

R7F0C80112ESP, R7F0C80212ESP

用户手册 硬件篇

8 位 单片机

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.

6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

注意事项

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，瑞萨电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，瑞萨电子已尽量做到合理注意，但是，瑞萨电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，瑞萨电子不承担任何责任。
3. 对于因使用本文档中的瑞萨电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，瑞萨电子不承担任何责任。本文档所记载的内容不应视为对瑞萨电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 用户不得更改、修改、复制或者以其他方式部分或全部地非法使用瑞萨电子的任何产品。对于用户或第三方因上述更改、修改、复制或以其他方式非法使用瑞萨电子产品的行为而遭受的任何损失，瑞萨电子不承担任何责任。
5. 瑞萨电子产品根据其质量等级分为两个等级：“标准等级”和“高质量等级”。每种瑞萨电子产品的推荐用途均取决于产品的质量等级，如下所示：

标准等级： 计算机、办公设备、通讯设备、测试和测量设备、视听设备、家用电器、机械工具、个人电子设备以及工业机器人等。

高质量等级： 运输设备（汽车、火车、轮船等）、交通控制系统、防灾系统、预防犯罪系统以及安全设备等。

瑞萨电子产品无意用于且未被授权用于可能对人类生命造成直接威胁的产品或系统及可能造成人身伤害的产品或系统（人工生命维持装置或系统、植埋于体内的装置等）中，或者可能造成重大财产损失的产品或系统（核反应堆控制系统、军用设备等）中。在将每种瑞萨电子产品用于某种特定应用之前，用户应先确认其质量等级。不得将瑞萨电子产品用于超出其设计用途之外的任何应用。对于用户或第三方因将瑞萨电子产品用于其设计用途之外而遭受的任何损害或损失，瑞萨电子不承担任何责任。

6. 使用本文档中记载的瑞萨电子产品时，应在瑞萨电子指定的范围内，特别是在最大额定值、电源工作电压范围、移动电源电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用瑞萨电子产品而产生的故障或损失，瑞萨电子不承担任何责任。
7. 虽然瑞萨电子一直致力于提高瑞萨电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，瑞萨电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当瑞萨电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。由于难于对微机软件单独进行评估，所以请用用户自行对最终产品或系统进行安全评估。
8. 关于环境保护方面的详细内容，例如每种瑞萨电子产品的环境兼容性等，请与瑞萨电子的营业部门联系。使用瑞萨电子产品时，请遵守对管制物质的使用或含量进行管理的所有相应法律法规（包括但不限于《欧盟 RoHS 指令》）。对于因用户未遵守相应法律法规而导致的损害或损失，瑞萨电子不承担任何责任。
9. 不可将瑞萨电子产品和技术用于或者嵌入日本国内或海外相应的法律法规所禁止生产、使用及销售的任何产品或系统中。也不可将本文档中记载的瑞萨电子产品或技术用于与军事应用或者军事用途有关的任何目的（如大规模杀伤性武器的开发等）。在将本文档中记载的瑞萨电子产品或技术进行出口时，应当遵守相应的出口管制法律法规，并按照上述法律法规所规定的程序进行。
10. 向第三方分销或处分产品或者以其他方式将产品置于第三方控制之下的瑞萨电子产品买方或分销商，有责任事先向上述第三方通知本文档规定的内容和条件；对于用户或第三方因非法使用瑞萨电子产品而遭受的任何损失，瑞萨电子不承担任何责任。
11. 在事先未得到瑞萨电子书面认可的情况下，不得以任何形式部分或全部转载或复制本文档。
12. 如果对本文档所记载的信息或瑞萨电子产品有任何疑问，或者用户有任何其他疑问，请向瑞萨电子的营业部门咨询。

（注 1） 瑞萨电子：在本文档中指瑞萨电子株式会社及其控股子公司。

（注 2） 瑞萨电子产品：指瑞萨电子开发或生产的任何产品。

关于 CMOS 器件的注意事项

① 输入引脚处的施加电压波形

输入噪声或由反射波引起的波形失真可能导致故障发生。如果由于噪声等影响，使CMOS器件的输入电压范围处于在 V_{IL} (MAX) 和 V_{IH} (MIN) 之间，器件可能发生故障。在输入电平固定时以及输入电平从 V_{IL} (MAX) 到 V_{IH} (MIN) 的过渡期间，要谨防颤振噪声进入器件。

② 未使用的输入引脚的处理

CMOS器件上未连接的输入端可能是故障源。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，从而导致故障。CMOS器件的工作方式与双极性或NMOS器件不同。CMOS器件的输入电平必须借助上拉或下拉电路固定于高电平或低电平。每一个未使用引脚只要有可能成为输出引脚时，都应该通过附加电阻连接到 V_{DD} 或GND。对未使用引脚的处理因器件而异，必须遵循与器件相关的规格和说明。

③ ESD防护措施

如果MOS器件周围有强电场，将会击穿氧化栅极，降低器件的工作性能。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。环境必须控制适当。如果空气干燥，应当使用加湿器。建议避免使用容易产生静电的绝缘体。半导体器件的存放和运输必须使用抗静电容器、静电屏蔽袋或导电材料包装。所有包括工作台和工作面的测试和测量工具必须良好接地。操作员应当佩戴手腕带以保证良好接地。不能用手直接接触半导体器件。对装配有半导体器件的PW板也应采取类似的静电防范措施。

④ 初始化之前的状态

上电并不一定定义MOS器件的初始状态。刚接通电源时，具有复位功能的MOS器件并没有被初始化。因此上电不能保证输出引脚的电平、输入/输出设置和寄存器的内容。器件在收到复位信号后才进行初始化。具有复位功能的器件在上电后必须立即进行复位操作。

⑤ 电源上电/断电序列

器件内部工作和外部接口使用不同电源的情况下，原则上应先在接通内部电源之后再接通外部电源。当关闭电源时，原则上先关闭外部电源再关闭内部电源。如果电源开关顺序相反，可能会对器件的内部元件施加电压，从而由于异常电流的流过而造成故障和降低元件的性能。须视具体器件和支配器件的相关规格来单独决定正确的上电/断电序列。

⑥ 断电状态期间的信号输入

不要在器件断电时输入信号或输入/输出上拉电源。因为输入信号或提供输入/输出上拉电源将引起电流注入，从而引起器件的误操作，并且此时流过器件的异常电流引起内部元件性能劣化。须视具体器件和支配器件的相关规格来单独决定断电状态期间的信号输入。

本手册的使用方法

读者	本手册适用于那些希望了解 R7F0C80112ESP, R7F0C80212ESP 的功能，并设计开发用于这些设备的应用系统的用户工程师。
目的	本手册用于帮助用户了解下述 结构 中所述的功能。
结构	R7F0C80112ESP, R7F0C80212ESP 手册分为两个部分：本手册和软件篇 (RL78 族共通)。

R7F0C80112ESP, R7F0C80212ESP 用户手册 硬件篇	RL78 族 用户手册 软件篇

• 引脚功能 • 内部模块功能 • 中断 • 其他片上外设功能 • 电特性	• CPU 功能 • 指令组 • 各指令的说明

如何阅读本手册	假定本手册的读者具备电气工程、逻辑电路以及单片机领域的常识 • 要了解大致的功能： → 请按照目录的顺序阅读本手册的内容。 • 寄存器格式的说明： → 关于尖括 (<>) 内的位数，其名称在汇编程序中定义为保留字，并在编译器中使用 #pragma sfr 指令定义为 sfr 变量。 • 要了解 R7F0C80112ESP, R7F0C80212ESP 单片机指令的详细情况： → 请参阅单独文件 RL78 族 用户手册 软件篇(R01US0015E)。

惯例

数据有效位:	高位在左，低位在右
有效低电平表示:	××× (在引脚或信号名称上有上划线)
注:	正文中使用 ‘注’ 标记的脚注
注意事项:	需特别注意的信息
备注:	补充信息
数字表示法:	二进制 …×××× 或 ××××B
	十进制 …××××
	十六进制 …××××H

相关文档

本手册中的相关文档可能包括初稿版本。但是，初稿版本并未特别标注。

器件相关文档

文档名称	文档编号
R7F0C80112ESP, R7F0C80212ESP 用户手册 硬件篇	本手册
RL78 族 用户手册 软件篇	R01US0015E

闪存编程器相关文档

文档名称	文档编号
PG-FP5 闪存编程器 用户手册	R20UT0008E

注意事项 上述列出的文件随时可能被更改，恕不另行通知。设计时，请确保使用各个文件的最新版本。

其他文件

文件名称	文件编号
瑞萨微型计算机 R78 族	R01CP0003E
半导体封装安装手册	注
半导体器件质量标准	C11531E
NEC 半导体器件可靠性质量管理	C10983E
预防静电放电(ESD)损坏半导体器件	C11892E
信赖性手册	R51ZZ0001E

注 请参阅“半导体封装安装手册”网站 (<http://www.renesas.com/products/package/manual/index.jsp>).

注意事项 上述列出的文件随时可能被更改，恕不另行通知。设计时，请确保使用各个文件的最新版本。

所有商标及注册商标分别归属于其所有者。
本用户手册仅为参考译文，对应的日文版和英文版具有正式效力。

目录

第一章 概述.....	1
1.1 特点.....	1
1.2 编号列表	2
1.3 引脚配置 (顶视图).....	3
1.4 引脚名称	3
1.5 框图.....	4
1.6 功能概述	5
第二章 引脚功能.....	6
2.1 端口功能	6
2.2 端口以外的引脚功能	7
2.3 引脚的输入/输出电路和未使用引脚的推荐连接	8
第三章 CPU 架构	10
3.1 地址空间	11
3.1.1 内部程序存储器空间	13
3.1.2 镜像区域	15
3.1.3 内部数据存储空间	16
3.1.4 特殊功能寄存器 (SFR) 区域.....	16
3.1.5 扩展特殊功能寄存器 (2nd SFR: 第二特殊功能寄存器) 区域.....	16
3.1.6 数据存储器寻址	16
3.2 处理器寄存器	18
3.2.1 控制寄存器	18
3.2.2 通用寄存器	20
3.2.3 ES 和 CS 寄存器.....	22
3.2.4 特殊功能寄存器(SFR).....	23
3.2.5 扩展特殊功能寄存器 (2nd SFR: 第二特殊功能寄存器)	26
3.3 指令地址寻址	29
3.3.1 相对寻址	29
3.3.2 立即寻址	29
3.3.3 表间接寻址	30
3.3.4 寄存器直接寻址	30
3.4 处理数据地址的寻址	31
3.4.1 隐含寻址	31
3.4.2 寄存器寻址	31
3.4.3 直接寻址	32
3.4.4 短直接寻址	33

3.4.5 SFR 寻址	34
3.4.6 寄存器间接寻址	35
3.4.7 基址寻址	36
3.4.8 基址变址寻址	40
3.4.9 堆栈寻址	41
第四章 端口功能.....	44
4.1 端口功能	44
4.2 端口配置	44
4.2.1 端口 0	44
4.2.2 端口 4	44
4.2.3 端口 12	44
4.2.4 端口 13	44
4.3 控制端口功能的寄存器.....	45
4.3.1 端口模式寄存器 0, 4 (PM0, PM4)	46
4.3.2 端口寄存器 0, 4, 12, 13 (P0, P4, P12, P13)	47
4.3.3 上拉电阻选择寄存器 0, 4, 12 (PU0, PU4, PU12)	48
4.3.4 端口输出模式寄存器 0 (POM0)	49
4.3.5 端口模式控制寄存器 0 (PMC0)	50
4.3.6 外围输入/输出重定向寄存器(PIOR)	51
4.4 端口功能的操作	52
4.4.1 输入/输出端口的写入	52
4.4.2 输入/输出端口的读取	52
4.4.3 输入/输出端口的运算	52
4.5 使用复用功能时相关寄存器的设置	53
4.5.1 使用复用功能时的基本思路	53
4.5.2 不使用复用功能的输出功能时寄存器的设置	54
4.5.3 使用端口功能以及复用功能的寄存器设置例	54
4.6 使用端口功能时的注意事项	57
4.6.1 端口寄存器 n (Pn)的位操作指令的相关注意事项	57
4.6.2 引脚设置时的注意事项	58
第五章 时钟发生电路	59
5.1 时钟发生电路的功能	59
5.2 时钟发生电路的配置	60
5.3 控制时钟发生电路的寄存器	62
5.3.1 外围允许寄存器 0 (PER0)	63
5.3.2 高速片上振荡器频率选择寄存器(HOCODIV)	64
5.4 系统时钟振荡电路	65
5.4.1 高速片上振荡器	65
5.4.2 低速片上振荡器	65

5.5 时钟发生电路的操作	65
5.6 控制时钟	67
5.6.1 高速片上振荡器的设置示例	67
5.6.2 CPU 时钟状态转换图	68
第六章 定时器阵列单元	70
6.1 定时器阵列单元功能	71
6.1.1 单通道操作功能	71
6.1.2 多通道联动操作功能	72
6.1.3 8 位定时器操作功能（仅限通道 1）	73
6.2 定时器阵列单元的配置	74
6.2.1 定时器/计数器寄存器 0n (TCR0n)	77
6.2.2 定时器数据寄存器 0n (TDR0n)	79
6.3 控制定时器阵列单元的寄存器	81
6.3.1 外围允许寄存器 0 (PER0)	82
6.3.2 定时器时钟选择寄存器 0 (TPS0)	83
6.3.3 定时器模式寄存器 0n (TMR0n)	85
6.3.4 定时器状态寄存器 0n (TSR0n)	89
6.3.5 定时器通道允许状态寄存器 0 (TE0, TEH0 (8 位模式))	90
6.3.6 定时器通道开始寄存器 0 (TS0, TSH0 (8 位模式))	91
6.3.7 定时器通道停止寄存器 0 (TT0, TTH0 (8 位模式))	92
6.3.8 定时器输出允许寄存器 0 (TOE0)	93
6.3.9 定时器输出寄存器 0 (TO0)	94
6.3.10 定时器输出电平寄存器 0 (TOL0)	95
6.3.11 定时器输出模式寄存器 0 (TOM0)	96
6.3.12 噪声滤波器允许寄存器 1 (NFEN1)	97
6.3.13 端口模式寄存器 0 (PM0)	98
6.4 定时器阵列单元的基本规则	99
6.4.1 多通道联动操作功能的基本规则	99
6.4.2 8 位定时器操作功能的基本规则（仅限通道 1）	100
6.5 计数器的操作	101
6.5.1 计数时钟(f_{TCLK})	101
6.5.2 计数器的开始时序	103
6.5.3 计数器的操作	104
6.6 通道输出（TO0n 引脚）控制	109
6.6.1 TO0n 引脚输出电路配置	109
6.6.2 TO0n 引脚输出设置	110
6.6.3 通道输出操作的注意事项	111
6.6.4 TO0n 位的集中操作	115
6.6.5 计数操作开始时的定时器中断和 TO0n 引脚输出	116
6.7 定时器阵列单元的单通道操作功能	117

6.7.1 作为间隔定时器/方波输出的操作	117
6.7.2 作为外部事件计数器操作	123
6.7.3 作为分频器的操作(仅限通道 0)	128
6.7.4 作为输入脉冲间隔测量的操作	132
6.7.5 输入信号高/低电平宽度测量的操作	137
6.7.6 作为延迟计数器的操作	141
6.8 定时器阵列单元的多通道联动操作功能	146
6.8.1 作为单触发脉冲输出功能操作	146
6.8.2 作为 PWM 功能的操作	153
6.9 使用定时器阵列单元时的注意事项	160
6.9.1 使用定时器输出时的注意事项	160
第七章 时钟输出/蜂鸣器输出控制电路	161
7.1 时钟输出/蜂鸣器输出控制电路的功能	161
7.2 时钟输出/蜂鸣器输出控制电路的配置	162
7.3 控制时钟输出/蜂鸣器输出控制电路的寄存器	162
7.3.1 时钟输出选择寄存器 0 (CKS0)	162
7.3.2 端口模式寄存器 0, 4 (PM0, PM4)	164
7.4 时钟输出/蜂鸣器输出控制电路的操作	165
7.4.1 作为输出引脚的操作	165
第八章 看门狗定时器	166
8.1 看门狗定时器的功能	166
8.2 看门狗定时器的配置	167
8.3 控制看门狗定时器的寄存器	168
8.3.1 看门狗定时器允许寄存器(WDTE)	168
8.4 看门狗定时器的操作	169
8.4.1 控制看门狗定时器的操作	169
8.4.2 设置看门狗定时器的溢出时间	170
第九章 A/D 转换器	171
9.1 A/D 转换器的功能	171
9.2 A/D 转换器的配置	173
9.3 A/D 转换器中使用的寄存器	175
9.3.1 外围允许寄存器 0 (PER0)	175
9.3.2 A/D 转换器模式寄存器 0 (ADM0)	176
9.3.3 A/D 转换器模式寄存器 2 (ADM2)	180
9.3.4 A/D 转换结果高位存储寄存器(ADCRH)	180
9.3.5 A/D 转换结果低位存储寄存器(ADCRL)	181
9.3.6 模拟输入通道选择寄存器(ADS)	182

9.3.7 端口模式控制寄存器 0 (PMC0).....	183
9.3.8 端口模式寄存器 0 (PM0).....	184
9.4 A/D 转换器的转换操作.....	185
9.5 输入电压和转换结果.....	187
9.6 A/D 转换器的操作模式.....	188
9.7 A/D 转换器的设置流程图.....	189
9.8 如何阅读 A/D 转换器特性表.....	190
9.8.1 分辨率	190
9.8.2 总误差	190
9.8.3 量化误差	190
9.8.4 零刻度误差	190
9.8.5 满刻度误差	191
9.8.6 积分线性误差.....	191
9.8.7 微分线性误差.....	191
9.8.8 转换时间	191
9.8.9 采样时间	191
9.9 A/D 转换器的注意事项.....	192
9.9.1 STOP 模式下的操作电流	192
9.9.2 ANI0 至 ANI3 引脚的输入范围	192
9.9.3 冲突操作	192
9.9.4 噪声对策	192
9.9.5 模拟输入(ANIn)引脚	193
9.9.6 模拟输入(ANIn)引脚的输入阻抗.....	193
9.9.7 中断请求标志(ADIF)	194
9.9.8 A/D 转换开始后的初次转换结果.....	194
9.9.9 读取 A/D 转换结果寄存器(ADCRH, ADCRL)	194
9.9.10 内部等效电路.....	195
9.9.11 开始 A/D 转换器的操作	195
第十章 串行阵列单元	196
10.1 串行阵列单元的功能	197
10.1.1 3 线串行输入/输出(CSI00)	197
10.1.2 UART (UART0)	198
10.2 串行阵列单元的配置	199
10.2.1 移位寄存器	201
10.2.2 串行数据寄存器 0n (SDR0nH, SDR0nL)	201
10.3 控制串行阵列单元的寄存器	203
10.3.1 外围允许寄存器 0 (PER0).....	204
10.3.2 串行时钟选择寄存器 0 (SPS0).....	205
10.3.3 串行模式寄存器 0n (SMR0nH, SMR0nL).....	206
10.3.4 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)	208

10.3.5 串行数据寄存器 0n (SDR0nH, SDR0nL)	211
10.3.6 串行标志清除触发寄存器 0n (SIR0n).....	212
10.3.7 串行状态寄存器 0n (SSR0n).....	213
10.3.8 串行通道开始寄存器 0 (SS0)	215
10.3.9 串行通道停止寄存器 0 (ST0)	216
10.3.10 串行通道允许状态寄存器 0 (SE0).....	217
10.3.11 串行输出允许寄存器 0 (SOE0)	218
10.3.12 串行输出寄存器 0 (SO0).....	219
10.3.13 串行时钟输出寄存器(CKO0).....	220
10.3.14 串行输出电平寄存器 0 (SOL0).....	221
10.3.15 噪声滤波器允许寄存器 0 (NFEN0).....	222
10.3.16 输入切换控制寄存器(ISC).....	223
10.3.17 端口输出模式寄存器 0 (POM0).....	224
10.3.18 端口模式寄存器 0 (PM0).....	225
10.4 操作停止模式	226
10.4.1 以单元单位停止操作	226
10.4.2 以通道单位停止操作	227
10.5 3 线串行输入/输出(CSI00)通信的操作	228
10.5.1 主发送.....	229
10.5.2 主接收.....	239
10.5.3 主发送/接收	248
10.5.4 从发送.....	258
10.5.5 从接收.....	268
10.5.6 从发送/接收	275
10.5.7 计算传送时钟频率.....	285
10.5.8 3 线串行输入/输出(CSI00)通信过程中发生错误时的处理步骤	286
10.6 UART(UART0)通信的操作	287
10.6.1 UART 发送.....	288
10.6.2 UART 接收.....	298
10.6.3 计算波特率	305
10.6.4 UART(UART0)通信过程中发生错误时的处理步骤.....	308
第十一章 中断功能	309
11.1 中断功能的类型	309
11.2 中断源和配置	309
11.3 控制中断功能的寄存器.....	313
11.3.1 中断请求标志寄存器 (IF0L, IF0H).....	314
11.3.2 中断屏蔽标志寄存器 (MK0L, MK0H)	315
11.3.3 优先级选择标志寄存器 (PR00L, PR00H, PR10L, PR10H)	316
11.3.4 外部中断上升沿允许寄存器 (EGP0), 外部中断下降沿允许寄存器 (EGN0).....	318
11.3.5 程序状态字(PSW).....	319

11.4 中断处理的操作	320
11.4.1 可屏蔽中断请求受理	320
11.4.2 软件中断请求受理	323
11.4.3 嵌套中断处理	323
11.4.4 保留中断请求	327
第十二章 按键中断功能	328
12.1 按键中断的功能	328
12.2 按键中断的配置	328
12.3 控制按键中断的寄存器	330
12.3.1 按键返回控制寄存器(KRCTL)	330
12.3.2 按键返回模式寄存器(KRM0)	331
12.3.3 按键返回标志寄存器 (KRF)	331
12.3.4 端口模式寄存器 0, 4 (PM0, PM4)	332
12.4 按键中断功能的操作	333
12.4.1 不使用按键中断标志时 (KRMD = 0)	333
12.4.2 使用按键中断标志时 (KRMD = 1)	334
第十三章 待机功能	337
13.1 概要	337
13.2 待机功能工作	338
13.2.1 HALT 模式	338
13.2.2 STOP 模式	340
第十四章 复位功能	343
14.1 复位操作的时序	345
14.2 复位期间的工作状态	347
14.3 确认复位源的寄存器	351
14.3.1 复位控制标志寄存器(RESF)	351
第十五章 可选择上电复位电路	353
15.1 可选择上电复位电路的功能	353
15.2 可选择上电复位电路的配置	354
15.3 可选择上电复位电路的操作	355
15.4 可选择上电复位电路的注意事项	356
第十六章 选项字节	357
16.1 选项字节的功能	357
16.1.1 用户选项字节(000C0H 至 000C2H)	357
16.1.2 片上调试选项字节 (000C3H)	357

16.2 用户选项字节的格式	358
16.3 片上调试选项字节的格式	360
16.4 选项字节的设置	361
第十七章 闪存 362	
17.1 使用闪存编程器进行串行编程	363
17.1.1 编程环境	364
17.1.2 通信模式	364
17.2 使用外部器件 (UART 内置) 进行编程	365
17.2.1 编程环境	365
17.2.2 通信方式	366
17.3 板上引脚连接	367
17.3.1 P40/TOOL0 引脚	367
17.3.2 RESET 引脚	367
17.3.3 端口引脚	367
17.3.4 电源	367
17.4 串行编程方法	368
17.4.1 串行编程的步骤	368
17.4.2 闪存的编程模式	369
17.4.3 通信模式	370
17.4.4 通信命令	370
17.5 使用 PG-FP5 时各指令的处理时间 (参考值)	370
第十八章 片上调试功能	371
18.1 连接 E1 片上调试仿真器	371
18.2 片上调试安全 ID	373
18.3 用户资源的预留	373
第十九章 十进制调整(BCD)电路	374
19.1 十进制调整电路的功能	374
19.2 十进制调整电路使用的寄存器	374
19.2.1 BCD 校正结果寄存器(BCDADJ)	374
19.3 十进制调整电路的操作	375
第二十章 指令集	377
20.1 操作列表使用规则	378
20.1.1 操作数标识符和标识方法	378
20.1.2 操作栏的说明	379
20.1.3 标志栏的说明	380
20.1.4 PREFIX 指令	380

20.2 操作列表	381
第二十一章 电特性.....	398
21.1 最大绝对额定值	398
21.2 振荡电路特性	399
21.2.1 片上振荡器特性	399
21.3 DC 特性	400
21.3.1 引脚特性	400
21.3.2 电源电流特性	402
21.4 AC 特性	403
21.5 串行接口特性	404
21.5.1 串行阵列单元	404
21.6 模拟特性	408
21.6.1 A/D 转换器特性	408
21.6.2 数据保持电源电压特性	408
21.6.3 SPOR 电路特性	409
21.6.4 电源电压上升倾斜特性	409
21.7 闪存编程特性	409
21.8 专用闪存编程器通信 (UART)	409
21.9 进入闪存编程模式的时序	410
第二十二章 封装尺寸图.....	411
附录 A 修订记录.....	412

第一章 概述

1.1 特点

- 最短指令执行时间可在高速(0.05 μ s @ 20 MHz 运行时)至超低速(0.8 μ s @ 1.25 MHz 运行时)之间更改
- 通用寄存器: 8 位 $\times$ 8 个寄存器
- ROM: 1 KB/2 KB, RAM: 128 字节/256 字节
- 可选择高速片上振荡器时钟: 20/10/5/2.5/1.25 MHz (典型值)
- 内置单电源闪存
- 片上调试功能
- 内置可选择的上电复位(SPOR)电路
- 片上看门狗定时器(可在专用低速片上振荡器时钟下运行)
- 片上按键中断功能: 6 个按键中断输入引脚
- 片上时钟输出/蜂鸣器输出控制电路
- 片上 BCD 调整
- 输入/输出端口: 8
- 定时器
 - 8/16 位定时器: 2 通道
 - 看门狗定时器: 1 通道
- 串行接口
 - CSI: 1 通道
 - UART: 1 通道
- 8/10 位分辨率 A/D 转换器: 4 通道
- 待机功能: HALT 或 STOP 模式
- 电源电压: $V_{DD} = 2.4$ 至 5.5 V
(由于包括了可选择上电复位(SPOR)电路的检测电压(V_{SPOR}), 请在 2.57(TYP.)至 5.5 V 的电压范围内使用)
- 运行环境温度: $T_A = -40$ 至 $+85^{\circ}\text{C}$
- ROM, RAM 容量

Flash ROM	RAM	产品名称
1KB	128B	R7F0C80112ESP
2KB	256B	R7F0C80212ESP

1.2 编号列表

图 1-1. R7F0C80112ESP, R7F0C80212ESP 的编号及内存大小、封装

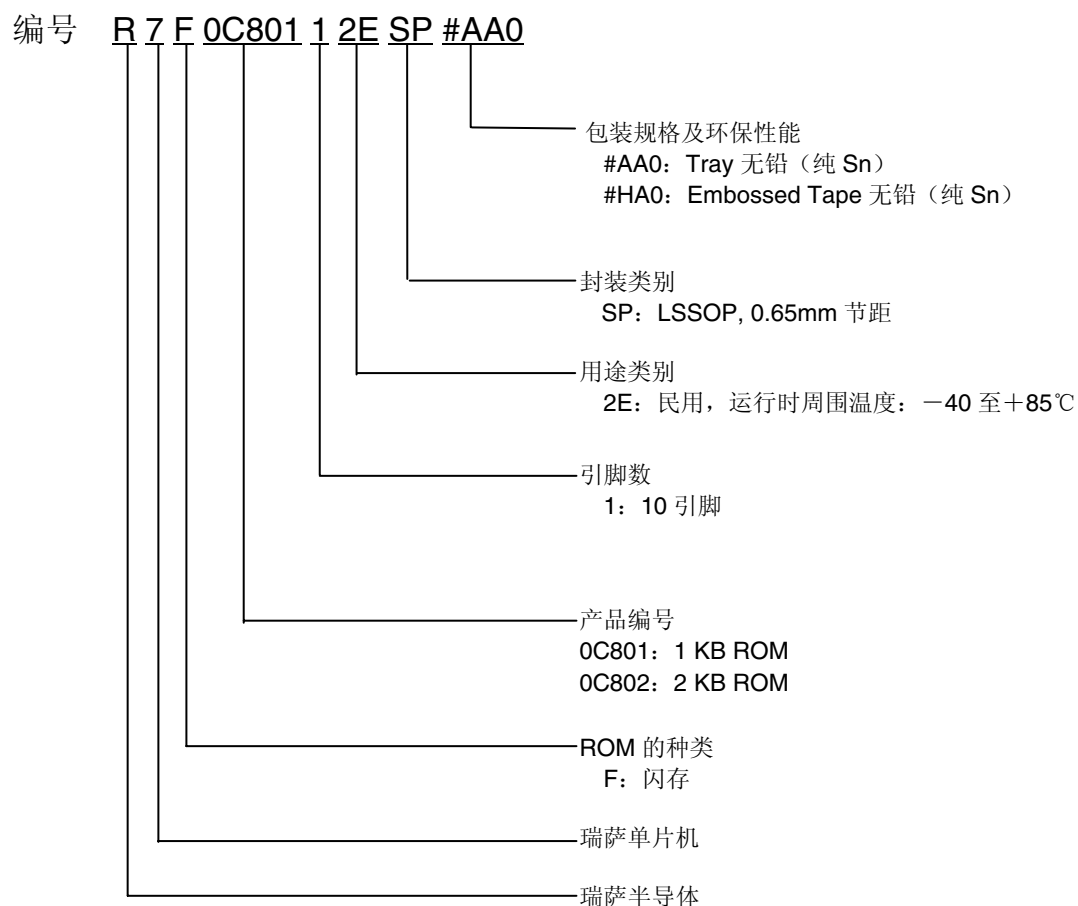
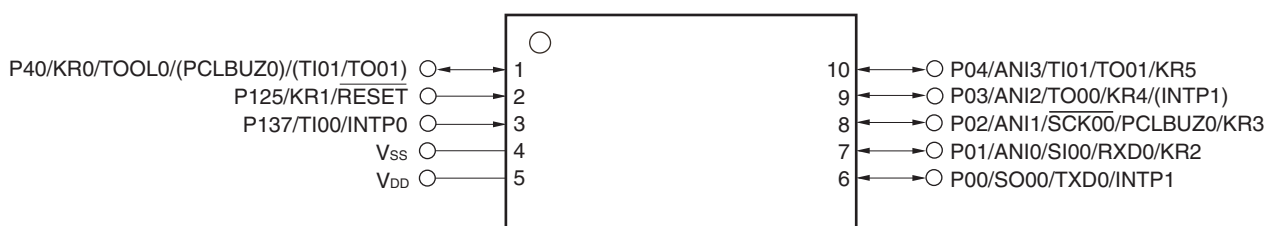


表 1-1. 订货编号列表

引脚数	封装	Flash ROM	RAM	包装规格及环保性能	订货编号
10引脚	10引脚塑封LSSOP 4.4×3.6 mm, 0.65 mm节距	1 KB	128 B	Tray无铅 (纯Sn)	R7F0C80112ESP#AA0
				Embossed Tape无铅 (纯Sn)	R7F0C80112ESP#HA0
		2 KB	256 B	Tray无铅 (纯Sn)	R7F0C80212ESP#AA0
				Embossed Tape无铅 (纯Sn)	R7F0C80212ESP#HA0

1.3 引脚配置 (顶视图)

- 10 引脚塑封 SSOP (4.4 x 3.6)

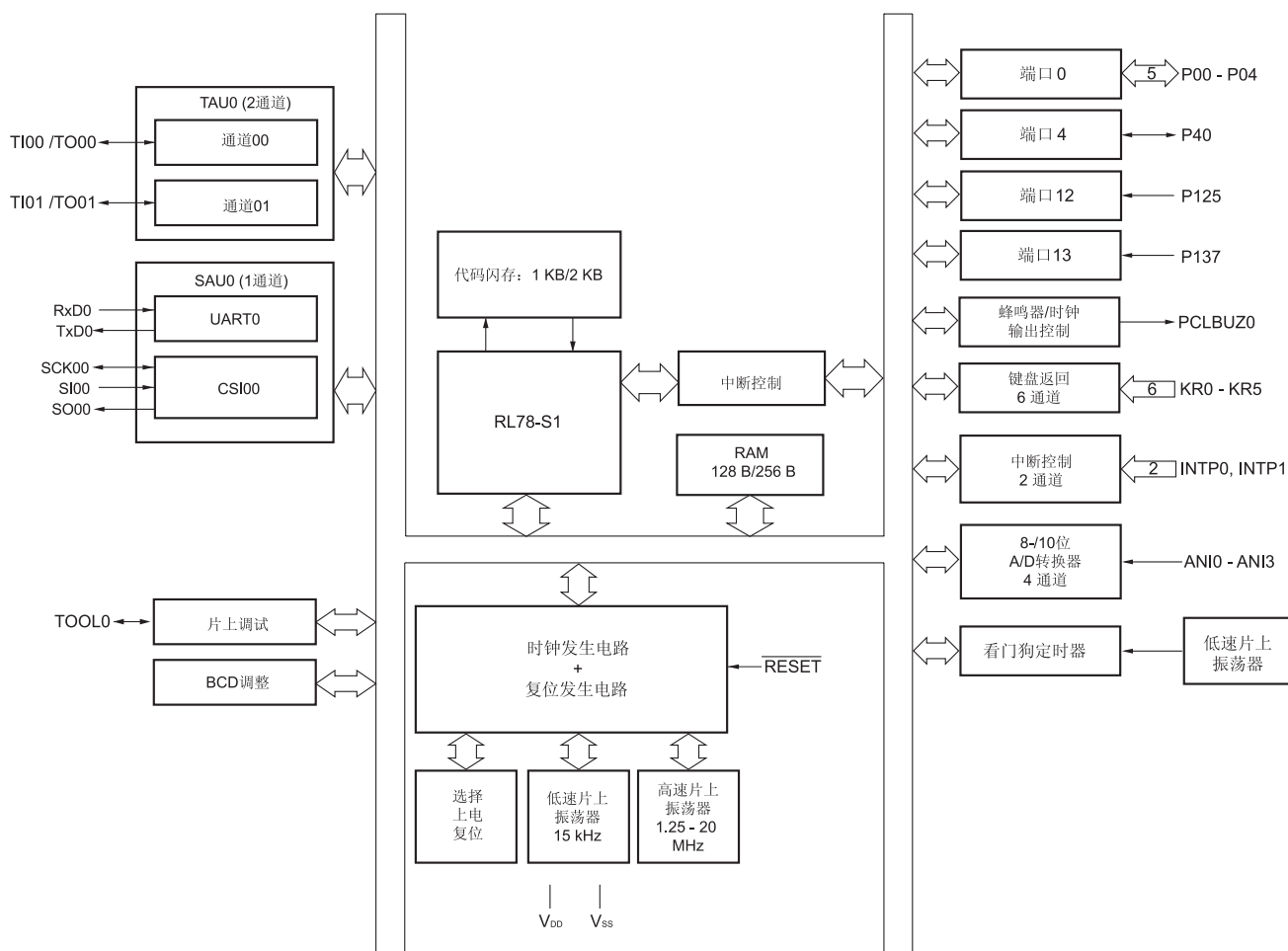


- 备注 1.** 有关引脚名称, 请参阅 **1.4 引脚名称**。
- 2.** 可以通过设置外围输入/输出重定向寄存器 0 (PIOR0)来指定上表括号内的功能。

1.4 引脚名称

ANI0 至 ANI3	: 模拟输入
INTP0, INTP1	: 外部中断输入
KR0 至 KR5	: 按键返回
P00 至 P04	: 端口 0
P40	: 端口 4
P125	: 端口 12
P137	: 端口 13
PCLBUZ0	: 可编程的时钟输出/蜂鸣器输出
RESET	: 复位
RxD0	: 接收数据
SCK00	: 串行时钟输入/输出
SI00	: 串行数据输入
SO00	: 串行数据输出
TI00, TI01	: 定时器输入
TO00, TO01	: 定时器输出
TOOL0	: 工具用数据输入/输出
TxD0	: 发送数据
VDD	: 电源
VSS	: 接地

1.5 框图



1.6 功能概述

本概述描述的是将外围输入/输出重定向寄存器(PIOR)设置为 00H 时的功能。

项目		R7F0C80212ESP	R7F0C80112ESP
代码闪存		2 KB	1 KB
RAM		256 B	128 B
主系统时钟	高速片上振荡器时钟	1.25 至 20 MHz ($V_{DD} = 2.7$ 至 5.5 V) 1.25 至 5 MHz ($V_{DD} = 2.4$ 至 5.5 V)	
低速片上振荡器时钟		15 kHz (典型值)	
通用寄存器		8 位寄存器 × 8 个	
最短指令执行时间		0.05 μ s (20 MHz 操作)	
指令集		- 数据传送 (8 位) - 加减法/逻辑运算 (8 位) - 乘法 (8 位 × 8 位) - 循环, 桶式移位和位操作 (置位, 复位, 测试和布尔运算)等	
输入/输出 端口	总数	8	
	CMOS 输入/输出	6 (N 沟开漏输出(V_{DD} 耐压): 1)	
	CMOS 输入	2	
定时器	16 位定时器	2 通道	
	看门狗定时器	1 通道	
	定时器输出	2 通道 (PWM 输出: 1)	
时钟输出/蜂鸣器输出		1	
		2.44 kHz 至 10 MHz: (外围硬件时钟: $f_{MAIN} = 20$ MHz 操作)	
8/10 位分辨率 A/D 转换器		4 通道	
串行接口		CSI: 1 通道/UART: 1 通道	
向量中断 源	内部	8	
	外部	3	
按键中断		6	
复位		- 通过 RESET 引脚复位 - 通过看门狗定时器的内部复位 - 通过可选上电复位的内部复位 - 通过执行非法指令的内部复位^{注1}	
可选择上电复位电路		- 检测电压 - 上升 (V_{SPOR}): 2.57 V/2.90 V/4.28 V (TYP.) - 下降 (V_{SPDR}): 2.40 V/2.70 V/4.00 V (MIN.)	
片上调试功能		有	
电源电压		$V_{DD} = 2.4$ 至 5.5 V ^{注2}	
运行环境温度		$T_A = -40$ 至 +85 °C	

注1. 执行指令代码 FFH 时, 产生非法指令。通过电路内置仿真器或片上调试仿真器进行仿真时, 不会因执行非法指令发生内部复位。

2. 由于包括了可选择上电复位(SPOR)电路的检测电压(V_{SPOR}), 请在 2.57(TYP.)至 5.5 V 的电压范围内使用。

第二章 引脚功能

2.1 端口功能

输入/输出、缓冲器和上拉电阻的设置复用功能中也有效。

功能名称	输入/输出	功能	复位后	复用功能
P00	输入/输出	端口 0。 5 位输入/输出端口。 可以以 1 位为单位指定为输入/输出。 可被设置为模拟输入。某引脚用于输入时，在端口模式控制寄存器 0 (PMC0) 中指定为数字或模拟。该寄存器可以以 1 位为单位进行指定。 P00 的输出可被设置为 N 沟开漏(V _{DD} 耐压)。 输入端口中可以通过软件设置(以 1 位为单位)使用片上上拉电阻。	输入端口	SO00/TXD0/INTP1
P01			模拟信号 输入端口	ANI0/SI00/RXD0/KR2
P02				ANI1/SCK00/PCLBUZ0/KR3
P03				ANI2/TO00/KR4/(INTP1)
P04				ANI3/TI01/TO01/KR5
P40	输入/输出	端口 4。 1 位输入/输出端口。 可以指定为输入/输出。 输入端口中可以通过软件设置使用片上上拉电阻。	输入端口	KR0/TOOL0/(PCLBUZ0)/(TI01/TO01)
P125	输入	端口 12 1 位输入专用端口。 可通过软件设置使用片上上拉电阻。	输入端口	KR1/RESET
P137	输入	端口 13 1 位输入专用端口	输入端口	TI00/INTP0

备注 可以通过设置外围输入/输出重定向寄存器 0 (PIOR0) 来指定上表括号内的功能。
 详情请参阅图 4-6. 外围输入/输出重定向寄存器(PIOR)的格式。

2.2 端口以外的引脚功能

功能名称	输入/输出	功能						
ANIO 或 ANI3	输入	A/D 转换器的模拟输入引脚 (参阅图 9-22. 模拟输入引脚的连接)						
INTP0, INTP1	输入	外部中断请求输入 指定有效沿：上升沿、下降沿或上升下降的双边沿						
KR0 或 KR5	输入	按键中断输入						
PCLBUZ0	输出	时钟/蜂鸣器输出						
$\overline{\text{RESET}}$	输入	外部复位输入 未使用外部复位引脚时，保持开路，或者在 $\text{PORTSELB} = 1$ 时将其连接至 V_{DD} 。						
RxD0	输入	UART0 串行数据输入						
TxD0	输出	UART0 串行数据输出						
$\overline{\text{SCK00}}$	输入/输出	CSI00 时钟输入/输出						
SI00	输入	CSI00 串行数据输入						
SO00	输出	CSI00 串行数据输出						
TI00, TI01	输入	向 16 位定时器 00 和 01 输入外部计数时钟/捕捉触发						
TO00, TO01	输出	16 位定时器 00 和 01 的定时器输出引脚						
V_{DD}	—	正电源						
V_{SS}	—	接地电位						
TOOL0	输入/输出	闪存编程器/调试器用数据输入/输出 启动后的操作模式由解除复位时的 TOOL0 引脚状态决定。当允许片上调试时通过外部电阻将该引脚连接到 V_{DD} (禁止下拉)。 <table><tr><th>TOOL0</th><th>操作模式</th></tr><tr><td>V_{DD}</td><td>正常操作模式</td></tr><tr><td>0 V</td><td>闪存的编程模式</td></tr></table> 有关详情，请参阅 17.4.2 闪存的编程模式。	TOOL0	操作模式	V_{DD}	正常操作模式	0 V	闪存的编程模式
TOOL0	操作模式							
V_{DD}	正常操作模式							
0 V	闪存的编程模式							

2.3 引脚的输入/输出电路和未使用引脚的推荐连接

表 2-1 和 2-2 表示的是引脚的输入/输出电路的类型和未使用引脚的推荐连接。

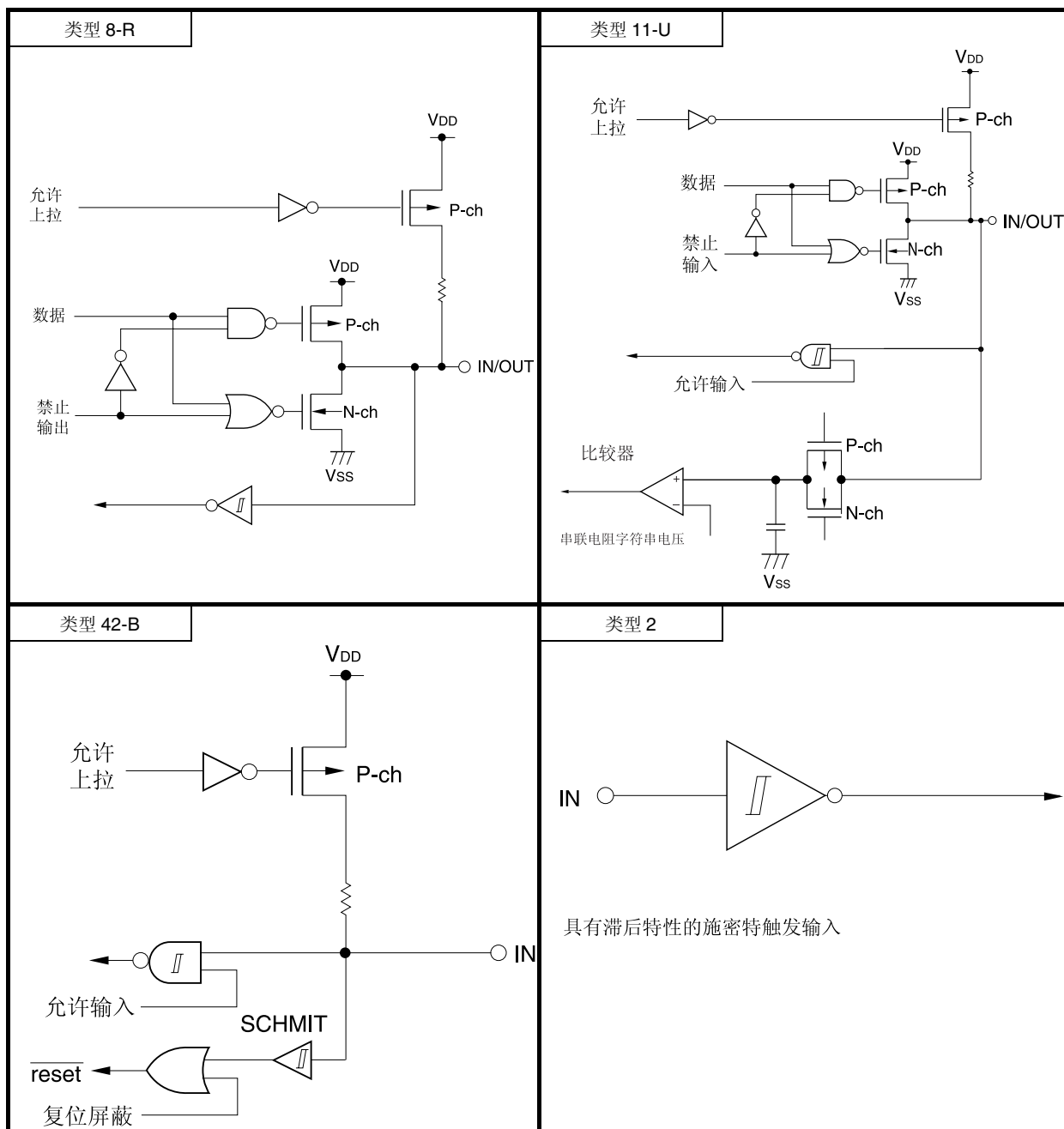
表 2-1. 未使用引脚的连接

引脚名称	输入/输出电路类型	输入/输出	未使用引脚的推荐连接
P00/SO00/TXD0/INTP1	8-R	输入/输出	输入： 通过电阻单独连接至 V _{DD} 或 V _{SS} 。 输出： 保持开路。
P01/ANI0/SI00/RXD0/KR2	11-U		
P02/ANI1/SCK00/PCLBUZ0/KR3			
P03/ANI2/TO00/KR4/(INTP1)			
P04/ANI3/TI01/TO01/KR5			
P40/KR0/TOOL0/(PCLBUZ0)/ (TI01/TO01)	8-R		通过电阻单独连接至 V _{DD} 。
P125/KR1/RESET	42-B	输入	保持开路，或着当 PORTSELB = 1 时连接至 V _{DD} 。
P137/INTP0	2		通过电阻单独连接至 V _{DD} 或 V _{SS} 。

备注 可以通过设置外围输入/输出重定向寄存器 0 (PIOR0)来指定上表括号内的功能。

详情请参阅图 4-6. 外围输入/输出重定向寄存器(PIOR)的格式。

图 2-1. 各引脚的输入/输出电路列表



第三章 CPU 架构

R7F0C80112ESP, R7F0C80212ESP安装有RL78-S1核。

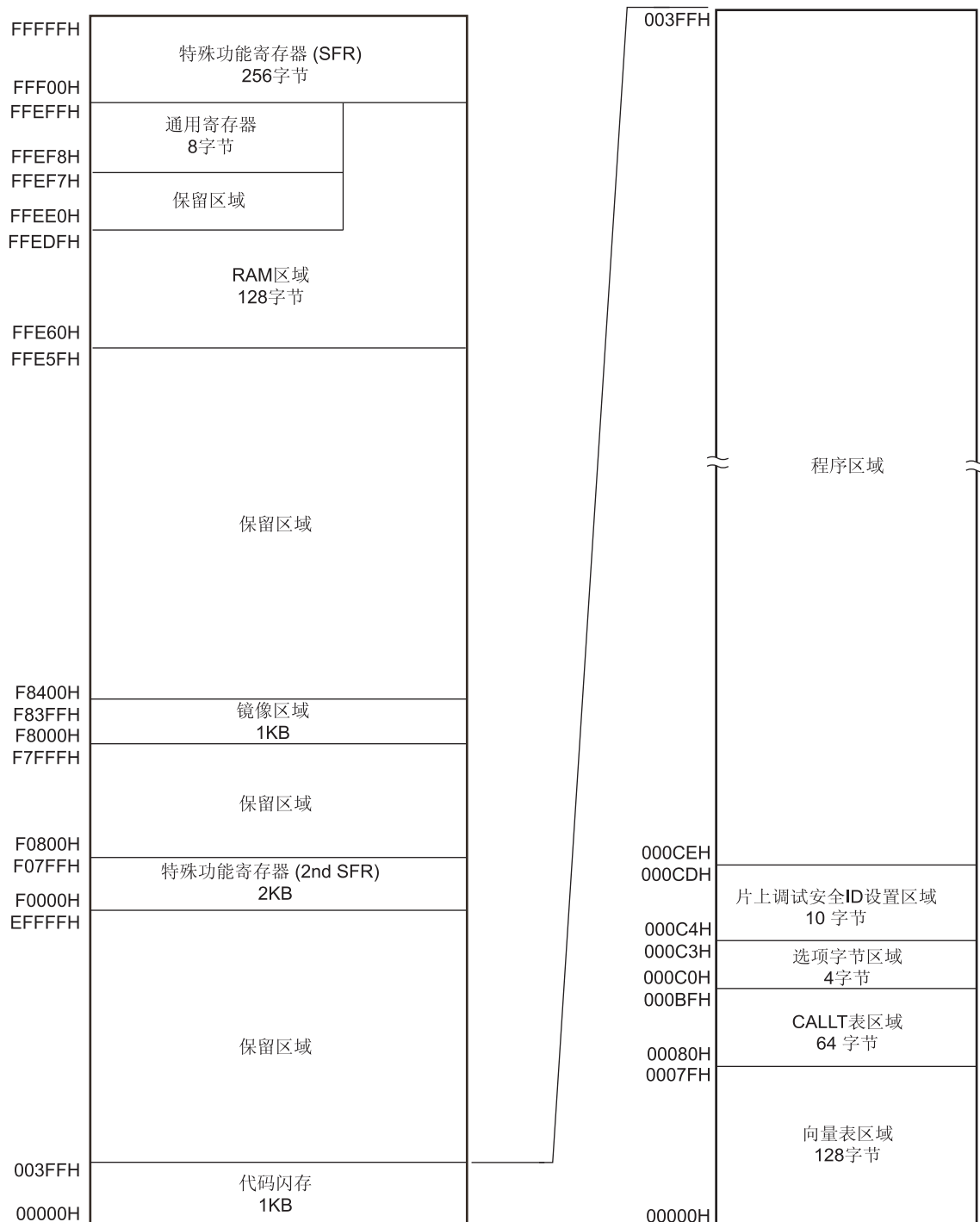
RL78-S1核的特徵如下所示。

- 3线程的CISC架构
- 地址空间：1MB
- 通用寄存器：8位寄存器 x 8个
- 指令和RL78-S2核的指令相同。但是，RL78-S1核与RL78-S2核中，以下指令所用时钟数不同。详情请参阅**第二十章 指令集**。
 - 16数据传送（MOVW, XCHW, ONEW, CLRW）
 - 16位运算（ADDW, SUBW, CMPW）
 - 乘法运算（MULU）
 - 16位加减法（INCW, DECW）
 - 16位移位（SHRW, SHLW, SARW）
 - 16位旋转（ROLWC）
 - 调用返回（CALL, CALLT, BRK, RET, RETI, RETB）
 - 堆栈操作（PUSH, POP, MOVW, ADDW, SUBW）

3.1 地址空间

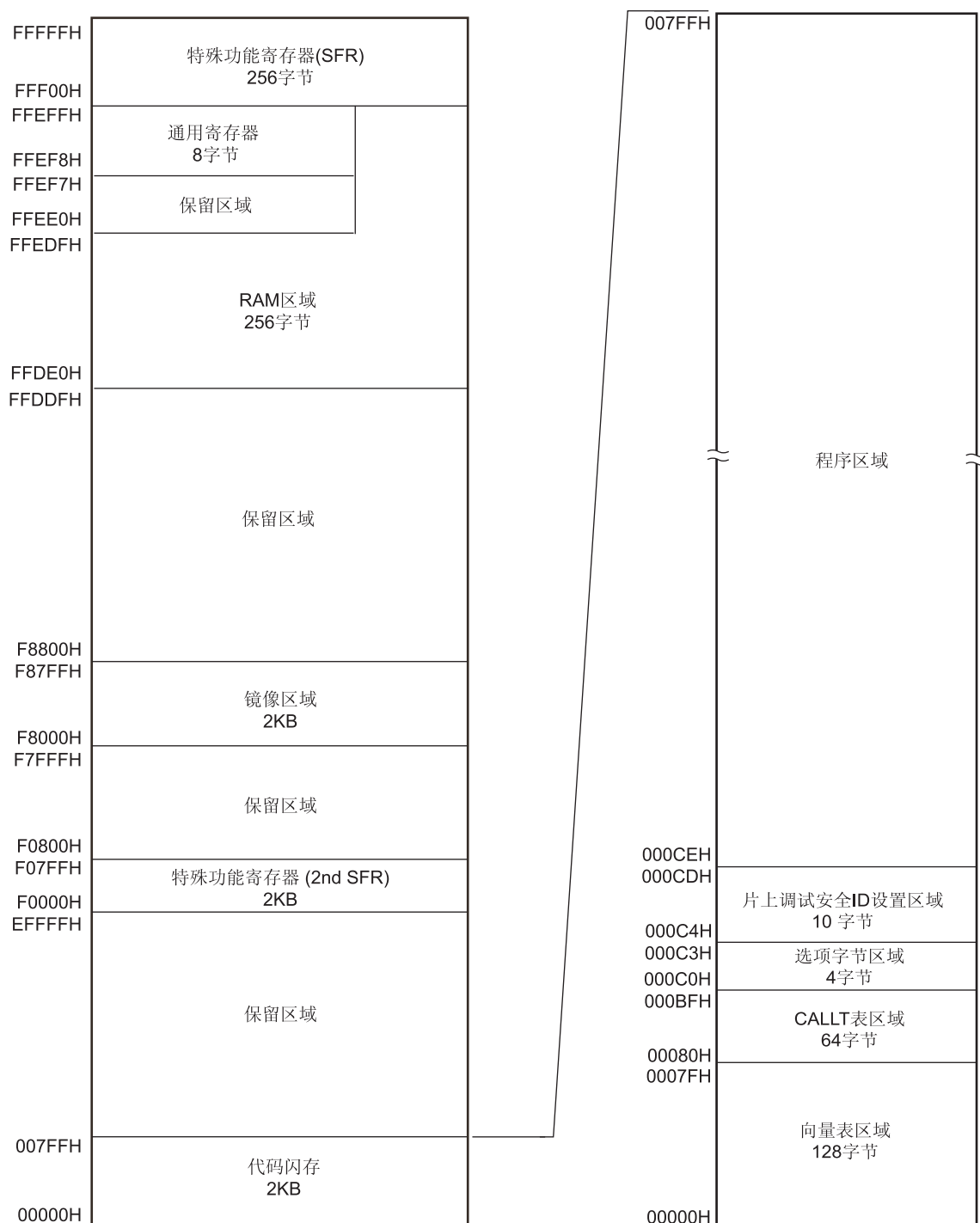
R7F0C80112ESP, R7F0C80212ESP 可以存取 1 MB 的地址空间。存储器映射图如图 3-1 和图 3-2 所示。

图 3-1. R7F0C80112ESP 的存储器映射图



注意事项 禁止存取保留区域。

图 3-2. R7F0C80212ESP 的存储器映射图



注意事项 禁止存取保留区域。

3.1.1 内部程序存储器空间

内部程序存储器空间中存储程序和数据表。

R7F0C80112ESP, R7F0C80212ESP 产品中内置的内部 ROM（闪存）如下所示。

表 3-1. 内部 ROM 容量

产品编号	内部 ROM	
	结构	容量
R7F0C80112ESP	闪存	1024 × 8 位 (00000H 至 003FFH)
R7F0C80212ESP		2048 × 8 位 (00000H 至 007FFH)

内部程序地址空间分为以下区域。

(1) 向量表区域

00000H 至 0007FH 的 128 字节区域作为向量表区域被保留。向量表区存放的是由于复位或各中断请求产生而进行转移时的程序的开始地址。另外，由于假设向量代码为 2 个字节，因此中断的跳转地址为 00000H 至 0FFFFH 的 64K 地址。

在该 16 位地址中，低 8 位存储于偶数地址，高 8 位存储于奇数地址。

表 3-2. 向量表

向量表地址	中断源
00002H	—
00004H	INTWDTI
00006H	INTP0
00008H	INTP1
0000AH	INTST0, INTCSI00
0000CH	INTSR0
0000EH	INTSRE0
00010H	INTTM01H
00012H	INTTM00
00014H	INTTM01
00016H	INTAD
00018H	INTKR
0007EH	BRK

(2) CALLT 指令表区域

00080H 至 000BFH 的 64 字节区域可以存储 2 字节调用指令 (CALLT)的子程序入口地址。将子程序入口地址设为 00000H 至 0FFFFH 范围内的一个值（因为地址代码为 2 个字节）。

(3) 选项字节区域

000C0H 至 000C3H 的 4 字节区域可用作选项字节区域。详情请参阅**第十六章 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。详情请参阅**第十八章 片上调试功能**。

3.1.2 镜像区域

闪存为 1 KB /2 KB 的产品中，00000H 至 003FFH/007FFH 的代码闪存区域被映射到 F8000H 至 F83FFH/F87FFH 区域（待映射的代码闪存区域通过处理器模式控制寄存器(PMC)设置）。

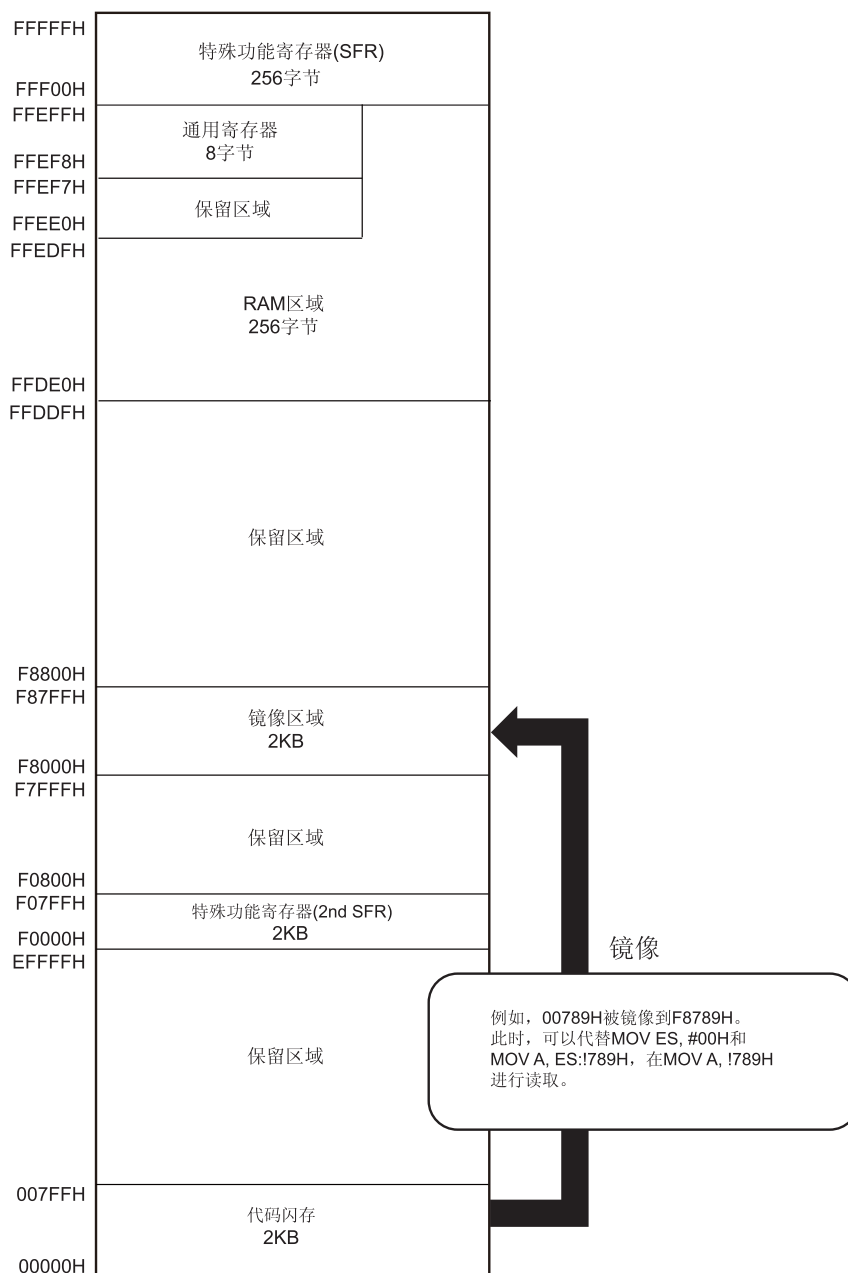
通过从 F8000H 至 F83FFH/F87FFH 区域读取数据，可以使用不用 ES 寄存器作为操作数的指令，因此，能够用短代码读取代码闪存的内容。

关于每种产品的镜像区域，请参阅 **3.1 地址空间**。

镜像区域为只读，不得从该区域取指令。

示例如下所示。

示例 R7F0C80112ESP (闪存: 2 KB)



3.1.3 内部数据存储空间

R7F0C80112ESP, R7F0C80212ESP 产品中内置了以下 RAM。

表 3-3. 内部 RAM 容量

产品编号	内部 RAM
R7F0C80112ESP	128 字节 (FFE60H 至 FFEDFH)
R7F0C80212ESP	256 字节 (FFDE0H 至 FFEDFH)

内部 RAM 可以用作数据区域以及写入并执行指令的程序区域。内部 RAM 用作堆栈存储器。

3.1.4 特殊功能寄存器 (SFR) 区域

片上外围硬件的特殊功能寄存器(SFR)被分配到 FFF00H 至 FFFFFH 区域(参阅 3.2.4 特殊功能寄存器(SFR) 中的表 3-4)。

注意事项 不得存取未分配 SFR 的地址。

3.1.5 扩展特殊功能寄存器 (2nd SFR: 第二特殊功能寄存器) 区域

片上外围硬件的扩展特殊功能寄存器(2nd SFR)被分配到 F0000H 至 F07FFH 区域(参阅 3.2.5 扩展特殊功能寄存器 (2nd SFR: 第二特殊功能寄存器) 中的表 3-5)。

将 SFR 区域 (FFF00H 至 FFFFFH 区) 以外的 SFR 分配到该区域。存取扩展 SFR 区域的指令比存取 SFR 区域的指令长 1 个字节。

注意事项 不得存取未分配扩展 SFR 的地址。

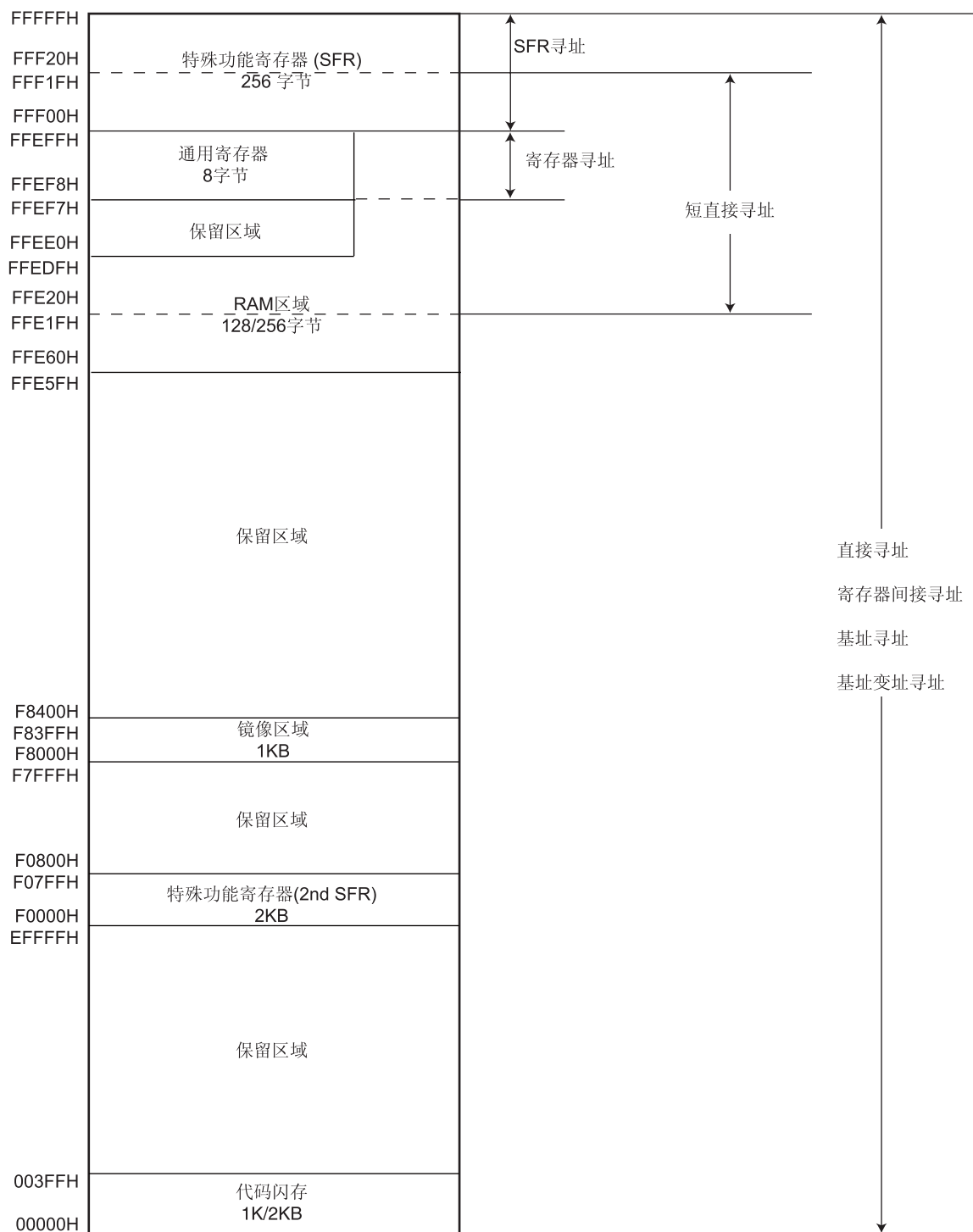
3.1.6 数据存储器寻址

寻址是一种方法，用于指定下一步将执行的指令的地址，或者执行指令时相关的寄存器或存储器等的地址。

考虑到可操作性等，R7F0C80112ESP, R7F0C80212ESP 提供了丰富的寻址方式，用于寻址执行指令时作为操作对象的存储器。尤其在具有数据存储器的区域，可以使用配合特殊功能寄存器(SFR)或通用寄存器等的功能而设计的特殊的寻址方法。数据存储器寻址之间的对应如图 3-3 所示。

每种寻址的详情请参阅 3.4 处理数据地址的寻址。

图 3-3. 数据存储器 and 寻址之间的对应



3.2 处理器寄存器

R7F0C80112ESP, R7F0C80212ESP 产品中内置了以下处理器寄存器。

3.2.1 控制寄存器

控制寄存器用于控制程序时序、状态和堆栈存储器。控制寄存器由程序计数器(PC)、程序状态字(PSW)和堆栈指针(SP)组成。

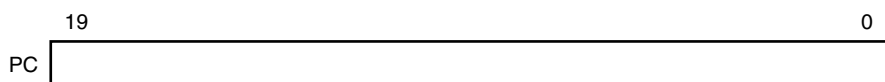
(1) 程序计数器(PC)

程序计数器是一个 20 位寄存器，用于存储即将执行的下个程序的地址信息。

在通常操作中，PC 根据待获取的指令的字节数自动递增。执行分支指令时，设置立即数和寄存器内容。

产生复位信号后，地址 0000H 和 0001H 处的复位向量表的值被设置至低 16 位，高 4 位被清除为 0000。

图 3-4. 程序计数器的格式

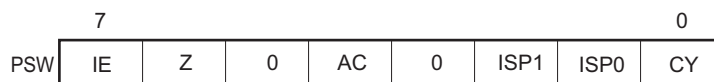


(2) 程序状态字(PSW)

程序状态字为 8 位寄存器，由通过执行指令来置位/复位的多个标志组成。

程序状态字内容在向量中断请求被受理或 PUSH PSW 指令被执行时存储于堆栈区中，并在执行 RETB、RETI 和 POP PSW 指令时恢复。产生复位信号后，PSW 寄存器被设置为 06H。

图 3-5. 程序状态字的格式



(a) 中断允许标志(IE)

该标志控制 CPU 的中断请求受理操作。

当 IE 为 0 时，变为中断禁止(DI)状态，禁止所有可屏蔽中断请求。

当其为 1 时，IE 标志设置为中断允许(EI)状态，通过受理中断的优先级标志 (ISP1、ISP0)、针对多种中断源的中断屏蔽标志和优先级指定标志来控制可屏蔽中断请求的受理。

IE 标志在执行 DI 指令或中断受理时复位(0)，在执行 EI 指令时被置位(1)。

(b) 零标志(Z)

当运算或比较结果为零或相同时，该标志被置位(1)。其他情况下被复位(0)。

(c) 辅助进位标志(AC)

如果运算结果在位 3 进位或在位 3 借位，则该标志被置位(1)。其他情况下被复位(0)。

(d) 受理中断的优先级标志(ISP1, ISP0)

可以受理的可屏蔽向量中断的优先级由该标志管理。优先级指定标志寄存器 (PR00L、PR00H、PR10L、PR10H) (参阅 11.3.3 优先级选择标志寄存器(PR00L, PR00H, PR10L, PR10H))所指定的低于 ISP0 和 ISP1 标志值的向量中断请求禁止受理。实际上是否受理向量中断请求由中断允许标志 (IE)控制。

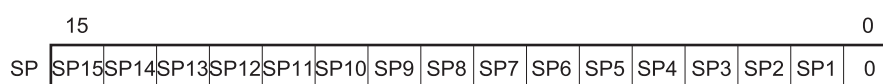
(e) 进位标志(CY)

该标志存储执行加/减指令时的上溢和下溢。执行循环指令时存储移出值，执行位操作指令时用作位累加器。

(3) 堆栈指针(SP)

这是一个 16 位寄存器，用于保存存储器堆栈区域的起始地址。仅限内部 RAM 区域可设为堆栈区域。

图 3-6. 堆栈指针的格式

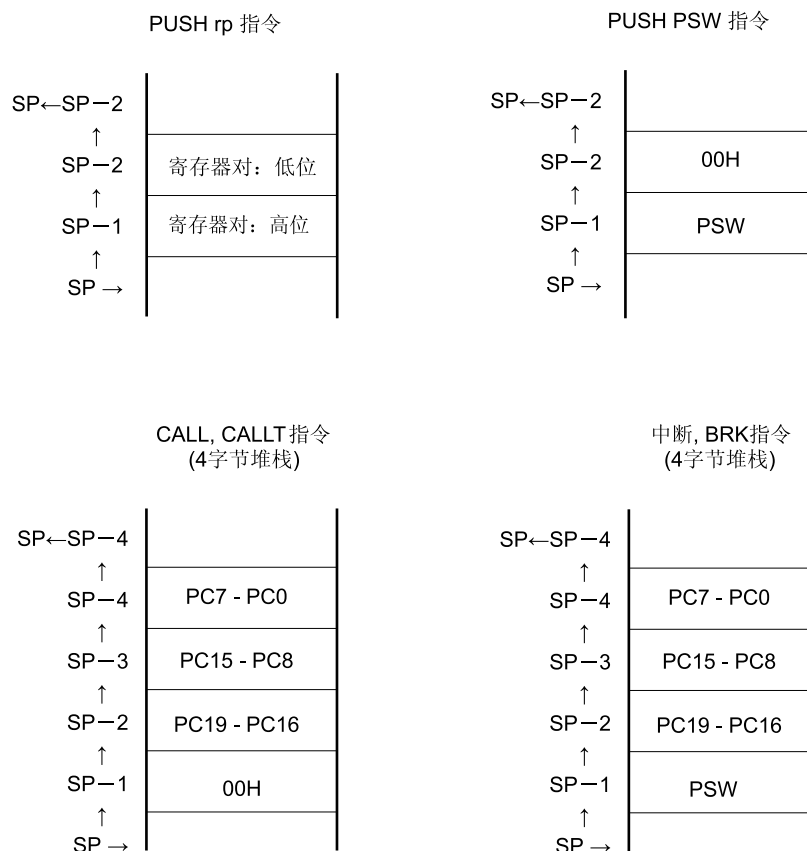


SP 在写入（压栈）堆栈存储器之前递减，在从堆栈存储器读取（出栈）之后递增。

各堆栈操作所保存的数据如图 3-7 所示。

注意事项 由于产生复位信号时会使 SP 的内容不定，因此必须在使用堆栈之前初始化 SP。

图 3-7. 保存到堆栈的数据



3.2.2 通用寄存器

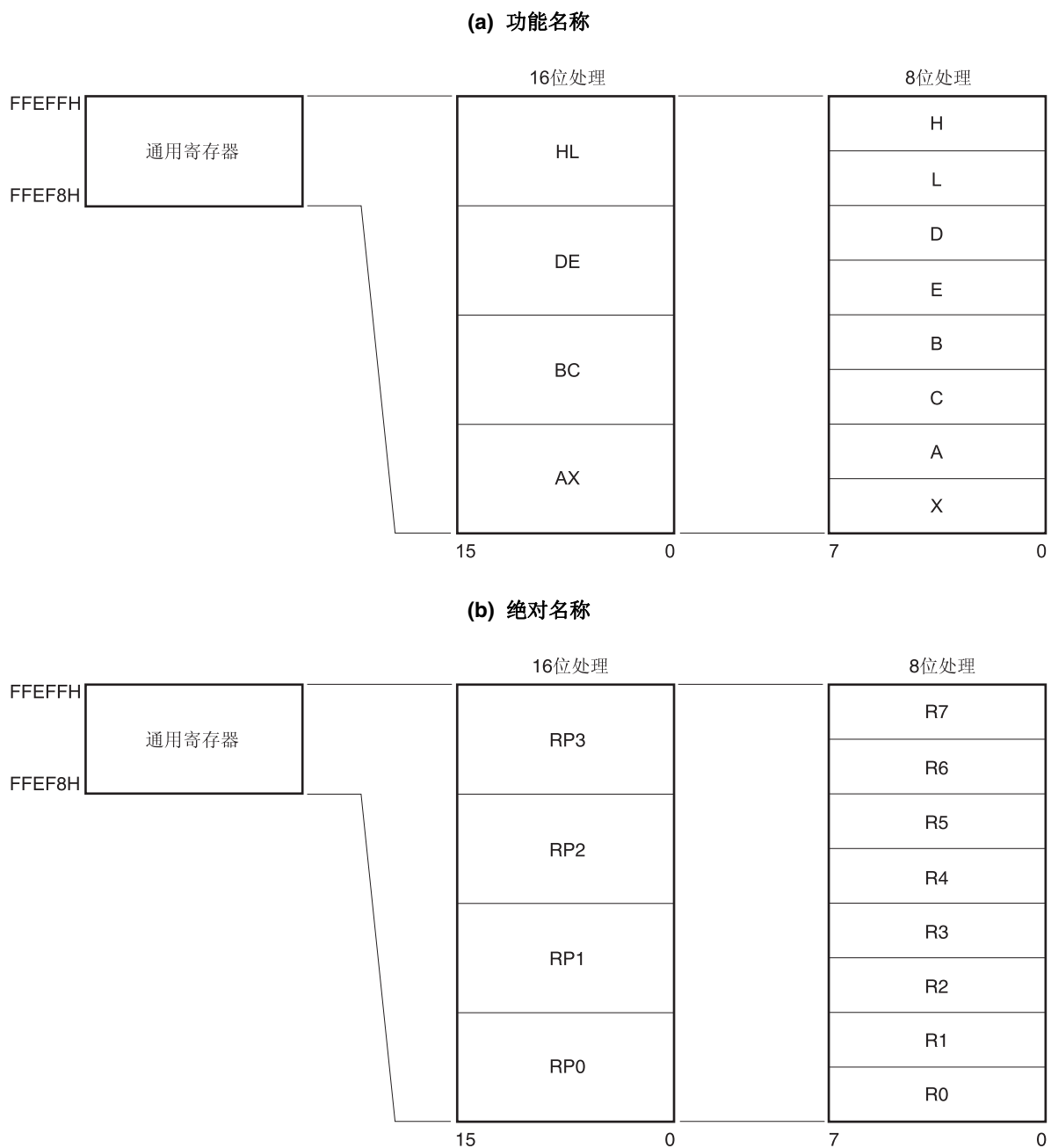
通用寄存器是包括 8 个 8 位寄存器（X、A、C、B、E、D、L 和 H）的寄存器组，被映射到数据存储器的地址（FFEF8H 至 FFEFFH）。

每个寄存器可以作为 1 个 8 位寄存器使用，也可以将 2 个 8 位寄存器组合起来作为 1 个 16 位寄存器（AX、BC、DE 和 HL）使用。

这些寄存器可以用功能名称（X、A、C、B、E、D、L、H、AX、BC、DE 和 HL）以及绝对名称（R0 至 R7 及 RP0 至 RP3）来描述。

注意事项 禁止使用通用寄存器（FFEF8H 至 FFEFFH）空间来获取指令，也不得将其用作堆栈区域。

图 3-8. 通用寄存器的配置



3.2.3 ES 和 CS 寄存器

ES 寄存器用于数据存取，CS 寄存器（寄存器直接寻址）用于在执行分支指令时分别指定高位地址。

ES 寄存器复位后的初始值为 0FH，CS 寄存器复位后的初始值为 00H。

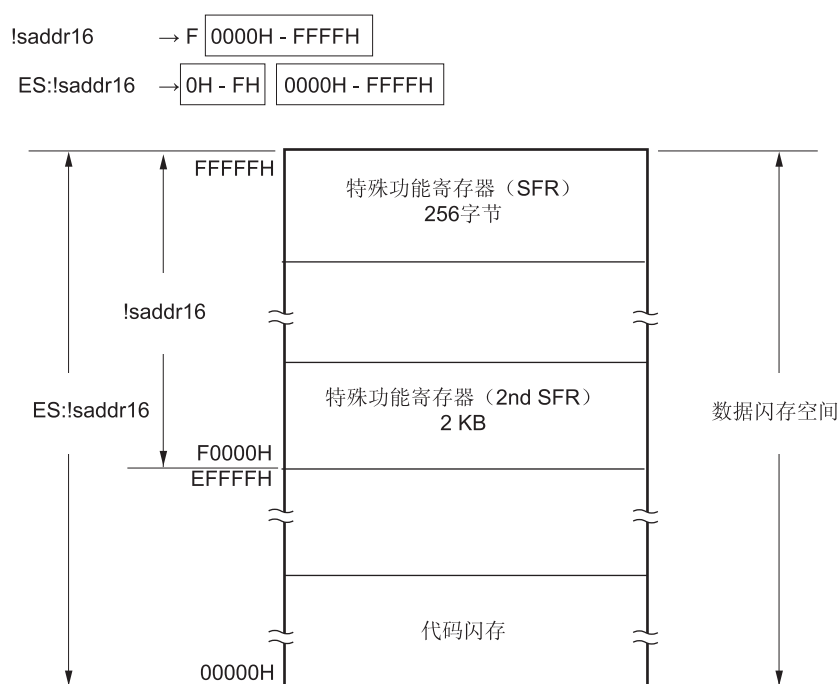
图 3-9. ES 和 CS 寄存器的配置

	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

通过 16 位地址可以对 F0000H 至 FFFFFH 的 64 KB 空间的数据区域进行存储，附加 ES:后，可扩展为 00000H 至 FFFFFH 的 1 MB 空间。

图 3-10. 数据存储区域的扩展



3.2.4 特殊功能寄存器(SFR)

与通用寄存器不同，每个 SFR 都具有特殊功能。

SFR 空间被分配到 FFF00H 至 FFFFFH 区域。

通过运算、传送和位操作指令，可以像操作通用寄存器一样操作 SFR。可操作的单位（1、8）因 SFR 的不同而异。操作单位的指定方法如下所示。

- 1 位操作

通过汇编程序的保留符号指定位操作指令的操作数(sfr.bit)。也可以为该操作指定地址。

- 8 位操作

通过汇编程序的保留符号指定 8 位操作指令的操作数(sfr)。也可以为该操作指定地址。

表 3-4 为 SFR 列表。表中所列项目的含义如下所示。

- 符号

表示特殊功能寄存器地址的符号。在汇编程序中作为保留字，在编译程序中通过 #pragma sfr 指令被定义为 sfr 变量。当使用汇编程序、调试器和仿真器时，可以作为指令的操作数。

- R/W

表示相对应的 SFR 能否读取或写入。

R/W: 可读/写

R: 只读

W: 只写

- 可操作的位单位

“√”表示可操作的位单位（1 或 8）。“—”表示不可以操作的位单位。

- 复位后

表示产生复位信号后各个寄存器的状态。

注意事项 不得存取未分配扩展 SFR 的地址。

备注 关于扩展 SFR (2nd SFR)，请参阅 3.2.5 扩展特殊功能寄存器 (2nd SFR: 第二特殊功能寄存器)。

表 3-4. SFR 列表 (1/2)

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	
FFF00H	端口寄存器 0	P0		R/W	√	√	00H
FFF04H	端口寄存器 4	P4		R/W	√	√	00H
FFF0CH	端口寄存器 12	P12		R	√	√	不定
FFF0DH	端口寄存器 13	P13		R	√	√	不定
FFF10H	串行数据寄存器 00L	TXD0/ SIO00	SDR00L	R/W	—	√	00H
FFF11H	串行数据寄存器 00H	—	SDR00H	R/W	—	√	00H
FFF12H	串行数据寄存器 01L	RXD0/ SIO01	SDR01L	R/W	—	√	00H
FFF13H	串行数据寄存器 01H	—	SDR01H	R/W	—	√	00H
FFF18H	定时器数据寄存器 00L	TDR00L		R/W	—	√	00H
FFF19H	定时器数据寄存器 00H	TDR00H		R/W	—	√	00H
FFF1AH	定时器数据寄存器 01L	TDR01L		R/W	—	√	00H
FFF1BH	定时器数据寄存器 01H	TDR01H		R/W	—	√	00H
FFF1EH	A/D 转换结果低位寄存器	ADCRL		R	—	√	00H
FFF1FH	A/D 转换结果高位寄存器	ADCRH		R	—	√	00H
FFF20H	端口模式寄存器 0	PM0		R/W	√	√	FFH
FFF24H	端口模式寄存器 4	PM4		R/W	√	√	FFH
FFF30H	A/D 转换器模式寄存器 0	ADM0		R/W	√	√	00H
FFF31H	模拟输入通道选择寄存器	ADS		R/W	√	√	00H
FFF34H	按键中断控制寄存器	KRCTL		R/W	√	√	00H
FFF35H	按键中断标志寄存器	KRF		R/W	√	√	00H
FFF37H	按键中断模式寄存器 0	KRM0		R/W	—	√	00H
FFF38H	外部中断上升沿允许寄存器 0	EGP0		R/W	√	√	00H
FFF39H	外部中断下降沿允许寄存器 0	EGN0		R/W	√	√	00H

表 3-4. SFR 列表 (2/2)

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	
FFFA5H	时钟输出选择寄存器 0	CKS0	R/W	√	√	00H
FFFA8H	复位控制标志寄存器	RESF	R	—	√	不定 注 1
FFFABH	看门狗定时器允许寄存器	WDTE	R/W	—	√	1AH/9AH 注 2
FFFE0H	中断请求标志寄存器 0L	IF0L	R/W	√	√	00H
FFFE1H	中断请求标志寄存器 0H	IF0H	R/W	√	√	00H
FFFE4H	中断屏蔽标志寄存器 0L	MK0L	R/W	√	√	FFH
FFFE5H	中断屏蔽标志寄存器 0H	MK0H	R/W	√	√	FFH
FFFE8H	优先级选择标志寄存器 00L	PR00L	R/W	√	√	FFH
FFFE9H	优先级选择标志寄存器 00H	PR00H	R/W	√	√	FFH
FFFECH	优先级选择标志寄存器 10L	PR10L	R/W	√	√	FFH
FF FEDH	优先级选择标志寄存器 10H	PR10H	R/W	√	√	FFH

注 1. RESF 寄存器的复位值因复位源而异。

2. WDTE 寄存器的复位值决定于选项字节的设置。

备注 关于扩展 SFR (2nd SFR)，请参阅 表 3-5 扩展 SFR (2nd SFR) 列表。

3.2.5 扩展特殊功能寄存器 (2nd SFR: 第二特殊功能寄存器)

与通用寄存器不同，每个扩展 SFR (2nd SFR)都具有特殊功能。

扩展 SFR 被分配到 F0000H 至 F07FFH 区域。将 SFR 区域 (FFF00H 至 FFFFFH 区) 以外的 SFR 分配到该区域。存取扩展 SFR 区域的指令比存取 SFR 区域的指令长 1 个字节。

通过运算、传送和位操作指令，可以像操作通用寄存器一样操作扩展 SFR。可操作的单位 (1、8) 因 SFR 的不同而异。

操作单位的指定方法如下所示。

- 1 位操作

通过汇编程序的保留符号指定位操作指令的操作数(!addr16.bit)。也可以为该操作指定地址。

- 8 位操作

通过汇编程序的保留符号指定 8 位操作指令的操作数(!addr16)。也可以为该操作指定地址。

表 3-5 为扩展 SFR 的列表。表中所列项目的含义如下所示。

- 符号

指示扩展 SFR 地址的符号。在汇编程序中作为保留字，在编译程序中通过 #pragma sfr 指令被定义为 sfr 变量。当使用汇编程序、调试器和仿真器时，可以作为指令的操作数。

- R/W

表示相对应的扩展 SFR 能否读取或写入。

R/W: 可读/写

R: 只读

W: 只写

- 可操作的位单位

“√”表示可操作的位单位 (1 或 8)。“—”表示不可以操作的位单位。

- 复位后

表示产生复位信号后各个寄存器的状态。

注意事项 不得存取未分配扩展 SFR 的地址。

备注 关于 SFR 区域中的 SFR，请参阅 3.2.4 特殊功能寄存器 (SFR)。

表 3-5. 扩展 SFR (2nd SFR)列表 (1/2)

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	
F0010H	A/D 转换器模式寄存器 2	ADM2	R/W	√	√	00H
F0030H	上拉电阻选择寄存器 0	PU0	R/W	√	√	00H
F0034H	上拉电阻选择寄存器 4	PU4	R/W	√	√	01H
F003CH	上拉电阻选择寄存器 12	PU12	R/W	√	√	00H
F0050H	端口输出模式寄存器 0	POM0	R/W	√	√	00H
F0060H	端口模式控制寄存器 0	PMC0	R/W	√	√	FFH
F0070H	噪声滤波器允许寄存器 0	NFEN0	R/W	√	√	00H
F0071H	噪声滤波器允许寄存器 1	NFEN1	R/W	√	√	00H
F0073H	输入切换控制寄存器	ISC	R/W	√	√	00H
F0077H	外围输入/输出重定向寄存器	PIOR	R/W	—	√	00H
F00A8H	高速片上振荡器调节寄存器	HOCODIV	R/W	—	√	不定
F00F0H	外围允许寄存器 0	PER0	R/W	√	√	00H
F00FEH	BCD 校正结果寄存器	BCDADJ	R	—	√	不定
F0100H	串行状态寄存器 00	SSR00	R	—	√	00H
F0102H	串行状态寄存器 01	SSR01	R	—	√	00H
F0108H	串行标志清除触发寄存器 00	SIR00	R/W	—	√	00H
F010AH	串行标志清除触发寄存器 01	SIR01	R/W	—	√	00H
F0110H	串行模式寄存器 00L	SMR00L	R/W	—	√	20H
F0111H	串行模式寄存器 00H	SMR00H	R/W	—	√	00H
F0112H	串行模式寄存器 01L	SMR01L	R/W	—	√	20H
F0113H	串行模式寄存器 01H	SMR01H	R/W	—	√	00H
F0118H	串行通信操作设置寄存器 00L	SCR00L	R/W	—	√	87H
F0119H	串行通信操作设置寄存器 00H	SCR00H	R/W	—	√	00H
F011AH	串行通信操作设置寄存器 01L	SCR01L	R/W	—	√	87H
F011BH	串行通信操作设置寄存器 01H	SCR01H	R/W	—	√	00H
F0120H	串行通道允许状态寄存器 0	SE0	R	√	√	00H
F0122H	串行通道开始寄存器 0	SS0	R/W	√	√	00H
F0124H	串行通道停止寄存器 0	ST0	R/W	√	√	00H
F0126H	串行时钟选择寄存器 0	SPS0	R/W	—	√	00H
F0128H	串行输出寄存器 0	SO0	R/W	—	√	03H
F0129H	串行时钟输出寄存器 0	CKO0	R/W	—	√	03H
F012AH	串行输出允许寄存器 0	SOE0	R/W	√	√	00H
F0134H	串行输出电平寄存器 0	SOL0	R/W	—	√	00H
F0180H	定时器计数器寄存器 00L	TCR00L	R	—	√	FFH
F0181H	定时器计数器寄存器 00H	TCR00H	R	—	√	FFH
F0182H	定时器计数器寄存器 01L	TCR01L	R	—	√	FFH
F0183H	定时器计数器寄存器 01H	TCR01H	R	—	√	FFH

表 3-5. 扩展 SFR (2nd SFR)列表 (2/2)

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	
F0190H	定时器模式寄存器 00L	TMR00L	R/W	—	√	00H
F0191H	定时器模式寄存器 00H	TMR00H	R/W	—	√	00H
F0192H	定时器模式寄存器 01L	TMR01L	R/W	—	√	00H
F0193H	定时器模式寄存器 01H	TMR01H	R/W	—	√	00H
F01A0H	定时器状态寄存器 00	TSR00	R	—	√	00H
F01A2H	定时器状态寄存器 01	TSR01	R	—	√	00H
F01B0H	定时器通道允许状态寄存器 0	TE0	R	√	√	00H
F01B1H	定时器通道允许状态寄存器 0 (8 位模式)	TEH0	R	√	√	00H
F01B2H	定时器通道开始寄存器 0	TS0	R/W	√	√	00H
F01B3H	定时器通道开始寄存器 0 (8 位模式)	TSH0	R/W	√	√	00H
F01B4H	定时器通道停止寄存器 0	TT0	R/W	√	√	00H
F01B5H	定时器通道停止寄存器 0 (8 位模式)	TTH0	R/W	√	√	00H
F01B6H	定时器时钟选择寄存器 0	TPS0	R/W	—	√	00H
F01B8H	定时器输出寄存器 0	TO0	R/W	—	√	00H
F01BAH	定时器输出允许寄存器 0	TOE0	R/W	√	√	00H
F01BCH	定时器输出电平寄存器 0	TOL0	R/W	—	√	00H
F01BEH	定时器输出模式寄存器 0	TOM0	R/W	—	√	00H

备注 关于 SFR 区域中的 SFR，请参阅 表 3-4 SFR 列表。

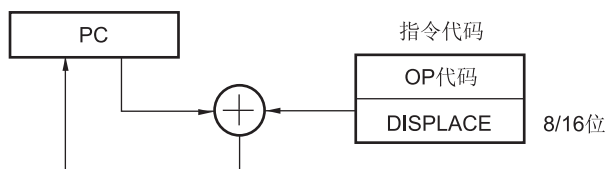
3.3 指令地址寻址

3.3.1 相对寻址

[功能]

相对寻址把指令字中的偏移量（有符号的补码数据：-128 至+127 或-32768 至+32767）与程序计数器(PC)的值（下一条指令的起始地址）之和存储在程序计数器(PC)中，指定转移目的地的程序地址。相对寻址仅用于分支指令。

图 3-11. 相对寻址的图示



3.3.2 立即寻址

[功能]

立即寻址把指令字中的立即数存储于程序计数器中，指定转移目的地的程序地址。

在立即寻址中，使用 **CALL !!addr20** 或 **BR !!addr20** 指定 20 位地址，使用 **CALL !addr16** 或 **BR !addr16** 指定 16 位地址。指定 16 位地址时，高 4 位设为 0000。

图 3-12. **CALL !!addr20/BR !!addr20** 的示例

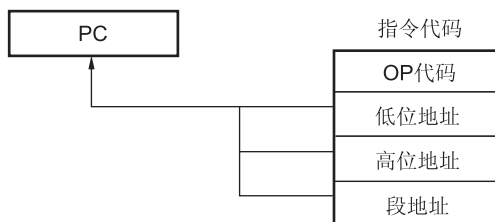
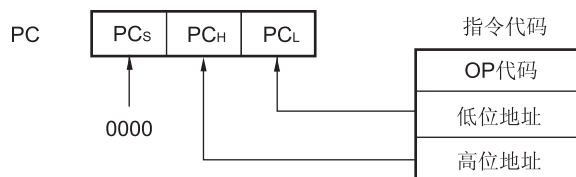


图 3-13. **CALL !addr16/BR !addr16** 的示例



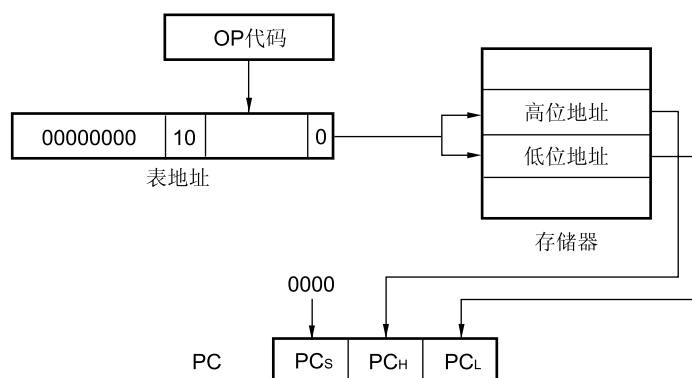
3.3.3 表间接寻址

[功能]

表间接寻址用指令字中的 5 位立即数在 CALLT 表区域（0080H 至 00BFH）中指定 1 个表地址，将表地址和下一个地址的内容以 16 位数据的形式存储在程序计数器(PC)中，指定程序地址。表间接寻址仅用于 CALLT 指令。

RL78 单片机中，仅在 00000H 至 0FFFFH 的 64KB 空间中允许分支。

图 3-14. 表间接寻址的图示

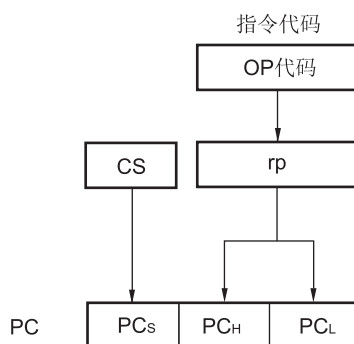


3.3.4 寄存器直接寻址

[功能]

寄存器直接寻址，是将通过指令字指定的当前寄存器组的通用寄存器对(AX/BC/DE/HL)和 CS 寄存器的内容，以 20 位数据的形式存储在程序计数器(PC)中，并指定程序地址。寄存器直接寻址只适用于 CALL AX、BC、DE、HL 和 BR AX 指令。

图 3-15. 寄存器直接寻址的图示



3.4 处理数据地址的寻址

3.4.1 隐含寻址

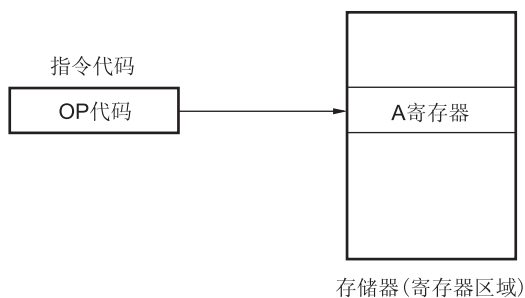
[功能]

用于存取具有特殊功能的寄存器（例如累加器）的指令，可以通过指令字直接指定，而无需使用指令字中的任何寄存器指定字段。

[操作数格式]

隐含寻址只适用于 MULU X。

图 3-16. 隐含寻址的图示



3.4.2 寄存器寻址

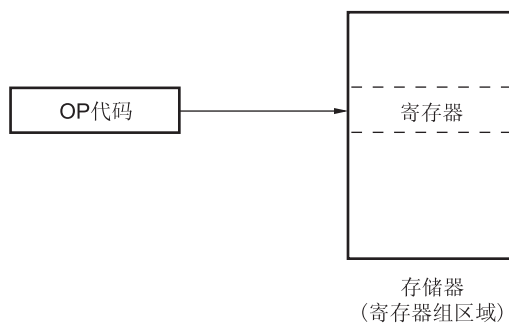
[功能]

寄存器寻址是将通用寄存器作为操作数来进行存取的一种寻址方式。通过 3 位的指令字选择 8 位寄存器，通过 2 位的指令字选择 16 位寄存器。

[操作数格式]

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

图 3-17. 寄存器寻址的图示



3.4.3 直接寻址

[功能]

直接寻址是将指令字中的立即数作为操作数地址，从而直接指定目标地址。

[操作数格式]

标识符	说明
!addr16	标号或16位立即数 (仅可以指定 F0000H至FFFFFH 的空间：高4位地址自动加 F)。
ES:!addr16	标号或16位立即数(高4位地址由ES寄存器指定)

图 3-18. ADDR16 的示例

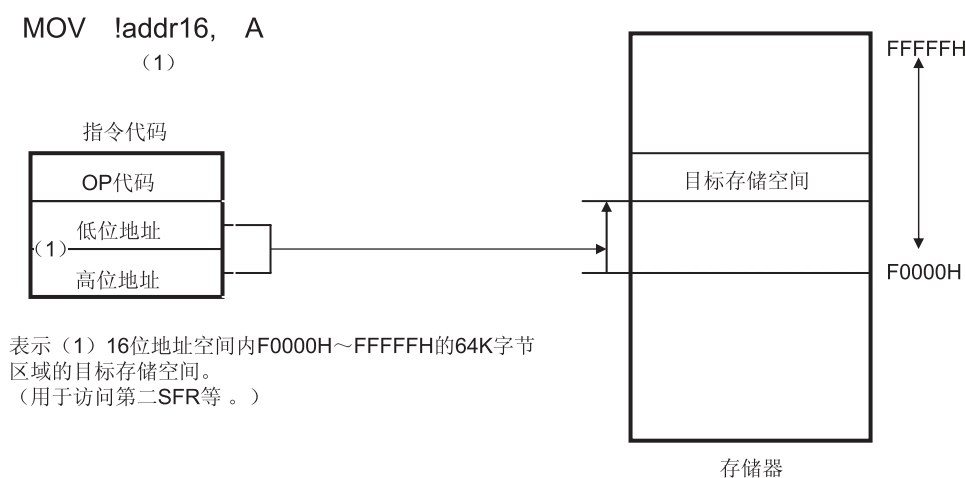
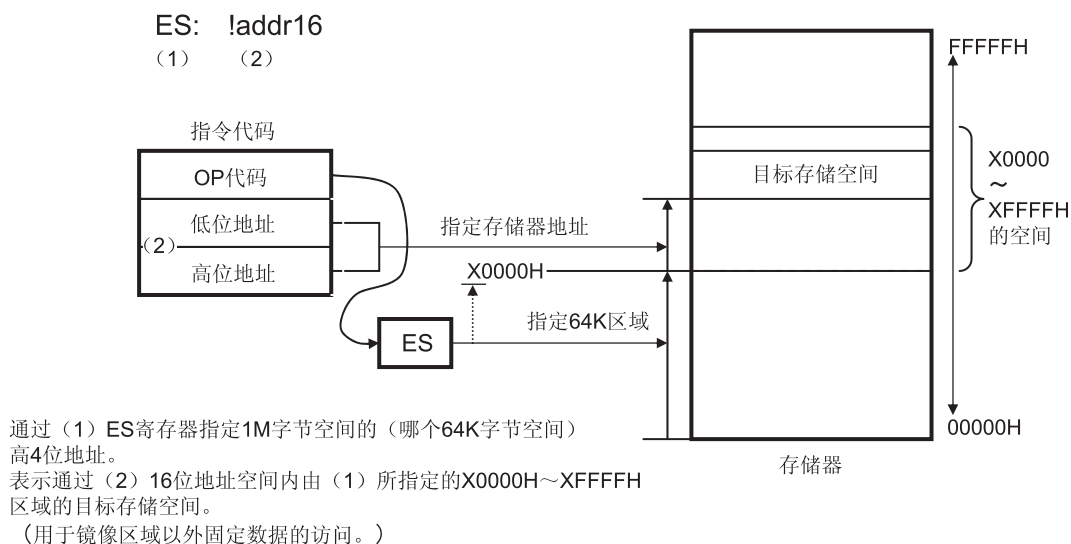


图 3-19. ES:ADDR16 的示例



3.4.4 短直接寻址

[功能]

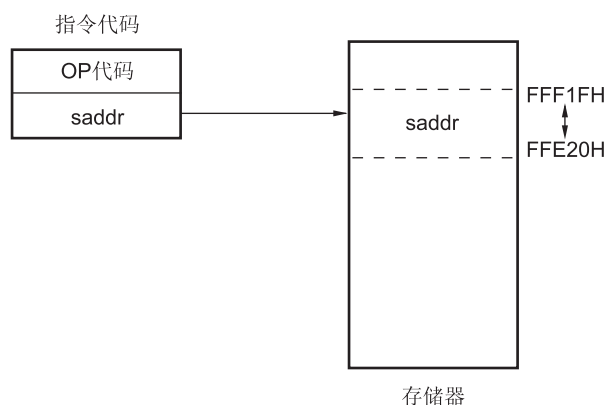
短直接寻址使用指令字中的 8 位数据直接指定目标地址。这种寻址方式只适用于从 FFE20H 至 FFF1FH 的空间。

请注意，从 FFEE0H 至 FFEF7H 的区域是禁止使用的。在具有 128 字节 RAM 的产品中，从 FFE20H 至 FFE5FH 的区域也是禁止使用的。

[操作数格式]

标识符	说明
SADDR	标号或 FFE20H至FFF1FH 的立即数
SADDRP	标号或FFE20H至FFF1FH 立即数 (仅限偶数地址可指定)

图 3-20. 短直接寻址的图示



备注 SADDR 和 SADDRP 用于以 16 位立即数（省略了实际地址的高 4 位）描述 FE20H 至 FF1FH 地址的值，或者以 20 位立即数描述 FFE20H 至 FFF1FH 地址的值。

无论采用 SADDR 还是 SADDRP，均指定存储器中 FFE20H 至 FFF1FH 空间内的地址。

3.4.5 SFR 寻址

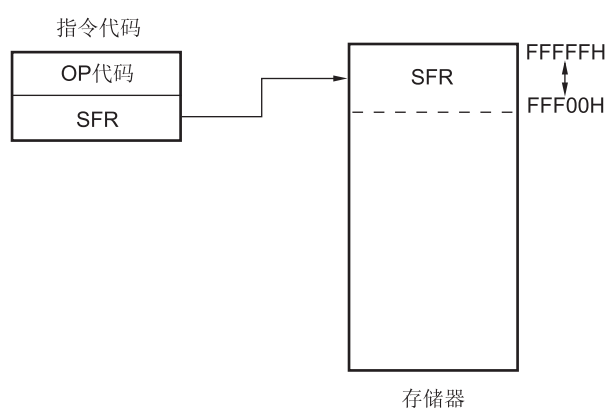
[功能]

SFR 寻址使用指令字中的 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	说明
SFR	SFR名称
SFRP	16位可操作SFR名称 (仅限偶数地址)

图 3-21. SFR 寻址的图示



3.4.6 寄存器间接寻址

[功能]

寄存器间接寻址是将指令字所指定的寄存器对的内容作为操作数地址，从而指定所对应的目标地址。

[操作数格式]

标识符	描述方法
–	[DE], [HL] (仅可以指定F0000H至FFFFFFH的空间)
–	ES:[DE], ES:[HL] (高4位地址由ES寄存器指定)

图 3-22. [DE], [HL]的示例

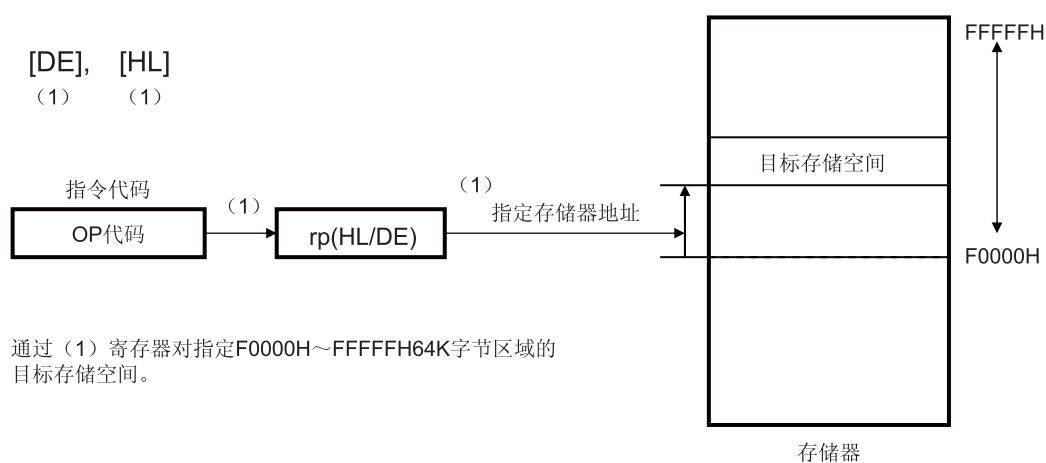
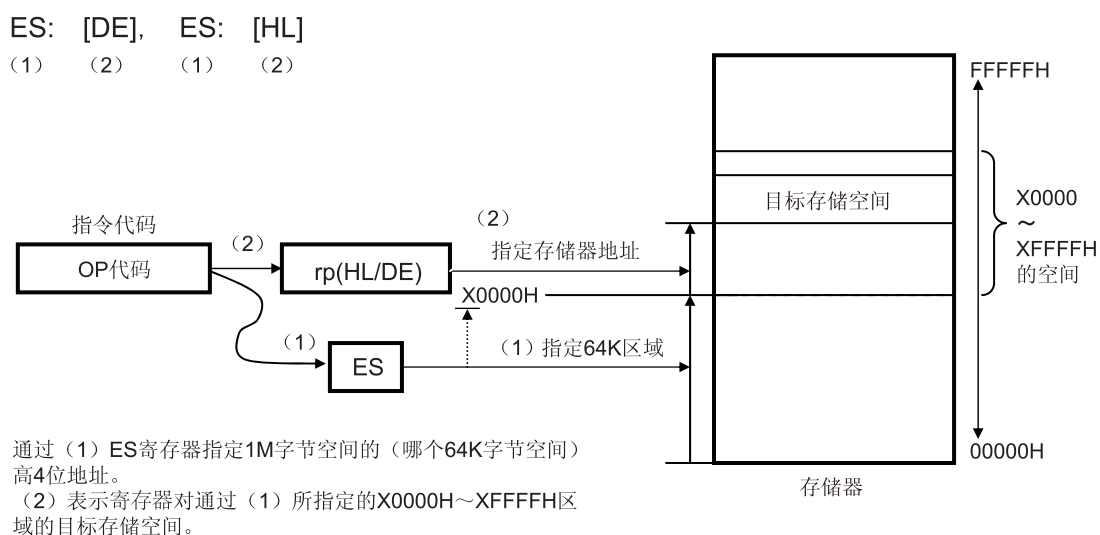


图 3-23. ES:[DE], ES:[HL]的示例



3.4.7 基址寻址

[功能]

基址寻址是将指令字所指定的寄存器对的内容或 16 位立即数作为基址，将 8 位立即数或 16 位立即数作为偏移量。并使用这些值之和指定目标地址。

[操作数格式]

标识符	描述方法
—	[HL + byte], [DE + byte], [SP + byte] (仅可以指定F0000H至FFFFFH的空间)
—	word[B], word[C] (仅可以指定F0000H至FFFFFH的空间)
—	word[BC] (仅可以指定F0000H至FFFFFH的空间)
—	ES:[HL + byte], ES:[DE + byte] (高4位地址由ES寄存器指定)
—	ES:word[B], ES:word[C] (高4位地址由ES寄存器指定)
—	ES:word[BC] (高4位地址由ES寄存器指定)

图 3-24. [SP+byte]的示例

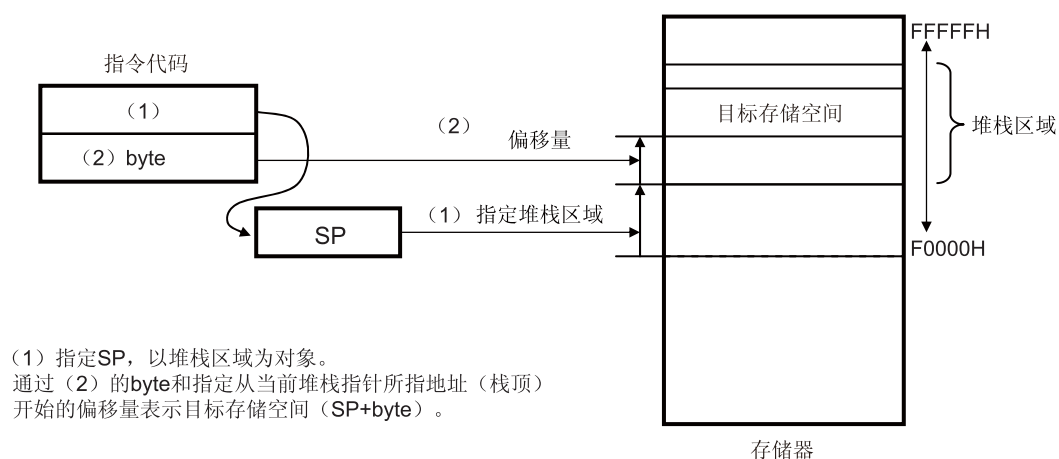


图 3-25. [HL + byte], [DE + byte]的示例

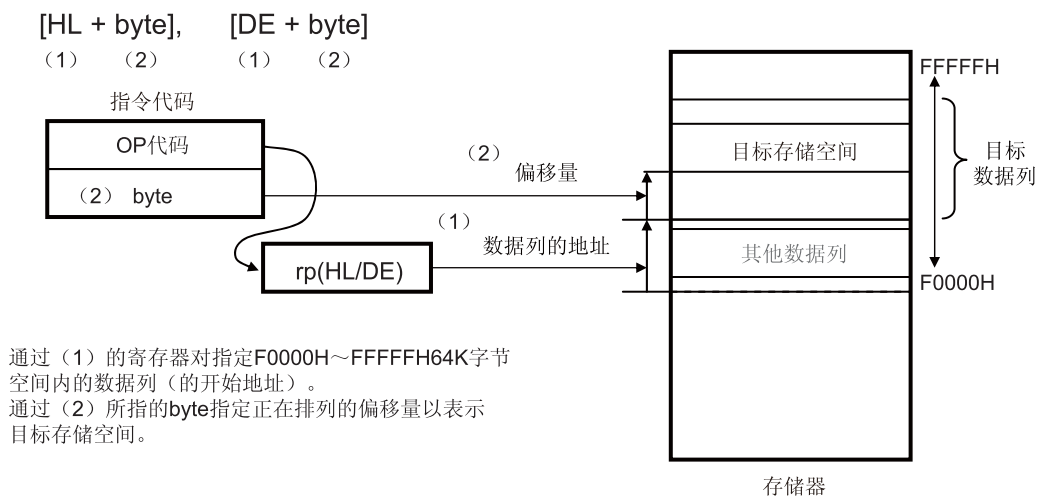


图 3-26. word[B], word[C]的示例

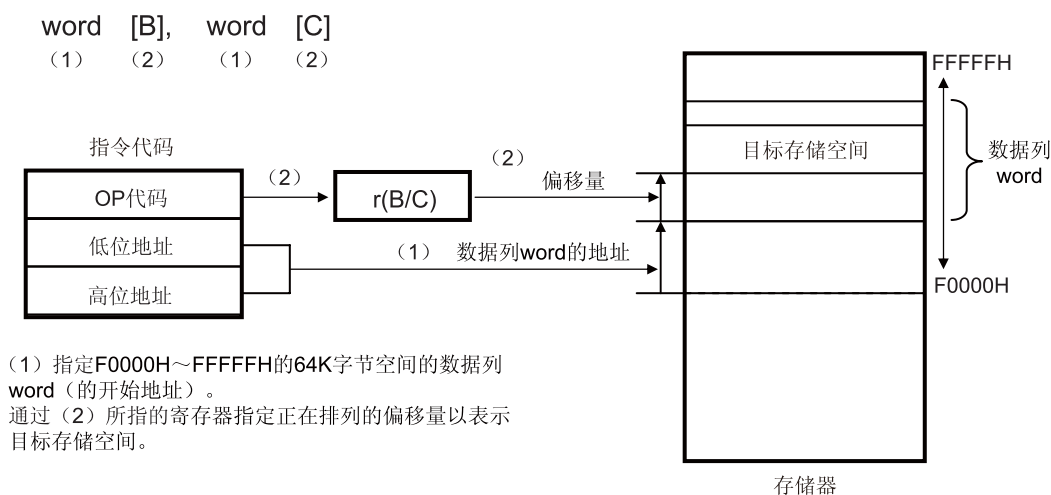


图 3-27. word[BC]的示例

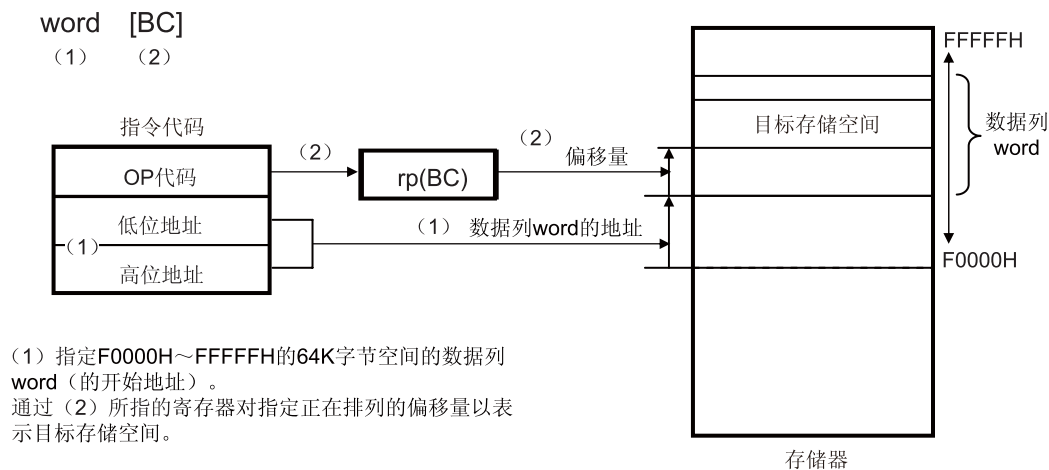


图 3-28. ES:[HL + byte], ES:[DE + byte]的示例

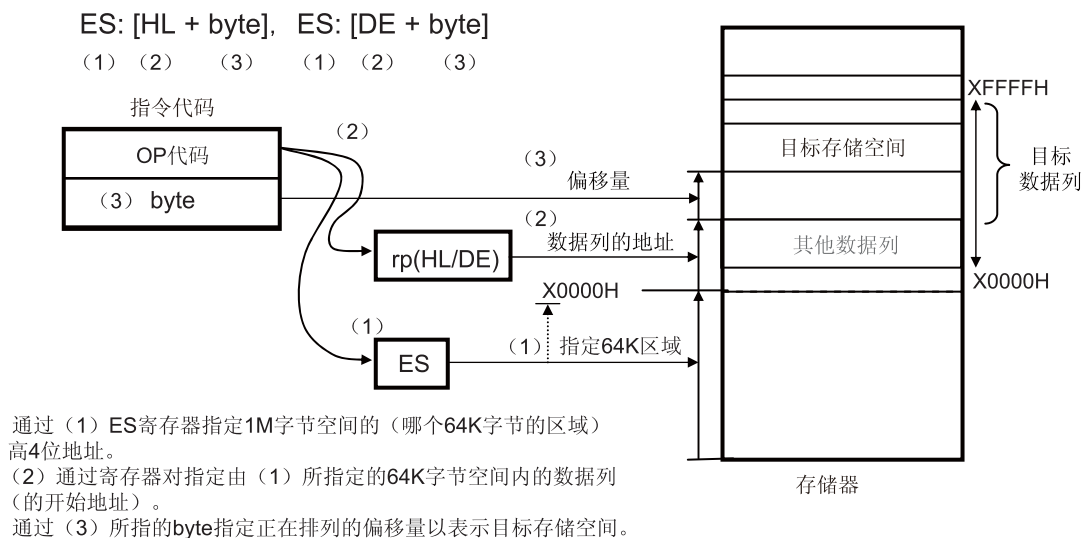


图 3-29. S:word[B], ES:word[C]的示例

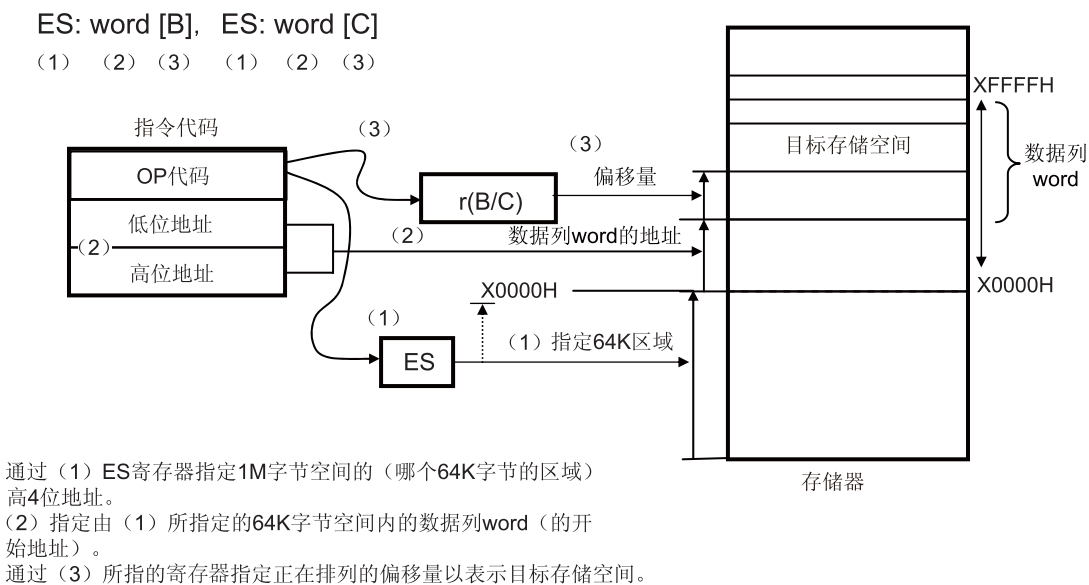
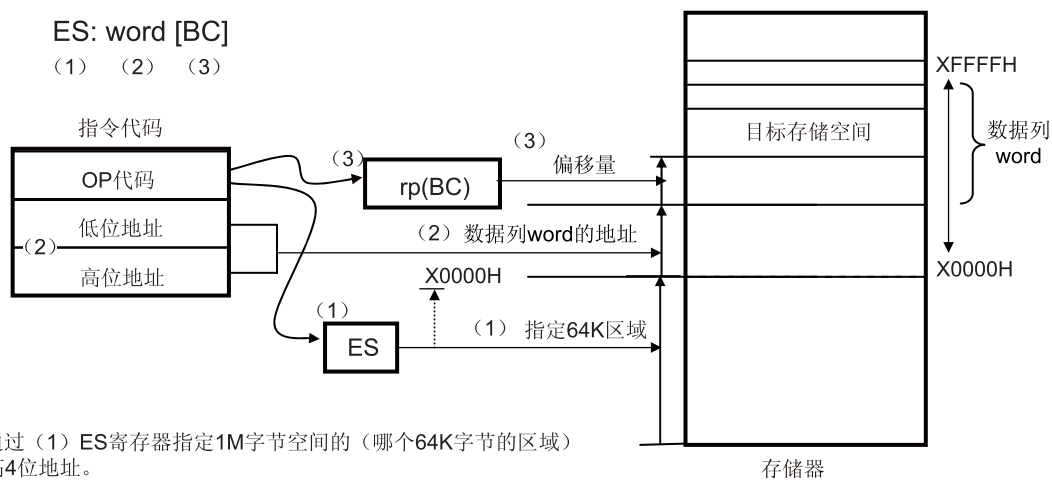


图 3-30. S:word[BC]的示例



通过 (1) ES寄存器指定1M字节空间的 (哪个64K字节的区域) 高4位地址。

(2) 指定由 (1) 所指定的64K字节空间内的数据列word (的开始地址)。

通过 (3) 所指的寄存器对指定正在排列的偏移量以表示目标存储空间。

3.4.8 基址变址寻址

[功能]

基址变址寻址是将指令字所指定的寄存器对的内容作为基址，将同样通过指令字所指定的 B 寄存器或 C 寄存器的内容作为偏移地址。并使用这些值之和指定目标地址。

[操作数格式]

标识符	描述方法
—	[HL+B], [HL+C] (仅可以指定F0000H至FFFFFH的空间)
—	ES:[HL+B], ES:[HL+C] (高4位地址由ES寄存器指定)

图 3-31. [HL+B], [HL+C]的示例

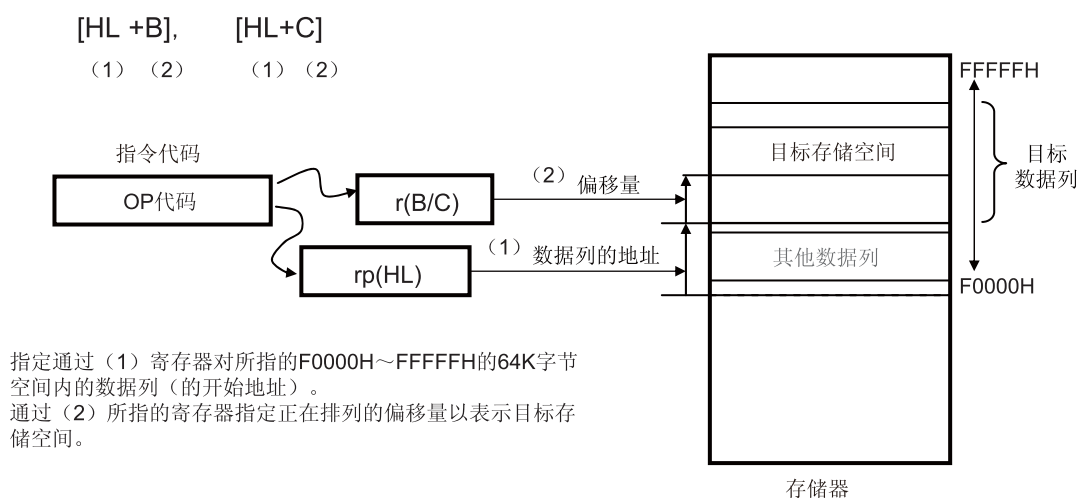
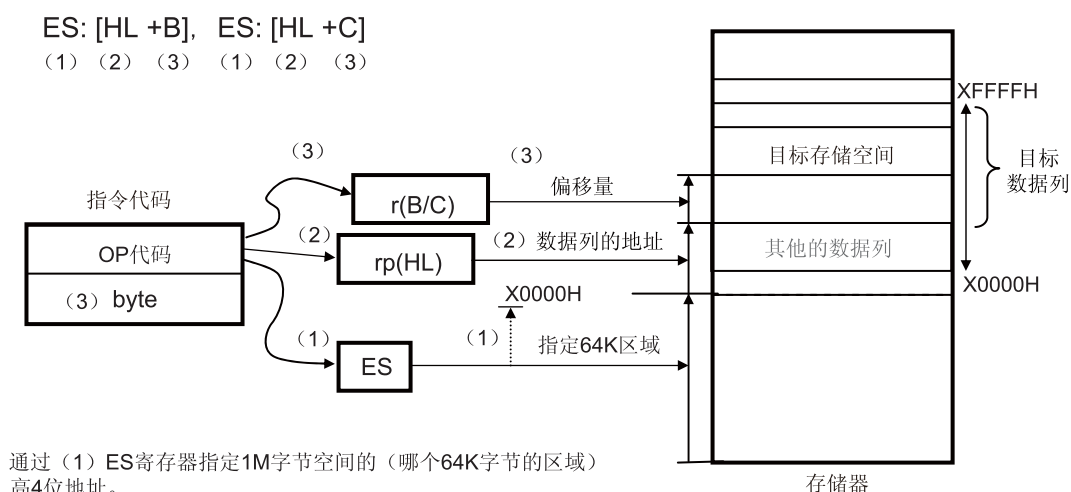


图 3-32. ES:[HL+B], ES:[HL+C]的示例



3.4.9 堆栈寻址

[功能]

通过堆栈指针(SP)的值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时，或者由于产生中断请求而保存/恢复寄存器时，将自动使用这种寻址方式。

仅能在内部 **RAM** 上设定堆栈区。

[操作数格式]

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

各堆栈操作中执行进栈/出栈的数据如图 3-33 至图 3-38 所示。

图 3-33. PUSH rp 的示例

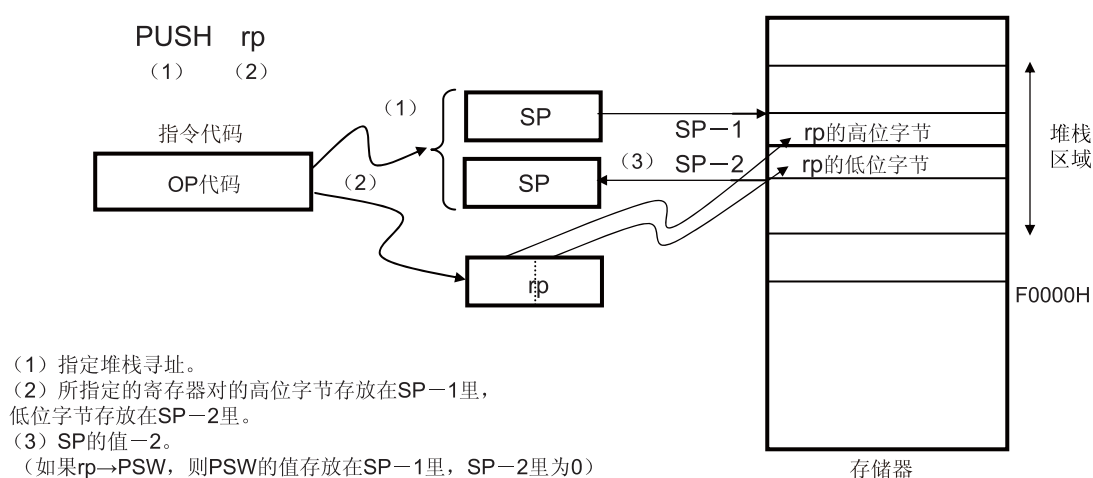


图 3-34. POP 的示例

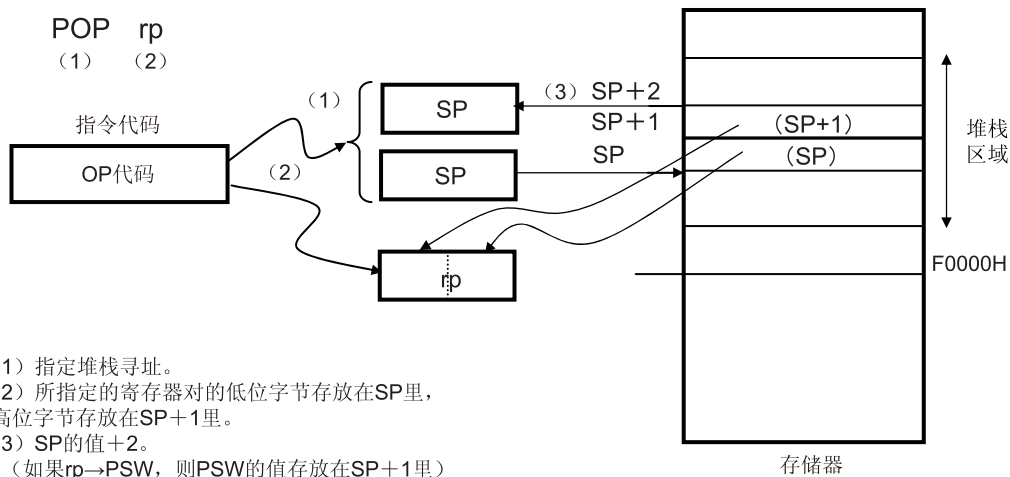


图 3-35. CALL, CALLT 的示例

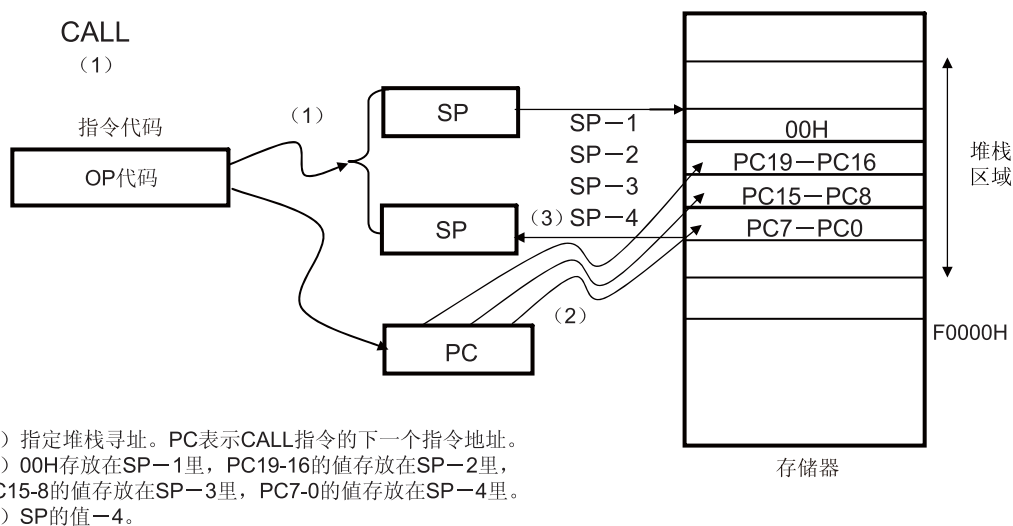


图 3-36. RET 的示例

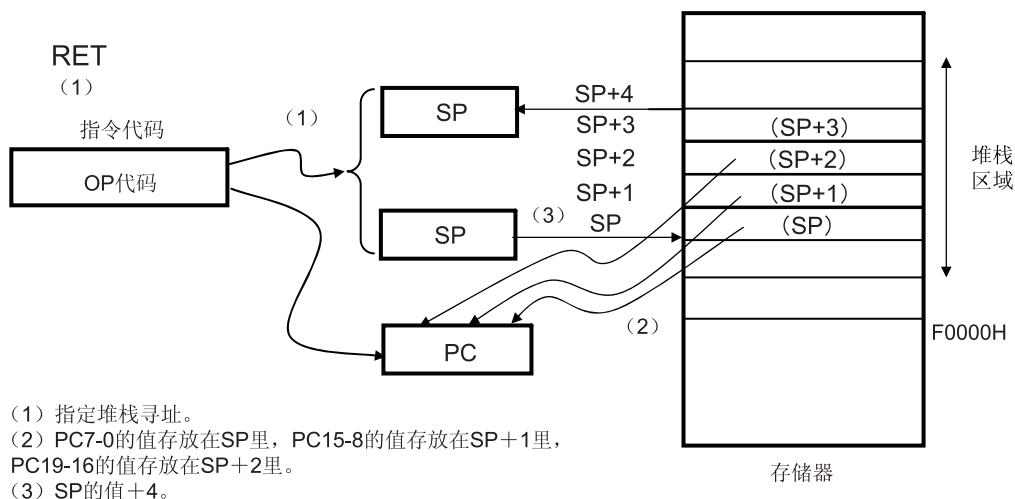
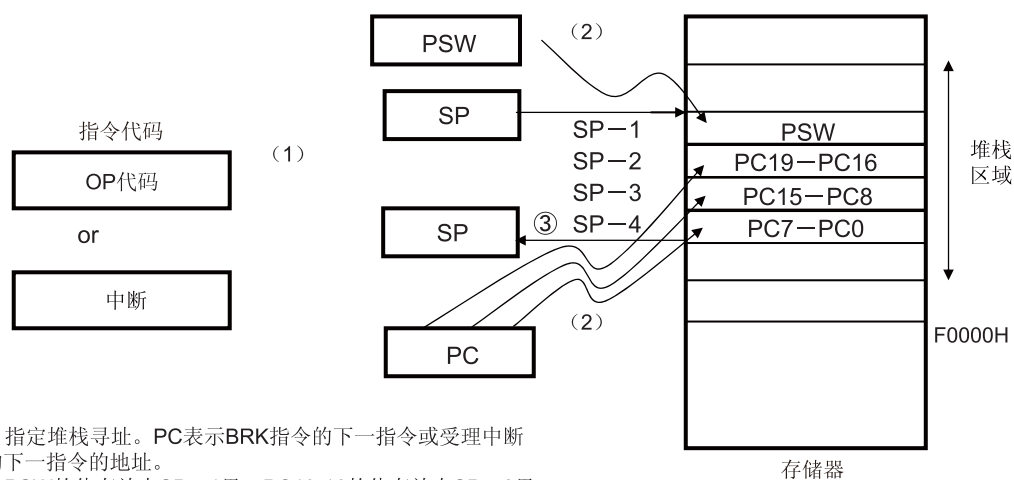
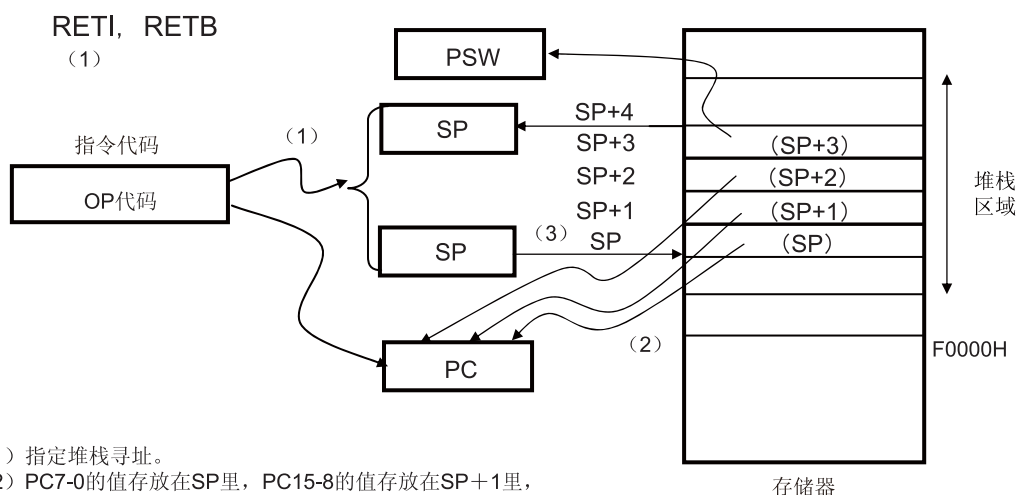


图 3-37. 中断, BRK 的示例



- (1) 指定堆栈寻址。PC表示BRK指令的下一指令或受理中断后的下一指令的地址。
- (2) PSW的值存放在SP-1里, PC19-16的值存放在SP-2里, PC15-8的值存放在SP-3里, PC7-0的值存放在SP-4里。
- (3) SP的值-4。

图 3-38. RETI, RETB 的示例



- (1) 指定堆栈寻址。
- (2) PC7-0的值存放在SP里, PC15-8的值存放在SP+1里, PC19-16的值存放在SP+2里, PSW的值存放在SP+3里。
- (3) SP的值+4。

第四章 端口功能

4.1 端口功能

R7F0C80112ESP, R7F0C80212ESP 单片机提供数字输入/输出端口，并支持多种控制操作。

除了作为数字输入/输出端口的功能以外，这些端口还具有多种复用功能。有关复用功能的详情，请参阅第二章 引脚功能。

4.2 端口配置

端口包括下列硬件。

表 4-1. 端口配置

项目	配置
控制寄存器	端口模式寄存器 0, 4 (PM0, PM4) 端口寄存器 0, 4, 12, 13 (P0, P4, P12, P13) 上拉电阻选择寄存器 0, 4, 12 (PU0, PU4, PU12) 端口输出模式寄存器 0 (POM0) 端口模式控制寄存器 0 (PMC0) 外围输入/输出重定向寄存器(PIOR)
端口	共计：8 (CMOS 输入/输出：6 (N 沟开漏输出(V_{DD} 耐压)：1)，CMOS 输入：2)
上拉电阻	共计：7

4.2.1 端口 0

端口 0 是具有输出锁存器的输入/输出端口。端口 0 可以通过端口模式寄存器 0(PM0)以 1 位为单位设为输入模式或输出模式。当 P00 至 P04 引脚用作输入引脚时，可通过上拉电阻选择寄存器 0(PU0)以 1 位为单位使用片上上拉电阻。

P00 引脚的输出可使用端口输出模式寄存器 0 (POM0)以 1 位为单位指定为 N 沟开漏输出 (V_{DD} 耐压)。

该端口也可以用作串行接口数据输入/输出、时钟输入/输出、模拟输入、按键返回输入、时钟/蜂鸣器输出、定时器输入/输出以及外部中断请求输入。

产生复位信号后，端口 0 为输入模式。

4.2.2 端口 4

端口 4 是具有输出锁存器的输入/输出端口。端口 4 可以通过端口模式寄存器 4(PM4)以 1 位为单位设为输入模式或输出模式。当 P40 引脚用作输入引脚时，可通过上拉电阻选择寄存器 4 (PU4)以 1 位为单位使用片上上拉电阻。

该端口也可以用于闪存编程器/调试器的按键返回输入和数据输入/输出。

4.2.3 端口 12

端口 12 是一个输入端口。P125 可通过上拉电阻选择寄存器 12 (PU12)使用片上上拉电阻 (在 $\overline{\text{RESET}}$ 输入^{*}时有效)。

该端口也可以用作按键返回输入和 $\overline{\text{RESET}}$ 输入。

注 投入电源后，P125 用作 $\overline{\text{RESET}}$ 输入。选项字节(000C1H)的 PORTSELB 位将此端口定义为 P125/KR1/SI01 或 $\overline{\text{RESET}}$ 。该引脚被设置为 P125/KR1 时，在可选上电复位(SPOR)电路的复位期间以及从解除 SPOR 电路的复位到开始普通操作为止的期间，不要向该引脚输入低电平。如果在此期间向该引脚输入了低电平，芯片将保持外部复位状态。另外，通电后可以启用上拉电阻。

4.2.4 端口 13

端口 13 是一个输入端口。

该端口也可以用作定时器输入和外部中断输入。

4.3 控制端口功能的寄存器

使用以下寄存器控制端口。

- 端口模式寄存器 0, 4 (PM0, PM4)
- 端口寄存器 0, 4, 12, 13 (P0, P4, P12, P13)
- 上拉电阻选择寄存器 0, 4, 12 (PU0, PU4, PU12)
- 端口输出模式寄存器 0 (POM0)
- 端口模式控制寄存器 0 (PMC0)
- 外围输入/输出重定向寄存器 (PIOR)

注意事项 各寄存器的未定义位因产品而异，须在初始值状态下使用。

表 4-2. PMx、Pxx、PUxx、POMx、PMCxx 寄存器及其对应的位

端口		位名称				
		Px 寄存器	PMx 寄存器	PUx 寄存器	POMx 寄存器	PMCx 寄存器
PORT0	0	P00	PM00	PU00	POM00	—
	1	P01	PM01	PU01	—	PMC01
	2	P02	PM02	PU02	—	PMC02
	3	P03	PM03	PU03	—	PMC03
	4	P04	PM04	PU04	—	PMC04
PORT4	0	P40	PM40	PU40	—	—
PORT12	5	P125	—	PU125	—	—
PORT13	7	P137	—	—	—	—

各寄存器的格式如下所述。

4.3.1 端口模式寄存器 0, 4 (PM0, PM4)

该寄存器以 1 位为单位指定端口的输入模式或输出模式。
使用 1 位或 8 位存储器操作指令设置该寄存器。
产生复位信号后，该寄存器被设置为 FFH。
当端口引脚用作复用功能的引脚时，端口模式寄存器的设置请参阅 4.5 使用复用功能时相关寄存器的设置。

图 4-1. 端口模式寄存器 0, 4 (PM0, PM4)的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W
PMmn		Pmn 引脚输入/输出模式的选择									
0		输出模式 (输出缓冲器启用)									
1		输入模式 (输出缓冲器关闭)									

m = 0, 4; n = 0 至 4

4.3.2 端口寄存器 0, 4, 12, 13 (P0, P4, P12, P13)

该寄存器设置端口的输出锁存器的值。

如果输入模式时读取数据，将得到引脚电平。如果输出模式时读取，将会得到输出锁存器的值^注。

使用 1 位或 8 位存储器操作指令设置该寄存器。

产生复位信号后，P12 和 P13 寄存器被设置为不定值，其他寄存器被清除为 00H。

注 被设为 A/D 转换器模拟输入的某个端口，在输入模式下读取此端口时，将始终返回 0 而非引脚电平。

此外，设置用作 P125 的 RESET 引脚的输出锁存器中，将始终读出 1。

图 4-2. 端口寄存器 0, 4, 12, 13 (P0, P4, P12, P13)的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	0	0	P04	P03	P02	P01	P00	FFF00H	00H (输出锁存器)	R/W
P4	0	0	0	0	0	0	0	P40	FFF04H	00H (输出锁存器)	R/W
P12	0	0	P125	0	0	0	0	0	FFF0CH	不定	R
P13	P137	0	0	0	0	0	0	0	FFF0DH	不定	R

Pmn	输出数据控制 (输出模式时)	输入数据读取 (输入模式时)
0	输出 0	输入低电平
1	输出 1	输入高电平

m = 0, 4, 12, 13; n = 0 至 5, 7

4.3.3 上拉电阻选择寄存器 0, 4, 12 (PU0, PU4, PU12)

该寄存器指定是否使用片上上拉电阻。在那些寄存器中使用片上上拉电阻的引脚，仅在满足以下三个设置条件的位才能以 1 位为单位使用片上上拉电阻。

- PMmn = 1(输入模式)
- 设置 PMCmn 寄存器的数字输入
- POM0n = 0

无论该寄存器如何设置，片上上拉电阻不会连接至设为输出模式的位，作为复用功能的输出引脚使用时也是同样如此。使用 1 位或 8 位存储器操作指令设置该寄存器。

产生复位信号后，PU4 被设置为 01H，PU12 被设置为 20H，PU0 被清除为 00H。

图 4-3. 上拉电阻选择寄存器 0, 4, 12 (PU0, PU4, PU12)的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	0	0	0	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W
PU12	0	0	PU125 注	0	0	0	0	0	F003CH	20H	R/W

PUmn	Pmn 引脚片上上拉电阻的选择
0	不连接片上上拉电阻
1	连接片上上拉电阻

注 仅在选择了 P125/KR1 功能(PORTSELB = 0)时可以操作该位。

备注 m = 0, 4, 12; n = 0 至 5

4.3.4 端口输出模式寄存器 0 (POM0)

该寄存器以 1 位为单位设置 CMOS 输出或 N 沟开漏输出。
使用 1 位或 8 位存储器操作指令设置该寄存器。
产生复位信号后，这些寄存器被清除为 00H。

图 4-4. 端口输出模式寄存器 0 (POM0)的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM0	0	0	0	0	0	0	0	POM00	F0050H	00H	R/W

POMmn	Pmn 引脚输出模式的选择
0	普通输出模式 输入模式时，启用 PUm _n 位。
1	N 沟开漏输出 (V _{DD} 耐压)模式 输入模式时，禁用 PUm _n 位。

m = 0; n = 0

4.3.5 端口模式控制寄存器 0 (PMC0)

该寄存器以 1 位为单位设置数字输入/输出或模拟输入。
使用 1 位或 8 位存储器操作指令设置该寄存器。
产生复位信号后，该寄存器被设置为 FFH。

图 4-5. 端口模式控制寄存器 0 (PMC0)的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PMC0	1	1	1	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W

PMCmn	Pmn 引脚数字输入/输出和模拟输入的选择
0	数字输入/输出(模拟输入以外的复用功能)
1	模拟输入

m = 0; n = 1 至 4

注意事项 通过端口模式寄存器 m (PMm)将用于 D/A 转换的通道设置为输入模式。

4.3.6 外围输入/输出重定向寄存器(PIOR)

该寄存器用于设置是否允许或禁止外围输入/输出重定向功能。
该功能用于切换被分配了复用功能的端口。
使用 PIOR 寄存器给端口分配重定向功能后，允许操作此功能。
并且，对重定向设置进行改变时要在允许功能操作之前进行。
使用 8 位存储器操作指令设置 PIOR 寄存器。
产生复位信号后，该寄存器被清除为 00H。

图 4-6. 外围输入/输出重定向寄存器(PIOR)的格式

地址: F0077H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PIOR	0	0	0	0	0	PIOR2	PIOR1	PIOR0

位	功能	设置值	
		0	1
PIOR2	INTP1	P00	P03
PIOR1	TI01/TO01	P04	P40
PIOR0	PCLBUZ0	P02	P40

注意事项 不能设置为 PIOR0 = 1 且 PIOR1 = 1。

4.4 端口功能的操作

如下所示，端口的操作因输入/输出模式设置的不同而异。

4.4.1 输入/输出端口的写入

(1) 输出模式

通过传送指令向输出锁存器写入值，输出锁存器的内容则从引脚输出。

写入输出锁存器的数据被一直保留，直到再次有数据被写入输出锁存器。

产生复位信号后，输出锁存器的数据。

(2) 输入模式

通过传送指令向输出锁存器写入值，但由于输出缓冲器关闭，引脚状态不发生变化。因此，对于既包含输入又包含输出的端口，可以使用字节写入。

写入输出锁存器的数据被一直保留，直到再次有数据被写入输出锁存器。

产生复位信号后，输出锁存器的数据。

4.4.2 输入/输出端口的读取

(1) 输出模式

通过传送指令读取输出锁存器的内容。输出锁存器的内容不发生变化。

(2) 输入模式

通过传送指令读取引脚状态。输出锁存器的内容不发生变化。

4.4.3 输入/输出端口的运算

(1) 输出模式

对输出锁存器内容进行运算，结果写入输出锁存器。输出锁存器的内容从引脚输出。

写入输出锁存器的数据被一直保留，直到再次有数据被写入输出锁存器。

产生复位信号后，输出锁存器的数据。

(2) 输入模式

读取引脚电平，并对其内容进行运算。运算结果写入输出锁存器，但由于输出缓冲器关闭，引脚状态不发生变化。因此，对于既包含输入又包含输出的端口，可以使用字节写入。

产生复位信号后，输出锁存器的数据。

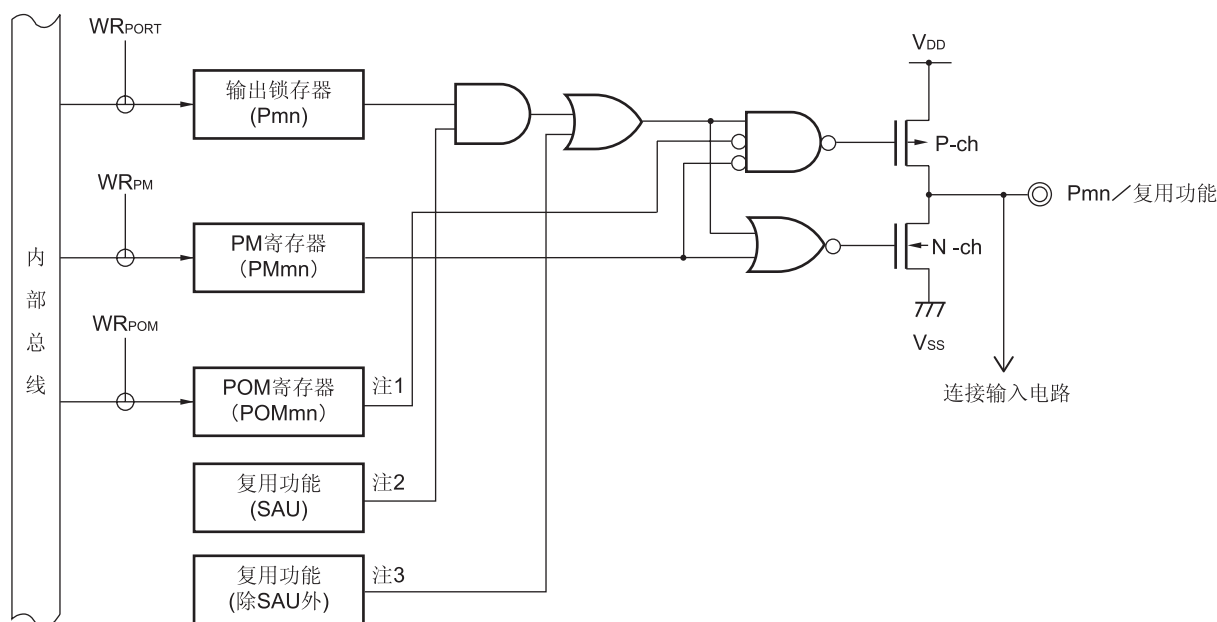
4.5 使用复用功能时相关寄存器的设置

4.5.1 使用复用功能时的基本思路

对于复用为模拟输入的引脚，必须在最初通过端口模式控制寄存器 0 (PMC0) 设定为用作模拟输入还是用作数字输入/输出。

用作数字输入/输出的引脚的输出电路基本结构如图 4-7 所示。SAU 功能（复用为端口输出锁存器的输出）的输出被输入至 AND 门。AND 门的输出被输入至 OR 门。SAU 以外的复用功能（TAU、时钟/蜂鸣器输出等）的输出也连接至 OR 门的输入。将这样的引脚用作端口功能或者复用功能时，必须保证不使用的复用功能不会阻碍要使用的功能的输出。这种情况下的基本设置思路如表 4-3 所示。

图 4-7. 引脚输出电路的基本结构



注 1. 没有 POM 寄存器时，认为该信号是 Low(0)。

2. 没有复用功能时，认为该信号是 High(1)。

3. 没有复用功能时，认为该信号是 Low(0)。

备注 m: 端口编号 (m = 0, 4, 12, 13), n: 位编号 (n = 0 至 7)

表 4-3. 基本的设置思路

所使用引脚的输出功能	不使用的复用功能的输出设置		
	端口功能	SAU的输出功能	SAU以外的输出功能
端口输出功能	-	输出High(1)	输出Low(0)
SAU的输出功能	High(1)	—	输出Low(0)
SAU以外的输出功能	Low(0)	输出High(1)	输出Low(0) ^注

注 除了 SAU，引脚还复用作其他多个输出功能的情况下，必须事先将不使用的复用功能的输出设置为 Low(0)。具体设置方法请参阅 4.5.2 不使用复用功能的输出功能时寄存器的设置。

4.5.2 不使用复用功能的输出功能时寄存器的设置

不使用引脚的复用功能输出时，请按照以下所示设置寄存器。此外，适用外围输入/输出重定向功能时，可以通过设置外围输入/输出重定向寄存器(PIOR)，切换为其他引脚输出。这样，可以使用分配给对应引脚的端口功能或其他复用功能。

(1) $Sop = 1 / TxDq = 1$ (不使用 SAU 串行输出(Sop/TxDq)时)

不使用串行输出(Sop/TxDq) (例如仅使用 SAU 串行输入) 的情况下，请将串行输出允许寄存器 m (SOEm)的对应于不使用输出的位设置为 0 (禁止输出)，将串行输出寄存器(SOm)的 SOMn 位设置为 1 (High)。即设置为与初始状态相同的值。

(2) $SCKp = 1$ (不使用 SAU 的通道 n 时)

不使用 SAU 的情况下，请将串行通道允许状态寄存器 m (SEm)的位 n (SEmn)设置为 0 (停止操作状态)，将串行输出允许寄存器 m (SOEm)的对应于不使用输出的位设置为 0 (禁止输出)，将串行输出寄存器(SOm)的 SOMn 位设置为 1 (High)。即设置为与初始状态相同的值。

(3) $TOmn = 0$ (不使用 TAU 的通道 n 输出时)

不使用 TAU 的 TOmn 输出的情况下，请将定时器输出允许寄存器 0 (TOE0)的对应于不使用输出的位设置为 0 (禁止输出)，将定时器输出寄存器 0 (TO0)的位设置为 0 (Low)。即设置为与初始状态相同的值。

(4) $PCLBUZn = 0$ (不使用时钟输出/蜂鸣器输出时)

不使用时钟输出/蜂鸣器输出的情况下，请将时钟输出选择寄存器 n (CKSn)的 PCLOEn 位设置为 0 (禁止输出)。即设置为与初始状态相同的值。

4.5.3 使用端口功能以及复用功能的寄存器设置例

使用端口功能以及复用功能的寄存器设置例如表 4-4 所示。请按照表 4-4 设置用于控制端口功能的寄存器。此外，关于表 4-4 的标记，请参阅以下备注。

备注 - : 不对应

x : 忽略

PIOR: 外围输入/输出重定向寄存器

POM0: 端口输出模式寄存器 0

PMC0: 端口模式控制寄存器 0

PMn: 端口模式寄存器 n (n = 0, 4)

Pm: 端口输出锁存器 (m = 0, 4, 12, 13)

可以通过设置外围输入/输出重定向寄存器(PIOR)来分配 () 内的功能。

表 4-4. 使用引脚功能时寄存器和输出锁存器的设置示例(1/2)

引脚名称	使用功能		PIOR	POM0	PMC0	PMn	Pm	复用功能输出	
	功能名称	输入/输出						SAU输出功能	SAU以外
P00	P00	输入	—	×	—	1	×	×	—
		输出	—	0	—	0	0/1	TxD0 / SO00 = 1	—
		N沟开漏输出	—	1	—	0	0/1		
	SO00	输出	—	0	—	0	1	×	—
	TXD0	输出	—	0/1	—	0	1	×	—
	INTP1	输入	PIOR2=0	×	—	1	×	×	—
P01	P01	输入	—	×	0	1	×	×	—
		输出	—	0	0	0	0/1	×	—
	ANI0	模拟输入	—	×	1	1	×	×	—
	SI00	输入	—	×	0	1	×	×	—
	RxD0	输入	—	×	0	1	×	×	—
	KR2	输入	—	×	0	1	×	×	—
P02	P02	输入	—	—	0	1	×	×	×
		输出	—	—	0	0	0/1	SCK00 = 1	PCLBUZ0 = 0
	ANI1	模拟输入	—	—	1	1	×	×	×
	SCK00	输入	—	—	0	1	×	×	×
		输出	—	—	0	0	1	×	PCLBUZ0 = 0
	PCLBUZ0	输出	PIOR0=0	—	0	0	0	SCK00 = 1	×
P03	P03	输入	—	—	0	1	×	—	×
		输出	—	—	0	0	0/1	—	TO00 = 0
	ANI2	模拟输入	—	—	1	1	×	—	×
	TO00	输出	—	—	0	0	0	—	×
	KR4	输入	—	—	0	1	×	—	×
	(INTP1)	输入	PIOR2=1	—	0	1	×	—	×
P04	P04	输入	—	—	0	1	×	—	×
		输出	—	—	0	0	0/1	—	TO01 = 0
	ANI3	模拟输入	—	—	1	1	×	—	×
	TI01	输入	PIOR1=0	—	0	1	×	—	×
	TO01	输出	PIOR1=0	—	0	0	0	—	×
	KR5	输入	—	—	0	1	×	—	×
P40	P40	输入	—	—	—	1	×	—	×
		输出	—	—	—	0	0/1	—	(PCLBUZ0) = 0 (TO01) = 0
	KR0	输入	—	—	—	1	×	—	×
	(PCLBUZ0)	输出	PIOR0=1	—	—	0	0	—	(TO01) = 0
	(TI01)	输入	PIOR1=1	—	—	1	×	—	×
	(TO01)	输出	PIOR1=1	—	—	0	0	—	(PCLBUZ0) = 0

表 4-4. 使用引脚功能时寄存器和输出锁存器的设置示例(2/2)

引脚 名称	使用功能		PIOR	POM0	PMC0	PMn	Pm	备注
	功能名称	输入/输出						
P125	P125	输入	—	—	—	—	x	选项字节000C1H
	KR1	输入	—	—	—	—	x	PORTSELB = 0
	RESET	输入	—	—	—	—	x	选项字节000C1H PORTSELB = 1
P137	P137	输入	—	—	—	—	x	—
	TI00	输入	—	—	—	—	x	—
	INTP0	输入	—	—	—	—	x	—

4.6 使用端口功能时的注意事项

4.6.1 端口寄存器 n (Pn)的位操作指令的相关注意事项

对同时具有输入和输出功能的端口执行位操作指令时，除了操作目标位以外，非操作对象的输入端口的输出锁存器值也可能被改写。

因此，将端口由输入模式切换为输出模式前，建议改写输出锁存器。

示例 当 P00 为输出端口、P01 至 P04 为输入端口（所有引脚状态均为高电平）且端口 0 的端口锁存器值为 00H 时，通过位操作指令将输出端口 P00 的输出从低电平改为高电平，则端口 0 的输出锁存器值成为 FFH。

说明： PMmn 位为 1 的端口，其写入或读取 Pn 寄存器的对象分别是输出锁存器和引脚状态。
在 RL78 单片机中，位操作指令按以下顺序执行。

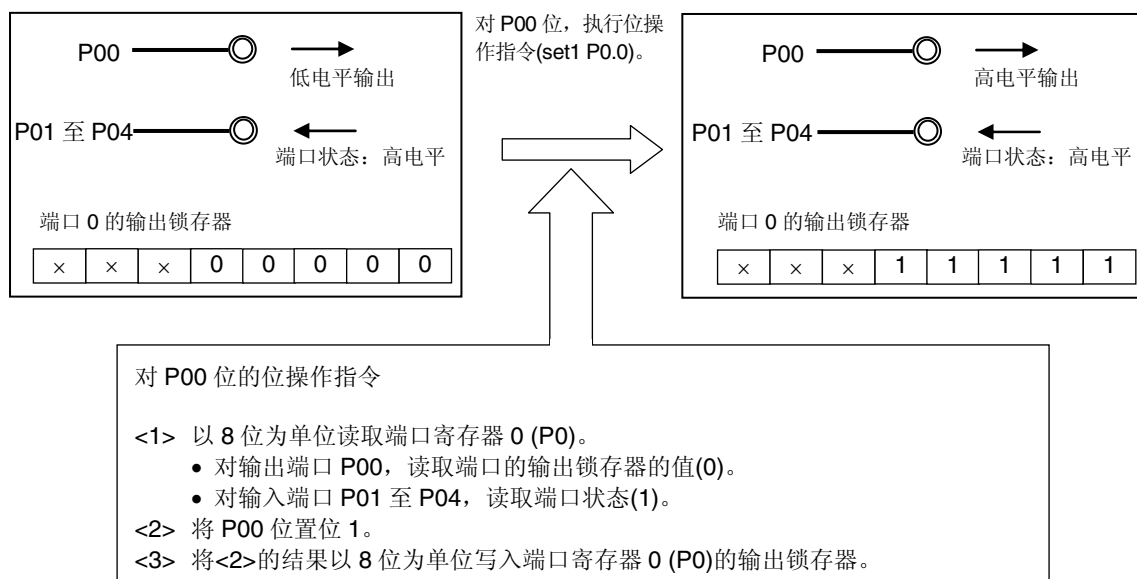
- <1> 以 8 位为单位读取 Pn 寄存器。
- <2> 操作目标位。
- <3> 以 8 位为单位写入 Pn 寄存器。

在步骤<1>中，输出端口 P00 的输出锁存器值 0 被读取，同时，输入端口 P01 至 P04 的引脚状态也被读取。如果 P01 至 P04 的引脚状态此时为高电平，则读取值为 EH。

步骤<2>的操作使该值变为 FH。

步骤<3>的操作将 FH 写入输出锁存器。

图 4-8. 位操作指令(P00)



4.6.2 引脚设置时的注意事项

对于被指定了多项功能的输出引脚，当选择了其中一项功能输出时，必须将其他不使用的复用功能设置为初始状态，以避免和所选的功能产生冲突。该注意事项也适用于通过外围输入/输出重定向寄存器 (PIOR)指定的功能。有关复用输出功能，请参阅 **4.5 使用复用功能时相关寄存器的设置**。

此外，输入引脚不需要特殊的设置，因为其复用输出功能被禁用(缓冲器输出为Hi-Z)。

建议停止不使用的外围功能，以降低功耗。

第五章 时钟发生电路

5.1 时钟发生电路的功能

时钟发生电路产生用于 CPU 和外围硬件的时钟。

可以选择以下三种系统时钟和时钟振荡电路。

(1) 主系统时钟

<1> 高速片上振荡器

通过选项字节 (000C2H) 可从以下范围内选择振荡频率: $f_{IH} = 20/10/5/2.5/1.25$ MHz (典型值)。在解除复位后, CPU 一定使用此高速片上振荡器时钟开始工作。可以通过执行 STOP 指令来停止振荡。

使用高速片上振荡器频率选择寄存器 (HOCODIV)可更改通过选项字节指定的频率。关于频率的详情, 请参阅图 5-3. 高速片上振荡器频率选择寄存器 (HOCODIV) 的格式。

通过下表的选项字节和高速片上振荡器频率选择寄存器 (HOCODIV)可指定高速片上振荡器的频率。

电源电压	振荡器频率 (MHz)				
	1.25	2.5	5	10	20
$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	√	√	√	√	√
$2.4\text{ V} \leq V_{DD} < 2.7\text{ V}^{\text{注}}$	√	√	√	—	—

注 由于包括了可选择上电复位(SPOR)电路的检测电压(V_{SPOR}), 请在 2.57(TYP.)至 5.5 V 的电压范围内使用。

(2) 低速片上振荡器时钟

该电路产生 $f_{IL} = 15\text{ kHz}$ (TYP.)的振荡时钟。

低速片上振荡器时钟不能用作 CPU 时钟。

只有看门狗定时器运行于低速片上振荡器时钟。

当选项字节(000C0H)的位 4 (WDTON)被设置为 1 时该时钟工作。

但是, 当 WDTON = 1 且选项字节(000C0H)的位 0(WDSTBYON)为 0 时, 执行 HALT 或 STOP 指令后低速片上振荡器停止振荡。

备注 f_{IH} : 高速片上振荡器时钟频率

f_{IL} : 低速片上振荡器时钟频率

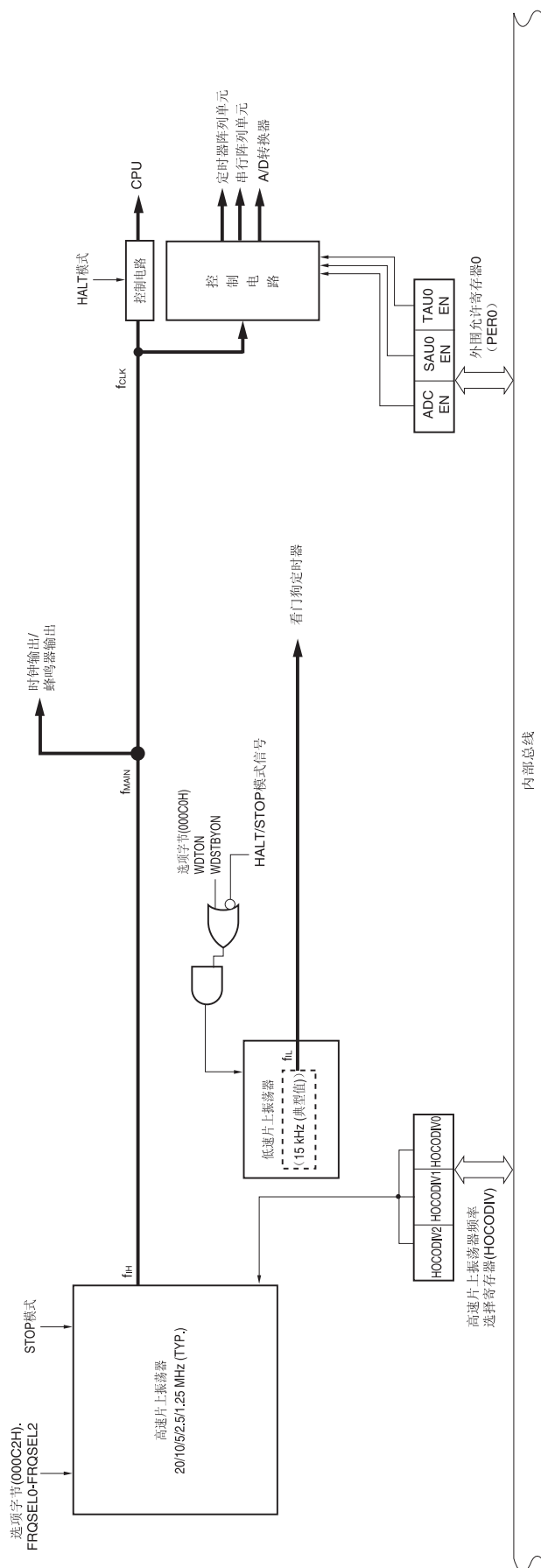
5.2 时钟发生电路的配置

时钟发生电路包括以下硬件。

表 5-1. 时钟发生电路的配置

项目	配置
控制寄存器	外围允许寄存器 0 (PER0) 高速片上振荡器频率选择寄存器(HOCODIV)
振荡电路	高速片上振荡器 低速片上振荡器

图 5-1. 时钟发生电路的框图



备注 f_{IH} : 高速片上振荡器时钟频率

f_{MAIN} : 主系统时钟频率

f_{CLK} : CPU/外围硬件时钟频率

f_{IL} : 低速片上振荡器时钟频率

5.3 控制时钟发生电路的寄存器

由以下寄存器来控制时钟发生电路。

- 外围允许寄存器 0 (PER0)
- 高速片上振荡器频率选择寄存器(HOCODIV)

5.3.1 外围允许寄存器 0 (PER0)

该寄存器用于允许或禁止向外围硬件供应时钟。通过停止向未使用的硬件供应时钟，可以降低功耗和噪声。

使用该寄存器控制的下列外围功能时，进行外围功能初始设置前，要将各功能对应的位置位(1)。

- A/D 转换器
- 串行阵列单元 0
- 串行阵列单元 0

使用 1 位或 8 位存储器操作指令设置 PER0 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 5-2. 外围允许寄存器 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	0	0	ADCEN	0	0	SAU0EN	0	TAU0EN

ADCEN	A/D 转换器输入时钟供应的控制
0	停止输入时钟的供应。 • 不可写入用于 A/D 转换器的 SFR。 • A/D 转换器处于复位状态。
1	允许输入时钟的供应。 • 可以读取和写入用于 A/D 转换器的 SFR。

SAU0EN	串行阵列单元 0 输入时钟供应的控制
0	停止输入时钟的供应。 • 不可写入用于串行阵列单元 0 的 SFR。 • 串行阵列单元 0 处于复位状态。
1	允许输入时钟的供应。 • 可以读取和写入用于串行阵列单元 0 的 SFR。

TAU0EN	定时器阵列单元输入时钟供应的控制
0	停止输入时钟的供应。 • 不可写入用于定时器阵列单元的 SFR。 • 定时器阵列单元处于复位状态。
1	允许输入时钟的供应。 • 可以读取和写入用于定时器阵列单元的 SFR。

5.3.2 高速片上振荡器频率选择寄存器(HOCODIV)

通过高速片上振荡器频率选择寄存器(HOCODIV)，可以改变由选项字节(000C2H)设置的高速片上振荡器的频率。

使用 8 位存储器操作指令设置 HOCODIV 寄存器。

产生复位信号后，该寄存器成为选项字节(000C2H) FRQSEL2 - FRQSEL0 设定的值。

图 5-3. 高速片上振荡器频率选择寄存器(HOCODIV) 的格式

地址: F00A8H 复位后: 选项字节(000C2H) FRQSEL2 - FRQSEL0 的设定值 R/W

符号	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV 2	HOCODIV 1	HOCODIV 0

HOCODIV 2	HOCODIV 1	HOCODIV 0	高速片上振荡器时钟频率的选择
0	0	1	20 MHz
0	1	0	10 MHz
0	1	1	5 MHz
1	0	0	2.5 MHz
1	0	1	1.25 MHz
其他			禁止设置

- 注意事项
1. 在更改频率之前和之后，在可操作电压范围内设置 HOCODIV 寄存器。
 2. 通过 HOCODIV 寄存器更改频率时，须经过以下的时间后频率才会变化。
 - 以更改前的频率工作最多 3 个时钟
 - 以更改后的频率等待最多 3 个 CPU/外围硬件时钟

5.4 系统时钟振荡电路

5.4.1 高速片上振荡器

R7F0C80112ESP, R7F0C80212ESP 内置有高速片上振荡器。通过选项字节(000C2H)可从以下范围中选择频率：20、10、5、2.5 或 1.25 MHz。解除复位后高速片上振荡器自动开始振荡。

5.4.2 低速片上振荡器

R7F0C80112ESP, R7F0C80212ESP 内置有低速片上振荡器。

低速片上振荡器时钟只能用作看门狗定时器的时钟。低速片上振荡器时钟不能用作 CPU 时钟。

当选项字节(000C0H)的位 4 (WDTON)被设置为 1 时该时钟工作。

看门狗定时器停止工作时，低速片上振荡器停止振荡。

5.5 时钟发生电路的操作

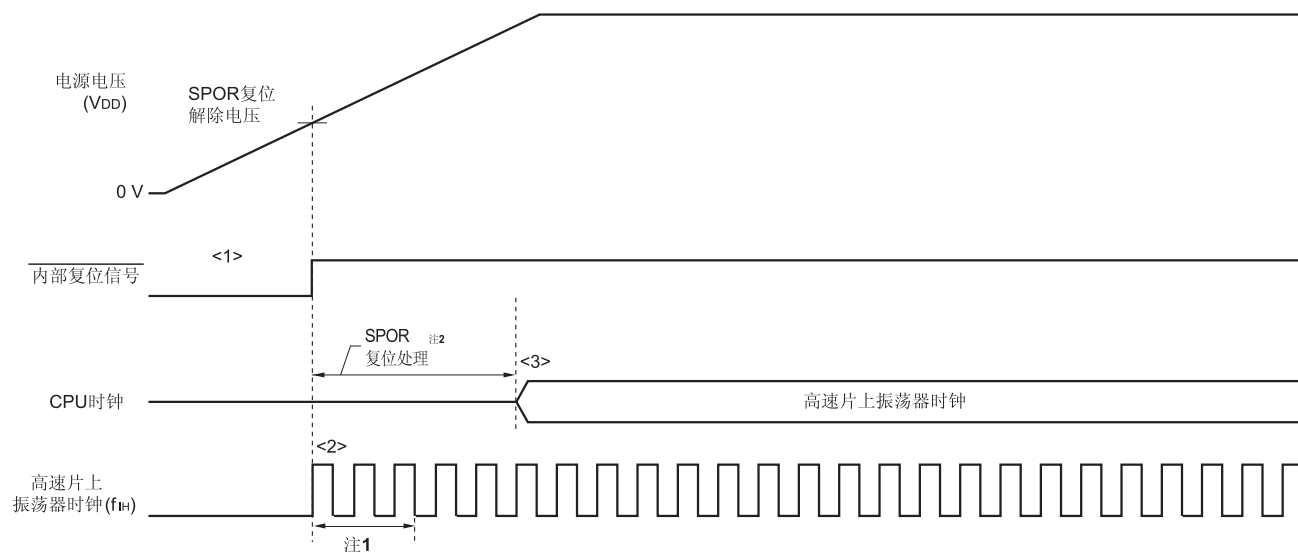
时钟发生电路产生如下所示时钟，并控制例如待机模式等 CPU 的操作模式(参阅图 5-1)。

- 主系统时钟 f_{MAIN}
- 高速片上振荡器时钟 f_{IH}
- 低速片上振荡器时钟 f_{IL}
- CPU/外围硬件时钟 f_{CLK}

在 R7F0C80112ESP, R7F0C80212ESP 中，解除复位后，CPU 将在高速片上振荡器开始输出时开始工作。

电源投入时，时钟发生电路的操作如图 5-4 所示。

图 5-4. 电源投入时时钟发生电路的操作



<1> 投入电源后, 可选择上电复位 (SPOR) 电路产生内部复位信号。

<2> 当电源电压超过 SPOR 电路的检测电压时, 复位被解除, 高速片上振荡器自动开始振荡。

<3> 解除复位后, 执行等待电压稳定和 SPOR 复位处理后, CPU 以高速片上振荡器时钟开始运行。

注 1. 高速片上振荡器时钟的振荡精度稳定等待时间包含在复位处理时间内。

2. 关于 SPOR 复位处理时间, 请参照第十五章 可选择上电复位电路。

5.6 控制时钟

5.6.1 高速片上振荡器的设置示例

解除复位后，CPU/外围硬件时钟 (f_{CLK}) 总是开始工作于高速片上振荡器时钟。通过选项字节(000C2H)的FRQSEL0 至 FRQSEL2，可以选择高速片上振荡器的频率。可通过高速片上振荡器的频率选择寄存器 (HOCODIV)来更改频率。

[选项字节设置]

地址：000C2H

选项字节 (000C2H)	7	6	5	4	3	2	1	0
	1	1	1	0	1	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL2	FRQSEL1	FRQSEL0	高速片上振荡器的频率
0	0	1	20 MHz
0	1	0	10 MHz
0	1	1	5 MHz
1	0	0	2.5 MHz
1	0	1	1.25 MHz
其他			禁止设置

[高速片上振荡器频率选择寄存器(HOCODIV)设置]

地址：F00A8H

HOCODIV	7	6	5	4	3	2	1	0
	0	0	0	0	0	HOCODIV 2	HOCODIV 1	HOCODIV 0

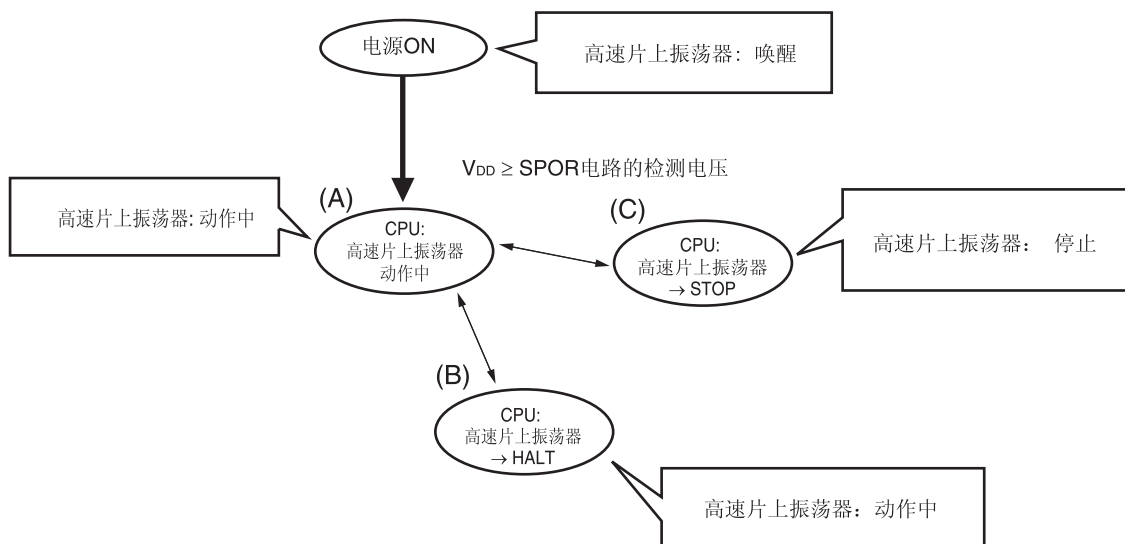
HOCODIV 2	HOCODIV 1	HOCODIV 0	选择频率
0	0	1	20 MHz
0	1	0	10 MHz
0	1	1	5 MHz
1	0	0	2.5 MHz
1	0	1	1.25 MHz
其他			禁止设置

- 注意事项
1. 在更改频率之前和之后，在可操作电压范围内设置 HOCODIV 寄存器。
 2. 通过 HOCODIV 寄存器更改频率时，须经过以下的时间后频率才会变化。
 - 以更改前的频率工作最多 3 个时钟
 - 以更改后的频率等待最多 3 个 CPU/外围硬件时钟

5.6.2 CPU 时钟状态转换图

本产品的 CPU 时钟状态转换图如图 5-5 所示。

图 5-5. CPU 时钟状态转换图



CPU 时钟转换和 SFR 寄存器的设置示例如表 5-2 所示。

表 5-2. CPU 时钟转换和 SFR 寄存器的设置示例(1/3)

(1) • 当 CPU 工作于高速片上振荡器时钟(A)时切换为 HALT 模式(B)

状态转换	设置
(A) → (B)	执行 HALT 指令

(2) • 当 CPU 工作于高速片上振荡器时钟(A)时切换为 STOP 模式(C)

(设置顺序) 

状态转换	设置	
(A) → (C)	停止无法工作于 STOP 模式的外围功能	执行 STOP 指令

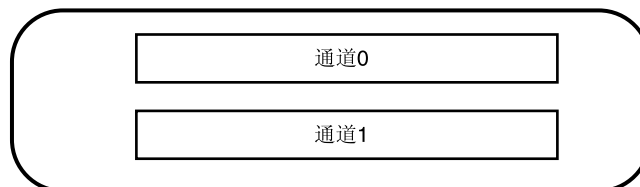
备注 表 5-2 的(A)至(C)对应于图 5-5 的(A)至(C)。

第六章 定时器阵列单元

定时器阵列单元具有 2 个 16 位定时器。

每个 16 位定时器被称为一个通道，可用作独立的定时器。另外，两个或更多通道可用来构建一个高精度的定时器。

定时器阵列单元



关于每个功能的详情，请参阅下表。

单通道操作功能	多通道联动操作功能
• 间隔定时器 (→ 参阅 6.7.1) • 方波输出 (→ 参阅 6.7.1) • 外部事件计数器 (→ 参阅 6.7.2) • 分频器[※] (→ 参阅 6.7.3) • 输入脉冲间隔测量 (→ 参阅 6.7.4) • 输入信号的高/低电平宽度的测量 (→ 参阅 6.7.5) • 延迟计数器 (→ 参阅 6.7.6)	• 单触发脉冲输出 (→ 参阅 6.8.1) • PWM 输出 (→ 参阅 6.8.2)

可以将通道 1 的 16 位定时器用作两个 8 位定时器（高位和低位）。用作 8 位定时器时可用的功能如下：

- 间隔定时器/方波输出
- 外部事件计数器(仅限低 8 位定时器)
- 延迟计数器(仅限低 8 位定时器)
- PWM 输出(仅限低 8 位定时器)

注 仅限通道 0。

6.1 定时器阵列单元功能

定时器阵列单元具有以下功能：

6.1.1 单通道操作功能

通过单独操作一个通道，从而不受其他通道操作模式影响地达到以下目的。

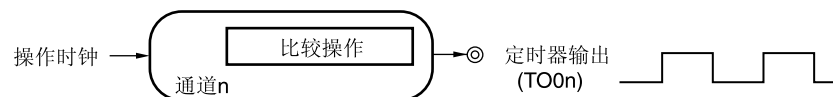
(1) 间隔定时器

单元中的各个定时器都可用作按照固定间隔产生中断(INTTM0n)的基准定时器。



(2) 方波输出

每当产生 INTTM0n 中断时执行交替操作，并从定时器输出引脚(TO0n)输出占空比 50% 的方波。



(3) 外部事件计数器

单元中的各个定时器可以用作事件计数器，当输入至定时器输入引脚(TI0n)的信号的有效边沿数达到指定值时，产生一个中断。



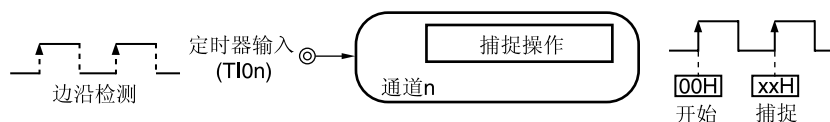
(4) 分频器功能

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



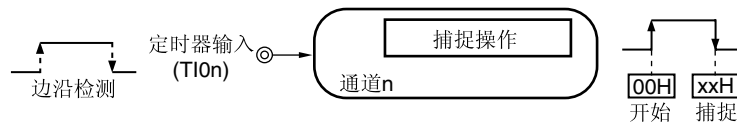
(5) 输入脉冲间隔测量

从输入至定时器引脚(TI0n)的脉冲信号的有效边沿开始计数。并且在下一个脉冲的有效边沿捕捉定时器的计数值。从而测量输入脉冲的间隔。



(6) 输入信号的高/低电平宽度的测量

从输入至定时器输入引脚(TI0n)的信号的一个边沿开始计数，在另一个边沿捕捉计数值。从而测量输入信号的高电平或低电平宽度。

**(7) 延迟计数器**

从输入至定时器输入引脚(TI0n)的信号的有效边沿开始计数，并在经过任意延迟期间后产生中断。



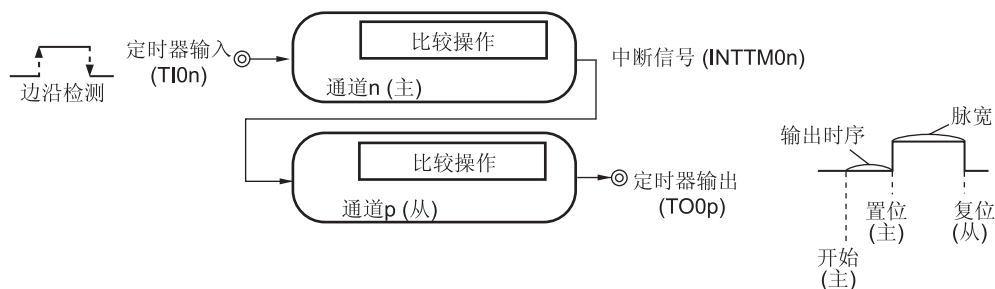
备注 n: 通道编号 (n = 0, 1)

6.1.2 多通道联动操作功能

通过联合使用主通道（主要用于控制周期的基准定时器）和从属通道（以主通道为基准而工作的定时器），来实现以下功能。

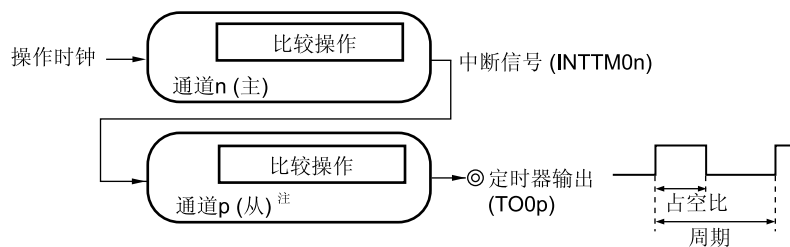
(1) 单触发脉冲输出

将两个通道成组使用，生成可任意设置输出时序和脉冲宽度的单触发脉冲。



(2) PWM（脉冲宽度调制）输出

将两个通道成组使用，生成可任意设置周期和占空比的脉冲。



注 通过通道 1 的低位可以实现该操作。

注意事项 使用多通道联动操作功能时须遵守以下规则。

详情请参阅 6.4.1 多通道联动操作功能的基本规则。

备注 n: 通道编号 (n = 0, 1)

p: 从属通道编号 (p = 0)

6.1.3 8 位定时器操作功能（仅限通道 1）

通过 8 位定时器操作功能，可以将 16 位定时器通道作为两个 8 位定时器通道来使用。该功能仅用于通道 1。

注意事项 使用 8 位定时器操作功能时须遵守以下规则。

详情请参阅 6.4.2 8 位定时器操作功能的基本规则（仅限通道 1）。

6.2 定时器阵列单元的配置

定时器阵列单元包含以下硬件。

表 6-1. 定时器阵列单元的配置

项目	配置
定时器/计数器	定时器/计数器寄存器 0n (TCR0nH, TCR0nL)
寄存器	定时器数据寄存器 0n (TDR0nH, TDR0nL)
定时器输入	TI00, TI01
定时器输出	TO00, TO01, 输出控制电路
控制寄存器	<单元设置块的寄存器> • 外围允许寄存器 0 (PER0) • 定时器时钟选择寄存器 0 (TPS0) • 定时器通道允许状态寄存器 0 (TE0, TEH0) • 定时器通道开始寄存器 0 (TS0, TSH0) • 定时器通道停止寄存器 0 (TT0, TTH0) • 定时器输出允许寄存器 0 (TOE0) • 定时器输出寄存器 0 (TO0) • 定时器输出电平寄存器 0 (TOL0) • 定时器输出模式寄存器 0 (TOM0) <每个通道的寄存器> • 定时器模式寄存器 0n (TMR0nH, TMR0nL) • 定时器状态寄存器 0n (TSR0n) • 噪声滤波器允许寄存器 1 (NFEN1) • 端口模式控制寄存器 0 (PMC0) • 端口模式寄存器 0, 4 (PM0, PM4) • 端口寄存器 0, 4 (P0, P4)

备注 n: 通道编号 (n = 0, 1)

定时器阵列单元的框图如图 6-1 和图 6-2 所示。

图 6-1. 定时器阵列单元的整体配置

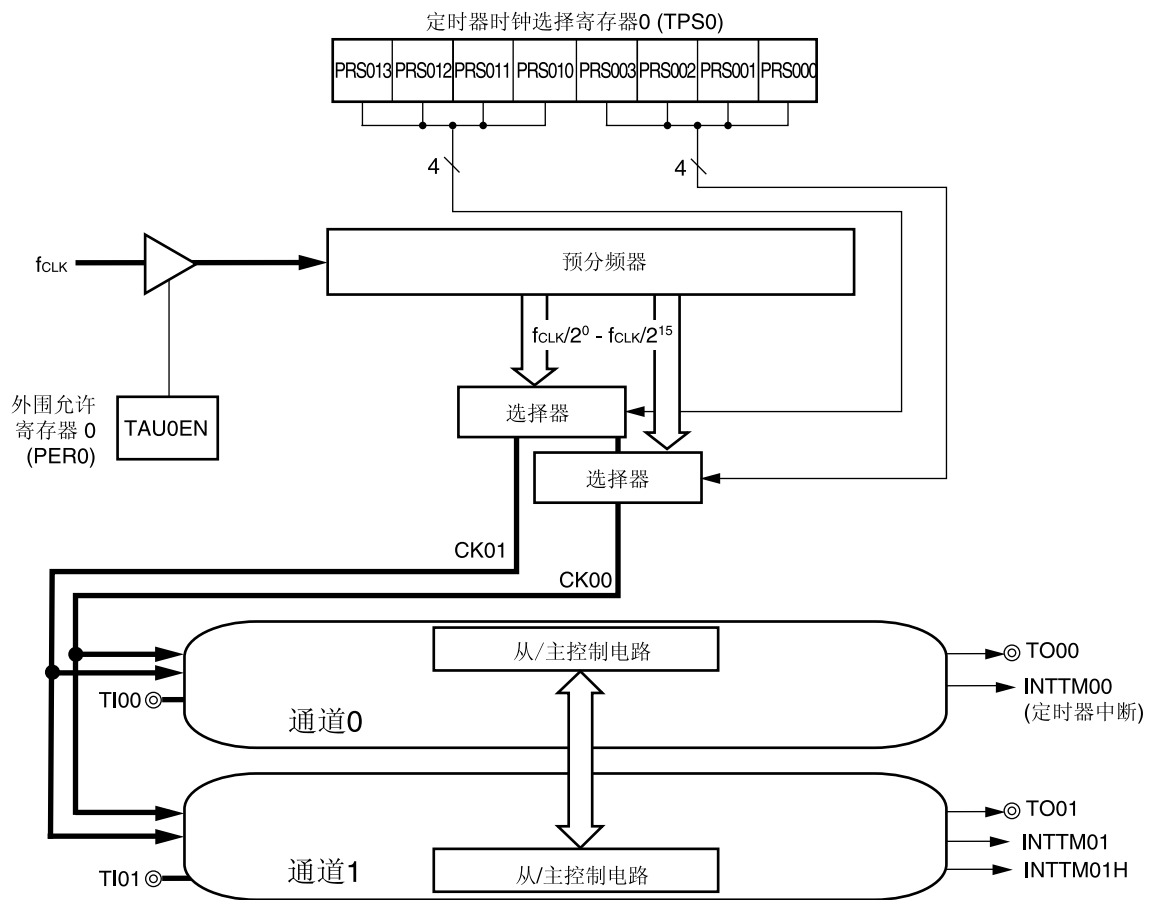
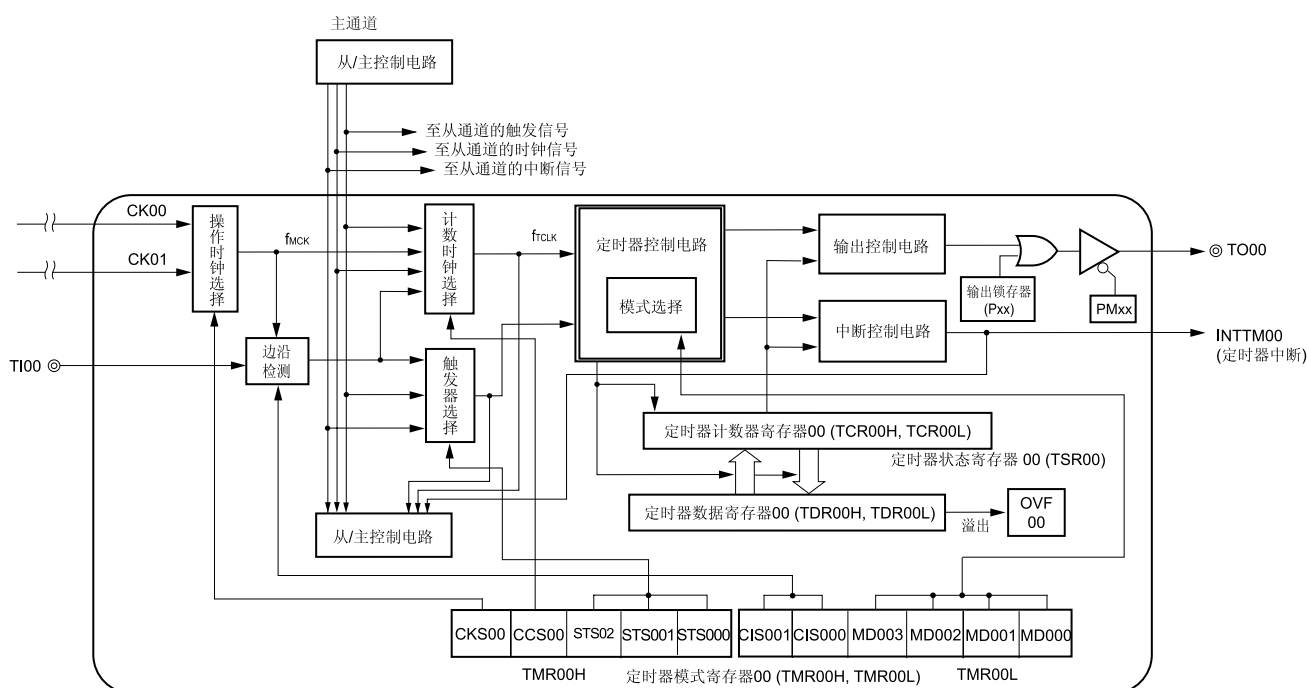
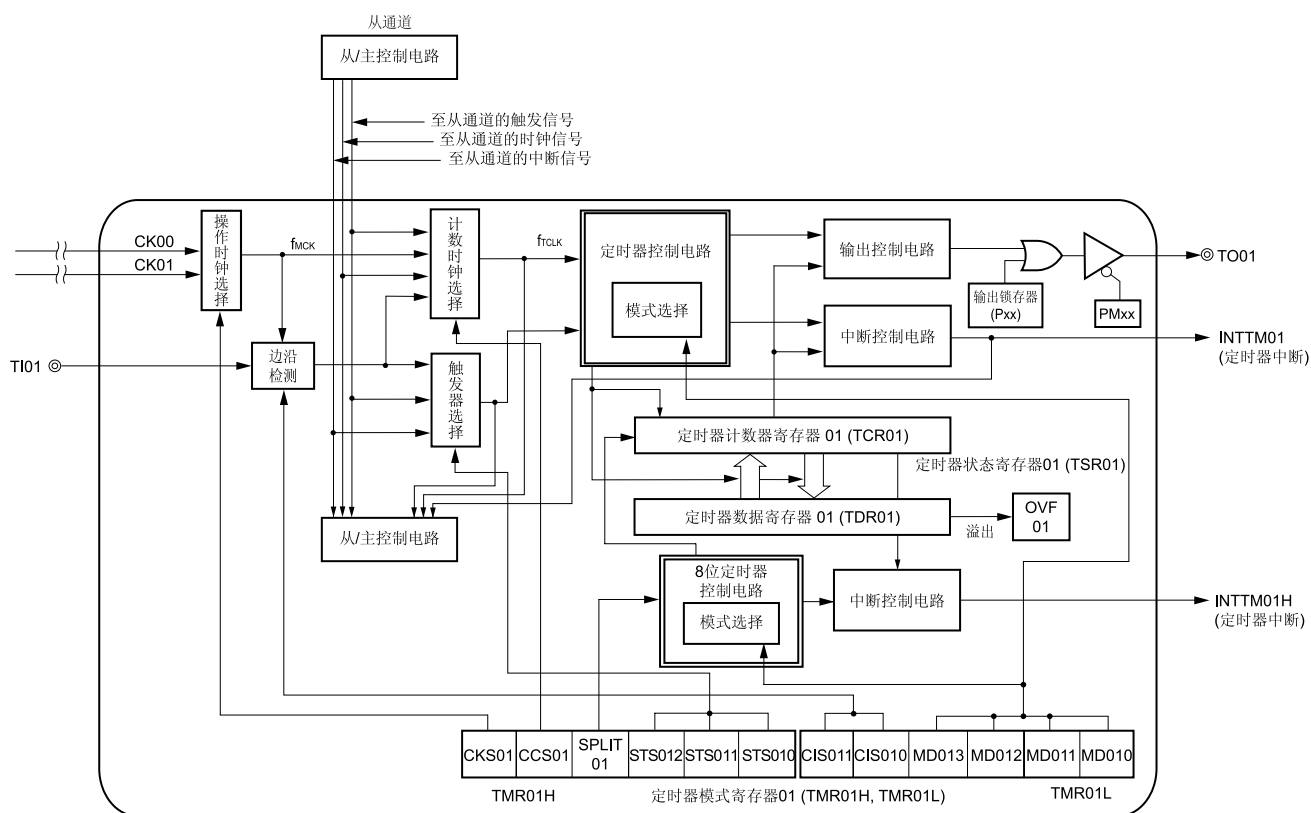


图 6-2. 定时器阵列单元的通道内部框图

(a) 通道 0



(b) 通道 1



6.2.1 定时器/计数器寄存器 0n (TCR0n)

TCR0n 寄存器由 2 个 8 位只读寄存器 TCR0nH 和 TCR0nL 组成，他们用于对计数时钟进行计数。

对 TCR0n 寄存器进行读取操作时，请连续存取 TCR0nH 和 TCR0nL 寄存器。

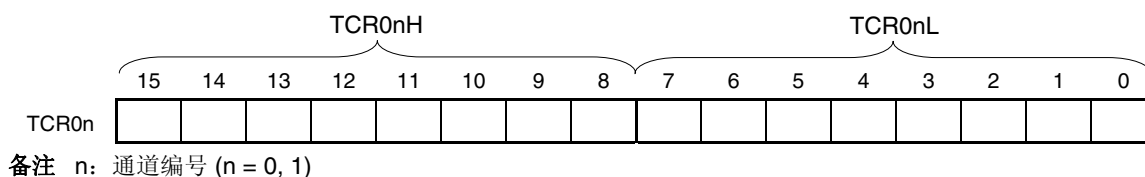
与计数时钟的上升沿同步，进行递增或递减计数。

通过定时器模式寄存器 0n (TMR0n)的 MD0n3 位至 MD0n0 位来选择操作模式，从而切换至递增或递减计数（参阅

6.3.3 定时器模式寄存器 0n (TMR0n)）。

图 6-3. 定时器/计数器寄存器 0n (TDR0n)的格式 (n = 0, 1)

地址: F0180H (TCR00L), F0181H (TCR00H) 复位后: FFH R
: F0182H (TCR01L), F0183H (TCR01H)



按照从 TCR0nL 寄存器到 TCR0nH 寄存器的顺序，连续对其进行读取操作。在进行连续读取操作时，如果插入了对 TCR0nL 寄存器的读取操作，将发生异常。

注意事项 通过执行 DI 指令设定为禁止中断的状态下，对 TCR0nH 和 TCR0nL 寄存器执行连续读取操作。

可以通过读取定时器计数器寄存器 0n (TCR0n)来读取计数值。

下列情况中计数值被设置为 FFFFH。

- 产生复位信号后
- 外围允许寄存器 0(PER0)的 TAU0EN 位被清除时
- 在 PWM 输出模式下完成从属通道计数时
- 在延迟计数模式下完成计数时
- 在单触发脉冲输出模式下完成主/从属通道计数时

下列情况中计数值被清除为 0000H。

- 在捕捉模式下输入开始触发时
- 在捕捉模式下完成捕捉时

注意事项 1. 即使在 TCR0n 寄存器被读取时，也不会将计数值捕捉至定时器数据寄存器 0n (TDR0n)。

2. 在 8 为定时器模式(SPLIT = 1)下使用通道 1 时，禁止读取 TCR01H 和 TDR01H 寄存器。

如下所示，TCR0n 寄存器读取值因操作模式变化和操作状态而异。

表 6-2. 各种操作模式下定时器/计数器寄存器 0n (TCR0n)的读取值

操作模式	计数模式	定时器/计数器寄存器0n (TCR0n)的读取值 ^注			
		解除复位后操作模式发生变化时的值	暂停计数操作(TT0n = 1)时的值	暂停计数操作后(TT0n = 1)操作模式发生变化时的值	一次计数后等待开始触发时的值
间隔定时器模式	递减计数	FFFFH	停止时的值	不定	—
捕捉模式	递增计数	0000H	停止时的值	不定	—
事件计数器模式	递减计数	FFFFH	停止时的值	不定	—
单计数模式	递减计数	FFFFH	停止时的值	不定	FFFFH
捕捉&单计数模式	递增计数	0000H	停止时的值	不定	TDR0n寄存器的捕捉值 + 1

注 表示当通道 n 为定时器操作停止状态(TE0n = 0)且计数操作刚被允许(TS0n = 1)时 TCR0n 寄存器的读取值。直到开始计数操作为止，该值被保持在 TCR0n 寄存器中。

备注 n: 通道编号 (n = 0, 1)

注意事项 1. 即使在 TCR0n 寄存器被读取时，也不会将计数值捕捉至定时器数据寄存器 0n (TDR0n)。
2. 在 8 位定时器模式(SPLIT = 1)下使用通道 1 时，禁止读取 TCR01H 和 TDR01H 寄存器。

6.2.2 定时器数据寄存器 0n (TDR0n)

TDR0 寄存器由 2 个 8 位寄存器(TCR0nH, TCR0nL)组成, 可以将其选择为捕捉或者比较功能。

使用定时器模式寄存器 0n (TMR0n)的 MD0n3 位至 MD0n0 位来选择操作模式, 从而切换至捕捉功能或比较功能。

用作比较寄存器时, TDR0nL 和 TDR0nH 寄存器的值可以随时更改。

对 TDR0n 寄存器进行存取时, 连续对 TDR0nH 和 TDR0nL 寄存器进行存取。

当处于 8 位定时器模式时 (即定时器模式寄存器 0n (TMR0n)的 SPLIT0n 位为 1 时), 能够以 8 位为单位改写 TDR0n 寄存器, 其中, 高 8 位用作 TDR0nH, 低 8 位用作 TDR0nL。

对于 TDR0nH 和 TDR0nL 寄存器进行读取数据或者写入数据操作时, 以下注意事项适用。

- 16 位定时器模式时 (当通道 1 和通道 3 的 TMR0nH 寄存器的位 3 (SPLIT0n)为 0 时)

写入 TDR0nH 和 TDR0nL 寄存器时, 必须按照从 TDR0nH 到 TDR0nL 的顺序连续执行写入。

读取 TDR0nH 和 TDR0nL 寄存器时, 必须按照从 TDR0nL 到 TDR0nH 的顺序连续执行读取。

在执行连续读取或连续写入操作的过程中, 如果插入对 TDR0nH 的写操作、对 TDR0nL 的读操作或者对 TCR0n 的读操作, 读取操作和写入操作将出现异常。

通过执行 DI 指令设定为禁止中断的状态下, 对 TDR0nH 和 TDR0nL 寄存器执行连续读取或连续写入操作。

- 8 位定时器模式时 (当通道 1 的 TMR0nH 寄存器的位 3 (SPLIT0n)为 1 时)

当处于 8 位定时器模式时, 能以 8 位为单位写入 TDR0nH 和 TDR0nL 寄存器。

读取 TDR0nH 寄存器时, 必须按照从 TDR0nL 到 TDR0nH 的顺序连续执行读取。

在执行连续读取操作的过程中, 如果插入对 TDR0nH 的写操作、对 TDR0nL 的读操作或者对 TCR0n 的读操作, 读取操作将出现异常。

通过执行 DI 指令设定为禁止中断的状态下, 对 TDR0nH 和 TDR0nL 寄存器执行连续读取操作。

备注 n: 通道编号 (n = 0, 1)

注意事项 在 8 位定时器模式(SPLIT = 1)下使用通道 1 时, 禁止读取 TCR01H 和 TDR01H 寄存器。

图 6-4. 定时器数据寄存器 0n (TDR0nH, TDR0nL)的格式 (n = 0)

地址: FFF18H (TDR00L), FFF19H (TDR00H), 复位后: 00H R/W

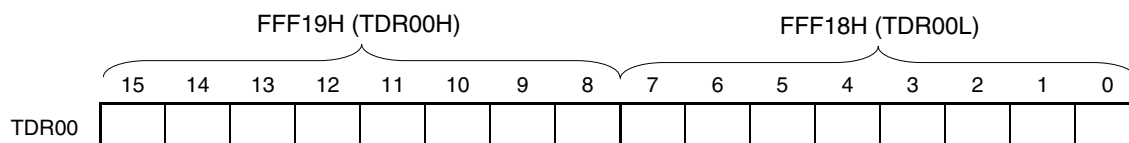
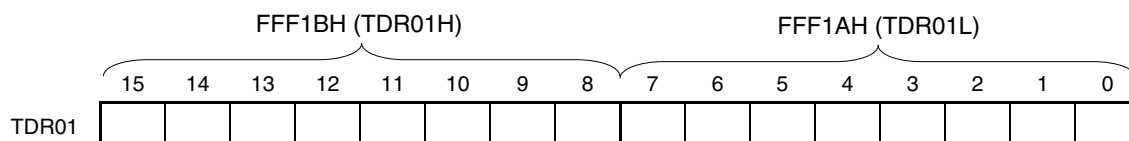


图 6-5. 定时器数据寄存器 0n (TDR0nH, TDR0nL)的格式 (n = 1)

地址: FFF1AH (TDR01L), FFF1BH (TDR01H), 复位后: 00H R/W



(i) 使用定时器数据寄存器 0n (TDR0nH, TDR0nL)作为比较寄存器时

从设置到 TDR0nH 和 TDR0nL 寄存器的值开始递减计数。当计数值达到 0000H 时, 产生中断信号(INTTM0n)。直到改写为止 TDR0n 寄存器将保持其值。

注意事项 当设置为比较功能时, 即使输入捕捉触发, TDR0n 寄存器也不会执行捕捉操作。

(ii) 使用定时器数据寄存器 0n (TDR0nH, TDR0nL)作为捕捉寄存器时

输入捕捉触发时, 定时器/计数器寄存器 0n (TCR0n)的计数值将被捕捉至 TDR0nH 和 TDR0nL 寄存器。可以选择 TIO_n 引脚的有效边沿作为捕捉触发。通过定时器模式寄存器 0n (TMR0n)进行选择。

备注 n: 通道编号 (n = 0, 1)

6.3 控制定时器阵列单元的寄存器

下列寄存器控制着定时器阵列单元。

- 外围允许寄存器 0 (PER0)
- 定时器时钟选择寄存器 0 (TPS0)
- 定时器通道允许状态寄存器 0 (TE0, TEH0)
- 定时器通道开始寄存器 0 (TS0, TSH0)
- 定时器通道停止寄存器 0 (TT0, TTH0)
- 定时器输出允许寄存器 0 (TOE0)
- 定时器输出寄存器 0 (TO0)
- 定时器输出电平寄存器 0 (TOL0)
- 定时器输出模式寄存器 0 (TOM0)
- 定时器模式寄存器 0n (TMR0nH, TMR0nL)
- 定时器状态寄存器 0n (TSR0n)
- 噪声滤波器允许寄存器 1 (NFEN1)
- 端口模式控制寄存器 0 (PMC0)
- 端口模式寄存器 0, 4 (PM0, PM4)
- 端口寄存器 0, 4 (P0, P4)

备注 n: 通道编号 (n = 0, 1)

6.3.1 外围允许寄存器 0 (PER0)

该寄存器用于允许或禁止向外围硬件供应时钟。通过停止向未使用的硬件供应时钟，可以降低功耗和噪声。

使用定时器阵列单元时，必须将该寄存器的位 0(TAU0EN)设置为 1。

使用 1 位或 8 位存储器操作指令设置 PER0 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 6-6. 外围允许寄存器 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	0	0	ADCEN	0	0	SAU0EN	0	TAU0EN

TAU0EN	定时器阵列单元输入时钟的控制
0	停止输入时钟的供应。 - 不可以写入用于定时器阵列单元的SFR。 - 定时器阵列单元处于复位状态。
1	允许输入时钟的供应。 可以读取/写入用于定时器阵列单元的SFR。

注意事项 1. 设置定时器阵列单元时，必须在 TAU0EN 位被设置为 1 的状态下先设置以下的寄存器。如果 TAU0EN = 0，向定时器阵列单元的控制寄存器的写入操作无效，而且，所有读取值均为初始值（噪声滤波器允许寄存器 1 (NFEN1)，端口模式寄存器 0、4 (PM0, PM4)，端口寄存器 0、4 (P0, P4)和端口模式控制寄存器 0 (PMC0) 除外）。

- 定时器时钟选择寄存器 0 (TPS0)
- 定时器通道允许状态寄存器 0 (TE0, TEH0)
- 定时器通道开始寄存器 0 (TS0, TSH0)
- 定时器通道停止寄存器 0 (TT0, TTH0)
- 定时器输出允许寄存器 0 (TOE0)
- 定时器输出寄存器 0 (TO0)
- 定时器输出电平寄存器 0 (TOL0)
- 定时器输出模式寄存器 0 (TOM0)
- 定时器模式寄存器 0n (TMR0nH, TMR0nL)
- 定时器状态寄存器 0n (TSR0n)

2. 必须将不确定位清除为 0。

6.3.2 定时器时钟选择寄存器 0 (TPS0)

TPS0 寄存器为 8 位寄存器，用于选择由外部预分频器供应至各通道共通的两种工作时钟(CK00, CK01)。

只有在以下情况下才可以在定时器操作期间改写 TPS0 寄存器。

如果需要改写 PRS000 至 PRS003 位($n = 0$ 至 3):

选择 CK00 作为工作时钟($CKS0n1 = 0$)的所有通道停止($TE0n = 0$)。

如果需要改写 PRS010 至 PRS013 位($n = 0$ 至 3):

选择 CK01 作为工作时钟($CKS0n1 = 1$)的所有通道停止($TE0n = 0$)。

使用 8 位存储器操作指令设置 TPS0 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 6-7. 定时器时钟选择寄存器 0 (TPS0)的格式

地址: F01B6H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TPS0	PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000

PRS 0k3	PRS 0k2	PRS 0k1	PRS 0k0		工作时钟(CK0k)的选择 [※] (k = 0, 1)				
					f _{CLK} = 1.25 MHz	f _{CLK} = 2.5 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz
0	0	0	0	f _{CLK}	1.25 MHz	2.5 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	f _{CLK} /2	625 kHz	1.25 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	f _{CLK} /2 ²	313 kHz	625 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	f _{CLK} /2 ³	156 kHz	313 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	f _{CLK} /2 ⁴	78.1 kHz	156 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	f _{CLK} /2 ⁵	39.1 kHz	78.1 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	f _{CLK} /2 ⁶	19.5 kHz	39.1 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	f _{CLK} /2 ⁷	9.77 kHz	19.5 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	f _{CLK} /2 ⁸	4.88 kHz	9.77 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	f _{CLK} /2 ⁹	2.44 kHz	4.88 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	f _{CLK} /2 ¹⁰	1.22 kHz	2.44 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	f _{CLK} /2 ¹¹	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	0	0	f _{CLK} /2 ¹²	305 Hz	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	f _{CLK} /2 ¹³	153 Hz	305 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	f _{CLK} /2 ¹⁴	76.3 Hz	153 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	f _{CLK} /2 ¹⁵	38.1 Hz	76.3 Hz	153 Hz	305 Hz	610 Hz

注 要改变 f_{CLK} 的时钟选择（通过更改系统时钟控制寄存器(CKC)的值）时，须停止定时器阵列单元 (TT0 = FFH)。

注意事项 如果 f_{CLK} (没有分频)被选择作为工作时钟 (CK0k)且将 TDR0m 设置为 0000H (m = 0, 1)，则不能检测出
来自定时器阵列单元的中断请求输出。

备注 1. f_{CLK}: CPU/外围硬件时钟频率

2. 以上寄存器的选择时钟中，其上升沿成为表示 f_{CLK} 的 1 个周期高电平的信号。详情请参阅 6.5.1 计数时钟 (f_{TCLK})。

6.3.3 定时器模式寄存器 0n (TMR0n)

TMR0n 寄存器由两个 8 位寄存器 TMR0nH 和 TMR0nL 组成，他们用于设置通道 n 的操作模式。该寄存器用于选择工作时钟(f_{MCK})，选择计数时钟，选择主/从属，选择 16 位或 8 位定时器（仅限通道 1），设置开始触发和捕捉触发，选择定时器输入的有效边沿，以及设置操作模式（间隔、捕捉、事件计数器、单计数、或者捕捉&单计数）。

当寄存器在运行（当 TE0n = 1）时，禁止改写 TMR0nH 和 TMR0nL 寄存器。

使用 8 位存储器操作指令设置 TMR0nH 和 TMR0nL 寄存器。

产生复位信号后，TMR0nH 和 TMR0nL 寄存器被清除为 00H。

注意事项 TMR0nH 寄存器的位 3 因通道而异。

TMR01H: SPLIT01 位

TMR01H: 固定为 0。

图 6-8. 定时器模式寄存器 0n (TMR0n)的格式 (1/4)

地址: F0190H (TMR00L), F0191H (TMR00H) 复位后: 00H R/W

: F0192H (TMR01L), F0193H (TMR01H)

符号	7	6	5	4	3	2	1	0
TMR01H	CKS011	0	0	CCS01	SPLIT01	STS012	STS011	STS010
符号	7	6	5	4	3	2	1	0
TMR00H	CKS001	0	0	CCS00	0	STS002	STS001	STS000
符号	7	6	5	4	3	2	1	0
TMR0nL (n = 0, 1)	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

CKS0n1	通道n工作时钟(f _{MCK})的选择
0	定时器时钟选择寄存器m (TP0m)设置的工作时钟CK00
1	定时器时钟选择寄存器0 (TPS0)设置的工作时钟CK01
工作时钟(f _{MCK})用于边沿检测电路。通过设置CCS0n位来产生计数时钟(f _{CLK})和采样时钟。	

CCS0n	通道n计数时钟(f _{CLK})的选择
0	由CKS0n0和CKS0n1位指定的工作时钟(f _{MCK})
1	Tl0n引脚的输入信号的有效边沿
计数时钟(f _{CLK})用于定时器/计数器，输出控制电路和中断控制电路。	

注意事项 1. 必须将不定位清除为 0。

2. 要改变 f_{CLK} 的时钟选择（通过更改系统时钟控制寄存器(CKC)的值）时，无论是在通过 CKS0n1 位指定的工作时钟(f_{MCK})和 Tl0n 引脚输入信号的有效边沿的两者中选择哪个作为计数时钟(f_{CLK})时，都须停止定时器阵列单元(TT0 = 0FH)。

备注 n: 通道编号 (n = 0, 1)

图 6-8. 定时器模式寄存器 0n (TMR0n)的格式 (2/4)

地址: F0190H (TMR00L), F0191H (TMR00H) 复位后: 00H R/W

: F0192H (TMR01L), F0193H (TMR01H)

符号	7	6	5	4	3	2	1	0
TMR01H	CKS011	0	0	CCS01	SPLIT01	STS012	STS011	STS010
符号	7	6	5	4	3	2	1	0
TMR00H	CKS001	0	0	CCS00	0	STS002	STS001	STS000
符号	7	6	5	4	3	2	1	0
TMR0nL (n = 0, 1)	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

(TMR01H 的位 3)

SPLIT01	用于通道1的8位定时器/16位定时器操作的选择
0	作为16位定时器工作。
1	作为8位定时器工作。

STS0n2	STS0n1	STS0n0	通道n的开始触发或者捕捉触发的设置
0	0	0	仅限软件触发开始有效（其他触发源不可选）。
0	0	1	TI0n引脚输入的有效边沿被同时用作开始触发和捕捉触发。
0	1	0	TI0n引脚的两个边沿分别被用作开始触发和捕捉触发。
1	0	0	使用主通道的中断信号（当该通道用作联动通道操作功能时的从属通道）。
其他			禁止设置

注 TMR00H 寄存器的位 3 为只读且被固定为 0，写入操作无效。

备注 n: 通道编号 (n = 0, 1)

图 6-8. 定时器模式寄存器 0n (TMR0n)的格式 (3/4)

符号

7	6	5	4	3	2	1	0
CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

TMR0nL
(n = 0, 1)

CIS 0n1	CIS 0n0	TIOn引脚输入有效边沿的选择
0	0	下降沿
0	1	上升沿
1	0	双边沿(测量低电平宽度时) 开始触发：下降沿，捕捉触发：上升沿
1	1	双边沿(测量高电平宽度时) 开始触发：上升沿，捕捉触发：下降沿
当STS0n2至STS0n0位的值不为010B且指定双边沿时，请将 CIS0n1至CIS0n0位设置为10B。		

备注 n: 通道编号 (n = 0, 1)

图 6-8. 定时器模式寄存器 0n (TMR0n)的格式 (4/4)

符号	7	6	5	4	3	2	1	0
TMR0nL (n = 0, 1)	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

MD 0n3	MD 0n2	MD 0n1	通道 n 操作模式的设置	对应功能	TCR 的计数操作
0	0	0	间隔定时器模式	间隔定时器/ 方波输出/ 分频器功能 / PWM 输出(主)	递减计数
0	1	0	捕捉模式	输入脉冲间隔测量	递增计数
0	1	1	事件计数器模式	外部事件计数器	递减计数
1	0	0	单计数模式	延迟计数器/ 单触发脉冲输出/ PWM 输出(从属)	递减计数
1	1	0	捕捉&单计数模式	输入信号的高/低电平宽度的测量	递增计数
其他			禁止设置		
MD0n0 位的操作因各模式的操作而不同（参阅下表）。					

操作模式 (由 MD0n3至MD0n1位的设置值 (参阅上表))	MD 0n0	开始计数和中断的设置
• 间隔定时器模式 (0, 0, 0)	0	开始计数时不发生定时器中断 (定时器输出也不发生变化)。
• 捕捉模式 (0, 1, 0)	1	开始计数时发生定时器中断 (定时器输出也会发生变化)。
• 事件计数器模式 (0, 1, 1)	0	开始计数时不发生定时器中断 (定时器输出也不发生变化)。
• 单计数模式 ^{注2} (1, 0, 0)	0	计数操作中的开始触发为无效。 此时不产生中断。
	1	计数操作中的开始触发为有效 ^{注3} 。 此时不产生中断。
• 捕捉&单计数模式 (1, 1, 0)	0	开始计数时不发生定时器中断 (定时器输出也不发生变化)。 计数操作中的开始触发为无效。 此时不产生中断。
其他		禁止设置

注 1. 单计数模式下，不控制开始计数操作时的中断输出(INTTM0n)和 TO0n 输出。

2. 操作过程中执行开始触发(TSMn = 1)后，计数器将被初始化，同时重新开始计数（不产生中断请求）。

备注 n: 通道编号 (n = 0, 1)

6.3.4 定时器状态寄存器 0n (TSR0n)

TSR0n 寄存器表示通道 n 的计数器的溢出状态。

TSR0n 寄存器仅在捕捉模式(MD0n3 至 MD0n1 = 010B)和捕捉&单计数模式(MD0n3 至 MD0n1 = 110B)下有效。在其他模式下不进行设置。关于 OVF 位在各种操作模式下的操作以及设置/清除条件，请参阅表 6-3。

使用 8 位存储器操作指令读取 TSR0n 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 6-9. 定时器状态寄存器 0n (TSR0n)的格式

地址: F01A0H (TSR00), F01A2H (TSR01) 复位后: 00H R

符号	7	6	5	4	3	2	1	0
TMR0n	0	0	0	0	0	0	0	OVF

OVF	通道n的计数器溢出状态
0	没有发生溢出。
1	发生溢出。
当OVF = 1，捕捉到下一个值而不产生溢出时，该标志被清除(OVF = 0)。	

表 6-3. 每种操作模式下的 OVF 位操作和设置/清除条件

定时器操作模式	OVF位	设置/清除条件
• 捕捉模式	清除	捕捉时没有发生溢出的情况下
• 捕捉&单计数模式	设置	捕捉时发生溢出的情况下
• 间隔定时器模式	清除	– (禁止使用)
• 事件计数器模式	设置	
• 单计数模式		

备注 1. 计数器溢出后 OVF 位不会立即改变，而是在随后的捕捉后改变。

2. n: 通道编号 (n = 0, 1)

6.3.5 定时器通道允许状态寄存器 0 (TE0,TEH0 (8 位模式))

TE0 和 TEH0 寄存器用于表示各通道的允许或停止定时器操作状态。

TE0 和 TEH0 寄存器的每个位对应于定时器通道开始寄存器 0 (TS0, TSH0)以及定时器通道停止寄存器 0 (TT0, TTH0)的每个位。当设置 TS0 和 TSH0 寄存器的某一位为 1 时，TE0 和 TEH0 寄存器的对应位也被设置为 1。当设置 TT0 和 TTH0 寄存器的某一位为 1 时，TE0 和 TEH0 寄存器的对应位被清除为 0。

使用 8 位存储器操作指令读取 TE0 和 TEH0 寄存器。

产生复位信号后，TE0 和 TEH0 寄存器被清除为 00H。

图 6-10. 定时器通道允许状态寄存器 0 (TE0)的格式

地址: F01B0H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
TE0	0	0	0	0	0	0	TE01	TE00

TE0n	表示通道n的操作允许或停止状态(n = 0, 1)
0	停止操作状态。
1	允许操作。
表示16位定时器的操作允许或停止状态。	
当通道1处于8位定时器模式时，TE01寄存器表示低8位定时器的操作允许或停止状态。	

图 6-11. 定时器通道允许状态寄存器 0 (TEH0)的格式

地址: F01B1H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
TEH0	0	0	0	0	0	0	TEH01	0

TEH01	表示通道1的操作允许或停止状态
0	停止操作状态。
1	允许操作。
当通道1处于8位定时器模式时，表示高8位定时器的操作允许或停止状态	

备注 n: 通道编号 (n = 0, 1)

6.3.6 定时器通道开始寄存器 0 (TS0, TSH0 (8 位模式))

TS0 和 TSH0 寄存器是触发寄存器，用于初始化定时器/计数器寄存器 0n (TCR0n)并开始各通道的计数操作。

当设置该寄存器的某一位为 1，定时器通道允许状态寄存器 0 (TE0, TEH0)的对应位也被设置为 1。因为 TSH0n 和 TS0n 位是触发位，允许操作(TE0n = 1)后立即被清除为 0。

使用 1 位或 8 位存储器操作指令设置 TS0 和 TSH0 寄存器。

产生复位信号后，TS0 和 TSH0 寄存器被清除为 00H。

图 6-12. 定时器通道开始寄存器 0 (TS0)的格式

地址: F01B2H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TS0	0	0	0	0	0	0	TS01	TS00

TS0n	允许（开始）通道n操作的触发
0	无触发操作。
1	将TE0n位设置为1并允许计数操作。 计数操作允许状态下，TCR0n寄存器的计数操作的开始因操作模式而异（参阅表6-4）。 当通道1处于8位定时器模式时，TS01为允许（开始）低8位定时器操作的触发。

图 6-13. 定时器通道开始寄存器 0 (TSH0)的格式

地址: F01B3H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TSH0	0	0	0	0	0	0	TSH01	0

TSH01	允许（开始）通道1操作的触发
0	无触发操作。
1	将TEH01位设置为1并允许计数操作。 计数操作允许状态下，在间隔定时器模式时TCR01寄存器开始计数操作(参阅表6-4)。 当通道1处于8位定时器模式时，作为允许（开始）高8位定时器操作的触发。

注意事项 1. 必须将不确定位清除为 0。

2. 从不使用 TI0n 引脚输入的功能切换到使用 TI0n 引脚输入的功能的情况下，从设置定时器模式寄存器 0n (TMR0n)后，到 TS0n 位被设置为 1 为止，需要等待以下期间。

当允许 TI0n 引脚噪声滤波器(TNFEN = 1)时：四个工作时钟(fmck)周期

当禁止 TI0n 引脚噪声滤波器(TNFEN = 0)时：两个工作时钟(fmck)周期

备注 1. TS0 和 TSH0 寄存器的读取值始终为 0。

2. n: 通道编号 (n = 0, 1)

6.3.7 定时器通道停止寄存器 0 (TT0, TTH0 (8 位模式))

TT0 和 TTH0 寄存器为用于停止各通道的计数操作的触发寄存器。

当设置 TT0 和 TTH0 寄存器的某一位为 1 时，定时器通道允许状态寄存器 0 (TE0, TEH0)的对应位被清除为 0。因为 TT0n 和 TTH0n 位是触发位，停止操作(TE0n, TTH0n = 0)后立即被清除为 0。

使用 1 位或 8 位存储器操作指令设置 TT0 和 TTH0 寄存器。

产生复位信号后，TT0 和 TTH0 寄存器被清除为 00H。

图 6-14. 定时器通道结束寄存器 0 (TT0)的格式

地址: F01B4H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TT0	0	0	0	0	0	0	TT01	TT00

TT0n	通道n的操作停止触发
0	无触发操作。
1	TE0n 被清除为 0。操作停止（产生停止触发）。 当通道1处于8位定时器模式时，TT01为停止低8位定时器操作的触发。

图 6-15. 定时器通道结束寄存器 0 (TTH0)的格式

地址: F01B5H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TTH0	0	0	0	0	0	0	TTH01	0

TTH01	通道1的操作停止触发
0	无触发操作。
1	TEH01 被清除为 0。操作停止（产生停止触发）。 当通道1处于8位定时器模式时，作为停止高8位定时器操作的触发。

注意事项 必须将不确定位清除为 0。

备注 1. TT0 和 TTH0 寄存器的读取值始终为 0。

2. n: 通道编号 (n = 0, 1)

6.3.8 定时器输出允许寄存器 0 (TOE0)

TOE0 寄存器用于允许或禁止各通道的定时器输出。

对于已经允许定时器输出的通道 n ，不能通过软件改写下文所述的定时器输出寄存器 0(TO0)的 $TO0n$ 位的值，同时，通过计数操作从定时器输出引脚($TO0n$)输出定时器输出功能反映的值。

使用 8 位存储器操作指令设置 TOE0 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 6-16. 定时器输出允许寄存器 0 (TOE0)的格式

地址: F01BAH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TOE0n	0	0	0	0	0	0	TOE01	TOE00

TOE0n	允许/禁止通道n的定时器输出
0	禁止定时器的输出。 定时器操作没有反映到TO0n位，固定输出。 允许写入TO0n位。
1	允许定时器的输出。 定时器操作反映到TO0n位，产生输出波形。 禁止TO0n位的写入（写入被忽略）。

注意事项 必须将不确定位清除为 0。

备注 n : 通道编号 ($n = 0, 1$)

6.3.9 定时器输出寄存器 0 (TO0)

TO0 寄存器是各通道的定时器输出的缓冲器寄存器。

该寄存器的各个位的值从各通道的定时器输出引脚(TO0n)输出。

仅限禁止定时器输出(TOE0n = 0)时可以用软件改写该寄存器的 TO0n 位。当允许定时器输出(TOE0n = 1)时，用软件改写该寄存器无效，且仅能通过定时器操作改变其值。

要将 TO0n 复用引脚用作端口功能引脚，须将相应的 TO0n 位设置为 0。

使用 8 位存储器操作指令设置 TO0 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 6-17. 定时器输出寄存器 0 (TO0)的格式

地址: F01B8H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	TO01	TO00

TO0n	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 必须将不确定位清除为 0。

备注 n: 通道编号 (n = 0, 1)

6.3.10 定时器输出电平寄存器 0 (TOL0)

TOL0 寄存器是用于控制各通道定时器输出电平的寄存器。

当允许定时器输出($TOE0n = 1$)且为多通道联动操作功能($TOM0n = 1$)时, 在设置或复位定时器输出信号的时序, 反映通过该寄存器的对于通道 n 的反相设置。主通道输出模式($TOM0n = 0$)时, 对该寄存器的设置无效。

使用 8 位存储器操作指令设置 TOL0 寄存器。

产生复位信号后, 该寄存器被清除为 00H。

图 6-18. 定时器输出电平寄存器 0 (TOL0)的格式

地址: F01BCH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	TOL01	0

TOL 0n	通道n定时器输出电平的控制
0	正逻辑输出(高电平有效)
1	负逻辑输出(低电平有效)

注意事项 必须将不确定位清除为 0。

备注 1. 定时器输出逻辑反相发生在下一个定时器输出信号改变时, 而不是在改写之后立即反相。

2. n : 通道编号 ($n = 0, 1$)

6.3.11 定时器输出模式寄存器 0 (TOM0)

TOM0 寄存器用于控制各通道的定时器输出模式。

某个通道作为单通道操作功能使用时，将使用通道的对应位设置为 0。

某个通道作为多通道联动操作功能（PWM 输出或单触发脉冲输出）使用时，将主通道的对应位设置为 0，将从属通道的对应位设置为 1。

当允许定时器输出($TOE0n = 1:n = 0, 1$)时，该寄存器对各通道 n 的设定，可以通过定时器输出信号置位复位的时序来反映。

使用 8 位存储器操作指令设置 TOM0 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 6-19. 定时器输出模式寄存器 0 (TOM0)的格式

地址: F01BEH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	TOM01	0

TOM0n	通道n定时器输出模式的控制
0	用作单通道操作功能
1	从属通道输出模式 (输出由主通道的定时器中断请求信号(INTTM00)置位，由从属通道的定时器中断请求信号(INTTM01)复位)

注意事项 必须将不定位清除为 0。

备注 n: 通道编号 ($n = 0, 1$)

6.3.12 噪声滤波器允许寄存器 1 (NFEN1)

NFEN1 寄存器用于设置是否将噪声滤波器用于输入各通道的定时器输入信号。

将需要消除噪声的引脚的对应位设置为 1，启用噪声滤波器。

当噪声滤波器启用时，对 CPU/外围硬件时钟(f_{MCK})时钟执行 4 个时钟的一致检测和同步。噪声滤波器关闭时，只对 CPU/外围硬件时钟(f_{CLK})执行 2 个时钟的一致检测和同步。详情请参阅 6.5.1 (2) 选择 TI0n 引脚输入信号的有效边沿时 (CCS0n = 1) 和 6.5.2 计数器的开始时序。

使用 1 位或 8 位存储器操作指令设置 NFEN1 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 6-20. 噪声滤波器允许寄存器 1 (NFEN1)的格式

地址: F0071H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
NFEN1	0	0	0	0	0	0	TNFEN01	TNFEN00
TNFEN0n		TI0n引脚输入信号的噪声滤波器的允许/禁止使用						
0		噪声滤波器关闭						
1		噪声滤波器启用						

备注 n: 通道编号(n = 0, 1)

6.3.13 端口模式寄存器 0 (PM0)

该寄存器以 1 位为单位设置端口 1 的输入/输出。

将用于定时器输出引脚的复用端口(例如 P04/ANI3/TI01/TO01/KR5)用作定时器输出时, 须将端口模式寄存器 0 (PM0)、端口寄存器 0 (P0)和端口模式控制寄存器 0 (PMC0)中对应于各端口的位设置为 0。

示例: 将 P04/ANI3/TI01/TO01/KR5 用作定时器输出时

将端口模式控制寄存器 0 的 PMC04 位设置为 0。

将端口模式寄存器 0 的 PM4 位设置为 0。

将端口寄存器 0 的 P04 位设置为 0。

将用于定时器输入引脚的复用端口(例如 P04/ANI3/TI01/TO01/KR5)用作定时器输入时, 须将端口模式寄存器 4 (PM4)中对应于各端口的位设置为 1。并且, 须将端口模式控制寄存器 4 (PMC4)中对应于各端口的位设置为 0。此时, 端口寄存器 4 (P4)的位可以设置为 0 或 1。

使用 1 位或 8 位存储器操作指令设置 PM0 寄存器。

产生复位信号后, 该寄存器被设置为 FFH。

备注 TI01、TO00 和 TO01 引脚复用为模拟输入引脚。使用定时器输入/输出功能时, 必须将用于切换数字输入/输出或模拟输入的 PMCx 寄存器的对应位设置为 0。

图 6-21. 端口模式寄存器 0 (PM0)的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W

PM0m	选择 P0m 引脚的输入/输出模式 (m = 0 至 4)
0	输出模式 (输出缓冲器启用)
1	输入模式 (输出缓冲器关闭)

注意事项 通过外围输入/输出重定向寄存器(PIOR)改变了外围输入输出组件时, 也使用 PM4。

备注 n: 通道编号(n = 0, 1)

6.4 定时器阵列单元的基本规则

6.4.1 多通道联动操作功能的基本规则

使用多通道联动操作功能，即联合使用主通道（主要用于对周期进行计数的基准定时器）和从属通道（以主通道为基准而工作的定时器）时，须遵守以下规则。

- (1) 只能将偶数通道（通道 0）设置为主通道。
- (2) 通道 1 可以设置为从属通道。
- (3) 与某个主通道联动的从属通道的工作时钟必须与该主通道时钟相同。与主通道联动操作的从属通道的 CKS011 位（定时器模式寄存器 01H (TMR01H) 的位 7）的值必须与主通道的相同。
- (4) 主通道可以向更高编号的通道发送 INTTM00（中断）、开始软件触发和计数时钟。
- (5) 从属通道不能将主通道的 INTTM00（中断）、开始软件触发或计数时钟用作源时钟。
- (6) 要同时启用联动操作的通道，因此必须同时设置联动通道的通道开始触发位(TS0n)。
- (7) 在计数操作时，可以仅设置主通道的 TS0n 位或者联动操作的所有通道的 TS0n 位。不能仅设置从属通道的 TS0n 位。
- (8) 要同时停止联动操作的通道，必须同时设置联动通道的通道停止触发位(TT0n)。
- (9) 定时器模式寄存器 0n (TMR0nH) 中没有主从设置位（固定为 0）。但是，由于通道 0 是最低通道，因此联动操作时可以将通道 0 用作主通道。

关于(2)，通道 1 的低 8 位可以设置为从属通道。此时，通道 1 的高 8 位可以用作间隔定时器。

备注 n: 通道编号 (n = 0, 1)

6.4.2 8 位定时器操作功能的基本规则（仅限通道 1）

通过 8 位定时器操作功能，可以将 16 位定时器通道作为两个 8 位定时器通道来使用。

该功能仅用于通道 1，并须遵守一些规则。

该功能的基本规则如下所示：

- (1) 8 位定时器操作功能仅适用于通道 1。
- (2) 作为 8 位定时器使用时，将定时器模式寄存器 01 (TMR01H) 的 SPLIT 位设置为 1。
- (3) 高 8 位定时器可以用作间隔定时器功能。
- (4) 开始操作时，高 8 位输出 INTTM01H（中断）（执行与设置 MD010 为 1 时相同的操作）。
- (5) 高 8 位定时器的工作时钟的选择须根据低位 TMR01H 寄存器的 CKS011 位进行操作。
- (6) 对于高 8 位定时器，通过操作 TSH01 位来开始通道操作，通过操作 TTH01 位来停止通道操作。可以使用 TEH01 位来确认通道状态。
- (7) 低 8 位定时器依据 TMR01H 和 TMR01L 寄存器的设置工作。以下四种功能支持低 8 位定时器操作：
 - 间隔定时器功能
 - 外部事件计数器功能
 - 延迟计数功能
 - PWM 输出
- (8) 对于低 8 位，通过操作 TS01 位来开始通道操作，通过操作 TT01 位来停止通道操作。可以使用 TE01 位来确认通道状态。
- (9) 在 16 位操作中，对 TSH01/TTH01 位进行的操作无效。通过操作 TS01 和 TT01 位来操作通道 n。TEH01 位不改变。
- (10) 在使用 8 位定时器功能时，不能使用联动操作功能（单触发脉冲）。

备注 n: 通道编号 (n = 0, 1)

注意事项 在 8 位定时器模式(SPLIT = 1)下使用通道 1 时，禁止读取 TCR01H 和 TDR01H 寄存器。

6.5 计数器的操作

6.5.1 计数时钟(f_{TCLK})

定时器阵列单元的计数时钟(f_{TCLK})可以通过定时器模式寄存器 0n (TMR0n)的 CCS0n 位从以下两者之中选择。

- 由 CKS0n1 位指定的工作时钟(f_{MCK})
- TI0n 引脚的输入信号的有效边沿

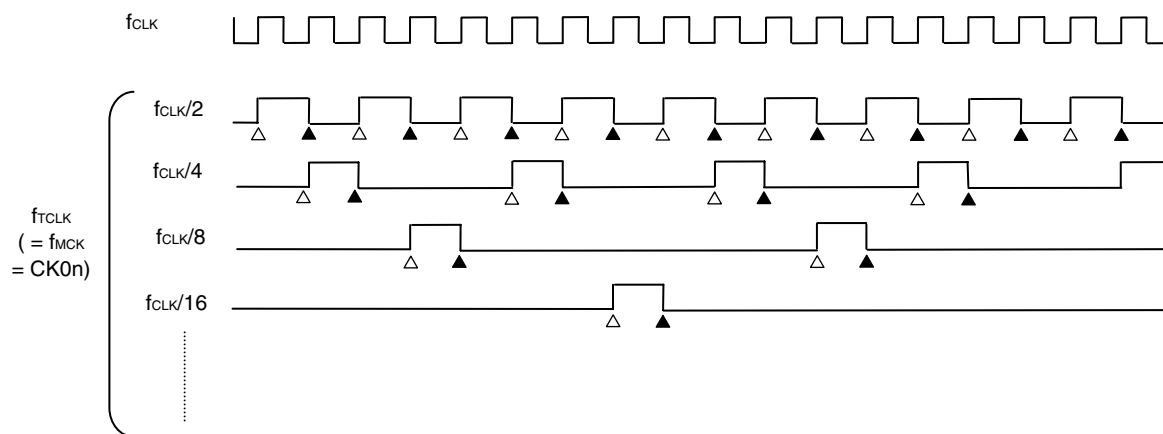
由于定时器阵列单元被设计为与 f_{CLK} 同步操作，计数时钟(f_{TCLK})的时序如下所示。

(1) 选择由 CKS0n1 位指定的工作时钟(f_{MCK})时 (CCS0n = 0)

根据定时器时钟选择寄存器 0 (TPS0)的设置，计数时钟(f_{TCLK}) 在 f_{CLK} 与 $f_{CLK}/2^{15}$ 之间进行选择。但是，当选择为 f_{CLK} 的分频时，由 TPS0 寄存器选择的时钟在上升沿后保持 1 个 f_{CLK} 周期的高电平。选择为 f_{CLK} 时，固定在高电平。

为了与 f_{CLK} 保持同步，定时器/计数器寄存器 0n (TCR0n)是在计数时钟的上升沿等待 1 个 f_{CLK} 时钟周期后开始计数。出于方便，将其称为“在计数时钟的上升沿计数”。

图 6-22. f_{CLK} 和计数时钟(f_{TCLK})的时序(CCS0n = 0 时)



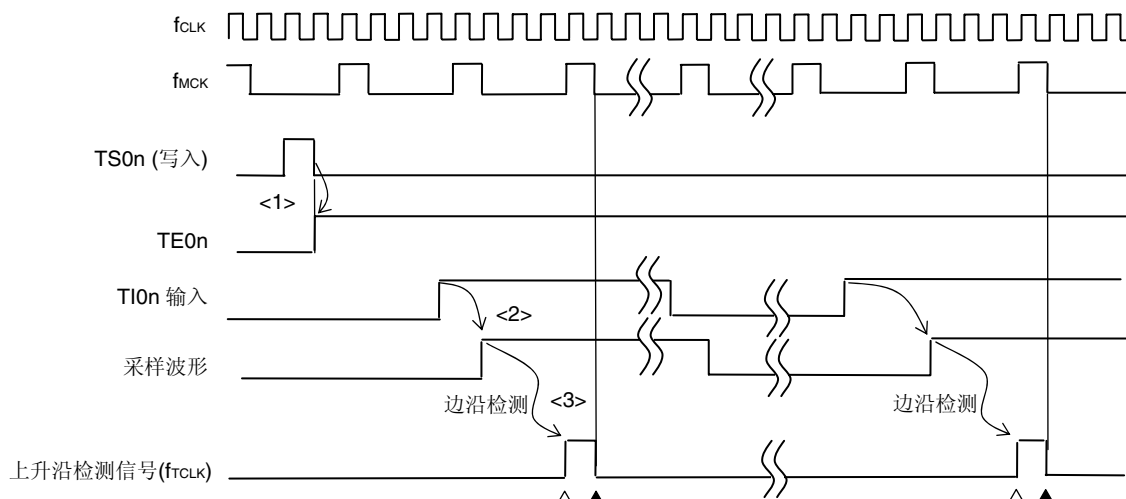
- 备注 1. Δ : 计数时钟的上升沿
 ▲ : 同步、计数器的递增/递减
2. f_{CLK} : CPU/外围硬件时钟
3. n: 通道编号 (n = 0, 1)

(2) 选择 TI0n 引脚输入信号的有效边沿时(CCS0n = 1)

计数时钟(f_{TCLK})为检测 TI0n 引脚输入信号的有效边沿并同步至下一个 f_{MCK} 上升沿的信号。计数时钟(f_{TCLK})比实际上的 TI0n 引脚的输入信号延迟了 1 至 2 个 f_{MCK} 时钟周期(使用噪声滤波器时,延迟 3 至 4 个时钟周期)。

为了与 f_{CLK} 保持同步,定时器/计数器寄存器 0n (TCR0n)是在计数时钟的上升沿等待 1 个 f_{CLK} 时钟周期后开始计数。出于方便,将其称为“在 TI0 引脚输入信号的有效边沿计数”。

图 6-23. f_{CLK} 和计数时钟(f_{TCLK})的时序(CCS0n = 1, 未使用噪声滤波器时)



<1> 设置 TS0n 位为 1 以启动定时器,并成为等待 TI0n 引脚输入信号的有效边沿的状态。

<2> f_{MCK} 对 TI0n 引脚输入信号的上升沿进行采样。

<3> 在采样信号的上升沿检测边沿,并输出检测信号(计数时钟)。

备注 1. Δ : 计数时钟的上升沿

$\blacktriangle$: 同步、计数器的递增/递减

2. f_{CLK} : CPU/外围硬件时钟

f_{MCK} : 通道 n 的工作时钟

3. 输入脉冲间隔测量、输入信号高/低电平的测量、延迟计数器以及单触发脉冲输出功能的 TI0n 引脚输入信号具有相同的波形,如图 6-23 所示。

4. n: 通道编号 (n = 0, 1)

6.5.2 计数器的开始时序

通过设置定时器通道开始寄存器 0 (TS0)的 TS0n 位, 可以允许操作定时器/计数器寄存器 0n(TCR0n)。

从计数操作允许状态到定时器/计数器寄存器 0n (TCR0n)计数开始为止的操作如表 6-4 所示。

表 6-4. 从计数操作允许状态到定时器计数寄存器 0n (TCR0n)计数开始为止的操作

定时器操作模式	设置为TS0n = 1时的操作
• 间隔定时器模式	从检测到开始触发(TS0n = 1)直到产生计数时钟为止不执行任何操作。 在第一个计数时钟将TDR0n寄存器的值载入TCR0n寄存器, 在之后的计数时钟执行递减计数操作 (参阅 6.5.3 (1) 间隔定时器模式下的操作)。
• 事件计数器模式	向TS0n位写入1, 把TDR0n寄存器的值载入TCR0n寄存器。 检测出TI0n的输入沿, 后续计数时钟执行递减计数操作。(参阅 6.5.3 (2) 事件计数器模式下的操作)。
• 捕捉模式	从检测到开始触发(TS0n = 1)直到产生计数时钟为止不执行任何操作。 在第一个计数时钟将0000H载入TCR0n寄存器, 在后续计数时钟执行递增计数操作(参阅 6.5.3 (3) 捕捉模式下的操作 (输入脉冲的间隔测量))。
• 单计数模式	在定时器停止(TE0n = 0)状态下, 向TS0n位写入1以进入等待开始触发状态。 从检测到开始触发直到产生计数时钟为止不执行任何操作。 在第一个计数时钟将TDRmn寄存器的值载入TCRmn寄存器, 在之后的计数时钟执行递减计数操作(参阅 6.5.3 (4) 单计数模式下的操作)。
• 捕捉&单计数模式	在定时器停止(TE0n = 0)状态下, 向TS0n位写入1以进入等待开始触发状态。 从检测到开始触发直到产生计数时钟为止不执行任何操作。 在第一个计数时钟将0000H载入TCR0n寄存器, 在后续计数时钟执行递增计数操作(参阅 6.5.3 (5) 捕捉&单计数模式下的操作 (测量高电平宽度))。

备注 n: 通道编号 (n = 0, 1)

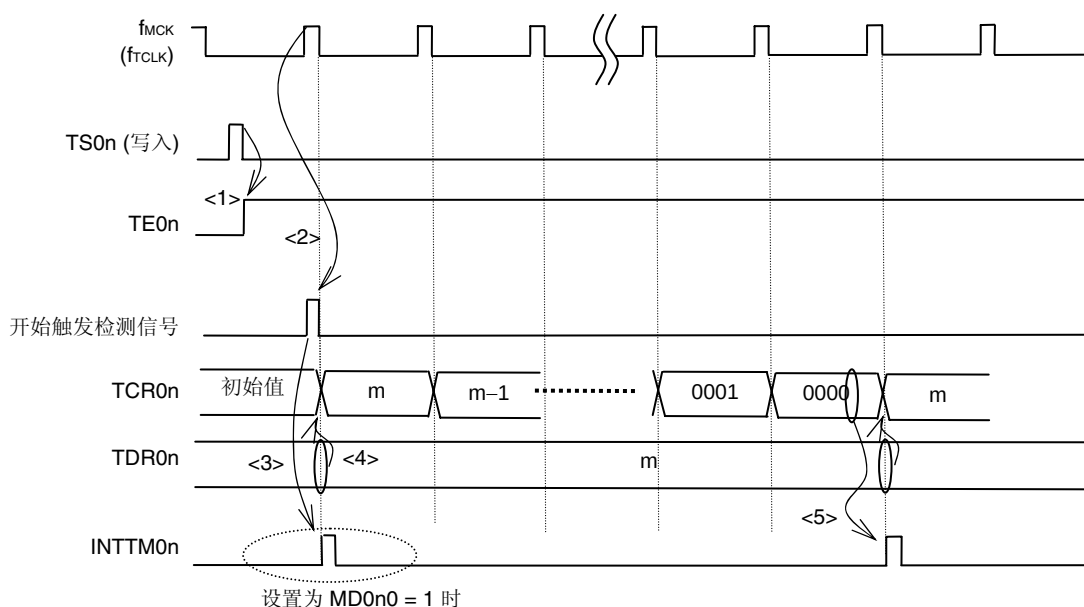
6.5.3 计数器的操作

下面对各模式中的计数器的操作进行说明。

(1) 间隔定时器模式下的操作

- <1> 通过向 $TS0n$ 位写入 1 允许定时器操作($TE0n = 1$)。直到产生计数时钟为止定时器/计数器寄存器 $0n$ ($TCR0n$)将保持初始值。
- <2> 允许操作时, 在第一个计数时钟产生开始触发信号。
- <3> 设置 $MD0n0$ 位为 1 时, 开始触发信号将产生 $INTTM0n$ 。
- <4> 在允许操作后的第一个计数时钟, 定时器数据寄存器 $0n$ ($TDR0n$)的值被载入 $TCR0n$ 寄存器, 并在间隔定时器模式下开始计数。
- <5> 当 $TCR0n$ 寄存器递减计数至其计数值为 0000H 时, 在下一个计数时钟处产生 $INTTM0n$, 并将定时器数据寄存器 $0n$ ($TDR0n$)的值载入 $TCR0n$ 寄存器, 继续计数。

图 6-24. 操作时序(间隔定时器模式下)



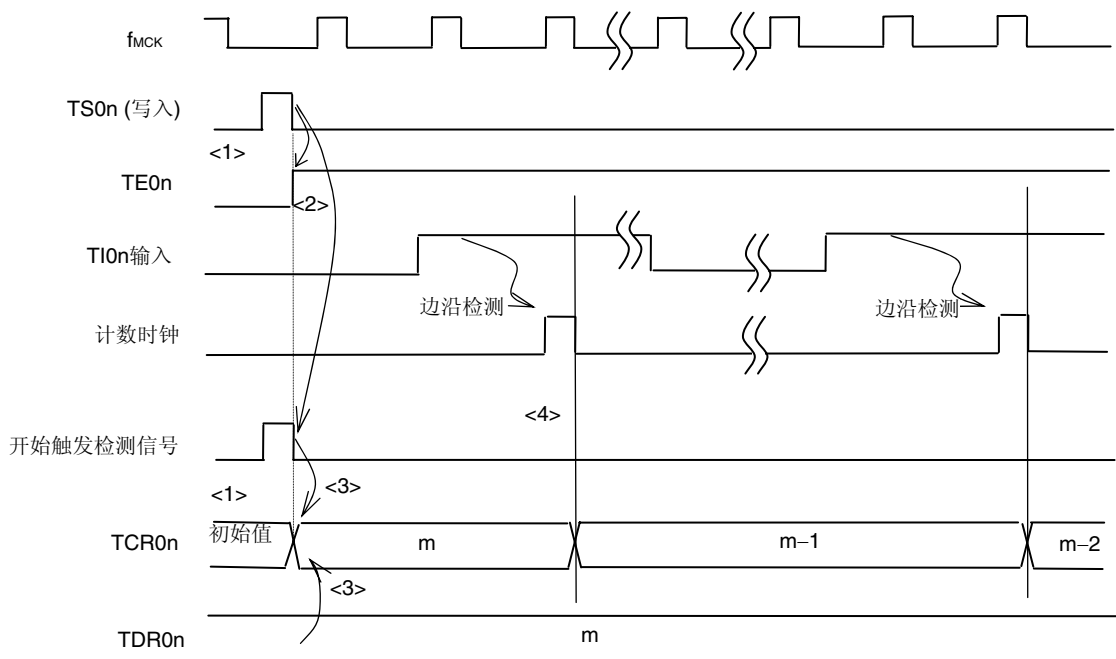
注意事项 在计数时钟的第一个周期操作中, 写 $TS0n$ 位之后到产生计数时钟为止会发生开始计数的延迟, 将产生最大为一个时钟周期的误差。需要获得开始计数时序信息时, 可以通过设置 $MD0n0 = 1$ 在开始计数时产生中断。

- 备注 1.** f_{MCK} 、开始触发检测信号和 $INTTM0n$ 与 f_{CLK} 同步后, 在一个时钟周期之内被激活。
- 2.** n : 通道编号 ($n = 0, 1$)

(2) 事件计数器模式下的操作

- <1> 停止操作($TE0n = 0$)时, 定时器/计数器寄存器 $0n$ (TCR $0n$)保持初始值。
- <2> 通过向 TS $0n$ 位写入 1 允许定时器操作($TE0n = 1$)。
- <3> 向 TS $0n$ 位写入 1 并且 TE $0n$ 位被设置为 1 后, 立即将定时器数据寄存器 $0n$ (TDR $0n$)的值载入到 TCR $0n$ 寄存器并开始计数。
- <4> 此后, 依据 TI $0n$ 输入的有效边沿的计数时钟对 TCR $0n$ 寄存器值进行递减计数。

图 6-25. 操作时序(事件计数器模式下)



备注 1. 不使用噪声滤波器时的时序如图 6-25 所示。噪声滤波器启用时, 边沿检测将比 TI $0n$ 输入再晚 2 个 fMCK 周期 (合计 3 至 4 个周期)。

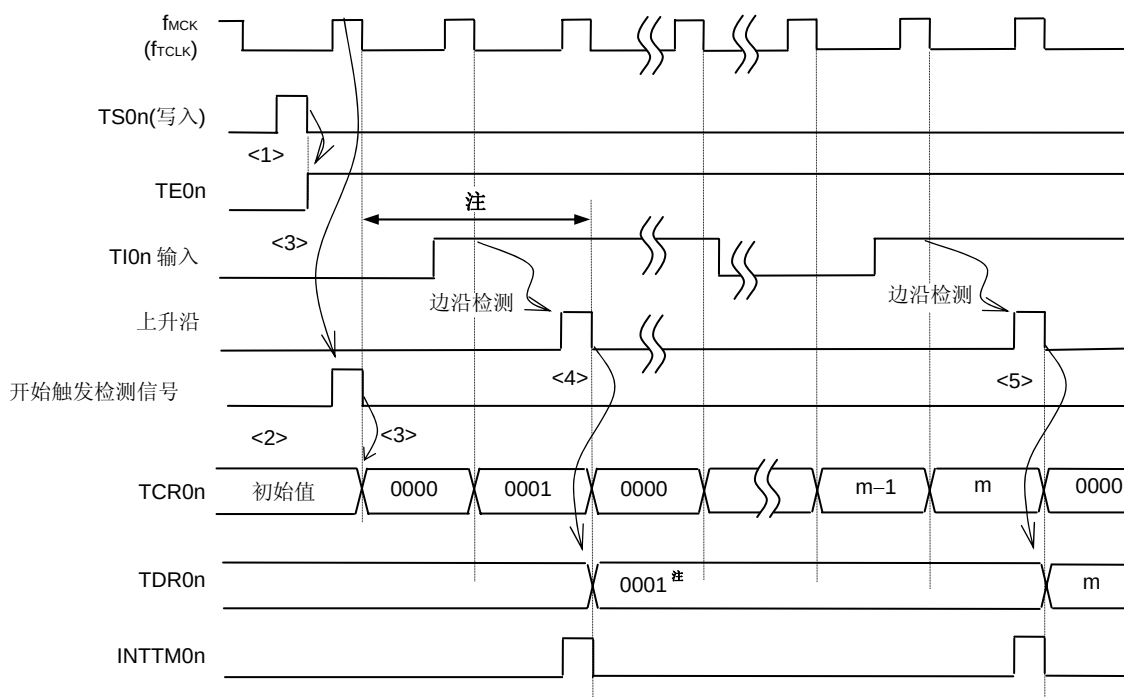
1 个周期的误差来源于 TI $0n$ 输入与计数时钟(fMCK)的周期之间的不同步。

2. n: 通道编号 (n = 0, 1)

(3) 捕捉模式下的操作 (输入脉冲的间隔测量)

- <1> 通过向 TS0n 位写入 1 允许定时器操作($TE0n = 1$)。
- <2> 直到产生计数时钟为止定时器/计数器寄存器 0n (TCR0n)将保持初始值。
- <3> 允许操作时, 在第一个计数时钟产生开始触发信号。将 0000H 载入 TCR0n 寄存器并以捕捉模式开始计数。(设置 MD0n0 位为 1 时, 开始触发信号将产生 INTTM0n。)
- <4> 检测到 TI0n 输入的有效边沿时, 将 TCR0n 寄存器的值捕捉至定时器数据寄存器 0n(TDR0n), 并产生 INTTM0n。此时捕捉值没有意义。TCR0n 寄存器从 0000H 开始继续计数。
- <5> 检测到下一个 TI0n 输入的有效边沿时, 将 TCR0n 寄存器的值捕捉至定时器数据寄存器 0n(TDR0n), 并产生 INTTM0n。

图 6-26. 操作时序(捕捉模式下: 输入脉冲的间隔测量)



注意事项 在计数时钟的第一个周期操作中, 写 TS0n 位之后到产生计数时钟为止会发生开始计数的延迟, 将产生最大为一个时钟周期的误差。需要获得开始计数时序信息时, 可以通过设置 MD0n0 = 1 在开始计数时产生中断。

注 当捕捉开始时, 如果时钟已经被输入到 TI0n (触发存在), 那么即便没有检测出边沿, 只要检测到触发就会开始计数。因此, 第一个捕捉值(<4>) 并不能代表脉冲间隔(上图中 0001 仅为 2 个时钟周期, 并不能代表脉冲间隔), 用户可忽略。

备注 1. 不使用噪声滤波器时的时序如图 6-26 所示。噪声滤波器启用时, 边沿检测将比 TI0n 输入再晚 2 个 f_{MCK} 周期 (合计 3 至 4 个周期)。

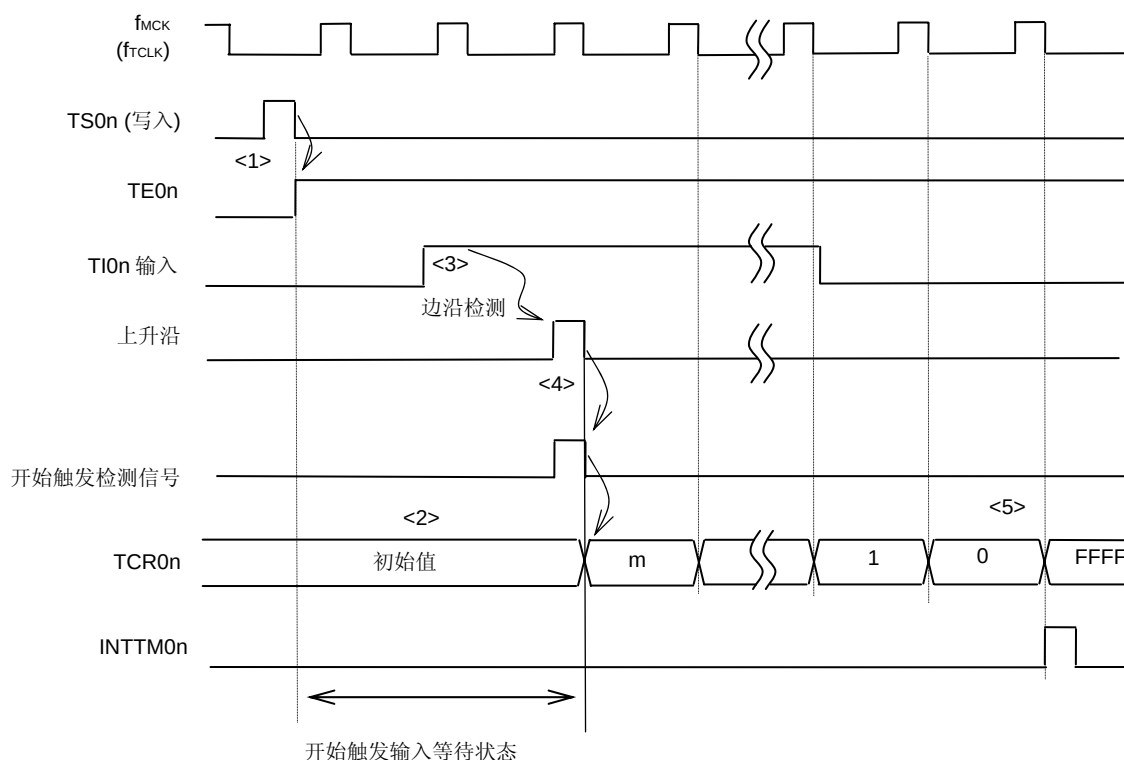
1 个周期的误差来源于 TI0n 输入与计数时钟(f_{MCK})的周期之间的不同步。

2. n: 通道编号 (n = 0, 1)

(4) 单计数模式下的操作

- <1> 通过向 TS0n 位写入 1 允许定时器操作($TE0n = 1$)。
- <2> 定时器/计数器寄存器 0n (TCR0n)保持初始值直到产生开始触发信号为止。
- <3> 检测到 TI0n 输入的上升沿。
- <4> 检测到开始触发信号时, 将定时器数据寄存器 0n (TDR0n)的值载入 TCR0n 寄存器并开始计数。
- <5> 当 TCR0n 寄存器递减计数至其计数值为 0000H 时, 产生 INTTM0n, TCR0n 寄存器的值变成 FFFFH 并停止计数。

图 6-27. 操作时序(单计数模式下)



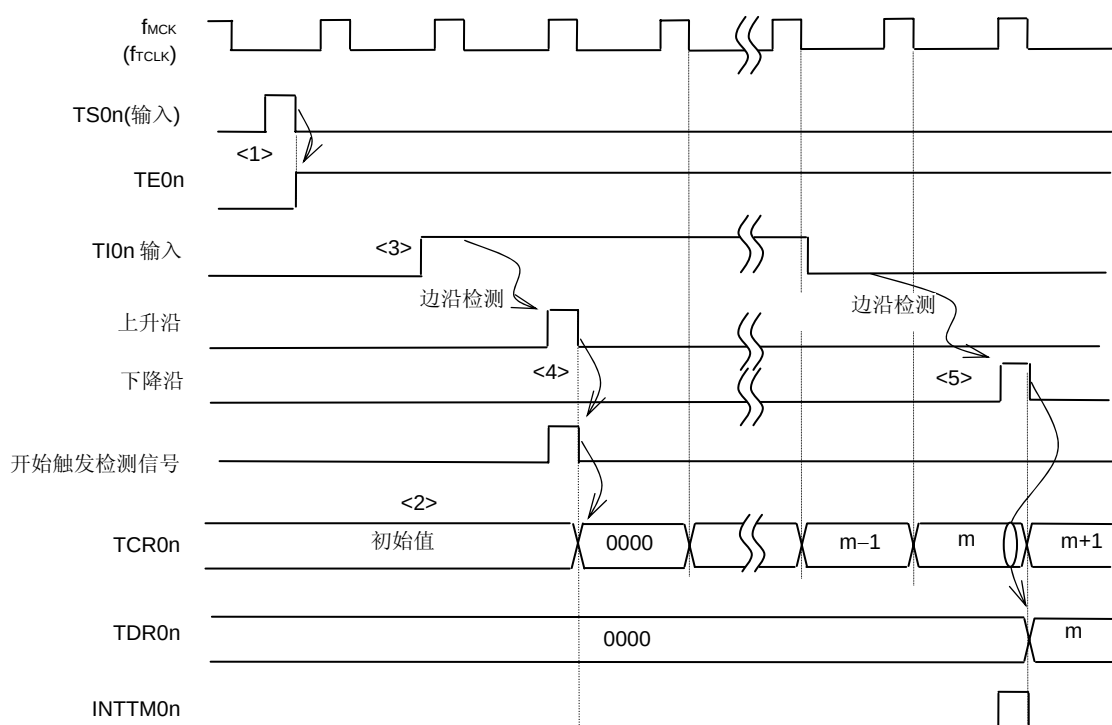
备注 1. 不使用噪声滤波器时的时序如图 6-27 所示。噪声滤波器启用时, 边沿检测将比 TI0n 输入再晚 2 个 f_{MCK} 周期 (合计 3 至 4 个周期)。1 个周期的误差来源于 TI0n 输入与计数时钟(f_{MCK})的周期之间的不同步。

2. n: 通道编号 (n = 0, 1)

(5) 捕捉&单计数模式下的操作（测量高电平宽度）

- <1> 通过向定时器通道开始寄存器 0(TS0)的 TS0n 位写入 1 允许定时器操作(TE0n = 1)。
- <2> 定时器/计数器寄存器 0n (TCR0n)保持初始值直到产生开始触发信号为止。
- <3> 检测到 TI0n 输入的上升沿。
- <4> 检测到开始触发信号时，将 0000H 值载入 TCR0n 寄存器并开始计数。
- <5> 检测到 TI0n 输入的下降沿时，将 TCR0n 寄存器的值捕捉至定时器数据寄存器 0n (TDR0n)，并产生 INTTM0n。

图 6-28. 操作时序(捕捉&单计数模式下：高电平宽度的测量)



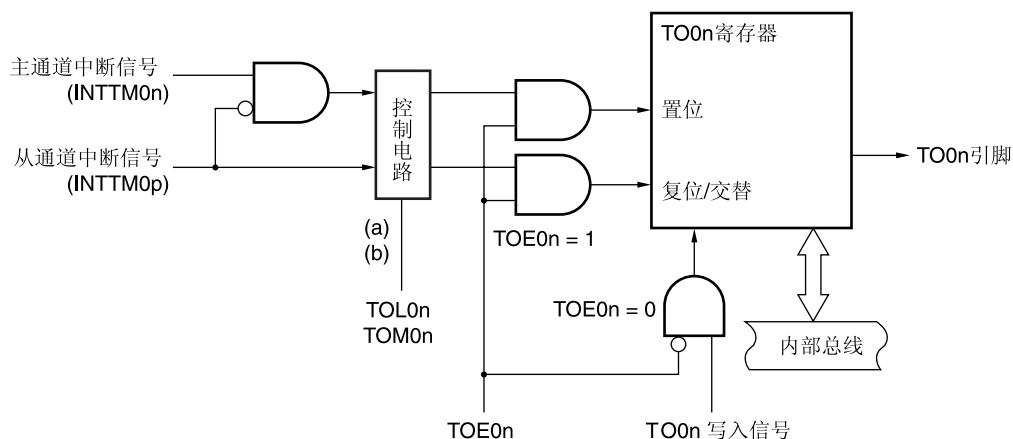
备注 1. 不使用噪声滤波器时的时序如图 6-28 所示。噪声滤波器启用时，边沿检测将比 TI0n 输入再晚 2 个 fMCK 周期（合计 3 至 4 个周期）。1 个周期的误差来源于 TI0n 输入与计数时钟(fMCK)的周期之间的不同步。

2. n: 通道编号 (n = 0, 1)

6.6 通道输出 (TO0n 引脚) 控制

6.6.1 TO0n 引脚输出电路配置

图 6-29. 输出电路配置



下面介绍 TO0n 引脚输出电路。

<1> 当允许定时器输出($TOE0n = 1$)时, 将 $INTTM0n$ (主通道定时器中断) 和 $INTTM0p$ (从属通道定时器中断) 的双方都发送至 $TO0$ 寄存器。向 $TO0$ 寄存器的写入 ($TO0n$ 写信号) 无效。

当 $TOE0n = 1$ 时，中断信号以外的信号不会改变 $TO0n$ 引脚输出。

初始化 TO0n 引脚输出电平时，需要设置为禁止定时器操作($TOE0n = 0$)并向 TO0 寄存器写入值。

- (a) 当 $TOM0n = 0$ (主通道输出模式) 时, 定时器输出电平寄存器 $0(TOL0)$ 的设定值被忽略, 仅将 $INTTM0p$ (从属通道定时器中断) 发送至定时器输出寄存器 $0(TO0)$ 。
- (b) 当 $TOM0n = 1$ (从属通道输出模式) 时, 将 $INTTM0n$ (主通道定时器中断) 和 $INTTM0p$ (从属通道定时器中断) 的双方都发送至 $TO0$ 寄存器。

此时，TOL0 寄存器变为有效，并按以下方式控制信号：

TOL0n = 0 时: 正逻辑输出 (INTTM0n → 设置, INTTM0p → 复位)

TOL0n = 1 时: 负逻辑输出 (INTTM0n → 复位, INTTM0p → 设置)

当同时产生 INTTM0n 和 INTTM0p (PWM 输出的 0%被输出) 时, INTTM0p (复位信号) 优先, INTTM0n (设置信号) 被屏蔽。

<2> 当禁止定时器输出($TOE0n = 0$)时, 对目标通道的 $TO0n$ 位的写入操作 ($TO0n$ 写信号) 有效。当禁止定时器输出($TOE0n = 0$)时, 不将 $INTTM0n$ (主通道定时器中断) 和 $INTTM0p$ (从属通道定时器中断) 的任意一方发送至 $TO0$ 寄存器。

<3> 任何时候均可读取 TO0 寄存器，确认 TO0n 引脚输出电平。

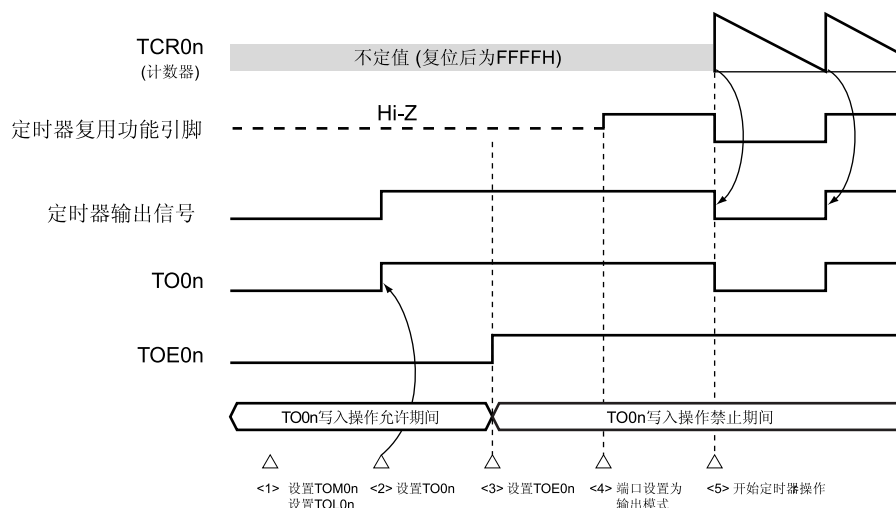
备注 n: 通道编号 $n = 0, 1$ (主通道时 $n = 0$)

p: 从属通道编号 p = 1

6.6.2 TO0n 引脚输出设置

从 TO0n 输出引脚的初始设置到开始定时器操作为止的步骤和状态变化如下图所示。

图 6-30. 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。

- TOM0n 位 (0: 主通道输出模式, 1: 从属通道输出模式)
- TOL0n 位 (0: 正逻辑输出, 1: 负逻辑输出)

<2> 通过设置定时器输出寄存器 0 (TO0)将定时器输出信号设置为初始状态。

<3> 向 TOE0n 位写入 1 以允许定时器输出操作 (禁止写入 TO0 寄存器)。

<4> 通过端口模式控制寄存器(PMCxx)将该端口设置为数字输入/输出 (参阅 6.3.13 端口模式寄存器 0 (PM0))。

<5> 将端口输入/输出设置设为输出 (参阅 6.3.13 端口模式寄存器 0 (PM0))。

<6> 允许定时器操作(TS0n = 1)。

备注 n: 通道编号 (n = 0, 1)

6.6.3 通道输出操作的注意事项

(1) 在定时器操作时 TO0、TOE0 和 TOL0 寄存器的设置值的变化

由于定时器操作（定时器计数寄存器 0n(TCR0n)和定时器数据寄存器 0n(TDR0n)的操作）独立于 TO0n 输出电路，更改在定时器输出寄存器 0(TO0)、定时器输出允许寄存器 0(TOE0)、定时器输出电平寄存器 0(TOL0)的设置值不会影响定时器操作，在定时器操作期间可以更改这些值。但是，各个定时器操作时若要从 TO0n 引脚输出预期波形的话，须将 TO0、TOE0、TOL0 和 TOM0 寄存器设置为各个操作时寄存器设置内容示例的值。

如果在接近各通道定时器中断(INTTM0n)发生时更改 TOE0 和 TOL0 寄存器（TO0 寄存器除外）的设置值，根据更改的执行是在定时器中断(INTTM0n)发生前还是发生后，输出至 TO0n 引脚的波形可能不同。

备注 n: 通道编号 (n = 0, 1)

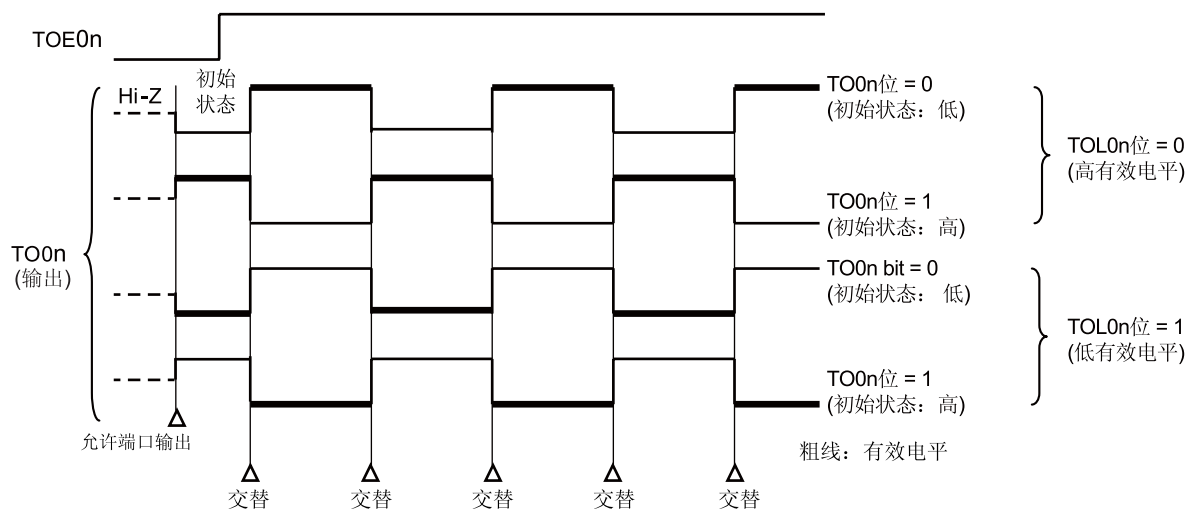
(2) TO0n 引脚的初始电平和开始定时器操作后的输出电平

在允许端口输出之前，禁止定时器输出(TOE0n = 0)的状态下，写入定时器输出寄存器 0(TO0)更改初始电平后，设置为允许定时器输出(TOE0n = 1)时，TO0n 引脚输出电平的变化如下所示。

(a) 在主通道输出模式(TOM0n = 0)设置下开始操作时

主通道输出模式(TOM0n = 0)时，定时器输出电平寄存器 0(TOL0)的设置无效。设置初始电平后开始定时器操作时，产生交替信号使 TO0n 引脚的输出电平反相。

图 6-31. 交替输出(TOM0n = 0)时的 TO0n 引脚输出状态

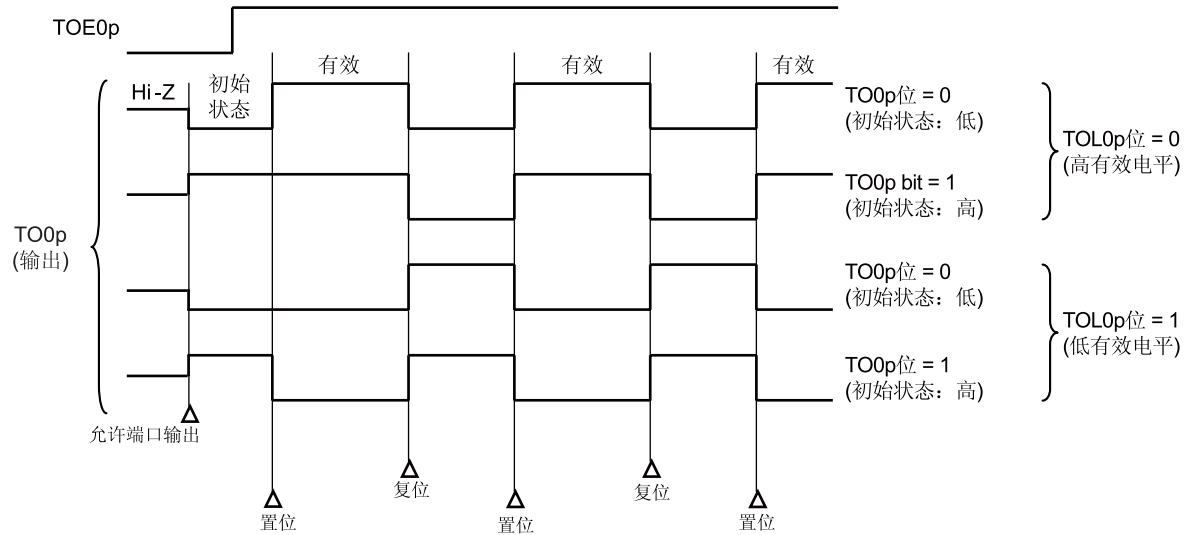


备注 1. 交替: TO0n 引脚输出状态反相

2. n: 通道编号 (n = 0, 1)

(b) 从属通道输出模式($TOM0p = 1$)设置下开始操作时 (PWM 输出)

从属通道输出模式($TOM0p = 1$)时, 有效电平取决于定时器输出电平寄存器 0($TOL0p$)的设置。

图 6-32. PWM 输出时 ($TOM0p = 1$)的 $TO0p$ 引脚输出状态

备注 1. 置位: $TO0p$ 引脚的输出信号从无效电平变为有效电平。

复位: $TO0p$ 引脚的输出信号从有效电平变为无效电平。

2. p : 通道编号 ($p = 1$)

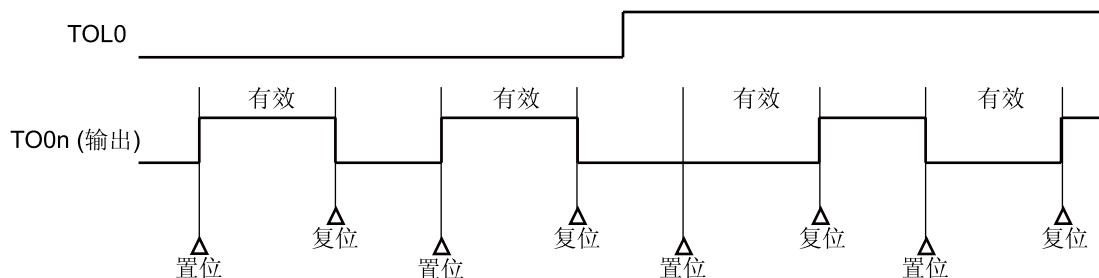
(3) 在从属通道输出模式($TOM0n = 1$)下 $TO0n$ 引脚的操作

(a) 在定时器操作期间改变定时器输出电平寄存器 0 (TOL0) 设置时

定时器操作期间更改 TOL0 寄存器的设置时，在 $TO0n$ 引脚变更条件的产生时序设置生效。改写 TOL0 寄存器不会立刻改变 $TO0n$ 引脚的输出电平。

当 $TOM0n$ 被设置为 1，在操作定时器($TE0n = 1$)期间更改 TOL0 寄存器的值时的操作如下所示。

图 6-33. 在定时器操作期间改变 TOL0 寄存器内容时的操作



备注 1. 置位： $TO0n$ 引脚的输出信号从无效电平变为有效电平。

复位： $TO0n$ 引脚的输出信号从有效电平变为无效电平。

2. n ：通道编号 ($n = 0, 1$)

(b) 置位/复位时序

为了在 PWM 输出时实现 0%/100%输出，使用从属通道，将产生主通道定时器中断($INTTM0n$)时 $TO0n$ 引脚/ $TO0n$ 位的设置时序延迟 1 个计数时钟周期。

同时产生设置条件和复位条件时，后者优先。

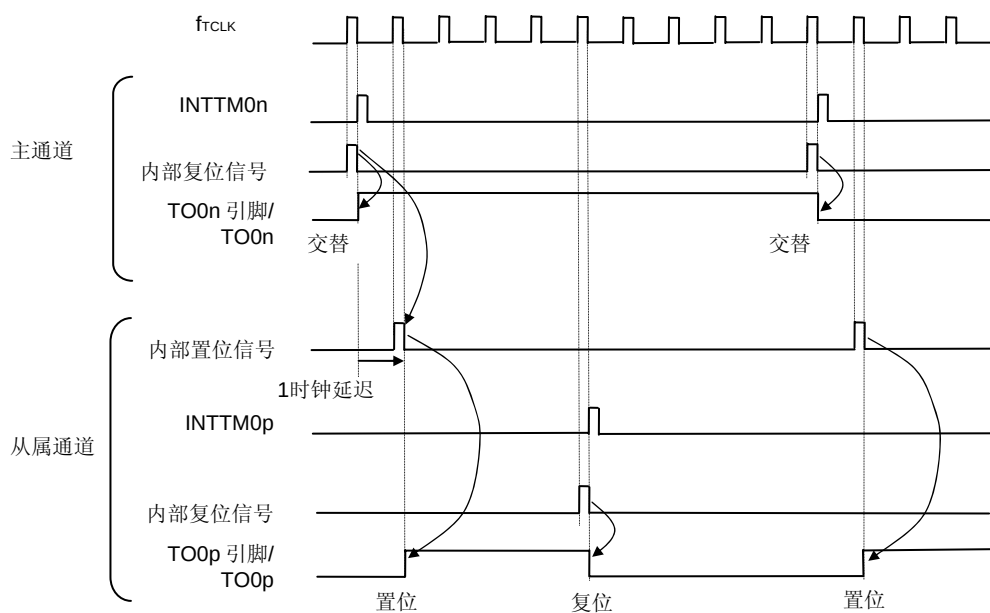
按以下方式设置主/从属通道时，设置/复位操作状态如图 6-34 所示。

主通道： $TOE0n = 1, TOM0n = 0, TOL0n = 0$

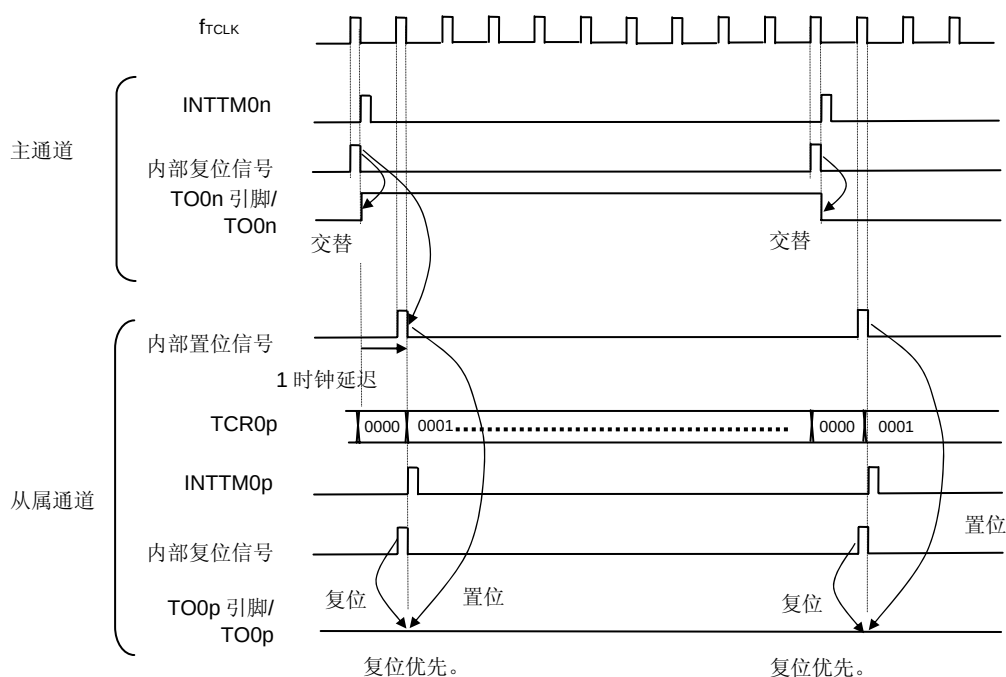
从属通道： $TOE0p = 1, TOM0p = 1, TOL0p = 0$

图 6-34. 置位/复位时序操作状态

(a) 基本操作时序



(b) 0 %占空时的操作时序



备注 1. 内部复位信号：TO0n 引脚复位/交替信号

内部置位信号：TO0n 引脚置位信号

2. n: 主通道编号 (n = 0)

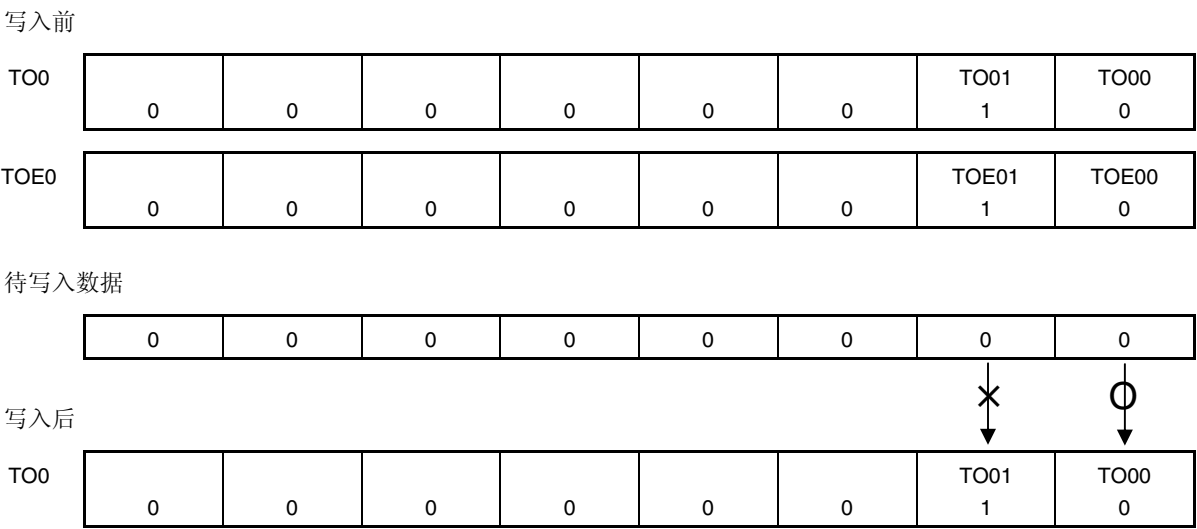
p: 从属通道编号 (p = 1)

6.6.4 TO0n 位的集中操作

在定时器输出寄存器 0(TO0)中，采用与定时器通道开始寄存器 0(TS0)相同的形式，在一个寄存器中配置有所有通道的设置位。因此，可以集中操作所有通道的 TO0n 位。

而且因为可以对只想要作为操作对象的通道输出 (TO0n) 的 TO0n 位进行写入 (TOE0n = 0)，所以对该寄存器的任意位都可以进行独立操作。

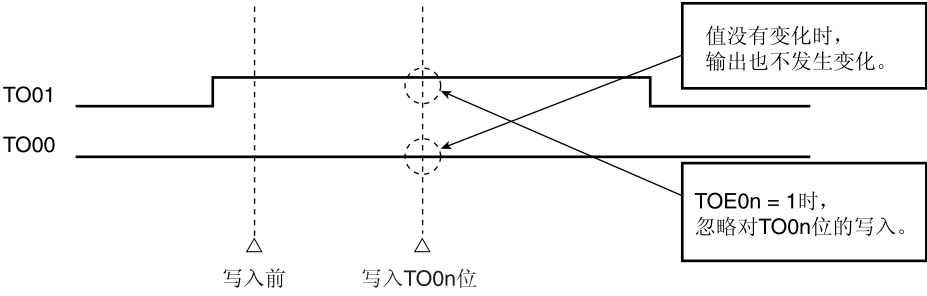
图 6-35. 集中操作 TO0n 位的示例



仅可写入 TOE0n = 0 的 TO0n 位，向 TOE0n = 1 的 TO0n 位的写入无效。

设置为 TOE0n = 1 的 TO0n（通道输出）不受写操作影响。即使对 TO0n 位执行写操作也是无效的，定时器操作导致的输出改变可以被正常执行。

图 6-36. 集中操作 TO0n 位时 TO0n 引脚的状态



备注 n: 通道编号 (n = 0, 1)

6.6.5 计数操作开始时的定时器中断和 TO0n 引脚输出

在间隔定时器模式或捕捉模式下，定时器模式寄存器 0n(T0R0n)的 MD0n0 位用于设置是否在计数开始时产生定时器中断。

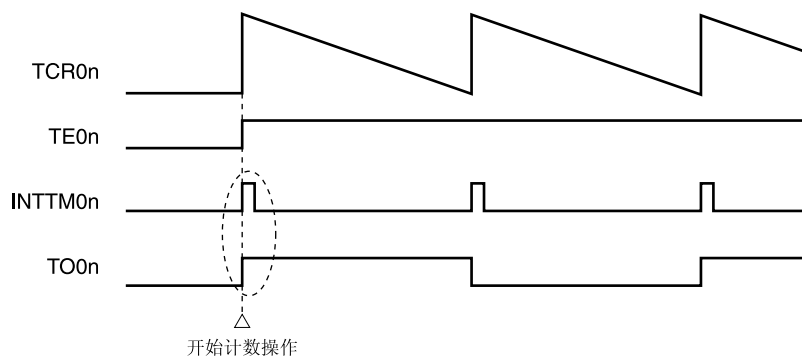
当 MD0n0 被设置为 1 时，可以通过定时器中断(INTTM0n)的产生来获知计数操作的开始时序。

在其他模式下，不控制计数操作开始时的定时器中断和 TO0n 输出。

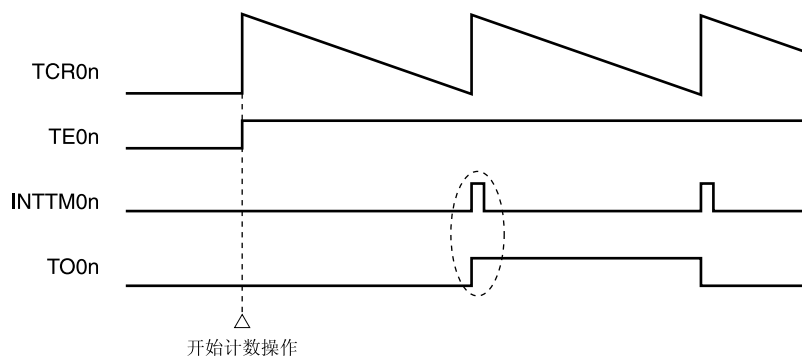
设置为间隔定时器模式(TOE0n = 1, TOM0n = 0) 时的操作示例如图 6-37 所示。

图 6-37. 计数操作开始时的定时器中断和 TO0n 输出的操作示例

(a) 当 MD0n0 设置为 1 时



(b) 当 MD0n0 设置为 0 时



当 MD0n0 被设置为 1 时，将在计数操作开始时输出定时器中断(INTTM0n)，TO0n 执行一次交替操作。

当 MD0n0 被设置为 0 时，不在计数操作开始时输出定时器中断(INTTM0n)，TO0n 也不改变。在计数一个周期之后，输出 INTTM0n，TO0n 执行一次交替操作。

备注 n: 通道编号 (n = 0, 1)

6.7 定时器阵列单元的单通道操作功能

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以固定间隔产生 INTTM0n（定时器中断）的基准定时器。
中断产生周期可以用下述表达式计算。

$$\text{INTTM0n (定时器中断) 的产生周期} = \text{计数时钟的周期} \times (\text{TDR0n 的设置值} + 1)$$

(2) 作为方波输出的操作

TO0n 将在产生 INTTM0n 时执行一次交替操作，并输出占空比为 50% 的方波。
从 TO0n 输出方波的周期和频率可以用下述表达式计算。

$$\bullet \text{ 从 TO0n 输出的方波周期} = \text{计数时钟的周期} \times (\text{TDR0n 的设置值} + 1) \times 2$$

$$\bullet \text{ 从 TO0n 输出的方波频率} = \text{计数时钟的频率} / \{(\text{TDR0n 的设置值} + 1) \times 2\}$$

定时器计数寄存器 0n (TCR0n) 在间隔定时器模式时充当递减计数器。

在定时器通道开始寄存器 0 (TS0) 的通道开始触发位 (TS0n) 被设置为 1 之后的第一个计数时钟时，定时器数据寄存器 0n (TDR0n) 的值被载入 TCR0n 寄存器。如果定时器模式寄存器 0n (TMR0n) 的 MD0n0 位在此时被设置为 0，则不输出 INTTM0n，TO0n 输出也不会交替。如果 TMR0n 寄存器的 MD0n0 位被设置为 1，则输出 INTTM0n，并交替 TO0n 输出。

此后，TCR0n 寄存器与计数时钟同步并进行递减计数。

当 TCR0n = 0000H 时，在下一个计数时钟输出 INTTM0n，并交替 TO0n 输出。同时，TDR0n 寄存器的值再次被载入 TCR0n 寄存器。之后重复同样的操作。

可以随时改写 TDR0n 寄存器。从下一个周期开始 TDR0n 寄存器的新值有效。

备注 n: 通道编号 (n = 0, 1)

图 6-38. 作为间隔定时器/方波输出的操作框图

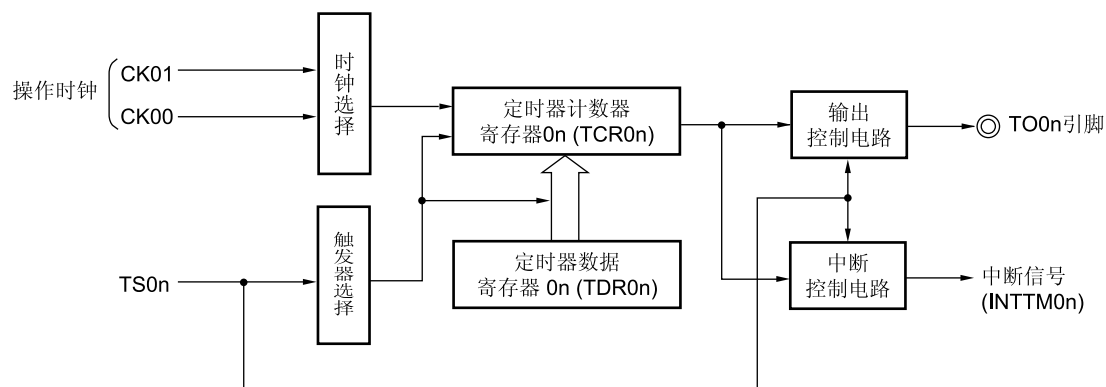
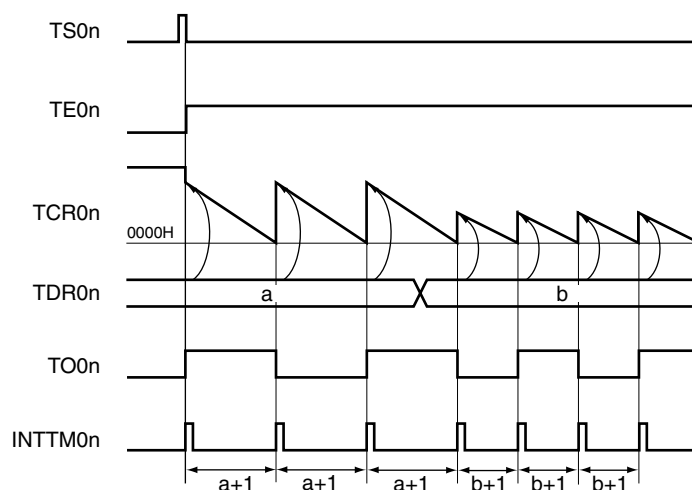


图 6-39. 作为间隔定时器/方波输出的操作基本时序示例 (MD0n0 = 1)

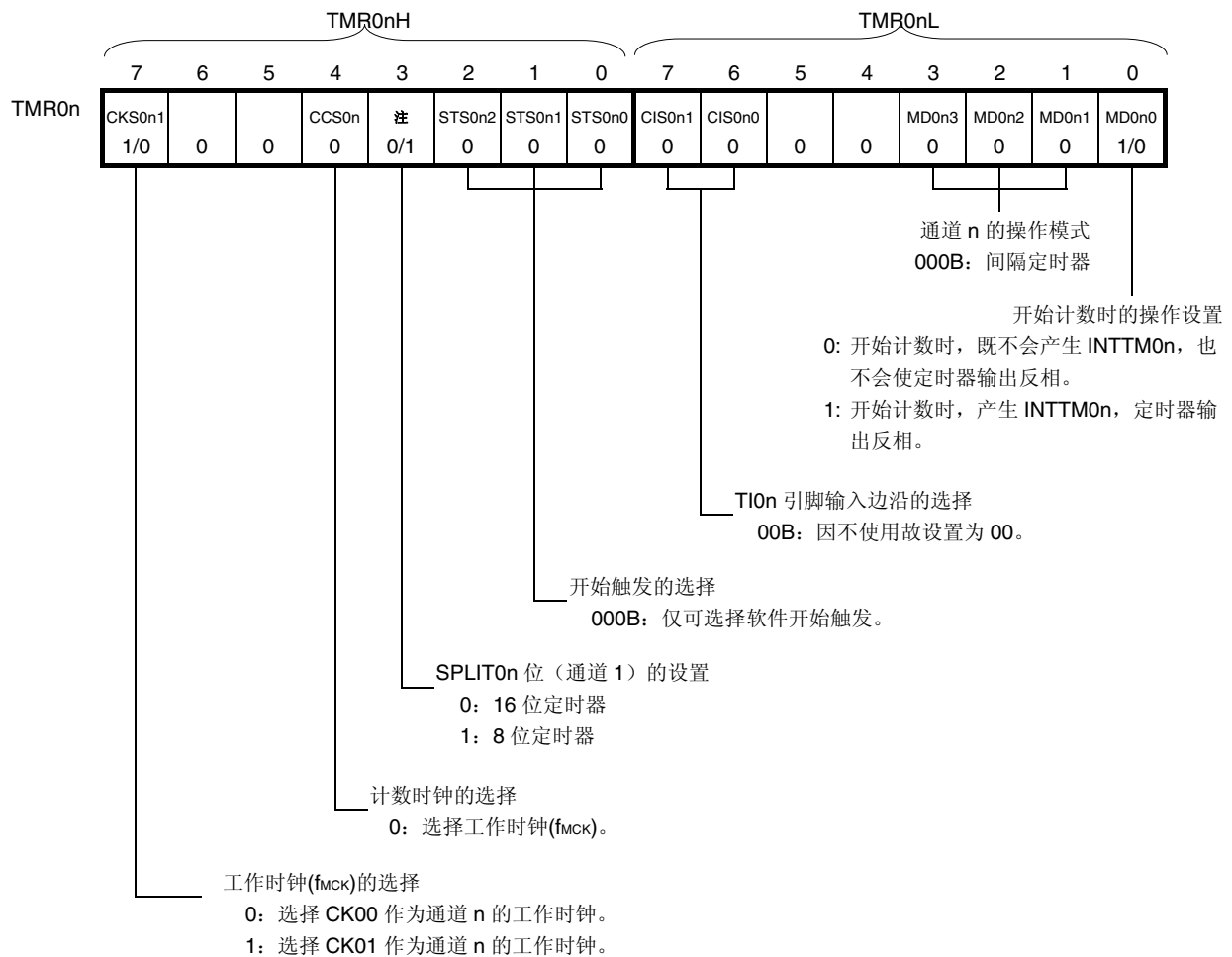


备注 1. n: 通道编号 (n = 0, 1)

2. TS0n: 定时器通道开始寄存器 0 (TS0) 的位 n
 TE0n: 定时器通道允许状态寄存器 0 (TE0) 的位 n
 TCR0n: 定时器计数寄存器 0n (TCR0n)
 TDR0n: 定时器数据寄存器 0n (TDR0n)
 TO0n: TO0n 引脚输出信号

图 6-40. 间隔定时器/方波输出时的寄存器设置内容示例 (1/2)

(a) 定时器模式寄存器 0n (TMR0nH, TMR0nL)



(b) 定时器输出寄存器 0 (TO0)

位 n	
TO0n	0: 由 TO0n 输出 0。
1/0	1: 由 TO0n 输出 1。

(c) 定时器输出允许寄存器 0 (TOE0)

位 n	
TOE0n	0: 停止通过计数动作来进行 TO0n 输出的操作。
1/0	1: 允许通过计数动作来进行 TO0n 输出的操作。

注 TMR01: SPLIT01 位

TMR00: 固定为 0

备注 n: 通道编号 (n = 0, 1)

图 6-40. 间隔定时器/方波输出时的寄存器设置内容示例 (2/2)

(d) 定时器输出电平寄存器 0 (TOL0)

TOL0	位 n	0: 当主通道输出模式 (TOM0n = 0) 时设置为 0。
	TOL0n 0	

(e) 定时器输出模式寄存器 0 (TOM0)

TOM0	位 n	0: 设置主通道输出模式。
	TOM0n 0	

备注 n: 通道编号 (n = 0, 1)

图 6-41. 间隔定时器/方波输出功能时的操作步骤 (1/2)

	软件操作	硬件状态
TAU初始设置		断电状态 (时钟供应被停止, 向各寄存器的写操作被禁止。)
	将外围允许寄存器0 (PER0)的TAU0EN位设置为1 (TAU0EN位为0时, 禁止读取/写入操作)。	上电状态。每个通道处于停止操作状态。 (开始时钟供应, 允许写入各寄存器。)
	设置定时器时钟选择寄存器0(TPS0)。 决定CK00和CK01的时钟频率。	
通道初始设置	设置定时器模式寄存器0n (TMR0n) (决定通道的操作模式)。 将间隔 (周期) 值设置到定时器数据寄存器0n (TDR0n)。	通道处于停止操作状态。 (供应时钟, 并产生一定的功耗。)
	使用TO0n输出时 将定时器输出模式寄存器0(TOM0)的TOM0n位设置为0 (主通道输出模式)。 将TOL0n位设置为0。 设置 TO0n位, 并决定 TO0n 输出的初始电平。	TO0n引脚处于Hi-Z输出状态。 当端口模式寄存器处于输出模式且端口寄存器为0时, 输出 TO0n的初始设置电平。
	将 TOE0n位设置为1, 并允许操作 TO0n。	因为操作通道处于停止状态, 所以TO0n不会变化。
	将端口寄存器和端口模式寄存器设置为0。	TO0n引脚输出TO0n 设置电平。
开始操作	(仅当使用 TO0n 输出并恢复操作时, 将 TOE0n位设置为1)。 将TS0n (TSH01)位设置为1。 TS0n (TSH01)位是触发位, 将自动返回为 0。	TEH0n (TEH01) = 1, 并且开始计数操作。 通过计数时钟输入将TDR0n 寄存器的值被载入定时器计数 寄存器 0n (TCR0n)。如果TMR0nL寄存器的MD0n0位被设置 为1, 将产生 INTTM0n, 同时, TO0n执行交替操作。
操作期间	不能更改 TMR0n寄存器、TOM0n和TOL0n位的设置值。 可以任意更改 TDR0n寄存器的设置值。 可以任意读取TCR0n寄存器。 不使用TSR0n寄存器。 可以更改TO0和TOE0寄存器的设置值。	计数器(TCR0n)递减计数。当计数值达到0000H时, TDR0n寄 存器的值再次被载入TCR0n寄存器, 继续计数操作。检测到 TCR0n = 0000H时, 产生 INTTM0n, TO0n执行交替操作。 之后重复以上操作。
停止操作	将TT0n (TTH01)位设置为1。 TT0n (TTH01)位是触发位, 将自动返回为 0。	TE0n (TEH01) = 0, 并且停止计数操作。 TCR0n寄存器保持计数值并停止计数操作。 TO0n输出不被初始化, 而是保持当前状态。
	将TOE0n位清除为0, 并对 TO0n位设值。	TO0n引脚输出设置电平到TO0n位。

恢复操作。

备注 n: 通道编号 (n = 0, 1)

注意事项 在 8 位定时器模式(SPLIT = 1)下使用通道 1 时, 禁止读取 TCR01H 和 TDR01H 寄存器。

图 6-41. 间隔定时器/方波输出功能时的操作步骤 (2/2)

	软件操作	硬件状态
TAU 停止	保持 TO0n引脚输出电平时 将希望保持的值设置到端口寄存器后，将 TO0n位清除为0。—————→	TO0n引脚输出电平由端口功能保持。
	不需要保持 TO0n 引脚输出电平时 不要求设置。 ----- 将PER0寄存器的TAU0EN位设置为0。—————→	断电状态 初始化所有电路，同时初始化各通道的SFR。 (TO0n位变为0，并且TO0n引脚变为端口模式。)

备注 n: 通道编号 (n = 0, 1)

注意事项 在 8 位定时器模式(SPLIT = 1)下使用通道 1 时，禁止读取 TCR01H 和 TDR01H 寄存器。

6.7.2 作为外部事件计数器操作

定时器阵列单元作为外部事件计数器使用时，用于对检测到的 $TI0n$ 引脚输入有效边沿（外部事件）的次数进行计数。达到指定计数次数时，事件计数器产生中断。指定的计数次数可以用下述表达式计算。

$$\text{指定计数次数} = \text{TDR0n 的设置值} + 1$$

定时器计数寄存器 $0n$ (TCR0n) 在事件计数器模式下用作递减计数器。

通过设置定时器通道开始寄存器 0 (TS0) 的任意通道开始触发位 (TS0n, TSH01) 为 1 ，将定时器数据寄存器 $0n$ (TDR0n) 的值载入 TCR0n 寄存器。

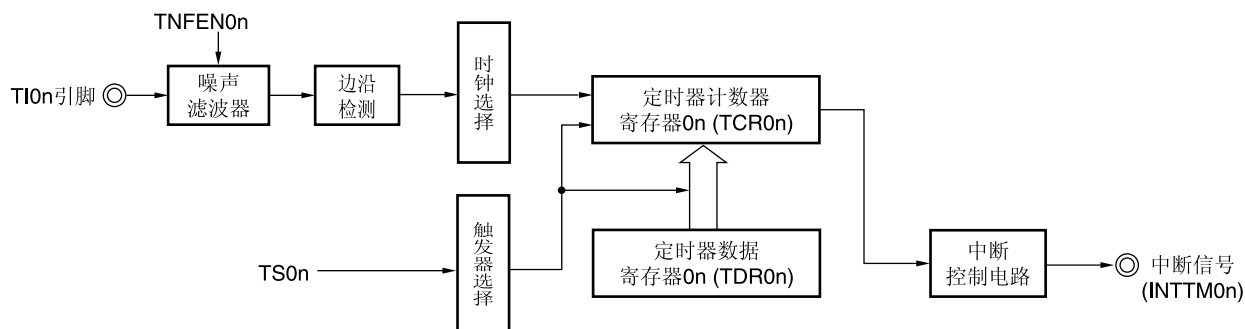
每当检测到 $TI0n$ 引脚输入有效边沿时，TCR0n 寄存器进行递减计数。当 $TCR0n = 0000H$ 时，TCR0n 寄存器将再次载入 TDR0n 寄存器的值，并输出 INTTM0n。

之后重复以上操作。

TO0n 引脚会根据外部事件而输出不规则的波形。所以请将定时器输出允许寄存器 0 (TOE0) 的 TOE0n 位设置为 0 以停止输出。

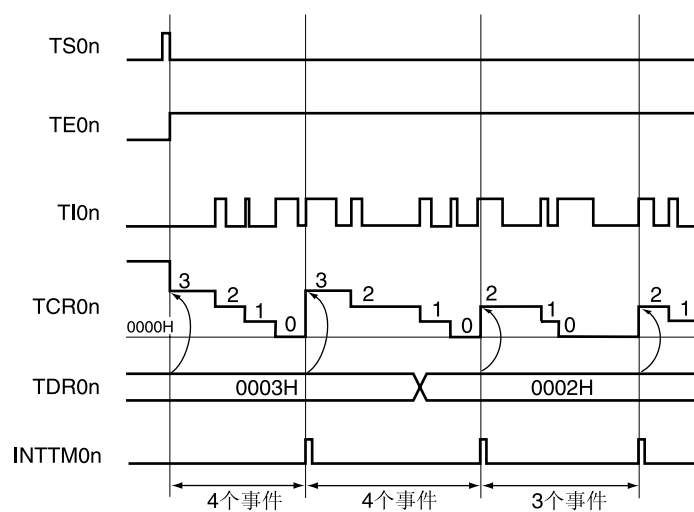
可以随时改写 TDR0n 寄存器。从下一个计数周期开始 TDR0n 寄存器的新值有效。

图 6-42. 作为外部事件计数器的操作框图



备注 n: 通道编号 (n = 0, 1)

图 6-43. 作为外部事件计数器的操作基本时序示例

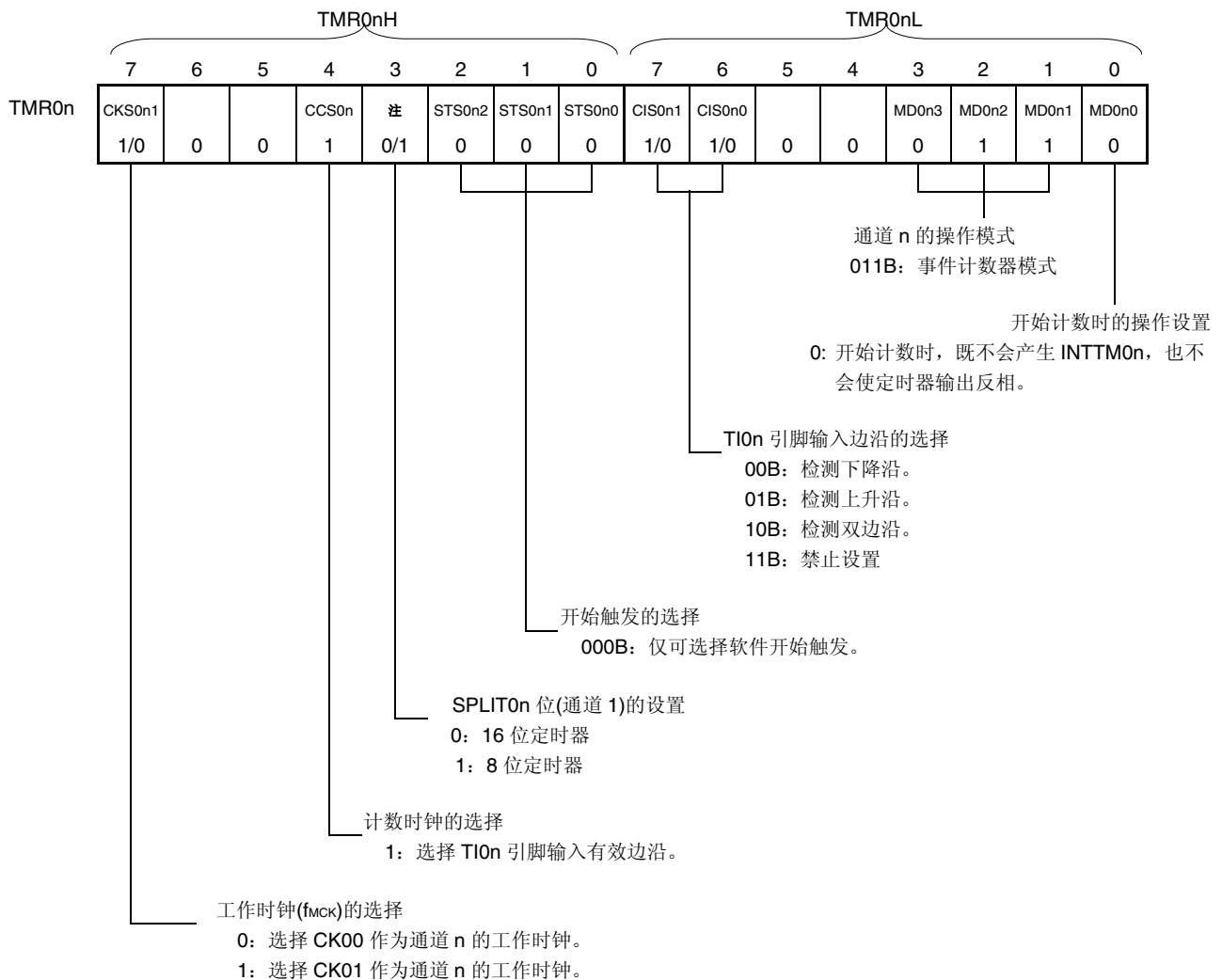


备注 1. n: 通道编号 (n = 0, 1)

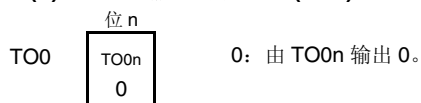
2. TS0n: 定时器通道开始寄存器 0 (TS0) 的位 n
- TE0n: 定时器通道允许状态寄存器 0 (TE0) 的位 n
- TI0n: TI0n 引脚输入信号
- TCR0n: 定时器计数寄存器 0n (TCR0n)
- TDR0n: 定时器数据寄存器 0n (TDR0n)

图 6-44. 外部事件计数器模式时的寄存器设置内容示例 (1/2)

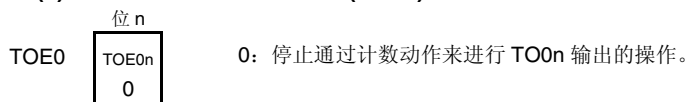
(a) 定时器模式寄存器 0n (TMR0nH, TMR0nL)



(b) 定时器输出寄存器 0 (TO0)



(c) 定时器输出允许寄存器 0 (TOE0)



注 TMR01: SPLIT01 位
TMR00: 固定为 0

备注 n: 通道编号 (n = 0, 1)

图 6-44. 外部事件计数器模式时的寄存器设置内容示例 (2/2)

(d) 定时器输出电平寄存器 0 (TOL0)

TOL0

位 n
TOL0n
0

 0: 当主通道输出模式 (TOM0n = 0) 时设置为 0。

(e) 定时器输出模式寄存器 0 (TOM0)

TOM0

位 n
TOM0n
0

 0: 设置主通道输出模式。

备注 n: 通道编号 (n = 0, 1)

图 6-45. 使用外部事件计数器功能时的操作步骤

	软件操作	硬件状态
TAU初始设置		断电状态 (时钟供应被停止, 向各寄存器的写操作被禁止。)
	将外围允许寄存器0 (PER0)的TAU0EN位设置为1 (TAU0EN位为0时, 禁止读取/写入操作)。	上电状态。每个通道处于停止操作状态。 (开始时钟供应, 允许写入各寄存器。)
	设置定时器时钟选择寄存器0(TPS0)。 决定CK00和CK01的时钟频率。	
通道初始设置	设置定时器模式寄存器0n (TMR0n) (决定通道的操作模式)。 将计数次数设置到定时器数据寄存器0n (TDR0n)。 设置噪声滤波器允许寄存器1 (NFEN1)。 将定时器输出允许寄存器0 (TOE0)的TOE0n位清除为0。	通道处于停止操作状态。 (供应时钟, 并产生一定的功耗。)
开始操作	将TS0n位设置为1。 TS0n位是触发位, 将自动返回为0。	TE0n = 1, 并且开始计数操作。 TDR0n 寄存器的值被载入定时器计数寄存器 0n (TCR0n), 进入TI0n 引脚输入边沿的等待检测状态。
操作期间	可以任意更改 TDR0n寄存器的设置值。 可以任意读取TCR0n寄存器。 不使用TSR0n寄存器。 不可更改 TMR0n寄存器、TOM0n、TOL0n、TO0n和TOE0n位的设置值。	每当检测到TI0n引脚的输入边沿时, 计数器(TCR0n) 都会递减计数。当计数值达到0000H时, TDR0n寄存器的值再次被载入TCR0n寄存器, 继续计数操作。检测到TCR0n = 0000H时输出INTTM0n。 之后重复以上操作。
停止操作	将TT0n位设置为1。 TT0n位是触发位, 将自动返回为0。	TE0n = 0, 并且停止计数操作。 TCR0n寄存器保持计数值并停止计数操作。
TAU停止	将PER0寄存器的TAU0EN位设置为0。	断电状态 初始化所有电路, 同时初始化各通道的SFR。

恢复操作。

备注 n: 通道编号 (n = 0, 1)

6.7.3 作为分频器的操作(仅限通道 0)

定时器阵列单元可用作分频器，将输入至 TI00 引脚的时钟进行分频，并从 TO00 引脚输出结果。

从 TO00 输出的分频时钟频率可使用下列表达式来计算。

- 当选择上升沿/下降沿时：
分频时钟频率 = 输入时钟频率 / {(TDR00 的设置值 + 1) × 2}
- 当选择双边沿时：
分频时钟频率 ≅ 输入时钟频率 / (TDR00 的设置值 + 1)

定时器计数寄存器 00 (TCR00) 在间隔定时器模式时充当递减计数器。

将定时器通道开始寄存器 0 (TS0) 的通道开始触发位 (TS00) 设置为 1 之后，TCR00 寄存器将在检测到 TI00 有效边沿时载入定时器数据寄存器 00 (TDR00) 的值。

如果定时器模式寄存器 00 (TMR00) 的 MD000 位在此时被设置为 0，则不输出 INTTM00，也不交替 TO00 输出。如果定时器模式寄存器 00 (TMR00) 的 MD000 位被设置为 1，则输出 INTTM00，并交替 TO00 输出。

此后，TCR00 寄存器在 TI00 引脚的有效边沿处递减计数。当 TCR00 = 0000H 时，交替 TO00 输出。同时，TCR00 寄存器再次载入 TDR00 寄存器的值并继续计数。

如果选择检测 TI00 引脚的两个边沿，则输入时钟的占空比误差将影响 TO00 输出的分频时钟周期。

TO00 输出时钟的周期包含 1 个工作时钟周期的采样误差。

TO00 输出的时钟周期 = 理想的 TO00 输出时钟周期 ± 1 个工作时钟周期 (误差)

可以随时改写 TDR00 寄存器。从下一个计数周期开始 TDR00 寄存器的新值有效。

图 6-46. 作为分频器的操作框图

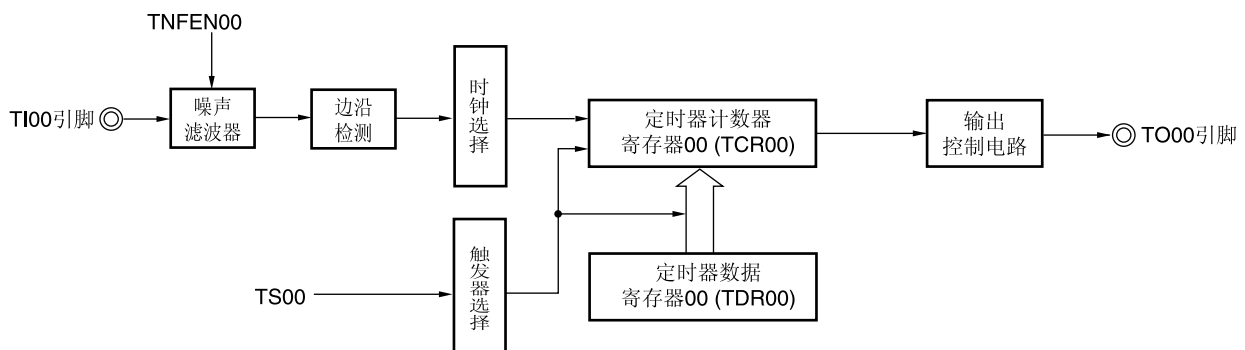
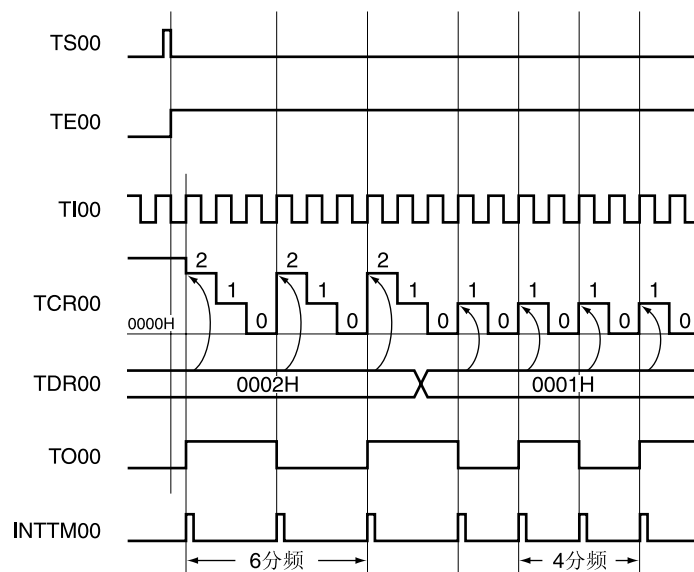


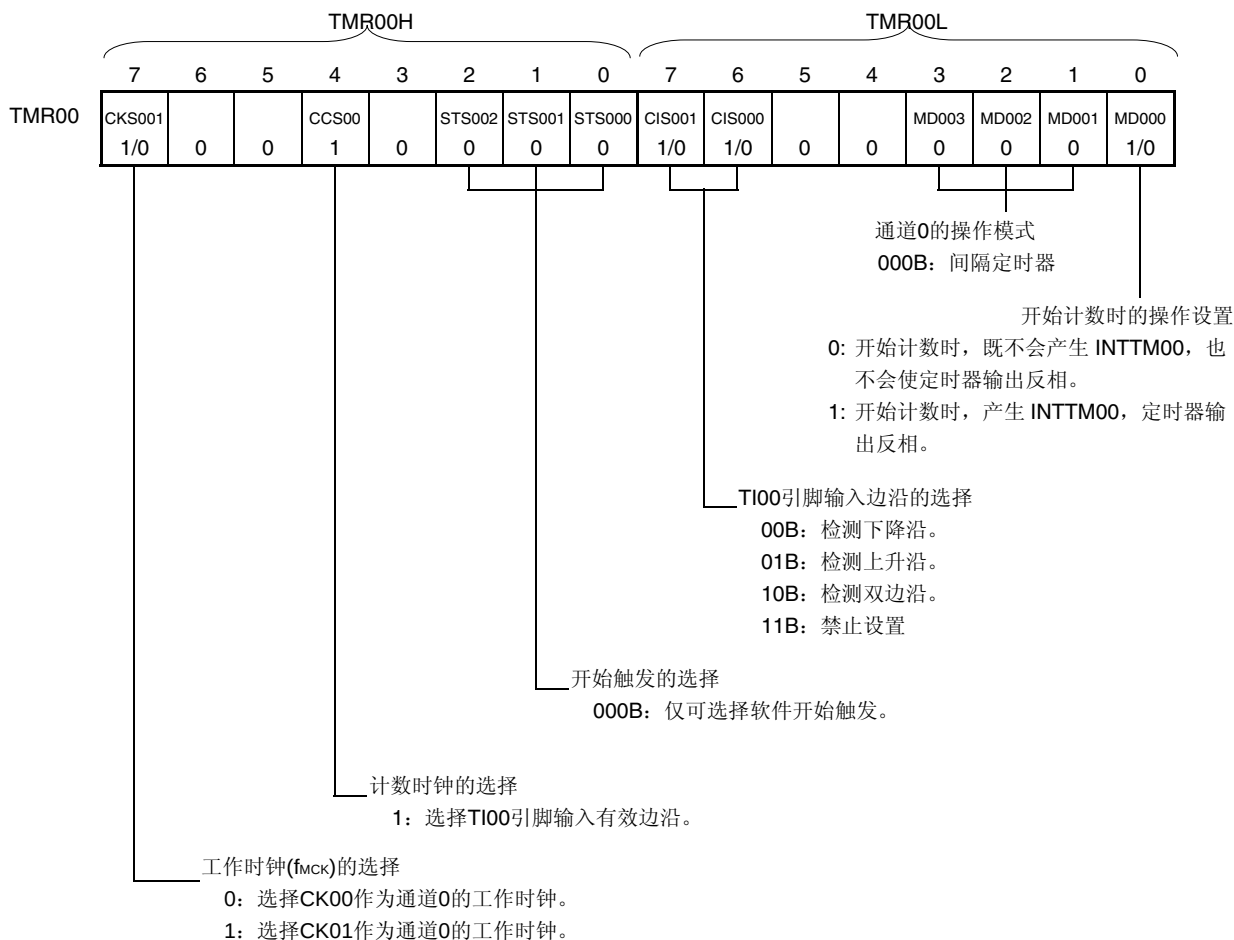
图 6-47. 作为分频器的操作基本时序示例 (MD000 = 1)



备注 TS00: 定时器通道开始寄存器 0 (TS0) 的位 n
 TE00: 定时器通道允许状态寄存器 0 (TE0) 的位 n
 TI00: TI00 引脚输入信号
 TCR00: 定时器计数寄存器 00 (TCR00)
 TDR00: 定时器数据寄存器 00 (TDR00)
 TO00: TO00 引脚输出信号

图 6-48. 作为分频器操作时的寄存器设置内容示例

(a) 定时器模式寄存器 00 (TMR00H, TMR00L)



(b) 定时器输出寄存器 0 (TO0)

位 0	
TO0	TO00
	1/0

0: 由 TO00 输出 0。
1: 由 TO00 输出 1。

(c) 定时器输出允许寄存器 0 (TOE0)

位 0	
TOE0	TOE00
	1/0

0: 停止通过计数动作来进行 TO00 输出的操作。
1: 允许通过计数动作来进行 TO00 输出的操作。

(d) 定时器输出电平寄存器 0 (TOL0)

位 0	
TOL0	TOL00
	0

0: 当主通道输出模式 (TOM00 = 0) 时设置为 0。

(e) 定时器输出模式寄存器 0 (TOM0)

位 0	
TOM0	TOM00
	0

0: 设置主通道输出模式。

图 6-49. 使用分频器功能时的操作步骤

	软件操作	硬件状态
TAU初始设置		断电状态 (时钟供应被停止, 向各寄存器的写操作被禁止。)
	将外围允许寄存器0 (PER0)的TAU0EN位设置为1 (TAU0EN位为0时, 禁止读取/写入操作)。	上电状态。每个通道处于停止操作状态。 (开始时钟供应, 允许写入各寄存器。)
	设置定时器时钟选择寄存器0(TPS0)。 决定CK00和CK01的时钟频率。	
通道初始设置	设置定时器模式寄存器0n (TMR0n)(决定通道的操作模式, 并选择检测边沿)。 将间隔(周期)值设置到定时器数据寄存器00(TDR00)。 设置噪声滤波器允许寄存器1 (NFEN1)。	通道处于停止操作状态。 (供应时钟, 并产生一定的功耗。)
	将定时器输出模式寄存器0(TOM0)的TOM00位设置为0 (主通道输出模式)。 将TOL00位设置为0。 设置TO00位并决定TO00输出的初始电平。	TO00引脚处于Hi-Z输出状态。 当端口模式寄存器处于输出模式且端口寄存器为0时, 输出TO00的初始设置电平。
	将TOE00位设置为1, 并允许操作TO00。	因为操作通道处于停止状态, 所以TO00不会变化。
	将端口寄存器和端口模式寄存器设置为0。	TO00引脚输出TO00的设置电平。
开始操作	将TOE00位设置为1(仅限恢复定时器操作时)。 将TS00位设置为1。 TS00位是触发位, 将自动返回为0。	TE00 = 1, 并且开始计数操作。 输入计数时钟时, TDR00寄存器的值被载入定时器计数寄存器00(TCR00)。如果TMR00寄存器的MD000位被设置为1, 将产生INTTM00, 同时, TO00执行交替操作。
操作期间	可以任意更改TDR00寄存器的设置值。 可以任意读取TCR00寄存器。 不使用TSR00寄存器。 可以更改TO0和TOE0寄存器的设置值。 不能更改TMR00寄存器、TOM00和TOL00位的设置值。	计数器(TCR00)递减计数。当计数值达到0000H时, TDR00寄存器的值将被再次载入TCR00寄存器, 并继续计数操作。检测到TCR00 = 0000H时, 产生INTTM00, TO00执行交替操作。之后重复以上操作。
停止操作	将TT00位设置为1。 TT00位是触发位, 将自动返回为0。	TE00 = 0, 并且停止计数操作。 TCR00寄存器保持计数值并停止计数操作。 TO00输出不被初始化, 而是保持当前状态。
	将TOE00设置为0, 并对TO00位设值。	TO00引脚输出TO00的设置电平。
TAU停止	保持TO00引脚输出电平时 将希望保持的值设置到端口寄存器后, 将TO00位清除为0。 不需要保持 TO00引脚输出电平时 不要求设置。	TO00引脚输出电平由端口功能保持。
	将PER0寄存器的TAU0EN位设置为0。	断电状态 初始化所有电路, 同时初始化各通道的SFR。 (TO00位清除为0, TO00引脚被设置为端口模式)。

恢复操作。

6.7.4 作为输入脉冲间隔测量的操作

可以在 TI0n 有效边沿捕捉计数值，并可测量输入 TI0n 的脉冲的间隔。

脉冲间隔可以用下述表达式计算。

$$\text{TI0n 输入脉冲间隔} = \text{计数时钟的周期} \times ((10000\text{H} \times \text{TSR0n: OVF}) + (\text{TDR0n 的捕捉值} + 1))$$

注意事项 使用定时器模式寄存器 0n (TMR0n) 的 CKS0n1 位选择的工作时钟对 TI0n 引脚输入进行采样，因此会发生最大为一个工作时钟周期的误差。

定时器计数寄存器 0n (TCR0n) 在捕捉模式下充当递增计数器。

在定时器通道开始寄存器 0 (TS0) 的通道开始触发位 (TS0n) 被设置为 1 时，TCR0n 寄存器将与计数时钟同步从 0000H 开始递增计数。

检测到 TI0n 引脚输入有效边沿时，TCR0n 寄存器的计数值被传送（捕捉）至定时器数据寄存器 0n (TDR0n)，同时，TCR0n 寄存器被清除为 0000H，并输出 INTTM0n。此时如果计数器溢出，则定时器状态寄存器 0n (TSR0n) 的 OVF 位被设置为 1。如果计数器不溢出，则 OVF 位被清除。之后重复以上操作。

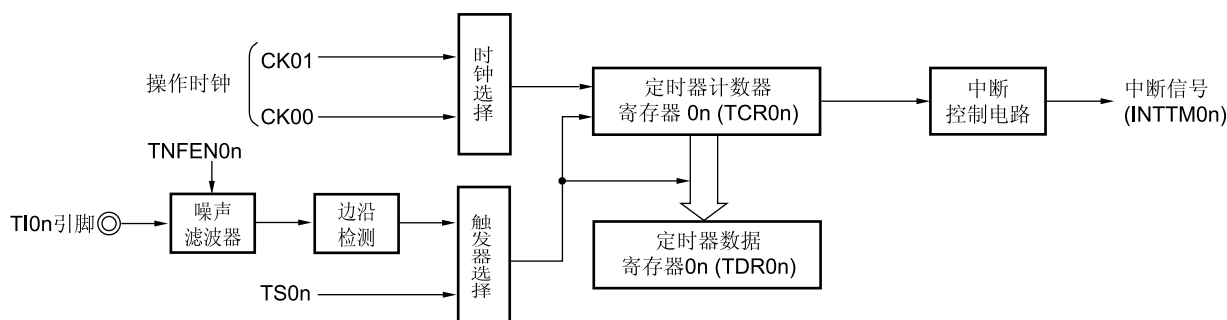
计数值被捕捉至 TDR0n 寄存器时，依据测量期间计数器是否发生溢出，TSR0n 寄存器的 OVF 位将被更新。从而可以确认捕捉值的溢出状态。

如果计数器进行了两个以上周期的完整计数，也能发生溢出，TSR0n 寄存器的 OVF 位可被设置为 1。但是，在发生两次以上的溢出时，不能通过 OVF 位测量出正常间隔值。

TMR0n 寄存器的 STS0n2 至 STS0n0 位设置为 001B，TI0n 的有效边沿可以用作开始触发和捕捉触发。

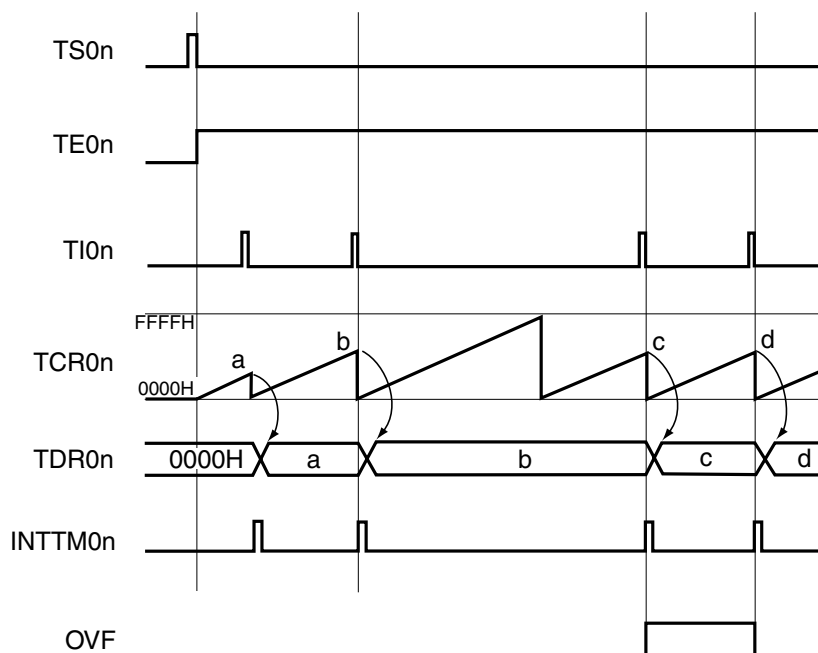
当 TE0n = 1 时，可以使用软件操作 (TS0n = 1) 来代替 TI0n 引脚输入作为捕捉触发。

图 6-50. 作为输入脉冲间隔测量的操作框图



备注 n: 通道编号 (n = 0, 1)

图 6-51. 作为输入脉冲间隔测量的操作基本时序示例 (MD0n0 = 0)

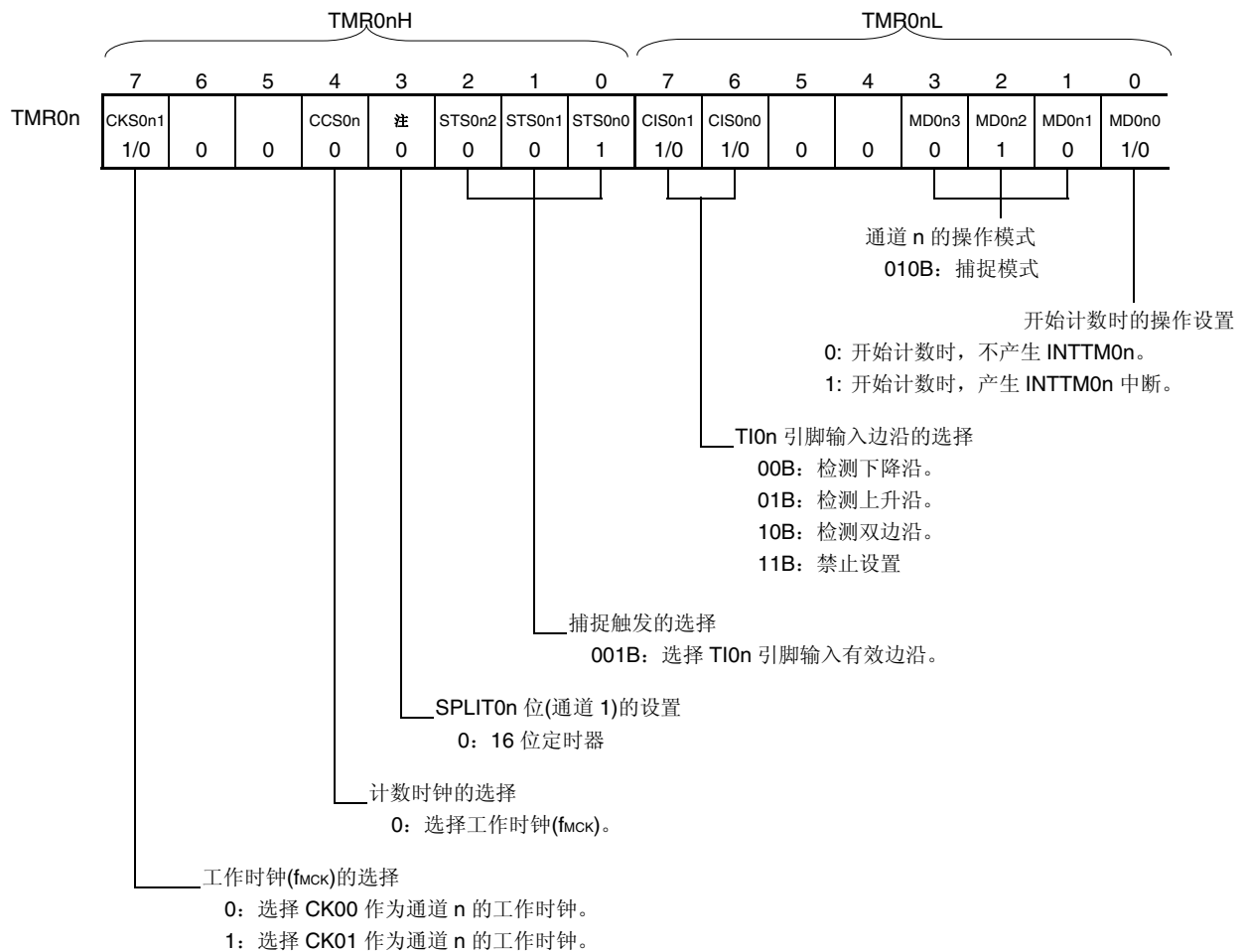


备注 1. n: 通道编号 (n = 0, 1)

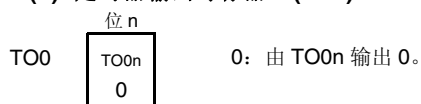
2. TS0n: 定时器通道开始寄存器 0 (TS0) 的位 n
- TE0n: 定时器通道允许状态寄存器 0 (TE0) 的位 n
- TI0n: TI0n 引脚输入信号
- TCR0n: 定时器计数寄存器 0n (TCR0n)
- TDR0n: 定时器数据寄存器 0n (TDR0n)
- OVF: 定时器状态寄存器 0n (TSR0n) 的位 0

图 6-52. 输入脉冲间隔测量时的寄存器设置内容示例

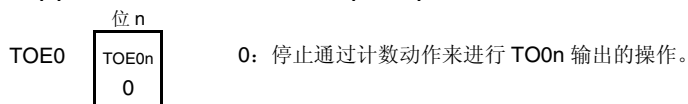
(a) 定时器模式寄存器 0n (TMR0nH, TMR0nL)



(b) 定时器输出寄存器 0 (TO0)



(c) 定时器输出允许寄存器 0 (TOE0)



注 TMR01: SPLIT01 位
TMR00: 固定为 0

备注 n: 通道编号 (n = 0, 1)

(d) 定时器输出电平寄存器 0 (TOL0)

TOL0

位 n
TOL0n
0

 0: 当主通道输出模式 (TOM0n = 0) 时设置为 0。

(e) 定时器输出模式寄存器 0 (TOM0)

TOM0

位 n
TOM0n
0

 0: 设置主通道输出模式。

备注 n: 通道编号 (n = 0, 1)

图 6-53. 使用输入脉冲间隔测量功能时的操作步骤

	软件操作	硬件状态
TAU初始设置		断电状态 (时钟供应被停止, 向各寄存器的写操作被禁止。)
	将外围允许寄存器0 (PER0)的TAU0EN位设置为1 (TAU0EN位为0时, 禁止读取/写入操作)。	上电状态。每个通道处于停止操作状态。 (开始时钟供应, 允许写入各寄存器。)
	设置定时器时钟选择寄存器0(TPS0)。 决定CK00和CK01的时钟频率。	
通道初始设置	设置定时器模式寄存器0n (TMR0n) (决定通道的操作模式)。 设置噪声滤波器允许寄存器1 (NFEN1)。	通道处于停止操作状态。 (供应时钟, 并产生一定的功耗。)
开始操作	将TS0n位设置为1。 TS0n位是触发位, 将自动返回为0。	TE0n = 1, 并且开始计数操作。 输入计数时钟时, 定时器计数寄存器0n (TCR0n)被清除为0000H。 当TMR0n寄存器的MD0n0位为1时, 产生INTTM0n。
操作期间	仅可以更改TMR0n寄存器的CIS0n1和CIS0n0位的设置值。 可以任意读取TDR0n寄存器。 可以任意读取TCR0n寄存器。 可以任意读取TSR0n寄存器。 不能更改 TOM0n、TOL0n、TO0n和TOE0n位的设置值。	计数器(TCR0n)从0000H开始递增计数。当检测到TI0n引脚输入有效边沿时, 计数值被传送(捕捉)至定时器数据寄存器0n (TDR0n)。同时, TCR0n寄存器被清除为0000H, 并产生 INTTM0n 信号。 此时如果发生溢出, 则定时器状态寄存器0n (TSR0n)的OVF位被设置; 如果不发生溢出, 则OVF位被清除。 之后重复以上操作。
停止操作	将TT0n位设置为1。 TT0n位是触发位, 将自动返回为0。	TE0n = 0, 并且停止计数操作。 TCR0n寄存器保持计数值并停止计数操作。 同时保持TSR0n寄存器的OVF位。
TAU停止	将PER0寄存器的TAU0EN位设置为0。	断电状态 初始化所有电路, 同时初始化各通道的SFR。

恢复操作。

备注 n: 通道编号 (n = 0, 1)

6.7.5 输入信号高/低电平宽度测量的操作

通过在 $TI0n$ 引脚输入的一个边沿开始计数，在另一个边沿捕捉计数次数，可以测量 $TI0n$ 的信号宽度（高电平宽度/低电平宽度）。 $TI0n$ 的信号宽度可用下述表达式计算。

$$TI0n \text{ 输入的信号宽度} = \text{计数时钟的周期} \times ((10000H \times \text{TSR0n: OVF}) + (\text{TDR0n 的捕捉值} + 1))$$

注意事项 使用定时器模式寄存器 $0n$ ($TMR0n$) 的 $CKS0n1$ 位选择的工作时钟对 $TI0n$ 引脚输入进行采样，因此会发生一个工作时钟周期的误差。

定时器计数寄存器 $0n$ ($TCR0n$) 在捕捉&单计数模式时用作递增计数器。

当定时器通道开始寄存器 0 ($TS0$) 的通道开始触发位 ($TS0n$) 被设置为 1 时， $TE0n$ 位将被设置为 1 ，同时 $TI0n$ 引脚被设置为开始边沿检测等待状态。

当检测到 $TI0n$ 引脚输入开始边沿（测量高电平宽度时为 $TI0n$ 引脚输入的上升沿）时，计数器与计数时钟同步并从 $0000H$ 开始递增计数。之后检测到有效捕捉边沿（测量高电平宽度时为 $TI0n$ 引脚输入的下降沿）时，计数值被传送至定时器数据寄存器 $0n$ ($TDR0n$)，同时输出 $INTTM0n$ 。此时如果计数器溢出，则定时器状态寄存器 $0n$ ($TSR0n$) 的 OVF 位被设置为 1 。如果计数器不溢出，则 OVF 位被清除。当值达到 传送至 $TDR0n$ 寄存器的值 $+ 1$ 时，停止 $TCR0n$ 寄存器，同时 $TI0n$ 引脚进入开始边沿检测等待状态。之后重复以上操作。

计数值被捕捉至 $TDR0n$ 寄存器时，依据测量期间计数器是否发生溢出， $TSR0n$ 寄存器的 OVF 位将被更新。从而可以确认捕捉值的溢出状态。

如果计数器进行了两个以上周期的完整计数，也能发生溢出， $TSR0n$ 寄存器的 OVF 位可被设置为 1 。但是，在发生两次以上的溢出时，不能通过 OVF 位测量出正常间隔值。

可以通过 $TMR0n$ 寄存器的 $CIS0n1$ 和 $CIS0n0$ 位来选择测量 $TI0n$ 引脚的高电平宽度或低电平宽度。

由于该功能用于测量 $TI0n$ 引脚输入的信号宽度，因此，在 $TE0n$ 位为 1 时，不能将 $TS0n$ 位设置为 1 。

$TMR0n$ 寄存器的 $CIS0n1, CIS0n0 = 10B$: 测量低电平宽度。

$TMR0n$ 寄存器的 $CIS0n1, CIS0n0 = 11B$: 测量高电平宽度。

图 6-54. 作为输入信号的高/低电平宽度测量的操作框图

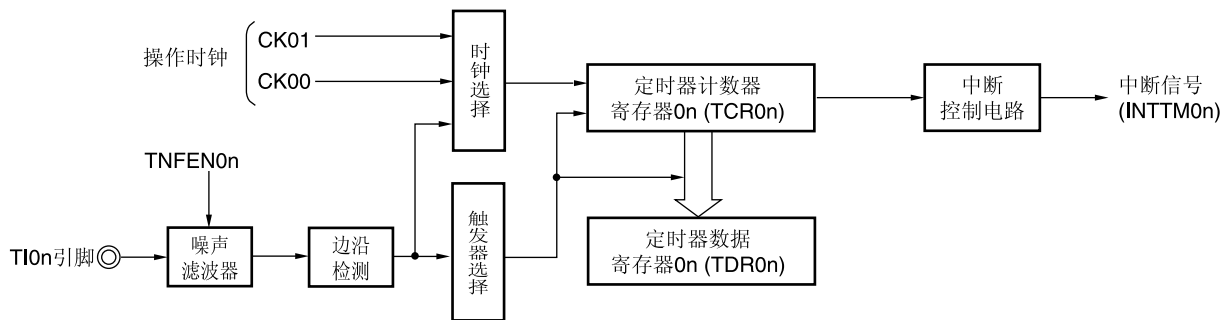
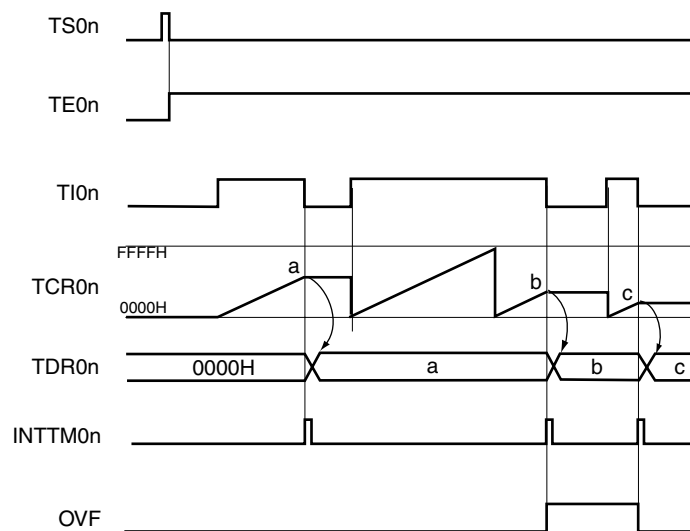


图 6-55. 作为输入信号的高/低电平宽度测量的操作基本时序示例

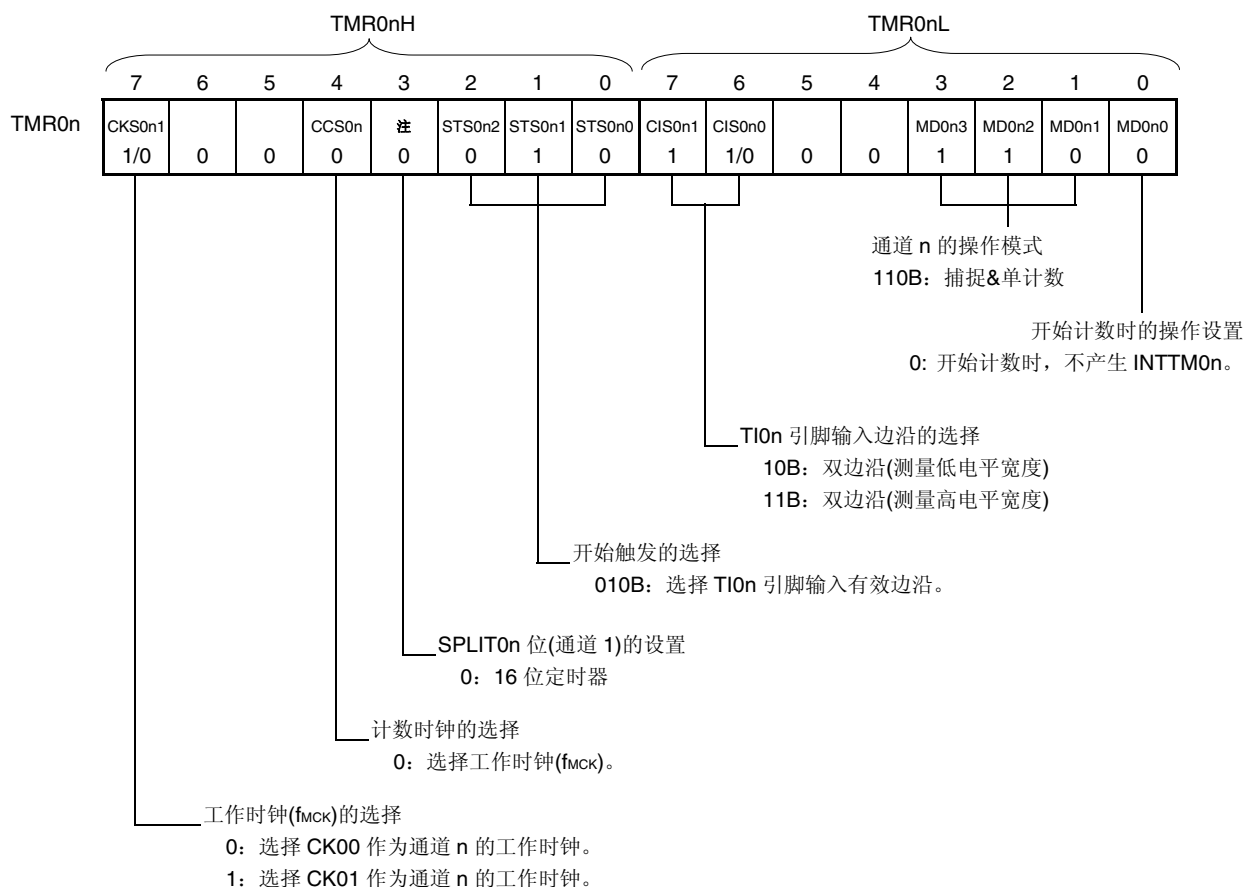


备注 1. n : 通道编号 ($n = 0, 1$)

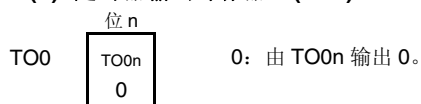
2. $TS0n$: 定时器通道开始寄存器 0 (TS0) 的位 n
- $TE0n$: 定时器通道允许状态寄存器 0 (TE0) 的位 n
- $TI0n$: $TI0n$ 引脚输入信号
- $TCR0n$: 定时器计数寄存器 $0n$ (TCR $0n$)
- $TDR0n$: 定时器数据寄存器 $0n$ (TDR $0n$)
- OVF : 定时器状态寄存器 $0n$ (TSR $0n$) 的位 0

图 6-56. 输入信号的高/低电平宽度测量时的寄存器设置内容示例

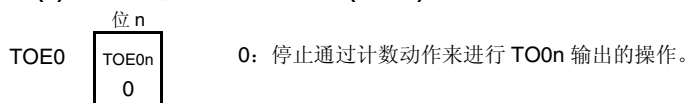
(a) 定时器模式寄存器 0n (TMR0nH, TMR0nL)



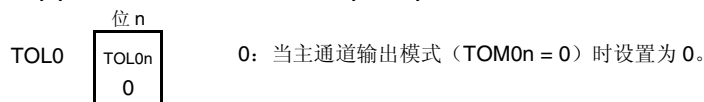
(b) 定时器输出寄存器 0 (TO0)



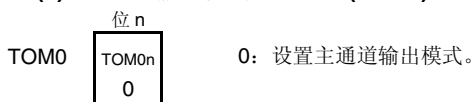
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)



注 TMR01: SPLIT01 位

TMR00: 固定为 0

备注 n: 通道编号 (n = 0, 1)

图 6-57. 使用输入信号高/低电平宽度测量功能时的操作步骤

	软件操作	硬件状态
TAU初始设置		断电状态 (时钟供应被停止, 向各寄存器的写操作被禁止。)
	将外围允许寄存器0 (PER0)的TAU0EN位设置为1 (TAU0EN位为0时, 禁止读取/写入操作)。	上电状态。每个通道处于停止操作状态。 (开始时钟供应, 允许写入各寄存器。)
	设置定时器时钟选择寄存器0(TPS0)。 决定CK00和CK01的时钟频率。	
通道初始设置	设置定时器模式寄存器0n (TMR0n) (决定通道的操作模式)。 设置噪声滤波器允许寄存器1 (NFEN1)。 将TOE0n位设置为0, 并且停止TO0n操作。	通道处于停止操作状态。 (供应时钟, 并产生一定的功耗。)
开始操作	将TS0n位设置为1。 TS0n位是触发位, 将自动返回为0。	TE0n = 1, 变为TI0n引脚开始边沿检测等待状态。
	检测TI0n引脚输入计数开始有效边沿。	将定时器计数寄存器 0n (TCR0n) 清除为0000H 并开始递增计数。
操作期间	可以任意更改 TDR0n寄存器的设置值。 可以任意读取TCR0n寄存器。 不使用TSR0n寄存器。 不可更改 TMR0n寄存器、TOM0n、TOL0n、TO0n和TOE0n位的设置值。	当检测到TI0n引脚开始边沿时, 计数器(TCR0n) 从0000H开始递增计数。如果检测到TI0n引脚的捕捉边沿, 则计数值将被传送到定时器数据寄存器0n (TDR0n), 并产生INTTM0n。 此时如果发生溢出, 则定时器状态寄存器0n (TSR0n)的OVF位被设置; 如果不发生溢出, 则OVF位被清除。停止TCR0n寄存器计数操作, 直至检测到下一个TI0n引脚开始边沿为止。
停止操作	将TT0n位设置为1。 TT0n位是触发位, 将自动返回为0。	TE0n = 0, 并且停止计数操作。 TCR0n寄存器保持计数值并停止计数操作。 同时保持TSR0n寄存器的OVF位。
TAU停止	将PER0寄存器的TAU0EN位设置为0。	断电状态 初始化所有电路, 同时初始化各通道的SFR。

恢复操作。

备注 n: 通道编号 (n = 0, 1)

6.7.6 作为延迟计数器的操作

当检测到 $TI0n$ 引脚输入的有效边沿（外部事件）时，可以开始递减计数，并在任意指定间隔后产生 $INTTM0n$ （定时器中断）。

也可通过软件设置 $TS0n = 1$ ，以任意间隔产生 $INTTM0n$ （定时器中断），并在 $TE0n$ 期间开始递减计数。

中断产生周期可以用下述表达式计算。

$$INTTM0n \text{ (定时器中断) 的产生周期} = \text{计数时钟的周期} \times (\text{TDR0n 的设置值} + 1)$$

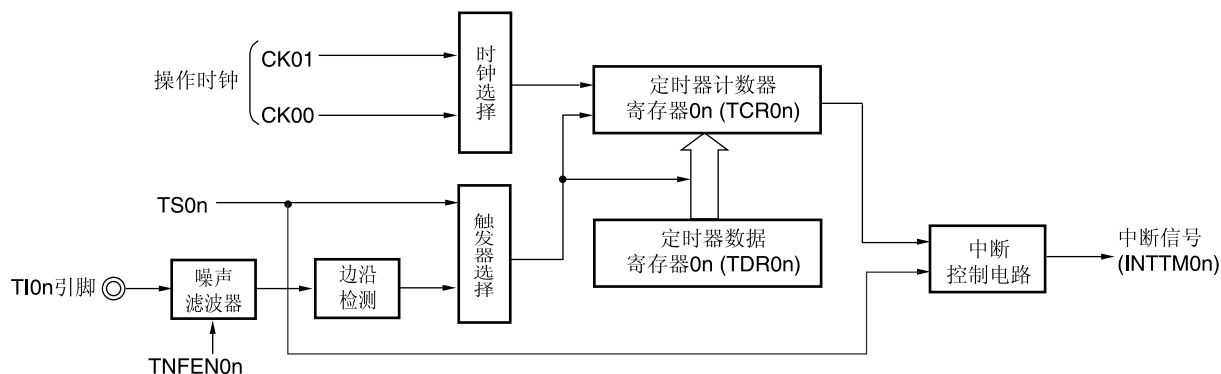
定时器计数寄存器 $0n$ ($TCR0n$) 在单计数模式时用作递减计数器。

当定时器通道开始寄存器 0 ($TS0$) 的通道开始触发位 ($TS0n$) 被设置为 1 时， $TE0n$ 位将被设置为 1 ，同时设置为 $TI0n$ 引脚输入有效边沿检测等待状态。

检测到 $TI0n$ 引脚输入有效边沿时定时器计数寄存器 $0n$ ($TCR0n$) 开始动作，并载入定时器数据寄存器 $0n$ ($TDR0n$) 的值。 $TCR0n$ 寄存器与计数时钟同步，从载入的 $TDR0n$ 寄存器的值开始递减计数。当 $TCR0n = 0000H$ 时，输出 $INTTM0n$ 中断并停止计数，直至检测到下一个 $TI0n$ 引脚输入有效边沿为止。

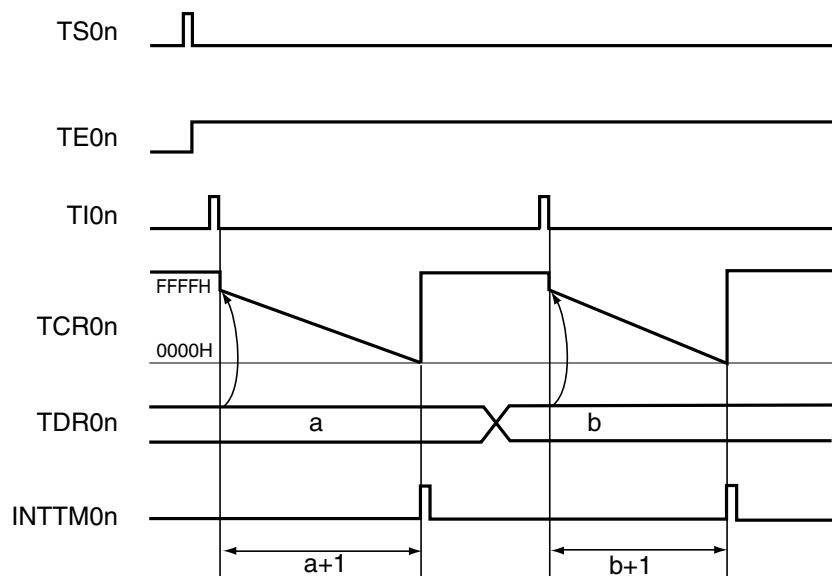
可以随时改写 $TDR0n$ 寄存器。从下一个周期开始 $TDR0n$ 寄存器的新值有效。

图 6-58. 作为延迟计数器的操作框图



备注 n: 通道编号 (n = 0, 1)

图 6-59. 作为延迟计数器的操作基本时序示例

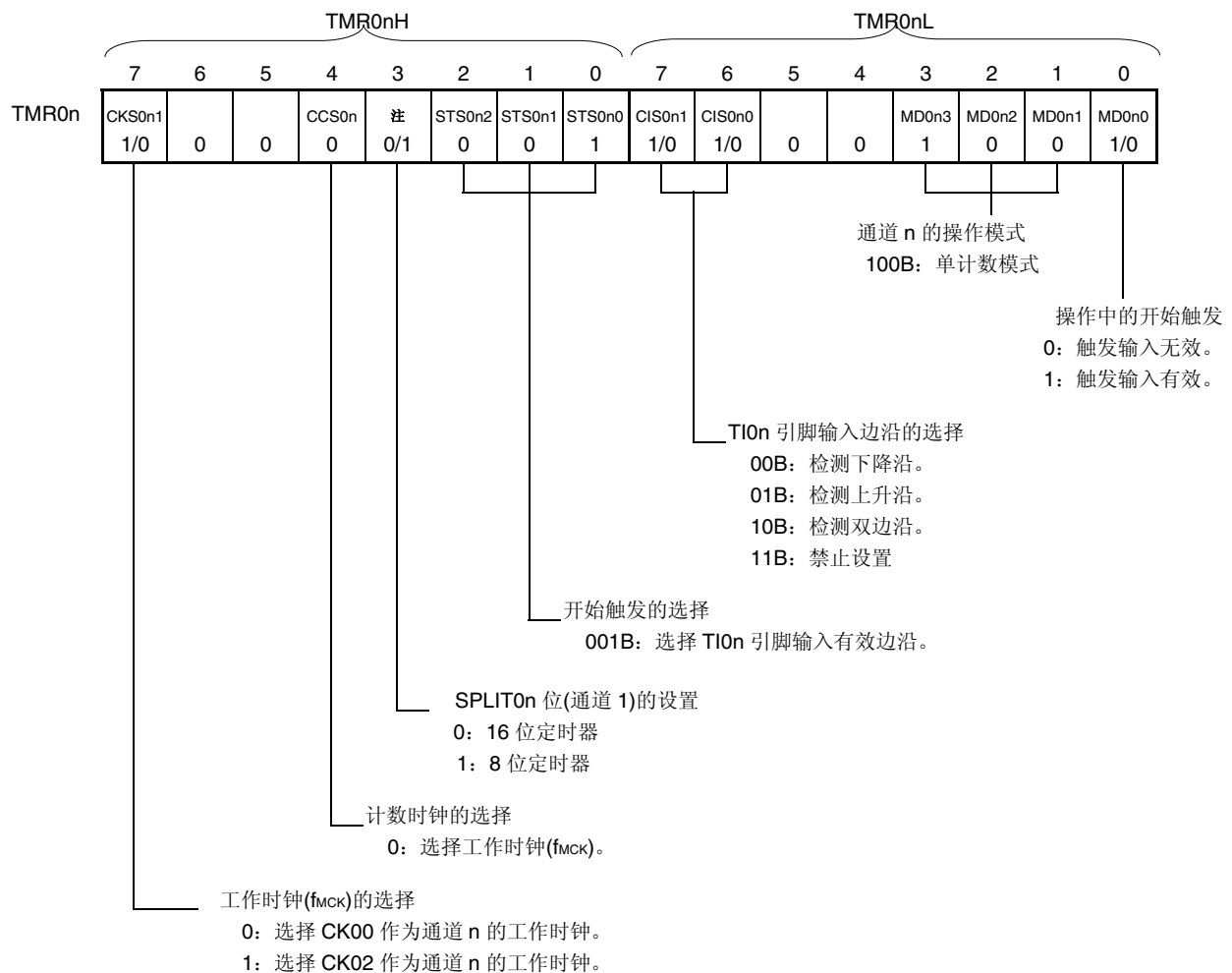


备注 1. n: 通道编号 (n = 0, 1)

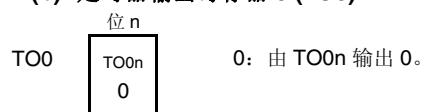
2. TS0n: 定时器通道开始寄存器 0 (TS0) 的位 n
- TE0n: 定时器通道允许状态寄存器 0 (TE0) 的位 n
- TIO0n: TIO0n 引脚输入信号
- TCR0n: 定时器计数寄存器 0n (TCR0n)
- TDR0n: 定时器数据寄存器 0n (TDR0n)

图 6-60. 使用延迟计数器功能时的寄存器设置内容示例 (1/2)

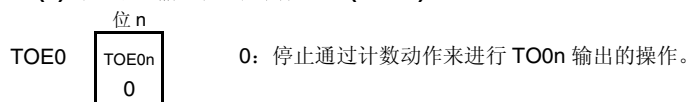
(a) 定时器模式寄存器 0n (TMR0nH, TMR0nL)



(b) 定时器输出寄存器 0 (TO0)



(c) 定时器输出允许寄存器 0 (TOE0)



注 TMR01: SPLIT01 位

TMR00: 固定为 0

备注 n: 通道编号 (n = 0, 1)

图 6-60. 使用延迟计数器功能时的寄存器设置内容示例 (2/2)

(d) 定时器输出电平寄存器 0 (TOL0)

TOL0

位 n
TOL0n
0

 0: 当主通道输出模式 (TOM0n = 0) 时设置为 0。

(e) 定时器输出模式寄存器 0 (TOM0)

TOM0

位 n
TOM0n
0

 0: 设置主通道输出模式。

备注 n: 通道编号 (n = 0, 1)

图 6-61. 使用延迟计数器功能时的操作步骤

	软件操作	硬件状态
TAU初始设置		断电状态 (时钟供应被停止, 向各寄存器的写操作被禁止。)
	将外围允许寄存器0 (PER0)的TAU0EN位设置为1 (TAU0EN位为0时, 禁止读取/写入操作)。 —————▶	上电状态。每个通道处于停止操作状态。 (开始时钟供应, 允许写入各寄存器。)
	设置定时器时钟选择寄存器0(TPS0)。 决定CK00和CK01的时钟频率。	
通道初始设置	设置定时器模式寄存器0n (TMR0n) (决定通道的操作模式)。 将 INTTM0n 输出延迟设置到定时器数据寄存器0n (TDR0n)。 设置噪声滤波器允许寄存器1 (NFEN1)。 将TOE0n位设置为0, 并且停止操作TO0n。	通道处于停止操作状态。 (供应时钟, 并产生一定的功耗。)
开始操作	将TS0n位设置为1。 —————▶ TS0n位是触发位, 将自动返回为0。	TE0n = 1, 变为TI0n引脚输入有效边沿检测等待状态。
	测TI0n引脚输入有效边沿。 —————▶	将TDR0n 寄存器的值载入定时器计数寄存器 0n (TCR0n)。
操作期间	可以任意更改 TDR0n寄存器的设置值。 可以任意读取TCR0n寄存器。 不使用TSR0n寄存器。	计数器(TCR0n)递减计数。当TCR0n递减计数至0000H时, 输出INTTM0n并停止计数 (使TCR0n保持FFFFH) 直到下一个TI0n引脚输入为止。之后重复以上操作。
停止操作	将TT0n位设置为1。 —————▶ TT0n位是触发位, 将自动返回为0。	TE0n = 0, 并且停止计数操作。 TCR0n寄存器保持计数值并停止计数操作。
TAU停止	将PER0寄存器的TAU0EN位设置为0。 —————▶	断电状态 初始化所有电路, 同时初始化各通道的SFR。

恢复操作。

备注 n: 通道编号 (n = 0, 1)

6.8 定时器阵列单元的多通道联动操作功能

6.8.1 作为单触发脉冲输出功能操作

通过联合使用两个通道，可以从输入 TI0n 引脚的信号产生一个任意延迟脉冲宽度的单触发脉冲。
延迟时间和脉冲宽度可用下述表达式计算。

$\begin{aligned}\text{延迟时间} &= \{\text{TDR0n (主) 的设置值} + 2\} \times \text{计数时钟周期} \\ \text{脉冲宽度} &= \{\text{TDR0p (从属) 的设置值}\} \times \text{计数时钟周期}\end{aligned}$

主通道工作于单计数模式，并对延迟计数。主通道的定时器计数寄存器 0n (TCR0n) 在检测到开始触发信号时开始工作，并载入定时器数据寄存器 0n (TDR0n) 的值。

TCR0n 寄存器与计数时钟同步，从载入的 TDR0n 寄存器的值开始递减计数。当 TCR0n = 0000H 时，输出 INTTM0n 中断并停止计数，直至检测到下一个开始触发为止。

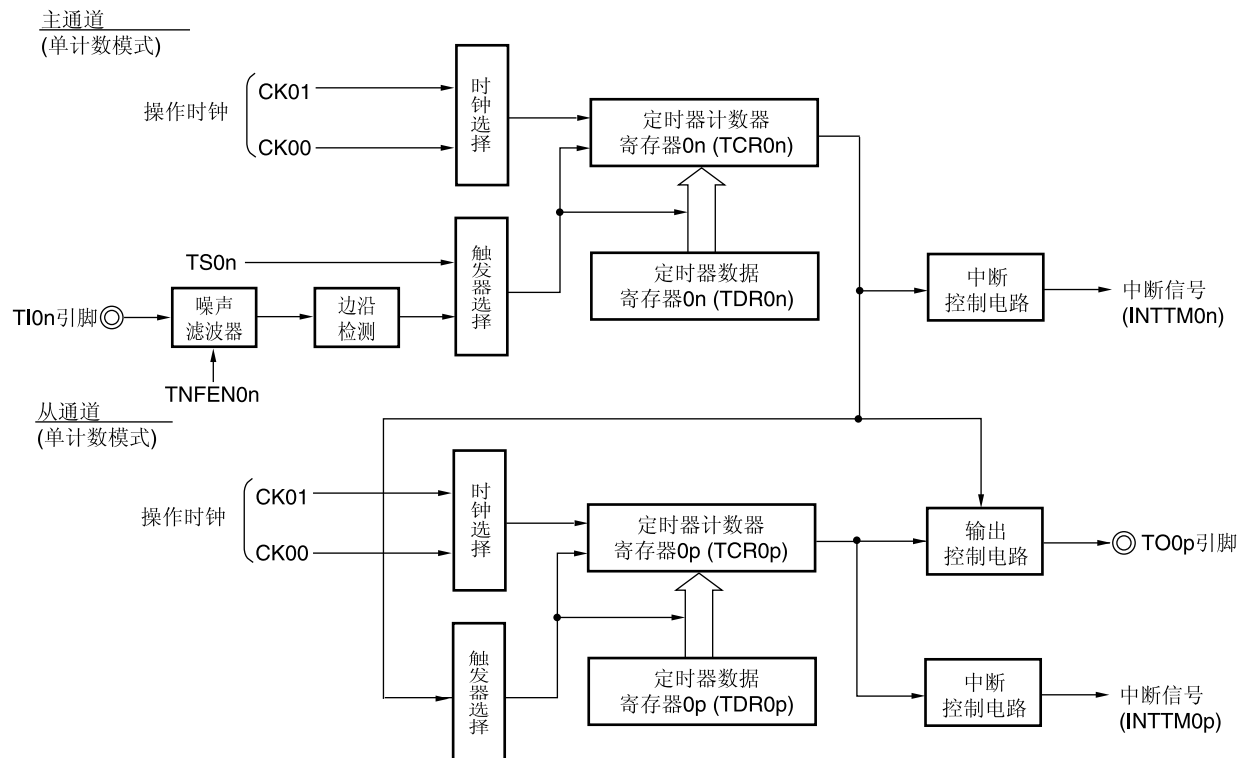
从属通道工作于单计数模式，并对脉冲宽度计数。将主通道的 INTTM0n 作为开始触发，从属通道的 TCR0p 寄存器开始工作，并载入 TDR0p 寄存器的值。TCR0p 寄存器与计数时钟同步，从载入的 TDR0p 寄存器值开始递减计数。当计数值 = 0000H 时，将输出 INTTM0p 中断并停止计数，直至检测到下一个开始触发（主通道的 INTTM0n）为止。从主通道产生 INTTM0n 并经过一个计数时钟后 TO0p 的输出电平成为有效电平，当 TCR0p = 0000H 时成为无效电平。

也可以不使用 TI0n 引脚输入，用软件操作 (TS0n = 1) 作为开始触发来输出单触发脉冲。

注意事项 主通道定时器数据寄存器 0n (TDR0n) 与从属通道 TDR0p 寄存器的载入时序不同。如果在操作过程中改写 TDR0n 和 TDR0p 寄存器，则会输出非法波形。请在产生 INTTM0n 后改写 TDR0n 寄存器，请在产生 INTTM0p 后改写 TDR0p 寄存器。

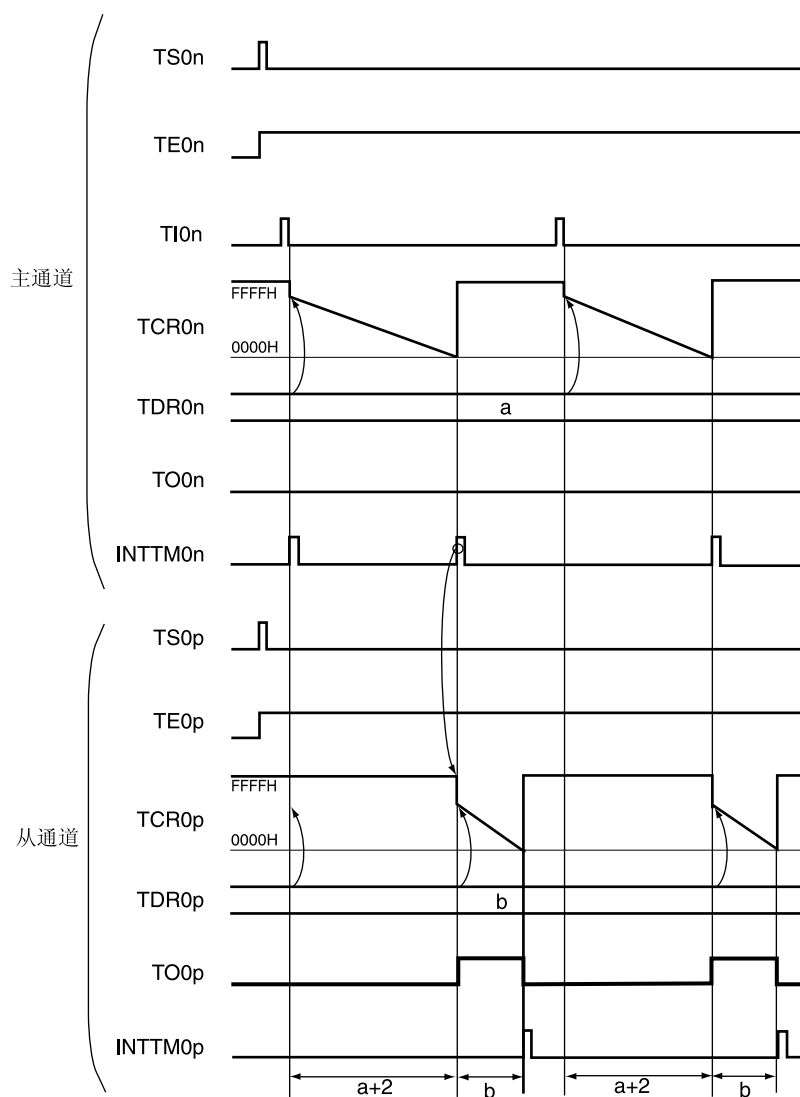
备注 n: 主通道编号 (n = 0)
p: 从属通道编号 (p = 1)

图 6-62. 作为单触发脉冲输出功能的操作框图



备注 n: 主通道编号 (n = 0)
p: 从属通道编号 (p = 1)

图 6-63. 作为单触发脉冲输出功能的操作基本时序示例



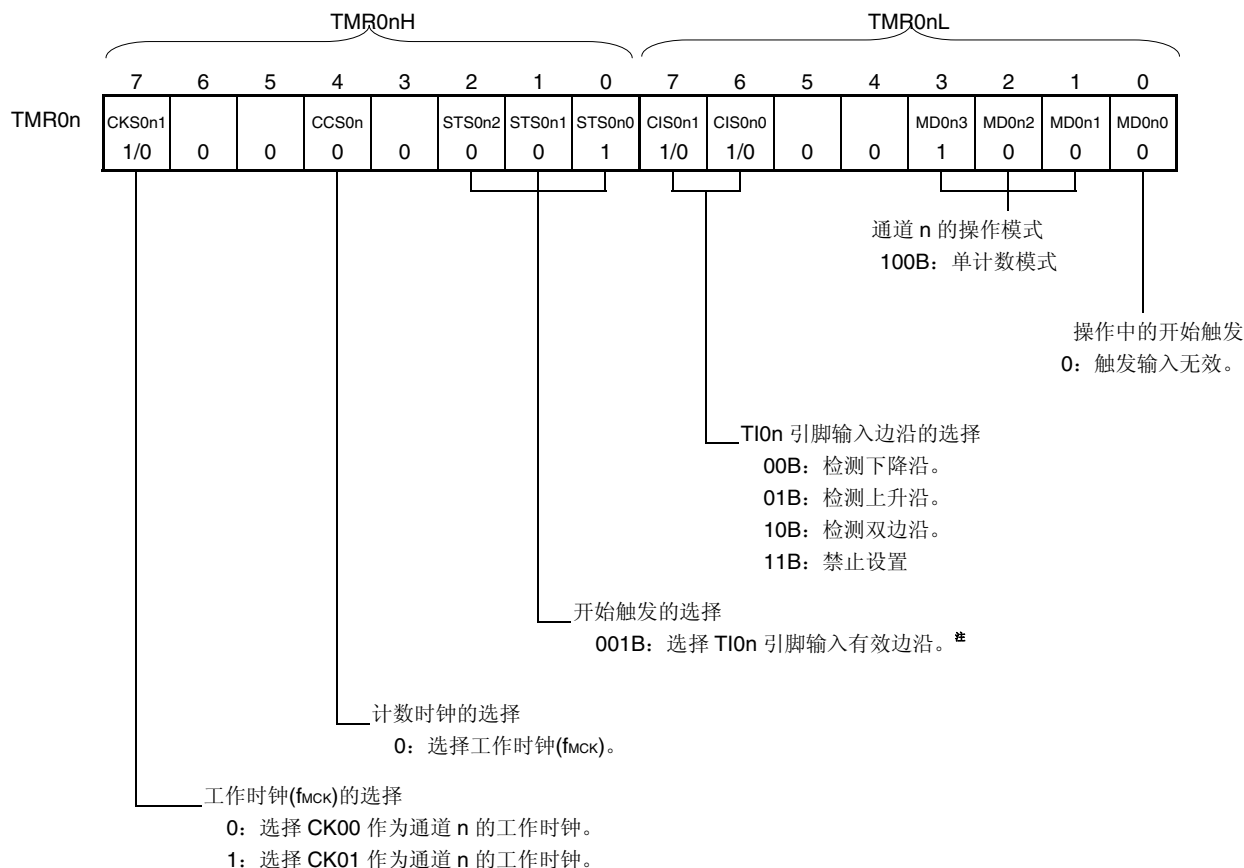
备注 1. n: 主通道编号 ($n = 0$)

p: 从属通道编号 ($p = 1$)

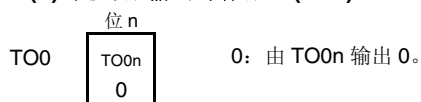
2. TS0n, TS0p: 定时器通道开始寄存器 0 (TS0) 的位 n、p
- TE0n, TE0p: 定时器通道允许状态寄存器 0 (TE0) 的位 n、p
- TI0n, TI0p: TI0n 和 TI0p 引脚输入信号
- TCR0n, TCR0p: 定时器计数寄存器 0n、0p (TCR0n、TCR0p)
- TDR0n, TDR0p: 定时器数据寄存器 0n、0p (TDR0n、TDR0p)
- TO0n, TO0p: TO0n 和 TO0p 引脚输出信号

图 6-64. 使用单触发脉冲输出功能(主通道)时的寄存器设置内容示例

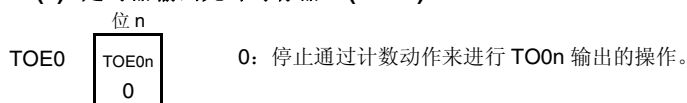
(a) 定时器模式寄存器 0n (TMR0nH, TMR0nL)



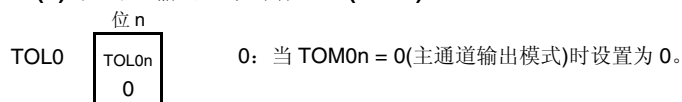
(b) 定时器输出寄存器 0 (TO0)



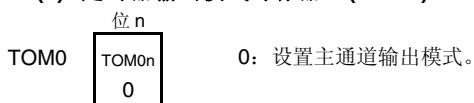
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



(e) 定时器输出模式寄存器 0 (TOM0)

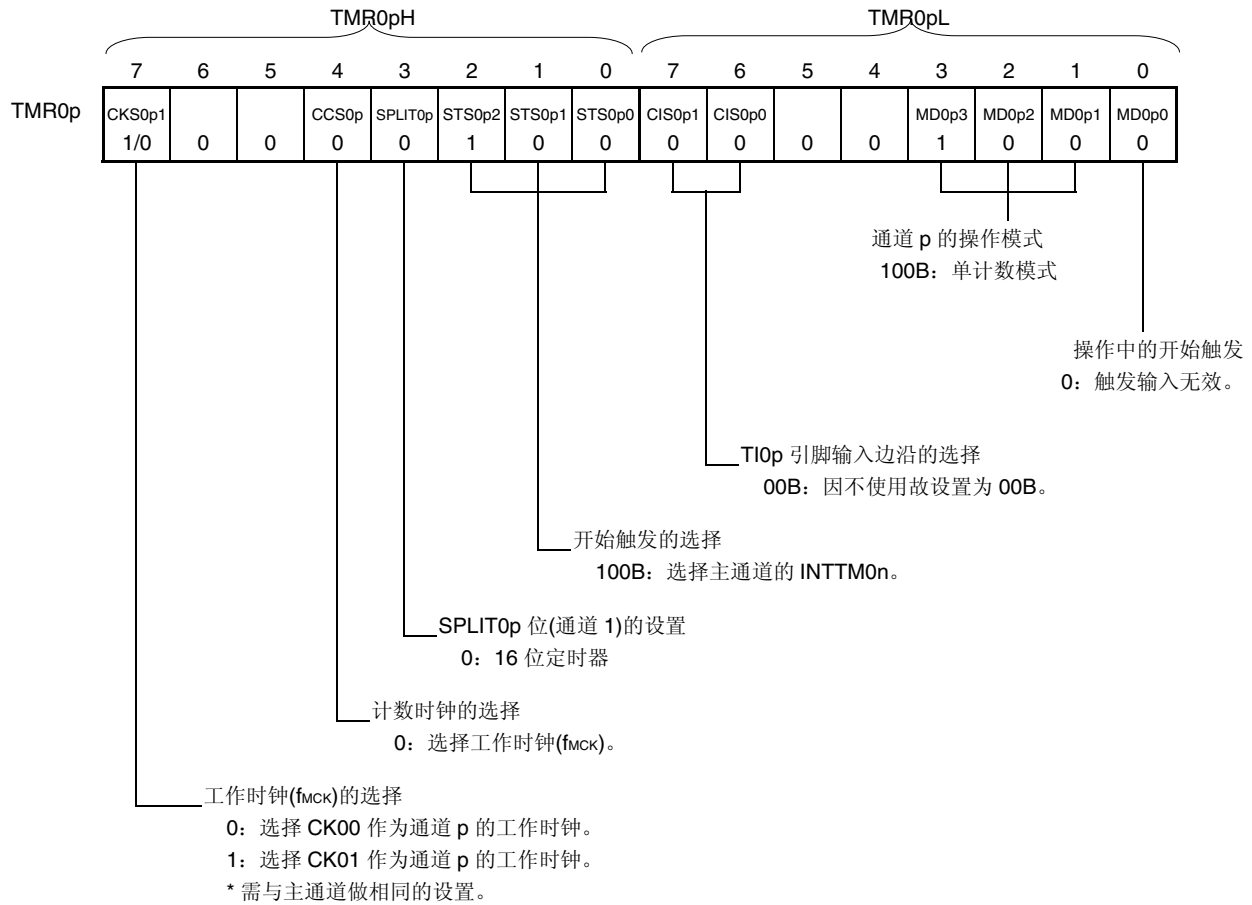


注 也可以不使用 TI0n 引脚输入, 用软件操作 (TS0n = 1)作为开始触发来输出单触发脉冲。

备注 n: 主通道编号 (n = 0)

图 6-65. 使用单触发脉冲输出功能(从属通道)时的寄存器设置内容示例

(a) 定时器模式寄存器 0p (TMR0pH, TMR0pL)



(b) 定时器输出寄存器 0 (TO0)

位 p	
TO0	TO0p
	1/0

0: 由 TO0p 输出 0。
 1: 由 TO0p 输出 1。

(c) 定时器输出允许寄存器 0 (TOE0)

位 p	
TOE0	TOE0p
	1/0

0: 停止通过计数动作来进行 TO0p 输出的操作。
 1: 允许通过计数动作来进行 TO0p 输出的操作。

(d) 定时器输出电平寄存器 0 (TOL0)

位 p	
TOL0	TOL0p
	1/0

0: 正逻辑输出 (高电平有效)
 1: 负逻辑输出 (低电平有效)

(e) 定时器输出模式寄存器 0 (TOM0)

位 p	
TOM0	TOM0p
	1

1: 设置从属通道输出模式。

注 TMR01: SPLIT01 位

TMR00: 固定为 0

备注 n: 主通道编号 (n = 0)

p: 从属通道编号 (p = 1)

图 6-66. 单触发脉冲输出功能的操作步骤 (1/2)

	软件操作	硬件状态
TAU初始设置		断电状态 (时钟供应被停止, 向各寄存器的写操作被禁止。)
	将外围允许寄存器0 (PER0)的TAU0EN位设置为1 (TAU0EN位为0时, 禁止读取/写入操作)。—————→	上电状态。每个通道处于停止操作状态。 (开始时钟供应, 允许写入各寄存器。)
	设置定时器时钟选择寄存器0(TPS0)。 决定CK00和CK01的时钟频率。	
通道初始设置	设置即将使用的两个通道的定时器模式寄存器0n, mp (TMR0n, TMR0p) (决定通道的操作模式)。 向主通道定时器数据寄存器0n (TDR0n) 设置输出延迟, 并向从属通道TDR0p寄存器设置脉冲宽度。 设置主通道的噪声滤波器允许寄存器1 (NFEN1)。	通道处于停止操作状态。 (供应时钟, 并产生一定的功耗。)
	设置从属通道。 设置定时器输出模式寄存器0(TOM0)的TOM0p位设置 为1(从属通道输出模式)。 设置TOL0p位。 设置 TO0p位并决定 TO0p 输出的初始电平。—————→	TO0p引脚处于Hi-Z输出状态。
	将TOE0p位设置为1, 并允许操作TO0p。—————→ 将端口寄存器和端口模式寄存器设置为0。—————→	当端口模式寄存器处于输出模式且端口寄存器为0时, 输出 TO0p的初始设置电平。 因为操作通道处于停止状态, 所以TO0p不会变化。 TO0p引脚输出TO0p设置电平。

备注 n: 主通道编号 (n = 0)

p: 从属通道编号 (p = 1)

图 6-66. 单触发脉冲输出功能的操作步骤 (2/2)

	软件操作	硬件状态
恢复操作。	开始操作	将TOE0p位（从属）设置为1(仅限恢复定时器操作时)。同时将定时器通道开始寄存器0 (TS0)的TS0n（主）和TS0p（从属）位设置为1。——→ TE0n和TE0p位被设置为1，主通道进入TI0n 输入边沿检测等待状态。计数器停止操作。 TS0n和TS0p位是触发位，将自动返回为0。 检测主通道的TI0n引脚输入有效边沿。——→ 主通道开始计数。
	操作期间	仅可以更改TMR0n寄存器的CIS0n1和CIS0n0位的设置值。 不能更改 TMR0p、TDR0n、TDR0p 寄存器、TOM0n、TOM0p、TOL0n 和 TOL0p 位的设置值。 可以任意读取TCR0n和TCR0p寄存器。 不使用 TSR0n和TSR0p寄存器。 可以更改从属通道的TO0n 和TOE0n 寄存器的设置值。 主通道在检测到 TI0n 引脚有效输入边沿时，将 TDR0n 寄存器的值载入定时器计数寄存器 0n (TCR0n)，计数器开始递减计数。当计数值达到TCRmn = 0000H时，输出INTTM0n中断并停止计数，直至检测到下一个TI0n引脚输入有效边沿为止。 从属通道（由主通道的INTTM0n 触发）将 TDR0p寄存器的值载入 TCR0p寄存器，计数器开始递减计数。TO0p的输出电平在从主通道产生INTTM0n并经过一个计数时钟之后成为有效电平。当TCR0p = 0000H时成为无效电平，停止计数操作。 之后重复以上操作。
	停止操作	同时将TT0n（主）和TT0p（从属）位设置为 1。TT0n 和 TT0p位是触发位，将自动返回为0。——→ TE0n, TE0p = 0，并且停止计数操作。TCR0n和TCR0p寄存器保持计数值并停止计数操作。TO0p输出不被初始化，而是保持当前状态。 将从属通道的TOE0p位设置为0，并对TO0p位设值。——→ TO0p引脚输出TO0p设置电平。
	TAU 停止	保持TO0p引脚输出电平时 将希望保持的值设置到端口寄存器后，将TO0p位清除为0。——→ TO0p引脚输出电平由端口功能保持。 不需要保持TO0p引脚输出电平时 不要求设置。 将PER0寄存器的TAU0EN位设置为0。——→ 断电状态 初始化所有电路，同时初始化各通道的SFR。 (TO0p位清除为0，TO0p引脚被设置为端口模式。)

备注 n: 主通道编号 (n = 0)

p: 从属通道编号 (p = 1)

6.8.2 作为 PWM 功能的操作

可以使用两个通道作为一组，来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用下述表达式计算。

$$\begin{aligned} \text{脉冲周期} &= \{\text{TDR0n (主) 的设置值} + 1\} \times \text{计数时钟周期} \\ \text{占空比 [\%]} &= \{\text{TDR0p (从属) 的设置值}\} / \{\text{TDR0n (主) 的设置值} + 1\} \times 100 \\ \text{0\% 输出: } &\text{TDR0p (从属) 的设置值} = 0000\text{H} \\ \text{100\% 输出: } &\text{TDR0p (从属) 的设置值} \geq \{\text{TDR0n (主) 的设置值} + 1\} \end{aligned}$$

备注 如果 TDR0p (从属) 的设置值 > (TDR0n (主) 的设置值 + 1)，则占空比将超过 100%，而合为 100% 输出。

主通道工作于间隔定时器模式。如果定时器通道开始寄存器 0 (TS0) 的通道开始触发位 (TS0n) 被设置为 1，则输出中断 (INTTM0n)，定时器数据寄存器 0n (TDR0n) 的设置值被载入定时器计数寄存器 0n (TCR0n)，计数器与计数时钟同步并递减计数。当计数器达到 0000H 时，输出 INTTM0n 中断，再次将 TDR0n 寄存器的值载入 TCR0n 寄存器，计数器递减计数。重复该操作直到定时器通道停止寄存器 0 (TT0) 的通道停止触发位 (TT0n) 被设置为 1 为止。

如果用两个通道来输出一个 PWM 波形，直到主通道递减计数至 0000H 为止的周期为 PWM 输出 (TO0p) 周期。

从属通道工作于单计数模式。通过将主通道的 INTTM0n 中断用作开始触发，TCR0p 寄存器载入 TDR0p 寄存器的值，计数器递减计数至 0000H。当计数器达到 0000H 时，输出 INTTM0p 中断并等待直到产生下一个开始触发 (主通道的 INTTM0n) 为止。

如果用两个通道来输出一个 PWM 波形，直到从属通道递减计数至 0000H 为止的期间为 PWM 输出 (TO0p) 占空。

PWM 输出 (TO0p) 在主通道产生 INTTM0n 中断并经过一个时钟之后成为有效电平，当从属通道的 TCR0p 寄存器变成 0000H 时成为无效电平。

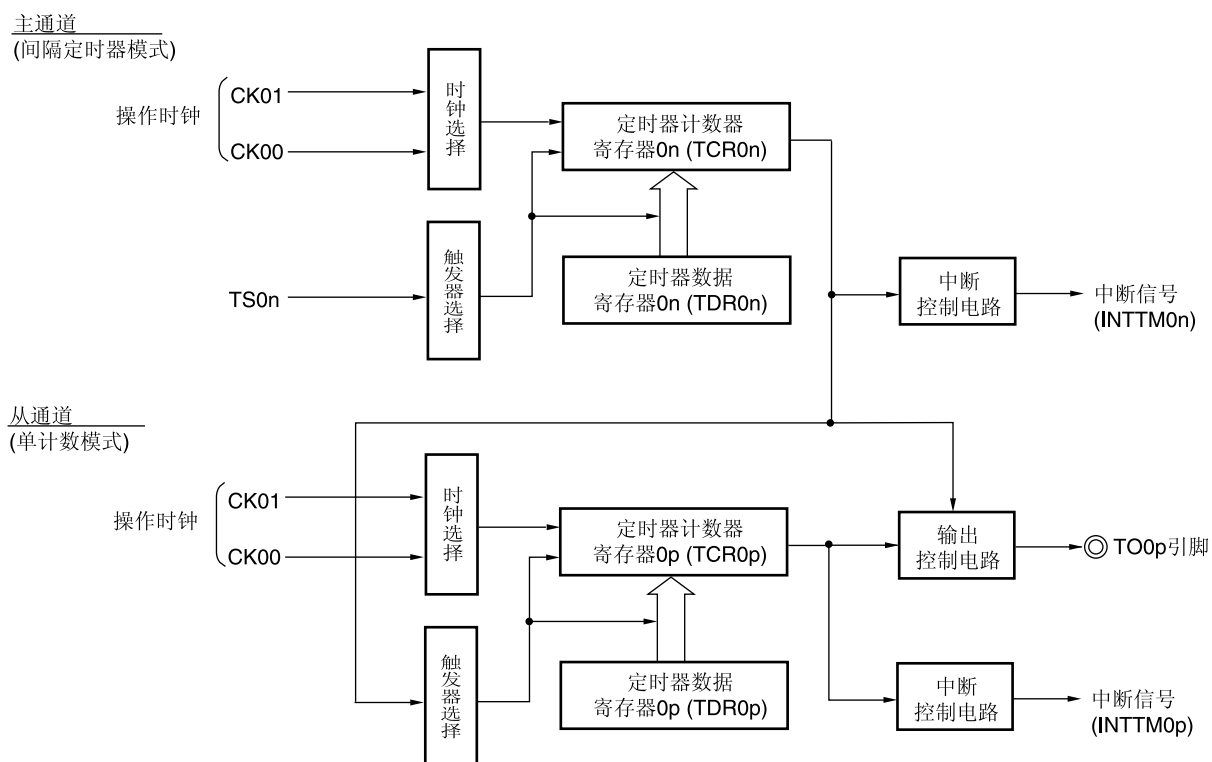
注意事项 1. 要同时改写主通道定时器数据寄存器 0n (TDR0nH, TDR0nL) 和从属通道的 TDR0pH、TDR0pL 寄存器，则需执行至少四次写存取。将 TDR0nH、TDR0nL、TDR0pH 和 TDR0pL 寄存器的值载入至 TCR0nH、TCR0nL、TCR0pH 和 TCR0pL 寄存器的时序为主通道 INTTM0n 发生的时候。因此，在主通道 INTTM0n 发生前和发生后分开执行改写时，TO0p 引脚将不能输出预期波形。将主的 TDR0nH 和 TDR0nL 寄存器和从属的 TDR0pH 和 TDR0pL 寄存器都改写时，必须要在主通道产生 INTTM0n 之后立即改写这四个寄存器。

2. 在 8 位定时器模式下使用 PWM 功能时，请将 TDR0n 寄存器 (主) 的高 8 位 (TDR0nH) 设置为 00H。

备注 n: 主通道编号 (n = 0)

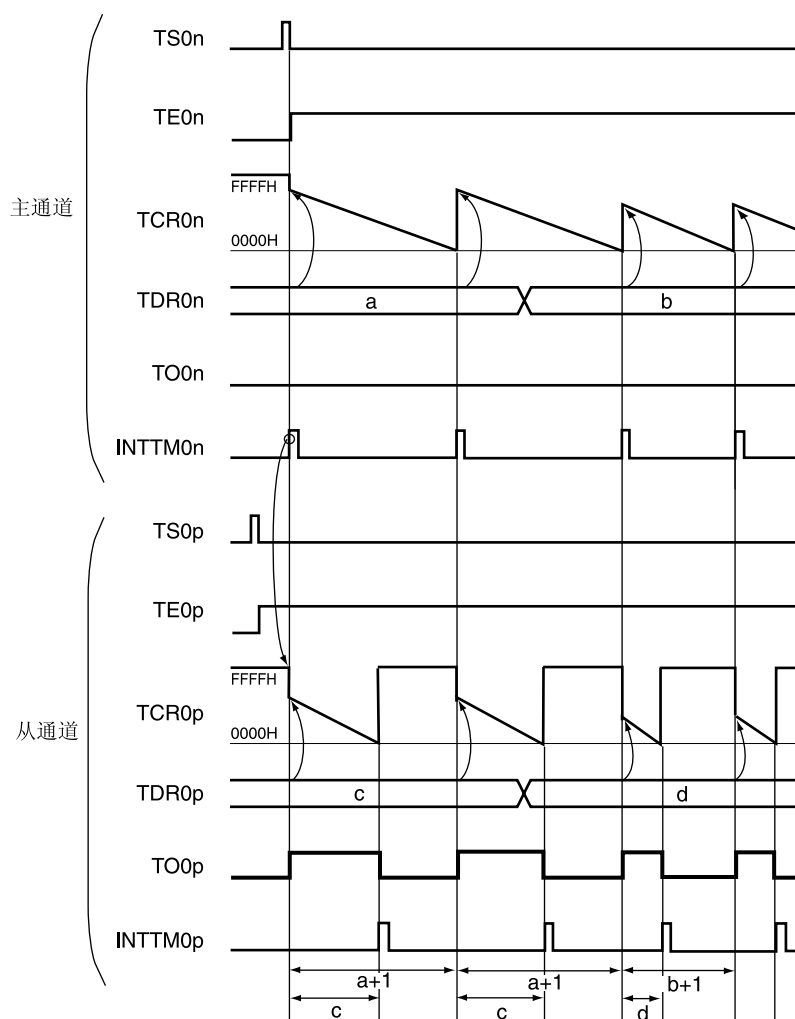
p: 从属通道编号 (p = 1)

图 6-67. 作为 PWM 功能的操作框图



备注 n: 主通道编号 (n = 0)
p: 从属通道编号 (p = 1)

图 6-68. 作为 PWM 功能的操作基本时序示例



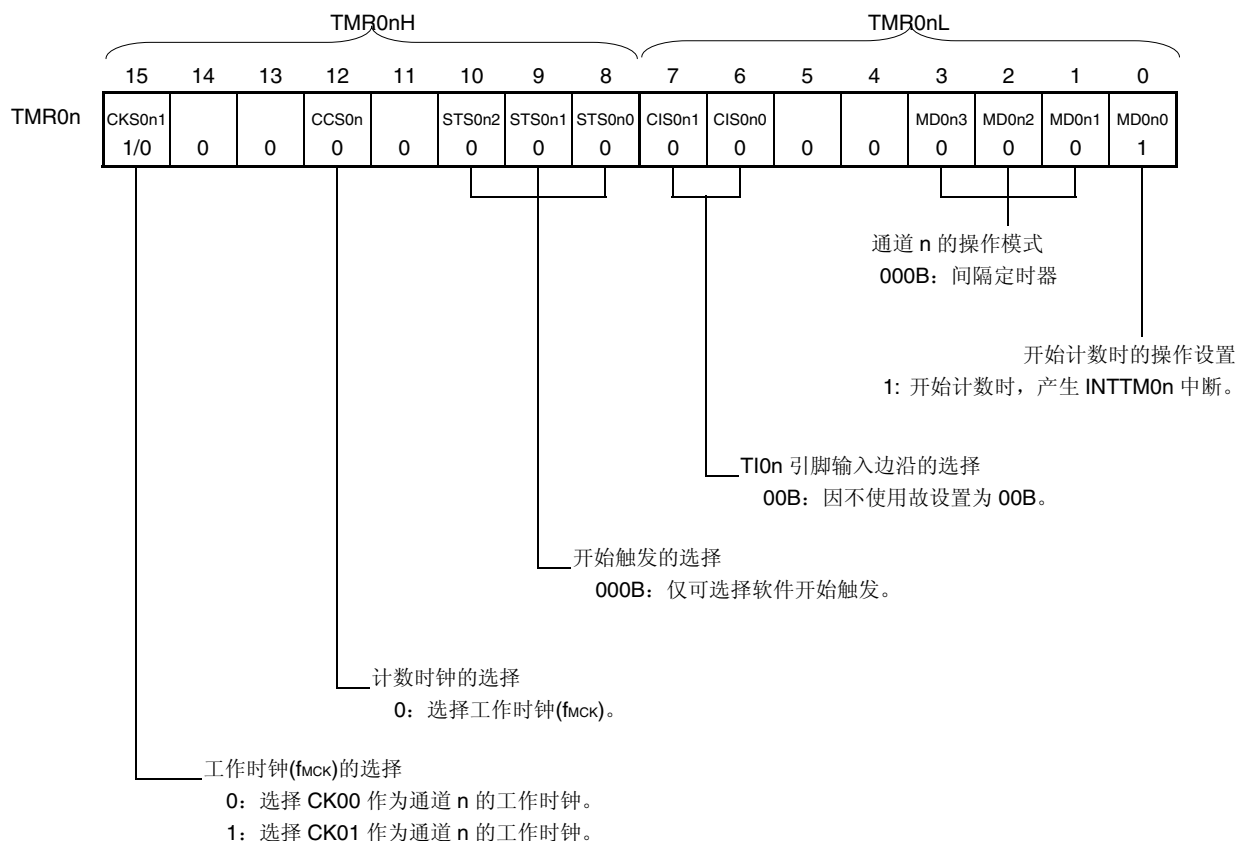
备注1. n: 主通道编号 ($n = 0$)

p: 从属通道编号 ($p = 1$)

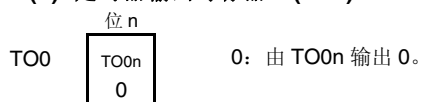
2. TS0n, TS0p: 定时器通道开始寄存器 0 (TS0) 的位 n、p
- TE0n, TE0p: 定时器通道允许状态寄存器 0 (TE0) 的位 n、p
- TCR0n, TCR0p: 定时器计数寄存器 0n、0p (TCR0n、TCR0p)
- TDR0n, TDR0p: 定时器数据寄存器 0n、0p (TDR0n、TDR0p)
- TO0n, TO0p: TO0n 和 TO0p 引脚输出信号

图 6-69. 使用 PWM 功能（主通道）时的寄存器设置内容示例

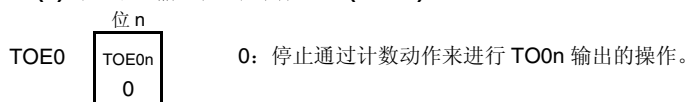
(a) 定时器模式寄存器 0n (TMR0nH, TMR0nL)



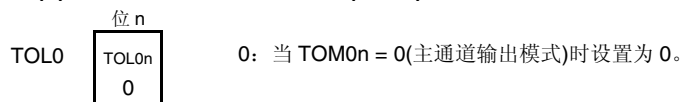
(b) 定时器输出寄存器 0 (TO0)



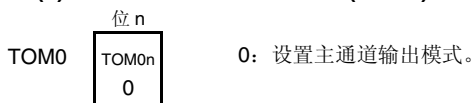
(c) 定时器输出允许寄存器 0 (TOE0)



(d) 定时器输出电平寄存器 0 (TOL0)



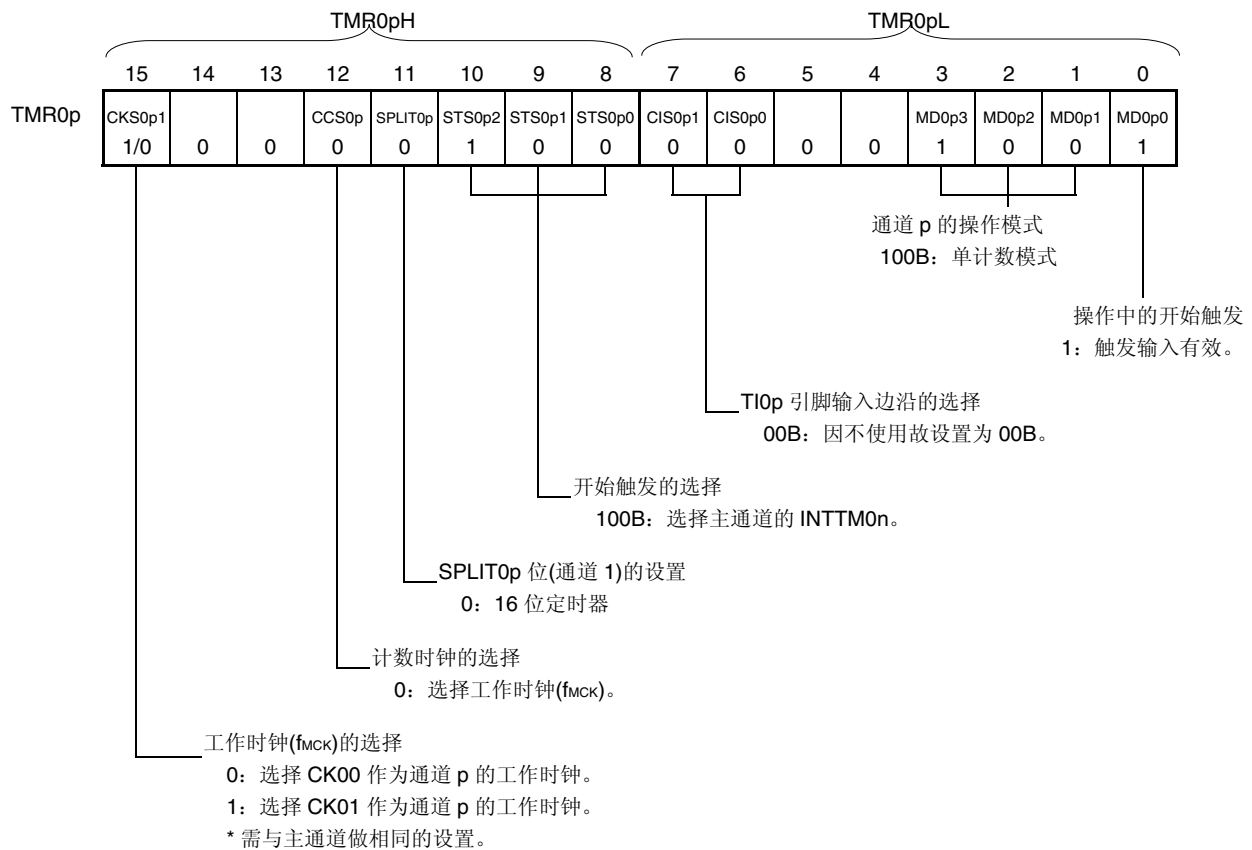
(e) 定时器输出模式寄存器 0 (TOM0)



备注 n: 主通道编号 (n = 0)

图 6-70. 使用 PWM 功能（从属通道）时的寄存器设置内容示例

(a) 定时器模式寄存器 0p (TMR0pH, TMR0pL)



(b) 定时器输出寄存器 0 (TO0)

位 p	
TO0p	0: 由 TO0p 输出 0。
1/0	1: 由 TO0p 输出 1。

(c) 定时器输出允许寄存器 0 (TOE0)

位 p	
TOE0p	0: 停止通过计数动作来进行 TO0p 输出的操作。
1/0	1: 允许通过计数动作来进行 TO0p 输出的操作。

(d) 定时器输出电平寄存器 0 (TOL0)

位 p	
TOL0p	0: 正逻辑输出（高电平有效）
1/0	1: 负逻辑输出（低电平有效）

(e) 定时器输出模式寄存器 0 (TOM0)

位 p	
TOM0p	1: 设置从属通道输出模式。
1	

备注 n: 主通道编号 (n = 0)
p: 从属通道编号 (p = 1)

图 6-71. 使用 PWM 功能时的操作步骤 (1/2)

	软件操作	硬件状态
TAU初始 设置		断电状态 (时钟供应被停止, 向各寄存器的写操作被禁止。)
	将外围允许寄存器0 (PER0)的TAU0EN位设置为1 (TAU0EN位为0时, 禁止读取/写入操作)。—————▶	上电状态。每个通道处于停止操作状态。 (开始时钟供应, 允许写入各寄存器。)
	设置定时器时钟选择寄存器0(TPS0)。 决定CK00和CK01的时钟频率。	
通道初始 设置	设置即将使用的两个通道的定时器模式寄存器0n, 0p (TMR0n, TMR0p) (决定通道的操作模式)。 向主通道的定时器数据寄存器0n (TDR0n) 设置间隔 (周 期) 值, 并向从属通道的TDR0p寄存器设置占空比。	通道处于停止操作状态。 (供应时钟, 并产生一定的功耗。)
	设置从属通道。 设置定时器输出模式寄存器0(TOM0)的TOM0p位设置为 1(从属通道输出模式)。	TO0p引脚处于Hi-Z输出状态。
	设置TOL0p位。 设置 TO0p位并决定 TO0p 输出的初始电平。—————▶	当端口模式寄存器处于输出模式且端口寄存器为0时, 输出 TO0p的初始设置电平。
	将TOE0p位设置为1, 并允许操作TO0p。—————▶	因为操作通道处于停止状态, 所以TO0p不会变化。
	将端口寄存器和端口模式寄存器设置为0。—————▶	TO0p引脚输出TO0p设置电平。

备注 n: 主通道编号 (n = 0)

p: 从属通道编号 (p = 1)

图 6-71. 使用 PWM 功能时的操作步骤 (2/2)

	软件操作	硬件状态
恢复操作。	开始操作 将TOE0p位（从属）设置为1(仅限恢复定时器操作时)。同时将定时器通道开始寄存器0 (TS0)的TS0n（主）和TS0p（从属）位设置为1。TS0n和TS0p位是触发位，将自动返回为0。	TE0n = 1, TE0p = 1 主通道开始计数时，产生INTTM0n中断。通过该中断触发，从属通道也开始计数。
	操作期间 不能更改 TMR0n和TMR0p寄存器、TOM0n、TOM0p、TOL0n和TOL0p位的设置值。 产生主通道INTTM0n之后，可以更改TDR0n和TDR0p寄存器的设置值。 可以任意读取TCR0n和TCR0p寄存器。 不使用 TSR0n和TSR0p寄存器。	主通道的计数器将 TDR0n 寄存器值载入定时器计数寄存器0n (TCR0n)，并递减计数。当计数值达到TCR0n = 0000H时，输出INTTM0n。同时，TDR0n寄存器的值将被载入TCR0n寄存器，计数器再次开始递减计数。 在从属通道，TDR0p寄存器的值载入TCR0p寄存器（由主通道的INTTM0n触发），计数器开始递减计数。TO0p的输出电平在主通道输出INTTM0n并经过一个计数时钟之后成为有效电平。当TCR0p = 0000H时成为无效电平，停止计数操作。 之后重复以上操作。
	停止操作 同时将TT0n（主）和TT0p（从属）位设置为 1。TT0n 和 TT0p位是触发位，将自动返回为0。	TE0n, TE0p = 0，并且停止计数操作。 TCR0n和TCR0p寄存器保持计数值并停止计数操作。 TO0p输出不被初始化，而是保持当前状态。
	将从属通道的TOE0p位设置为0，并对TO0p位设置。	TO0p引脚输出TO0p设置电平。
TAU 停止	保持TO0p引脚输出电平时 将希望保持的值设置到端口寄存器后，将TO0p位清除为0。	TO0p引脚输出电平由端口功能保持。
	不需要保持TO0p引脚输出电平时 不要求设置。 将PER0寄存器的TAU0EN位设置为0。	断电状态 初始化所有电路，同时初始化各通道的SFR。 (TO0p位清除为0，TO0p引脚被设置为端口模式。)

备注 n: 主通道编号 (n = 0)

p: 从属通道编号 (p = 1)

6.9 使用定时器阵列单元时的注意事项

6.9.1 使用定时器输出时的注意事项

在有些产品中，可能为某些引脚指定定时器输出和其他复用功能。这种情况下，必须将其他复用功能的输出设置为其初始状态。

详情请参阅**4.5 使用复用功能时相关寄存器的设置**。

第七章 时钟输出/蜂鸣器输出控制电路

7.1 时钟输出/蜂鸣器输出控制电路的功能

时钟输出控制电路用于输出供应至外围 IC 的时钟。

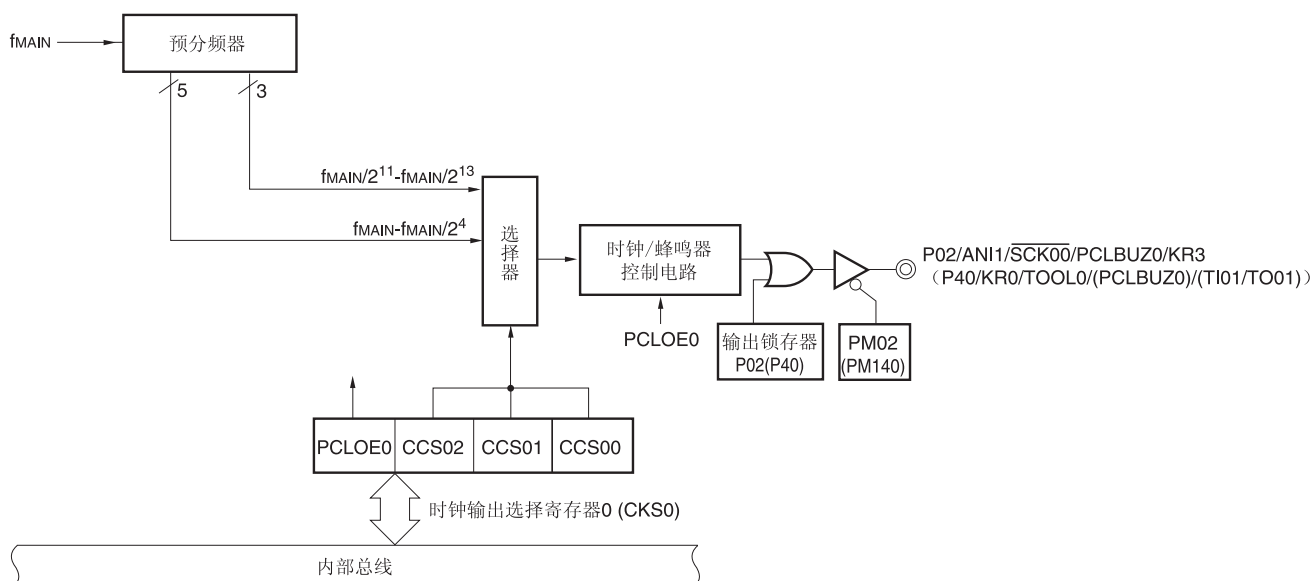
蜂鸣器输出功能用于输出蜂鸣器频率的方波。

该引脚可选择为时钟输出用或蜂鸣器输出用。

PCLBUZ0 引脚用来输出使用时钟输出选择寄存器 0(CKS0)选择的时钟。

时钟输出/蜂鸣器输出控制电路的框图如图 7-1 所示。

图 7-1. 时钟输出/蜂鸣器输出控制电路的框图



注意事项 关于可从 PCLBUZ0 引脚输出的频率，请参阅 21.4 AC 特性。

备注 可以通过设置外围输入/输出重定向寄存器 0 (PIOR0)来指定上表括号内的功能。

7.2 时钟输出/蜂鸣器输出控制电路的配置

时钟输出/蜂鸣器输出控制电路包含以下硬件。

表 7-1. 时钟输出/蜂鸣器输出控制电路的配置

项目	配置
控制寄存器	时钟输出选择寄存器 0 (CKS0) 端口模式寄存器 0 (PM0) [端口模式寄存器 4 (PM4)] 端口寄存器 0 (P0) [端口寄存器 4 (P4)]

备注 可以通过设置外围输入/输出重定向寄存器(PIOR)来指定上表括号内的功能。

7.3 控制时钟输出/蜂鸣器输出控制电路的寄存器

以下两个寄存器用于控制时钟输出/蜂鸣器输出控制电路。

- 时钟输出选择寄存器 0 (CKS0)
- 端口模式寄存器 0 (PM0) [端口模式寄存器 4 (PM4)]

备注 可以通过设置外围输入/输出重定向寄存器 0 (PIOR0)来指定括号内的功能。

7.3.1 时钟输出选择寄存器 0 (CKS0)

该寄存器设置时钟输出或蜂鸣器频率输出引脚(PCLBUZ0)的输出允许/禁止，并设置输出时钟。

使用 1 位或 8 位存储器操作指令设置 CKS0 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 7-2. 时钟输出选择寄存器 0 (CKS0)的格式

地址: FFFA5H 复位后: 00H R/W

符号

<7> 6 5 4 3 2 1 0

CKS0

PCLOE0	0	0	0	0	CCS02	CCS01	CCS00

PCLOE0	PCLBUZ0 引脚的输出允许/禁止的指定
0	禁止输出 (默认)
1	允许输出

CCS02	CCS01	CCS00		PCLBUZ0 引脚输出时钟选择				
				f _{MAIN} (MHz)				
				1.25	2.5	5	10	20
0	0	0	f _{MAIN}	1.25 MHz	2.5 MHz	5 MHz [※]	10 MHz [※]	禁止设置 [※]
0	0	1	f _{MAIN} /2	625 kHz	1.25 MHz	2.5 MHz	5 MHz [※]	10 MHz [※]
0	1	0	f _{MAIN} /2 ²	312.5 kHz	625 kHz	1.25 MHz	2.5 MHz	5 MHz [※]
0	1	1	f _{MAIN} /2 ³	156.3 kHz	312.5 kHz	625 kHz	1.25 MHz	2.5 MHz
1	0	0	f _{MAIN} /2 ⁴	78.1 kHz	156.3 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	1	f _{MAIN} /2 ¹¹	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	0	f _{MAIN} /2 ¹²	305 Hz	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	1	f _{MAIN} /2 ¹³	153 Hz	305 Hz	610 Hz	1.22 kHz	2.44 kHz

注 可用的输出时钟取决于工作电压范围。详情请参阅 21.4 AC 特性。

注意事项 1. 将 PCLBUZ0 引脚设置为禁止输出(PCLOE0 = 0)后, 才可以更改输出时钟。

2. 若要切换至 STOP 模式, 将 PCLBUZ0 引脚设置为禁止输出(PCLOE0 = 0)后, 至少等待 PCLBUZ0 引脚输出时钟的 1.5 个时钟后, 再执行 STOP 指令。

备注 f_{MAIN}: 主系统时钟频率

7.3.2 端口模式寄存器 0, 4 (PM0, PM4)

这些寄存器以 1 位为单位设置端口 0 和 4 的输入/输出。

当使用 P02/ANI1/ $\overline{\text{SCK00}}$ /PCLBUZ0/KR3 (P40/KR0/TOOL0/(PCLBUZ0)/(TI01/TO01))引脚作为时钟输出和蜂鸣器输出时, 将 PM02 (PM40)位和 P02 (P40)位的输出锁存器清除为 0。并将端口模式控制寄存器 0 的 PMC02 位设置为 0。

使用 1 位或 8 位存储器操作指令设置 PM0 (PM4)寄存器。

产生复位信号后, 该寄存器被设置为 FFH。

图 7-3. 端口模式寄存器 0, 4 (PM0, PM4)的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
	PM02	P02 引脚输入/输出模式的选择									
	0	输出模式 (输出缓冲器启用)									
	1	输入模式 (输出缓冲器关闭)									
符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W
	PM40	P40 引脚输入/输出模式的选择									
	0	输出模式 (输出缓冲器启用)									
	1	输入模式 (输出缓冲器关闭)									

备注 当 PIOR 寄存器的 PIOR0 位为 1 时, 括号中的内容适用。

7.4 时钟输出/蜂鸣器输出控制电路的操作

该引脚可选择为时钟输出用或蜂鸣器输出用。

PCLBUZ0 引脚用来输出使用时钟输出选择寄存器 0(CKS0)选择的时钟/蜂鸣器频率。

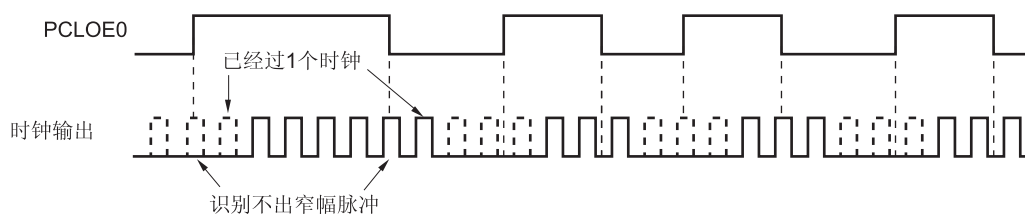
7.4.1 作为输出引脚的操作

PCLBUZ0 引脚按照以下步骤输出。

- <1> 将 PCLBUZ0 引脚复用端口的端口模式寄存器(PMxx)、端口寄存器(Pxx)以及端口模式控制寄存器(PMCxx)的对应位设置为 0。
- <2> 使用 PCLBUZ0 引脚的时钟输出选择寄存器(CKS0)的位 0 至位 2 (CCS00 至 CCS02) 选择输出频率 (输出禁止状态)。
- <3> 将 CKS0 寄存器的位 7(PCLOE0)设置为 1, 以允许时钟/蜂鸣器输出。

备注 切换为允许或禁止时钟输出 (PCLOE0 位) 且经过一个时钟之后, 输出时钟的控制电路才开始或停止输出时钟。此时不会输出窄幅脉冲。使用 PCLOE0 位允许或停止输出以及输出时钟的时序如图 7-4 所示。

图 7-4. PCLBUZ0 引脚的时钟输出时序



注意事项 将 PCLBUZ0 引脚设置为停止输出(PCLOE0 = 0)后, 如果在 PCLBUZ0 引脚输出时钟的 1.5 个时钟之内切换到 STOP 模式, 则 PCLBUZ0 引脚的时钟输出宽度将变短。将 PCLBUZ0 引脚设置为停止输出后, 必须经过 PCLBUZ0 引脚输出时钟的 1.5 个时钟周期以上的时间后, 才能执行 STOP 指令。

第八章 看门狗定时器

8.1 看门狗定时器的功能

看门狗定时器通过用户选项字节(000C0H)设置计数操作。

看门狗定时器工作于低速片上振荡器时钟。

看门狗定时器用于检测程序失控。检测到程序失控时产生内部复位信号。

下述情况时检测到程序失控。

- 当看门狗定时器发生溢出时
- 对看门狗定时器允许寄存器 (WDTE) 执行位操作指令时
- 将 ACH 以外的数据写入 WDTE 寄存器时

由于看门狗定时器而发生复位时，复位控制标志寄存器(RESF)的位 4(WDTRF)被设置为 1。关于 RESF 寄存器的详情，请参阅 **第十四章 复位功能**。

达到 75%的溢出时间 + $3/(4 \times f_{IL})$ 时，产生间隔中断。

8.2 看门狗定时器的配置

看门狗定时器包括以下硬件。

表 8-1. 看门狗定时器的配置

项目	配置
控制寄存器	看门狗定时器允许寄存器(WDTE)

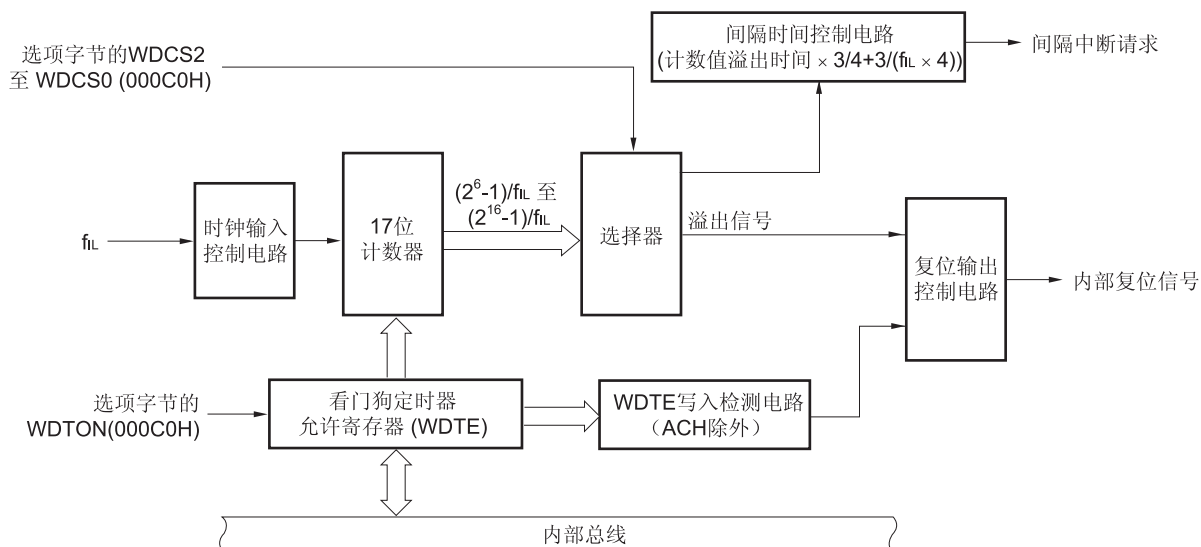
使用选项字节设置计数器工作方式和溢出时间。

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)
看门狗定时器的计数器操作控制	位4 (WDTON)
看门狗定时器的溢出时间	位3至1 (WDCS2至WDCS0)
看门狗定时器的计数器操作控制 (HALT/STOP模式下)	位0 (WDSTBYON)

备注 有关选项字节，请参阅第十六章 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

可使用看门狗定时器允许寄存器 (WDTE)控制看门狗定时器。

8.3.1 看门狗定时器允许寄存器(WDTE)

向 WDTE 寄存器写入 ACH 可以清除看门狗定时器计数器，并且再次开始计数。

使用 8 位存储器操作指令设置该寄存器。

产生复位信号后，该寄存器被设置为 1AH或 9AH[※]。

图 8-2. 看门狗定时器允许寄存器 (WDTE)的格式

地址: FFFABH 复位后: 1AH/9AH[※] R/W

符号	7	6	5	4	3	2	1	0
WDTE								

注 WDTE 寄存器复位值因选项字节(000C0H)的 WDTON 位的设置值而异。将 WDTON 位设置为 1 以启动看门狗定时器。

WDTON位设置值	WDTE寄存器复位值
0 (禁止看门狗定时器的计数操作)	1AH
1 (允许看门狗定时器的计数操作)	9AH

- 注意事项
1. 将 ACH 以外的值写入 WDTE 寄存器会产生内部复位信号。
 2. 对 WDTE 寄存器执行位操作指令会产生内部复位信号。
 3. WDTE 寄存器的读取值为 1AH/9AH (与写入值(ACH)不同)。

8.4 看门狗定时器的操作

8.4.1 控制看门狗定时器的操作

<1> 使用看门狗定时器时，下列操作由选项字节(000C0H)指定。

- 将选项字节(000C0H)的位 4(WDTON)设置为 1 以允许看门狗定时器的计数操作（解除复位后计数器开始操作）(详情请参阅 第十六章)。

WDTON	看门狗定时器计数器
0	禁止计数器操作(复位后停止计数)
1	允许计数器操作(复位后开始计数)

- 用选项字节 (000C0H)的位 3 至 1 (WDCS2 至 WDCS0) 设置溢出时间 (详情请参阅 8.4.2 和 第十六章)。

<2> 解除复位后，看门狗定时器开始计数。

<3> 在看门狗定时器开始计数之后到选项字节设置的溢出时间之前，向看门狗定时器允许寄存器(WDTE)写入 ACH，则看门狗定时器被清除并且再次开始计数。

<4> 如果超过溢出时间前未将 ACH 写入 WDTE 寄存器，将产生内部复位信号。

下列情况下也会产生内部复位信号：

- 对 WDTE 寄存器执行位操作指令时
- 将 ACH 以外的数据写入 WDTE 寄存器时

- 注意事项**
- 通过向 WDTE 寄存器写入 ACH 来清除看门狗定时器时，实际溢出时间与选项字节设置的溢出时间可能具有最大 1/f_{IL} 秒的差异。
 - 直到计数值溢出之前，都可以清除看门狗定时器。
 - 如下所示，看门狗定时器在 HALT 和 STOP 模式下的操作因选项字节(000C0H)的位 0(WDSTBYON) 的设置值而异。

WDSTBYON = 0：停止看门狗定时器操作。

WDSTBYON = 1：继续看门狗定时器操作。

如果 WDSTBYON = 0，解除 HALT 或 STOP 模式后看门狗定时器恢复计数。此时，计数器被清除为 0 并开始计数。

如果 WDSTBYON = 1，禁止设置为 WDTON = 0。

8.4.2 设置看门狗定时器的溢出时间

用选项字节(000C0H)的位 3 至位 1 (WDCS2 至 WDCS0) 设置看门狗定时器的溢出时间。

如果发生溢出，则产生内部复位信号。在溢出时间之前，通过向看门狗定时器允许寄存器(WDTE)写入 ACH，清除当前计数，看门狗定时器重新开始计数。

可设置的溢出时间如下所示。

达到 $3/4$ 的溢出时间 + $3/(f_{IL} \times 4)$ 时，产生间隔中断(INTWDTI)。

表 8-3. 设置溢出时间和间隔中断时间

WDCS2	WDCS1	WDCS0	溢出时间 (WDTRES)	间隔中断时间 (计数值上溢 $\times 3/4 + 3/(f_{IL} \times 4)$)
0	0	0	$(2^6 - 1)/f_{IL}$ (2.1 ms)	1.6 ms
0	0	1	$(2^7 - 1)/f_{IL}$ (4.23 ms)	3.2 ms
0	1	0	$(2^8 - 1)/f_{IL}$ (8.5 ms)	6.4 ms
0	1	1	$(2^9 - 1)/f_{IL}$ (17.03 ms)	12.8 ms
1	0	0	$(2^{11} - 1)/f_{IL}$ (68.23 ms)	51.2 ms
1	0	1	$(2^{13} - 1)/f_{IL}$ (273.03 ms)	204.8 ms
1	1	0	$(2^{14} - 1)/f_{IL}$ (546.1 ms)	409.6 ms
1	1	1	$(2^{16} - 1)/f_{IL}$ (2184.5 ms)	1638.4 ms

注意事项 即使在产生 INTWDTI 之后，看门狗定时器也会继续计数（直到将 ACH 写入看门狗定时器允许寄存器 (WDTE)为止）。如果未在溢出时间之前将 ACH 写入 WDTE 寄存器，则会产生内部复位信号。

备注 f_{IL} : 低速片上振荡器时钟频率

第九章 A/D 转换器

9.1 A/D 转换器的功能

A/D 转换器用于将模拟输入信号转换为数字值，被配置为可控制最多 4 个通道的模拟输入（AN10 至 AN13）。通过 A/D 转换器模式寄存器 2 (ADM2) 的 ADTYP 位，为 A/D 转换器选择 10 位分辨率或者 8 位分辨率。

A/D 转换器具有以下功能。

- A/D 转换

在 AN10 至 AN13 中选择 1 个通道的模拟输入，通过软件启动 10 位或 8 位分辨率 A/D 转换操作。完成 A/D 转换后产生中断请求(INTAD)。A/D 转换器的工作电压范围是 2.4 至 5.5 V。

9.2 A/D 转换器的配置

A/D 转换器包括以下硬件。

(1) ANI0 至 ANI3 引脚

这些是 A/D 转换器的 4 个通道的模拟输入引脚。它们用于输入模拟信号以转换成数字信号。被选为模拟输入引脚以外的其他引脚可以用作输入/输出端口引脚。

(2) 采样&保持电路

采样&保持电路对来自输入电路的各个模拟输入电压依次进行采样，并将其发送至 A/D 电压比较器。在 A/D 转换期间，该电路保持采样得到的模拟输入电压。

(3) A/D 电压比较器

A/D 电压比较器将比较电压生成电路产生的分接电压与模拟输入电压进行比较。对于比较结果，如果模拟输入电压大于基准电压($1/2 V_{DD}$)，则置位逐次逼近寄存器(SAR)的最高有效位(MSB)。如果模拟输入电压小于基准电压($1/2 V_{DD}$)，则将 SAR 寄存器的 MSB 位复位。

然后，SAR 寄存器的位 8 被自动置 1，并进行下一次比较。根据位 9 的设置结果值选择比较电压生成电路的分接电压。

位 9 = 0: ($1/4 V_{DD}$)

位 9 = 1: ($3/4 V_{DD}$)

将比较电压生成电路的分接电压与模拟输入电压进行比较，并根据比较结果，对 SAR 寄存器的位 8 进行操作。

模拟输入电压 $\geq$ 比较电压生成电路的分接电压：位 8 = 1

模拟输入电压 $\leq$ 比较电压生成电路的分接电压：位 8 = 0

继续进行比较，直到 SAR 寄存器的位 0 为止。

当以 8 位分辨率执行 A/D 转换时，继续进行比较直到 SAR 寄存器的位 2 为止。

(4) 比较电压生成电路

比较电压生成电路用于生成从模拟输入引脚输入的电压的比较电压。

(5) 逐次逼近寄存器(SAR)

SAR 寄存器从最高有效位(MSB)开始按顺序在每位设置分接电压数据, 该数据为与模拟输入引脚的电压值匹配的来自比较电压生成电路的分接电压值。

当设置到了 SAR 寄存器的最低有效位(LSB) (A/D 转换结束), 则 SAR 寄存器中的内容 (转换结果) 将保持在 A/D 转换结果高位存储寄存器(ADCRH)和 A/D 转换结果低位存储寄存器(ADCRL)中。当所有指定的 A/D 转换都结束时, 产生 A/D 转换结束中断请求信号 (INTAD)。

(6) A/D 转换结果高位存储寄存器(ADCRH)

ADCRH 为 8 位寄存器, 用于保持 10 位 A/D 转换结果的高 8 位。结果的低 2 位被存储在 ADCRL。

(7) A/D 转换结果低位存储寄存器(ADCRL)

ADCRL 为 8 位寄存器, 用于保持 10 位 A/D 转换结果的低 2 位(ADCR1, ADCR0)。该寄存器的低 6 位固定为 0。

(8) 控制电路

该电路用于控制将被转换为数字信号的模拟输入的转换时间, 并用于开始或停止转换操作。当 A/D 转换完成时, 该控制电路产生 INTAD 信号。

9.3 A/D 转换器中使用的寄存器

A/D 转换器使用以下寄存器。

- 外围允许寄存器 0 (PER0)
- A/D 转换器模式寄存器 0 (ADM0)
- A/D 转换器模式寄存器 2 (ADM2)
- A/D 转换结果高位存储寄存器(ADCRH)
- A/D 转换结果低位存储寄存器(ADCRL)
- 模拟输入通道选择寄存器(ADS)
- 端口模式控制寄存器 0 (PMC0)
- 端口模式寄存器 0 (PM0)

9.3.1 外围允许寄存器 0 (PER0)

该寄存器用于允许或禁止向外围硬件供应时钟。通过停止向未使用的硬件供应时钟，可以降低功耗和噪声。

使用 A/D 转换器时，必须将该寄存器的位 5(ADCEN)设置为 1。

使用 1 位或 8 位存储器操作指令设置 PER0 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 9-2. 外围允许寄存器 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	0	0	ADCEN	0	0	SAU0EN	0	TAU0EN

ADCEN	A/D转换器输入时钟供应的控制
0	停止输入时钟的供应。 • 不可写入用于A/D转换器的SFR。 • A/D转换器处于复位状态。
1	允许输入时钟的供应。 • 可以读取/写入用于A/D转换器的SFR。

注意事项 1. 设置 A/D 转换器时，必须在 ADCEN 位被设置为 1 的状态下设置以下寄存器。如果 ADCEN = 0，将忽略对 A/D 转换器控制寄存器进行的写操作，而且寄存器的读取值均为初始值（端口模式寄存器 0 (PM0)和端口模式控制寄存器 0 (PMC0)除外）。

- A/D 转换器模式寄存器 0 (ADM0)
- A/D 转换器模式寄存器 2 (ADM2)
- A/D 转换结果高位存储寄存器(ADCRH)
- A/D 转换结果低位存储寄存器(ADCRL)
- 模拟输入通道选择寄存器(ADS)

2. 必须将不确定位清除为 0。

9.3.2 A/D 转换器模式寄存器 0 (ADM0)

该寄存器用于设置进行 A/D 转换的模拟输入的转换时间，并开始/停止转换。

使用 1 位或 8 位存储器操作指令设置 ADM0 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 9-3. A/D 转换器模式寄存器 0 (ADM0)的格式

地址: FFF30H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	0	0	FR1 ^{注1}	FR0 ^{注1}	0	LV0 ^{注1}	ADCE

ADCS	A/D转换操作控制
0	停止转换操作 (转换停止/待机状态)
1	允许转换操作 (转换操作状态)
<清零条件> • ADCS清零时。 • A/D 转换结束时，自动清零。 <置位条件> • ADCE置1，ADCS清零时。	

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

注 1. 关于 FR1、FR0、LV0 位和 A/D 转换的详情，请参阅表 9-2. A/D 转换时间的选择。

2. A/D 电压比较器的操作由 ADCS 和 ADCE 位控制，从操作开始到操作稳定需要 0.1 μ s。因此，在 ADCE 位被设置为 1 且经过 0.1 μ s 或更长时间之后，将 ADCS 位设置为 1，则此时的转换结果相对于初次转换结果是有效的。若未等到 0.1 μ s 就将 ADCS 位设置为 1，则忽略转换结果。

注意事项 1. 请在转换处于待机状态 (ADCS = 0, ADCE = 1) 或者停止状态 (ADCS = 0, ADCE = 0) 时更改 FR1、FR0、LV0 位和 ADCS 位。

2. 禁止设置为 ADCS = 0、ADCE = 0。

3. 禁止使用 8 位操作指令同时将 ADCS = 0、ADCE = 0 更改为 ADCS = 1、ADCE = 1。请必须按照 9.7 A/D 转换器的设置流程图 中描述的顺序来设置这些位。

4. 必须将不确定位清除为 0。

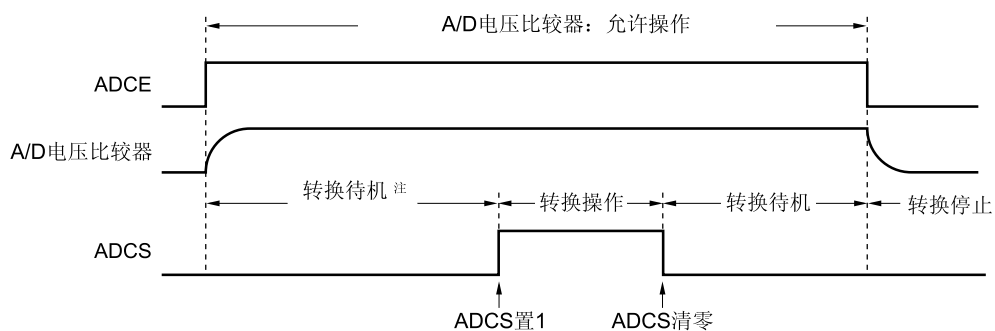
5. 进行转换的过程中 (ADCS = 1)，禁止再次将 ADCS 位设置为 1。在转换过程中 (ADCS = 1) 需要再次对同一通道进行转换的情况下，必须停止转换操作 (ADCS = 0) 后，重新开始进行 A/D 转换操作 (ADCS = 1)。

转换停止状态 (ADCS = 0, ADCE = 0) 时，即使向 ADCS 写入 1 也不能将其设置为 1。

表 9-1. ADCS 和 ADCE 位的设置

ADCS	ADCE	A/D 转换操作
0	0	转换停止状态
0	1	转换待机状态
1	0	禁止设置
1	1	转换操作状态

图 9-4. 使用 A/D 电压比较器时的时序图



注 从 ADCE 位上升到 ADCS 位上升的时间必须为 $0.1 \mu\text{s}$ 或更长，以使内部电路稳定。

表 9-2. A/D 转换时间的选择

(1) $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$

A/D转换器模式寄存器0 (ADM0)			转换时钟	转换时钟数	转换时间	转换时间的选择				
FR1	FR0	LV0				f _{CLK} = 1.25 MHz	f _{CLK} = 2.5 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz
0	0	0	f _{CLK} /8	19 f _{AD} (采样时钟数: 7 f _{AD})	184/f _{CLK}	禁止设置	禁止设置	禁止设置	18.4	禁止设置
0	1		f _{CLK} /4		92/f _{CLK}			18.4	9.2	
1	0		f _{CLK} /2		46/f _{CLK}		18.4	9.2	4.6	
1	1		f _{CLK}		23/f _{CLK}	18.4	9.2	4.6	禁止设置	

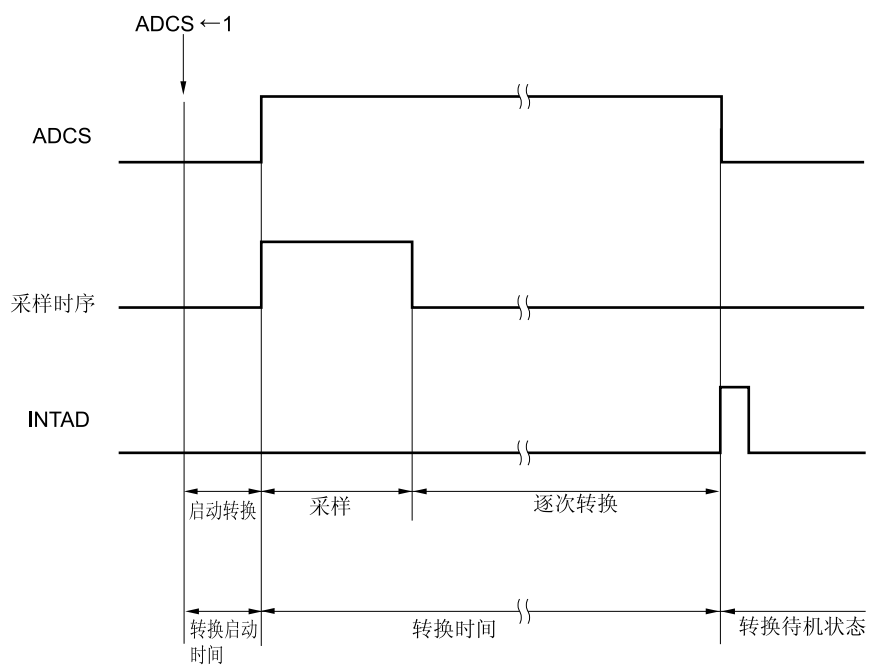
(2) $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$

A/D转换器模式寄存器0 (ADM0)			转换时钟	转换时钟数	转换时间	转换时间的选择				
FR1	FR0	LV0				f _{CLK} = 1.25 MHz	f _{CLK} = 2.5 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz
0	0	0	f _{CLK} /8	19 f _{AD} (采样时钟数: 7 f _{AD})	184/f _{CLK}	禁止设置	禁止设置	禁止设置	18.4	9.2
0	1		f _{CLK} /4		92/f _{CLK}			18.4	9.2	4.6
1	0		f _{CLK} /2		46/f _{CLK}		18.4	9.2	4.6	禁止设置
1	1		f _{CLK}		23/f _{CLK}	18.4	9.2	4.6	禁止设置	
0	0	1	f _{CLK} /8	17 f _{AD} (采样时钟 数: 5 f _{AD})	136/f _{CLK}	禁止设置	禁止设置	禁止设置	13.6	6.8
0	1		f _{CLK} /4		68/f _{CLK}			13.6	6.8	3.4
1	0		f _{CLK} /2		34/f _{CLK}		13.6	6.8	3.4	禁止设置
1	1		f _{CLK}		17/f _{CLK}	13.6	6.8	3.4	禁止设置	

- 注意事项 1. 暂停 A/D 转换(ADCS = 0)之后再写 FR1、FR0 和 LV0 位改写为不同数据。
2. 上述转换时间不包括转换启动时间。第一次转换中要加上转换启动时间。选择转换时间时要考虑到时钟频率误差。
3. 不要同时改写 FR1、FR0、LV1 和 ADCS 位。

备注 f_{CLK}: CPU/外围硬件时钟频率

图 9-5. A/D 转换器采样和 A/D 转换的时序



9.3.3 A/D 转换器模式寄存器 2 (ADM2)

该寄存器用于选择 A/D 转换器的分辨率。

使用 1 位或 8 位存储器操作指令设置 ADM2 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 9-6. A/D 转换器模式寄存器 2 (ADM2)的格式

地址: F0010H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	<0>
ADM2	0	0	0	0	0	0	0	ADTYP

ADTYP	A/D 转换分辨率
0	10 位分辨率
1	8 位分辨率

注意事项 仅在停止状态下（即 A/D 转换器模式寄存器 0 (ADM0)的 ADCS 和 ADCE 位被设置为 0 时）改写 ADM2 寄存器的值。

9.3.4 A/D 转换结果高位存储寄存器(ADCRH)

该寄存器为 8 位寄存器，用于保持 10 位 A/D 转换结果的高 8 位。结果的低 2 位被存储在 ADCRL。

使用 8 位存储器操作指令读取 ADCRH 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 9-7. A/D 转换结果高位存储寄存器(ADCRH)的格式

地址: FFF1FH 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ADCRH	ADCR9	ADCR8	ADCR7	ADCR6	ADCR5	ADCR4	ADCR3	ADCR2

注意事项 写入 A/D 转换器模式寄存器 0 (ADM0)和模拟输入通道选择寄存器(ADS)之后，ADCRH/ADCRL 寄存器的内容可能变成不定。在转换完成后，写入 ADM0 和 ADS 寄存器之前读取转换结果。

9.3.5 A/D 转换结果低位存储寄存器(ADCRL)

该寄存器为 8 位寄存器，用于保持 10 位 A/D 转换结果的低 2 位。低 6 位固定为 0。

使用 8 位存储器操作指令读取 ADCRL 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 9-8. A/D 转换结果低位存储寄存器(ADCRL)的格式

地址: FFF1EH 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ADCRL	ADCR1	ADCR0	0	0	0	0	0	0

存储了 10 位分辨率 A/D 转换结果后的状态如图 9-9 所示。每次 A/D 转换结束后，从逐次逼近寄存器(SAR)载入转换结果。转换结果的高 8 位被存储在 ADCRH，结果的低 2 位被存储在 ADCRL。

图 9-9. 存储了 10 位分辨率 A/D 转换结果后的状态

ADCRH								ADCRL							
ADCR9	ADCR8	ADCR7	ADCR6	ADCR5	ADCR4	ADCR3	ADCR2	ADCR1	ADCR0	0	0	0	0	0	0

- 注意事项**
1. 写入 A/D 转换器模式寄存器 0 (ADM0)和模拟输入通道选择寄存器(ADS)之后，ADCRH/ADCRL 寄存器的内容可能变成不定。在转换完成后，写入 ADM0 和 ADS 寄存器之前读取转换结果。如果不在上述时序进行，则可能读取错误的转换结果。
 2. 不产生 INTAD 的情况下，A/D 转换结果将不被存储到 ADCRH 和 ADCRL 寄存器。
 2. 当选择 8 位分辨率 A/D 转换（即当 A/D 转换器模式寄存器 2(ADM2)的 ADTYP 位为 1 时）且读取 ADCRH 和 ADCRL 寄存器时，从低 2 位（ADCR1 和 ADCR0）读取 0。请注意，当选择 8 位分辨率 A/D 转换且在 A/D 转换完成之前读取 ADCRH 和 ADCRL 寄存器时，从低 2 位（ADCR1 和 ADCR0）可能读取 0。

9.3.6 模拟输入通道选择寄存器(ADS)

该寄存器用于设置执行 A/D 转换的模拟电压的输入通道。

使用 1 位或 8 位存储器操作指令设置 ADS 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 9-10. 模拟输入通道选择寄存器(ADS)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	0	0	ADS1	ADS0

ADS1	ADS0	模拟输入通道	输入源
0	0	ANI0	P01/ANI0 引脚
0	1	ANI1	P02/ANI1 引脚
1	0	ANI2	P03/ANI2 引脚
1	1	ANI3	P04/ANI3 引脚

注意事项 1. 必须将不确定位清除为 0。

2. 对于用作模拟输入端口的端口，通过端口模式寄存器 0 (PM0)将其设置为输入模式，通过端口模式控制寄存器 0 (PMC0)将其设置为模拟输入模式。

3. 执行 A/D 转换时(ADCS = 1)，禁止写入 ADS 寄存器。切换模拟输入通道时，必须在转换待机状态 (ADCS = 0, ADCE = 1) 或者转换停止状态 (ADCS = 0, ADCE = 0) 下改写 ADS 寄存器。

9.3.7 端口模式控制寄存器 0 (PMC0)

使用该寄存器以 1 位为单位设置端口 0 的数字输入/输出或模拟输入。
用作模拟输入端口时，将 PMC01、PMC02、PMC03 和 PMC04 位设置为 1。
使用 1 位或 8 位存储器操作指令设置 PMC0 寄存器。
产生复位信号后，该寄存器被设置为 FFH。

图 9-11. 端口模式控制寄存器 0 (PMC0)的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PMC0	1	1	1	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W

PMCmn	选择 Pmn 引脚数字输入/输出和模拟输入(m = 0; n = 1 至 4)
0	数字输入/输出 (模拟输入以外的复用功能)
1	模拟输入

- 注意事项
1. 通过端口模式寄存器 0 (PM0)将由 PMC0 和 PMC1 寄存器设置为模拟输入的端口设置为输入模式。
 2. 对于由 PMC0 寄存器设置为数字输入/输出的引脚，不要使用模拟输入通道选择寄存器(ADS)对其进行设置。

9.3.8 端口模式寄存器 0 (PM0)

该寄存器以 1 位为单位设置端口的输入/输出。

当使用 ANI0 至 ANI3 引脚作为模拟输入端口时，将 PM0n 位设置为 1。此时，P0n 的输出锁存器可能为 0 或 1。

如果 PM0n 位被设置为 0，则不能用作模拟输入端口。

使用 1 位或 8 位存储器操作指令设置 PM0 和 PMC0 寄存器。

产生复位信号后，该寄存器被设置为 FFH。

注意事项 读取被设置为模拟输入端口的引脚时，读取值不是引脚电平而是一直为 0。

图 9-12. 端口模式寄存器 0 (PM0)的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W

PMmn	Pmn 引脚输入/输出模式选择(m = 0; n = 0 至 4)
0	输出模式 (输出缓冲器启用)
1	输入模式 (输出缓冲器关闭)

如下所示，P01/ANI0 至 P04/ANI3 的引脚功能，因端口模式控制寄存器 0 (PMC0)、模拟输入通道选择寄存器(ADS)和端口模式寄存器 0 (PM0)的设置而异。

表 9-3. ANI0 至 ANI3 引脚的功能

PMC0	PM0	ADS	功能
数字输入/输出	输入模式	—	数字输入
	输出模式	—	数字输出
模拟输入	输入模式	选择ANI	模拟输入 (被转换)
		不选择ANI	模拟输入 (不被转换)
	输出模式	选择ANI	禁止设置
		不选择ANI	

9.4 A/D 转换器的转换操作

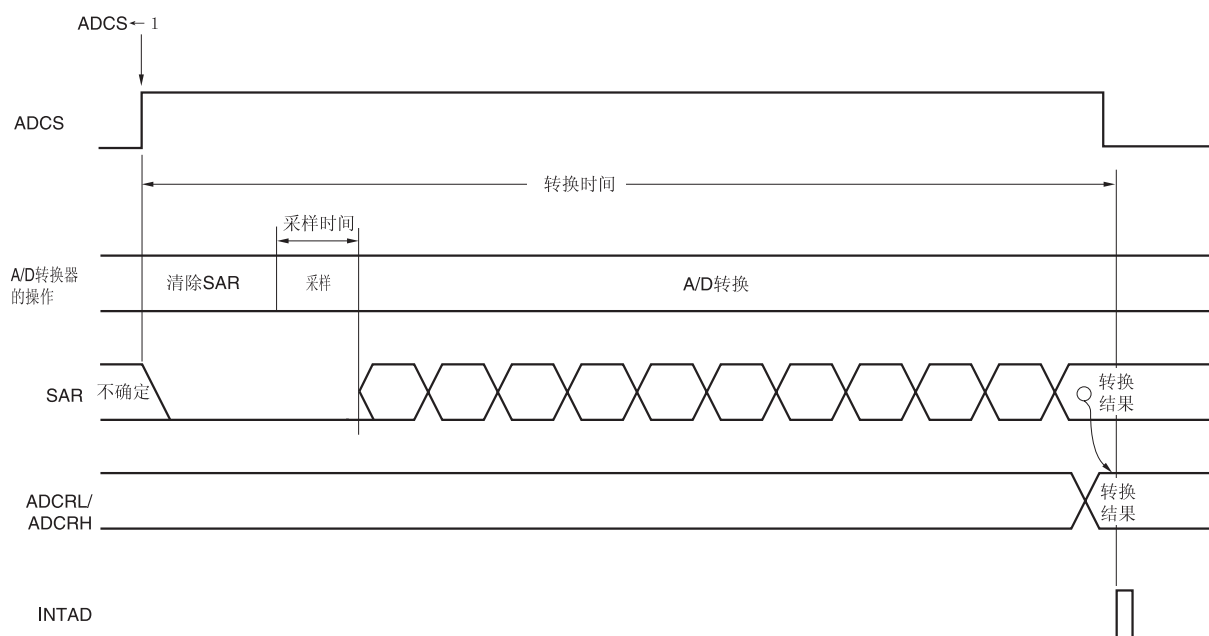
A/D 转换器的转换操作如下所示。

- <1> 通过采样&保持电路对选择的模拟输入通道的输入电压进行采样。
- <2> 当采样进行一定时间时，采样&保持电路被置于保持状态且保持采样电压，直到 A/D 转换操作结束为止。
- <3> 置位逐次逼近寄存器(SAR)的位 9。通过分接选择器将串联电阻串的分接电压设置为 $(1/2) V_{DD}$ 。
- <4> 使用电压比较器对串联电阻串的分接电压与采样电压之间的电压差进行比较。如果模拟输入大于 $(1/2) V_{DD}$ ，SAR 寄存器的 MSB 位保持为 1。如果模拟输入小于 $(1/2) V_{DD}$ ，MSB 位复位为 0。
- <5> 接着，自动将 SAR 寄存器的位 8 设置为 1，并进行下一次比较。根据下述位 9 的当前设置值选择串联电阻串的分接电压。
 - 位 9 = 1: $(3/4) V_{DD}$
 - 位 9 = 0: $(1/4) V_{DD}$对分接电压与采样电压进行比较，并对 SAR 寄存器的位 8 进行如下操作：
 - 采样电压 $\geq$ 分接电压：位 8 = 1
 - 采样电压 $<$ 分接电压：位 8 = 0
- <6> 照此继续进行比较，直到 SAR 寄存器的位 0 为止。
- <7> 完成 10 位的比较时，一个有效数字结果值将保持于 SAR 寄存器之中，该结果值被传送至 A/D 转换结果寄存器 (ADCRH, ADCRL)，并被锁存。
同时，产生 A/D 转换结束中断请求(INTAD)。
A/D 转换结束后，ADCS 位将被自动清除为 0，系统进入 A/D 转换待机状态。

备注 有两种 A/D 转换结果寄存器。

- ADCRL 寄存器(8 位): 存储 10 位 A/D 转换值的低 2 位
- ADCRH 寄存器(8 位): 存储 10 位 A/D 转换值的高 8 位或 8 位 A/D 转换值

图 9-13. A/D 转换器的转换操作



通过软件将 A/D 转换器模式寄存器 0 (ADM0) 的位 7 (ADCS) 设置为 1，进行 1 次的 A/D 转换。
产生复位信号后，A/D 转换结果寄存器(ADCRL, ADCRH)被清除为 00H。

9.5 输入电压和转换结果

输入至模拟输入引脚（ANIO 至 ANI3）的模拟输入电压与理论 A/D 转换结果（存储于 A/D 转换结果寄存器(ADCR) (= ADCRH + ADCRL)中）之间的关系可用以下表达式表示。

$$\text{SAR} = \text{INT} \left(\frac{V_{\text{AIN}}}{V_{\text{DD}}} \times 1024 + 0.5 \times 1024 + 0.5 \right)$$

$$\text{ADCR} = \text{SAR} \times 64$$

或

$$\left(\frac{\text{ADCR}}{64} - 0.5 \right) \times \frac{V_{\text{DD}}}{1024} \leq V_{\text{AIN}} < \left(\frac{\text{ADCR}}{64} + 0.5 \right) \times \frac{V_{\text{DD}}}{1024}$$

此处，INT()：返回括号中数值的整数部分的函数

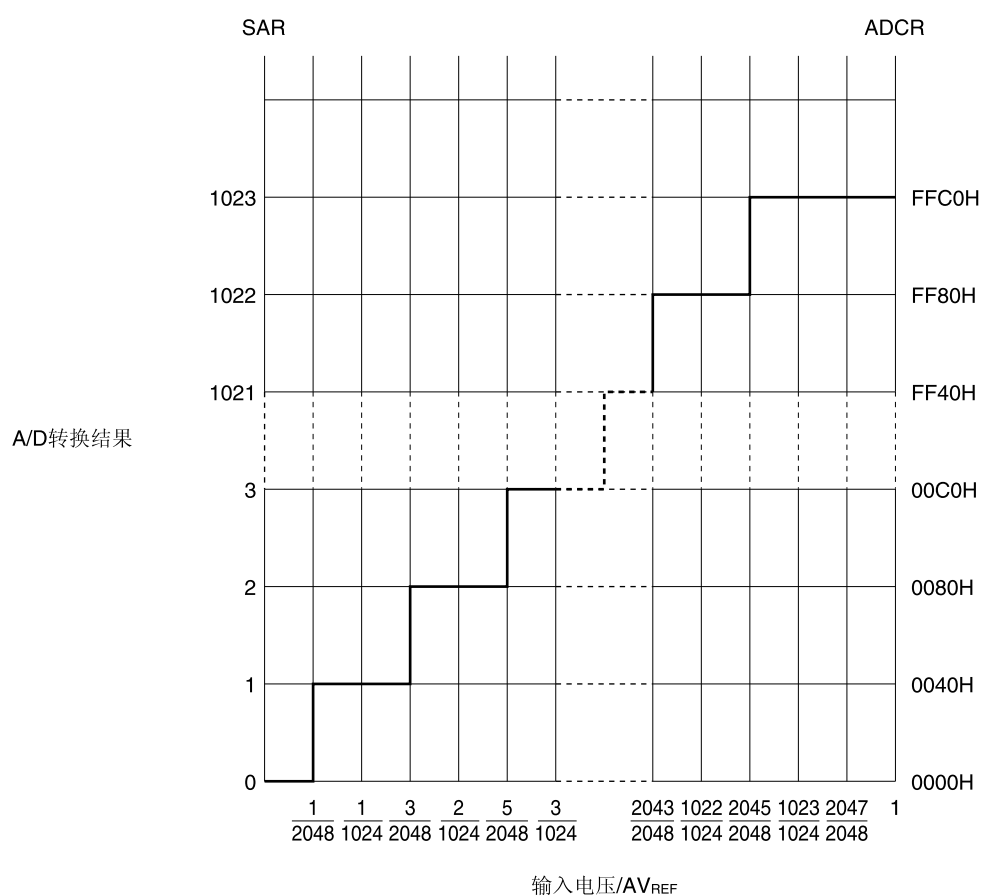
V_{AIN} ：模拟输入电压

ADCR：A/D 转换结果寄存器(ADCRH + ADCRL)的值

SAR：逐次逼近寄存器

图 9-14 表示模拟输入电压与 A/D 转换结果之间的关系。

图 9-14. 模拟输入电压与 A/D 转换结果之间的关系

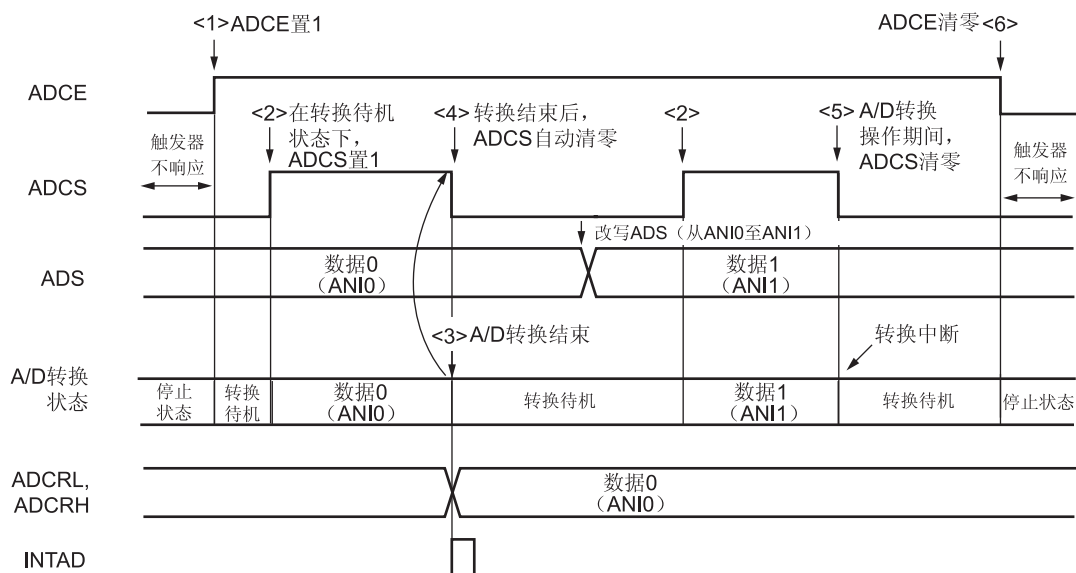


9.6 A/D 转换器的操作模式

A/D 转换器的操作如下所示。另外，设置步骤请参阅 9.7 A/D 转换器的设置流程图。

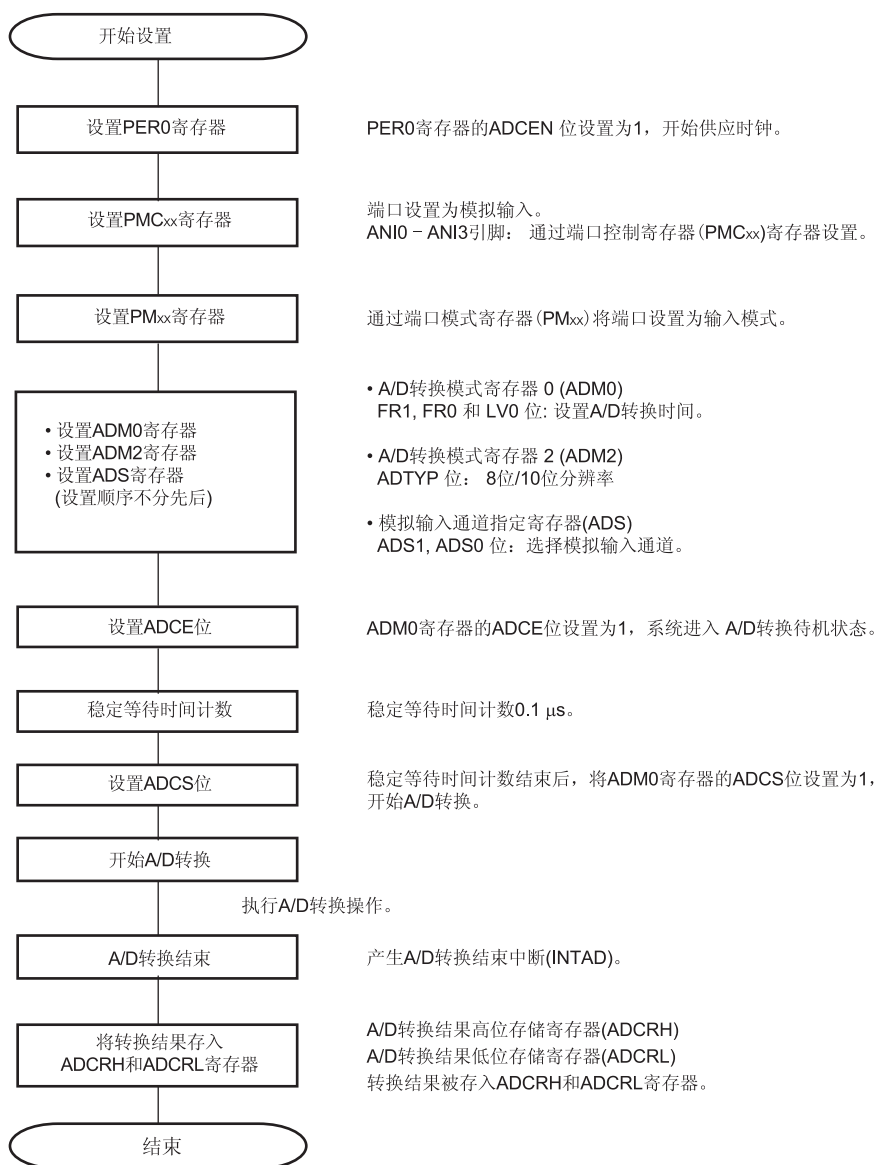
- <1> 在转换停止状态下，A/D 转换器模式寄存器 0(ADM0)的 ADCE 位被设置为 1，系统进入 A/D 转换待机状态。
- <2> 软件计数至稳定等待时间(0.1 μ s)之后，ADM0 寄存器的 ADCS 位被设置为 1，以执行由模拟输入通道选择寄存器(ADS)指定的模拟输入的 A/D 转换。
- <3> A/D 转换结束时，转换结果存储于 A/D 转换结果寄存器(ADCRL, ADCRH)，并产生 A/D 转换结束中断请求信号(INTAD)。
- <4> A/D 转换结束后，ADCS 位将被自动清除为 0，系统进入 A/D 转换待机状态。
- <5> 转换操作过程中将 ADCS 清除为 0 时，当前 A/D 转换中断，系统进入 A/D 转换待机状态。
- <6> 在 A/D 转换待机状态下，当 ADCE 被清除为 0 时，A/D 转换器进入停止状态。当 ADCE=0 时，将 ADCS 设置为 1 的操作无效，不开始 A/D 转换。

图 9-15. 操作时序的示例



9.7 A/D 转换器的设置流程图

A/D 转换器的设置流程图如下所示。



9.8 如何阅读 A/D 转换器特性表

以下将介绍 A/D 转换器的专用术语。

9.8.1 分辨率

可以分辨的最低模拟输入电压。即是说，每位数字输出的模拟输入电压的百分比被称为 **1LSB**（最低有效位）。1LSB 对满刻度的比率被表示为 %FSR（满刻度范围）。

当分辨率为 10 位时，1LSB 表示如下：

$$\begin{aligned} 1\text{LSB} &= 1/2^{10} = 1/1024 \\ &= 0.098\%\text{FSR} \end{aligned}$$

精度与分辨率无关，而是取决于总误差。

9.8.2 总误差

总误差是实际测量值和理论值之间的最大误差值。

将零刻度误差、满刻度误差、积分线性误差和微分线性误差以及这些组合所产生的误差综合起来，即为总误差。

请注意，总误差特性表中不包括量化误差。

9.8.3 量化误差

当模拟值被转换为数字值时，必然会出现 $\pm 1/2\text{LSB}$ 的误差。在 A/D 转换器中， $\pm 1/2\text{LSB}$ 范围内的模拟输入电压被转换成同样的数字代码，因此量化误差不可避免。

请注意，量化误差不包括在特性表中的总误差、零刻度误差、满刻度误差、积分线性误差和微分线性误差之中。

图 9-16. 总误差

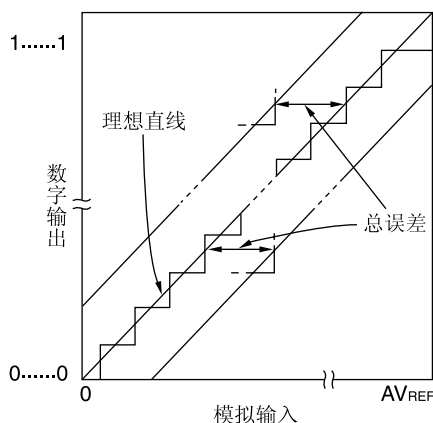
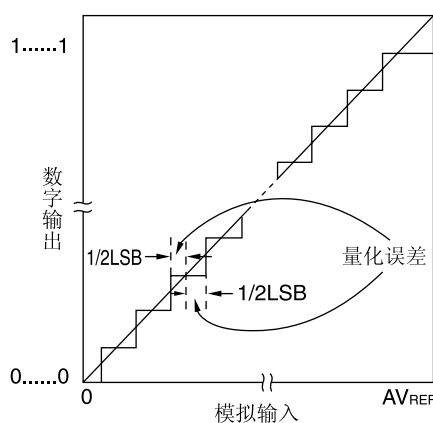


图 9-17. 量化误差



9.8.4 零刻度误差

零刻度误差是指当数字输出从 0.....000 变为 0.....001 时，模拟输入电压的实际测量值和理论值(1/2LSB)之间的差值。

如果实际测量值大于理论值，则表示当数字输出从 0.....001 变为 0.....010 时，模拟输入电压的实际测量值和理论值(3/2LSB)之间的差值。

9.8.5 满刻度误差

满刻度误差是当数字输出从 1.....110 变为 1.....111 时，模拟输入电压的实际测量值和理论值（满刻度 - 3/2LSB）之间的差值。

9.8.6 积分线性误差

积分线性误差说明了转换特性偏离理想线性关系的程度。它表示当零刻度误差和满刻度误差均为 0 时，实际测量值和理想直线之间的最大误差。

9.8.7 微分线性误差

微分线性误差是指，输出代码的理想宽度为 1LSB 时，某个代码输出宽度的实际测量值与理想值之间的差值。

图 9-18. 零刻度误差

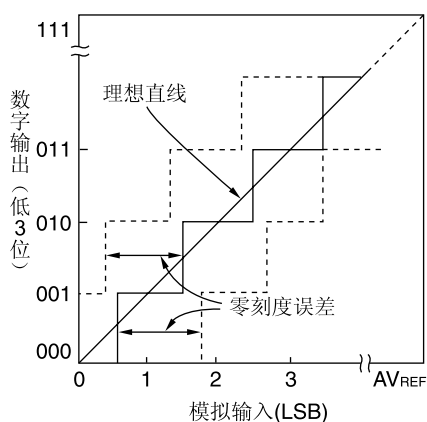


图 9-19. 满刻度误差

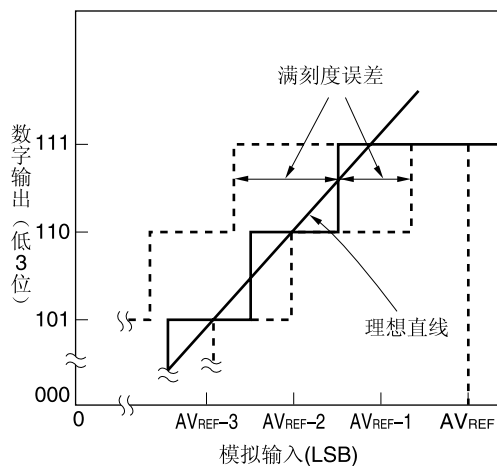


图 9-20. 积分线性误差

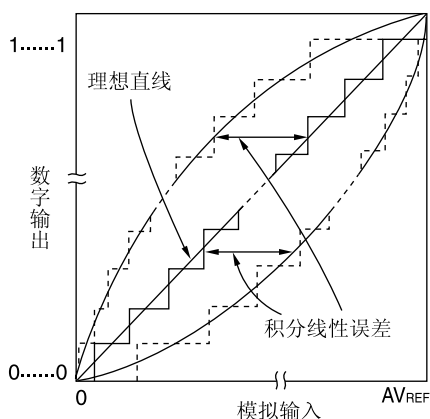
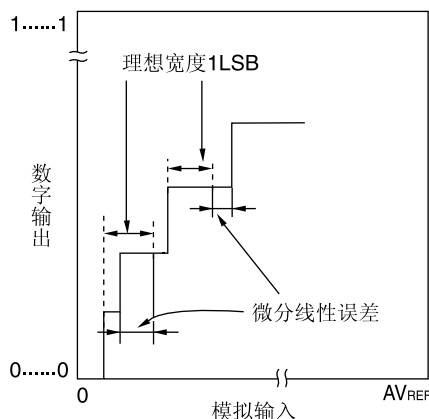


图 9-21. 微分线性误差



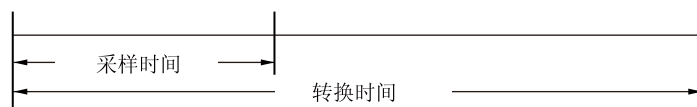
9.8.8 转换时间

转换时间是指从开始采样到获得数字输出的时间。

特性表中的转换时间包括采样时间。

9.8.9 采样时间

采样时间是指为了将模拟电压载入采样&保持电路而打开模拟开关的时间。



9.9 A/D 转换器的注意事项

9.9.1 STOP 模式下的操作电流

停止 A/D 转换器（将 A/D 转换器模式寄存器 0(ADM0)的位 7(ADCS)设置为 0）之后切换至 STOP 模式。同时，可通过将 ADM0 寄存器的位 0(ADCE)设置为 0 来减少操作电流。

9.9.2 ANI0 至 ANI3 引脚的输入范围

请遵守 ANI0 至 ANI3 引脚输入电压的额定范围。如果向模拟输入通道输入大于或等于 V_{DD} 或者小于或等于 V_{SS} 的电压（即使在绝对最大额定范围内），则该通道的转换值将不定。另外，也可能影响到其他通道的转换值。

9.9.3 冲突操作

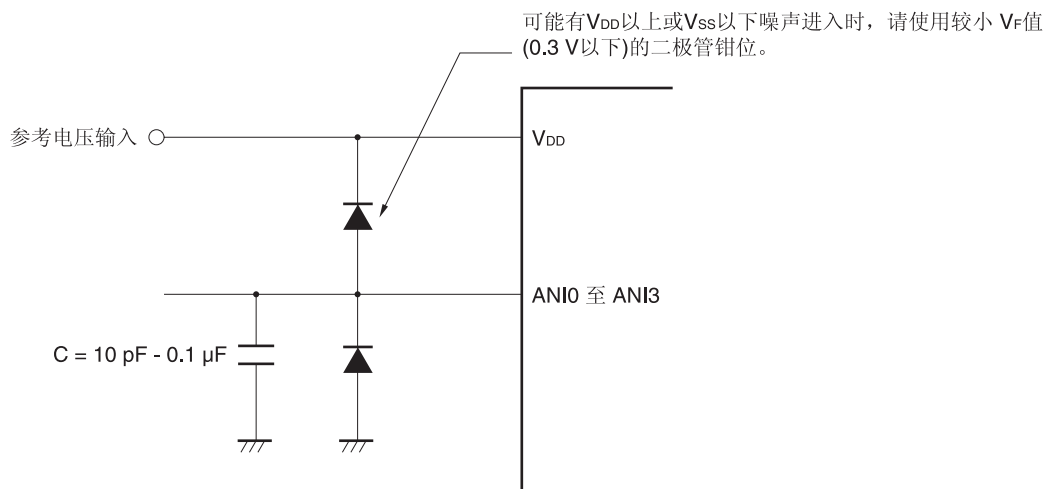
转换结束时，向 ADCRH 或 ADCRL 寄存器的写入操作与向 A/D 转换器模式寄存器 0 (ADM0)写入 0 的操作（ADCS 除外）发生冲突时，优先写入 ADM0 寄存器。不执行对 ADCRH 或 ADCRL 寄存器的写入，也不产生转换结束中断信号 (INTAD)。

9.9.4 噪声对策

为了保持 10 位分辨率，必须注意输入至 V_{DD} 和 ANI0 至 ANI13 引脚的噪声。

- <1> 连接到电源的电容器请使用具备等效电阻低且频率响应良好的电容器。
- <2> 模拟输入源的输出阻抗越高，影响越大。如图 9-22 所示，建议外接 C 以减少噪声。
- <3> 转换过程中，不要切换引脚。
- <4> 转换开始后立即设置为 HALT 模式可以提高精度。

图 9-22. 模拟输入引脚的连接



9.9.5 模拟输入(ANIn)引脚

- <1> 模拟输入引脚（ANI0 至 ANI3）也用作输入端口引脚（P01 至 P04）。
选择 ANI0 至 ANI3 引脚中的任意一个引脚并对其执行 A/D 转换时，不要在转换过程中改变复用端口 P01 至 P04 的输出值。否则，可能导致转换分辨率下降。
- <2> 如果将正在执行 A/D 转换的引脚相邻的引脚用作数字输入/输出端口的话，由于耦合噪声的影响，可能会取得和预期不同的 A/D 转换值。因此，请不要输入或输出这样的脉冲。

9.9.6 模拟输入(ANIn)引脚的输入阻抗

A/D 转换器在采样时间对内部的采样电容充电，并进行采样。

因此，不进行采样时仅通过漏电流，采样时还通过对电容充电的电流。所以输入阻抗将依据是否进行采样而产生波动。

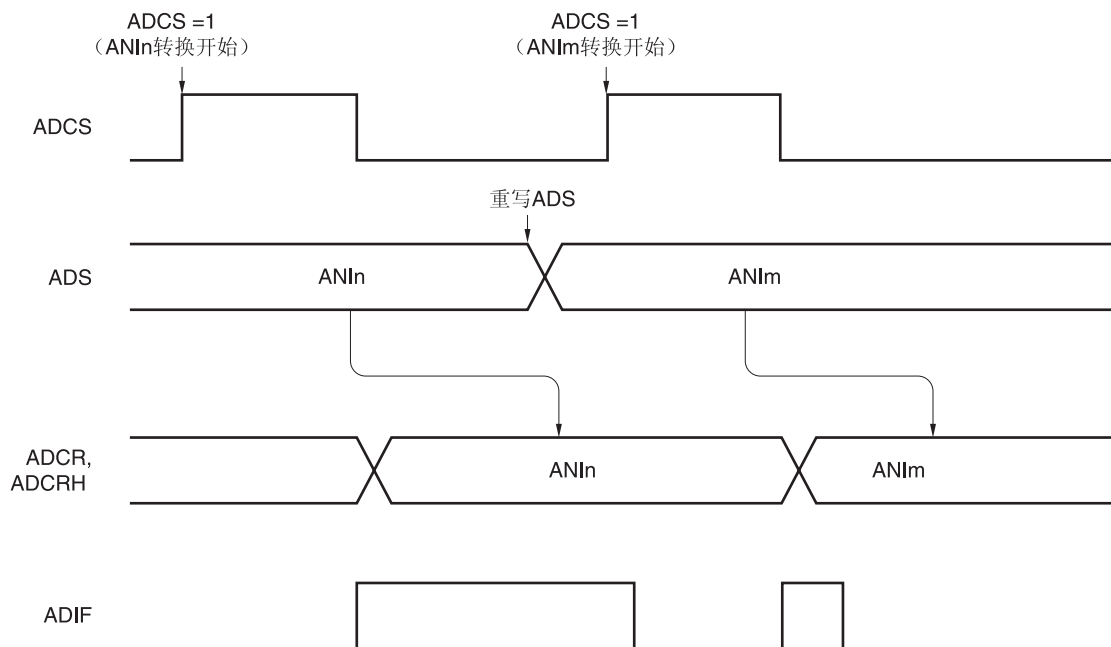
但是，为了保证采样有效，建议将模拟输入源的输出阻抗保持在 $1\text{k}\Omega$ 以内，并且将大约 $0.1\ \mu\text{F}$ 的电容连接至 ANI0 至 ANI3 引脚(参阅图 9-22)。

9.9.7 中断请求标志(ADIF)

即使更改模拟输入通道选择寄存器(ADS)，中断请求标志(ADIF)也不会被清除。

另外，暂停 A/D 转换后又重新开始时，须在重新开始 A/D 转换之前清除 ADIF 标志。

图 9-23. 产生 A/D 转换结束中断请求的时序



9.9.8 A/D 转换开始后的初次转换结果

如果在 ADCE 位设置为 1 之后的 0.1 μ s 内将 ADCS 位设置为 1，则开始 A/D 转换操作后初次产生的 A/D 转换值可能不在额定范围内。须采取例如轮询 A/D 转换结束中断请求(INTAD)以舍弃初次的转换结果等措施。

9.9.9 读取 A/D 转换结果寄存器(ADCRH, ADCRL)

如果对 A/D 转换器模式寄存器 0 (ADM0)、模拟输入通道选择寄存器(ADS)和端口模式控制寄存器(PMCxx)执行写入操作，ADCRH 和 ADCRL 寄存器的内容将成为不确定值，可能导致不能读出正确的转换结果。转换操作结束后，请在读取转换结果后再写入 ADM0、ADS 和 PMCxx 寄存器。

9.9.10 内部等效电路

模拟输入模块的等效电路如下所示。

图 9-24. ANIn 引脚的内部等效电路

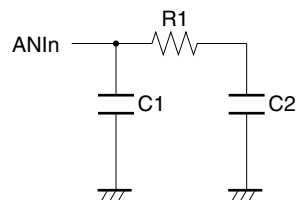


表 9-4. 等效电路的电阻和电容值 (参考值)

V_{DD}	引脚	R1 [kΩ]	C1 [pF]	C2 [pF]
$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	ANI0至ANI3	40	8	1.7
$2.4\text{ V} \leq V_{DD} < 2.7\text{ V}$	ANI0至ANI3	200		1.7

备注 表 9-4 所示电阻和电容值均非保证值。

9.9.11 开始 A/D 转换器的操作

在 V_{DD} 的电压稳定后，开始进行 A/D 转换器的操作。

第十章 串行阵列单元

串行阵列单元 0 有两个串行通道。各通道可以实现 3 线串行(CSI)和 UART 通信。

R7F0C80112ESP, R7F0C80212ESP 中支持的各通道的功能分配如下所示。

单元	通道	用作 CSI	用作 UART
0	0	CSI00	UART0
	1	—	

不能在复数种的通信方式下同时使用 1 个通道。

10.1 串行阵列单元的功能

R7F0C80112ESP, R7F0C80212ESP 中支持的各个串行接口均具有如下特点。

10.1.1 3 线串行输入/输出(CSI00)

与主通道输出的串行时钟(SCK)同步后执行数据的发送或接收。

3 线串行通信为时钟同步式通信功能，通过三条通信线路实现：一条用于串行时钟(SCK)、一条用于发送串行数据(SO)、一条用于接收串行数据(SI)。

有关具体的设置示例，请参阅 10.5 3 线串行输入/输出(CSI00)通信的操作。

[数据的发送/接收]

- 数据长度为 7 或 8 位
- 发送/接收数据的相控制
- MSB/LSB 选择
- 发送/接收数据的电平设置

[时钟控制]

- 主/从选择
- 输入/输出时钟的相控制
- 通过预分频器和通道内部计数器设置传送周期
- 最大传送速率[※]

主通信时: Max. f_{CLK}/4

从通信时: Max. f_{CLK}/6

[中断功能]

- 传送结束中断/缓冲器空中断

[错误检测标志]

- 溢出错误

注 在满足 SCK 周期时间(t_{KCY})特性的范围内使用时钟(参阅第二十一章 电特性)。

10.1.2 UART (UART0)

这是一种使用 2 线：串行数据发送线(TxD)和串行数据接收线(RxD)的异步通信功能。使用这两条通信线路，各数据帧（由一个起始位、数据、奇偶校验位和停止位构成）在单片机与其他通信方之间（以内部波特率）异步传送。全双工 UART 通信可以使用一个发送专用通道（偶数通道）和一个接收专用通道（奇数通道）来实现。

有关具体的设置示例，请参阅 **10.6 UART(UART0)通信的操作**。

[数据的发送/接收]

- 数据长度为 7 或 8 位
- MSB/LSB 选择
- 发送/接收数据的电平设置及反相选择
- 附加奇偶校验位和奇偶校验功能
- 附加停止位

[中断功能]

- 传送结束中断/缓冲器空中断
- 出现帧错误、奇偶检验错误或溢出错误时的错误中断

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

10.2 串行阵列单元的配置

定时器阵列单元包含以下硬件。

表 10-1. 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 0n (SDR0nH, SDR0nL)
串行时钟输入/输出	SCK00 引脚 (用作 3 线串行输入/输出)
串行数据输入	SI00 引脚 (用作 3 线串行输入/输出), RxD0 引脚 (用作 UART)
串行数据输出	SO00 引脚 (用作 3 线串行输入/输出), TxD0 引脚 (用作 UART), 输出控制电路
控制寄存器	<单元设置块的寄存器> • 外围允许寄存器 0 (PER0) • 串行时钟选择寄存器 0 (SPS0) • 串行通道允许状态寄存器 0 (SE0) • 串行通道开始寄存器 0 (SS0) • 串行通道停止寄存器 0 (ST0) • 串行输出允许寄存器 0 (SOE0) • 串行输出寄存器 0 (SO0) • 串行时钟输出寄存器 0 (CKO0) • 串行输出电平寄存器 0 (SOL0) • 噪声滤波器允许寄存器 0 (NFEN0) • 输入切换控制寄存器 (ISC)
	<每个通道的寄存器> • 串行数据寄存器 0n (SDR0nH, SDR0nL[※]) • 串行模式寄存器 0n (SMR0nH, SMR0nL) • 串行通信操作设置寄存器 0n (SCR0n) • 串行状态寄存器 0n (SSR0n) • 串行标志清除触发寄存器 0n (SIR0n)
	• 端口输出模式寄存器 0 (POM0) • 端口模式控制寄存器 0 (PMC0) • 端口模式寄存器 0 (PM0) • 端口寄存器 0 (P0)

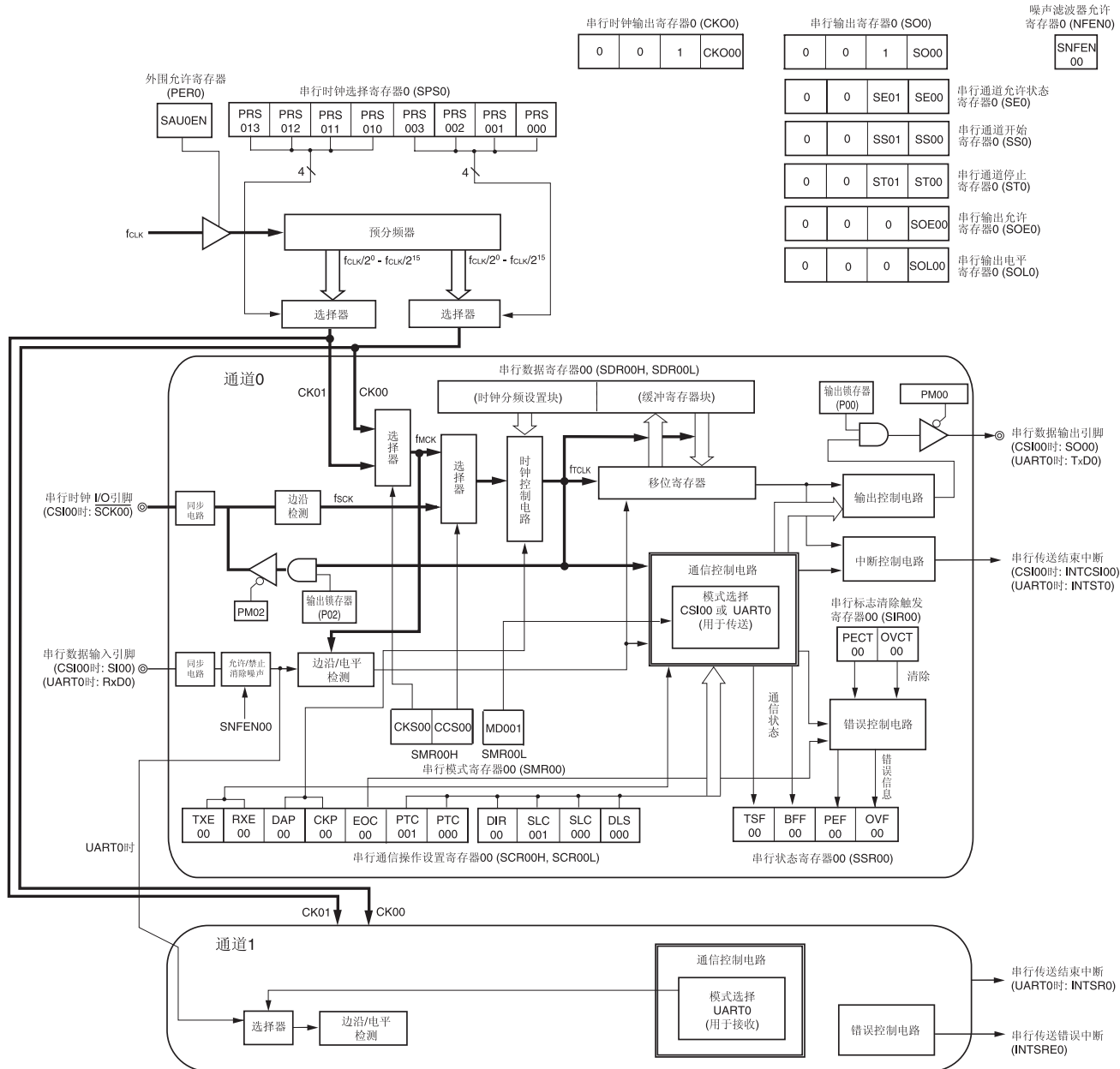
注 依据通信模式，串行数据寄存器 0nL (SDR0nL)可以按以下 SFR 名称读取或写入。

- CSIp 通信时: SIOp (CSIp 数据寄存器)
- UART0 接收期间: RXD0 (UARTq 接收数据寄存器)
- UART0 发送期间: TXD0 (UARTq 发送数据寄存器)

备注 n: 通道编号 (n = 0, 1), p: CSI 编号 (p = 00), q: UART 编号 (q = 0)

串行阵列单元 0 的框图如图 10-1 所示。

图 10-1. 串行阵列单元 0 的框图



10.2.1 移位寄存器

这是一个 9 位寄存器，可将并行数据转换为串行数据，反之亦然。

接收期间，将输入至串行输入引脚的数据转换为并行数据。

发送数据时，将设置到该寄存器的值转换为串行数据从串行输出引脚输出。

移位寄存器不能直接通过程序操作。

使用串行数据寄存器 0nL (SDR0nL) 的低 8 位对于移位寄存器进行读取或写入。

10.2.2 串行数据寄存器 0n (SDR0nH, SDR0nL)

SDR0nH 和 SDR0nL 寄存器是通道 n 的发送/接收数据寄存器 (8 位)。SDR00H 和 SDR01L 作为发送/接收缓冲寄存器，SDR00H 和 SDR01L 寄存器的位 7 至位 1 作为设置工作时钟(f_{MCK}, f_{CLK})分频比的寄存器。

接收数据时，由移位寄存器转换的并行数据被存储于 SDR00H 和 SDR01L 寄存器。发送数据时，把传送至移位寄存器的发送值设置到 SDR00H 和 SDR01L 寄存器。

依据串行通信操作设置寄存器 0n (SCR0nL) 位 0 (DLS0n0) 的设置，存储于 SDR00H 和 SDR01L 寄存器的数据的长度如下所示 (与数据输出顺序无关)：

- 7 位数据长度 (存储于 SDR0nL 寄存器的位 0 至位 6)
- 8 位数据长度 (存储于 SDR0nL 寄存器的位 0 至位 7)

可以以 8 位为单位读取或写入 SDR0nH 和 SDR0nL 寄存器。

依据通信模式，SDR0n 寄存器能够以 8 位为单位以以下 SFR 名称读取或写入。但是，要注意停止操作(SE0n = 0) 时，禁止以 8 位为单位写入。

- CSIp 通信时： SIOp (CSIp 数据寄存器)
- UART0 接收期间： RXD0 (UARTq 接收数据寄存器)
- UARTq 发送期间： TXD0 (UARTq 发送数据寄存器)

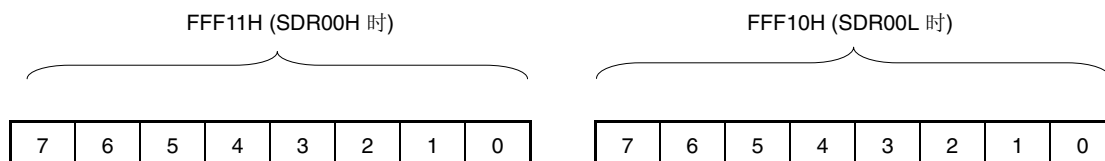
产生复位信号后，SDR0nH 和 SDR0nL 寄存器被清除为 00H。

备注 1. 完成数据接收后，未被重写结果的位 0 至 7 将保持值 0。

2. n: 通道编号 (n = 0, 1), p: CSI 编号 (p = 00), q: UART 编号 (q = 0)

图 10-2. 串行数据寄存器 0n (SDR0nH, SDR0nL)的格式 (n = 0, 1)

地址: FFF10H (SDR00L), FFF11H (SDR00H), FFF12H (SDR01L), FFF13H (SDR01H) 复位后: 00H R/W



备注 关于 SDR0nH 寄存器的高 7 位功能, 请参阅 **10.3 控制串行阵列单元的寄存器**。

10.3 控制串行阵列单元的寄存器

串行阵列单元由下列寄存器控制。

- 外围允许寄存器 0 (PER0)
- 串行时钟选择寄存器 0 (SPS0)
- 串行模式寄存器 0n (SMR0nH, SMR0nL)
- 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)
- 串行数据寄存器 0n (SDR0nH, SDR0nL)
- 串行标志清除触发寄存器 0n (SIR0n)
- 串行状态寄存器 0n (SSR0n)
- 串行通道开始寄存器 0 (SS0)
- 串行通道停止寄存器 0 (ST0)
- 串行通道允许状态寄存器 0 (SE0)
- 串行输出允许寄存器 0 (SOE0)
- 串行输出电平寄存器 0 (SOL0)
- 串行输出寄存器 0 (SO0)
- 串行时钟输出寄存器(CKO0)
- 噪声滤波器允许寄存器 0 (NFEN0)
- 输入切换控制寄存器(ISC)
- 端口输出模式寄存器 0 (POM0)
- 端口模式控制寄存器 0 (PMC0)
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)

备注 n: 通道编号 (n = 0, 1)

10.3.1 外围允许寄存器 0 (PER0)

PER0 用于允许或禁止向外围硬件供应时钟。通过停止向未使用的硬件供应时钟，可以降低功耗和噪声。

当使用串行阵列单元 0 时，必须将该寄存器的位 2 (SAU0EN) 设置为 1。

使用 1 位或 8 位存储器操作指令设置 PER0 寄存器。

产生复位信号后，PER0 寄存器被清除为 00H。

图 10-3. 外围允许寄存器 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	0	0	ADCEN	0	0	SAU0EN	0	TAU0EN

SAU0EN	串行阵列单元 0 输入时钟供应的控制
0	停止输入时钟的供应。 - 不可写入用于串行阵列单元 0 的 SFR。 - 串行阵列单元 0 处于复位状态。
1	允许输入时钟的供应。 可以读取和写入用于串行阵列单元 0 的 SFR。

注意事项 1. 设置串行阵列单元 0 时，必须在 SAU0EN 位被设置为 1 的状态下先设置以下的寄存器。如果 SAU0EN = 0，对串行阵列单元 0 的控制寄存器进行的写入操作无效，而且寄存器的读取值总为初始值（噪声滤波器允许寄存器 0 (NFEN0)、输入切换控制寄存器(ISC)、端口输出模式寄存器 0 (POM0)、端口模式寄存器 0 (PM0)、端口模式控制寄存器 0 (PMC0)以及端口寄存器 0 (P0)除外）。

- 串行时钟选择寄存器0 (SPS0)
- 串行模式寄存器0n (SMR0nH, SMR0nL)
- 串行通信操作设置寄存器0n (SCR0nH, SCR0nL)
- 串行数据寄存器0n (SDR0nH, SDR0nL)
- 串行标志清除触发寄存器0n (SIR0n)
- 串行状态寄存器0n (SSR0n)
- 串行通道开始寄存器0 (SS0)
- 串行通道停止寄存器0 (ST0)
- 串行通道允许状态寄存器0 (SE0)
- 串行输出允许寄存器0 (SOE0)
- 串行输出电平寄存器0 (SOL0)
- 串行输出寄存器0 (SO0)
- 串行时钟输出寄存器(CKO0)

2. 必须将不确定位清除为 0。

10.3.2 串行时钟选择寄存器 0 (SPS0)

SPS0 寄存器为 8 位寄存器，用于选择供应给各个通道共通的两种工作时钟(CK00, CK01)。通过 SPS0 寄存器的位 7 至位 4 选择 CK01，位 3 至位 0 选择 CK00。

当寄存器在运行（当 SE0n = 1）时，禁止改写 SPS0 寄存器。

使用 8 位存储器操作指令设置 SPS0 寄存器。

产生复位信号后，SPS0 寄存器被清除为 00H。

图 10-4. 串行时钟选择寄存器 0 (SPS0)的格式

地址: F0126H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
SPS0	PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000

PRS On3	PRS On2	PRS On1	PRS On0		工作时钟(CK00, CK01)的选择 ^注				
					f _{CLK} = 1.25 MHz	f _{CLK} = 2.5 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz
0	0	0	0	f _{CLK}	1.25 MHz	2.5 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	f _{CLK} /2	625 kHz	1.25 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	f _{CLK} /2 ²	313 kHz	625 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	f _{CLK} /2 ³	156 kHz	313 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	f _{CLK} /2 ⁴	78 kHz	156 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	f _{CLK} /2 ⁵	39 kHz	78 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	f _{CLK} /2 ⁶	19.5 kHz	39 kHz	78 kHz	156 kHz	313 kHz
0	1	1	1	f _{CLK} /2 ⁷	9.8 kHz	19.5 kHz	39 kHz	78 kHz	156 kHz
1	0	0	0	f _{CLK} /2 ⁸	4.9 kHz	9.8 kHz	19.5 kHz	39 kHz	78 kHz
1	0	0	1	f _{CLK} /2 ⁹	2.5 kHz	4.9 kHz	9.8 kHz	19.5 kHz	39 kHz
1	0	1	0	f _{CLK} /2 ¹⁰	1.22 kHz	2.5 kHz	4.9 kHz	9.8 kHz	19.5 kHz
1	0	1	1	f _{CLK} /2 ¹¹	625 Hz	1.22 kHz	2.5 kHz	4.9 kHz	9.8 kHz
1	1	0	0	f _{CLK} /2 ¹²	313 Hz	625 Hz	1.22 kHz	2.5 kHz	4.9 kHz
1	1	0	1	f _{CLK} /2 ¹³	152 Hz	313 Hz	625 Hz	1.22 kHz	2.5 kHz
1	1	1	0	f _{CLK} /2 ¹⁴	78 Hz	152 Hz	313 Hz	625 Hz	1.22 kHz
1	1	1	1	f _{CLK} /2 ¹⁵	39 Hz	78 Hz	152 Hz	313 Hz	625 Hz

注 要更改 f_{CLK} 的时钟选择（通过更改系统时钟控制寄存器(CKC)的值）时，必须在停止串行阵列单元(SAU)操作（串行通道停止寄存器 0 (ST0) = 03H）之后才可以执行。

备注 1. f_{CLK}: CPU/外围硬件时钟频率

2. n = 0

10.3.3 串行模式寄存器 0n (SMR0nH, SMR0nL)

SMR0nH 和 SMR0nL 寄存器用于设置通道 n 的操作模式。也用于选择工作时钟(f_{MCK})，指定是否可以输入串行时钟(f_{SCK})，设置开始触发、操作模式 (CSI 或 UART) 和中断源。该寄存器还用于仅在 UART 模式下反相接收数据的电平。

当寄存器在运行 (当 SE0n = 1) 时，禁止改写 SMR0nH 和 SMR0nL 寄存器。但是，可以在操作过程中改写 MD0n0 位。

使用 8 位存储器操作指令设置 SMR0nH 和 SMR0nL 寄存器。
产生复位信号后，SMR0nH 和 SMR0nL 寄存器分别被设置为 00H 和 20H。

图 10-5. 串行模式寄存器 0n (SMR0nH)的格式

地址: F0111H (SMR00H), F0113H (SMR01H)
复位后: 00H R/W
符号: SMR0nH

7	6	5	4	3	2	1	0
CKS 0n	CCS 0n	0	0	0	0	0	STS 0n [※]

CKS 0n	通道 n 工作时钟(f _{MCK})的选择
0	由 SPS0 寄存器设置的工作时钟 CK00
1	由 SPS0 寄存器设置的工作时钟 CK01
工作时钟(f _{MCK})用于边沿检测电路。另外，根据 CCS0n 位和 SDR0nH 寄存器的设置，产生传送时钟(f _{TCLK})。	

CCS 0n	通道 n 的传送时钟(f _{TCLK})的选择
0	由 CKS0n 位指定的分频工作时钟 f _{MCK}
1	来自 SCKp 引脚的输入时钟 f _{SCK} (CSI 模式的从传送)
传送时钟 f _{TCLK} 用于移位寄存器、通信控制电路、输出控制电路、中断控制电路和错误控制电路。当 CCS0n = 0 时，使用 SDR0nH 寄存器的高 7 位设置工作时钟(f _{MCK})的分频比。	

STS 0n [※]	开始触发源的选择
0	仅限软件触发有效 (CSI 和 UART 发送时选择)。 CSI 从属时，将 SCK 输入的有效边沿作为开始触发
1	RxD0 引脚的有效沿 (UART 接收时选择)
将 SS0 寄存器设置为 1 并且满足以上条件时，开始传送。	

注 仅限 SMR01H 寄存器。SMR00H 寄存器中请设置为 0。

注意事项 请勿改变不定位的值(固定为 0 或 1)。

备注 n: 通道编号 (n = 0, 1)

图 10-6. 串行模式寄存器 0n (SMR0nL)的格式

地址: F0110H (SMR00L), F0112H (SMR01L)
复位后: 20H R/W
符号: SMR0nL

7	6	5	4	3	2	1	0
0	SIS 0n0 注	1	0	0	0	MD 0n1	MD 0n0

SIS 0n0 注	UART 模式下，通道 n 接收数据的电平反相的控制
0	下降沿被检测为起始位。 按原样捕捉被输入的通信数据。
1	上升沿被检测为起始位。 将被输入的通信数据反相并捕捉。

MD 0n1	通道 n 操作模式的设置
0	CSI 模式
1	UART 模式

MD 0n0	通道 n 中断源的选择
0	传送结束中断
1	缓冲器空中断 (当数据从 SDR0nL 寄存器传送至移位寄存器时。)
连续发送时，将 MD0n0 位设置为 1，并且在 SDR0nL 的数据为空时，写入下一个发送数据。	

注 仅限 SMR01L 寄存器。SMR00L 寄存器中请设置为 0。

注意事项 请勿改变不定位的值(固定为 0 或 1)。

备注 n: 通道编号 (n = 0, 1)

10.3.4 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)

SCR0nH 和 SCR0nL 寄存器是通道 n 的通信操作设置寄存器。该寄存器用于设置数据发送/接收模式、数据和时钟相位、是否屏蔽错误信号、奇偶检验位、起始位、停止位和数据长度。

当寄存器在运行（当 SE0n = 1）时，禁止改写 SCR0nH 和 SCR0nL 寄存器。

使用 8 位存储器操作指令设置 SCR0nH 和 SCR0nL 寄存器。

产生复位信号后，SCR0nH 和 SCR0nL 寄存器分别被设置为 00H 和 87H。

图 10-7. 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)的格式 (1/2)

地址: F0119H (SCR00H) , F011BH (SCR01H)

复位后: 00H R/W

符号: SCR0nH

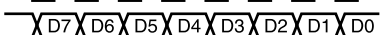
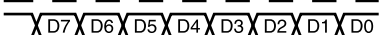
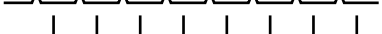
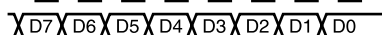
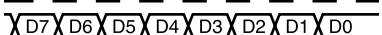
地址: F0118H (SCR00L) , F011AH (SCR01L)

复位后: 87H R/W

符号: SCR0nL

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
TXE	RXE	DAP	CKP	0	EOC	PTC	PTC	DIR	0	SLC0	SLC	0	1	1	DLS
0n	0n	0n	0n		0n	0n1	0n0	0n		n1 ^{※1}	0n0				0n0

TXE0n	RXE0n	通道 n 操作模式的设置
0	0	禁止通信
0	1	仅接收
1	0	仅发送
1	1	发送/接收

DAP0n	CKP0n	CSI 模式下数据和时钟相位的选择	类型
0	0	SCK00  SO00  SIO0 输入时序 	1
0	1	SCK00  SO00  SIO0 输入时序 	2
1	0	SCK00  SO00  SIO0 输入时序 	3
1	1	SCK00  SO00  SIO0 输入时序 	4
UART 模式下必须设置为 DAP0n, CKP0n = 0, 0。			

EOC0n	选择是否屏蔽错误中断信号(INTSREx (x = 0))
0	屏蔽错误中断 INTSREx (不屏蔽 INTSRx)。
1	允许产生错误中断 INTSREx (如果出现错误, INTSRx 将被屏蔽)。
CSI 模式、UART 发送期间设置为 EOC0n = 0 ^{※2} 。	

注 1. 仅限 SCR00L 寄存器。

2. 不清除 EOC0n 就使用 CSI0n 的情况下, 有可能产生错误中断 INTSREn。

注意事项 必须将 SCR0nL 寄存器的位 2 和位 1 设置为 1, 将 SCR0nH 和 SCR0nL 寄存器的不定位清除为 0。

备注 n: 通道编号 (n = 0, 1)

图 10-7. 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)的格式 (2/2)

地址: F0119H (SCR00H) , F011BH (SCR01H)

复位后: 00H R/W

符号: SCR0nH

地址: F0118H (SCR00L) , F011AH (SCR01L)

复位后: 87H R/W

符号: SCR0nL

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
TXE	RXE	DAP	CKP	0	EOC	PTC	PTC	DIR	0	SLC0	SLC	0	1	1	DLS
0n	0n	0n	0n		0n	0n1	0n0	0n		n1 ^{注1}	0n0				0n0

PTC 0n1	PTC 0n0	UART 模式下奇偶校验位的设置	
		发送	接收
0	0	不输出奇偶校验位	无奇偶校验的接收
0	1	输出零奇偶校验 ^{注2} 。	无奇偶校验判断
1	0	输出偶数校验	偶数校验判断
1	1	输出奇数校验	奇数校验判断

CSI 模式下必须设置为 PTC0n1, PTC0n0 = 0, 0。

DIR0n	CSI 和 UART 模式下数据传送序列的选择
0	MSB 优先的输入/输出数据
1	LSB 优先的输入/输出数据

SLC0n 1 ^{注1}	SLC0n 0	UART 模式下停止位的设置
0	0	无停止位
0	1	停止位长度 = 1 位
1	0	停止位长度 = 2 位 (仅限 n = 0)
1	1	禁止设置

选择传送结束中断时, 传送完所有停止位时产生中断。
 UART 接收时将停止位长度设置为 1 位(SLC0n1, SLC0n0 = 0,)。
 CSI 模式时设置为无停止位(SLC0n1, SLC0n0 = 0, 0)。

DLS0n0	CSI 和 UART 模式下数据长度的设置
0	7 位数据长度 (存储于 SDR0nL 寄存器的位 0 至位 6)
1	8 位数据长度 (存储于 SDR0nL 寄存器的位 0 至位 7)

注 1. 仅限 SCR00L 寄存器。

2. 不论数据内容如何, 始终附加 0。

注意事项 必须将 SCR0nL 寄存器的位 2 和位 1 设置为 1, 将 SCR0nH 和 SCR0nL 寄存器的不定位清除为 0。

备注 n: 通道编号 (n = 0, 1)

10.3.5 串行数据寄存器 0n (SDR0nH, SDR0nL)

SDR0nH 和 SDR0nL 寄存器是通道 n 的发送/接收数据寄存器。SDR0nH 寄存器用作设置工作时钟(f_{MCK})分频比的寄存器。SDR0nL 寄存器用作发送和接收的缓冲器寄存器。

SMR0nH 寄存器的 CCS0n 位被设置为 0 时, 通过 SDR0nH[7:1]设置工作时钟的分频时钟作为传送时钟。CCS0n 位被设置为 1 时, SDR0nH[7:1]固定为 00000000B。使用 SCKp 引脚的输入时钟 f_{SCK} (CSI 模式的从属传送) 作为传送时钟。

SDR0n 寄存器用作发送/接收缓冲寄存器。接收过程中，由移位寄存器转换的并行数据被存储于 **SDR0n** 寄存器，在发送过程中，发送至移位寄存器的数据将被设置到 **SDR0n** 寄存器。

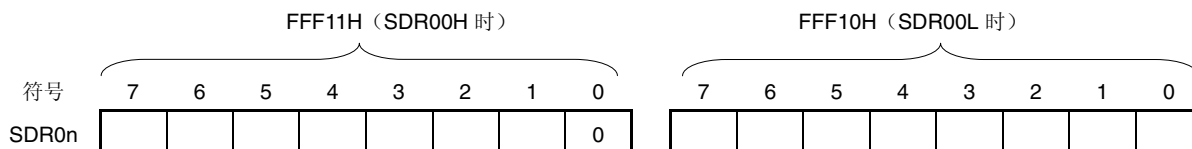
可以以 8 位为单位读取或写入 SDR0nH 和 SDR0nL 寄存器。

但是，仅在停止操作时($SE0n = 0$)可以写入和读取 $SDR0nH[7:1]$ 。操作过程中(当 $SE0n = 1$)写入 $SDR0nH$ 和 $SDR0nL$ 寄存器时，向 $SDR0nH$ 寄存器的写入操作无效，可以写入 $SDR0nL$ 寄存器。操作过程中($SE0n = 1$)， $SDR0nH$ 寄存器的读取值总为 0 值。

产生复位信号后，SDR0nH 和 SDR0nL 寄存器被清除为 00H。

图 10-8. 串行数据寄存器 0n (SDR0n) 的格式

地址: FFF10H (SDR00L), FFF11H (SDR00H) 复位后: 00H R/W
FFF12H (SDR01L), FFF13H (SDR01H)

[illegible]

注意事项 使用 **UART** 时，禁止设置为 **SDR0nH[7:1] = 0000000B** 至 **0000001B**。将 **SDR0nH[7:1]** 设置为 **0000010B** 或更大。

使用 CSI 时，禁止设置为 SDR00H[7:1] = (0000000B)。

备注 1. 关于 SDR0nL 寄存器的功能, 请参阅 10.2 串行阵列单元的配置。

- 2. n: 通道编号 ($n = 0, 1$)**

10.3.6 串行标志清除触发寄存器 0n (SIR0n)

SIR0n 寄存器是用于清除通道 n 各个错误标志的触发寄存器。

当该寄存器的各位 (FECT0n、PECT0n、OVCT0n) 被设置为 1 时, 串行状态寄存器 0n (SSR0n)的相应位 (FEF0n、PEF0n、OVF0n) 将被清除为 0。由于 SIR0n 寄存器是一种触发寄存器, 当 SSR0n 寄存器的相应位被清零时, 它们会被立即清零。

使用 8 位存储器操作指令设置 SIR0n 寄存器。

产生复位信号后, SIR0n 寄存器被清除为 00H。

图 10-9. 串行标志清除触发寄存器 0n (SIR0n)的格式

地址: F0108H (SIR00) , F010AH (SIR01) , 复位后: 00H R/W

符号:	7	6	5	4	3	2	1	0
SIR0n	0	0	0	0	0	FECT0n ^注	PECT0n	OVCT0n

FECT0n ^注	通道 n 的帧错误标志的清除触发
0	不清除
1	将 SSR0n 寄存器的 FEF0n 位清除为 0。

PECT0n	通道 n 的奇偶校验错误标志的清除触发
0	不清除
1	将 SSR0n 寄存器的 PEF0n 位清除为 0。

OVCT0n	通道 n 的溢出错误标志的清除触发
0	不清除
1	将 SSR0n 寄存器的 OVF0n 位清除为 0。

注 仅限 SIR01 寄存器。

注意事项 必须将不定位设置为 0。

- 备注 1. n: 通道编号 (n = 0, 1)
2. SIR0n 寄存器的读取值总是为 00H。

10.3.7 串行状态寄存器 0n (SSR0n)

SSR0n 寄存器用于指示通道 n 的通信状态和错误发生状态。该寄存器指示的错误为帧错误、奇偶校验错误和溢出错误。

使用 8 位存储器操作指令读取 SSR0n 寄存器。

产生复位信号后，SSR0n 寄存器被清除为 00H。

图 10-10. 串行状态寄存器 0n (SSR0n)的格式(1/2)

地址: F0100H (SSR00) , F0102H (SSR01) , 复位后: 00H R

符号:	7	6	5	4	3	2	1	0
SSR0n	0	TSF0n	BFF0n	0	0	FEF0n*	PEF0n	OVF0n

TSF0n	通道 n 的通信状态显示标志
0	通信停止或暂停。
1	通信正在进行。
<清除条件>	
• ST0 寄存器的 ST0n 位被设置为 1 (通信停止)或者 SS0 寄存器的 SS0n 位被设置为 1(通信暂停)。	
• 通信结束。	
<设置条件>	
• 通信开始。	

BFF0n	通道 n 的缓冲寄存器状态显示标志
0	有效数据没有保存在 SDR0nL 寄存器中。
1	有效数据保存在 SDR0nL 寄存器中。
<清除条件>	
• 发送时，从 SDR0nL 寄存器至移位寄存器的发送数据的传送结束。	
• 接收时，来自 SDR0nL 寄存器的接收数据的读取结束。	
• ST0 寄存器的 ST0n 位被设置为 1 (通信停止)或者 SS0 寄存器的 SS0n 位被设置为 1(通信允许)。	
<设置条件>	
• 当 SCR0nH 寄存器的 TXE0n 位被设置为 1（各种通信模式下的发送或者发送和接收模式）时，发送数据被写入 SDR0nL 寄存器。	
• 当 SCR0nH 寄存器的 RXE0n 位被设置为 1（各种通信模式下的接收或者发送和接收模式）时，接收数据被存储于 SDR0nL 寄存器中。	
• 发生接收错误。	

注 仅限 SSR01 寄存器。

注意事项 如果在 BFF0n = 1 时将数据写入 SDR0nL 寄存器，存储于该寄存器中的发送/接收数据将被丢弃，并检测到溢出错误(OVF0n = 1)。

备注 n: 通道编号 (n = 0, 1)

图 10-10. 串行状态寄存器 0n (SSR0n)的格式(2/2)

地址: F0100H (SSR00), F0102H (SSR01), 复位后: 00H R

符号:	7	6	5	4	3	2	1	0
SSR0n	0	TSF0n	BFF0n	0	0	FEF0n [※]	PEF0n	OVF0n

FEF0n [※]	通道 n 的帧错误检测标志
0	无错误发生。
1	错误发生 (UART 接收期间)。
<清除条件> - 将 1 写入 SIR0n 寄存器的 FECT0n 位。 <设置条件> - 当 UART 接收结束时未检测到停止位。	

PEF0n	通道 n 的奇偶校验错误检测标志
0	无错误发生。
1	奇偶校验错误发生 (UART 接收期间)。
<清除条件> - 将 1 写入 SIR0n 寄存器的 PECT0n 位。 <设置条件> - 当 UART 接收结束时, 发送数据的奇偶校验和奇偶校验位不匹配(奇偶校验错误)。	

OVF0n	通道 n 的溢出错误检测标志
0	无错误发生。
1	错误发生。
<清除条件> - 将 1 写入 SIR0n 寄存器的 OVCT0n 位。 <设置条件> - 当 SCR0nH 寄存器的 RXE0n 被设置为 1 (各种通信模式下的接收或者发送和接收模式) 时, 即使接收数据存储于 SDR0nL 寄存器中, 也不会读取这些数据, 而是写入发送数据或下一接收数据。 - CSI 模式的从发送或者发送和接收时, 尚未准备好发送数据。	

注 仅限 SSR01 寄存器。

备注 n: 通道编号 (n = 0, 1)

10.3.8 串行通道开始寄存器 0 (SS0)

SS0 寄存器是对各个通道进行设置以允许开始通信/计数的触发寄存器。

当将 1 写入该寄存器(SS0n)的一位时，串行通道允许状态寄存器 0 (SE0)的对应位(SE0n) 将被设置为 1（允许操作）。由于 SS0n 位为触发位，所以当 SE0n = 1 时，该位将立即被清除为 0。

使用 8 位存储器操作指令设置 SS0 寄存器。

产生复位信号后，SS0 寄存器被清除为 00H。

图 10-11. 串行通道开始寄存器 0 (SS0)的格式

地址: F0122H (SS0) 复位后: 00H R/W

符号:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01	SS00

SS0n	通道 n 的操作开始触发
0	无触发操作。
1	将 SE0n 位设置为 1，进入通信等待状态 ^注 。

注 如果在传送操作时 SS0n 被设置为 1，则传送停止且接口进入等待状态。此时，控制寄存器和移位寄存器、SCK0n 和 SO0n 引脚、以及 FEF0n、PEF0n 和 OVF0n 标志保持原值。

- 注意事项 1.** 必须将不确定位清除为 0。
- 2.** 对于 UART 接收，将 SCR0nH 寄存器的 RXE0n 位设置为 1，然后确保在经过 4 个或更多 f_{MCK} 时钟之后将 SS0n 设置为 1。

- 备注 1.** n: 通道编号 (n = 0, 1)
- 2.** SS0 寄存器的读取值总是为 00H。

10.3.9 串行通道停止寄存器 0 (ST0)

ST0 寄存器是对各个通道进行设置以允许停止通信/计数的触发寄存器。

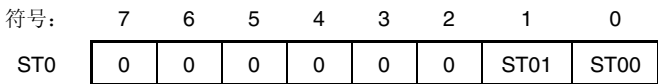
当将 1 写入该寄存器(ST0n)的一位时，串行通道允许状态寄存器 0 (SE0)的对应位(SE0n) 将被设置为 0（停止操作）。由于 ST0n 位为触发位，所以当 SE0n = 0 时，该位将立即被清除为 0。

使用 1 位或 8 位存储器操作指令设置 ST0 寄存器。

产生复位信号后，ST0 寄存器被清除为 00H。

图 10-12. 串行通道结束寄存器 0 (ST0)的格式

地址: F0124H 复位后: 00H R/W



ST0n	通道 n 的操作停止触发
0	无触发操作。
1	将 SE0n 位清除为 0，停止通信操作 [※]

注 控制寄存器和移位寄存器、SCK0n 和 SO0n 引脚、以及 FEF0n、PEF0n 和 OVF0n 标志保持原值。

注意事项 必须将不确定位清除为 0。

- 备注 1. n: 通道编号 (n = 0, 1)
2. ST0 寄存器的读取值总是为 00H。

10.3.10 串行通道允许状态寄存器 0 (SE0)

SE0 寄存器用于确认各个通道的串行发送/接收操作处于允许或禁止状态。

当将 1 写入串行通道开始寄存器 0 (SS0)的一个位时，该寄存器的对应位将被设置为 1。当将 1 写入串行通道停止寄存器 0 (ST0)的一个位时，该寄存器的对应位将被清除为 0。

对于允许操作的通道 n，不能通过软件改写串行时钟输出寄存器 0 (CKO0)的 CKO0n 位的值（通道 n 的串行时钟输出），通信操作反映的值由串行时钟引脚输出。

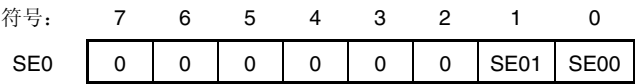
对于禁止操作的通道 n，可以通过软件设置 CKO0 寄存器的 CKO0n 位的值，并从串行时钟引脚输出该值。这样，任何波形（如开始条件/停止条件的波形）均可由软件创建。

使用 1 位或 8 位存储器操作指令可以读取 SE0 寄存器。

产生复位信号后，SE0 寄存器被清除为 00H。

图 10-13. 串行通道允许状态寄存器 0 (SE0)的格式

地址: F0120H (SE0) 复位后: 00H R



SE0n	表示通道 n 的操作允许或禁止状态
0	禁止操作(停止)。
1	允许操作。

备注 n: 通道编号 (n = 0, 1)

10.3.11 串行输出允许寄存器 0 (SOE0)

SOE0 寄存器用于设置是否允许或禁止各个通道的串行通信操作的输出。

对于允许串行输出的通道 n，不能通过软件改写串行输出寄存器 0 (SO0)的 SO0n 位的值，通信操作反映的值由串行数据输出引脚输出。

对于停止串行输出的通道 n，可以通过软件设置 SO0 寄存器的 SO0n 位的值，并从串行数据输出引脚输出该值。这样，任何波形（如开始条件/停止条件的波形）均可由软件创建。

使用 1 位或 8 位存储器操作指令设置 SOE0 寄存器。

产生复位信号后，SOE0 寄存器被清除为 00H。

图 10-14. 串行输出允许寄存器 0 (SOE0)的格式

地址: F012AH (SOE0)	复位后: 00H	R/W								
符号:	7	6	5	4	3	2	1	0		
SOE0	0	0	0	0	0	0	0	0	SOE00	

SOE0n	通道 n 的串行输出允许/禁止
0	通过串行通信操作来禁止输出。
1	允许串行通信操作输出。

注意事项 必须将不确定位清除为 0。

备注 n: 通道编号 (n = 0)

10.3.12 串行输出寄存器 0 (SO0)

SO0 寄存器是用于各个通道串行输出的缓冲寄存器。

该寄存器的 SO0n 位的值从通道 n 的串行数据输出引脚输出。

该寄存器的 SO0n 位只有在串行输出被禁止时 (SOE0n = 0) 才可用软件改写。当允许串行输出(SOE0n = 1)时，用软件改写无效，且该寄存器的值只能通过串行通信操作更改。

要将串行接口引脚用作端口功能引脚，须将相应的 SO0n 位设置为 1。

使用 8 位存储器操作指令设置 SO0 寄存器。

产生复位信号后，SO0 寄存器被清除为 03H。

图 10-15. 串行输出寄存器 0 (SO0)的格式

地址: F0128H (SO0) 复位后: 03H R/W

符号: 7 6 5 4 3 2 1 0

SO0

0	0	0	0	0	0	1	SO00

SO0n	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。

备注 n: 通道编号 (n = 0)

10.3.13 串行时钟输出寄存器(CKO0)

CKO0 寄存器是用于各个通道串行时钟输出的缓冲寄存器。

该寄存器的 CKO0n 位的值从通道 n 的串行时钟输出引脚输出。

该寄存器的 CKO0n 位只有在串行输出被禁止时 (SE0n = 0) 才可用软件改写。当允许通道操作(SE0n = 1)时，用软件改写无效，且该寄存器的值只能通过串行通信操作更改。

要将串行接口引脚用作端口功能引脚，须将相应的 CKO0n 位设置为 1。

使用 8 位存储器操作指令设置 CKO0 寄存器。

产生复位信号后，CKO0 寄存器被清除为 03H。

图 10-16. 串行时钟输出寄存器(CKO0)的格式

地址: F0129H (CKO0) 复位后: 03H R/W

符号: 7 6 5 4 3 2 1 0

CKO0

0	0	0	0	0	0	0	1	CKO00

CKO0n	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。

备注 n: 通道编号 (n = 0)

10.3.14 串行输出电平寄存器 0 (SOL0)

SOL0 寄存器是用于设置各个通道数据输出电平反相的寄存器。

仅限 UART 模式时可以设置该寄存器。CSI 模式时，必须将对应位设置为 0。

仅当允许串行输出时 (SOE0n = 1)，通过该寄存器对通道 n 所作的反相设置才会体现在引脚输出上。当串行输出被禁止时 (SOE0n = 0)，SO0n 位的值将按原样输出。

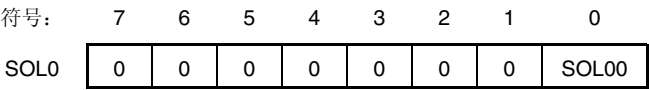
运行 (SE0n = 1) 时，禁止改写 SOL0 寄存器。

使用 8 位存储器操作指令设置 SOL0 寄存器。

产生复位信号后，SOL0 寄存器被清除为 00H。

图 10-17. 串行输出电平寄存器 0 (SOL0)的格式

地址: F0134H (SOL0) 复位后: 00H R/W



SOL0n	UART 模式时，通道 n 的发送数据电平反相的选择
0	通信数据按原样输出。
1	通信数据反相并输出。

注意事项 必须将不确定位清除为 0。

备注 n: 通道编号 (n = 0)

10.3.15 噪声滤波器允许寄存器 0 (NFEN0)

NFEN0 寄存器用来设置噪声滤波器是否可用于各通道串行数据输入引脚的输入信号。

用于 CSI 通信的引脚中，将该寄存器的对应位清除为 0，禁止使用噪声滤波器。

用于 UART 通信的引脚中，将该寄存器的对应位设置为 1，允许使用噪声滤波器。

噪声滤波器启用时，和目标通道的工作时钟(f_{MCK})进行同步后，执行 2 个时钟的一致检测。噪声滤波器关闭时，仅执行和目标通道工作时钟(f_{MCK})的同步。

使用 1 位或 8 位存储器操作指令设置 NFEN0 寄存器。

产生复位信号后，NFEN0 寄存器被清除为 00H。

图 10-18. 噪声滤波器允许寄存器 0 (NFEN0)的格式

地址: F0070H

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	0	0	SNFEN00

SNFEN00	RxD0(RxD0/P01)引脚的噪声滤波器的使用
0	噪声滤波器关闭
1	噪声滤波器启用

作为 RxD0 引脚使用时，请设置 SNFEN00 为 1。

作为 RxD0 引脚以外使用时，请清除 SNFEN00 为 0。

注意事项 必须将不定位清除为 0。

10.3.16 输入切换控制寄存器(ISC)

ISC 寄存器的 ISC1 和 ISC0 位用于在 UART0 接收过程中修正波特率时，实现与外部中断及定时器阵列单元的联合。

当位 0 被设置为 1 时，串行数据输入(RxD0)引脚的输入信号将被选为外部中断输入(INTP0)，可用于检测 INTP0 中断形式的唤醒信号。

当位 1 被设置为 1 时，串行数据输入(RxD0)引脚的输入信号将被选为定时器输入，可在检测到唤醒信号后测量脉冲宽度以修正波特率。

使用 1 位或 8 位存储器操作指令设置 ISC 寄存器。

产生复位信号后，ISC 寄存器被清除为 00H。

图 10-19. 输入切换控制寄存器(ISC)的格式

地址: F00730H 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的通道 1							
0	选择 TI01 引脚的输入信号作为定时器输入（普通操作）							
1	选择 RxD0 引脚的输入信号作为定时器输入 (检测唤醒信号并测量脉冲宽度以修正波特率)							
ISC0	切换外部中断(INTP0)							
0	选择 INTP0 引脚的输入信号作为外部中断输入（普通操作）							
1	选择 RxD0 引脚的输入信号作为外部中断输入（检测唤醒信号）							

注意事项 必须将不确定位清除为 0。

10.3.17 端口输出模式寄存器 0 (POM0)

该寄存器以 1 位为单位设置端口 1 的输出模式。

另外，用 POM0 寄存器以及 PMxx、PMCxx 和 PUxx 寄存器设定是否使用片上上拉电阻（请参阅 4.3.3 上拉电阻选择寄存器 0, 4, 12 (PU0, PU4, PU12)）。

使用 1 位或 8 位存储器操作指令设置 POM0 寄存器。

产生复位信号后，POM0 寄存器被清除为 00H。

图 10-20. 端口输出模式寄存器 0 (POM0)的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM0	0	0	0	0	0	0	0	POM00	F0050H	00H	R/W

POM0n	P0n 引脚输出缓冲器的选择(n = 0)														
0	普通输出模式														
1	N 沟开漏输出(V _{DD} 耐压)模式														

10.3.18 端口模式寄存器 0 (PM0)

该寄存器以 1 位为单位设置端口 1 的输入/输出。

将复用于串行数据输出引脚或串行时钟输出引脚的端口（例如 P02/ANI1/ $\overline{\text{SCK00}}$ /PCLBUZ0/KR3），用作串行数据输出或串行时钟输出时，将与各端口对应的端口模式寄存器(PM0)位和端口模式控制寄存器(PMC0)位设置为 0。此时将与各端口对应的端口寄存器(P0)位设置为 1。

示例：将 P02/ANI1/ $\overline{\text{SCK00}}$ /PCLBUZ0/KR3 用于串行时钟输出时

将端口模式控制寄存器 0 的 PMC02 位设为 0。

将端口模式寄存器 0 的 PM02 位设为 0。

将端口寄存器 0 的 P02 位设为 1。

将复用于串行数据输入引脚或串行时钟输入引脚的端口（例如 P02/ANI1/ $\overline{\text{SCK00}}$ /PCLBUZ0/KR3），用作串行数据输入或串行时钟输入时，将与各端口对应的端口模式寄存器(PM0)位设置为 1。并将与各端口对应的端口模式控制寄存器(PMC0)位设置为 0。这时可以将端口寄存器(P0)位设置为 0 或 1。

示例：将 P02/ANI1/ $\overline{\text{SCK00}}$ /PCLBUZ0/KR3 用于串行数据输入或串行时钟输入时

将端口模式控制寄存器 0 的 PMC02 位设为 0。

将端口模式寄存器 0 的 PM02 位设置为 1。

将端口寄存器 1 的 P02 位设置为 0 或 1。

使用 8 位存储器操作指令设置 PM0 寄存器。

产生复位信号后，PM0 寄存器被设置为 FFH。

图 10-21. 端口模式寄存器 0 (PM0)的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W

PM0n	选择 P0n 引脚的输入/输出模式 (n = 0 至 4)
0	输出模式 (输出缓冲器启用)
1	输入模式 (输出缓冲器关闭)

10.4 操作停止模式

串行阵列单元的各串行接口均有操作停止模式。

在此模式下，不能执行串行通信，因而可降低功耗。

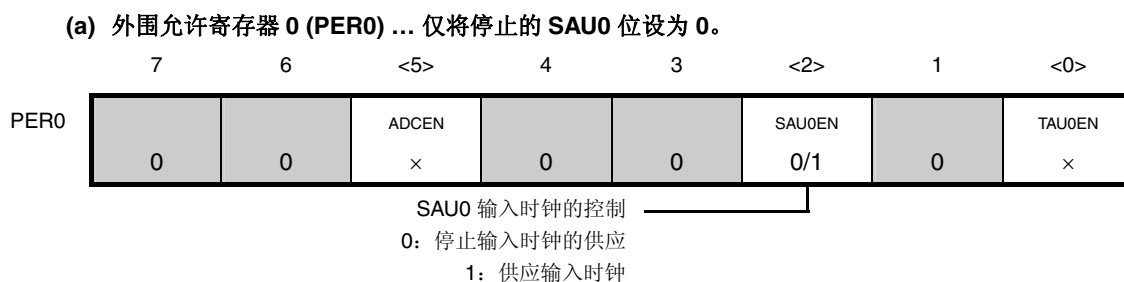
另外，在该模式下，串行接口功能复用引脚可用作端口功能引脚。

10.4.1 以单元单位停止操作

使用外围允许寄存器 0 (PER0) 设置按照单元单位停止操作。

PER0 寄存器用于允许或禁止向外围硬件供应时钟。通过停止向未使用的硬件供应时钟，可以降低功耗和噪声。

图 10-22. 以单元单位停止操作时外围允许寄存器 0 (PER0) 的设置



注意事项 1. 如果 **SAU0EN = 0**，对串行阵列单元 0 的控制寄存器进行的写入操作无效，而且寄存器的读取值总为初始值。

但如下的寄存器除外：

- 输入切换控制寄存器(ISC)
- 噪声滤波器允许寄存器 0 (NFEN0)
- 端口输出模式寄存器 0 (POM0)
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)

2. 必须将不确定位清除为 0。

备注 ：禁止设置 (由硬件固定设置)

x：串行阵列单元未使用的位（取决于其他外围功能的设置）

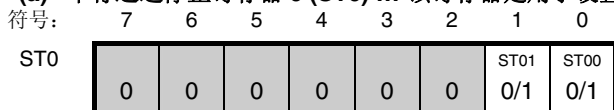
0/1：可根据用户的用途设为 0 或 1

10.4.2 以通道单位停止操作

使用以下各种寄存器设置通道单位停止操作。

图 10-23. 以通道单位停止操作时每个寄存器的设置

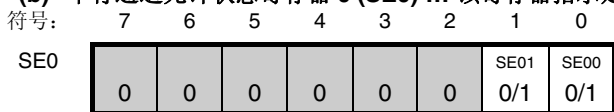
(a) 串行通道停止寄存器 0 (ST0) ... 该寄存器是用于设置是否允许各个通道的通信/计数的触发寄存器。



1: 将SE0n清除为0并且停止通信操作

* 由于ST0n位为触发位, 所以当SE0n = 0时, 该位将立即被清除为0。

(b) 串行通道允许状态寄存器 0 (SE0) ... 该寄存器指示是否允许或停止各个通道的数据发送/接收操作。

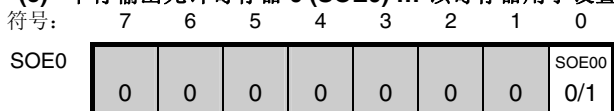


0: 停止操作

* SE0 寄存器为只读状态寄存器。通过设置 ST0 寄存器来停止操作。

对于已经停止操作的通道, 可以通过软件设置 SO0 寄存器的 CKO0n 位的值。

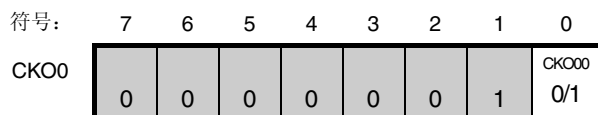
(c) 串行输出允许寄存器 0 (SOE0) ... 该寄存器用于设置允许或停止各个通道的串行通信操作的输出。



0: 使用串行通信操作来停止输出

* 对于通道 n (其串行输出已停止), 可以用软件设置 SO0 寄存器的 SO0n 位的值。

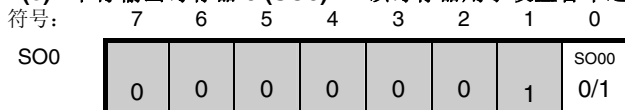
(d) 串行时钟输出寄存器 0 (CKO0) ... 该寄存器是各个通道的串行输出的缓冲寄存器。



1: 串行时钟输出值为1

* 在将对应于各个通道的引脚用作端口功能引脚时, 将相应的 CKO0n 位设置为 1。

(e) 串行输出寄存器 0 (SO0) ... 该寄存器用于设置各个通道的串行输出值。



1: 串行数据输出值为1

* 在将对应于各个通道的引脚用作端口功能引脚时, 将相应的 SO0n 位设置为 1。

备注 1. n: 通道编号 (n = 0, 1)

2. : 禁止设置(固定为初始值), 0/1: 可根据用户的用途设为 0 或 1

10.5 3 线串行输入/输出(CSI00)通信的操作

这是一种采用三线的时钟同步式通信功能：串行时钟线(SCK) 和串行数据线 (SI 和 SO)。

[数据的发送/接收]

- 数据长度为 7 或 8 位
- 发送/接收数据的相控制
- MSB/LSB 选择
- 发送/接收数据的电平设置

[时钟控制]

- 输入/输出时钟的相控制
- 通过预分频器和通道内部计数器设置传送周期
- 最大传送速率

从通信时：Max. $f_{CLK}/6$ [※]

[中断功能]

- 传送结束中断/缓冲器空中断

[错误检测标志]

- 溢出错误

注 在满足 SCK 周期时间(t_{KCY})特性的范围内使用时钟(参阅第二十一章 电特性)。

单元	通道	用作 CSI	用作 UART
0	0	CSI00	UART0
	1	—	

3 线串行输入/输出 (CSI00)具有如下七种类型的通信操作。

- 主发送(参阅 10.5.1)
- 主接收(参阅 10.5.2)
- 主发送/接收(参阅 10.5.3)
- 从发送(参阅 10.5.4)
- 从接收(参阅 10.5.5)
- 从发送/接收(参阅 10.5.6)

10.5.1 主发送

主发送指 R7F0C80112ESP, R7F0C80212ESP 输出传送时钟，并将数据发送至另一个器件。

3线串行输入/输出	CSI00
目标通道	SAU0的通道0
使用引脚	SCK00, SO00
中断	INTCSI00 可选择传送结束中断(单传送模式时)或缓冲器空中断(连续传送模式时)
错误检测标志	无
传送数据长度	7或8位
传送速率	Max. $f_{CLK}/4$ [Hz] (SDR0nH[7:1] = 1以上) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] ^注
数据相位	可通过SCR0nH寄存器的DAP0n位选择 • DAP0n = 0: 从开始串行时钟操作时开始数据输出。 • DAP0n = 1: 从开始串行时钟操作的半个时钟前开始数据输出。
时钟相位	可通过SCR0nH寄存器的CKP0n位选择 • CKP0n = 0: 不反相 (在SCK的下降沿输出数据，在其上升沿输入数据) • CKP0n = 1: 反相 (在SCK的上升沿输出数据，在其下降沿输入数据)
数据方向	MSB或LSB优先

注 在满足以上条件及电特性中的外围功能特性(参阅第二十一章 电特性)的范围内使用此操作。

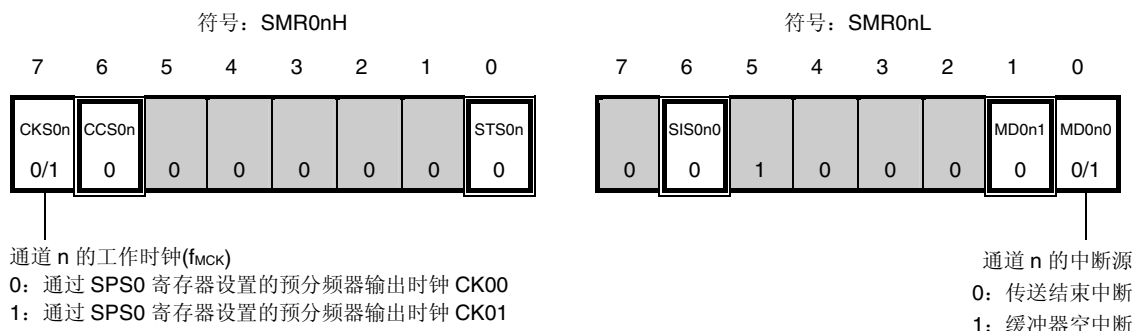
备注 1. f_{CLK} : 系统时钟频率

2. $n = 0$

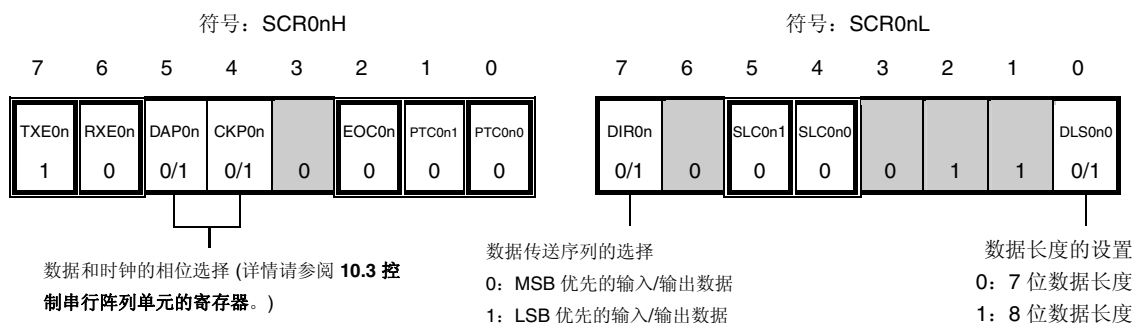
(1) 寄存器设置

图 10-24. 3 线串行输入/输出 (CSI00)的主发送时的寄存器设置内容示例(1/2)

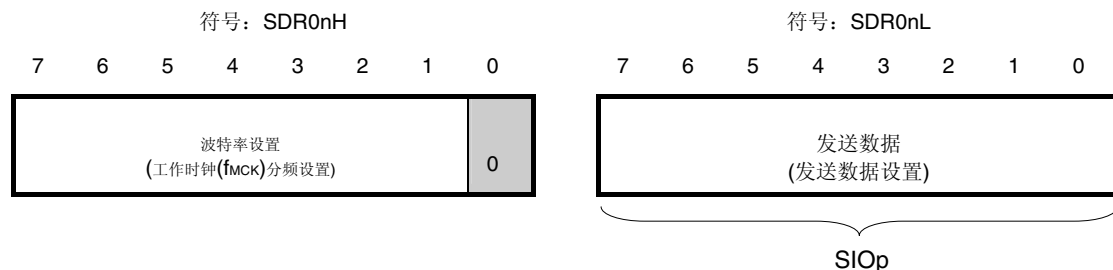
(a) 串行模式寄存器 0n (SMR0nH, SMR0nL)



(b) 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)



(c) 串行数据寄存器 0n (SDR0nH, SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) ... 仅对目标通道的位进行设置。

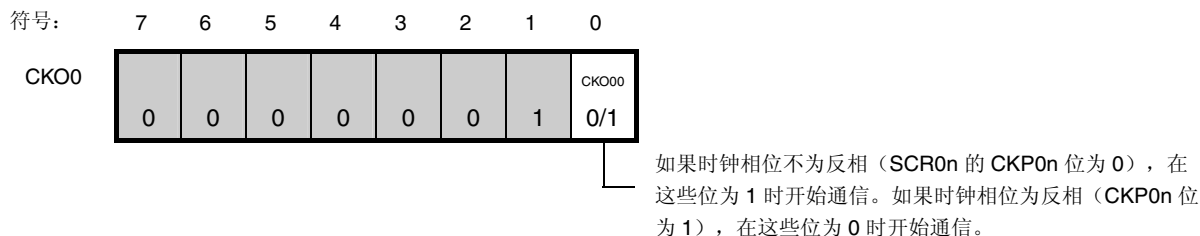


图 10-24. 3 线串行输入/输出 (CSI00)的主发送时的寄存器设置内容示例(2/2)

(e) 串行输出寄存器 0 (SO0) ... 仅对目标通道的位进行设置。

符号: 7 6 5 4 3 2 1 0

SO0								SO00
	0	0	0	0	0	0	1	0/1

(f) 串行输出允许寄存器 0 (SOE0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SOE0								SOE00
	0	0	0	0	0	0	0	0/1

(g) 串行通道开始寄存器 0 (SS0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SS0								SS01	SS00
	0	0	0	0	0	0	0	0	0/1

备注 1. n = 0, p: CSI 编号(p = 00)

2. ☐: 设置固定于 CSI 主机发送模式, ☐: 禁止设置 (设为初始值)

x: 此模式下不能使用的位 (任意模式下均不使用时设为初始值)

0/1: 可根据用户的用途设为 0 或 1

(2) 操作步骤

图 10-25. 主发送的初始设置步骤

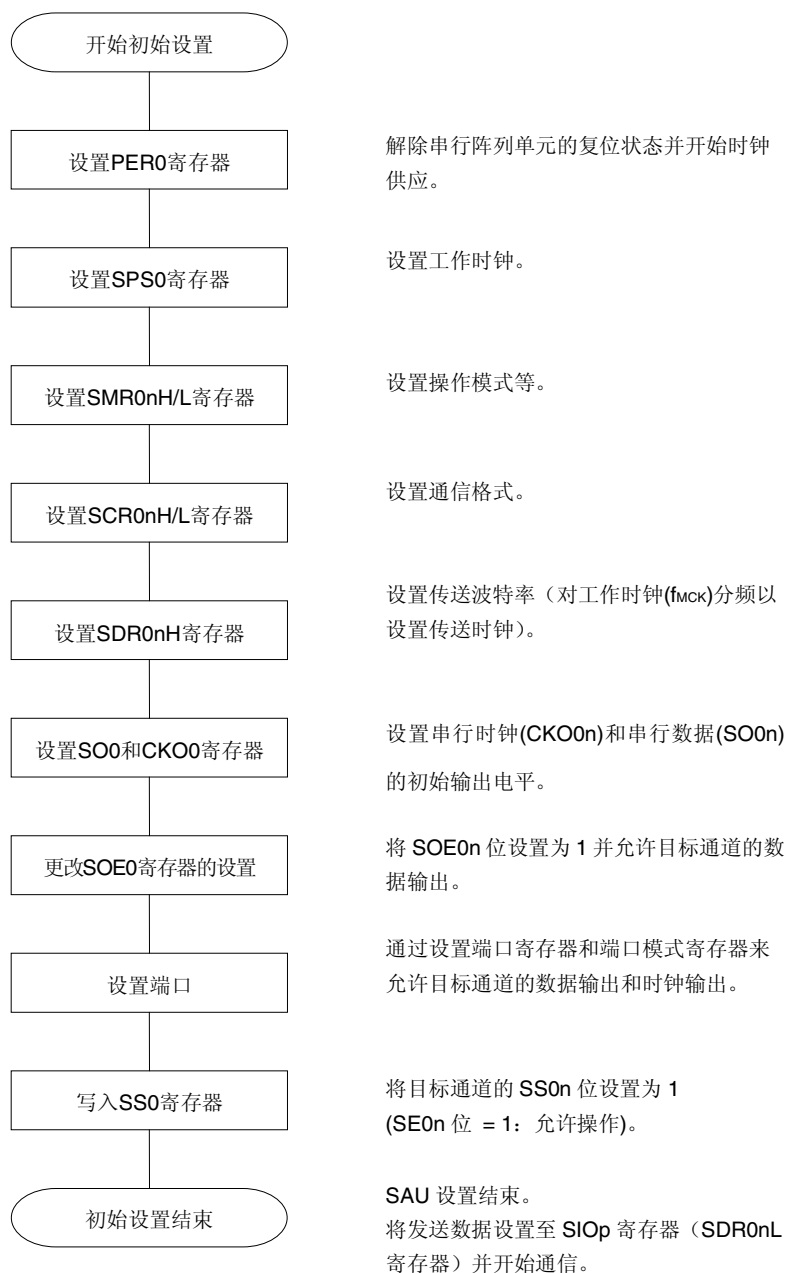


图 10-26. 停止主发送的步骤

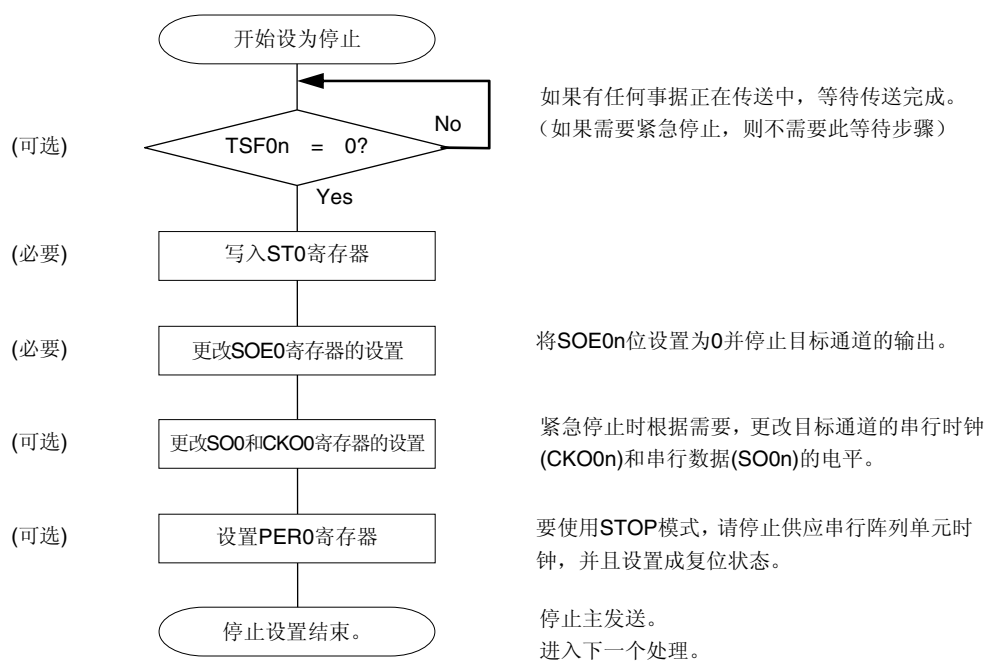


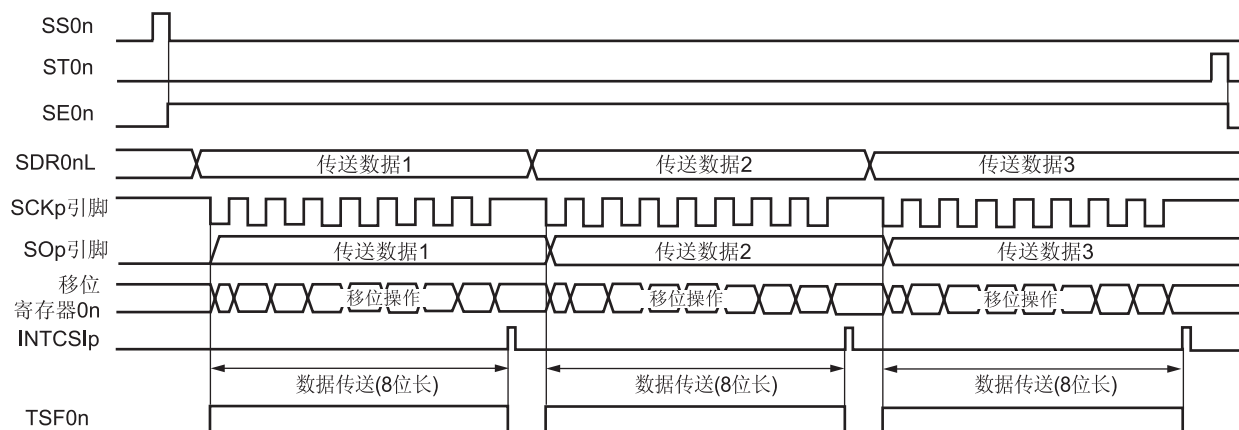
图 10-27. 恢复主发送的步骤



备注 在停止主发送的步骤中改写 PER0 而停止时钟供应时,请等待通信目标(从设备)停止或者通信结束,然后执行初始设置,而不是重传设置。

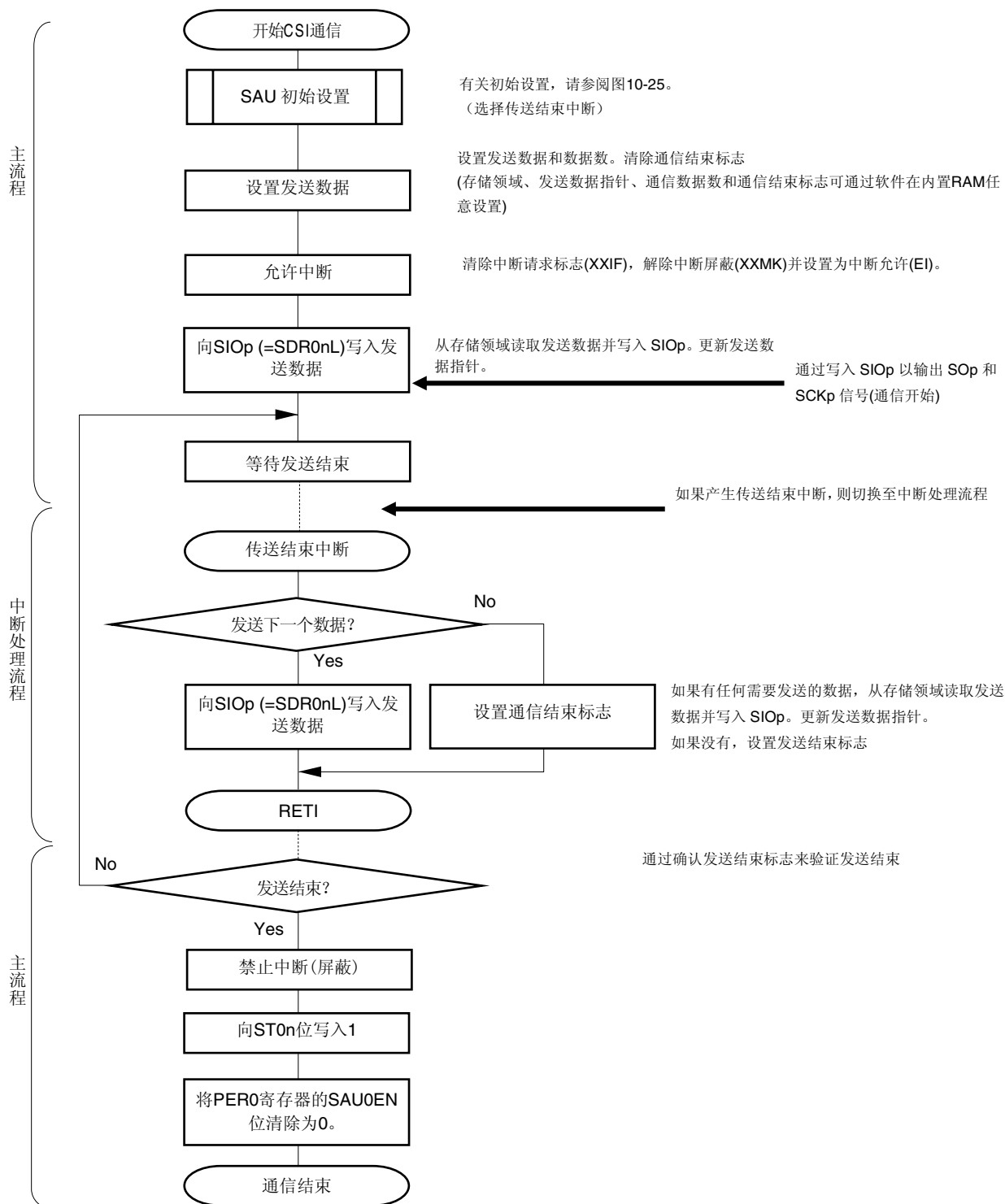
(3) 处理流程 (单发送模式时)

图 10-28. 主发送(单发送模式时)的时序图
(类型 1: DAP0n = 0, CKP0n = 0)



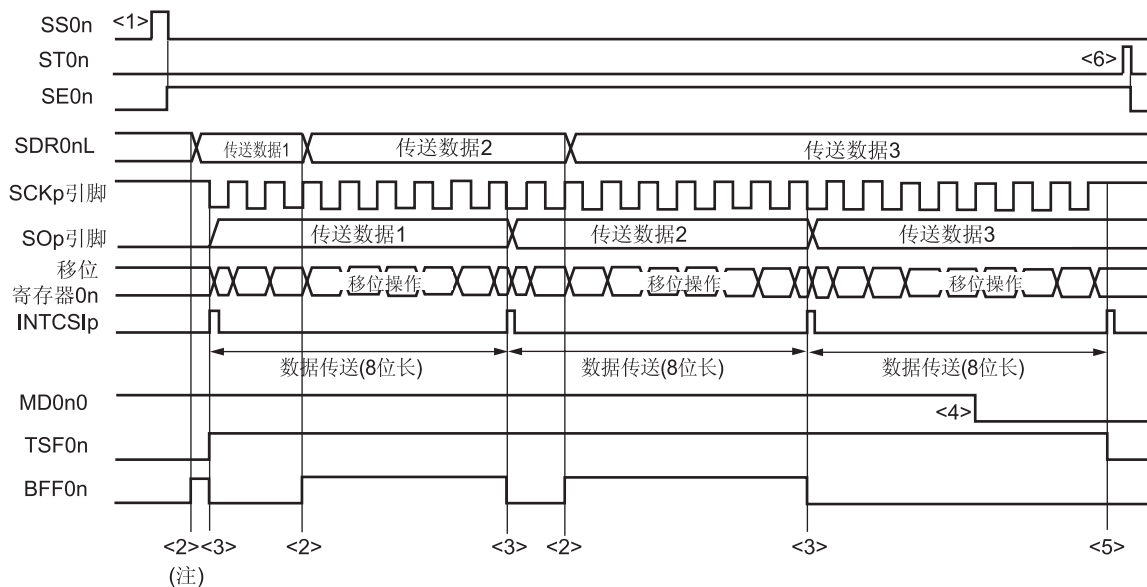
备注 n = 0, p: CSI 编号(p = 00)

图 10-29. 主发送 (单发送模式时)的流程图



(4) 处理流程 (连续发送模式时)

图 10-30. 主发送(连续发送模式时)的时序图
(类型 1: DAP0n = 0, CKP0n = 0)



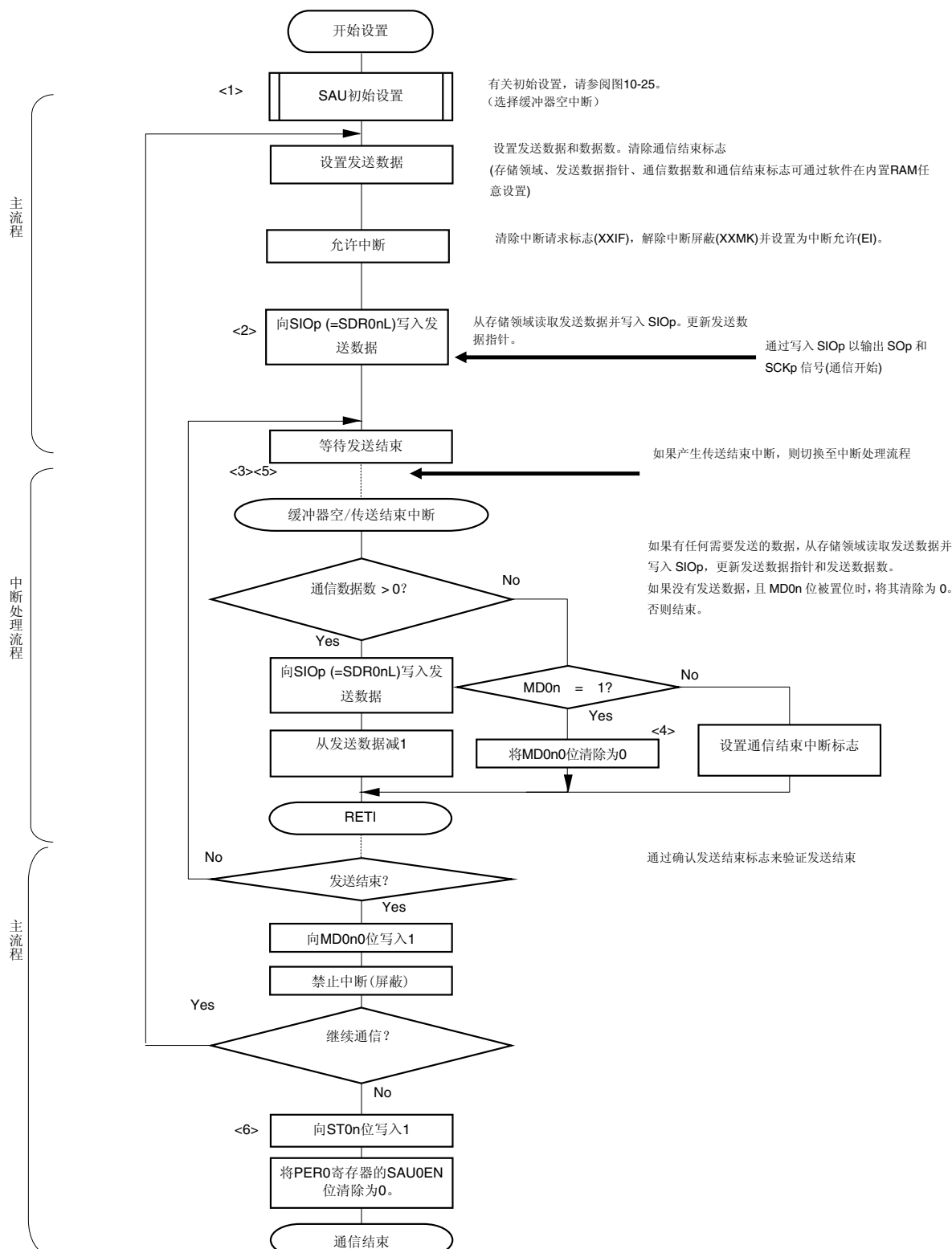
注 如果在串行状态寄存器 0n (SSR0n) 的 BFF0n 位为 1 时将发送数据写入 SDR0nL 寄存器 (将有效数据存储于串行数据寄存器 0n (SDR0nL) 中), 则发送数据将被覆盖。

注意事项 即使在操作过程中, 也可改写串行模式寄存器 0n (SMR0nL) 的 MD0n0 位。

但是必须在开始传送最后一位之前改写, 以便在最后发送数据的传送结束中断前完成改写。

备注 n = 0, p: CSI 编号(p = 00)

图 10-31. 主发送 (连续发送模式时)的流程图



备注 此图中的<1>至<6>对应于图 10-30. 主发送 (连续发送模式时)的时序图中的<1>至<6>。

10.5.2 主接收

主接收指 R7F0C80112ESP, R7F0C80212ESP 输出传送时钟，并从其他器件接收数据。

3线串行输入/输出	CSI00
目标通道	SAU0的通道0
使用引脚	SCK00, SI00
中断	INTCSI00 可选择传送结束中断(单传送模式时)或缓冲器空中断(连续传送模式时)
错误检测标志	仅限溢出错误检测标志 (OVF0n)
传送数据长度	7或8位
传送速率	Max. $f_{CLK}/4$ [Hz] (SDR0nH[7:1] = 1以上) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] ^{注2}
数据相位	可通过SCR0nH寄存器的DAP0n位选择 • DAP0n = 0: 从开始串行时钟操作时开始数据输入。 • DAP0n = 1: 从开始串行时钟操作的半个时钟前开始数据输入。
时钟相位	可通过SCR0nH寄存器的CKP0n位选择 • CKP0n = 0: 正相 • CKP0n = 1: 反相
数据方向	MSB或LSB优先

注 在满足以上条件及电特性中的 AC 特性(参阅第二十一章 电特性)的范围内使用此操作。

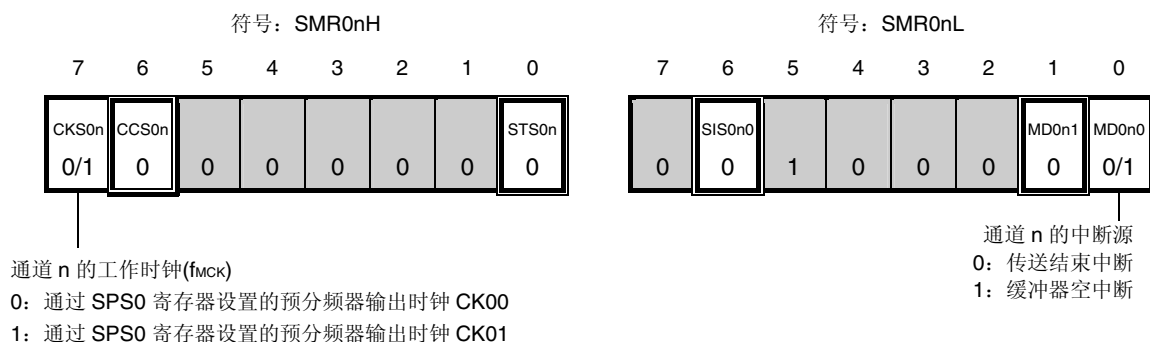
备注 1. f_{CLK} : 系统时钟频率

2. $n = 0$

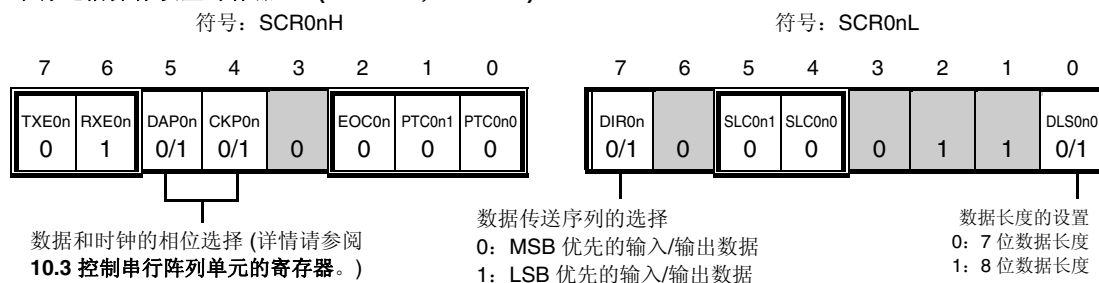
(1) 寄存器设置

图 10-32. 3 线串行输入/输出(CSI00)的主接收时的寄存器设置内容示例(1/2)

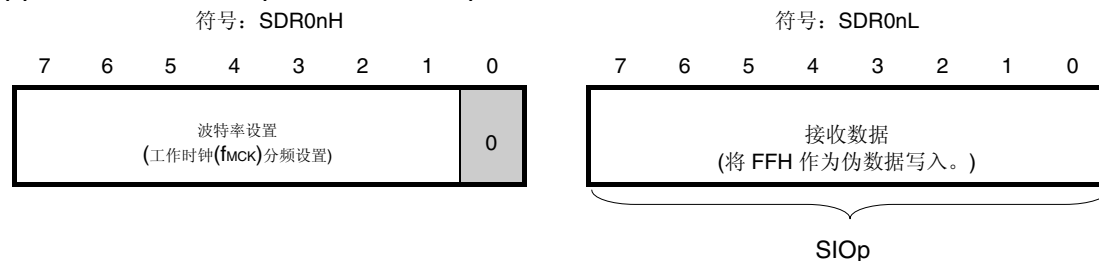
(a) 串行模式寄存器 0n (SMR0nH, SMR0nL)



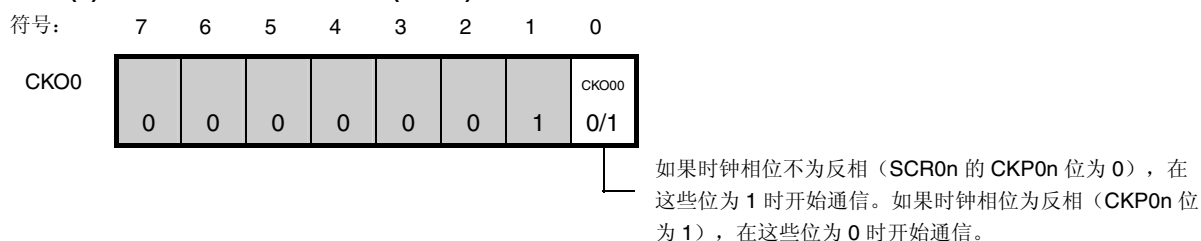
(b) 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)



(c) 串行数据寄存器 0n (SDR0nH, SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) ... 仅对目标通道的位进行设置。



(e) 串行输出寄存器 0 (SO0) ... 此模式下不使用的寄存器。

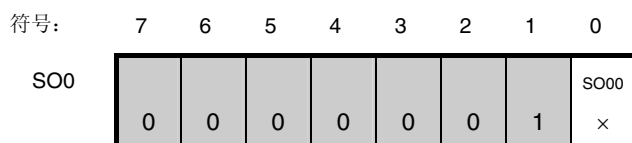


图 10-32. 3 线串行输入/输出(CSI00)的主接收时的寄存器设置内容示例(2/2)

(f) 串行输出允许寄存器 0 (SOE0) ... 此模式下不使用的寄存器。

符号:	7	6	5	4	3	2	1	0
SOE0								SOE00
	0	0	0	0	0	0	0	x

(g) 串行通道开始寄存器 0 (SS0) ... 仅将目标通道的位设置为 1。

符号:	7	6	5	4	3	2	1	0
SS0							SS01	SS00
	0	0	0	0	0	0	0	0/1

备注 1. n = 0, p: CSI 编号(p = 00)

2. ☐ : 设置固定于 CSI 主发送模式, ☐ : 禁止设置 (设为初始值)
 x: 此模式下不能使用的位 (任意模式下均不使用时设置为初始值)
 0/1: 可根据用户的用途设置为 0 或 1

(2) 操作步骤

图 10-33. 主接收的初始设置步骤

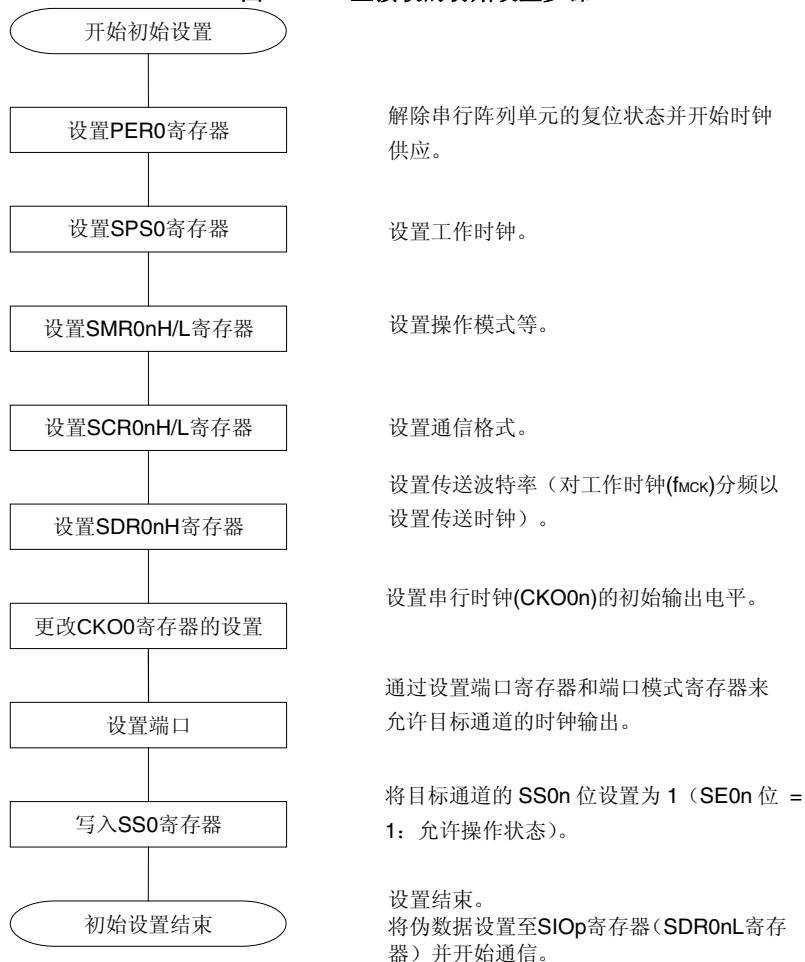


图 10-34. 停止主接收的步骤

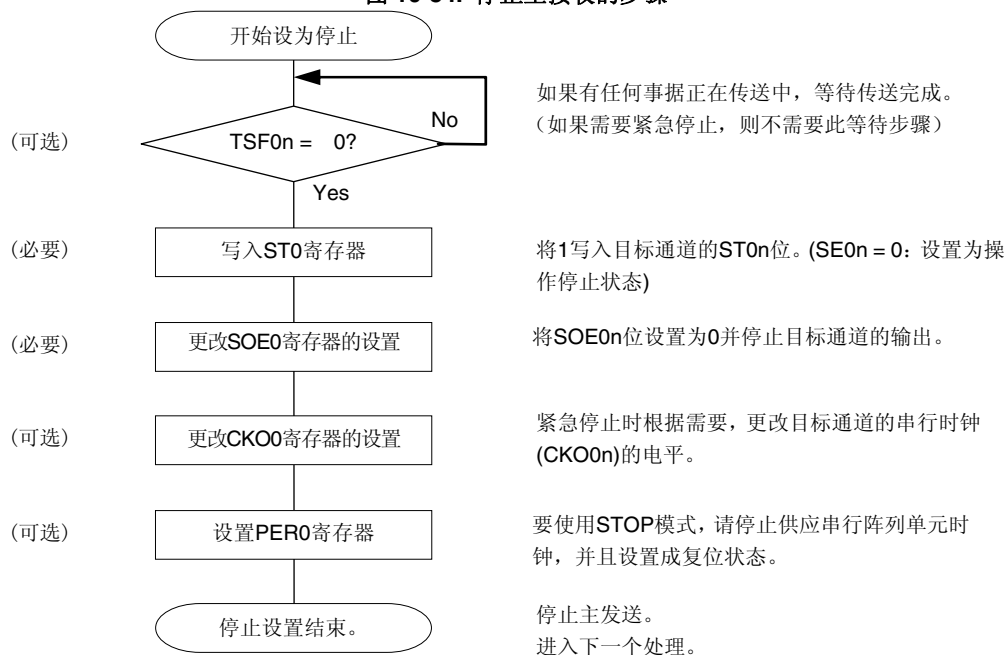
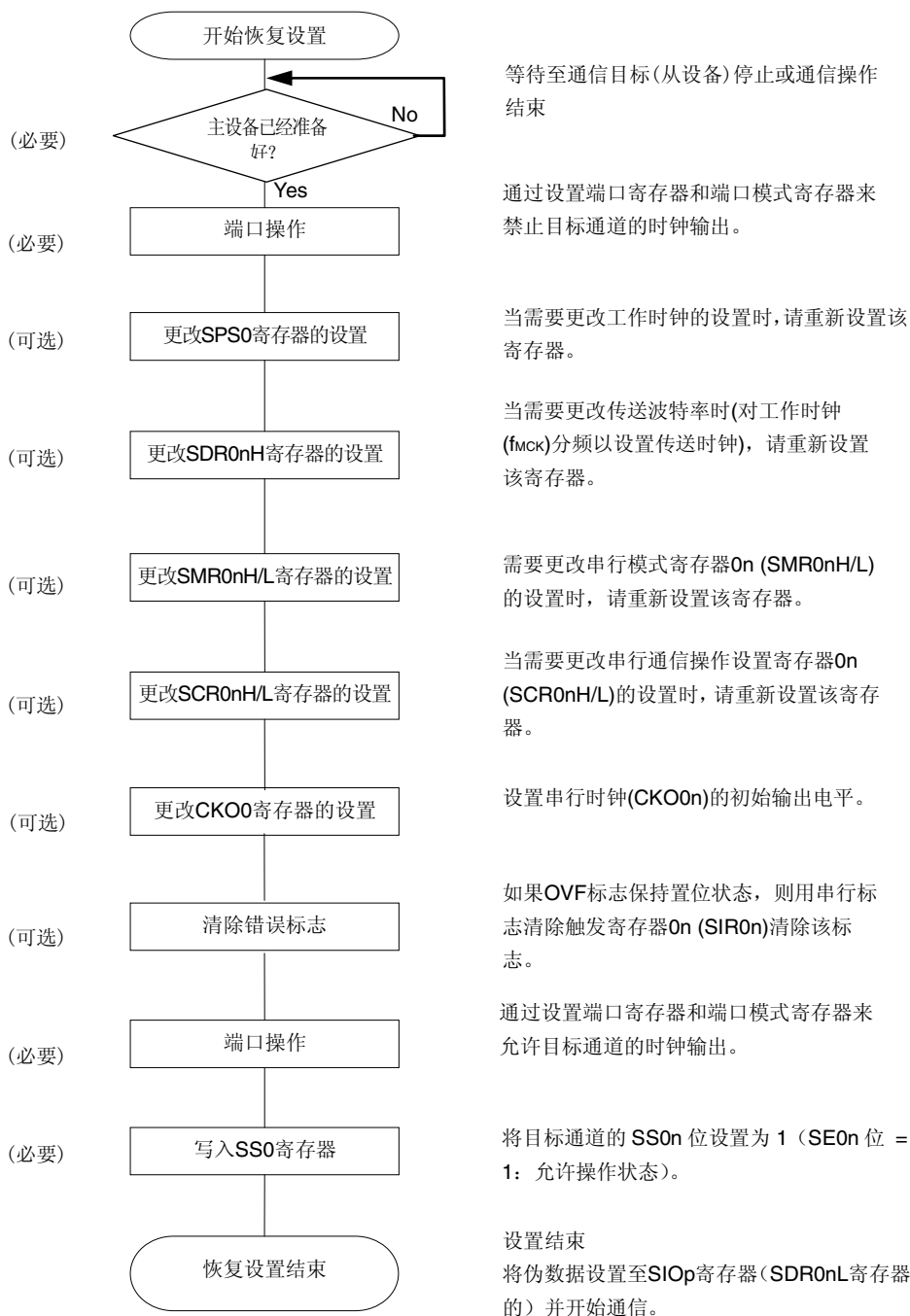


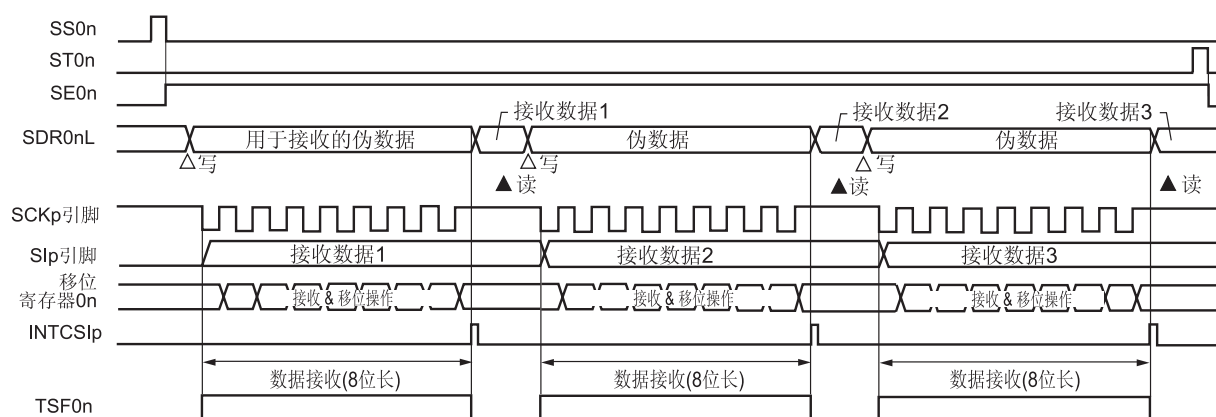
图 10-35. 恢复主接收的步骤



备注 在停止主发送的步骤中改写 PER0 而停止时钟供应时,请等待通信目标(从设备)停止或者通信结束,然后执行初始设置,而不是重传设置。

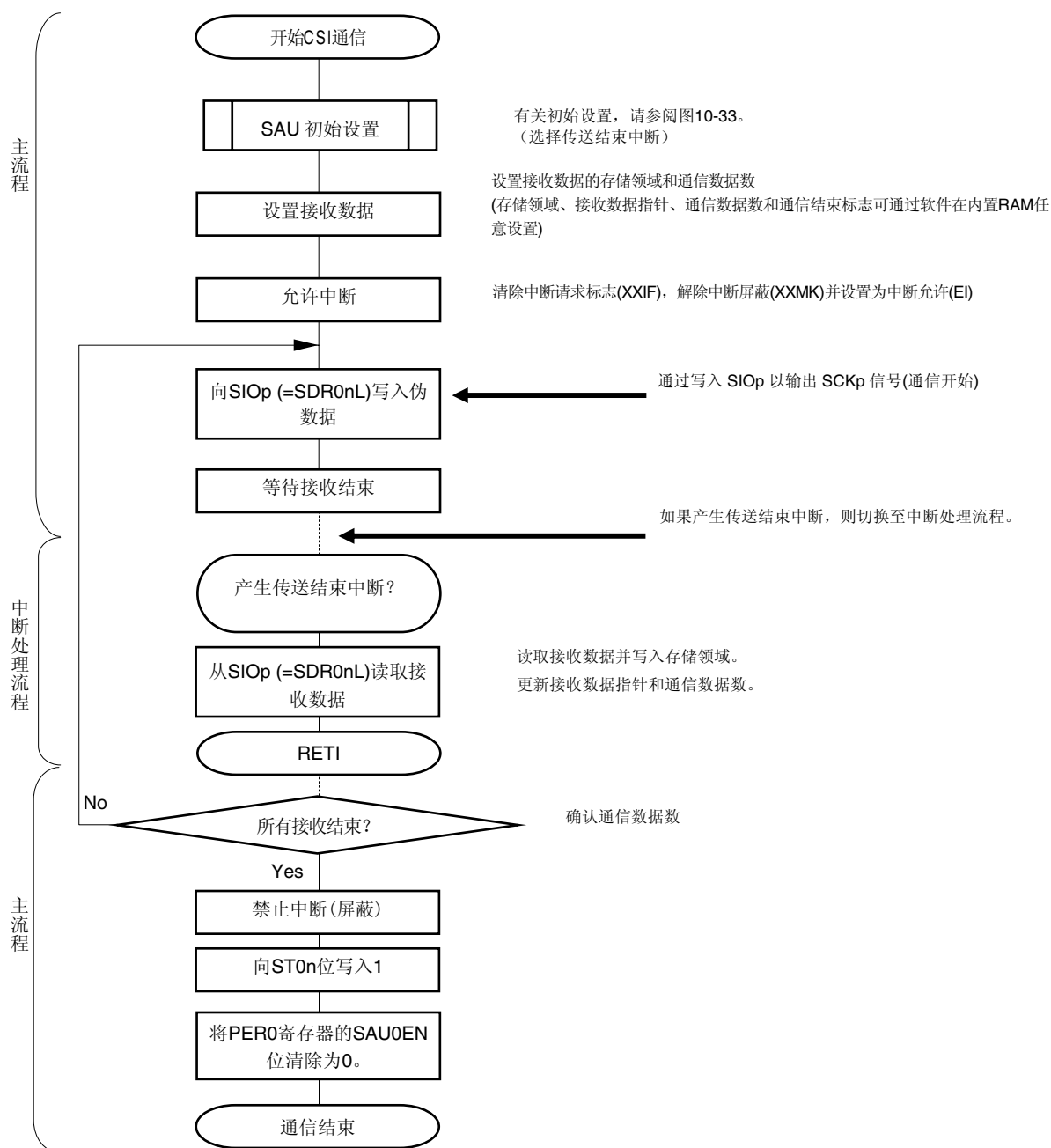
(3) 处理流程 (单接收模式时)

图 10-36. 主接收(单接收模式时)的时序图(类型 1: DAP0n = 0, CKP0n =)



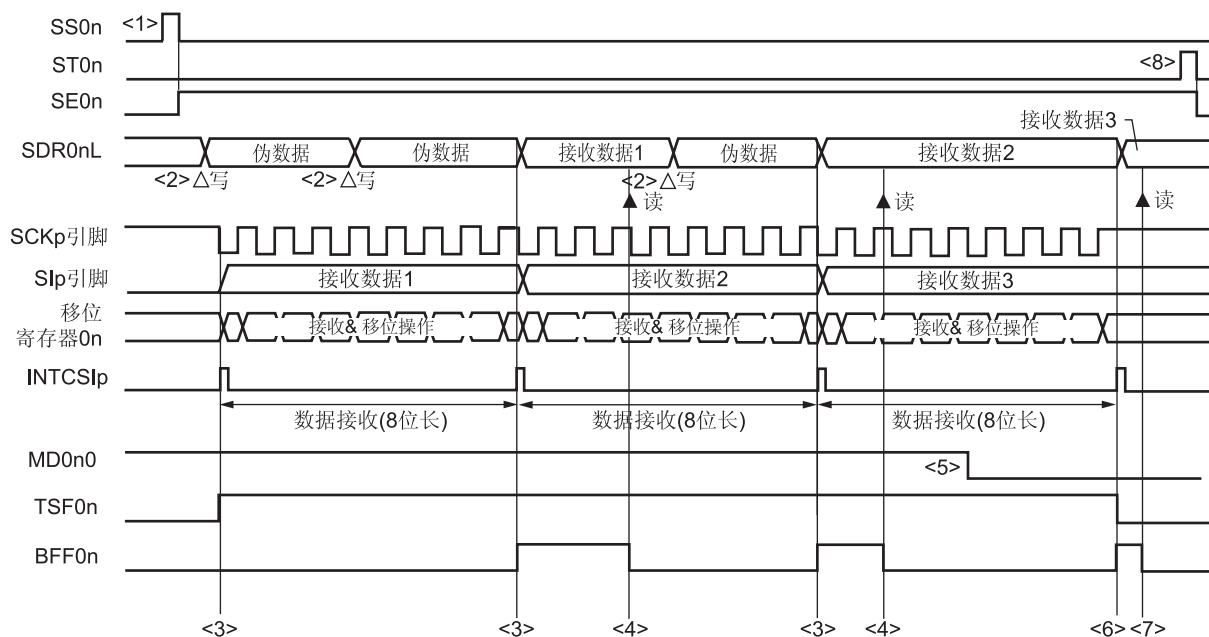
备注 n = 0, p: CSI 编号(p = 00)

图 10-37. 主接收 (单接收模式时)的流程图



(4) 处理流程 (连续接收模式时)

图 10-38. 主接收 (连续接收模式时)的时序图(类型 1: DAP0n = 0, CKP0n = 0)



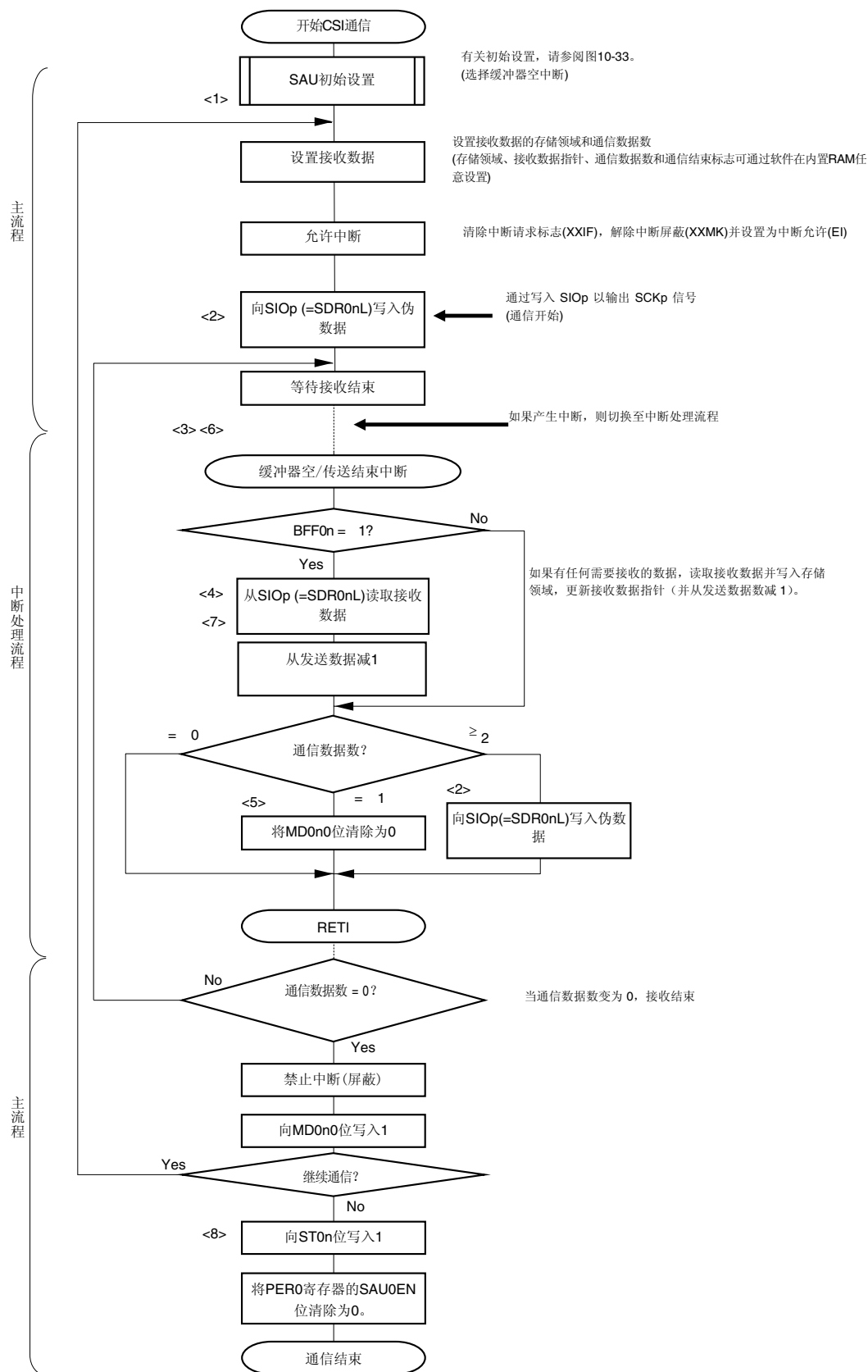
注意事项 即使在操作过程中也可改写 MD0n0 位。

但是，一定要在开始接收最后一位之前改写，以便在最后接收数据的传送结束中断之前完成改写。

备注 1. 此图中的<1>至<8>对应于图 10-39 主接收 (连续接收模式时)的流程图中的<1>至<8>。

2. n = 0, p: CSI 编号(p = 00)

图 10-39. 主接收 (连续接收模式时)的流程图



备注 此图中的<1>至<8>对应于图 10-38 主接收 (连续接收模式时)的时序图中的<1>至<8>。

10.5.3 主发送/接收

主发送/接收指 R7F0C80112ESP, R7F0C80212ESP 输出传送时钟，并将数据发送至其他器件或从其他器件接收数据。

3线串行输入/输出	CSI00
目标通道	SAU0的通道0
使用引脚	SCK00, SI00, SO00
中断	INTCSI00 可选择传送结束中断(单传送模式时)或缓冲器空中断(连续传送模式时)
错误检测标志	仅限溢出错误检测标志 (OVF0n)
传送数据长度	7或8位
传送速率	Max. $f_{CLK}/4$ [Hz] (SDR0nH[7:1] = 1以上) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] ^{注2}
数据相位	可通过SCR0nH寄存器的DAP0n位选择 • DAP0n = 0: 从开始串行时钟操作时开始数据输入/输出。 • DAP0n = 1: 从开始串行时钟操作的半个时钟前开始数据输入/输出。
时钟相位	可通过SCR0nH寄存器的CKP0n位选择 • CKP0n = 0: 正相 • CKP0n = 1: 反相
数据方向	MSB或LSB优先

注 在满足以上条件及电特性中的 AC 特性(参阅第二十一章 电特性)的范围内使用此操作。

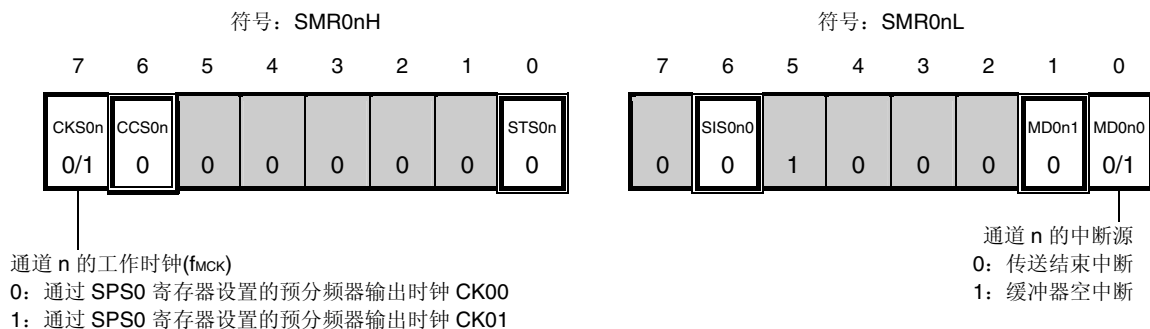
备注 1. f_{CLK} : 系统时钟频率

2. $n = 0$

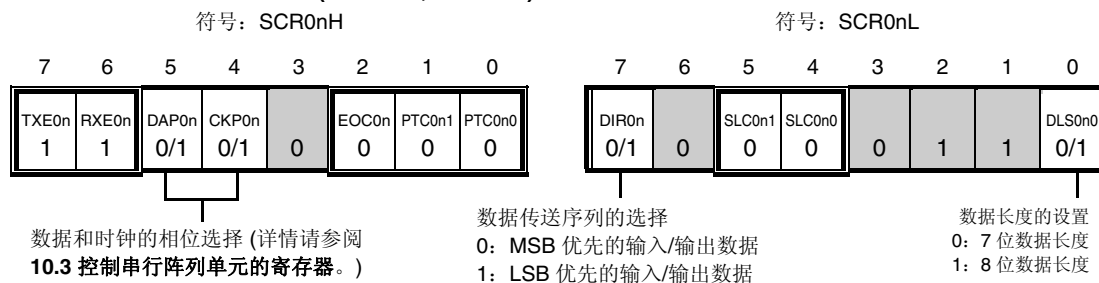
(1) 寄存器设置

图 10-40. 3 线串行输入/输出 (CSI00)的主发送/接收时的寄存器设置内容示例(1/2)

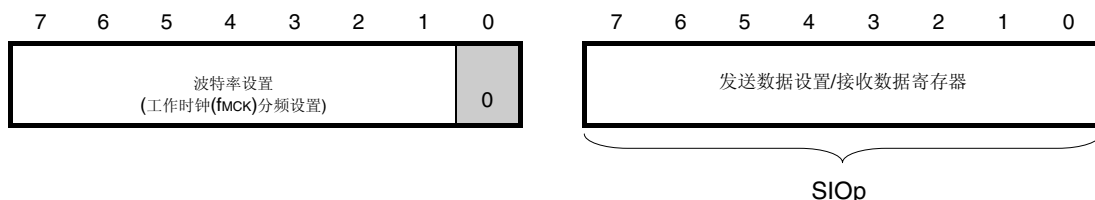
(a) 串行模式寄存器 0n (SMR0nH, SMR0nL)



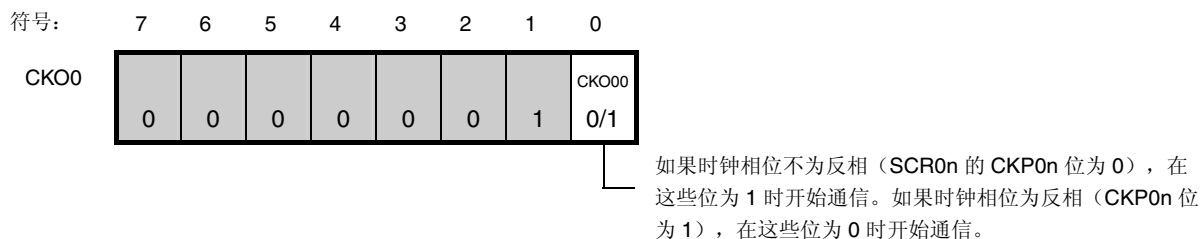
(b) 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)



(c) 串行数据寄存器 0n (SDR0nH, SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) ... 仅对目标通道的位进行设置。



(e) 串行输出寄存器 0 (SO0) ... 仅对目标通道的位进行设置。

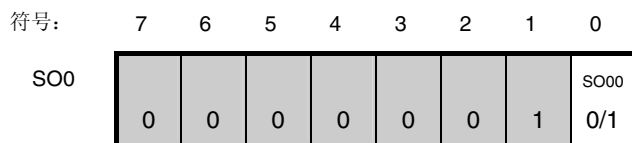


图 10-40. 3 线串行输入/输出 (CSI00)的主发送/接收时的寄存器设置内容示例(2/2)

(f) 串行输出允许寄存器 0 (SOE0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SOE0								SOE00
	0	0	0	0	0	0	0	0/1

(g) 串行通道开始寄存器 0 (SS0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SS0							SS01	SS00
	0	0	0	0	0	0	0	0/1

备注 1. n = 0, p: CSI 编号(p = 00)

2. ☐: 设置固定于 CSI 主发送/接收模式

☐: 禁止设置 (设为初始值)

x: 此模式下不能使用的位 (任意模式下均不使用时设置为初始值)

0/1: 可根据用户的用途设置为 0 或 1

(2) 操作步骤

图 10-41. 主发送/接收的初始设置步骤

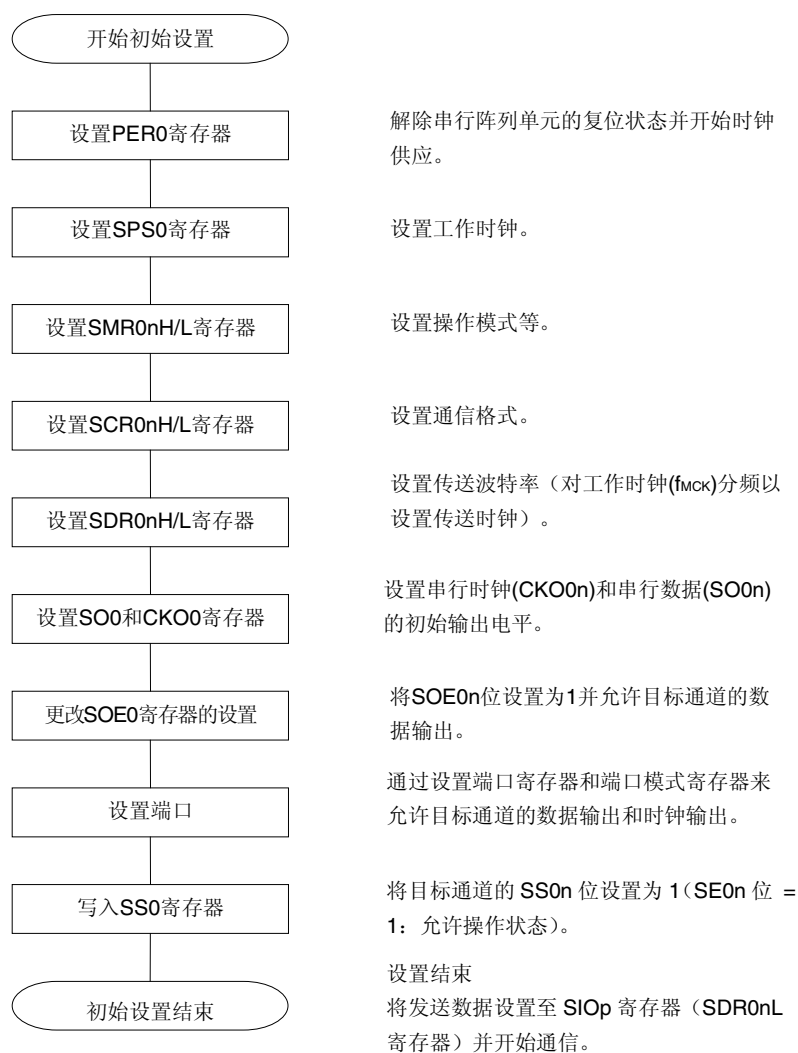


图 10-42. 停止主发送/接收的步骤

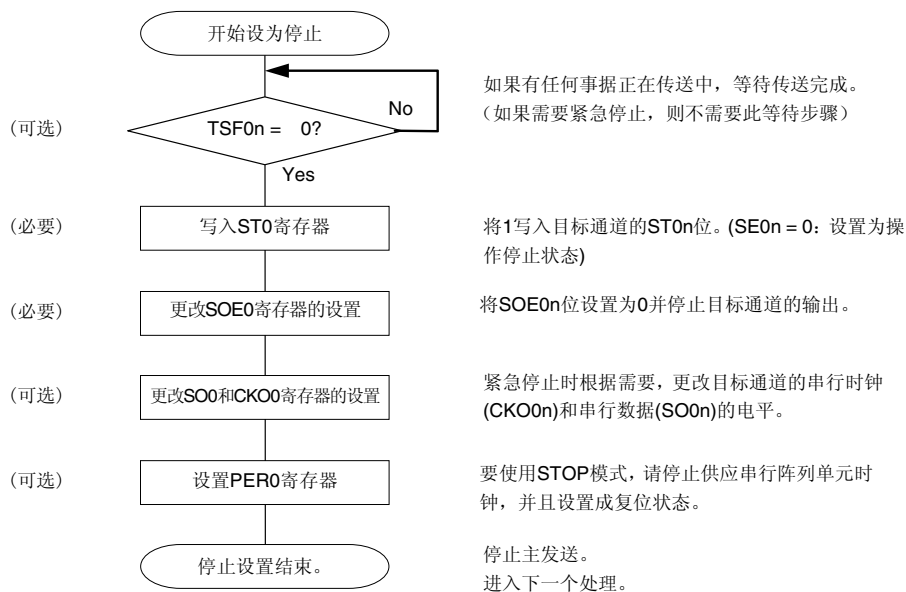


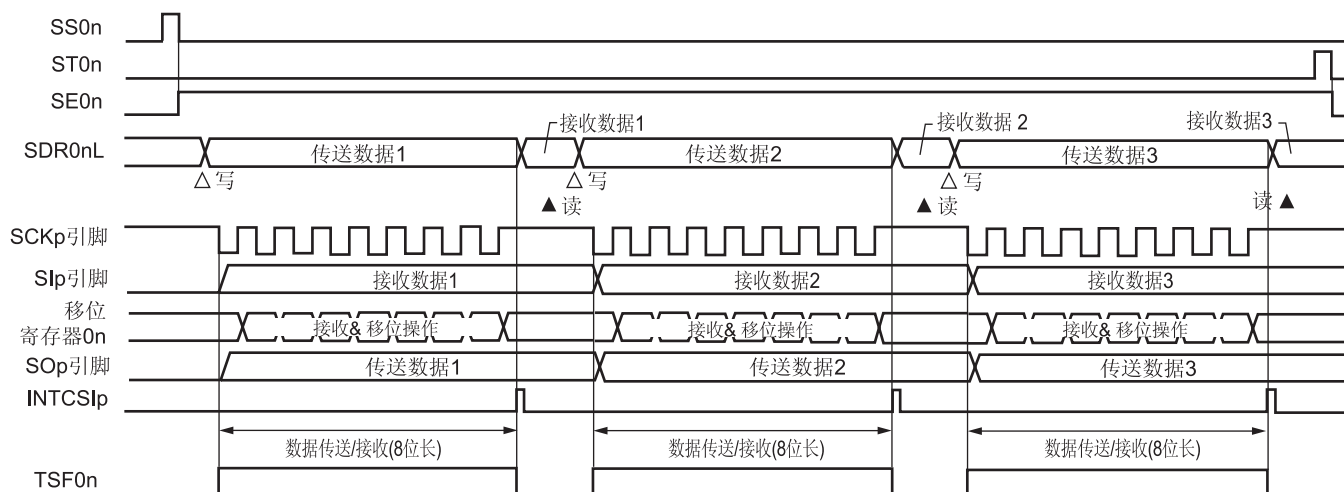
图 10-43. 恢复主发送/接收的步骤



备注 在停止主发送的步骤中改写 PER0 而停止时钟供应时,请等待通信目标(从设备)停止或者通信结束,然后执行初始设置,而不是重传设置。

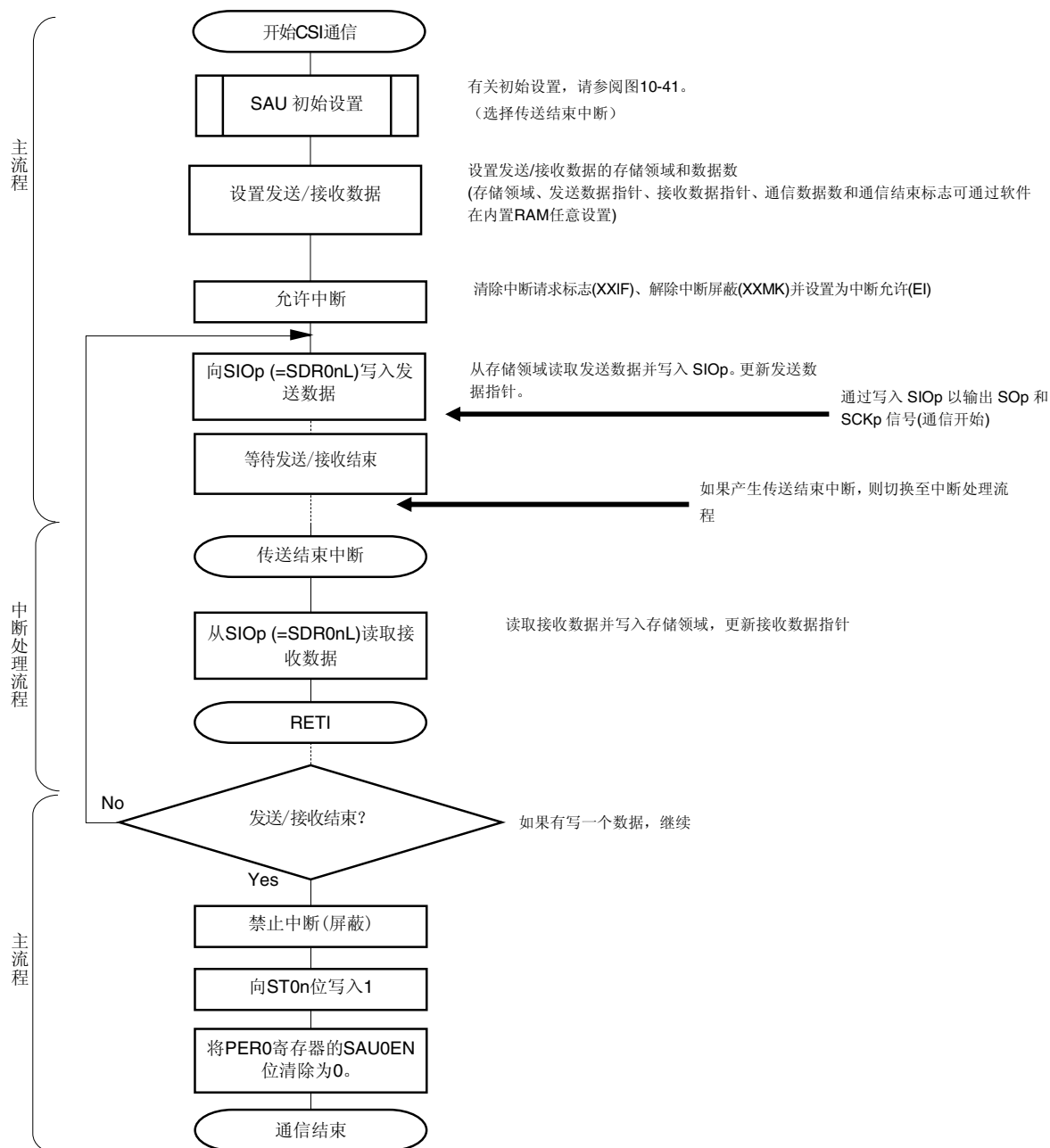
(3) 处理流程 (单发送/接收模式时)

图 10-44. 主发送/接收 (单发送/接收模式时)的时序图
(类型 1: DAP0n = 0, CKP0n = 0)



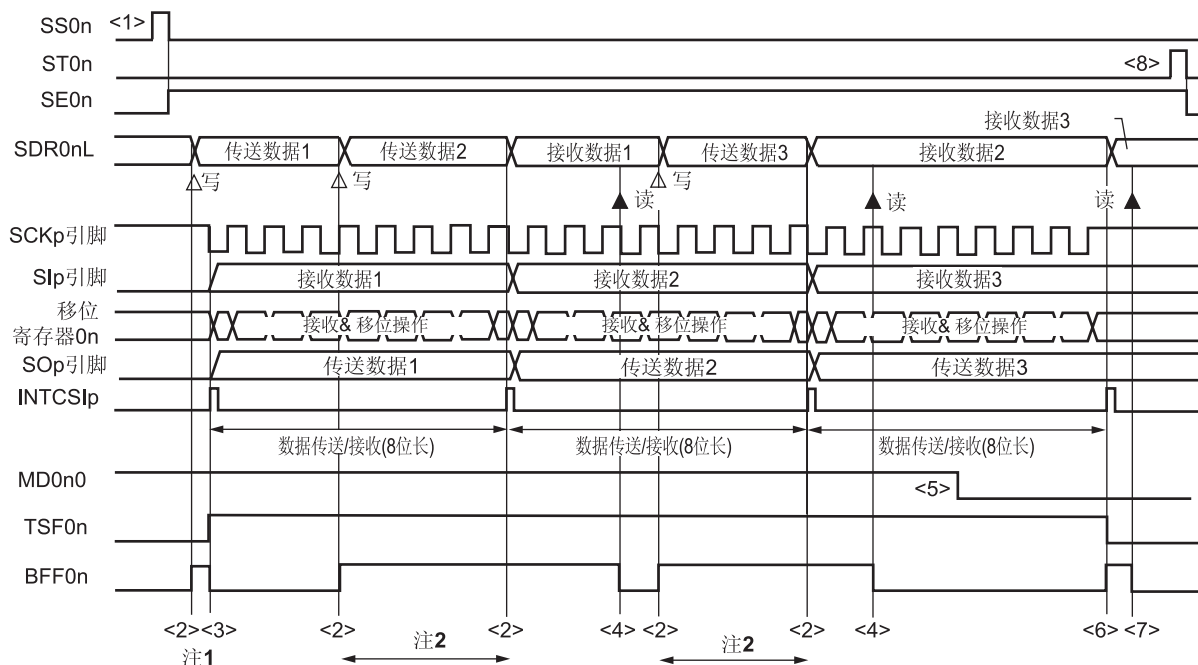
备注 n = 0, p: CSI 编号(p = 00)

图 10-45. 主发送/接收 (单发送/接收模式时)的流程图



(4) 处理流程 (连续发送/接收模式时)

图 10-46. 主发送/接收 (连续发送/接收模式时)的时序图
(类型 1: DAP0n = 0, CKP0n = 0)



注 1. 如果在串行状态寄存器 0n (SSR0n)的 BFF0n 位为 1 时将发送数据写入 SDR0nL 寄存器 (将有效数据存储于串行数据寄存器 0n (SDR0nL)中), 则发送数据将被覆盖。

2. 在此过程中可以通过读取 SDR0nL 寄存器来读取发送数据。此时, 传送操作不受影响。

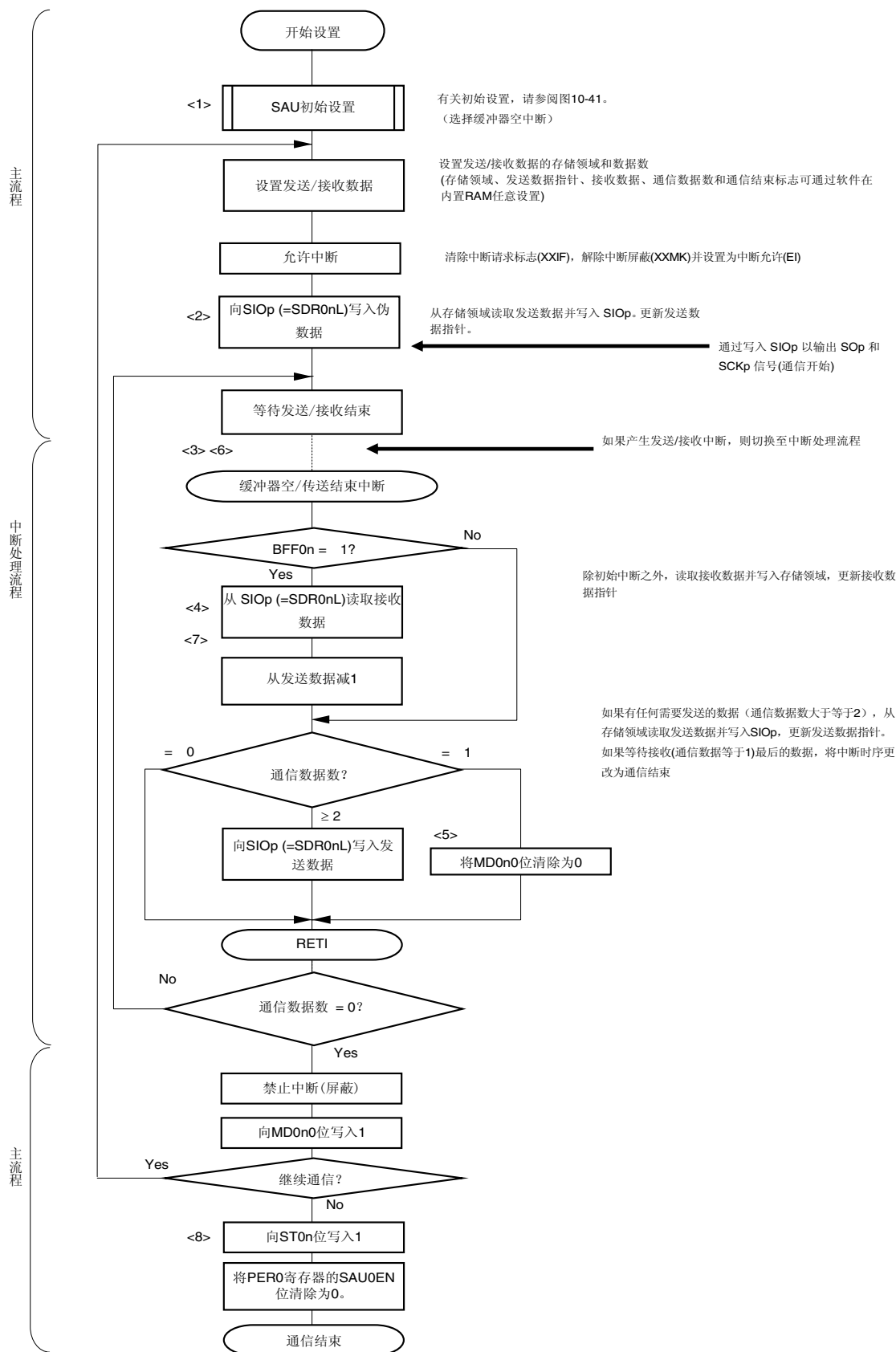
注意事项 即使在操作过程中, 也可改写串行模式寄存器 0n (SMR0nL)的 MD0n0 位。

但是, 一定要在开始传送最后一位之前改写, 以便在最后发送数据的传送结束中断之前完成改写。

备注 1. 此图中的<1>至<8>对应于图 10-47 主发送/接收 (连续发送/接收模式时)的流程图中的<1>至<8>。

2. n = 0, p: CSI 编号 (p = 00)

图 10-47. 主发送/接收 (连续发送/接收模式时)的流程图



备注 此图中的<1>至<8>对应于图 10-46 主发送/接收 (连续发送/接收模式时)的时序图中的<1>至<8>。

10.5.4 从发送

从发送是指，当接收来自另一个器件的传送时钟输入时，R7F0C80112ESP, R7F0C80212ESP 将数据发送至其他器件。

3线串行输入/输出	CSI00
目标通道	SAU0的通道0
使用引脚	$\overline{\text{SCK00}}$, SO00
中断	INTCSI00 可选择传送结束中断(单传送模式时)或缓冲器空中断(连续传送模式时)
错误检测标志	仅限溢出错误检测标志 (OVF0n)
传送数据长度	7或8位
传送速率	Max. $f_{\text{MCK}}/6$ [Hz] ^{※1, 2}
数据相位	可通过SCR0nH寄存器的DAP0n位选择 • DAP0n = 0: 从开始串行时钟操作时开始数据输出。 • DAP0n = 1: 从开始串行时钟操作的半个时钟前开始数据输出。
时钟相位	可通过SCR0nH寄存器的CKP0n位选择 • CKP0n = 0: 正相 • CKP0n = 1: 反相
数据方向	MSB或LSB优先

注 1. 输入至 $\overline{\text{SCK00}}$ 引脚的外部串行时钟是在内部通过采样使用，因此最大传送速率为 $f_{\text{MCK}}/6$ [Hz]。通过设置 SPS0 寄存器，使该频率为最小 $f_{\text{SCK}}/2$ （由 SDR0nH 寄存器设置）。

2. 在满足以上条件及电特性中的 AC 特性(参阅第二十一章 电特性)的范围内使用此操作。

备注 1. f_{CLK} : 系统时钟频率

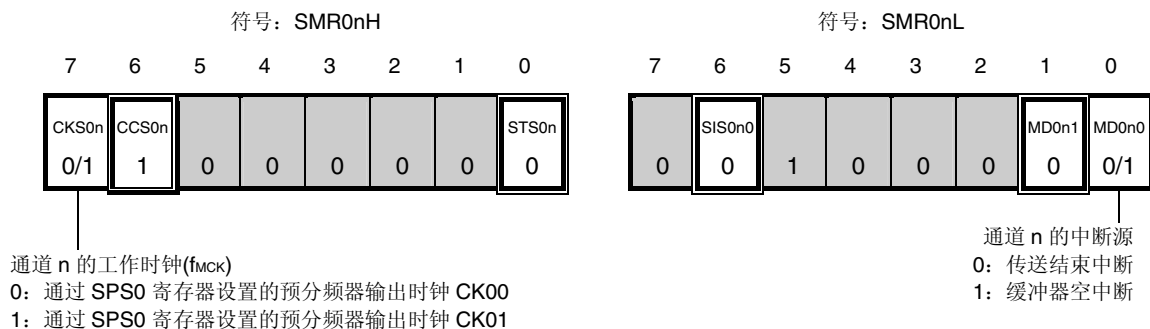
f_{SCK} : 串行时钟频率

2. $n = 0$

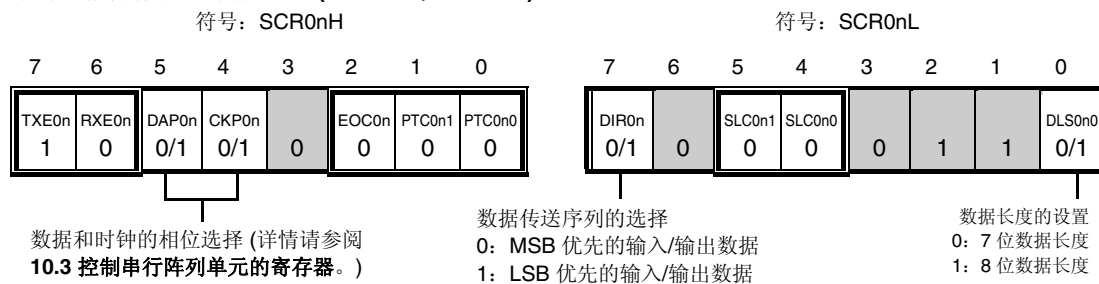
(1) 寄存器设置

图 10-48. 3 线串行输入/输出(CSI00)的从发送时的寄存器设置内容示例(1/2)

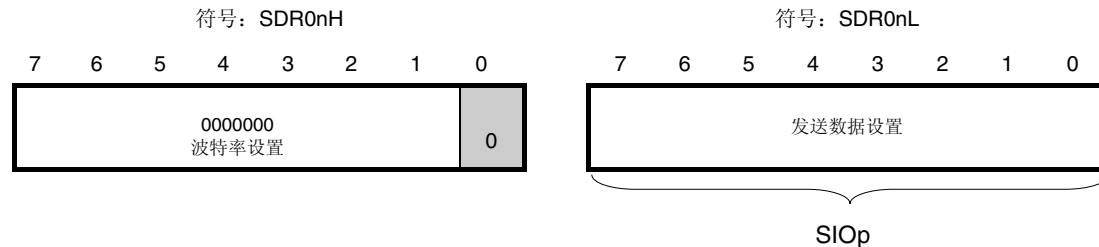
(a) 串行模式寄存器 0n (SMR0nH, SMR0nL)



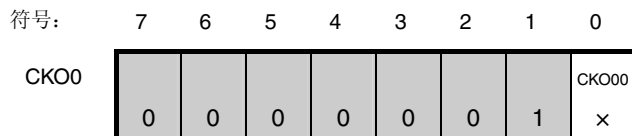
(b) 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)



(c) 串行数据寄存器 0n (SDR0nH, SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) ... 该模式下不使用。



(e) 串行输出寄存器 0 (SO0) ... 仅对目标通道的位进行设置。

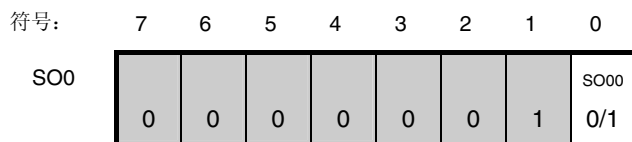


图 10-48. 3 线串行输入/输出(CSI00)的从发送时的寄存器设置内容示例(2/2)

(f) 串行输出允许寄存器 0 (SOE0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SOE0								SOE00
	0	0	0	0	0	0	0	0/1

(g) 串行通道开始寄存器 0 (SS0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SS0							SS01	SS00
	0	0	0	0	0	0	0	0/1

备注 1. n = 0 p: CSI 编号(p = 00)

2. ☐ : 设置固定于 CSI 主发送模式, ☐ : 禁止设置 (设为初始值)
 x: 此模式下不能使用的位 (任意模式下均不使用时设为初始值)
 0/1: 可根据用户的用途设为 0 或 1

(2) 操作步骤

图 10-49. 从发送的初始设置步骤

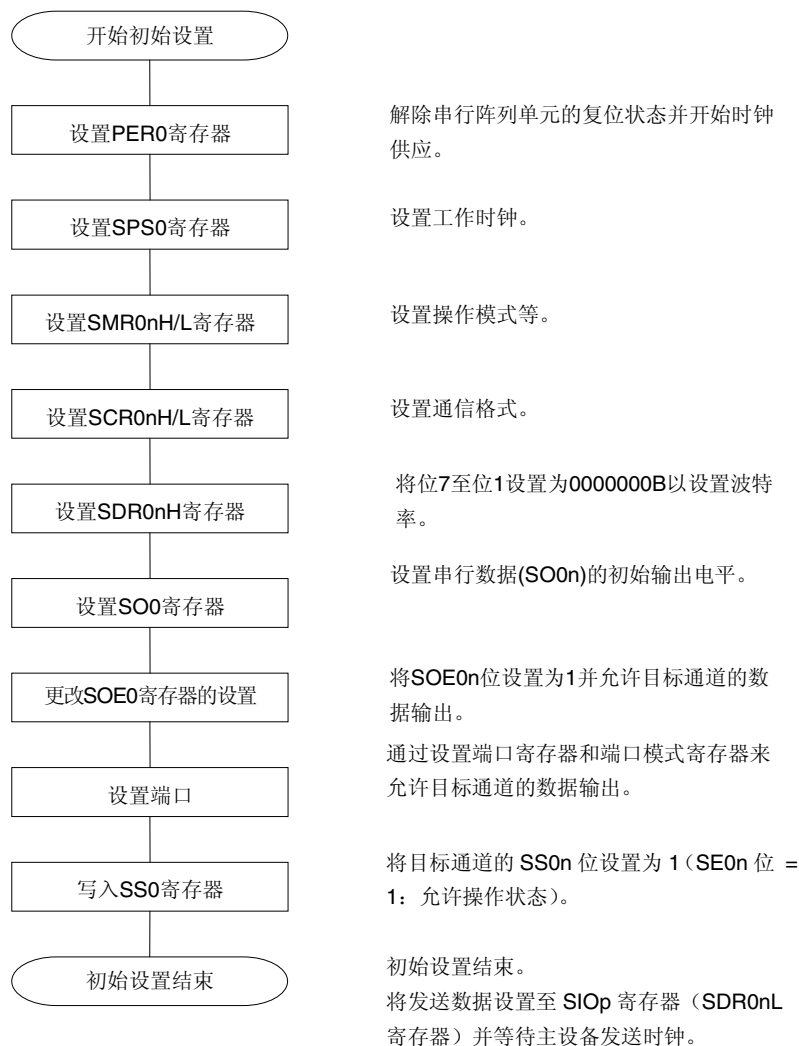


图 10-50. 停止从发送的步骤

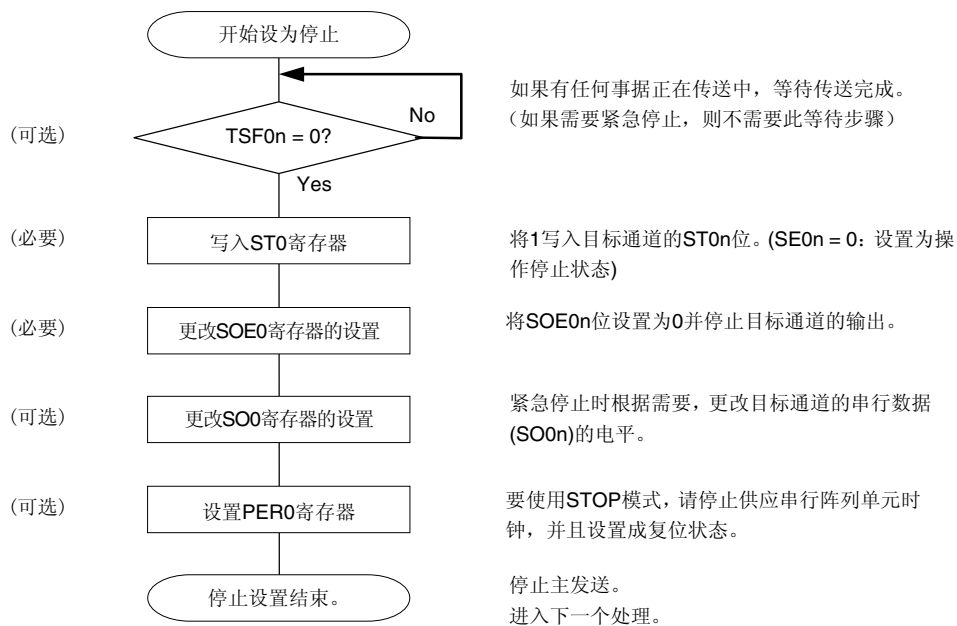


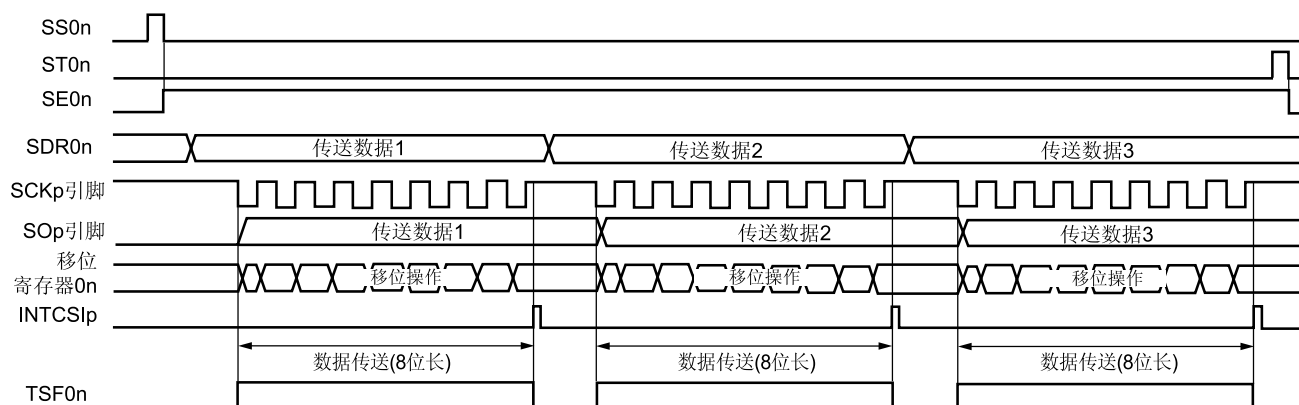
图 10-51. 恢复从发送的步骤



备注 在停止主发送的步骤中改写 PER0 而停止时钟供应时, 请等待通信目标 (主设备) 停止或者通信结束, 然后执行初始设置, 而不是重传设置。

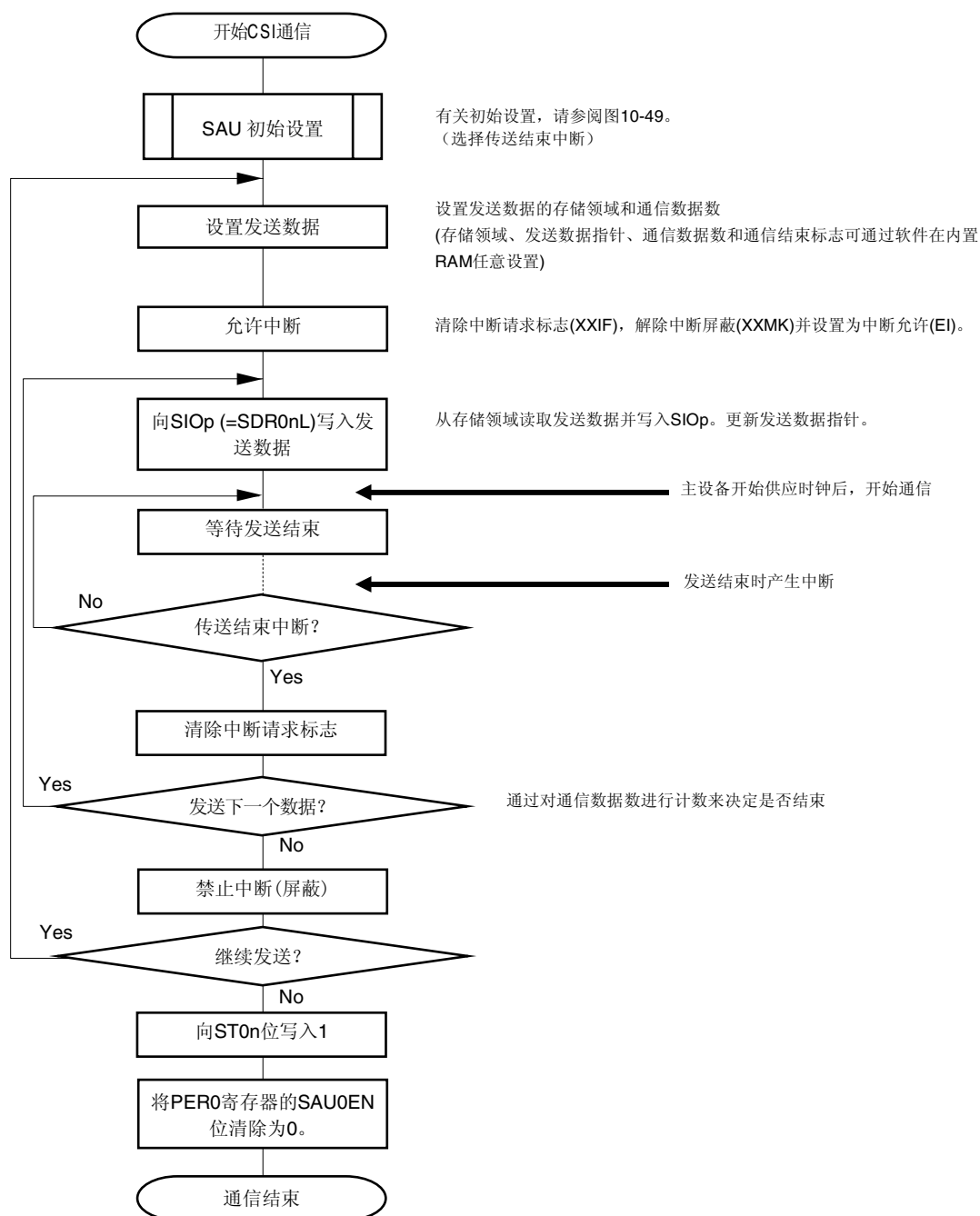
(3) 处理流程 (单发送模式时)

图 10-52. 从发送(单发送模式时)的时序图
(类型 1: DAP0n = 0, CKP0n = 0)



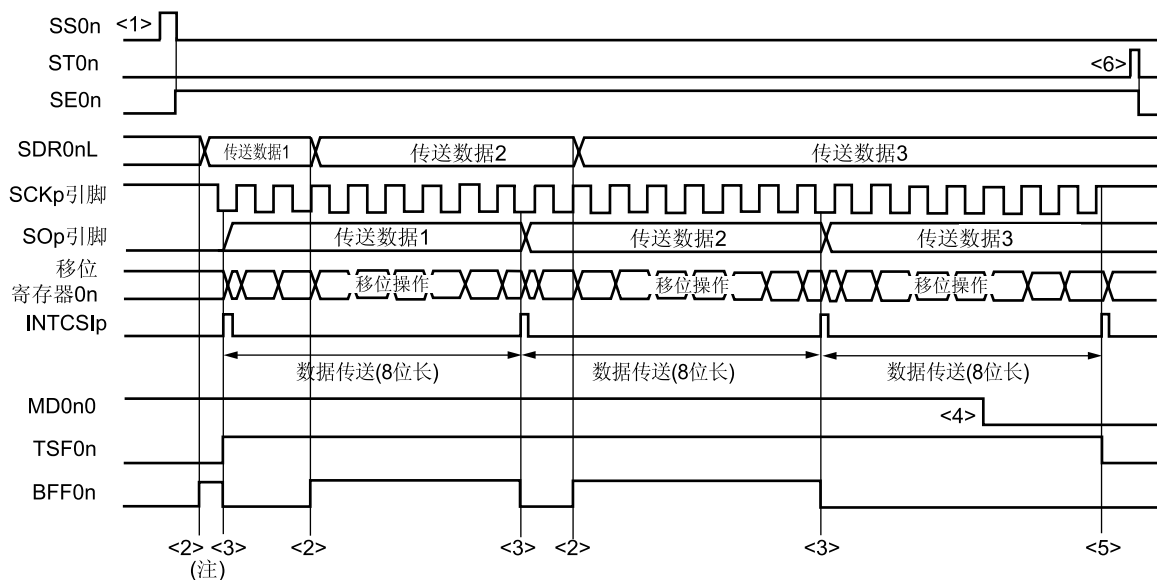
备注 n = 0, p: CSI 编号(p = 00)

图 10-53. 从发送(单发送模式时)的流程图



(4) 处理流程 (连续发送模式时)

图 10-54. 从发送(连续发送模式时)的时序图
(类型 1: DAP0n = 0, CKP0n = 0)

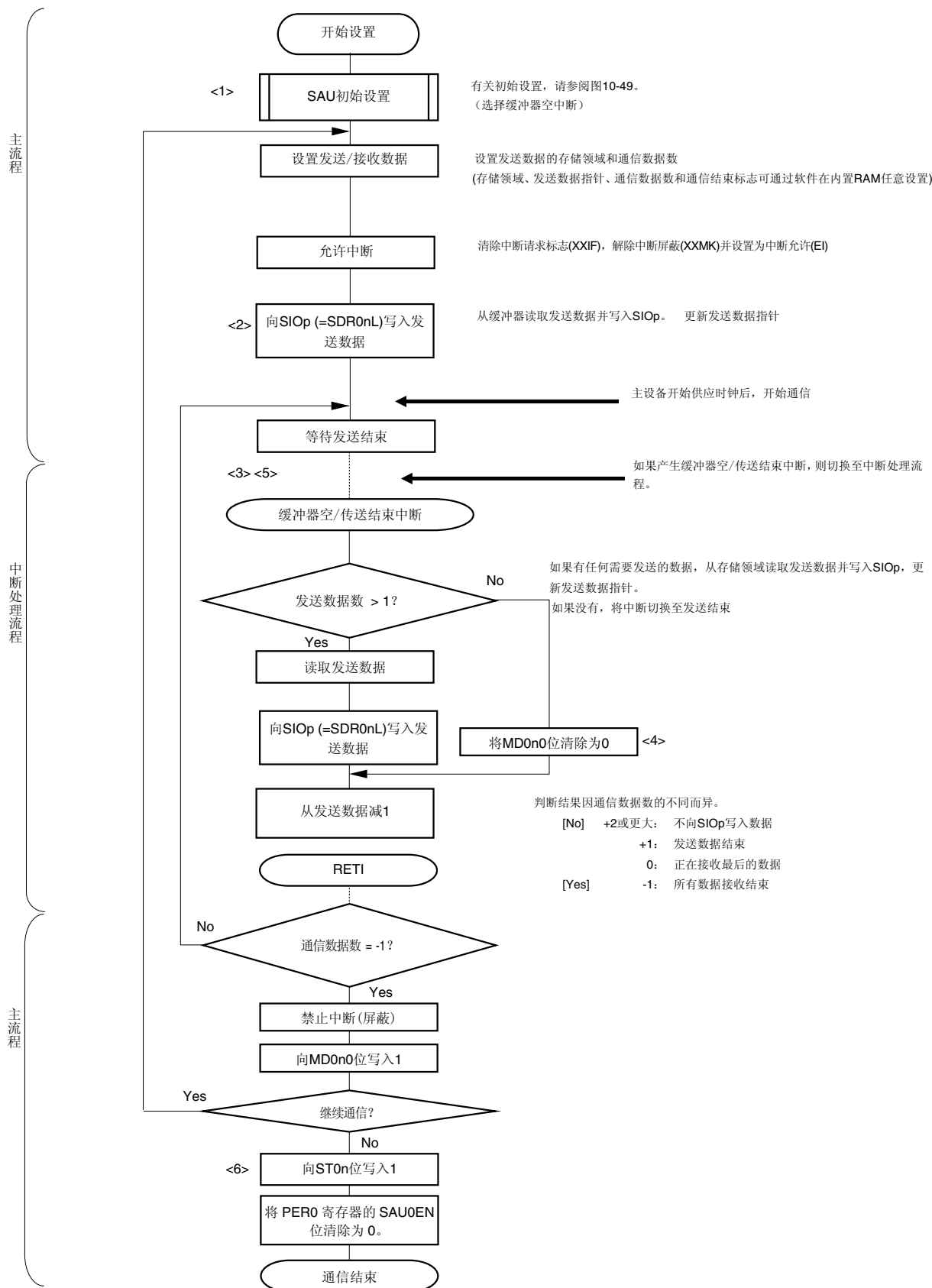


注 如果在串行状态寄存器 0n (SSR0n)的 BFF0n 位为 1 时将发送数据写入 SDR0nL 寄存器 (将有效数据存储于串行数据寄存器 0n (SDR0nL)中), 则发送数据将被覆盖。

注意事项 即使在操作过程中, 也可改写串行模式寄存器 0n (SMR0nL)的 MD0n0 位。但是, 要在最后一位开始传送前改写。

备注 n = 0, p: CSI 编号(p = 00)

图 10-55. 从发送(连续发送模式时)的流程图



备注 此图中的<1>至<6>对应于图 10-54 从发送(连续发送模式时)的时序图中的<1>至<6>。

10.5.5 从接收

从接收是指，当接收来自另一个器件的传送时钟输入时，R7F0C80112ESP, R7F0C80212ESP 从其他器件接收数据。

3线串行输入/输出	CSI00
目标通道	SAU0的通道0
使用引脚	SCK00, SI00
中断	INTCSI00 可选择传送结束中断(单传送模式时)或缓冲器空中断(连续传送模式时)
错误检测标志	仅限溢出错误检测标志 (OVF0n)
传送数据长度	7或8位
传送速率	Max. $f_{MCK}/6$ [Hz] ^{注1, 2}
数据相位	可通过SCR0nH寄存器的DAP0n位选择 • DAP0n = 0: 从开始串行时钟操作时开始数据输出。 • DAP0n = 1: 从开始串行时钟操作的半个时钟前开始数据输出。
时钟相位	可通过SCR0nH寄存器的CKP0n位选择 • CKP0n = 0: 正相 • CKP0n = 1: 反相
数据方向	MSB或LSB优先

注 1. 输入至 SCK00 引脚的外部串行时钟是在内部通过采样使用，因此最大传送速率为 $f_{MCK}/6$ [Hz]。通过设置 SPS0 寄存器，使该频率为最小 $f_{SCK}/2$ （由 SDR0nH 寄存器设置）。

2. 在满足以上条件及电特性中的 AC 特性(参阅第二十一章 电特性)的范围内使用此操作。

备注 1. f_{CLK} : 系统时钟频率

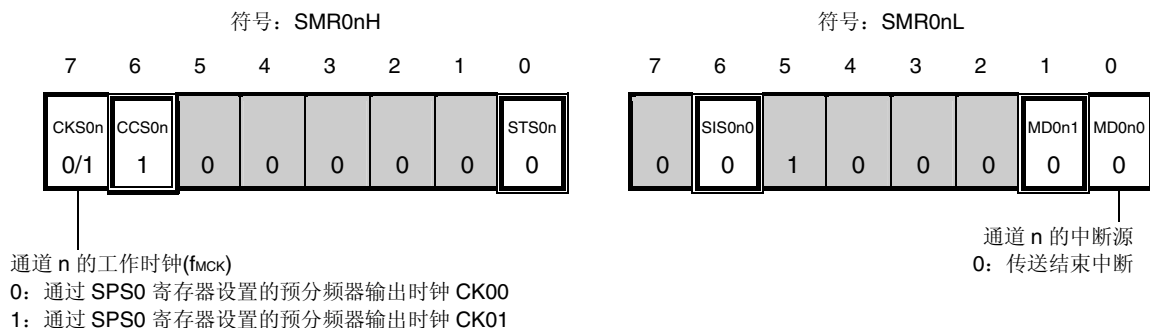
f_{SCK} : 串行时钟频率

2. $n = 0$

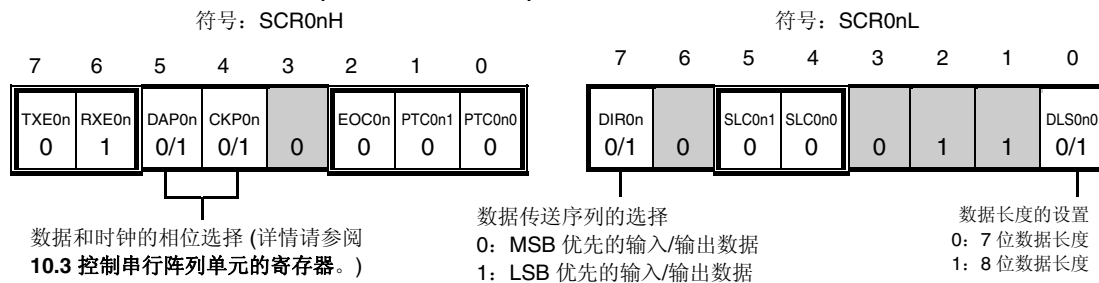
(1) 寄存器设置

图 10-56. 3 线串行输入/输出(CSI00)的从接收时的寄存器设置内容示例(1/2)

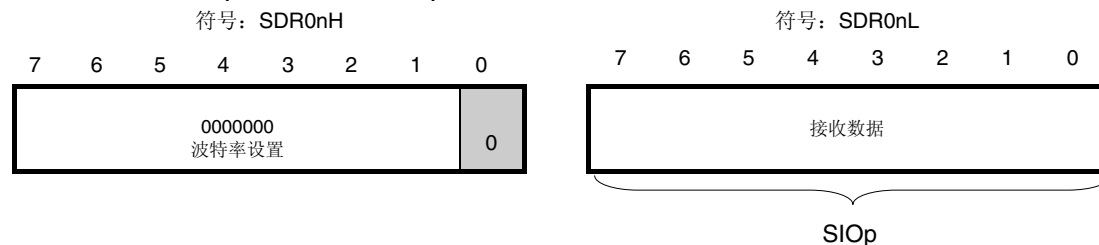
(a) 串行模式寄存器 0n (SMR0nH, SMR0nL)



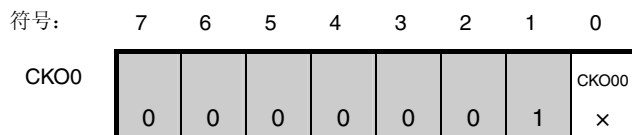
(b) 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)



(c) 串行数据寄存器 0n (SDR0nH, SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) ... 此模式下不使用的寄存器。



(e) 串行输出寄存器 0 (SO0) ...此模式下不使用的寄存器。

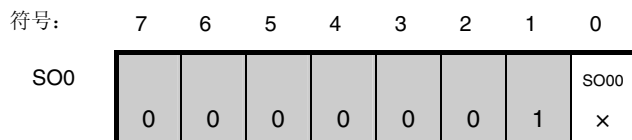


图 10-56. 3 线串行输入/输出(CSI00)的从接收时的寄存器设置内容示例(2/2)

(f) 串行输出允许寄存器 0 (SOE0) ...此模式下不使用的寄存器。

符号: 7 6 5 4 3 2 1 0

SOE0								SOE00
	0	0	0	0	0	0	0	x

(g) 串行通道开始寄存器 0 (SS0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SS0							SS01	SS00
	0	0	0	0	0	0	0	0/1

备注 1. n = 0, p: CSI 编号(p = 00)

2. ☐: 设置固定于 CSI 主发送模式, ☐: 禁止设置 (设为初始值)
 x: 此模式下不能使用的位 (任意模式下均不使用时设置为初始值)
 0/1: 可根据用户的用途设置为 0 或 1

(2) 操作步骤

图 10-57. 从接收的初始设置步骤

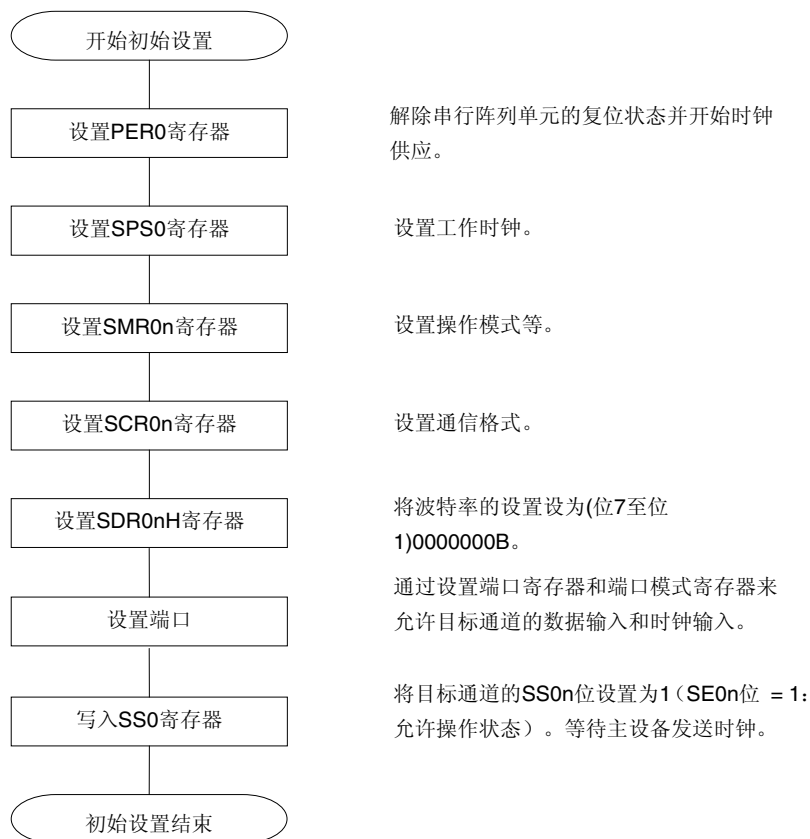


图 10-58. 停止从接收的步骤

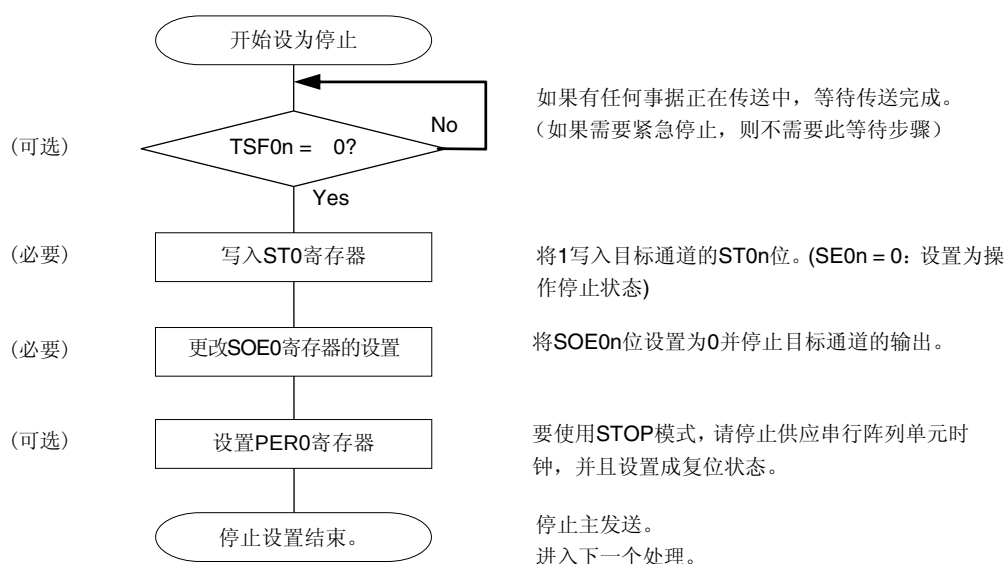
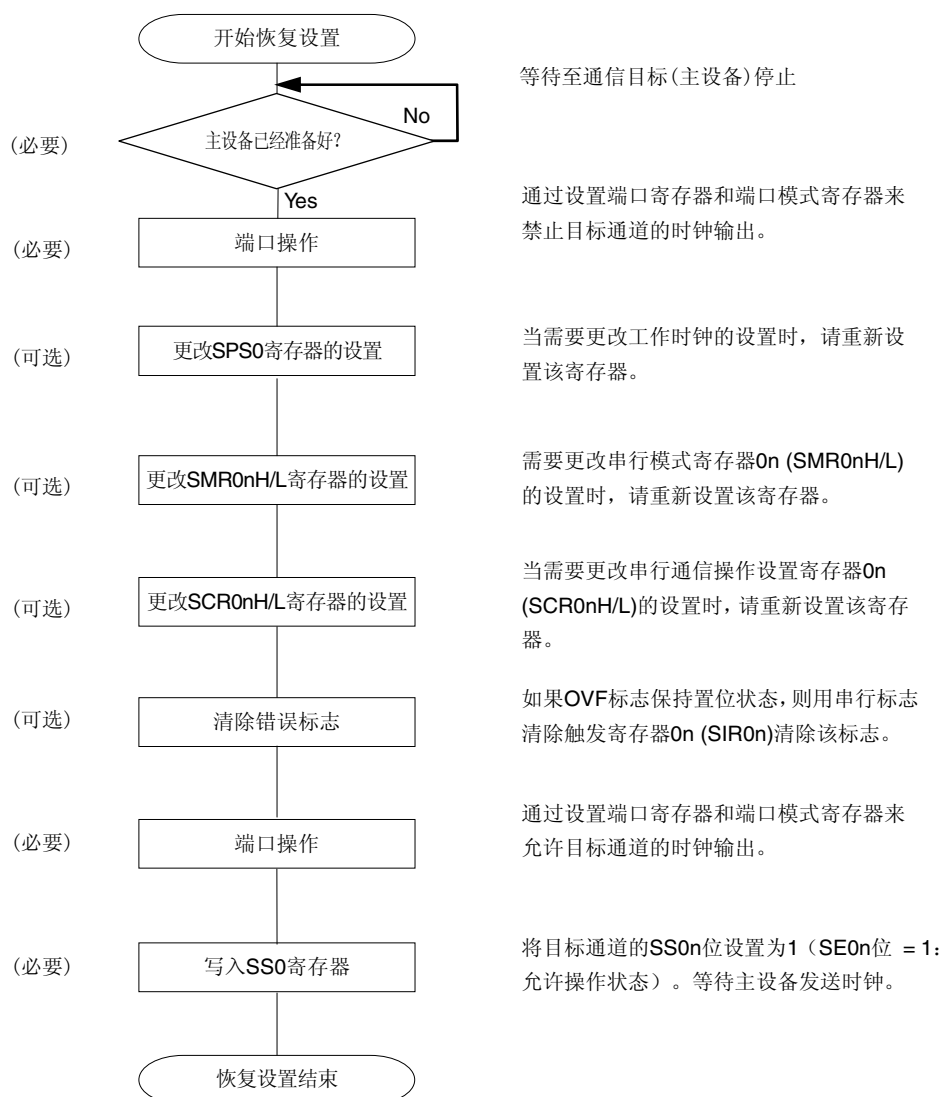


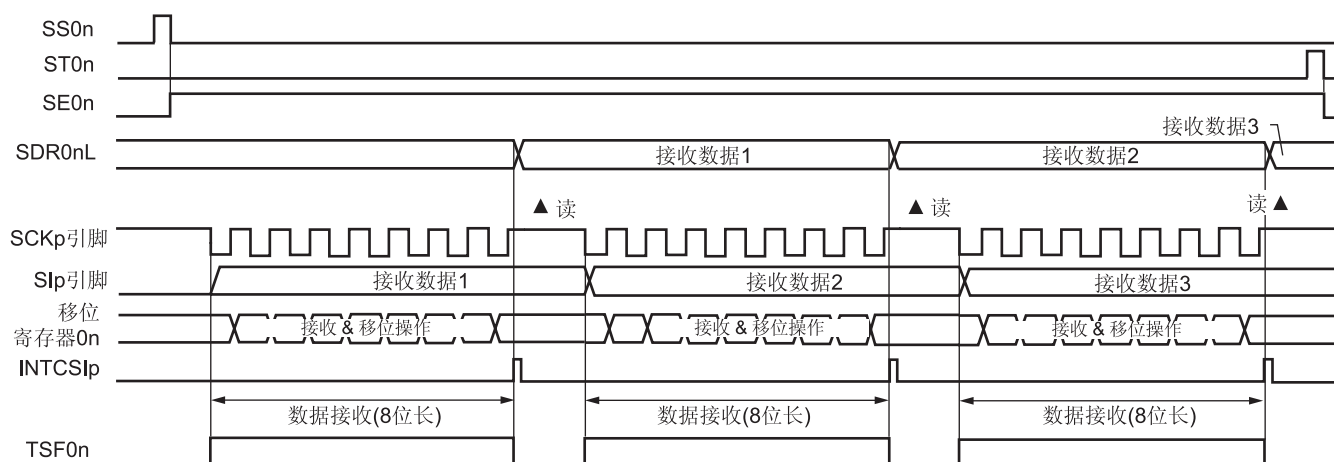
图 10-59. 恢复从接收的步骤



备注 在停止主发送的步骤中改写 PER0 而停止时钟供应时, 请等待通信目标 (主设备) 停止或者通信结束, 然后执行初始设置, 而不是重传设置。

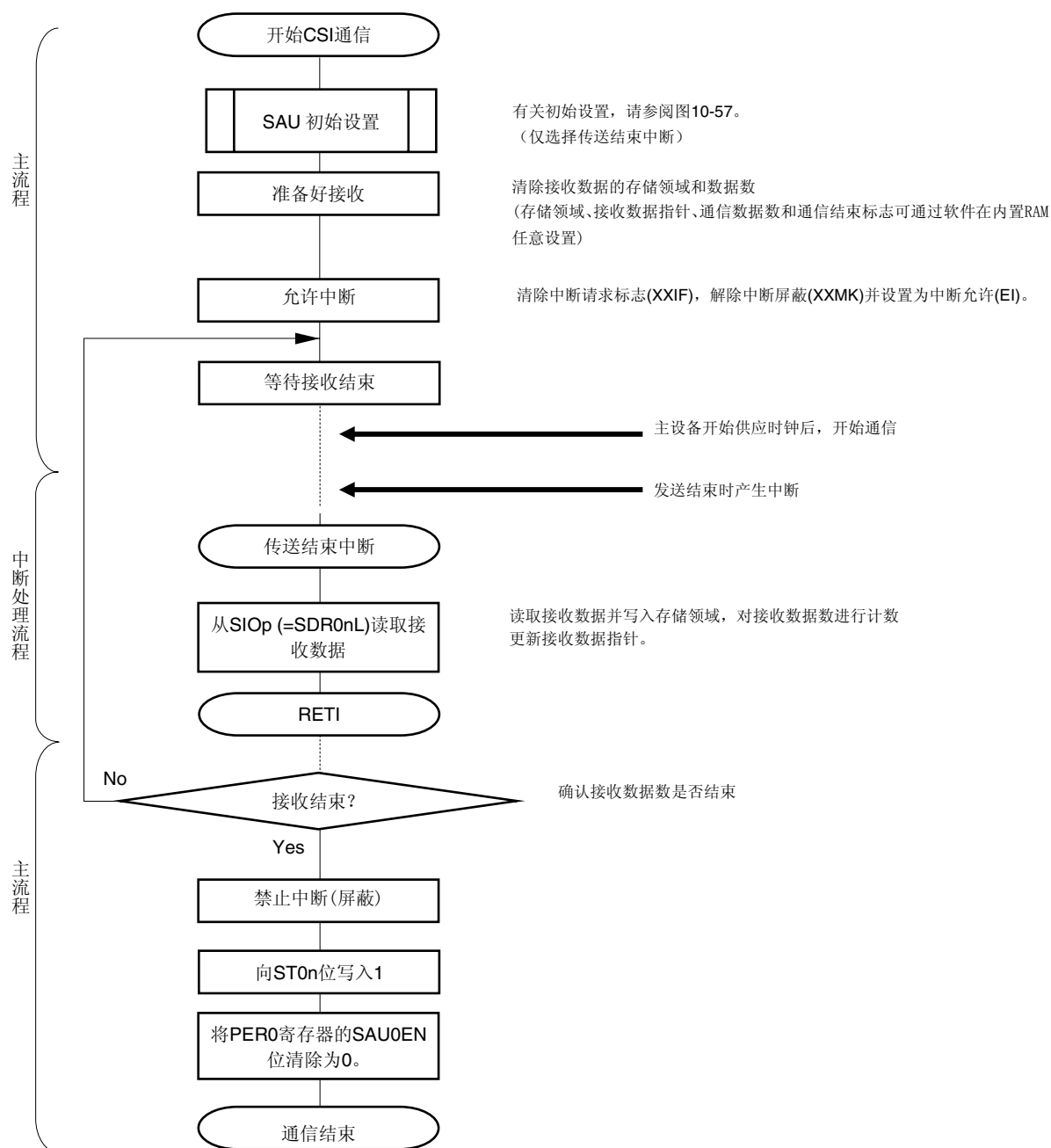
(3) 处理流程 (单接收模式时)

图 10-60. 从接收(单接收模式时)的时序图(类型 1: DAP0n = 0, CKP0n =)



备注 n = 0, p: CSI 编号(p = 00)

图 10-61. 从接收(单接收模式时)的流程图



10.5.6 从发送/接收

从发送/接收是指，当接收来自另一个器件的传送时钟输入时，R7F0C80112ESP, R7F0C80212ESP 发送数据至其他器件或者从其他器件接收数据。

3线串行输入/输出	CSI00
目标通道	SAU0的通道0
使用引脚	$\overline{\text{SCK00}}$, SI00, SO00
中断	INTCSI00 可选择传送结束中断(单传送模式时)或缓冲器空中断(连续传送模式时)
错误检测标志	仅限溢出错误检测标志 (OVF0n)
传送数据长度	7或8位
传送速率	Max. $f_{\text{MCK}}/6$ [Hz] ^{注1, 2}
数据相位	可通过SCR0nH寄存器的DAP0n位选择 • DAP0n = 0: 从开始串行时钟操作时开始数据输出。 • DAP0n = 1: 从开始串行时钟操作的半个时钟前开始数据输出。
时钟相位	可通过SCR0nH寄存器的CKP0n位选择 • CKP0n = 0: 正相 • CKP0n = 1: 反相
数据方向	MSB或LSB优先

注 1. 输入至 $\overline{\text{SCK00}}$ 引脚的外部串行时钟是在内部通过采样使用，因此最大传送速率为 $f_{\text{MCK}}/6$ [Hz]。通过设置 SPS0 寄存器，使该频率为最小 $f_{\text{sck}}/2$ （由 SDR0nH 寄存器设置）。

2. 在满足以上条件及电特性中的 AC 特性(参阅第二十一章 电特性)的范围内使用此操作。

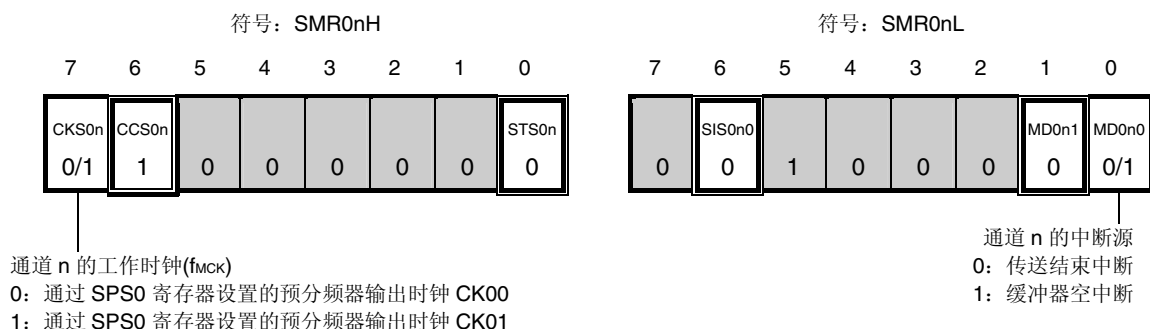
备注 1. f_{CLK} : 系统时钟频率
 f_{sck} : 串行时钟频率

2. $n = 0$

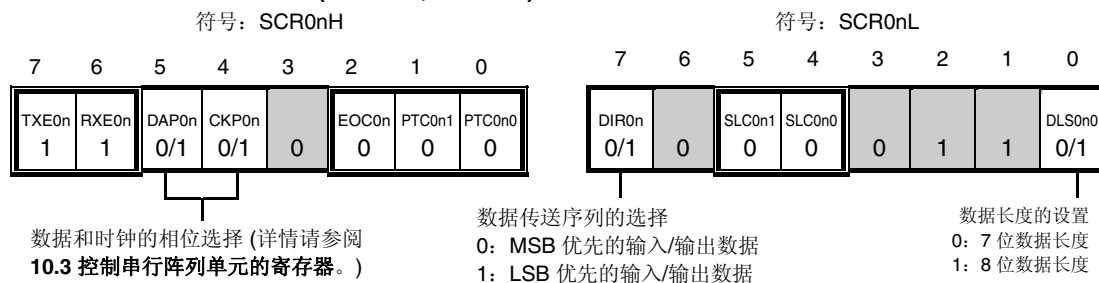
(1) 寄存器设置

图 10-62. 3 线串行输入/输出(CSI00)的从发送/接收时的寄存器设置内容示例(1/2)

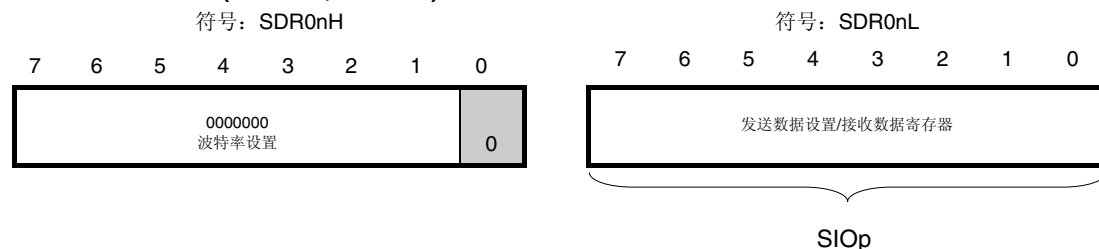
(a) 串行模式寄存器 0n (SMR0nH, SMR0nL)



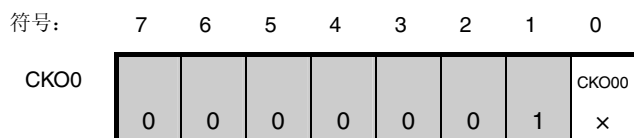
(b) 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)



(c) 串行数据寄存器 0n (SDR0nH, SDR0nL)



(d) 串行时钟输出寄存器 0 (CKO0) ... 该模式下不使用。



(e) 串行输出寄存器 0 (SO0) ... 仅对目标通道的位进行设置。

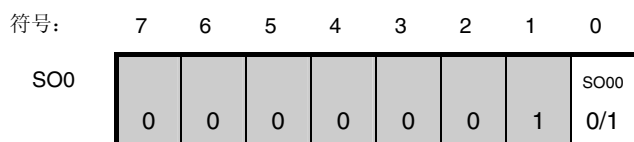


图 10-62. 3 线串行输入/输出(CSI00)的从发送/接收时的寄存器设置内容示例(2/2)

(f) 串行输出允许寄存器 0 (SOE0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SOE0								SOE00
	0	0	0	0	0	0	0	0/1

(g) 串行通道开始寄存器 0 (SS0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SS0								SS01 SS00
	0	0	0	0	0	0	0	0 0/1

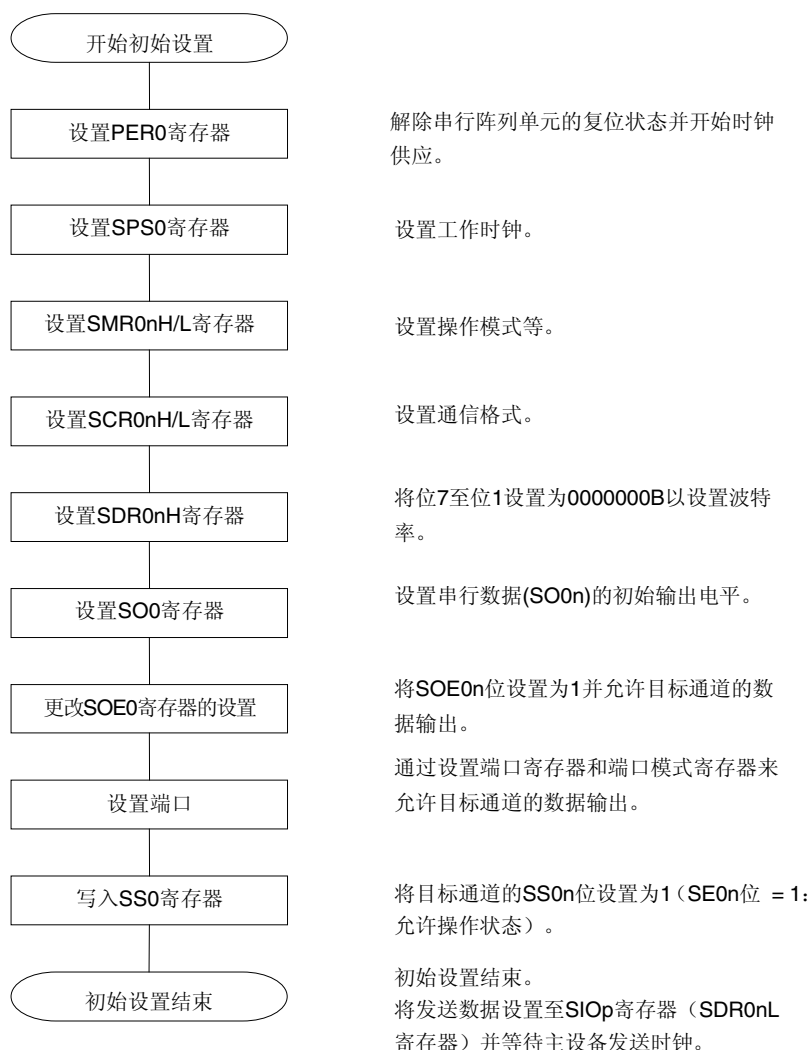
注意事项 必须在主时钟启动之前，把发送数据设置至 SIOp 寄存器。

备注 1. n = 0, p: CSI 编号(p = 00)

- 2.** ☐: 设置固定于 CSI 主发送模式, ☐: 禁止设置 (设为初始值)
 x: 此模式下不能使用的位 (任意模式下均不使用时设为初始值)
 0/1: 可根据用户的用途设为 0 或 1

(2) 操作步骤

图 10-63. 从发送/接收的初始设置步骤



注意事项 必须在主时钟启动之前，把发送数据设置至 SIOp 寄存器。

图 10-64. 停止从发送/接收的步骤

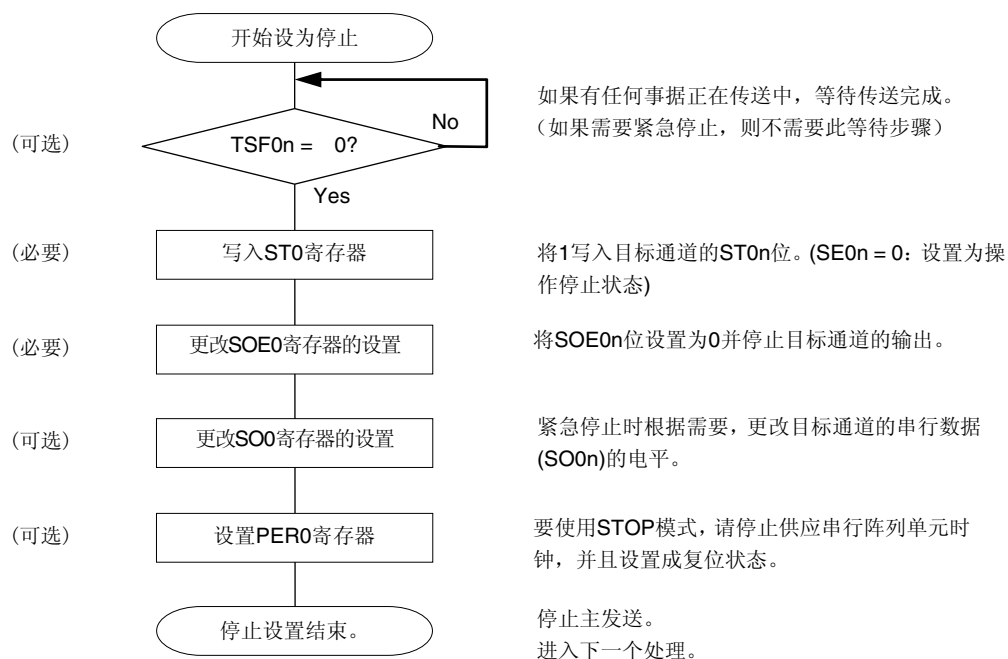


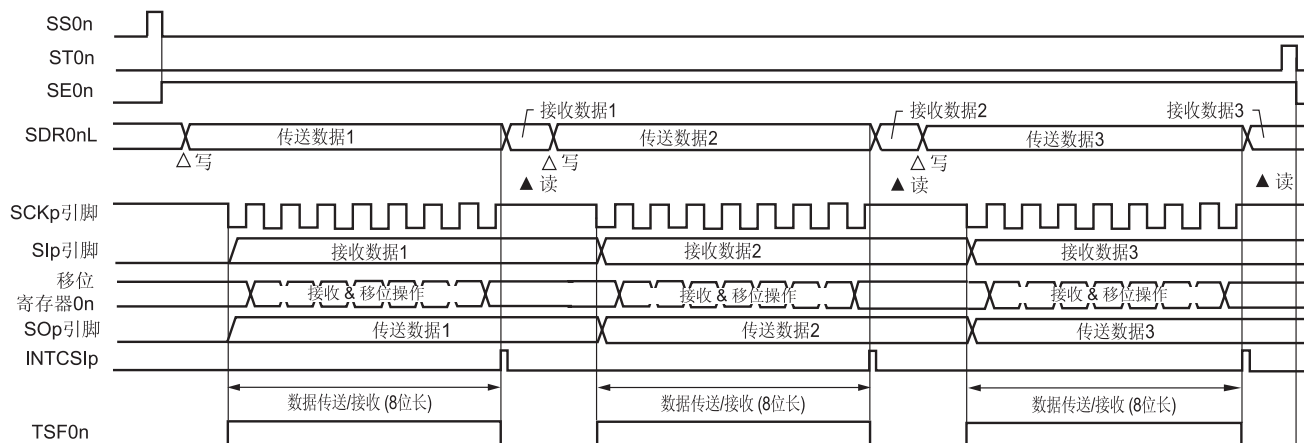
图 10-65. 恢复从发送/接收的步骤



- 注意事项**
1. 必须在主时钟启动之前, 把发送数据设置至 SIOp 寄存器。
 2. 在停止主发送的步骤中改写 PER0 而停止时钟供应时, 请等待通信目标 (主设备) 停止或者通信结束, 然后执行初始设置, 而不是重传设置。

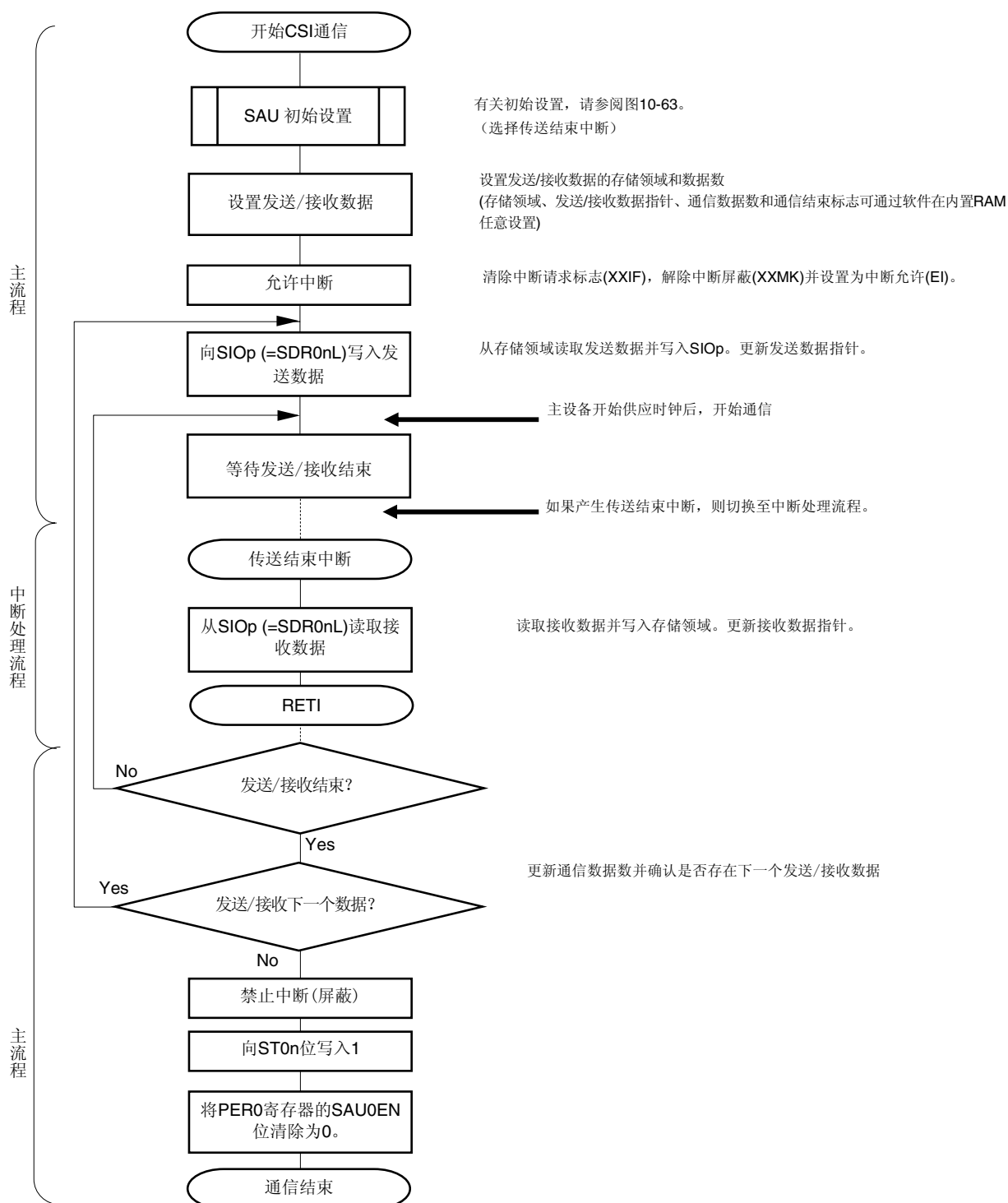
(3) 处理流程 (单发送/接收模式时)

图 10-66. 从发送/接收 (单发送/接收模式时)的时序图
(类型 1: DAP0n = 0, CKP0n = 0)



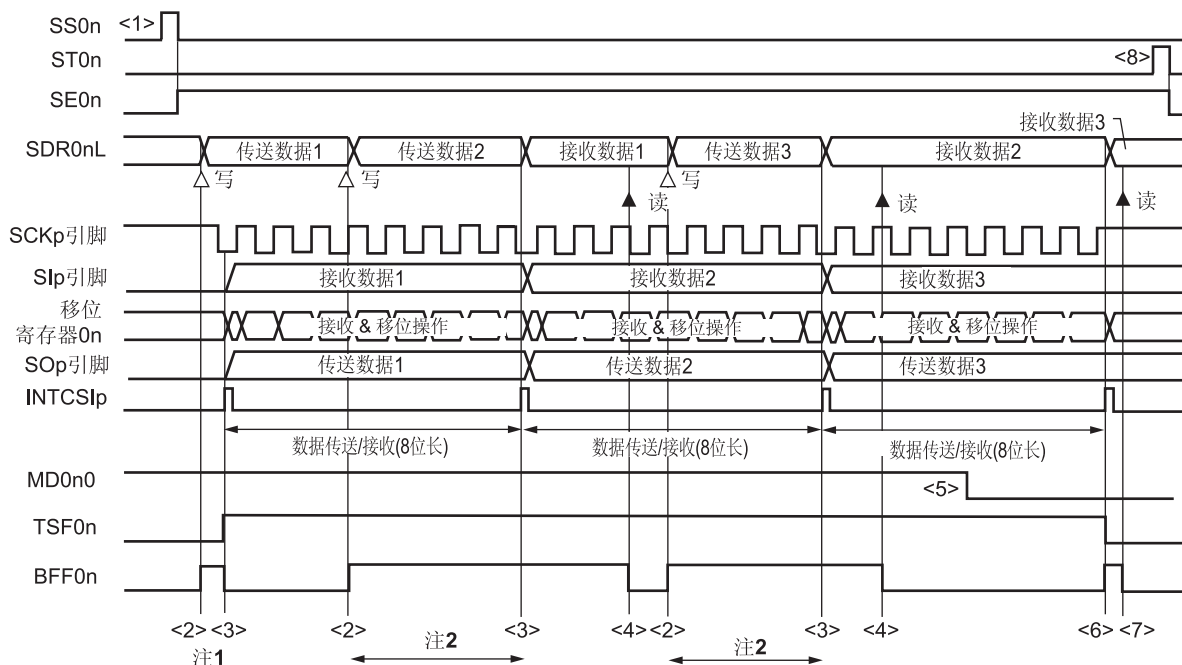
备注 n = 0, p: CSI 编号(p = 00)

图 10-67. 从发送/接收(单发送/接收模式时)的流程图



(4) 处理流程 (连续发送/接收模式时)

图 10-68. 从发送/接收 (连续发送/接收模式时)的时序图
(类型 1: DAP0n = 0, CKP0n = 0)



- 注 1. 如果在串行状态寄存器 0n (SSR0n) 的 BFF0n 位为 1 时将发送数据写入 SDR0nL 寄存器 (将有效数据存储于串行数据寄存器 0n (SDR0nL) 中), 则发送数据将被覆盖。
2. 在此过程中可以通过读取 SDR0nL 寄存器来读取发送数据。此时, 传送操作不受影响。

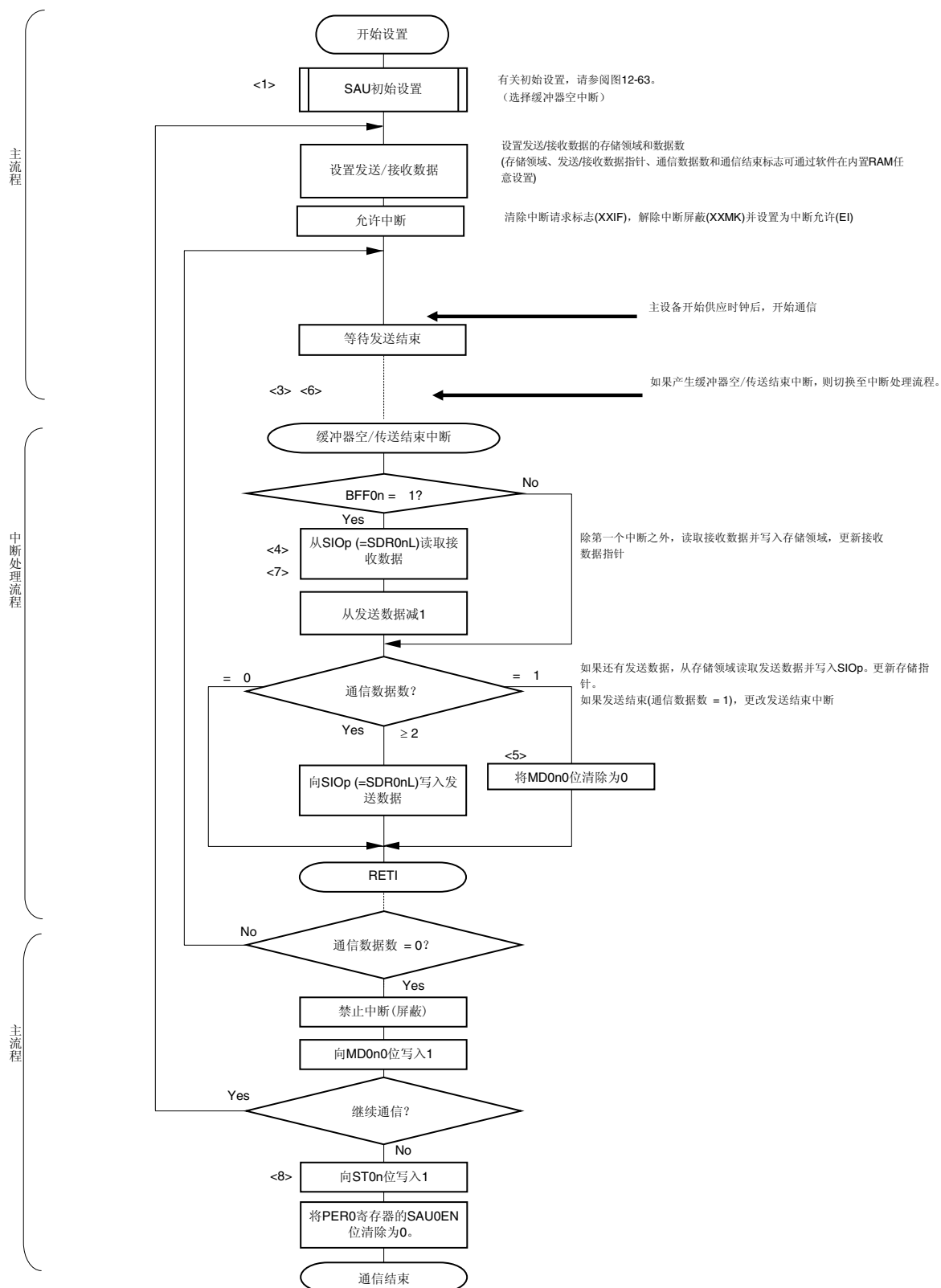
注意事项 即使在操作过程中, 也可改写串行模式寄存器 0n (SMR0nL) 的 MD0n0 位。

但是, 一定要在开始传送最后一位之前改写, 以便在最后发送数据的传送结束中断之前完成改写。

备注 1. 此图中的<1>至<8>对应于图 10-69 从发送/接收(连续发送/接收模式时)的流程图中的<1>至<8>。

2. n = 0, p: CSI 编号(p = 00)

图 10-69. 从发送/接收(连续发送/接收模式时)的流程图



注意事项 必须在主时钟启动之前, 把发送数据设置至 SIOp 寄存器。

备注 此图中的<1>至<8>对应于图 10-68 从发送/接收(连续发送/接收模式时)的时序图中的<1>至<8>。

10.5.7 计算传送时钟频率

三线串行输入/输出 (CSI00)通信的传送时钟频率可以通过以下表达式计算。

(1) 主设备

$$(\text{传送时钟频率}) = \{ \text{对应通道的工作时钟}(\text{f}_{\text{MCK}})\text{频率} \} / (\text{SDRnH}[7:1] + 1) / 2 \text{ [Hz]}$$

(1) 从设备

$$(\text{传送时钟频率}) = \{ \text{由主设备供应的串行时钟}(\text{SCK})\text{频率} \}^{\frac{1}{2}} \text{ [Hz]}$$

注 允许的最大传送时钟频率为 $\text{f}_{\text{CLK}}/6$ 。

备注 SDR0nH[7:1]的值为串行数据寄存器 0n (SDR0nH)的位 7 至位 1 的值(0000000B 至 1111111B)，即 0 至 127。

工作时钟(f_{MCK})决定于串行时钟选择寄存器 0 (SPS0)和串行模式寄存器 0n (SMR0nH)的位 7 (CKS0n)。

10.5.8 3 线串行输入/输出(CSI00)通信过程中发生错误时的处理步骤

在三线串行输入/输出 (CSI00)通信过程中发生错误时的处理步骤如图 10-70 所示。

图 10-70. 发生溢出错误时的处理步骤

软件操作	硬件状态	备注
读取串行数据寄存器 0n (SDR0nL)。 —————→	SSR0n 寄存器的 BFF0n 位被清除为 0, 并允许通道 n 接收数据。	这是为了防止在处理错误的过程中完成了下一次接收时所发生的溢出错误。
读取串行状态寄存器 0n (SSR0n)。		识别出错误类型, 并用读取值来清除错误标志。
将 1 写入串行标志清除触发寄存器 0n (SIR0n)。	错误标志被清除。	只有在读取期间可以清除错误, 其方法为将从 SSR0n 寄存器读取的值不做修改地直接写入 SIR0n 寄存器。

备注 n = 0

10.6 UART(UART0)通信的操作

这是一种使用 2 线：串行数据发送线(TxD)和串行数据接收线(RxD)的异步通信功能。使用这两条通信线路，各数据帧（由一个起始位、数据、奇偶校验位和停止位构成）在单片机与其他通信方之间（以内部波特率）异步传送。全双工 UART 通信可以使用一个发送专用通道（偶数通道）和一个接收专用通道（奇数通道）来实现。

[数据的发送/接收]

- 数据长度为 7 或 8 位
- MSB/LSB 选择
- 发送/接收数据的电平设置和反转选择（选择是否反转电平）
- 附加奇偶校验位和奇偶校验功能
- 停止位附加和停止位校验功能

[中断功能]

- 传送结束中断/缓冲器空中断
- 出现帧错误、奇偶检验错误或溢出错误时的错误中断

[错误检测标志]

- 帧错误、奇偶校验错误、溢出错误

单元	通道	用作 CSI	用作 UART
0	0	CSI00	UART0
	1	—	

注意事项 选择了 UART 操作时，偶数通道仅能用于发送，奇数通道仅能用于接收。

UART 有以下四种类型的通信操作。

- UART 发送(参阅 10.6.1)
- UART 接收(参阅 10.6.2)

10.6.1 UART 发送

UART 发送操作用于将数据从 R7F0C80112ESP, R7F0C80212ESP 异步发送至另一器件（调步同期）。
在用于 UART 的两个通道中，偶数通道用于 UART 发送。

UART	UART0
目标通道	SAU0的通道0
使用引脚	TxD0
中断	INTST0 可选择传送结束中断(单传送模式时)或缓冲器空中断(连续传送模式时)
错误检测标志	无
传送数据长度	7或8位 (仅限UART0)
传送速率	Max. $f_{MCK}/6$ [bps] ($SDR0nH[7:1] = 2$ 或更大, Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] [※]
数据相位	正相输出 (默认: 高电平) 反相输出 (默认: 低电平)
奇偶校验位	以下可选 • 无奇偶校验位 • 附加零校验 • 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	MSB或LSB优先

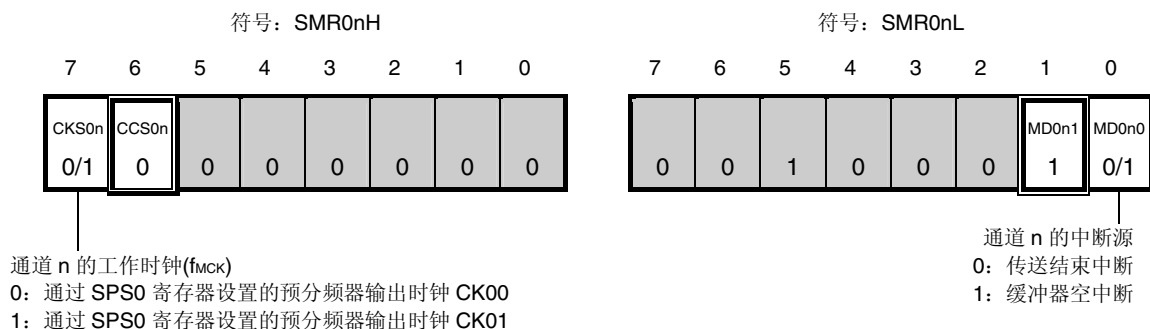
注 在满足以上条件及电特性中的外围功能特性(参阅第二十一章 电特性)的范围内使用此操作。

- 备注 1. f_{MCK} : 目标通道的工作时钟频率
 f_{CLK} : 系统时钟频率
2. n: 通道编号 (n = 0)

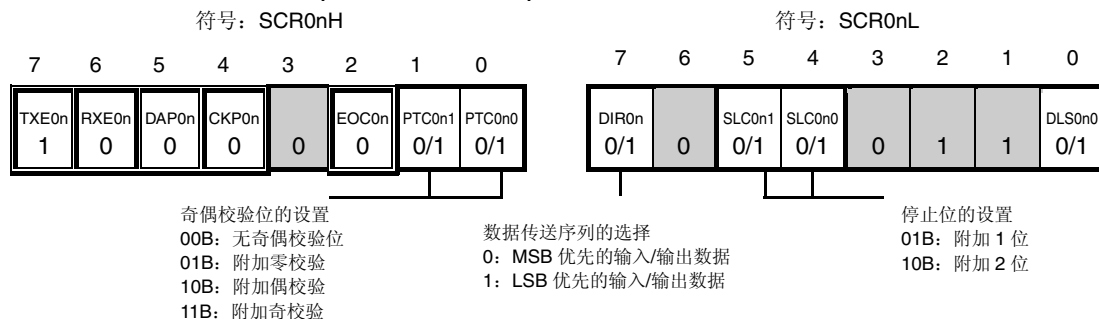
(1) 寄存器设置

图 10-71. UART 发送(UART0)时的寄存器设置内容示例 (1/2)

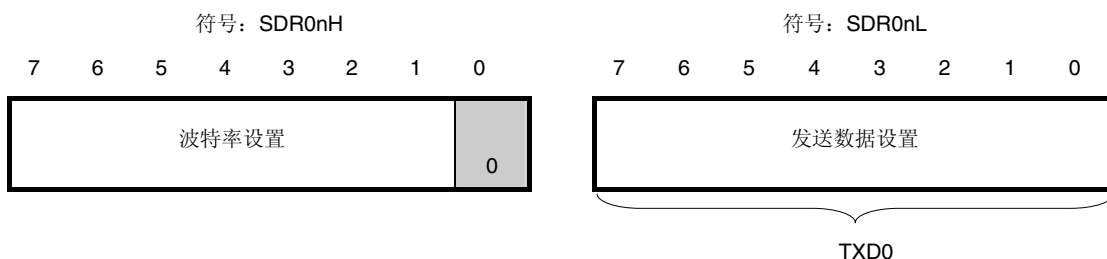
(a) 串行模式寄存器 0n (SMR0nH, SMR0nL)



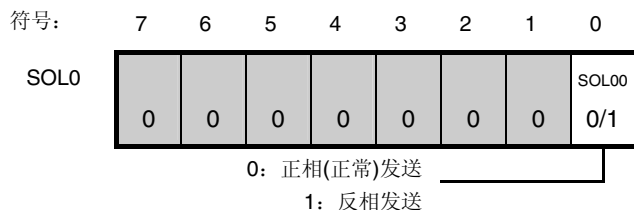
(b) 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)



(c) 串行数据寄存器 0n (SDR0nH, SDR0nL)



(d) 串行输出电平寄存器 0 (SOL0) ... 仅对目标通道的位进行设置。



备注 1. n = 0

2. ☐: 设置固定于 CSI 主发送模式, ☐: 禁止设置 (设为初始值)
0/1: 可根据客户的用途设为 0 或 1

图 10-71. UART 发送(UART0)时的寄存器设置内容示例 (2/2)

(e) 串行时钟输出寄存器 0 (CKO0) ... 仅对目标通道的位进行设置。

符号: 7 6 5 4 3 2 1 0

CKO0								CKO00
	0	0	0	0	0	0	1	×

(f) 串行输出寄存器 0 (SO0) ... 仅对目标通道的位进行设置。

符号: 7 6 5 4 3 2 1 0

SO0								SO00
	0	0	0	0	0	0	1	0/1 ^注

0: 串行数据输出值为 0
1: 串行数据输出值为 1

(g) 串行输出允许寄存器 0 (SOE0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SOE0								SOE00
	0	0	0	0	0	0	0	0/1

(h) 串行通道开始寄存器 0 (SS0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SS0								SS01 SS00
	0	0	0	0	0	0	×	0/1

注 开始发送前, 当目标通道的 SOL00 位为 0 时, 必须设置为 1, 当目标通道的 SOL00 位为 1 时, 则清除为 0。在通信操作过程中, 该值因通信数据而异。

备注1. n = 0

2. ■ : 禁止设置(设置为初始值)

x: 此模式下不能使用的位 (任意模式下均不使用时设置为初始值)

0/1: 可根据用户的用途设为 0 或 1

(2) 操作步骤

图 10-72. UART 发送的初始设置步骤

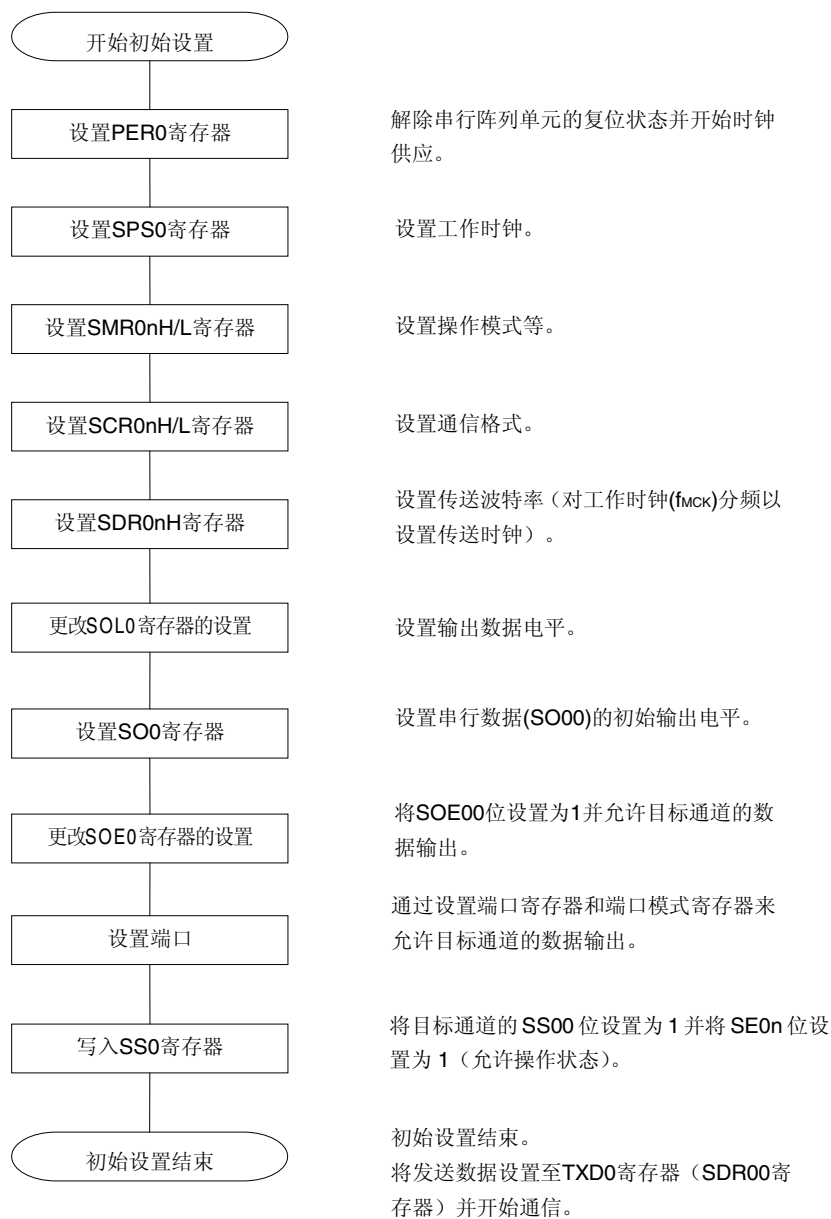


图 10-73. 停止 UART 发送的步骤

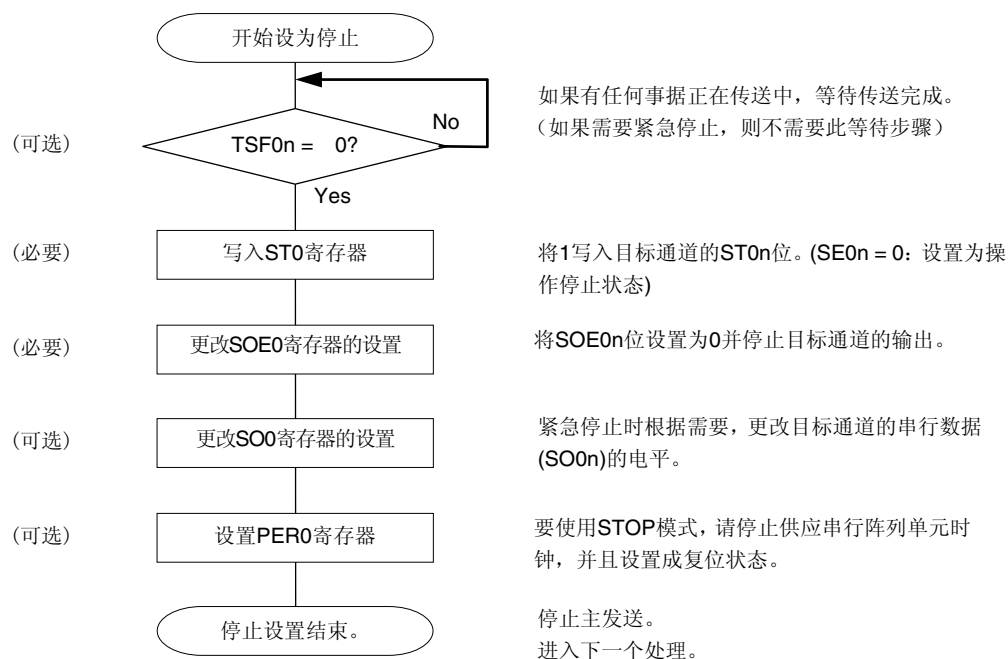


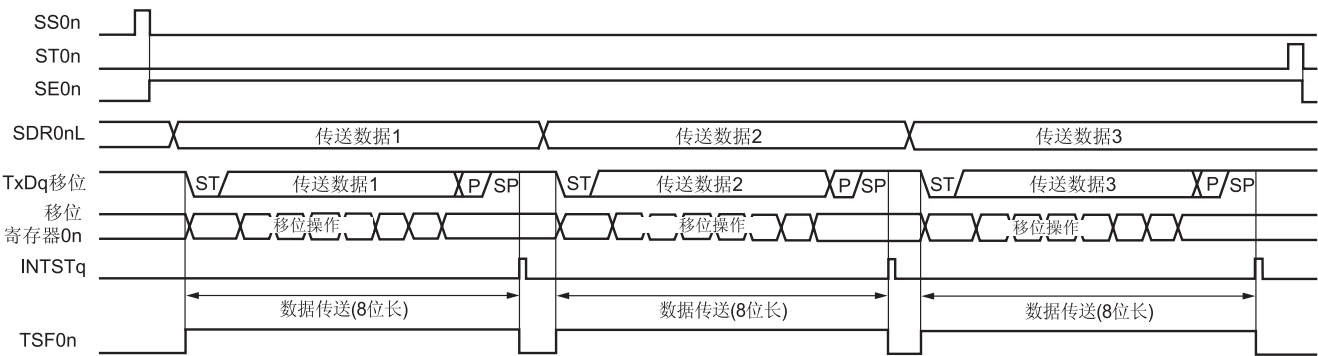
图 10-74. 恢复 UART 发送的步骤



备注 在停止主发送的步骤中改写 PER0 而停止时钟供应时，请等待通信目标停止或者通信结束，然后执行初始设置，而不是重传设置。

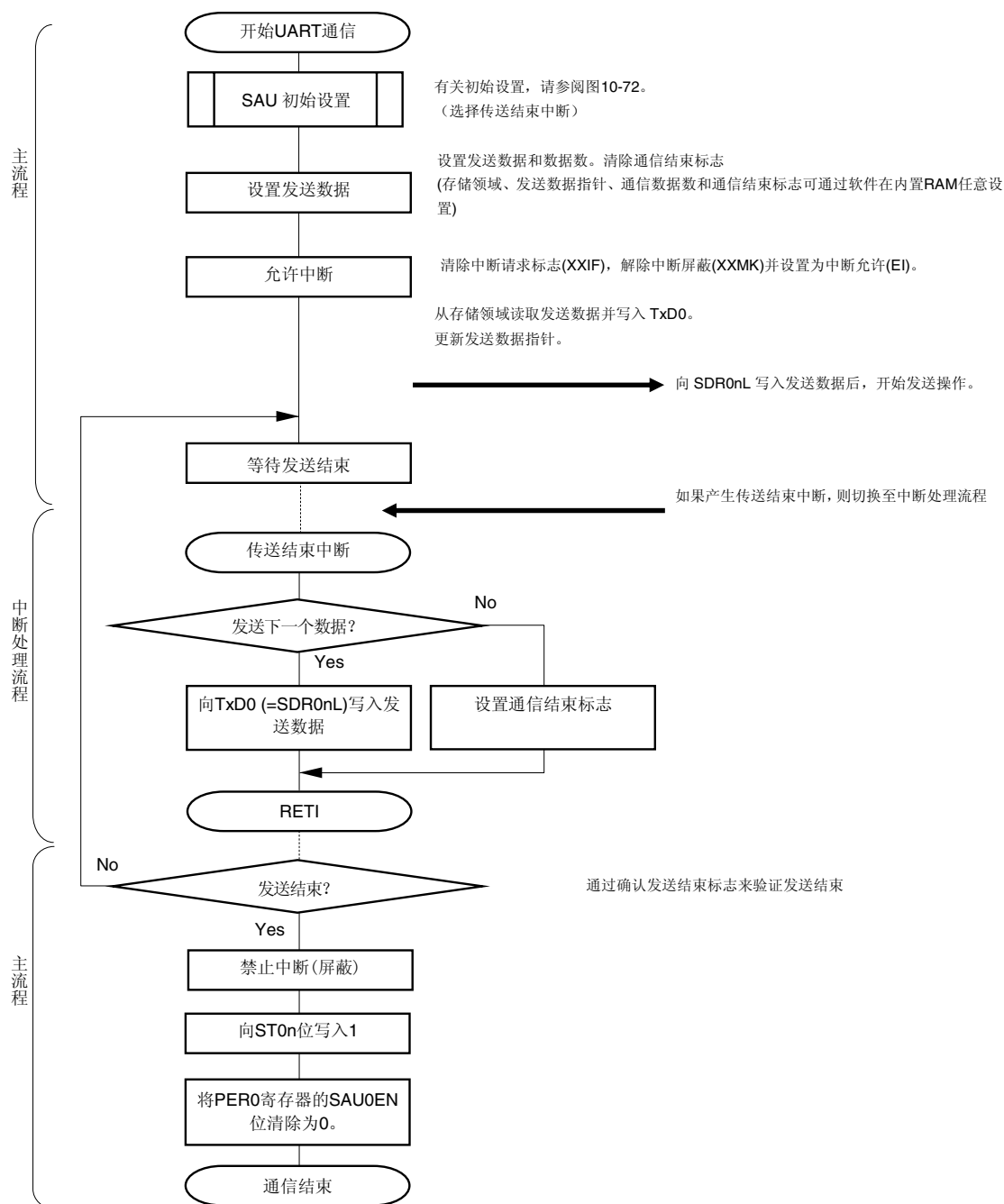
(3) 处理流程 (单发送模式时)

图 10-75. UART 发送(单发送模式时)的时序图



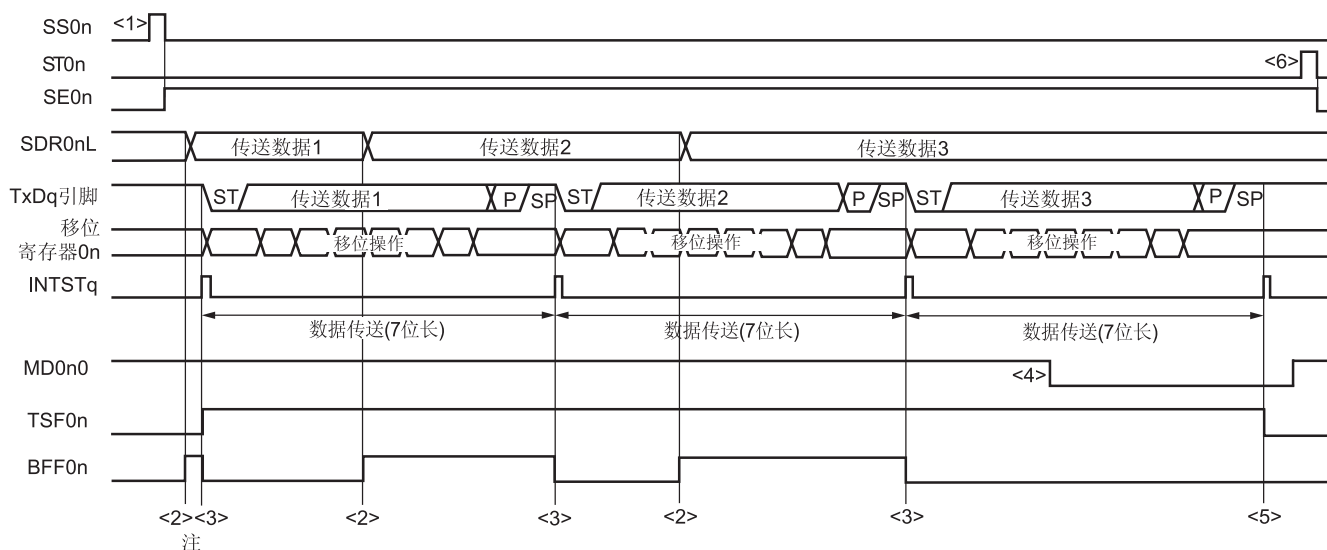
备注 q: UART 编号(q = 0), n = 0

图 10-76. UART 发送(单发送模式时)的流程图



(4) 处理流程 (连续发送模式时)

图 10-77. UART 发送(连续发送模式时)的时序图



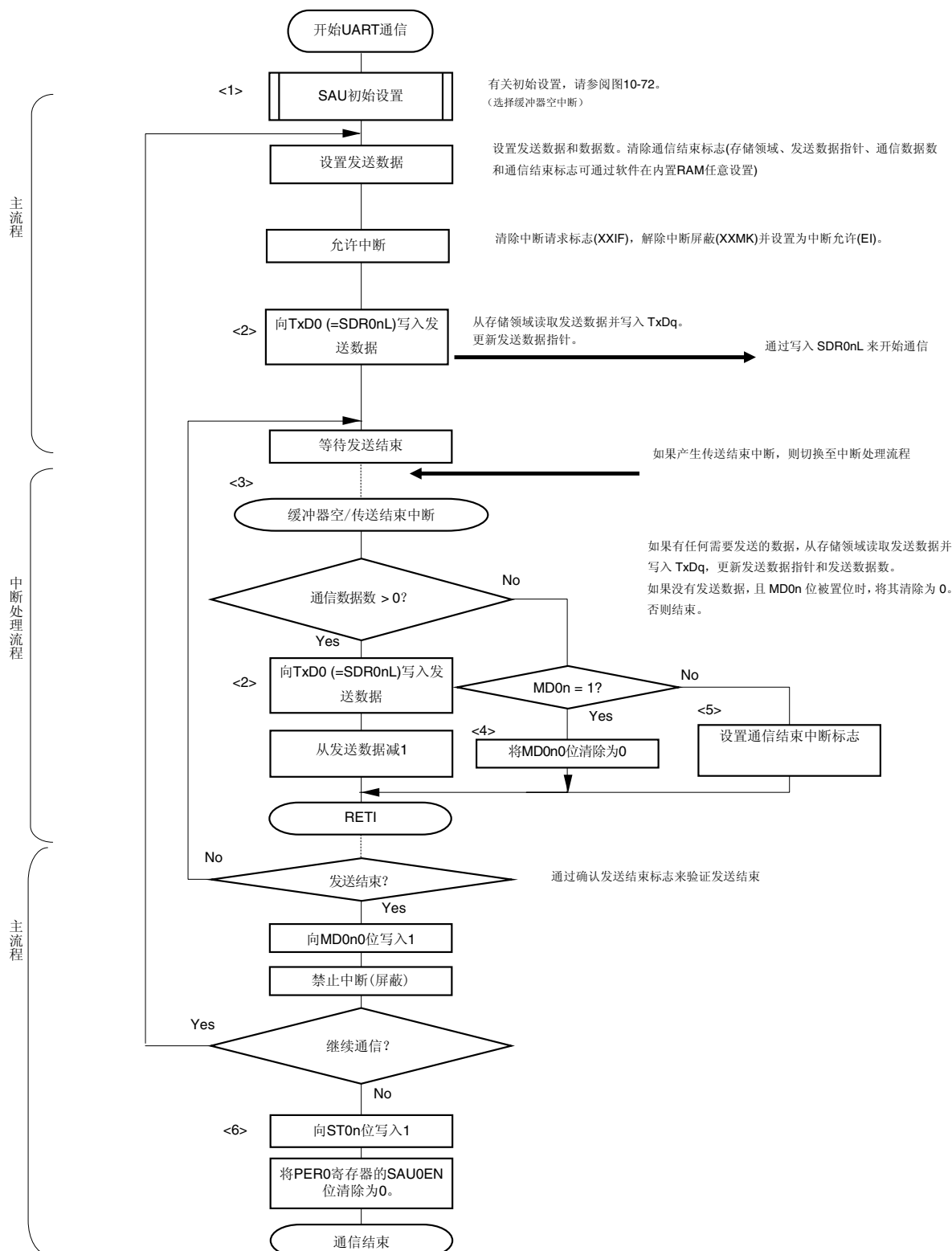
注 如果在串行状态寄存器 0n (SSR0n)的 BFF0n 位为 1 时将发送数据写入 SDR0nL 寄存器 (将有效数据存储于串行数据寄存器 0n (SDR0nL)中), 则发送数据将被覆盖。

注意事项 即使在操作过程中, 也可改写串行模式寄存器 0n (SMR0nL)的 MD0n0 位。

但是必须在开始传送最后一位之前改写, 以便在最后发送数据的传送结束中断前完成改写。

备注 q: UART 编号(q = 0), n = 0

图 10-78. UART 发送(连续发送模式时)的流程图



备注 此图中的<1>至<6>对应于图 10-77. UART 发送(连续发送模式时)的时序图中的<1>至<6>。

10.6.2 UART 接收

UART 接收操作指 R7F0C80112ESP, R7F0C80212ESP 从另一器件异步接收数据（调步同期）。

用于 UART 的两个通道中的奇数通道被用作 UART 接收。必须设置奇数和偶数通道的 SMR 寄存器。

UART	UART0
目标通道	SAU0的通道1
使用引脚	RxD0
中断	INTSR0 仅限传送结束中断(禁止设置缓冲器空中断。)
错误中断	INTSRE0
错误检测标志	- 帧错误检测标志 (FEF0n) - 奇偶校验错误检测标志 (PEF0n) - 溢出错误检测标志 (OVF0n)
传送数据长度	7或8位 (仅限UART0)
传送速率	Max. $f_{MCK}/6$ [bps] (SDR0nH[7:1] = 2或更大), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] ^注
数据相位	正相输出 (默认: 高电平) 反相输出 (默认: 低电平)
奇偶校验位	以下可选 - 无奇偶校验 - 无奇偶指定 (0奇偶) - 附加偶校验 - 附加奇校验
停止位	1位验证
数据方向	MSB或LSB优先

注 在满足以上条件及电特性中的外围特性(参阅第二十一章 电特性)的范围内使用此操作。

备注 1. f_{MCK} : 目标通道的工作时钟频率

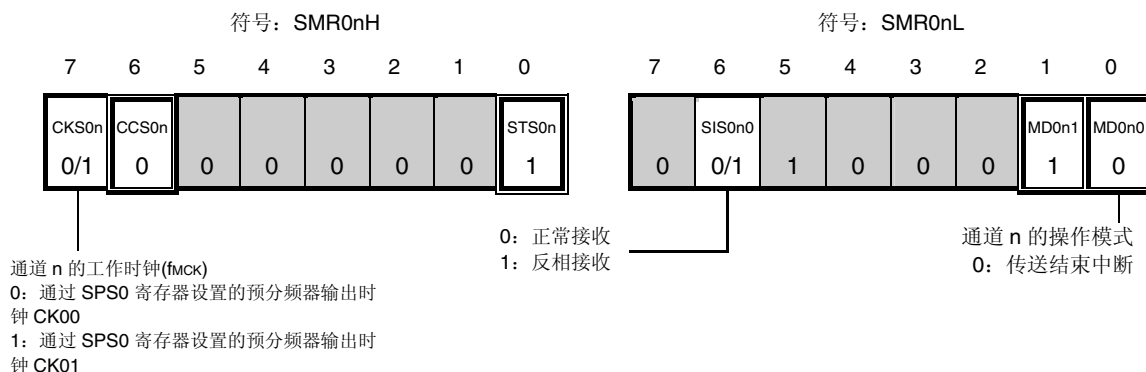
f_{CLK} : 系统时钟频率

2. n: 通道编号 (n = 1)

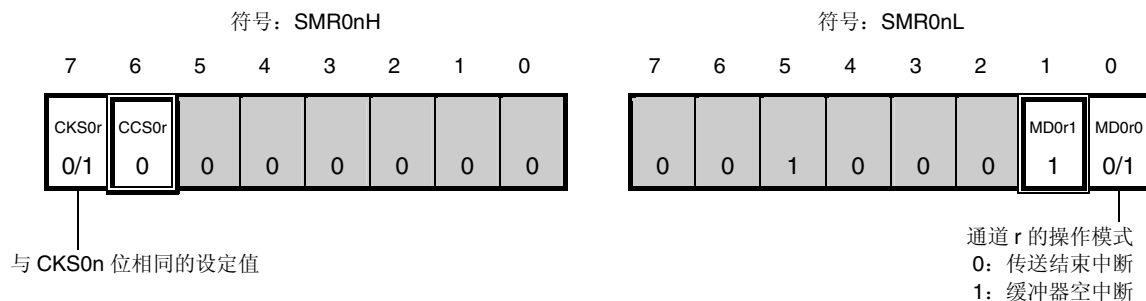
(1) 寄存器设置

图 10-79. UART 接收(UART0) 时的寄存器设置内容示例 (1/2)

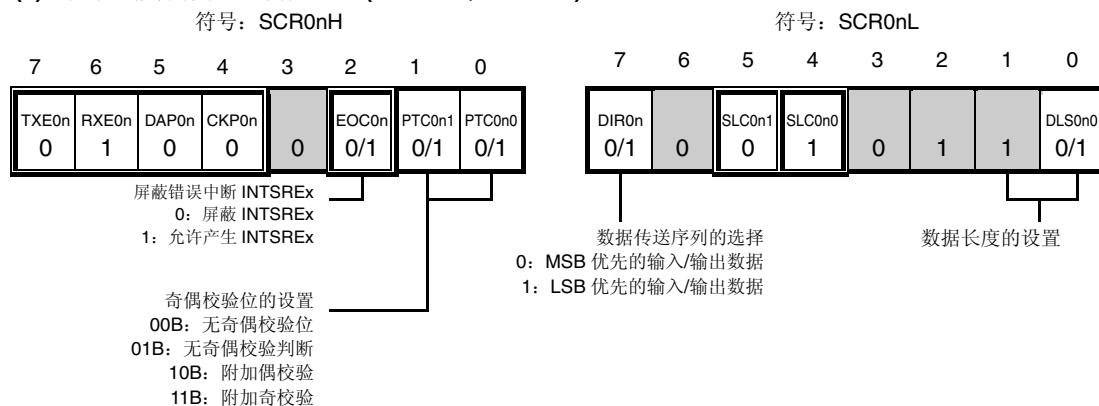
(a) 串行模式寄存器 0n (SMR0nH, SMR0nL)



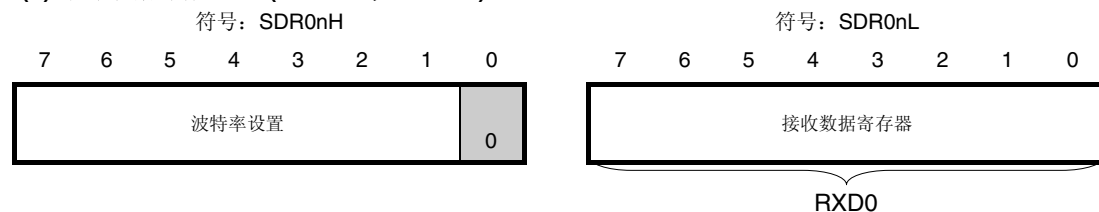
(b) 串行模式寄存器 0r (SMR0rH, SMR0rL)



(c) 串行通信操作设置寄存器 0n (SCR0nH, SCR0nL)



(d) 串行数据寄存器 0n (SDR0nH, SDR0nL)



注意事项 对于 UART 接收, 必须设置通道 r 的 SMR0r 寄存器, 使其与通道 n 配对。

备注 1. n: 通道编号 (n = 1),

r: 通道编号 (r = n - 1) q: UART 编号 (q = 0)

2. □: 设置固定于 UART 主发送模式, ■: 禁止设置 (设为初始值)

0/1: 可根据用户的用途设为 0 或 1

图 10-79. UART 接收(UART0) 时的寄存器设置内容示例 (2/2)

(e) 串行时钟输出寄存器 0 (CKO0) ... 此模式下不使用的寄存器。

符号: 7 6 5 4 3 2 1 0

CKO0								CKO00
	0	0	0	0	0	0	1	×

(f) 串行输出寄存器 0 (SO0) ...此模式下不使用的寄存器。

符号: 7 6 5 4 3 2 1 0

SO0								SO00
	0	0	0	0	0	0	1	×

(g) 串行输出允许寄存器 0 (SOE0) ... 此模式下不使用的寄存器。

符号: 7 6 5 4 3 2 1 0

SOE0								SOE00
	0	0	0	0	0	0	0	×

(h) 串行通道开始寄存器 0 (SS0) ... 仅将目标通道的位设置为 1。

符号: 7 6 5 4 3 2 1 0

SS0							SS01	SS00
	0	0	0	0	0	0	0/1	×

注意事项 对于 UART 接收, 必须设置通道 r 的 SMR0r 寄存器, 使其与通道 0 配对。

备注 1. n: 通道编号 (n = 1)

 r: 通道编号(r = n - 1) q: UART 编号(q = 0)

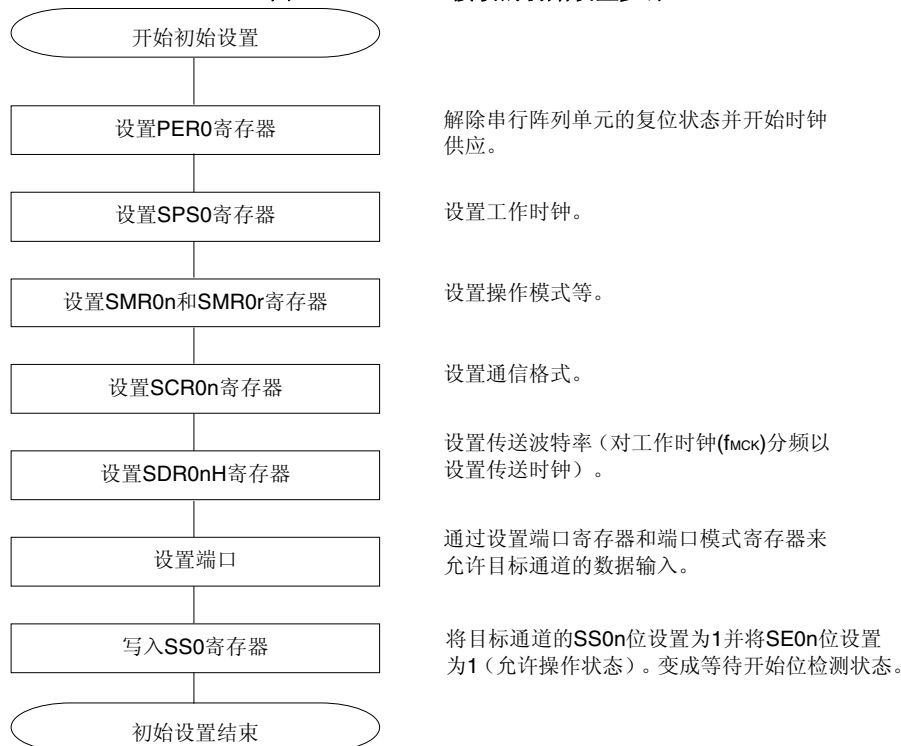
2.  : 禁止设置(设置为初始值)

 x: 此模式下不能使用的位 (任意模式下均不使用时设为初始值)

 0/1: 可根据客户的用途设为 0 或 1

(2) 操作步骤

图 10-80. UART 接收的初始设置步骤



注意事项 在设置 SCR0n 寄存器的 RXE0n 位为 1 后，必须在经过 4 个或更多 f_{CLK} 时钟后设置 SS0n 寄存器为 1。

图 10-81. 停止 UART 接收的步骤

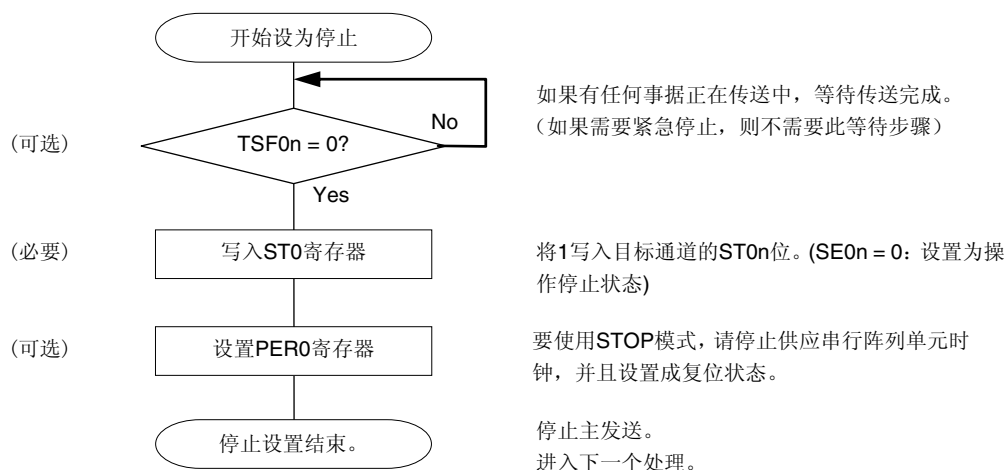
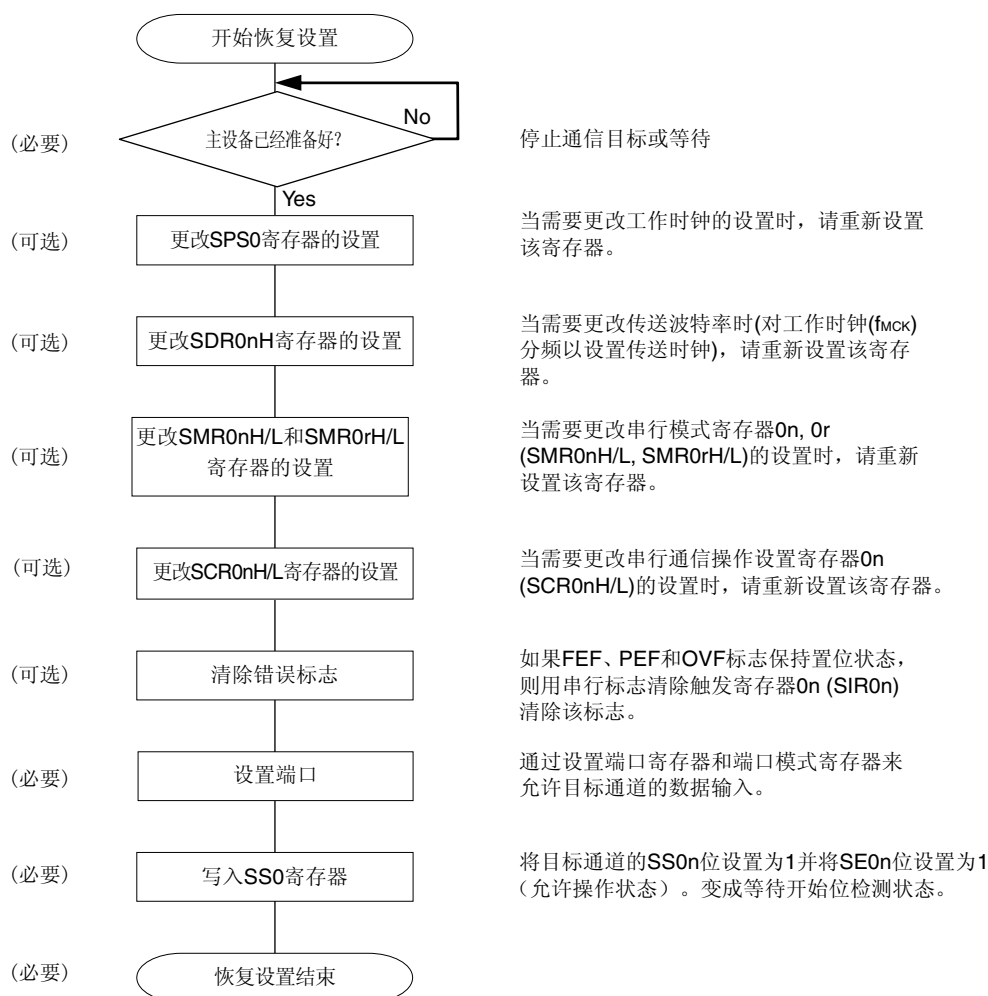


图 10-82. 恢复 UART 接收的步骤

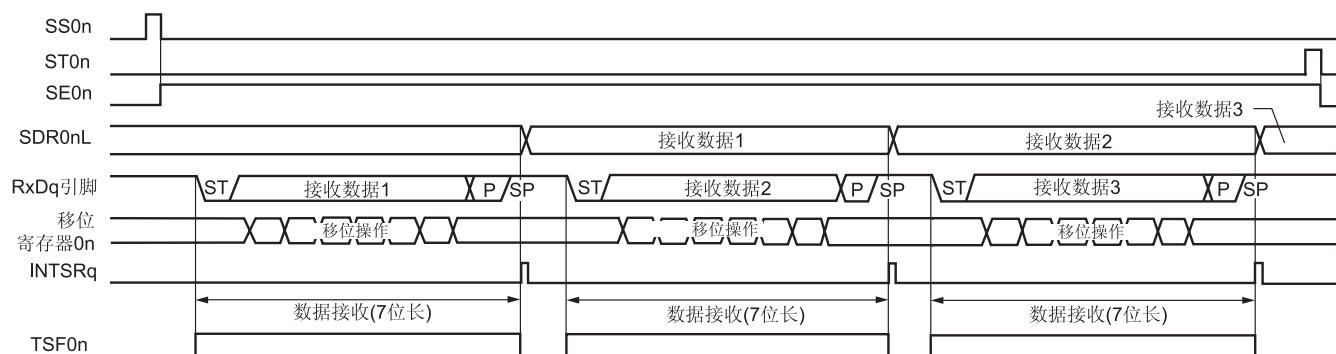


注意事项 在设置 SCR0n 寄存器的 RXE0n 位为 1 后，必须在经过 4 个或更多 f_{CLK} 时钟后设置 SS0n 寄存器为 1。

备注 在停止主发送的步骤中改写 PER0 而停止时钟供应时，请等待通信目标（从设备）停止或者通信结束，然后执行初始设置，而不是重传设置。

(3) 处理流程

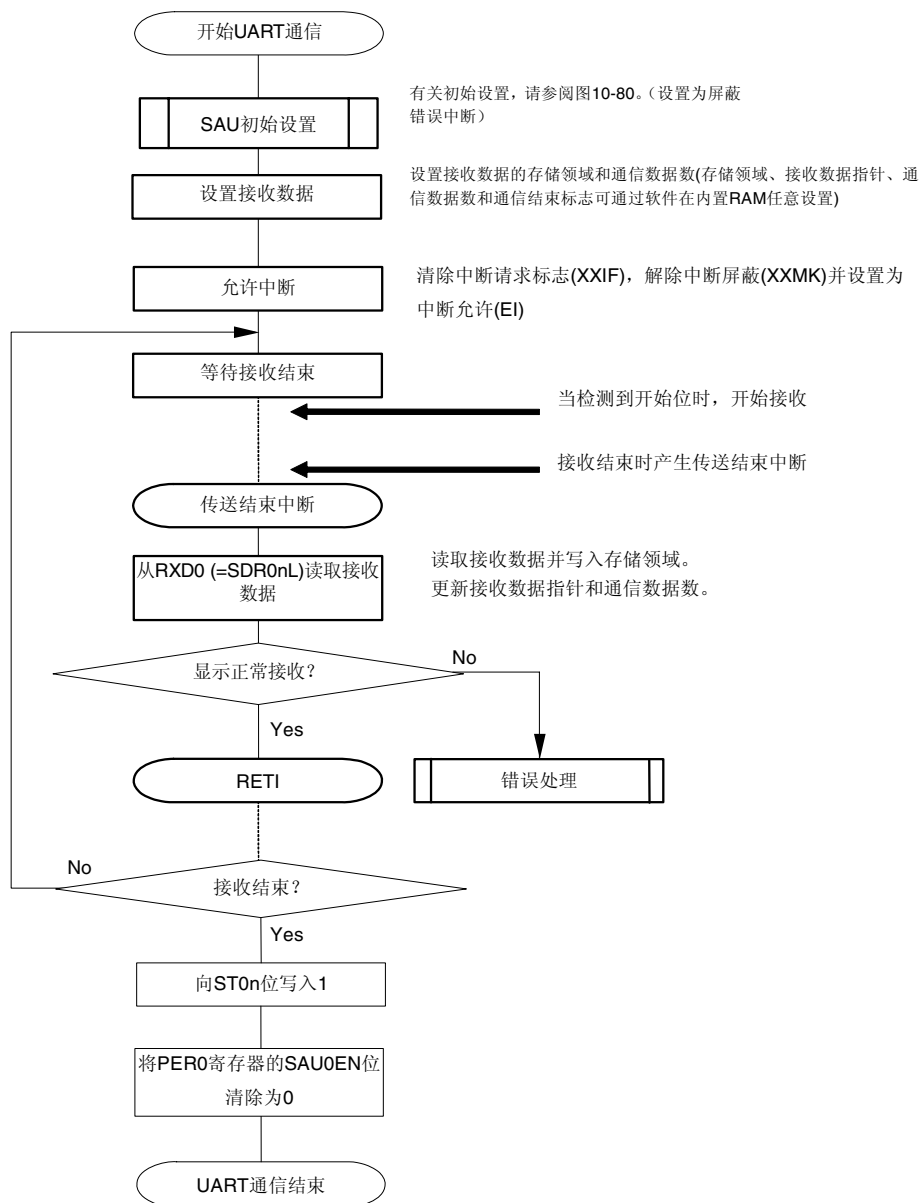
图 10-83. UART 接收的时序图



备注 n: 通道编号 (n = 1)

r: 通道编号 (r = n - 1) q: UART 编号 (q = 0)

图 10-84. UART 接收的流程图



10.6.3 计算波特率

(1) 波特率计算公式

用于 UART (UART0)通信的波特率可以通过以下表达式计算。

$$(\text{波特率}) = \{\text{目标通道的工作时钟}(f_{\text{MCK}})\text{频率}\} \div (\text{SDR0nH}[7:1] + 1) \div 2 [\text{bps}]$$

注意事项 禁止将串行数据寄存器 0n (SDR0nH)设置为 SDR0nH[7:1] = (0000000B 至 0000001B)。

备注 1. 当使用 UART 时, SDR0nH[7:1]的值为 SDR0nH 寄存器位 15 至位 9 的值(0000010B 至 1111111B), 即 2 至 127。

2. n = 0, 1

工作时钟(f_{MCK}) 取决于串行时钟选择寄存器 0 (SPS0)以及串行模式寄存器 0n (SMR0nH)的位 7 (CKS0n 位)。

(2) 发送期间的波特率误差

发送过程中 UART(UART0)通信的波特率误差可以用以下计算式算出。确保发送侧的波特率处于接收侧允许的波特率范围之内。

$$(\text{波特率误差}) = (\text{计算波特率值}) \div (\text{目标波特率}) \times 100 - 100 [\%]$$

$f_{\text{CLK}} = 20 \text{ MHz}$ 时的 UART 波特率的设置示例如下所示。

UART 波特率 (目标波特率)	$f_{\text{CLK}} = 20 \text{ MHz}$			
	工作时钟(f_{MCK})	SDR0nH[7:1]	计算波特率	与目标波特率之间的误差
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 %

(3) 接收时的波特率允许范围

UART (UART0)通信过程中允许接收的波特率范围可以用以下计算式算出。确保发送侧的波特率处于接收侧允许的波特率范围之内。

$$(\text{最大可接收波特率}) = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$$

$$(\text{最小可接收波特率}) = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$$

Brate: 接收方的计算波特率值 (参阅 10.6.3 (1) 波特率计算公式。)

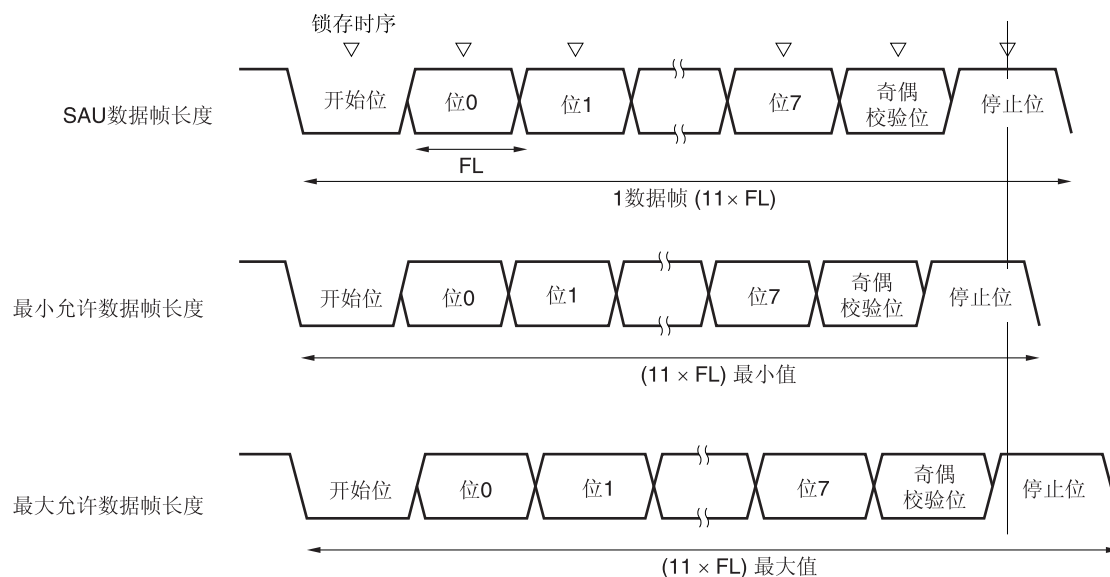
k: SDR0nH[7:1] + 1

Nfr: 1 数据帧长度 [位]

= (开始位) + (数据长度) + (奇偶校验位) + (停止位)

备注 n = 1

图 10-85. 接收时的允许波特率范围 (1 数据帧长度 = 11 位时)



如图 10-85 所示，接收数据的锁存时序取决于检测到起始位之后由串行数据寄存器 0nH (SDR0nH)的位 7 至位 1 设置的分频比。如果在该锁存时间之前收到最后数据（停止位），则可以正常进行接收。

10.6.4 UART(UART0)通信过程中发生错误时的处理步骤

在 UART (UART0)通信过程中发生的错误的处理步骤如图 10-86 和 10-87 所示。

图 10-86. 发生奇偶校验错误或溢出错误时的处理步骤

软件操作	硬件状态	备注
读取串行数据寄存器 0n (SDR0nL)。→	SSR0n 寄存器的 BFF0n 位被清除为 0，并允许通道 n 接收数据。	这是为了防止在处理错误的过程中完成了下一次接收时所发生的溢出错误。
读取串行状态寄存器 0n (SSR0n)。		识别出错误类型，并用读取值来清除错误标志。
将 1 写入串行标志清除触发寄存器 0n (SIR0n)。	错误标志被清除。	只有在读取期间可以清除错误，其方法为将从 SSR0n 寄存器读取的值不做修改地直接写入 SIR0n 寄存器。

图 10-87. 发生帧错误时的处理步骤

软件操作	硬件状态	备注
读取串行数据寄存器 0n (SDR0nL)。→	SSR0n 寄存器的 BFF0n 位被清除为 0，并允许通道 n 接收数据。	这是为了防止在处理错误的过程中完成了下一次接收时所发生的溢出错误。
读取串行状态寄存器 0n (SSR0n)。		识别出错误类型，并用读取值来清除错误标志。
写入串行标志清除触发寄存器 0n (SIR0n)。	错误标志被清除。	只有在读取期间可以清除错误，其方法为将从 SSR0n 寄存器读取的值不做修改地直接写入 SIR0n 寄存器。
将串行通道停止寄存器 0 (ST0)的 ST0n 位设置为 1。	串行通道允许状态寄存器 0 (SE0)的 SE0n 位被清除为 0，通道 n 停止工作。	
与其他通信方同步		起始位的偏移会使系统判断为出现了帧错误，因此需要重建与其他通信方的同步并恢复通信。
将串行通道开始寄存器 0 (SS0)的 SS0n 位设置为 1。	串行通道允许状态寄存器 0 (SE0)的 SE0n 位被设置为 1，并允许操作通道 n。	

备注 n = 0, 1

第十一章 中断功能

中断功能是指可将执行中的程序切换为其他所需处理的一种功能。分支处理结束后，返回到原来中断了的执行程序。

		中断源数量
可屏蔽中断	外部	3
	内部	8

11.1 中断功能的类型

中断功能有以下两种。

(1) 可屏蔽中断

接受屏蔽控制的中断。通过设置优先级选择标志寄存器(PR00L、PR00H、PR10L、PR10H)，可以将可屏蔽中断的优先级分成四组。

发生高优先级中断时，对低优先级中断可以执行嵌套中断处理。如果同时产生优先级相同的两个或多个中断请求，则根据向量中断处理的优先级进行处理。关于优先级顺序，请参阅表 11-1。

产生待机解除信号，解除 STOP 和 HALT 模式。

可屏蔽中断分为外部中断请求和内部中断请求。

(2) 软件中断

这是通过执行 BRK 指令而产生的向量中断。即使在禁止中断时也能得到受理。软件中断不受中断优先级控制。

11.2 中断源和配置

中断源包括可屏蔽中断和软件中断。另外还具有最多四种的重启源(参阅表 11-1)。当产生复位或各种中断请求而分支时，因为存储着程序起始地址的向量代码各为两个字节，因此，中断跳转目标的地址为 00000H 至 0FFFFH 的 64 K 地址。

表 11-1. 中断源列表

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			
可屏蔽	0	INTWDTI	看门狗定时器间隔 (溢出时间的 75%+3/4f _{IL})	内部	0004H	(a)
	1	INTP0	检测引脚输入边沿	外部	0006H	(b)
	2	INTP1			0008H	
	3	INTST0/ INTCSI00/	UART0 发送的传送结束或缓冲器空中断 /CSI00 传送结束或缓冲器空中断	内部	000AH	(a)
	4	INTSR0	UART0 接收传送结束		000CH	
	5	INTSRE0	发生 UART0 接收通信错误		000EH	
	6	INTTM01H	定时器通道 1 的计数结束或操作开始 (高 8 位定时器工作时)		0010H	
	7	INTTM00	定时器通道 0 的计数结束或操作开始		0012H	
	8	INTTM01	定时器通道 1 的计数结束、捕捉完成或操作开始 (16 位定时器工作时或者低 8 位定时器工作时)		0014H	
	9	INTAD	A/D 转换结束		0016H	
	10	INTKR	检测按键返回信号	外部	0018H	(c)
软件	-	BRK	执行 BRK 指令	-	007EH	(d)
复位	-	RESET	RESET 引脚输入	-	0000H	-
		SPOR	可选择的上电复位			
		WDT	看门狗定时器的溢出			
		TRAP	执行非法指令 ^{注3}			

注 1. 如果同时发生两个或更多可屏蔽中断，默认优先级将决定中断的顺序。0 表示最高优先级，10 表示最低优先级。

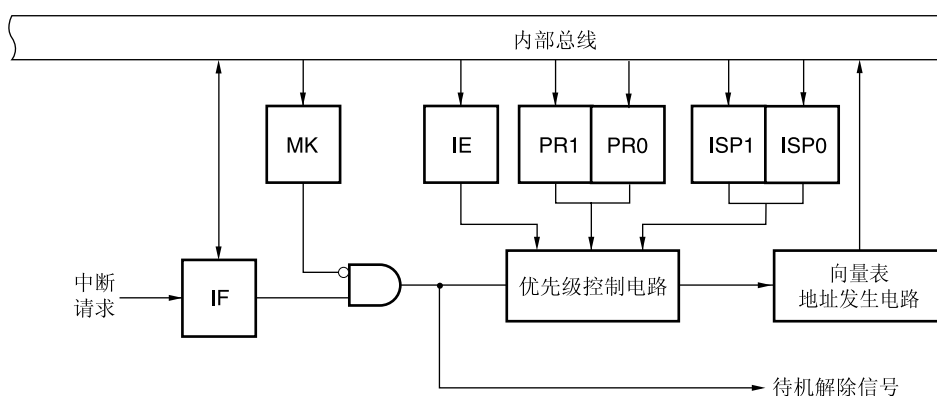
2. 基本配置类型(a)至(d)对应于图 11-1 中的(a)至(d)。

3. 执行指令代码 FFH 时产生。

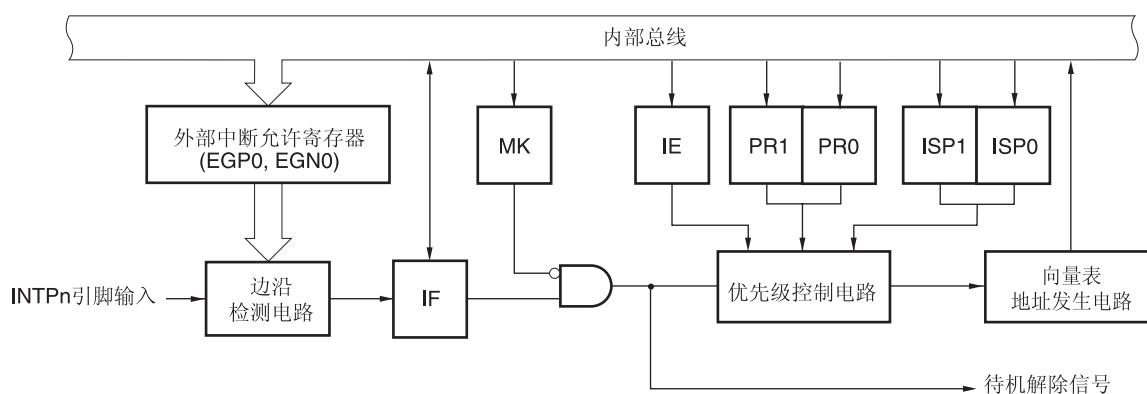
通过片上调试仿真器进行仿真时，不会因执行非法指令发生内部复位。

图 11-1. 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)



IF: 中断请求标志

IE: 中断允许标志

ISP0: 受理中断的优先级标志 0

ISP1: 受理中断的优先级标志 1

MK: 中断屏蔽标志

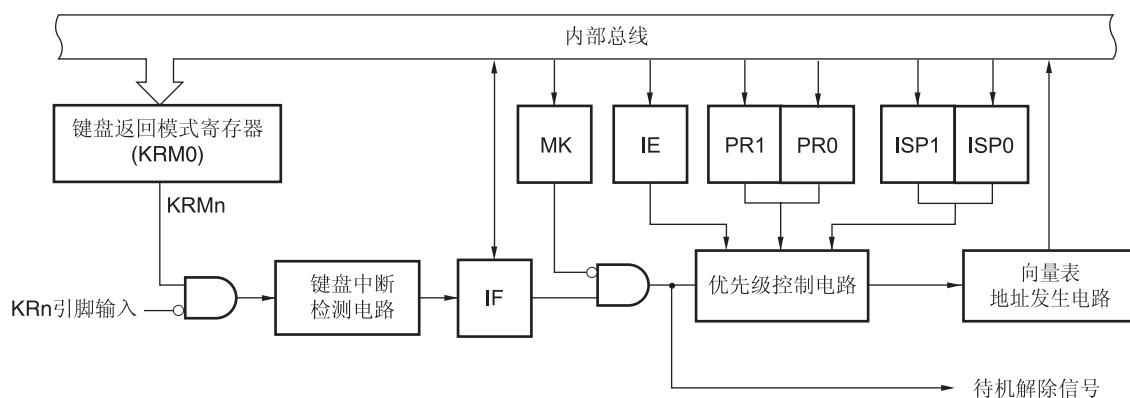
PR0: 优先级指定标志 0

PR1: 优先级指定标志 1

备注 n = 0, 1

图 11-1. 中断功能的基本配置 (2/2)

(c) 外部可屏蔽中断(INTKR)



IF: 中断请求标志

IE: 中断允许标志

ISP0: 受理中断的优先级标志 0

ISP1: 受理中断的优先级标志 1

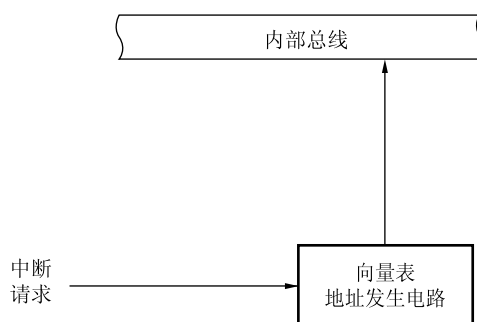
MK: 中断屏蔽标志

PR0: 优先级指定标志 0

PR1: 优先级指定标志 1

备注 n = 0 至 5

(d) 软件中断



11.3 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)
- 中断屏蔽标志寄存器 (MK0L, MK0H)
- 优先级选择标志寄存器 (PR00L, PR00H, PR10L, PR10H)
- 外部中断上升沿允许寄存器 (EGP0)
- 外部中断下降沿允许寄存器 (EGN0)
- 程序状态字 (PSW)

各中断请求源对应的中断请求标志、中断屏蔽标志和优先级指定标志如表 11-2 所示。

表 11-2. 中断请求源对应的标志

中断源	中断请求标志		中断屏蔽标志		优先指定标志	
		寄存器		寄存器		寄存器
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTP0	PIF0		PMK0		PPR00, PPR10	
INTP1	PIF1		PMK1		PPR01, PPR11	
INTST0 ^注	STIF0 ^注		STMK0 ^注		STPR00, STPR10 ^注	
INTCSI00 ^注	CSIIF00 ^注		CSIMK00 ^注		CSIPR000, CSIPR100 ^注	
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	
INTTM01H	TMIF01H		TMMK01H		TMPR001H, TMPR101H	
INTTM00	TMIF00	IF0H	TMMK00	MK0H	TMPR000, TMPR100	PR00H PR10H
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTAD	ADIF		ADMK		ADPR0, ADPR1	
INTKR	KRIF		KRMK		KRPR0, KRPR1	

注 两个中断源 INTST0 和 INTCSI00 中，如果产生任意一个，则 IF0L 寄存器的位 3 将被设置为 1。

MK0L、PR00L 和 PR10L 寄存器的位 3 支持这两种中断源。

11.3.1 中断请求标志寄存器 (IF0L, IF0H)

在产生相应的中断请求或者执行指令时，中断请求标志被设置为 1。在受理中断请求、产生复位信号、或者执行指令时，该寄存器被清除为 0。

当一个中断被受理时，中断请求标志将自动清零，然后进行中断处理。

使用 1 位或 8 位存储器操作指令设置 IF0L 和 IF0H 寄存器。产生复位信号后，这些寄存器被清除为 00H。

备注 执行向该寄存器写入数据的指令时，该指令执行时钟数将增加 2 个时钟。

图 11-2. 中断请求标志寄存器 (IF0L, IF0H) 的格式

地址: FFFE0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	TMIF00	TMIF01H	SREIF0	SRIF0	STIF0 CSIF00	PIF1	PIF0	WDTIIF

地址: FFFE1H 复位后: 00H R/W

符号	7	6	5	4	3	<2>	<1>	<0>
IF0H	0	0	0	0	0	KRIF	ADIF	TMIF01

XXIFXX	中断请求标志
0	不产生中断请求信号
1	产生中断请求，处于中断请求状态

- 注意事项 1. 请勿更改不定位的数值。
2. 在操作中中断请求标志寄存器的标志时，要使用位操作指令 (CLR1)。当使用 C 语言描述时，由于编译后的汇编程序必须为位操作指令 (CLR1)，所以要使用位操作指令，如 IF0L.0 = 0 或 `_asm("clr1 IF0L, 0")` 进行操作。

如果以 C 语言描述程序（例如 `IF0L &= 0xfe`），并进行编译，则将成为有三条指令的汇编程序。

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

在这种情况下，即使在 `mov a, IF0L` 与 `mov IF0L, a` 之间将同一中断请求标志寄存器 (IF0L) 的另一位的请求标志设置为 1，该标志也会在 `mov IF0L, a` 时被清除为 0。

11.3.2 中断屏蔽标志寄存器 (MK0L, MK0H)

中断屏蔽标志用于允许/禁止相应的可屏蔽中断处理。

使用 1 位或 8 位存储器操作指令设置 MK0L 和 MK0H 寄存器。产生复位信号后，该寄存器被设置为 FFH。

备注 执行向该寄存器写入数据的指令时，该指令执行时钟数将增加 2 个时钟。

图 11-3. 中断屏蔽标志寄存器 (MK0L, MK0H) 的格式

地址: FFFE4H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	TMMK00	TMMK01H	SREMK0	SRMK0	STMK0 CSIMK00	PMK1	PMK0	WDTIMK

地址: FFFE5H 复位后: FFH R/W

符号	7	6	5	4	3	<2>	<1>	<0>
MK0H	1	1	1	1	1	KRMK	ADMK	TMMK01

XXMKXX	中断处理的控制
0	允许中断处理
1	禁止中断处理

注意事项 请勿更改不定位的数值。

11.3.3 优先级选择标志寄存器 (PR00L, PR00H, PR10L, PR10H)

优先级选择标志寄存器用于设置相应的可屏蔽中断优先级。

通过组合使用 PR0xy 和 PR1xy 寄存器 (0xy = 0L, 0H) 来设置优先级。

使用 1 位或 8 位存储器操作指令设置 PR00L、PR00H、PR10L 和 PR10H 寄存器。

备注 执行向该寄存器写入数据的指令时，该指令执行时钟数将增加 2 个时钟。

图 11-4. 优先级选择标志寄存器 (PR00L, PR00H, PR10L, PR10H) 的格式

地址: FFFE8H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	TMPR000	TMPR001H	SREPR00	SRPR00	STPR00 CSIPR000	PPR01	PPR00	WDTIPR0

地址: FFECH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	TMPR100	TMPR101H	SREPR10	SRPR10	STPR10 CSIPR100	PPR11	PPR10	WDTIPR1

地址: FFE9H 复位后: FFH R/W

符号	7	6	5	4	3	<2>	<1>	<0>
PR00H	1	1	1	1	1	KRPR0	ADPR0	TMPR001

地址: FFFEDH 复位后: FFH R/W

符号	7	6	5	4	3	<2>	<1>	<0>
PR10H	1	1	1	1	1	KRPR1	ADPR1	TMPR101

XXPR1X	XXPR0X	选择优先等级
0	0	指定等级 0 (高优先等级)
0	1	指定等级 1
1	0	指定等级 2
1	1	指定等级 3 (低优先等级)

注意事项 请勿更改不定位的数值。

11.3.4 外部中断上升沿允许寄存器 (EGP0)，外部中断下降沿允许寄存器 (EGN0)

这些寄存器用于设定 INTP0 和 INTP1 的有效边沿。

使用 1 位或 8 位存储器操作指令设置 EGP0 和 EGN0 寄存器。

产生复位信号后，这些寄存器被清除为 00H。

图 11-5. 外部中断上升沿允许寄存器 (EGP0) 和外部中断下降沿允许寄存器的格式 (EGN0)

地址: FFF38H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGP0	0	0	0	0	0	0	EGP1	EGP0

地址: FFF39H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGN0	0	0	0	0	0	0	EGN1	EGN0

EGPn	EGNn	选择 INTPn 引脚的有效边沿 (n = 0, 1)
0	0	禁止检测边沿
0	1	下降沿
1	0	上升沿
1	1	上升和下降沿

注意事项 将外部中断功能使用的输入端口切换至输出模式时，可能会检测到有效边沿而产生 INTPn 中断。要切换至输出模式时，请先设置为禁止检测边沿（EGPn, EGNn = 0, 0），然后再将端口模式寄存器 (PMxx) 设置为 0。

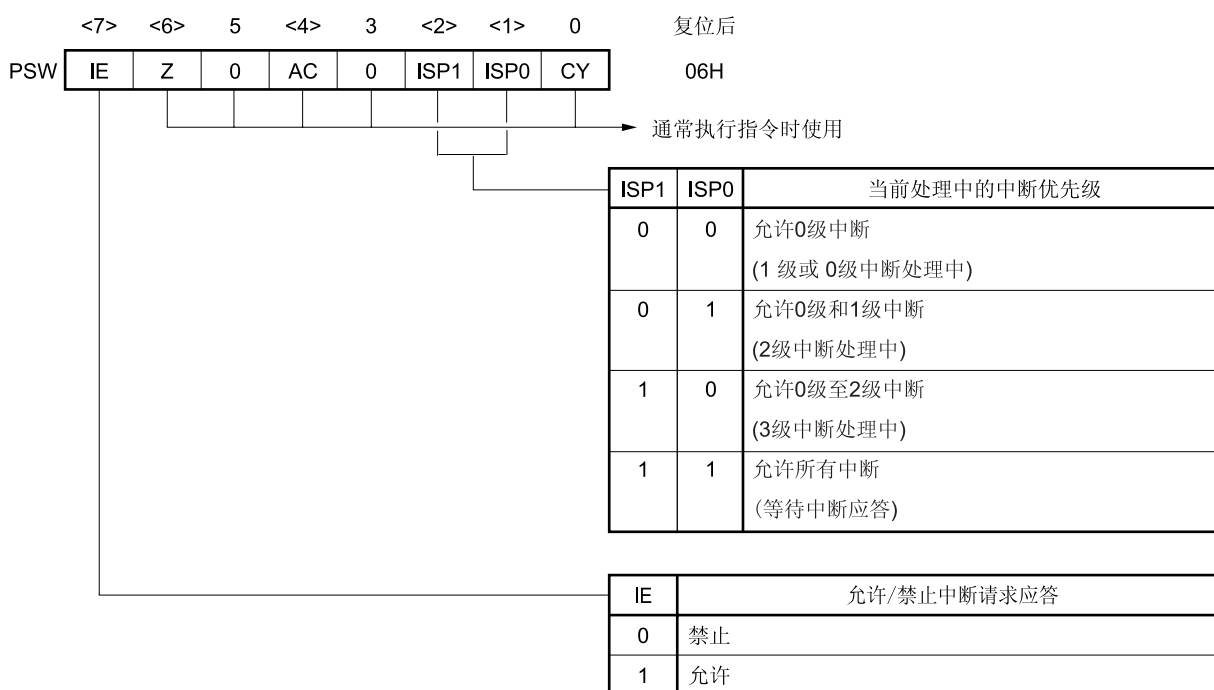
11.3.5 程序状态字(PSW)

程序状态字为用于保存指令执行结果和中断请求当前状态的寄存器。设置允许/禁止可屏蔽中断的 IE 标志以及控制嵌套中断处理的 ISP0 和 ISP1 标志被映射至 PSW。

除 8 位读/写以外，该寄存器还可以使用位操作指令和专用指令（EI 和 DI）来执行操作。此外，受理向量中断请求及执行 BRK 指令时，PSW 的内容将被自动保存至堆栈，且 IE 标志复位至 0。如果可屏蔽中断请求被受理，则被受理中断的优先级指定标志的内容将被传送至 ISP0 和 ISP1 标志。PSW 的内容也可通过 PUSH PSW 指令保存至堆栈。这些内容可用 RETI、RETB 和 POP PSW 指令从堆栈中恢复。

产生复位信号后，PSW 被设置为 06H。

图 11-6. 程序状态字的配置



11.4 中断处理的操作

11.4.1 可屏蔽中断请求受理

当中断请求标志被设置为 1 且与该中断请求对应的屏蔽(MK)标志被清除为 0 时, 该可屏蔽中断可被受理。在中断允许状态(当 IE 标志被设置为 1)下, 向量中断处理被受理。但是, 在处理较高优先级中断请求期间, 不受理低优先级向量中断请求。

从产生可屏蔽中断请求到执行向量中断处理的时间如下面的表 11-3 所示。

关于中断请求受理时序, 请参阅图 11-8 和 11-9。

表 11-3. 从产生可屏蔽中断到处理的时间

	最短时间	最长时间 ^注
处理时间	11 个时钟	18 个时钟

注 执行内部 RAM 区的指令时, 不适用以上的最长时间。

备注 1 个时钟: $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

如果同时产生两个或更多可屏蔽中断请求, 则优先级指定标志所指定的优先级较高的请求将被首先受理。如果两个或更多嵌套中断请求的优先级相同, 则默认优先级最高的请求被首先受理。

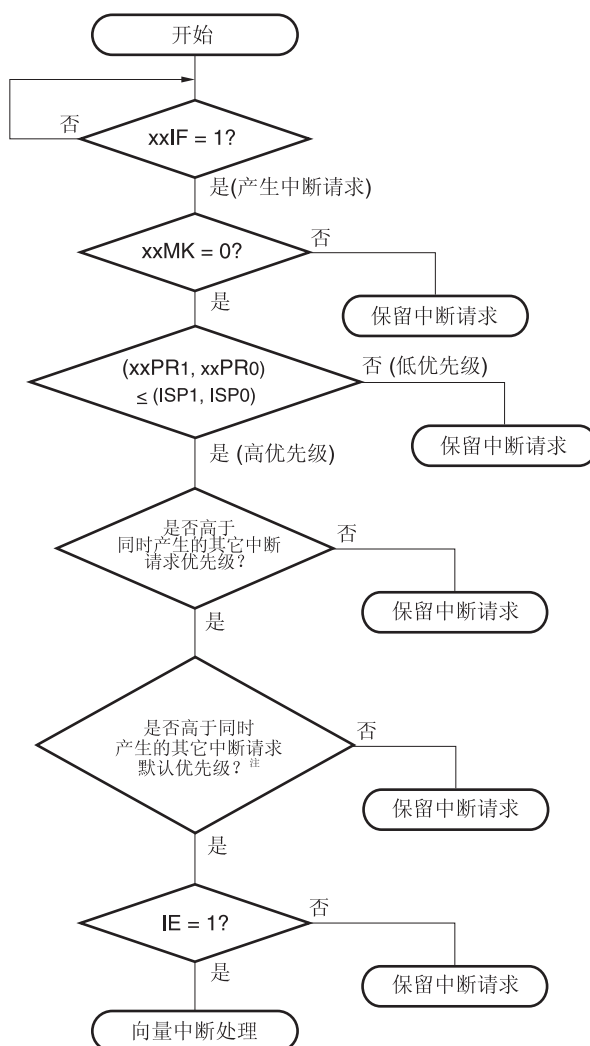
被保留的中断请求将在可以受理时被受理。

中断请求受理的流程如图 11-7 所示。

如果可屏蔽中断请求被受理, 则将按先 PSW 后 PC 的顺序, 其内容被保存至堆栈之中, IE 标志被复位为 0, 已受理中断对应的优先级指定标志的内容被传送至 ISP1 和 ISP0 标志。各中断请求确定的向量表数据被载入 PC 并分支。

可使用 RETI 指令从中断恢复。

图 11-7. 中断请求受理的处理流程



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: 用于控制可屏蔽中断请求的受理的标志(1 = 允许, 0 = 禁止)

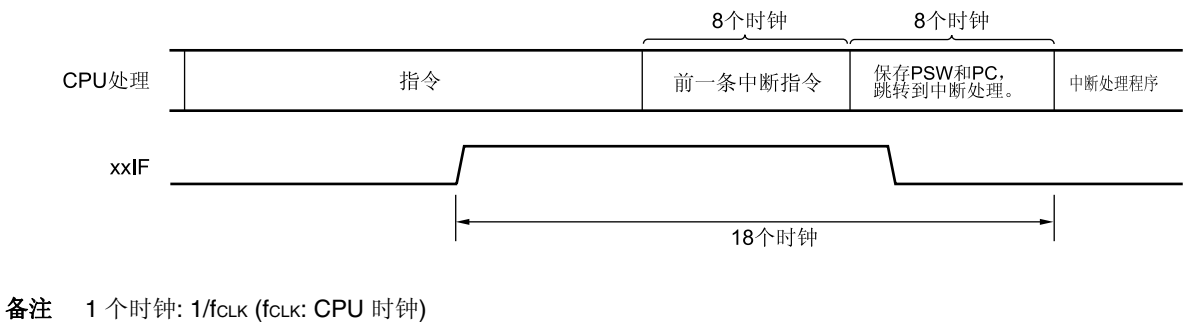
ISP0, ISP1: 表示正在处理的中断的优先级的标志(参阅图 11-6)

注 关于默认优先级, 请参阅表 11-1 中断源列表。

图 11-8. 中断请求受理时序 (最短时间)



图 11-9. 中断请求受理时序 (最长时间)



11.4.2 软件中断请求受理

通过执行 **BRK** 指令受理软件中断请求。不能禁止软件中断。

如果软件中断请求被受理，则将按先程序状态字 (**PSW**)后程序计数器(**PC**)的顺序，其内容被保存至堆栈中，**IE** 标志被复位为 0，向量表(0007EH, 0007FH)的内容被载入 **PC** 并分支。

可使用 **RETB** 指令从软件中断恢复。

注意事项 不能使用 **RETI** 指令从软件中断恢复。

11.4.3 嵌套中断处理

在执行一个中断的过程中受理了另一个中断请求时，发生嵌套中断处理。

除非选择中断请求受理允许状态(**IE** = 1)，否则不会发生嵌套中断处理。当某个中断请求被受理，中断请求受理被禁止 (**IE** = 0)。因此，若要允许嵌套中断处理，则须在中断处理期间用 **EI** 指令将 **IE** 标志设置为 1，以允许中断受理。

另外，即使允许中断，也可能不允许嵌套中断处理，这取决于中断优先级控制。优先级控制包括：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嵌套中断处理。

在中断允许状态下，如果产生的中断请求的优先级高于正在处理的中断的优先级，则受理嵌套中断处理。如果在中断处理期间产生的中断的优先级等于或低于正在处理的中断的优先级，则不受理嵌套中断处理。因中断处于禁止状态或者优先级较低而不处理的中断请求将被保留。在当前中断的处理结束后，被保留的中断请求将在执行至少一条主处理指令之后被受理。

表 11-4 展示了允许嵌套中断处理的中断请求之间的关系，图 11-10 为嵌套中断处理示例。

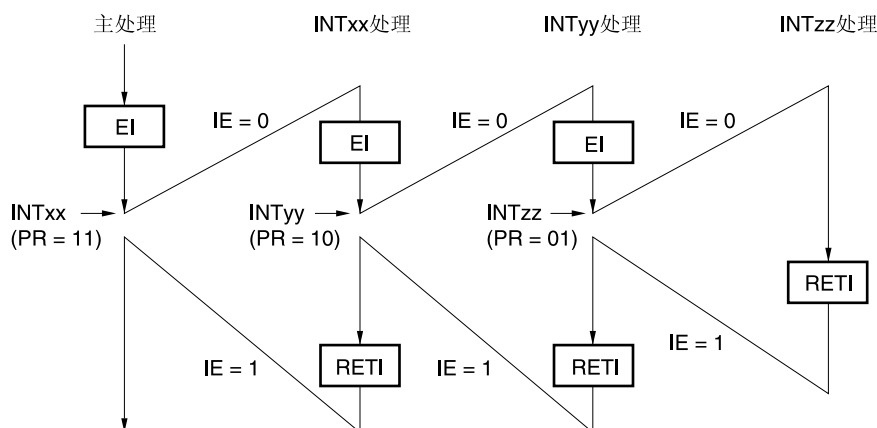
表 11-4. 中断处理期间允许嵌套中断处理的中断请求之间的关系

正在处理的中断 嵌套中断请求		可屏蔽中断请求								软件中断请求
		优先等级0 (PR = 00)		优先等级1 (PR = 01)		优先等级2 (PR = 10)		优先等级3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
可屏蔽中断	ISP1 = 0 ISP0 = 0	○	×	×	×	×	×	×	×	○
	ISP1 = 0 ISP0 = 1	○	×	○	×	×	×	×	×	○
	ISP1 = 1 ISP0 = 0	○	×	○	×	○	×	×	×	○
	ISP1 = 1 ISP0 = 1	○	×	○	×	○	×	○	×	○
软件中断		○	×	○	×	○	×	○	×	○

- 备注 1.** ○: 允许嵌套中断处理
- 2.** ×: 禁止嵌套中断处理
- 3.** ISP0、ISP1 和 IE 为包含在 PSW 中的标志。
 ISP1 = 0, ISP0 = 0: 正在处理等级 1 或等级 0 的中断。
 ISP1 = 0, ISP0 = 1: 正在处理等级 2 的中断。
 ISP1 = 1, ISP0 = 0: 正在处理等级 3 的中断。
 ISP1 = 1, ISP0 = 1: 等待中断受理。
 IE = 0: 禁止中断请求受理。
 IE = 1: 允许中断请求受理。
- 4.** PR 为包含在 PR00L, PR00H, PR10L, PR10H 寄存器中的标志。
 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: 通过 $\times\times PR1\times = 1, \times\times PR0\times = 0$ 指定等级 2
 PR = 11: 通过 $\times\times PR1\times = 1, \times\times PR0\times = 1$ 指定等级 3 (较低优先级)

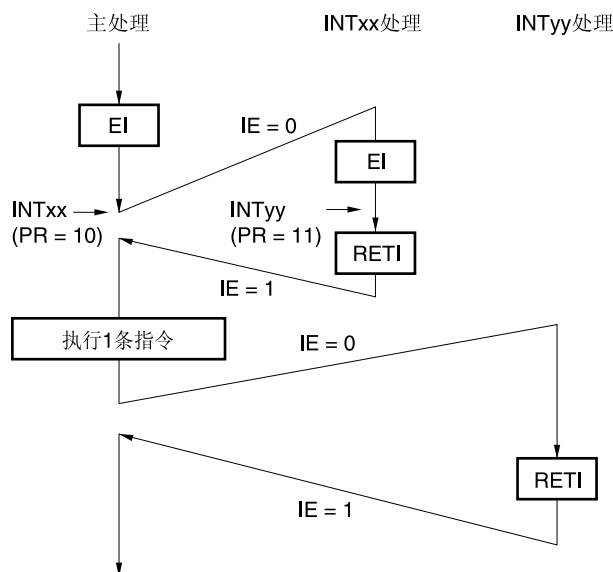
图 11-10. 嵌套中断处理示例 (1/2)

示例 1. 发生两次嵌套中断处理



在处理中断 INTxx 时，受理 INTyy 和 INTzz 这两个中断请求，发生嵌套中断处理。在受理各中断请求之前，必须执行 EI 指令以允许中断请求受理。

示例 2. 由于优先级控制，不发生嵌套中断处理。



在处理中断 INTxx 时产生的中断请求 INTyy 不被受理，因为其优先级低于 INTxx，不发生嵌套中断处理。INTyy 中断请求被保留，并在执行了主处理指令之后被受理。

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: 通过 $\times\times PR1\times = 1$, $\times\times PR0\times = 0$ 指定等级 2

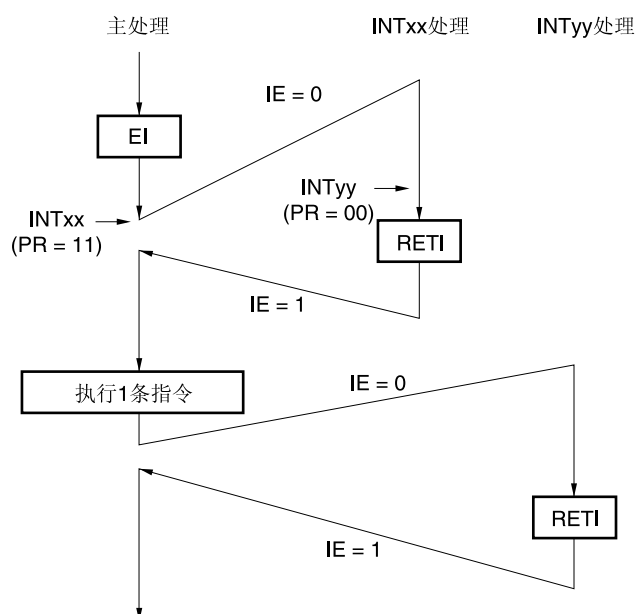
PR = 11: 通过 $\times\times PR1\times = 1$, $\times\times PR0\times = 1$ 指定等级 3 (较低优先级)

IE = 0: 禁止中断请求受理

IE = 1: 允许中断请求受理

图 11-10. 嵌套中断处理示例 (2/2)

示例 3. 因为不允许中断，所以不发生嵌套中断处理



在处理中断 INT_{xx} 时因为不允许中断（未使用 EI 指令），所以中断请求 INT_{yy} 不被受理，不发生嵌套中断处理。 INT_{yy} 中断请求被保留，并在执行了主处理指令之后被受理。

$PR = 00$: 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

$PR = 10$: 通过 $\times\times PR1\times = 1, \times\times PR0\times = 0$ 指定等级 2

$PR = 11$: 通过 $\times\times PR1\times = 1, \times\times PR0\times = 1$ 指定等级 3 (较低优先级)

$IE = 0$: 禁止中断请求受理

$IE = 1$: 允许中断请求受理

11.4.4 保留中断请求

有些指令，即使在执行指令期间产生了中断请求，中断请求受理也会被保留，直到下一条指令执行完成为止。这些指令（中断请求保留指令）如下所示。

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, \$addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- 对 IF0L、IF0H、MK0L、MK0H、PR00L、PR00H、PR10L 和 PR10H 的各个寄存器的写入指令

保留中断请求的时序如图 11-11 所示。

图 11-11. 保留中断请求



- 备注**
1. 指令 N: 中断请求保留指令
 2. 指令 M: 中断请求保留指令以外的指令

第十二章 按键中断功能

12.1 按键中断的功能

通过设置按键返回模式寄存器(KRM)并向按键中断输入引脚 (KR0 至 KR5) 输入上升/下降沿, 可以产生按键中断 (INTKR)。

表 12-1. 按键中断检测引脚的分配

按键中断引脚	按键返回模式寄存器 (KRM0)	按键返回标志寄存器 (KRF)
KR0	KRM00	KRF0
KR1	KRM01	KRF1
KR2	KRM02	KRF2
KR3	KRM03	KRF3
KR4	KRM04	KRF4
KR5	KRM05	KRF5

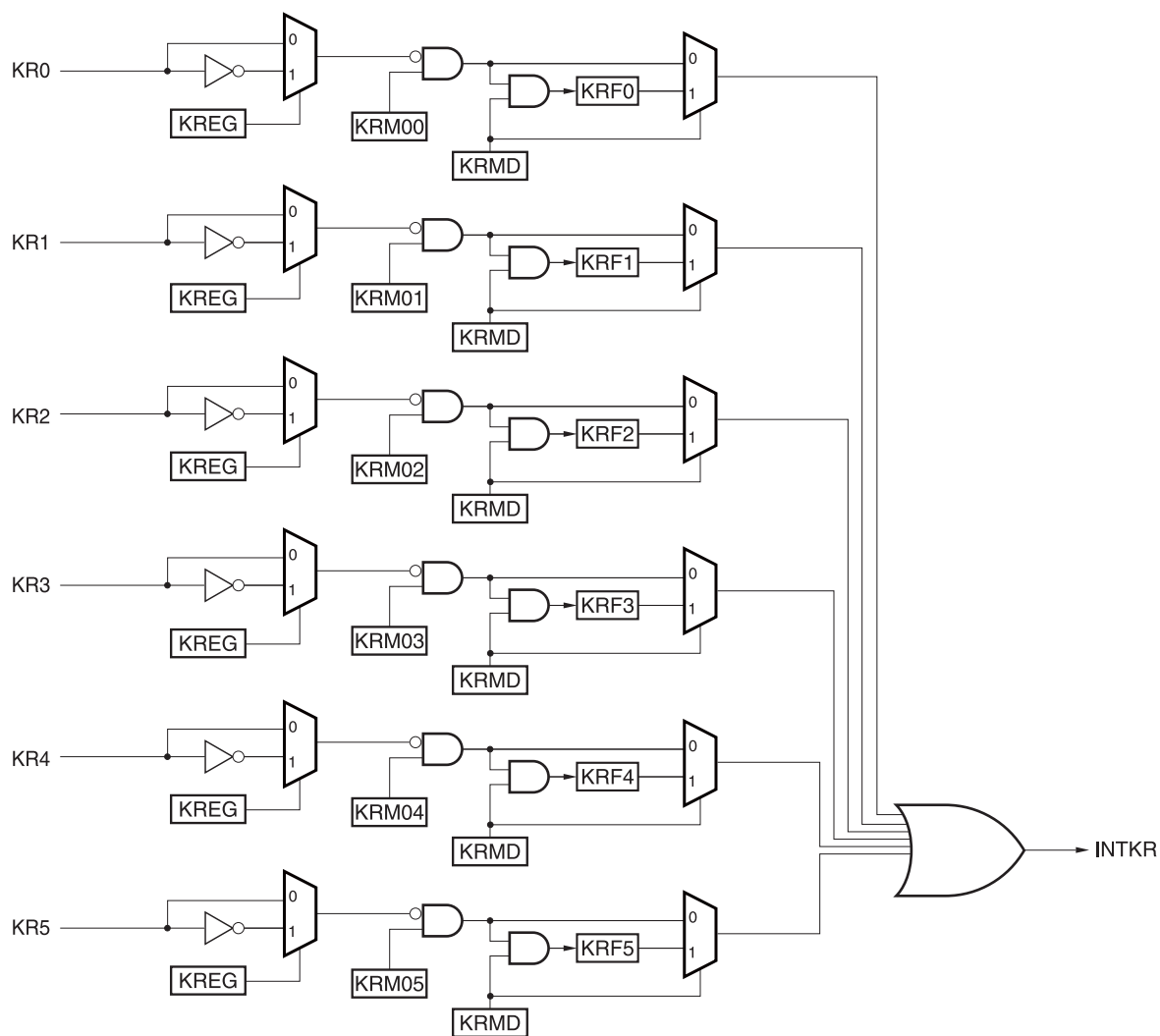
12.2 按键中断的配置

按键中断包括以下硬件。

表 12-2. 按键中断的配置

项目	配置
控制寄存器	按键返回控制寄存器(KRCTL) 按键返回模式寄存器(KRM0) 按键返回标志寄存器 (KRF) 端口模式寄存器 0, 4 (PM0, PM4) 端口寄存器 0, 4 和 12 (P0, P4, P12)

图 12-1. 按键中断的框图



12.3 控制按键中断的寄存器

按键中断功能由以下五个寄存器控制：

- 按键返回控制寄存器(KRCTL)
- 按键返回模式寄存器(KRMO)
- 按键返回标志寄存器 (KRF)
- 端口模式寄存器 0, 4 (PM0, PM4)
- 端口寄存器 0, 4 和 12 (P0, P4, P12)

12.3.1 按键返回控制寄存器(KRCTL)

该寄存器控制按键返回标志 (KRF0 至 KRF5) 的使用和设置检测沿。

使用 1 位或 8 位存储器操作指令设置 KRCTL 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 12-2. 按键返回控制寄存器 (KRCTL) 的格式

地址：FFF34H 复位后：00H R/W

符号	7	6	5	4	3	2	1	0
KRCTL	KRMD	0	0	0	0	0	0	KREG

KRMD	按键返回标志 (KRF0 至 KRF5) 的使用						
0	不使用按键返回标志						
1	使用按键返回标志						

KREG	检测沿的选择 (KR0 至 KR5)						
0	下降沿						
1	上升沿						

KRMD	KREG	中断功能
0	0	按键中断、外部中断（根据端口电平确定） ^注
0	1	外部中断（根据端口电平确定）
1	0	外部中断（根据标志确定）
1	1	

注 检测到下降沿时，外部中断功能和按键中断功能的功能及操作相同。

12.3.2 按键返回模式寄存器(KRM0)

该寄存器用于设置按键中断模式。

使用 1 位或 8 位存储器操作指令设置 KRM0 寄存器。

产生复位信号后，这些寄存器被清除为 00H。

图 12-3. 按键返回模式寄存器 (KRM0)的格式

地址: FFF37H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
KRM0	0	0	KRM05	KRM04	KRM03	KRM02	KRM01	KRM00

KRM0n	按键中断模式的控制 (n = 0 至 5)
0	不检测按键中断信号
1	检测按键中断信号

- 注意事项**
1. 选择下降沿(KRMD = 0)检测到按键中断信号时(KRM0n = 1)，请通过外部电阻将对应的输入引脚上拉至 V_{DD}。对于 KR1、KR6 至 KR9 引脚，还可以通过将按键中断输入引脚 PU01 至 PU04、PU40 和 PU125 (上拉电阻选项寄存器 0, 4, 12(PU0, PU4 和 PU12) 的对应位设置为 1 来使用内部上拉电阻。
 2. KREG 被设置为 0 时向按键中断输入引脚输入低电平，或者 KREG 被设置为 1 时向按键中断输入引脚输入高电平的状态下，如果 KRM0 寄存器的目标位被置位，则产生中断。若要忽略该中断，则须在通过中断屏蔽标志禁止中断处理之后设置 KRM0 寄存器。然后清除中断请求标志，并允许中断处理。
 2. 按键中断模式下未使用的位可以用作通常端口。

12.3.3 按键返回标志寄存器 (KRF)

该寄存器控制按键返回标志 (KRF0 至 KRF5)。

使用 1 位或 8 位存储器操作指令设置 KRF 寄存器。

产生复位信号后，该寄存器被清除为 00H。

图 12-4. 键返回标志寄存器 (KRF)的格式

地址: FFF35H 复位后: 00H R/W ^注

符号	7	6	5	4	3	2	1	0
KRF	0	0	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0

KRFn	按键中断标志
0	没有检测到按键中断标志。
1	检测到按键中断标志。

注 写入 1 无效。要清除 KRFn，须使用 8 位存储器操作指令向目标位写 0，向其他位写 1。

注意事项 不使用按键中断标志 (KRMD = 0)时，禁止存取 KRF 寄存器。

12.3.4 端口模式寄存器 0, 4 (PM0, PM4)

该寄存器以 0 位为单位设置端口 1 和 4 的输入和输出。
使用 1 位或 8 位存储器操作指令设置 PM0 和 PM4 寄存器。
产生复位信号后，该寄存器被设置为 FFH。

图 12-5. 端口模式寄存器 0, 4 (PM0, PM4)的格式

地址: FFF20H 复位后: FFH R/W

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

PM0n	P0n/KRm 引脚的输入/输出模式选择 (n = 1 至 4, m = 2 至 5)
0	输出模式 (输出缓冲器启用)
1	输入模式 (输出缓冲器关闭)

地址: FFF24H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM4	1	1	1	1	1	1	1	PM40

PM4n	P40/KR0 引脚输入/输出模式的选择
0	输出模式 (输出缓冲器启用)
1	输入模式 (输出缓冲器关闭)

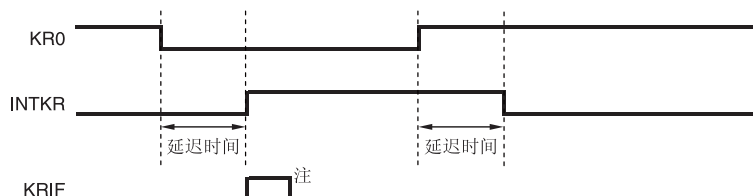
12.4 按键中断功能的操作

12.4.1 不使用按键中断标志时 (KRMD = 0)

通过将 KREG 位设置的有效沿输入至按键中断输入引脚 (KR0 至 KR5)，从而产生按键中断(INTKR)。产生按键中断(INTKR)后，通过读取端口寄存器来确认输入电平，从而可以确定输入了有效沿的通道。

INTKR 信号根据按键中断输入引脚 (KR0 至 KR5) 的输入电平而变化。

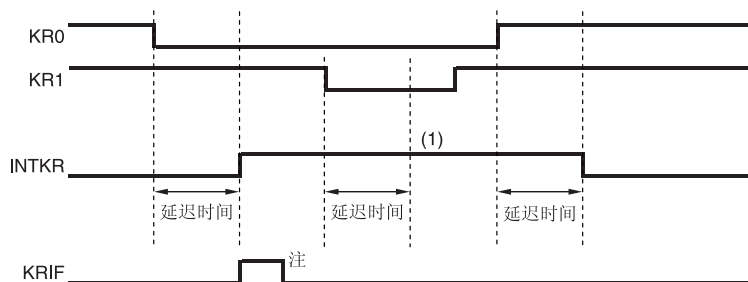
图 12-6. 向一个通道输入按键中断时 INTKR 信号的操作 (设置为 KRMD = 0, KREG = 0 时)



注 受理向量中断请求或者通过软件执行清除。

向多个按键中断输入引脚输入有效沿时的操作如图 12-7 所示。向一个引脚输入低电平 (设置为 KREG = 0 时) 的期间，由于 INTKR 信号被置位，在此期间内，即使向其他引脚输入下降沿也不会再次产生按键中断(INTKR) (图中的 (1))。

图 12-7. 向多个通道输入按键中断时 INTKR 信号的操作 (设置为 KRMD = 0, KREG = 0 时)



注 受理向量中断请求或者通过软件执行清除。

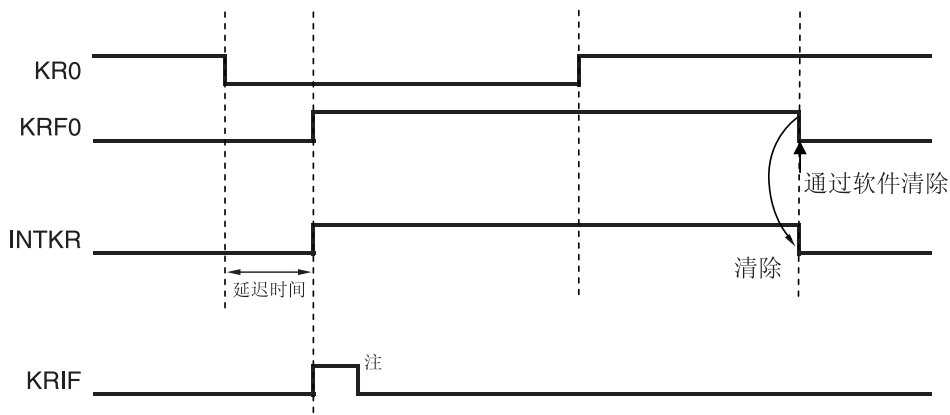
12.4.2 使用按键中断标志时 (KRMD = 1)

通过将 KREG 位设置的有效沿输入至按键中断输入引脚 (KR0 至 KR5)，从而产生按键中断(INTKR)。产生按键中断(INTKR)后，通过读取按键返回标志寄存器 (KFR) 来确认输入电平，从而可以确定输入了有效沿的通道。

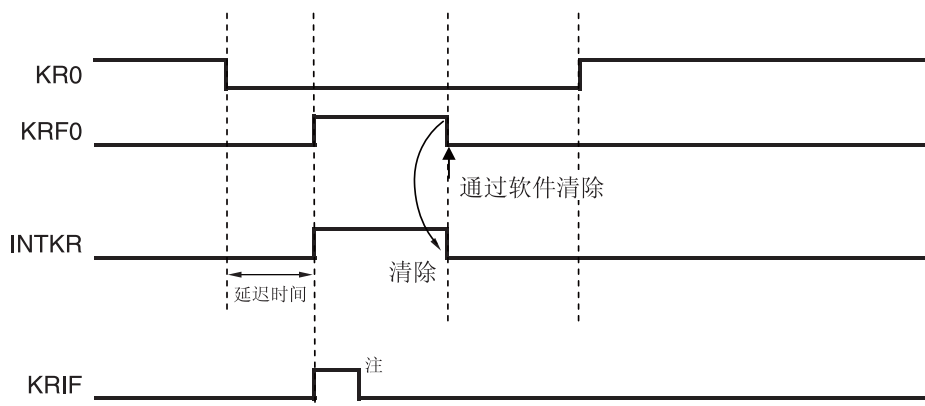
设置为 KRMD = 1 时，通过清除 KFR 寄存器的对应位可以清除 INTKR 信号。如图 12-8 所示，针对一个通道的 1 个下降沿 (设置为 KREG = 0 时)，不论 KRFn 位的清除时序是在上升沿之前还是之后，仅产生 1 次中断。

图 12-8. 使用按键中断标志时 INTKR 信号的基本操作 (设置为 KRMD = 1, KREG = 0 时)

(a) 向 KR0 引脚输入上升沿之后清除 KRF0



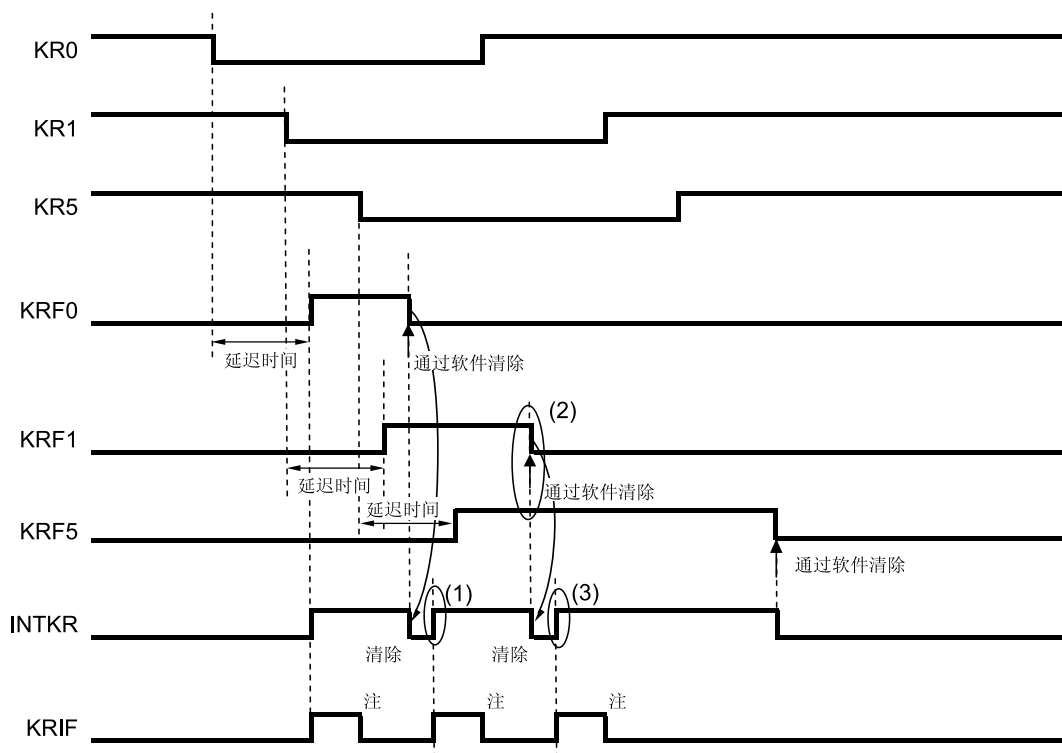
(b) 向 KR0 引脚输入上升沿之前清除 KRF0



注 受理向量中断请求或者通过软件执行清除。

向多个按键中断输入引脚输入有效沿时的操作如图 12-9 所示。向 KR0 引脚输入下降沿之后，也向 KR0 和 KR5 引脚输入下降沿（设置为 KREG = 0 时）。清除 KRF0 位时，由于 KRF1 位被置位，将在清除 KRF0 位的 1 个时钟(f_{CLK})后产生按键中断(INTKR)（图中的(1)）。向 KR5 引脚输入下降沿，在清除 KRF1 位时，由于 KR5 引脚被置位（图中的(2)），将在清除 KRF1 位的 1 个时钟(f_{CLK})后产生按键中断(INTKR)（图中的(3)）。如此所示，可以向多个通道输入有效沿产生按键中断(INTKR)。

图 12-9. 向多个通道输入按键中断时 INTKR 信号的操作（设置为 KRMD = 1, KREG = 0 时）

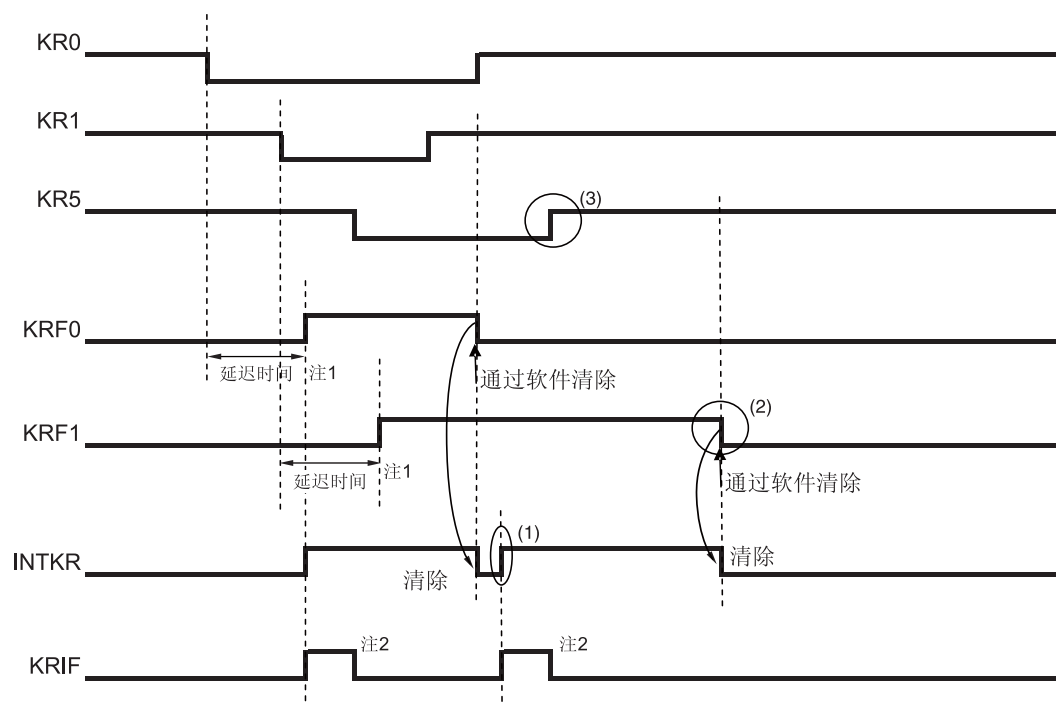


注 受理向量中断请求或者通过软件执行清除。

备注 f_{CLK}: CPU/外围硬件时钟频率

向 KR5 引脚输入有效沿时不产生按键中断(INTKR) 的操作如图 12-10 所示。向 KR0 引脚输入下降沿之后，向 KR1 和 KR5 引脚也输入下降沿（设置为 $KREG = 0$ 时）。清除 KRF0 位时，虽然 KR1 引脚为高电平，由于 KRF1 位被置位，将在清除 KRF0 位的 1 个时钟(f_{CLK})后产生按键中断(INTKR)（图中的(1)）。在清除 KRF1 位（图中的(2)）之前，由于 KR5 引脚为高电平（图中的(3)），KR5 引脚不产生按键中断(INTKR)。

图 12-10. 向 KR5 输入有效沿时不产生 INTKR 信号的情况（设置为 $KRMD = 1$, $KREG = 0$ 时）



注 1. 延迟时间最长可达到按键中断输入高电平宽度或低电平宽度的最大值（请参阅 21.4 AC 特性）。

2. 受理向量中断请求或者通过软件执行清除。

备注 f_{CLK} : CPU/外围硬件时钟频率

第十三章 待机功能

13.1 概要

待机功能可以降低系统的操作电流，有以下三种模式可供选择。

(1) HALT 模式

通过执行 HALT 指令设置为 HALT 模式。在 HALT 模式下，CPU 工作时钟停止。如果在设置 HALT 模式之前高速片上振荡器正在工作，则时钟的振荡将继续。在该模式下，操作电流降低幅度不如 STOP 模式，但 HALT 模式在产生中断请求时立即重启操作以及经常执行间歇操作方面非常有效。

(2) STOP 模式

通过执行 STOP 指令设置为 STOP 模式。在 STOP 模式下，高速片上振荡器停止工作，并停止整个系统，从而大幅降低 CPU 操作电流。

由于可以通过中断请求解除该模式，因而可以执行间歇操作。

在以上任何一种模式下，寄存器、标志和数据存储器全部保持设置待机模式之前的内容。同时还保持输入/输出端口输出锁存器和输出缓冲器状态。

- 注意事项**
1. 使用待机功能时，建议通过以下步骤降低 A/D 转换器的功耗：首先将 A/D 转换器模式寄存器 0 (ADM0) 的位 7 (ADCS)和位 0 (ADCE)清除为 0，以停止 A/D 转换操作，然后执行 STOP 指令。
 2. 可通过选项字节选择低速片上振荡器在 HALT 或 STOP 模式下，是继续振荡还是停止振荡。详情请参阅第十六章 选项字节。

13.2 待机功能工作

13.2.1 HALT 模式

(1) HALT 模式

通过执行 HALT 指令来设置 HALT 模式。

HALT 模式时的工作状态如下所示。

注意事项 由于中断请求信号用于解除 HALT 模式，因此，如果有一个中断请求标志被置 1 且中断屏蔽标志被清 0 的中断源，则不能切换到 HALT 模式。

表 13-1. HALT 模式时的工作状态

HALT 模式设置			CPU 以主系统时钟工作期间，执行 HALT 指令时
项目			CPU 以高速片上振荡器时钟(f _{IH})工作时
系统时钟			停止向 CPU 供应时钟。
高速片上振荡器时钟	f _{IH}		继续工作(不可以停止)
低速片上振荡器时钟	f _{IL}		通过选项字节(000C0H)的位 0 (WDSTBYON)和位 4 (WDTON)进行设置。 ● WDTON = 0: 停止 ● WDTON = 1 且 WDSTBYON = 1: 振荡 ● WDTON = 1 且 WDSTBYON = 0: 停止
CPU			停止工作
代码闪存			停止工作
RAM			
端口(锁存器)			保持设置为 HALT 模式前的状态
定时器阵列单元			可以工作
看门狗定时器			由选项字节 (000C0H)的位 0 (WDSTBYON)设置 ● WDSTBYON = 0: 停止工作 ● WDSTBYON = 1: 继续工作
时钟输出/蜂鸣器输出			可以工作
A/D 转换器			
串行阵列单元(SAU)			
可选择的上电复位功能			
外部中断			
按键中断功能			

备注 停止工作：在切换到 HALT 模式时自动停止工作。

禁止工作：在切换到 HALT 模式前使工作停止。

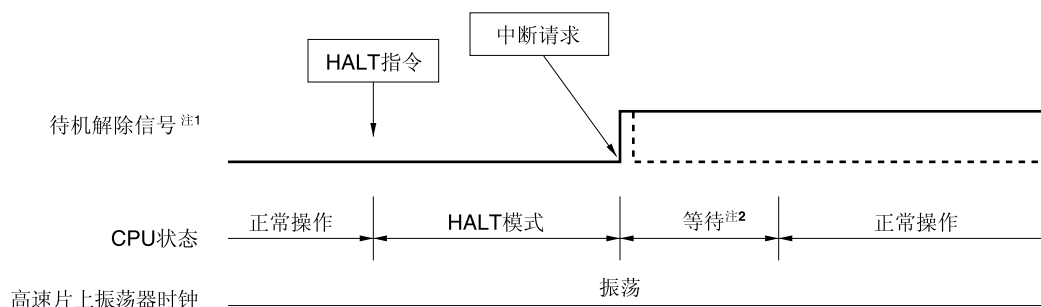
(2) HALT 模式的解除

可使用以下两种方法解除 HALT 模式。

(a) 通过未屏蔽中断请求解除

当未屏蔽中断产生中断请求时，HALT 模式被解除。如果允许受理中断，则执行向量中断处理。如果不允许受理中断，则执行下一个地址的指令。

图 13-1. 通过产生中断请求解除 HALT 模式



注 1. 关于待机解除信号的详情，请参阅图 11-1. 中断功能的基本配置。

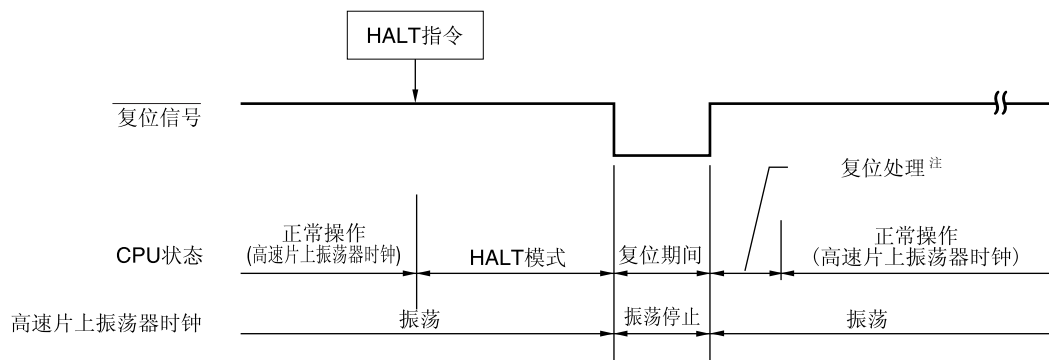
2. 解除 HALT 模式需要等待的时间：
- | | |
|-------------|-------------|
| 执行向量中断处理时： | 28 至 29 个时钟 |
| 不执行向量中断处理时： | 20 至 21 个时钟 |

备注 虚线表示导致待机模式解除的中断请求被受理时的情况。

(b) 通过产生复位信号解除 HALT 模式

当产生复位信号时，HALT 模式被解除，然后，和正常复位操作一样，跳转至复位向量地址之后执行程序。

图 13-2. 通过产生复位信号解除 HALT 模式



注 复位处理时间请参阅第十四章 复位功能。此外，SPOR 电路引起的复位处理时间请参阅第十五章 可选择上电复位电路。

13.2.2 STOP 模式

(1) STOP 模式的设置和工作状态

通过执行 STOP 指令来设置 STOP 模式。

注意事项 由于中断请求信号用于解除 STOP 模式，因此，如果有一个中断请求标志被置 1 且中断屏蔽标志被清 0 的中断源，则进入 STOP 模式后，在 f_{IH} 的振荡稳定时间后 STOP 模式将被解除。

STOP 模式时的工作状态如下所示。

表 13-2. STOP 模式时的工作状态

STOP 模式设置			CPU 工作期间，执行 STOP 指令时
项目			CPU 以高速片上振荡器时钟(f_{IH})工作时
系统时钟			停止向 CPU 供应时钟。
高速片上振荡器时钟	f_{IH}		停止
低速片上振荡器时钟	f_{IL}		由选项字节(000C0H)的位 0 (WDSTBYON)和位 4 (WDTON)设置 • WDTON = 0: 停止 • WDTON = 1 且 WDSTBYON = 1: 振荡 • WDTON = 1 且 WDSTBYON = 0: 停止
CPU			停止工作
代码闪存			
RAM			停止工作
端口(锁存器)			保持设置为 STOP 模式前的状态
定时器阵列单元			禁止工作
看门狗定时器			由选项字节 (000C0H)的位 0 (WDSTBYON)设置 • WDSTBYON = 0: 停止工作 • WDSTBYON = 1: 继续工作
时钟输出/蜂鸣器输出			禁止工作
A/D 转换器			
串行阵列单元(SAU)			
可选择的上电复位功能			可以工作
外部中断			
按键中断功能			

备注 停止工作：在切换到 STOP 模式时自动停止工作。

禁止工作：在切换到 STOP 模式前使工作停止。

注意事项 1. 对于工作于在 STOP 模式下停止振荡的时钟的外围硬件，在解除 STOP 模式后，必须将其重启才能使用该外围硬件。

2. 若要使低速片上振荡器时钟停止于 STOP 模式，必须预先设置选项字节使看门狗定时器操作停止于 HALT/STOP 模式(000C0H 的位 0 (WDSTBYON) = 0)。

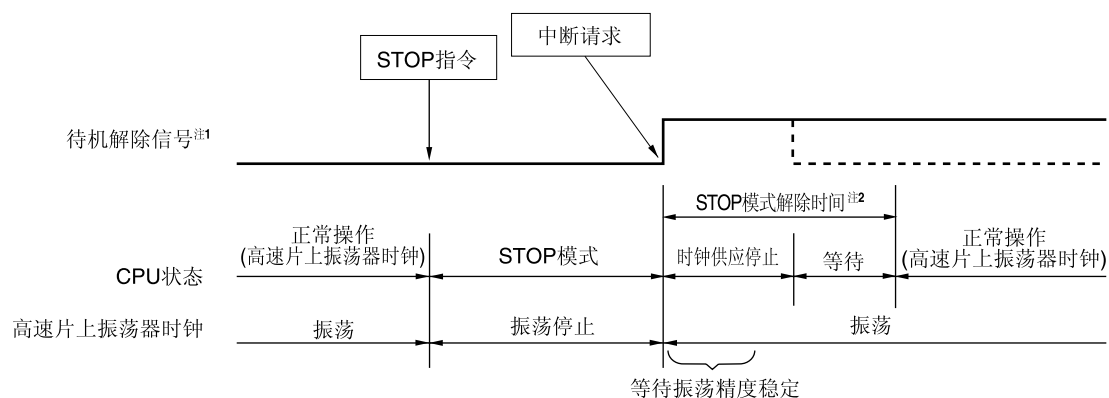
(2) STOP 模式的解除

可使用以下两种方法解除 STOP 模式。

(a) 通过未屏蔽中断请求解除 STOP 模式

当未屏蔽中断产生中断请求时，STOP 模式被解除。经过振荡稳定时间之后，如果允许受理中断，则执行向量中断处理。如果不允许受理中断，则执行下一个地址的指令。

图 13-3. 通过产生中断请求解除 STOP 模式



注 1. 关于待机解除信号的详情，请参阅图 11-1. 中断功能的基本配置。

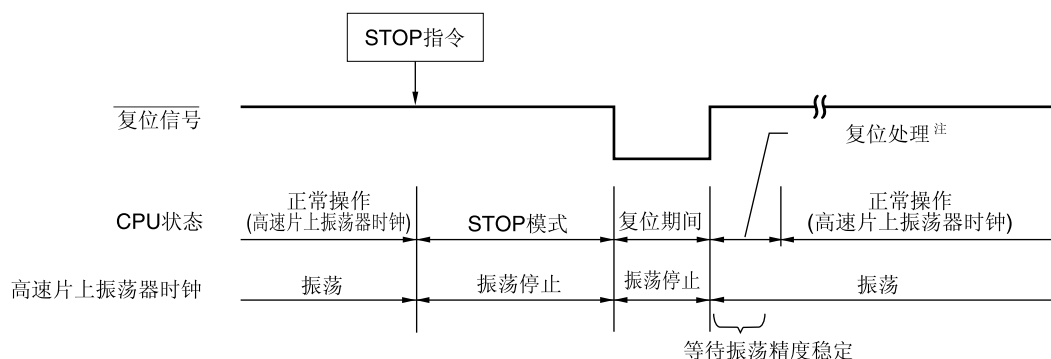
2. 解除 STOP 模式需要等待的时间：
- | | |
|---------------|------------------|
| 停止供应时钟： | 27 μ s(TYP.) |
| 等待 | |
| · 执行向量中断处理时： | 11 个时钟 |
| · 不执行向量中断处理时： | 3 个时钟 |

备注 1. 停止供应时钟的时间随着温度条件及是否处于 STOP 模式而变化。

2. 虚线表示导致待机模式解除的中断请求被受理时的情况。

(b) 通过产生复位信号解除 STOP 模式

当产生复位信号时，STOP 模式被解除，然后，和正常复位操作一样，跳转至复位向量地址之后执行程序。

图 13-4. 通过产生复位信号解除 STOP 模式

注 复位处理时间请参阅**第十四章 复位功能**。此外，SPOR 电路引起的复位处理时间请参阅**第十五章 可选择上电复位电路**。

第十四章 复位功能

以下四种操作方法能够产生复位信号。

- (1) 经由 $\overline{\text{RESET}}$ 引脚输入外部复位
- (2) 通过看门狗定时器的程序失控检测的内部复位
- (3) 通过比较可选择上电复位(SPOR)电路的电源电压与检测电压的内部复位
- (4) 通过执行非法指令的内部复位^{注1}
- (5) 通过数据保持电压的复位^{注2}

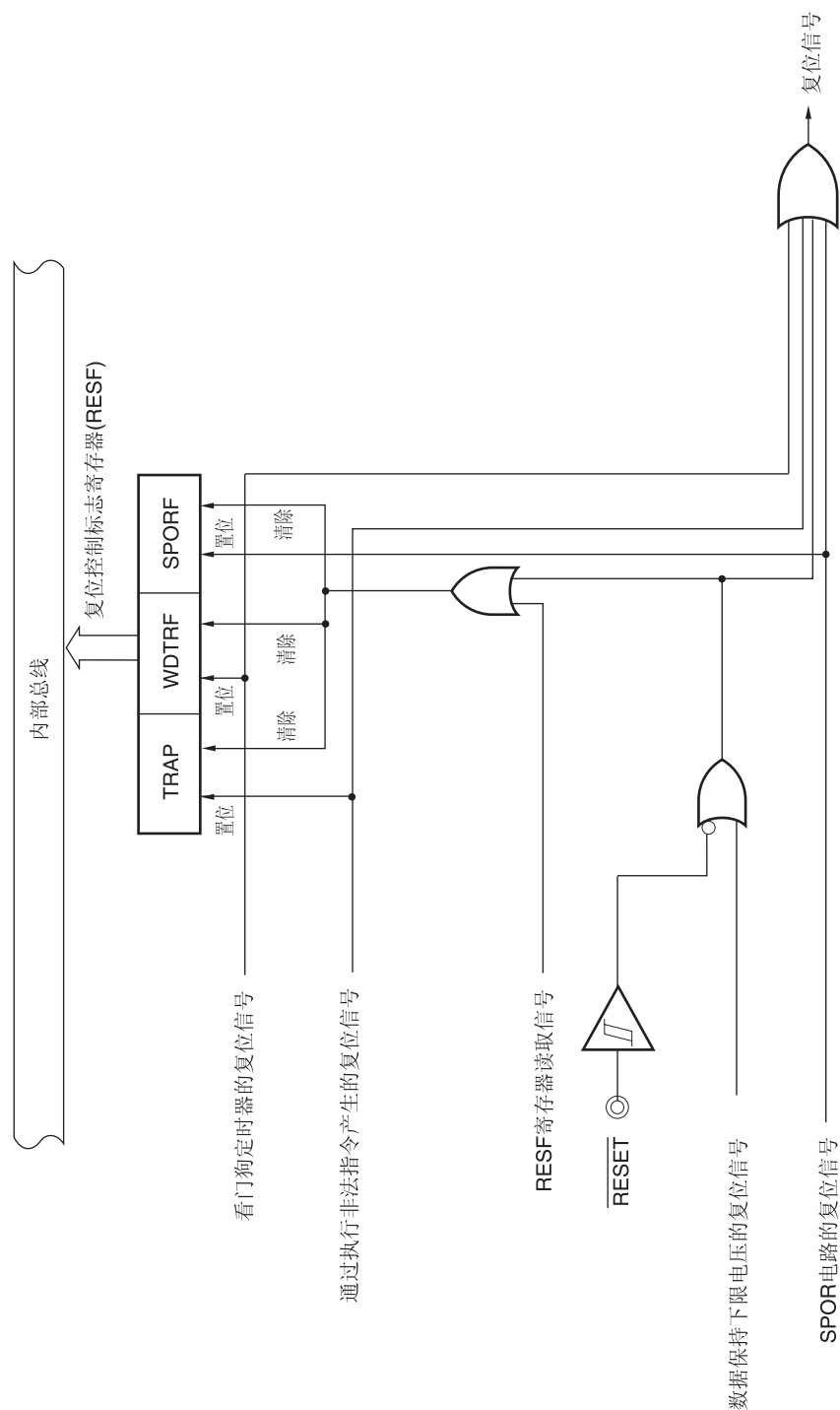
产生复位信号后，内部复位和外部复位都以地址 0000H 和 0001H 处的内容，作为程序的起始地址，开始执行程序。

注 1. 执行指令代码 FFH 时，产生非法指令。
通过片上调试仿真器进行仿真时，不会因执行非法指令发生内部复位。

2. $V_{DD} \geq$ 数据保持电压时： 数据不被复位
 $V_{DD} <$ 数据保持电压时： 数据被复位
数据被复位的最大电压为数据保持电压特性。

- 注意事项**
1. 将用户选项字节(000C1H)的 PORTSELB 位设置为 1 以使 P125 引脚作为 $\overline{\text{RESET}}$ 工作，向 $\overline{\text{RESET}}$ 引脚输入一个 10 μs 或以上的低电平，以执行外部复位。
若要在加电时执行外部复位，必须向 $\overline{\text{RESET}}$ 引脚输入低电平后打开电源，在电源电压处于 21.4 AC 特性所示工作电压范围内的状态下保持至少 10 μs 的低电平，然后输入高电平。
 2. 产生复位信号时，高速片上振荡器时钟和低速片上振荡器时钟停止振荡。
 3. 复位后，各个 SFR 和第二 SFR 被初始化，端口引脚的状态如下所示：
 - P40：产生外部复位或数据保持电源电压复位的期间成为高阻抗。产生其他复位的期间以及受理复位后成为高电平（连接内部上拉电阻）。
 - P125：产生外部复位的期间成为低电平（向 $\overline{\text{RESET}}$ 引脚输入低电平）。产生其他复位的期间以及受理复位后成为高电平（连接内部上拉电阻）。
 - P40、P125 以外的端口：产生复位的期间以及受理复位后成为高阻抗。

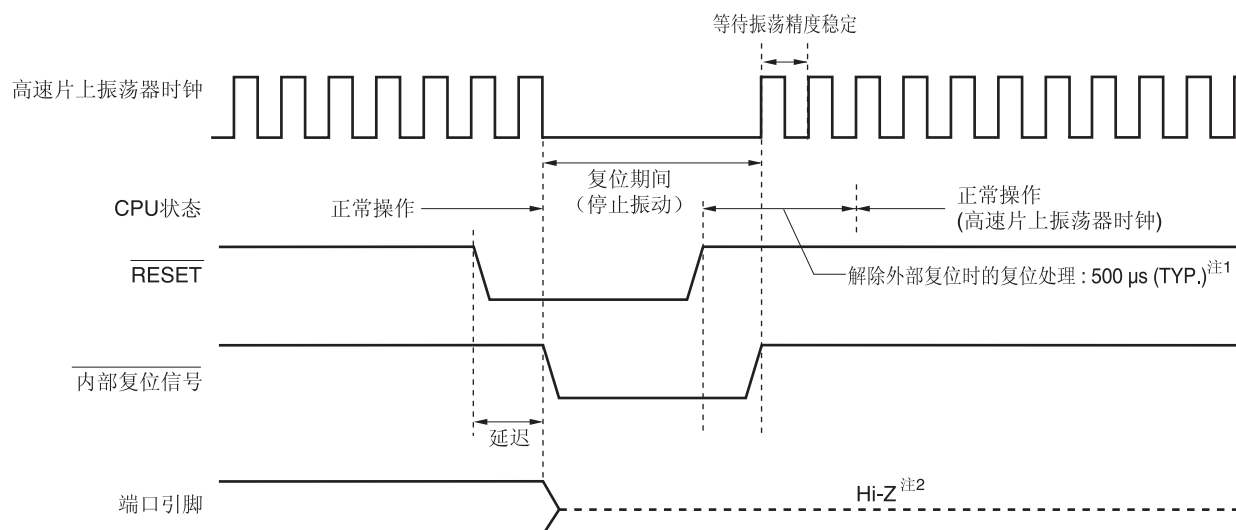
图 14-1. 复位功能框图



14.1 复位操作的时序

当向 $\overline{\text{RESET}}$ 引脚输入低电平时 LSI 复位，当向 $\overline{\text{RESET}}$ 引脚输入高电平时复位状态解除。处理复位后，开始以高速片上振荡器时钟作为工作时钟执行程序。

图 14-2. 通过 $\overline{\text{RESET}}$ 输入的复位时序



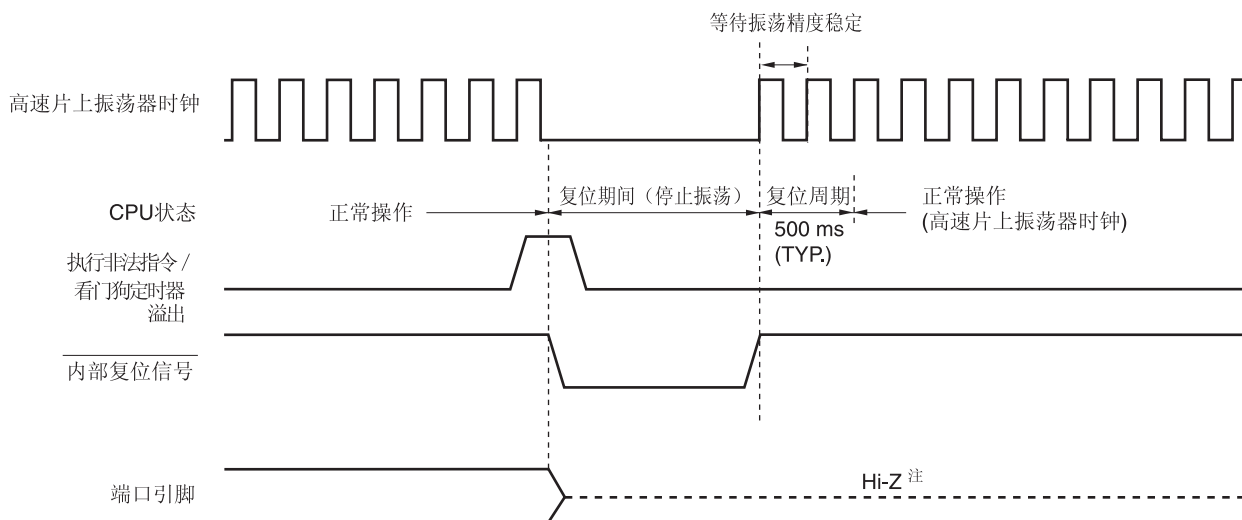
注1. 打开电源时，在解除外部复位时需要的复位处理时间之前，还需要经过 SPOR 复位处理时间 (MAX. 3.39ms)。

2. 端口引脚(P40)的状态如下所示：

- 外部复位或者数据保持电源电压引起的复位期间成为高阻抗
- 受理复位后成为高电平（连接内部上拉电阻）

在通过看门狗定时器溢出或执行非法指令引起复位的情况下，自动解除复位状态。处理复位后，开始以高速片上振荡器时钟作为工作时钟执行程序。

图 14-3. 通过看门狗定时器溢出或执行非法指令的复位时序



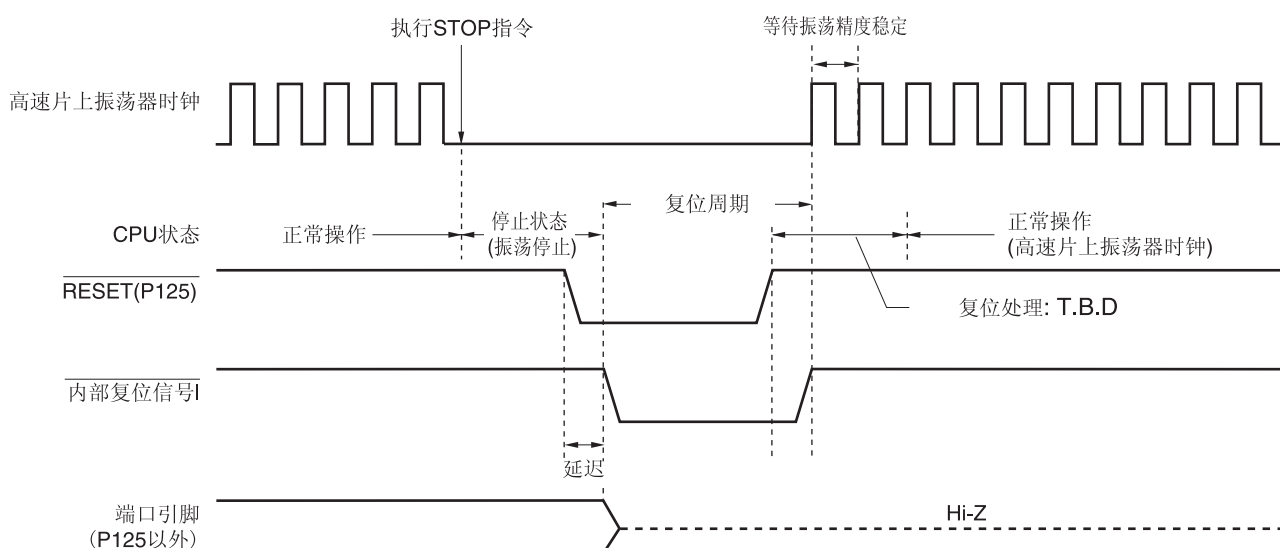
注 端口引脚(P40, P125)的状态如下所示:

- 复位期间和受理复位后成为高电平（连接内部上拉电阻）

注意事项 看门狗定时器产生内部复位时，看门狗定时器也会被复位。

备注 关于可选择上电复位(SPOR)电路的检测电压引起的复位时序，请参阅第十五章 可选择上电复位电路。

图 14-4. STOP 模式时通过输入 RESET 的复位时序



14.2 复位期间的工作状态

复位期间的工作状态如表 14-1 所示。受理复位后硬件的状态如表 14-2 所示。

表 14-1. 复位期间的工作状态

项目			复位期间
系统时钟			停止向 CPU 供应时钟。
高速片上振荡器时钟	f _{IH}		停止工作
低速片上振荡器时钟	f _{IL}		
CPU			停止工作
代码闪存			停止工作
RAM			停止工作
端口(锁存器)			高阻抗 ^注
定时器阵列单元			停止工作
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)			
可选择的上电复位功能			可进行检测操作
外部中断			停止工作
按键中断功能			

- 注 P40 和 P125 的引脚状态如下所示
- P40: 产生外部复位或数据保持电源电压复位的期间成为高阻抗。产生其他复位的期间以及受理复位后成为高电平（连接内部上拉电阻）。
 - P125: 产生外部复位的期间成为低电平（向 $\overline{\text{RESET}}$ 引脚输入低电平）。产生其他复位的期间以及受理复位后成为高电平（连接内部上拉电阻）。

表 14-2. 受理复位后各硬件的状态 (1/3)

硬件		受理复位后 ^{注1}
程序计数器(PC)		复位向量表(0000H, 0001H)的内容被设置。
堆栈指针(SP)		不定
程序状态字(PSW)		06H
RAM	数据存储器	不定
	通用寄存器	不定
端口寄存器(P0, P4, P12, P13) (输出锁存器)		P0: 00H, P4: 01H, P12, P13: 不定
端口模式寄存器(PM0, PM4)		FFH
端口模式控制寄存器(PMC0)		FFH
端口输出模式寄存器(POM0)		00H
上拉电阻选择寄存器(PU0, PU4, PU12)		PU0: 00H, PU4: 01H, PU12: 20H
外围输入/输出重定向寄存器(PIOR)		00H
噪声滤波器允许寄存器 0, 1 (NFEN0, NFEN1)		00H
外围允许寄存器 0 (PER0)		00H
高速片上振荡器频率选择寄存器(HOCODIV)		不定
定时器阵列单元	定时器数据寄存器 00H/L (TDR00H, TDR00L)	00H
	定时器数据寄存器 01H/L (TDR01H, TDR01)	00H
	定时器模式寄存器 00H/L (TMR00H, TMR00L)	00H
	定时器模式寄存器 01H/L (TMR01H, TMR01L)	00H
	定时器状态寄存器 00 (TSR00)	00H
	定时器状态寄存器 01 (TSR01)	00H
	定时器计数器寄存器 00H/L (TCR00H, TCR00L)	FFH
	定时器计数器寄存器 01H/L (TCR01H, TCR01L)	FFH
	定时器通道允许状态寄存器 0 (TE0, TEH0)	00H
	定时器通道开始寄存器 0 (TS0, TSH0)	00H
	定时器通道停止寄存器 0 (TT0, TTH0)	00H
	定时器时钟选择寄存器 0 (TPS0)	00H
	定时器输出寄存器 0 (TO0)	00H
	定时器输出允许寄存器 0 (TOE0)	00H
	定时器输出电平寄存器 0 (TOL0)	00H
	定时器输出模式寄存器 0 (TOM0)	00H
时钟输出/蜂鸣器输出	时钟输出选择寄存器 0 (CKS0)	00H
看门狗定时器	允许寄存器(WDTE)	1AH/9AH ^{注2}

注 1. 产生复位信号时以及等待振荡稳定期间时, 各硬件状态中仅有 PC 的内容为不定。其他的硬件状态在复位后也保持不变。

2. WDTE 的复位值取决于选项字节(WDTON 位)的设置。

表 14-2. 受理复位后各硬件的状态 (2/3)

硬件		受理复位后 ^注
A/D 转换器	A/D 转换结果低位存储寄存器(ADCRL)	00H
	A/D 转换结果高位存储寄存器(ADCRH)	00H
	A/D 转换器模式寄存器 0, 2 (ADM0, ADM2)	00H
	模拟输入通道选择寄存器(ADS)	00H
串行阵列单元(SAU)	串行数据寄存器 00H/L (SDR00H, SDR00L)	00H
	串行数据寄存器 01H/L (SDR01H, SDR01L)	00H
	串行状态寄存器 00, 01 (SSR00, SSR01)	00H
	串行标志清除触发寄存器 00, 01 (SIR00, SIR01)	00H
	串行模式寄存器 00H, 01H (SMR00H, SMR01H)	00H
	串行模式寄存器 00L, 01L (SMR00L, SMR01L)	20H
	串行通信操作设置寄存器 00H, 01H (SCR00H, SCR01H)	00H
	串行通信操作设置寄存器 00L, 01L (SCR00L, SCR01L)	87H
	串行通道允许状态寄存器 0 (SE0)	00H
	串行通道开始寄存器 0 (SS0)	00H
	串行通道停止寄存器 0 (ST0)	00H
	串行时钟选择寄存器 0 (SPS0)	00H
	串行输出寄存器 0 (SO0)	03H
	串行时钟输出寄存器 0 (CKO0)	03H
	串行输出允许寄存器 0 (SOE0)	00H
	串行输出电平寄存器 0 (SOLO)	00H
按键中断	按键返回控制寄存器(KRCTL)	00H
	按键返回模式寄存器(KRM0)	00H
	按键返回标志寄存器 (KRF)	00H

注 产生复位信号时以及等待振荡稳定期间时，各硬件状态中仅有 PC 的内容为不定。其他的硬件状态在复位后也保持不变。

表 14-2. 受理复位后各硬件的状态 (3/3)

硬件		受理复位后 ^{注1}
复位功能	复位控制标志寄存器(RESF)	注 2
中断	请求标志寄存器 0L, 0H (IF0L, IF0H)	00H
	屏蔽标志寄存器 0L, 0H (MK0L, MK0H)	FFH
	优先级选择标志寄存器 00L, 00H, PR10L, PR10H, (PR00L, PR00H, PR10L, PR10H)	FFH
	外部中断上升沿允许寄存器 0 (EGP0)	00H
	外部中断下降沿允许寄存器 0 (EGN0)	00H

注 1. 产生复位信号时以及等待振荡稳定期间时, 各硬件状态中仅有 PC 的内容为不定。其他的硬件状态在复位后也保持不变。

2. 这些值因复位源而异。

复位源		RESET 输入	通过执行非法指令的复位	通过 WDT 的复位	通过 SPOR 的复位	通过数据保持电源电压的复位
寄存器	RESF	清除 (0)				
	TRAP 位		置位(1)	保持	保持	清除 (0)
	WDTRF 位		保持	置位(1)		
	SPORF 位		保持	保持	置位(1)	

注 $V_{DD} \geq$ 数据保持电压时: 数据不被复位

$V_{DD} <$ 数据保持电压时: 数据被复位

数据被复位的最大电压为数据保持电压特性。

14.3 确认复位源的寄存器

14.3.1 复位控制标志寄存器(RESF)

RL78 单片机中存在着多种复位源。复位控制标志寄存器(RESF)用于存储产生了复位请求的复位源。

使用 8 位存储器操作指令读取 RESF 寄存器。

通过外部复位或数据保持下限电压引起复位，以及读取 RESF 寄存器，可清除 TRAP、WDTRF 和 SPORF 标志。

图 14-5. 复位控制标志寄存器(RESF)的格式

地址: FFFA8H 复位后: 注 1 R

符号	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	0	0	SPORF

TRAP	执行非法指令的内部复位请求 ^{注2}
0	无内部复位请求，或 RESF 寄存器被清除。
1	产生内部复位请求。

WDTRF	看门狗定时器(WDT) 的内部复位请求
0	无内部复位请求，或 RESF 寄存器被清除。
1	产生内部复位请求。

SPORF	可选择上电复位(SPOR)电路的内部复位请求
0	无内部复位请求，或 RESF 寄存器被清除。
1	产生内部复位请求。

注 1. 复位后的值因复位源而异。

2. 执行指令代码 FFH 时，产生非法指令。

通过片上调试仿真器进行仿真时，不会因执行非法指令发生内部复位。

注意事项 不可使用 1 位存储器操作指令读取数据。

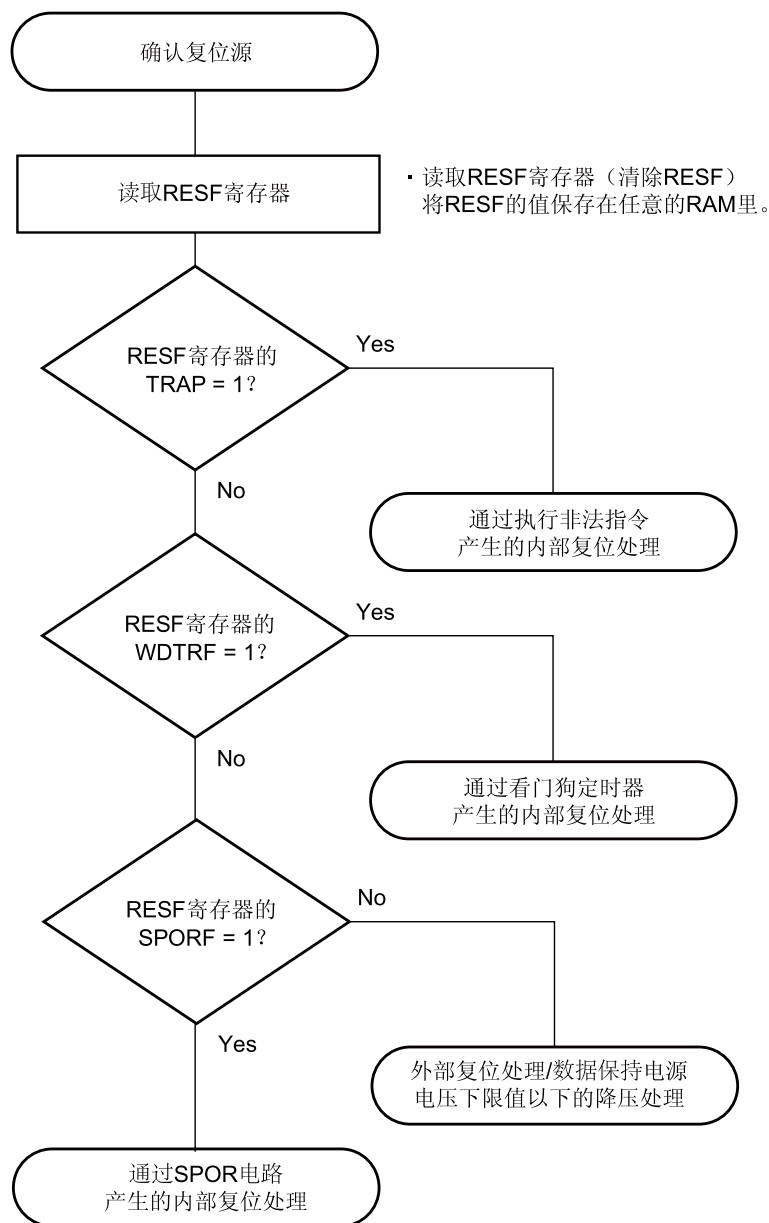
产生复位请求时 RESF 寄存器的状态如图 14-3 所示。

表 14-3. 产生复位请求时 RESF 寄存器的状态

标志 \ 复位源	RESET 输入	通过执行非法指令的复位	通过 WDT 的复位	通过 SPOR 的复位	通过数据保持下限电压的复位
TRAP 位	清除 (0)	置位(1)	保持	保持	清除 (0)
WDTRF 位		保持	置位(1)	保持	
SPORF 位		保持	保持	置位(1)	

使用 8 位存储器操作指令读取 RESF 寄存器后，该寄存器将被自动清除。
确认复位源的步骤如图 14-6 所示。

图 14-6. 确认复位源



第十五章 可选择上电复位电路

15.1 可选择上电复位电路的功能

可选择上电复位(SPOR)电路具有以下功能。

- 打开电源时产生内部复位信号。
- 电源电压(V_{DD})超过检测电压($V_{DD} \geq V_{SPOR}$)时解除复位。
- 对电源电压(V_{DD})和检测电压(V_{SPDR})进行比较, 当 $V_{DD} < V_{SPDR}$ 时产生内部复位信号。
- 电源电压的检测电压(V_{SPOR} , V_{SPDR}), 可通过选项字节(000C1H)从 4 阶段的检测电平中选择 (详情参阅 **16.2 用户选项字节的格式**)。

如果发生复位, 则复位控制标志寄存器(RESF)的位 0 (SPORF)将被设置为 1。关于 RESF 寄存器的详情, 请参阅**第十四章 复位功能**。

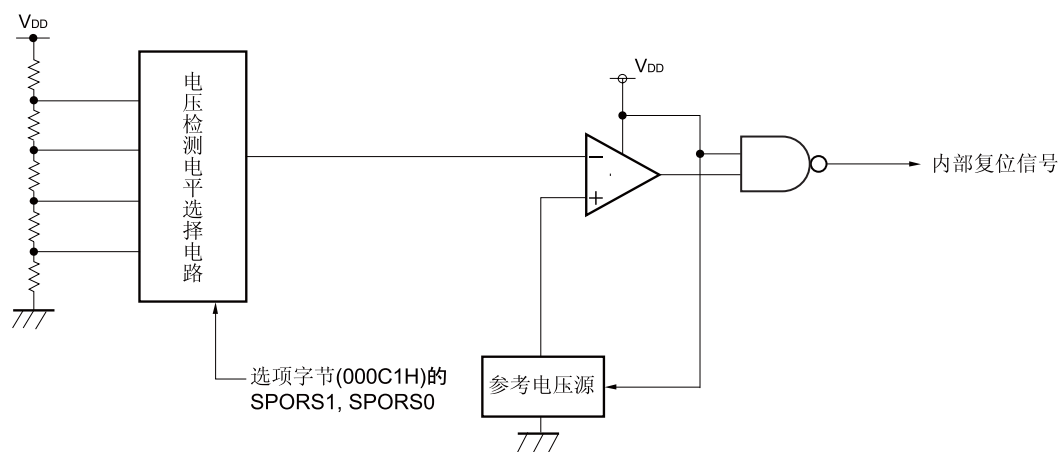
注意事项 复位控制标志寄存器(RESF)中所有标志的值被保持, 直到 V_{DD} 达到数据保持下限电压为止。

备注 V_{SPOR} : SPOR 电源上升检测电压
 V_{SPDR} : SPOR 电源下降检测电压
详情请参阅 **21.6.3 SPOR 电路特性**。

15.2 可选择上电复位电路的配置

可选择上电复位电路的框图如图 15-1 所示。

图 15-1. 可选上电复位电路的框图



15.3 可选择上电复位电路的操作

通过选项字节 000C1H 设置电压检测电平。

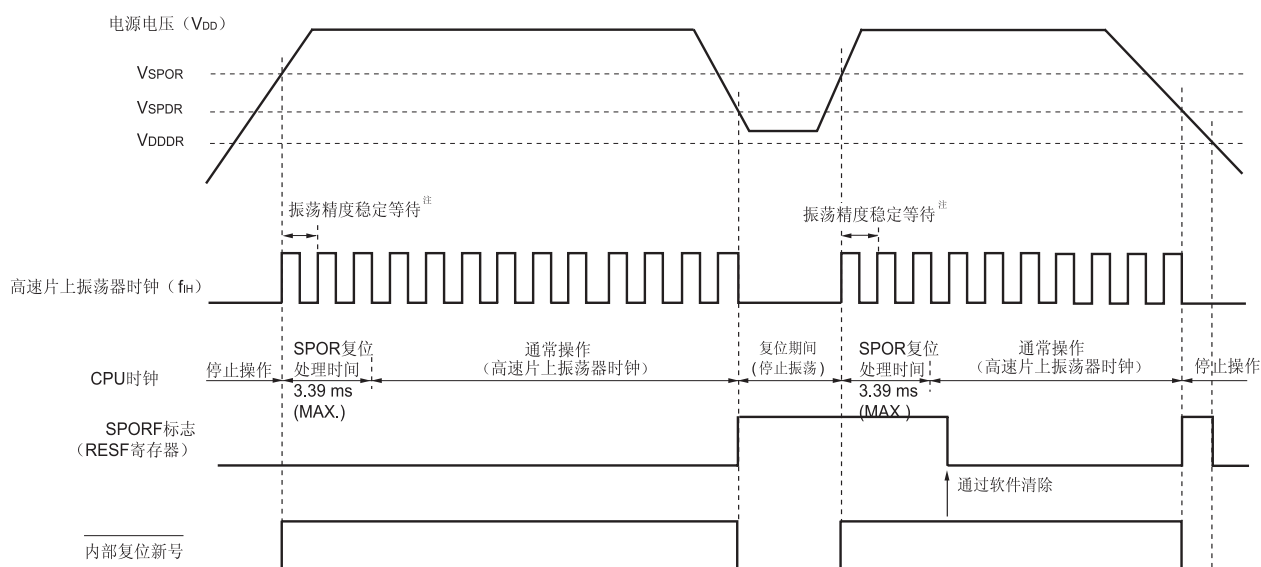
打开电源时产生内部复位信号。

保持内部复位状态，直到电源电压(V_{DD})超过电压检测电平(V_{SPOR})为止。当电源电压(V_{DD})超过电压检测电平(V_{LVD})时，解除内部复位。

工作电压下降过程中，当电源电压(V_{DD})超过电压检测电平(V_{SPDR})时，产生内部复位。

可选择上电复位电路产生内部复位信号的时序如图 15-2 所示。

图 15-2. 产生内部复位信号的时序



注 内部复位处理时间包括高速片上振荡器时钟的震荡精度稳定等待时间。

备注 V_{SPOR} : SPOR 电源上升检测电压

V_{SPDR} : SPOR 电源下降检测电压

V_{DDDR} : 数据保持电源电压

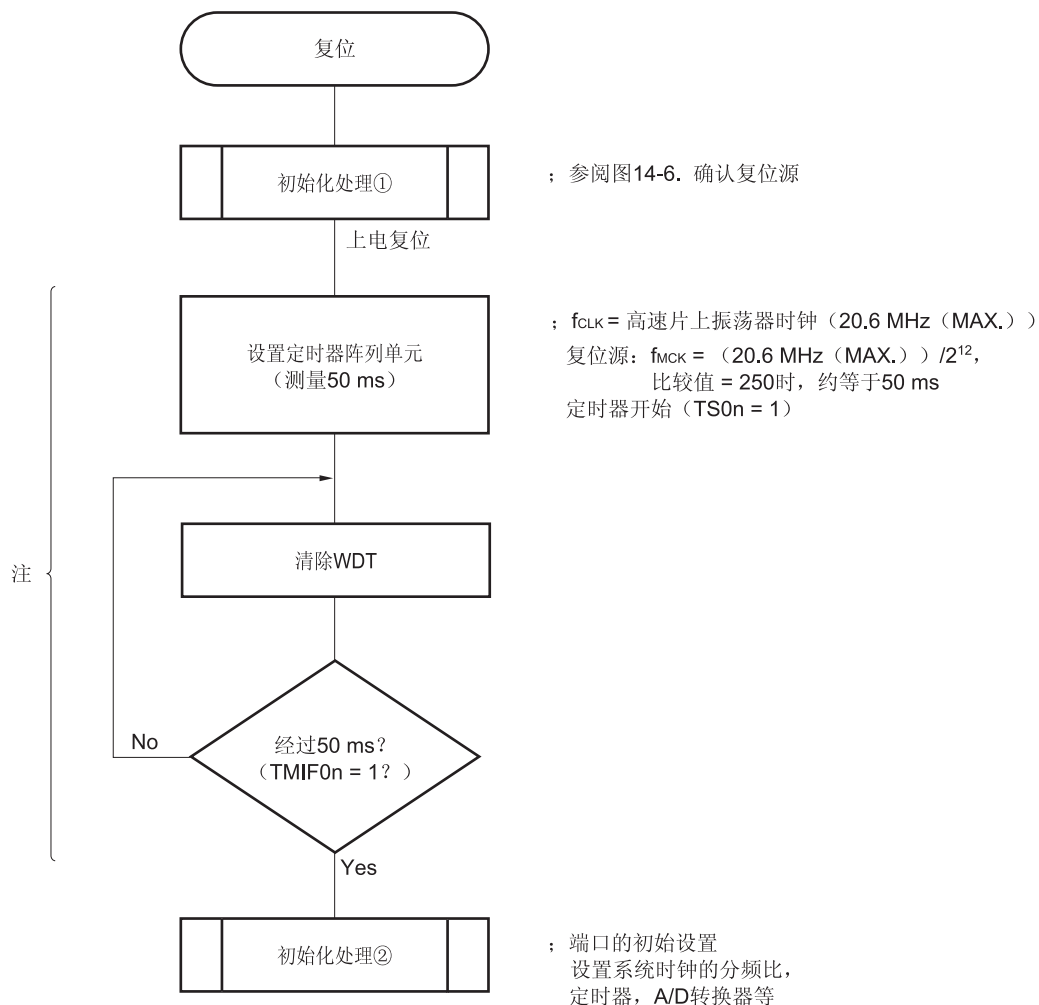
15.4 可选择上电复位电路的注意事项

一些系统的构成中，如果在一定期间内，电源电压(V_{DD})在 SPOR 检测电压(V_{SPOR} , V_{SPDR})的附近上下波动，则将重复进入复位状态和解除复位状态。通过按下述方法进行处理，可以任意指定从解除复位到单片机开始工作为止的时间。

处理方法：

使用定时器等，在解除复位后，分别使各个系统等待不同的电源电压波动期间，然后对端口等进行初期设置。

图 15-3. 电源检测电平附近的电源电压波动在 50ms 内的时软件处理示例



注 在此期间又一次产生复位时，不转移到初始化处理(2)。

备注 n: 通道编号 (n_++_) (n = 0, 1)

第十六章 选项字节

16.1 选项字节的功能

R7F0C80112ESP, R7F0C80212ESP 中的闪存地址 000C0H 至 000C3H 为选项字节区域。

选项字节由用户选项字节 (000C0H 至 000C2H)和片上调试选项字节 (000C3H)构成。

上电或复位启动时，将自动按照选项字节设置指定的功能。在使用本产品时，必须使用选项字节来设置以下功能。此外，没有分配功能的位须保持为初始值。

16.1.1 用户选项字节(000C0H 至 000C2H)

(1) 000C0H

- 看门狗定时器的操作
 - 计数器的操作允许或停止
 - HALT 或 STOP 模式时的操作停止或启用
- 看门狗定时器的时间设置
 - 看门狗定时器的溢出时间的设置
 - 看门狗定时器的间隔中断的设置

(2) 000C1H

- 设置 SPOR 检测电平 (V_{SPOR})
- 控制 P125/KR1/RESET 引脚
 - 选择 P125/KR1 或 RESET

(3) 000C2H

- 高速片上振荡器频率的设置
 - 可选择 1.25 至 20 MHz 的高速片上振荡器时钟

16.1.2 片上调试选项字节 (000C3H)

- 片上调试操作的控制
 - 禁止/允许片上调试操作

16.2 用户选项字节的格式

用户选项字节的格式如下所示。

图 16-1. 用户选项字节(000C0H)的格式

地址: 000C0H

7	6	5	4	3	2	1	0
1	1	1	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTON	看门狗定时器的操作控制
0	禁止计数器操作(复位后停止计数)
1	允许计数器操作(复位后开始计数)

WDCS2	WDCS1	WDCS0	看门狗定时器的溢出时间 (WDTRES)	看门狗定时器的间隔中断时间 (计数值溢出×3/4+3/(f _{IL} ×4))
0	0	0	(2 ⁶ -1)/ f _{IL} (2.1 ms)	1.6 ms
0	0	1	(2 ⁷ -1)/ f _{IL} (4.23 ms)	3.2 ms
0	1	0	(2 ⁸ -1)/ f _{IL} (8.5 ms)	6.4 ms
0	1	1	(2 ⁹ -1)/ f _{IL} (17.03 ms)	12.8 ms
1	0	0	(2 ¹¹ -1)/ f _{IL} (68.23 ms)	51.2 ms
1	0	1	(2 ¹³ -1)/ f _{IL} (273.03 ms)	204.8 ms
1	1	0	(2 ¹⁴ -1)/ f _{IL} (546.1 ms)	409.6 ms
1	1	1	(2 ¹⁶ -1)/ f _{IL} (2184.5 ms)	1638.4 ms

WDSTBYON	看门狗定时器的操作控制 (HALT/STOP 模式)
0	HALT/STOP 模式时, 停止计数器操作
1	HALT/STOP 模式时, 允许计数器操作

- 注意事项
1. 必须向位 5 至位 7 写入 1。
 2. 禁止设置为 WDTON=0, WDSTBON=1。

备注 f_{IL}: 低速片上振荡器时钟频率

图 16-2. 用户选项字节(000C1H)的格式

地址: 000C1H

7	6	5	4	3	2	1	0
1	1	1	PORTSELB	SPORS1	SPORS0	1	1

- 设置 SPOR 检测电压

检测电压		选项字节设置值	
V _{SPOR}	V _{SPDR}	SPORS1	SPORS0
上升沿	下降沿		
4.28 V	4.00 V	0	0
2.90 V	2.70 V	0	1
2.57 V	2.40 V	1	0
其他		禁止设置	

- 控制 P125/RESET 引脚

PORTSELB	控制 P125/KR1/RESET 引脚
0	端口功能 (P125/KR1)
1	RESET 输入 (总可连接内部上拉电阻)

注意事项 必须向位 7 至位 5、1、0 写入 1。

- 备注 1. 有关 SPOR 电路的详情, 请参阅第十五章 可选择上电复位电路。
2. 检测电压的上升沿是 TYP.值, 下降沿是 MIN.值。详情请参阅 21.6.3 SPOR 电路特性。

图 16-3. 用户选项字节(000C2H)的格式

地址: 000C2H

7	6	5	4	3	2	1	0
1	1	1	1	1	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL2	FRQSEL1	FRQSEL0	高速片上振荡器的频率
0	0	1	20 MHz
0	1	0	10 MHz
0	1	1	5 MHz
1	0	0	2.5 MHz
1	0	1	1.25 MHz
其他			禁止设置

注意事项 必须向位 7 至位 3 写入 1。

16.3 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 16-4. 片上调试选项字节(000C3H)的格式

地址: 000C3H

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	1

OCDENSET	片上调试操作的控制
0	禁止片上调试操作
1	允许片上调试操作。 [※]

注 在片上调试安全 ID 验证失败时不擦除闪存数据。

注意事项 仅限位 7 (OCDENSET)可以指定值。
必须将位 6 至位 0 设置为 0000101B。

备注 位 3 和位 1 的值在使用片上调试功能时将被覆盖，因此在设置后将变得不稳定。
但是，请确保在设置时将位 3 至位 1 设为默认值（0、1 和 0）。

16.4 选项字节的设置

除通过描述源以外，用户选项字节和片上调试选项字节也可以通过连接器选项进行设置。如下所示，即使已经在源中描述，使用连接器选项时，将优先采用连接器选项的设置。

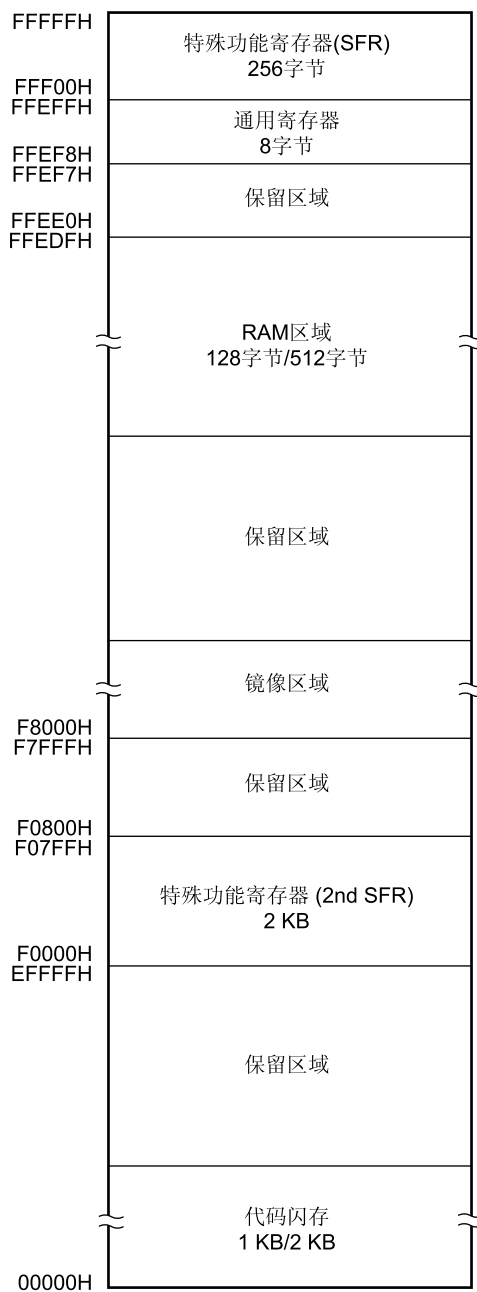
以下为选项字节设置的软件设置示例。

OPT	CSEG	OPT_BYTE	
	DB	F7H	; 允许看门狗定时器操作
			; 看门狗定时器的溢出时间为 $2^9/f_{IL}$
			; HALT/STOP 模式时，停止看门狗定时器操作
	DB	E7H	; 给 V _{SPDR} 选择 2.70 V，给 V _{SPOR} 选择 2.90 V
			; 使用端口功能（P125/KR1）
	DB	FDH	; 高速片上振荡器频率选择 1.25 MHz
	DB	85H	; 允许片上调试操作

注意事项 用汇编语言指定选项字节，将 **OPT_BYTE** 用作 **CSEG** 伪指令的重定位属性。

第十七章 闪存

RL78 单片机内置闪存，可对其执行程序的写入、擦除和改写。



闪存的编程方法如下所示。

通过闪存编程器或者外部器件（UART 通信）的串行编程对代码闪存进行改写。

- 使用闪存编程器进行串行编程（参阅 17.1）
使用专用闪存编程器进行板上和板外编程。
- 使用外部器件（UART 通信）进行串行编程（参阅 17.2）
通过与外部器件（单片机或 ASIC）的 UART 通信，进行板上编程。

17.1 使用闪存编程器进行串行编程

下列专用闪存编程器可用于将数据写入 RL78 单片机的内部闪存。

- PG-FP5, FL-PR5
- E1 片上调试仿真器

通过专用的闪存编程器，可以板上或板外向闪存写入数据。

(1) 板上编程

闪存的内容可以在将 RL78 单片机安装在目标系统之后改写。连接专用闪存编程器的连接器必须安装在目标系统中。

(2) 板外编程

在将 RL78 单片机安装到目标系统之前，可以用一个专用编程适配器（FA 系列）将数据写入闪存。

备注 FL-PR5 和 FA 系列是 Naito Densei Machida Mfg. Co., Ltd 的产品。

表 17-1. R7F0C80112ESP, R7F0C80212ESP 和专用闪存编程器的连线表

专用闪存编程器的引脚配置				引脚名称	引脚编号
信号名称		输入/输出	引脚功能		
PG-FP5, FL-PR5	E1 片上调试仿真器				
—	TOOL0	输入/输出	发送/接收信号	TOOL0/P40	1
SI/RxD	—	输入/输出	发送/接收信号		
—	$\overline{\text{RESET}}$	输出	复位信号	$\overline{\text{RESET}}$	2
/RESET	—	输出			
V _{DD}		输入/输出	V _{DD} 电压生成/ 电源监视	V _{DD}	5
GND		—	接地	V _{SS}	4
EMV _{DD}		—	TOOL 引脚驱动电源	V _{DD}	5

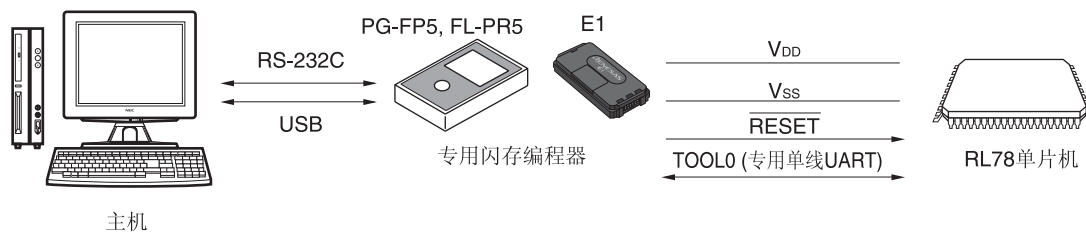
备注 在使用闪存编程器进行闪存编程时，上表未列出的引脚可以保持开路。

关于 RL78 单片机和连接器的连接内容，参见各编程器的用户手册。关于与 E1 的连接内容，请参阅 **18.1 连接 E1 片上调试仿真器**。

17.1.1 编程环境

将程序写入 RL78 单片机的闪存时所需环境如下所示。

图 17-1. 将程序写入闪存时所需环境



需要一台主机来控制专用闪存编程器。

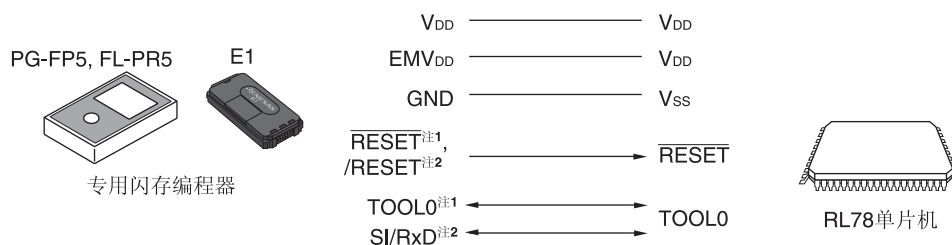
作为专用闪存编程器与 RL78 单片机之间的接口，专用的单线 UART 使用 TOOL0 引脚执行写入、擦除等操作。

17.1.2 通信模式

专用闪存编程器与 RL78 单片机之间的通信，是使用 RL78 单片机的专用单线 UART，通过 TOOL0 引脚，以串行通信方式实现的。

传送速率：固定为 115.2 kbps

图 17-2. 与专用闪存编程器的通信



- 注 1. 使用 E1 片上调试仿真器时。
 2. 使用 PG-FP5 或 FL-PR5 时。

专用闪存编程器为 RL78 单片机产生以下信号。有关详情，请参阅 PG-FP5、FL-PR5 或 E1 片上调试仿真器的各个手册。

表 17-2. 引脚连接

专用闪存编程器			RL78 单片机
信号名称		输入/输出	引脚名称
PG-FP5, FL-PR5	E1 片上调试仿真器		
V _{DD}		输入/输出	V _{DD} 电压生成/电源监视
GND		—	接地
EMV _{DD}		—	TOOL0 引脚驱动电源
/RESET	—	输出	RESET
—	RESET	输出	
—	TOOL0	输入/输出	TOOL0
SI/RxD	—	输入/输出	

17.2 使用外部器件（UART 内置）进行编程

使用通过 UART 连接到 RL78 单片机的外部器件（单片机或 ASIC），可以板上对内置闪存写入数据。

17.2.1 编程环境

向 RL78 单片机的闪存写入程序时所需的环境如下所示。

图 17-3. 向闪存写入程序时所需的环境



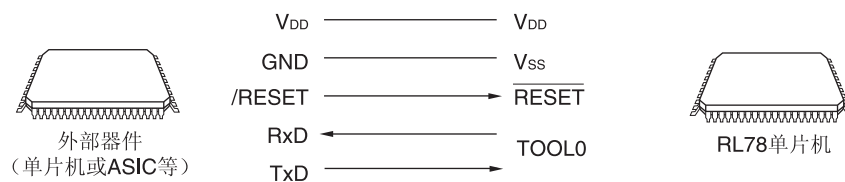
通过外部器件在板上执行对 RL78 单片机的写入和删除。不能板外写入。

17.2.2 通信方式

外部器件和 RL78 单片机之间的通信是使用 TOOL0 引脚通过专用 UART 执行串行通信。

传送速率：固定为 115.2 kbps

图 17-4. 和外部器件的通信



外部器件对 RL78 单片机产生以下的信号。

表 17-3. 引脚连接列表

外部器件			RL78单片机
信号名称	输入/输出	引脚功能	引脚名称
VDD	输入/输出	VDD电压产生/电压监视	VDD
GND	—	接地	VSS
RESETOUT	输出	复位信号输出	RESET
RxD	输入	接收信号	TOOL0
TxD	输出	发送信号	

17.3 板上引脚连接

若要使用闪存编程器板上写入闪存，则目标系统上必须提供连接着专用闪存编程器的连接器。首先，电路板上要提供用于选择正常操作模式或闪存编程模式的功能。

当设置为闪存编程模式时，所有在闪存编程过程中未使用的引脚，其状态与复位后的瞬时状态相同。因此，如果外部器件未能在复位后立即识别出该状态，则必须按以下方式处理这些引脚。

关于闪存的编程模式，请参阅 **17.4.2 闪存的编程模式**。

17.3.1 P40/TOOL0 引脚

在闪存编程模式下，通过一个外部 1 k Ω 上拉电阻将该引脚连接至专用闪存编程器。

将 P40/TOOL0 引脚用作端口引脚时，在 P40 引脚为低电平的情况下如果解除复位状态，则复位处理时间将增加几百 ms 而且解除复位状态后返回 RESF 的值为 10H。

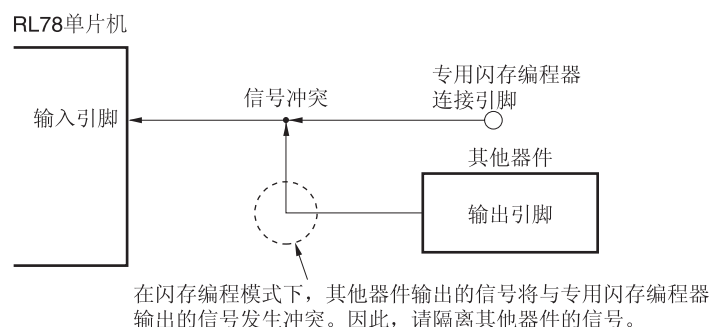
备注 SAU 引脚不用于 RL78 单片机和专用闪存编程器之间的通信，因为使用的是单线 UART（TOOL0 引脚）。

17.3.2 RESET 引脚

如果将专用闪存编程器和外部器件的复位信号连接至 $\overline{\text{RESET}}$ 引脚，而该引脚已经和板上的复位信号发生电路相连时，会发生信号冲突。为避免这种冲突，连接时，须和复位信号发生电路相隔离。

设置为闪存编程模式时，如果从用户系统输入复位信号，则不能对闪存进行正确编程。不要输入专用闪存编程器和外部器件的复位信号之外的任何信号。

图 17-5. 信号冲突 ($\overline{\text{RESET}}$ 引脚)



17.3.3 端口引脚

在闪存编程模式下，所有在闪存编程过程中未使用的引脚将进入与复位后的瞬时状态相同的状态。如果连接至端口的外部器件未能在复位后立即识别出端口状态，则必须通过一个电阻将端口引脚连接至 V_{DD} 或 V_{SS} 。

17.3.4 电源

使用闪存编程器的电源输出时，须将 V_{DD} 引脚连接至闪存编程器的 V_{DD} ，并将 V_{SS} 引脚连接至闪存编程器的 GND 。

使用板上电源时，则须按正常操作模式连接。

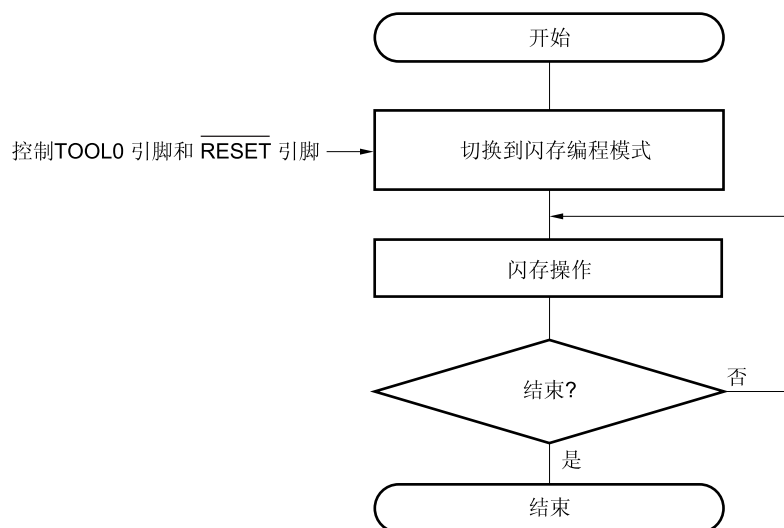
但是，在使用闪存编程器向闪存进行写操作时，即使采用板上电源电压，也必须将 V_{DD} 和 V_{SS} 引脚连接至闪存编程器的 V_{DD} 和 GND ，以使用闪存编程器上的电源监控功能。

17.4 串行编程方法

17.4.1 串行编程的步骤

通过串行编程对代码闪存进行改写的步骤如下所示。

图 17-6. 代码闪存的操作步骤



关于闪存的编程模式，请参阅 **17.4.2 闪存的编程模式**。

17.4.2 闪存的编程模式

要通过串行编程改写代码闪存的内容，须切换到闪存编程模式。

- 使用专用闪存编程器进行串行编程

连接 **RL78** 单片机和专用闪存编程器。通过和专用闪存编程器的通信，自动切换到闪存编程模式。闪存编程模式的工作电压是 **4.5V 至 5.5V**。

- 使用外部器件（UART 通信）进行串行编程

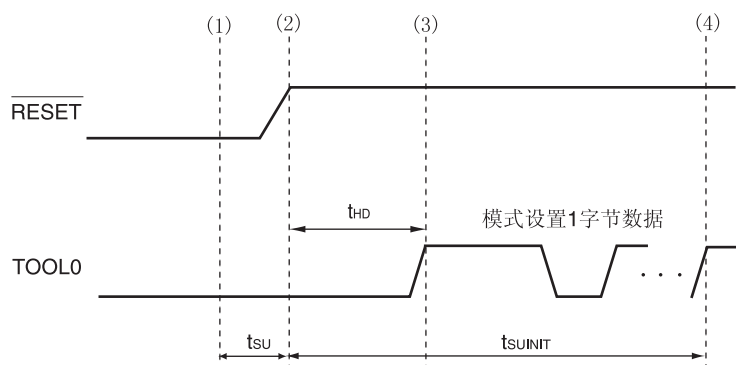
将 **TOOL0** 引脚设置为低电平后，解除复位（参阅表 17-4）。然后，按照图 17-7 所示步骤(1)到(4)，切换到闪存编程模式。

此外，闪存编程模式的工作电压为 **4.5V 至 5.5V**。

表 17-4. 解除复位时 **TOOL0** 引脚和操作模式的关系

TOOL0	操作模式
V_{DD}	通常操作模式
0 V	闪存编程模式

图 17-7. 进入闪存编程模式



- (1) 向 **TOOL0** 引脚输入低电平
- (2) 解除外部复位（必须先解除 **SPOR** 复位）
- (3) 解除 **TOOL0** 引脚的低电平
- (4) 通过执行 **UART** 接收来设置进入模式

备注 **tSUINIT**: 在此区间内，解除复位后 **100 ms** 之内必须完成初始设定通信。

tSU: 从 **TOOL0** 引脚被设置为低电平到解除外部复位的时间。

tHD: 解除外部复位后，**TOOL0** 引脚保持低电平的时间。

详情请参阅 **21.9 进入闪存编程模式的时序**。

17.4.3 通信模式

RL78 单片机的通信模式如下所示。

表 17-5. 通信模式

通信模式	标准设置 ^{注1}				使用引脚
	端口	速率 ^{注2}	频率	乘法率	
单线 UART	UART	115200 bps	—	—	TOOL0

注 1. 闪存编程器的 GUI 标准设置的设置项目。

2. 由于 UART 通信收到波特率误差和信号波形的钝化等因素影响，使用前请进行评估。

17.4.4 通信命令

RL78 单片机通过表 17-6 所示的命令来执行串行编程。

从专用闪存编程器和外部器件发送到 RL78 单片机的信号被称作“命令”，单片机执行与各命令相对应功能的处理。

表 17-6. 闪存控制命令

分类	命令名称	功能
CRC 检查	CRC check	计算检查和
擦除后写入	Write after erase	擦除闪存的数据后写入数据

17.5 使用 PG-FP5 时各指令的处理时间（参考值）

将 PG-FP5 用作专用闪存编程起时各指令的处理时间（参考值）如下所示。

表 17-7. 使用 PG-FP5 时各指令的处理时间（参考值）

PG-FP5 的指令	代码闪存	
	1 K 字节	2 K 字节
	R7F0C80112ESP	R7F0C80212ESP
删除后写入	1.0 s	1.0 s
CRC 检查	0.5 s	0.5 s

备注 各指令的处理时间（参考值）为 TYP.值，条件如下所示：

Port : TOOL0（单线 UART）

Speed : 115,200 bps

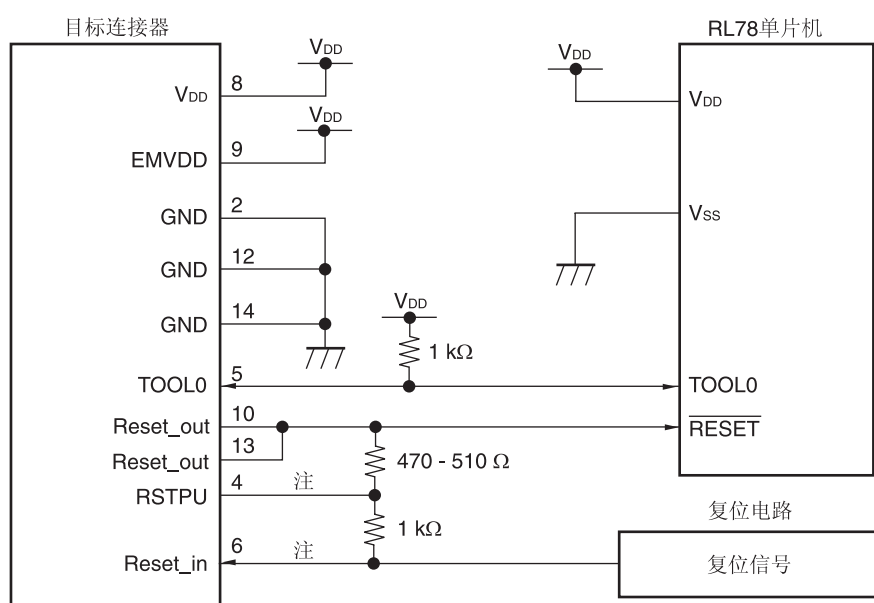
第十八章 片上调试功能

18.1 连接 E1 片上调试仿真器

RL78 单片机使用 V_{DD} 、 $\overline{\text{RESET}}$ 、TOOL0 和 V_{SS} 引脚通过 E1 片上调试仿真器与主机进行通信。通过使用 TOOL0 引脚的单线 UART 实现串行通信。

注意事项 RL78 单片机提供用于开发和评估的片上调试功能。对于量产产品，不得使用片上调试功能，因为在使用该功能时，可能会超过闪存保证的可重写次数，因而会使产品可靠性无法保证。对于使用片上调试功能时发生的问题，瑞萨电子不承担任何责任。

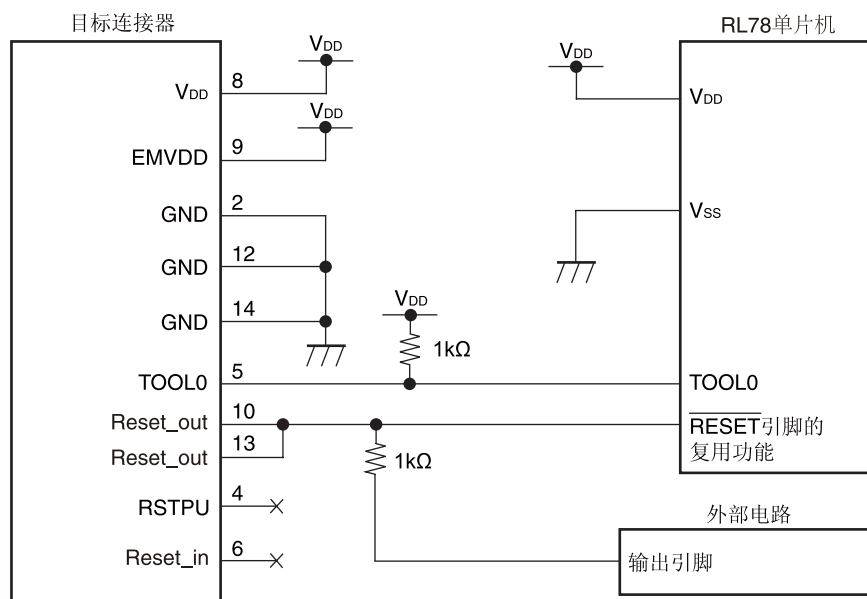
图 18-1. 和 E1 片上调试仿真器的连接示例



注 在闪存编程期间，不需要连接虚线。

使用 $\overline{\text{RESET}}$ 引脚的复用功能的目标系统，须隔离与外部电路的连接。

图 18-2. E1 片上调试仿真器和 R7F0C80112, R7F0C80212 的连接示例 (使用 $\overline{\text{RESET}}$ 引脚的复用功能时)



18.2 片上调试安全 ID

RL78 单片机在闪存 000C3H 处有一个片上调试操作控制位（参阅第十六章 选项字节），并在 000C4H 至 000CDH 处有一个片上调试安全 ID 设置区，用于防止第三方读取存储器内容。

表 18-1. 片上调试安全 ID

地址	片上调试安全 ID
000C4H 至 000CDH	任意 10 个字节的 ID 码

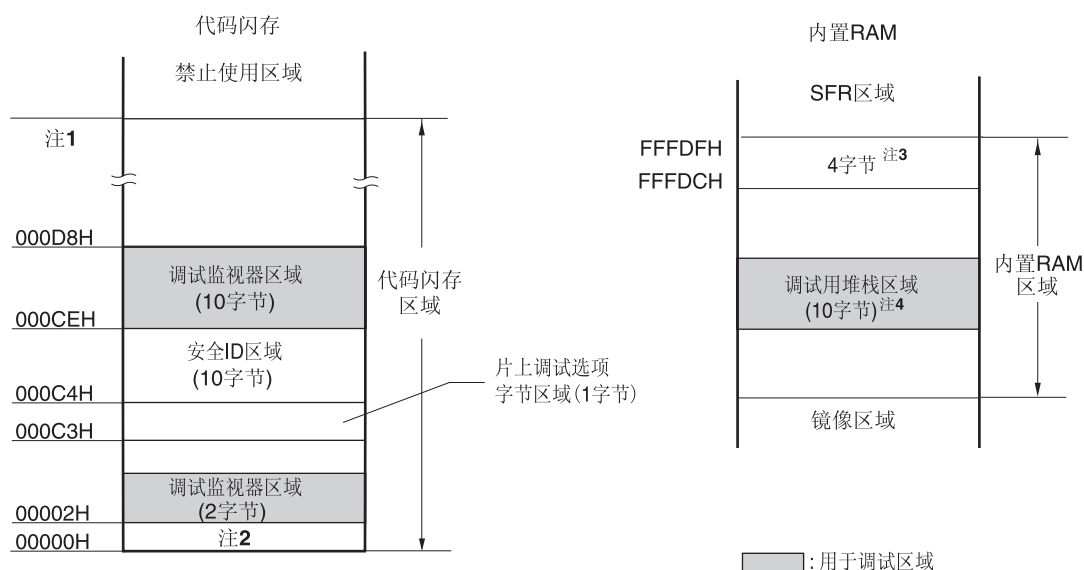
18.3 用户资源的预留

要实现 RL78 单片机与 E1 片上调试仿真器之间的通信以及各调试功能，需提前预留存储器区空间。如果使用瑞萨电子的编译器，则可通过连接器选项来进行设置。

(1) 存储器空间的预留

图 18-3 中的阴影部分是预留区，用于放置调试监控程序，因此不能将用户程序或数据分配在该空间。在使用片上调试功能时，必须保证这些空间不被用户程序使用。另外，禁止通过用户程序改写该区域。

图 18-3. 调试监控程序的内存空间分配



注 1. 地址因产品而异。

产品名称	地址
R7F0C80112ESP	003FFH
R7F0C80212ESP	007FFH

- 在调试中，复位向量被改写为分配给监控程序的地址。
- 使用实时 RAM 监视器(RRM)功能和 Dynamic Memory Modification (DMM)功能时，占用 FFEDCH 至 FFEDFH 的 4 个字节。不使用 RRM 功能和 DMM 功能时，可以用作内部 RAM。
- 由于该区域被分配在紧邻堆栈区之下，因此，该区域的地址随着堆栈的增减而变化。即所用堆栈区将额外占用 10 字节。

第十九章 十进制调整(BCD)电路

19.1 十进制调整电路的功能

通过该电路，可以获得 BCD 码（二进制编码的十进制）相加/减的 BCD 码格式的结果。

通过执行以 A 寄存器为操作数的加/减运算，并对 BCD 校正结果寄存器(BCDADJ)执行加/减运算，来获得十进制校正运算结果。

19.2 十进制调整电路使用的寄存器

十进制调整电路使用如下的寄存器。

- BCD 校正结果寄存器(BCDADJ)

19.2.1 BCD 校正结果寄存器(BCDADJ)

通过以 A 寄存器为操作数的加/减指令，获得 BCD 码的加/减结果所需要的校正值被存储于 BCDADJ 寄存器。

BCDADJ 寄存器的读取值因读取时 A 寄存器以及 CY 和 AC 标志的值而异。

使用 8 位存储器操作指令读取 BCDADJ 寄存器。

产生复位后，该寄存器为不定值。

图 19-1. BCD 校正结果寄存器的格式 (BCDADJ)

地址：F00FEH 复位后：不定 R

符号	7	6	5	4	3	2	1	0
BCDADJ								

19.3 十进制调整电路的操作

十进制调整电路的基本操作如下所示。

(1) 加法：以 BCD 码值计算出 BCD 码值与另一个 BCD 码值相加的结果

- <1> 被执行加法运算的 BCD 码值存储于 A 寄存器中。
- <2> 通过将 A 寄存器的值与第二个操作数（被加的另一个 BCD 码的值）在二进制格式下相加，二进制运算结果将存储于 A 寄存器中，校正值存储于 BCD 校正结果寄存器(BCDADJ)中。
- <3> 通过在二进制格式下将 A 寄存器的值（二进制的加法运算结果）与 BCDADJ 寄存器（校正值）相加来实现十进制校正运算，校正结果存储于 A 寄存器和 CY 标志中。

注意事项 BCDADJ 寄存器的读取值因读取时 A 寄存器以及 CY 和 AC 标志的值而异。因此，在执行指令<2>之后，必须先执行指令<3>再执行其他指令。若要在中断允许状态下执行 BCD 校正，则须在中断函数中对 A 寄存器执行进栈和出栈操作。通过 RETI 指令来恢复 PSW（CY 标志和 AC 标志）。

示例如下所示。

示例 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	—

示例 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	66H
ADD A, !BCDADJ ; <3>	00H	1	1	—

示例 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) 减法：以 BCD 码值计算出从 BCD 码值减去另一个 BCD 码值的结果

- <1> 被执行减法运算的 BCD 码值存储于 A 寄存器中。
- <2> 在二进制格式下通过从 A 寄存器减去第二个操作数（被减的 BCD 码的值），二进制的运算结果将存储于 A 寄存器中，校正值存储于 BCD 校正结果寄存器(BCDADJ)中。
- <3> 通过在二进制格式下从 A 寄存器（二进制的减运算结果）减去 BCDADJ 寄存器的值（校正值）来实现十进制校正运算，校正结果存储于 A 寄存器和 CY 标志中。

注意事项 BCDADJ 寄存器的读取值因读取时 A 寄存器以及 CY 和 AC 标志的值而异。因此，在执行指令<2>之后，必须先执行指令<3>再执行其他指令。若要在中断允许状态下执行 BCD 校正，则须在中断函数中对 A 寄存器执行进栈和出栈操作。通过 RETI 指令来恢复 PSW（CY 标志和 AC 标志）。

示例如下所示。

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	—

第二十章 指令集

本章列出了 RL78 单片机的 RL78-S1 核的指令集。有关各操作和操作代码的详情，请参阅独立文档 **RL78 族用户手册软件篇 (R01US0015E)**。

备注 RL78-S1 核与 RL78-S2 核的所有指令都相同。但是，有一部分指令占用的时钟数不同。关于 RL78-S1 核与 RL78-S2 核中占用不同时钟数的指令，请参阅 **20.2 操作列表** 的阴影部分。

20.1 操作列表使用规则

20.1.1 操作数标识符和标识方法

遵循着指令操作数标识符的标识方法，操作数列于各指令的减操作数摊一栏中（详见汇编器规范）。当存在两种或更多的标识方法时，应选择其中一种。大写字母以及#、!、!!、\$、\$!、[]和ES:符号为关键字，描述时将保持原样。每个符号的含义如下。

- #: 立即数
- !: 16 位绝对地址
- !!: 20 位绝对地址
- \$: 8 位相对地址
- \$!: 16 位相对地址
- []: 间接地址
- ES:: 扩展地址

立即数可以由一个对应的数值或标号来描述。使用标号时，必须使用#、!、!!、\$、\$!、[]和ES:符号描述。

对于操作数的寄存器标识符 *r* 和 *rp*，可使用功能名称（X、A、C 等）或绝对名称（下表 20-1 的括号中名称，R0、R1、R2 等）进行描述。

表20-1. 操作数标识符和标识方法

标识符	标识方法
<i>r</i> <i>rp</i> <i>sfr</i> <i>sfrp</i>	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) 特殊功能寄存器符号(SFR 符号) FFF00H 至 FFFFFH 区 特殊功能寄存器符号(可 16 位操作的 SFR 符号。仅用于偶数地址 ^{*)} FFF00H 至 FFFFFH 区
<i>saddr</i> <i>saddrp</i>	FFE20H 至 FFF1FH 区立即数或标号 FFE20H 至 FFF1FH 区立即数或标号(仅用于偶数地址 ^{*)})
<i>addr20</i> <i>addr16</i> <i>addr5</i>	00000H 至 FFFFFH 区的立即数或标号 0000H 至 FFFFH 区的立即数或标号 (自动添加 F 至顶部。16 位数据时仅用于偶数地址 ^{*)}) 0080H 至 00BFH 区的立即数或标号 (指定位 5 至 1，仅用于偶数地址)
<i>word</i> <i>byte</i> <i>bit</i>	16 位立即数或标号 8 位立即数或标号 3 位立即数或标号

注 指定为偶数地址时，位0 = 0。

备注 特殊功能寄存器的符号，可用于描述操作数 *sfr*。关于特殊功能寄存器的符号，请参阅表 3-4. **SFR 列表**。扩展特殊功能寄存器的符号，可用于描述操作数 *addr16*。关于扩展特殊功能寄存器的符号，请参阅表 3-5. **扩展 SFR (2nd SFR) 列表**。

20.1.2 操作栏的说明

在各指令的操作栏中，指令执行时的动作作用以下的符号表示。

表20-2. 操作栏的符号

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
IE	中断要求允许标志
()	()内的地址或寄存器的内容所指示的存储器的内容
X _H , X _L	16 位寄存器: X _H = 高 8 位, X _L = 低 8 位
X _S , X _H , X _L	20 位寄存器: X _S = (位 19 至 16), X _H = (位 15 至 8), X _L = (位 7 至 0)
∧	逻辑与 (AND)
∨	逻辑或 (OR)
⊕	异或 (exclusive OR)
—	反转数据
addr5	16 位立即数 (仅限 0080H 至 00BFH 的偶数地址)
addr16	16 位立即数
addr20	20 位立即数
jdisp8	有符号的 8 位数据 (偏移量)
jdisp16	有符号的 16 位数据 (偏移量)

20.1.3 标志栏的说明

在各指令的标志栏中，指令执行时标志值的变化用以下符号表示。

表20-3. 标志栏的符号

符号	标志值的变化
(Blank)	无变化
0	清除为 0
1	设置为 1
×	根据结果设置为 1/清除为 0
R	恢复以前保存的值

20.1.4 PREFIX 指令

通过将“ES:”作为 PREFIX 指令码置于指令之前，可以将可存取数据区域从 64 KB 空间（F0000H 至 FFFFFH 区）扩展至 1 MB 空间（00000H 至 FFFFFH 区）。把 PREFIX 指令码作为前缀附加到目标指令上时，只有正好位于 PREFIX 指令码之后的一条指令可以按照添加了 ES 寄存器值的地址被执行。

在 PREFIX 指令码与紧随其后的指令之间，不接受中断和 DMA 传送。

表20-4. PREFIX指令码的示例

指令	指令码				
	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 指令之前，以 MOV ES, A 等指令预先设定 ES 寄存器的值。

20.2 操作列表

表20-5. 操作列表(1/17)

指令组	助记符	操作数	字节	时钟		时钟	标志
				注1	注2		Z AC CY
8 位数据 传送	MOV	r, #byte	2	1	—	$r \leftarrow \text{byte}$	
		PSW, #byte	3	3	—	$\text{PSW} \leftarrow \text{byte}$	× × ×
		CS, #byte	3	1	—	$\text{CS} \leftarrow \text{byte}$	
		ES, #byte	2	1	—	$\text{ES} \leftarrow \text{byte}$	
		!addr16, #byte	4	1	—	$(\text{addr16}) \leftarrow \text{byte}$	
		ES:!addr16, #byte	5	2	—	$(\text{ES}, \text{addr16}) \leftarrow \text{byte}$	
		saddr, #byte	3	1	—	$(\text{saddr}) \leftarrow \text{byte}$	
		sfr, #byte	3	1	—	$\text{sfr} \leftarrow \text{byte}$	
		[DE+byte], #byte	3	1	—	$(\text{DE}+\text{byte}) \leftarrow \text{byte}$	
		ES:[DE+byte], #byte	4	2	—	$((\text{ES}, \text{DE})+\text{byte}) \leftarrow \text{byte}$	
		[HL+byte], #byte	3	1	—	$(\text{HL}+\text{byte}) \leftarrow \text{byte}$	
		ES:[HL+byte], #byte	4	2	—	$((\text{ES}, \text{HL})+\text{byte}) \leftarrow \text{byte}$	
		[SP+byte], #byte	3	1	—	$(\text{SP}+\text{byte}) \leftarrow \text{byte}$	
		word[B], #byte	4	1	—	$(\text{B}+\text{word}) \leftarrow \text{byte}$	
		ES:word[B], #byte	5	2	—	$((\text{ES}, \text{B})+\text{word}) \leftarrow \text{byte}$	
		word[C], #byte	4	1	—	$(\text{C}+\text{word}) \leftarrow \text{byte}$	
		ES:word[C], #byte	5	2	—	$((\text{ES}, \text{C})+\text{word}) \leftarrow \text{byte}$	
		word[BC], #byte	4	1	—	$(\text{BC}+\text{word}) \leftarrow \text{byte}$	
		ES:word[BC], #byte	5	2	—	$((\text{ES}, \text{BC})+\text{word}) \leftarrow \text{byte}$	
		A, r ^{注3}	1	1	—	$\text{A} \leftarrow r$	
		r, A ^{注3}	1	1	—	$r \leftarrow \text{A}$	
		A, PSW	2	1	—	$\text{A} \leftarrow \text{PSW}$	
		PSW, A	2	3	—	$\text{PSW} \leftarrow \text{A}$	× × ×
		A, CS	2	1	—	$\text{A} \leftarrow \text{CS}$	
		CS, A	2	1	—	$\text{CS} \leftarrow \text{A}$	
		A, ES	2	1	—	$\text{A} \leftarrow \text{ES}$	
		ES, A	2	1	—	$\text{ES} \leftarrow \text{A}$	
		A, !addr16	3	1	4	$\text{A} \leftarrow (\text{addr16})$	
		A, ES:!addr16	4	2	5	$\text{A} \leftarrow (\text{ES}, \text{addr16})$	
		!addr16, A	3	1	—	$(\text{addr16}) \leftarrow \text{A}$	
		ES:!addr16, A	4	2	—	$(\text{ES}, \text{addr16}) \leftarrow \text{A}$	
		A, saddr	2	1	—	$\text{A} \leftarrow (\text{saddr})$	
		saddr, A	2	1	—	$(\text{saddr}) \leftarrow \text{A}$	

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(fCLK)。

2. 程序存储器区域被存取时的 CPU 时钟数(fCLK)。

3. 除 $r = \text{A}$ 以外。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(2/17)

指令组	助记符	操作数	字节	时钟		时钟	标志		
				注1	注2		Z	AC	CY
8 位数据 传送	MOV	A, sfr	2	1	—	$A \leftarrow \text{sfr}$			
		sfr, A	2	1	—	$\text{sfr} \leftarrow A$			
		A, [DE]	1	1	4	$A \leftarrow (\text{DE})$			
		[DE], A	1	1	—	$(\text{DE}) \leftarrow A$			
		A, ES:[DE]	2	2	5	$A \leftarrow (\text{ES}, \text{DE})$			
		ES:[DE], A	2	2	—	$(\text{ES}, \text{DE}) \leftarrow A$			
		A, [HL]	1	1	4	$A \leftarrow (\text{HL})$			
		[HL], A	1	1	—	$(\text{HL}) \leftarrow A$			
		A, ES:[HL]	2	2	5	$A \leftarrow (\text{ES}, \text{HL})$			
		ES:[HL], A	2	2	—	$(\text{ES}, \text{HL}) \leftarrow A$			
		A, [DE+byte]	2	1	4	$A \leftarrow (\text{DE} + \text{byte})$			
		[DE+byte], A	2	1	—	$(\text{DE} + \text{byte}) \leftarrow A$			
		A, ES:[DE+byte]	3	2	5	$A \leftarrow ((\text{ES}, \text{DE}) + \text{byte})$			
		ES:[DE+byte], A	3	2	—	$((\text{ES}, \text{DE}) + \text{byte}) \leftarrow A$			
		A, [HL+byte]	2	1	4	$A \leftarrow (\text{HL} + \text{byte})$			
		[HL+byte], A	2	1	—	$(\text{HL} + \text{byte}) \leftarrow A$			
		A, ES:[HL+byte]	3	2	5	$A \leftarrow ((\text{ES}, \text{HL}) + \text{byte})$			
		ES:[HL+byte], A	3	2	—	$((\text{ES}, \text{HL}) + \text{byte}) \leftarrow A$			
		A, [SP+byte]	2	1	—	$A \leftarrow (\text{SP} + \text{byte})$			
		[SP+byte], A	2	1	—	$(\text{SP} + \text{byte}) \leftarrow A$			
		A, word[B]	3	1	4	$A \leftarrow (\text{B} + \text{word})$			
		word[B], A	3	1	—	$(\text{B} + \text{word}) \leftarrow A$			
		A, ES:word[B]	4	2	5	$A \leftarrow ((\text{ES}, \text{B}) + \text{word})$			
		ES:word[B], A	4	2	—	$((\text{ES}, \text{B}) + \text{word}) \leftarrow A$			
		A, word[C]	3	1	4	$A \leftarrow (\text{C} + \text{word})$			
		word[C], A	3	1	—	$(\text{C} + \text{word}) \leftarrow A$			
		A, ES:word[C]	4	2	5	$A \leftarrow ((\text{ES}, \text{C}) + \text{word})$			
		ES:word[C], A	4	2	—	$((\text{ES}, \text{C}) + \text{word}) \leftarrow A$			
		A, word[BC]	3	1	4	$A \leftarrow (\text{BC} + \text{word})$			
		word[BC], A	3	1	—	$(\text{BC} + \text{word}) \leftarrow A$			
		A, ES:word[BC]	4	2	5	$A \leftarrow ((\text{ES}, \text{BC}) + \text{word})$			
		ES:word[BC], A	4	2	—	$((\text{ES}, \text{BC}) + \text{word}) \leftarrow A$			

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(fCLK)。

2. 程序存储器区域被存取时的 CPU 时钟数(fCLK)。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(3/17)

指令组	助记符	操作数	字节	时钟		时钟	标志
				注1	注2		Z AC CY
8 位数据 传送	MOV	A, [HL+B]	2	1	4	$A \leftarrow (HL + B)$	
		[HL+B], A	2	1	—	$(HL + B) \leftarrow A$	
		A, ES:[HL+B]	3	2	5	$A \leftarrow ((ES, HL) + B)$	
		ES:[HL+B], A	3	2	—	$((ES, HL) + B) \leftarrow A$	
		A, [HL+C]	2	1	4	$A \leftarrow (HL + C)$	
		[HL+C], A	2	1	—	$(HL + C) \leftarrow A$	
		A, ES:[HL+C]	3	2	5	$A \leftarrow ((ES, HL) + C)$	
		ES:[HL+C], A	3	2	—	$((ES, HL) + C) \leftarrow A$	
		X, laddr16	3	1	4	$X \leftarrow (addr16)$	
		X, ES:laddr16	4	2	5	$X \leftarrow (ES, addr16)$	
		X, saddr	2	1	—	$X \leftarrow (saddr)$	
		B, laddr16	3	1	4	$B \leftarrow (addr16)$	
		B, ES:laddr16	4	2	5	$B \leftarrow (ES, addr16)$	
		B, saddr	2	1	—	$B \leftarrow (saddr)$	
		C, laddr16	3	1	4	$C \leftarrow (addr16)$	
		C, ES:laddr16	4	2	5	$C \leftarrow (ES, addr16)$	
		C, saddr	2	1	—	$C \leftarrow (saddr)$	
		ES, saddr	3	1	—	$ES \leftarrow (saddr)$	
	XCH	A, r ^{※3}	1 (r = X) 2 (r = X以外)	1	—	$A \leftrightarrow r$	
		A, laddr16	4	2	—	$A \leftrightarrow (addr16)$	
		A, ES:laddr16	5	3	—	$A \leftrightarrow (ES, addr16)$	
		A, saddr	3	2	—	$A \leftrightarrow (saddr)$	
		A, sfr	3	2	—	$A \leftrightarrow sfr$	
		A, [DE]	2	2	—	$A \leftrightarrow (DE)$	
		A, ES:[DE]	3	3	—	$A \leftrightarrow (ES, DE)$	
		A, [HL]	2	2	—	$A \leftrightarrow (HL)$	
		A, ES:[HL]	3	3	—	$A \leftrightarrow (ES, HL)$	
		A, [DE+byte]	3	2	—	$A \leftrightarrow (DE + \text{byte})$	
		A, ES:[DE+byte]	4	3	—	$A \leftrightarrow ((ES, DE) + \text{byte})$	
		A, [HL+byte]	3	2	—	$A \leftrightarrow (HL + \text{byte})$	
		A, ES:[HL+byte]	4	3	—	$A \leftrightarrow ((ES, HL) + \text{byte})$	

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(f_{CLK})。

2. 程序存储器区域被存取时的 CPU 时钟数(f_{CLK})。

3. 除 r = A 以外。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(4/17)

指令组	助记符	操作数	字节	时钟		时钟	标志		
				注1	注2		Z	AC	CY
8 位数据 传送	XCH	A, [HL+B]	2	2	—	$A \leftrightarrow (HL+B)$			
		A, ES:[HL+B]	3	3	—	$A \leftrightarrow ((ES, HL)+B)$			
		A, [HL+C]	2	2	—	$A \leftrightarrow (HL+C)$			
		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	—	$C \leftarrow 01H$			
		!addr16	3	1	—	$(addr16) \leftarrow 01H$			
		ES:!addr16	4	2	—	$(ES, addr16) \leftarrow 01H$			
		saddr	2	1	—	$(saddr) \leftarrow 01H$			
	CLRB	A	1	1	—	$A \leftarrow 00H$			
		X	1	1	—	$X \leftarrow 00H$			
		B	1	1	—	$B \leftarrow 00H$			
		C	1	1	—	$C \leftarrow 00H$			
		!addr16	3	1	—	$(addr16) \leftarrow 00H$			
		ES:!addr16	4	2	—	$(ES, addr16) \leftarrow 00H$			
		saddr	2	1	—	$(saddr) \leftarrow 00H$			
	MOVS	[HL+byte], X	3	1	—	$(HL+byte) \leftarrow X$	×		×
		ES:[HL+byte], X	4	2	—	$(ES, HL+byte) \leftarrow X$	×		×
16 位数据 传送	MOVW	rp, #word	3	2	—	$rp \leftarrow word$			
		saddrp, #word	4	2	—	$(saddrp) \leftarrow word$			
		sfrp, #word	4	2	—	$sfrp \leftarrow word$			
		AX, rp ^{※3}	1	2	—	$AX \leftarrow rp$			
		rp, AX ^{※3}	1	2	—	$rp \leftarrow AX$			
		AX, !addr16	3	2	5	$AX \leftarrow (addr16)$			
		!addr16, AX	3	2	—	$(addr16) \leftarrow AX$			
		AX, ES:!addr16	4	3	6	$AX \leftarrow (ES, addr16)$			
		ES:!addr16, AX	4	3	—	$(ES, addr16) \leftarrow AX$			
		AX, saddrp	2	2	—	$AX \leftarrow (saddrp)$			
		saddrp, AX	2	2	—	$(saddrp) \leftarrow AX$			
		AX, sfrp	2	2	—	$AX \leftarrow sfrp$			
		sfrp, AX	2	2	—	$sfrp \leftarrow AX$			

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(fCLK)。

2. 程序存储器区域被存取时的 CPU 时钟数(fCLK)。

3. 除 rp = AX 以外。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(5/17)

指令组	助记符	操作数	字节	时钟		时钟	标志
				注1	注2		Z AC CY
16 位 数 数据传送	MOVW	AX, [DE]	1	2	5	$AX \leftarrow (DE)$	
		[DE], AX	1	2	—	$(DE) \leftarrow AX$	
		AX, ES:[DE]	2	3	6	$AX \leftarrow (ES, DE)$	
		ES:[DE], AX	2	3	—	$(ES, DE) \leftarrow AX$	
		AX, [HL]	1	2	5	$AX \leftarrow (HL)$	
		[HL], AX	1	2	—	$(HL) \leftarrow AX$	
		AX, ES:[HL]	2	3	6	$AX \leftarrow (ES, HL)$	
		ES:[HL], AX	2	3	—	$(ES, HL) \leftarrow AX$	
		AX, [DE+byte]	2	2	5	$AX \leftarrow (DE+byte)$	
		[DE+byte], AX	2	2	—	$(DE+byte) \leftarrow AX$	
		AX, ES:[DE+byte]	3	3	6	$AX \leftarrow ((ES, DE) + byte)$	
		ES:[DE+byte], AX	3	3	—	$((ES, DE) + byte) \leftarrow AX$	
		AX, [HL+byte]	2	2	5	$AX \leftarrow (HL + byte)$	
		[HL+byte], AX	2	2	—	$(HL + byte) \leftarrow AX$	
		AX, ES:[HL+byte]	3	3	6	$AX \leftarrow ((ES, HL) + byte)$	
		ES:[HL+byte], AX	3	3	—	$((ES, HL) + byte) \leftarrow AX$	
		AX, [SP+byte]	2	2	—	$AX \leftarrow (SP + byte)$	
		[SP+byte], AX	2	2	—	$(SP + byte) \leftarrow AX$	
		AX, word[B]	3	2	5	$AX \leftarrow (B + word)$	
		word[B], AX	3	2	—	$(B + word) \leftarrow AX$	
		AX, ES:word[B]	4	3	6	$AX \leftarrow ((ES, B) + word)$	
		ES:word[B], AX	4	3	—	$((ES, B) + word) \leftarrow AX$	
		AX, word[C]	3	2	5	$AX \leftarrow (C + word)$	
		word[C], AX	3	2	—	$(C + word) \leftarrow AX$	
		AX, ES:word[C]	4	3	6	$AX \leftarrow ((ES, C) + word)$	
		ES:word[C], AX	4	3	—	$((ES, C) + word) \leftarrow AX$	
		AX, word[BC]	3	2	5	$AX \leftarrow (BC + word)$	
		word[BC], AX	3	2	—	$(BC + word) \leftarrow AX$	
		AX, ES:word[BC]	4	3	6	$AX \leftarrow ((ES, BC) + word)$	
		ES:word[BC], AX	4	3	—	$((ES, BC) + word) \leftarrow AX$	

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(f_{CLK})。

2. 程序存储器区域被存取时的 CPU 时钟数(f_{CLK})。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(6/17)

指令组	助记符	操作数	字节	时钟		时钟	标志
				注1	注2		Z AC CY
16 位 数 据 传 送	MOVW	BC, laddr16	3	2	5	BC ← (addr16)	
		BC, ES:laddr16	4	3	6	BC ← (ES, addr16)	
		DE, laddr16	3	2	5	DE ← (addr16)	
		DE, ES:laddr16	4	3	6	DE ← (ES, addr16)	
		HL, laddr16	3	2	5	HL ← (addr16)	
		HL, ES:laddr16	4	3	6	HL ← (ES, addr16)	
		BC, saddrp	2	2	-	BC ← (saddrp)	
		DE, saddrp	2	2	-	DE ← (saddrp)	
		HL, saddrp	2	2	-	HL ← (saddrp)	
	XCHW	AX, rp ^{注3}	1	2	-	AX ↔ rp	
	ONEW	AX	1	2	-	AX ← 0001H	
		BC	1	2	-	BC ← 0001H	
	CLRW	AX	1	2	-	AX ← 0000H	
		BC	1	2	-	BC ← 0000H	
8 位 操 作	ADD	A, #byte	2	1	-	A, CY ← A + byte	× × ×
		saddr, #byte	3	2	-	(saddr), CY ← (saddr)+byte	× × ×
		A, r ^{注4}	2	1	-	A, CY ← A + r	× × ×
		r, A	2	1	-	r, CY ← r + A	× × ×
		A, laddr16	3	1	4	A, CY ← A + (addr16)	× × ×
		A, ES:laddr16	4	2	5	A, CY ← A + (ES, addr16)	× × ×
		A, saddr	2	1	-	A, CY ← A + (saddr)	× × ×
		A, [HL]	1	1	4	A, CY ← A + (HL)	× × ×
		A, ES:[HL]	2	2	5	A, CY ← A + (ES, HL)	× × ×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL+byte)	× × ×
		A, ES:[HL+byte]	3	2	5	A, CY ← A + ((ES, HL)+byte)	× × ×
		A, [HL+B]	2	1	4	A, CY ← A + (HL+B)	× × ×
		A, ES:[HL+B]	3	2	5	A, CY ← A + ((ES, HL)+B)	× × ×
		A, [HL+C]	2	1	4	A, CY ← A + (HL+C)	× × ×
		A, ES:[HL+C]	3	2	5	A, CY ← A + ((ES, HL) + C)	× × ×

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(f_{CLK})。

2. 程序存储器区域被存取时的 CPU 时钟数(f_{CLK})。

3. 除 rp = AX 以外。

4. 除 r = A 以外。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(7/17)

指令组	助记符	操作数	字节	时钟		时钟	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, $r^{\#3}$	2	1	–	$A, CY \leftarrow A + r + CY$	×	×	×
		r, A	2	1	–	$r, CY \leftarrow r + A + CY$	×	×	×
		A, laddr16	3	1	4	$A, CY \leftarrow A + (\text{addr16}) + CY$	×	×	×
		A, ES:laddr16	4	2	5	$A, CY \leftarrow A + (ES, \text{addr16}) + CY$	×	×	×
		A, saddr	2	1	–	$A, CY \leftarrow A + (saddr) + CY$	×	×	×
		A, [HL]	1	1	4	$A, CY \leftarrow A + (HL) + CY$	×	×	×
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A + (ES, HL) + CY$	×	×	×
		A, [HL+byte]	2	1	4	$A, CY \leftarrow A + (HL + \text{byte}) + CY$	×	×	×
		A, ES:[HL+byte]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + \text{byte}) + CY$	×	×	×
		A, [HL+B]	2	1	4	$A, CY \leftarrow A + (HL + B) + CY$	×	×	×
		A, ES:[HL+B]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + B) + CY$	×	×	×
		A, [HL+C]	2	1	4	$A, CY \leftarrow A + (HL + C) + CY$	×	×	×
		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, $r^{\#3}$	2	1	–	$A, CY \leftarrow A - r$	×	×	×
		r, A	2	1	–	$r, CY \leftarrow r - A$	×	×	×
		A, laddr16	3	1	4	$A, CY \leftarrow A - (\text{addr16})$	×	×	×
		A, ES:laddr16	4	2	5	$A, CY \leftarrow A - (ES, \text{addr16})$	×	×	×
		A, saddr	2	1	–	$A, CY \leftarrow A - (saddr)$	×	×	×
		A, [HL]	1	1	4	$A, CY \leftarrow A - (HL)$	×	×	×
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A - (ES, HL)$	×	×	×
		A, [HL+byte]	2	1	4	$A, CY \leftarrow A - (HL + \text{byte})$	×	×	×
		A, ES:[HL+byte]	3	2	5	$A, CY \leftarrow A - ((ES, HL) + \text{byte})$	×	×	×
		A, [HL+B]	2	1	4	$A, CY \leftarrow A - (HL + B)$	×	×	×
		A, ES:[HL+B]	3	2	5	$A, CY \leftarrow A - ((ES, HL) + B)$	×	×	×
		A, [HL+C]	2	1	4	$A, CY \leftarrow A - (HL + C)$	×	×	×
		A, ES:[HL+C]	3	2	5	$A, CY \leftarrow A - ((ES, HL) + C)$	×	×	×

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(f_{CLK})。

2. 程序存储器区域被存取时的 CPU 时钟数(f_{CLK})。

3. 除 r = A 以外。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(8/17)

指令组	助记符	操作数	字节	时钟		时钟	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, $r^{\#3}$	2	1	–	$A, CY \leftarrow A - r - CY$	×	×	×
		r, A	2	1	–	$r, CY \leftarrow r - A - CY$	×	×	×
		A, laddr16	3	1	4	$A, CY \leftarrow A - (\text{addr16}) - CY$	×	×	×
		A, ES:laddr16	4	2	5	$A, CY \leftarrow A - (\text{ES}, \text{addr16}) - CY$	×	×	×
		A, saddr	2	1	–	$A, CY \leftarrow A - (saddr) - CY$	×	×	×
		A, [HL]	1	1	4	$A, CY \leftarrow A - (\text{HL}) - CY$	×	×	×
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A - (\text{ES}, \text{HL}) - CY$	×	×	×
		A, [HL+byte]	2	1	4	$A, CY \leftarrow A - (\text{HL} + \text{byte}) - CY$	×	×	×
		A, ES:[HL+byte]	3	2	5	$A, CY \leftarrow A - ((\text{ES}, \text{HL}) + \text{byte}) - CY$	×	×	×
		A, [HL+B]	2	1	4	$A, CY \leftarrow A - (\text{HL} + B) - CY$	×	×	×
		A, ES:[HL+B]	3	2	5	$A, CY \leftarrow A - ((\text{ES}, \text{HL}) + B) - CY$	×	×	×
		A, [HL+C]	2	1	4	$A, CY \leftarrow A - (\text{HL} + C) - CY$	×	×	×
		A, ES:[HL+C]	3	2	5	$A, CY \leftarrow A - ((\text{ES}, \text{HL}) + C) - CY$	×	×	×
	AND	A, #byte	2	1	–	$A \leftarrow A \wedge \text{byte}$	×		
		saddr, #byte	3	2	–	$(saddr) \leftarrow (saddr) \wedge \text{byte}$	×		
		A, $r^{\#3}$	2	1	–	$A \leftarrow A \wedge r$	×		
		r, A	2	1	–	$R \leftarrow r \wedge A$	×		
		A, laddr16	3	1	4	$A \leftarrow A \wedge (\text{addr16})$	×		
		A, ES:laddr16	4	2	5	$A \leftarrow A \wedge (\text{ES}, \text{addr16})$	×		
		A, saddr	2	1	–	$A \leftarrow A \wedge (saddr)$	×		
		A, [HL]	1	1	4	$A \leftarrow A \wedge (\text{HL})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \wedge (\text{ES}, \text{HL})$	×		
		A, [HL+byte]	2	1	4	$A \leftarrow A \wedge (\text{HL} + \text{byte})$	×		
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \wedge ((\text{ES}, \text{HL}) + \text{byte})$	×		
		A, [HL+B]	2	1	4	$A \leftarrow A \wedge (\text{HL} + B)$	×		
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \wedge ((\text{ES}, \text{HL}) + B)$	×		
		A, [HL+C]	2	1	4	$A \leftarrow A \wedge (\text{HL} + C)$	×		
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \wedge ((\text{ES}, \text{HL}) + C)$	×		

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(fCLK)。

2. 程序存储器区域被存取时的 CPU 时钟数(fCLK)。

3. 除 $r = A$ 以外。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(9/17)

指令组	助记符	操作数	字节	时钟		时钟	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, $r^{\#3}$	2	1	—	$A \leftarrow A \vee r$	×
		r, A	2	1	—	$r \leftarrow r \vee A$	×
		A, laddr16	3	1	4	$A \leftarrow A \vee (\text{addr16})$	×
		A, ES:laddr16	4	2	5	$A \leftarrow A \vee (\text{ES:addr16})$	×
		A, saddr	2	1	—	$A \leftarrow A \vee (\text{saddr})$	×
		A, [HL]	1	1	4	$A \leftarrow A \vee (\text{H})$	×
		A, ES:[HL]	2	2	5	$A \leftarrow A \vee (\text{ES:HL})$	×
		A, [HL+byte]	2	1	4	$A \leftarrow A \vee (\text{HL} + \text{byte})$	×
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + \text{byte})$	×
		A, [HL+B]	2	1	4	$A \leftarrow A \vee (\text{HL} + \text{B})$	×
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + \text{B})$	×
		A, [HL+C]	2	1	4	$A \leftarrow A \vee (\text{HL} + \text{C})$	×
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + \text{C})$	×
	XOR	A, #byte	2	1	—	$A \leftarrow A \oplus \text{byte}$	×
		saddr, #byte	3	2	—	$(\text{saddr}) \leftarrow (\text{saddr}) \oplus \text{byte}$	×
		A, $r^{\#3}$	2	1	—	$A \leftarrow A \oplus r$	×
		r, A	2	1	—	$r \leftarrow r \oplus A$	×
		A, laddr16	3	1	4	$A \leftarrow A \oplus (\text{addr16})$	×
		A, ES:laddr16	4	2	5	$A \leftarrow A \oplus (\text{ES:addr16})$	×
		A, saddr	2	1	—	$A \leftarrow A \oplus (\text{saddr})$	×
		A, [HL]	1	1	4	$A \leftarrow A \oplus (\text{HL})$	×
		A, ES:[HL]	2	2	5	$A \leftarrow A \oplus (\text{ES:HL})$	×
		A, [HL+byte]	2	1	4	$A \leftarrow A \oplus (\text{HL} + \text{byte})$	×
		A, ES:[HL+byte]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + \text{byte})$	×
		A, [HL+B]	2	1	4	$A \leftarrow A \oplus (\text{HL} + \text{B})$	×
		A, ES:[HL+B]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + \text{B})$	×
		A, [HL+C]	2	1	4	$A \leftarrow A \oplus (\text{HL} + \text{C})$	×
		A, ES:[HL+C]	3	2	5	$A \leftarrow A \oplus ((\text{ES:HL}) + \text{C})$	×

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(f_{CLK})。

2. 程序存储器区域被存取时的 CPU 时钟数(f_{CLK})。

3. 除 r = A 以外。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(10/17)

指令组	助记符	操作数	字节	时钟		时钟	标志		
				注1	注2		Z	AC	CY
8 位操作	CMP	A, #byte	2	1	–	A – byte	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r ^{注3}	2	1	–	A – r	×	×	×
		r, A	2	1	–	r – A	×	×	×
		A, !addr16	3	1	4	A – (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A – (ES:addr16)	×	×	×
		A, saddr	2	1	–	A – (saddr)	×	×	×
		A, [HL]	1	1	4	A – (HL)	×	×	×
		A, ES:[HL]	2	2	5	A – (ES:HL)	×	×	×
		A, [HL+byte]	2	1	4	A – (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A – ((ES:HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A – (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A – ((ES:HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A – (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A – ((ES:HL)+C)	×	×	×
	CMP0	A	1	1	–	A – 00H	×	0	0
		X	1	1	–	X – 00H	×	0	0
		B	1	1	–	B – 00H	×	0	0
		C	1	1	–	C – 00H	×	0	0
		!addr16	3	1	4	(addr16) – 00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16) – 00H	×	0	0
		saddr	2	1	–	(saddr) – 00H	×	0	0
	CMPS	X, [HL+byte]	3	1	4	X – (HL+byte)	×	×	×
		X, ES:[HL+byte]	4	2	5	X – ((ES:HL)+byte)	×	×	×

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(f_{CLK})。

2. 程序存储器区域被存取时的 CPU 时钟数(f_{CLK})。

3. 除 r = A 以外。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(11/17)

指令组	助记符	操作数	字节	时钟		时钟	标志		
				注1	注2		Z	AC	CY
16 位 操作	ADDW	AX, #word	3	2	—	AX, CY ← AX+word	×	×	×
		AX, AX	1	2	—	AX, CY ← AX+AX	×	×	×
		AX, BC	1	2	—	AX, CY ← AX+BC	×	×	×
		AX, DE	1	2	—	AX, CY ← AX+DE	×	×	×
		AX, HL	1	2	—	AX, CY ← AX+HL	×	×	×
		AX, !addr16	3	2	5	AX, CY ← AX+(addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX, CY ← AX+(ES:addr16)	×	×	×
		AX, saddrp	2	2	—	AX, CY ← AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX, CY ← AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	3	6	AX, CY ← AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	2	—	AX, CY ← AX – word	×	×	×
		AX, BC	1	2	—	AX, CY ← AX – BC	×	×	×
		AX, DE	1	2	—	AX, CY ← AX – DE	×	×	×
		AX, HL	1	2	—	AX, CY ← AX – HL	×	×	×
		AX, !addr16	3	2	5	AX, CY ← AX – (addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, saddrp	2	2	—	AX, CY ← AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX, CY ← AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	3	6	AX, CY ← AX – ((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	2	—	AX – word	×	×	×
		AX, BC	1	2	—	AX – BC	×	×	×
		AX, DE	1	2	—	AX – DE	×	×	×
		AX, HL	1	2	—	AX – HL	×	×	×
		AX, !addr16	3	2	5	AX – (addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX – (ES:addr16)	×	×	×
		AX, saddrp	2	2	—	AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	3	6	AX – ((ES:HL)+byte)	×	×	×
乘	MULU	X	1	2	—	AX ← A×X			

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(f_{CLK})。

2. 程序存储器区域被存取时的 CPU 时钟数(f_{CLK})。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(12/17)

指令组	助记符	操作数	字节	时钟		时钟	标志		
				注1	注2		Z	AC	CY
递增/递减	INC	r	1	1	—	$r \leftarrow r+1$	×	×	
		laddr16	3	2	—	$(addr16) \leftarrow (addr16)+1$	×	×	
		ES:laddr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)+1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr)+1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)+1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)+1$	×	×	
	DEC	r	1	1	—	$r \leftarrow r-1$	×	×	
		laddr16	3	2	—	$(addr16) \leftarrow (addr16)-1$	×	×	
		ES:laddr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16)-1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr)-1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte)-1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)-1$	×	×	
	INCW	rp	1	2	—	$rp \leftarrow rp+1$			
		laddr16	3	4	—	$(addr16) \leftarrow (addr16)+1$			
		ES:laddr16	4	5	—	$(ES, addr16) \leftarrow (ES, addr16)+1$			
		saddrp	2	4	—	$(saddrp) \leftarrow (saddrp)+1$			
		[HL+byte]	3	4	—	$(HL+byte) \leftarrow (HL+byte)+1$			
		ES: [HL+byte]	4	5	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)+1$			
	DECW	rp	1	2	—	$rp \leftarrow rp-1$			
		laddr16	3	4	—	$(addr16) \leftarrow (addr16)-1$			
		ES:laddr16	4	5	—	$(ES, addr16) \leftarrow (ES, addr16)-1$			
		saddrp	2	4	—	$(saddrp) \leftarrow (saddrp)-1$			
		[HL+byte]	3	4	—	$(HL+byte) \leftarrow (HL+byte)-1$			
		ES: [HL+byte]	4	5	—	$((ES:HL)+byte) \leftarrow ((ES:HL)+byte)-1$			
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	2	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow 0) \times cnt$			×
	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	2	—	$(CY \leftarrow AX_{15}, AX_m \leftarrow AX_{m-1}, AX_0 \leftarrow 0) \times cnt$			×
		BC, cnt	2	2	—	$(CY \leftarrow BC_{15}, BC_m \leftarrow BC_{m-1}, BC_0 \leftarrow 0) \times cnt$			×
	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	2	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow AX_{15}) \times cnt$			×

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(fclk)。

2. 程序存储器区域被存取时的 CPU 时钟数(fclk)。

备注 1. 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

2. cnt 为移位位数。

表20-5. 操作列表(13/17)

指令组	助记符	操作数	字节	时钟		时钟	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	2	—	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			×
		BC, 1	2	2	—	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			×
位操作	MOV1	CY, A.bit	2	1	—	$CY \leftarrow A.bit$			×
		A.bit, CY	2	1	—	$A.bit \leftarrow CY$			
		CY, PSW.bit	3	1	—	$CY \leftarrow PSW.bit$			×
		PSW.bit, CY	3	4	—	$PSW.bit \leftarrow CY$	×	×	
		CY, saddr.bit	3	1	—	$CY \leftarrow (saddr).bit$			×
		saddr.bit, CY	3	2	—	$(saddr).bit \leftarrow CY$			
		CY, sfr.bit	3	1	—	$CY \leftarrow sfr.bit$			×
		sfr.bit, CY	3	2	—	$sfr.bit \leftarrow CY$			
		CY, [HL].bit	2	1	4	$CY \leftarrow (HL).bit$			×
		[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, A.bit	2	1	—	$CY \leftarrow CY \wedge A.bit$			×
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \wedge PSW.bit$			×
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \wedge (saddr).bit$			×
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \wedge sfr.bit$			×
		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, A.bit	2	1	—	$CY \leftarrow CY \vee A.bit$			×
		CY, PSW.bit	3	1	—	$CY \leftarrow CY \vee PSW.bit$			×
		CY, saddr.bit	3	1	—	$CY \leftarrow CY \vee (saddr).bit$			×
		CY, sfr.bit	3	1	—	$CY \leftarrow CY \vee sfr.bit$			×
		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. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(fclk)。

2. 程序存储器区域被存取时的 CPU 时钟数(fclk)。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(14/17)

指令组	助记符	操作数	字节	时钟		时钟	标志		
				注1	注2		Z	AC	CY
位操作	XOR1	CY, A.bit	2	1	–	$CY \leftarrow CY \oplus A.bit$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \oplus PSW.bit$			×
		CY, saddr.bit	3	1	–	$CY \leftarrow CY \oplus (saddr).bit$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \oplus sfr.bit$			×
		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	A.bit	2	1	–	$A.bit \leftarrow 1$			
		PSW.bit	3	4	–	$PSW.bit \leftarrow 1$	×	×	×
		!addr16.bit	4	2	–	$(addr16).bit \leftarrow 1$			
		ES:!addr16.bit	5	3	–	$(ES, addr16).bit \leftarrow 1$			
		saddr.bit	3	2	–	$(saddr).bit \leftarrow 1$			
		sfr.bit	3	2	–	$sfr.bit \leftarrow 1$			
		[HL].bit	2	2	–	$(HL).bit \leftarrow 1$			
		ES:[HL].bit	3	3	–	$(ES, HL).bit \leftarrow 1$			
	CLR1	A.bit	2	1	–	$A.bit \leftarrow 0$			
		PSW.bit	3	4	–	$PSW.bit \leftarrow 0$	×	×	×
		!addr16.bit	4	2	–	$(addr16).bit \leftarrow 0$			
		ES:!addr16.bit	5	3	–	$(ES, addr16).bit \leftarrow 0$			
		saddr.bit	3	2	–	$(saddr).bit \leftarrow 0$			
		sfr.bit	3	2	–	$sfr.bit \leftarrow 0$			
		[HL].bit	2	2	–	$(HL).bit \leftarrow 0$			
		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	–	$CY \leftarrow \neg CY$			×

注 1. 内部 RAM 区域、SFR 区域或扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(f_{CLK})。

2. 程序存储器区域被存取时的 CPU 时钟数(f_{CLK})。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(15/17)

指令组	助记符	操作数	字节	时钟		时钟	标志
				注1	注2		Z AC CY
调用 / 返回	CALL	rp	2	4	—	$(SP-2) \leftarrow (PC+2)_S, (SP-3) \leftarrow (PC+2)_H,$ $(SP-4) \leftarrow (PC+2)_L, PC \leftarrow CS, rp,$ $SP \leftarrow SP-4$	
		\$!addr20	3	4	—	$(SP-2) \leftarrow (PC+3)_S, (SP-3) \leftarrow (PC+3)_H,$ $(SP-4) \leftarrow (PC+3)_L, PC \leftarrow PC+3+jdisp16,$ $SP \leftarrow SP-4$	
		!addr16	3	4	—	$(SP-2) \leftarrow (PC+3)_S, (SP-3) \leftarrow (PC+3)_H,$ $(SP-4) \leftarrow (PC+3)_L, PC \leftarrow 0000, addr16,$ $SP \leftarrow SP-4$	
		!!addr20	4	4	—	$(SP-2) \leftarrow (PC+4)_S, (SP-3) \leftarrow (PC+4)_H,$ $(SP-4) \leftarrow (PC+4)_L, PC \leftarrow addr20,$ $SP \leftarrow SP-4$	
	CALLT	[addr5]	2	6	—	$(SP-2) \leftarrow (PC+2)_S, (SP-3) \leftarrow (PC+2)_H,$ $(SP-4) \leftarrow (PC+2)_L, PC_S \leftarrow 0000,$ $PC_H \leftarrow (0000, addr5+1),$ $PC_L \leftarrow (0000, addr5),$ $SP \leftarrow SP-4$	
	BRK	-	2	7	—	$(SP-1) \leftarrow PSW, (SP-2) \leftarrow (PC+2)_S,$ $(SP-3) \leftarrow (PC+2)_H, (SP-4) \leftarrow (PC+2)_L,$ $PC_S \leftarrow 0000,$ $PC_H \leftarrow (0007FH), PC_L \leftarrow (0007EH),$ $SP \leftarrow SP-4, IE \leftarrow 0$	
	RET	-	1	7	—	$PC_L \leftarrow (SP), PC_H \leftarrow (SP+1),$ $PC_S \leftarrow (SP+2), SP \leftarrow SP+4$	
	RETI	-	2	8	—	$PC_L \leftarrow (SP), PC_H \leftarrow (SP+1),$ $PC_S \leftarrow (SP+2), PSW \leftarrow (SP+3),$ $SP \leftarrow SP+4$	R R R
	RETB	-	2	8	—	$PC_L \leftarrow (SP), PC_H \leftarrow (SP+1),$ $PC_S \leftarrow (SP+2), PSW \leftarrow (SP+3),$ $SP \leftarrow SP+4$	R R R

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(f_{CLK})。

2. 程序存储器区域被存取时的 CPU 时钟数(f_{CLK})。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(16/17)

指令组	助记符	操作数	字节	时钟		时钟	标志
				注1	注2		Z AC CY
堆栈操作	PUSH	PSW	2	2	—	(SP - 1) ← PSW, (SP - 2) ← 00H, SP ← SP - 2	
		rp	1	2	—	(SP - 1) ← rp _H , (SP - 2) ← rp _L , SP ← SP - 2	
	POP	PSW	2	4	—	PSW ← (SP+1), SP ← SP + 2	R R R
		rp	1	2	—	rp _L ← (SP), rp _H ← (SP+1), SP ← SP + 2	
	MOVW	SP, #word	4	2	—	SP ← word	
		SP, AX	2	2	—	SP ← AX	
		AX, SP	2	2	—	AX ← SP	
		HL, SP	3	2	—	HL ← SP	
		BC, SP	3	2	—	BC ← SP	
		DE, SP	3	2	—	DE ← SP	
	ADDW	SP, #byte	2	2	—	SP ← SP + byte	
	SUBW	SP, #byte	2	2	—	SP ← SP - byte	
无条件转移	BR	AX	2	3	—	PC ← CS, AX	
		\$addr20	2	3	—	PC ← PC + 2 + jdisp8	
		!addr20	3	3	—	PC ← PC + 3 + jdisp16	
		!addr16	3	3	—	PC ← 0000, addr16	
		!!addr20	4	3	—	PC ← addr20	
条件转移	BC	\$addr20	2	2/4 ^{注3}	—	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	\$addr20	2	2/4 ^{注3}	—	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	\$addr20	2	2/4 ^{注3}	—	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	\$addr20	2	2/4 ^{注3}	—	PC ← PC + 2 + jdisp8 if Z = 0	
	BH	\$addr20	3	2/4 ^{注3}	—	PC ← PC + 3 + jdisp8 if (Z∨CY)=0	
	BNH	\$addr20	3	2/4 ^{注3}	—	PC ← PC + 3 + jdisp8 if (Z∨CY)=1	
	BT	saddr.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1	
		sfr.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, \$addr20	3	3/5 ^{注3}	—	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1	
		[HL].bit, \$addr20	3	3/5 ^{注3}	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
		ES:[HL].bit, \$addr20	4	4/6 ^{注3}	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1	

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(f_{CLK})。

2. 程序存储器区域被存取时的 CPU 时钟数(f_{CLK})。

3. 表示“当不符合条件/符合条件时”的时钟数。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

表20-5. 操作列表(17/17)

指令组	助记符	操作数	字节	时钟		时钟	标志
				注1	注2		Z AC CY
有条件分支	BF	saddr.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 0	
		sfr.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, \$addr20	3	3/5 ^{注3}	—	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 if PSW.bit = 0	
		[HL].bit, \$addr20	3	3/5 ^{注3}	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
		ES:[HL].bit, \$addr20	4	4/6 ^{注3}	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0	
	BTCLR	saddr.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit	
		sfr.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, \$addr20	3	3/5 ^{注3}	—	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
		PSW.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, \$addr20	3	3/5 ^{注3}	—	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
		ES:[HL].bit, \$addr20	4	4/6 ^{注3}	—	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit	
有条件跳跃	SKC	—	2	1	—	Next instruction skip if CY = 1	
	SKNC	—	2	1	—	Next instruction skip if CY = 0	
	SKZ	—	2	1	—	Next instruction skip if Z = 1	
	SKNZ	—	2	1	—	Next instruction skip if Z = 0	
	SKH	—	2	1	—	Next instruction skip if (Z∨CY)=0	
	SKNH	—	2	1	—	Next instruction skip if (Z∨CY)=1	
CP 控制	NOP	—	1	1	—	No Operation	
	EI	—	3	4	—	IE ← 1 (Enable Interrupt)	
	DI	—	3	4	—	IE ← 0 (Disable Interrupt)	
	HALT	—	2	3	—	Set HALT Mode	
	STOP	—	2	3	—	Set STOP Mode	

注 1. 内部 RAM 区域、SFR 区域及扩展 SFR 区域被存取，或者未存取数据时的 CPU 时钟数(f_{CLK})。

2. 程序存储器区域被存取时的 CPU 时钟数(f_{CLK})。

3. 表示“当不符合条件/符合条件时”的时钟数。

备注 时钟数是指程序存储于 ROM（闪存）区域时的时钟数。如果从内部 RAM 区域获取指令，最大为该值的 2 倍加 3 个时钟。

第二十一章 电特性

- 注意事项 1.** R7F0C80112ESP, R7F0C80212ESP 提供用于开发和评估的片上调试功能。对于量产产品,不得使用片上调试功能,因为在使用该功能时,可能会超过闪存保证的可重写次数,因而会使产品可靠性无法保证。对于使用片上调试功能时发生的问题,瑞萨电子不承担任何责任。
- 2.** 由于包括了可选择上电复位(SPOR)电路的检测电压(V_{SPOR}),请在 2.57(TYP.)至 5.5 V 的电压范围内使用 R7F0C80112ESP, R7F0C80212ESP。

21.1 最大绝对额定值

(T_A = 25°C)

参数	符号	条件		额定值	单位
电源电压	V _{DD}			-0.5 至+6.5	V
输入电压	V _{I1}			-0.3 至 V _{DD} + 0.3 [※]	V
输出电压	V _{O1}			-0.3 至 V _{DD} + 0.3	V
高电位输出电流	I _{OH1}	单个引脚		-40	mA
		所有引脚共计 -140 mA	P40	-40	mA
			P00 至 P04	-100	mA
低电位输出电流	I _{OL1}	单个引脚		40	mA
		所有引脚共计 140 mA	P40	40	mA
			P00 至 P04	100	mA
运行环境温度	T _A			-40 至+85	°C
存储温度	T _{stg}			-65 至+150	°C

注 必须为 6.5 V 或更低。

注意事项 任何参数如果超过最大额定值,即使只有一瞬间超过也可能损坏产品质量。也就是说,最大绝对额定值是对产品造成物理损坏的临界值,因此必须确保在不超过最大绝对额定值下使用产品。

- 备注 1.** 除非另有说明,否则复用功能引脚的特性与端口引脚的特性相同。
- 2.** 基准电压为 V_{SS}。

21.2 振荡电路特性

21.2.1 片上振荡器特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$)

振荡电路	参数	条件	MIN.	TYP.	MAX.	单位
高速片上振荡器振荡频率 ^{註1,2}	f_{IH}		1.25		20	MHz
高速片上振荡器振荡频率精度		$T_A = -40$ 至 $+85^{\circ}\text{C}$	-3		+3	%
		$T_A = 0$ 至 $+40^{\circ}\text{C}$	-2		+2	%
低速片上振荡器振荡频率 ^{註3}	f_{IL}			15		kHz

注 1. 可通过选项字节(000C2H)的位 0 至 2 来选择高速片上振荡器的频率。

2. 仅表示振荡电路特性。关于指令执行时间, 请参阅 AC 特性。

3. 仅表示振荡电路的特性。

21.3 DC 特性

21.3.1 引脚特性

(TA = -40 至+85°C, 2.4 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

参数	符号	条件			MIN.	TYP.	MAX.	单位
高电位输出电流	I _{OH1}	P00 ^{註1} , P01, P02 至P04, P40	单个引脚				-10.0 ^{註3}	mA
		P40	总计 ^{註2}	4.0 V ≤ V _{DD} ≤ 5.5 V			-10.0	mA
				2.7 V ≤ V _{DD} < 4.0 V			-2.0	mA
				2.4 V ≤ V _{DD} < 2.7 V			-1.5	mA
		P00 ^{註1} , P01, P02 至P04	总计 ^{註2}	4.0 V ≤ V _{DD} ≤ 5.5 V			-50.0	mA
				2.7 V ≤ V _{DD} < 4.0 V			-10.0	mA
				2.4 V ≤ V _{DD} < 2.7 V			-7.5	mA
		引脚总计 ^{註2}						-60.0
低电位输出电流	I _{OL1}	P00 至 P04, P40	单个引脚				20.0 ^{註3}	mA
		P40	总计 ^{註2}	4.0 V ≤ V _{DD} ≤ 5.5 V			20.0	mA
				2.7 V ≤ V _{DD} < 4.0 V			3.0	mA
				2.4 V ≤ V _{DD} < 2.7 V			0.6	mA
		P00 至 P04	总计 ^{註2}	4.0 V ≤ V _{DD} ≤ 5.5 V			80.0	mA
				2.7 V ≤ V _{DD} < 4.0 V			12.0	mA
				2.4 V ≤ V _{DD} < 2.7 V			2.4	mA
		引脚总计 ^{註2}						100.0
高电位输入电压	V _{IH1}				0.8 V _{DD}		V _{DD} ^{註4}	V
低电位输入电压	V _{IL1}				0		0.2 V _{DD}	V
高电位输出电压	V _{OH1}	4.0 V ≤ V _{DD} ≤ 5.5 V		I _{OH} = - 10 mA	V _{DD} -1.5			V
				I _{OH} = - 3.0 mA	V _{DD} -0.7			V
		2.7 V ≤ V _{DD} ≤ 5.5 V		I _{OH} = - 2.0 mA	V _{DD} -0.6			V
				2.4 V ≤ V _{DD} ≤ 5.5 V		I _{OH} = - 1.5 mA	V _{DD} -0.5	
低电位输出电压	V _{OL1}	4.0 V ≤ V _{DD} ≤ 5.5 V		I _{OH} = 20 mA			1.3	V
				I _{OH} = 8.5 mA			0.7	V
		2.7 V ≤ V _{DD} ≤ 5.5 V		I _{OH} = 3.0 mA			0.6	V
				I _{OH} = 1.5 mA			0.4	V
		2.4 V ≤ V _{DD} ≤ 5.5 V		I _{OH} = 0.6 mA			0.4	V
高电位输入漏电流	I _{LIH1}	V _I = V _{DD}					1	μA
低电位输入漏电流	I _{LIL1}	V _I = V _{SS}					-1	μA
内置上拉电阻	RU	V _I = V _{SS}			10	20	100	KΩ

注 1. N 沟道开漏输出 (VDD 耐压) 模式时, 不输出高电位。

2. 满足占空比 ≤ 70% 条件时的输出电流值。可以通过以下的公式算出占空比 > 70% 时的输出电流值 (占空比改变为 n % 时)。

- 引脚总计输出电流 = (IOH × 0.7) / (n × 0.01)

<计算例> 当 IOH = - 10.0 mA 时, n = 80 %

引脚总计输出电流 = (- 10.0 × 0.7) / (80 × 0.01) ≈ - 8.7 mA

- 引脚总计输出电流 = $(I_{OL} \times 0.7) / (n \times 0.01)$

<计算例>当 $I_{OL} = 10.0 \text{ mA}$ 时, $n = 80 \%$

引脚总计输出电流 = $(10.0 \times 0.7) / (80 \times 0.01) \approx 8.7 \text{ mA}$

1 个引脚中能够通过的电流不随占空比而变化。此外, 只能通过低于绝对最大额定值的电流。

3. 请勿超过总计电流值。
4. 即使在 N 沟道开漏输出 (V_{DD} 耐压) 模式下, V_{IH} 的最大值也为 V_{DD} 。

注意事项 1. P00 在 N 沟开漏模式下不输出高电平。

2. 即使 P00 在 N 沟开漏模式下, V_{IH} 的最大值(MAX.)为 V_{DD} 。

备注 特别指出的情况除外, 复用引脚和端口引脚具有相同的特性。

21.3.2 电源电流特性

(T_A = -40 至 +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

参数	符号	条件				MIN.	TYP.	MAX.	单位
电源电流 ^{註1}	I _{DD1}	操作 模式	基本操作	f _{IH} = 20 MHz	V _{DD} = 3.0V / 5.0V		0.91		mA
			通常操作	f _{IH} = 20 MHz	V _{DD} = 3.0V / 5.0V		1.57	2.04	
				f _{IH} = 5 MHz	V _{DD} = 3.0V / 5.0V		0.85	1.15	
	I _{DD2} ^{註2}	HALT	f _{IH} = 20 MHz	V _{DD} = 3.0V / 5.0V		350	820	μA	
			f _{IH} = 5 MHz	V _{DD} = 3.0V / 5.0V		290	600		
	I _{DD3} ^{註3}	STOP	V _{DD} = 3V				0.56	2.00	μA
	WDT电源电流 ^{註4}	I _{WDT}	f _{IL} = 15 kHz					0.31	
ADC电源电流 ^{註5}	I _{ADC}	最快转换时		V _{DD} = 5.0 V			1.30	1.90	mA
				V _{DD} = 3.0 V			0.50		

- 注 1. 流入 V_{DD} 的总电流，包括输入引脚电位固定至 V_{DD} 或 V_{SS} 时的输入漏电流。MAX.栏下方的值包括外围操作电流。但是，不包括流过 A/D 转换器、输入/输出端口和片上上拉/下拉电阻的电流。
2. 通过内存执行 HALT 指令时。
3. 当看门狗定时器和 A/D 转换器停止时。MAX.栏下方的值包括漏电流。
4. 仅流入看门狗定时器的电流（包括 15 kHz 低速片上振荡器的操作电流）。当看门狗定时器工作时，RL78 单片机的电源电流值为 I_{DD1}、I_{DD2} 或 I_{DD3} 与 I_{WDT} 之和。
5. 仅流入 A/D 转换器的电流。当 A/D 转换器工作于操作模式或 HALT 模式时，RL78 单片机的电流值为 I_{DD1} 或 I_{DD2} 与 I_{ADC} 之和。

- 备注 1. f_{IL}: 低速片上振荡器时钟频率
2. f_{IH}: 高速片上振荡器时钟频率
3. TYP.值的温度条件是 T_A = 25 °C。

21.4 AC 特性

(TA = -40 至+85°C, 2.4 V ≤ VDD ≤ 5.5 V, VSS = 0 V)

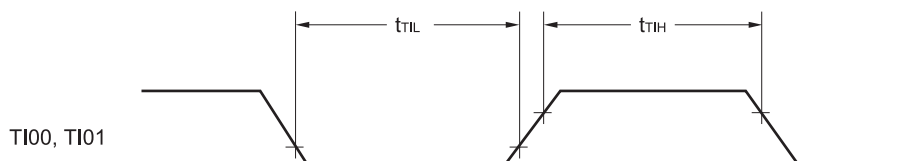
项目	符号	条件	MIN.	TYP.	MAX.	单位
指令周期(最短指令执行时间)	T _{CY}	主系统时钟 (f _{MAIN})操作	2.7 V ≤ V _{DD} ≤ 5.5 V	0.05		μs
			2.4 V ≤ V _{DD} ≤ 5.5 V	0.2		μs
TI00, TI01 输入高电位, 低电位宽度	t _{TIH} , t _{TIL}	Noise filter is not used	1/f _{MCK} +10			ns
TO00, TO01 输出频率	f _{TO}	4.0 V ≤ V _{DD} ≤ 5.5 V			10	MHz
		2.7 V ≤ V _{DD} < 4.0 V			5	MHz
		2.4 V ≤ V _{DD} < 2.7 V			2.5	MHz
PCLBUZ0 输出频率	f _{PCL}	4.0 V ≤ V _{DD} ≤ 5.5 V			10	MHz
		2.7 V ≤ V _{DD} < 4.0 V			5	MHz
		2.4 V ≤ V _{DD} < 2.7 V			2.5	MHz
RESET 低电位宽度	t _{RSL}		10			μs

备注 f_{MCK}: 定时器阵列单元的工作时钟频率。

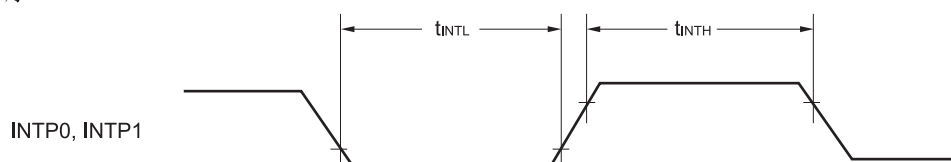
AC时序检测点



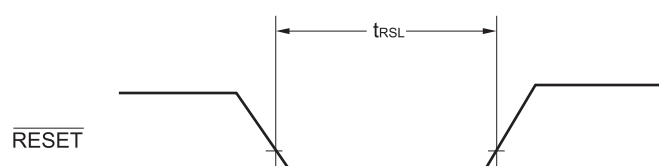
TI/TO时序



中断请求输入时序



RESET输入时序



21.5 串行接口特性

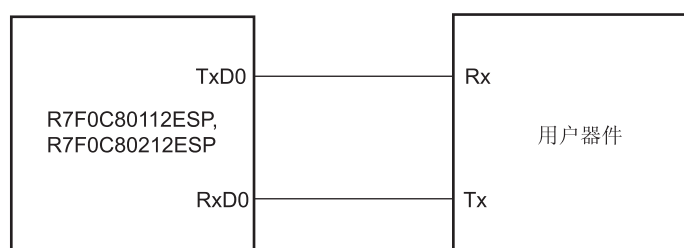
21.5.1 串行阵列单元

(1) UART 模式 (专用波特率生成器输出)

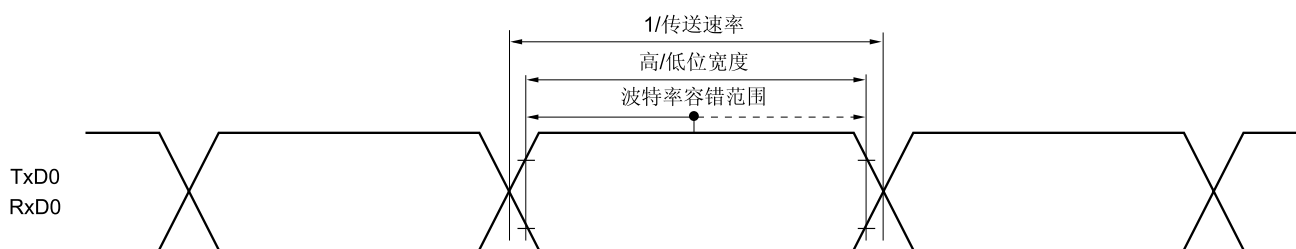
($T_A = -40$ 至 $+85^{\circ}\text{C}$, $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
传送速率					$f_{MCK}/6$	bps
		最大传送速率的理论值 $f_{CLK} = f_{MCK} = 20\text{ MHz}$			3.3	Mbps

UART 模式连接图



UART 模式时的位宽(参考)



注意事项 使用端口输出模式寄存器 0 (POM0)将 TxD0 引脚设置为普通输出模式。

备注 f_{MCK} : 串行阵列单元的工作时钟频率。

(由串行模式寄存器 mn (SMR00)的 CKSmn 位设置的工作时钟。m: 单元编号, n: 通道编号(mn = 00))

(2) CSI 模式（主模式，SCKp...内部时钟输出）

(T_A = -40 至 +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
$\overline{\text{SCKp}}$ 周期	t _{KCY1}	2.7 V ≤ V _{DD} ≤ 5.5 V	200 ^{※1}			ns
		2.4 V ≤ V _{DD} ≤ 5.5 V	800 ^{※1}			ns
$\overline{\text{SCKp}}$ 高电位/低电位宽度	t _{KH1} , t _{KL1}	2.7 V ≤ V _{DD} ≤ 5.5 V	t _{KCY1} /2 - 18			ns
		2.4 V ≤ V _{DD} ≤ 5.5 V	t _{KCY1} /2 - 50			ns
Slp建立时间 (至 $\overline{\text{SCKp}}\uparrow$) ^{※2}	t _{SIK1}	2.7 V ≤ V _{DD} ≤ 5.5 V	47			ns
		2.4 V ≤ V _{DD} ≤ 5.5 V	110			ns
Slp保持时间 (从 $\overline{\text{SCKp}}\uparrow$) ^{※3}	t _{KSI1}		19			ns
从 $\overline{\text{SCKp}}\downarrow$ 至SO _p 输出的延迟时间 ^{※4}	t _{KSO1}	C = 30 pF ^{※5}			25	ns

注 1. 需设定为 4/f_{CLK} 或更高的值。2. 当 DAP_{mn} = 0 和 CKP_{mn} = 0, 或 DAP_{mn} = 1 和 CKP_{mn} = 1 时。DAP_{mn} = 0 和 CKP_{mn} = 1, 或 DAP_{mn} = 1 和 CKP_{mn} = 0 时, Slp 建立时间变为至 $\overline{\text{SCKp}}\downarrow$ 。3. 当 DAP_{mn} = 0 和 CKP_{mn} = 0, 或 DAP_{mn} = 1 和 CKP_{mn} = 1 时。DAP_{mn} = 0 和 CKP_{mn} = 1, 或 DAP_{mn} = 1 和 CKP_{mn} = 0 时, Slp 保持时间变为至 $\overline{\text{SCKp}}\downarrow$ 。4. 当 DAP_{mn} = 0 和 CKP_{mn} = 0, 或 DAP_{mn} = 1 和 CKP_{mn} = 1 时。DAP_{mn} = 0 和 CKP_{mn} = 1, 或 DAP_{mn} = 1 和 CKP_{mn} = 0 时, SO_p 输出的延迟时间变为从 $\overline{\text{SCKp}}\uparrow$ 。5. C 为 $\overline{\text{SCKp}}$ 和 SO_p 输出线路的负载电容。注意事项 使用端口输出模式寄存器 0 (POM0)将 SO_p 引脚和 $\overline{\text{SCKp}}$ 引脚设置为普通输出模式。

备注 p: CSI 编号(p = 00), m: 单元编号(m = 0), n: 通道编号(n = 0)

(3) CSI 模式 (从模式, SCKp...外部时钟输入)

(T_A = -40 至 +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

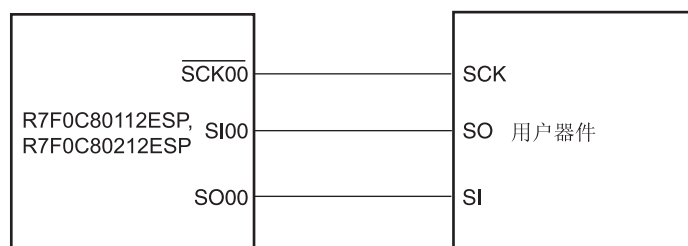
参数	符号	条件	MIN.	TYP.	MAX.	单位
SCKp 周期	t _{KCY2}	2.7 V ≤ V _{DD} ≤ 5.5 V	f _{MCK} = 20 MHz	8/f _{MCK}		ns
			f _{MCK} ≤ 20 MHz	6/f _{MCK}		ns
		2.4 V ≤ V _{DD} < 2.7 V		6/f _{MCK}		ns
SCKp 高电位/低电位宽度	t _{KH2} , t _{KL2}	2.4 V ≤ V _{DD} ≤ 5.5 V	t _{KCY} /2			ns
Slp建立时间 (至SCKp↑) ^{※1}	t _{SIK2}	2.7 V ≤ V _{DD} ≤ 5.5 V	1/f _{MCK} +20			ns
		2.4 V ≤ V _{DD} < 2.7 V	1/f _{MCK} +30			ns
Slp保持时间 (从SCKp↑) ^{※1}	t _{KSI2}	2.4 V ≤ V _{DD} ≤ 5.5 V	1/f _{MCK} +31			ns
从SCKp↓至SOp输出的延迟时间 ^{※2}	t _{KSO2}	C = 30 pF ^{※3}	2.7 V ≤ V _{DD} ≤ 5.5 V		2/f _{MCK} +50	ns
			2.4 V ≤ V _{DD} < 2.7 V		2/f _{MCK} +110	ns

- 注 1. 当 DAPmn = 0 和 CKPmn = 0, 或 DAPmn = 1 和 CKPmn = 1 时。DAPmn = 0 和 CKPmn = 1, 或 DAPmn = 1 和 CKPmn = 0 时, Slp 建立时间变为至 SCKp↓。
2. 当 DAPmn = 0 和 CKPmn = 0, 或 DAPmn = 1 和 CKPmn = 1 时。DAPmn = 0 和 CKPmn = 1, 或 DAPmn = 1 和 CKPmn = 0 时, Slp 保持时间变为从 SCKp↓。
3. 当 DAPmn = 0 和 CKPmn = 0, 或 DAPmn = 1 和 CKPmn = 1 时。DAPmn = 0 和 CKPmn = 1, 或 DAPmn = 1 和 CKPmn = 0 时, SOp 输出的延迟时间变为从 SCKp↑。
4. C 为 SOp 输出线路的负载电容。

注意事项 使用端口输出模式寄存器 0 (POM0)将 SOp 引脚设置为普通输出模式。

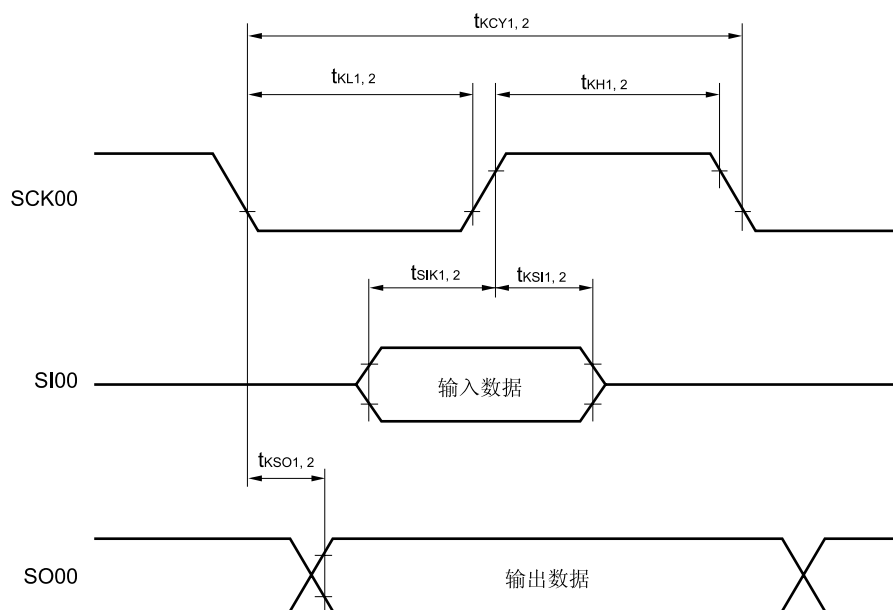
- 备注 1. p: CSI 编号(p = 00), m: 单元编号(m = 0), n: 通道编号(n = 0)
2. f_{MCK}: 串行阵列单元的工作时钟频率。
(由串行模式寄存器 mn (SMRmn)的 CKSmn 位设置的操作时钟。m: 单元编号, n: 通道编号(mn = 00))

CSI 模式连接框图



CSI 模式串行传送时序

($\text{DAP00} = 0$ 和 $\text{CKP00} = 0$, 或 $\text{DAP00} = 1$ 和 $\text{CKP00} = 1$ 时。)



21.6 模拟特性

21.6.1 A/D 转换器特性

(目标 ANI 引脚: ANI0 至 ANI3)

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件		MIN.	TYP.	MAX.	单位
分辨率	RES			8		10	bit
总误差 ^{注1}	AINL	10 位分辨率	$V_{DD} = 5\text{ V}$		± 1.7	$\pm 3.1^{\text{注2}}$	LSB
			$V_{DD} = 3\text{ V}$		± 2.3	$\pm 4.5^{\text{注2}}$	LSB
转换时间	t_{CONV}		$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	3.4		18.4	μs
			$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	4.6		18.4	μs
零刻度误差 ^{注1}	E_{ZS}	10 位分辨率	$V_{DD} = 5\text{ V}$			$\pm 0.19^{\text{注2}}$	%FSR
			$V_{DD} = 3\text{ V}$			$\pm 0.39^{\text{注2}}$	%FSR
满刻度误差 ^{注1}	E_{FS}	10 位分辨率	$V_{DD} = 5\text{ V}$			$\pm 0.29^{\text{注2}}$	%FSR
			$V_{DD} = 3\text{ V}$			$\pm 0.42^{\text{注2}}$	%FSR
积分线性误差 ^{注1}	ILE	10 位分辨率	$V_{DD} = 5\text{ V}$			$\pm 1.8^{\text{注2}}$	LSB
			$V_{DD} = 3\text{ V}$			$\pm 1.7^{\text{注2}}$	LSB
微分线性误差 ^{注1}	DLE	10 位分辨率	$V_{DD} = 5\text{ V}$			$\pm 1.4^{\text{注2}}$	LSB
			$V_{DD} = 3\text{ V}$			$\pm 1.5^{\text{注2}}$	LSB
模拟输入电压	V_{AIN}			0		V_{DD}	V

注 1. 不包括量化误差 ($\pm 1/2\text{LSB}$)。

2. MAX.值为正态分布的平均值 $\pm 3\sigma$ 的数值。未做出厂检查。

21.6.2 数据保持电源电压特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $V_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
数据保持电源电压范围	V_{DDDR}		1.9		5.5	V

注意事项 当电源电压不低于数据保持电源电压范围的MIN.值时, 数据将被保持。此外, 即使低于数据保持电源电压范围的MIN.值, 也可能出现RAM和RESF寄存器中的数据没有被清除的情况。

21.6.3 SPOR 电路特性

(T_A = -40 至 +85°C, V_{SS} = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
检测电源电压	V _{SPOR0}	电源上升时		4.28		V
		电源下降时	4.00			V
	V _{SPOR1}	电源上升时		2.90		V
		电源下降时	2.70			V
	V _{SPOR2}	电源上升时		2.57		V
		电源下降时	2.40			V
最小脉冲宽度*	T _{LSPW}		300			μs

注 当 V_{DD} 小于 V_{SPDR} 时, SPOR 复位操作需要的时间。

21.6.4 电源电压上升倾斜特性

(T_A = -40 至 +85°C, V_{SS} = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
电源电压上升倾斜	S _{VDD}				54	V/ms

21.7 闪存编程特性

(T_A = 0 至 +40°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
代码闪存重写次数 ^{注1, 2, 3}	C _{enwr}	保持20年(T _A = 85°C)	100			次

注 1. “擦除 1 次 + 擦除后写入 1 次”被认为是 1 次重写。保持年限为重写之后到下次重写为止的间隔时间。

2. 当使用闪存编程器和瑞萨电子自编程库时。

3. 该特性是本公司进行信赖性实验所得的结果, 表示闪存的特性。

21.8 专用闪存编程器通信 (UART)

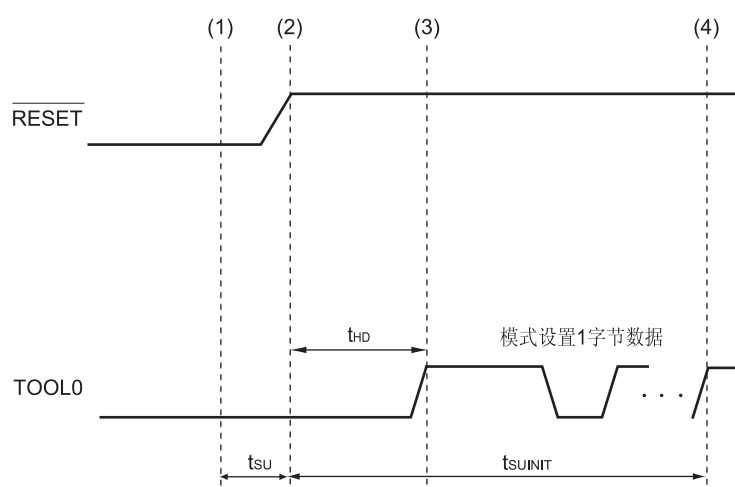
(T_A = -40 至 +85 °C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

项目	符号	条件	MIN.	TYP.	MAX.	单位
传送速率				115,200		bps

备注 闪存编程时的传送速率固定为 115,200 bps。

21.9 进入闪存编程模式的时序

项目	符号	条件	MIN.	TYP.	MAX.	单位
从解除外部复位到完成初始设定通信的时间	t _{SUINIT}	必须在已解除SPOR复位的状态下，解除外部复位。 T _A = 0~40℃			100	ms
从TOOL0引脚被设置为低电平到解除外部复位的时间	t _{SU}	必须在已解除SPOR复位的状态下，解除外部复位。	10			μs
解除外部复位后，TOOL0引脚保持低电平的时间	t _{HD}	必须在已解除SPOR复位的状态下，解除外部复位。	1			ms



- (1) 向 TOOL0 引脚输入低电平
- (2) 解除外部复位（必须先解除 SPOR 复位）
- (3) 解除 TOOL0 引脚的低电平
- (4) 通过执行 UART 接收来设置进入模式

备注 t_{SUINIT}: 在此区间内，解除复位后 100 ms 之内必须完成初始设定通信。

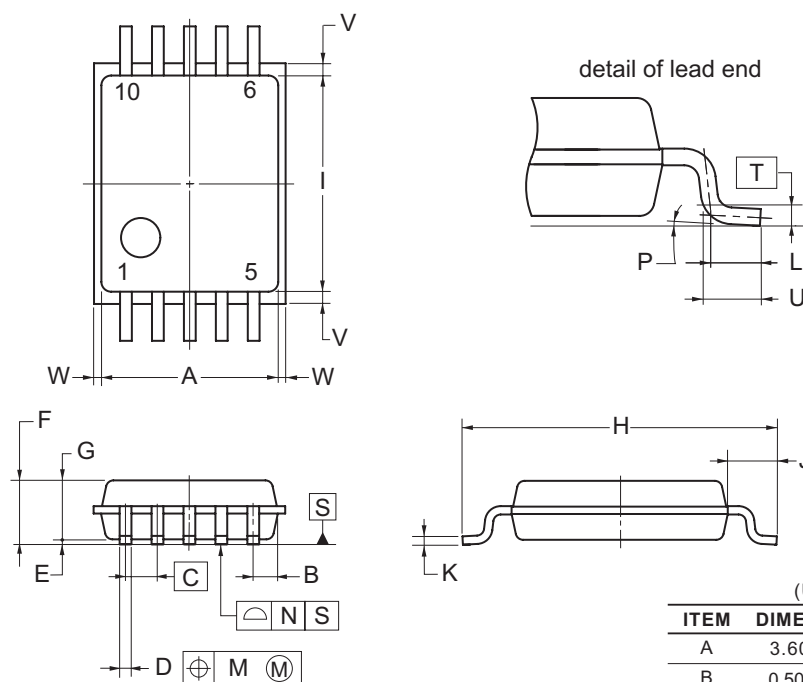
t_{SU}: 从 TOOL0 引脚被设置为低电平到解除外部复位的时间。

t_{HD}: 解除外部复位后，TOOL0 引脚保持低电平的时间。

第二十二章 封装尺寸图

R7F0C80112ESP, R7F0C80212ESP

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP10-4.4x3.6-0.65	PLSP0010JA-A	P10MA-65-CAC-2	0.05

**NOTE**

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

(UNIT:mm)	
ITEM	DIMENSIONS
A	3.60±0.10
B	0.50
C	0.65 (T.P.)
D	0.24±0.08
E	0.10±0.05
F	1.45 MAX.
G	1.20±0.10
H	6.40±0.20
I	4.40±0.10
J	1.00±0.20
K	0.17 ^{+0.08} _{-0.07}
L	0.50
M	0.13
N	0.10
P	3° ^{+5°} _{-3°}
T	0.25 (T.P.)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

©2012 Renesas Electronics Corporation. All rights reserved.

附录 A 修订记录

A.1 修订记录

版本	说明	章
Rev.1.00	初次制作 2013.09.20	

R7F0C80112ESP, R7F0C80212ESP 用户手册 硬件篇

Publication Date: Rev.1.00 Sep 20, 2013

Published by: Renesas Electronics Corporation



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.

2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited

1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.

7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.

Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited

Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.

80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.

11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

R7F0C80112ESP, R7F0C80212ESP