置 EOCmn = 0。 UART 接收过程中, 设置 EOCmn = 1。	

PTC mn1	PTC mn0	在UART模式中设置校验位	
		发送	接收
0	0	不输出校验位	不接受校验
0	1	输出 0 校验	无奇偶校验判断
1	0	输出偶校验	判断偶校验
1	1	输出奇校验	判断奇校验
在CSI模式和简易I ² C模式下, 确保设置PTCmn1=0、PTCmn0=0。			

DIR mn	在CSI和UART模式中选择数据传输顺序
0	首先输入/输出数据MSB
1	首先输入/输出数据LSB
在简易I ² C模式下确保清除DIRmn=0。	

SLC mn1	SLC mn0	在UART模式中设置停止位
0	0	无停止位
0	1	停止位长度 = 1 位
1	0	停止位长度 = 2 位
1	1	设置禁止
当选择传输结束中断, 在全部停止位完成传输后产生中断。 在UART接收过程中以及简易I ² C模式下, 设置 1 位停止位 (SLCmn1, SLCmn0 = 0, 1)。 在CSI模式中, 设置无停止位 (SLCmn1, SLCmn0 = 0, 0)。		

注 当使用 CSI01 但不使用 EOC01 = 0 时, 可能产生中断错误 INTSRE0。

注意事项 确保将位 3、6 和 11 清“0”。确保设置位 2 为“1”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13

(5) 串行数据寄存器 mn(SDRmn)的高 7 位

SDRmn 是通道 n 的发送/接收数据寄存器（16 位）。第 7 位至第 0 位用作发送/接收缓冲寄存器，第 15 位至第 9 位用来设置工作时钟 (MCK) 的分频比。

如果串行模式寄存器 mn (SMRmn) 的 CCSmn 位被清 0，由 SDRmn 的高 7 位设置的工作时钟通过分频设置为传输时钟。

关于 SDRmn 的低 8 位的功能，参见 11.2 串行阵列单元的配置。

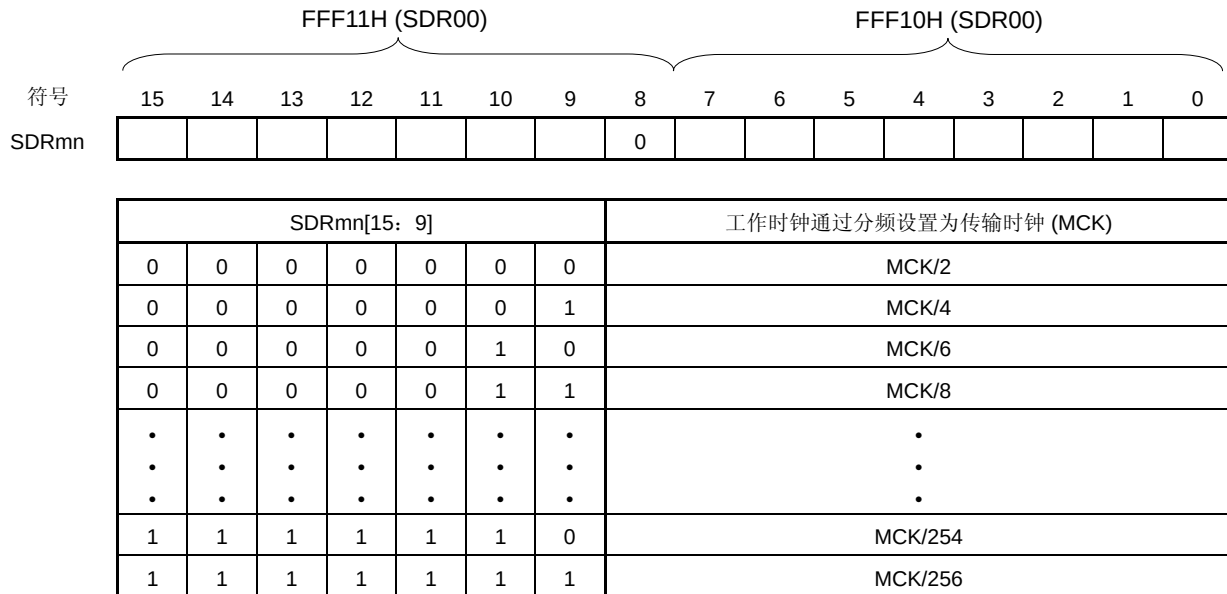
SDRmn 可以以 16 位为单元读取或写入。

然而，仅当操作停止时 (SEmn = 0)，高 7 位才可以被读写。在工作 (SEmn = 1) 期间，仅向 SDRmn 的低 8 位写入数值。工作过程中读取 SDRmn，读取值总是 0。

复位后这个寄存器的值为 0000H。

图 11-8. 串行数据寄存器 mn(SDRmn)的格式

地址: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), 复位后: 0000H R/W
 FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),
 FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11),
 FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)



注意事项 1. 第 8 位必须清“0”。

2. 当使用 UART 时，禁止设置 SDRmn[15: 9] = (0000000B, 0000001B)。

备注 1. 关于 SDRmn 的低 8 位的功能，参见 11.2 串行阵列单元的配置。

2. m: 单元编号 (m = 0, 1)

n: 通道 (n = 0 至 3)

mn = 00 至 03, 10 至 13

(6) 串行状态寄存器 mn (SSRmn)

SSRmn 是指示通道 n 的通信状态和错误发生状态的寄存器。此寄存器可指出帧错误、校验错误和溢出错误。

SSRmn 可以通过一个 16 位存储器操作指令来读取。

SSRmn 的低 8 位可由带 SSRmnL 的 8 位操作指令设置。

复位后这个寄存器的值为 0000H。

图 11-9. 串行状态寄存器 mn(SSRmn) (1/2) 的格式

地址: F0100H, F0101H (SSR00) 至 F0106H, F0107H (SSR03), 复位后: 0000H R
F0140H, F0141H (SSR10) 至 F0146H, F0147H (SSR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn	PEF mn	OVF mn

TSF mn	通道n的通信状态指示标志
0	通信没有执行。
1	通信执行中。
因为这个标志是一个更新标志，当通信操作完成后它将被自动清除。当STmn/SSmn位设为1时，这个标志同样被清除。	

BFFmn	通道n的缓冲寄存器状态指示标志
0	有效数据未被保存在SDRmn寄存器中。
1	有效数据被保存在SDRmn寄存器中。
这是一个更新标志。当完成从SDRmn寄存器到切换寄存器的传输时，它将自动清除。接收过程中，当数据从SDRmn寄存器中读出时，它将自动清除。当STmn/SSmn位设为1时，这个标志同样自动清除。 当SCRmn寄存器的TXEmn位设为1（每个通信模式下的发送或发送/接收模式），如果传输数据写入SDRmn寄存器，这个标志自动清除。当SCRmn寄存器的RXEmn位 = 1（每个通信模式下的接收或者发送/接收模式）时，如果接收数据保存在SDRmn寄存器，它就被自动设置。发生接收错误时它也被设置。 如果BFFmn=1时，数据写入SDRmn寄存器，保存在寄存器中的发送/接收数据将被丢弃，并且会检测到溢出错误(OVFmn=1)。	

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13

图 11-9. 串行状态寄存器 mn(SSRmn) (2/2) 的格式

地址: F0100H, F0101H (SSR00) 至 F0106H, F0107H (SSR03), 复位后: 0000H R
F0140H, F0141H (SSR10) 至 F0146H, F0147H (SSR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn	PEF mn	OVF mn

FEF mn	通道n的帧错误检测标志
0	没有错误产生。
1	在UART接收时产生一个帧错误。 <帧错误原因> 如果UART接收完成后, 没有检测到停止位, 则产生一个帧错误。
这是一个累积标志, 直到SIRmn寄存器的FECTmn位写入1, 它才被清零。	

PEF mn	通道n的校验错误检测标志
0	没有错误产生。
1	在UART接收时产生一个校验错误或者在I ² C发送中没有检测到ACK。 <校验错误原因> - 在UART接收中, 如果发送数据的校验和校验位不匹配, 则产生一个校验错误。 - 在I²C发送中, 如果ACK信号没有在ACK接收时序中从从设备返回, 则没有检测到ACK。
这是一个累积标志, 直到SIRmn寄存器的PECTmn位写入1, 它才被清零。	

OVF mn	通道n的溢出错误检测标志
0	没有错误产生。
1	产生一个溢出错误。 <溢出错误的原因> - 保存在SDRmn寄存器中的接收数据没有读取, 并且发送数据或者下一个接收数据被写入。 - 在CSI模式中, 发送数据不准备为从设备的发送或者发送/接收。
这是一个累积标志, 直到SIRmn寄存器的OVCTmn位写入1, 它才被清零。	

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13

(7) 串行标志清零触发寄存器 mn (SIRmn)

SIRmn 是个触发寄存器，它用于清除通道 n 的每个错误标志。

当该寄存器的每个位(FECTmn, PECTmn, OVCTmn)均设为 1 时，串行状态寄存器 mn 中的对应位(FEFmn, PEFmn, OVFmn) 将被清零。因为 SIRmn 是触发寄存器，在 SSRmn 的对应位清零后将立即清零。

SIRmn 可由 16 位存储器操作指令设置。

SIRmn 的低 8 位可用带 SIRmnL 的 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 11-10. 串行标志清除触发寄存器 mn (SIRmn)的格式

地址: F0108H, F0109H (SIR00) 至 F010EH, F010FH (SIR03), 复位后: 0000H R/W
F0148H, F0149H (SIR10) 至 F014EH, F014FH (SIR13)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FEC Tmn	PEC Tmn	OVC Tmn

FEC Tmn	清除通道n的帧错误的触发
0	没有触发操作
1	将SSRmn寄存器的FEFmn位清为0。

PEC Tmn	清除通道n的校验错误的触发
0	没有触发操作
1	将SSRmn寄存器的PEFmn位清为0。

OVC Tmn	清除通道n的溢出错误的触发
0	没有触发操作
1	将SSRmn寄存器的OVFmn位清为0。

注意事项 确保将位 15 至 3 清“0”。

备注 1.m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13
 2.读 SIRmn 寄存器时，读取的值总是为 0000H。

(8) 串行通道使能状态寄存器 m (SEm)

SEm 表示每个通道的数据发送/接收操作是使能还是禁止。

当串行通道启动寄存器 m (SSm)位写入 1 时，设置该寄存器的相应位为 1。当串行通道停止寄存器 m (STm)的位写入 1 时，清除该寄存器的相应位为 0。

使能操作的通道 n 不能由软件重写串行输出寄存器 m (SOM)的 CKOmn 的值（下面介绍），但是由通信操作反映的数值会从串行时钟引脚输出。

停止操作的通道 n 可以通过软件设置 SOM 寄存器的 CKOmn 的值，并且从串行时钟引脚输出它的值。这样，任何波形，例如启动条件/停止条件，都可以通过软件产生。

SEm 可以通过一个 16 位存储器操作指令来读取。

SEm 中低 8 位可以由 1 位或 8 位 SEmL 存储器操作指令来进行设置。

复位后这个寄存器的值为 0000H。

图 11-11. 串行通道使能状态寄存器 m (SEm)的格式

地址: F0120H, F0121H(SE0), F0160H, F0161H (SE1) 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SEm 3	SEm 2	SEm 1	SEm 0

SEm n	指示通道n的操作使能/停止状态
0	操作停止（此时控制寄存器和切换寄存器的值、串行时钟I/O 引脚状态、串行数据输出引脚，且FEF、PEF、和OVF错误标志会保留 [※] ）。
1	操作使能

注 SSRmn 寄存器的位 6 和位 5 (TSFmn, BFFmn)被清除。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13

(9) 串行通道启动寄存器 m (SSm)

SSm 是用来使能启动每个通道的通信/计数的触发寄存器。

当 1 写入到此寄存器 (SSmn) 的一个位，串行通道使能状态寄存器 m (SEm)的相应位(SEmn)设为 1。因为 SSmn 是一个触发位，当 SEmn= 1 时将立即清零。

SSm 可由 16 位存储器操作指令设置。

SSm 中低 8 位可以由 1 位或 8 位 SSmL 存储器操作指令来进行设置。

复位后这个寄存器的值为 0000H。

图 11-12. 串行通道启动寄存器 m (SSm)的格式

地址: F0122H, F0123H(SS0), F0162H, F0163H (SS1) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm 3	SSm 2	SSm 1	SSm 0

SSmn	通道n的操作起始触发
0	没有触发操作
1	设置SEmn为1，并且进入通信等待状态（如果信操作已经执行，则操作停止、等待启动条件）。

注意事项 确保将位 15 至 4 清“0”。

备注 1.m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13

 2.读 SSm 寄存器时，读取的值总是为 0000H。

(10) 串行通道停止寄存器 m (STm)

STm 是用来使能停止通道的通信/计数触发寄存器。

当 1 写入到此寄存器 (STmn)的一个位时，串行通道使能状态寄存器 m (SEm)的相应位(SEmn)清除为 0。因为 STmn 是一个触发位，当 SEmn= 0 时将立即清零。

STm 可以通过 16 位存储器操作指令来设置。

STm 中低 8 位可以由 1 位或 8 位 STmL 存储器操作指令来进行设置。

复位后这个寄存器的值为 0000H。

图 11-13. 串行通道停止寄存器 m (STm)的格式

地址: F0124H, F0125H (ST0), F0164H, F0165H (ST1) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	STm 3	STm 2	STm 1	STm 0

STm n	通道n的操作停止触发
0	没有触发操作
1	清除SEmn为0，并停止通信操作。 (此时控制寄存器和切换寄存器的值、串行时钟I/O引脚状态的值、串行数据输出引脚的值，且FEF、PEF和OVF错误标志值会保留 ^注)。

注 SSRmn 寄存器的位 6 和位 5 (TSFmn, BFFmn)被清除。

注意事项 确保将位 15 至 4 清“0”。

备注 1. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13
 2. 读 STm 寄存器时，读取的值总是为 0000H。

(11) 串行输出使能寄存器 m (SOEm)

SOEm 是用来使能或停止每个通道的串行通信输出的寄存器。

使能串行输出的通道 n 不能通过软件重写串行输出寄存器 m (SOM)的 SOMn 的值（后面将介绍），但由通信操作反映的数值可从串行数据输出引脚输出。

对于串行输出停止的通道 n，SOM 寄存器的 SOMn 值可以通过软件设置，并且这个值可以从串行数据输出引脚输出。这样，启动条件和停止条件的任何波形可以通过软件产生。

SOEm 可由 16 位存储器操作指令设置。

SOEm 中低 8 位可以由 1 位或 8 位 SOEmL 存储器操作指令来进行设置。

复位后这个寄存器的值为 0000H。

图 11-14. 串行输出使能寄存器 m (SOEm)的格式

地址: F012AH, F012BH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02	SOE01	SOE00

地址: F016AH, F016BH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12	0	SOE10

SOE mn	通道n的串行输出使能/禁止
0	通过串行通信操作停止输出
1	通过串行通信操作使能输出

注意事项 确认清除 SOE0 的 15 至 3 位，并将 SOE1 中的位 1 和位 15 设为“0”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2),
 mn = 00 至 02, 10, 12

(12) 串行输出寄存器 m (SOM)

SOM 是每个通道的串行输出缓冲寄存器。

这个寄存器每位的值从通道 n 的串行数据输出引脚来输出。

这个寄存器每位 (n + 8) 的值从通道 n 的串行时钟输出引脚来输出。

只有在串行输出为禁用(SOEmn = 0) 状态下时该寄存器的 SOMn 才可以通过软件重写。当串行输出使能时(SOEmn = 1)，忽略通过软件重写，并且该寄存器的值仅可通过一个串行通信操作来改变。

只有在通道操作停止(SEmn = 0) 状态下时，该寄存器的 CKOmn 才可以通过软件重写。当通道操作使能时(SEmn = 1)，忽略通过软件重写，并且 CKOmn 的值仅可通过一个串行通信操作来改变。

要将 P02/SO10/TxD1，P03/SI10/SDA10/RxD1，P04/SCK10/SCL10，P10/SCK00，P12/SO00/TxD0，P13/TxD3，P43/SCK01，P45/SO01，P142/SCK20/SCL20，P143/SI20/SDA20/RxD2 或者 P144/SO20/TxD2 引脚作为端口功能引脚时，需要将相应 CKOmn 和 SOMn 位设置为“1”。

SOM 可由 16 位存储器操作指令设置。

复位后这个寄存器的值为 0F0FH。

图 11-15. 串行输出寄存器 m (SOM)格式

地址: F0128H, F0129H 复位后: 0F0FH R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO 02	CKO 01	CKO 00	0	0	0	0	1	SO 02	SO 01	SO 00

地址: F0168H, F0169H 复位后: 0F0FH R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO 10	0	0	0	0	1	SO 12	1	SO 10

CKO mn	通道n的串行时钟输出
0	串行时钟输出值为“0”。
1	串行时钟输出值为“1”。

SO mn	通道n的串行数据输出
0	串行数据输出值为“0”。
1	串行数据输出值为“1”。

注意事项 确认设置 SO0 的 11 至 3 位设为 1，并将 SO1 中的位 11 至 9、位 3 及位 1 设为“1”。并且确保清除 SOM 的位 15 至 12 和 7 至 4 为“0”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2),
mn = 00 至 02, 10, 12

(13) 串行输出电平寄存器 m (SOLm)

SOLm 是用来设置每个通道数据输出电平反转的寄存器。

仅在 UART 模式下可以设置这个寄存器。确保在 CSI 模式和简易 I²C 模式中设置为 0000H。

仅当串行输出使能 (SOEmn = 1) 时，通过该寄存器将通道 n 反向，由引脚输出反映。当串行输出禁止 (SOEmn = 0) 时，SOMn 位的值将输出。

当寄存器在操作时（当 SEMn = 1 时）禁止重写 SOLm。

SOLm 可由 16 位存储器操作指令设置。

SOLm 中低 8 位可以由 8 位 SOLmL 存储器操作指令来进行设置。

复位后这个寄存器的值为 0000H。

图 11-16. 串行输出电平寄存器 m (SOLm)的格式

地址: F0134H, F0135H (SOL0), F0174H, F0175H (SOL1) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL m2	0	SOL m0

SOL mn	在UART模式中选择通道n的发送数据反向电平														
0	通信数据原样输出														
1	通信数据反向输出														

注意事项 确保将位 15 至 3 和 1 清“0”。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 00, 02, 10, 12

(14) 输入切换控制寄存器 (ISC)

通过 UART3、外部中断和定时器阵列单元 TAU0 的联合操作，ISC 用于实现 LIN 总线通信操作。

当 0 位设置为 1 时，串行数据输入 (RxD3) 引脚的输入信号可选择为外部中断 (INTP0)，该中断可以用来检测唤醒信号。

当第 1 位设置为 1 时，串行数据输入 (RxD3) 引脚的输入信号选择为定时器输入，所以，同步中断域和同步域的脉冲宽度可以由定时器测量。

ISC 可以通过一个 1 位或者 8 位存储器操作指令来设置。

复位后这个寄存器的值为 00H。

图 11-17. 输入切换控制寄存器(ISC)的格式

地址: FFF3CH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	切换定时器阵列单元(TAU0)的通道7输入
0	使用TI07引脚的输入信号作为定时器输入（正常模式）。
1	RxD3引脚的输入信号被用作定时器输入（唤醒信号检测）。

ISC0	切换外部中断 (INTP0)输入
0	使用INTP0引脚的输入信号作为外部中断（正常模式）。
1	使用RxD3引脚输入信号作为外部中断 (用来测量同步中断域和同步域的脉冲宽度)。

注意事项 确保将位 7 至 2 清“0”。

(15) 噪声过滤使能寄存器 0 (NFEN0)

NFEN0 用于设置是否为每个通道从串行数据输入引脚的输入信号使用噪声过滤。

通过清除该寄存器的对应位为 0，禁止用于 CSI 或简易 I²C 通信的引脚的噪声滤波器。

通过设置这个寄存器的相应位为 1，使能用于 UART 通信引脚的噪声过滤。

当使能噪声过滤，CPU/外围硬件工作时钟 (f_{CLK}) 与 2 时钟同步匹配检测的。

NFEN0 可以通过一个 1 位或者 8 位存储器操作指令来设置。

复位后这个寄存器的值为 00H。

图 11-18. 噪声过滤使能寄存器 0 (NFEN0)

地址: F0060H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
NFEN0	0	SNFEN30	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN30	RxD3/P14引脚噪声滤波器的使用
0	噪声过滤关闭
1	噪声过滤开启
设置SNFEN30为1来使用RxD3引脚。 将SNFEN30清0来使用P14引脚。	

SNFEN20	RxD2/SDA20/SI20/P143引脚噪声滤波器的使用
0	噪声过滤关闭
1	噪声过滤开启
设置SNFEN20为1来使用RxD2引脚。 清除SNFEN20为0，来使用SDA20，SI20，和P143引脚。	

SNFEN10	RxD1/SDA10/SI10/P03引脚噪声滤波器的使用
0	噪声过滤关闭
1	噪声过滤开启
设置SNFEN10为1来使用RxD1引脚。 清除SNFEN10为0，来使用SDA10，SI10，和P03引脚。	

SNFEN00	RxD0/SI00/P11引脚噪声滤波器的使用
0	噪声过滤关闭
1	噪声过滤开启
设置SNFEN00为1来使用RxD0引脚。 清除SNFEN00为0，来使用SI00和P11引脚。	

注意事项 确保将位 7，5，3 和 1 清为“0”。

(16) 端口输入模式寄存器 0, 1, 14 (PIM0, PIM1, PIM14)

这些寄存器以 1 位单元设置端口 0、1 和 14 的输入缓冲器。

PIM0、PIM1 和 PIM14 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生可将这些寄存器清零(00H)。

图 11-19. 端口输入模式寄存器 0, 1 和 14(PIM0, PIM1, PIM14)的格式

地址: F0040H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PIM0	0	0	0	PIM04	PIM03	0	0	0

地址: F0041H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PIM1	0	0	0	0	0	0	PIM11	PIM10

地址: F004EH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PIM14	0	0	0	0	PIM143	PIM142	0	0

PIMmn	Pmn 引脚输入缓冲选择(m = 0、1、14; n = 0 至 4)							
0	普通输入缓冲器							
1	TTL输入缓冲器							

(17) 端口输出模式寄存器 0, 1, 14 (POM0, POM1, POM14)

这些寄存器以 1 位为单元设置端口 0、1 和 14 的输出模式。

POM0, POM1 和 POM14 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生可将这些寄存器清零(00H)。

图 11-20. 端口输出模式寄存器 0, 1 和 14(POM0, POM1, POM14)的格式

地址: F0050H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
POM0	0	0	0	POM04	POM03	POM02	0	0

地址: F0051H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
POM1	0	0	0	0	0	POM12	0	POM10

地址: F005EH	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
POM14	0	0	0	POM144	POM143	POM142	0	0

POMmn	Pmn引脚输出缓冲器选择 (m = 0、1和14; n = 0、2至4)							
0	正常输出模式							
1	N-ch漏极开路 (V _{DD} 耐压) 模式							

(18) 端口模式寄存器 0, 1, 4, 14 (PM0, PM1, PM4, PM14)

这些寄存器以 1 位为单元设定端口 0、1、4 和 14 的输入/输出。

当使用 P02/SO10/TxD1, P03/SI10/SDA10/RxD1, P04/ $\overline{\text{SCK10}}$ /SCL10, P10/ $\overline{\text{SCK00}}$ /, P12/SO00/TxD0/, P13/TxD3/, P43/ $\overline{\text{SCK01}}$, P45/SO01, P142/ $\overline{\text{SCK20}}$ /SCL20, P143/SI20/SDA20/RxD2 和 P144/SO20/TxD2 引脚作为串行数据输出或串行时钟输出时, 将 PM02, PM03, PM04, PM10, PM12, PM13, PM43, PM45, PM142, PM143 和 PM144 位清零并设置 P02, P03, P04, P10, P12, P13, P43, P45, P142, P143 和 P144 的输出锁存为 1。

当使用 P03/SI10/RxD1/SDA10, P04/ $\overline{\text{SCK10}}$ /SCL10, P10/ $\overline{\text{SCK00}}$, P11/SI00/RxD0, P14/RxD3, P43/ $\overline{\text{SCK01}}$, P44/SI01, P142/ $\overline{\text{SCK20}}$ /SCL20 和 P143/SI20/RxD2/SDA20 引脚作为串行数据输入或串行时钟输入时, 设置 PM03, PM04, PM10, PM11, PM14, PM43, PM44, PM142, and PM143 位为 1。此时, P03, P04, P10, P11, P14, P43, P44, P142 和 P143 的输出锁存可能为 0 或 1。

PM0, PM1, PM4 和 PM14 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生将这些寄存器设置为 FFH。

图 11-21. 端口模式寄存器 0、1、4 和 14 (PM0, PM1, PM4, PM14) 的格式

地址: FFF20H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01	PM00

地址: FFF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

地址: FFF24H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

地址: FFF2EH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140

PMmn	Pmn引脚I/O模式选择 (m = 0, 1, 4, 14; n = 0 至 7)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

11.4 操作停止模式

串行阵列单元的每个串行接口都有操作停止模式。

在这种模式下，串行通信不能执行，因此可以降低功耗。

此外，在此模式下 P02/SO10/TxD1, P03/SI10/SDA10/RxD1, P04/ $\overline{\text{SCK10}}$ /SCL10, P10/ $\overline{\text{SCK00}}$, P11/SI00/RxD0, P12/SO00/TxD0, P13/TxD3, P14/RxD3, P43/ $\overline{\text{SCK01}}$, P44/SI01, P45/SO01, P142/ $\overline{\text{SCK20}}$ /SCL20, P143/SI20/SDA20/RxD2 或 P144/SO20/TxD2 引脚也可用作普通端口。

11.4.1 由部件停止操作

由部件停止操作可以通过外围使能寄存器 0(PER0)来设置。

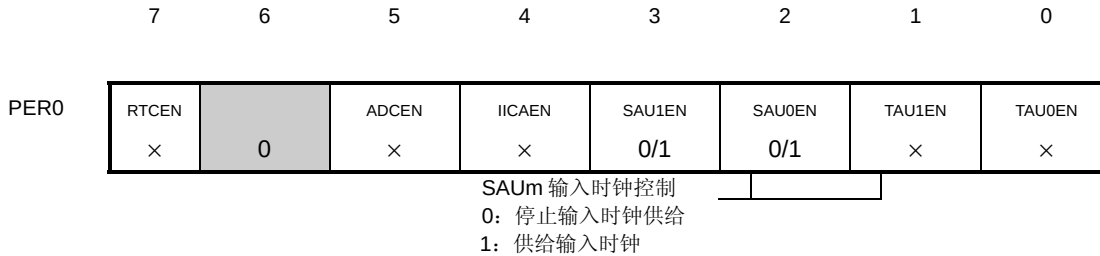
PER0 用于使能或者禁止使用每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少消耗和噪声。

要停止串行阵列单元 0 的操作，设置位 2(SAU0EN)为 0。

要停止串行阵列单元 1 的操作，设置位 3(SAU1EN)为 0。

图 11-22. 由部件停止操作时，外围使能寄存器 0 (PER0) 的设置

(a) 外设使能寄存器 0 (PER0) ... 仅设置将要停止的 SAUm 位为 0。



注意事项 1. 如果 SAUmEN=0，向串行阵列单元的控制寄存器写入被忽略，且即使读取寄存器，也只能读到默认值（除噪声滤波器使能寄存器 (NFEN0)、端口输入模式寄存器(PIM0, PIM1, PIM14)、端口输出模式寄存器 (POM0, POM1, POM14)、端口模式寄存器 (PM0, PM1, PM4, PM14)和端口寄存器 (P0, P1, P4, P14)外）。

2. 确保清除位 6 为 0。

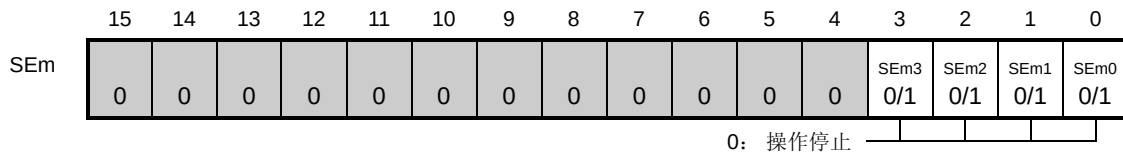
备注 m: 单元编号 (m = 0, 1), : 禁止设置(由硬件给出固定值)
 ×: 位不与串行阵列单元共同使用(根据其他外设功能的设定)
 0/1: 根据用户的使用方法设为 0 或 1

11.4.2 由通道停止操作

由通道停止操作使用以下每一个寄存器设置。

图 11-23. 由通道停止操作时，每个寄存器的设置 (1/2)

(a) 串行通道使能状态寄存器 **m(SEm)** ... 该寄存器表示每个通道的数据发送/接收操作被使能还是被禁止。



* SEm 寄存器是一个只读寄存器，它可以使用 STm 寄存器来停止操作。

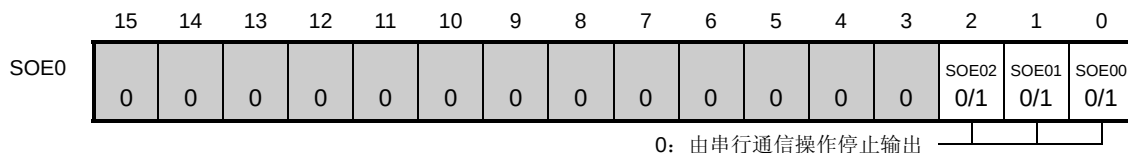
当通道停止操作时，SOm 寄存器的 CKOmn 的值可以通过软件设置。

(b) 串行通道停止寄存器 **m(STm)** ... 该寄存器为触发寄存器，用来使能每个通道停止通信/计数。

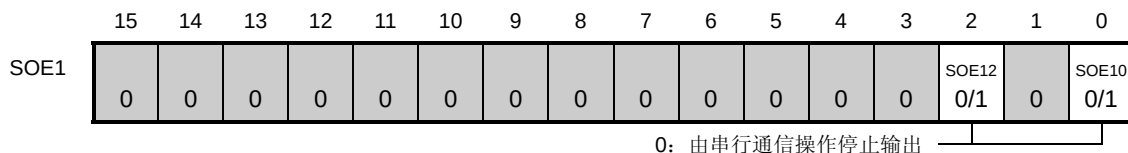


*因为 STmn 是触发位，当 SEmn = 0 时，它立即被清除。

(c) 串行输出使能寄存器 **m(SOEm)**... 该寄存器是用来使能或停止每个通道的串行通信输出的寄存器。



*对于串行输出停止的通道 n，SO0 寄存器的 SO0n 值可以通过软件设置。



*对于串行输出停止的通道 n，SO1 寄存器的 SO10 和 SO12 值可以通过软件设置。

备注 m: 单元编号(m = 0, 1), n: 通道编号 (n = 0 to 3)

■: 禁止设置(由硬件设置固定值), 0/1: 根据用户的使用方法设为 0 或 1

图 11-23. 由通道停止操作时，每个寄存器的设置 (2/2)

(d) 串行输出寄存器 m(SOm) ... 该寄存器为缓冲寄存器，用于每个通道的串行输出

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0						CKO02	CKO01	CKO00						SO02	SO01	SO00
	0	0	0	0	1	0/1	0/1	0/1	0	0	0	0	1	0/1	0/1	0/1
	1: 串行时钟输出值是“1”					1: 串行数据输出值是“1”										

* 当使用对应于每个通道的引脚作为端口功能引脚时，设置对应的 CKO0n 和 SO0n 位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1								CKO10						SO12		SO10
	0	0	0	0	1	1	1	0/1	0	0	0	0	1	0/1	1	0/1
	1: 串行时钟输出值是“1”					1: 串行数据输出值是“1”										

* 当使用对应于每个通道的引脚作为端口功能引脚时，设置对应的 CKO0n,SO10 位和 SO10 位为 1。

备注 m: 单元编号(m = 0, 1), n: 通道标号 (n = 0 至 3)

■ : 禁止设置 (由硬件设置固定值), 0/1: 根据用户的使用方法设为 0 或 1

11.5 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 通信的操作

这是使用 3 线：串行时钟 (SCK) 和串行数据 (SI 和 SO) 线的一个时钟通信功能。

[数据发送/接收]

- 数据长度 7 或 8 位
- 发送/接收数据的相位控制
- MSB/LSB 可选
- 发送/接收数据的电平设置

[时钟控制]

- 主/从选择
 - I/O 时钟的相位控制
 - 通过每个通道的预分频器和内部计数器设置传输周期
 - 最大传输速率
- 主通信期间：最大值 $f_{CLK}/4$ ；从通信期间：最大值 $f_{MCK}/6$ ^註

[中断功能]

- 传输结束中断/缓冲空中断

[错误检测标志]

- 溢出错误

注 使用满足 $\overline{SCK}$ 周期时间(t_{KCY})特性范围的时钟（参见第 27 章电气特性（目标值））。

支持 3 线串行 I/O(CSI00、CSI01、CSI10 和 CSI20)的通道是 SAU0 的通道 0 至 2 和 SAU1 的通道 0。

单位	通道	用作CSI	用作UART	用作简易I ² C
0	0	CSI00	UART0	—
	1	CSI01		—
	2	CSI10	UART1	IIC10
	3	—		—
1	0	CSI20	UART2	IIC20
	1	—	UART3（支持LIN总线）	—
	2	—		—
	3	—		—

3 线串行 I/O (CSI00, CSI01, CIS10, CSI20) 执行下列通信操作的 6 种类型。

- 主发送 (参见 11.5.1.)
- 主接收 (参见 11.5.2.)
- 主发送/接收 (参见 11.5.3.)
- 从发送 (参见 11.5.4.)
- 从接收 (参见 11.5.5.)
- 从发送/接收 (参见 11.5.6.)

11.5.1 主发送

主发送是 78K0R/KG3-L 输出一个传输时钟和发送数据到另外一个设备。

3线串行I/O	CSI00	CSI01	CSI10	CSI20
目标通道	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用引脚	$\overline{\text{SCK00}}$, SO00	$\overline{\text{SCK01}}$, SO01	$\overline{\text{SCK10}}$, SO10	$\overline{\text{SCK20}}$, SO20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	可以选择传输结束中断（在单发送模式下）或者缓冲空中断（在连续发送模式下）			
错误检测标志	无			
发送数据长度	7 或 8 位			
传输速率	最大值 $f_{\text{CLK}}/4$ [MHz], 最小值 $f_{\text{CLK}}/(2 \times 2^{11} \times 128)$ [MHz] ^注 f_{CLK} : 系统时钟频率			
数据相位	通过DAPmn位可选择 • DAPmn = 0: 从串行时钟开始操作时, 数据输出开始。 • DAPmn = 1: 在串行时钟操作开始之前半个时钟, 数据输出开始。			
时钟相位	通过 CKPmn位选择 • CKPmn = 0: 正向 • CKPmn = 1: 反向			
数据方向	首先是MSB 或者LSB			

注 请在满足上面条件和电气特性中的 AC 特性的范围内, 使用这个操作（参见 第 27 章 电气特性（目标值））。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10

(1) 寄存器设置

图 11-24. 用于 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 的主发送的寄存器内容的示例

(a) 串行输出寄存器 m (SOM)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	1	CKOm2 0/1	CKOm1 0/1	CKOm0 0/1	0	0	0	0	1	SOM2 0/1	SOM1 0/1	SOM0 0/1

如果数据相位是正向的(CKPmn = 0)，则当这些位为 1 时，通信开始。如果相位是反向的 (CKPmn = 1)，则当这些位为 0 时，通信开始。

(b) 串行输出使能寄存器 m (SOEm)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	SOEm1 0/1	SOEm0 0/1

(c) 串行通道启动寄存器 m (SSm)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 0/1	SSm0 0/1

(d) 串行模式寄存器 mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0						STSmn 0		SISm0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

通道 n 的工作模式
0: 传输结束中断
1: 缓冲空中断

(e) 串行通信操作设置寄存器 mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1		EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1		SLCmn1 0	SLCmn0 0		DLSmn2 1	DLSmn1 1	DLSmn0 0/1

(f) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)

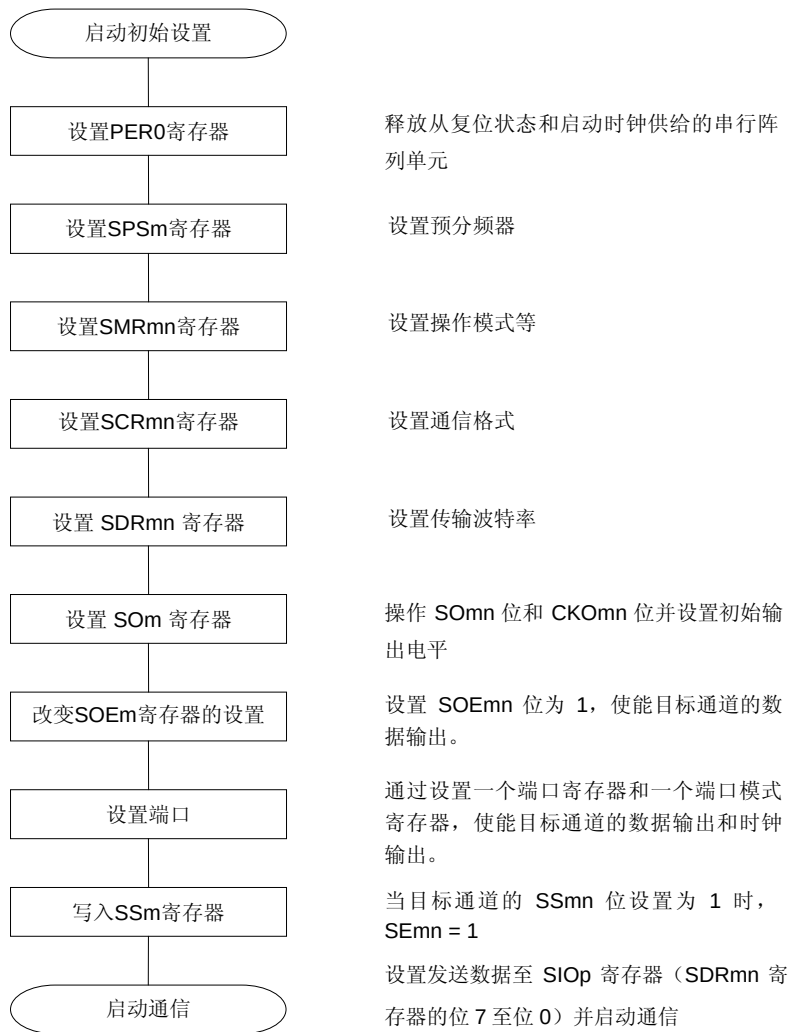
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	设置波特率							0	设置发送数据							

SIOp

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
p: CSI 编号 (p = 00, 01, 10, 20)
□: 在 CSI 主发送模式下固定设置, .□: 禁止设置 (设为初始值)
x: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)
0/1: 根据用户的使用设置为 0 或 1

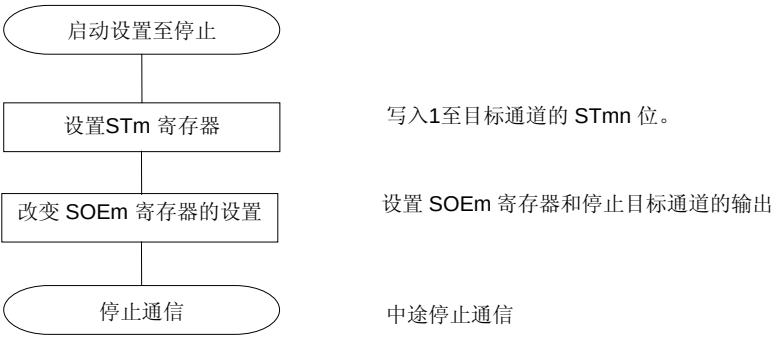
(2) 操作程序

图 11-25. 主发送的初始化设置程序



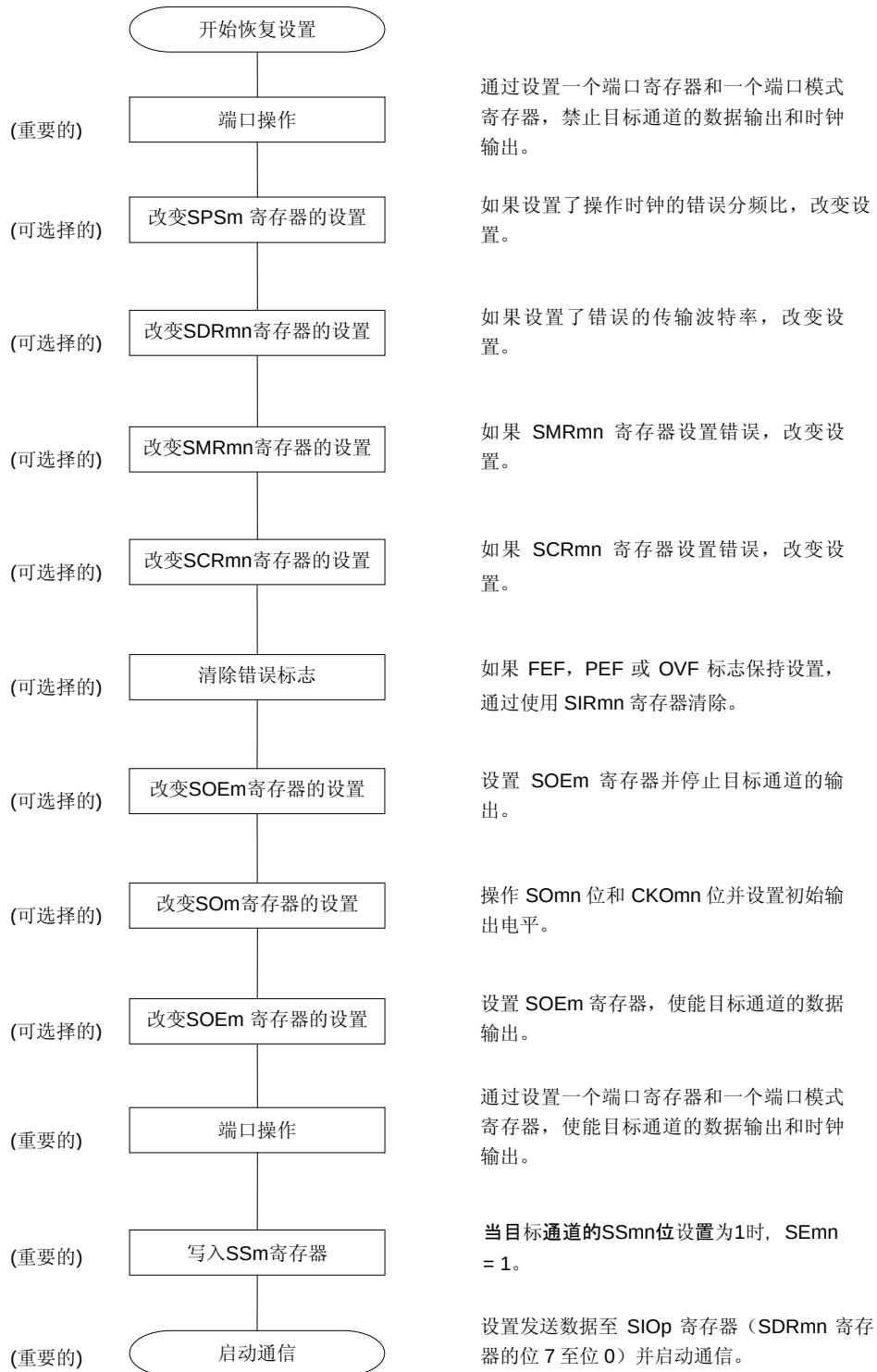
注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 11-26. 主发送的停止程序



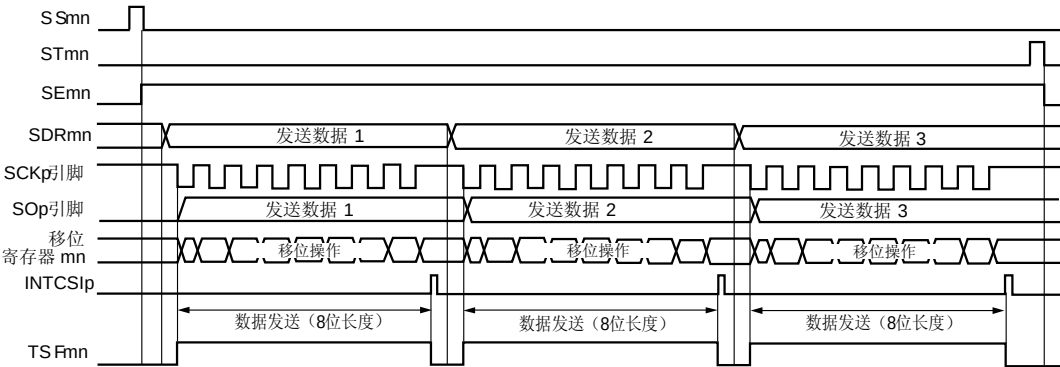
- 备注**
- 1. 即使通信停止后，引脚电平依然保持。要恢复操作，重新设置 SOm 寄存器（参见 图 11-27 恢复主发送的程序）。
 - 2. p: CSI 编号 (p = 00, 01, 10, 20)

图 11-27. 恢复主发送的程序



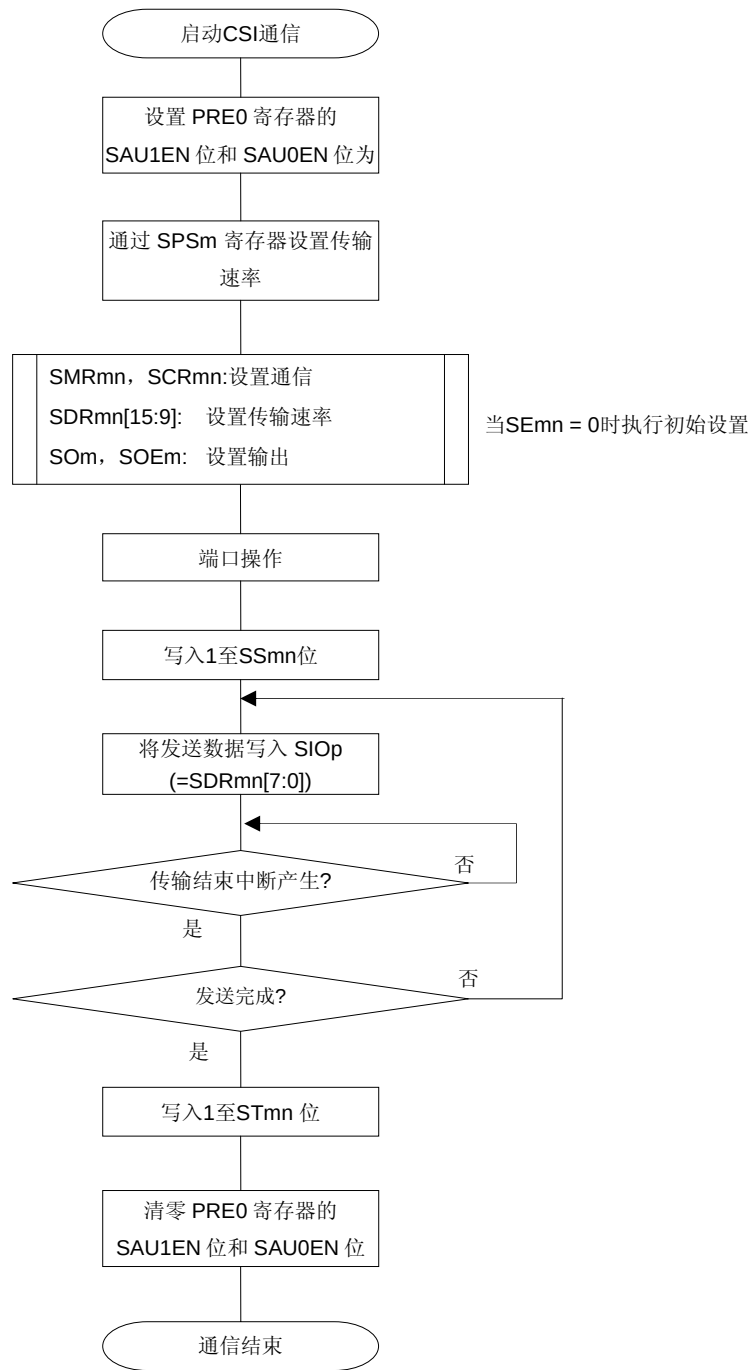
(3) 处理流程（在单发送模式下）

图 11-28. 主发送时序图（在单发送模式下）
(类型 1: DAPmn = 0, CKPmn = 0)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
p: CSI 编号 (p = 00, 01, 10, 20)

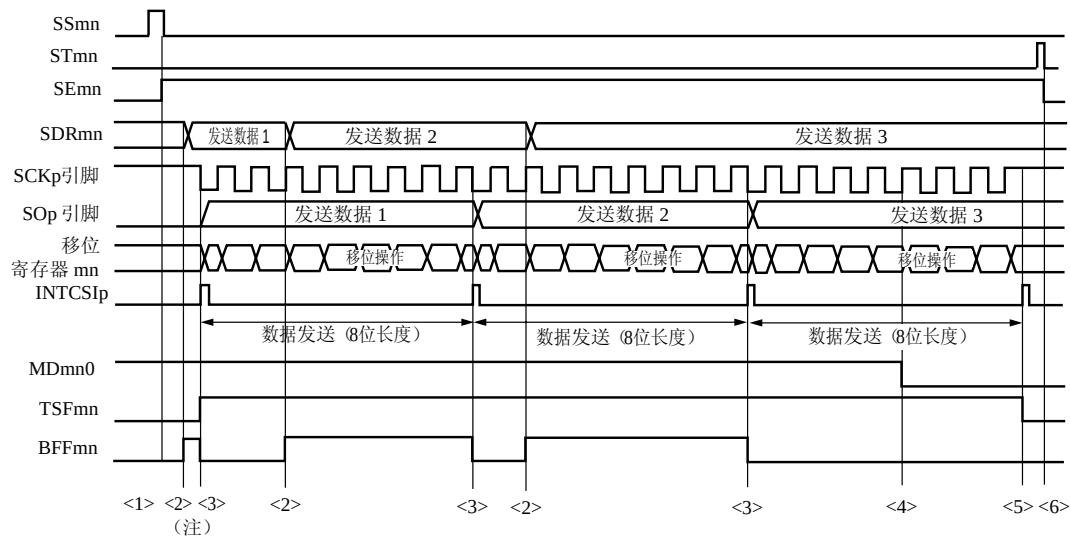
图 11-29. 主发送的流程图（在单发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

(4) 处理流程（在连续发送模式下）

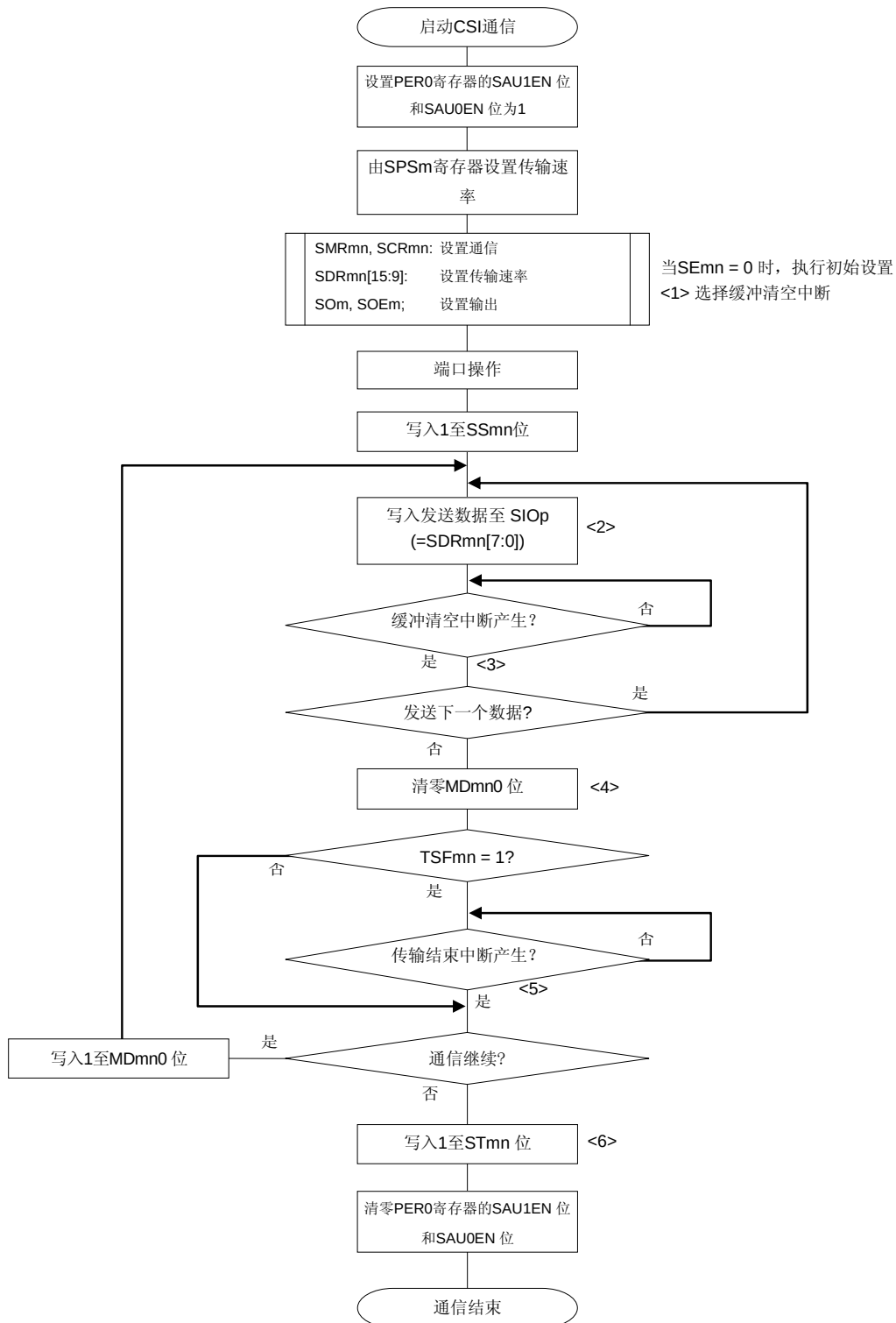
图 11-30. 主发送的时序图（在连续发送模式下）
(类型 1: DAPmn = 0, CKPmn = 0)



注意事项 即使在操作过程中，MDmn0 位也可以重写。
但是，要在最后一位开始发送之前重写它，这样它可以在最后一个发送数据的传输结束中断之前重写。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
p: CSI 编号 (p = 00, 01, 10, 20)

图 11-31. 主发送的流程图（在连续发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

备注 图中的<1>至<6> 对应图 11-30 主发送的时序图（在连续发送模式下）中的<1>至<6>。

11.5.2 主接收

主接收是 78K0R/KG3-L 输出一个传输时钟并接收其它设备的数据。

3线串行I/O	CSI00	CSI01	CSI10	CSI20
目标通道	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用引脚	$\overline{\text{SCK00}}$, SI00	$\overline{\text{SCK01}}$, SI01	$\overline{\text{SCK10}}$, SI10	$\overline{\text{SCK20}}$, SI20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	仅传输结束中断（禁止设置缓冲空中断。）			
错误检测标志	仅溢出错误检测标志 (OVFmn)			
传输数据长度	7 或 8 位			
传输速率	最大值 $f_{\text{CLK}}/4$ [MHz], 最小值 $f_{\text{CLK}}/(2 \times 2^{11} \times 128)$ [MHz] ^注 f_{CLK} : 系统时钟频率			
数据相位	通过 DAPmn 位可选择 • DAPmn = 0: 从串行时钟开始操作时, 数据输入开始 • DAPmn = 1: 在串行时钟操作开始之前半个时钟, 数据输入开始。			
时钟相位	通过 CKPmn 位选择 • CKPmn = 0: 正向 • CKPmn = 1: 反向			
数据方向	首先是MSB 或者LSB			

注 请在满足上面条件和电气特性中的 AC 特性的范围内, 使用这个操作（参见 第 27 章 电气特性（目标值））。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10

(1) 寄存器设置

图 11-32. 用于 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 的主接收的寄存器内容的示例

(a) 串行输出寄存器 m (SOMm)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOMm	0	0	0	0	1	CKOm2 0/1	CKOm1 0/1	CKOm0 0/1	0	0	0	0	1	SOM2 ×	SOM1 ×	SOM0 ×

如果数据相位是正向的(CKPmn = 0)，则当这些位为 1 时，通信开始。如果相位是反向的(CKPmn = 1)，则当这些位为 0 时，通信开始。

(b) 串行输出使能寄存器 m (SOEm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 ×	SOEm0 ×

(c) 串行通道启动寄存器 m (SSm)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 0/1	SSm0 0/1

(d) 串行模式寄存器 mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SIsmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

通道 n 的工作模式
0: 传输结束中断

(e) 串行通信操作设置寄存器 mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	DLSmn2 1	DLSmn1 1	DLSmn0 0/1

(f) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	设置波特率							0	接收数据寄存器 (写入 FFH 作为虚拟数据)							

SIOp

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10

p: CSI 编号 (p = 00, 01, 10, 20)

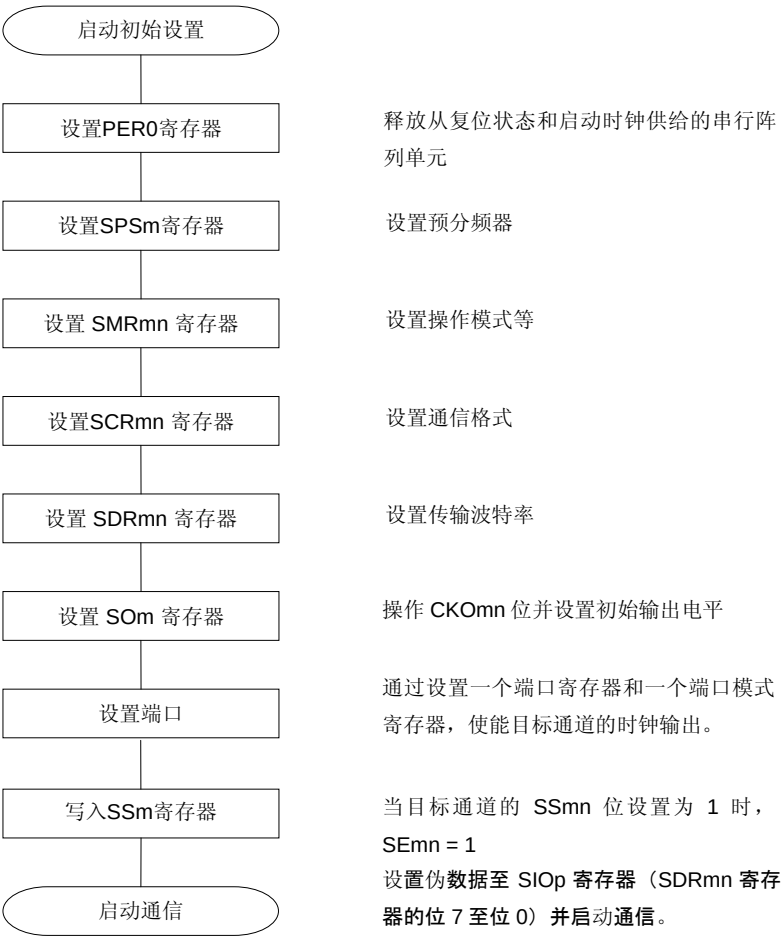
□.: 在 CSI 主接收模式下固定设置, ■.: 禁止设置 (设为初始值)

x: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1

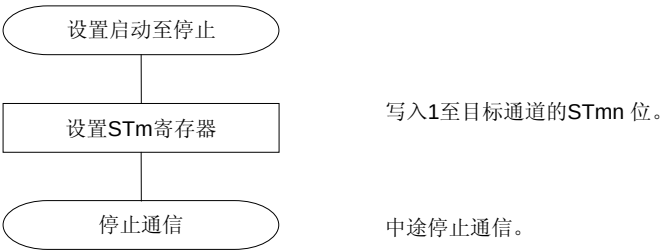
(2) 操作程序

图 11-33. 主接收的初始化程序



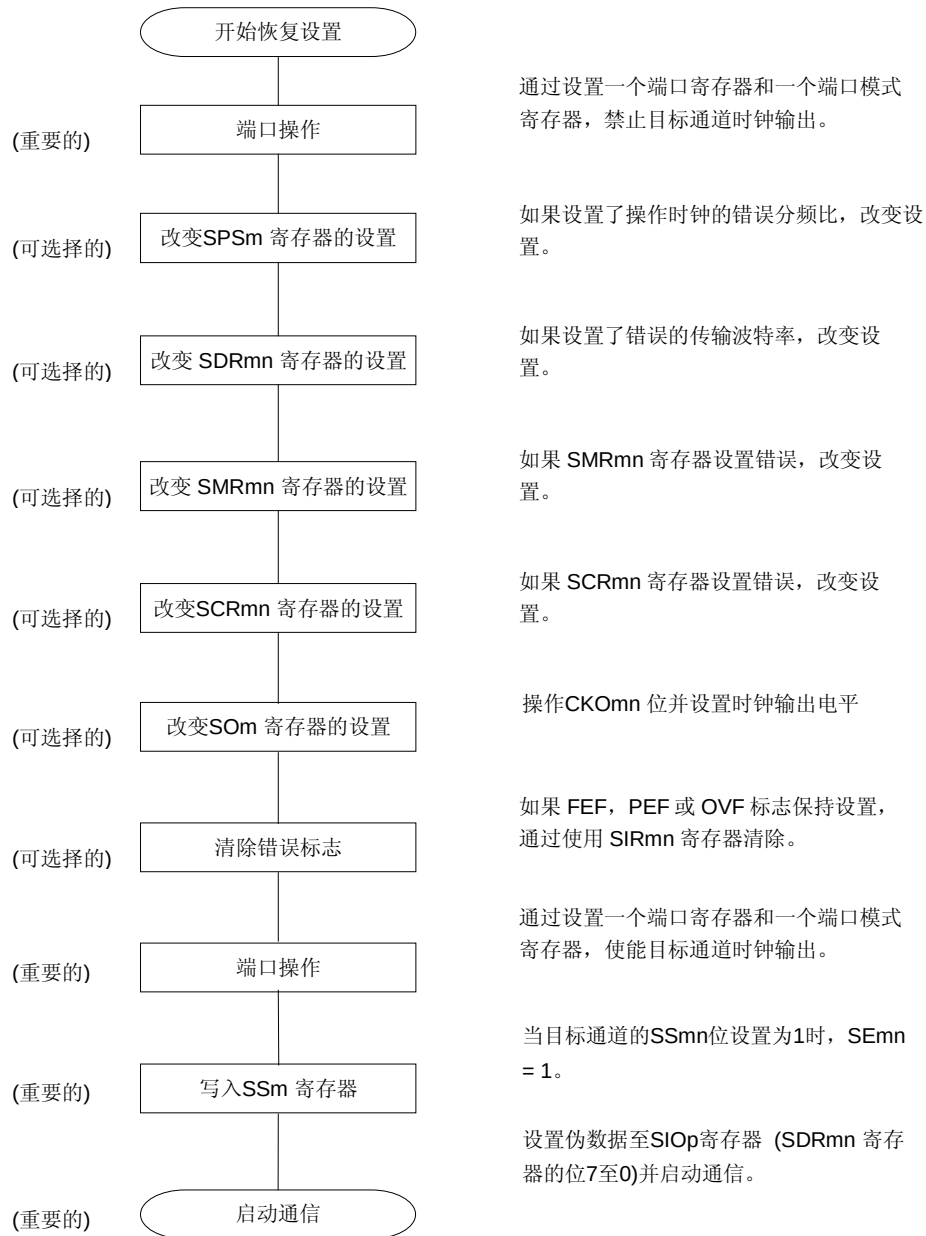
注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 11-34. 主接收的停止程序



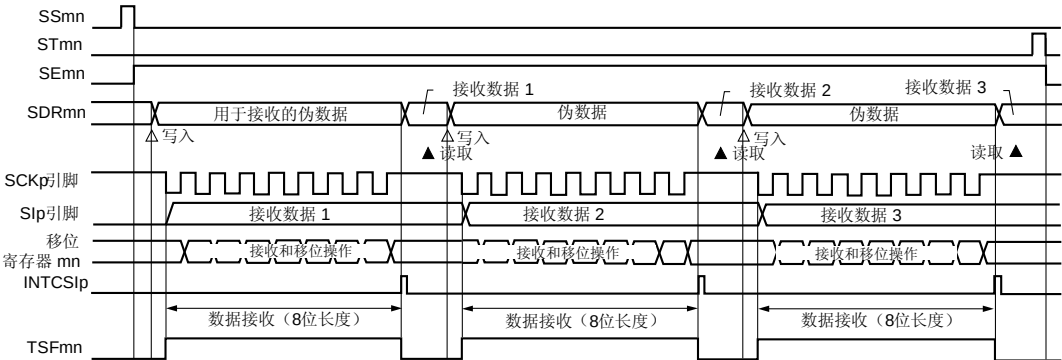
备注 即使通信停止后，引脚电平依然保持。要恢复操作，重新设置 SOM 寄存器（参见 图 11-35 恢复主接收的程序）。

图 11-35. 恢复主接收的程序



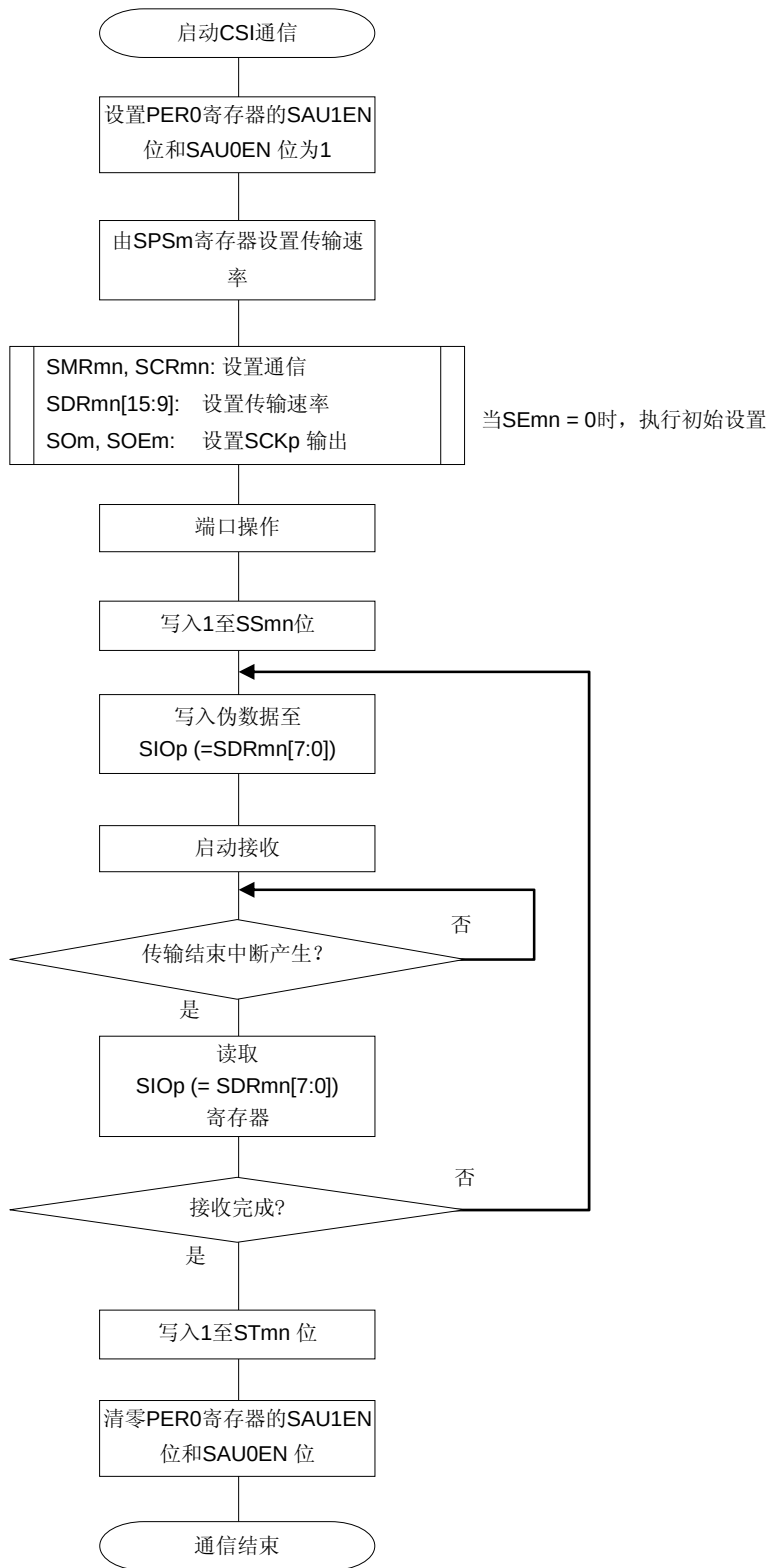
(3) 处理流程（在单接收模式下）

图 11-36. 主接收的时序图（在单接收模式下）
(类型 1: DAPmn = 0, CKPmn = 0)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
p: CSI 编号 (p = 00, 01, 10, 20)

图 11-37. 主接收的流程图（在单接收模式下）



注意事项 在设置 PER0 寄存器为 1 后, 确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

11.5.3 主发送/接收

主发送/接收是 78K0R/KG3-L 输出一个传输时钟和向/从另外一个设备发送/接收数据。

3线串行I/O	CSI00	CSI01	CSI10	CSI20
目标通道	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用引脚	$\overline{\text{SCK00}}$, SI00, SO00	$\overline{\text{SCK01}}$, SI01, SO01	$\overline{\text{SCK10}}$, SI10, SO10	$\overline{\text{SCK20}}$, SI20, SO20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	可以选择传输结束中断（在单发送模式下）或者缓冲空中断（在连续发送模式下）			
错误检测标志	仅溢出错误检测标志 (OVFmn)			
发送数据长度	7 或 8 位			
传输速率	最大值 $f_{\text{CLK}}/4$ [MHz], 最小值 $f_{\text{CLK}}/(2 \times 2^{11} \times 128)$ [MHz] ^注 f_{CLK} : 系统时钟频率			
数据相位	通过DAPmn位可选择 • DAPmn = 0: 从串行时钟操作开始时,数据I/O开始。 • DAPmn = 1: 在串行时钟操作开始之前半个时钟, 数据I/O开始。			
时钟相位	通过 CKPmn位选择 • CKPmn = 0: 正向 • CKPmn = 1: 反向			
数据方向	首先是MSB或者LSB			

注 请在满足上面条件和电气特性中的 AC 特性的范围内, 使用这个操作（参见 第 27 章 电气特性（目标值））。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10

(1) 寄存器设置

图 11-38. 用于 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 的主发送/接收的寄存器内容的示例

(a) 串行输出寄存器 m (SOM)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	1	CKOm2 0/1	CKOm1 0/1	CKOm0 0/1	0	0	0	0	1	SOM2 0/1	SOM1 0/1	SOM0 0/1

如果数据相位是正向的(CKPmn = 0)，则当这些位为 1 时，通信开始。如果相位是反向的(CKPmn = 1)，则当这些位为 0 时，通信开始。

(b) 串行输出使能寄存器 m (SOEm)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	SOEm1 0/1	SOEm0 0/1

(c) 串行通道启动寄存器 m (SSm)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 0/1	SSm0 0/1

(d) 串行模式寄存器 mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SlSmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

通道 n 的工作模式
0: 传输结束中断
1: 缓冲空中断

(e) 串行通信操作设置寄存器 mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	DLSmn2 1	DLSmn1 1	DLSmn0 0/1

(f) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOP)

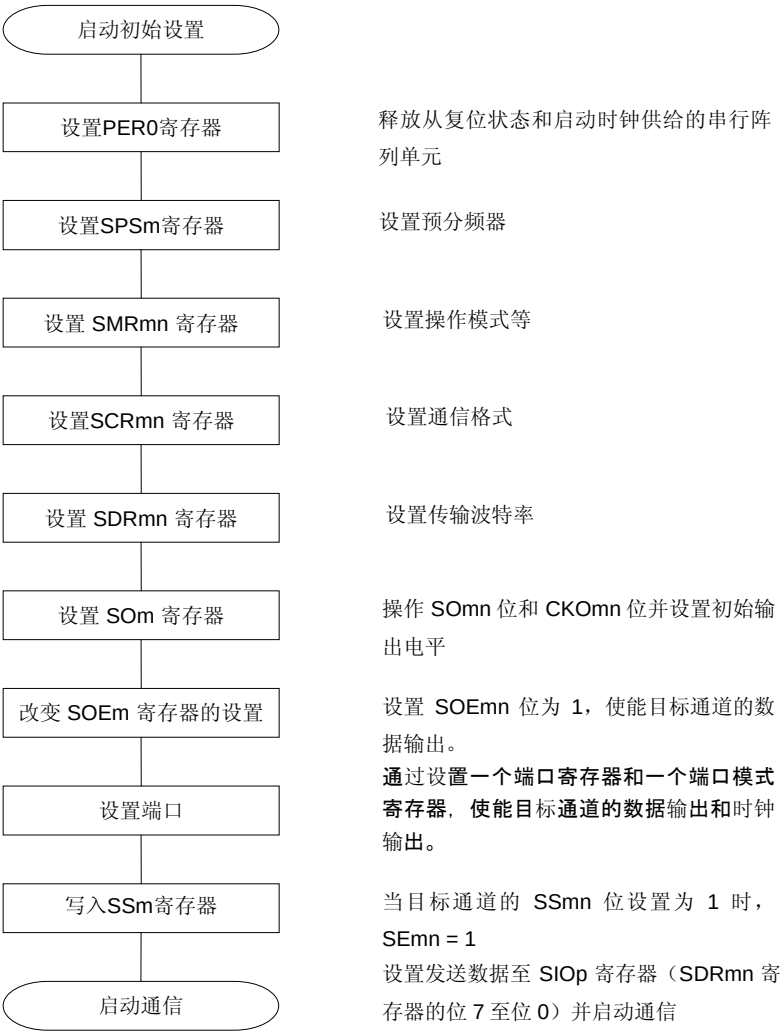
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	设置波特率								0	设置发送数据/接收数据寄存器						

SIOP

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
p: CSI 编号 (p = 00, 01, 10, 20)
□: CSI 主发送/接收模式为固定设置, ■: 禁止设置 (设为初始值)
×: 不能用于该模式的位 (当不能用于任何模式时设置初始值)
0/1: 根据用户使用设定为 0 或 1

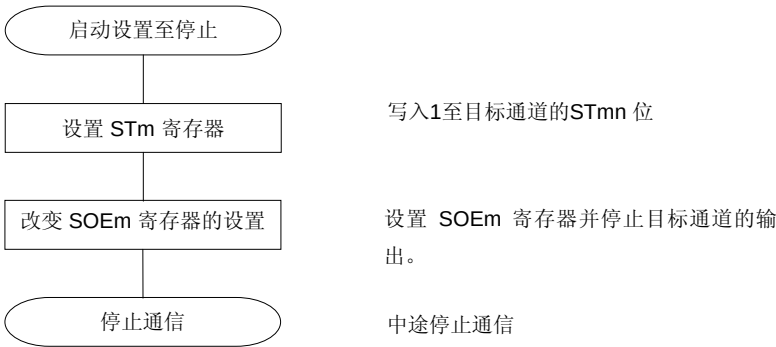
(2) 操作程序

图 11-39. 主发送/接收的初始化程序



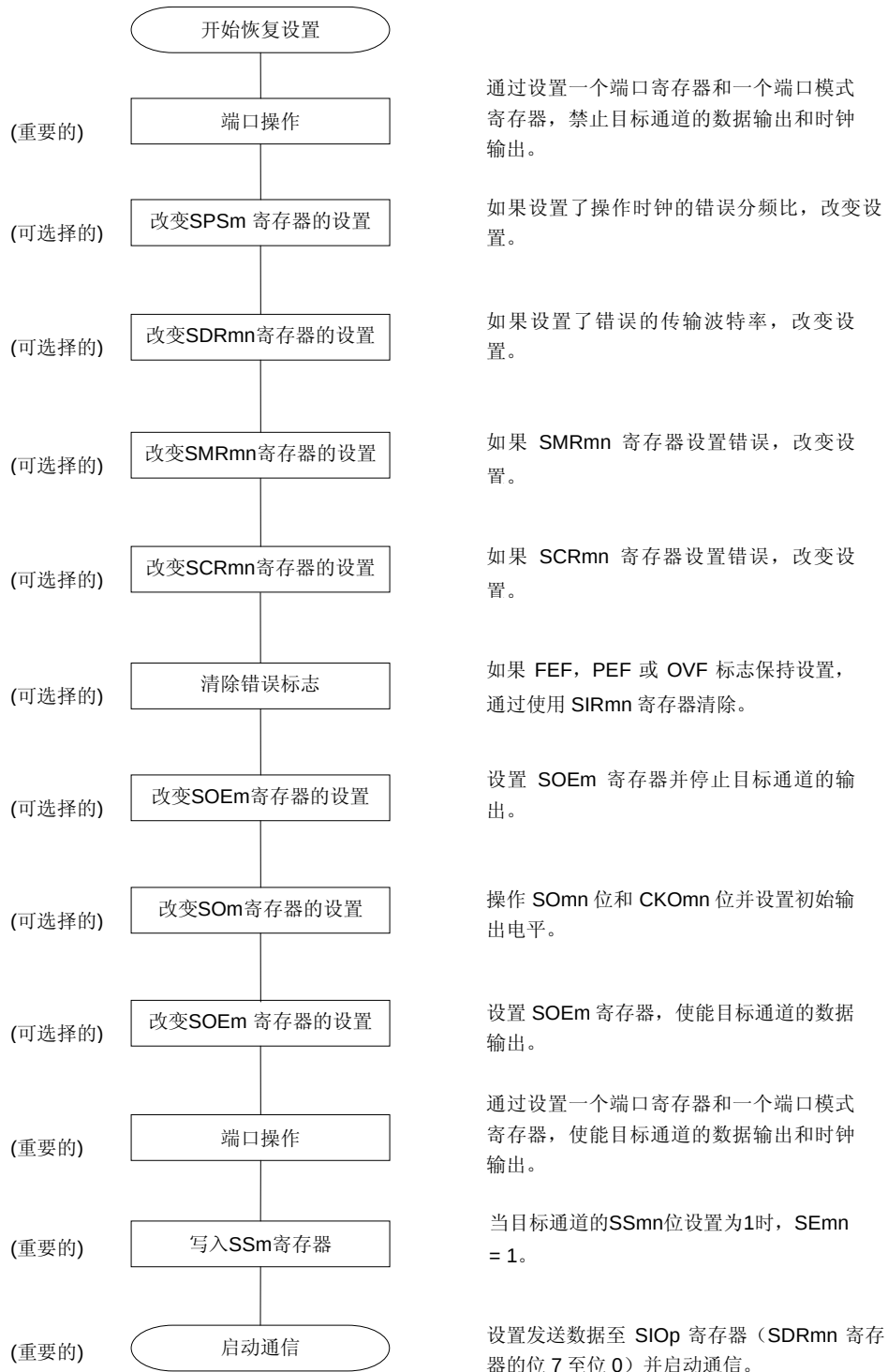
注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 11-40. 主发送/接收的停止程序



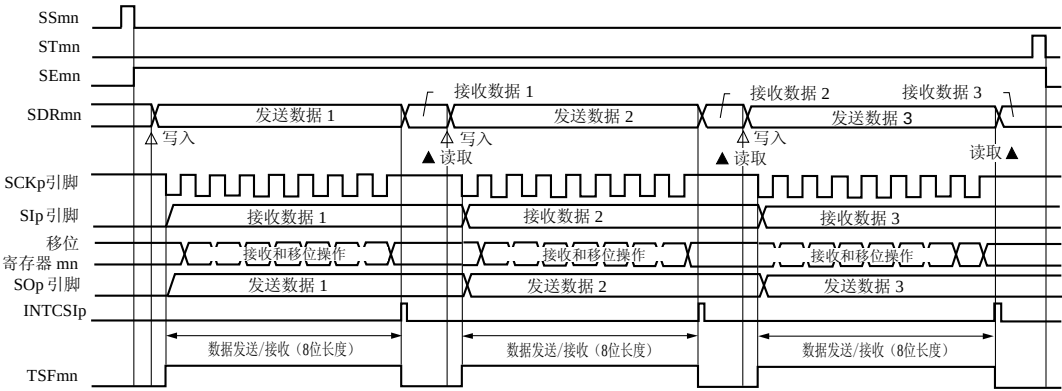
备注 即使通信停止后，引脚电平依然保持。要恢复操作，重新设置 S0mn 寄存器 (参见 图 11-41 恢复主发送/接收的程序)。

图 11-41. 恢复主发送/接收的程序



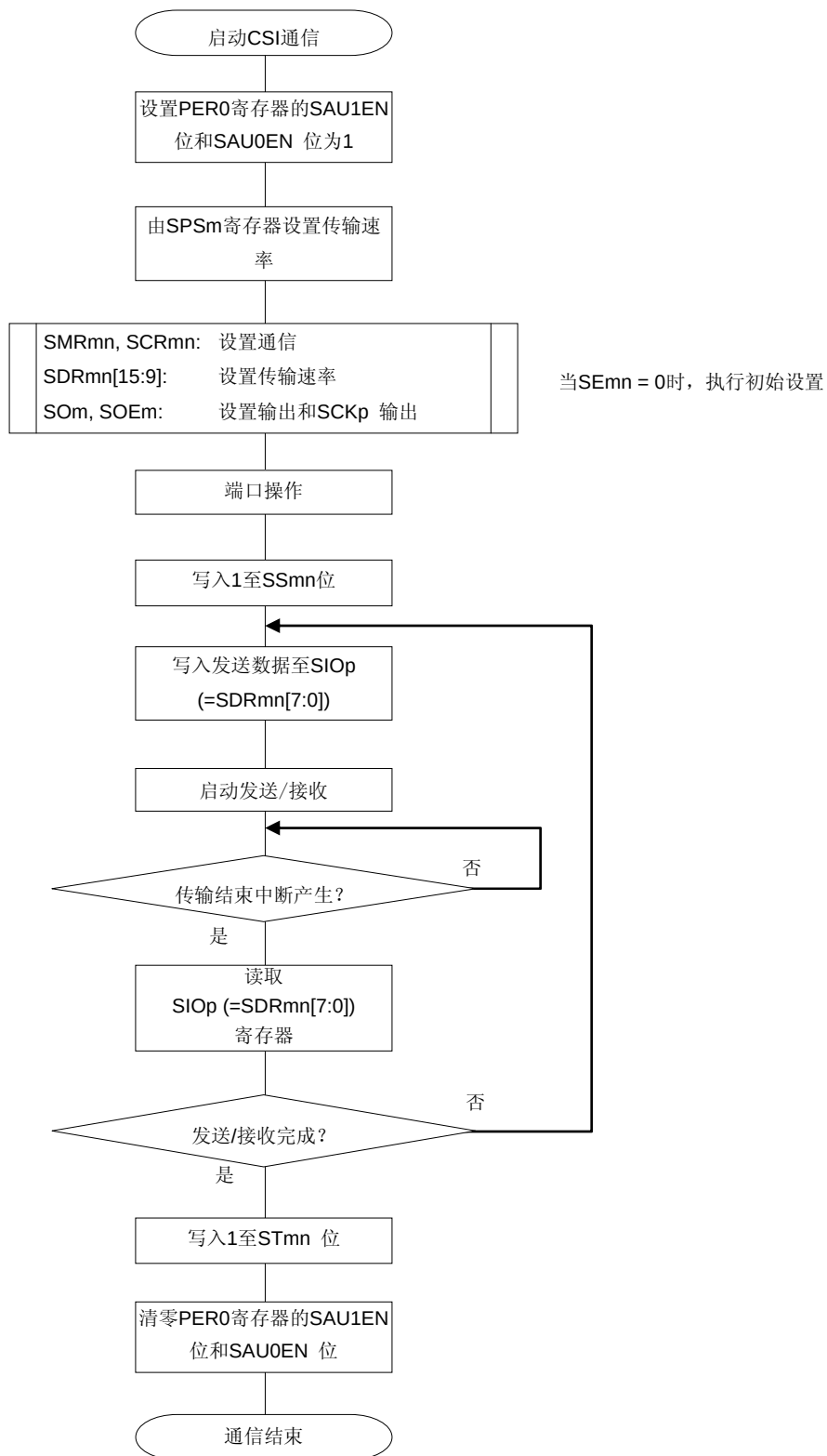
(3) 处理流程（在单发送/接收模式下）

图 11-42. 主发送/接收的时序图（在单发送/接收模式下）
(类型 1: DAPmn = 0, CKPmn = 0)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
p: CSI 编号 (p = 00, 01, 10, 20)

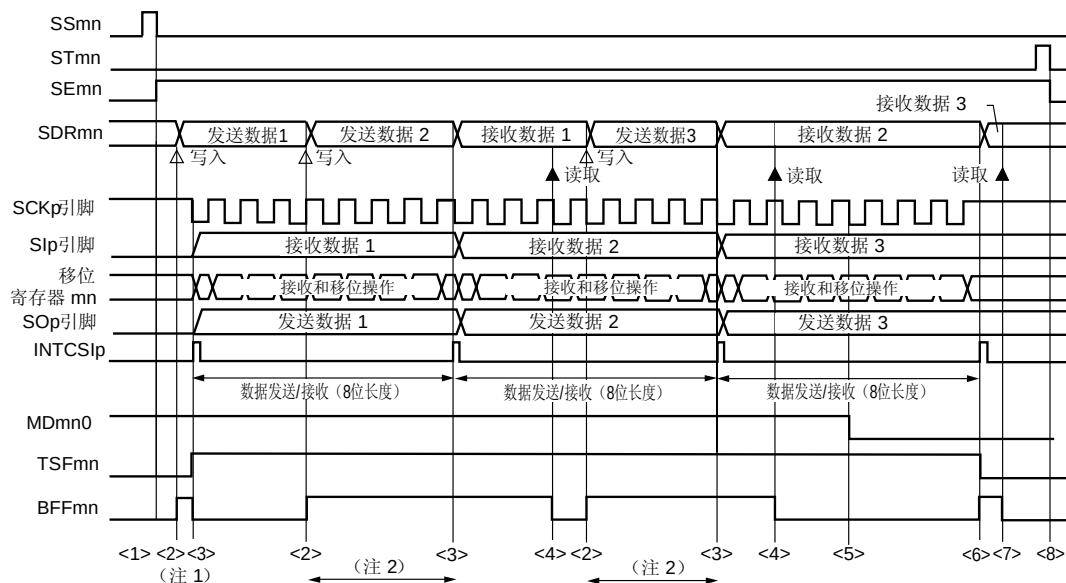
图 11-43. 主发送/接收的流程图（在单发送/接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

(4) 处理流程（在连续发送/发送模式下）

图 11-44. 主发送/发送的时序图（在连续发送/接收模式下）
(类型 1: DAPmn = 0, CKPmn = 0)



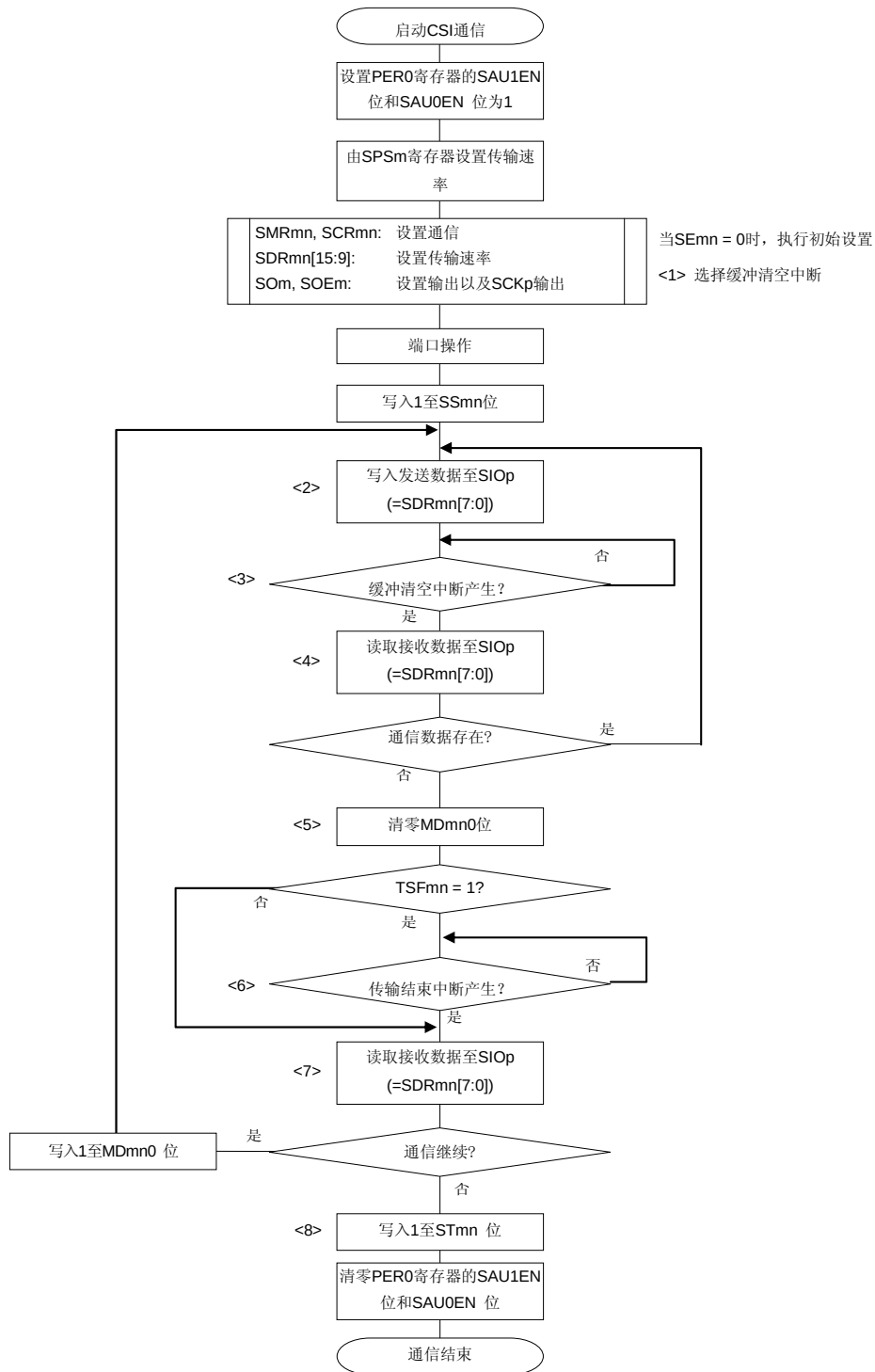
- 注
1. 在 BFFmn = 1 的情况下，当发送数据写入 SDRmn 寄存器时，发送数据可以重写。
 2. 在这期间，发送数据可以通过读取 SDRmn 寄存器来读取。此时发送操作不受影响。

注意事项 即使在操作过程中，MDmn0 位也可以重写。

但是，要在最后一位开始发送之前重写它，这样它可以在最后一个发送数据的传输结束中断之前重写。

- 备注
1. 图中的<1>至<8> 对应图 11-45 主发送/接收的流程图（在连续发送/接收模式下）中的<1>至<8>。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
p: CSI 编号 (p = 00, 01, 10, 20)

图 11-45. 主发送/接收的流程图（在连续发送/接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

备注 图中的<1>至<8> 对应图 11-44 主发送/接收的时序图（在连续发送/接收模式下）中的<1>至<8>。

11.5.4 从发送

从发送是在传输时钟由另外的设备输入的状态下，78K0R/KG3-L 发送数据到另一个设备。

3线串行I/O	CSI00	CSI01	CSI10	CSI20
目标通道	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用引脚	$\overline{\text{SCK00}}$, SO00	$\overline{\text{SCK01}}$, SO01	$\overline{\text{SCK10}}$, SO10	$\overline{\text{SCK20}}$, SO20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	可以选择传输结束中断（在单传输模式下）或者缓冲空中断（在连续传输模式下）			
错误检测标志	仅溢出错误检测标志 (OVFmn)			
发送数据长度	7 或 8 位			
传输速率	最大值: $f_{\text{MCK}}/6$ [MHz] ^{注1, 2}			
数据相位	通过DAPmn位可选择 • DAPmn = 0: 从串行时钟开始操作时，数据输出开始。 • DAPmn = 1: 在串行时钟操作开始之前半个时钟，数据输出开始。			
时钟相位	通过 CKPmn位选择 • CKPmn = 0: 正向 • CKPmn = 1: 反向			
数据方向	首先是MSB 或者LSB			

- 注**
1. 因为外部串行时钟输入到引脚 $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$ 和 $\overline{\text{SCK20}}$ 是内部采样并使用，最大发送速率为 $f_{\text{MCK}}/6$ [MHz]。
 2. 请在满足上面条件和电气特性中的 AC 特性的范围内，使用这个操作（参见 第 27 章 电气特性（目标值））。

- 备注**
1. f_{MCK} : 目标通道的工作时钟(MCK)频率
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10

(1) 寄存器设置

图 11-46. 用于 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 的从发送的寄存器内容的示例

(a) 串行输出寄存器 m (SOM)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	1	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	1	SOM2 0/1	SOM1 0/1	SOM0 0/1

(b) 串行输出使能寄存器 m (SOEm)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	SOEm1 0/1	SOEm0 0/1

(c) 串行通道启动寄存器 m (SSm)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 0/1	SSm0 0/1

(d) 串行模式寄存器 mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

通道 n 的工作模式
0: 传输结束中断
1: 缓冲空中断

(e) 串行通信操作设置寄存器 mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	DLSmn2 1	DLSmn1 1	DLSmn0 0/1

(f) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	设置波特率							0	设置发送数据							

SIOp

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10

p: CSI 编号 (p = 00, 01, 10, 20)

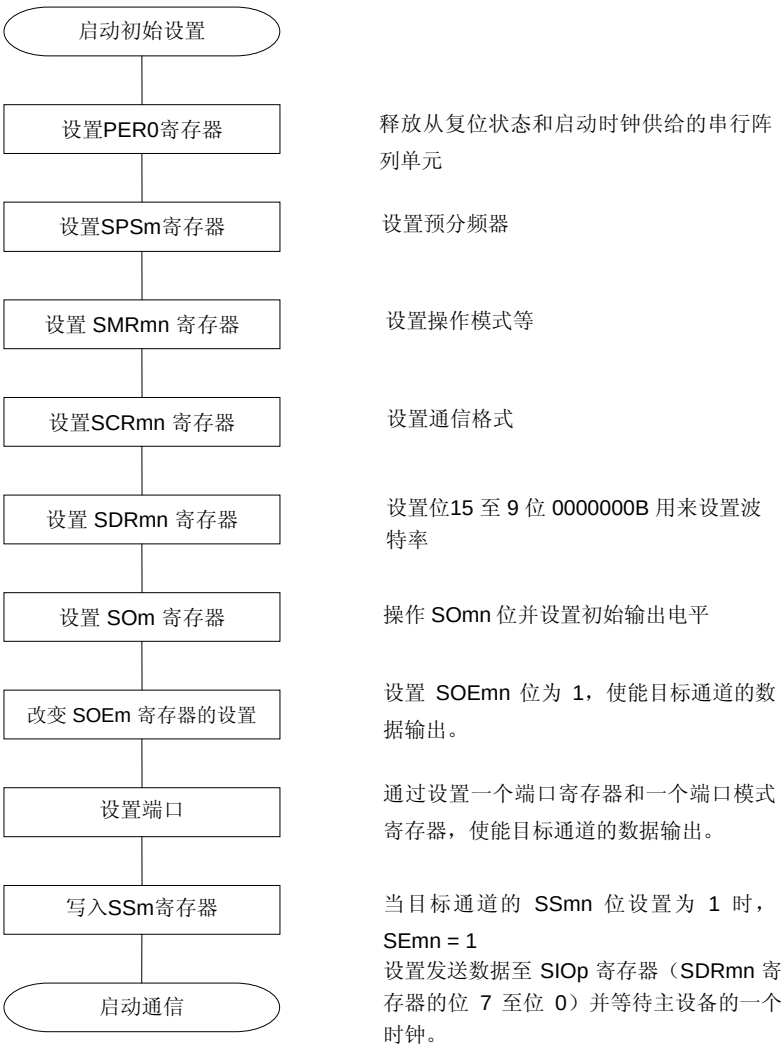
□: 在 CSI 从发送模式下固定设置, .□: 禁止设置 (设为初始值)

x: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1

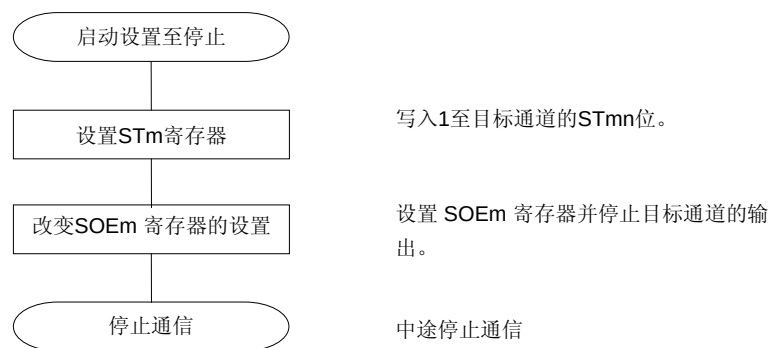
(2) 操作程序

图 11-47. 从发送的初始化程序



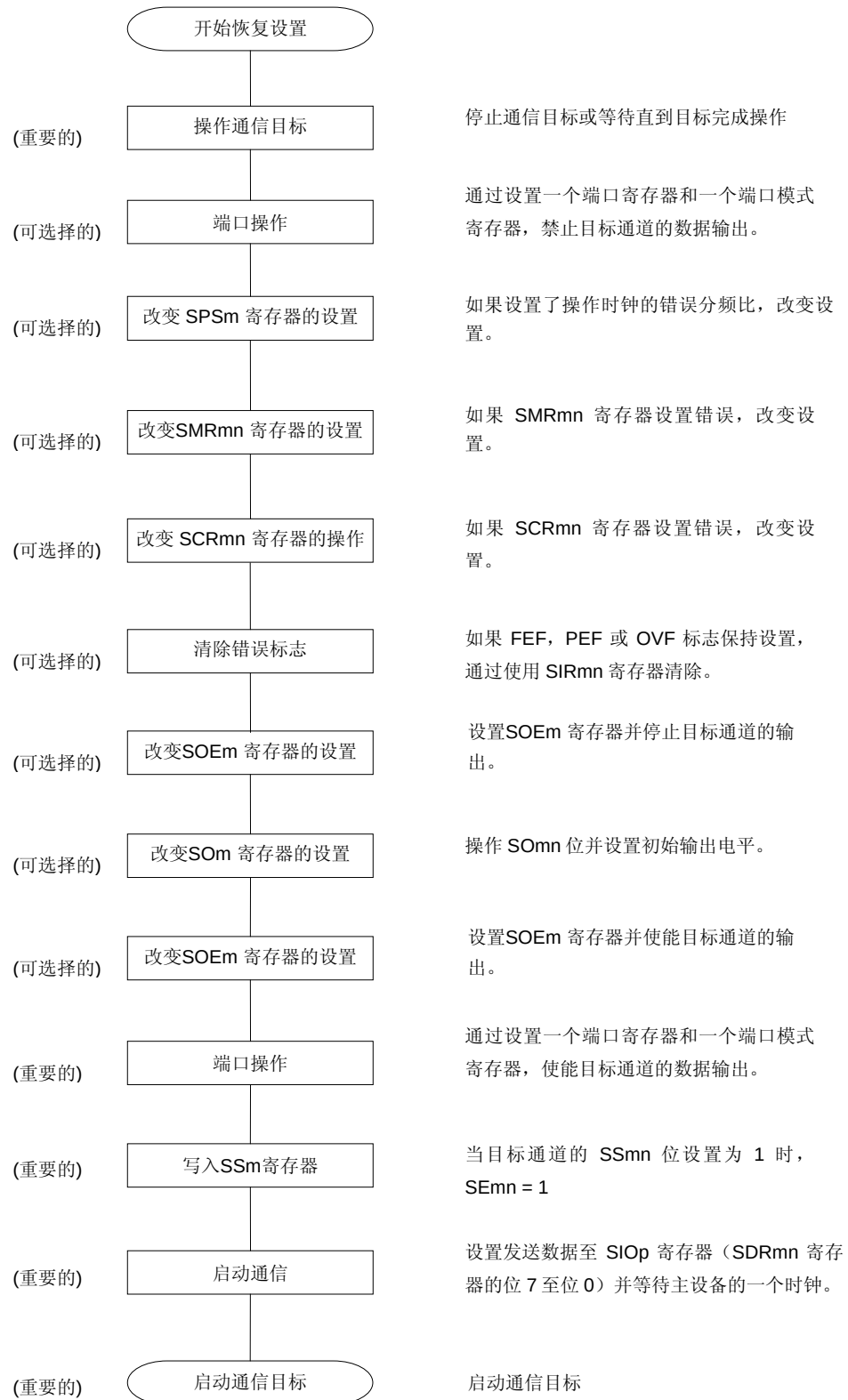
注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 11-48. 从发送的停止程序



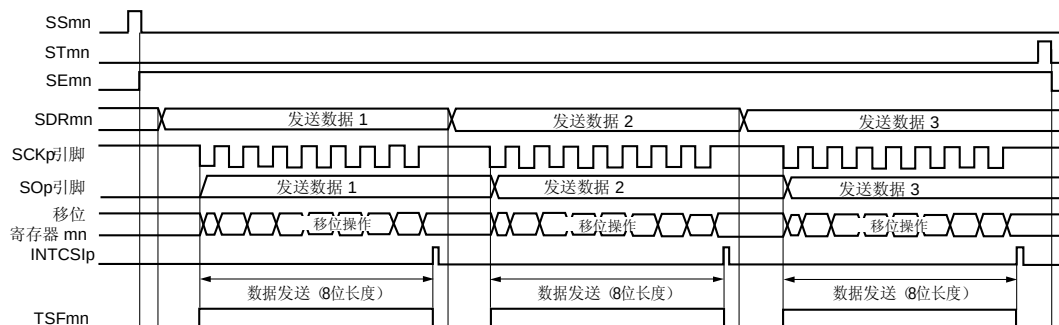
备注 即使通信停止后，引脚电平依然保持。要恢复操作，重新设置 S0m 寄存器（参见 图 11-49 恢复从发送的程序）。

图 11-49. 恢复从发送的程序



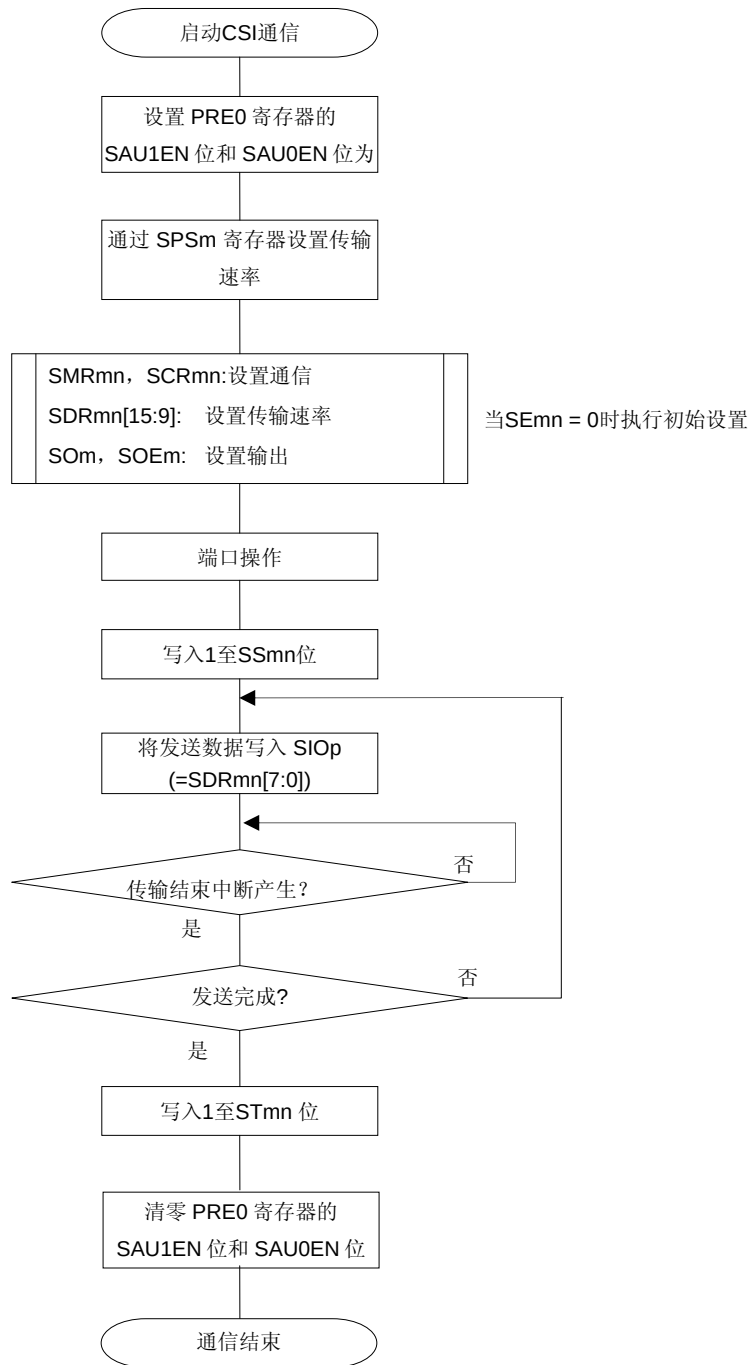
(3) 处理流程（在单发送模式下）

图 11-50. 从发送的时序图（在单发送模式下）
 (类型 1: DAPmn = 0, CKPmn = 0)



备注 m: 单元编号 ($m = 0, 1$), n: 通道编号 ($n = 0$ 至 2), $mn = 00$ 至 $02, 10$
 p: CSI 编号 ($p = 00, 01, 10, 20$)

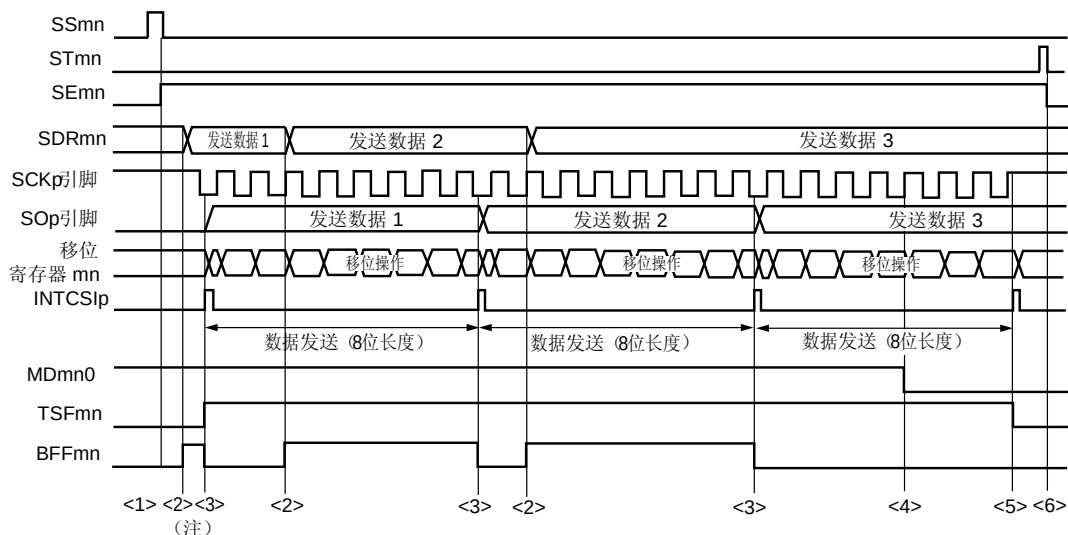
图 11-51. 从发送的流程图（在单发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

(4) 处理流程（在连续发送模式下）

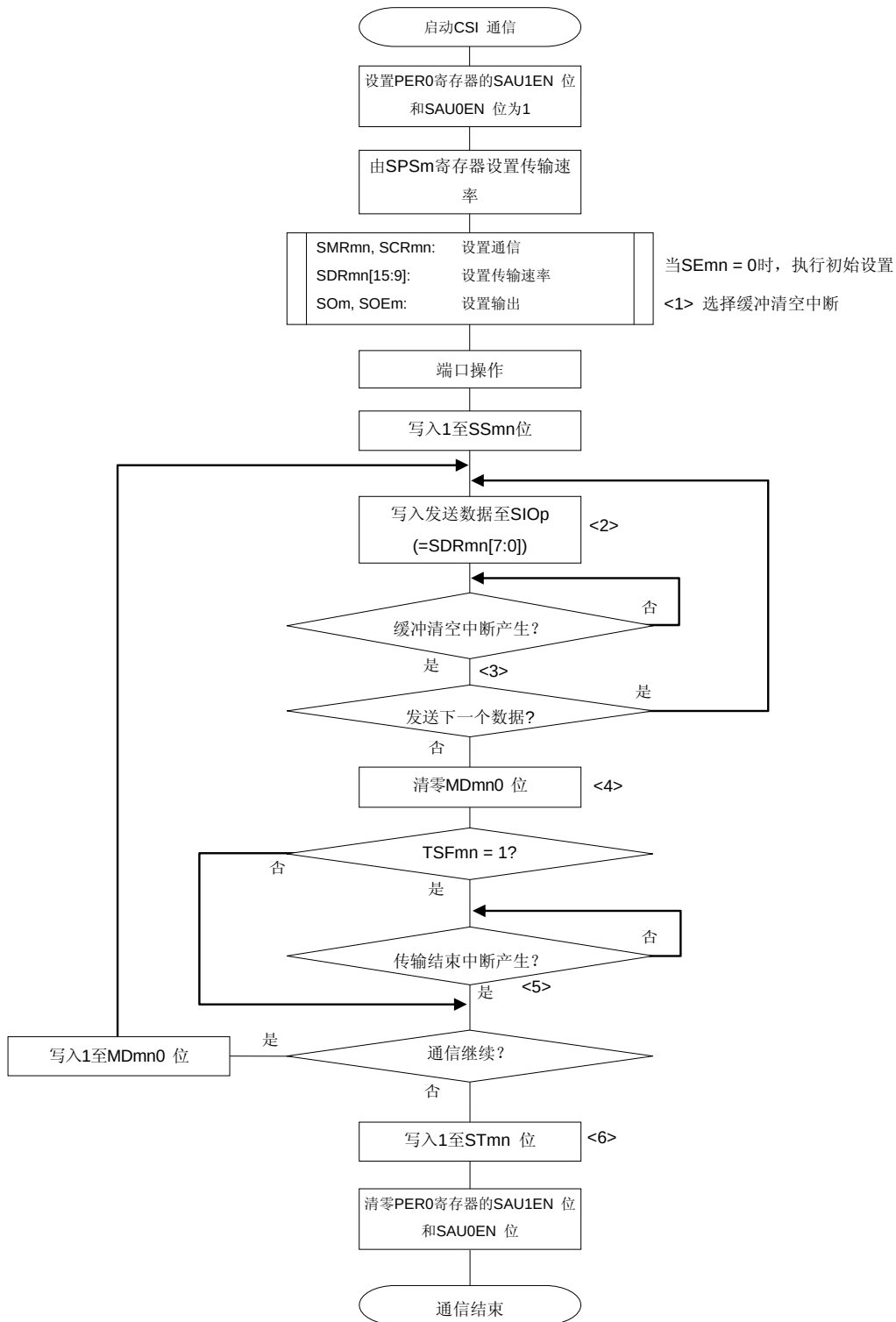
图 11-52. 从发送的时序图（在连续发送模式下）
(类型 1: DAPmn = 0, CKPmn = 0)



注 在 BFFmn = 1 的情况下，当发送数据写入 SDRmn 寄存器时，发送数据可以重写。

注意事项 即使在操作过程中，MDmn0 位也可以重写。但是必须在最后一位开始发送之前重写它。

图 11-53. 从发送的流程图（在连续发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

备注 图中的<1>至<6> 对应图 11-52 从发送的时序图（在连续发送模式下）中的<1>至<6>。

11.5.5 从接收

从接收是在传输时钟从另外一个设备输入的状态下，78K0R/KG3-L 从另外一个设备接收数据。

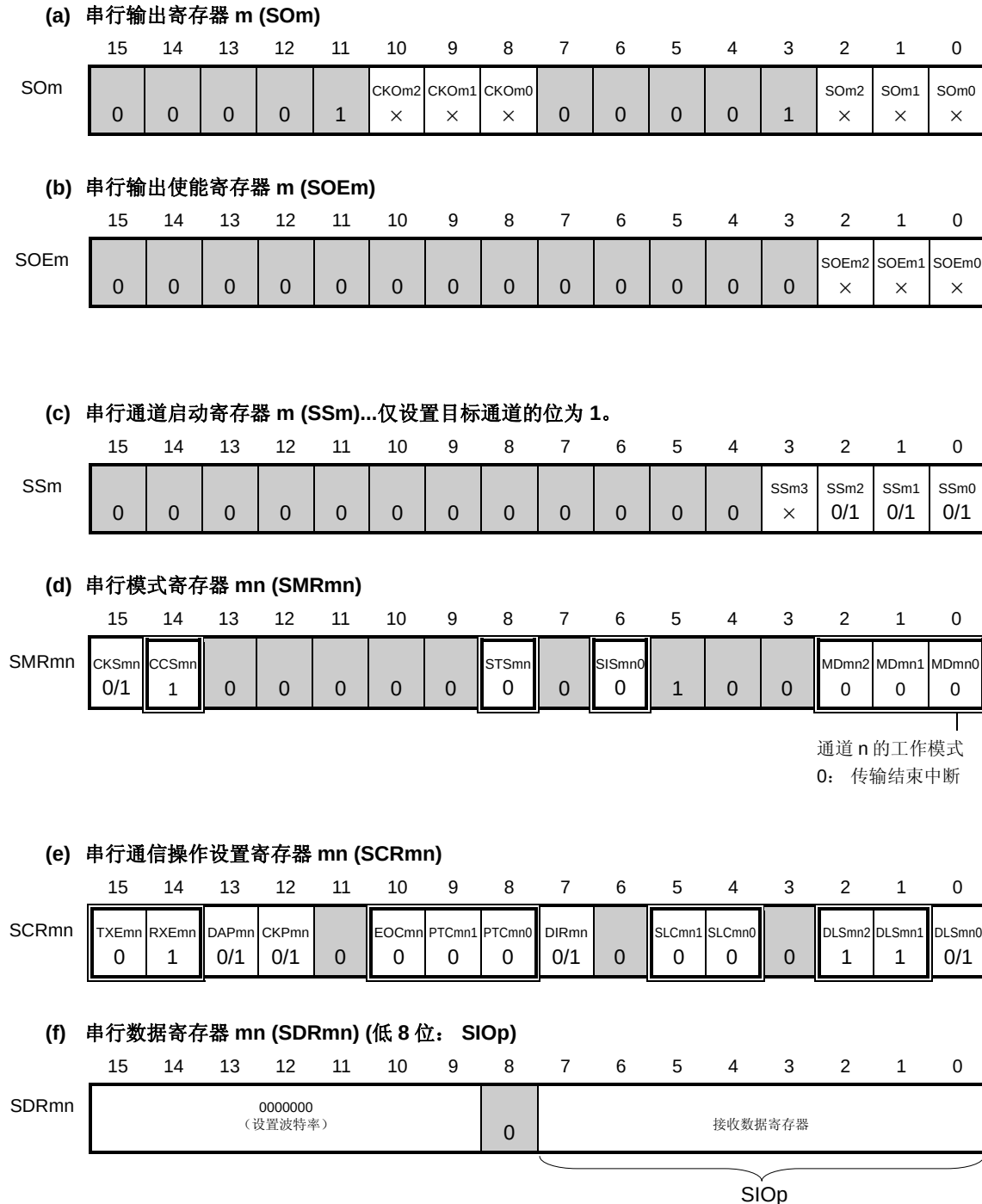
3线串行I/O	CSI00	CSI01	CSI10	CSI20
目标通道	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用引脚	$\overline{\text{SCK00}}$, SI00	$\overline{\text{SCK01}}$, SI01	$\overline{\text{SCK10}}$, SI10	$\overline{\text{SCK20}}$, SI20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	仅传输结束中断（禁止设置缓冲空中断。）			
错误检测标志	仅溢出错误检测标志 (OVFmn)			
发送数据长度	7 或 8 位			
传输速率	最大值: $f_{\text{MCK}}/6$ [MHz] ^{注1, 2}			
数据相位	通过DAPmn位可选择 • DAPmn = 0: 从串行时钟开始操作时，数据输入开始。 • DAPmn = 1: 在串行时钟操作开始之前半个时钟，数据输入开始。			
时钟相位	通过 CKPmn位选择 • CKPmn = 0: 正向 • CKPmn = 1: 反向			
数据方向	首先是MSB或者LSB			

- 注**
1. 因为外部串行时钟输入到引脚 $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$ 和 $\overline{\text{SCK20}}$ 是内部采样并使用，最大传输速率为 $f_{\text{MCK}}/6$ [MHz]。
 2. 请在满足上面条件和电气特性中的 AC 特性的范围内，使用这个操作（参见 第 27 章 电气特性（目标值））。

- 备注**
1. f_{MCK} : 目标通道的工作时钟(MCK)频率
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10

(1) 寄存器设置

图 11-54. 用于 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 的从接收的寄存器内容的示例



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
p: CSI 编号 (p = 00, 01, 10, 20)
.: 在 CSI 从接收模式下固定设置, .: 禁止设置 (设为初始值)
x: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)
0/1: 根据用户的使用设置为 0 或 1

(2) 操作程序

图 11-55. 从接收的初始化程序



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 11-56. 从接收的停止程序

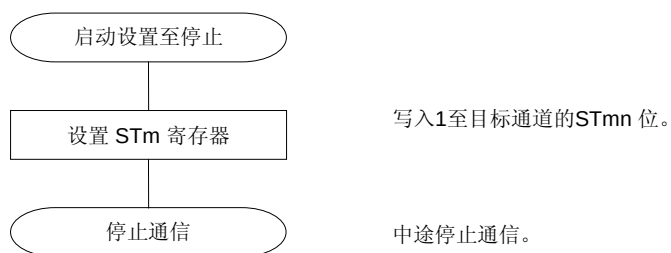
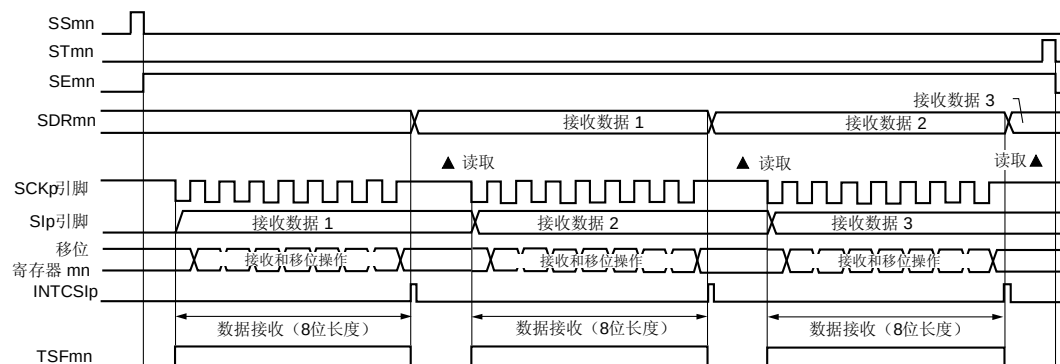


图 11-57. 恢复从接收的程序



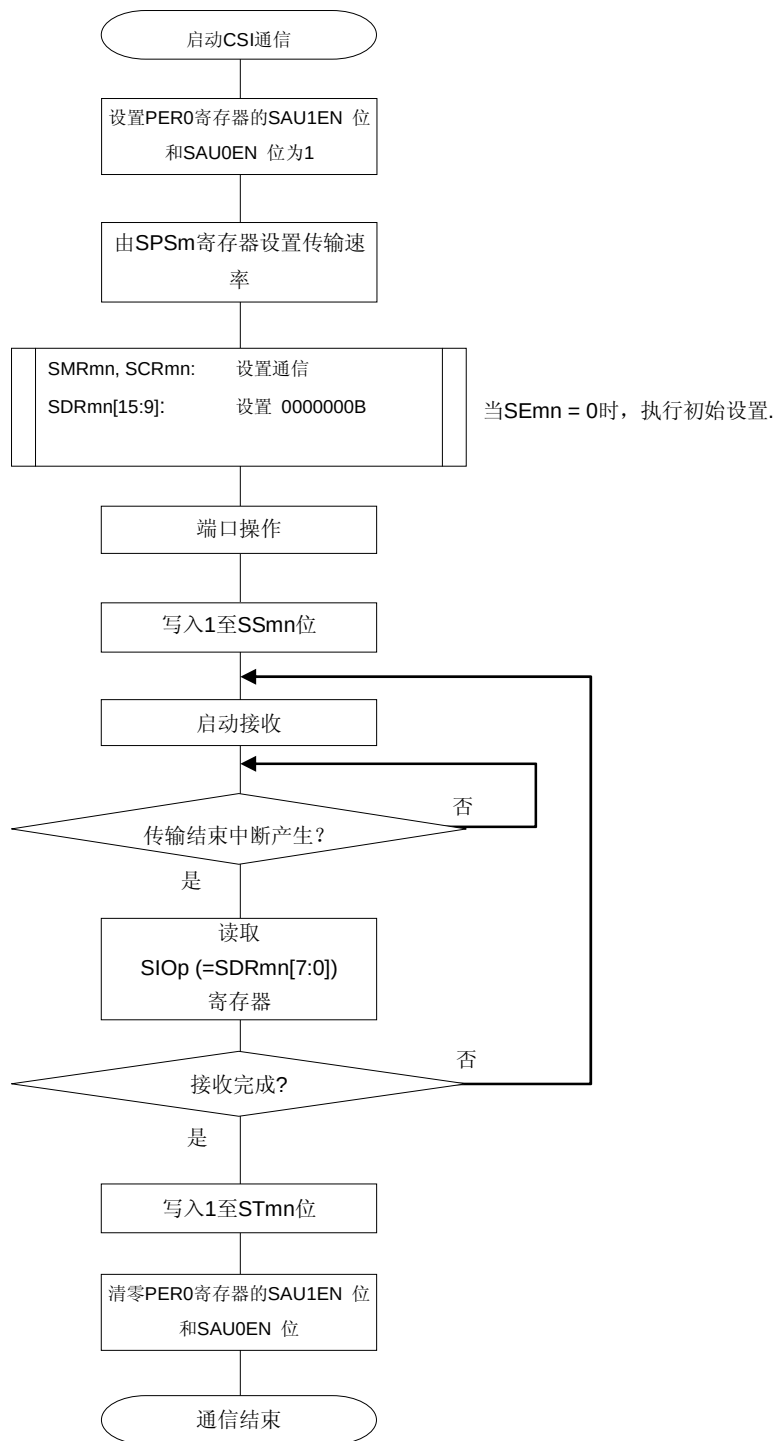
(3) 处理流程（在单接收模式下）

图 11-58. 从接收的时序图（在单接收模式下）
 (类型 1: DAPmn = 0, CKPmn = 0)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
 p: CSI 编号 (p = 00, 01, 10, 20)

图 11-59. 从接收的流程图 (在单接收模式下)



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

11.5.6 从发送/接收

从发送/接收是在传输时钟由另一个设备输入的状态下，78K0R/KG3-L 发送/接收数据到/从另一个设备。

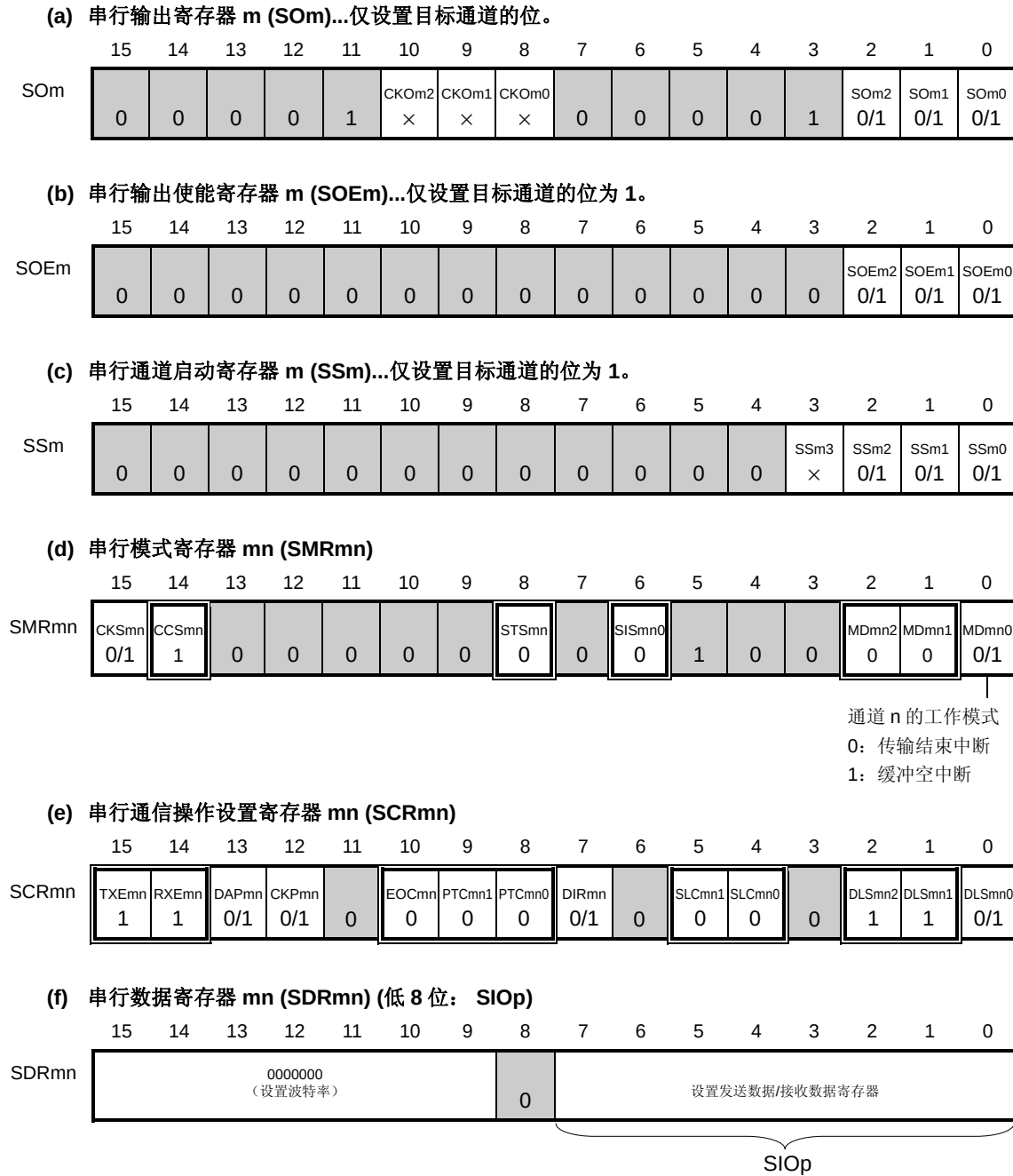
3线串行I/O	CSI00	CSI01	CSI10	CSI20
目标通道	SAU0的通道0	SAU0的通道1	SAU0的通道2	SAU1的通道0
使用引脚	$\overline{\text{SCK00}}$, SI00, SO00	$\overline{\text{SCK01}}$, SI01, SO01	$\overline{\text{SCK10}}$, SI10, SO10	$\overline{\text{SCK20}}$, SI20, SO20
中断	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	可以选择传输结束中断（在单发送模式下）或者缓冲空中断（在连续传输模式下）			
错误检测标志	仅溢出错误检测标志 (OVFmn)			
发送数据长度	7 或 8 位			
传输速率	最大值: $f_{\text{MCK}}/6$ [MHz] ^{注1, 2}			
数据相位	通过DAPmn位可选择 • DAPmn = 0: 从串行时钟开始操作时, 数据 I/O 开始。 • DAPmn = 1: 在串行时钟操作开始之前半个时钟, 数据 I/O 开始。			
时钟相位	通过 CKPmn位选择 • CKPmn = 0: 正向 • CKPmn = 1: 反向			
数据方向	首先是MSB或者LSB			

- 注**
1. 因为外部串行时钟输入到引脚 $\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$ 和 $\overline{\text{SCK20}}$ 是内部采样并使用, 最大传输速率为 $f_{\text{MCK}}/6$ [MHz]。
 2. 请在满足上面条件和电气特性中的 AC 特性的范围内, 使用这个操作 (参见 第 27 章 电气特性 (目标值))。

- 备注**
1. f_{MCK} : 目标通道的工作时钟(MCK)频率
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10

(1) 寄存器设置

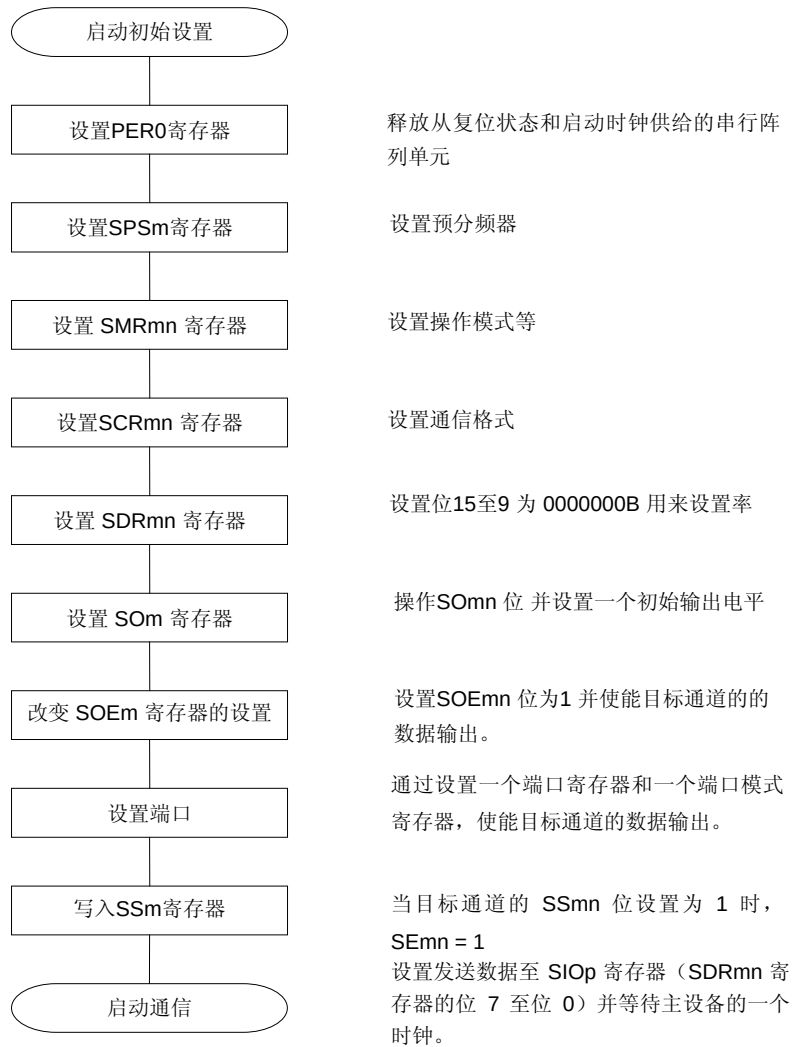
图 11-60. 用于 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 的从发送/接收的寄存器内容的示例



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
p: CSI 编号 (p = 00, 01, 10, 20)
□.: 在 CSI 从发送/接收模式下固定设置, ■.: 禁止设置 (设为初始值)
x: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)
0/1: 根据用户的使用设置为 0 或 1

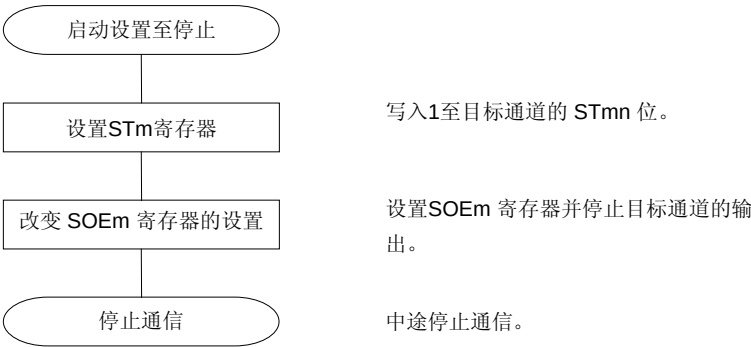
(2) 操作程序

图 11-61. 从发送/接收的初始化程序



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 11-62. 从发送/接收的停止程序



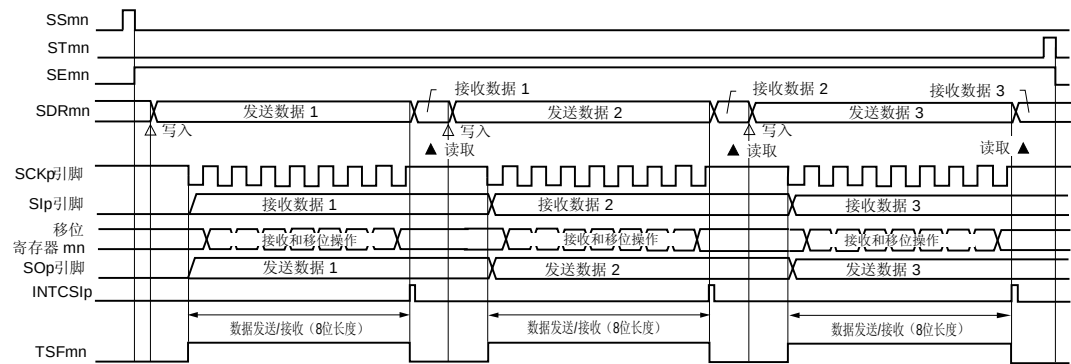
备注 即使通信停止后，引脚电平依然保持。要恢复操作，重新设置 SOm 寄存器 (参见 图 11-63 恢复从发送/接收的程序)。

图 11-63. 恢复从发送/接收的程序



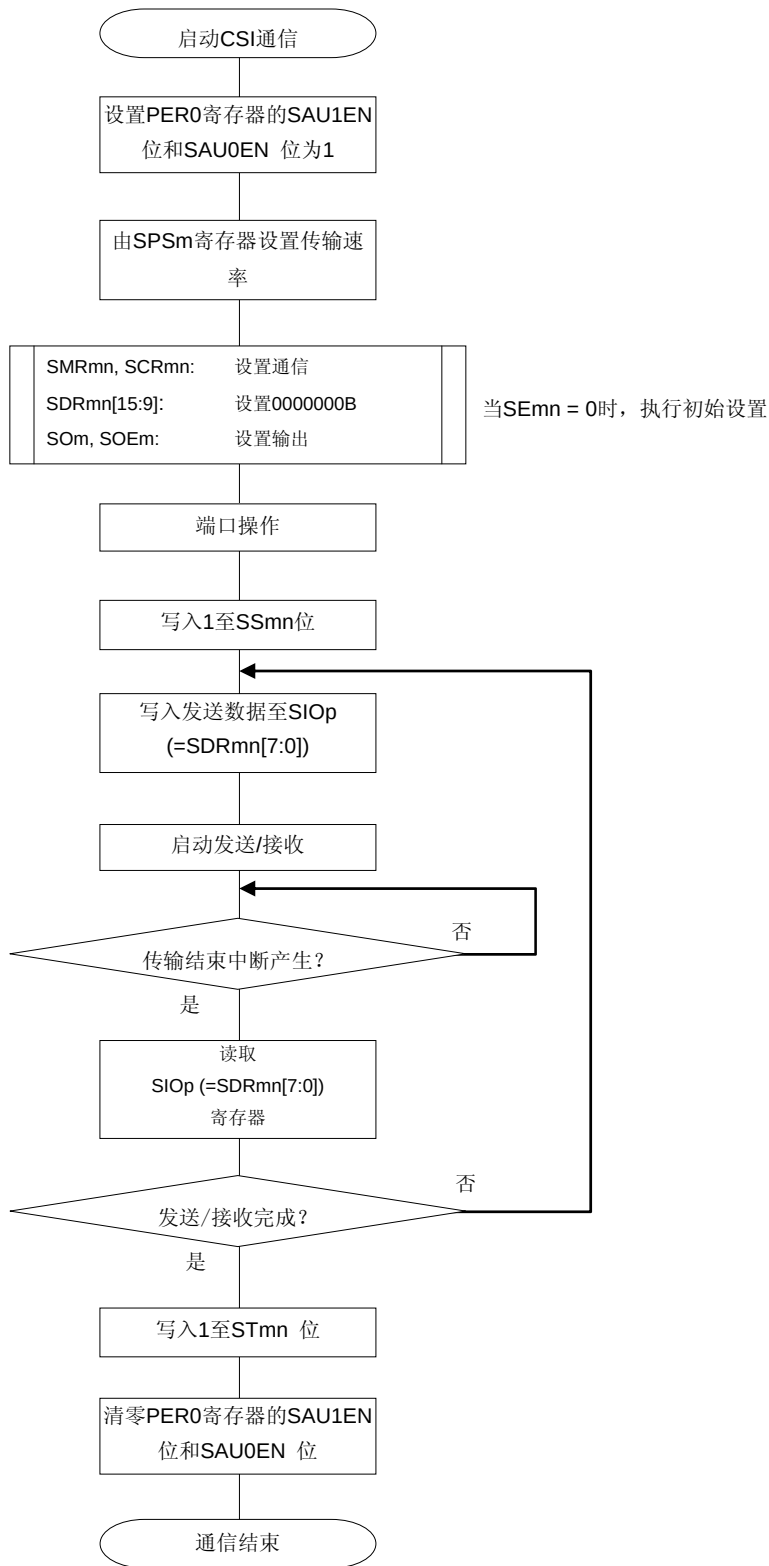
(3) 处理流程（在单发送/接收模式下）

图 11-64. 从发送/接收的时序图（在单发送/接收模式下）
(类型 1: DAPmn = 0, CKPmn = 0)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
p: CSI 编号 (p = 00, 01, 10, 20)

图 11-65. 从发送/接收的流程图（在单发送/接收模式下）

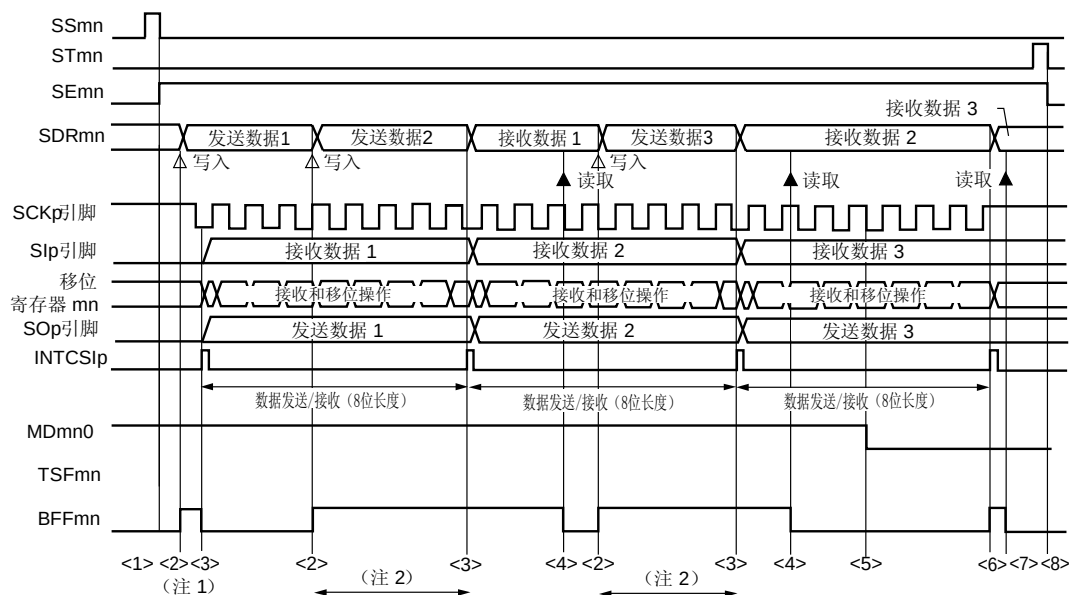


注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

(4) 处理流程（在连续发送/发送模式下）

图 11-66. 从发送/发送的时序图（在连续发送/发送模式下）

(类型 1: DAPmn = 0, CKPmn = 0)



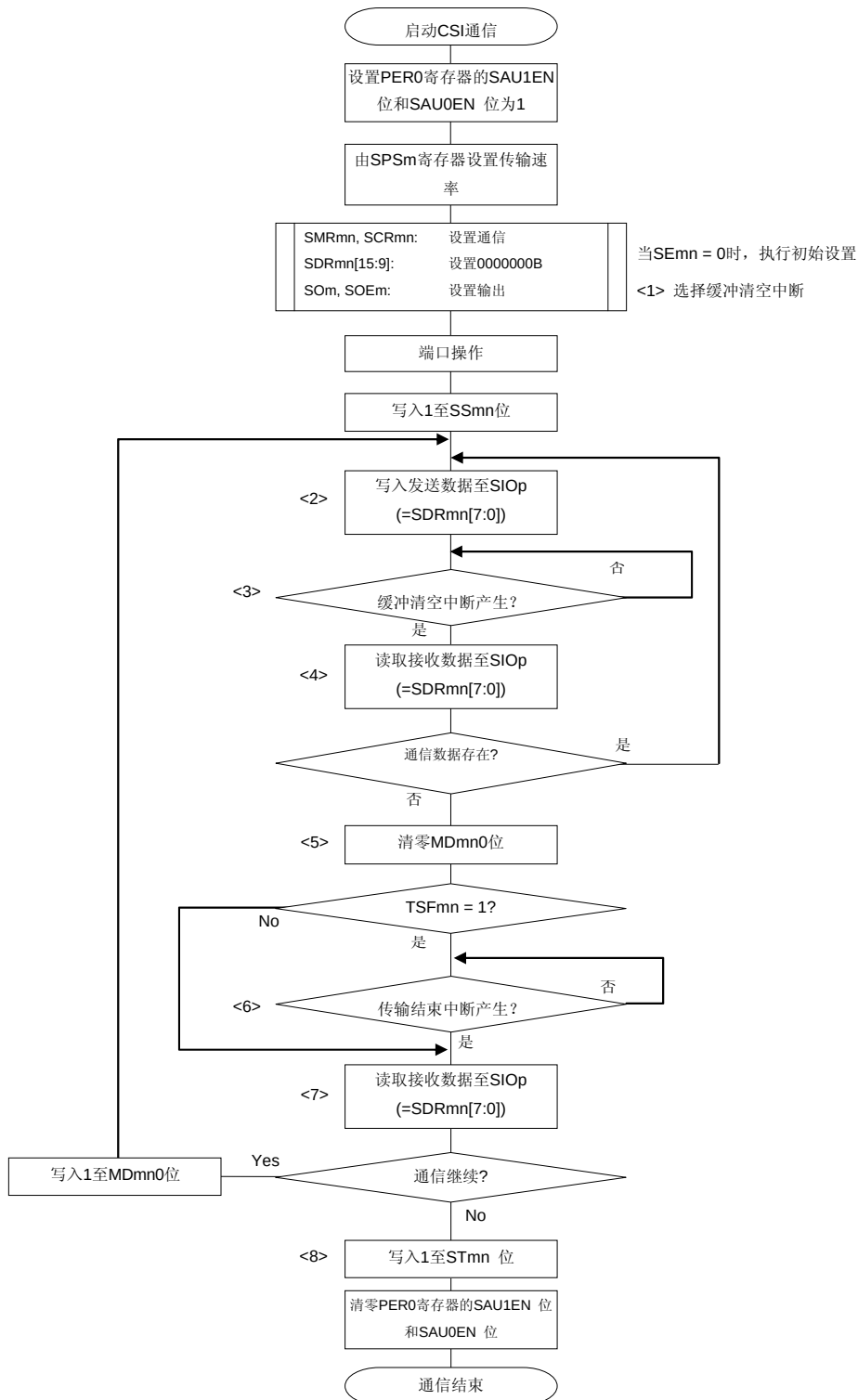
- 注 1. 在 BFFmn = 1 的情况下，当发送数据写入 SDRmn 寄存器时，发送数据可以重写。
 2. 在这期间，发送数据可以通过读取 SDRmn 寄存器来读取。此时传输操作不受影响。

注意事项 即使在操作过程中，MDmn0 位也可以重写。

但是，要在最后一位开始传输之前重写它，这样它可以在最后一个发送数据的传输结束中断之前重写。

- 备注 1. 图中的<1>至<8>对应图 11-67 从发送/接收的流程图（在连续发送/接收模式下）中的<1>至<8>。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10
 p: CSI 编号 (p = 00, 01, 10, 20)

图 11-67. 从发送/接收的流程图（在连续发送/接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

备注 图中的<1>至<8>对应图 11-66 从发送/接收的时序图（在连续发送/接收模式下）中的<1>至<8>。

11.5.7 计算传输时钟频率

用于 3 线串行 I/O (CSI00, CSI01, CSI10, CSI20) 通信的传输时钟频率，可以通过下面公式计算。

(1) 主

$$(\text{传输时钟频率}) = \{\text{目标通道的工作时钟(MCK)频率}\} \div (\text{SDRmn}[15:9] + 1) \div 2 \text{ [Hz]}$$

(2) 从

$$(\text{传输时钟频率}) = \{\text{从主提供的串行时钟 (SCK)频率}\}^{\frac{1}{2}} \text{ [Hz]}$$

注 使能的最大传输时钟频率是 $f_{\text{MCK}} / 6$ 。

- 备注**
1. SDRmn[15:9]的值是 SDRmn 寄存器的位 15 至位 9 的值 (0000000B 至 1111111B)，因此它的取值范围为 0 至 127。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10

工作时钟 (MCK)由串行时钟选择寄存器 m (SPSm)和串行模式寄存器 mn (SMRmn)的位 15 (CKSmn)来确定。

表 11-2. 工作时钟的选择

SMRmn 寄存器	SPSm 寄存器								工作时钟(MCK) ^{‡1}	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f _{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f _{CLK}	20 MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	10 MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	5 MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	2.5 MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	1.25 MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	625 kHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	313 kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	156 kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	78.1 kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	39.1 kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	19.5 kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{‡2}	
1	0	0	0	0	X	X	X	X	f _{CLK}	20 MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	10 MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	5 MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	2.5 MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	1.25 MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	625 kHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	313 kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	156 kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	78.1 kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	39.1 kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	19.5 kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{‡2}	
其它值									设置禁止	

- 注**
1. 当改变用于 f_{CLK} 的时钟（通过改变系统时钟控制寄存器 (CKC) 的值）时，在停止串行阵列单元 (SAU) 的操作 (STm = 000FH) 之后，才这样做。当选择 INTTM02 和 INTTM03 为工作时钟时，同样停止定时器阵列单元(TAU0) (TT0 = 00FFH)。
 2. 无论 f_{CLK} 的频率为多少（主系统时钟，副系统时钟），通过操作间隔定时器选择 f_{SUB}/4 作为计数时钟（设定 TIS0 寄存器的 TIS02 (如果 m = 0)或 TIS03 (如果 m = 1)为 1），并且通过 SPSm 寄存器中 TAU0 选择 INTTM02 和 INTTM03，从而 SAU 可工作在固定的副系统时钟分频比。但是，当改变 f_{CLK}，必须按照上面注 1 描述的停止 SAU 和 TAU0。

备注

1. X: 不必考虑
2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2), mn = 00 至 02, 10

11.6 UART(UART0 至 UART3)通信的操作

这是一个起始停止同步功能，使用 2 条线：串行数据发送(TxD)和串行数据接收(RxD)线。它和通信部分(通过使用内部波特率)同步发送或者接收数据。全双工 UART 通信可以看作是使用两个通道，一个专门发送(偶通道)，一个专门接收(奇通道)。

[数据发送/接收]

- 5, 7, 或者 8 位的数据长度
- 选择 MSB/LSB
- 发送/接收数据的电平设置和反向选择
- 附加校验位和校验检测功能
- 附加停止位

[中断功能]

- 传输结束中断/缓冲空中断
- 帧错误、校验错误、或者溢出错误时产生错误中断

[错误检测标志]

- 帧错误、校验错误、或者溢出错误

UART3(单元 1 的 2 和 3 通道)支持 LIN 总线

[LIN 总线功能]

- 唤醒信号检测
- 同步中断域 (SBF)检测
- 同步域测量、波特率测量

} 使用外部中断(INTP0)或定时器阵列单元(TAU0)

UART0 使用 SAU0 的通道 0 和 1。

UART1 使用 SAU0 的通道 2 和 3。

UART2 使用 SAU1 的通道 0 和 1。

UART3 使用 SAU1 的通道 2 和 3。

单位	通道	用作CSI	用作UART	用作简易I ² C
0	0	CSI00	UART0	—
	1	CSI01		—
	2	CSI10	UART1	IIC10
	3	—		—
1	0	CSI20	UART2	IIC20
	1	—		—
	2	—	UART3(支持LIN总线)	—
	3	—		—

注意事项 当使用串行阵列单元 0 和 1 作为 UART 时，传输端(偶数通道)和接收端(奇数通道)的通道只能用作 UART。

UART 执行下面 4 种通信操作。

- UART 发送 (参见 11.6.1.)
- UART 接收 (参见 11.6.2.)
- LIN 发送(仅 UART3) (参见 11.6.3.)
- LIN 接收(仅 UART3) (参见 11.6.4.)

11.6.1 UART 发送

UART 发送是从 78K0R/KG3-L 发送数据到另外一个设备的同步操作(同步起始停止)。

UART 使用的两个通道，偶通道用于 UART 发送。

UART	UART0	UART1	UART2	UART3
目标通道	SAU0的通道0	SAU0的通道2	SAU1的通道0	SAU1的通道2
使用引脚	TxD0	TxD1	TxD2	TxD3
中断	INTST0	INTST1	INTST2	INTST3
	可以选择传输结束中断(在单传输模式下)或者缓冲空中断(在连续传输模式下)			
错误检测标志	无			
传输数据长度	5, 7, 或 8 位			
传输速率	最大值 $f_{MCK}/6$ [bps] ($SDRmn[15:9] = 2$ 或更大), 最小值 $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] [※]			
数据相位	正向输出(默认: 高电平) 反向输出(默认: 低电平)			
校验位	以下可选 • 无校验位 • 附加0校验 • 附加偶校验 • 附加奇校验			
停止位	以下可选 • 附加1位 • 附加2位			
数据方向	首先是MSB 或者LSB			

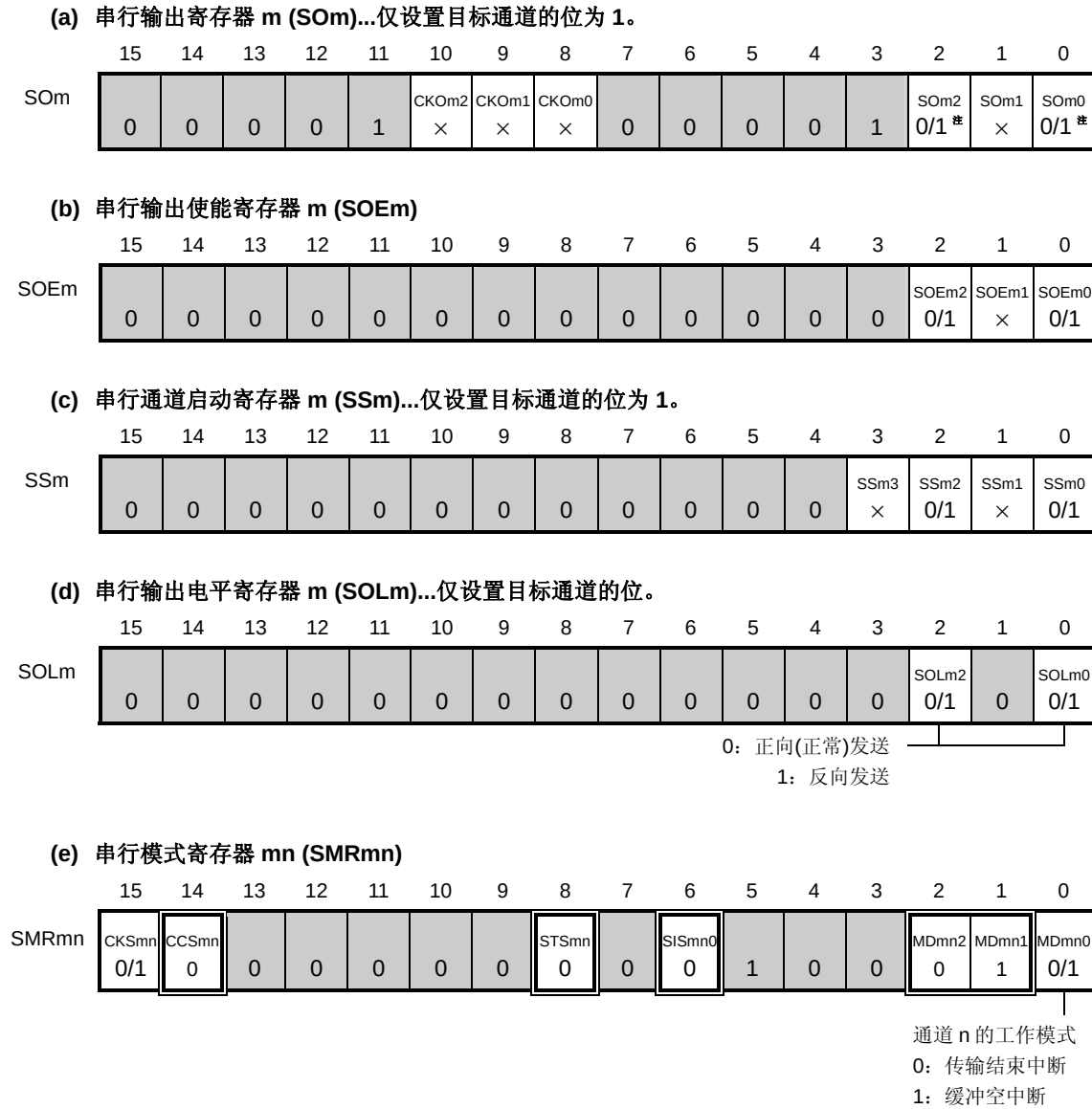
注 请在满足上面条件和电气特性中的 AC 特性的范围内, 使用这个操作(参见 第 27 章 电气特性(目标值))。

备注

1. f_{MCK} : 目标通道的工作时钟(MCK)频率
 f_{CLK} : 系统时钟频率
2. m : 单元编号 ($m = 0, 1$), n : 通道编号 ($n = 0, 2$), $mn = 00, 02, 10, 12$

(1) 寄存器设置

图 11-68. 用于 UART (UART0 至 UART3) 的 UART 发送的寄存器内容的示例 (1/2)

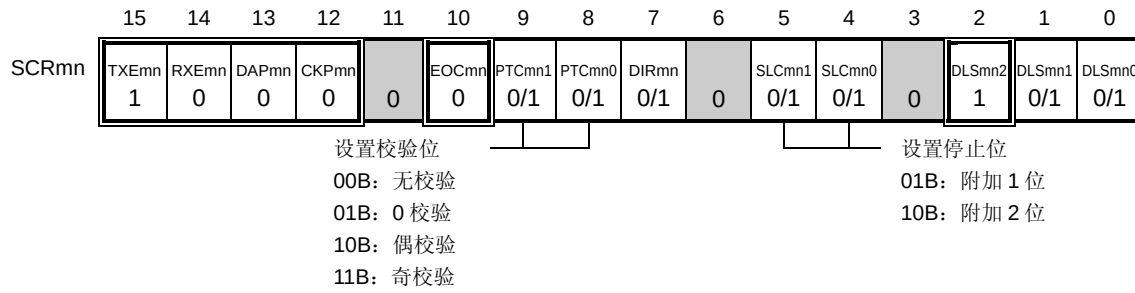


注 在启动发送前，确保当目标通道的 SOLmn 位设置为 0 时，该位设置为 1，当目标通道的 SOLmn 位设置为 1 时，该位设置为 0。这个值根据通信期间数据的变化而变化。

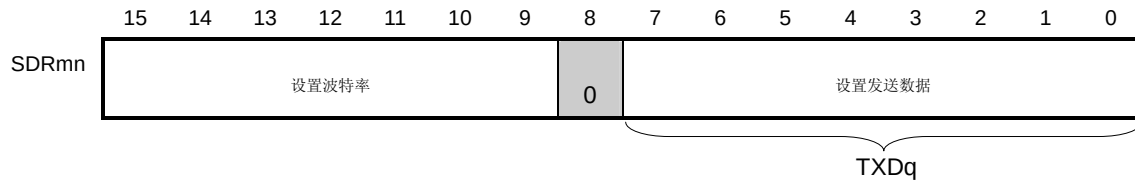
备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 00, 02, 10, 12
q: UART 编号 (q = 0 至 3)
□.: 在 UART 发送模式下固定设置, ■.: 禁止设置(由硬件给出固定值)
×: 此模式下不能使用的位(不在任何模式下使用时，设为初始值)
0/1: 根据用户的使用设置为 0 或 1

图 11-68. 用于 UART (UART0 至 UART3) 的 UART 发送的寄存器内容的示例 (2/2)

(f) 串行通信操作设置寄存器 mn (SCRmn)



(g) 串行数据寄存器 mn (SDRmn) (低 8 位: TXDq)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 00, 02, 10, 12

q: UART 编号 (q = 0 至 3)

□: 在 UART 发送模式下固定设置, ■: 禁止设置(设为初始值)

0/1: 根据用户的使用设置为 0 或 1

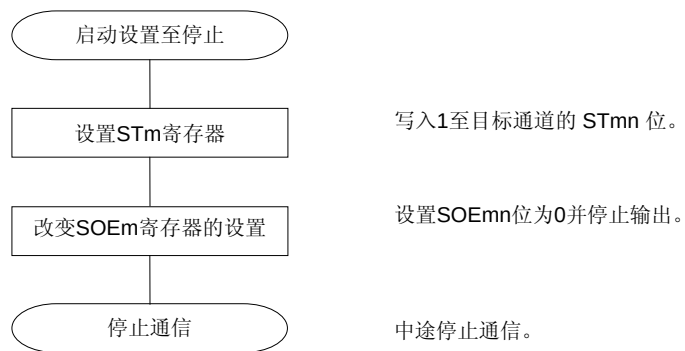
(2) 操作程序

图 11-69. UART 发送的初始化程序



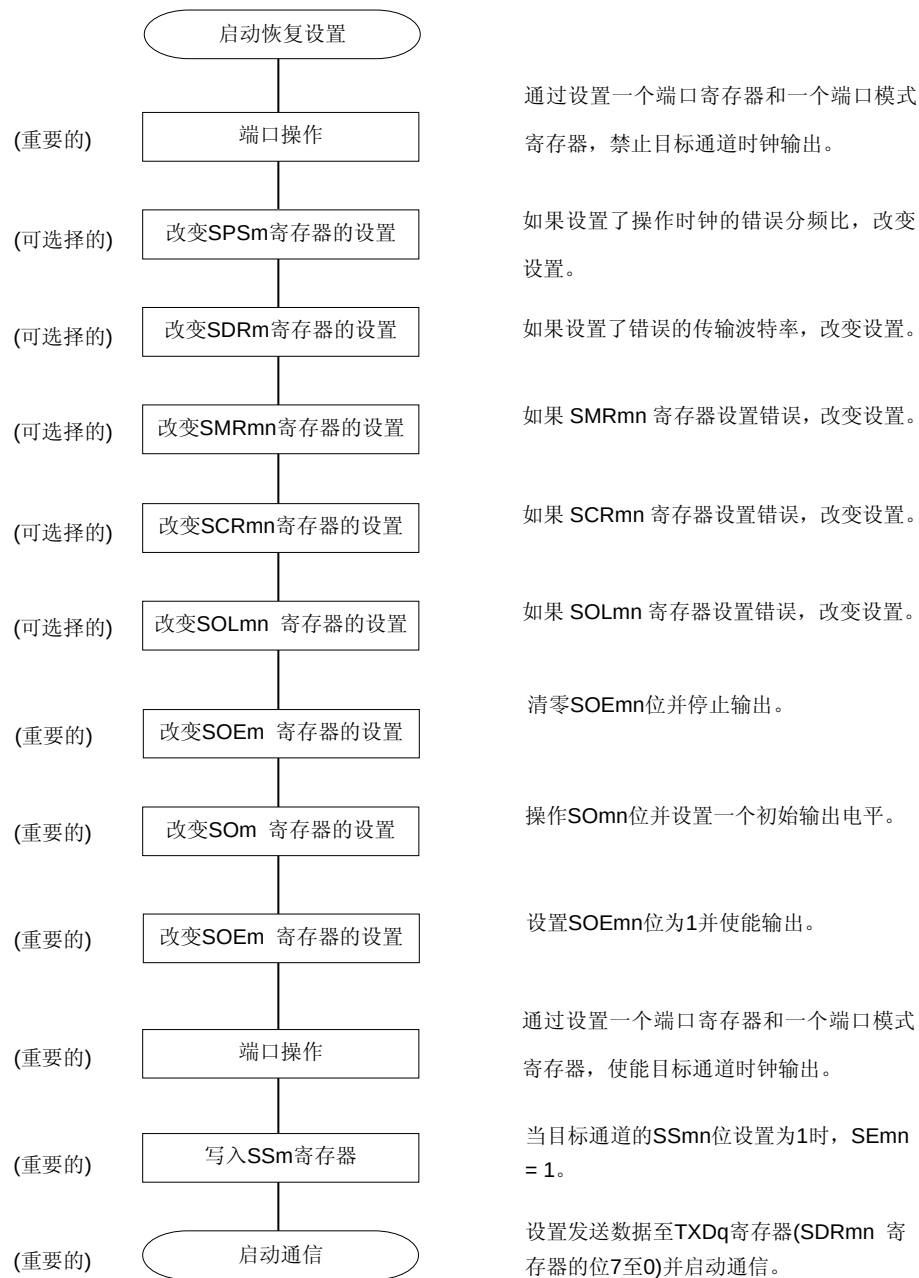
注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 11-70. UART 发送的停止程序



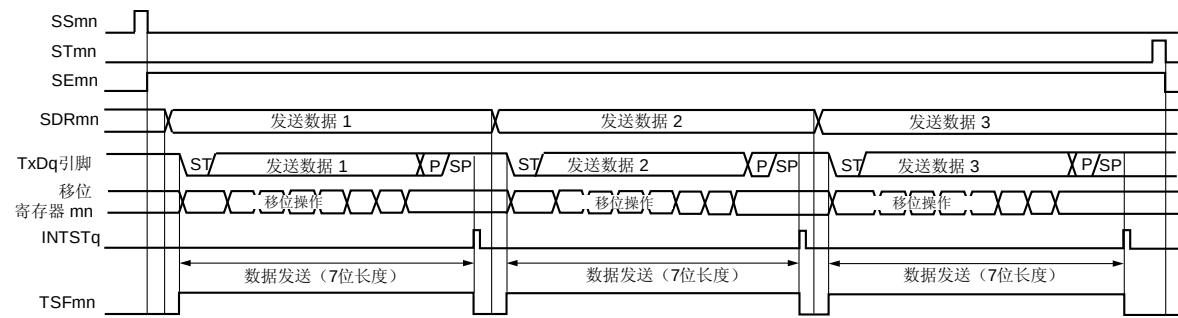
备注 即使通信停止后，引脚电平依然保持。要恢复操作，重新设置 SOm 寄存器(参见 图 11-71 恢复 UART 发送的程序)。

图 11-71. 重新 UART 发送的程序



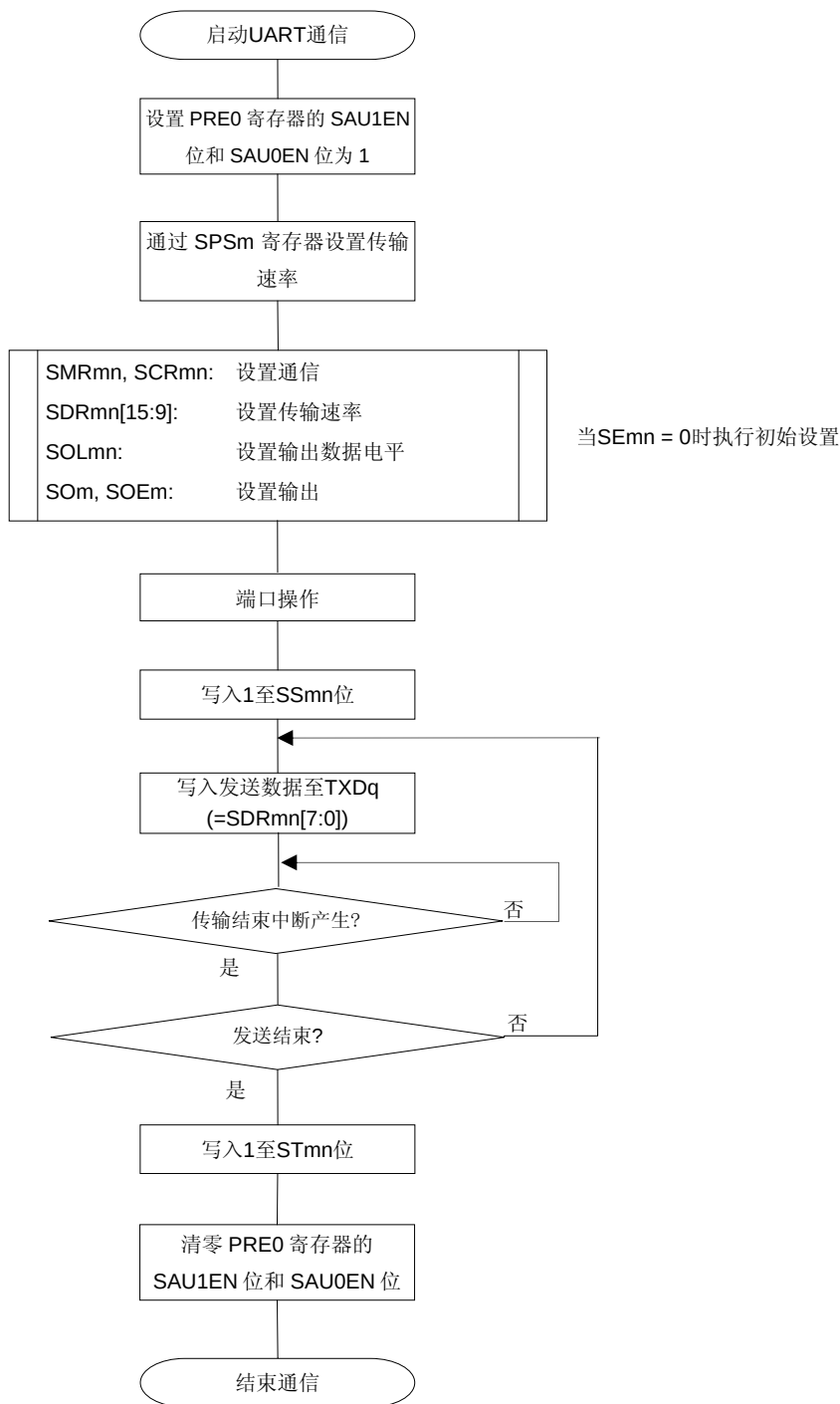
(3) 处理流程(在单发送模式下)

图 11-72. UART 发送的时序图(在单发送模式下)



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 00, 02, 10, 12
q: UART 编号 (q = 0 至 3)

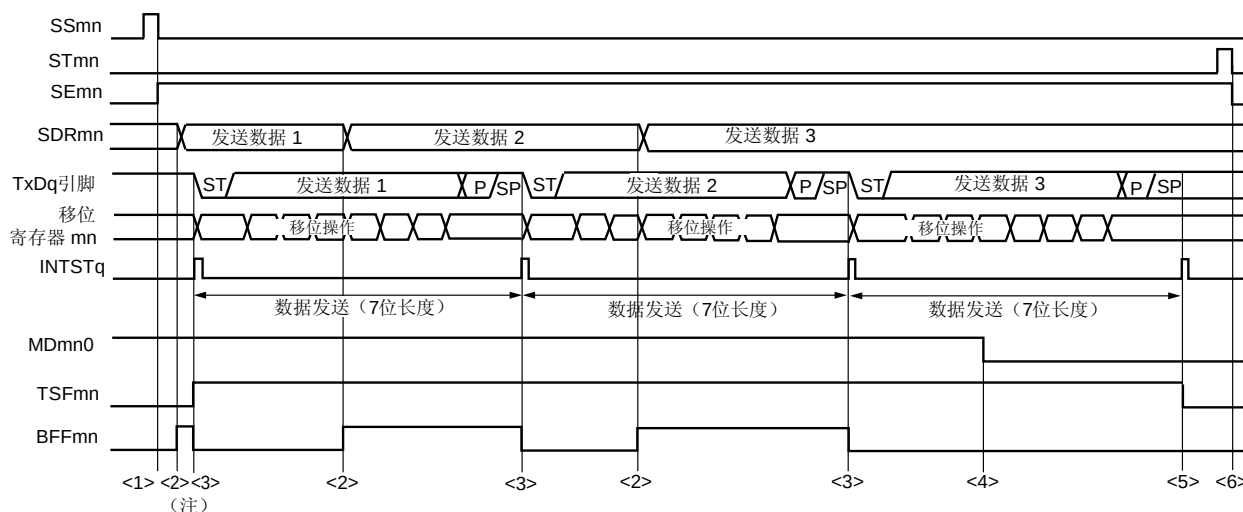
图 11-73. UART 发送的流程图(在单发送模式下)



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

(4) 处理流程(在连续发送模式下)

图 11-74. UART 发送的时序图(在连续发送模式下)



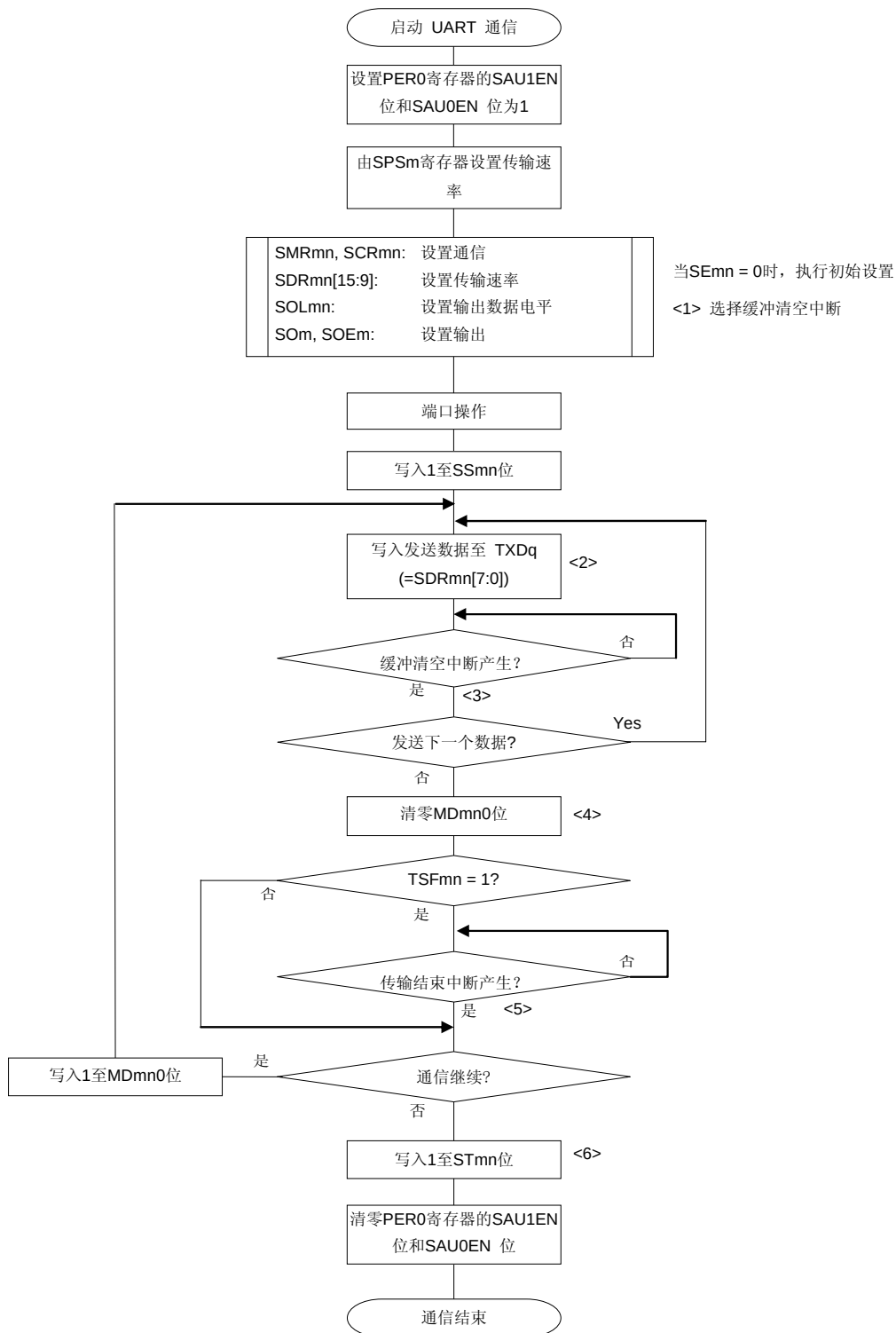
注 在 BFFmn = 1 的情况下，当发送数据写入 SDRmn 寄存器时，发送数据可以重写。

注意事项 即使在操作过程中，MDmn0 位也可以重写。

但是，要在最后一位开始传输之前重写它，这样它可以在最后一个发送数据的传输结束中断之前重写。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 00, 02, 10, 12
q: UART 编号 (q = 0 至 3)

图 11-75. UART 发送的流程图(在连续发送模式下)



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

备注 图中的<1>至<6> 对应图 11-74 UART 发送的时序图(在连续发送模式下)中的<1>至<6>。

11.6.2 UART 接收

UART 接收是一个 78K0R/KG3-L 从其它设备异步接收数据的操作(启动-停止同步)。

对于 UART 接收, 使用用于 UART 两个通道中的奇数通道。必须设置 SMR 寄存器的偶数通道。

UART	UART0	UART1	UART2	UART3
目标通道	SAU0的通道1	SAU0的通道3	SAU1的通道1	SAU1的通道3
使用引脚	RxD0	RxD1	RxD2	RxD3
中断	INTSR0	INTSR1	INTSR2	INTSR3
	仅传输结束中断(禁止设置缓冲空中断。)			
错误中断	INTSRE0	INTSRE1	INTSRE2	INTSRE3
错误检测标志	- 帧错误检测标志 (FEFmn) - 奇偶校验错误检测标志 (PEFmn) - 溢出错误检测标志 (OVFmn)			
传输数据长度	5、7或8位			
传输速率	最大值 $f_{MCK}/6$ [bps] ($SDRmn[15:9] = 2$ 或更大), 最小值 $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] [※]			
数据相位	正向输出(默认: 高电平) 反向输出(默认: 低电平)			
校验位	以下可选 - 无校验位(无校验检查) - 附加0校验(无校验检查) - 附加偶校验 - 附加奇校验			
停止位	附加 1 位			
数据方向	首先是MSB或者LSB			

注 请在满足上面条件和电气特性中的 AC 特性的范围内, 使用这个操作(参见 第 27 章 电气特性(目标值))。

备注

1. f_{MCK} : 目标通道的工作时钟(MCK)频率
 f_{CLK} : 系统时钟频率
2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 1, 3), mn = 01, 03, 11, 13

(1) 寄存器设置

图 11-76. 用于 UART (UART0 至 UART3) 的 UART 接收的寄存器内容的示例 (1/2)

(a) 串行输出寄存器 m (SOM)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	1	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	1	SOM2 ×	SOM1 ×	SOM0 ×

(b) 串行输出使能寄存器 m (SOEm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 ×	SOEm0 ×

(c) 串行通道启动寄存器 m (SSm)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 ×	SSm1 ×	SSm0 ×

(d) 穿行模式寄存器 mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 1	0	SlSmn0 0/1	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0

0: 正向(正常)接收
1: 反向接收

通道 n 的工作模式
0: 传输结束中断

(e) 穿行模式寄存器 mr (SMRmr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmr	CKSmr 0/1	CCSmr 0	0	0	0	0	0	STSmr 0	0	SlSmr0 0	1	0	0	MDmr2 0	MDmr1 1	MDmr0 0/1

与 CKSmn 的设定值相同

通道 r 的工作模式:
0: 传输结束中断
1: 缓冲空中断

(f) 串行通信操作设置寄存器 mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	EOCmn 1	PTCmn1 0/1	PTCmn0 0/1	DIRmn 0/1	0	SLCmn1 0	SLCmn0 1	0	DLSmn2 1	DLSmn1 0/1	DLSmn0 0/1

注意事项 对于 UART 接收, 确保设置通道 r 的 SMRmr, 通道 r 是与通道 n 成对使用的偶数通道。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 1, 3), mn = 01, 03, 11, 13

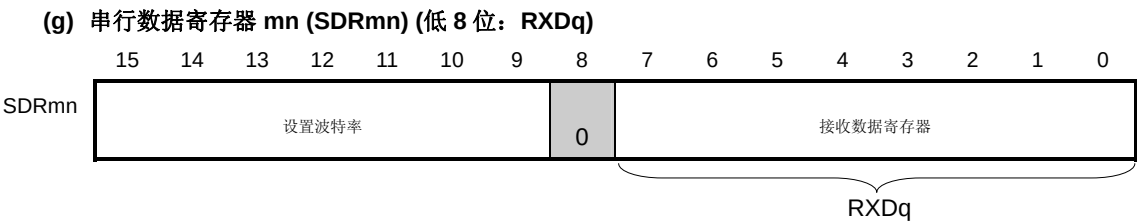
r: 通道编号 (r = n - 1), q: UART 编号 (q = 0 至 3)

□: 在 UART 接收模式下固定设置, .: 禁止设置(设为初始值)

×: 此模式下不能使用的位(不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1。

图 11-76. 用于 UART (UART0 至 UART3) 的 UART 接收的寄存器内容的示例 (2/2)

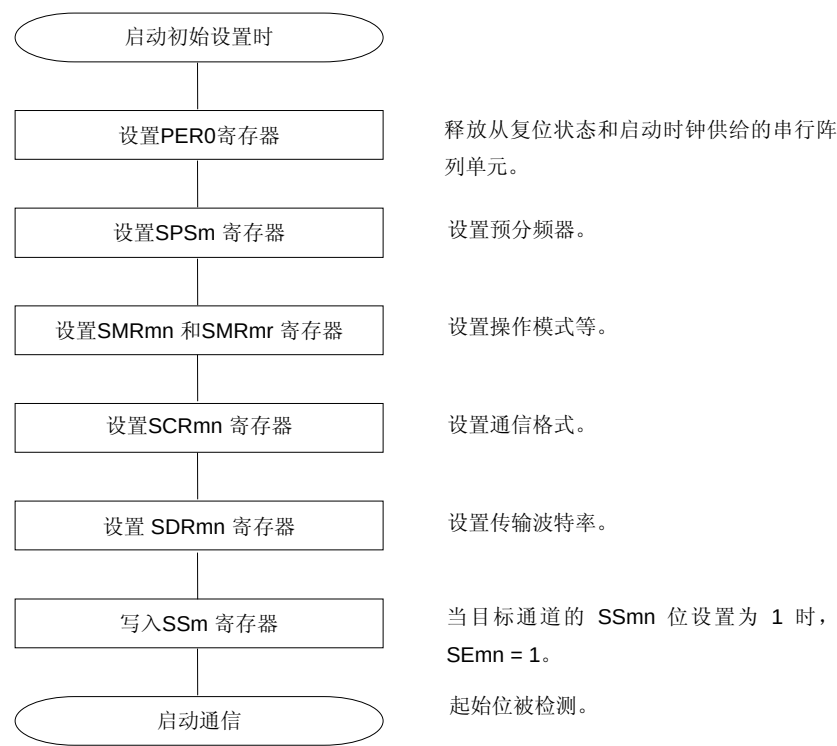


注意事项 对于 UART 接收，确保设置通道 r 的 SMRmr，通道 r 是与通道 n 成对使用的偶数通道。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 1, 3), mn = 01, 03, 11, 13
r: 通道编号 (r = n - 1), q: UART 编号 (q = 0 至 3)
■: 禁止设置(设为初始值)

(2) 操作程序

图 11-77. UART 接收的初始化程序



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

图 11-78. UART 接收的停止程序

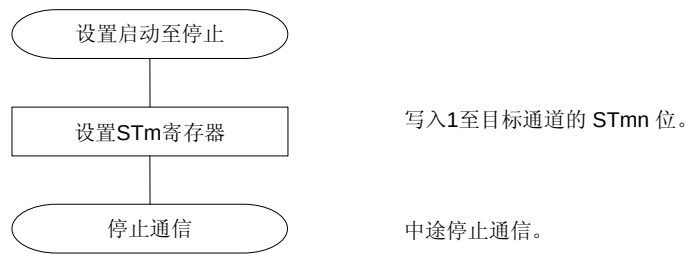
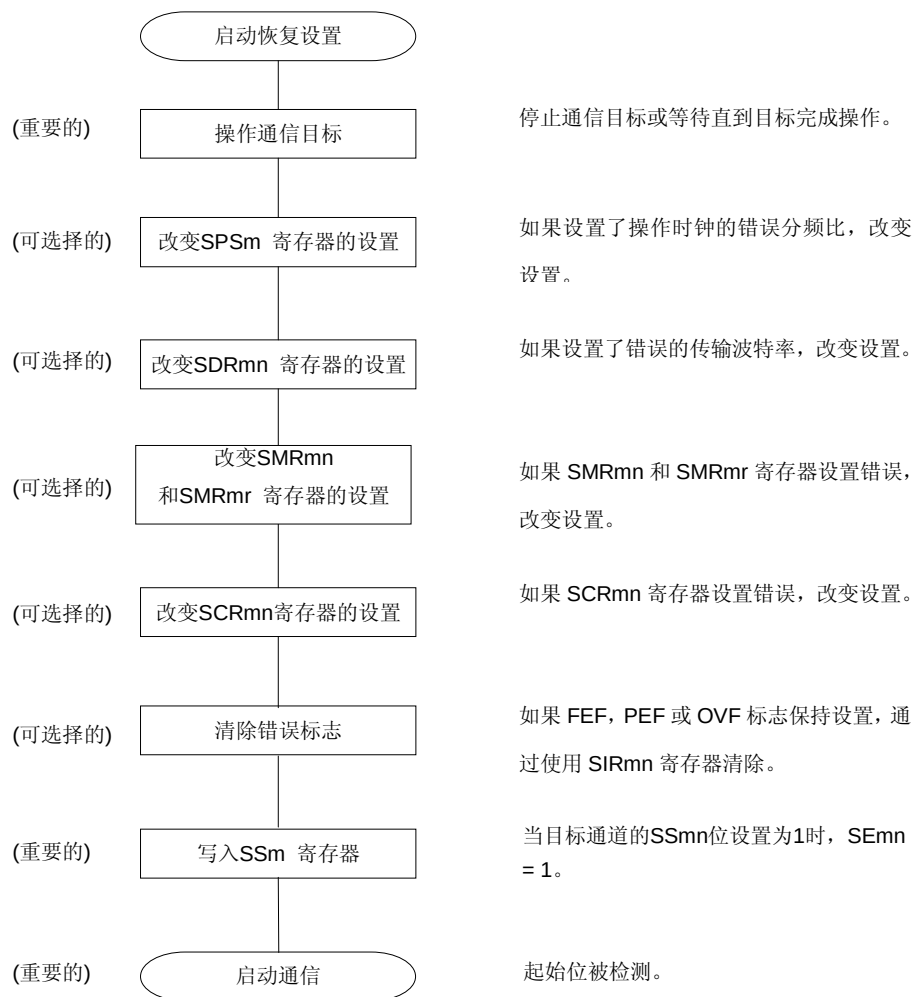
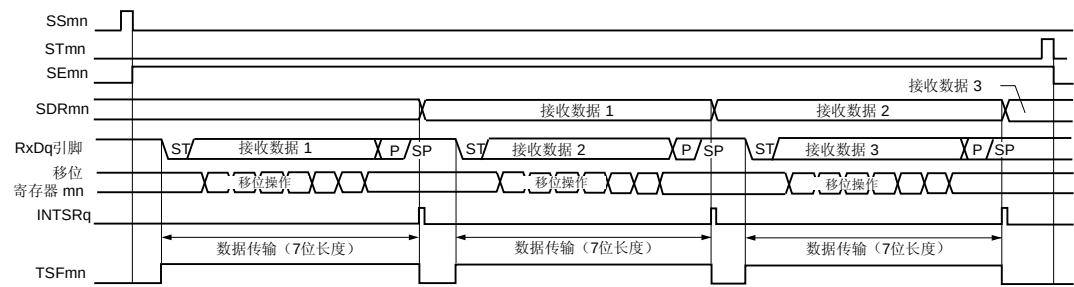


图 11-79. 重新 UART 接收的程序



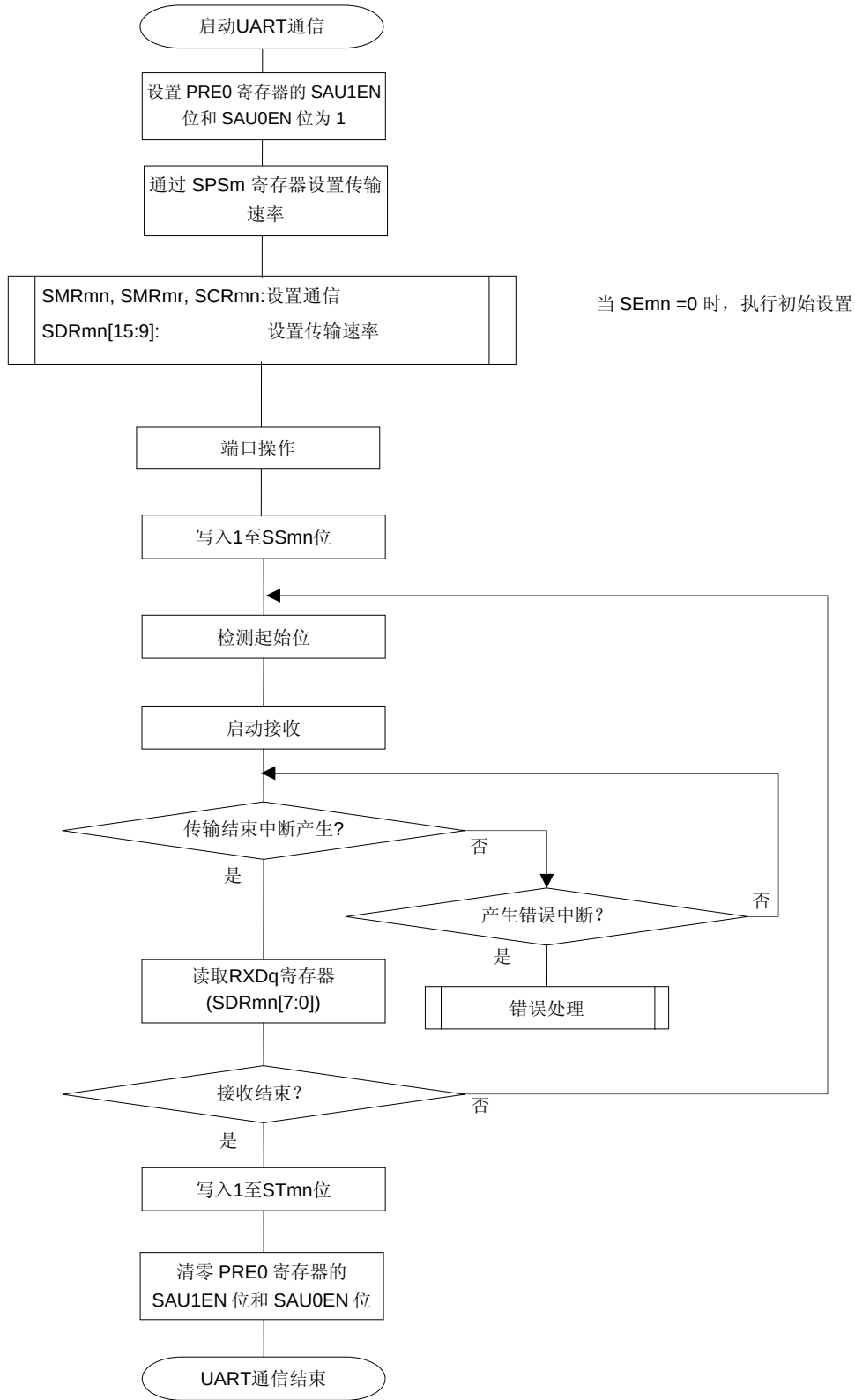
(3) 处理流程

图 11-80. UART 接收的时序图



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 1, 3), mn = 01, 03, 11, 13
q: UART 编号 (q = 0 至 3)

图 11-81. UART 接收的流程图



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

11.6.3 LIN 发送

在 UART 发送中，UART3 支持 LIN 通信。

关于 LIN 发送，使用单元 1(SAU1)的通道 2。

UART	UART0	UART1	UART2	UART3
支持LIN的通信	不支持	不支持	不支持	支持
目标通道	—	—	—	SAU1的通道2
使用引脚	—	—	—	TxD3
中断	—	—	—	INTST3
	可以选择传输结束中断(在单传输模式下)或者缓冲空中断(在连续传输模式下)			
错误检测标志	无			
传输数据长度	8位			
传输速率	最大值 $f_{MCK}/6$ [bps] (SDR12 [15:9] = 2或更大)，最小值 $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] [‡]			
数据相位	正向输出(默认：高电平) 反向输出(默认：低电平)			
校验位	以下可选 • 无校验位 • 附加0校验 • 附加偶校验 • 附加奇校验			
停止位	以下可选 • 附加1位 • 附加2位			
数据方向	首先是MSB或者LSB			

注 请在满足上面条件和电气特性中的 AC 特性的范围内，使用这个操作(参见 第 27 章 电气特性(目标值))。

备注 f_{MCK} : 目标通道的工作时钟(MCK)频率
 f_{CLK} : 系统时钟频率

LIN 代表本地互连网络并且是一个计划用于降低汽车网络费用的低速((1 至 20 kbps))串行通信协议。

LIN 的通信是单主通信，可以连接 15 个从设备到一个主设备。

这些从设备用于控制通过 LIN 连接到主设备的切换器，制动器和传感器。

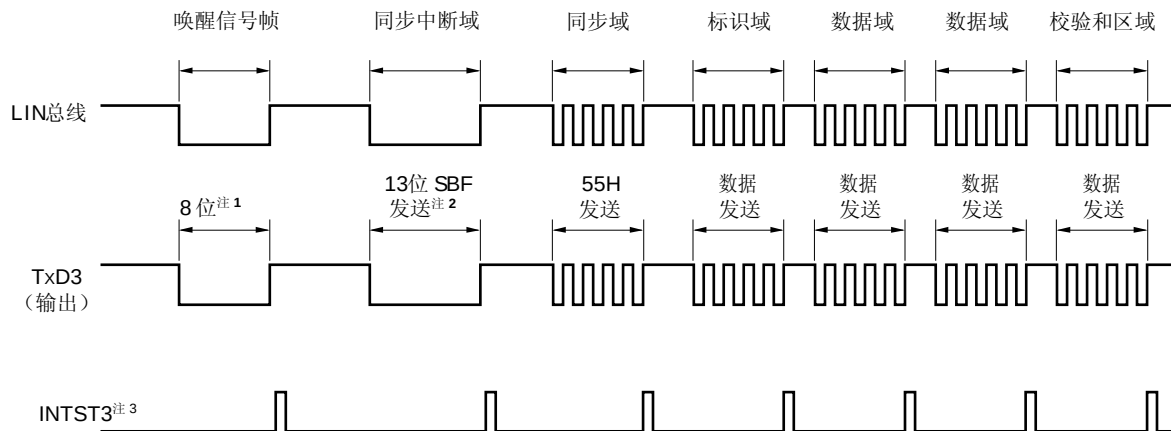
通常，主设备连接到例如 CAN(控制区域网络)的一个网络。

LIN 总线是一个单线总线，它的网络节点通过收发器连接到 ISO9141。

通过 LIN 协议，主设备发送一个含有波特率信息的帧。从设备从主设备接收这个帧，并纠正波特率误差。如果从设备的波特率误差在 $\pm 15\%$ ，则可以建立这个通信。

图 11-82 描述了一个发送 LIN 的操作。

图 11-82. LIN 的操作的发送

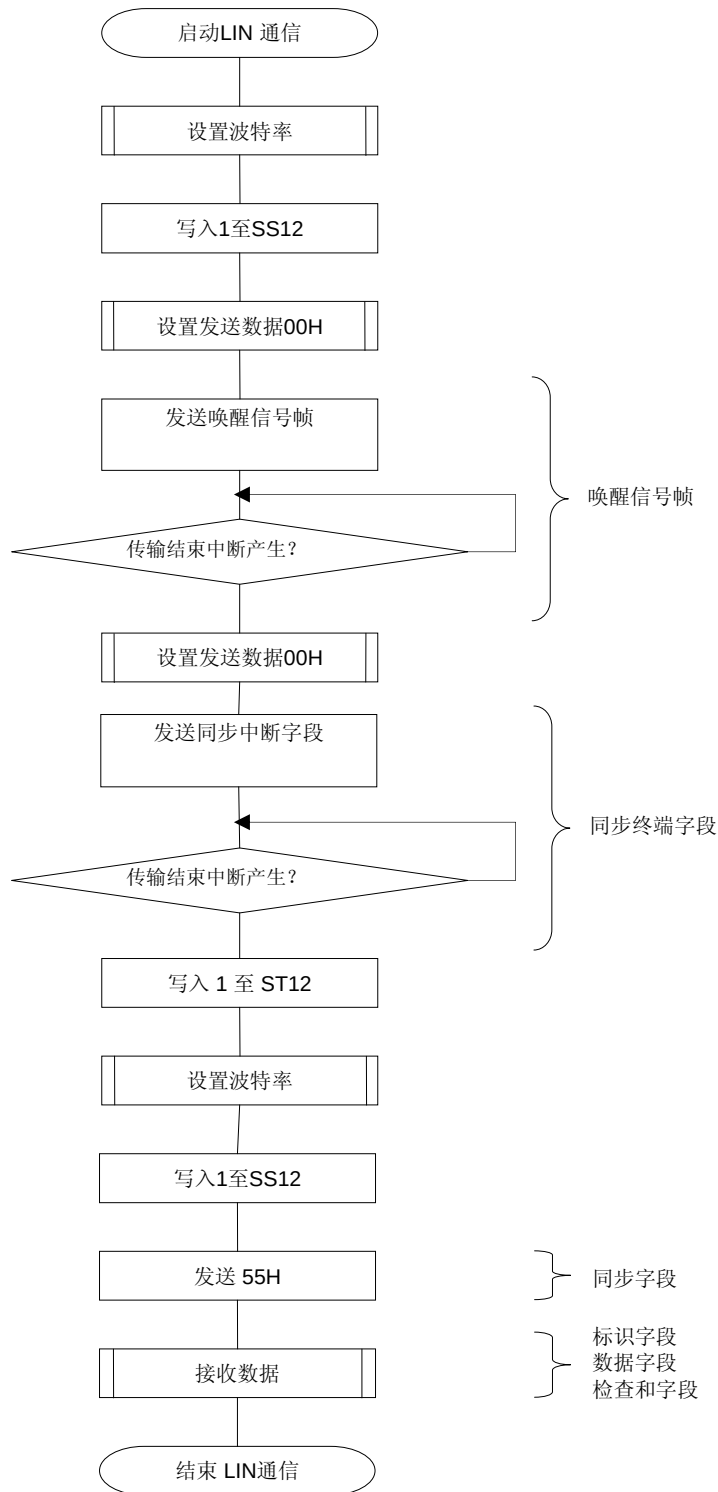


- 注
1. 设置波特率满足唤醒信号的标准并且发送数据 00H。
 2. 同步中断域定义为长度为 13 位并且输出一个低电平。因此，当主传输的波特率为 N [bps]时，同步中断域的波特率计算如下。

$$(\text{同步中断区域的波特率}) = 9/13 \times N$$
 通过在这个波特率下发送数据 00H，产生一个同步中断域
 3. INTST3 是发送完成后的输出。当执行 SBF 发送时，同样也输出 INTST3。

备注 域的间隔是通过软件控制的。

图 11-83. LIN 发送的流程图



11.6.4 LIN 接收

在 UART 接收中，UART3 支持 LIN 通信。

对于 LIN 接收，使用单元 1 的通道 3(SAU1)。

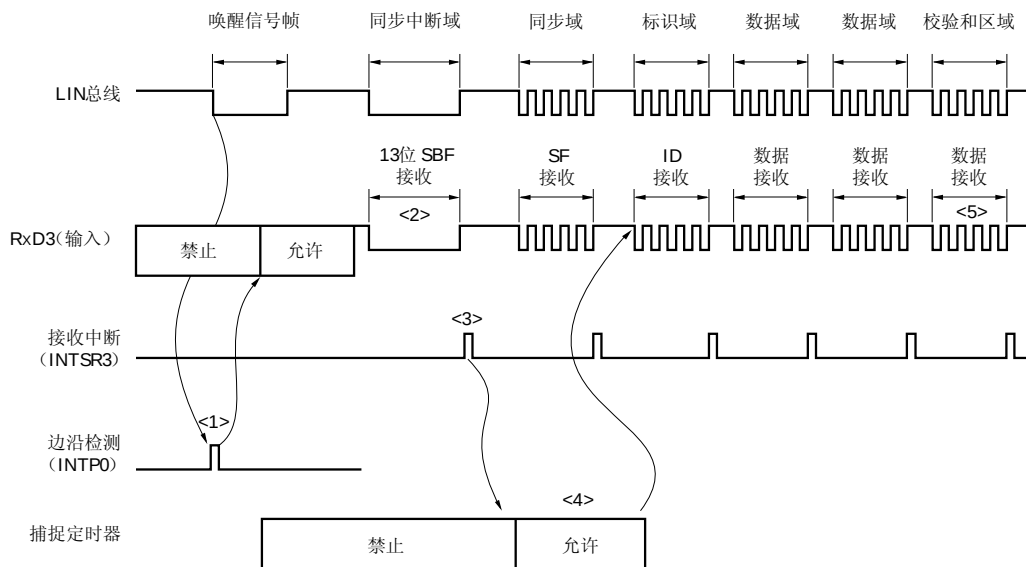
UART	UART0	UART1	UART2	UART3
支持LIN的通信	不支持	不支持	不支持	支持
目标通道	—	—	—	SAU1的通道3
使用引脚	—	—	—	RxD3
中断	—	—	—	INTSR3
	仅传输结束中断(禁止设置缓冲空中断。)			
错误中断	—	—	—	INTSRE3
错误检测标志	• 帧错误检测标志(FEF13) • 奇偶校验错误检测标志 (PEF13) • 溢出错误检测标志 (OVF13)			
传输数据长度	8位			
传输速率	最大值 $f_{MCK}/6$ [bps] (SDR13 [15:9] = 2或更大), 最小值 $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] [※]			
数据相位	正向输出(默认: 高电平) 反向输出(默认: 低电平)			
校验位	以下可选 • 无校验位 • 附加0校验 • 附加偶校验 • 附加奇校验			
停止位	以下可选 • 附加1位 • 附加2位			
数据方向	首先是MSB或者LSB			

注 请在满足上面条件和电气特性中的 AC 特性的范围内，使用这个操作(参见 第 27 章 电气特性(目标值))。

备注 f_{MCK} : 目标通道的工作时钟(MCK)频率
 f_{CLK} : 系统时钟频率

图 11-84 描述了 LIN 的接收操作。

图 11-84. LIN 的接收操作



这是下面信号处理。

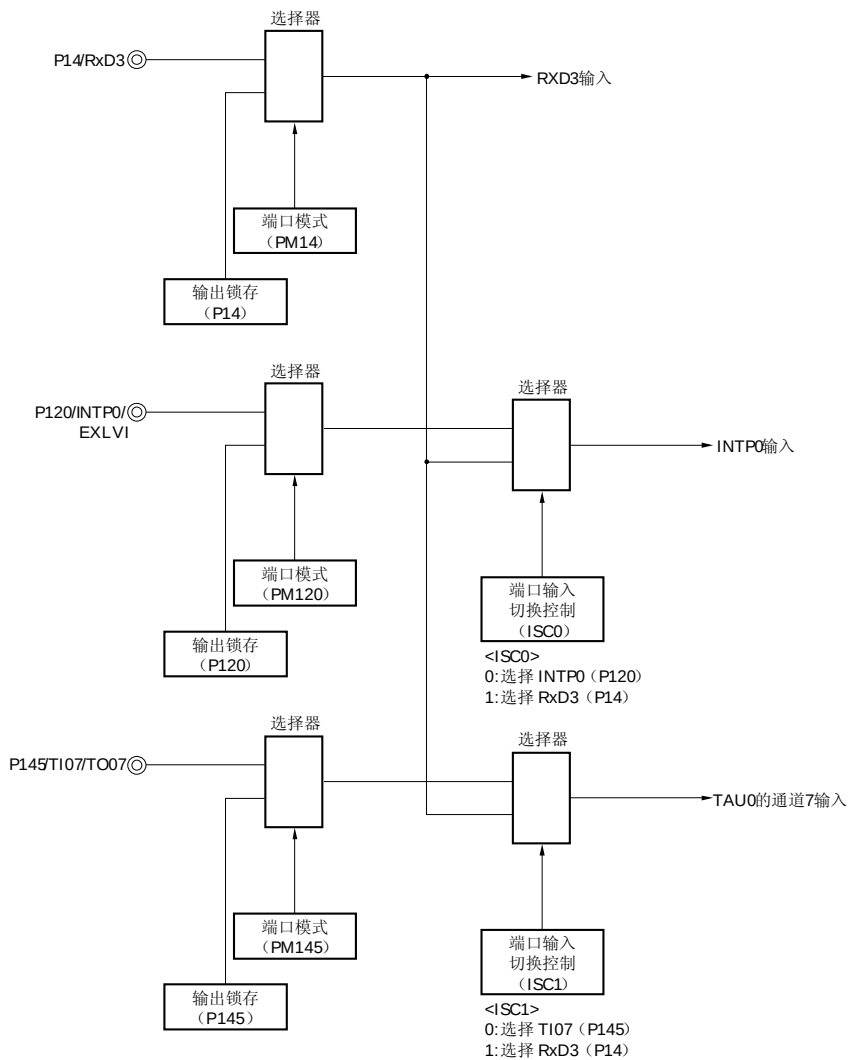
- <1> 通过检测一个引脚的中断沿(INTP0)来检测唤醒信号。当检测到唤醒信号，允许 UART3 的接收(RXE13 = 1)并等待 SBF 接收。
- <2> 当检测到 SBF 的起始位时，接收以设置的波特率开始且产生的串行数据保存在 RxD3 寄存器中(= 串行数据寄存器 13 (SDR13)的位 7 至 0)。当检测到停止位，产生接收结束中断请求 (INTSR3)。当 11 位或者更多的低电平的数据检测为 SBF，则判断 SBF 接收正确结束。如果少于 11 位的低电平的数据检测为 SBF，则产生接收错误，并且系统返回到 SBF 接收等待状态。
- <3> 当 SBF 正确接收完成时，开始定时器阵列单元 TAU0 的通道 7 并测量同步中断区域的位间隔(脉冲宽度)(见 6.7.5 用作输入信号高/低电平宽度测量的操作)
- <4> 从同步域(SF)的位间隔计算波特率误差。停止 UART3 一次并且调节(重新设置)波特率。
- <5> 校验和区域可以通过软件来辨别。另外，在接收到校验和区域之后，执行初始化 UART3，并且等待 SBF 的接收可以通过软件来执行。

图 11-85 展示了一个执行 LIN 接收的端口的配置。

从 LIN 的主设备发送的唤醒信号通过检测外部中断(INTP0)的有效沿来接收。从主通道发送的同步域的长度，可以利用定时器阵列单元(TAU0)的外部事件捕捉操作来测量，来计算波特率误差。

通过控制端口输入(ISC0/ISC1)的切换，用于接收的端口输入(RxD3)的输入源可以输入到外部中断引脚(INTP0)和定时器阵列单元(TAU0)。

图 11-85. LIN 操作接收的端口配置



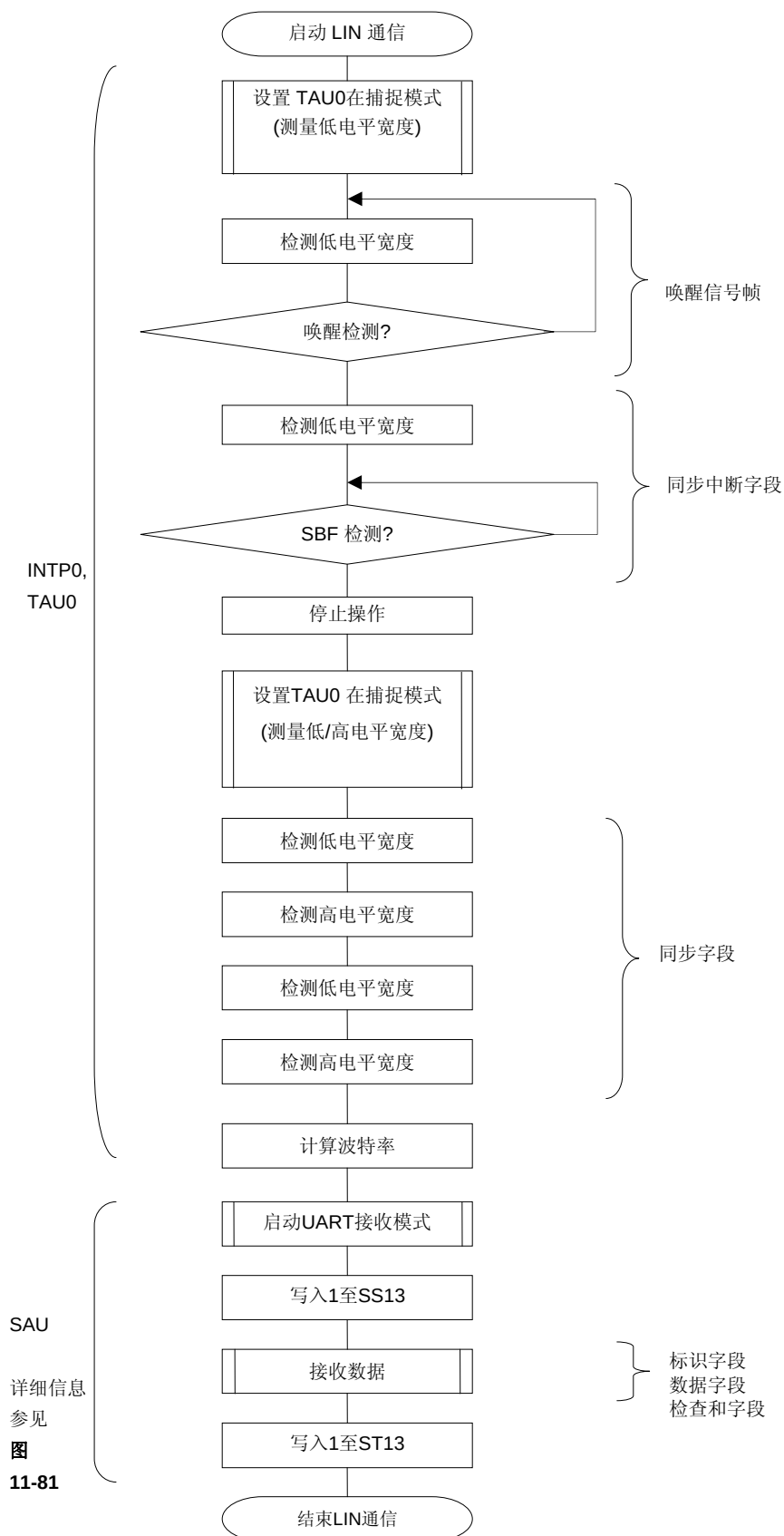
备注 ISC0、ISC1：输入切换控制寄存器 (ISC) 的第 0 位和第 1 位(参阅 图 11-17。)

用于 LIN 通信的外围功能如下。

<使用的外围功能>

- 外部中断(INTP0)；唤醒信号检测
用途： 检测唤醒信号边沿和通信开始
- 定时器阵列单元(TAU0) 的通道 7；波特率错误检测。
用途： 检测同步区域(SF)的长度并除以位数从而检测错误(输入到 RxD3 的边沿间隔在捕捉模式中测量。)
- 串行阵列单元(SAU1)的通道 2 和 3(UART3)

图 11-86. LIN 接收的流程图



11.6.5 计算波特率

(1) 波特率计算公式

UART(UART0 至 UART3)通信的波特率可以按照下列表达式来计算。

$$(\text{波特率}) = \{ \text{目标通道的工作时钟(MCK)频率} \} \div (\text{SDRmn}[15:9] + 1) \div 2 \text{ [bps]}$$

注意事项 禁止设置 $\text{SDRmn}[15:9] = (0000000\text{B}, 0000001\text{B})$ 。

- 备注**
1. 当使用 UART 时，SDRmn[15:9]的值是 SDRmn 寄存器的位 15 至位 9 的值(0000010B 至 1111111B)，因此它的取值为 2 至 127。
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13

工作时钟(MCK)由串行时钟选择寄存器 m (SPSm)和串行模式寄存器 mn (SMRmn)的位 15(CKSmn)来确定。

表 11-3. 工作时钟的选择

SMRmn 寄存器	SPS _m 寄存器								工作时钟(MCK) ^{‡1}	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f _{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f _{CLK}	20 MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	10 MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	5 MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	2.5 MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	1.25 MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	625 kHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	313 kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	156 kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	78.1 kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	39.1 kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	19.5 kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{‡2}	
1	0	0	0	0	X	X	X	X	f _{CLK}	20 MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	10 MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	5 MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	2.5 MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	1.25 MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	625 kHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	313 kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	156 kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	78.1 kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	39.1 kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	19.5 kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{‡2}	
其它值									设置禁止	

- 注**
1. 当改变用于 f_{CLK} 的时钟(通过改变系统时钟控制寄存器 (CKC) 的值)时, 在停止串行阵列单元 (SAU) 的操作 (STm = 000FH) 之后, 才这样做。当选择 INTTM02 和 INTTM03 为工作时钟时, 同样停止定时器阵列单元(TAU0) (TT0 = 00FFH)。
 2. 无论 f_{CLK} 的频率为多少(主系统时钟, 副系统时钟), 通过操作间隔定时器选择 f_{SUB}/4 作为计数时钟 (设定 TIS0 寄存器的 TIS02 (如果 m = 0)或 TIS03 (如果 m = 1)为 1), 并且通过 SPSm 寄存器中 TAU0 选择 INTTM02 和 INTTM03, 从而 SAU 可工作在固定的副系统时钟分频比。但是, 当改变 f_{CLK}, 必须按照上面注 1 描述的停止 SAU 和 TAU0。

备注 1. X: 不必考虑

2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13

(2) 发送期间的波特率误差

发送期间的 UART (UART0 至 UART3)通信的波特率误差可以通过下面公式计算。确保发送方的波特率在接收方的允许波特率范围之内。

$$(\text{波特率出错}) = (\text{计算的波特率数值}) \div (\text{目标波特率}) \times 100 - 100 [\%]$$

这在 $f_{\text{CLK}} = 20 \text{ MHz}$ 下设置 UART 波特率的例子。

UART 波特率 (目标波特率)	$f_{\text{CLK}} = 20 \text{ MHz}$			
	工作时钟(MCK)	SDRmn[15:9]	计算的波特率	目标波特率的误差
300 bps	$f_{\text{CLK}}/2^9$	64	300.48 bps	+0.16 %
600 bps	$f_{\text{CLK}}/2^8$	64	600.96 bps	+0.16 %
1200 bps	$f_{\text{CLK}}/2^7$	64	1201.92 bps	+0.16 %
2400 bps	$f_{\text{CLK}}/2^6$	64	2403.85 bps	+0.16 %
4800 bps	$f_{\text{CLK}}/2^5$	64	4807.69 bps	+0.16 %
9600 bps	$f_{\text{CLK}}/2^4$	64	9615.38 bps	+0.16 %
19200 bps	$f_{\text{CLK}}/2^3$	64	19230.8 bps	+0.16 %
31250 bps	$f_{\text{CLK}}/2^3$	39	31250.0 bps	±0.0 %
38400 bps	$f_{\text{CLK}}/2^2$	64	38461.5 bps	+0.16 %
76800 bps	$f_{\text{CLK}}/2$	64	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	64	153846 bps	+0.16 %
312500 bps	f_{CLK}	31	312500 bps	±0.0 %

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 00, 02, 10, 12

(3) 接收允许的波特率范围

在 UART (UART0 至 UART3) 通信期间，用于接收允许的波特率范围可以通过下面公式计算。确保发送方的波特率在接收方的允许波特率范围之内。

$$(\text{最大接收波特率}) = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$$

$$(\text{最小接收波特率}) = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$$

Brate: 在接收方计算的波特率 (参见 11.6.5 (1) 波特率计算公式。)

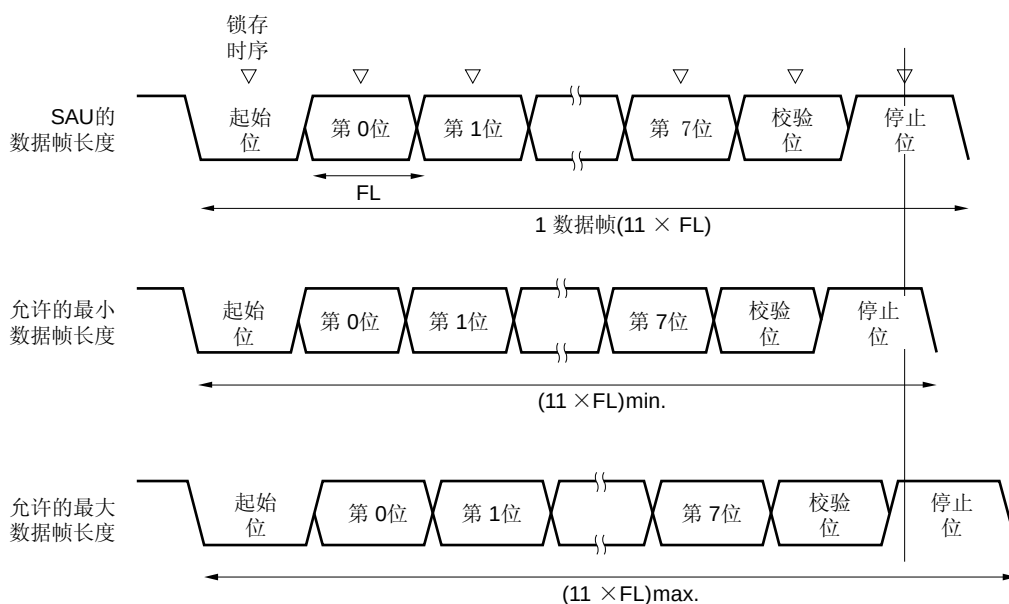
k: SDRmn[15:9] + 1

Nfr: 1 数据帧长度[位]

= (起始位) + (数据长度) + (校验位) + (停止位)

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 1, 3), mn = 01, 03, 11, 13

图 11-87. 接收允许的波特率范围 (1 数据帧长度 = 11 位)



如图 11-87 所示，在检测到起始位之后，锁存接收数据的时序由分频比决定，而分频比是通过设置串行数据寄存器 mn (SDRmn) 的位 15 至位 9 获得。如果在获得这个时序之前接收到最后一个数据(停止位)，可以正确接收这个数据。

11.7 简易 I²C (IIC10, IIC20)通信操作

这是一个通过使用 2 线：串行时钟(SCL)和串行数据(SDA)，用于和两个或者更多设备通信的时钟通信功能。这个通信功能计算执行具有例如 EEPROM，flash 存储器，和 A/D 转换器设备的单通信，因此，仅可以通过主使用并且不具有等待检测功能。确保通过使用软件，和操作控制寄存器，可以读取起始停止条件的 AC 特性。

[数据发送/接收]

- 主发送、主接收(仅具有主功能的主设备)
- ACK 输出功能^{*}和 ACK 检测功能
- 8 位数据长度
(当发送一个地址，高 7 位描述地址，最后一位用于 R/W 控制。)
- 手动产生启动条件和停止条件

[中断功能]

- 传输结束中断

[错误检测标志]

- 校验错误(ACK 错误)

* [简易 I²C 不支持的功能]

- 从发送、从接收
- 仲裁丢失检测功能
- 等待检测功能

注 当通过写入 0 到 SOEmn(SOEm 寄存器)位接收最后数据时，ACK 不输出并且停止串行通信数据的输出。详情参见 **11.7.3 (2) 处理流程**。

备注 1. 要使用 I²C 总线的全部功能，参见 **第 12 章 串行接口 IICA**。
2. m: 单元编号(m = 0, 1), n: 通道编号(n = 0, 2), mn = 02, 10

支持简易 I²C(IIC10, IIC20)的通道是 SAU0 的通道 2 和 SAU1 的通道 0。

单位	通道	用作CSI	用作UART	用作简易I ² C
0	0	CSI00	UART0	—
	1	CSI01		—
	2	CSI10	UART1	IIC10
	3	—		—
1	0	CSI20	UART2	IIC20
	1	—		—
	2	—	UART3(支持LIN总线)	—
	3	—		—

简易 I²C (IIC10, IIC20) 执行以下 4 种类型的通信操作。

- 发送地址区域 (见 **11.7.1**.)
- 发送数据 (见 **11.7.2**.)
- 接收数据 (见 **11.7.3**.)
- 产生停止条件 (见 **11.7.4**.)

11.7.1 发送地址区域

发送地址区域是一个第一次执行 I²C 通信来定义主(从)目标的传输操作。产生启动条件后，在一帧里发送一个地址(7 位)和传输方向(1 位)。

简易 I ² C	IIC10	IIC20
目标通道	SAU0 的通道 2	SAU1 的通道 0
使用引脚	SCL10, SDA10 [*]	SCL20, SDA20 [*]
中断	INTIIC10	INTIIC20
	仅传输结束中断(禁止设置缓冲空中断。)	
错误检测标志	奇偶校验错误检测标志 (PEFmn)	
传输数据长度	8 位(高 7 位作为地址，最后一位用于 R/W 控制)	
传输速率	最大值 $f_{CLK}/4$ MHz f_{CLK} : 系统时钟频率 然而，I ² C 的每个模式下必须满足下列条件。 • 最大值 400 kHz(第一模式) • 最大值 100 kHz(标准模式)	
数据电平	正向输出(默认: 高电平)	
校验位	无检验位	
停止位	附加 1 位(用于 ACK 接收时序)	
数据方向	首先是 MSB	

注 为了通过简易 I²C 进行通信，对于端口输出模式寄存器(POM0, POM14)设定 N 沟道开源输出(V_{DD} 容限)模式(POM03, POM143 = 1)(详情参见 4.3 寄存器控制端口功能)。当与不同电势的外部设备通信时，也可以对时钟输入/输出引脚(SCL10, SCL20)设置 N 沟道开源输出(V_{DD} 容限)模式(POM04, POM142 = 1)(详情参见 4.4.4 连接至不同电势的外部设备(2.5V, 3V))。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 02, 10

(1) 寄存器设置

图 11-88. 简易 I²C (IIC10, IIC20)地址域发送寄存器内容的举例

(a) 串行输出寄存器 m (SOM)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	1	CKOm2 0/1	CKOm1 ×	CKOm0 0/1	0	0	0	0	1	SOM2 0/1	SOM1 ×	SOM0 0/1

通过操作 SOMn 位，产生启动条件。

(b) 串行输出使能寄存器 m (SOEm)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	SOEm1 ×	SOEm0 0/1

直到产生启动条件，SOEmn = 0，并且在产生后 SOEmn = 1。

(c) 串行通道启动寄存器 m (SSm)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

(d) 串行模式寄存器 mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0						STSmn 0		SISmn0 0				MDmn2 1	MDmn1 0	MDmn0 0

通道 n 的工作模式
0: 传输结束中断

(e) 串行通信操作设置寄存器 mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0		EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0		SLCmn1 0	SLCmn0 1		DLsmn2 1	DLsmn1 1	DLsmn0 1

设置校验位
00B: 无校验

设置停止位
01B: 附加 1 位(ACK)

(f) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOr)

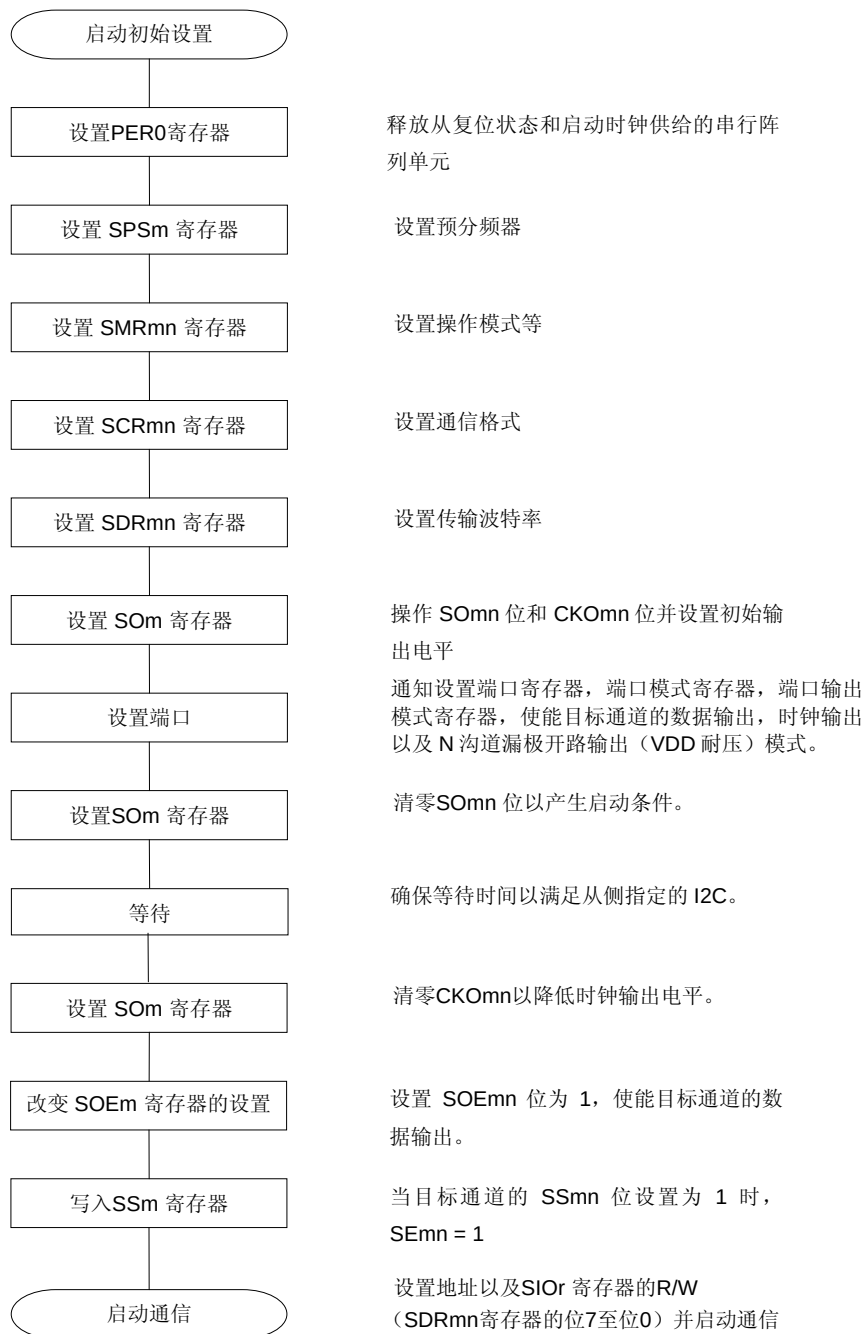
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	设置波特率							0	设置发送数据 (地址 + RW)							

SIOr

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 02, 10, r: IIC 编号 (r = 10, 20)
 □.: 在 IIC 模式下固定设置, ■.: 禁止设置(设为初始值)
 x: 此模式下不能使用的位(不在任何模式下使用时, 设为初始值)
 0/1: 根据用户的使用设置为 0 或 1

(2) 操作程序

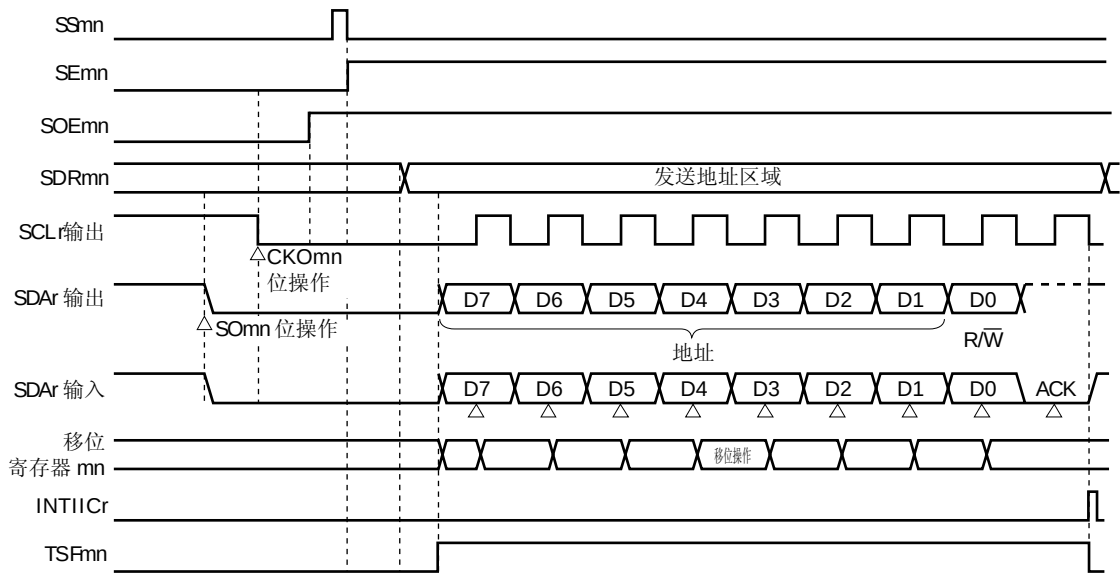
图 11-89. 地址域发送的初始化程序



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPSm 寄存器进行设置。

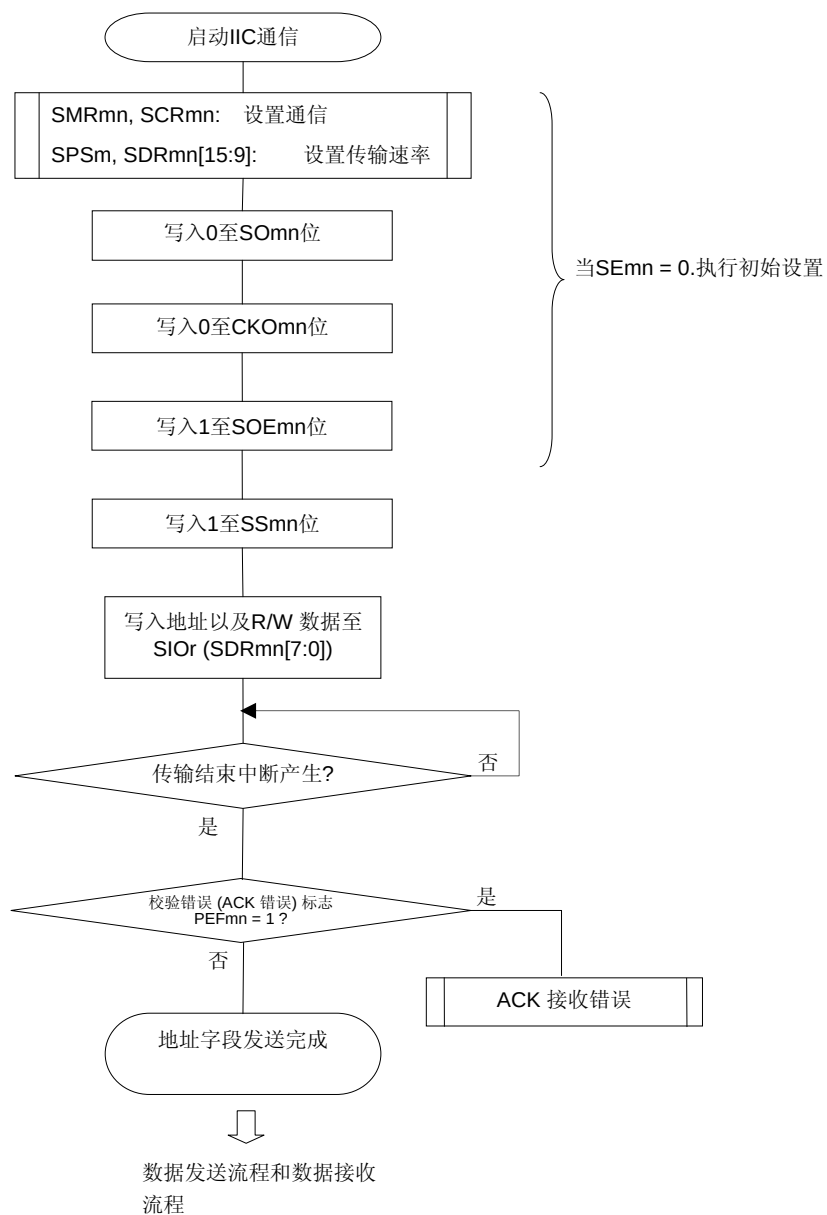
(3) 处理流程

图 11-90. 地址域发送的时序图



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 02, 10, r: IIC 编号 (r = 10, 20)

图 11-91. 地址域发送的流程图



11.7.2 数据发送

数据发送是一个在接收到地址域后传输(从)目标数据的操作。全部数据发送到从设备后，产生一个停止条件并且总线释放。

简易I ² C	IIC10	IIC20
目标通道	SAU0的通道2	SAU1的通道0
使用引脚	SCL10, SDA10 [※]	SCL20, SDA20 [※]
中断	INTIIC10	INTIIC20
	仅传输结束中断(禁止设置缓冲空中断。)	
错误检测标志	奇偶校验错误检测标志 (PEFmn)	
传输数据长度	8位	
传输速率	最大值 $f_{CLK}/4$ MHz f_{CLK} : 系统时钟频率 然而, I ² C的每个模式下必须满足下列条件。 • 最大值 400 kHz(第一模式) • 最大值 100 kHz(标准模式)	
数据电平	正向输出(默认: 高电平)	
校验位	无校验位	
停止位	附加1位(用于ACK接收时序)	
数据方向	首先是MSB	

注 为了通过简易 I²C 进行通信, 对于端口输出模式寄存器(POM0, POM14)设定 N 沟道开源输出(V_{DD} 容限)模式 (POM03, POM143 = 1)(详情参见 **4.3 寄存器控制端口功能**)。当与不同电势的外部设备通信时, 也可以对时钟输入/输出引脚(SCL10, SCL20)设置 N 沟道开源输出(V_{DD} 容限)模式(POM04, POM142 = 1)(详情参见 4.4.4 连接至不同电势的外部设备(2.5V, 3V))。

备注 m: 单元编号(m = 0, 1), n: 通道编号(n = 0, 2), mn = 02, 10

(1) 寄存器设置

图 11-92. 简易 I²C(IIC10, IIC20)的数据发送寄存器内容示例

(a) 串行输出寄存器 m (SOM) ...不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM	0	0	0	0	1	CKOm2 0/1 ^注	CKOm1 ×	CKOm0 0/1 ^注	0	0	0	0	1	SOm2 0/1 ^注	SOm1 ×	SOm0 0/1 ^注

(b) 串行输出使能寄存器 m (SOEm) ...不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	SOEm1 ×	SOEm0 0/1

(c) 串行通道启动寄存器 m (SSm) ...不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

(d) 串行模式寄存器 mn (SMRmn) ...不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SiSmn0 0	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(e) 串行通信操作设置寄存器 mn (SCRmn) ...不要操作该寄存器的位，

除了位 TXEmn 和

RXEmn 位，在数据

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0	SLCmn0 1	0	DLSmn2 1	DLSmn1 1	DLSmn0 1

(f) 串行数据寄存器 mn (SDRmn) (低 8 位: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	设置波特率							0	设置发送数据							
									SIOr							

注 该值由通信操作过程中的通信数据决定。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 02, 10, r: IIC 编号 (r = 10, 20)

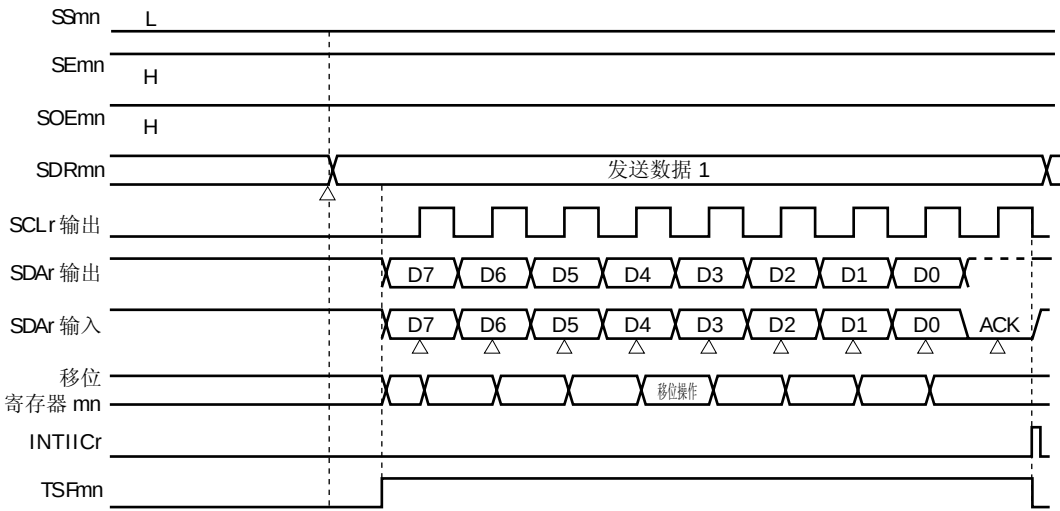
□.: 在 IIC 模式下固定设置, ..□: 禁止设置(设为初始值)

x: 此模式下不能使用的位(不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1

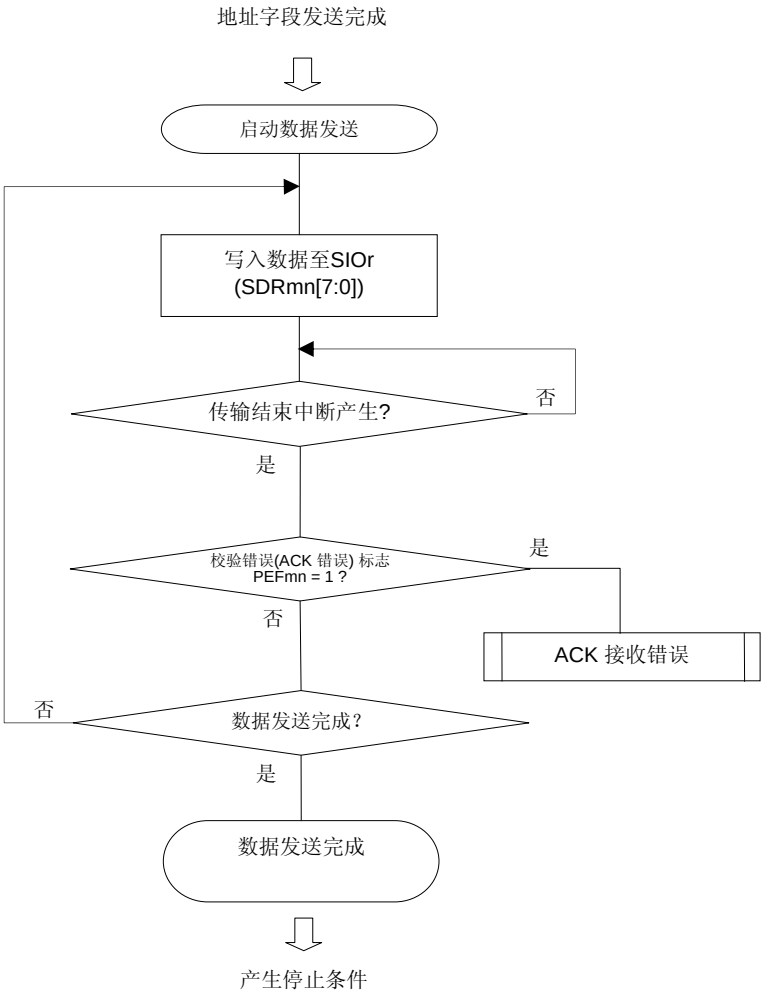
(2) 处理流程

图 11-93. 数据发送的时序图



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 02, 10, : IIC 编号 (r = 10, 20)

图 11-94. 数据发送的流程图



11.7.3 数据接收

数据接收是一个在发送到地址域后接收传输(从)目标数据的操作。从设备的数据全部接收后，产生一个停止条件并且总线释放。

简易I ² C	IIC10	IIC20
目标通道	SAU0的通道2	SAU1的通道0
使用引脚	SCL10, SDA10 [*]	SCL20, SDA20 [*]
中断	INTIIC10	INTIIC20
	仅传输结束中断(禁止设置缓冲空中断。)	
错误检测标志	无	
传输数据长度	8位	
传输速率	最大值 $f_{CLK}/4$ MHz f_{CLK} : 系统时钟频率 然而, I ² C的每个模式下必须满足下列条件。 • 最大值 400 kHz(第一模式) • 最大值 100 kHz(标准模式)	
数据电平	正向输出(默认: 高电平)	
校验位	无检验位	
停止位	附加1位(用于ACK接收时序)	
数据方向	首先是MSB	

注 为了通过简易 I²C 进行通信, 对于端口输出模式寄存器(POM0, POM14)设定 N 沟道开源输出(V_{DD} 容限)模式 (POM03, POM143 = 1)(详情参见 **4.3 寄存器控制端口功能**)。当与不同电势的外部设备通信时, 也可以对时钟输入/输出引脚(SCL10, SCL20)设置 N 沟道开源输出(V_{DD} 容限)模式(POM04, POM142 = 1)(详情参见 **4.4.4 连接至不同电势的外部设备(2.5V, 3V)**)。

备注 m: 单元编号(m = 0, 1), n: 通道编号(n = 0, 2,)mn = 02, 10

(1) 寄存器设置

图 11-95. 简易 I²C(IIC10, IIC20)的数据接收寄存器内容示例

(a) 串行输出寄存器 m (SOM) ...不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM						CKOm2	CKOm1	CKOm0						SOM2	SOM1	SOM0
	0	0	0	0	1	0/1 ^注	×	0/1 ^注	0	0	0	0	1	0/1 ^注	×	0/1 ^注

(b) 串行输出使能寄存器 m (SOEm) ...不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm														SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0/1

(c) 串行通道启动寄存器 m (SSm) ...不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm														SSm3	SSm2	SSm1
	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	×	0/1

(d) 串行模式寄存器 mn (SMRmn)...不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(e) 串行通信操作设置寄存器 mn (SCRmn)...不要操作该寄存器的位，

除了位 TXEmn 和

RXEmn 位，在数据

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0		DLSmn2	DLSmn1	DLSmn0
	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(f) 串行数据寄存器 mn (SDRmn) (低 8 位: SIO_r)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	设置波特率								0	设置伪发送数据 (FFH)						
										SIO _r						

注 该值由通信操作过程中的通信数据决定。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 02, 10, r: IIC 编号 (r = 10, 20)

□.: 在 IIC 模式下固定设置, ■.: 禁止设置(设为初始值)

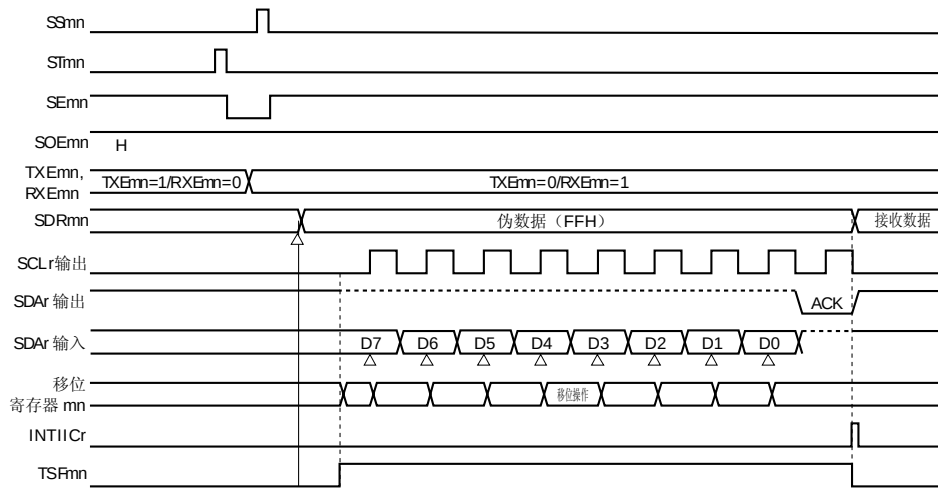
x: 此模式下不能使用的位(不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1

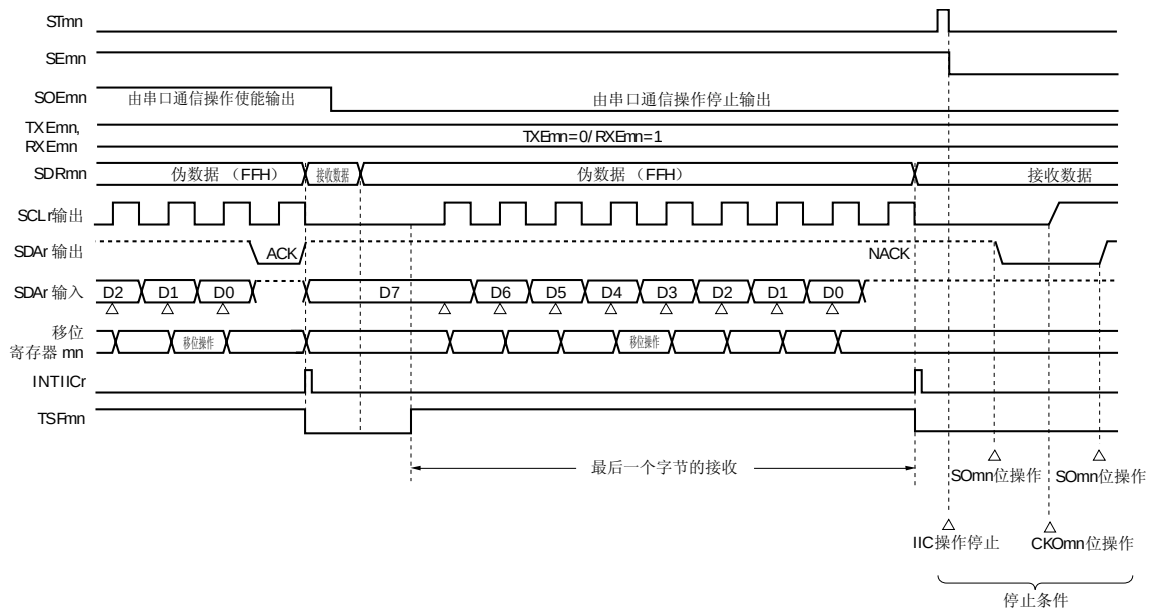
(2) 处理流程

图 11-96. 数据接收的时序图

(a) 启动数据接收

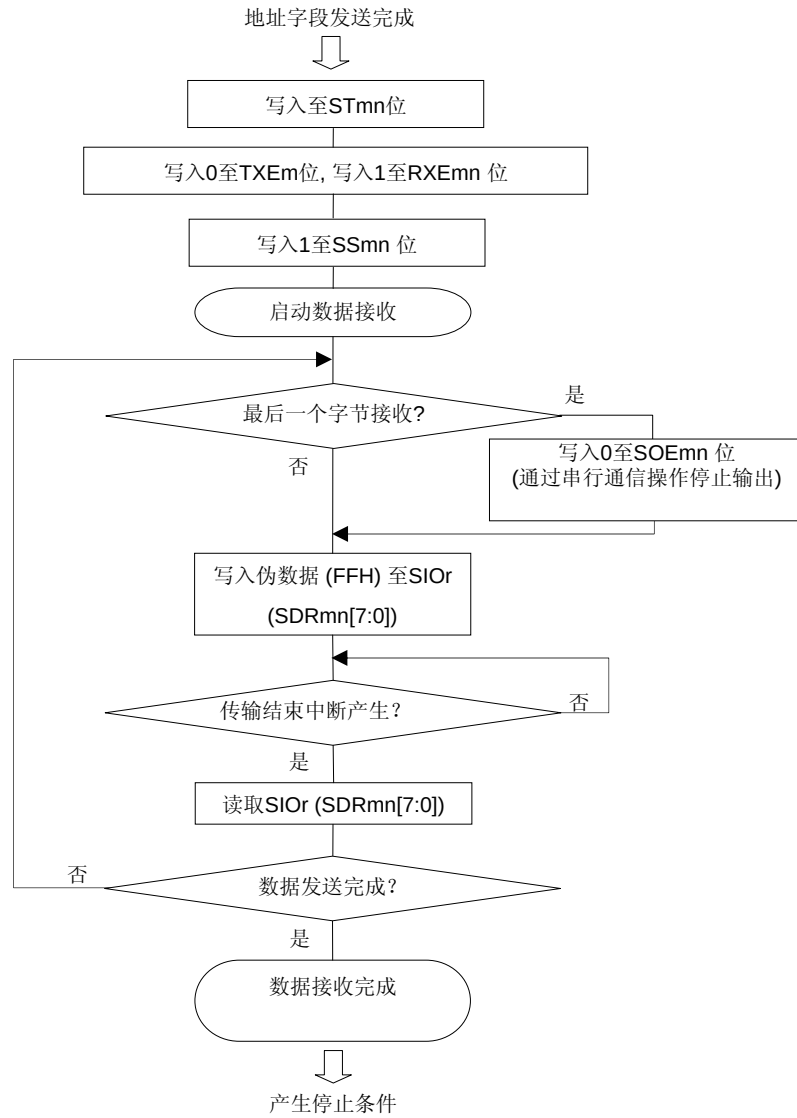


(b) 接收最后数据



备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 02, 10, : IIC 编号 (r = 10, 20)

图 11-97. 数据接收的流程图



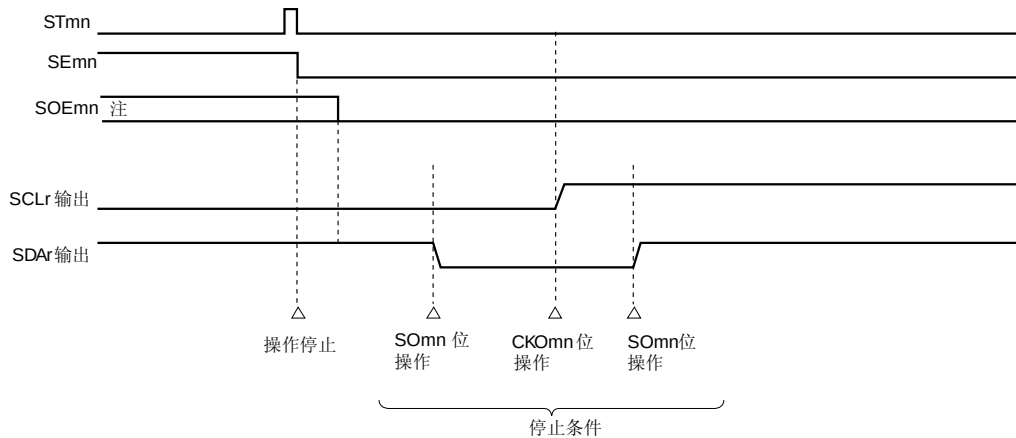
注意事项 当接收到最后数据时，ACK 不输出(NACK)。通过设置“1”至 STmn 位来停止操作并产生停止条件，从而完成通信。

11.7.4 停止条件的产生

全部数据发送到从设备或者从从设备读取到全部数据后，产生一个停止条件并且总线释放。

(1) 处理流程

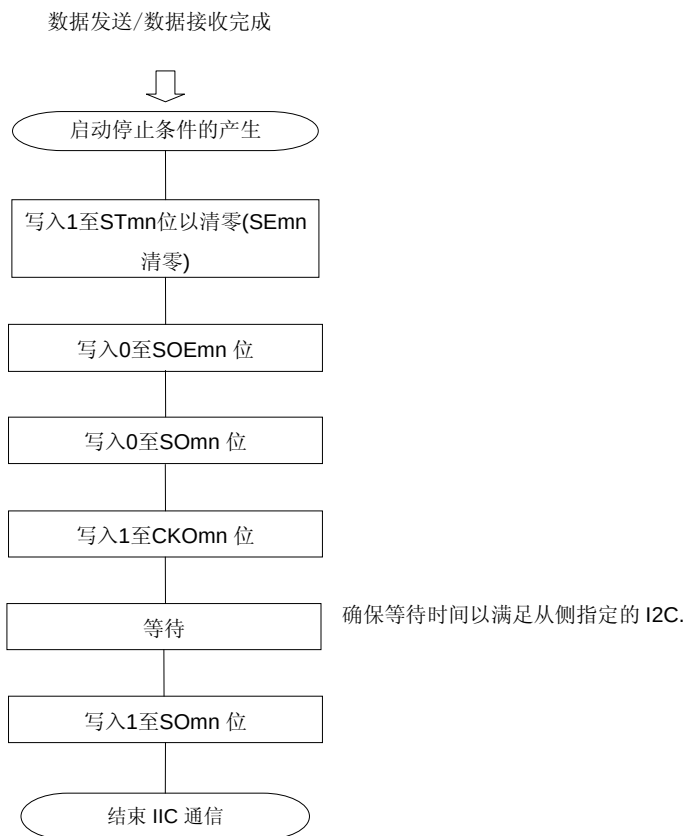
图 11-98. 停止条件的产生的时序图



注 接收操作期间，接收最后数据之前，将 SOEmn 位设置为 0。

备注 m: 单元编号 ($m = 0, 1$), n: 通道编号 ($n = 0, 2$), mn = 02, 10, r: IIC 编号 ($r = 10, 20$)

图 11-99. 停止条件的产生的流程图



11.7.5 计算传输速率

简易 I²C (IIC10, IIC20) 通信的传输速率可以按照下列表达式来计算。

$$(\text{传输速率}) = \{\text{目标通道的工作时钟(MCK)频率}\} \div (\text{SDRmn}[15:9] + 1) \div 2$$

- 备注**
1. SDRmn[15:9]的值是 SDRmn 寄存器的位 15 至位 9 的值(0000000B 至 1111111B)，因此它的取值范围为 0 至 127。
 2. m: 单元编号(m = 0, 1), n: 通道编号(n = 0, 2), mn = 02, 10

工作时钟(MCK)由串行时钟选择寄存器 m (SPSm)和串行模式寄存器 mn(SMRmn)的位 15(CKSmn)来确定。

表 11-4. 工作时钟的选择

SMRmn 寄存器	SPSm 寄存器								工作时钟(MCK) ^{‡1}	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f _{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f _{CLK}	20 MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	10 MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	5 MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	2.5 MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	1.25 MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	625 kHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	313 kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	156 kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	78.1 kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	39.1 kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	19.5 kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{‡2}	
1	0	0	0	0	X	X	X	X	f _{CLK}	20 MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	10 MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	5 MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	2.5 MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	1.25 MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	625 kHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	313 kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	156 kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	78.1 kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	39.1 kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	19.5 kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 如果 m = 0, INTTM03 如果 m = 1 ^{‡2}	
其它值									设置禁止	

- 注**
1. 当改变用于 f_{CLK} 的时钟(通过改变系统时钟控制寄存器 (CKC) 的值)时, 在停止串行阵列单元 (SAU) 的操作 (STm = 000FH) 之后, 才这样做。当选择 INTTM02 和 INTTM03 为工作时钟时, 同样停止定时器阵列单元(TAU0) (TT0 = 00FFH)。
 2. 无论 f_{CLK} 的频率为多少(主系统时钟, 副系统时钟), 通过操作间隔定时器选择 f_{SUB}/4 作为计数时钟 (设定 TIS0 寄存器的 TIS02 (如果 m = 0)或 TIS03 (如果 m = 1)为 1), 并且通过 SPSm 寄存器中 TAU0 选择 INTTM02 和 INTTM03, 从而 SAU 可工作在固定的副系统时钟分频比。但是, 当改变 f_{CLK}, 必须按照上面注 1 描述的停止 SAU 和 TAU0。

备注 1. X: 不必考虑

2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 02, 10

这有一个 $MCK = f_{CLK} = 20\text{ MHz}$ 下设置 IIC 传输速率的例子。

IIC 传输模式 预期传输速率)	$f_{CLK} = 20\text{ MHz}$			
	工作时钟(MCK)	SDRmn[15:9]	计算出来的传输速率	预期传输速率的误差
100 kHz	f_{CLK}	99	100 kHz	0.0%
400 kHz	f_{CLK}	24	400 kHz	0.0%

11.8 错误的处理程序

下面图 11-100 至 11-102 是如果发生错误的处理程序。

图 11-100. 校验错误或者溢出错误的处理程序

软件操作	硬件状态	备注
读SDRmn寄存器。	► BFF = 0, 并且通道n能够接收数据。	在错误处理中如果完成下一次接收防止溢出错误。
读SSRmn寄存器		定义错误类型并且读取用于清除错误标志的值。
写入SIRmn寄存器。	► 清除错误标志。	只有在读取点上产生的错误才能清除, 将从SSRmn寄存器读取的值写入到SIRmn寄存器而无须修改。

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 3), mn = 00 至 03, 10, 13

图 11-101. 帧错误时的处理程序

软件操作	硬件状态	备注
读SDRmn寄存器。	► BFF = 0, 并且通道n能够接收数据。	在错误处理中如果完成下一次接收防止溢出错误。
读SSRmn寄存器		定义错误类型并且读取用于清除错误标志的值。
写入SIRmn寄存器。	► 清除错误标志。	只有在读取点上产生的错误才能清除, 将从SSRmn寄存器读取的值写入到SIRmn寄存器而无须修改。
设置STmn位为1。	► SEMn = 0, 通道n停止操作。	
和其它部分通信同步		由于起始位发生移位的帧错误, 和其它部分通信的同步将重建, 并恢复通信。
设置SSmn位为1。	► SEMn = 1, 允许通道n操作。	

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3), mn = 00 至 03, 10 至 13

图 11-102. 简易 I²C 模式中校验错误(ACK 错误)的处理程序

软件操作	硬件状态	备注
读SDRmn寄存器。	► BFF = 0, 并且通道n能够接收数据。	在错误处理中如果完成下一次接收防止溢出错误。
读SSRmn寄存器		定义错误类型并且读取用于清除错误标志的值。
写入SIRmn寄存器。	► 清除错误标志。	只有在读取点上产生的错误才能清除, 将从SSRmn寄存器读取的值写入到SIRmn寄存器而无须修改。
设置STmn位为1。	► SEmn = 0, 通道n停止操作。	因为没有返回ACK从设备没有准备好接收。因此, 产生一个停止条件, 总线释放, 通信从启动条件重新开始。或者产生一个复位条件并且可以重新从地址发送开始。
产生停止条件		
产生启动条件		
设置SSmn位为1。	► SEmn = 1, 允许通道n操作。	

备注 m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0, 2), mn = 02, 10

11.9 寄存器设置和引脚之间的关系

表 11-5 至表 11-12 所示为串行阵列单元 0 和 1 的每个通道的寄存器设置和引脚之间的关系。

表 11-5. 寄存器设置和引脚之间的关系(单元 0 的通道 0: CSI00、UART0 发送)

SE0 ^{注 1}	MD002	MD001	SOE00	SO00	CKO00	TXE00	RXE00	PM10	P10	PM1 ^{注 2}	P11 ^{注 2}	PM12	P12	操作模式	引脚功能		
															SCK00/ P10	SI00/ RxD0/P11 ^{注 2}	SO00/ TxD0/P12
0	0	0	0	1	1	0	0	× 注 3	× 注 3	× 注 3	× 注 3	× 注 3	× 注 3	操作停止模式	P10	P11	P12
	0	1														P11/RxD0	
1	0	0	0	1	1	0	1	1	×	1	×	×	×	CSI00 从机接收	SCK00 (输入)	SI00	P12
			1	0/1 ^{注 4}	1	1	0	1	×	×	×	0	1	CSI00 从机发送	SCK00 (输入)	P11	SO00
			1	0/1 ^{注 4}	1	1	1	1	×	1	×	0	1	从设备 CSI00 发送/接收	SCK00 (输入)	SI00	SO00
			0	1	0/1 ^{注 4}	0	1	0	1	1	×	×	×	CSI00 主机接收	SCK00 (输出)	SI00	P12
			1	0/1 ^{注 4}	0/1 ^{注 4}	1	0	0	1	×	×	0	1	CSI00 主机发送	SCK00 (输出)	P11	SO00
			1	0/1 ^{注 4}	0/1 ^{注 4}	1	1	0	1	1	×	0	1	主设备 CSI00 发送/接收	SCK00 (输出)	SI00	SO00
	0	1	1	0/1 ^{注 4}	1	1	0	×	×	×	×	0	1	UART0 发送 ^{注 5}	P10	P11/RxD0	TxD0

- 注
1. SE0 寄存器是只读寄存器，使用 SS0 和 ST0 寄存器来设置它。
 2. 当单元 0 的通道 1 设置成 UART0 接收，该引脚变成一个 RxD0 功能引脚(参见表 11-6)。在这种情况下，单元 0 的通道 0 必须选择为操作停止模式或 UART0 发送。
 3. 该引脚可以设置为端口功能引脚。
 4. 取决于通信操作，该项为 0 或 1。详情请参阅 11.3 (12) 串行输出寄存器 m (S0m)。
 5. 当成对使用 UART0 发送和接收时，设置单元 0 的通道 1 为 UART0 接收(参阅表 11-6)。

备注 X: 不必考虑

表 11-6. 寄存器设置和引脚之间的关系(单元 0 的通道 1: CSI01、UART0 接收)

SE0 1 ^{注 1}	MD 012	MD 011	SOE 01	SO01	CKO 01	TXE 01	RXE 01	PM 43	P43	PM44	P44	PM 45	P45	PM1 1 ^{注 2}	P11 1 ^{注 2}	操作模式	引脚功能			
																	SCK01/ P43	SI01/P44	SO01/ P45	SI00/ RxD0/P11 ^{注 2}
0	0	0	0	1	1	0	0	× ^{注 3}	× ^{注 3}	× ^{注 3}	× ^{注 3}	× ^{注 3}	× ^{注 3}	× ^{注 3}	× ^{注 3}	操作停止 模式	P43	P44	P45	SI00/P11
	0	1																		
1	0	0	0	1	1	0	1	1	×	1	×	×	×	×	×	CSI01 从 机接收	SCK01 (输入)	SI01	P45	SI00/P11
			1	0/1 ^{注 4}	1	1	0	1	×	×	×	0	1	×	×	CSI01 从 机发送	SCK01 (输入)	P44	SO01	SI00/P11
			1	0/1 ^{注 4}	1	1	1	1	×	1	×	0	1	×	×	CSI01 从 机发送/接 收	SCK01 (输入)	SI01	SO01	SI00/P11
			0	1	0/1 ^{注 4}	0	1	0	1	1	×	×	×	×	×	CSI01 主 机接收	SCK01 (输出)	SI01	P45	SI00/P11
			1	0/1 ^{注 4}	0/1 ^{注 4}	1	0	0	1	×	×	0	1	×	×	CSI01 主 机发送	SCK01 (输出)	P44	SO01	SI00/P11
			1	0/1 ^{注 4}	0/1 ^{注 4}	1	1	0	1	1	×	0	1	×	×	CSI01 主 机发送/接 收	SCK01 (输出)	SI01	SO01	SI00/P11
			0	1	1	0	1	×	×	×	×	×	×	1	×	UART0 接 收 ^{注 5, 6}	P43	P44	P45	RxD0

- 注**
1. SE0 寄存器是只读寄存器，使用 SS0 和 ST0 寄存器来设置它。
 2. 当单元 0 的通道 1 设置成 UART0 接收，该引脚变成一个 RxD0 功能引脚。在这种情况下，将单元 0 的通道 0 设置为操作停止模式或 UART0 发送(参见表 11-5)。
当单元 0 的通道 0 设置为 CSI00，该引脚就不能用作一个 RxD0 功能引脚。在这种情况下，将单元 0 的通道 1 设置为操作停止模式或 CSI01。
 3. 该引脚可以设置为端口功能引脚。
 4. 取决于通信操作，该项为 0 或 1。详情请参阅 11.3 (12) 串行输出寄存器 m (SOM)。
 5. 当成对使用 UART0 发送和接收时，设置单元 0 的通道 0 为 UART0 发送(参阅表 11-5)。
 6. UART0 接收过程中，也必须设置单元 0 的通道 0 的 SMR00 寄存器。详细信息，请参阅 11.6.2 (1) 寄存器设置。

备注 X: 不必考虑

表 11-7. 寄存器设置和引脚之间的关系(单元 0 的通道 2: CSI10、UART1 发送、IIC10)

SE0 2 ^{注 1}	MD 022	MD 021	SOE 02	SO 02	CKO 02	TXE 02	RXE 02	PM 04	P04	PM03 注 2	P03 注 2	PM02	P02	操作模式	引脚功能							
															$\overline{\text{SCK10/}}$ SCL10/P04	SI10/SDA10/ RxD1/P03 注 2	SO10/ TxD1/P02					
0	0	0	0	1	1	0	0	× 注 3	× 注 3	× 注 3	× 注 3	× 注 3	× 注 3	操作停止模式	P04	P03	P02					
	0	1														P03/RxD1						
	1	0														P03						
1	0	0	0	1	1	0	1	1	×	1	×	× 注 3	× 注 3	CSI10 从机接收	$\overline{\text{SCK10}}$ (输入)	SI10	P02					
			1	0/1 注 4	1	1	0	1	×	× 注 3	× 注 3	0	1	CSI10 从机发送	$\overline{\text{SCK10}}$ (输入)	P03	SO10					
			1	0/1 注 4	1	1	1	1	×	1	×	0	1	CSI10 从机发送/接收	$\overline{\text{SCK10}}$ (输入)	SI10	SO10					
			0	1	0/1 注 4	0	1	0	1	1	×	× 注 3	× 注 3	CSI10 主机接收	$\overline{\text{SCK10}}$ (输出)	SI10	P02					
			1	0/1 注 4	0/1 注 4	1	0	0	1	× 注 3	× 注 3	0	1	CSI10 主机发送	$\overline{\text{SCK10}}$ (输出)	P03	SO10					
			1	0/1 注 4	0/1 注 4	1	1	0	1	1	×	0	1	CSI10 主机发送/接收	$\overline{\text{SCK10}}$ (输出)	SI10	SO10					
	0	1	1	0/1 注 4	1	1	0	×	× 注 3	× 注 3	0	1	UART1 发送 ^{注 5}	P04	P03/RxD1	TxD1						
0	1	0	0	0/1 注 6	0/1 注 6	0	0	0	1	0	1	× 注 3	× 注 3	IIC10 启动条件	SCL10	SDA10	P02					
						1	0															
						0	1															
			1	0/1 注 4	0/1 注 4	1	0	0	1	0	1	×	× 注 3	×	注 3	IIC10 地址域发送	SCL10	SDA10	P02			
						1	0/1 注 4	0/1 注 4	1	0	0	1	0	1	×	× 注 3	×	注 3	IIC10 数据 发送	SCL10	SDA10	P02
						1	0/1 注 4	0/1 注 4	0	1	0	1	0	1	×	× 注 3	×	注 3	IIC10 数据 接收	SCL10	SDA10	P02
			0	0/1 注 7	0/1 注 7	0	0	0	1	0	1	×	× 注 3	×	注 3	IIC10 停止条件	SCL10	SDA10	P02			
1	0																					
0	1																					

- 注
1. SE0 寄存器是只读寄存器，使用 SS0 和 ST0 寄存器来设置它。
 2. 当单元 0 的通道 3 设置成 UART1 接收，该引脚变成一个 RxD1 功能引脚(参见表 11-8)。在这种情况下，单元 0 的通道 2 必须选择为操作停止模式或 UART1 发送。

3. 该引脚可以设置为端口功能引脚。
4. 取决于通信操作，该项为 0 或 1。详情请参阅 11.3 (12) 串行输出寄存器 m (SOM)。
5. 当成对使用 UART1 发送和接收时，设置单元 0 的通道 3 为 UART1 接收(参阅表 11-8)。
6. 在产生一个启动条件之前，将 CKO02 位设为 1。当启动条件产生时，将 SO02 位由 1 清除为 0。
7. 在产生一个停止条件之前，将 CKO02 位设为 1。当停止条件产生时，将 SO02 位由 0 设为 1。

备注 X: 不必考虑

表 11-8. 寄存器设置和引脚之间的关系(单元 3 的通道 0: UART1 接收)

SE03 ^{注 1}	MD032	MD031	TXE03	RXE03	PM03 ^{注 2}	P03 ^{注 2}	操作模式	引脚功能
								SI10/SDA10/RxD1/P03 ^{注 2}
0	0	1	0	0	×	×	操作停止模式	SI10/SDA10/P03 ^{注 2}
1	0	1	0	1	1	×	UART1 接收 ^{注 4, 5}	RxD1

- 注
1. SE0 寄存器是只读寄存器，使用 SS0 和 ST0 寄存器来设置它。
 2. 当单元 0 的通道 3 设置成 UART1 接收，该引脚变成一个 RxD1 功能引脚。在这种情况下，将单元 0 的通道 2 设置为操作停止模式或 UART1 发送(参见表 11-7)。
当单元 0 的通道 2 设置为 CSI10 或 IIC10，该引脚就不能当作一个 RxD1 功能引脚使用。在这种情况下，将单元 0 的通道 3 设置为操作停止模式。
 3. 该引脚可以设置为端口功能引脚。
 4. 当成对使用 UART1 发送和接收时，设置单元 0 的通道 2 为 UART1 发送(参阅表 11-7)。
 5. UART1 接收过程中，也必须设置单元 0 的通道 2 的 SMR02 寄存器。详细信息，请参阅 11.6.2 (1)寄存器设置。

备注 X: 不必考虑

表 11-9. 寄存器设置和引脚之间的关系(单元 1 的通道 0: CSI20、UART2 发送、IIC20)

SE1 0 ^{注 1}	MD 102	MD 101	SOE 10	SO 10	CKO 10	TXE 10	RXE 10	PM 142	P14 2	PM1 43 ^{注 2}	P143 ^{注 2}	PM 144	P144	操作模式	引脚功能		
															$\overline{\text{SCK20/}}$ SCL20/P142	SI20/SDA20/ RxD2/P143 ^{注 2}	SO20/ TxD2/P144
0	0	0	0	1	1	0	0	$\times$ 注 3	$\times$ 注 3	$\times$ 注 3	$\times$ 注 3	$\times$ 注 3	$\times$ 注 3	操作停止模式	P142	P143	P144
	0	1														P143/RxD2	
	1	0														P143	
1	0	0	0	1	1	0	1	1	$\times$	1	$\times$	$\times$ 注 3	$\times$ 注 3	CSI20 从机接收	$\overline{\text{SCK20}}$ (输入)	SI20	P144
			1	0/1 注 4	1	1	0	1	$\times$	$\times$ 注 3	$\times$ 注 3	0	1	CSI20 从机发送	$\overline{\text{SCK20}}$ (输入)	P143	SO20
			1	0/1 注 4	1	1	1	1	$\times$	1	$\times$	0	1	CSI20 从机发送/接收	$\overline{\text{SCK20}}$ (输入)	SI20	SO20
			0	1	0/1 注 4	0	1	0	1	1	$\times$	$\times$ 注 3	$\times$ 注 3	CSI20 主机接收	$\overline{\text{SCK20}}$ (输出)	SI20	P144
			1	0/1 注 4	0/1 注 4	1	0	0	1	$\times$ 注 3	$\times$ 注 3	0	1	CSI20 主机发送	$\overline{\text{SCK20}}$ (输出)	P143	SO20
			1	0/1 注 4	0/1 注 4	1	1	0	1	1	$\times$	0	1	CSI20 主机发送/接收	$\overline{\text{SCK20}}$ (输出)	SI20	SO20
	0	1	1	0/1 注 4	1	1	0	$\times$ 注 3	$\times$ 注 3	$\times$ 注 3	$\times$ 注 3	0	1	UART2 发送 ^{注 5}	P142	P143/RxD2	TxD2
0	1	0	0	0/1 注 6	0/1 注 6	0	0	0	1	0	1	$\times$ 注 3	$\times$ 注 3	IIC20 启动条件	SCL20	SDA20	P144
						1	0										
						0	1										
1			1	0/1 注 4	0/1 注 4	1	0	0	1	0	1	$\times$ 注 3	$\times$ 注 3	IIC20 地址域发送	SCL20	SDA20	P144
			1	0/1 注 4	0/1 注 4	1	0	0	1	0	1	$\times$ 注 3	$\times$ 注 3	IIC20 数据 发送	SCL20	SDA20	P144
			1	0/1 注 4	0/1 注 4	0	1	0	1	0	1	$\times$ 注 3	$\times$ 注 3	IIC20 数据 接收	SCL20	SDA20	P144
0			0	0/1 注 7	0/1 注 7	0	0	0	1	0	1	$\times$ 注 3	$\times$ 注 3	IIC20 停止条件	SCL20	SDA20	P144
						1	0										
						0	1										

- 注
1. SE1 寄存器是只读寄存器，使用 SS1 和 ST1 寄存器来设置它。
 2. 当单元 1 的通道 1 设置成 UART2 接收，该引脚变成一个 RxD2 功能引脚(参见表 11-10)。在这种情况下，单元 1 的通道 0 必须选择为操作停止模式或 UART2 发送。
 3. 该引脚可以设置为端口功能引脚。

4. 取决于通信操作，该项为 0 或 1。详情请参阅 11.3 (12) 串行输出寄存器 m (SOM)。
5. 当成对使用 UART2 发送和接收时，设置单元 1 的通道 1 为 UART2 接收(参阅表 11-10)。
6. 在产生一个启动条件之前，将 CKO10 位设为 1。当启动条件产生时，将 SO10 位由 1 清除为 0。
7. 在产生一个停止条件之前，将 CKO10 位设为 1。当停止条件产生时，将 SO10 位由 0 设为 1。

备注 X: 不必考虑

表 11-10. 寄存器设置和引脚之间的关系(单元 1 的通道 1: UART2 接收)

SE11 ^{注 1}	MD112	MD111	TXE11	RXE11	PM143 ^{注 2}	P143 ^{注 2}	操作模式	引脚功能
								SI20/SDA20/RxD2/P143 ^{注 2}
0	0	1	0	0	× ^{注 3}	× ^{注 3}	操作停止模式	SI20/SDA20/P143
1	0	1	0	1	1	×	UART2 接收 ^{注 4, 5}	RxD2

- 注
1. SE1 寄存器是只读寄存器，使用 SS1 和 ST1 寄存器来设置它。
 2. 当单元 1 的通道 1 设置成 UART2 接收，该引脚变成一个 RxD2 功能引脚。在这种情况下，将单元 1 的通道 0 设置为操作停止模式或 UART2 发送(参见表 11-9)。
当单元 1 的通道 0 设置为 CSI20 或 IIC20，该引脚就不能当作一个 RxD2 功能引脚使用。在这种情况下，将单元 1 的通道 1 设置为操作停止模式。
 3. 该引脚可以设置为端口功能引脚。
 4. 当成对使用 UART2 发送和接收时，设置单元 0 的通道 1 为 UART2 发送(参阅表 11-9)。
 5. UART2 接收过程中，也必须设置单元 0 的通道 1 的 SMR10 寄存器。详细信息，请参阅 11.6.2 (1)寄存器设置。

备注 X: 不必考虑

表 11-11. 寄存器设置和引脚之间的关系(单元 2 的通道 1: UART3 发送)

SE12 ^{注 1}	MD122	MD121	SOE12	SO12	TXE12	RXE12	PM13 ^{注 2}	P13 ^{注 2}	操作模式	引脚功能
										TxD3/P13
0	0	1	0	1	0	0	×	×	操作停止模式	P13
1	0	1	1	0/1 ^{注 4}	1	0	0	1	UART3 发送 ^{注 5}	TxD3

- 注
1. SE1 寄存器是只读寄存器，使用 SS1 和 ST1 寄存器来设置它。
 2. 当单元 1 的通道 3 设置成 UART1 接收，该引脚变成一个 RxD3 功能引脚(参见表 11-12)。在这种情况下，将单元 1 的通道 2 设置为操作停止模式或 UART2 发送。
 3. 该引脚可以设置为端口功能引脚。
 4. 取决于通信操作，该项为 0 或 1。详情请参阅 11.3 (12) 串行输出寄存器 m (SOM)。
 5. 当成对使用 UART3 发送和接收时，设置单元 1 的通道 3 为 UART3 接收(参阅表 11-12)。

备注 X: 不必考虑

表 11-12. 寄存器设置和引脚之间的关系(单元 1 的通道 3: UART3 接收)

SE13 ^{注 1}	MD132	MD131	TXE13	RXE13	PM14 ^{注 2}	P14 ^{注 2}	操作模式	引脚功能
								RxD3/P14 ^{注 2}
0	0	1	0	0	×	×	操作停止模式	P14
1	0	1	0	1	1	×	UART3 接收 ^{注 4, 5}	RxD3

- 注
1. SE1 寄存器是只读寄存器，使用 SS1 和 ST1 寄存器来设置它。
 2. 当单元 1 的通道 1 设置成 UART3 接收，该引脚变成一个 RxD3 功能引脚。在这种情况下，将单元 1 的通道 2 设置为操作停止模式或 UART3 发送(参见表 11-11)。
 3. 该引脚可以设置为端口功能引脚。
 4. 当成对使用 UART3 发送和接收时，设置单元 1 的通道 2 为 UART3 发送(参阅表 11-11)。
 5. UART3 接收过程中，也必须设置单元 1 的通道 2 的 SMR12 寄存器。详细信息，请参阅 11.6.2 (1)寄存器设置。

备注 X: 不必考虑

12.1 串行接口 IICA 的功能

串行接口 IICA 有如下三种模式。

(1) 操作停止模式

不执行串行发送时采用此模式。因此它可用来降低功耗。

(2) I²C 总线模式(支持多个主设备)

带有多个设备的、用于 8 位数据传输的此模式通过两条线进行：串行时钟线 (SCL0) 和串行数据总线 (SDA0)。

该模式以 I²C 总线格式编译，并且主设备可通过串行数据总线产生“开始条件”、“地址”、“传输方向指定”、“数据”和“停止条件”的数据至从设备。从设备会通过硬件自动检测这些接收状态和数据。这个功能可以简易控制 I²C 总线的应用程序部分。

由于 SCL0 和 SDA0 引脚用于漏极开路输出，IICA 需要连接上拉电阻用于串行时钟线和串行数据总线。

(3) 唤醒模式

在 STOP 模式下，当主设备或本地地址接收到外部代码时，可通过产生中断请求信号(INTIICA)释放 STOP 模式。这可通过使用 IICA 控制寄存器 1(IICCTL1)的 WUP 位来设置。

图 12-1 所示为串行接口 IICA 的框图。

图 12-1. 串行接口 IICA 的框图

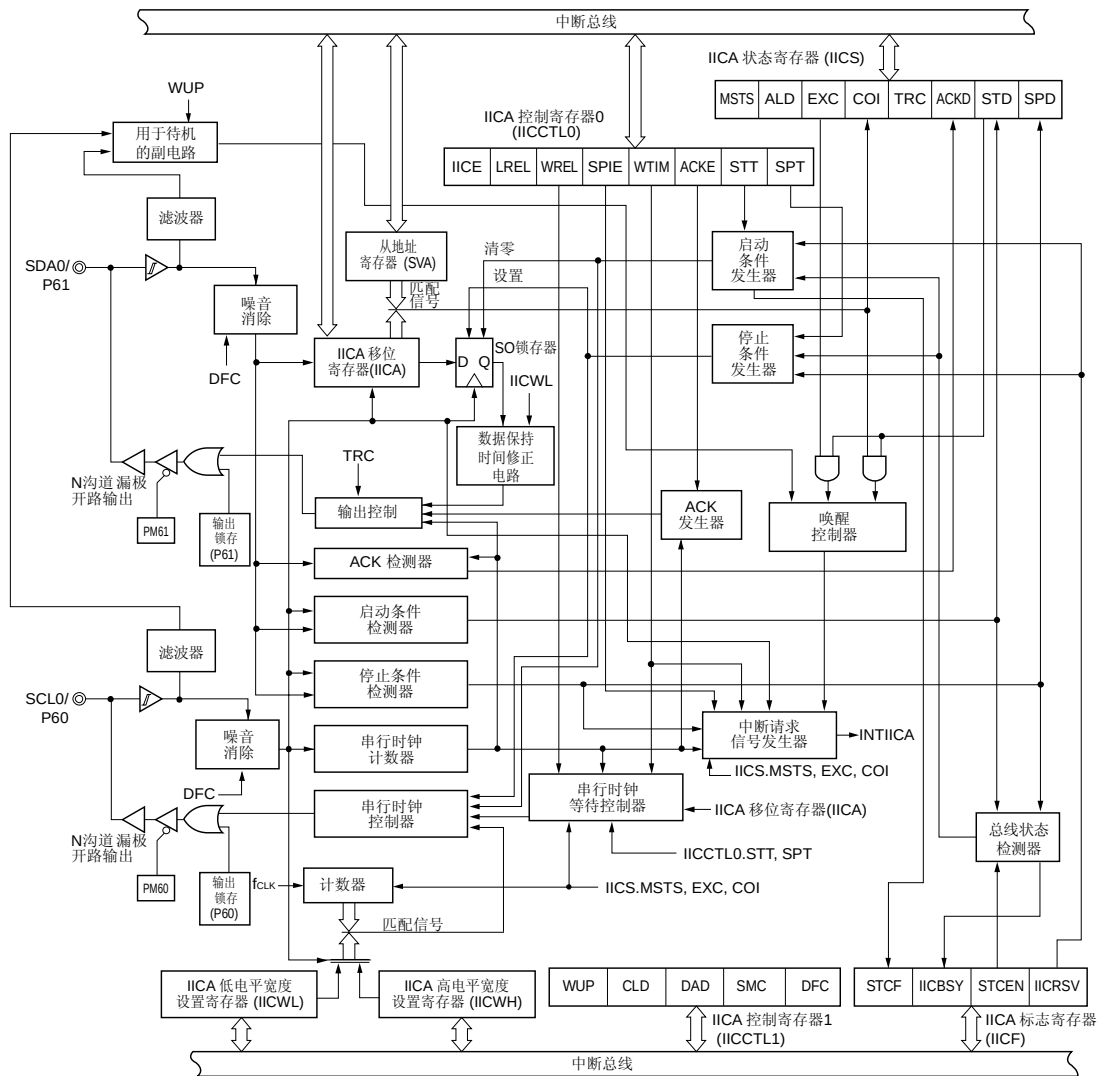
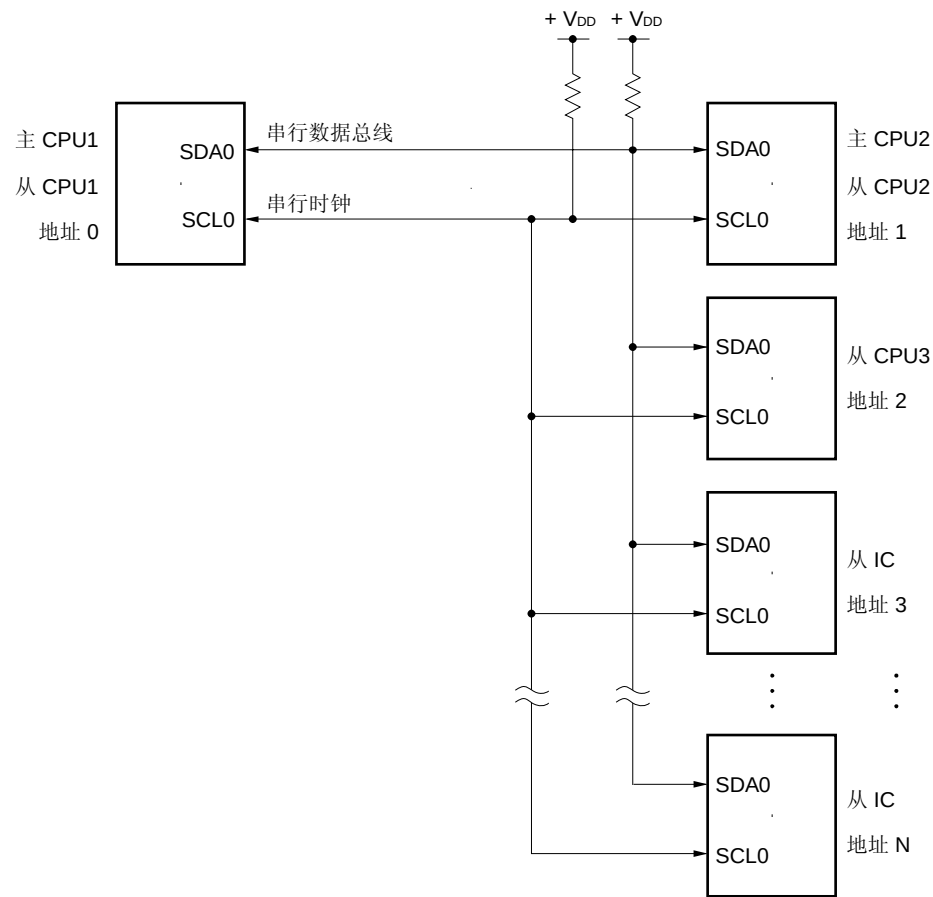


图 12-2 显示了串行总线配置示例。

图 12-2. 使用I²C总线的串行总线配置示例



12.2 串行接口 IICA 的配置

串行接口 IICA 包括如下硬件。

表 12-1. 串行接口 IICA 的配置

项目	配置
寄存器	IICA 移位寄存器(IICA) 从地址寄存器(SVA)
控制寄存器	外围使能寄存器 0 (PER0) IICA 控制寄存器 0 (IICCTL0) IICA 状态寄存器(IICS) IICA 标志寄存器(IICF) IICA 控制寄存器 1 (IICCTL1) IICA 低电平宽度设置寄存器 (IICWL) IICA 高电平宽度设置寄存器 (IICWH) 端口模式寄存器 6 (PM6) 端口寄存器 6 (P6)

(1) IICA 移位寄存器 (IICA)

IICA 用于把 8 位串行数据转换和 8 位并行数据进行相互转换，并且和串行时钟同步。IICA 既可以用来发送又可以用来接收。

对 IICA 的读写操作可以控制实际的发送和接收操作。

在等待期间，通过对 IICA 写入数据，可以取消等待状态并启动数据发送操作。

IICA 可以以 8 位存储器操作指令设置。

复位信号的产生将 IICA 清为 00H。

图 12-3. IICA 移位寄存器 0 的格式 (IICA)

地址: FFF50H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
IICA								

- 注意事项
1. 在数据发送期间，不要对 IICA 进行写操作。
 2. 仅在等待期间，才能对 IICA 进行读或写。除等待期间外，禁止在其它的通信状态下访问 IICA。
然而，当设备作为主设备时，在通信触发位 (STT) 设为 1 之后，IICA 只能写入一次。
 3. 当通信保留时，在检测到停止条件并引起中断触发之后将数据写入 IICA。

(2) 从地址寄存器 (SVA)

在从模式下，该寄存器储存 7 位本地地址{A6, A5, A4, A3, A2, A1, A0}。

SVA 可以以 8 位存储器操作指令设置。

但是，当 STD=1 时(当检测到启动条件时)，禁止重写该寄存器。

复位信号的产生将 SVA 清为 00H。

图 12-4. 从地址寄存器的格式 (SVA)

地址: F0234H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SVA	A6	A5	A4	A3	A2	A1	A0	0 ^注

注 第 0 位固定为 0。

(3) SO 锁存器

SO 锁存器用来保持 SDA0 引脚的输出电平。

(4) 唤醒控制器

当从主设备接收到的地址与从地址寄存器 (SVA) 中的设置地址值匹配，或者当接收到一个扩展代码时，该电路将产生一个中断请求信号 (INTIICA)。

(5) 串行时钟计数器

该计数器对发送/接收操作期间输出或输入的串行时钟进行计数，同时用于验证 8 位数据是否已发送或接收。

(6) 中断请求信号发生器

此电路控制中断请求信号 (INTIICA) 的产生。

一个 I²C 中断请求由下列两个触发条件产生：

- 串行时钟的第 8 或第 9 个下降沿(通过 WTIM 位设置)
- 当检测到一个停止条件时，产生中断请求(通过 SPIE 位设置)

备注 WTIM 位： IICA 控制寄存器 0 的第 3 位 (IICCTL0)

SPIE 位： IICA 控制寄存器 0 的第 4 位 (IICCTL0)

(7) 串行时钟控制器

在主设备模式下，此电路通过 SCL0 引脚，从采样时钟产生时钟输出。

(8) 串行时钟等待控制器

用于控制等待时序。

(9) ACK 信号发生器，停止条件检测器，启动条件检测器和 ACK 信号检测器

这些电路用于产生和检测每种状态。

(10) 数据保持时间修正电路

此电路根据串行时钟的下降沿产生数据保持时间。

(11) 启动条件发生器

当 STT 位设为 1 时，该电路产生一个启动条件。

但是，在禁止通信预约的状态下 (IICRSV 位 = 1)，只要总线尚未释放 (IICBSY 位 = 1)，启动条件请求将忽略并且 STCF 位置为 1。

(12) 停止条件发生器

当 SPT 位设为 1 时，此电路产生一个停止条件。

(13) 总线状态检测器

此电路通过检测启动条件和停止条件来确定总线是否释放。

然而，如果操作后总线状态无法立即检测到，则初始状态由 **STCEN** 位来设置。

备注	STT 位：	IICA 控制寄存器 0 的第 1 位 (IICCTL0)
	SPT 位：	IICA 控制寄存器 0 的第 0 位 (IICCTL0)
	IICRSV 位：	IICA 标志寄存器 (IICF) 的第 0 位
	IICBSY 位：	IICA 标志寄存器 (IICF) 的第 6 位
	STCF 位：	IICA 标志寄存器 (IICF) 的第 7 位
	STCEN 位：	IICA 标志寄存器 (IICF) 的第 1 位

12.3 控制串行接口 IICA 的寄存器

串行接口 IICA 由下列 9 个寄存器控制。

- 外围使能寄存器 0 (PER0)
- IICA 控制寄存器 0 (IICCTL0)
- IICA 标志寄存器 (IICF)
- IICA 状态寄存器 (IICS)
- IICA 控制寄存器 1 (IICCTL1)
- IICA 低电平宽度设置寄存器 (IICWL)
- IICA 高电平宽度设置寄存器 (IICWH)
- 端口模式寄存器 6 (PM6)
- 端口寄存器 6 (P6)

(1) 外围使能寄存器 0 (PER0)

PER0 用于允许或者禁止使用每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少消耗和噪音。

当使用串行接口 IICA 时，确保设置此寄存器的第 4 位 (IICAEN) 为 1。

可以通过 1 位或者 8 位存储器操作指令来设置 PER0。

复位后这个寄存器的值为 00H。

图 12-5. 外围使能寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

IICAEN	串行接口 IICA 输入时钟的控制
0	停止输入时钟的供给 • 由串行接口 IICA 使用的 SFR 不能写入。 • 串行接口 IICA 在复位状态。
1	供给输入时钟 • 由串行接口 IICA 使用的 SFR 能读/写。

注意事项 1. 设置串行接口 IICA 时，确保首先将 IICAEN 设置为 1。如果 IICAEN = 0，向串行接口 IICA 的控制寄存器写入忽略，即使读取该寄存器，也只能读到默认值。

2. 确保将 PER0 寄存器的位 6 清为 0。

(2) IICA 控制寄存器 0 (IICCTL0)

此寄存器用于允许 / 停止 I²C 的操作、设置等待时间和设置 I²C 的其它操作。

IICCTL0 可由 1 位或 8 位存储器操作指令设置。但是，当 IICE 位 = 0 或在等待期间，设置 SPIE、WTIM 和 ACKE 位。当 IICE 位从“0”设置为“1”时可同时设置这些位。

复位后这个寄存器的值为 00H。

图 12-6. IICA 控制寄存器 0 (IICCTL0) 的格式(1/4)

地址: F0230H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTL0	IICE	LREL	WREL	SPIE	WTIM	ACKE	STT	SPT

IICE	I ² C操作允许
0	停止操作。复位IICA状态寄存器 (IICS) ^{‡1} 。停止内部操作。
1	允许操作
当 SCL0 和 SDA0 是高电平时, 必须将该位设置为 1。	
清零条件 (IICE = 0)	设置条件 (IICE = 1)
- 通过指令进行清零 - 复位	通过指令进行设置

LREL ^{‡2,3}	退出通信
0	正常操作
1	此时从当前的通信模式退出, 进入待机状态。当执行此操作后设置的值自动清零。 用于那些已接受了本地无关扩展代码的情况。 SCL0 和 SDA0 线设置为高阻抗状态。 清除下列 IICA 控制寄存器 0(IICCTL0)和 IICA 状态寄存器(IICS)的标志。 • STT • SPT • MSTs • EXC • COI • TRC • ACKD • STD
从通信模式退出后一直保持待机状态, 直到满足下列通信条件。 • 检测到停止条件后, 在主设备模式下重新启动。 • 启动后, 发生地址匹配或接收到扩展代码。	
清零条件 (LREL = 0)	设置条件 (LREL = 1)
- 执行后自动清零 - 复位	通过指令进行设置

WREL ^{‡2,3}	等待取消
0	不取消等待
1	取消等待。在等待取消后, 此设置自动清除。
第 9 个时钟脉冲在发送状态 (TRC = 1) 下, 在等待期间设置 WREL(等待取消) 时, SDA0 线将进入高阻抗状态 (TRC = 0)。	
清零条件 (WREL = 0)	设置条件 (WREL = 1)
- 执行后自动清零 - 复位	通过指令进行设置

- 注
1. 复位 IICS 寄存器、IICF 寄存器的 STCF 和 IICBSY 位以及 IICCTL1 寄存器的 CLD 和 DAD 位。
 2. IICE0 为 0 时, 该位的信号无效。
 3. LREL 和 WREL 位的读值总是为 0。

注意事项 当 SCL0 处于高电平、SDA0 处于低电平, 并且 IICCTL1 寄存器的 DFC 为 1 时, 如果使能 I²C 操作 (IICE = 1) 的话, 将会立即检测到启动条件。I²C 操作允许 (IICE = 1) 后, 通过 1 位存储器操作指令, 立即设置 LREL (1)。

图 12-6. IICA 控制寄存器 0 (IICCTL0) 的格式(2/4)

SPIE ^{注 1}	当检测到停止条件时，允许/禁止产生中断请求
0	禁止
1	允许
如果 IICCTL1 寄存器的 WUP 为 1，那么即使 SPIE = 1 也不会产生停止条件的中断。	
清零条件 (SPIE = 0)	设置条件 (SPIE = 1)
- 通过指令进行清零 - 复位	通过指令进行设置

WTIM ^{注 1}	对等待和中断请求产生的控制
0	在第 8 个时钟的下降沿产生中断请求。 主设备模式：输出 8 个时钟之后，将时钟输出设置为低电平且设置等待状态。 从设备模式：输入 8 个时钟之后，将时钟设置为低电平且为主设备设置等待状态。
1	在第 9 个时钟的下降沿产生中断请求。 主设备模式：输出 9 个时钟之后，将时钟输出设置为低电平且设置等待状态。 从设备模式：输入 9 个时钟之后，将时钟设置为低电平且为主设备设置等待状态。
地址传输期间，在第 9 个时钟的下降沿产生一个中断，而与该位的设置无关。地址传输完成之后该位设置才有效。在主模式时，在地址传输的第九个时钟的下降沿插入等待。对于接收到本地地址的从设备，在发出响应(ACK)后的第九个时钟的下降沿插入等待。但是，当从设备已经收到一个扩展代码时，将在第 8 个时钟的下降沿插入一个等待信号。	
清零条件 (WTIM = 0)	设置条件 (WTIM = 1)
- 通过指令进行清零 - 复位	通过指令进行设置

ACKE ^{注 1,2}	响应控制
0	禁止响应
1	允许响应。在第九个时钟周期，SDA0 设置为低电平。
清零条件 (ACKE = 0)	设置条件 (ACKE = 1)
- 通过指令进行清零 - 复位	通过指令进行设置

- 注
1. IICE 为 0 时，该位的信号无效。在此期间设置此位。
 2. 在地址传输期间及编码为非扩展代码时，所设的值无效。
在从设备模式下并且地址匹配时，无论设置何值都将产生响应信号。

图 12-6. IICA 控制寄存器 0 (IICCTL0) 的格式(3/4)

STT ^注	启动条件触发
0	不产生启动条件
1	当总线释放时 (待机状态下, 当 IICBSY = 0): 产生一个启动条件 (作为主设备启动)。SCL0 处于高电平时, SDA0 从高电平变为低电平时产生启动条件。然后, 经过额定时间, SCL0 变为低电平 (等待状态)。 有第三方通信时: - 当允许通信预约时 (IICRSV = 0) - 该位作为启动条件预约标志。设置为 1 时, 在总线释放之后自动产生一个启动条件。 - 当禁止通信预约时 (IICRSV = 1) - STCF 设置为 1 且 STT 清零。不会产生启动条件。 等待状态 (主设备模式): 退出等待状态之后产生一个重复启动条件。
设置时序注意事项 - 主设备接收: 发送期间不能设为 1。仅在 ACKE 清零且最终接收已通报从设备时的等待期间, 才能设为 1。 - 主设备发送: 应答期间内, 不能正常产生起始条件。在第 9 个时钟输出后的等待期间可以设置为 1。 - 不能和 SPT 同时设为 1。 - STT 清零禁止前, 将其重复设置为 1。	
清零条件 (STT = 0)	设置条件 (STT = 1)
- 通信预约禁止时, 通过将 STT 设置为 1 进行清零。 - 由仲裁失败清零 - 通过主设备产生启动条件后进行清零 - 由 LREL = 1(退出通信)清零 - 当 IICE = 0(操作停止) - 复位	通过指令进行设置

注 IICE 为 0 时, 该位的信号无效。

备注 1. 在数据设置后再读取时, 第 1 位 (STT)将会变为 0。

2. IICRSV: IIC 标志寄存器(IICF)的第 0 位

STCF: IIC 标志寄存器(IICF)的第 7 位

图 12-6. IICA 控制寄存器 0 (IICCTL0) 的格式(4/4)

SPT	停止条件触发器
0	不产生停止条件
1	产生停止条件(结束主设备发送) SDA0 变为低电平后, 设置 SCL0 为高电平或等待其变为高电平。然后, 经过额定时间后, SDA0 从低电平变为高电平, 则产生一个停止条件。
设置时序注意事项 - 主设备接收: 发送期间不能设为 1。 仅在 ACKE 清零且最终接收已通报从设备时的等待期间, 才能设为 1。 - 主设备发送: 应答期间内, 不能正常产生停止条件。在第 9 个时钟输出后的等待期间可以设为 1。 - 不能和 STT 同时设为 1。 - 仅在主设备模式下, SPT 才能设置为 1。 - 当 WTIM 清零, 如果在 8 个时钟输出之后的等待期间将 SPT 设为 1, 那么请注意, 在第 9 个时钟的高电平期间将产生一个停止条件。在 8 个时钟输出后的等待期间, 应将 WTIM 从 0 变为 1, 并且在第 9 个时钟输出后的等待期间, SPT 应设置为 1。 - SPT 清零禁止前, 将其重复设置为 1。	
清零条件 (SPT = 0)	设置条件 (SPT = 1)
- 由仲裁失败清零 - 检测到停止条件后自动清零 - 由 LREL = 1(退出通信)清零 - 当 IICE = 0(操作停止) - 复位	通过指令进行设置

注意事项 当 IICA 状态寄存器 (IICS) 的第 3 位 (TRC) 设为 1, 在第 9 个时钟和等待取消期间, WREL 设置为 1 时。之后, TRC 清零且 SDA0 线设为高阻抗状态。

备注 数据设置后再读取时, 第 0 位 (SPT) 将会变为 0。

(3) IICA 状态寄存器 (IICS)

此寄存器用于表示 I²C 的状态。

仅当 STT = 1 且在等待期间, 才能通过 1 位或 8 位存储器操作指令读取 IICS。

复位后这个寄存器的值为 00H。

注意事项 当 IICA 控制寄存器 1 (IICCTL1) 设为 1 禁止时, 读取 IICA 状态寄存器 (IICS)。当 WUP 从 1 变为 0, 无论有何 INTIICA 中断请求, 改变状态都不会反映出来, 直到检测到下一个启动条件或停止条件。因此, 要使用唤醒模式, 检测到中断后, 由检测到停止条件产生的中断使能 (SPIE = 1), 并读取 IICS 寄存器。

图 12-7 IICA 状态寄存器格式 (IICS) (1/3)

地址: FFF51H

复位后: 00H

R

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS	MSTS	ALD	EXC	COI	TRC	ACKD	STD	SPD

MSTS	主设备状态
0	从设备状态或通信待机状态
1	主设备通信状态
清零条件 (MSTS = 0)	
- 检测到停止条件时 - ALD = 1 时(仲裁失败) - 由 LREL = 1(退出通信)清零 - 当 IICE 从 1 变为 0(操作停止)时 - 复位	
设置条件 (MSTS = 1)	
产生启动条件时	

ALD	仲裁失败的检测
0	该状态即可表示无仲裁, 也可表示仲裁结果为“胜利”。
1	该状态显示仲裁结果为“失败”。 MSTS 清零
清零条件 (ALD = 0)	
- 读取 IICS 之后自动清零^注 - 当 IICE 从 1 变为 0(操作停止)时 - 复位	
设置条件 (ALD = 1)	
当仲裁结果为“失败”时。	

EXC	扩展代码接收的检测
0	未接收到扩展代码
1	接收到扩展代码
清零条件 (EXC = 0)	
- 检测到启动条件时 - 检测到停止条件时 - 由 LREL = 1(退出通信)清零 - 当 IICE 从 1 变为 0(操作停止)时 - 复位	
设置条件 (EXC = 1)	
当接收的地址数据的高四位为“0000”或“1111”时(在第八个时钟的上升沿设置)。	

注 当对 IICS 之外的位执行 1 位存储器操作指令, 此寄存器也会清零。因此, 当使用 ALD 位时, 读取其它位数据之前应先读取该位数据。

备注 LREL: IICA 控制寄存器 0 的第 6 位 (IICCTL0)
IICE: IICA 控制寄存器 0 的第 7 位 (IICCTL0)

图 12-7 IICA 状态寄存器格式 (IICS) (2/3)

COI	地址匹配的检测	
0	地址不匹配	
1	地址匹配	
清零条件 (COI = 0)		设置条件 (COI = 1)
• 检测到启动条件时 • 检测到停止条件时 • 由 LREL = 1(退出通信)清零 • 当 IICE 从 1 变为 0(操作停止)时 • 复位		• 当接收地址和本地地址(从设备地址寄存器 (SVA)) 匹配时(设置在第 8 个时钟的上升沿)

TRC	发送/接收状态的检测	
0	接收状态(非发送状态)。SDA0 线设为高阻抗状态。	
1	发送状态。允许将 SO0 锁存器中的内容输出到 SDA0 线(在第 1 个字节的第 9 个时钟的下降沿才有效)。	
清零条件 (TRC = 0)		设置条件 (TRC = 1)
<主设备模式和从设备模式> • 检测到停止条件时 • 由 LREL = 1(退出通信)清零 • 当 IICE 从 1 变为 0(操作停止)时 • 由 WREL = 1[※] 清零(等待取消) • ALD 从 0 变为 1(仲裁失败)时 • 复位 <主设备模式> • 当“1”输出至首字节的 LSB 时(传输方向指定位) <从设备模式> • 检测到启动条件时 • 当“0”输入至首字节的 LSB 时(传输方向指定位) <未用于通信时>		<主设备模式> • 产生启动条件时 • 当“0”输出至首字节的 LSB 时(传输方向指定位) <从设备模式> • 当“1”输入至首字节的 LSB 时(传输方向指定位)

注 当 IICA 状态寄存器 (IICS) 的第 3 位 (TRC) 为 1 时, 如果通过在第 9 个时钟设置 IICA 控制寄存器 0 (IICCTL0) 的第 5 位 (WREL) 为 1 来取消等待状态, 则 TRC 清零且 SDA0 线进入高阻抗状态。

备注 LREL: IICA 控制寄存器 0 的第 6 位 (IICCTL0)
IICE:IICE: 控制寄存器 0 的第 7 位 (IICCTL0)

图 12-7 IICA 状态寄存器格式 (IICS) (3/3)

ACKD	响应信号的检测 (ACK)	
0	未检测到响应信号	
1	已检测到响应信号	
清零条件 (ACKD = 0)		设置条件 (ACKD = 1)
• 检测到停止条件时 • 在下一个字节的第一个时钟的上升沿 • 由 LREL = 1(退出通信)清零 • 当 IICE 从 1 变为 0(操作停止)时 • 复位		• 在 SCL0 的第 9 个时钟的上升沿将 SDA0 线设为低电平之后

STD	启动条件的检测	
0	未检测到启动条件	
1	已检测到启动条件。这表示地址传输阶段有效。	
清零条件 (STD = 0)		设置条件 (STD = 1)
• 检测到停止条件时 • 在地址传输之后的下一个字节的第 1 个时钟的上升沿 • 由 LREL = 1(退出通信)清零 • 当 IICE 从 1 变为 0(操作停止)时 • 复位		• 检测到启动条件时

SPD	停止条件的检测	
0	未检测到停止条件	
1	已检测到停止条件。主设备通信结束，释放总线。	
清零条件 (SPD = 0)		设置条件 (SPD = 1)
• 在设置该位并检测到启动条件后的地址传输字节第 1 个时钟的上升沿 • 当 IICE 从 1 变为 0(操作停止)时 • 复位		• 检测到停止条件时

备注 LREL: IICA 控制寄存器 0 的第 6 位 (IICCTL0)
 IICE: IICA 控制寄存器 0 的第 7 位 (IICCTL0)

(4) IICA 标志寄存器 (IICF)

此寄存器用于设置 I²C 的操作模式，并指示 I²C 总线的状态。
IICF 可由 1 位或 8 位存储器操作指令设置。但是，STCF 和 IICBSY 位是只读的。
IICRSV 位用于允许/禁止通信预约功能。
STCEN 用于设置 IICBSY 的初始值。
仅当 I²C 操作禁止 (IICA 控制寄存器 0 (IICCTL0) 的第 7 位 (IICE) = 0) 时，IICRSV 和 STCEN 位才可写。当操作允许时，可以读取 IICF 寄存器。
复位后这个寄存器的值为 00H。

图 12-8. IICA 标志寄存器的格式 (IICF)

地址: FFF52H 复位后: 00H R/W^注

符号	<7>	<6>	5	4	3	2	<1>	<0>
IICF	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT 清零标志
0	产生启动条件
1	产生启动条件不成功: STT 标志清零
清零条件 (STCF = 0)	
设置条件 (STCF = 1)	
- 由 STT = 1 清零 - 当 IICE = 0 (操作停止) - 复位	当禁止通信预约时 (IICRSV = 1), 产生启动条件不成功且 STT 清零。

IICBSY	I ² C 总线状态标志
0	总线释放状态 (当 STCEN = 1 时的通信初始状态)
1	总线通信状态 (当 STCEN = 0 时的通信初始状态)
清零条件 (IICBSY = 0)	
设置条件 (IICBSY = 1)	
- 检测停止条件 - 当 IICE = 0 (操作停止) - 复位	- 检测启动条件 - 当 STCEN = 0 时 设置 IICE

STCEN	初始启动允许触发
0	允许操作后 (IICE = 1), 在检测到一个停止条件后允许产生一个启动条件。
1	允许操作后 (IICE = 1), 没有检测到一个停止条件后允许产生一个启动条件。
清零条件 (STCEN = 0)	
设置条件 (STCEN = 1)	
- 指令清零 - 启动条件的检测 - 复位	通过指令进行设置

IICRSV	通信预约功能禁止位
0	允许通信预约
1	禁止通信预约
清零条件 (IICRSV = 0)	
设置条件 (IICRSV = 1)	
- 指令清零 - 复位	通过指令进行设置

注 第 6 位和第 7 位是只读的。

- 注意事项**
1. 仅当操作停止 (IICE = 0) 时, 才可对 STCEN 进行写操作。
 2. 当 STCEN = 1 时, 不论总线实际处于何种状态, 总线释放状态 (IICBSY = 0) 都可以识别, 因此当第一个启动条件产生 (STT = 1) 时, 必须确保过程中没有第三方通信, 以防通信破坏。
 3. 仅当操作停止 (IICE = 0) 时, 才可对 IICRSV 进行写操作。

备注

STT: IICA 控制寄存器 0 的第 1 位 (IICCTL0)

IICE: IICA 控制寄存器 0 的第 7 位 (IICCTL0)

(5) IICA 控制寄存器 1 (IICCTL1)

此寄存器用于设置I²C的操作模式，并检测SCL0 和SDA0 引脚的状态。
IICCTL1 可由 1 位或 8 位的存储器操作指令设置。但是，CLD 和 DAD 位是只读的。
当 IICA 控制寄存器 0 (IICCTL0)的位 7 (IICE) 为 0 时，除了 WUP 位，设置(IICCTL1)。
复位后这个寄存器的值为 00H。

图 12-9. IICA 控制寄存器 1 (IICCTL1) 的格式 (1/2)

地址: F0231H 复位后: 00H R/W[※]

符号	<7>	6	<5>	<4>	<3>	<2>	1	0
IICCTL1	WUP	0	CLD	DAD	SMC	DFC	0	0

WUP	地址匹配唤醒的控制
0	停止模式下，地址匹配唤醒功能的停止操作。
1	停止模式下，地址匹配唤醒功能的允许操作。
在 WUP = 1 时切换到 STOP 模式，至少在设置(1) WUP 的 3 个时钟后才能执行 STOP 指令。 地址匹配或接收到扩展代码后，清零 WUP。通过清零 WUP，可进入后续通信。(必须释放等待并在 WUP 清零后写入传输数据。) 当地址匹配或已接收到一个扩展代码时的中断时序，在 WUP = 1 时与 WUP = 0 时(将发生一个时钟采样区别的延时)的中断时序相同。而且，当 WUP = 1，即使 SPIE 位设为 1 也不产生停止条件中断。当通过从串行接口 IICA 产生的中断之外的源设置 WUP = 0 时，不能执行主设备操作，直到检测到后续启动条件或停止条件。通过设置 STT 位为 1，不输出一个启动条件，且不等待后续启动条件或停止条件的检测。	
清零条件 (WUP = 0)	设置条件 (WUP = 1)
• 通过指令清除(在地址匹配或接收到扩展代码后)	• 通过指令设置(当MSTS、EXC和COI为“0”，且STD也为“0”(不进入通信)) [※]

注 在下列期间，必须检查 IICS 状态且必须设置 WUP。

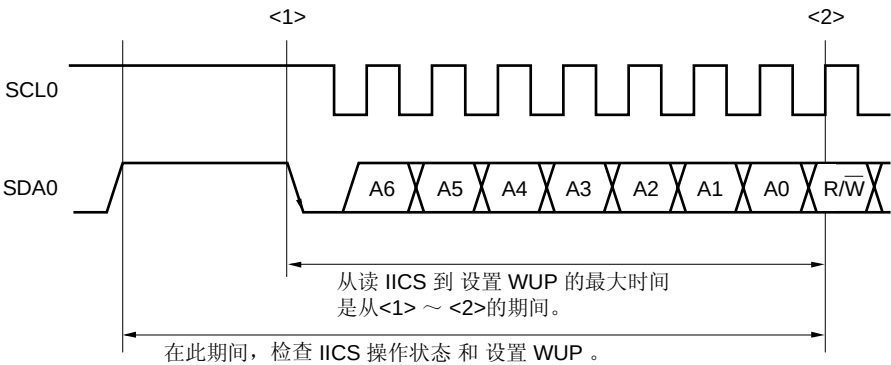


图 12-9. IICA 控制寄存器 1 (IICCTL1) 的格式 (2/2)

CLD	SCL0 引脚电平的检测 (仅当 IICE = 1 时有效)	
0	SCL0 引脚检测为低电平。	
1	SCL0 引脚检测为高电平。	
清零条件 (CLD = 0)		设置条件 (CLD = 1)
• SCL0 引脚为低电平时 • 当 IICE = 0(操作停止) • 复位		• SCL0 引脚为高电平时

DAD	SDA0 引脚电平的检测 (仅当 IICE = 1 时有效)	
0	SDA0 引脚检测为低电平。	
1	SDA0 引脚检测为高电平。	
清零条件 (DAD = 0)		设置条件 (DAD = 1)
• SDA0 引脚为低电平时 • 当 IICE = 0(操作停止) • 复位		• SDA0 引脚为高电平时

SMC	操作模式切换	
0	标准模式	
1	高速模式	

DFC	数字滤波器操作控制
0	关闭数字滤波器
1	打开数字滤波器
数字滤波器只能用于高速模式。 在高速模式下，无论 DFC 位是设置为 1 还是清零，传输时钟都不会发生变化。 数字滤波器用于高速模式下的噪声消除。	

注 第 4 位和第 5 位是只读的。

备注 IICE: IICA 控制寄存器 0 的第 7 位 (IICCTL0)

(6) IICA 低电平宽度设置寄存器 (IICWL)

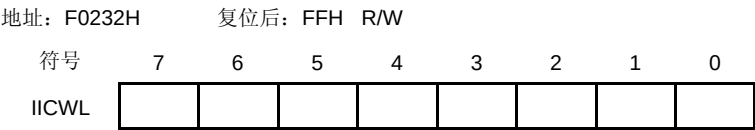
此寄存器用于设置 SCL0 引脚信号的低电平宽度，该信号由主模式下串行接口 IICA 输出。

IICWL 可以以 8 位存储器操作指令设置。

当 IICA 控制寄存器 0 (IICCTL0) 的位 7 (IICE) 设为 0 时，设置 IICWL。

复位信号的产生将此寄存器设置为 FFH。

图 12-10. IICA 低电平宽度设置寄存器(IICWL)格式



(7) IICA 高电平宽度设置寄存器 (IICWH)

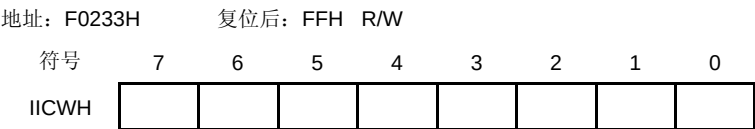
此寄存器用于设置 SCL0 引脚信号的高电平宽度，该信号由主模式下串行接口 IICA 输出。

IICWH 可以以 8 位存储器操作指令设置。

当 IICA 控制寄存器 0 (IICCTL0) 的第 7 位 (IICE) 设为 0 时，设置 IICWH。

复位信号的产生将此寄存器设置为 FFH。

图 12-11. IICA 高电平宽度设置寄存器(IICWH)格式



(8) 端口模式寄存器 6 (PM6)

此寄存器可以以 1 位单元设置端口 6 的输入/输出操作模式。

当 P60/SCL0 引脚用于时钟 I/O 同时将 P61/SDA0 引脚用于串行数据 I/O 时，将 PM60 和 PM61 清零，且将 P60 和 P61 的输出锁存器清零。

在设置输出模式之前，应将 IICE (IICA 控制寄存器 0 (IICCTL0) 的第 7 位) 设为 1，因为当 IICE 为 0 时，P60/SCL0 和 P61/SDA0 引脚输出一个低电平 (固定)。

可由 1 位或 8 位存储器操作指令设置 PM6。

复位信号的产生将此寄存器设置为 FFH。

图 12-12. 端口模式寄存器 6 (PM6)的格式

地址: FFF26H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60

PM6n	P6n引脚模式选择 (n = 0, 7)
0	输出模式 (输出缓冲开启)
1	输入模式 (输出缓冲停止)

12.4 I²C 总线模式功能

12.4.1 引脚配置

串行时钟引脚 (SCL0) 和串行数据总线引脚 (SDA0) 配置如下。

(1) SCL0..... 此引脚用于串行时钟的输入和输出。

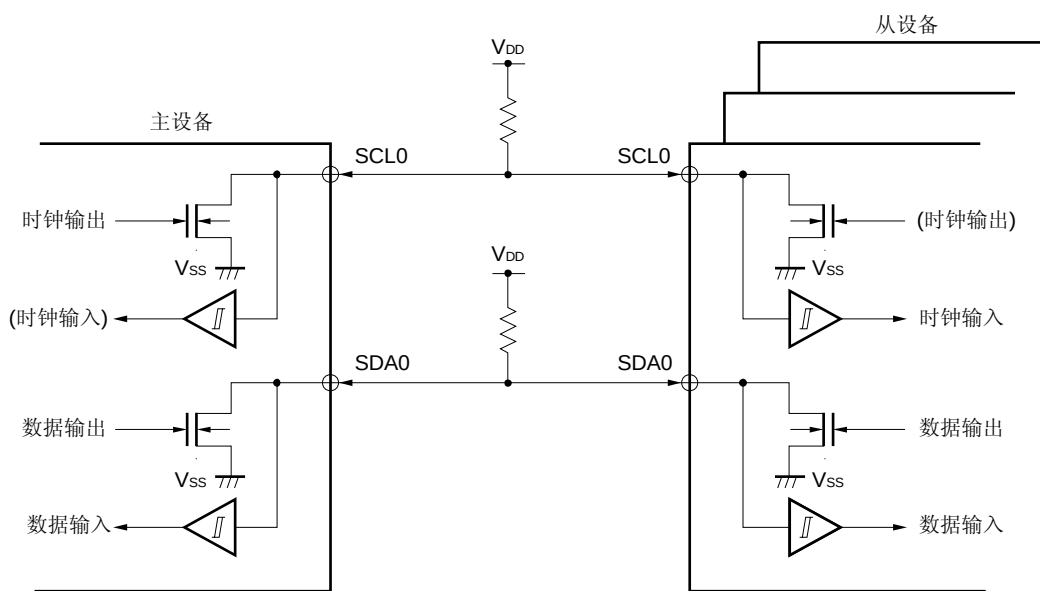
此引脚是主设备和从设备的 N 沟道漏极开路输出。该输入是施密特 (Schmitt) 输入。

(2) SDA0 此引脚用于串行数据的输入和输出。

此引脚是主设备和从设备的 N 沟道漏极开路输出。该输入是施密特 (Schmitt) 输入。

由于串行时钟线和串行数据总线的输出都是 N 沟道漏极开路输出，因此需要一个外部上拉电阻。

图 12-13. 引脚配置框图



12.4.2 通过 IICWL 和 IICWH 寄存器设置传输时钟

(1) 在主设备端设置传输时钟

$$\text{传输时钟} = \frac{\text{IICWL} + \text{IICWH} + f_{\text{CLK}}(t_{\text{R}} + t_{\text{F}})}{f_{\text{CLK}}}$$

此时，IICWL 和 IICWH 的最佳设置值如下。
(设置值的小数部分四舍五入。)

- 当高速模式时

$$\begin{aligned}\text{IICWL} &= \frac{0.52}{\text{传输时钟}} \times f_{\text{CLK}} \\ \text{IICWH} &= \left(\frac{0.48}{\text{传输时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{CLK}}\end{aligned}$$

- 当普通模式时

$$\begin{aligned}\text{IICWL} &= \frac{0.48}{\text{传输时钟}} \times f_{\text{CLK}} \\ \text{IICWH} &= \left(\frac{0.53}{\text{传输时钟}} - t_{\text{R}} - t_{\text{F}} \right) \times f_{\text{CLK}}\end{aligned}$$

(2) 在从设备端设置 IICWL 和 IICWH

(设置值的小数部分截去。)

- 当高速模式时

$$\begin{aligned}\text{IICWL} &= 1.3 \mu\text{S} \times f_{\text{CLK}} \\ \text{IICWH} &= (1.2 \mu\text{S} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{CLK}}\end{aligned}$$

- 当普通模式时

$$\begin{aligned}\text{IICWL} &= 4.7 \mu\text{S} \times f_{\text{CLK}} \\ \text{IICWH} &= (5.3 \mu\text{S} - t_{\text{R}} - t_{\text{F}}) \times f_{\text{CLK}}\end{aligned}$$

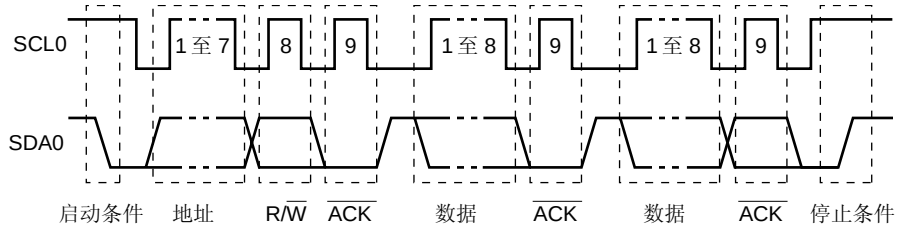
备注

IICWL: IICA 低电平宽度设置寄存器
IICWH: IICA 高电平宽度设置寄存器
 t_{F} : SDA0 和 SCL0 信号下降时间
 t_{R} : SDA0 和 SCL0 信号上升时间
 f_{CLK} : CPU/外围硬件时钟频率

12.5 I²C 总线定义和控制方法

以下部分描述了 I²C 总线的串行数据通信格式和 I²C 总线所使用的信号。图 12-14 显示了 I²C 总线的串行数据总线上用于“开始条件”、“地址”、“数据”和“停止条件”的传输时序。

图 12-14. I²C 总线串行数据传输时序



主设备产生启动条件、从设备地址和停止条件。

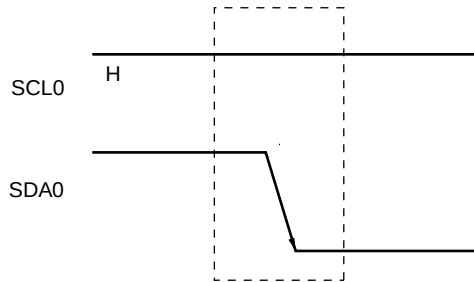
响应信号 ($\overline{\text{ACK}}$) 可以由主设备或者从设备产生(通常，由接收 8 位数据的设备输出)。

主设备持续输出串行时钟 (SCL0)。但在从设备中，可以延长 SCL0 的低电平阶段同时可以插入等待时间。

12.5.1 启动条件

SCL0 引脚处于高电平且 SDA0 引脚从高电平变为低电平时，满足启动条件。当启动串行发送时，SCL0 引脚和 SDA0 引脚的启动条件是由主设备产生并发给从设备的信号。当使用从设备时，可以检测到启动条件。

图 12-15. 启动条件



检测到一个停止条件 (SPD: IICA 状态寄存器 0 (IICS) 的第 0 位=1) 后，IICA 控制寄存器 0 (IICCTL0) 的第 1 位 (STT) 设置 (为 1) 时，输出一个启动条件。当检测到一个启动条件时，IICS 的位 1 (STD) 设为 (1)。

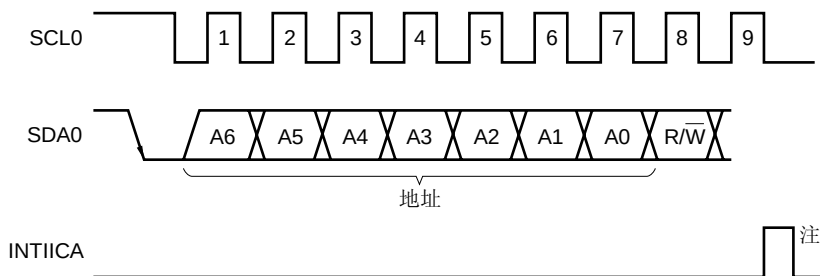
12.5.2 地址

地址由启动条件之后的 7 位数据指定。

一个地址是一个输出的 7 位数据段，用于选择一个通过总线与主设备相连的从设备。因此，通过总线相连的每个从设备必须有一个唯一的地址。

从设备包括具有以下功能的硬件：检测启动条件，检查 7 位地址数据是否与存储在从设备地址寄存器 (SVA) 中的数据匹配。如果地址数据和 SVA 的值匹配，则此从设备将选中与主设备通信，直到主设备产生一个启动条件或停止条件。

图 12-16. 地址



注 如果在从设备操作期间，接收到的是数据而不是本地地址或扩展代码时，则 INTIICA 不会发出。

当包括从设备地址和传输方向 (如 **12.5.3 传输方向指示** 中所描述的) 的总共 8 位信息一起写入 IICA 移位寄存器 (IICA) 时，地址输出。接收到的地址写入 IICA。

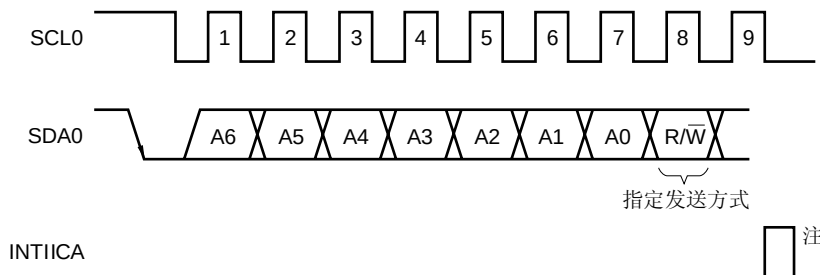
将从设备地址分配给 IICA 的高 7 位。

12.5.3 传输方向指示

除了 7 位地址数据之外，主设备还发送 1 位数据用于指示传输方向。

当传输方向指定位的值为“0”时，表明主设备正发送数据至从设备。当传输方向指定位的值为“1”时，表明主设备正接收来自从设备的数据。

图 12-17. 传输方向指示



注 如果在从设备操作期间，接收到的是数据而不是本地地址或扩展代码时，则 INTIICA 不会发出。

12.5.4 响应信号 ($\overline{ACK}$)

$\overline{ACK}$ 信号用于检测处于发送方和接收方的串行数据的状态。

当接收方收到 8 位数据后将返回一个 $\overline{ACK}$ 信号。

发送方通常在发送了 8 位数据后接收到 $\overline{ACK}$ 信号。当 $\overline{ACK}$ 信号从接收方返回时，认为已正确接收并继续之后的处理。是否检测到 $\overline{ACK}$ 信号，可以通过 IICA 状态寄存器 (IICS) 的第 2 位 (ACKD) 进行检查。

主设备接收到最后一个数据项时，不再返回 $\overline{ACK}$ 信号而是产生一个停止条件。如果从设备在接收到数据后未返回 $\overline{ACK}$ 信号，主设备将输出一个停止条件或重启条件并且停止发送。如果未返回 $\overline{ACK}$ 信号，可能是如下原因。

- <1> 接收操作没有正常进行。
- <2> 接收到最后一个数据项。
- <3> 由地址指示的接收方不存在。

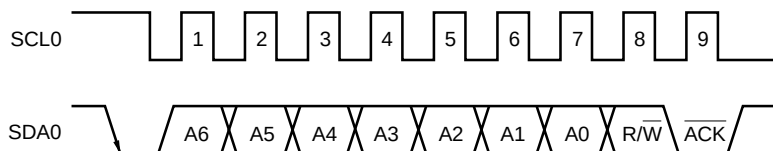
要产生 $\overline{ACK}$ 信号，接收方必须在第 9 个时钟时使 SDA0 线变为低电平(表示正常接收)。

通过设置 IICA 控制寄存器 0 (IICCTL0) 的第 2 位 (ACKE) 为 1，自动产生 $\overline{ACK}$ 信号。IICS 寄存器的第 3 位 (TRC) 可第 8 位的数据(紧接着 7 位地址信息之后)设置。通常，设置 ACKE=1 用于接收 (TRC = 0)。

如果一个从设备在接收期间 (TRC = 0)，不再收到任何数据或者不再需要下一个数据项，则从设备必须通过将 ACKE 清零来通知主设备，这样它就不会再收到任何数据。

接收期间 (TRC = 0)，当主设备不需要下一个数据项时，必须将 ACKE 清零，这样就不会产生 $\overline{ACK}$ 。在这种方式下，主设备通知作为发送方的从设备，不再需要其它数据(发送过程将停止)。

图 12-18. $\overline{ACK}$



接收到本地地址时，无论 ACKE 的值是多少，都将自动产生 $\overline{ACK}$ 信号。当接收到一个非本地地址时，则不产生 $\overline{ACK}$ 信号 (NACK)。

接收到一个扩展代码时，如果 ACKE 预先设为 1，则将产生 $\overline{ACK}$ 信号。

接收到数据时如何产生 $\overline{ACK}$ 信号，将根据等待时序的设置而变化。

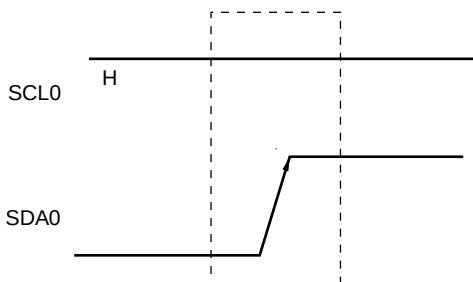
- 当选择 8 时钟等待状态时 (IICCTL0 寄存器的第 3 位(WTIM)= 0):
释放等待状态之前，通过将 ACKE 设为 1，在 SCL0 引脚第 8 个时钟的下降沿产生 $\overline{ACK}$ 信号。
- 当选择 9 时钟等待状态时 (IICCTL0 寄存器的第 3 位(WTIM)= 1):
通过预先设置 ACKE 为 1，产生 $\overline{ACK}$ 信号。

12.5.5 停止条件

SCL0 引脚处于高电平时，将 SDA0 引脚从低电平变为高电平可以产生一个停止条件。

停止条件是一个信号，它在串行发送操作结束时由主设备产生并发送给从设备。当作为从设备使用时，可以检测到停止条件。

图 12-19. 停止条件



当 IICA 控制寄存器 0 (IICCTL0) 的第 0 位 (SPT) 设为 1 时，会产生一个停止条件。当检测到这个停止条件时，IICA 状态寄存器 0 (IICS) 的第 0 位 (SPD) 设为 1，且当 IICCTL0 的第 4 位 (SPIE) 设为 1 时，产生 INTIICA。

12.5.6 等待

等待状态用来通知通信另一方，设备(主设备或从设备)正准备发送或者接收数据(即处于等待状态)。

设置 SCL0 引脚为低电平，通知通信另一方设备已进入等待状态。当主设备和从设备的等待状态取消时，下一个数据发送才能开始。

图 12-20. 等待(1/2)

(1) 当主设备有 9 个时钟等待而从设备有 8 个时钟等待时(主设备发送，从设备接收，同时 ACKE = 1)

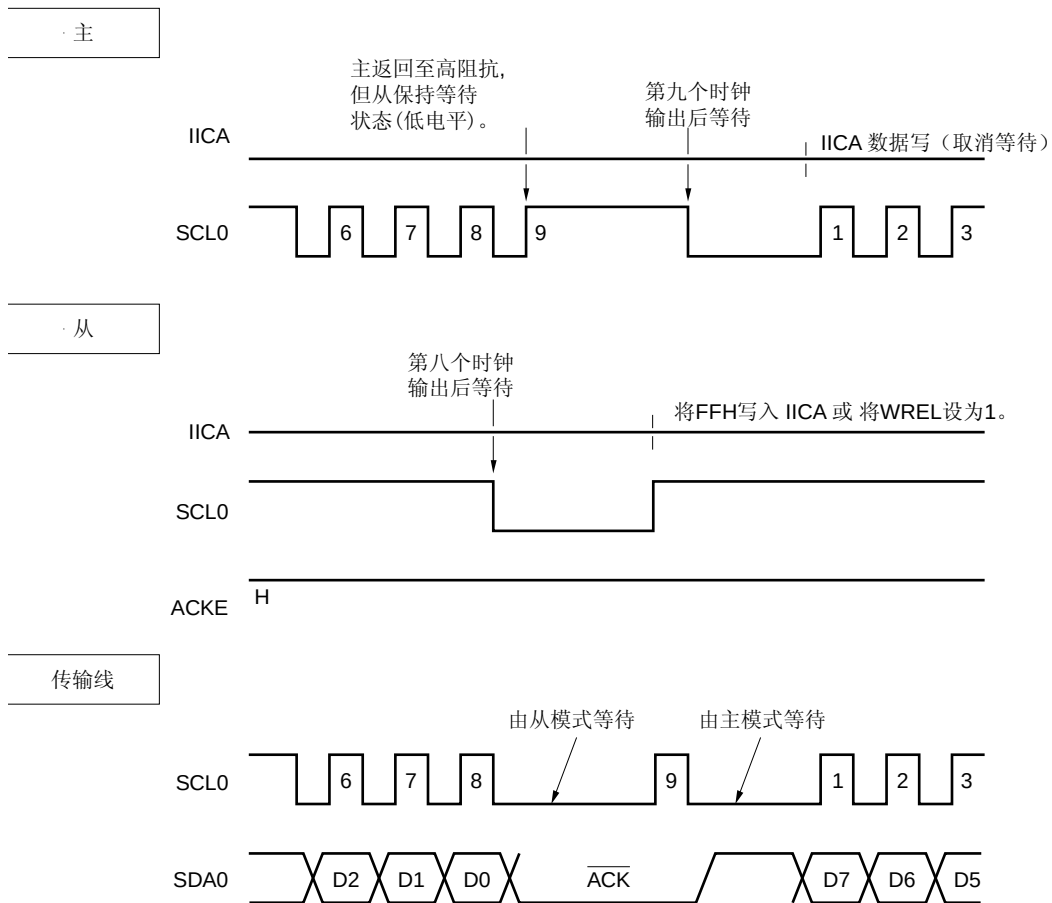
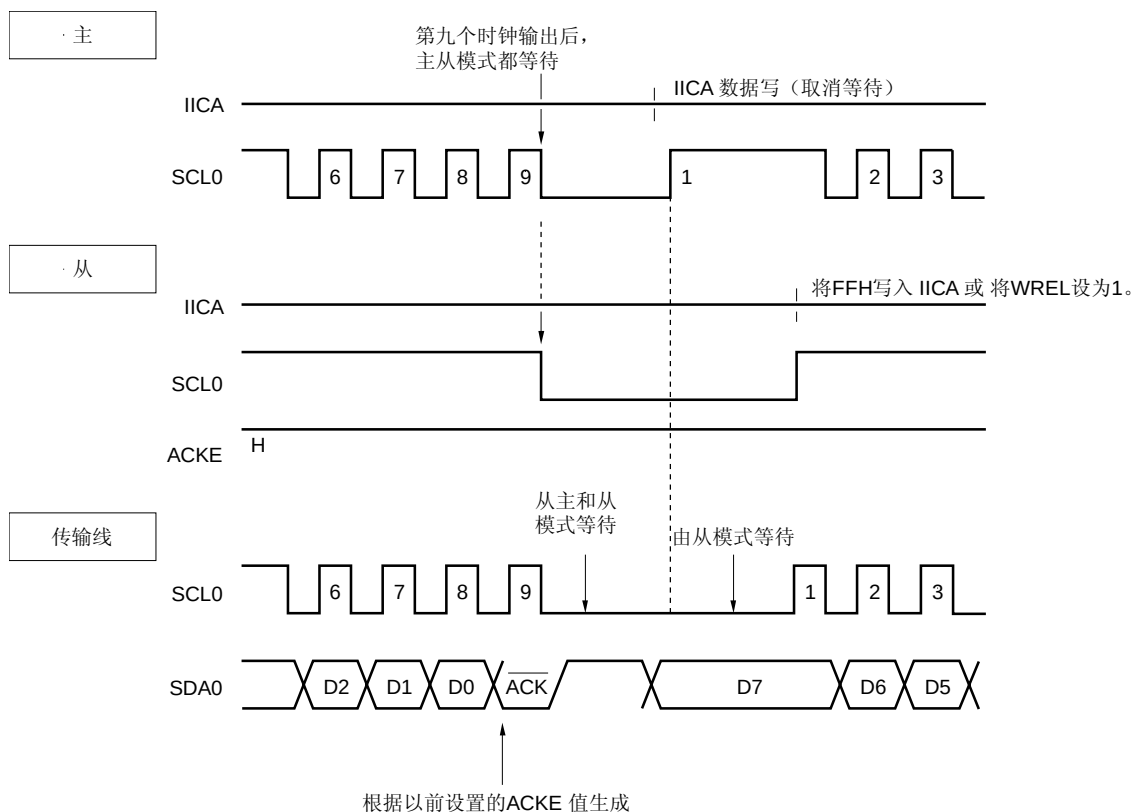


图 12-20. 等待(2/2)

(2) 当主设备和从设备都有一个 9 个时钟等待时
(主设备发送，从设备接收，同时 ACKE = 1)



备注 ACKE: IICA 控制寄存器 0 的第 2 位 (IICCTL0)
WREL: IICA 控制寄存器 0 的第 5 位 (IICCTL0)

根据 IICA 控制寄存器 0 (IICCTL0) 的第 3 位 (WTIM) 的设置，自动产生一个等待。

通常，当 IICCTL0 的第 5 位 (WREL) 设置为 1 时或者在将 FFH 写入 IICA 移位寄存器 (IICA) 时，接收方取消等待状态，而当数据写入 IICA 时，发送方取消等待状态。

主设备也能通过以下任何一种方法取消等待状态。

- 通过把 IICCTL0 的第 1 位 (STT) 设为 1
- 通过把 IICCTL0 的第 0 位 (SPT) 设为 1

12.5.7 取消等待

I²C 通常通过如下处理方式取消一个等待状态。

- 把数据写入 IICA 移位寄存器 (IICA)
- 设置 IICA 控制寄存器 0 (IICCTL0) 的第 5 位 (WREL)(取消等待)
- 设置 IICCTL0 寄存器的第 1 位 (STT)(产生启动条件)[※]
- 设置 IICCTL0 寄存器的第 0 位 (SPT)(产生停止条件)[※]

注 仅在主模式下

当执行以上取消等待处理过程时，I²C 取消等待状态且恢复通信。

要取消一个等待状态并发送数据(包括地址)，则向 IICA 写数据。

要在取消一个等待状态之后接收数据，或完成数据发送操作，则设置 IICA 控制寄存器 0 (IICCTL0) 的第 5 位(WREL)为 1。

要在取消一个等待状态之后产生一个重启条件，则设置 IICCTL0 的第 1 位 (STT) 为 1。

要在取消一个等待状态之后产生一个停止条件，则设置 IICCTL0 的第 0 位 (SPT) 为 1。

对于一个等待状态只能执行一次取消操作。

例如，如果通过设置 WREL 为 1 在取消一个等待状态之后将数据写入 IICA，则可能会将一个不正确的值输出到 SDA0，这是因为改变 SDA0 线的时序和 IICA 写操作时序相冲突。

除此以外，如果通信已中止时 IICE 清零，则通信停止，这样就可以取消等待状态。

如果 I²C 总线由于噪声产生死锁，通过设置 IICCTL0 的位 6(LREL)使得设备退出通信，因此等待状态取消。

注意事项 如果当 IICCTL1 寄存器的位 7(WUP)=1 时上述的等待取消过程执行，则等待不会取消。

12.5.8 中断请求 (INTIICA)产生时序和等待控制

IICA 控制寄存器 0 (IICCTL0) 的第 3 位 (WTIM) 的设置决定了 INTIICA 产生的时序和相应的等待控制，如表 12-2 所示。

表 12-2. INTIICA 产生时序和等待控制

WTIM	从设备操作期间			主设备操作期间		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9 ^{#1,2}	8 ^{#2}	8 ^{#2}	9	8	8
1	9 ^{#1,2}	9 ^{#2}	9 ^{#2}	9	9	9

- 注 1.** 仅当与从设备地址寄存器 (SVA) 设置的地址匹配时，在第 9 个时钟的下降沿产生从设备的 INTIICA 信号和等待周期。
- 此时，不论 IICCTL0 的第 2 位 (ACKE) 设置为何值，都将产生 $\overline{\text{ACK}}$ 信号。对于接收到一个扩展代码的一个从设备，在第 8 个时钟的下降沿将产生 INTIICA。
- 但是，如果重启之后地址不匹配，将在第 9 个时钟的下降沿产生 INTIICA，但不会产生等待。
- 2.** 如果接收地址与从设备地址寄存器 (SVA) 的内容不匹配，同时也未接收到扩展代码，则 INTIICA 和等待周期都不会产生。

备注 表中的数字是指串行时钟的时钟信号个数。 中断请求和等待控制和这些时钟信号的下降沿同步。

(1) 地址传输/接收期间

- 从设备操作：中断和等待时序取决于以上注 1、2 中所描述的条件，与 WTIM 位无关。
- 主设备操作：中断和等待时序在第 9 个时钟的下降沿发生，与 WTIM 位无关。

(2) 数据接收期间

- 主/从设备操作：中断和等待时序取决于 WTIM 位。

(3) 数据发送期间

- 主/从设备操作：中断和等待时序取决于 WTIM 位。

(4) 等待取消的方法

四种取消等待的方法如下所示。

- 把数据写入 IICA 移位寄存器 (IICA)
- 设置 IICA 控制寄存器 0 (IICCTL0) 的第 5 位 (WREL)(取消等待)
- 设置 IICCTL0 寄存器的第 1 位 (STT)(产生启动条件)[#]
- 设置 IICCTL0 寄存器的第 0 位 (SPT)(产生停止条件)[#]

注 仅在主模式下

当选择 8 个时钟等待时 (WTIM = 0)，必须先决定是否产生 $\overline{\text{ACK}}$ 信号，再考虑等待取消。

(5) 停止条件的检测

当检测到一个停止条件时 (仅当 SPIE = 1)，INTIICA 将会产生。

12.5.9 地址匹配的检测方法

在I²C总线模式下，主设备可以通过发送相应的从设备地址来选择指定的从设备。

可以通过硬件自动检测到地址匹配。当设置从地址寄存器(SVA)的地址与主设备发出的从地址匹配时，或当接收到一个扩展码时，产生一个中断请求(INTIICA)。

12.5.10 误差检测

I²C总线模式下，在数据发送期间，串行数据总线 (SDA0) 的状态发送设备的IICA移位寄存器 (IICA) 捕获，因此发送前的IICA数据可以和已发送的IICA数据相比较，以此来实现对发送错误的检测。当比较值不匹配时，可以判断存在一个发送错误。

12.5.11 扩展码

- (1) 当接收地址的高四位为“0000”或“1111”时，外部代码接收标志(EXC)设为 1 来接收外部代码并在第八个时钟的下降沿发出中断请求(INTIICA)。存放在从设备地址寄存器 (SVA) 中的本地地址不受影响。
- (2) 如果“11110xx0”设置到 SVA，通过 10 位地址传输，主设备发送“11110xx0”结果如下。注意在第 8 个时钟的下降沿产生 INTIICA。

- 数据的高 4 位相等：EXC = 1
- 7 位数据相等：COI = 1

备注 EXC: IICA 状态寄存器(IICF)的位 5
COI: IICA 状态寄存器(IICF)的位 4

- (3) 由于中断请求发生之后的处理过程随扩展代码之后的数据而有所不同，因此这样的处理过程由软件来执行。如果当一个从设备正在操作时接收到扩展代码，那么即使它的地址不匹配，该从设备也将参与通信。例如：接收到扩展码之后，如果你不希望操作目标设备作为从设备，设置 IICA 控制寄存器 0(IICCTL0)的位 6(LREL)为 1，用来设置下一次通信操作的待机状态。

表 12-3. 主扩展代码的位定义

从设备地址	R/W 位	描述
0 0 0 0 0 0 0	0	通用调用地址
1 1 1 1 0 x x	0	10 位从地址指定(在地址验证期间)
1 1 1 1 0 x x	1	10 位从地址指定(在地址匹配之后，当读取指令执行时)

备注 除上述以外的相关扩展指令的详细内容参见由NXP半导体发行的 I²C 总线说明。

12.5.12 仲裁

当多个主设备同时产生启动条件时 (在 STD 设为 1 之前, 将 STT 设为 1), 执行主设备间的通信, 调整时钟数目直至数据改变。这种操作称为仲裁。

当这些主设备之一仲裁失败时, 可通过仲裁失败产生的时序, 设置 IICA 状态寄存器 (IICS) 中的仲裁失败标志 (ALD) 为 1。同时 SCL0 和 SDA0 线都变为高阻抗, 从而释放总线。

基于下一个中断请求的时序 (第 8 或第 9 个时钟, 检测到一个停止条件时, 等等) 以及由软件设置的 ALD=1, 检测出仲裁失败。

有关中断请求时序的详细信息, 参见 12.5.8 中断请求 (INTIICA) 产生时序和等待控制。

备注 STD: IICA 状态寄存器(IICF)的位 1
 STT: IICA 控制寄存器 0 的第 1 位 (IICCTL0)

图 12-21. 仲裁时序举例

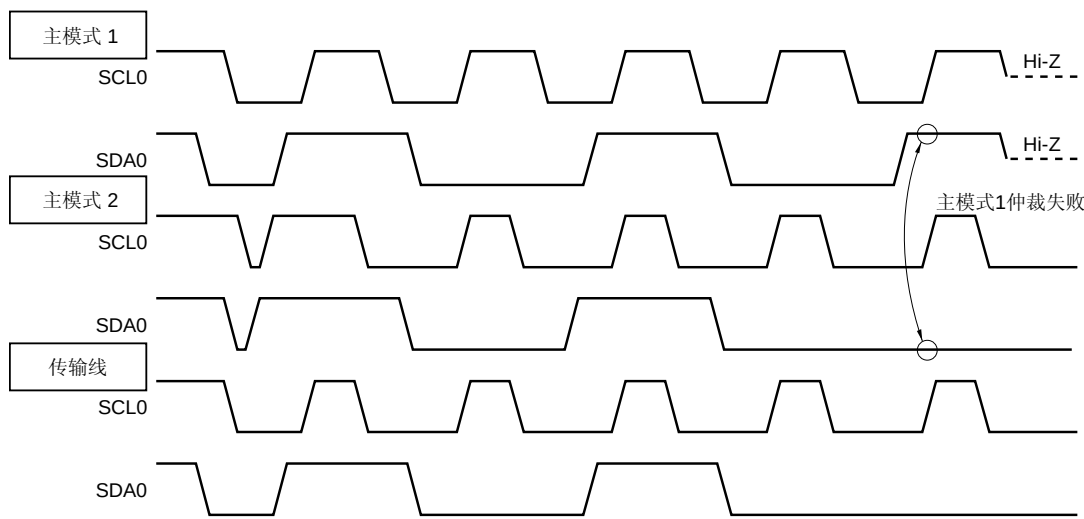


表 12-4. 仲裁期间的状态和中断请求产生时序

仲裁期间的状态	中断请求产生时序
地址传输期间	字节发送后第 8 或第 9 个时钟的下降沿 ^{註1}
地址传输后读/写数据	
扩展代码发送期间	
扩展代码发送后读/写数据	
数据发送期间	
数据发送之后的 ACK 信号发送期间	
数据发送期间检测到重启条件时	
数据发送期间检测到停止条件时	停止条件产生时 (当SPIE = 1) ^{註2}
准备产生重启条件时数据为低电平	字节发送后第 8 或第 9 个时钟的下降沿 ^{註1}
准备产生重启条件时检测到停止条件	停止条件产生时 (当SPIE = 1) ^{註2}
准备产生停止条件时数据为低电平	字节发送后第 8 或第 9 个时钟的下降沿 ^{註1}
准备产生重启条件时 SCL0 为低电平	

注 1. 当 WTIM (IICA 控制寄存器 0 (IICCTL0) 的第 3 位) = 1 时,将在第 9 个时钟的下降沿产生一个中断请求。当 WTIM = 0 且已接收到扩展代码的从设备地址时, 将在第 8 个时钟的下降沿产生一个中断请求。

 2. 如果有可能发生仲裁, 则对主设备操作设置 SPIE = 1。

备注 SPIE: IICA 控制寄存器 0 的第 4 位 (IICCTL0)

12.5.13 唤醒功能

I²C总线从功能上当接收到一个本地地址或扩展码时，产生中断请求信号 (INTIICA)。

此功能通过在地址不匹配时防止产生不必要的 INTIICA 信号，从而使处理更加有效。

当检测到一个启动条件时，设置唤醒待机模式。仲裁失败可能将主设备(已产生启动条件)转变为从设备，如果因此而发送地址时，则该唤醒待机模式有效。

然而，当检测到一个停止条件时，设置 IICA 控制寄存器 0(IICCTL0)的位 4(SPIE)，与唤醒功能无关，决定中断请求是否使能或禁止。

要在停止模式下使用唤醒功能，设置 WUP 为 1。无论操作时钟如何，都可以接收这些地址。在接收到一个本地地址及扩展代码时，也产生一个中断请求信号(INTIICA)。此中断产生后，通过一个指令将 WUP 位清零，回到正常操作。

当设置 WUP 位及地址匹配情况下将 WUP 位清零的流程如下。

图 12-22. 当设置 WUP = 1 时的流程

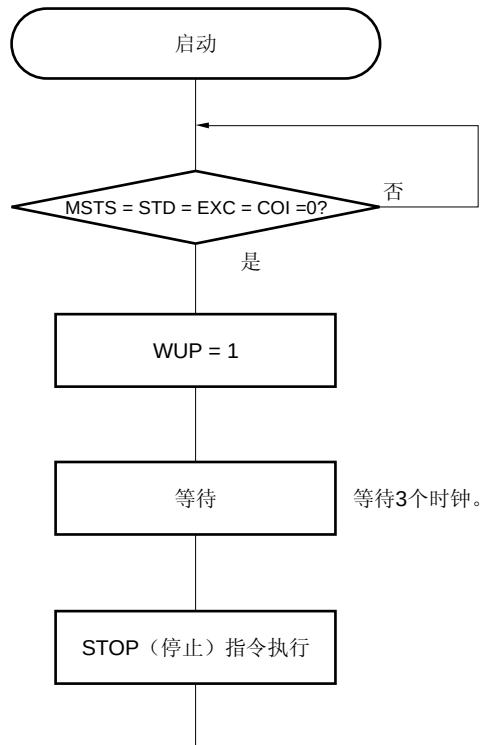
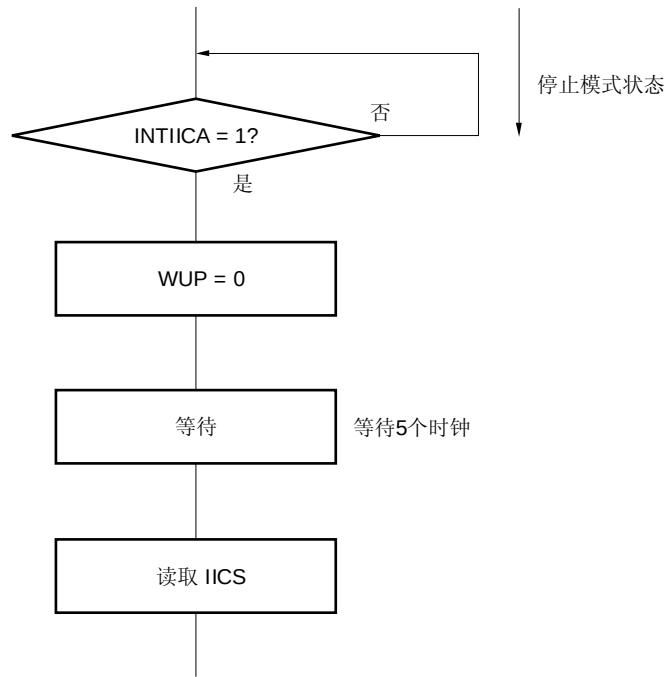


图 12-23. 当地址匹配时设置 WUP = 0 时的流程(包括扩展代码接收)

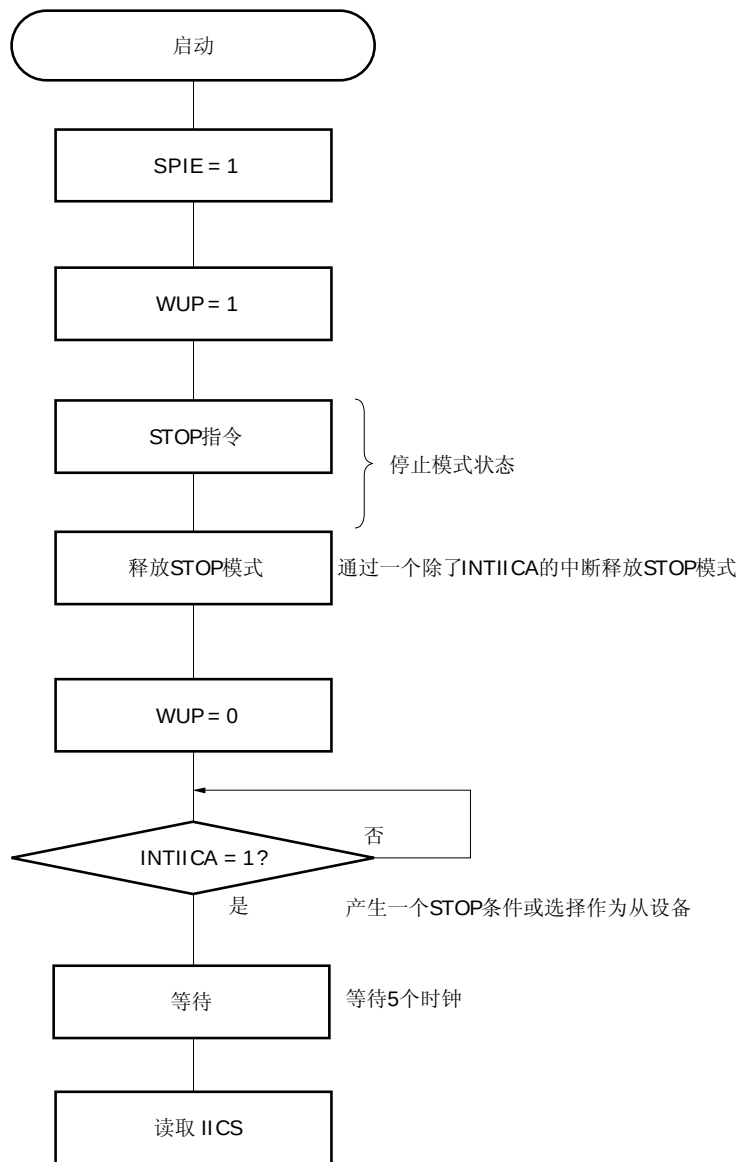


在检查串口IICA的操作状态后，对应于操作执行，过程执行

使用下列流程执行释放 STOP (停止) 模式的处理，并非从串行接口 IICA 产生中断请求 (INTIICA)。

- 主设备操作：流程如图 12-24 所示。
- 从设备操作：流程如图 12-23 所示。

图 12-24. 当释放 STOP 模式之后作为主设备操作时(并非通过 INTIICA)



在检查串口 IICA 的操作状态后，对应于操作执行，过程执行

12.5.14 通信预约

(1) 当通信预约功能允许时 (IICA 标志寄存器 (IICF) 的第 0 位 (IICRSV) = 0)

当没有参与通信时，启动主设备通信，当总线释放时，预约通信用来使能启动条件的发送。下面两个状态包含在设备不参与通信的状态中。

- 当仲裁结果不是主/从设备操作时
- 当接收到扩展码和禁止从操作时($\overline{ACK}$ 没有返回，通过设置 IICA 控制寄存器 0(IICCTL0)的位 6(LREL)为 1 并退出通信来释放总线)。

如果 IICCTL0 的位 1(STT) 设置为 1，当设备不参与通信时，总线释放后(当检测到一个停止条件时)，一个启动条件自动产生并设置等待状态。

如果在 IICCTL0 的第 4 位 (SPIE) 设为 1 之后，将一个地址写入 IICA 移位寄存器 (IICA)，并且通过总线释放(检测停止状态) 产生中断请求信号 (INTIICA) 来检测该地址，则设备将以主设备方式自动开始通信。在检测到停止条件前写入 IICA 的数据是无效的。

当 STT 已设为 1 时，操作模式(作为启动条件或作为通信预约)取决于总线的状态。

- 如果总线已释放 产生一个起始条件
- 如果总线未释放(待机状态) 通信预约

STT 设为 1 且经历了等待时间后，通过使用 MSTS(IICA 状态寄存器 (IICS) 的第 7 位)，来检查通信预约是否操作。

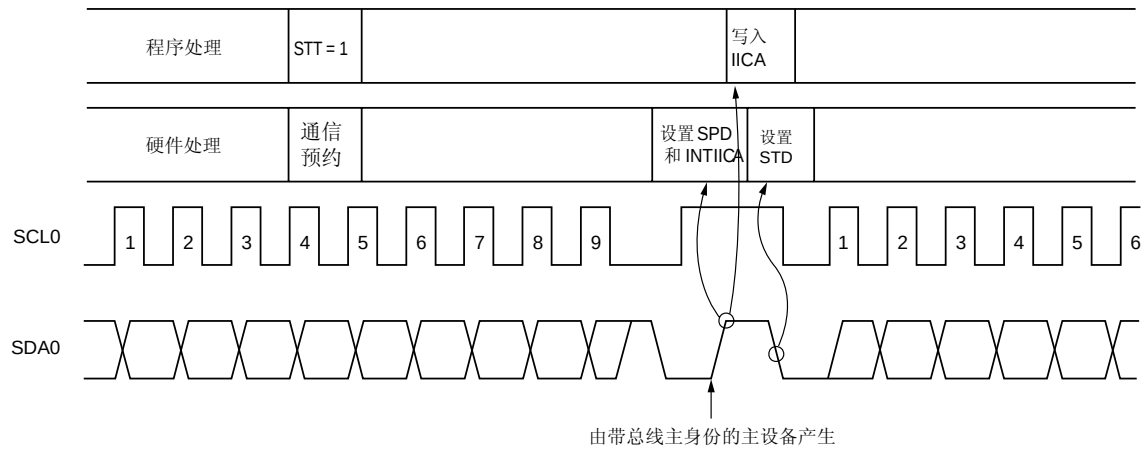
使用软件来保证等待时间，此时间可通过下面的公式计算。

等待时间，设置 STT = 1 来检查 MSTS 标志：
$$(IICWL \text{ 设置值} + IICWH \text{ 设置值} + 4) + t_F \times 2 \times f_{CLK} \text{ [时钟]}$$

备注	IICWL:	IICA 低电平宽度设置寄存器
	IICWH:	IICA 高电平宽度设置寄存器
	t _F :	SDA0 和 SCL0 信号下降时间
	f _{CLK} :	CPU/外围硬件时钟频率

图 12-25 显示通信预约时序。

图 12-25. 通信预约时序



备注 IICA: IICA 移位寄存器
STT: IICA 控制寄存器 0 的第 1 位 (IICCTL0)
STD: IICA 状态寄存器(IICS)的位 1
SPD: IICA 状态寄存器(IICS)的位 0

通过图 12-26 所示的时序接受通信预约 IICA 状态寄存器 (IICS) 的第 1 位 (STD) 设为 1 之后，通过在检测到一个停止条件之前设置 IICA 控制寄存器 0 (IICCTL0) 的第 1 位 (STT) 为 1，可以进行通信预约。

图 12-26. 接受通信预约的时序

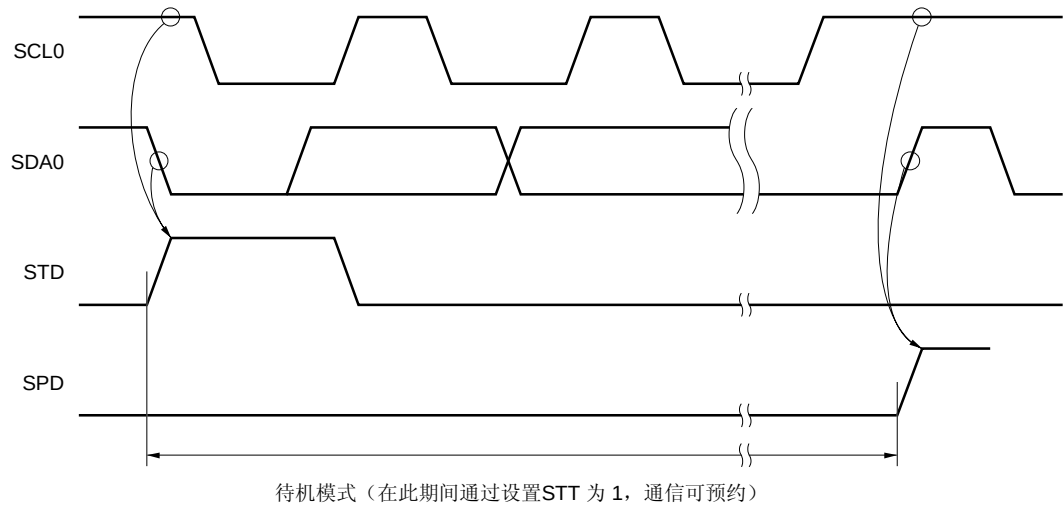
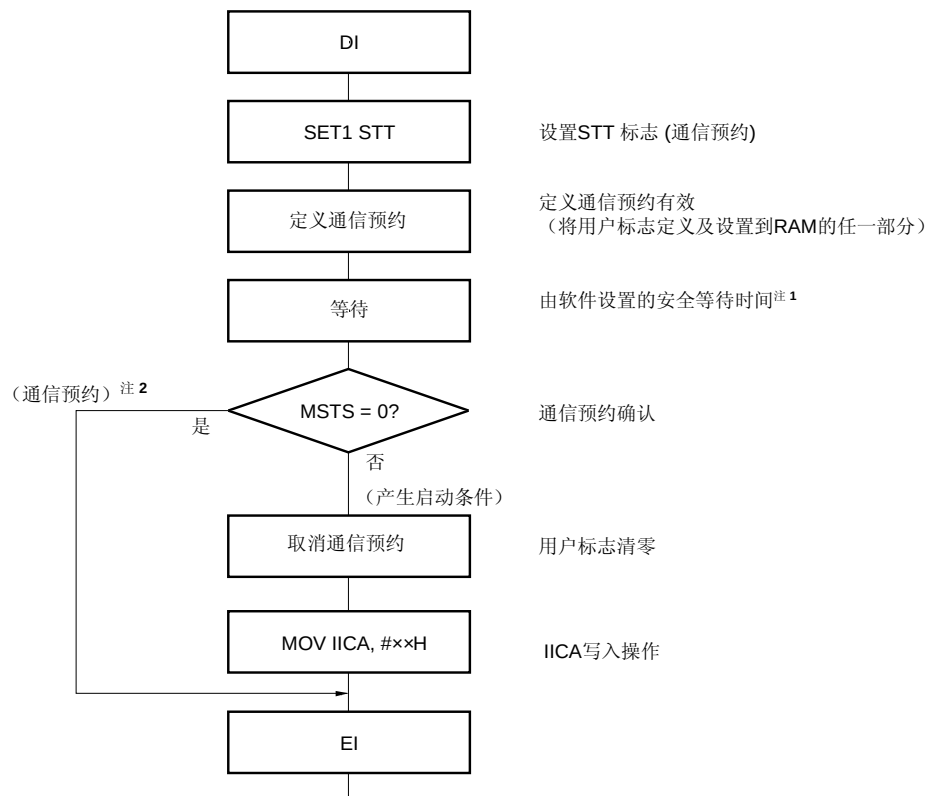


图 12-27 显示通信预约协议。

图 12-27. 通信预约协议



- 注
1. 等待时间计算如下。
(IICWL 设置值 + IICWH 设置值 + 4) + $t_F \times 2 \times f_{CLK}$ [时钟]
 2. 当一个停止条件中断请求产生时，通信预约执行一个对 IICA 移位寄存器 (IICA) 的写操作。

备注

STT: IICA 控制寄存器 0 的第 1 位 (IICCTL0)

MSTS: IICA 状态寄存器(IICS)的位 7

IICA: IICA 移位寄存器

IICWL: IICA 低电平宽度设置寄存器

IICWH: IICA 高电平宽度设置寄存器

t_F : SDA0 和 SCL0 信号下降时间

f_{CLK} : CPU/外围硬件时钟频率

(2) 当通信预约功能禁止时(IICA 标志寄存器 (IICF) 的第 0 位 (IICRSV) = 1)

如果 IICA 控制寄存器 0(IICCTL0)的位 1(STT)设置为 1，在总线通信期间，当设备不参与通信时，请求拒绝并且不会产生一个启动条件。下面两个状态包含在设备不参与通信的状态中。

- 当仲裁结果不是主/从设备操作时
- 当接收到扩展码和禁止从操作时($\overline{\text{ACK}}$ 没有返回，通过设置 IICCTL0 的位 6(LREL)为 1 并退出通信来释放总线)。

要确认是产生了一个启动条件还是请求拒绝，请检查 STCF(IICF 的第 7 位)。设置 STT = 1 后，直到 STCF 设置为 1，需要 5 个时钟。因此，必须通过软件确保时间。

12.5.15 注意事项

(1) 当 STCEN (IICA 标志寄存器 (IICF) 的第 1 位) = 0 时

在 I²C 操作允许 (IICE = 1) 后, 无论总线的实际状态如何, 立刻可以识别总线通信状态 (IICBSY (IICF 的第 6 位) = 1)。从检测到非停止条件模式变为主设备通信模式时, 首先产生一个停止条件以释放总线, 然后进行主设备通信。

当使用多个主设备时, 如果总线未释放(没有检测到停止条件时), 则不能进行主设备通信。

按下列步骤产生停止条件。

<1> 设置 IICA 控制寄存器 1 (IICCTL1)。

<2> 设置 IICA 控制寄存器 0 (IICCTL0) 的位 7 (IICE) 为 1。

<3> 设置 IICCTL0 的位 0 (SPT) 为 1。

(2) 当 STCEN = 1

在 I²C 操作允许 (IICE = 1) 后, 无论总线的实际状态如何, 立刻可以识别总线释放状态 (IICBSY = 0)。要产生第一个启动条件 (STT (IICA 控制寄存器 0 (IICCTL0) 的第 1 位) = 1), 必须确认总线已释放, 以免干扰其它通信。

(3) 如果其它 I²C 通信已经进行

如果 SDA0 引脚为低电平时 SCL0 引脚为高电平时允许 I²C 操作, 且设备已进行通信, 则 I²C 的模块可以识别 SDA0 引脚已经变为低电平 (检测一个启动条件)。如果此时总线上的值识别为一个扩展代码, 则返回 ACK 信号, 但这会干扰其它 I²C 通信。为了避免这种情况, 可按下列步骤启动 I²C。

<1> 当检测到一个停止条件时, 将 IICCTL0 的第 4 位 (SPIE) 清零, 从而禁止产生一个中断请求信号 (INTIICA)。

<2> 设置 IICCTL0 的第 7 位 (IICE) 为 1, 从而允许 I²C 的操作。

<3> 等待对启动条件的检测。

<4> $\overline{\text{ACK}}$ 信号返回之前 (设置 IICE 为 1 之后 4 到 80 个时钟), 设置 IICCTL0 的第 6 位 (LREL) 为 1, 从而强行禁止检测。

(4) 设置 STT 和 SPT (IICCTL0 的位 1 和 0) 之后及其清零禁止之前, 再次将其设置。

(5) 当发送预约时, 设置 SPIE (IICCTL0 的第 4 位) 为 1, 以便检测到停止条件时产生一个中断请求。中断请求产生后, 当通信数据写入 IICA 时, 发送过程开始。如果检测到停止条件时不产生中断请求, 设备将处于等待状态, 因为开始通信时不产生中断请求。但是, MSTs (IICS 的第 7 位) 由软件检测时, 不必设置 SPIE 为 1。

12.5.16 通信操作

以下显示了三种操作过程和其流程图。

(1) 主设备操作(单主设备系统)

在单主设备系统中，将 78K0R/KG3-L 用作主设备时的流程图显示如下。

该流程图分为初始化设置和通信过程。在程序开始处执行初始化设置。如果需要和从设备通信，准备通信并且执行通信处理。

(2) 主设备操作(多主设备系统)

在 I²C 总线的多主设备系统，当总线参与通信时，不能通过 I²C 总线规范判断总线是否释放。这样，当数据和时钟有一段时间处于高电平时(1 帧)，在总线释放状态 78K0R/KG3-L 参与通信。

该流程分为初始化设置，通信等待和通信处理。此处忽略了当 78K0R/KG3-L 仲裁失败而指定为从设备的处理，只显示了作为主设备的处理。在启动时执行初始化设置参与通信。然后，主设备等待通信请求或者从设备等待详细指定。在通信处理中执行实际通信，它支持用从设备发送/接收，并在主设备之间进行仲裁。

(3) 从设备操作

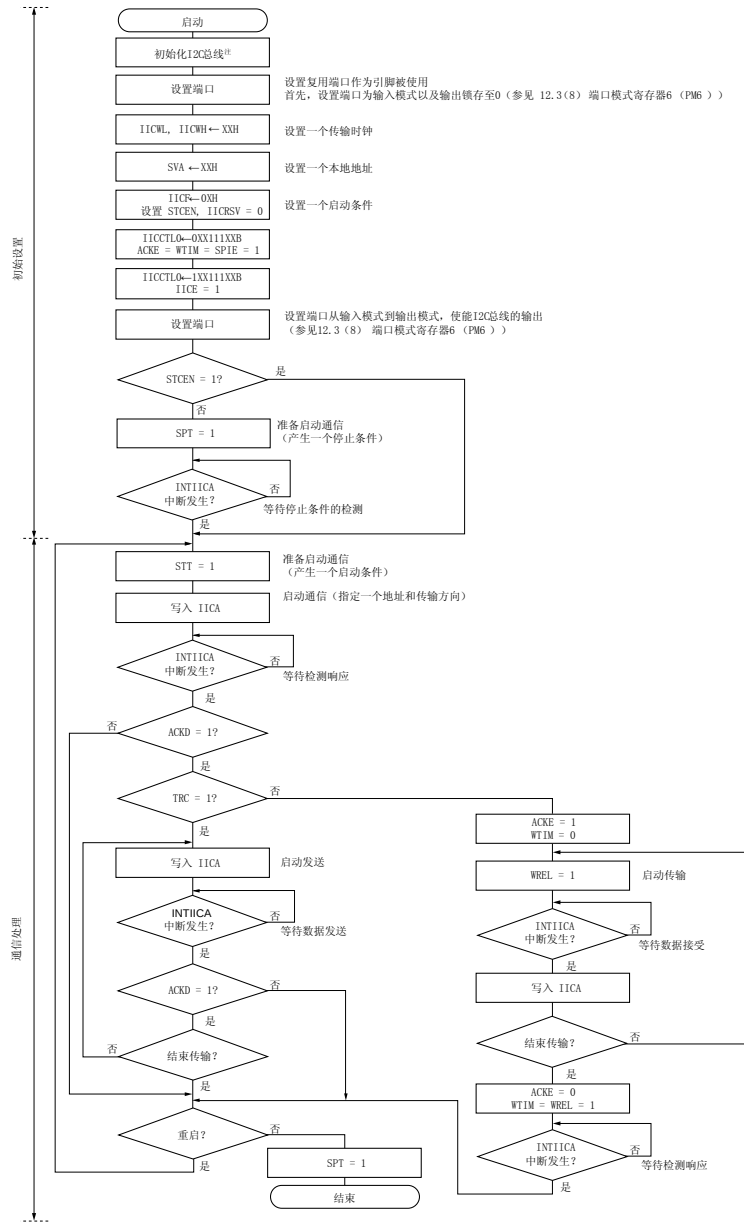
78K0R/KG3-L 用作 I²C 总线的从设备的示例显示如下。

当用作从设备时，由中断开始操作。启动时执行初始设置，然后等待 INTIICA 中断产生(通信等待)。当一个 INTIICA 中断产生时，判断通信状态并且判断结果作为一个标志传递给主程序处理。

通过检查标志，执行所需的通信处理。

(1) 主设备操作(单主设备系统)

图 12-28. 主设备操作(单主设备系统)



注 根据正在通信的产品规范释放 (SCL0 和 SDA0 引脚 = 高电平) I²C 总线。如果 EEPROM 向 SDA0 引脚输出一个低电平, 例如, 设置 SCL0 引脚为输出端口模式, 且从输出端口输出一个时钟脉冲直到 SDA0 引脚持续处于高电平。

备注 遵照正在通信的产品规范, 注意发送和接收的格式。

(2) 主设备操作(多主设备系统)

图 12-29. 主设备操作(多主设备系统)(1/3)

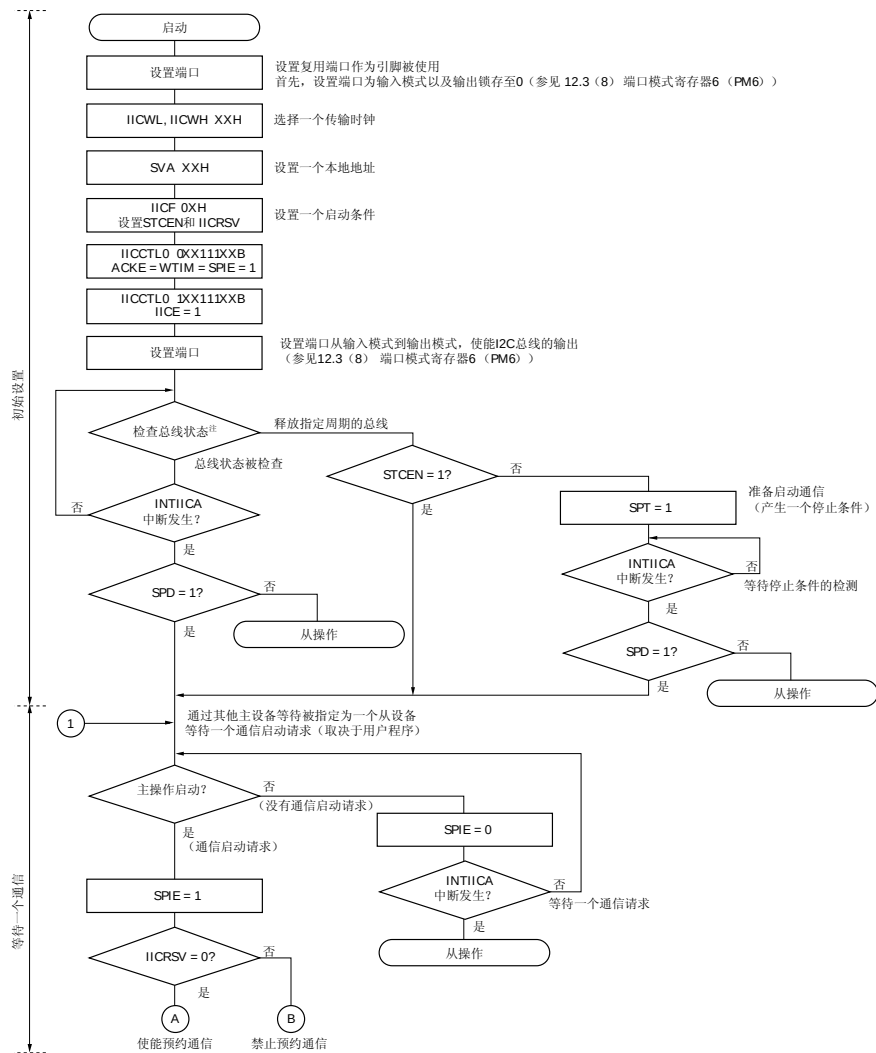
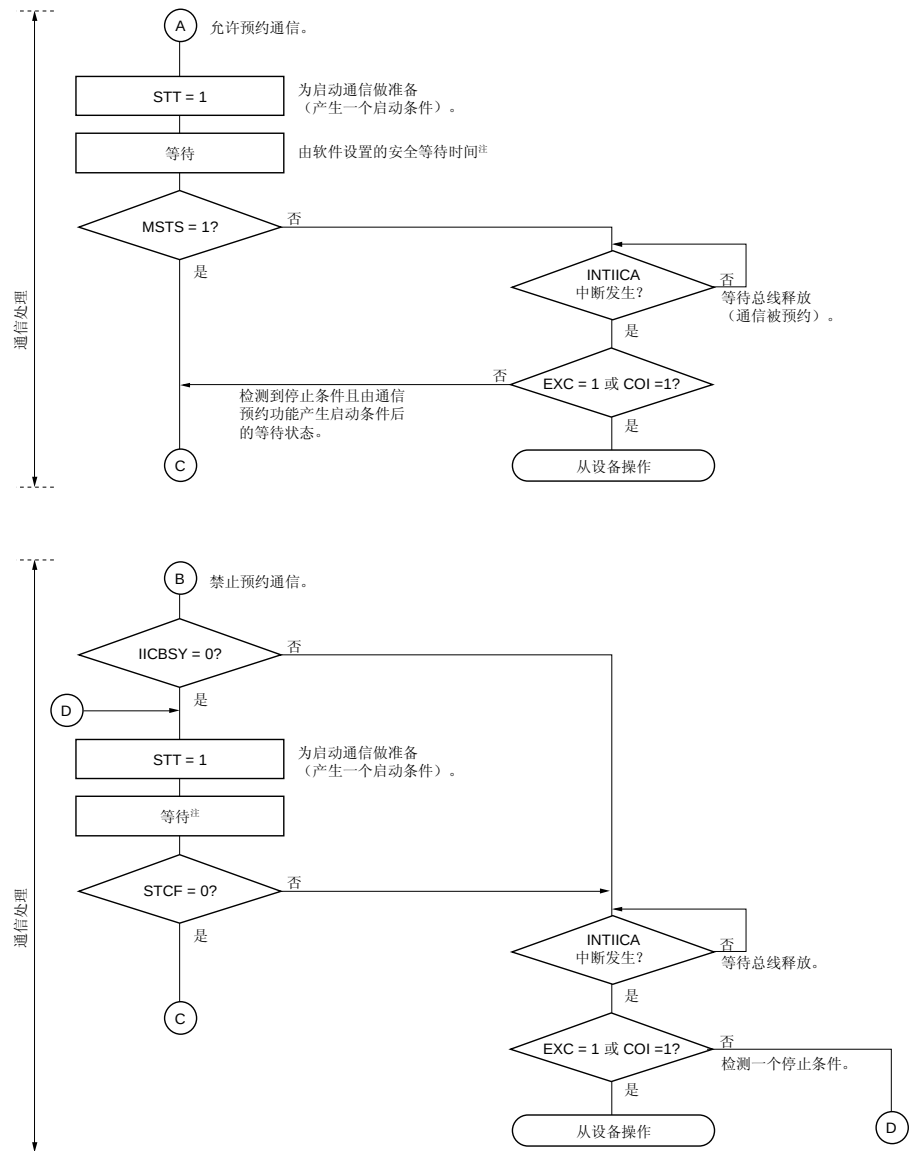


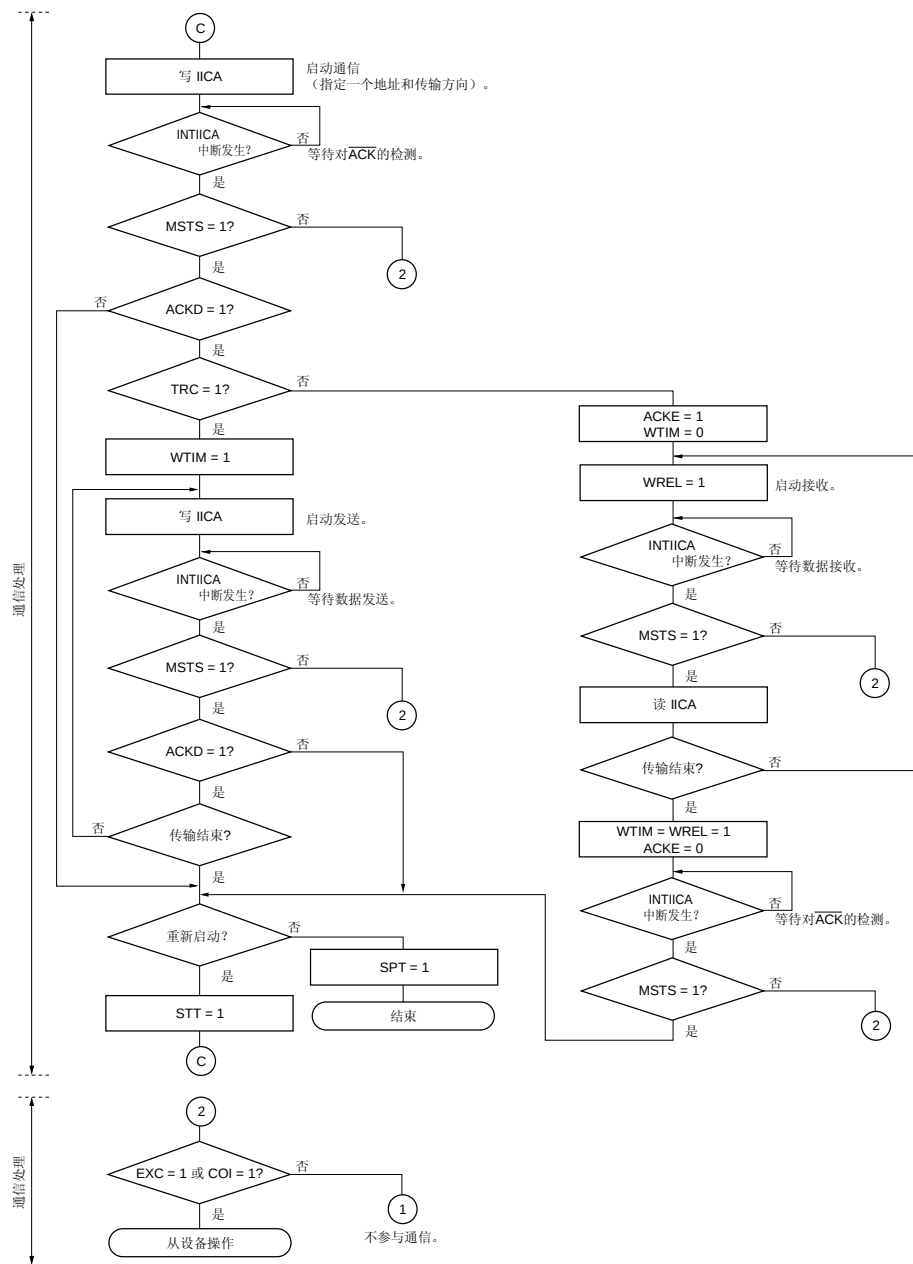
图 12-29. 主设备操作(多主设备系统)(2/3)



注 等待时间计算如下。
(IICWL 设置值 + IICWH 设置值 + 4) + t_F × 2 × f_{CLK} [时钟]

备注 IICWL: IICA 低电平宽度设置寄存器
IICWH: IICA 高电平宽度设置寄存器
t_F: SDA0 和 SCL0 信号下降时间
f_{CLK}: CPU/外围硬件时钟频率

图 12-29. 主设备操作(多主设备系统)(3/3)

**备注**

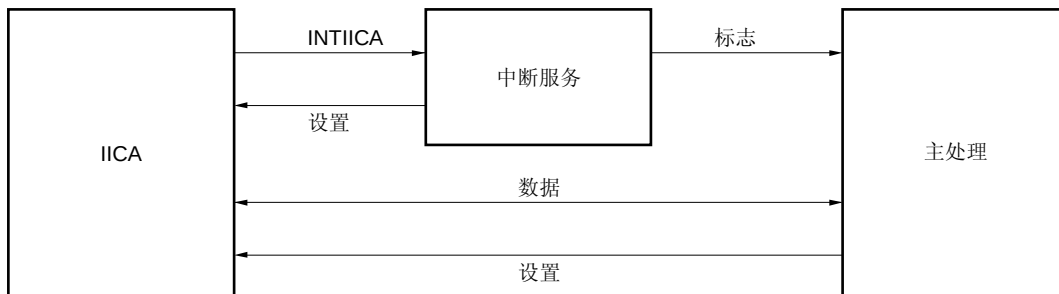
1. 遵照正在通信的产品规范，注意发送和接收的格式。
2. 要将此设备用作多主设备系统中的一个主设备时，则应在每次 INTIICA 中断发生后读取 MSTS 位来检查仲裁结果。
3. 要将此设备用作多主设备系统中的一个从设备时，则在每次 INTIICA 中断发生后，使用 IICS 和 IICF 寄存器来检查状态，并决定下一步的处理。

(3) 从设备操作

从设备操作过程如下所示。

基本上，从设备操作是由事件驱动的。因此，有必要执行 INTIICA 中断处理过程(该中断处理过程必须实质性地改变操作状态，例如通信期间的停止条件检测)。

在下列说明中，假定数据通信不支持扩展代码。同时假定 INTIICA 中断服务只执行状态转换处理，且实际数据通信过程由主程序执行。



因此，通过准备以下 3 个标志，并将其传递给主程序而不是 INTIICA，就可以执行数据通信处理过程了。

<1> 通信模式标志

此标志指示下列两种通信状态。

- 清除模式：未进行数据通信
- 通信模式：进行数据通信(从有效地址检测到停止条件检测，不检测来自主设备的 $\overline{\text{ACK}}$ 信号，地址不匹配)

<2> 准备标志

此标志表示允许进行数据通信。其功能和普通数据通信下的 INTIICA 中断相同。通过中断服务可以设置此标志，通过主程序可以将该标志清零。通信启动时，可由中断服务将此标志清零。但是，在发送第一个数据时不能由中断服务设置准备标志。因此，发送第一个数据时该标志未清零(地址匹配解释为下一个数据的请求)。

<3> 通信方向标志

此标志指示通信方向。它的值与 TRC 相同。

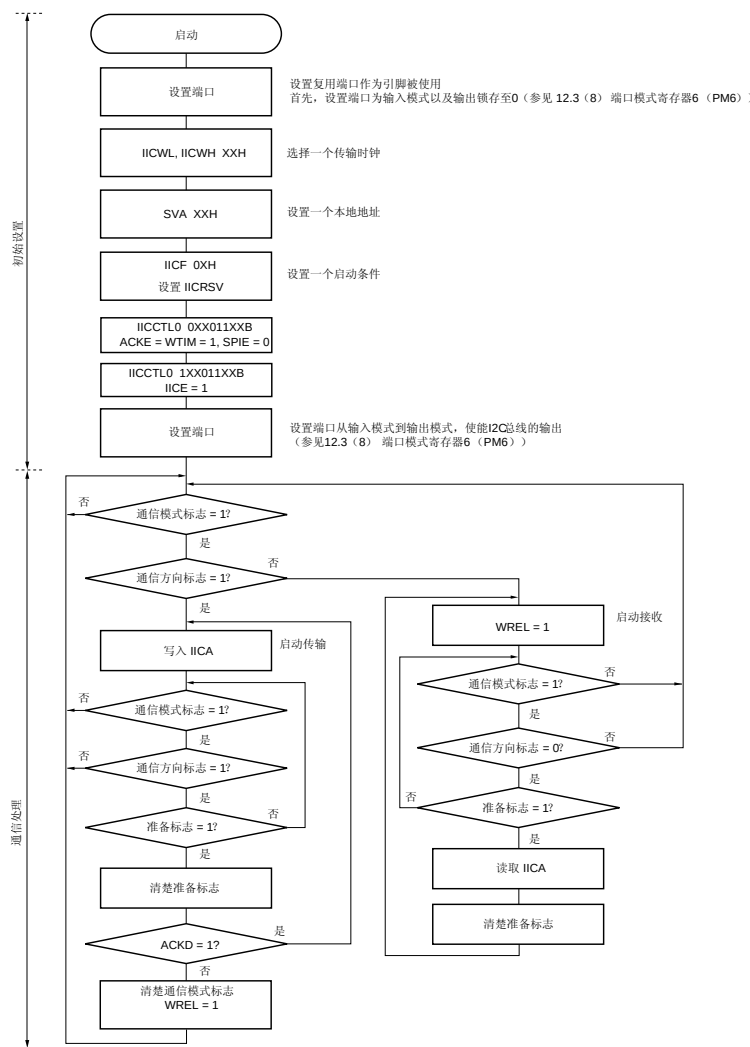
从设备操作的主程序处理过程说明如下。

启动串行接口 **IICA** 且等待到通信允许。当允许通信时，通过使用通信模式标志和准备标志来执行通信(由中断来执行停止条件和启动条件的处理。这里，利用标志来检查状态。)

重复发送操作直到主设备不再返回 **ACK** 信号。如果没有从主设备返回 **ACK** 信号，则通信结束。

对于接收操作，则要接收一定量的数据。通信完成时，**ACK** 信号不作为下一个数据返回。此后，主设备产生一个停止条件或重启条件。使用这种方法可从通信状态退出。

图 12-30. 从设备操作流程图中 (1)



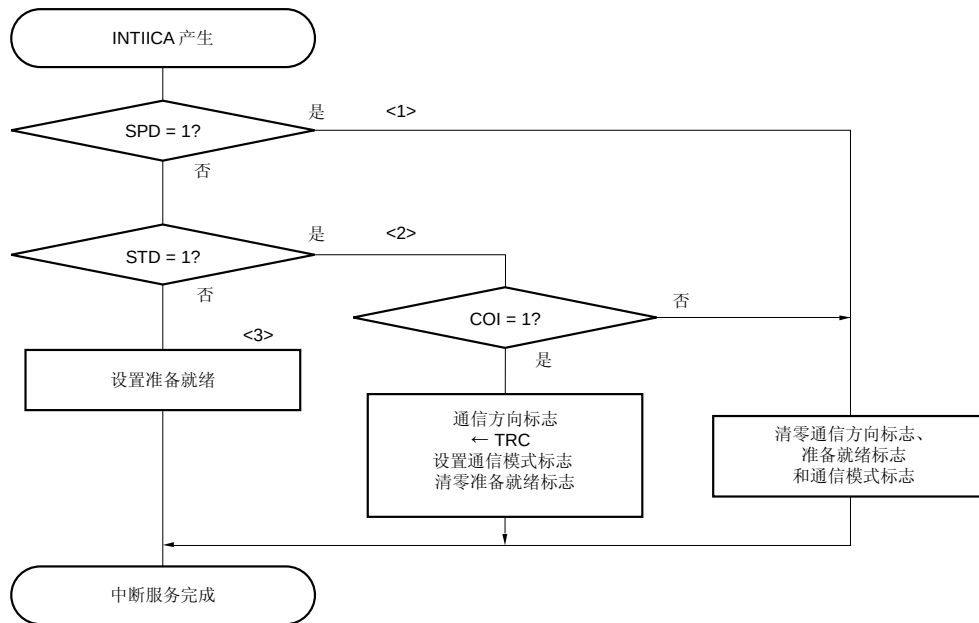
备注 遵照正在通信的产品规范，注意发送和接收的格式。

以下说明了一个带 INTIICA 中断的从设备的处理过程的示例(处理中假设未使用扩展代码)。INTIICA 中断检查状态,同时执行下列操作。

- <1> 若发出停止条件则通信停止。
- <2> 若发出启动条件,检查地址并在地址不匹配时结束通信。如果地址匹配,则设置通信模式,取消等待,且处理过程从中断返回(准备标志清零)。
- <3> 对于数据发送/接收,仅需设置准备标志。处理过程从中断返回,同时I²C总线保持等待状态。

备注 以上<1>至<3>与图 12-31 从设备操作流程图(2)中<1>至<3>相对应。

图 12-31. 从设备操作流程图(2)



12.5.17 产生 I²C 中断请求 (INTIICA) 的时序

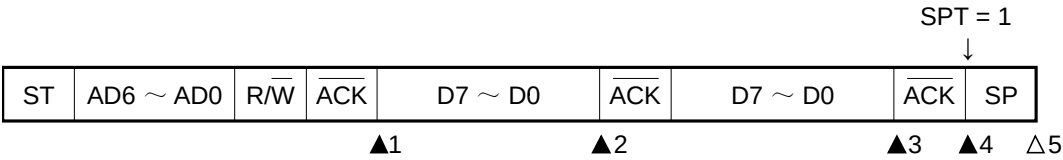
发送或接收数据的时序、中断请求信号 INTIICA 的产生、当产生 INTIICA 信号时 IICS 寄存器的值，如下所示。

备注	ST:	起始条件
	AD6 至 AD0:	地址
	R/W:	发送方向指示
	ACK:	应答
	D7 至 D0:	数据
	SP:	停止条件

(1) 主设备操作

(a) 启动 ~ 地址 ~ 数据 ~ 数据 ~ 停止（发送/接收）

(i) 当 **WTIM = 0**

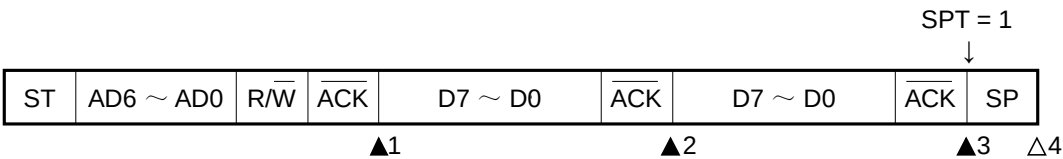


- ▲1: IICS = 1000×110B
▲2: IICS = 1000×100B
▲3: IICS = 1000×000B (设置 WTIM 为 1)^注
▲4: IICS = 1000××00B (设置 SPT 为 1)
Δ5: IICS = 00000001B

注 要产生一个停止条件，应设置 **WTIM** 为 **1** 且改变产生 **INTIICA** 中断请求信号的时序。

备注 ▲: 一直产生
 Δ: 仅当 **SPIE = 1** 时产生
 ×: 不必考虑

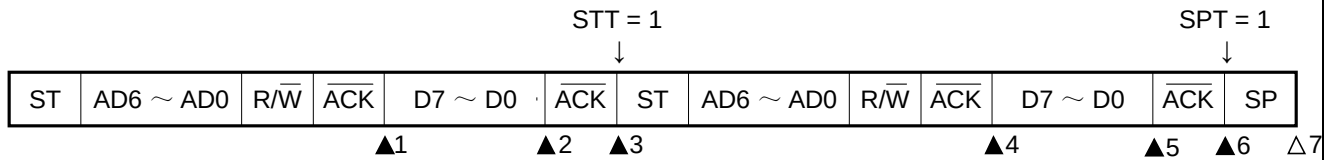
(ii) 当 **WTIM = 1**



- ▲1: IICS = 1000×110B
▲2: IICS = 1000×100B
▲3: IICS = 1000××00B (设置 SPT 为 1)
Δ4: IICS = 00000001B

备注 ▲: 一直产生
 Δ: 仅当 **SPIE = 1** 时产生
 ×: 不必考虑

(b) 启动 ~ 地址 ~ 数据 ~ 启动 ~ 地址 ~ 数据 ~ 停止 (重启)

(i) 当 $WTIM = 0$ 

▲1: IICS = 1000×110B

▲2: IICS = 1000×000B (设置 $WTIM$ 为 1)^{注1}▲3: IICS = 1000××00B (清除 $WTIM$ 为 0^{注2}, 设置 STT 为 1)

▲4: IICS = 1000×110B

▲5: IICS = 1000×000B (设置 $WTIM$ 为 1)^{注3}▲6: IICS = 1000××00B (设置 SPT 为 1)

△7: IICS = 00000001B

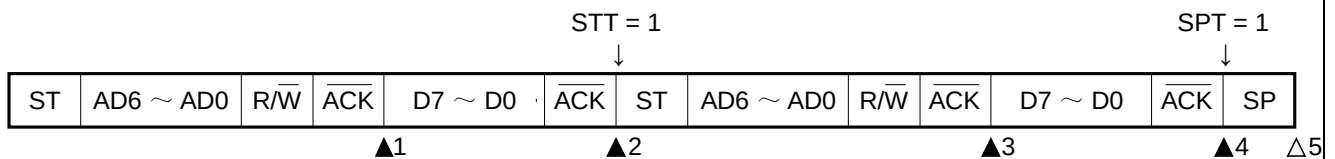
- 注
1. 要产生一个启动条件, 应设置 $WTIM$ 为 1 且改变产生 $INTIICA$ 中断请求信号的时序。
 2. 清除 $WTIM$ 为 0 来保存最初设置。
 3. 要产生一个停止条件, 应设置 $WTIM$ 为 1 且改变产生 $INTIICA$ 中断请求信号的时序。

备注

▲: 一直产生

△: 仅当 $SPIE = 1$ 时产生

×: 不必考虑

(ii) 当 $WTIM = 1$ 

▲1: IICS = 1000×110B

▲2: IICS = 1000××00B (设置 STT 为 1)

▲3: IICS = 1000×110B

▲4: IICS = 1000××00B (设置 SPT 为 1)

△5: IICS = 00000001B

备注

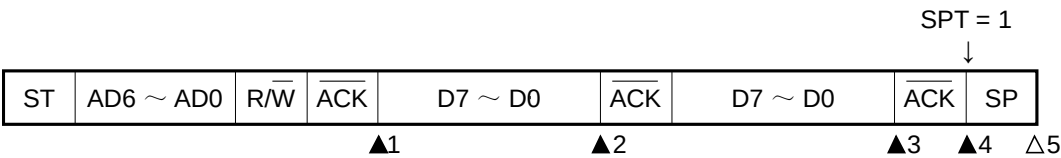
▲: 一直产生

△: 仅当 $SPIE = 1$ 时产生

×: 不必考虑

(c) 启动 ~ 代码 ~ 数据 ~ 数据 ~ 停止 (扩展代码发送)

(i) 当 $WTIM = 0$

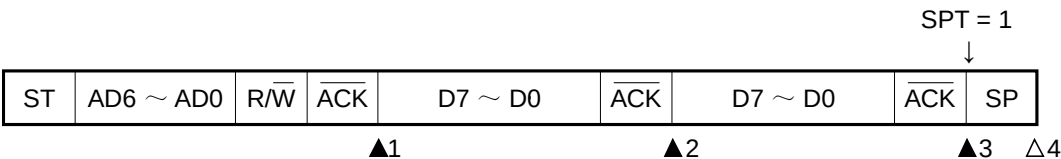


- ▲1: IICS = 1010×110B
▲2: IICS = 1010×000B
▲3: IICS = 1010×000B (设置 WTIM 为 1)^注
▲4: IICS = 1010××00B (设置 SPT 为 1)
Δ5: IICS = 00000001B

注 要产生一个停止条件，应设置 WTIM 为 1 且改变产生 INTIICA 中断请求信号的时序。

备注 ▲: 一直产生
 Δ: 仅当 SPIE = 1 时产生
 ×: 不必考虑

(ii) 当 $WTIM = 1$



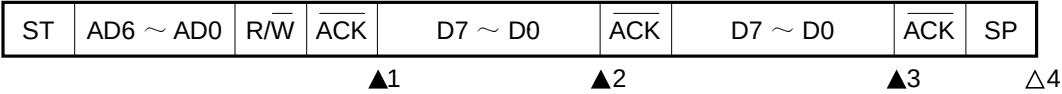
- ▲1: IICS = 1010×110B
▲2: IICS = 1010×000B
▲3: IICS = 1010××00B (设置 SPT 为 1)
Δ4: IICS = 00001001B

备注 ▲: 一直产生
 Δ: 仅当 SPIE = 1 时产生
 ×: 不必考虑

(2) 从设备操作（从地址数据接收）

(a) 启动 ~ 地址 ~ 数据 ~ 数据 ~ 停止

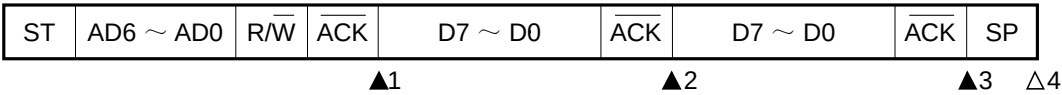
(i) 当 WTIM = 0



- ▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×000B
△4: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 SPIE = 1 时产生
 ×: 不必考虑

(ii) 当 WTIM = 1

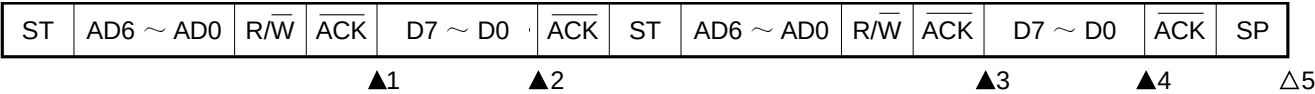


- ▲1: IICS = 0001×110B
▲2: IICS = 0001×100B
▲3: IICS = 0001××00B
△4: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 SPIE = 1 时产生
 ×: 不必考虑

(b) 启动 ~ 地址 ~ 数据 ~ 启动 ~ 地址 ~ 数据 ~ 停止

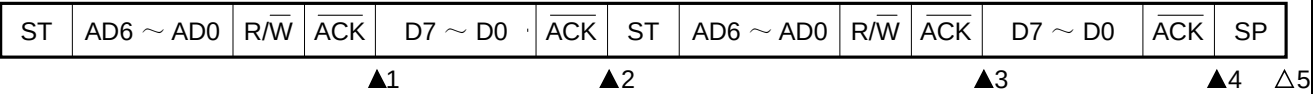
(i) 当 $WTIM = 0$ (重新启动后, 和 SVA 匹配)



- ▲1: IICS = 0001×110B
- ▲2: IICS = 0001×000B
- ▲3: IICS = 0001×110B
- ▲4: IICS = 0001×000B
- △5: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 $SPIE = 1$ 时产生
 ×: 不必考虑

(ii) 当 $WTIM = 1$ (重新启动后, 和 SVA 匹配)

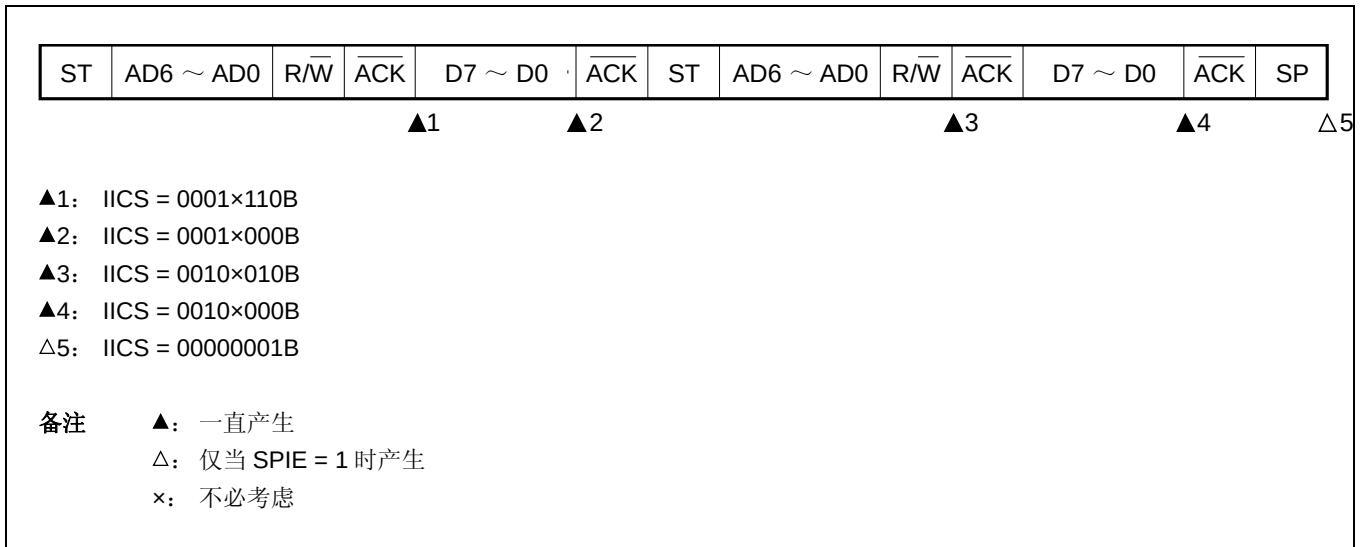


- ▲1: IICS = 0001×110B
- ▲2: IICS = 0001××00B
- ▲3: IICS = 0001×110B
- ▲4: IICS = 0001××00B
- △5: IICS = 00000001B

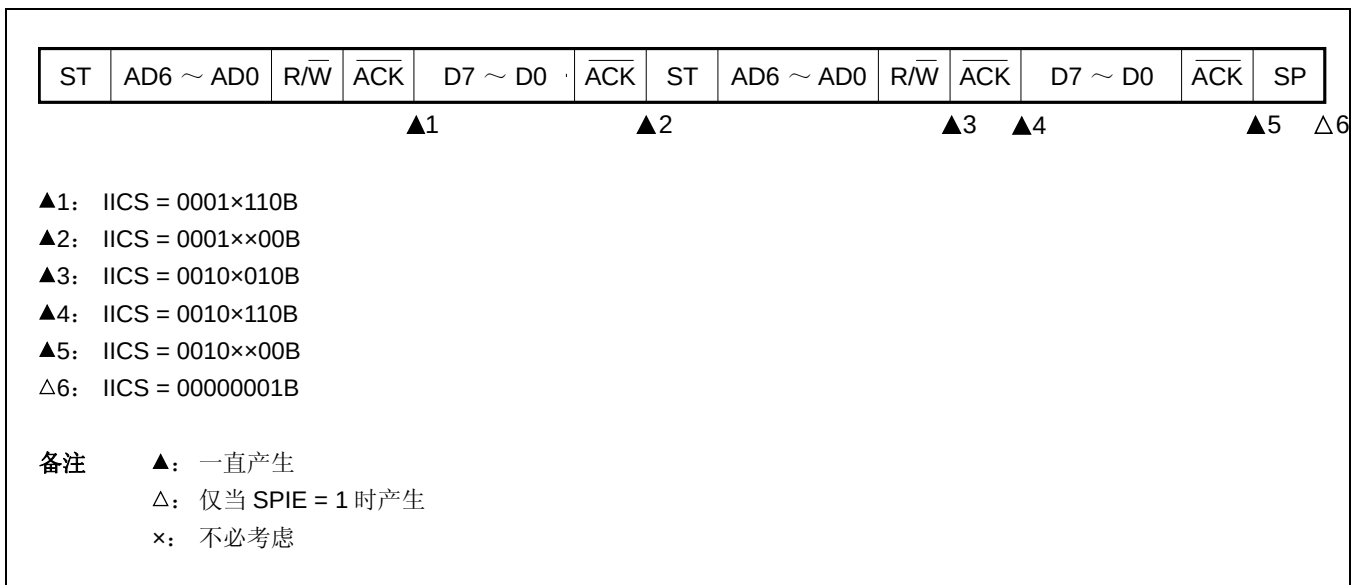
备注 ▲: 一直产生
 △: 仅当 $SPIE = 1$ 时产生
 ×: 不必考虑

(c) 启动 ~ 地址 ~ 数据 ~ 启动 ~ 代码 ~ 数据 ~ 停止

(i) 当 $WTIM = 0$ (重新启动后, 和地址不匹配 (= 扩展代码))

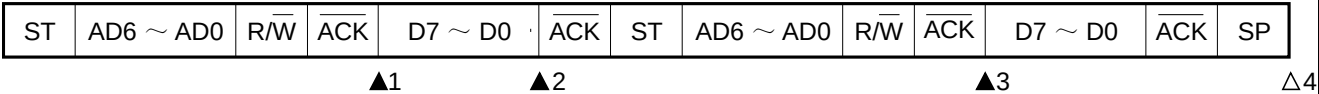


(ii) 当 $WTIM = 1$ (重新启动后, 和地址不匹配 (= 扩展代码))



(d) 启动 ~ 地址 ~ 数据 ~ 启动 ~ 地址 ~ 数据 ~ 停止

(i) 当 $WTIM = 0$ (重新启动后, 和地址不匹配 (= 非扩展代码))



▲1: IICS = 0001×110B

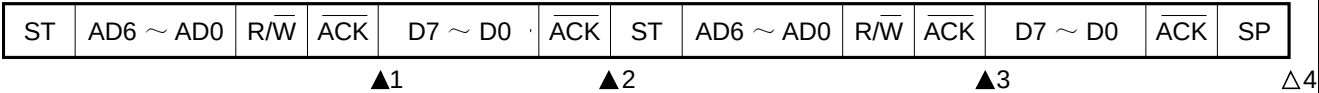
▲2: IICS = 0001×000B

▲3: IICS = 00000110B

△4: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 SPIE = 1 时产生
 ×: 不必考虑

(ii) 当 $WTIM = 1$ (重新启动后, 和地址不匹配 (= 非扩展代码))



▲1: IICS = 0001×110B

▲2: IICS = 0001××00B

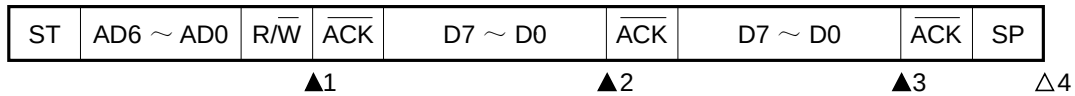
▲3: IICS = 00000110B

△4: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 SPIE = 1 时产生
 ×: 不必考虑

(3) 从设备操作（当接收扩展代码时）

在接收扩展代码时设备一直参与通信。

(a) 启动 ~ 代码 ~ 数据 ~ 数据 ~ 停止**(i) 当 WTIM = 0**

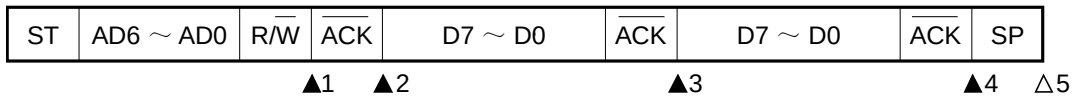
▲1: IICS = 0010×010B

▲2: IICS = 0010×000B

▲3: IICS = 0010×000B

△4: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 SPIE = 1 时产生
 ×: 不必考虑

(ii) 当 WTIM = 1

▲1: IICS = 0010×010B

▲2: IICS = 0010×110B

▲3: IICS = 0010×100B

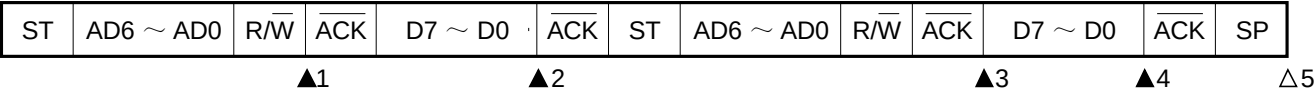
▲4: IICS = 0010××00B

△5: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 SPIE = 1 时产生
 ×: 不必考虑

(b) 启动 ~ 代码 ~ 数据 ~ 启动 ~ 地址 ~ 数据 ~ 停止

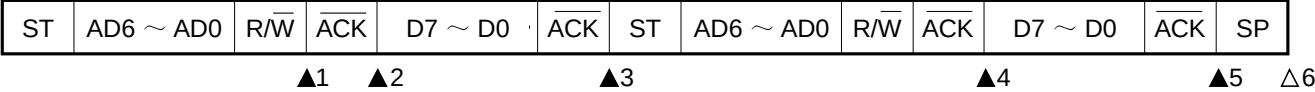
(i) 当 $WTIM = 0$ (重新启动后, 和 SVA 匹配)



- ▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0001×110B
▲4: IICS = 0001×000B
△5: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 $SPIE = 1$ 时产生
 ×: 不必考虑

(ii) 当 $WTIM = 1$ (重新启动后, 和 SVA 匹配)



- ▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010××00B
▲4: IICS = 0001×110B
▲5: IICS = 0001××00B
△6: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 $SPIE = 1$ 时产生
 ×: 不必考虑

(c) 启动 ~ 代码 ~ 数据 ~ 启动 ~ 代码 ~ 数据 ~ 停止

(i) 当 $WTIM0 = 0$ (复位后, 接收扩展代码)



▲1: IICS = 0010×010B

▲2: IICS = 0010×000B

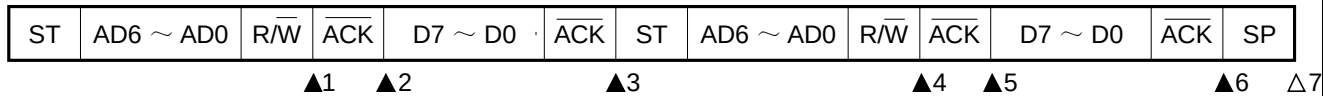
▲3: IICS = 0010×010B

▲4: IICS = 0010×000B

△5: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 $SPIE = 1$ 时产生
 ×: 不必考虑

(ii) 当 $WTIM0 = 1$ (复位后, 接收扩展代码)



▲1: IICS = 0010×010B

▲2: IICS = 0010×110B

▲3: IICS = 0010××00B

▲4: IICS = 0010×010B

▲5: IICS = 0010×110B

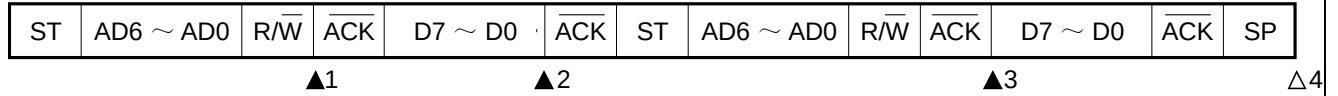
▲6: IICS = 0010××00B

△7: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 $SPIE = 1$ 时产生
 ×: 不必考虑

(d) 启动 ~ 代码 ~ 数据 ~ 启动 ~ 地址 ~ 数据 ~ 停止

(i) 当 $WTIM = 0$ (重新启动后, 和地址不匹配 (= 非扩展代码))



▲1: IICS = 00100010B

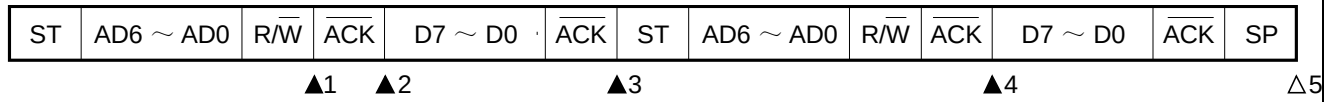
▲2: IICS = 00100000B

▲3: IICS = 00000110B

△4: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 $SPIE = 1$ 时产生
 ×: 不必考虑

(ii) 当 $WTIM = 1$ (重新启动后, 和地址不匹配 (= 非扩展代码))



▲1: IICS = 00100010B

▲2: IICS = 00100110B

▲3: IICS = 00100×00B

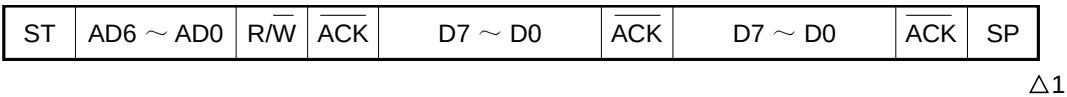
▲4: IICS = 00000110B

△5: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 $SPIE = 1$ 时产生
 ×: 不必考虑

(4) 没有通信的操作

(a) 启动 ~ 代码 ~ 数据 ~ 数据 ~ 停止



△1: IICS = 00000001B

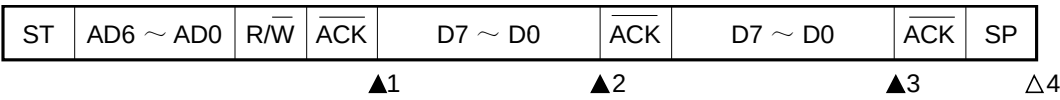
备注 △: 仅当 SPIE = 1 时产生

(5) 仲裁失败操作（仲裁失败后作为从设备操作）

当此设备在多主设备系统用作一个主设备时，每次产生中断请求信号 INTIICA 读取 MST5 位来检查仲裁结果。

(a) 在从地址数据发送时产生仲裁失败

(i) 当 WTIM = 0



▲1: IICS = 0101×110B

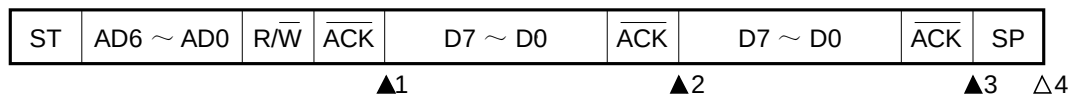
▲2: IICS = 0001×000B

▲3: IICS = 0001×000B

△4: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 SPIE = 1 时产生
 ×: 不必考虑

(ii) 当 $WTIM = 1$

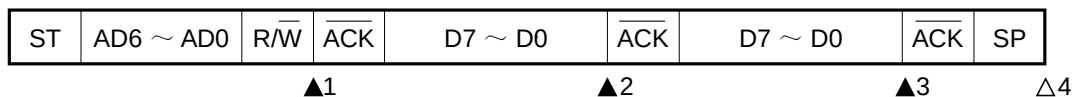


- $\Delta 4$: IICS = 00000001B

备注 ▲: 一直产生
 Δ: 仅当 SPIE = 1 时产生
 x: 不必考虑

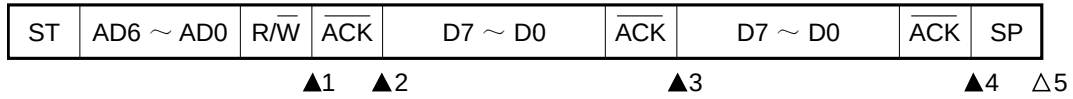
(b) 在扩展代码发送时产生仲裁失败

(i) 当 $WTIM = 0$



- Δ4: IICS = 00000001B

备注 ▲：一直产生
 △：仅当 SPIE = 1 时产生
 ×：不必考虑

(ii) 当 $WTIM = 1$ 

▲1: IICS = 0110×010B

▲2: IICS = 0010×110B

▲3: IICS = 0010×100B

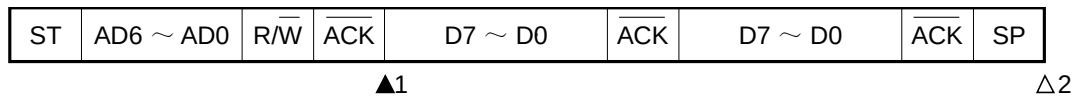
▲4: IICS = 0010××00B

Δ5: IICS = 00000001B

备注 ▲: 一直产生
 Δ: 仅当 $SPIE = 1$ 时产生
 ×: 不必考虑

(6) 当发送仲裁失败时的操作（仲裁失败后没有通信）

当此设备在多主设备系统用作一个主设备时，每次产生中断请求信号 INTIICA 读取 MSTS 位来检查仲裁结果。

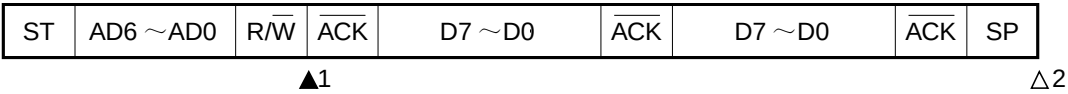
(a) 从地址数据发送期间，产生仲裁失败时（当 $WTIM = 1$ ）

▲1: IICS = 01000110B

Δ2: IICS = 00000001B

备注 ▲: 一直产生
 Δ: 仅当 $SPIE = 1$ 时产生

(b) 在扩展代码发送时产生仲裁失败

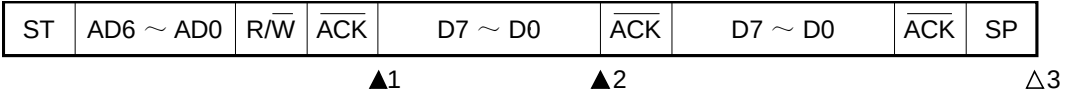


▲1: IICS = 0110×10B
通过软件设置 LREL = 1
△2: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 SPIE = 1 时产生
 ×: 不必考虑

(c) 在数据发送过程中产生仲裁失败

(i) 当 WTIM = 0

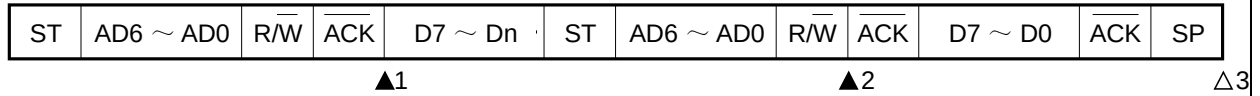


▲1: IICS = 10001110B
▲2: IICS = 01000000B
△3: IICS = 00000001B

备注 ▲: 一直产生
 △: 仅当 SPIE = 1 时产生



(ii) 扩展代码



▲1: IICS = 1000×110B

▲2: IICS = 01100010B

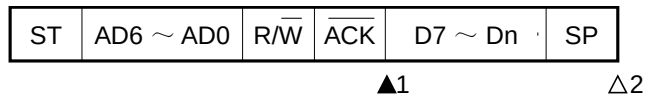
通过软件设置 $LREL = 1$

$\Delta 3$: IICS = 00000001B

备注

- ▲：一直产生
- △：仅当 SPIE = 1 时产生
- ×：不必考虑
- n = 6 至 0

(e) 由于在数据传输过程中的停止条件而导致损失



▲1: IICS = 10000110B

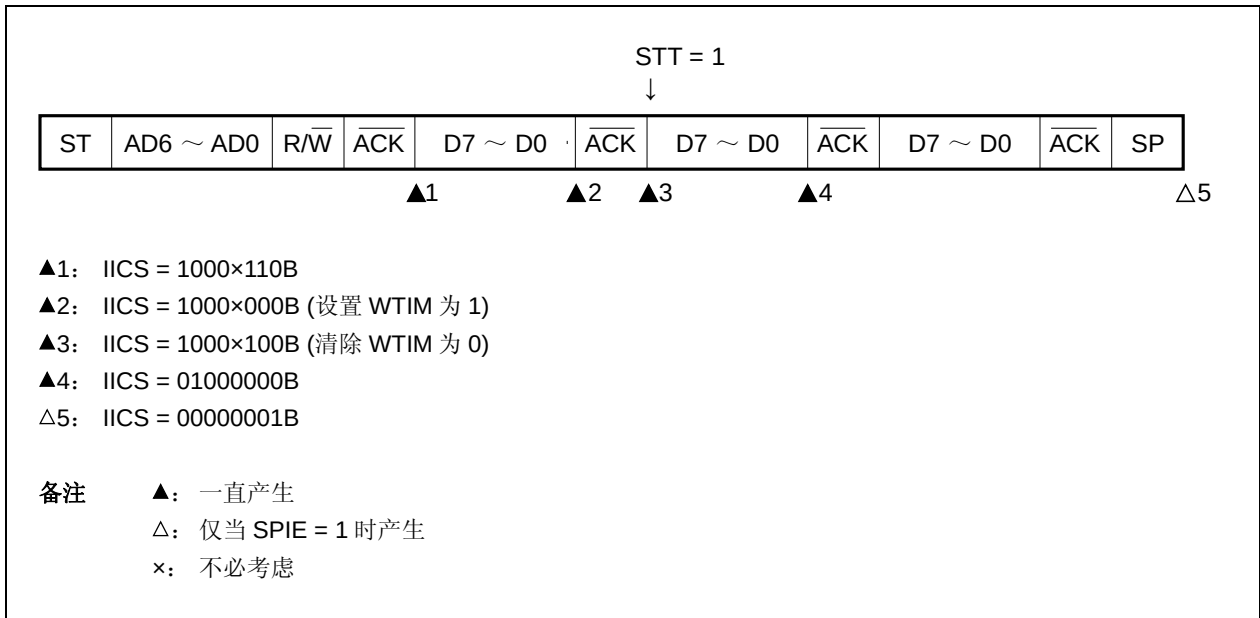
$\Delta 2$: IICS = 01000001B

备注

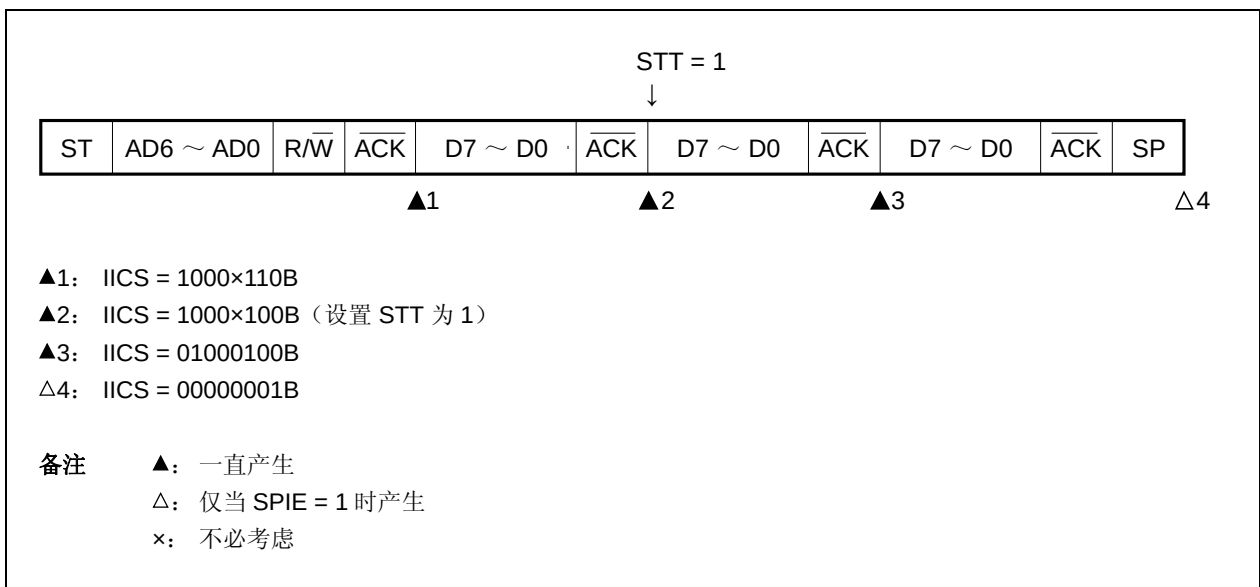
- ▲：一直产生
- △：仅当 SPIE = 1 时产生
- ×：不必考虑
- n = 6 至 0

(f) 由于在试图产生重启条件时的低电平数据而导致仲裁失败

(i) 当 $WTIM = 0$

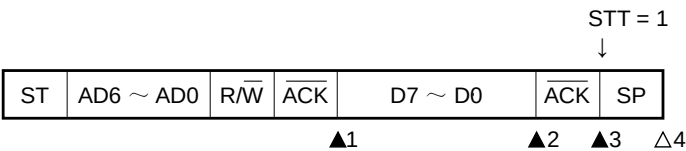


(ii) 当 $WTIM = 1$



(g) 由于在试图产生重启条件时的停止条件而导致仲裁失败产生

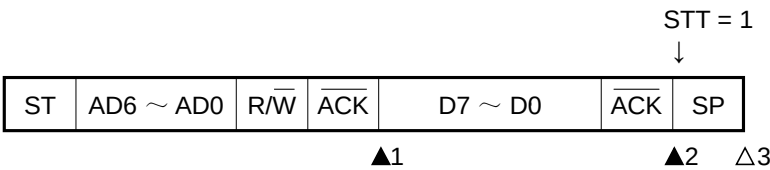
(i) 当 $WTIM = 0$



- ▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (设置 $WTIM$ 为 1)
▲3: IICS = 1000××00B (设置 STT 为 1)
△4: IICS = 01000001B

备注 ▲: 一直产生
 △: 仅当 $SPIE = 1$ 时产生
 ×: 不必考虑

(ii) 当 $WTIM = 1$

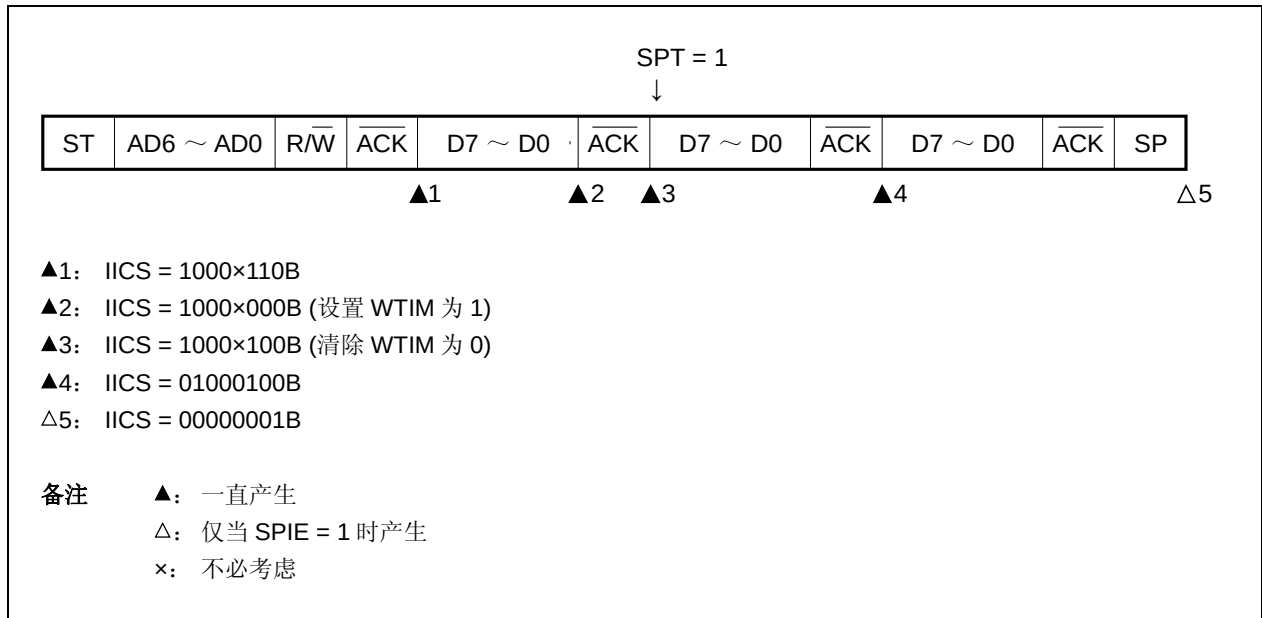


- ▲1: IICS = 1000×110B
▲2: IICS = 1000××00B (设置 STT 为 1)
△3: IICS = 01000001B

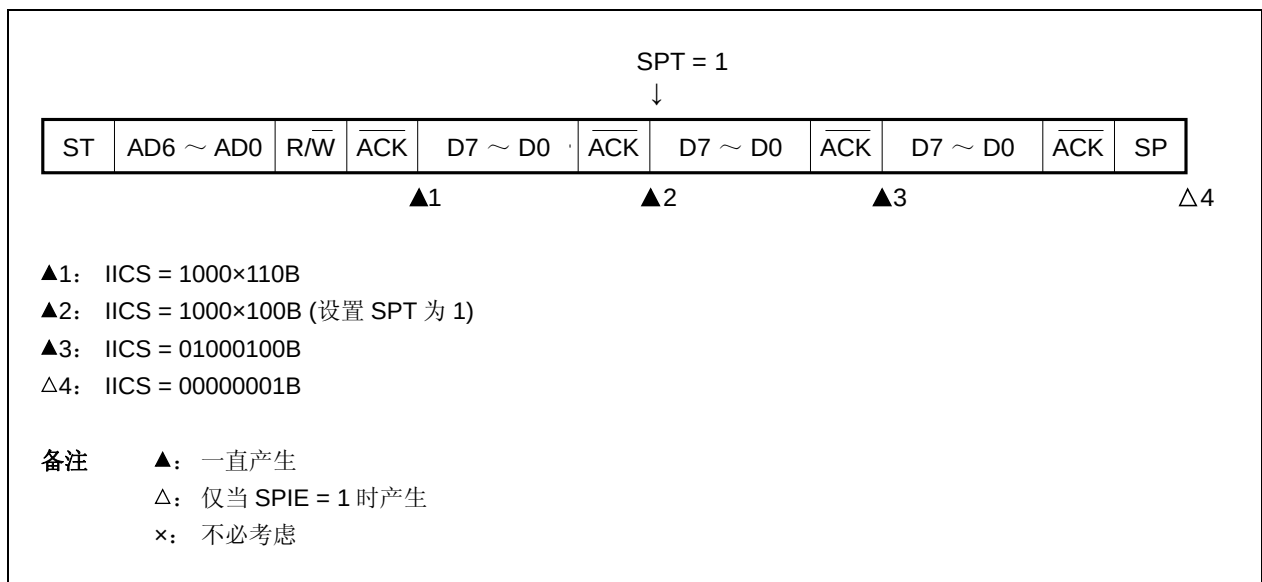
备注 ▲: 一直产生
 △: 仅当 $SPIE = 1$ 时产生
 ×: 不必考虑

(h) 由于在试图停止条件时的低电平数据而导致仲裁失败产生

(i) 当 $WTIM = 0$



(ii) 当 $WTIM = 1$



12.6 时序图

当使用 I²C 总线模式，主设备通过串行总线输出一个地址来选择一个从设备作为通信对象。

输出从地址后，主设备发送 TRC 位（IICA 状态寄存器 (IICS) 的位 3），来指定数据传输方向，且用从设备启动串行通信。

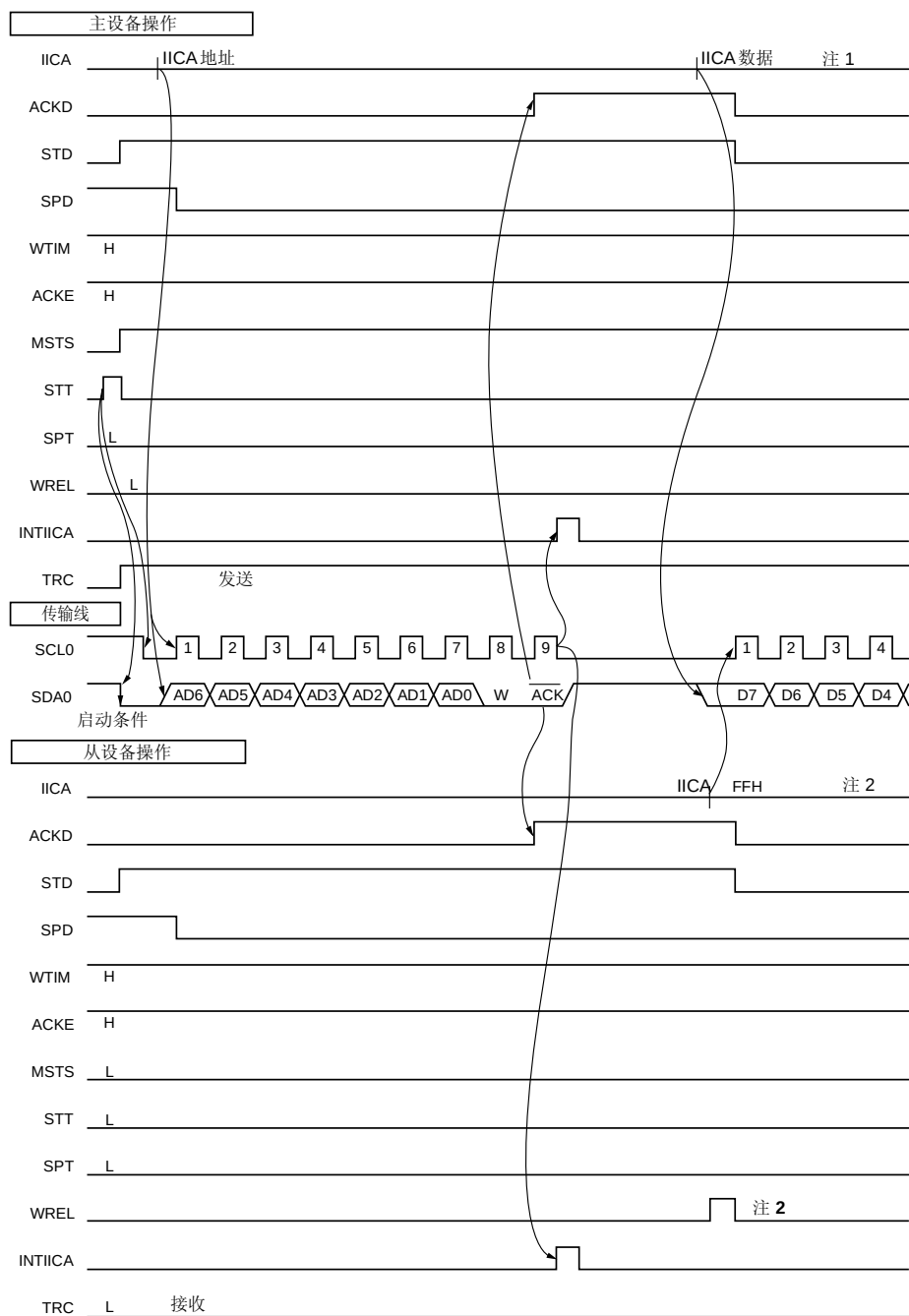
图 12-32 和 12-33 显示了数据通信的时序图。

IICA 移位寄存器(IICA)的移位操作与串行时钟(SCL0)的下降沿同步。发送数据传输到 SO 锁存并且通过 SDA0 引脚输出 (MSB 优先)。

通过 SDA0 引脚，在 SCL0 的上升沿，数据输入被捕捉到 IICA。

图 12-32. 主设备到从设备通信的例子
(当为每个主设备和从设备选择 9 个时钟等待时) (1/3)

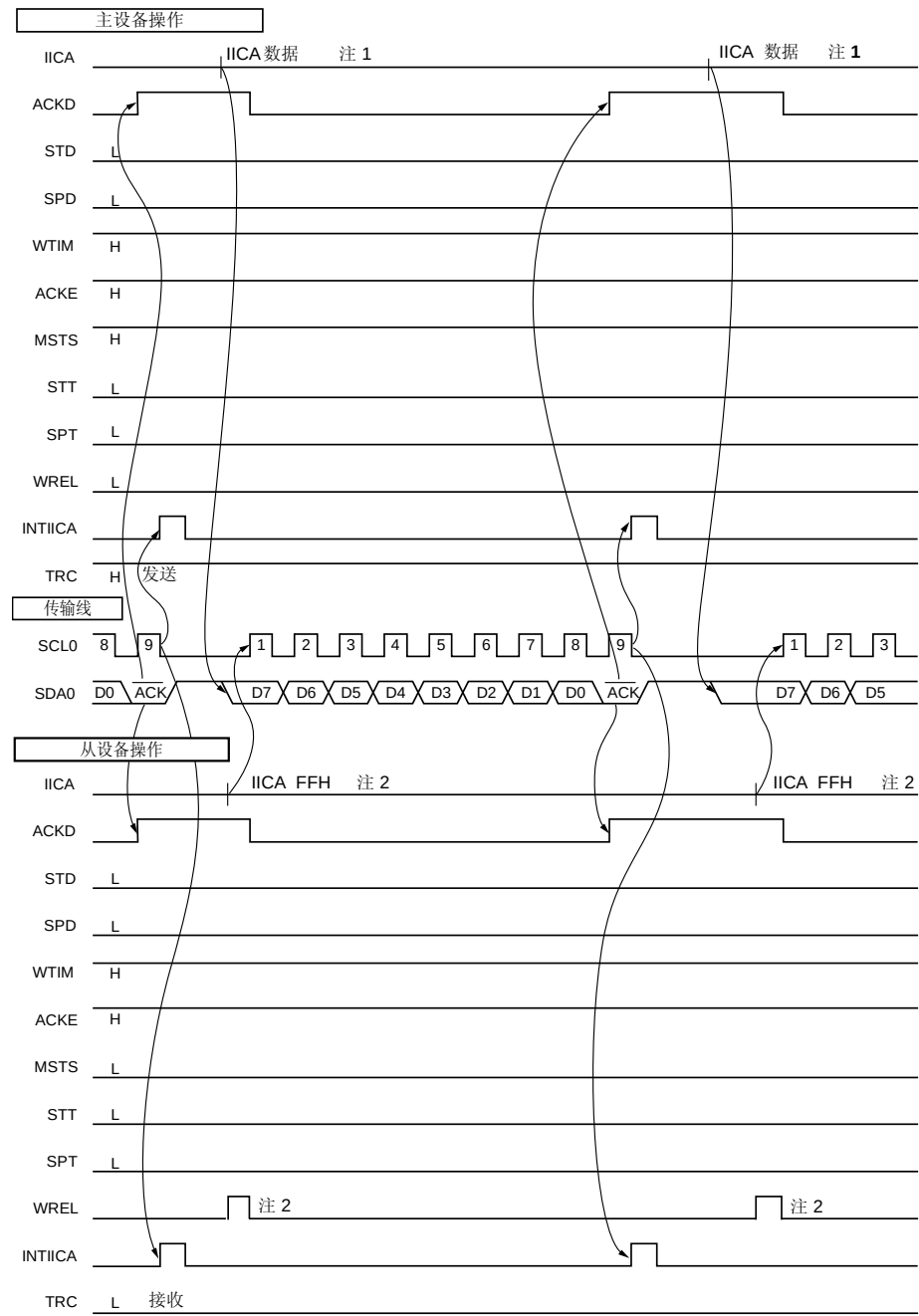
(1) 启动条件 ~ 地址



- 注
1. 将数据写入 IICA, 而不是设置 WREL, 可以在主发送期间取消一个等待状态。
 2. 取消从等待, 写入“FFH”至 IICA 或设置 WREL。

图 12-32. 主设备到从设备通信的例子
(当为每个主设备和从设备选择 9 个时钟等待时) (2/3)

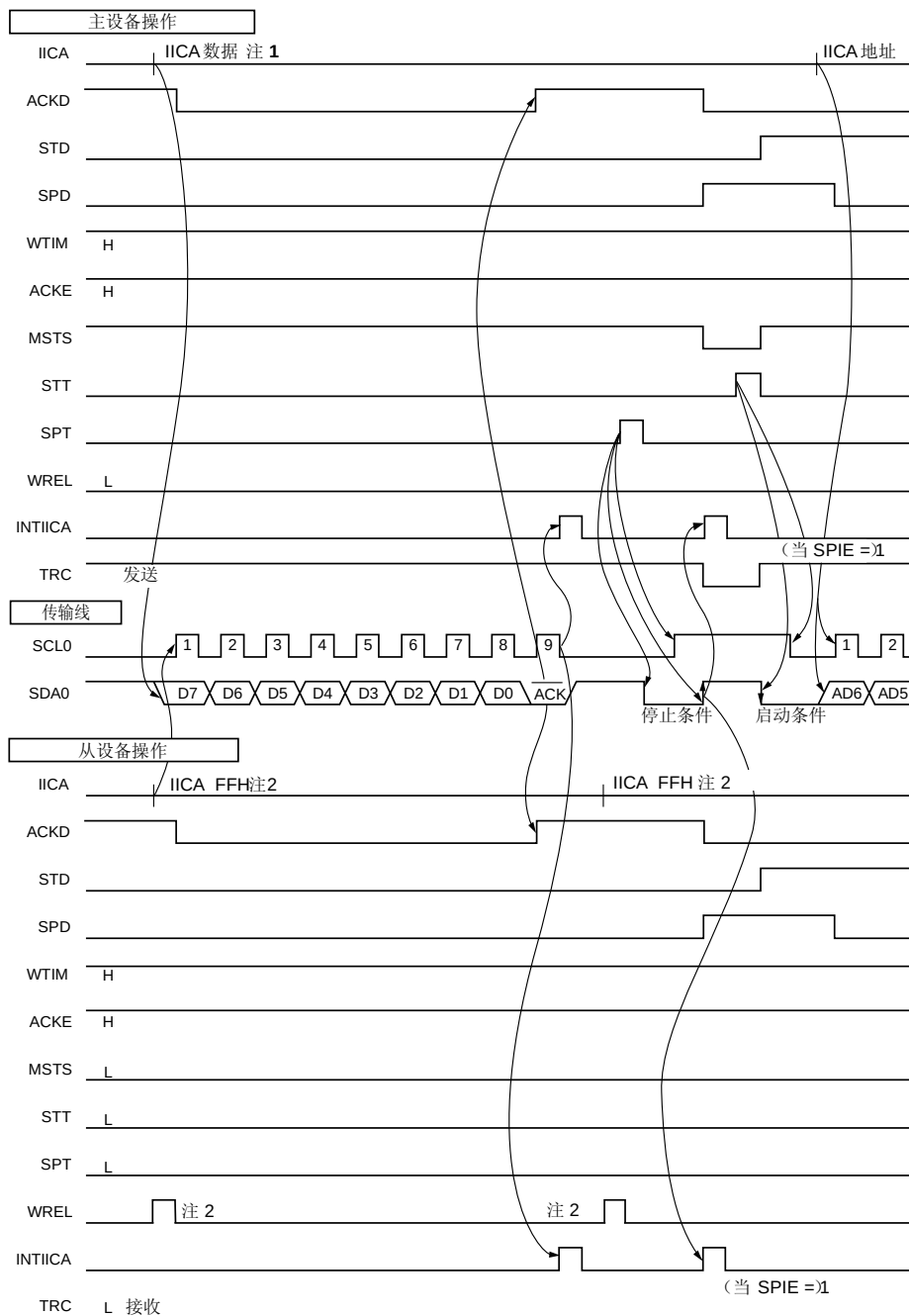
(2) 数据



- 注 1. 将数据写入 IICA，而不是设置 WREL，可以在主发送期间取消一个等待状态。
- 注 2. 取消从等待，写入“FFH”至 IICA 或设置 WREL。

图 12-32. 主设备到从设备通信的例子
(当为每个主设备和从设备选择 9 个时钟等待时) (3/3)

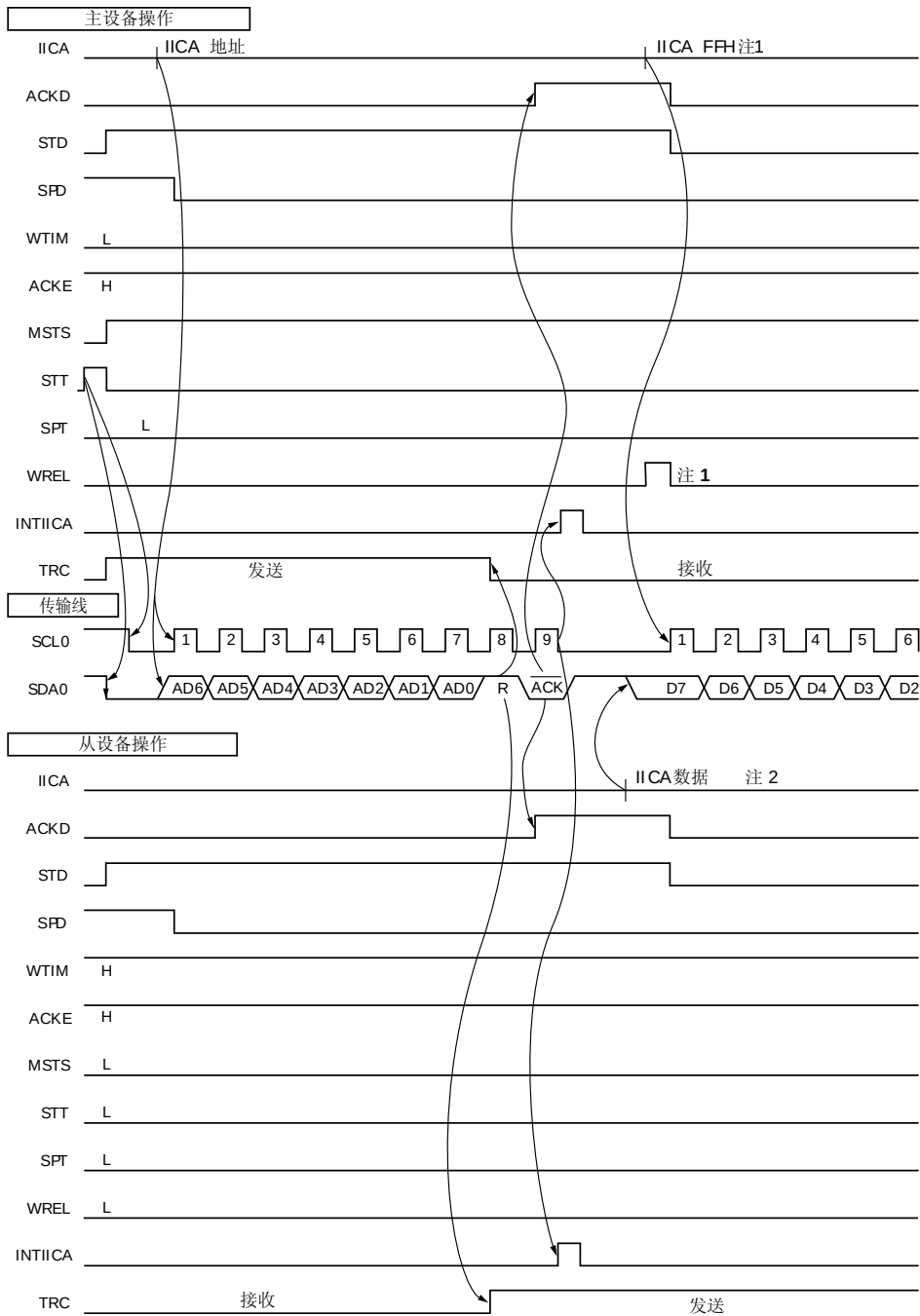
(3) 停止条件



- 注
1. 将数据写入 IICA，而不是设置 WREL，可以在主发送期间取消一个等待状态。
 2. 取消从等待，写入“FFH”至 IICA 或设置 WREL。

图 12-33. 主设备到从设备通信的例子
 (当为主设备选择 8 个时钟等待, 从设备选择 9 个时钟等待时) (1/3)

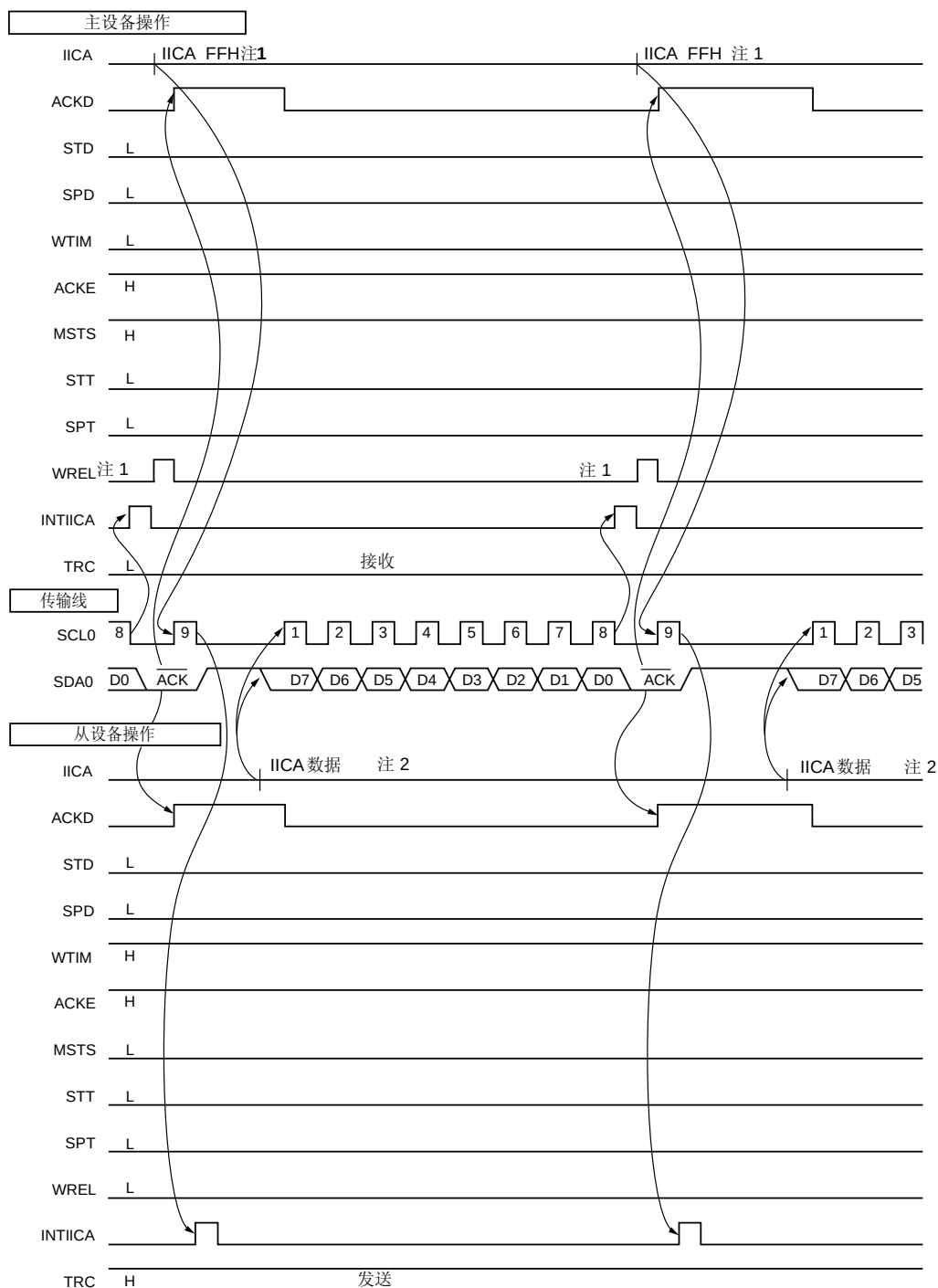
(1) 启动条件 ~ 地址



- 注
1. 取消主等待, 写入“FFH”至 IICA 或设置 WREL。
 2. 将数据写入 IICA, 而不是设置 WREL, 可以在从发送期间取消一个等待状态。

图 12-33. 主设备到从设备通信的例子
(当为主设备选择 8 个时钟等待，从设备选择 9 个时钟等待时) (2/3)

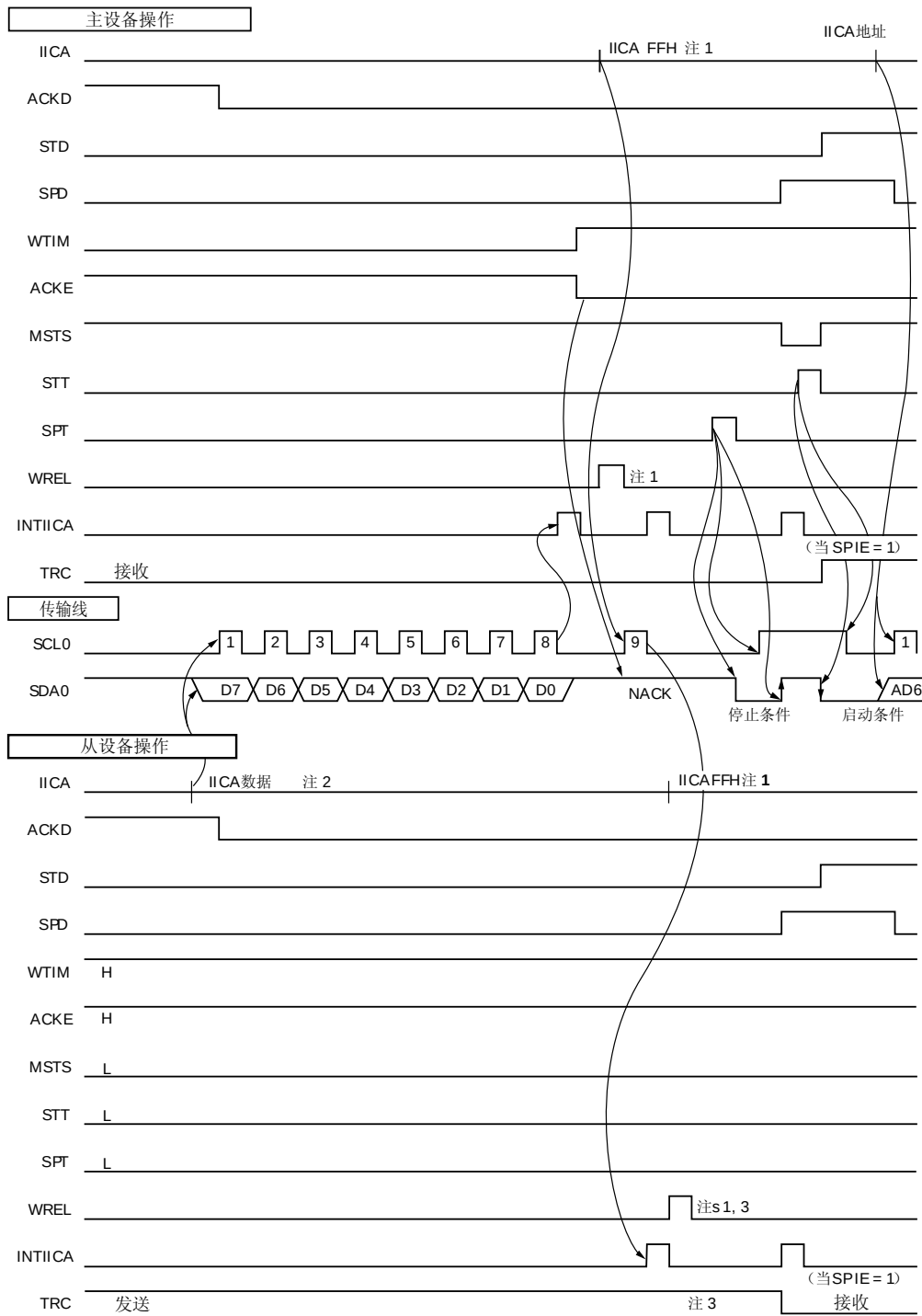
(2) 数据



- 注 1. 取消主等待，写入“FFH”至 IICA 或设置 WREL。
2. 将数据写入 IICA，而不是设置 WREL，可以在从发送期间取消一个等待状态。

图 12-33. 主设备到从设备通信的例子
(当为主设备选择 8 个时钟等待, 从设备选择 9 个时钟等待时) (3/3)

(3) 停止条件



- 注
1. 取消等待, 写入“FFH”至 IICA 或设置 WREL。
 2. 将数据写入 IICA, 而不是设置 WREL, 可以在从发送期间取消一个等待状态。
 3. 如果在从发送期间一个等待状态通过设置 WREL 而被取消的话, TRC 会被清零。

13.1 乘法器/除法器的功能

乘法器/除法器的功能如下。

- 16 位 × 16 位 = 32 位 (乘法)
- 32 位 ÷ 32 位 = 32 位, 32 位余数 (除法)

13.2 乘法器/除法器的配置

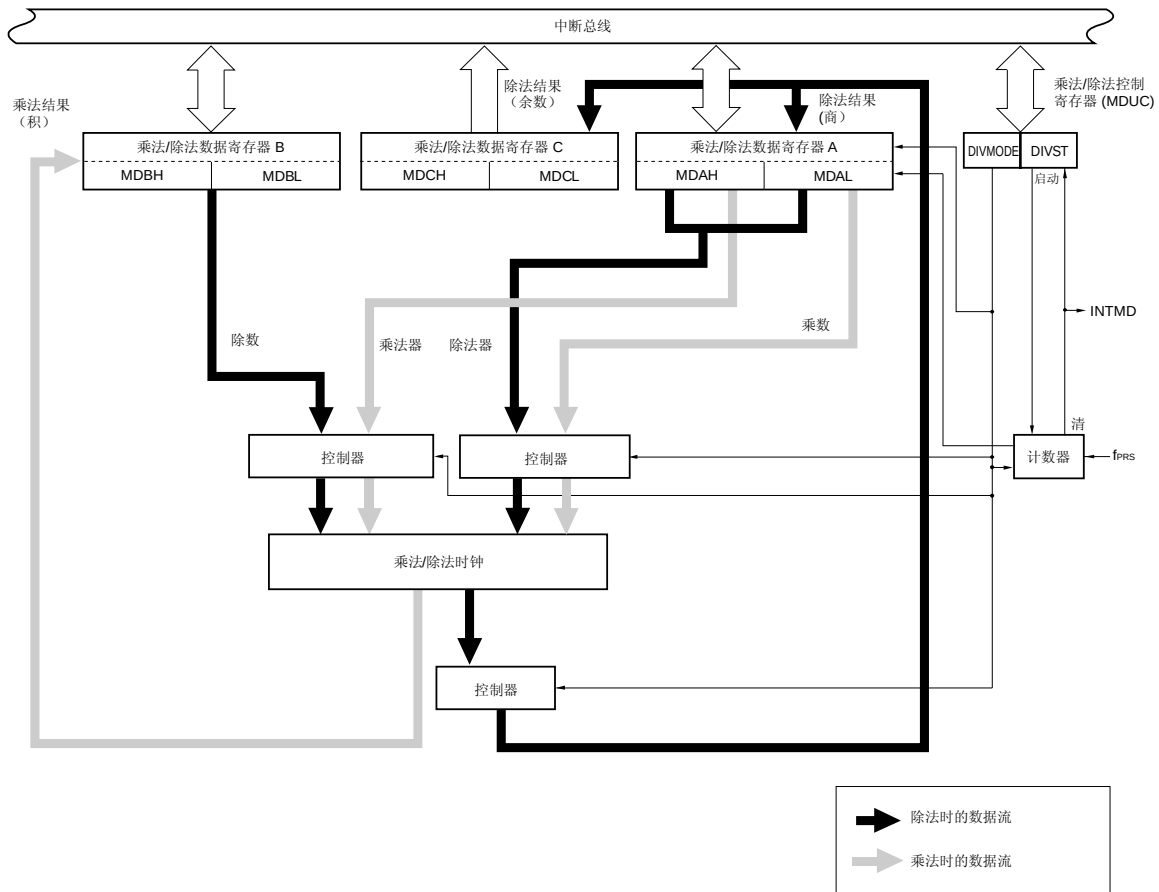
乘法器/除法器包含下列硬件。

表 13-1. 乘法器/除法器的配置

项目	配置
寄存器	乘法/除法数据寄存器 A (L) (MDAL) 乘法/除法数据寄存器 A (H) (MDAH) 乘法/除法数据寄存器 B (L) (MDBL) 乘法/除法数据寄存器 B (H) (MDBH) 乘法/除法数据寄存器 C (L) (MDCL) 乘法/除法数据寄存器 C(H)(MDCH)
控制寄存器	乘法/除法控制寄存器 (MDUC)

图 13-1 所示为乘法器/除法器的框图。

图 13-1. 乘法器/除法器的框图



(1) 乘法/除法数据寄存器 A (MDAH, MDAL)

MDAH 和 MDAL 寄存器设置的值用于乘法或除法运算并存放运算结果。在乘法模式下设置乘数和被乘数，在除法模式下设置被除数。另外，在除法模式下运算数据（商）存放在 MDAH 和 MDAL 寄存器中。

MDAH 和 MDAL 可以通过 16 位操作指令来设置。

复位信号将这些寄存器清零为 0000H。

图 13-2. 乘法/除法数据寄存器 A(MDAH, MDAL)的格式

地址: FFFF0H, FFFF1H, FFFF2H, FFFF3H 复位后: 0000H, 0000H R/W



- 注意事项
- 1. 在除法运算过程中不要重写 MDAH 和 MDAL(当乘法/除法控制寄存器(MDUC)为 81H 时)。此时运算将被执行，但是运算结果将是一个不确定值。
 - 2. 不保证在除法运算过程中(当 MDUC 为 81H 时)读取的 MDAH 和 MDAL 的值。

下表所示为执行运算时 MDAH 和 MDAL 的功能。

表 13-2. 执行运算时 MDAH 和 MDAL 的功能

DIVMODE	操作模式	设置	操作结果
0	乘法模式	MDAH: 乘数 MDAL: 被乘数	—
1	除法模式	MDAH: 除数（高 16 位） MDAL: 被除数（低 16 位）	MDAH: 除法结果（商） 高 16 位 MDAL: 除法结果（商） 低 16 位

备注 DIVMODE: 乘法/除法控制寄存器 (MDUC) 的第 7 位

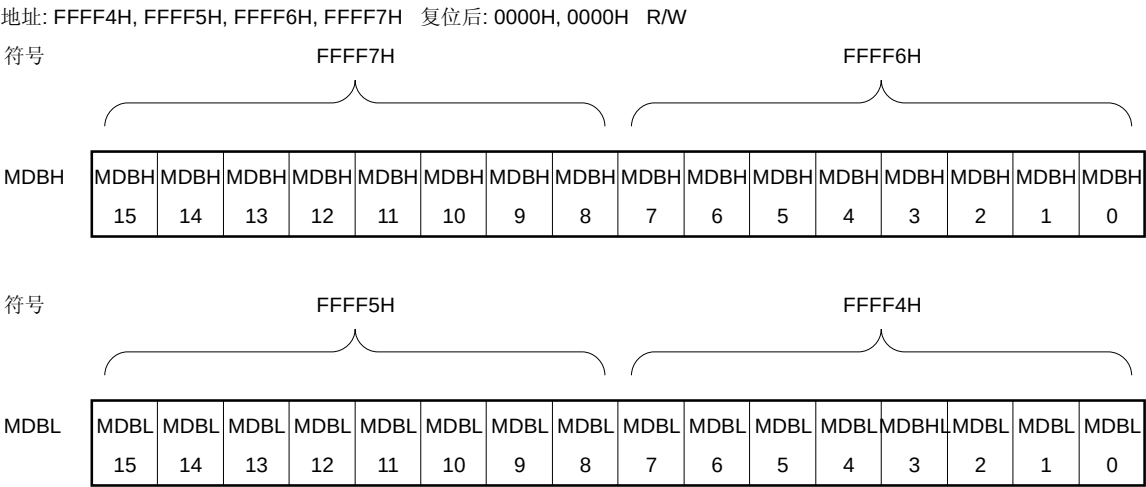
(2) 乘法/除法数据寄存器 B (MDBL, MDBH)

MDBH 和 MDBL 寄存器设置的值用于乘法或除法运算并存放运算结果。在乘法模式下存放运算结果（积），在除法模式下设置除数。

MDBH 和 MDBL 可以通过 16 位操作指令来设置。

复位信号将这些寄存器清零为 0000H。

图 13-3. 乘法/除法数据寄存器 B(MDBH,MDBL)的格式



- 注意事项
1. 在除法运算过程中不要重写 MDBH 和 MDBL (当乘法/除法控制寄存器 (MDUC) 为 81H 时)。运算结果将不确定。
 2. 在除法模式下不要将 MDBH 和 MDBL 设置为 0000H。否则运算结果将会是个不确定值。

下表所示为执行运算时 MDBH 和 MDBL 的功能。

表 13-3. 运算执行中 MDBH 和 MDBL 的功能

DIVMODE	操作模式	设置	操作结果
0	乘法模式	—	MDBH: 乘法结果 (积) 高 16 位 MDBL: 乘法结果 (积) 低 16 位
1	除法模式	MDBH: 除数 (高 16 位) MDBL: 被除数 (低 16 位)	—

备注 DIVMODE: 乘法/除法控制寄存器 (MDUC) 的第 7 位

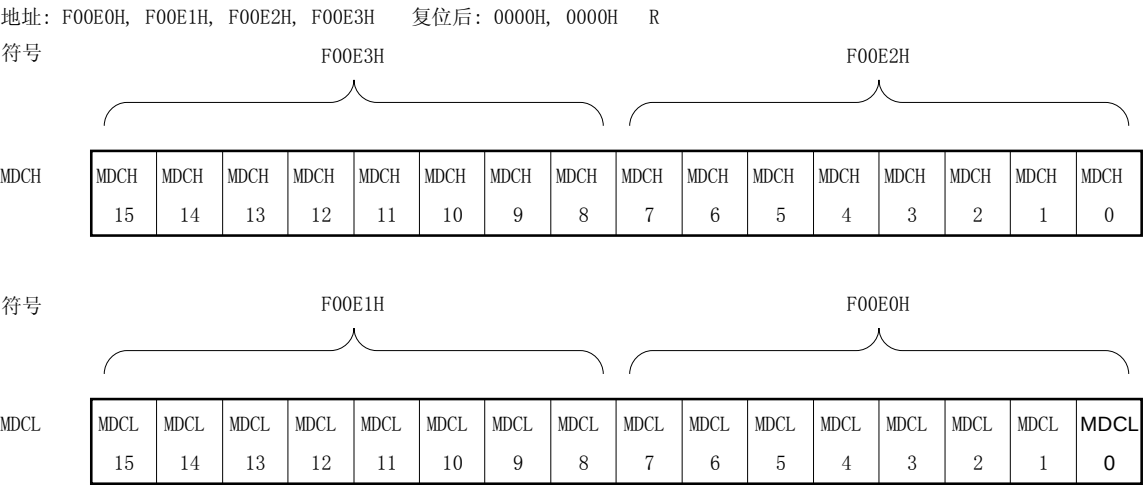
(3) 乘法/除法数据寄存器 C (MDCL, MDCH)

在除法模式下，MDCH 和 MDCL 寄存器存放运算结果的余数值。它们不可在乘法模式中使用。

MDCH 和 MDCL 可由 16 位操作指令读取。

复位信号将这些寄存器清零为 0000H。

图 13-4. 乘法/除法数据寄存器 C(MDCH, MDCL)的格式



注意事项 不保证在除法运算过程中(当乘法/除法控制寄存器 (MDUC) 为 81H 时) 读取的 MDCH 和 MDCL 的值。

表 13-4. 执行运算时 MDCH 和 MDCL 的功能

DIVMODE	操作模式	设置	操作结果
0	乘法模式	—	—
1	除法模式	—	MDCH: 余数 (高 16 位) MDCL: 余数 (低 16 位)

备注 DIVMODE: 乘法/除法控制寄存器 (MDUC) 的第 7 位

如下，根据执行乘法和执行除法不同寄存器的配置也有所不同。

- 进行乘法时的配置寄存器
 <乘数 A> <乘数 B> <积>
 MDAL (第 15 至 0 位) × MDAH (第 15 至 0 位) = [MDBH (第 15 至 0 位), MDBL (第 15 至 0 位)]
- 进行除法时的配置寄存器
 <被除数> <除数>
 [MDAH (第 15 至 0 位), MDAL (第 15 位至第 0 位)] ÷ [MDBH (第 15 至 0 位), MDBL (第 15 至 0 位)] =
 <商> <余数>
 [MDAH (位 15 至 0), MDAL (位 15 至 0)] ... [MDCH (位 15 至 0), MDCL (位 15 至 0)]

13.3 乘法器/除法器的寄存器

乘法器/除法器由乘法/除法控制寄存器 (MDUC) 进行控制。

(1) 乘法/除法控制寄存器 (MDUC)

MDUC 是一个控制乘法器/除法器运算的 8 位寄存器。
MDUC 可以由 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 00H。

图 13-5. 乘法/除法控制寄存器 (MDUC) 的格式

地址: F00E8H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	<0>
MDUC	DIVMODE	0	0	0	0	0	0	DIVST

DIVMODE	工作模式(乘法/除法)选择
0	乘法模式
1	除法模式

DIVST ^注	除法操作开始/停止
0	除法操作处理完成
1	除法运算开始/除法运算处理中

注 DIVST 在除法模式中只能设为(1)。在除法模式中，将 DIVST 设为(1)开始除法运算。当运算结束后，DIVST 自动清(0)。在乘法模式下，将乘数和被乘数分别设置到 MDAH 和 MDAL 后，运算自动开始。

- 注意事项
- 1. 在运算过程中不要重写 DIVMODE(当 DIVST 为 1 时)。如果被重写，运算结果将会是个不确定的值。
 - 2. 在除法运算过程中(当 DIVST 为 1 时)，DIVST 不能通过软件清(0)。

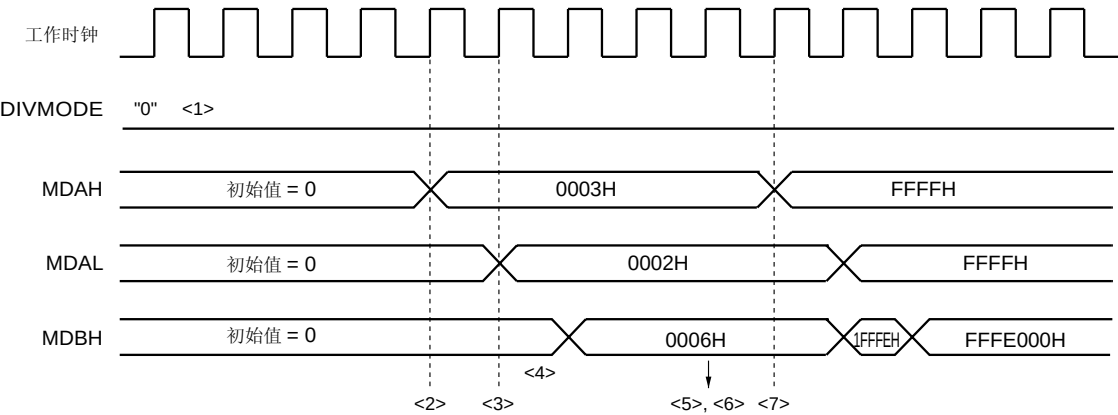
13.4 乘法器/除法器的操作

13.4.1 乘法运算

- 初始设置
 - <1> 设置乘法/除法控制寄存器 (MDUC) 的第 7 位 (DIVMODE) 为 0。
 - <2> 设置被乘数到乘法/除法数据寄存器 A (L) (MDAL)。
 - <3> 设置乘数到乘法/除法数据寄存器 A (H) (MDAH)。(执行步骤<2>和<3>的顺序时, 没有优先顺序。)当乘数和被乘数分别设置到 MDAH 和 MDAL 后, 乘法运算自动开始。
- 操作处理期间
 - <4> 等待至少一个时钟。一个时钟发布后, 操作将结束。
- 运算结束
 - <5> 从乘法 / 除法数据寄存器 B (L) (MDBL)中读取乘积(较低的 16 位)。
 - <6> 从乘法 / 除法数据寄存器 B (H) (MDBH)中读取乘积(较高的 16 位)。(执行步骤<5>和<6>的顺序没有任何优先顺序)
- 下一操作
 - <7> 下一步要执行乘法运算, 从乘法运算的"初始设置"开始。
 - <8> 要执行下一步除法操作, 从 13.4.2 除法运算的"初始设置"开始。

备注 步骤<1>至<7>与图 13-6 中的<1>至<7>相对应。

图 13-6. 乘法运算 (0003H × 0002H) 的时序图



13.4.2 除法操作

- 初始设置
 - <1> 设置乘法 / 除法控制寄存器 (MDUC) 的第 7 位 (DIVMODE) 为 1。
 - <2> 设置被除数 (较高的 16 位) 到乘法/除法数据寄存器 A (H) (MDAH)。
 - <3> 设置被除数 (较低的 16 位) 到乘法/除法数据寄存器 A (L) (MDAL)。
 - <4> 设置除数 (较高的 16 位) 到乘法/除法数据寄存器 B (H) (MDBH)。
 - <5> 设置除数 (较低的 16 位) 到乘法/除法数据寄存器 B (L) (MDBL)。
 - <6> 设置 MDUC 的位 0 (DIVST) 为 1。

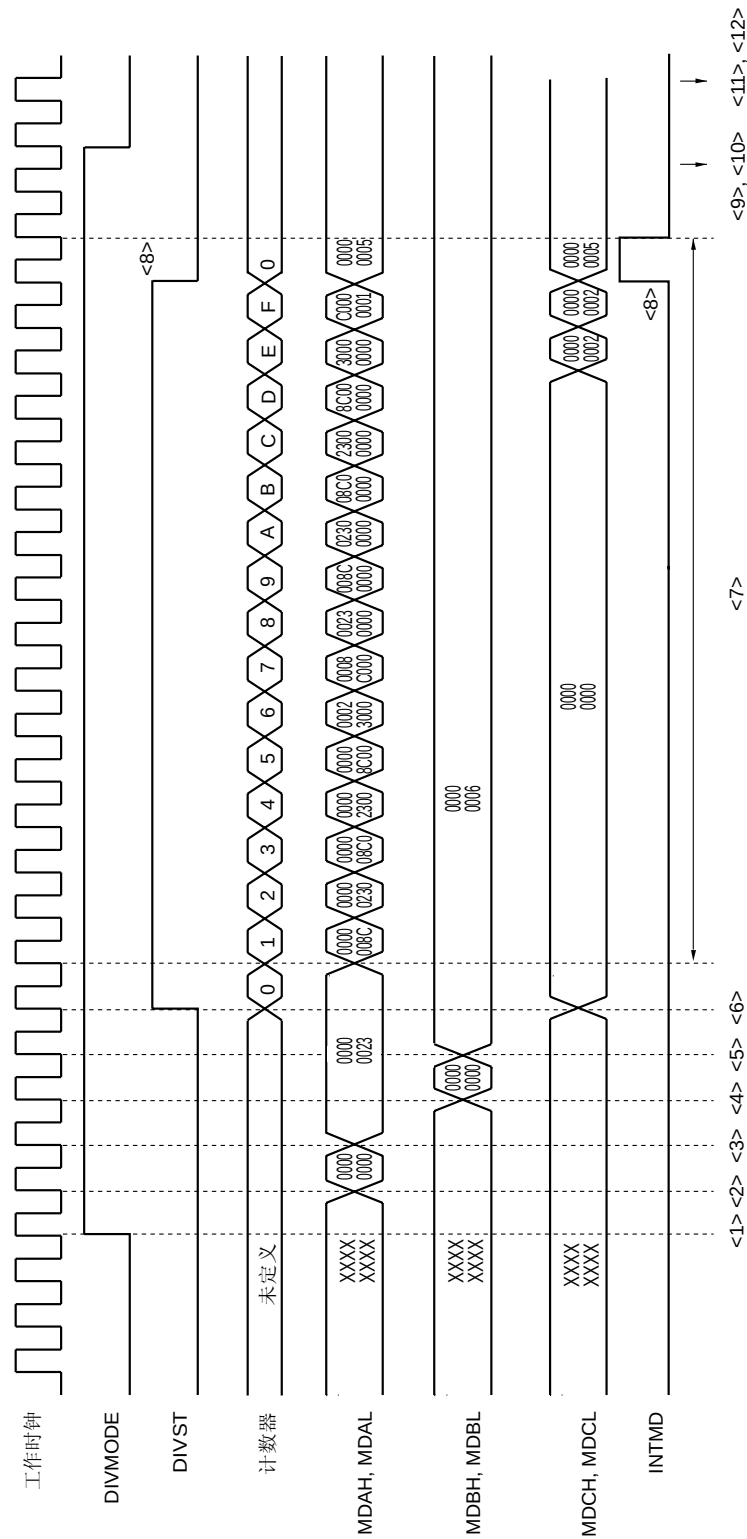
(执行步骤<2>至<5>的顺序时, 没有优先顺序)
- 操作处理期间
 - <7> 下列过程之一完成后, 运算将结束。
 - 至少等待 16 个时钟(当 16 个时钟发布后, 运算将结束。)
 - 检查 DIVST 是否被清除
 - 产生一个除法完成中断 (INTMD)

(运算过程中, MDBL、MDBH、MDCH 和 MDCL 的读取数值不被保证。)
- 运算结束
 - <8> DIVST 被清 (0) 且产生一个中断请求信号 (INTMD) (运算结束)。
 - <9> 从 MDAL 中读取商(较低的 16 位)。
 - <10> 从 MDAH 中读取商(较高的 16 位)。
 - <11> 从乘法/除法数据寄存器 C (L) (MDCL) 中读取余数(较低的 16 位)。
 - <12> 从乘法/除法数据寄存器 C (H) (MDCH) 中读取余数(较高的 16 位)。

(执行步骤<9>至<12>的顺序时, 没有优先顺序。)
- 下一操作
 - <13> 要执行下一步乘法运算, 从 13.4.1 乘法运算的"初始设置"开始。
 - <14> 下一步要执行除法运算, 从除法运算的"初始设置"开始。

备注 步骤<1>至<12>与图 13-7 中的<1>至<12>相对应

图 13-7. 除法运算 (示例: $35 \div 6 = 5$, 余数 5) 的时序图



第 14 章 DMA 控制器

78K0R/KG3-L 有一个内部 DMA(直接存储器访问)控制器。

数据可以自动在外围硬件支持 DMA 的 SFR 与内部 RAM 之间传输，而不经 CPU。

这样，CPU 的正常内部操作和数据传输能够与 SFR 和内部 RAM 之间的传输并行执行，因此能够处理大容量的数据。另外，使用通信的实时控制、定时器和 A/D 也可以实现。

14.1 DMA 控制器

- DMA 通道个数：2
- 传输单位：8 或 16 位
- 最大传输单元：1024 次
- 传输类型：2 周期传输(一次传输在 2 个时钟内操作，在操作期间，CPU 停止)
- 传输模式：单传输模式
- 传输请求：可从下列外围硬件中断中选择
 - A/D 转换器
 - 串行接口(CSI00, CSI01, CSI10, UART0, UART1, UART3 或 IIC10)
 - 定时器(通道 0, 1, 4, 或 5)
- 传输目标：在 SFR 内部 RAM 之间

这里是使用 DMA 的实例。

- 串行接口的连续传输
- 模拟数据的批传输
- 固定间隔捕捉 A/D 转换结果
- 固定间隔捕捉端口数值

14.2 DMA 控制器的配置

DMA 控制器包括下列硬件。

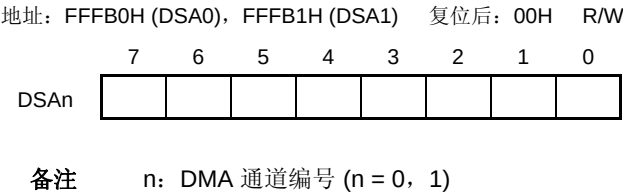
表 14-1. DMA 控制器的配置

项目	配置
地址寄存器	• DMA SFR 地址寄存器 0, 1 (DSA0, DSA1) • DMA RAM 地址寄存器 0, 1 (DRA0, DRA1)
计数寄存器	• DMA 字节计数寄存器 0, 1 (DBC0, DBC1)
控制寄存器	• DMA 模式控制寄存器 0, 1 (DMC0, DMC1) • DMA 操作模式控制寄存器 0, 1 (DRC0, DRC1)

(1) DMA SFR 地址寄存器 n (DSAn)

此为 8 位寄存器，用来设置 DMA 通道 n 的传输源或者目的地的 SFR 地址。
设置 SFR 地址的低八位 FFF00H 至 FFFFFH。
此寄存器并不自动增加，但固定为一个特定的数值。
16 位传输模式下，忽略最低位，并当作偶地址。
DSAn 可以进行 8 位读写。但是，DMA 传输期间不能写入。
复位后这个寄存器的值为 00H。

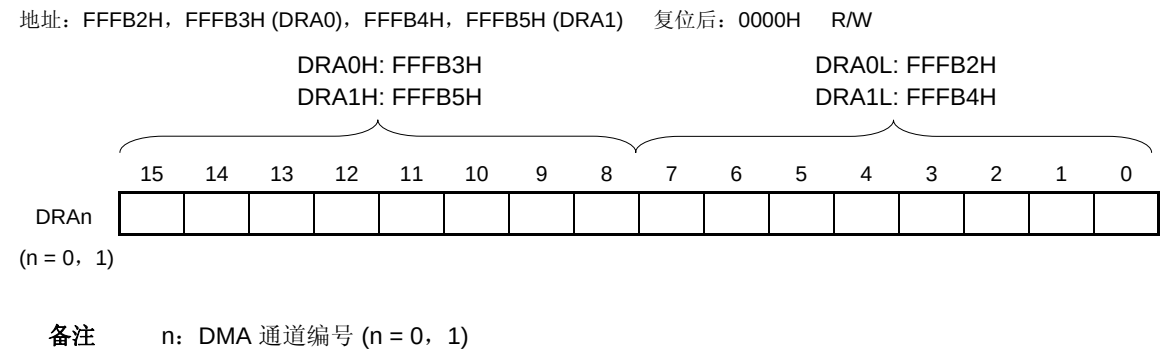
图 14-1. DMA SFR 地址寄存器 n (DSAn)的格式



(2) DMA RAM 地址寄存器 n (DRAn)

此为 16 位寄存器，用来设置 DMA 通道 n 的传输源或目的地的 RAM 地址。
除通用寄存器外(μPD78F1013 的情况下：FE700H 至 FFEDFH)，内部 RAM 区域的地址，可设置到该寄存器。
设置 RAM 地址的低 16 位。
此寄存器在 DMA 传输开始时自动增加。在 8 位传输模式下，+1 增加；在 16 位传输模式下，+2 增加。DMA 传输从设置在 DRAn 寄存器中的地址开始。当最后地址的数据传输完成时，DRAn 停止，其内部的值在 8 位传输模式下为最后地址+1，在 16 位传输模式下为最后地址+2。
16 位传输模式下，忽略最低位，并当作偶地址。
DRAn 可以进行 8 位或 16 位读写。但是，DMA 传输期间不能写入。
复位后这个寄存器的值为 0000H。

图 14-2. DMA RAM 地址寄存器 n (DRAn)的格式



(3) DMA 字节计数寄存器 n (DBCn)

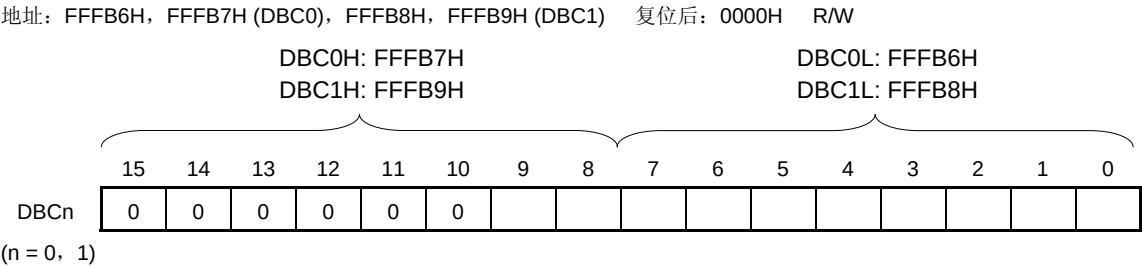
此为 10 位寄存器，用来设置 DMA 通道 n 执行传输的次数值。执行 DMA 传输之前，确保将传输的次数值设置在此 DBCn 寄存器中(最多 1024 次)。

每一次 DMA 传输执行完成，此寄存器自动减少。在 DMA 传输期间读取该 DBCn 寄存器，可知道传输的剩余次数。

DBCn 可以进行 8 位或 16 位读写。但是，DMA 传输期间不能写入。

复位后这个寄存器的值为 0000H。

图 14-3. DMA 字节计数寄存器 n (DBCn) 的格式



DBCn[9:0]	传输次数值 (DBCn 写入时)	剩下的次数值 (DBCn 读取时)
000H	1024	完成传输或者等待 1024 次 DMA 传输
001H	1	等待剩下的 1 次 DMA 传输
002H	2	等待剩下的 2 次 DMA 传输
003H	3	等待剩下的 3 次 DMA 传输
⋮	⋮	⋮
3FEH	1022	等待剩下的 1022 次 DMA 传输
3FFH	1023	等待剩下的 1023 次 DMA 传输

- 注意事项
1.

确保将位 15 至 10 清“0”。
2.

如果指定了通用寄存器或者连续传输超出了内部 RAM 空间，通用寄存器或者 SFR 空间将写入或者读取，导致这些空间的数据丢失。确保在内部 RAM 空间设置传输次数值。

备注 n: DMA 通道编号 (n = 0, 1)

14.3 控制 DMA 控制器的寄存器

DMA 控制器由下列寄存器控制。

- DMA 模式控制寄存器 n (DMCn)
- DMA 操作控制寄存器 n (DRCn)

备注 n: DMA 通道编号 (n = 0, 1)

(1) DMA 模式控制寄存器 n (DMCn)

DMCn 寄存器用来设置 DMA 通道 n 的传输模式。它用来选择传输方向、数据大小、挂起设置和启动源。第 7 位 (STGn) 是启动 DMA 的软件触发。

操作期间(当 DSTn = 1 时)，DMCn 的第 6 位、第 5 位和第 3 至第 0 位禁止重写。

DMCn 可由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 14-4. DMA 模式控制寄存器 n (DMCn) (1/2)的格式

地址: FFFBAH (DMC0), FFFBBH (DMC1) 复位后: 00H R/W

符号

	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn [#]	DMA传输开始软件触发
0	没有触发操作
1	当DMA操作允许时 (DENn = 1)，DMA传输开始。
当DMA操作允许时 (DENn = 1)，通过在此位写入1，开始DMA传输。 当读取此位时，一直为0。	

DRSn	选择DMA传输方向
0	SFR到内部RAM
1	内部RAM到SFR

DSn	制定DMA传输的传输数据大小
0	8位
1	16位

DWAITn	DMA传输的挂起
0	DMA开始请求之后，执行DMA传输(不保持挂起)。
1	如果有，保持DMA启动请求挂起状态。
挂起的DMA传输可以通过清除DWAITn数值为0来开始。当DWAITn数值设置为1时，有效的保持DMA传输挂起需要2个时钟。	

注 不管 IFCn0 至 IFCn3 为何数值，软件触发 (STGn) 都可以使用。

备注 n: DMA 通道编号 (n = 0, 1)

图 14-4. DMA 模式控制寄存器 n (DMCn) (2/2)的格式

地址: FFFBAH (DMC0), FFFBBH (DMC1) 复位后: 00H R/W

符号	<7>		<6>		<5>	<4>	3	2	1	0
DMCn	STGn		DRSn		DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

IFCn 3	IFCn 2	IFCn 1	IFCn 0	选择DMA开始源 [*]	
				触发信号	触发内容
0	0	0	0	—	通过中断禁止DMA传输。 (只允许软件触发。)
0	0	1	0	INTTM00	定时器通道0中断
0	0	1	1	INTTM01	定时器通道1中断
0	1	0	0	INTTM04	定时器通道4中断
0	1	0	1	INTTM05	定时器通道5中断
0	1	1	0	INTST0/INTCSI00	UART0发送结束中断/CSI00传输结束中断
0	1	1	1	INTSR0/INTCSI01	UART0接收结束中断/CSI01传输结束中断
1	0	0	0	INTST1/INTCSI10/INTIIC10	UART1发送结束中断/ CSI10传输结束中断/ IIC10传输结束中断
1	0	0	1	INTSR1	UART1接收结束中断
1	0	1	0	INTST3	UART3 发送结束中断
1	0	1	1	INTSR3	UART3接收结束中断
1	1	0	0	INTAD	A/D转换结束中断
其它值				设置禁止	

注 不管 IFCn0 至 IFCn3 为何数值，软件触发 (STGn) 都可以使用。

备注 n: DMA 通道编号 (n = 0, 1)

(2) DMA 操作控制寄存器 n (DRCn)

DRCn 寄存器用来禁止或允许 DMA 通道 n 的传输。
操作期间(当 DSTn = 1 时)，此寄存器的第 7 位 (DENn) 禁止重写。
DRCn 可由 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 00H。

图 14-5. DMA 操作控制寄存器 n (DRCn)的格式

地址: FFFBCH (DRC0), FFFBDH (DRC1) 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	<0>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA操作允许标志
0	禁止DMA通道n的操作(停止DMA的工作时钟)。
1	允许DMA通道n的操作。
DMA操作允许后 (DENn = 1)，当DSTn = 1时，DMA控制器等待1个DMA触发。	

DSTn	DMA传输模式标志
0	DMA通道n的DMA传输完成。
1	DMA通道n的DMA传输没有完成(仍在执行中)。
DMA操作允许后 (DENn = 1)，当DSTn = 1时，DMA控制器等待1个DMA触发。 当由IFCn3 至 IFCn0设置的软件触发 (STGn) 或开始源触发输入时，DMA传输开始。 当DMA传输完成之后，此位自动清零0。执行期间将此位置0，以强制中止DMA传输。	

- 注意事项
1. 当 DMA 传输完成时，DSTn 标志自动清除为 0。只有当 DSTn = 0 时，允许写入 DENn 标志。
当没有等待 DMAn 中断(INTDMA n)产生而中止 DMA 传输时，要设置 DSTn 为 0，然后设置 DENn 为 0(详细信息，参考 14.5.5 软件强行中止)。
 2. 当 OSMC 寄存器的 FSEL 位设为 1 时，在设置后的至少三个时钟内，不要允许 (DENn = 1) DMA 操作。

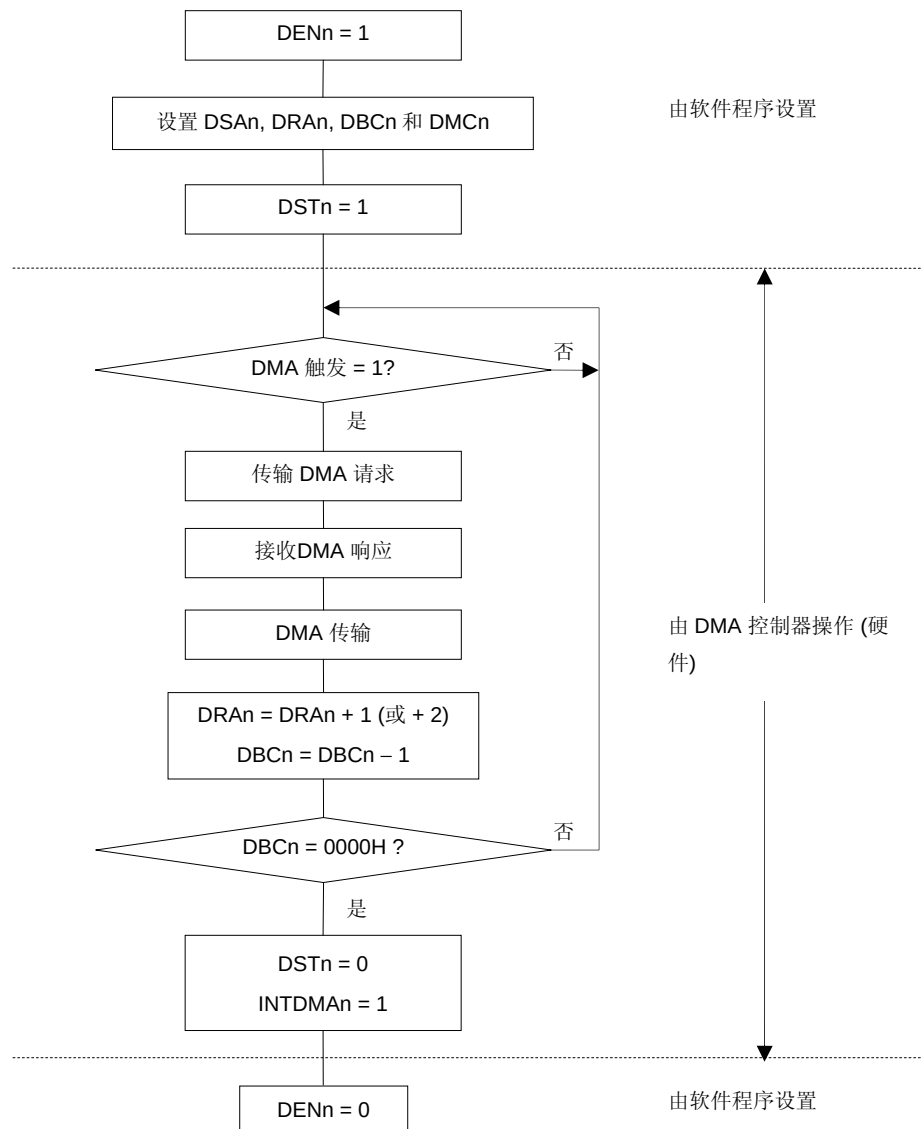
备注 n: DMA 通道编号 (n = 0, 1)

14.4 DMA 控制器的操作

14.4.1 操作过程

- <1> 当 DENn = 1 时，DMA 控制器操作允许。其它寄存器写入之前，确保将 DENn 设置为 1。使用 8 位操作指令写入 80H。
- <2> 设置 1 个 SFR 地址，1 个 RAM 地址，传输次数值，并且在 DSA_n, DRA_n, DBC_n 和 DMC_n 寄存器中设置 DMA 传输的传输模式。
- <3> 当 DST_n = 1 时，DMA 控制器等待 DMA 触发。使用 8 位操作指令写入 81H。
- <4> 当由 IFC_n3 至 IFC_n0 指定的软件触发 (STG_n) 或启动源触发输入时，DMA 传输开始。
- <5> 当由 DBC_n 寄存器设置的传输次数值变为 0 时，传输完成，并且产生一个中断 (INTDMA_n)，传输自动停止。
- <6> 当 DMA 控制器不使用时，通过将 DEN_n 清除为 0，来停止 DMA 控制器的操作。

图 14-6. 操作顺序



备注 n: DMA 通道编号 (n = 0, 1)

14.4.2 传输模式

下面 4 种模式通过使用 DMCn 寄存器的第 6 位和第 5 位(DRSn 和 DSn)选择 DMA 传输的模式。

DRSn	DSn	DMA传输模式
0	0	从1字节数据的SFR(固定地址)到RAM(地址通过+1增加)的传输
0	1	从2字节数据的SFR(固定地址)到RAM(地址通过+2增加)的传输
1	0	从1字节数据的RAM(地址通过+1增加)到SFR(固定地址)传输
1	1	从2字节数据的RAM(地址通过+2增加)到SFR(固定地址)传输

通过这些传输模式，最多 1024 个字节的数据可以通过串行接口连续传输，A/D 转换的结果数据也可以连续传输，并且可以通过使用定时器在固定的时间间隔内扫描端口数据。

14.4.3 DMA 传输的终止

DBCn = 00H 并且 DMA 传输完成时，DSTn 位自动清除为 0。中断请求 (INTDMA_n) 产生，并且传输中止。

当 DSTn 位清除为 0 以强制中止 DMA 传输，DBCn 和 DRAn 寄存器在传输中止时保持其值。

如果传输强行中止，中断请求 (INTDMA_n) 不产生。

备注 n: DMA 通道编号 (n = 0, 1)

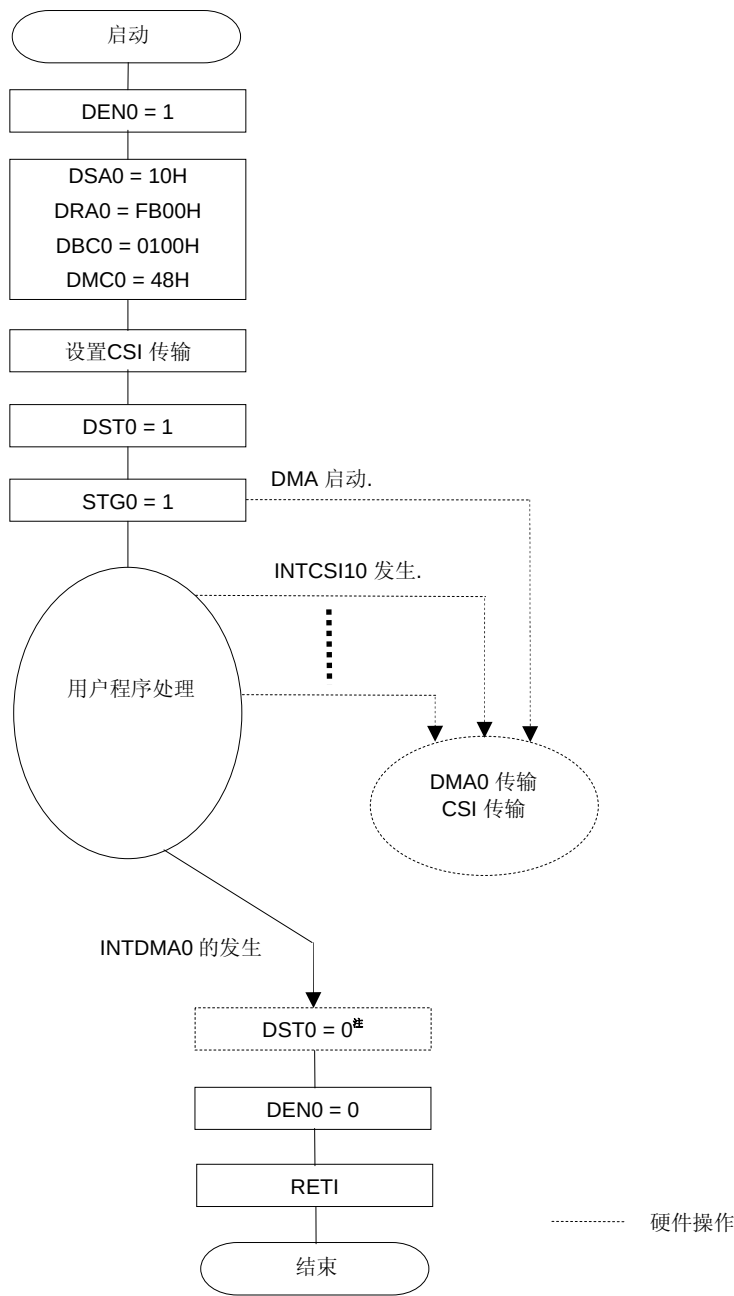
14.5 DMA 控制器设置的实例

14.5.1 CSI 连续发送

用于 CSI 连续发送的设置实例的流程图如下所示。

- CSI10 的连续发送
- DMA 通道 0 用于 DMA 传输。
- DMA 启动源: INTCSI10(软件触发 (STG0) 只用于第一个启动源)
- CSI10 的中断由 IFC03 至 IFC00 指定(DMC0 寄存器的第 3 位至第 0 位)= 1000B。
- 传输 RAM 的 FFB00H 至 FFBFFH(256 字节)到 CSI 的发送缓冲器 (SIO10) 的 FFF10H。

图 14-7. CSI 连续发送的设置实例



注 当 DMA 传输完成时，DST0 标志自动清除为 0。
只有当 DST0 = 0 时，DEN0 标志允许。为了中止 DMA 传输，而不等待 DMA0 (INTDMA0) 中断产生，把 DST0 设置为 0，然后把 DEN0 也设置为 0 (详细信息，参考 14.5.5 软件强行中断)。

连续发送的第一次触发并不由 CSI 中断启动。要由软件触发启动。

第二次的 CSI 发送和以后的发送自动执行。

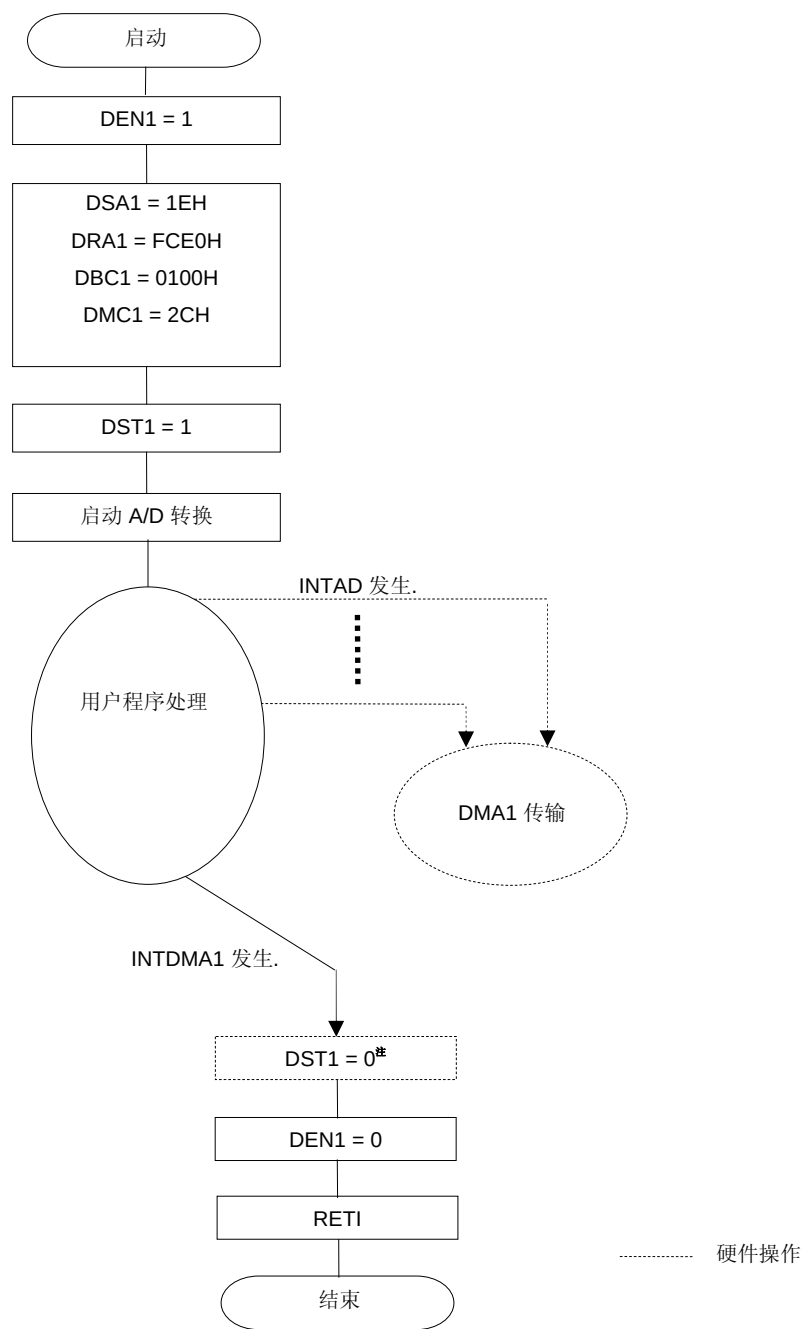
一旦最后的数据写入发送缓冲器，DMA 中断 (INTDMA0) 马上产生。此时，CSI 的最后数据正在发送。为了再次开始 DMA 传输，因此要一直等到 CSI 传输完成。

14.5.2 A/D 转换结果的连续捕捉

表示 A/D 转换结果连续捕捉设置实例的流程图如下所示。

- A/D 转换结果的连续捕捉。
- DMA 通道 1 用于 DMA 传输。
- DMA 启动源：INTAD
- A/D 中断由 IFC13 至 IFC10(DMC1 寄存器的第 3 位至第 0 位)= 1100B 来指定。
- 把 10 位 A/D 转换结果寄存器的 FFF1EH 和 FFF1FH(2 个字节)传输到 RAM 的 FFCE0H 至 FFEDFH 的 512 个字节中。

图 14-8. A/D 转换结果连续捕捉设置的实例



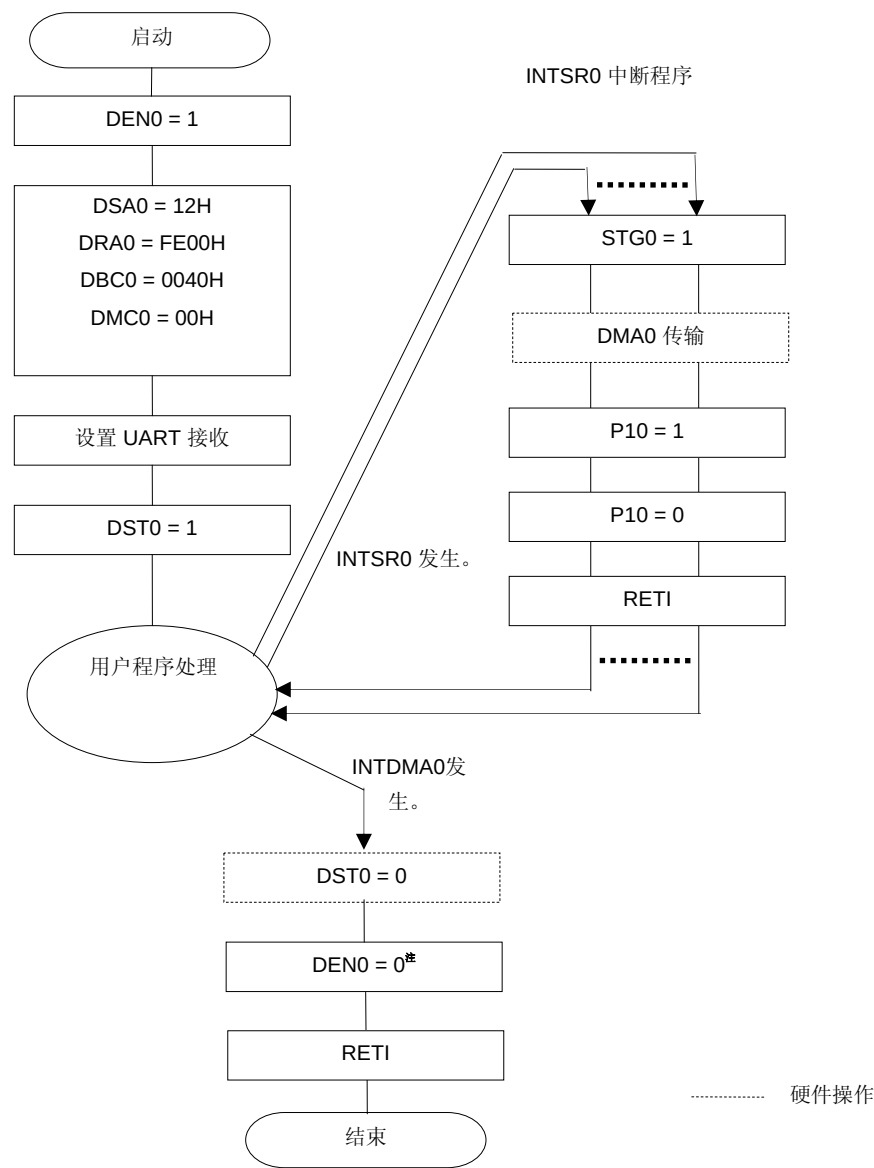
注 当 DMA 传输完成时，DST1 标志自动清除为 0。
只有当 DST1 = 0 时，DEN1 标志允许。为了中止 DMA 传输，而不等待 DMA1 (INTDMA1)中断产生，把 DST1 设置为 0，然后把 DEN1 也设置为 0 (详细信息，参考 14.5.5 软件强行中止)。

14.5.3 UART 连续接收+ACK 发送

表示 UART 连续接收+ACK 发送设置实例的流程图如下所示。

- 从 UART0 连续接收数据，并在接收完成之后输出 ACK 给 P10。
- DMA 通道 0 用于 DMA 传输。
- DMA 启动源：软件触发(禁止中断产生时的 DMA 传输。)
- 传输 UART 接收数据寄存器 0 (RXD0) 的 FFF12H 到 RAM 的 FFE00H 至 FFE3FH 的 64 字节。

图 14-9. UART 连续接收+ACK 发送设置的实例



注 当 DMA 传输完成时，DST0 标志自动清除为 0。
只有当 DST0 = 0 时，DEN0 标志允许。为了中止 DMA 传输，而不等待 DMA0 (INTDMA0) 中断产生，把 DST0 设置为 0，然后把 DEN0 也设置为 0 (详细信息，参考 14.5.5 软件强行中断)。

备注 此为一个软件触发用作 DMA 启动源的实例。
如果不发送 ACK，并且只有数据从 UART 中连续接收，那么 UART 接收结束中断 (INTSR0) 可以用于启动 DMA 用于数据接收。

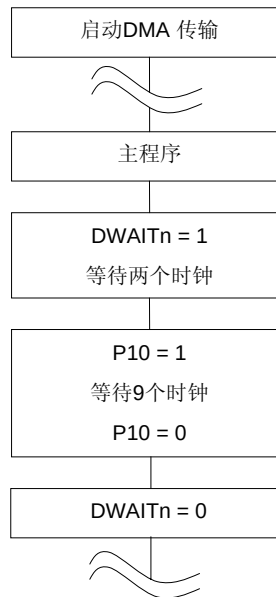
14.5.4 通过 DWAITn 保持 DMA 传输挂起

启动 DMA 传输时，指令执行期间传输进行。此时，CPU 的操作停止，并且延迟 2 个时钟。如果这对设定系统的操作造成了问题，DMA 传输可以通过设置 DWAITn 为 1 来保持挂起。

例如，为了从 P10 引脚输出脉宽为 10 个操作频率的时钟的脉冲，如果 DMA 传输中途开始，那么脉冲宽度将增加到 12。这种情况下，DMA 传输可以通过设置 DWAITn 为 1 来保持挂起。

设置 DWAITn 为 1 后，保持 DMA 传输挂起需要 2 个时钟。

图 14-10. 通过 DWAITn 保持 DMA 传输挂起设置的实例



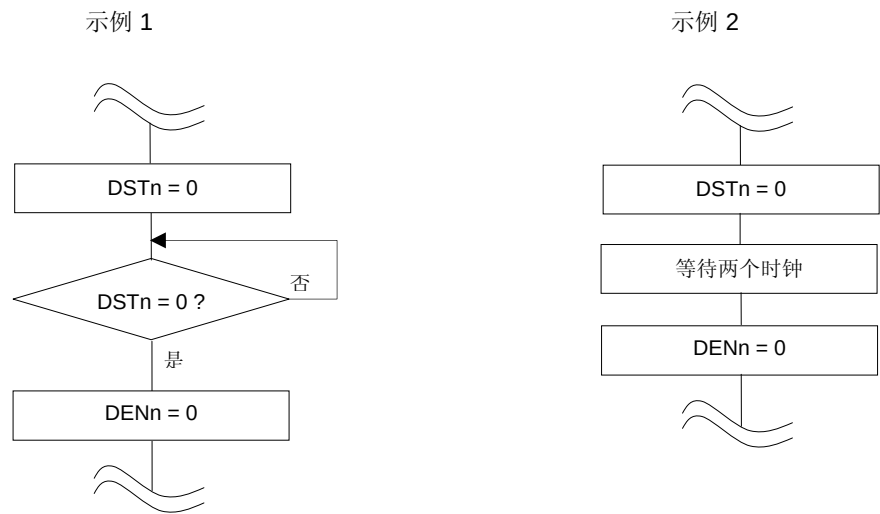
- 备注
1. n: DMA 通道编号 (n = 0, 1)
 2. 1 个时钟: $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

14.5.5 软件强行中止

DSTn 由软件设置为 0 之后，DMA 传输需要 2 个时钟才能真正停止并且 DSTn 位设置为 0。为了由软件强行中止 DMA 传输，而不等待 DMAn 中断 (INTDMA_n) 的产生，需要执行下列的操作。

- 由软件设置 DSTn 为 0(用 8 位操作指令写入 DRCn = 80H)，通过查询确认 DSTn 已清除为 0，然后设置 DENn 为 0(使用 DRCn = 00H 以 8 位操作指令写入)。
- 由软件设置 DSTn 为 0(用 8 位操作指令写入 DRCn = 80H)，然后在 2 个或者更多个时钟之后设置 DENn 为 0(使用 DRCn = 00H 以 8 位操作指令写入)。

图 14-11. DMA 传输的强行中止



- 备注
1. n: DMA 通道编号 (n = 0, 1)
 2. 1 个时钟: $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

14.6 使用 DMA 控制器的注意事项

(1) DMA 的优先级

DMA 传输期间，即使从其它 DMA 通道来的请求已产生，也保持挂起。挂起的 DMA 传输在当前的 DMA 传输完成之后开始。在一段较短的时间内^注两个 DMA 通道的请求依次产生，并依次传输，当一个 DMA 通道的请求完成之后，另一个的请求就执行。这种情况下，要在第一个 DMA 传输和接着的 DMA 传输之间执行 1 到 2 条指令。

但是，如果需要同时产生 2 个 DMA 请求，DMA 通道 0 的优先级高于 DMA 通道 1。

如果 DMA 请求和中断请求同时产生，DMA 传输优先，然后执行中断服务程序。

注 较短的时间是指 8 个或者更少的 CPU 时钟。时钟长度和 DMA 操作之间的关系如下。

- 1 个时钟周期：禁止设置 DMA 请求不能接受。
- 2 至 4 个时钟周期：请求连续产生的那个通道的 DMA 传输执行。
执行。
- 5 至 8 个时钟周期：是执行成功生成请求进行通道的 DMA 传输
还是执行其他通道的 DMA 请求，取决于 CPU 指令的执行次数。

(2) DMA 响应时间

DMA 传输的响应时间如下。

表 14-2. DMA 传输的响应时间

	最短时间	最长时间
响应时间	2个时钟 ^注	10个时钟 ^注

注 不包括 DMA 传输的两个时钟

备注 1 个时钟：1/fCLK (fCLK: CPU 时钟)

但是，下列情况下，DMA 传输可能会进一步延迟。DMA 传输延迟的时钟个数依据条件不同而不同。

- 由 RAM 执行指令
- 执行 DMA 挂起指令

(3) 待机模式下的操作

DMA 控制器的待机模式按照下列所示进行。

表 14-3. 待机模式下 DMA 的操作

状态	DMA操作
HALT模式	正常操作
STOP模式	停止操作 如果DMA传输和STOP指令执行冲突，那么DMA传输可能破坏。因此，执行STOP指令之前，要停止DMA。

(4) DMA 挂起指令

下列指令之后，即使 DMA 请求产生，DMA 传输也要立即保持挂起。

- CALL !addr16
- CALL &!addr16
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- 分别用于寄存器 IF0L、IF0H、IF1L、IF1H、IF2L、IF2H、MK0L、MK0H、MK1L、MK1H、MK2L、MK2H、PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H 和 PSW 的位操作指令，以及以 ES 寄存器为操作数的 8 位操作指令

(5) 如果指定了通用寄存器区域或者内部 RAM 之外的区域时的操作

在 DMA 传输期间，由 DRA0n 指示的地址增加。如果地址增加到通用寄存器区域，或者超过了内部 RAM 区域，下列操作将执行。

- 在从 SFR 到 RAM 传输的模式下
地址数据丢失。
- 在从 RAM 到 SFR 传输的模式下
不确定数据传输到 SFR。

任何一种情况下，都会发生误操作或者对系统造成损坏。因此，确保地址在内部 RAM 区域而不是在通用寄存器区域。



第 15 章 中断功能

15.1 中断功能的类型

可分为以下两种类型。

(1) 可屏蔽中断

这类中断可进行屏蔽控制。通过设置优先级指定标志寄存器

(PR00L,PR00H,PR01L,PR01H,PR02L,PR02H,PR10L,PR10H,PR11L,PR11H、PR12L、PR12H) 将可屏蔽中断分为四个优先级组。

支持中断嵌套，即在低优先级的中断服务子程序中可以响应高优先级的中断请求。如果同时产生两个或两个以上具有相同优先级的中断请求，则根据向量中断服务的优先级进行处理。优先级顺序，参见表 15-1。

产生一个待机释放信号，并释放 STOP 和 HALT 模式。

可屏蔽中断包括外部中断请求和内部中断请求。

外部：13，内部：33

(2) 软件中断

这是通过执行 BRK 指令产生的一类向量中断。即使禁止中断时也可以响应这类中断。软件中断不受中断优先级控制。

15.2 中断源和配置

78K0R/KG3-L 系列中，共有 47 种中断源，包括可屏蔽的中断和软件中断。另外，还具有 5 种复位源(参见表 15-1)。存放程序启动地址的向量代码当转向由于产生复位或多个中断请求时各为 2 字节，因此中断跳至 64K 地址 00000H 至 0FFFFH。

表 15-1. 中断源列表(1/3)

中断类型	默认优先级 ^{注 1}	中断源		内部/ 外部	向量表 地址	基本配置 类型 ^{注 2}
		名称	触发器			
可屏蔽的	0	INTWDTI	看门狗定时器间隔定时 ^{注 3} (溢出时间的 75%)	内部	0004H	(A)
	1	INTLVI	低电压检测 ^{注 4}		0006H	
	2	INTP0	引脚输入沿检测	外部	0008H	(B)
	3	INTP1			000AH	
	4	INTP2			000CH	
	5	INTP3			000EH	
	6	INTP4			0010H	
	7	INTP5			0012H	
	8	INTST3	UART3 发送传输结束或缓冲空中断	内部	0014H	(A)
	9	INTSR3	UART3 接收传输结束		0016H	
	10	INTSRE3	UART3 接收发生通信错误		0018H	
	11	INTDMA0	DMA0 传输结束		001AH	
	12	INTDMA1	DMA1 传输结束		001CH	
	13	INTST0 /INTCSI00	UART0 发送传输结束或缓冲空中断/CSI00 传输 中断或缓冲空中断		001EH	
	14	INTSR0 /INTCSI01	UART0 接收传输结束/CSI01 传输结束或缓冲清 空中断		0020H	
	15	INTSRE0	UART0 接收发生通信错误		0022H	
	16	INTST1 /INTCSI10 /INTIIC10	UART1 发送传输结束或缓冲清空中断/CSI10 传 输结束或缓冲空中断/IIC10 传输结束		0024H	
	17	INTSR1	UART1 接收传输结束		0026H	
	18	INTSRE1	UART1 接收发生通信错误		0028H	
	19	INTIICA	IICA 通讯结束		002AH	
	20	INTTM00	定时器阵列单元 0 的通道 0 计数或捕捉结束		002CH	
	21	INTTM01	定时器阵列单元 0 的通道 1 计数或捕捉结束		002EH	
	22	INTTM02	定时器阵列单元 0 的通道 2 计数或捕捉结束		0030H	
	23	INTTM03	定时器阵列单元 0 的通道 3 计数或捕捉结束		0032H	

- 注
1. 当两个或两个以上可屏蔽中断同时产生时，采用默认的优先级来决定向量中断的处理顺序。优先级 0 的优先级别最高，而优先级 45 的优先级别最低。
 2. 基本配置类型 (A)至(D) 与图 15-1 中 (A)至 (D)相对应。
 3. 当选项字节 (000C0H) 第 7 位 (WDTINT) 置 1 时。
 4. 当低电压检测寄存器 (LVIM) 的第 1 位(LVIMD)清零时。

表 15-1. 中断源列表(2/3)

中断类型	默认优先级 ^{注1}	中断源		内部/ 外部	向量表 地址	基本配置类型 ^{注2}
		名称	触发器			
可屏蔽的	24	INTAD	A/D 转换结束	内部	0034H	(A)
	25	INTRTC	实时计数器的固定周期信号/报警匹配检测		0036H	
	26	INTRTCI	实时计数器的间隔信号检测		0038H	
	27	INTKR	按键返回信号检测	外部	003AH	(C)
	28	INTST2 /INTCSI20 /INTIIC20	UART2 发送传输结束或缓冲清空中断/CSI20 传输结束或缓冲清空中断/IIC20 传输结束	内部	003CH	(A)
	29	INTP6	引脚输入沿检测	外部	003EH	(B)
	30	INTTM13	定时器阵列单元 1 的通道 3 计数或捕捉结束	内部	0040H	(A)
	31	INTTM04	定时器阵列单元 0 的通道 4 计数或捕捉结束		0042H	
	32	INTTM05	定时器阵列单元 0 的通道 5 计数或捕捉结束		0044H	
	33	INTTM06	定时器阵列单元 0 的通道 6 计数或捕捉结束		0046H	
	34	INTTM07	定时器阵列单元 0 的通道 7 计数或捕捉结束		0048H	
	35	INTSR2	UART2 接收传输结束		004AH	
	36	INTP7	引脚输入沿检测	外部	004CH	(B)
	37	INTP8			004EH	
	38	INTP9			0050H	
	39	INTP10			0052H	
	40	INTP11			0054H	
	41	INTTM10	定时器阵列单元 1 的通道 0 计数或捕捉结束	内部	0056H	(A)
	42	INTTM11	定时器阵列单元 1 的通道 1 计数或捕捉结束		0058H	
	43	INTTM12	定时器阵列单元 1 的通道 2 计数或捕捉结束		005AH	

- 注
1. 当两个或两个以上可屏蔽中断同时产生时，采用默认的优先级来决定向量中断的处理顺序。优先级 0 的优先级别最高，而优先级 45 的优先级别最低。
 2. 基本配置类型 (A)至(D) 与图 15-1 中 (A)至 (D)相对应。

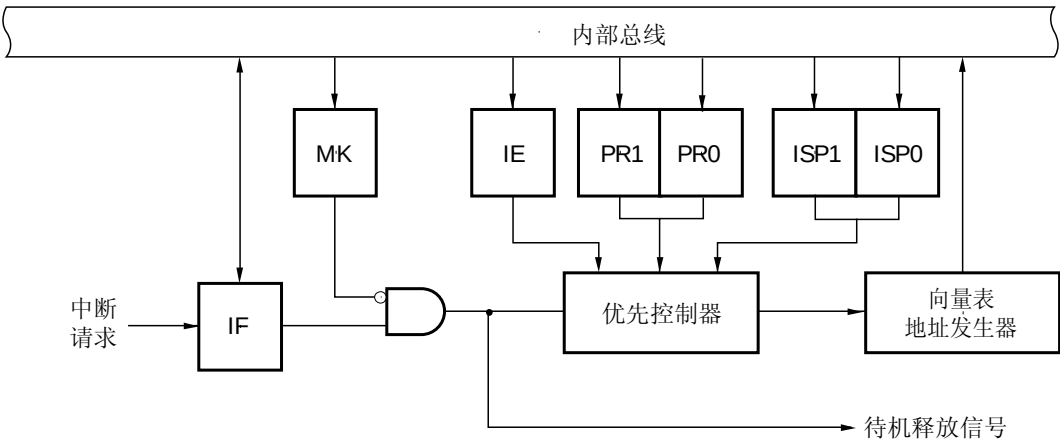
表 15-1. 中断源列表(3/3)

中断类型	默认优先级 ^{注 1}	中断源		内部/ 外部	向量表 地址	基本配置 类型 ^{注 2}
可屏蔽的	44	INTSRE2	UART2 接收发生通信错误	内部	005CH	(A)
	45	INTMD	除法操作结束		005EH	
软件	—	BRK	BRK 指令执行	—	007EH	(D)
复位	—	RESET	RESET 引脚输入	—	0000H	—
		POC	上电清零			
		LVI	低电压检测 ^{注 3}			
		WDT	看门狗定时器溢出			
		TRAP	非法指令执行 ^{注 4}			

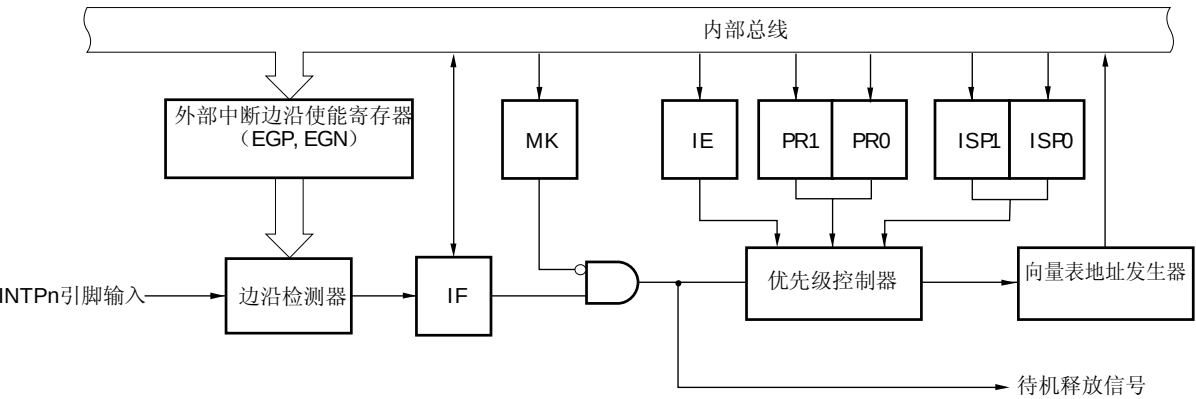
- 注
- 1. 当两个或两个以上可屏蔽中断同时产生时，采用默认的优先级来决定向量中断的处理顺序。优先级 0 的优先级别最高，而优先级 45 的优先级别最低。
 - 2. 基本配置类型 (A)至(D) 与图 15-1 中 (A)至 (D)相对应。
 - 3. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 设置为 1 时。
 - 4. 当执行 FFH 中的指令代码时。
由非法指令执行引起的复位，不能通过在线仿真器或片上调试仿真器终止。

图 15-1. 中断功能的基本配置(1/2)

(A) 内部可屏蔽中断



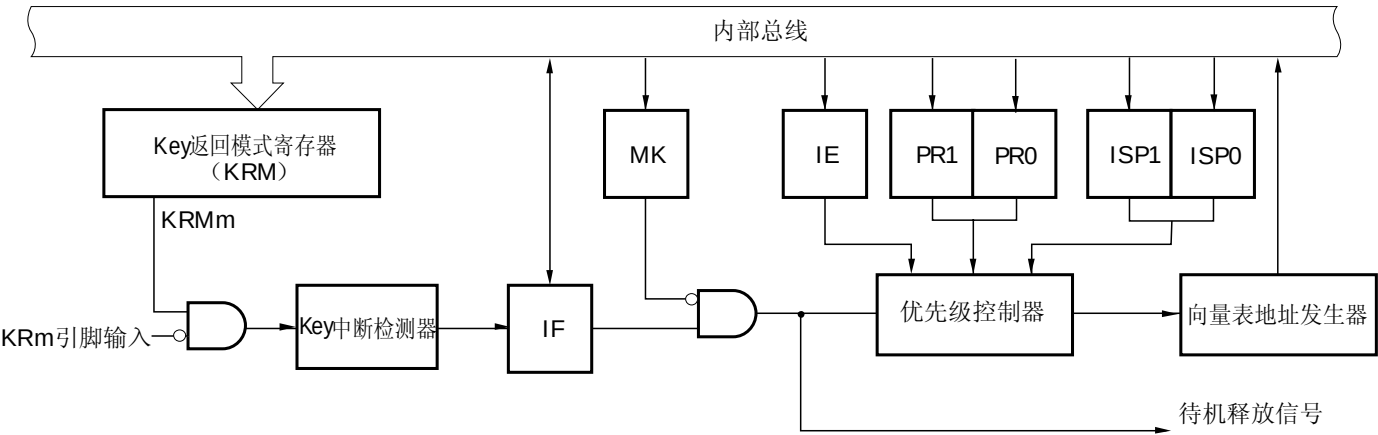
(B) 外部可屏蔽中断(INTPn)



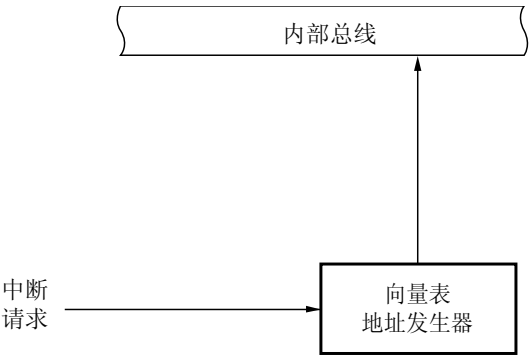
- 备注
- 1. IF: 中断请求标志
 - IE: 中断使能标志
 - ISP0: 服务优先级标志 0
 - ISP1: 服务优先级标志 1
 - MK: 中断屏蔽标志
 - PR0: 优先级指定标志 0
 - PR1: 优先级指定标志 1
 - 2. n = 0 至 11

图 15-1. 中断功能的基本配置(2/2)

(C) 外部可屏蔽中断(INTKR)



(D) 软件中断



- 备注
1. IF: 中断请求标志
IE: 中断使能标志
ISP0: 服务优先级标志 0
ISP1: 服务优先级标志 1
MK: 中断屏蔽标志
PR0: 优先级指定标志 0
PR1: 优先级指定标志 1
 2. m = 0 至 7

15.3 控制中断功能的寄存器

以下 6 种寄存器用于控制中断功能。

- 中断请求标志寄存器 (IF0L,IF0H,IF1L,IF1H,IF2L,IF2H)
- 中断屏蔽标志寄存器 (MK0L,MK0H,MK1L,MK1H,MK2L,MK2H)
- 优先级指定标志寄存器
(PR00L,PR00H,PR01L,PR01H,PR02L,PR02H,PR10L,PR10H,PR11L,PR11H,PR12L,PR12H)
- 外部中断上升沿使能寄存器 (EGP0, EGP1)
- 外部中断下降沿使能寄存器 (EGN0, EGN1)
- 程序状态字 (PSW)

表 15-2 列出了与中断请求源相对应的中断请求标志，中断屏蔽标志和优先级指定标志。

表 15-2. 与中断请求源相对应的标志 (1/2)

中断源	中断请求标志		中断屏蔽标志		优先级指定标志	
		寄存器		寄存器		寄存器
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1	
INTP0	PIF0		PMK0		PPR00, PPR10	
INTP1	PIF1		PMK1		PPR01, PPR11	
INTP2	PIF2		PMK2		PPR02, PPR12	
INTP3	PIF3		PMK3		PPR03, PPR13	
INTP4	PIF4		PMK4		PPR04, PPR14	
INTP5	PIF5		PMK5		PPR05, PPR15	
INTST3	STIF3	IF0H	STMK3	MK0H	STPR03, STPR13	PR00H, PR10H
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13	
INTSRE3	SREIF3		SREMK3		SREPR03, SREPR13	
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10	
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11	
INTST0 ^{注1}	STIF0 ^{注1}		STMK0 ^{注1}		STPR00, STPR10 ^{注1}	
INTCSI00 ^{注1}	CSIIF00 ^{注1}		CSIMK00 ^{注1}		CSIPR000,CSIPR100 ^{注1}	
INTSR0 ^{注2}	SRIF0 ^{注2}		SRMK0 ^{注2}		SRPR00, SRPR10 ^{注2}	
INTCSI01 ^{注2}	CSIIF01 ^{注2}		CSIMK01 ^{注2}		CSIPR001,CSIPR101 ^{注2}	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	

- 注
1. 不要同时使用 UART0 和 CSI00，因为它们共享中断请求源标志。如果中断源 INTST0 和 INTCSI00 中的 1 个产生中断，则 IF1H 的位 5 设置为 1。MK0H,PR00H 和 PR10H 的位 5 支持这 3 个中断源。
 2. 不要同时使用 UART0 和 CSI01，因为它们共享中断请求源标志。如果中断源 INTSR0 和 INTCSI01 中的一个产生中断，则 IF0H 的位 6 设置为 1。MK0H、PR00H 和 PR10H 的位 6 支持这 3 个中断源。

表 15-2. 与中断请求源相对应的标志 (2/2)

中断源	中断请求标志		中断屏蔽标志		优先级指定标志	
		寄存器		寄存器		寄存器
INTST1 ^{注1}	STIF1 ^{注1}	IF1L	STMK1 ^{注1}	MK1L	STPR01, STPR11 ^{注1}	PR01L, PR11L
INTCSI10 ^{注1}	CSIIF10 ^{注1}		CSIMK10 ^{注1}		CSIPR010, CSIPR110 ^{注1}	
INTIIC10 ^{注1}	IICIF10 ^{注1}		IICMK10 ^{注1}		IICPR010, IICPR110 ^{注1}	
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11	
INTSRE1	SREIF1		SREMK1		SREPR01, SREPR11	
INTIICA	IICAIF		IICAMK		IICAPR0, IICAPR1	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102	
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1	
INTRTCI	RTCIF		RTCIMK		RTCIPR0, RTCIPR1	
INTKR	KRIF		KRMK		KRPR0, KRPR1	
INTST2 ^{注2}	STIF2 ^{注2}		STMK2 ^{注2}		STPR02, STPR12 ^{注2}	
INTCSI20 ^{注2}	CSIIF20 ^{注2}		CSIMK20 ^{注2}		CSIPR020, CSIPR120 ^{注2}	
INTIIC20 ^{注2}	IICIF20 ^{注2}		IICMK20 ^{注2}		IICPR020, IICPR120 ^{注2}	
INTP6	PIF6		PMK6		PPR06, PPR16	
INTTM13	TMIF13		TMMK13		TMPR013, TMPR113	
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106	
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12	
INTP7	PIF7		PMK7		PPR07, PPR17	
INTP8	PIF8		PMK8		PPR08, PPR18	
INTP9	PIF9		PMK9		PPR09, PPR19	
INTP10	PIF10		PMK10		PPR010, PPR110	
INTP11	PIF11	IF2H	PMK11	MK2H	PPR011, PPR111	PR02H, PR12H
INTTM10	TMIF10		TMMK10		TMPR010, TMPR110	
INTTM11	TMIF11		TMMK11		TMPR011, TMPR111	
INTTM12	TMIF12		TMMK12		TMPR012, TMPR112	
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12	
INTMD	MDIF		MDMK		MDPR0, MDPR1	

- 注 1. 不要同时使用 UART1, CSI10, 和 IIC10, 因为它们共享中断请求源标志。如果中断源 INTST1, INTCSI10, 和 INTIIC10 中的 1 个产生, IF1L 的第 0 位设置为 1。MK1L, PR01L, 和 PR11L 的第 0 位支持这 3 个中断源。
2. 不要同时使用 UART2, CSI20, 和 IIC20, 因为它们共享中断请求源标志。如果中断源 INTST2, INTCSI20, 和 INTIIC20 中的 1 个产生, IF1L 的位 4 设置为 1。MK1L, PR01L, 和 PR11L 的位 4 支持这 3 个中断源。

(1) 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

当产生相关的中断请求或执行指令时，这些中断请求标志置 1。当执行的指令是响应中断请求或复位输入时，这些标志清零。

当响应中断时，中断请求标志自动清零，然后进入中断服务程序。

可由 1 位或 8 位存储器操作指令设置 1H, IF2L 和 IF2H。当 IF0L 与 IF0H, IF1L 与 IF1H 和 IF2L 与 IF2H 组合起来形成 16 位寄存器 IF0, IF1 和 IF2 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零 00H。

备注 如果执行写入此寄存器数据的指令，指令执行的时钟个数增加 2。

图 15-2. 中断请求标志寄存器的格式 (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (1/2)

地址: FFFE0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIIF	WDTIIF

地址: FFFE1H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0	SRIF0 CSIIIF01	STIF0 CSIIIF00	DMAIF1	DMAIF0	SREIF3	SRIF3	STIF3

地址: FFFE2H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICAIF	SREIF1	SRIF1	STIF1 CSIIIF10 IICIF10

地址: FFFE3H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04	TMIF13	PIF6	STIF2 CSIIIF20 IICIF20	KRIF	RTCIIF	RTCIF	ADIF

地址: FFFD0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	PIF10	PIF9	PIF8	PIF7	SRIF2	TMIF07	TMIF06	TMIF05

图 15-2. 中断请求标志寄存器的格式 (IF0L、IF0H、IF1L、IF1H、IF2L、IF2H) (2/2)

地址: FFFD1H 复位后: 00H R/W

符号	7	6	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	0	0	MDIF	SREIF2	TMIF12	TMIF11	TMIF10	PIF11

XXIFX	中断请求标志
0	没有产生中断请求信号
1	产生中断请求, 中断请求状态

- 注意事项
1. 确保 IF2H 的位 6 和位 7 清零。
 2. 当退出待机模式后并要使用定时器、串行接口或 A/D 转换器时, 先将中断请求标志清零后再对这些部件进行操作。噪音可能会设置中断请求标志。
 3. 当使用中断请求标志寄存器中的标志时, 使用一位存储器操作指令 (CLR1)。在 C 语言中描述时, 应使用诸如“IF0L.0 = 0;”或“_asm(“clr1 IF0L, 0”);”的位操作指令, 因为编译的汇编程序必须为 1 位存储器操作指令(CLR1)。
如果用 C 语言描述的程序使用 8 位存储器操作指令如“IF0L &= 0xfe;”并编译的话, 将变成 3 个指令的汇编程序。

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

在这种情况下, 即使相同中断请求标志寄存器(IF0L)中的另一位的请求标志在“mov a, IF0L”和“mov IF0L, a”, 之间时序上设为 1 时, 该标志在“mov IF0L, a”。时清零。因此在 C 语言中使用 8 位存储器操作指令时必须小心。

(2) 中断屏蔽标志寄存器 (MK0L、MK0H、MK1L、MK1H、MK2L、MK2H)

这些中断屏蔽标志用于使能/禁止相关的可屏蔽中断服务。
可由 1 位或 8 位存储器操作指令设置 MK0L,MK0H,MK1L,MK1H,MK2L,和 MK2H。当 MK0L 与 MK0H,MK1L 与 MK1H,MK2L 与 MK2H 组合起来形成 16 位寄存器 MK0,MK1 和 MK2 时,可由 16 位存储器操作指令设置这些寄存器。
复位信号的产生将这些寄存器设置为 FFH。

备注 如果执行写入此寄存器数据的指令, 指令执行的时钟个数增加 2。

图 15-3. 中断屏蔽标志寄存器的格式 (MK0L、MK0H、MK1L、MK1H、MK2L、MK2H)

地址: FFFE4H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

地址: FFE5H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0	SRMK0 CSIMK01	STMK0 CSIMK00	DMAMK1	DMAMK0	SREMK3	SRMK3	STMK3

地址: FFFE6H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICAMK	SREMK1	SRMK1	STMK1 CSIMK10 IICMK10

地址: FFFE7H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	TMMK04	TMMK13	PMK6	STMK2 CSIMK20 IICMK20	KRMK	RTCIMK	RTCMK	ADMK

地址: FFFD4H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	PMK10	PMK9	PMK8	PMK7	SRMK2	TMMK07	TMMK06	TMMK05

地址: FFFD5H 复位后: FFH R/W

符号	7	6	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	1	1	MDMK	SREMK2	TMMK12	TMMK11	TMMK10	PMK11

XXMKX	中断服务控制
0	使能中断服务
1	禁止中断服务

注意事项 确保 MK2H 的位 6 和位 7 置 1。

(3) 优先级指定标志寄存器 (PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H)

这些优先级指定标志寄存器用于设置相关的可屏蔽中断优先级次序。

优先级可通过组合使用 PR0xy 和 PR1xy 寄存器设置 (xy = 0L、0H、1L、1H、2L 或 2H)。

可由 1 位或 8 位存储器操作指令设置 PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、和 PR12H。当 PR00L 与 PR00H、PR01L 与 PR01H、PR02L 与 PR02H、PR10L 与 PR10H、PR11L 与 PR11H、和 PR12L 与 PR12H 组合起来形成 16 位寄存器 PR00、PR01、PR02、PR10、PR11 和 PR12 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的产生将这些寄存器设置为 FFH。

备注 如果执行写入此寄存器数据的指令，指令执行的时钟个数增加 2。

图 15-4. 优先级指定标志寄存器的格式

(PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H) (1/2)

地址: FFFE8H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

地址: FFECH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

地址: FFFE9H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00	SRPR00 CSIPR001	STPR00 CSIPR000	DMAPR01	DMAPR00	SREPR03	SRPR03	STPR03

地址: FFFEDH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10	SRPR10 CSIPR101	STPR10 CSIPR100	DMAPR11	DMAPR10	SREPR13	SRPR13	STPR13

地址: FFFEAH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICAPR0	SREPR01	SRPR01	STPR01 CSIPR010 IICPR010

地址: FFFEEH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICAPR1	SREPR11	SRPR11	STPR11 CSIPR110 IICPR110

图 15-4. 优先级指定标志寄存器的格式

(PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H) (2/2)

地址: FFEBH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	TMPR004	TMPR013	PPR06	STPR02 CSIPR020 IICPR020	KRPR0	RTCIPR0	RTCPR0	ADPR0

地址: FFEFH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	TMPR104	TMPR113	PPR16	STPR12 CSIPR120 IICPR120	KRPR1	RTCIPR1	RTCPR1	ADPR1

地址: FFFD8H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	PPR010	PPR09	PPR08	PPR07	PPR06	TMPR007	TMPR006	TMPR005

地址: FFFDCH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	PPR110	PPR19	PPR18	PPR17	PPR16	TMPR107	TMPR106	TMPR105

地址: FFFD9H 复位后: FFH R/W

符号	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	1	1	MDPR0	SREPR02	TMPR012	TMPR011	TMPR010	PPR011

地址: FFFDDH 复位后: FFH R/W

符号	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	1	1	MDPR1	SREPR12	TMPR112	TMPR111	TMPR110	PPR111

XXPR1X	XXPR0X	优先级选择
0	0	指定优先级 0 (高优先级)
0	1	指定优先级 1
1	0	指定优先级 2
1	1	指定优先级 3 (低优先级)

注意事项 确保 PR02H 和 PR12H 的位 6 和位 7 至 1。

(4) 外部中断上升沿使能寄存器 (EGP0, EGP1), 外部中断下降沿使能寄存器(EGN0, EGN1)

这两个寄存器用于指定 INTP0 至 INTP11 的有效沿。

EGP0, EGP1, EGN0 和 EGN1 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生可将这些寄存器清零 00H。

图 15-5. 外部中断上升沿使能寄存器 (EGP0, EGP1)
和外部中断下降沿使能寄存器 (EGN0, EGN1)的格式

地址: FFF38H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

地址: FFF39H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

地址: FFF3AH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	EGP8

地址: FFF3BH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	EGN8

EGPn	EGNn	INTPn 引脚有效沿的选择 (n = 0 至 11)
0	0	禁止脉冲沿检测
0	1	下降沿
1	0	上升沿
1	1	兼有上升沿和下降沿

表 15-3 显示了与 EGPn 和 EGNn 相关的端口。

表 15-3. EGPn 和 EGNn 相关的端口

检测使能位		脉冲沿检测端口	中断请求信号
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P46	INTP1
EGP2	EGN2	P47	INTP2
EGP3	EGN3	P30	INTP3
EGP4	EGN4	P31	INTP4
EGP5	EGN5	P16	INTP5
EGP6	EGN6	P140	INTP6
EGP7	EGN7	P141	INTP7
EGP8	EGN8	P74	INTP8
EGP9	EGN9	P75	INTP9
EGP10	EGN10	P76	INTP10
EGP11	EGN11	P77	INTP11

注意事项 在从外部中断功能切换到端口功能时，可能会检测到脉冲沿，因此通过将 EGPn 和 EGNn 清零，可选择端口模式。

备注 n = 0 至 11

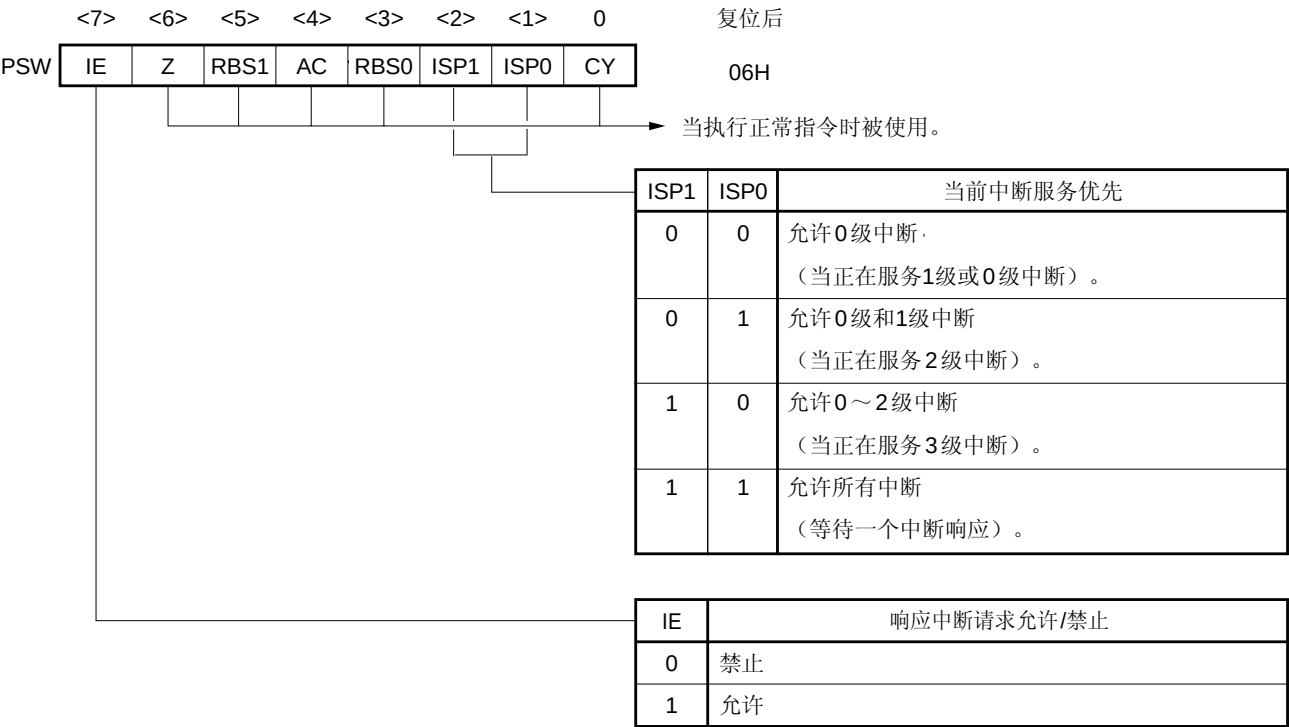
(5) 程序状态字 (PSW)

PSW 是用于保存指令执行结果和中断请求的当前状态的寄存器。在 PSW 中包含 IE 标志(设置使能/禁止可屏蔽中断)和 ISP0 与 ISP1 标志(控制中断嵌套)。

除了 8 位读/写操作指令外，还可使用位操作指令和专用指令 (EI 和 DI) 对该寄存器进行操作。在响应向量中断请求时，如果执行 BRK 指令，则将 PSW 的内容自动保存到堆栈中，并且将 IE 标志复位为 0。如果响应可屏蔽中断请求，则将响应中断的优先级指定标志的内容传输到 ISP0 和 ISP1 标志中。执行 PUSH PSW 指令将 PSW 的内容保存到堆栈中。而执行 RETI、RETB 和 POP PSW 指令可将这些内容从堆栈中恢复。

复位信号的产生将程序状态字的内容设置为 06H。

图 15-6. 程序状态字的格式



15.4 中断服务操作

15.4.1 可屏蔽的中断响应

当中断请求标志=1 且与该中断请求相关的屏蔽标志(MK) 清零时，可以响应这个可屏蔽中断请求。如果处于中断使能状态(IE 标志=1)，则可以响应向量中断请求。但在一个较高优先级中断请求服务期间，不响应低优先级的中断请求。

从一个可屏蔽中断请求产生，到中断服务执行所经历的时间，如表 15-4 所示。

中断请求响应时序可参见图 15-8 和 15-9。

表 15-4. 从可屏蔽中断产生到执行中断服务所需要的时间

	最短时间	最长时间 ^注
时间	9 个时钟	14 个时钟

注 如果是在 RET 指令执行之前产生一个中断请求，则等待时间会更长。

备注 1 个时钟：1/fCLK (fCLK：CPU 时钟)

如果同时产生两个或两个以上的可屏蔽中断请求，则首先响应优先级指定标志中优先级别高的请求。如果两个或两个以上的中断请求具有相同的优先级别，则首先响应具有最高默认优先级的中断请求。

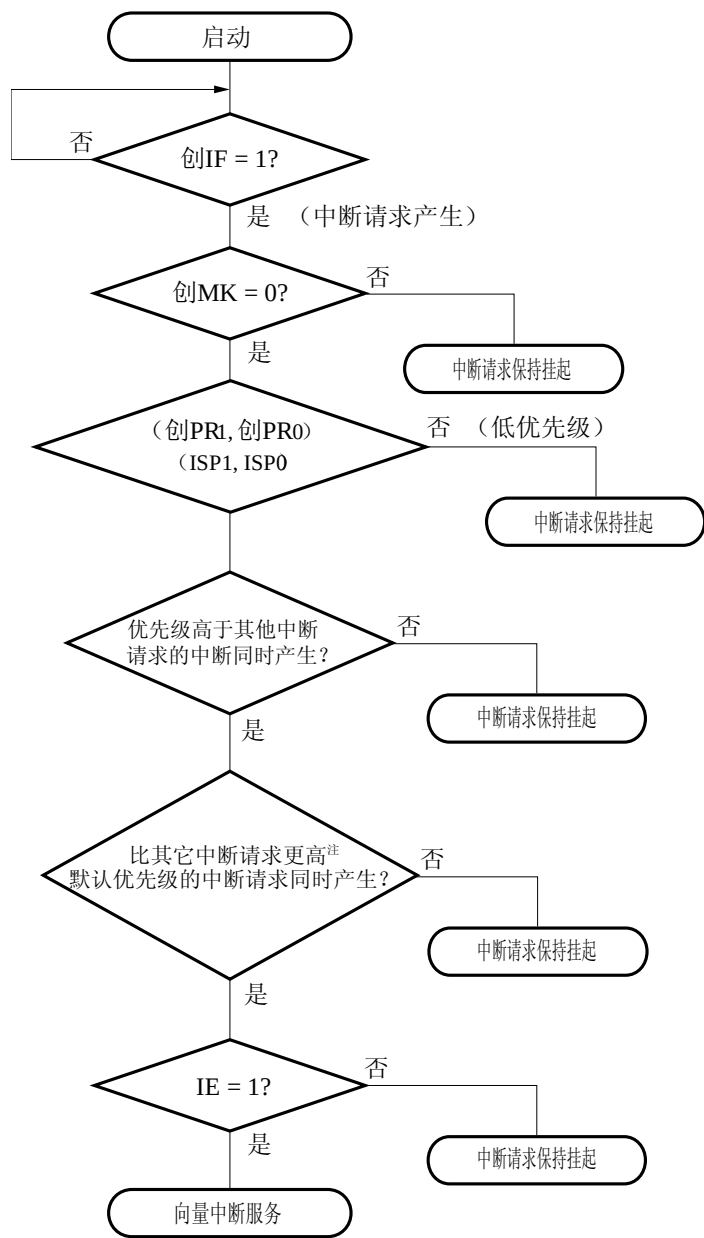
当使能响应中断请求时，处于等待状态的中断请求响应。

图 15-7 显示了中断请求响应算法。

如果响应了一个可屏蔽中断请求，则将 PSW、PC 的内容依次保存到堆栈中，然后将 IE 标志复位为 0，并将与响应的中断相关的优先级指定标志的内容传输到 ISP1 和 ISP0 中。将用于每个中断请求的向量表数据发送到 PC 中并分支。

可通过执行 RETI 指令从中断返回。

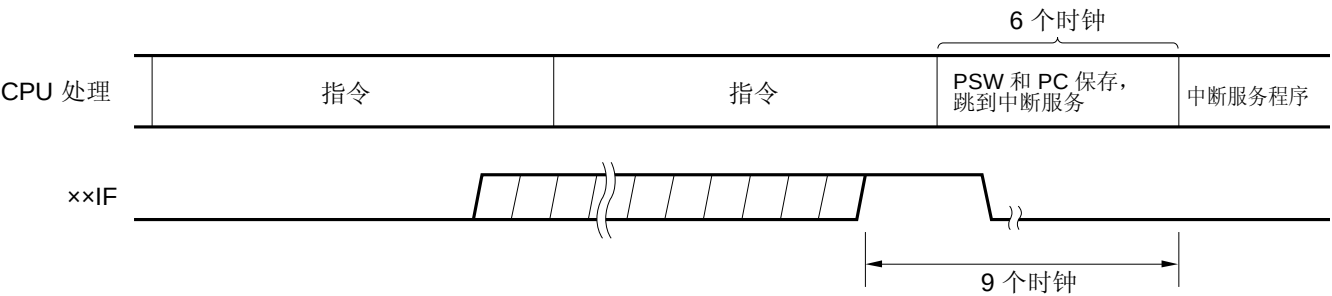
图 15-7. 中断请求响应处理算法



- xxIF: 中断请求标志
- xxMK: 中断屏蔽标志
- xxPR0: 优先级指定标志 0
- xxPR1: 优先级指定标志 1
- IE: 控制可屏蔽中断请求响应的标志(1 = 使能、0 = 禁止)
- ISP0, ISP1: 指示正在服务的中断优先级别的标志(参见 图 15-6)

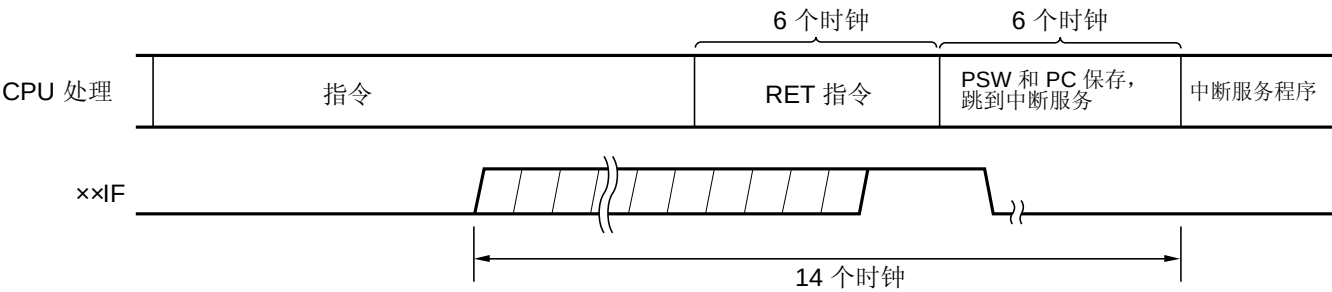
注 关于默认优先级，参考 表 15-1 中断源列表

图 15-8. 中断请求响应时序(最短时间)



备注 1 个时钟: $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

图 15-9. 中断请求响应时序(最长时间)



备注 1 个时钟: $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

15.4.2 软件中断请求响应

执行 BRK 指令可响应软件中断。 软件中断不能禁止。

如果响应了一个软件中断请求，则将程序状态字 (PSW) 和程序计数器 (PC) 的内容依次保存到堆栈中，然后将 IE 标志复位为 0，并将向量表的内容 (0007EH, 0007FH) 发送到 PC 中并分支。

可通过执行 RETB 指令从软件中断返回。

注意事项 不能使用 RETI 指令从软件中断返回。

15.4.3 中断嵌套

在执行一个中断服务程序时，又响应了其它中断，这时就产生了中断嵌套。

除非选择允许中断请求响应状态 (IE = 1)，否则不会产生中断嵌套。在响应一个中断请求时，禁止响应其它中断请求 (IE = 0)。因此，如果要允许中断嵌套，必须在中断服务期间执行 EI 指令，将 IE 标志置 1，从而允许响应其它中断请求。

此外，即使允许中断，也不一定允许中断嵌套，这是因为受到中断优先级控制的限制。可使用两类优先级控制方式：默认优先级控制和可编程优先级控制。可编程优先级控制用于中断嵌套。

在中断允许状态中，如果产生的中断请求的优先级与正在服务的中断优先级相等或高于它，则响应该中断请求，从而产生中断嵌套。如果产生的中断请求的优先级低于正在服务的中断优先级，则不响应该中断请求。由于禁止中断或中断请求的优先级别较低，这些不允许的中断请求处于等待状态。当前终端服务结束时，则在执行至少一条主程序指令后才可以响应处于挂起状态的中断请求。表 15-5 显示了允许中断嵌套的中断请求之间的关系，图 15-10 为中断嵌套示例。

表 15-5. 中断服务期间允许进行中断嵌套的中断请求之间的关系

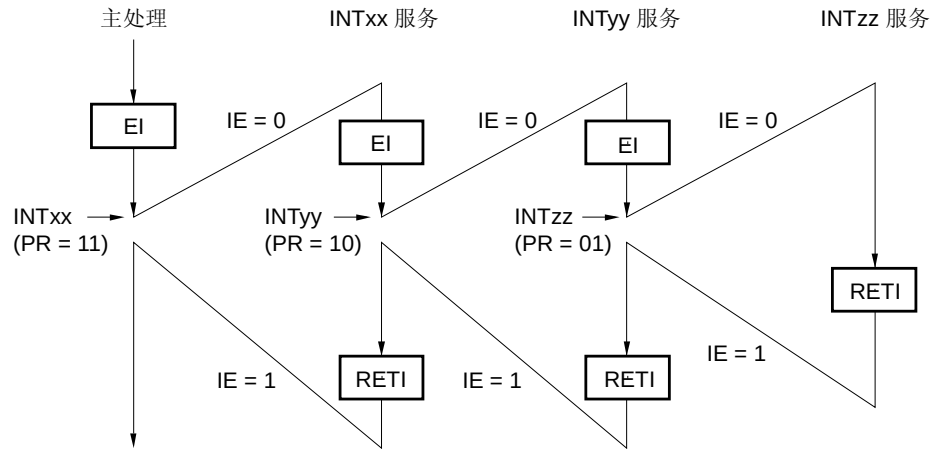
中断嵌套 正在服务的中断		可屏蔽中断请求								软件中断请求
		优先级 0 (PR = 00)		优先级 1 (PR = 01)		优先级 2 (PR = 10)		优先级 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
可屏蔽的中断	ISP1 = 0 ISP0 = 0	○	×	×	×	×	×	×	×	○
	ISP1 = 0 ISP0 = 1	○	×	○	×	×	×	×	×	○
	ISP1 = 0 ISP0 = 1	○	×	○	×	○	×	×	×	○
	ISP1 = 1 ISP0 = 1	○	×	○	×	○	×	○	×	○
软件中断		○	×	○	×	○	×	○	×	○

备注

- ：使能中断嵌套
- ×：禁止中断嵌套
- ISP0, ISP1, 和 IE 为 PSW 中的标志位。
 ISP1 = 0, ISP0 = 0：正在服务优先级 1 或 0 的中断。
 ISP1 = 0, ISP0 = 1：正在服务优先级 2 的中断。
 ISP1 = 1, ISP0 = 0：正在服务优先级 3 的中断。
 ISP1 = 1, ISP0 = 1：等待中断响应。
 IE = 0：禁止响应中断请求。
 IE = 1：使能响应中断请求。
- PR 为 PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L 和 PR12H 中的标志位。
 PR = 00：通过 $\times \times PR1 \times = 0, \times \times PR0 \times = 0$, 指定等级 0 (较高优先级)
 PR = 01：通过 $\times \times PR1 \times = 0, \times \times PR0 \times = 1$, 指定等级 1
 PR = 10：用 $\times \times PR1 \times = 1, \times \times PR0 \times = 0$ 指定优先级 2
 PR = 11：指定等级 3 为 $\times \times PR1 \times = 1, \times \times PR0 \times = 1$ (较低优先级)

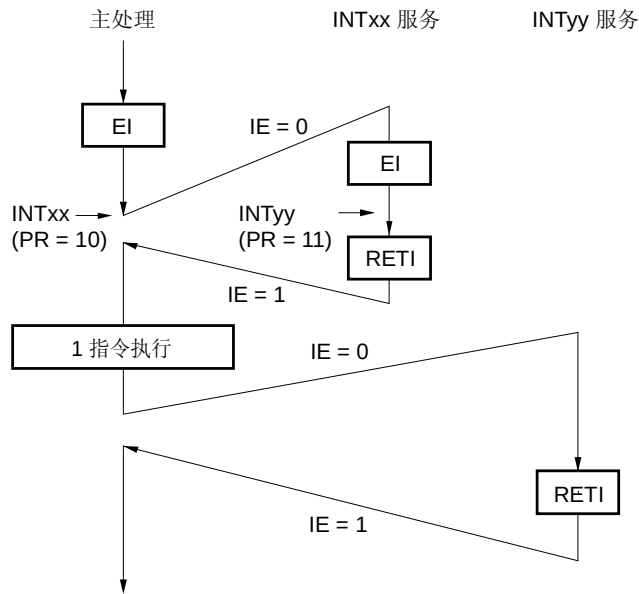
图 15-10. 中断嵌套示例(1/2)

例 1. 产生两次中断嵌套



在进行 INTxx 中断服务期间，响应了两个中断请求 INTyy 和 INTzz，这时就产生了中断嵌套。在响应每个中断之前，必须先执行 EI 指令才能允许响应中断请求。

例 2. 由于优先级控制没有产生中断嵌套

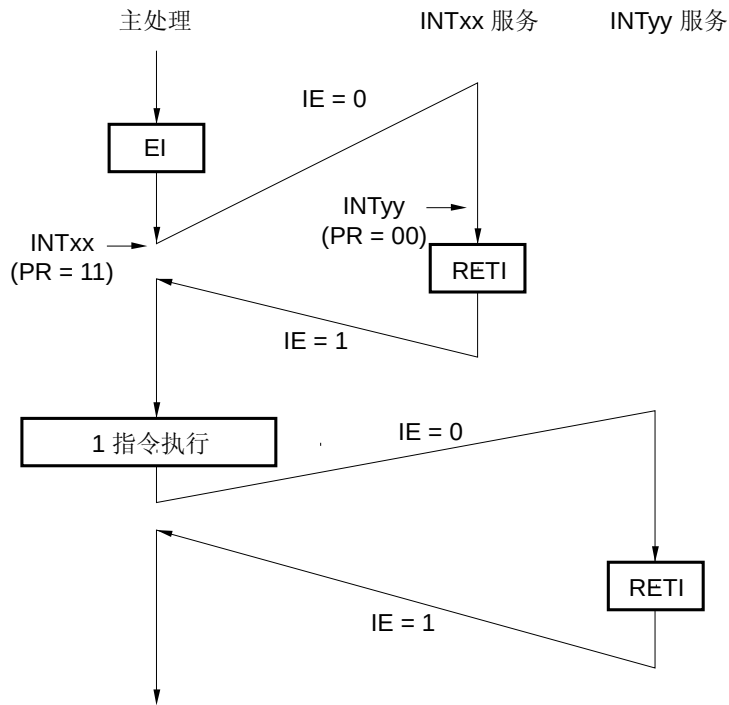


在 INTxx 中断服务期间不响应中断请求 INTyy，因为 INTyy 的优先级低于 INTxx，这样就不会产生中断嵌套。中断请求 INTyy 处于等待状态，在执行一条主程序指令后响应该中断请求。

- PR = 00: 通过 $\times\times PR1\times = 0, \times\times PR0\times = 0$, 指定等级 0(较高优先级)
- PR = 01: 通过 $\times\times PR1\times = 0, \times\times PR0\times = 1$, 指定等级 1
- PR = 10: 用 $\times\times PR1\times = 1, \times\times PR0\times = 0$ 指定优先级 2
- PR = 11: 指定等级 3 为 $\times\times PR1\times = 1, \times\times PR0\times = 1$ (较低优先级)
- IE = 0: 禁止中断请求响应。
- IE = 1: 使能响应中断请求。

图 15-10. 中断嵌套示例(2/2)

例 3. 由于不允许中断没有产生中断嵌套



在 INTxx 中断服务期间不允许中断(不执行 EI 指令), 因此不响应中断请求 INTyy, 这样也不会产生中断嵌套。中断请求 INTyy 处于等待状态, 在执行一条主程序指令后响应该中断请求。

PR = 00: 通过 $\times \times \text{PR}1 \times = 0, \times \times \text{PR}0 \times = 0$, 指定等级 0(较高优先级)

PR = 01: 通过 $\times \times \text{PR}1 \times = 0, \times \times \text{PR}0 \times = 1$, 指定等级 1

PR = 10: 用 $\times \times \text{PR}1 \times = 1, \times \times \text{PR}0 \times = 0$ 指定优先级 2

PR = 11: 指定等级 3 为 $\times \times \text{PR}1 \times = 1, \times \times \text{PR}0 \times = 1$ (较低优先级)

IE = 0: 禁止中断请求响应。

IE = 1: 允许响应中断请求。

15.4.4 保持中断请求

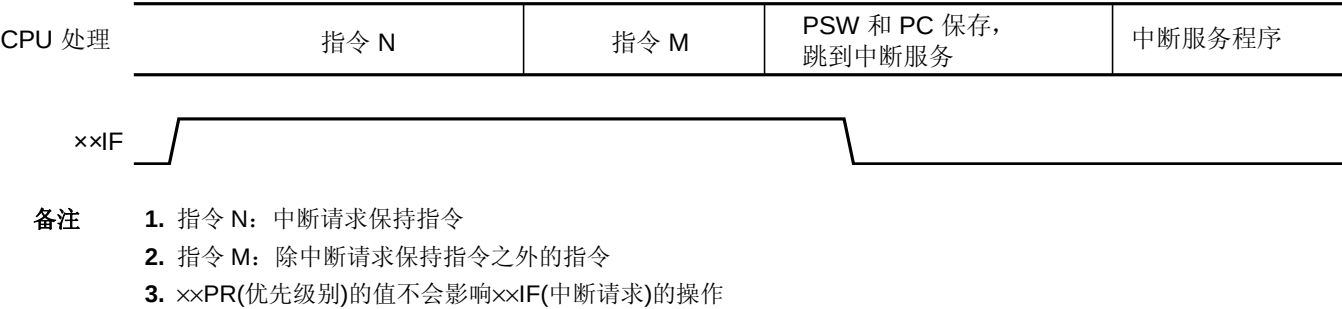
在某些指令执行期间，即使出现中断请求，请求响应也要保持等待状态，直到下一条指令执行结束。以下列出这类指令(中断请求保持指令)。

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, \$addr8
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- 用于 IF0L、IF0H、IF1L、IF1H、IF2L、IF2H、MK0L、MK0H、MK1L、MK1H、MK2L、MK2H、PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L 和 PR12H 寄存器的操作指令。

注意事项 BRK 指令不属于上述列出的中断请求保持指令。但通过执行 BRK 指令激活的软件中断会将 IE 标志清零。因此，即使在执行 BRK 指令期间产生可屏蔽中断请求，该中断请求也不会响应。

图 15-11 显示了处于等待状态的中断请求时序。

图 15-11. 中断请求保持



16.1 按键中断的功能

可通过设置按键返回模式寄存器 (KRM) 和向按键中断输入引脚 (KR0 至 KR7) 输入一个下降沿产生按键中断 (INTKR)。

表 16-1. 按键中断检测引脚的分配

标志	描述
KRM0	以 1 位单元控制 KR0 信号。
KRM1	以 1 位单元控制 KR1 信号。
KRM2	以 1 位单元控制 KR2 信号。
KRM3	以 1 位单元控制 KR3 信号。
KRM4	以 1 位单元控制 KR4 信号。
KRM5	以 1 位单元控制 KR5 信号。
KRM6	以 1 位单元控制 KR6 信号。
KRM7	以 1 位单元控制 KR7 信号。

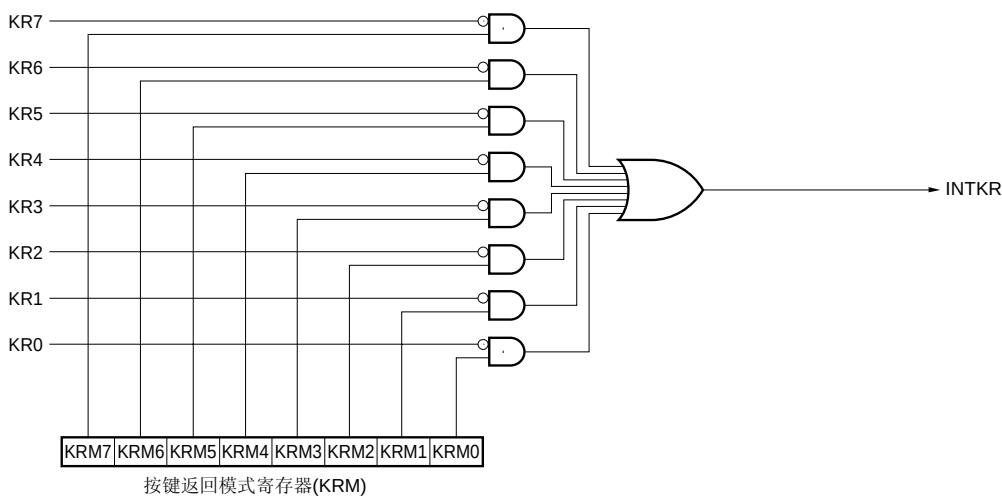
16.2 按键中断的配置

按键中断包括以下硬件。

表 16-2 按键中断的配置

项目	配置
控制寄存器	按键返回模式寄存器(KRM)

图 16-1. 按键中断的框图



16.3 控制按键中断的寄存器

(1) 按键返回模式寄存器(KRM)

该寄存器使用 KR0 至 KR7 信号分别控制 KRM0 至 KRM7 位。
KRM 可由 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 00H。

图 16-2. 按键返回模式寄存器 (KRM) 的格式

地址: FFF37H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	按键中断模式寄存器
0	未检测到按键中断信号
1	检测按键中断信号

- 注意事项
1. 如果 KRM0 至 KRM7 中任何一位设置为 1，则将相应的上拉电阻寄存器 7 (PU7) 的第 0 位至第 7 位 (PU70 至 PU77) 设置为 1。
 2. 如果设置 KRM 寄存器的目标位的同时向键中断输入引脚输入低电平时，将产生中断。为了忽略此中断，在使用中断屏蔽标志禁止中断服务后设置 KRM 寄存器。然后，清除中断请求标志并在等待键中断输入低电平宽度（250 ns 或更多）后使能中断服务。
 3. 在按键中断模式中未使用的位，可用作正常端口。

备注 n = 0 至 7

第 17 章 待机功能

17.1 待机功能及配置

17.1.1 待机功能

待机功能用于减少系统的工作电流，有以下两种模式。

(1) HALT 模式

通过执行 HALT 指令设置 HALT 模式。在 HALT 模式中，CPU 操作时钟停止。HALT 模式设置之前，如果高速系统时钟振荡器、内部高速振荡器、20 MHz 内部高速振荡器、或副系统时钟振荡器正在运行，则每种时钟的振荡继续。在此模式下，工作电流的下降在不如 STOP 模式中下降得多，但 HALT 模式对于中断请求产生后立即重启操作和频繁进行间断操作非常有效。

(2) STOP 模式

通过执行 STOP 指令设置 STOP 模式。在 STOP 模式中，高速系统时钟振荡器和内部高速振荡器停止操作，整个系统的操作停止，这样 CPU 的工作电流将会大幅下降。

因为可通过中断请求清除该模式，所以可以执行中断操作。但是，选择 X1 时钟时，在释放 STOP 模式后，因为需要一段等待时间以确保振荡器振荡稳定，因此如果需要在产生中断请求后立即进行处理，则应选择 HALT 模式。

在这两种模式中，寄存器、标志和数据存储器的内容将会保持进入待机模式前的内容。I/O 端口输出锁存器和输出缓冲器的状态也将保持。

- 注意事项**
1. 仅当 CPU 使用主系统时钟时，才能使用 STOP 模式。当 CPU 使用副系统时钟时，不能设置 STOP。而当 CPU 无论是使用主系统时钟还是副系统时钟，均可以使用 HALT 模式。
 2. 当切换到 STOP 模式时，确保在执行 STOP 指令前，停止使用主系统时钟的外围硬件的操作。
 3. 当使用待机功能时，建议采用以下步骤降低 A/D 转换器的操作电流：首先将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 和第 0 位 (ADCE) 清零以停止 A/D 转换操作，然后执行 STOP 指令。
 4. 在 HALT 或 STOP 模式下，可通过选项字节选择内部低速振荡器是继续还是停止。详细信息参考第 22 章 选项字节。
 5. 当 CPU 使用 20 MHz 内部高速振荡时钟时，不能执行 STOP 指令。确保在变换到内部高速振荡时钟工作后执行 STOP 指令。

17.1.2 控制待机功能的寄存器

待机功能由以下两个寄存器控制。

- 振荡稳定时间计数器状态寄存器(OSTC)
- 振荡稳定时间选择寄存器(OSTS)

备注 有关启动、停止以及选择时钟的寄存器的情况，可参考**第 5 章 时钟发生器**。

(1) 振荡稳定时间计数状态寄存器(OSTC)

这是表明 X1 时钟振荡稳定时间计数器的计数状态的寄存器。

X1 时钟振荡稳定时间在如下情况可以检测。

- 在内部高速振荡时钟或副系统时钟用作 CPU 时钟期间，如果 X1 时钟开始振荡。
- 当内部高速振荡时钟用作 X1 时钟振荡的 CPU 时钟时，如果进入 STOP 模式然后释放。

可由 1 位或 8 位存储器操作指令读取 OSTC。

当复位释放(通过 RESET 输入、POC、LVI、WDT 和执行非法指令复位)、STOP 指令以及 MSTOP(CSC 寄存器的第 7 位) = 1 时，将此寄存器清零 (00H)。

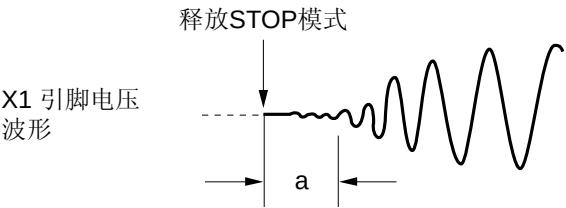
图 17-1. 振荡稳定时间计数器状态寄存器(OSTC)的格式

地址: FFFA2H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	振荡稳定时间的状态		
									$f_x = 10\text{ MHz}$	$f_x = 20\text{ MHz}$
0	0	0	0	0	0	0	0	$2^8/f_x\text{ max.}$	25.6 $\mu\text{s max.}$	12.8 $\mu\text{s max.}$
1	0	0	0	0	0	0	0	$2^9/f_x\text{ min.}$	25.6 $\mu\text{s min.}$	12.8 $\mu\text{s min.}$
1	1	0	0	0	0	0	0	$2^9/f_x\text{ min.}$	51.2 $\mu\text{s min.}$	25.6 $\mu\text{s min.}$
1	1	1	0	0	0	0	0	$2^{10}/f_x\text{ min.}$	102.4 $\mu\text{s min.}$	51.2 $\mu\text{s min.}$
1	1	1	1	0	0	0	0	$2^{11}/f_x\text{ min.}$	204.8 $\mu\text{s min.}$	102.4 $\mu\text{s min.}$
1	1	1	1	1	0	0	0	$2^{13}/f_x\text{ min.}$	819.2 $\mu\text{s min.}$	409.6 $\mu\text{s min.}$
1	1	1	1	1	1	0	0	$2^{15}/f_x\text{ min.}$	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	$2^{17}/f_x\text{ min.}$	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/f_x\text{ min.}$	26.21 ms min.	13.11 ms min.

- 注意事项
- 在经历上述时间后，各个位从 MOST8 起按顺序设置为 1，并保持为 1。
 - 振荡稳定时间计数器的计数达到 OSTS 设置的振荡稳定时间。如果已进入 STOP 模式，并在 CPU 使用内部高速振荡时钟时释放 STOP 模式，则可以按以下方式设置振荡稳定时间。
 - 预期的 OSTC 振荡稳定时间 $\leq$ OSTS 设置的振荡稳定时间因此需要注意，在释放 STOP 模式后，只有达到 OSTS 设置的振荡稳定时间的状态设置到 OSTC。
 - X1 时钟振荡稳定等待时间不包含直到时钟振荡开始的时间(下为“a”)。



备注 f_x : X1 时钟振荡频率

(2) 振荡稳定时间选择寄存器(OSTS)

此寄存器用于选择 STOP 模式释放时的 X1 时钟振荡稳定等待时间。

在释放 STOP 模式后，且 CPU 使用 X1 时钟时，操作等待由 OSTS 设置的时间。

当释放 STOP 模式后，且 CPU 使用内部高速振荡时钟时，可通过使用 OSTC 确认是否已经历了预期的振荡稳定时间。振荡稳定时间可检测是否等于 OSTC 设置的时间。

可由 8 位存储器操作指令设置 OSTS。

复位信号的产生将 OSTS 设置为 07H。

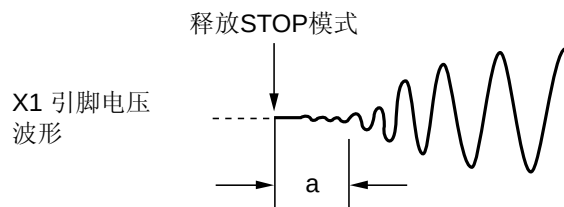
图 17-2. 振荡稳定时间选择寄存器(OSTS)的格式

地址: FFFA3H 复位后: 07H R/W

符号	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0		振荡稳定时间选择	
				$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$
0	0	0	$2^9/f_x$	25.6 μs	设置禁止
0	0	1	$2^9/f_x$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_x$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_x$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_x$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_x$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_x$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_x$	26.21 ms	13.11 ms

- 注意事项**
1. 如果在 X1 时钟用作 CPU 时钟时要设置 STOP 模式，则应在执行 STOP 指令之前设置 OSTS。
 2. 禁止设置振荡稳定时间为 20 μs 或更短。
 3. 改变 OSTS 寄存器的设置之前，确认 OSTC 寄存器的计数操作已经完成。
 4. 在 X1 时钟振荡稳定期间，不要修改 OSTS 寄存器的值。
 5. 振荡稳定时间计数器的计数达到 OSTS 设置的振荡稳定时间。如果已进入 STOP 模式，并在 CPU 使用内部高速振荡时钟时释放 STOP 模式，则可以按以下方式设置振荡稳定时间。
 - 预期的 OSTC 振荡稳定时间 $\leq$ OSTS 设置的振荡稳定时间
 因此需要注意，在释放 STOP 模式后，只有达到 OSTS 设置的振荡稳定时间的状态设置到 OSTC。
 6. X1 时钟振荡稳定等待时间不包含直到时钟振荡开始的时间(下为“a”)。



备注 f_x : X1 时钟振荡频率

17.2 待机功能的操作

17.2.1 HALT 模式

(1) HALT 模式

通过执行 HALT 指令设置 HALT 模式。无论设置前 CPU 时钟是高速系统时钟、内部高速振荡时钟、20 MHz 内部高速振荡时钟、还是副系统时钟，都可以设置为 HALT 模式。

HALT 模式中的操作状态如下所示。

表 17-1. HALT 模式的操作状态 (1/2)

HALT 模式设置 项目			当 CPU 使用主系统时钟并执行 HALT 指令时		
			当 CPU 使用内部高速振荡时钟 (f _{IH})或 20 MHz 内部高速振荡时钟(f _{IH20})时，	当 CPU 使用 X1 时钟 (f _x) 时	当 CPU 使用外部主系统时钟 (f _{EX}) 时
系统时钟			停止 CPU 时钟		
	主系统时钟	f _{IH} , f _{IH20}	操作继续 (不能停止)		保持设置 HALT 模式前的状态
		f _x	保持设置 HALT 模式前的状态	操作继续 (不能停止)	不能操作
		f _{EX}		不能操作	操作继续 (不能停止)
	副系统时钟	f _{XT}	保持设置 HALT 模式前的状态		
	f _{IL}		通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WDTON) 来设置 • WDTON = 0: 停止， • WDTON = 1 且 WDSTBYON = 1: 振荡， • WDTON = 1 且 WDSTBYON = 0: 停止		
CPU			操作停止		
Flash 存储器			操作停止		
RAM			操作停止。但是，在 HALT 模式之前的状态设置的电压高于 POC 检测电压。		
端口(锁存器)			保持设置 HALT 模式前的状态		
定时器阵列单元 TAU			可操作		
实时计数器 (RTC)					
看门狗定时器			通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WDTON) 来设置 • WDTON = 0: 停止， • WDTON = 1 且 WDSTBYON = 1: 操作， • WDTON = 1 且 WDSTBYON = 0: 停止		
时钟输出/蜂鸣器输出			可操作		
A/D 转换器					
串行阵列单元 (SAU)					
串行接口(IICA)					
乘法器/除法器					
DMA 控制器					
上电清零功能					
低电压检测功能					
外部中断					
按键中断功能					

备注 f_{IH} : 内部高速振荡时钟
 f_{IH20} : 20 MHz 内部高速振荡时钟
 f_x : X1 时钟
 f_{EX} : 外部主系统时钟
 f_{XT} : XT1 时钟
 f_{IL} : 内部低速振荡时钟

表 17-1. HALT 模式的操作状态 (2/2)

HALT 模式设置			当 CPU 使用副系统时钟并执行 HALT 指令时
项目			当 CPU 使用 XT1 时钟(f_{XT})时
系统时钟			停止 CPU 时钟
主系统时钟	f_{IH}, f_{IH20}	保持设置 HALT 模式前的状态	
	f_X		
	f_{EX}		
副系统时钟		f_{XT}	操作继续 (不能停止)
f_{IL}		通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WDTON) 来设置 • WDTON = 0: 停止, • WDTON = 1 且 WDSTBYON = 1: 振荡, • WDTON = 1 且 WDSTBYON = 0: 停止	
CPU			操作停止
Flash 存储器			操作停止 (在低功耗电流模式下等待状态)
RAM			操作停止。但是, 在 HALT 模式之前的状态设置的电压高于 POC 检测电压。
端口(锁存器)			保持设置 HALT 模式前的状态
定时器阵列单元 TAU			可操作
实时计数器 (RTC)			
看门狗定时器			通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WDTON) 来设置 • WDTON = 0: 停止, • WDTON = 1 且 WDSTBYON = 1: 操作, • WDTON = 1 且 WDSTBYON = 0: 停止
时钟输出/蜂鸣器输出			可操作
A/D 转换器			不能操作
串行阵列单元 (SAU)			可操作
串行接口(IICA)			不能操作
乘法器/除法器			可操作
DMA 控制器			
上电清零功能			
低电压检测功能			
外部中断			
按键中断功能			

备注

f_{IH} : 内部高速振荡时钟

f_{IH20} : 20 MHz 内部高速振荡时钟

f_X : X1 时钟

f_{EX} : 外部主系统时钟

f_{XT} : XT1 时钟

f_{IL} : 内部低速振荡时钟

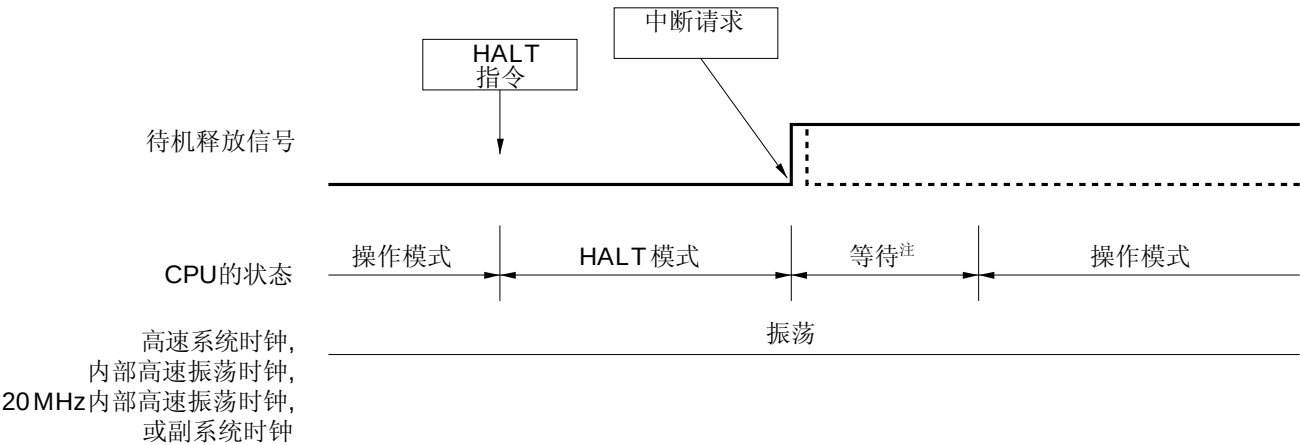
(2) 释放 HALT 模式

以下两种方式可以释放 HALT 模式。

(a) 由未屏蔽的中断请求释放

当产生一个未屏蔽的中断时，释放 HALT 模式。如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

图 17-3. 通过产生中断请求释放 HALT 模式



注 等待时间如下：

- 当产生向量中断服务时
 - 当使用主系统时钟时：10 至 12 个时钟
 - 当使用副系统时钟时：8 至 10 个时钟
- 当不产生向量中断服务时
 - 当使用主系统时钟时：5 或 6 个时钟
 - 当使用副系统时钟时：3 或 4 个时钟

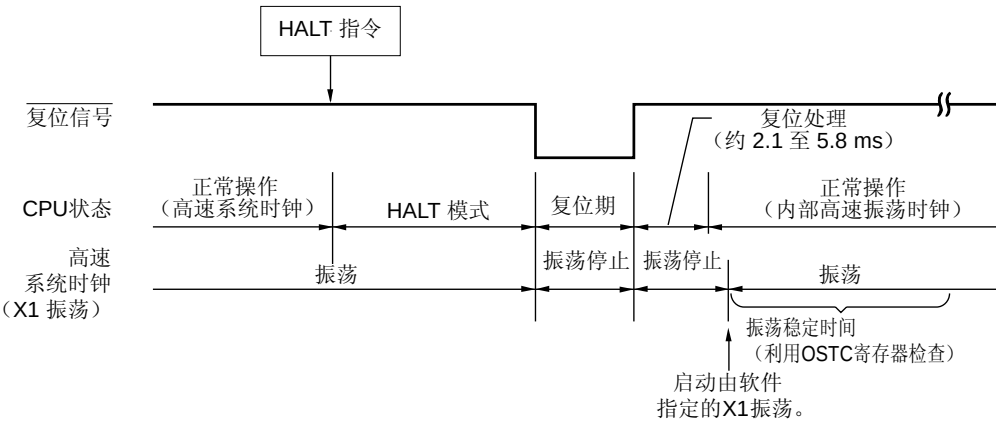
备注 虚线表示释放待机电态的中断请求响应的情况。

(b) 由复位信号的产生释放

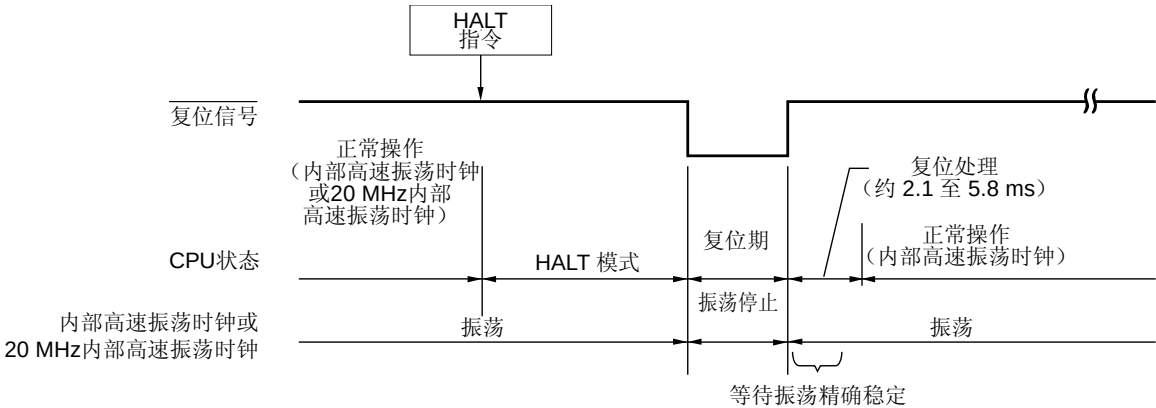
当产生复位信号时，释放 HALT 模式，然后在进行正常复位操作后，程序从复位向量指向的地址处开始执行。

图 17-4. 通过复位释放 HALT 模式

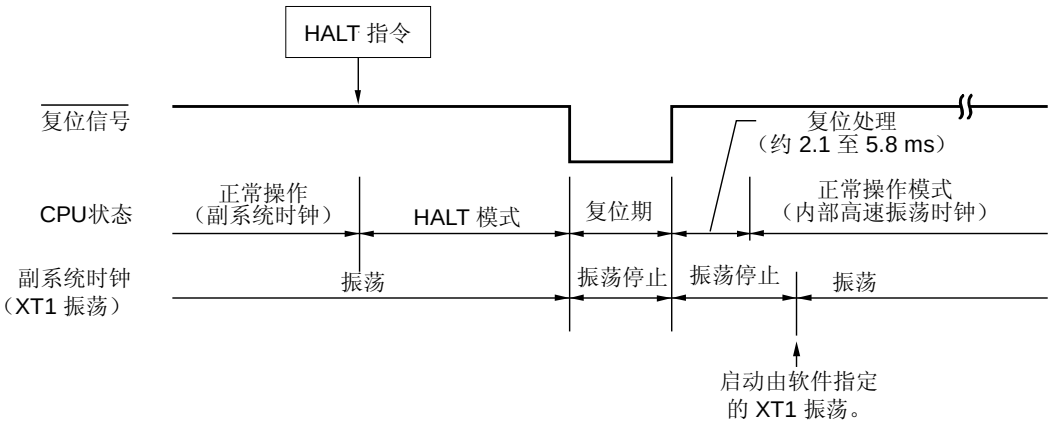
(1) 当 CPU 使用高速系统时钟时



(2) CPU 使用内部高速振荡时钟或 20 MHz 内部高速振荡时钟
高速振荡时钟作为 CPU 时钟使用



(3) 当 CPU 使用副系统时钟



备注 f_x : X1 时钟振荡频率

17.2.2 STOP 模式

(1) STOP 模式设置及操作状态

通过执行 STOP 指令，设置 STOP 模式。而且只有在设置前 CPU 使用内部高速振荡时钟、X1 时钟、或者外部主系统时钟，才可以设置 STOP 模式。

- 注意事项**
1. 由于中断请求信号用于释放待机模式，因此如果一个中断源的中断请求标志设置且中断屏蔽标志清零时，则立即释放待机模式。因此，在执行 STOP 指令后，STOP 模式立即复位到 HALT 模式，并且在经历了振荡稳定时间选择寄存器 (OSTS) 设置的等待时间后，系统返回操作模式。
 2. 当 CPU 使用 20 MHz 内部高速振荡时钟时，不能执行 STOP 指令。确保在变换到内部高速振荡时钟工作后执行 STOP 指令。

STOP 模式中的操作状态如下所示。

表 17-2. STOP 模式的操作状态

STOP 模式设置 项目			当 CPU 使用主系统时钟并执行 STOP 指令时		
			当 CPU 使用内部高速振荡时钟 (f _{IH}) 时	当 CPU 使用 X1 时钟 (f _x) 时	当 CPU 使用外部主系统时钟 (f _{EX}) 时
系统时钟			停止 CPU 时钟		
	主系统时钟	f _{IH}	停止		
		f _x			
		f _{EX}			
	副系统时钟	f _{XT}	保持设置 STOP 模式前的状态		
	f _{IL}		通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WDTON) 来设置 • WDTON = 0: 停止, • WDTON = 1 且 WDSTBYON = 1: 振荡, • WDTON = 1 且 WDSTBYON = 0: 停止		
CPU			操作停止		
Flash 存储器			操作停止		
RAM			操作停止。但是, 在 STOP 模式之前的状态设置的电压高于 POC 检测电压。		
端口(锁存器)			保持设置 STOP 模式前的状态		
定时器阵列单元 TAU			禁止操作		
实时计数器 (RTC)			可操作		
看门狗定时器			通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WDTON) 来设置 • WDTON = 0: 停止, • WDTON = 1 且 WDSTBYON = 1: 操作, • WDTON = 1 且 WDSTBYON = 0: 停止		
时钟输出/蜂鸣器输出			只有当副系统时钟选作计数时钟时才可操作		
A/D 转换器			禁止操作		
串行阵列单元 (SAU)					
串行接口(IICA)			由地址相等操作唤醒		
乘法器/除法器			禁止操作		
DMA 控制器					
上电清零功能			可操作		
低电压检测功能					
外部中断					
按键中断功能					

备注

f_{IH} : 内部高速振荡时钟

f_X : X1 时钟

f_{EX} : 外部主系统时钟

f_{XT} : XT1 时钟

f_{IL} : 内部低速振荡时钟

- 注意事项**
1. 在 STOP 模式释放后，要使用在 STOP 模式下停止操作的外围硬件，以及在 STOP 模式下时钟停止振荡的外围硬件，必须重新启动外围硬件。
 2. 要在 STOP 模式下停止内部低速振荡时钟，可在 HALT/STOP 模式下使用选项字节停止看门狗定时器的操作 (000C0H 的第 0 位(WDSTBYON) = 0)，然后执行 STOP 指令。
 3. 当 CPU 使用高速系统时钟(X1 振荡)时，为了在 STOP 模式释放后缩短振荡稳定时间，可在执行 STOP 指令前将 CPU 时钟暂时切换到内部高速振荡时钟。STOP 模式释放后，在将 CPU 时钟从内部高速振荡时钟切换到高速系统时钟(X1 振荡)之前，应使用振荡稳定时间计数器的状态寄存器 (OSTC) 检测振荡稳定时间。
 4. 当 CPU 使用 20 MHz 内部高速振荡时钟时，不能执行 STOP 指令。确保在变换到内部高速振荡时钟工作后执行 STOP 指令。

(2) 释放 STOP 模式

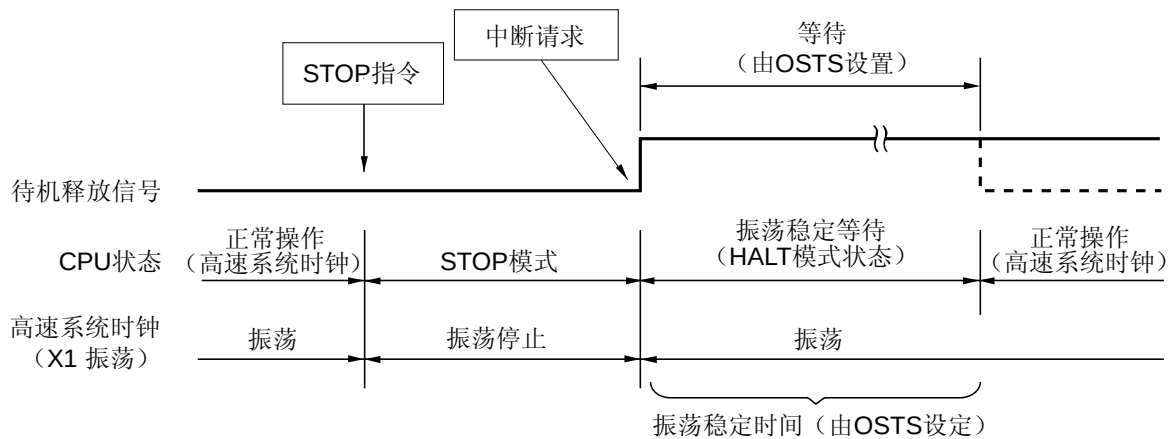
STOP 模式可通过以下两种方式释放。

(a) 由未屏蔽的中断请求释放

当产生一个未屏蔽的中断请求时，释放 STOP 模式。经历振荡稳定时间后，如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

图 17-5. 通过产生中断请求释放 STOP 模式(1/2)

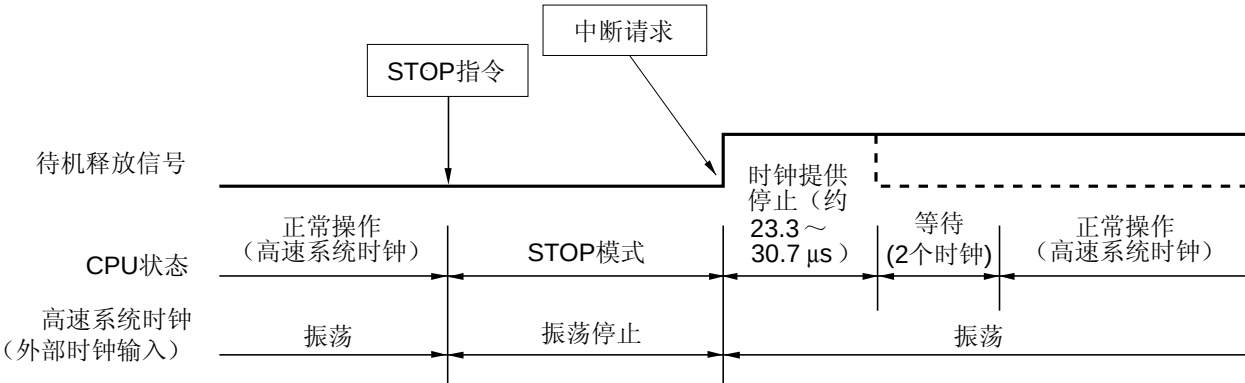
(1) 当 CPU 使用高速系统时钟(X1 振荡器)时



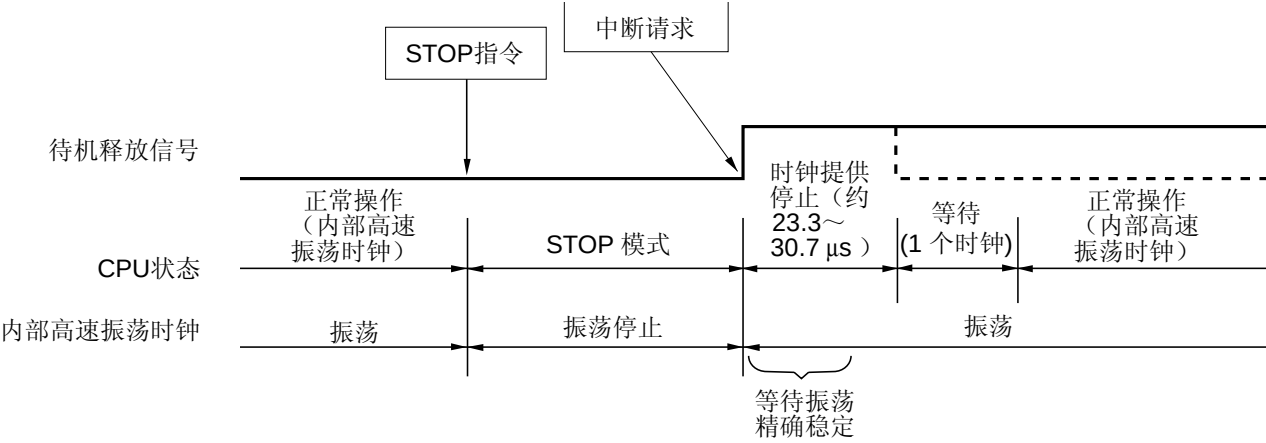
备注 虚线表示释放待机模式的中断请求响应的情况。

图 17-5. 通过产生中断请求释放 STOP 模式(2/2)

(2) 当 CPU 使用高速系统时钟(外部时钟输入)时



(3) 当 CPU 使用内部高速振荡时钟时



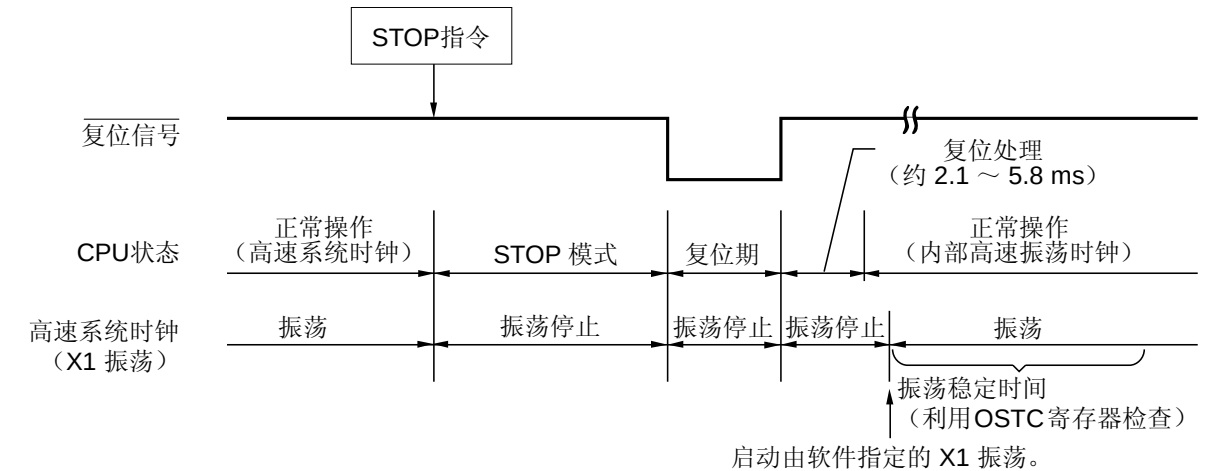
备注 虚线表示释放待机模式的中断请求响应的情况。

(b) 由复位信号的产生释放

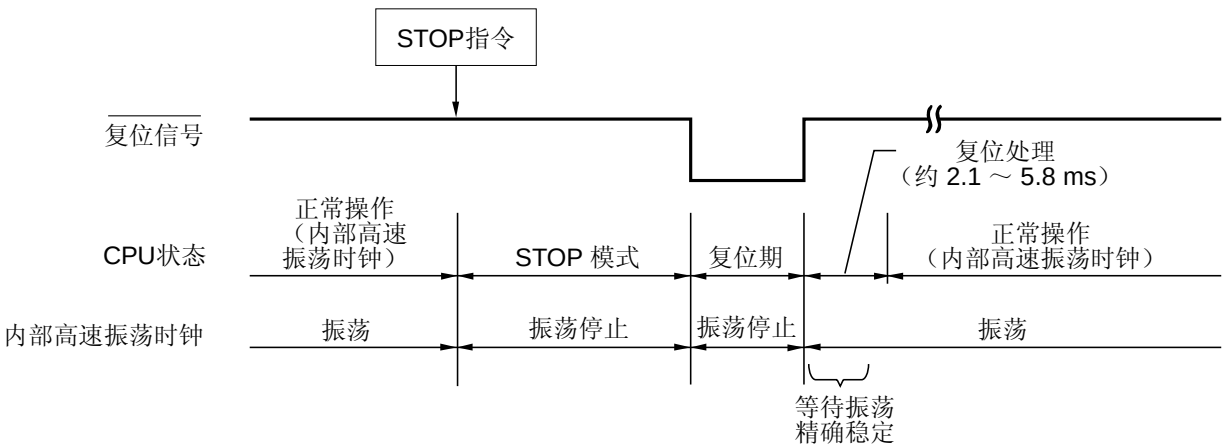
当产生复位信号时，释放 STOP 模式，然后在正常复位操作情况下，程序转向复位向量地址然后执行。

图 17-6. 由复位释放 STOP 模式

(1) 当 CPU 使用高速系统时钟时



(2) 当 CPU 使用内部高速振荡时钟时



备注 f_x : X1 时钟振荡频率

第 18 章 复位功能

以下 5 种操作可产生一个复位信号。

- (1) 由 $\overline{\text{RESET}}$ 引脚输入的外部复位信号
- (2) 由看门狗定时器程序循环检测引起的内部复位
- (3) 通过比较上电清零 (POC) 电路的检测电压和供电电压引起的内部复位
- (4) 内部复位通过供电电压或外部输入引脚的输入电压(EXLVI)同低电压检测电路(LVI)的检测电压比较实现。
- (5) 由执行非法指令引起的内部复位[※]

外部复位与内部复位一样，当产生复位信号时，程序都是从地址 0000H 和 0001H 处开始执行。

当低电平输入到 $\overline{\text{RESET}}$ 引脚、看门狗定时器溢出、POC和LVI电路的电压检测或执行非法指令[※]，都可以使复位有效，且硬件的各项设置状态如表 18-1 和 18-2 所示。在复位信号产生期间或在复位释放后振荡稳定时间期间，每个引脚均为高阻抗，但是P130 除外，它是低电平输出。

当 $\overline{\text{RESET}}$ 引脚输入低电平时，设备复位。当 $\overline{\text{RESET}}$ 引脚输入高电平时，设备从复位状态释放，并在复位处理后使用内部高速振荡时钟执行程序。由看门狗定时器引起的复位自动释放，并在复位处理后使用内部高速振荡时钟执行程序 (参见图 18-2 至 18-4)。复位后当 $V_{DD} \geq V_{POR}$ 或 $V_{DD} \geq V_{LVI}$ 时，由 POC 和 LVI 电路电压检测引起的复位自动释放，在复位操作后使用内部高速振荡时钟执行程序。（参见第 19 章 上电清零电路和第 20 章 低电压检测电路）。

注 当执行 FFH 中的指令码时，产生非法操作。

由非法指令执行引起的复位，不能通过在线仿真器或片上调试仿真器终止。

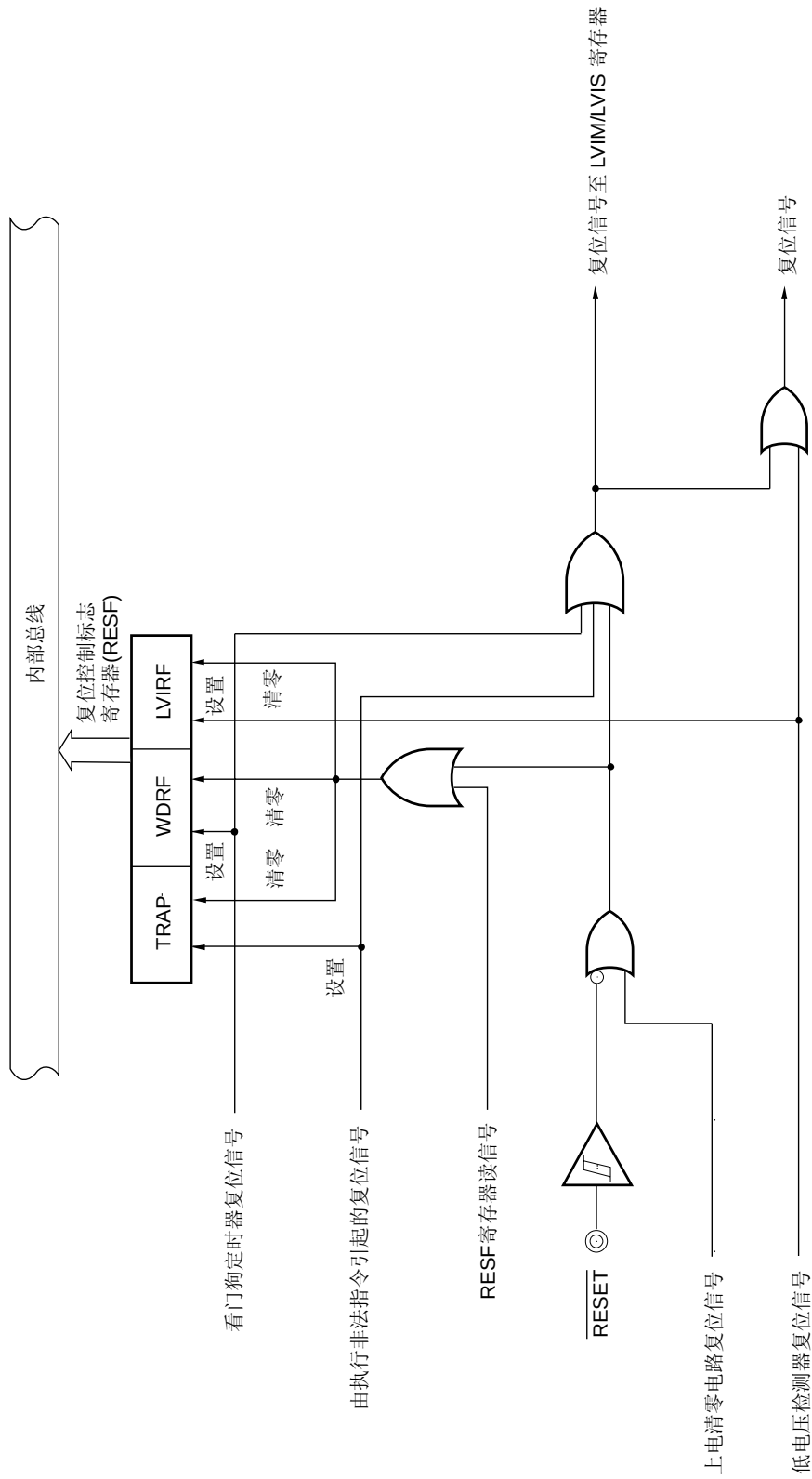
注意事项 1. 对于外部复位，输入 $\overline{\text{RESET}}$ 引脚的低电平的时间至少应为 10 μs 。

（要实现电源应用上的一个外部复位，那么必须持续一个至少 10 μs 的低电平，在此期间供电电压在操作范围内($V_{DD} \geq 1.8 \text{ V}$)。）

2. 复位输入期间，X1 时钟、XT1 时钟、内部高速振荡时钟和内部低速振荡时钟停止振荡。外部主系统时钟输入无效。
3. 在通过复位释放 STOP 模式时，复位输入期间保持 STOP 模式下的 RAM 内容。
4. 当复位有效时，端口引脚 P130 设为低电平输出，其它端口引脚都变成高阻态，因为 SFR 和第二个 SFR 初始化。

备注 V_{POR} : POC 供电上升检测电压

图 18-1. 复位功能的框图



注意事项 LVI 电路内部复位功能不能复位 LVI 电路。

备注

图 18-2. 由 $\overline{\text{RESET}}$ 输入进行复位的时序

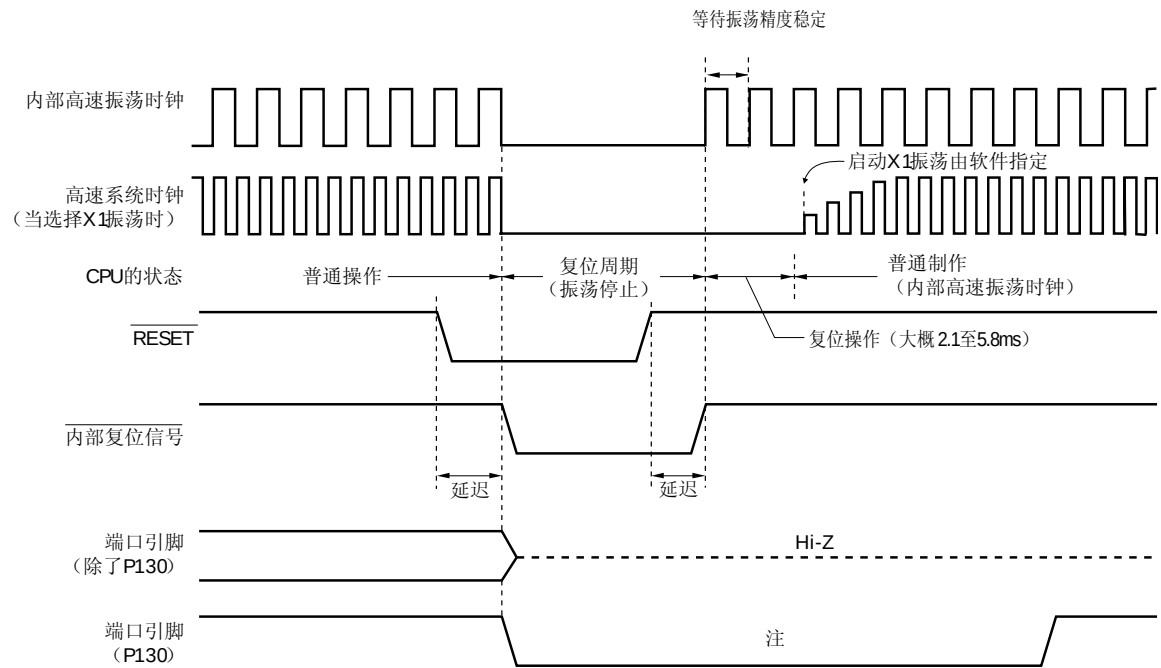
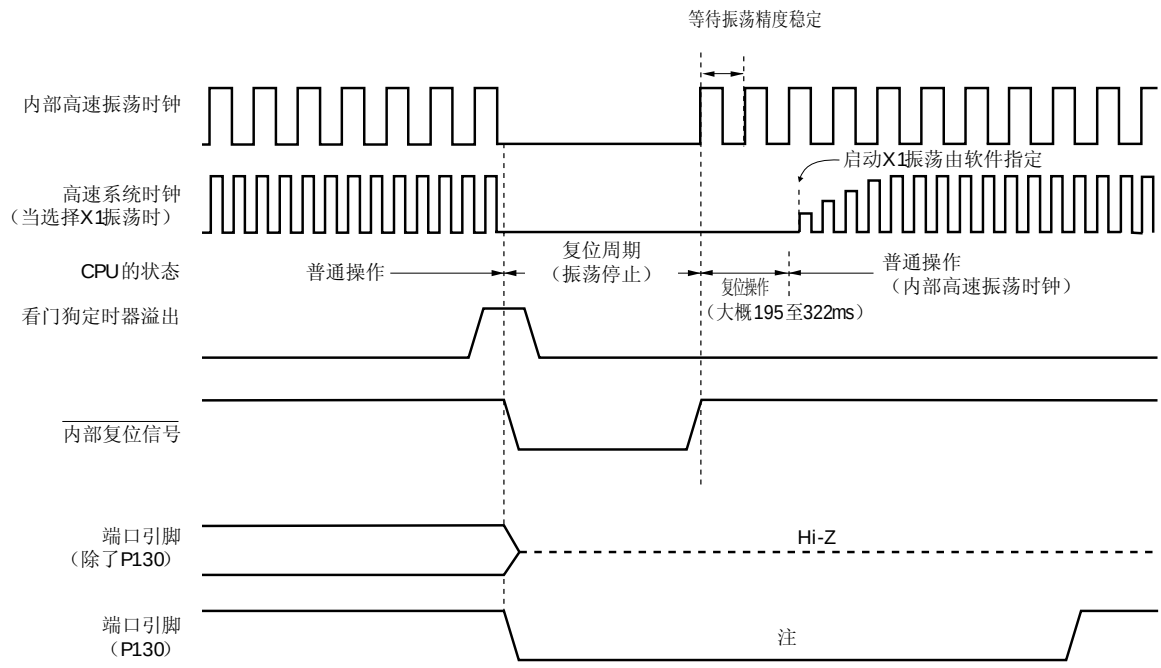


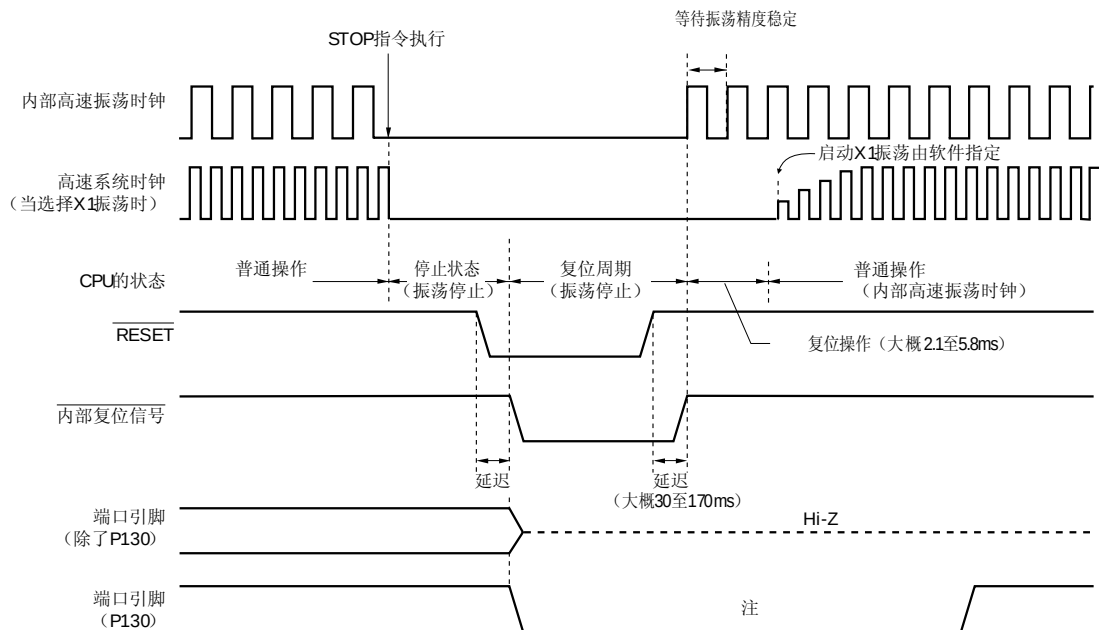
图 18-3. 由于看门狗定时器溢出进行复位的时序



注 当复位有效前 P130 设置为高电平输出，则作为外部设备的复位信号，P130 的输出信号可以虚拟输出。因为当复位有效时，P130 输出低电平。要释放复位信号到外部设备，则由软件设置 P130 为高电平输出。

注意事项 看门狗定时器内部复位也能复位看门狗定时器。

图 18-4. 在 STOP 模式下通过 RESET 输入进行复位的时序



注 当复位有效前 P130 设置为高电平输出，则作为外部设备的复位信号，P130 的输出信号可以虚拟输出。因为当复位有效时，P130 输出低电平。要释放复位信号到外部设备，则由软件设置 P130 为高电平输出。

备注 关于上电清零电路和低电压检测器的复位时序，可参见第 19 章 上电清零电路和第 20 章 低电压检测电路。

表 18-1. 复位期间的操作状态

项 目			复位期间
系统时钟			CPU 时钟 停止
主系统时钟	f _{IH} , f _{IH20}	操作停止	
	f _X	操作停止（X1 和 X2 引脚为输入端口模式）	
	f _{EX}	时钟输入无效（引脚为输入端口模式）	
副系统时钟	f _{XT}	操作停止（XT1 和 XT2 引脚为输入端口模式）	
f _{IL}		操作停止	
CPU			
Flash 存储器			
RAM			
端口（锁存器）			设置 P130 为低电平输出。除 P130 外的端口引脚变成高阻抗状态。
定时器阵列单元 TAU			操作停止
实时计数器 (RTC)			
看门狗定时器			
时钟输出/蜂鸣器输出			
A/D 转换器			
串行阵列单元 (SAU)			
串行接口(IICA)			
乘法器/除法器			
DMA 控制器			
上电清零功能			检测操作可能
低电压检测功能			操作停止（但是，在 LVI 复位时操作继续）
外部中断			操作停止
按键中断功能			

备注

f_{IH} : 内部高速振荡时钟

f_{IH20} : 20 MHz 内部高速振荡时钟

f_X : X1 振荡时钟

f_{EX} : 外部主系统时钟

f_{XT} : XT1 振荡时钟

f_{IL} : 内部低速振荡时钟

表 18-2. 复位响应后的硬件状态(1/3)

硬件		复位响应后 ^{注1}
程序计数器 (PC)		设置复位向量表的内容 (0000H、0001H)
堆栈指针 (SP)		未定义
程序状态字 (PSW)		06H
RAM	数据存储器	不确定 ^{注2}
	通用寄存器	不确定 ^{注2}
处理器模式控制寄存器 (PMC)		00H
端口寄存器 (P0 至 P9, P11 至 P15) (输出锁存)		00H
端口模式寄存器	PM0 至 PM9, PM11, PM12, PM14, PM15	FFH
	PM13	FEH
端口输入模式寄存器 0, 1, 14 (PIM0, PIM1, PIM14)		00H
端口输出模式寄存器 0, 1, 14 (POM0, POM1, POM14)		00H
上拉电阻可选寄存器 (PU0, PU1, PU3 至 PU9, PU11 至 PU14)		00H
时钟操作模式控制寄存器(CMC)		00H
时钟操作状态控制寄存器(CSC)		C0H
系统时钟控制寄存器(CKC)		09H
20 MHz 内部高速振荡控制寄存器(DSCCTL)		00H
振荡稳定时间计数器状态寄存器(OSTC)		00H
振荡稳定时间选择寄存器(OSTS)		07H
噪声滤波器使能寄存器 0 至 2 (NFEN0 至 NFEN2)		00H
外围使能寄存器 0, 1(PER0, PER1)		00H
操作速度模式控制寄存器(OSMC)		00H
定时器阵列单元 (TAU)	定时器数据寄存器 00 至 07, 10 至 13 (TDR00 至 TDR07, TDR10 至 TDR13)	0000H
	定时器模式寄存器 00 至 07, 10 至 13(TMR00 至 TMR07, TMR10 至 TMR13)	0000H
	定时器状态寄存器 00 至 07, 10 至 13 (TSR00 至 TSR07, TSR10 至 TSR13)	0000H
	定时输入选择寄存器 0, 1 (TIS0, TIS1)	00H
	定时器计数器寄存器 00 至 07, 10 至 13 (TCR00 至 TCR07, TCR10 至 TCR13)	FFFFH
	定时器通道使能状态寄存器 0, 1 (TE0, TE1)	0000H
	定时器通道启动触发寄存器 0, 1 (TS0, TS1)	0000H
	定时器通道停止触发寄存器 0, 1 (TT0, TT1)	0000H
	定时器时钟选择寄存器 0, 1 (TPS0, TPS1)	0000H
	定时器输出寄存器 0, 1 (TO0, TO1)	0000H
	定时器输出使能寄存器 0, 1 (TOE0, TOE1)	0000H
	定时器输出电平寄存器 0, 1 (TOL0, TOL1)	0000H
	定时器输出模式寄存器 0, 1 (TOM0, TOM1)	0000H

- 注
1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。
 2. 当在待机模式下执行复位时，即使在复位后预复位状态也将保持。

表 18-2. 复位响应后的硬件状态(2/3)

硬件		复位响应后 ^{注1}
实时计数器	副计数寄存器 (RSUBC)	0000H
	秒计数寄存器 (SEC)	00H
	分钟计数寄存器 (MIN)	00H
	小时计数寄存器 (HOUR)	12H
	星期计数寄存器 (WEEK)	00H
	日计数寄存器 (DAY)	01H
	月计数寄存器 (MONTH)	01H
	年计数寄存器 (YEAR)	00H
	钟表错误修正寄存器 (SUBCUD)	00H
	分钟报警寄存器 (ALARMWM)	00H
	小时报警寄存器 (ALARMWH)	12H
	星期报警寄存器 (ALARMWW)	00H
	控制寄存器 0 (RTCC0)	00H
	控制寄存器 1 (RTCC1)	00H
	控制寄存器 2 (RTCC2)	00H
时钟输出/ 蜂鸣器输出控制 输出控制器	时钟输出选择寄存器 0, 1 (CKS0, CKS1)	00H
看门狗定时器	使能寄存器 (WDTE)	1AH/9AH ^{注2}
A/D 转换器	10 位 A/D 转换结果寄存器 (ADCR)	0000H
	8 位 A/D 转换结果寄存器 (ADCRH)	00H
	模式寄存器 (ADM)	00H
	模式输入通道指定寄存器 (ADS)	00H
	A/D 端口配置 寄存器 (ADPC)	10H
串行阵列单元 (SAU)	串行数据寄存器 00 至 03, 10 至 13 (SDR00 至 SDR03, SDR10 至 SDR13)	0000H
	串行状态寄存器 00 至 03, 10 至 13 (SSR00 至 SSR03, SSR10 至 SSR13)	0000H
	串行标志清除触发寄存器 00 至 03, 10 至 13 (SIR00 至 SIR03, SIR10 至 SIR13)	0000H
	串行模式寄存器 00 至 03, 10 至 13 (SMR00 至 SMR03, SMR10 至 SMR13)	0020H
	串行通信操作设置寄存器 00 至 03, 10 至 13 (SCR00 至 SCR03, SCR10 至 SCR13)	0087H
	串行通道使能状态寄存器 0, 1 (SE0, SE1)	0000H
	串行通道启动寄存器 0, 1 (SS0, SS1)	0000H
	串行通道停止寄存器 0, 1 (ST0, ST1)	0000H
	串行时钟选择寄存器 0, 1 (SPS0, SPS1)	0000H
	串行输出寄存器 0, 1 (SO0, SO1)	0F0FH
	串行输出使能寄存器 0, 1 (SOE0, SOE1)	0000H
	串行输出寄存器 0, 1 (SOL0, SOL1)	0000H

- 注 1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。
2. WDTE 的复位数值由选项字节设置确定。

表 18-2. 复位响应后的硬件状态(3/3)

硬件		复位响应后状态 ^{註1}
串行接口 IICA	IICA 移位寄存器(IICA)	00H
	IICA 状态寄存器 (IICS)	00H
	IICA 标志寄存器 (IICF)	00H
	IICA 控制寄存器 0 (IICCTL0)	00H
	IICA 控制寄存器 1 (IICCTL1)	00H
	IICA 低电平宽度设置寄存器 (IICWL)	FFH
	IICA 高电平宽度设置寄存器 (IICWH)	FFH
	从地址寄存器 (SVA)	00H
乘法器/除法器	乘法/除法数据寄存器 A (L) (MDAL)	0000H
	乘法/除法数据寄存器 A (H) (MDAH)	0000H
	乘法/除法数据寄存器 B (L) (MDBL)	0000H
	乘法/除法数据寄存器 B (H) (MDBH)	0000H
	乘法/除法数据寄存器 C (L) (MDCL)	0000H
	乘法/除法数据寄存器 C(H)(MDCH)	0000H
	乘法/除法控制寄存器 (MDUC)	00H
按键中断	按键返回模式寄存器(KRM)	00H
复位功能	复位控制标志寄存器(RESF)	不确定 ^{註2}
低电压检测器	低电压检测寄存器 (LVIM)	00H ^{註3}
	低电压检测电平选择寄存器 (LVIS)	0EH ^{註2}
稳压器	稳压器模式控制寄存器 (RMC)	00H
DMA 控制器	SFR 地址寄存器 0, 1 (DSA0, DSA1)	00H
	RAM 址寄存器 0L, 0H, 1L, 1H (DRA0L, DRA0H, DRA1L, DRA1H)	00H
	字节计数寄存器 0L, 0H, 1L, 1H (DBC0L, DBC0H, DBC1L, DBC1H)	00H
	模式控制寄存器 0, 1 (DMC0, DMC1)	00H
	操作控制寄存器 0, 1 (DRC0, DRC1)	00H
中断	请求标志寄存器 0L、0H、1L、1H、2L、2H (IF0L、IF0H、IF1L、IF1H、IF2L、IF2H)	00H
	屏蔽标志寄存器 0L、0H、1L、1H、2L、2H (MK0L、MK0H、MK1L、MK1H、MK2L、MK2H)	FFH
	优先级指定标志寄存器 00L、00H、01L、01H、02L、02H、10L、10H、11L、11H、12L、12H (PR00L、PR00H、PR01L、PR01H、PR10L、PR10H、PR11L、PR11H、PR02L、PR02H、PR12L、PR12H)	FFH
	外部中断上升沿使能寄存器 0, 1 (EGP0, EGP1)	00H
	外部中断下降沿使能寄存器 0, 1 (EGN0, EGN1)	00H
BCD 纠错电路	BCD 修正结果寄存器 (BCDADJ)	未定义

(注列在下页中)

- 注
- 1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。
 - 2. 这些值依据复位源的不同而变化。

复位源		RESET 输入	由 POC 复位	由执行非法指令复位	由 WDT 复位	由 LVI 复位
寄存器						
RESF	TRAP 位	清除 (0)	清除 (0)	设置为 (1)	保持	保持
	WDRF 位			保持	设置为 (1)	保持
	LVIRF 位			保持	保持	设置为 (1)
LVIS		清除 (0EH)	清除 (0EH)	清零 (0EH)	清零 (0EH)	保持

- 3. 此数值依据复位源和选项字节而不同。

18.1 确认复位源的寄存器

许多内部复位产生源存在于 78K0R/KG3-L 中。复位控制标志寄存器(RESF)常常用来存储产生复位请求的源。可由 8 位存储器操作指令读取 RESF。

RESET 输入、由 (POC) 电路引起的复位输入以及读取 RESF 可将 RESF 清零 00H。

图 18-5. 复位控制标志寄存器 (RESF) 的格式

地址: FFFA8H 复位后: 未定义 R

符号

7

6

5

4

3

2

1

0

RESF

TRAP ^{#1}	未定义	未定义	WDRF ^{#1}	未定义	未定义	未定义	LVIRF ^{#1}

TRAP	由执行非法指令引起内部复位请求 ^{#2}
0	不产生内部复位请求，或对 RESF 清零。
1	产生内部复位请求。

WDRF	看门狗定时器产生的内部复位请求(WDT)
0	不产生内部复位请求，或对 RESF 清零。
1	产生内部复位请求。

LVIRF	由低电压检测电路（LVI）产生的内部复位请求
0	不产生内部复位请求，或对 RESF 清零。
1	产生内部复位请求。

- 注
1. 复位后的值依据复位源而变化。
 2. 当执行 FFH 中的指令码时, 产生非法操作。
由非法指令执行引起的复位, 不能通过在线仿真器或片上调试仿真器终止。

- 注意事项
1. 不要用 1 位存储器操作指令读取数据。
 2. 不要仅仅基于 RESF 寄存器 8 位数据的读取值就作出判断, 因为 TRAP、WDRF 和 LVIRF 之外的位都是不确定的。
 3. 当使用 LVI 默认开始功能 (000C1H 的第 0 位(LVIOFF) = 0) 时, 依据上电波形, LVIRF 标志可能一开始就变成 1。

产生复位请求时的 RESF 状态如表 18-3 所示。

表 18-3. 产生复位请求时的 RESF 状态

复位源 标志	RESET 输入	由 POC 复位	由执行非法指令复位	由 WDT 复位	由 LVI 复位
TRAP	清除 (0)	清除 (0)	设置为 (1)	保持	保持
WDRF			保持	设置为 (1)	保持
LVIRF			保持	保持	设置为 (1)

第 19 章 上电清零电路

19.1 上电清零电路的功能

上电清零电路 (POC) 有以下功能。

- 在上电时产生内部复位信号。
当供电电压 (V_{DD}) 超过 $1.61\text{ V} \pm 0.09\text{ V}^{\text{注}}$ 时, 释放复位信号。

注意事项 如果低电压检测器 (LVI) 由选项字节默认设置为 ON, 那么复位信号直到供电电压 (V_{DD}) 超过 $2.07\text{ V} \pm 0.2\text{ V}^{\text{注}}$ 才释放。

- 比较供电电压 (V_{DD}) 和检测电压 ($V_{PDR} = 1.59\text{ V} \pm 0.09\text{ V}^{\text{注}}$), 并在 $V_{DD} < V_{PDR}$ 时产生内部复位信号。

注 这些均为初步数据, 可能会改变。

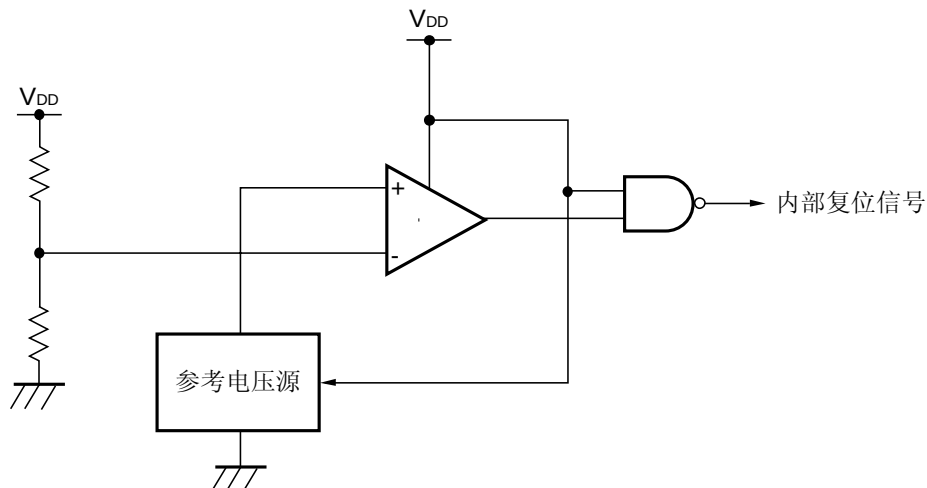
注意事项 如果在 POC 电路产生一个内部复位信号, 则复位控制标志寄存器 (RESF) 的 TRAP、WDRF 和 LVIRF 清零。

备注 此产品集成了多种可以产生内部复位信号的硬件功能。复位控制标志寄存器 (RESF) 中有一个标志用于指示复位源, 用于看门狗定时器 (WDT)、低电压检测电路 (LVI)、或者执行非法指令产生内部复位信号时。当 WDT 或 LVI 产生内部复位信号时或执行非法指令时, 不对 RESF 清零(00H), 标志位设置为 1。
需要了解 RESF 的详细信息, 可参见第 18 章 复位功能。

19.2 上电清零电路的配置

上电清零电路的框图如图 19-1 所示。

图 19-1. 上电清零电路的框图



19.3 上电清零电路的操作

- 在上电时产生一个内部复位信号。当供电电压 (V_{DD}) 超过检测电压 ($V_{PDR} = 1.61\text{ V} \pm 0.09\text{ V}^{\text{B}}$) 时，释放复位信号。

注意事项 如果低电压检测器 (LVI) 由选项字节默认设置为 ON，那么复位信号直到供电电压 (V_{DD}) 超过 $2.07\text{ V} \pm 0.2\text{ V}^{\text{B}}$ 才释放。

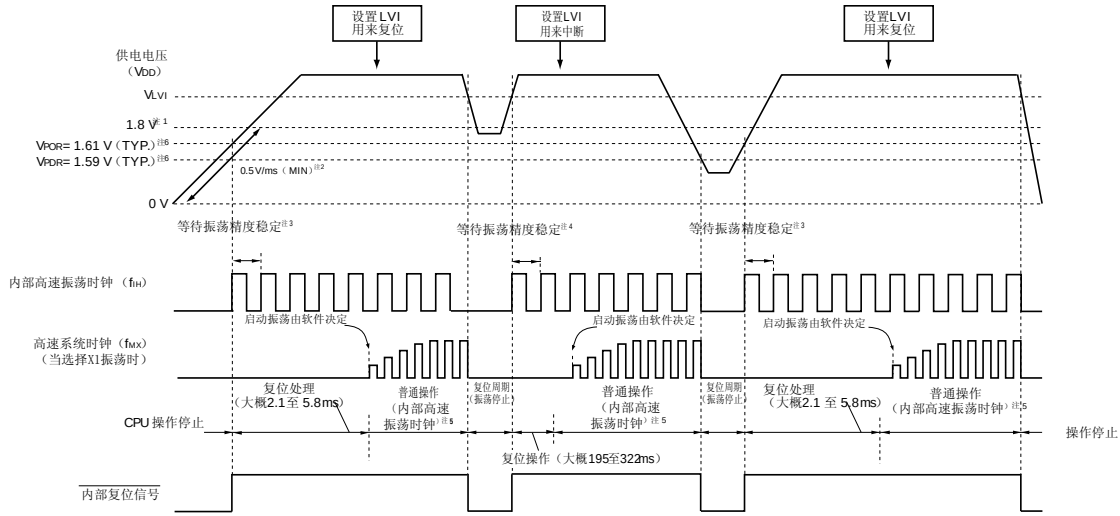
- 比较供电电压 (V_{DD}) 和检测电压 ($V_{PDR} = 1.59\text{ V} \pm 0.09\text{ V}^{\text{B}}$)。在 $V_{DD} < V_{PDR}$ 时，产生内部复位信号。

注 这些均为初步数据，可能会改变。

由上电清零电路和低电压检测电路产生的内部复位信号的时序如下所示。

图 19-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序 (1/2)

(1) 当上电之后 LVI 为 OFF 时(选项字节: LVIOFF = 1)



- 注
1. 保证操作的电压范围: $1.8V \leq V_{DD} \leq 5.5V$ 。当供电电压下降时,为使低于 $1.8V$ 的状态为复位状态,可以使用低电压检测电路的复位功能,或通过 **RESET** 引脚输入低电平。
 2. 如果上电后电压升至 $1.8V$ 期间的速率低于 $0.5V/ms$ (最小值),在电压达到 $1.8V$ 之前向 **RESET** 引脚输入一个低电平,或者通过选项字节将 LVI 设置为 ON(选项字节: LVIOFF = 0)。
 3. 复位处理时间,比如等待内部电压稳定的时间,包括内部高速振荡时钟的振荡精确稳定时间。
 4. 内部复位处理时间包括内部高速振荡时钟的振荡精确稳定时间。
 5. 可以选择内部高速振荡时钟和高速系统时钟或副系统时钟作为 CPU 时钟。如果要使用 X1 时钟,可以通过 OSTC 确认是否经历了振荡稳定时间。而如果要使用 XT1 时钟,可以通过定时器功能来确认是否经历了振荡稳定时间。
 6. 这只是初步数据,可能会变化。

注意事项 复位状态释放后,通过软件设置低电压检测电路(参考第 20 章 低电压检测电路)。

备注

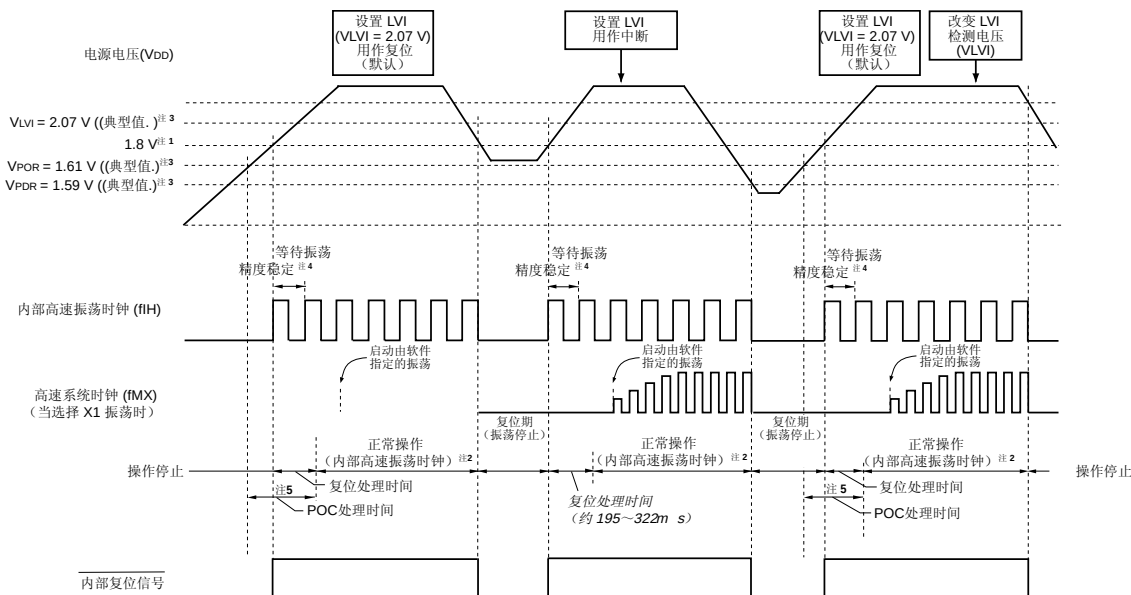
V_{LVI} : LVI 检测电压

V_{POR} : POC 供电上升检测电压

V_{PDR} : POC 供电下降检测电压

图 19-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序 (2/2)

(2) 当上电之后 LVI 为 ON 时 (选项字节: LVIOFF = 0)



- 注
1. 保证操作的电压范围: $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ 。当供电电压下降时, 为使低于 1.8 V 的状态为复位状态, 可以使用低电压检测电路的复位功能, 或通过 **RESET** 引脚输入低电平。
 2. 可以选择内部高速振荡时钟和高速系统时钟或副系统时钟作为 CPU 时钟。如果要使用 X1 时钟, 可以通过 OSTC 确认是否经历了振荡稳定时间。而如果要使用 XT1 时钟, 可以通过定时器功能来确认是否经历了振荡稳定时间。
 3. 这些均为初步数据, 可能会改变。
 4. 内部复位处理时间包括内部高速振荡时钟的振荡精确稳定时间。
 5. 下面的时间是到达 POC 检测电压 (1.61 V (典型值)) 和开始正常操作之间需要的时间。
 - 当从 1.61 V (典型值) 达到 2.07 V (典型值) 的时间小于 5.8 ms :
在 1.61 V (典型值) 和启动正常操作之间需要 2.1 至 6.2 ms 的 POC 处理时间。
 - 当从 1.61 V (典型值) 达到 2.07 V (典型值) 的时间大于 5.8 ms :
从 2.07 V (典型值) 到开始正常操作之间, 需要 195 至 $322\text{ }\mu\text{s}$ 的复位处理时间。

注意事项 复位状态释放后, 通过软件设置低电压检测电路(参考第 20 章 低电压检测电路)。

备注

- VLVI: LVI 检测电压
- VPOR: POC 供电上升检测电压
- VPDR: POC 供电下降检测电压

19.4 上电清零电路使用注意事项

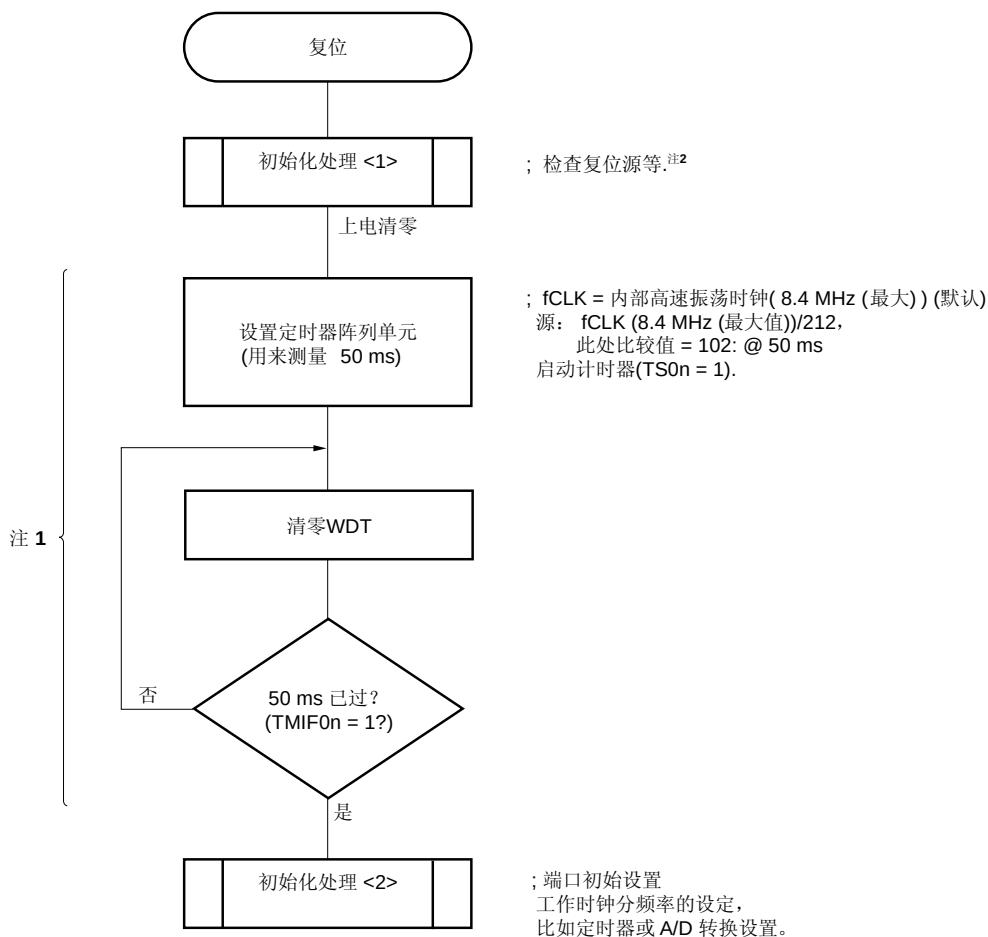
在系统中，如果供电电压 (V_{DD}) 在接近 POC 检测电压 (V_{POR} , V_{PDR}) 时的一段时间内产生波动，则系统可能会反反复复位和从复位状态释放。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动所需要的时间。

<方法>

在释放复位信号后，通过一个使用定时器的软件计数器，等待系统供电电压的波动期，然后对端口初始化。

图 19-3. 复位释放后软件处理过程示例 (1/2)

- 如果供电电压在接近 POC 检测电压时的波动时间为 50 ms 或更少

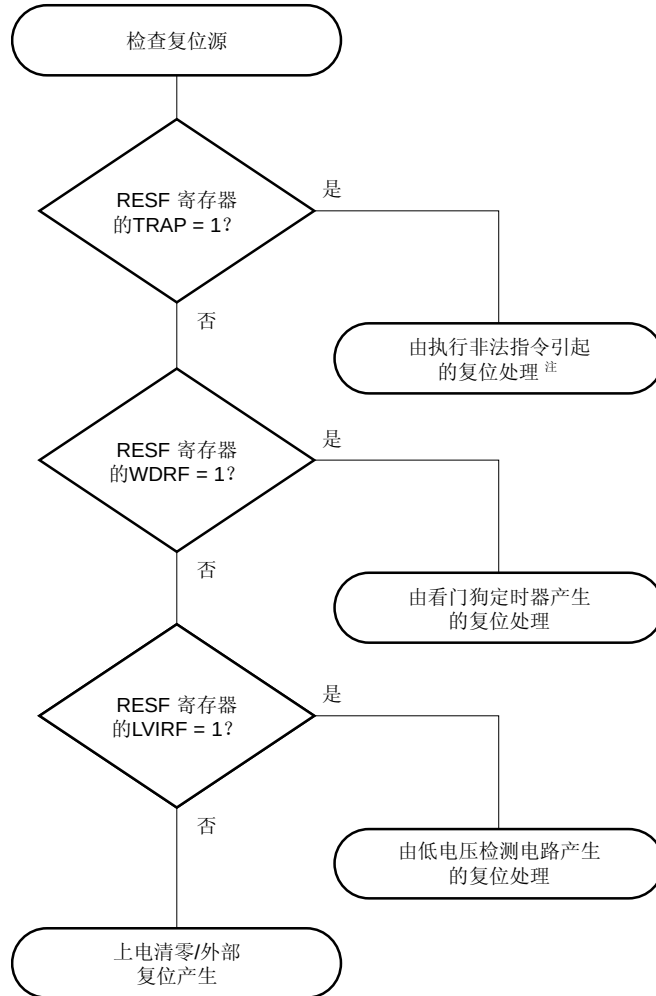


- 注 1. 如果在此期间再次产生复位，则不启动初始化操作<2>。
2. 流程图显示在下页。

备注 n = 0 至 7

图 19-3. 复位释放后软件处理过程示例 (2/2)

• 检查复位源



注 当执行 FFH 中的指令码时，产生非法操作。
由执行非法指令引起的复位不能通过在线仿真器或片上调试仿真器调试终止。
调试仿真器。

第 20 章 低电压检测电路

20.1 低电压检测电路的功能

低电压检测电路 (LVI) 有以下功能。

- LVI电路比较供电电压 (V_{DD}) 和检测电压 (V_{LVI})，或比较从外部输入引脚 (EXLVI) 的输入电压和检测电压 ($V_{EXLVI} = 1.21\text{ V} \pm 0.1\text{ V}^{\text{※}}$)，并产生内部复位信号或内部中断信号。
- 低电压检测电路 (LVI) 可由选项字节默认设置为ON。如果设置为ON，供电电压从POC检测电压 ($V_{POR}=1.61\text{V}$ (典型值))或更低电压开始上电时，当供电电压 (V_{DD}) < 检测电压 ($V_{LVI}=2.07\text{V} \pm 0.2\text{V}^{\text{※}}$) 时，会产生内部复位信号。之后，当供电电压 (V_{DD}) < 检测电压 ($V_{LVI} = 2.07\text{ V} \pm 0.1\text{ V}^{\text{※}}$) 时，产生内部复位信号。
- 可由软件选择检测供电电压 (V_{DD}) 或外部输入引脚的输入电压 (EXLVI)。
- 可由软件选择复位或中断操作。
- 可通过软件改变供电电压的检测电平 (V_{LVI} , 16 电平)。
- 在 STOP 模式下可操作。

注 这只是初步数据，可能会变化。

依据软件设置，可产生以下复位和中断信号。

供电电压 (V_{DD}) 的电平检测选择 (LVISEL = 0)		从外部输入引脚(EXLVI) 的输入电压检测电平选择 (LVISEL = 1)	
选择复位 (LVIMD = 1).	选择中断 (LVIMD = 0).	选择复位 (LVIMD = 1).	选择中断 (LVIMD = 0).
当 $V_{DD} < V_{LVI}$ 时，产生内部复位信号，并且当 $V_{DD} \geq V_{LVI}$ 时，释放复位信号。	当 V_{DD} 降至低于 V_{LVI} ($V_{DD} < V_{LVI}$) 时，产生内部中断信号，或者当 V_{DD} 变为 V_{LVI} 或更高时 ($V_{DD} \geq V_{LVI}$)。	当 $EXLVI < V_{EXLVI}$ 时，产生内部复位信号，并且当 $EXLVI \geq V_{EXLVI}$ 时，释放复位信号。	当 $EXLVI$ 降至低于 V_{EXLVI} ($EXLVI < V_{EXLVI}$) 时，产生内部中断信号，或者当 $EXLVI$ 变为 V_{EXLVI} 或更高时 ($EXLVI \geq V_{EXLVI}$)。

备注 LVISEL: 低电压检测寄存器 (LVIM) 的第 2 位
LVIMD: LVIM 的第 1 位。

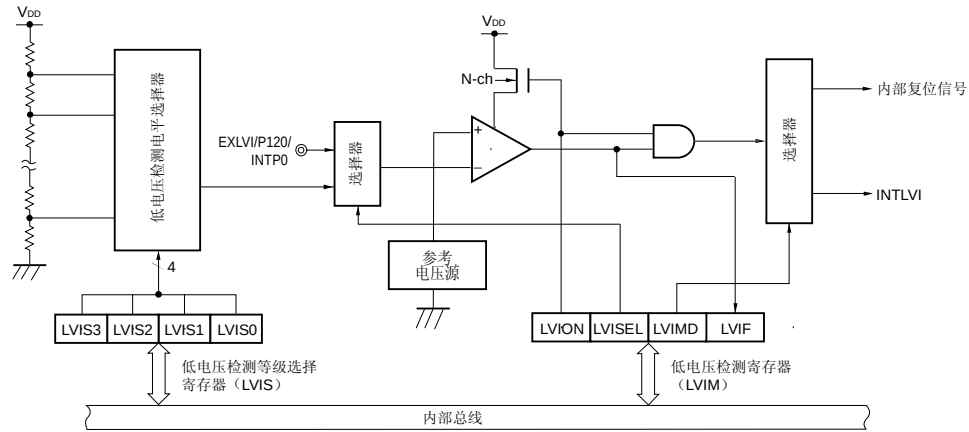
当低电压检测电路运行时，供电电压或者外部输入引脚的输入电压可以通过读取低电压检测标志(LVIM 的位 0: LVIF) 得知其是否高于或低于检测电平。

当低电压检测电路用作产生复位时，如果复位产生，复位控制标志寄存器 (RESF) 的第 0 位 (LVIRF) 被置 1。需要了解 RESF 的详细信息，可参见第 18 章 复位功能。

20.2 低电压检测电路的配置

低电压检测电路的框图如图 20-1 所示。

图 20-1. 低电压检测电路的框图



20.3 控制低电压检测电路的寄存器

低电压检测电路由以下寄存器控制。

- 低电压检测寄存器 (LVIM)
- 低电压检测电平选择寄存器 (LVIS)
- 端口模式寄存器 12 (PM12)

(1) 低电压检测寄存器 (LVIM)

该寄存器用于设置低电压检测和操作模式。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位后这个寄存器的值为 00H。

图 20-2. 低电压检测寄存器 (LVIM) 的格式

地址: FFFA9H 复位后: 00H^{注1} R/W^{注2}

符号	<7>	6	5	4	3	<2>	<1>	<0>
LVIM	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION ^{注3,4}	使能低电压检测操作
0	禁止操作
1	使能操作

LVISEL ^{注3}	电压检测的选择
0	检测供电电压 (V_{DD}) 的电平
1	检测外部输入引脚的输入电压 ($EXLVI$) 的电平

LVIMD ^{注3}	低电压检测操作模式(中断/复位)的选择
0	- LVISEL = 0: 当供电电压 V_{DD} 降至低于检测电压 V_{LVI} ($V_{DD} < V_{LVI}$) 时, 产生内部中断信号, 或者当 V_{DD} 变为 V_{LVI} 或更高时 ($V_{DD} \geq V_{LVI}$)。 - LVISEL = 1: 当外部输入引脚($EXLVI$)的输入电压下降低于检测电压(V_{EXLVI}) ($EXLVI < V_{EXLVI}$)或当 $EXLVI$ 成为 V_{EXLVI} 或更高($EXLVI \geq V_{EXLVI}$)时将产生中断信号。
1	- LVISEL = 0: 当供电电压 $V_{DD} < V_{LVI}$ 时, 产生内部复位信号, 并且当 $V_{DD} \geq V_{LVI}$ 时, 释放复位信号。 - LVISEL = 1: 当从外部输入引脚来的输入电压 $EXLVI < V_{EXLVI}$ 时, 产生内部复位信号, 并且当 $EXLVI \geq V_{EXLVI}$ 时, 释放复位信号。

LVIF	低电压检测标
0	- LVISEL = 0: 供电电压(V_{DD})大于检测电压(V_{LVI})或当禁止 LVI 操作时 - LVISEL = 1: 外部输入引脚的输入电压($EXLVI$) $\geq$ 检测电压(V_{EXLVI}), 或当禁止操作时
1	- LVISEL = 0: 供电电压 (V_{DD}) $<$ 检测电压 (V_{LVI}) - LVISEL = 1: 外部输入引脚的输入电压($EXLVI$) $<$ 检测电压(V_{EXLVI})

- 注
- 复位数值依据复位源和选项字节的设置而改变。
LVI 复位此寄存器不清零(00H)。
当除去 LVI 的复位信号应用时, 如果选项字节 LVIOFF = 0 则设置为“82H”, 如果选项字节 LVIOFF = 1 则设置为“00H”。
 - 第 0 位只读。
 - 在产生除 LVI 复位以外的复位时, LVION、LVIMD 和 LVISEL 被清零。LVI 复位时, 这些位不被清零。

注 4. 当 LVION 设置为 1 时, LVI 电路中比较器的操作就开始。在 LVION 设置为 1 和当用 LVIF 确认电压之间, 使用软件进行下列时间段的等待。

- 运行稳定时间 (10 μ s (最大值))
- 最小脉冲宽度 (200 μ s (最小值))

在此期间不管电压数值为何, LVIF 的数值可以设置/清零, 因此就不能使用。同时, 在此期间 LVIIF 中断请求标志设置为 1。

- 注意事项
1. 为了停止 LVI, 通过使用一位存储器操作指令确保清除(0)LVION。
 2. 外部输入引脚的输入电压 (EXLVI) 必须满足 $EXLVI < V_{DD}$
 3. 当 LVI 用于中断模式(LVIMD = 0)且 LVISEL 设为 0 时, 将产生用于禁止 LVI 操作(清除 LVION)的中断请求信号(INTLVI), 此时供电电压(V_{DD})小于或等于检测电压(V_{LVI}) (若 LVISEL = 1, 则外部输入引脚(EXLVI)的输入电压小于或等于检测电压(V_{EXLVI})), 并可能设置 LVIIF 至 1。
 4. 写入此寄存器之后读取 LVIM, 保证一个或更多时钟的时间。

(2) 低电压检测电平选择寄存器 (LVIS)

该寄存器用于选择低电压检测电平。
可由 1 位或 8 位存储器操作指令设置该寄存器。
复位信号产生输入，设置此寄存器为 0EH。

图 20-3. 低电压检测电平选择寄存器 (LVIS) 的格式

地址: FFFAAH 复位后: 0EH^{注1} R/W

符号	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	检测电平
0	0	0	0	$V_{LV10} (4.22 \pm 0.1 \text{ V})^{\text{注}2}$
0	0	0	1	$V_{LV11} (4.07 \pm 0.1 \text{ V})^{\text{注}2}$
0	0	1	0	$V_{LV12} (3.92 \pm 0.1 \text{ V})^{\text{注}2}$
0	0	1	1	$V_{LV13} (3.76 \pm 0.1 \text{ V})^{\text{注}2}$
0	1	0	0	$V_{LV14} (3.61 \pm 0.1 \text{ V})^{\text{注}2}$
0	1	0	1	$V_{LV15} (3.45 \pm 0.1 \text{ V})^{\text{注}2}$
0	1	1	0	$V_{LV16} (3.30 \pm 0.1 \text{ V})^{\text{注}2}$
0	1	1	1	$V_{LV17} (3.15 \pm 0.1 \text{ V})^{\text{注}2}$
1	0	0	0	$V_{LV18} (2.99 \pm 0.1 \text{ V})^{\text{注}2}$
1	0	0	1	$V_{LV19} (2.84 \pm 0.1 \text{ V})^{\text{注}2}$
1	0	1	0	$V_{LV110} (2.68 \pm 0.1 \text{ V})^{\text{注}2}$
1	0	1	1	$V_{LV111} (2.53 \pm 0.1 \text{ V})^{\text{注}2}$
1	1	0	0	$V_{LV112} (2.38 \pm 0.1 \text{ V})^{\text{注}2}$
1	1	0	1	$V_{LV113} (2.22 \pm 0.1 \text{ V})^{\text{注}2}$
1	1	1	0	$V_{LV114} (2.07 \pm 0.1 \text{ V})^{\text{注}2}$
1	1	1	1	$V_{LV115} (1.91 \pm 0.1 \text{ V})^{\text{注}2}$
其它值				设置禁止

- 注 1. 复位数值依据复位源而改变。
 如果 LVIS 寄存器被 LVI 设置，它不会复位而是保持当前值。若非 LVI 的复位有效，则该寄存器的值复位至“0EH”。
2. 这些均为初步数据，可能会改变。

注意事项 1. 确保将位 4 至 7 清“0”。

- 注意事项 2. 按照下列的方法之一可改变 LVIS 的数值。
- 当停止 LVI 之后改变数值
 - <1> 停止 LVI (LVION = 0)。
 - <2> 改变 LVIS 寄存器。
 - <3> 设置中断模式 (LVIMD = 0)。
 - <4> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <5> 使能 LVI 操作 (LVION = 1)。
 - <6> 在取消 LVI 中断屏蔽之前(LVIMK = 0)，用软件清除 LVIIF 标志因为当使能 LVI 工作时
 - 当设置用作中断模式 (LVIMD = 0) 后，改变数值
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 设置中断模式 (LVIMD = 0)。
 - <3> 改变 LVIS 寄存器。
 - <4> 在取消 LVI 中断屏蔽之前(LVIMK = 0)，用软件清除 LVIIF 标志，因为当 LVIS 寄存器改变时会设置它。
3. 当检测到外部输入引脚的输入电压 (EXLVI) 时，检测电压 (V_{EXLVI}) 恒定。因此，不必设置 LVIS。
4. 写入此寄存器之后读取 LVIM，保证一个或更多时钟的时间。

(3) 端口模式寄存器 12 (PM12)

当使用 P120/EXLVI/INTP0 引脚进行外部低电压检测电压输入时，设置 PM120=1。此时，P120 的输出锁存器的值可以为 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM12。

复位信号的产生将此寄存器设置为 FFH。

图 20-4. 端口模式寄存器 12 (PM12) 的格式

地址: FFF2CH 复位后: FFH R/W							
符号	7	6	5	4	3	2	1 0
PM12	1	1	1	1	1	1	1 PM120

PM120	P120 引脚 I/O 模式选择
0	输出模式(输出缓冲器开)
1	输入模式(输出缓冲器关)

20.4 低电压检测电路的操作

低电压检测电路能够用于以下两种模式。

(1) 用于复位 (LVIMD = 1)

- 如果 LVISEL = 0，比较供电电压 (V_{DD}) 与检测电压 (V_{LVI})，当 V_{DD} < V_{LVI} 时产生内部复位信号，而当 V_{DD} ≥ V_{LVI} 时释放内部复位。
- 如果 LVISEL = 1，比较外部输入引脚的输入电压 (EXLVI) 和检测电压 (V_{EXLVI})，在 EXLVI < V_{EXLVI} 时产生内部复位信号，而在 EXLVI ≥ V_{EXLVI} 时释放内部复位。

备注 低电压检测电路 (LVI) 可由选项字节默认设置为 ON。如果设置为 ON，供电电压从 POC 检测电压 (V_{POR}=1.61V (典型值)) 或更低电压开始上电时，当供电电压 (V_{DD}) < 检测电压 (V_{LVI}=2.07V±0.2V[Ⓓ]) 时，会产生内部复位信号。之后，当供电电压 (V_{DD}) < 检测电压 (V_{LVI} = 2.07 V ±0.1 V[Ⓓ]) 时，产生内部复位信号。

(2) 用作中断 (LVIMD = 0)

- 如果 LVISEL = 0，比较供电电压 (V_{DD}) 与检测电压 (V_{LVI})。当 V_{DD} 降至低于 V_{LVI} (V_{DD} < V_{LVI}) 时，或者当 V_{DD} 变为 V_{LVI} 或更高时 (V_{DD} ≥ V_{LVI})，产生中断信号 (INTLVI)。
- 如果 LVISEL = 1，比较外部输入引脚的输入电压 (EXLVI) 和检测电压 (V_{EXLVI} = 1.21V±0.1V[Ⓓ])。当 EXLVI 降至低于 V_{EXLVI} (EXLVI < V_{EXLVI}) 时，或者当 EXLVI 变为 V_{EXLVI} 或更高时 (EXLVI ≥ V_{EXLVI})，产生中断信号 (INTLVI)。

注 这只是初步数据，可能会变化。

当低电压检测电路运行时，供电电压或者外部输入引脚的输入电压可以通过读取低电压检测标志 (LVIM 的位 0: LVIF) 得知其是否高于或低于检测电平。

备注 LVIMD: 低电压检测寄存器 (LVIM) 的第 1 位
LVISEL: LVIM 的第 2 位。

20.4.1 当用于复位

(1) 当检测供电电压 (V_{DD}) 等级时

(a) 当 LVI 默认启动功能 ($LVIOFF = 1$) 设置停止时

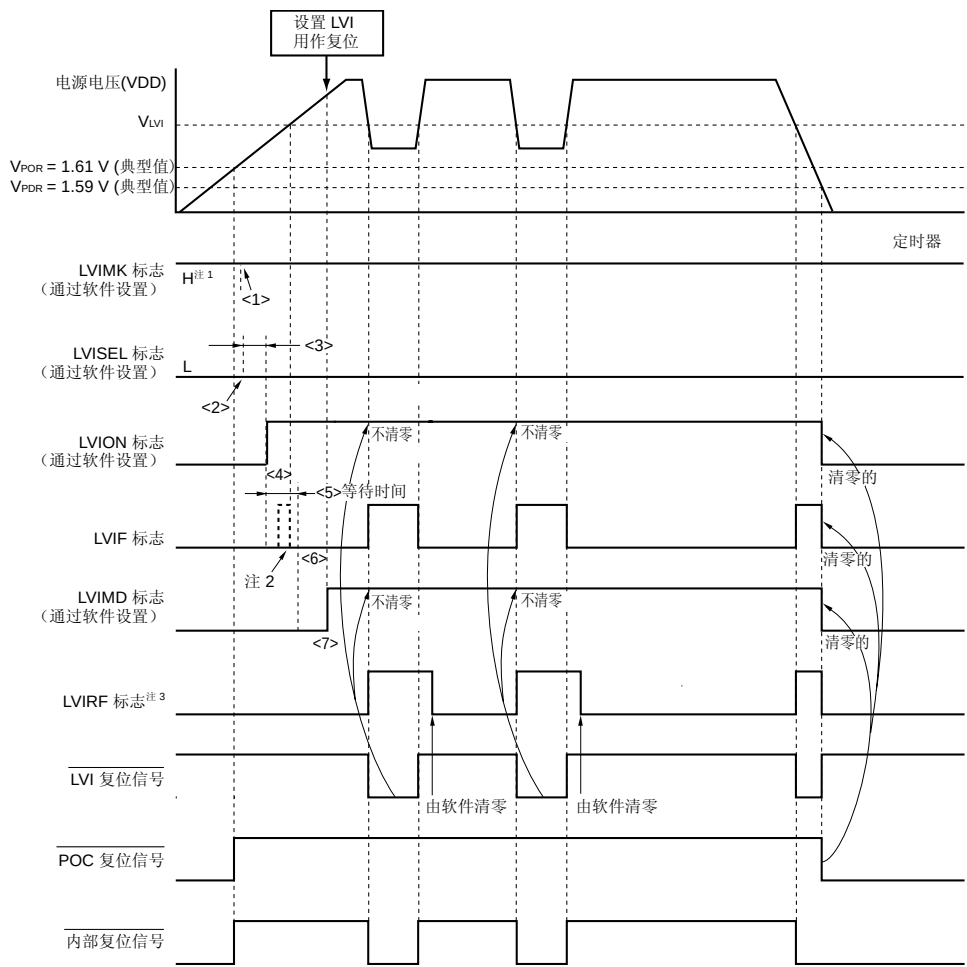
- 当启动操作时
 - <1> 屏蔽 LVI 中断 ($LVIMK = 1$).
 - <2> 将低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清零(检测供电电压 (V_{DD}) 等级)(默认值)。
 - <3> 使用低电压检测电平选择寄存器 (LVIS) 的第 3 至 0 ($LVIS3$ 至 $LVIS0$) 位设置检测电压。
 - <4> 设置 LVIM 的第 7 位 ($LVION$) =1(使能 LVI 的操作)。
 - <5> 使用软件等待下列时间段(总计 $210 \mu s$).
 - 运行稳定时间 ($10 \mu s$ (最大值))
 - 最小脉冲宽度 ($200 \mu s$ (最小值))
 - <6> 等到 LVIM 的第 0 位 ($LVIF$) 检查出(供电电压 (V_{DD}) $\geq$ 检测电压(V_{Lvi}))
 - <7> 设置 LVIM 的第 1 位 ($LVIMD$) 为 1(当检测到电平时, 产生复位)。

图 20-5 显示了由低电压检测电路产生的内部复位信号的时序。 该时序图中的标号与上面的<1>至<7>对应。

- 注意事项**
1. 务必执行<1>。当 $LVIMK = 0$ 时, 在过程<4>之后, 立即发生中断。
 2. 当 $LVIMD=1$ 时如果供电电压(V_{DD}) 大于等于检测电压 (V_{Lvi}), 则不会产生内部复位信号。

- 当停止操作时
通过使用一位存储器操作指令, 确保清零 $LVIMD$ 和 $LVION$ 。

图 20-5. 低电压检测电路内部复位信号产生时序
(位: LVISEL = 0, 选项字节: LVIOFF = 1)



- 注
1. 通过复位信号发生设置 LVIMK 标志为“1”。
 2. 可以设置中断请求标志寄存器的 LVIIF 标志和 LVIF 标志为 (1)。
 3. LVIRF 为复位控制标志寄存器 (RESF) 的第 0 位。需要了解 RESF 的详细信息，可参见第 18 章 复位功能。

- 备注
1. 上述图 20-5 中的<1> 至 <7>对应于 20.4.1 (1)(a) 当 LVI 默认启动功能 (LVIOFF = 1) 设置停止时中的“当启动操作”中描述的<1> 至 <7>。
 2. V_{POR} : POC 供电上升检测电压
 V_{PDR} : POC 供电下降检测电压

(b) 当 LVI 默认启动功能设置使能时 (LVIOFF = 0)

- 当启动操作时

在下列初始设置状态下开启。

- 设置 LVIM 的第 7 位 (LVION) 为 1(使能 LVI 操作)
- 把低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清零 0(检测供电电压的电平 (V_{DD}))
- 设置低电压检测电平选择寄存器 (LVIS) 为 0EH(默认数值: $V_{LVI} = 2.07V \pm 0.1V$)
- 设置 LVIM 的第 1 位 (LVIMD) 为 1(当检测到电平时产生复位)
- 设置 LVIM 的位 0(LVIF)至 0 (“供电电压(V_{DD})大于等于检测电压(V_{LVI})”)

图 20-6 显示了由低电压检测电路产生的内部复位信号的时序。

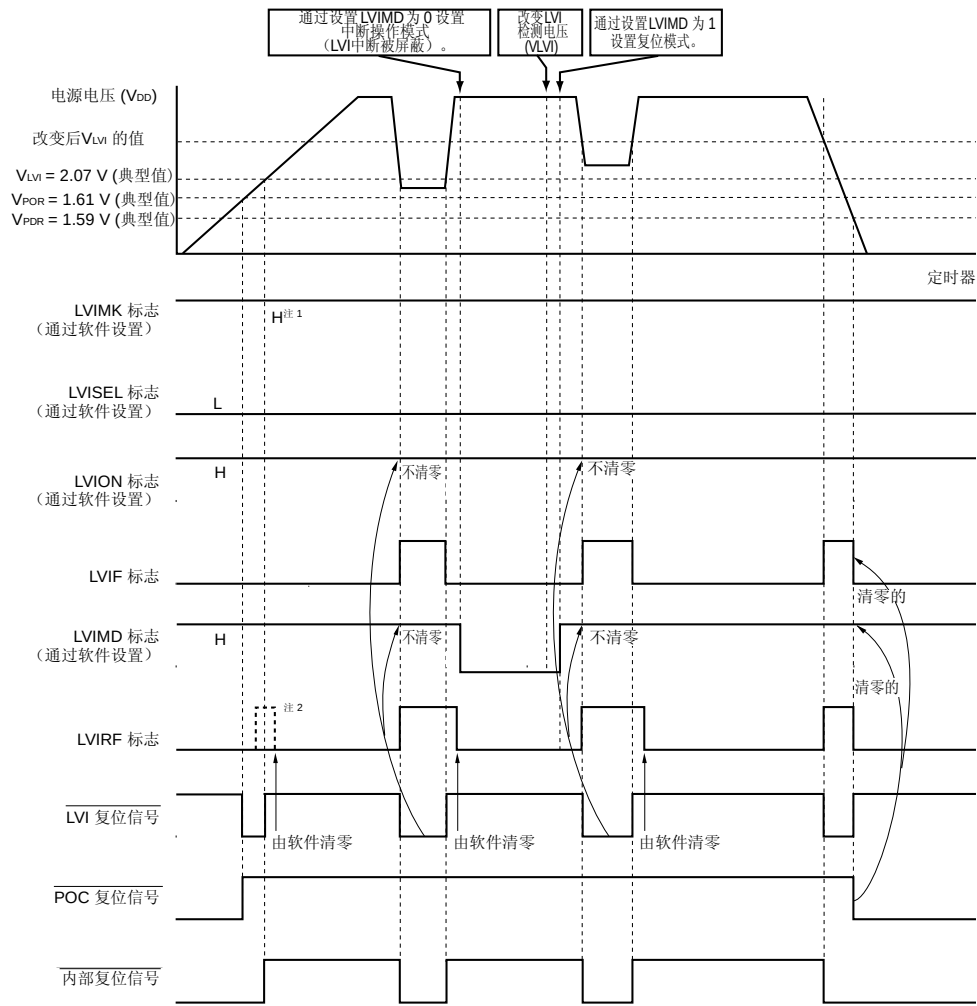
- 当停止操作时

通过使用一位存储器操作指令，确保清零 LVIMD 和 LVION。

注意事项 当 LVI 默认开始功能使用时，如果通过软件设置 LVI 操作禁止，则操作如下：

- 当 LVION = 0 时不执行低电压检测。
- 如果当 LVION = 0 时发生复位，在复位释放后 CPU 启动时 LVION 将被重置为 1。有一个阶段低电压检测不能正常执行，但由于 WDT 和非法指令执行产生一个复位。通过 LVI 进行脉冲宽度检测时最大必须不超过 200 μs 。复位发生时设置 LVION = 1，CPU 开始操作，而不需要等待 LVI 稳定时间。

图 20-6. 低电压检测电路内部复位信号产生时序
(位: LVISEL = 0, 选项字节: LVIOFF = 0)



- 注
1. 通过复位信号发生设置 LVIMK 标志为“1”。
 2. LVIRF 为复位控制标志寄存器 (RESF) 的第 0 位。
当使用 LVI 的默认开始功能 (000C1H 的第 0 位 (LVIOFF) = 0), LVIRF 标志可能会由于上电波形而从开始就变为 1。
需要了解 RESF 的详细信息, 可参见第 18 章 复位功能。

备注

V_{POR} : POC 供电上升检测电压

V_{PDR} : POC 供电下降检测电压

(2) 当检测外部输入引脚的输入电压 (EXLVI) 电平时

- 当启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1).
 - <2> 设置低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) = 1(检测外部输入引脚的输入电压 (EXLVI) 电平)。
 - <3> 设置 LVIM 的第 7 位 (LVION) = 1(使能 LVI 的操作)。
 - <4> 使用软件等待下列时间段(总计 210 μ s).
 - 运行稳定时间 (10 μ s (最大值))
 - 最小脉冲宽度 (200 μ s (最小值))
 - <5> 等到 LVIM 的第 0 位 (LVIF) 检测到(外部输入引脚的输入电压 (EXLVI) $\geq$ 检测电压 ($V_{EXLVI} = 1.21$ V (典型值))。
 - <6> 设置 LVIM 的第 1 位 (LVIMD) = 1(当检测到电平时产生复位信号)。

图 20-7 显示了由低电压检测电路产生的内部复位信号的时序。该时序图中的标号与上面的<1>至<6>对应。

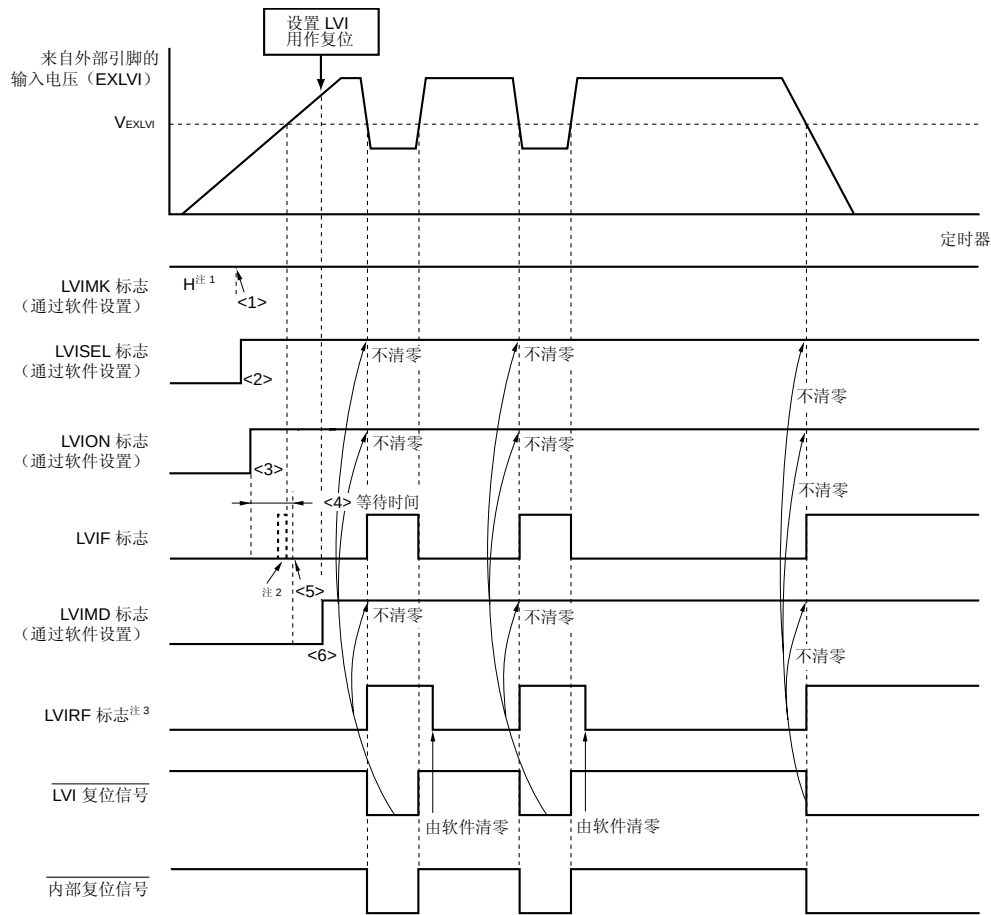
注意事项 1. 务必执行<1>。当 LVIMK = 0 时，在完成<3>后可能立即会产生中断。

2. 如果在 LVIMD=1 时，外部输入引脚的输入电压 (EXLVI) $\geq$ 检测电压 ($V_{EXLVI} = 1.21$ V (典型值))，则不会产生内部复位信号。

3. 外部输入引脚的输入电压 (EXLVI) 必须满足 $EXLVI < V_{DD}$

- 当停止操作时
通过使用一位存储器操作指令，确保清零 LVIMD 和 LVION。

图 20-7. 低电压检测电路内部复位信号产生时序
(位: LVISEL = 1)



- 注
1. 通过复位信号发生设置 LVIMK 标志为“1”。
 2. 可以设置中断请求标志寄存器的 LVIIF 标志和 LVIF 标志为 (1)。
 3. LVIRF 为复位控制标志寄存器 (RESF) 的第 0 位。需要了解 RESF 的详细信息，可参见第 18 章 复位功能。

备注 上述图 20-7 中的<1> 至 <6>对应于 20.4.1 (2)当检测外部输入引脚的输入电压 (EXLVI)电平时中的“当启动操作”中描述的<1> 至 <6>。

20.4.2 当作为中断使用时

(1) 当检测供电电压 (V_{DD}) 等级时

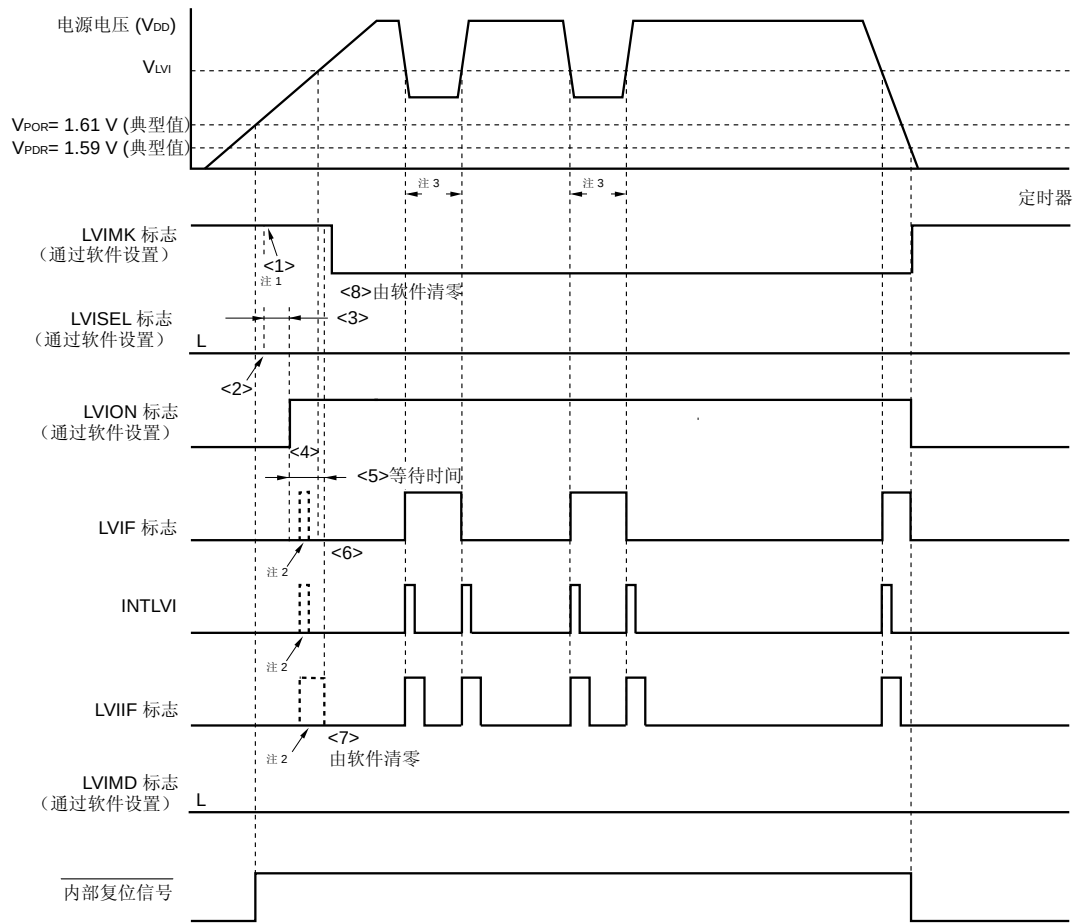
(a) 当 LVI 默认启动功能 ($LVIOFF = 1$) 设置停止时

- 当启动操作时
 - <1> 屏蔽 LVI 中断 ($LVIMK = 1$).
 - <2> 将低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清零(检测供电电压 (V_{DD}) 等级)(默认值)。
将 LVIM 的第 1 位 (LVIMD) 清零(当检测到电平时产生中断信号)(默认值)。
 - <3> 使用低电压检测电平选择寄存器 (LVIS) 的位 3 至 0 ($LVIS3$ 至 $LVIS0$) 位设置检测电压。
 - <4> 设置 LVIM 的第 7 位 ($LVION$) =1(使能 LVI 的操作)。
 - <5> 使用软件等待下列时间段(总计 $210\ \mu s$).
 - 运行稳定时间 ($10\ \mu s$ (最大值))
 - 最小脉冲宽度 ($200\ \mu s$ (最小值))
 - <6> 在 LVIM 的位 0(LVIF)处, 当检测 V_{DD} 的下降沿时, 确认“供电电压(V_{DD}) $\geq$ 检测电压(V_{Lvi})”, 或当检测 V_{DD} 的上升沿时, 确认“供电电压 (V_{DD}) $<$ 检测电压 (V_{Lvi})”。
 - <7> 将 LVI 的中断请求标志 (LVIIIF)清零。
 - <8> 释放 LVI 的中断屏蔽标志 (LVIMK) 。
 - <9> 执行 EI 指令(当使用向量中断时)。

图 20-8 显示了由低电压检测电路产生的中断信号的时序。 该时序图中的标号与上面的<1>至<8>对应。

- 当停止操作时
通过使用 1 位存储器操作指令, 确保清零 LVION。

图 20-8. 低电压检测电路中断信号产生时序
(位: LVISEL = 0, 选项字节: LVIOFF = 1)



- 注
1. 通过复位信号发生设置 LVIMK 标志为“1”。
 2. 产生中断请求信号 (INTLVI)，且可以设置 LVIF 和 LVIIF 标志为 (1)。
 3. 若禁止 LVI 操作时，当供电电压(V_{DD})小于或等于检测电压 (V_{LVI})时，将产生中断请求信号(INTLVI)并可能设置 LVIIF 为 1。

- 备注
1. 上述图 20-8 中的<1> 至 <8>对应于 20.4.2 (1)(a) 当 LVI 默认启动功能 (LVIOFF = 1) 设置停止时的“当启动操作”中描述的<1> 至 <8>。
 2. V_{POR}: POC 供电上升检测电压
V_{PDR}: POC 供电下降检测电压

(b) 当 LVI 默认启动功能设置使能时 (LVIOFF = 0)

- 当启动操作时

- <1> 在下列初始设置状态下开启。
 - 设置 LVIM 的第 7 位 (LVION) 为 1(使能 LVI 操作)
 - 把低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清零 0(检测供电电压的电平 (VDD))
 - 设置低电压检测电平选择寄存器 (LVIS) 为 0EH(默认数值: $V_{LVI} = 2.07V \pm 0.1V$)
 - 设置 LVIM 的第 1 位 (LVIMD) 为 1(当检测到电平时产生复位)
 - 设置 LVIM 的位 0(LVIF)为 0(检测“供电电压(VDD) $\geq$ 检测电压(VLVI)”状态的下降沿)
- <2> 将 LVIM 的位 1 (LVIMD) 清零(当检测到电平时产生中断信号)(默认值)。
- <3> 释放 LVI 的中断屏蔽标志 (LVIMK) 。
- <4> 执行 EI 指令(当使用向量中断时)。

图 20-9 显示了由低电压检测电路产生的中断信号的时序。该时序图中的标号与上面的<1>至<3>对应。

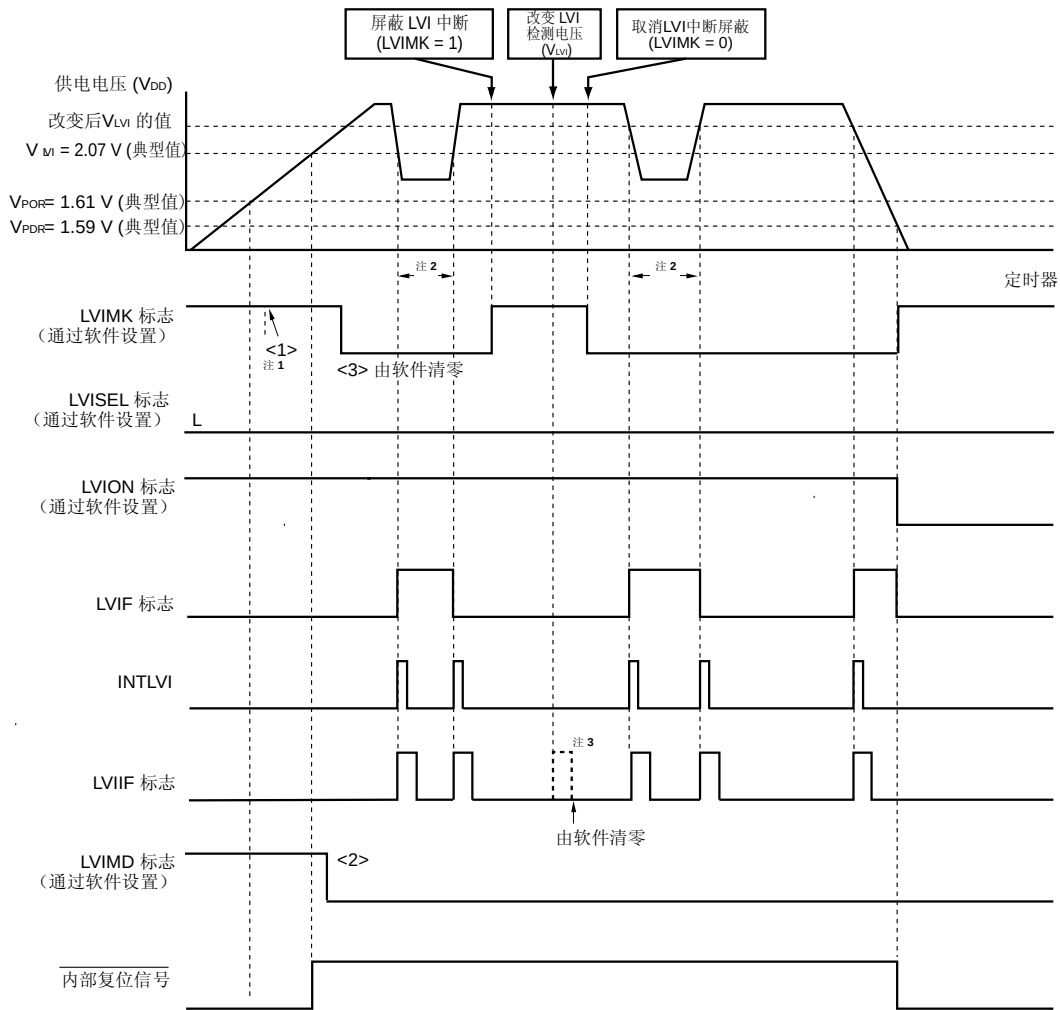
- 当停止操作时

通过使用 1 位存储器操作指令，确保清零 LVION。

注意事项 1. 当 LVI 默认开始功能使用时，如果通过软件设置 LVI 操作禁止，则操作如下：

- 当 LVION = 0 时不执行低电压检测。
 - 如果当 LVION = 0 时发生复位，在复位释放后 CPU 启动时 LVION 将被重置为 1。有一个阶段低电压检测不能正常执行，但由于 WDT 和非法指令执行产生一个复位。
通过 LVI 进行脉冲宽度检测时最大必须不超过 200 μs 。复位发生时设置 LVION = 1，CPU 开始操作，而不需要等待 LVI 稳定时间。
2. 当使用 LVI 的默认开始功能 (000C1H 的第 0 位 (LVIOFF) = 0)，LVIRF 标志可能会由于上电波形而从开始就变为 1。需要了解 RESF 的详细信息，可参见第 18 章 复位功能。

图 20-9. 低电压检测电路中断信号产生时序
(位: LVISEL = 0, 选项字节: LVIOFF = 0)



- 注
1. 通过复位信号发生设置 LVIMK 标志为“1”。
 2. 如果当供电电压 (V_{DD}) 低于或者等于检测电压 (V_{LVI}), 禁止 LVI 操作, 会产生中断请求信号 (INTLVI) 并且 LVIIF 会设置为 1。
 3. 当 LVI 检测电压改变时, LVIIF 标志可能会被设置。

- 备注
1. 上述图 20-9 中的<1> 至 <3>对应于 20.4.2 (1)(b) 当 LVI 默认启动功能 (LVIOFF = 0) 设置使能时的“当启动操作”中描述的<1> 至 <3>。
 2. V_{POR} : POC 供电上升检测电压
 V_{PDR} : POC 供电下降检测电压

(2) 当检测外部输入引脚的输入电压 (EXLVI) 电平时

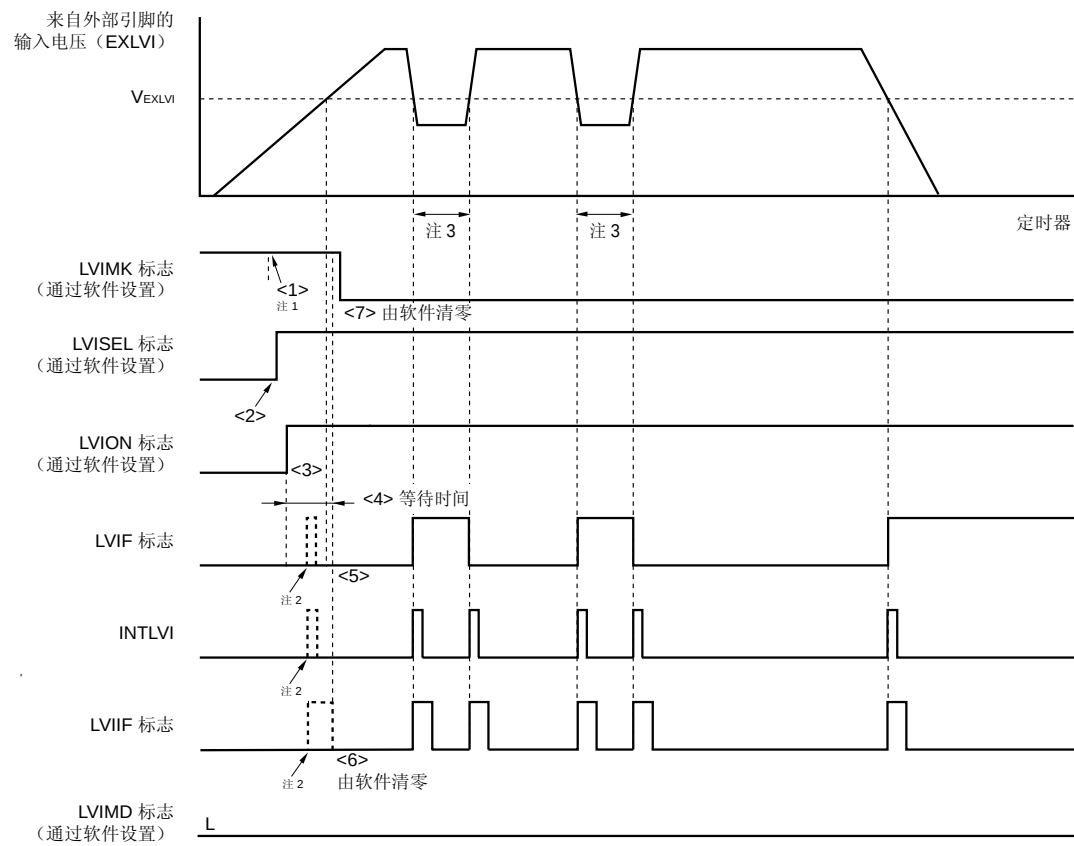
- 当启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 设置低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) = 1(检测外部输入引脚的输入电压 (EXLVI) 电平)。
将 LVIM 的第 1 位 (LVIMD) 清零(当检测到电平时产生中断信号)(默认值)。
 - <3> 设置 LVIM 的第 7 位 (LVION) = 1(使能 LVI 的操作)。
 - <4> 使用软件等待下列时间段(总计 210 μ s)。
 - 运行稳定时间 (10 μ s (最大值))
 - 最小脉冲宽度 (200 μ s (最小值))
 - <5> 在 LVIM 的位 0(LVIF)处, 当检测 EXLVI 的下降沿时, 确认“来自外部输入引脚的输入电压(EXLVI)大于等于检测电压($V_{EXLVI} = 1.21$ V (典型值))”, 或当检测 EXLVI 的上升沿时, 确认“来自外部输入引脚的输入电压(EXLVI) < 检测电压 ($V_{EXLVI} = 1.21$ V (典型值))”。
 - <6> 将 LVI 的中断请求标志 (LVIIIF)清零。
 - <7> 释放 LVI 的中断屏蔽标志 (LVIMK) 。
 - <8> 执行 EI 指令(当使用向量中断时)。

图 20-10 显示了由低电压检测电路产生的中断信号的时序。 该时序图中的标号与上面的<1>至<7>对应。

注意事项 外部输入引脚的输入电压 (EXLVI) 必须满足 $EXLVI < V_{DD}$

- 当停止操作时
通过使用 1 位存储器操作指令, 确保清零 LVION。

图 20-10. 低电压检测电路中断信号产生时序
(位: LVISEL = 1)



- 注
1. 通过复位信号发生设置 LVIMK 标志为“1”。
 2. 产生中断请求信号 (INTLVI)，且可以设置 LVIF 和 LVIIF 标志为 (1)。
 3. 若禁止 LVI 操作时，当来自外部输入引脚的输入电压(EXLVI)小于或等于检测电压(V_{EXLVI})时，将产生中断请求信号(INTLVI)并可能设置 LVIIF 为 1。

备注 当检测外部输入引脚(EXLVI)输入电压的电平时，图 20-10 中的<1> 至 <7>对应于 20.4.2 (2)中“当启动操作”描述中的<1> 至 <7>。

20.5 低电压检测电路的注意事项

(1) 当供电电压 (V_{DD}) 在接近 LVI 检测电压 (V_{LVI}) 周围而频繁波动时采取的方法

在系统中，如果供电电压 (V_{DD}) 在接近 LVI 检测电压 (V_{LVI}) 的一段时间内产生波动，这时根据低电压检测电路的使用情况进行如下操作。

操作示例 1：用作复位时

系统可能会反复进行复位和释放复位。

在这种情况下，可采用下面的方法任意设置从复位解除到微控制器的启动所经历的时间。

<方法>

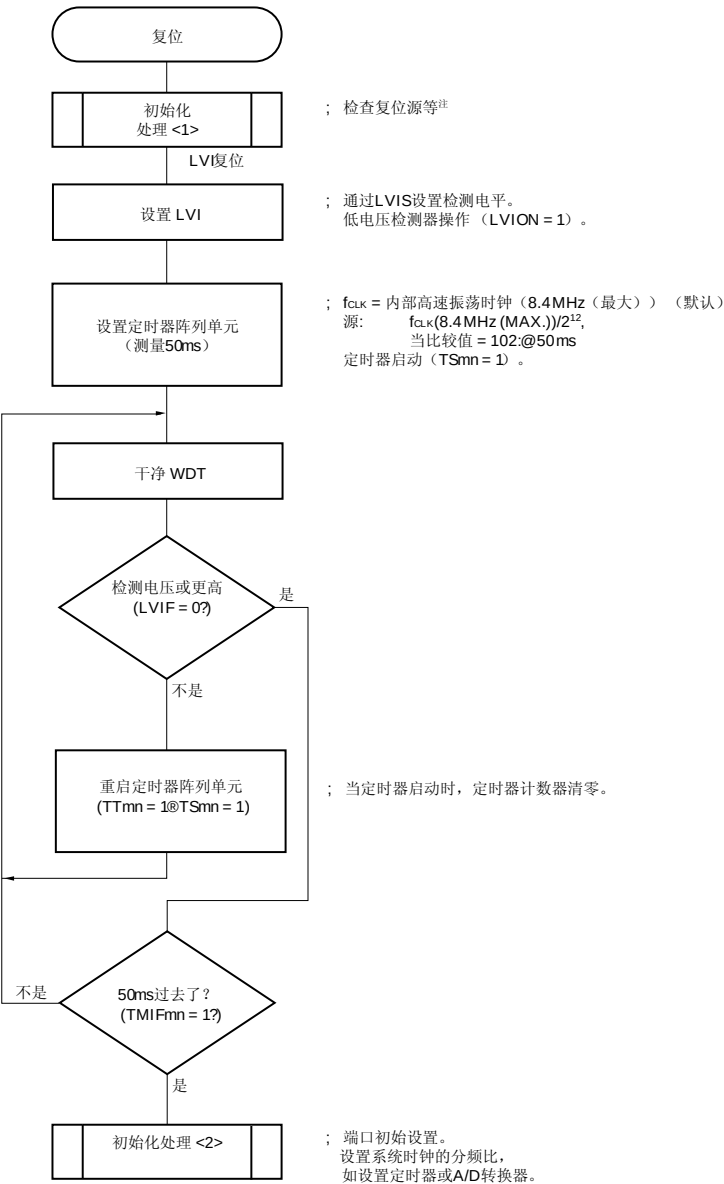
在释放复位信号后，通过软件计数器(使用一个定时器)等待系统的供电电压的波动期，然后对端口初始化(参见图 20-11)。

备注 如果低电压检测寄存器(LVIM)的位 2(LVISEL)置为“1”，上述意思改变如下。

- 电源电压 (V_{DD}) → 来自外部引脚的输入电压 ($EXLVI$)
- 检测电压 (V_{LVI}) → 检测电压 ($V_{EXLVI} = 1.21V$)

图 20-11. 复位释放后软件处理过程示例 (1/2)

- 如果供电电压在接近 LVI 检测电压时的波动时间为 50 ms 或更少

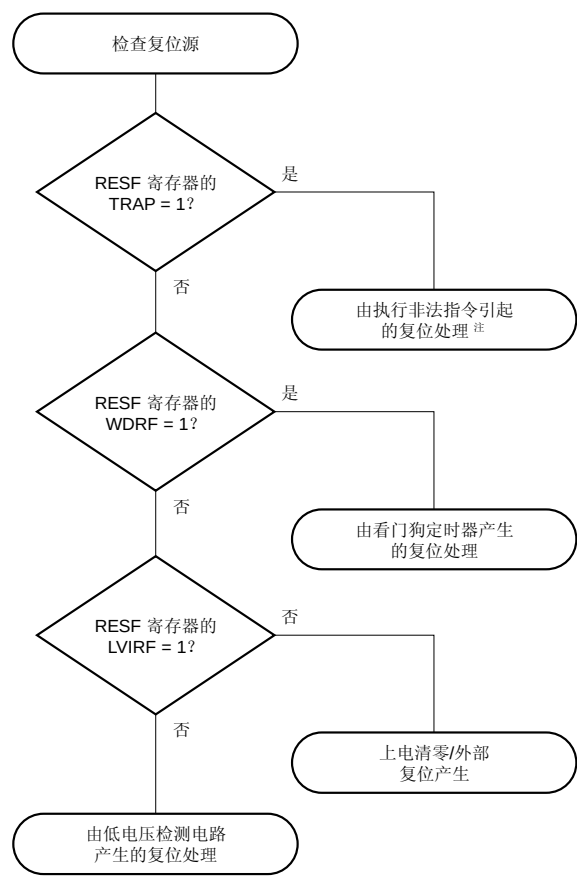


注 流程图显示在下页

- 备注
1. 如果低电压检测寄存器(LVIM)的位 2(LVISEL)置为“1”，上述意思改变如下。
 - 电源电压 (V_{DD}) → 来自外部引脚的输入电压 (EXLVI)
 - 检测电压 (V_{LVI}) → 检测电压 (V_{EXLVI} = 1.21V)
 2. m = 0, 1, n = 0 至 7, mn = 00 至 07, 10 至 13

图 20-11. 复位释放后软件处理过程示例 (2/2)

• 检查复位源



注 当指令码 FFH 执行时产生非法指令。
由执行非法指令引起的复位不能通过在线仿真器或片上调试仿真器调试终止。
调试仿真器。

备注 如果低电压检测寄存器(LVIM)的位 2(LVISEL)置为“1”，上述意思改变如下。

- 电源电压 (VDD) → 来自外部引脚的输入电压 (EXLVI)
- 检测电压 (VLVI) → 检测电压 (VEXLVI = 1.21V)

操作示例 2：用作中断时

系统可能会频繁产生中断。
可采取下列方法。

<方法>

通过使用低电压检测寄存器(LVIM)的位 0(LVIF)进行 LVI 中断例程时，当检测 V_{DD} 的下降沿时，确认供电电压 (V_{DD})大于等于检测电压(V_{LVI})，或当检测 V_{DD} 的上升沿时，确认“供电电压 (V_{DD})小于检测电压 (V_{LVI})”。将中断请求标志寄存器 0L (IF0L) 的第 1 位 (LVIF) 清零。

对于供电电压在 LVI 检测电压附近比较长时间的波动的系统，在等待供电电压波动时间之后采取以上方法。

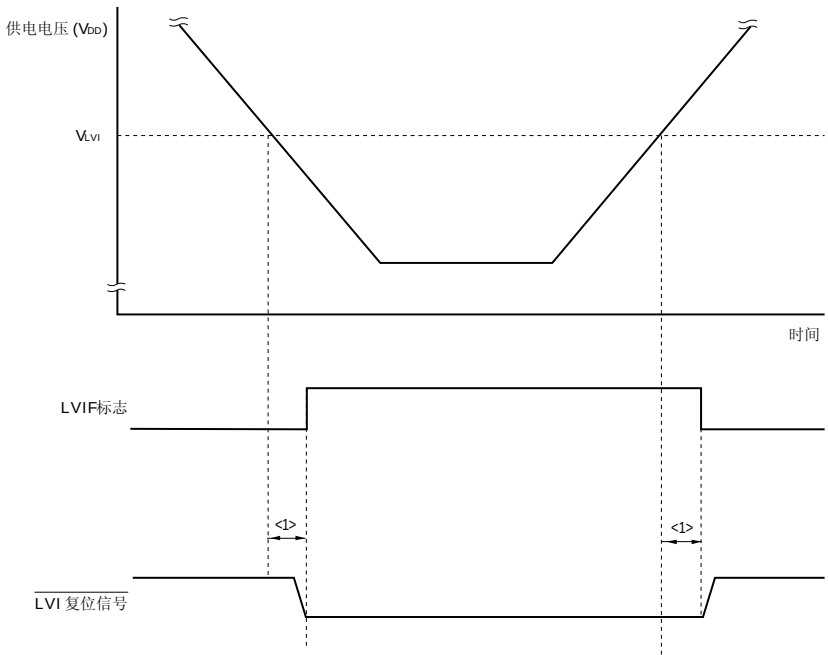
备注 如果低电压检测寄存器(LVIM)的位 2(LVISEL)置为“1”，上述意思改变如下。

- 电源电压 (V_{DD}) → 来自外部引脚的输入电压 (EXLVI)
- 检测电压 (V_{LVI}) → 检测电压 ($V_{EXLVI} = 1.21V$)

(2) 从 LVI 复位源产生时开始直到 LVI 复位产生或释放时的延时

从供电电压 (V_{DD}) < LVI 检测电压 (V_{LVI}) 时到 LVI 复位产生时，会有一些延时。
同样，也会有一些延时当 LVI 检测电压 (V_{LVI})小于等于供电电压 (V_{DD})时，直到 LVI 复位释放(见图 20-12)。

图 20-12. 从 LVI 复位源产生时开始到 LVI 复位产生或释放时的延时



<1>： 最小脉冲宽度 (200 μs (最小值))

21.1 稳压器概述

78K0R/KG3-L 包含一个在恒电压下操作设备的电路。此时，为了稳定稳压器的输出电压，把 REGC 引脚通过一个电容(0.47 至 $1\mu\text{F}$ ：目标值)连接到 V_{SS} 上。然而，当使用的 STOP 模式从内部高速振荡时钟和外部主系统时钟运行起已经进入的话，推荐使用 $0.47\mu\text{F}$ 。同时，使用性能良好的电容器，因为它用来稳定内部电压的。

稳压器的输出电压正常为 2.4V(典型值), 在低功耗电流模式下为 1.8V(典型值)。

21.2 控制稳压器的寄存器

(1) 稳压器模式控制寄存器 (RMC)

此寄存器设置稳压器的输出电压。

RMC 可用 8 位存储器操作指令设置。

复位输入设置此寄存器为 00H。

图 21-1. 稳压器模式控制寄存器 (RMC) 的格式

地址: F00F4H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
RMC								

RMC[7:0]	控制稳压器的输出电压
5AH	固定为低功耗电流模式(1.8 V)
00H	根据条件切换正常模式(2.4 V) 和低功耗电流模式(1.8 V)(参考 表 21-1)
其它值	设置禁止

注意事项 1. RMC 寄存器只有在低功耗电流模式下才可以重写(参考表 21-1)。换句话说,当高速系统时钟(f_{MX})、高速内部振荡时钟(f_{IH})和 20 MHz 内部高速振荡时钟(f_{IH20})都停止时,CPU 操作使用副系统时钟(f_{XT})期间,重写此寄存器。

2. 当设置固定为低功耗电流模式时，RMC 寄存器可在下列情况下使用。

< CPU 时钟选择 X1 时钟时>

$f_X \leq 5 \text{ MHz}$ 并且 $f_{CLK} \leq 1 \text{ MHz}$

<当 CPU 时钟选择高速内部振荡时钟、外部输入时钟或者副系统时钟时>

 $f_{CLK} \leq 1 \text{ MHz}$

3. 在低功耗电流模式下，自编程功能禁止。

(注意事项在下页列出)

注意事项 4. 在改变 RMC 寄存器之后，需要等待来改变操作速度模式控制寄存器(OSMC)当设置低功耗电流模式等待 2ms，设置普通电流模式等待 10 μ s，在下面的过程中描述。

- 当设置为低功耗电流模式
 - <1> 为 f_{CLK} 选择频率 1 MHz。
 - <2> 设置 RMC 为 5AH (设置稳压器为低功耗电流模式)。
 - <3> 等待 2 ms。
 - <4> 分别设置 OSMC 的 FLPC 和 FSEL 为 1 和 0。
- 当设置为普通电流模式时
 - <1> 设置 RMC 为 00H (设置稳压器为普通电流模式)。
 - <3> 等待 10 μ s。
 - <3> 改变 OSMC 的 FLPC 和 FSEL。
 - <4> 改变 f_{CLK} 频率。

表 21-1. 稳压器输出电压条件

模式	输出电压	条件
低功耗电流模式	1.8 V	RESET 引脚复位中
		STOP 模式下(OCD 模式除外)
		CPU 操作使用副系统时钟 (f _{XT}) 期间，高速系统时钟 (f _{MX})、高速内部振荡时钟 (f _{IH}) 和 20 MHz 内部高速振荡时钟 (f _{IH20}) 都停止时。
		HALT 模式下设置为 CPU 操作使用副系统时钟(f _{XT}) 期间，当高速系统时钟 (f _{MX})、高速内部振荡时钟 (f _{IH}) 和 20 MHz 内部高速振荡时钟 (f _{IH20}) 都停止时。
普通电流模式	2.4 V	其它值

第 22 章 选项字节

22.1 选项字节的功能

78K0R/KG3-L 的 flash 存储器的地址 000C0H 至 000C3H 组成一个选项字节区域。

选项字节由用户选项字节 (000C0H 至 000C2H) 和片上调试选项字节 (000C3H) 组成。

当打开电源或从复位状态重启设备时，设备自动参考选项字节，并设置指定的功能。使用该产品时，必须使用选项字节设置以下几项功能。

在自编程期间使用引导交换功能时，将 000C0H 至 000C3H 切换到 010C0H 至 010C3H。因此，事先将 000C0H 至 000C3H 和 010C0H 至 010C3H 设置为相同的值。

注意事项 必须将 FFH 设置为 000C2H (000C2H/ 010C2H，当使用引导交换操作时)。

22.1.1 用户选项字节 (000C0H 至 000C2H/ 010C0H 至 010C2H)

(1) 000C0H/ 010C0H

- 看门狗定时器的操作
 - 在 HALT 或 STOP 模式下操作停止或允许。
- 看门狗定时器的间隔时间的设置
- 看门狗定时器的操作
 - 操作停止或允许。
- 看门狗定时器的窗口打开期间的设置
- 看门狗定时器的间隔中断的设置
 - 使用的或未使用的

注意事项 由于在引导交换时 000C0H 要被 010C0H 替换，因此要对 010C0H 设置与 000C0H 相同的值。

(2) 000C1H/010C1H

- 复位释放时的 LVI 设置（上电时）
 - LVI 由复位释放默认开或关（仅由 $\overline{\text{RESET}}$ 引脚复位，不包括 LVI、POC、WDT、或非法指令）。
- 内部高速振荡器频率的设置
 - 从 1 MHz、8 MHz 或 20 MHz 中选择

注意事项 由于在引导交换时 000C1H 要被 010C1H 替换，因此要对 010C1H 设置与 000C1H 相同的值。

(3) 000C2H/010C2H

- 必须设置 FFH，因为这些地址为保留区域。

注意事项 当使用引导交换操作时，设置 FFH 到 010C2H，因为 000C2H 要被 010C2H 替换。

22.1.2 片上调试选项字节 (000C3H/ 010C3H)

- 片上调试操作的控制
 - 片上调试操作禁止或允许。
- 当片上调试安全 ID 验证失败时，flash 存储器数据的处理
 - 如果片上调试安全 ID 验证失败，flash 存储器的数据被擦除或不擦除。

注意事项 由于在引导交换时 000C3H 要被 010C3H 替换，因此要对 010C3H 设置与 000C3H 相同的值。

22.2 用户选项字节的格式

用户选项字节的格式如下所示。

图 22-1. 用户选项字节 (000C0H/010C0H) 的格式(1/2)

地址: 000C0H/010C0H^{註1}

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON
WDTINT	看门狗定时器的间隔中断的使用						
0	间隔中断未使用。						
1	当达到溢出时间的 75%时，间隔中断产生。						
WINDOW1	WINDOW0	看门狗定时器窗口打开周期 ^{註2}					
0	0	25%					
0	1	50%					
1	0	75%					
1	1	100%					
WDTON	看门狗定时器计数器的操作控制						
0	计数器操作禁止（复位后计数停止）						
1	计数器操作允许（复位后计数开始）						
WDCS2	WDCS1	WDCS0	看门狗定时器溢出时间 (f _{IL} = 33 kHz (最大))				
0	0	0	2 ⁷ /f _{IL} (3.88 ms)				
0	0	1	2 ⁸ /f _{IL} (7.76 ms)				
0	1	0	2 ⁹ /f _{IL} (15.52 ms)				
0	1	1	2 ¹⁰ /f _{IL} (31.03 ms)				
1	0	0	2 ¹² /f _{IL} (124.12 ms)				
1	0	1	2 ¹⁴ /f _{IL} (496.48 ms)				
1	1	0	2 ¹⁵ /f _{IL} (992.97 ms)				
1	1	1	2 ¹⁷ /f _{IL} (3971.88 ms)				

图 22-1. 用户选项字节 (000C0H/010C0H) 的格式(2/2)

地址: 000C0H/010C0H^{#1}

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDSCS2	WDSCS1	WDSCS0	WDSTBYON
WDSTBYON	看门狗定时器计数器的操作控制 (HALT/STOP 模式)						
0	计数器操作在 HALT/STOP 模式下停止 ^{#2}						
1	计数器操作在 HALT/STOP 模式下允许						

- 注
1. 由于在引导交换时 000C0H 要被 010C0H 替换, 因此要对 010C0H 设置与 000C0H 相同的值。
 2. 当 WDSTBYON = 0 时, 窗口打开周期是 100%, 不论 WINDOW1 和 WINDOW0 为何值。

注意事项 看门狗定时器在 Flash 存储器自编程和 EEPROM 仿真期间继续操作。处理期间, 中断响应时间被延迟。设置溢出时间和窗口大小时应考虑延迟。

备注 f_{IL}: 内部低速振荡时钟频率

图 22-2. 选项字节 (000C1H/010C1H) 的格式

地址: 000C1H/010C1H^{#1}

7	6	5	4	3	2	1	0
1	1	1	1	1	FRQSEL2	FRQSEL1	LVIOFF

FRQSEL2	FRQSEL1	内部高速振荡器频率
0	1	8 MHz/20 MHz ^{註 2}
1	0	1 MHz ^{註 3}
与上述不同的值		设置禁止

LVIOFF	LVI 在电源应用时的设置
0	复位释放（上电时）时，LVI 默认打开（允许 LVI 默认启动功能）
1	复位释放（上电时）时，LVI 默认关闭（停止 LVI 默认启动功能）

- 注
1. 由于在引导交换时 000C1H 要被 010C1H 替换, 因此要对 010C1H 设置与 000C1H 相同的值。
 2. 当选择 8 MHz 或 20 MHz 时, 复位释放后 8 MHz 内部高速振荡器自动启动振荡。要使用 20 MHz 内部高速振荡器来操作微控制器, 通过设置 20 MHz 内部高速振荡控制寄存器 (DSCCTL) 的位 0 (DSCON) 为 1 启动振荡并使得 $V_{DD} \geq 2.7 V$ 。电路不能改变到 1 MHz。
当微控制器操作时, 内部高速振荡器。
 3. 当选择了 1 MHz 时, 微控制器操作在 1 MHz 内部高速振荡器, 复位释放后 电路不能改变至 8 MHz 或 20 MHz 内部高速振荡器, 当微控制器操作时。

(注意事项在下页中列出。)

- 注意事项
- 1. 确保设置位 7 至 3 为“1”。
 - 2. 当 LVI 默认开始功能使用时，如果通过软件设置 LVI 操作禁止，则操作如下：
 - 当 LVION = 0 时不执行低电压检测。
 - 如果当 LVION = 0 时发生复位，在复位释放后 CPU 启动时 LVION 将被重置为 1。有一个阶段低电压检测不能正常执行，但由于 WDT 和非法指令执行产生一个复位。
通过 LVI 进行脉冲宽度检测时最大必须不超过 200 μ s。复位发生时设置 LVION = 1，CPU 开始操作，而不需要等待 LVI 稳定时间。

图 22-3. 选项字节 (000C2H/010C2H) 的格式

地址: 000C2H/010C2H[#]

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1

注 必须将 FFH 设置为 000C2H，因为这些地址在保留区域。当使用引导交换操作时也要设置 FFH 为 010C2H，因为 000C2H 要被 010C2H 替换。

22.3 片上调试选项字节的格式

片上调试选项字节的格式如下所示。

图 22-4. 片上调试选项字节 (000C3H/010C3H) 的格式

地址: 000C3H/010C3H[#]

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	片上调试操作的控制
0	0	禁止片上调试操作。
0	1	设置禁止
1	0	使能片上调试操作。 当片上调试安全 ID 验证失败时，擦除 flash 存储器数据
1	1	使能片上调试操作。 当片上调试安全 ID 验证失败时，不擦除 flash 存储器数据

注 由于在引导交换时 000C3H 要被 010C3H 替换，因此要对 010C3H 设置与 000C3H 相同的值。

注意事项 第 7 位和第 0 位（OCDENSET 和 OCDERSD）只能被指定一个值。
000010B 的第 6 位必须设置为 1。

备注 当使用片上调试功能时，第 3 位至第 1 位的值被写入，设置后其值变得不确定。
但是，设置时必须将第 3 位至第 1 位设置为默认值（0、1、和 0）。

22.4 选项字节的设置

除了在源程序中描述之外，用户选项字节和片上调试选项字节可通过 RA78K0R 或 PM+链接器选项设置。当这样做时，通过链接器选项设置的内容优先，即使源中有如下文提到的描述存在。

关于如何设置链接器选项，参见 RA78K0R 汇编程序包用户手册。

选项字节设置的软件描述示例如下所示。

OPT	CSEG	OPT_BYTE	
	DB	10H	； 不要使用看门狗定时器的间隔中断， ； 使能看门狗定时器操作， ； 看门狗定时器的窗口打开周期是 25%， ； 看门狗定时器的溢出时间是 $2^{10}/f_{IL}$ ， ； HALT/STOP 模式期间，停止看门狗定时器的操作
	DB	0FBH	； 为内部高速振荡器选择 8 MHz 或 20 MHz ； 停止 LVI 默认启动功能
	DB	0FFH	； 预约区域
	DB	85H	； 允许片上调试操作，不要擦除 Flash 存储器 ； 当安全 ID 认证失败时的数据

在自编程期间，使用引导交换功能时，将 000C0H 至 000C3H 切换到 010C0H 至 010C3H。因此，描述至 010C0H 至 010C3H，设置为如下与 000C0H 至 000C3H 相同的值。

OPT2	CSEG	AT	010C0H	
	DB		10H	； 不要使用看门狗定时器的间隔中断， ； 允许看门狗定时器操作， ； 看门狗定时器的窗口打开周期是 25%， ； 看门狗定时器的溢出时间是 $2^{10}/f_{IL}$ ， ； HALT/STOP 模式期间，停止看门狗定时器的操作
	DB		0FBH	； 为内部高速振荡器选择 8 MHz 或 20 MHz ； 停止 LVI 默认启动功能
	DB		0FFH	； 预约区域
	DB		85H	； 允许片上调试操作，不要擦除 Flash 存储器 ； 当安全 ID 认证失败时的数据

注意 要通过汇编语言指定选项字节，应使用 OPT_BYTE 作为 CSEG 伪指令的重定位属性名。要为使用引导交换功能而在 010C0H 至 010C3H 设置选项字节，应使用重定位属性 AT 来指定一个绝对地址。

第 23 章 FLASH 存储器

78K0R/KG3-L 上集成了 Flash 存储器，可以安装在电路板上实现写入、擦除和重写操作。

23.1 用 Flash 存储器编程器写入

通过使用专用 Flash 存储器编程器，数据可以 on-board 或 off-board 写入到 Flash 存储器。

(1) On-board 编程

当目标系统上安装了 78K0R/KG3-L 系列产品后，Flash 存储器的内容可以重写。必须将连接专用 Flash 编程器的连接器安装在目标系统中。

(2) Off-board 编程

在 78K0R/KG3-L 安装到目标系统之前，可使用专用程序适配器（FA 系列）将数据写入 Flash 存储器。

备注 FA 系列是 Naito Densei Machida Mfg. Co., Ltd 公司的产品。

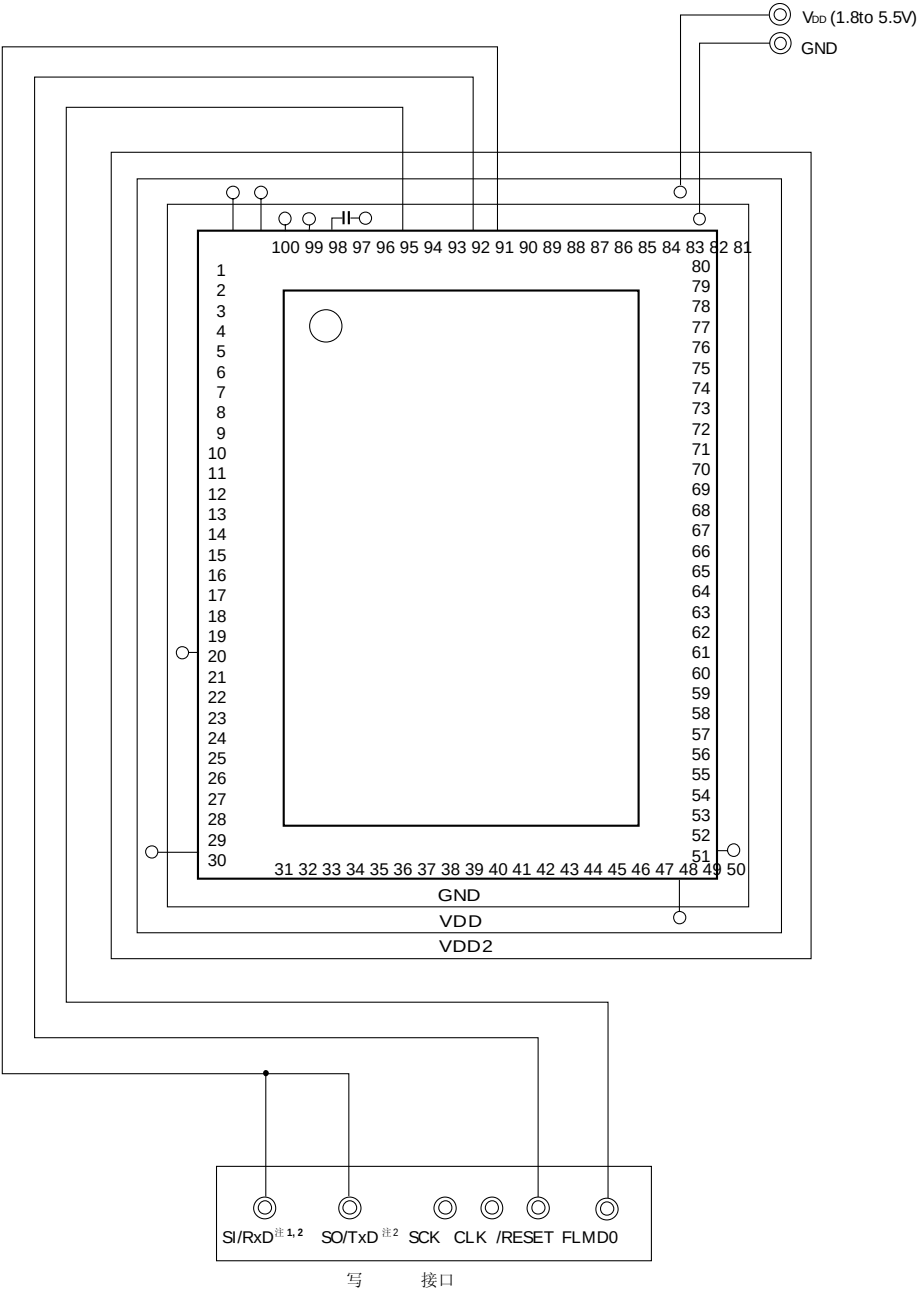
表 23-1. 78K0R/KG3-L 和专用 Flash 存储器编程器之间的写操作

专用 Flash 存储器编程器的引脚配置			GF 包		GC 包	
信号名称	I/O	引脚功能	引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD ^{注 1, 2}	输入	接收信号	TOOL0/P40	89	TOOL0/P40	12
SO/TxD ^{注 2}	输出	发送信号				
SCK	输出	传输时钟	—	—	—	—
CLK	输出	时钟 输出	—	—	—	—
/RESET	输出	复位信号	$\overline{\text{RESET}}$	90	$\overline{\text{RESET}}$	13
FLMD0	输出	模式信号	FLMD0	93	FLMD0	16
V _{DD}	I/O	V _{DD} 电压产生/ 电压监测	V _{DD}	99	V _{DD}	22
			EV _{DD0}	100	EV _{DD0}	23
			EV _{DD1}	30	EV _{DD1}	53
			AV _{REF}	50	AV _{REF}	73
GND	—	地	V _{SS}	97	V _{SS}	20
			EV _{SS0}	98	EV _{SS0}	21
			EV _{SS1}	20	EV _{SS1}	43
			AV _{SS}	51	AV _{SS}	74

- 注
1. 当使用 PG-FP5 或 FL-PR5 时无需连接此引脚。
 2. 当使用 QB-MINI2 时，连接 SI/RxD 或 SO/TxD。

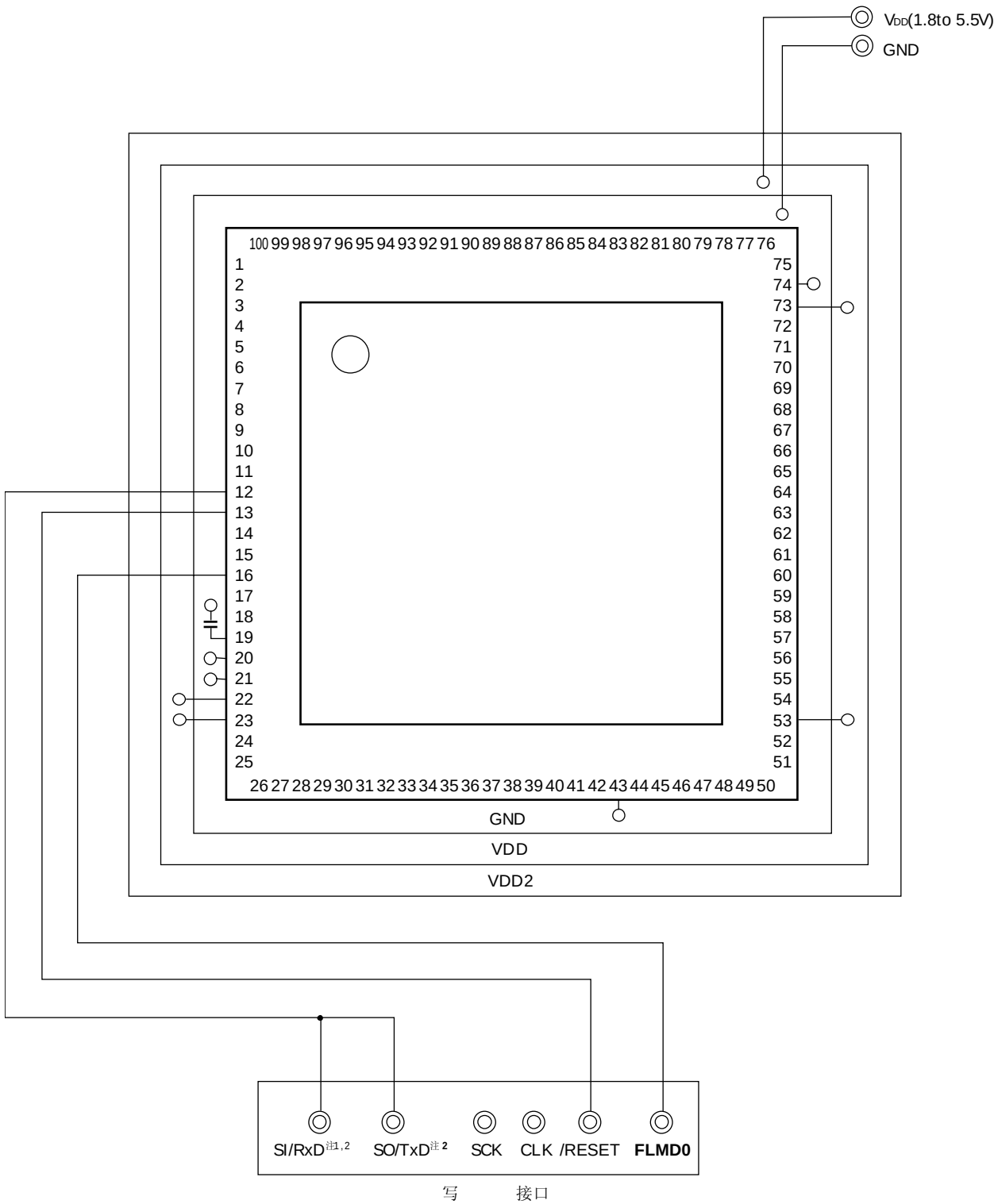
在使用适配器对 Flash 存储器进行写操作时，可采用如下所示的连接方式。

图 23-1. 使用适配器对 Flash 存储器进行写操作示例(GF 封装)



- 注
1. 当使用 PG-FP5 或 FL-PR5 时无需连接此引脚。
 2. 当使用 QB-MINI2 时，连接 SI/RxD 或 SO/TxD。

图 23-2. 使用适配器对 Flash 存储器进行写操作示例(GC 封装)

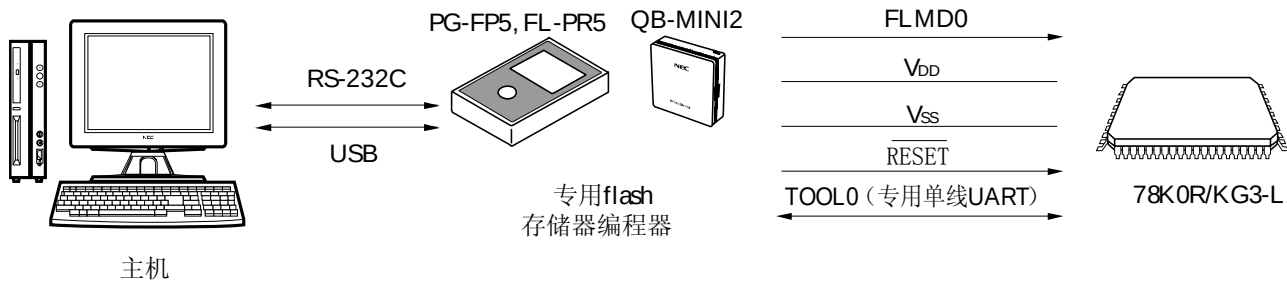


注 1. 当使用 PG-FP5 或 FL-PR5 时无需连接此引脚。
2. 当使用 QB-MINI2 时，连接 SI/RxD 或 SO/TxD。

23.2 编程环境

以下是向 78K0R/KG3-L 的 Flash 存储器写入程序所需的环境。

图 23-3. 向 Flash 存储器写入的环境



需要有一个控制专用 Flash 编程器的主机。

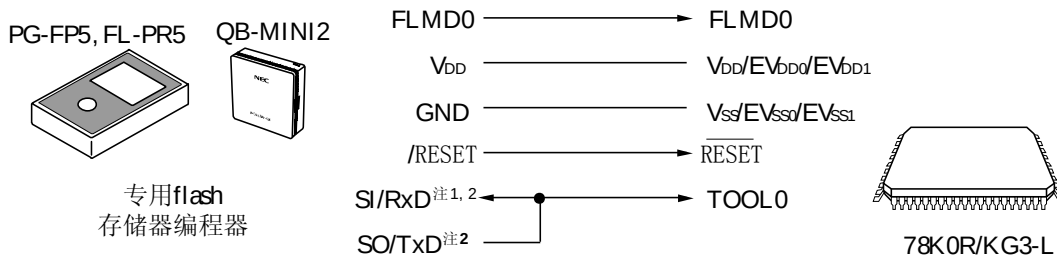
专用 Flash 存储器编程器与 78K0R/KG3-L 的接口 TOOL0 引脚通过专用单线 UART 进行写入和擦除等操作。Flash 存储器 off-board 写操作时必须使用一个专用编程适配器 (FA 系列)。

23.3 通信模式

专用 Flash 存储器编程器和 78K0R/KG3-L 之间的通信由串行通信连接，使用 TOOL0 引脚通过 78K0R/KG3-L 的专用单线 UART 进行通讯。

传输速率：115,200 bps 至 1,000,000 bps

图 23-4. 专用 Flash 存储器编程器的通信



- 注
1. 当使用 PG-FP5 或 FL-PR5 时无需连接此引脚。
 2. 当使用 QB-MINI2 时，连接 SI/RxD 或 SO/TxD。

专用 flash 存储器编程器产生 78K0R/KG3-L 以下信号，详情参见 PG-FP5, FL-PR5 或 MINICUBE2 的手册

表 23-2. 引脚连接

专用 Flash 存储器编程器			78K0R/KG3-L	连接
信号名称	I/O	引脚功能	引脚名称	
FLMD0	输出	模式信号	FLMD0	⊙
V _{DD}	I/O	V _{DD} 电压产生/电源监测	V _{DD} , EV _{DD0} , EV _{DD1} , AV _{REF}	⊙
GND	—	地	V _{SS} , EV _{SS0} , EV _{SS1} , AV _{SS}	⊙
CLK	输出	时钟 输出	—	×
/RESET	输出	复位信号	$\overline{\text{RESET}}$	⊙
SI/RxD ^{注 1, 2}	输入	接收信号	TOOL0	⊙
SO/TxD ^{注 2}	输出	发送信号		
SCK	输出	传输时钟	—	×

注 1. 当使用 PG-FP5 或 FL-PR5 时无需连接此引脚。
 2. 当使用 QB-MINI2 时，连接 SI/RxD 或 SO/TxD。

备注 ⊙: 确保连接引脚。
 ×: 引脚不必连接。

23.4 On Board 方式的引脚连接

对 Flash 存储器进行 on-board 写入操作时，目标系统必须有连接专用 Flash 编程器的连接器。电路板上首先要提供一个选择功能，可以选择正常操作模式或 Flash 存储器编程模式。

当设置 Flash 存储器编程模式时，那些不用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此如果外部设备不能立即识别复位后的状态，则必须采用如下方式连接引脚。

23.4.1 FLMD0 引脚

(1) 在 Flash 存储器编程模式下

当通过 Flash 存储器编程器写入数据时，将此引脚直接连接到 Flash 存储器编程器。这提供了一个 V_{DD} 电平的写入电压到 FLMD0 引脚。
FLMD0 引脚不必由外部进行下拉，因为复位时会被内部下拉。要外部下拉，使用 1kΩ到 200kΩ的电阻。

(2) 在普通操作模式下

普通操作期间，建议此引脚开路。
复位释放前 FLMD0 引脚必须保持 V_{SS} 电位，但是不必由外部进行下拉，因为复位时会被内部下拉。但是，必须保持选择下拉（如 FLMDPUP = “0”，缺省值），通过使用背景事件控制寄存器(BECTL)中的位 7(FLMDPUP)（见 23.5 (1)背景事件控制寄存器）。要处于外部下拉状态，需要使用 200kΩ或更小值的电阻。
自编程功能和使用编程器对 Flash 存储器进行重写的功能可以通过硬件来禁止，将该引脚直接连接到 V_{SS} 引脚即可。

(3) 在自编程模式下

使用自编程功能时，此引脚推荐开路。要外部下拉，使用 100kΩ 到 200kΩ 的电阻。

在自编程模式下，设置在自编程库切换到上拉状态。

图 23-5. FLMD0 引脚连接示例

**23.4.2 TOOL0 引脚**

在 Flash 存储器编程模式下，该引脚直接连接到专用 Flash 存储器编程器，或通过外部电阻连接到 EV_{DD0} 或 EV_{DD1} 进行上拉。

当在普通操作模式下允许片上调试功能时，经外部电阻连接到 EV_{DD0} 或 EV_{DD1} 对该引脚进行上拉，必须在复位释放前确保输入 V_{DD} 到 TOOL0 引脚（禁止上拉此引脚）。

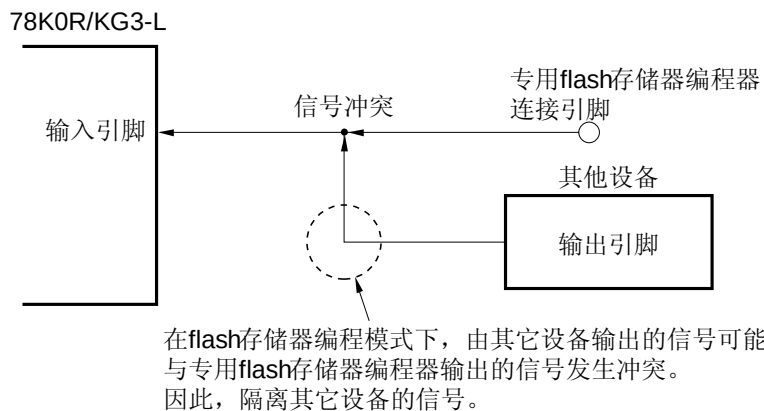
备注 因为使用了单线 UART，SAU 和 IICA 引脚不能用于 78K0R/KG3-L 和专用 Flash 存储器编程器之间的通信。

23.4.3 RESET 引脚

如果将专用 Flash 编程器的复位信号连接到 $\overline{\text{RESET}}$ 引脚（已连接到板上的复位信号发生器），则会产生信号冲突。为了避免这种情况，应隔离与复位信号发生器的连接。

在 Flash 存储器编程模式下，如果从用户系统输入复位信号，则不能对 Flash 存储器进行正确编程。因此除了专用 Flash 编程器的复位信号外，不要输入其它信号。

图 23-6. 信号冲突($\overline{\text{RESET}}$ 引脚)



23.4.4 端口引脚

当设置 Flash 存储器编程模式时，那些不用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此，如果外部设备（与端口相连）不能立即识别复位后的状态，则必须通过一个电阻将端口引脚连接到 VDD 或 VSS。

23.4.5 REGC 引脚

与普通操作期间相同，通过一个电容（0.47 至 1 μ F：推荐值）连接 REGC 引脚到 GND。然而，当使用的 STOP 模式从内部高速振荡时钟和外部主系统时钟的操作已经进入的话，推荐使用 0.47 μ F。同时，使用性能良好的电容器，因为它是用来稳定内部电压的。

23.4.6 X1 和 X2 引脚

以与普通操作模式相同的状态连接 X1 和 X2。

备注 在 Flash 存储器编程模式下，使用内部高速振荡时钟 (f_{IH})。

23.4.7 供电电压

要使用 Flash 编程器输出的供电电压，则将 VDD 引脚与 Flash 编程器的 VDD 相连，将 VSS 引脚与 Flash 编程器的 GND 相连。

要使用 on-board 供电电压，要按正常操作模式连接。

然而，当 on-board 供电电压使用时，如果要使用 Flash 编程器的电压监视功能，必须将 VDD 和 VSS 引脚分别与 Flash 编程器的 VDD 和 GND 相连。

其他供电电压 (EVDD0, EVDD1, EVSS0, EVSS1, AVREF 和 AVSS) 与正常操作模式中的相同。

23.5 控制 Flash 存储器的寄存器

(1) 背景事件控制寄存器(BECTL)

即使 FLMD0 引脚不由外部控制，可通过 BECTL 寄存器设置为自编程模式从而由软件控制。

但是，根据 FLMD0 引脚的处理，可能不能由软件设置到自编程模式。使用 BECTL 时，建议 FLMD0 引脚悬空。当通过外部将其下拉时，使用阻值大于 100 k Ω 的电阻。另外，在普通操作模式下，使用 BECTL 和下拉选项。在自编程模式下，在自编程库中设置被切换到上拉。

BECTL 寄存器可由 1 位或 8 位存储器操作指令设置。

复位输入设置此寄存器为 00H。

图 23-7. 背景事件控制寄存器 (BECTL) 的格式

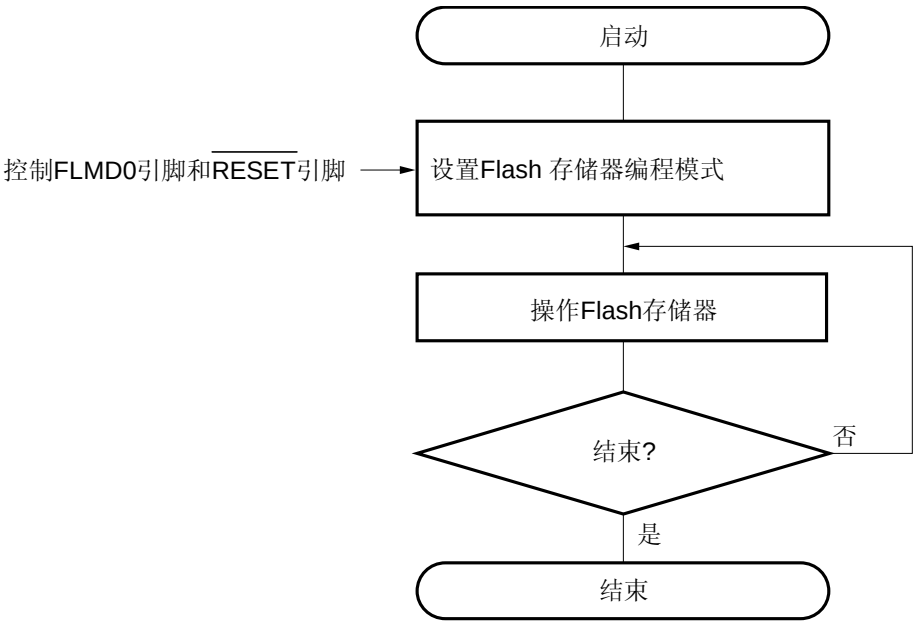
地址: FFFBEH	复位后: 00H	R/W						
符号	<7>	6	5	4	3	2	1	0
BECTL	FLMDPUP	0	0	0	0	0	0	0
FLMDPUP		FLMD0引脚的软件控制						
0		选择下拉						
1		选择上拉						

23.6 编程方法

23.6.1 控制 Flash 存储器

下图显示了操作 Flash 存储器的过程。

图 23-8. Flash 存储器操作过程



23.6.2 Flash 存储器编程模式

要使用专用 Flash 存储器编程器重写 flash 存储器的内容，需在 Flash 存储器编程模式下设置 78K0R/KG3-L。而要设置该模式，必须将 FLMD0 引脚和 TOOL0 引脚连接到 V_{DD}，并对复位信号清零。

当 on-board 写 Flash 存储器时，使用跳线改变模式。

图 23-9. Flash 存储器编程模式

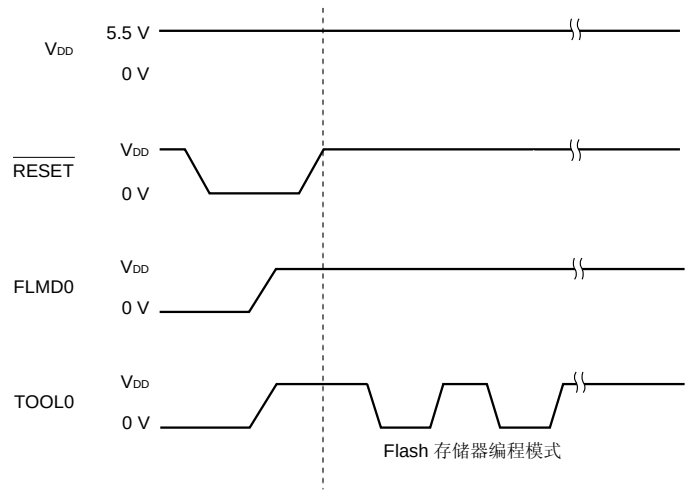


表 23-3. 复位释放后 FLMD0 引脚与操作模式之间的关系

FLMD0	操作模式
0 V	普通操作模式
V _{DD}	Flash 存储器编程模式

23.6.3 选择通信模式

78K0R/KE3-L 的通信模式如下。

表 23-4. 通信模式

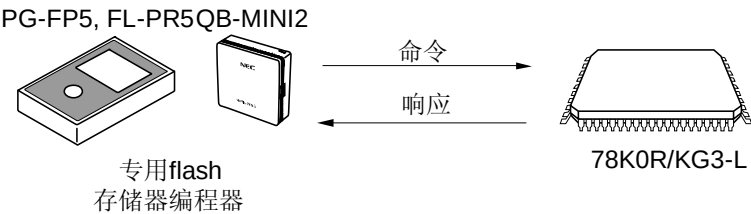
通信模式	标准设置 ^{注 1}				使用的引脚
	端口	速度	频率	倍率	
单线模式 (单线 UART)	UART	115,200 bps, 250,000 bps, 500,000 bps, 1 Mbps ^{注 2}	—	—	TOOL0

- 注
- 1. Flash 存储器编程器的 GUI 上的标准设置选项。
 - 2. 波特率误差以外的因素，如信号波形瞬变，也会影响 UART 通信，因此必须像测量波特率误差一样严格测量瞬变。

23.6.4 通信命令

78K0R/KE3-L 使用命令与专用 Flash 存储器编程器进行通信。从 Flash 存储器编程器发往 78K0R/KE3-L 的信号称为命令，从 78K0R/KE3-L 发往专用 Flash 存储器编程器的信号称为响应。

图 23-10. 通信命令



78K0R/KE3-L 的 Flash 存储器控制命令列表如下。所有这些命令都由编程器发出，78K0R/KE3-L 根据接收到的命令进行相应的处理。

表 23-5. Flash 存储器控制命令

分类	命令名称	功能
校验	校验	将 Flash 存储器指定区域的内容和从编程器发送的数据比较。
擦除	芯片擦除	擦除全部 Flash 存储器内容。
	块擦除	擦除 Flash 存储器的指定区域。
空白检测	块空白检测	检查 Flash 存储器的指定块擦除状态。
写入	编程	在 Flash 存储器的指定区域写入数据。
获取信息	硅标记	获取 78K0R/KG3-L 信息（比如部件编号和 flash 存储器配置）。
	版本获得	获取 78K0R/KG3-L 固件版本。
	校验和	获得指定区域的校验和数据。
安全	安全设置	设置安全信息。
其它	复位	用于检查通信的同步状态。
	波特率设置	当选择 UART 通信模式时设置波特率。

78K0R/KG3-L 根据专用 Flash 存储器编程器发出的命令返回一个响应。78K0R/KG3-L 发出的响应列表如下。

表 23-6. 响应名称

响应名称	功能
ACK	响应命令/数据
NAK	响应非法命令/数据

23.7 安全设置

78K0R/KG3-L 支持安全性功能，该功能禁止重写 Flash 存储器内的用户程序，未授权者无法改变程序。

使用安全性设置命令可以执行如下操作。在设置下一个编程模式时，安全性设置有效。

- 禁止一次性擦除（片擦除）

在 on-board/off-board 编程期间，该项设置禁止对 Flash 存储器的所有块执行块擦除和一次性擦除(片擦除)命令。

一旦禁止一次性擦除（片擦除）命令，所有的禁止设置项（包括禁止一次性擦除（片擦除））不能被取消。

注意事项 在对一次性擦除进行安全性设置后，不能再对该设置执行擦除操作。此外，由于禁止执行擦除命令，即使执行了写命令，与已经写到 Flash 存储器中的数据不同的数据也不能被写入。

- 禁止块擦除

在 on-board/off-board 编程期间，该项设置禁止对 Flash 存储器中的指定块进行擦除。但可通过自编程进行块擦除。

- 禁止写入

在 on-board/off-board 编程期间，该项设置禁止对 Flash 存储器所有块进行写入和块擦除。但可通过自编程进行块写入。

- 禁止重写引导簇 0

这项设置禁止在 Flash 存储器的引导簇 0（00000H 至 00FFFH）执行块擦除和写命令。并且执行一次性擦除（片擦除）命令。

Flash 存储器在默认设置下，允许使用一次性擦除（片擦除）、块擦除、写和重写引导簇 0 命令。以上安全性设置仅用于在 on-board/off-board 编程和自编程。各项安全性设置可以结合使用。

通过执行批量擦除（片擦除）命令，可以取消禁止擦除块和写命令。

表 23-7 表示了当 78K0R/KG3-L 允许使用安全功能时，擦除命令和写命令之间的关系。

备注 在自编程期间禁止写入和擦除，使用 Flash 密封窗口功能（详见 23.8.2）。

表 23-7. 允许安全性功能和命令之间的关系

(1) On-board/off-board 编程期间

有效的安全性设置	执行命令		
	批量擦除（片擦除）	块擦除	写入
禁止批量擦除（片擦除）	不能批量擦除	块不能被擦除。	有效 ^注
禁止块擦除	可以批量擦除		有效
禁止写			无效
禁止重写引导簇 0	不能批量擦除	引导簇 0 不能被擦除。	引导簇 0 不能被写入。

注 确认没有数据已写在被写的区域。因为批量擦除（片擦除）禁止后，数据不能被擦除，所以在还有数据没被擦除的情况下不要写数据。

(2) 自编程期间

有效的安全性设置	执行命令	
	块擦除	写入
禁止批量擦除（片擦除）	块可被擦除。	有效
禁止块擦除		
禁止写		
禁止重写引导簇 0	引导簇 0 不能被擦除。	引导簇 0 不能被写入。

备注 在自编程期间禁止写入和擦除，使用 Flash 密封窗口功能（详见 23.8.2）。

表 23-8. 在每个编程模式下设置安全性

(1) On-board/off-board 编程

安全	安全设置	如何禁止安全设置
禁止批量擦除（片擦除）	通过专用 Flash 存储器编程器的 GUI 设置，等。	复位后不能禁止。
禁止块擦除		执行批量擦除（片擦除）命令。
禁止写		
禁止重写引导簇 0		复位后不能禁止。

(2) 自编程

安全	安全设置	如何禁止安全设置
禁止批量擦除（片擦除）	通过信息库设置。	复位后不能禁止。
禁止块擦除		在 on-board/off-board 编程期间（在自编程期间不能禁止）执行批量擦除（片擦除）命令。
禁止写		
禁止重写引导簇 0		

23.8 通过自编程进行 Flash 存储器编程

78K0R/KG3-L 支持自编程功能，即可以通过用户程序重写 Flash 存储器。由于该功能允许用户应用程序利用 78K0R/KG3-L 自编程库重写 flash 存储器，因此可用于对程序升级。

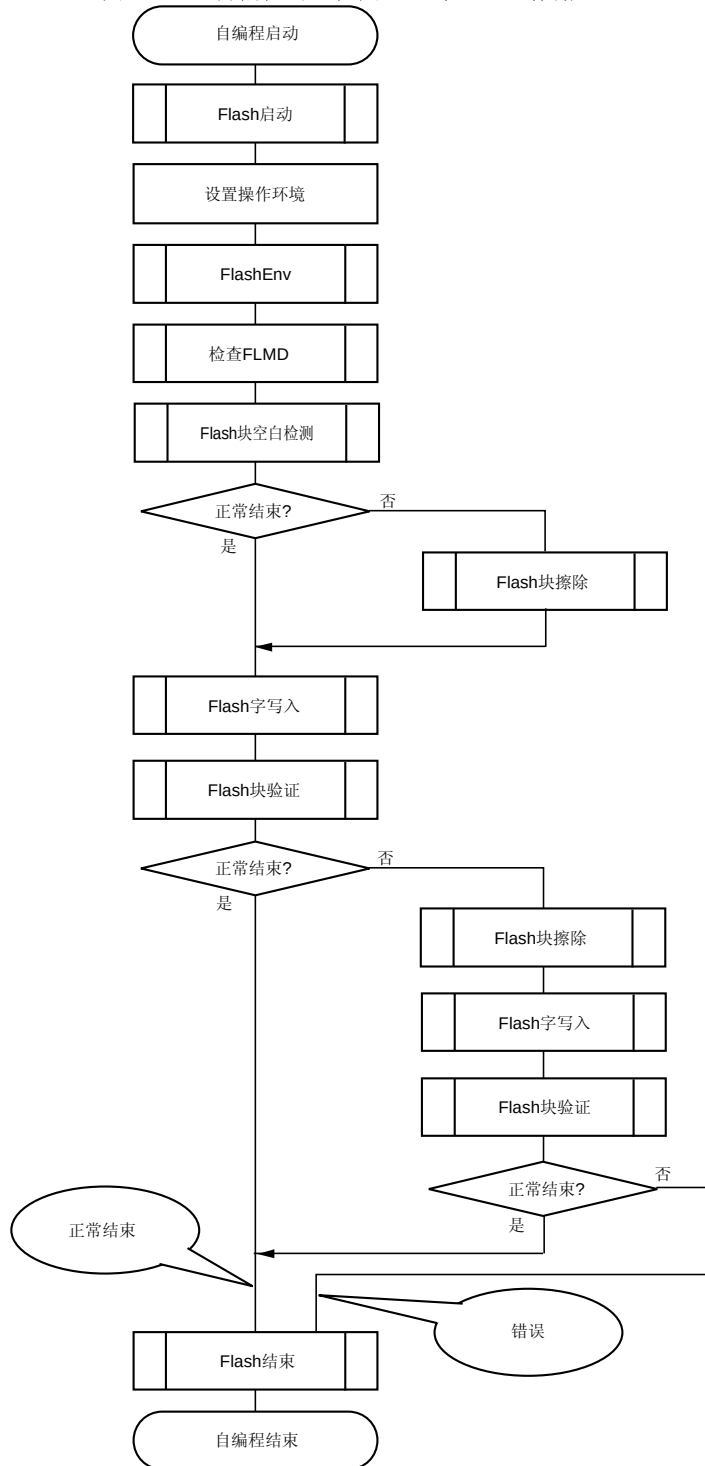
如果自编程期间产生中断，可以暂时停止自编程操作，而执行中断服务程序。如果在 EI 状态一个未屏蔽中断请求产生，则请求分支直接从自编程库转向中断程序。而在其后恢复自编程模式后，自编程可以继续进行。但是，中断响应时间与在普通模式下不同。

备注 关于自编程功能和 78K0R/KG3-L 自编程库的详细信息，参考 **78K0R 微控制器自编程库类型 02 的用户手册 (U19193E)**。

- 注意事项**
1. 当 CPU 使用副系统时钟时，不能使用自编程功能。
 2. 在自编程模式下，调用自编程开始库 (FlashStart)。
 3. 在自编程期间要禁止一个中断，与在普通操作模式下方法相同，在通过 DI 指令执行将 IE 标志清零的状态下，执行自编程库。要允许一个中断，将中断屏蔽标志清零，通过 EI 指令设置 IE 标志为 (1)，然后执行自编程库。
 4. 在低功耗电流模式下，自编程功能禁止。低功耗电流模式的详细内容，参见第 21 章 稳压器。
 5. 在执行自编程库功能时禁止 DMA 操作 (DENn = 0)。

下图所示为通过使用自编程库重写 Flash 存储器的流程图。

图 23-11. 自编程的流程图（重写 Flash 存储器）



备注 关于自编程库的详细信息，参考 **78K0R 微控制器自编程库类型 02 的用户手册 (U19193E)**。

23.8.1 引导交换功能

如果在自编程期间由于电源失效或其它原因而导致引导区域的重写失败，则由于引导区域的数据损失，不能通过复位或覆盖重启程序。

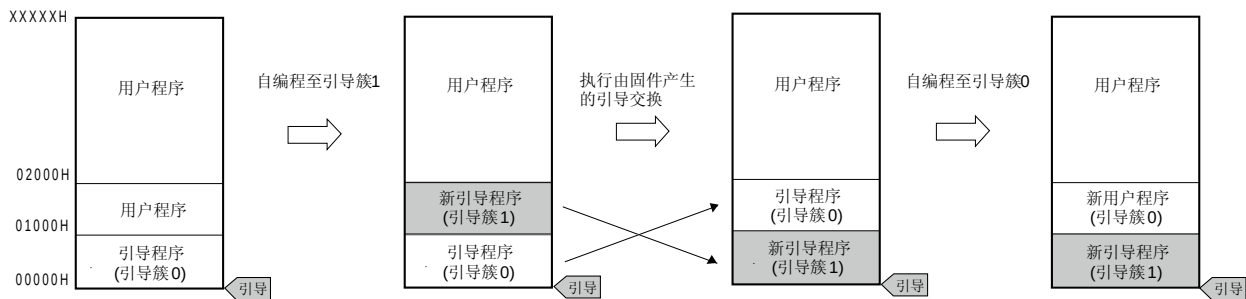
使用引导交换功能可以避免这种情况。

在擦除引导簇 0th（这是一个引导程序区域）之前，先通过自编程将一个新的引导程序写入引导簇 1。当将该程序正确写入引导簇 1 后，通过使用 78K0R/KG3-L 固件的设置信息功能将引导簇 1 与引导簇 0 的内容交换，这样引导簇 1 就被用作引导区域。之后对源引导程序区域，即引导簇 0，进行擦除或写操作。

这样，即使在重写引导程序区域时出现电源失效，也能够正确执行程序，这是因为程序在复位并再次执行时实行了从引导簇 1 的引导交换。

注 引导簇是一个 4 KB 区域，引导簇 0 和 1 通过引导交换功能进行交换。

图 23-12. 引导交换功能

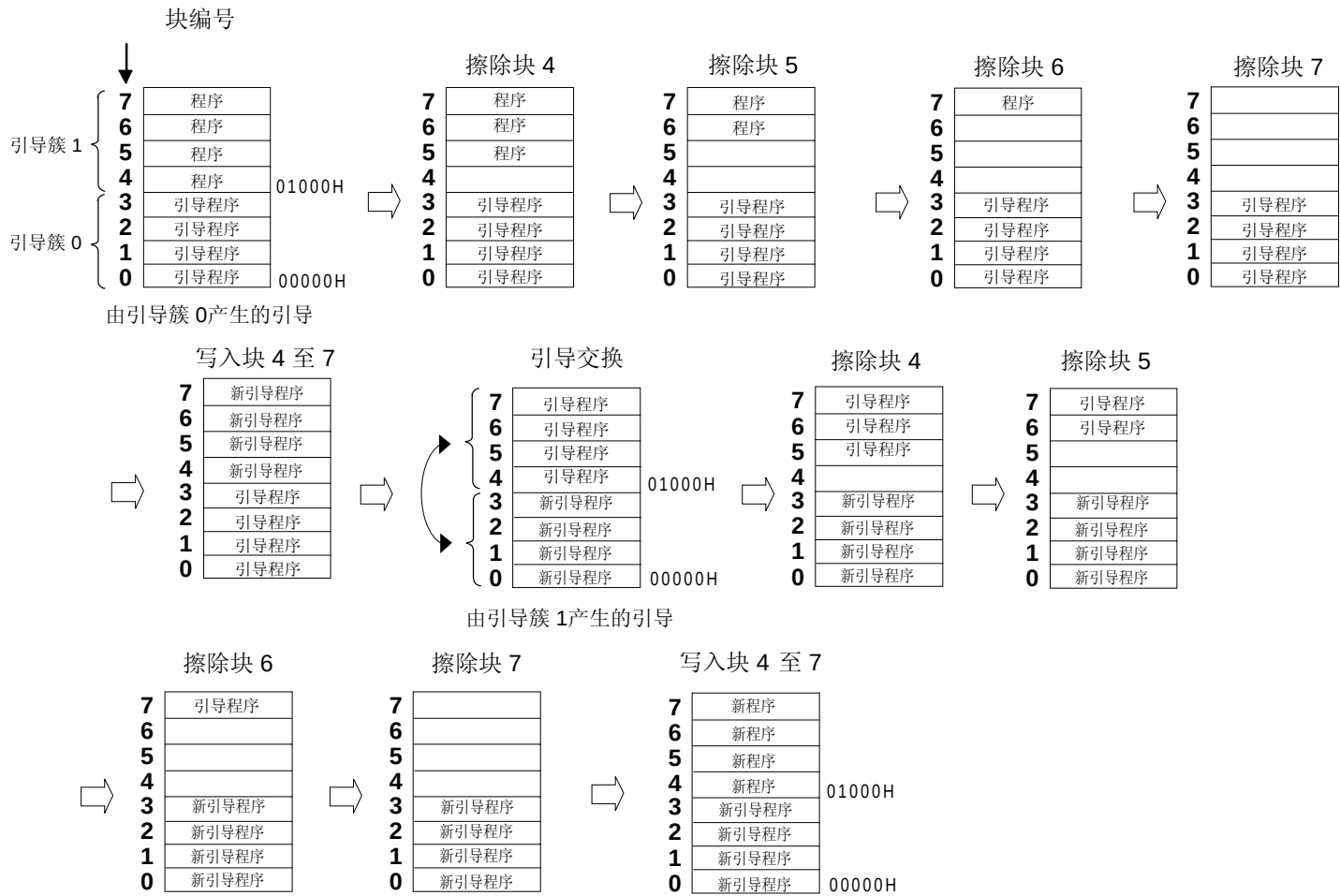


上图的示例如下。

引导簇 0：引导交换前的引导程序区域

引导簇 1：引导交换后的引导程序区域

图 23-13. 引导交换执行示例

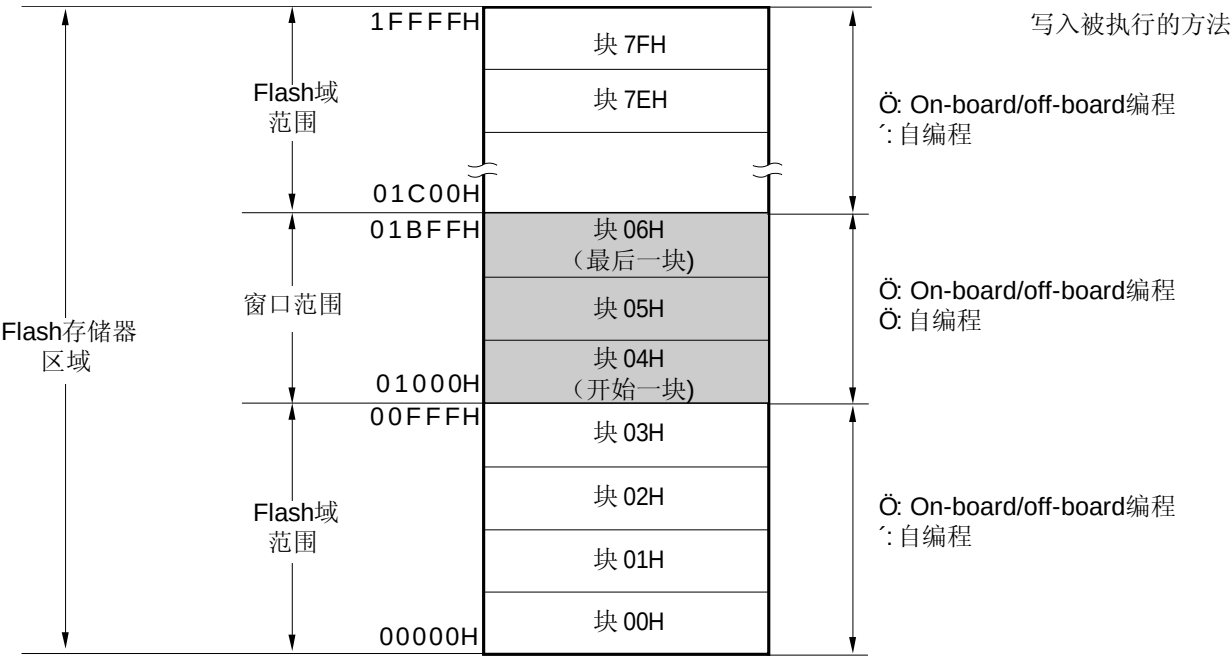


23.8.2 Flash 保护窗口功能

Flash 保护窗口功能作为一种为自编程提供的安全功能。只有在自编程期间，其禁止写入和擦除指定窗口范围之外的区域。

可通过指定起始和结束区块来设置窗口范围。通过在 on-board/off-board 编程和自编程，可以设置或改变窗口范围。在自编程期间禁止写入和擦除窗口范围之外的区域。但是，在 on-board/off-board 编程期间，指定窗口范围以外的区域可以写入和擦除。

图 23-14. Flash 保护窗口设置示例
(目标器件: μ PD78F1014, 开始区块: 04H, 结束区块: 06H)



注意事项 如果引导簇 0 的重写禁止区域超出 Flash 保护窗口范围，则禁止重写引导簇 0 有优先权。

表 23-9. Flash保护窗口功能设置/修改方法和命令之间的关系

编程条件	窗口范围设置/修改方法	执行命令	
		块擦除	写入
自编程	通过设置信息库指定开始和结束块。	只有在窗口范围内允许块擦除。	只有在窗口范围内允许写入。
On-board/off-board 编程	在专用 Flash 存储器编程器的 GUI 指定开始和结束块，等。	超出窗口范围块擦除也允许。	超出窗口范围写入也允许。

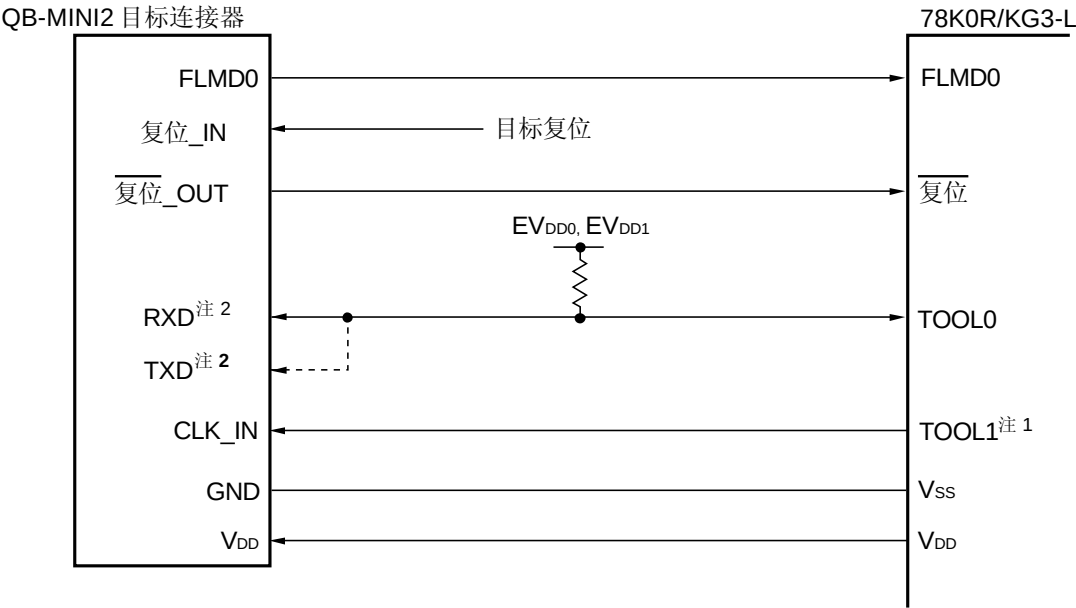
备注 参见 23.7 安全设置，禁止在 on-board/off-board 编程时写入/擦除。

24.1 连接 QB-MINI2 到 78K0R/KG3-L

78K0R/KG3-L 使用 V_{DD} 、FLMD0、 $\overline{\text{RESET}}$ 、TOOL0、TOOL1^{‡ 1}和 V_{SS} 引脚，通过一个片上调试仿真器(QB-MINI2)实现与主机的通信。

注意事项 78K0R/KG3-L 具有片上调试功能，可用于开发和评价。在量产时不要使用产品的片上调试功能，因为 flash 存储器中保证的可重写次数可能会超出，从而不能保证产品的可靠性。对于因使用片上调试功能而产生的问题，NEC 电子不承担任何责任。

图 24-1. QB-MINI2 和 78K0R/KG3-L 的连接示例



- 注**
1. 在 1 线模式下通讯不需要连接，但在 2 线模式下通信需要连接。此时，因为当不连接 QB-MINI2 时 TOOL1 为未使用引脚，根据表 2-2 未使用引脚的连接执行必要的连接。
 2. 由于 QB-MINI2 内 RXD 和 TXD 短接，因此虚线连接不是必须的。当使用其它 Flash 存储器编程器时，在编程器内 RXD 和 TXD 可能不会短接。在这种情况下，他们必须在目标系统上短接。

注意事项 当以双线模式通信时，从 TOOL1 引脚输出的时钟频率为 CPU 时钟频率或 CPU 时钟频率的一半。电阻或铁氧体磁珠可用作对抗由时钟导致的电源电压波动的方法。

备注 自编程在片上调试时，建议 FLMD0 引脚悬空。要通过外部下拉，使用大于 100 kΩ 的电阻。

1 线模式（单线 UART）使用 TOOL0 引脚，或 2 线模式使用 TOOL0 和 TOOL1 引脚，用于串行通信。对于 flash 存储器编程，使用 1 线模式 1 线模式或 2 线模式被用于片上调试。表 24-1 所列为 1 线模式和 2 线模式之间的区别。

表 24-1. 1 线模式和 2 线模式之间的区别

通信模式	Flash 存储器编程功能	调试功能
1 线模式	可用	• 不支持伪实时 RAM 监控 (RRM) 功能
2 线模式	无	• 支持伪实时 RAM 监控 (RRM) 功能。

备注 2 线模式不用于 Flash 编程，因此，即使 TOOL1 引脚连接到 QB-MINI2 的 CLK_IN，写入操作没有问题，正常执行。

24.2 片上调试安全 ID

在 78K0R/KG3-L 中，在 Flash 存储器地址 000C3H 中，有片上调试操作控制位（参见第 22 章 选项字节），在地址 000C4H 至 000CDH 中，有片上调试安全 ID 设置，用于防止第三方读取存储器内容。

当使用引导交换功能时，由于 000C3H、000C4H 至 000CDH 与 010C3H 及 010C4H 至 010CDH 的内容被交换，因此预先设置与 010C3H 以及 010C4H 至 010CDH 的内容相同的值。

关于片上调试安全 ID 详细信息，请参考 **QB-MINI2 程序功能片上调试仿真器用户手册 (U18371E)**。

表 24-2. 片上调试安全 ID

地址	片上调试安全 ID
000C4H 至 000CDH	任何 10 字节的 ID 代码
010C4H 至 010CDH	

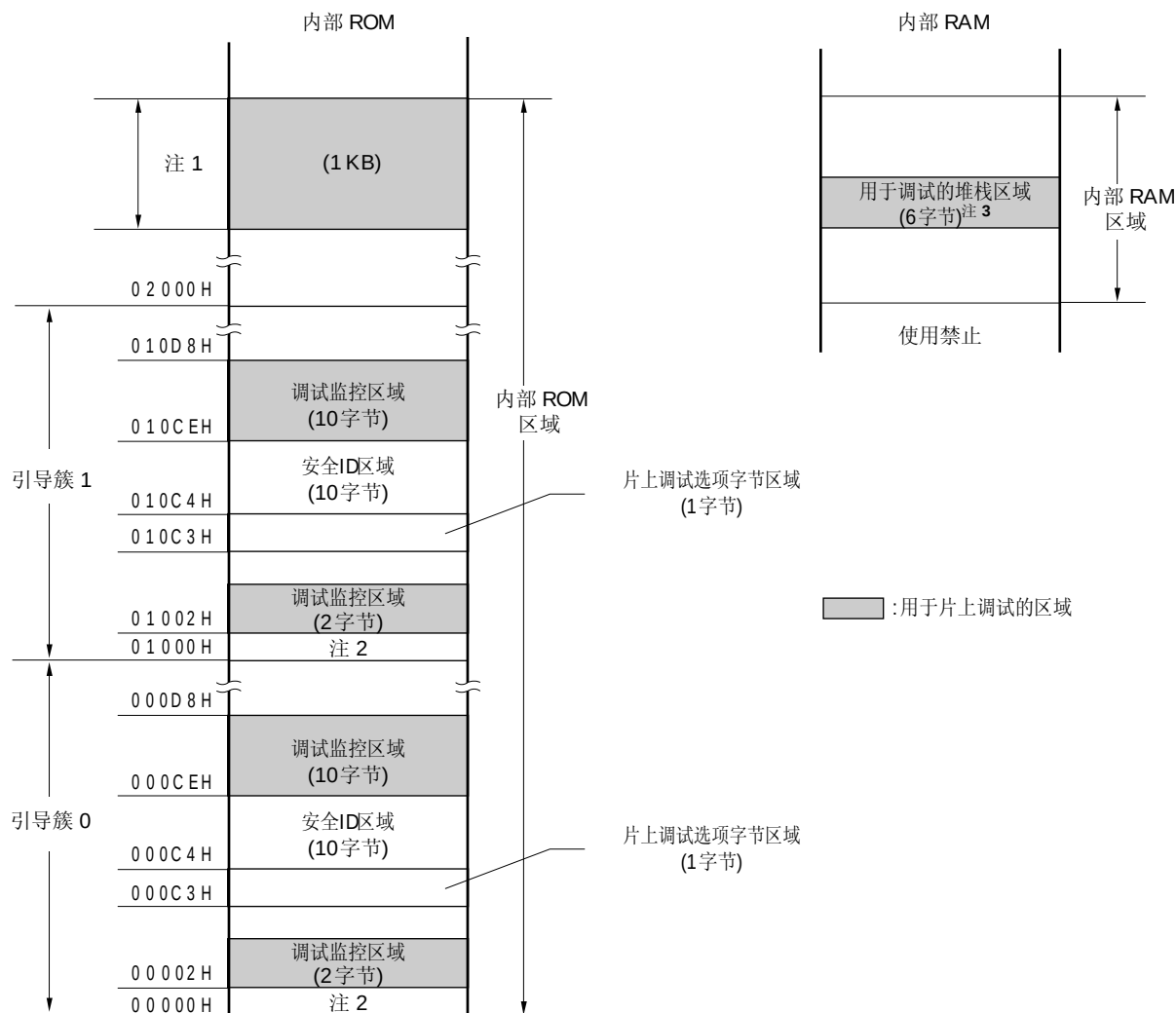
24.3 用户资源的确保

要执行 78K0R/KG3-L 和 QB-MINI2 之间的通信，及每个调试功能，必须预先确存储器空间。如果使用 NEC 电子的汇编程序 RA78K0R 或编译器 CC78K0R，可以通过链接选项设置。

(1) 存储空间的确

图 24-2 中的阴影部分是保留给调试监控程序的区域，所以用户的程序或数据不能被分配在这些空间。当使用片上调试功能时，这些空间必须是安全的，不能被用户程序使用。而且，这些区域也不能被用户程序修改。

图 24-2. 分配了调试监控程序的存储器空间



注 1. 根据产品不同，地址区别如下

产品	内部 ROM	地址
μPD78F1013	96 KB	17C00H 至 17FFFH
μPD78F1014	128 KB	1FC00H 至 1FFFFH

- 调试中，复位向量被重写到分配给监控程序的地址。
- 由于此区域被分配在堆栈区域前一格，所以此区域的地址根据堆栈的增加和减少而不同。也就是说，用于堆栈区域需要 6 个额外字节。

更多确保存储器空间的方法的详细内容，请参考 **QB-MINI2 具有编程功能的片上调试仿真器 (U18371E)**。

25.1 BCD 修正电路功能

通过此电路，可将 BCD（二进制码十进数）码与 BCD 码的加/ 减的结果，生成 BCD 码。
将 A 寄存器作为操作数，加/ 减 BCDADJ 寄存器，通过执行加/减运算，可获得十进数修正操作结果。

25.2 BCD 修正电路使用的寄存器

BCD 修正电路使用如下寄存器。

- BCD 修正结果寄存器 (BCDADJ)

(1) BCD 修正结果寄存器 (BCDADJ)

通过将 A 寄存器作为操作数的加/减指令，获得的加/减结果，其修正值以 BCD 码保存在 BCDADJ 寄存器中。
从 BCDADJ 寄存器读取的值，根据 A 寄存器读取时的值和 CY 与 AC 标志的不同而不同。
BCDADJ 可以由 8 位存储器操作指令读取。
复位输入后此寄存器的值不确定。

图 25-1. BCD 修正结果寄存器 (BCDADJ) 的格式



25.3 BCD 修正电路操作

BCD 修正电路的基本操作如下。

(1) 加指令：通过使用一个 BCD 码值，计算一个 BCD 码值与另一个 BCD 码值相加的结果

- <1> 要执行加法的 BCD 码值存储在 A 寄存器中。
- <2> 通过 A 寄存器与第二操作数（待加 BCD 码的值）的值以二进制相加，二进制运行结果存储在 A 寄存器中，修正值存储在 BCDADJ 寄存器中。
- <3> 通过把 A 寄存器（二进制相加结果）的二进制码的值加到 BCDADJ 寄存器（修正值）上，完成十进制数的修正，并且修正结果存储在 A 寄存器和 CY 标志中。

注意事项 从 BCDADJ 寄存器读取的值，根据 A 寄存器读取时的值和 CY 与 AC 标志的不同而不同。因此，在指令 <2> 后执行指令 <3>，而不执行任何其它指令。要在中断允许状态下执行 BCD 修正，需要在中断功能中保存和恢复 A 寄存器。PSW（CY 标志和 AC 标志）通过 RETI 指令恢复。

下面所示为例子。

例 1: $99 + 89 = 188$

指令	A 寄存器	CY 标志	AC 标志	BCDADJ 寄存器
MOV A, #99H ; <1>	99H	—	—	—
ADD A, #89H ; <2>	22H	1	1	66H
ADD A, !BCDADJ ; <3>	88H	1	0	—

例 2: $85 + 15 = 100$

指令	A 寄存器	CY 标志	AC 标志	BCDADJ 寄存器
MOV A, #85H ; <1>	85H	—	—	—
ADD A, #15H ; <2>	9AH	0	0	06H
ADD A, !BCDADJ ; <3>	00H	1	1	—

例 3: $80 + 80 = 160$

指令	A 寄存器	CY 标志	AC 标志	BCDADJ 寄存器
MOV A, #80H ; <1>	80H	—	—	—
ADD A, #80H ; <2>	00H	1	0	60H
ADD A, !BCDADJ ; <3>	60H	1	0	—

(2) 减指令：通过使用一个 BCD 码值，计算从一个 BCD 码值中减去另一个 BCD 码值的结果

- <1> 被减的 BCD 码值存储在 A 寄存器中。
- <2> 从 A 寄存器中按照二进制减去第二个操作数（减数的 BCD 码值），二进制减法结果存储在 A 寄存器中，修正值存储在 BCDADJ 寄存器中。
- <3> 从 A 寄存器（二进制相减结果）中减去 BCDADJ 寄存器的值（修正值）完成十进制数修正，并将修正结果存储在 A 寄存器和 CY 标志中。

注意事项 从 BCDADJ 寄存器读取的值，根据 A 寄存器读取时的值和 CY 与 AC 标志的不同而不同。因此，在指令 <2> 后执行指令 <3>，而不执行任何其它指令。要在中断允许状态下执行 BCD 修正，需要在中断功能中保存和恢复 A 寄存器。PSW（CY 标志和 AC 标志）通过 RETI 指令恢复。

下面所示为例子。

示例：91 – 52 = 39

指令	A 寄存器	CY 标志	AC 标志	BCDADJ 寄存器
MOV A, #91H ; <1>	91H	–	–	–
SUB A, #52H ; <2>	3FH	0	1	06H
SUB A, !BCDADJ ; <3>	39H	0	0	–

第 26 章 指令集

本章以表格方式列出了 78K0R 微控制器的指令集。如需了解每种指令的操作和操作代码，可参阅 **78K0R 微控制器指令用户手册 (U17792E)**。

备注 **表 26-5 操作列表**中的阴影部分表示操作或指令格式是 78K0R 微控制器新添加的。

26.1 操作列表使用规则

26.1.1 操作数标识符和指定方法

操作数在每个指令的“操作数”栏中进行描述，与指令操作数标识符的描述方法一致 (详细内容请参考汇编程序规范)。如果有两种或两种以上的标识方法，可选其中之一。大写字母和符号，**#**，**!**，**!!**，**\$**，**\$!**，**[]**，和 **ES:** 是关键字，必须按其原样书写。每种符号的含义如下所示。

- **#**: 立即数标识
- **!**: 16 位绝对地址标识
- **!!**: 20 位绝对地址标识
- **\$**: 8 位相对地址标识
- **\$!**: 16 位相对地址标识
- **[]**: 间接地址标识
- **ES**: 扩展地址标识

立即数用来描述一个数值型数据或标号。当使用标号时，注意必须加上符号，**#**，**!**，**!!**，**\$**，**\$!**，**[]**和 **ES:**。

对应操作数寄存器标识符 **r** 和 **rp**，功能名称 (**X**，**A**，**C**，等) 或绝对名称 (下表括号中的名称: **R0**，**R1**，**R2** 等) 都可用于标识。

表 26-1. 操作数标识符和描述方法

标识符	描述方法
r	X (R0) ， A (R1) ， C (R2) ， B (R3) ， E (R4) ， D (R5) ， L (R6) ， H (R7)
rp	AX (RP0) ， BC (RP1) ， DE (RP2) ， HL (RP3)
sfr	特殊功能寄存器符号 (SFR 符号) FFF00H 至 FFFFFH
sfrp	特殊功能寄存器符号 (16 位可操作表 SFR 符号。仅偶数地址 [※]) FFF00H 至 FFFFFH
saddr	FFE20H 至 FFF1FH 立即数或标号
saddrp	FFE20H 至 FF1FH 立即数或标号 (仅用于偶地址 [※])
addr20	00000H 至 FFFFFH 立即数或标号
addr16	0000H 至 FFFFH 立即数或标号 (仅用于 16 位数据传输指令的偶地址 [※])
addr5	0080H 至 00BFH 立即数或标号 (仅用于偶地址)
word	16 位立即数或标号
byte	8 位立即数或标号
bit	3 位立即数或标号
RBn	RB0 至 RB3

注 当奇地址指定时，第0位 = 0。

备注 特殊功能寄存器可描述位符号的操作数 **sfr**。关于特殊功能寄存器的符号，参见表 3-5 **SFR** 列表。

扩展特殊功能寄存器可描述位符号的操作数!addr16。关于特殊功能寄存器的符号，参见表 3-6 扩展 **SFR** 列表 (第二 **SFR**)。

26.1.2 操作栏描述

指令执行时的运算显示在“运算”栏中，并使用下列符号。

表26-2. “运算”栏中的符号

符号	功能
A	A 寄存器; 8 位累加器
X	X 寄存器
B	B 寄存器
C	C 寄存器
D	D 寄存器
E	E 寄存器
H	H 寄存器
L	L 寄存器
ES	ES 寄存器
CS	CS 寄存器
AX	AX 寄存器对; 16 位累加器
BC	BC 寄存器对
DE	DE 寄存器对
HL	HL 寄存器对
PC	程序计数器
SP	堆栈指针
PSW	程序状态字
CY	进位标志
AC	辅助进位标志
Z	零标志
RBS	寄存器 bank 选择标志
IE	中断请求允许标志
()	括号中的地址或寄存器所指的存储单元的内容
X _H , X _L	16 位寄存器: XH = 高 8 位, XL = 低 8 位
X _S , X _H , X _L	20 位寄存器: XS = (位 19 至 16), XH = (位 15 至 8), XL = (位 7 至 0)
^	逻辑与 (AND)
∨	逻辑或 (OR)
⊕	逻辑异或 (异或)
—	数据取反
addr16	16 位立即数
addr20	20 位立即数
jdisp8	带符号的 8 位数据 (偏移量)
jdisp16	带符号的 16 位数据 (偏移量)

26.1.3 标志操作栏的描述

指令执行时的标志值的改变显示在“标志”栏中，并使用下列符号。

表26-3. “标志”栏中的符号

符号	标志值的改变
(Blank)	不变
0	清零
1	设置为 1
×	根据结果进行设置/清零
R	恢复先前保存的值

26.1.4 PREFIX 指令

带“ES:”的指令具有 PREFIX 操作码作为前缀以扩展可访问数据区域至 1 MB 空间(00000H 至 FFFFFH)，通过将 ES 寄存器的值添加到 F0000H 至 FFFFFH 的 64 KB 区域。当 PREFIX 操作码作为前缀附加到目标指令时，在 PREFIX 操作码后只会立即执行一条指令作为添加 ES 寄存器值的地址。
在 PREFIX 指令代码和指令之间不会立即响应中断。

表26-4 PREFIX操作码的使用示例

指令	操作代码				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	—
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	—	—	—	—
MOV A, ES:[HL]	11H	8BH	—	—	—

注意事项 在执行 PREFIX 指令前，用 MOV ES、A 等设置 ES 寄存器的值。

26.2 操作列表

表 26-5. 操作列表 (1/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据 传输	MOV	r, #byte	2	1	–	$r \leftarrow \text{byte}$			
		saddr, #byte	3	1	–	$(\text{saddr}) \leftarrow \text{byte}$			
		sfr, #byte	3	1	–	$\text{sfr} \leftarrow \text{byte}$			
		!addr16, #byte	4	1	–	$(\text{addr16}) \leftarrow \text{byte}$			
		A, r ^{注 3}	1	1	–	$A \leftarrow r$			
		r, A ^{注 3}	1	1	–	$r \leftarrow A$			
		A, saddr	2	1	–	$A \leftarrow (\text{saddr})$			
		saddr, A	2	1	–	$(\text{saddr}) \leftarrow A$			
		A, sfr	2	1	–	$A \leftarrow \text{sfr}$			
		sfr, A	2	1	–	$\text{sfr} \leftarrow A$			
		A, !addr16	3	1	4	$A \leftarrow (\text{addr16})$			
		!addr16, A	3	1	–	$(\text{addr16}) \leftarrow A$			
		PSW, #byte	3	3	–	$\text{PSW} \leftarrow \text{byte}$	×	×	×
		A, PSW	2	1	–	$A \leftarrow \text{PSW}$			
		PSW, A	2	3	–	$\text{PSW} \leftarrow A$	×	×	×
		ES, #byte	2	1	–	$\text{ES} \leftarrow \text{byte}$			
		ES, saddr	3	1	–	$\text{ES} \leftarrow (\text{saddr})$			
		A, ES	2	1	–	$A \leftarrow \text{ES}$			
		ES, A	2	1	–	$\text{ES} \leftarrow A$			
		CS, #byte	3	1	–	$\text{CS} \leftarrow \text{byte}$			
		A, CS	2	1	–	$A \leftarrow \text{CS}$			
		CS, A	2	1	–	$\text{CS} \leftarrow A$			
		A, [DE]	1	1	4	$A \leftarrow (\text{DE})$			
		[DE], A	1	1	–	$(\text{DE}) \leftarrow A$			
		[DE + byte], #byte	3	1	–	$(\text{DE} + \text{byte}) \leftarrow \text{byte}$			
		A, [DE + byte]	2	1	4	$A \leftarrow (\text{DE} + \text{byte})$			
		[DE + byte], A	2	1	–	$(\text{DE} + \text{byte}) \leftarrow A$			
		A, [HL]	1	1	4	$A \leftarrow (\text{HL})$			
		[HL], A	1	1	–	$(\text{HL}) \leftarrow A$			
		[HL + byte], #byte	3	1	–	$(\text{HL} + \text{byte}) \leftarrow \text{byte}$			

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. r = A 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $\times 2 + 3$ 。

表 26-5. 操作列表 (2/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据传输	MOV	A, [HL + byte]	2	1	4	$A \leftarrow (HL + \text{byte})$			
		[HL + byte], A	2	1	–	$(HL + \text{byte}) \leftarrow A$			
		A, [HL + B]	2	1	4	$A \leftarrow (HL + B)$			
		[HL + B], A	2	1	–	$(HL + B) \leftarrow A$			
		A, [HL + C]	2	1	4	$A \leftarrow (HL + C)$			
		[HL + C], A	2	1	–	$(HL + C) \leftarrow A$			
		word[B], #byte	4	1	–	$(B + \text{word}) \leftarrow \text{byte}$			
		A, word[B]	3	1	4	$A \leftarrow (B + \text{word})$			
		word[B], A	3	1	–	$(B + \text{word}) \leftarrow A$			
		word[C], #byte	4	1	–	$(C + \text{word}) \leftarrow \text{byte}$			
		A, word[C]	3	1	4	$A \leftarrow (C + \text{word})$			
		word[C], A	3	1	–	$(C + \text{word}) \leftarrow A$			
		word[BC], #byte	4	1	–	$(BC + \text{word}) \leftarrow \text{byte}$			
		A, word[BC]	3	1	4	$A \leftarrow (BC + \text{word})$			
		word[BC], A	3	1	–	$(BC + \text{word}) \leftarrow A$			
		[SP + byte], #byte	3	1	–	$(SP + \text{byte}) \leftarrow \text{byte}$			
		A, [SP + byte]	2	1	–	$A \leftarrow (SP + \text{byte})$			
		[SP + byte], A	2	1	–	$(SP + \text{byte}) \leftarrow A$			
		B, saddr	2	1	–	$B \leftarrow (\text{saddr})$			
		B, !addr16	3	1	4	$B \leftarrow (\text{addr16})$			
		C, saddr	2	1	–	$C \leftarrow (\text{saddr})$			
		C, !addr16	3	1	4	$C \leftarrow (\text{addr16})$			
		X, saddr	2	1	–	$X \leftarrow (\text{saddr})$			
		X, !addr16	3	1	4	$X \leftarrow (\text{addr16})$			
		ES:!addr16, #byte	5	2	–	$(ES, \text{addr16}) \leftarrow \text{byte}$			
		A, ES:!addr16	4	2	5	$A \leftarrow (ES, \text{addr16})$			
		ES:!addr16, A	4	2	–	$(ES, \text{addr16}) \leftarrow A$			
		A, ES:[DE]	2	2	5	$A \leftarrow (ES, DE)$			
		ES:[DE], A	2	2	–	$(ES, DE) \leftarrow A$			
		ES:[DE + byte], #byte	4	2	–	$((ES, DE) + \text{byte}) \leftarrow \text{byte}$			
		A, ES:[DE + byte]	3	2	5	$A \leftarrow ((ES, DE) + \text{byte})$			
		ES:[DE + byte], A	3	2	–	$((ES, DE) + \text{byte}) \leftarrow A$			

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $x2+3$ 。

表 26-5. 操作列表 (3/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据 传输	MOV	A, ES:[HL]	2	2	5	$A \leftarrow (ES, HL)$			
		ES:[HL], A	2	2	–	$(ES, HL) \leftarrow A$			
		ES:[HL + byte], #byte	4	2	–	$((ES, HL) + \text{byte}) \leftarrow \text{byte}$			
		A, ES:[HL + byte]	3	2	5	$A \leftarrow ((ES, HL) + \text{byte})$			
		ES:[HL + byte], A	3	2	–	$((ES, HL) + \text{byte}) \leftarrow A$			
		A, ES:[HL + B]	3	2	5	$A \leftarrow ((ES, HL) + B)$			
		ES:[HL + B], A	3	2	–	$((ES, HL) + B) \leftarrow A$			
		A, ES:[HL + C]	3	2	5	$A \leftarrow ((ES, HL) + C)$			
		ES:[HL + C], A	3	2	–	$((ES, HL) + C) \leftarrow A$			
		ES:word[B], #byte	5	2	–	$((ES, B) + \text{word}) \leftarrow \text{byte}$			
		A, ES:word[B]	4	2	5	$A \leftarrow ((ES, B) + \text{word})$			
		ES:word[B], A	4	2	–	$((ES, B) + \text{word}) \leftarrow A$			
		ES:word[C], #byte	5	2	–	$((ES, C) + \text{word}) \leftarrow \text{byte}$			
		A, ES:word[C]	4	2	5	$A \leftarrow ((ES, C) + \text{word})$			
		ES:word[C], A	4	2	–	$((ES, C) + \text{word}) \leftarrow A$			
		ES:word[BC], #byte	5	2	–	$((ES, BC) + \text{word}) \leftarrow \text{byte}$			
		A, ES:word[BC]	4	2	5	$A \leftarrow ((ES, BC) + \text{word})$			
		ES:word[BC], A	4	2	–	$((ES, BC) + \text{word}) \leftarrow A$			
		B, ES:!addr16	4	2	5	$B \leftarrow (ES, \text{addr16})$			
		C, ES:!addr16	4	2	5	$C \leftarrow (ES, \text{addr16})$			
		X, ES:!addr16	4	2	5	$X \leftarrow (ES, \text{addr16})$			
	XCH	A, r ^{#3}	1 (r=X) 2 (除 r=X)	1	–	$A \longleftrightarrow r$			
		A, saddr	3	2	–	$A \longleftrightarrow (\text{saddr})$			
		A, sfr	3	2	–	$A \longleftrightarrow \text{sfr}$			
		A, !addr16	4	2	–	$A \longleftrightarrow (\text{addr16})$			
		A, [DE]	2	2	–	$A \longleftrightarrow (DE)$			
		A, [DE + byte]	3	2	–	$A \longleftrightarrow (DE + \text{byte})$			
		A, [HL]	2	2	–	$A \longleftrightarrow (HL)$			
		A, [HL + byte]	3	2	–	$A \longleftrightarrow (HL + \text{byte})$			
		A, [HL + B]	2	2	–	$A \longleftrightarrow (HL + B)$			
		A, [HL + C]	2	2	–	$A \longleftrightarrow (HL + C)$			

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. r = A 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $x2+3$ 。

表 26-5. 操作列表 (4/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据传输	XCH	A, ES:!addr16	5	3	—	$A \leftrightarrow (ES, \text{addr16})$			
		A, ES:[DE]	3	3	—	$A \leftrightarrow (ES, DE)$			
		A, ES:[DE + byte]	4	3	—	$A \leftrightarrow ((ES, DE) + \text{byte})$			
		A, ES:[HL]	3	3	—	$A \leftrightarrow (ES, HL)$			
		A, ES:[HL + byte]	4	3	—	$A \leftrightarrow ((ES, HL) + \text{byte})$			
		A, ES:[HL + B]	3	3	—	$A \leftrightarrow ((ES, HL) + B)$			
		A, ES:[HL + C]	3	3	—	$A \leftrightarrow ((ES, HL) + C)$			
	ONEB	A	1	1	—	$A \leftarrow 01H$			
		X	1	1	—	$X \leftarrow 01H$			
		B	1	1	—	$B \leftarrow 01H$			
		C	1	1	—	$C \leftarrow 01H$			
		saddr	2	1	—	$(saddr) \leftarrow 01H$			
		!addr16	3	1	—	$(addr16) \leftarrow 01H$			
		ES:!addr16	4	2	—	$(ES, addr16) \leftarrow 01H$			
	CLRB	A	1	1	—	$A \leftarrow 00H$			
		X	1	1	—	$X \leftarrow 00H$			
		B	1	1	—	$B \leftarrow 00H$			
		C	1	1	—	$C \leftarrow 00H$			
		saddr	2	1	—	$(saddr) \leftarrow 00H$			
		!addr16	3	1	—	$(addr16) \leftarrow 00H$			
		ES:!addr16	4	2	—	$(ES, addr16) \leftarrow 00H$			
	MOVS	[HL + byte], X	3	1	—	$(HL + \text{byte}) \leftarrow X$	×		×
		ES:[HL + byte], X	4	2	—	$(ES, HL + \text{byte}) \leftarrow X$	×		×
16 位数据传输	MOVW	rp, #word	3	1	—	$rp \leftarrow \text{word}$			
		saddrp, #word	4	1	—	$(saddrp) \leftarrow \text{word}$			
		sfrp, #word	4	1	—	$sfrp \leftarrow \text{word}$			
		AX, saddrp	2	1	—	$AX \leftarrow (saddrp)$			
		saddrp, AX	2	1	—	$(saddrp) \leftarrow AX$			
		AX, sfrp	2	1	—	$AX \leftarrow sfrp$			
		sfrp, AX	2	1	—	$sfrp \leftarrow AX$			
		AX, rp ^{注 3}	1	1	—	$AX \leftarrow rp$			
		rp, AX ^{注 3}	1	1	—	$rp \leftarrow AX$			

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. rp = AX 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 x2+3。

表 26-5. 操作列表 (5/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位数 数据传输	MOVW	AX, !addr16	3	1	4	$AX \leftarrow (addr16)$			
		!addr16, AX	3	1	–	$(addr16) \leftarrow AX$			
		AX, [DE]	1	1	4	$AX \leftarrow (DE)$			
		[DE], AX	1	1	–	$(DE) \leftarrow AX$			
		AX, [DE + byte]	2	1	4	$AX \leftarrow (DE + byte)$			
		[DE + byte], AX	2	1	–	$(DE + byte) \leftarrow AX$			
		AX, [HL]	1	1	4	$AX \leftarrow (HL)$			
		[HL], AX	1	1	–	$(HL) \leftarrow AX$			
		AX, [HL + byte]	2	1	4	$AX \leftarrow (HL + byte)$			
		[HL + byte], AX	2	1	–	$(HL + byte) \leftarrow AX$			
		AX, word[B]	3	1	4	$AX \leftarrow (B + word)$			
		word[B], AX	3	1	–	$(B + word) \leftarrow AX$			
		AX, word[C]	3	1	4	$AX \leftarrow (C + word)$			
		word[C], AX	3	1	–	$(C + word) \leftarrow AX$			
		AX, word[BC]	3	1	4	$AX \leftarrow (BC + word)$			
		word[BC], AX	3	1	–	$(BC + word) \leftarrow AX$			
		AX, [SP + byte]	2	1	–	$AX \leftarrow (SP + byte)$			
		[SP + byte], AX	2	1	–	$(SP + byte) \leftarrow AX$			
		BC, saddrp	2	1	–	$BC \leftarrow (saddrp)$			
		BC, !addr16	3	1	4	$BC \leftarrow (addr16)$			
		DE, saddrp	2	1	–	$DE \leftarrow (saddrp)$			
		DE, !addr16	3	1	4	$DE \leftarrow (addr16)$			
		HL, saddrp	2	1	–	$HL \leftarrow (saddrp)$			
		HL, !addr16	3	1	4	$HL \leftarrow (addr16)$			
		AX, ES:!addr16	4	2	5	$AX \leftarrow (ES, addr16)$			
		ES:!addr16, AX	4	2	–	$(ES, addr16) \leftarrow AX$			
		AX, ES:[DE]	2	2	5	$AX \leftarrow (ES, DE)$			
		ES:[DE], AX	2	2	–	$(ES, DE) \leftarrow AX$			
		AX, ES:[DE + byte]	3	2	5	$AX \leftarrow ((ES, DE) + byte)$			
		ES:[DE + byte], AX	3	2	–	$((ES, DE) + byte) \leftarrow AX$			
		AX, ES:[HL]	2	2	5	$AX \leftarrow (ES, HL)$			
		ES:[HL], AX	2	2	–	$(ES, HL) \leftarrow AX$			

- 注 1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
2. 当程序存储器区域被访问时。

- 备注 1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $\times 2 + 3$ 。

表 26-5. 操作列表 (6/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位数 数据传输	MOVW	AX, ES:[HL + byte]	3	2	5	$AX \leftarrow ((ES, HL) + \text{byte})$			
		ES:[HL + byte], AX	3	2	–	$((ES, HL) + \text{byte}) \leftarrow AX$			
		AX, ES:word[B]	4	2	5	$AX \leftarrow ((ES, B) + \text{word})$			
		ES:word[B], AX	4	2	–	$((ES, B) + \text{word}) \leftarrow AX$			
		AX, ES:word[C]	4	2	5	$AX \leftarrow ((ES, C) + \text{word})$			
		ES:word[C], AX	4	2	–	$((ES, C) + \text{word}) \leftarrow AX$			
		AX, ES:word[BC]	4	2	5	$AX \leftarrow ((ES, BC) + \text{word})$			
		ES:word[BC], AX	4	2	–	$((ES, BC) + \text{word}) \leftarrow AX$			
		BC, ES:!addr16	4	2	5	$BC \leftarrow (ES, \text{addr16})$			
		DE, ES:!addr16	4	2	5	$DE \leftarrow (ES, \text{addr16})$			
		HL, ES:!addr16	4	2	5	$HL \leftarrow (ES, \text{addr16})$			
	XCHW	AX, rp ^{注 3}	1	1	–	$AX \leftrightarrow rp$			
	ONEW	AX	1	1	–	$AX \leftarrow 0001H$			
		BC	1	1	–	$BC \leftarrow 0001H$			
	CLRW	AX	1	1	–	$AX \leftarrow 0000H$			
		BC	1	1	–	$BC \leftarrow 0000H$			
8 位操作	ADD	A, #byte	2	1	–	$A, CY \leftarrow A + \text{byte}$	x	x	x
		saddr, #byte	3	2	–	$(saddr), CY \leftarrow (saddr) + \text{byte}$	x	x	x
		A, r ^{注 4}	2	1	–	$A, CY \leftarrow A + r$	x	x	x
		r, A	2	1	–	$r, CY \leftarrow r + A$	x	x	x
		A, saddr	2	1	–	$A, CY \leftarrow A + (saddr)$	x	x	x
		A, !addr16	3	1	4	$A, CY \leftarrow A + (\text{addr16})$	x	x	x
		A, [HL]	1	1	4	$A, CY \leftarrow A + (HL)$	x	x	x
		A, [HL + byte]	2	1	4	$A, CY \leftarrow A + (HL + \text{byte})$	x	x	x
		A, [HL + B]	2	1	4	$A, CY \leftarrow A + (HL + B)$	x	x	x
		A, [HL + C]	2	1	4	$A, CY \leftarrow A + (HL + C)$	x	x	x
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A + (ES, \text{addr16})$	x	x	x
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A + (ES, HL)$	x	x	x
		A, ES:[HL + byte]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + \text{byte})$	x	x	x
		A, ES:[HL + B]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + B)$	x	x	x
		A, ES:[HL + C]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + C)$	x	x	x

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. rp = AX 除外
 4. r = A 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 x2+3。

表 26-5. 操作列表 (7/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	ADDC	A, #byte	2	1	–	$A, CY \leftarrow A + \text{byte} + CY$	×	×	×
		saddr, #byte	3	2	–	$(saddr), CY \leftarrow (saddr) + \text{byte} + CY$	×	×	×
		A, r ^{注 3}	2	1	–	$A, CY \leftarrow A + r + CY$	×	×	×
		r, A	2	1	–	$r, CY \leftarrow r + A + CY$	×	×	×
		A, saddr	2	1	–	$A, CY \leftarrow A + (saddr) + CY$	×	×	×
		A, !addr16	3	1	4	$A, CY \leftarrow A + (\text{addr16}) + CY$	×	×	×
		A, [HL]	1	1	4	$A, CY \leftarrow A + (HL) + CY$	×	×	×
		A, [HL + byte]	2	1	4	$A, CY \leftarrow A + (HL + \text{byte}) + CY$	×	×	×
		A, [HL + B]	2	1	4	$A, CY \leftarrow A + (HL + B) + CY$	×	×	×
		A, [HL + C]	2	1	4	$A, CY \leftarrow A + (HL + C) + CY$	×	×	×
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A + (ES, \text{addr16}) + CY$	×	×	×
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A + (ES, HL) + CY$	×	×	×
		A, ES:[HL + byte]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + \text{byte}) + CY$	×	×	×
		A, ES:[HL + B]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + B) + CY$	×	×	×
		A, ES:[HL + C]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + C) + CY$	×	×	×
	SUB	A, #byte	2	1	–	$A, CY \leftarrow A - \text{byte}$	×	×	×
		saddr, #byte	3	2	–	$(saddr), CY \leftarrow (saddr) - \text{byte}$	×	×	×
		A, r ^{注 3}	2	1	–	$A, CY \leftarrow A - r$	×	×	×
		r, A	2	1	–	$r, CY \leftarrow r - A$	×	×	×
		A, saddr	2	1	–	$A, CY \leftarrow A - (saddr)$	×	×	×
		A, !addr16	3	1	4	$A, CY \leftarrow A - (\text{addr16})$	×	×	×
		A, [HL]	1	1	4	$A, CY \leftarrow A - (HL)$	×	×	×
		A, [HL + byte]	2	1	4	$A, CY \leftarrow A - (HL + \text{byte})$	×	×	×
		A, [HL + B]	2	1	4	$A, CY \leftarrow A - (HL + B)$	×	×	×
		A, [HL + C]	2	1	4	$A, CY \leftarrow A - (HL + C)$	×	×	×
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A - (ES:\text{addr16})$	×	×	×
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A - (ES:HL)$	×	×	×
		A, ES:[HL + byte]	3	2	5	$A, CY \leftarrow A - ((ES:HL) + \text{byte})$	×	×	×
		A, ES:[HL + B]	3	2	5	$A, CY \leftarrow A - ((ES:HL) + B)$	×	×	×
		A, ES:[HL + C]	3	2	5	$A, CY \leftarrow A - ((ES:HL) + C)$	×	×	×

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. r = A 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $x2+3$ 。

表 26-5. 操作列表 (8/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	SUBC	A, #byte	2	1	–	$A, CY \leftarrow A - \text{byte} - CY$	×	×	×
		saddr, #byte	3	2	–	$(saddr), CY \leftarrow (saddr) - \text{byte} - CY$	×	×	×
		A, r ^{注 3}	2	1	–	$A, CY \leftarrow A - r - CY$	×	×	×
		r, A	2	1	–	$r, CY \leftarrow r - A - CY$	×	×	×
		A, saddr	2	1	–	$A, CY \leftarrow A - (saddr) - CY$	×	×	×
		A, !addr16	3	1	4	$A, CY \leftarrow A - (\text{addr16}) - CY$	×	×	×
		A, [HL]	1	1	4	$A, CY \leftarrow A - (HL) - CY$	×	×	×
		A, [HL + byte]	2	1	4	$A, CY \leftarrow A - (HL + \text{byte}) - CY$	×	×	×
		A, [HL + B]	2	1	4	$A, CY \leftarrow A - (HL + B) - CY$	×	×	×
		A, [HL + C]	2	1	4	$A, CY \leftarrow A - (HL + C) - CY$	×	×	×
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A - (ES:\text{addr16}) - CY$	×	×	×
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A - (ES:HL) - CY$	×	×	×
		A, ES:[HL + byte]	3	2	5	$A, CY \leftarrow A - ((ES:HL) + \text{byte}) - CY$	×	×	×
		A, ES:[HL + B]	3	2	5	$A, CY \leftarrow A - ((ES:HL) + B) - CY$	×	×	×
		A, ES:[HL + C]	3	2	5	$A, CY \leftarrow A - ((ES:HL) + C) - CY$	×	×	×
	AND	A, #byte	2	1	–	$A \leftarrow A \wedge \text{byte}$	×		
		saddr, #byte	3	2	–	$(saddr) \leftarrow (saddr) \wedge \text{byte}$	×		
		A, r ^{注 3}	2	1	–	$A \leftarrow A \wedge r$	×		
		r, A	2	1	–	$r \leftarrow r \wedge A$	×		
		A, saddr	2	1	–	$A \leftarrow A \wedge (saddr)$	×		
		A, !addr16	3	1	4	$A \leftarrow A \wedge (\text{addr16})$	×		
		A, [HL]	1	1	4	$A \leftarrow A \wedge (HL)$	×		
		A, [HL + byte]	2	1	4	$A \leftarrow A \wedge (HL + \text{byte})$	×		
		A, [HL + B]	2	1	4	$A \leftarrow A \wedge (HL + B)$	×		
		A, [HL + C]	2	1	4	$A \leftarrow A \wedge (HL + C)$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \wedge (ES:\text{addr16})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \wedge (ES:HL)$	×		
		A, ES:[HL + byte]	3	2	5	$A \leftarrow A \wedge ((ES:HL) + \text{byte})$	×		
		A, ES:[HL + B]	3	2	5	$A \leftarrow A \wedge ((ES:HL) + B)$	×		
		A, ES:[HL + C]	3	2	5	$A \leftarrow A \wedge ((ES:HL) + C)$	×		

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. r = A 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 x2+3。

表 26-5. 操作列表 (9/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	OR	A, #byte	2	1	–	$A \leftarrow A \vee \text{byte}$	×		
		saddr, #byte	3	2	–	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	×		
		A, r ^{注 3}	2	1	–	$A \leftarrow A \vee r$	×		
		r, A	2	1	–	$r \leftarrow r \vee A$	×		
		A, saddr	2	1	–	$A \leftarrow A \vee (\text{saddr})$	×		
		A, !addr16	3	1	4	$A \leftarrow A \vee (\text{addr16})$	×		
		A, [HL]	1	1	4	$A \leftarrow A \vee (\text{HL})$	×		
		A, [HL + byte]	2	1	4	$A \leftarrow A \vee (\text{HL} + \text{byte})$	×		
		A, [HL + B]	2	1	4	$A \leftarrow A \vee (\text{HL} + B)$	×		
		A, [HL + C]	2	1	4	$A \leftarrow A \vee (\text{HL} + C)$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \vee (\text{ES:addr16})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \vee (\text{ES:HL})$	×		
		A, ES:[HL + byte]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + \text{byte})$	×		
		A, ES:[HL + B]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + B)$	×		
		A, ES:[HL + C]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + C)$	×		
	XOR	A, #byte	2	1	–	$A \leftarrow A \nabla \text{byte}$	×		
		saddr, #byte	3	2	–	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$	×		
		A, r ^{注 3}	2	1	–	$A \leftarrow A \nabla r$	×		
		r, A	2	1	–	$r \leftarrow r \nabla A$	×		
		A, saddr	2	1	–	$A \leftarrow A \nabla (\text{saddr})$	×		
		A, !addr16	3	1	4	$A \leftarrow A \nabla (\text{addr16})$	×		
		A, [HL]	1	1	4	$A \leftarrow A \nabla (\text{HL})$	×		
		A, [HL + byte]	2	1	4	$A \leftarrow A \nabla (\text{HL} + \text{byte})$	×		
		A, [HL + B]	2	1	4	$A \leftarrow A \nabla (\text{HL} + B)$	×		
		A, [HL + C]	2	1	4	$A \leftarrow A \nabla (\text{HL} + C)$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \nabla (\text{ES:addr16})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \nabla (\text{ES:HL})$	×		
		A, ES:[HL + byte]	3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + \text{byte})$	×		
		A, ES:[HL + B]	3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + B)$	×		
		A, ES:[HL + C]	3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + C)$	×		

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. r = A 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $x2+3$ 。

表 26-5. 操作列表 (10/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	CMP	A, #byte	2	1	–	A – byte	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r ^{# 3}	2	1	–	A – r	×	×	×
		r, A	2	1	–	r – A	×	×	×
		A, saddr	2	1	–	A – (saddr)	×	×	×
		A, !addr16	3	1	4	A – (addr16)	×	×	×
		A, [HL]	1	1	4	A – (HL)	×	×	×
		A, [HL + byte]	2	1	4	A – (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A – (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A – (HL + C)	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		A, ES:!addr16	4	2	5	A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A – ((ES:HL) + C)	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
	CMPO	A	1	1	–	A – 00H	×	×	×
		X	1	1	–	X – 00H	×	×	×
		B	1	1	–	B – 00H	×	×	×
		C	1	1	–	C – 00H	×	×	×
		saddr	2	1	–	(saddr) – 00H	×	×	×
		!addr16	3	1	4	(addr16) – 00H	×	×	×
		ES:!addr16	4	2	5	(ES:addr16) – 00H	×	×	×
	CMPS	X, [HL + byte]	3	1	4	X – (HL + byte)	×	×	×
		X, ES:[HL + byte]	4	2	5	X – ((ES:HL) + byte)	×	×	×

注 1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。

2. 当程序存储器区域被访问时。

3. r = A 除外

备注 1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})

2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 x2+3。

表 26-5. 操作列表 (11/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位操作	ADDW	AX, #word	3	1	–	AX, CY $\leftarrow$ AX + word	×	×	×
		AX, AX	1	1	–	AX, CY $\leftarrow$ AX + AX	×	×	×
		AX, BC	1	1	–	AX, CY $\leftarrow$ AX + BC	×	×	×
		AX, DE	1	1	–	AX, CY $\leftarrow$ AX + DE	×	×	×
		AX, HL	1	1	–	AX, CY $\leftarrow$ AX + HL	×	×	×
		AX, saddrp	2	1	–	AX, CY $\leftarrow$ AX + (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY $\leftarrow$ AX + (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY $\leftarrow$ AX + (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY $\leftarrow$ AX + (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY $\leftarrow$ AX + ((ES:HL) + byte)	×	×	×
	SUBW	AX, #word	3	1	–	AX, CY $\leftarrow$ AX – word	×	×	×
		AX, BC	1	1	–	AX, CY $\leftarrow$ AX – BC	×	×	×
		AX, DE	1	1	–	AX, CY $\leftarrow$ AX – DE	×	×	×
		AX, HL	1	1	–	AX, CY $\leftarrow$ AX – HL	×	×	×
		AX, saddrp	2	1	–	AX, CY $\leftarrow$ AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY $\leftarrow$ AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY $\leftarrow$ AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY $\leftarrow$ AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY $\leftarrow$ AX – ((ES:HL) + byte)	×	×	×
	CMPW	AX, #word	3	1	–	AX – word	×	×	×
		AX, BC	1	1	–	AX – BC	×	×	×
		AX, DE	1	1	–	AX – DE	×	×	×
		AX, HL	1	1	–	AX – HL	×	×	×
		AX, saddrp	2	1	–	AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX – ((ES:HL) + byte)	×	×	×
乘	MULU	X	1	1	–	AX $\leftarrow$ A $\times$ X			

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $x2+3$ 。

表 26-5. 操作列表 (12/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
递增/递减	INC	r	1	1	—	$r \leftarrow r + 1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr) + 1$	×	×	
		!addr16	3	2	—	$(addr16) \leftarrow (addr16) + 1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte) + 1$	×	×	
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) + 1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) + 1$	×	×	
	DEC	r	1	1	—	$r \leftarrow r - 1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr) - 1$	×	×	
		!addr16	3	2	—	$(addr16) \leftarrow (addr16) - 1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte) - 1$	×	×	
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) - 1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) - 1$	×	×	
	INCW	rp	1	1	—	$rp \leftarrow rp + 1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp) + 1$			
		!addr16	3	2	—	$(addr16) \leftarrow (addr16) + 1$			
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte) + 1$			
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) + 1$			
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) + 1$			
	DECW	rp	1	1	—	$rp \leftarrow rp - 1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp) - 1$			
		!addr16	3	2	—	$(addr16) \leftarrow (addr16) - 1$			
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte) - 1$			
		ES:!addr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) - 1$			
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) - 1$			
移位	SHR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow 0) \times cnt$			×
	SHRW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow 0) \times cnt$			×
	SHL	A, cnt	2	1	—	$(CY \leftarrow A_7, A_m \leftarrow A_{m-1}, A_0 \leftarrow 0) \times cnt$			×
		B, cnt	2	1	—	$(CY \leftarrow B_7, B_m \leftarrow B_{m-1}, B_0 \leftarrow 0) \times cnt$			×
		C, cnt	2	1	—	$(CY \leftarrow C_7, C_m \leftarrow C_{m-1}, C_0 \leftarrow 0) \times cnt$			×
	SHLW	AX, cnt	2	1	—	$(CY \leftarrow AX_{15}, AX_m \leftarrow AX_{m-1}, AX_0 \leftarrow 0) \times cnt$			×
		BC, cnt	2	1	—	$(CY \leftarrow BC_{15}, BC_m \leftarrow BC_{m-1}, BC_0 \leftarrow 0) \times cnt$			×
	SAR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow A_7) \times cnt$			×
	SARW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow AX_{15}) \times cnt$			×

- 注 1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注 1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $x2+3$ 。
 3. cnt 表示位移数。

表 26-5. 操作列表 (13/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
循环	ROR	A, 1	2	1	–	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			×
	ROL	A, 1	2	1	–	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			×
	RORC	A, 1	2	1	–	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			×
	ROLC	A, 1	2	1	–	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			×
	ROLWC	AX,1	2	1	–	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			×
		BC,1	2	1	–	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			×
位操作	MOV1	CY, saddr.bit	3	1	–	$CY \leftarrow (saddr).bit$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow sfr.bit$			×
		CY, A.bit	2	1	–	$CY \leftarrow A.bit$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow PSW.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow (HL).bit$			×
		saddr.bit, CY	3	2	–	$(saddr).bit \leftarrow CY$			
		sfr.bit, CY	3	2	–	$sfr.bit \leftarrow CY$			
		A.bit, CY	2	1	–	$A.bit \leftarrow CY$			
		PSW.bit, CY	3	4	–	$PSW.bit \leftarrow CY$	×	×	
		[HL].bit, CY	2	2	–	$(HL).bit \leftarrow CY$			
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow (ES, HL).bit$			×
		ES:[HL].bit, CY	3	3	–	$(ES, HL).bit \leftarrow CY$			
	AND1	CY, saddr.bit	3	1	–	$CY \leftarrow CY \wedge (saddr).bit$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \wedge sfr.bit$			×
		CY, A.bit	2	1	–	$CY \leftarrow CY \wedge A.bit$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \wedge PSW.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \wedge (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \wedge (ES, HL).bit$			×
	OR1	CY, saddr.bit	3	1	–	$CY \leftarrow CY \vee (saddr).bit$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \vee sfr.bit$			×
		CY, A.bit	2	1	–	$CY \leftarrow CY \vee A.bit$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \vee PSW.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \vee (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \vee (ES, HL).bit$			×

- 注 1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注 1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $x2+3$ 。

表 26-5 操作列表 (14/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
位操作	XOR1	CY, saddr.bit	3	1	–	$CY \leftarrow CY \oplus (saddr).bit$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \oplus sfr.bit$			×
		CY, A.bit	2	1	–	$CY \leftarrow CY \oplus A.bit$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \oplus PSW.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \oplus (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \oplus (ES, HL).bit$			×
	SET1	saddr.bit	3	2	–	$(saddr).bit \leftarrow 1$			
		sfr.bit	3	2	–	$sfr.bit \leftarrow 1$			
		A.bit	2	1	–	$A.bit \leftarrow 1$			
		!addr16.bit	4	2	–	$(addr16).bit \leftarrow 1$			
		PSW.bit	3	4	–	$PSW.bit \leftarrow 1$	×	×	×
		[HL].bit	2	2	–	$(HL).bit \leftarrow 1$			
		ES:!addr16.bit	5	3	–	$(ES, addr16).bit \leftarrow 1$			
		ES:[HL].bit	3	3	–	$(ES, HL).bit \leftarrow 1$			
	CLR1	saddr.bit	3	2	–	$(saddr).bit \leftarrow 0$			
		sfr.bit	3	2	–	$sfr.bit \leftarrow 0$			
		A.bit	2	1	–	$A.bit \leftarrow 0$			
		!addr16.bit	4	2	–	$(addr16).bit \leftarrow 0$			
		PSW.bit	3	4	–	$PSW.bit \leftarrow 0$	×	×	×
		[HL].bit	2	2	–	$(HL).bit \leftarrow 0$			
		ES:!addr16.bit	5	3	–	$(ES, addr16).bit \leftarrow 0$			
		ES:[HL].bit	3	3	–	$(ES, HL).bit \leftarrow 0$			
	SET1	CY	2	1	–	$CY \leftarrow 1$			1
	CLR1	CY	2	1	–	$CY \leftarrow 0$			0
	NOT1	CY	2	1	–	$CY \leftarrow \overline{CY}$			×

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $x2+3$ 。

表 26-5. 操作列表 (15/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
调用/ 返回	CALL	rp	2	3	–	$(SP - 2) \leftarrow (PC + 2)_S, (SP - 3) \leftarrow (PC + 2)_H,$ $(SP - 4) \leftarrow (PC + 2)_L, PC \leftarrow CS, rp,$ $SP \leftarrow SP - 4$			
		\$!addr20	3	3	–	$(SP - 2) \leftarrow (PC + 3)_S, (SP - 3) \leftarrow (PC + 3)_H,$ $(SP - 4) \leftarrow (PC + 3)_L, PC \leftarrow PC + 3 +$ jdisp16, $SP \leftarrow SP - 4$			
		!addr16	3	3	–	$(SP - 2) \leftarrow (PC + 3)_S, (SP - 3) \leftarrow (PC + 3)_H,$ $(SP - 4) \leftarrow (PC + 3)_L, PC \leftarrow 0000, addr16,$ $SP \leftarrow SP - 4$			
		!!addr20	4	3	–	$(SP - 2) \leftarrow (PC + 4)_S, (SP - 3) \leftarrow (PC + 4)_H,$ $(SP - 4) \leftarrow (PC + 4)_L, PC \leftarrow addr20,$ $SP \leftarrow SP - 4$			
	CALLT	[addr5]	2	5	–	$(SP - 2) \leftarrow (PC + 2)_S, (SP - 3) \leftarrow (PC + 2)_H,$ $(SP - 4) \leftarrow (PC + 2)_L, PC_S \leftarrow 0000,$ $PC_H \leftarrow (0000, addr5 + 1),$ $PC_L \leftarrow (0000, addr5),$ $SP \leftarrow SP - 4$			
	BRK	–	2	5	–	$(SP - 1) \leftarrow PSW, (SP - 2) \leftarrow (PC + 2)_S,$ $(SP - 3) \leftarrow (PC + 2)_H, (SP - 4) \leftarrow (PC + 2)_L,$ $PC_S \leftarrow 0000,$ $PC_H \leftarrow (0007FH), PC_L \leftarrow (0007EH),$ $SP \leftarrow SP - 4, IE \leftarrow 0$			
	RET	–	1	6	–	$PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1),$ $PC_S \leftarrow (SP + 2), SP \leftarrow SP + 4$			
	RETI	–	2	6	–	$PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1),$ $PC_S \leftarrow (SP + 2), PSW \leftarrow (SP + 3),$ $SP \leftarrow SP + 4$	R	R	R
	RETB	–	2	6	–	$PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1),$ $PC_S \leftarrow (SP + 2), PSW \leftarrow (SP + 3),$ $SP \leftarrow SP + 4$	R	R	R

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $x2+3$ 。

表 26-5. 操作列表 (16/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
堆栈操作	PUSH	PSW	2	1	–	$(SP - 1) \leftarrow PSW, (SP - 2) \leftarrow 00H,$ $SP \leftarrow SP - 2$			
		rp	1	1	–	$(SP - 1) \leftarrow rp_H, (SP - 2) \leftarrow rp_L,$ $SP \leftarrow SP - 2$			
	POP	PSW	2	3	–	$PSW \leftarrow (SP + 1), SP \leftarrow SP + 2$	R	R	R
		rp	1	1	–	$rp_L \leftarrow (SP), rp_H \leftarrow (SP + 1), SP \leftarrow SP + 2$			
	MOVW	SP, #word	4	1	–	$SP \leftarrow word$			
		SP, AX	2	1	–	$SP \leftarrow AX$			
		AX, SP	2	1	–	$AX \leftarrow SP$			
		HL, SP	3	1	–	$HL \leftarrow SP$			
		BC, SP	3	1	–	$BC \leftarrow SP$			
		DE, SP	3	1	–	$DE \leftarrow SP$			
	ADDW	SP, #byte	2	1	–	$SP \leftarrow SP + byte$			
	SUBW	SP, #byte	2	1	–	$SP \leftarrow SP - byte$			
无条件转移	BR	AX	2	3	–	$PC \leftarrow CS, AX$			
		\$addr20	2	3	–	$PC \leftarrow PC + 2 + jdisp8$			
		\$!addr20	3	3	–	$PC \leftarrow PC + 3 + jdisp16$			
		!addr16	3	3	–	$PC \leftarrow 0000, addr16$			
		!!addr20	4	3	–	$PC \leftarrow addr20$			
条件转移	BC	\$addr20	2	$2/4^{注3}$	–	$PC \leftarrow PC + 2 + jdisp8$ 如果 $CY = 1$			
	BNC	\$addr20	2	$2/4^{注3}$	–	$PC \leftarrow PC + 2 + jdisp8$ 如果 $CY = 0$			
	BZ	\$addr20	2	$2/4^{注3}$	–	$PC \leftarrow PC + 2 + jdisp8$ 如果 $Z = 1$			
	BNZ	\$addr20	2	$2/4^{注3}$	–	$PC \leftarrow PC + 2 + jdisp8$ 如果 $Z = 0$			
	BH	\$addr20	3	$2/4^{注3}$	–	$PC \leftarrow PC + 3 + jdisp8$ 如果 $(Z \vee CY)=0$			
	BNH	\$addr20	3	$2/4^{注3}$	–	$PC \leftarrow PC + 3 + jdisp8$ 如果 $(Z \vee CY)=1$			
	BT	saddr.bit, \$addr20	4	$3/5^{注3}$	–	$PC \leftarrow PC + 4 + jdisp8$ 如果 $(saddr).bit = 1$			
		sfr.bit, \$addr20	4	$3/5^{注3}$	–	$PC \leftarrow PC + 4 + jdisp8$ 如果 $sfr.bit = 1$			
		A.bit, \$addr20	3	$3/5^{注3}$	–	$PC \leftarrow PC + 3 + jdisp8$ 如果 $A.bit = 1$			
		PSW.bit, \$addr20	4	$3/5^{注3}$	–	$PC \leftarrow PC + 4 + jdisp8$ 如果 $PSW.bit = 1$			
		[HL].bit, \$addr20	3	$3/5^{注3}$	6/8	$PC \leftarrow PC + 3 + jdisp8$ 如果 $(HL).bit = 1$			
		ES:[HL].bit, \$addr20	4	$4/6^{注3}$	7/9	$PC \leftarrow PC + 4 + jdisp8$ 如果 $(ES, HL).bit = 1$			

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. 这表示“不满足条件/满足条件”的时钟数。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $x2+3$ 。

表 26-5. 操作列表 (17/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
条件转移	BF	saddr.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 如果(saddr).bit = 0			
		sfr.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 如果 sfr.bit = 0			
		A.bit, \$addr20	3	3/5 ^{注3}	—	PC ← PC + 3 + jdisp8 如果 A.bit = 0			
		PSW.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 如果 PSW.bit = 0			
		[HL].bit, \$addr20	3	3/5 ^{注3}	6/8	PC ← PC + 3 + jdisp8 如果(HL).bit = 0			
		ES:[HL].bit, \$addr20	4	4/6 ^{注3}	7/9	PC ← PC + 4 + jdisp8 如果(ES, HL).bit = 0			
	BTCLR	saddr.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 如果(saddr).bit = 1 然后复位(saddr).bit			
		sfr.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 如果 sfr.bit = 1 然后复位 sfr.bit			
		A.bit, \$addr20	3	3/5 ^{注3}	—	PC ← PC + 3 + jdisp8 如果 A.bit = 1 然后复位 A.bit			
		PSW.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 如果 PSW.bit = 1 然后复位 PSW.bit	×	×	×
		[HL].bit, \$addr20	3	3/5 ^{注3}	—	PC ← PC + 3 + jdisp8 如果(HL).bit = 1 然后复位(HL).bit			
		ES:[HL].bit, \$addr20	4	4/6 ^{注3}	—	PC ← PC + 4 + jdisp8 如果(ES, HL).bit = 1 然后复位(ES, HL).bit			
条件跳转	SKC	—	2	1	—	如果 CY = 1 跳过下一条指令			
	SKNC	—	2	1	—	如果 CY = 0 跳过下一条指令			
	SKZ	—	2	1	—	如果 Z = 1 跳过下一条指令			
	SKNZ	—	2	1	—	如果 Z = 0 跳过下一条指令			
	SKH	—	2	1	—	如果(Z ∨ CY) = 0 跳过下一条指令			
	SKNH	—	2	1	—	如果(Z ∨ CY) = 1 跳过下一条指令			
CP 控制	SEL	Rb _n	2	1	—	RBS[1:0] ← n			
	NOP	—	1	1	—	无操作			
	EI	—	3	4	—	IE ← 1 (允许中断)			
	DI	—	3	4	—	IE ← 0 (禁止中断)			
	HALT	—	2	3	—	设置 HALT 模式			
	STOP	—	2	3	—	设置 STOP 模式			

- 注
1. 当内部 RAM 区域、SFR 区域或扩展 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. 这表示“不满足条件/满足条件”的时钟数。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器选择的一个周期的 CPU 时钟(f_{CLK})
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。当从内部 RAM 区域取到指令时，时钟数的最大值为时钟数 $\times 2 + 3$ 。
 3. n 表示寄存器 bank 的序号 (n = 0 至 3)。

第 27 章 电气特性（目标值）

注意事项 1. 这些规范所示为目标值，设备评估后可能改变。

2. 78K0R/KG3-L 具有片上调试功能，可用于开发和评价。在量产时不要使用产品的片上调试功能，因为 flash 存储器中保证的可重写次数可能会超出，从而不能保证产品的可靠性。对于因使用片上调试功能而产生的问题，NEC 电子不承担任何责任。

最大额定值 (TA = 25°C) (1/2)

参数	符号	条件	额定值	单位
供电电压	V _{DD}		−.5 至 +6.5	V
	EV _{DD0} , EV _{DD1}	EV _{DD0} = EV _{DD1}	−.5 至 +6.5	V
	V _{SS}		−.5 至 +.3	V
	EV _{SS0} , EV _{SS1}	EV _{SS0} = EV _{SS1}	−.5 至 +.3	V
	AV _{REF}		−.5 至 V _{DD} +0.3 ^{‡1}	V
	AV _{SS}		−.5 至 +.3	V
REGC 引脚输入电压	V _I REGC	REGC	−0.3 至 3.6 和 −0.3 至 V _{DD} +0.3 ^{‡2}	V
输入电压	V _{I1}	P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P64 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120 至 P124, P131, P140 至 P145, EXCLK, RESET, FLMD0	−0.3 至 EV _{DD0} , EV _{DD1} +0.3 和 −0.3 至 V _{DD} +0.3 ^{‡1}	V
	V _{I2}	P60 至 P63 (N 沟道漏极开路)	−.3 至 +6.5	V
	V _{I3}	P20 至 P27, P150 至 P157	−0.3 至 AV _{REF} +0.3 和 −0.3 至 V _{DD} +0.3 ^{‡1}	V
输出电压	V _{O1}	P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P60 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120, P130, P131, P140 至 P145	−0.3 至 EV _{DD0} , EV _{DD1} +0.3 ^{‡1}	V
	V _{O2}	P20 至 P27, P150 至 P157	−0.3 至 AV _{REF} +0.3	V

注 1. 必须小于等于 6.5 V。

2. 通过一个电容（0.47 至 1 μ F：目标值）连接 REGC 引脚和 V_{SS}。该值控制 REGC 引脚的绝对最大额定值。不要将此电压用于这个引脚。

注意事项 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。

备注 除非另外说明，复用功能引脚的特性与端口引脚相同。

最大额定值 (TA = 25°C) (2/2)

参数	符号	条件		额定值	单位
模拟 输入电压	V _{AN}	ANI0 至 ANI15		−0.3 至 AV _{REF} +0.3 ^注 和−0.3 至 V _{DD} +0.3 ^注	V
输出电流,高	I _{OH1}	每个引脚	P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P64 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120, P130, P131, P140 至 P145	−10	mA
		所有引脚总和 −80mA	P00 至 P04, P40 至 P47, P120, P130, P131, P140 至 P145	−25	mA
			P05, P06, P10 至 P17, P30, P31, P50 至 P57, P64 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111	−55	mA
	I _{OH2}	每个引脚	P20 至 P27, P150 至 P157	−0.5	mA
		所有引脚总和		−2	mA
	输出电流, 低	I _{OL1}	每个引脚	P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P60 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120, P130, P131, P140 至 P145	30
所有引脚总和 200 mA			P00 至 P04, P40 至 P47, P120, P130, P131, P140 至 P145	60	mA
			P05, P06, P10 至 P17, P30, P31, P50 至 P57, P60 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111	140	mA
I _{OL2}		每个引脚	P20 至 P27, P150 至 P157	1	mA
		所有引脚总和		5	mA
工作环境温度		T _A	在普通操作模式下		−40 至 +85
	在 Flash 存储器编程模式下				
存储温度	T _{stg}			−65 至 +150	°C

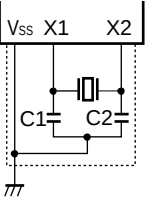
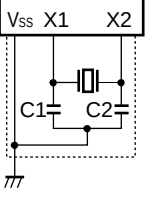
注 必须小于等于 6.5 V。

注意事项 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。

备注 除非另外说明，复用功能引脚的特性与端口引脚相同。

X1 振荡器特性

($T_A = -40$ 至 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

振荡器	建议使用的电路	参数	条件	最小值	典型值	最大值	单位
陶瓷振荡器		X1 时钟振荡频率 (fx) [※]	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		20.0	MHz
			$1.8\text{ V} \leq V_{DD} \leq 2.7\text{ V}$	2.0		5.0	MHz
晶体振荡器		X1 时钟振荡频率 (fx) [※]	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		20.0	MHz
			$1.8\text{ V} \leq V_{DD} \leq 2.7\text{ V}$	2.0		5.0	MHz

注 仅表示振荡器的特性。如需了解指令执行时间，可参见 AC 特性。

注意事项 1. 在使用 X1 时钟时，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接电容时产生不利影响。

- 连接线越短越好。
- 连接线不应与其它信号线交叉。
- 不要在流经较大变化电流的信号线周围布线。
- 始终保持振荡器电容器的接地点电压与 V_{SS} 相同。
- 不要将电容的地信号接入大电流地。
- 不要从振荡器获取信号。

2. 由于复位后 CPU 使用内部高速振荡时钟进行操作，因此用户要使用振荡稳定时间计数器的状态寄存器（OSTC）检测 X1 时钟振荡稳定时间。在充分评估了所使用的振荡器的振荡稳定时间后，再确定 OSTC 和振荡稳定时间选择寄存器（OSTS）的振荡稳定时间。

备注 对于振荡器选择和振荡常数，用户需要自己评价振荡特性或请振荡器厂商评价。

内部振荡器特性

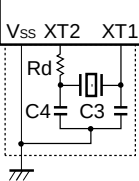
($T_A = -40$ 至 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

振荡器	参数	条件	最小值	典型值	最大值	单位
内部高速振荡时钟 频率*	f_{IH1M}			1.0		MHz
	f_{IH8M}			8.0		MHz
	f_{IH20M}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		20		MHz
内部低速振荡时钟 频率	f_{IL}			30		kHz

注 这只显示了振荡器特性。如需了解指令执行时间，可参见 AC 特性。

XT1 振荡器特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

振荡器	建议使用的电路	项目	条件	最小值	典型值	最大值	单位
晶体振荡器		XT1 时钟振荡频率 (f_{XT}) [※]		32	32.768	35	kHz

注 仅表示振荡器的特性。如需了解指令执行时间，可参见 AC 特性。

注意事项 1. 在使用 XT1 时钟时，上图中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接电容时产生不利影响。

- 连接线越短越好。
- 连接线不应与其它信号线交叉。
- 不要在流经较大变化电流的信号线周围布线。
- 始终保持振荡器电容器的接地点电压与 Vss 相同。
- 不要将电容的地信号接入大电流地。
- 不要从振荡器获取信号。

2. XT1 振荡器是一个低振幅电路，用于降低功耗，但比 X1 振荡器更容易受到噪音干扰。因此在使用 XT1 时钟时应特别注意连线方式。

备注 对于振荡器选择和振荡常数，用户需要自己评价振荡特性或请振荡器厂商评价。

DC 特性 (1/10)

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

项目	符号	条件		最小值	典型值	最大值	单位
输出电流, 高 ^{註1}	I _{OH1}	P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P64 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120, P130, P131, P140 至 P145 的每个引脚	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-3.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-1.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-1.0	mA
		P00 至 P04, P40 至 P47, P120, P130, P131, P140 至 P145 的总和 (当占空比 = 70% ^{註2})	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-20.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-10.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-5.0	mA
		P05, P06, P10 至 P17, P30, P31, P50 至 P57, P64 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111 的总和 (当占空比 = 70% ^{註2})	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-30.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-19.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-10.0	mA
		所有引脚总和 (当占空比 = 60% ^{註2}) (当占空比 = 70% ^{註2})	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-50.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-29.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-15.0	mA
	I _{OH2}	每个引脚对应 P20 至 P27, P150 至 P157	$AV_{REF} \leq V_{DD}$			-0.1	mA

注 1. 即使电流从 EV_{DD0} 或 EV_{DD1} 引脚流向输出引脚，设备工作时的电流值也可以得到保证。

2. 占空比为 60% 或 70% 的条件下指定。

可通过以下表达式计算已经改变占空比的输出电流值（当占空比从 70% 改变为 n%）。

- 引脚的总输出电流 = $(I_{OH} \times 0.7) / (n \times 0.01)$

<举例> 当 n = 50% 并且 I_{OH} = -20.0 mA

$$\text{引脚的总输出电流} = (-20.0 \times 0.7) / (50 \times 0.01) = -28.0\text{ mA}$$

但是，允许流入每个引脚的电流不会随占空比而变化。高于最大额定值的电流不能流入引脚。

注意事项 P02 至 P04, P10, P12, P40, P142 至 P144 在 N 沟道漏极开路模式下不输出高电平。

备注 除非另外说明，复用功能引脚的特性与端口引脚相同。

DC 特性 (2/10)

($T_A = -40$ 至 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

项目	符号	条件	最小值	典型值	最大值	单位
输出电流, 低 ^{注1}	I _{OL1}	P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P64 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120, P130, P131, P140 至 P145 的每个引脚	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		8.5	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		1.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		0.5	mA
		每个引脚对应 P60 至 P63	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		15.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		3.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		2.0	mA
		P00 至 P04, P40 至 P47, P120, P130, P131, P140 至 P145 的总和 (当占空比 = 70% ^{注2})	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		20.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		15.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		9.0	mA
		P05, P06, P10 至 P17, P30, P31, P50 至 P57, P60 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111 的总和 (当占空比 = 70% ^{注2})	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		45.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		35.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		20.0	mA
		所有引脚总和 (当占空比 = 70% ^{注2})	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		65.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		50.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		29.0	mA
	I _{OL2}	每个引脚对应 P20 至 P27, P150 至 P157	$AV_{REF} \leq V_{DD}$		0.4	mA

- 注
- 即使电流从输出引脚流向 EV_{SS0} , EV_{SS1} , V_{SS} 引脚和 AV_{SS} 引脚, 器件工作时的电流值也可以得到保证。
 - 占空比为 70%的条件下指定。
可通过以下表达式计算已经改变占空比的输出电流值（当占空比从 70%改变为 n%）。
 - 引脚的总输出电流 = $(I_{OL} \times 0.7)/(n \times 0.01)$
 <举例> $n = 50\%$ 并且 $I_{OL} = 20.0\text{ mA}$
 引脚的总输出电流 = $(20.0 \times 0.7)/(50 \times 0.01) = 28.0\text{ mA}$
 但是, 允许流入每个引脚的电流不会随占空比而变化。高于最大额定值的电流不能流入引脚。

备注 除非另外说明, 复用功能引脚的特性与端口引脚相同。

DC 特性 (3/10)

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

项目	符号	条件	最小值	典型值	最大值	单位
输入电压, 高	V_{IH1}	P01, P02, P12, P13, P15, P41, P45, P52, P56, P57, P80 至 P87, P91, P111, P121 至 P124, P144	$0.7V_{DD}$		V_{DD}	V
	V_{IH2}	P00, P03 至 P06, P10, P11, P14, P16, P17, P30, P31, P40, P42 至 P44, P46, P47, P50, P51, P53 至 P55, P64 至 P67, P70 至 P77, P110, P120, P131, P140 至 P143, P145, EXCLK, RESET	普通输入缓冲器 $0.8V_{DD}$		V_{DD}	V
	V_{IH3}	P03, P04, P10, P11, P142, P143	TTL 输入缓冲器 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.2	V_{DD}	V
			TTL 输入缓冲器 $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	2.0	V_{DD}	V
			TTL 输入缓冲器 $1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	1.6	V_{DD}	V
	V_{IH4}	P20 至 P27, P150 至 P157	$2.7\text{ V} \leq AV_{REF} \leq V_{DD}$ $AV_{REF} = V_{DD} < 2.7\text{ V}$	$0.7AV_{REF}$	AV_{REF}	V
	V_{IH5}	P60 至 P63	$0.7V_{DD}$		6.0	V
	V_{IH6}	FLMD0	$0.9V_{DD}$ 注		V_{DD}	V

注 当用在 Flash 存储器编程模式下时, 必须大于等于 $0.9V_{DD}$ 。

- 注意事项**
1. 即使在 N 沟道漏极开路模式下, 引脚 P02 至 P04, P10, P12, P40 以及 P142 至 P144 的 V_{IH} 的最大值也是 V_{DD} 。
 2. 对于 P122/EXCLK, V_{IH} 和 V_{IL} 的值根据输入端口模式或外部时钟模式而不同。
确保满足在外部时钟输入模式下的 EXCLK 的 DC 特性。

备注 除非另外说明, 复用功能引脚的特性与端口引脚相同。

DC 特性 (4/10)

($T_A = -40$ 至 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

项目	符号	条件	最小值	典型值	最大值	单位
输入电压, 低	V_{IL1}	P01, P02, P12, P13, P15, P41, P45, P52, P56, P57, P80 至 P87, P91, P111, P121 至 P124, P144	0		$0.3V_{DD}$	V
	V_{IL2}	P00, P03 至 P06, P10, P11, P14, P16, P17, P30, P31, P40, P42 至 P44, P46, P47, P50, P51, P53 至 P55, P64 至 P67, P70 至 P77, P110, P120, P131, P140 至 P143, P145, EXCLK, RESET	0		$0.2V_{DD}$	V
	V_{IL3}	P03, P04, P10, P11, P142, P143	TTL 输入缓冲器 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0	0.8	V
			TTL 输入缓冲器 $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	0	0.5	V
			TTL 输入缓冲器 $1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0	0.2	V
	V_{IL4}	P20 至 P27, P150 至 P157	$2.7\text{ V} \leq AV_{REF} \leq V_{DD}$	0	$0.3AV_{REF}$	V
			$AV_{REF} = V_{DD} < 2.7\text{ V}$			
	V_{IL5}	P60 至 P63	0		$0.3V_{DD}$	V
	V_{IL6}	FLMD0 [※]	0		$0.1V_{DD}$	V

注 当禁止写入 flash 存储器时, 将 FLMD0 引脚直接连接到 V_{SS} , 并保持一个小于 $0.1V_{DD}$ 的电压

注意事项 对于 P122/EXCLK, V_{IH} 和 V_{IL} 的值根据输入端口模式或外部时钟模式而不同。
确保满足在外部时钟输入模式下的 EXCLK 的 DC 特性。

备注 除非另外说明, 复用功能引脚的特性与端口引脚相同。

DC 特性 (5/10)

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

项目	符号	条件	最小值	典型值	最大值	单位
输出电压, 高	VOH1	P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P64 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120, P130, P131, P140 至 P145	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OH1} = -3.0\text{ mA}$	$V_{DD} - .7$		V
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OH1} = -1.0\text{ mA}$	$V_{DD} - .5$		V
	VOH2	P20 至 P27, P150 至 P157	$AV_{REF} \leq V_{DD}$, $I_{OH2} = -0.1\text{ mA}$	$AV_{REF} - 0.5$		V
输出电压, 低	VOL1	P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P64 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120, P130, P131, P140 至 P145	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 8.5\text{ mA}$		0.7	V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 1.0\text{ mA}$		0.5	V
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 0.5\text{ mA}$		0.4	V
	VOL2	P20 至 P27, P150 至 P157	$AV_{REF} = V_{DD}$, $I_{OL2} = 0.4\text{ mA}$		0.4	V
	VOL3	P60 至 P63	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 15.0\text{ mA}$		2.0	V
			$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 5.0\text{ mA}$		0.4	V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 3.0\text{ mA}$		0.4	V
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 2.0\text{ mA}$		0.4	V

备注 除非另外说明, 复用功能引脚的特性与端口引脚相同。

DC 特性 (6/10)

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

项目	符号	条件	最小值	典型值	最大值	单位
输入漏电流, 高	I_{LH1}	P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P60 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120, P131, P140 至 P145, FLMD0, RESET	$V_I = V_{DD}$		1	μA
	I_{LH2}	P20 至 P27, P150 至 P157	$V_I = AV_{REF}$ $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$		1	μA
			$V_I = AV_{REF}$ $AV_{REF} = V_{DD} < 2.7\text{ V}$			
	I_{LH3}	P121 至 P124 (X1, X2, XT1, XT2)	$V_I = V_{DD}$	在输入端口	1	μA
				在振荡器连接	10	μA
输入漏电流, 低	I_{LIL1}	P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P60 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120, P131, P140 至 P145, FLMD0, RESET	$V_I = V_{SS}$		-1	μA
	I_{LIL2}	P20 至 P27, P150 至 P157	$V_I = V_{SS}$ $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$		-1	μA
			$V_I = V_{SS}$ $AV_{REF} = V_{DD} < 2.7\text{ V}$			
	I_{LIL3}	P121 至 P124 (X1, X2, XT1, XT2)	$V_I = V_{SS}$	在输入端口	-1	μA
				在振荡器连接	-10	μA

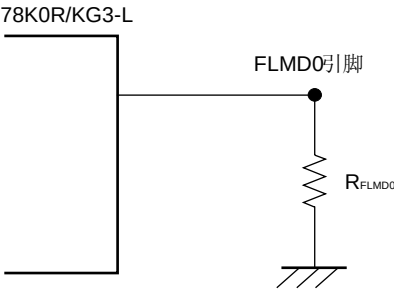
备注 除非另外说明，复用功能引脚的特性与端口引脚相同。

DC 特性 (7/10)

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

项目	符号	条件		最小值	典型值	最大值	单位
片内上拉电阻	R_U	P00 至 P06, P10 至 P17, P30, P31, P40 至 P47, P50 至 P57, P64 至 P67, P70 至 P77, P80 至 P87, P91, P110, P111, P120, P131, P140 至 P145	$V_I = V_{SS}$, 在输入端口	10	20	100	$\text{k}\Omega$
FLMD0 引脚外部下拉电阻 ^注	R_{FLMD0}	当使用软件允许自编程模式设置时		100			$\text{k}\Omega$

注 建议 FLMD0 引脚悬空。如果引脚需要外部下拉，设置 R_{FLMD0} 为 $100\text{ k}\Omega$ 或以上。



备注 除非另外说明，复用功能引脚的特性与端口引脚相同。

DC 特性 (8/10)

($T_A = -40$ 至 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

参数	符号	条件		最小值	典型值	最大值	单位	
供电电流	I _{DD1} ^{註 1}	操作模式	f _{MX} = 20 MHz ^{註 2} , V _{DD} = 5.0 V	方波输入		6.5	T.B.D	mA
			振荡器连接		6.8	T.B.D	mA	
			f _{MX} = 20 MHz ^{註 2} , V _{DD} = 3.0 V	方波输入		6.5	T.B.D	mA
			振荡器连接		6.8	T.B.D	mA	
			f _{MX} = 10 MHz ^{註 2, 3} , V _{DD} = 5.0 V	方波输入		3.5	T.B.D	mA
			振荡器连接		3.6	T.B.D	mA	
			f _{MX} = 10 MHz ^{註 2, 3} , V _{DD} = 3.0 V	方波输入		3.5	T.B.D	mA
			振荡器连接		3.6	T.B.D	mA	
			f _{MX} = 5 MHz ^{註 2, 3} , V _{DD} = 3.0 V	方波输入		1.9	T.B.D	mA
			振荡器连接		2.0	T.B.D	mA	
			f _{MX} = 5 MHz ^{註 2, 3} , V _{DD} = 2.0 V	方波输入		1.5	T.B.D	mA
			振荡器连接		1.5	T.B.D	mA	
			f _{IH20} = 20 MHz ^{註 4}	V _{DD} = 5.0 V		6.8	T.B.D	mA
			V _{DD} = 3.0 V		6.8	T.B.D	mA	
			f _{IH} = 8 MHz ^{註 4}	V _{DD} = 5.0 V		2.8	T.B.D	mA
			V _{DD} = 3.0 V		2.8	T.B.D	mA	
			f _{IH} = 1 MHz ^{註 4}	V _{DD} = 3.0 V		210	T.B.D	μA
			f _{SUB} = 32.768 kHz ^{註 5} , T _A = -40 至 +70°C	V _{DD} = 5.0 V		5.0	T.B.D	μA
			V _{DD} = 3.0 V		5.0	T.B.D	μA	
			V _{DD} = 2.0 V		5.0	T.B.D	μA	
			f _{SUB} = 32.768 kHz ^{註 5} , T _A = -40 至 +85°C	V _{DD} = 5.0 V		5.0	T.B.D	μA
			V _{DD} = 3.0 V		5.0	T.B.D	μA	
			V _{DD} = 2.0 V		5.0	T.B.D	μA	

- 注
1. 当输入引脚电平固定在 V_{DD} 或 V_{SS} 时的流入 V_{DD} , EV_{DD0} , EV_{DD1} 和 AV_{REF} 1 的总电流，包括输入漏电流。最大值栏位下的数值包括外设工作电流。但是不包括流过 A/D 转换器、LVI 电路、I/O 端口和片内上拉/下拉电阻的电流。
 2. 当内部高速振荡器、20MHz 内部高速振荡器和副系统时钟停止时。
 3. 当 AMPH（时钟操作模式控制寄存器 (CMC) 的第 0 位）= 0、FLPC 与 FSEL（操作速度模式控制寄存器 (OSMC) 的第 1 和 0 位）都为 0 时。
 4. 当高速系统时钟和副系统时钟停止时。
 5. 当内部高速振荡器，20MHz 内部高速振荡器和高速副系统时钟停止时。
停止 当看门狗定时器停止时。

备注

f_{MX} : 高速系统时钟频率（X1 时钟振荡频率或外部主系统时钟频率）

f_{IH20} : 20 MHz 内部高速振荡时钟频率

f_{IH} : 内部高速振荡时钟频率

f_{SUB} : 副系统时钟频率（XT1 时钟振荡频率）

DC 特性 (9/10)

($T_A = -40$ 至 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

参数	符号	条件		最小值	典型值	最大值	单位	
供电电流	I _{DD2} ^{註 1}	HALT 模式	f _{MX} = 20 MHz ^{註 2} , V _{DD} = 5.0 V	方波输入		1.1	T.B.D	mA
				振荡器连接		1.4	T.B.D	mA
			f _{MX} = 20 MHz ^{註 2} , V _{DD} = 3.0 V	方波输入		1.1	T.B.D	mA
				振荡器连接		1.4	T.B.D	mA
			f _{MX} = 10 MHz ^{註 2, 3} , V _{DD} = 5.0 V	方波输入		0.65	T.B.D	mA
				振荡器连接		0.75	T.B.D	mA
			f _{MX} = 10 MHz ^{註 2, 3} , V _{DD} = 3.0 V	方波输入		0.65	T.B.D	mA
				振荡器连接		0.75	T.B.D	mA
			f _{MX} = 5 MHz ^{註 2, 3} , V _{DD} = 3.0 V	方波输入		0.39	T.B.D	mA
				振荡器连接		0.44	T.B.D	mA
			f _{MX} = 5 MHz ^{註 2, 3} , V _{DD} = 2.0 V	方波输入		0.3	T.B.D	mA
				振荡器连接		0.35	T.B.D	mA
			f _{IH20} = 20 MHz ^{註 4}	V _{DD} = 5.0 V		1.4	T.B.D	mA
				V _{DD} = 3.0 V		1.4	T.B.D	mA
			f _{IH} = 8 MHz ^{註 4}	V _{DD} = 5.0 V		0.45	T.B.D	mA
				V _{DD} = 3.0 V		0.45	T.B.D	mA
			f _{IH} = 1 MHz ^{註 4}	V _{DD} = 3.0 V		65	T.B.D	μA
			f _{SUB} = 32.768 kHz ^{註 5} , T _A = -40 至 +70°C	V _{DD} = 5.0 V		1.0	T.B.D	μA
				V _{DD} = 3.0 V		1.0	T.B.D	μA
				V _{DD} = 2.0 V		1.0	T.B.D	μA
	f _{SUB} = 32.768 kHz ^{註 5} , T _A = -40 至 +85°C	V _{DD} = 5.0 V		1.0	T.B.D	μA		
		V _{DD} = 3.0 V		1.0	T.B.D	μA		
		V _{DD} = 2.0 V		1.0	T.B.D	μA		
I _{DD3} ^{註 6}	STOP 模式	T _A = -40 至 +70°C			0.3	T.B.D	μA	
		T _A = -40 至 +85°C			0.3	T.B.D	μA	

- 注
1. 当输入引脚电平固定在 V_{DD} 或 V_{SS} 时的流入 V_{DD} , EV_{DD0} , EV_{DD1} 和 AV_{REF} 的总电流, 包括输入漏电流。最大值包括外围设备工作电流。但是不包括流过 A/D 转换器、LVI 电路、I/O 端口和片内上拉/下拉电阻的电流。在通过 flash 存储器执行 HALT 指令期间。
 2. 当内部高速振荡器、20MHz 内部高速振荡器和副系统时钟停止时。
 3. 当 AMPH (时钟操作模式控制寄存器 (CMC) 的第 0 位) = 0、FLPC 与 FSEL (操作速度模式控制寄存器 (OSMC) 的第 1 和 0 位) 都为 0 时。
 4. 当高速系统时钟和副系统时钟停止时。
 5. 当内部高速振荡器, 20MHz 内部高速振荡器和高速副系统时钟停止时。当看门狗定时器停止时。当实时计数器操作时。
 6. 当输入引脚电平固定在 V_{DD} 或 V_{SS} 时的流入 V_{DD} , EV_{DD0} , EV_{DD1} 和 AV_{REF} 的总电流, 包括输入漏电流。最大值包括外围设备工作电流。但是不包括流过 A/D 转换器、LVI 电路、I/O 端口和片内上拉/下拉电阻的电流。当副系统时钟停止时。当看门狗定时器停止时。

备注

f_{MX} : 高速系统时钟频率 (X1 时钟振荡频率或外部主系统时钟频率)

f_{IH20} : 20 MHz 内部高速振荡时钟频率

f_{IH} : 内部高速振荡时钟频率

f_{SUB} : 副系统时钟频率 (XT1 时钟振荡频率)

DC 特性 (10/10)

($T_A = -40$ 至 $+85^\circ\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

参数	符号	条件			最小值	典型值	最大值	单位
RTC 工作电流	I_{RTC} ^{注 1, 2}	$f_{SUB} = 32.768\text{ kHz}$			$V_{DD} = 3.0\text{ V}$	0.2	1.0	μA
					$V_{DD} = 2.0\text{ V}$	0.2	1.0	
看门狗定时器工作电流	I_{WDT} ^{注 2, 3}	$f_{IL} = 30\text{ kHz}$				0.4	T.B.D	μA
A/D 转换器工作电流	I_{ADC} ^{注 4}	以最大速度转换期间	高速模式 1	$AV_{REF} = V_{DD} = 5.0\text{ V}$		1.72	3.2	mA
			高速模式 2	$AV_{REF} = V_{DD} = 3.0\text{ V}$		0.72	1.6	mA
			普通模式	$AV_{REF} = V_{DD} = 5.0\text{ V}$		0.86	1.9	mA
			升压模式	$AV_{REF} = V_{DD} = 3.0\text{ V}$		0.37	0.8	mA
LVI 工作电流	I_{LVI} ^{注 5}					9	18	μA

- 注**
1. 电流只经过实时计数器（XT1 振荡器的工作电流除外）。当实时计数器操作在操作模式或 HALT 模式下，78K0R/ KG3-L 的电流值的典型值，是 I_{DD1} 或 I_{DD2} 的典型值与 I_{RTC} 的典型值的和。 I_{DD1} 和 I_{DD2} 最大值也包括实时计数器工作电流。当实时计数器运行在 $f_{CLK} = f_{SUB}/2$ 期间， I_{DD2} 的典型值包括实时计数器工作电流。
 2. 当内部高速振荡器和高速系统时钟停止时。
 3. 电流只经过看门狗定时器（包括 30 kHz 内部振荡器的工作电流）。当看门狗定时器操作在 $f_{CLK} = f_{SUB}/2$ 或 STOP 模式时，78K0R/KG3-L 的电流值是 I_{DD1} 、 I_{DD2} 或 I_{DD3} 和 I_{WDT} 的和。
 4. 电流只流经 A/D 转换器（ AV_{REF} 引脚）。当 A/D 转换器在操作模式或 HALT 模式时 78K0R/KG3-L 的电流值是 I_{DD1} 或 I_{DD2} 和 I_{ADC} 的总和。
 5. 电流只经过 LVI 电路。当 LVI 电路在操作、HALT 或 STOP 模式时，78K0R/KG3-L 的电流值是 I_{DD1} 、 I_{DD2} 或 I_{DD3} 、 I_{LVI} 的总和。

备注

f_{IL} : 内部低速振荡时钟频率

f_{SUB} : 副系统时钟频率（XT1 时钟振荡频率）

f_{CLK} : CPU/外围硬件时钟频率

AC 特性

(1) 基本操作(1/6)

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $1.8\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

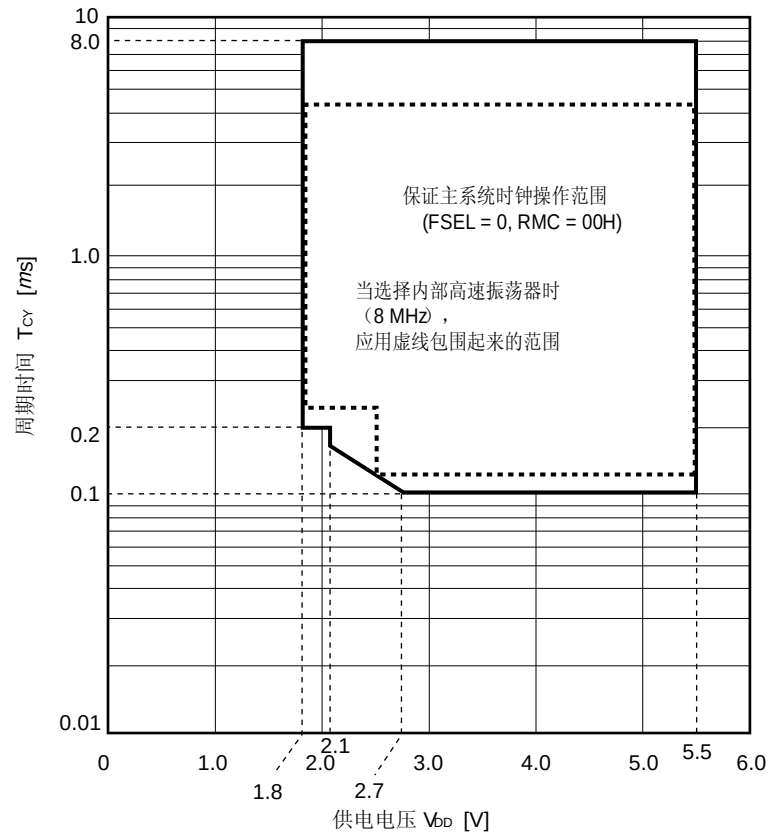
项目	符号	条件			最小值	典型值	最大值	单位
指令周期 (指令最短执行时间)	T _{CY}	主系统时钟 (f _{MAIN})操作	普通	2.7V≤V _{DD} ≤5.5V	0.05		8	μs
			电流模式	1.8V≤V _{DD} <2.7V	0.2		8	μs
			低功耗电流模式		1		8	μs
		副系统时钟 (f _{SUB}) 操作			57.2	61	62.5	μs
		在自编程模式下	普通电流模式	2.7V≤V _{DD} ≤5.5V	0.05		0.5	μs
				1.8V≤V _{DD} <2.7V	0.2		0.5	μs
外部主系统时钟频率	f _{EX}	2.7 V ≤ V _{DD} ≤ 5.5 V			2.0		20.0	MHz
		1.8 V ≤ V _{DD} < 2.7 V			2.0		5.0	MHz
外部主系统时钟输入高电平宽度, 低电平宽度	t _{EXH} , t _{EXL}	2.7 V ≤ V _{DD} ≤ 5.5 V			24			ns
		1.8 V ≤ V _{DD} < 2.7 V			96			ns
TI00 至 TI07 和 TI10 至 TI13 输入高电平宽度, 低电平宽度	t _{TIH} , t _{TIL}				2/f _{MCK} +10			ns
TO00 至 TO07 和 TO10 至 TO13 输出频率	f _{TO}	2.7 V ≤ V _{DD} ≤ 5.5 V					10	MHz
		1.8 V ≤ V _{DD} < 2.7 V					5	MHz
PCLBUZ0, PCLBUZ1 输出频率	f _{PCL}	2.7 V ≤ V _{DD} ≤ 5.5 V					10	MHz
		1.8 V ≤ V _{DD} < 2.7 V					5	MHz
中断输入高电平宽度, 低电平宽度	t _{INTH} , t _{INTL}				1			μs
按键中断输入低电平宽度	t _{KR}				250			ns
RESET _‾ 低电平宽度	t _{RSL}				10			μs

备注

1. f_{MCK} : 定时器阵列单元工作时钟频率
(由 TMRmn 寄存器的 CKSmn 位设置工作时钟。m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 7))
2. 关于稳压器的正常电流模式和低功耗电流模式的详细信息, 请参考第 21 章 稳压器。

(1) 基本操作(2/6)

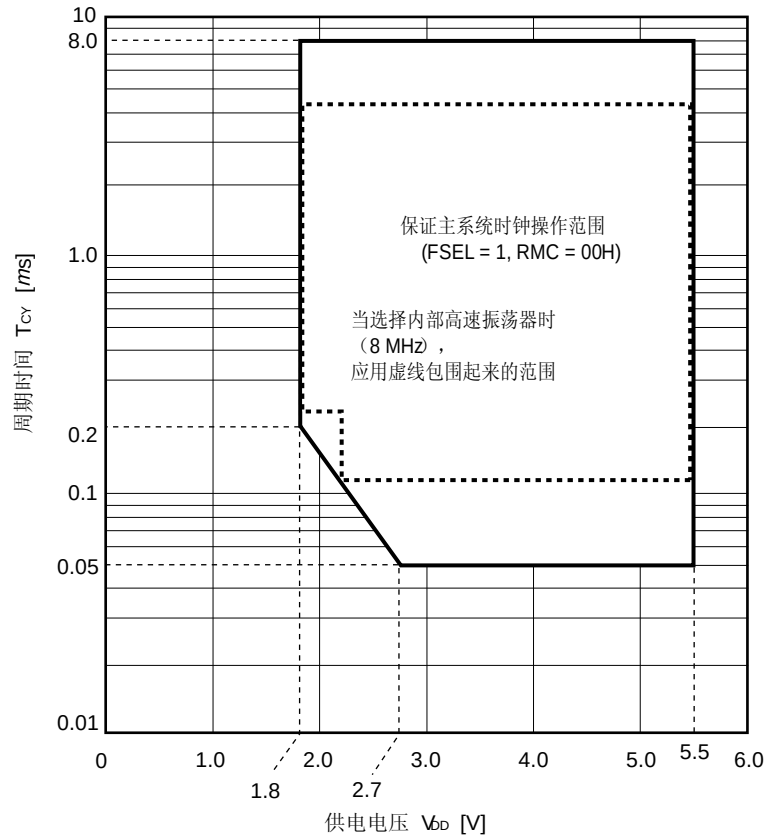
在主系统时钟工作期间的指令最短执行时间 (FSEL = 0, RMC = 00H)



备注 FSEL: 工作速度模式控制寄存器 (OSMC) 的位 0。
RMC: 稳压器模式控制寄存器

(1) 基本操作(3/6)

在主系统时钟工作期间的指令最短执行时间 (FSEL = 1, RMC = 00H)

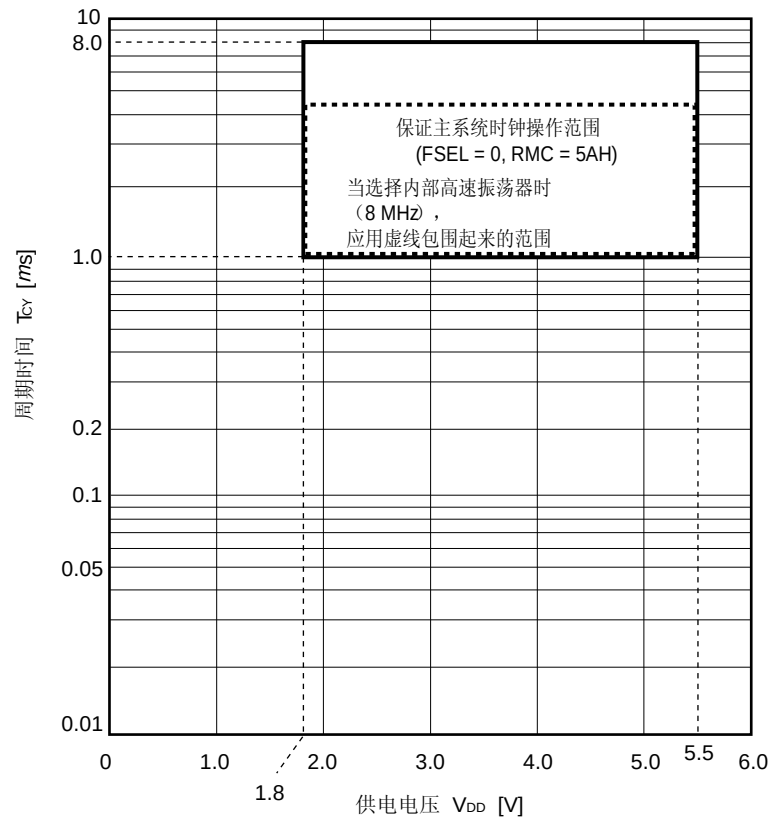


注意事项 当 $V_{DD} \leq 2.7$ V 时, 设置 FSEL = 0 转移到 STOP 模式。
当时钟工作在 10 MHz 或更低时, 设置 FSEL = 0。

备注 FSEL: 工作速度模式控制寄存器 (OSMC) 的位 0。
RMC: 稳压器模式控制寄存器

(1) 基本操作(4/6)

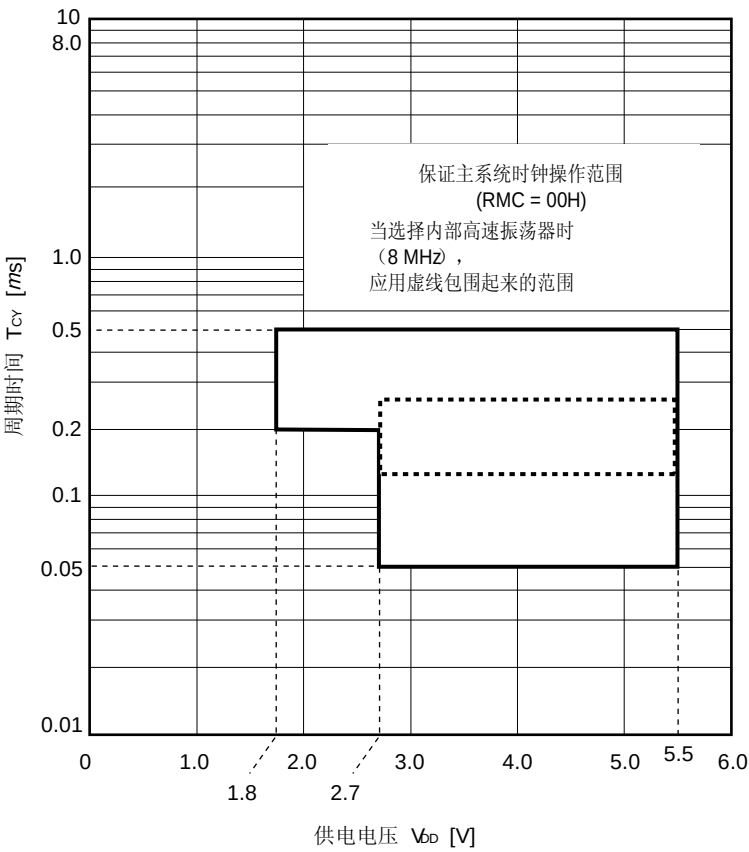
在主系统时钟工作期间的指令最短执行时间 (FSEL = 0, RMC = 5AH)



- 备注
1. FSEL: 工作速度模式控制寄存器 (OSMC) 的位 0。
RMC: 稳压器模式控制寄存器
 2. 当 RMC 设置为 5AH 时, 在整个电压范围是 1MHz (最大值)。

(1) 基本操作(5/6)

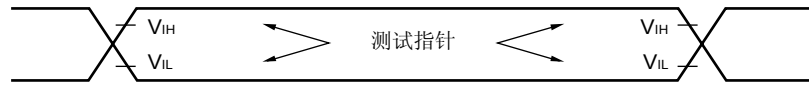
在自编程模式下的指令最短执行时间 (RMC = 00H)



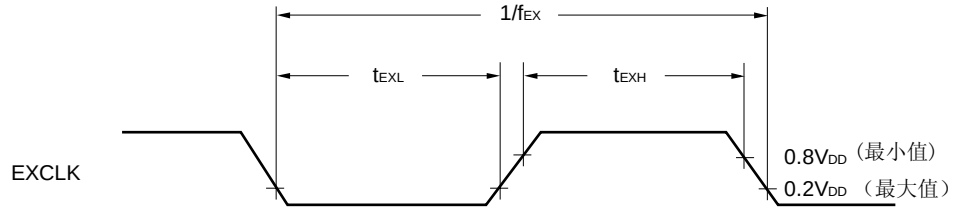
- 备注
- 1. RMC: 稳压器模式控制寄存器
 - 2. 当 RMC 设置为 5AH 或者 CPU 使用副系统时钟时，自编程功能无法使用。

(1) 基本操作(6/6)

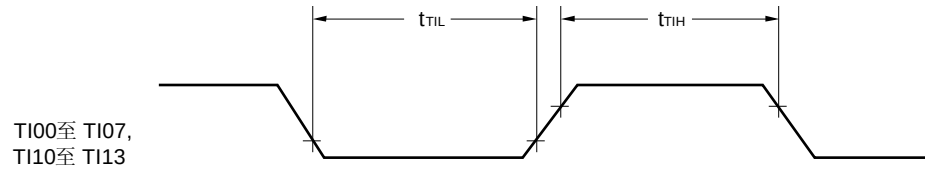
AC 时序测试点 (不包括外部总线接口)



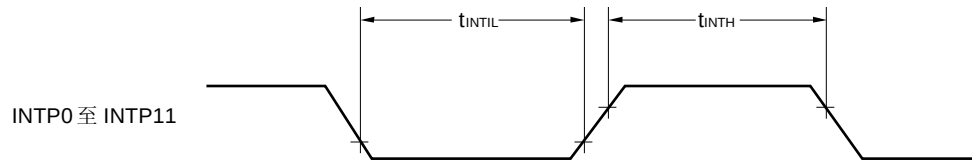
外部主系统时钟时序



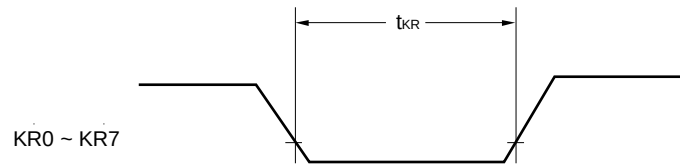
TI 时序



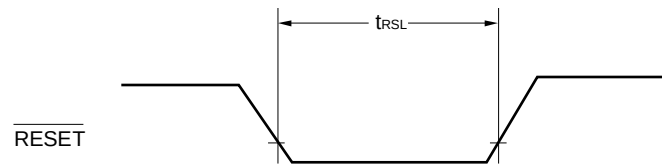
中断请求输入时序



按键中断输入时序



$\overline{RESET}$ 输入时序



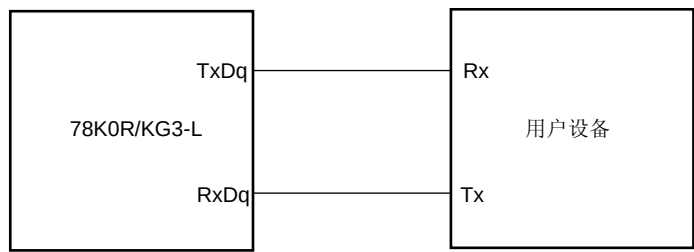
(2) 串行接口：串行阵列单元(1/17)

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

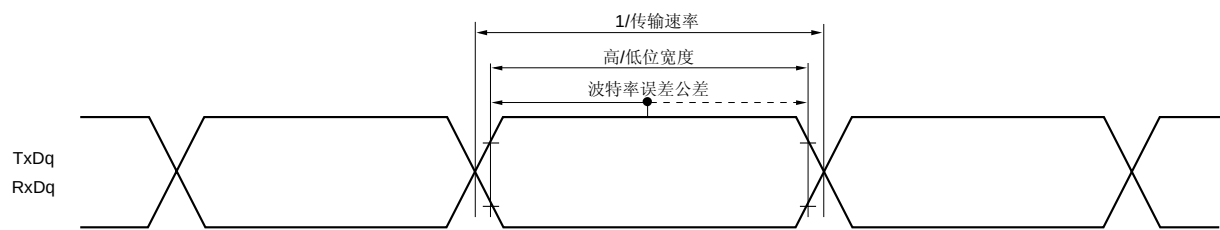
(a) 相同电平下通信 (UART 模式) (专用波特率发生器输出)

参数	符号	条件	最小值	典型值	最大值	单位
传输速率					$f_{MCK}/6$	bps
		$f_{CLK} = 20\text{ MHz}$, $f_{MCK} = f_{CLK}$			3.3	Mbps

UART 模式连接图 (相同电平下通信)



UART 模式位宽度 (相同电平下通信) (参考)



注意事项 使用 PIMg 和 POMg 寄存器，为 RxDq 选择正常输入缓冲器、为 TxDq 选择正常输出模式。

- 备注
- 1. q: UART 编号(q =0 至 3), g: PIM 和 POM 编号(g = 0, 1, 14)
 - 2. f_{MCK} : 串行阵列单元工作时钟频率
(由 SMRmn 寄存器的 CKSmn 位设置工作时钟。m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3))

(2) 串行接口: 串行阵列单元(2/17)

(T_A = -40 至 +85°C, 1.8 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)(b) 相同电平下通信 (CSI 模式) (主设备模式、 $\overline{\text{SCKp}}$... 内部时钟输出)

参数	符号	条件	最小值	典型值	最大值	单位
$\overline{\text{SCKp}}$ 周期时间	t_{KCY1}	4.0 V ≤ V _{DD} ≤ 5.5 V	200			ns
		2.7 V ≤ V _{DD} < 4.0 V	400			ns
		1.8 V ≤ V _{DD} < 2.7 V	800			ns
$\overline{\text{SCKp}}$ 高/低电平宽度	$t_{\text{KH1}},$ t_{KL1}	4.0 V ≤ V _{DD} ≤ 5.5 V	$t_{\text{KCY1}}/2 - 20$			ns
		2.7 V ≤ V _{DD} < 4.0 V	$t_{\text{KCY1}}/2 - 35$			ns
		1.8 V ≤ V _{DD} < 2.7 V	$t_{\text{KCY1}}/2 - 80$			ns
SI 设置时间 (到 $\overline{\text{SCKp}}\uparrow$) ^{注 1}	t_{SIK1}	4.0 V ≤ V _{DD} ≤ 5.5 V	70			ns
		1.8 V ≤ V _{DD} < 2.7 V	100			ns
		1.8 V ≤ V _{DD} < 2.7 V	190			ns
Slp 保持时间 (从 $\overline{\text{SCKp}}\uparrow$) ^{注 1}	t_{KSI1}		30			ns
从 $\overline{\text{SCKp}}\downarrow$ 到 SOp 输出的延迟时间 ^{注 2}	t_{KSO1}	C = 50 pF ^{注 3}			40	ns

- 注
1. 当 DAPmn = 0 和 CKPmn = 0, 或 DAPmn = 1 和 CKPmn = 1 时。Slp 设置时间变为“至 $\overline{\text{SCKp}}\downarrow$ ”, 当 DAPmn = 0 和 CKPmn = 1, 或 DAPmn = 1 和 CKPmn = 0。
 2. 当 DAPmn = 0 和 CKPmn = 0, 或 DAPmn = 1 和 CKPmn = 1 时。Slp 保持时间变为“从 $\overline{\text{SCKp}}\downarrow$ ”, 当 DAPmn = 0 和 CKPmn = 1, 或 DAPmn = 1 和 CKPmn = 0。
 3. C 是 $\overline{\text{SCKp}}$ 和 SOp 输出线的负载电容。

注意事项 使用 PIMg 和 POMg 寄存器, 为 Slp 选择正常输入缓冲器、为 SOp 和 $\overline{\text{SCKp}}$ 选择正常输出模式。

- 备注
1. p: CSI 编号(p = 00, 01, 10, 20), g: PIM 和 POM 编号 (g = 0, 1, 14)
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2))

(2) 串行接口: 串行阵列单元(3/17)

(TA = -40 至 +85°C, 1.8 V ≤ VDD = EVDD0 = EVDD1 ≤ 5.5 V, VSS = EVSS0 = EVSS1 = AVSS = 0 V)

(c) 相同电平下通信 (CSI 模式) (从设备模式、 $\overline{\text{SCKp}}$... 外部时钟输入)中

参数	符号	条件	最小值	典型值	最大值	单位
$\overline{\text{SCKp}}$ 周期时间	t_{KCY2}	16 MHz < f_{MCK}	8/ f_{MCK}			ns
		$f_{\text{MCK}} \leq 16 \text{ MHz}$	6/ f_{MCK}			ns
$\overline{\text{SCKp}}$ 高/低电平宽度	$t_{\text{KH2}}, t_{\text{KL2}}$		$f_{\text{KCY2}}/2$			ns
Slp 设置时间 (到 $\overline{\text{SCKp}}\uparrow$) ^{注 1}	t_{SIK2}		1/ $f_{\text{MCK}}+80$			ns
Slp 保持时间 (从 $\overline{\text{SCKp}}\uparrow$) ^{注 1}	t_{SI2}		50			ns
从 $\overline{\text{SCKp}}\downarrow$ 到 SOp 输出的延迟时间 ^{注 2}	t_{KSO2}	C = 50 pF ^{注 3}	4.0 V ≤ VDD ≤ 5.5 V		1/ $f_{\text{MCK}}+120$	ns
			2.7 V ≤ VDD < 4.0 V		1/ $f_{\text{MCK}}+120$	ns
			1.8 V ≤ VDD < 2.7 V		1/ $f_{\text{MCK}}+180$	ns

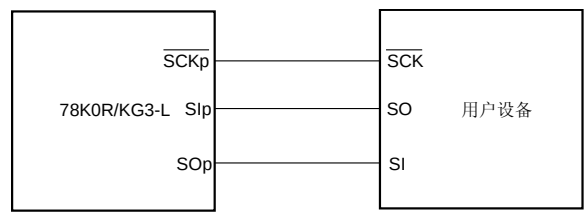
- 注
1. 当 DAPmn = 0 和 CKPmn = 0, 或 DAPmn = 1 和 CKPmn = 1 时。Slp 设置时间变为“至 $\overline{\text{SCKp}}\downarrow$ ”, 当 DAPmn = 0 和 CKPmn = 1, 或 DAPmn = 1 和 CKPmn = 0。
 2. 当 DAPmn = 0 和 CKPmn = 0, 或 DAPmn = 1 和 CKPmn = 1 时。到 SOp 输出的延迟时间变为“从 $\overline{\text{SCKp}}\uparrow$ ”, 当 DAPmn = 0 和 CKPmn = 1, 或 DAPmn = 1 和 CKPmn = 0。
 3. C 是 $\overline{\text{SCKp}}$ 和 SOp 输出线的负载电容。

注意事项 使用 PIMg 和 POMg 寄存器, 为 Slp 和 $\overline{\text{SCKp}}$ 选择正常输入缓冲器、为 SOp 选择正常输出模式。

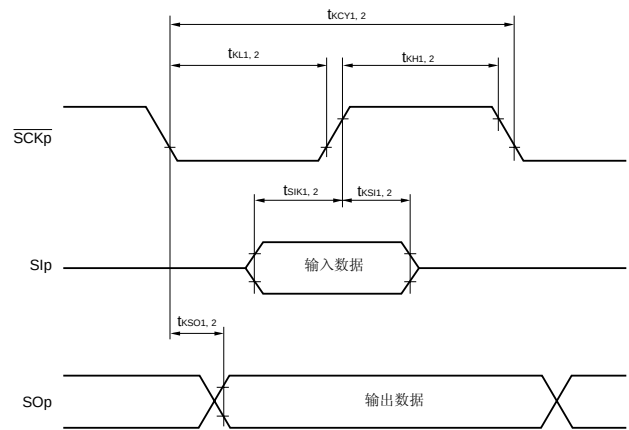
- 备注
1. p: CSI 编号(p = 00, 01, 10, 20), g: PIM 和 POM 编号 (g = 0, 1, 14)
 2. f_{MCK} : 串行阵列单元工作时钟频率
(由 SMRmn 寄存器的 CKSmn 位设置工作时钟。m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2))

(2) 串行接口：串行阵列单元(4/17)

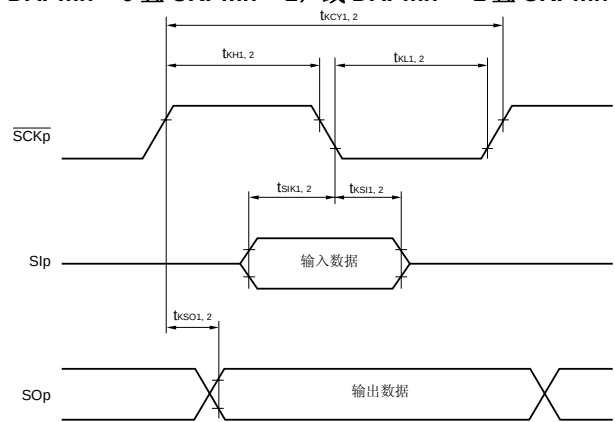
CSI 模式连接图 (相同电平下通信)



CSI 模式串行传输时序 (相同电平下通信)
(当 $\text{DAPmn} = 0$ 且 $\text{CKPmn} = 0$, 或 $\text{DAPmn} = 1$ 且 $\text{CKPmn} = 1$)



CSI 模式串行传输时序 (相同电平下通信)
(当 $\text{DAPmn} = 0$ 且 $\text{CKPmn} = 1$, 或 $\text{DAPmn} = 1$ 且 $\text{CKPmn} = 0$)



备注

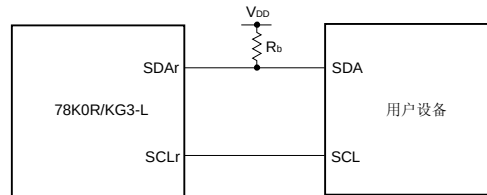
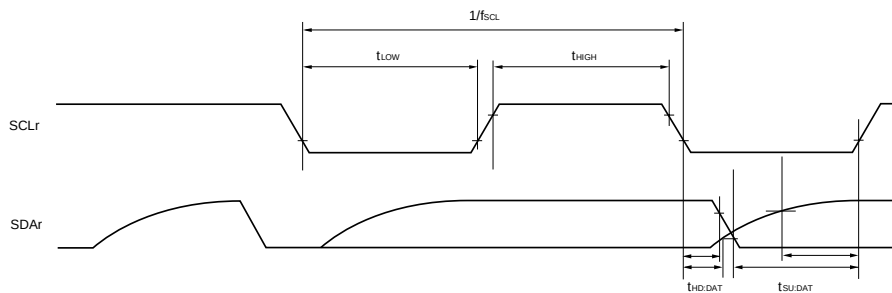
- 1. p: CSI 编号 (p = 00, 01, 10, 20)
- 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2)

(2) 串行接口：串行阵列单元(5/17)

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $2.7\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

(d) 相同电平下通信 (简易 I^2C 模式)

参数	符号	条件	最小值	最大值	单位
SCLr 时钟频率	f_{SCL}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $C_b = 100\text{ pF}$, $R_b = 3\text{ k}\Omega$		400	kHz
保持时间, 当 SCLr = "L"	t_{LOW}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $C_b = 100\text{ pF}$, $R_b = 3\text{ k}\Omega$	995		ns
保持时间, 当 SCLr = "H"	t_{HIGH}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $C_b = 100\text{ pF}$, $R_b = 3\text{ k}\Omega$	995		ns
数据设置时间 (接收)	$t_{SU, DAT}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $C_b = 100\text{ pF}$, $R_b = 3\text{ k}\Omega$	$1/f_{MCK} + 120$		ns
数据保持时间 (发送)	$t_{HD, DAT}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $C_b = 100\text{ pF}$, $R_b = 3\text{ k}\Omega$	0	T.B.D	ns

简易 I^2C 模式连接图 (相同电平下通信)简易 I^2C 模式串行传输时序 (相同电平下通信)

注意事项 用 PIMg 和 POMg 寄存器，为 SDAr 选择普通输入缓存器和 N 沟道漏极开路输出 (V_{DD} 耐压) 模式，为 SCLr 选择普通输出模式输出。

备注

- $R_b[\Omega]$: 通讯线 (SDAr) 上拉电阻
 $C_b[F]$: 通信线 (SCLr, SDAr) 负载电容
- r: IIC 编号($r = 10, 20$), g: PIM 和 POM 编号($g = 0, 14$)
- f_{MCK} : 串行阵列单元工作时钟频率
(由 SMRMn 寄存器的 CKSmn 位设置工作时钟。m: 单元编号 ($m = 0, 1$),
n: 通道编号 ($n = 0, 2$), $mn = 02, 10$)

(2) 串行接口：串行阵列单元(6/17)

(T_A = -40 至+85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(e) 在不同电势 (2.5V, 3V) (UART 模式) (专用波特率发生器输出) 通信期间 (1/2)

参数	符号	条件		最小值	典型值	最大值	单位
传输速率		接收	4.0 V ≤ V _{DD} ≤ 5.5 V,			f _{MCK} /6	bps
			2.7 V ≤ V _b ≤ 4.0 V	f _{CLK} = 20 MHz, f _{MCK} = f _{CLK}		3.3	Mbps
			2.7 V ≤ V _{DD} < 4.0 V,			f _{MCK} /6	bps
			2.3 V ≤ V _b ≤ 2.7 V	f _{CLK} = 20 MHz, f _{MCK} = f _{CLK}		3.3	Mbps

注意事项 使用 PIMg 和 POMg 寄存器，为 RxDq 选择 TTL 输入缓冲器，为 TxDq 选择 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。

备注

1. q: UART 编号(q = 0 至 2), g: PIM 和 POM 编号(g = 0, 1, 14)
2. f_{MCK}: 串行阵列单元工作时钟频率
(由 SMRmn 寄存器的 CKSmn 位设置工作时钟。m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3))
3. 以下的 V_{IH} 和 V_{IL} 是在 UART 模式中使用不同电平通信时的串行阵列单元 AC 特性的观察点。
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) 串行接口：串行阵列单元(7/17)

(TA = -40 至+85°C, 2.7 V ≤ VDD = EVDD0 = EVDD1 ≤ 5.5 V, VSS = EVSS0 = EVSS1 = AVSS = 0 V)

(e) 在不同电势 (2.5V, 3V) (UART 模式) (专用波特率发生器输出) 通信期间 (2/2)

参数	符号	条件		最小值	典型值	最大值	单位
传输速率		发送	4.0 V ≤ VDD ≤ 5.5 V,			注 1	
			2.7 V ≤ Vb ≤ 4.0 V	fCLK = 16.8 MHz, fMCK = fCLK, Cb = 50 pF, Rb = 1.4 kΩ, Vb = 2.7 V		2.8 ^{※2}	Mbps
			2.7 V ≤ VDD < 4.0 V,			注 3	
			2.3 V ≤ Vb ≤ 2.7 V	fCLK = 19.2 MHz, fMCK = fCLK, Cb = 50 pF, Rb = 2.7 kΩ, Vb = 2.3 V		1.2 ^{※4}	Mbps

注 1. 使用 fMCK/6 或下面公式计算的较小的最大传输速率是有效最大传输速率。
当 4.0 V ≤ VDD = EVDD0 = EVDD1 ≤ 5.5 V 和 2.7 V ≤ Vb ≤ 4.0 V 时，用于计算传输速率的公式。

$$\text{最大传输速率} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{波特率误差（理论值）} = \frac{\frac{1}{\text{传输速率} \times 2} - C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})}{(\frac{1}{\text{传输速率}}) \times \text{传输位数}} \times 100 \text{ [%]}$$

* 该值是发送和接收端之间相对差的理论值。

2. 作为示例，当满足“条件”栏中描述的条件时，将计算该值。参见上述注 1，根据用户实际条件来计算最大传输速率。

3. 使用 fMCK/6 或下面公式计算的较小的最大传输速率是有效最大传输速率。

当 2.7 V ≤ VDD = EVDD0 = EVDD1 < 4.0 V 和 2.3 V ≤ Vb ≤ 2.7 V 时，用于计算传输速率的公式。

$$\text{最大传输速率} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{波特率误差（理论值）} = \frac{\frac{1}{\text{传输速率} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{传输速率}}) \times \text{传输位数}} \times 100 \text{ [%]}$$

* 该值是发送和接收端之间相对差的理论值。

4. 作为示例，当满足“条件”栏中描述的条件时，将计算该值。参见上述注 3，根据用户实际条件来计算最大传输速率。

注意事项 使用 PIMg 和 POMg 寄存器，为 RxDq 选择 TTL 输入缓冲器，为 TxDq 选择 N 沟道漏极开路输出 (VDD 耐压) 模式。

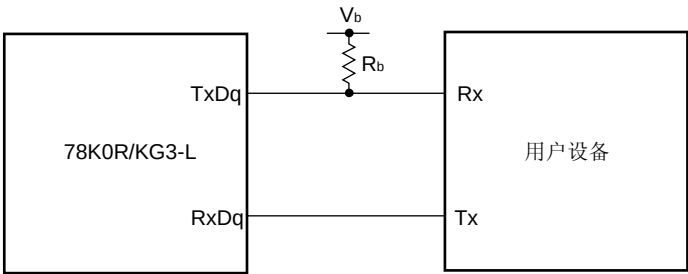
(备注在下页列出)

(2) 串行接口: 串行阵列单元(8/17)

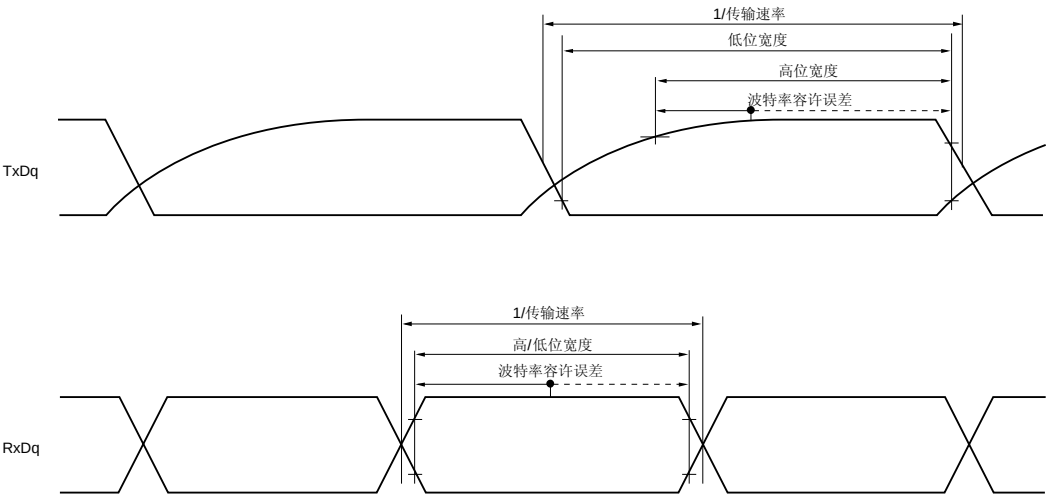
- 备注
1. $R_b[\Omega]$: 通讯线 (TxDq) 上拉电阻,
 $C_b[F]$: 通信线 (TxDq) 负载电容, $V_b[V]$: 通信线电压
 2. q: UART 编号(q = 0 至 2), g: PIM 和 POM 编号(g = 0, 1, 14)
 3. f_{MCK} : 串行阵列单元工作时钟频率
 (由 SMRmn 寄存器的 CKSmn 位设置工作时钟。m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 3))
 4. 以下的 V_{IH} 和 V_{IL} 是在 UART 模式中使用不同电平通信时的串行阵列单元 AC 特性的观察点。
 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}, 2.7\text{ V} \leq V_b \leq 4.0\text{ V}: V_{IH} = 2.2\text{ V}, V_{IL} = 0.8\text{ V}$
 $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}, 2.3\text{ V} \leq V_b \leq 2.7\text{ V}: V_{IH} = 2.0\text{ V}, V_{IL} = 0.5\text{ V}$

(2) 串行接口：串行阵列单元(9/17)

UART 模式连接图 (不同电平通信期间)



UART 模式位宽度 (不同电势通信期间) (参考)



注意事项 使用 PIMg 和 POMg 寄存器，为 RxDq 选择 TTL 输入缓冲器，为 TxDq 选择 N 沟道漏极开路输出(V_{DD} 耐压) 模式。

备注 1. R_b[Ω]: 通信线 (TxDq) 上拉电阻, V_b[V]: 通信线电压
2. q: UART 编号(q = 0 至 2), g: PIM 和 POM 编号(g = 0, 1, 14)

(2) 串行接口: 串行阵列单元(10/17)

(TA = -40 至 +85°C, 2.7 V ≤ VDD = EVDD0 = EVDD1 ≤ 5.5 V, VSS = EVSS0 = EVSS1 = AVSS = 0 V)

(f) 不同电势通信期间 (2.5 V, 3 V) (CSI 模式) (主设备模式, SCKp... 内部时钟输出) (1 / 2)

参数	符号	条件	最小值	典型值	最大值	单位
SCKp 周期时间	t _{KCY1}	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	500			ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	1000			ns
SCKp 高电平宽度	t _{KH1}	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	t _{KCY1} /2 – 120			ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	t _{KCY1} /2 – 275			ns
SCKp 低电平宽度	t _{KL1}	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	t _{KCY1} /2 – 20			ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	t _{KCY1} /2 – 35			ns
Slp 设置时间 (到 SCKp↑) ^{‡1}	t _{SIK1}	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	195			ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	380			ns
Slp 保持时间 (从 SCKp↑) [‡]	t _{KSI1}	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	30			ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	30			ns
从 SCKp↓ 到 SOp 输出的延迟时间 [‡]	t _{KSO1}	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ			165	ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ			320	ns

注 当 DAPmn = 0 且 CKPmn = 0, 或 DAPmn = 1 且 CKPmn = 1。

注意事项 使用 PIMg 和 POMg 寄存器, 为 Slp 选择 TTL 输入缓冲器, 为 SOp 和 SCKp 选择 N 沟道漏极开路输出 (VDD 耐压) 模式。

- 备注
1. p: CSI 编号(p = 01, 10, 20), g: PIM 和 POM 编号 (g = 0, 1, 14)
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2)
 3. Rb[Ω]: 通信线 (SCKp, SOp) 上拉电阻,
Cb[F]: 通信线 (Slp, SOp, SCKp) 负载电容, Vb[V]: 通信线电压
 4. 以下的 VIH 和 VIL 是在 CSI 模式下使用不同电平通信时的串行阵列单元的 AC 特性。
2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb ≤ 4.0 V: VIH = 2.2 V, VIL = .8 V
2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V: VIH = 2.0 V, VIL = 0.5 V

(2) 串行接口: 串行阵列单元(11/17)

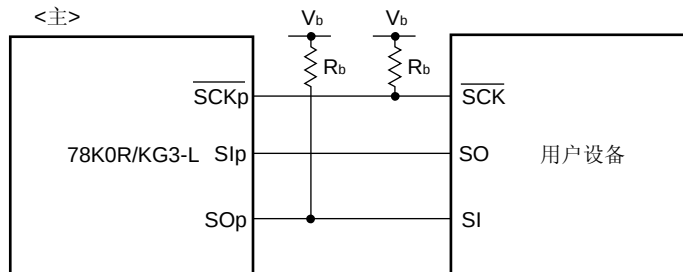
(TA = -40 至 +85°C, 2.7 V ≤ VDD = EVDD0 = EVDD1 ≤ 5.5 V, VSS = EVSS0 = EVSS1 = AVSS = 0 V)

(f) 不同电势通信期间 (2.5 V, 3 V) (CSI 模式) (主设备模式, SCKp... 内部时钟输出) (2 / 2)

参数	符号	条件	最小值	典型值	最大值	单位
Slp 设置时间 (到 SCKp↓) ^{註1}	tSIK1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	70			ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	100			ns
Slp 保持时间 (从 SCKp↓) ^註	tKSI1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	30			ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	30			ns
从 SCKp↑ 到 SOp 输出的延迟时间 ^註	tKSO1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ			40	ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ			40	ns

注 当 DAPmn = 0 且 CKPmn = 1, 或 DAPmn = 1 且 CKPmn = 0。

CSI 模式连接图 (不同电平通信期间)

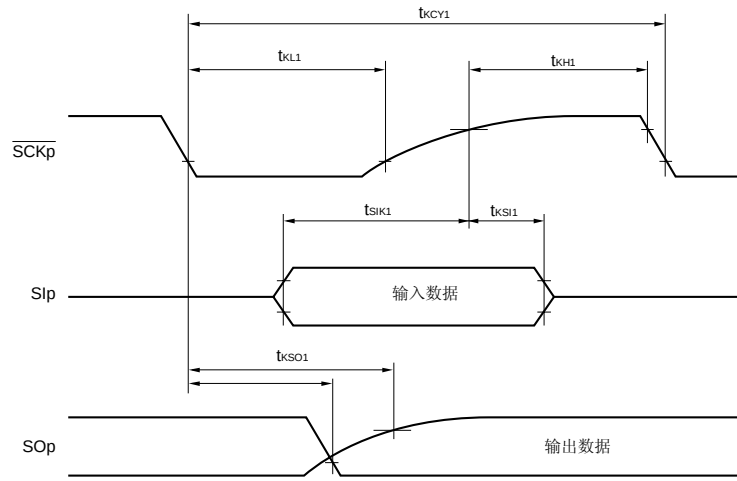


注意事项 使用 PIMg 和 POMg 寄存器, 为 Slp 选择 TTL 输入缓冲器, 为 SOp 和 SCKp 选择 N 沟道漏极开路输出 (VDD 耐压) 模式。

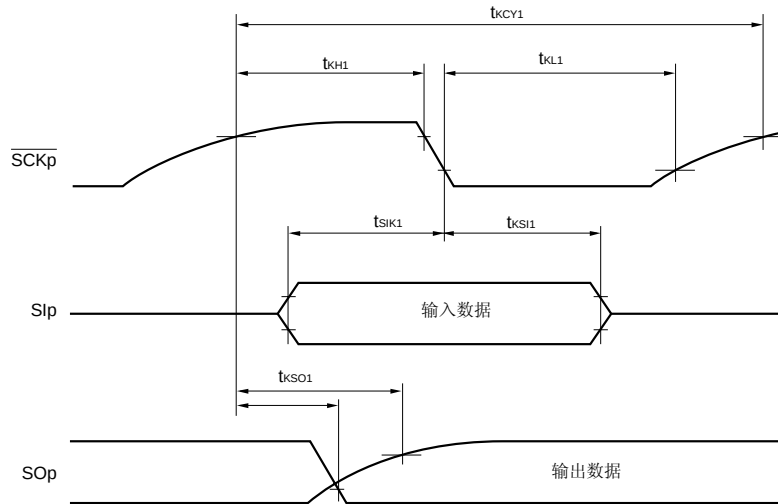
- 备注**
1. p: CSI 编号(p = 01, 10, 20), g: PIM 和 POM 编号 (g = 0, 1, 14)
 2. m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2)
 3. Rb[Ω]: 通信线 (SCKp, SOp) 上拉电阻,
Cb[F]: 通信线 (Slp, SOp, SCKp) 负载电容, Vb[V]: 通信线电压
 4. 以下的 VIH 和 VIL 是在 CSI 模式下使用不同电平通信时的串行阵列单元的 AC 特性。
 4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V: VIH = 2.2 V, VIL = 0.8 V
 2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V: VIH = 2.0 V, VIL = 0.5 V

(2) 串行接口：串行阵列单元(12/17)

CSI 模式传输时序 (主模式) (不同电势下通信)
(当 $DAPmn = 0$ 且 $CKPmn = 0$ ，或 $DAPmn = 1$ 且 $CKPmn = 1$)



CSI 模式传输时序 (主模式) (不同电势下通信)
(当 $DAPmn = 0$ 且 $CKPmn = 1$ ，或 $DAPmn = 1$ 且 $CKPmn = 0$)



注意事项 使用 PIMg 和 POMg 寄存器，为 Slp 选择 TTL 输入缓冲器，为 SOp 和 SCKp 选择 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。

备注

1. p: CSI 编号($p = 01, 10, 20$)，g: PIM 和 POM 编号 ($g = 0, 1, 14$)
2. m: 单元编号 ($m = 0, 1$)，n: 通道编号 ($n = 0$ 至 2)

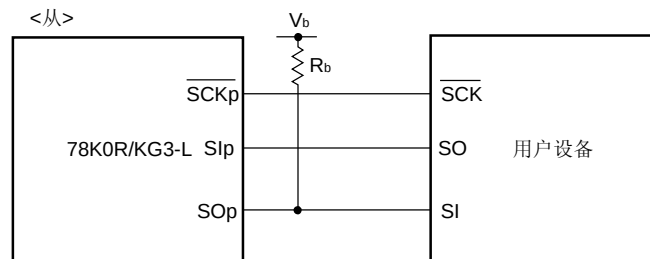
(2) 串行接口：串行阵列单元(13/17)

(T_A = -40 至 +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)(g) 不同电势通信期间 (2.5 V, 3 V) (CSI 模式) (从设备模式, $\overline{\text{SCKp}}$... 外部时钟输入)

参数	符号	条件	最小值	典型值	最大值	单位
$\overline{\text{SCKp}}$ 周期时间	t_{KCY2}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V	16.6 MHz < f _{MCK}	12/f _{MCK}		ns
			12.5 MHz < f _{MCK} ≤ 16.6 MHz	10/f _{MCK}		ns
			8.3 MHz < f _{MCK} ≤ 12.5 MHz	8/f _{MCK}		ns
			f _{MCK} ≤ 8.3 MHz	6/f _{MCK}		ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V	17.5 MHz < f _{MCK}	18/f _{MCK}		ns
			15 MHz < f _{MCK} ≤ 17.5 MHz	16/f _{MCK}		ns
			12.5 MHz < f _{MCK} ≤ 15 MHz	14/f _{MCK}		ns
			10 MHz < f _{MCK} ≤ 12.5 MHz	12/f _{MCK}		ns
			7.5 MHz < f _{MCK} ≤ 10 MHz	10/f _{MCK}		ns
			5 MHz < f _{MCK} ≤ 7.5 MHz	8/f _{MCK}		ns
			f _{MCK} ≤ 5 MHz	6/f _{MCK}		ns
$\overline{\text{SCKp}}$ 高/低电平宽度	$t_{\text{KH2}},$ t_{KL2}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V	$t_{\text{KCY2}}/2 - 20$			ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V	$t_{\text{KCY2}}/2 - 35$			ns
Slp 设置时间 (到 $\overline{\text{SCKp}}\uparrow$) ^{注 1}	t_{SIK2}		1/f _{MCK} + 90			ns
Slp 保持时间 (从 $\overline{\text{SCKp}}\uparrow$) ^{注 1}	t_{KSIZ}		50			ns
从 $\overline{\text{SCKp}}\downarrow$ 到 SOp 输出的延迟时间 ^{注 2}	t_{KSO2}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 50 pF, R _b = 1.4 kΩ			1/f _{MCK} + 245	ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 50 pF, R _b = 2.7 kΩ			1/f _{MCK} + 400	ns

- 注 1. 当 DAPmn = 0 和 CKPmn = 0, 或 DAPmn = 1 和 CKPmn = 1 时。Slp 设置时间变为“至 $\overline{\text{SCKp}}\downarrow$ ”, 当 DAPmn = 0 和 CKPmn = 1, 或 DAPmn = 1 和 CKPmn = 0。
2. 当 DAPmn = 0 和 CKPmn = 0, 或 DAPmn = 1 和 CKPmn = 1 时。到 SOp 输出的延迟时间变为“从 $\overline{\text{SCKp}}\uparrow$ ”, 当 DAPmn = 0 和 CKPmn = 1, 或 DAPmn = 1 和 CKPmn = 0。

CSI 模式连接图（不同电平通信期间）



(注意事项和备注在下页中给出。)

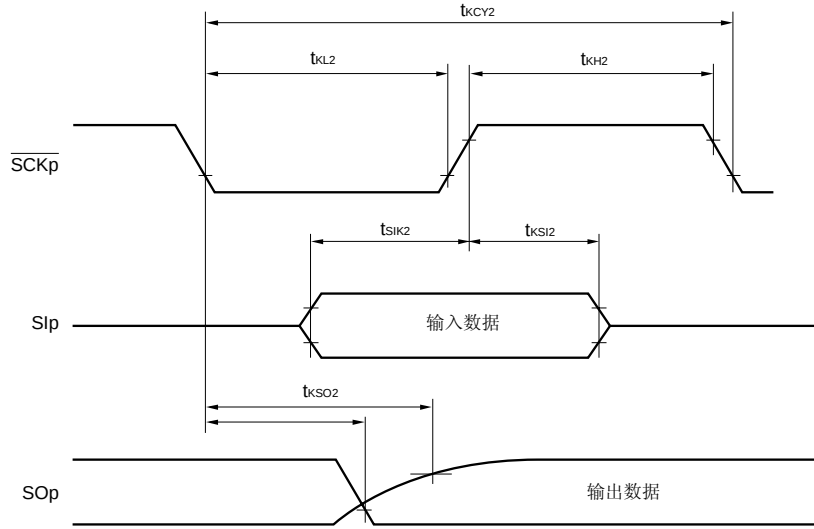
(2) 串行接口：串行阵列单元(14/17)

注意事项 使用 PIMg 和 POMg 寄存器，为 SIp 和 $\overline{\text{SCKp}}$ 选择 TTL 输入缓冲器，为 SOp 选择 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。

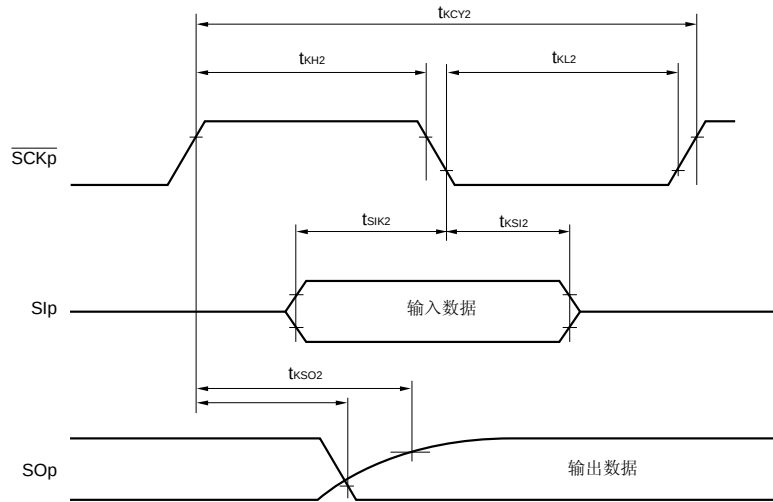
- 备注**
1. p: CSI 编号(p = 01, 10, 20), g: PIM 和 POM 编号 (g = 0, 1, 14)
 2. $R_b[\Omega]$: 通信线 (SOp) 上拉电阻,
 $C_b[F]$: 通信线 (SOp, $\overline{\text{SCKp}}$)负载电容, $V_b[V]$: 通信线电压
 3. f_{MCK} : 串行阵列单元工作时钟频率
 (由 SMRmn 寄存器的 CKSmn 位设置工作时钟。m: 单元编号 (m = 0, 1), n: 通道编号 (n = 0 至 2))
 4. 以下的 V_{IH} 和 V_{IL} 是在 CSI 模式下使用不同电平通信时的串行阵列单元的 AC 特性。
 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$: $V_{IH} = 2.2\text{ V}$, $V_{IL} = 0.8\text{ V}$
 $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$, $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$: $V_{IH} = 2.0\text{ V}$, $V_{IL} = 0.5\text{ V}$

(2) 串行接口: 串行阵列单元(15/17)

CSI 模式传输时序 (从模式) (不同电势下通信)
(当 $DAPmn = 0$ 且 $CKPmn = 0$, 或 $DAPmn = 1$ 且 $CKPmn = 1$)



CSI 模式传输时序 (从模式) (不同电势下通信)
(当 $DAPmn = 0$ 且 $CKPmn = 1$, 或 $DAPmn = 1$ 且 $CKPmn = 0$)



注意事项 使用 PIMg 和 POMg 寄存器, 为 Slp 和 SCKp 选择 TTL 输入缓冲器, 为 SOp 选择 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。

备注

1. p: CSI 编号 ($p = 01, 10, 20$), g: PIM 和 POM 编号 ($g = 0, 1, 14$)
2. m: 单元编号 ($m = 0, 1$), n: 通道编号 ($n = 0$ 至 2)

(2) 串行接口: 串行阵列单元(16/17)

(T_A = -40 至 +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)(h) 不同电势通信期间 (2.5 V, 3 V) (简易 I²C 模式)

参数	符号	条件	最小值	最大值	单位
SCLr 时钟频率	f _{SCL}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 1.4 kΩ		400	kHz
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ		400	kHz
保持时间, 当 SCLr = "L"	t _{LOW}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 1.4 kΩ	1065		ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ	1065		ns
保持时间, 当 SCLr = "H"	t _{HIGH}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 1.4 kΩ	445		ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ	445		ns
数据设置时间 (接收)	t _{SU: DAT}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 1.4 kΩ	1/f _{MCK} + 190		ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ	1/f _{MCK} + 190		ns
数据保持时间 (发送)	t _{HD: DAT}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 100 pF, R _b = 1.4 kΩ	0	T.B.D	ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b ≤ 2.7 V, C _b = 100 pF, R _b = 2.7 kΩ	0	T.B.D	ns

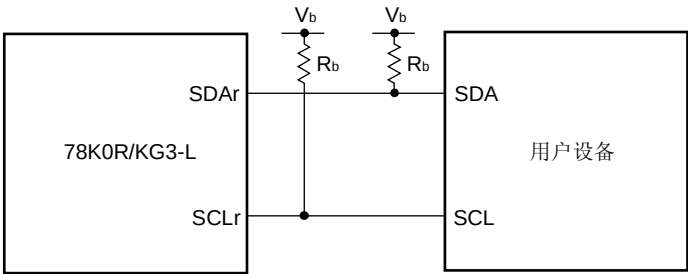
注意事项 通过 PIMg 和 POMg 寄存器, 为 SDAr 选择普通输入缓冲器和 N 沟道漏极开路输出 (V_{DD} 耐压) 模式, 为 SCLr 选择 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。

备注

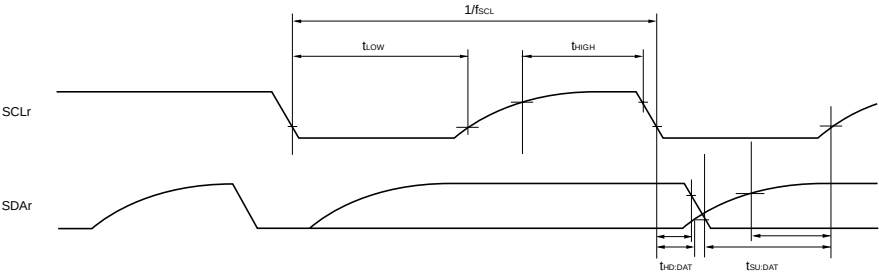
- R_b[Ω]: 通讯线 (SDAr, SCLr) 上拉电阻,
C_b[F]: 通信线 (SDAr, SCLr) 负载电容, V_b[V]: 通信线电压
- r: IIC 编号 (r = 10, 20), g: PIM 和 POM 编号 (g = 0, 14)
- f_{MCK}: 串行阵列单元工作时钟频率
(由 SMRmn 寄存器的 CKSmn 位设置工作时钟。m: 单元编号 (m = 0, 1),
n: 通道编号 (n = 0, 2), mn = 02, 10)
- 以下的 V_{IH} 和 V_{IL} 是在 CSI 模式下使用不同电平通信时的串行阵列单元的 AC 特性
在简易 I²C 模式的不同电势下进行通信。
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) 串行接口：串行阵列单元(17/17)

简易 I²C 模式连接图 (不同电平通信期间)



简易 I²C 模式串行传输时序 (不同电平通信期间)



注意事项 通过 PIMg 和 POMg 寄存器，为 SDAr 选择普通输入缓冲器和 N 沟道漏极开路输出 (V_{DD} 耐压)模式，为 SCLr 选择 N 沟道漏极开路输出 (V_{DD} 耐压)模式。

备注

1. R_b[Ω]: 通信线 (SDAr, SCLr) 上拉电阻, V_b[V]: 通信线电压
2. r: IIC 编号(r = 10, 20), g: PIM 和 POM 编号 (g = 0, 14)

(3) 串行接口：IICA

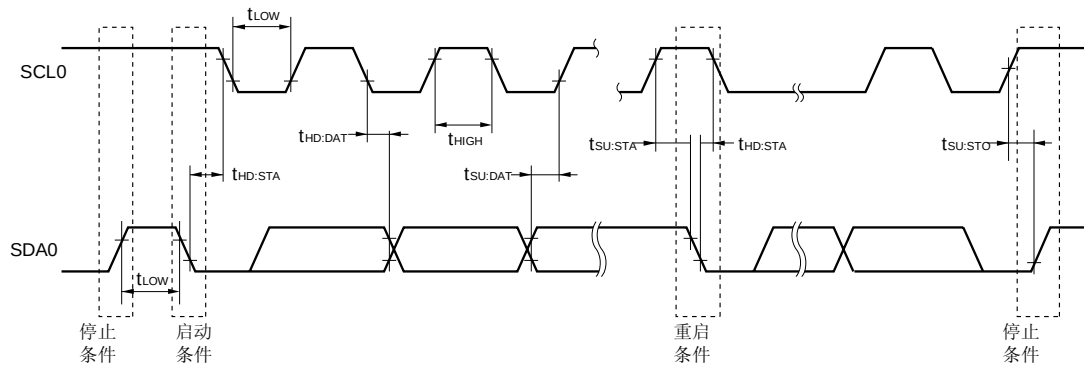
($T_A = -40$ 至 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$)

(a) IICA

参数	符号	条件	标准模式		高速模式		单位
			最小值	最大值	最小值	最大值	
SCL0 时钟频率	f_{SCL}		0	100	0	400	kHz
重启条件的设置时间 ^{注 1}	$t_{\text{SU: STA}}$		4.7		0.6		μs
保持时间	$t_{\text{HD: STA}}$		4.0		0.6		μs
保持时间, 当 SCL0 == "L"	t_{LOW}		4.7		1.3		μs
保持时间, 当 SCL0 = "H"	t_{HIGH}		4.0		0.6		μs
数据设置时间（接收）	$t_{\text{SU: DAT}}$		250		100		ns
数据保持时间（发送） ^{注 2}	$t_{\text{HD: DAT}}$		0	3.45	0	0.9	μs
停止条件的设置时间	$t_{\text{SU: STO}}$		4.0		0.6		μs
总线释放时间	t_{BUF}		4.7		1.3		μs

- 注
1. 当检测到启动/重启条件时, 在这个周期后第一个时钟脉冲产生。
 2. 正常传输时 $t_{\text{HD: DAT}}$ 的最大值并在 $\overline{\text{ACK}}$ （响应）时序中插入等待状态。

IICA 串行传输时序



(4) 串行接口：片上调试 (UART)

(TA = -40 至 +85°C, 1.8 V ≤ VDD = EVDD0 = EVDD1 ≤ 5.5 V, VSS = EVSS0 = EVSS1 = AVSS = 0 V)

(a) 片上调试 (UART)

参数	符号	条件	最小值	典型值	最大值	单位
传输速率			fCLK/2 ¹²		fCLK/6	bps
		Flash 存储器编程模式			3.33	Mbps
TOOL1 输出频率	fTOOL1	2.7 V ≤ VDD ≤ 5.5 V			10	MHz
		1.8 V ≤ VDD < 2.7 V			2.5	MHz

A/D 转换器特性

(T_A = -40 至 +85°C, 1.8 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

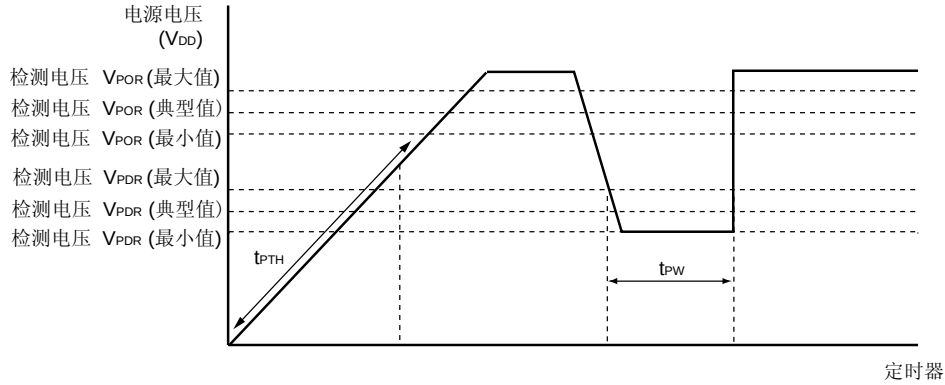
参数	符号	条件	最小值	典型值	最大值	单位
分辨率	RES				10	位
总误差 ^{注 1, 2}	AINL	4.0 V ≤ AV _{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV _{REF} < 4.0 V			±0.6	%FSR
		1.8 V ≤ AV _{REF} < 2.7 V			±1.2	%FSR
转换时间	t _{CONV}	4.0 V ≤ AV _{REF} ≤ 5.5 V	高速模式 1	2.5		66.6 μs
			普通模式	6.1		66.6 μs
		2.7 V ≤ AV _{REF} < 4.0 V	高速模式 2	4.5		66.6 μs
			普通模式	12.2		66.6 μs
		1.8 V ≤ AV _{REF} < 2.7 V	升压模式	27.0		66.6 μs
零度误差 ^{注 1, 2}	E _{ZS}	4.0 V ≤ AV _{REF} ≤ 5.5 V	普通模式		±0.4	%FSR
		2.7 V ≤ AV _{REF} < 4.0 V	普通模式		±0.6	%FSR
满度误差 ^{注 1, 2}	E _{FS}	4.0 V ≤ AV _{REF} ≤ 5.5 V	普通模式		±0.4	%FSR
		2.7 V ≤ AV _{REF} < 4.0 V	普通模式		±0.6	%FSR
积分线性误差 ^{注 1}	I _{LE}	4.0 V ≤ AV _{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV _{REF} < 4.0 V			±4.5	LSB
		1.8 V ≤ AV _{REF} < 2.7 V			±6.5	LSB
微分线性误差 ^{注 1}	D _{LE}	4.0 V ≤ AV _{REF} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV _{REF} < 4.0 V			±2.0	LSB
		1.8 V ≤ AV _{REF} < 2.7 V			±2.0	LSB
模拟 输入电压	V _{AIN}		AV _{SS}		AV _{REF}	V

- 注 1. 不包括量化误差 (±1/2 LSB)。
 2. 此值表示对满度的百分比 (%FSR)。

POC 电路特性 ($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V_{POR}	供电电压上升时间	1.52	1.61	1.70	V
	V_{PDR}	供电电压下降时间	1.50	1.59	1.68	V
供电电压上升斜率	t_{PTH}	改变 V_{DD} 的趋势: $0\text{ V} \rightarrow V_{POR}$	0.5			V/ms
最小脉冲宽度	t_{PW}	当电压下降时	200			μs
检测延迟时间					200	μs

POC 电路时序



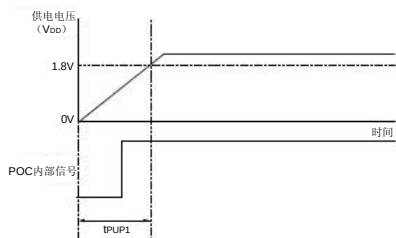
供电电压上升时间 ($T_A = -40$ 至 $+85^\circ\text{C}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
上升到 1.8 V (V_{DD} (最小值)) [※] 的最大时间 (V_{DD} : $0\text{ V} \rightarrow 1.8\text{ V}$)	t_{PUP1}	当不使用 $\overline{\text{RESET}}$ 输入时, LVI 默认的启动功能设置为停止 (LVIOFF (选项字节) = 1)。			3.6	ms
上升到 1.8 V (V_{DD} (最小值)) [※] 的最大时间 (释放 $\overline{\text{RESET}}$ 输入 $\rightarrow V_{DD}$: 1.8 V)	t_{PUP2}	当使用 $\overline{\text{RESET}}$ 复位输入时, LVI 默认启动功能设置停止 (LVIOFF (选项字节) = 1)。			1.88	ms

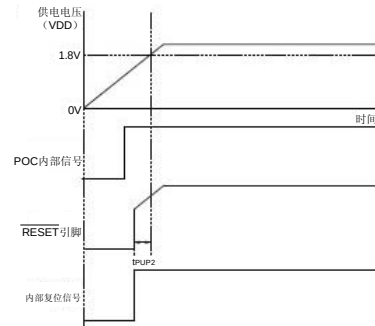
注 确保供电电压上升时间小于此值。

供电电压上升时间时序

- 当不使用 $\overline{\text{RESET}}$ 引脚输入时



- 当使用 $\overline{\text{RESET}}$ 引脚输入时（在 POC 释放之后，由 $\overline{\text{RESET}}$ 引脚释放外部复位）



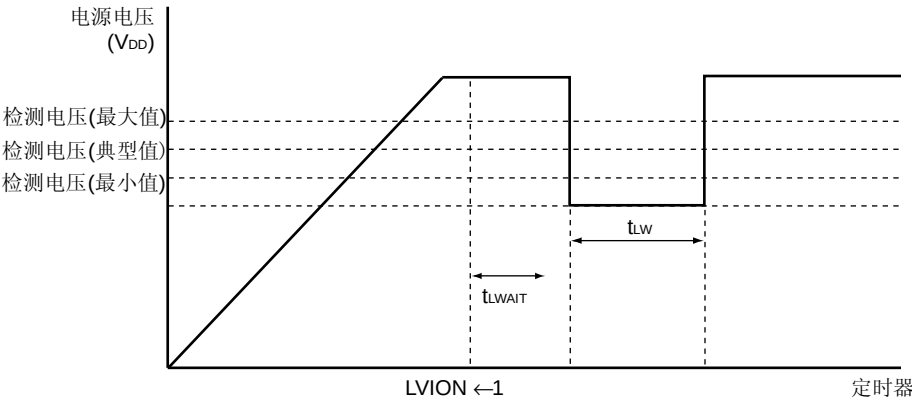
LVI 电路特性(T_A = -40 至 +85°C，V_{PDR} ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V，V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

参数		符号	条件	最小值	典型值	最大值	单位
检测电压	供电电压电平	V _{LVI0}		4.12	4.22	4.32	V
		V _{LVI1}		3.97	4.07	4.17	V
		V _{LVI2}		3.82	3.92	4.02	V
		V _{LVI3}		3.66	3.76	3.86	V
		V _{LVI4}		3.51	3.61	3.71	V
		V _{LVI5}		3.35	3.45	3.55	V
		V _{LVI6}		3.20	3.30	3.40	V
		V _{LVI7}		3.05	3.15	3.25	V
		V _{LVI8}		2.89	2.99	3.09	V
		V _{LVI9}		2.74	2.84	2.94	V
		V _{LVI10}		2.58	2.68	2.78	V
		V _{LVI11}		2.43	2.53	2.63	V
		V _{LVI12}		2.28	2.38	2.48	V
		V _{LVI13}		2.12	2.22	2.32	V
		V _{LVI14}		1.97	2.07	2.17	V
		V _{LVI15}		1.81	1.91	2.01	V
	外部输入引脚 ^{註 1}	V _{EXLVI}	EXLVI < V _{DD} , 1.8 V ≤ V _{DD} ≤ 5.5 V	1.11	1.21	1.31	V
	电源应用上的供电电压	V _{PUPLVI}	当设置 LVI 默认开始功能允许时	1.87	2.07	2.27	V
最小脉冲宽度		t _{LW}		200			μs
检测延迟时间						200	μs
操作稳定等待时间 ^{註 2}		t _{LWAIT}				10	μs

- 注
- 1. 使用 EXLVI/P120/INTP0 引脚。
 - 2. 从设置低电压检测寄存器 (LVIM) 的第 7 位 (LVION) 为 1 到操作稳定所需要的时间。

备注 V_{LVI(n-1)} > V_{LVI}n: n = 1 至 15

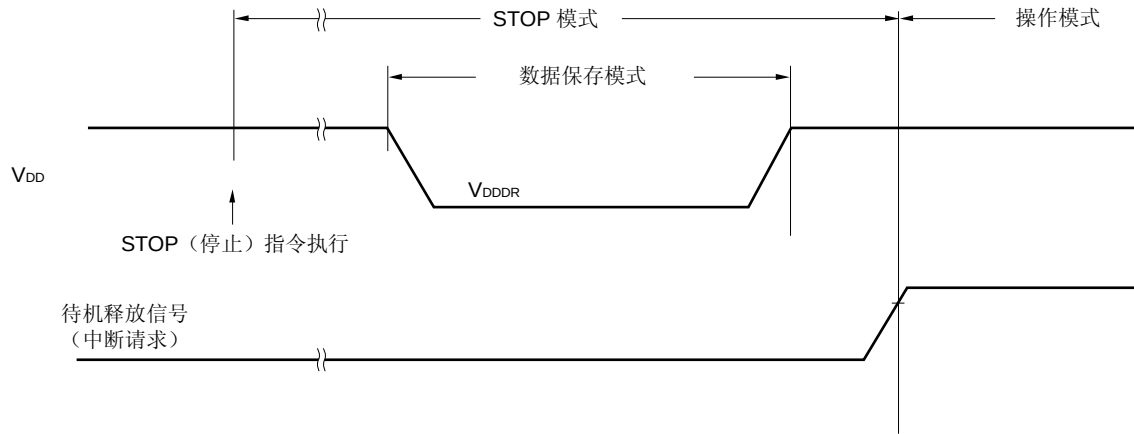
LVI 电路时序



数据存储器停止模式低供电电压数据保持特性($T_A = -40$ 至 $+85^{\circ}\text{C}$)

参数	符号	条件	最小值	典型值	最大值	单位
数据保持供电电压	V_{DDDR}		1.5 ^注		5.5	V

注 该值根据 POC 检测电压而变化。当电压降低时，数据可保存直到 POC 复位前，当 POC 复位后数据不可保存。



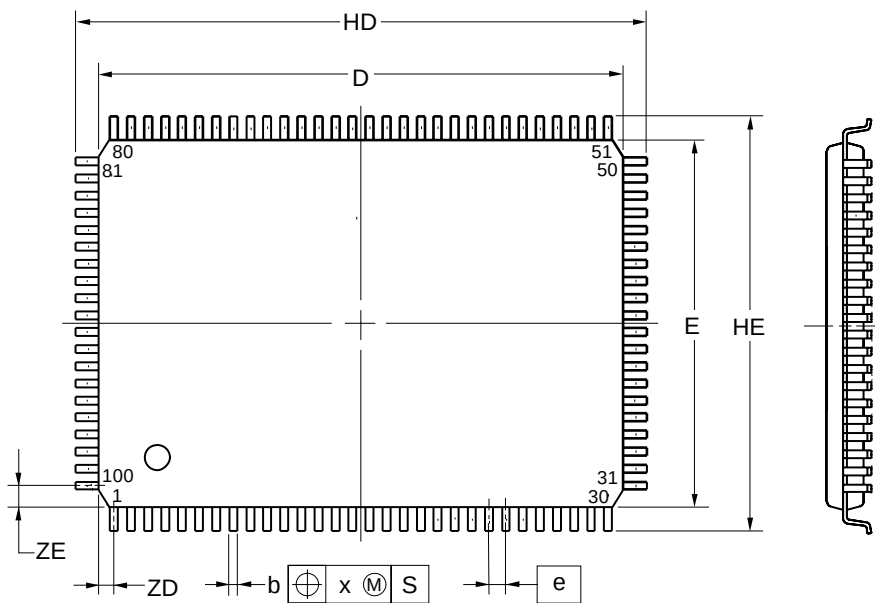
Flash 存储器编程特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$, $V_{SS} = EV_{SS0} = EV_{SS1} = 0\text{ V}$)

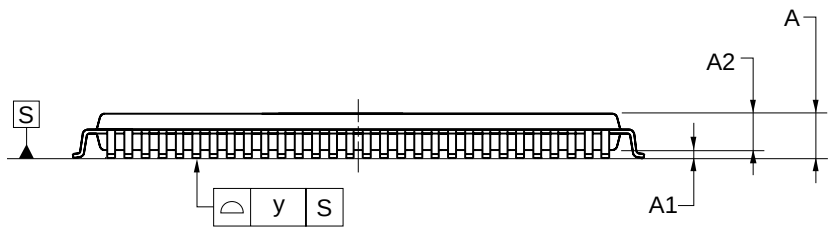
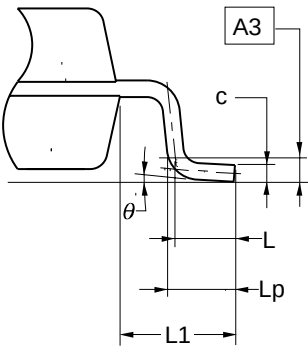
参数	符号	条件	最小值	典型值	最大值	单位
V_{DD} 供电电流	I_{DD}	典型值 = 10 MHz, 最大值 = 20 MHz		6	20	mA
CPU/外围硬件时钟频率	f_{CLK}	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2		20	MHz
重写数（每个时钟的删除数）	C_{erwr}	用于更新程序，当使用 flash 存储器编程器和 NEC 电子的自编程库	保存 15 年	1000		次
		用于更新数据，当使用日电电子的 EEPROM 仿真库时	保存 5 年	10,000		次

备注 当多次更新数据时，使用 flash 存储器更新数据。

100引脚塑封LQFP（14x20）



铅端的详细信息



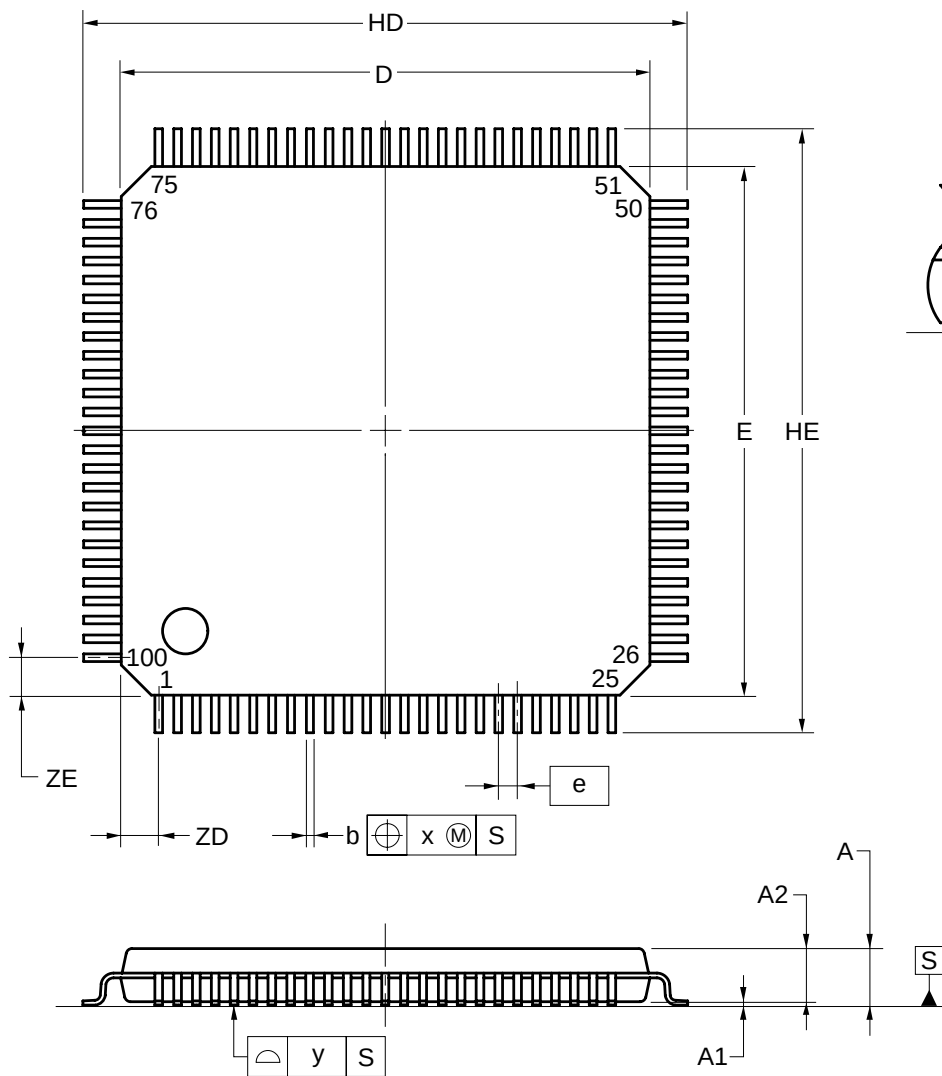
注
每个末端中心线应该在定位在实际位置的0.13mm处，在最大材料条件下

(UNIT:mm)

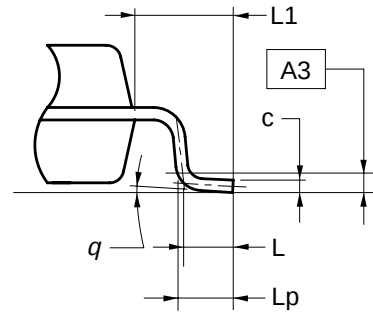
ITEM	尺寸
D	20.00 ±0.20
E	14.00 ±0.20
HD	22.00 ±0.20
HE	16.00 ±0.20
A	1.60 MAX .
A1	0.10 ±0.05
A2	1.40 ±0.05
A3	0.25
b	0.30 ^{+0.08} _{-0.04}
c	0.125 ^{+0.075} _{-0.025}
L	0.50
Lp	0.60 ±0.15
L1	1.00 ±0.20
θ	3° ^{+5°} _{-3°}
e	0.65
x	0.13
y	0.10
ZD	0.575
ZE	0.825

P100GF-65-G AS

100引脚塑封 LQFP(FINEPTICH) (14x14)



铅端的详细信息



(UNIT:mm)

ITEM	尺寸
D	14.00 ±0.20
E	14.00 ±0.20
HD	16.00 ±0.20
HE	16.00 ±0.20
A	1.60 MAX .
A1	0.10 ±0.05
A2	1.40 ±0.05
A3	0.25
b	0.20 ^{+0.07} / _{-0.03}
c	0.125 ^{+0.075} / _{-0.025}
L	0.50
Lp	0.60 ±0.15
L1	1.00 ±0.20
q	3° ^{+5°} / _{-3°}
e	0.50
x	0.08
y	0.08
ZD	1.00
ZE	1.00

P100GC-50-UEU-1

© NEC Electronics Corporation 2007

附录 A 开发工具

以下开发工具可用来开发采用 78K0R/KG3-L 的系统。

图 A-1 显示了开发工具的组成。

- **支持 PC98-NX 系列**

除特别指定之外，IBM PC/AT™ 支持的产品兼容PC98-NX系列计算机。当使用PC98-NX系列计算机时，参考IBM PC/AT兼容说明。

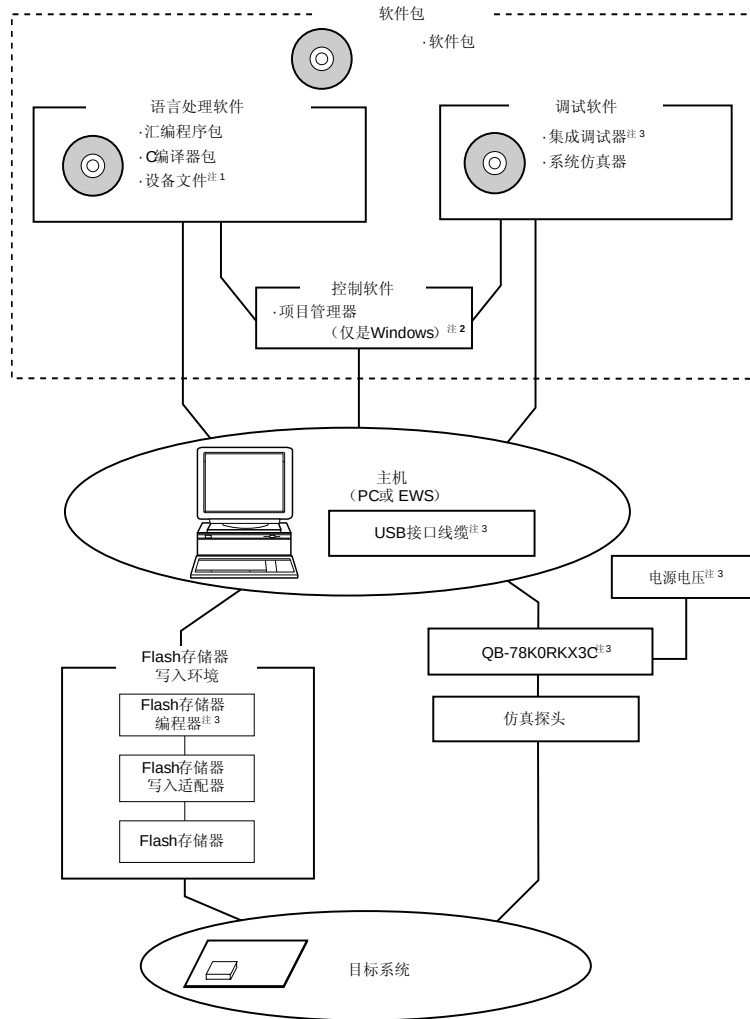
- **Windows™**

除特别指定之外，“Windows”表示下列操作系统 OS。

- Windows 98
- Windows NT™
- Windows 2000
- Windows XP

图 A-1 开发工具的配置(1/2)

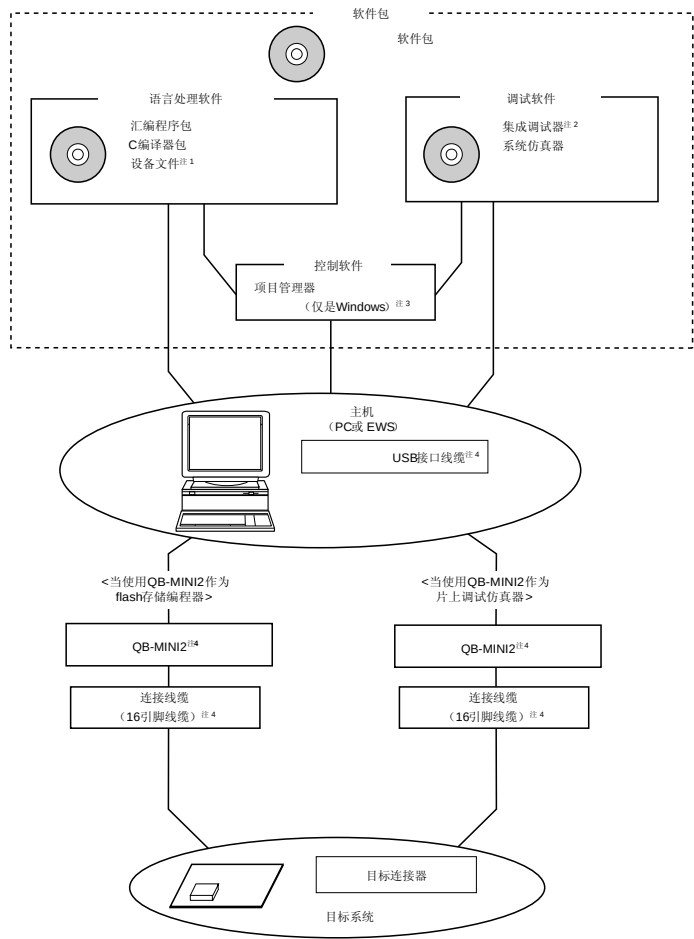
(1) 当使用在线仿真器 QB-78K0RKX3C 时



- 注
1. 开发中。
 2. 汇编编译器中包括项目管理器 PM+。
PM+仅用于 Windows。
 3. 在线仿真器 QB-78K0RKX3C 还带有集成调试器 ID78K0R-QB，具有编程功能的片上调试仿真器 QB-MINI2，电源和 USB 接口电缆。其它产品均独立销售。

图 A-1 开发工具的配置(2/2)

(2) 当使用具有编程功能的片上调试仿真器 QB-MINI2 时



- 注
- 1. 开发中。
 - 2. 可从开发工具下载网站 (<http://www.necel.com/micro/ods/eng/index.html>) 下载集成调试器 (ID78K0R-QB)。
 - 3. 汇编编译器中包括项目管理器 PM+。
PM+仅用于 Windows。
 - 4. 购买片上调试仿真器 QB-MINI2 时还提供 USB 接口电缆、连接电缆(10 引脚电缆和 16 引脚电缆)和 78K0-OCD 板。其它产品均独立销售。另外, 可从 MINICUBE2 的下载网站 (<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>) 下载操作 QB-MINI2 的软件。

A.1 软件包

SP78K0R	开发工具（软件包）适用于 78K0R 微控制器。
78K0R 系列软件包	部件编号: μ SxxxxSP78K0R

备注 xxxx 部件编号根据所使用的主设备和操作系统而异。

μ SxxxxSP78K0R

xxxx	主设备	操作系统	存储介质
AB17	PC-9800 系列,	Windows (日文版)	CD-ROM
BB17	兼容 IBM PC/AT	Windows (英文版)	

A.2 语言处理软件

RA78K0R 汇编编译器	该汇编编译器将助记符编写的程序转换为微处理器可执行的目标代码。 该汇编编译器可以自动建立符号表和对分支指令进行优化。 该汇编编译器应当和设备文件（DF781014）一起使用。 <在 PC 环境中使用 RA78K0R 的注意事项> 该汇编编译器是基于 DOS 应用的。它也可以在 Windows 环境下使用，但必须在 Windows 环境下使用项目管理器(包含在汇编编译器软件包中)。 部件编号: μSxxxxRA78K0R
CC78K0R C 编译器	该编译器将 C 语言编写的程序转换为微控制器可执行的目标代码。 该编译器应当结合汇编编译器软件包和设备文件(两者均单独出售)一起使用。 <在 PC 环境中使用 CC78K0R 的注意事项> 该 C 编译器是基于 DOS 应用的。它也可以在 Windows 环境下使用，但必须在 Windows 环境下使用项目管理器(包含在汇编编译器软件包中)。 部件编号: μSxxxxCC78K0R
DF781014 ^{※1,2} 设备文件	该文件包含设备特有的信息。 该设备文件应当结合工具(RA78K0R、CC78K0R、用于 78K0R 的 SM+、ID78K0R-QB) (均为独立销售)一起使用。 相应的 OS 和主机随使用工具而变化。 部件编号: μSxxxxDF781014

- 注**
1. 开发中。
 2. DF781014 可以与 RA78K0R、CC78K0R、用于 78K0R 的 SM+ 和 ID78K0R-QB 一起使用。

备注 xxxx 部件编号根据所使用的主设备和操作系统而异。

μSxxxxRA78K0R

μSxxxxCC78K0R

xxxx	主设备	操作系统	供应中介
AB17	PC-9800 系列, 兼容 IBM PC/AT	Windows (日文版)	CD-ROM
BB17		Windows (英文版)	

μSxxxxDF781014

xxxx	主设备	操作系统	供应中介
AB13	PC-9800 系列, 兼容 IBM PC/AT	Windows (日文版)	3.5 寸 2HD FD
BB13		Windows (英文版)	

A.3 控制软件

PM+ 项目管理器	本控制软件的设计使用户在 Windows 环境下能够更有效地开发程序。可在项目管理器中执行诸如启动编辑器、编译和启动调试器等用户编程开发时所使用的全部操作。 <注意> 本项目管理器包含在汇编工具包中(RA78K0R)。 只能在 Windows 下使用。

A.4 Flash 存储器编程工具

A.4.1 当使用 Flash 存储器编程器 PG-FP5, FL-PR5 时

PG-FP5, FL-PR5 Flash 存储器编程器	Flash 存储器编程器，专用于片上 Flash 存储器的微控制器。
FA-78F1014GF-GAS-RX (支持 RoHS), FA-78F1014GC-UEU-RX (支持 RoHS) Flash 存储器编程适配器	Flash 存储器编程适配器，用于连接到 Flash 存储器编程器使用。 - FA-78F1014GF-GAS-RX:100 引脚塑封 LQFP (GF-GAS 类型) - FA-78F1014GC-UEU-RX:100 引脚塑封 LQFP (GC-UEU 类型)

备注 FL-PR5, FA-78F1014GF-GAS-RX 和 FA-78F1014GC-UEU-RX 是 Naito Densei Machida Mfg. Co.,Ltd. 的产品。

A.4.2 当使用具有编程功能的片上调试仿真器 QB-MINI2 时

QB-MINI2 具有编程功能的片上调试仿真器	这是具有片上 Flash 存储器的微控制器专用的 Flash 存储器编程器。当使用 78K0R 开发应用系统时，它也可以当作片上调试仿真器，用于调试硬件和软件。 购买 QB-MINI2 时还提供 USB 接口电缆、连接电缆(10 引脚电缆和 16 引脚电缆)和 78K0-OCD 板。要使用 78K0R/KG3-L，需配备 USB 接口电缆和 16 引脚连接线。

备注 另外，可从 MINICUBE2 的下载网站
(<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>)下载操作 QB-MINI2 的软件。

A.5 调试工具 (硬件)

A.5.1 当使用在线仿真器 QB-78K0RKX3C 时

QB-78K0RKX3C 在线仿真器	当使用 78K0R/KG3-L 开发应用系统时，采用该在线仿真器进行硬件和软件调试。该在线仿真器支持集成调试器(ID78K0R-QB)。该仿真器与电源和仿真探头，和用于连接此仿真器到主机的 USB 一起使用。
QB-144-CA-01 引脚检测适配器	该适配器用于示波器等波形检测。
QB-144-EP-02S 仿真探头	这是一种可以灵活使用的仿真探头，用于连接在线仿真器和目标系统。
QB-100GC-EA-07T, QB-100GF-EA-05T 交换适配器	该适配器用于实现从在线仿真器到目标连接器的引脚转换。 • QB-100GC-EA-07T: 100 引脚塑封 LQFP (GC-UEU 类型) • QB-100GF-EA-05T: 100 引脚塑封 LQFP (GF-GAS 类型)
QB-100GC-YS-01T, QB-100GF-YS-01T 空间适配器	该适配器用于调整目标系统和在线仿真器之间的高度。 • QB-100GC-YS-01T: 100 引脚塑封 LQFP (GC-UEU 类型) • QB-100GF-YS-01T: 100 引脚塑封 LQFP (GF-GAS 类型)
QB-100GC-YQ-01T, QB-100GF-YQ-01T YQ 连接器	该连接器用于连接目标连接器和交换适配器。 • QB-100GC-YQ-01T: 100 引脚塑封 LQFP (GC-UEU 类型) • QB-100GF-YQ-01T: 100 引脚塑封 LQFP (GF-GAS 类型)
QB-100GC-HQ-01T, QB-100GF-HQ-03T 安装适配器	该适配器用于通过插座安装目标设备。 • QB-100GC-HQ-01T: 100 引脚塑封 LQFP (GC-UEU 类型) • QB-100GF-HQ-03T: 100 引脚塑封 LQFP (GF-GAS 类型)
QB-100GC-NQ-01T, QB-100GF-NQ-01T 目标连接器	该目标连接器用于安装到目标系统上。 • QB-100GC-NQ-01T: 100 引脚塑封 LQFP (GC-UEU 类型) • QB-100GF-NQ-01T: 100 引脚塑封 LQFP (GF-GAS 类型)

- 备注**
1. QB-78K0RKX3C 提供一个电源和 USB 接口电缆。作为控制软件，还提供集成调试器 ID78K0R-QB 和具有编程功能的片上调试仿真器 QB-MINI2。
 2. 封装内容根据产品型号而不同，如下所示。

封装内容 产品型号	在线仿真器	仿真探头	交换适配器	YQ连接器	目标连接器
QB-78K0RKX3C-ZZZ	QB-78K0RKX3C	无			
QB-78K0RKX3C-T100GC		QB-144-EP-02S	QB-100GC-EA-07T	QB-100GC-YQ-01T	QB-100GC-NQ-01T
QB-78K0RKX3C-T100GF			QB-100GF-EA-05T	QB-100GF-YQ-01T	QB-100GF-NQ-01T

A.5.2 当使用具有编程功能的片上调试仿真器 QB-MINI2

QB-MINI2 具有编程功能的片上调试仿真器	当使用 78K0R 微控制器开发应用系统时，该片上调试仿真器用于调试硬件和软件。也可专门当作具有片上 Flash 存储器的微控制器的 Flash 存储器编程器使用。 购买 QB-MINI2 时还提供 USB 接口电缆、连接电缆(10 引脚电缆和 16 引脚电缆)和 78K0-OCD 板。要使用 78K0R/KG3-L，需配备 USB 接口电缆和 16 引脚连接线。

备注 另外，可从 MINICUBE2 的下载网站
(<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>)下载操作 QB-MINI2 的软件。

A.6 调试工具 (软件)

用于 78K0R 的 SM+ 系统仿真器	用于 78K0R 的 SM+是基于 Windows 的软件。 在主机上模拟目标系统操作时，该仿真器用来执行 C 源程序或汇编程序级的调试操作。 对 78K0R 使用 SM+可在硬件开发的独立基础上执行应用逻辑测试和性能测试，从而提供更高的开发效率和软件质量。 用于 78K0R 的 SM+应当结合设备文件 (DF781014)一起使用。 部件编号: μ SxxxxSM781000
ID78K0R-QB 集成调试器	该调试器支持 78K0R 微控制器的在线仿真器。ID78K0R-QB 是基于 Windows 的软件。 它改善了 C 兼容的调试功能，并使用集成窗口功能(结合源程序、分开显示和跟踪结果的存储器显示)来显示源程序的跟踪结果。 ID78K0R-QB 应当结合设备文件一起使用。 部件编号: μ SxxxxID78K0R-QB

备注 xxxx 部件编号根据所使用的主设备和操作系统而异。

μ Sxxxx SM781000

μ Sxxxx ID78K0R-QB

xxxx	主设备	操作系统	供应中介
AB17	PC-9800 系列，	Windows (日文版)	CD-ROM
BB17	兼容 IBM PC/AT	Windows (英文版)	

详细信息请联系:

中国区

MCU 技术支持热线:

电话: +86-400-700-0606 (普通话)

服务时间: 9:00-12:00, 13:00-17:00 (不含法定节假日)

网址:

<http://www.cn.necel.com/> (中文)

<http://www.necel.com/> (英文)

[北京]

日电电子(中国)有限公司

中国北京市海淀区知春路 27 号

量子芯座 7, 8, 9, 15 层

电话: (+86) 10-8235-1155

传真: (+86) 10-8235-7679

[深圳]

日电电子(中国)有限公司深圳分公司

深圳市福田区益田路卓越时代广场大厦 39 楼

3901, 3902, 3909 室

电话: (+86) 755-8282-9800

传真: (+86) 755-8282-9899

[上海]

日电电子(中国)有限公司上海分公司

中国上海市浦东新区银城中路 200 号

中银大厦 2409-2412 和 2509-2510 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230

[香港]

香港日电电子有限公司

香港九龙旺角太子道西 193 号新世纪广场

第 2 座 16 楼 1601-1613 室

电话: (+852) 2886-9318

传真: (+852) 2886-9022

2886-9044

上海恩益禧电子国际贸易有限公司

中国上海市浦东新区银城中路 200 号

中银大厦 2511-2512 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230

[成都]

日电电子(中国)有限公司成都分公司

四川省成都市二环路南三段 15 号

天华大厦 608 室

电话: (+86)28-8512-5224

传真: (+86)28-8512-5334

[长春]

日电电子(中国)有限公司长春分公司

吉林省长春市朝阳区

西安大路 727 号中银大厦 A 座 1609 室

电话: (+86)431-8859-7533 / 8859-8533

传真: (+86)431-8680-2944

[大连]

日电电子(中国)有限公司长春分公司

大连市中山路 88 号天安国际大厦 2701 室

电话: (+86)411-8230-8815 / 8230-8825

传真: (+86)411-8230-8835