

用户手册

μ PD79F9211

16 位单片机

[备忘录]

① 输入引脚处的电压波形

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

② 未使用的输入引脚的处理

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

③ ESD 防护措施

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

④ 初始化之前的状态

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

⑤ 电源开关顺序

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

对于每个设备电源的正确开关顺序必须依据设备的规范说明分别进行判断。

⑥ 电源关闭状态下的输入信号

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

Windows 和 Windows NT 是 Microsoft Corporation 在美国和/或其他国家的注册商标或商标。
PC/AT 是国际商用机器公司的商标。
EEPROM 是日电电子公司的商标。
SuperFlash 是 Silicon Storage Technology, Inc.在包括美国和日本在内许多国家的注册商标。

注意事项: 本产品使用Silicon Storage Technology, Inc.许可的SuperFlash® 技术。

- 本文档所登载的内容有效期截止至 2010 年 1 月, 信息先于产品的生产周期发布。将来可能未经预先通知而更改。在实际进行生产设计时, 请参阅各产品最新的数据表或数据手册等相关资料以获取本公司产品的最新规格。
- 并非所有的产品和/或型号都向每个国家供应。请向本公司销售代表查询产品供应及其他信息。
- 未经本公司事先书面许可, 禁止复制或转载本文件中的内容。否则因本文档所登载内容引发的错误, 本公司概不负责。
- 本公司对于因使用本文件中列明的本公司产品而引起的, 对第三者的专利、版权以及其它知识产权的侵权行为概不负责。本文件登载的内容不应视为本公司对本公司或其他人所有的专利、版权以及其它知识产权作出任何明示或默示的许可及授权。
- 本文件中的电路、软件以及相关信息仅用以说明半导体产品的运作和应用实例。用户如在设备设计中应用本文件中的电路、软件以及相关信息, 应自行负责。对于用户或其他人因使用了上述电路、软件以及相关信息而引起的任何损失, 本公司概不负责。
- 虽然本公司致力于提高半导体产品的质量及可靠性, 但用户应同意并知晓, 我们仍然无法完全消除出现产品缺陷的可能。为了最大限度地减少因本公司半导体产品故障而引起的对人身、财产造成损害(包括死亡)的危险, 用户务必在其设计中采用必要的安全措施, 如冗余度、防火和防故障等安全设计。
- 本公司产品质量分为: “标准等级”、“专业等级”以及“特殊等级”三种质量等级。

“特殊等级”仅适用于为特定用途而根据用户指定的质量保证程序所开发的日电电子产品。另外, 各种日电电子产品的推荐用途取决于其质量等级, 详见如下。用户在选用本公司的产品时, 请事先确认产品的质量等级。

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

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

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

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

(注)

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

引言

读者对象 本手册适用于希望了解 μ PD79F9211 功能，并设计开发应用系统和程序的工程师。
目标产品如下。

μ PD79F9211

目的 本手册用于帮助用户了解下面**组件**中描述的功能。

组件 μ PD79F9211 手册主要分为两个部分：手册和指令（与 78K0R 系列通用）。

μ PD79F9211 用户手册 (本手册)	78K0R 微控制器 用户手册 指令
----------------------------------	--------------------------

- 引脚功能
- 内部模块功能
- 中断
- 其它内置外设功能
- 电气特性（目标值）
- CPU 功能
- 指令集
- 指令扩展

手册使用方法 在阅读本手册前，读者应掌握电子工程、逻辑电路和微控制器等方面的一般知识。

- 如果读者要了解产品功能：
→ 请按**目录**顺序阅读本手册。 标记“<R>”表示主要修改点。 通过复制 PDF 文件中的“<R>”并填入“查找：”栏内，可方便地搜索到修改点。
- 如何解释寄存器格式：
→ 括号中的位名在 RA78K0R 中被定义为保留字，并且在 CC78K0R 中直接通过 #pragma sfr 定义为一个 sfr 变量。
- 如果读者希望了解 78K0R 系列指令的详细信息：
→ 可参阅 **78K0R 系列指令用户手册 (U17792E)**。

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

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

设备文档

文档名称	文档编号
μPD79F9211 用户手册	本手册
78K0R 微控制器指令用户手册	U17792E

开发工具的相关文档 (软件) (用户手册)

	文档名称	文档编号
<R>	CC78K0R Ver. 2.00 C 编译器。	操作
		语言
<R>	RA78K0R 版 1.20 汇编程序包	操作
		语言
<R>	PM+ Ver. 6.30	U18416E
		U17839E
	ID78K0R-QB Ver. 3.20 集成调试器	操作

开发工具的相关文档 (硬件) (用户手册)

文档名称	文档编号
QB-MINI2 具有编程功能的片上调试仿真器	U18371E

与Flash存储器编程相关文档

文档名称	文档编号
PG-FP4 Flash 存储器编程器用户手册	U15260E
PG-FP5 Flash 存储器编程器用户手册	U18865E

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

其他文档

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

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

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

目 录

第 1 章 概述	16
1.1 特性	17
1.2 应用	18
1.3 订购信息	18
1.4 引脚配置（顶视图）	19
1.5 78K0R/lx3 微控制器产品列表	20
1.6 框图	21
1.7 功能概述	22
第 2 章 引脚功能	24
2.1 引脚功能列表	24
2.1.1 44 引脚产品	25
2.2 引脚功能描述	29
2.2.1 P10 至 P13 (端口 1)	29
2.2.2 P20 至 P27 (端口 2)	29
2.2.3 P30 至 P32 (端口 3)	29
2.2.4 P40 至 P41 (端口 4)	31
2.2.5 P50 至 P52 (端口 5)	31
2.2.6 P70 至 P75 (端口 7)	32
2.2.7 P80 至 P83 (端口 8)	34
2.2.8 P120 至 P124 (端口 12)	34
2.2.9 P150 和 P151 (端口 15)	35
2.2.10 AVREF	35
2.2.12 AVSS	36
2.2.13 RESET	36
2.2.14 REGC	36
2.2.15 VDD	36
2.2.16 VSS	36
2.2.17 FLMD0	36
2.3 引脚 I/O 电路和未使用引脚的推荐连接方式	37
2.3.1 44 引脚产品	37
第 3 章 CPU 架构	43
3.1 存储器空间	43
3.1.1 内部程序存储器空间	45
3.1.2 镜像区域	47
3.1.3 内部数据存储器空间	48
3.1.4 特殊功能寄存器(SFR)区域	49
3.1.5 扩展特殊功能寄存器（第二 SFR：第二 特殊功能寄存器）区域	49
3.1.6 数据存储器寻址	50
3.2 处理器寄存器	51
3.2.1 控制寄存器	51
3.2.2 通用寄存器	53
3.2.3 ES 和 CS 寄存器	55
3.2.4 特殊功能寄存器 (SFRs)	56

3.2.5 扩展特殊功能寄存器（第二SFR：第二特殊功能寄存器）	61
3.3 指令地址寻址	67
3.3.1 相对寻址	67
3.3.2 立即寻址	67
3.3.3 表间接寻址	68
3.3.4 寄存器直接寻址	69
3.4 用于处理数据地址的寻址	70
3.4.1 隐含寻址	70
3.4.2 寄存器地址	70
3.4.3 直接寻址	71
3.4.4 短直接寻址	72
3.4.5 SFR寻址	73
3.4.6 寄存器间接寻址	74
3.4.7 基址寻址	75
3.4.8 相对基址寻址	78
3.4.9 堆栈寻址	79
第4章 端口功能	80
4.1 端口功能	80
4.1.1 44 引脚产品	81
4.2 端口配置	82
4.2.1 端口 1	83
4.2.2 端口 2	85
4.2.3 端口 3	87
4.2.4 端口 4	90
4.2.5 端口 5	92
4.2.6 端口 7	95
4.2.7 端口 8	100
4.2.8 端口 12	103
4.2.9 端口 15	106
4.3 寄存器控制端口功能	108
4.4 端口功能操作	115
4.4.1 写入 I/O 端口	115
4.4.2 从 I/O 端口读取	115
4.4.3 I/O 端口的操作	115
4.4.4 以不同电源电压（2.5 V、3 V）连接到外部设备	116
4.5 使用复用功能时端口模式寄存器和输出锁存的设置	118
4.6 端口寄存器n (Pn)的 1 位操作指令的注意事项	121
第5章 时钟发生器	122
5.1 时钟发生器的功能	122
5.2 时钟发生器的配置	123
5.3 控制时钟发生器的寄存器	125
5.4 系统时钟振荡器	139
5.4.1 X1 振荡器	139
5.4.2 XT1 振荡器	139
5.4.3 内部高速振荡器	142
5.4.4 40MHz内部高速振荡器	142

5.4.5 内部低速振荡器	142
5.4.6 预分频器.....	142
5.5 时钟发生器操作	143
5.6 控制时钟	149
5.6.1 控制高速系统时钟的示例.....	149
5.6.2 控制内部高速振荡时钟的示例	152
5.6.3 控制副系统时钟示例.....	154
5.6.4 内部低速振荡时钟的控制示例	156
5.6.5 CPU时钟状态迁移图	157
5.6.6 CPU时钟改变之前的状况与改变之后的处理.....	164
5.6.7 CPU时钟和主系统时钟切换所需的时间	166
5.6.8 时钟振荡停止前的状况	167
第 6 章 定时器阵列单元TAUS	168
6.1 定时器阵列单元的功能TAUS	168
6.1.1 独立操作时每个通道的功能	168
6.1.2 和其它通道组合操作时每个通道的功能	169
6.1.3 LIN总线支持功能（仅限通道 7）	170
6.2 定时器阵列单元TAUS的配置	171
6.3 控制定时器阵列单元TAUS的寄存器.....	178
6.4 通道输出 (TOn引脚) 控制.....	203
6.4.1 TOn引脚输出电路配置	203
6.4.2 TOn引脚输出设置.....	204
6.4.3 通道输出操作的注意事项.....	205
6.4.4 TOn位的集中操作.....	208
6.4.5 关于计数操作开始时的定时器中断和TOn引脚输出.....	209
6.5 通道输入 (Tin引脚) 控制	210
6.5.1 Tin边沿检测电路	210
6.6 定时器阵列单元TAUS的基本功能	211
6.6.1 独立操作功能和组合操作功能的概述.....	211
6.6.2 组合操作功能的基本规则.....	211
6.6.3 组合操作功能的基本规则的应用范围.....	212
6.7 作为独立通道的定时器阵列单元TAUS的操作	213
6.7.1 作为间隔定时器/方波输出的操作	213
6.7.2 作为外部事件计数操作	217
6.7.3 作分频器操作	220
6.7.4 作为输入脉冲间隔测量的操作	224
6.7.5 用作输入信号高/ 低电平宽度测量的操作	228
6.8 定时器阵列单元TAUS的多通道操作.....	232
6.8.1 作为PWM功能操作.....	232
6.8.2 作为单脉冲输出功能操作.....	239
6.8.3 作为多路PWM输出功能的操作	246
第 7 章 变频控制功能.....	253
7.1 功能概述	253
7.2 变频控制功能的配置	254
7.3 控制定时器阵列单元TAUS和变频控制功能模块的寄存器	258
7.4 使用变频控制功能的操作	271

7.4.1 作为实时输出功能的操作（类型 1）	271
7.4.2 作为实时输出功能的操作（类型 2）	278
7.4.3 作为 6 相PWM输出功能操作	285
7.4.4 作为三角波PWM输出功能操作	292
7.4.5 作为带有死区时间的三角波PWM输出功能操作	299
7.4.6 作为 6 相三角波PWM输出功能操作	308
7.4.7 中断信号缩短功能	317
7.4.8 作为A/D转换触发输出功能的操作(类型 1)	323
7.4.9 作为A/D转换触发输出功能的操作(类型 2)	329
7.4.10 作为连接实时输出功能的操作（类型 1）	336
7.4.11 作为连接实时输出功能的操作（类型 2）	344
7.4.12 作为连接实时输出功能的操作（类型 3）	352
7.4.13 作为非互补调制输出功能的操作（类型 1）	361
7.4.14 作为非互补调制输出功能的操作（类型 2）	371
7.4.15 作为非互补调制输出功能的操作	381
第 8 章 比较器/可编程增益放大器	392
8.1 比较器和可编程增益放大器的功能	392
8.2 比较器和可编程增益放大器的配置	394
8.3 控制比较器和可编程增益放大器的寄存器	394
8.4 比较器和可编程增益放大器的操作	400
8.4.1 启动比较器和可编程增益放大器的操作	400
8.4.2 停止比较器和可编程增益放大器的操作	405
第 9 章 实时计数器	406
9.1 实时计数器的功能	406
9.2 实时计数器的配置	406
9.3 控制实时计数器的寄存器	408
9.4 实时计数器操作	420
9.4.1 启动实时计数器的操作	420
9.4.2 读/写实时计数器	421
9.4.3 设置实时计数器报警	423
第 10 章 看门狗定时器	424
10.1 看门狗定时器的功能	424
10.2 看门狗定时器的配置	425
10.3 控制看门狗定时器的寄存器	426
10.4 看门狗定时器的操作	427
10.4.1 看门狗定时器的控制操作	427
10.4.2 看门狗定时器溢出时间的设置	428
10.4.3 看门狗定时器窗口打开周期的设置	429
10.4.4 看门狗定时器间隔中断的设置	430
第 11 章 A/D转换器	431
11.1 A/D转换器的功能	431
11.2 A/D转换器的配置	432
11.3 A/D转换器使用的寄存器	434
11.4 A/D转换器操作	446

11.4.1 A/D转换器的基本操作	446
11.4.2 输入电压和转换结果	448
11.4.3 触发器模式选择	449
11.4.4 A/D转换器操作模式	450
11.5 A/D转换器特征表的阅读方法	453
11.6 A/D转换器的注意事项	455
第 12 章 串行阵列单元.....	459
12.1 串行阵列单元的功能	459
12.1.1 3 线串行 I/O (CSI00、CSI01、CSI10)	459
12.1.2 UART (UART0, UART1)	460
12.1.3 简易 I ² C (IIC10)	461
12.2 串行阵列单元的配置	462
12.3 控制串行阵列单元的寄存器	466
12.4 操作停止模式	488
12.4.1 由部件停止操作	488
12.4.2 由通道停止操作	489
12.5 3 线串行 I/O (CSI00、CSI01、CSI10) 通信的操作	490
12.5.1 主发送	491
12.5.2 主接收	500
12.5.3 主发送/接收	506
12.5.4 从发送	514
12.5.5 从接收	523
12.5.6 从发送/接收	529
12.5.7 计算发送时钟频率	538
12.6 UART (UART0, UART1) 通信的操作	540
12.6.1 UART 发送	541
12.6.2 UART 接收	551
12.6.3 LIN 发送	558
12.6.4 LIN 接收	561
12.6.5 计算波特率	565
12.6.5 计算波特率	566
12.7 简易 I ² C (IIC10) 通信操作	570
12.7.1 地址域发送	571
12.7.2 数据发送	576
12.7.3 数据接收	579
12.7.4 停止条件的产生	583
12.7.5 计算发送速率	584
12.8 错误的处理程序	587
12.9 寄存器设置和引脚之间的关系	589
第 13 章 乘法器/除法器.....	593
13.1 乘法器/除法器的功能	593
13.2 乘法器/除法器的配置	593
13.3 乘法器/除法器的寄存器	598
13.4 乘法器/除法器的操作	599
13.4.1 乘法运算	599
13.4.2 除法操作	600

第 14 章 DMA 控制器	602
14.1 DMA控制器	602
14.2 DMA控制器的配置	603
14.3 控制DMA控制器的寄存器	606
14.4 DMA控制器的操作	609
14.4.1 操作顺序	609
14.4.2 传输模式	610
14.4.3 DMA传输中止	610
14.5 DMA控制器设置的实例	611
14.5.1 CSI连续传输	611
14.5.2 A/D转换结果的连续捕捉	613
14.5.3 UART连续接收+ACK 传输	615
14.5.4 通过DWAITn保持DMA传输挂起	617
14.5.5 软件强行中止	618
14.6 使用DMA控制器的注意事项	619
第 15 章 中断函数	621
15.1 中断函数的类型	621
15.2 中断源和配置	621
15.3 控制中断函数的寄存器	625
15.4 中断服务操作	636
15.4.1 可屏蔽的中断响应	636
15.4.2 软件中断请求响应	639
15.4.3 中断嵌套	639
15.4.4 保持中断请求	643
第 16 章 待机功能	644
16.1 待机功能及配置	644
16.1.1 待机功能	644
16.1.2 控制待机功能的寄存器	645
16.2 待机功能的操作	648
16.2.1 HALT模式	648
16.2.2 STOP模式	653
第 17 章 复位功能	659
17.1 确认复位源的寄存器	668
第 18 章 上电清零电路	669
18.1 上电清零电路的功能	669
18.2 上电清零电路的配置	670
18.3 上电清零电路的操作	670
18.4 上电清零电路使用注意事项	673
第 19 章 低电压检测电路	675
19.1 低电压检测电路的功能	675
19.2 低电压检测电路的配置	676
19.3 控制低电压检测电路的寄存器	676
19.4 低电压检测电路的操作	681

19.4.1 当用于复位	682
19.4.2 用作中断时	688
19.5 低电压检测电路的注意事项	694
第 20 章 稳压器	698
20.1 稳压器概述	698
20.2 控制稳压器的寄存器	698
第 21 章 选项字节	699
21.1 选项字节的功能	699
21.1.1 用户选项字节 (000C0H ~ 000C2H/ 010C0H ~ 010C2H)	699
21.1.2 片上调试选项字节 (000C3H/ 010C3H)	700
21.2 用户选项字节的格式	700
21.3 片上调试选项字节的格式	702
21.4 选项字节的设置	703
第 22 章 FLASH 存储器	704
22.1 用Flash存储器编程器写入	704
22.2 编程环境	706
22.3 通信模式	706
22.4 On Board方式的引脚连接	707
22.4.1 FLMD0 引脚	707
22.4.2 TOOL0 引脚	708
22.4.3 RESET 引脚	708
22.4.4 端口引脚	709
22.4.5 REGC 引脚	709
22.4.6 X1 和X2 引脚	709
22.4.7 供电电压	709
22.5 控制Flash 存储器的寄存器	709
22.6 编程方法	710
22.6.1 控制Flash存储器	710
22.6.2 Flash 存储器编程模式	710
22.6.3 选择通信模式	711
22.6.4 通信命令	711
22.7 安全设置	713
22.8 通过自编程进行Flash存储器编程	715
22.8.1 引导交换功能	717
22.8.2 Flash保护窗口功能	719
第 23 章 片上调试功能	720
23.1 连接QB-MINI2 到μPD79F9211	720
23.2 片上调试安全ID	721
23.3 用户资源的确保	721
第 24 章 BCD修正电路	723
24.1 BCD 修正电路功能	723
24.2 BCD修正电路使用的寄存器	723
24.3 BCD 修正电路操作	724

第 25 章 指令集 726

25.1 操作列表使用规则..... 726

25.1.1 操作数标识符和指定方法 726

25.1.2 操作栏描述 727

25.1.3 标志操作栏的描述 728

25.1.4 PREFIX指令..... 728

25.2 操作列表 729

第 26 章 电气特性（目标值） 746

第 27 章 封装图 788

附录A 开发工具..... 789

A.1 软件包 791

A.2 语言处理软件 791

A.3 Flash存储器编程工具 792

A.3.1 当使用Flash存储器编程器PG-FP5、PG-FP4 时..... 792

A.3.2 当使用具有编程功能的片上调试仿真器QB-MINI2..... 792

A.4 调试工具 (硬件) 793

A.4.2 当使用具有编程功能的片上调试仿真器QB-MINI2 时 793

第 1 章 概述

μPD79F9211 是一款 16 位的单片微控制器，它采用 78K0R CPU 内核并且集成多个外围功能，如 ROM/RAM、多功能定时器、多功能串行接口、A/D 转换器、可编程增益放大器(PGA)、比较器、实时计数器和看门狗定时器。

该产品设计用于通常用两块芯片控制的变频器控制应用。多功能定时器 (定时器阵列单元 TAUS) 在 40MHz 的最大分辨率下可进行多种 PWM 操作，并提供多种功能，如带死区时间的 PWM (互补 PWM × 2 通道) 输出功能，带死区时间的 6 相 PWM 输出功能和 DC 变频实时输出功能并可实现变频器控制。此外，多功能定时器配备了双通道比较器用于自动防故障装置应用以及当过电流产生时可设置 PWM 为高阻抗状态。

考虑到马达应用，内置了内部高速振荡器(CPU 时钟为 20 MHz，定时器为 40 MHz)。这可以解决诸如外部振荡器停振等问题。

μPD79F9211 可在各种情况下提供高价值性能。

1.1 特性

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

项目 产品型号	程序存储器 (ROM)		数据存储器 (RAM)
μ PD79F9211	Flash 存储器	16 KB	1 KB

<R> 注 当使用自编程功能时，为 2KB。

- <R>
- 片上 8 MHz 和 40 MHz 内部高速振荡时钟
 - 片上单电源 Flash 存储器（禁止芯片擦除/块擦除/写入功能）
 - 自编程（具有引导交换功能/Flash 窗口屏蔽功能）
 - 片上调试功能
 - 片上上电清零电路 (POC) 和低电压检测电路 (LVI)

- <R>
- 片上看门狗定时器（用专用内部低速振荡时钟操作可操作）
 - 片上乘法器/除法器（16 位 \times 16 位, 32 位 \div 32 位）
 - 片上 BCD 调整

- I/O 端口：

44 引脚产品：37

- 定时器：14 通道

- 16 位定时器：12 通道
- 看门狗定时器：单通道
- 实时计数器：单通道
- 片上马达控制选项单元

- 片上比较器/可编程增益放大功能

- 串行接口

- CSI：2 通道/UART（支持 LIN 总线）：单通道
- CSI：单通道/UART：单通道/简易 I²C：单通道

- 10 位分辨率 A/D 转换器(AV_{REF} = 2.7~5.5 V)

44 引脚产品：10 通道

- 电源供电电压：V_{DD} = 2.7~5.5 V

- 工作环境温度：T_A = -40 ~ +85°C

1.2 应用

- 家用电器
 - 空气净化器
 - 空调设备
 - 冰箱
 - 洗碗机
- 电动自行车

1.3 订购信息

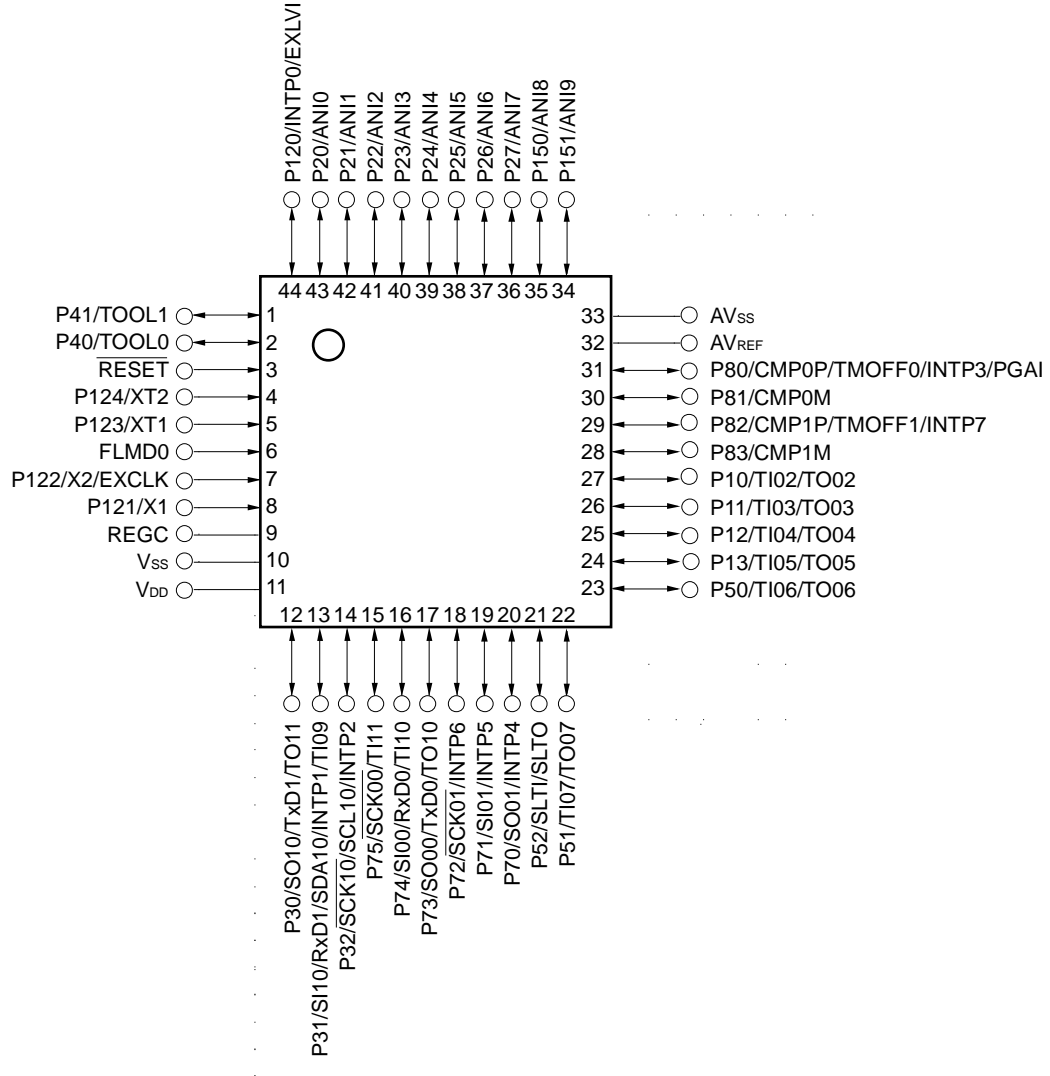
- **Flash 存储器版本（无铅产品）**

产品型号	封装
μPD79F9211GB-8ES-A	44 引脚塑封 LQFP (10 × 10)

注意事项 μPD79F9211 具有片上调试功能，可用于开发和评价。在量产时不要使用产品中的片上调试功能，因为 flash 存储器中保证的可重写次数可能会超出，从而不能保证产品的可靠性。对于因使用片上调试功能而产生的问题，NEC 电子不承担任何责任。

1.4 引脚配置（顶视图）

- 44 引脚塑封 LQFP (10 × 10)



- 注意事项
1. 使 AV_{SS} 与 V_{SS} 同电势。
 2. 通过电容（推荐：0.47 ~ 1 μ F）连接 REGC 引脚到 V_{SS}。

引脚标识

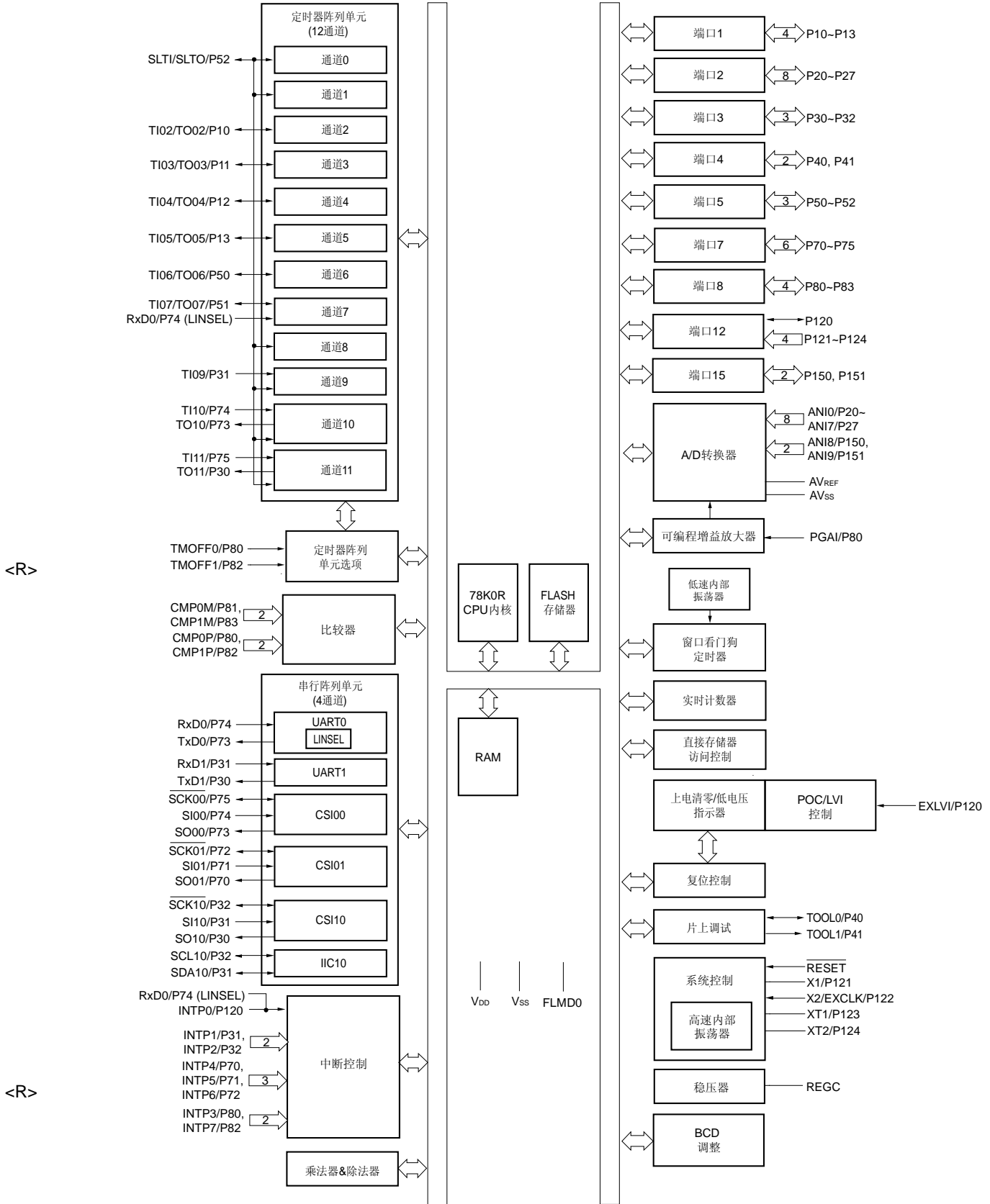
ANI0~ANI9:	模拟输入	$\overline{\text{RESET}}$:	复位
AVREF:	模拟参考电压	RxD0、RxD1:	接收数据
AVss:	模拟接地	$\overline{\text{SCK00}}$ 、 $\overline{\text{SCK01}}$ 、 $\overline{\text{SCK10}}$:	串行时钟输入/输出
CMP0M、CMP1M:	比较器输入(负)	SCL10:	串行时钟输入/输出
CMP0P、CMP1P:	比较器输入(正)	SDA10:	串行数据输入/输出
EXCLK:	外部时钟输入 (主系统时钟)	SI00、SI01、SI10:	串行数据输入
EXLVI:	外部电势输入 用于低电压检测器	SLTI:	可选定时器输入
FLMD0:	Flash 编程模式	SLTO:	可选定时器输出
INTP0~INTP7:	外部中断输入	SO00、SO01、SO10:	串行数据输出
P10~P13:	端口 1	TI02~TI07、TI09~TI11:	定时器输入
P20~P27:	端口 2	TMOFF0、TMOFF1:	定时器关闭输入
P30~P32:	端口 3	TO02~TO07、TO10、TO11:	定时器输出
P40、P41:	端口 4	TOOL0:	对工具的数据输入/输出
P50~P52:	端口 5	TOOL1:	对工具的时钟输出
P70~P75:	端口 7	TxD0、TxD1:	发送数据
P80~P83:	端口 8	VDD:	电源电压
P120~P124:	端口 12	Vss:	地
P150、P151:	端口 15	X1、X2:	晶体振荡器 (主系统 时钟)
PGAI:	可编程增益放大器输入	XT1、XT2:	晶体振荡器 (副系统 时钟)
REGC:	稳压器电容		

1.5 UPD79F9211 微控制器产品列表

ROM	RAM	UPD79F9211
		44 引脚
16 KB	1 KB	$\mu\text{PD79F9211}^{\text{※}}$

注 当使用自编程功能时，为 2KB。

1.6 框图



1.7 功能概述

(1/2)

<R> <R>	项目		μPD79F9211
	内部存储器	Flash 存储器 (支持自编程)	16 KB
		RAM	1 KB
	存储器空间		1 MB
	主系统时钟 (振荡频率)	高速系统时钟	X1 (晶体/陶瓷)振荡，外部主系统时钟输入(EXCLK) 2 ~ 20 MHz: V _{DD} = 2.7 ~ 5.5 V
		内部高速振荡时钟	内部振荡 8 MHz(典型值): V _{DD} = 2.7 ~ 5.5 V
		40 MHz 内部高速振荡 时钟	内部振荡 40 MHz(典型值): V _{DD} = 2.7 ~ 5.5 V
	副系统时钟 (振荡频率)		XT1(晶体)振荡 32.768 kHz (典型值) : V _{DD} = 2.7 ~ 5.5 V
	内部低速振荡时钟 (WDT 专用)		内部振荡 30 kHz (典型值) : V _{DD} = 2.7 ~ 5.5 V
	通用寄存器		8 位 × 32 寄存器 (8 位 × 8 寄存器 × 4 组)
指令最短执行时间		0.05 μs (高速系统时钟: f _{MX} = 20 MHz 操作) 0.125 μs (内部高速振荡时钟: f _{IH} = 8 MHz (典型值) 操作) 61 μs (副系统时钟: f _{SUB} = 32.768 kHz 操作)	
<R>	指令集		• 8 位操作、16 位操作 • 乘法 (8 位 × 8 位) • 位操作 (设置、复位、测试、布尔操作) 等
I/O 端口		44 引脚产品 总计 37 CMOS I/O 33 CMOS 输入 4 CMOS 输出 — N 沟道漏极 I/O (6 V 耐压) —	
定时器		• 16 位定时器: 12 通道 • 看门狗定时器: 单通道 • 实时计数器: 单通道	
	定时器输出	9 (PWM 输出: 9 ^{注2})	

- 注
1. 当使用自编程功能时, 为 2KB。
 2. 输出的数量根据设置而不同。

(2/2)

项目		μ PD79F9211
A/D 转换器		10 位分辨率 \times 10 通道 ($AV_{REF} = 2.7 \sim 5.5V$)
串行接口		<ul style="list-style-type: none"> • CSI: 2 通道/UART (支持 LIN 总线: 单通道) • CSI: 单通道/UART: 单通道/简易 I²C: 单通道
乘法器/除法器		16 位 \times 16 位 = 32 位 (乘法) 32 位 \div 32 位 = 32 位 (除法)
DMA 控制器		2 通道
向量中断源	内部	33
	外部	8
复位		<ul style="list-style-type: none"> • 使用 RESET 引脚复位 • 使用看门狗定时器内部复位 • 使用上电清零内部复位 • 使用低电压检测电路内部复位 • 非法指令执行引起内部复位^注
上电清零电路		<ul style="list-style-type: none"> • 启动复位: 1.61 \pm 0.09 V • 关机复位: 1.59 \pm 0.09 V
低电压检测器		1.91 V \sim 4.22 V (16 阶段)
片上调试功能		提供
供电电压		$V_{DD} = 2.7 \sim 5.5 V$
工作环境温度		$T_A = -40 \sim +85^{\circ}C$
封装		44 引脚塑封 LQFP (10 \times 10) (引脚间距: 0.8 mm)

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

第 2 章 引脚功能

2.1 引脚功能列表

有 2 种类型的引脚 I/O 缓冲器供给电源：AVREF 和 VDD。下表显示了这些供电电源与引脚之间的关系。

表 2-1. 引脚 I/O 缓冲区供电电源

供电电源	对应的引脚
AVREF	P20 ~ P27, P150、P151 P80 ~ P83
VDD	<ul style="list-style-type: none">• 除去 P20~P27、P80~P83 以外的端口引脚• 除去端口引脚以外的引脚

2.1.1 44 引脚产品

(1) 端口功能 (1/2):

功能名称	I/O	功能	复位后	复用功能
P10	I/O	端口 1。 4 位 I/O 端口。 可以指定输入/输出为 1 位单元模式。 通过软件设置，可以定义片内上拉电阻的使用。	输入端口	TI02/TO02
P11				TI03/TO03
P12				TI04/TO04
P13				TI05/TO05
P20~P27	I/O	端口 2。 8 位 I/O 端口。 可以指定输入/输出为 1 位单元模式。	数字输入 端口	ANI0~ANI7
P30	I/O	端口 3。 3 位 I/O 端口。 P31 和 P32 输入可设置成 TTL 缓冲器。 P30 至 P32 的输出可设置成 N 沟道漏极开路输出 (V _{DD} 耐压)。 可以指定输入/输出为 1 位单元模式。 通过软件设置，可以定义片内上拉电阻的使用。	输入端口	SO10/TxD1/TO11
P31				SI10/RxD1/SDA10/ INTP1/TI09
P32				SCK10/SCL10/ INTP2
P40 [※]	I/O	端口 4。 2 位 I/O 端口。 可以指定输入/输出为 1 位单元模式。 通过软件设置，可以定义片内上拉电阻的使用。	输入端口	TOOL0
P41				TOOL1
P50	I/O	端口 5。 3 位 I/O 端口。 可以指定输入/输出为 1 位单元模式。 通过软件设置，可以定义片内上拉电阻的使用。	输入端口	TI06/TO06
P51			输入端口	TI07/TO07
P52			输入端口	SLTI/SLTO
P70	I/O	端口 7。 6 位 I/O 端口。 P71、P72、P74 和 P75 的输入可以被设置到 TTL 缓冲器。 P70、P72、P73 和 P75 的输出可设置为 N 沟道漏极开路输出 (V _{DD} 耐压)。 可以指定输入/输出为 1 位单元模式。 通过软件设置，可以定义片内上拉电阻的使用。	输入端口	SO01/INTP4
P71				SI01/INTP5
P72				SCK01/INTP6
P73				SO00/TxD0/TO10
P74				SI00/RxD0/TI10
P75				SCK00/TI11
P80	I/O	端口 8。 4 位 I/O 端口。 可以以 1 位单元输入/输出模式。 P80 至 P83 的输入可以设置为比较器输入或可编程增益放大器输入。	模拟输入	CMP0P/TMOFF0/ INTP3/PGAI
P81				CMP0M
P82				CMP1P/TMOFF1/ INTP7
P83				CMP1M

注 如果通过选项字节使能片上调试，要确认上拉外部 P40/TOOL0 引脚 (参见在 2.2.4 P40、P41 (端口 4)的注意事项)。

(1) 端口功能 (2/2): 44 引脚产品

功能名称	I/O	功能	复位后	复用功能
P120	I/O	端口 12。 1 位 I/O 端口和 4 位输入端口。 仅对 P120，可以指定输入/输出为 1 位单元模式。 仅对 P120，通过软件设置，可以定义片内上拉电阻的使用。	输入端口	INTP0/EXLVI
P121	输入			X1
P122				X2/EXCLK
P123				XT1
P124				XT2
P150、P151	I/O	端口 15。 2 位 I/O 端口。 可以以 1 位单元输入/输出模式。	数字输入 端口	ANI8、ANI9

(2) 无端口功能 (1/2):

功能名称	I/O	功能	复位后	复用功能
ANI0~ANI7	输入	A/D 转换器模拟输入	数字输入 端口	P20~P27
ANI8、ANI9				P150、P151
CMP0M	输入	在比较器 0 的(-)端输入电压	模拟输入	P81
CMP0P	输入	在比较器 0 的(+)端输入电压		P80/TMOFF0/ INTP3/PGAI
CMP1M	输入	在比较器 1 的(-)端输入电压		P83
CMP1P	输入	在比较器 1 的(+)端输入电压		P82/TMOFF1/INTP7
EXLVI	输入	为外部低电压检测提供的电势输入	输入端口	P120/INTP0
INTP0	输入	可指定有效沿（上升沿、下降沿、或双沿）时的外部中断请求输入	输入端口	P120/EXLVI
INTP1				P31/SI10/RxD1/ SDA10/TI09
INTP2				P32/SCK10/SCL10
INTP3			模拟输入	P80/CMP0P/ TMOFF0/PGAI
INTP4			输入端口	P70/SO01
INTP5				P71/SI01
INTP6				P72/SCK01
INTP7			模拟输入	P82/CMP1P/TMOFF1
PGAI	输入	可编程增益放大器输入	模拟输入	P80/CMP0P/ TMOFF0/INTP3
REGC	-	为内部操作，连接校准器输出（2.4 V）稳定电容。 通过电容器(0.47~1 μ F：推荐)连接到 V _{SS} 。	-	-
RESET	输入	系统复位输入	-	-
RxD0	输入	串行数据输入至 UART0	输入端口	P74/SI00/TI10
RxD1		串行数据输入至 UART1		P31/SI10/SDA10/ INTP1/TI09
SCK00	I/O	CSI00、CSI01 和 CSI10 的时钟输入/输出	输入端口	P75/TI11
SCK01				P72/INTP6
SCK10				P32/SCL10/INTP2
SCL10	I/O	简易 I ² C 的时钟输入/输出	输入端口	P32/SCK10/INTP2
SDA10	I/O	简易 I ² C 的串行数据 I/O	输入端口	P31/SI10/RxD1/ INTP1/TI09
SI00	输入	串行数据输入至 CSI00、CSI01 和 CSI10	输入端口	P74/RxD0/TI10
SI01				P71/INTP5
SI10				P31/RxD1/SDA10/ INTP1/TI09
SLTI	输入	16 位定时器 00、01、08、09、10、11 输入	输入端口	P52/SLTO
SLTO	输出	16 位定时器 00、01、08、09、10、11 输出	输入端口	P52/SLTI
SO00	输出	从 CSI00、CSI01 和 CSI10 的串行数据输出	输入端口	P73/TxD0/TO10
SO01				P70/INTP4
SO10				P30/TxD1/TO11

(2) 无端口功能 (2/2):

功能名称	I/O	功能	复位后	复用功能
TI02	输入	外部计数时钟输入至 16 位定时器 02	输入端口	P10/TO02
TI03		外部计数时钟输入至 16 位定时器 03		P11/TO03
TI04		外部计数时钟输入至 16 位定时器 04		P12/TO04
TI05		外部计数时钟输入至 16 位定时器 05		P13/TO05
TI06		外部计数时钟输入至 16 位定时器 06		P50/TO06
TI07		外部计数时钟输入至 16 位定时器 07		P51/TO07
TI09		外部计数时钟输入至 16 位定时器 09		P31/SI10/RxD1/ SDA10/INTP1
TI10		外部计数时钟输入至 16 位定时器 10		P74/SI00/RxD0
TI11		外部计数时钟输入至 16 位定时器 11		P75/SCK00
TMOFF0	输入	定时器引脚高阻控制输入	模拟输入	P80/CMP0P/INTP3/ PGAI
TMOFF1				P82/CMP1P/INTP7
TO02	输出	16 位定时器 02 输出	输入端口	P10/TI02
TO03		16 位定时器 03 输出		P11/TI03
TO04		16 位定时器 04 输出		P12/TI04
TO05		16 位定时器 05 输出		P13/TI05
TO06		16 位定时器 06 输出		P50/TI06
TO07		16 位定时器 07 输出		P51/TI07
TO10		16 位定时器 10 输出		P73/SO00/TxD0
TO11		16 位定时器 11 输出		P30/SO10/TxD1
TxD0	输出	从 UART0 的串行数据输出	输入端口	P73/SO00/TO10
TxD1		从 UART1 的串行数据输出		P30/SO10/TO11
X1	—	主系统时钟的振荡器连接	输入端口	P121
X2	—		输入端口	P122/EXCLK
EXCLK	输入	主系统时钟的外部时钟输入	输入端口	P122/X2
XT1	—	副系统时钟的振荡器连接	输入端口	P123
XT2	—		输入端口	P124
V _{DD}	—	正向供电电源（除去 P20 至 P27、P80 至 P83、P150、P151 和端口以外的端口引脚）	—	—
AV _{REF}	—	<ul style="list-style-type: none"> A/D 转换器和比较器参考电压输入 对 P20 至 P27、P80 至 P83、P150、P151、A/D 转换器、可编程增益放大器和比较器使用的正向电源 	—	—
V _{SS}	—	地电位（除 P20 至 P27、P80 至 P83、P150、P151 以外的端口引脚以及除端口以外的端口引脚）	—	—
AV _{SS}	—	A / D 转换器、可编程增益放大器、比较器、P20 至 P27、P80 至 P83、P150 和 P151 的地电位	—	—
FLMD0	—	Flash 存储器编程模式设置	—	—
TOOL0	I/O	Flash 存储器编程器/调试器的数据 I/O	输入端口	P40
TOOL1	输出	调试器的时钟输出	输入端口	P41

2.2 引脚功能描述

2.2.1 P10 至P13 (端口 1)

P10 至 P13 功能为 4 位 I/O 端口。这些引脚也作为定时器 I/O 的功能。

可以指定下列操作模式为 1 位单元。

(1) 端口模式

P10 至 P13 功能为 4 位 I/O 端口。P10 至 P13 可通过端口模式寄存器 1(PM1)按位设置为输入或输出端口。由上拉电阻选择寄存器 1 (PU1) 指定使用片内上拉电阻。

(2) 控制模式

P10 至 P13 功能作为定时器 I/O。

(a) TI02 至 TI05

这些是用于输入外部计数时钟/捕捉触发到 16 位定时器 02 至 05 的引脚。

(b) TO02 至 TO05

这些是 16 位定时器 02 至 05 的定时器输出引脚。

2.2.2 P20 至P27 (端口 2)

P20 至 P27 作为 8 位 I/O 端口的功能。这些引脚也具有作为 A/D 转换器模拟输入的功能。

可以指定下列操作模式为 1 位单元。

(1) 端口模式

P20 至 P27 作为 8 位 I/O 端口的功能。通过端口模式寄存器 2 (PM2)，P20 至 P27 可以以 1 位单元设置为输入或输出端口。

(2) 控制模式

P20 至 P27 作为 A/D 转换器模拟输入引脚 (ANI0 至 ANI7) 的功能。当把这些引脚作为模拟输入引脚使用时，参见 12.6 (5) ANI0/P20 至 ANI7/P27 和 ANI8/P150。

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

2.2.3 P30 至P32 (端口 3)

P30 至 P32 功能作为 3 位 I/O 端口。这些引脚也有作为串行接口数据 I/O、时钟 I/O、外部中断请求输入和定时器 I/O 的功能。

使用端口输入模式寄存器 3 (PIM3)，可以通过一个常规输入缓冲器或 TTL 输入缓冲器以 1 位单元指定 P30 和 P31 引脚的输入。

使用端口输出模式寄存器 3 (POM3)，可以以 1 位单元指定 P30 至 P32 引脚输出为普通 CMOS 输出或 N 沟道漏极开路输出(V_{DD} 耐压)。

可以指定下列操作模式为 1 位单元。

(1) 端口模式

P30 至 P32 功能作为 3 位 I/O 端口。P30 至 P32 可通过端口模式寄存器 3(PM3) 按位设置为输入或输出端口。由上拉电阻选择寄存器 3 (PU3) 指定使用片内上拉电阻。

(2) 控制模式

P30 至 P32 有作为串行接口数据 I/O、时钟 I/O、外部中断请求输入和定时器 I/O 的功能。

(a) SI10

这是串行接口 CSI10 的串行数据输入引脚。

(b) SO10

这是串行接口 CSI10 的串行数据输出引脚。

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

这是串行接口 CSI10 的串行时钟 I/O 引脚。

(d) TxD1

这是串行接口 UART1 的串行数据输出引脚。

(e) RxD1

这是串行接口 UART1 的串行数据输入引脚。

(f) SDA10

这是简易 I²C 串行接口的串行数据 I/O 引脚。

(g) SCL10

这是简易 I²C 串行接口的串行时钟 I/O 引脚。

(h) INTP1、INTP2

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

(i) TI09

这是用于输入外部计数时钟/捕捉触发到 16 位定时器 09 的引脚。

(j) TO11

这是 16 位定时器 11 的定时器输出引脚。

注意事项 使用 P30/SO10/TxD1/TO11 和 P32/ $\overline{\text{SCK10}}$ /SCL10/INTP2 作为通用端口，设置串行通信操作设置寄存器 02(SCR02)到默认状态(0087H)。此外，清除端口输出模式寄存器 3(POM3)至 00H。

2.2.4 P40 至P41 (端口 4)

P40 至 P41 功能作为 2 位 I/O 端口。这些引脚有作为 Flash 存储器编程器/调试器和时钟输出的数据 I/O 的功能。
可以指定下列操作模式为 1 位单元。

(1) 端口模式

P40 至 P41 功能作为 2 位 I/O 端口。P40 至 P41 可通过端口模式寄存器 4(PM4) 按位设置为输入或输出端口。由上拉电阻选择寄存器 4 (PU4) 指定使用片内上拉电阻。

当允许片上调试（通过选项字节）时，确保连接一个外部上拉电阻到 P40。

(2) 控制模式

P40 和 P41 有作为 Flash 存储器编程器/调试器和时钟输出的数据 I/O 的功能。

(a) TOOL0

这是 Flash 存储器编程器/调试器的数据 I/O 引脚。

当允许片上调试时，确认此外部引脚为上拉状态(下拉禁止)。

(b) TOOL1

这是调试器的时钟输出引脚。

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

1 线模式：可用作端口（P41）。

2 线模式：可用作 TOOL1 引脚，不能作为端口（P41）使用。

注意事项 P40/TOOL0 引脚功能与下面(a)至(c)中描述的不同。

在(b)或(c)的情况中，确保指定的连接。

(a) 在普通操作模式下，通过选项字节(000C3H)禁止片上调试(OCDENSET = 0)时

=> 此引脚用作端口引脚(P40)。

(b) 在普通操作模式下，通过选项字节(000C3H)允许片上调试(OCDENSET = 1)时

=> 通过一个外部电阻连接该引脚到 V_{DD}，并且在复位释放前始终输入高电平到该引脚。

(c) 当使用片上调试功能，或在 Flash 存储器编程器写入模式下时

=> 此引脚作为 TOOL0 使用。

直接连接该引脚到片上调试仿真器或 Flash 存储器编程器上，或者通过外部电阻连接到 V_{DD} 使得该引脚上拉。

2.2.5 P50 至P52 (端口 5)

P50 至 P52 功能作为 3 位 I/O 端口。这些引脚也作为定时器 I/O 的功能。

可以指定下列操作模式为 1 位单元。

(1) 端口模式

P50 至 P52 功能作为 3 位 I/O 端口。通过端口模式寄存器 5(PM5)，P50 至 P52 可按位设置为输入或输出端口。由上拉电阻选择寄存器 5(PU5)指定使用片内上拉电阻。

(2) 控制模式

P50 至 P52 功能作为定时器 I/O。

(a) TI06、TI07

这些是用于输入外部计数时钟/捕捉触发到 16 位定时器 06 至 07 的引脚。

(b) TO06、TO07

这些是 16 位定时器 06 至 07 的定时器输出引脚。

(c) SLTI

通过设置输入切换控制寄存器 (ISC)，此引脚用于输入一个外部计数时钟或一个捕捉触发到 16 位定时器 00、01、08、09、10 和 11。

(d) SLTO

通过设置输入切换控制寄存器 (ISC)，此引脚用作 16 位定时器 00、01、08、09、10 和 11 的一个定时器输出引脚。

2.2.6 P70 至 P75 (端口 7)

P70 至 P75 功能作为 6 位 I/O 端口。这些引脚也有作为串行接口数据 I/O、时钟 I/O、外部中断请求输入和定时器 I/O 的功能。

利用端口输入模式寄存器 7(PIM7)，可通过一个正常的输入缓冲器或 TTL 输入缓冲器，以 1 位单元指定输入至 P71、P72、P74 和 P75 引脚。

利用端口输出模式寄存器 7(POM7)，从 P70、P72、P73 和 P75 引脚的输出，可以以 1 位单元指定为正常 CMOS 输出或 N 沟道漏极开路输出 (V_{DD} 耐压)。

可以指定下列操作模式为 1 位单元。

(1) 端口模式

P70 至 P75 功能作为 6 位 I/O 端口。通过端口模式寄存器 7(PM7)，P70 至 P75 可以以 1 位单元设置为输入或输出端口。由上拉电阻选择寄存器 7(PU7)指定使用片内上拉电阻。

(2) 控制模式

P70 至 P75 有作为串行接口数据 I/O、时钟 I/O、外部中断请求输入和定时器 I/O 的功能。

(a) TI10、TI11

这些是用于输入一个外部计数时钟/捕捉触发到 16 位定时器 10 和 11 的引脚。

(b) TO10

这是 16 位定时器 10 的定时器输出引脚。

(c) SI00、SI01

这些是串行接口 CSI00 和 CSI01 的串行数据输入引脚。

(d) SO00、SO01

这些是串行接口 CSI00 和 CSI01 的串行数据输出引脚。

(e) $\overline{\text{SCK00}}$ 、 $\overline{\text{SCK01}}$

这些是串行接口 CSI00 和 CSI01 的串行时钟 I/O 引脚。

(f) TxD0

这是串行接口 UART0 的串行数据输出引脚。

(g) RxD0

这是串行接口 UART0 的串行数据输入引脚。

(h) INTP4 至 INTP6

这些是外部中断请求输入引脚，用它可指定有效沿（上升沿、下降沿，或双沿）。

注意事项 要将 P70/SO01/INTP4 、 P72/SCK01/INTP6 、 $\text{P73/SO00/TxD0/TO10}$ 和 P75/SCK00/TI11 用作通用端口，需要设置串行通信操作设置寄存器 00 和 01（SCR00 和 SCR01）到默认状态(0087H)。此外，要清除端口输出模式寄存器 7(POM7)为 00H。

2.2.7 P80 至 P83（端口 8）

P80 至 P83 作为 4 位 I/O 端口的功能。这些引脚也具有比较器 0 和 1 的(+)端的输入电压、比较器 0 和 1 的(-)端的输入电压、定时器引脚 Hi-Z 控制输入、外部中断请求输入和可编程增益放大器输入的功能。

<R> P80 至 P83 引脚的数据输入，必须使用端口输入模式寄存器 8(PIM8)，以 1 位单元设置允许或禁止。可以指定下列操作模式为 1 位单元。

(1) 端口模式

P80 至 P83 作为 4 位 I/O 端口的功能。通过端口模式寄存器 8(PM8)，P80 至 P83 可以 1 位单元设置为输入或输出端口。

(2) 控制模式

P80 至 P83 具有比较器 0 和 1 的(+)端的输入电压、比较器 0 和 1 的(-)端的输入电压、定时器引脚 Hi-Z 控制输入、外部中断请求输入和可编程增益放大器输入的功能。

(a) CMP0P、CMP1P

这些是在比较器 0 和 1 的(+)端的输入电压引脚。

(b) CMP0M、CMP1M

这些是在比较器 0 和 1 的(-)端的输入电压引脚。

(c) TMOFF0、TMOFF1

这些是定时器引脚 Hi-Z 控制输入引脚。

(d) INTP3、INTP7

这些是外部中断请求输入引脚，用它可指定有效沿（上升沿、下降沿，或双沿）。

(e) PGAI

这是可编程增益放大器输入引脚。

2.2.8 P120 至 P124（端口 12）

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

可以指定下列操作模式为 1 位单元。

(1) 端口模式

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

P121 至 P124 作为 4 位输入端口功能。

(2) 控制模式

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

(a) INTP0

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

(b) EXLVI

这是外部低电压检测的电压输入引脚。

(c) X1、X2

这些引脚用于主系统时钟振荡器连接。

(d) XT1、XT2

这些引脚用于副系统时钟振荡器连接。

(e) EXCLK

这是用于主系统时钟的外部时钟输入引脚。

2.2.9 P150 和 P151（端口 15）

P150 和 P151 功能作为 2 位 I/O 端口。这些引脚也具有作为 A/D 转换器模拟输入的功能。

可以指定下列操作模式为 1 位单元。

(1) 端口模式

P150 和 P151 功能作为 2 位 I/O 端口。通过端口模式寄存器 15(PM15)，P150 和 P151 可以以 1 位单元设置为输入或输出端口。

(2) 控制模式

P150 和 P151 有作为 A/D 转换器模拟输入引脚(ANI8 至 ANI19)的功能。当这些引脚作为模拟输入引脚使用时，参见 12.6 (5) ANI0/P20 至 ANI7/P27 和 ANI8/P150。

注意事项 复位释放后，将 ANI8/P150 和 ANI9/P151 设置到数字输入（通用端口）模式。

2.2.10 AV_{REF}

这是 A/D 转换器和比较器参考电压输入引脚，也是 P20 至 P27、P150 至 P151、P80 至 P83、A/D 转换器、可编程增益放大器和比较器的正向电源供电引脚。

当端口 2、端口 15 和端口 8 的所有引脚用作模拟端口引脚时，设置 AV_{REF} 的电势为 $2.7\text{ V} \leq \text{AV}_{\text{REF}} \leq \text{V}_{\text{DD}}$ 。当端口 2、端口 15 和端口 8 的一根或多根引脚用作数字端口引脚，或当 A/D 转换器、可编程增益放大器和比较器未被使用时，使 AV_{REF} 与 V_{DD} 同电势。

2.2.12 AVss

A/D 转换器、可编程增益放大器、比较器、P20 至 P27、P150 至 P151、P80 至 P83 的地电势引脚。即使当 A/D 转换器、可编程增益放大器和比较器未被使用时，始终使该引脚与 Vss 同电势。

2.2.13 RESET

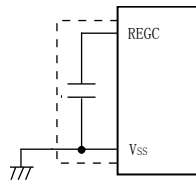
这是低有效系统复位输入引脚。

当外部复位引脚不使用时，直接连接此引脚到 VDD 或通过一个电阻连接。

2.2.14 REGC

这是用于连接内部操作的校准器输出(2.4V)稳定电容的引脚。通过一个电容器 (0.47~1 μF ：推荐) 连接该引脚到 Vss。然而，当使用的 STOP 模式从内部高速振荡时钟和外部主系统时钟运行起已经进入的话，推荐使用 0.47 μF 。

同时，使用性能良好的电容器，因为它用来稳定内部电压的。



注意事项 上图中虚线部分，请使用尽可能短的配线连接。

2.2.15 VDD

VDD 是用于除 P20 至 P27、P150 和 P80 至 P83 以外的端口引脚以及其它端口的正向电源引脚。

2.2.16 Vss

Vss 是用于除 P20 至 P27、P150 和 P80 至 P83 以外的端口引脚以及其它端口的地电位引脚。

2.2.17 FLMD0

此引脚用于设置 Flash 存储器编程模式。

请执行下列操作过程中的一个。

(a) 在普通操作模式下

普通操作期间，建议此引脚开路。

复位释放前 FLMD0 引脚必须保持 Vss 电位，但是不必由外部进行下拉，因为复位时会被内部下拉。但是，必须保持选择下拉（如 FLMDPUP = “0”，缺省值），通过使用背景事件控制寄存器(BECTL)中的位 7(FLMDPUP)（见 24.5 (1)背景事件控制寄存器）。要处于外部下拉状态，需要使用 200k Ω 或更小值的电阻。

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

(b) 在自编程模式下

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

在自编程模式下，设置在自编程库切换到上拉状态。

(c) 在 Flash 存储器编程模式下

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

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

2.3 引脚I/O电路和未使用引脚的推荐连接方式

2.3.1 44 引脚产品

表 2-3 为引脚 I/O 电路类型和未使用引脚的建议连接方式。

表 2-3. 未使用引脚的连接（1/3）

引脚名称	I/O 电路类型	I/O	未使用引脚的推荐连接方式
P10/TI02/TO02	8-R	I/O	输入： 通过电阻分别连接到 V _{DD} 或 V _{SS} 。 输出： 悬空。
P11/TI03/TO03			
P12/TI04/TO04			
P13/TI05/TO05			
P20/ANI0~P27/ANI7 ^注	11-G		输入： 通过一个电阻单独连接到 AV _{REF} 或者 AV _{SS} 。 输出： 悬空。
P30/SO10/TxD1/TO11	5-AG		输入： 通过电阻分别连接到 V _{DD} 或 V _{SS} 。 输出： 悬空。
P31/SI10/RxD1/SDA10/ INTP1/TI09	5-AN		<当 N 沟道漏极开路> 输出时 • 设置端口输出锁存器至 0： 悬空。 • 设置端口输出锁存器至 1： 通过一个电阻单独连接到 V _{DD} 或者 V _{SS} 。
P32/SCK10/SCL10/INTP2			
P40/TOOL0	8-R		<当片上调试允许时> 上拉此引脚(禁止下拉)。 <当片上调试禁止时> 输入： 通过电阻分别连接到 V _{DD} 或 V _{SS} 。 输出： 悬空。
P41/TOOL1			输入： 通过电阻分别连接到 V _{DD} 或 V _{SS} 。 输出： 悬空。
P50/TI06/TO06			
P51/TI07/TO07			
P52/SLTI/SLTO			

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

表 2-3. 未使用引脚的连接 (2/3)

引脚名称	I/O 电路类型	I/O	未使用引脚的推荐连接方式
P70/SO01/INTP4	8-R	I/O	输入： 通过电阻分别连接到 V _{DD} 或 V _{SS} 。 输出： 悬空。 <当 N 沟道漏极开路> 输出时 • 设置端口输出锁存器至 0： 悬空。 • 设置端口输出锁存器至 1： 通过一个电阻单独连接到 V _{DD} 或者 V _{SS} 。
P71/SI01/INTP5	5-AN		输入： 通过电阻分别连接到 V _{DD} 或 V _{SS} 。 输出： 悬空。
P72/SCK01/INTP6			输入： 通过电阻分别连接到 V _{DD} 或 V _{SS} 。 输出： 悬空。
P73/SO00/TxD0/TO10	8-R		<当 N 沟道漏极开路> 输出时 • 设置端口输出锁存器至 0： 悬空。 • 设置端口输出锁存器至 1： 通过一个电阻单独连接到 V _{DD} 或者 V _{SS} 。
P74/SI00/RxD0/TI10	5-AN		输入： 通过电阻分别连接到 V _{DD} 或 V _{SS} 。 输出： 悬空。
P75/SCK00/TI11			输入： 通过电阻分别连接到 V _{DD} 或 V _{SS} 。 输出： 悬空。 <当 N 沟道漏极开路> 输出时 • 设置端口输出锁存器至 0： 悬空。 • 设置端口输出锁存器至 1： 通过一个电阻单独连接到 V _{DD} 或者 V _{SS} 。
P80/CMP0P/TMOFF0/INTP3/PGAI	11-J		输入： 通过一个电阻单独连接到 AV _{REF} 或者 AV _{SS} 。 输出： 悬空。
P81/CMP0M	11-H		
P82/CMP1P/TOMOFF1/INTP7	11-I		
P83/CMP1M	11-H		
P120/INTP0/EXLVI	8-R		输入： 通过电阻分别连接到 V _{DD} 或 V _{SS} 。 输出： 悬空。
P121/X1 ^{‡1}	37-C	输入	通过电阻分别连接到 V _{DD} 或 V _{SS} 。
P122/X2/EXCLK ^{‡1}			
P123/XT1 ^{‡1}	37-B		
P124/XT2 ^{‡1}			
P150/ANI8, P151/ANI9 ^{‡2}	11-G	I/O	输入： 通过一个电阻单独连接到 AV _{REF} 或者 AV _{SS} 。 输出： 悬空。

注 1. 当这些引脚不使用时，在输入端口模式下，使用上面推荐的连接(参见 图 5-2 时钟操作模式控制寄存器 (CMC) 的格式)。

2. P150/ANI8 和 P151/ANI9 复位释放后设置为数字输入端口。

表 2-3. 未使用引脚的连接 (3/3)

引脚名称	I/O 电路类型	I/O	未使用引脚的推荐连接方式
AV _{REF}	—	—	<p><当 P20~P27、P150、P151 或 P80~P83 中一个或多个被设置为数字端口时> 使该引脚与 V_{DD} 同电势。</p> <p><当 P20~P27、P150、P151 和 P80~P83 全部被设置为模拟端口时> 使该引脚的电势为 $2.7\text{ V} \leq \text{AV}_{\text{REF}} \leq \text{V}_{\text{DD}}$。</p>
AV _{SS}	—	—	使得该引脚与 V _{SS} 同电势。
FLMD0	2-W	—	悬空或通过一个 100kΩ 或更大值的电阻连接到 V _{SS} 。
RESET	2	输入	直接或通过一个电阻连接到 V _{DD} 。
REGC	—	—	通过一个电容器(0.47~1 μF: 推荐)连接到 V _{SS} 。

图 2-1. 引脚 I/O 电路列表(1/3)

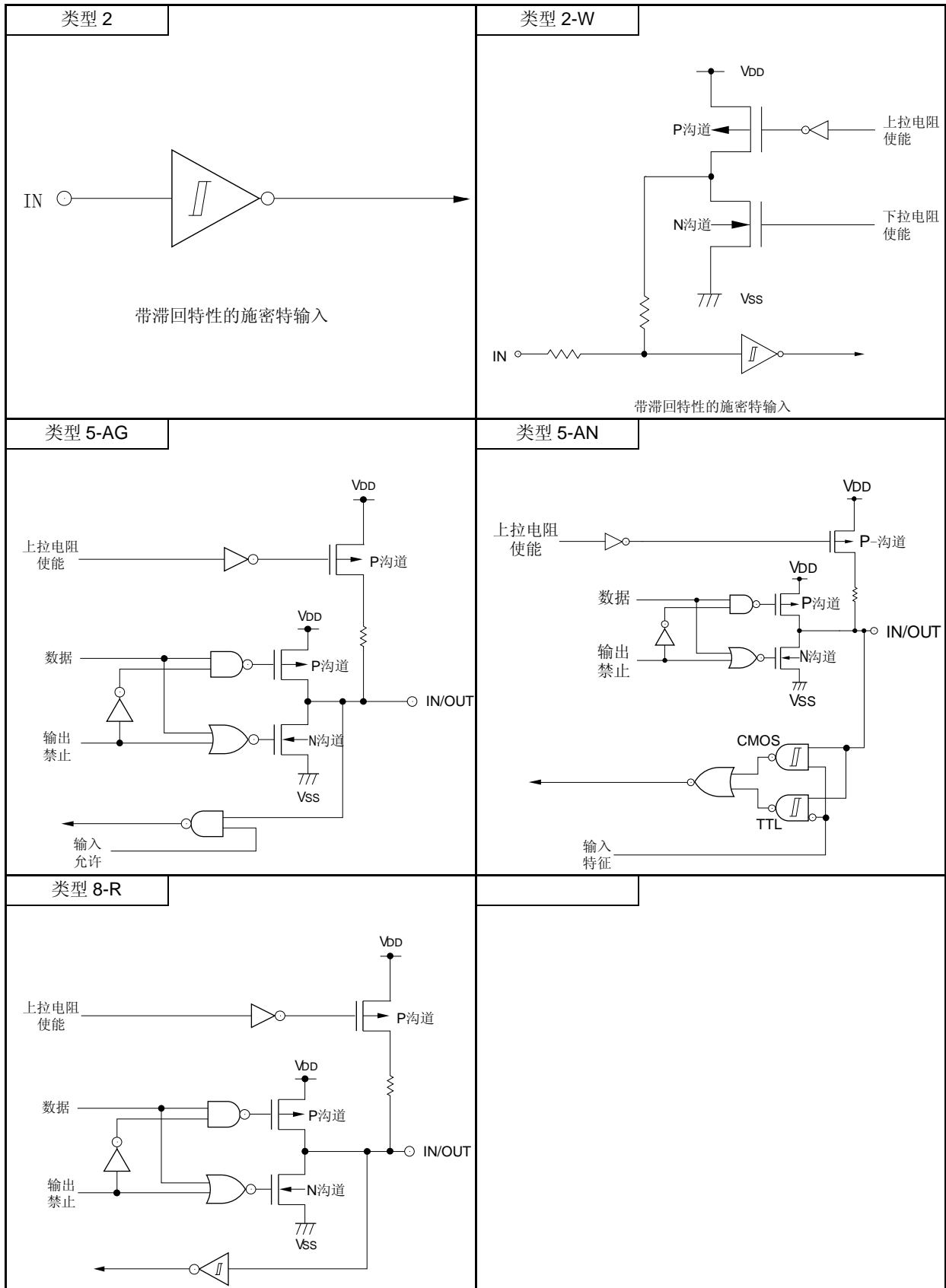


图 2-1. 引脚 I/O 电路列表(2/3)

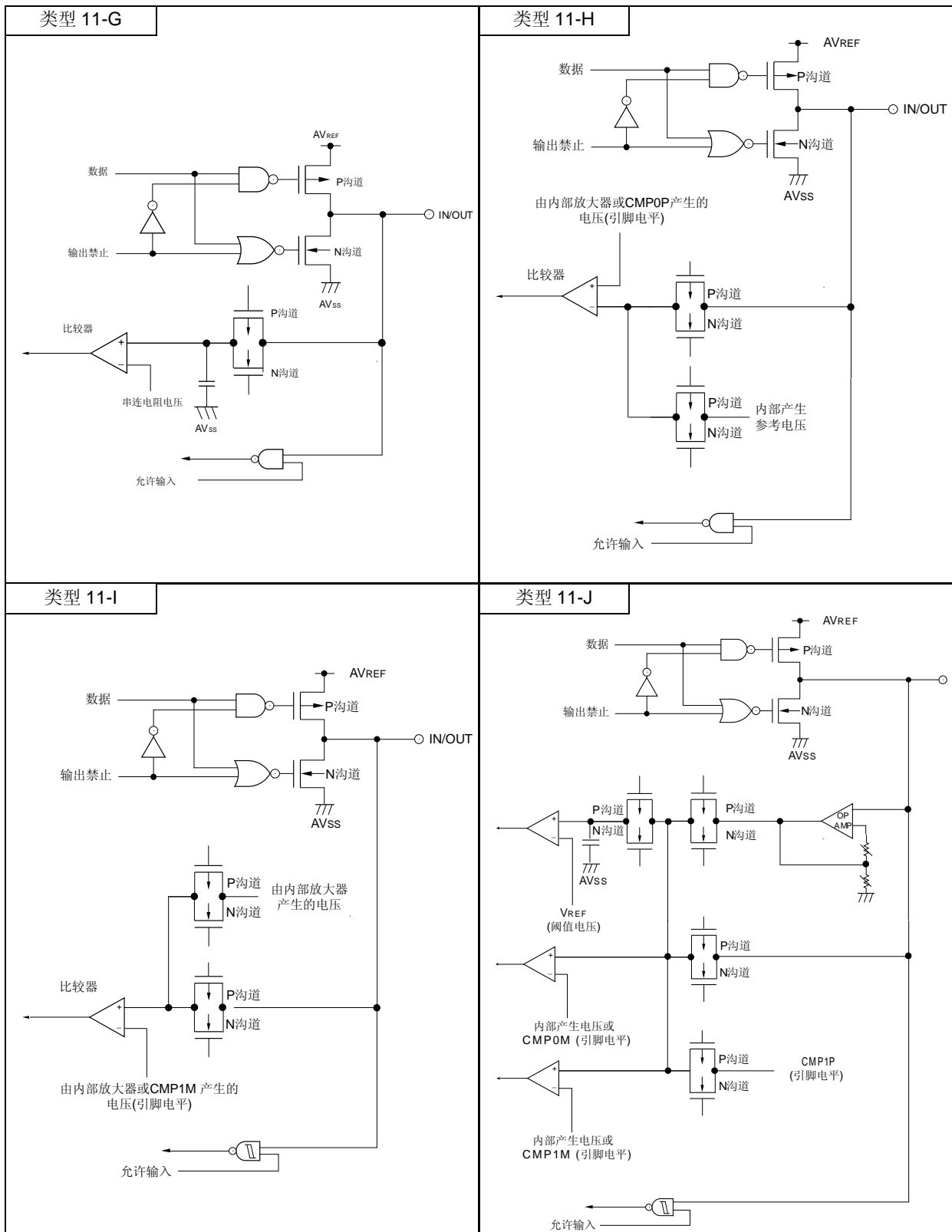
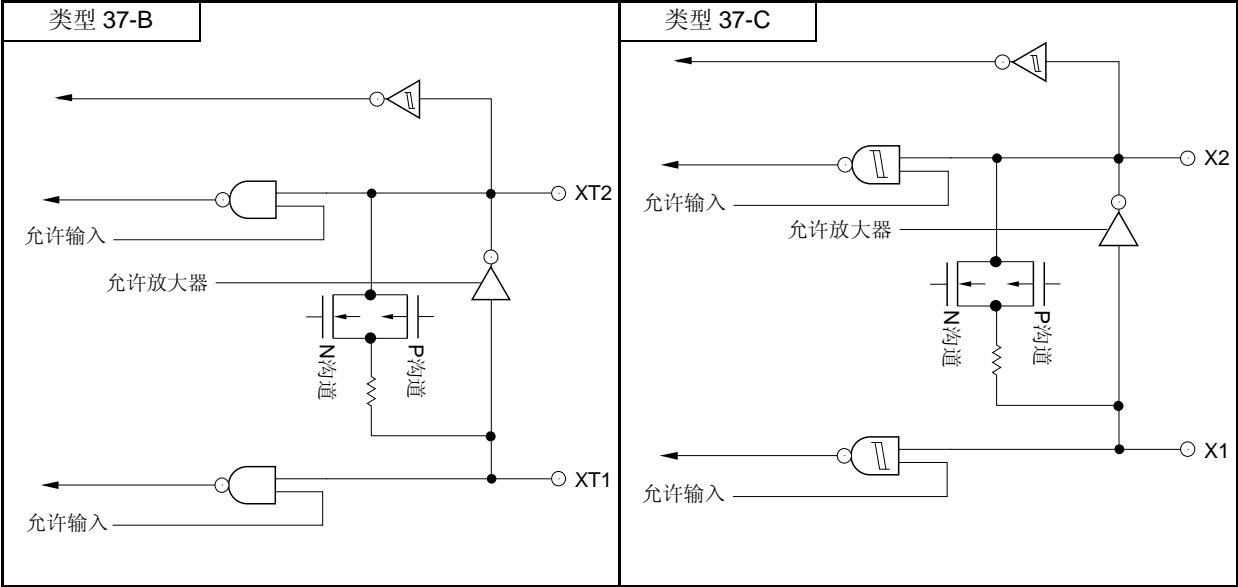


图 2-1. 引脚 I/O 电路列表(3/3)

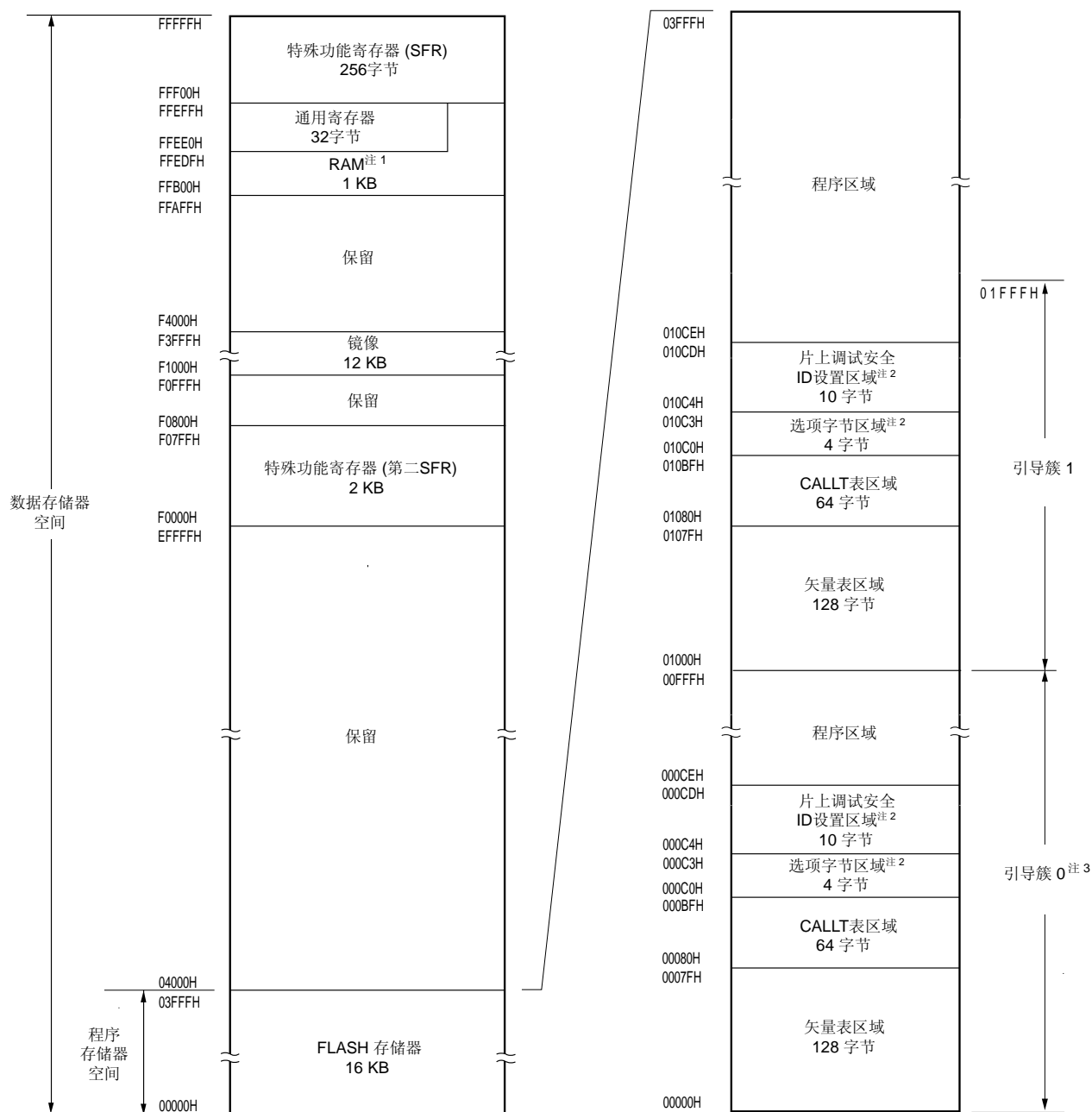


第3章 CPU 架构

3.1 存储器空间

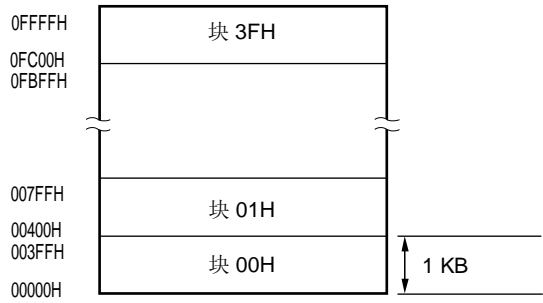
μPD79F9211 的产品可存取 1MB 的空间。图 3-1 显示了存储空间映射图。

图 3-1. 存储器映射图 (μPD79F9211)



- 注
1. 当使用自编程功能时，区域 FFE20H~FFEDFH 禁止使用。因为此区域用于自编程库。
 2. 当未使用引导交换时：设置选项字节到 000C0H~000C3H，及片上调试安全 ID 到 000C4H~000CDH。
当使用引导交换时：设置选项字节到 000C0H~000C3H 和 010C0H~010C3H，及片上调试安全 ID 到 000C4H~000CDH 和 010C4H~010CDH。
 3. 根据安全设置，可以禁止写入引导簇 0（参见 24.7 安全设置）。

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



Flash 存储器的地址值和块编号之间的关系如下所示。

表 3-1. Flash 存储器中地址值和块编号之间的对应关系

地址值		地址值		地址值		地址值	
00000H~003FFH	00H	04000H~043FFH	10H	08000H~083FFH	20H	0C000H~0C3FFH	30H
00400H~007FFH	01H	04400H~047FFH	11H	08400H~087FFH	21H	0C400H~0C7FFH	31H
00800H~00BFFH	02H	04800H~04BFFH	12H	08800H~08BFFH	22H	0C800H~0CBFFH	32H
00C00H~00FFFH	03H	04C00H~04FFFH	13H	08C00H~08FFFH	23H	0CC00H~0CFFFH	33H
01000H~013FFH	04H	05000H~053FFH	14H	09000H~093FFH	24H	0D000H~0D3FFH	34H
01400H~017FFH	05H	05400H~057FFH	15H	09400H~097FFH	25H	0D400H~0D7FFH	35H
01800H~01BFFH	06H	05800H~05BFFH	16H	09800H~09BFFH	26H	0D800H~0DBFFH	36H
01C00H~01FFFH	07H	05C00H~05FFFH	17H	09C00H~09FFFH	27H	0DC00H~0DFFFH	37H
02000H~023FFH	08H	06000H~063FFH	18H	0A000H~0A3FFH	28H	0E000H~0E3FFH	38H
02400H~027FFH	09H	06400H~067FFH	19H	0A400H~0A7FFH	29H	0E400H~0E7FFH	39H
02800H~02BFFH	0AH	06800H~06BFFH	1AH	0A800H~0ABFFH	2AH	0E800H~0EBFFH	3AH
02C00H~02FFFH	0BH	06C00H~06FFFH	1BH	0AC00H~0AFFFH	2BH	0EC00H~0EFFFH	3BH
03000H~033FFH	0CH	07000H~073FFH	1CH	0B000H~0B3FFH	2CH	0F000H~0F3FFH	3CH
03400H~037FFH	0DH	07400H~077FFH	1DH	0B400H~0B7FFH	2DH	0F400H~0F7FFH	3DH
03800H~03BFFH	0EH	07800H~07BFFH	1EH	0B800H~0BBFFH	2EH	0F800H~0FBFFH	3EH
03C00H~03FFFH	0FH	07C00H~07FFFH	1FH	0BC00H~0BFFFH	2FH	0FC00H~0FFFFH	3FH

备注 μ PD79F9211: 块编号 00H~0FH

3.1.1 内部程序存储器空间

内部程序存储空间用于存储程序和表格数据。

μPD79F9211 产品内部 ROM(Flash 存储器)的情况如下表所示。

表 3-2. 内部 ROM 容量

产品型号	内部 ROM	
	结构	容量
μPD79F9211	Flash 存储器	16384 × 8 位 (00000H~03FFFH)

内部程序存储空间被分为以下几个区域。

(1) 向量表区域

00000H 至 0007FH 的 128 字节保留为向量表区域。在向量表区域中存放的是用于分支复位或每个中断请求的产生的程序的起始地址。

在 16 位地址中，低 8 位存于偶地址中，高 8 位存于奇地址中。

表 3-3. 向量表

向量表地址	中断源	向量表地址	中断源
00000H	RESET 输入、POC、LVI、WDT、TRAP	0002CH	INTTM00
		0002EH	INTTM01
00004H	INTWDTI	00030H	INTTM02
00006H	INTLVI	00032H	INTTM03
00008H	INTP0	00034H	INTAD
0000AH	INTP1	00036H	INTRTC
0000CH	INTP2	00038H	INTRTCI
0000EH	INTP3/INTTMOFF0	0003CH	INTTMM0
00010H	INTP4	0003EH	INTTMV0
00012H	INTP5	00040H	INTMD
00014H	INTTMAD	00042H	INTTM04
00016H	INTCMP0	00044H	INTTM05
00018H	INTCMP1	00046H	INTTM06
0001AH	INTDMA0	00048H	INTTM07
0001CH	INTDMA1	0004AH	INTP6
0001EH	INTST0/INTCSI00	0004CH	INTP7/INTTMOFF1
00020H	INTST0/INTCSI01	0004EH	INTTMM1
00022H	INTSRE0	00050H	INTTMV1
00024H	INTST1/INTCSI10/INTIIC10	00052H	INTTM08
00026H	INTSR1	00054H	INTTM09
00028H	INTSRE1	00056H	INTTM10

(2) CALLT 指令表区域

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

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

(3) 选项字节区域

000C0H~000C3H 的 4 字节的区域可以用作选项字节区域。当使用引导交换功能时，在 010C0H~010C3H 设置选项字节。详细信息参考第 21 章 选项字节。

(4) 片上调试安全 ID 设置区域

000C4H~000CDH 和 010C4H~010CDH 的 10 字节区域可用作片上调试安全 ID 设置区域。不使用引导交换功能时，在 000C4H~000CDH 区域设置片上调试安全 ID。而当使用引导交换功能时，在 000C4H~000CDH 和 010C4H~010CDH 区域设置。 详细信息参考第 23 章 片上调试功能。

3.1.2 镜像区域

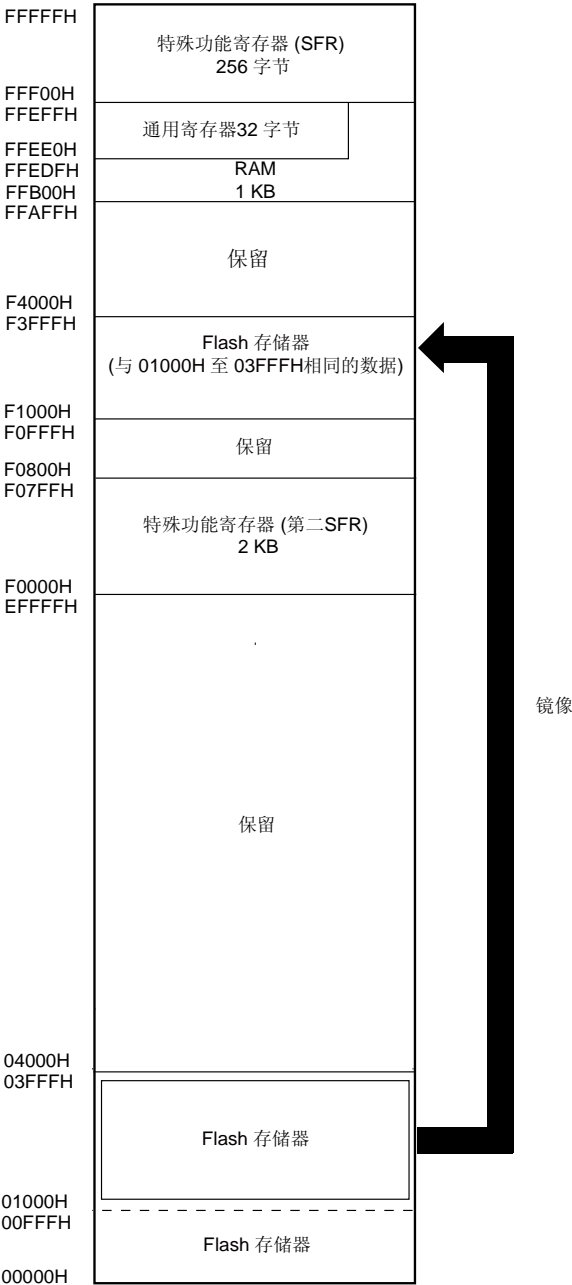
μPD79F9211 把 00000H~0FFFFH 的数据 Flash 区域，镜像到 F0000H~FFFFFFH (由处理器模式控制寄存器(PMC)设置)。

通过从 F0000H~FFFFFFH 读数据，数据 Flash 的内容可以用更短的代码读取。但是，数据 Flash 区域不能镜像到 SFR、扩展 SFR、RAM 和禁止使用的区域。

镜像区域只能读取，并且任何指令都不能从此区域中获取。

举例如下。

示例 μPD79F9211 (Flash 存储器: 16 KB, RAM: 1 KB)



PMC 寄存器描述如下。

- 处理器模式控制寄存器 (PMC)
此寄存器选择 Flash 存储器空间镜像区域 F0000H~FFFFFH。
PMC 只能由 1 位或 8 位存储器操作指令设置。
复位信号发生设置此寄存器为 00H。

图 3-5.处理器模式控制寄存器 (PMC)的配置格式

地址:	FFFFEH	复位后:	00H	R/W						
符号	7	6	5	4	3	2	1	<0>		
PMC	0	0	0	0	0	0	0	MAA		
MAA	从区域F0000H~FFFFFH中选择用于镜像的Flash存储器空间									
0	00000H~0FFFFFH镜像到F0000H~FFFFFH									
1	设置禁止									

- 注意事项
1. 务必将该寄存器的位 0 (MAA)设置为 0。
 2. 设置 PMC 后，等待至少一个指令周期并访问镜像区域。
 3. 在初始化设置期间，设置 PMC 优先于操作 DMA 控制器。在初始化设置期外，禁止重写 PMC。

3.1.3 内部数据存储器空间

μPD79F9211 产品包括如下 RAM。

表 3-4. 内部 RAM 容量

产品型号	内部 RAM
μPD79F9211	1024 × 8 位 (FEB00H~FFEFFH)

32 字节的区域 FFEE0H 至 FFEFFH 分配了四个通用寄存器 bank，每个寄存器 bank 包括 8 个 8 位寄存器。该区域可用作指令写入和执行的程序区域。但是，在通用寄存器中禁止执行指令。
内部高速 RAM 可用作堆栈存储器。

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

3.1.4 特殊功能寄存器(SFR)区域

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

注意事项 不要访问那些 SFR 未分配的地址区域。

3.1.5 扩展特殊功能寄存器（第二 SFR：第二 特殊功能寄存器）区域

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

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

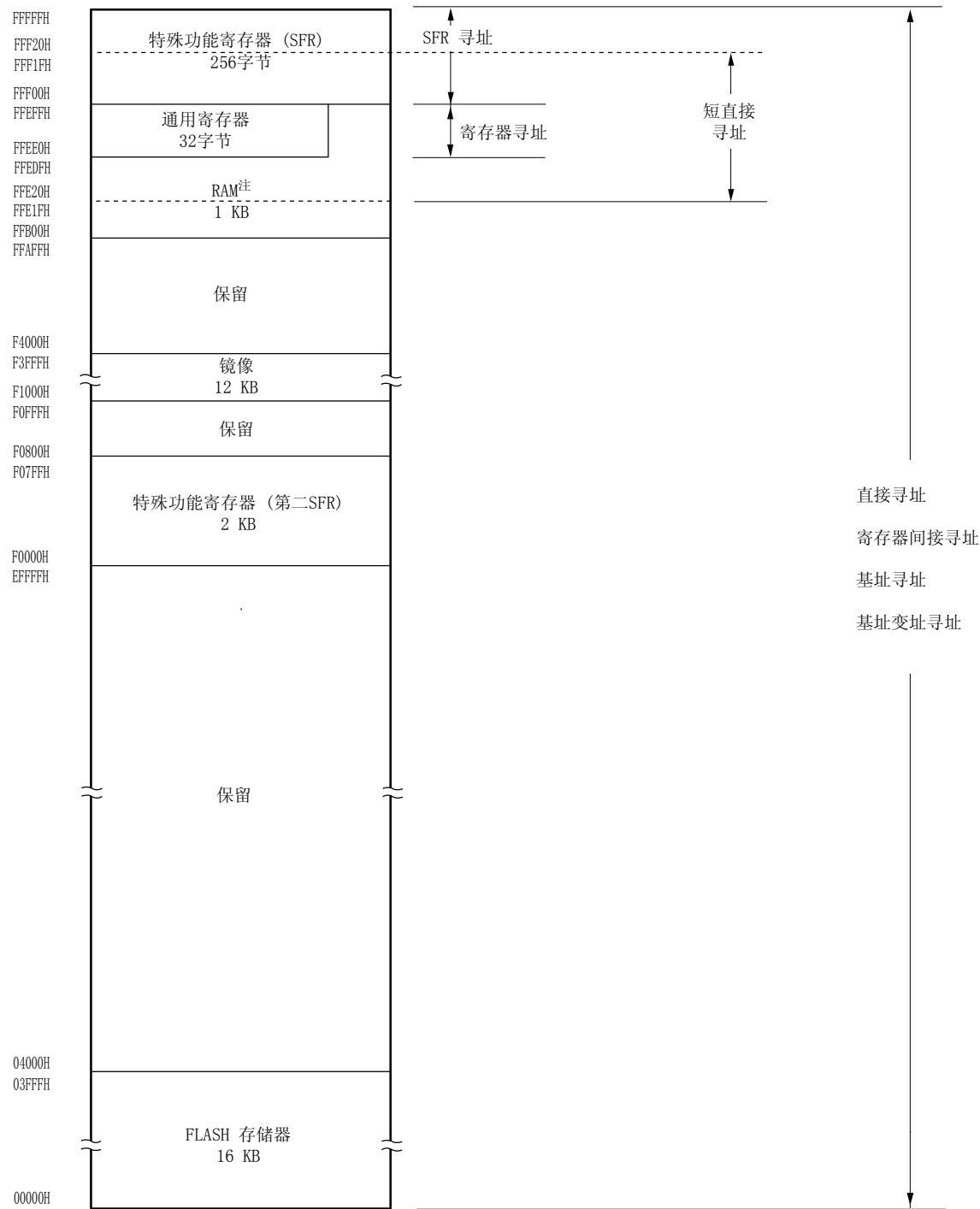
注意事项 不要访问扩展 SFR 没有分配的地址。

3.1.6 数据存储寻址

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

基于可操作性和其他考虑，在μPD79F9211 中提供了几种用于与指令执行相关的存储器寻址方式。因为有专门的数据存储区域，故可使用专为特殊功能寄存器(SFR)和通用寄存器功能设计的特殊寻址方式。图 3-6 和 3-9 显示了数据存储器和寻址之间的关系。

图 3-2. 数据存储器和选址之间的对应关系(μPD79F9211)



注 当使用自编程功能时，区域 FFE20H~FFE1FH 禁止使用。因为此区域用于自编程库。

3.2 处理器寄存器

μPD79F9211 产品包含以下几种处理器寄存器。

3.2.1 控制寄存器

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

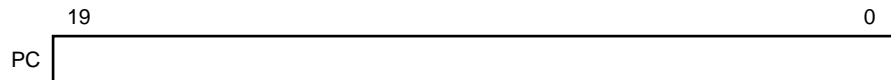
(1) 程序计数器 (PC)

程序计数器是一个 20 位的寄存器，用于存放下一条要执行程序地址信息。

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

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

图 3-3. 程序计数器的格式



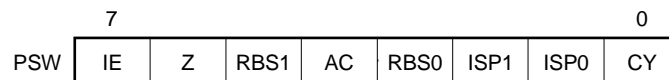
(2) 程序状态字 (PSW)

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

<R> 根据向量中断请求响应或者执行 PUSH PSW 指令，程序状态字的内容入栈；执行 RETB、RETI 和 POP PSW 指令，程序状态字的值自动恢复。

复位信号的产生将程序状态字的内容设置为 06H。

图 3-4. 程序状态字的格式



(a) 中断使能标志 (IE)

该标志用于控制 CPU 响应中断请求操作。

当 IE 为 0 时，表示禁止中断(DI)，即禁止所有可屏蔽中断请求。

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

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

(b) 零标志 (Z)

当运行结果为零时，该标志设置为(1)。其它情况该标志置 0。

(c) 寄存器 bank 选择标志 (RBS0, RBS1)

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

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

(d) 辅助进位标志 (AC)

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

(e) 优先服务标志(ISP1, ISP0)

该标志用来管理可屏蔽向量中断响应的优先级。通过优先级指定标志寄存器(PRn0L、PRn0H、PRn1L、PRn1H、PRn2L、PRn2H)向量中断请求指定低于 ISP0 和 ISP1 的值（参见 17.3 (3)）不能响应。实际的请求响应由中断使能标志 (IE) 控制。

备注 n = 0, 1

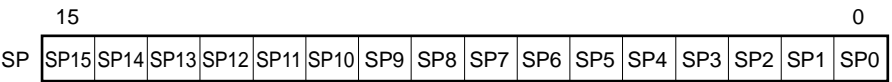
(f) 进位标志 (CY)

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

(3) 堆栈指针 (SP)

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

图 3-5. 堆栈指针的格式



在向堆栈写（存入）数据时，堆栈指针 SP 递减，而从堆栈中读出（恢复）数据时，堆栈指针累加。
堆栈的数据存储/恢复操作过程如图 3-13 所示。

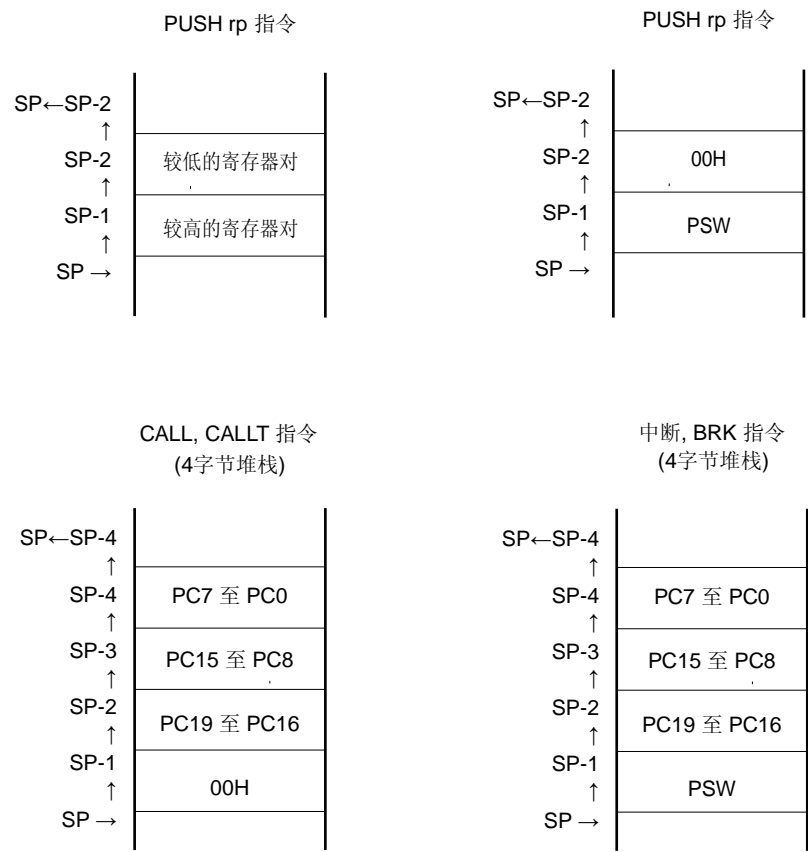
注意事项 由于复位信号的产生使 SP 的内容不确定，所以在使用堆栈前必须先对 SP 初始化。

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

<R>

<R>

图 3-6. 将数据存入堆栈存储器



3.2.2 通用寄存器

通用寄存器映射到数据存储器特定的地址空间为 FFE0H~FFEFH。通用寄存器共有四组，每一组由 8 个 8 位寄存器（X、A、C、B、E、D、L 和 H）组成。

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

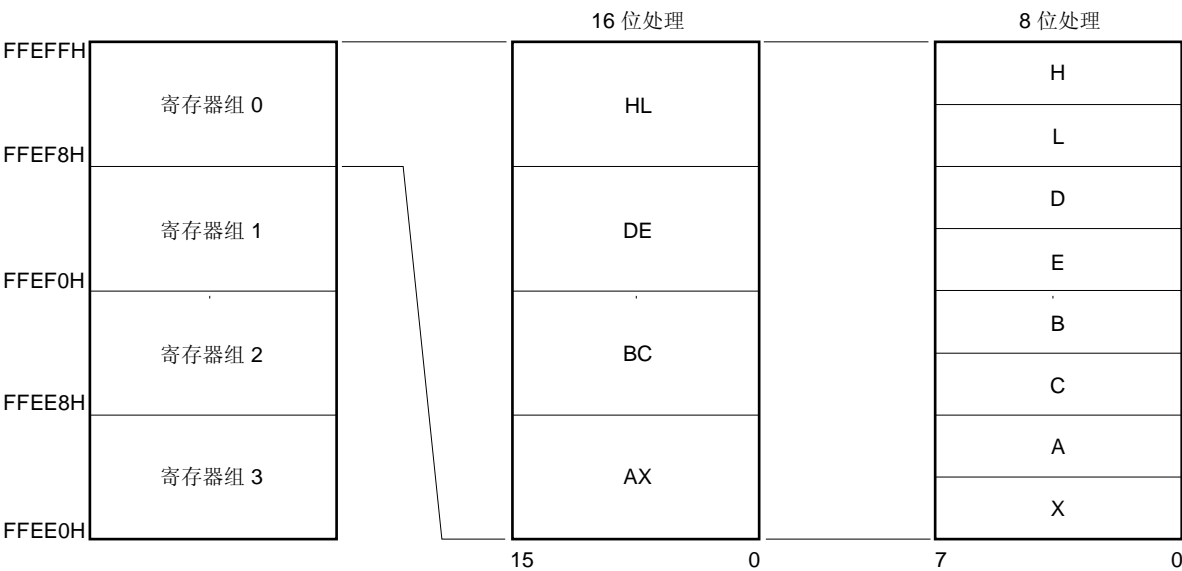
描述通用寄存器时，可以使用功能名称（X、A、C、B、E、D、L、H、AX、BC、DE 和 HL）或绝对名称（R0~R7 和 RP0~RP3）。

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

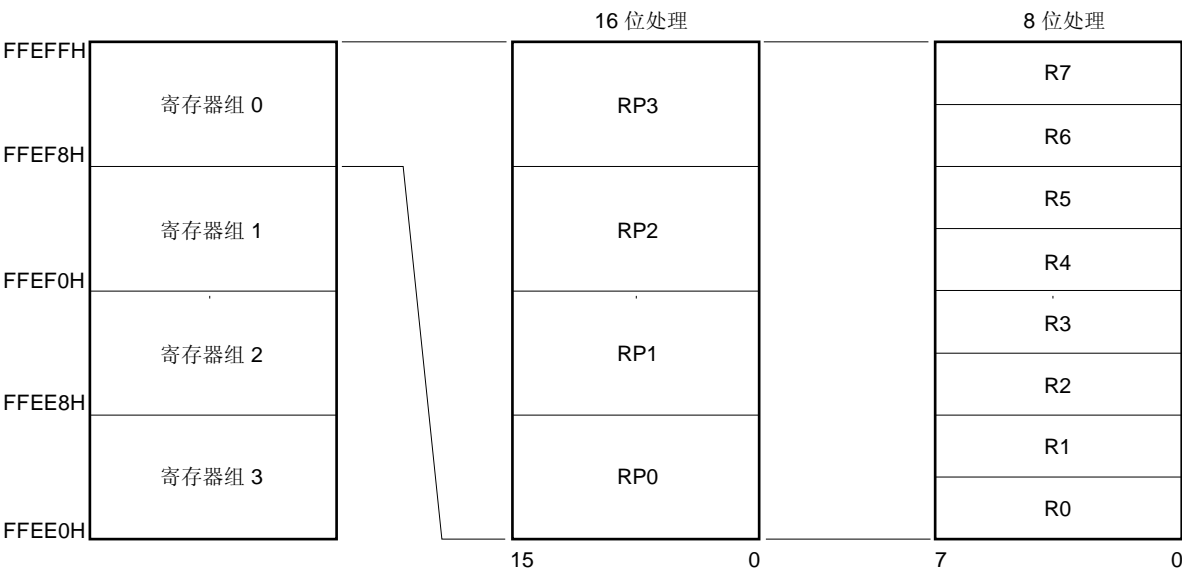
注意事项 禁止使用通用寄存器 (FFE0H~FFEFH) 空间获取指令或作为堆栈区域。

图 3-7. 通用寄存器的配置

(a) 功能名称



(b) 绝对名称



3.2.3 ES 和 CS 寄存器

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

图 3-8. ES 和 CS 寄存器的配置

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

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

3.2.4 特殊功能寄存器 (SFRs)

与通用寄存器不同，每个 SFR 都有特定的功能。

SFR 分配在 FFF00H~FFFFFH 区域。

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

每种位单元操作的描述如下。

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

表 3-5 为 SFR 列表。表中术语的含义如下。

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

注意事项 不要访问扩展 SFR 没有分配的地址。

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

表 3-5. SFR 列表 (1/4)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
FFF01H	端口寄存器 1	P1		R/W	√	√	—	00H
FFF02H	端口寄存器 2	P2		R/W	√	√	—	00H
FFF03H	端口寄存器 3	P3		R/W	√	√	—	00H
FFF04H	端口寄存器 4	P4		R/W	√	√	—	00H
FFF05H	端口寄存器 5	P5		R/W	√	√	—	00H
FFF07H	端口寄存器 7	P7		R/W	√	√	—	00H
FFF08H	端口寄存器 8	P8		R/W	√	√	—	00H
FFF0CH	端口寄存器 12	P12		R/W	√	√	—	未定义
FFF0FH	端口寄存器 15	P15		R/W	√	√	—	00H
FFF10H	串行数据寄存器 00	TXD0/ SIO00	SDR00	R/W	—	√	√	0000H
FFF11H		—			—	—		
FFF12H	串行数据寄存器 01	RXD0/ SIO01	SDR01	R/W	—	√	√	0000H
FFF13H		—			—	—		
FFF18H	定时器数据寄存器 00	TDR00		R/W	—	—	√	0000H
FFF19H								
FFF1AH	定时器数据寄存器 01	TDR01		R/W	—	—	√	0000H
FFF1BH								
FFF1EH	10 位 A/D 转换结果寄存器	ADCR		R	—	—	√	0000H
FFF1FH	8 位 A/D 转换结果寄存器	ADCRH		R	—	√	—	00H
FFF21H	端口模式寄存器 1	PM1		R/W	√	√	—	FFH
FFF22H	端口模式寄存器 2	PM2		R/W	√	√	—	FFH
FFF23H	端口模式寄存器 3	PM3		R/W	√	√	—	FFH
FFF24H	端口模式寄存器 4	PM4		R/W	√	√	—	FFH
FFF25H	端口模式寄存器 5	PM5		R/W	√	√	—	FFH
FFF27H	端口模式寄存器 7	PM7		R/W	√	√	—	FFH
FFF28H	端口模式寄存器 8	PM8		R/W	√	√	—	FFH
FFF2CH	端口模式寄存器 12	PM12		R/W	√	√	—	FFH
FFF2FH	端口模式寄存器 15	PM15		R/W	√	√	—	FFH
FFF30H	A/D 转换器模式寄存器	ADM		R/W	√	√	—	00H
FFF31H	模拟输入通道指定寄存器	ADS		R/W	√	√	—	00H
FFF38H	外部中断上升沿使能寄存器 0	EGP0		R/W	√	√	—	00H
FFF39H	外部中断下降沿使能寄存器 0	EGN0		R/W	√	√	—	00H
FFF3CH	输入切换控制寄存器	ISC		R/W	√	√	—	00H
FFF3EH	定时器输入选择寄存器 0	TIS0		R/W	√	√	—	00H
FFF42H	A/D 转换器模式寄存器 1	ADM1		R/W	√	√	—	00H

表 3-5. SFR 列表 (2/4)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
FFF44H	串行数据寄存器 02	TXD1/ SIO10	SDR02	R/W	—	√	√	0000H
FFF45H		—			—	—		
FFF46H	串行数据寄存器 03	RXD1	SDR03	R/W	—	√	√	0000H
FFF47H		—			—	—		
FFF64H	定时器数据寄存器 02	TDR02		R/W	—	—	√	0000H
FFF65H								
FFF66H	定时器数据寄存器 03	TDR03		R/W	—	—	√	0000H
FFF67H								
FFF68H	定时器数据寄存器 04	TDR04		R/W	—	—	√	0000H
FFF69H								
FFF6AH	定时器数据寄存器 05	TDR05		R/W	—	—	√	0000H
FFF6BH								
FFF6CH	定时器数据寄存器 06	TDR06		R/W	—	—	√	0000H
FFF6DH								
FFF6EH	定时器数据寄存器 07	TDR07		R/W	—	—	√	0000H
FFF6FH								
FFF70H	定时器数据寄存器 08	TDR08		R/W	—	—	√	0000H
FFF71H								
FFF72H	定时器数据寄存器 09	TDR09		R/W	—	—	√	0000H
FFF73H								
FFF74H	定时器数据寄存器 10	TDR10		R/W	—	—	√	0000H
FFF75H								
FFF76H	定时器数据寄存器 11	TDR11		R/W	—	—	√	0000H
FFF77H								
FFF90H	副计数寄存器	RSUBC		R	—	—	√	0000H
FFF91H								
FFF92H	秒计数 寄存器	SEC		R/W	—	√	—	00H
FFF93H	分钟计数 寄存器	MIN		R/W	—	√	—	00H
FFF94H	小时计数 寄存器	HOUR		R/W	—	√	—	12H [※]
FFF95H	星期 计数 寄存器	WEEK		R/W	—	√	—	00H
FFF96H	日期计数 寄存器	DAY		R/W	—	√	—	01H
FFF97H	月计数 寄存器	MONTH		R/W	—	√	—	01H
FFF98H	年计数 寄存器	YEAR		R/W	—	√	—	00H
FFF99H	钟表错误修正寄存器	SUBCUD		R/W	—	√	—	00H
FFF9AH	分钟报警寄存器	ALARMWM		R/W	—	√	—	00H
FFF9BH	小时报警寄存器	ALARMWH		R/W	—	√	—	12H
FFF9CH	星期报警寄存器	ALARMWW		R/W	—	√	—	00H

注 如果 AMPM 位 (RTCC0 寄存器的第 3 位) 复位后设置为 1, 此寄存器的值为 00H。

表 3-5. SFR 列表 (3/4)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
FFF9DH	实时计数器控制寄存器 0	RTCC0		R/W	√	√	—	00H
FFF9EH	实时计数器控制寄存器 1	RTCC1		R/W	√	√	—	00H
FFF9FH	实时计数器控制寄存器 2	RTCC2		R/W	√	√	—	00H
FFFA0H	时钟操作模式控制寄存器	CMC		R/W	—	√	—	00H
FFFA1H	时钟操作状态控制寄存器	CSC		R/W	√	√	—	C0H
FFFA2H	振荡稳定时间计数器状态寄存器	OSTC		R	√	√	—	00H
FFFA3H	振荡稳定时间选择寄存器	OSTS		R/W	—	√	—	07H
FFFA4H	时钟控制寄存器	CKC		R/W	√	√	—	09H
FFFA5H	时钟输出选择寄存器 0	CKS0		R/W	√	√	—	00H
FFFA8H	复位控制 标志寄存器	RESF		R	—	√	—	未定义 ^{注 1}
FFFA9H	低电压检测寄存器	LVIM		R/W	√	√	—	00H ^{注 2}
FFFAAH	低电压检测电平选择寄存器	LVIS		R/W	√	√	—	0EH ^{注 3}
FFFABH	看门狗定时器使能寄存器	WDTE		R/W	—	√	—	1A/9A ^{注 4}
FFFB0H	DMA SFR 地址寄存器 0	DSA0		R/W	—	√	—	00H
FFFB1H	DMA SFR 地址寄存器 1	DSA1		R/W	—	√	—	00H
FFFB2H	DMA RAM 地址寄存器 0L	DRA0L	DRA0	R/W	—	√	√	00H
FFFB3H	DMA RAM 地址寄存器 0H	DRA0H		R/W	—	√		00H
FFFB4H	DMA RAM 地址寄存器 1L	DRA1L	DRA1	R/W	—	√	√	00H
FFFB5H	DMA RAM 地址寄存器 1H	DRA1H		R/W	—	√		00H
FFFB6H	DMA 字节计数寄存器 0L	DBC0L	DBC0	R/W	—	√	√	00H
FFFB7H	DMA 字节计数寄存器 0H	DBC0H		R/W	—	√		00H
FFFB8H	DMA 字节计数寄存器 1L	DBC1L	DBC1	R/W	—	√	√	00H
FFFB9H	DMA 字节计数寄存器 1H	DBC1H		R/W	—	√		00H
FFFB AH	DMA 模式控制寄存器 0	DMC0		R/W	√	√	—	00H
FFFB BH	DMA 模式控制寄存器 1	DMC1		R/W	√	√	—	00H
FFFB CH	DMA 操作控制寄存器 0	DRC0		R/W	√	√	—	00H
FFFB DH	DMA 操作控制寄存器 1	DRC1		R/W	√	√	—	00H
FFFB EH	背景事件控制 寄存器	BECTL		R/W	√	√	—	00H
FFFC0H	—	PFCMD ^{注 5}		—	—	—	—	未定义
FFFC2H	—	PFS ^{注 5}		—	—	—	—	未定义
FFFC4H	—	FLPMC ^{注 5}		—	—	—	—	未定义
FFFD0H	中断请求标志寄存器 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	中断请求标志寄存器 2H	IF2H		R/W	√	√		00H
FFFD4H	中断屏蔽标志寄存器 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	中断屏蔽标志寄存器 2H	MK2H		R/W	√	√		FFH
FFFD8H	优先级指定标志寄存器 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	优先级指定标志寄存器 02H	PR02H		R/W	√	√		FFH
FFFD CH	优先级指定标志寄存器 12L	PR12L	PR12	R/W	√	√	√	FFH
FFFD DH	优先级指定标志寄存器 12H	PR12H		R/W	√	√		FFH

- 注
1. RESF 的复位值根据复位源不同而不同。
 2. LVIM 的复位值根据复位源和选项字节的设置的不同而不同。
 3. LVIS 的复位值根据复位源不同而不同。
 4. WDTE 的复位值由选项字节的设置决定。
 5. 不要直接对该 SFR 进行操作，因为其用于自编程库。

表 3-5. SFR 列表 (4/4)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
FFFE0H	中断请求标志寄存器 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	中断请求标志寄存器 0H	IF0H		R/W	√	√		00H
FFFE2H	中断请求标志寄存器 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	中断请求标志寄存器 1H	IF1H		R/W	√	√		00H
FFFE4H	中断屏蔽标志寄存器 0L	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H	中断屏蔽标志寄存器 0H	MK0H		R/W	√	√		FFH
FFFE6H	中断屏蔽标志寄存器 1L	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H	中断屏蔽标志寄存器 1H	MK1H		R/W	√	√		FFH
FFFE8H	优先级指定标志寄存器 00L	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H	优先级指定标志寄存器 00H	PR00H		R/W	√	√		FFH
FFFEAH	优先级指定标志寄存器 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFEBH	优先级指定标志寄存器 01H	PR01H		R/W	√	√		FFH
FFFECH	优先级指定标志寄存器 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFEDH	优先级指定标志寄存器 10H	PR10H		R/W	√	√		FFH
FFFEEH	优先级指定标志寄存器 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	优先级指定标志寄存器 11H	PR11H		R/W	√	√		FFH
FFFF0H	乘法/除法数据寄存器 A(L)	MDAL/MULA		R/W	—	—	√	0000H
FFFF1H								
FFFF2H	乘法/除法数据寄存器 A(H)	MDAH/MULB		R/W	—	—	√	0000H
FFFF3H								
FFFF4H	乘法/除法数据寄存器 B(H)	MDBH/MULOH		R/W	—	—	√	0000H
FFFF5H								
FFFF6H	乘法/除法数据寄存器 B(L)	MDBL/MULOL		R/W	—	—	√	0000H
FFFF7H								
FFFFEH	处理器模式控制寄存器	PMC		R/W	√	√	—	00H

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

3.2.5 扩展特殊功能寄存器（第二SFR：第二特殊功能寄存器）

与通用寄存器不同，每个扩展 SFR（第二 SFR）都有特殊的功能。

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

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

每种位单元操作的描述如下。

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

表 3-6 为扩展 SFR 列表。表中术语的含义如下。

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

注意事项 不要访问扩展 SFR 没有分配的地址。

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

表 3-6. 扩展 SFR (第二 SFR) 列表 (1/5)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作位单元			复位后	
				1 位	8 位	16 位		
F0017H	A/D 端口配置寄存器	ADPC	R/W	–	√	–	10H	
F0031H	上拉电阻选项寄存器 1	PU1	R/W	√	√	–	00H	
F0033H	上拉电阻选项寄存器 3	PU3	R/W	√	√	–	00H	
F0034H	上拉电阻选项寄存器 4	PU4	R/W	√	√	–	00H	
F0035H	上拉电阻选项寄存器 5	PU5	R/W	√	√	–	00H	
F0037H	上拉电阻选项寄存器 7	PU7	R/W	√	√	–	00H	
F003CH	上拉电阻选项寄存器 12	PU12	R/W	√	√	–	00H	
F0043H	端口输入模式寄存器 3	PIM3	R/W	√	√	–	00H	
F0047H	端口输入模式寄存器 7	PIM7	R/W	√	√	–	00H	
F0048H	端口输入模式寄存器 8	PIM8	R/W	√	√	–	00H	
F0053H	端口输出模式寄存器 3	POM3	R/W	√	√	–	00H	
F0057H	端口输出模式寄存器 7	POM7	R/W	√	√	–	00H	
F0060H	噪声滤波器使能寄存器 0	NFEN0	R/W	√	√	–	00H	
F0061H	噪声滤波器使能寄存器 1	NFEN1	R/W	√	√	–	00H	
F0062H	噪声滤波器使能寄存器 2	NFEN2	R/W	√	√	–	00H	
F00E0H	乘法/除法数据寄存器 C (L)	MDCL	R	–	–	√	0000H	
F00E2H	乘法/除法数据寄存器 C (H)	MDCH	R	–	–	√	0000H	
F00E8H	乘法/除法控制寄存器	MDUC	R/W	√	√	–	00H	
F00F0H	外围设备使能寄存器 0	PER0	R/W	√	√	–	00H	
F00F1H	外围设备使能寄存器 1	PER1	R/W	√	√	–	00H	
F00F2H	外围设备使能寄存器 2	PER2	R/W	√	√	–	00H	
F00F3H	操作速度模式控制寄存器	OSMC	R/W	–	√	–	00H	
F00F4H	校准器模式控制寄存器	RMC	R/W	–	√	–	00H	
F00F6H	40 MHz 内部高速振荡控制寄存器	DSCCTL	R/W	√	√	–	00H	
F00FEH	BCD 调整结果寄存器	BCDADJ	R	–	√	–	00H	
F0100H	串行状态寄存器 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–		–	–			
F0102H	串行状态寄存器 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–		–	–			
F0104H	串行状态寄存器 02	SSR02L	SSR02	R	–	√	√	0000H
F0105H		–		–	–			
F0106H	串行状态寄存器 03	SSR03L	SSR03	R	–	√	√	0000H
F0107H		–		–	–			
F0108H	串行标志清零触发寄存器 00	SIR00L	SIR00	R/W	–	√	√	0000H
F0109H		–		–	–			
F010AH	串行标志清零触发寄存器 01	SIR01L	SIR01	R/W	–	√	√	0000H
F010BH		–		–	–			
F010CH	串行标志清零触发寄存器 02	SIR02L	SIR02	R/W	–	√	√	0000H
F010DH		–		–	–			
F010EH	串行标志清零触发寄存器 03	SIR03L	SIR03	R/W	–	√	√	0000H
F010FH		–		–	–			

表 3-6. 扩展 SFR（第二 SFR）列表 (2/5)

地址	特殊功能寄存器 (SFR) 名称	符号		R/W	可操作位单元			复位后
					1 位	8 位	16 位	
F0110H	串行模式寄存器 00	SMR00		R/W	—	—	√	0020H
F0111H								
F0112H	串行模式寄存器 01	SMR01		R/W	—	—	√	0020H
F0113H								
F0114H	串行模式寄存器 02	SMR02		R/W	—	—	√	0020H
F0115H								
F0116H	串行模式寄存器 03	SMR03		R/W	—	—	√	0020H
F0117H								
F0118H	串行通信操作设置寄存器 00	SCR00		R/W	—	—	√	0087H
F0119H								
F011AH	串行通信操作设置寄存器 01	SCR01		R/W	—	—	√	0087H
F011BH								
F011CH	串行通信操作设置寄存器 02	SCR02		R/W	—	—	√	0087H
F011DH								
F011EH	串行通信操作设置寄存器 03	SCR03		R/W	—	—	√	0087H
F011FH								
F0120H	串行通道使能状态寄存器 0	SE0L	SE0	R	√	√	√	0000H
F0121H		—			—	—		
F0122H	串行通道开始触发寄存器 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		—			—	—		
F0124H	串行通道停止触发寄存器 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		—			—	—		
F0126H	串行时钟选择寄存器 0	SPS0L	SPS0	R/W	—	√	√	0000H
F0127H		—			—	—		
F0128H	串行输出寄存器 0	SO0		R/W	—	—	√	0F0FH
F0129H								
F012AH	串行输出使能寄存器 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		—			—	—		
F0134H	串行输出电平寄存器 0	SOL0L	SOL0	R/W	—	√	√	0000H
F0135H		—			—	—		
F0180H	定时器计数器寄存器 00	TCR00		R	—	—	√	FFFFH
F0181H								
F0182H	定时器计数器寄存器 01	TCR01		R	—	—	√	FFFFH
F0183H								
F0184H	定时器计数器寄存器 02	TCR02		R	—	—	√	FFFFH
F0185H								
F0186H	定时器计数器寄存器 03	TCR03		R	—	—	√	FFFFH
F0187H								
F0188H	定时器计数器寄存器 04	TCR04		R	—	—	√	FFFFH
F0189H								
F018AH	定时器计数器寄存器 05	TCR05		R	—	—	√	FFFFH
F018BH								
F018CH	定时器计数器寄存器 06	TCR06		R	—	—	√	FFFFH
F018DH								

表 3-6. 扩展 SFR (第二 SFR) 列表 (3/5)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作位单元			复位后
				1 位	8 位	16 位	
F018EH	定时器计数器寄存器 07	TCR07	R	—	—	√	FFFFH
F018FH							
F0190H	定时器模式寄存器 00	TMR00	R/W	—	—	√	0000H
F0191H							
F0192H	定时器模式寄存器 01	TMR01	R/W	—	—	√	0000H
F0193H							
F0194H	定时器模式寄存器 02	TMR02	R/W	—	—	√	0000H
F0195H							
F0196H	定时器模式寄存器 03	TMR03	R/W	—	—	√	0000H
F0197H							
F0198H	定时器模式寄存器 04	TMR04	R/W	—	—	√	0000H
F0199H							
F019AH	定时器模式寄存器 05	TMR05	R/W	—	—	√	0000H
F019BH							
F019CH	定时器模式寄存器 06	TMR06	R/W	—	—	√	0000H
F019DH							
F019EH	定时器模式寄存器 07	TMR07	R/W	—	—	√	0000H
F019FH							
F01A0H	定时器状态寄存器 00	TSR00	R	—	—	√	0000H
F01A1H							
F01A2H	定时器状态寄存器 01	TSR01	R	—	—	√	0000H
F01A3H							
F01A4H	定时器状态寄存器 02	TSR02	R	—	—	√	0000H
F01A5H							
F01A6H	定时器状态寄存器 03	TSR03	R	—	—	√	0000H
F01A7H							
F01A8H	定时器状态寄存器 04	TSR04	R	—	—	√	0000H
F01A9H							
F01AAH	定时器状态寄存器 05	TSR05	R	—	—	√	0000H
F01ABH							
F01ACH	定时器状态寄存器 06	TSR06	R	—	—	√	0000H
F01ADH							
F01AEH	定时器状态寄存器 07	TSR07	R	—	—	√	0000H
F01AFH							
F01B0H	定时器通道允许状态寄存器 0	TE0	R	—	—	√	0000H
F01B1H							
F01B2H	定时器通道开始寄存器 0	TS0	R/W	—	—	√	0000H
F01B3H							
F01B4H	定时器通道停止寄存器 0	TT0	R/W	—	—	√	0000H
F01B5H							
F01B6H	定时器时钟选择寄存器 0	TPS0	R/W	—	—	√	0000H
F01B7H							
F01B8H	定时器输出寄存器 0	TO0	R/W	—	—	√	0000H
F01B9H							

表 3-6. 扩展 SFR（第二 SFR）列表 (4/5)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作位单元			复位后
				1 位	8 位	16 位	
F01BAH	定时器输出允许寄存器 0	TOE0	R/W	–	–	√	0000H
F01BBH							
F01BCH	定时器输出电平寄存器 0	TOL0	R/W	–	–	√	0000H
F01BDH							
F01BEH	定时器输出模式寄存器 0	TOM0	R/W	–	–	√	0000H
F01BFH							
F01C0H	定时器计数器寄存器 08	TCR08	R	–	–	√	FFFFH
F01C1H							
F01C2H	定时器计数器寄存器 09	TCR09	R	–	–	√	FFFFH
F01C3H							
F01C4H	定时器计数器寄存器 10	TCR10	R	–	–	√	FFFFH
F01C5H							
F01C6H	定时器计数器寄存器 11	TCR11	R	–	–	√	FFFFH
F01C7H							
F01C8H	定时器模式寄存器 08	TMR08	R/W	–	–	√	0000H
F01C9H							
F01CAH	定时器模式寄存器 09	TMR09	R/W	–	–	√	0000H
F01CBH							
F01CCH	定时器模式寄存器 10	TMR10	R/W	–	–	√	0000H
F01CDH							
F01CEH	定时器模式寄存器 11	TMR11	R/W	–	–	√	0000H
F01CFH							
F01D0H	定时器状态寄存器 08	TSR08	R	–	–	√	0000H
F01D1H							
F01D2H	定时器状态寄存器 09	TSR09	R	–	–	√	0000H
F01D3H							
F01D4H	定时器状态寄存器 10	TSR10	R	–	–	√	0000H
F01D5H							
F01D6H	定时器状态寄存器 11	TSR11	R	–	–	√	0000H
F01D7H							
F01E8H	定时器三角波输出模式寄存器 0	TOT0	R/W	–	–	√	0000H
F01EAH	定时器实时输出启用寄存器 0	TRE0	R/W	–	–	√	0000H
F01ECH	定时器实时输出寄存器 0	TRO0	R/W	–	–	√	0000H
F01EEH	定时器实时控制寄存器 0	TRC0	R/W	–	–	√	0000H
F01F0H	定时器模组输出启用寄存器 0	TME0	R/W	–	–	√	0000H
F01F2H	定时器延时输出启用寄存器 0	TDE0	R/W	–	–	√	0000H
F0220H	TAU 选择模式寄存器	OPMR	R/W	–	–	√	0000H
F0222H	TAU 选择状态寄存器	OPSR	R	–	–	√	0000H
F0224H	TAU 选择高阻启动触发寄存器	OPHS	R/W	–	–	√	0000H
F0226H	TAU 选择高阻停止触发寄存器	OPHT	R/W	–	–	√	0000H

表 3-6. 扩展 SFR（第二 SFR）列表 (5/5)

地址	特殊功能寄存器 (SFR) 名称	符号	R/W	可操作位单元			复位后
				1 位	8 位	16 位	
F0240H	可编程增益放大器控制寄存器	OAM	R/W	√	√	–	00H
F0241H	比较器 0 控制寄存器	C0CTL	R/W	√	√	–	00H
F0242H	比较器 0 内部参考电压设定寄存器	C0RVM	R/W	√	√	–	00H
F0243H	比较器 1 控制寄存器	C1CTL	R/W	√	√	–	00H
F0244H	比较器 1 内部参考电压设定寄存器	C1RVM	R/W	√	√	–	00H

备注 对于 SFR 区域中的 SFR，参见表 3-5 SFR 列表。

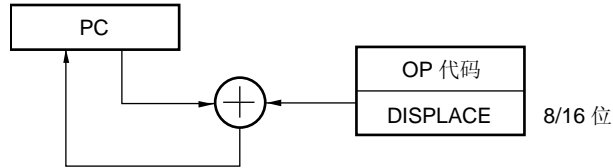
3.3 指令地址寻址

3.3.1 相对寻址

[功能]

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

图 3-9. 相对寻址的图示



3.3.2 立即寻址

[功能]

将指令字中的立即数赋给程序计数器(PC), 然后将该地址作为分支地址。

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

图 3-10. CALL !!addr20 或 BR !!addr20 的例子

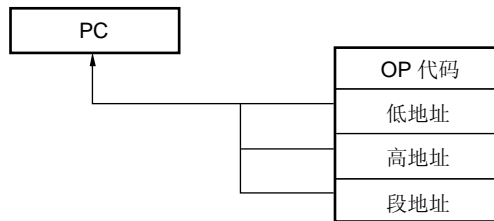
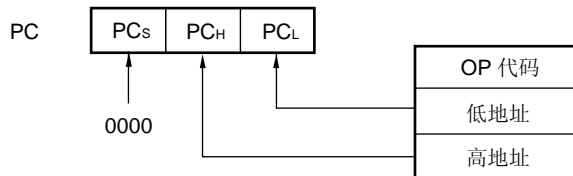


图 3-11. CALL !addr16 或 BR !addr16 的例子



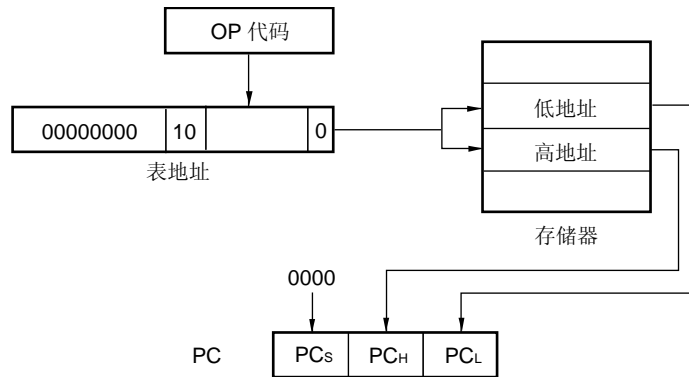
3.3.3 表间接寻址

[功能]

表间接寻址是以指令字中的 5 位立即数，指定 CALLT 表区域(0080H~00BFH)的表地址，将该内容和程序计数器 (PC)中的下一个地址作为 16 位数据存储在表地址，并指定程序地址。表间接寻址仅适用于 CALLT 指令。

在 UPD79F9211 微控制器中，仅允许在从 00000H~0FFFFH 的 64 KB 空间中分支。

图 3-12. 表间接寻址的图示

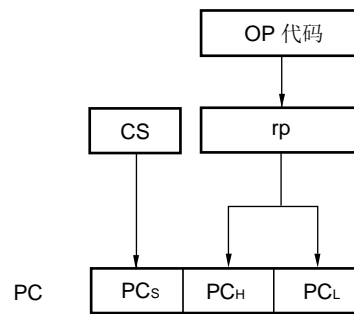


3.3.4 寄存器直接寻址

[功能]

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

图 3-13. 寄存器直接寻址的图示



3.4 用于处理数据地址的寻址

3.4.1 隐含寻址

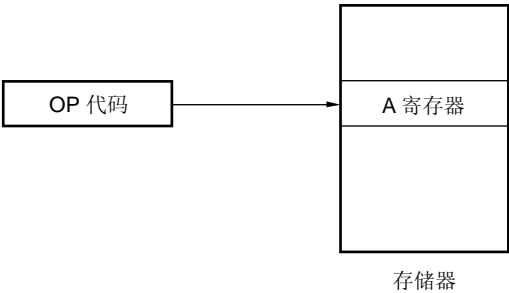
[功能]

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

[操作数格式]

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

图 3-14. 隐含寻址图示



3.4.2 寄存器地址

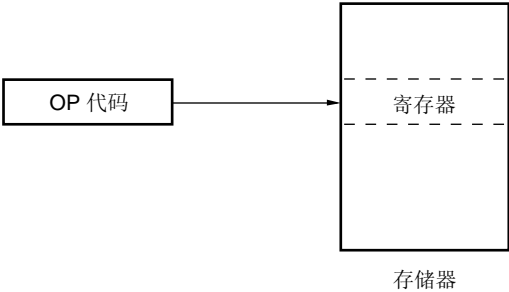
[功能]

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

[操作数格式]

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

图 3-15. 寄存器寻址图示



3.4.3 直接寻址

[功能]

将指令字中的立即数作为操作数地址，直接指定目标地址。

[操作数格式]

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

图 3-16. ADDR16 的示例

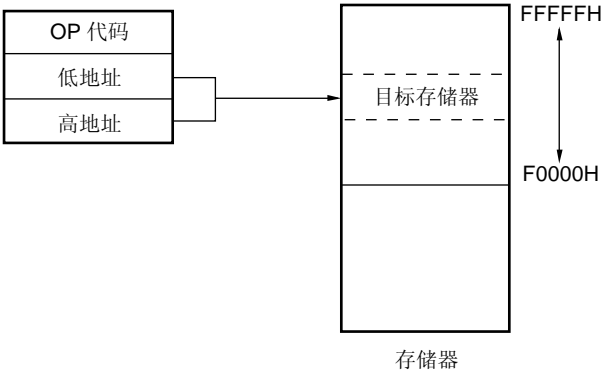
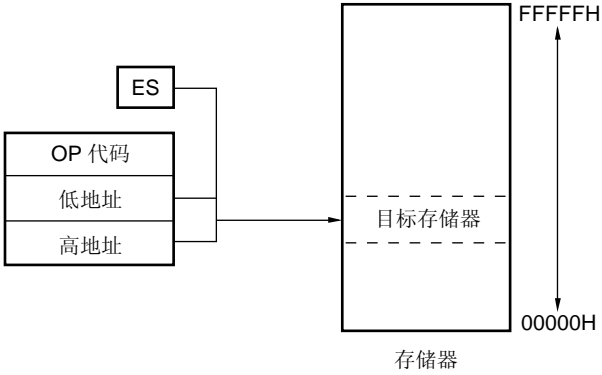


图 3-17. ES:ADDR16 的示例



3.4.4 短直接寻址

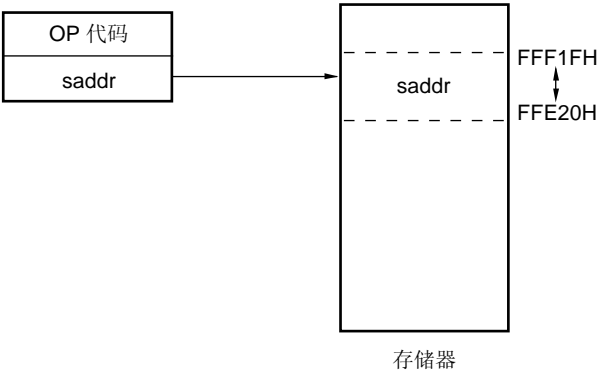
[功能]

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

[操作数格式]

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

图 3-18. 短直接寻址图示



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

3.4.5 SFR寻址

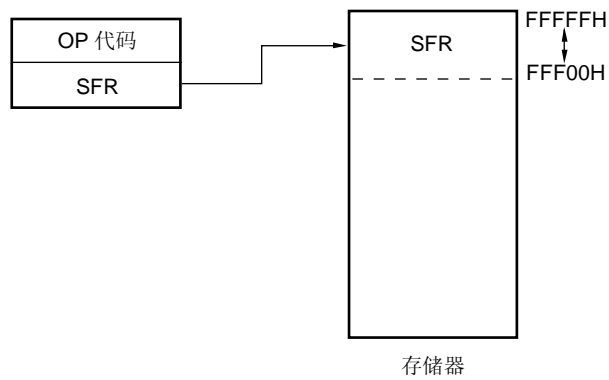
[功能]

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

[操作数格式]

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

图 3-19. SFR 寻址图示



3.4.6 寄存器间接寻址

[功能]

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

[操作数格式]

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

图 3-20. [DE]、[HL]的示例

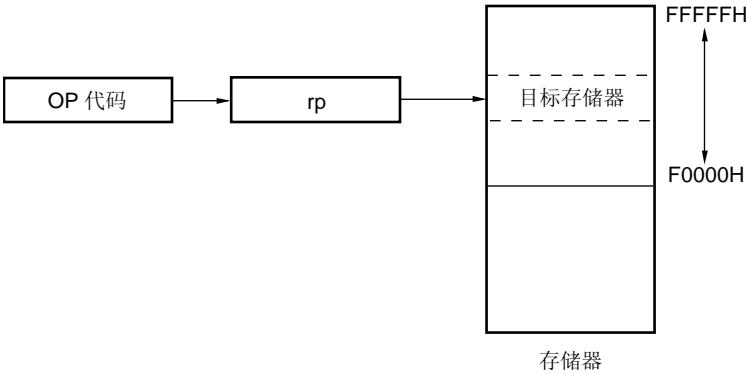
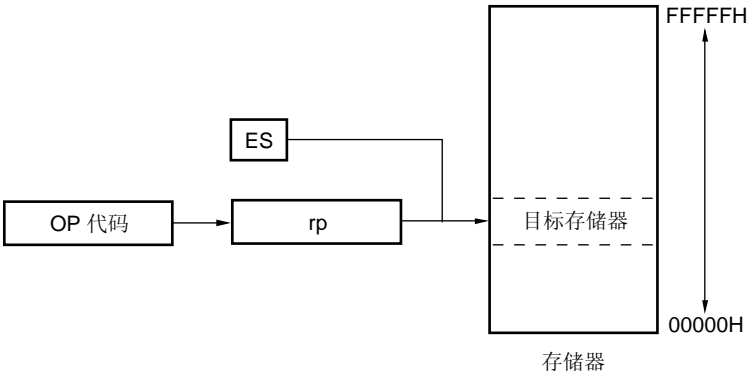


图 3-21. ES:[DE]、ES:[HL]的示例



3.4.7 基址寻址

[功能]

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

[操作数格式]

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

图 3-22. [SP + 字节]的示例

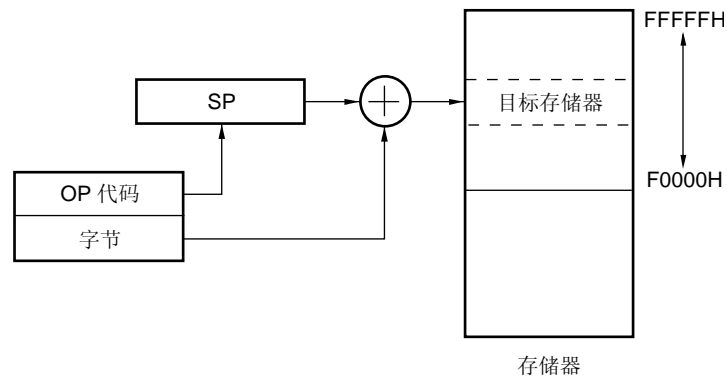


图 3-23. [HL + 字节]、[DE + 字节]的示例

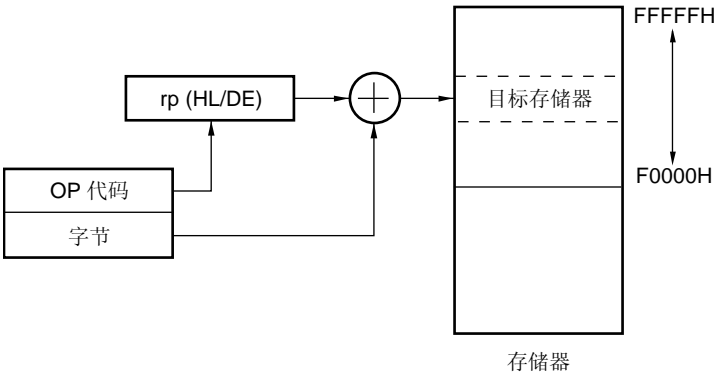


图 3-24. 字[B]、字[C]的示例

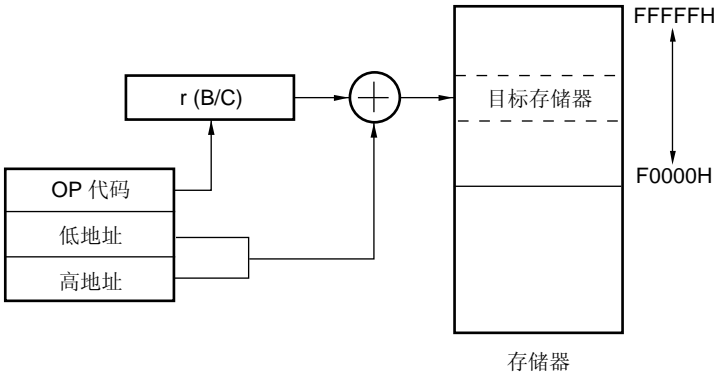


图 3-25. 字[BC]的示例

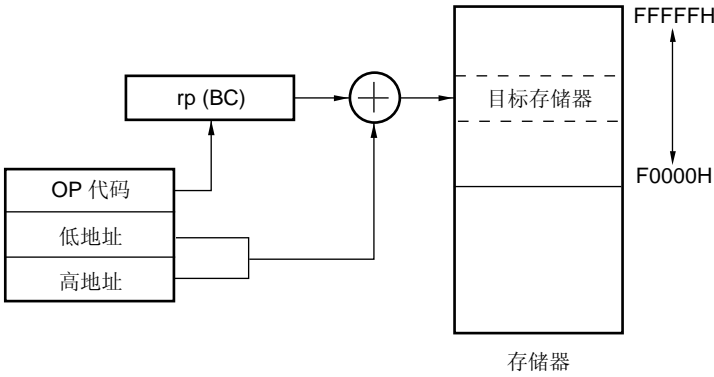


图 3-26. ES:[HL + 字节]、ES:[DE + 字节]的示例

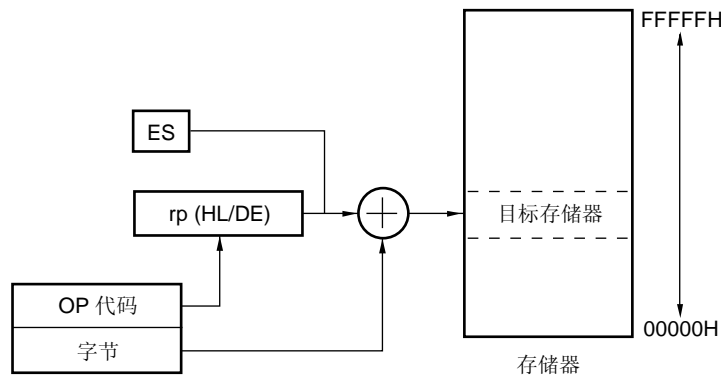


图 3-27. ES: 字[B]、ES: 字[C] 的示例

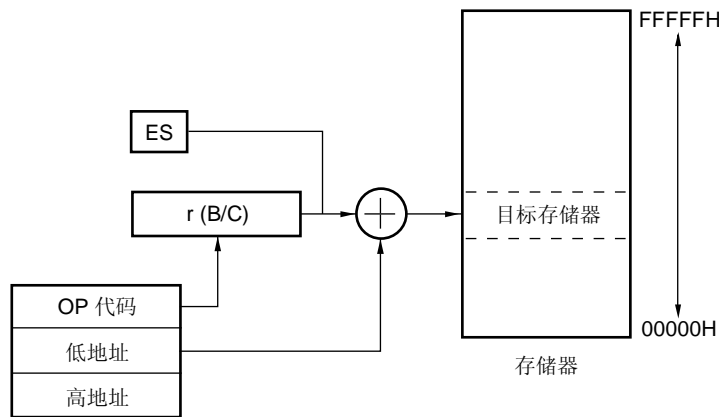
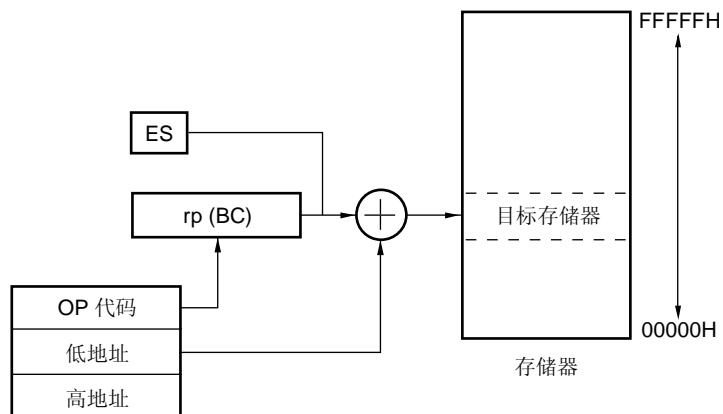


图 3-28. ES: 字[BC] 的示例



3.4.8 相对基址寻址

[功能]

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

[操作数格式]

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

图 3-29. [HL+B]、[HL+C] 的示例

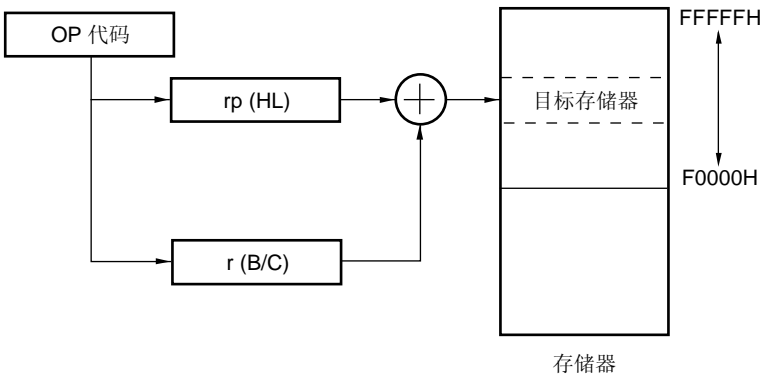
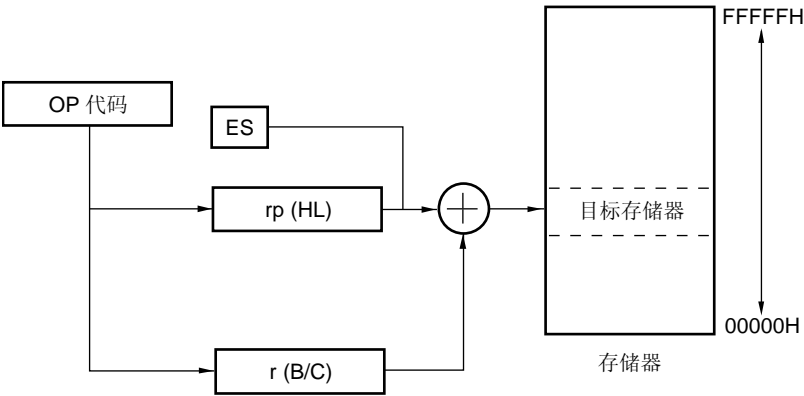


图 3-30. ES: [HL+B]、ES: [HL+C] 的示例



3.4.9 堆栈寻址

[功能]

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

该堆栈仅应用于内部 RAM 区域。

[操作数格式]

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

第 4 章 端口功能

4.1 端口功能

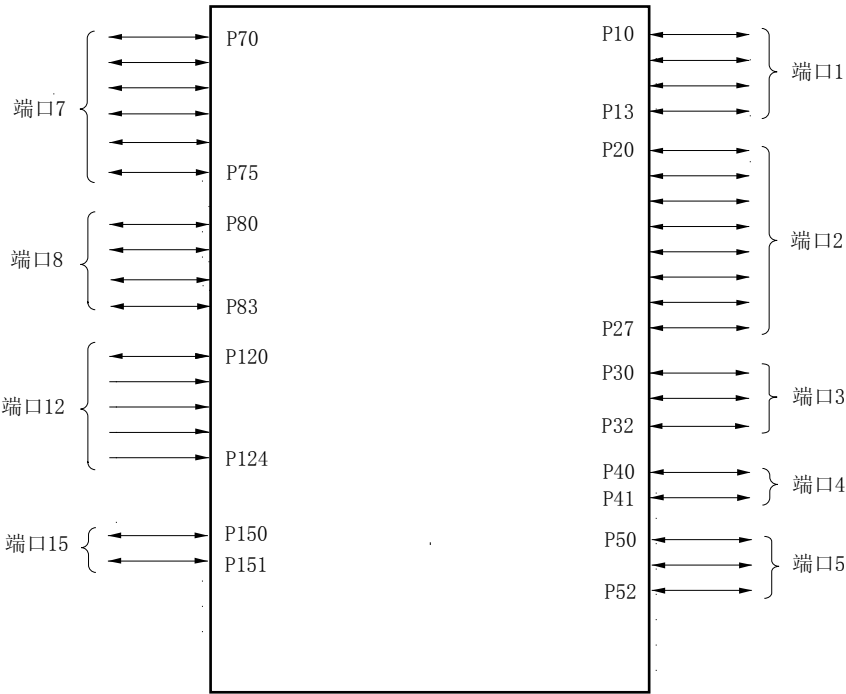
有 2 种类型的引脚 I/O 缓冲器供电电源：AVREF 和 VDD。下表显示了这些供电电源与引脚之间的关系。

表 4-1. 引脚 I/O 缓冲器供电电源

供电电源	对应的引脚
AVREF	P20 ~ P27, P150、P151 P80 ~ P83
VDD	• 除去 P20~P27、P150~P151、P80~P83 以外的端口引脚 • 除去端口引脚以外的引脚

μPD79F9211 产品提供如图 4-1 所示的端口，这些端口允许多种控制操作。每个端口的功能如表 4-2 所示。除了作为数字 I/O 端口功能，这些端口还有若干复用功能。复用功能的详细信息请见第 2 章 引脚功能。

图 4-1. 端口类型



4.1.1 44 引脚产品

表 4-2. 端口功能 (1/2)

功能名称	I/O	功能	复位后	复用功能
P10	I/O	端口 1。 4 位 I/O 端口。 可以指定输入/输出为 1 位单元模式。 通过软件设置, 可以定义片内上拉电阻的使用。	输入端口	TI02/TO02
P11				TI03/TO03
P12				TI04/TO04
P13				TI05/TO05
P20~P27	I/O	端口 2。 8 位 I/O 端口。 可以指定输入/输出为 1 位单元模式。	数字输入 端口	ANI0~ANI7
P30	I/O	端口 3。 3 位 I/O 端口。 P31 和 P32 输入可设置成 TTL 缓冲器。 P30 至 P32 的输出可设置成 N 沟道漏极开路输出 (V _{DD} 耐压)。 可以指定输入/输出为 1 位单元模式。 通过软件设置, 可以定义片内上拉电阻的使用。	输入端口	SO10/TxD1/TO11
P31				SI10/RxD1/SDA10/ INTP1/TI09
P32				SCK10/SCL10/ INTP2
P40 [※]	I/O	端口 4。 2 位 I/O 端口。 可以指定输入/输出为 1 位单元模式。 通过软件设置, 可以定义片内上拉电阻的使用。	输入端口	TOOL0
P41				TOOL1
P50	I/O	端口 5。 3 位 I/O 端口。 可以指定输入/输出为 1 位单元模式。 通过软件设置, 可以定义片内上拉电阻的使用。	输入端口	TI06/TO06
P51			输入端口	TI07/TO07
P52			输入端口	SLTI/SLTO
P70	I/O	端口 7。 6 位 I/O 端口。 P71、P72、P74 和 P75 的输入可以被设置到 TTL 缓冲器。 P70、P72、P73 和 P75 的输出可设置为 N 沟道漏极开路输出 (V _{DD} 耐压)。 可以指定输入/输出为 1 位单元模式。 通过软件设置, 可以定义片内上拉电阻的使用。	输入端口	SO01/INTP4
P71				SI01/INTP5
P72				SCK01/INTP6
P73				SO00/TxD0/TO10
P74				SI00/RxD0/TI10
P75				SCK00/TI11
P80	I/O	端口 8。 4 位 I/O 端口。 可以以 1 位单元输入/输出模式。 P80 至 P83 的输入可以设置为比较器输入或可编程增益放大器输入。	模拟输入	CMP0P/TMOFF0/ INTP3/PGAI
P81				CMP0M
P82				CMP1P/TMOFF1/ INTP7
P83				CMP1M

注 如果通过选项字节使能片上调试, 要确认上拉外部 P40/TOOL0 引脚 (参见在 2.2.4 P40、P41 (端口 4)的注意事项)。

表 4-2. 端口功能 (2/2)

功能名称	I/O	功能	复位后	复用功能
P120	I/O	端口 12。 1 位 I/O 端口和 4 位输入端口。 仅对 P120，可以指定输入/输出为 1 位单元模式。 仅对 P120，通过软件设置，可以定义片内上拉电阻的使用。	输入端口	INTP0/EXLVI
P121	输入			X1
P122				X2/EXCLK
P123				XT1
P124				XT2
P150、P151	I/O	端口 15。 2 位 I/O 端口。 可以以 1 位单元输入/输出模式。	数字输入 端口	ANI8、ANI9

4.2 端口配置

端口包括如下硬件。

表 4-3. 端口配置

项目	配置
控制寄存器	端口模式寄存器(PM1～PM5、PM7、PM8、PM12、PM15) 端口寄存器(P1～P5、P7、P8、P12、P15) 上拉电阻选项寄存器(PU1、PU3～PU5、PU7、PU12) 端口输入模式寄存器(PIM3、PIM7、PIM8) 端口输出模式寄存器(POM3、POM7) A/D 端口配置寄存器(ADPC)
端口	总共: 37 (CMOS I/O: 33, CMOS 输入: 4)
上拉电阻	总计: 19

4.2.1 端口 1

端口 1 是具有输出锁存功能的 4 位 I/O 端口。通过使用端口模式寄存器 1(PM1)，可以以 1 位单元设置端口 1 为输入或输出模式。当 P10 至 P13 被用于输入端口时，内部上拉电阻的使用可以通过上拉电阻选项寄存器 1(PU1)按位指定。

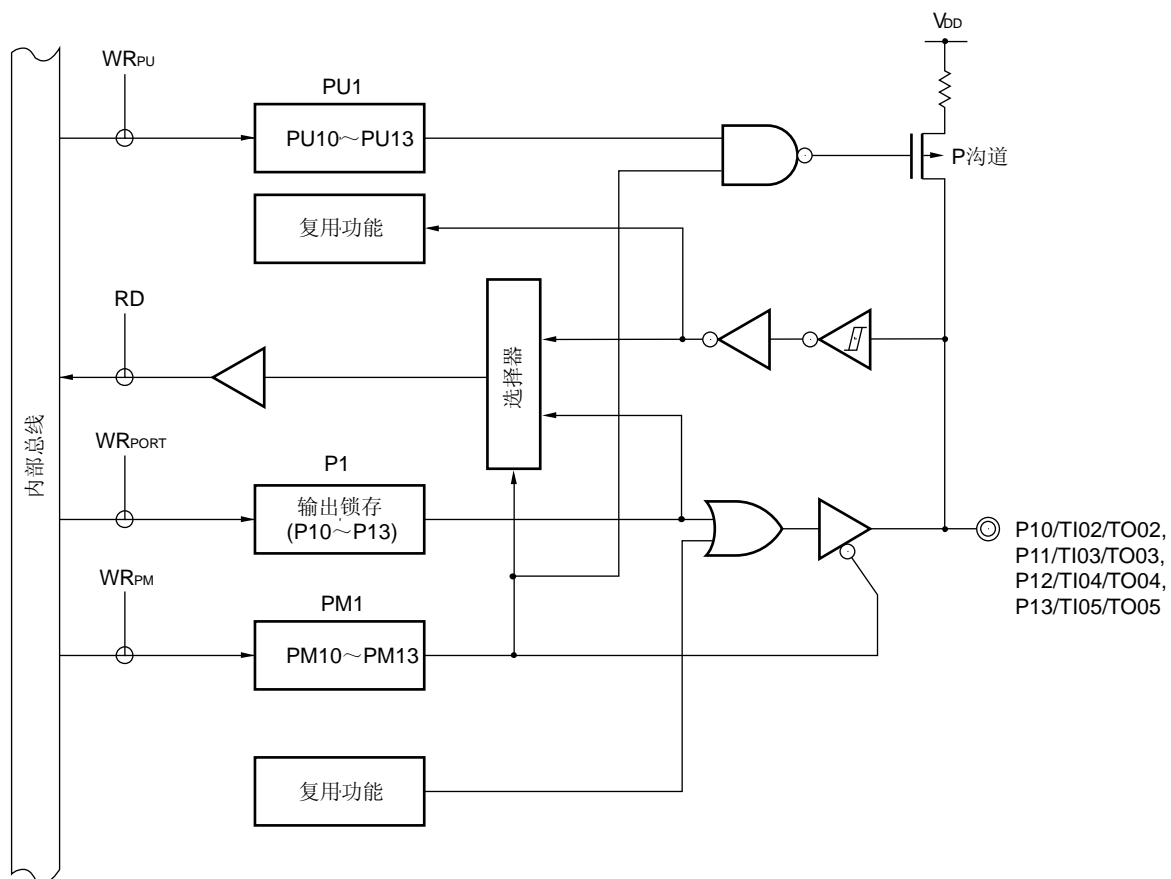
此端口也可用于定时器 I/O。

复位信号发生将端口 1 设置为输入模式。

图 4-2 所示为端口 1 的框图。

注意事项 P10/TI02/TO02、P11/TI03/TO03、P12/TI04/TO04 或 P13/TI05/TO05 作为普通端口时，应设置定时器输出寄存器 0(TO0)的位 2 至位 5(TO02 至 TO05)以及定时器输出使能寄存器 0(TOE0)的位 2 至位 5(TOE02 至 TOE05)为“0”，即默认设置。

图 4-2. P10 至 P13 的框图



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

4.2.2 端口 2

端口 2 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 2(PM2)，可以以 1 位单元设置端口 2 为输入或输出模式。

此端口也可用作 A/D 转换器模拟输入。

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

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

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

表 4-4. P20/ANI0 至 P27/ANI7 引脚的设置功能

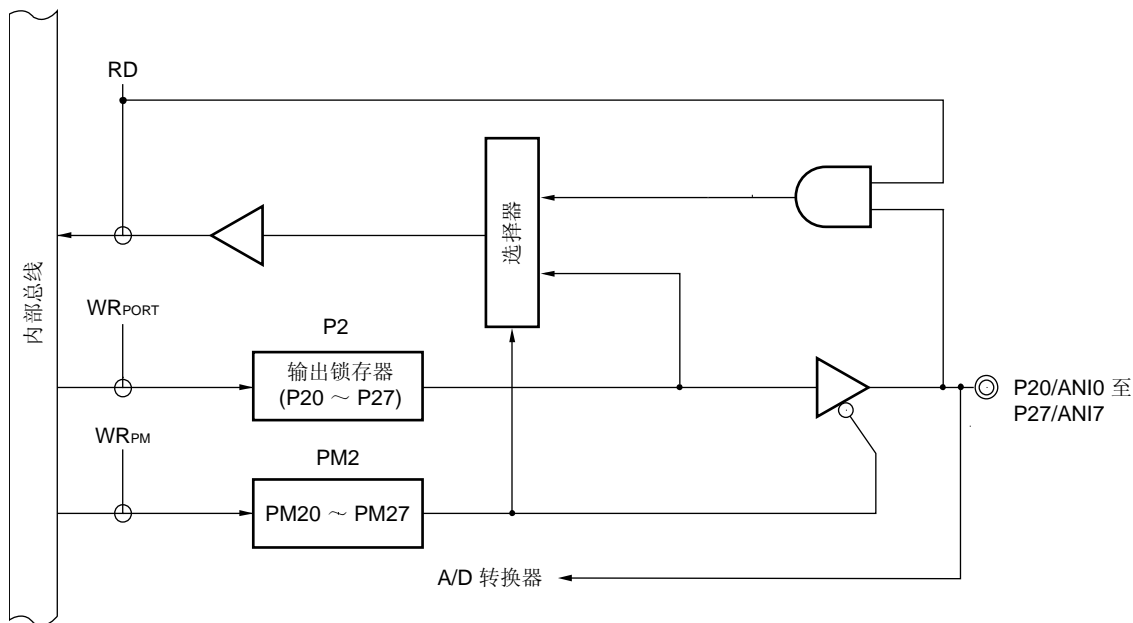
ADPC	PM2	ADS	P20/ANI0~P27/ANI7 引脚
数字 I/O 选择	输入模式	—	数字输入
	输出模式	—	数字输出
模拟输入选择	输入模式	选择 ANI。	模拟输入（转换）
		不要选择 ANI。	模拟输入（不转换）
	输出模式	选择 ANI。	设置禁止
		不要选择 ANI。	

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

图 4-3 所示为端口 2 的框图。

注意事项 当端口 2 作为数字端口时，AVREF 引脚电位要与 VDD 相同。

图 4-3. P20~P27 的框图



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

4.2.3 端口 3

端口 3 是具有输出锁存功能的 3 位 I/O 端口。通过使用端口模式寄存器 3(PM3)，可以以 1 位单元设置端口 3 为输入或输出模式。如果 P30 至 P32 用于输入端口，则内部上拉电阻的使用可以通过上拉电阻选项寄存器 3(PU3)按位指定。

通过端口输入模式寄存器 3(PIM3)，可以以 1 位单元指定 P31 和 P32 引脚通过普通缓冲器或 TTL 输入缓冲器的输入。

通过端口输出模式寄存器 3(POM3)，可以以 1 位单元指定 P30 至 P32 引脚的输出为 N 沟道漏极开路输出（V_{DD} 耐压）。

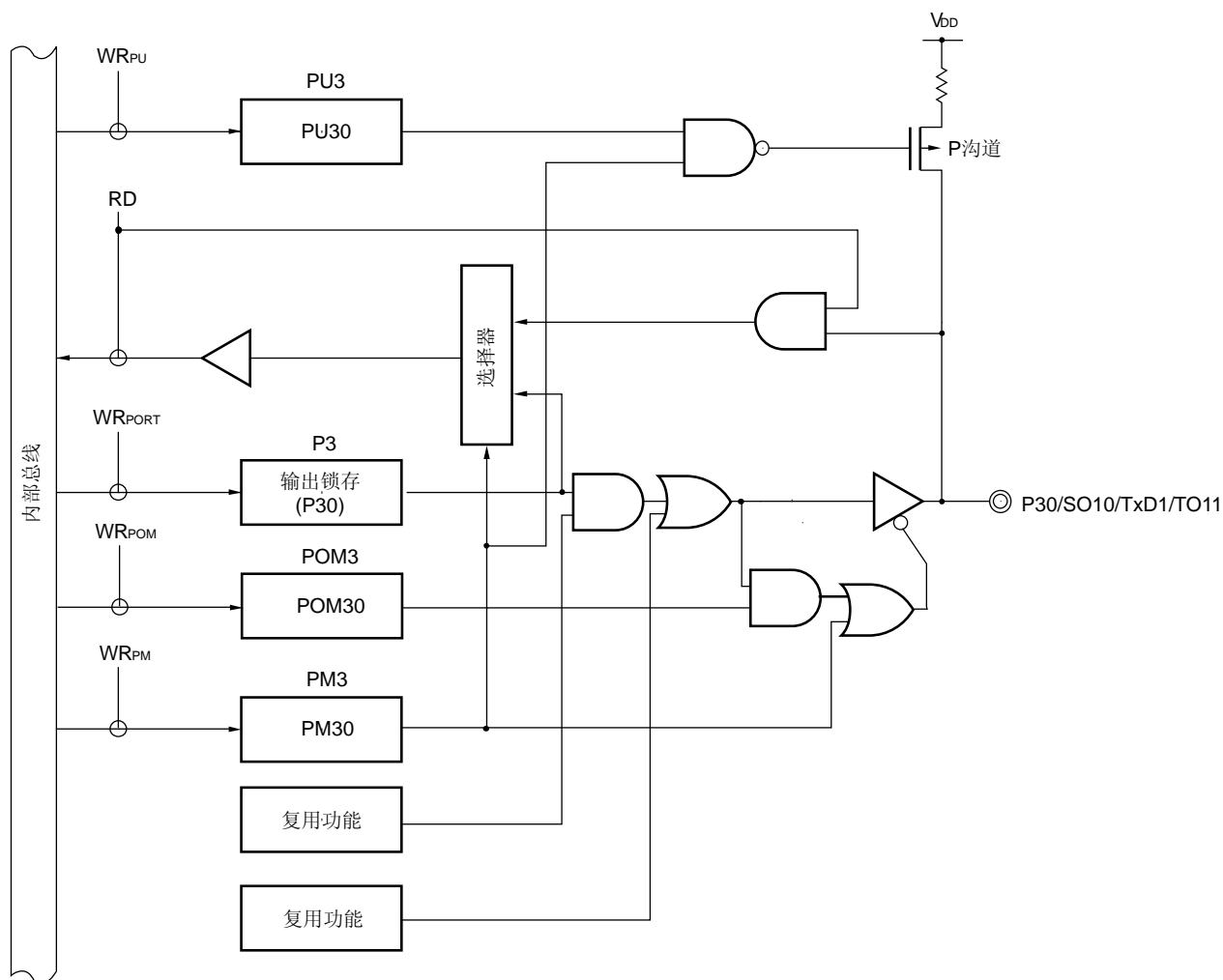
此端口也作为串行接口数据 I/O、时钟 I/O、外部中断请求输入和定时器的 I/O。

复位信号发生将端口 3 设置为输入模式。

图 4-4 和 4-5 所示为端口 3 的框图。

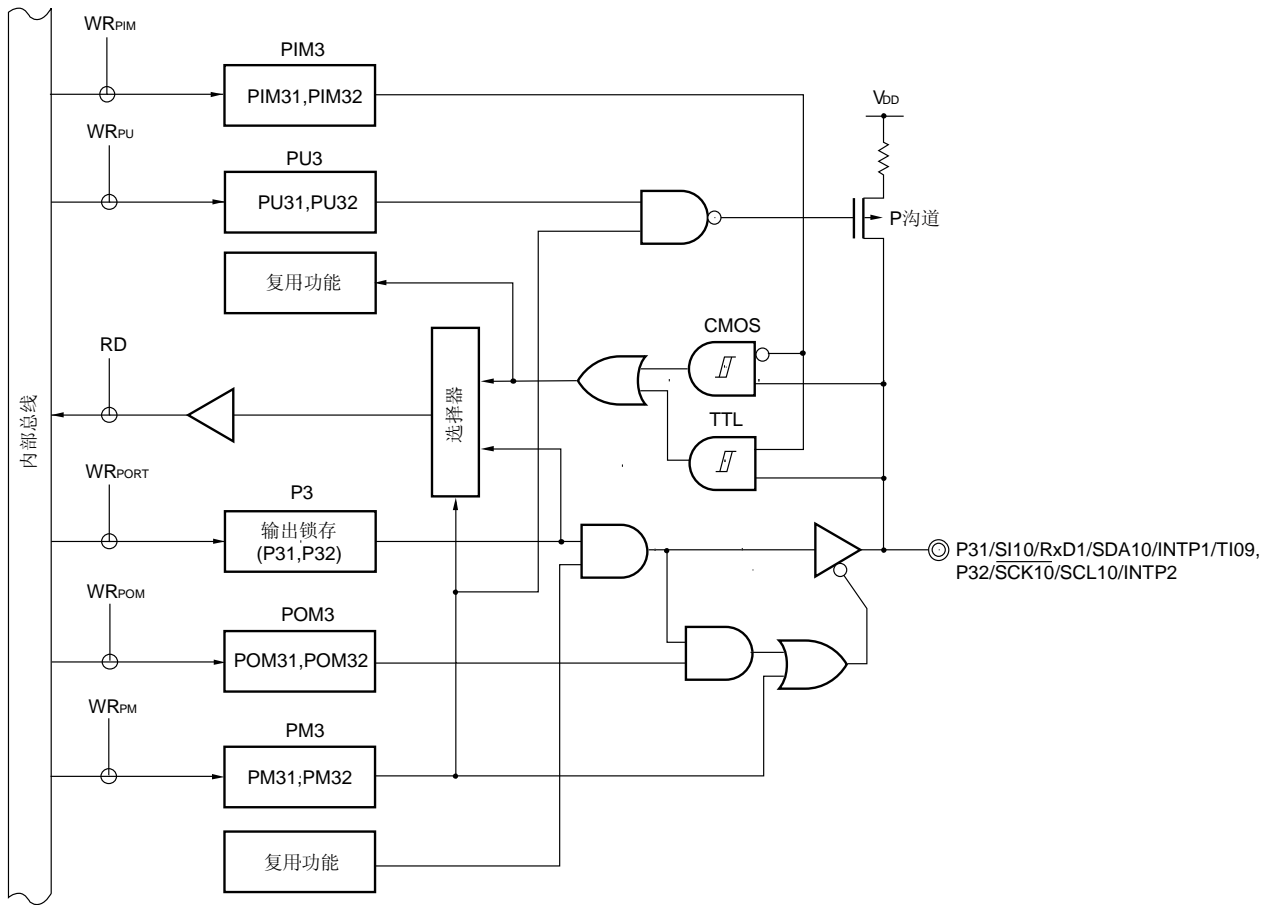
注意事项 要将 P30/SO10/TxD1/TO11、P31/SI10/RxD1/SDA10/INTP1/TI09、P32/ $\overline{\text{SCK10}}$ /SCL10/INTP2 作为通用端口使用，注意串行阵列单元设置。详细信息请参考表 12-9 寄存器设置和引脚间的关系（通道 2：CSI10、UART1 发送、IIC10）和表 12-10 寄存器设置和引脚间的关系（通道 3：UART1 接收）。

图 4-4. P30 的框图



P3: 端口寄存器 3
 PU3: 上拉电阻选项寄存器 3
 POM3: 端口输出模式寄存器 3
 PM3: 端口模式寄存器 3
 RD: 读信号
 WR_{xx}: 写信号

图 4-5. P31 和 P32 的框图



P3: 端口寄存器 3
PU3: 上拉电阻选项寄存器 3
PIM3: 端口输入模式寄存器 3
POM3: 端口输出模式寄存器 3
PM3: 端口模式寄存器 3
RD: 读信号
WR_{xx}: 写信号

4.2.4 端口 4

端口 4 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 4(PM4)，可以以 1 位单元设置端口 4 为输入或输出模式。当 P40 至 P41 引脚被用于输入端口时，内部上拉电阻的使用可以由上拉电阻选项寄存器 4(PU4)*按位指定。

此端口也可用于 Flash 存储器编程器/调试器的数据 I/O 和时钟输出。

复位信号发生将端口 4 设置为输入模式。

图 4-7 所示为端口 4 的框图。

注 连接工具时，P40 和 P41 引脚不能连接上拉电阻。

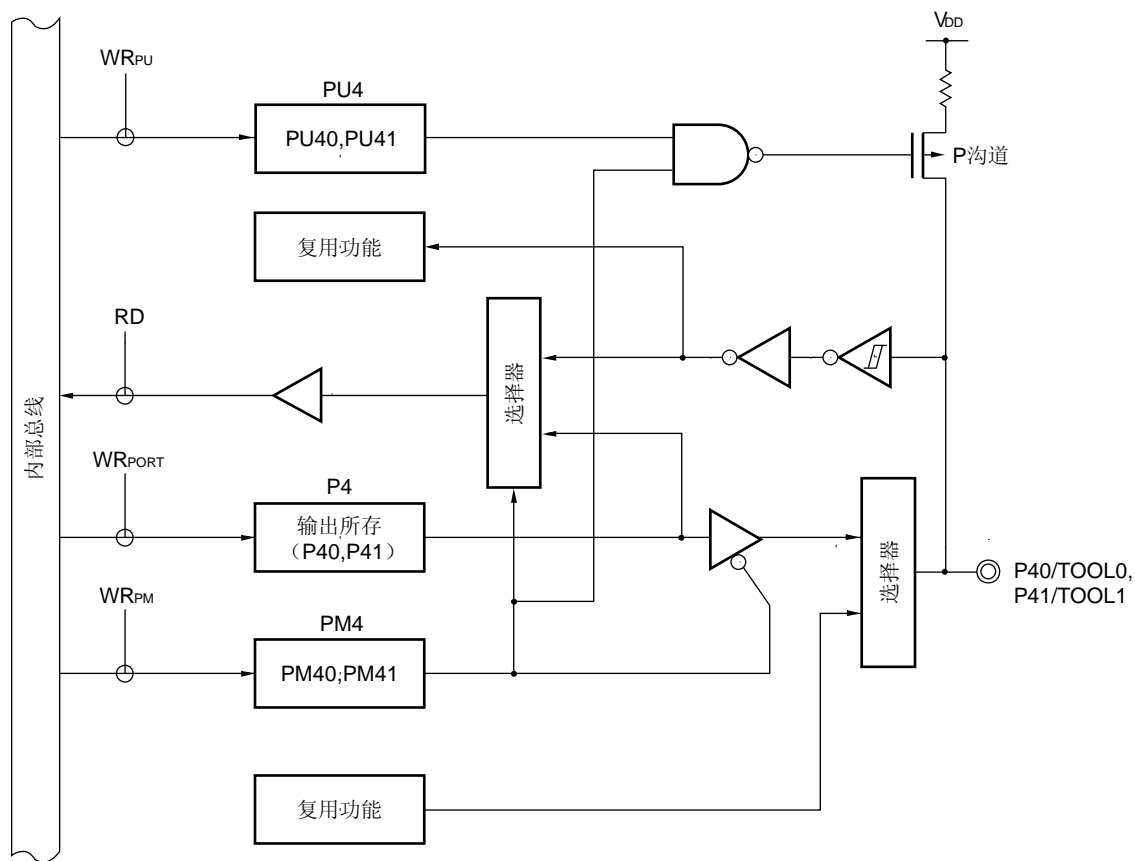
注意 当连接工具时，P40 引脚不能作为端口引脚使用。

使用片内调试功能时，P41 引脚通过模式设置可用于如下调试。

1 线模式：可用作端口（P41）。

2 线模式：可用作 TOOL1 引脚，不能作为端口（P41）使用。

图 4-6. P40 和 P41 框图



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

4.2.5 端口 5

端口 5 是具有输出锁存功能的 3 位 I/O 端口。通过使用端口模式寄存器 5(PM5)，可以以 1 位单元设置端口 5 为输入或输出模式。当 P50 至 P52 作为输入端口时，内部上拉电阻的使用可以通过上拉电阻选项寄存器 5(PU5)以 1 位单元的方式指定。

此端口也可用于定时器 I/O。

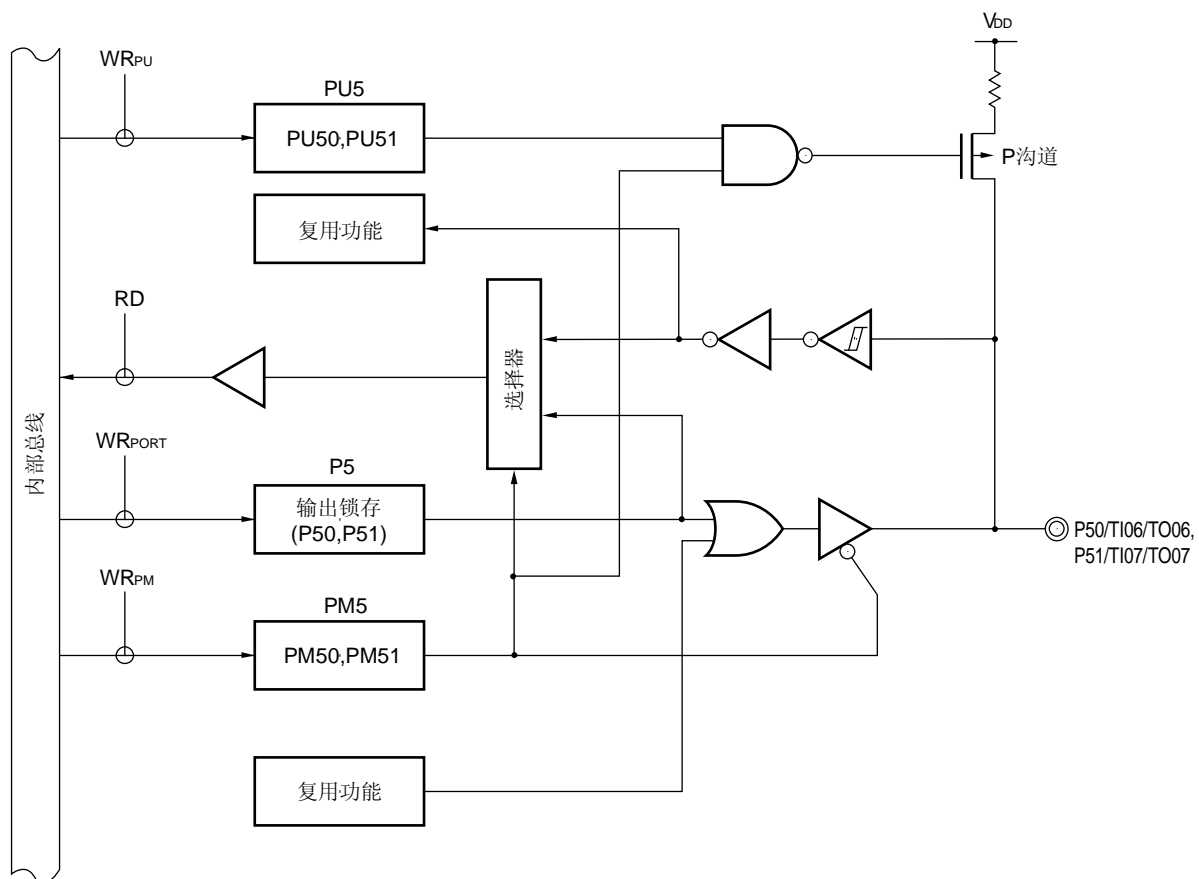
复位信号发生将端口 5 设置为输入模式。

图 4-7 和 4-9 所示为端口 5 的框图。

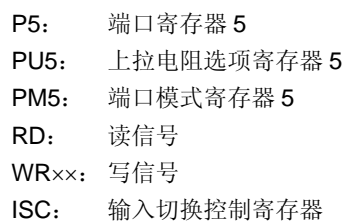
- 注意事项**
1. 为了使用 P50/TI06/TO06 或 P51/TI07/TO07 作为通用端口，设置定时器输出寄存器 0 (TO0)的位 6 和 7(TO06, TO07)和定时器输出使能寄存器 0 (TOE0)的位 6 和 7(TOE06, TOE07)为“0”，与其默认设置一致。
 2. 为了使用 P52/SLTI/SLTO 作为通用端口，检查输入切换控制寄存器(ISC)中所选择的通道中所对应的定时器 I/O 引脚。而且，设置定时器输出寄存器 0 (TO0)的位 n (TO0n) 和定时器输出使能寄存器 0 (TOE0)的位 n (TOE0n)皆为“0”，该设置与其初始设置相同。

备注 n = 00、01、08~11

图 4-7. P50 和 P51 的框图



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



4.2.6 端口 7

端口 7 是具有输出锁存功能的 6 位 I/O 端口。通过使用端口模式寄存器 7(PM7)，可以以 1 位单元设置端口 7 为输入或输出模式。当 P70 和 P75 引脚用作输入端口时，通过上拉电阻选项寄存器 7(PU7)，片内上拉电阻的使用可以以 1 位单元指定。

利用端口输入模式寄存器 7(PIM7)，可通过一个正常的输入缓冲器或 TTL 输入缓冲器，以 1 位单元指定输入至 P71、P72、P74 和 P75 引脚。

利用端口输出模式寄存器 7(POM7)，从 P70、P72、P73 和 P75 引脚的输出，可以以 1 位单元指定为 N 沟道漏极开路输出（V_{DD} 耐压）。

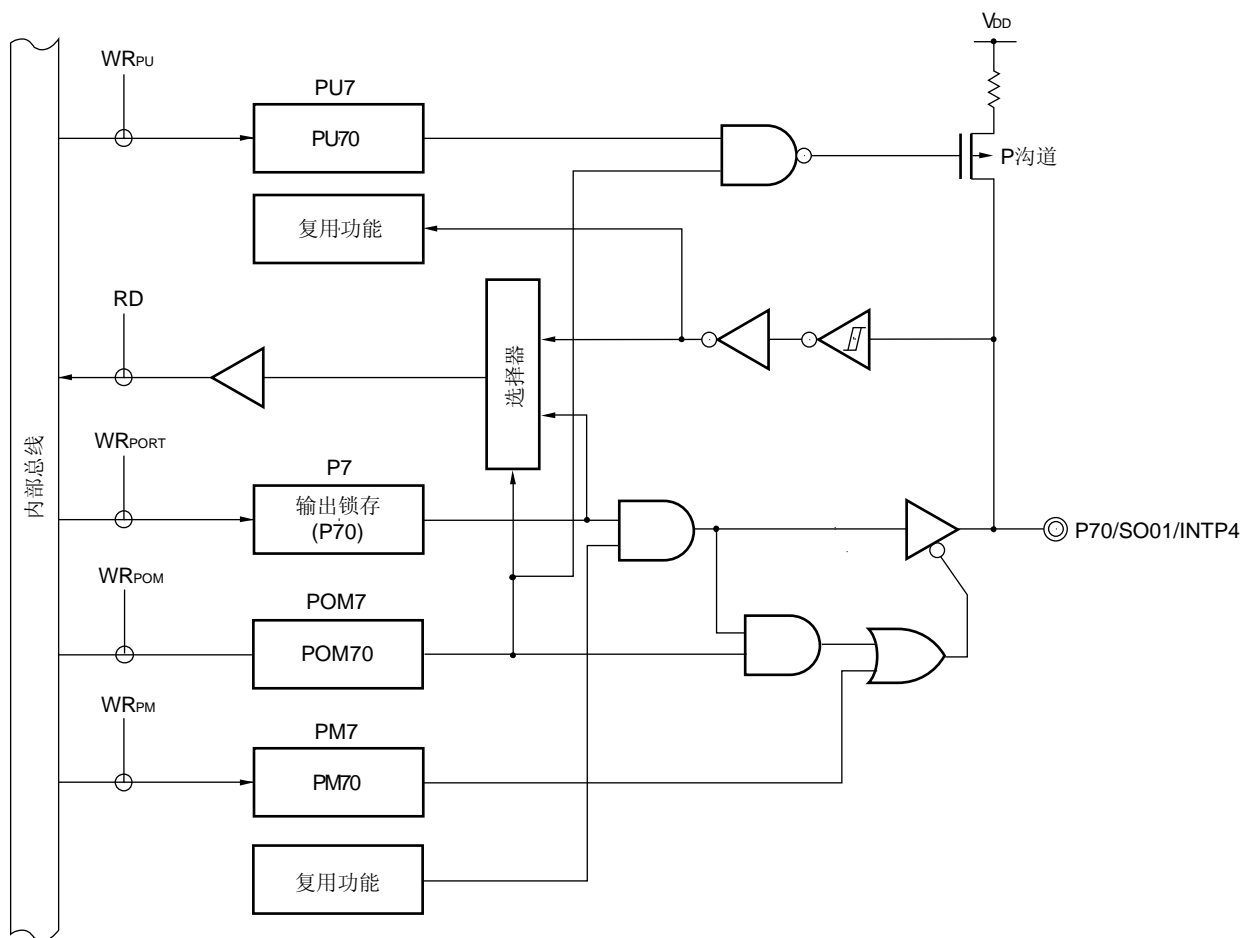
此端口也作为串行接口数据 I/O、时钟 I/O、外部中断请求输入和定时器的 I/O。

复位信号发生将端口 7 设置为输入模式。

图 4-9 至 4-12 所示为端口 7 的框图。

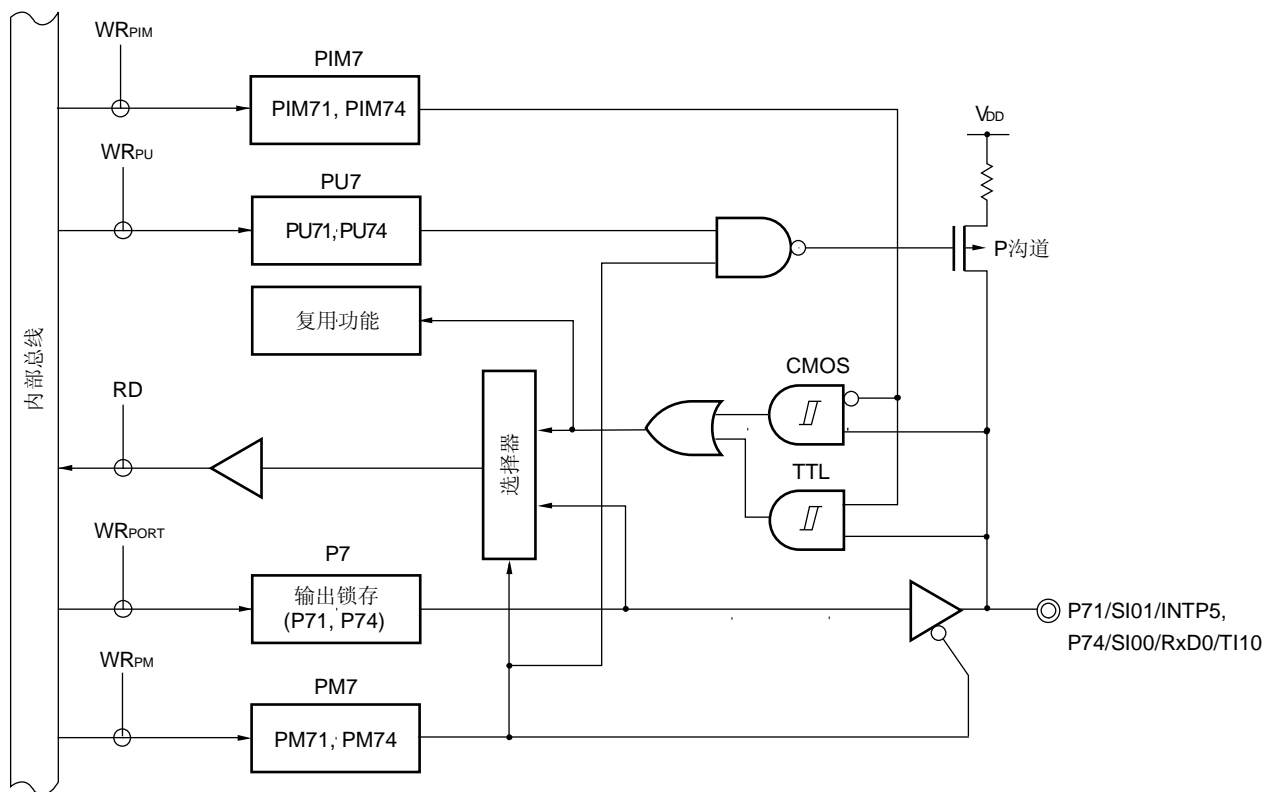
- 注意事项**
1. 要将 P70/SO01/INTP4、P71/SI01/INTP5、P72/SCK01/INTP6、P73/SO00/TxD0/TO10、P74/SI00/RxD0/TI10、P75/SCK00/TI11 用作通用端口，注意串行阵列单元设置。详细信息请参考表 12-6 寄存器设置和引脚之间的关系（通道 0：CSI00，UART0 发送）和表 12-8 寄存器设置和引脚之间的关系（通道 1：CSI01，UART0 接收）。
 2. 为了使用 P73/SO00/TxD0/TO10 作为通用端口，设置定时器输出寄存器 0 (TO0) 的位 10 (TO10) 和定时器输出使能寄存器 0 (TOE0) 的位 10 (TOE10) 为“0”，与其默认设置一致。

图 4-9. P70 的框图



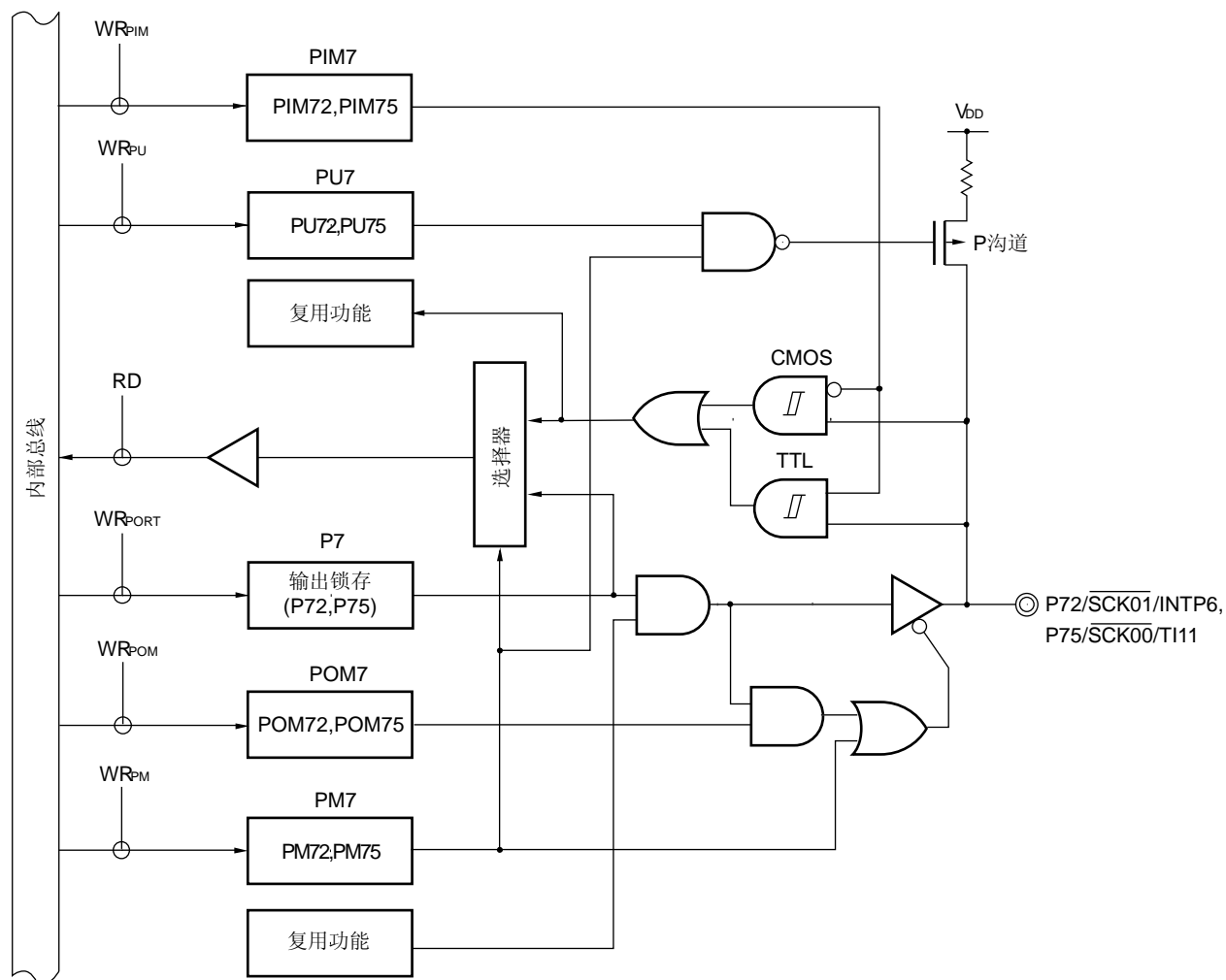
P7: 端口寄存器 7
 PU7: 上拉电阻选项寄存器 7
 POM7: 端口输出模式寄存器 7
 PM7: 端口模式寄存器 7
 RD: 读信号
 WR_{xx}: 写信号

图 4-10. P71 和 P74 的框图



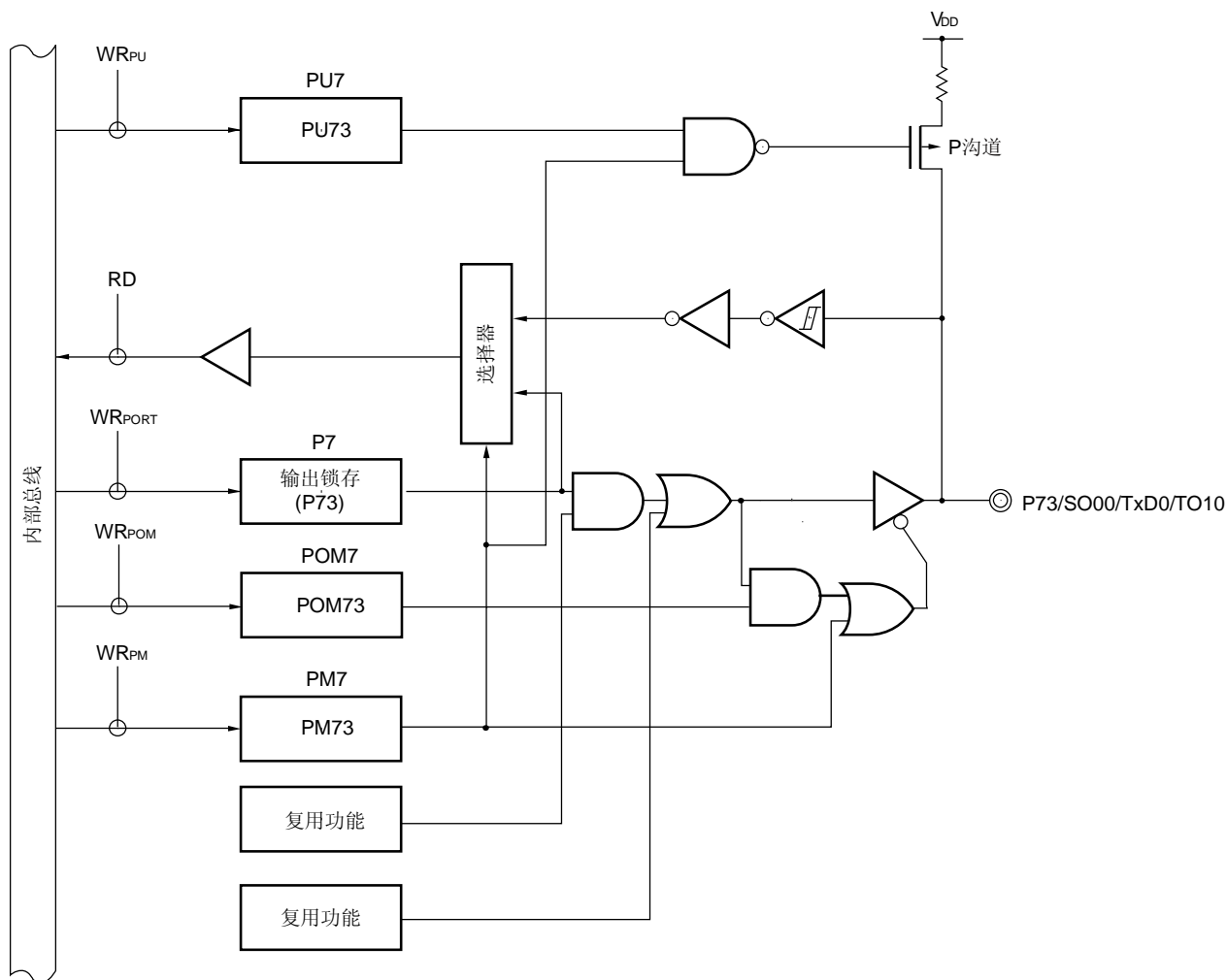
P7:	端口寄存器 7
PU7:	上拉电阻选项寄存器 7
PM7:	端口模式寄存器 7
PIM7:	端口输入模式寄存器 7
RD:	读信号
WRxx:	写信号

图 4-11. P72 和 P75 的框图



P7: 端口寄存器 7
 PU7: 上拉电阻选项寄存器 7
 PM7: 端口模式寄存器 7
 PIM7: 端口输入模式寄存器 7
 POM7: 端口输出模式寄存器 7
 RD: 读信号
 WR_{xx}: 写信号

图 4-12. P73 的框图



P7: 端口寄存器 7
 PU7: 上拉电阻选项寄存器 7
 PM7: 端口模式寄存器 7
 POM7: 端口输出模式寄存器 7
 RD: 读信号
 WR_{xx}: 写信号

4.2.7 端口 8

端口 8 是具有输出锁存功能的 4 位 I/O 端口。通过使用端口模式寄存器 8(PM8)，可以以 1 位单元设置端口 8 为输入或输出模式。

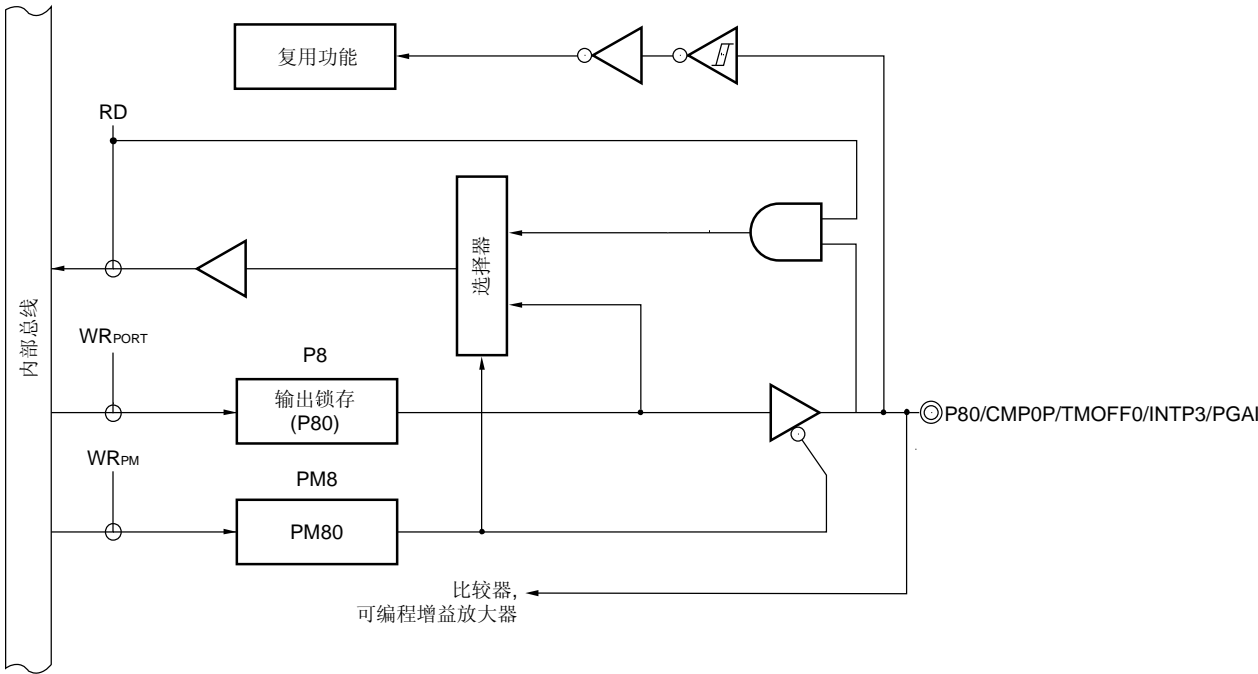
<R> P80 至 P83 引脚的数据输入，必须使用端口输入模式寄存器 8(PIM8)，以 1 位单元设置允许或禁止。

此端口也可用作比较器 0 和 1(+)端的输入电压，比较器 0 和 1(-)端的输入电压，定时器引脚高阻的控制输入，外部中断请求输入和可编程增益放大器的输入。

复位信号发生将端口 8 设置为输入模式。

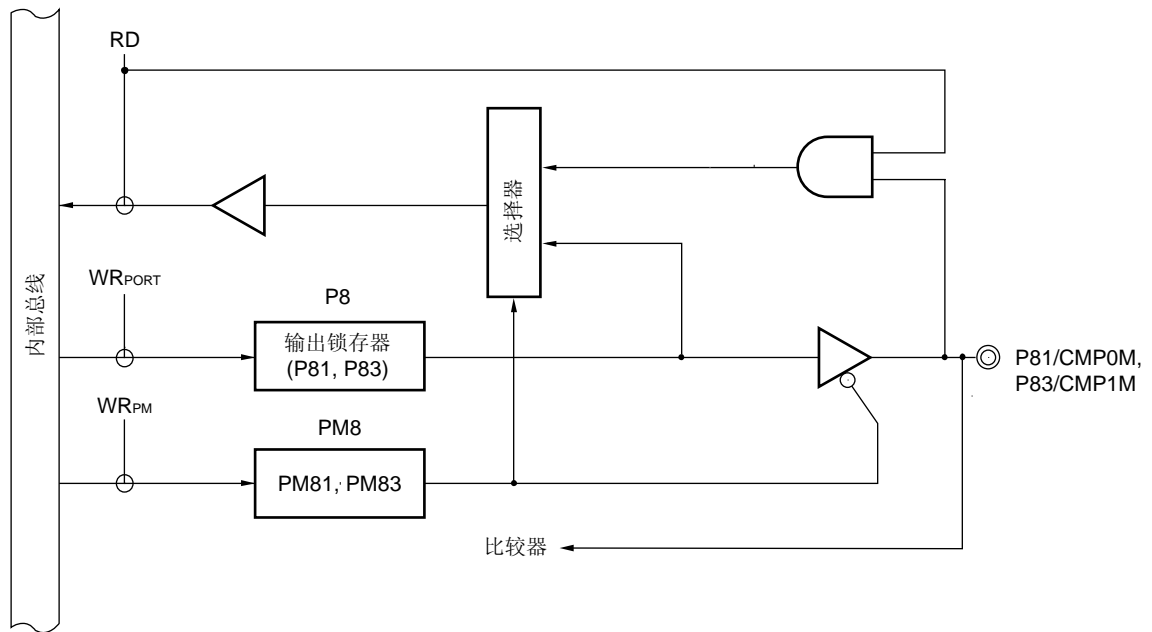
图 4-13 至 4-15 所示为端口 8 的框图。

图 4-13. P80 的框图

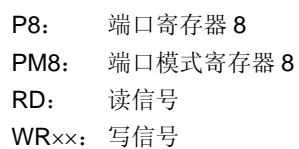


P8: 端口寄存器 8
 PM8: 端口模式寄存器 8
 RD: 读信号
 WR_{xx}: 写信号

图 4-14. P81 和 P83 框图



P8: 端口寄存器 8
 PM8: 端口模式寄存器 8
 RD: 读信号
 WR_{xx}: 写信号



4.2.8 端口 12

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

P121 至 P124 是 4 位输入端口。

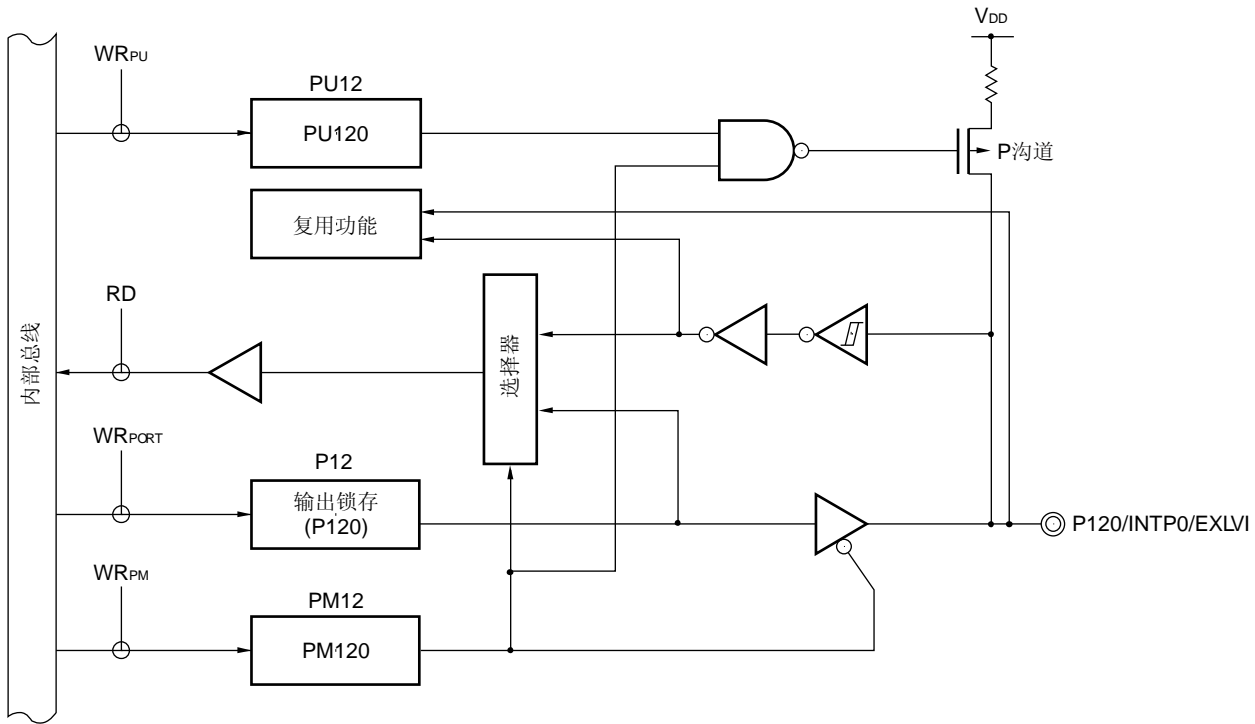
此端口也可用作外部中断请求输入、外部低电压检测的电压输入、主系统时钟振荡器连接、副系统时钟振荡器连接和主系统时钟的外部时钟输入。

复位信号发生将端口 12 设置为输入模式。

图 4-16 和 4-18 所示为端口 12 的框图。

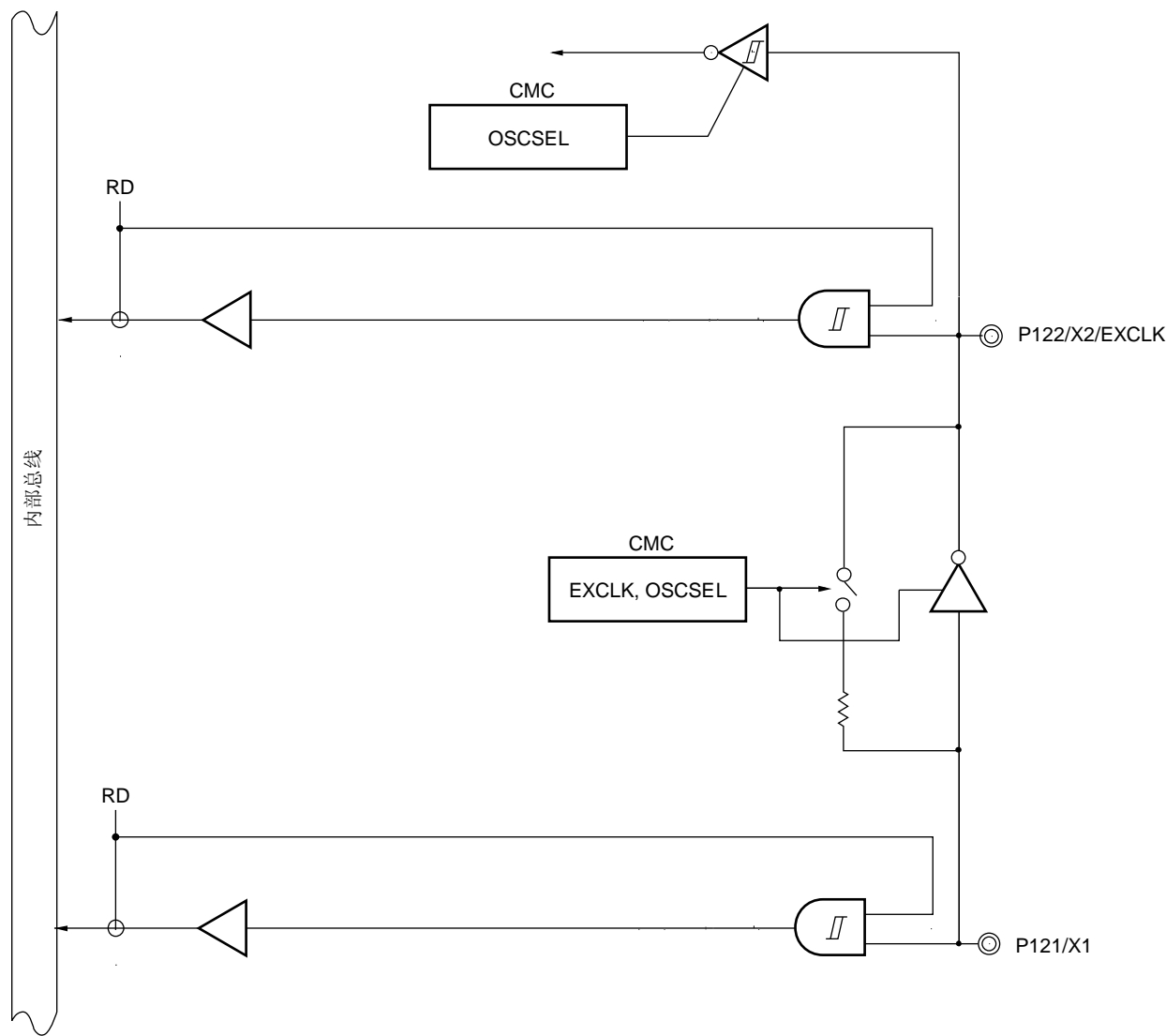
注意事项 P121 至 P124 的功能设置只有在复位释放后可用。端口一旦设置为振荡器连接，除非执行复位，否则不能用作输入端口。

图 4-16 P120 的框图



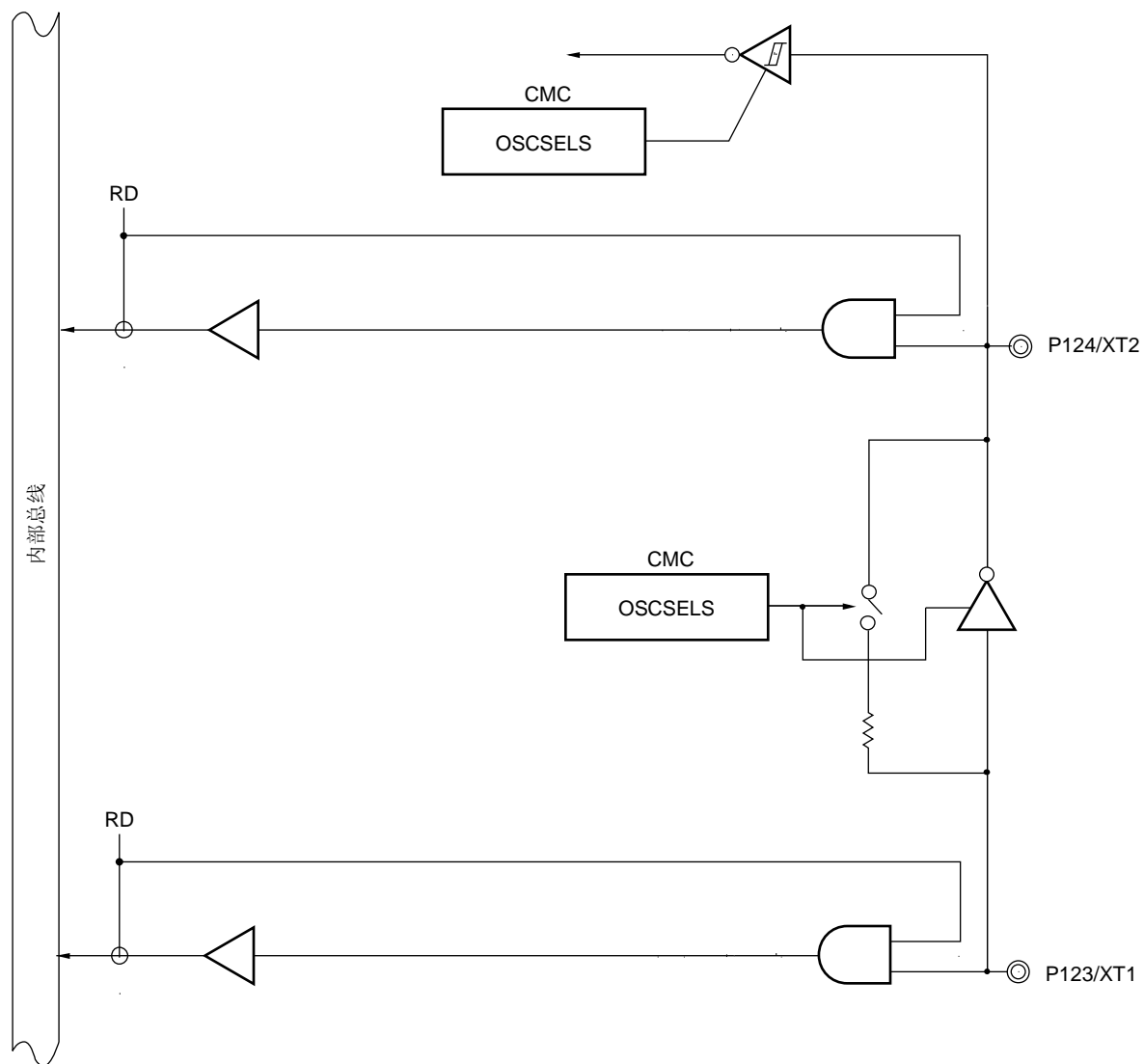
- P12: 端口寄存器 12
- PU12: 上拉电阻选项寄存器 12
- PM12: 端口模式寄存器 12
- RD: 读信号
- WR_{xx}: 写信号

图 4-17. P121 和 P122 的框图



CMC: 时钟操作模式控制寄存器
RD: 读信号

图 4-18 P123 和 P124 的框图



CMC: 时钟操作模式控制寄存器

RD: 读信号

4.2.9 端口 15

端口 15 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 15(PM15)，可以以 1 位单元设置端口 15 为输入或输出模式。

此端口也可用作 A/D 转换器模拟输入。

要将 P150/ANI8 和 P151/ANI9 用作数字输入引脚，可通过 A/D 端口配置寄存器(ADPC)设置其为数字 I/O 模式，通过 PM15 设置其为输入模式。从低位开始使用这些引脚。

要将 P150/ANI8 和 P151/ANI9 用作数字输出引脚，通过 A/D 端口配置寄存器 (ADPC) 设置为数字 I/O 模式，并且通过 PM15 设置为输出模式。

表 4-9. P150/ANI8 和 P151/ANI9 引脚的设置功能

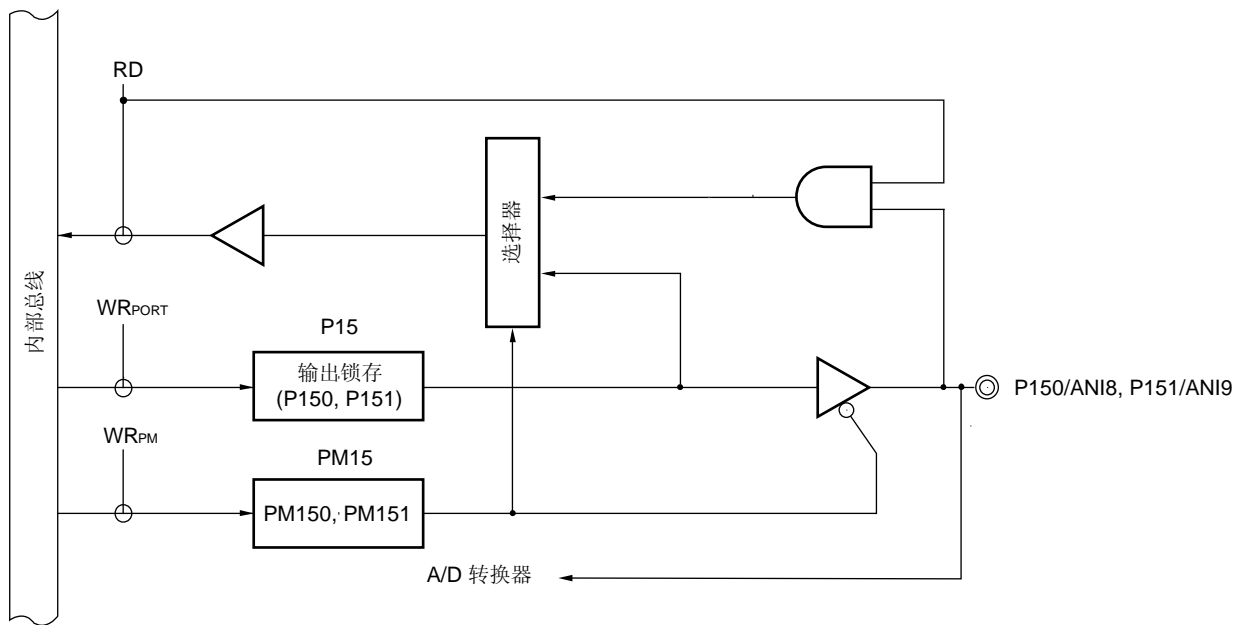
ADPC	PM15	ADS	P150/ANI8 和 P151/ANI9 引脚
数字 I/O 选择	输入模式	–	数字输入
	输出模式	–	数字输出
模拟输入选择	输入模式	选择 ANI。	模拟输入（转换）
		不要选择 ANI。	模拟输入（不转换）
	输出模式	选择 ANI。	设置禁止
		不要选择 ANI。	

复位信号发生时，P150/ANI8 和 P151/ANI9 所有引脚被设置为数字输入模式。

图 4-19 所示为端口 15 的框图。

注意事项 端口 15 用作数字端口时，设置 AVREF0 引脚的电压与 VDD 引脚相同。

图 4-19. P150 和 P151 的框图



P15: 端口寄存器 15
 PM15: 端口模式寄存器 15
 RD: 读信号
 WR_{xx}: 写信号

4.3 寄存器控制端口功能

端口功能由下列六种类型的寄存器控制。

- 端口模式寄存器 (PM1~ PM5、PM7、PM8、PM12、PM15)
- 端口寄存器 (P1 ~ P5、P7、P8、P12、P15)
- 上拉电阻选项寄存器 (PU1、PU3~PU5、PU7、PU12)
- 端口输入模式寄存器(PIM3、PIM7、PIM8)
- 端口输出模式寄存器(POM3、POM7)
- A/D 端口配置 寄存器 (ADPC)

(1) 端口模式寄存器 (PM1 至 PM8、PM12、PM15) 这类寄存器以 1 位单元指定端口的输入或输出模式。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生将这些寄存器设置为 FFH。

当端口引脚用作复用功能引脚时，请参考 **4.5 使用复用功能时，设置端口模式寄存器和输出锁存器**，对端口模式寄存器进行设置。

图 4-20. 端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM1	1	1	1	1	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	1	1	1	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26H	FFH	R/W
PM7	1	1	PM175	PM174	PM73	PM72	PM171	PM170	FFF27H	FFH	R/W
PM8	1	1	1	1	PM83	PM82	PM81	PM80	FFF28H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM15	1	1	1	1	1	0	PM151	PM150	FFF2FH	FFH	R/W
PMmn	Pmn 引脚 I/O 模式选择 (m = 1 ~ 8、12、15; n = 0 ~ 7)										
0	输出模式（输出缓冲器开）										
1	输入模式（输出缓冲器关）										

注意 以下位必须设置为1。

- PM1 寄存器的第 4 至 7 位
- PM3 寄存器的第 3 至 7 位
- PM4 寄存器的第 2 至 7 位
- PM5 寄存器的第 3 至 7 位
- PM6 寄存器的第 2 至 7 位
- PM7 的第 6 位和第 7 位
- PM8 寄存器的第 4 至 7 位
- PM12 寄存器的第 1 至 7 位
- PM15 寄存器的第 3 至 7 位

(2) 端口寄存器 (P1 至 P8、P12、P14、P15)

当数据从一个端口输出时，这类寄存器将芯片输出的数据写入。
如果在输入模式下读端口，则读取的是引脚电平。如果在输出模式下读端口，则读取的是输出锁存器的值^注。
可以由 1 位或 8 位存储器操作指令设置这些寄存器。
复位信号的产生可将这些寄存器清零(00H)。

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

图 4-21. 端口寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W																																															
P1	0	0	0	0	P13	P12	P11	P10	FFF01H	00H (输出锁存)	R/W																																															
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (输出锁存)	R/W																																															
P3	0	0	0	0	0	P32	P31	P30	FFF03H	00H (输出锁存)	R/W																																															
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H (输出锁存)	R/W																																															
P5	0	0	0	0	0	P52	P51	P50	FFF05H	00H (输出锁存)	R/W																																															
P7	0	0	P75	P74	P73	P72	P71	P70	FFF07H	00H (输出锁存)	R/W																																															
P8	0	0	0	0	P83	P82	P81	P80	FFF08H	00H (输出锁存)	R/W																																															
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	未定义	R/W ^注																																															
P15	0	0	0	0	0	0	P151	P150	FFF0FH	00H (输出锁存)	R/W																																															
<table border="1"> <tr> <th rowspan="2">Pmn</th><th colspan="11">m = 1~8、12、14、15; n = 0~7</th></tr> <tr> <th colspan="5">输出数据控制 (输出模式)</th><th colspan="6">输入数据读取 (输入模式)</th></tr> <tr> <td>0</td><td colspan="5">输出 0</td><td colspan="6">输入低电平</td></tr> <tr> <td>1</td><td colspan="5">输出 1</td><td colspan="6">输入高电平</td></tr> </table>												Pmn	m = 1~8、12、14、15; n = 0~7											输出数据控制 (输出模式)					输入数据读取 (输入模式)						0	输出 0					输入低电平						1	输出 1					输入高电平					
Pmn	m = 1~8、12、14、15; n = 0~7																																																									
	输出数据控制 (输出模式)					输入数据读取 (输入模式)																																																				
0	输出 0					输入低电平																																																				
1	输出 1					输入高电平																																																				

注 P121 至 P124 只读。

(3) 上拉电阻选项寄存器 (PU1、PU3~PU5、PU7、PU12)

这些寄存器指定是否使用 P10~P13、P30~P32、P40、P41、P50~P52、P70~P75 或 P120 的内部上拉电阻。PU1、PU3~PU5、PU7 和 PU12 中的内部上拉电阻设置成使用时，可以以 1 位单元的方式用于设置成输入模式的引脚。不论是否设置 PU1、PU3~PU5、PU7 和 PU12，设置成输出模式的引脚和用作复位功能的输出引脚都不连接内部上拉电阻。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零(00H)。

图 4-22. 上拉电阻选项寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU1	0	0	0	0	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	PU32	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	00H	R/W
PU5	0	0	0	0	0	PU52	PU51	PU50	F0035H	00H	R/W
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PUmn	Pmn 引脚片内上拉电阻选择 (m = 1、3~5、7、12; n = 0~5)										
0	片内上拉电阻不连接										
1	片内上拉电阻连接										

(4) 端口输入模式寄存器 (PIM3、PIM7、PIM8)

PIM3 和 PIM7 寄存器以 1 位单元设置 P31、P32、P71、P72、P74 或 P75 的输入缓冲器。

在与不同电位的外部器件串行通讯期间，可以选择 TTL 输入缓冲器。

<R> PIM8 用来以 1 位单元允许或禁止 P80 至 P83 的数据输入。当使用比较器或运算增益放大器时，通过软件处理来禁止数字输入（使用 P8n 引脚作为模拟输入）。要使用端口功能和复用功能，数字输入必须允许，因为其默认时是禁止的（使用 P8n 引脚作为模拟输入）。

可以由 1 位或 8 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零(00H)。

图 4-23. 端口输入模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PIM3	0	0	0	0	0	PIM32	PIM31	0	F0043H	00H	R/W
PIM7	0	0	PIM75	PIM74	0	PIM72	PIM71 [※]	0	F0047H	00H	R/W
PIM8	0	0	0	0	PIM83	PIM82	PIM81	PIM80	F0048H	00H	R/W
PIMmn	Pmn 引脚输入缓冲器选择 (m = 3 和 7; n = 1、2、4、5)										
0	普通输入缓冲器										
1	TTL 输入缓冲器										
PIM8n	P8n 引脚数字输入允许/禁止选择 (n = 0 至 3)										
0	数字输入禁止（使用 P8n 引脚作为模拟输入）。										
1	数字输入允许										

(5) 端口输出模式寄存器 (POM3、POM7)

这些寄存器以 1 位单元设置 P30~P32、P70、P72、P73 或 P75 的输出模式。
在与不同电位的外部设备串行通讯期间，以及 SDA10 引脚与相同电位的外部设备简易 I²C 通信期间，都可以选择 N 沟道漏极开路输出（V_{DD} 耐压）。
可以由 1 位或 8 位存储器操作指令设置这些寄存器。
复位信号的产生可将这些寄存器清零(00H)。

图 4-24. 端口输入模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
POM3	0	0	0	0	0	POM32	POM31	POM30	F0053H	00H	R/W
POM7	0	0	POM75 ^注	0	POM73	POM72 ^注	0	POM70 ^注	F0057H	00H	R/W
POMmn	Pmn 引脚输出模式选择 (m = 3 和 7; n = 0~3 和 5)										
0	正常输出模式.										
1	N 沟道漏极开路（V _{DD} 耐压）模式										

(6) A/D 端口配置 寄存器 (ADPC)

此寄存器切换 ANI0/P20~ANI7/P27、ANI8/P150、ANI9/P151 引脚到端口的数字 I/O 或 A/D 转换器的模拟输入。

ADPC 可以由 1 位或 8 位存储器操作指令设置。

复位信号的发生设置此寄存器为 10H。

图 4-25. A/D 端口配置寄存器 (ADPC) 的格式

地址: F0017H 复位后: 10H R/W

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

ADP C4	ADP C3	ADP C2	ADP C1	ADP C0	模拟输入 (A) / 数字 I/O (D) 切换									
					端口 5		端口 2							
					ANI9 /P110	ANI8 /P150	ANI7 /P27	ANI6 /P26	ANI5 /P25	ANI4 /P24	ANI3 /P23	ANI2 /P22	ANI1 /P21	ANI0 /P20
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	A	A	D	D	D	D	D	D
0	0	1	1	1	A	A	A	D	D	D	D	D	D	D
0	1	0	0	0	A	A	D	D	D	D	D	D	D	D
0	1	0	0	1	A	D	D	D	D	D	D	D	D	D
0	1	0	1	0	D	D	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D
与上述不同的值					设置禁止									

- 注意事项**
1. 利用端口模式寄存器 2 和 15 (PM2、PM15)，设置输入模式下的用于 A/D 转换的通道。
 2. 不要去设置那些通过模拟输入通道选项寄存器 (ADS)、由 ADPC 设置为数字 I/O 的引脚。

4.4 端口功能操作

对输入模式和输出模式的端口操作是不同的，具体如下所示。

4.4.1 写入 I/O 端口

(1) 输出模式

使用传送指令对输出锁存器进行写操作，输出锁存器的内容从引脚输出。

一旦数据写入输出锁存器，它将一直保存到新数据被写入。

当复位信号产生时，输出锁存器的内容被清零。

(2) 输入模式

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

一旦数据写入输出锁存器，它将一直保存到新数据被写入。

当复位信号产生时，输出锁存器的内容被清零。

4.4.2 从 I/O 端口读取

(1) 输出模式

使用传送指令读取输出锁存器的内容。输出锁存器的内容不会改变。

(2) 输入模式

使用传送指令读取引脚状态。输出锁存器的内容不会改变。

4.4.3 I/O 端口的操作

(1) 输出模式

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

一旦数据写入输出锁存器，它将一直保存到新数据被写入。

当复位信号产生时，输出锁存器的内容被清零。

(2) 输入模式

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

当复位信号产生时，输出锁存器的内容被清零。

4.4.4 以不同电源电压（2.5 V、3 V）连接到外部设备

当端口 3 和 7 的部分工作在 $V_{DD} = 4.0V \sim 5.5V$ 时，与工作在 2.5V 或 3V 的供电电压的外部设备相连接的 I/O 可以实现。

关于输入，由端口输入模式寄存器（PIM3 和 PIM7）可以实现 CMOS/TTL 按位切换。

另外，关于输出，利用端口输出模式寄存器（POM3 和 POM7），通过将输出缓冲器切换为 N 沟道漏极开路 (V_{DD} 耐压)，就可支持不同的电源电压。

(1) 使用 UART0、UART1、CSI00、CSI01 和 CSI10 功能的 I/O 引脚时的设置过程

(a) 用于 2.5V 或 3V 的输入端口

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

UART0 的情况下: P74
 UART1 的情况下: P31
 CSI00 的情况下: P74、P75
 CSI01 的情况下: P71、P72
 CSI10 的情况下: P31、P32

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

备注 n = 3 和 7

(b) 用于 2.5V 或 3V 的输出端口

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

UART0 的情况下: P73
 UART1 的情况下: P30
 CSI00 的情况下: P73、P75
 CSI01 的情况下: P70、P72
 CSI10 的情况下: P30、P32

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

备注 n = 3 和 7

(2) 使用 IIC10 功能的 I/O 引脚时的设置过程

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

在简易 IIC10 的情况中： P31, P32

- <3> 设置相关端口的输出锁存为 1。
- <4> 将 POM3 寄存器的相应位设置为 1 来设置成 N 沟道漏极开路输出(V_{DD} 耐压) 模式。
- <5> 设置 PM3 寄存器的相应位为输出模式(数据 I/O 可以是输出模式)。
同时，输出数据为高电平，因此引脚为高阻状态。
- <6> 实现串行阵列单元操作，并且将此模式设置成简易 I^2C 模式。

4.5 使用复用功能时端口模式寄存器和输出锁存的设置

要使用端口的复用功能，按下表 4-11 设置端口模式寄存器和输出锁存器。

表 4-11. 使用复用功能时端口模式寄存器和输出锁存的设置(1/2)

引脚名称	复用功能		PM _{xx}	P _{xx}
	功能名称	I/O		
P10	TI02	输入	1	×
	TO02	输出	0	0
P11	TI03	输入	1	×
	TO03	输出	0	0
P12	TI04	输入	1	×
	TO04	输出	0	0
P13	TI05	输入	1	×
	TO05	输出	0	0
P20~P27 ^{注1}	ANI0~ANI7 ^{注1}	输入	1	×
P30	SO10	输出	0	1
	TxD1	输出	0	1
	TO11	输出	0	0
P31	SI10	输入	1	×
	RxD1	输入	1	×
	SDA10	I/O	0	1
	INTP1	输入	1	×
	TI09	输入	1	×
P32	SCK10	输入	1	×
		输出	0	1
	SCL10	I/O	0	1
	INTP2	输入	1	×
P40	TOOL0	I/O	×	×
P41	TOOL1	输出	×	×
P50	TI06	输入	1	×
	TO06	输出	0	0
P51	TI07	输入	1	×
	TO07	输出	0	0
P52	SLTI	输入	1	×
	SLTO	输出	0	0

备注 ×: 不必考虑
 PM_{xx}: 端口模式寄存器
 P_{xx}: 端口输出锁存

(注列在下页中)

表 4-11. 使用复用功能时端口模式寄存器和输出锁存的设置(2/2)

引脚名称	复用功能		PM _{xx}	P _{xx}
	功能名称	I/O		
P70	SO01	输出	0	1
	INTP4	输入	1	×
P71	SI01	输入	1	×
	INTP5	输入	1	×
P72	SCK01	输入	1	×
		输出	0	1
	INTP6	输入	1	×
P73	SO00	输出	0	1
	TxD0	输出	0	1
	TO10	输出	0	0
P74	SI00	输入	1	×
	RxD0	输入	1	×
	TI11	输入	1	×
P75	SCK00	输入	1	×
		输出	0	1
	TI11	输入	1	×
P80 [※]	CMP0P	输入	1	×
	TMOFF0	输入	1	×
	INTP3	输入	1	×
	PGAI [※]	输入	1	×
P81	CMP0M	输入	1	×
P82	CMP1P	输入	1	×
	TMOFF1	输入	1	×
	INTP7	输入	1	×
P83	CMP1M	输入	1	×
P120	INTP0	输入	1	×
	EXLVI	输入	1	×
P150~P151 [※]	ANI8~ANI9 [※]	输入	1	×

备注 ×: 不必考虑
 PM_{xx}: 端口模式寄存器
 P_{xx}: 端口输出锁存

(注列在下页。)

注 ANI0/P20 至 ANI7/P27、ANI8/P150 至 ANI9/P151 和 PGAI/P80 引脚的功能可通过 A/D 端口配置寄存器 (ADPC)、模拟输入通道选项寄存器 (ADS)、PM2、PM15 和 PM8 选择。

表 4-12. ANI0/P20 至 ANI7/P27、ANI8/P150 至 ANI9/P151 和 PGAI/P80 引脚的设置功能

ADPC	PM2、PM15、 PM8	ADS	ANI0/P20 至 ANI7/P27、 ANI8/P150 至 ANI9/P151 和 PGAI/P80 引脚
数字 I/O 选择	输入模式	–	数字输入
	输出模式	–	数字输出
模拟输入选择	输入模式	选择 ANI。	模拟输入（转换）
		不要选择 ANI。	模拟输入（不转换）
	输出模式	选择 ANI。	设置禁止
		不要选择 ANI。	

4.6 端口寄存器n (Pn)的 1 位操作指令的注意事项

当对一个端口执行 1 位操作指令，该端口既有输入功能又有输出功能，没被操作的输入端口的输出锁存值可能会被写入目标位以外。

所以，建议在切换一个端口从输入模式到输出模式时，要重写输出锁存。

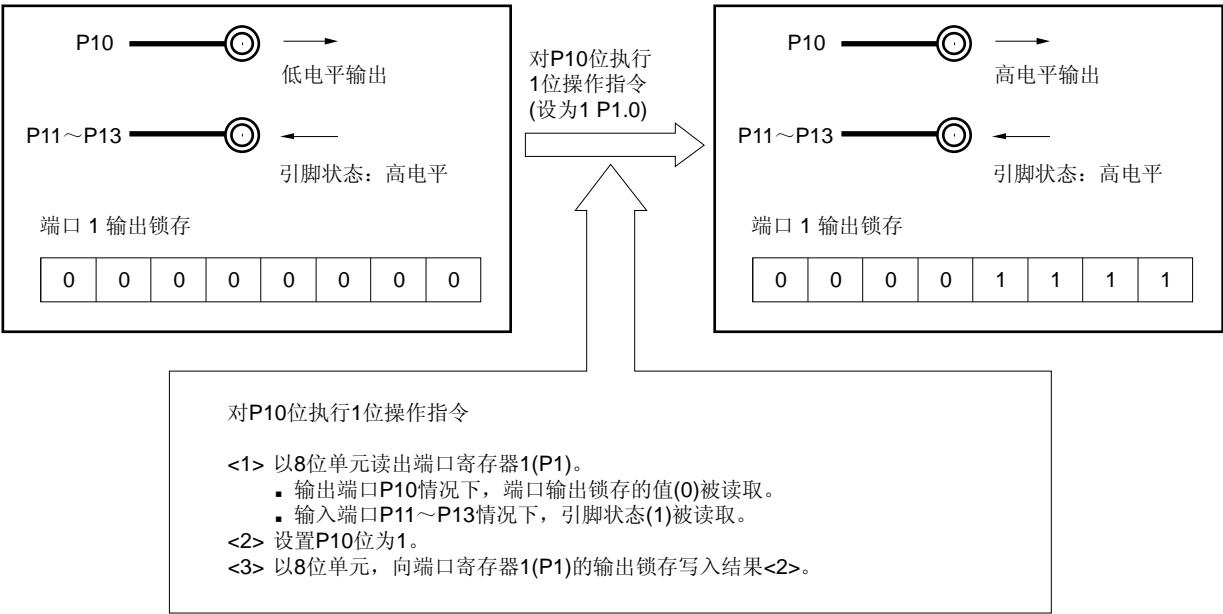
<示例> 当 P10 作为输出端口，P11~P13 作为输入端口 (所有引脚状态均为高电平)，并且该端口 1 锁存值为 00H 时，如果 P10 输出端口的输出通过一位操作指令使其由低电平变到高电平，那么，该端口锁存的值是 0FH。

解释: 从 PMnm 位是 1 的端口 Pn 寄存器写入和读出的目标值分别是输出锁存和引脚状态。
在μPD79F9211 中，1 位操作指令是按照如下次序执行的。

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

在第<1>步中，输出端口上的 P10 的输出锁存值(0)被读出，同时，输入端口上的 P11~P13 引脚的状态也被读出。如果此时 P11 至 P13 引脚状态为高电平，那么读出的值为 0EH。
通过第<2>步的操作，使该值变为 0FH。
通过第<3>步的操作，0FH 写入到输出锁存中去。

图 4-26. 位操作指令(P10)



5.1 时钟发生器的功能

时钟发生器用于产生时钟，并提供给 CPU 和外部硬件设备。
可以使用以下三种系统时钟和时钟振荡器。

(1) 主系统时钟

<1> X1 振荡器

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

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

<R>

<2> 内部高速振荡器

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

<R>

<3> 40 MHz 内部高速振荡器

该振荡电路产生 $f_{DSC} = 40$ MHz（典型值）的时钟。可以通过设置 40MHz 内部高速振荡器控制寄存器（DSCCTL）的第 0 位（DSCON）为 1 来开始振荡，设置 DSCON 为 0 来停止振荡。

复位释放后，8MHz 内部高速振荡器自动开始振荡。之后，当 DSCCTL 寄存器的位 DSCON 设置为 1 时，40 MHz 内部高速振荡开始振荡。

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

通过设置 MCM0（系统时钟控制寄存器（CKC）的第 4 位）可以选择高速系统时钟（X1 时钟或者外部主系统时钟）或内部高速振荡时钟作为主系统时钟。此外，通过设置 SELDSC（40 MHz 高速振荡控制寄存器（DSCCTL）的位 2），可选择 40 MHz 高速振荡时钟。

(2) 副系统时钟

• XT1 时钟振荡器

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

备注 fx: X1 时钟振荡频率
 f_{IH}: 内部高速振荡时钟频率
 fdsc: 40 MHz 内部高速振荡时钟频率
 f_{EX}: 外部主系统时钟频率
 f_{SUB}: 副系统时钟频率

(3) 内部低速振荡时钟（专用于看门狗定时器的时钟）

• 内部低速振荡器

此电路振荡时钟 $f_{IL} = 30\text{ kHz}$ （典型值）。
内部低速振荡时钟不能用作 CPU 时钟。工作在内部低速振荡时钟的硬件只有看门狗定时器。
当看门狗定时器停止时，振荡停止。

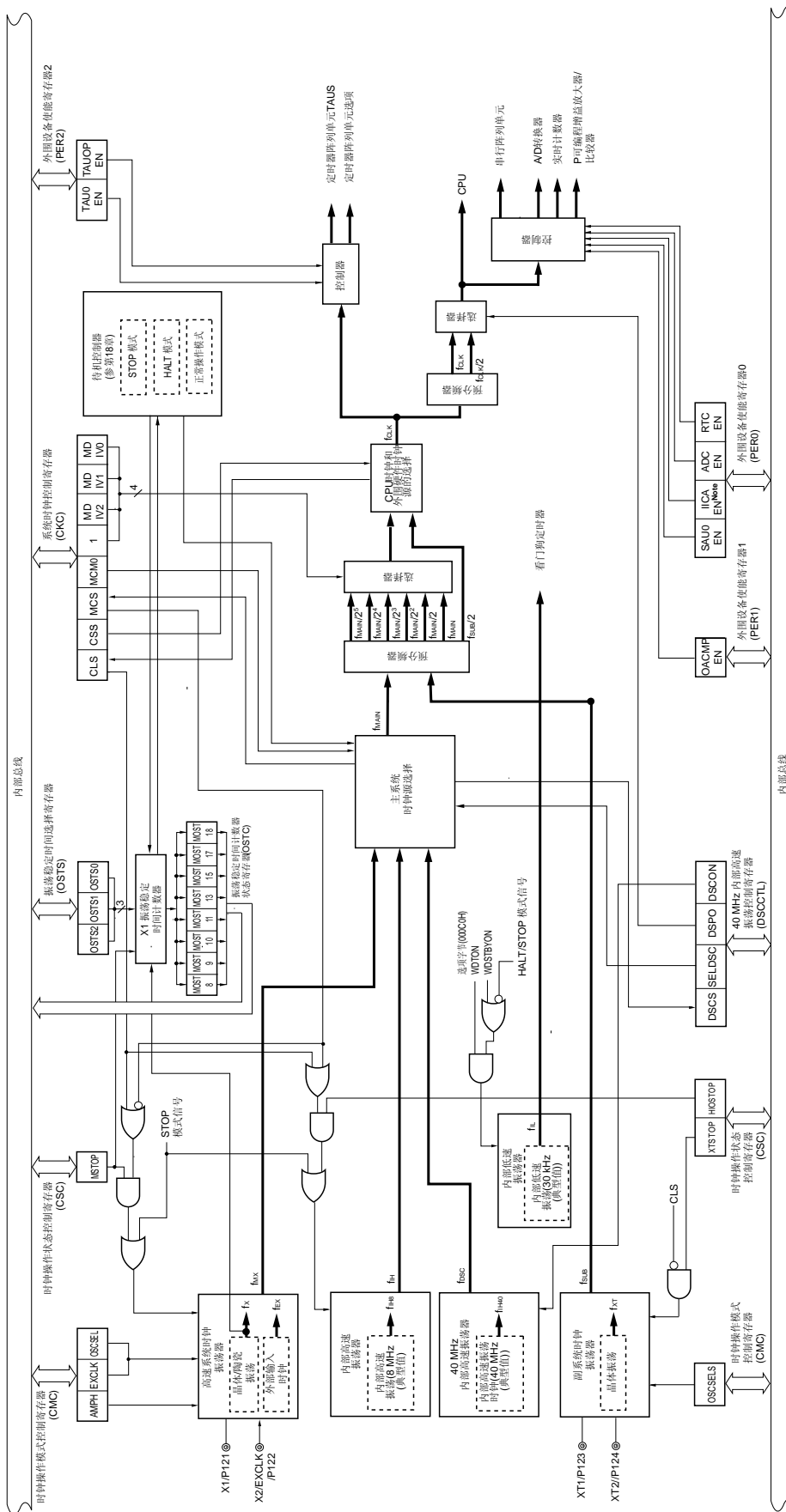
备注 1. f_{IL}: 内部低速振荡时钟频率
 2. 以下情况看门狗定时器停止。
 • 当选项字节（000C0H）的第 4 位（WDTON）= 0
 • 当选项字节（000C0H）的第 4 位（WDTON）= 1 和第 0 位（WDSTBYON）= 0 时，如果执行 HALT 或 STOP 指令

5.2 时钟发生器的配置

时钟发生器包括以下硬件。

表 5-1. 时钟发生器的配置

项目	配置
控制寄存器	时钟操作模式控制寄存器（CMC） 时钟操作状态控制寄存器（CSC） 振荡稳定时间计数器状态寄存器（OSTC） 振荡稳定时间选择寄存器（OSTS） 系统时钟控制寄存器（CKC） 40 MHz 高速振荡控制寄存器（DSCCTL） 外围设备使能寄存器 0、1、2（PER0、PER1、PER2） 操作速度模式控制寄存器（OSMC）
振荡器	X1 振荡器 XT1 振荡器 内部高速振荡器 内部低速振荡器



备注	fx:	X1 时钟振荡频率
	fIH:	内部高速振荡时钟频率
	fDSC:	40 MHz 内部高速振荡时钟频率
	fEX:	外部主系统时钟频率
	fMX:	高速系统时钟频率
	fMAIN:	主系统时钟频率
	fXT:	XT1 时钟振荡频率
	fSUB:	副系统时钟频率
	fCLK:	CPU/外围硬件时钟频率
	fIL:	内部低速振荡时钟频率

5.3 控制时钟发生器的寄存器

以下八个寄存器用于控制时钟发生器。

- 时钟操作模式控制寄存器 (CMC)
- 时钟操作状态控制寄存器 (CSC)
- 振荡稳定时间计数器状态寄存器 (OSTC)
- 振荡稳定时间选择寄存器 (OSTS)
- 系统时钟控制寄存器 (CKC)
- 40 MHz 内部高速振荡控制寄存器 (DSCCTL)
- 外围设备使能寄存器 0, 1, 2 (PER0, PER1, PER2)
- 操作速度模式控制寄存器 (OSMC)

(1) 时钟操作模式控制寄存器 (CMC)

此寄存器用于设置 X1/P121、X2/EXCLK/P122、XT1/P123 和 XT2/P124 引脚的操作模式，并选择振荡器。
复位释放后 CMC 只能由 8 位存储器操作指令写入。此寄存器可由 1 位或 8 位存储器操作指令读取。
复位后这个寄存器的值为 00H。

图 5-2. 时钟操作模式控制寄存器 (CMC) 的格式

地址: FFFA0H 复位后: 00H R/W

<R>

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

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

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

AMPH	高速系统时钟振荡频率的控制
0	2 MHz ≤ f _{MX} ≤ 10 MHz
1	10 MHz < f _{MX} ≤ 20 MHz

- 注意事项
1. CMC 只能在复位释放后通过 8 位存储器操作指令写入。

2. 复位释放后，X1 或 XT1 振荡开始前，通过设置时钟操作状态控制寄存器 (CSC) 设置 CMC。

3. 如果 X1 时钟振荡频率超过 10MHz，确保设置 AMPH 为 1。

4. 即使当 CMC 寄存器已设定为默认值使用时，为防止程序循环的错误，建议复位释放后将其置为默认值 (00H)。

备注

f_{MX}: 高速系统时钟频率

(2) 时钟操作状态控制寄存器 (CSC)

<R> 该寄存器用来控制高速系统时钟、内部高速振荡时钟和副系统时钟的运行（除 40 MHz 内部高速振荡时钟和内部低速振荡时钟）。

CSC 可由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器设置为 C0H。

图 5-3. 时钟操作状态控制寄存器 (CSC) 的格式

地址: FFFA1H 复位后: C0H R/W

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

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

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

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

- 注意事项
- 1. 复位释放后，在启动由 **MSTOP** 设置的 **X1** 振荡或由 **XTSTOP** 设置的 **XT1** 振荡前，设置时钟操作模式控制寄存器 (**CMC**)。
 - 2. 要使由 **MSTOP** 设置的 **X1** 振荡开始，可通过使用振荡稳定时间计数状态寄存器 (**OSTC**) 来检查 **X1** 时钟的振荡稳定时间。
 - 3. 不要停止通过 **OSC** 寄存器为 **CPU** 外围硬件时钟 (**f_{CLK}**) 选择的时钟。
 - 4. 停止时钟振荡（外部时钟输入无效）的寄存器标志设置以及停止时钟振荡前的条件如下所示。

表 5-2. 停止时钟振荡前的条件和标志设置

时钟	停止时钟前的条件 (外部时钟输入无效)	CSC 寄存器标志 的设置
X1 时钟	<ul style="list-style-type: none"> • CLS = 0 且 MCS = 0 • CLS = 1 (CPU 和外围硬件时钟工作在除高速系统时钟以外的时钟。) 	MSTOP = 1
外部主系统时钟		
副系统时钟	<ul style="list-style-type: none"> • CLS = 0 (CPU 和外围硬件时钟工作在除副系统时钟以外的时钟。) 	XTSTOP = 1
内部高速振荡时钟	<ul style="list-style-type: none"> • CLS = 0 且 MCS = 1 • CLS = 1 (CPU 和外围硬件时钟工作在除内部高速振荡器时钟以外的时钟。) 	HIOSTOP = 1

(3) 振荡稳定时间计数器状态寄存器 (OSTC)

这是表明 X1 时钟振荡稳定时间计数器的计数状态的寄存器。

在下列情况下，可检测 X1 时钟振荡稳定时间。

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

可由 1 位或 8 位存储器操作指令读取 OSTC。

复位信号产生时，STOP 指令和 MSTOP (CSC 寄存器的第 7 位) = 1 将 OSTC 清零为 00H。

备注 振荡稳定时间计数器在以下情况下开始计数。

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

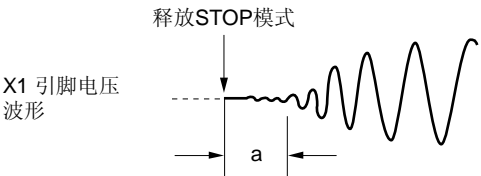
图 5-4. 振荡稳定时间计数器状态寄存器（OSTC）的格式

地址: FFFA2H 复位后: 00H R

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

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

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



备注 f_x : X1 时钟振荡频率

(4) 振荡稳定时间选择寄存器 (OSTS)

此寄存器用于选择 STOP 模式释放时的 X1 时钟振荡稳定等待时间。

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

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

可由 8 位存储器操作指令设置 OSTS。

复位信号产生将 OSTS 置为 07H。

图 5-5. 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFFA3H 复位后: 07H R/W

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

OSTS2	OSTS1	OSTS0		振荡稳定时间选择	
				$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$
0	0	0	$28/f_x$	$25.6 \mu\text{s}$	设置禁止
0	0	1	$29/f_x$	$51.2 \mu\text{s}$	$25.6 \mu\text{s}$
0	1	0	$210/f_x$	$102.4 \mu\text{s}$	$51.2 \mu\text{s}$
0	1	1	$211/f_x$	$204.8 \mu\text{s}$	$102.4 \mu\text{s}$
1	0	0	$213/f_x$	$819.2 \mu\text{s}$	$409.6 \mu\text{s}$
1	0	1	$215/f_x$	3.27 ms	1.64 ms
1	1	0	$217/f_x$	13.11 ms	6.55 ms
1	1	1	$218/f_x$	26.21 ms	13.11 ms

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

2. 禁止设置振荡稳定时间为 $20 \mu\text{s}$ 或更短。

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

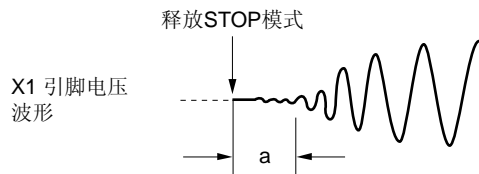
4. 在 X1 时钟振荡稳定期间，不要修改 OSTS 寄存器的值。

5. 振荡稳定时间计数器的计数达到 OSTS 设置的振荡稳定时间。

在如下情况，设置 OSTS 的振荡稳定时间的值大于振荡开始后 OSTC 寄存器检测到的计数值。

- 在内部高速振荡时钟或副系统时钟用作 CPU 时钟期间，如果 X1 时钟开始振荡。
- 当内部高速振荡时钟用作 X1 时钟振荡的 CPU 时钟时，如果进入 STOP 模式然后释放。
(注，而且，仅由 OSTS 设置的振荡稳定时间状态在 STOP 模式释放后被设置到 OSTC)

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



备注 f_x : X1 时钟振荡频率

(5) 系统时钟控制寄存器 (CKC)

此寄存器用于选择 CPU/外围硬件时钟和分频比。

CKC 可由 1 位或 8 位存储器操作指令设置。

复位信号产生将此寄存器置为 09H。

图 5-6. 系统时钟控制寄存器 (CKC) 的格式

地址: FFFA4H 复位后: 09H R/W^{注1}

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

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

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

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

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

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

(注意事项 1 至 3 列于下页中)

- 注意事项
1. 第 3 位必须设置为 1。
 2. 由 CSS、MCM0 和 MDIV2 至 MDIV0 设置的时钟供给 CPU 和外围硬件。如果 CPU 时钟改变，供给外围硬件（除实时计数器和看门狗定时器外）的时钟也同时改变。因此，当 CPU/外围工作硬件时钟改变时，停止每个外围设备的功能。
 3. 如果把外围硬件时钟用作副系统时钟，则我们不能保证 A/D 转换器的操作。外围硬件的操作特性，参见描述各种外围硬件的章节及第 26 章 电气特性（目标值）。

在 μ PD79F9211 中执行速度最快的指令是单周期指令。CPU 时钟（fCLK）与最小指令执行时间的关系如表 5-3 所示。

表 5-3. CPU 时钟和最小指令执行时间之间的关系

CPU 时钟 (由 MDIV2 至 MDIV0 位设置的值)	最小指令执行时间: $1/f_{CLK}$			
	主系统时钟 (CSS = 0)			副系统时钟 (CSS = 1)
	高速系统时钟 (MCM0 = 1)		内部高速振荡时钟 (MCM0 = 0)	
	操作在 10 MHz	操作在 20 MHz	在 8 MHz (典型值) 操作	操作在 32.768 kHz
f_{MAIN}	0.1 μs	0.05 μs	0.125 μs (典型值)	—
$f_{MAIN}/2$	0.2 μs	0.1 μs	0.25 μs (典型值) (默认)	—
$f_{MAIN}/2^2$	0.4 μs	0.2 μs	0.5 μs (典型值)	—
$f_{MAIN}/2^3$	0.8 μs	0.4 μs	1.0 μs (典型值)	—
$f_{MAIN}/2^4$	1.6 μs	0.8 μs	2.0 μs (典型值)	—
$f_{MAIN}/2^5$	3.2 μs	1.6 μs	4.0 μs (典型值)	—
$f_{SUB}/2$	—		—	61 μs

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

<R>

(6) 40 MHz 内部高速振荡控制寄存器 (DSCCTL)

此寄存器控制 40MHz 内部高速振荡时钟 (DSC) 功能。
它可用于选择是否使用 40MHz 内部高速振荡时钟 (f_{DSC}) 作为外围硬件时钟，该外围硬件时钟支持 40MHz。
DSCCTL 只能由 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 00H。

图 5-7. 40 MHz 内部高速振荡控制寄存器 (DSCCTL) 的格式

地址: F00F6H 复位后: 00H R/W^注

符号	7	6	5	4	<3>	<2>	<1>	<0>
DSCCTL	0	0	0	0	DSCS	SELDSC	DSPO	DSCON
DSCS	CPU/外围硬件时钟 (f _{CLK}) 状态标志							
0	通过模式, 时钟							
1	DSC 模式							
SELDSC	选择是否把 DSC 输出作为 CPU/外围硬件时钟 (f _{CLK})							
0	由 CKC 寄存器选择时钟 (时钟直通模式)。							
1	提供 40MHz 内部高速振荡时钟 (f _{DSC}) (DSC 模式)。							
DSPO	40 MHz 操作选择位							
0	除了当 f _{CLK} = 40 MHz (典型值) 以外							
1	当 f _{CLK} = 40 MHz (典型值) 向支持 40 MHz 的外围硬件提供一个 f _{CLK} 时钟, 向一个 CPU 或不支持 40 MHz 的外围硬件提供一个 f _{CLK} /2 时钟。							
DSCON	40MHz 内部高速振荡时钟 (f _{DSC}) 运行允许/禁止							
0	禁止操作。							
1	使能操作。							

注 第 3 位只读。

与选择 CPU/外围硬件时钟相关的位（f_{CLK}）如下所示。

<R>

表 5-4. CPU/外围硬件时钟（f_{CLK}）和位设置之间的关系

CMC 寄存器			CSC 寄存器			CKC 寄存器		DSCCTL 寄存器			CPU/外围硬件时钟（f _{CLK} ）
EXCLK	OSCSEL	OSCSELS	MSTOP	XTSTOP	HIOSTOP	CSS	MCM0	SELDSC	DSPO	DSCON	
0	1	×	0	×	×	0	1	0	0	×	高速系统时钟 （X1振荡） （2至20 MHz）
1	1	×	0	×	×	0	1	0	0	×	高速系统时钟 （外部输入时钟） （2至20 MHz）
×	×	×	×	×	0	0	0	0	0	×	内部高速振荡时钟 （8 MHz（典型值.））
×	×	1	×	0	×	1	×	0	0	×	副系统时钟（32.768 kHz）
×	×	×	×	×	0	0	0	1	1	1	40 MHz内部高速振荡时钟 （40 MHz（典型值.）） ^{注 2}

- 注
1. 当 CSS 设置为 1 时，禁止改变 MCM0 位的值。
 2. 禁止提供 40 MHz 时钟，因为 CPU 或外围硬件不支持 40 MHz，
通过设置 DSPO（DSCCTL 寄存器的位 1）为 1 使提供的时钟减半（f_{CLK}/2）。

备注 ×: 不必考虑

(7) 外围设备使能寄存器 0, 1, 2 (PER0, PER1, PER2)

这些寄存器用于外围硬件模块指令的允许和禁止。为了降低功耗和噪声，不使用的硬件时钟也被停止。
PER0、PER1、PER2 可由 1 位或 8 位存储器操作指令设置。
复位信号产生将此寄存器置为 00H。

图 5-8. 外围设备使能寄存器的格式 (1/3)

地址: F00F0H 复位后: 00H R/W							
符号	<7>	6	<5>	<4>	3	<2>	1 0
PER0	RTCEN	0	ADCEN	0	0	SAU0EN	0 0
地址: F00F1H 复位后: 00H R/W							
符号	7	6	5	4	<3>	2	1 0
PER1	0	0	0	0	OACMPEN	0	0 0
地址: F00F2H 复位后: 00H R/W							
符号	7	6	5	4	3	2	<1> <0>
PER2	0	0	0	0	0	0	TAUOPEN TAU0EN
<R>	RTCEN	实时计数器 (RTC) 输入时钟的控制 [✱]					
	0	停止输入时钟供给。 • 由实时计数器 (RTC) 使用的 SFR 不能写入。 • 实时计数器 (RTC) 在复位状态。					
	1	供给输入时钟 • 用于实时计数器 (RTC) 的 SFR 可读写。					
<R>	ADCEN	A/D 转换器输入时钟的控制					
	0	停止输入时钟供给。 • 被 A/D 转换器使用的 SFR 是不可重写。 • A/D 转换器在复位状态。					
	1	供给输入时钟 • 用于 A/D 转换器的 SFR 可读写。					

注 当通过从 CPU 访问实时计数器 (RTC) 来使用寄存器时，输入时钟可由 RTCEN 控制。
 实时计数器 (RTC) 来自 CPU。RTCEN 不能控制供给 RTC 的操作时钟 (f_{SUB})。

- 注意事项 确保下列位清为 0。
- PER0 寄存器的位 0、1、3、4 和 6
 - PER1 寄存器的 0 至 2 位及 4 至 7 位
 - PER2 寄存器的 2 至 7 位

图 5-8. 外围设备使能寄存器的格式（2/3）

地址：F00F0H 复位后：00H R/W

寄存器	<7>	6	<5>	<4>	3	<2>	1	0
PER0	RTCEN	0	ADCEN	0	0	SAU0EN	0	0

地址：F00F1H 复位后：00H R/W

寄存器	7	6	5	4	<3>	2	1	0
PER1	0	0	0	0	OACMPEN	0	0	0

地址：F00F2H 复位后：00H R/W

寄存器	7	6	5	4	3	2	<1>	<0>
PER2	0	0	0	0	0	0	TAUOPEN	TAU0EN

SAU0EN	串行阵列单元输入时钟的控制
0	停止输入时钟供给。 <ul style="list-style-type: none">不能写入串行阵列单元使用的 SFR。串行阵列单元处于复位状态。
1	供给输入时钟 <ul style="list-style-type: none">可以对串行阵列单元使用的 SFR 进行读写。

OACMPEN	控制可编程增益放大器输入时钟
0	停止输入时钟供给。 <ul style="list-style-type: none">不能写入可编程增益放大器使用的 SFR。可编程增益放大器处于复位状态。
1	供给输入时钟 <ul style="list-style-type: none">可以对运算放大器使用的 SFR 进行读写。

- 注意事项 确保下列位清为 0。
- PER0 寄存器的位 0、1、3、4 和 6
 - PER1 寄存器的 0~2 位及 4~7 位
 - PER2 寄存器的 2~7 位

图 5-8. 外围设备使能寄存器的格式（3/3）

地址：F00F0H 复位后：00H R/W

寄存器	<7>	6	<5>	<4>	3	<2>	1	0
PER0	RTCEN	0	ADCEN	0	0	SAU0EN	0	0

地址：F00F1H 复位后：00H R/W

寄存器	7	6	5	4	<3>	2	1	0
PER1	0	0	0	0	OACMPEN	0	0	0

地址：F00F2H 复位后：00H R/W

寄存器	7	6	5	4	3	2	<1>	<0>
PER2	0	0	0	0	0	0	TAUOPEN	TAU0EN

TAUOPEN	变频控制功能输入时钟的控制
0	停止输入时钟供给。 <ul style="list-style-type: none">不能写入变频控制功能使用的 SFR。反相器控制功能处于复位状态。
1	供给输入时钟 <ul style="list-style-type: none">可以对变频控制功能使用的 SFR 进行读写。

TAU0EN	定时器阵列单元 TAUS 输入时钟的控制
0	停止输入时钟供给。 <ul style="list-style-type: none">由定时器阵列单元 TAUS 使用的 SFR 不能写入。定时器阵列单元 TAUS 处于复位状态。
1	供给输入时钟 <ul style="list-style-type: none">可以对定时器阵列单元 TAUS 使用的 SFR 进行读写。

- 注意事项 确保下列位清为 0。
- PER0 寄存器的位 0、1、3、4 和 6
 - PER1 寄存器的 0~2 位及 4~7 位
 - PER2 寄存器的 2~7 位

(8) 操作速度模式控制寄存器 (OSMC)

此寄存器用于控制 flash 存储器高速操作的递升电路。

如果微控制器在系统时钟 10MHz 或更小值的低速下操作，通过设置此寄存器到默认值 (00H) 可以降低功耗。

OSMC 可由 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 5-9. 操作速度模式控制寄存器 (OSMC) 的格式

地址: F00F3H 复位后: 00H R/W

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

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

- 注意事项**
- 只有当复位释放后，OSMC 才能通过 8 位存储器操作指令写入。
 - 在执行下列两个操作之前，把“1”写入 FSEL。
 - 修改时钟优先于分频 f_{CLK} 为一个不是 f_{IH} 的时钟。
 - 操作 DMA 控制器。
 - 当向 FSEL 标志写入 1 时，CPU 等待。

当 f_{CLK} = f_{IH} 时等待时间为 15 μs ~ 20 μs (目标值)，当 f_{CLK} = f_{IH}/2 时 30 μs ~ 40 μs (目标值)。

但是，即使当 CPU 等待时，f_x 的振荡稳定时间计数可继续。
 - 为了增加 f_{CLK} 至 10 MHz 或更高，设置 FSEL 为“1”，然后在 2 个或更多时钟之后改变 f_{CLK}。
 - 即使当 FSEL = 1 时，系统时钟可以在 10 MHz 或更小频率下操作。

当设置 FSEL 为“1”时，同时 V_{DD} ≥ 2.25 V。

当设置 FSEL = 1，确保下列计时中 V_{DD} ≥ 2.25 V，即使 f_{CLK} 被分频。

 - 当从 STOP 模式中释放 f_{IH} 或 f_{EX} 出来时，此模式为 f_{CLK} 选择
 - 当把 f_{CLK} 从 f_{SUB} 切换到 f_{MAIN}

5.4 系统时钟振荡器

5.4.1 X1 振荡器

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

同样可以输入外部时钟。此时，把时钟信号输入到 EXCLK 引脚。

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

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

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

当引脚也不用作输入端口引脚时，参见表 2-2 未使用引脚的连接。

图 5-10 为 X1 振荡器的外部电路示例。

图 5-10. X1 振荡器的外部电路的示例



注意事项在下页中列出。

5.4.2 XT1 振荡器

具有晶体振荡器的 XT1 振荡器（标准：32.768 kHz）连接到 XT1 和 XT2 引脚的方式振荡。

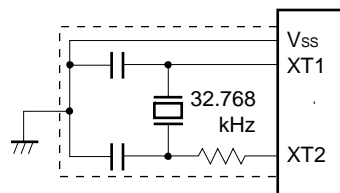
要使用 XT1 振荡器，设置时钟操作模式控制寄存器（CMC）的第 4 位（OSCSELS）为 1。

当 XT1 振荡器不使用时，设置输入端口模式（OSCSELS = 0）。

当引脚也不用作输入端口引脚时，参见表 2-2 未使用引脚的连接。

图 5-11 为 XT1 振荡器的外部电路示例。

图 5-11. XT1 振荡器（晶体振荡）的外部电路示例



注意事项在下页中列出。

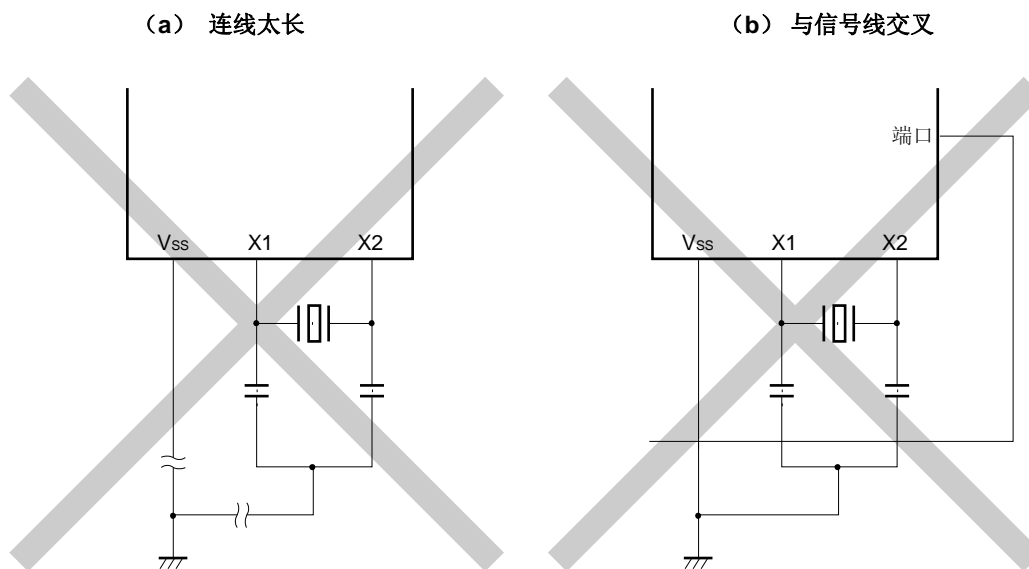
注意事项 1. 在使用 X1 振荡器和 XT1 振荡器时，为防止电线电容的不利影响，图 5-10 和图 5-11 中虚线包围的部分应按照如下方法配线。

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

注意 XT1 振荡器被设计成低振幅电路，以降低功耗。

图 5-12 为不正确的振荡器连接示例。

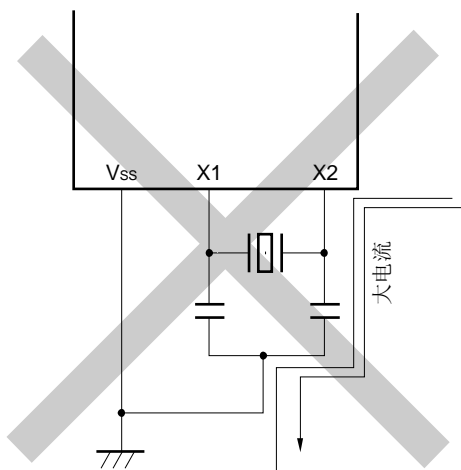
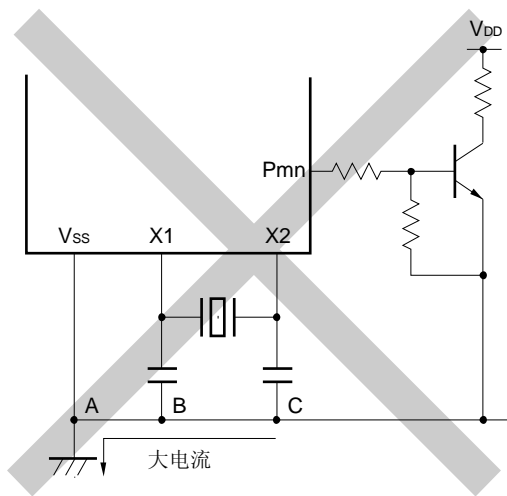
图 5-12. 不正确的振荡器连接示例 (1/2)



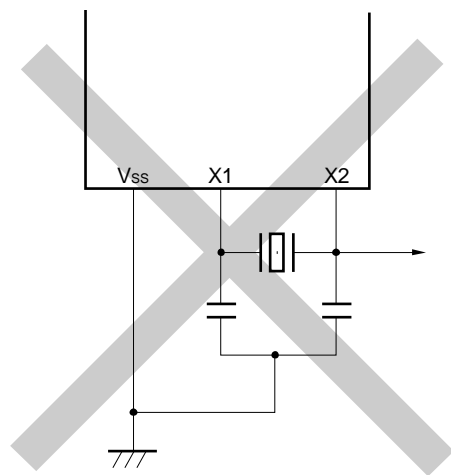
备注 在使用副系统时钟时，分别用 XT1 和 XT2 代替 X1 和 X2。串联电阻也接在 XT2 这边。

图 5-12. 不正确的振荡器连接示例 (2/2)

(c) 连接线周围有变化大的电流

(d) 电流经过振荡器的地线
(A、B、C 三点电势波动)

(e) 获取信号



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

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

5.4.3 内部高速振荡器

<R> μ PD79F9211 (8MHz (典型值)) 产品中内置了高速振荡器。可以通过时钟操作状态控制寄存器 (CSC) 的第 0 位 (HIOSTOP) 控制振荡。

复位释放后, 内部高速振荡器自动开始振荡。

5.4.4 40MHz 内部高速振荡器

<R> μ PD79F9211 (40MHz (典型值)) 产品中内置了 40MHz 高速振荡器。可以通过 40MHz 内部高速振荡器控制寄存器 (DSCCTL) 的第 0 位 (DSCON) 来控制振荡。

复位释放后, 8MHz 内部高速振荡器自动开始振荡。之后, 当 DSCCTL 寄存器的位 DSCON 设置为 1 时, 40 MHz 内部高速振荡开始振荡。

5.4.5 内部低速振荡器

μ PD79F9211 产品中包含内部低速振荡器。

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

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

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

5.4.6 预分频器

通过主系统时钟和副系统时钟分频, 预分频器产生一个 CPU/外围硬件时钟。

5.5 时钟发生器操作

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

- 主系统时钟 f_{MAIN}
 - 高速系统时钟 f_{MX}
 - X1 时钟 f_X
 - 外部主系统时钟 f_{EX}
- 内部高速振荡时钟 f_{IH}
- 40MHz 内部高速振荡时钟 f_{DSC}
- 副系统时钟 f_{SUB}
- 内部低速振荡时钟 f_{IL}
- CPU/外围硬件时钟 f_{CLK}

<R>

在 μ PD79F9211 中，当复位释放后内部高速振荡器开始输出时，CPU 开始操作，因此具有以下特点。

（1） 增强安全功能

如果 X1 时钟被默认设置为 CPU 时钟，在 X1 时钟遭到损坏或连接错误时设备不能操作，因此复位释放后也不操作。但是，如果 CPU 的初始时钟是内部高速振荡时钟，则复位释放后，内部高速振荡时钟可以启动设备。这样，复位源可以由软件检测，在异常出现期间，只需执行最少的安全操作 便可以安全关闭系统。

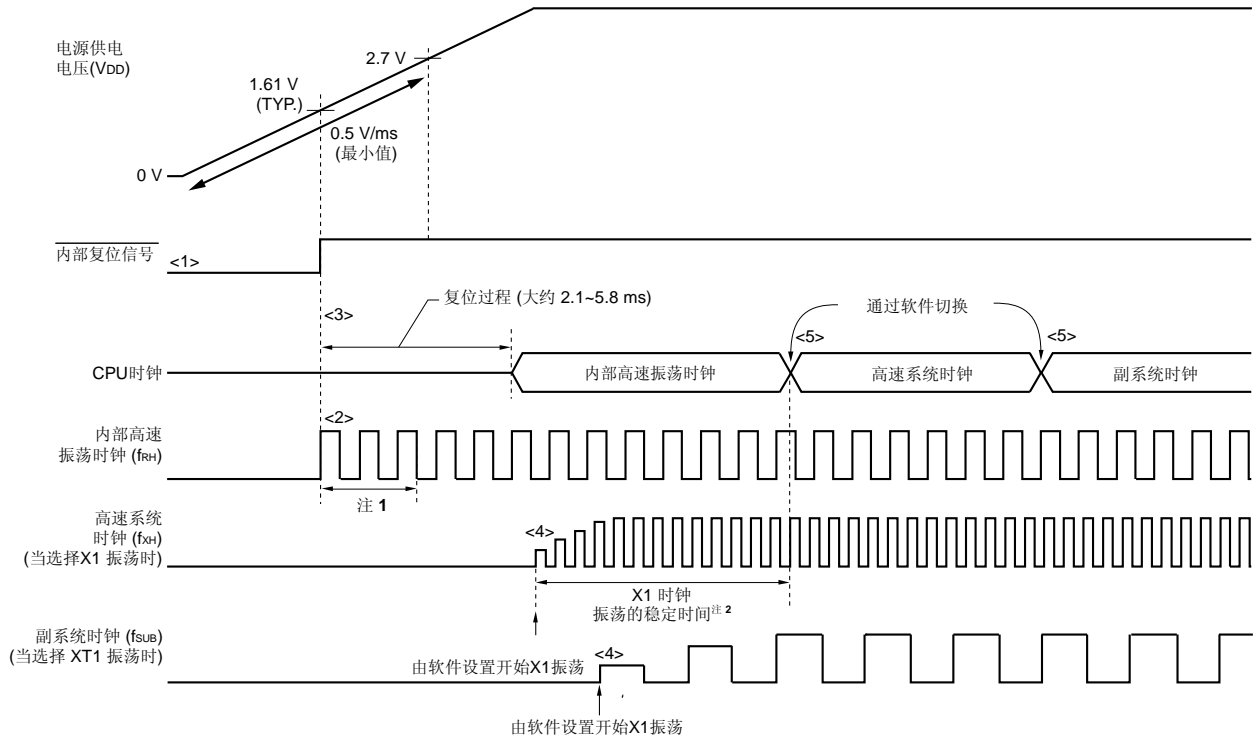
（2） 改善性能

由于 CPU 可以在不必等待 X1 时钟振荡稳定时间情况下就启动，所以总的性能得到了改善。

图 5-13~图 5-16 显示了供电电源打开时，时钟发生器的操作。

<R>

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



<1> 当电源开启时，通过上电清零（POC）电路产生一个内部复位信号。

<2> 当供电电压超过 1.61V（典型值），复位释放并且内部高速振荡器自动开始振荡。

<R> <3> 当电源电压上升速度为 0.5 V/ms（最小值）时，CPU 在复位过程（如在复位释放后等待电源电压或稳压器稳定）之后开始运行在内部高速振荡时钟上。

<4> 通过软件设置 X1 或 XT1 时钟的振荡开始（参见 5.6.1 高速系统时钟控制示例中的（1）和 5.6.3 副系统时钟控制示例中的（1））。

<5> 当 CPU 时钟切换到 X1 或 XT1 时钟时，等待时钟振荡稳定，然后通过软件设置切换（参见 5.6.1 高速系统时钟控制示例中的（3）和 5.6.3 副系统时钟控制示例中的（2））。

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

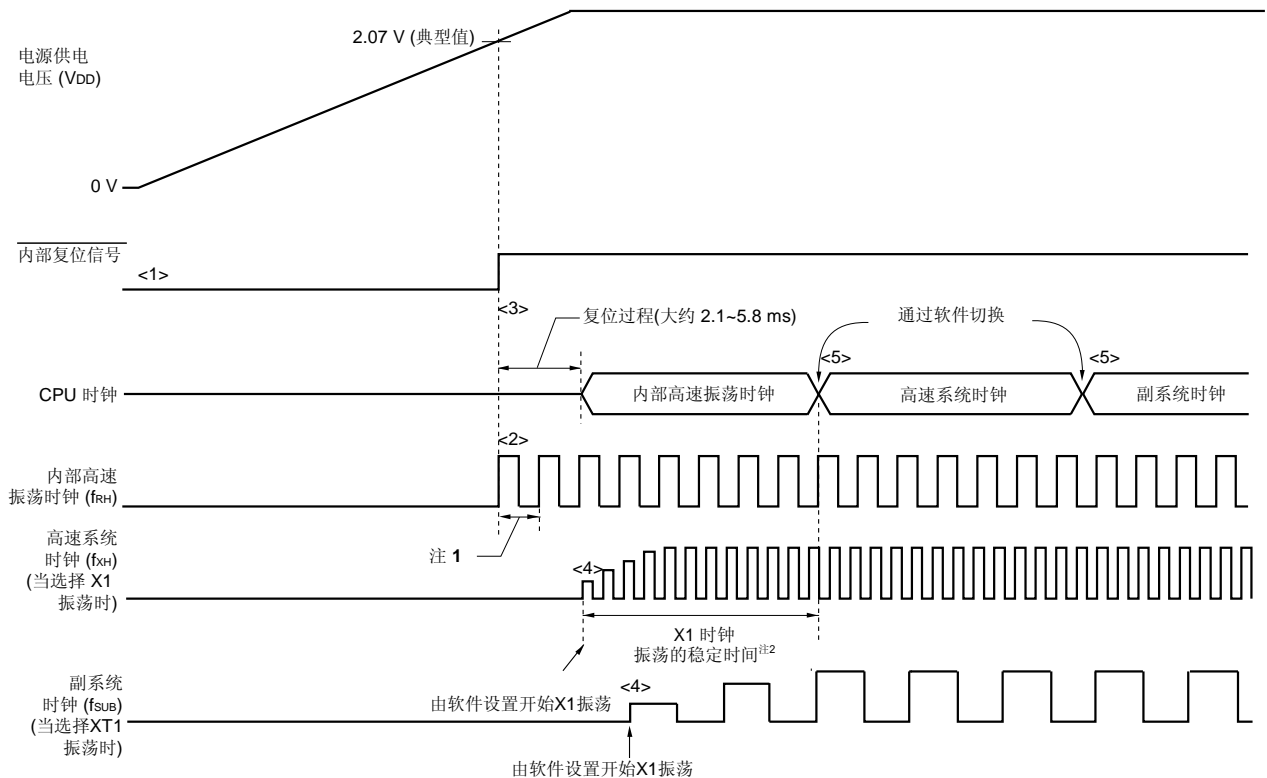
注意事项 1. 在供电电压达到 2.7V 之前，如果电压上升斜率小于 0.5 V/ms（最大值），则从供电开始至电压达到 2.7V 为止，输入一个低电平到电源的 $\overline{\text{RESET}}$ 引脚，或者通过使用选项字节（LVIOFF = 0）设置 LVI 默认开始功能停止（见图 5-14）。通过这种方式，CPU 的操作时序与 <2> 及由 $\overline{\text{RESET}}$ 引脚释放复位后的时序相同，如图 5-13 所示。

2. 当使用 EXCLK 引脚的外部时钟输入时，不需要等待振荡稳定时间。

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

<R>

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



<1> 当电源开启时，通过上电清零（POC）电路产生一个内部复位信号。

<2> 当供电电压超过 2.07V（典型值），复位释放并且内部高速振荡器自动开始振荡。

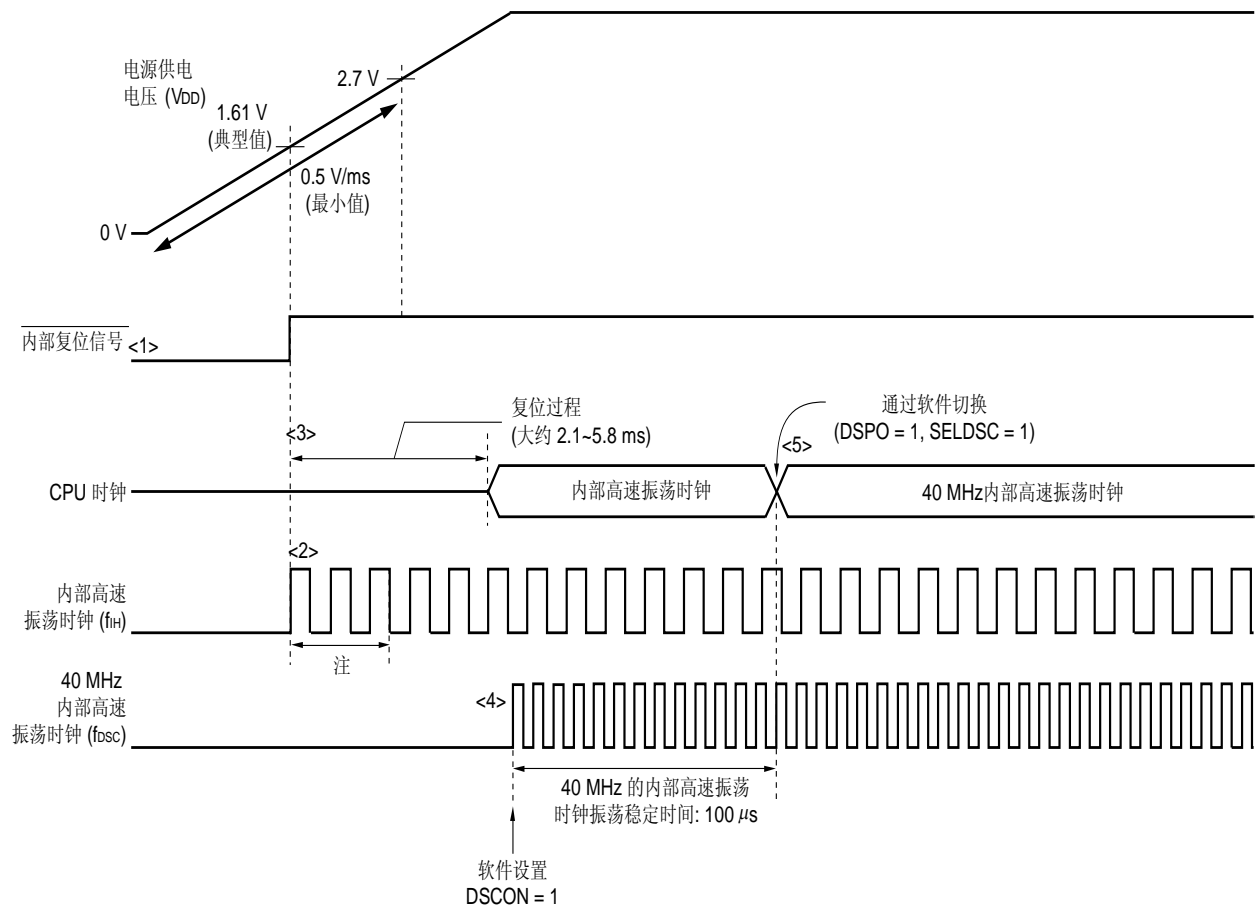
<3> 复位释放并进行了复位处理后，CPU 开始使用内部高速振荡时钟操作。

<4> 通过软件设置 X1 或 XT1 时钟的振荡开始（参见 5.6.1 高速系统时钟控制示例中的（1）和 5.6.3 副系统时钟控制示例中的（1））。

<5> 当 CPU 时钟切换到 X1 或 XT1 时钟时，等待时钟振荡稳定，然后通过软件设置切换（参见 5.6.1 高速系统时钟控制示例中的（3）和 5.6.3 副系统时钟控制示例中的（2））。

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

<R> 图 5-15. 供电电源打开时钟发生器的操作
(当设置 LVI 默认开始功能停止 (选项字节: LVIOFF = 1) 且改变至 40 MHz 内部高速振荡时钟时)



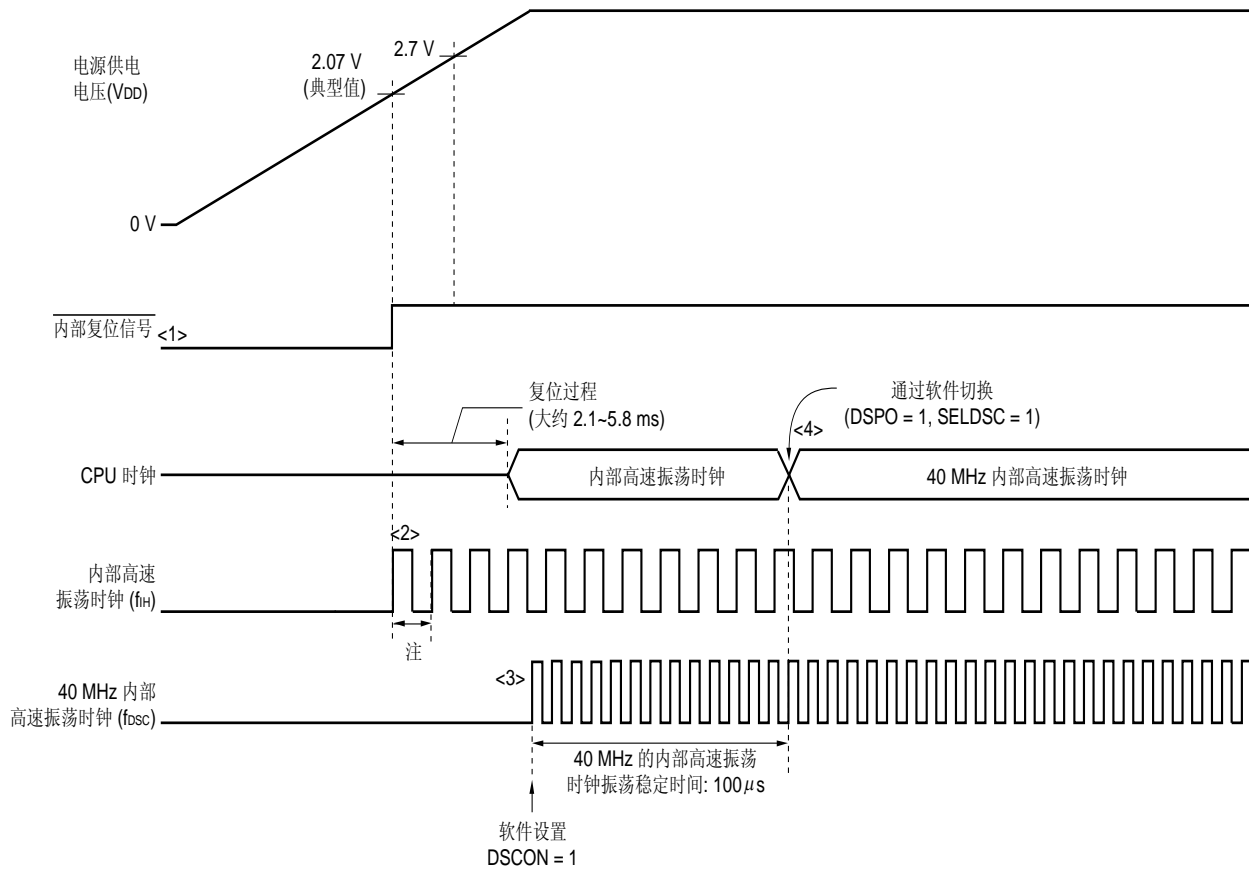
- <1> 当电源开启时，通过上电清零（POC）电路产生一个内部复位信号。
- <2> 当供电电压超过 1.61V（典型值），复位释放并且内部高速振荡器自动开始振荡。
- <R> <3> 当供电电压上升斜率为 0.5 V/ms（最小值）时，在复位过程（比如等待供电电压或复位释放后稳压器稳定）后，CPU 在内部高速振荡时钟上开始操作。
- <4> 通过软件设置 DSCON = 1。
- <5> 在等待 100 μs 后通过软件设置 DSPO = 1 和 SELDSC = 1 来转换时钟。

<R> 注 内部复位处理时间包括内部高速振荡时钟的振荡精确稳定时间。

注意事项 在供电电压达到 2.7V 之前，如果电压上升斜率小于 0.5 V/ms（最小值），则从供电开始至电压达到 2.7V 为止，输入低电平到电源的 RESET 引脚，或者通过使用选项字节（LVIOFF = 0）设置 LVI 默认开始功能停止（见图 5-16）。通过这种方式，CPU 的操作时序与 <2> 及由 RESET 引脚释放复位后的时序相同，如图 5-15 所示。

<R>

图 5-16. 供电电源打开时时钟发生器的操作
(当设置 LVI 默认开始功能允许 (选项字节: LVIOFF = 0) 且变更至 40MHz 内部高速振荡时钟时)



- <1> 当电源开启时，通过上电清零（POC）电路产生一个内部复位信号。
- <2> 当供电电压超过 2.07V（典型值），复位释放并且内部高速振荡器自动开始振荡。
- <3> 复位释放并进行了复位处理后，CPU 开始使用内部高速振荡时钟操作。
- <4> 通过软件设置 DSCON = 1。
- <5> 在等待 100 μs 后通过软件设置 DSPO = 1 和 SELDSC = 1 来转换时钟。

注 内部复位处理时间包括内部高速振荡时钟的振荡精确稳定时间。

注意事项 在供电电压达到 1.61（典型值）后，必需有一个电压振荡稳定时间。如果在供电电源振荡稳定时间内，电源电压从 1.61 V（典型值）上升到 2.07 V（典型值），在复位处理前自动产生供电电源振荡稳定时间。

5.6 控制时钟

5.6.1 控制高速系统时钟的示例

如下两种高速系统时钟可用。

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

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

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

下面描述的示例是以下情况时的设置过程示例。

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

(1) X1 时钟振荡时的设置过程示例

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

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

<R>

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
0	1	0	0/1	0	0	0	0

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

<R>

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
0	1	0	0/1	0	0	0	1

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

<2> 控制 X1 时钟振荡（CSC 寄存器）

若 MSTOP 被清零，则 X1 振荡器开始振荡。

<3> 等待 X1 时钟振荡稳定

检测 OSTC 寄存器并等待所需的时间。

在等待时间内，其它软件处理过程的执行可以使用内部高速振荡时钟。

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

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

2. 在供电电压达到时钟使用的可操作电压后，设置 X1 时钟（参见第 26 章 电气特性（目标值））。

(2) 使用外部主系统时钟时设置过程示例

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

<R>

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
1	1	0	0/1	0	0	0	0/1

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

<2> 控制外部主系统时钟输入 (CSC 寄存器)

MSTOP 清零时, 允许外部主系统时钟的输入。

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

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

2. 在供电电压达到时钟使用的可操作电压后, 设置外部主系统时钟 (参见第 26 章 电气特性 (目标值))。

(3) 使用高速系统时钟作为 CPU/外部硬件时钟时的设置过程示例

<1> 设置高速系统时钟振荡[‡]

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

注 当高速系统时钟已经运行时, 不需要设置<1>。

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

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

注 f_{MX} < 4 MHz 时设置禁止。

<3> 如果某些外围硬件模块不使用，可以停止硬件模块的输入时钟供给。

(PER0 寄存器)

RTCEN	0	ADCEN	0	0	SAU0EN	0	0
-------	---	-------	---	---	--------	---	---

(PER1 寄存器)

0	0	0	0	OACMPEN	0	0	0
---	---	---	---	---------	---	---	---

(PER2 寄存器)

0	0	0	0	0	0	TAUOPEN	TAU0EN
---	---	---	---	---	---	---------	--------

xxxEN	输入时钟控制
0	停止输入时钟供给。
1	供给输入时钟

注意事项 确保下列位清为 0。

- PER0 寄存器的位 0、1、3、4 和 6
- PER1 寄存器的 0 至位及 4 至 7 位
- PER2 寄存器的 2 至 7 位

备注

RTCEN: 实时计数器输入时钟的控制

ADCEN: A/D 转换器输入时钟的控制

SAU0EN: 控制串行阵列单元输入时钟

OACMPEN: 控制可编程增益放大器输入时钟

TAUOPEN: 控制变频器控制功能输入时钟

TAU0EN: 控制定时器阵列单元输入时钟

(4) 停止高速系统时钟时设置过程示例

在下列两种情况下，高速系统时钟停止（如果使用外部时钟，禁止时钟输入）。

- 执行 STOP 指令
- 设置 MSTOP 为 1

(a) 执行 STOP 指令

<1> 设置停止外围硬件

停止那些不能在 STOP 模式下使用的外部硬件（如需了解不能在 STOP 模式下使用的外部硬件，可参见第 16 章 待机功能）。

<2> 释放 STOP 模式后 X1 时钟振荡稳定时间的设置

如果在进入 STOP 模式前，X1 时钟开始振荡，则在执行 STOP 指令前设置 OSTS 寄存器的值。

<3> 执行 STOP 指令

执行 STOP 指令时，系统处于 STOP 模式下并且 X1 振荡停止（禁止外部时钟输入）。

(b) 通过将 **MSTOP** 设置为 1 来停止 **X1** 振荡（禁止外部时钟输入）

<1> 确定 CPU 时钟状态（CKC 寄存器）

根据 **CLS** 和 **MCS** 确认 CPU 没有使用高速系统时钟。

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

<R>

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟或 40 MHz 内部高速振荡时钟
0	1	高速系统时钟
1	×	副系统时钟

<2> **X1** 时钟振荡重启后设置 **X1** 时钟振荡的稳定时间^注

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

<3> 停止高速系统时钟（**CSC** 寄存器）

MSTOP 设置为 1 时，**X1** 振荡停止（外部时钟的输入禁止）。

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

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

5.6.2 控制内部高速振荡时钟的示例

下面描述的示例是以下情况时的设置过程示例。

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

(1) 内部高速振荡时钟重启振荡时设置过程示例^注

<1> 内部高速振荡时钟重新振荡的设置（**CSC** 寄存器）

当 **HIOSTOP** 清为 0 时，内部高速振荡时钟重新振荡。

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

(2) 当内部高速振荡时钟作为 CPU/外围硬件时钟时设置过程示例

<1> 内部高速振荡时钟的重启振荡^注

（参见 5.6.2 (1) 当重启内部高速振荡时钟时设置过程示例）。

注 当内部高速振荡时钟运行时，不需要设置<1>。

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

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

注意事项 如果在重启内部高速振荡时钟后，将 CPU/外围硬件时钟从高速系统时钟切换到内部高速振荡时钟，要经过至少 10 μs。

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

(3) 停止内部高速振荡时钟时设置过程示例

可以用如下两种方式停止内部高速振荡时钟。

- 执行 STOP 指令
- 设置 HIOSTOP 为 1

(a) 执行 STOP 指令

<1> 外围硬件的设置

停止那些不能在 STOP 模式下使用的外部硬件（如需了解不能在 STOP 模式下使用的外部硬件，可参见第 16 章 待机功能）。

<2> 释放 STOP 模式后 X1 时钟振荡稳定时间的设置

如果在进入 STOP 模式前，X1 时钟开始振荡的话，则在执行 STOP 指令前设置 OSTC 寄存器的值。

<3> 执行 STOP 指令

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

(b) 通过设置 HIOSTOP 为 1 停止内部高速振荡时钟

<1> 确定 CPU 时钟状态（CKC 寄存器）

根据 CLS 和 MCS 确认 CPU 没有使用内部高速振荡时钟。

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

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟或 40 MHz 内部高速振荡时钟
0	1	高速系统时钟
1	x	副系统时钟

<2> 停止内部高速振荡时钟（CSC 寄存器）

当 HIOSTOP 设置为 1 时，内部高速振荡时钟停止。

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

<R>

5.6.3 控制副系统时钟示例

通过连接晶体振荡器到 XT1 和 XT2 引脚，副系统时钟可振荡。

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

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

下面描述的示例是以下情况时的设置过程示例。

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

注意事项 当副系统时钟作为 CPU 时钟时，副系统时钟也提供给外围硬件（除实时计数器和看门狗定时器外）。此时，不能保证 A/D 转换器的操作。外围硬件的操作特性，参见描述各种外围硬件的章节及第 26 章 电气特性（目标值）。

(1) 副系统时钟振荡的设置示例

<1>设置 P123/XT1 和 P124/XT2 引脚（CMC 寄存器）

<R>

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
0/1	0/1	0	1	0	0	0	0/1

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

<2> 副系统时钟（CSC 寄存器）振荡控制

如果 XTSTOP 清为 0，XT1 振荡器开始振荡。

<3> 等待副系统时钟振荡稳定

通过软件等待副系统时钟的振荡稳定时间，使用定时器功能。

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

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

(2) 使用副系统时钟作为 CPU 时钟时的设置过程示例

<1> 设置副系统时钟振荡*

(参见 5.6.3 (1) 副系统时钟振荡时设置过程示例。)

注 当副系统时钟已经运行时，不需要设置<1>。

<2> 设置副系统时钟作为 CPU 时钟 (CKC 寄存器) 的源时钟

CSS	CPU/外围硬件时钟 (fCLK) 的选择
1	fSUB/2

注意事项 当副系统时钟作为 CPU 时钟时，副系统时钟也提供给外围硬件（除实时计数器和看门狗定时器外）。此时，不能保证 A/D 转换器的操作。外围硬件的操作特性，参见描述各种外围硬件的章节及第 26 章 电气特性（目标值）。

(3) 停止副系统时钟时的设置过程示例

<1> 确定 CPU 时钟状态 (CKC 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用副系统时钟。

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

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟或 40 MHz 内部高速振荡时钟
0	1	高速系统时钟
1	×	副系统时钟

<2> 停止副系统时钟 (CSC 寄存器)

XTSTOP 设置为 1 时，停止 XT1 振荡。

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

<R>

5.6.4 内部低速振荡时钟的控制示例

内部低速振荡时钟不能用作 CPU 时钟。只能用作看门狗定时器时钟。

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

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

（1）停止内部低速振荡时钟时设置过程示例

内部低速振荡时钟在下面两种情况下可被停止。

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

（2）内部低速振荡时钟重新振荡时设置过程示例

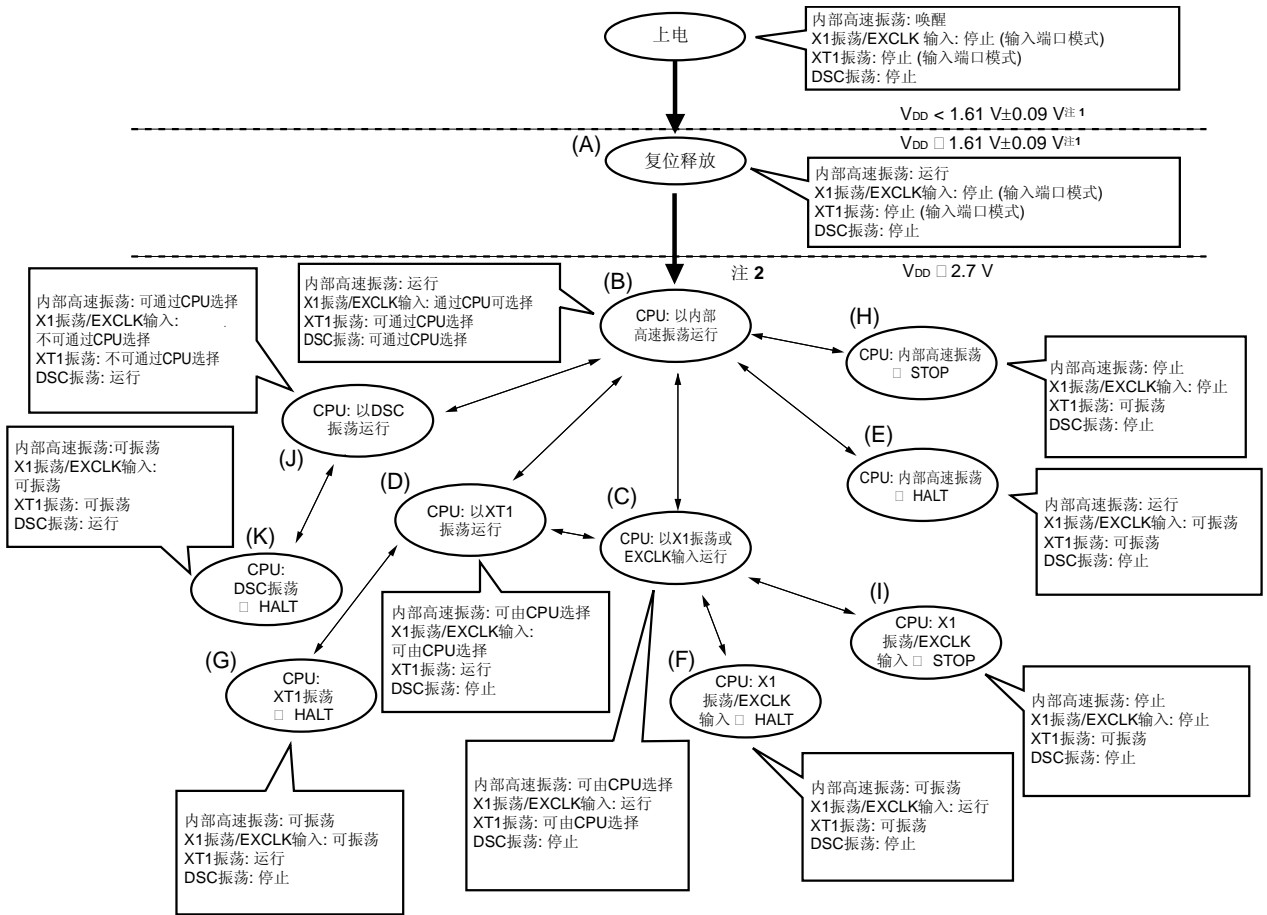
内部低速振荡时钟可重启如下。

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

5.6.5 CPU时钟状态迁移图

图 5-17 显示了该产品 CPU 时钟状态迁移图。

图 5-17. CPU 时钟状态迁移图



注 1. 初步值可能改变。

<R> 2. 在复位释放后，在 4 MHz (8 MHz/2) 上在进入 STOP 模式之前，因为已经通过设定系统时钟控制寄存器 (CKC) 至 09H 来选择 $f_{CLK} = f_{IH}/2$ 。

备注 1. 如果低电压检测电路 (LVI) 通过选项字节默认设置为开启，直到电源电压 (V_{DD}) 超过 $2.07 V \pm 0.2 V^{\text{注}1}$ 复位信号才会释放。
在复位处理后状态变到上图的 B)。

2. DSC: 40 MHz 内部高速振荡时钟

表 5-5 显示了 CPU 时钟的迁移过程与 SFR 寄存器设置示例。

表 5-5. CPU 时钟切换与 SFR 寄存器设置示例（1/6）

（1）复位释放后（A），CPU 使用内部高速振荡时钟（B）

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

（2）复位释放后（A），CPU 使用高速系统时钟（C）

（复位释放后，CPU 立即使用内部高速振荡时钟（B）。）

（SFR 寄存器设置序列）

SFR 寄存器的设置标志 状态迁移	CMC 寄存器 ^{註1}			CSC 寄存器	OSMC 寄存器	OSTC 寄存器	CKC 寄存器
	EXCLK	OSCSEL	AMPH	MSTOP	FSEL		MCM0
（A）→（B）→（C） （X1 时钟：2 MHz ≤ f _x ≤ 10 MHz）	0	1	0	0	0	必须检查	1
（A）→（B）→（C） （X1 时钟：10 MHz < f _x ≤ 20 MHz）	0	1	1	0	1 ^{註2}	必须检查	1
（A）→（B）→（C） （外部主时钟）	1	1	0/1	0	0/1	不能检查	1

注 1. CMC 和 OSMC 寄存器只有在复位后通过 8 位存储器操作指令写入一次。

2. 当 f_{CLK} > 10 MHz，FSEL = 1

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

注意事项 当供电电压达到时钟的可操作电压后，可以设置时钟（参见第 26 章 电气特性（目标值））。

（3）复位释放后（A），CPU 使用副系统时钟（D）

（复位释放后，CPU 立即使用内部高速振荡时钟（B）。）

（SFR 寄存器设置序列）

SFR 寄存器的设置标志 状态迁移	CMC 寄存器 ^註	CSC 寄存器	等待振荡稳定	CKC 寄存器
	OSCSELS	XTSTOP		CSS
（A）→（B）→（D）	1	0	必要	1

注 只有当复位释放后，才能通过 8 位存储器操作指令写入 CMC 寄存器。

备注 表 5-5 中的（A）至（K）对应图 5-17 中的（A）至（K）。

表 5-5. CPU 时钟切换与 SFR 寄存器设置示例 (2/6)

(4) 复位释放后 (A) CPU 在使用 40 MHz 内部高速振荡时钟的操作 (J)

(复位释放后, CPU 立即使用内部高速振荡时钟 (B) 。)

(SFR 寄存器设置序列)

SFR 寄存器的设置标志 状态迁移	DSCCTL 寄存器 DSCON	等待振荡稳定	DSCCTL 寄存器 DSPO	DSCCTL 寄存器 SELDSC
(A) → (B) → (J)	1	必要 (100 μ s)	1	1

(5) CPU 时钟从内部高速振荡时钟 (B) 切换到高速系统时钟 (C)

(SFR 寄存器设置序列)

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

如果这些寄存器已经设置则不必要

如果 CPU 使用高速系统时钟则不必要

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

注意事项 当供电电压达到时钟的可操作电压后, 可以设置时钟 (参见 第 26 章 电气特性 (目标值))。

备注 表 5-5 中的 (A) 至 (K) 对应图 5-17 中的 (A) 至 (K)。

表 5-5. CPU 时钟切换与 SFR 寄存器设置示例（3/6）

(6) CPU 时钟从内部高速振荡时钟（B）切换到副系统时钟（D）

(SFR 寄存器设置序列) →

SFR 寄存器的设置标志	CMC 寄存器*	CSC 寄存器	等待振荡稳定	CKC 寄存器
	OSCELS	XTSTOP		CSS
状态迁移				
(B) → (D)	1	0	必要	1

如果 CPU 工作在副系统时钟则不需要

注 只有当复位释放后，才能通过 8 位存储器操作指令写入 CMC 寄存器。

(7) CPU 时钟从内部高速振荡时钟（B）变换到 40 MHz 内部高速振荡时钟（J）

(SFR 寄存器设置序列) →

SFR 寄存器的设置标志	DSCCTL 寄存器*	等待振荡稳定	DSCCTL 寄存器	DSCCTL 寄存器
	DSCON		DSPO	SELDSC
状态迁移				
(B) → (J)	1	必要 (100 μs)	1	1

如果 CPU 使用 40 MHz 内部高速振荡时钟运行则不必要

(8) CPU 时钟从高速系统时钟（C）变换到内部高速振荡时钟（B）

(SFR 寄存器设置序列) →

SFR 寄存器的设置标志	CSC 寄存器	振荡精确稳定时间	CKC 寄存器
	HIOSTOP		MCM0
状态迁移			
(C) → (B)	0	10 ms	0

如果 CPU 使用内部高速振荡时钟则不必要

备注 表 5-5 中的（A）至（K）对应图 5-17 中的（A）至（K）。

表 5-5. CPU 时钟切换与 SFR 寄存器设置示例（4/6）

(9) CPU 时钟从高速系统时钟（C）变换到副系统时钟（D）

(SFR 寄存器设置序列) →

SFR 寄存器的设置标志 状态迁移	CMC 寄存器 ^註	CSC 寄存器	等待振荡稳定	CKC 寄存器
	OSCELS	XTSTOP		CSS
(C) → (D)	1	0	必要	1

如果 CPU 工作
在副系统时钟则不必要

注 只有当复位释放后，才能通过 8 位存储器操作指令写入 CMC 寄存器。

(10) CPU 时钟从副系统时钟（D）切换到内部高速振荡时钟（B）

(SFR 寄存器设置序列) →

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

如果 CPU 使用内部高速
振荡时钟则不必要

如果这些寄存器已经设
置则不必要

备注 表 5-5 中的（A）至（K）对应图 5-17 中的（A）至（K）。

表 5-5. CPU 时钟切换与 SFR 寄存器设置示例 (5/6)

(11) CPU 时钟从副系统时钟 (D) 切换到高速系统时钟 (C)

(SFR 寄存器设置序列)

SFR 寄存器的设置标志 状态迁移	CMC 寄存器 ^{注1}			OSTS 寄存器	CSC 寄存器	OSMC 寄存器	OSTC 寄存器	CKC 寄存器	
	EXCLK	OSCESEL	AMPH		MSTOP	FSEL		MCM0	CSS
(D) → (C) (X1 时钟: 2 MHz ≤ f _x ≤ 10 MHz)	0	1	0	注 2	0	0	必须 检查	1	0
(D) → (C) (X1 时钟: 10 MHz < f _x ≤ 20 MHz)	0	1	1	注 2	0	1 ^{注3}	必须 检查	1	0
(D) → (C) (外部主时 钟)	1	1	0/1	注 2	0	0/1	不能 检查	1	0

如果这些寄存器已经设置则
不必要

如果 CPU 使用高速系统时钟则不必要

如果这些寄存器
已经设置则不必
要

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

注意事项 当供电电压达到时钟的可操作电压后，可以设置时钟（参见 第 26 章 电气特性（目标值））。

(12) CPU 时钟从 40 MHz 内部高速振荡时钟 (J) 变换到内部高速振荡时钟 (B)

(SFR 寄存器设置序列)

SFR 寄存器的设置标志 状态迁移	CSC 寄存器	CKC 寄存器	DSCCTL 寄存器		
	HIOSTOP	MCM0	SELDSC	DSPO	DSCON
(J) → (B)	0	0	0	0	0

如果CPU使用内
部高速振荡时钟
则不必要如果这些寄存器
已经设置则不必
要

备注 表 5-5 中的 (A) 至 (K) 对应图 5-17 中的 (A) 至 (K)。

表 5-5. CPU 时钟切换与 SFR 寄存器设置示例 (6/6)

- (13) • CPU 使用内部高速振荡时钟 (B) 时, HALT 模式 (E) 的设置。
- CPU 使用高速系统时钟 (C) 时, HALT 模式 (F) 的设置。
 - CPU 使用副系统时钟 (D) 时, HALT 模式 (G) 的设置。
 - 当 CPU 运行在 40 MHz 内部高速振荡时钟时 (J), HALT 模式 (K) 的设置。

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

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

(设置顺序) →

状态迁移		设置		
(B) → (H)	X1 停止	停止那些不能在 STOP 模式下使用的外部功能	—	执行 STOP 指令
	X1 振荡		设置 OSTS 寄存器	
(C) → (I)				

备注 表 5-5 中的 (A) 至 (K) 对应图 5-17 中的 (A) 至 (K)。

5.6.6 CPU时钟改变之前的状况与改变之后的处理

CPU 时钟改变之前的状况与改变之后的处理显示如下。

表 5-6. 改变 CPU 时钟 (1/2)

CPU 时钟		改变前的状况	改变后的处理
改变前	改变后		
内部高速振荡时钟	X1 时钟	X1 振荡稳定 • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • 经历了振荡稳定时间	可以停止内部高速振荡器 (HIOSTOP = 1)。
	外部主系统时钟	允许来自 EXCLK 引脚的外部时钟输入 • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	副系统时钟	XT1 振荡稳定 • OSCSELS = 1, XTSTOP = 0 • 经历了振荡稳定时间	
	40 MHz 内部高速振荡时钟	DSC 振荡稳定 • 设置 DSCON = 1 后, 经历了振荡稳定时间 (100 μ s) • DSPO = 1, SELDSC = 1	
X1 时钟	内部高速振荡时钟	内部高速振荡器振荡 • RSTOP = 0	X1 振荡停止 (MSTOP = 1)。
	外部主系统时钟	切换不允许 (要切换时钟, 执行复位后再设置它。)	—
	副系统时钟	XT1 振荡稳定 • OSCSELS = 1, XTSTOP = 0 • 经历了振荡稳定时间	X1 振荡停止 (MSTOP = 1)。
	40 MHz 内部高速振荡时钟	不能进行传送, 除非时钟转换为内部高速振荡时钟。	—
外部主系统时钟	内部高速振荡时钟	内部高速振荡器振荡 • RSTOP = 0	外部主系统时钟输入禁止 (MSTOP = 1)。
	X1 时钟	切换不允许 (要切换时钟, 执行复位后再设置它。)	—
	副系统时钟	XT1 振荡稳定 • OSCSELS = 1, XTSTOP = 0 • 经历了振荡稳定时间	外部主系统时钟输入禁止 (MSTOP = 1)。
	40 MHz 内部高速振荡时钟	不能进行传送 除非时钟转换为内部高速振荡时钟。	—

表 5-6. 改变 CPU 时钟 (2/2)

CPU 时钟		改变前的状况	改变后的处理
改变前	改变后		
副系统时钟	内部高速振荡时钟	内部高速振荡器振荡且选择内部高速振荡时钟作为主系统时钟 • HIOSTOP = 0, MCS = 0	XT1 振荡停止 (XTSTOP = 1)
	X1 时钟	X1 振荡稳定且选择高速系统时钟作为主系统时钟 • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • 经历了振荡稳定时间 • MCS = 1	
	外部主系统时钟	允许来自 EXCLK 引脚的外部时钟输入 并选择高速系统时钟作为主系统时钟 • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	40 MHz 内部高速振荡时钟	不能进行传送, 除非时钟转换为内部高速振荡时钟。	—
40 MHz 内部高速振荡时钟	内部高速振荡时钟	内部高速振荡器振荡且选择内部高速振荡时钟作为主系统时钟 • HIOSTOP = 0, MCS = 0 • SELDSC = 0, DSPO = 0	可停止 40 MHz 内部高速振荡时钟 (DSCON=0)
	X1 时钟	不能进行传送 除非时钟转换为内部高速振荡时钟。	—
	外部主系统时钟	不能进行传送 除非时钟转换为内部高速振荡时钟。	—
	副系统时钟	不能进行传送 除非时钟转换为内部高速振荡时钟。	—

5.6.7 CPU时钟和主系统时钟切换所需的时间

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

修改 CKC 后，实际的切换操作不会立即执行；操作在予切换时钟上将继续几个时钟（见表 5-7~表 5-10）。

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

当切换 CPU 时钟时，外围硬件时钟也被切换。

<R>

表 5-7. 主系统时钟切换需要的最长时间

时钟 A	切换方法	时钟 B	备注
f_{MAIN}	\longleftrightarrow (改变分频比)	f_{MAIN}	请参阅表 5-8
f_{IH}	\longleftrightarrow	f_{MX}	请参阅表 5-9
f_{MAIN}	\longleftrightarrow	f_{SUB}	请参阅表 5-10

<R>

表 5-8. $f_{MAIN} \leftrightarrow f_{MAIN}$ 所需的最大时钟数（改变分频比）

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

<R>

表 5-9. $f_{IH} \leftrightarrow f_{MX}$ 所需的最大时钟数

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

（注意事项 1 和 2 列于下页中）

<R>

表 5-10. f_{MAIN} ↔ f_{SUB} 所需的最大时钟数

切换前的设置值		切换后的设置值	
CSS		CSS	
		0 (f _{CLK} = f _{MAIN})	1 (f _{CLK} = f _{SUB})
0 (f _{CLK} = f _{MAIN})	f _{MAIN} < f _{SUB}		2 + f _{MAIN} /f _{SUB} 时钟
	f _{MAIN} > f _{SUB}		1 + 2f _{MAIN} /f _{SUB} 时钟
1 (f _{CLK} = f _{SUB})	f _{MAIN} < f _{SUB}	1 + 2f _{SUB} /f _{MAIN} 时钟	
	f _{MAIN} > f _{SUB}	2 + f _{SUB} /f _{MAIN} 时钟	

- 备注**
1. 表 5-8～表 5-10 中列出的时钟数是切换前的 CPU 时钟数。
 2. 通过舍去小数部分，计算表 5-8～表 5-10 的时钟数。

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

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

5.6.8 时钟振荡停止前的状况

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

表 5-11. 时钟振荡停止前的状况及标志位设置

时钟	时钟振荡停止前的状况 (禁止外部时钟输入)	SFR 寄存器的标志位 设置
内部高速振荡时钟	MCS = 1 或 CLS = 1 (CPU 使用内部高速振荡时钟以外的时钟运行)	HIOSTOP = 1
X1 时钟 外部主系统时钟	MCS = 0 或 CLS = 1 (CPU 使用高速系统时钟以外的时钟运行)	MSTOP = 1
副系统时钟	CLS = 0 (CPU 使用副系统时钟以外的时钟运行)	XTSTOP = 1
40 MHz 内部高速振荡时钟	SELDSC = 0、DSPO = 0 (主系统时钟工作在 40 MHz 内部高速振荡时钟以外的时钟下)	DSCON = 0

第 6 章 定时器阵列单元TAUS

定时器阵列单元 TAUS 的每一单元有 12 个 16 位定时器。每个 16 位定时器叫一个通道，可以作为一个独立定时器。此外，可采用两个或更多的“通道”来创建高精度定时器。

<R>
<R>
<R>
<R>

<R>

<R>

独立操作功能 (在本章解释)	组合操作功能 (在本章解释)	变频控制功能 (可参考第 7 章 变频控制功能)
<ul style="list-style-type: none">• 间隔定时器• 方波输出• 外部事件计数器• 分频功能• 输入脉冲间隔测• 输入信号高/ 低电平宽度	<ul style="list-style-type: none">• PWM 输出• 单脉冲输出• 多路 PWM 输出	<ul style="list-style-type: none">• 实时输出功能（类型 1）• 实时输出功能（类型 2）• 6 相 PWM 输出功能• 三角波 PWM 输出功能• 带有死区时间的三角波 PWM 输出功能• 带有死区的 6 相三角波 PWM 输出功能操作• 中断信号分选功能• A/D 转换触发输出功能（类型 1）• A/D 转换触发输出功能（类型 2）• 链接实时输出功能（类型 1）• 链接实时输出功能（类型 2）• 链接实时输出功能（类型 3）• 非互补调制输出功能（类型 1）• 非互补调制输出功能（类型 2）• 互补调制输出功能

通道 7 与串行阵列单元的 UART0 可以组合实现 LIN 总线接收处理。

6.1 定时器阵列单元的功能TAUS

定时器阵列单元 TAUS 有如下功能。

6.1.1 独立操作时每个通道的功能

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

(1) 间隔定时器

每个单元的每个定时器可用作产生固定间隔中断（INTTMn）的参考定时器。

(2) 方波输出

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

(3) 外部事件计数器

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

(4) 分频功能

从定时器输入引脚 (TIm) 输入的时钟被分频, 并且从输出引脚 (TOm) 输出。

(5) 输入脉冲间隔测量

通过脉冲信号有效沿输入到定时器输入引脚 (Tin、SLTI) 时来开始计数。在下个脉冲有效沿捕捉定时器的计数值。这样, 可以测量输入脉冲的间隔。

(6) 输入信号高/低电平宽度测量

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

备注 1. n = 00~11

定时器输入引脚情况下 (TIn): n = 02-07、09-11

定时器输出引脚情况下 (TOn): n = 02-07、10、11

2. m = 10、11

6.1.2 和其它通道组合操作时每个通道的功能

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

(1) PWM (脉冲宽度调制器) 输出

两个通道用作设置产生指定周期和指定占空比的脉冲。

(2) 单脉冲输出

两个通道用作设置产生指定延迟时间和指定脉冲宽度的单脉冲。

(3) 多路 PWM (脉冲宽度调制器) 输出

通过扩展 PWM 功能, 使用一个主通道和两个以上从通道, 可以产生九种类型指定周期和额指定占空比的 PWM 信号。

6.1.3 LIN总线支持功能（仅限通道 7）

（1）唤醒信号的检测

定时器在输入到 UART0 的串行数据输入引脚（RxD0）信号的下降沿开始计数，在上升沿捕捉定时器计数值。这样，可以测量低电平宽度。如果低电平宽度大于指定值，则认为是一个唤醒信号。

（2）同步中断域检测

在检测到唤醒信号后，定时器在输入到 UART0 的串行数据输入引脚（RxD0）的信号的下降沿开始计数，在上升沿捕捉定时器计数值。这样，可以测量低电平宽度。如果低电平宽度大于指定值，则认为是同步中断域。

（3）同步域的脉冲宽度测量

检测到同步中断区域后，可以测量到输入至 UART0 的串行数据输入引脚（RxD0）信号的低电平宽度和高电平宽度。这样通过测量到的同步域的位间隔，可计算出波特率。

6.2 定时器阵列单元TAUS的配置

定时器阵列单元 TAUS 包含以下硬件。

表 6-1. 定时器阵列单元 TAUS 的配置

项目	配置
定时器/计数器	定时器计数寄存器 n (TCRn)
寄存器	定时器数据寄存器 n (TDRn)
定时器输入	TI02 ~ TI07、TI09、TI10、TI11、SLTI 引脚、RxD0 引脚（对于 LIN 总线）
定时器输出	TO02~TO07、TO10、TO11、SLTO 引脚，输出控制器
控制寄存器	<p><单元设置块的寄存器></p> <ul style="list-style-type: none"> • 外围设备使能寄存器 2 (PER2) • 定时器时钟选择寄存器 0 (TPS0) • 定时器通道使能状态寄存器 0 (TE0) • 定时器通道起始寄存器 0 (TS0) • 定时器通道停止寄存器 0 (TT0) • 定时器输入选择寄存器 0 (TIS0) • 定时器输出使能寄存器 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) • 定时器三角波输出模式寄存器 0 (TOT0) ^注 • 定时器死区时间输出启用寄存器 0 (TDE0) ^注 • 定时器实时输出寄存器 0 (TRO0) ^注 • 定时器实时输出使能寄存器 0 (TRE0) ^注 • 定时器实时控制寄存器 0 (TRC0) ^注 • 定时器调制输出使能寄存器 0 (TME0) ^注 • TAU 选择模式寄存器 (OPMR) ^注 • TAU 选择状态寄存器 (OPSR) ^注 • TAU 可选高阻起始触发寄存器 (OPHS) ^注 • TAU 可选高阻停止触发寄存器 (OPHT) ^注 <p><每个通道的寄存器></p> <ul style="list-style-type: none"> • 定时器模式寄存器 n (TMRn) • 定时器状态寄存器 n (TSRn) • 输入切换控制寄存器 (ISC) • 噪声滤波使能寄存器 1、2 (NFEN1、NFEN2) • 端口模式寄存器 1、3、5、7 (PM1、PM3、PM5、PM7) • 端口寄存器 1、3、5、7 (P1、P3、P5、P7)

注 这些寄存器用于变频控制功能。要了解详细情况，请参考第 7 章 变频控制功能。

备注 n: 通道编号 (n = 00~11)

• 定时器 I/O 引脚配置

通过设置输入切换控制寄存器（ISC），P52/SLTI/SLTO 引脚可被分配为通道 0、1 和 8 至 11 的定时器 I/O。（输入切换控制寄存器（ISC）的详细信息，请参阅 6.3（23）输入切换控制寄存器（ISC）。）

可为通道 0、1 和 8 至 11 选择下列 I/O 引脚。

表 6-2. 可为通道 0、1 和 8 至 11 选择的 I/O 引脚

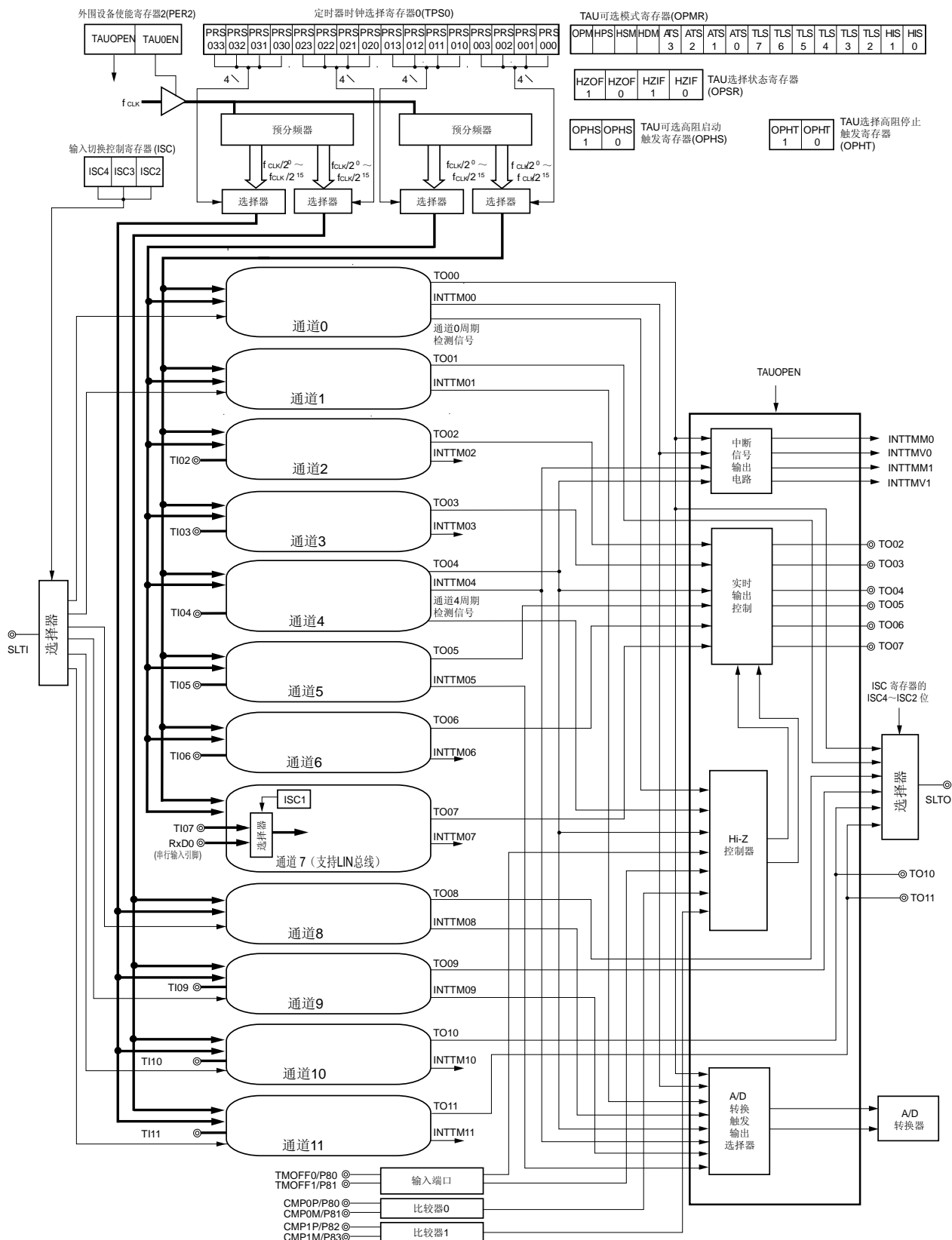
选择 I/O 引脚的通道	输入引脚	输出引脚
通道 0	P52/SLTI/SLTO 引脚	P52/SLTI/SLTO 引脚
通道 1	P52/SLTI/SLTO 引脚	P52/SLTI/SLTO 引脚
通道 8	P52/SLTI/SLTO 引脚	P52/SLTI/SLTO 引脚
通道 9	• P31/TI09 引脚 • P52/SLTI/SLTO 引脚	P52/SLTI/SLTO 引脚
通道 10	• P74/TI10 引脚 • P52/SLTI/SLTO 引脚	• P73/TO10 引脚 • P52/SLTI/SLTO 引脚
通道 11	• P75/TI11 引脚 • P52/SLTI/SLTO 引脚	• P30/TO11 引脚 • P52/SLTI/SLTO 引脚

注意事项 此后，计时器 I/O 引脚被描述成 TIn 和 TOn（n = xx），也包括 SLTI 和 SLTO 引脚的选择。

- 备注**
1. 当定时器输入和定时器输出被同一个引脚分享，则只能使用定时器输入或定时器输出中的一个功能。
 2. 对 P52/SLTI/SLTO 引脚，只有上述的一个通道可以被分配为定时器 I/O 引脚。
 3. 对上述通道以外的通道（通道 2 至 7），SLTI 和 SLTO 引脚不能被选为定时器 I/O。

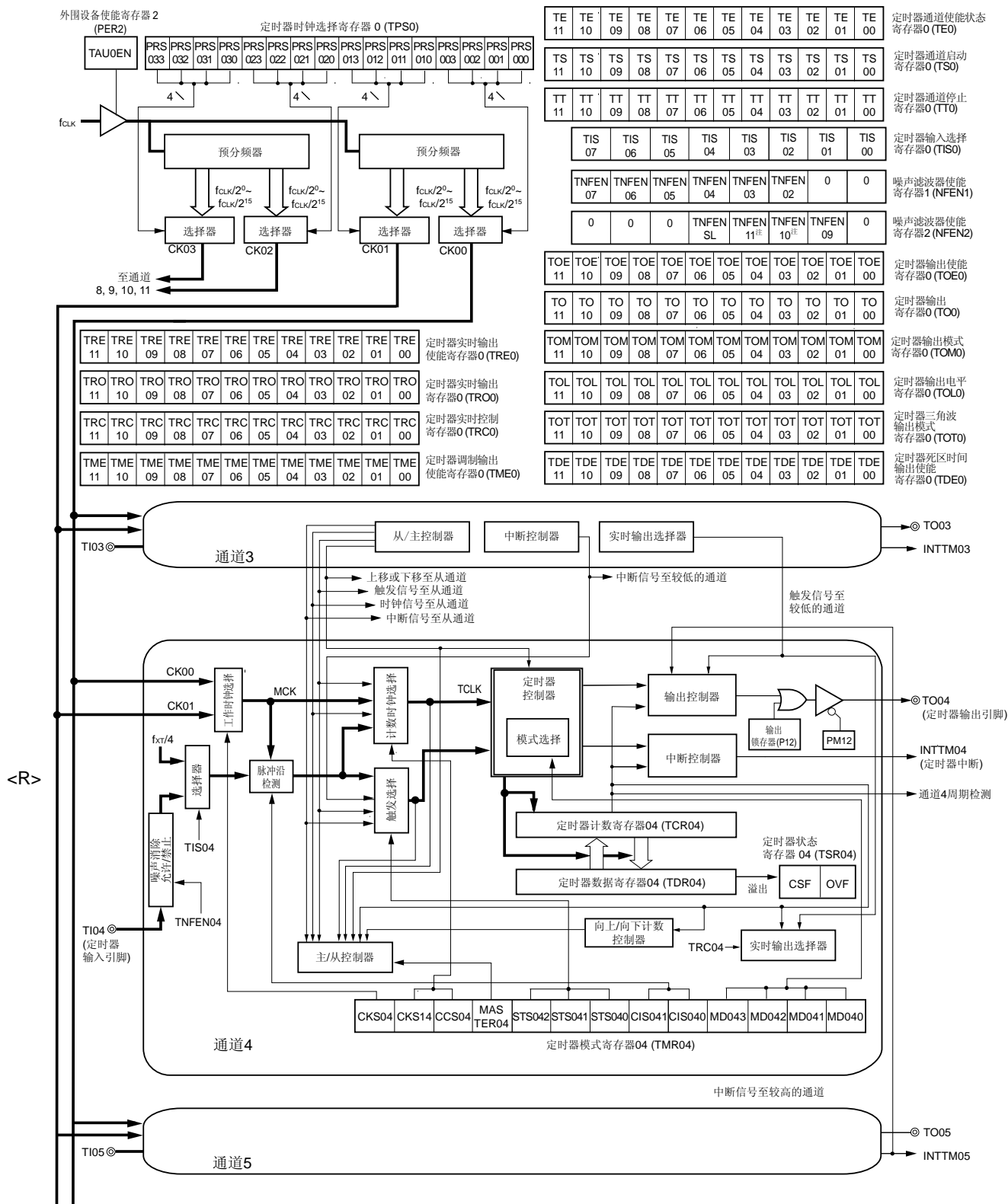
图 6-1 所示为框图。

表 6-1. 定时器阵列单元 TAUS 的整体配置



备注 图 6-1 中的配置图也包括用于变频控制功能的寄存器和引脚。变频控制功能的详细情况, 请参考第 7 章变频控制功能。

图 6-2. 每个通道的框图（例：通道 4）



备注 图 6-2 中的框图也包括用于变频控制功能的寄存器和引脚。变频控制功能的详细情况，请参考第 7 章 变频控制功能。

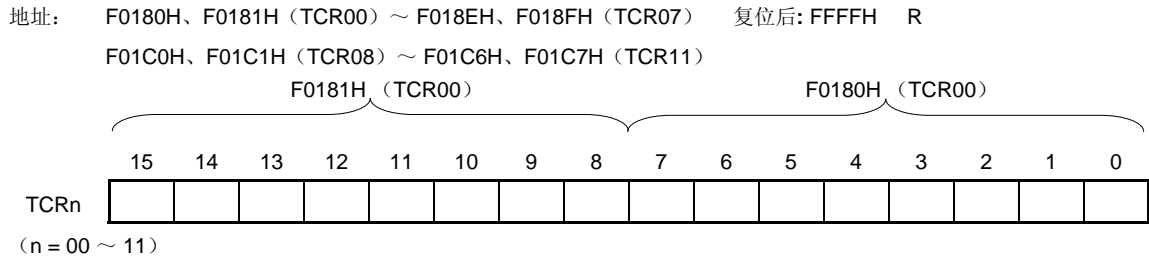
(1) 定时器计数寄存器 n (TCRn)

TCRn 是 16 位只读寄存器，用于时钟计数。

此计数器的值与计数时钟上升沿同步递增或递减。

计数器的方向是递增或递减，由 TMRn 的 MDn4~MDn0 位选择的操作模式决定。

图 6-3. 定时器计数寄存器 n (TCRn) 的格式



可通过读取 TCRn 获得计数值。

如下情况下计数值设置为 FFFFH。

- 复位信号发生时
- 外围设备使能寄存器 2 (PER2) 的 TAU0EN 位清零时

计数值在如下情况中清除为 0000H。

- 在捕捉模式下启动触发输入时
- 在捕捉模式下捕捉操作已完成

注意事项 即使读取 TCRn 时，计数值也不会被捕捉到 TDRn。

TCRn 寄存器读取值根据操作模式改变和操作状态而有如下差异。

<R>

表 6-3. TCRn 寄存器在不同操作模式下的读取值

操作模式	计数模式	TCRn 寄存器的读取值 ^{※1}			
		复位后操作模式改变	计数操作暂停后操作模式改变 (TTn = 1)	计数操作暂停后重启 (TTn = 1)	一次计数后启动触发等待状态期间
间隔定时器模式	递减	FFFFH	未定义	操作终止时的值	—
捕捉模式	递增	0000H	未定义	操作终止时的值	—
事件计数器模式	递减	FFFFH	未定义	操作终止时的值	—
一次计数模式	递减	FFFFH	未定义	操作终止时的值	FFFFH
捕捉&一次计数模式	递增	0000H	未定义	操作终止时的值	TDRn 寄存器的捕捉值 + 1
上下计数模式 ^{※2}	递减和递增计数	FFFFH	未定义	操作终止时的值	—

- 注
1. 显示当 TEn = 0 且 TSn 被设置为"1"时 TCRn 寄存器的读取值。该读取值保存在 TCRn 寄存器直到计数操作开始。
 2. 这些运行模式用于变频控制功能。要了解详细情况，请参考第 7 章 变频控制功能。

备注 n = 00~11

(2) 定时器数据寄存器 n (TDRn)

此寄存器为 16 位寄存器，可选择捕捉功能和比较功能。

通过使用 TMRn 的 MDn4 至 MDn0 位选择操作模式可切换捕捉功能或比较功能。

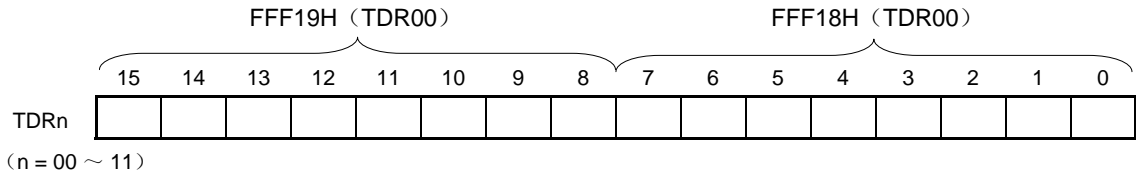
TDR_n 的值在任何时间都可改变。

此寄存器可以以 16 位单元读或写。

复位后这个寄存器的值为 0000H。

图 6-4. 定时器数据寄存器 n (TDRn) 的格式

地址: FFF18H、FFF19H (TDR00)、FFF1AH、FFF1BH (TDR01)、复位后: 0000H R/W
FFF64H、FFF65H (TDR02) ~ FFF76H、FFF77H (TDR11)



(i) **TDRn** 用作比较寄存器时

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

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

(ii) **TDRn 用作捕捉寄存器时**

当捕捉触发输入时，TCRn 的计数值被捕捉到 TDRn。

TIn 引脚的有效沿可选择作为捕捉触发。此选择由 TMRn 进行。

备注 n = 00~11

6.3 控制定时器阵列单元TAUS的寄存器

定时器阵列单元 TAUS 被以下寄存器控制。

- 外围设备使能寄存器 2(PER2)
- 定时器时钟选择寄存器 0(TPS0)
- 定时器模式寄存器 n(TMRn)
- 定时器状态寄存器 n(TSRn)
- 定时器通道使能状态寄存器 0(TE0)
- 定时器通道启动寄存器 0(TS0)
- 定时器通道停止寄存器 0(TT0)
- 定时输入选择寄存器 0(TIS0)
- 定时器输出使能寄存器 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)
- 定时器三角波输出模式寄存器 0 (TOT0)^注
- 定时器死区时间输出使能寄存器 0 (TDE0)^注
- 定时器实时输出寄存器 0 (TRO0)^注
- 定时器实时输出使能寄存器 0 (TRE0)^注
- 定时器实时控制寄存器 0 (TRC0)^注
- 定时器调制输出使能寄存器 0 (TME0)^注
- TAU 选择模式寄存器 (OPMR)^注
- TAU 选择状态寄存器 (OPSR)^注
- TAU 可选高阻启动触发寄存器 (OPHS)^注
- TAU 可选高阻停止触发寄存器 (OPHT)^注
- 输入切换控制寄存器 (ISC)
- 噪声滤波器使能寄存器 1、2(NFEN1、NFEN2)
- 端口模式寄存器 1、3、5、7(PM1、PM3、PM5、PM7)
- 端口寄存器 1、3、5、7 (P1、P3、P5、P7)

注 这些寄存器用于变频控制功能。要了解详细情况，请参考第7章 变频控制功能。

备注 n = 00 ~ 11

(1) 外围设备使能寄存器 2(PER2)

PER2 用于使能或禁止每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少消耗和噪音。

使用定时器阵列单元 TAUS 时，确保设置此寄存器的第 0 位 (TAU0EN) 为 1。

当使用变频控制功能时，确认设置第一位(TAUOPEN) 为 1。(变频控制功能的详细情况，请参考第 7 章 变频控制功能。)

PER2 可以通过 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

- 注意事项 1. 设置定时器阵列单元 TAUS 时，首先确保设置 TAU0EN 为 1。如果 TAU0EN = 0，定时器阵列单元 TAUS 控制寄存器的写入将被忽略，并且所有读取值都是默认值。同样，当使用变频功能时，先设置 TAUOPEN 为 1。
2. 确保清除 PER2 寄存器的第 2 至 7 位为 0。

图 6-5. 外围设备使能寄存器 2(PER2)的格式

地址: F00F2H 复位后: 00H R/W

符号	7	6	5	4	3	2	<1>	<0>
PER2	0	0	0	0	0	0	TAUOPEN	TAU0EN

TAU0EN	定时器阵列单元 TAUS 输入时钟的控制
0	停止输入时钟的供给 <ul style="list-style-type: none">由定时器阵列单元 TAUS 使用的 SFR 不能写入。定时器阵列单元 TAUS 处于复位状态。
1	供给输入时钟 <ul style="list-style-type: none">由定时器阵列单元 TAUS 使用的 SFR 可读/写。

TAUOPEN	变频控制模块输入时钟的控制
0	停止输入时钟的供给 <ul style="list-style-type: none">由变频控制模块使用的 SFR 不能写入。变频控制模块在复位状态。
1	供给输入时钟 <ul style="list-style-type: none">由变频控制模块使用的 SFR 可读/写。

(2) 定时器时钟选择寄存器 0(TPS0)

TPS0 是 16 位寄存器，用于选择四种类型的工作时钟 (CK00、CK01、CK02、CK03)，这些时钟可以共同提供给每个通道。

可通过每个位设置的工作时钟如下。

PRS000 ~ PRS003 : CK00 (可设置为定时器通道 0 至 7)
PRS010 ~ PRS013 : CK01 (可设置为定时器通道 0 至 7)
PRS020 ~ PRS023 : CK02 (可设置为定时器通道 8 至 11)
PRS030 ~ PRS033 : CK03 (可设置为定时器通道 8 至 11)

定时器操作期间只有在如下情况下可以重写 TPS0。

PRS000 ~ PRS003 位: 当运行停止状态下所有通道都设为 $CKSn = 0$ ($TEn = 0$) ($n = 00 \sim 07$)
PRS010 ~ PRS013 位: 当运行停止状态下所有通道都设为 $CKSn = 1$ ($TEn = 0$) ($n = 00 \sim 07$)
PRS020 ~ PRS023 位: 当所有设为 $CKSn = 0$ 的通道都运行在停止状态下 ($TEn = 0$) ($n = 08 \sim 11$)
PRS030 ~ PRS033 位: 当所有设为 $CKSn = 1$ 的通道都运行在停止状态下 ($TEn = 0$) ($n = 08 \sim 11$)

TPS0 可由 16 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 6-6. 定时器时钟选择寄存器 0 (TPS0)的格式

地址: F01B6H, F01B7H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0	PRS 033	PRS 032	PRS 031	PRS 030	PRS 023	PRS 022	PRS 021	PRS 020	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 0m3	PRS 0m2	PRS 0m1	PRS 0m0		工作时钟(CK0m) ^注 的选择			
					f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz	
0	0	0	0	f _{CLK}	5 MHz	10 MHz	20 MHz	40 MHz
0	0	0	1	f _{CLK} /2	2.5 MHz	5 MHz	10 MHz	20 MHz
0	0	1	0	f _{CLK} /2 ²	1.25 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	1	f _{CLK} /2 ³	625 kHz	1.25 MHz	2.5 MHz	5 MHz
0	1	0	0	f _{CLK} /2 ⁴	312.5 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	1	f _{CLK} /2 ⁵	156.2 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	1	0	f _{CLK} /2 ⁶	78.1 kHz	156.2 kHz	312.5 kHz	625 kHz
0	1	1	1	f _{CLK} /2 ⁷	39.1 kHz	78.1 kHz	156.2 kHz	312.5 kHz
1	0	0	0	f _{CLK} /2 ⁸	19.5 kHz	39.1 kHz	78.1 kHz	156.2 kHz
1	0	0	1	f _{CLK} /2 ⁹	9.76 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	1	0	f _{CLK} /2 ¹⁰	4.88 kHz	9.76 kHz	19.5 kHz	39.1 kHz
1	0	1	1	f _{CLK} /2 ¹¹	2.44 kHz	4.88 kHz	9.76 kHz	19.5 kHz
1	1	0	0	f _{CLK} /2 ¹²	1.22 kHz	2.44 kHz	4.88 kHz	9.76 kHz
1	1	0	1	f _{CLK} /2 ¹³	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	1	0	f _{CLK} /2 ¹⁴	305 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	1	f _{CLK} /2 ¹⁵	153 Hz	305 Hz	610 Hz	1.22 kHz

注 当改变 f_{CLK} (由系统时钟控制寄存器 (CKC) 的值改变) 时钟时, 停止定时器阵列单元 TAUS (TT0 = 00FFH)。

备注 1. f_{CLK}: CPU/外围硬件时钟频率
2. m = 0~3

(3) 定时器模式寄存器 n(TMRn)

<R>

TMRn 设定通道 n 的工作模式。可用来选择工作时钟(MCK)、计数时钟，无论定时器作为主/从、开始触发器和捕捉除法器、定时器输入的有效沿进行操作，以及运行模式(间隔、捕捉、事件计数器、一次计数、捕捉&一次计数，或向上和向下计数[※])。

当寄存器操作时 (当 TE0 = 1) 禁止重写 TMRn。但是第 7 位和第 6 位 (CISn1、CISn0) 可以重写，即使当寄存器一些功能正在操作时 (当 TE0 = 1) (详细情况，请参考 6.7 定时器阵列单元 TAUS 的独立通道操作和 6.8 定时器阵列单元 TAUS 的多通道操作)。

TMRn 可由 16 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

注 这些模式用于变频控制功能。要了解详细情况，请参考第 7 章 变频控制功能。

图 6-7. 定时器模式寄存器 n (TMRn) 的格式(1/3)

地址: F0190H、F0191H (TMR00) ~ F019EH、F019FH (TMR07) 复位后: 0000H R/W
F01C8H、F01C9H (TMR08) ~ F01CEH、F01CFH (TMR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKS n	0	CCS 1n	CCS 0n	MAST ERn	STS n2	STS n1	STS n0	CIS n1	CIS n0	0	MD n4	MD n3	MD n2	MD n1	MD n0

CKS n	通道 n 的工作时钟 (MCK) 选择	
0	由 PRS 寄存器设置操作时钟 CK00: 定时器通道 0 至 7 由 PRS 寄存器设置操作时钟 CK02: 定时器通道 8 至 11	
1	由 PRS 寄存器设置操作时钟 CK01: 定时器通道 0 至 7 由 PRS 寄存器设置操作时钟 CK03: 定时器通道 8 至 11	
通过沿检测器来使用工作时钟 MCK。根据 CCS1n 和 CCS0n 位的设置，产生一个采样时钟和一个计数时钟(TCLK)。		

<R>

CCS 1n	CCS 0n	通道 n 的计数时钟(TCLK) 选择
0	0	工作时钟 MCK 由 CKSn 位指定
0	1	从 TIn 引脚输入的输入信号的有效沿
1	0	选择主通道计数时钟 (当通道用作组合操作功能中的从通道时) [※] 。
1	1	选择主通道中断信号 (当通道用作组合操作功能中的从通道时) [※] 。
计数时钟(TCLK)用于定时器/计数器、输出控制器和中断控制器。		

注 这些设置用于变频控制功能。要了解详细情况，请参考第 7 章 变频控制功能。

注意事项 确保将位 14 和 5 清“0”。

备注 n = 00~11

图 6-7. 定时器模式寄存器 n (TMRn) 的格式(2/3)

地址: F0190H、F0191H (TMR00) ~ F019EH、F019FH (TMR07) 复位后: 0000H R/W
F01C8H、F01C9H (TMR08) ~ F01CEH、F01CFH (TMR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKSn	0	CCSn1n	CCSn0n	MASTERn	STSn2	STSn1	STSn0	CISn1	CISn0	0	MDn4	MDn3	MDn2	MDn1	MDn0

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

<R>

STSn2	STSn1	STSn0	通道 n 的启动触发或捕捉触发设置
0	0	0	只有软件触发启动有效 (其它触发源不被选择)。
0	0	1	TIn 引脚输入的有效沿用作启动触发和捕捉触发。
0	1	0	TIn 引脚输入的双沿分别作为启动触发和捕捉触发。
1	0	0	使用主通道的中断信号 (当通道用作组合操作功能中的从通道时)。
1	1	0	使用死区时间触发产生通道的触发 [※] 。
1	1	1	使用主通道的增减控制触发 [※] 。
其它值			设置禁止

CISn1	CISn0	TIn 引脚输入有效沿的选择
0	0	下降沿
0	1	上升沿
1	0	双沿 (当低电平宽度被测量时) 启动触发: 下降沿, 捕捉触发: 上升沿
1	1	双沿 (当高电平宽度被测量时) 启动触发: 上升沿, 捕捉触发: 下降沿
如果当 STSn2~STSn0 位的值不是 010B 时指定双沿, 设置 CISn1~CISn0 位为 10B。		

注 这些设置用于变频控制功能。要了解详细情况, 请参考第 7 章 变频控制功能。

备注 n = 00~11

图 6-7. 定时器模式寄存器 n (TMRn) 的格式(3/3)

地址: F0190H、F0191H (TMR00) ~ F019EH、F019FH (TMR07) 复位后: 0000H R/W
F01C8H、F01C9H (TMR08) ~ F01CEH、F01CFH (TMR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRn	CKS n	0	CCS 1n	CCS 0n	MAST ERn	STS n2	STS n1	STS n0	CIS n1	CIS n0	0	MD n4	MD n3	MD n2	MD n1	MD n0

<R>	MD n4	MD n3	MD n2	MD n1	MD n0	通道 n 的操作模式	TCR 的计数操作	独立操作
	0	0	0	0	1/0	间隔定时器模式	递减计数	可能
	0	0	1	0	1/0	捕捉模式	递增计数	可能
	0	0	1	1	0	事件计数器模式	递减计数	可能
	0	1	0	0	1/0	一次计数模式	递减计数	不可能
	0	1	1	0	0	捕捉&一次计数模式	递增计数	可能
	1	0	0	1	0	上下计数模式※1	向上和向下计数	不可能
	其它值					设置禁止		
	MDn0 位的操作根据每种操作模式的不同而有所差异 (参考下表)。							

<R>	操作模式 (由 MDn4~MDn1 位的值设置 (参考上表))	MD n0	开始计数和中断的设置
	• 间隔定时器模式 (0, 0, 0, 0)	0	当计数开始时定时器 中断不产生 (定时器输出也不改变)。
	• 捕捉模式 (0, 0, 1, 0)	1	当计数开始时, 定时器中断产生 (定时器输出也改变)。
	• 事件计数器模式 (0, 0, 1, 1)	0	当计数开始时定时器 中断不产生 (定时器输出也不改变)。
	• 上下计数模式 ^{※1} (1, 0, 0, 1)		
	• 一次计数模式 ^{※2} (0, 1, 0, 0)	0	在计数操作期间启动触发无效。 此期间, 也不产生中断。
		1	在计数操作期间启动触发有效 ^{※3} 。 此期间, 也会产生中断。
	• 捕捉&一次计数模式 (0, 1, 1, 0)	0	当计数开始时定时器中断不产生 (定时器输出也不改变)。 在计数操作期间启动触发无效。 此期间, 中断也不产生。
其它值			设置禁止

注 1. 这些设置用于变频控制功能。要了解详细情况, 请参考第7章 变频控制功能。

- <R>
- 在一次计数模式中, 当启动一个计数操作且未控制 TOn 输出时, 中断输出 (INTTMn)。
 - 如果操作期间产生了启动触发 (TSn = 1), 计数器清零, 中断产生, 重新开始计数。

备注 n = 00~11

(4) 定时器状态寄存器 n(TSRn)

TSRn 指示通道 n 的计数器的溢出状态。

TSRn 只有在捕捉模式(MDn4~MDn1 = 0010B)和捕捉&一次计数模式(MDn4~MDn1 = 0110B)下有效。在其它模式下不进行设置。

此外，CSF 只在上下计数模式下(MDn4~MDn1 = 1001B)^{※1}有效。它不能在任何其它模式下被设置。

对于 OVF 位在每种操作模式下的操作和设置/清零条件参见表 6-4。

TSRn 可由 16 位存储器操作指令读取。

复位后这个寄存器的值为 0000H。

图 6-8. 定时器状态寄存器 n (TSRn) 的格式

地址: F01A0H、F01A1H (TSR00)~F01AEH、F01AFH (TSR07) 复位后: 0000H R
F01D0H、F01D1H (TSR08) ~ F01D6H、F01D7H (TSR11)

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSF ※1、2	OVF

CSF ※1、2	计数时钟的向上计数或向下计数 (仅上下计数模式)
0	表示计数时钟是向上计数。
1	表示计数时钟是向下计数。

OVF	通道 n 的计数器溢出状态 (仅捕捉模式和捕捉&一次计数模式)
0	溢出不发生。
1	溢出发生。
OVF = 1 时，如果当下一个被捕捉到的值没有溢出，此标志清零 (OVF = 0)。	

注 1. 该操作模式或位，被用于变频控制功能。要了解详细情况，请参考第 7 章 变频控制功能。

2. 通道 0，作为最高通道总是固定为“0”，因为 CSF 是基于最高主通道的高/低信号产生的。

<R>

<R>

表 6-4. 在每种操作模式下 OVF 位操作和置位/清零条件

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

注 这些运行模式用于变频控制功能。要了解详细情况，请参考第 7 章 变频控制功能。

备注 OVF 位在计数器溢出以后不立即改变，但是后面捕捉时改变。

(5) 定时器通道使能状态寄存器 0 (TE0)

TE0 用于允许或停止每个通道的定时器操作。

当定时器通道启动寄存器 0 (TS0) 的一位设为 1，则该寄存器的对应位被设为 1。当定时器通道停止寄存器 0 (TT0) 的一位设为 1，则该寄存器的对应位被清零。

TE0 可以通过一个 16 位存储器操作指令来读取。

复位后这个寄存器的值为 0000H。

图 6-9. 定时器通道使能状态寄存器 0 (TE0)的格式

地址: F01B0H, F01B1H 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	TE11	TE10	TE09	TE08	TE07	TE06	TE05	TE04	TE03	TE02	TE01	TE00

TE _n	指示通道 n 的操作允许/停止状态														
0	操作停止。														
1	操作允许。														

备注 n = 00~11

(6) 定时器通道启动寄存器 0(TS0)

TS0 是用于清零定时器计数器 (TCRn) 和启动每个通道计数操作的触发寄存器。

此寄存器的一位 (TSn) 设置为 1 时, 定时器通道使能状态寄存器 0 (TE0) 的相应位 (TEn) 也会设置为 1。TSn 是触发位, 当 TEn = 1 时立即清零。

TS0 可通过一个 16 位存储器操作指令来设置。

复位后这个寄存器的值为 0000H。

图 6-10. 定时器通道启动寄存器 0 (TS0) 的格式

地址: F01B2H, F01B3H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	TS11	TS10	TS09	TS08	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

TSn	通道 n 的操作使能(启动)触发
0	没有触发操作
1	把 TEn 设为 1 且允许计数操作。 TCRn 在计数操作使能状态下启动的计数操作根据每个操作模式的不同而不同(参考表 6-5)。

注意事项 确保将位 15 至 12 清“0”。

备注 1. 当读取 TS0 寄存器时, 读取值始终为 0。

2. n = 00~11

表 6-5. 从计数操作使能状态到 TCRn 计数启动的操作(1/2)

定时器操作模式	设置 TSn = 1 时的操作
<ul style="list-style-type: none"> 间隔定时器模式 	从启动触发检测 (TSn = 1) 到产生计数时钟前, 没有操作执行。 第一个计数时钟加载 TDRn 的值至 TCRn, 后面的计数时钟执行向下计数操作 (参考 6.3 (6) (a) 间隔定时器模式和上下计数模式下的开始时序)。
<ul style="list-style-type: none"> 事件计数器模式 	把 1 写入 TSn 位以把 TDRn 的值加载至 TCRn。 后面的计数时钟执行向下计数操作。 由 TMRn 寄存器的 STSn2~STSn0 位选择外部触发检测不启动计数操作。(参考 6.3 (6) (b) 事件计数器模式下的开始时序)。
<ul style="list-style-type: none"> 捕捉模式 	从启动触发检测直到计数时钟发生没有操作执行。 第一个计数时钟加载 0000H~TCRn, 后面的计数时钟执行向上计数操作 (参考 6.3 (6) (c) 捕捉模式下的开始时序)。

<R>

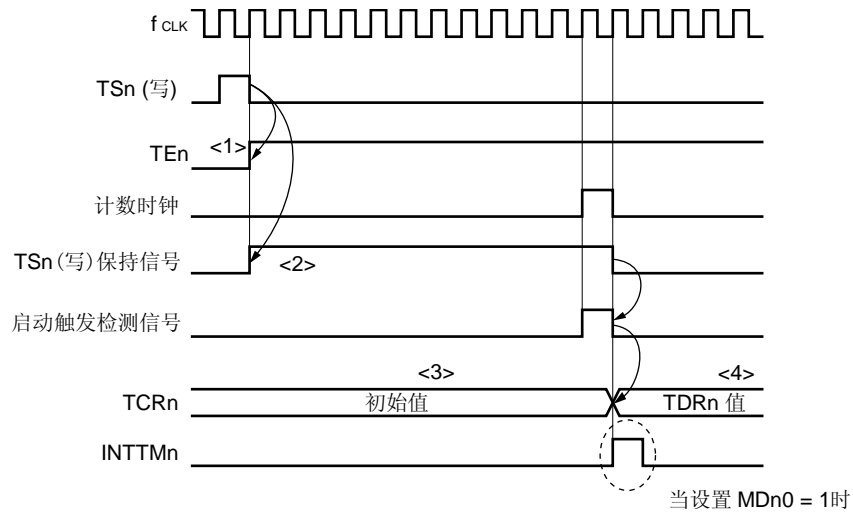
表 6-5. 从计数操作使能状态到 TCRn 计数启动的操作(2/2)

定时器操作模式	设置 TS _n = 1 时的操作
<ul style="list-style-type: none">一次计数模式	当 TE _n = 0 时，把 1 写入 TS _n 位以设置启动触发等待状态。 从启动触发检测直到计数时钟发生没有操作执行。 第一个计数时钟加载 TDR _n 的值至 TCR _n ，后面的计数时钟执行向下计数操作 (参考 6.3 (6) (d) 一次计数模式下的开始时序)。
<ul style="list-style-type: none">捕捉&一次计数模式	当 TE _n = 0 时，把 1 写入 TS _n 位以设置启动触发等待状态。 从启动触发检测直到计数时钟发生没有操作执行。 第一个计数时钟加载 0000H 到 TCR _n ，后续计数时钟执行向上计数操作 (参见 6.3 (6) (e)捕捉&一次计数模式下的开始时序)。
<ul style="list-style-type: none">上下计数模式^注	从启动触发检测 (TS _n = 1) 到产生计数时钟前，没有操作执行。 第一个计数时钟加载 TDR _n 的值至 TCR _n ，后面的计数时钟执行向下计数操作 (参考 6.3 (6) (a) 间隔定时器模式和上下计数模式下的开始时序)。

注 这些运行模式用于变频控制功能。要了解详细情况，请参考第 7 章 变频控制功能。

(a) 间隔定时器模式和上下计数模式下的开始时序[※]

- <1> 通过写入 1 至 TSn，设置 TEn = 1。
- <2> 写入 TSn 的数据会一直保持，直到产生计数时钟。
- <3> TCRn 初始值会一直保持，直到产生计数时钟。
- <4> 计数时钟产生时，将“TDRn 值”加载到 TCRn 并且开始计数。

图 6-11. 开始时序(在间隔定时器模式和上下计数模式下[※])

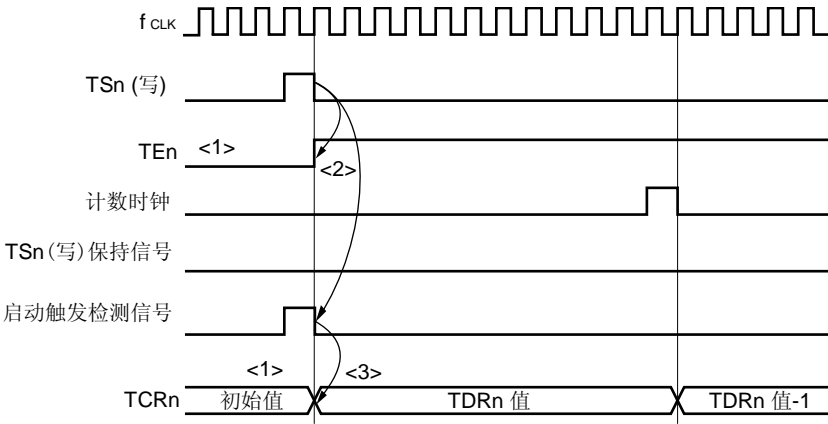
注 这些运行模式用于变频控制功能。要了解详细情况，请参考第 7 章 变频控制功能。

注意事项 由于在写入 TSn 后直到产生计数时钟才会开始计数，所以计数时钟第一个周期操作会产生最大为一个时钟的误差。当需要知道计数开始时间的时候，通过设置 MDn0 = 1 可在计数开始时产生一个中断。

(b) 事件计数器模式下的开始时序

- <1> TEn 设置为 0 时, TCRn 保持初始值。
- <2> 通过写入 1 至 TSn, 设置 TEn 为 1。
- <3> 一旦 1 写入 TSn 以及 TEn 设置为 1, 加载“TDRn 值”到 TCRn 上并开始计数。
- <4> 然后, TCRn 的值根据计数时钟向下计数。

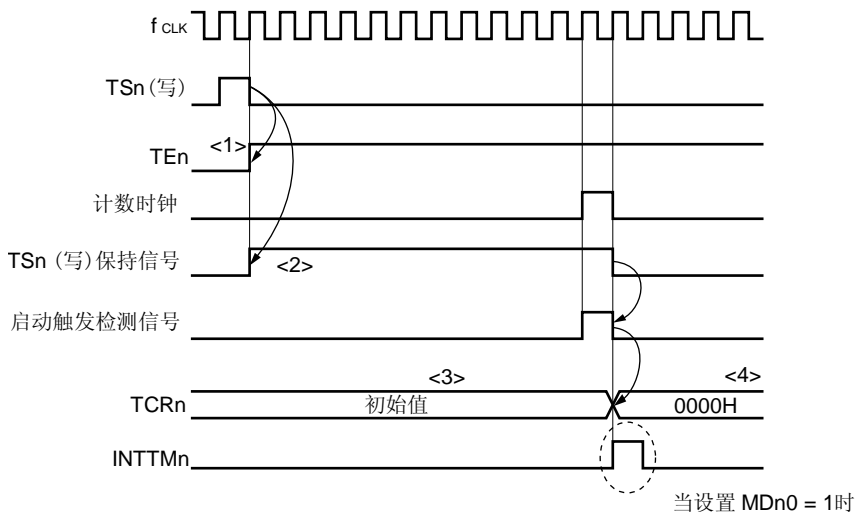
图 6-12. 开始时序(事件计数器模式下)



(c) 捕捉模式下的开始时序

- <1> 通过写入 1 至 TSn, 设置 TEn = 1。
- <2> 写入 TSn 的数据会一直保持, 直到产生计数时钟。
- <3> TCRn 初始值会一直保持, 直到产生计数时钟。
- <4> 计数时钟产生时, 0000H 加载到 TCRn 并且开始计数。

图 6-13. 开始时序(捕捉模式下)

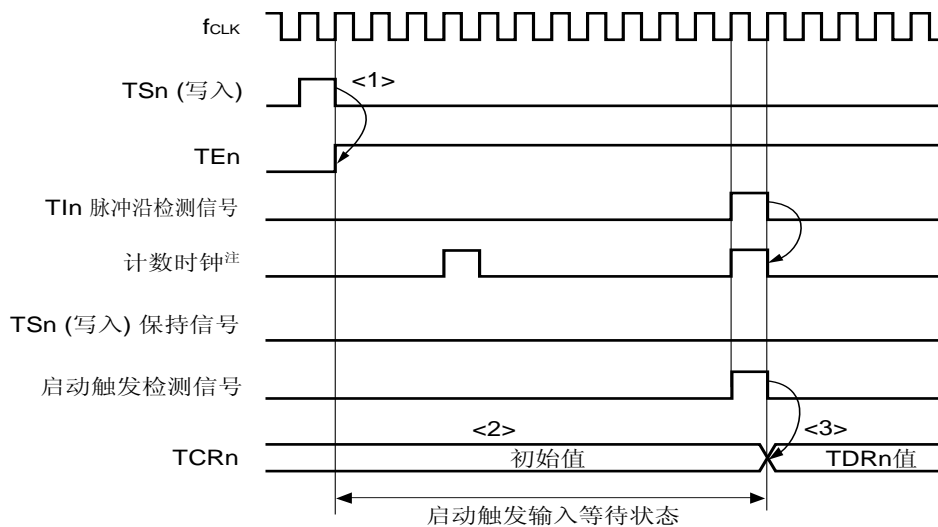


注意事项 由于在写入 TSn 后直到产生计数时钟才会开始计数，所以计数时钟第一个周期操作会产生最大为一个时钟的误差。当需要知道计数开始时间的时候，通过设置 MDn0 = 1 可在计数开始时产生一个中断。

(d) 一次计数模式下的开始时序

- <1> 通过写入 1 至 TSn，设置 TEn = 1。
- <2> 进入启动触发输入等待状态，TCRn 保持初始值。
- <3> 检测到启动触发，“TDRn 值”被加载到 TCRn 并且计数开始。

图 6-14. 开始时序(一次计数模式下)



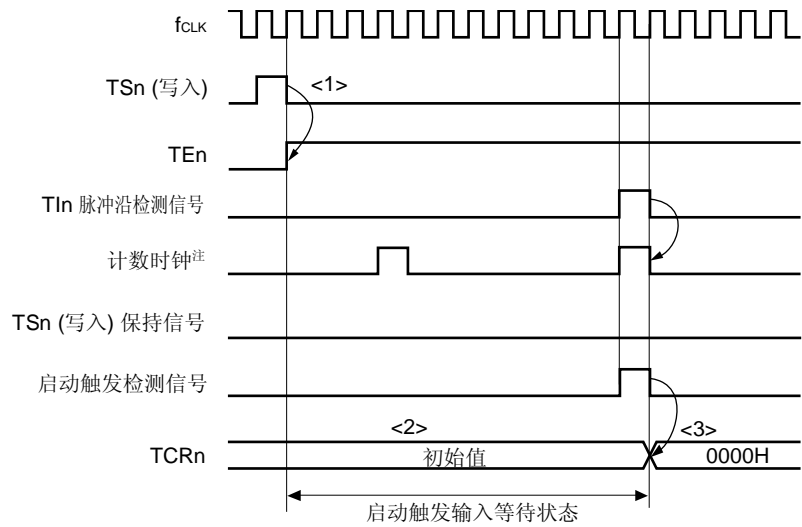
注 当设置一次计数模式时，工作时钟 (MCK) 被选择作为计数时钟 (CCS1n、CCS0n = 0)。

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

<R> (e) 捕捉&一次计数模式下的开始时序

- <1> 通过写入 1 至 TS_n，设置 TE_n = 1。
- <2> 进入启动触发输入等待状态，TCR_n 保持初始值。
- <3> 检测到启动触发，0000H 被加载到 TCR_n 并且计数开始。

图 6-15. 开始时序(在捕捉和一次计数模式下)



注 当设置捕捉&一次计数模式时，工作时钟(MCK)被选作计数时钟(CCS1_n、CCS0_n = 0)。

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

(7) 定时器通道停止寄存器 0(TT0)

TT0 是用于定时器计数器 (TCRn) 清零和启动每个通道计数操作的触发寄存器。
当此寄存器的某位(TTn)设置为 1 时，定时器通道使能状态寄存器 0(TE0)的相应位(Ten)清为 0。TTn 是一个触发器位且在 Ten=0 时，立即清除归 0。
TT0 可由 16 位存储器操作指令设置。
TT0 的低 8 位可以通过 TT0L 由 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 0000H。

图 6-16. 定时器通道停止寄存器 0 (TT0)的格式

地址: F01B4H, F01B5H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	TT11	TT10	TT09	TT08	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

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

注意事项 确保将位 15 至 12 清“0”。

备注 1. 当读取 TT0 寄存器时，读取值始终为 0。
 2. n = 00~11

(8) 定时输入选择寄存器 0(TIS0)

TIS0 用于选择对每个通道定时器输入引脚 (TIn) 的输入信号或被 4 分频的副系统时钟 (fXT/4) 是否有效。
TIS0 可由 1 位或 8 位存储器操作指令设置
复位后这个寄存器的值为 00H。

图 6-17. 定时器输入选择寄存器 0 (TIS0)的格式

地址: FFF3EH 复位后: 00H R/W

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

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

备注 n = 00~07

(9) 定时器输出使能寄存器 0(TOE0)

TOE0 用于允许或禁止每个通道的定时器输出。

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

TOE0 可由 16 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 6-18. 定时器输出使能寄存器 0 (TOE0)的格式

地址: F01BAH, F01BBH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE0	0	0	0	0	TOE 11	TOE 10	TOE 09	TOE 08	TOE 07	TOE 06	TOE 05	TOE 04	TOE 03	TOE 02	TOE 01	TOE 00

TOEn	通道 n 的定时器输出允许/禁止
0	TOn 操作被计数操作 (定时器通道输出位) 停止。 允许写入 TOn 位。 TOn 引脚作为数据输出功能，并且其输出电平设置为 TOn。 TOn 引脚的输出电平可由软件进行操作。
1	TOn 操作通过计数操作 (定时器通道输出位) 被允许。 禁止写入 TOn 位 (写入被忽略)。 TOn 引脚作为定时器输出功能，且 TOEn 根据定时器的运行进行置 1 或清零。 TOn 引脚可以根据定时器的运行输出方波或 PWM。

注意事项 确保将位 15~12 清“0”。

备注 n = 00~11

TO0 是每个通道的定时器输出的一个缓冲寄存器。

此寄存器中每个位的值从每个通道的定时器输出引脚 (TO_n) 输出。

只有当定时器输出禁止 (TO_{En} = 0) 时此寄存器可以通过软件重写。当定时器输出允许时 (TO_{En} = 1)，通过软件重写此寄存器被忽略，只有定时器操作才会改变该值。

为了使用 P10/TO02、P11/TO03、P12/TO04、P13/TO05、P50/TO06、P51/TO07、P73/TO10、P30/TO11 或 P52/SLTO 引脚作为端口功能引脚，需设置相应的 TO_n 位至“0”。

TO0 可由 16 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

地址: F01B8H, F01B9H 复位后: 0000H R/W																
符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	TO 11	TO 10	TO 09	TO 08	TO 07	TO 06	TO 05	TO 04	TO 03	TO 02	TO 01	TO 00

备注 n = 00~11

(11) 定时器输出电平寄存器 0(TOL0)

TOL0 是用于控制每个通道定时器输出电平的寄存器。
当在组合操作模式 (TOMn = 1) 中允许定时器输出 (TOEn = 1) 时，通过此寄存器设置通道 n 反向输出，反映在对定时器输出信号的置位或复位。在反转模式 (TOMn = 0) 下，此寄存器设置无效。
TOL0 可由 16 位存储器操作指令设置。
复位后这个寄存器的值为 0000H。

图 6-20. 定时器输出电平寄存器 0 (TOL0)的格式

地址: F01BCH, F01BDH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	TOL 11	TOL 10	TOL 09	TOL 08	TOL 07	TOL 06	TOL 05	TOL 04	TOL 03	TOL 02	TOL 01	TOL 00

TOL n	通道 n 的定时器输出电平控制
0	正向逻辑输出 (高电平有效) 增加死区时间到正向逻辑端，当死区时间输出使能寄存器 0 (TDE0) 的 TDEn = 1 为 1 ^注 。
1	反向输出 (低电平有效) 增加死区时间到反向逻辑端，当死区时间输出使能寄存器 0 (TDE0) 的 TDEn = 1 为 1 ^注 。

注 这些设置用于变频控制功能。要了解详细情况，请参考第 7 章 变频控制功能。

注意事项 确保将位 15 至 12 清“0”。

- 备注
1. 如果此寄存器的值在定时器操作期间被重写，当定时器输出信号变为下一个，定时器输出反向，在寄存器值重写后立即代替。

2. n = 00~11

(12) 定时器输出模式寄存器 0(TOM0)

TOM0 用于每个通道的定时器输出模式控制。

当通道用于独立操作功能时，设置所使用通道的相应位为 0。

当通道用于组合操作功能时，设置主通道的相应位为 0，从通道的相应位为 1。

通过此寄存器对各通道 n 的设置反映在定时器输出被允许 (TOEn = 1)^注 时定时器输出信号的设置或复位。

当定时器操作停止 (TE0 = 1) 状态时，TOM0 可被写入。

TOM0 可由 16 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

注 使用变频控制功能时每个通道 n 的设置反映在当 TREn=0 且定时器输出允许 (TOEn=1) 时定时器输出信号时序当 TREn 或 TMen = 1 时的设置或复位。

图 6-21. 定时器输出模式寄存器 0 (TOM0)的格式

地址: F01BEH, F01BFH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	TOM 11	TOM 10	TOM 09	TOM 08	TOM 07	TOM 06	TOM 05	TOM 04	TOM 03	TOM 02	TOM 01	TOM 00

TOM n	通道 n 的定时器输出模式控制														
0	反转模式 (由定时器中断请求信号 (INTTMn) 产生反转输出)														
1	通过设置定时器三角波输出模式寄存器 (TOT0) 来选择组合操作模式 ^注 。														

注 当变频控制功能被使用时，通过设置定时器三角波输出模式寄存器(TOT)来选择组合操作模式。要了解详细情况，请参考第 7 章 变频控制功能。

注意事项 确保将位 15 至 12 清“0”。

备注 n: 通道编号, m: 从通道编号

n = 00~11 (主通道时 n = 00、02、04、06、08、10)

n < m ≤ 11(关于主/从通道之间关系的详细信息，请参考 6.6.3 组合操作功能的基本规则的应用范围。)

(13) 定时器三角波输出模式寄存器 0(TOT0)**(14) 定时器实时输出使能寄存器 0(TRE0)****(15) 定时器实时输出寄存器 0(TRO0)****(16) 定时器实时控制寄存器 0(TRC0)****(17) 定时器死区时间输出使能寄存器 0(TDE0)****(18) 定时器调制输出使能寄存器 0(TME0)****(19) TAU 可选模式寄存器(OPMR)****(20) TAU 可选状态寄存器(OPSR)****(21) TAU 可选高阻启动触发寄存器(OPHS)****(22) TAU 可选高阻停止触发寄存器(OPHT)**

上述寄存器用于变频控制功能。关于每个寄存器的变频控制设置，请参考第 7 章 变频控制功能。

(23) 输入切换控制寄存器 (ISC)

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

当 ISC1 位设置为 1 时，串行数据输入引脚 (RxD0) 的输入信号被选择为定时器输入信号。

当选择 P52/SLTI/SLTO 引脚作为定时器通道 0、1 和 8 至 11 的定时器 I/O 引脚，ISC4 至 ISC2 位被设置。

ISC 可以通过一个 1 位或者 8 位存储器操作指令来设置。

复位后这个寄存器的值为 00H。

图 6-22. 输入切换控制寄存器(ISC)的格式

地址: FFF3CH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ISC	0	0	0	ISC4	ISC3	ISC2	ISC1	ISC0

ISC4	ISC3	ISC2	选择 P52/SLTI/SLTO 引脚作为定时器 I/O 引脚											
			通道 0		通道 1		通道 8		通道 9		通道 10		通道 11	
			输入 引脚	输出 引脚	输入 引脚	输出 引脚	输入 引脚	输出 引脚	输入 引脚	输出 引脚	输入 引脚	输出 引脚	输入 引脚	输出 引脚
0	0	0	–	–	P52/ SLTI	P52/ SLTO	–	–	P31/ TI09	–	P74/ TI10	P73/ TO10	P75/ TI11	P30/ TO11
0	0	1	P52/ SLTI	P52/ SLTO	–	–	–	–	P31/ TI09	–	P74/ TI10	P73/ TO10	P75/ TI11	P30/ TO11
0	1	0	–	–	–	–	P52/ SLTI	P52/ SLTO	P31/ TI09	–	P74/ TI10	P73/ TO10	P75/ TI11	P30/ TO11
0	1	1	–	–	–	–	–	–	P52/ SLTI	P52/ SLTO	P74/ TI10	P73/ TO10	P75/ TI11	P30/ TO11
1	0	0	–	–	–	–	–	–	P31/ TI09	–	P52/ SLTI	P52/ SLTO	P75/ TI11	P30/ TO11
1	0	1	–	–	–	–	–	–	P31/ TI09	–	P74/ TI10	P73/ TO10	P52/ SLTI	P52/ SLTO
与上述不同的值			设置禁止											

ISC1	切换定时器阵列单元 TAUS 的通道 7 输入
0	使用 TI07 引脚的输入信号作为定时器输入(正常模式)。
1	RxD0 引脚的输入信号被用作定时器输入(唤醒信号检测)。

ISC0	切换外部中断 (INTP0)输入
0	使用 INTP0 引脚的输入信号作为外部中断(正常模式)。
1	使用 RxD0 引脚输入信号作为外部中断 (用来测量同步中断域和同步域的脉冲宽度)。

注意事项 确保将位 7 至 5 清“0”。

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

(24) 噪声滤波器使能寄存器 1、2(NFEN1、NFEN2)

NFEN1 和 NFEN2 用来设置是否要对每个通道的定时器输入信号使用噪声滤波器。

通过在噪声消除需要引脚上设置相应位为 1，允许噪声滤波器。

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

NFEN1 和 NFEN2 可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生可将这些寄存器清零(00H)。

图 6-23. 噪声滤波器使能寄存器 1, 2 (NFEN1, NFEN2) (1/2)的格式

地址: F0061H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	0	0

地址: F0062H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
NFEN2	0	0	0	TNFENSL	TNFEN11	TNFEN10	TNFEN09	0

TNFENSL	允许/禁止使用 SLTI/SLTO/P52 引脚输入信号的噪声滤波器
0	噪音过滤关闭
1	噪音过滤开启

TNFEN11	允许/禁止使用 TI11/SCK00/P75 引脚输入信号的噪声滤波器
0	噪音过滤关闭
1	噪音过滤开启

TNFEN10	使能/禁止使用 TI10/SI00/RxD0/P74 引脚输入信号的噪声滤波器
0	噪音过滤关闭
1	噪音过滤开启

TNFEN09	使能/禁止使用 TI09/SI10/RxD1/SDA10/INTP1/P31 引脚输入信号的噪声滤波器
0	噪音过滤关闭
1	噪音过滤开启

TNFEN07	允许/禁止使用 TI07/TO07/P51 引脚或 RxD0/SI00/P74 引脚输入信号的噪声滤波器 ^注
0	噪音过滤关闭
1	噪音过滤开启

TNFEN06	使能/禁止使用 TI06/TO06/P50 引脚输入信号的噪声滤波器
0	噪音过滤关闭
1	噪音过滤开启

注 通过设置 ISC 寄存器的 ISC1，可切换可用引脚。

ISC1 = 0: 可选择是否使用 TI07 引脚的噪声滤波器。

ISC1 = 1: 可选择是否使用 RxD0 引脚的噪声滤波器。

图 6-23. 噪声滤波器使能寄存器 1, 2 (NFEN1, NFEN2) (2/2)的格式

地址: F0061H 复位后: 00H R/W

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

地址: F0062H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
NFEN2	0	0	0	TNFENSL	TNFEN11	TNFEN10	TNFEN09	0

TNFEN05	允许/禁止使用 TI05/TO05/P13 引脚输入信号的噪声滤波器
0	噪音过滤关闭
1	噪音过滤开启

TNFEN04	允许/禁止使用 TI04/TO04/P12 引脚输入信号的噪声滤波器
0	噪音过滤关闭
1	噪音过滤开启

TNFEN03	允许/禁止使用 TI03/TO03/P11 引脚输入信号的噪声滤波器
0	噪音过滤关闭
1	噪音过滤开启

TNFEN02	允许/禁止使用 TI02/TO02/P10 引脚输入信号的噪声滤波器
0	噪音过滤关闭
1	噪音过滤开启

(25) 端口模式寄存器 1、3、5、7(PM1、PM3、PM5、PM7)

这些寄存器以 1 位为单元设定端口 1、3、5 和 7 的输入/输出。

当使用 P10/TO02/TI02、P11/TO03/TI03、P12/TO04/TI04、P13/TO05/TI05、P50/TO06/TI06、P51/TO07/TI07、P73/TO10/SO00/TxD0 和 P30/TO11/SO10/TxD1 引脚作为定时器输出时，设置 PM10 至 PM13、PM30、PM50、PM51 和 PM73，以及 P10 至 P13、P30、P50、P51 和 P73 的输出锁存为 0。

当使用 P10/TO02/TI02、P11/TO03/TI03、P12/TO04/TI04、P13/TO05/TI05、P50/TO06/TI06、P51/TO07/TI07、P31/TI09/SI10/RxD1/SDA10/INTP1、P74/TI10/SI00/RxD0 和 P75/TI11/SCK00 引脚作为定时器输入时，设置 PM10 至 PM13、PM31、PM50、PM51、P74 和 PM75 为 1。此时，输出锁存 P10 至 P13、P31、P50、P51、P74 和 P75 可能为 0 或 1。

PM1、PM3、PM5 和 PM7 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生将这些寄存器设置为 FFH。

图 6-24. 端口模式寄存器 1、3、5 和 7(PM1、PM3、PM5、PM7)的格式

地址: FFF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	1	1	1	1	PM13	PM12	PM11	PM10

地址: FFF23H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	PM32	PM31	PM30

地址: FFF25H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM5	1	1	1	1	1	PM52	PM51	PM50

地址: FFF27H 复位后: FFH R/W

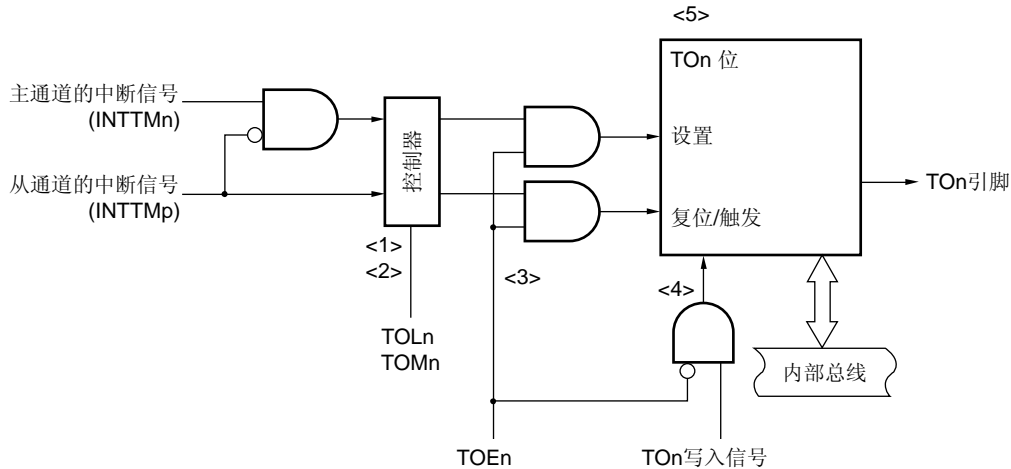
符号	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PMmn	Pmn 引脚 I/O 模式选择 (m = 1, 3, 5, 7; n = 0 ~ 5)
0	输出模式(输出缓冲器开)
1	输入模式(输出缓冲器关)

6.4 通道输出 (TOn引脚) 控制

6.4.1 TOn引脚输出电路配置

图 6-25. 输出电路配置



TOn 引脚输出电路描述如下。

<1> TOMn = 0 (反转模式) 时, 忽略 TOLn 位的设置值并且只有 INTTMp (从通道的定时器中断) 发送到 TOn 位。

<2> TOMn = 1 (组合操作模式) 时, INTTMn (主通道的定时器中断) 和 INTTMp (从通道的定时器中断) 均发送到 TOn 位。

这时, TOLn 位有效, 并且信号的控制如下:

当 TOLn = 0: 正向操作(INTTMn →置位, INTTMp → 复位)

当 TOLn = 1: 反向操作(INTTMn →复位, INTTMp → 置位)

INTTMn 和 INTTMp 同时产生时, (PWM 输出 0%时), INTTMp (复位信号) 优先, 并且 INTTMn (设置信号)被屏蔽。

<3> TOEn = 1 时, INTTMn (主通道的定时器中断) 和 INTTMp (从通道的定时器中断) 发送到 TOn 位。TOn 位写入 (TOn 写入信号) 无效。

TOEn = 1 时, TOn 引脚输出只随中断信号改变。

要初始化 TOn 引脚输出电平, 需要设置 TOEn = 0 并且给 TOn 写入一个值。

<4> TOEn = 0 时,写入目标通道的 TOn 位 (TOn 写入信号) 有效。TOEn = 0 时, INTTMn (主通道的定时器中断) 和 INTTMp (从通道的定时器中断) 都不发送到 TOn 位。

<5> TOn 位总是可读, 可以检测 TOn 引脚输出电平。

备注 n: 通道编号, p: 从通道编号

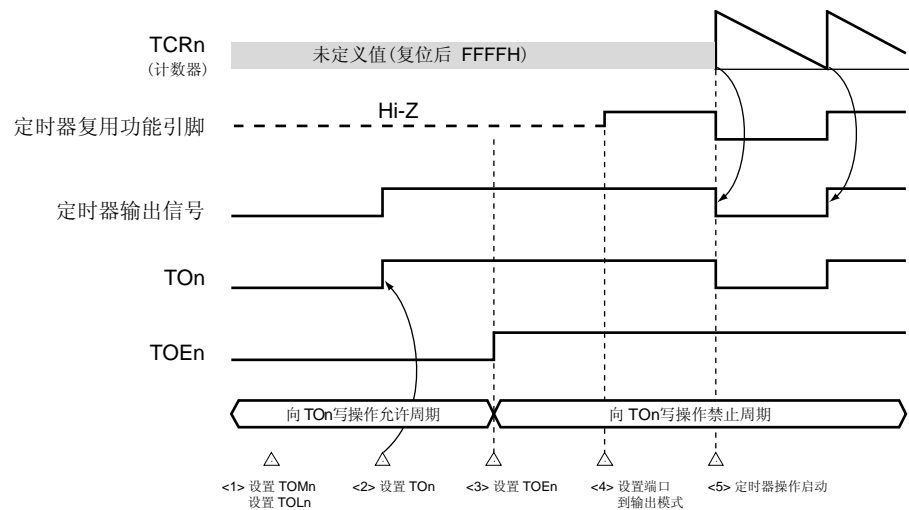
n = 00 至 11 (然而, 作为主通道的情况下, n = 00、02、04、06、08、10)

n < p ≤ 11 (关于主通道和从通道之间关系的详细信息, 参考 6.6.3 组合操作功能的基本规则应用范围。)

6.4.2 TOn引脚输出设置

下图所示为 TOn 输出引脚从初始设置到定时器操作开始的过程和状态迁移。

图 6-26. 从定时器输出设置到操作开始的状态迁移



<1> 设置定时器输出的操作模式。

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

<2> 定时器输出信号通过设置 TOn 恢复到初始状态。

<3> 通过写入 1 到 TOEn 可以允许定时器输出操作 (TOn 禁止写入)。

<4> 端口 I/O 设置为输出 (参考 6.3 (25) 端口模式寄存器 1、3、5、7)。

<5> 定时器操作允许(TSn = 1)。

备注 n = 00~11

6.4.3 通道输出操作的注意事项

(1) 定时器操作期间改变寄存器 TO0, TOE0 和 TOL0 中的设置值

因为定时器操作(TCRn 和 TDRn 的操作) 与 TOn 输出电路无关, 并且改变设置 TO0, TOE0 和 TOL0 的值不会影响定时器操作, 所以在定时器操作期间这些值可以被改变。

当 TOE0, TOL0 和 TOM0 (除 TO0 外) 中设置的值在靠近定时器中断 (INTTMn) 时改变, 根据该值是紧随定时器中断 (INTTMn) 信号之前或之后改变, 输出到 TOn 引脚的波形可能会不同。

备注 n = 00~11

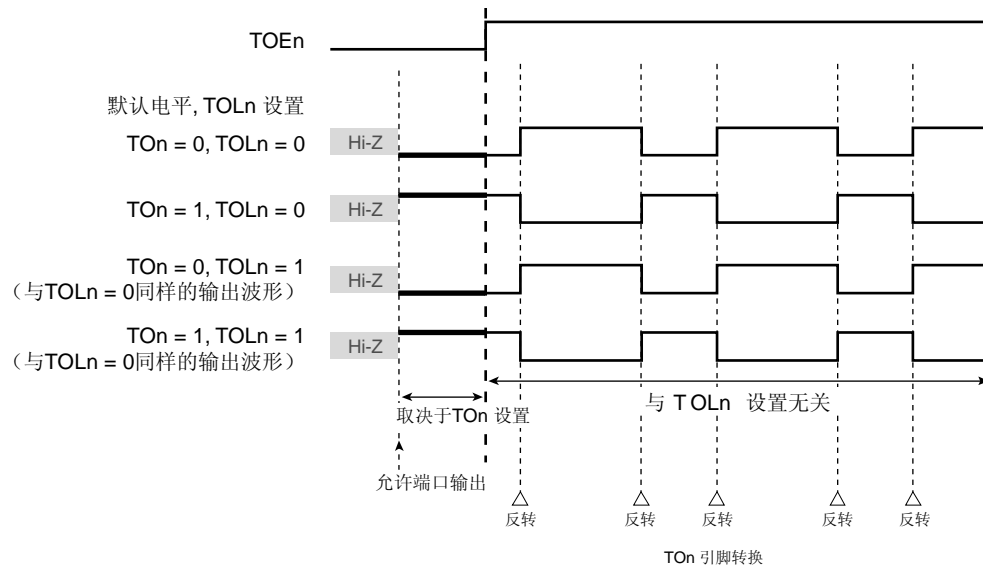
(2) TOn 引脚的默认电平和定时器操作启动后的输出电平

在改变默认电平之后, 在允许端口输出和设置 TOEn = 1 之前, 当在 TOEn = 0 的状态写入完成时 TOn 引脚输出电平的迁移如下图所示。

(a) 设置 TOMn = 0 启动操作时 (反转输出)

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

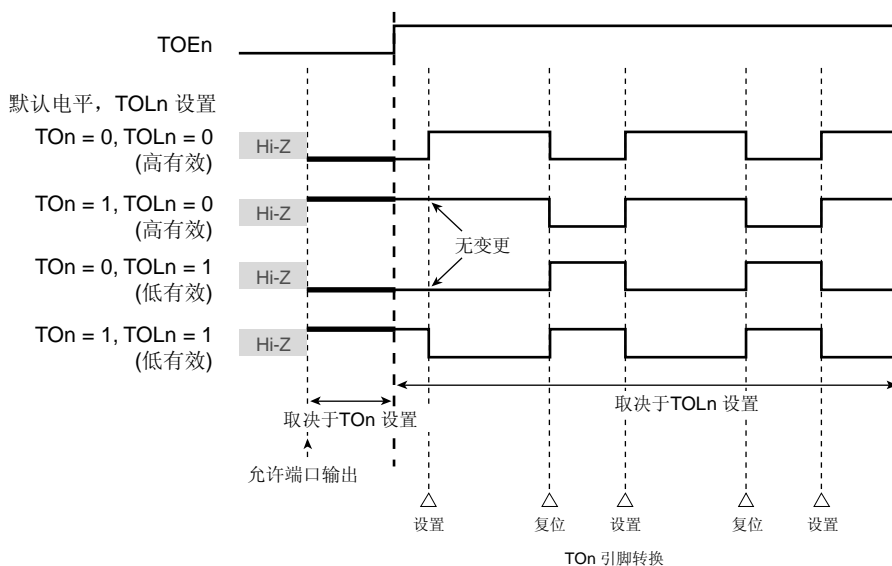
图 6-27. 反转输出时 TOn 引脚输出状态 (TOMn = 0)



备注 1. 反转: 相反的 TOn 引脚输出状态
2. n = 00~11

(b) 设置 $TOMn = 1$ 启动操作时(组合操作模式 (PWM 输出))

$TOMn = 1$ 时, 有效电平由 $TOLn$ 设置决定。

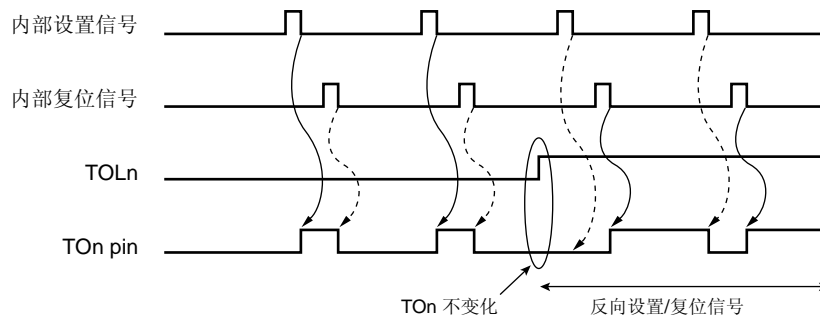
图 6-28. PWM 输出时 TON 引脚输出状态 ($TOMn = 1$)

- 备注**
1. 设置: TON 引脚的输出信号从无效电平变为有效电平。
复位: TON 引脚的输出信号从有效电平变为无效电平。
 2. $n = 00 \sim 11$

(3) TON 引脚在组合操作模式 ($TOMn = 1$) 下的操作(a) 当 $TOLn$ 设置在定时器操作期间已经被改变

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

下图所示为 $TOLn$ 的值在定时器操作期间 ($TOMn = 1$) 被改变时的操作。

图 6-29. 当 $TOLn$ 在定时器操作期间被改变时的操作

- 备注**
1. 设置: TON 引脚的输出信号从无效电平变为有效电平。
复位: TON 引脚的输出信号从有效电平变为无效电平。
 2. $n = 00 \sim 11$

(b) 置位/复位时序

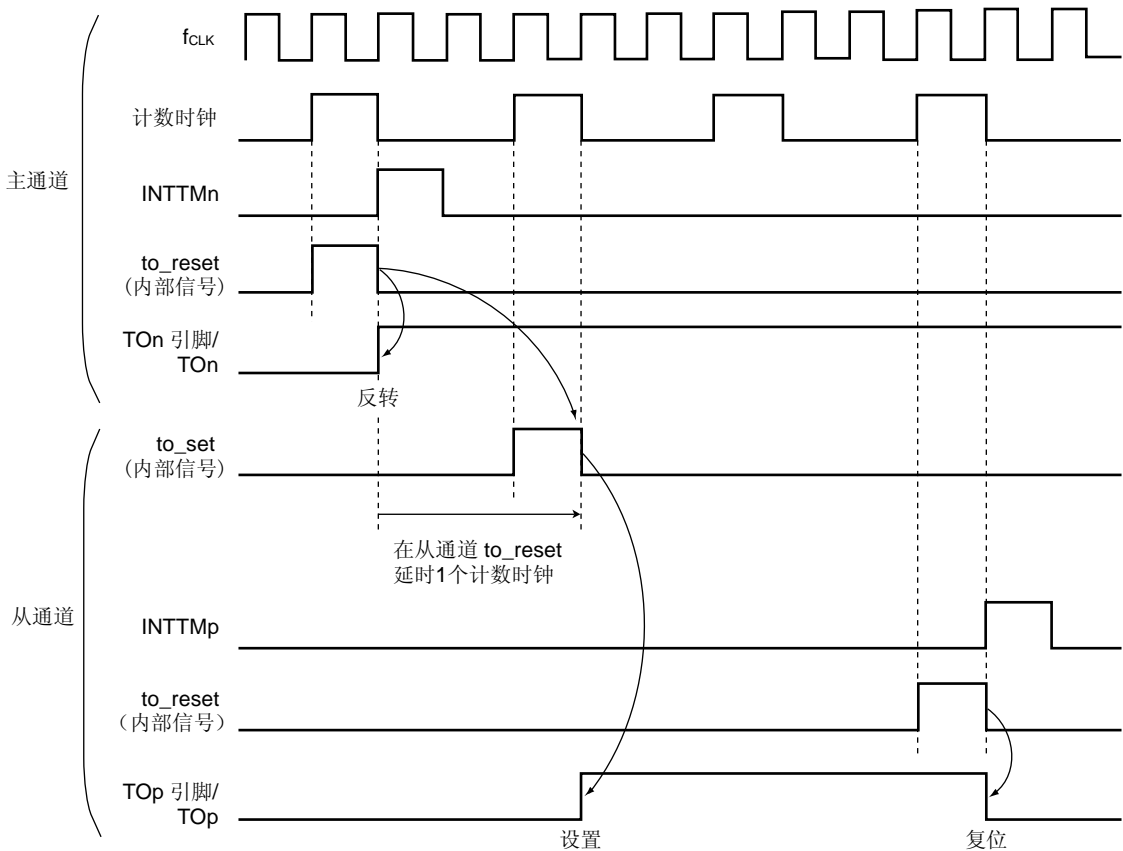
PWM 输出要实现 0%或 100%输出，主通道定时器中断(INTTMn)发生时，TOn 引脚/TOn 的设置时序通过从通道的定时器中断(INTTMp)延迟一个计数时钟。

如果设置条件和复位条件同时被满足，复位具有较高的优先级。

图 6-30 所示为主/从通道设置如下时，置位/复位操作状态。

主通道： TOEn = 1、TOMn = 0、TOLn = 0
从通道： TOEp = 1、TOMp = 1、TOLp = 0

图 6-30. 置位/复位时序操作状态

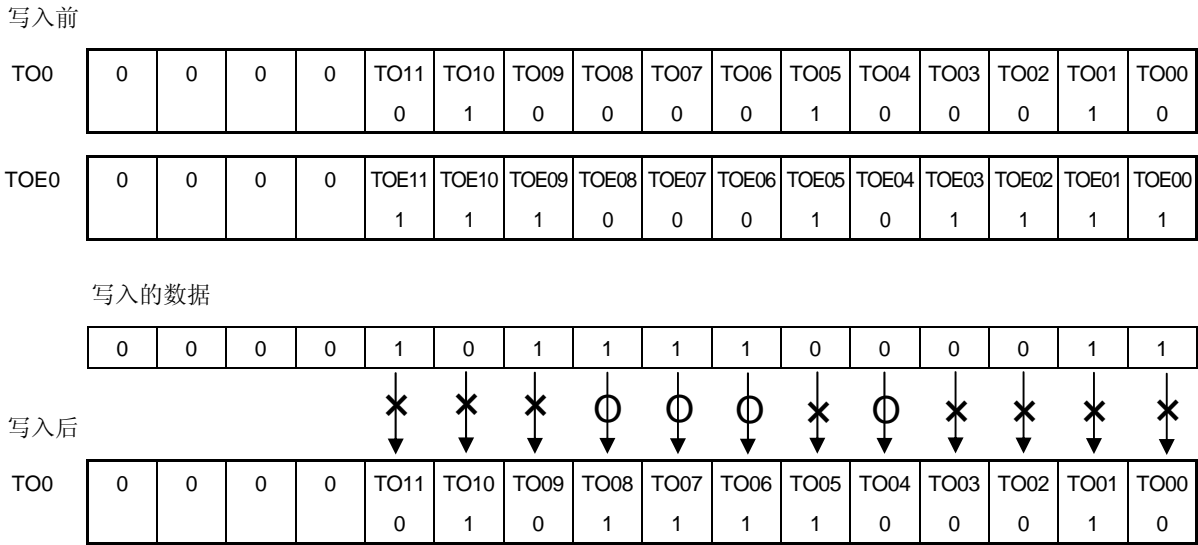


- 备注
- 1. to_reset: TOn 引脚复位/反转信号
to_set: TOn 引脚置位信号
 - 2. n: 通道编号, p: 从通道编号
n = 00 ~ 11 (然而, 作为主通道的情况下, n = 00、02、04、06、08、10)
n < p ≤ 11 (关于主通道和从通道之间关系的详细信息, 参考 6.6.3 组合操作功能的基本规则应用范围。)

6.4.4 TOn位的集中操作

在 TO0 寄存器中，所有通道的设置位可在一个寄存器中定位，与 TS0 寄存器(通道启动触发器)相同。因此所有通道的 TOn 可集中操作。另外只有通过设置和想要操作的 Ton(通道输出)所对应的 TOEn = 0 才可以进行逐位操作。

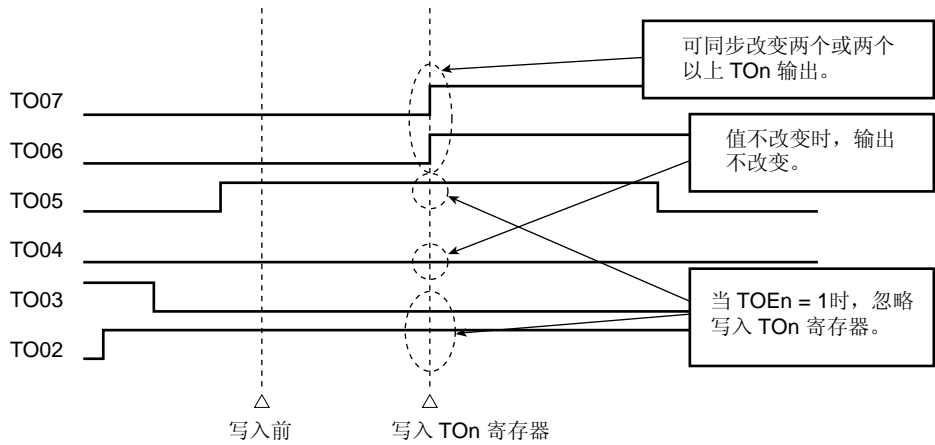
图 6-31. TOn 位集中操作的示例



只有 TOEn = 0 才可以写入 TOn 位，忽略 TOEn = 1 的 TOn 的写入。

设置为 TOEn = 1 的 TOn (通道输出) 不受写操作影响。即使对 TOn 进行写操作，也会被忽略并且由定时器操作进行的输出改变照常完成。

图 6-32. TOn 位的集中操作影响 TOn 引脚状态



注意事项 当 TOEn = 1 时，即使各个定时器的定时器中断 (INTTMn) 输出和写入 TOn 的内容有冲突，TOn 引脚也可以正常输出。

备注 n = 00~11

6.4.5 关于计数操作开始时的定时器中断和 TOn 引脚输出

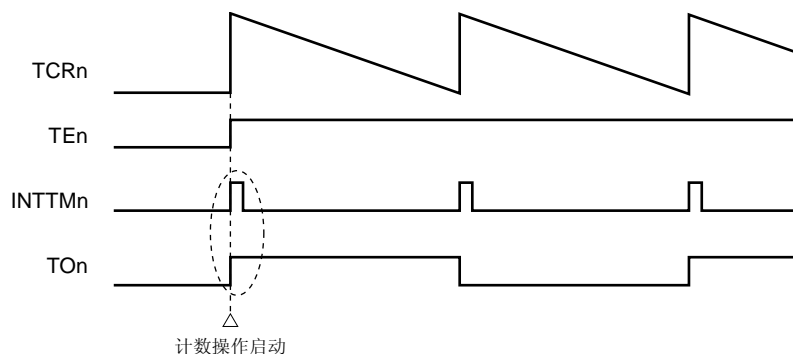
在间隔定时器模式或捕捉模式下，TMRn 寄存器中的 MDn0 位用于设置是否在计数启动时产生定时器中断。

当 MDn0 设置为 1 时，通过定时器中断 (INTTMn) 的产生就可以知道计数操作开始时间。

在其它模式下，在计数操作开始的定时器中断和 TOn 输出都不进行控制。

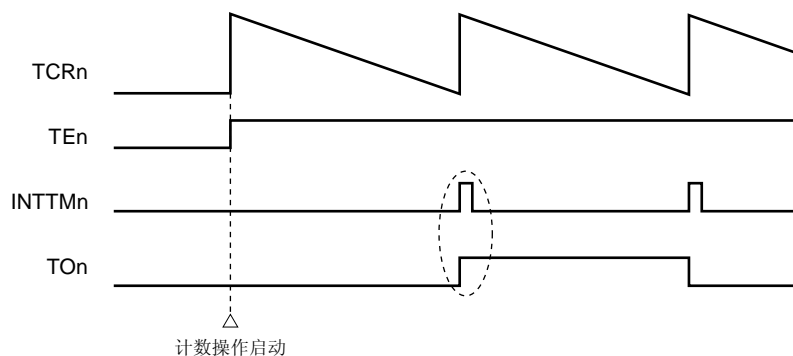
图 6-32 和 6-33 所示为使用间隔定时器模式 (TOEn = 1、TOMn = 0) 时的操作示例。

图 6-33. 当 MDn0 设置为 1



MDn0 设置为 1 时，定时器中断 (INTTMn) 在计数操作开始时输出，并且 TOn 执行反转操作。

图 6-34. 当 MDn0 设置为 0



MDn0 设置为 0 时，定时器中断 (INTTMn) 在计数操作开始时不输出，并且 TOn 也不改变。一个计数周期后，INTTMn 输出并且 TOn 执行反转操作。

备注 n = 00~11

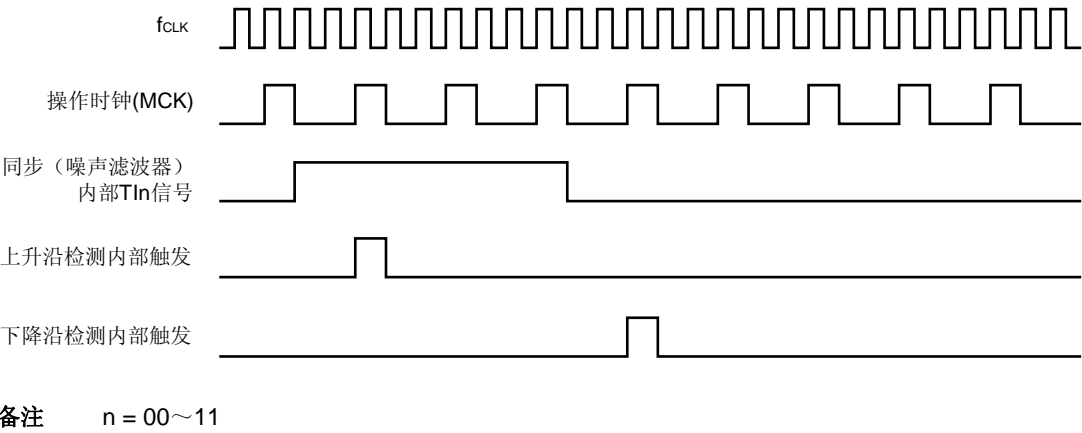
6.5 通道输入 (TIn引脚) 控制

6.5.1 TIn边沿检测电路

(1) 沿检测基本操作时序

沿检测电路采样的执行与工作时钟(MCK)一致。

图 6-35. 沿检测基本操作时序



6.6 定时器阵列单元TAUS的基本功能

6.6.1 独立操作功能和组合操作功能的概述

定时器阵列单元 TAUS 由若干个通道组成，独立操作功能允许每个通道独立操作，也可以和两个或更多通道组合在一起实现组合操作功能。

任意通道在其它通道的任何操作模式下均可使用独立操作功能，。

组合操作功能通过将一个主通道(主要是计数周期的参考定时器)和一个从通道(与主通道操作一致的定时器)组合在一起实现，使用此功能时必须遵守下面几条规则。

6.6.2 组合操作功能的基本规则

使用组合操作功能的基本规则如下所示。

(1) 只有偶数通道(通道 0、2、4 等)可设置为主通道。

(2) 除通道 0 以外的任意通道都可设置为从通道。

(3) 从通道必须低于主通道。

举例：如果通道 2 设置为主通道，通道 3 以后的通道 (通道 3、4、5，等)可设置为从通道。

(4) 一个主通道可以设置两个或更多从通道。

(5) 当使用两个以上主通道时，与从通道之间有主通道的从通道不能被设置。

举例：如果通道 0 和 4 设置为主通道，通道 1 到 3 可被设置为主通道 0 的从通道。通道 5~7 不能设置作为主通道 0 的从通道。

(6) 主通道可以发送 INTTMn (中断)，启动软件触发，计数时钟到更低通道。

(7) 从通道可使用主通道的 INTTMn (中断)，启动软件触发，和计数时钟，但是不能发送自己的 INTTMn (中断)，启动软件触发，和计数时钟到更低通道。

(8) 主通道不能使用来自更高主通道的 INTTMn (中断)，启动软件触发和计数时钟。

(9) 要同时启动组合操作的通道，组合通道的 TSn 位必须同时设置。

(10) 要同时停止参与组合的通道，组合通道的 TTn 位必须同时设置。

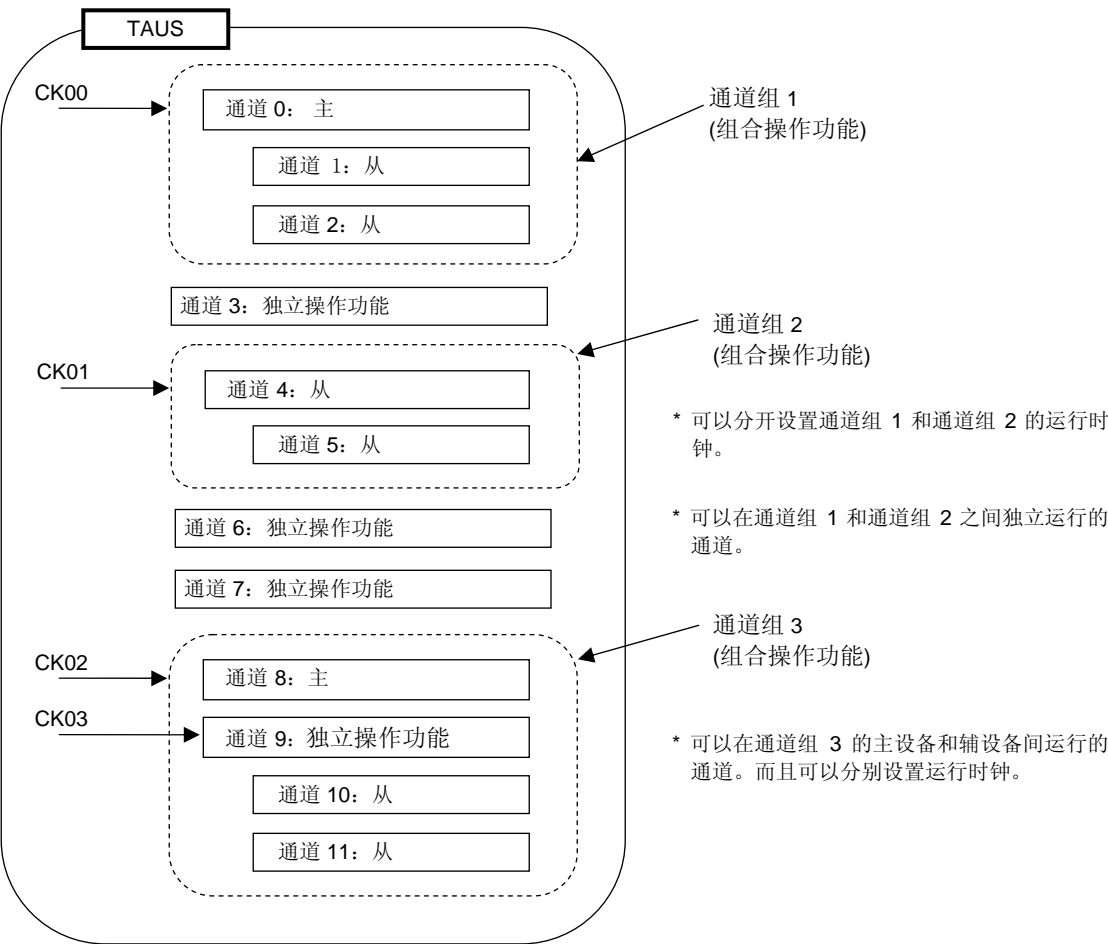
备注 n = 00 ~ 11 (然而，作为主通道的情况下，n = 00、02、04、06、08、10。)

6.6.3 组合操作功能的基本规则的应用范围

组合操作功能的规则是适用于一个通道组(一个主通道和从通道组成一个组合操作功能)的规则。

如果指定两个或更多不能组合操作的通道组，则 6.6.2 组合操作功能的基本规则中的组合操作功能的基本规则不会应用到通道组。

实例



6.7 作为独立通道的定时器阵列单元TAUS的操作

6.7.1 作为间隔定时器/方波输出的操作

(1) 间隔定时器

定时器阵列单元可用作参考定时器，产生固定间隔的 INTTMn (定时器中断)。
可通过下面的表达式计算出中断产生周期。

$$\text{INTTMn (定时器中断) 的产生周期} = \text{计数时钟周期} \times (\text{TDRn 的设置值} + 1)$$

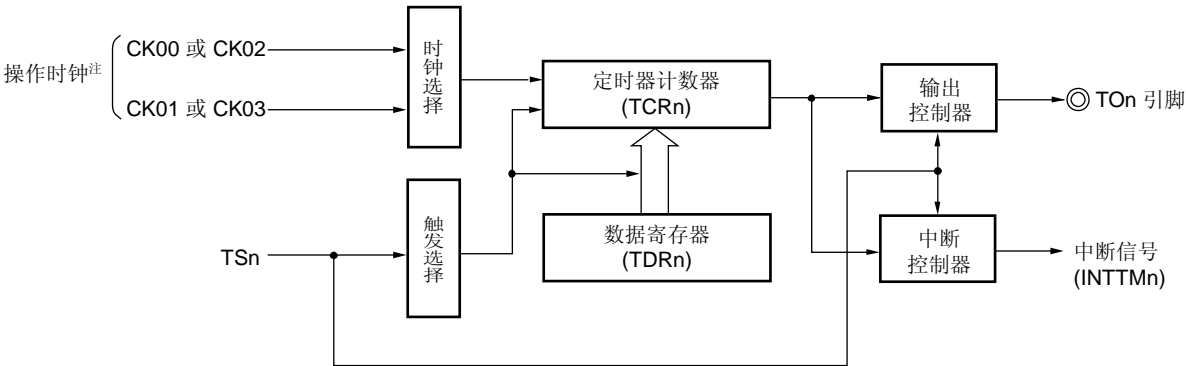
(2) 作为方波输出操作

一旦 INTTMn 发生，TOn 就执行反转操作，并且输出一个占空比为 50%的方波。
TOn 输出方波的周期和频率可以通过下面的表达式计算出。

- 从 TOn 方波输出的周期 = 计数时钟周期 × (TDRn 的设置值 + 1) × 2
 - 从 TOn 方波输出的频率 = 计数时钟频率 / {(TDRn 的设置值 + 1) × 2}

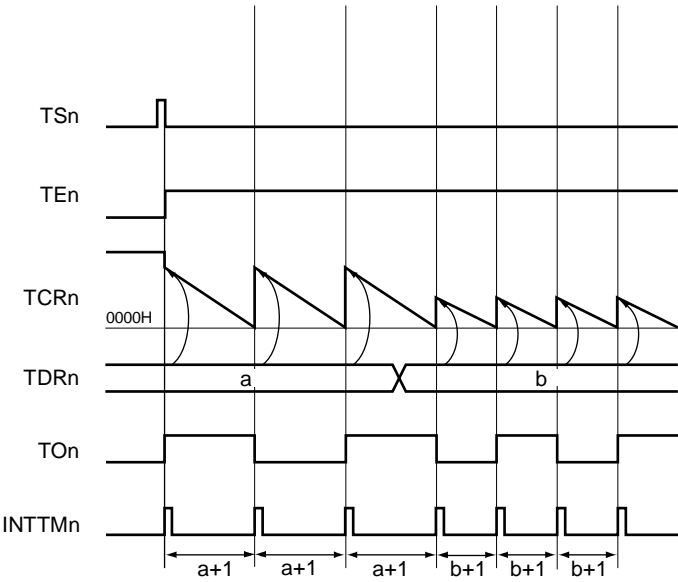
TCRn 在间隔定时器模式下作为向下计数器操作。
通道启动触发位(TSn)设置为 1 后，在第一个计数时钟时，TCRn 加载 TDRn 的值。如果此时 TMRn 的 MDn0 = 0，则不产生 INTTMn 并且 TOn 不反转。如果 TMRn 的 MDn0 = 1，输出 INTTMn 并且 TOn 反转。
之后，TCRn 随着计数时钟而向下计数。
当 TCRn = 0000H 时，INTTMn 输出并且 TOn 在下一个计数时钟反转。此时 TCRn 再次加载 TDRn 的值。之后，重复相同的操作。
TDRn 在任何时间可以重写。TDRn 的新值从下个周期开始有效。

图 6-36. 作为间隔定时器/方波输出操作的框图



注 通道 0 至 7 的工作时钟从 CK00 和 CK01 中选择，通道 8 至 11 的工作时钟从 CK02 和 CK03 中选择。
备注 n = 00~11

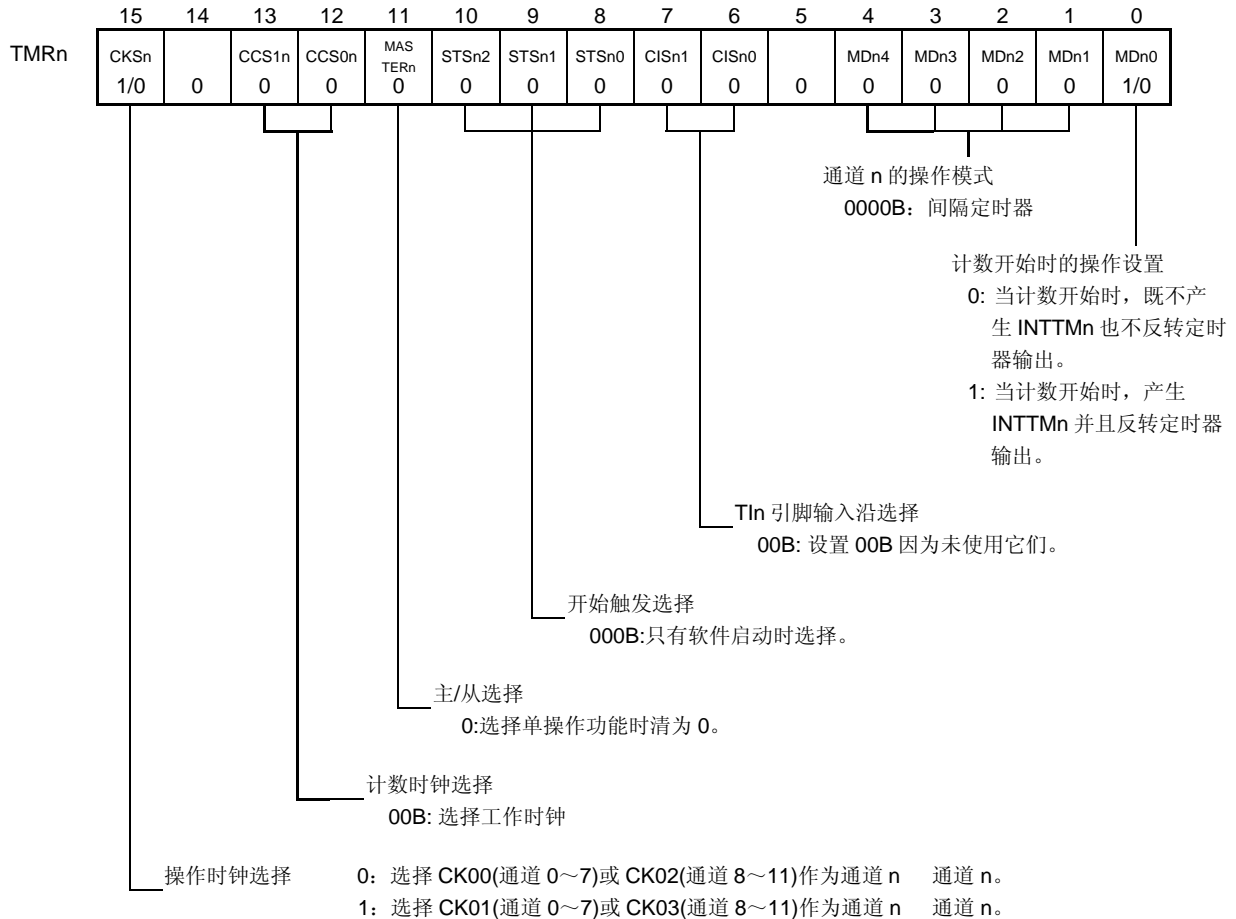
图 6-37. 作为间隔定时器/方波输出 (MDn0 = 1) 操作的基本时序示例



备注 n = 00~11

图 6-38. 作为间隔定时器/方波输出操作期间的寄存器设置内容的示例

(a) 定时器模式寄存器 n(TMRn)



(b) 定时器输出寄存器 0(TO0)

第 n 位	
TO0	TOn 1/0
	0: 从 TOn 输出 0。 1: 从 TOn 输出 1。

(c) 定时器输出使能寄存器 0(TOE0)

第 n 位	
TOE0	TOEn 1/0
	0: 通过计数操作, 停止 TOn 输出。 1: 通过计数操作, 允许 TOn 输出。

(d) 定时器输出电平寄存器 0(TOL0)

第 n 位	
TOL0	TOLn 0
	0: 当 TOMn = 0 时 (反转模式) 清零。

(e) 定时器输出模式寄存器 0(TOM0)

第 n 位	
TOM0	TOMn 0
	0: 设置反转模式。

备注 n = 00~11

图 6-39. 间隔定时器/方波输出功能的操作过程

	软件操作	硬件状态
	TAUS 默认设置	掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER2 寄存器的 TAU0EN 位为 1。—————▶	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
	设置 TPS0 寄存器。 把 CK00 和 CK01 的时钟频率用于通道 0~7, 把 CK02 和 CK03 的时钟频率用于通道 8~11。	
通道默认设置	设置 TMRn 寄存器 (决定通道的操作模式)。	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置间隔 (周期) 值到 TDRn 寄存器。	
	使用 TOn 输出时 把 TOM0 寄存器的 TOMn 位清零 (反转模式)。 把 TOLn 位清零。 设置 TOn 位, 并赋予 TOn 默认输出电平。—————▶	TOn 引脚进入高阻输出状态。
	设置 TOEn 至 1 并允许 TOn 操作。—————▶	当端口模式寄存器在输出模式下并且端口寄存器为 0 时, TOn 默认设置电平为输出。
	将端口寄存器和端口模式寄存器清为 0。—————▶	因为通道停止运行, 所以 TOn 不会改变。 TOn 引脚输出 TOn 设置电平。
操作开始	设置 TOEn 为 1 (只当操作恢复时)。 设置 TSn 位为 1。—————▶ 由于是触发位, TSn 位自动恢复为 0。	TEn=1, 并且计数操作开始。 在计数时钟输入时 TDRn 的值被加载到 TCRn。如果 TMRn 寄存器的 MDn0 位为 1 时产生 INTTMn, 并且 TOn 执行反转操作。
操作期间	TMRn, TOM0 和 TOL0 寄存器的设置值不能改变。 TDRn 寄存器的设置值不能改变。 TCRn 寄存器总是可读。 不使用 TSRn 寄存器。 TO0 和 TOE0 寄存器的设置值可以改变。	计数器 (TCRn) 向下计数。当计数值达到 0000H 时, TDRn 的值被再次加载到 TCRn, 并且计数操作继续。通过检测 TCRn=0000H, INTTMn 产生并且 TOn 执行反转操作。之后, 重复上面的操作。
操作停止	TTn 位设置为 1。—————▶ 由于是触发位, TTn 位自动恢复为 0。	TEn=0, 并且计数操作停止。 TCRn 保持计数值并停止。 TOn 输出不被初始化而保持当前状态。
	TOEn 清除为 0 并且值设置到 TOn 位。—————▶	TOn 引脚输出 TOn 设置电平。
TAUS 停止	需要保持 TOn 引脚输出电平时 在把需要保持的值设置到端口寄存器后, 清零 TOn 位。—————▶	TOn 引脚输出由电平端口功能保持。
	不需要保持 TOn 引脚输出电平时, 将端口模式寄存器切换到输入模式。—————▶	tmf TOn 引脚输出电平变为高阻输出状态。
	PER2 寄存器的 TAU0EN 位清零。—————▶	掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。 (TOn 位清为 0, 并且 TOn 引脚设置为端口模式。)

继续操作。

备注 n = 00~11

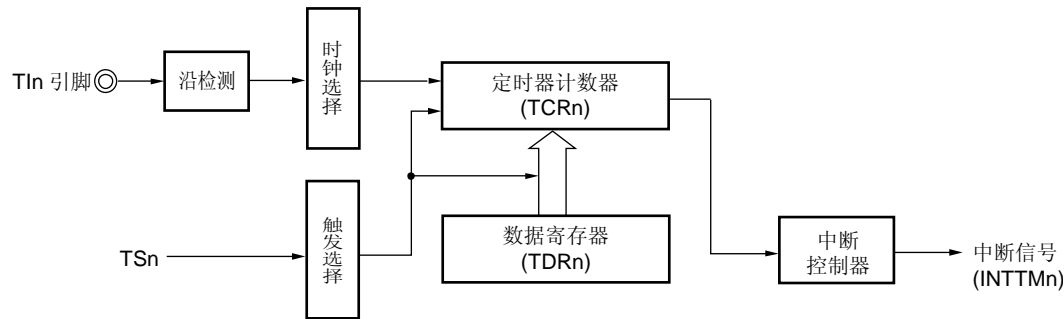
6.7.2 作为外部事件计数操作

定时器阵列单元可用作外部事件计数，对 **TIn** 引脚检测到的有效输入沿 (外部事件) 的数量进行计数。当达到指定计数值时，事件计数器产生一个中断。指定的计数号可通过下面的表达式计算得出。

指定的计数值 = TDRn 的设置值 + 1

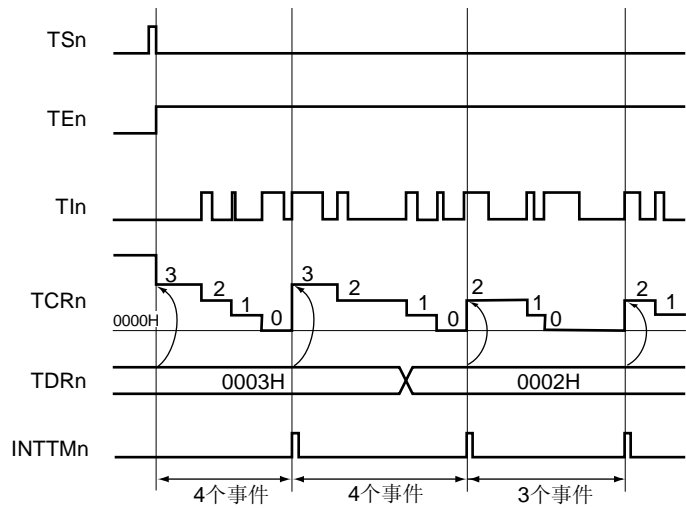
在事件计数模式下 **TCRn** 作为向下计数器操作。
通道启动触发位 (**TSn**) 设置为 1 时，**TCRn** 加载 **TDRn** 的值。
每次在 **TIn** 引脚检测到有效输入沿时，**TCRn** 递减一次。当 **TCRn** = 0000H 时，**TCRn** 再次加载 **TDRn** 的值并输出 **INTTMn**。
之后，重复上面的操作。
TON 不能被使用，因为它的波形由外部事件决定，并且不规则。
TDRn 在任何时间可以重写。在下一个计数周期 **TDRn** 的新值有效。

图 6-40. 作为外部事件计数器操作的框图



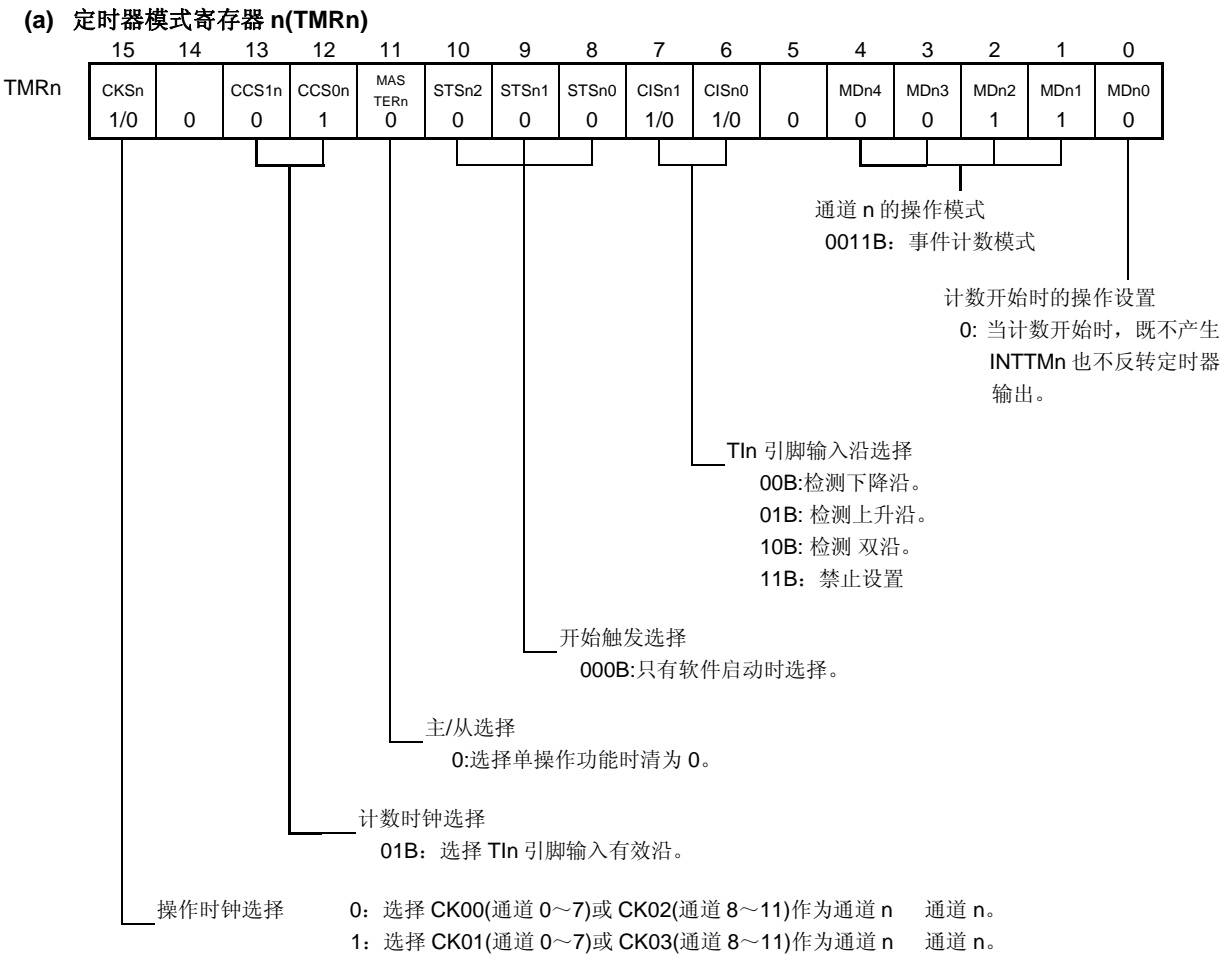
备注 n = 00~11

图 6-41. 作为外部事件计数器操作的基本时序示例



备注 n = 00~11

图 6-42. 在外部事件计数器模式下寄存器设置内容的示例



备注 n = 00~11

图 6-43. 使用外部事件计数器功能时的操作过程

	软件操作	硬件状态
继续操作。	TAUS 默认设置	掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER2 寄存器的 TAU0EN 位为 1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
	设置 TPS0 寄存器。 把 CK00 和 CK01 的时钟频率用于通道 0~7, 把 CK02 和 CK03 的时钟频率用于通道 8~11。	
	通道默认设置 设置 TMRn 寄存器 (决定通道的操作模式)。 设置计数值到 TDRn 寄存器。 把 TOE0 寄存器的 TOEn 位清零。	通道停止操作。 (供给时钟并且产生一些功耗。)
	操作开始 设置 TSn 位为 1。——> 由于是触发位, TSn 位自动恢复为 0。	TEn=1, 并且计数操作开始。 TDRn 的值被加载到 TCRn 并且等待 TIn 引脚输入沿的检测。
	操作期间 TDRn 寄存器的设置值不能改变。 TCRn 寄存器总是可读。 不使用 TSRn 寄存器。 TMRn、TOM0、TOL0、TO0 和 TOE0 寄存器的设置值不能改变。	计数器 (TCRn) 在每次检测到 TIn 引脚输入沿时向下计数。 当计数值达到 0000H 时, TDRn 的值被再次加载到 TCRn, 并且计数操作继续。通过检测 TCRn = 0000H, 产生 INTTMn 输出。 之后, 重复上面的操作。
	操作停止 TTn 位设置为 1。——> 由于是触发位, TTn 位自动恢复为 0。	TEn=0, 并且计数操作停止。 TCRn 保持计数值并停止。
	TAUS 停止 PER2 寄存器的 TAU0EN 位清零。——>	掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。

备注 n = 00~11

6.7.3 作分频器操作

定时器阵列单元可用作分频器，把输入到 TIn 引脚和从 TOn 输出结果的时钟分频。
从 TOn 输出的分频时钟频率可由下面的表达式计算。

- 当选择上升沿/下降沿时:
分频时钟频率=输入时钟频率/((TDRn 的设置值 + 1) × 2)
 - 当选择双沿时:
分频时钟频率≡ 输入时钟频率/(TDRn 的设定值 + 1)

TCRn 在间隔定时器模式下作为向下计数器操作。

通道启动触发位 (TSn) 设置为 1 后，当检测到 TIn 有效沿时，TCRn 加载 TDRn 的值。如果此时 TMRn 的 MDn0 = 0，INTTMn 不输出并且 TOn 不反转。如果 TMRn 的 MDn0 = 1，输出 INTTMn 并且 TOn 反转。

之后，TCRn 在 TIn 有效沿向下计数。当 TCRn = 0000H 时，触发 TOn。同时，TCRn 再次加载 TDRn 的值并继续计数。

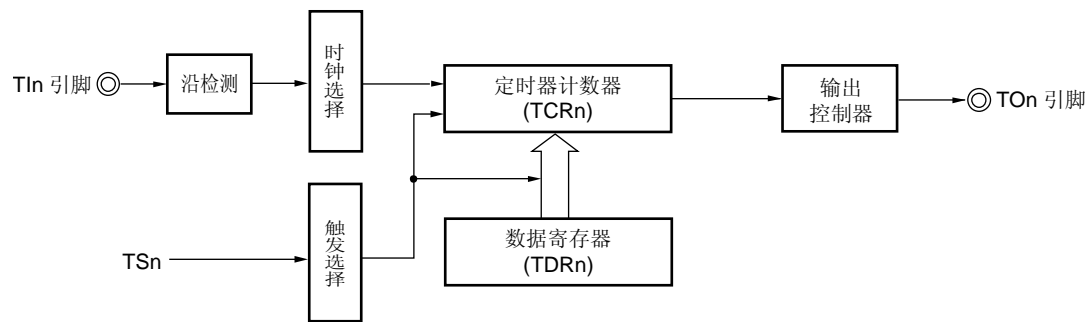
如果选择 TIn 的双沿检测，输入时钟的占空比误差将影响 TOn 输出的分频时钟周期。

TOn 输出时钟周期包括工作时钟的一个周期的一个采样误差。

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

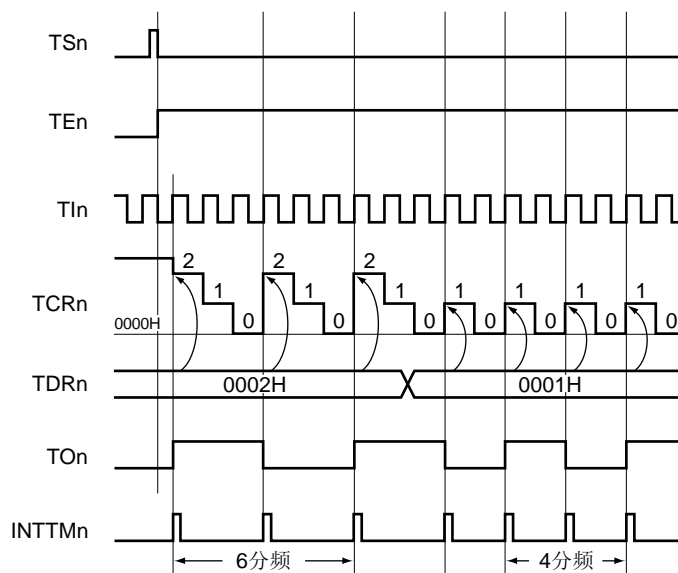
TDRn 在任何时间可以重写。在下一个计数周期 TDRn 的新值有效。

图 6-44. 作为分频器操作的框图



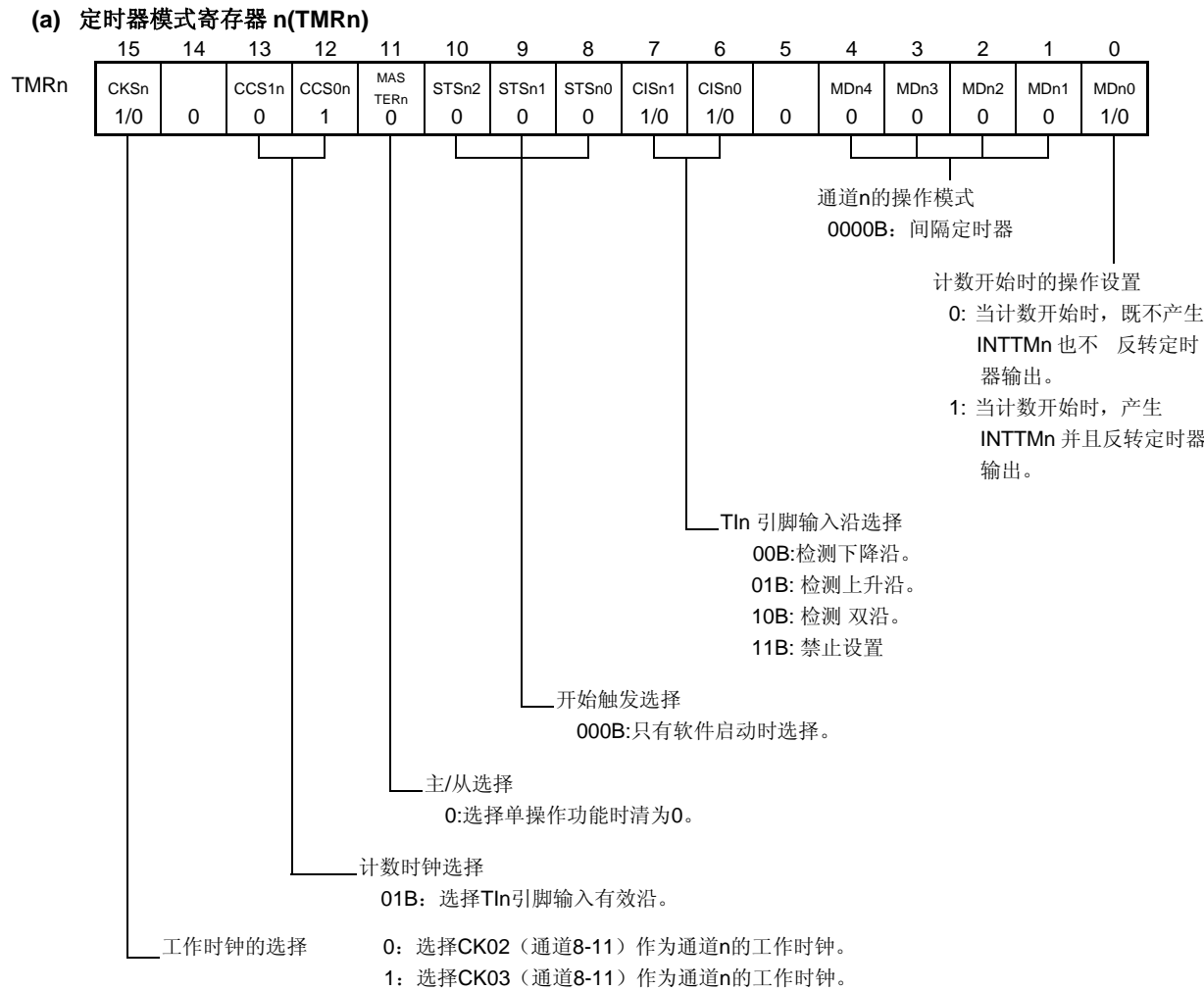
备注 n = 10, 11

图 6-45. 作为分频器 (MDn0 = 1) 操作的基本时序示例



备注 n = 10, 11

图 6-46. 作为分频器操作期间的寄存器设置内容的示例



备注 n = 10, 11

图 6-47. 当使用分频功能时的操作过程

	软件操作	硬件状态
继续操作。	TAUS 默认设置	掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER2 寄存器的 TAU0EN 位为 1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
	设置 TPS0 寄存器。 把 CK00 和 CK01 的时钟频率用于通道 0~7, 把 CK02 和 CK03 的时钟频率用于通道 8~11。	
	通道默认设置	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置 TMRn 寄存器 (决定通道的操作模式)。 设置间隔 (周期) 值到 TDRn 寄存器。 将 TOM0 寄存器的 TOMn 位清除为 0 (反转模式)。 把 TOLn 位清零。 设置 TOn 位, 并赋予 TOn 默认输出电平。 设置 TOEn 至 1 并允许 TOn 操作。 将端口寄存器和端口模式寄存器清为 0。	TON 引脚进入高阻输出状态。 当端口模式寄存器在输出模式下并且端口寄存器为 0 时, TOn 默认设置电平为输出。 因为通道停止运行, 所以 TOn 不会改变。 TON 引脚输出 TOn 设置电平。
继续操作。	操作开始	设置 TOEn 为 1 (仅当操作恢复时)。 设置 TSn 位为 1。 由于是触发位, TSn 位自动恢复为 0。
		TEn=1, 并且计数操作开始。 在计数时钟输入时 TDRn 的值被加载到 TCRn。如果 TMRn 寄存器的 MDn0 位为 1 时产生 INTTMn, 并且 TOn 执行反转操作。
	操作期间	TDRn 寄存器的设置值不能改变。 TCRn 寄存器总是可读。 不使用 TSRn 寄存器。 TO0 和 TOE0 寄存器设置值能改变。 TMRn, TOM0 和 TOL0 寄存器的设置值不能改变。
		计数器 (TCRn) 向下计数。当计数值达到 0000H 时, TDRn 的值被再次加载到 TCRn, 并且计数操作继续。通过检测 TCRn=0000H, INTTMn 产生并且 TOn 执行反转操作。之后, 重复上面的操作。
继续操作。	操作停止	TTn 位设置为 1。 由于是触发位, TTn 位自动恢复为 0。
		TEn=0, 并且计数操作停止。 TCRn 保持计数值并停止。 TON 输出不被初始化而保持当前状态。
		TOEn 清除为 0 并且值设置到 TO0n 位。 TON 引脚输出 TOn 设置电平。
继续操作。	TAUS 停止	需要保持 TOn 引脚输出电平时 在把需要保持的值设置到端口寄存器后, 清零 TOn 位。
		不需要保持 TOn 引脚输出电平时 切换端口模式寄存器到输入模式。
		TON 引脚输出电平由端口功能保持。 TON 引脚输出电平变为高阻输出状态。
备注	PER2 寄存器的 TAU0EN 位清零。	掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。 (TON 位清除为 0, 并且 TOn 引脚设置为端口模式)。

n = 10, 11

6.7.4 作为输入脉冲间隔测量的操作

在 TIn 有效沿可以捕捉计数值，可以测量输入到 TIn 的脉冲间隔。
脉冲间隔可通过下面的公式计算。

$$\text{TIn 输入脉冲间隔} = \text{计数时钟周期} \times ((10000\text{H} \times \text{TSRn: OVF}) + (\text{TDRn 的捕捉值} + 1))$$

注意事项 对 TIn 引脚输入进行采样的工作时钟是由 TMRn 寄存器的 CKSn 位选择，因此会产生和工作时钟一致的误差。

TCRn 在捕捉模式下作为向上计数器操作。

当通道启动触发器 (TSn) 设置为 1 时，TCRn 从 0000H 开始与计数时钟同步向上计数。

当 TIn 引脚检测到输入的有效沿时，计数值被发送 (捕捉) 到 TDRn，同时计数器 (TCRn) 被清除为 0000H，并且产生 INTTMn。如果计数器在此时溢出，TSRn 寄存器的 OVF 位设置为 1。如果计数器没有溢出，OVF 位清零。之后，重复上面的操作。

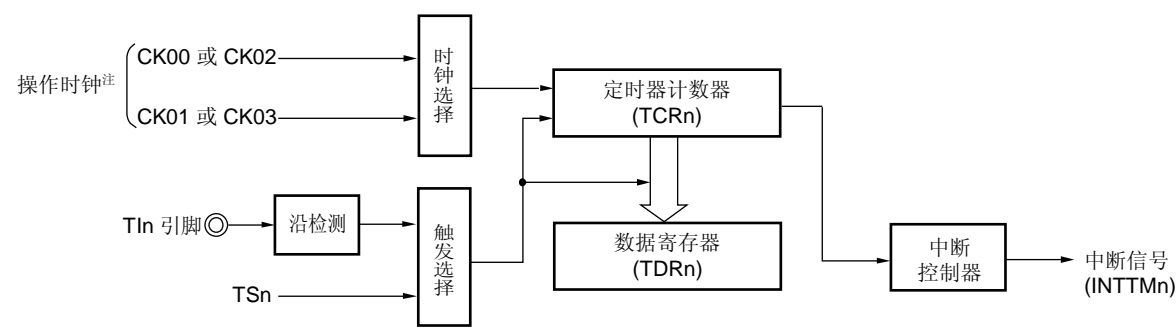
计数值一旦被捕捉到 TDRn 寄存器，TSRn 寄存器的 OVF 位就根据测量期间计数器是否溢出而更新。因此，可以检测到捕捉值的溢出状态。

如果计数器持续满计数达到两个或两个周期以上，则判断为一次溢出发生，TSRn 寄存器的 OVF 位设置为 1。但是 OVF 位作为累计标志，如果溢出发生一次以上，则无法测量到正确的间隔值。

设置 TMRn 寄存器的 STSn2~STSn0 位为 001B，将 TIn 的有效沿作为启动触发和捕捉触发。

当 TEn = 1，软件操作 (TSn = 1) 可作为捕捉触发，而不是 TIn 引脚输入。

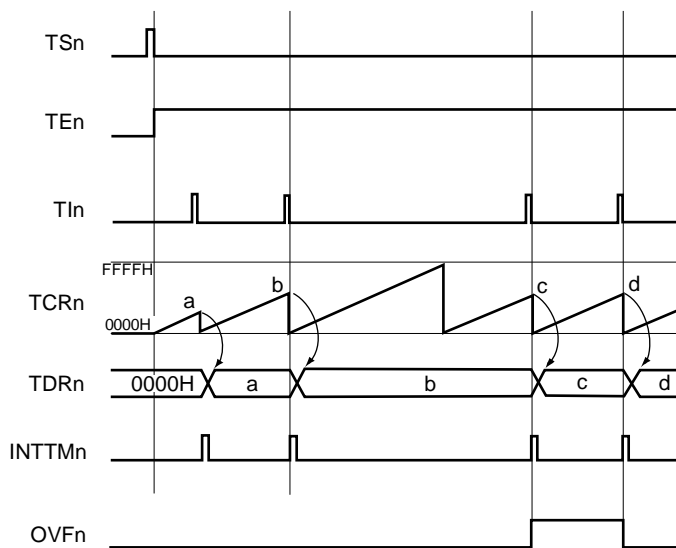
图 6-48. 作为输入脉冲间隔测量操作的框图



注 通道 0~7 的工作时钟从 CK00 和 CK01 中选择，通道 8~11 的工作时钟从 CK02 和 CK03 中选择。

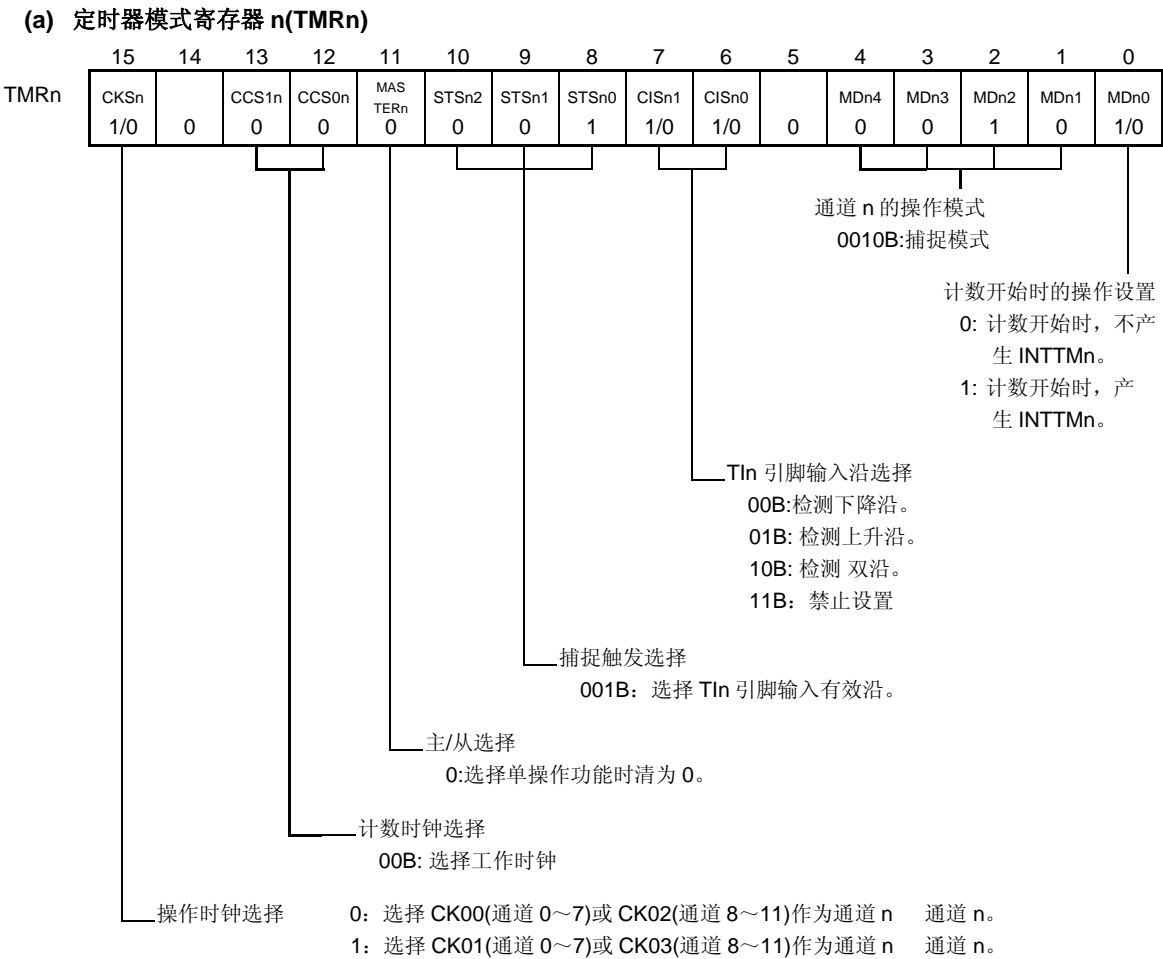
备注 n = 00~11

图 6-49. 作为输入脉冲间隔测量 (MDn0 = 0) 操作的基本时序示例



备注 n = 00~11

图 6-50. 测量输入脉冲间隔的寄存器设置内容的示例



- (b) 定时器输出寄存器 0(TO0)
- 第 n 位
- TO0
- | |
|-----|
| TOn |
| 0 |
- 0: 从 TOn 输出 0。
- (c) 定时器输出使能寄存器 0(TOE0)
- 第 n 位
- TOE0
- | |
|------|
| TOEn |
| 0 |
- 0: 通过计数操作, 停止 TOn 输出操作。
- (d) 定时器输出电平寄存器 0(TOL0)
- 第 n 位
- TOL0
- | |
|------|
| TOLn |
| 0 |
- 0: 当 TOMn = 0 (反转模式) 时清零。
- (e) 定时器输出模式寄存器 0(TOM0)
- 第 n 位
- TOM0
- | |
|------|
| TOMn |
| 0 |
- 0: 设置反转模式。

备注 n = 00~11

图 6-51. 当使用输入脉冲间隔测量功能时的操作过程

继续操作。

	软件操作	硬件状态
TAUS 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER2 寄存器的 TAU0EN 位为 1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
	设置 TPS0 寄存器。 把 CK00 和 CK01 的时钟频率用于通道 0~7, 把 CK02 和 CK03 的时钟频率用于通道 8~11。	
通道默认设置	设置 TMRn 寄存器 (决定通道的操作模式)。	通道停止操作。 (供给时钟并且产生一些功耗。)
操作开始	设置 TSn 位为 1。 由于是触发位, TSn 位自动恢复为 0。	TEn=1,并且计数操作开始。 当计数时钟输入时 TCRn 被清为 0000H。 当 TMRn 寄存器的 MDn0 位为 1 时, 产生 INTTMn。
操作期间	只有 TMRn 寄存器的 CISn1 和 CISn0 的设置值可以改变。 TDRn 寄存器一直可读。 TCRn 寄存器总是可读。 TSRn 寄存器一直可读。 TOM0、TOL0、TO0 和 TOE0 寄存器的设置值不能改变。	计数器 (TCRn) 从 0000H 开始向上计数。当检测到 Tin 引脚输入有效沿时, 计数值发送 (捕捉) 到 TDRn。同时, TCRn 被清除为 0000H 且产生 INTTMn 信号。 如果计数器在此时溢出, TSRn 寄存器的 OVF 位设置为 1。如果计数器没有溢出, OVF 位清零。 之后, 重复上面的操作。
操作停止	TTn 位设置为 1。 由于是触发位, TTn 位自动恢复为 0。	TEn=0, 并且计数操作停止。 TCRn 保持计数值并停止。 TSRn 寄存器的 OVF 位也被保持。
TAUS 停止	PER2 寄存器的 TAU0EN 位清零。	掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。

备注 n = 00~11

6.7.5 用作输入信号高/ 低电平宽度测量的操作

在 TIn 的一个沿开始计数，并且在另一个沿捕捉计数的数量，可以测量 TIn 的信号宽度 (高电平宽度/低电平宽度)。TIn 的信号宽度可以由下面的公式计算出。

$$\text{TIn 输入的信号宽度} = \text{计数时钟周期} \times ((10000\text{H} \times \text{TSRn: OVF}) + (\text{TDRn 的捕捉值} + 1))$$

注意事项 对 TIn 引脚输入进行采样的工作时钟是由 TMRn 寄存器的 CKSn 位选择，因此会产生和工作时钟一致的误差。

TCRn 在捕捉&一次计数模式下作为向上计数器。

当通道启动触发器 (TSn) 设置为 1，TEn 设置为 1 并且设置 TIn 引脚为开始沿检测等待状态。

当检测到 TIn 开始有效沿 (当测量高电平宽度时为 TIn 的上升沿)，计数器与计数时钟同步向上计数。之后当检测到有效捕捉沿 (当测量高电平宽度时 TIn 的下降沿)，计数值发送到 TDRn，同时，产生 INTTMn。如果计数器在此时溢出，TSRn 寄存器的 OVF 位设置为 1。如果计数器没有溢出，OVF 位清零。TCRn 在“传输至 TDRn+1”处停止，并且设置 TIn 引脚起始沿检测等待状态。之后，重复上面的操作。

计数值一旦被捕捉到 TDRn 寄存器，TSRn 寄存器的 OVF 位就根据测量期间计数器是否溢出而更新。因此，可以检测到捕捉值的溢出状态。

如果计数器持续满计数达到两个或两个周期以上，则判断为一次溢出发生，TSRn 寄存器的 OVF 位设置为 1。但是 OVF 位作为累计标志，如果溢出发生一次以上，则无法测量到正确的间隔值。

通过 TMRn 寄存器的 CISn1 和 CISn0 位可以选择测量 TIn 引脚的高电平宽度或低电平宽度。

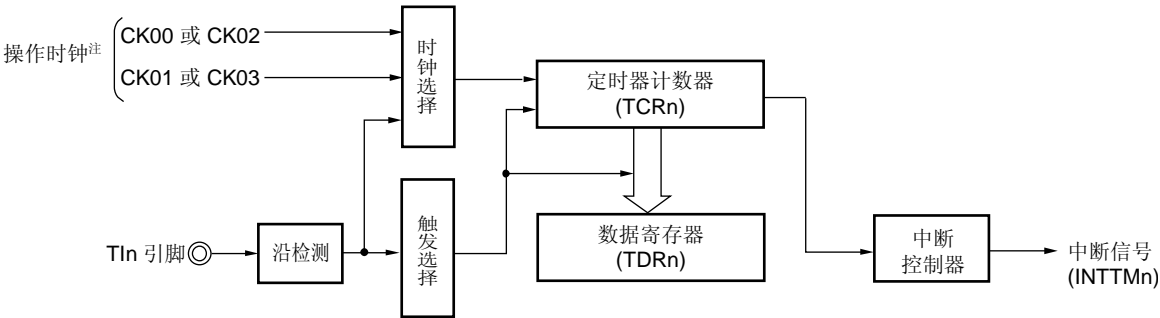
因为该功能用于测量 TIn 引脚输入的信号宽度，当 TEn 为 1 时 TSn 不能设置为 1。

TMRn 的 CISn1、CISn0 = 10B: 测量低电平宽度。

TMRn 的 CISn1、CISn0 = 11B: 测量高电平宽度。

备注 n = 00~11

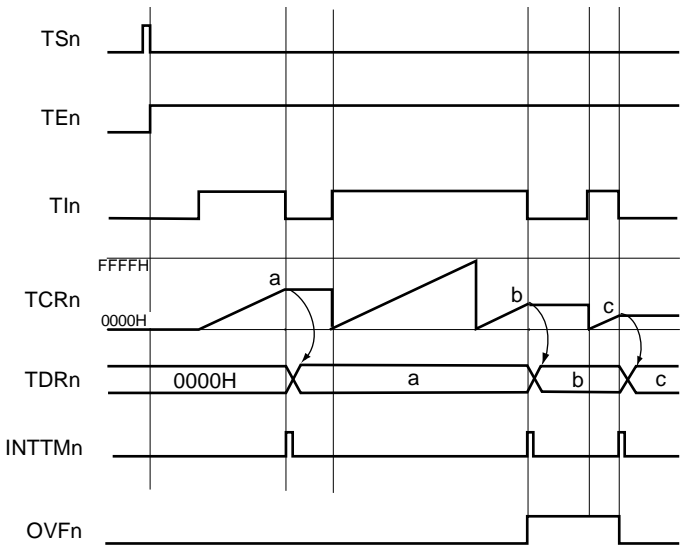
图 6-52.作为输入信号高/ 低电平宽度测量操作的框图



注 通道 0~7 的工作时钟从 CK00 和 CK01 中选择，通道 8~11 的工作时钟从 CK02 和 CK03 中选择。

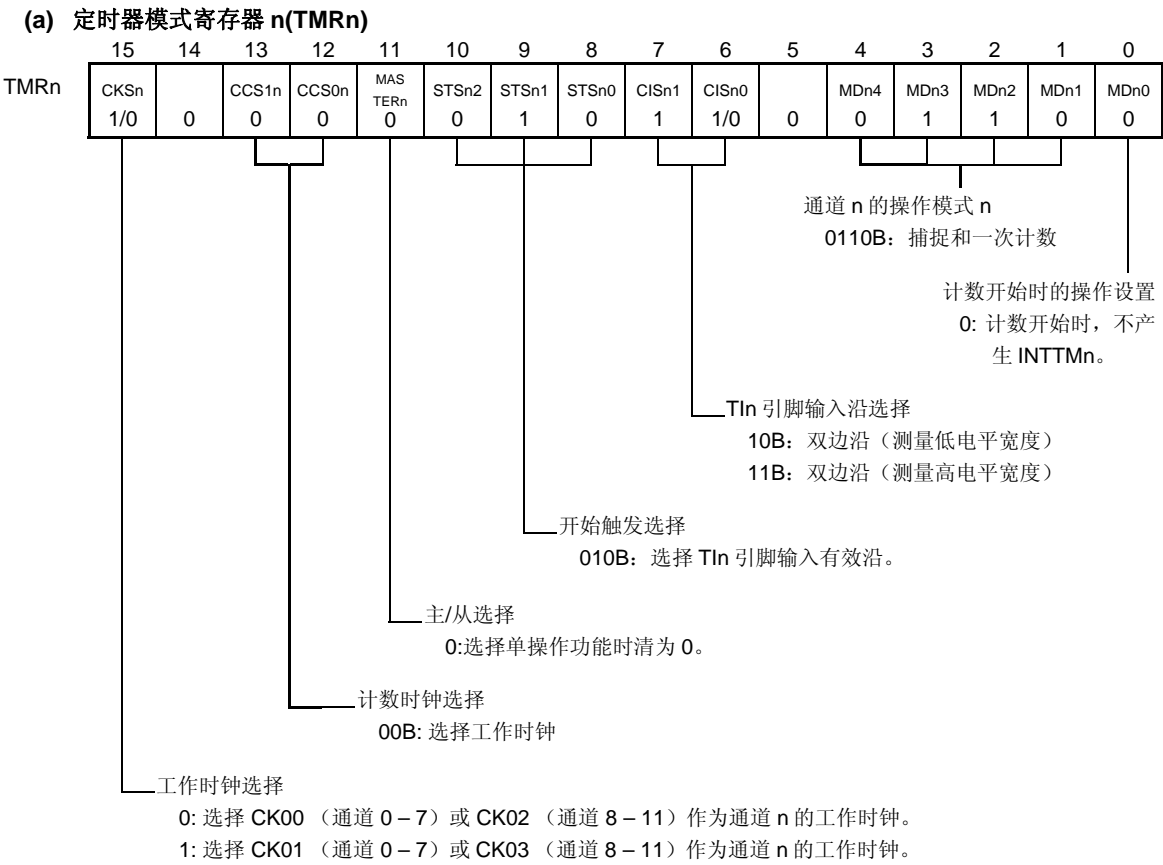
备注 n = 00~11

图 6-53. 作为输入信号高/ 低电平宽度测量操作的基本时序示例

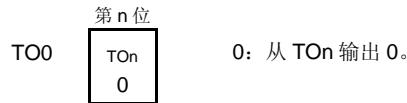


备注 n = 00~11

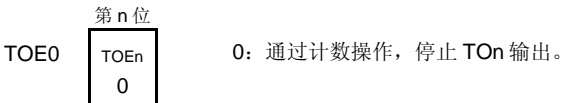
图 6-54. 测量输入信号高/ 低电平宽度的寄存器设置内容的示例



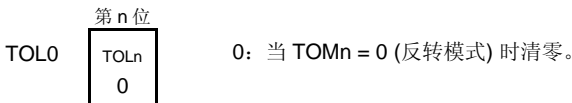
(b) 定时器输出寄存器 0(TO0)



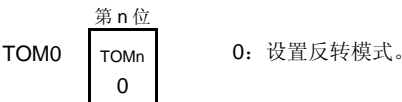
(c) 定时器输出使能寄存器 0(TOE0)



(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 = 00~11

图 6-55. 使用输入信号高/低电平宽度测量功能的操作过程

	软件操作	硬件状态
继续操作。	TAUS 默认设置	掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER2 寄存器的 TAU0EN 位为 1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
	设置 TPS0 寄存器。 把 CK00 和 CK01 的时钟频率用于通道 0~7, 把 CK02 和 CK03 的时钟频率用于通道 8~11。	
	通道默认设置 设置 TMRn 寄存器 (决定通道的操作模式)。 把 TOEn 清零并停止 TOn 的运行。	通道停止操作。 (供给时钟并且产生一些功耗。)
	操作开始 设置 TSn 位为 1。 由于是触发位, TSn 位自动恢复为 0。	TEn = 1, 设置 TIn 引脚开始边沿检测等待状态。
	操作期间 TDRn 寄存器的设置值不能改变。 TCRn 寄存器总是可读。 不使用 TSRn 寄存器。 TMRn、TOM0、TOL0、TO0 和 TOE0 寄存器的设置值不能改变。	检测 TIn 引脚输入计数开始有效沿。 TCRn 清除为 0000H 并且开始向上计数。 当 TIn 引脚启动沿被检测到时, 计数器 (TCRn) 从 0000H 开始向上计数。如果 TIn 引脚的捕捉沿被检测到, 计数值就会传输到 TDRn 并且产生 INTTMn。 如果计数器在此时溢出, TSRn 寄存器的 OVF 位设置为 1。如果计数器没有溢出, OVF 位清零。TCRn 停止计数操作直到下一个 TIn 引脚启动沿被检测到。 之后, 重复上面的操作。
	操作停止 TTn 位设置为 1。 由于是触发位, TTn 位自动恢复为 0。	TEn=0, 并且计数操作停止。 TCRn 保持计数值并停止。 TSRn 寄存器的 OVF 位也被保持。
	TAUS 停止 PER2 寄存器的 TAU0EN 位清零。	掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。

备注 n = 00~11

6.8 定时器阵列单元TAUS的多通道操作

6.8.1 作为PWM功能操作

两个通道可用于设置产生任意周期和占空比的脉冲。
输出脉冲的周期和占空比可通过下面的公式计算出。

脉冲周期 = { TDRn (主通道) 的设置值 + 1} × 计数时钟周期

占空比 [%] = { TDRm (从设备) 的设置值}/{ TDRn (主设备) 的设置值 + 1} × 100

0% 输出: TDRm(从设备)的设置值=0000H

100%输出: TDRm(从设备) 的设置值≥{TDRn (主设备) 的设置值+ 1}

主通道操作在间隔定时器模式并计数周期。当通道启动触发器 (TSn) 设置为 1 时，INTTMn 输出。TCRn 从 TDRn 加载到的值开始与计数时钟同步向下计数。当 TCRn=0000H 时，将输出 INTTMn。TCRn 再次加载 TDRn 的值。之后，重复类似的操作。

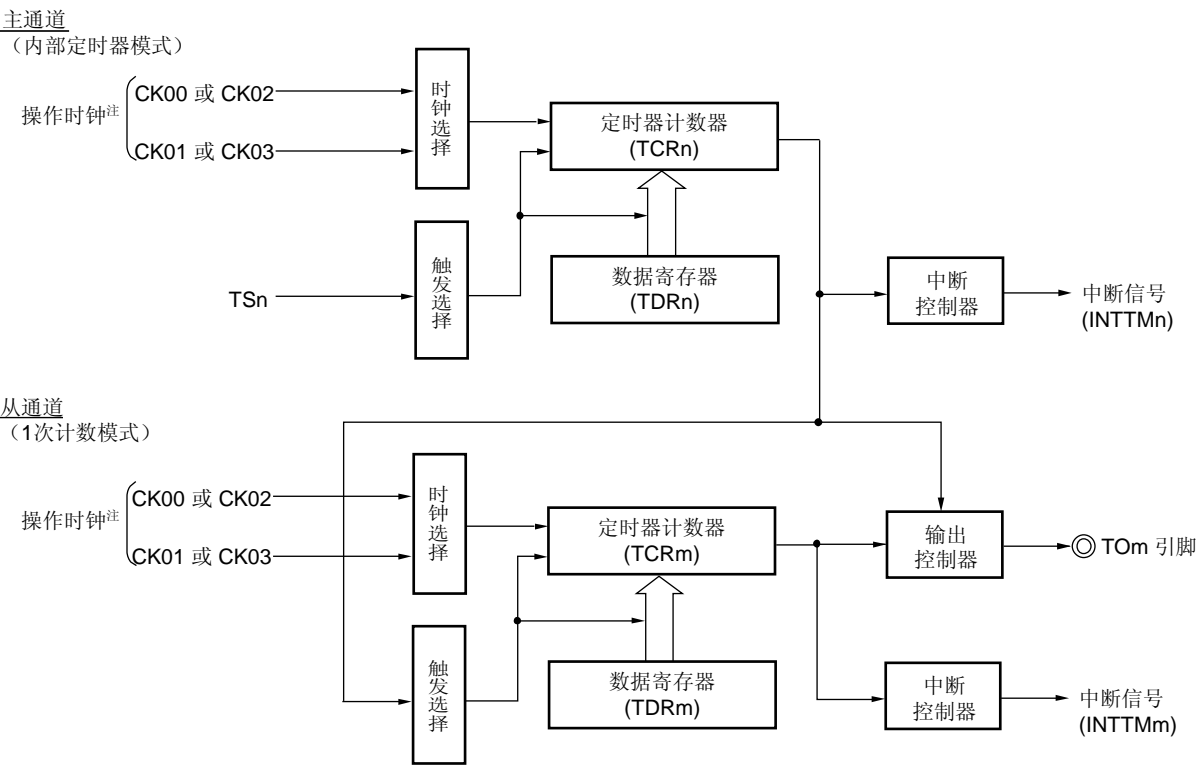
从通道的 TCRm 运行于一次计数模式，计数占空比，并且从 TOn 引脚输出 PWM 波形。从通道的 TCRm 加载 TDRm 的值，通过主通道的 INTTMn 作为启动触发，并且直到下一个启动触发 (主通道的 INTTMn) 输入时停止计数。

TOn 的输出电平在主通道产生 INTTMn 后的一个计数时钟后开始有效，TCRm = 0000H 时变为无效。

注意事项 要重写主通道的 TDRn 和从通道的 TDRm，需要两次写入访问。在主通道的 INTTMn 发生时，TDRn 和 TDRm 的值加载到 TCRn 和 TCRm。因此，当两次重写分别在主通道的 INTTMn 发生之前和之后执行时，TOn 引脚不能输出预期的波形。要重写主通道的 TDRn 和从通道的 TDRm，必须在主通道的 INTTMn 产生后立即重写这两个寄存器。

备注 n = 00、02、04、06、08、10
 m = n + 1

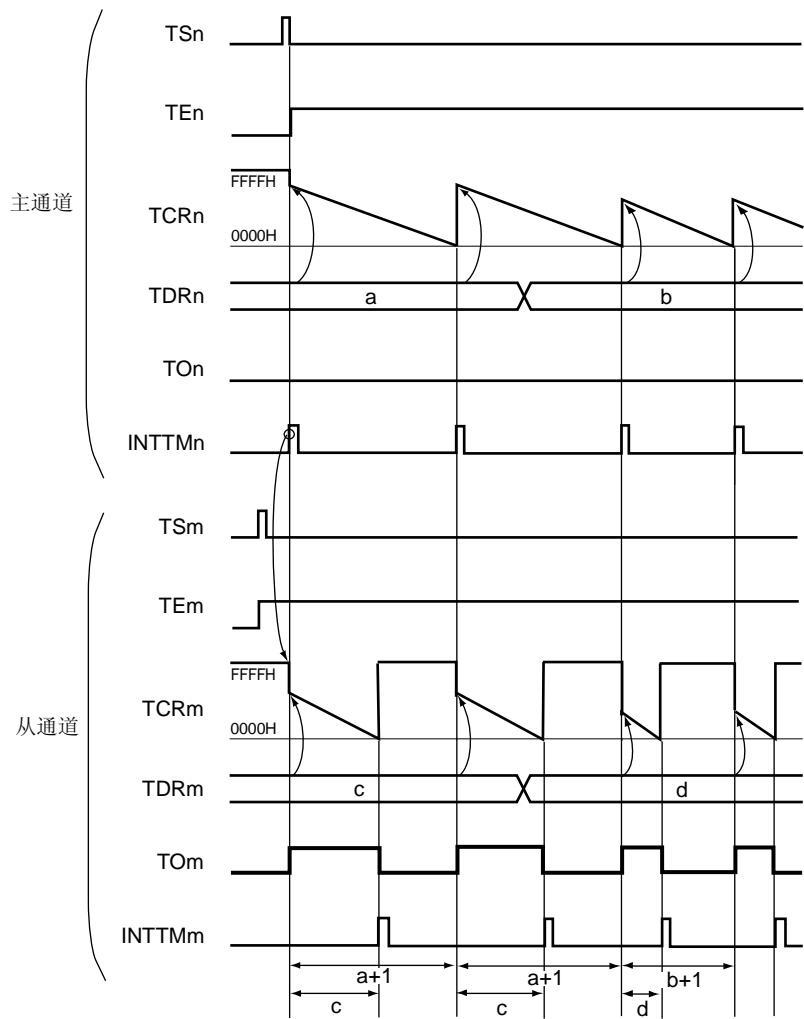
图 6-56. 作为 PWM 功能操作的框图



注 通道 0~7 的工作时钟从 CK00 和 CK01 中选择，通道 8~11 的工作时钟从 CK02 和 CK03 中选择。

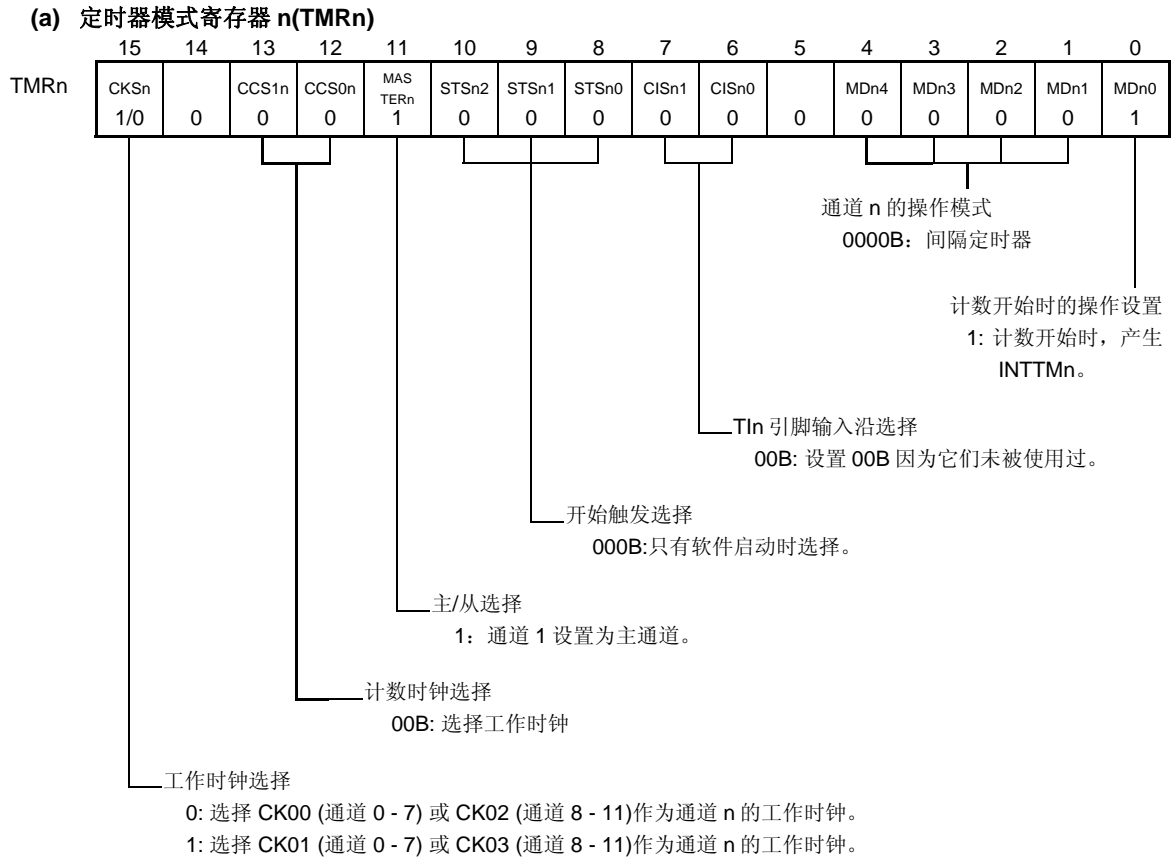
备注 n = 00、02、04、06、08、10
m = n + 1

图 6-57. 作为 PWM 功能操作的基本时序示例

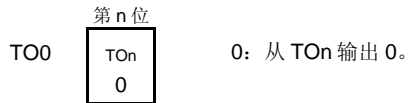


备注 $n = 00, 02, 04, 06, 08, 10$
 $m = n + 1$

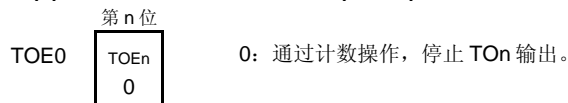
图 6-58. 使用 PWM 功能(主通道)时寄存器设置内容的示例



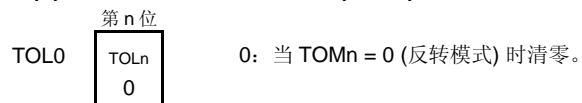
(b) 定时器输出寄存器 0(TO0)



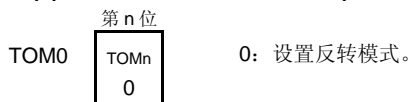
(c) 定时器输出使能寄存器 0(TOE0)



(d) 定时器输出电平寄存器 0(TOL0)

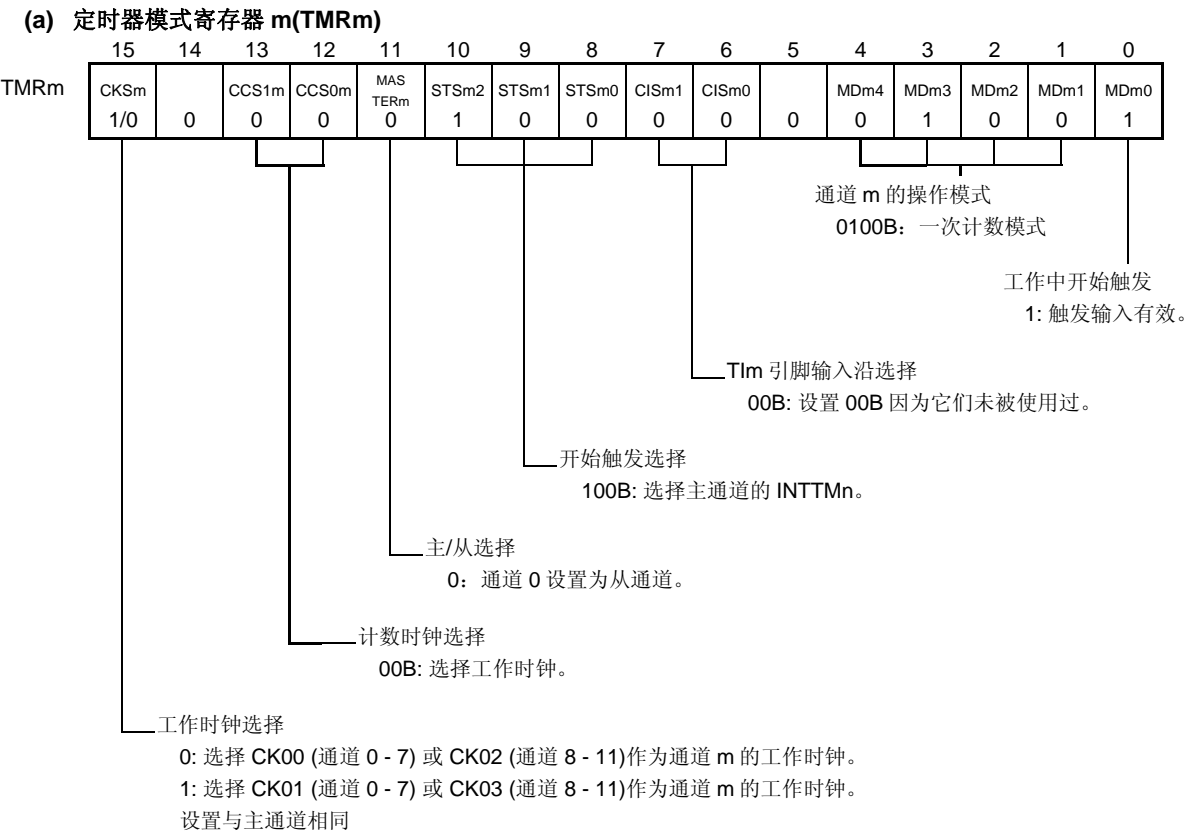


(e) 定时器输出模式寄存器 0(TOM0)



备注 n = 00、02、04、06、08、10

图 6-59. 当作为 PWM 功能(从通道)使用时寄存器设置的示例



(b) 定时器输出寄存器 0(TO0)

第 m 位	
TO0	<div><div>TOm</div><div>1/0</div></div>
	0: 从 TOm 输出 0。 1: 从 TOm 输出 1。

(c) 定时器输出使能寄存器 0(TOE0)

第 m 位	
TOE0	<div><div>TOEm</div><div>1/0</div></div>
	0: 通过计数操作, 停止 TOm 输出操作。 1: 通过计数操作, 允许 TOm 输出操作。

(d) 定时器输出电平寄存器 0(TOL0)

第 m 位	
TOL0	<div><div>TOLm</div><div>1/0</div></div>
	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)

(e) 定时器输出模式寄存器 0(TOM0)

第 m 位	
TOM0	<div><div>TOMm</div><div>1</div></div>
	1: 设置组合操作模式。

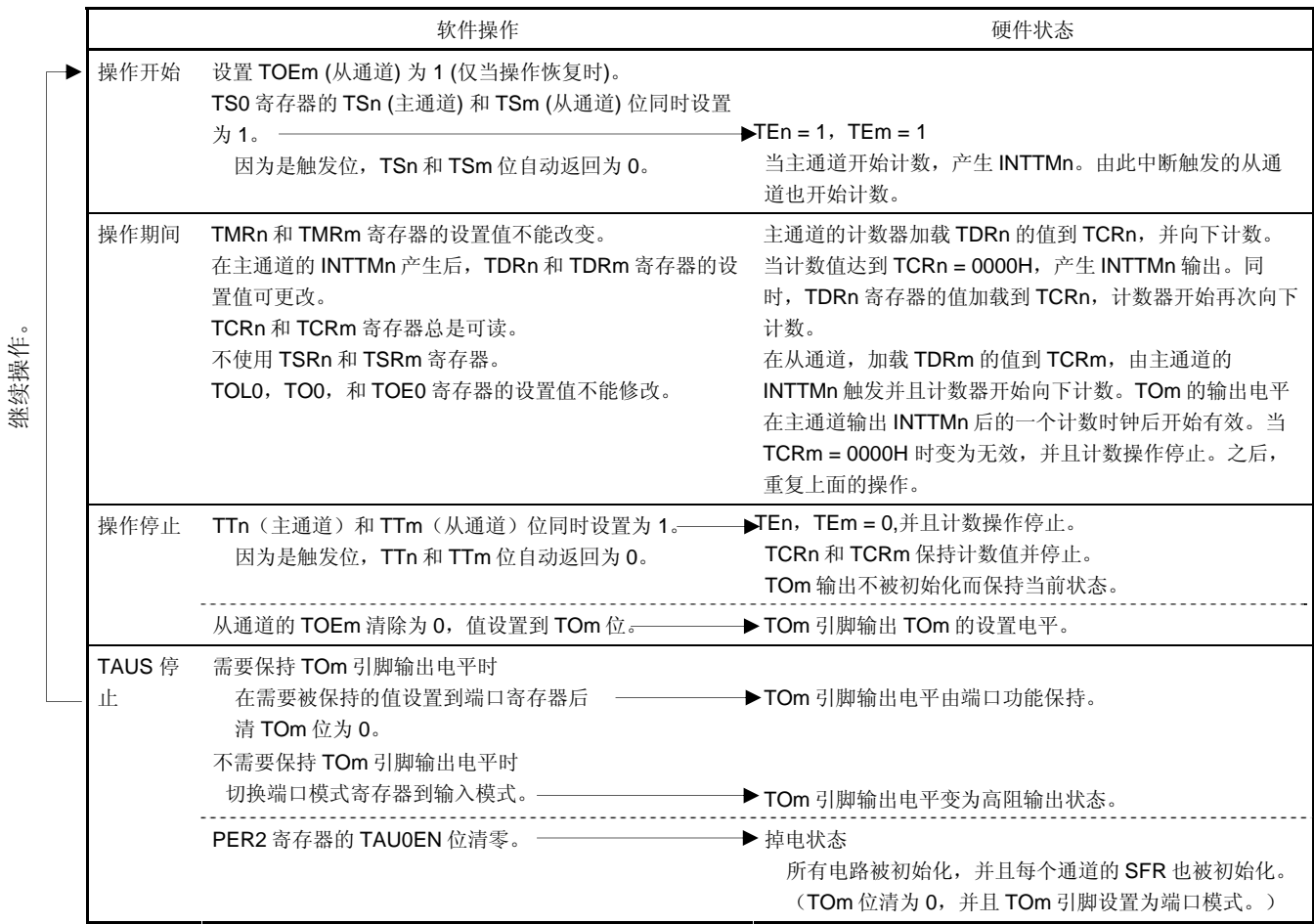
备注 n = 00、02、04、06、08、10
 m = n + 1

图 6-60. 使用 PWM 功能时的操作过程 (1/2)

软件操作	硬件状态
<p>TAUS 默认设置</p> <p>设置 PER2 寄存器的 TAU0EN 位为 1。</p> <p>设置 TPS0 寄存器。 把 CK00 和 CK01 的时钟频率用于通道 0~7，把 CK02 和 CK03 的时钟频率用于通道 8~11。</p>	<p>掉电状态 (停止供给时钟并且所有寄存器禁止写入。)</p> <p>上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)</p>
<p>通道默认设置</p> <p>设置要使用的两个通道的 TMRn 和 TMRm 寄存器 (决定通道的操作模式)。 间隔 (周期) 值设置到主通道的 TDRn 寄存器, 占空比设置到从通道的 TDRm 寄存器。</p> <p>设置从通道。 TOM0 寄存器的 TOMm 位设置为 1 (组合操作模式)。 设置 TOLm 位。 设置 TOm 位并决定 TOm 输出的默认电平。</p> <p>设置 TOEm 至 1 并允许 TOM 操作。 将端口寄存器和端口模式寄存器清为 0。</p>	<p>通道停止操作。 (供给时钟并且产生一些功耗。)</p> <p>TOm 引脚为高阻输出状态。</p> <p>当端口模式寄存器在输出模式下并且端口寄存器为 0 时, 输出 TOM 默认设置电平。</p> <p>因为通道停止工作, TOM 不会改变。</p> <p>TOm 引脚输出 TOM 的设置电平。</p>

备注 n = 00、02、04、06、08、10
 m = n + 1

图 6-60. 使用 PWM 功能时的操作过程 (2/2)



备注 n = 00、02、04、06、08、10
 m = n + 1

6.8.2 作为单脉冲输出功能操作

将两个通道作为一组，通过输入到 TIn 的引脚信号可以产生任意延迟脉冲宽度的单脉冲。

延迟时间和脉冲宽度可通过下面的公式计算出。

$$\begin{aligned}\text{延迟时间} &= \{\text{TDRn (主通道) 的设置值} + 2\} \times \text{计数时钟周期} \\ \text{脉冲宽度} &= \{\text{TDRm (从通道) 的设置值}\} \times \text{计数时钟周期}\end{aligned}$$

主通道运行于一次计数模式，计数延迟。主通道的 TCRn 在检测到启动触发时开始操作，TDRn 的值赋给 TCRn。TCRn 与计数时钟同步，由加载的 TDRn 值开始向下计数。当 TCRn = 0000H 时，输出 INTTMn 并且停止计数直到检测到下一个启动触发。

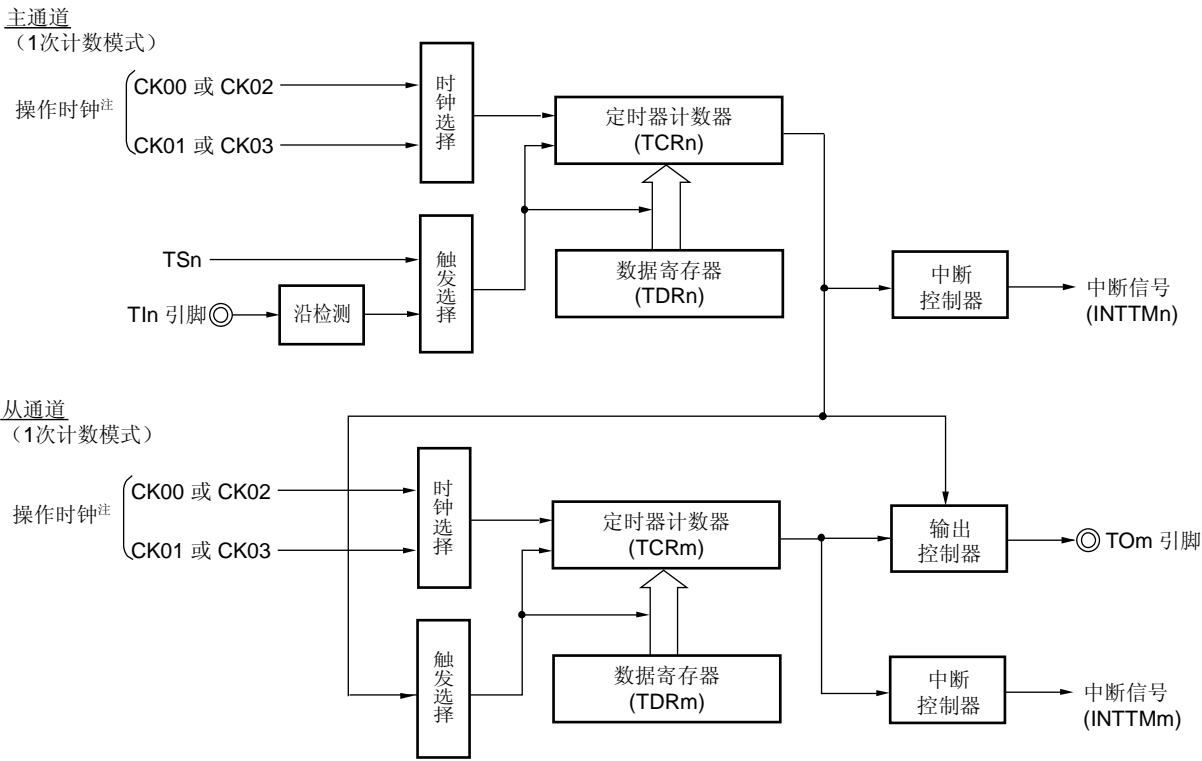
从通道操作在一次计数模式下，对脉冲宽度计数。通过主通道的 INTTMn 作为启动触发，从通道的 TCRm 开始计数，并加载 TDRm 值。TCRm 与计数值同步，由加载的 TDRm 值开始向下计数。当 TCRm = 0000H 时，输出 INTTMm 并停止计数直到检测到下一个启动触发（主通道的 INTTMn）。TOM 的输出电平在主通道产生 INTTMn 后的一个计数时钟后开始有效，TCRm = 0000H 时变为无效。

代替 TIn 引脚输入，单脉冲输出也可通过软件操作 (TSn = 1) 作为启动触发。

注意 主通道的 TDRn 的时序与从通道的 TDRm 的不同。如果在操作中 TDRn 和 TDRm 被重写，则输出不规则波形。必须在将要重写通道的 INTTMn 发生之后重写 TDRn 和 TDRm。

备注 n = 02、04、06、10
m = n + 1

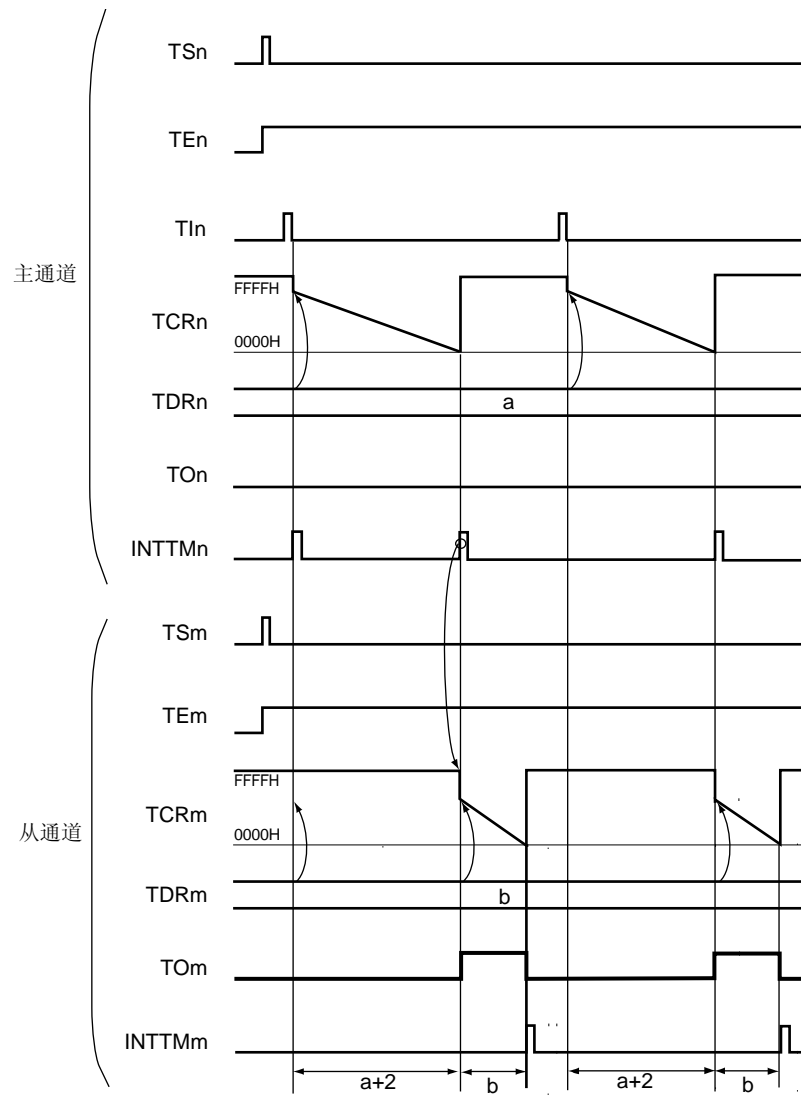
图 6-61. 作为单脉冲输出功能操作的框图



注 通道 0~7 的工作时钟从 CK00 和 CK01 中选择，通道 8~11 的工作时钟从 CK02 和 CK03 中选择。

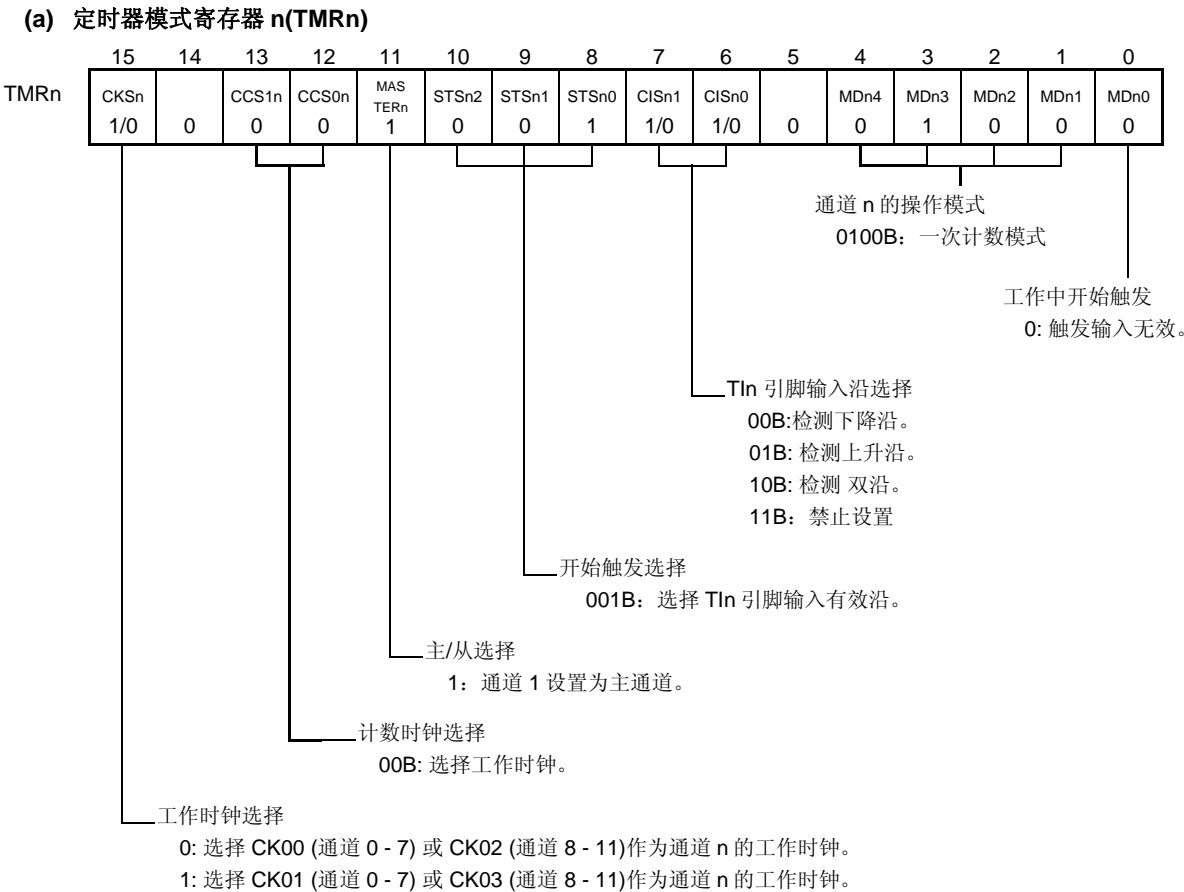
备注 n = 02、04、06、10
m = n + 1

图 6-62. 作为单脉冲输出功能操作的基本时序的示例

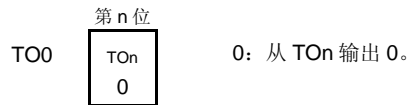


备注 $n = 02、04、06、10$
 $m = n + 1$

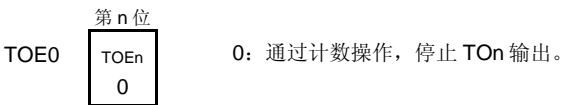
图 6-63. 当作为单脉冲输出功能 (主通道) 时寄存器设置的示例



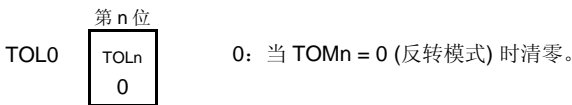
(b) 定时器输出寄存器 0(TO0)



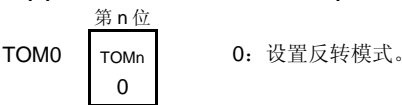
(c) 定时器输出使能寄存器 0(TOE0)



(d) 定时器输出电平寄存器 0(TOL0)

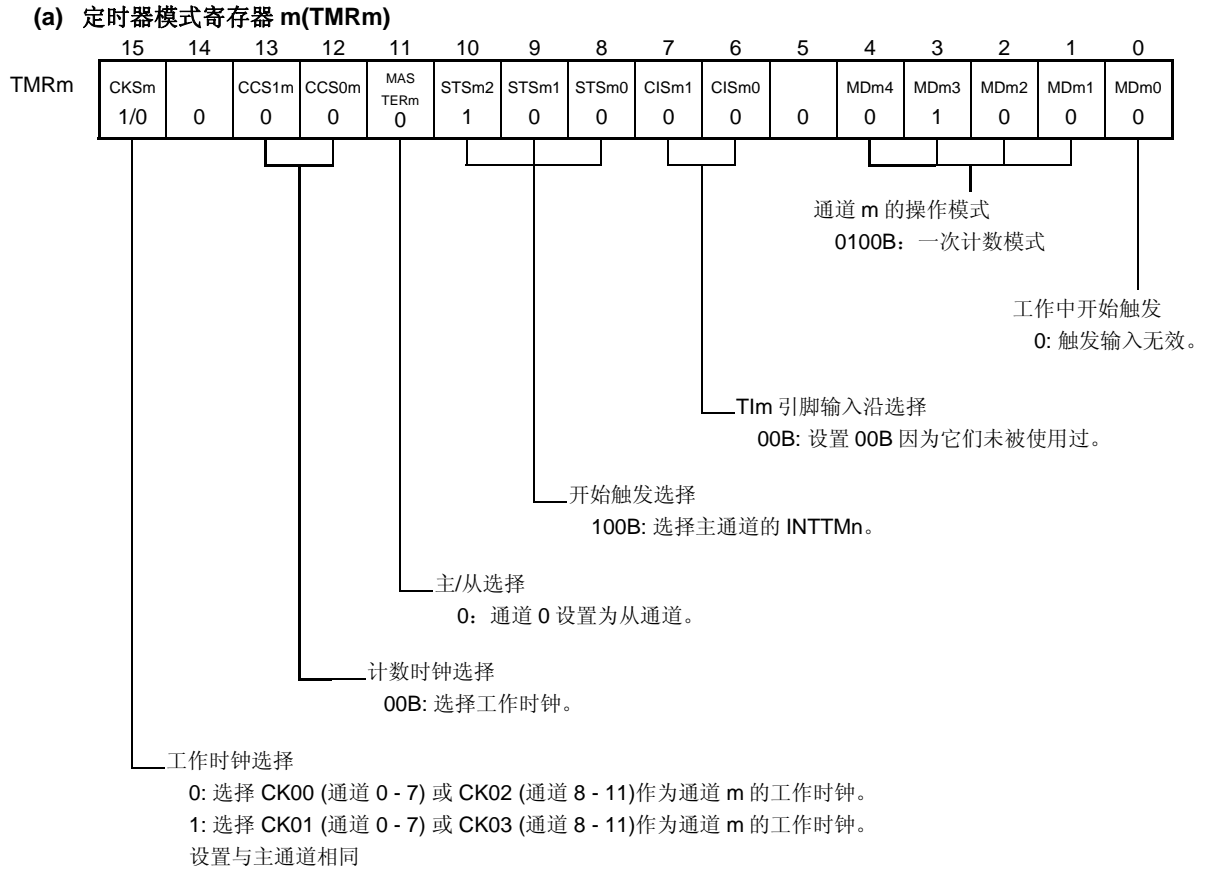


(e) 定时器输出模式寄存器 0(TOM0)



备注 n = 02、04、06、10

图 6-64. 当作为单脉冲输出功能(从通道)时寄存器设置的示例



(b) 定时器输出寄存器 0(TO0)

TO0	第 m 位	
	TOm	0: 从 TOm 输出 0。 1: 从 TOm 输出 1。
	1/0	

(c) 定时器输出使能寄存器 0(TOE0)

TOE0	第 m 位	
	TOEm	0: 通过计数操作, 停止 TOm 输出操作。 1: 通过计数操作, 允许 TOm 输出操作。
	1/0	

(d) 定时器输出电平寄存器 0(TOL0)

TOL0	第 m 位	
	TOLm	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)
	1/0	

(e) 定时器输出模式寄存器 0(TOM0)

TOM0	第 m 位	
	TOMm	1: 设置组合操作模式。
	1	

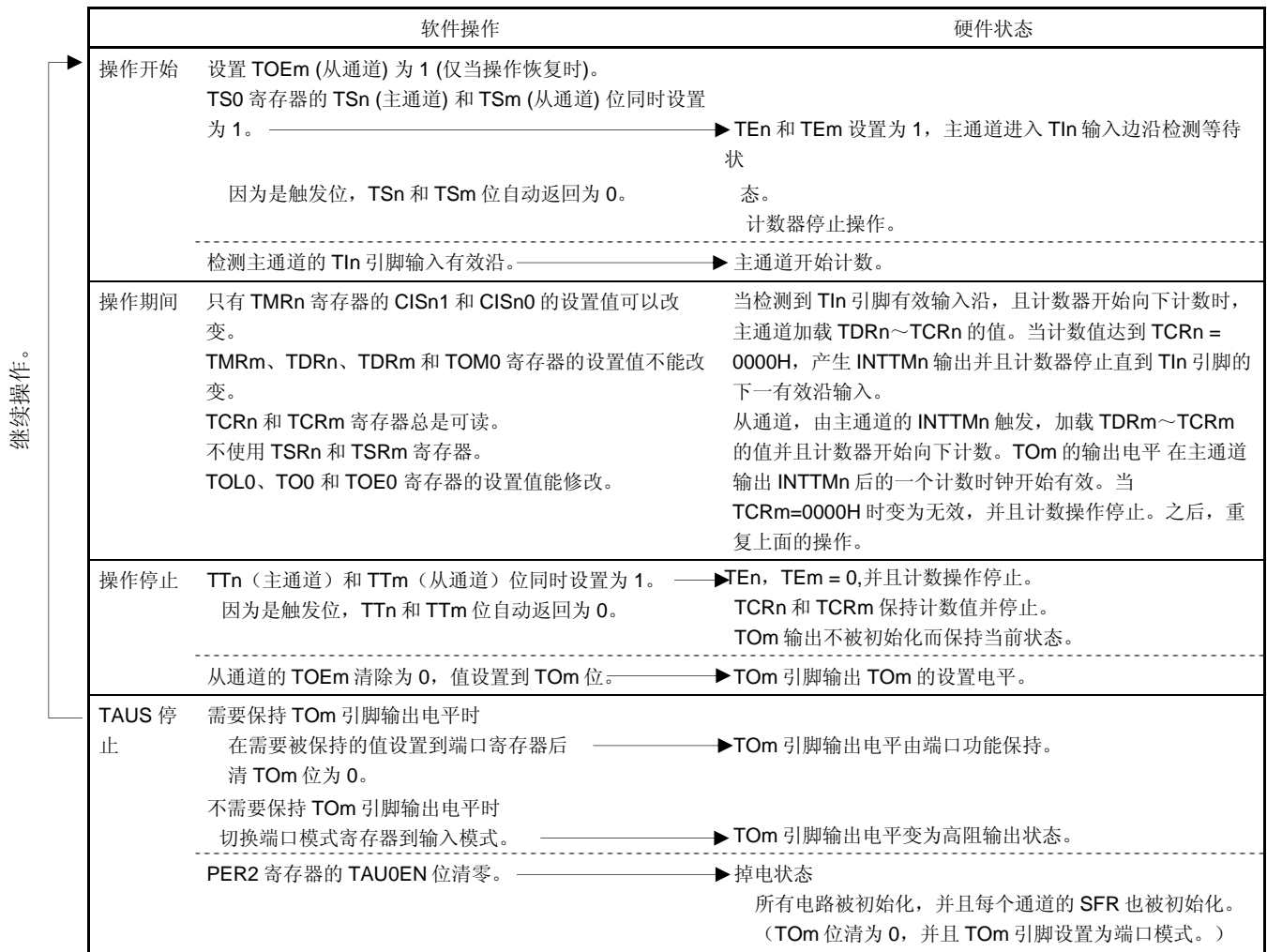
备注 n = 02、04、06、10
m = n + 1

图 6-65. 单脉冲输出功能的操作过程(1/2)

软件操作		硬件状态
TAUS 默认设置		掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 PER2 寄存器的 TAU0EN 位为 1。—————▶	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
	设置 TPS0 寄存器。 把 CK00 和 CK01 的时钟频率用于通道 0~7，把 CK02 和 CK03 的时钟频率用于通道 8~11。	
通道默认设置	设置要使用的两个通道的 TMRn 和 TMRm 寄存器（决定通道的操作模式）。 输出延迟设置到主通道的 TDRn 寄存器，脉冲宽度设置到从通道的 TDRm 寄存器。	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置从通道。 TOM0 寄存器的 TOMm 位设置为 1 (组合操作模式)。 设置 TOLm 位。	TOM 引脚为高阻输出状态。
	设置 TOM 位并决定 TOM 输出的默认电平。—————▶	当端口模式寄存器在输出模式下并且端口寄存器为 0 时，输出 TOM 默认设置电平。
	设置 TOEm 至 1 并允许 TOM 操作。—————▶	因为通道停止工作，TOM 不会改变。
	将端口寄存器和端口模式寄存器清为 0。 —————▶	TOM 引脚输出 TOM 的设置电平。

备注 n = 02、04、06、10
 m = n + 1

图 6-65. 单脉冲输出功能的操作过程(2/2)



备注 n = 02、04、06、10
 m = n + 1

6.8.3 作为多路PWM输出功能的操作

通过执行 PWM 功能和使用两个或两个以上通道，可产生多个 PWM 输出信号。
例如，当使用两个从通道时，输出脉冲的周期和占空比可通过下面的公式计算出。

$$\begin{aligned} \text{脉冲周期} &= \{ \text{TDRn (主通道) 的设置值} + 1 \} \times \text{计数时钟周期} \\ \text{占空比 1 [\%]} &= \{ \text{TDRp (从通道 1) 的设置值} \} / \{ \text{TDRn (主通道) 的设置值} + 1 \} \times 100 \\ \text{占空比 2 [\%]} &= \{ \text{TDRq (从通道 2) 的设置值} \} / \{ \text{TDRn (主通道) 的设置值} + 1 \} \times 100 \end{aligned}$$

备注 如果 TDRp (从通道 1)的设置值 > {(TDRn 的设置值 (主通道) + 1)}, 或者如果{TDRq (从通道 2)的设置值} > {(TDRn 的设置值 (主通道) + 1)}, 仍作为 100%输出。

主通道的 TCRn 运行于间隔定时器模式，计数周期。

从通道 1 的 TCRp 运行于一次计数模式，计数占空比，从 TOp 引脚输出 PWM 波形。TCRp 把 TDRp 的值加载到 TCRp 中，主通道的 INTTMn 作为启动触发，向下计数开始。当 TCRp = 0000H 时，TCRp 输出 INTTMp，并且停止计数直到下一个启动触发 (主通道的 INTTMn) 被输入。TOp 的输出电平在主通道产生 INTTMn 后的一个计数时钟后开始有效，TCRp = 0000H 时变为无效。

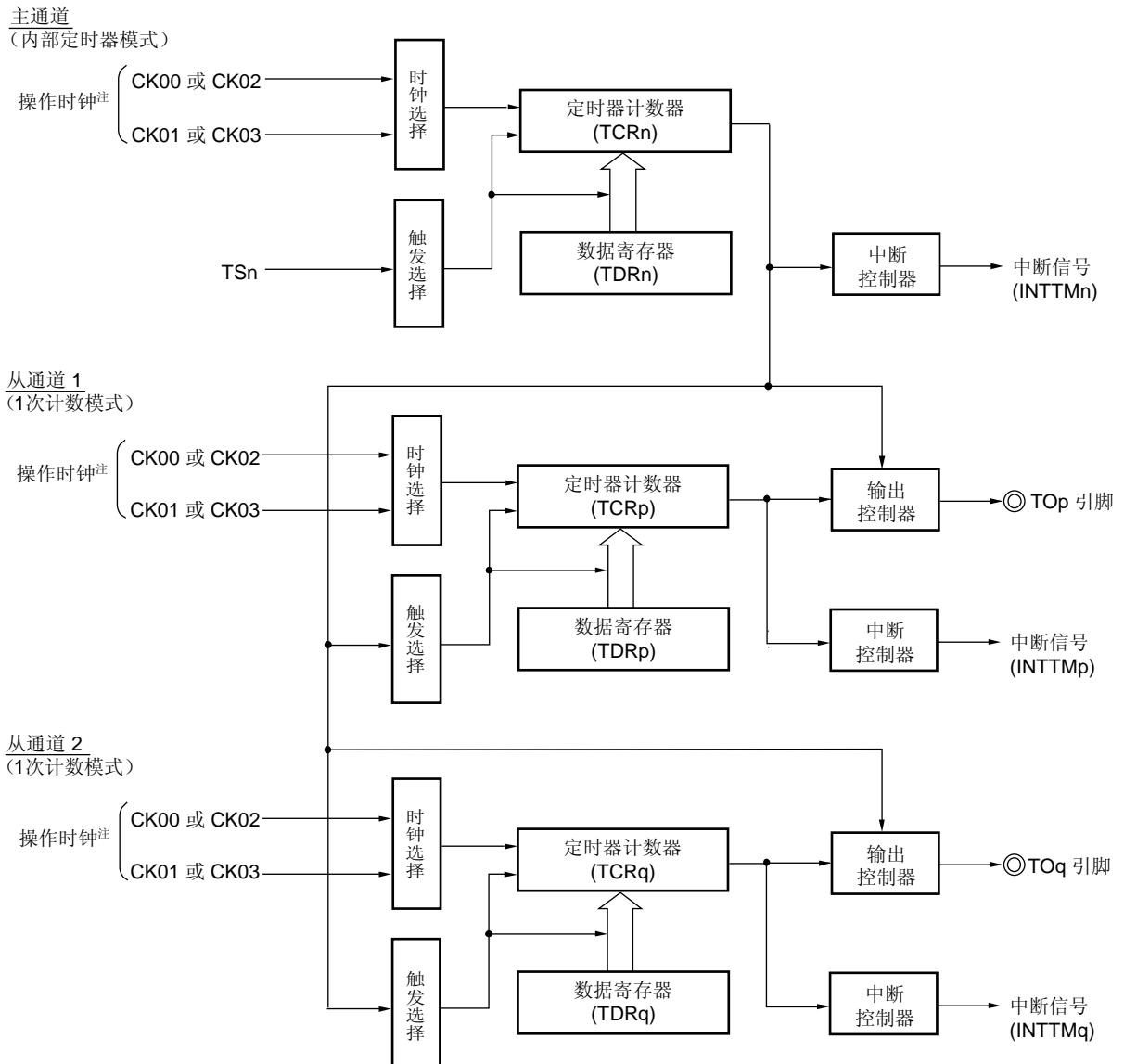
与从通道 1 的 TCRp 相同，从通道 2 的 TCRq 运行于一次计数模式，计数占空比，且从 TOq 引脚输出 PWM 波形。当 TCRq = 0000H 时，TCRq 输出 INTTMq，并且停止计数直到下一个启动触发 (主通道的 INTTMn) 被输入。TOq 的输出电平在主通道产生 INTTMn 后的一个计数时钟后开始有效，TCRq = 0000H 时变为无效。

<R> 当通道 0 用作上面的主通道，最多可同时输出九种类型 PWM 信号。

注意事项 要重写主通道的 TDRn 和从通道 1 的 TDRp，写访问必须至少执行两次。因为在主通道的 INTTMn 产生后 TDRn 和 TDRp 的值赋给 TCRn 和 TCRp，如果重写在主通道的 INTTMn 发生之前和之后分别执行，TOp 引脚可能无法输出预期波形。要重写主通道的 TDRn 和从通道的 TDRp，必须在主通道的 INTTMn 发生之后立即重写这两个寄存器 (这也适用于从通道 2 的 TDRq)。

备注 n = 00、02、04、06、08、10
 n < p < q ≤ 11
 然而，p 和 q 是连续整数。

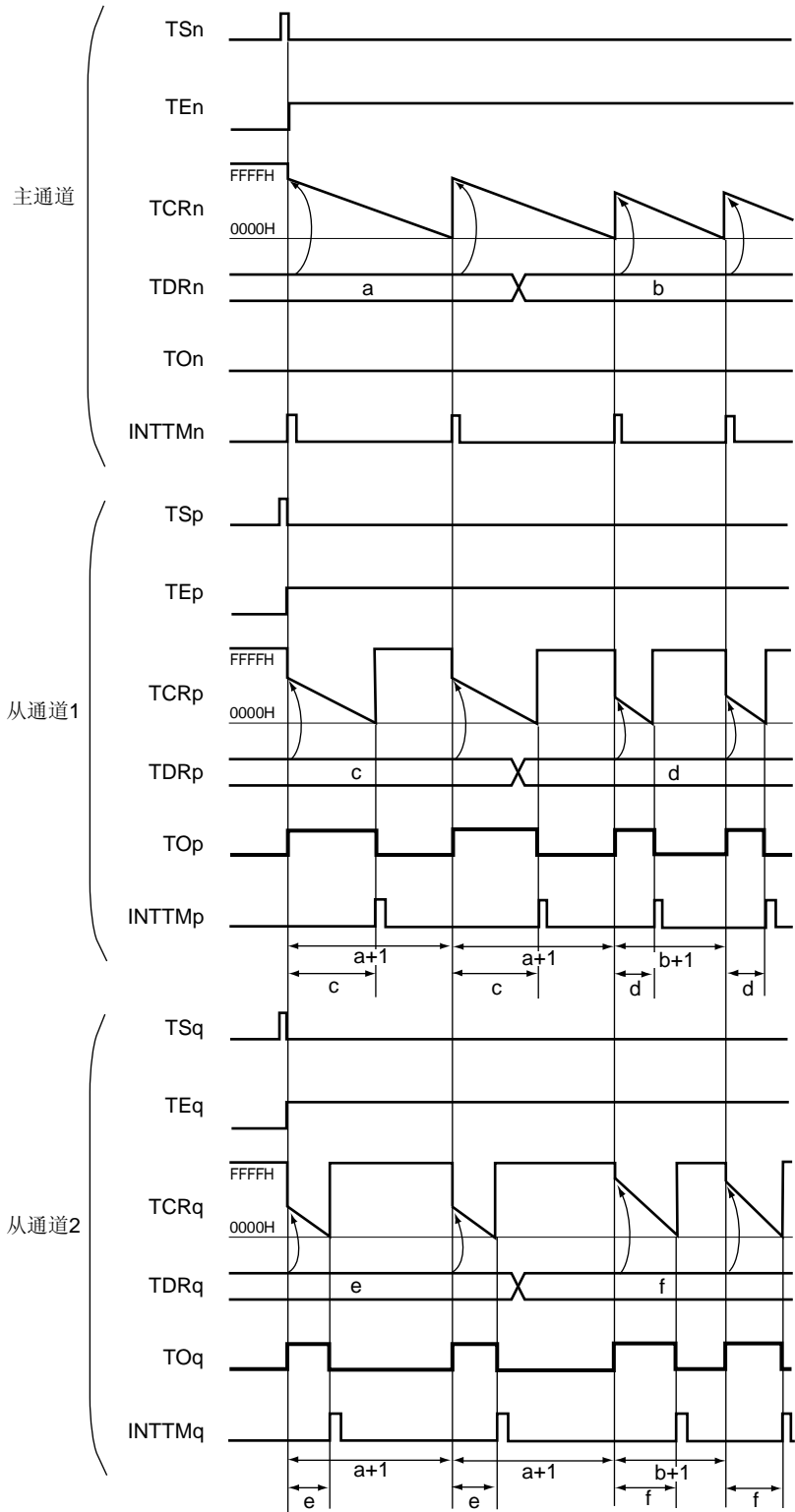
图 6-66. 作为多路 PWM 输出功能操作的框图 (输出两种类型 PWM)



注 通道 0~7 的工作时钟从 CK00 和 CK01 中选择，通道 8~11 的工作时钟从 CK02 和 CK03 中选择。

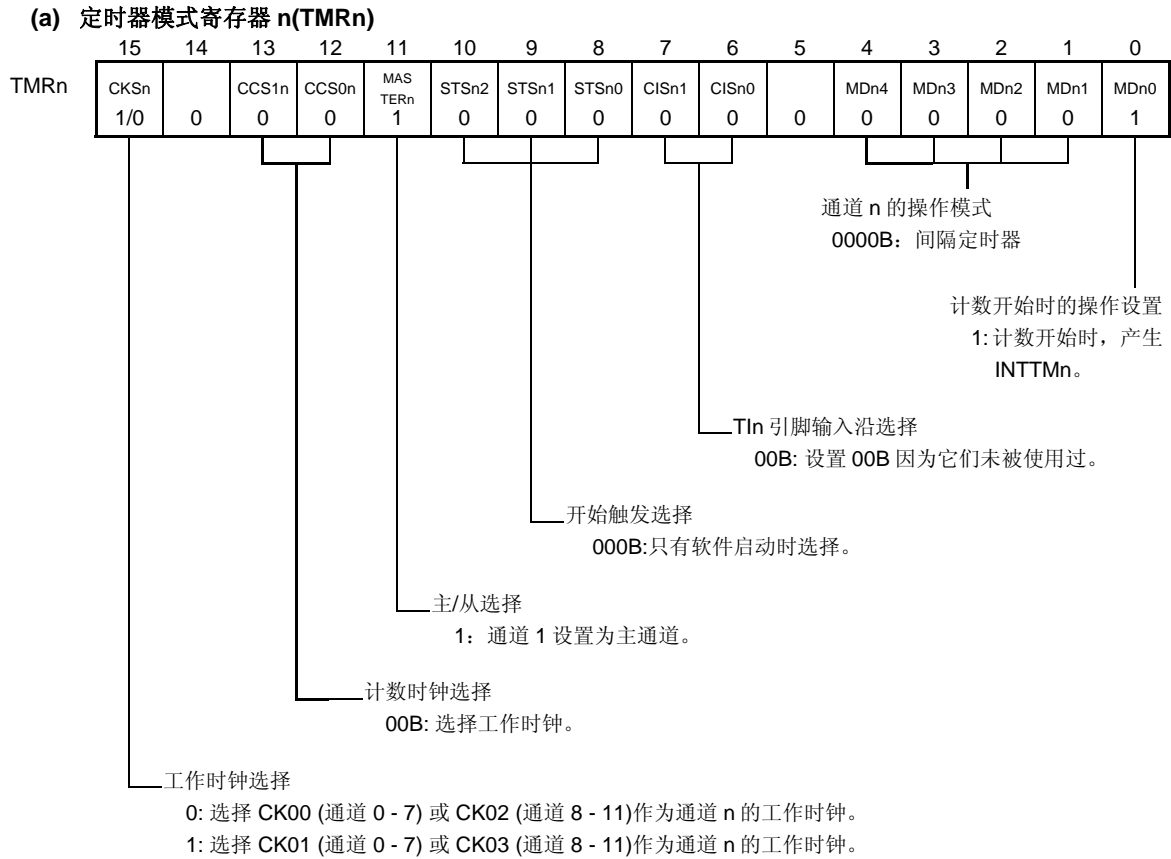
备注 n = 00、02、04、06、08、10
 $n < p < q \leq 11$
 然而，p 和 q 是连续整数。

图 6-67. 作为多路 PWM 输出功能 (输出两种类型 PWM) 操作的基本时序示例

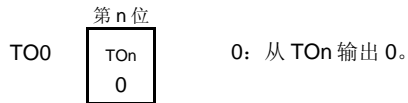


备注 $n = 00, 02, 04, 06, 08, 10$
 $n < p < q \leq 11$
 然而, p 和 q 是连续整数。

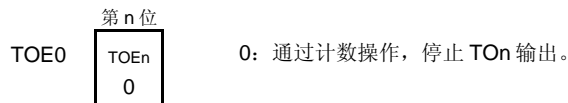
图 6-68. 当使用多路 PWM 输出功能(主通道)时寄存器设置内容的示例



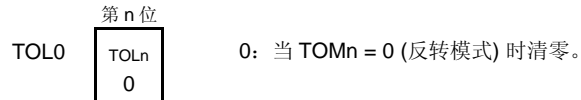
(b) 定时器输出寄存器 0(TO0)



(c) 定时器输出使能寄存器 0(TOE0)



(d) 定时器输出电平寄存器 0(TOL0)

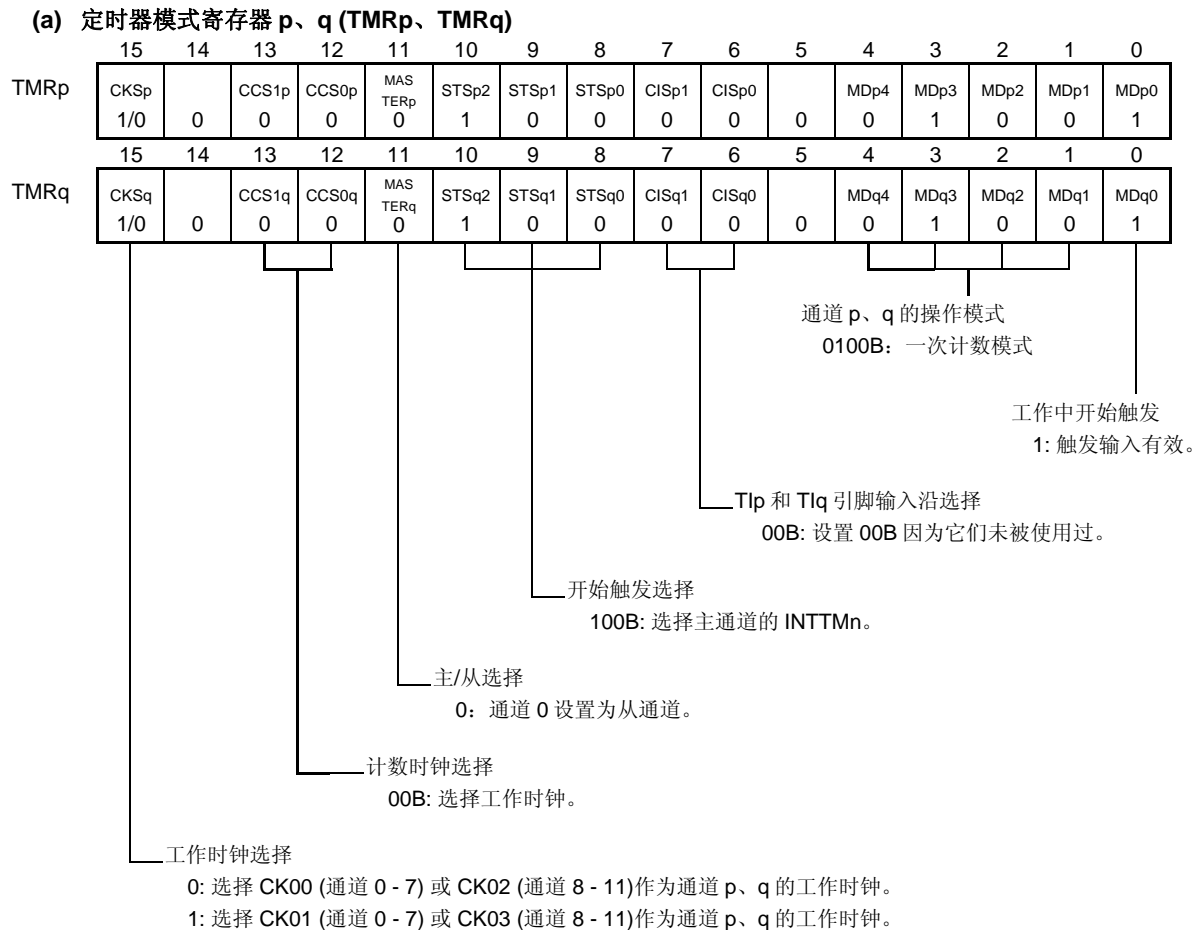


(e) 定时器输出模式寄存器 0(TOM0)



备注 n = 00、02、04、06、08、10

图 6-69. 作为多路 PWM 输出功能(从通道)时寄存器设置的示例(输出两种类型 PWMs)



(b) 定时器输出寄存器 0(TO0)

第 q 位 第 p 位		
TOq	TOp	0: 从 TOp 或 TOq 输出 0。
1/0	1/0	1: 从 TOp 或 TOq 输出 1。

(c) 定时器输出使能寄存器 0(TOE0)

第 q 位 第 p 位		
TOEq	TOEp	0: 通过计数操作, 停止 TOp 或 TOq 的输出操作。
1/0	1/0	1: 通过计数操作, 允许 TOp 或 TOq 的输出操作。

(d) 定时器输出电平寄存器 0(TOL0)

第 q 位 第 p 位		
TOLq	TOLp	0: 正向逻辑输出 (高有效)
1/0	1/0	1: 反向输出 (低有效)

(e) 定时器输出模式寄存器 0(TOM0)

第 q 位 第 p 位		
TOMq	TOMp	1: 设置组合操作模式。
1	1	

备注 n = 00、02、04、06、08、10 n < p < q ≤ 11 (然而, p 和 q 为连续整数。)

图 6-70. 当用作多路 PWM 输出功能时的操作过程 (输出两种类型 PWM) (1/2)

	软件操作	硬件状态
TAUS 默认设置	设置 PER2 寄存器的 TAU0EN 位为 1。	掉电状态 (停止供给时钟并且所有寄存器禁止写入。)
	设置 TPS0 寄存器。 把 CK00 和 CK01 的时钟频率用于通道 0~7, 把 CK02 和 CK03 的时钟频率用于通道 8~11。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
通道默认设置	设置要使用的每个通道的 TMRn、TMRp 和 TMRq 寄存器 (决定通道的操作模式)。 一个间隔 (周期) 值设置到主通道的 TDRn 寄存器, 占空比设置到从通道的 TDRp 和 TDRq 寄存器。	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置从通道。 把 TOM0 寄存器的 TOMp 和 TOMq 位设为 1 (组合操作模式)。 清除 TOLp 和 TOLq 位为 0。 设置 TOP 和 TOq 位, 并赋予 TOP 和 TOq 输出默认电平。	TOP 和 TOq 引脚达到高阻输出状态。
	设置 TOEp 或 TOEq 为 1 并允许 TOP 或 TOq 操作。	当端口模式寄存器在输出模式下并且端口寄存器为 0 时, 输出 TOP 和 TOq 默认设置电平。
	将端口寄存器和端口模式寄存器清为 0。	TOP 或 TOq 不会改变因为通道停止工作。 TOP 和 TOq 引脚输出 TOP 和 TOq 的设置值。
备注	n = 00、02、04、06、08、10 n < p < q ≤ 11 然而, p 和 q 是连续整数。	

图 6-70. 当用作多路 PWM 输出功能时的操作过程 (输出两种类型 PWM) (2/2)

	软件操作	硬件状态
继续操作。	操作开始 设置 TOEp 和 TOEq (从通道) 为 1 (仅当恢复操作时)。TS0 寄存器的 TSn (主通道), TSp 和 TSq (从通道) 位同时设置为 1。 因为是触发位, TSn、TSp 和 TSq 位自动返回为 0。	TEn = 1, TEp、TEq = 1 当主通道开始计数, 产生 INTTMn。由此中断触发的从通道也开始计数。
	操作期间 TMRn、TMRp、TMRq、TOM0 和 TOE0 寄存器的设置值不能改变。 在主通道的 INTTMn 产生后, TDRn、TDRp 和 TDRq 寄存器的设置值可更改。 TCRn、TCRp 和 TCRq 寄存器总是可读。 不使用 TSRn、TSRp 和 TSRq 寄存器。 TOM0, TOL0, TO0, 和 TOE0 寄存器的设置值不能修改。	主通道的计数器加载 TDRn 的值到 TCRn 并向下计数。当计数值达到 TCRn=0000H, 产生 INTTMn。同时, TDRn 寄存器的值加载到 TCRn, 计数器开始再次向下计数。 在从通道, 加载 TDRp 的值到 TCRp, 由主通道的 INTTMn 触发并且计数器开始向下计数。TOp 的输出电平在主通道输出 INTTMn 后的一个计数时钟开始有效。当 TCRp = 0000H 时变为无效, 且计数操作停止。 在从通道 2, TDRq 的值发送到 TCRq, 由主通道的 INTTMn 触发, 计数器开始向下计数。TOq 的输出电平在主通道输出 INTTMn 后的一个计数时钟开始有效。当 TCRq = 0000H 时变为无效, 并且计数操作停止。之后, 重复上面的操作。
	操作停止 TTn 位 (主通道)、TTP 和 TTq (从通道) 位同时设置为 1。 因为是触发位, TTn、TTP 和 TTq 位自动返回为 0。	TEn、TEp、TEq = 0, 并且计数操作停止。 TCRn、TCRp 和 TCRq 保持计数值并停止。 TOp 和 TOq 输出不被初始化而保持当前状态。
	从通道的 TOEp 或 TOEq 清除为 0 且值设置到 TOP 和 TOq 位。	TOP 和 TOq 引脚输出 TOP 和 TOq 的设置值。
	TAUS 停止 需要保持 TOP 和 TOq 引脚输出电平时 把需要保持的值设置到端口寄存器后清除 TOP 位和 TOq 位为 0。 不需要保持 TOP 和 TOq 引脚输出电平时, 切换端口模式寄存器到输入模式。 PER2 寄存器的 TAU0EN 位清零。	TOP 和 TOq 引脚输出电平由端口功能保持。 TOP 和 TOq 引脚输出电平变为高阻输出状态。 掉电状态 所有电路被初始化, 并且每个通道的 SFR 也被初始化。 (TOP 和 TOq 位清除为 0, 并且 TOP 和 TOq 引脚设置为端口模式。)

备注 n = 00、02、04、06、08、10
 n < p < q ≤ 11
 然而, p 和 q 是连续整数。

第 7 章 变频控制功能

7.1 功能概述

μPD79F9211 可作为变频控制功能使用，并且通过采用定时器阵列单元 TAUS(以下简称为“TAUS”)和 TAUS 选项单元可用于马达控制。可使用最大分辨率为 40MHz 的工作时钟(当使用内部高速振荡时钟时)。此外，可产生 A/D 转换器的开始时间。

下列操作可通过使用变频控制功能进行。

(1) 互补 PWM 输出功能

- 6 相 PWM 输出功能(三角波调制、上臂、下臂，6 个输出)
通过组合使用 TAUS 的通道 7 可产生任意周期，占空比以及死区时间的波形。
- 半桥输出功能(2 个输出×2)
通过组合使用 TAUS 的通道 4 可产生任意周期，占空比以及死区时间的波形。
- 全桥输出功能(4 个输出)
该功能使用上述的半桥输出。

(2) 非互补 PWM 输出功能

- 6 相 PWM 输出功能(锯齿波调制或三角波调制，6 个输出)
通过组合使用 TAUS 的通道 7 可产生任意频率，占空比和没有死区时间的波形。

(3) 中断缩短功能

用于设置周期的通道 0 和 4 的中断可被缩短。

(4) 任意死区时间设置功能

可以设置任意死区时间。

(5) A/D 转换器开始时间设定功能(可产生四种类型的时间。)

A/D 转换器开始时间可使用 TAUS 的通道 8 和 9 进行输出。

(6) 可进行 0%和 100%输出。

互补 PWM 输出功能和非互补 PWM 输出功能均可进行 0%和 100%的输出。

(7) 对每个引脚都可进行定时器正向和反向输出的设置。

(8) 实时输出功能(PWM 调制可使用此功能)

(9) 强制输出停止功能

当使用内部比较器检测到有效沿时，TO02 至 TO07 的输出可设置为高阻。

当使用外部引脚输入(TMOFF0、TMOFF1)检测到有效沿时，TO02 至 TO07 的输出可设置为高阻。

7.2 变频控制功能的配置

可通过增加定时器阵列单元 TAUS 的功能来实现变频功能。

下列显示了定时器阵列单元 TAUS 和变频控制功能模块的硬件配置。

表 7-1. 定时器阵列单元 TAUS 和变频控制功能模块的配置

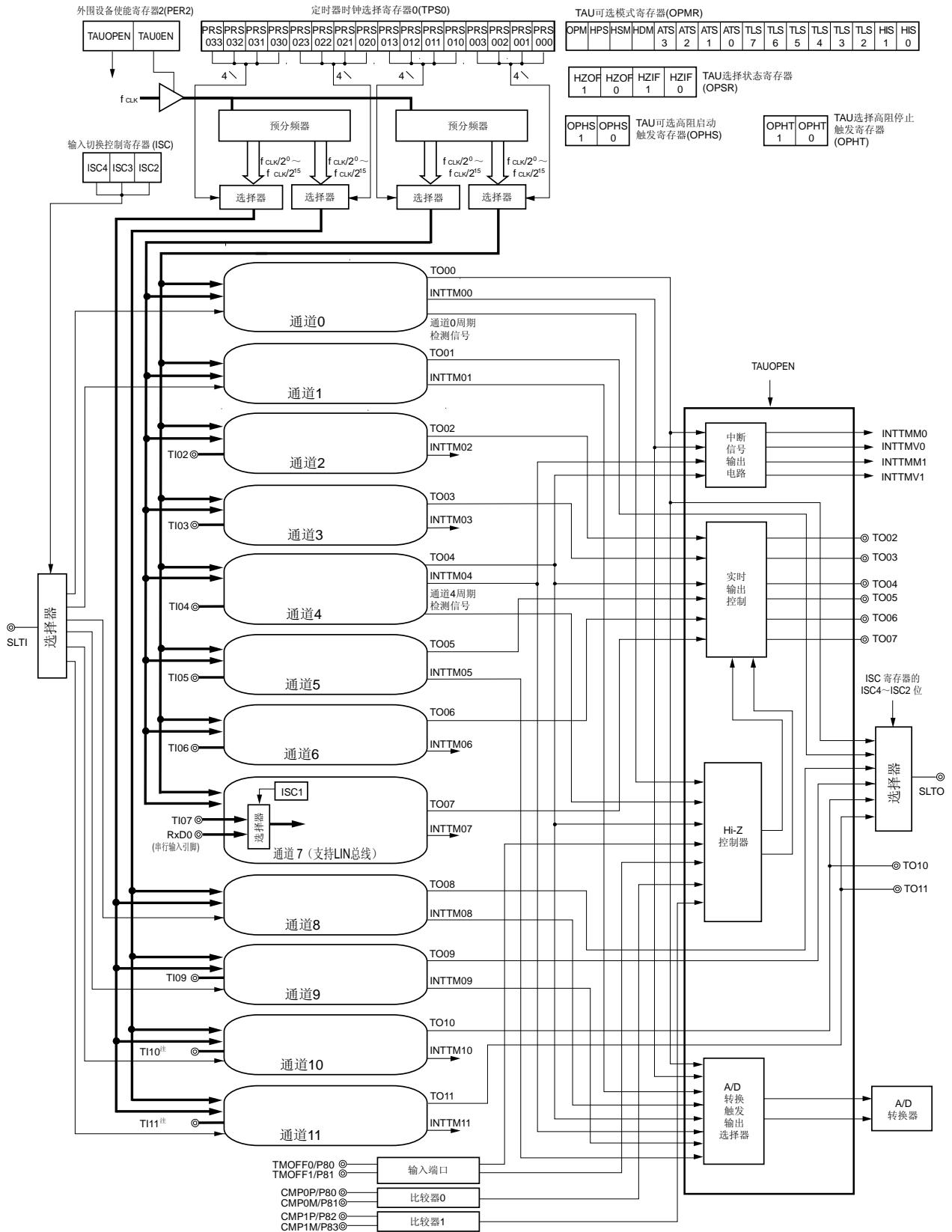
项目	配置
定时器/计数器	定时器计数寄存器 n(TCRn) [※]
寄存器	定时器数据寄存器 n(TDRn) [※]
定时器输入	TI02 ~ TI07、TI09、TI10、TI11、SLTI 引脚、RxD0 引脚 (对于 LIN 总线)
定时器输出	TO02~TO07、TO10、TO11、SLTO 引脚, 输出控制电路
控制寄存器	<p><单元设置块的寄存器></p> <ul style="list-style-type: none"> • 外围设备使能寄存器 2(PER2)[※] • 定时器时钟选择寄存器 0(TPS0)[※] • 定时器通道使能状态寄存器 0(TE0)[※] • 定时器通道启动寄存器 0(TS0)[※] • 定时器通道停止寄存器 0(TT0)[※] • 定时器输入选择寄存器 0(TIS0)[※] • 定时器输出使能寄存器 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)[※] • 定时器三角波输出模式寄存器 0(TOT0) • 定时器死区时间输出使能寄存器 0(TDE0) • 定时器实时输出寄存器 0(TRO0) • 定时器实时输出使能寄存器 0(TRE0) • 定时器实时控制寄存器 0(TRC0) • 定时器调制输出使能寄存器 0(TME0) • TAU 可选模式寄存器(OPMR) • TAU 可选状态寄存器(OPSR) • TAU 可选高阻启动触发寄存器(OPHS) • TAU 可选高阻停止触发寄存器(OPHT) <p><每个通道的寄存器></p> <ul style="list-style-type: none"> • 定时器模式寄存器 n(TMRn)[※] • 定时器状态寄存器 n(TSRn)[※] • 输入切换控制寄存器(ISC)[※]当支持 LIN 总线时使用 • 噪声滤波器使能寄存器 1、2 (NFEN1、NFEN2)[※] • 端口模式寄存器 1、3、5、7 (PM1、PM3、PM5、PM7)[※] • 端口寄存器 1、3、5、7 (P1、P3、P5、P7)[※]

注 1. 通过增加定时器阵列单元 TAUS 的可选功能实现变频控制功能。在此章节中, 只描述用于变频控制功能的寄存器。关于其它与定时器阵列单元 TAUS 共用的寄存器, 可参考 第 6 章 定时器阵列单元 TAUS。

备注 n: 通道编号(n =00~11)

图 7-1 所示为框图。

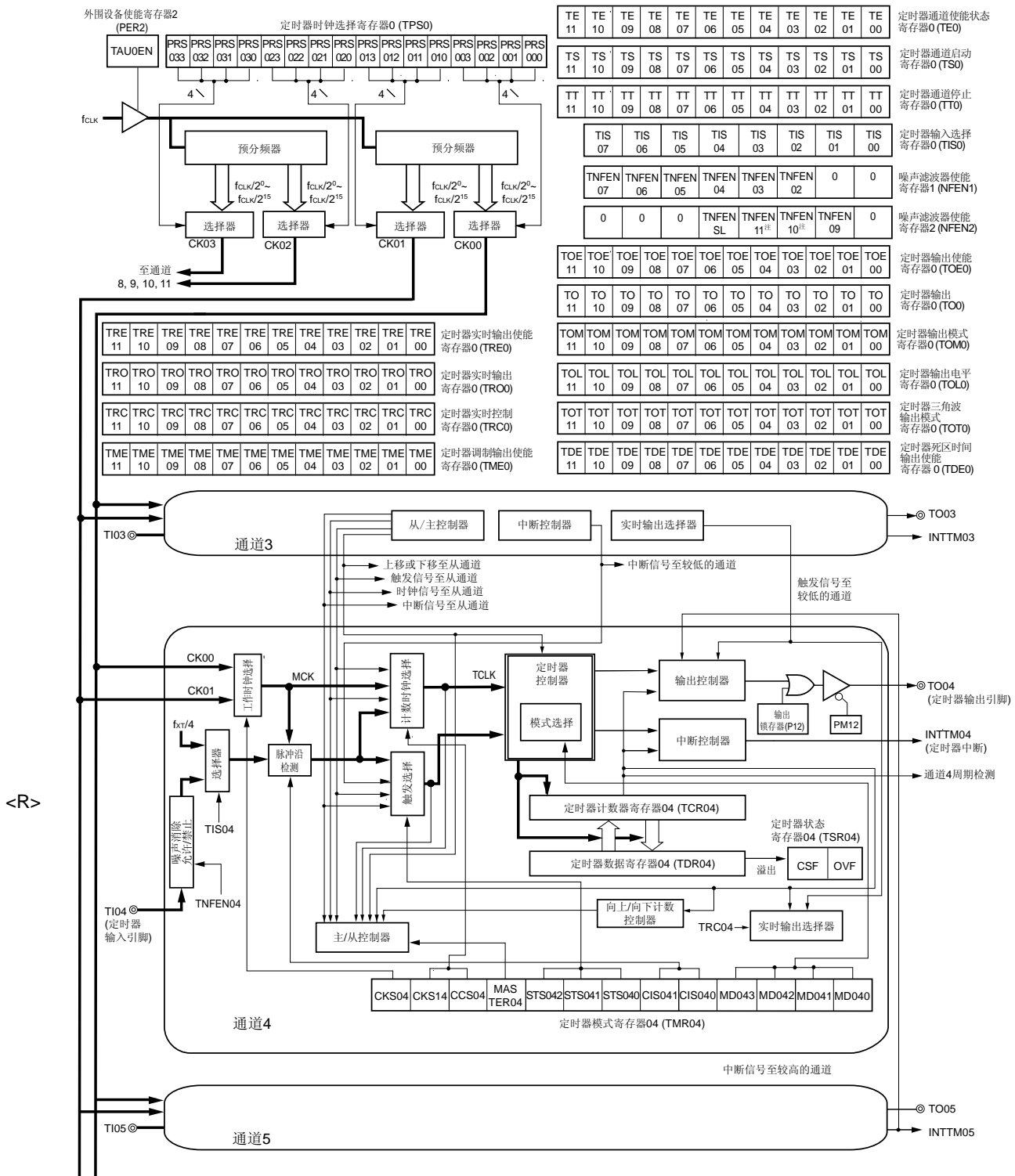
图 7-1. 定时器阵列单元 TAUS 和变频控制功能模块的整体配置



备注

图 7-1 中的配置图包括用于定时器阵列单元 TAUS 的寄存器和引脚。要了解定时器阵列单元 TAUS 的详细信息，可参考第 6 章 定时器阵列单元 TAUS。

图 7-2. 每个通道的框图



备注 图 7-2 中的框图包括共用于定时器阵列单元 TAUS 的寄存器和引脚。要了解定时器阵列单元 TAUS 的详细信息，可参考第 6 章 定时器阵列单元 TAUS。

- (1) 定时器计数寄存器 n(TCRn)
- (2) 定时器数据寄存器 n(TDRn)

上述寄存器和定时器阵列单元 TAUS 共用。

详细情况，请参考 **6.2 定时器阵列单元 TAUS 的配置**。

7.3 控制定时器阵列单元TAUS和变频控制功能模块的寄存器

下列寄存器控制定时器阵列单元 TAUS 和变频控制功能模块。

- 外围设备使能寄存器 2(PER2)
- 定时器时钟选择寄存器 0(TPS0)^注
- 定时器模式寄存器 n(TMRn)^注
- 定时器状态寄存器 n(TSRn)^注
- 定时器通道使能状态寄存器 0(TE0)^注
- 定时器通道启动寄存器 0(TS0)^注
- 定时器通道停止寄存器 0(TT0)^注
- 定时器输入选择寄存器 0(TIS0)^注
- 定时器输出使能寄存器 0(TOE0)^注
- 定时器输出寄存器 0(TO0)^注
- 定时器输出电平寄存器 0(TOL0)^注
- 定时器输出模式寄存器 0(TOM0)^注
- 定时器三角波输出模式寄存器 0(TOT0)
- 定时器死区时间输出使能寄存器 0(TDE0)
- 定时器实时输出寄存器 0(TRO0)
- 定时器实时输出使能寄存器 0(TRE0)
- 定时器实时控制寄存器 0(TRC0)
- 定时器调制输出使能寄存器 0(TME0)
- TAU 可选模式寄存器(OPMR)
- TAU 可选状态寄存器(OPSR)
- TAU 可选高阻启动触发寄存器(OPHS)
- TAU 可选高阻停止触发寄存器(OPHT)
- 输入切换控制寄存器(ISC)^注
- 噪声滤波器使能寄存器 1、2 (NFEN1、NFEN2)^注
- 端口模式寄存器 1、3、5、7(PM1、PM3、PM5、PM7)^注
- 端口寄存器 1、3、5、7 (P1、P3、P5、P7)^注

注 可通过增加定时器阵列单元 TAUS 的可选功能来实现变频控制功能。在此章节中，只描述用于变频控制功能的寄存器。关于其它与定时器阵列单元 TAUS 共用的寄存器，可参考 **第6章 定时器阵列单元 TAUS**。

备注 n = 00~11

(1) 外围设备使能寄存器 2(PER2)

PER2 用于使能或禁止每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少消耗和噪音。

使用定时器阵列单元 TAUS 和变频控制功能时，确保设置此寄存器的位 0(TAU0EN)和位 1(TAUOPEN)为 1。

PER2 可以通过 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

- 注意事项**
1. 设置定时器阵列单元 TAUS 和变频控制功能时，确保先行设置 TAU0EN 和 TAUOPEN 为 1。如果 TAU0EN 和 TAUOPEN 被设为 0，控制定时器阵列单元 TAUS 和变频控制功能的寄存器的写入被忽略，并且所有读取值都是默认值。
 2. 确保清除 PER2 寄存器的第 2 至 7 位为 0。

图 7-3. 外围设备使能寄存器 2(PER2)的格式

地址：F00F2H 复位后：00H R/W

符号	7	6	5	4	3	2	<1>	<0>
PER2	0	0	0	0	0	0	TAUOPEN	TAU0EN

TAU0EN	定时器阵列单元 TAUS 输入时钟的控制
0	停止输入时钟的供给 <ul style="list-style-type: none"> 由定时器阵列单元 TAUS 使用的 SFR 不能写入。 定时器阵列单元 TAUS 处于复位状态。
1	供给输入时钟 <ul style="list-style-type: none"> 由定时器阵列单元 TAUS 使用的 SFR 可读/写。

TAUOPEN	变频控制模块输入时钟的控制
0	停止输入时钟的供给 <ul style="list-style-type: none"> 由变频控制模块使用的 SFR 不能写入。 变频控制模块在复位状态。
1	供给输入时钟 <ul style="list-style-type: none"> 由变频控制模块使用的 SFR 可读/写。

- (2) 定时器时钟选择寄存器 0(TPS0)
- (3) 定时器模式寄存器 n(TMRn)
- (4) 定时器状态寄存器 n(TSRn)
- (5) 定时器通道使能状态寄存器 0(TE0)
- (6) 定时器通道启动寄存器 0(TS0)
- (7) 定时器通道停止寄存器 0(TT0)
- (8) 定时输入选择寄存器 0(TIS0)
- (9) 定时器输出使能寄存器 0(TOE0)
- (10) 定时器输出寄存器 0(TO0)
- (11) 定时器输出电平寄存器 0(TOL0)
- (12) 定时器输出模式寄存器 0(TOM0)

上述寄存器(2)至(12)和定时器阵列单元 TAUS 共用。
详细情况，请参考 6.2 定时器阵列单元 TAUS 的配置。

(13) 定时器三角波输出模式寄存器 0(TOT0)

TOT0 用于控制通过设置 TOM0 寄存器的 TOMn 为 1 来设置的通道的定时器输出模式。
当 TREN、TMEn=0 或 TREN、TMEn=1，以及定时器输出使能(TEEn=1)且 TOMn 被设为 1 的状态下，通过此寄存器对各通道 n 的设置在定时器输出信号被置位或复位时反映。
当定时器操作为停止(TE0 = 0)状态时，TOT0 可被重写入。
TOT0 可以通过 16 位存储器操作指令来设置。
复位后这个寄存器的值为 0000H。

图 7-4. 定时器三角波输出模式寄存器 0(TOT0)的格式

地址: F01E8H、F01E9H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOT0	0	0	0	0	TOT 11	TOT 10	TOT 09	TOT 08	TOT 07	TOT 06	TOT 05	TOT 04	TOT 03	TOT 02	TOT 01	TOT 00

TOTn	通道 n 的组合操作模式选择
0	由主通道的定时器中断请求信号(INTTMn)置位，由从通道的定时器中断请求信号(INTTMm)复位。
1	由向下状态的定时器中断请求信号(INTTMn)置位，由向上状态的定时器中断请求信号(INTTMn)*复位。

注 当产生三角波 PWM 时设置从通道为 TOTn = 1。

注意事项 确保设置第 15 至 12 位为 0。

备注 n: 通道编号, m: 从通道编号
 n=00~11 (主通道: n=00、02、04、06、08、10)
 n<m≤11

(14) 定时器实时输出使能寄存器 0(TRE0)

TRE0 是当使用实时输出功能时，设置各通道定时器输出允许/停止的寄存器。

当定时器运行停止(TE0=0)时，TRE0 可被重写。

TRE0 可以通过 16 位存储器操作指令来设置。

复位后这个寄存器的值为 0000H。

图 7-5. 定时器实时输出使能寄存器 0(TRE0)的格式

地址：F01EAH、F01EBH 复位后：0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRE0	0	0	0	0	TRE	TRE	TRE	TRE	TRE	TRE	TRE	TRE	TRE	TRE	TRE	TRE
					11	10	09	08	07	06	05	04	03	02	01	00

TREn	通道 n 的组合操作模式选择
0	停止实时输出。
1	允许实时输出。

注意事项 确保设置第 15 至 12 位为 0。

备注 n = 00~11

(15) 定时器实时输出寄存器 0(TRO0)

TRO0 是用于实时输出功能的定时器输出缓冲寄存器。当实时输出允许时，此寄存器中每个位的值从每个通道的定时器输出引脚(TOn)输出。

当实时输出停止时(TREn=0)，TRO0 设置不会影响定时器的的工作。当实时输出允许时(TREn=1)，TOn 引脚输出根据实时输出和定时器操作改变。

TRO0 可以通过 16 位存储器操作指令来设置。

复位后这个寄存器的值为 0000H。

图 7-6. 定时器实时输出寄存器 0(TRO0)的格式

地址：F01ECH、F01EDH 复位后：0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRO0	0	0	0	0	TRO	TRO	TRO	TRO	TRO	TRO	TRO	TRO	TRO	TRO	TRO	TRO
					11	10	09	08	07	06	05	04	03	02	01	00

TROn	实时输出电平
0	低电平
1	高电平

注意事项 确保设置第 15 至 12 位为 0。

备注 n = 00~11

(16) 定时器实时控制寄存器 0(TRC0)

TRC0 是设置产生实时输出触发通道的寄存器。
当定时器运行停止(TE0=0)时，TRC0 可被重写。
TRC0 可以通过 16 位存储器操作指令来设置。
复位后这个寄存器的值为 0000H。

图 7-7. 定时器实时控制寄存器 0(TRC0)的格式

地址： F01EEH、 F01EFH 复位后： 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRC0	0	0	0	0	TRC	TRC	TRC	TRC	TRC	TRC	TRC	TRC	TRC	TRC	TRC	TRC
					11	10	09	08	07	06	05	04	03	02	01	00

TRCn	实时输出触发功能的选择
0	不要作为实时输出触发产生通道。 设置为 TRCn = 1 的较高通道的通道定时器中断请求信号(INTTMn)成为实时输出触发。
1	作为实时输出触发产生通道。 本通道定时器中断请求信号(INTTMn)成为较低通道的实时输出触发。

注意事项 确保设置第 15 至 12 位为 0。

备注 n = 00~11

(17) 定时器死区时间输出使能寄存器 0(TDE0)

TDE0 允许或禁止每个通道的定时器输出死区时间控制。

通过此寄存器对各通道 n 的设置 在定时器输出被允许(TOEn=1)且 TOMn 和 TOTn 被设为 1 的状态下，在定时器输出信号的置位或复位时被反映。

偶数通道和奇数通道的 TDEn 值必须一致用来进行死区时间控制，因为死区时间控制是以一组偶数和奇数通道(偶数通道+1)进行操作的。

TDE0 可以通过 16 位存储器操作指令来设置。

复位后这个寄存器的值为 0000H。

图 7-8. 定时器死区时间输出使能寄存器 0(TDE0)的格式

地址： F01F2H、 F01F3H 复位后： 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDE0	0	0	0	0	TDE	TDE	TDE	TDE	TDE	TDE	TDE	TDE	TDE	TDE	TDE	TDE
					11	10	09	08	07	06	05	04	03	02	01	00

TDEn	死区时间控制的选择
0	禁止死区时间控制。
1	使能死区时间控制。

注意事项 确保设置第 15 至 12 位为 0。

备注 n = 00~11

(18) 定时器调制输出使能寄存器 0(TME0)

TME0 是控制定时器输出和实时输出的调制输出功能的操作使能/禁止的寄存器。
TME0 可以通过 16 位存储器操作指令来设置。
复位后这个寄存器的值为 0000H。

图 7-9. 定时器调制输出使能寄存器 0(TME0)的格式

地址： F01F0H、 F01F1H 复位后： 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TME0	0	0	0	0	TME	TME	TME	TME	TME	TME	TME	TME	TME	TME	TME	TME
					11	10	09	08	07	06	05	04	03	02	01	00

TME _n	实时输出电平
0	禁止定时器输出和实时输出的调制输出。
1	使能定时器输出和实时输出的调制输出。

注意事项 确保设置第 15 至 12 位为 0。

备注 n = 00~11

(19) TAU 可选模式寄存器(OPMR)

OPMR 设置变频控制功能可选单元的操作模式。

OPMR 可由 16 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 7-10. TAU 可选模式寄存器(OPMR)的格式 (1/2)

地址: F0220H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPMR	OPM	HPS	HSM	HDM	ATS	ATS	ATS	ATS	TLS	TLS	TLS	TLS	TLS	TLS	HIS0	HIS
					3	2	1	0	7	6	5	4	3	2	1	0

OPM	操作模式选项
0	6 相输出控制模式(TO02~TO07 变为高阻控制对象, 高阻控制和取消由 HDM 位设置。)
1	半桥输出控制模式(当通道 0 和 4 为周期寄存器时) (TO02 和 TO03 通过 TMOFF0 引脚或内部比较器 CMP0 被设为高阻。通过 TMOFF1 引脚或内部比较器 CMP1、TO06 和 TO07 被设为高阻。高阻态由 HSM 位取消。)

HPS	高阻输入引脚选择
0	使用 TMOFF0 和 TMOFF1 引脚作为高阻控制信号。
1	使用内部比较器输出信号作为高阻控制信号。

HSM	高阻取消方法的选择 (当 OPM=1)
0	在检测到内部比较器(CMP0/CMP1)或 TMOFF0、TMOFF1 的无效沿之后, 高阻态可在周期内被同步取消。
1	在检测到由软件写入的边沿之后, 高阻态可在周期内被同步取消。

HDM	高阻取消方法的选择 (当 OPM=0)
0	两级过电流检测模式 (当检测到内部比较器 0(CMP0 侧)或 TMOFF0 的有效沿之后, 被设置为高阻态; 当检测到无效沿之后, 高阻态可在周期内被同步取消。此外, 当检测到内部比较器 1(CMP1 侧)或 TMOFF1 的有效沿之后, 高阻态被设置; 当检测到软件写入的边沿之后, 高阻态可在周期内被同步取消。)
1	过电流/电动势检测模式 (通过内部比较器 0 输出反转或 TMOFF0 有效沿反转, 并检测到过电流侧(高电势 CMP1 或 TMOFF1)和电动势侧(低电势 CMP0 或 TMOFF0)之后, 高阻态被设置。在检测到内部比较器或 TMOFF0、TMOFF1 的无效沿之后, 高阻态可在周期内被同步取消。)

图 7-10. TAU 可选模式寄存器(OPMR)的格式 (2/2)

地址: F0220H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPMR	OPM	HPS	HSM	HDM	ATS	ATS	ATS	ATS	TLS	TLS	TLS	TLS	TLS	TLS	HIS	HIS
					3	2	1	0	7	6	5	4	3	2	1	0

ATS	ATS	A/D1 转换输出触发的选择 OPM=0: 通道 9 用于产生中断 OPM=1: 通道 5 用于产生中断
3	2	
0	0	在主通道的向下状态周期中的匹配中断时产生 A/D 触发。
0	1	在主通道的向上状态周期中的匹配中断时产生 A/D 触发。
1	0	在主通道的向上或向下状态周期中的匹配中断。
1	1	在主通道的向上或向下状态周期中的匹配中断+主通道的谷型中断。

ATS	ATS	A/D0 转换输出触发的选择 OPM=0: 通道 8 用于产生中断 OPM=1: 通道 1 用于产生中断
1	0	
0	0	在主通道的向下状态周期中的匹配中断时产生 A/D 触发。
0	1	在主通道的向上状态周期中的匹配中断时产生 A/D 触发。
1	0	在主通道的向上或向下状态周期中的匹配中断。
1	1	在主通道的向上或向下状态周期中的匹配中断+主通道的谷型中断。

TLS7 至 TLS2	输出反向控制
0	定时器输出(TOn)进行正向输出。
1	定时器输出(TOn)进行反向输出。

HIS1	TMOFF1 有效沿的选择
0	设置下降沿有效。
1	设置上升沿有效。

HIS0	TMOFF0 有效沿的选择
0	设置下降沿有效。
1	设置上升沿有效。

备注 n=02~07

(20) TAU 可选状态寄存器(OPSR)

OPSR 显示马达控制可选单元的不同状态。

OPSR 可由 16 位存储器操作指令读取。

复位后这个寄存器的值为 0000H。

图 7-11. TAU 可选状态寄存器(OPSR)的格式

地址： F0222H 复位后： 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPSR	0	0	0	0	0	0	HZO F1	HZO F0	0	0	0	0	0	0	HZIF 1	HZIF 0

HZOF1	高阻输出信号 1 的工作状态
0	正常状态(定时器输出)
1	高阻输出状态

HZOF0	高阻输出信号 0 的工作状态
0	正常状态(定时器输出)
1	高阻输出状态

HZIF1	TMOFF1 引脚/内部比较器 0 的输出信号状态
0	当 OPMR 的 HPS 为 0 时，TMOFF1 引脚处于低电平。 当 OPMR 的 HPS 为 1 时，内部比较器 1 的输出信号处于低电平。
1	当 OPMR 的 HPS 为 0 时，TMOFF1 引脚处于高电平。 当 OPMR 的 HPS 为 1 时，内部比较器 1 的输出信号处于高电平。

HZIF0	TMOFF0 引脚/内部比较器 0 的输出信号状态
0	当 OPMR 的 HPS 为 0 时，TMOFF0 引脚处于低电平。 当 OPMR 的 HPS 为 1 时，内部比较器 0 的输出信号处于低电平。
1	当 OPMR 的 HPS 为 0 时，TMOFF0 引脚处于高电平。 当 OPMR 的 HPS 为 1 时，内部比较器 0 的输出信号处于高电平。

(21) TAU 可选高阻启动触发寄存器 (OPHS)

OPHS 设置高阻控制器的软件触发。当软件触发被设为有效时，设置该寄存器为 1 来启动 TOn 引脚的高阻输出。

OPHS 可由 16 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 7-12. TAU 可选高阻启动触发寄存器 (OPHS) 的格式

地址: F0224H 复位后: 0000H W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPHS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OPHS1	OPHS0

OPHS1	高阻控制输出信号 1
0	—
1	高电平（输出端口高阻输出）

OPHS0	高阻控制输出信号 0
0	—
1	高电平（输出端口高阻输出）

注意事项 确保设置第 15 至 2 位为 0。

备注 n=02~07

(22) TAU 可选高阻停止触发寄存器 (OPHT)

OPHT 设置高阻控制器的软件触发。当软件触发被设为有效时，设置该寄存器为 1 来取消 TOn 引脚的高阻态。
OPHT 可由 16 位存储器操作指令设置。
复位后这个寄存器的值为 0000H。

图 7-13. TAU 可选高阻停止触发寄存器 (OPHT) 的格式

地址: F0226H 复位后: 0000H W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPHT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OPHT1	OPHT0

OPHT1	高阻控制输出信号 1
0	—
1	低电平（输出端口高阻取消）

OPHT0	高阻控制输出信号 0
0	—
1	低电平（输出端口高阻取消）

注意事项 确保设置第 15 至 2 位为 0。

备注 n=02~07

- (23) 输入切换控制寄存器 (ISC)
- (24) 噪声滤波器使能寄存器 1、2(NFEN1、NFEN2)
- (25) 端口模式寄存器 1、3、5、7(PM1、PM3、PM5、PM7)

上述寄存器(23)至(25)和定时器阵列单元 TAUS 共用。
详细情况，请参考 6.2 定时器阵列单元 TAUS 的配置。

7.4 使用变频控制功能的操作

7.4.1 作为实时输出功能的操作（类型 1）

通过使用实时输出触发产生通道的 INTTMn 输出，可以把 TROn 和 TROm 的值由 TOn 和 TOm 输出。

该功能是在 **6.7.1 作为间隔定时器/方波输出的操作**中描述功能的扩展功能。

实时输出触发产生通道（TRCn=1 被设置的通道）在一个固定的间隔内输出 INTTMn 并且产生一个实时输出触发。

实时输出通道（TRCm=0 被设置的通道）通过实时输出触发把输出 TROm 的设置值从 TOm 输出。

可通过下面的表达式计算出中断产生周期。

$$\text{INTTMn 产生周期} = \text{计数时钟周期} \times (\text{TDRn 的设置值} + 1)$$

被设置 TRCn=1 的通道成为实时输出触发产生通道并且运行在间隔定时器模式下。

通道启动触发位（TSn）设置为 1 后，TCRn 在第一个计数时钟加载 TDRn 的值。此时，当 TMRn 的 MDn0=0 时 INTTMn 不输出且 TOn 不反转。当 TMRn 的 MDn0=1 时 INTTMn 输出且 TOn 反转。

之后，TCRn 随着计数时钟而向下计数。

当 TCRn 变为 0000H 时，INTTMn 输出且 TOn 在下一个计数时钟反转。此时 TCRn 再次加载 TDRn 的值。此后继续相同的操作。

在实时输出触发产生通道的 INTTMn 输出时，TROn 的设置值从 TOn 处输出。

实时输出触发产生通道（TRCn=1）的低通道的 TOm（实时输出通道（TRCm=0））由 TREm 和 TRCm 进行控制。单单重写 TROm 不会改变 TOm 的输出电平。

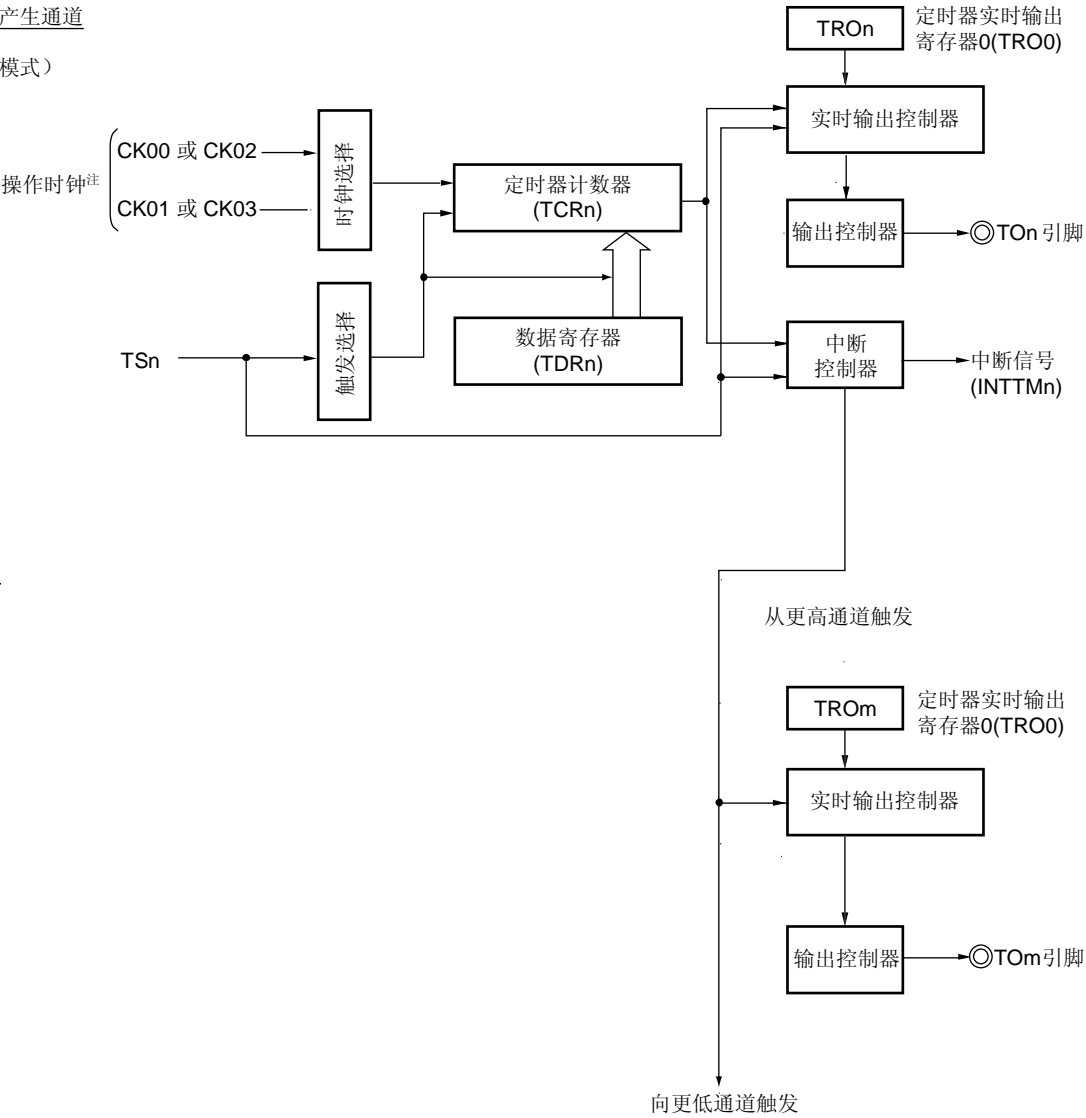
实时输出通道（TRCm=0）的 TREm 为 1 时，在实时输出触发产生通道的 INTTMn 输出时 TOm 输出 TROm 的设置值。当低通道的 TREm 或 TRCm 为 0 或 1 时，在实时输出触发产生通道的 INTTMn 输出时 TOm 不反转。

当此功能被使用时，低通道的 TCRm、TDRm 和 INTTMm 可用于不同功能。

备注 n = 01~10
 m = 02~11

图 7-14 作为实时输出功能操作的框图（类型 1）

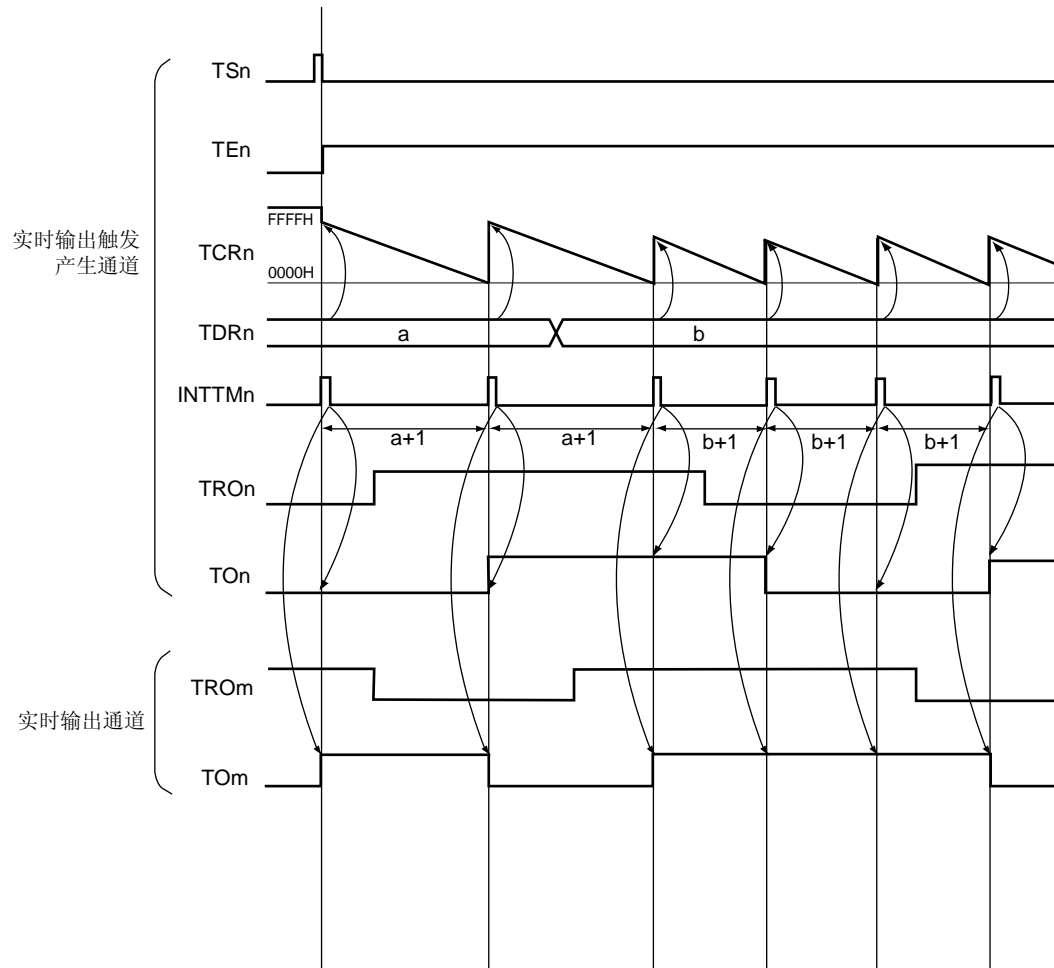
实时输出触发产生通道
TRCn = 1
(间隔定时器模式)



注 通道 0 至 7 的工作时钟从 CK00 和 CK01 中选择，通道 8 至 11 的工作时钟从 CK02 和 CK03 中选择。

备注 n = 01~10
m = 02~11

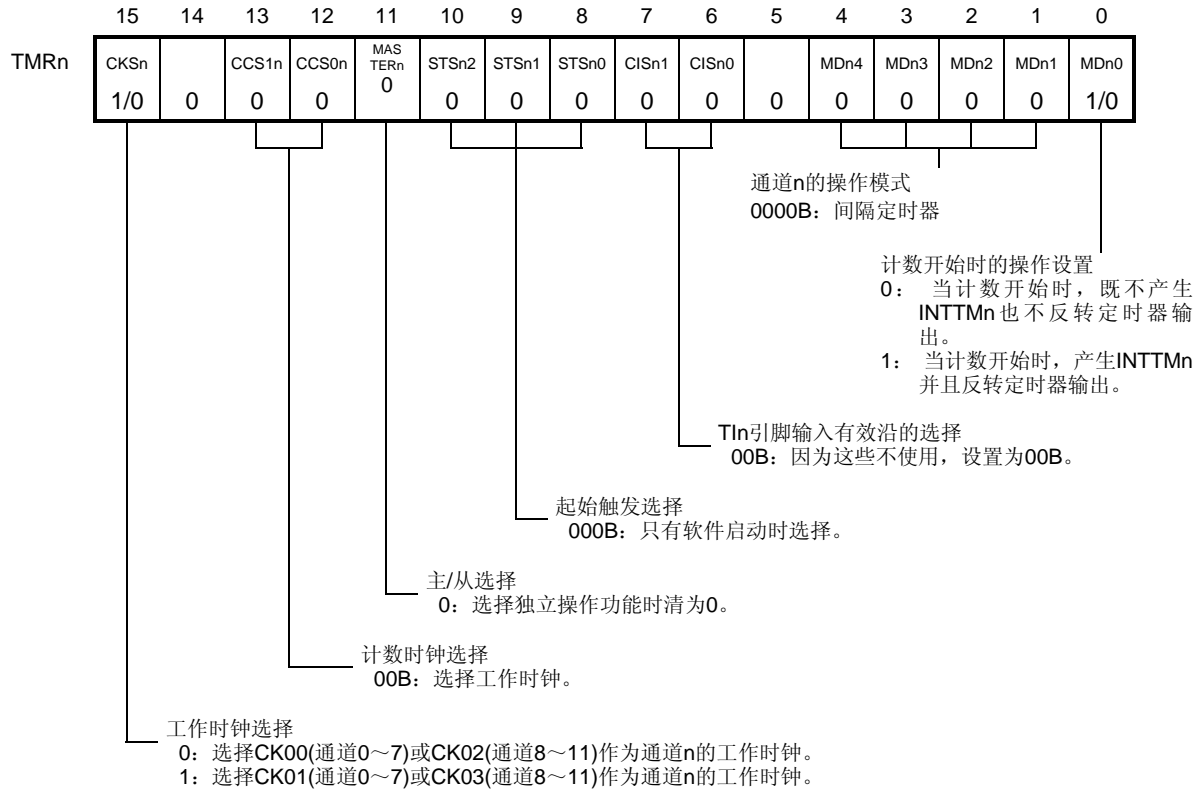
图 7-15. 作为实时输出功能操作的基本时序示例(类型 1) (MDn0=1)



备注
n = 01~10
m = 02~11

图 7-16. 作为实时输出功能操作的寄存器内容设置示例(类型 1)(1/2)

(a) 实时输出触发产生通道(TRCn=1)的定时器模式寄存器 n(TMRn)



(b) 实时输出触发产生通道(TRCn=1)的其它寄存器

TOE0:TOEn	0 : 通过计数操作, 停止 TOn 输出。 1: 通过计数操作, 允许TOn输出。
TO0:TOn	0 : 从 TOn 输出低电平。 1: 从TOn输出高电平。
TOM0:TOMn	0: 反转模式
TOT0:TOTn	0: 当TOMn=0(反转模式)时设为0。
TOL0:TOLn	0: 当TOMn=0(反转模式)时设为0。
TDE0:TDEn	0: 停止死区时间控制。
TRE0:TREn	0 : 停止实时输出。 1: 允许实时输出。
TRO0:TROn	0 : 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCn	1: 作为实时输出触发产生通道运行。
TME0:TMEn	0: 停止调制输出。

备注 n = 01~10

图 7-16. 作为实时输出功能操作的寄存器内容设置示例(类型 1)(2/2)

(c) 实时输出通道($TRCm=0$)的定时器模式寄存器 m(TMRm)

实时输出功能(类型 1)中, 当 $TRCm$ 被设为 0 时通道 $TMRm$ 可任意设置。

(d) 实时输出通道($TRCm=0$)的其它寄存器

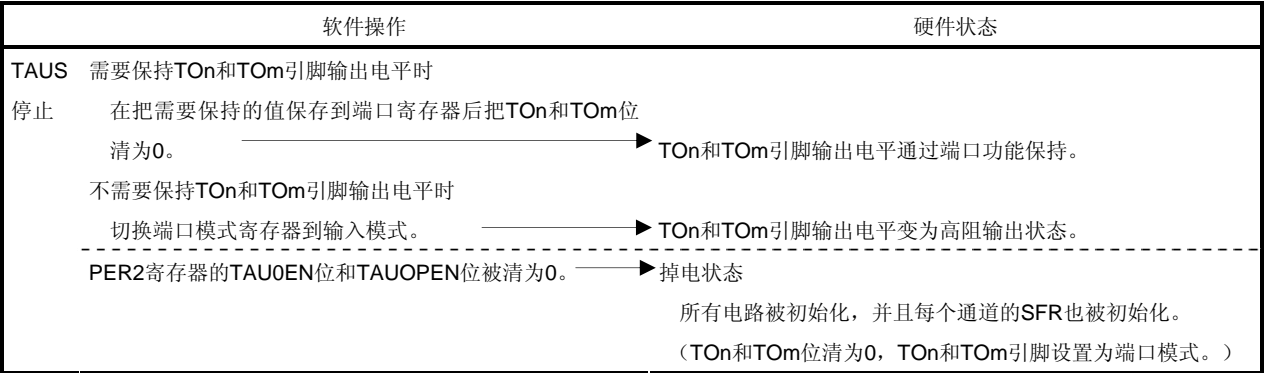
TOE0:TOEm	0 : 通过计数操作, 停止 TOm 输出操作。 1: 通过计数操作, 允许 TOm 输出操作。
TO0:TOm	0 : 从 TOm 输出低电平。 1: 从 TOm 输出高电平。
TOM0:TOMm	0: 当 $TREm=1$ 时设为 0(允许实时输出)。
TOT0:TOTm	0: 当 $TOMm = 0$ (反转模式)时设为 0。
TOL0:TOLm	0: 当 $TOMm = 0$ (反转模式)时设为 0。
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	1: 允许实时输出。
TRO0:TROm	0 : 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。

备注 $m = 02 \sim 11$

图 7-17. 实时输出功能的操作过程（类型 1）（1/2）

	软件操作	硬件状态
	TAUS 默认设置	掉电状态 (停止供给时钟, 并且所有寄存器不能写入。)
	设置PER2寄存器的TAU0EN位和TAUOPEN位为1。→	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
	设置TPS0 寄存器。 把CK00和CK01的时钟频率用于通道0~7, 把CK02和CK03的时钟频率用于通道8~11。	
<R>	通道默认设置 [实时输出触发产生通道 (TRCn=1)]	通道停止操作。 (供给时钟并且产生一些功耗。)
	设置TMRn寄存器 (决定通道的操作模式)。 设置间隔 (周期) 值到TDRn寄存器。 设置TRCm位为1 (触发产生通道)。 设置TREm位为1 (允许实时输出)。 [实时输出通道 (TRCm = 0)] 设置TRCm位为0 (非触发产生通道)。 设置TREm位为1 (允许实时输出)。	TON和TOM引脚为高阻输出状态。
	设置TOEn和TOEm位为1并允许 TON和TOM输出。 将端口寄存器和端口模式寄存器清为0。→	TON和TOM不改变因为通道停止工作。 TON 和 TOM引脚输出TON和TOM的设定值。
操作继续。	操作开始 设置TOEn和TOEm位为1 (仅当操作恢复时)。 设置触发产生通道的TSn位为1。 由于是触发位, TSn位自动恢复为0。	[实时输出触发产生通道 (TRCn=1)] TEn=1, 并且计数操作开始。 在计数时钟输入时TDRn的值被加载到TCRn。如果TMRn寄存器的MDn0位为1, 将产生INTTMn。
	操作期间 TDRn寄存器的设置值不能改变。 TCRn寄存器总是可读。 TROn和TROm位的设置值可以改变。	计数器 (TCRn) 向下计数。当计数值达到0000H时, TDRn的值被再次加载到TCRn, 并且计数操作继续。通过检测TCRn=0000H, 产生INTTMn。之后, 重复上面的操作。 在INTTMn输出时序下, 实时输出通道的TROm设置值从TOM处输出。
	操作停止 将TTn位设置为1。 由于是触发位, TTn位自动恢复为0。	TEn=0, 并且计数操作停止。 TCRn保持计数值并停止。 TON输出没有被初始化但保持当前状态并停止。
	TOEn和TOEm位被清零且数值被设置到TON和TOM。→	TON和TOM的设定值初始化TON和TOM输出。
备注	n = 01~10, m = 02~11	

图 7-17. 实时输出功能的操作过程（类型 1）（2/2）



备注 n = 01～10， m = 02～11

7.4.2 作为实时输出功能的操作（类型2）

通过使用实时输出触发产生通道的 INTTMn 输出，可以把 TROn 和 TROm 的值由 TOn 和 TOm 输出。通过外部引脚输入沿检测或软件输入检测可进行实时输出。

实时输出功能（类型2）是 6.7.4 作为输入脉冲间隔测量的操作中描述的功能的扩展功能。

实时输出触发产生通道（TRCn=1 被设置的通道）在一个固定的间隔内输出 INTTMn 并且由 TIn 引脚输入的有效沿检测或设置通道开始触发（TSn）为 1 而产生的实时输出触发。

实时输出通道（TRCm=0 被设置的通道）通过实时输出触发把输出 TROm 的设置值从 TOm 输出。TCRn 的值在 INTTMn 产生的被捕捉到 TDRn 中，但是 TDRn 的值没有意义。

被设置 TRCn=1 的通道成为实时输出触发产生通道并且作为向上计数器运行在捕捉模式下。

当 TEn 为 0 时通道启动触发器（TSn）设置为 1 或当 TIn 输入的有效边沿被检测到时，TCRn 从 0000H 开始与计数时钟同步向上计数。

此时，当 TMRn 寄存器的 MDn0=0 时 INTTMn 不输出且 TOn 不反转。当 TMRn 寄存器的 MDn0=1 时 INTTMn 输出且 TOn 反转。

当 TIn 引脚检测到输入的有效沿或当通道启动触发器（TSn）被设置为 1 时，计数值被发送（捕获）到 TDRn，同时计数器（TCRn）被清为 0000H，并且产生 INTTMn。

在实时输出触发产生通道的 INTTMn 输出时，TROn 的设置值从 TOn 处输出。

实时输出触发产生通道（TRCn=1）的 TOm 低通道（实时输出通道）由 TREm 位进行控制。单单重写 TROm 不会改变 TOm 的输出电平。

实时输出通道（TRCm=0）的 TREm 为 1 时，在实时输出触发产生通道的 INTTMn 输出时 TOm 输出 TROm 的设置值。当低通道的 TREm 或 TRCm 为 0 或 1 时，在实时输出触发产生通道的 INTTMn 输出时不进行实时输出。

当使用实时输出功能（类型2）时，低通道的 TCRm、TDRm 和 INTTMm 可用于不同功能。

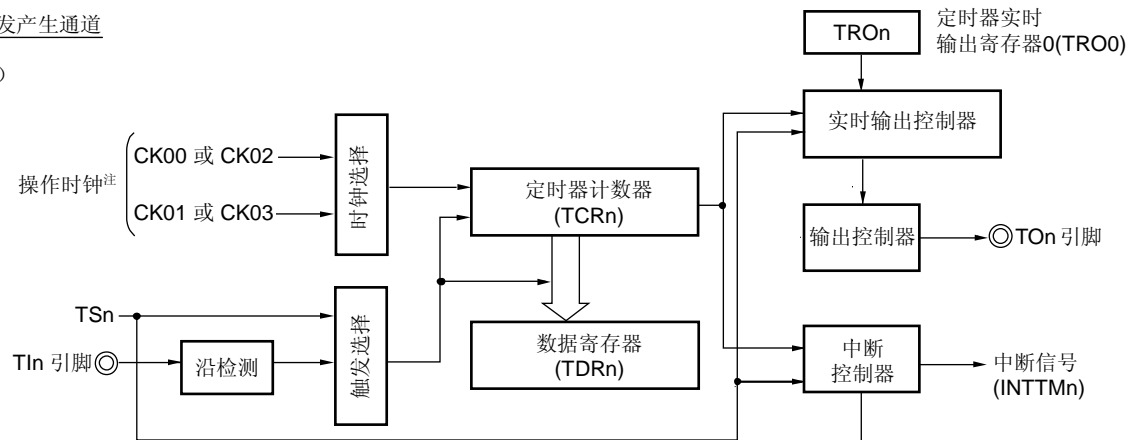
备注 n = 01~10
 m = 02~11

图 7-18 作为实时输出功能操作的框图（类型 2）

实时输出触发产生通道

TRCn = 1

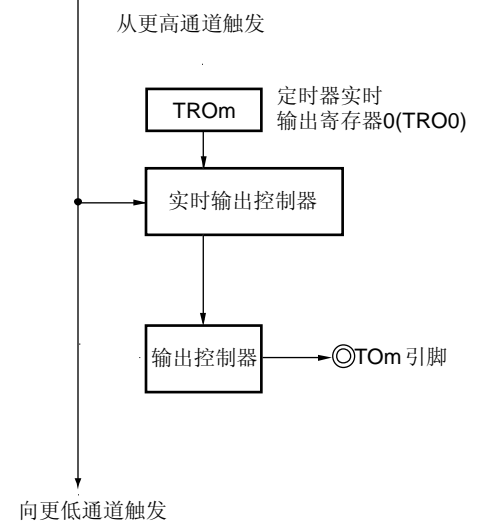
（捕捉模式）



实时输出通道

TRCm = 0

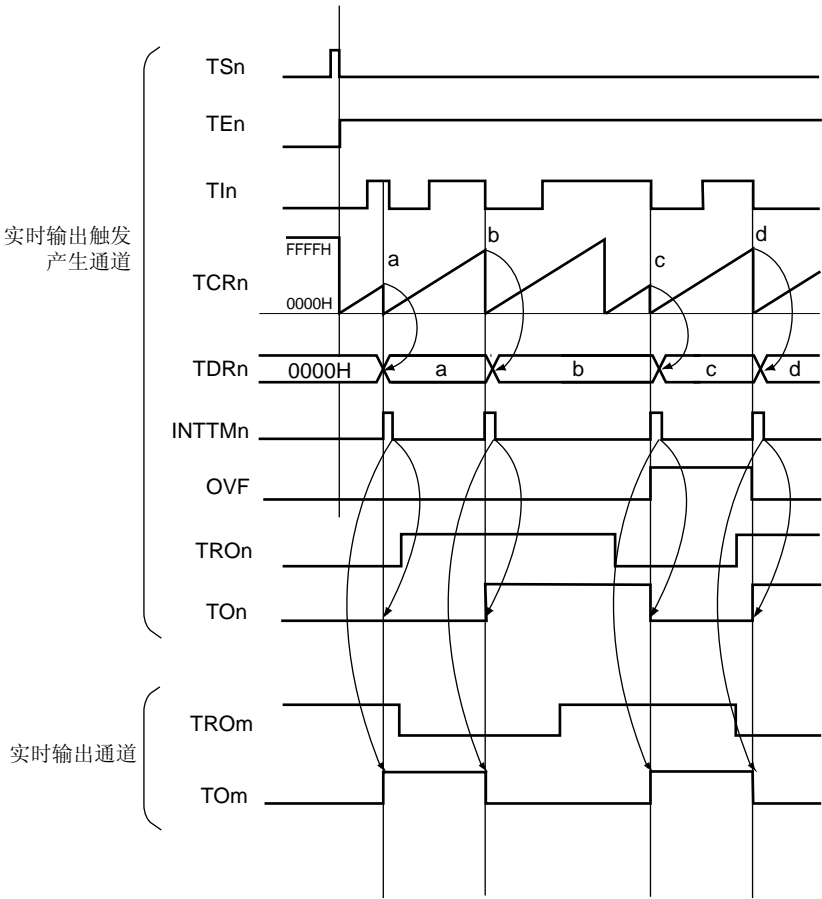
（任意模式）



注 通道 0 至 7 的工作时钟从 CK00 和 CK01 中选择，通道 8 至 11 的工作时钟从 CK02 和 CK03 中选择。

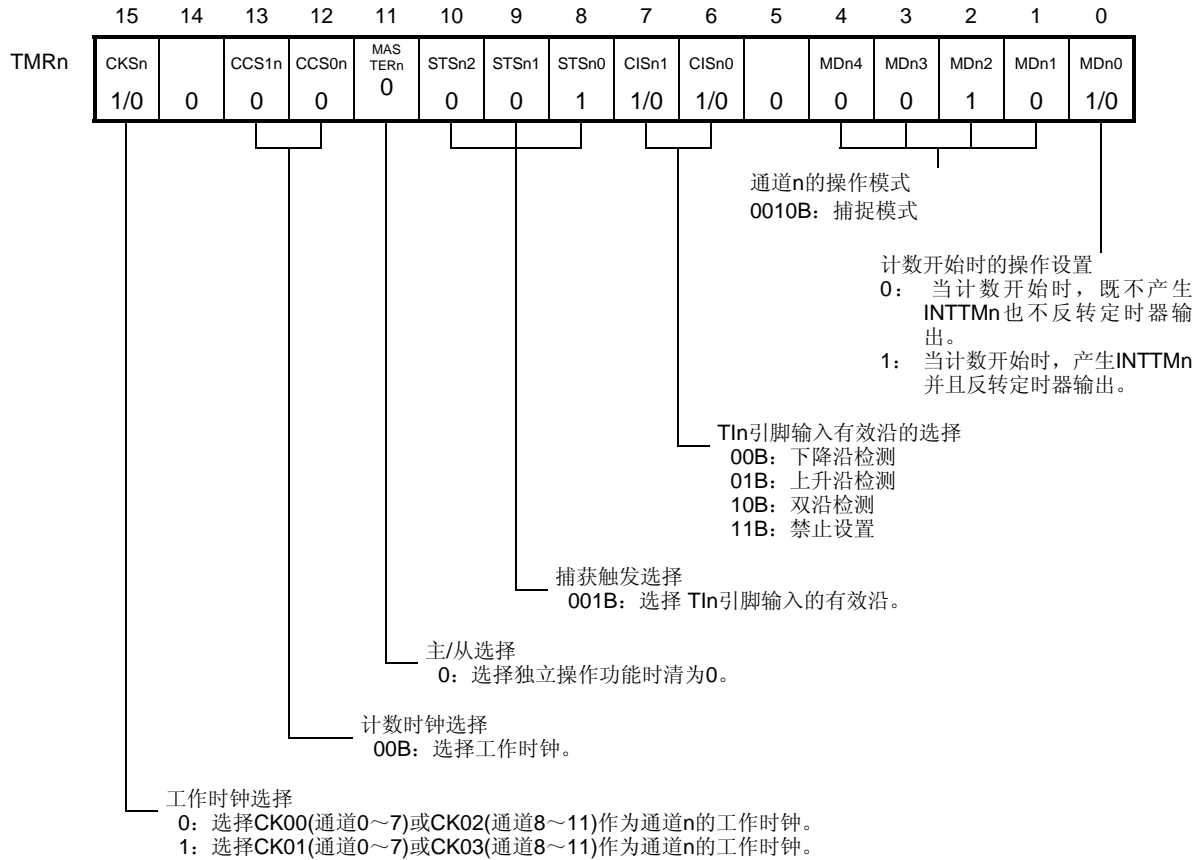
备注 n = 01~10
m = 02~11

图 7-19. 作为实时输出功能操作的基本时序示例(类型 2) (MDn0=1)



备注 n = 01~10
 m = 02~11

图 7-20. 作为实时输出功能操作的寄存器内容设置示例（类型 2）(1/2)

(a) 实时输出触发产生通道($TRCn=1$)的定时器模式寄存器 n (TMRn)(b) 实时输出触发产生通道($TRCn=1$)的其它寄存器

TOE0:TOEn	0: 通过计数操作, 停止TOn输出。 1: 通过计数操作, 允许TOn输出。
TO0:TOn	0: 从TOn输出低电平。 1: 从TOn输出高电平。
TOM0:TOMn	0: 反转模式
TOT0:TOTn	0: 当TOMn=0(反转模式)时设为0。
TOL0:TOLn	0: 当TOMn=0(反转模式)时设为0。
TDE0:TDEn	0: 停止死区时间控制。
TRE0:TREn	0: 停止实时输出。 1: 允许实时输出。
TRO0:TROn	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCn	1: 作为实时输出触发产生通道运行。
TME0:TMEn	0: 停止调制输出。

备注 $n = 01 \sim 10$

图 7-20. 作为实时输出功能操作的寄存器内容设置示例（类型 2）(2/2)

(c) 实时输出通道(TRCm=0)的定时器模式寄存器 m(TMRm)

实时输出功能（类型 2）中，当 TRCm 被设为 0 时通道 TMRm 可任意设置。

(d) 实时输出通道(TRCm=0)的其它寄存器

TOE0:TOEm	0: 通过计数操作，停止TOM输出操作。 1: 通过计数操作，允许TOM输出操作。
TO0:TOM	0: 从TOM输出低电平。 1: 从TOM输出高电平。
TOM0:TOMm	0: 当TREm=1时设为0（允许实时输出）。
TOT0:TOTm	0: 当TOMm = 0(反转模式)时设为0。
TOL0:TOLm	0: 当TOMm = 0(反转模式)时设为0。
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	1: 允许实时输出。
TRO0:TROm	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。

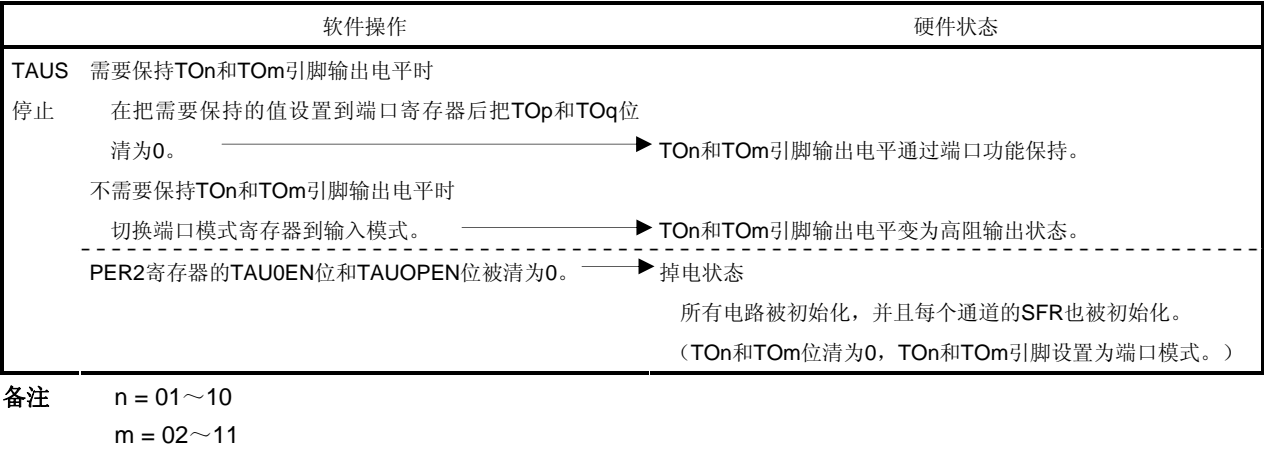
备注 m = 02~11

图 7-21. 实时输出功能的操作过程（类型 2）（1/2）

软件操作		硬件状态
TAUS 默认设置	<p>设置PER2寄存器的TAU0EN位和TAUOPEN位为1。——▶</p> <p>设置TPS0 寄存器。</p> <p>把CK00和CK01的时钟频率用于通道0至7，把CK02和CK03的时钟频率用于通道8至11。</p>	<p>掉电状态</p> <p>（停止供给时钟，并且所有寄存器不能写入。）</p> <p>上电状态。每个通道停止操作。</p> <p>（开始供给时钟并且每个寄存器允许写入。）</p>
通道默认设置	<p>实时输出触发产生通道（TRCn=1）</p> <p>设置TMRn寄存器（决定通道的操作模式）。</p> <p>设置TRCm位为1（触发产生通道）。</p> <p>设置TREM位为1（允许实时输出）。</p> <p>实时输出通道(TRCm=0)</p> <p>设置TRCm位为0（非触发产生通道）。</p> <p>设置TREM位为1（允许实时输出）。</p> <p>设置TOEn和TOEm位为1并允许TON和TOM输出。</p> <p>将端口寄存器和端口模式寄存器清为0。——▶</p>	<p>通道停止操作。</p> <p>（供给时钟并且产生一些功耗。）</p> <p>TON和TOM引脚为高阻输出状态。</p> <p>TON和TOM不改变因为通道停止工作。</p> <p>产品的TON和TOM引脚输出TON和TOM的设置值。</p>
操作开始	<p>设置TOEn和TOEm位为1（仅当操作恢复时）。</p> <p>设置触发产生通道的TSn位为1。——▶</p> <p>由于是触发位，TSn位自动恢复为0。</p>	<p>实时输出触发产生通道（TRCn=1）</p> <p>TEn=1,并且计数操作开始。</p> <p>通过计数时钟输入把TCRn清为0000H。如果TMRn寄存器的MDn0位为1，将产生INTTMn。</p>
操作期间	<p>TMRn寄存器只能改变CISn1和CISn0位的设置值。</p> <p>TROn和TROM位的设置值可以改变。</p>	<p>计数器（TCRn）从0000H开始向上计数并且当Tin引脚输入的有效沿被检测到时传输（捕捉）计数值到TDRn。此时，TCRn清为0000H，并且INTTMn发生。</p> <p>此时发生或未发生溢出，则TSRn寄存器的OVF位将被设置或清除。之后，重复上面的操作。</p> <p>在INTTMn输出时序下，实时输出通道的TROm设置值从TOM处输出。</p>
操作停止	<p>将TTn位设置为1。——▶</p> <p>由于是触发位，TTn位自动恢复为0。</p> <p>TOEn和TOEm位被清零且数值被设置到TON和TOM位。——▶</p>	<p>TEn=0，并且计数操作停止。</p> <p>TCRn保持计数值并停止。</p> <p>TCRn也保存TSRn寄存器的OVF位。</p> <p>TON输出没有被初始化但保持当前状态并停止。</p> <p>TON和TOM引脚输出TON和TOM的设置值。</p>
备注	<p>n = 01~10</p> <p>m = 02~11</p>	

操作继续

图 7-21. 实时输出功能的操作过程（类型 2）（2/2）



7.4.3 作为 6 相 PWM 输出功能操作

通过扩展 PWM 功能和使用 8 个通道组合使用可产生一个 6 相的 PWM 波形。

总数达 6 个 PWM 输出信号，分别输出来自从通道 2、从通道 3、从通道 4、从通道 5、从通道 6、从通道 7 的信号。从通道 1 可用于任何操作模式。（在此功能中，从通道 1 的操作模式将不被固定。）

输出脉冲的周期和占空比可通过下面的公式计算出。

脉冲周期={TDR00（主通道）的设置值+1}×计数时钟周期

占空比[%]=(TDRm（从设备）的设置值)/(TDR00（主设备）的设置值+1)×100

0%输出: TDRm（从设备）的设置值=0000H

100%输出: TDRm（从设备）的设置值≥TDR00（主设备）的设置值+1

备注 虽然当 TDRm（从设备）的设置值>(TDR00（主设备）的设置值+1)时占空比值会超出 100%，但是，仍作为 100%输出。

主通道的 TCR00 运行于间隔定时器模式，计数周期。

在 6 相 PWM 输出功能中，从通道 1 的操作模式将不被固定，可以被自由设置。

（使用调制输出功能的话，从通道 1 作为实时输出触发产生通道使用。）

从通道 2-7 的 TCRm 运行于一次计数模式，计数占空比，并且从 T0m 引脚输出 PWM 波形。TDRm 的值加载到 TCRm，主通道的 INTTM00 作为启动触发，开始向下计数。当 TCRm=0000H 时，TCRm 输出 INTTMm，并且停止计数直到下一个启动触发（主通道的 INTTM00）被输入。T0m 的输出电平在主通道产生 INTTM00 后的一个计数时钟后开始有效，TCRm=0000H 时变为无效。

主通道的 TDR00 和从通道的 TDRm 在下一周期有效（主通道的 INTTM00 产生）。

注意事项 1. 要重写主通道的 TDR00 和从通道 2~7 的 TDRm，写访问必须至少执行两次。因为在主通道的 INTTM00 产生后 TDR00 和 TDRm 的值赋给 TCR00 和 TCRm，如果重写在主通道的 INTTM00 发生之前和之后分别执行，T0m 引脚可能无法输出预期波形。要重写主通道的 TDR00 和从通道的 TDRm，必须在主通道的 INTTM00 产生后立即重写这两个寄存器。

2. 当 TE00 = 1 或 TEn = 1 时，TS00 或 TSm 不能设为“1”（强制重启）。当 TE00 = 1 或 TEm = 1 时，如果 TS00 或 TSm 设为“1”，0 计数器的值（TCR00 或 TCRm）将为非法且 T0m 不能输出预期的波形。

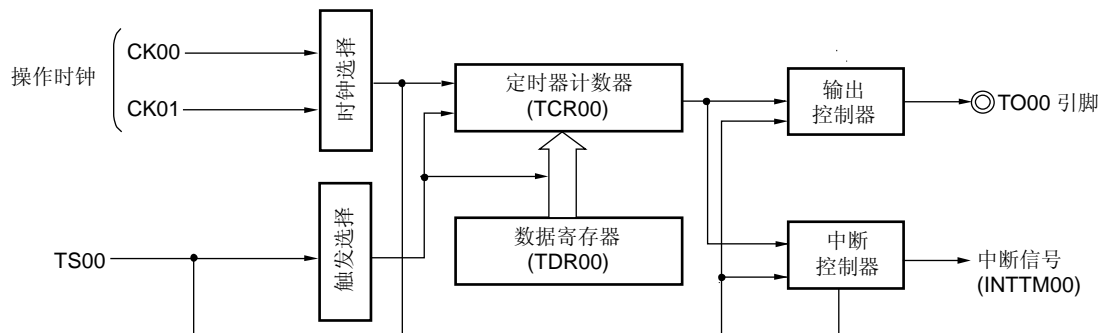
备注 m = 02~07

<R>

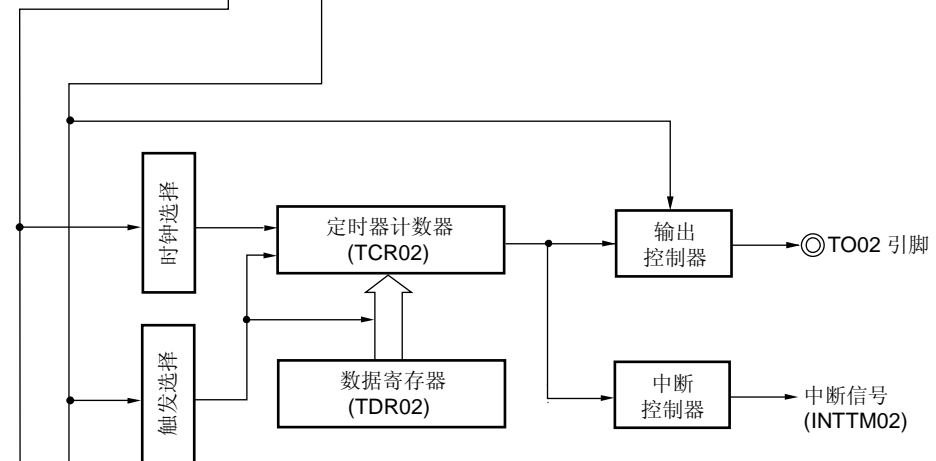
图 7-22. 作为 6 相 PWM 输出功能操作的框图

主通道

(间隔定时器模式)

从通道2

(一次计数模式)

从通道7

(一次计数模式)

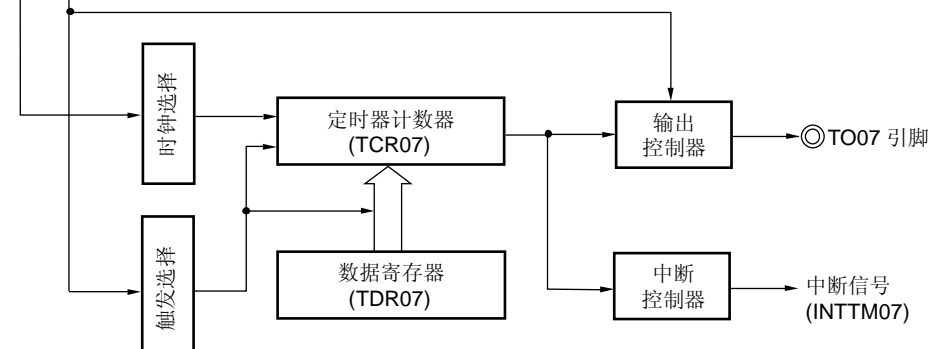


图 7-23. 作为 6 相 PWM 输出功能操作的基本时序示例

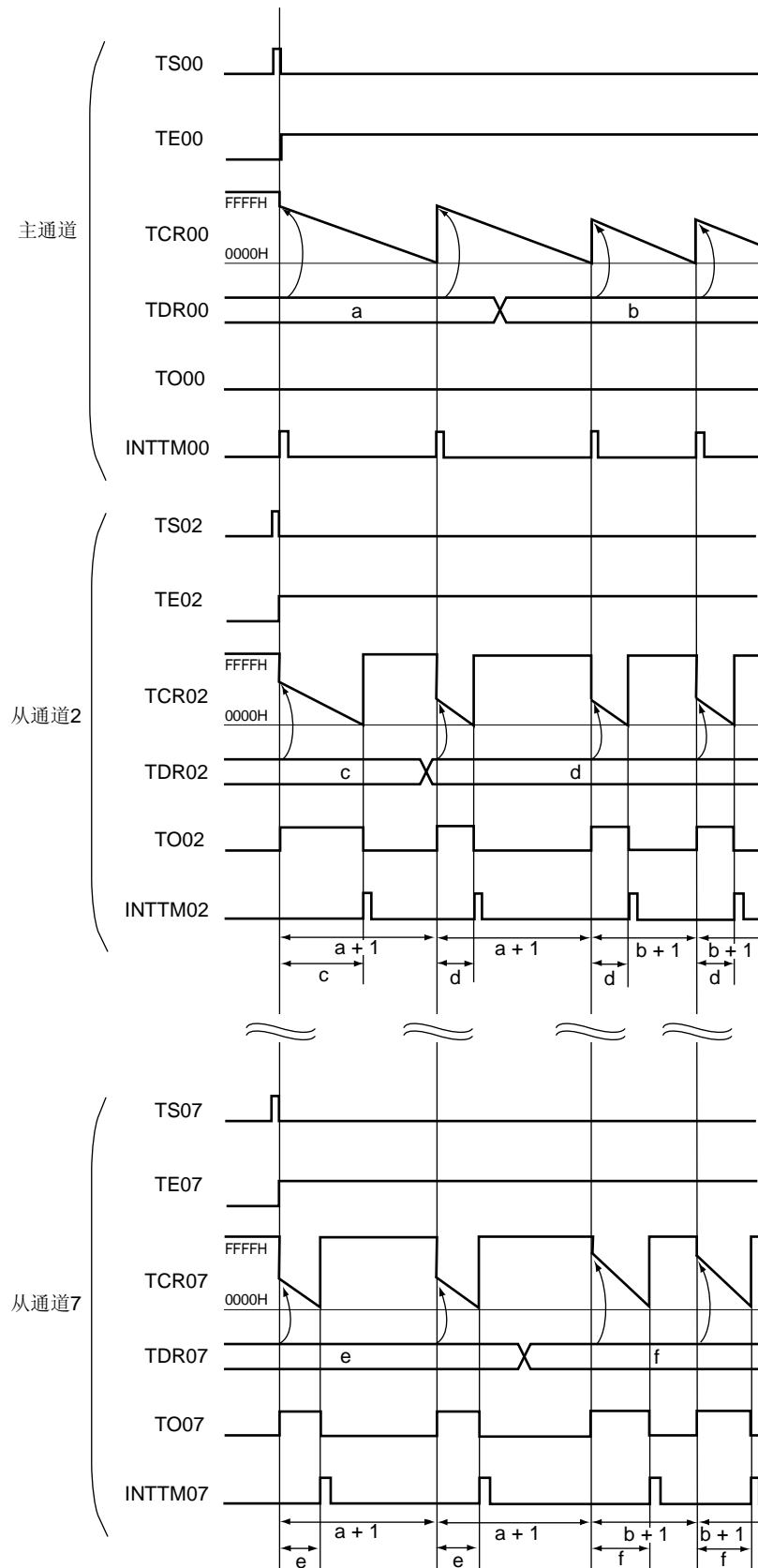
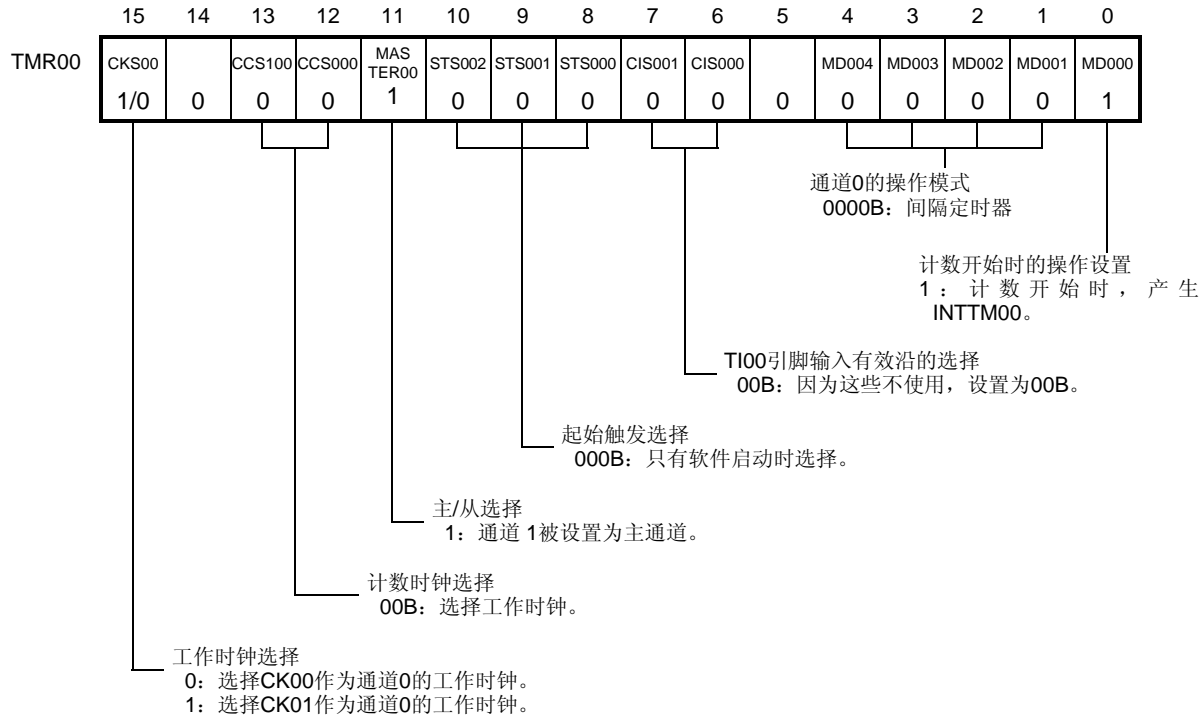


图 7-24. 当使用 6 相 PWM 输出功能（主通道）时寄存器设置的示例

(a) 定时器模式寄存器 00 (TMR00)

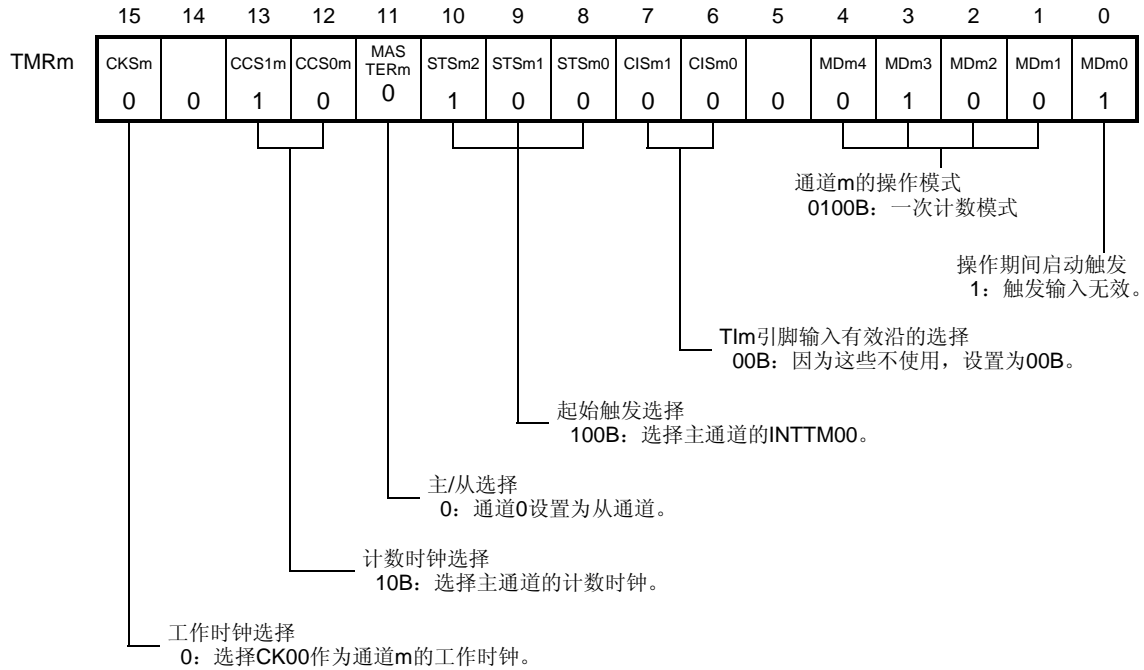


(b) 其它寄存器

TOE0:TOE00	0: 通过计数操作停止 TO00 输出操作。
TO0:TO00	0: 从TO00输出低电平。
TOM0:TOM00	0: 当TOE00 = 0时设为0 (通过计数操作停止TO00输出)。
TOT0:TOT00	0: 当TOM00 = 0(反转模式)时设为0。
TOL0:TOL00	0: 当TOM00 = 0(反转模式)时设为0。
TDE0:TDE00	0: 停止死区时间控制。
TRE0:TRE00	0: 停止实时输出。
TRO0:TRO00	0: 当TRE00=0 (停止实时输出) 时设为0。
TRC0:TRC00	0: 作为实时输出触发产生通道不运行。
TME0:TME00	0: 停止调制输出。

图 7-25. 当使用 6 相 PWM 输出功能（从通道 2~7）时寄存器设置的示例

(a) 定时器模式寄存器 m (TMRm)



(b) 其它寄存器

TOE0:TOEm	0: 通过计数操作, 停止T0m输出操作。 1: 通过计数操作, 允许T0m输出操作。
TO0:T0m	0: 从T0m输出低电平。 1: 从T0m输出高电平。
TOM0:TOMm	1: 设置组合操作模式。
TOT0:TOTm	0: 产生不同于三角波PWM输出。
TOL0:TOLm	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	0: 停止实时输出。
TRO0:TROm	0: 当 TREm = 0时设为0 (停止实时输出)。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。

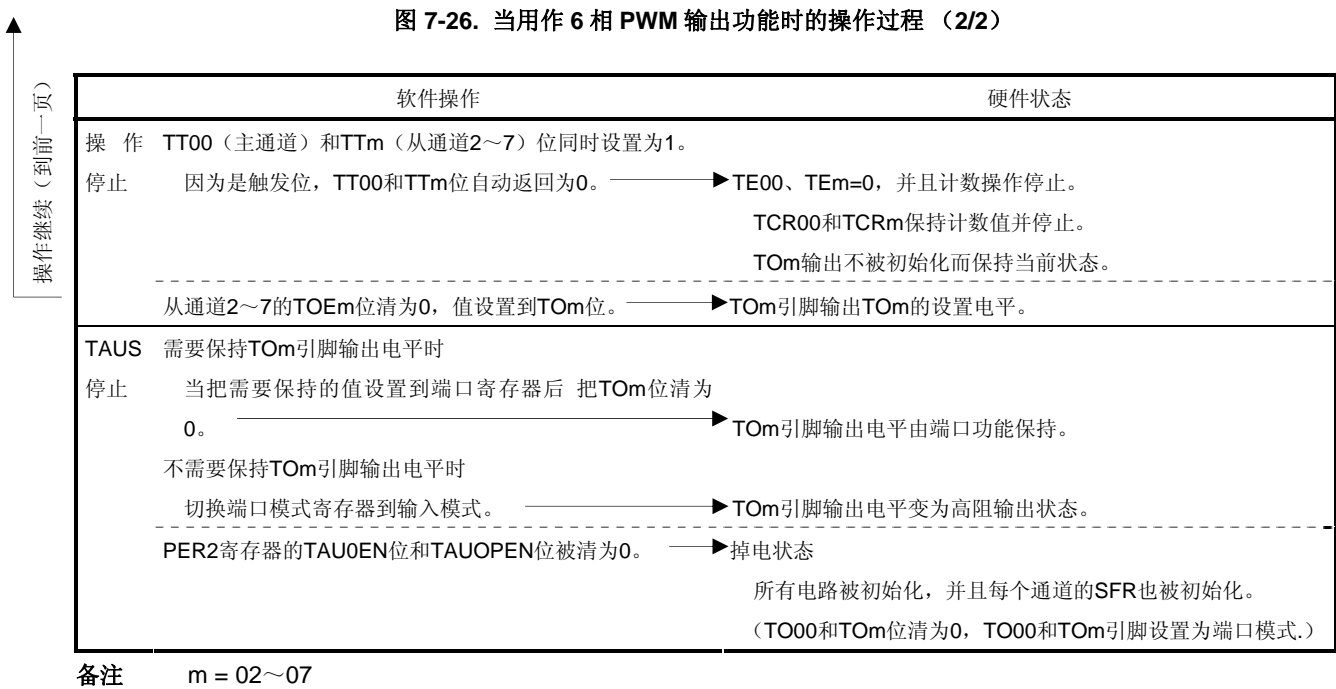
备注 m = 02~07

图 7-26. 当用作 6 相 PWM 输出功能时的操作过程 (1/2)

软件操作		硬件状态
TAUS		掉电状态
默认		(停止供给时钟, 并且所有寄存器不能写入。)
设置	设置PER2寄存器的TAU0EN位和TAUOPEN位为1。——▶	上电状态。每个通道停止操作。
		(开始供给时钟并且每个寄存器允许写入。)
	设置TPS0 寄存器。	
	决定CK00和CK01的时钟频率。	
通道	设置要使用的每个通道的TMR00和TMRm寄存器 (决定通	通道停止操作。
默认	道的操作模式)。	(供给时钟并且产生一些功耗。)
设置	间隔 (周期) 值设置到主通道的TDR00寄存器, 占空比设	
	置到从通道2~7的TDRm寄存器。	
	设置从通道2~7。	T0m引脚为高阻输出状态。
	设置T0Mm位为1 (组合操作模式)。	
	设置TOTm 位为 0 (产生波形不同于三角波 PWM 输	
	出)。	
	设置TOLm位并决定T0m输出的有效电平。	
	设置T0m位并决定T0m输出的默认电平。——▶	当端口模式寄存器在输出模式下, 并且端口寄存器为0时, T0m
		输出默认设置电平。
	设置TOEm位为1并使能T0m操作。——▶	因为通道停止工作T0m不改变。
	将端口寄存器和端口模式寄存器清为0。——▶	T0m引脚输出T0m的设置电平。
操作	设置TOEm (从通道2~7) 为1 (仅当操作恢复时)。	
开始	TS0寄存器的TS00 (主通道) 和TSm (从通道2~7) 位同时设置为1。	TE00=1、TEm=1
	因为是触发位, TS00和TSm位自动返回为0。	当主通道开始计数, 产生INTTM00。因为该中断触发, 从通
		道2~7也开始计数。
操作	TDR00和TDRm寄存器的设置值在主通道的INTTM00产生	主通道的计数器加载TDR00的值到TCR00, 并向下计数。当计
期间	后可被修改。	数值达到TCR00=0000H, 产生INTTM00。同时, TDR00寄存
	TCR00和TCRm寄存器总是可读。	器的值加载到TCR00, 计数器开始再次向下计数。
	TOL0、T00 和TOE0 寄存器的设置值能修改。	在从通道2~7, TDRm寄存器的值发送到TCRm, 由主通道的
		INTTM00触发, 计数器开始向下计数。T0m的输出电平在主通
		道输出INTTM00后的一个计数时钟开始有效。当TCRm=0000H
		时变为无效, 并且计数操作停止。之后, 重复上面的操作。

备注 m = 02~07

图 7-26. 当用作 6 相 PWM 输出功能时的操作过程（2/2）



7.4.4 作为三角波PWM输出功能操作

多重通道可组合使用来输出用于马达控制的三角波 PWM。

由主通道设置周期，由从通道输出三角波 PWM。当一个周期输出多重三角波 PWM 时，可通过增加从通道来增加三角波 PWM 输出。

输出脉冲的周期和占空比可通过下面的公式计算出。

脉冲周期（上/下）= {TDR00（主通道）的设置值+1}×2×计数时钟周期

占空比[%]={TDR00（主设备）的设置值+1-TDRm（从设备）的设置值}/{TDR00（主设备）的设置值+1}×100

0%输出: TDRm（从设备）的设置值≥{TDR00（主设备）的设置值+1}

100%输出: TDRm（从设备）的设置值=0000H

备注 虽然当 TDRm（从设备）的设置值>{TDR00（主设备）的设置值+1}时占空比会超出 0%，但是，仍然作为 0%输出。

主通道操作在间隔定时器模式并计数周期。

通道启动触发位（TS00）设置为 1 后，TCR00 在第一个计数时钟加载 TDR00 的值。此时，当 TMR00 的 MD0n=0 时 INTTM00 不输出且 TO00 不反转。当 TMRn 的 MDn=1 时 INTTM00 输出且 TO00 反转。

之后，TCRn 随着计数时钟而向下计数。

当 TCR00 变为 0000H 时，INTTM00 输出且 TO00 在下一个计数时钟反转。此时 TCR00 再次加载 TDR00 的值。此后继续相同的操作。

主通道计数的两个周期内产生一个载波周期。

从通道的计数操作通过定义主通道的第一个周期作为从通道的向下状态和第二个周期作为从通道的向上状态进行控制。

主通道的 TO00 输出向上和向下状态。

为了输出向上和向下状态，当 TOE00 寄存器的 TOE0 为 0 时，必须操作 TO00 寄存器的 TO0 且设置默认电平。

TMR00 寄存器的 MD000 为 0 时设置 TO0 的 TO00 为 1；当 MD000 为 1 时设置 TO00 为 0。

通过设置默认电平，在向下状态下一个高电平从 TO00 中输出且在向上状态下输出一个低电平。

从通道的 TCRm 运行在上下计数模式下并计数占空比。

在通道启动触发位（TSm）被设为 1 后，TCRm 在第一个计数时钟下加载 TDRm 的值。此后，根据主通道的操作进行向上计数和向下计数的切换。当 TCRm 变为 0000H 时，将输出 INTTMm。

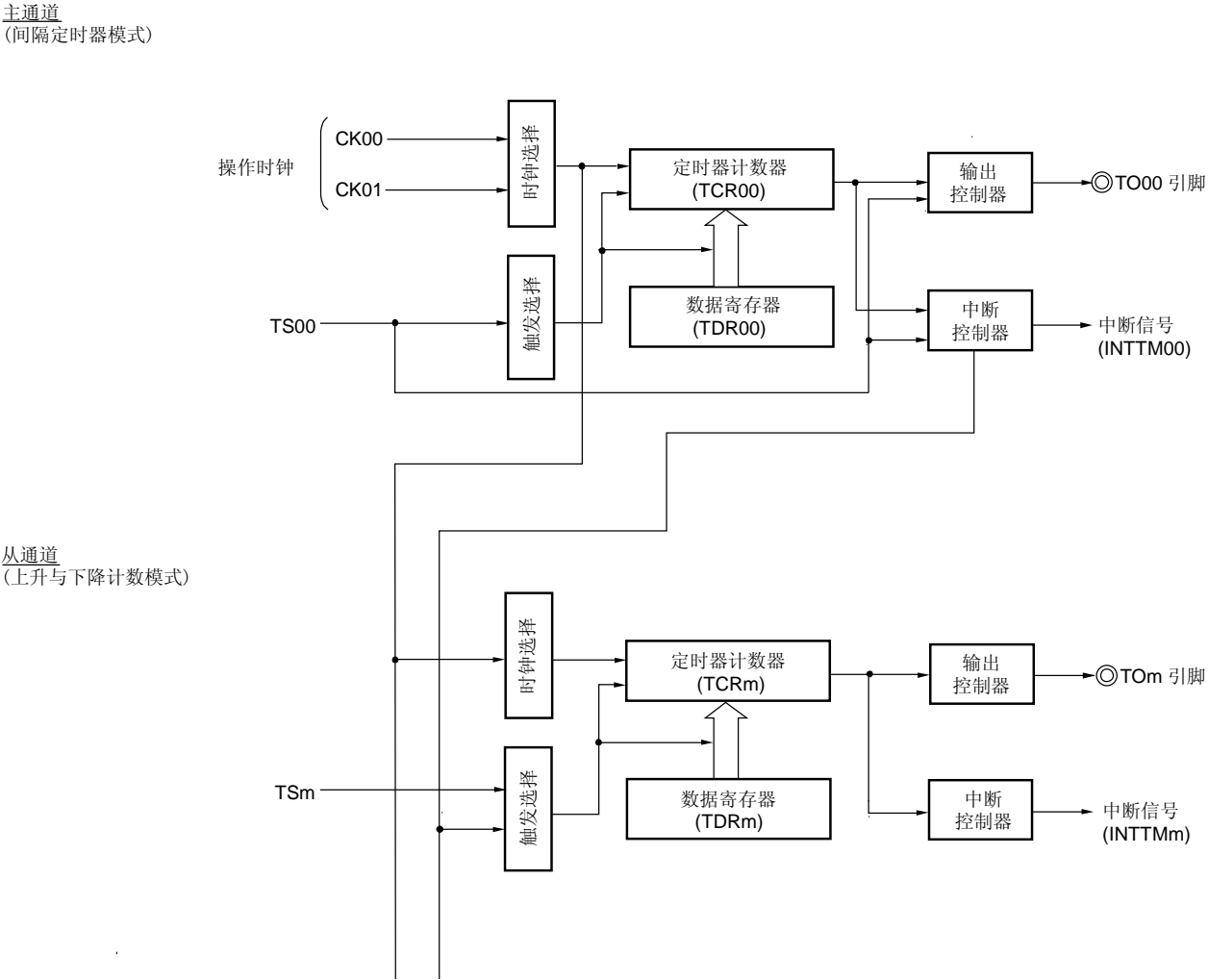
TOm 输出在向下计数期间当 TCRm 产生 INTTMm 时变为有效电平，并且在向上计数期间当 TCRm 产生 INTTMm 时变为无效电平。

在主通道的向上状态下当 INTTM00 产生时，TCRm 再次加载 TDRm 的值。此后继续相同的操作。

备注 m = 02~07

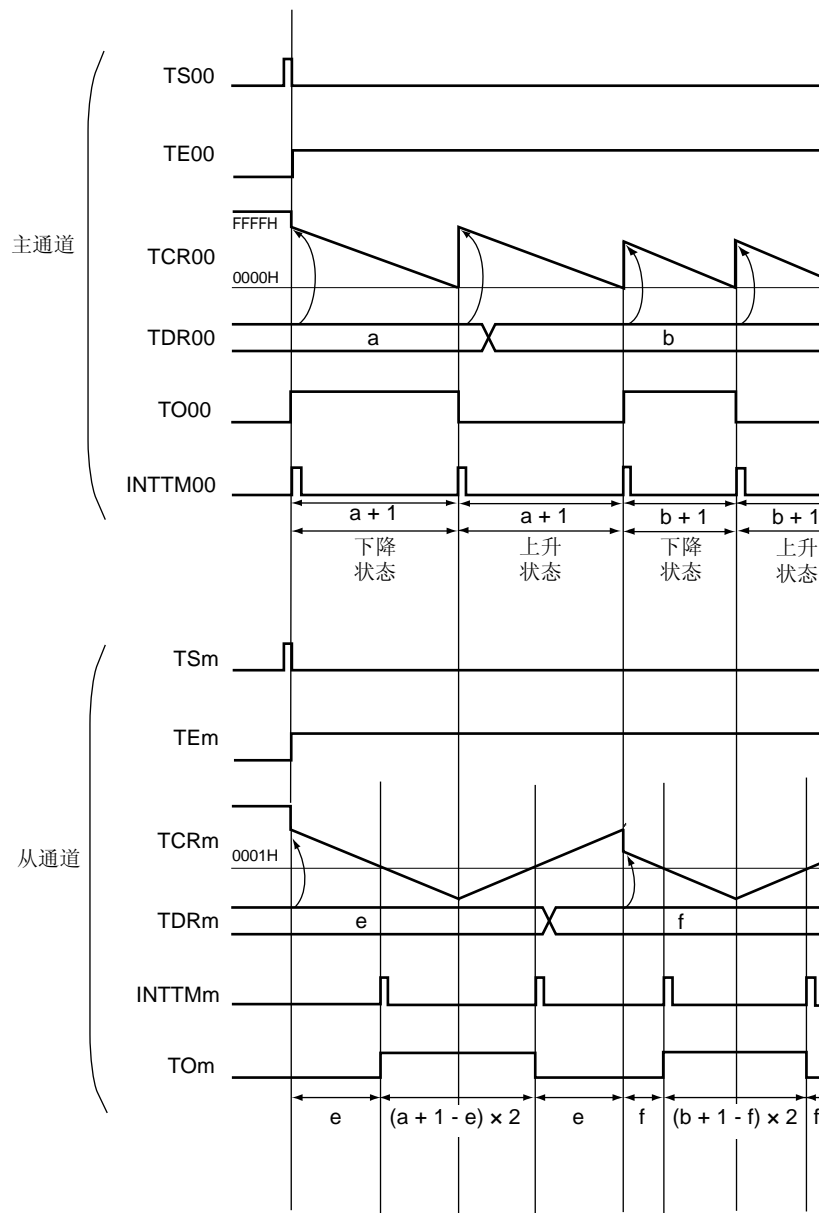
- <R> 注意事项 1. 主通道的 **TDR00** 必须在从通道的递增状态周期重写（计数状态由从通道的 **CSF**（**TSRm** 寄存器）或主通道的 **TO00** 输出电平进行判断）。当 **TDR00** 的值分别在向上或向下状态下重写时，向上状态和向下状态的周期会不同并且 **TO00** 引脚不能输出预期的波形，因为被重写的主通道的 **TDR00** 的值在下一向下状态周期内变为有效。
- <R> 2. 当 **TE00 = 1** 或 **TE_n = 1** 时，**TS00** 或 **TS_m** 不能设为“1”（强制重启）。当 **TE00 = 1** 或 **TE_m = 1** 时，如果 **TS00** 或 **TS_m** 设为“1”，0 计数器的值（**TCR00** 或 **TCR_m**）将为非法且 **TO_m** 不能输出预期的波形。

图 7-27. 作为三角波 PWM 输出功能操作的框图



备注 m = 02~07

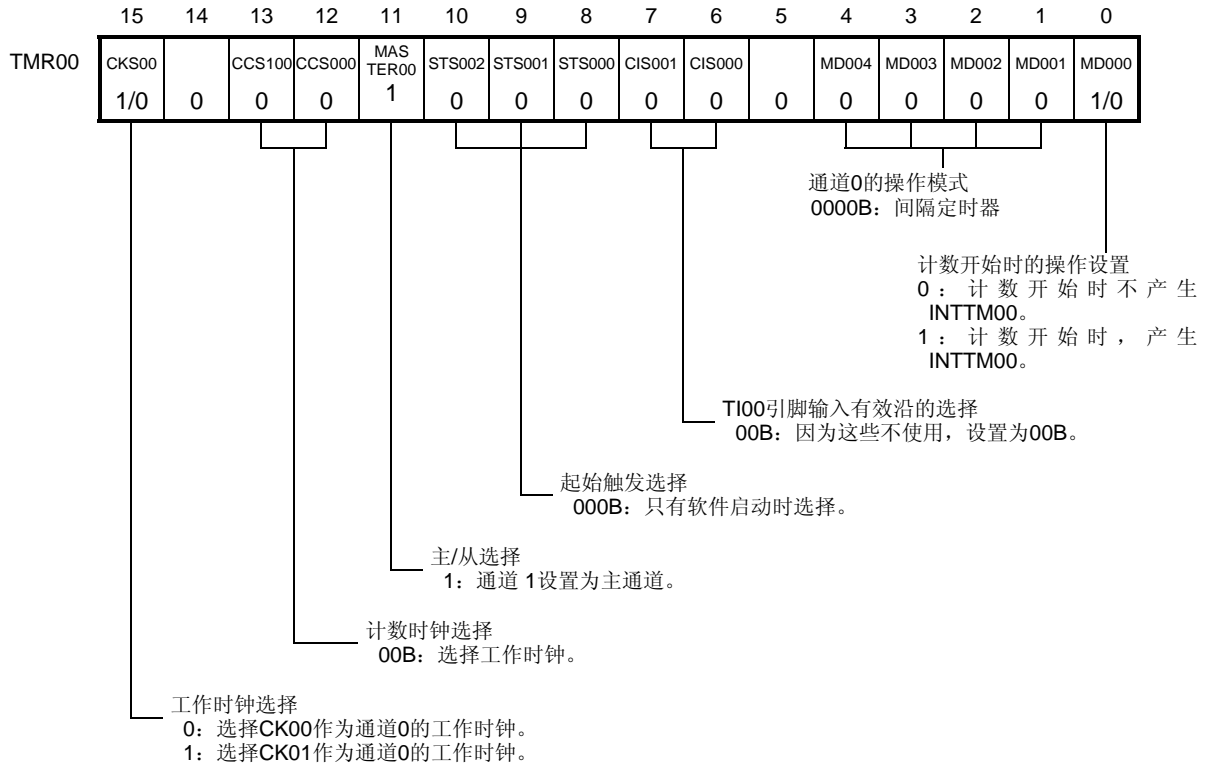
图 7-28. 作为三角波 PWM 输出功能操作的基本时序示例



备注 $m = 02 \sim 07$

图 7-29. 当使用三角波 PWM 输出功能（主通道）时寄存器设置的示例

(a) 定时器模式寄存器 00 (TMR00)



(b) 其它寄存器

<R>

TOE0: TOE00 ^注	0: 通过计数操作停止 TO00 输出操作。 1: 通过计数操作允许TO00输出操作。
TO0:TO00	0: 从TO00输出低电平。 1: 从TO00输出高电平。
TOM0:TOM00	0: 设置反转模式。
TOT0:TOT00	0: 当TOM00 = 0 (反转模式) 时设为0。
TOL0:TOL00	0: 当TOM00 = 0 (反转模式) 时设为0。
TDE0:TDE00	0: 停止死区时间控制。
TRE0:TRE00	0: 停止实时输出。
TRO0:TRO00	0: 当TRE00=0 (停止实时输出) 时设为0。
TRC0:TRC00	0: 作为实时输出触发产生通道不运行。
TME0:TME00	0: 停止调制输出。

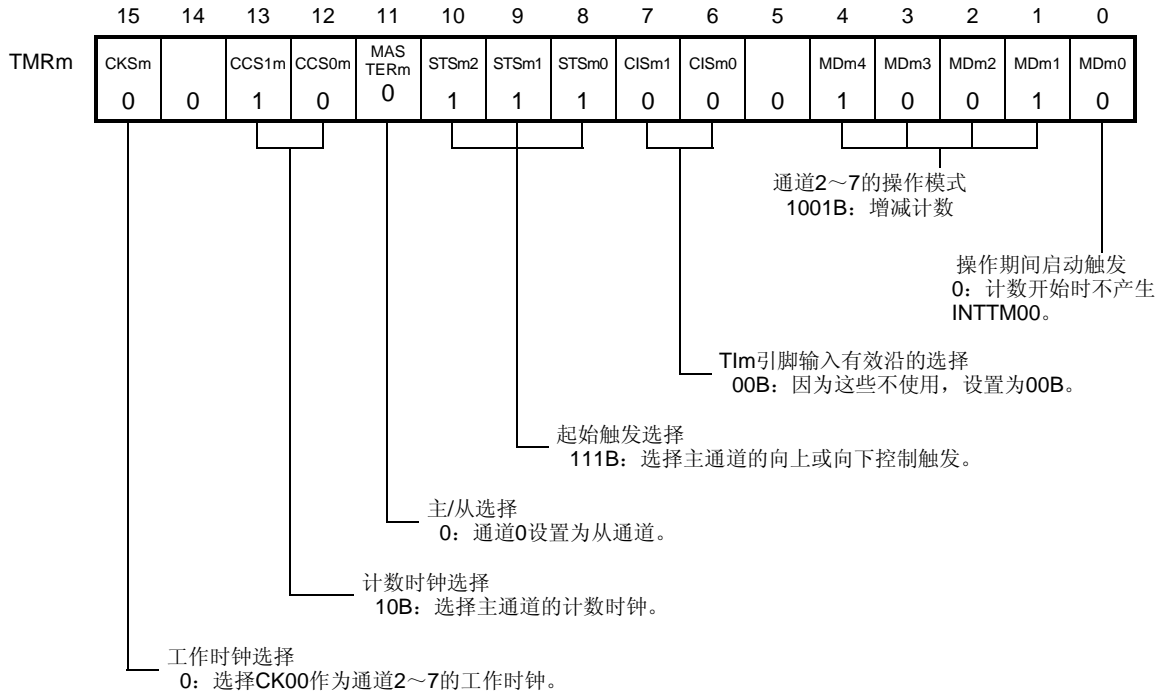
<R>

注 在下列情况中, 设置主通道的 TOE00 至“1”。

- 当使用一个 INTTMM0、INTTMV0、INTTMM1 或 INTTMV1 中断信号
- 利用 OPMR、OPHS 或 OPHT 寄存器, 控制一个高阻输出或通过控制器选择一个 A/D 转换触发时

图 7-30. 当使用三角波 PWM 输出功能（从通道）时寄存器设置的示例

(a) 定时器模式寄存器 m (TMRm)



(b) 其它寄存器

TOE0:TOEm	0: 通过计数操作, 停止TOM输出操作。 1: 通过计数操作, 允许TOM输出操作。
TO0:TOM	0: 从TOM输出低电平。 1: 从TOM输出高电平。
TOM0:TOMm	1: 设置组合操作模式。
TOT0:TOTm	1: 设置三角波PWM输出。
TOL0:TOLm	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	0: 停止实时输出。
TRO0:TROm	0: 当 TREm = 0时设为0 (停止实时输出)。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。

备注 m = 02~07

图 7-31. 当用作三角波 PWM 输出功能时的操作过程 (1/2)

软件操作	硬件状态
TAUS 默认设置	掉电状态 (停止供给时钟, 并且所有寄存器不能写入。)
设置PER2寄存器的TAU0EN位和TAUOPEN位为1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
设置TPS0 寄存器。	
决定CK00和CK01的时钟频率。	
通道默认设置	通道停止操作。 (供给时钟并且产生一些功耗。)
设置要使用的每个通道的TMR00和TMRm寄存器 (决定通道的操作模式)。	
间隔 (周期) 值设置到主通道的TDR00寄存器, 占空比设置到从通道的TDRm寄存器。	
设置主通道。	TO00和TOm引脚为高阻输出状态。
设置TOM0寄存器的TOM00位为0 (反转模式)。	
设置从通道。	
设置TOM0寄存器的TOMm位为1 (组合操作模式)。	
设置TOT0寄存器的TOTm位为1 (三角波 PWM 输出)。	
设置TOLm位为0 (正向逻辑输出)。	
设置TOM位并决定TOM输出的默认电平。	当端口模式寄存器在输出模式下, 并且端口寄存器为0时, TOn输出默认设置电平。
设置TOE00和TOEm位为1并使能TO00和TOm操作。	TO00和TOm不改变因为通道停止工作。
将端口寄存器和端口模式寄存器清为0。	TO00和TOm引脚输出TO00和TOm的设置值。
操作开始	设置TOE00 (主通道) 和TOEm (从通道) 位为1 (仅当操作恢复时)。
TS0寄存器的TS00 (主通道) 和TSm (从通道) 位同时设置为1。	TE00=1、TEm=1
因为是触发位, TS00和TSm位自动返回为0。	当主通道和从通道开始计数且TMR00寄存器的MD000位为1时, 产生INTTM00。

备注 m = 02~07

操作继续 (从下一页)

图 7-31. 当用作三角波 PWM 输出功能时的操作过程 (2/2)

操作继续。(到前一页)

软件操作	硬件状态
操作期间 TDR00（主通道）寄存器的设定值必须在从通道p的向上状态周期下被改变。 TDRm（从通道）寄存器的设置值能改变。 TCR00和TCRm寄存器总是可读。 TSRm（从通道）寄存器总是可读。	在主通道下会产生一个周期且从通道的计数操作被控制。 TCR00 加载 TDR00 的值并向下计数。当计数值达到 TCR00=0000H，产生INTTM00。同时，TDR00寄存器的值加载到TCR00，计数器开始再次向下计数。 在从通道下主通道的INTTM00作为触发使用来转换向下计数和向上计数。 INTTMm根据TCRm=0001H的检测和TOm输出三角波PWM而产生。 在主通道下TCR00再次加载TDR00的值并通过向上状态下产生INTTM00来继续计数操作。 之后，重复上面的操作。
操作停止 TT00（主通道）和TTm（从通道）位同时设置为1。因为触发位，TT00和TTm位自动返回为0。	TE00、TEm=0，并且计数操作停止。 TCR00和TCRm保持计数值并停止。 TO00和TOm输出不被初始化而保持当前状态。
TOE00 和 TOEm 位 被 清 零 且 数 值 被 设 置 到TO00和TOm位。	TO00和TOm引脚输出TO00和TOm的设置值。
TAUS 停 止 需要保持TO00和TOm引脚输出电平时 当把需要保持的值设置到端口寄存器后把TO00和TOm位清为0。 不需要保持TO00和TOm引脚输出电平时 切换端口模式寄存器到输入模式。 PER2寄存器的TAU0EN位和TAUOPEN位被清为0。	TO00和TOm引脚输出电平通过端口功能保持。 TO00和TOm引脚输出电平变为高阻输出状态。 掉电状态 所有电路被初始化，并且每个通道的SFR也被初始化。 (TO00和TOm位清为0，TO00和TOm引脚设置为端口模式.)

备注 m = 02~07

7.4.5 作为带有死区时间的三角波PWM输出功能操作

带有死区时间的三角波 PWM 输出功能使用 4 个通道组合(通道 0~3 或通道 4~7)，来输出一个三角波 PWM 的波形(带有死区时间)。

从通道 2 和 3，以及从通道 6 和 7 输出的带有死区时间的三角波 PWM 输出信号。从通道 1 和 5 可在任何操作模式下工作。(在此功能中，从通道 1 和 5 的操作模式将不被固定。)

输出脉冲周期、占空比(正相)和占空比(反相)可通过下面的公式计算出。

$$\begin{aligned}\text{脉冲周期(增/减)} &= \{\text{TDRn(主通道)的设置值}+1\} \times 2 \times \text{计数时钟周期} \\ \text{占空比(正相)[\%]} &= \{\{\text{TDRn(主通道)的设置值}+1\} - \{\text{TDRp(从通道 p)的设置值}\} \times 2 - \{\text{TDRq(从通道 q)的设置值}+1\}\} \times \text{计数时钟周期} \\ \text{占空比(反相)[\%]} &= \{\{\text{TDRn(主通道)的设置值}+1\} - \{\text{TDRp(从通道 p)的设置值}\} \times 2 + \{\text{TDRq(从通道 q)的设置值}+1\}\} \times \text{计数时钟周期}\end{aligned}$$

当死区时间功能被使用时，输出波形中将包含误差。正相波的输出宽度将按死区时间的长度缩短，而反相波的输出宽度将按死区时间的长度增加。因此由于误差，0%和 100%附近的输出变换将不再为线性。

$$\begin{aligned}0\% \text{输出: } & \text{TDRp(从设备 p)的设置值} \geq \text{TDRn(主设备)的设置值}+1 \\ 100\% \text{输出: } & \text{TDRp(从设备 p)的设置值} = 0000\text{H}\end{aligned}$$

使用通道 0 和 4 作为主通道。

TCRn 在间隔定时器模式下作为向下计数器操作。

通道启动触发位(TSn)设置为 1 后，TCRn 在第一个计数时钟加载 TDRn 的值。此时，当 TMRn 的 MDn0=0 时 INTTMn 不输出且 TOn 不反转。当 TMRn 的 MDn0=1 时 INTTMn 输出且 TOn 反转。

之后，TCRn 随着计数时钟而向下计数。

当 TCRn 变为 0000H 时，INTTMn 输出且 TOn 在下一个计数时钟反转。此时 TCRn 再次加载 TDRn 的值。此后继续相同的操作。

主通道计数的两个周期内产生一个载波周期。

从通道的计数操作通过定义主通道的第一个周期作为从通道的向下状态和第二个周期作为从通道的向上状态进行控制。

主通道的 TOn 输出向上和向下状态。

为了输出向上和向下状态，当 TOE0 寄存器的 TOEn 为 0 时，必须操作 TO0 寄存器的 TOn 且设置默认电平。

当 TMRn 寄存器的 MDn0=0 时，设置 TO0 寄存器的 TOn 为 1，以及当 MDn0=1 时，设置 TOn 为 0。

通过设置默认电平，在向下状态下一个高电平从 TO00 中输出且在向上状态下输出一个低电平。

备注 n = 00、04
 p = 02、06
 q = 03、07

从通道 1 和 5 不用作带有死区时间的三角波 PWM 输出功能

通过使用从通道 2、6 和从通道 3、7 的组合，来控制死区时间。带有死区时间的三角波 PWM 输出功能使用主通道 0、从通道 2、从通道 3、主通道 4、从通道 6 以及从通道 7。

从通道 2、6 的 TCRp 运行在上下计数模式下并计数占空比。在通道启动触发位 (TSp) 被设为 1 后，TCRp 在第一个计数时钟下加载 TDRp 的值。此后，向上计数和向下计数由主通道的操作进行切换。当 TCRp 变为 0001H 时，将输出 INTTMp。

当在主通道的向上状态下发生 INTTMn 时，TCRp 再次加载 TDRp 的值。此后继续相同的操作。

从通道 3、7 的 TCRq 运行于一次计数模式，并计数死区时间。

<R> TCRq 加载 TDRq 的值，并通过使用计数开始计时和从通道 2、6 的 INTTMp 作为启动触发器进行向下计数。当

<R> TCRq = 0000H 时，输出 INTTMq 并且停止计数直到检测到下一个启动触发输入（从通道 2、6 的 INTTMp）。不能使用从通道 3、7 的 INTTMq，因为发生在载波周期内的次数号码不能被指定（0~3 次）。

通过从通道 2、6（占空）和从通道 3、7（死区时间）的计数操作（INTTMp、INTTMq）改变 TOP 和 TOq，并输出带有死区时间的三角波 PWM 波形。通过控制从通道 2、6 和从通道 3、7 的 TOL0 寄存器的 TOLp 和 TOLq 位，来输出正相波形和反相波形。

由 TCRp 寄存器在向下计数时产生的 INTTMp 作为启动触发器的从通道 3、7 产生的 INTTMq，作为 TOP（TOLp=0）的设置条件。在 TCRp 向上计数时从通道 2、6 产生的 INTTMp，作为 TOP（TOLp=0）的复位条件。

在 TCRp 向下计数时产生的 INTTMp 作为 TOq（TOLq=1）的设置条件。由 TCRp 寄存器在向下计数时产生的 INTTMp 作为启动触发器的从通道 3、7 产生的 INTTMq，作为 TOq（TOLq=0）的复位条件。

<R> 如果 TOP 和 TOq 的设置条件和复位条件冲突，设置条件优先。

通过设置 TOLp 和 TOLq，带死区的 PWM 波形可在从通道 2、6 和 3、7 的正相和负相之间切换。

<R> 注意事项 1. 主通道的 TDRn 必须在从通道 2 和 6 的递增状态周期下重写（计数状态由从通道的 CSF（TSRp 寄存器）或主通道的 TOn 输出电平进行判断）。当向下状态期间 TDRn 的值重写，向下和向上状态期间不同且不能输出一个预期波形，因为主通道被重写的 TDRn 的值在下一周期会变为有效。

<R> 2. 当 TEn = 1、TEp = 1 或 TEq = 1 时，TSn、TSp、或 TSq 不能设为“1”（强制重启）。当 TEn = 1、TEp = 1 或 TEq = 1 时，若 TSn、TSp 或 TSq 设为“1”，则计数器的值（TCRn、TCRp 或 TCRq）将非法且 TOn、TOP 或 TOq 将不能输出预期波形。

备注 n = 00、04
p = 02、06
q = 03、07

从通道 2、6 的 TDRp 的值从下一载波周期开始变为有效（向上和向下触发检测）。

从通道 3、7 的 TDRq 的值从下一开始时序变为有效（死区时间控制触发检测）。

<R>

向上状态期间从通道 2、6 的 INTTMp 检测之后，推荐执行从通道 3、7 的 TDRq 重写。

图 7-32. 作为带有死区时间的三角波 PWM 输出功能操作的框图

主通道0、4
(间隔定时器模式)

从通道2、6
(递增与递减计数模式)

从通道3、7
(一次计数模式)

备注 n = 00、04
 p = 02、06
 q = 03、07

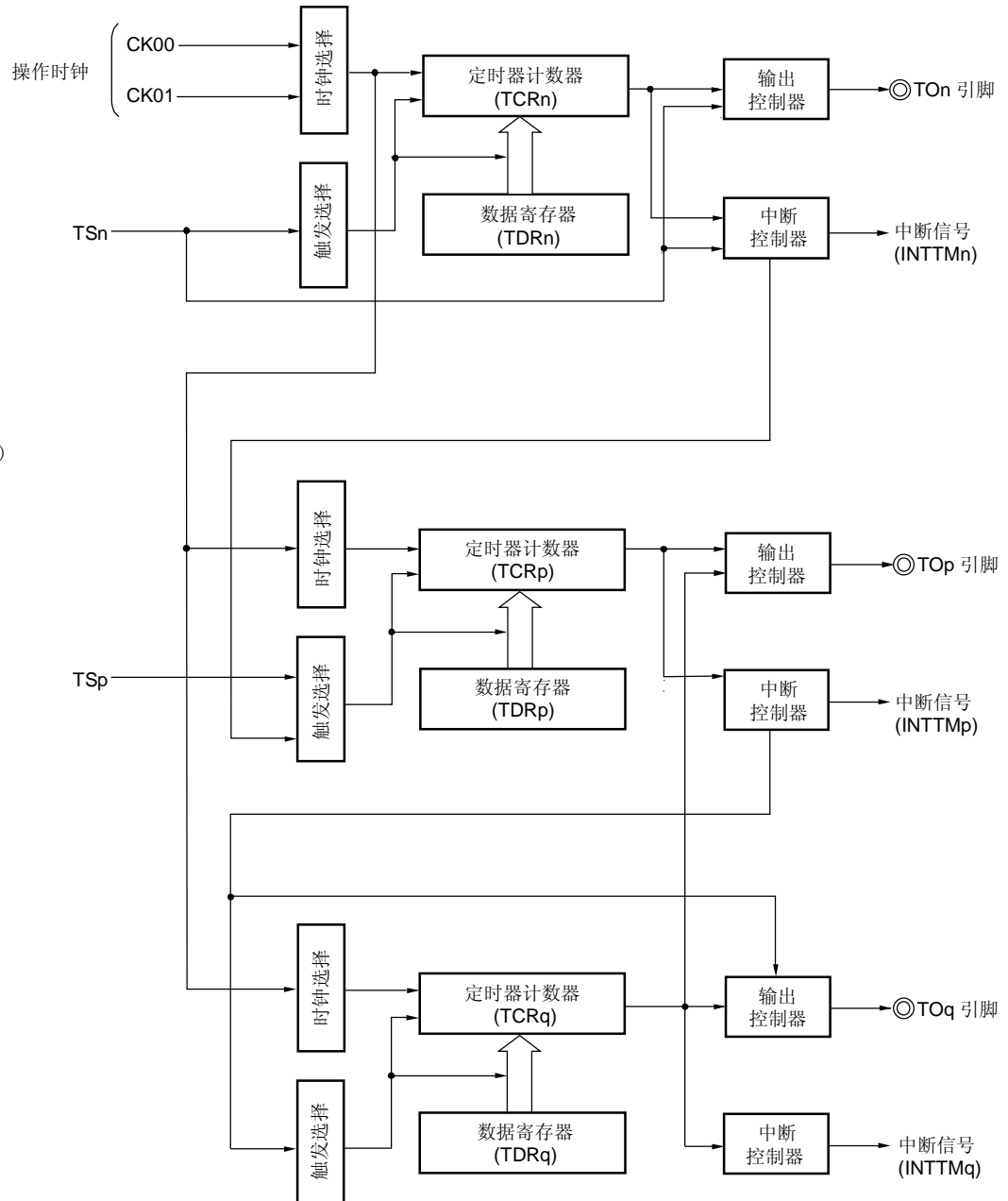
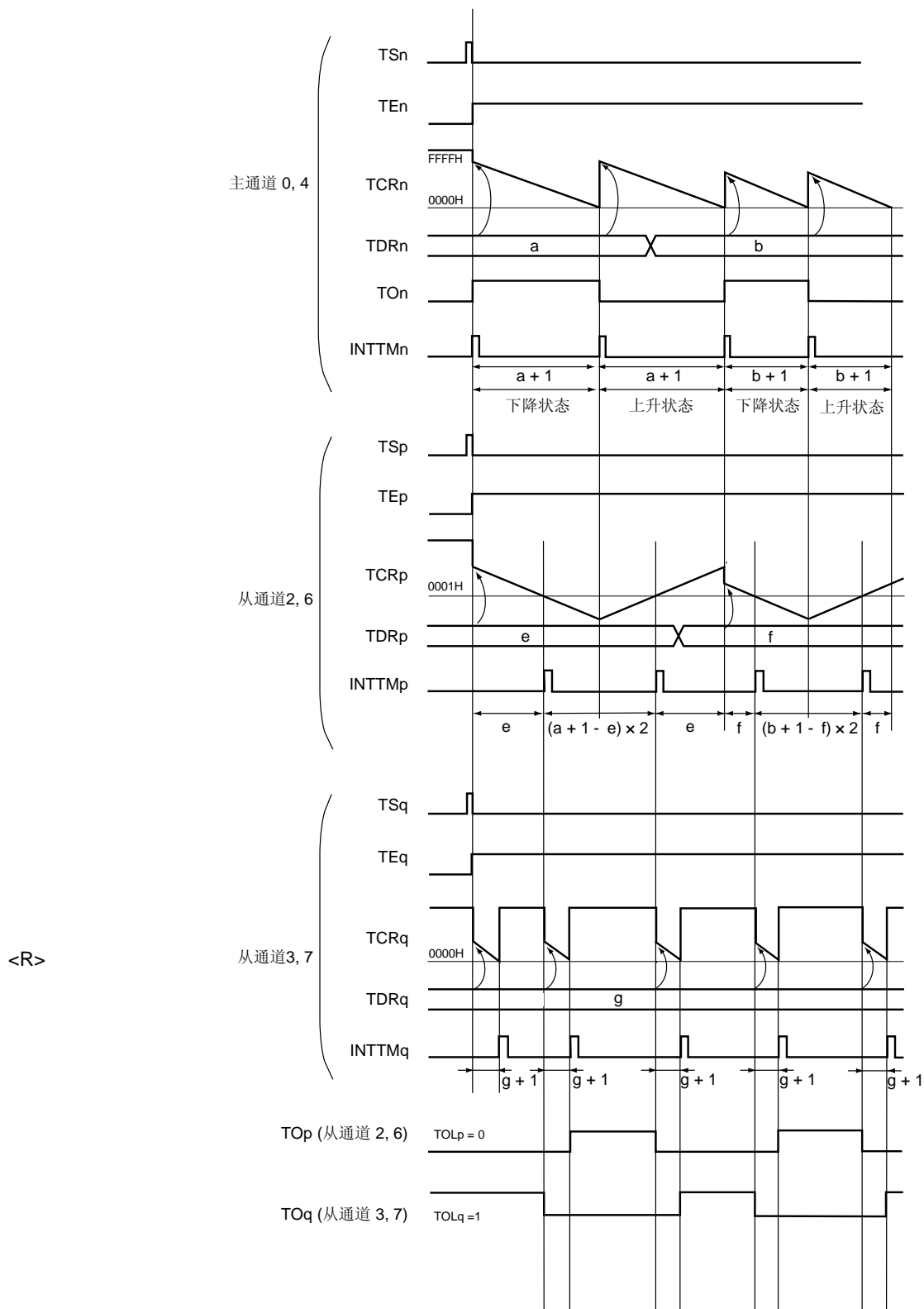


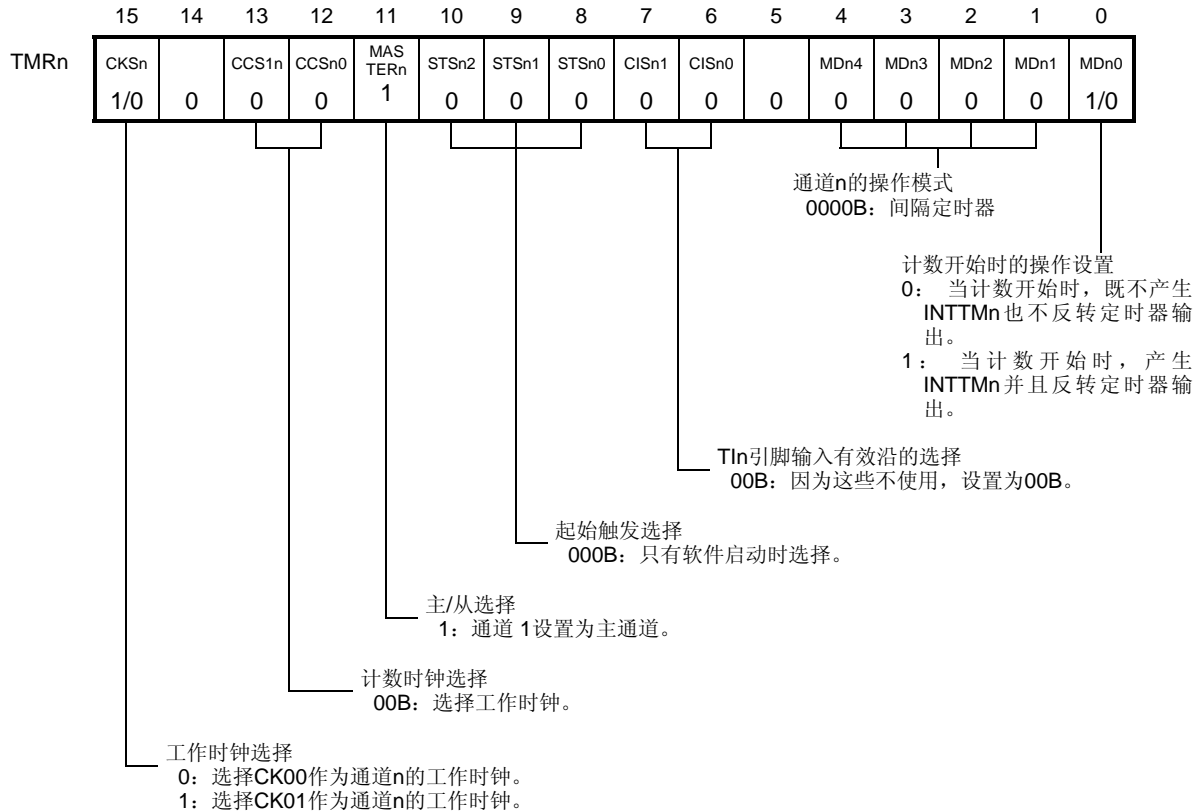
图 7-33. 例如操作基本时序作为寂静时间的三角波 PWM 输出函数



备注 n = 00、04
 p = 02、06
 q = 03、07

图 7-34. 当使用带有死区时间的三角波 PWM 输出功能(主通道 0、4)时寄存器设置内容的示例

(a) 定时器模式寄存器 n(TMRn)



(b) 其它寄存器

<R>

TOE0: TOEn ^注	0: 通过计数操作, 停止TOn输出。 1: 通过计数操作, 允许TOn输出。
TO0:TOn	0: 从TOn输出低电平。 1: 从TOn输出高电平。
TOM0:TOMn	0: 设置反转模式。
TOT0:TOTn	0: 当TOMn=0(反转模式)时设为0。
TOL0:TOLn	0: 当TOMn=0(反转模式)时设为0。
TDE0:TDEn	0: 停止死区时间控制。
TRE0:TREn	0: 停止实时输出。
TRO0:TROn	0: 当 TREn = 0时设为0(停止实时输出)。
TRC0:TRCn	0: 作为实时输出触发产生通道不运行。
TME0:TMEn	0: 停止调制输出。

<R>

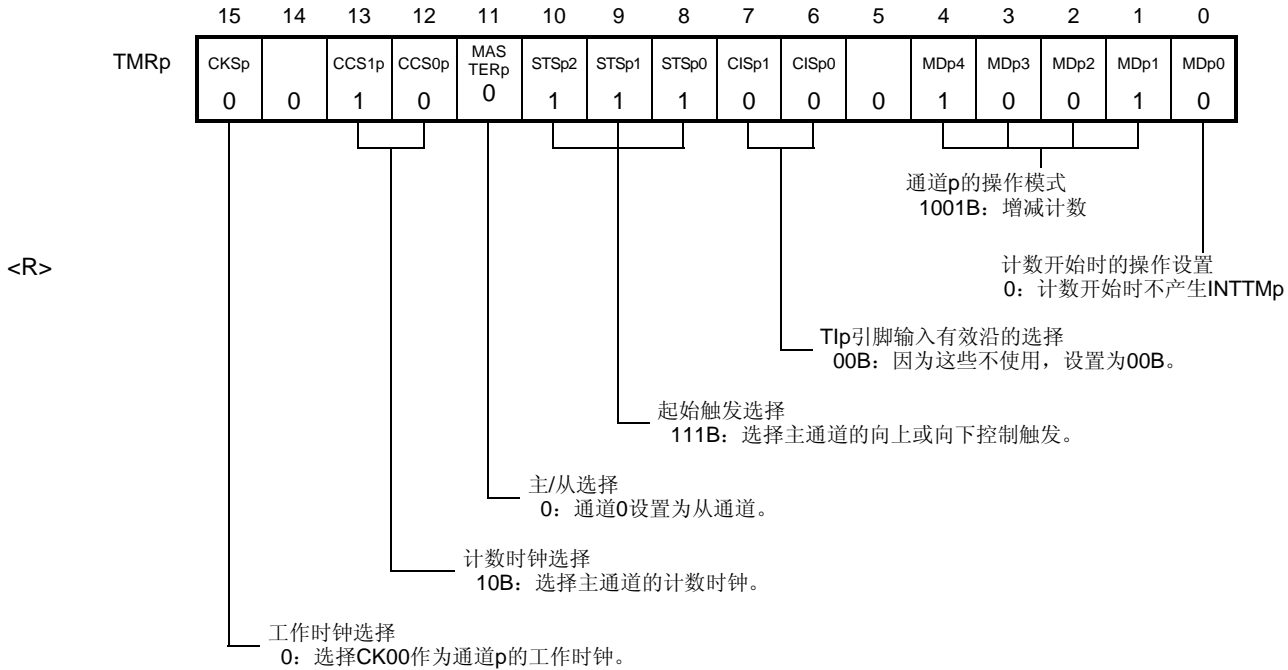
注 在下列情况中, 设置主通道的 TOEn 至“1”。

- 当使用一个 INTTMM0、INTTMV0、INTTMM1 或 INTTMV1 中断信号
- 利用 OPMR、OPHS 或 OPHT 寄存器, 控制一个高阻输出或通过控制器选择一个 A/D 转换触发时

备注 n = 00、04

图 7-35. 当使用带有死区时间的三角波 PWM 输出功能(从通道 2、6)时寄存器设置内容的示例

(a) 定时器模式寄存器 p(TMRp)



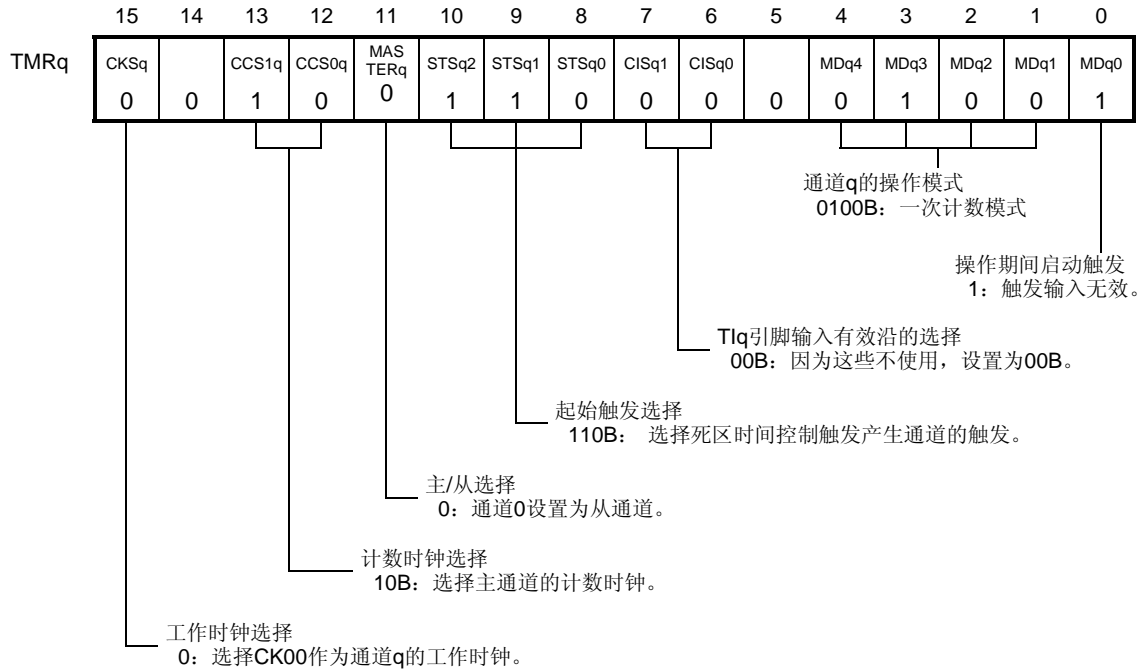
(b) 其它寄存器

TOE0:TOEp	0: 通过计数操作, 停止 TOp 输出操作。 1: 通过计数操作, 允许 TOp 输出操作。
TO0:TOp	0: 从 TOp 输出低电平。 1: 从 TOp 输出高电平。
TOM0:TOMp	1: 设置组合操作模式。
TOT0:TOTp	1: 产生三角波PWM输出。
TOL0:TOLp	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)
TDE0:TDEp	1: 允许死区时间控制。
TRE0:TREp	0: 停止实时输出。
TRO0:TROp	0: 当 TREp = 0时设为0(停止实时输出)。
TRC0:TRCp	0: 作为实时输出触发产生通道不运行。
TME0:TMEp	0: 停止调制输出。

备注 p = 02、06

图 7-36. 当使用带有死区时间的三角波 PWM 输出功能(从通道 3、7)时寄存器设置内容的示例

(a) 定时器模式寄存器 q(TMRq)



(b) 其它寄存器

TOE0:TOEq	0: 通过计数操作, 停止TOq输出操作。 1: 通过计数操作, 允许 TOq 输出操作。
TO0:TOq	0: 从TOq输出低电平。 1: 从TOq输出高电平。
TOM0:TOMq	1: 设置组合操作模式。
TOT0:TOTq	1: 产生三角波PWM。
TOL0:TOLq	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)
TDE0:TDEq	1: 允许死区时间控制。
TRE0:TREq	0: 停止实时输出。
TRO0:TROq	0: 当 TREq = 0时设为0(停止实时输出)。
TRC0:TRCq	0: 作为实时输出触发产生通道不运行。
TME0:TMEq	0: 停止调制输出。

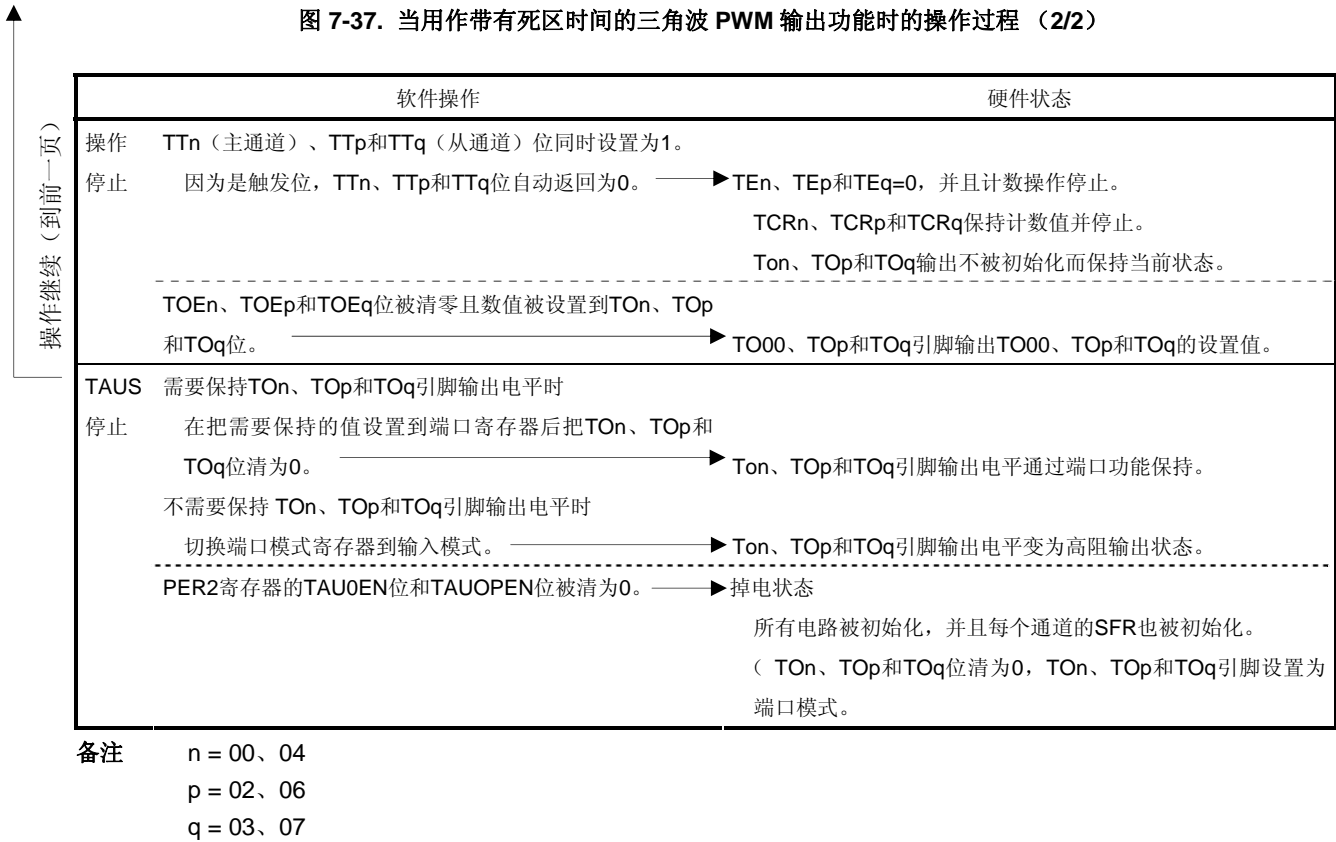
备注 q = 03、07

图 7-37. 当用作带有死区时间的三角波 PWM 输出功能时的操作过程 (1/2)

	软件操作	硬件状态
TAUS 默认 设置	设置PER2寄存器的TAU0EN位和TAUOPEN位为1。 设置TPS0寄存器。 决定CK00和CK01的时钟频率。	掉电状态 (停止供给时钟, 并且所有寄存器不能写入。) 上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
通道 默认 设置	设置要使用的四个通道的TMRn、TMRp和TMRq寄存器 (决定通道的操作模式)。 间隔(周期)值设置到主通道的TDRn寄存器, 占空比设置到从通道2、6的TDRp寄存器, 以及设置死区时间宽度到从通道3、7的TDRq寄存器。 设置从通道。 TOM0寄存器的TOMp和TOMq位以及TOT0寄存器的TOTp和TOTq位被设为1(三角波PWM产生)。 设置TOLp和TOLq位并决定TOp和TOq输出的有效电平。 设置TDEp和TDEq位为1(允许死区时间控制)。	通道停止操作。 (供给时钟并且产生一些功耗。) TO00、TOp和TOq引脚为高阻输出状态。
	设置TON、TOp和TOq位并决定TON、TOp和TOq的默认电平。	当端口模式寄存器在输出模式下, 并且端口寄存器为0时, TON、TOp和TOq输出默认设置电平。
	设置TOEn、TOEp和TOEq位为1并使能TON、TOp和TOq操作。	TON、TOp和TOq不改变因为通道停止工作。
	将端口寄存器和端口模式寄存器清为0。	TON、TOp和TOq引脚输出TON、TOp和TOq的设置值。
操作 开始	设置TOEm(主通道)、TOEp和TOEq(从通道)位为1(仅当操作恢复时)。 TS0寄存器的TSn(主通道)、TSp和TSq(从通道)位同时设置为1。 因为是触发位, TSn、TSp和TSq位自动返回为0。	TEEn=1、TEp=1、TEq=1 当主通道和从通道2、6开始计数且TMRn寄存器的MDn0位为1时, 产生INTTMn。从通道3、7一直等待直到从通道2、6检测出INTTMp。
操作 期间	TDRn(主通道)寄存器的设置值必须在从通道2、6的向上状态周期下被改变。 TDRp和TDRq(从通道)寄存器的设置值不能修改。 TCRn、TCRp和TCRq寄存器总是可读。 TSRp(从通道)寄存器总是可读。	在主通道产生一个周期且从通道的计数操作被控制。在从通道2、6产生一个PWM占空, 在从通道3、7产生死区时间。 通过从通道2、6和从通道3、7的组合操作, 从TOp和TOq引脚输出带有死区时间的三角波PWM波形。
备注	n = 00、04 p = 02、06 q = 03、07	

操作继续 (从下一页)

图 7-37. 当用作带有死区时间的三角波 PWM 输出功能时的操作过程（2/2）



7.4.6 作为6相三角波PWM输出功能操作

6相三角波 PWM 输出功能使用 8 个通道组合来输出一个 6 相三角波 PWM 的波形(带有死区时间)。

6 相三角波 PWM 输出功能是 7.4.5 作为带有死区时间的三角波 PWM 输出功能操作中所描述功能的扩展。

从通道 2、从通道 3、从通道 4、从通道 5、从通道 6 以及从通道 7 输出 6 相三角波 PWM 输出信号。从通道 1 可在任何操作模式下工作。(在此功能中, 从通道 1 的操作模式将不被固定。)

输出脉冲周期、占空比(正相)和占空比(反相)可通过下面的公式计算出。

脉冲周期(上/下)={TDR00 (主通道)的设置值+1}×2×计数时钟周期

占空比(正相)[%]={TDR00(主通道)的设置值+1}-{TDRp(从通道 p)的设置值}×2-{TDRq(从通道 q)的设置值+1}×计数时钟周期

占空比(反相)[%]={TDR00(主通道)的设置值+1}-{TDRp(从通道 p)的设置值}×2+{TDRq(从通道 q)的设置值+1}×计数时钟周期

当死区时间功能被使用时, 输出波形中将包含误差。正相波的输出宽度将按死区时间的长度缩短, 而反相波的输出宽度将按死区时间的长度增加。因此由于误差, 0%和 100%附近的输出变换将不再为线性。

0%输出: TDRp (从设备 p)的设置值≥TDR00 (主设备)的设置值+1

100%输出: TDRp (从设备 p)的设置值=0000H

主通道操作在间隔定时器模式并计数周期。

主通道计数的两个周期内产生一个载波周期。

从通道的计数操作通过定义主通道的第一个周期作为从通道的向下状态和第二个周期作为从通道的向上状态进行控制。

主通道的 TO00 输出向上和向下状态。

为了输出向上和向下状态, 当 TOE00 寄存器的 TOE0 为 0 时, 必须操作 TO00 寄存器的 TO0 且设置默认电平。

TMR0 寄存器的 MD000 为 0 时设置 TO0 寄存器的 TO00 为 1; 当 MD000 为 1 时设置 TO00 为 0。

通过设置默认电平, 在向下状态下一个高电平从 TO00 中输出且在向上状态下输出一个低电平。

从通道 1 不作为 6 相三角波 PWM 输出功能。

(为了使用调制输出功能, 从通道 1 作为实时输出触发产生通道使用。)

通过使用从通道 2、4、6 和从通道 3、5、7 的组合, 来控制死区时间。6 相三角波 PWM 输出功能使用从通道 2 和 3、从通道 4 和 5 以及从通道 6 和 7 的组合。TOp 和 TOq 的输出操作说明如下。

从通道 2、4、6 的 TCRp 运行在上下计数模式下并计数占空比。在通道启动触发位 (TSp) 被设为 1 后, TCRp 在第一个计数时钟下加载 TDRp 的值。此后, 向上计数和向下计数由主通道的操作进行切换。当 TCRp 变为 0001H 时, 将输出 INTTMp。

当在主通道的向上状态下产生 INTTM00 时, TCRp 再次加载 TDRp 的值。此后继续相同的操作。

从通道 3、5、7 的 TCRq 运行于一次计数模式, 并计数死区时间。

<R> TCRq 加载 TDRq 的值, 并通过使用计数开始计时和从通道 2、4、6 的 INTTMp 作为启动触发器进行向下计数。当
<R> TCRq = 0000H 时, 输出 INTTMq 并且停止计数直到检测到下一个启动触发输入 (从通道 3、5、7 的 INTTMq)。不能使用从通道 3、5、7 的 INTTMq, 因为发生在载波周期内的次数号码不能被指定 (0~3 次)。

通过从通道 2、4、6 (占空) 和从通道 3、5、7 (死区时间) 的计数操作 (INTTMp、INTTMq) 改变 TOp 和 TOq, 并输出 6 相三角波 PWM 波形。通过控制从通道 2、4、6 和从通道 3、5、7 的 TOL0 寄存器的 TOLp 和 TOLq 位, 来输出正相波形和反相波形。

由 TCRp 寄存器在向下计数时产生的 INTTMp 作为启动触发器的从通道 3、5、7 产生的 INTTMq, 作为 TOp (TOLp=0) 的设置条件。在 TCRp 向上计数时从通道 2、4、6 产生的 INTTMp, 作为 TOp (TOLp=0) 的复位条件。

在 TCRp 向下计数时产生的 INTTMp 作为 Toq (TOLq=1) 的设置条件。由 TCRp 寄存器在向下计数时产生的 INTTMp 作为启动触发器的从通道 3、5、7 产生的 INTTMq, 作为 TOq (TOLq=0) 的复位条件。

<R> 如果 TOp 和 TOq 的设置条件和复位条件冲突, 设置条件优先。

通过设置 TOLp 和 TOLq, 带死区的 PWM 波形可在从通道 2、4、6 和 3、5、7 的正相和负相之间切换。

<R> **注意事项 1.** 主通道的 TDR00 必须在从通道 2、4、6 的递增状态周期下重写 (计数状态由从通道的 CSF (TSRp 寄存器) 或主通道的 TO00 输出电平进行判断。)。当向下状态期间 TDR00 的值重写, 向下和向上状态期间不同且不能输出一个预期波形, 因为主通道重写的 TDRn 的值在下一周期会变为有效。

<R> **2.** 当 TE00 = 1、TEp = 1、或 TEq = 1 时, TS00、TSp 或 TSq 不能设为 “1” (强制重启)。当 TE00 = 1、TEp = 1、或 TEq = 1 时, 若 TS00、TSp、或 TSq 设为 “1”, 则计数器的值 (TCR00、TCRp、或 TCRq) 将非法且 TO00、TOp、或 TOq 将不能输出预期波形。

备注 p = 02、04、06
q = 03、05、07

从通道 2、4、6 的 TDRp 的值从下一载波周期开始变为有效（向上和向下触发检测）。

从通道 3、5、7 的 TDRq 的值从下一开始时序变为有效（死区时间控制触发检测）。

<R> 向上状态期间从通道 2、4、6 的 INTTMp 检测之后，推荐执行从通道 3、5、7 的 TDRq 重写。

图 7-38. 作为 6 相三角波 PWM 输出功能操作的框图

主通道

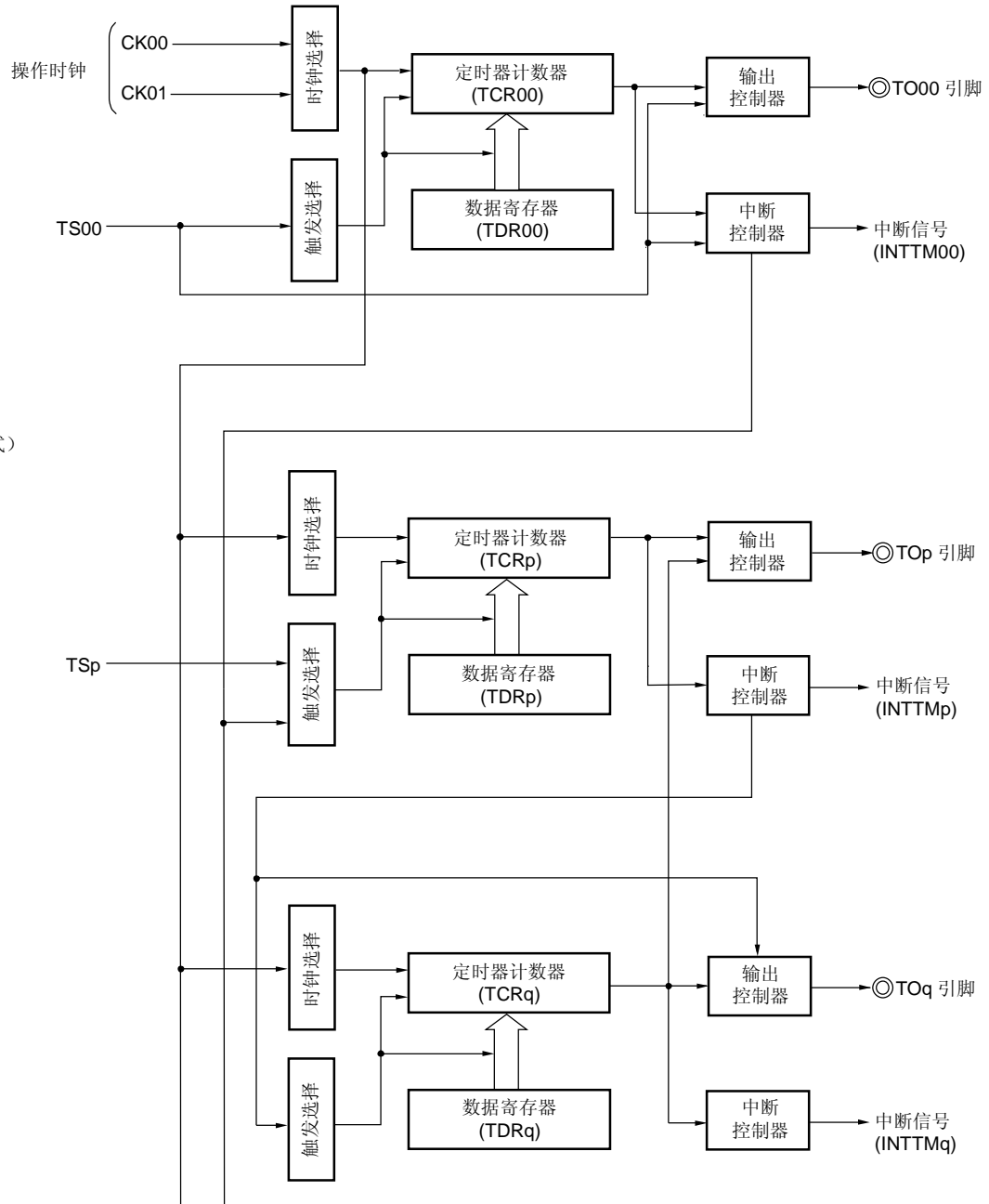
（间隔定时器模式）

从通道2、4、6

（递增与递减计数模式）

从通道3、5、7

（一次计数模式）



备注

p = 02、04、06

q = 03、05、07

图 7-39. 作为 6 相三角波 PWM 输出功能操作的基本时序示例

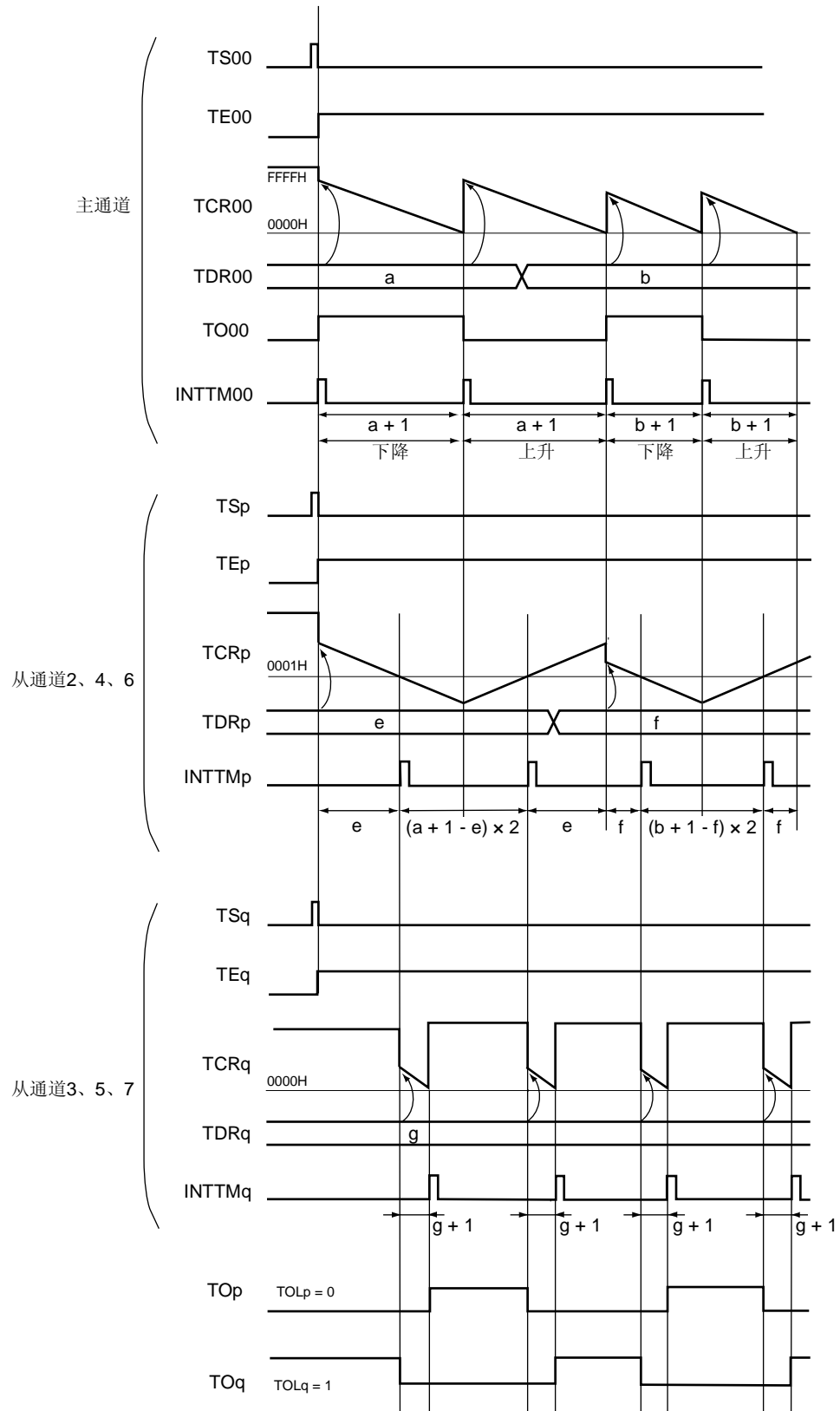
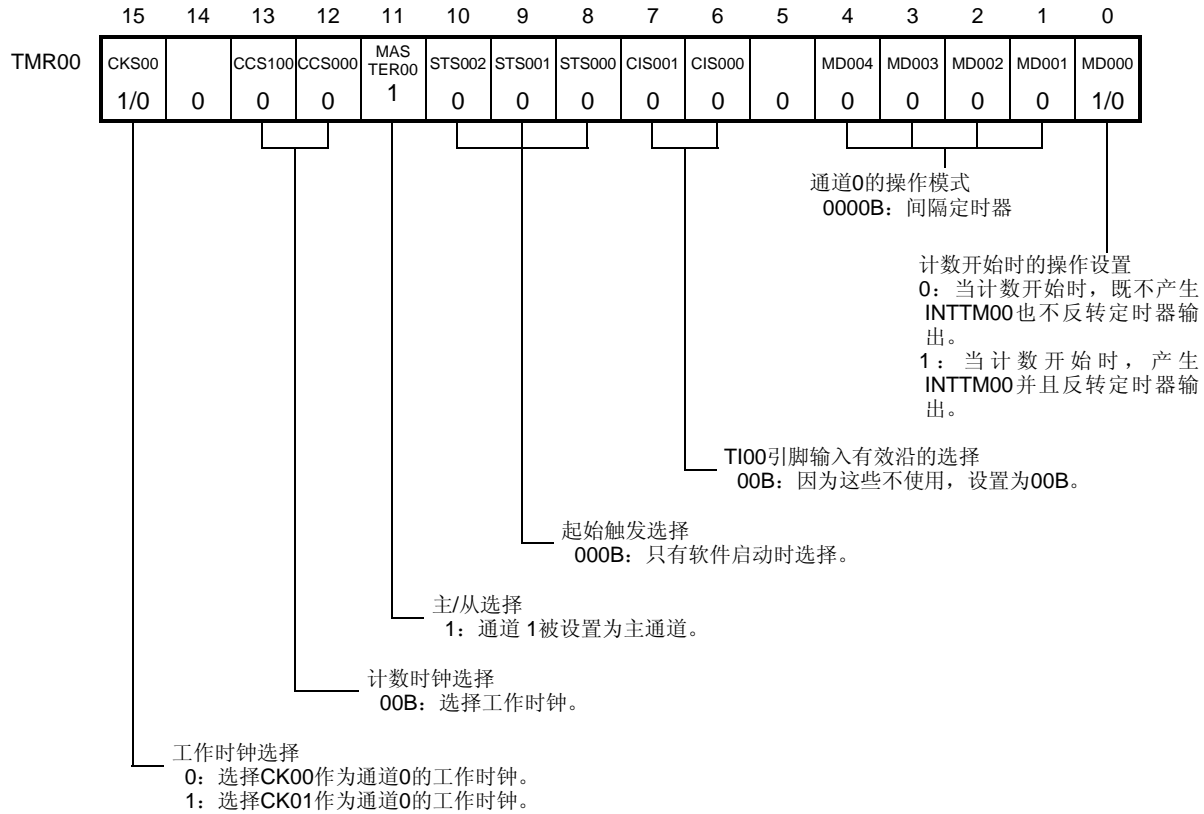


图 7-40. 6 相三角波 PWM 输出功能(主通道)的寄存器设置示例

(a) 定时器模式寄存器 00(TMR00)



(b) 其它寄存器

<R>

TOE0: TOE00 [※]	0: 通过计数操作停止 TO00 输出操作。 1: 通过计数操作允许TO00输出操作。
TO0:TO00	0: 从TO00输出低电平。 1: 从TO00输出高电平。
TOM0:TOM00	0: 设置反转模式。
TOT0:TOT00	0: 当TOM00 = 0(反转模式)时设为0。
TOL0:TOL00	0: 当TOM00 = 0(反转模式)时设为0。
TDE0:TDE00	0: 停止死区时间控制。
TRE0:TRE00	0: 停止实时输出。
TRO0:TRO00	0: 当TRE00=0(停止实时输出)时设为0。
TRC0:TRC00	0: 作为实时输出触发产生通道不运行。
TME0:TME00	0: 停止调制输出。

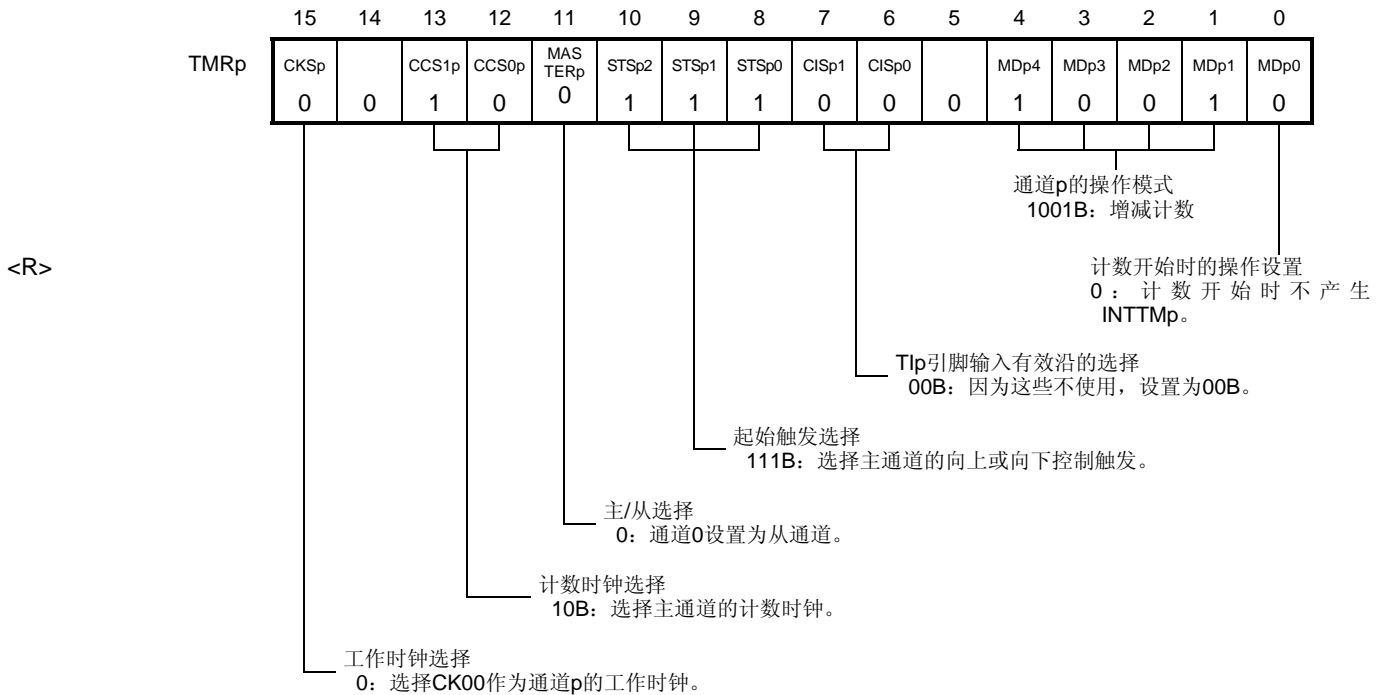
<R>

注 在下列情况中, 设置主通道的 TOE00 至“1”。

- 当使用一个 INTTMM0、INTTMV0、INTTMM1 或 INTTMV1 中断信号
- 利用 OPMR、OPHS 或 OPHT 寄存器, 控制一个高阻输出或通过控制器选择一个 A/D 转换触发时

图 7-41. 6 相三角波 PWM 输出功能(从通道 2、4、6)的寄存器设置示例

(a) 定时器模式寄存器 p(TMRp)



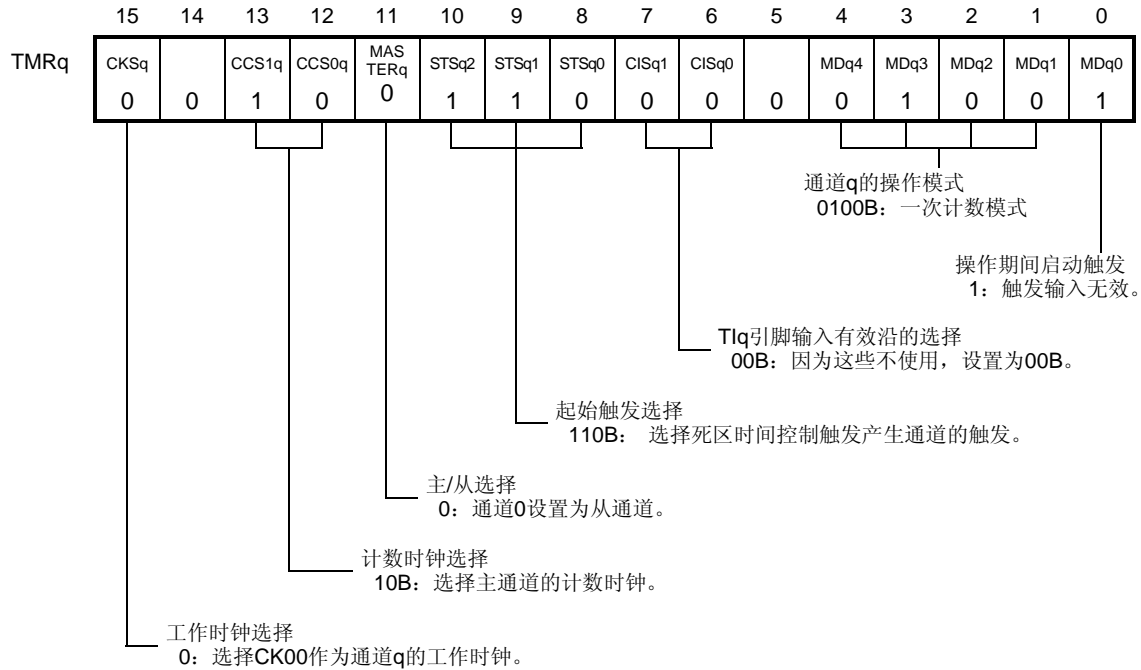
(b) 其它寄存器

TOE0:TOEp	0: 通过计数操作, 停止 TOp 输出操作。 1: 通过计数操作, 允许 TOp 输出操作。
TO0:TOp	0: 从 TOp 输出低电平。 1: 从 TOp 输出高电平。
TOM0:TOMp	1: 设置组合操作模式。
TOT0:TOTp	1: 设置三角波PWM输出。
TOL0:TOLp	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)
TDE0:TDEp	1: 允许死区时间控制。
TRE0:TREp	0: 停止实时输出。
TRO0:TROp	0: 当 TREp = 0时设为0(停止实时输出)。
TRC0:TRCp	0: 作为实时输出触发产生通道不运行。
TME0:TMEp	0: 停止调制输出。

备注 p = 02、04、06

图 7-42. 6 相三角波 PWM 输出功能(从通道 3、5、7)的寄存器设置示例

(a) 定时器模式寄存器 q(TMRq)



(b) 其它寄存器

TOE0:TOEq	0: 通过计数操作, 停止TOq输出操作。 1: 通过计数操作, 允许 TOq 输出操作。
TO0:TOq	0: 从TOq输出低电平。 1: 从TOq输出高电平。
TOM0:TOMq	1: 设置组合操作模式。
TOT0:TOTq	1: 设置三角波PWM输出。
TOL0:TOLq	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)
TDE0:TDEq	1: 允许死区时间控制。
TRE0:TREq	0: 停止实时输出。
TRO0:TROq	0: 当 TREq = 0时设为0(停止实时输出)。
TRC0:TRCq	0: 作为实时输出触发产生通道不运行。
TME0:TMEq	0: 停止调制输出。

备注 q = 03、05、07

图 7-43. 当用作 6 相三角波 PWM 输出功能时的操作过程 (1/2)

软件操作	硬件状态
TAUS 默认设置	掉电状态 (停止供给时钟, 并且所有寄存器不能写入。)
设置PER2寄存器的TAU0EN位和TAUOPEN位为1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
设置TPS0 寄存器。	
决定CK00和CK01的时钟频率。	
通道默认设置	通道停止操作。 (供给时钟并且产生一些功耗。)
设置要使用的通道的TMR00、TMRp和TMRq寄存器 (决定通道的操作模式)。	
间隔 (周期) 值设置到主通道的TDR00寄存器, 占空比设置到从通道2、4、6的TDRp寄存器, 以及设置死区时间宽度到从通道 3、5、7的TDRq寄存器。	TO00、TOp和TOq引脚为高阻输出状态。
设置从通道。	
TOM0寄存器的TOMp和TOMq位以及TOT0寄存器的TOTp和TOTq位被设为1 (三角波PWM输出)。	
设置TOLp和TOLq位并决定TOp和TOq的有效电平。	
设置TDE0寄存器的TDEp和TDEq位为1 (允许死区时间控制)。	
设置TO00、TOp和TOq位并决定TO00、TOp和TOq的默认电平。	当端口模式寄存器在输出模式下, 并且端口寄存器为0时, TO00、TOp和TOq输出默认设置电平。
设置TOE00、TOEp和TOEq位为1并使能TO00、TOp和TOq操作。	TO00、TOp和TOq不改变因为通道停止工作。
将端口寄存器和端口模式寄存器清为0。	TO00、TOp和TOq引脚输出TOp和TOq 的设置值。
操作开始	设置TOE00 (主通道)、TOEp和 TOEq(从通道)位为1 (仅当操作恢复时)。
TS0寄存器的TS00 (主通道)、TSp和TSq (从通道) 位同时设置为1。	TE00 = 1、TEp = 1、TEq = 1
因为是触发位, TS00、TSp和TSq位自动返回为0。	当主通道和从通道3、5、7开始计数且TMR00寄存器的MD000位为1时, 产生INTTM00。

备注 p = 02、04、06
q = 03、05、07

操作继续 (从下一页)

图 7-43. 当用作 6 相三角波 PWM 输出功能时的操作过程 (2/2)

软件操作		硬件状态
操作期间	<p>TDR00 (主通道) 寄存器的设定值必须在从通道p的向上状态周期下被改变。</p> <p>TDRp和TDRq (从通道) 寄存器的设置值不能修改。</p> <p>TCR00、TCRp和TCRq寄存器总是可读。</p> <p>TSRp(从通道)寄存器总是可读。</p>	<p>在主通道产生一个周期且从通道的计数操作被控制。在从通道2、4、6产生一个PWM占空，在从通道3、5、7产生死区时间。</p> <p>通过从通道2、4、6和从通道3、5、7的组合操作，从TOp和TOq引脚输出带有死区时间的三角波PWM波形。</p>
操作停止	<p>TT00 (主通道)、TTp 和 TTq (从通道) 位同时设置为1。</p> <p>因为是触发位，TT00、TTp和TTq位自动返回为0。</p> <p>TOE00, TOEp和TOEq位被清零且数值被设置到TO00, TOp和TOq位。</p>	<p>TE00、TEp和TEq=0，并且计数操作停止。</p> <p>TCR00、TCRp和TCRq保持计数值并停止。</p> <p>TO00、TOp和TOq输出不被初始化而保持当前状态。</p> <p>TO00, TOp和TOq引脚输出TO00, TOp和TOq的设置值。</p>
TAUS 停止	<p>需要保持TO00、TOp和TOq引脚输出电平时</p> <p>在把需要保持的值设置到端口寄存器后把TO00、TOp和TOq位清为0。</p> <p>不需要保持TO00、TOp和TOq引脚输出电平时</p> <p>切换端口模式寄存器到输入模式。</p> <p>PER2寄存器的TAU0EN位和TAUOPEN位被清为0。</p>	<p>TO00、TOp和TOq引脚输出电平通过端口功能保持。</p> <p>TO00、TOp和TOq引脚输出电平变为高阻输出状态。</p> <p>掉电状态</p> <p>所有电路被初始化，并且每个通道的SFR也被初始化。</p> <p>(TO00、TOp和TOq位清为0，TO00、TOp和TOq引脚设置为端口模式。)</p>

备注

p = 02、04、06

q = 03、05、07

操作继续 (到前一页)

7.4.7 中断信号缩短功能

中断信号缩短功能使用两个通道组合，把主通道的 INTTMn 进行指定次数的缩短后从从通道输出。

中断信号缩短功能把从通道作为(7.4.6) 6 相三角波 PWM 输出功能的子功能使用。因此主通道的设置与(7.4.6) 6 相三角波 PWM 输出功能的设置一致。缩短的中断次数可以通过下面的表达式来计算。

$$\begin{aligned} & \text{用于缩短的中断次数} = \text{TDRm 的设置值(从通道)} \\ & \rightarrow \text{每} \{ \text{TDRm(从通道)} + 1 \} \text{次, 主通道的 INTTMn 从通道的 INTTMm 输出} \end{aligned}$$

主通道的 TCRn 在间隔定时器模式下向下计数。

通过设置通道启动触发位(TSn)为 1 使得 TCRn 加载 TDRn 的值。此时，当 TMRn 的 MDn0=0 时 INTTMn 不输出且 TOn 不反转。当 TMRn 的 MDn0=1 时 INTTMn 输出且 TOn 反转。此后，TCRn 与计数时钟同步向下计数。当 TCRn 变为 0000H 时，INTTMn 输出且 TOn 在下一个计数时钟反转。此时 TCRn 再次加载 TDRn 的值。此后继续相同的操作。

从通道在事件计数器模式下作为向下计数器工作，并且控制主通道的 INTTMn 信号缩短。

通过设置通道启动触发位(TSm)为 1 使得 TCRm 加载 TDRm 的值。

TCRm 与主通道的 INTTMn 输出同步向下计数，并且再次加载 TDRm 的值以及当 TCRm 变为 0000H 时输出 INTTMm。此后继续相同的操作。

TOn 不能使用，因为 TOn 由于使用外部事件而变为不规则波形。

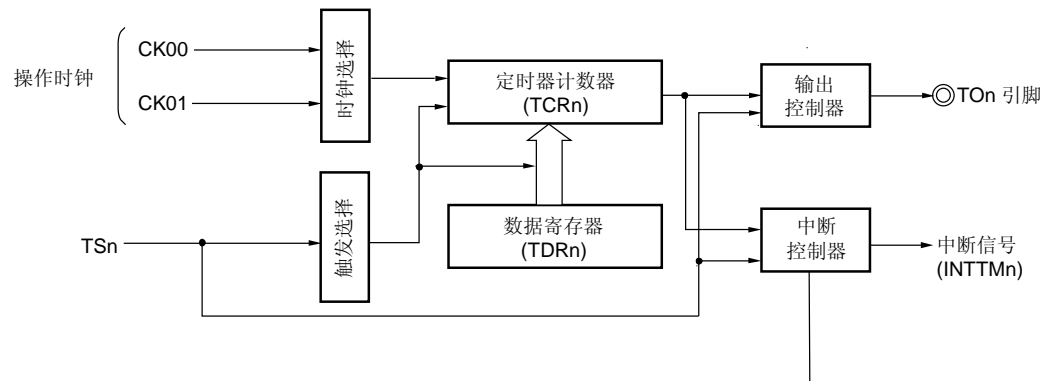
主通道的 TDRn 在下一开始时序后变为有效(主通道 INTTMn 产生)。

从通道的 TDRm 在下一开始时序后变为有效(从通道 INTTMm 产生)。

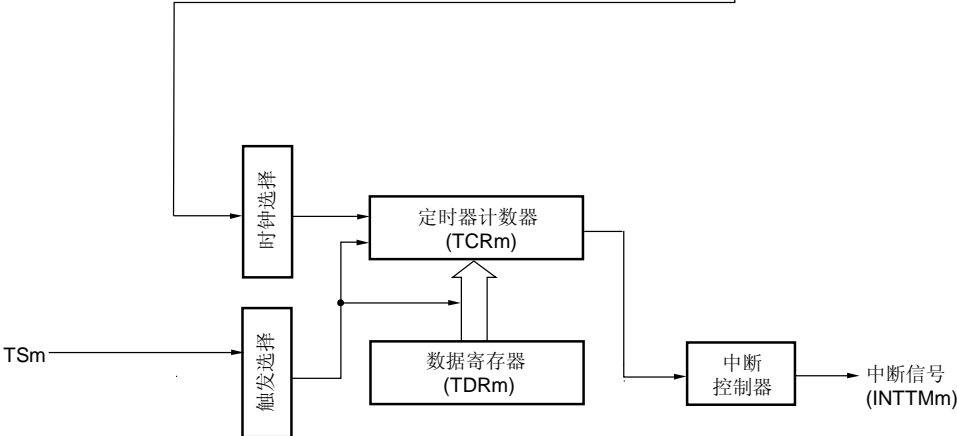
备注 n = 00
 m = 01

图 7-44. 作为中断信号缩短功能操作的框图

主通道
(间隔定时器模式)

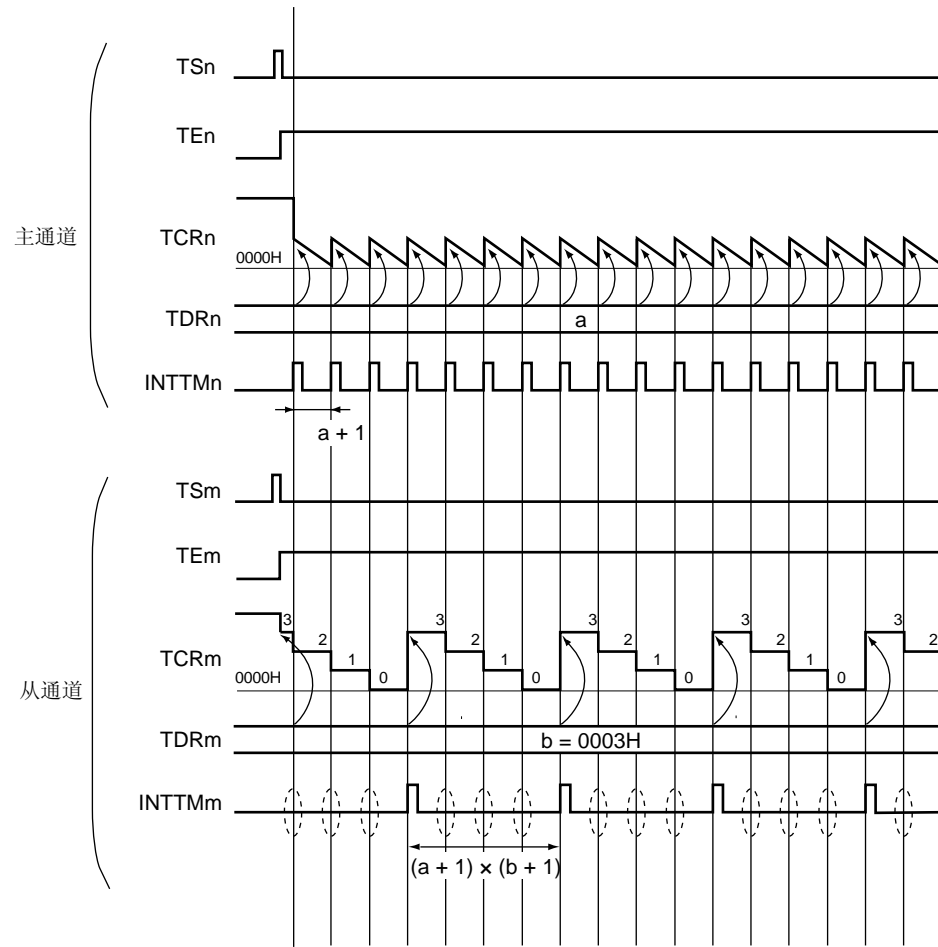


从通道
(事件计数模式)



备注 n = 00
 m = 01

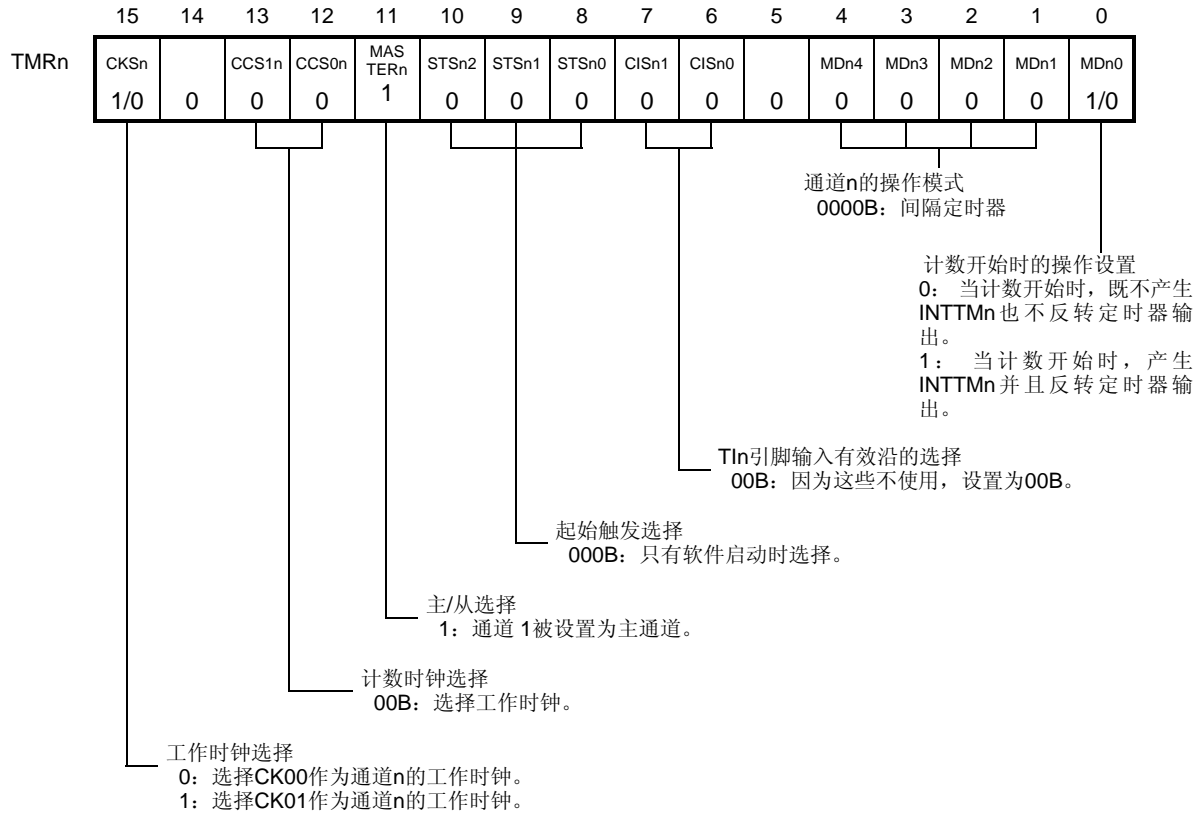
图 7-45. 作为中断信号缩短功能操作的基本时序示例



备注 $n = 00$
 $m = 01$

图 7-46. 当作为中断信号缩短功能(主通道)使用时寄存器设置的示例

(a) 定时器模式寄存器 n(TMRn)



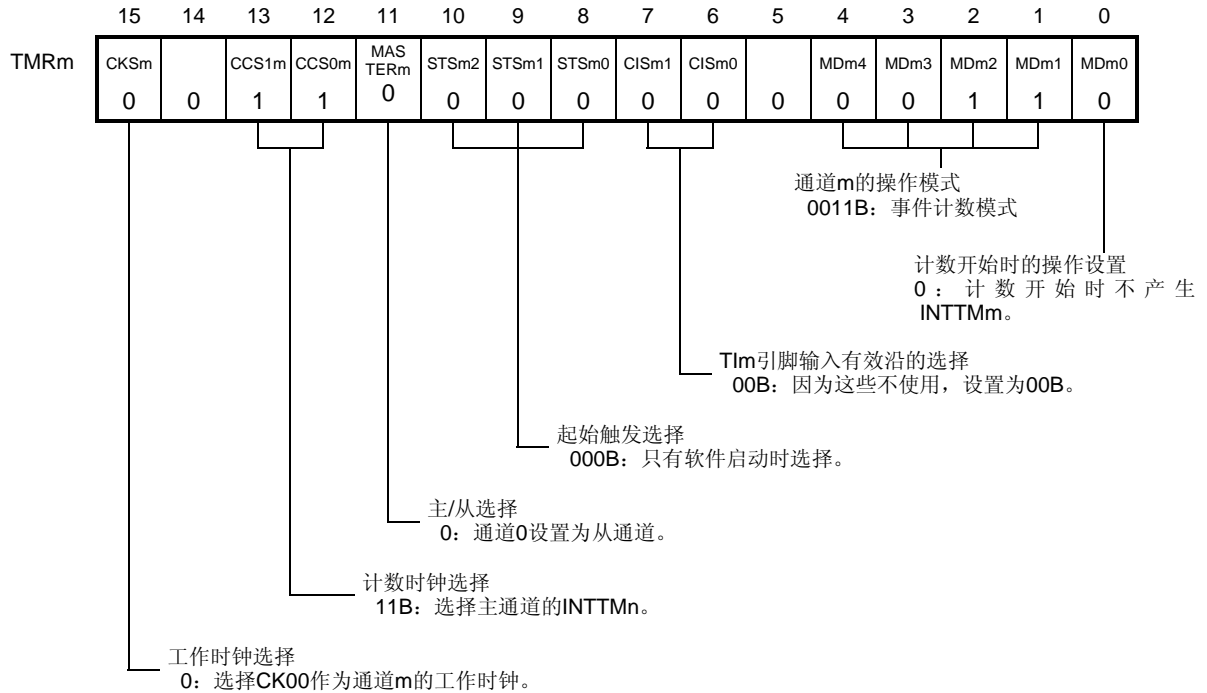
(b) 其它寄存器

TOE0:TOEn	0: 通过计数操作, 停止TON输出。
TO0:TON	0: 从TON输出低电平。
TOM0:TOMn	0: 设置反转模式。
TOT0:TOTn	0: 当TOMn=0(反转模式)时设为0。
TOL0:TOLn	0: 当TOMn=0(反转模式)时设为0。
TDE0:TDEn	0: 停止死区时间控制。
TRE0:TREn	0: 停止实时输出。
TRO0:TROn	0: 当 TREn = 0时设为0(停止实时输出)。
TRC0:TRCn	0: 作为实时输出触发产生通道不运行。
TME0:TMEn	0: 停止调制输出。

备注 n = 00

图 7-47. 当作为中断信号缩短功能(从通道)使用时寄存器设置的示例

(a) 定时器模式寄存器 m(TMRm)



(b) 其它寄存器

TOE0:TOEm	0: 通过计数操作, 停止TOM输出操作。
TO0:TOM	0: 从TOM输出低电平。
TOM0:TOMm	0: 当TOEm=0时设为0(通过计数操作停止TOM操作)。
TOT0:TOTm	0: 当TOMm = 0(反转模式)时设为0。
TOL0:TOLm	0: 当TOMm = 0(反转模式)时设为0。
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	0: 停止实时输出。
TRO0:TROm	0: 当 TREm = 0时设为0(停止实时输出)。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。

备注 n = 00
 m = 01

图 7-48. 当使用中断信号缩短功能时的操作过程



7.4.8 作为A/D转换触发输出功能的操作(类型 1)

A/D 转换触发输出功能使用两个通道组合来输出 A/D 转换触发。

由从通道输出 A/D 转换触发信号

可使用多重从通道来增加 A/D 转换触发的输出。

A/D 转换触发输出功能把主通道作为(7.4.3) 6 相 PWM 输出功能的子功能。因此主通道的设置与(7.4.3) 6 相 PWM 输出功能的设置一致。可通过下面的表达式计算出 A/D 转换触发脉冲产生周期。

$$\text{A/D 转换触发脉冲产生周期(从载波周期开始到检测到 INTTMn 的间隔)} = \{\text{TDRm 的设置值(从通道)} + 1\} \times \text{计数时钟周期}$$

主通道的 TCRn 运行于间隔定时器模式，计数周期。

从通道的 TCRm 运行在一次计数模式下并计数占空比。通过使用主通道的 INTTMn 作为启动触发，主通道的 TCRm 加载 TDRm 值，并向下计数。当 TCRm 变为 0000H 时，TCRm 输出 INTTMm 并停止计数直到输入下一个启动触发(主通道的 INTTMn)。

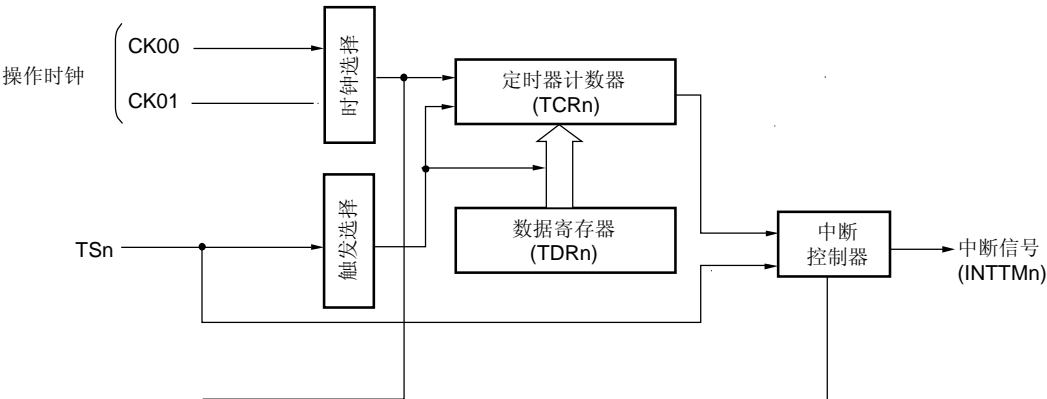
主通道的 TDRn 和从通道的 TDRm 在下一周期有效(主通道的 INTTMn 产生)。

<R> 注意事项 当 $TE_n = 1$ 或 $TE_m = 1$ 时， TS_n 或 TS_m 不能设为 1 (强制重启)。如果当 $TE_n = 1$ 或 $TE_m = 1$ 时， TS_n 或 TS_m 设为 1，0 计数器的值 (TCRn 或 TCRm) 将为非法。

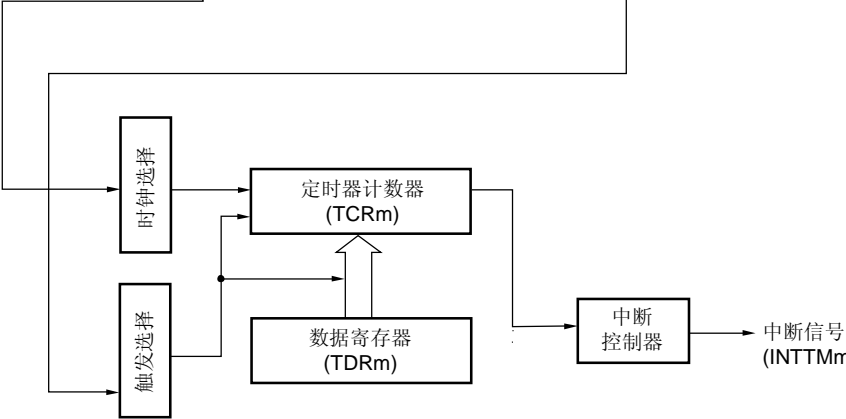
备注 当 $n = 00$: $m = 01$ 或 $m = 08、09$
当 $n = 04$: $m = 05$

图 7-49. 作为 A/D 转换触发输出功能操作的框图（类型 1）

主通道
(间隔定时器模式)

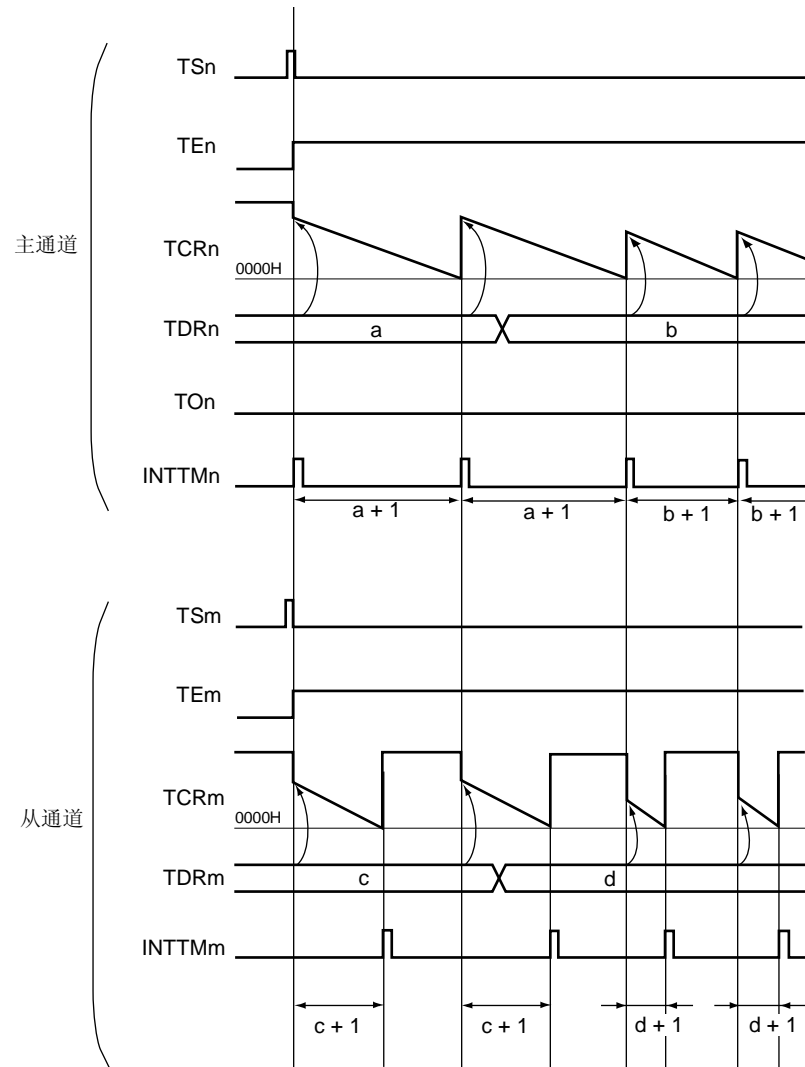


从通道
(1次计数模式)



备注 当 n = 00: m = 01 或 m = 08、09
当 n = 04: m = 05

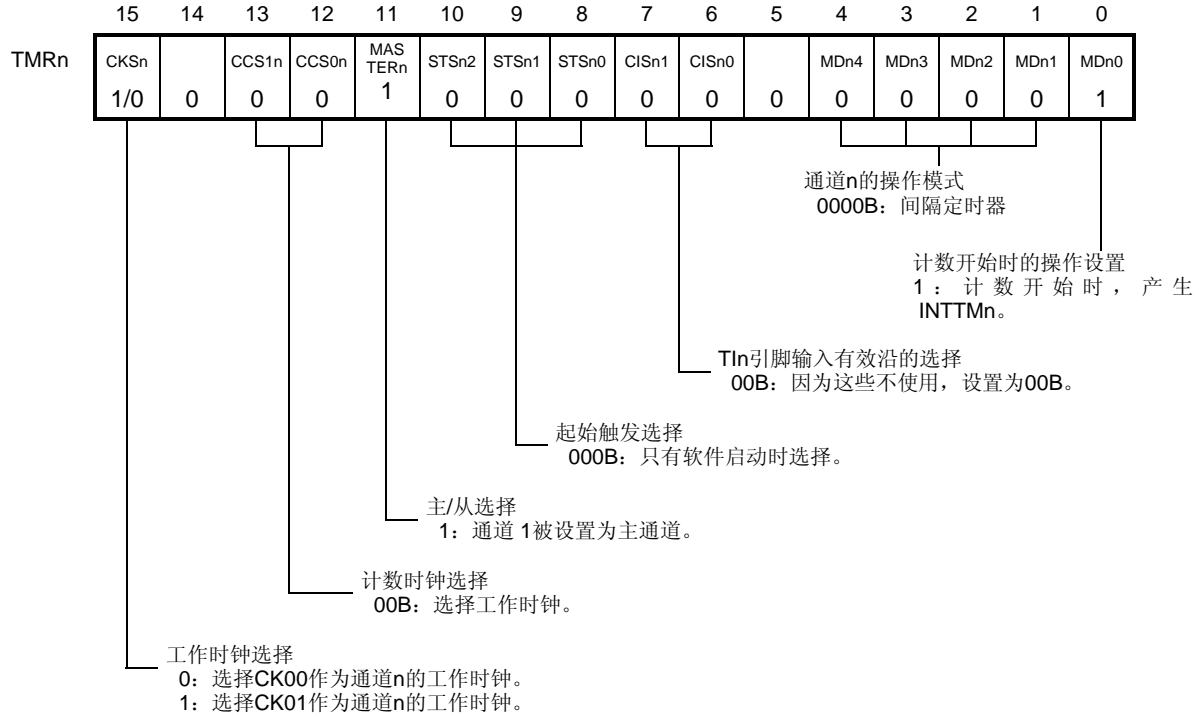
图 7-50. 作为 A/D 转换触发输出功能操作的基本时序示例（类型 1）



备注 当 n = 00: m = 01 或 m = 08、09
当 n = 04: m = 05

图 7-51. 当使用 A/D 转换触发输出功能(类型 1)(主通道)时寄存器设置的示例

(a) 定时器模式寄存器 n(TMRn)



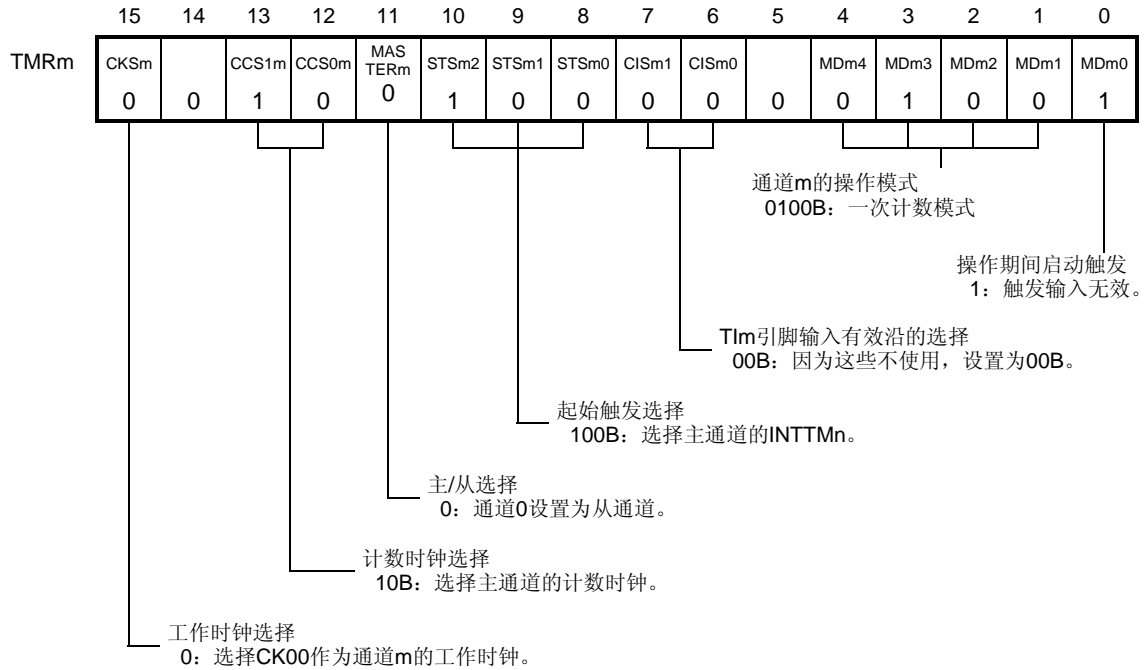
(b) 其它寄存器

TOE0:TOEn	0: 通过计数操作, 停止TOn输出。
TO0:TOn	0: 从TOn输出低电平。
TOM0:TOMn	0: 当TOEn=0时设为0(通过计数操作停止TOn输出)。
TOT0:TOTn	0: 当TOMn=0(反转模式)时设为0。
TOL0:TOLn	0: 当TOMn=0(反转模式)时设为0。
TDE0:TDEn	0: 停止死区时间控制。
TRE0:TREn	0: 停止实时输出。
TRO0:TROn	0: 当 TREn = 0时设为0(停止实时输出)。
TRC0:TRCn	0: 作为实时输出触发产生通道不运行。
TME0:TMEn	0: 停止调制输出。

备注 当 n = 00: m = 01 或 m = 08、09
当 n = 04: m = 05

图 7-52. 当使用 A/D 转换触发输出功能(类型 1)(从通道)时寄存器设置的示例

(a) 定时器模式寄存器 m(TMRm)

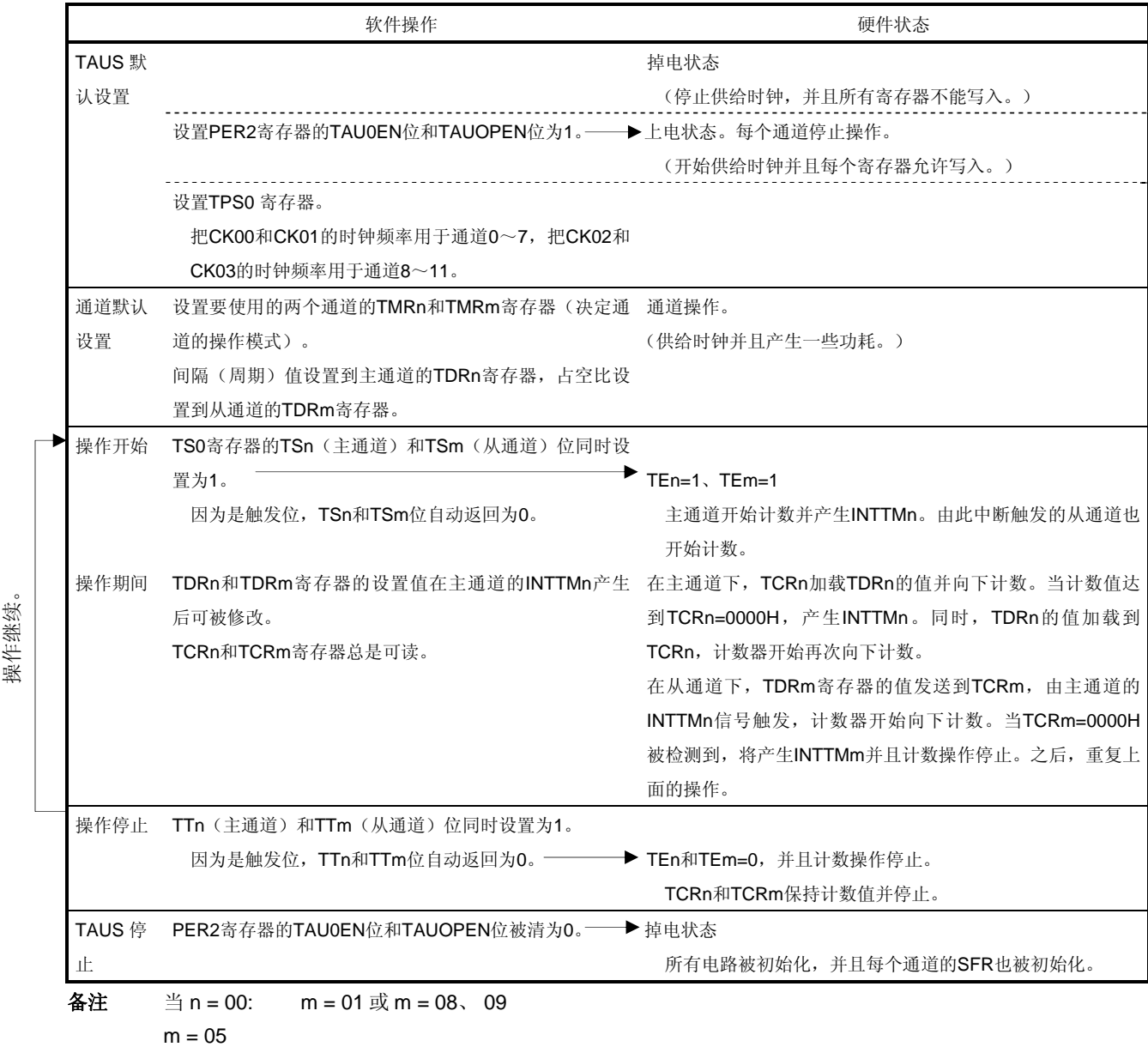


(b) 其它寄存器

TOE0:TOEm	0: 通过计数操作, 停止TOn输出。
TO0:TOM	0: 从TOM输出低电平。
TOM0:TOMm	0: 当TOEm=0时设为0(通过计数操作停止TOM操作)。
TOT0:TOTm	0: 当TOMm = 0(反转模式)时设为0。
TOL0:TOLm	0: 当TOMm = 0(反转模式)时设为0。
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	0: 停止实时输出。
TRO0:TROm	0: 当 TREm = 0时设为0(停止实时输出)。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。

备注 当 n = 00: m = 01 或 m = 08、09
当 n = 04: m = 05

图 7-53. 使用 A/D 转换触发输出功能（类型 1）时的操作过程



7.4.9 作为 A/D 转换触发输出功能的操作(类型 2)

A/D 转换触发输出功能使用两个通道组合来输出 A/D 转换触发。

由从通道输出 A/D 转换触发信号

可使用多重从通道来增加 A/D 转换触发的输出。

A/D 转换触发输出功能把从通道作为(7.4.6) 6 相三角波 PWM 输出功能的子功能。因此主通道的设置与(7.4.6) 6 相三角波 PWM 输出功能的设置一致。可通过下面的表达式计算出 A/D 转换触发脉冲产生周期。

A/D 转换触发脉冲产生周期(从载波周期开始到在向下状态中检测到 INTTMn 的间隔)={TDRm 的设置值(从通道)+1}×计数时钟周期

设置 TDRm 范围(从通道):0000H<TDRm(从通道)<{TDRn 的设置值(主通道)+1}

*从向下状态下检测到 INTTMm 直到向上状态下检测到 INTTMm 的间隔

={{TDRn 的设置值(主通道)+1}-{TDRm 的设置值(从通道)}}×2×计数时钟周期

主通道的 TCRn 运行于间隔定时器模式，计数周期。

通过设置通道启动触发位(TSn)为 1，TCRn 加载 TDRn 的值。

此时，当 TMRn 的 MDn0 为 0 时 INTTMn 不输出且 TOn 不反转。当 TMRn 的 MDn0 为 1 时 INTTMn 输出且 TOn 反转。此后，TCRn 与计数时钟一同向下计数。当 TCRn 变为 0000H 时，INTTMn 输出且 TOn 在下一个计数时钟反转。此时 TCRn 再次加载 TDRn 的值。此后继续相同的操作。

主通道计数的两个周期内产生一个载波周期。

从通道的计数操作通过定义主通道的第一个周期作为从通道的向下状态和第二个周期作为从通道的向上状态进行控制。

主通道的 TOn 输出向上和向下状态。

为了输出向上和向下状态，当 TOE0 寄存器的 TOEn 为 0 时，必须操作 TO0 寄存器的 TOn 且设置默认电平。

当 TMR0 寄存器的 MDn0=0 时，设置 TO0 寄存器的 TOn 为 1，以及当 MDn0=1 时，设置 TOn 为 0。

通过设置默认电平，在向下状态下一个高电平从 TOn 中输出且在向上状态下输出一个低电平。

从通道的 TCRm 运行在上下计数模式下并计数占空比。在通道启动触发位(TSm)被设为 1 后，TRm 在第一个计数时钟下加载 TDRm 的值。此后，向上计数和向下计数由主通道的操作进行切换。当 TCRm 变为 0001H 时，将输出 INTTMm。

当在主通道的向上状态下产生 INTTMn 时，TCRm 再次加载 TDRm 的值。此后继续相同的操作。

备注 当 n = 00: m = 01 或 m = 08、09
当 n = 04: m = 05

<R> 注意事项 1. 主通道的 TDRn 必须在从通道的递增状态周期下重写（计数状态由从通道的 CSF（TSRm 寄存器）或主通道的 TOn 输出电平进行判断）。当向下状态期间 TDRn 的值重写，向下和向上状态期间不同且不能输出一个预期波形，因为主通道被重写的 TDRn 的值在下一周期会变为有效。

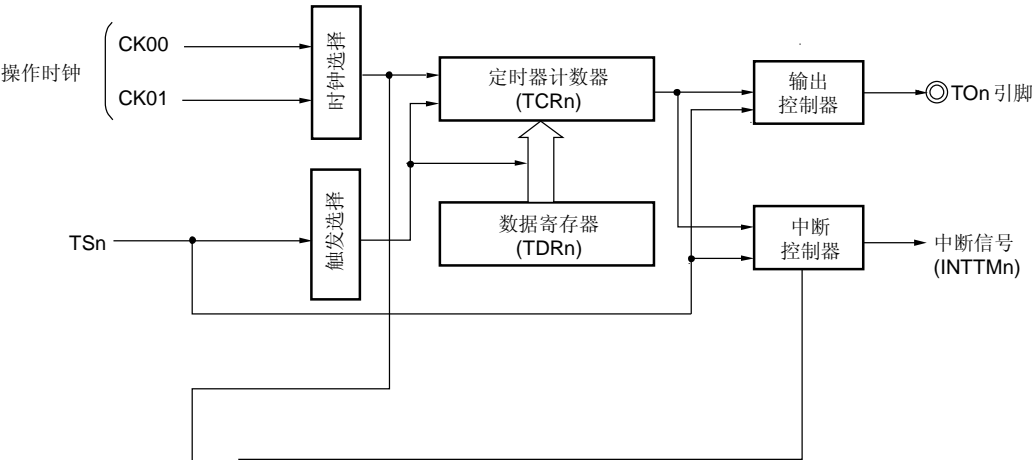
<R> 2. 当 TEn = 1 或 TEm = 1 时，TSn 或 TSm 不能设为“1”（强制重启）。如果当 TEn = 1 或 TEm = 1 时，TSn 或 TSm 设为“1”，0 计数器的值（TCRn 或 TCRm）将为非法。

从通道的 TDRm 从下一载波周期开始有效（向上和向下触发检测）。

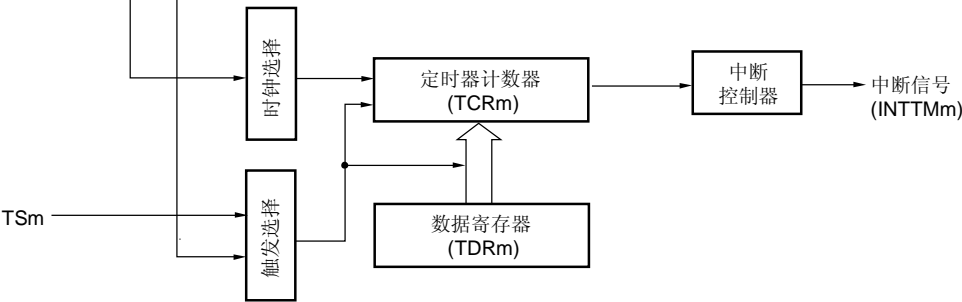
备注 当 n = 00: m = 01 或 m = 08、09
当 n = 04: m = 05

图 7-54. 作为 A/D 转换触发输出功能操作的框图（类型 2）

主通道
(间隔定时器模式)

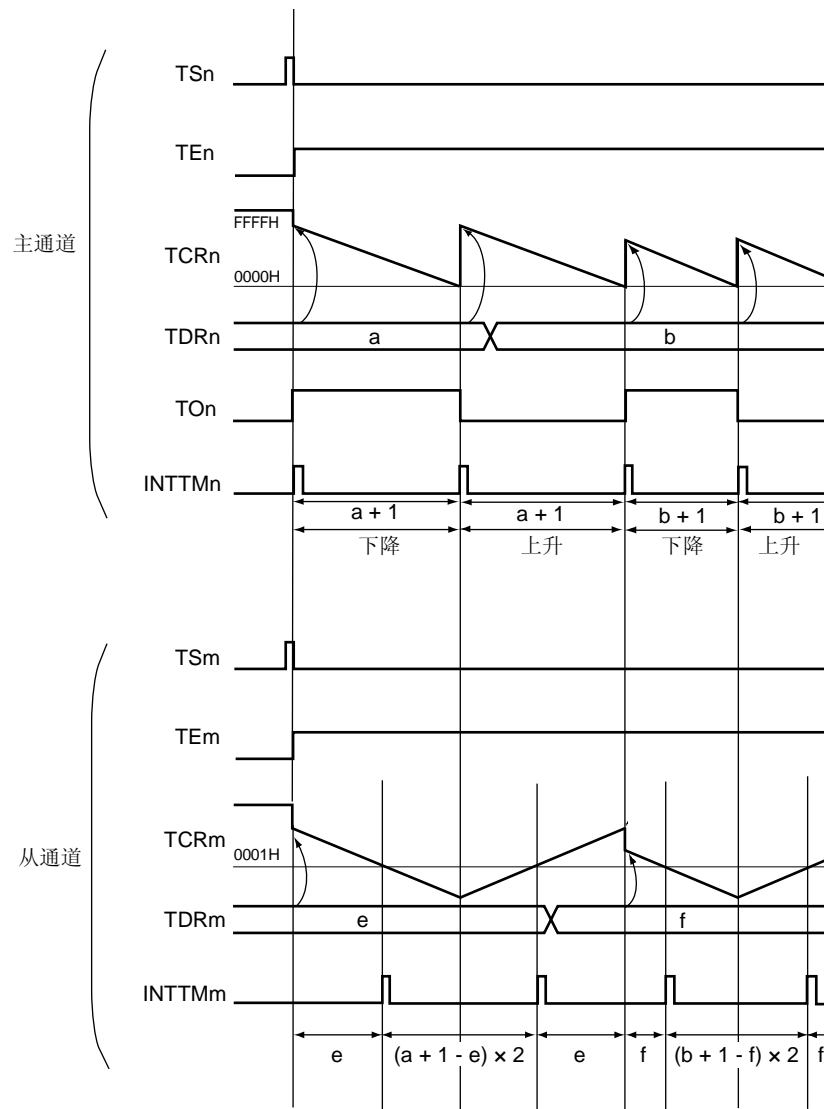


从通道
(上升与下降计数模)



备注 当 n = 00: m = 01 或 m = 08、09
当 n = 04: m = 05

图 7-55. 作为 A/D 转换触发输出功能操作的基本时序示例（类型 2）



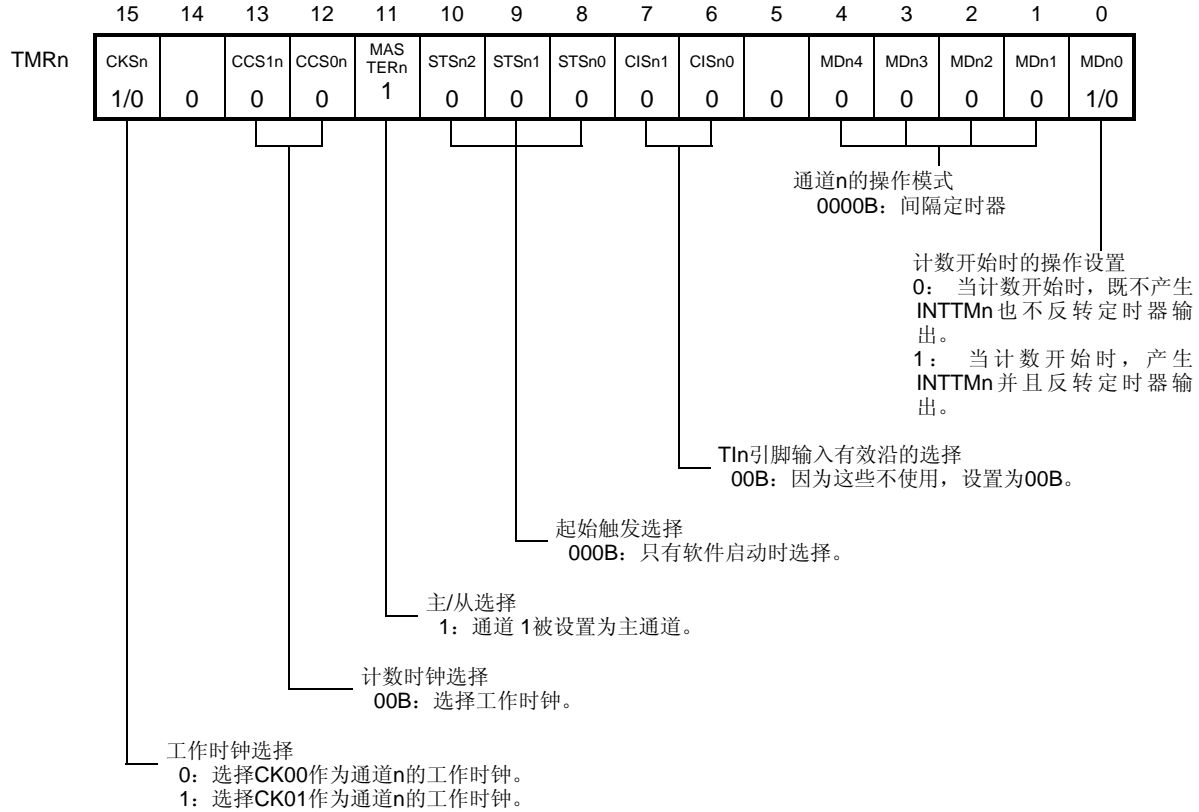
备注

当 n = 00: m = 01 或 m = 08、09

当 n = 04: m = 05

图 7-56. 当使用 A/D 转换触发输出功能(类型 2)(主通道)时寄存器设置的示例

(a) 定时器模式寄存器 n(TMRn)



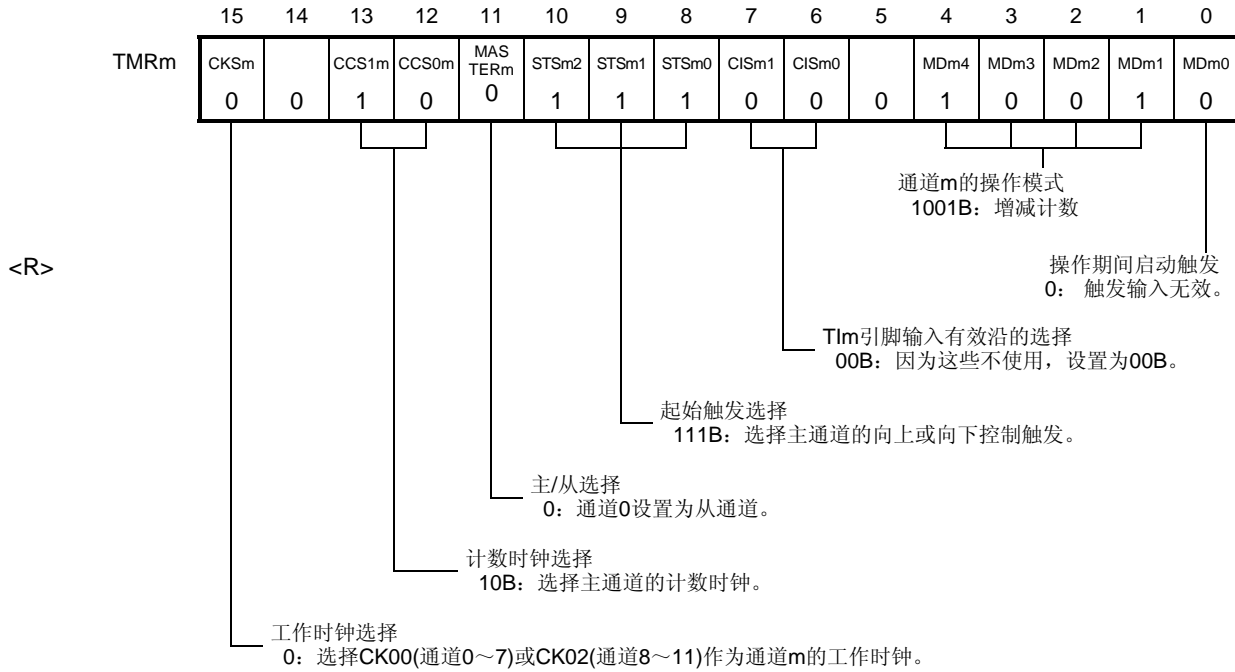
(b) 其它寄存器

TOE0:TOEn	0: 通过计数操作, 停止TOn输出。 1: 通过计数操作, 允许TOn输出。
TO0:TOn	0: 从TOn输出低电平。 1: 从TOn输出高电平。
TOM0:TOMn	0: 设置反转模式。
TOT0:TOTn	0: 当TOMn=0(反转模式)时设为0。
TOL0:TOLn	0: 当TOMn=0(反转模式)时设为0。
TDE0:TDEn	0: 停止死区时间控制。
TRE0:TREn	0: 停止实时输出。
TRO0:TROn	0: 当 TREn = 0时设为0(停止实时输出)。
TRC0:TRCn	0: 作为实时输出触发产生通道不运行。
TME0:TMEn	0: 停止调制输出。

备注 当 n = 00: m = 01 或 m = 08、09
当 n = 04: m = 05

图 7-57. 当使用 A/D 转换触发输出功能(类型 2)(从通道)时寄存器设置的示例

(a) 定时器模式寄存器 m(TMRm)



(b) 其它寄存器

TOE0:TOEm	0: 通过计数操作, 停止TON输出。
TO0:TOM	0: 从TOM输出低电平。
TOM0:TOMm	0: 当TOEm=0时设为0(通过计数操作停止TON操作)。
TOT0:TOTm	0: 当TOMm = 0(反转模式)时设为0。
TOL0:TOLm	0: 当TOMm = 0(反转模式)时设为0。
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	0: 停止实时输出。
TRO0:TROm	0: 当 TREm = 0时设为0(停止实时输出)。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。

备注 当 n = 00: m = 01 或 m = 08、 09
 当 n = 04: m = 05

图 7-58. 使用 A/D 转换触发输出功能（类型 2）时的操作过程(1/2)

	软件操作	硬件状态
TAUS		掉电状态
默认		（停止供给时钟，并且所有寄存器不能写入。）
设置	设置PER2寄存器的TAU0EN位和TAUOPEN位为1。——▶	上电状态。每个通道停止操作。
		（开始供给时钟并且每个寄存器允许写入。）
	设置TPS0 寄存器。	
	把CK00和CK01的时钟频率用于通道0~7，把CK02和CK03的时钟频率用于通道8~11。	
通道	设置要使用的通道的TMRn和TMRm寄存器（决定通道的操作模式）。	通道停止操作。
默认		（供给时钟并且产生一些功耗。）
设置	间隔（周期）值设置到主通道的TDRn寄存器，中断宽度设置到从通道的TDRm寄存器。	TOn引脚进入高阻输出状态。
	设置主通道。	
	设置TOM0寄存器的TOMn位为0（反转模式）。	
	设置TDE0寄存器的TDEn位为0（使能死区时间控制）。	
	设置TOn位并决定TOn位的默认电平。——▶	当端口模式寄存器在输出模式下，并且端口寄存器为0时，TOn输出默认设置电平。
	设置TOEn位为1并允许TOn操作。——▶	因为通道停止工作，TOn不改变。
	将端口寄存器和端口模式寄存器清为0。——▶	TOn引脚输出TOn设置电平。
操作	设置TOEn（主通道）位为1（仅当操作恢复时）。	
开始	TS0寄存器的TSn（主通道）和TSM（从通道）位同时设置为1。——▶	TEn=1、TEm=1
	因为是触发位，TSn和TSM位自动返回为0。	当主通道和从通道开始计数且TMRn寄存器的MD0n位为1时，产生INTTMn。
操作	TDRn（主通道）寄存器的设置值必须在向上状态周期下被改变。	在主通道下，TCRn加载TDRn的值并向下计数。当计数值达到TCRn=0000H，产生INTTMn。同时，TDRn的值加载到TCRn，计数器开始再次向下计数。
期间	TDRm（从通道）寄存器的设置值能改变。	在从通道下，TCRm加载TDRm的值并且根据主通道的操作进行向下计数和向上计数的切换。INTTMm的产生和计数操作的停止根据TCRm=0001H的检测而定。在主通道的向上状态下通过INTTMn的产生使得TCRm再次加载TDRm的值和继续计数操作。
	TCRn和TCRm寄存器总是可读。	
	TSRm（从通道）寄存器总是可读。	

备注 当 n = 00: m = 01 或 m = 08、09
 当 n = 04: m = 05

操作继续（从下一页）

图 7-58. 使用 A/D 转换触发输出功能（类型 2）时的操作过程（2/2）

↑
操作继续（到前一页）

	软件操作	硬件状态
操作停止	<p>TTn（主通道）和TTm（从通道）位同时设置为1。</p> <p>因为触发位，TTn和TTm位自动返回为0。————→</p> <p>设置TOEn（主通道）位为0，TOn位被赋值。————→</p>	<p>TEn和TEm=0，并且计数操作停止。</p> <p>TCRn和TCRm保持计数值并停止。</p> <p>TOn输出不被初始化但保持当前状态。</p> <p>TOn引脚输出TOn设置电平。</p>
TAUS 停止	<p>需要保持TOn引脚输出电平</p> <p>当把需要保持的值设置到端口寄存器后，把TOn位清为0。————→</p> <p>不需要保持TOn引脚输出电平时</p> <p>切换端口模式寄存器到输入模式。————→</p> <p>PER2寄存器的TAU0EN位和TAUOPEN位被清为0。————→</p>	<p>TOn引脚输出电平由端口功能保持。</p> <p>TOn引脚输出电平变为高阻输出状态。</p> <p>掉电状态</p> <p>所有电路被初始化，并且每个通道的SFR也被初始化。（TOn位清为0，并且TOn引脚设置为端口模式。）</p>

备注

当 n = 00: m = 01 或 m = 08、09

当 n = 04: m = 05

<R> 7.4.10 作为连接实时输出功能的操作（类型 1）

通过使用从通道 1 的 INTTMn 输出（实时输出触发产生通道），可以把从通道 2~7（实时输出通道）的 TROm 的值由 TOm 输出。

此功能中，从通道 1 被用作 **7.4.3 6 相 PWM 输出功能**中描绘的一个扩展功能使用。因此，主通道的设置与 **7.4.3 6 相 PWM 输出功能**中的设置一致。

如果从通道 1 的 TRCn 被设置为 1，通过与主通道的组合操作产生的 INTTMn 被用作实时输出触发。通过实时输出触发，从通道 2~7 的实时输出通道从 TOm 输出 TROm 的设定值。

从通道 1 增加一个延时到主通道的 INTTM00 且产生一个实时输出触发。

主通道的 INTTM00 的延迟值可通过下面的公式计算。

$$\text{主通道的 INTTM00 的延迟} = (\text{TDRn (从通道 1) 的设置值} + 1) \times \text{计数时钟周期}$$

主通道操作在间隔定时器模式并计数周期。

通道启动触发位 (TS00) 设置为 1 后，在第一个计数时钟 TCR00 加载 TDR00 的值。这时，通过设置 TMR00 的 MD000 为“1”输出 INTTM00。

之后，TCR00 随着计数时钟向下计数。

当 TCR00 变为 0000H 时，在下一个计数时钟 INTTM00 输出。此时 TCR00 再次加载 TDR00 的值。此后继续相同的操作。

从通道 1 的 TCRn 运行于一次计数模式，并产生一个实时输出触发。TDRm 的值加载到 TCRm，主通道的 INTTM00 作为启动触发，开始向下计数。当 TCRn = 0000H 时，TCRn 输出 INTTMn，并且停止计数直到输入下一个启动触发（主通道的 INTTM00）。把 TROn 和 TROm 的设定值，可以在从通道 1 的 INTTMn 输出时序，由 TOn 和 TOm 输出。

从通道 1 (TRCn=1) 的低通道（从通道 2~7）的 TOm 是由 TREm 和 TRCm 位进行控制。

当低通道 (TRCm=0) 的 TREm 为“1”时，此通道作为实时输出通道运行而且在从通道 1 的 INTTMn 输出计时，TOm 输出 TROm 的设定值。当低通道的 TREm 或 TRCm 为 0 或 1，在实时输出触发生成通道的 INTTMn 输出时序中，TOm 不反转。

当此功能被使用时，低通道的 TCRm、TDRm 和 INTTMm 可用于不同功能。

主通道的 TDR00 和 TDRn 和从通道 1 从下一周期有效（主通道的 INTTM00 产生）。

注意事项 当 TE00 = 1 或 TEn = 1 时，TS00 或 TSn 不能被设为“1”（强制重启）。当 TE00 = 1 或 TEn = 1 时，如果 TS00 或 TSn 被设为“1”，0 计数器的值（TCR00 或 TCRn）将为非法且 TOm 不能输出预期的波形。

备注 n = 01
m = 02~07

图 7-59 作为实时输出功能操作的框图（类型 1）

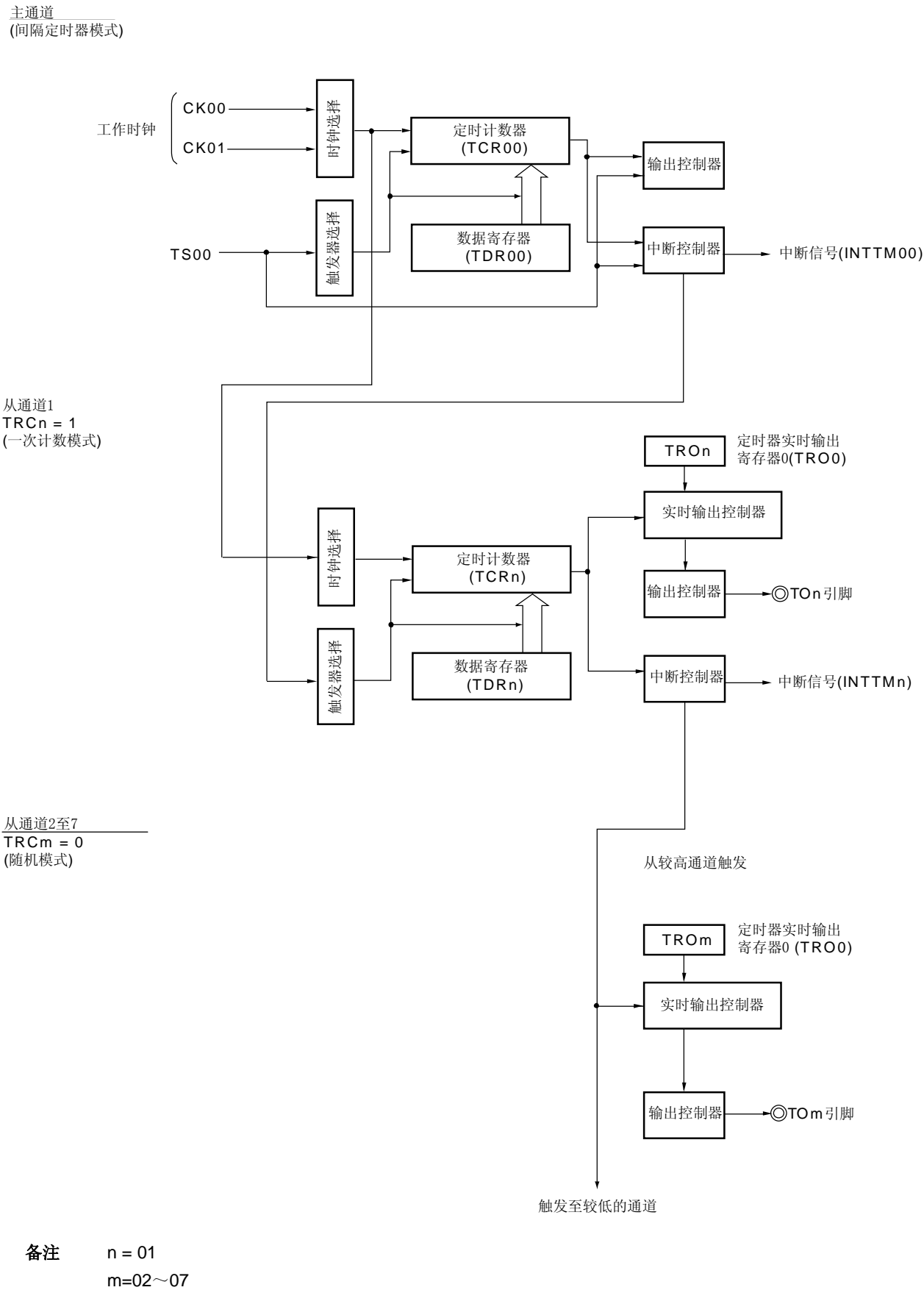
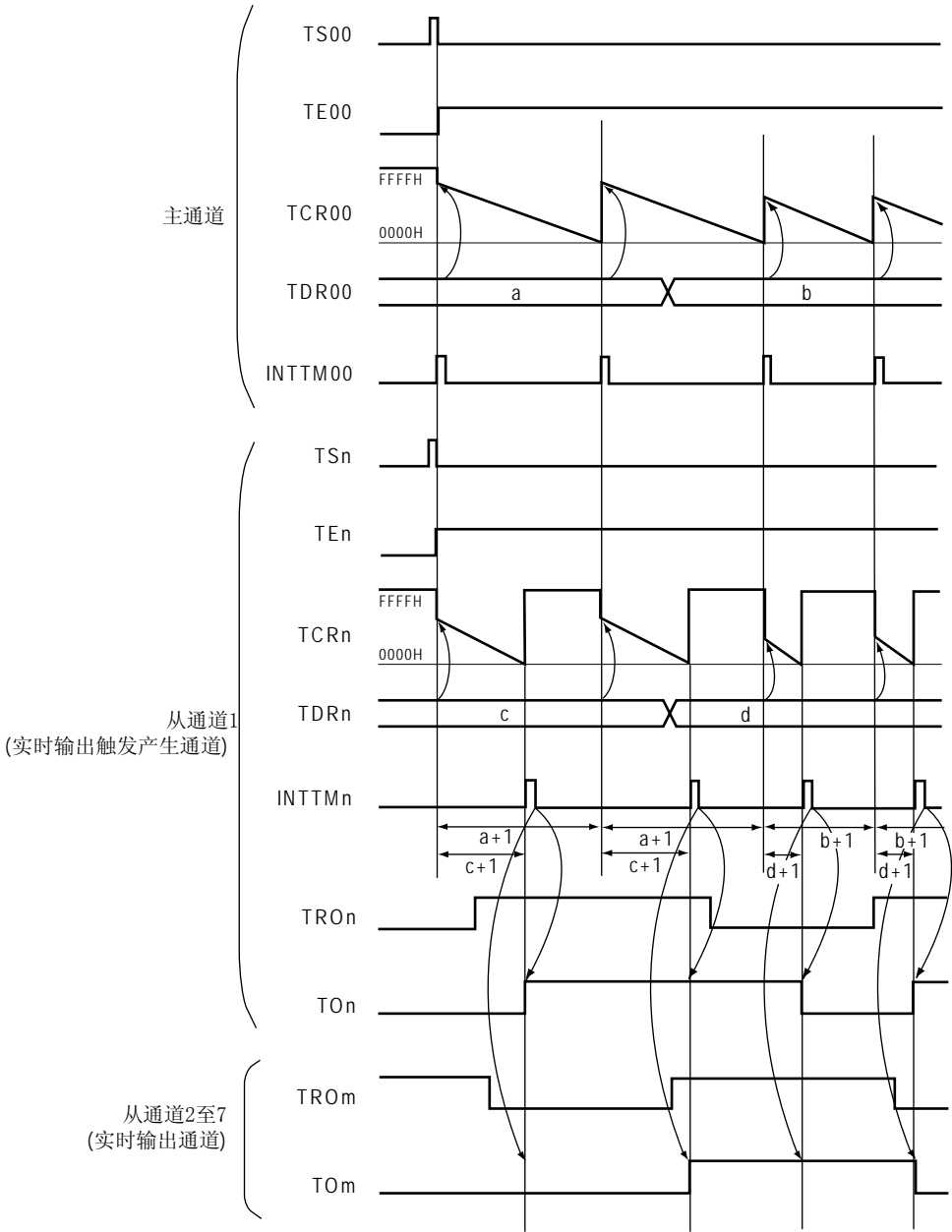


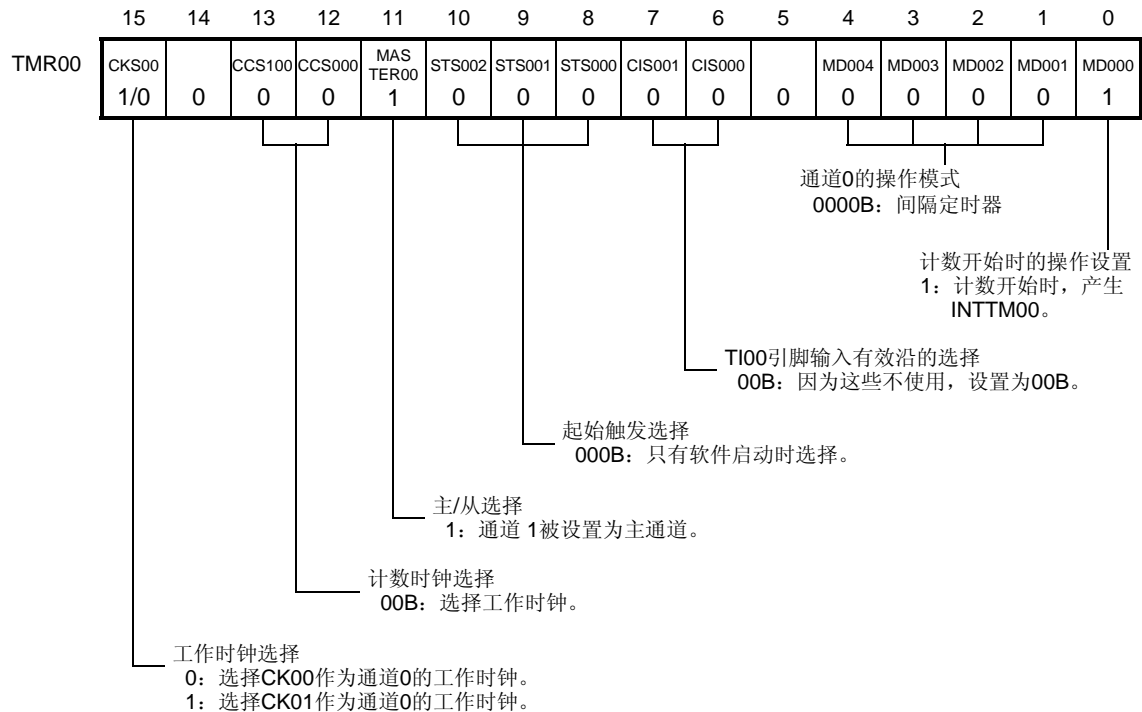
图 7-60 作为实时输出功能基本时序的示例（类型 1）



备注 n = 01
 m=02~07

图 7-61. 当使用实时输出功能时寄存器设置内容的示例(类型 1)(主通道)

(a) 定时器模式寄存器 00(TMR00)

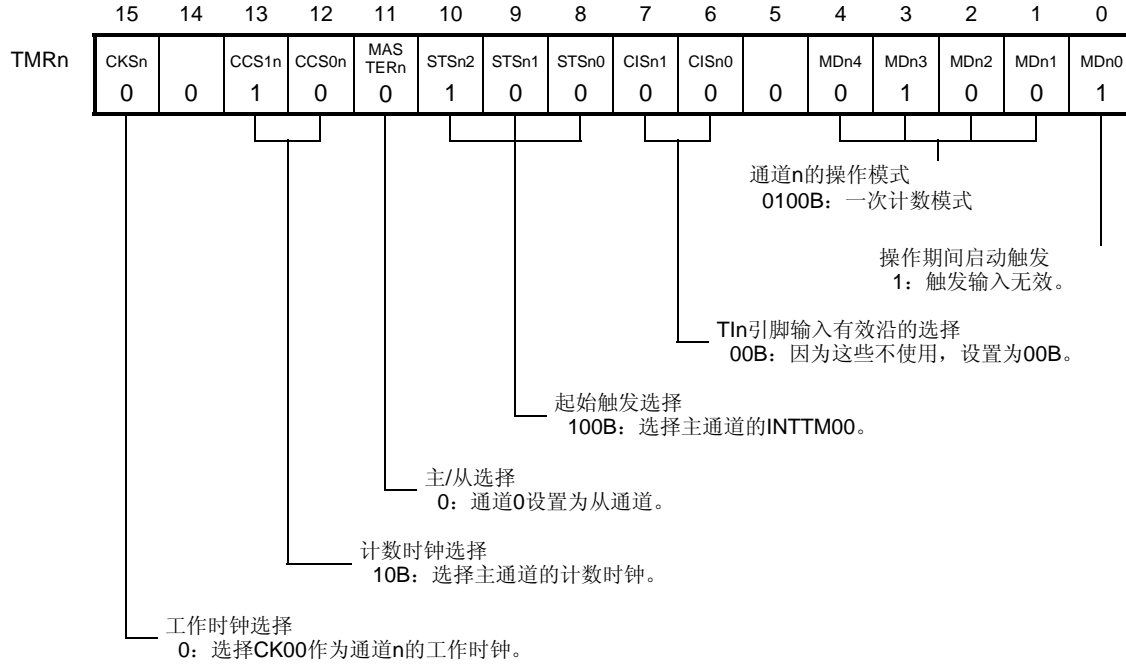


(b) 其它寄存器

TOE0:TOE00	0: 通过计数操作停止 TO00 输出操作。
TO0:TO00	0: 从TO00输出低电平。
TOM0:TOM00	0: 当TOE00 = 0时设为0(通过计数操作停止TO00输出)。
TOT0:TOT00	0: 当TOM0 = 0(反转模式)时设为0。
TOL0:TOL00	0: 当TOM0 = 0(反转模式)时设为0。
TDE0:TDE00	0: 停止死区时间控制。
TRE0:TRE00	0: 停止实时输出。
TRO0:TRO00	0: 当TRE00=0(停止实时输出)时设为0。
TRC0:TRC00	0: 作为实时输出触发产生通道不运行。
TME0:TME00	0: 停止调制输出。

图 7-62. 当使用实时输出功能时寄存器设置内容的示例（类型1）（从通道1）

（a）从通道1（实时输出触发产生通道（TRCn=1））的定时器模式寄存器n（TMRn）



（b）从通道1（实时输出触发产生通道（TRCn=1））的其它寄存器

TOE0:TOEn	0: 通过计数操作, 停止TOn输出。 1: 通过计数操作, 允许TOn输出。
TO0:TOn	0: 从TOn输出低电平。 1: 从TOn输出高电平。
TOM0:TOMn	0: 反转模式
TOT0:TOTn	0: 当TOMn=0(反转模式)时设为0。
TOL0:TOLn	0: 当TOMn=0(反转模式)时设为0。
TDE0:TDEn	0: 停止死区时间控制。
TRE0:TREn	0: 停止实时输出。 1: 允许实时输出。
TRO0:TROn	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCn	1: 作为实时输出触发产生通道运行。
TME0:TMEn	0: 停止调制输出。

备注 n = 01

图 7-63. 当使用实时输出功能时寄存器设置内容的示例（类型 1）（从通道 2~7）

(a) 从通道 2~7（实时输出通道（TRCm=0））的定时器模式寄存器 m（TMRm）

实时输出功能（类型 1）中，当 TRCm 被设为 0 时，通道 TMRm 可任意设置。

(b) 从通道 2~7（实时输出通道（TRCm=0））的其它寄存器

TOE0:TOEm	0: 通过计数操作，停止TOM输出操作。 1: 通过计数操作，允许TOM输出操作。
TO0:TOM	0: 从TOM输出低电平。 1: 从TOM输出高电平。
TOM0:TOMm	0: 当TREm=1时设为0（允许实时输出）。
TOT0:TOTm	0: 当TOMm = 0（反转模式）时设为0。
TOL0:TOLm	0: 当TOMm = 0（反转模式）时设为0。
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	1: 允许实时输出。
TRO0:TROM	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。

备注 m = 02~07

图 7-64. 实时输出功能的操作过程（类型 1）（1/2）

软件操作	硬件状态
TAUS 默认设置	掉电状态 (停止供给时钟，并且所有寄存器不能写入。)
设置PER2寄存器的TAU0EN位和TAUOPEN位为1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
设置TPS0 寄存器。 决定CK00和CK01的时钟频率。	
通道默认设置	通道停止操作。 (供给时钟并且产生一些功耗。)
设置要使用的每个通道的TMR00和TMRn寄存器（决定通道的操作模式）。 一个间隔（周期）值设置到主通道的TDR00寄存器和从通道的TDRn寄存器。	
[实时输出触发产生通道（从通道1）] 设置TRCn位为1（触发产生通道）。 设置TREN位为1（允许实时输出）。 [实时输出通道（从通道2~7）] 设置TRCm位为0（非触发产生通道）。 设置TREM位为1（允许实时输出）。	TON和TOM引脚为高阻输出状态。
设置 TOEn 和 TOEm 位 为 1 并 允 许 TON和TOM输出。	TON和TOM不改变因为通道停止工作。
将端口寄存器和端口模式寄存器清为0。	TON 和 TOM引脚输出TON和TOM的设定值。
操作开始 设置TOEn（从通道1）和TOEm（从通道2~7）位为1 (仅当操作恢复时)。 TS0 寄存器的TS00（主通道）和TSn（从通道1）位同时设置为1。 因为是触发位，TS00和TSn位自动返回为0。	TE00 = 1、 TEn = 1 当主通道开始计数，产生INTTM00。由此中断触发的，从通道1也开始计数。

备注 n = 01
 m=02~07

操作继续（从下一页）

图 7-64. 实时输出功能的操作过程（类型 1）（2/2）

	软件操作	硬件状态
操作期间	<p>TDR00和TDRn 寄存器的设置值可以改变。</p> <p>TCR00和 TCRn寄存器总是可读。</p> <p>TROn和TROm位的设置值可以改变。</p>	<p>主通道的计数器加载TDR00的值到TCR00，并向下计数。当计数值达到TCR00=0000H，产生INTTM00。同时， TDR00寄存器的值加载到TCR00，计数器开始再次向下计数。</p> <p>在从通道 1， TDRn寄存器的值传送到TCRn， 由主通道的INTTM00触发，计数器开始向下计数。当计数器向下计数 0000H，产生INTTMn输出且计数操作停止。之后，重复上面的操作。在INTTMn输出时序下，从通道2~7（实时输出通道）的TROm的设定值从TOM处输出。</p>
操作停止	<p>TT00（主通道）和 TTn（从通道 1）位同时设置为 1。</p> <p>因为是触发位，TT00 和 TTn 位自动返回为 0。</p>	<p>TE00、TEn=0，并且计数操作停止。</p> <p>TCR00和TCRn保持计数值并停止。</p> <p>TOn和TOM输出未被初始化而保持当前状态。</p>
	<p>TOEn和TOEm位被清零且数值被设置到TOn和TOM。</p>	<p>TOn和TOM的设定值初始化TOn和TOM输出。</p>
TAUS 停止	<p>需要保持TOn和TOM引脚输出电平时</p> <p>在把需要保持的值保存到端口寄存器后把TOn和TOM位清为0。</p> <p>不需要保持TOn和TOM引脚输出电平时</p> <p>切换端口模式寄存器到输入模式。</p> <p>PER2寄存器的TAU0EN位和TAUOPEN位被清为0。</p>	<p>TOn和TOM引脚输出电平通过端口功能保持。</p> <p>TOn和TOM引脚输出电平变为高阻输出状态。</p> <p>掉电状态</p> <p>所有电路被初始化，并且每个通道的SFR也被初始化。（TOn和TOM位清为0， TOn和TOM引脚设置为端口模式。）</p>

备注 n = 01
 m=02~07

操作继续（到前一页）

<R> 7.4.11 作为连接实时输出功能的操作（类型 2）

通过使用从通道 1 的 INTTMn 输出（实时输出触发产生通道），可以把从通道 2~7（实时输出通道）的 TROm 的值由 TOm 输出。

此功能中，从通道 1 被用作 **7.4.6 6 相三角形波 PWM 输出功能**中描绘的一个子功能，或者被用作 **7.4.7 中断信号缩短功能**中描绘的一个扩展功能。因此，主通道的设置与 **7.4.6 6 相三角形波 PWM 输出功能**及 **7.4.7 作为中断信号缩短功能**的设置一致。

如果从通道 1 的 TRCn 被设置为 1，由从通道指定的时间数缩短的主通道的 INTTM00，INTTMn 被用作实时输出触发。通过实时输出触发，从通道 2~7 的实时输出通道从 TOm 输出 TROm 的设定值。

缩短的中断次数可以通过下面的表达式来计算。

$$\begin{aligned} & \text{用于缩短的中断次数} = \text{TDRn 的设定值（从通道 1）} \\ & \rightarrow \text{每} \{ \text{TDRn（从通道 1）设定值} + 1 \} \text{次，由从通道的 INTTMn 输出主通道的 INTTM00} \end{aligned}$$

主通道操作在间隔定时器模式并计数周期。

通道启动触发位（TS00）设置为 1 后，在第一个计数时钟 TCR00 加载 TDR00 的值。这时，通过设置 TMR00 的 MD000 为“1”输出 INTTM00。

之后，TCR00 随着计数时钟向下计数。

当 TCR00 变为 0000H 时，在下一个计数时钟 INTTM00 输出。此时 TCR00 再次加载 TDR00 的值。此后继续相同的操作。

从通道 1 生成一个实时输出触发。INTTMn，主通道的 INTTM00 进行缩短控制，通过采用主通道的 INTTM00 作为计数时钟输出和在事件计数器模式中运行输出。通过设置通道启动触发位（TSn）为 1，使得 TCRn 加载 TDRn 的值。当 TCRn 变成 0000H 时，随着主通道的 INTTM00 输出，TCRn 向下计数，且再次加载 TDRn 的值和输出 INTTMn。此后继续相同的操作。把 TROn 和 TROm 的设定值，可以在从通道 1 的 INTTMn 输出时序，由 TOn 和 TOm 输出。

从通道 1（TRCn=1）的低通道（从通道 2~7）的 TOm 是由 TREm 和 TRCm 位进行控制。

当低通道（TRCm=0）的 TREm 为“1”时，此通道作为实时输出通道运行而且在从通道 1 的 INTTMn 输出计时，TOm 输出 TROm 的设定值。当低通道的 TREm 或 TRCm 为 0 或 1，在实时输出触发生成通道的 INTTMn 输出时序中，TOm 不反转。

当此功能被使用时，低通道的 TCRm、TDRm 和 INTTMm 可用于不同功能。

主通道的 TDR00 在下一开始时序后变为有效（主通道 INTTM00 产生）。

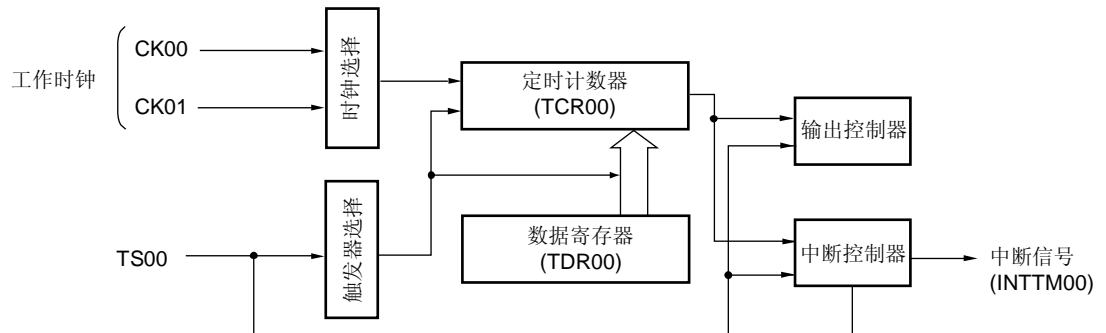
从通道 1 的 TDRn 在下一开始时序后变为有效（从通道 1 INTTMn 产生）。

注意事项 当 TE00 = 1 或 TEn = 1 时，TS00 或 TSn 不能被设为“1”（强制重启）。当 TE00 = 1 或 TEn = 1 时，如果 TS00 或 TSn 被设为“1”，0 计数器的值（TCR00 或 TCRn）将为非法且 TOm 不能输出预期的波形。

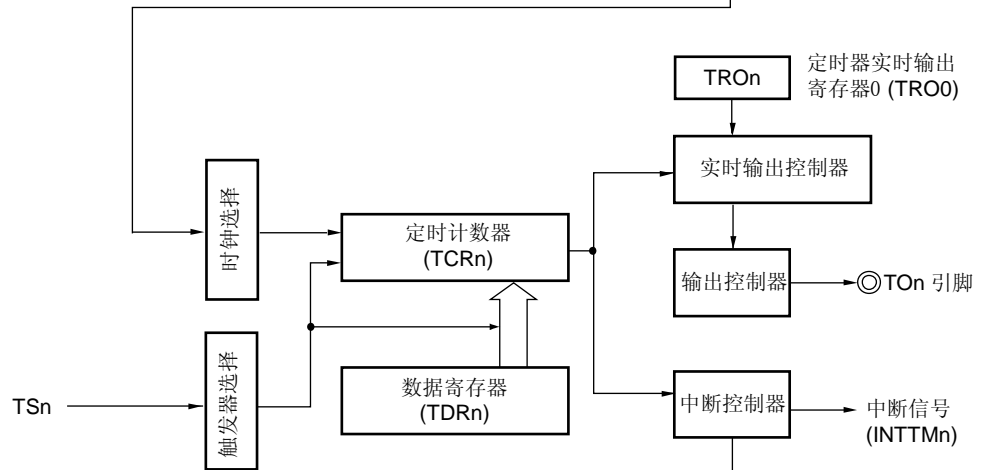
备注 n = 01
m = 02~07

图 7-65 作为实时输出功能操作的框图（类型 2）

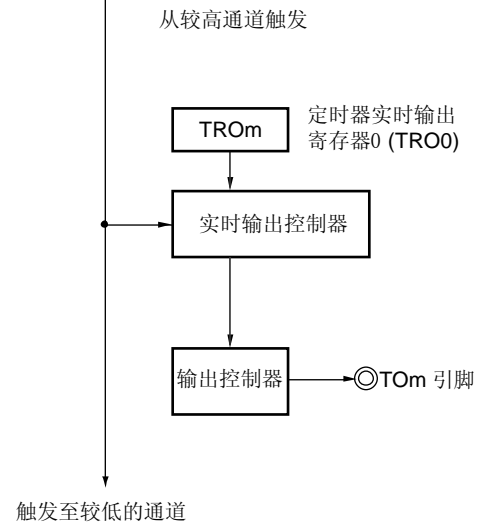
主通道
(间隔定时器模式)



从通道1
TRCn = 1
(事件计数器模式)

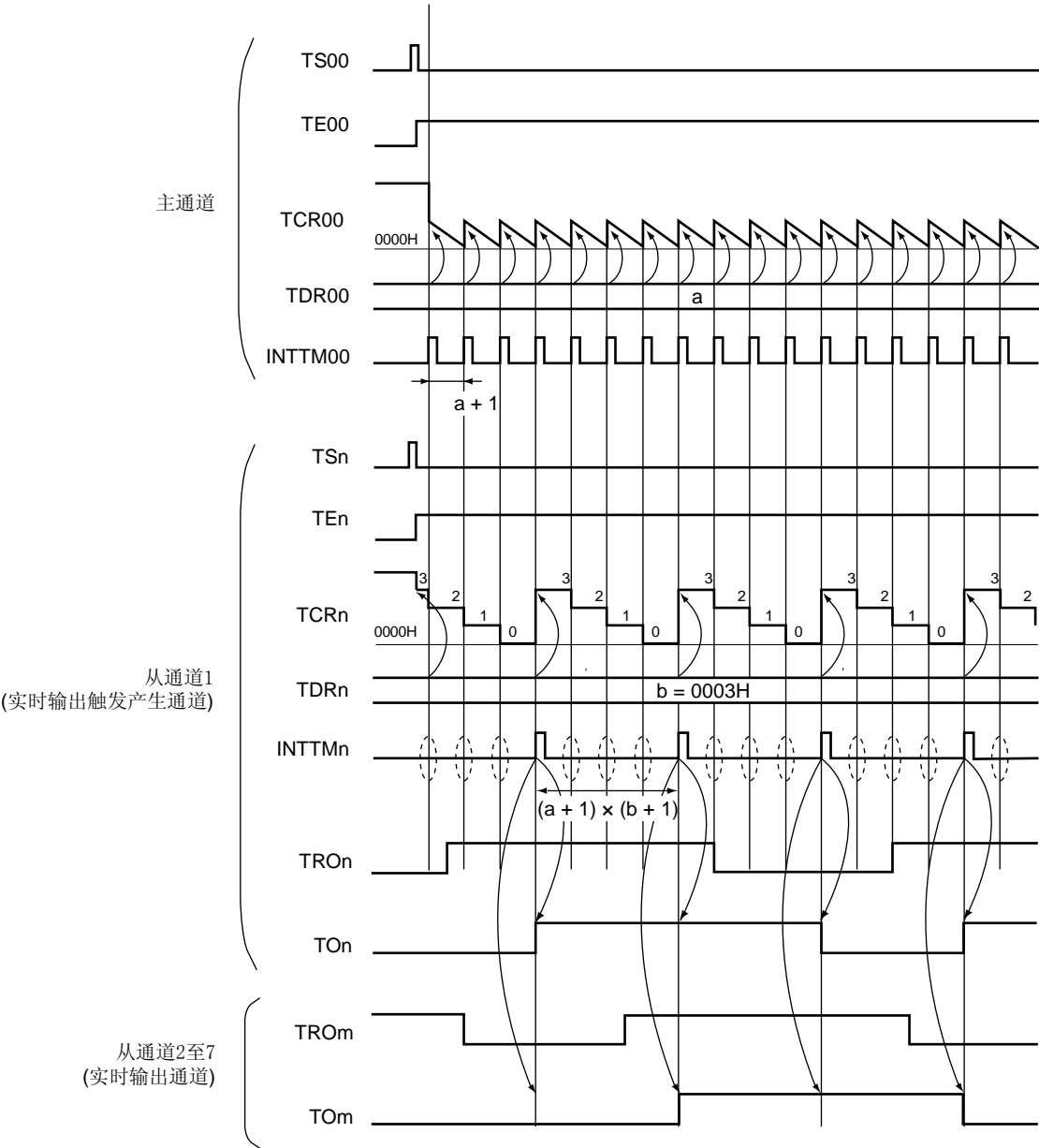


从通道2至7
TRCm = 0
(随机模式)



备注 n = 01
 m=02~07

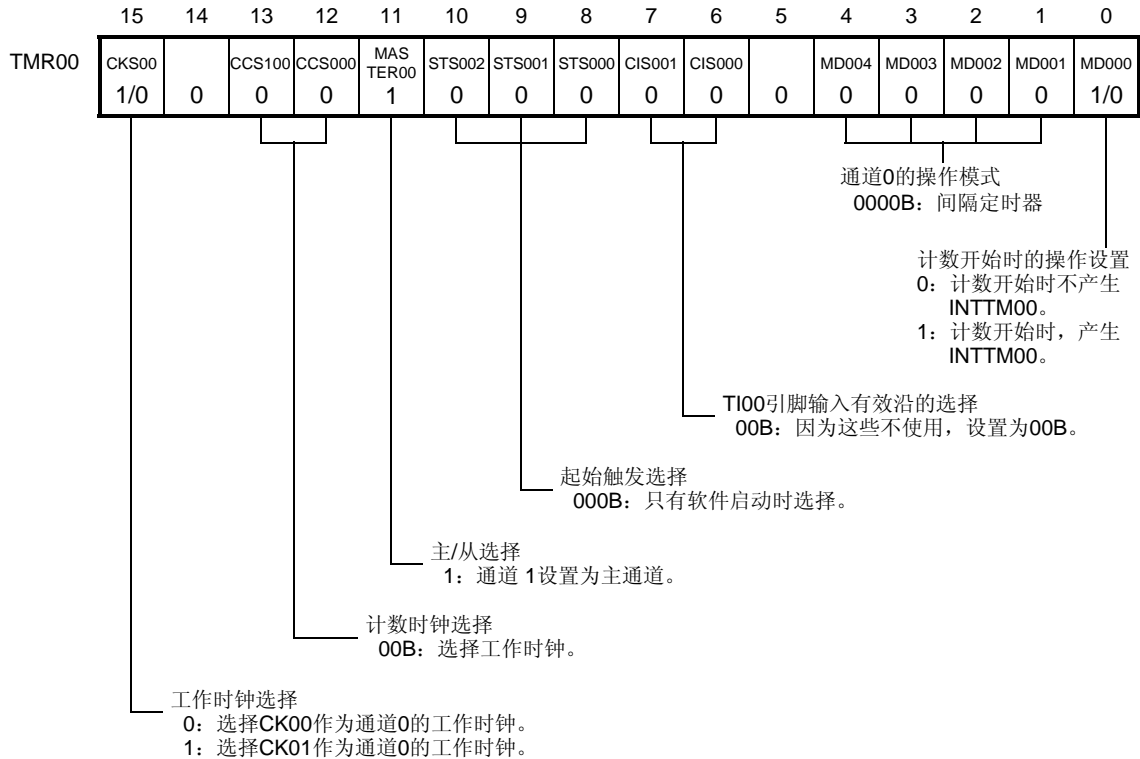
图 7-66 作为实时输出功能基本时序的示例（类型 2）



备注 n = 01
 m=02~07

图 7-67. 当使用实时输出功能时寄存器设置内容的示例（类型 2）（主通道）

(a) 定时器模式寄存器 00 (TMR00)

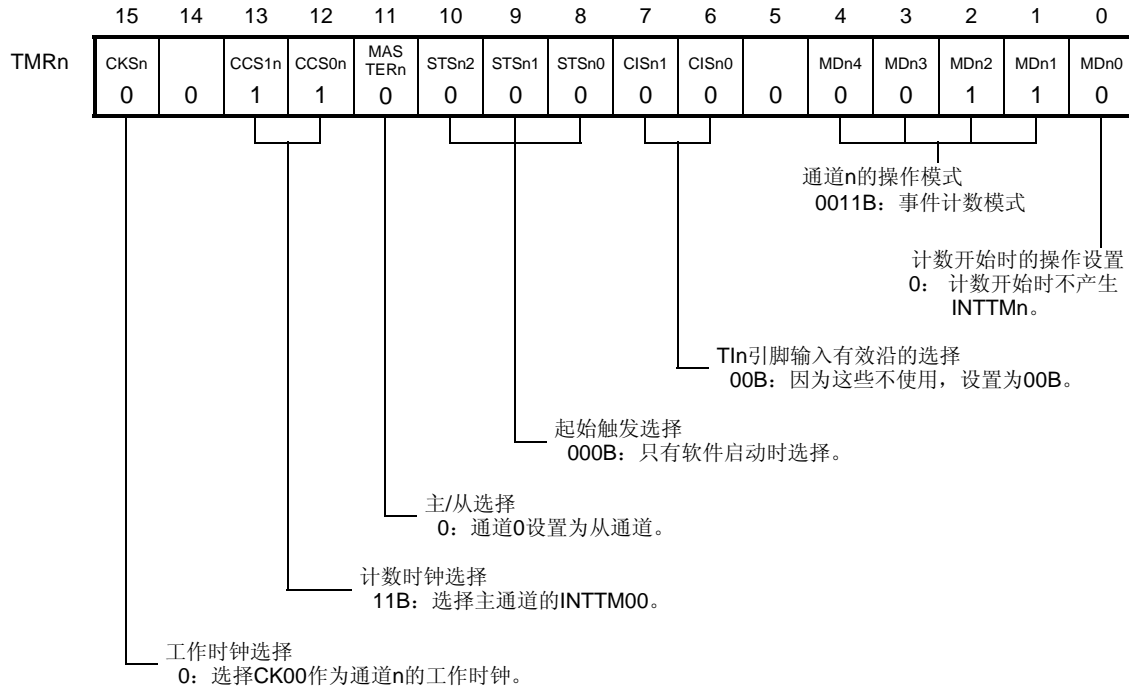


(b) 其它寄存器

TOE0:TOE00	0: 通过计数操作停止 TO00 输出操作。
TO0:TO00	0: 从TO00输出低电平。
TOM0:TOM00	0: 反转模式
TOT0:TOT00	0: 当TOM0 = 0(反转模式)时设为0。
TOL0:TOL00	0: 当TOM0 = 0(反转模式)时设为0。
TDE0:TDE00	0: 停止死区时间控制。
TRE0:TRE00	0: 停止实时输出。
TRO0:TRO00	0: 当TRE00=0（停止实时输出）时设为0。
TRC0:TRC00	0: 作为实时输出触发产生通道不运行。
TME0:TME00	0: 停止调制输出。

图 7-68. 当使用实时输出功能时寄存器设置内容的示例（类型 2）（从通道 1）

(a) 从通道 1（实时输出触发产生通道（TRCn=1））的定时器模式寄存器 n（TMRn）



(b) 从通道 1（实时输出触发产生通道（TRCn=1））的其它寄存器

TOE0:TOEn	0: 通过计数操作，停止TOn输出。 1: 通过计数操作，允许TOn输出。
TO0:TOn	0: 从TOn输出低电平。 1: 从TOn输出高电平。
TOM0:TOMn	0: 反转模式
TOT0:TOTn	0: 当TOMn=0(反转模式)时设为0。
TOL0:TOLn	0: 当TOMn=0(反转模式)时设为0。
TDE0:TDEn	0: 停止死区时间控制。
TRE0:TREn	0: 停止实时输出。 1: 允许实时输出。
TRO0:TROn	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCn	1: 作为实时输出触发产生通道运行。
TME0:TMEn	0: 停止调制输出。

备注 n = 01

图 7-69. 当使用实时输出功能时寄存器设置内容的示例（类型 2）（从通道 2~7）

(a) 从通道 2~7（实时输出通道（TRCm=0））的定时器模式寄存器 m（TMRm）

实时输出功能（类型 2）中，当 TRCm 被设为 0 时，通道 TMRm 可任意设置。

(b) 从通道 2~7（实时输出通道（TRCm=0））的其它寄存器

TOE0:TOEm	0: 通过计数操作，停止TOM输出操作。 1: 通过计数操作，允许TOM输出操作。
TO0:TOM	0: 从TOM输出低电平。 1: 从TOM输出高电平。
TOM0:TOMm	0: 当TREm=1时设为0（允许实时输出）。
TOT0:TOTm	0: 当TOMm = 0(反转模式)时设为0。
TOL0:TOLm	0: 当TOMm = 0(反转模式)时设为0。
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	1: 允许实时输出。
TRO0:TROM	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。

备注 m = 02~07

图 7-70. 实时输出功能的操作过程（类型 2）（1/2）

软件操作	硬件状态
TAUS 默认设置	掉电状态 (停止供给时钟，并且所有寄存器不能写入。)
设置PER2寄存器的TAU0EN位和TAUOPEN位为1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
设置TPS0 寄存器。 决定CK00和CK01的时钟频率。	
通道默认设置	通道停止操作。 (供给时钟并且产生一些功耗。)
设置要使用的每个通道的TMR00和TMRn寄存器（决定通道的操作模式）。 一个间隔（周期）值设置到主通道的TDR00寄存器，用于缩短的中断次数设置到从通道1的TDRn寄存器。	
[实时输出触发产生通道（从通道1）] 设置TRCn位为1（触发产生通道）。 设置TREN位为1（允许实时输出）。 [实时输出通道（从通道2~7）] 设置TRCm位为0（非触发产生通道）。 设置TREM位为1（允许实时输出）。	TON和TOM引脚为高阻输出状态。
设置 TOEn 和 TOEm 位 为 1 并 允 许 TON和TOM输出。将端口寄存器和端口模式寄存器清为0。	TON和TOM不改变因为通道停止工作。 TON 和 TOM引脚输出TON和TOM的设定值。
操作开始 设置TOEn（从通道1）和TOEm（从通道2~7）位为1（仅当操作恢复时）。 TS0 寄存器的TS00（主通道）和TSn（从通道1）位同时设置为1。 因为是触发位，TS00和TSn位自动返回为0。	TE00 = 1、 TEn = 1 在主通道下，通过计数时钟输入，TCR00加载TDR00的值。当TMR00 寄存器的MD000位为1时，INTTM00发生。在从通道下，TCRn加载TDRn的值，并进入检测主通道INTTM00 的等待状态。

备注 n = 01
 m=02~07

操作继续（从下一页）

图 7-70. 实时输出功能的操作过程（类型 2）（2/2）

	软件操作	硬件状态
操作期间	在主通道的INTTM00产生后，TDR00和TDRn寄存器的设置值可被修改。 TCR00和 TCRn寄存器总是可读。 TROn和TROm位的设置值可以改变。	主通道的计数器（TCRn）向下计数。当计数值达到TCR00 = 0000H时，TDR00的值被再次加载到TCR00，并且计数操作继续。通过检测TCR00 = 0000H，INTTM00发生。 每次检测到一个主通道的INTTM00信号时，从通道1的计数器（TCRn）就会向下计数。当计数值达到TCRn=0000H时，TDRn的值被再次加载到TCRn，并且计数操作继续。通过检测TCRn=0000H，产生INTTMn。之后，重复上面的操作。在INTTMn输出时序下，从通道2~7（实时输出通道）的TROm的设定值从TOM处输出。
操作停止	TT00（主通道）和TTn（从通道1）位同时设置为1。 因为是触发位，TT00 和 TTn 位自动返回为 0。	►TE00、TEn=0，并且计数操作停止。 TCR00和TCRn保持计数值并停止。 TROn和TROm输出未被初始化而保持当前状态。 ----- TOEn和TOEm位被清零且数值被设置到TON和TOM。►TON和TOM的设定值初始化TON和TOM输出。
TAUS 停止	需要保持TON和TOM引脚输出电平时 在把需要保持的值保存到端口寄存器后把TON和TOM位清为0。 不需要保持TON和TOM引脚输出电平时 切换端口模式寄存器到输入模式。 PER2寄存器的TAU0EN位和TAUOPEN位被清为0。	►TON和TOM引脚输出电平通过端口功能保持。 ----- ►TON和TOM引脚输出电平变为高阻输出状态。 ----- ►掉电状态 所有电路被初始化，并且每个通道的SFR也被初始化。 （TON和TOM位清为0，TON和TOM引脚设置为端口模式。）

备注 n = 01
 m=02~07

操作继续（到前一页）

<R> 7.4.12 作为连接实时输出功能的操作（类型 3）

通过使用从通道 1 的 INTTMn 输出（实时输出触发产生通道），可以把从通道 2~7（实时输出通道）的 TROm 的值由 TOn 输出。

此功能中，从通道 1 被用作 7.4.6 6 相三角波 PWM 输出功能中描绘的一个子功能使用。因此，主通道的设置与 7.4.6 6 相三角波 PWM 输出功能中的设置一致。

如果从通道 1 的 TRCn 被设置为 1，通过与主通道的组合操作产生的 INTTMn 被用作实时输出触发。通过实时输出触发，从通道 2~7 的实时输出通道从 TOn 输出 TROm 的设定值。

通过使用主通道的 INTTM00 作为从通道 1 的计数时钟，此功能启动一个由软件控制的计数操作。将 TSn 设为 1 后，中断产生周期可由下列表达式计算。

$$\begin{aligned} & \text{从 TSn 设为“1”至 INTTMn 输出的周期} \\ &= \{ \text{TDRn (从通道 1) 的设置值} + 1 \} \times \text{从通道 1 的计数时钟周期} \\ &= (\text{TDRn (从通道 1) 的设置值} + 1) \times (\text{TDR00 (主通道) 的设置值} + 1) \times \text{主通道的计数时钟周期} \end{aligned}$$

备注 一个少于从通道 1 的计数时钟周期的采样误差发生，因为 TSn = 1 的设置保持挂起，直至从通道 1 的计数时钟发生。

主通道操作在间隔定时器模式并计数周期。

通道启动触发位 (TS00) 设置为 1 后，在第一个计数时钟 TCR00 加载 TDR00 的值。这时，通过设置 TMR00 的 MD000 为“1”输出 INTTM00。

之后，TCR00 随着计数时钟向下计数。

当 TCR00 变为 0000H 时，在下一个计数时钟 INTTM00 输出。此时 TCR00 再次加载 TDR00 的值。此后继续相同的操作。

从通道 1 的 TCRn 运行于一次计数模式，并产生一个实时输出触发。通过设置从通道的 TSn = 1 作为启动触发，TCRn 加载 TDRn 的值至 TCRn，且根据主通道的 INTTM00 输出，向下计数。当 TCRn = 0000H 时，将输出 INTTMn 并且停止计数直到下一个启动触发 (TSn = 1) 输入。当 TSn 第二次设置为“1”且从通道 1 在运行时（此时 TCRn 向下计数），将忽略第二个开始触发。INTTMn 输出后，TSn 设为 1。把 TROn 和 TROm 的设定值，可以在从通道 1 的 INTTMn 输出时序，由 TOn 和 TOm 输出。

从通道 1 (TRCn=1) 的低通道（从通道 2~7）的 TOn 是由 TREm 和 TRCm 位进行控制。

当低通道 (TRCm=0) 的 TREm 为“1”时，此通道作为实时输出通道运行而且在从通道 1 的 INTTMn 输出计时，TOn 输出 TROm 的设定值。当低通道的 TREm 或 TRCm 为 0 或 1，在实时输出触发生成通道的 INTTMn 输出时序中，TOn 不反转。

当此功能被使用时，低通道的 TCRm、TDRm 和 INTTMm 可用于不同功能。

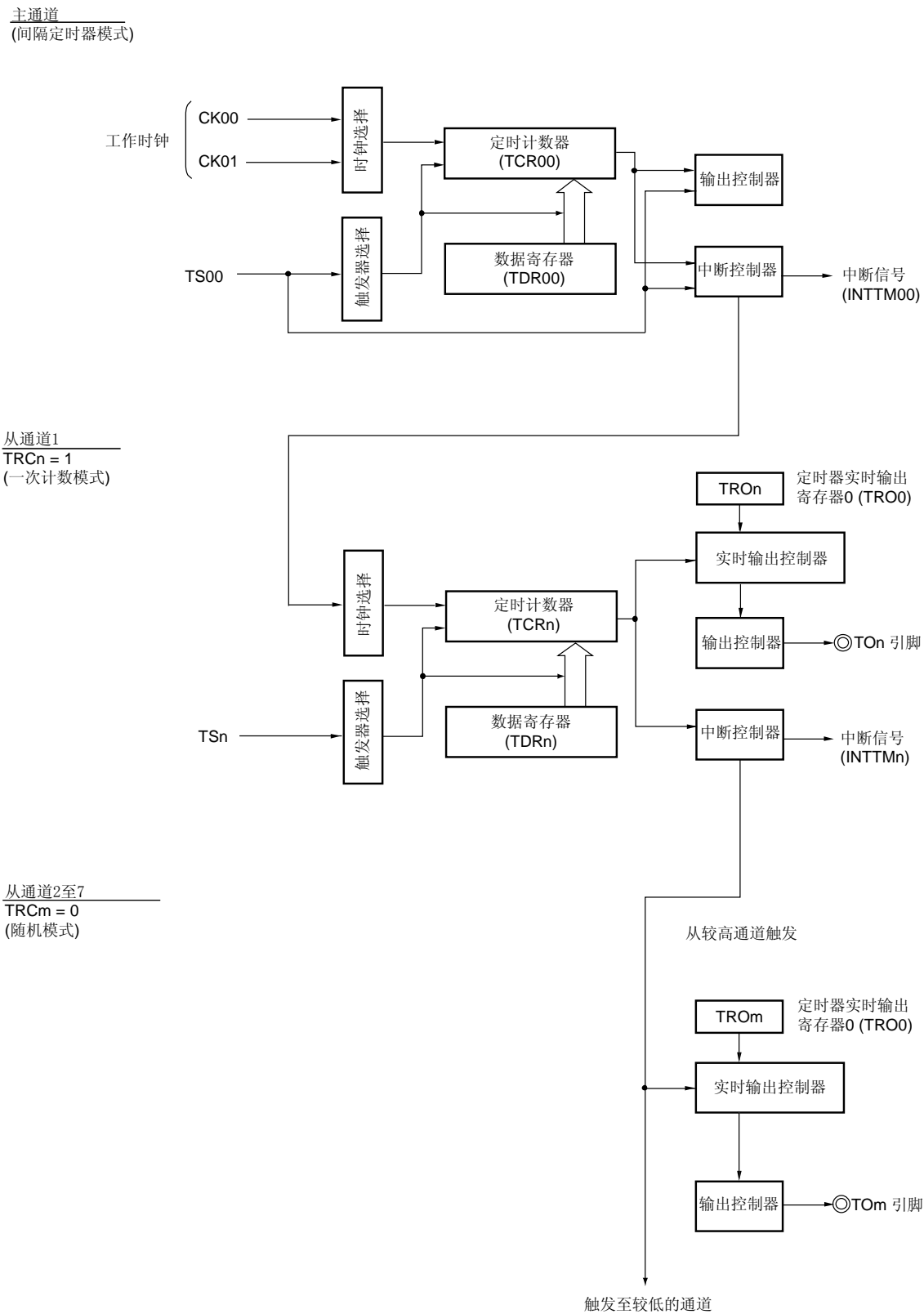
备注 n = 01
m = 02~07

主通道的 TDR00 在下一开始时序后变为有效（主通道 INTTM00 产生）。
从通道 1 的 TDRn 在下一开始时序后变为有效（设置从通道 1 的 TSn = 1）。

注意事项 TE00 为“1”期间，TS00 不能设为“1”（强制重启）。当 TE00 为“1”时，如果 TS00 被设为“1”，计数器的值（TCR00）将为非法且 T0m 不能输出一个预期的波形。当 TE00 和 TEn 为“1”，从通道 1 的 TSn 可被操作。

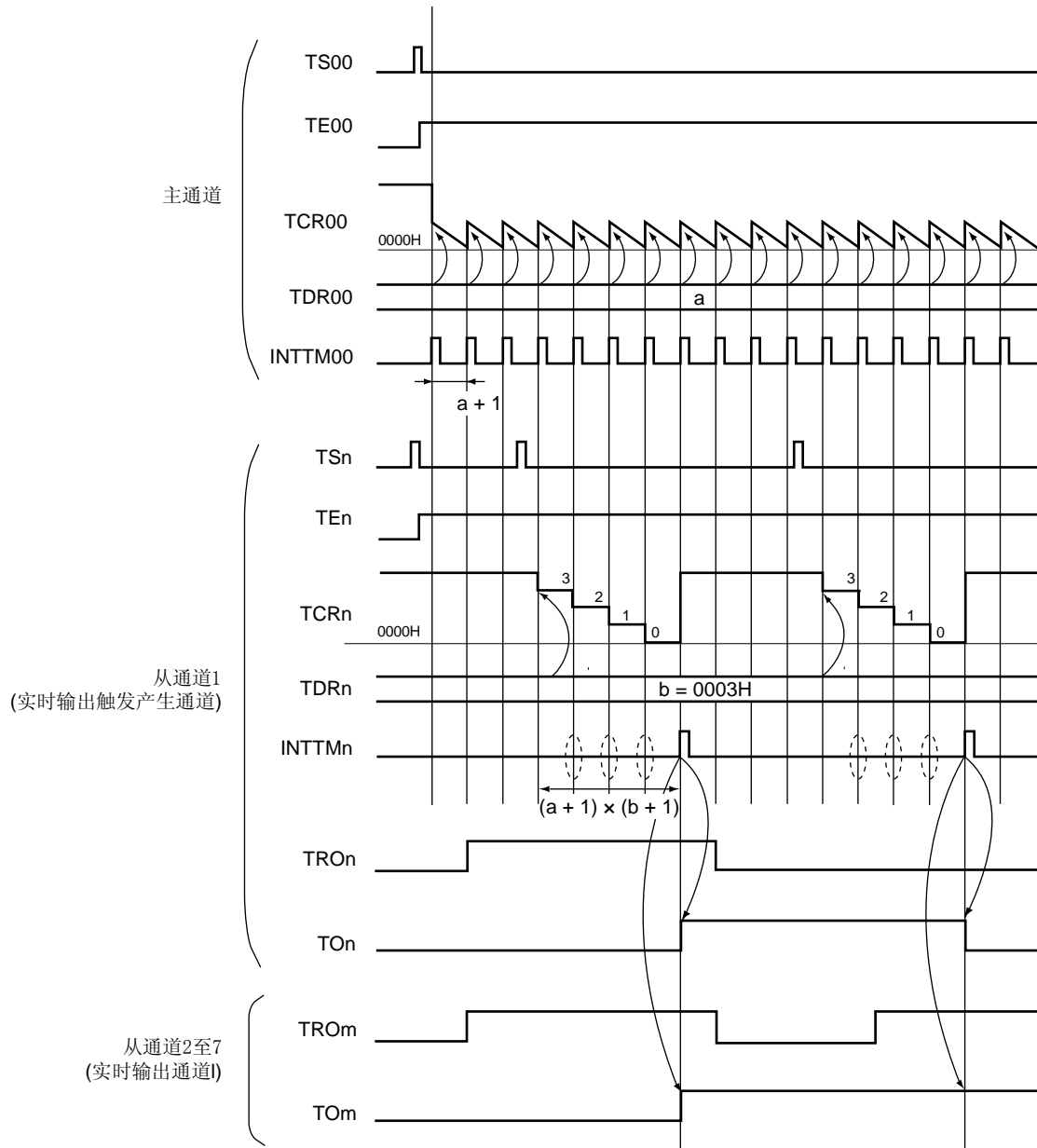
备注 n = 01
m=02~07

图 7-71 作为实时输出功能操作的框图（类型 3）



备注 n = 01
m=02~07

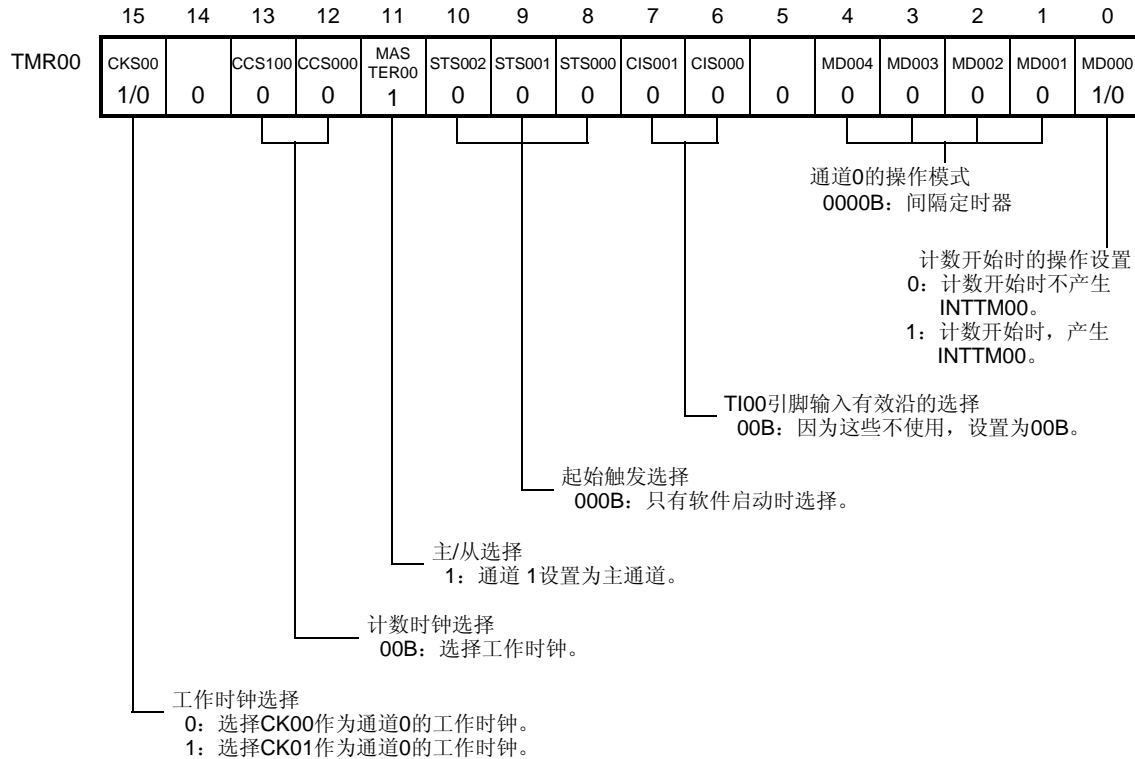
图 7-72 作为实时输出功能基本时序的示例（类型 3）



备注 n = 01
 m=02~07

图 7-73. 当使用实时输出功能时寄存器设置内容的示例（类型 3）（主通道）

(a) 定时器模式寄存器 00 (TMR00)

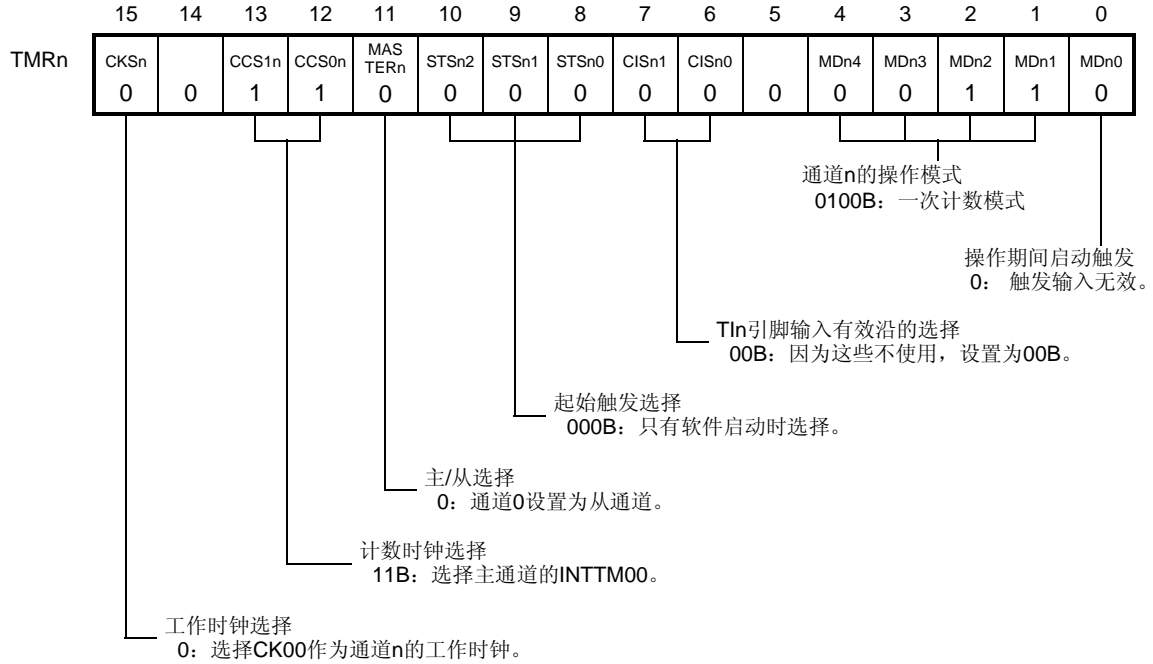


(b) 其它寄存器

TOE0:TOE00	0: 通过计数操作停止 TO00 输出操作。
TO0:TO00	0: 从TO00输出低电平。
TOM0:TOM00	0: 当TOE00 = 0时设为0(通过计数操作停止TO00输出)。
TOT0:TOT00	0: 当TOM00 = 0(反转模式)时设为0。
TOL0:TOL00	0: 当TOM00 = 0(反转模式)时设为0。
TDE0:TDE00	0: 停止死区时间控制。
TRE0:TRE00	0: 停止实时输出。
TRO0:TRO00	0: 当TRE00=0(停止实时输出)时设为0。
TRC0:TRC00	0: 作为实时输出触发产生通道不运行。
TME0:TME00	0: 停止调制输出。

图 7-74. 当使用实时输出功能时寄存器设置内容的示例（类型 3）（从通道 1）

(a) 从通道 1（实时输出触发产生通道（TRCn=1））的定时器模式寄存器 n（TMRn）



(b) 从通道 1（实时输出触发产生通道（TRCn=1））的其它寄存器

TOE0:TOEn	0: 通过计数操作, 停止TOn输出。 1: 通过计数操作, 允许TOn输出。
TO0:TOn	0: 从TOn输出低电平。 1: 从TOn输出高电平。
TOM0:TOMn	0: 反转模式
TOT0:TOTn	0: 当TOMn=0(反转模式)时设为0。
TOL0:TOLn	0: 当TOMn=0(反转模式)时设为0。
TDE0:TDEn	0: 停止死区时间控制。
TRE0:TREn	0: 停止实时输出。 1: 允许实时输出。
TRO0:TROn	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCn	1: 作为实时输出触发产生通道运行。
TME0:TMEn	0: 停止调制输出。

备注 n = 01

图 7-75. 当使用实时输出功能时寄存器设置内容的示例（类型 3）（从通道 2~7）

(a) 从通道 2~7（实时输出通道（TRCm=0））的定时器模式寄存器 m（TMRm）

实时输出功能（类型 3）中，当 TRCm 被设为 0 时，通道 TMRm 可任意设置。

(b) 从通道 2~7（实时输出通道（TRCm=0））的其它寄存器

TOE0:TOEm	0: 通过计数操作，停止TOM输出操作。 1: 通过计数操作，允许TOM输出操作。
TO0:TOM	0: 从TOM输出低电平。 1: 从TOM输出高电平。
TOM0:TOMm	0: 当TREm=1时设为0（允许实时输出）。
TOT0:TOTm	0: 当TOMm = 0(反转模式)时设为0。
TOL0:TOLm	0: 当TOMm = 0(反转模式)时设为0。
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	1: 允许实时输出。
TRO0:TROm	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。

备注 m = 02~07

图 7-76. 实时输出功能的操作过程（类型 3）（1/2）

软件操作		硬件状态	
TAUS 默认设置		掉电状态 (停止供给时钟, 并且所有寄存器不能写入。)	
	设置PER2寄存器的TAU0EN位和TAUOPEN位为1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)	
	设置TPS0 寄存器。		
	决定CK00和CK01的时钟频率。		
通道默认设置	设置要使用的每个通道的TMR00和TMRn寄存器 (决定通道的操作模式)。	通道停止操作。 (供给时钟并且产生一些功耗。)	
	一个间隔 (周期) 值设置到主通道的TDR00寄存器, 且从设置TSn至“1”直到INTTMn输出为止的周期设置到从通道1的TDRn寄存器中。		
	[实时输出触发产生通道 (从通道1)] 设置TRCn位为1 (触发产生通道)。 设置TREN位为1 (允许实时输出)。	TOn和TOm引脚为高阻输出状态。	
	[实时输出通道 (从通道2~7)] 设置TRCm位为0 (非触发产生通道)。 设置TREM位为1 (允许实时输出)。		
操作开始	设置TOEn和TOEm位为1并允许TOn和TOm输出。	TOn和TOm不改变因为通道停止工作。	
	将端口寄存器和端口模式寄存器清为0。	TOn 和 TOm引脚输出TOn和TOm的设定值。	
	设置TOEn (从通道1) 和TOEm (从通道2~7) 位为1 (仅当操作恢复时)。		
	TS0 寄存器的TS00 (主通道) 和TSn (从通道1) 位同时设置为1。 因为是触发位, TS00和TSn位自动返回为0。	TE00 = 1、TEn = 1 当主通道开始计数, 产生INTTM00。 在从通道1, 进入一个启动触发等待状态 (TSn = 1)。	

备注 n = 01
 m=02~07

操作继续 (从下一页)

图 7-77. 实时输出功能的操作过程（类型 3）（2/2）

软件操作		硬件状态
操作期间	TDR00和TDRn 寄存器的设置值可以改变。 TCR00和 TCRn寄存器总是可读。 TROn和TROm位的设置值可以改变。	主通道的计数器加载TDR00的值到TCR00，并向下计数。当计数值达到TCR00=0000H，产生INTTM00。同时， TDR00寄存器的值加载到TCR00，计数器开始再次向下计数。 在从通道1， TDRn寄存器的值传送到TCRn，由TSn = 1触发，而且计数器开始向下计数。当计数器向下计数 0000H，产生INTTMn输出且计数操作停止。之后，重复上面的操作。在INTTMn输出时序下，从通道2~7（实时输出通道）的TROm的设定值从TOM处输出。
操作停止	TT00（主通道）和TTn（从通道1）位同时设置为1。 因为是触发位， TT00 和 TTn 位自动返回为 0。——▶TE00、TEn=0，并且计数操作停止。 TCR00和TCRn保持计数值并停止。 TOn和TOM输出未被初始化而保持当前状态。 ----- TOEn和TOEm位被清零且数值被设置到TOn和TOM。——▶TOn和TOM的设定值初始化TOn和TOM输出。	
TAUS 停止	需要保持TOn和TOM引脚输出电平时 在把需要保持的值保存到端口寄存器后把TOn和TOM 位清为0。 ——▶TOn和TOM引脚输出电平通过端口功能保持。 不需要保持TOn和TOM引脚输出电平时 切换端口模式寄存器到输入模式。 ——▶TOn和TOM引脚输出电平变为高阻输出状态。 ----- PER2寄存器的TAU0EN位和TAUOPEN位被清为0。——▶掉电状态 所有电路被初始化，并且每个通道的SFR也被初始化。 （TOn和TOM位清为0， TOn和TOM引脚设置为端口模式。）	

备注 n = 01
 m=02~07

操作继续（到前一页）

<R>

7.4.13 作为非互补调制输出功能的操作（类型 1）

为执行非互补调制输出，必须从霍尔传感器检测出一个触发器信号，且软件操作必须被执行。

非互补调制输出功能为一个周期设置了 6 个占空宽度，而且产生一个 6 相 PWM 输出。准备 1 个周期生成通道和 6 个占空比生成通道（6 相 PWM 输出功能）。关于 6 相 PWM 功能的详细信息，请参见 7.4.3 作为 6 相 PWM 输出功能操作。

而且，根据对应于产生 PWM 的 T0m 引脚的 TROM 的值产生实时输出。准备了一个实时输出触发产生通道。关于实时输出触发的详细信息，请参见 7.4.10 作为连接实时输出功能的操作（类型 1）。

非互补调制输出由定时器输出（PWM）产生，由 TROM 和 TME m 操作。

主通道操作在间隔定时器模式并计数周期。

从通道 1 运行于一次计数模式，并产生一个实时输出触发。为匹配实时输出触发产生时序（TROM 反射时序）和由从通道 2~7 产生的 PWM 的有效时序，设置 TDR01 为 0000H 并操作从通道 1。

在一次计数模式中运行从通道 2~7，计数占空比，并产生 6 相 PWM 波形。

而且，模組输出可通过操作 TME m 来控制。当 TME m 为“1”时，可由 T0m 引脚输出的 PWM 的逻辑与和实时输出来执行非互补调制输出。当 TME m 为“0”时，将从 T0m 引脚中输出实时输出设定值（TROM）。

要调制 PWM 和实时输出并将二者从 T0m 引脚输出，操作必须从低电平的从通道 2~7 的 T0m 的默认电平开始。假设从通道 2 至 7 的 TOLm 固定为“0”，且若需要反相控制时使用 TLS2 至 TLS7 (OPMR 寄存器)。

PWM 的调制控制和实时输出是在 TROM 为“1”的基础上进行的（有效电平：高电平）。PWM 输出的调制输出不能在 TROM 设为“0”时进行。

注意事项 当 TE00 = 1、TE01 = 1 或 TEm = 1 时，TS00、TS01 或 TSm 不能被设为“1”（强制重启）。当 TE00 = 1、TE01 = 1 或 TEm = 1 时，若 TS00、TS01 或 TSm 被设为“1”，则计数器的值（TCR00、TCR01 或 TCRm）将非法且 T0m 将不能输出预期波形。

备注 m = 02~07

图 7-78 作为非互补调制输出功能操作的框图（类型 1）

主通道
(间隔定时器模式)

从通道1
TRC01 = 1
(一次计数模式)

从通道2
TRC02 = 0
(一次计数模式)

从通道7
TRC07 = 0
(一次计数模式)

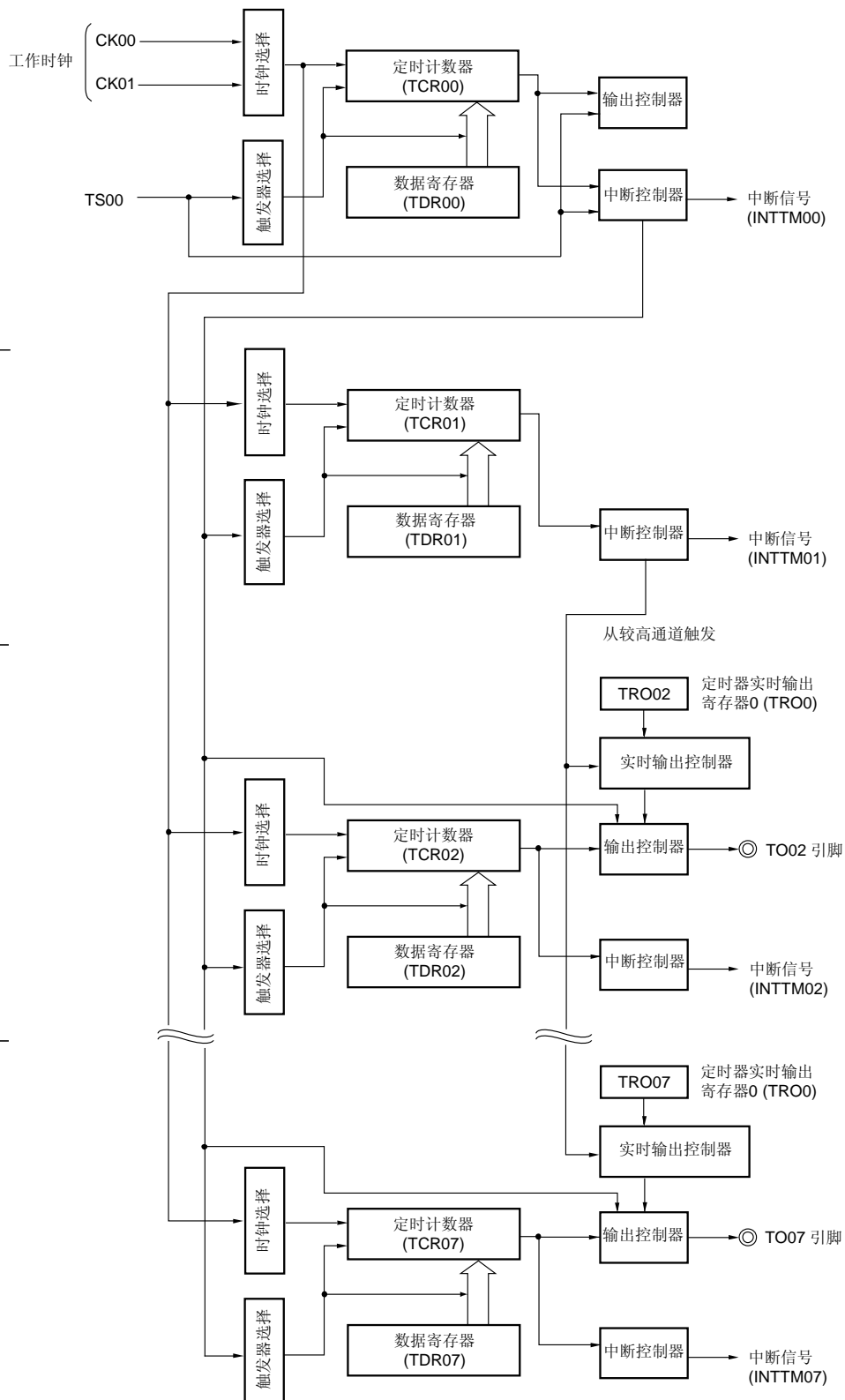
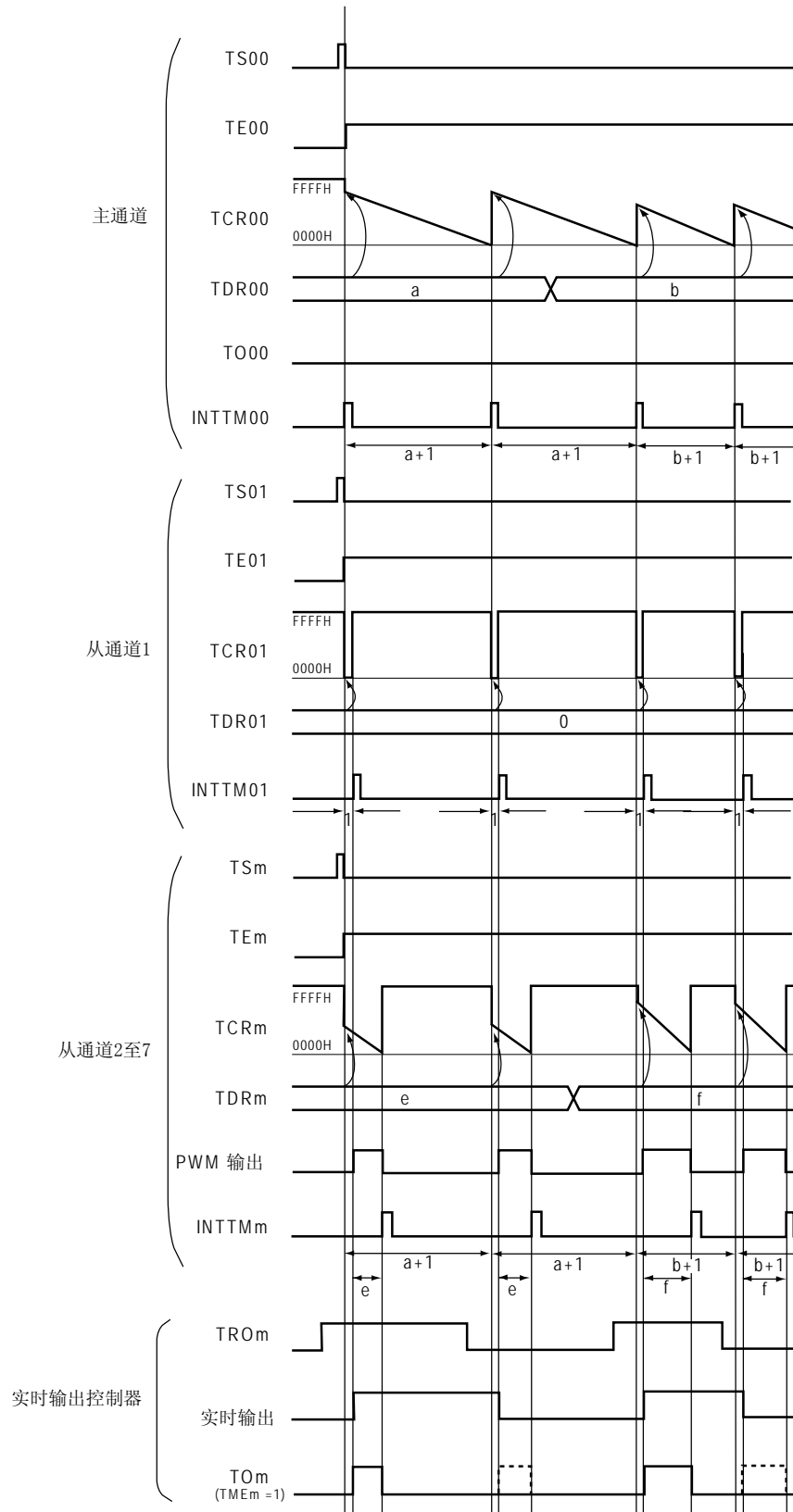


图 7-79 作为非互补调制输出功能操作的基本时序示例（类型 1）



备注 m=02~07

120° 磁控制期间，非互补调制输出显示如下。

为了在全调制、前半调制和后半调制之间切换，TROm 和 TME m 必须在运行期间被操作。

表 7-2. 120° 磁控制期间，非互补输出列表（全调制）

控制寄存器	设置 TROm、TME m 的值						TOm 输出					
	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7
TROm	0	1	1	0	0	0	低	PWM	PWM	低	低	低
TME m	1	1	1	1	1	1						
TROm	0	1	0	0	1	0	低	PWM	低	低	PWM	低
TME m	1	1	1	1	1	1						
TROm	0	0	0	1	1	0	低	低	低	PWM	PWM	低
TME m	1	1	1	1	1	1						
TROm	1	0	0	1	0	0	PWM	低	低	PWM	低	低
TME m	1	1	1	1	1	1						
TROm	1	0	0	0	0	1	PWM	低	低	低	低	PWM
TME m	1	1	1	1	1	1						
TROm	0	0	1	0	0	1	低	低	PWM	低	低	PWM
TME m	1	1	1	1	1	1						

表 7-3. 120° 磁控制期间，非互补输出列表（前半调制）

控制寄存器	设置 TROm、TME m 的值						TOm 输出					
	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7
TROm	0	1	1	0	0	0	低	PWM	高	低	低	低
TME m	0	1	0	0	0	0						
TROm	0	1	0	0	1	0	低	高	低	低	PWM	低
TME m	0	0	0	0	1	0						
TROm	0	0	0	1	1	0	低	低	低	PWM	高	低
TME m	0	0	0	1	0	0						
TROm	1	0	0	1	0	0	PWM	低	低	高	低	低
TME m	1	0	0	0	0	0						
TROm	1	0	0	0	0	1	高	低	低	低	低	PWM
TME m	0	0	0	0	0	1						
TROm	0	0	1	0	0	1	低	低	PWM	低	低	高
TME m	0	0	1	0	0	0						

备注 m = 02~07

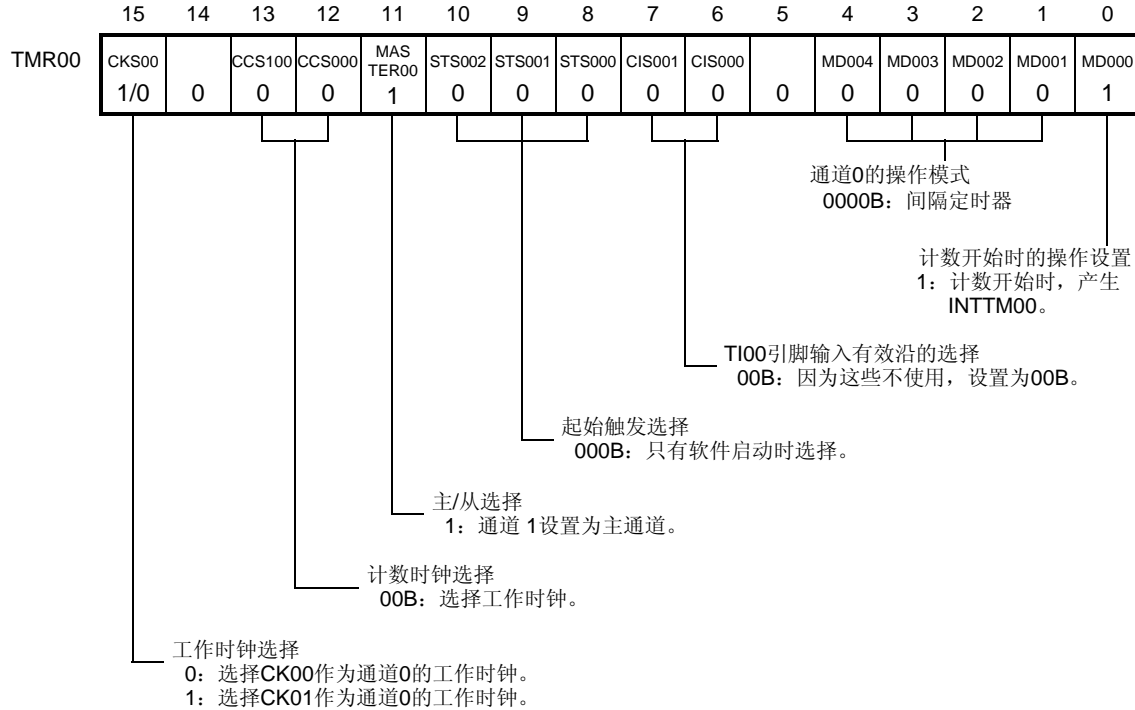
表 7-4. 120° 磁控制期间，非互补输出列表（后半调制）

控制寄存器	设置 TROm、TMEm 的值						TOM 输出					
	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7
TROm	0	1	1	0	0	0	低	高	PWM	低	低	低
TMEm	0	0	1	0	0	0						
TROm	0	1	0	0	1	0	低	PWM	低	低	高	低
TMEm	0	1	0	0	0	0						
TROm	0	0	0	1	1	0	低	低	低	高	PWM	低
TMEm	0	0	0	0	1	0						
TROm	1	0	0	1	0	0	高	低	低	PWM	低	低
TMEm	0	0	0	1	0	0						
TROm	1	0	0	0	0	1	PWM	低	低	低	低	高
TMEm	1	0	0	0	0	0						
TROm	0	0	1	0	0	1	低	低	高	低	低	PWM
TMEm	0	0	0	0	0	1						

备注 m = 02~07

图 7-80. 当使用非互补调制输出功能时寄存器设置内容的示例（类型 1）（主通道）

(a) 定时器模式寄存器 00(TMR00)

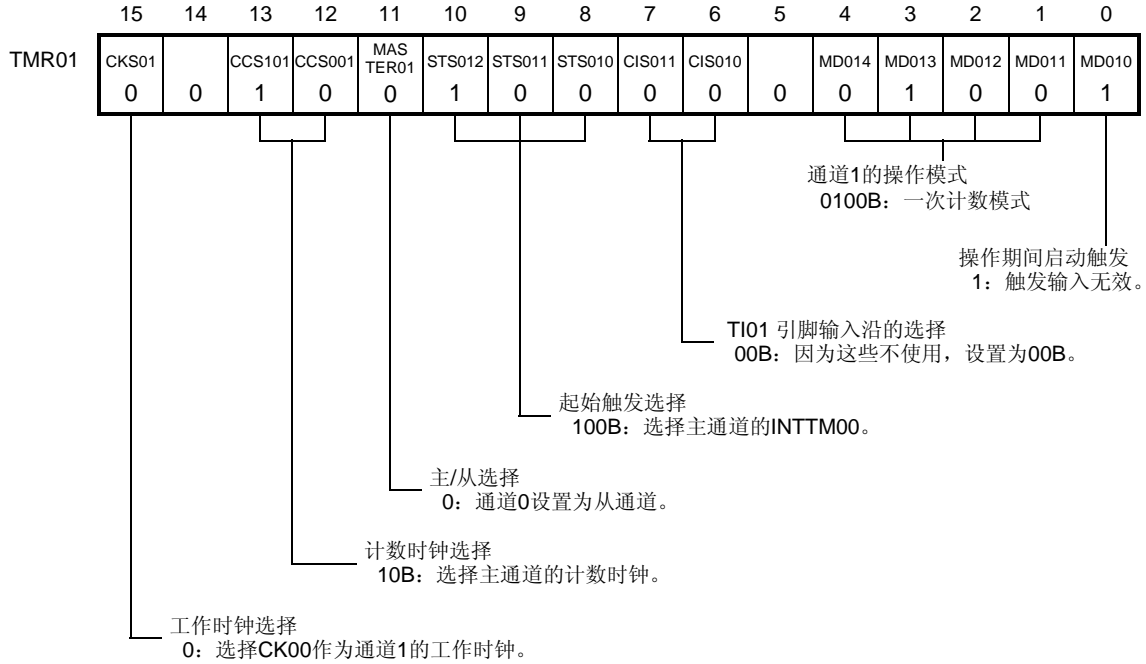


(b) 其它寄存器

TOE0:TOE00	0: 通过计数操作停止 TO00 输出操作。
TO0:TO00	0: 从TO00输出低电平。
TOM0:TOM00	0: 当TOE00 = 0时设为0（通过计数操作停止TO00输出）。
TOT0:TOT00	0: 当TOM00 = 0(反转模式)时设为0。
TOL0:TOL00	0: 当TOM00 = 0(反转模式)时设为0。
TDE0:TDE00	0: 停止死区时间控制。
TRE0:TRE00	0: 停止实时输出。
TRO0:TRO00	0: 当TRE00=0（停止实时输出）时设为0。
TRC0:TRC00	0: 作为实时输出触发产生通道不运行。
TME0:TME00	0: 停止调制输出。

图 7-81. 当使用非互补调制输出功能时寄存器设置内容的示例（类型 1）（从通道 1）

(a) 从通道 1（实时输出触发产生通道（TRC01 = 1））的定时器模式寄存器 01（TMR01）

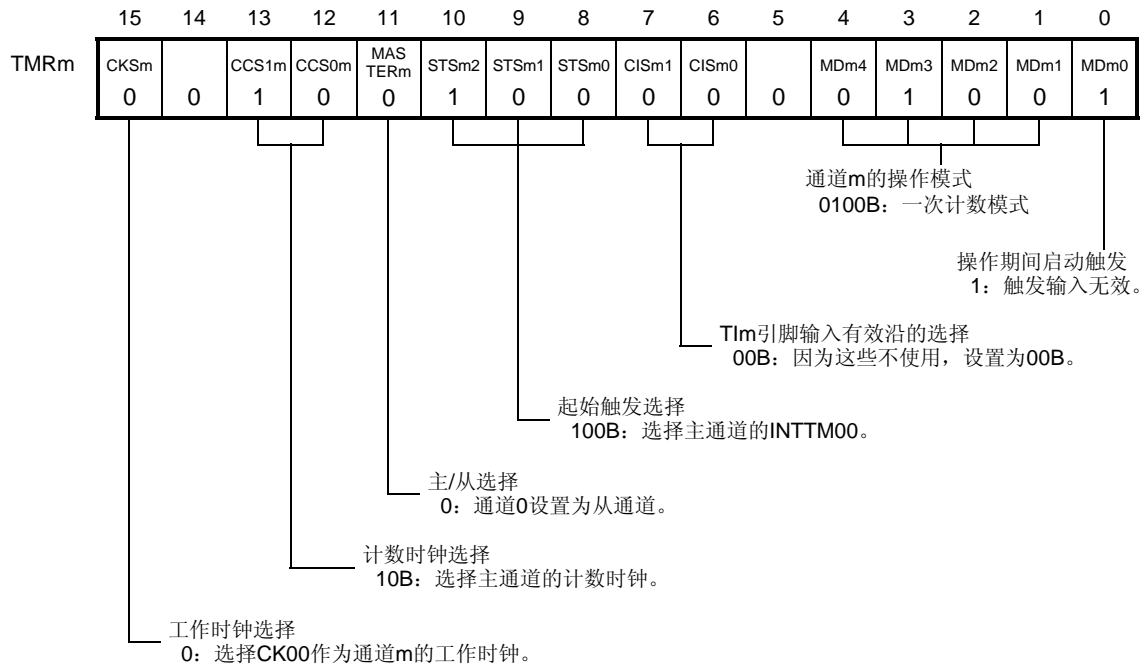


(b) 从通道 1（实时输出触发产生通道（TRC01 = 1））的其它寄存器

TOE0: TOE01	0: 通过计数操作停止TO01输出操作。
TO0: TO01	0: 从TO01输出一个低电平。
TOM0: TOM01	0: 当TOE01 = 0时设为0（通过计数操作停止TO01输出操作）。
TOT0: TOT01	0: 当TOM01 = 0（反转模式）时设为0。
TOL0: TOL01	0: 当TOM01 = 0（反转模式）时设为0。
TDE0: TDE01	0: 停止死区时间控制。
TRE0: TRE01	0: 停止实时输出。
TRO0: TRO01	0: 当TRE01 = 0（停止实时输出）时设为0。
TRC0: TRC01	1: 作为实时输出触发产生通道运行。
TME0: TME01	0: 停止调制输出。

图 7-82. 当使用非互补调制输出功能时寄存器设置内容的示例（类型1）（从通道2~7）

(a) 从通道2~7（实时输出通道（TRCm=0））的定时器模式寄存器m（TMRm）



(b) 从通道2~7（实时输出通道（TRCm=0））的其它寄存器

TOE0:TOEm	0: 通过计数操作, 停止TOm输出操作。 1: 通过计数操作, 允许TOm输出操作。
TO0:TOm	0: 从TOm输出低电平。 1: 从TOm输出高电平。
TOM0:TOMm	1: 设置组合操作模式。
TOT0:TOTm	0: 产生不同于三角波PWM输出。
TOL0:TOLm	0: 正向逻辑输出 (高有效)
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	1: 允许实时输出。
TRO0:TROm	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。 1: 允许调制输出。

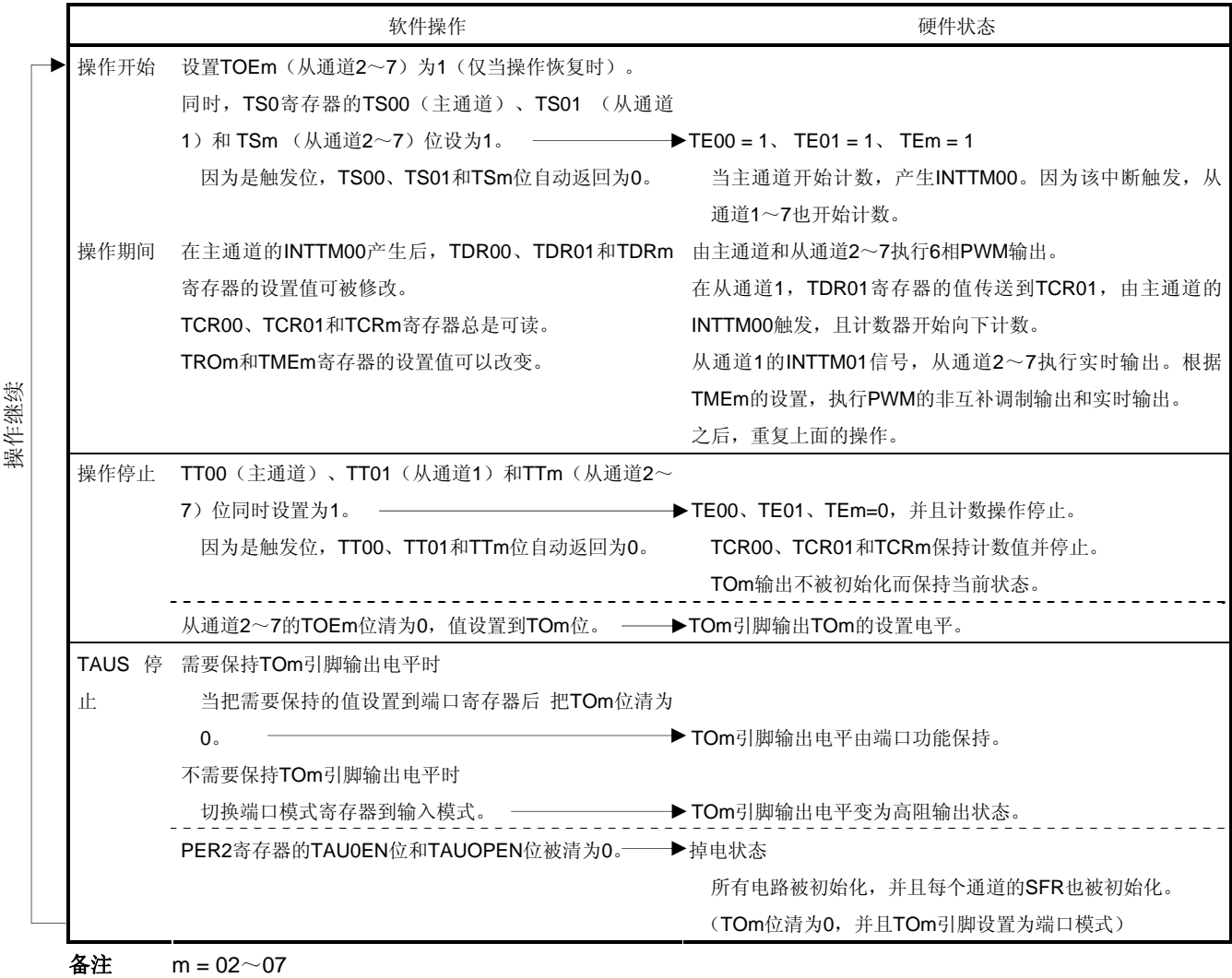
备注 m = 02~07

图 7-83. 当使用非互补调制输出功能时的操作过程 (1/2)

软件操作	硬件状态
TAUS 默认设置	掉电状态 (停止供给时钟, 并且所有寄存器不能写入。)
设置PER2寄存器的TAU0EN位和TAUOPEN位为1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
设置TPS0 寄存器。 决定CK00和CK01的时钟频率。	
通道默认设置	通道停止操作。 (供给时钟并且产生一些功耗。)
设置要使用的8个通道的TMR00, TMR01和TMRm寄存器 (决定通道的操作模式)。 间隔 (周期) 值设置到主通道的TDR00寄存器, 设置0000H 至从通道1的 TDR01 寄存器, 占空比设置到从通道m的TDRm寄存器。	
设置从通道1。 设置TRC01位为1 (触发产生通道)。 设置从通道2~7。 设置TOMm位为1(组合操作模式)。 设置TOTm 位为 0 (产生波形不同于三角波PWM输出)。 设置TOLm位并决定TOM输出的有效电平。 设置TDEm位为0 (停止死区时间控制)。 设置TREM位为1 (允许实时输出)。 设置实时输出电平至TROm位。 设置TRCm位为0 (非触发产生通道)。 设置TME m位并决定调制输出控制。	TOM引脚为高阻输出状态。
设置TOM位并决定TOM输出的默认电平。	当端口模式寄存器在输出模式下, 并且端口寄存器为0时, TOM输出默认设置电平。
设置TOEm位为1并使能TOM操作。	因为通道停止工作TOM不改变。
将端口寄存器和端口模式寄存器清为0。	TOM引脚输出TOM的设置电平。

备注 m = 02~07

图 7-83. 当使用非互补调制输出功能时的操作过程（2/2）



<R>

7.4.14 作为非互补调制输出功能的操作（类型 2）

为执行非互补调制输出，必须从霍尔传感器检测出一个触发器信号，且软件操作必须被执行。

非互补调制输出功能为一个周期设置了 6 个占空宽度，而且产生三角波 PWM 输出。准备 1 个周期生成通道和 6 个占空比生成通道（三角波 PWM 输出功能）。关于三角波 PWM 功能的详细信息，请参见 **7.4.4 作为三角波 PWM 输出功能操作**。

而且，根据对应于产生 PWM 的 T0m 引脚的 TROM 的值产生实时输出。准备了一个实时输出触发产生通道。关于实时输出触发的详细信息，请参见 **7.4.11 作为连接实时输出功能的操作（类型 2）**

非互补调制输出由定时器输出（PWM）产生，由 TROM 和 TME_m 操作。

主通道操作在间隔定时器模式并计数周期。

从通道 1 运行于事件计数模式，并产生一个实时输出触发。主通道的 INTTM00 输出个数被缩短，且产生一个实时输出触发。

从通道 2~7 运行于向上向下计数模式，计数占空比，并产生一个三角波 PWM 波形。

而且，模組输出可通过操作 TME_m 来控制。当 TME_m 为“1”时，可由 T0m 引脚输出的 PWM 的逻辑与和实时输出来执行非互补调制输出。当 TME_m 为“0”时，将从 T0m 引脚中输出实时输出设定值（TROM）。

要调制 PWM 和实时输出并将二者从 T0m 引脚输出，操作必须从低电平的从通道 2~7 的 T0m 的默认电平开始。假设从通道 2 至 7 的 TOL_m 固定为“0”，且若需要反相控制时使用 TLS2 至 TLS7 (OPMR 寄存器)。

PWM 的调制控制和实时输出是在 TROM 为“1”的基础上进行的（有效电平：高电平）。PWM 输出的调制输出不能在 TROM 设为“0”时进行。

注意事项 当 TE00 = 1、TE01 = 1 或 TEm = 1 时，TS00、TS01 或 TS_m 不能被设为“1”（强制重启）。当 TE00 = 1、TE01 = 1 或 TEm = 1 时，若 TS00、TS01 或 TS_m 被设为“1”，则计数器的值（TCR00、TCR01 或 TCR_m）将非法且 T0m 将不能输出预期波形。

备注 m = 02~07

图 7-84 作为非互补调制输出功能操作的框图（类型 2）

主通道
(间隔定时器模式)

从通道1
TRC01 = 1
(事件计数器模式)

从通道2
TRC02 = 0
(向上/向下计数模式)

从通道 7
TRC07 = 0
(向上/向下计数模式)

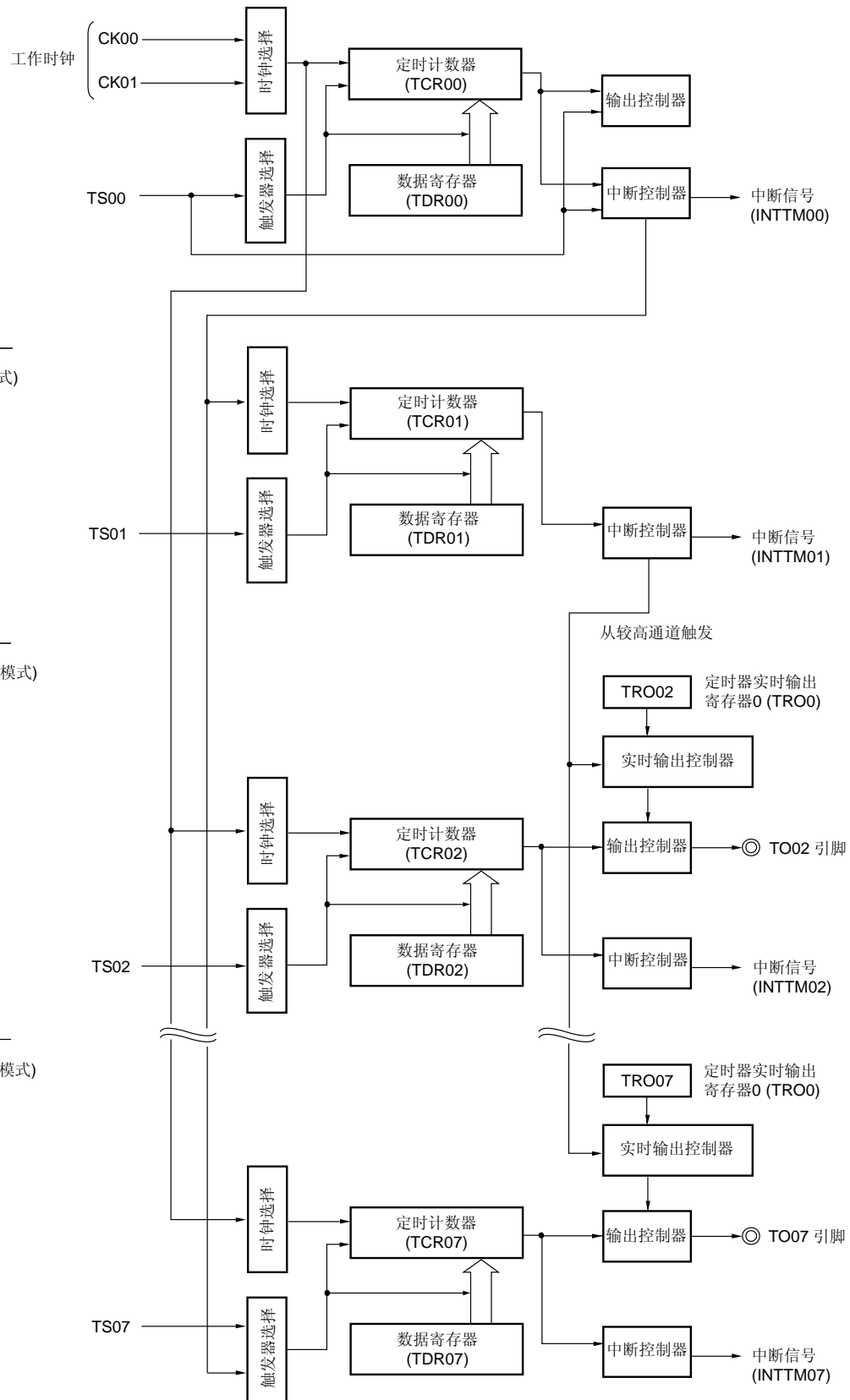
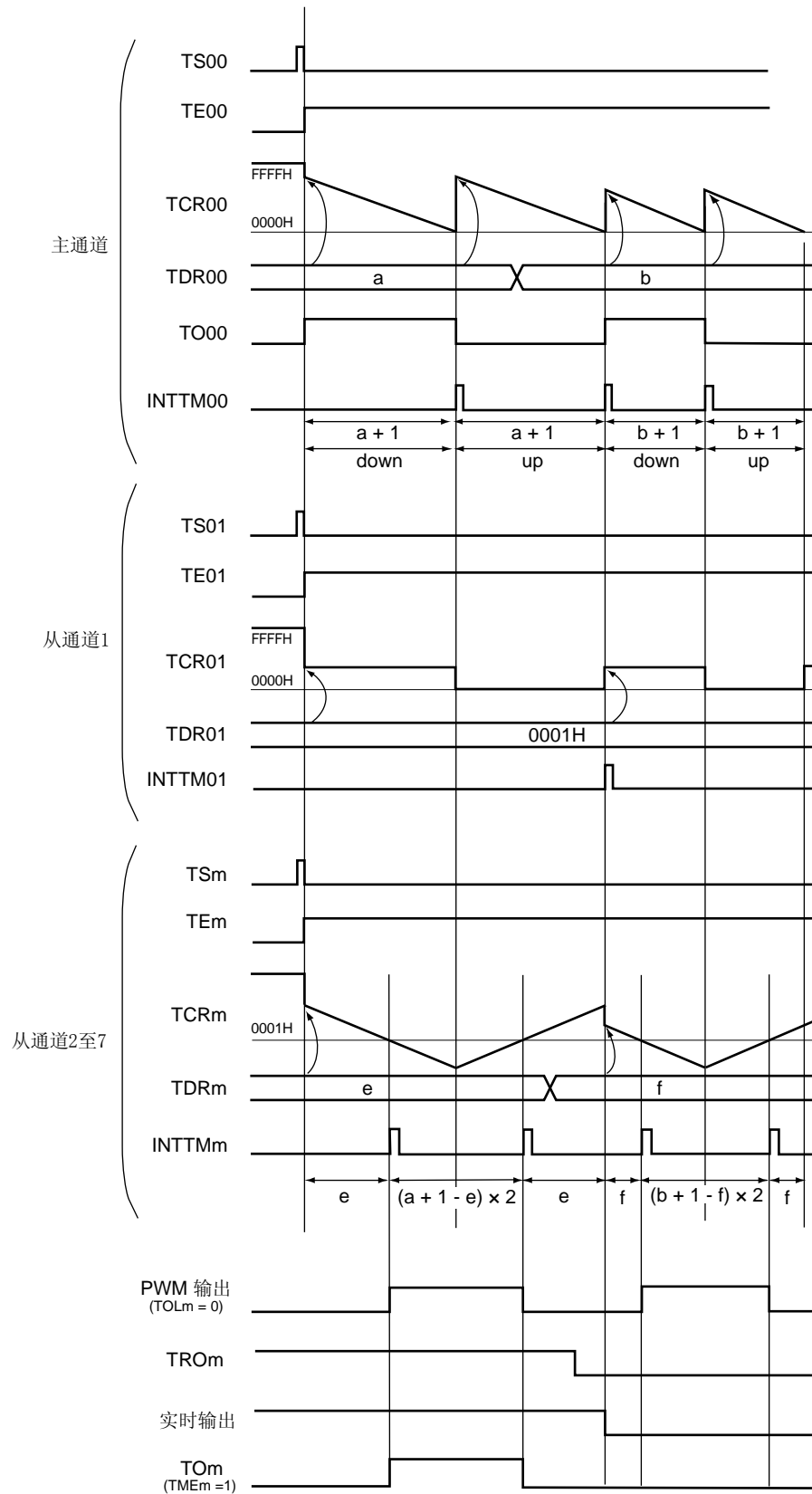


图 7-85 作为非互补调制输出功能操作的基本时序示例（类型 2）



备注 m = 02~07

120° 磁控制期间，非互补调制输出显示如下。

为了在全调制、前半调制和后半调制之间切换，TROm 和 TME m 必须在运行期间被操作。

表 7-5. 120° 磁控制期间，非互补输出列表（全调制）

控制寄存器	设置 TROm、TME m 的值						TOm 输出					
	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7
TROm	0	1	1	0	0	0	低	PWM	PWM	低	低	低
TME m	1	1	1	1	1	1						
TROm	0	1	0	0	1	0	低	PWM	低	低	PWM	低
TME m	1	1	1	1	1	1						
TROm	0	0	0	1	1	0	低	低	低	PWM	PWM	低
TME m	1	1	1	1	1	1						
TROm	1	0	0	1	0	0	PWM	低	低	PWM	低	低
TME m	1	1	1	1	1	1						
TROm	1	0	0	0	0	1	PWM	低	低	低	低	PWM
TME m	1	1	1	1	1	1						
TROm	0	0	1	0	0	1	低	低	PWM	低	低	PWM
TME m	1	1	1	1	1	1						

表 7-6. 120° 磁控制期间，非互补输出列表（前半调制）

控制寄存器	设置 TROm、TME m 的值						TOm 输出					
	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7
TROm	0	1	1	0	0	0	低	PWM	高	低	低	低
TME m	0	1	0	0	0	0						
TROm	0	1	0	0	1	0	低	高	低	低	PWM	低
TME m	0	0	0	0	1	0						
TROm	0	0	0	1	1	0	低	低	低	PWM	高	低
TME m	0	0	0	1	0	0						
TROm	1	0	0	1	0	0	PWM	低	低	高	低	低
TME m	1	0	0	0	0	0						
TROm	1	0	0	0	0	1	高	低	低	低	低	PWM
TME m	0	0	0	0	0	1						
TROm	0	0	1	0	0	1	低	低	PWM	低	低	高
TME m	0	0	1	0	0	0						

备注 m = 02~07

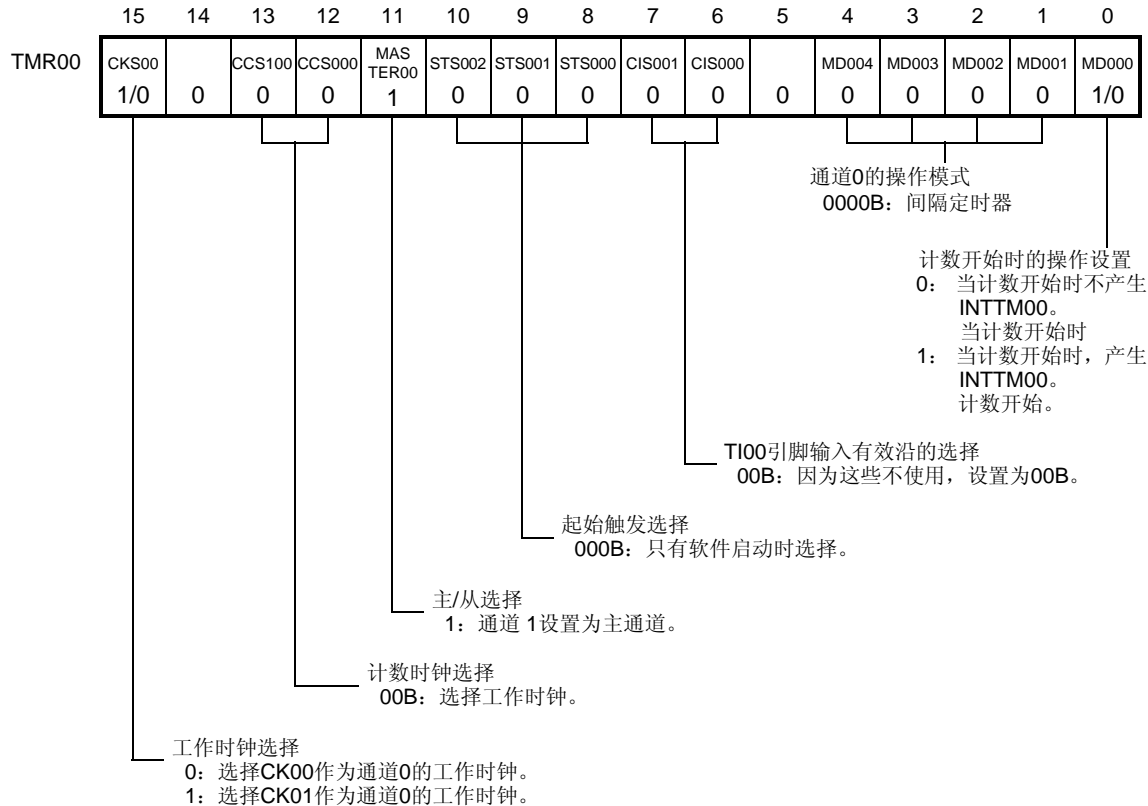
表 7-7. 120° 磁控制期间，非互补输出列表（后半调制）

控制寄存器	设置 TROm、TMEm 的值						TOM 输出					
	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7
TROm	0	1	1	0	0	0	低	高	PWM	低	低	低
TMEm	0	0	1	0	0	0						
TROm	0	1	0	0	1	0	低	PWM	低	低	高	低
TMEm	0	1	0	0	0	0						
TROm	0	0	0	1	1	0	低	低	低	高	PWM	低
TMEm	0	0	0	0	1	0						
TROm	1	0	0	1	0	0	高	低	低	PWM	低	低
TMEm	0	0	0	1	0	0						
TROm	1	0	0	0	0	1	PWM	低	低	低	低	高
TMEm	1	0	0	0	0	0						
TROm	0	0	1	0	0	1	低	低	高	低	低	PWM
TMEm	0	0	0	0	0	1						

备注 m = 02~07

图 7-86. 当使用非互补调制输出功能时寄存器设置内容的示例（类型 2）（主通道）

(a) 定时器模式寄存器 00(TMR00)



(b) 其它寄存器

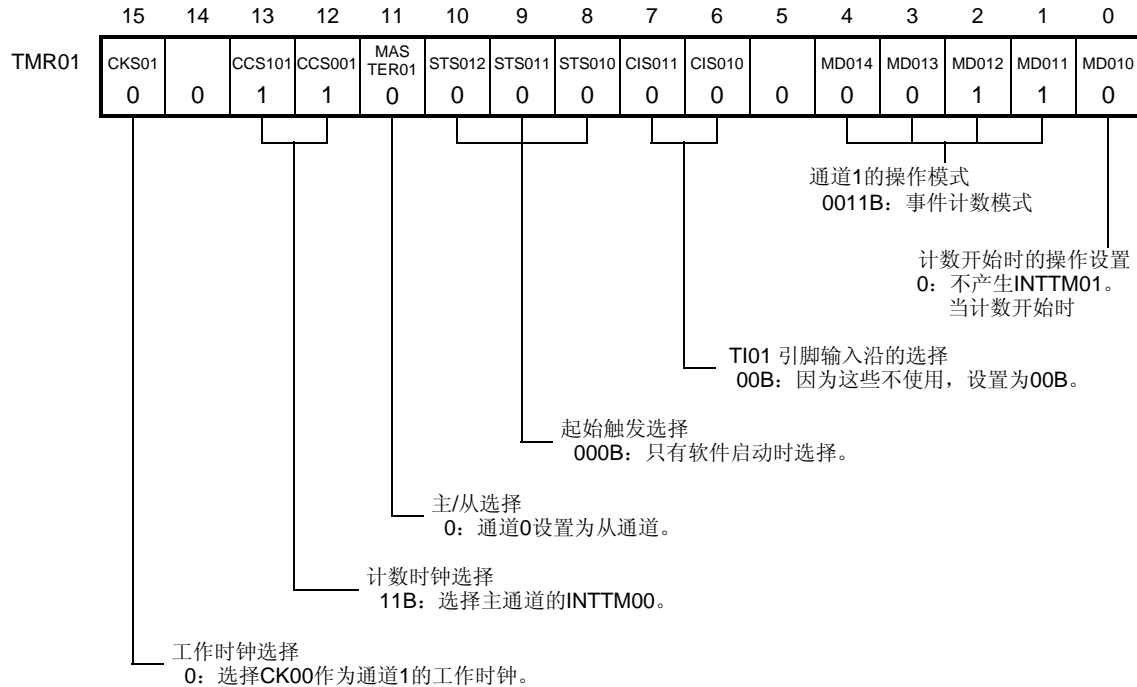
TOE0:TOE00*	0: 通过计数操作停止 TO00 输出操作。 1: 通过计数操作允许TO00输出操作。
TO0:TO00	0: 从TO00输出低电平。 1: 从TO00输出高电平。
TOM0:TOM00	0: 当TOE00 = 0时设为0（通过计数操作停止TO00输出）。
TOT0:TOT00	0: 当TOM00 = 0(反转模式)时设为0。
TOL0:TOL00	0: 当TOM00 = 0(反转模式)时设为0。
TDE0:TDE00	0: 停止死区时间控制。
TRE0:TRE00	0: 停止实时输出。
TRO0:TRO00	0: 当TRE00=0（停止实时输出）时设为0。
TRC0:TRC00	0: 作为实时输出触发产生通道不运行。
TME0:TME00	0: 停止调制输出。

注 在下列情况中, 设置主通道的 TOE00 至“1”。

- 当使用一个 INTTMM0、INTTMV0、INTTMM1 或 INTTMV1 中断信号
- 利用 OPMR、OPHS 或 OPHT 寄存器, 控制一个高阻输出或通过控制器选择一个 A/D 转换触发时

图 7-87. 当使用非互补调制输出功能时寄存器设置内容的示例（类型 2）（从通道 1）

(a) 从通道 1（实时输出触发产生通道（TRC01 = 1））的定时器模式寄存器 01（TMR01）

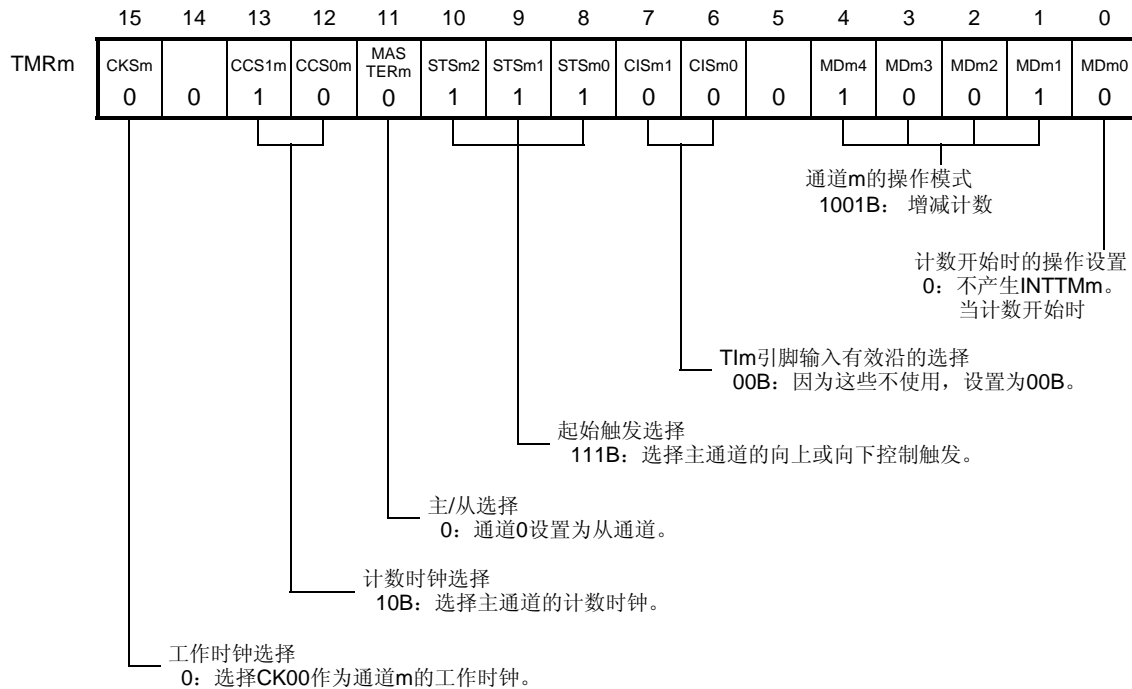


(b) 从通道 1（实时输出触发产生通道（TRC01 = 1））的其它寄存器

TOE0:TOE01	0: 通过计数操作停止TO01输出操作。
TO0:TO01	0: 从TO01输出一个低电平。
TOM0:TOM01	0: 当TOE01 = 0时设为0（通过计数操作停止TO01输出操作）。
TOT0:TOT01	0: 当TOM01 = 0（反转模式）时设为0。
TOL0:TOL01	0: 当TOM01 = 0（反转模式）时设为0。
TDE0:TDE01	0: 停止死区时间控制。
TRE0:TRE01	0: 停止实时输出。
TRO0:TRO01	0: 当TRE01 = 0（停止实时输出）时设为0。
TRC0:TRC01	1: 作为实时输出触发产生通道运行。
TME0:TME01	0: 停止调制输出。

图 7-88. 当使用非互补调制输出功能时寄存器设置内容的示例（类型2）（从通道2~7）

(a) 从通道2~7（实时输出通道（TRCm=0））的定时器模式寄存器m（TMRm）



(b) 从通道2~7（实时输出通道（TRCm=0））的其它寄存器

TOE0:TOEm	0: 通过计数操作, 停止TOm输出操作。 1: 通过计数操作, 允许TOm输出操作。
TO0:TOm	0: 从TOm输出低电平。 1: 从TOm输出高电平。
TOM0:TOMm	1: 设置组合操作模式。
TOT0:TOTm	1: 设置三角波PWM输出。
TOL0:TOLm	0: 正向逻辑输出 (高有效)
TDE0:TDEm	0: 停止死区时间控制。
TRE0:TREm	1: 允许实时输出。
TRO0:TROm	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCm	0: 作为实时输出触发产生通道不运行。
TME0:TMEm	0: 停止调制输出。 1: 允许调制输出。

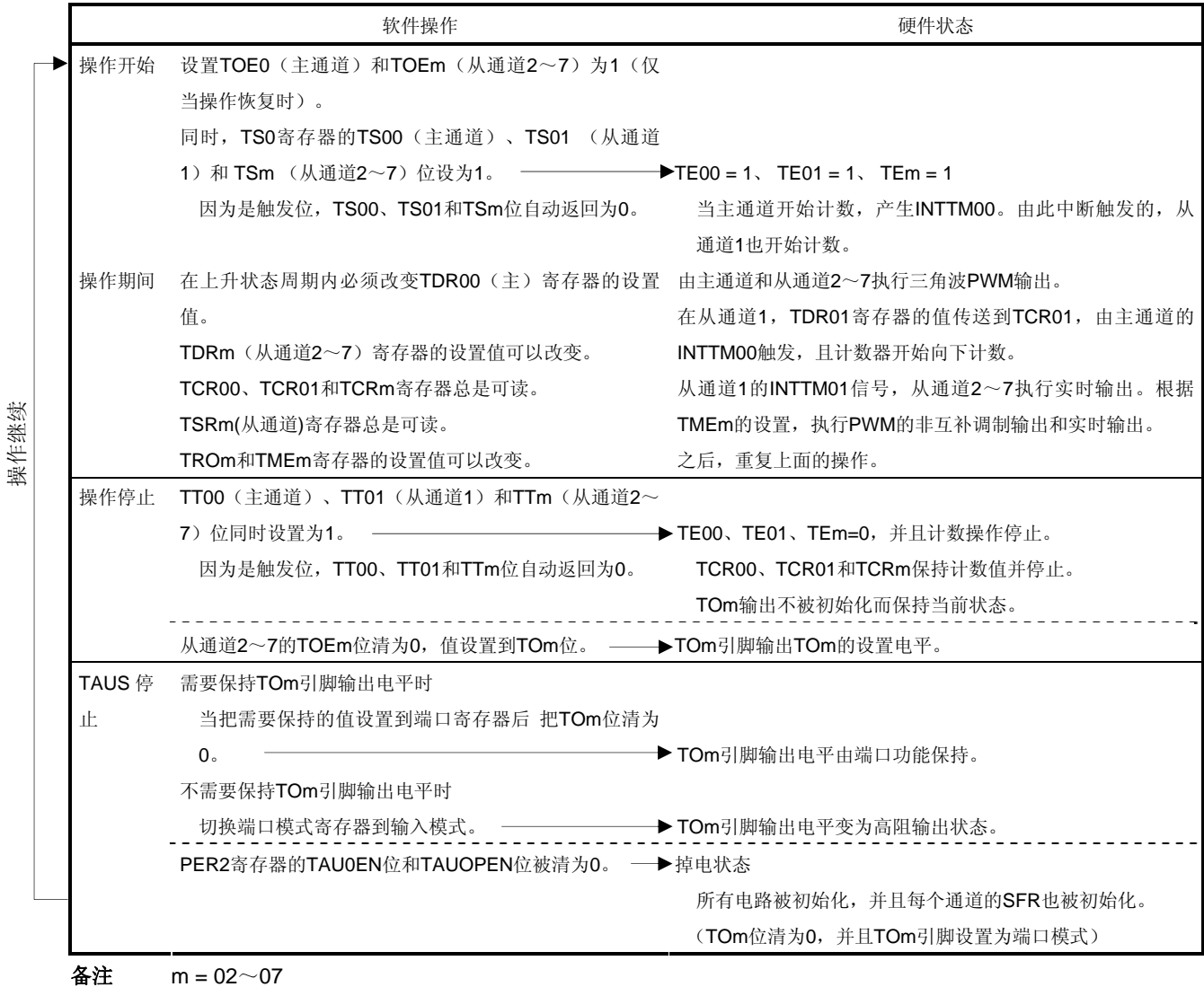
备注 m = 02~07

图 7-89. 当使用非互补调制输出功能时的操作过程（类型 2）（1/2）

软件操作	硬件状态
<p>TAUS 默认设置</p> <p>设置PER2寄存器的TAU0EN位和TAUOPEN位为1。——▶</p> <p>设置TPS0 寄存器。</p> <p>决定CK00和CK01的时钟频率。</p>	<p>掉电状态</p> <p>（停止供给时钟，并且所有寄存器不能写入。）</p> <p>上电状态。每个通道停止操作。</p> <p>（开始供给时钟并且每个寄存器允许写入。）</p>
<p>通道默认设置</p> <p>设置要使用的8个通道的TMR00，TMR01和TMRm寄存器（决定通道的操作模式）。间隔（周期）值设置到主通道的TDR00寄存器，缩短的中断个数设置到从通道1的TDR01寄存器，占空比设置到从通道2～7的TDRm寄存器。</p> <p>设置从通道1。</p> <p>设置TRC01位为1（触发产生通道）。</p> <p>设置从通道2～7。</p> <p>设置TOMm位为1(组合操作模式)。</p> <p>设置TOTm位为1（三角波PWM输出）。</p> <p>设置TOLm位为0(正向逻辑输出)。</p> <p>设置TDEm位为0（停止死区时间控制）。</p> <p>设置TREM位为1（允许实时输出）。</p> <p>设置实时输出电平至TROm位。</p> <p>设置TRCm位为0（非触发产生通道）。</p> <p>设置TMEm位并决定调制输出控制。</p> <p>设置TOm位并决定TOm输出的默认电平。——▶</p> <p>设置TOEm位为1并使能TOm操作。——▶</p> <p>将端口寄存器和端口模式寄存器清为0。——▶</p>	<p>通道停止操作。</p> <p>（供给时钟并且产生一些功耗。）</p> <p>TOm引脚为高阻输出状态。</p> <p>当端口模式寄存器在输出模式下，并且端口寄存器为0时，TOm输出默认设置电平。</p> <p>因为通道停止工作TOm不改变。</p> <p>TOm引脚输出TOm的设置电平。</p>

备注 m = 02～07

图 7-89. 当使用非互补调制输出功能时的操作过程（类型 2）（2/2）



<R>

7.4.15 作为非互补调制输出功能的操作

为执行非互补调制输出，必须从霍尔传感器检测出一个触发器信号，且软件操作必须被执行。

互补调制输出功能在一个周期内设置 3 个占空比宽度，通过添加死区时间至 6 个 TOP 和 TOQ 可产生 6 相三角波 PWM 输出。准备 1 个周期产生通道，3 个占空比产生通道和 3 个死区时间产生通道（6 相三角波 PWM 输出功能）。关于 6 相三角波 PWM 功能的详细信息，请参见 **7.4.6 作为 6 相三角波 PWM 输出功能操作**。

而且，根据对应于产生 PWM 的 TOP 和 TOQ 引脚的 TROP 和 TROa 的值产生实时输出。准备了一个实时输出触发产生通道。关于实时输出触发的详细信息，请参见 **7.4.11 作为连接实时输出功能的操作（类型 2）**。

非互补调制输出由定时器输出（PWM）产生，由 TROP、TROQ、TMEp、TMEq、TOLp 和 TOLq 操作。

主通道操作在间隔定时器模式并计数周期。

从通道 1 的 TCRn 运行于事件计数模式，并产生一个实时输出触发。主通道的 INTTM00 输出个数被缩短，且产生一个实时输出触发。

从通道 2、4、6 运行在上下计数模式下并计数占空比。

从通道 3、5、7 运行在上下计数模式下并计数占空比。

通过从通道 2、4、6（占空）和从通道 3、5、7（死区时间）的计数操作（INTTMp、INTTMq）来改变 TOP 和 TOQ，从而输出一个带死区时间的 6 相三角波 PWM 波形。

而且，模組输出可通过操作 TMEp 和 TMEq 来控制。当 TMEp 和 TMEq 为“1”时，调制 PWM 和实时输出并从 TOP 和 TOQ 引脚输出。当 TMEp 和 TMEq 为“0”时，实时输出设定值（TROP 和 TROq）将由 TOP 和 TOQ 引脚输出。

要调制 PWM 和实时输出并将二者从 TOP 和 TOQ 引脚输出，根据从通道 2~7 的 TOLp 和 TOLq 的设置值来决定 TOP 和 TOQ 引脚的默认电平（当 TOLp 和 TOLq 为“0”时，默认电平为低电平。当 TOLp 和 TOLq 为“1”时，默认电平为高电平。）

随着非互补调制输出，通过从通道 2 和 3、4 和 5、6 和 7 成对使用，输出一个 PWM 调制波形。操作如下。

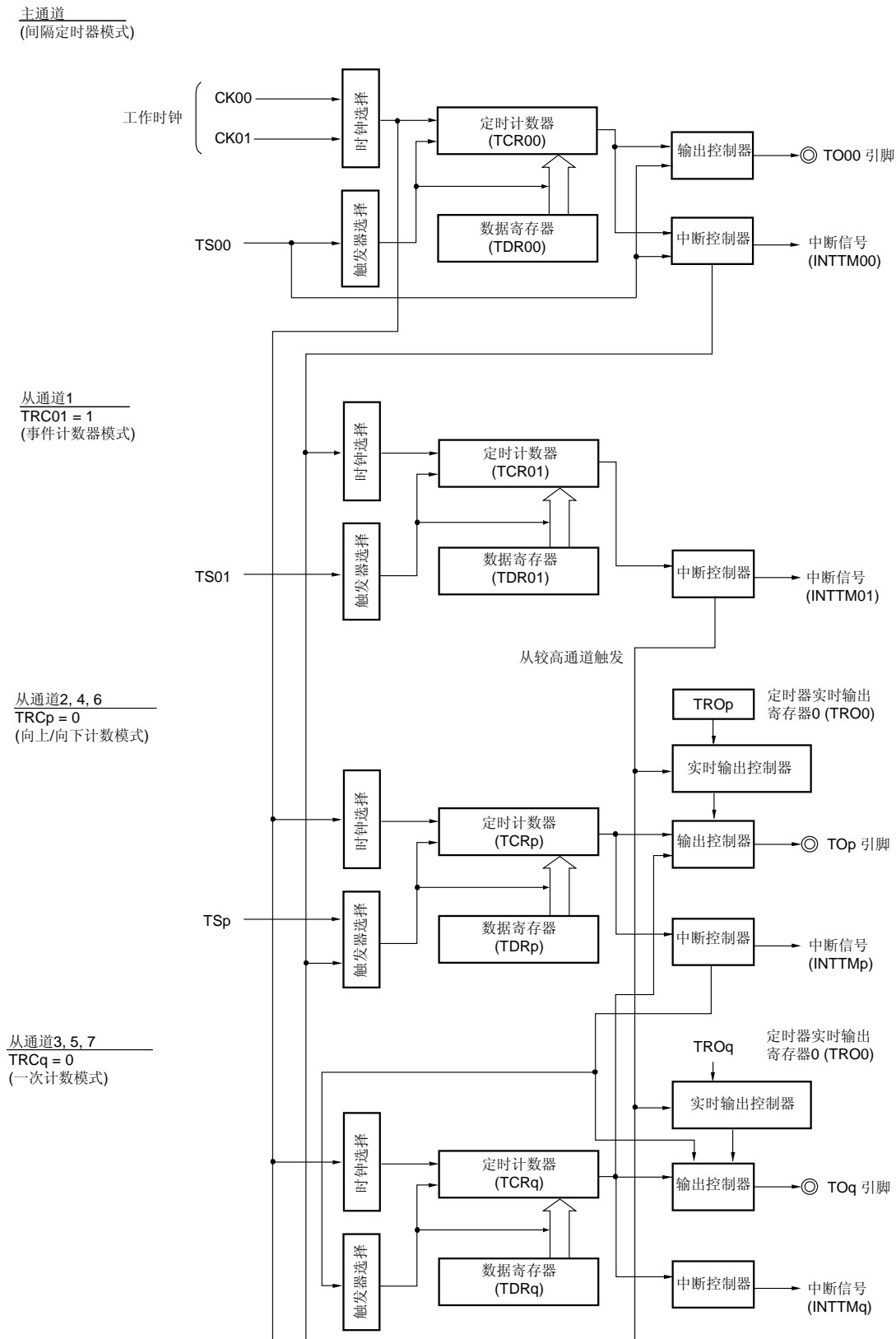
- 当 TROP 和 TROq 分别设置为 1 和 0 时，正相位 PWM 和反相位 PWM 将分别从 TOP 和 TOQ 引脚输出。
- 当 TROP 和 TROq 分别设置为 0 和 1 时，反相位 PWM 和正相位 PWM 将分别从 TOP 和 TOQ 引脚输出。

一个正相 PWM 从一个 TROP 设置为“1”的通道 TOP 引脚输出，一个反相 PWM 从一个与 TROP 为“1”的通道成对通道的 TOQ 引脚输出。

注意事项 当 TE00 = 1、TEp = 1 或 TEq = 1 时，TS00、TSp 或 TSq 不能设为“1”（强制重启）。当 TE00 = 1、TEp = 1 或 TEq = 1 时，若 TS00、TSp 或 TSq 设为“1”，则计数器的值（TCR00、TCRp 或 TCRq）将非法且 TOP 或 TOQ 将不能输出预期波形。

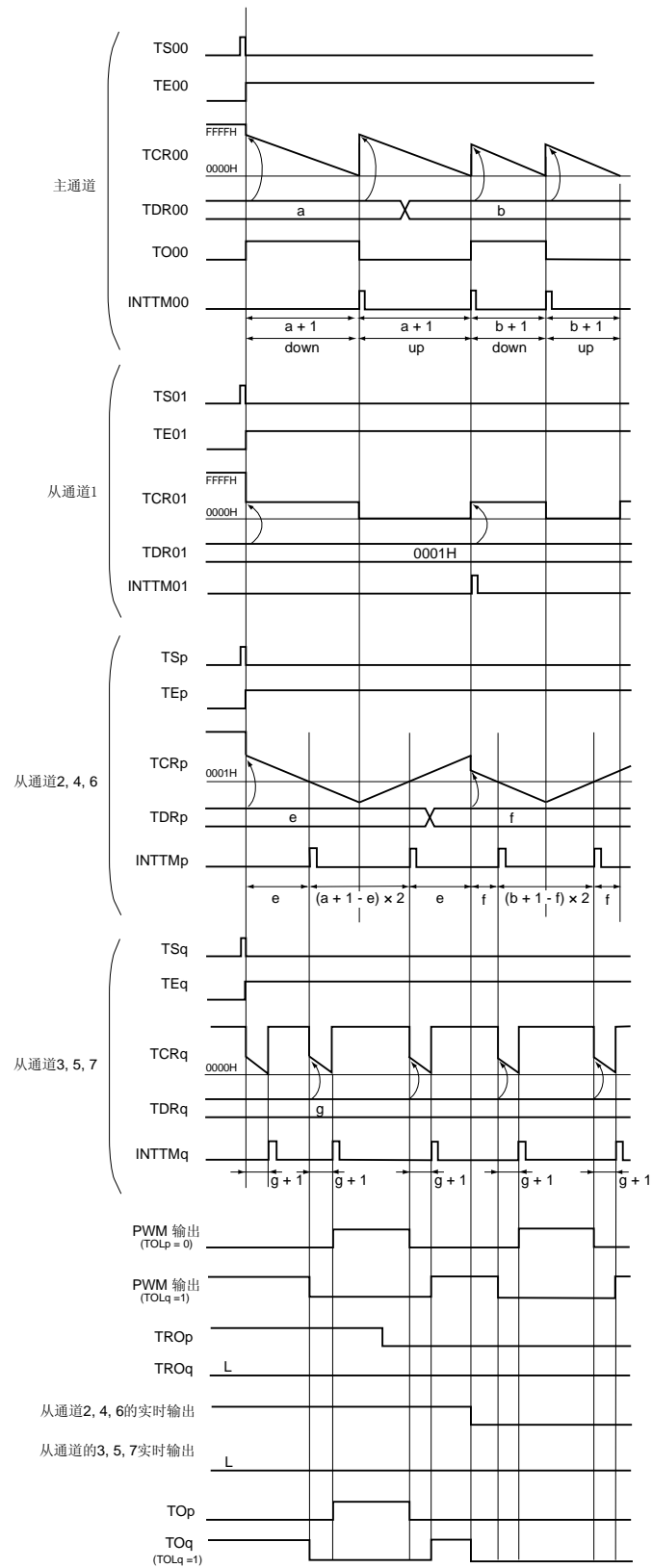
备注 p = 02、04、06
q = 03、05、07

图 7-90. 非互补调制输出功能操作的框图



备注 p = 02、04、06
 q = 03、05、07

图 7-91. 作为非互补调制输出功能操作的基本时序的示例



备注 $p = 02、04、06$
 $q = 03、05、07$

120° 磁控制期间，互补输出显示如下。

为了在正相 PWM 和反相 PWM 之间切换，TOLp 和 TOLq 必须在运行期间被操作。

为了在全调制、前半调制和后半调制之间切换，TROp、TROq、TMEp 和 TMEq 必须在运行期间被操作。

表 7-8. 120° 磁控制期间互补输出列表（全调制）

控制寄存器	设置 TROp、TROq、TMEp、TMEq、TOLp、TOLq 的值						TOp、TOq 输出					
	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7
TROp、q	0	1	1	0	0	0	反相 PWM	正相 PWM	正相 PWM	反相 PWM	低	低
TMEp、q	1	1	1	1	1	1						
TOLp、q	1	0	0	1	1	1						
TROp、q	0	1	0	0	1	0	反相 PWM	正相 PWM	低	低	正相 PWM	反相 PWM
TMEp、q	1	1	1	1	1	1						
TOLp、q	1	0	1	1	0	1						
TROp、q	0	0	0	1	1	0	低	低	反相 PWM	正相 PWM	正相 PWM	反相 PWM
TMEp、q	1	1	1	1	1	1						
TOLp、q	1	1	1	0	0	1						
TROp、q	1	0	0	1	0	0	正相 PWM	反相 PWM	反相 PWM	正相 PWM	低	低
TMEp、q	1	1	1	1	1	1						
TOLp、q	0	1	1	0	1	1						
TROp、q	1	0	0	0	0	1	正相 PWM	反相 PWM	低	低	反相 PWM	正相 PWM
TMEp、q	1	1	1	1	1	1						
TOLp、q	0	1	1	1	1	0						
TROp、q	0	0	1	0	0	1	低	低	正相 PWM	反相 PWM	反相 PWM	正相 PWM
TMEp、q	1	1	1	1	1	1						
TOLp、q	1	1	0	1	1	0						

表 7-9. 120° 磁控制期间互补输出列表（前半调制）

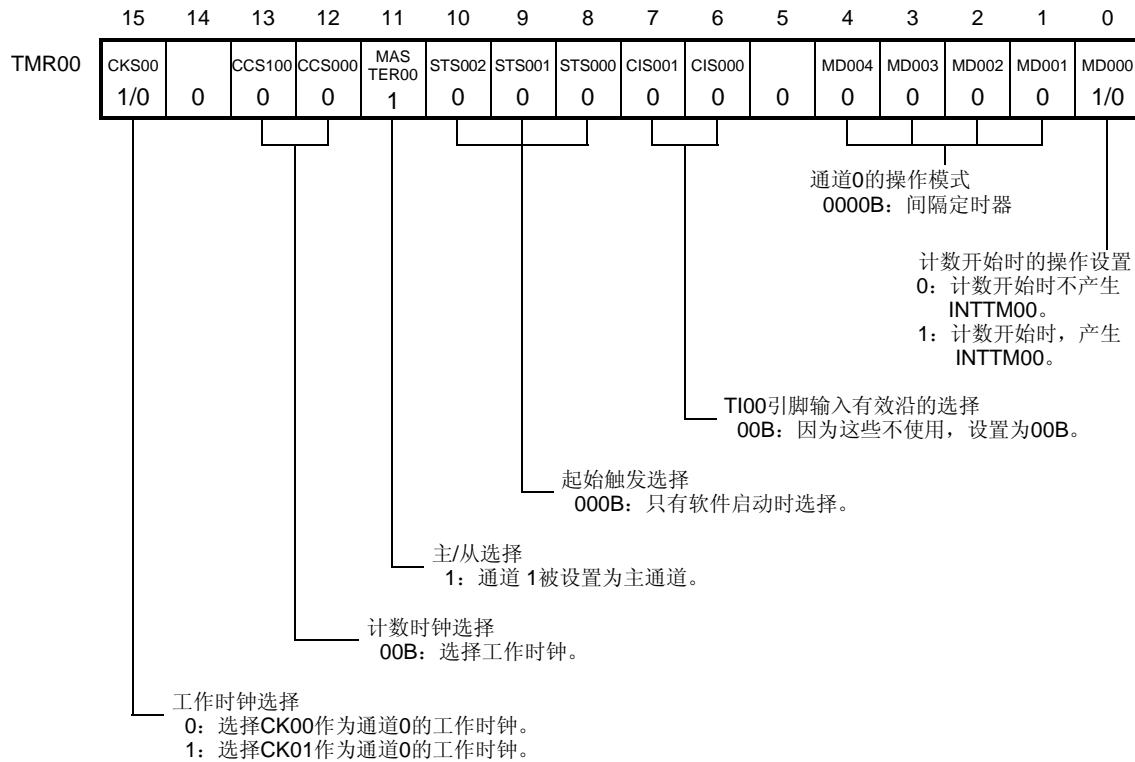
控制寄存器	设置 TROp、TROq、TMEp、TMEq、TOLp、TOLq 的值						TOP、TOq 输出					
	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7
TROp、q	0	1	1	0	0	0	反相 PWM	正相 PWM	高	低	低	低
TMEp、q	1	1	0	0	0	0						
TOLp、q	1	0	0	1	1	1						
TROp、q	0	1	0	0	1	0	低	高	低	低	正相 PWM	反相 PWM
TMEp、q	0	0	0	0	1	1						
TOLp、q	1	1	1	1	0	1						
TROp、q	0	0	0	1	1	0	低	低	反相 PWM	正相 PWM	高	低
TMEp、q	0	0	1	1	0	0						
TOLp、q	1	1	1	0	0	1						
TROp、q	1	0	0	1	0	0	正相 PWM	反相 PWM	低	高	低	低
TMEp、q	1	1	0	0	0	0						
TOLp、q	0	1	1	0	1	1						
TROp、q	1	0	0	0	0	1	高	低	低	低	反相 PWM	正相 PWM
TMEp、q	0	0	0	0	1	1						
TOLp、q	0	1	1	1	1	0						
TROp、q	0	0	1	0	0	1	低	低	正相 PWM	反相 PWM	低	高
TMEp、q	0	0	1	1	0	0						
TOLp、q	1	1	0	1	1	0						

表 7-10. 120° 磁控制期间互补输出列表（后半调制）。

控制寄存器	设置 TROp、TROq、TMEp、TMEq、TOLp、TOLq 的值						TOP、TOq 输出					
	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7	从通道 2	从通道 3	从通道 4	从通道 5	从通道 6	从通道 7
TROp、q	0	1	1	0	0	0	低	高	正相 PWM	反相 PWM	低	低
TMEp、q	0	0	1	1	0	0						
TOLp、q	1	0	0	1	1	1						
TROp、q	0	1	0	0	1	0	反相 PWM	正相 PWM	低	低	高	低
TMEp、q	1	1	0	0	0	0						
TOLp、q	1	0	1	1	0	1						
TROp、q	0	0	0	1	1	0	低	低	低	高	正相 PWM	反相 PWM
TMEp、q	0	0	0	0	1	1						
TOLp、q	1	1	1	0	0	1						
TROp、q	1	0	0	1	0	0	高	低	反相 PWM	正相 PWM	低	低
TMEp、q	0	0	1	1	0	0						
TOLp、q	0	1	1	0	1	1						
TROp、q	1	0	0	0	0	1	正相 PWM	反相 PWM	低	低	低	高
TMEp、q	1	1	0	0	0	0						
TOLp、q	0	1	1	1	1	0						
TROp、q	0	0	1	0	0	1	低	低	高	低	反相 PWM	正相 PWM
TMEp、q	0	0	0	0	1	1						
TOLp、q	1	1	0	1	1	0						

图 7-92. 当使用互补调制输出功能(主通道)时寄存器设置内容的示例

(a) 定时器模式寄存器 00(TMR00)

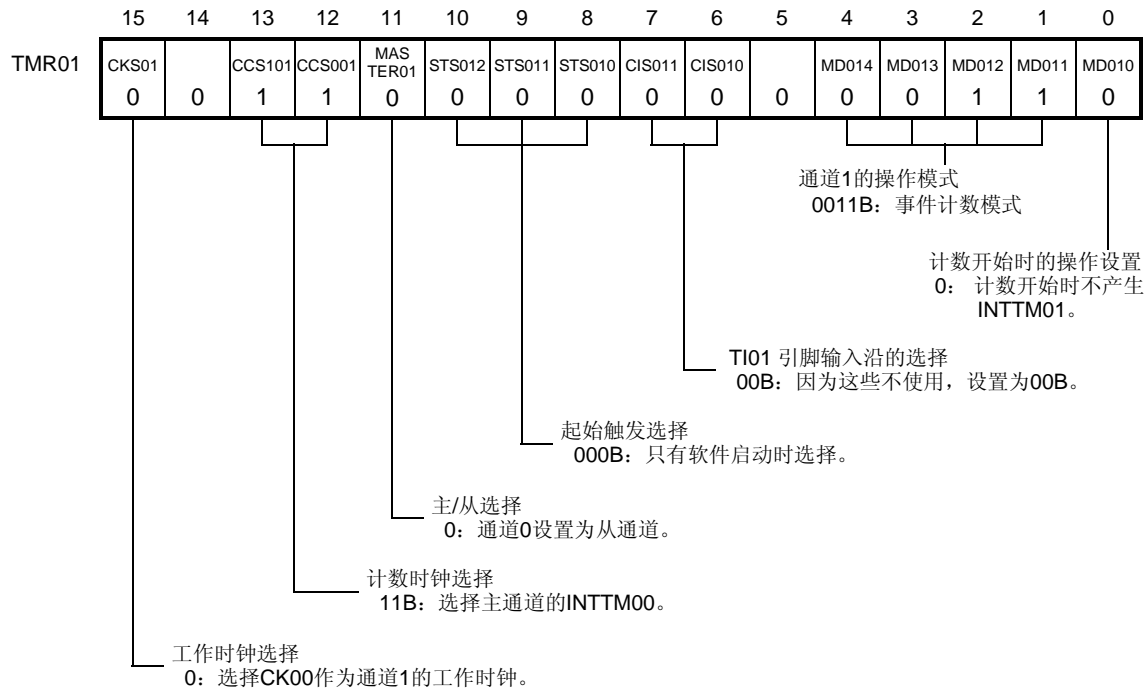


(b) 其它寄存器

TOE0:TOE00	0: 通过计数操作停止 TO00 输出操作。 1: 通过计数操作允许TO00输出操作。
TO0:TO00	0: 从TO00输出低电平。 1: 从TO00输出高电平。
TOM0:TOM00	0: 当TOE00 = 0时设为0 (通过计数操作停止TO00输出)。
TOT0:TOT00	0: 当TOM00 = 0(反转模式)时设为0。
TOL0:TOL00	0: 当TOM00 = 0(反转模式)时设为0。
TDE0:TDE00	0: 停止死区时间控制。
TRE0:TRE00	0: 停止实时输出。
TR00:TR000	0: 当TRE00=0 (停止实时输出) 时设为0。
TRC0:TRC00	0: 作为实时输出触发产生通道不运行。
TME0:TME00	0: 停止调制输出。

图 7-93. 当使用互补调制输出功能（从通道 1）时寄存器设置内容的示例

(a) 从通道 1（实时输出触发产生通道（TRC01 = 1））的定时器模式寄存器 01（TMR01）

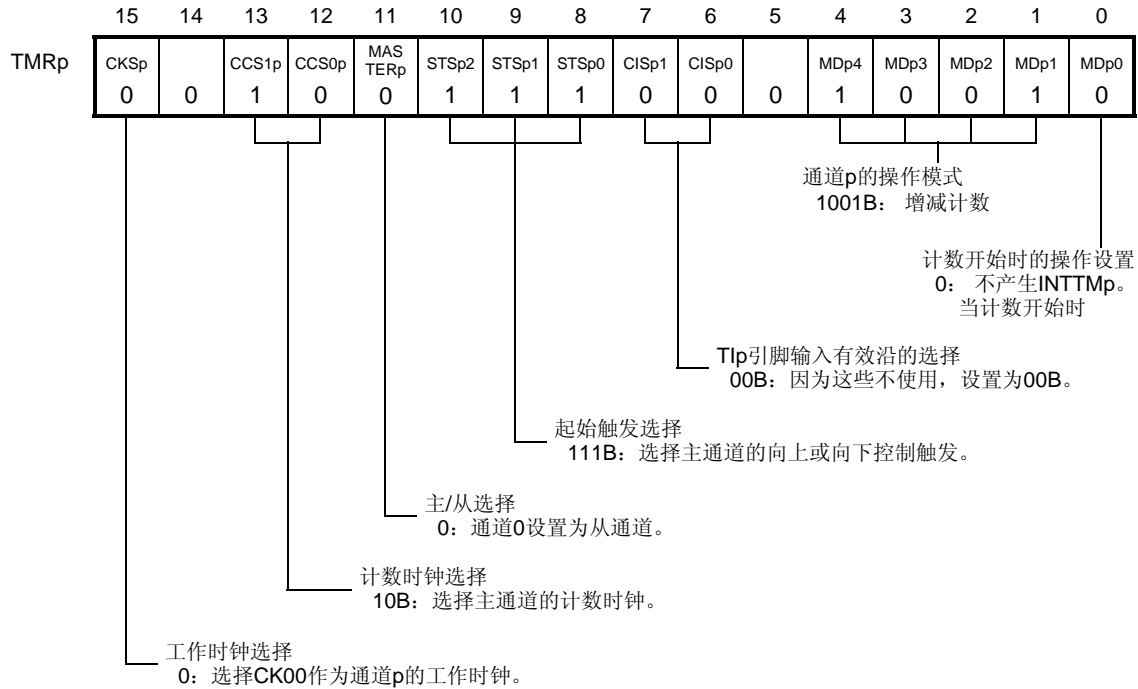


(b) 从通道 1（实时输出触发产生通道（TRC01 = 1））的其它寄存器

TOE0: TOE01	0: 通过计数操作停止TO01输出操作。
TO0: TO01	0: 从TO01输出一个低电平。
TOM0: TOM01	0: 当TOE01 = 0时设为0（通过计数操作停止TO01输出操作）。
TOT0: TOT01	0: 当TOM01 = 0（反转模式）时设为0。
TOL0: TOL01	0: 当TOM01 = 0（反转模式）时设为0。
TDE0: TDE01	0: 停止死区时间控制。
TRE0: TRE01	0: 停止实时输出。
TRO0: TRO01	0: 当TRE01 = 0（停止实时输出）时设为0。
TRC0: TRC01	1: 作为实时输出触发产生通道运行。
TME0: TME01	0: 停止调制输出。

图 7-94. 当使用互补调制输出功能（从通道 2、4、6）时寄存器设置内容的示例

(a) 从通道 2、4、6（实时输出通道（TRCp = 0））的定时器模式寄存器 p（TMRp）



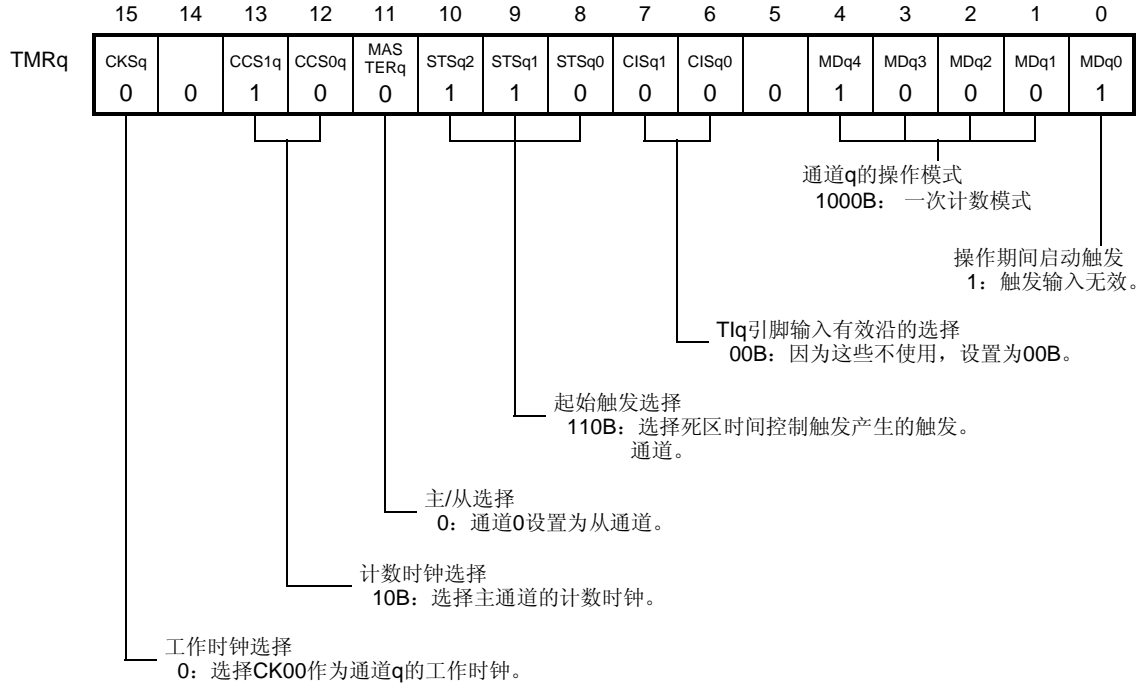
(b) 从通道 2、4、6（实时输出通道（TRCp = 0））的其它寄存器

TOE0:TOEp	0: 通过计数操作, 停止 TOp 输出操作。 1: 通过计数操作, 允许 TOp 输出操作。
TO0:TOp	0: 从 TOp 输出低电平。 1: 从 TOp 输出高电平。
TOM0:TOMp	1: 设置组合操作模式。
TOT0:TOTp	1: 设置三角波PWM输出。
TOL0:TOLp	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)
TDE0:TDEp	1: 允许死区时间控制。
TRE0:TREp	1: 允许实时输出。
TRO0:TROp	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCp	0: 作为实时输出触发产生通道不运行。
TME0:TMEp	0: 停止调制输出。 1: 允许调制输出。

备注 p = 02、04、06

图 7-95. 当使用互补调制输出功能（从通道 3、5、7）时寄存器设置内容的示例

(a) 从通道 3、5、7（实时输出通道（TRCq = 0））的定时器模式寄存器 q（TMRq）



(b) 从通道 3、5、7（实时输出通道（TRCq = 0））的其它寄存器

TOE0:TOEq	0: 通过计数操作, 停止TOq输出操作。 1: 通过计数操作, 允许 TOq 输出操作。
TO0:TOq	0: 从TOq输出低电平。 1: 从TOq输出高电平。
TOM0:TOMq	1: 设置组合操作模式。
TOT0:TOTq	1: 设置三角波PWM输出。
TOL0:TOLq	0: 正向逻辑输出 (高有效) 1: 反向输出 (低有效)
TDE0:TDEq	1: 允许死区时间控制。
TRE0:TREq	1: 允许实时输出。
TRO0:TROq	0: 作为实时输出输出低电平。 1: 作为实时输出输出高电平。
TRC0:TRCq	0: 作为实时输出触发产生通道不运行。
TME0:TMEq	0: 停止调制输出。 1: 允许调制输出。

备注 q = 03、05、07

图 7-96. 当使用互补调制输出功能时的操作过程（1/2）

软件操作	硬件状态
TAUS 默认设置	掉电状态 (停止供给时钟, 并且所有寄存器不能写入。)
设置PER2寄存器的TAU0EN位和TAUOPEN位为1。	上电状态。每个通道停止操作。 (开始供给时钟并且每个寄存器允许写入。)
设置TPS0 寄存器。 决定CK00和CK01的时钟频率。	
通道默认设置	通道停止操作。 (供给时钟并且产生一些功耗。)
设置要使用的8个通道的TMR00、TMR01、TMRp和TMRq寄存器（决定通道的操作模式）。 间隔（周期）值设置到主通道的TDR00寄存器，缩短的中断个数设置到从通道1的TDR01寄存器，占空比设置到从通道2、4、6的TDRp寄存器，死区时间宽度设置到从通道3、5、7的TDRq寄存器。	
设置从通道1。 设置TRC01位为1（触发产生通道）。 设置从通道2~7。 设置TOMp和TOMq位为1（组合操作模式）。 设置TOTp和TOTq位为1（三角波PWM输出）。 设置TOLp和TOLq位为0（正向逻辑输出）。 设置TDEp和TDEq位为0（停止死区时间控制）。 设置TREp和TREq位为1（允许实时输出）。 设置实时输出电平至TROp和TROq位。 设置TRCp和TRCq位为0（非触发产生通道）。 设置TMEp和TMEq位并决定调制输出控制。	TO00、TOp和TOq引脚为高阻输出状态。
设置TO00、TOp和TOq位并决定TOM输出的默认电平。	当端口模式寄存器在输出模式下，并且端口寄存器为0时，TO00、TOp和TOq输出默认设置电平。
设置TOE00、TOEp、TOEq位为1并允许TOM操作。	TOp和TOq不改变因为通道已停止操作。
将端口寄存器和端口模式寄存器清为0。	TO00、TOp和TOq引脚输出TOM的设定值。

备注 p = 02、04、06
q = 03、05、07

图 7-96. 当使用互补调制输出功能时的操作过程 (2/2)



8.1 比较器和可编程增益放大器的功能

比较器和可编程增益放大器有以下功能。

○ 比较器

- 比较器具有 2 个通道（CMP0 和 CMP1）。
- 负端输入引脚（CMP0M, CMP1M）和正端输入引脚（CMP0P, CMP1P）可以连接。
- 可编程增益放大器的输出信号可用作比较器的正端输入信号。（在这种情况下，输出信号同时输入到比较器 0 和 1 的通道。）
- 可以选择 CMP0M 和 CMP1M 引脚输入和内部产生的参考电压（每个比较器有六种组合）作为参考电压。
- 可以选择噪声消除数字滤波器的消除宽度功能。
- 当检测到过载电压时产生中断请求（INTCMP0 和 INTCMP1）。
- 比较器的输出信号连接到定时器阵列单元并设置定时器输出引脚（TOn）为高阻态。

○ 可编程增益放大器

- 可编程增益放大器把一个输入的模拟电压放大输出。可在八个放大系数中选择一个。
- 可编程增益放大器的输出信号可用作比较器的正端输入信号。（在这种情况下，输出信号同时输入到比较器 0 和 1 的通道。）
- 可编程增益放大器的输出信号可以被选择作为 A/D 转换器的模拟输入。

○ 过载电压检测功能

当过电流流过可编程增益放大器和比较器时，设置定时器输出引脚（TOn）为高阻态。此外，可以在以下两个功能中选择一个来设置引脚为高阻态。

- 两级过电流检测功能
- 过电流/电动势检测功能

通过设置两个参考电压数值（比较器 0 < 比较器 1）来设置引脚为高阻态，并且输入同样的信号到比较器 0 和 1 的正端输入。

• 两级过电流检测功能

<1> 比较器 0 的参考电压 < 输入信号电压 < 比较器 1 的参考电压

→ 设置 TOn 引脚为高阻态。

当输入信号电压低于比较器 0 的参考电压时，定时器输出与定时周期同步自动重新启动。

<2> 比较器 1 的参考电压 < 输入信号电压

→ 设置 TOn 引脚为高阻态。

当输入信号电压低于比较器 0 的参考电压时，通过设置寄存器，定时器输出与下一个定时周期同步自动重新启动。

- 过电流/电动势检测功能

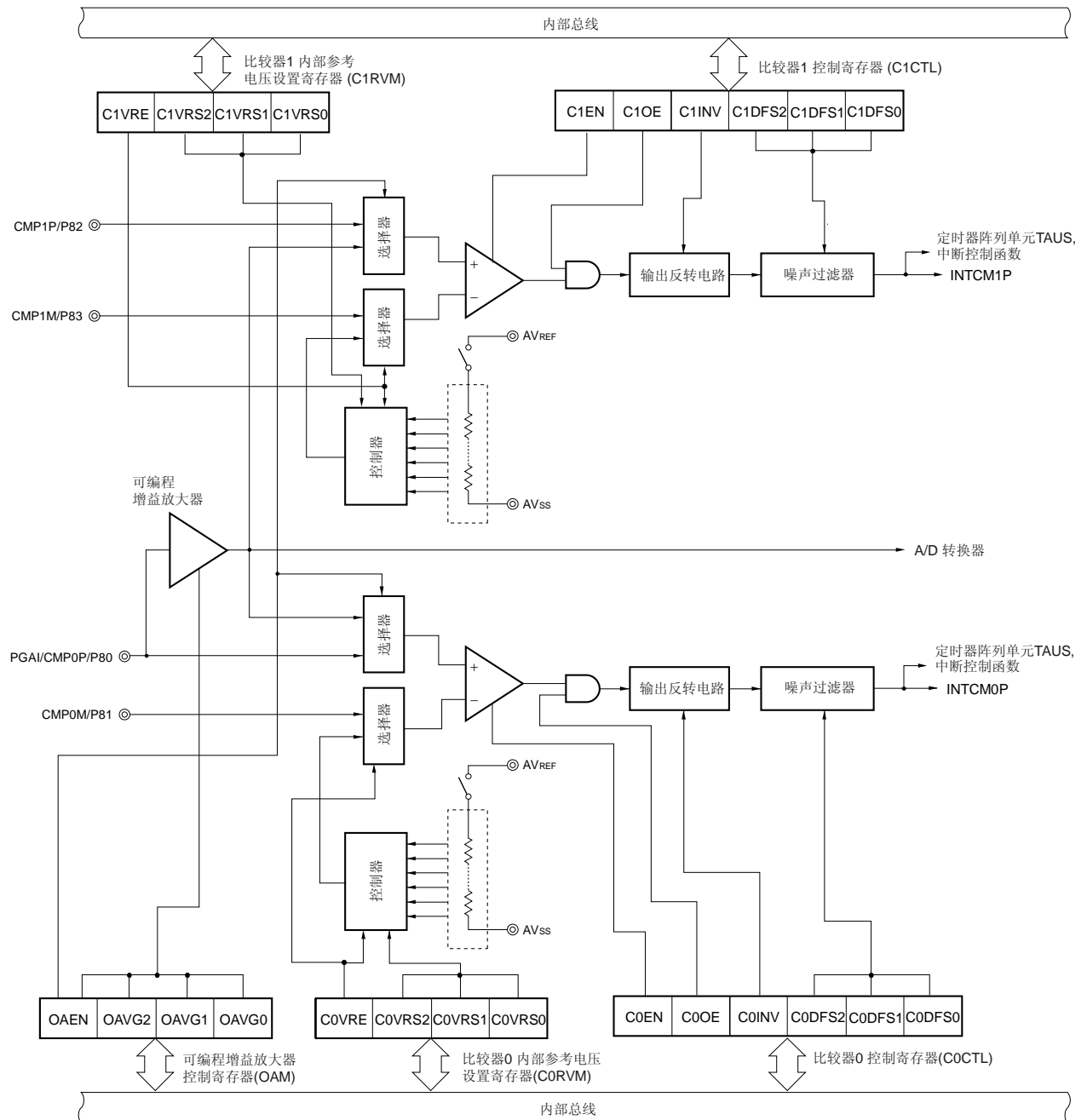
<1> 输入信号电压(电动势) < 比较器 0 或比较器 1 < 输入信号电压(过电流)

→ 设置 TOn 引脚为高阻态。

当输入信号电压高于比较器 0 的参考电压或低于比较器 1 的参考电压时，定时器输出与定时周期同步自动重新启动。

备注 n: 定时器通道编号(n = 02~07)

图 8-1 比较器/可编程增益放大器的框图



8.2 比较器和可编程增益放大器的配置

比较器和可编程增益放大器包含下列硬件。

表 8-1 比较器和可编程增益放大器的配置

项目	配置
控制寄存器	外围设备使能寄存器1(PER1) 可编程增益放大器控制寄存器(OAM) 比较器 0 和 1 控制寄存器 (C0CTL, C1CTL) 比较器 0 和 1 内部参考电压设定寄存器 (C0RVM, C1RVM) 端口输入模式寄存器 8(PIM8) 端口模式寄存器 8(PM8)

8.3 控制比较器和可编程增益放大器的寄存器

比较器和可编程增益放大器使用以下 8 个寄存器。

- 外围设备使能寄存器 1(PER1)
- 可编程增益放大器控制寄存器(OAM)
- 比较器 0 和 1 控制寄存器 (C0CTL, C1CTL)
- 比较器 0 和 1 内部参考电压设定寄存器 (C0RVM, C1RVM)
- 端口输入模式寄存器 8(PIM8)
- 端口模式寄存器 8(PM8)

(1) 外围设备使能寄存器 1(PER1)

此寄存器用来设置每个外围硬件模块是否能使用。停止向未硬件供给时钟可以降低功耗和噪声。

确认设置第 3 位(OACMPEN)为 1 来使用比较器和可编程增益放大器。

PER1 可由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

- 注意事项
- 1. 当设置比较器或可编程增益放大器时，首先确认设置 OACMPEN 为 1。当 OACMPEN 设置为 0 时，写入比较器或可编程增益放大器的控制寄存器将被忽略并且所有读出数值被初始化。
 - 2. 确认设置 PER1 寄存器的位 0~2 和位 4~7 为“0”。

图 8-2. 外围设备使能寄存器 1(PER1)的格式

地址: F00F1H 复位后: 00H R/W

符号	7	6	5	4	<3>	2	1	0
PER1	0	0	0	0	OACMPEN	0	0	0

OACMPEN	可编程增益放大器输入时钟控制
0	停止输入时钟供给 <ul style="list-style-type: none">• 用于比较器或可编程增益放大器的SFR不能写入。• 比较器或可编程增益放大器复位。
1	供给输入时钟 <ul style="list-style-type: none">• 用于比较器或可编程增益放大器的SFR能读出和写入。

(2) 可编程增益放大器控制寄存器(OAM)

该寄存器用来允许或禁止可编程增益放大器的操作及设置放大系数。

OAM 可由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 8-3 可编程增益放大器控制寄存器(OAM)的格式

地址: F0240H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
OAM	OAEN	0	0	0	0	OAVG2	OAVG1	OAVG0

OAEN	可编程增益放大器操作控制
0	停止操作
1	允许操作 允许从可编程增益放大器输入引脚外部输入(PGAI) 输入可编程增益放大器的输出信号作为比较器0和1的正端输入电压

OAVG2	OAVG1	OAVG0	输入电压放大系数设置
0	0	0	×1
0	0	1	×4
0	1	0	×6
0	1	1	×8
1	0	0	×10
1	0	1	×12
与上述不同的值			设置禁止

注意事项 1. 在允许(OAEN = 1)可编程增益放大器工作前, 设置放大系数。禁止在工作允许状态(OAEN = 1)中改变放大系数。

2. 当 CnCTL 寄存器的 CnEN 设置为 1 时, 因为可编程增益放大器输出信号成为比较器 0 和 1 的正端输入电压, 同时可编程增益放大器进入工作允许状态, 所以比较器 0 和 1 可能会有误检测。因此, 请在比较器输出信号停止状态时设置 OAM。

备注 n = 0, 1

(3) 比较器 n 控制寄存器(CnCTL)

此寄存器用来控制比较器 n 的操作、允许或禁止比较器输出、输出反向以及设置噪声消除宽度。

CnCTL 可由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 8-4. 比较器 n 控制寄存器(CnCTL)的格式

地址：F0241H (C0CTL)，F0242H (C1CTL) 复位后：00H R/W

符号	<7>	6	5	4	3	2	1	0
CnCTL	CnEN	0	0	CnOE	CnINV	CnDFS2	CnDFS1	CnDFS0

CnEN	比较器操作控制
0	停止操作
1	允许操作 允许输入到比较器n的正端和负端的外部引脚

CnOE	允许或禁止比较器输出
0	禁止输出（输出信号=固定在低电平）
1	允许输出

CnINV	输出反相设置
0	正向
1	反向

CnDFS2	CnDFS1	CnDFS0	噪声消除宽度设置（f _{CLK} = 20 MHz）
0	0	0	未使用噪声滤波器
0	0	1	250 ns
0	1	0	500 ns
0	1	1	1 μs
1	0	0	2 μs
与上述不同的值			设置禁止

- 注意事项**
1. 在设置比较器输出为禁止状态（CnOE = 0）后，重写 CnINV 和 CnDFS2~CnDFS0。
 2. 使用噪声消除宽度，根据设定值，可能会多消除一个 CPU 时钟（f_{CLK}）。
（例如：当 f_{CLK} = 20 MHz，CnDFS2~CnDFS0 = 001，噪声消除宽度 = 250~300 ns）
 3. 当把比较器和一个可编程增益放大器组合操作的时候，在设置可编程增益放大器的操作之后，再设置比较器操作（见图 8-10 和图 8-11）。
 4. 当 CnRVM 寄存器的 CnVRE 设置为（1）时，无论比较器操作（CnEN）是允许或禁止的值，比较器负端的外部引脚输入将被阻断。

- 备注**
1. f_{CLK}: CPU 或外围硬件时钟频率
 2. n = 0, 1

(4) 比较器 n 内部参考电压选择寄存器(CnRVM)

此寄存器用于设置比较器 n 的内部参考电压。内部参考电压能从使用 AV_{REF} 的六组电压中选择。

CnRVM 能由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 8-5. 比较器 n 内部参考电压选择寄存器 (CnRVM) 的格式

地址: F0243H (C0RVM), F0244H (C1RVM) 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
CnRVM	CnVRE	0	0	0	0	CnVRS2	CnVRS1	CnVRS0

CnVRE	内部参考电压操作控制
0	停止操作
1	允许操作 连接内部参考电压到比较器n的负端输入

CnVRS2	CnVRS1	CnVRS0	参考电压设定	
			比较器0(n = 0)可设置的参考电压	比较器1(n = 1)可设置的参考电压
0	0	0	设置禁止	
0	0	1	$2AV_{REF}/16$	$3AV_{REF}/16$
0	1	0	$4AV_{REF}/16$	$5AV_{REF}/16$
0	1	1	$6AV_{REF}/16$	$7AV_{REF}/16$
1	0	0	$8AV_{REF}/16$	$9AV_{REF}/16$
1	0	1	$10AV_{REF}/16$	$11AV_{REF}/16$
1	1	0	$12AV_{REF}/16$	$13AV_{REF}/16$
1	1	1	设置禁止	

注意事项 1. 当内部参考电压操作停止(CnVRE = 0)时比较器操作由 CnEN 控制。

2. 当 CnVRE 设置为(1)时, 不管比较器操作值(CnEN)是允许或禁止, 比较器负端的外部引脚输入将被阻断。

3. 在允许内部参考电压(CnVRE = 1)操作前设置参考电压。禁止在工作允许状态(CnVRE = 1)中改变参考电压设置。

备注 n = 0, 1

(5) 端口输入模式寄存器 8 (PIM8)

<R> 寄存器以 1 位单元允许或禁止端口 8 的数字输入。
使用比较器或可编程增益放大器设为数字输入禁止（使用端口 8 作为模拟输入）。因为默认值是数字输入禁止（使用端口 8 作为模拟输入），所以使用端口功能或外部中断功能和定时器高阻控制功能时，要设置数字输入允许。
PIM8 可以通过 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 00H。

图 8-6. 端口输入模式寄存器 8 (PIM8) 的格式

地址: F0048H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PIM8	0	0	0	0	PIM83	PIM82	PIM81	PIM80

<R>

PIM8n	选择允许或禁止P8n引脚数字输入（n = 0~3）
0	数字输入禁止（使用端口8作为模拟输入）。
1	数字输入允许

(6) 端口模式寄存器 8 (PM8)

该寄存器以 1 位单元用于设置端口 8 的输入或输出。
设置 PM80~PM83 位为 1 来使用 P80/CMP0P/TMOFF0/INTP3/PGAI、P81/CMP0M、P82/CMP1P/TMOFF1/INTP7 或 P83/CMP1M 引脚作为比较器正端或负端输入功能或可编程增益放大器输入功能。
此时，P80~P83 的输出锁存可以是 0 或 1。
PM80~PM83 可以通过 1 位或 8 位存储器操作指令设置。
复位信号的产生将此寄存器设置为 FFH。

图 8-7. 端口模式寄存器 8 (PM8) 的格式

地址: FFF28H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM8	1	1	1	1	PM83	PM82	PM81	PM80

PM8n	P8n引脚I/O模式选择（n = 0~3）
0	输出模式（输出缓冲器开）
1	输入模式（输出缓冲器关）

注意事项 当选择 CMP0P、CMP1P 引脚作为比较器正端输入，并且负端使用内部参考电压时，作为 CMP0M、CMP1M 引脚的复用端口功能可用于输入模式。但是，禁止使用输出模式。此外，访问端口寄存器 8 (P8) 也被禁止。

8.4 比较器和可编程增益放大器的操作

8.4.1 启动比较器和可编程增益放大器的操作

启动比较器和可编程增益放大器操作的过程，对应每种使用方法，分别描述如下。

- 仅使用比较器
 - 使用外部引脚输入作为比较器参考电压（图 8-8）
 - 使用内部参考电压作为比较器参考电压（图 8-9）
- 使用比较器和可编程增益放大器（使用可编程增益放大器输出电压作为比较器比较电压输入）
 - 使用外部引脚输入作为比较器参考电压（图 8-10）
 - 使用内部参考电压作为比较器参考电压（图 8-11）
- 使用可编程增益放大器输出电压作为 A/D 转换器模拟输入（图 8-12）

图 8-8. 使用外部引脚输入作为比较器参考电压（仅使用比较器）

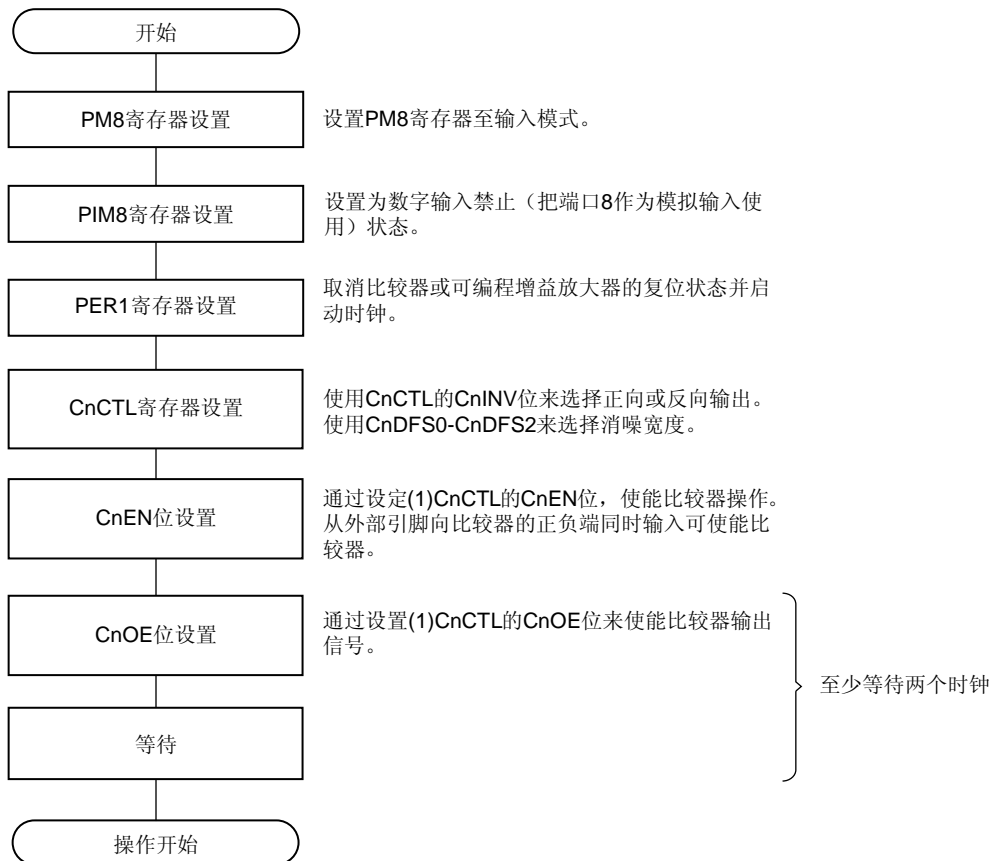


图 8-9. 使用内部参考电压作为比较器参考电压(仅使用比较器)

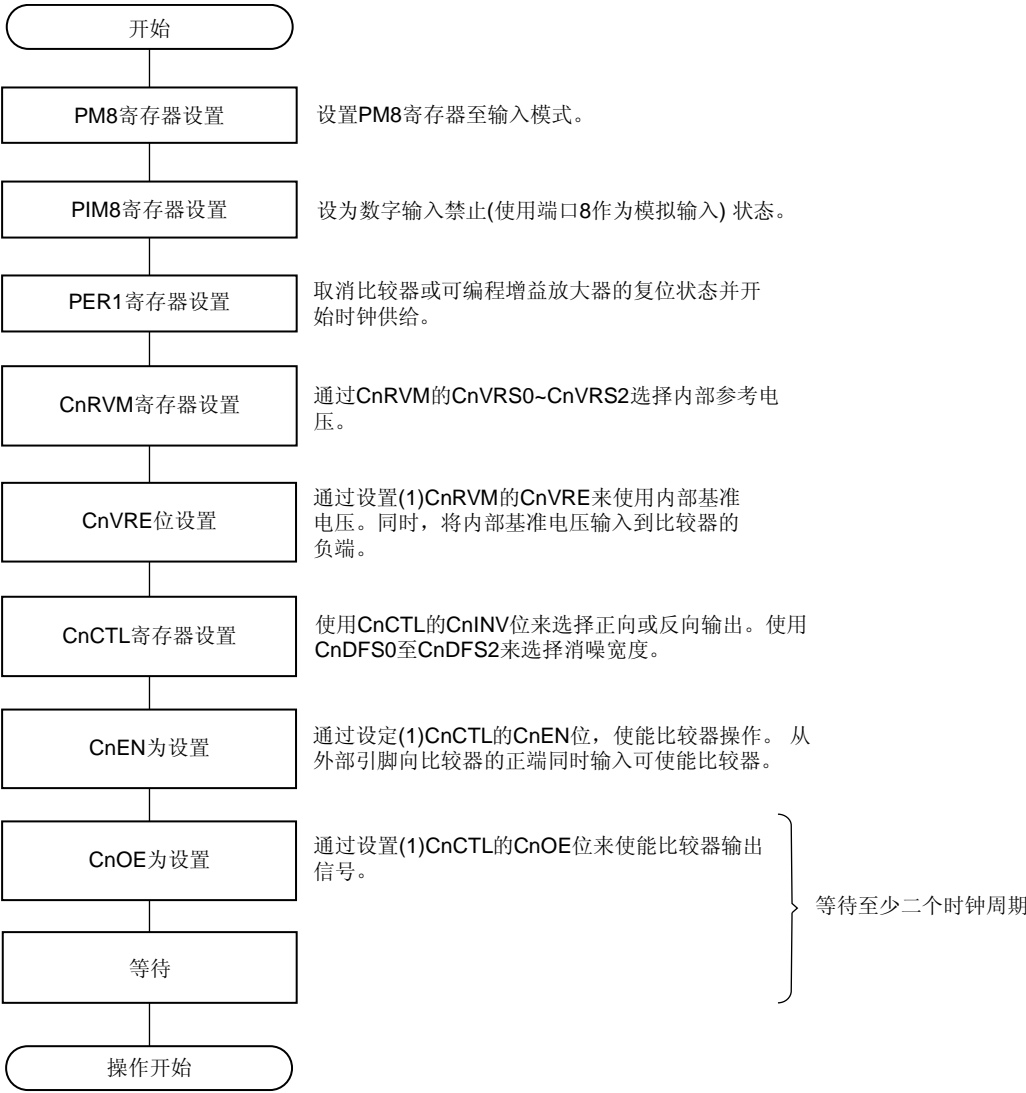
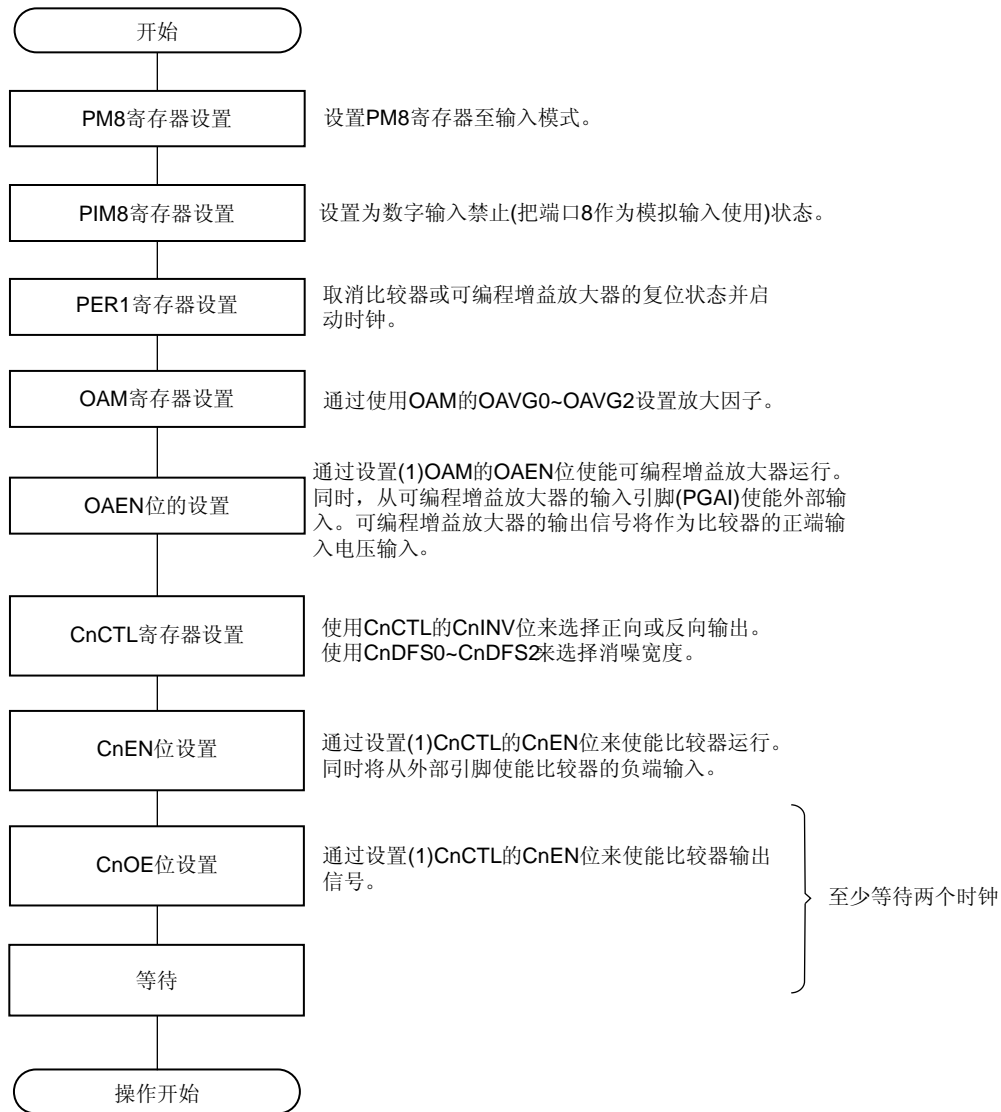


图 8-10 使用外部引脚输入作为比较器参考电压（使用比较器和可编程增益放大器）



备注 n = 0, 1

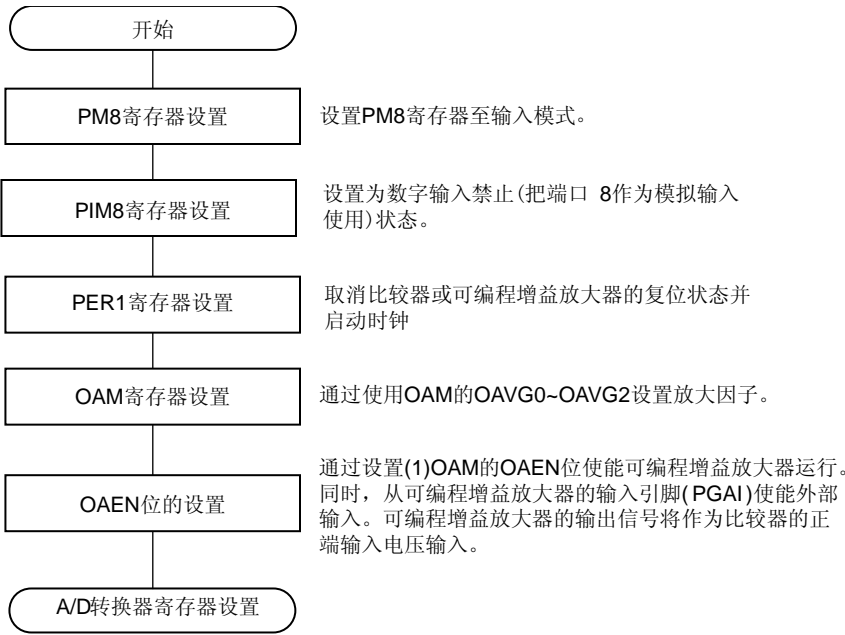
图 8-11 使用内部参考电压作为比较器参考电压（使用比较器和可编程增益放大器）



备注 n = 0, 1

通过使用 A/D 转换器的模拟输入通道配置寄存器（ADS），选择可编程增益放大器输出信号作为模拟输入之前，完成下面设置（参考 12.4.1 A/D 转换器的基本操作）。

图 8-12 使用可编程增益放大器输出电压作为 A/D 转换器模拟输入



备注 n = 0, 1

8.4.2 停止比较器和可编程增益放大器的操作

停止比较器和可编程增益放大器操作的过程，对应每种使用方法，分别描述如下。

- 仅使用比较器（图 8-13）
- 使用可编程增益放大器输出电压作为比较器比较电压输入（图 8-14）
- 使用可编程增益放大器输出电压作为 A/D 转换器模拟输入（图 8-15）

图 8-13. 仅使用比较器

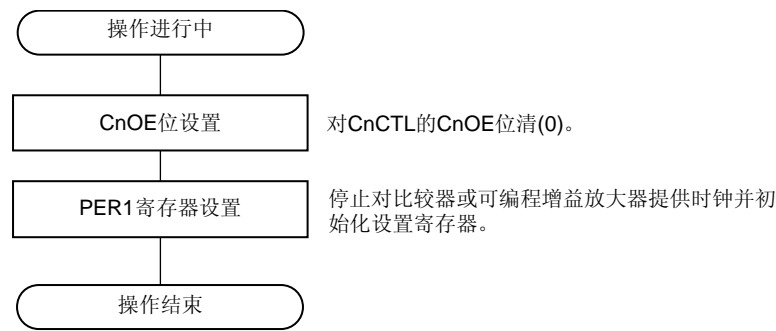


图 8-14 使用可编程增益放大器输出电压作为比较器比较电压输入

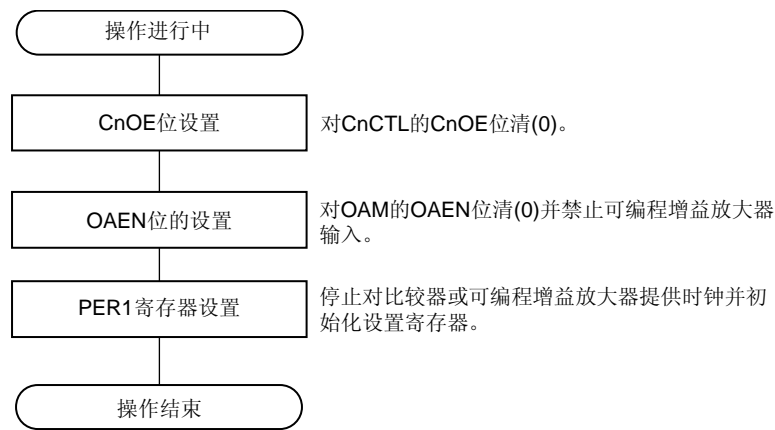
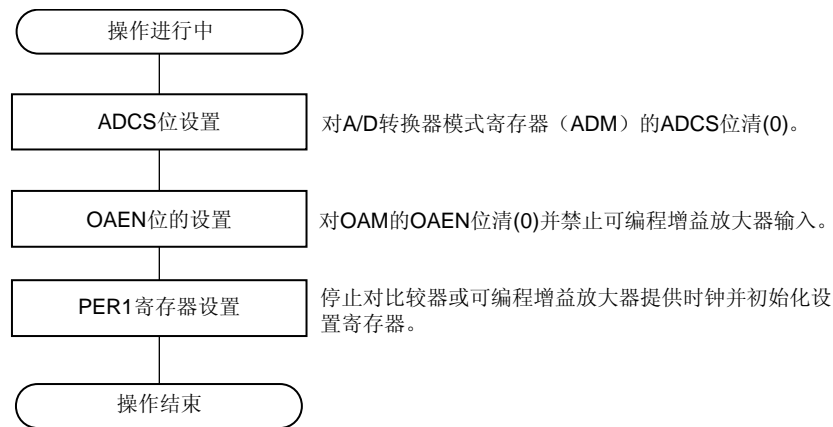


图 8-15 使用可编程增益放大器输出电压作为 A/D 转换器模拟输入



备注 n = 0, 1

第 9 章 实时计数器

9.1 实时计数器的功能

实时计数器有如下特点。

- 具有年、月、星期、日、小时、分钟和秒的计数器，可以共计 99 年。
- 固定周期中断功能（周期：1 月~0.5 秒）
- 闹钟中断功能（报警：星期、小时、分钟）
- 间隔中断功能

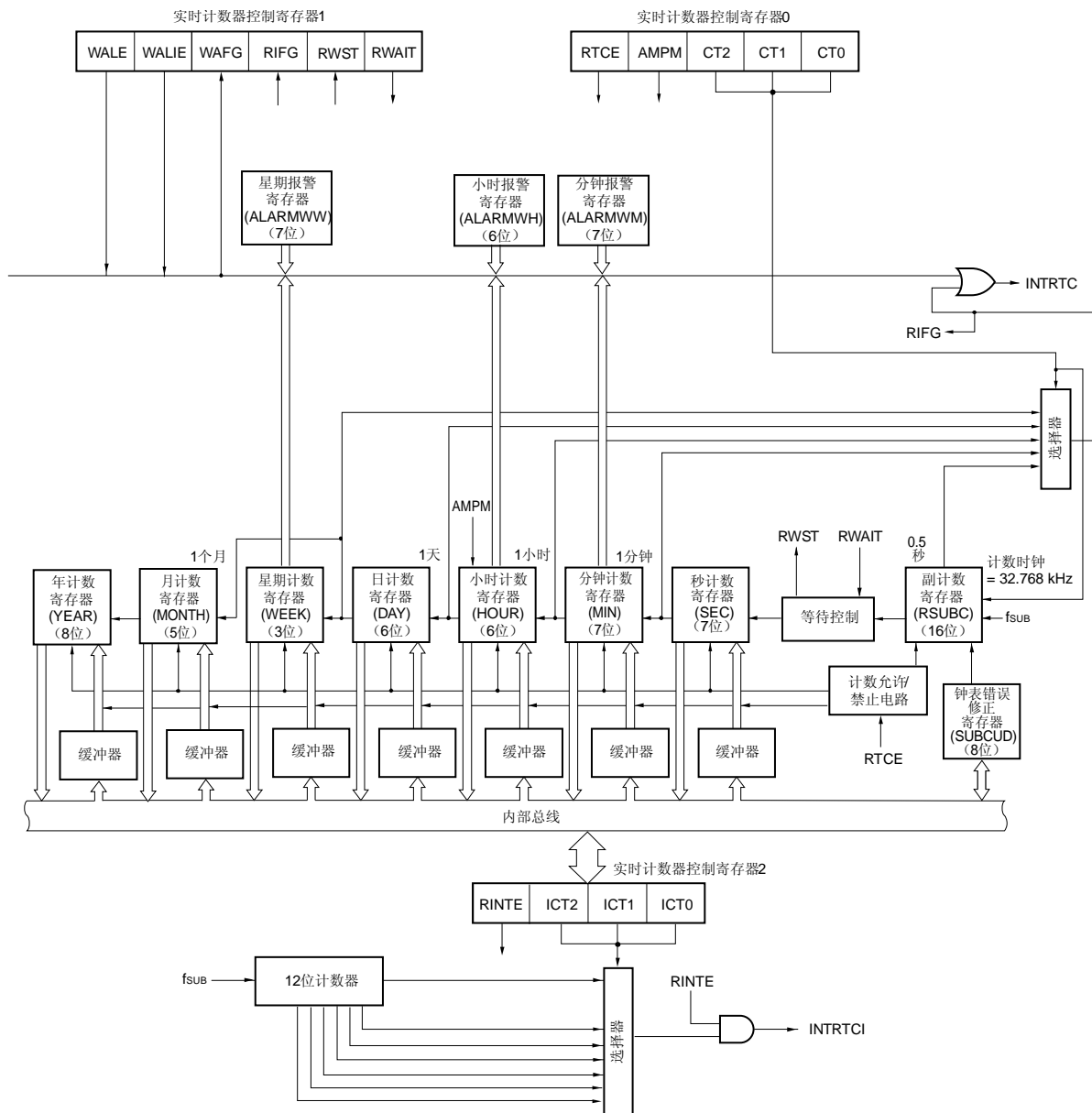
9.2 实时计数器的配置

实时计数器包括如下硬件。

表 9-1. 实时计数器的配置

项目	攝柯
控制寄存器	外围设备使能寄存器 0 (PER0)
	实时计数器控制寄存器 0 (RTCC0)
	实时计数器控制寄存器 1 (RTCC1)
	实时计数器控制寄存器 2 (RTCC2)
	副计数寄存器 (RSUBC)
	秒计数寄存器 (SEC)
	分钟计数寄存器 (MIN)
	小时计数寄存器 (HOUR)
	日计数寄存器 (DAY)
	星期计数寄存器 (WEEK)
	月计数寄存器 (MONTH)
	年计数寄存器 (YEAR)
	钟表错误修正寄存器 (SUBCUD)
	分钟报警寄存器 (ALARMWM)
	小时报警寄存器 (ALARMWH)
	星期报警寄存器 (ALARMWW)

图 9-1. 实时计数器的框图



9.3 控制实时计数器的寄存器

以下 16 个寄存器用于控制实时计数器。

(1) 外围设备使能寄存器 0 (PER0)

PER0 用于允许或者禁止使用每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少消耗和噪音。
使用实时计数器时，务必设置此寄存器的第 7 位 (RTCEN) 为 1。
可以通过 1 位或者 8 位存储器操作指令来设置 PER0。
复位后这个寄存器的值为 00H。

图 9-2. 外围设备使能寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	6	<5>	<4>	3	<2>	1	0
PER0	RTCEN	0	ADCEN	0	0	SAU0EN	0	0

RTCEN	实时计数器(RTC)输入时钟的控制 ^注
0	停止输入时钟的供给 <ul style="list-style-type: none">由实时计数器(RTC)使用的 SFR 不能写入。实时计数器(RTC)在复位状态。
1	供给输入时钟 <ul style="list-style-type: none">由实时计数器(RTC)使用的 SFR 可读/写。

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

注意事项: 1. 使用实时计数器时，首先设置 RTCEN 位为 1，直至副系统时钟(f_{SUB})振荡稳定。如果 RTCEN = 0，写入到实时计数器的控制寄存器被忽略，并且，即使读取寄存器，只能读到默认值。
2. 确保将 PER0 寄存器的第 0、1、3、4 和 6 位清零。

(2) 实时计数器控制寄存器 0 (RTCC0)

RTCC0 寄存器是一个 8 位寄存器，它用于启动或停止实时计数器操作、设置 12 或 24 小时系统和固定周期中断功能。
可以由 1 位或 8 位存储器操作指令设置 RTCC0。
复位后这个寄存器的值为 00H。

图 9-3. 实时计数器控制寄存器 0 (RTCC0)的格式

地址: FFF9DH 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
RTCC0	RTCE	0	0	0	AMPM	CT2	CT1	CT0

RTCE	实时计数器操作控制
0	停止计数器操作。
1	启动计数器操作。

AMPM	选择12-/24小时系统
0	12 小时系统(a.m. 和 p.m. 显示)
1	24 小时系统

- 改变 AMPM 的值, 要将 RWAIT (RTCC1 的第 0 位)置为 1, 并重置小时计数寄存器(HOUR)。
- 表 9-2 表示显示的时间数字显示。

CT2	CT1	CT0	固定周期中断(INTRTC)选择
0	0	0	不使用固定周期中断功能
0	0	1	每 0.5 秒一次(与秒累加同步)
0	1	0	每 1 秒一次(与秒累加相同)
0	1	1	每 1 分钟一次(每分钟的 00 秒)
1	0	0	每 1 小时一次(每小时的 00 分 00 秒)
1	0	1	每 1 天一次(每天的 00 小时, 00 分, 00 秒)
1	1	×	每 1 月一次(每月的第 1 天, 00 小时 a. m., 00 分, 00 秒)

CT2~CT0 值改变后, 将中断请求标志清零。

备注 ×: 不必考虑

表 9-2. 时间数字显示

24 小时系统	12 小时系统	24 小时系统	12 小时系统
00	12 (AM12)	12	32 (PM12)
01	01 (AM1)	13	21 (PM1)
02	02 (AM2)	14	22 (PM2)
03	03 (AM3)	15	23 (PM3)
04	04 (AM4)	16	24 (PM4)
05	05 (AM5)	17	25 (PM5)
06	06 (AM6)	18	26 (PM6)
07	07 (AM7)	19	27 (PM7)
08	08 (AM8)	20	28 (PM8)
09	09 (AM9)	21	29 (PM9)
10	10 (AM10)	22	30 (PM10)
11	11 (AM11)	23	31 (PM11)

(3) 实时计数器控制寄存器 1 (RTCC1)

RTCC1 寄存器是 8 位寄存器，它用于控制报警中断功能和计数器等待时间。
可以由 1 位或 8 位存储器操作指令设置 RTCC1。
复位后这个寄存器的值为 00H。

图 9-4. 实时计数器控制寄存器 1 (RTCC1)的格式 (1/2)

地址: FFF9EH 复位后: 00H R/W

符号	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	报警操作控制
0	匹配操作无效
1	匹配操作有效
设置报警寄存器(WALIE 的 RTCC1 位、ALARMWM 寄存器、ALARMWH 寄存器和 ALARMWW 寄存器)，禁止 WALE(将其清零)。	

WALIE	控制报警中断(INTRTC)功能操作
0	报警匹配时不产生中断。
1	报警匹配时产生中断。

WAFG	报警检测状态标志
0	报警不匹配
1	检测报警匹配
此状态标志显示报警匹配检测。仅当 WALE = 1 和检测到报警匹配后置一个时钟(32 kHz)为 1 时有效。当写入“0”时，清零该标志。写入“1”为无效。	

图 9-4. 实时计数器控制寄存器 1 (RTCC1)的格式 (2/2)

RIFG	固定周期中断状态标志
0	不产生固定周期中断。
1	产生固定周期中断。
此标志显示固定周期中断产生状态。当产生固定周期中断时，设置为“1”。当写入“0”时，清零该标志。写入“1”为无效。	

RWST	实时计数器等待状态标志
0	计数器操作。
1	读或写计数器值的模式。
此状态标志表明 RWAIT 设置是否有效。 读或写计数器值之前，请确认此标志值为 1。	

RWAIT	实时计数器的等待控制
0	设置计数器操作。
1	停止 SEC~YEAR 计数器。读或写计数器值的模式。
此位控制计数器操作。 确认写入“1”来读取或写入计数器值。 因为 RSUBC 操作继续，在 1 秒内完成读或写，并将此位清零。当 RWAIT = 1 时，它占用 1 个 时钟 (32kHz) 直到计数器值被读出或写入。 当 RWAIT = 1 时，如果 RSUBC 溢出，在 RWAIT = 0 后它累加。如果秒计数寄存器写入，因为 RSUBC 清零，它将不能累加。	

- <R>
- 注意事项

如果以 1 位操作指令执行写入 RTCC1 寄存器，RIFG 和 WAFG 标志可能被清零。因此，要执行 RTCC1 寄存器写入，必须使用 8 位操作指令。为防止 RIFG 和 WAFG 标志在写入、空写入时被清零，将相应的位设为 1。如果 RIFG 和 WAFG 标志未被使用，且值可能被改变，则 RTCC1 寄存器可以通过使用 1 位操作指令写入。
- 备注

固定周期中断和报警匹配中断使用相同的中断源 (INTRTC)。当同时使用这两种类型中断时，在 INTRTC 产生的基础上，通过检查固定周期中断状态标志(RIFG)和报警检测状态标志(WAFG)，来判断产生了哪种中断。

(4) 实时计数器控制寄存器 2 (RTCC2)

RTCC2 寄存器是一个 8 位寄存器，它用于控制间隔中断功能。

可以由 1 位或 8 位存储器操作指令设置 RTCC2。

复位后这个寄存器的值为 00H。

图 9-5. 实时计数器控制寄存器 2 (RTCC2)的格式

地址: FFF9FH 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
RTCC2	RINTE	0	0	0	0	ICT2	ICT1	ICT0

RINTE	ICT2	ICT1	ICT0	选择间隔中断(INTRTCI)
0	×	×	×	不产生间隔中断
1	0	0	0	$2^6/f_{XT}$ (1.953125 ms)
1	0	0	1	$2^7/f_{XT}$ (3.90625 ms)
1	0	1	0	$2^8/f_{XT}$ (7.8125 ms)
1	0	1	1	$2^9/f_{XT}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{XT}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{XT}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{XT}$ (125 ms)

当 RINTE = 0 时，更改 ICT2、ICT1 和 ICT0。

(5) 副计数寄存器 (RSUBC)

RSUBC 寄存器为 16 位寄存器，用于计算实时计数器的 1 秒的参考时间。它的取值从 0000H~7FFFH，且以 32.768kHz 的时钟来计 1 秒。

可以 16 位存储器操作指令设置 RSUBC。

复位后这个寄存器的值为 0000H。

- 注意事项
1. 当使用 SUBCUD 寄存器纠正，值将变为 8000H 或更大。
 2. 写入秒计数寄存器产生复位会将该寄存器清零。
 3. 在操作期间读取该寄存器，因为读出的是变化中的值，不能保证读出值的有效。

图 9-6. 副计数寄存器(RSUBC)的格式

地址: FFF90H 复位后: 0000H R

符号	7	6	5	4	3	2	1	0
RSUBC	SUBC7	SUBC6	SUBC5	SUBC4	SUBC3	SUBC2	SUBC1	SUBC0

地址: FFF91H 复位后: 0000H R

符号	7	6	5	4	3	2	1	0
RSUBC	SUBC15	SUBC14	SUBC13	SUBC12	SUBC11	SUBC10	SUBC9	SUBC8

(6) 秒计数寄存器 (SEC)

SEC 寄存器是 8 位寄存器，为 0~59（十进制）值和显示计数的秒值。
当副计数器溢出，它将开始计数。
当数据写入该寄存器时，将写入缓冲器并有 2 个时钟（32.768 kHz）的延迟。以 BCD 码设置 00~59 十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。
可以 8 位存储器操作指令设置 SEC。
复位后这个寄存器的值为 00H。

图 9-7. 秒计数寄存器 (SEC) 的格式

地址: FFF92H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

(7) 分钟计数寄存器 (MIN)

MIN 寄存器是 8 位寄存器，为 0~59（十进制）值和显示计数的分钟值。
当秒计数器溢出它将开始计数。
当数据写入该寄存器时，将写入缓冲器并有 2 个时钟（32.768 kHz）的延迟。以 BCD 码设置 00~59 十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。
可以 8 位存储器操作指令设置 MIN。
复位后这个寄存器的值为 00H。

图 9-8. 分钟计数寄存器 (MIN) 的格式

地址: FFF93H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

(8) 小时计数寄存器 (HOUR)

HOUR 寄存器是 8 位寄存器，为 0~23 或 1~12（十进制）值和显示计数的小时值。
当分钟计数器溢出，它将开始计数。
当数据写入该寄存器时，将写入缓冲器并有 2 个时钟（32.768 kHz）的延迟。以 BCD 码设置 00~23、01~12 或 21~32 十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。
可以 8 位存储器操作指令设置 HOUR。
复位信号的产生将寄存器置为 12H。
如果当 AMPM 位（RTCC0 寄存器的第 3 位）置为 1，复位后该寄存器的值清零（00H）。

图 9-9. 小时计数寄存器 (HOUR) 的格式

地址: FFF94H	复位后: 12H	R/W						
符号	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

注意事项 当 AMPM = 0（如果选择 12 小时系统），HOUR 的第 5 位（HOUR20）显示 AM(0)/PM(1)。

(9) 日计数寄存器 (DAY)

DAY 寄存器是 8 位寄存器，为 1~31（十进制）值和显示计数的天数值。
当小时计数器溢出它将开始计数。
计数器计数如下。

- 01 ~ 31（一月、三月、五月、七月、八月、十月、十二月）
- 01 ~ 30（四月、六月、九月、十一月）
- 01 ~ 29（二月、闰年）
- 01 ~ 28（二月、正常年）

当数据写入该寄存器时，将写入缓冲器并有 2 个时钟（32.768 kHz）的延迟。以 BCD 码设置 01~31 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。
可以 8 位存储器操作指令设置 DAY。
复位信号的产生将寄存器置为 01H。

图 9-10. 日期计数寄存器 (DAY) 的格式

地址: FFF96H	复位后: 01H	R/W						
符号	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

(10) 星期计数寄存器（WEEK）

WEEK 寄存器是 8 位寄存器，为 0~6（十进制）值和显示计数的星期值。
当日计数器溢出它将开始计数。
当数据写入该寄存器时，将写入缓冲器并有 2 个时钟（32.768 kHz）的延迟。以 BCD 码设置 00~06 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。
可以 8 位存储器操作指令设置 WEEK。
复位后这个寄存器的值为 00H。

图 9-11. 星期计数寄存器(WEEK)的格式

地址: FFF95H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

<R> 注意事项 对应于月计数寄存器和日寄存器的值不会自动存贮到星期计数寄存器。
 复位解除后，确保设置星期计数寄存器如下。

日	WEEK
星期日	00H
星期一	01H
星期二	02H
星期三	03H
星期四	04H
星期五	05H
星期六	06H

(11) 月计数寄存器 (MONTH)

MONTH 寄存器是 8 位寄存器，为 0~12(十进制)值和显示计数的月值。
当日计数器溢出它将开始计数。
当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 01~12 的十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。
可以 8 位存储器操作指令设置 MONTH。
复位信号的产生将寄存器置为 01H。

图 9-12. 月计数寄存器(MONTH)的格式

地址: FFF97H	复位后: 01H	R/W						
符号	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

(12) 年计数寄存器 (YEAR)

YEAR 寄存器是 8 位寄存器，为 0~99(十进制)值和显示计数的月值。
当月计数器溢出它将开始计数。
值 00、04、08、...、92 和 96 表示闰年。
当数据写入该寄存器时，将写入缓冲器并有 2 个时钟(32.768 kHz)的延迟。以 BCD 码设置 00~99 十进制数到该寄存器。如果值超出范围，寄存器值在 1 个周期后将返回到正常值。
可以 8 位存储器操作指令设置 YEAR。
复位后这个寄存器的值为 00H。

图 9-13. 年计数寄存器(YEAR)的格式

地址: FFF98H 复位后: 00H R/W							
符号	7	6	5	4	3	2	1 0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2 YEAR1

(13) 钟表错误修正寄存器 (SUBCUD)

该寄存器用作纠正副计数器(RSUBC)的计数值。
可以 8 位存储器操作指令设置 SUBCUD。
复位后这个寄存器的值为 00H。

图 9-14. 钟表错误修正寄存器(SUBCUD)的格式

地址: FFF99H 复位后: 00H R/W							
符号	7	6	5	4	3	2	1 0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1 F0

DEV	设置钟表纠错时间
0	当秒数字为 00, 20 或 40 时纠正钟表错误。
1	仅当秒数字为 00 时纠正钟表错误。

F6	设置钟表纠错方法
0	由 $\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2$ 增大
1	由 $\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2$ 减小
当 (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *) 时, 钟表错误不能修正。 /F5 ~ /F0 相应位的反向值 (当 111100 时为 000011)。	

(14) 分钟报警寄存器(ALARMWM)

该寄存器用作设置报警分钟。

可以 8 位存储器操作指令设置 ALARMWM。

复位后这个寄存器的值为 00H。

注意事项 以 BCD 码设置 00~59 十进制数到该寄存器。如果设置的值超出范围，报警将不检测。

图 9-15. 分钟报警寄存器(ALARMWM)的格式

地址: FFF9AH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

(15) 小时报警寄存器(ALARMWH)

该寄存器用于设置报警小时。

可以 8 位存储器操作指令设置 ALARMWH。

复位信号的产生将寄存器置为 12H。

如果当 AMPM 位(RTCC0 寄存器的第 3 位)置为 1，复位后该寄存器的值清零(00H)。

注意事项 以 BCD 码设置 00~23、01~12 或 21~32 十进制数到该寄存器。如果设置的值超出范围，报警将不检测。

图 9-16. 小时报警寄存器(ALARMWH)的格式

地址: FFF9BH 复位后: 12H R/W

符号	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

注意事项 如果 AMPM = 0(选择 12 小时系统)，ALARMWH 的第 5 位 (WH20) 显示 AM(0)/PM(1)。

(16) 星期报警寄存器(ALARMWW)

该寄存器用于设置报警数据。

可以 8 位存储器操作指令设置 ALARMWW。

复位后这个寄存器的值为 00H。

图 9-17. 星期报警寄存器(ALARMWW)的格式

地址: FFF9CH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

以下是设置报警的示例。

报警的时间	日							12 小时显示				24 小时显示			
	星期日	星期一	星期二	星期三	星期四	星期五	星期六	小时	小时	分钟	分钟	小时	小时	分钟	分钟
	W	W	W	W	W	W	W	10	1	10	1	10	1	10	1
	0	1	2	3	4	5	6								
每天, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
每天, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
每天, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
星期一到星期五 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
星期日 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
星期一、星期三、星期五 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

9.4 实时计数器操作

9.4.1 启动实时计数器的操作

图 9-18. 启动实时计数器的操作流程



注 首先设置 RTCEN 位为 1，副系统时钟(f_{SUB})振荡稳定。

9.4.2 读/写实时计数器

当 $RWAIT = 1$ 时，读出或写入计数器。

图 9-19. 读出实时计数器的操作流程



注 设置 STOP 模式前必须置 $RWST = 0$ 。

注意事项 在 1 秒内完成置 $RWAIT$ 为 1 到清 $RWAIT$ 为 0 的连续操作。

备注 SEC、MIN、HOUR、WEEK、DAY、MONTH 和 YEAR 将顺次读出。
不用设置所有的寄存器，仅部分寄存器可被读出。

图 9-20. 写入实时计数器的操作流程



注 设置 STOP 模式前必须置 RWST = 0 。

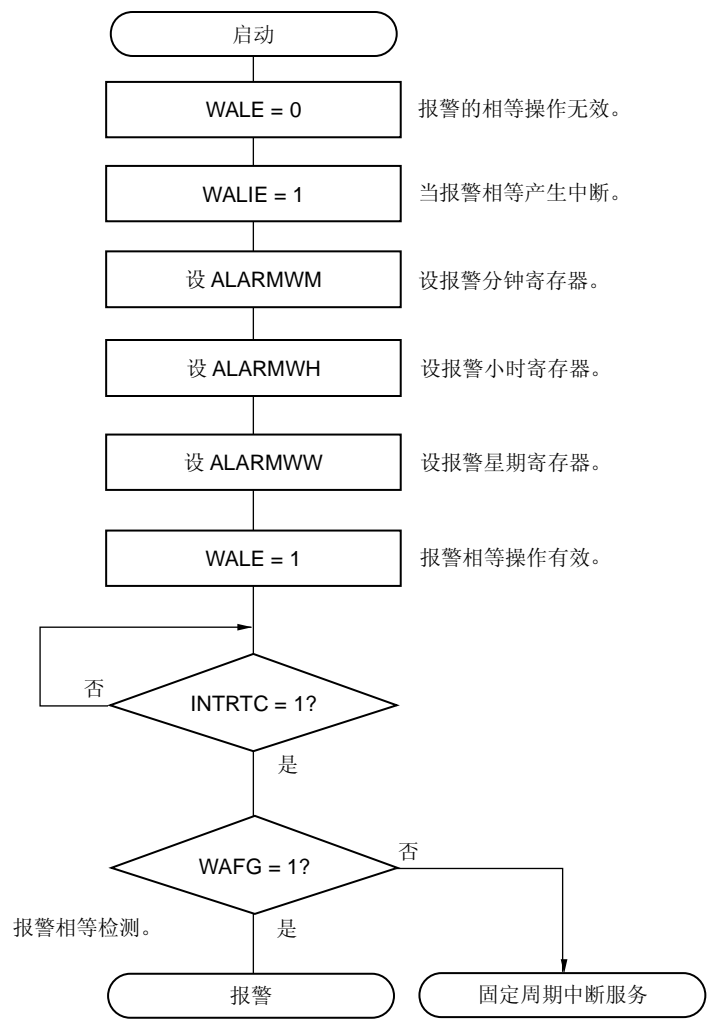
注意事项 在 1 秒内完成置 RWAIT 为 1 到清 RWAIT 为 0 的连续操作。

备注 SEC、MIN、HOUR、WEEK、DAY、MONTH 和 YEAR 将顺次写入。
不用设置所有的寄存器，仅部分寄存器可被写入。

9.4.3 设置实时计数器报警

当 WALE = 0 时，设置报警时间。

图 9-21. 报警设置流程图



- 备注
1. 可以以任意顺序写入 ALARMWM、ALARMWH 和 ALARMWW。
 2. 固定周期中断和报警匹配中断使用相同的中断源(INTRTC)。 当同时使用这两种类型中断时，在 INTRTC 产生的基础上，通过检查固定周期中断状态标志(RIFG)和报警检测状态标志(WAFG)，来判断产生了哪种中断。

第 10 章 看门狗定时器

10.1 看门狗定时器的功能

看门狗定时器使用内部低速振荡时钟。

看门狗定时器用于检测不希望出现的程序循环。如果检测到一个程序循环，将产生一个内部复位信号。出现以下情况时检测程序循环。

- 如果看门狗定时器计数器溢出
- 如果对看门狗定时器使能寄存器 (WDTE) 执行位操作指令时。
- 如果“ACH”之外的数据写入 WDTE
- 如果在窗口关闭期间将数据写入 WDTE

当由看门狗定时器产生复位时，复位控制标志寄存器 (RESF) 的第 4 位 (WDRF) 置 1。RESF 的详细信息，可以参阅第 17 章 复位功能。

当达到溢出时间的 75% 时，可产生一个间隔中断。

10.2 看门狗定时器的配置

看门狗定时器包括如下硬件。

表 10-1. 看门狗定时器的配置

项目	摄柯
控制寄存器	看门狗定时器使能寄存器 (WDTE)

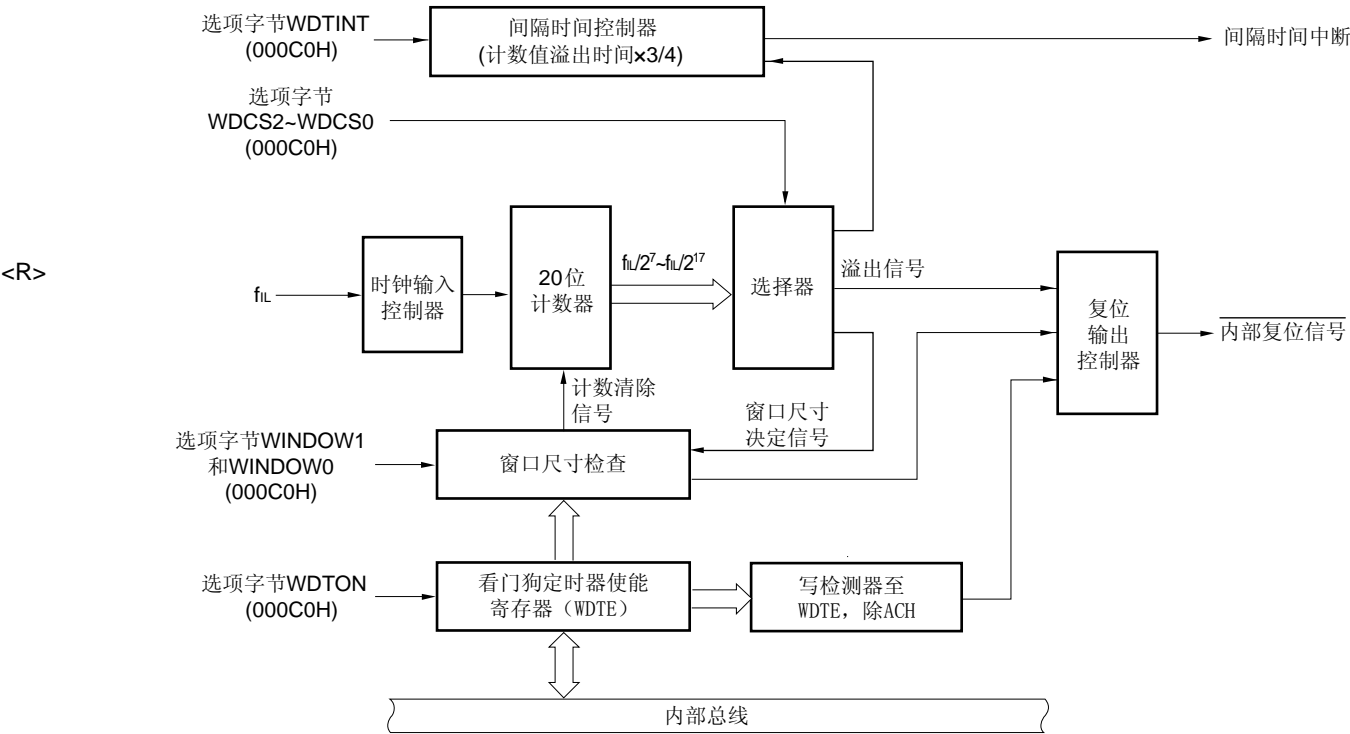
由选项字节设置如何控制计数器的操作、溢出时间和窗口打开周期。

表 10-2. 选项字节和看门狗定时器的设置

看门狗定时器的设置	选项字节 (000C0H)
看门狗定时器间隔中断	第 7 位(WDTINT)
窗口打开周期	第 6 和 5 位 (WINDOW1, WINDOW0)
控制看门狗定时器的计数器操作	第 4 位(WDTON)
看门狗定时器的溢出时间	第 3 ~1 位(WDCS2~WDCS0)
控制看门狗定时器的计数器操作(在 HALT/STOP 模式)	第 0 位 (WDSTBYON)

备注 关于选项字节，见第 21 章 选项字节。

图 10-1. 看门狗定时器的框图



10.3 控制看门狗定时器的寄存器

看门狗定时器由看门狗定时器使能寄存器(WDTE)控制。

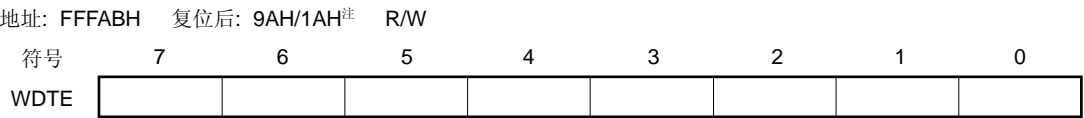
(1) 看门狗定时器使能寄存器 (WDTE)

将“ACH”写入 WDTE 可以对看门狗定时器计数器清零并再次开始计数操作。

可以由 8 位存储器操作指令设置该寄存器。

复位信号的产生将该寄存器设置为 9AH 或 1AH^注。

图 10-2. 看门狗定时器使能寄存器 (WDTE)的格式



注 WDTE 的复位值随选项字节 (000C0H) 的 WDTON 位的设置值而变化。如果要对看门狗定时器进行操作，设置 WDTON = 1。

WDTON 设置值	WDTE 复位值
0（禁止看门狗定时器计数的操作）	1AH
1（使能看门狗定时器计数的操作）	9AH

- 注意事项
1. 如果写入 WDTE 的值不是“ACH”，将产生一个内部复位信号。
 2. 如果使用 1 位存储器操作指令对 WDTE 进行操作，将产生一个内部复位信号。
 3. 从 WDTE 读取的值为 9AH/1AH(与写入值(ACH)不同)。

10.4 看门狗定时器的操作

10.4.1 看门狗定时器的控制操作

1. 当使用看门狗定时器时，由选项字节（000C0H）指定其操作。
 - 通过设置选项字节（000C0H）的第 4 位（WDTON）为 1，允许看门狗定时器的计数操作（复位释放后，计数器开始计数）（如需了解详细信息，参见第 19 章）。

WDTON	看门狗定时器计数器
0	计数器操作禁止（复位后计数停止）
1	计数器操作允许（复位后计数开始）

- 通过使用选项字节（000C0H）的第 3 位～第 1 位（WDCS2～WDCS0），设置溢出时间（如需了解详细信息，参见 10.4.2 和第 21 章）。
 - 通过使用选项字节（000C0H）的第 6 位和第 5 位（WINDOW1 和 WINDOW0），设置窗口打开周期（如需了解详细信息，参见 10.4.3 和第 21 章）。
2. 复位释放后，看门狗定时器开始计数。
 3. 在看门狗计时器开始计数之后和由选项字节设置的溢出时间之前写“ACH”至 WDTE，看门狗定时器清零且重新开始计数。
 4. 在上述步骤后对 WDTE 进行第二次写操作，或在复位释放后的窗口打开期间对 WDTE 进行第二次写操作。如果在窗口关闭期间对 WDTE 进行写操作，则会产生内部复位信号。
 5. 如果超出溢出时间而未写入“ACH”至 WDTE 的话，将产生一个内部复位信号。
出现以下情况时会产生内部复位信号。
 - 如果对看门狗定时器使能寄存器（WDTE）执行位操作指令时。
 - 如果“ACH”之外的数据写入 WDTE

- 注意事项**
1. 复位释放后数据第一次写入 WDTE，看门狗定时器将在任何时序下被清零，与窗口打开时间无关。只要在溢出时间之前，窗口打开时间与寄存器写入时间一样长。看门狗定时器开始再次计数。
 2. 如果通过写入“ACH”至 WDTE 对看门狗定时器清零的话，实际溢出时间可能会与选项字节设置溢出时间差 $2/f_{IL}$ 秒。
 3. 在计数值溢出之前，可将看门狗定时器立即清零。

<示例> 当溢出时间设置为 $2^{10}/f_{IL}$ 时，写入“ACH”的计数值可达 3FH。

注意事项 4. 在 **HALT** 和 **STOP** 模式下看门狗定时器的操作，依据选项字节 (000C0H) 的第 0 位 (WDSTBYON) 的设置值而不同，如下所示。

	WDSTBYON = 0	WDSTBYON = 1
HALT 模式	看门狗定时器操作停止	看门狗定时器操作继续
STOP 模式		

如果 **WDSTBYON = 0**，则在 **HALT** 或 **STOP** 模式释放后，看门狗定时器恢复计数。此时，计数器清零，开始计数。

在 **STOP** 模式释放后当操作时钟为 **X1** 振荡时钟时，经过振荡稳定时间后，**CPU** 开始工作。

因此，如果从 **STOP** 模式释放到看门狗定时器溢出之间的周期比较短，在振荡稳定期间发生一个溢出，引起复位。

从而，当操作时钟为 **X1** 振荡时钟，并且看门狗定时器在由间隔中断引起的 **STOP** 模式释放后被清零时，设置溢出时间要考虑振荡稳定时间。

5. 在 **Flash** 存储器自编程和 **EEPROM™** 仿真期间，看门狗定时器继续运行。处理期间，中断响应时间被延迟。设置溢出时间和窗口大小时应考虑延迟。

10.4.2 看门狗定时器溢出时间的设置

通过使用选项字节(000C0H)的第 3 位～第 1 位(WDCS2～WDCS0)，设置看门狗定时器的溢出时间。

如果出现溢出，将产生内部复位信号。清除当前计数，看门狗定时器通过在溢出时间之前的窗口打开周期器件写入“ACH”至 **WDTE** 开始重新计数。

溢出时间设置如下所示。

<R>

表 10-3. 看门狗定时器溢出时间的设置

WDCS2	WDCS1	WDCS0	看门狗定时器的溢出时间 (f _{IL} = 33 kHz (最大.))
0	0	0	2 ⁷ /f _{IL} (3.88 ms)
0	0	1	2 ⁸ /f _{IL} (7.76 ms)
0	1	0	2 ⁹ /f _{IL} (15.52 ms)
0	1	1	2 ¹⁰ /f _{IL} (31.03 ms)
1	0	0	2 ¹² /f _{IL} (124.12 ms)
1	0	1	2 ¹⁴ /f _{IL} (496.48 ms)
1	1	0	2 ¹⁵ /f _{IL} (992.97 ms)
1	1	1	2 ¹⁷ /f _{IL} (3971.88 ms)

注意事项 看门狗定时器在 **Flash** 存储器自编程和 **EEPROM** 仿真期间继续操作。处理期间，中断响应时间被延迟。设置溢出时间和窗口大小时应考虑延迟。

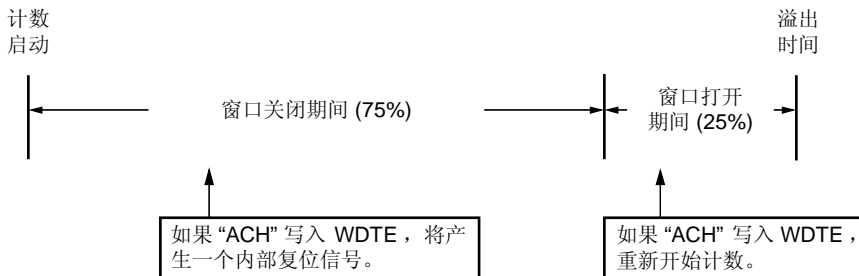
备注 f_{IL}: 内部低速振荡时钟频率

10.4.3 看门狗定时器窗口打开周期的设置

通过使用选项字节(000C0H)的第 6 位和第 5 位(WINDOW1, WINDOW0), 设置看门狗定时器的窗口打开周期。窗口使用情况描述如下。

- 如果在窗口打开周期内写入“ACH”至 WDTE, 看门狗定时器清零且重新开始计数。
- 如果在窗口关闭周期内写入“ACH”至 WDTE, 将检测到异常并产生内部复位信号。

示例: 如果窗口打开周期变为 25%



注意事项 复位释放后数据第一次写入 WDTE, 看门狗定时器将在任何时序下被清零, 与窗口打开时间无关。只要在溢出时间之前, 窗口打开时间与寄存器写入时间一样长。看门狗定时器开始再次计数。

窗口打开周期设置如下所示。

表 10-4. 看门狗定时器窗口打开周期的设置

WINDOW1	WINDOW0	看门狗定时器的窗口打开周期
0	0	25%
0	1	50%
1	0	75%
1	1	100%

- 注意事项**
1. 看门狗定时器在 Flash 存储器自编程和 EEPROM 仿真期间继续操作。处理期间, 中断响应时间被延迟。设置溢出时间和窗口大小时应考虑延迟。
 2. 当选项字节(000C0H)的第 0 位(WDSTBYON) = 0 时, 无论 WINDOW1 和 WINDOW0 为何值, 窗口打开周期为 100%。
 3. 如果看门狗定时器符合下面任一条件, 不要设置窗口打开周期为 25%。
 - 当通过 STOP 模式或软件停止所有主系统时钟 (内部高速振荡时钟、X1 时钟、和外部主系统时钟) 时。

备注 如果溢出时间设置为 $2^{10}/f_{IL}$ ，窗口关闭时间和打开时间如下所示。

	窗口打开周期的设置			
	25%	50%	75%	100%
窗口关闭时间	0~28.44 ms	0~18.96 ms	0~9.48 ms	无
窗口打开时间	28.44~31.03 ms	18.96~31.03 ms	9.48~31.03 ms	0~31.03 ms

<当窗口打开周期为 25%>

- 溢出时间:
 $2^{10}/f_{IL}$ (最大值) = $2^{10}/33 \text{ kHz}$ (最大值) = 31.03 ms
- 窗口关闭时间:
 $0 \sim 2^{10}/f_{IL}$ (最小值) $\times (1 - 0.25) = 0 \sim 2^{10}/27 \text{ kHz}$ (最小值) $\times 0.75 = 0 \sim 28.44 \text{ ms}$
- 窗口打开时间:
 $2^{10}/f_{IL}$ (最小值) $\times (1 - 0.25) \sim 2^{10}/f_{IL}$ (最大值) = $2^{10}/27 \text{ kHz}$ (最小值) $\times 0.75 \sim 2^{10}/33 \text{ kHz}$ (最大值)
 = 28.44~31.03 ms

10.4.4 看门狗定时器间隔中断的设置

根据选项字节 (000C0H) 的第 7 位(WDTINT)的设置，当达到溢出时间的 75%时可产生一个间隔中断 (INTWDTI)。

表 10-5. 看门狗定时器间隔中断的设置

WDTINT	看门狗定时器间隔中断的使用
0	使用间隔中断。
1	达到溢出时间的 75% 时产生间隔中断。

注意事项 在 **STOP** 模式释放后当操作时钟为 **X1** 振荡时钟时，经过振荡稳定时间后，**CPU** 开始工作。

因此，如果从 **STOP** 模式释放到看门狗定时器溢出之间的周期比较短，在振荡稳定期间发生一个溢出，引起复位。

从而，当操作时钟为 **X1** 振荡时钟，并且看门狗定时器在由间隔中断引起的 **STOP** 模式释放后被清零时，设置溢出时间要考虑振荡稳定时间。

备注 即使 INTWDTI 产生后看门狗定时器继续计数(直到 ACH 写入到 WDTE 寄存器)。如果在溢出时间前 ACH 未写入到 WDTE 寄存器，产生一个内部复位信号。

第 11 章 A/D转换器

11.1 A/D转换器的功能

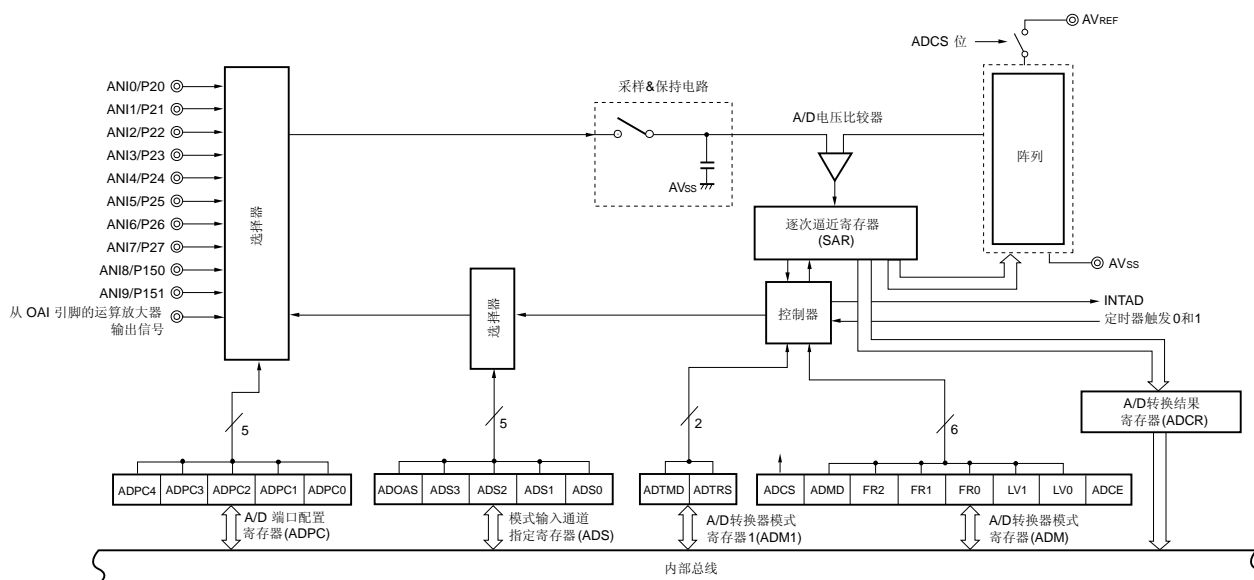
A/D 转换器具有 10 位分辨率，它能把模拟输入信号转换为数字信号，并配置为控制总的 12 通道模拟输入包括 11 通道 A/D 转换器模拟输入（ANI0~ANI9）和一个内部可编程增益放大器输出信号（PGAO）。

A/D 转换器有如下功能。

• 10 位分辨率 A/D 转换

从模拟输入 ANI0~ANI9 中选择一个通道，重复执行 10 位分辨率 A/D 转换。每次 A/D 转换结束，都产生一个中断请求（INTAD）。

图 11-1. A/D 转换器的框图



11.2 A/D转换器的配置

A/D 转换器包括以下硬件。

(1) ANI0~ANI9 引脚

这些是 A/D 转换器的模拟输入引脚。44 引脚产品有 10 通道(ANI0~ANI9)作为模拟输入引脚。用于输入模拟信号，以便转换成数字信号。除了那些被选为模拟输入的引脚外，其它引脚均可用作 I/O 端口引脚。

<R> (2) PGA0

这是 PGAI 引脚的可编程增益放大器输出信号。通过选择可编程增益放大器的输出信号作为模拟输入，A/D 转换器可进行 A/D 转换。

(3) 采样与保持电路

采样保持电路采样每个从输入电路发出的模拟输入电压，并发送到 A/D 电压比较器。此电路也在 A/D 转换期间保持已采样的模拟输入电压。

(4) A/D 电压比较器

A/D 电压比较器把从阵列中产生的电压分流和模拟输入电压进行比较。如果比较结果是模拟输入电压比参考电压 ($1/2 AV_{REF}$)大的话，则逐次逼近寄存器(SAR)的最高有效位(MSB)被置 1。如果模拟输入电压比参考电压 ($1/2 AV_{REF}$)小的话，SAR 的 MSB 被复位。

在此之后，SAR 寄存器的第 10 位自动置 1，开始下一个比较。对于已被设定的结果，阵列的电压分流由第 11 位的值进行选择。

第 11 位 = 0: ($1/4 AV_{REF}$)

第 11 位 = 1: ($3/4 AV_{REF}$)

比较阵列电压分流和模拟输入电压并根据比较结果对 SAR 寄存器的第 10 位进行操作。

模拟输入电压 \geq 阵列电压分流: 第 10 位 = 1

模拟输入电压 \leq 阵列电压分流: 第 10 位 = 0

比较继续直到 SAR 寄存器的位 0。

(5) 阵列

该阵列产生和模拟输入引脚输入电压进行比较的比较电压。

(6) 逐次逼近寄存器 (SAR)

SAR 寄存器是 12 位寄存器，设置那些来自阵列、与模拟输入引脚的电压值相一致的电压分流数据，一次一位并从最高有效位(MSB)开始。

当电压被转换成数字值,并写入最低有效位(LSB)时(A/D 转换结), SAR 寄存器(转换结果)的内容将被保留在 A/D 转换结果寄存器(ADCR)中。当所有指定 A/D 转换操作完成后,产生 A/D 转换结束中断请求信号(INTAD)。

(7) 10 位 A/D 转换结果寄存器 (ADCR)

每次 A/D 转换结束,从逐次逼近寄存器中载入 A/D 转换结果,且 ADCR 寄存器在其高 10 位中保留 A/D 转换结果(低 6 位放置为 0)。

(8) 8 位 A/D 转换结果寄存器 (ADCRH)

每次 A/D 转换结束,从逐次逼近寄存器中载入 A/D 转换结果,且 ADCR 寄存器存储 A/D 转换结果的高 8 位。

(9) 控制器

该电路用于控制模拟输入信号被转换成数字信号的转换时间,并启动和停止转换操作。当 A/D 转换结束时,该控制器产生中断 INTAD。

(10) AV_{REF} 引脚

该引脚输入 A/D 转换器、可编程增益放大器、比较器和 A/D 转换器的电源引脚、以及比较器的参考电压。当端口 2、端口 8 和端口 15 的所有引脚被用作模拟端口引脚时,要使 AV_{REF} 电压在 $2.7\text{ V} \leq \text{AV}_{\text{REF}} \leq \text{V}_{\text{DD}}$ 。当端口 2、8 和端口 15 的一个或多个引脚被用作数字端口引脚时,使得 AV_{REF} 和 V_{DD} 同电势。

根据 AV_{REF} 和 AV_{SS} 之间的电压,将输入到 ANI0~ANI9 的模拟信号转换成数字信号。

(11) AV_{SS} 引脚

这是 A/D 转换器的信号地引脚。即使不使用 A/D 转换器时,也保持此引脚的电压与 V_{SS} 引脚一致。

11.3 A/D转换器使用的寄存器

A/D 转换器使用下面八个寄存器。

- 外围设备使能寄存器 0 (PER0)
- A/D 转换器模式寄存器 (ADM)
- A/D 转换器模式寄存器 1(ADM1)
- A/D 端口配置 寄存器 (ADPC)
- 模式输入通道指定寄存器(ADS)
- 端口模式寄存器 2、8、15 (PM2、PM8、PM15)
- 10 位 A/D 转换结果寄存器 (ADCR)
- 8 位 A/D 转换结果寄存器 (ADCRH)

(1) 外围设备使能寄存器 0 (PER0)

PER0 用于允许或者禁止使用每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少消耗和噪音。
使用 A/D 转换器时，确保设置此寄存器的第 5 位 (ADCEN)为 1。
可以通过 1 位或者 8 位存储器操作指令来设置 PER0。
复位后这个寄存器的值为 00H。

图 11-2. 外围设备使能寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	6	<5>	<4>	3	<2>	1	0
PER0	RTCEN	0	ADCEN	0	0	SAU0EN	0	0

ADCEN	A/D 转换器输入时钟的控制
0	停止输入时钟的供给 <ul style="list-style-type: none">• 被 A/D 转换器使用的 SFR 是不可重写。• A/D 转换器在复位状态。
1	供给输入时钟 <ul style="list-style-type: none">• 被 A/D 转换器使用的 SFR 是可读/写的。

- 注意事项 1. 设置 A/D 转换器时，首先务必设置 ADCEN 为 1。如果 ADCEN = 0，写入到 A/D 转换器的控制寄存器被忽略，即使读取寄存器，只能读到默认值。
2. 确保将 PER0 寄存器的第 0、1、3、4 和 6 位清零。

(2) A/D 转换器模式寄存器 (ADM)

该寄存器设置模拟输入的 A/D 转换时间，并启动/停止转换。

ADM 可由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 11-3. A/D 转换器模式寄存器 (ADM) 的格式

地址: FFF30H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	ADMD	FR2 ^{注1}	FR1 ^{注1}	FR0 ^{注1}	LV1 ^{注1}	LV0 ^{注1}	ADCE

ADCS	A/D 转换操作控制
0	停止转换操作
1	允许转换操作

ADMD	A/D 转换操作模式指定
0	选择模式
1	扫描模式

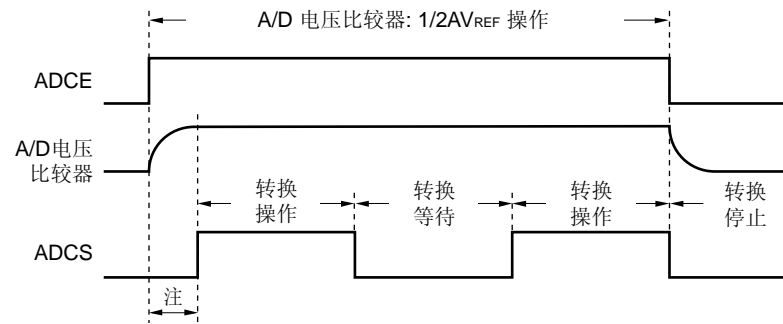
ADCE	A/D 电压比较器操作控制 ^{注2}
0	停止 A/D 电压比较器操作
1	允许 A/D 电压比较器操作 (A/D 电压比较器: 1/2 AV _{REF} 操作)

- 注
1. FR2~FR0、LV1、LV0 和 A/D 转换的详细情况，参见表 11-2 A/D 转换时间选择。
 2. A/D 电压比较器的运行由 ADCS 和 ADCE 控制，并且从运行开始到运行稳定的时间为 1 μ s。因此，当 ADCE 被设置为 1 后，至少经过 1 μ s 的时间，再将 ADCS 置 1 时，此时的转换结果优先于第一次的转换结果。否则，忽略第一次转换的数据。

表 11-1. ADCS 和 ADCE 的设置

ADCS	ADCE	A/D 转换操作
0	0	停止状态(不存在直流功耗)
0	1	转换等待模式(A/D 电压比较器: 1/2AV _{REF} 操作，只有比较器产生功耗)
1	0	设置禁止
1	1	转换模式(A/D 电压比较器: 1/2AV _{REF} 操作)

图 11-4. 使用 A/D 电压比较器时的时序图



注 从 ADCE 位上升到 ADCS 位下降所需时间必须至少为 1 μ s，以便稳定内部电路。

注意事项 在以不同的数据重写 FR0~FR2 位、LV1 和 LV0 位之前，A/D 转换必须停止。

表 11-2. A/D 转换时间选择

(1) $4.0V \leq AV_{REF} \leq 5.5V$

A/D 转换器模式寄存器 (ADM)					模式	转换时间选择				转换时钟 (f _{AD})
FR2	FR1	FR0	LV1	LV0		f _{CLK} = 2 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz	
0	0	0	0	0	标准	设置禁止	设置禁止	34.2 μs	17.1 μs	f _{CLK} /20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f _{CLK} /10
0	1	0					27.6 μs	13.8 μs	6.9 μs	f _{CLK} /8
0	1	1				52.0 μs	20.8 μs	10.4 μs	5.2 μs	f _{CLK} /6
1	0	0				35.0 μs	14.0 μs	7.0 μs	设置禁止	f _{CLK} /4
1	0	1				26.5 μs	10.6 μs	5.3 μs		f _{CLK} /3
1	1	0				18.0 μs	7.2 μs	设置禁止		f _{CLK} /2
1	1	1				9.5 μs	设置禁止		f _{CLK}	
×	×	×	0	1	增压	设置禁止				—
0	0	0	1	0	高速 1	设置禁止	64.4 μs	32.2 μs	16.1 μs	f _{CLK} /20
0	0	1					32.4 μs	16.2 μs	8.1 μs	f _{CLK} /10
0	1	0				65.0 μs	26.0 μs	13.0 μs	6.5 μs	f _{CLK} /8
0	1	1				49.0 μs	19.6 μs	9.8 μs	4.9 μs	f _{CLK} /6
1	0	0				33.0 μs	13.2 μs	6.6 μs	3.3 μs	f _{CLK} /4
1	0	1				25.0 μs	10.0 μs	5.0 μs	2.5 μs	f _{CLK} /3
1	1	0				17.0 μs	6.8 μs	3.4 μs	设置禁止	f _{CLK} /2
1	1	1				9.0 μs	3.6 μs	设置禁止		f _{CLK}
0	0	0	1	1	高速 2	设置禁止	设置禁止	34.2 μs	17.1 μs	f _{CLK} /20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f _{CLK} /10
0	1	0					27.6 μs	13.8 μs	6.9 μs	f _{CLK} /8
0	1	1				52.0 μs	20.8 μs	10.4 μs	5.2 μs	f _{CLK} /6
1	0	0				35.0 μs	14.0 μs	7.0 μs	3.5 μs	f _{CLK} /4
1	0	1				26.5 μs	10.6 μs	5.3 μs	设置禁止	f _{CLK} /3
1	1	0				18.0 μs	7.2 μs	3.6 μs		f _{CLK} /2
1	1	1				9.5 μs	3.8 μs	设置禁止		f _{CLK}

注意事项 1. 在以不同的数据重写 FR0~FR2 位、LV1 和 LV0 位之前，A/D 转换必须停止(ADCS = 0)。

2. 上述的转换时间不包括时钟频率误差。在选择转换时间时，要将时钟频率误差考虑进去。

备注 f_{CLK}: CPU/外围硬件时钟频率

(2) $2.7V \leq AV_{REF} \leq 5.5 V$

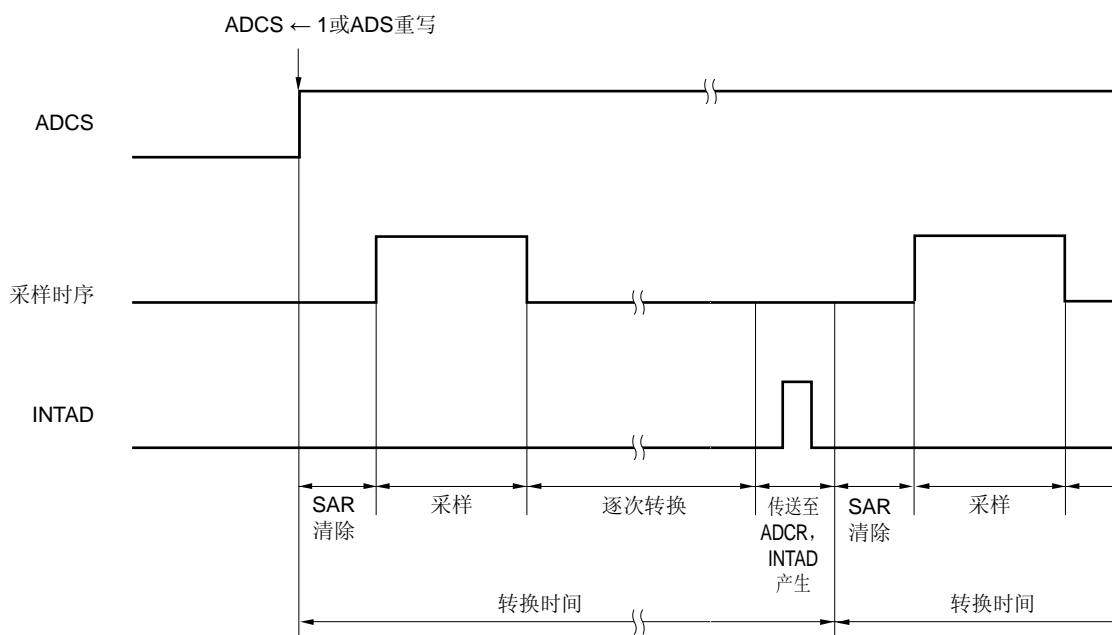
A/D 转换器模式寄存器 (ADM)					模式	转换时间选择				转换时钟 (f _{AD})
FR2	FR1	FR0	LV1	LV0		f _{CLK} = 2 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz	
0	0	0	0	0	标准	设置禁止	设置禁止	34.2 μs	17.1 μs	f _{CLK} /20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f _{CLK} /10
0	1	0					27.6 μs	13.8 μs	设置禁止	f _{CLK} /8
0	1	1					52.0 μs	20.8 μs		10.4 μs
1	0	0				35.0 μs	14.0 μs	设置禁止		f _{CLK} /4
1	0	1				26.5 μs	10.6 μs			f _{CLK} /3
1	1	0				18.0 μs	设置禁止		f _{CLK} /2	
1	1	1				9.5 μs			f _{CLK}	
×	×	×	0	1	增压	设置禁止				—
×	×	×	1	0	高速 1	设置禁止				—
0	0	0	1	1	高速 2	设置禁止	设置禁止	34.2 μs	17.1 μs	f _{CLK} /20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f _{CLK} /10
0	1	0					27.6 μs	13.8 μs	6.9 μs	f _{CLK} /8
0	1	1					52.0 μs	20.8 μs	10.4 μs	5.2 μs
1	0	0				35.0 μs	14.0 μs	7.0 μs	3.5 μs	f _{CLK} /4
1	0	1				26.5 μs	10.6 μs	5.3 μs	设置禁止	f _{CLK} /3
1	1	0				18.0 μs	7.2 μs	3.6 μs		f _{CLK} /2
1	1	1				9.5 μs	3.8 μs	设置禁止		f _{CLK}

注意事项 1. 在以不同的数据重写 FR0~FR2 位、LV1 和 LV0 位之前，A/D 转换必须停止(ADCS = 0)。

2. 上述的转换时间不包括时钟频率误差。在选择转换时间时，要将时钟频率误差考虑进去。

备注 f_{CLK} : CPU/外围硬件时钟频率

图 11-5. A/D 转换器采样和 A/D 转换时序



(3) A/D 转换器模式寄存器 1(ADM1O)

该寄存器设置 A/D 转换的开始触发。
ADM1 可以通过 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 00H。

图 11-6. A/D 转换器模式寄存器 1(ADM1)的格式

地址: FFF42H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	<0>
ADM1	ADTMD	0	0	0	0	0	0	ADTRS

ADTMD	A/D 触发器模式选择
0	软件触发模式
1	定时器触发模式 (硬件触发模式)

ADTRS	定时器触发信号选择
0	定时器触发信号 0
1	定时器触发信号 1

注意事项 A/D 转换期间，ADM1 不可写。转换操作停止(ADCS = 0)时，可写。

备注 关于定时器触发信号的详情，参考 7.4.8 作为 A/D 转换触发输出功能的操作(类型 1)和 7.4.9 作为 A/D 转换触发输出功能的操作(类型 2)。

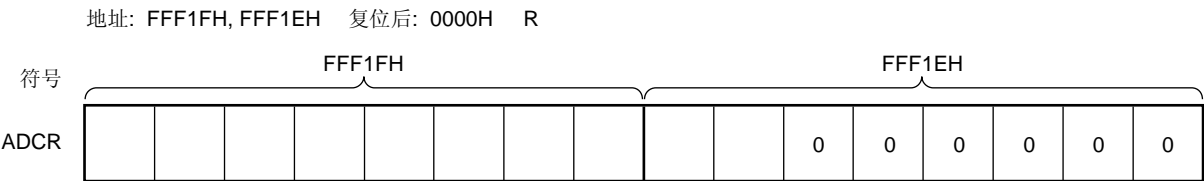
(4) 12 位 A/D 转换结果寄存器 (ADCR)

这是一个 16 位寄存器，在选择模式下用来存储 A/D 转换结果。寄存器的低 6 位恒为 0。每次 A/D 转换结束，从逐次逼近寄存器中载入转换结果。转换结果的高 8 位存储在 FFF1FH 中，而转换结果的低 2 位存储在 FFF1EH 的高 2 位中。

ADCR 可由 16 位存储器操作指令读取。

复位后这个寄存器的值为 0000H。

图 11-7. 10 位 A/D 转换结果寄存器(ADCR)的格式



注意事项 在对 A/D 转换模式寄存器(ADM)、模拟输入通道选择寄存器(ADS)和 A/D 端口配置寄存器(ADPC)赋值时，ADCR 的内容可能是不确定的。在对 ADM、ADS 和 ADPC 进行写操作前且转换结束后读取转换结果。如果没有使用上述时序，可能会读取到不正确的转换结果。

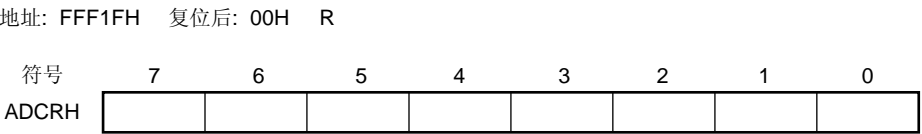
(5) 8 位 A/D 转换结果寄存器 (ADCRH)

这是一个 8 位寄存器，用来存储 A/D 转换结果。可以存储 10 位分辨率的高 8 位。

ADCRH 可由 8 位存储器操作指令读取。

复位后这个寄存器的值为 00H。

图 11-8. 8 位 A/D 转换结果寄存器(ADCRH)的格式



注意事项 在对 A/D 转换模式寄存器(ADM)、模拟输入通道选择寄存器(ADS)和 A/D 端口配置寄存器(ADPC)赋值时，ADCRH 的内容可能是不确定的。在对 ADM、ADS 和 ADPC 进行写操作前且转换结束后读取转换结果。如果没有使用上述时序，可能会读取到不正确的转换结果。

(6) 模式输入通道指定寄存器(ADS)

该寄存器用来选择被 A/D 转换的模拟电压的输入通道。

ADS 可由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 11-9. 模拟输入通道指定寄存器 (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	ADOAS	0	0	ADS3	ADS2	ADS1	ADS0

○ 选择模式 (ADMD = 0)

ADOAS	ADS3	ADS2	ADS1	ADS0	模拟输入通道	输入源
0	0	0	0	0	ANI0	P20/ANI0 引脚
0	0	0	0	1	ANI1	P21/ANI1 引脚
0	0	0	1	0	ANI2	P22/ANI2 引脚
0	0	0	1	1	ANI3	P23/ANI3 引脚
0	0	1	0	0	ANI4	P24/ANI4 引脚
0	0	1	0	1	ANI5	P25/ANI5 引脚
0	0	1	1	0	ANI6	P26/ANI6 引脚
0	0	1	1	1	ANI7	P27/ANI7 引脚
0	1	0	0	0	ANI8	P150/ANI8 引脚
0	1	0	0	1	ANI9	P151/ANI9 引脚
1	×	×	×	×	PGAO	可编程增益放大器输出信号
与上述不同的值					设置禁止	

(注和注意事项列在下页中。)

○ 选择模式 (ADMD = 1)

ADOAS	ADS3	ADS2	ADS1	ADS0	模拟输入通道			
					扫描 0	扫描 1	扫描 2	扫描 3
0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	0	1	1	0	ANI6	ANI7	ANI8	ANI9
1	0	0	0	0	PGAO	ANI0	ANI1	ANI2
1	0	0	0	1	PGAO	ANI1	ANI2	ANI3
1	0	0	1	0	PGAO	ANI2	ANI3	ANI4
1	0	0	1	1	PGAO	ANI3	ANI4	ANI5
1	0	1	0	0	PGAO	ANI4	ANI5	ANI6
1	0	1	0	1	PGAO	ANI5	ANI6	ANI7
1	0	1	1	0	PGAO	ANI6	ANI7	ANI8
1	0	1	1	1	PGAO	ANI7	ANI8	ANI9
与上述不同的值					ANI0	ANI1	ANI2	ANI3

- 注意事项
1. 确保将位 4、5 和 7 清“0”。
 2. 利用端口模式寄存器 2、8 和 15 (PM2、PM8、PM15)，设置一个输入模式下的用于 A/D 转换的通道。
 3. 不要通过 ADS 设置已经被 ADPC 设置为数字 I/O 的引脚。
 4. 在设置可编程增益放大器的操作之后，从 PGAI 引脚选择可编程增益放大器的输出信号(PGAO)作为模拟输入(参考 11.4.1 A/D 转换器的基本操作)。

备注 ×: 不必考虑

(7) A/D 端口配置寄存器 (ADPC)

这个寄存器用于将 ANI0/P20~ANI7/P27 和 ANI8/P150、ANI19/P151 引脚切换为 A/D 转换器的模拟输入或是数字 I/O 端口。

ADPC 可由 8 位存储器操作指令设置。

复位信号的发生设置此寄存器为 10H。

图 11-10. A/D 端口配置寄存器 (ADPC) 的格式

地址: F0017H 复位后: 10H R/W

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

ADP C4	ADP C3	ADP C2	ADP C1	ADP C0	模拟输入 (A) / 数字 I/O (D) 切换									
					端口 5		端口 2							
					ANI9 /P110	ANI8 /P150	ANI7 /P27	ANI6 /P26	ANI5 /P25	ANI4 /P24	ANI3 /P23	ANI2 /P22	ANI1 /P21	ANI0 /P20
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	A	A	D	D	D	D	D	D
0	0	1	1	1	A	A	A	D	D	D	D	D	D	D
0	1	0	0	0	A	A	D	D	D	D	D	D	D	D
0	1	0	0	1	A	D	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D
与上述不同的值					设置禁止									

- 注意事项**
1. 利用端口模式寄存器 2 和 15 (PM2, PM15)，设置一个输入模式下的用于 A/D 转换的通道。
 2. 不要通过 ADS 设置已经被 ADPC 设置为数字 I/O 的引脚。

(8) 端口输入模式寄存器 8(PIM8)

<R> 此寄存器可允许或禁止在 1 位单元内的端口 8 的数据输入。
通过禁止数字输入，使 PGAI 引脚作为模拟输入。使能数字输入以使用端口功能或外部中断和定时器高阻控制功能，因为初始状态时禁止数字输入(使用 P8n 引脚作为模拟输入)。
PIM8 可以通过 1 位或 8 位存储器操作指令设置。
复位后这个寄存器的值为 00H。

图 11-11. 端口输入模式寄存器 8 (PIM8) 的格式

地址: F0048H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PIM8	0	0	0	0	PIM83	PIM82	PIM81	PIM80

PIM8n	选择允许或禁止 P8n 引脚数字输入(n = 0~3)
0	数字输入禁止(使用 P8n 引脚作为模拟输入)。
1	数字输入允许

(9) 端口模式寄存器 2、8 和 15(PM2、PM8、PM15)

当使用 ANI0/P20~ANI7/P27、ANI8/P150、ANI9/P151 和 PGAI/P80 引脚作为模拟输入端口时，设置 PM20~PM27、PM80 和 PM150~PM151 为 1。此时 P20~P27、P80、和 P150~P151 的输出锁存可为 0 或 1。
如果 PM20~PM27、PM80 和 PM150~PM151 设置为 0，则其不能用作模拟输入端口引脚。
PM2、PM8 和 PM15 可由 1 位或 8 位存储器操作指令设置。
复位信号的产生将这些寄存器设置为 FFH。

注意事项 如果引脚设置为模拟输入端口的话，读取的值不是引脚电平而是始终为“0”。

图 11-12. 端口模式寄存器 2、8 和 15 (PM2、PM8、PM15) 的格式

地址: FFF22H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

地址: FFF28H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM8	1	1	1	1	PM83	PM82	PM81	PM80

地址: FFF2FH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM15	1	1	1	1	1	0	PM151	PM150

PMmn	Pmn 引脚 I/O 模式选择 (mn = 20~27、80~83、150~151)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

注意事项 确认设置 PM15 的位 2 为“1”，以及 P15 的位 2 为“0”。

根据 ADPC、ADS、PM2、PM8 和 PM15 的设置，ANI0/P20~ANI7/P27、PGAI/P80 和 ANI8/P150~ANI9/P151 引脚如下所示。

表 11-3 ANI0/P20~ANI7/P27、PGAI/P80 和 ANI8/P150~ANI9/P151 引脚的设置功能

ADPC	PM2、PM8 和 PM15	ADS	ANI0/P20~ANI7/P27、PGAI/P80 和 ANI8/P150~ANI9/P151 引脚
数字 I/O 选择	输入模式	—	数字输入
	输出模式	—	数字输出
模拟输入选择	输入模式	选择 ANI。	模拟输入 (转换)
		不要选择 ANI。	模拟输入 (不转换)
	输出模式	选择 ANI。	设置禁止
		不要选择 ANI。	

11.4 A/D转换器操作

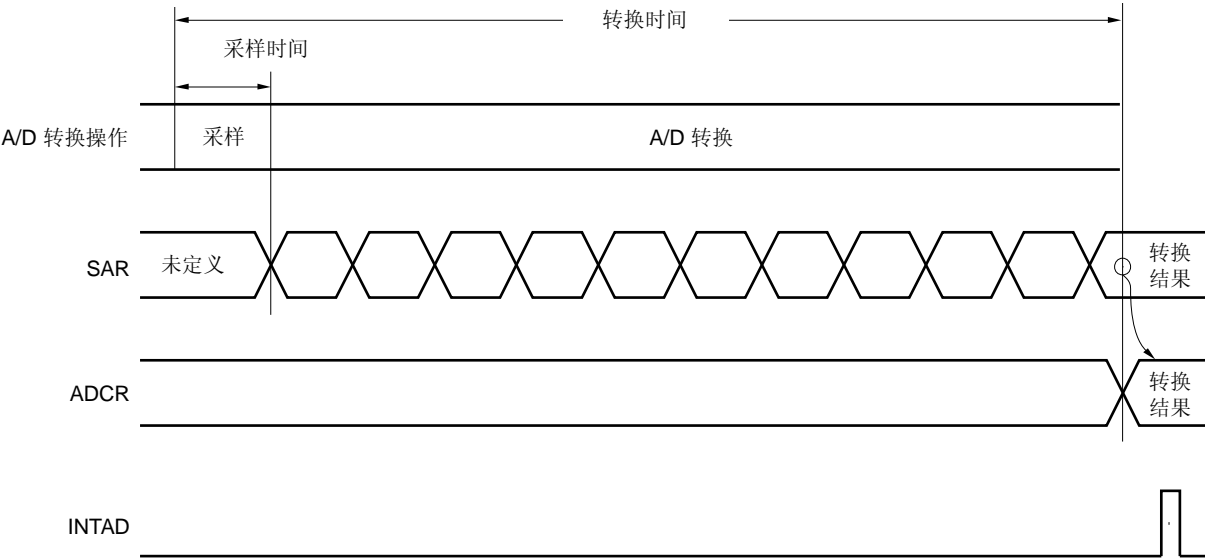
11.4.1 A/D转换器的基本操作

- <1> 把外围设备使能寄存器 0 (PER0) 的第 5 位 (ADCEN) 置 1 来启动输入时钟的供给到 A/D 转换器。
- <2> 通过 ADM 的第 5 位~第 1 位(FR2~FR0、LV1 和 LV0)设置 A/D 转换时间, 以及 ADM 的第 6 位(ADMD)来设置运行模式。
- <3> 把 A/D 转换模式寄存器 (ADM) 的第 0 位(ADCE)置 1, 启动 A/D 电压比较器操作。
- <4> 通过 A/D 端口配置寄存器 (ADPC) 设置 A/D 转换到模拟输入的通道, 并且通过端口模式寄存器(PM2、PM8 和 PM15)设置到输入模式。
- <R> <5> 设置可编程增益放大器操作到设置从 PGAI 引脚(PGAO)的可编程增益放大器输出信号, 作为模拟输入通道(参考 8.4.1 启动比较器和可编程增益放大器的操作)。
- <6> 使用模拟输入通道指定寄存器 (ADS)为 A/D 转换选择一个通道。
- <7> 使用 A/D 转换器模式寄存器 1(ADM1)进行设置触发模式。
- <8> 通过设置 ADM 的第 7 位 (ADCS)为 1 开始转换操作。
如果定时器的触发模式在步骤<7>中被设置时, 定时器进入触发等待状态。
(<9>~<15>操作由硬件执行。)
- <9> 到选择的模拟输入通道的输入电压由采样&保持电路采样。
- <10> 在经过一段时间的采样后, 采样&保持电路处于保持状态, 且在 A/D 转换操作结束前一直保持采样电压。
- <11> 设置逐次逼近寄存器(SAR)的第 9 位。通过分接选择器将串联电阻的分接电压置为 $(1/2)AV_{REF}$ 。
- <12> 由电压比较器比较串联电阻串的分接电压与采样电压。如果模拟输入大于 $(1/2)AV_{REF}$, 则 SAR 的 MSB 保留设置为 1。如果模拟输入小于 $(1/2)AV_{REF}$, 则 MSB 复位为 0。
- <13> 接下来, SAR 的第 8 位自动置 1, 并进入下一个比较过程。根据第 9 位的预置值选择串联电阻串的分接电压, 具体描述如下。
- 位 9 = 1: $(3/4)AV_{REF}$
 - 位 9 = 0: $(1/4)AV_{REF}$
- 比较分接电压与采样电压, 并设置 SAR 的第 8 位, 如下所示。
- 采样电压 \geq 电压分流: 位 8 = 1
 - 采样电压 $<$ 电压分流: 位 8 = 0
- <14> 按此方式继续进行比较, 直至 SAR 的第 0 位。
- <15> 全部 10 位比较完成后, 在 SAR 中保留一个有效的数值结果, 然后将结果传送至 A/D 转换结果寄存器(ADCR, ADCRH)中, 并锁存。
同时也会产生 A/D 转换结束中断请求 (INTAD)。
- <16> 重复步骤<9>~<15>, 直到 ADCS 被清为 0。
将 ADCS 清零, 以停止 A/D 转换器操作。
当 ADCE = 1 时, 若要重新启动 A/D 转换操作, 应从步骤<8>开始。当 ADCE = 0 时, 若要再次启动 A/D 转换操作, 设置 ADCE=1, 等待至少 $1\mu s$ 然后从步骤<8>开始操作。如要改变 A/D 转换的通道, 则从步骤<6>开始。

注意事项 必须确保步骤 <3>~<8>之间的操作时间至少为 $1\mu s$ 。

- 备注 有两种类型的 A/D 转换结果寄存器可以使用。
- ADCR (16 位): 存储 10 位 A/D 转换值
 - ADCRH (8 位): 存储 8 位 A/D 转换值

图 11-13. A/D 转换器的基本操作



A/D 转换操作是连续执行的，直到用软件将 A/D 转换器模式寄存器(ADM)的第 7 位(ADCS)复位(0)。

在 A/D 转换期间，如果对模拟输入通道选择寄存器(ADS)进行写操作，则转换操作被初始化，且若 ADCS 被设置为 (1)，则转换操作重新开始。

复位信号会将 A/D 转换结果寄存器(ADCR, ADCRH)的内容设置为 0000H 或 00H。

11.4.2 输入电压和转换结果

输入到模拟输入引脚(ANIO~ANI9、PGAI)的模拟输入电压与理论上的 A/D 转换结果(存储在 10 位 A/D 转换结果寄存器(ADCR)中)之间的关系表示如下。

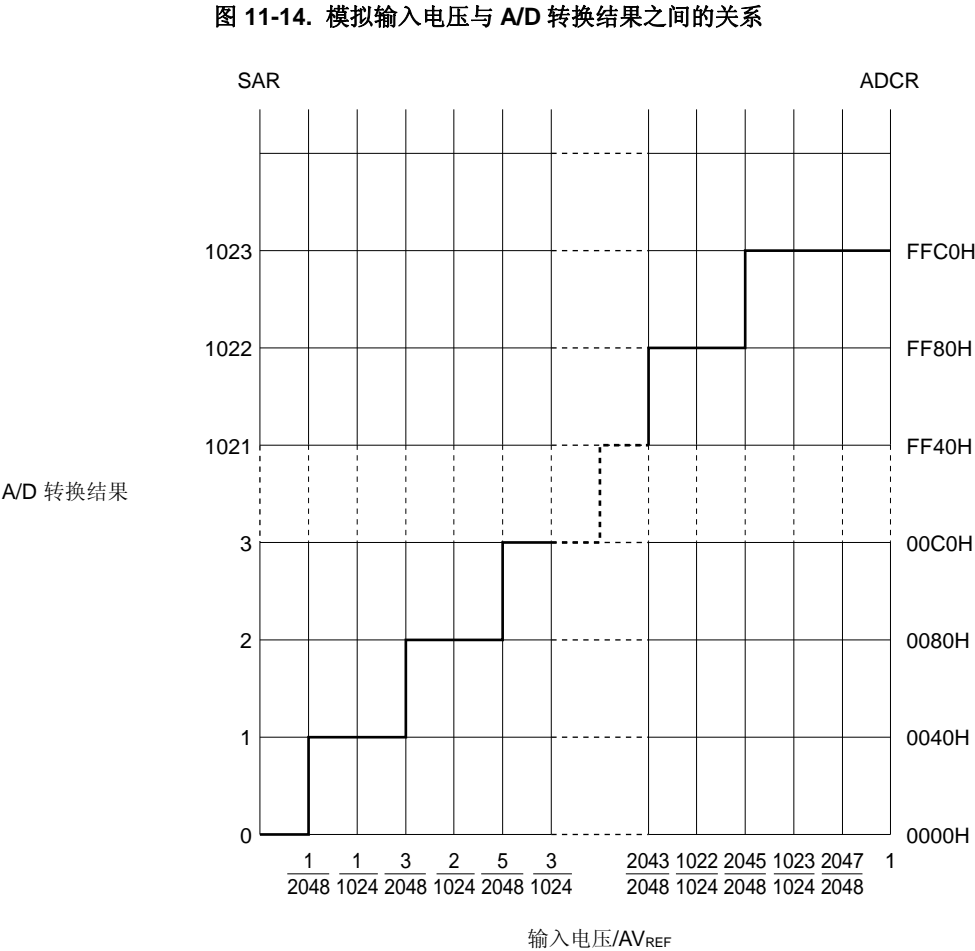
$$\begin{aligned} \text{SAR} &= \text{INT} \left(\frac{V_{\text{AIN}}}{V_{\text{REF}}} \times 1024 + 0.5 \right) \\ \text{ADCR} &= \text{SAR} \times 64 \end{aligned}$$

或

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

- 其中， INT(): 该函数返回括号中值的整数部分。
V_{AIN}: 模拟 输入电压
V_{REF}: V_{REF} 引脚电压
ADCR: A/D 转换结果寄存器(ADCR)的值
SAR: 逐次逼近寄存器

图 11-14 显示模拟输入电压与 A/D 转换结果之间的关系。



11.4.3 触发器模式选择

提供以下两个触发模式来设置 A/D 转换开始时序。这些触发模式由 ADM1 寄存器设置。

- 软件触发模式
- 定时器触发模式 (硬件触发模式)

(1) 软件触发模式

<R>

该模式用于启动模拟输入通道(ANI0~ANI9、PGA0)的 A/D 转换，它是通过设置 ADCS 为 1、由模拟输入通道指定寄存器(ADS)选择的。

在 A/D 转换完成后，只要 ADCS 位未被清零，A/D 转换将重复进行。

在转换中如果 ADM、ADM1 或 ADS 寄存器被写入，A/D 转换将中止。这种情况下，如果是选择模式，A/D 转换会从起始处重新开始；如果是扫描模式，A/D 转换会从扫描 0 重新开始。

(2) 定时器触发模式 (硬件触发模式)

该模式通过设置 ADCS 为 1 和检测定时器触发信号 0 和 1 开始对由模拟输入通道专用寄存器(ADS)选择的模拟输入通道(ANI0~ANI9、PGA0)进行 A/D 转换。

在 A/D 转换完成后，只要 ADCS 位未被清零，A/D 转换将重复进行。

如果在 A/D 转换中产生一个定时器触发信号或者在转换中 ADM、ADM1 或 ADS 寄存器被写入，A/D 转换将中止。这种情况下，如果是选择模式，A/D 转换会从起始处重新开始；如果是扫描模式，A/D 转换会从扫描 0 重新开始。

11.4.4 A/D 转换器操作模式

选择模式和扫描模式被提供作为 A/D 转换器的操作模式。

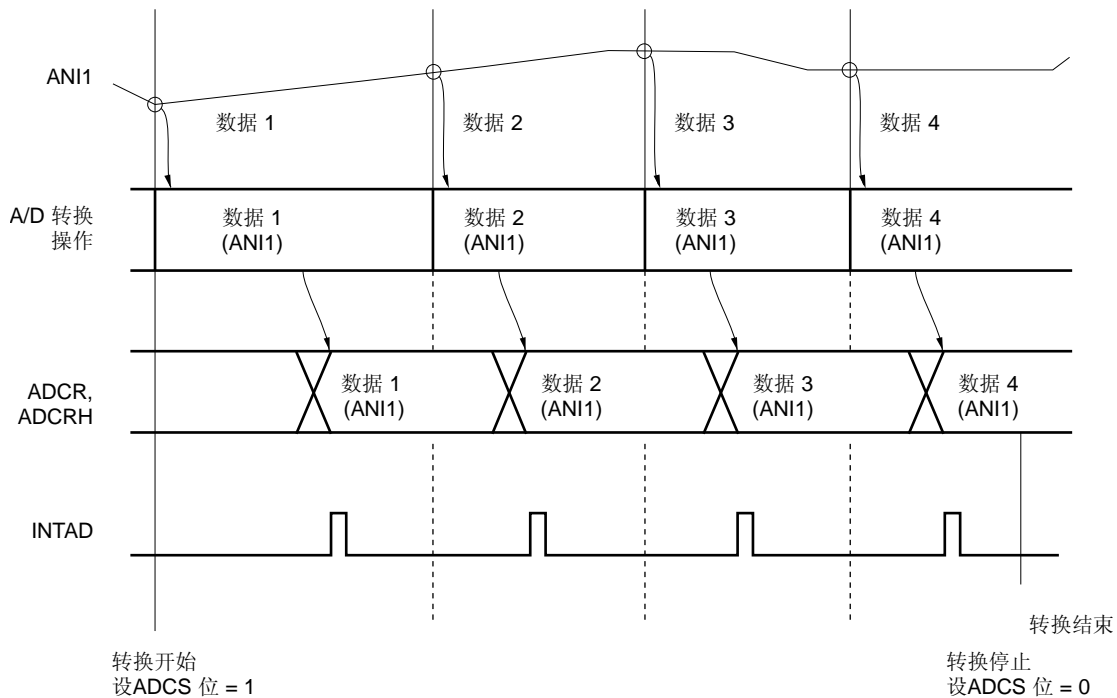
(1) 选择模式

由模拟输入通道选择寄存器(ADS)指定的一个模拟输入，当 A/D 转换器模式寄存器(ADM)的 ADMD 位为 0 时，进行 A/D 转换。

当 A/D 转换完成时，转换结果存于 A/D 转换结果寄存器(ADCR)中，并产生一个 A/D 转换结束中断请求信号(INTAD)。

在转换中如果 ADM、ADM1 或 ADS 寄存器被写入，A/D 转换将中止。这种情况下，无论是软件触发模式或定时器触发模式(硬件触发模式)，A/D 转换都会从起始处重新开始。

图 11-15. 选择模式运行时序的示例



(2) 扫描模式

扫描 0~3 的四个模拟输入通道，由模拟输入通道选择寄存器(ADS)指定。当 A/D 转换器模式寄存器(ADM)的 ADMD 位为 1 时，将先后进行 A/D 转换。A/D 转换按顺序进行，从由扫描 0 指定的模拟输入通道开始。

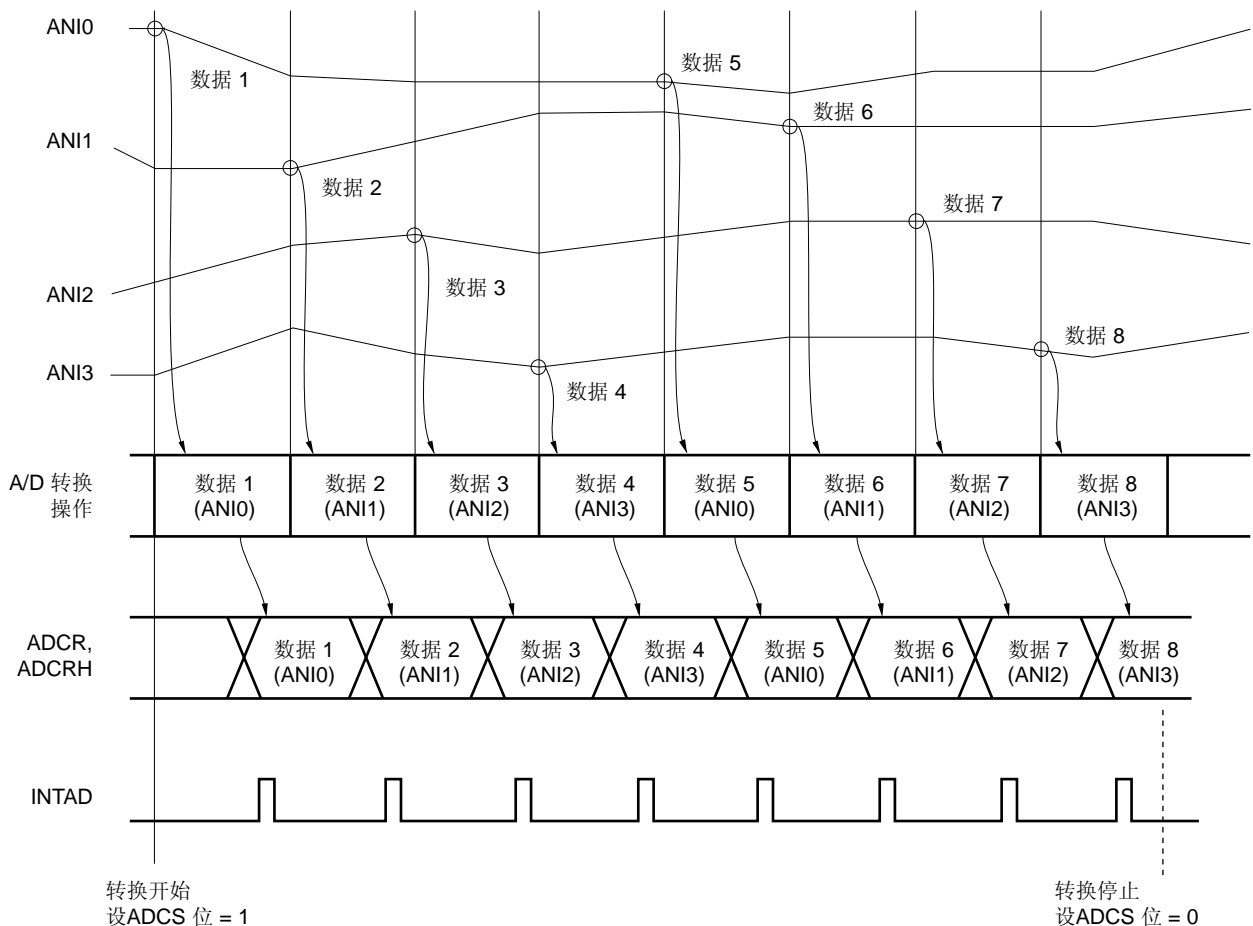
当一个模拟输入的 A/D 转换完成时，转换结果存于 A/D 转换结果寄存器(ADCR)中，并产生一个 A/D 转换结束中断请求信号(INTAD)。

所有模拟输入通道的 A/D 转换结果存放在 ADCR 中。因此建议，一旦一个模拟输入通道的 A/D 转换完成后，把 ADCR 中的内容存放到 RAM 中。

当一个 A/D 转换结束时，无论是否为触发模式，下一个 A/D 转换马上开始。

在转换中如果 ADM、ADM1 或 ADS 寄存器被写入，A/D 转换将中止。这种情况下，无论是软件触发模式或定时器触发模式(硬件触发模式)，A/D 转换都从扫描 0 的模拟输入通道处重新开始。

图 11-16. 扫描模式运行时序示例



设置方法如下。

- <1> 设置外围设备使能寄存器 0 (PER0) 的第 5 位 (ADCEN) 为 1。
- <2> 通过 ADM 的第 5 位~第 1 位 (FR2~FR0、LV1 和 LV0) 选择转换时间, 以及 ADM 的第 6 位 (ADMD) 来选择运行模式。
- <3> 设置 A/D 转换器模式寄存器 (ADM) 的第 0 位 (ADCE) 为 1。
- <4> 利用 A/D 端口配置寄存器 (ADPC) 的第 4~0 位 (ADPC4~ADPC0)、端口模式寄存器 2 (PM2) 的第 7~0 位 (PM27~PM20)、端口模式寄存器 8 (PM8) 的第 0 位 (PM80) 和端口模式寄存器 15 (PM15) 的第 2~0 位 (PM152~PM150), 设置用于模拟输入模式的通道。
- <R> <5> 设置可编程增益放大器操作到设置从 PGAI 引脚 (PGAO) 的可编程增益放大器输出信号, 作为模拟输入通道 (参考 8.4.1 启动比较器和可编程增益放大器的操作)。
- <6> 通过模拟输入通道选择寄存器 (ADS) 的第 6 位和第 3 位~第 0 位 (ADOAS、ADS3~ADS0) 选择通道。
- <7> 通过 A/D 转换器模式寄存器 1 (ADM1) 的第 0 位和第 7 位 (ADTRS、ADTMD) 进行设置触发模式。
- <8> 将 ADM 的第 7 位 (ADCS) 置 1, 启动 A/D 转换。
- <9> 当 A/D 转换结束, 产生一个中断请求信号 (INTAD)。
- <10> 将 A/D 转换数据传送至 A/D 转换结果寄存器 (ADCR, ADCRH) 中。
- <改变通道>
 - <11> 通过 ADS 的第 6 和 3~0 位 (ADOAS、ADS3~ADS0) 来改变通道从而启动 A/D 转换。
 - <12> 当 A/D 转换结束, 产生一个中断请求信号 (INTAD)。
 - <13> 将 A/D 转换数据传送至 A/D 转换结果寄存器 (ADCR, ADCRH) 中。
- <完成 A/D 转换>
 - <14> ADCS 清为 0。
 - <15> ADCE 清为 0。
 - <16> 将外围设备使能寄存器 0 (PER0) 的第 5 位 (ADCEN) 清 0。

- 注意事项**
1. 必须确保步骤 <3>~<8> 之间的操作时间至少为 1 μ s。
 2. <3> 可以在<4>和<6>之间进行。
 3. <3> 可以被省略。但在这种情况下, 忽略<8>之后的第一次转换结果。
 4. 步骤<9>~<12>所经历的时间不同于通过 ADM 的第 5~1 位 (FR2~FR0、LV1、LV0) 设置的转换时间。步骤<11>~<12>所经历的时间为 FR2~FR0、LV1、LV0 设置的转换时间。

11.5 A/D 转换器特征表的阅读方法

以下介绍 A/D 转换器中的专用术语。

(1) 分辨率

这是可被识别的最低模拟输入电压。这是可识别的最小的模拟输入电压，即每位数字输出的模拟输入电压的百分比，称为 1LSB（最低有效位）。对于满度的 1LSB 的百分比用 %FSR（满度范围）表示。

当分辨率为 10 位时 1LSB 表示如下。

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

精确度与分辨率无关，而由总误差决定。

(2) 总误差

总误差是指实际测量值与理论值之间的最大误差。

零度误差、满度误差、积分线性误差和微分线性误差等组合起来表示总误差。

注意量化误差不属于特征表中总误差的范围。

(3) 量化误差

当模拟值转换成数字值时，通常会产生 $\pm 1/2\text{LSB}$ 的误差。在一个 A/D 转换器中，相差 $\pm 1/2\text{LSB}$ 的模拟输入电压被转换成相同的数字代码，因此量化误差不可避免。

注意量化误差不属于特征表中总误差、零度误差、满度误差、积分线性误差和微分线性误差的范围。

图 11-17. 总误差

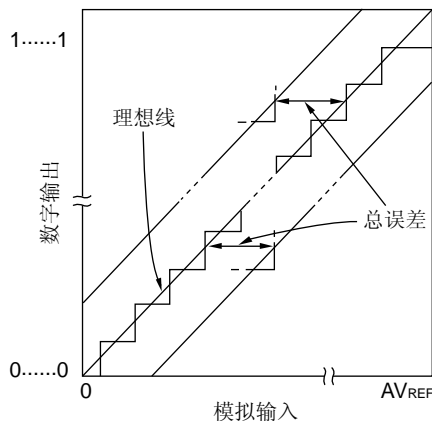
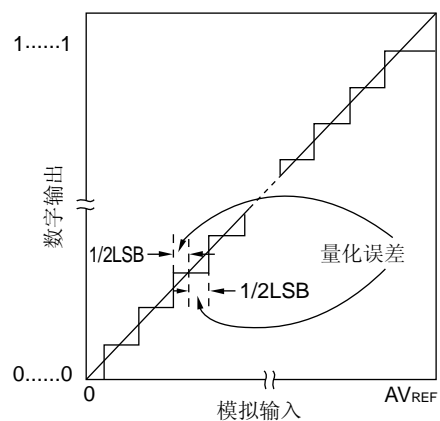


图 11-18. 量化误差



(4) 零度误差

零度误差表示当数字输出范围在 0.....000~0.....001 之间时，模拟输入电压的实际测量值与理论值 ($1/2\text{LSB}$) 之间的误差。

如果实际测量值大于理论值，零度误差表示当数字输出范围在 0.....001~0.....010 之间时，模拟输入电压的实际测量值与理论值 ($3/2\text{LSB}$) 之间的误差。

(5) 满度误差

满度误差表示当数字输出范围在 $1\ldots110 \sim 1\ldots111$ 之间时，模拟输入电压的实际测量值与理论值（满度 $-3/2\text{LSB}$ ）之间的误差。

(6) 积分线性误差

积分线性误差说明了转换特征偏离理想线性关系的程度。它表示当零度误差和满度误差均为 0 时，实际测量值与理想直线之间误差的最大值。

(7) 微分线性误差

当代码输出的理想宽度为 1LSB 时，微分线性误差表示实际测量值与理想值之间的差距。

图 11-19. 零度误差

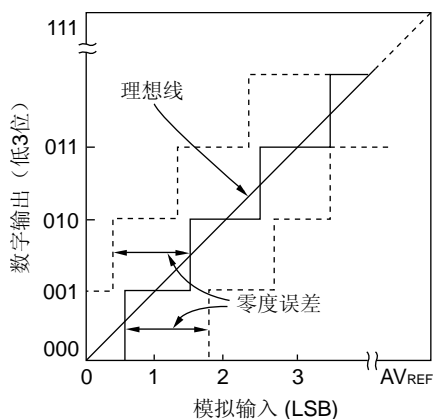


图 11-20. 满度误差

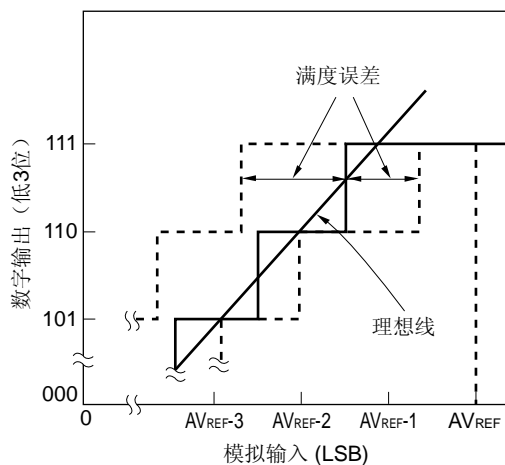


图 11-21. 积分线性误差

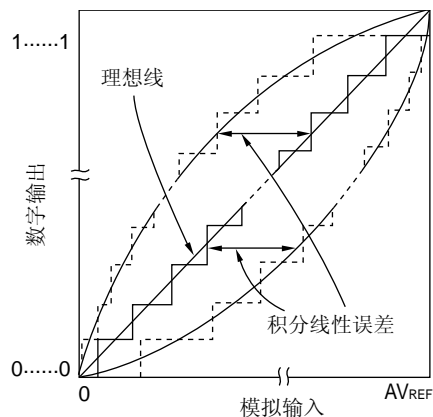
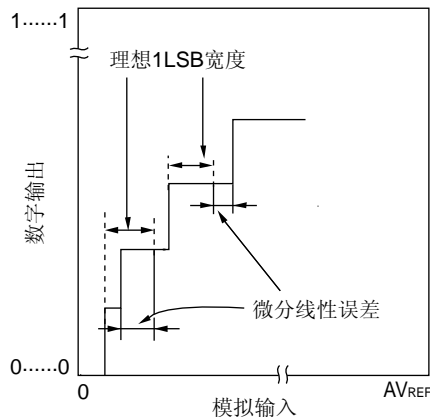


图 11-22. 微分线性误差

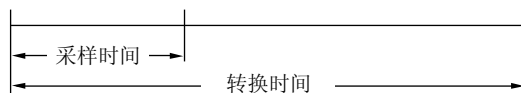
**(8) 转换时间**

转换时间表示从开始采样到获取数字输出所经历的时间。

采样时间包含在特征表中的转换时间中。

(9) 采样时间

采样时间表示模拟电压的模拟开关被打开到模拟电压被采样&保持电路采样所需的时间。



11.6 A/D转换器的注意事项

(1) STOP 模式中的操作电流

在 STOP 模式中 A/D 转换器停止操作。此时，将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 和第 0 位 (ADCE) 清零，可以降低操作电流。

若要从待机状态重新启动，将中断请求标志寄存器 1L (IF1L) 的第 0 位 (ADIF) 清零 (0)，然后开始操作。

(2) ANI0~ANI9 的输入范围

观察 ANI0~ANI9 输入电压的额定范围。如果输入到模拟输入通道的电压大于等于 AV_{REF} 或者小于等于 AV_{SS} (即使在绝对最大额定范围之内)，则该通道的转换值不确定。此外，其它通道的转换值也可能受影响。

(3) 冲突操作

<1> A/D 转换结果寄存器 (ADCR, ADCRH) 的写操作，与由转换结束后的指令对 ADCR 或 ADCRH 的读操作之间的冲突。

ADCR 或 ADCRH 读操作的优先级高。在执行读操作后，才将新的转换结果写入 ADCR 或 ADCRH。

<2> 转换结束后，ADCR 或 ADCRH 的写操作与 A/D 转换器模式寄存器 (ADM) 的写操作、模拟输入通道选择寄存器 (ADS)、或 A/D 端口配置寄存器 (ADPC) 的写操作之间的冲突。

ADM、ADS 或 ADPC 的写操作的优先级高。不执行 ADCR 或 ADCRH 的写操作，也不会产生转换结束中断信号 (INTAD)。

(4) 解决噪音问题的方法

为了保持 10 位分辨率，必须注意输入到 AV_{REF} 引脚和 ANI0~ANI9 引脚的噪音。

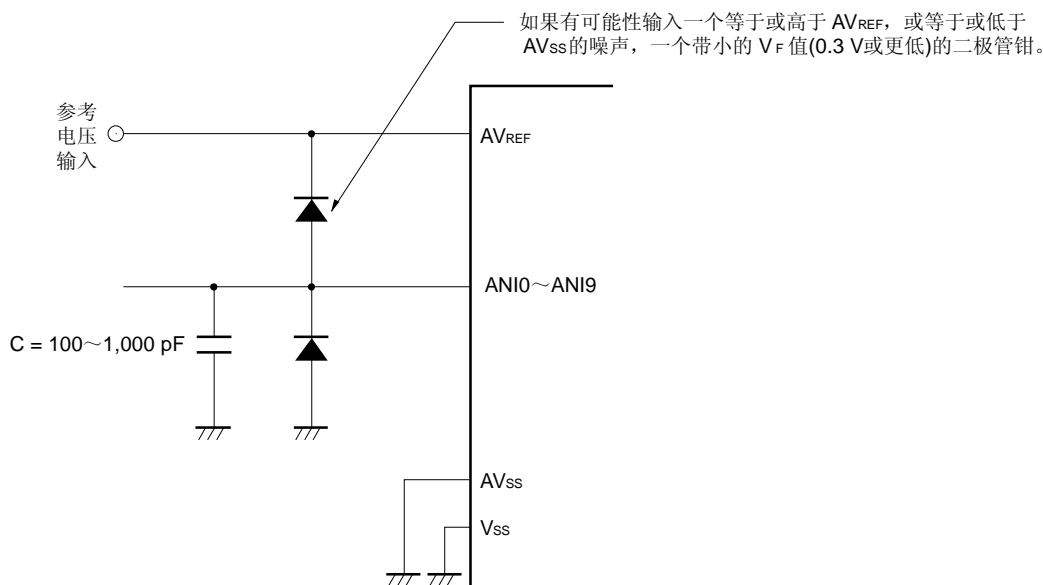
<1> 连接一个低等效电阻和优质频率响应的电容到电源上。

<2> 模拟输入源的输出阻抗越大，干扰就越大。为了降低噪音，建议按图 11-23 所示连接外部 C。

<3> 在转换过程中不要切换引脚。

<4> 如果在转换开始后立即设置 HALT 模式，则可以改善精度。

图 11-23. 模拟输入引脚连接



(5) $ANI0/P20 \sim ANI7/P27$ 和 $ANI8/P150$ 、 $ANI19/P151$

<1> 模拟输入引脚 ($ANI0 \sim ANI7$)也可用作输入端口引脚 ($P20 \sim P27$)。

模拟输入引脚 ($ANI8 \sim ANI9$)也可用作输入端口引脚 ($P150 \sim P151$)。

当选择 $ANI0 \sim ANI9$ 中的任意一个执行 A/D 转换时，转换过程中不要访问 $P20 \sim P27$ 和 $P150 \sim P151$ ；否则转换分辨率可能会降低。选择用作 $P20 \sim P27$ 和 $P150 \sim P151$ 的引脚时，推荐从离 AV_{REF} 最远的 $ANI0/P20$ 开始。

<2> 如果正在进行 A/D 转换的引脚的相邻引脚有数字脉冲，则由于噪音耦合，有可能得不到预期的 A/D 转换值。因此在进行 A/D 转换时不要在相邻引脚引用脉冲。

(6) $ANI0 \sim ANI9$ 引脚的输入阻抗

采样期间 A/D 转换器对采样电容充电，以便进行采样。

因此当不进行采样时仅有漏电流经过，而在采样期间则有对电容充电的电流。因此根据是正在进行采样还是在其它状态，输入阻抗会发生波动。

但是，为确保采样的有效性，建议保持模拟输入源的输出阻抗在 $1k\Omega$ 以下，并将一个约 $100pF$ 的电容连接到 $ANI0 \sim ANI9$ 引脚上（参考图 11-23）。

(7) AV_{REF} 引脚输入阻抗

在 AV_{REF} 与 AV_{SS} 引脚之间连接几十千欧的串联电阻串。

因此，如果参考电压源的输出阻抗很高，相当于一系列电阻连接到 AV_{REF} 与 AV_{SS} 引脚之间的串联电阻串，会导致较大的参考电压误差。

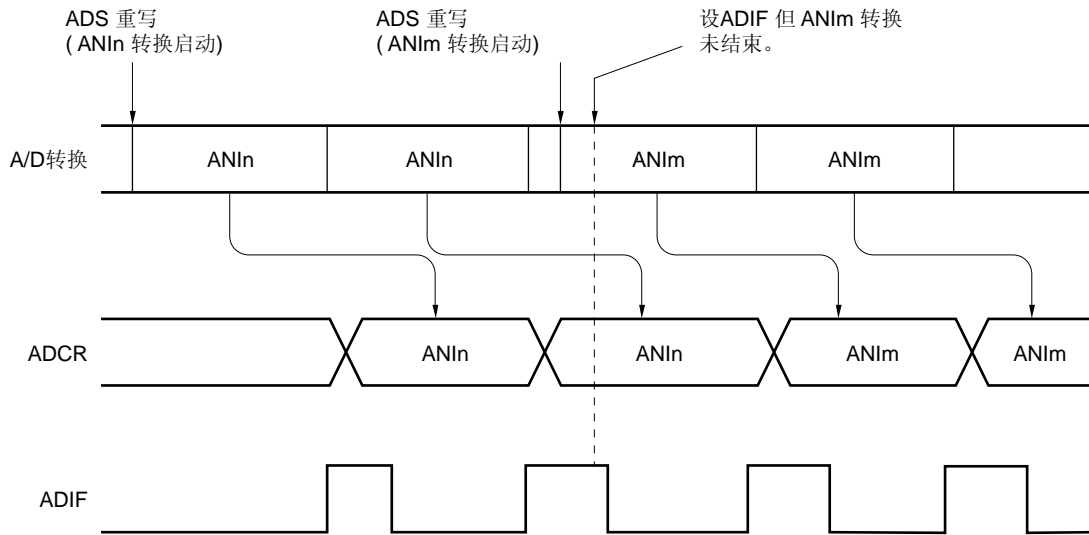
(8) 中断请求标志 (ADIF)

即使模拟输入通道指定寄存器(ADS)的值被修改，中断请求标志(ADIF)也不会被清零。

因此，如果在 A/D 转换期间有一个模拟输入引脚发生变化，则在 ADS 被修改之前，模拟输入通道的 A/D 转换结果和改变前的模拟通道的 ADIF 可能被设置。此时需要注意的是，当 ADS 修改后立即读取 ADIF 时，即使修改后的模拟输入的 A/D 转换尚未结束，也会设置 ADIF。

当 A/D 转换停止然后又继续时，在 A/D 转换继续操作前清除 ADIF。

图 11-24. A/D 转换结束中断请求产生的时序



备注 1. $n = 0 \sim 9$
2. $m = 0 \sim 9$

(9) A/D 转换刚开始时的转换结果

在 A/D 转换开始后，若在 ADCE 置 1 后的 $1 \mu\text{s}$ 内对 ADCS 置 1，或者 ADCE=0 时 ADCS 置 1，那么第一次的 A/D 转换值可能不在额定范围内。可采取措施，如挂起 A/D 转换结束中断请求(INTAD)，并删除第 1 次转换结果。

(10) A/D 转换结果寄存器 (ADCR, ADCRH) 读操作

当对 A/D 转换器模式寄存器 (ADM)、A/D 转换器模式寄存器 1 (ADM1)、模拟输入通道专用寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 进行写入操作时，ADCR 和 ADCRH 的内容可能不确定。转换结束后要读取转换结果，应该在 ADM、ADM1、ADS 或 ADPC 写入操作前进行。如果采用与上述不同的时序操作可能会读取到不正确的转换结果。

(11) 内部等效电路

模拟输入模块的等效电路如下所示。

图 11-25. ANIn 引脚的内部等效电路

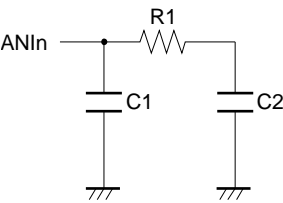


表 11-4. 等效电路的电阻和电容 (参考值)

AVREF	模式	R1	C1	C2
4.0 V ≤ VDD ≤ 5.5 V	标准	5.2 kΩ	8 pF	6.26 pF
	高速 1	5.2 kΩ		
	高速 2	7.8 kΩ		
2.7 V ≤ VDD < 4.0 V	标准	18.6 kΩ		
	高速 2	7.8 kΩ		

- 备注
- 1. 表 11-4 所显示的电阻和电容值不是保证值。
 - 2. n = 0~9

第 12 章 串行阵列单元

每个串行阵列单元有四个串行通道，可以将两个或更多不同的串行接口(3 线串行(CSI)、UART 和简易 I²C)组合起来使用。

μPD79F9211 支持的每个通道功能如下所示。

通道	用作 CSI	用作 UART	用作简易 I ² C
0	CSI00	UART0(支持 LIN 总线)	—
1	CSI01		—
2	CSI10	UART1	IIC10
3	—		—

(整合例子) 当在“UART0”用于通道 0 和 1 时，不能使用 CSI00[‡]和 CSI01[‡]，但可使用 CSI10、UART1 或 IIC10。

12.1 串行阵列单元的功能

μPD79F9211 支持的每个串行接口有以下特性。

12.1.1 3 线串行 I/O (CSI00、CSI01、CSI10)

使用 3 线时钟通信功能：串行时钟 (SCK) 和串行数据 (SI 和 SO) 线。

[数据发送/接收]

- 数据长度 7 或 8 位
- 发送/接收数据的相位控制
- MSB/LSB 可选
- 发送/接收数据的电平设置

[时钟控制]

- 主/从选择
- I/O 时钟的相位控制
- 通过每个通道的预分频器和内部计数器设置发送周期

[中断功能]

- 发送结束中断/缓冲空中断

[错误检测标志]

- 溢出错误

12.1.2 UART (UART0, UART1)

使用 2 线起始停止同步功能：串行数据发送 (TxD) 和串行数据接收 (RxD) 线。它和通信部分(通过使用内部波特率)同步发送或者接收数据。全双工 UART 通信可以被看作是使用两个通道，一个专门发送(偶通道)，一个专门接收(奇通道)。

[数据发送/接收]

- 5, 7, 或者 8 位的数据长度
- 选择 MSB/LSB
- 发送/接收数据的电平设置和反向选择
- 附加校验位和校验检测功能
- 附加停止位

[中断功能]

- 发送结束中断/缓冲空中断
- 帧错误、校验错误、或者溢出错误时产生错误中断

[错误检测标志]

- 帧错误、校验错误、或者溢出错误

UART0(0 和 1 通道)接受 LIN-总线

[LIN 总线功能]

- 唤醒信号检测
 - 同步中断域 (SBF)检测
 - 同步域测量、波特率测量
- } 使用外部中断 (INTP0) 或定时器阵列单元 TAUS。

12.1.3 简易 I²C (IIC10)

使用 2 线用于和两个或者更多设备通信的时钟通信功能：串行时钟(SCL)和串行数据(SDA)。该简易 I²C 专为具有如 EEPROM、Flash 存储器、或者 A/D 转换器设备的单通信设计，因此，它的功能只能作为主，不具有检测等待状态的功能。

确保通过使用软件，和操作控制寄存器，可以读取起始停止条件的 AC 特性。

[数据发送/接收]

- 主发送、主接收(仅具有主功能的主设备)
- ACK 输出功能*和 ACK 检测功能
- 8 位数据长度(当发送地址时，通过高 7 位指定地址，剩下的 1 位用于 R/W 控制)
- 手动产生启动条件和停止条件

[中断功能]

- 发送结束中断

[错误检测标志]

- 校验错误(ACK 错误)

* [简易 I²C 不支持的功能]

- 从发送、从接收
- 仲裁丢失检测功能
- 等待检测功能

注 当接收最后一个数据时，如果 0 写入 SOE02 (SOE0 寄存器)位且串行通信数据输出被停止，ACK 就不会输出。详细信息请参阅 12.7.3 (2)中的处理流程。

12.2 串行阵列单元的配置

串行阵列单元包括下面硬件。

表 12-1. 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(SDR0n) [#] 的低 8 位
串行时钟 I/O	SCK00、SCK01、SCK10 引脚(用于 3 线串行 I/O)、SCL10 引脚(用于简易 I ² C)
串行数据输入	SI00、SI01、SI10 引脚(用于 3 线串行 I/O)、RxD0 引脚(用于 UART 支持 LIN-总线)、RxD1 引脚(用于 UART)
串行数据输出	SO00、SO01、SO10 引脚(用于 3 线串行 I/O)、TxD0 引脚(用于 UART 支持 LIN-总线)、TxD1 引脚(用于 UART)、输出控制
串行数据 I/O	SDA10 引脚(用于简易 I ² C)
控制寄存器	<div> <位设置时钟寄存器> <ul style="list-style-type: none"> • 外围设备使能寄存器 0 (PER0) • 串行时钟选择寄存器 0 (SPS0) • 串行通道使能状态寄存器 0 (SE0) • 串行通道启动寄存器 0 (SS0) • 串行通道停止寄存器 0 (ST0) • 串行输出使能寄存器 0 (SOE0) • 串行输出寄存器 0 (SO0) • 串行输出电平寄存器 0 (SOL0) • 输入切换控制寄存器 (ISC) • 噪声滤波器使能寄存器 0 (NFEN0) </div> <div> <每个通道的寄存器> <ul style="list-style-type: none"> • 串行数据寄存器 0n(SDR0n) • 串行模式寄存器 0n(SMR0n) • 串行通信操作设置寄存器 0n (SCR0n) • 串行状态寄存器 0n(SSR0n) • 串行标志清零触发寄存器 0n(SIR0n) • 端口输入模式寄存器 3、7(PIM3、PIM7) • 端口输出模式寄存器 3、7(POM3、POM7) • 端口模式寄存器 3、7(PM3、PM7) • 端口寄存器 3、7(P3、P7) </div>

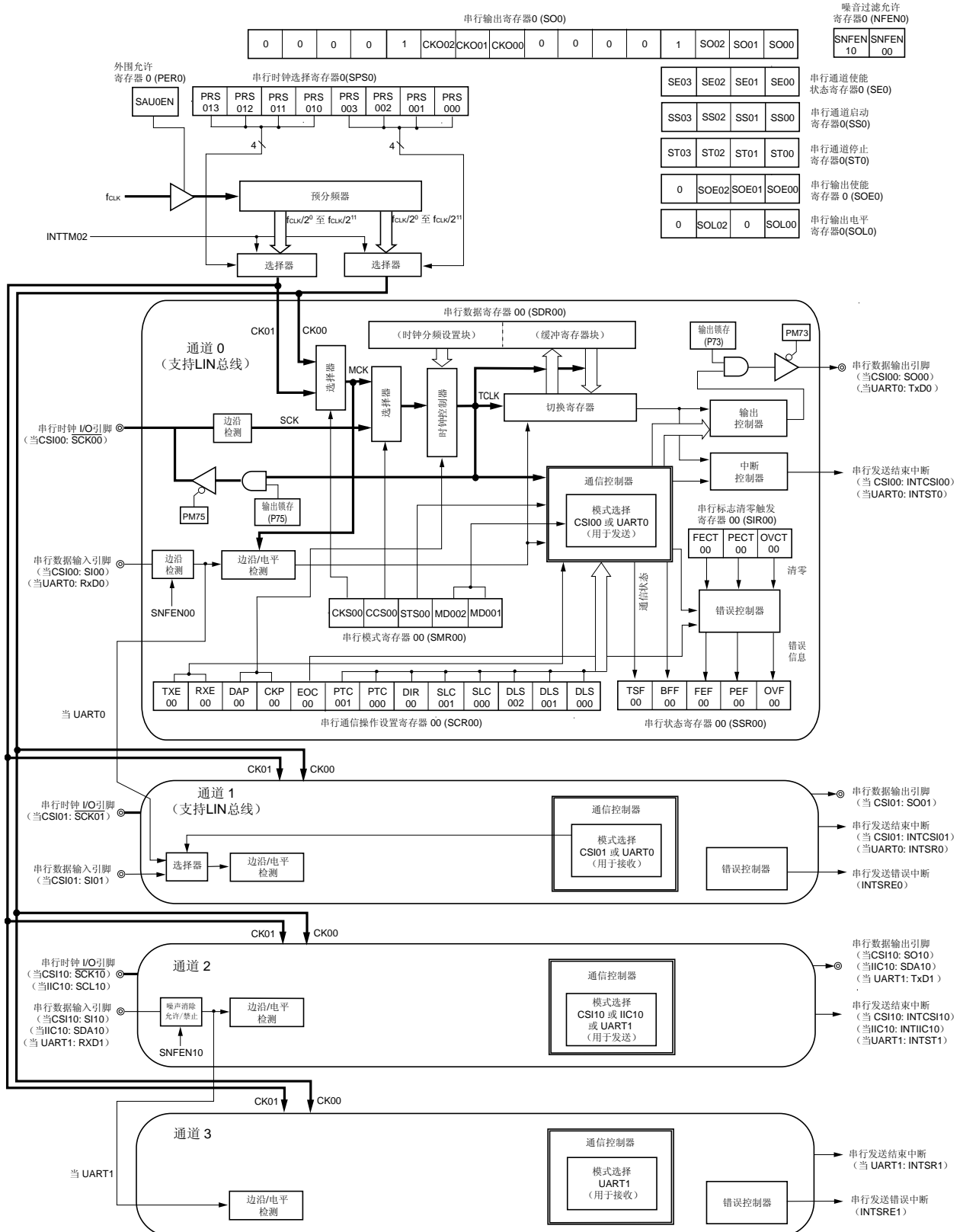
注 根据通信模式的不同,串行数据寄存器 0n (SDR0n) 的低 8 位可以使用以下 SFR 来读取或写入。

- CSIp 通信...SIOp(CSIp 数据寄存器)
- UARTq 接受...RXDq(UARTq 接收数据寄存器)
- UARTq 发送...TXDq(UARTq 发送数据寄存器)
- IIC10 通信...SIO10(IIC10 数据寄存器)

备注 n: 单元编号(n = 0~3)
p: CSI 编号(p = 00、01、10)
q: UART 编号(q = 0, 1)

图 12-1 展示了串行阵列单元的框图。

图 12-1. 串行阵列单元 1 的结构图



(1) 移位寄存器

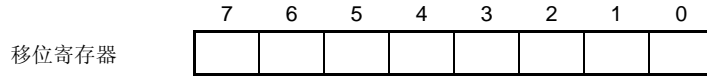
这是一个将并行数据转换到串行数据或者相反的 8 位寄存器。

接收期间，它把输入到串行引脚的数据转换为并行数据。

当数据发送时，设置到这个寄存器的值如同从串行输出引脚的串行数据一样输出。

移位寄存器不能直接通过程序操作。

要读取或写入移位寄存器，使用串行数据寄存器 0n (SDR0n) 的低 8 位。



(2) 串行数据寄存器 0n (SDR0n) 的低 8 位

SDR0n 是通道 n 的发送/接收数据寄存器（16 位）。第 7 位～第 0 位用作发送/接收缓冲寄存器，第 15 位～第 9 位用来设置工作时钟（MCK）的分频比。

当接受数据时，通过保存在低 8 位的移位寄存器来转换并行数据。当发送数据时，设置要发送的数据到移位寄存器的低 8 位。

根据 SCR0n 寄存器的第 0 位～第 2 位 (DLS0n0～DLS0n2) 的设置,保存到该寄存器的低 8 位的数据如下所示，与数据的输出顺序无关。

- 5 位的数据长度（保存在 SDR0n 寄存器的第 0 位～第 4 位中）（只有在 UART 模式下可以设置）
- 7 位的数据长度（保存在 SDR0n 寄存器的第 0 位～第 6 位中）
- 8 位的数据长度（保存在 SDR0n 寄存器的第 0 位～第 7 位中）

SDR0n 可以以 16 位为单元读取或写入。

根据通信模式的不同，SDR0n 的低 8 位可作为下列 SFR 进行读或写。

- CSIp 通信...SIOp (CSIp 数据寄存器)
- UARTq 接受...RXDq (UARTq 接收数据寄存器)
- UARTq 发送...TXDq (UARTq 发送数据寄存器)
- IIC10 通信...SIO10 (IIC10 数据寄存器)

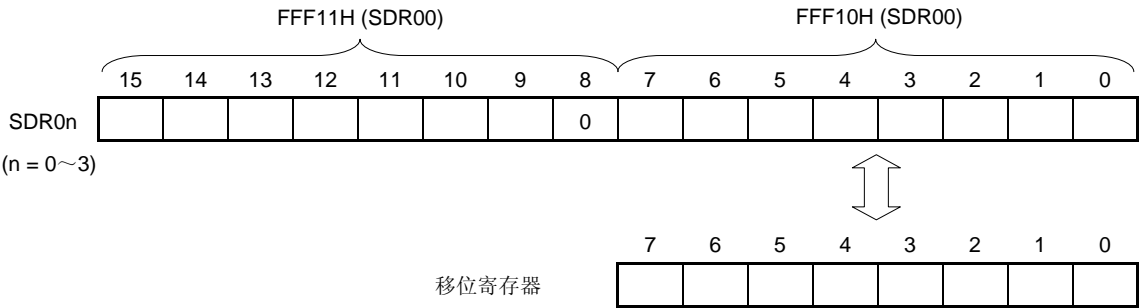
注意 当操作停止后禁止写入 8 位单位 (SE0n = 0)。

复位后这个寄存器的值为 0000H。

- 备注**
1. 在接收数据后，存放在位 0～7 的“0”超出数据长度。
 2. n: 单元编号 (n = 0～3)，
p: CSI 编号 (p = 00、01、10)
q: UART 编号 (q = 0, 1)

图 12-2. 串行数据寄存器 0n(SDR0n)的格式

地址: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), 复位后: 0000H R/W
 FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03)



注意事项 第 8 位必须清“0”。

- 备注
1. 关于 SDR0n 的高 7 位的功能，见 12.3 控制串行阵列单元的寄存器。.
 2. n: 单元编号 (n = 0~3)

12.3 控制串行阵列单元的寄存器

通过下列寄存器控制串行阵列单元。

- 外围设备使能寄存器 0 (PER0)
- 串行时钟选择寄存器 0 (SPS0)
- 串行模式寄存器 0n (SMR0n)
- 串行通信操作设置寄存器 0n (SCR0n)
- 串行数据寄存器 0n (SDR0n)
- 串行状态寄存器 0n (SSR0n)
- 串行标志清零触发寄存器 0n (SIR0n)
- 串行通道使能状态寄存器 0 (SE0)
- 串行通道启动寄存器 0 (SS0)
- 串行通道停止寄存器 0 (ST0)
- 串行输出使能寄存器 0 (SOE0)
- 串行输出电平寄存器 0 (SOL0)
- 串行输出寄存器 0 (SO0)
- 输入切换控制寄存器 (ISC)
- 噪音过滤允许寄存器 0 (NFEN0)
- 端口输入模式寄存器 3、7 (PIM3, PIM7)
- 端口输出模式寄存器 3、7 (POM3, POM7)
- 端口模式寄存器 3、7 (PM3, PM7)
- 端口寄存器 3、7 (P3, P7)

备注 n: 通道 (n = 0~3)

(1) 外围设备使能寄存器 0 (PER0)

PER0 用于允许或者禁止使用每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少消耗和噪音。

当使用串行阵列单元时，确保设置该寄存器的位 2(SAU0EN)为 1。

可以通过 1 位或者 8 位存储器操作指令来设置 PER0。

复位后这个寄存器的值为 00H。

图 12-3. 外围设备使能寄存器 0 (PER0) 的格式

地址: F00F0H 复位后: 00H R/W

符号	<7>	6	<5>	<4>	3	<2>	1	0
PER0	RTCEN	0	ADCEN	0	0	SAU0EN	0	0

SAU0EN	串行阵列单元输入时钟的控制
0	停止输入时钟的供给。 <ul style="list-style-type: none"> 串行阵列单元使用的 SFR 不能写入。 串行阵列单元处于复位状态。
1	供给输入时钟。 <ul style="list-style-type: none"> 串行阵列单元使用的 SFR 可以被读/写。

注意事项 1. 设置串行阵列单元时，首先确保设置 SAU0EN 为 1。如果 SAU0EN=0，向串行阵列单元的控制寄存器写入被忽略，且即使读取寄存器，也只能读到默认值(除了输入切换控制寄存器(ISC)、噪声滤波器允许寄存器 (NFEN0)、端口输入模式寄存器(PIM3, PIM7)、端口输出模式寄存器 (POM3, POM7)、端口模式寄存器 (PM3, PM7)和端口寄存器(P3, P7)外)。

2. 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

3. 确保将 PER0 寄存器的第 0、1、3、4 和 6 位清零。

(2) 串行时钟选择寄存器 0(SPS0)

SPS0 是一个 16 位寄存器，用于选择两种类型的工作时钟(CK00, CK01)，这些时钟通常供给每个通道。CK01 由 SPS0 的第 7 位~第 4 位选择，CK00 通过将第 3 位设置为 0 选择。

当寄存器在工作时(当 SE0n =1 时)，禁止重写 SPS0。

SPS0 可以通过 16 位存储器操作指令来设置。

SPS0 的低 8 位 SPS0L 可用 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 12-4. 串行时钟选择寄存器 0 (SPS0) 的格式

地址: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPS0	0	0	0	0	0	0	0	0	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS Op3	PRS Op2	PRS Op1	PRS Op0	工作时钟(CKOp) ^{※1} 的选择				
					f _{CLK} = 2 MHz	f _{CLK} = 5 MHz	f _{CLK} = 10 MHz	f _{CLK} = 20 MHz
0	0	0	0	f _{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	f _{CLK} /2	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	f _{CLK} /2 ²	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	f _{CLK} /2 ³	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	f _{CLK} /2 ⁴	125 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	f _{CLK} /2 ⁵	62.5 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	f _{CLK} /2 ⁶	31.3 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	f _{CLK} /2 ⁷	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	f _{CLK} /2 ⁸	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	f _{CLK} /2 ⁹	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	f _{CLK} /2 ¹⁰	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	f _{CLK} /2 ¹¹	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	1	1	INTTM02 ^{注2}				
其它值				设置禁止				

- 注 1. 当改变用于 f_{CLK} 的时钟(通过改变系统时钟控制寄存器 (CKC) 的值)时, 在停止串行阵列单元 (SAU) 的操作 (ST0 = 000FH) 之后, 才这样做。当选择 INTTM02 为工作时钟时, 同样停止定时器阵列单元 TAUS (TT0 = 00FFH)。
2. 利用 SPS0 寄存器, 通过设置 TAUS 的 TIS0 寄存器的 TIS02 位为 1, 为输入时钟选择 f_{SUB}/4, 并且选择 INTTM02, 可以在副系统时钟的一个固定分频比时操作 SAU, 无论 f_{CLK} 频率(主系统时钟、副系统时钟)如何。但是, 当改变 f_{CLK}, 必须按照上面注 1 描述的停止 SAU 和 TAUS。

注意事项 1. 确保将位 15~8 清“0”。

2. 在设置 PER0 寄存器为 1 后, 确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

备注 1. f_{CLK}: CPU/外围硬件时钟频率

f_{SUB}: 副系统时钟频率

2. p = 0, 1

(3) 串行模式寄存器 0n (SMR0n)

SMR0n 是设置通道 n 的操作模式的一个寄存器。它也用作选择工作时钟 (MCK)、指定串行时钟 (SCK) 是否能够输入、设置启动触发、操作模式 (CSI, UART, 或者 I²C) 和中断源。这个寄存器同样仅在 UART 模式中用于反向接收数据的电平。

当寄存器操作时 (SE0n=1) 时, SMR0n 禁止重写。但是, MD0n0 位可以在工作期间重写。

SMR0n 可以通过 16 位存储器操作指令来设置。

复位后这个寄存器的值为 0020H。

图 12-5. 串行模式寄存器 0n(SMR0n)(1/2)的格式

地址: F0110H, F0111H (SMR00) ~ F0116H, F0117H (SMR03) 复位后: 0020H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS 0n	CCS 0n	0	0	0	0	0	STS 0n	0	SIS 0n0	1	0	0	MD 0n2	MD 0n1	MD 0n0

CKS 0n	通道 n 的工作时钟 (MCK) 选择
0	通过 PRS 寄存器设置的预分频器输出时钟 CK00
1	通过 PRS 寄存器设置的预分频器输出时钟 CK01
通过沿检测器来使用工作时钟 MCK。另外, 根据 CCS0n 位的设置和 SDR0n 寄存器的高 7 位的设置, 产生一个发送时钟(TCLK)。	

CCS 0n	通道 n 的发送时钟 (TCLK) 的选择
0	由 CKS0n 的位指定的分频工作时钟 MCK
1	从 SCK 引脚输入时钟(在 CSI 模式下从发送)
发送时钟 TCLK 用于切换寄存器、通信控制器、输出控制器、中断控制器和错误控制器。当 CCS0n = 0 时, MCK 的分频比由 SDR0n 寄存器的高 7 位来设置。	

STS 0n	选择一个启动触发源
0	仅软件触发有效 (为 CSI、UART 发送、及简易 I ² C 选择)。
1	RxD 引脚有效沿(为 UART 接收选择)
在 SS0 寄存器被设置为 1 后, 当以上条件满足时, 启动发送。	

注意事项 确保将位 13~9、7、4 和 3 清“0”。确保设置位 5 为“1”。

备注 n: 单元编号 (n = 0~3)

图 12-5. 串行模式寄存器 0n(SMR0n)(2/2)的格式

地址: F0110H, F0111H (SMR00) ~ F0116H, F0117H (SMR03) 复位后: 0020H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS 0n	CCS 0n	0	0	0	0	0	STS 0n	0	SIS 0n0	1	0	0	MD 0n2	MD 0n1	MD 0n0

SIS 0n0	控制在 UART 模式下, 通道 n 接收数据的电平的反向															
0	下降沿检测作为启动位。 捕捉输入通信数据。															
1	上升沿检测作为启动位。 捕捉反向输入通信数据。															

MD 0n2	MD 0n1	通道 n 的操作模式设置													
0	0	CSI 模式													
0	1	UART 模式													
1	0	简易 I ² C 模式													
1	1	设置禁止													

MD0n0	通道 n 的中断源的选择
0	发送结束中断
1	缓冲空中断
对于逐次发送，当 SDR0n 数据发送后通过设置 MD0n0 为 1 写下一个发送数据。	

备注 n: 单元编号 (n = 0~3)

(4) 串行通信操作设置寄存器 0n (SCR0n)

SCR0n 是通道 n 的通信操作设置寄存器。它被用来设置数据发送/接收模式、数据和时钟的相位、错误信号是否被屏蔽、奇偶位、启动位、停止位和数据长度。

当寄存器在操作(当 SE0n=1 时)，禁止修改 SCR0n。

SCR0n 可以通过 16 位存储器操作指令来设置。

复位后这个寄存器的值为 0087H。

图 12-6. 串行通信操作设置寄存器 0n (SCR0n) (1/3) 的格式

地址: F0118H, F0119H (SCR00) ~ F011EH, F011FH (SCR03) 复位后: 0087H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE 0n	RXE 0n	DAP 0n	CKP 0n	0	EOC 0n	PTC 0n1	PTC 0n0	DIR 0n	0	SLC 0n1	SLC 0n0	0	DLS 0n2	DLS 0n1	DLS 0n0

TXE 0n	RXE 0n	通道 n 的操作模式设置
0	0	不开始通信。
0	1	仅接收
1	0	仅发送
1	1	发送/接收

DAP 0n	CKP 0n	在 CSI 模式下，数据和时钟相位的选择
0	0	
0	1	
1	0	
1	1	
在 UART 模式和简易 I ² C 模式下，确保设置 DAP0n、CKP0n = 0。		

注意事项 确保将位 3、6 和 11 清“0”。确保设置位 2 为“1”。

备注 n: 单元编号(n = 0~3)

p: CSI 编号(p = 00、01、10)

图 12-6. 串行通信操作设置寄存器 0n (SCR0n) (2/3) 的格式

地址: F0118H, F0119H (SCR00) ~ F011EH, F011FH (SCR03) 复位后: 0087H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE 0n	RXE 0n	DAP 0n	CKP 0n	0	EOC 0n	PTC 0n1	PTC 0n0	DIR 0n	0	SLC 0n1	SLC 0n0	0	DLS 0n2	DLS 0n1	DLS 0n0

EOC 0n	错误中断信号 (INTSREx (x = 0,1)) 屏蔽的选择
0	屏蔽错误中断 INTSREx(INTSRx 没有被屏蔽)。
1	允许产生错误中断 INTSREx(如果发生错误则屏蔽 INTSRx)。
在 CSI 模式、简易 I ² C 模式和 UART 发送过程中 ^注 , 设置 EOC0n = 0。	
UART 接收过程中, 设置 EOC0n = 1。	

PTC 0n1	PTC 0n0	在 UART 模式中设置校验位	
		发送	接收
0	0	不输出校验位。	不接受校验
0	1	输出 0 校验。	无奇偶校验判断
1	0	输出偶校验。	判断偶校验。
1	1	输出奇校验。	判断奇校验。
在 CSI 模式和简易 I ² C 模式下, 确保设置 PTC0n1=0、PTC0n0=0。			

DIR 0n	在 CSI 和 UART 模式中选择数据发送顺序
0	输入/输出数据 MSB。
1	输入/输出数据 LSB。
在简易 I ² C 模式下确保清除 DIR0n=0。	

SLC 0n1	SLC 0n0	在 UART 模式中设置停止位
0	0	无停止位
0	1	停止位长度 = 1 位
1	0	停止位长度 = 2 位
1	1	设置禁止
当选择发送结束中断, 在全部停止位完成发送后产生中断。		
在 UART 接收过程中以及简易 I ² C 模式下, 设置 1 位停止位(SLC0n1, SLC0n0 = 0, 1)。		
在 CSI 模式中, 设置无停止位(SLC0n1, SLC0n0 = 0, 0)。		

注 当使用 CSI01 但不使用 EOC01 = 0 时, 可能产生中断错误 INTSRE0。

注意事项 确保将位 3、6 和 11 清“0”。确保设置位 2 为“1”。

备注 n: 单元编号 (n = 0~3)

图 12-6. 串行通信操作设置寄存器 0n (SCR0n) (3/3) 的格式

地址： F0118H, F0119H (SCR00) ~ F011EH, F011FH (SCR03) 复位后： 0087H R/W

符号

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE 0n	RXE 0n	DAP 0n	CKP 0n	0	EOC 0n	PTC 0n1	PTC 0n0	DIR 0n	0	SLC 0n1	SLC 0n0	0	DLS 0n2	DLS 0n1	DLS 0n0

DLS 0n2	DLS 0n1	DLS 0n0	在 CSI 和 UART 模式中设置数据长度
1	0	0	5 位的数据长度(保存在 SDR0n 寄存器的第 0 位~第 4 位中) (只有在 UART 模式下可以设置)
1	1	0	7 位的数据长度(保存在 SDR0n 寄存器的第 0 位~第 6 位中)
1	1	1	8 位的数据长度(保存在 SDR0n 寄存器的第 0 位~第 7 位中)
其它值			设置禁止
在简易 I ² C 模式下，确保设置 DLS0n0=1。			

注意事项 确保将位 3、6 和 11 清“0”。确保设置位 2 为“1”。

备注 n: 单元编号 (n = 0~3)

(5) 串行数据寄存器 0n (SDR0n) 的高 7 位

SDR0n 是通道 n 的发送/接收数据寄存器(16 位)。第 7 位~第 0 位用作发送/接收缓冲寄存器，第 15 位~第 9 位用来设置工作时钟 (MCK) 的分频比。

如果串行模式寄存器 (SMR0n) 的 CCS0n 位被清 0，由 SDR0n 的高 7 位设置的工作时钟通过分频设置为发送时钟。

关于 SDR0n 的低 8 位的功能，参见 12.2 串行阵列单元的配置。

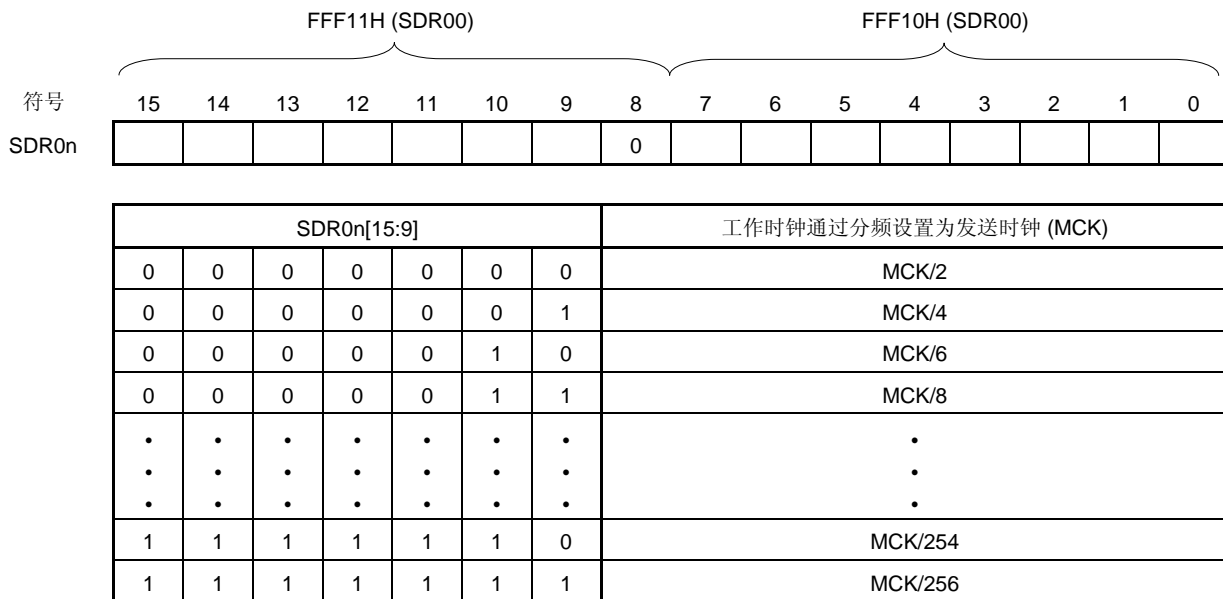
SDR0n 可以以 16 位为单元读取或写入。

然而，仅当操作停止时 (SE0n = 0)，高 7 位才可以被读写。操作运行中 (SE0n = 1)，值仅可以写入 SDR0n 的低 8 位。工作过程中读取 SDR0n，读取值总是 0。

复位后这个寄存器的值为 0000H。

图 12-7. 串行数据寄存器 0n(SDR0n)的格式

地址: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), 复位后: 0000H R/W
FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03)



注意事项 1. 第 8 位必须清“0”。

2. 当使用 UART 时，禁止设置 SDR0n[15:9] = (0000000B, 0000001B)。

备注 1. 关于 SDR0n 的低 8 位的功能，参见 12.2 串行阵列单元的配置。

2. n: 通道 (n = 0~3)

(6) 串行状态寄存器 0n(SSR0n)

SSR0n 是指通道 n 的通信状态和错误发生状态的寄存器。此寄存器可指出帧错误、校验错误和溢出错误。

SSR0n 可由 16 位存储器操作指令读取。

SSR0n 的低 8 位 SSR0nL 可用 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 12-8. 串行状态寄存器 0n(SSR0n) (1/2) 的格式

地址: F0100H, F0101H (SSR00)~F0106H, F0107H (SSR03) 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR0n	0	0	0	0	0	0	0	0	0	TSF 0n	BFF 0n	0	0	FEF 0n	PEF 0n	OVF 0n

TSF 0n	通道 n 的通信状态指示标志
0	通信没有执行。
1	通信执行中。
因为这个标志是一个更新标志，当通信操作完成后它将被自动清除。当 ST0n/SS0n 位设为 1 时，这个标志同样被清除。	

BFF 0n	通道 n 的缓冲寄存器状态指示标志
0	有效数据未被保存在 SDR0n 寄存器中。
1	有效数据被保存在 SDR0n 寄存器中。
<p>这是一个更新标志。当完成从 SDR0n 寄存器到切换寄存器的发送时，它将被自动清除。接收过程中，当数据从 SDR0n 寄存器中读出时，它将被自动清除。当 ST0n/SS0n 位设为 1 时，这个标志同样被自动清除。</p> <p>当 SCR0n 寄存器的 TXE0n 位设为 1 (每个通信模式下的发送或接收模式)，如果发送数据写入 SDR0n 寄存器，这个标志被自动清除。当 SCR0n 寄存器的 RXE0n 位 = 1 (每个通信模式下的发送或者接收模式)时，如果接收数据保存在 SDR0n 寄存器，它就被自动设置。发生接收错误时它也被设置。</p> <p>如果 BFF0n=1 时，数据写入 SDR0n 寄存器，保存在寄存器中的发送/接收数据将被丢弃，并且会检测到溢出错误 (OVF0n=1)。</p>	

备注 n: 单元编号 (n = 0~3)

图 12-8. 串行状态寄存器 0n(SSR0n) (2/2) 的格式

地址: F0100H, F0101H (SSR00)~F0106H, F0107H (SSR03) 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSR0n	0	0	0	0	0	0	0	0	0	TSF 0n	BFF 0n	0	0	FEF 0n	PEF 0n	OVF 0n

FEF 0n	通道 n 的帧错误检测标志
0	没有错误产生。
1	在 UART 接收时产生一个帧错误。 <帧错误原因> 如果 UART 接收完成后, 没有检测到停止位, 则产生一个帧错误。
这是一个累积标志, 直到 SIR0n 寄存器的 FECT0n 位写入 1, 它才被清零。	

PEF 0n	通道 n 的校验错误检测标志
0	没有错误产生。
1	在 UART 接收时产生一个校验错误或者在 I ² C 发送中没有检测到 ACK。 <校验错误原因> <ul style="list-style-type: none"> 在 UART 接收中, 如果发送数据的校验和校验位不匹配, 则产生一个校验错误。 在 I²C 发送中, 如果 ACK 信号没有在 ACK 接收时序中从设备返回, 则没有检测到 ACK。
这是一个累积标志, 直到 SIR0n 寄存器的 PECT0n 位写入 1, 它才被清零。	

OVF 0n	通道 n 的溢出错误检测标志
0	没有错误产生。
1	产生一个溢出错误。 <溢出错误的原因> <ul style="list-style-type: none"> 保存在 SDR0n 寄存器中的接收数据没有读取, 并且发送数据或者下一个接收数据被写入。 在 CSI 模式中, 数据发送还没有准备好从设备的发送或者接收。
这是一个累积标志, 直到 SIR0n 寄存器的 OVCT0n 位写入 1, 才被清零。	

备注 n: 单元编号 (n = 0~3)

(7) 串行标志清零触发寄存器 0n (SIR0n)

SIR0n 是个触发寄存器，它用于清除通道 n 的每个错误标志。

当该寄存器的每一位 (FECT0n、PECT0n、OVCT0n) 被设置为 1 时，串行状态寄存器 0n 的对应位 (FEF0n、PEF0n、OVF0n) 被清除为 0。因为 SIR0n 是个触发寄存器，当 SSR0n 的对应位被清除时，它立即被清除。

SIR0n 可以通过 16 位存储器操作指令来设置。

SIR0n 的低 8 位 SIR0nL 能用 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 12-9. 串行标志清除触发寄存器 0n (SIR0n) 的格式

地址: F0108H, F0109H (SIR00)~F010EH, F010FH (SIR03) 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIR0n	0	0	0	0	0	0	0	0	0	0	0	0	0	FEC T0n	PEC T0n	OVC T0n

FEC T0n	清除通道 n 的帧错误的触发															
0	没有触发操作															
1	清除 SSR0n 寄存器的 FEF0n 位为 0。															

PEC T0n	清除通道 n 的校验错误的触发															
0	没有触发操作															
1	清除 SSR0n 寄存器的 PEF0n 位为 0。															

OVC T0n	清除通道 n 的溢出错误的触发															
0	没有触发操作															
1	清除 SSR0n 寄存器的 OVF0n 位为 0。															

注意事项 确保将位 15~3 清“0”。

- 备注
- 1. n: 单元编号 (n = 0~3)
 - 2. 读 SIR0n 寄存器时，读取的值总是为 0000H。

(8) 串行通道使能状态寄存器 0 (SE0)

SE0 表示每个通道的数据发送/接收操作是允许还是禁止。

当串行通道启动寄存器 0 (SS0) 的位写入 1 时，设置该寄存器的相应位为 1。当串行通道停止寄存器 0 (ST0) 的位写入 1 时，清除该寄存器的相应位为 0。

使能操作的通道 n 不能由软件重写串行输出寄存器 0 (SO0) 的 CKO0n 的值 (下面介绍)，但是由通信操作反映的数值会从串行时钟引脚输出。

停止操作的通道 n 可以通过软件设置 SO0 寄存器的 CKO0n 的值，并且从串行时钟引脚输出它的值。这样，任何波形，例如启动条件/停止条件，都可以通过软件产生。

SE0 可由 16 位存储器操作指令读取。

SE0 低 8 位 SE0L 可用 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 12-10. 串行通道使能状态寄存器 0 (SE0) 的格式

地址: F0120H, F0121H 复位后: 0000H R

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE0 3	SE0 2	SE0 1	SE0 0

SE0 n	指示通道 n 的操作允许/停止状态
0	操作停止（此时控制寄存器和切换寄存器的值、串行时钟 I/O 引脚状态、串行数据输出引脚，且 FEF、PEF、和 OVF 错误标志会保留 ^註 ）。
1	操作允许

注 SSR0n 寄存器的第 6 位和第 5 位 (TSF0n, BFF0n) 被清除。

备注 n: 单元编号 (n = 0~3)

(9) 串行通道启动寄存器 0(SS0)

SS0 是用来允许启动每个通道的通信/计数的触发寄存器。

当 1 写入到此寄存器 (SS0n) 的某一位，串行通道使能状态寄存器 0(SE0)的相应位(SE0n)设为 1。因为 SS0n 是一个触发位，当 SE0n = 1 时将立即清零。

SS0 可以通过 16 位存储器操作指令来设置。

SS0 低 8 位 SS0L 可用 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 12-11. 串行通道启动寄存器 0(SS0)的格式

地址: F0122H, F0123H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

SS0n	通道 n 的操作起始触发
0	没有触发操作
1	设置 SE0n 为 1，并且进入通信等待状态（如果信操作已经执行，则操作停止、等待启动条件）。

- 注意事项** 确保将位 15~4 清“0”。
- 备注**
- 1. n: 单元编号 (n = 0~3)
 - 2. 读 SS0 寄存器时，读取的值总是为 0000H。

(10) 串行通道停止寄存器 0(ST0)

ST0 是用来允许停止每个通道的通信/计数触发寄存器。

当 1 写入到此寄存器 (ST0n) 的一个位，串行通道允许状态寄存器 0(SE0)的相应位设为 0。因为 ST0n 是一个触发位，当 SE0n = 0 时它被立即清零。

ST0 可以通过 16 位存储器操作指令来设置。

ST0 低 8 位 ST0L 可用 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 12-12. 串行通道停止寄存器 0 (ST0) 的格式

地址: F0124H, F0125H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST0 3	ST0 2	ST0 1	ST0 0

ST0n	通道 n 的操作停止触发
0	没有触发操作
1	清除 SE0n 为 0,并停止通信操作。 (此时控制寄存器和切换寄存器的值、串行时钟 I/O 引脚状态的值、串行数据输出引脚的值, 且 FEF、PEF 和 OVF 错误标志值会保留 [※])。

注 SSR0n 寄存器的第 6 位和第 5 位 (TSF0n, BFF0n) 被清除。

注意事项 确保将位 15~4 清“0”。

- 备注 1. n: 单元编号 (n = 0~3)
2. 读 ST0 寄存器时, 读取的值总是为 0000H。

(11) 串行输出使能寄存器 0 (SOE0)

SOE0 是用来使能或停止每个通道的串行通信输出的寄存器。

使能串行输出的通道 n 不能通过软件重写串行输出寄存器 0 (SO0)的 SO0n 的值 (后面将介绍)，但由通信操作反映的数值可从串行数据输出引脚输出。

对于串行输出停止的通道 n，SO0 寄存器的 SO0n 值可以通过软件设置，并且这个值可以从串行数据输出引脚输出。这样，启动条件和停止条件的任何波形可以通过软件产生。

SOE0 可以通过 16 位存储器操作指令来设置。

SOE0 的低 8 位 SOE0L 能用 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 12-13. 串行输出使能寄存器 0(SOE0)的格式

地址: F012AH, F012BH 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02	SOE01	SOE00

SOE0n	通道 n 的串行输出允许/禁止
0	通过串行通信操作停止输出。
1	通过串行通信操作允许输出。

注意事项 确保将 SOE0 的位 15~3 清“0”。

备注 n: 单元编号 (n = 0~2)

(12) 串行输出寄存器 0 (SO0)

SO0 是每个通道的串行输出缓冲寄存器。

这个寄存器每位的值从通道 n 的串行数据输出引脚来输出。

这个寄存器每位 (n + 8) 的值从通道 n 的串行时钟输出引脚来输出。

仅当串行输出禁止时 (SOE0n = 0)，此寄存器的 SO0n 可以通过软件重写。当串行输出允许时 (SOE0n = 1)，通过软件重写被忽略，并且该寄存器的值仅可通过一个串行通信操作来改变。

仅当通道操作禁止时 (SE0n = 0)，此寄存器的 CKO0n 可以通过软件重写。当通道操作允许时 (SE0n = 1)，通过软件重写被忽略，并且 CKO0n 的值仅可通过一个串行通信操作来改变。

为了使用下列引脚作为端口功能引脚，设置相应的 CKO0n 和 SO0n 位至“1”。

- 44 引脚: P30/SO10/TxD1/TO11、P31/SI10/RxD1/SDA10/INTP1/TI09、
P32/ $\overline{\text{SCK10}}$ /SCL10/INTP2、P70/SO01/INTP4、P72/ $\overline{\text{SCK01}}$ /INTP6、
P73/SO00/TxD0/TO10、P75/ $\overline{\text{SCK00}}$ /TI11

SO0 可以通过 16 位存储器操作指令来设置。

复位后这个寄存器的值为 0F0FH。

图 12-14. 串行输出寄存器 0 (SO0) 的格式

地址: F0128H, F0129H 复位后: 0F0FH R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO 02	CKO 01	CKO 00	0	0	0	0	1	SO 02	SO01	SO 00

CKO 0n	通道 n 的串行时钟输出
0	串行时钟输出值为“0”。
1	串行时钟输出值为“1”。

SO 0n	通道 n 的串行数据输出
0	串行数据输出值为“0”。
1	串行数据输出值为“1”。

注意事项 确保将 SO0 的位 11 和 3 置“1”。并且确保将 SO0 的位 15~12 和 7~4 清“0”。

备注 n: 单元编号 (n = 0~2)

(13) 串行输出电平寄存器 0(SOL0)

SOL0 是用来设置每个通道数据输出电平反转的寄存器。

仅在 UART 模式下可以设置这个寄存器。确保在 CSI 模式和简易 I²C 模式中设置为 0000H。

仅当串行输出允许 (SOE0n = 1) 时，通过该寄存器将通道 n 反向，由引脚输出反映。当串行输出禁止 (SOE0n = 0) 时，SO0n 位的值输出。

当寄存器在操作时（当 SE0n=1 时），禁止修改 SOL0。

SOL0 可以通过 16 位存储器操作指令来设置。

SOL0 的低 8 位 SOL0L 可用 8 位存储器操作指令设置。

复位后这个寄存器的值为 0000H。

图 12-15. 串行输出电平寄存器 0 (SOL0) 的格式

地址: F0134H, F0135H 复位后: 0000H R/W

符号	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL02	0	SOL00

SOL0n	在 UART 模式中选择通道 n 的发送数据反向电平
0	通信数据原样输出。
1	通信数据反向并输出。

注意事项 确保将位 15~3 和 1 清“0”。

备注 n: 通道 (n = 0, 2)

(14) 输入切换控制寄存器 (ISC)

通过 UART0、外部中断和定时器阵列单元 TAUS 的联合操作，ISC 用于实现 LIN 总线通信操作。

当 0 位设置为 1 时，串行数据输入 (RxD0) 引脚的输入信号可选择为外部中断 (INTP0)，该中断可以用来检测唤醒信号。

当第 1 位设置为 1 时，串行数据输入 (RxD0) 引脚的输入信号选择为定时器输入，所以，同步中断域和同步域的脉冲宽度可以由定时器测量。

ISC 可以通过一个 1 位或者 8 位存储器操作指令来设置。

复位后这个寄存器的值为 00H。

图 12-16. 输入切换控制寄存器(ISC)的格式

地址: FFF3CH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ISC	0	0	0	ISC4	ISC3	ISC2	ISC1	ISC0

ISC1	切换定时器阵列单元 TAUS 的通道 7 输入
0	使用 TI07 引脚的输入信号作为定时器输入（正常模式）。
1	RxD0 引脚的输入信号被用作定时器输入（唤醒信号检测）。

ISC0	切换外部中断（INTP0）输入
0	使用 INTP0 引脚的输入信号作为外部中断（正常模式）。
1	使用 RxD0 引脚输入信号作为外部中断 (用来测量同步中断域和同步域的脉冲宽度)。

注意事项 确保将位 7~5 清“0”。

(15) 噪音过滤允许寄存器 0 (NFEN0)

NFEN0 用于设置是否为每个通道从串行数据输入引脚的输入信号使用噪音过滤。
通过清除该寄存器的对应位为 0，禁止用于 CSI 或简易 I²C 通信的引脚的噪声滤波器。
通过设置这个寄存器的相应位为 1，允许用于 UART 通信引脚的噪音过滤。
当允许噪音过滤，CPU/外围工作时钟 (f_{CLK}) 是和 2 时钟匹配检测同步的。
NFEN0 可以通过一个 1 位或者 8 位存储器操作指令来设置。
复位后这个寄存器的值为 00H。

图 12-17. 噪音过滤使能寄存器 0 (NFEN0)

地址: F0060H

复位后: 00H

R/W

符号	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	SNFEN10	0	SNFEN00

SNFEN10	RxD1/SDA10/SI10/INTP1/TI09/P31 引脚噪声滤波器的使用
0	噪音过滤关闭
1	噪音过滤开启
设置 SNFEN10 为 1 来使用 RxD1 引脚。 清除 SNFEN10 为 0 来使用 SDA10、SI10、INTP1、TI09 和 P31 引脚。	

SNFEN00	使用 RxD0/SI00/TI10/P74 引脚的噪声滤波器
0	噪音过滤关闭
1	噪音过滤开启
设置 SNFEN00 为 1 来使用 RxD0 引脚。 清除 SNFEN00 为 0 来使用 SI00、TI10 和 P74 引脚。	

注意事项 确保将位 7~3 和 1 清“0”。

(16) 端口输入模式寄存器 3, 7 (PIM3, PIM7)

这些寄存器以 1 位单元设置端口 3 和 7 的输入缓冲器。

PIM3 和 PIM7 可以通过 1 位或 8 位存储器操作指令来设置。

复位信号的产生可将这些寄存器清零(00H)。

图 12-18. 端口输入模式寄存器 3 和 7 的格式 (PIM3 和 PIM7)

地址: F0043H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PIM3	0	0	0	0	0	PIM32	PIM31	0

地址: F0047H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
PIM7	0	0	PIM75	PIM74	0	PIM72	PIM71	0

PIMmn	Pmn 引脚输入缓冲器选择(m = 3、7; n = 1、2、4、5)
0	普通输入缓冲器
1	TTL 输入缓冲器

(17) 端口输出模式寄存器 3, 7 (POM3, POM7)

这些寄存器以 1 位为单元设置端口 3 和 7 的输出模式。

POM3 和 POM7 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生可将这些寄存器清零(00H)。

图 12-19. 端口输入模式寄存器 3 和 7 的格式 (POM3 和 POM7)

地址: F0053H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
POM3	0	0	0	0	0	POM32	POM31	POM30

地址: F0057H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	0
POM7	0	0	POM75	0	POM73	POM72	0	POM70

POMmn	Pmn 引脚输出缓冲器选择(m = 3、7; n = 0~3, 5)
0	正常输出模式
1	N-ch 漏极开路 (V _{DD} 耐压) 模式

(18) 端口模式寄存器 3、7(PM3, PM7)

这些寄存器以 1 位单元设置端口 3 和 7 端口的输入/输出。

当使用 P30/SO10/TxD1/TO11、P31/SI10/RxD1/SDA10/INTP1/TI09、P32/SCK10/SCL10/INTP2、P70/SO01/INTP4、P72/SCK01/INTP6、P73/SO00/TxD0/TO10 和 P75/SCK00/TI11 引脚用于串行数据输出或者串行时钟输出时，将 PM30 ~ PM32、PM70、PM72、PM73 和 PM75 清零，并设置 PM30 ~ PM32、PM70、PM72、PM73 和 PM75 的输出锁存为 1。

当使用 P31/SI10/RxD1/SDA10/INTP1/TI09、P32/SCK10/SCL10/INTP2、P71/SI01/INTP5、P72/SCK01/INTP6、P74/SI00/RxD0/TI10 和 P75/SCK00/TI11 引脚用于串行数据输入或串行时钟输入时，设置 PM31、PM32、PM71、PM72、PM74 和 PM75 位至 1。此时，P31、P32、P71、P72、P74 和 P75 的输出锁存器可为 0 或 1。

PM3 和 PM7 可以通过 1 位或 8 位存储器操作指令设置。

复位信号的产生将这些寄存器设置为 FFH。

图 12-20. 端口模式寄存器 3 和 7 (PM3, PM7) 的格式

地址: FFF23H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	PM32	PM31	PM30

地址: FFF27H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PMmn	Pmn 引脚 I/O 模式选择 (m = 3, 7; n = 0~5)
0	输出模式 (输出缓冲器开)
1	输入模式 (输出缓冲器关)

12.4 操作停止模式

串行阵列单元的每个串行接口都有操作停止模式。

在这种模式下，串行通信不能执行，因此可以降低功耗。

此外，在此模式下，以下引脚可以作为普通端口引脚。

- 44 引脚：
P30/SO10/TxD1/TO11、P31/SI10/RxD1/SDA10/INTP1/TI09、
P32/ $\overline{\text{SCK10}}$ /SCL10/INTP2、P70/SO01/INTP4、P71/SI01/INTP5、
P72/ $\overline{\text{SCK01}}$ /INTP6、P73/SO00/TxD0/TO10、P74/SI00/RxD0/TI10、
P75/ $\overline{\text{SCK00}}$ /TI11

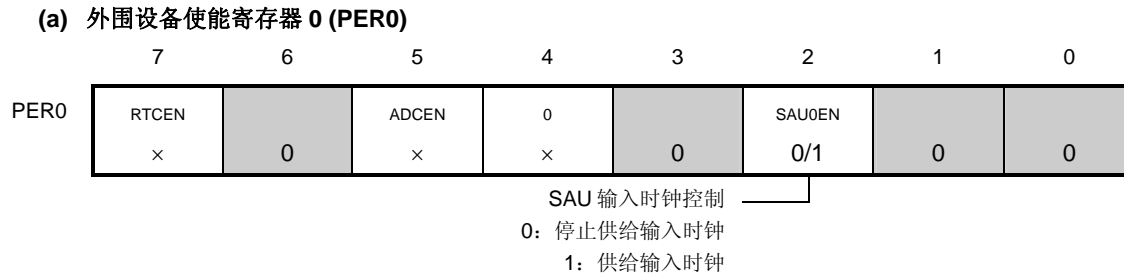
12.4.1 由部件停止操作

由部件停止操作可以通过外围使能寄存器 0(PER0)来设置。

PER0 用于允许或者禁止使用每个外围硬件模块。不能通过停止供应到硬件模块的时钟来减少消耗和噪音。

要停止串行阵列单元的操作，设置位 2(SAU0EN)为 0。

图 12-21. 由部件停止操作时，外围使能寄存器 0 (PER0) 的设置



注意事项 1. 如果 **SAU0EN=0**，向串行阵列单元的控制寄存器写入被忽略，且即使读取寄存器，也只能读到默认值（除了输入切换控制寄存器(ISC)、噪声滤波器允许寄存器 (NFEN0)、端口输入模式寄存器 (PIM3, PIM7)、端口输出模式寄存器 (POM3, POM7)、端口模式寄存器 (PM3, PM7)和端口寄存器(P3, P7)外）。

2. 确保将 PER0 寄存器的第 0、1、3、4 和 6 位清零。

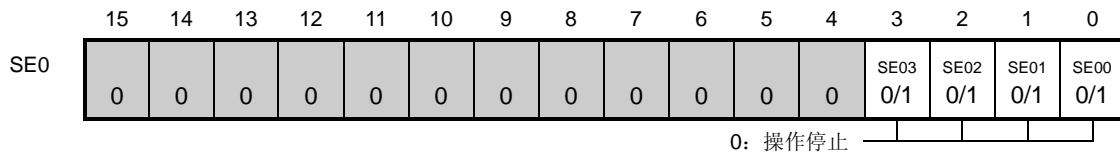
- 备注**
- : 禁止设置(由固件给出固定值)
 - ×: 位不与串行阵列单元共同使用(根据其他外设功能的设定)
 - 0/1: 根据用户的使用方法设为 0 或 1

12.4.2 由通道停止操作

由通道停止操作使用以下每一个寄存器设置。

图 12-22. 由通道停止操作时，每个寄存器的设置 (1/2)

(a) 串行通道使能状态寄存器 0(SE0) ...该寄存器表示每个通道的数据发送/接收操作被允许还是被禁止。



* SE0 寄存器是一个只读寄存器，它可以使用 ST0 寄存器来停止操作。

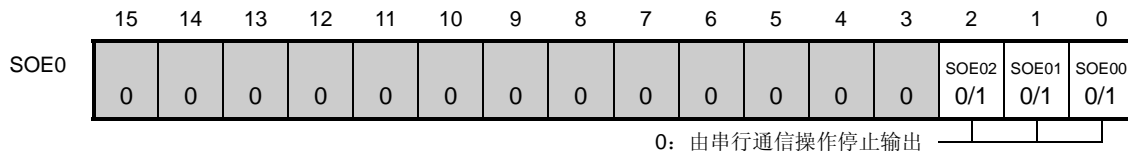
当通道停止操作时，SO0 寄存器的 CKO0n 的值可以通过软件设置。

(b) 串行通道停止寄存器 0(ST0) ... 该寄存器为触发寄存器，用来使能每个通道停止通信/计数。



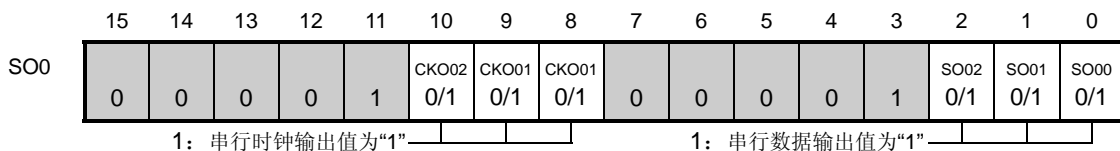
* 因为 ST0n 是触发位，当 SE0n = 0 时，它立即被清除。

(c) 串行输出使能寄存器 0(SOE0)... 该寄存器是用来使能或停止每个通道的串行通信输出的寄存器。



* 对于通道 n，串行输出停止，SO0 寄存器的 SO0n 值可以通过软件设置。

(d) 串行输出寄存器 0(SO0) ...该寄存器为缓冲寄存器，用于每个通道的串行输出。



* 当使用对应于每个通道的引脚作为端口功能引脚时，设置对应的 CKO0n 和 SO0n 位为 1。

备注 n: 通道编号(n = 0~3)

■: 设置禁止（由硬件固定），0/1: 根据用户的用法设置为 0 或 1

12.5 3 线串行 I/O (CSI00、CSI01、CSI10) 通信的操作

使用 3 线的时钟通信功能：串行时钟 (SCK) 和串行数据 (SI 和 SO) 线。

[数据发送/接收]

- 数据长度 7 或 8 位
- 发送/接收数据的相位控制
- MSB/LSB 可选
- 发送/接收数据的电平设置

[时钟控制]

- 主/从选择
- I/O 时钟的相位控制
- 通过每个通道的预分频器和内部计数器设置发送周期

[中断功能]

- 发送结束中断/缓冲空中断

[错误检测标志]

- 溢出错误

支持 3 线串行 I/O (CSI00、CSI01、CSI10) 的通道是 SAU 的通道 0~2 和通道。

通道	用作 CSI	用作 UART	用作简易 I ² C
0	CSI00	UART0 (支持 LIN 总线)	—
1	CSI01		—
2	CSI10	UART1	IIC10
3	—		—

3 线串行 I/O (CSI00、CSI01、CSI10) 执行下列通信操作的 6 种类型。

- 主发送 (参见 12.5.1.)
- 主接收 (参见 12.5.2.)
- 主发送/接收 (参见 12.5.3.)
- 从发送 (参见 12.5.4.)
- 从接收 (参见 12.5.5.)
- 从发送/接收 (参见 12.5.6.)

12.5.1 主发送

主发送是μPD79F9211 输出一个发送时钟和发送数据到另外一个设备。

3 线串行 I/O	CSI00	CSI01	CSI10
目标通道	SAU 的通道 0	SAU 的通道 1	SAU 的通道 2
使用引脚	SCK00, SO00	SCK01, SO01	SCK10, SO10
中断	INTCSI00	INTCSI01	INTCSI10
	可以选择发送结束中断（在单发送模式下）或者缓冲空中断（在连续发送模式下）。		
错误检测标志	无		
发送数据长度	7 或 8 位		
发送速率	Max. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] ^註 f_{CLK} : 系统时钟频率。		
数据相位	通过 DAP0n 位可选择 <ul style="list-style-type: none"> • DAP0n = 0: 从串行时钟开始操作时, 开始输出数据。 • DAP0n = 1: 在串行时钟操作开始之前半个时钟, 开始输出数据。 		
时钟相位	通过 CKP0n 位选择 <ul style="list-style-type: none"> • CKP0n = 0: 正向 • CKP0n = 1: 反向 		
数据方向	MSB 或者 LSB		

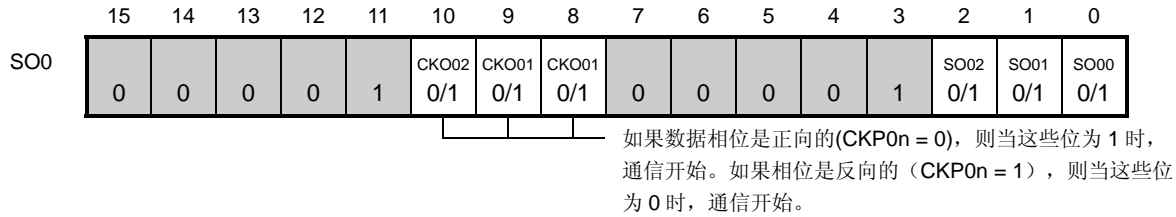
注 请在满足上面条件和电气特性中的 AC 特性的范围内, 使用这个操作（参见 第 26 章 电气特性（目标））。

备注 n: 通道编号 (n=0~2)

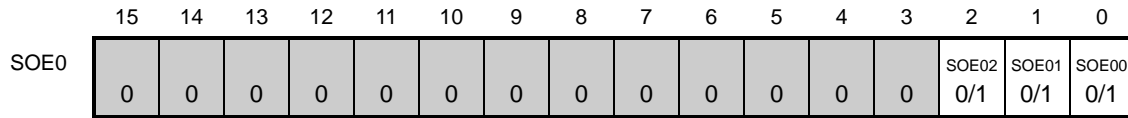
(1) 寄存器设置

图 12-23. 用于 3 线串行 I/O (CSI00、CSI01、CSI10) 的主发送的寄存器内容的示例

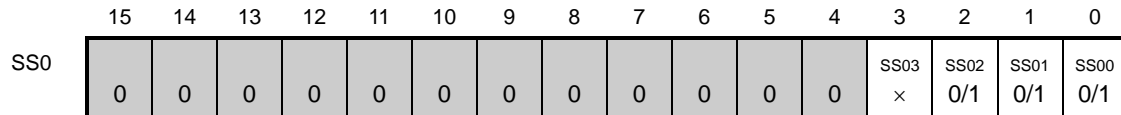
(a) 串行输出寄存器 0 (SO0)...仅设置目标通道的位。



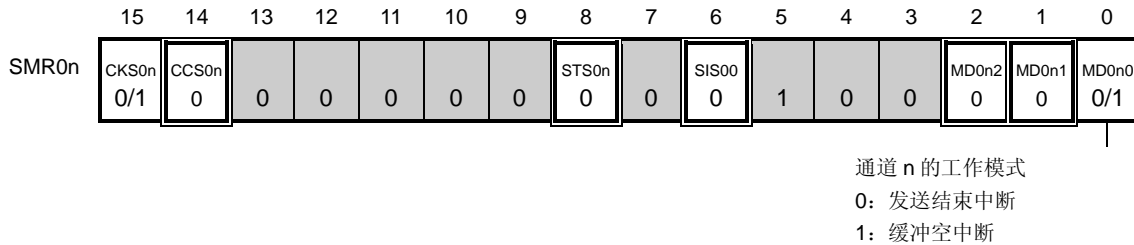
(b) 串行输出使能寄存器 0(SOE0)...仅设置目标通道的位为 1。



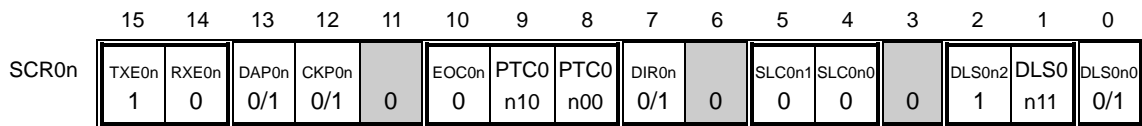
(c) 串行通道启动寄存器 0 (SS0)...仅设置目标通道的位为 1。



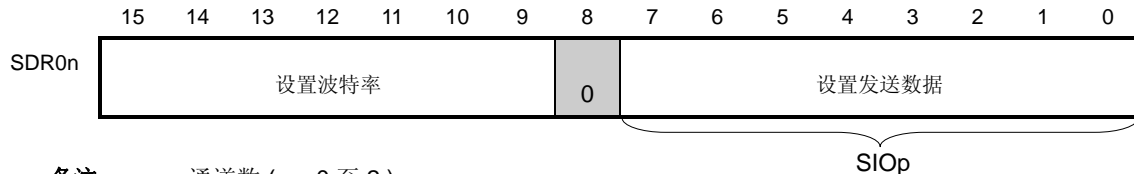
(d) 串行模式寄存器 0n (SMR0n)



(e) 串行通信操作设置寄存器 0n (SCR0n)



(f) 串行数据寄存器 0n (SDR0n) (低 8 位: SI0p)



备注

n: 通道数 (n = 0 至 2)

p: CSI 数 (p = 00, 01, 10)

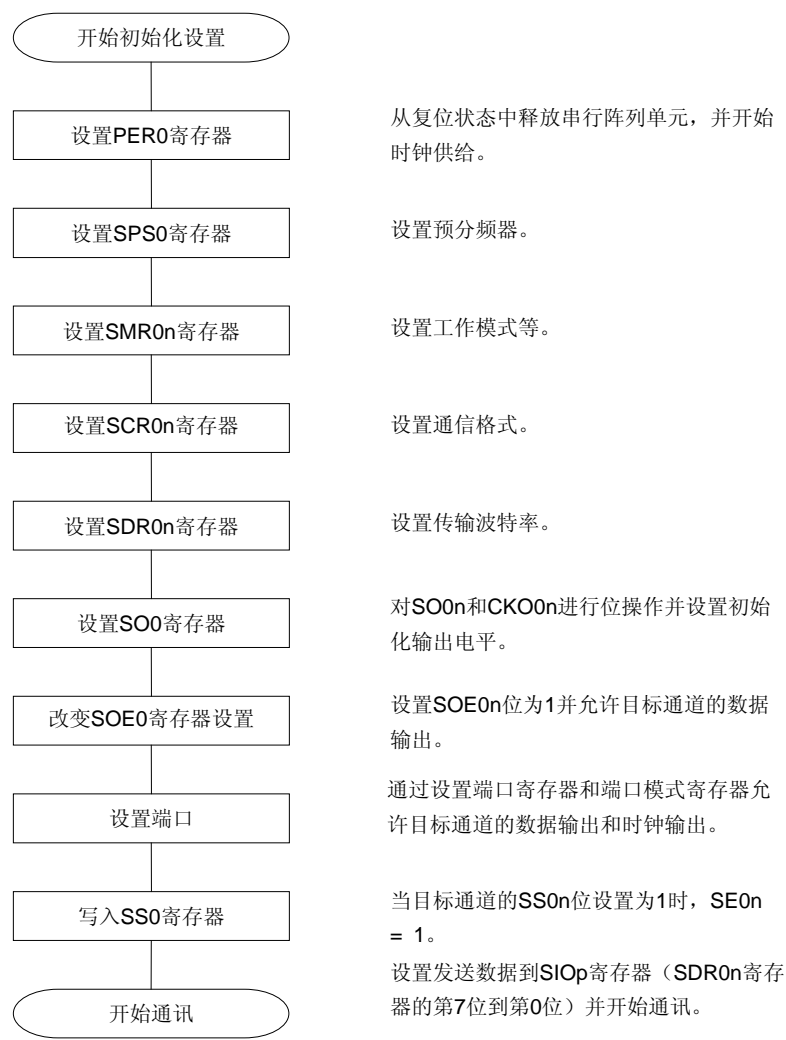
□: 在 CSI 主发送模式下固定设置, ■: 禁止设置 (设为初始值)

X: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1

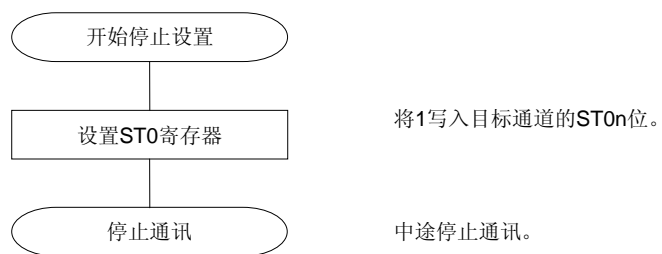
(2) 操作程序

图 12-24. 主发送的初始化设置程序



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

图 12-25. 主发送的停止程序



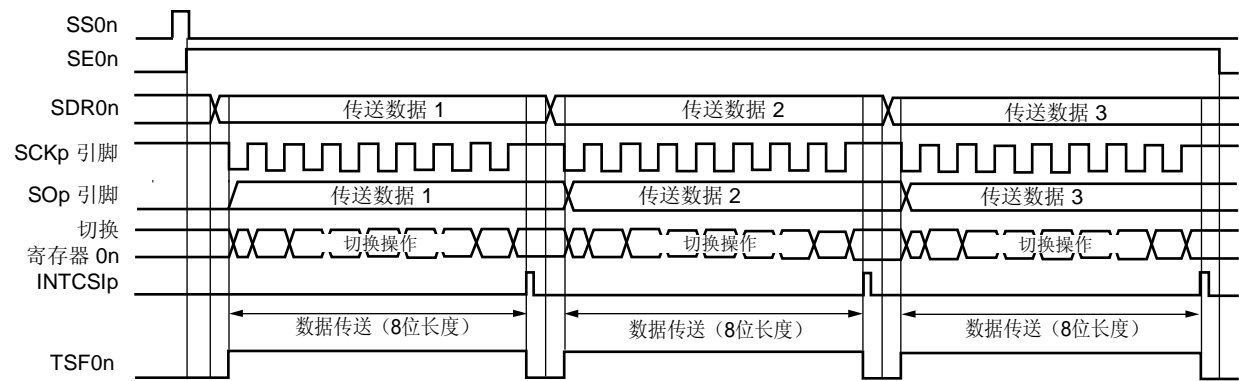
备注 即使通信停止后，引脚电平依然保持。要恢复操作，重新设置 SO0 寄存器（参见 图 12-26 恢复主发送的程序）。

图 12-26. 恢复主发送的程序



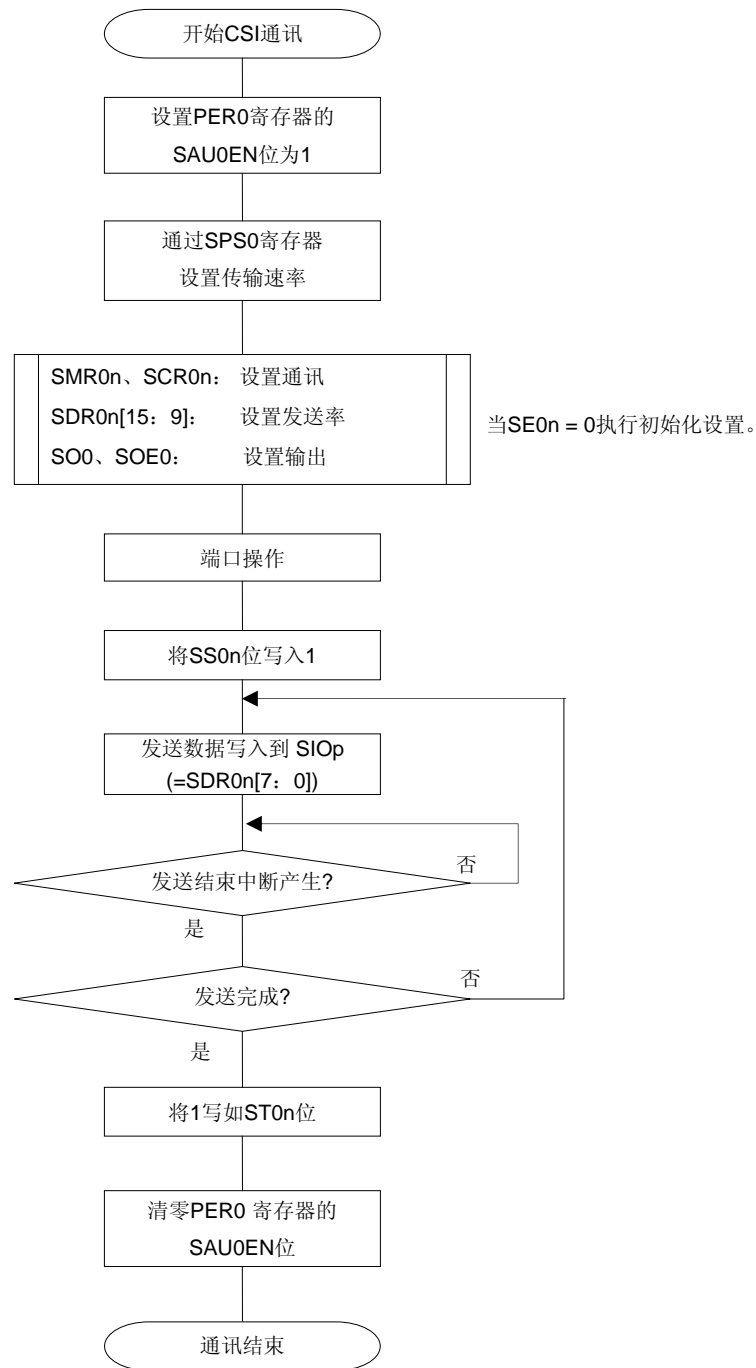
(3) 处理流程（在单发送模式下）

图 12-27. 主发送的时序图（在单发送模式下）



备注 n: 通道编号 (n = 0~2)
p: CSI 编号 (p = 00、01、10)

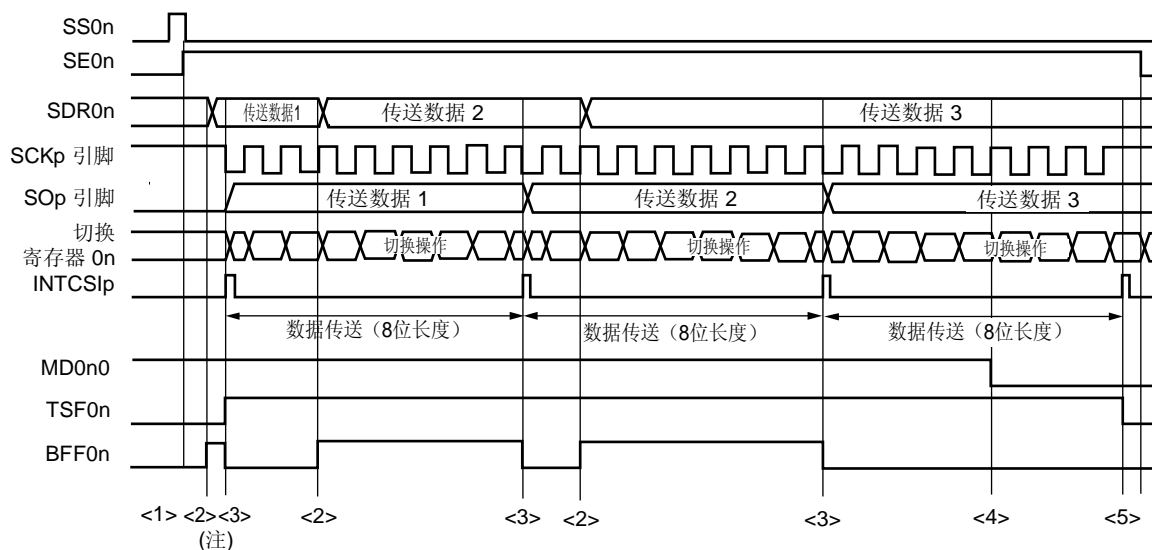
图 12-28. 主发送的流程图（在单发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

(4) 处理流程（在连续发送模式下）

图 12-29. 主发送的时序图（在连续发送模式下）



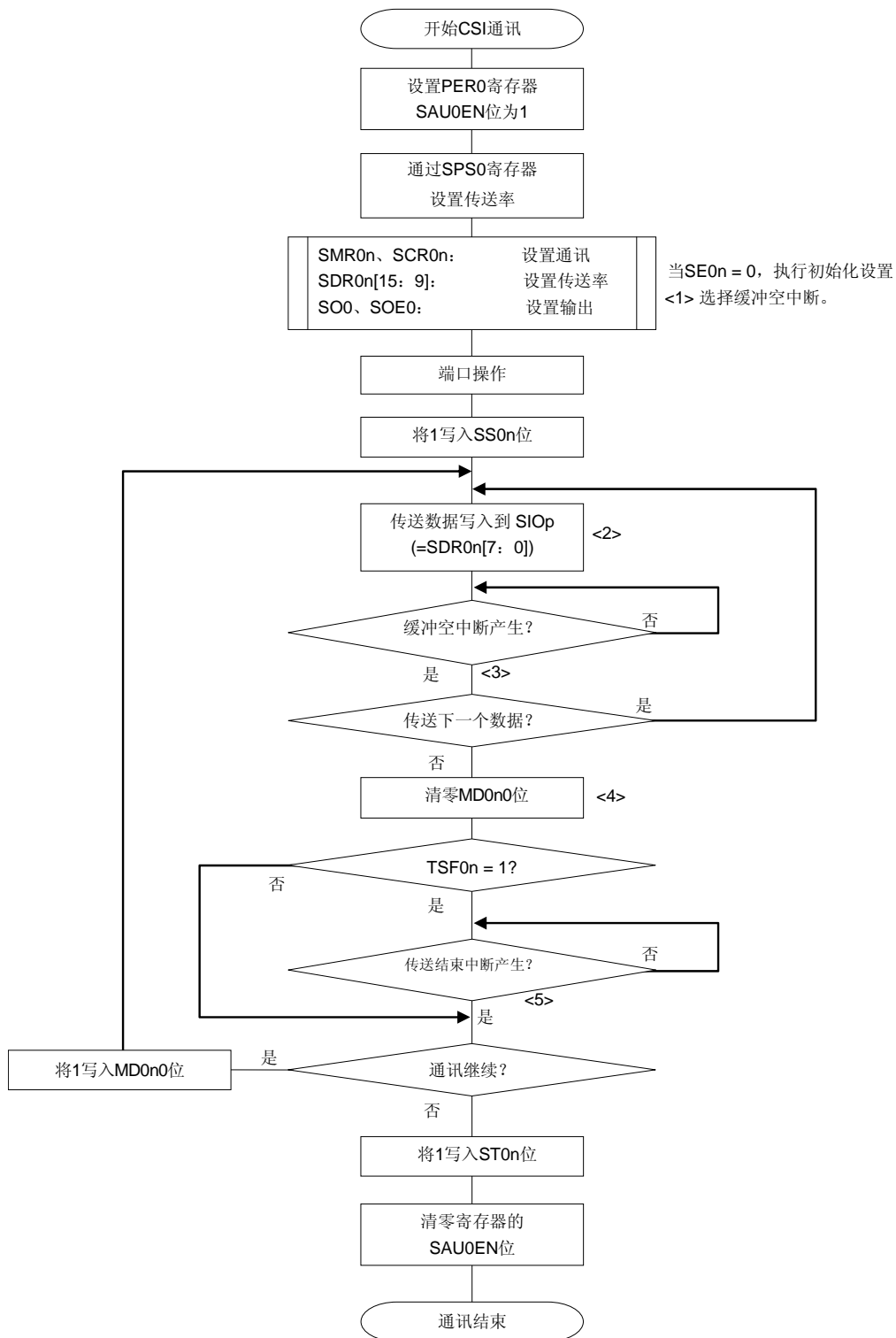
注 在 BFF0n = 1 的情况下，当发送数据写入 SDR0n 寄存器时，发送数据可以被重写。

注意事项 即使在操作过程中，MD0n0 位也可以被重写。

但是，要在最后一位开始发送之前重写它，这样它可以在最后一个发送数据的发送结束中断之前被重写。

备注 n: 通道编号 (n = 0~2)
p: CSI 编号 (p = 00、01、10)

图 12-30. 主发送的流程图（在连续发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

备注 图中的<1>~<5> 对应图 12-29 主发送的时序图（在连续发送模式下）中的<1>~<5>。

12.5.2 主接收

主接收是 μ PD79F9211 输出一个发送时钟和接收其它设备的数据。

3 线串行 I/O	CSI00	CSI01	CSI10
目标通道	SAU 的通道 0	SAU 的通道 1	SAU 的通道 2
使用引脚	SCK00, SI00	SCK01, SI01	SCK10, SI10
中断	INTCSI00	INTCSI01	INTCSI10
	仅发送结束中断（禁止设置缓冲空中断。）		
错误检测标志	只有溢出错误检测标志 (OVF0n)		
发送数据长度	7 或 8 位		
发送速率	最大 fCLK/4 [MHz], 最小 fCLK/ ($2 \times 2^{11} \times 128$) [MHz] [‡] fCLK: 系统时钟频率		
数据相位	通过 DAP0n 位可选择 <ul style="list-style-type: none"> • DAP0n = 0: 从串行时钟开始操作时,数据输入开始。 • DAP0n = 1: 在串行时钟操作开始之前半个时钟, 数据输入开始。 		
时钟相位	通过 CKP0n 位选择 <ul style="list-style-type: none"> • CKP0n = 0: 正向 • CKP0n = 1: 反向 		
数据方向	MSB 或者 LSB		

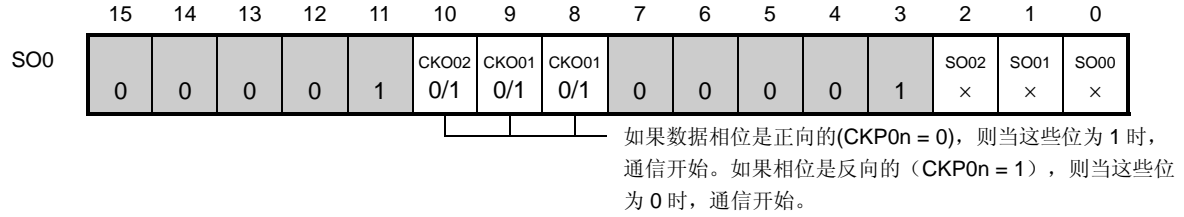
注 请在满足上面条件和电气特性中的 AC 特性的范围内, 使用这个操作（参见 第 26 章 电气特性（目标））。

备注 n: 通道编号 (n=0~2)

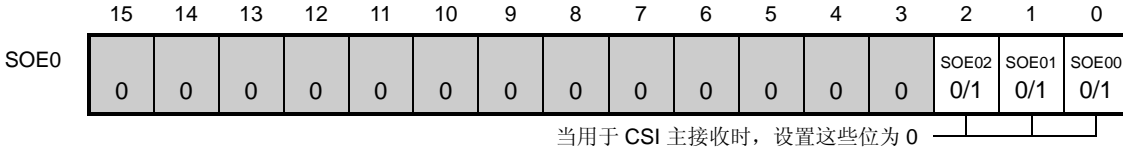
(1) 寄存器设置

图 12-31. 用于 3 线串行 I/O (CSI00、CSI01、CSI10) 的主接收的寄存器内容的示例

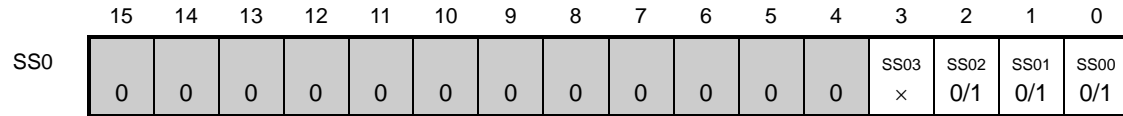
(a) 串行输出寄存器 0 (SO0)...仅设置目标通道的位。



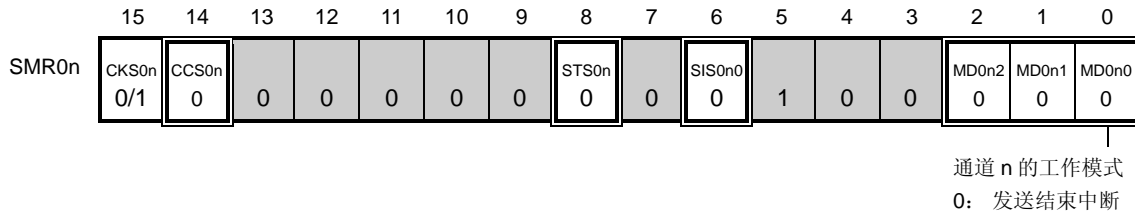
(b) 串行输出使能寄存器 0 (SOE0)



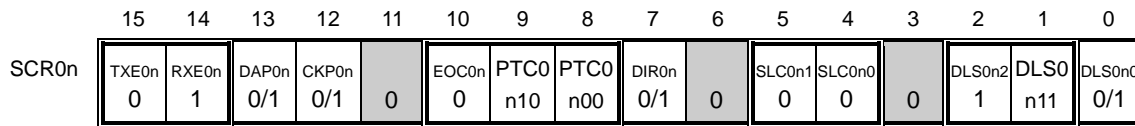
(c) 串行通道启动寄存器 0 (SS0)...仅设置目标通道的位为 1。



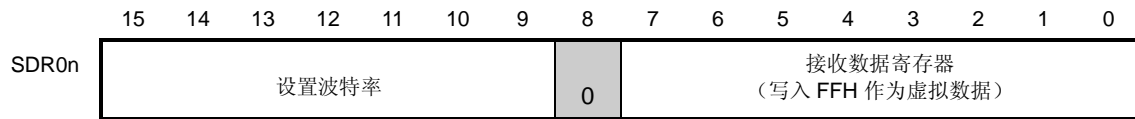
(d) 串行模式寄存器 0n (SMR0n)



(e) 串行通信操作设置寄存器 0n (SCR0n)



(f) 串行数据寄存器 0n (SDR0n) (低 8 位: SIOp)



备注

n: 通道数 (n = 0 至 2)

p: CSI 数 (p = 00, 01, 10)

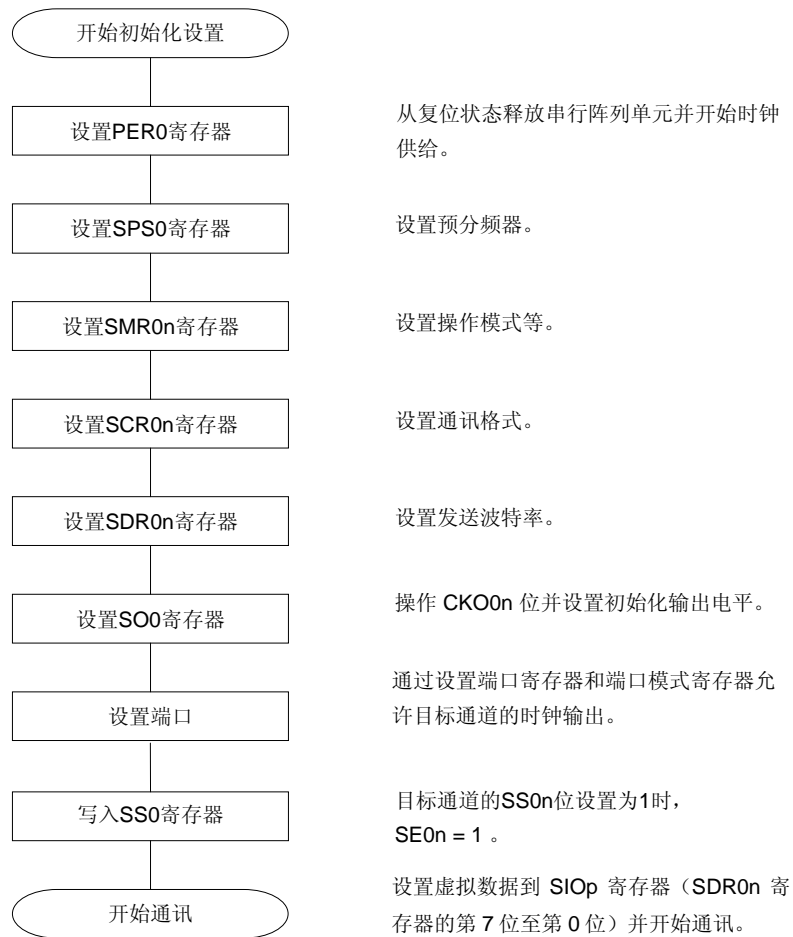
□: 在 CSI 主发送模式下固定设置, ■: 禁止设置 (设为初始值)

X: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1

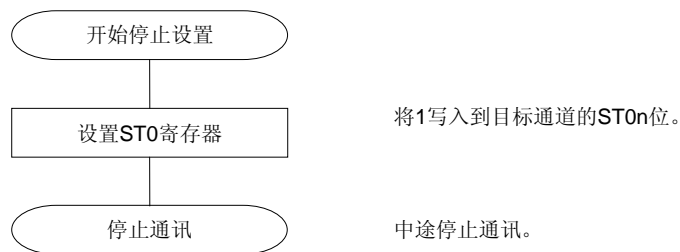
(2) 操作程序

图 12-32. 主接收的初始化程序



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

图 12-33. 主接收的停止程序



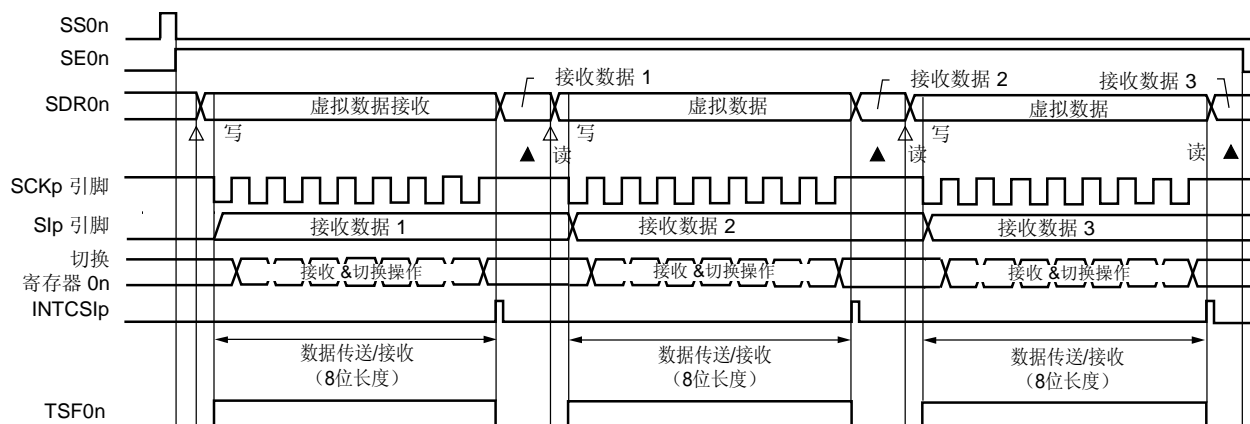
备注 即使通信停止后，引脚电平依然保持。要重新操作，重新设置 SO0 寄存器（参见 图 12-34 恢复主接收的程序）。

图 12-34. 恢复主接收的程序



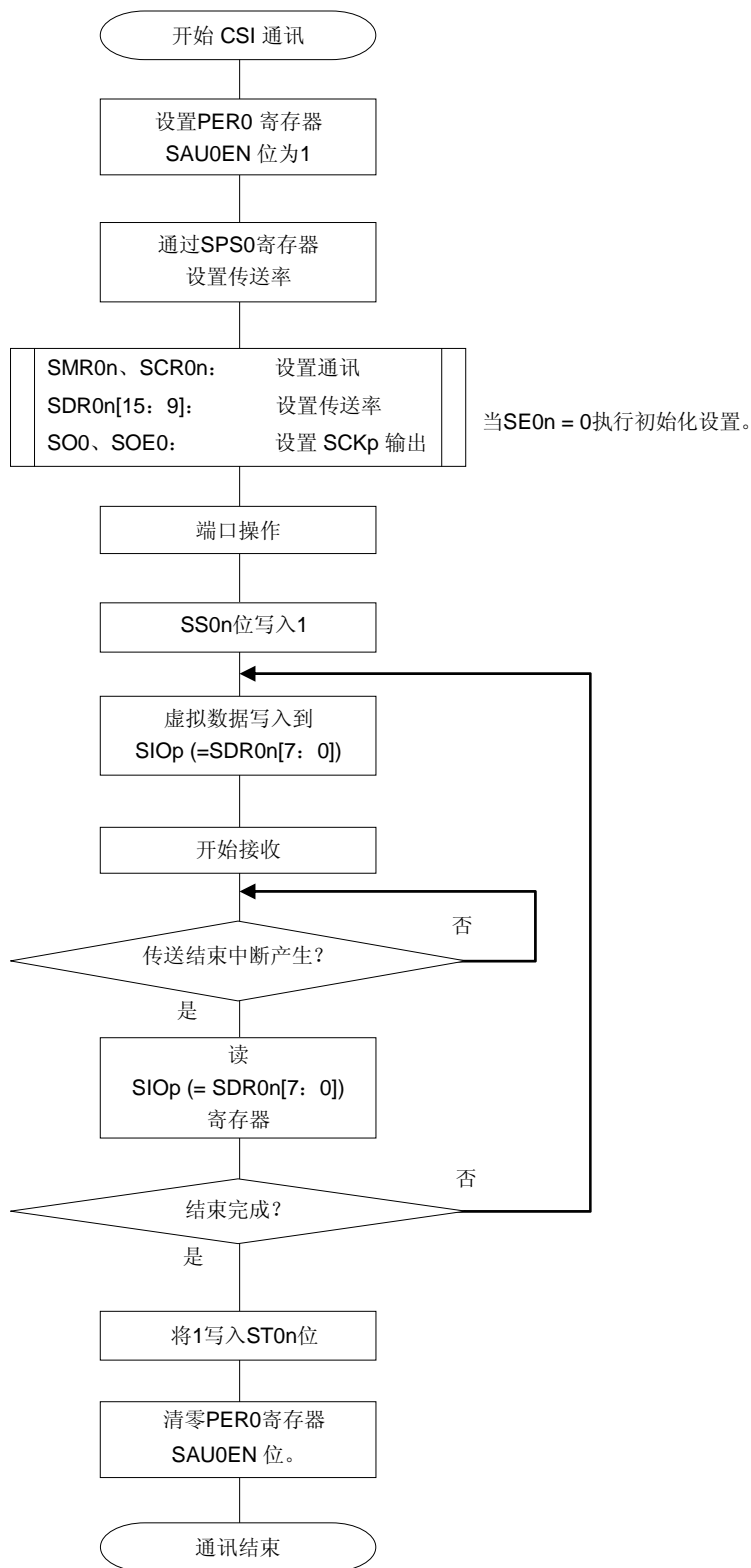
(3) 处理流程（在单接收模式下）

图 12-35. 主接收的时序图（在单接收模式下）



备注 n: 通道编号 (n = 0~2)
p: CSI 编号 (p = 00、01、10)

图 12-36. 主接收的流程图（在单接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

12.5.3 主发送/接收

主发送/接收是由 μ PD79F9211 输出发送时钟，并且发送数据到其它设备，或从其他设备接收数据。

3 线串行 I/O	CSI00	CSI01	CSI10
目标通道	SAU 的通道 0	SAU 的通道 1	SAU 的通道 2
使用引脚	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10
中断	INTCSI00	INTCSI01	INTCSI10
	可以选择发送结束中断（在单发送模式下）或者缓冲空中断（在连续发送模式下）		
错误检测标志	只有溢出错误检测标志 (OVF0n)		
发送数据长度	7 或 8 位		
发送速率	最大 fCLK/4 [MHz], 最小 fCLK/ ($2 \times 2^{11} \times 128$) [MHz] ^注 fCLK: 系统时钟频率		
数据相位	通过 DAP0n 位可选择 <ul style="list-style-type: none"> • DAP0n = 0: 从串行时钟操作开始时,数据 I/O 开始。 • DAP0n = 1: 在串行时钟操作开始之前半个时钟, 数据 I/O 开始。 		
时钟相位	通过 CKP0n 位选择 <ul style="list-style-type: none"> • CKP0n = 0: 正向 • CKP0n = 1: 反向 		
数据方向	MSB 或者 LSB		

注 请在满足上面条件和电气特性中的 AC 特性的范围内，使用这个操作（参见 第 26 章 电气特性（目标））。

备注 n: 通道编号 (n=0~2)

(1) 寄存器设置

图 12-37. 用于 3 线串行 I/O（CSI00、CSI01、CSI10）的主发送/接收的寄存器内容的示例

(a) 串行输出寄存器 0 (SO0)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1	CKO01 0/1	CKO01 0/1	0	0	0	0	1	SO02 0/1	SO01 0/1	SO00 0/1

如果数据相位是正向的(CKP0n = 0)，则当这些位为 1 时，通信开始。如果相位是反向的 (CKP0n = 1)，则当这些位为 0 时，通信开始。

(b) 串行输出使能寄存器 0(SOE0)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	SOE01 0/1	SOE00 0/1

(c) 串行通道启动寄存器 0 (SS0)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 0/1	SS00 0/1

(d) 串行模式寄存器 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 0	0	0	0	0	0	STS0n 0	0	SIS0n0 0	1	0	0	MD0n2 0	MD0n1 0	MD0n0 0/1

通道 n 的工作模式
0: 发送结束中断
1: 缓冲空中断

(e) 串行通信操作设置寄存器 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 1	RXE0n 1	DAP0n 0/1	CKP0n 0/1	0	EOC0n 0	PTC0 n10	PTC0 n00	DIR0n 0/1	0	SLC0n1 0	SLC0n0 0	0	DLS0n2 1	DLS0 n11	DLS0n0 0/1

(f) 串行数据寄存器 0n (SDR0n) (低 8 位: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	设置波特率								0	设置发送数据/接收数据寄存器						

SIOp

备注

n: 通道数 (n = 0 至 2)

p: CSI 数 (p = 00, 01, 10)

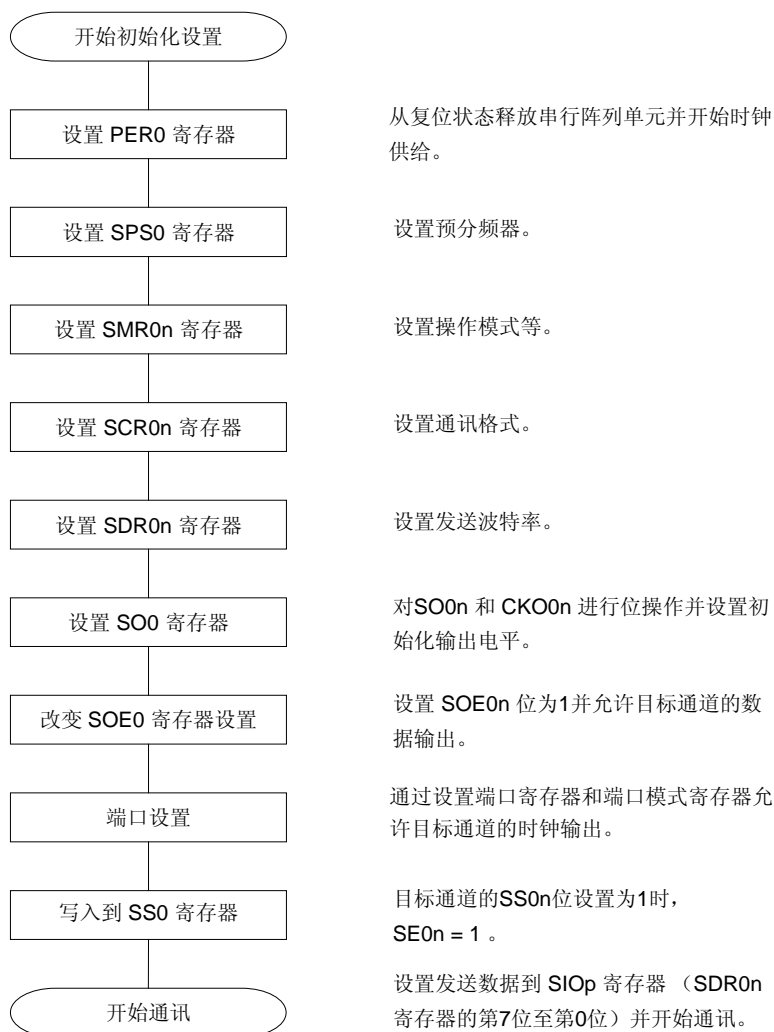
□: 在 CSI 主发送模式下固定设置, ■: 禁止设置 (设为初始值)

X: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1

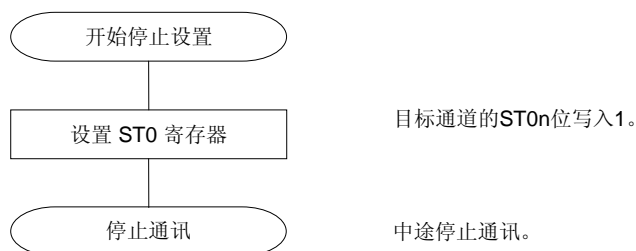
(2) 操作程序

图 12-38. 主发送/接收的初始化程序



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

图 12-39. 主发送/接收的停止程序



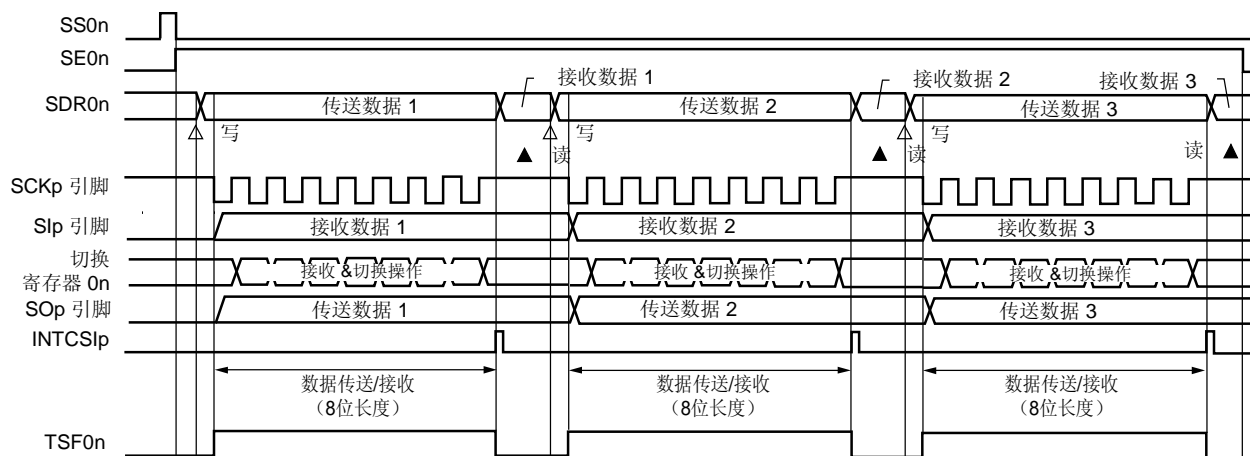
备注 即使通信停止后，引脚电平依然保持。要重新操作，重新设置 SO0 寄存器 (参见 图 12-40 恢复主发送/接收的程序)。

图 12-40. 恢复主发送/接收的程序



(3) 处理流程（在单发送/接收模式下）

图 12-41. 主发送/接收的时序图（在单发送/接收模式下）

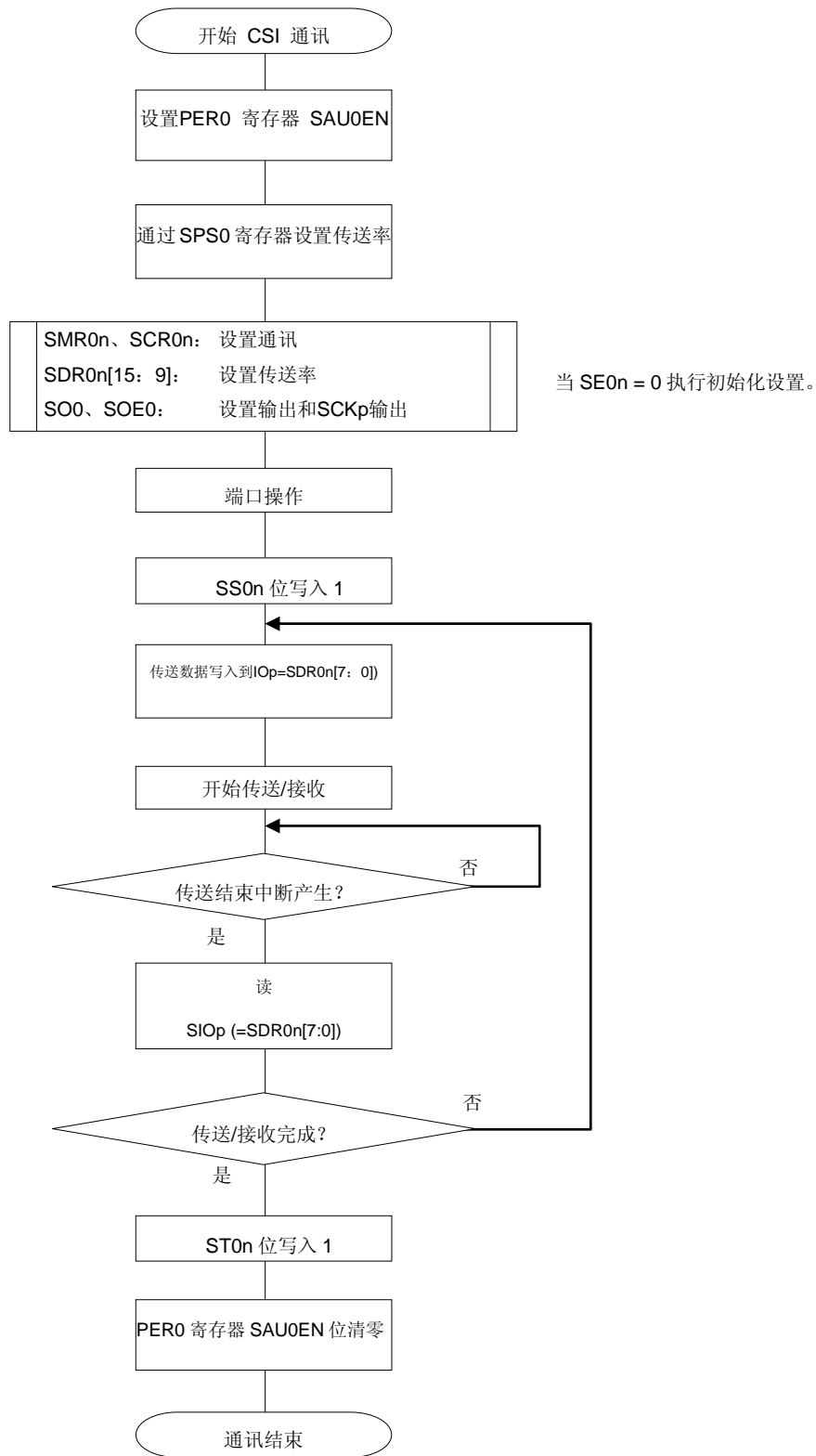


备注

n: 通道编号 (n = 0~2)

p: CSI 编号 (p = 00、01、10)

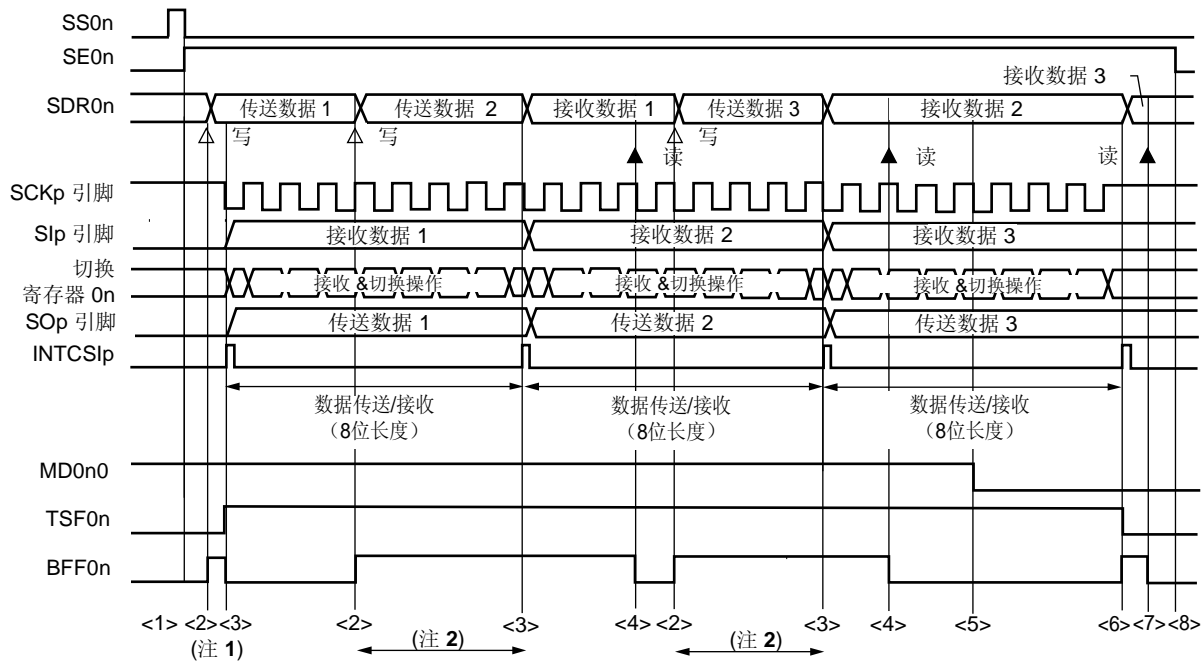
图 12-42. 主发送/接收的流程图（在单发送/接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

(4) 处理流程（在连续发送/发送模式下）

图 12-43. 主发送/发送的时序图（在连续发送/接收模式下）



注 1. 在 $BFF0n = 1$ 的情况下，当发送数据写入 SDR0n 寄存器时，发送数据可以被重写。

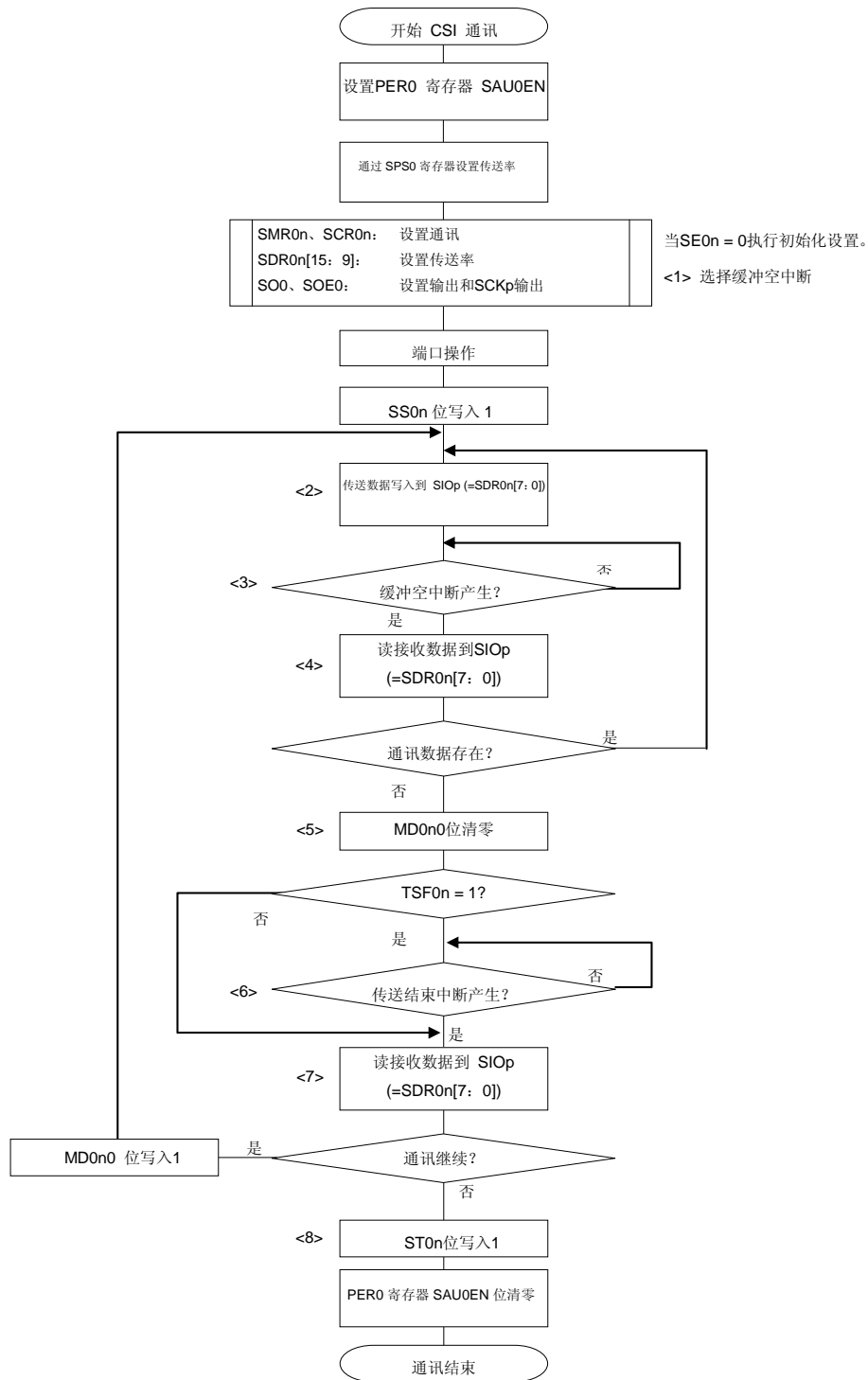
2. 在这期间，发送数据可以通过读取 SDR0n 寄存器来读取。此时发送操作不受影响。

注意事项 即使在操作过程中，MD0n0 位也可以被重写。

但是，要在最后一位开始发送之前重写它，这样它可以在最后一个发送数据的发送结束中断之前被重写。

- 备注**
1. 图中的 <1>~<8> 对应图 12-44 主发送/接收的流程图（在连续发送/接收模式下）中的 <1>~<8>。
 2. n: 通道编号 ($n = 0 \sim 2$)
p: CSI 编号 ($p = 00, 01, 10$)

图 12-44. 主发送/接收的流程图（在连续发送/接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

备注 图中的<1>~<8> 对应图 12-44 主发送/接收的时序图（在连续发送/接收模式下）中的<1>~<8>。

12.5.4 从发送

从发送是 μ PD79F9211 发送数据到另一个设备，发送时钟由通讯对象输入。

3 线串行 I/O	CSI00	CSI01	CSI10
目标通道	SAU 的通道 0	SAU 的通道 1	SAU 的通道 2
使用引脚	SCK00, SO00	SCK01, SO01	SCK10, SO10
中断	INTCSI00	INTCSI01	INTCSI10
	可以选择发送结束中断（在单发送模式下）或者缓冲空中断（在连续发送模式下）。		
错误检测标志	只有溢出错误检测标志 (OVF0n)		
发送数据长度	7 或 8 位		
<R> 发送速率	最大值 $f_{MCK}/6$ [MHz] 注 1,2		
数据相位	通过 DAP0n 位可选择 <ul style="list-style-type: none"> • DAP0n = 0: 从串行时钟开始操作时，数据输出开始。 • DAP0n = 1: 在串行时钟操作开始之前半个时钟，数据输出开始。 		
时钟相位	通过 CKP0n 位选择 <ul style="list-style-type: none"> • CKP0n = 0: 正向 • CKP0n = 1: 反向 		
数据方向	MSB 或者 LSB		

- 注 1. 因为外部串行时钟输入到引脚 SCK00、SCK01 和 SCK10 是内部采样并使用，最大发送速率为 $f_{CLK}/6$ [MHz]。
2. 请在满足上面条件和电气特性中的 AC 特性的范围内，使用这个操作（参见 第 26 章 电气特性（目标））。

- 备注 1. f_{MCK} : 目标通道的工作时钟(MCK)频率
 f_{CLK} : 系统时钟频率
2. n: 通道编号 (n=0~2)

(1) 寄存器设置

图 12-45. 用于 3 线串行 I/O (CSI00、CSI01、CSI10) 的从发送的寄存器内容的示例

(a) 串行输出寄存器 0 (SO0)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 ×	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 0/1	SO01 0/1	SO00 0/1

(b) 串行输出使能寄存器 0 (SOE0)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	SOE01 0/1	SOE00 0/1

(c) 串行通道启动寄存器 0 (SS0)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 0/1	SS00 0/1

(d) 串行模式寄存器 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 1	0	0	0	0	0	STS0n 0	0	SIS0n0 0	1	0	0	MD0n2 0	MD0n1 0	MD0n0 0/1

通道 n 的工作模式

0: 发送结束中断

1: 缓冲空中断

(e) 串行通信操作设置寄存器 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 1	RXE0n 0	DAP0n 0/1	CKP0n 0/1	0	EOC0n 0	PTC0 n10	PTC0 n00	DIR0n 0/1	0	SLC0n1 0	SLC0n0 0	0	DLS0n2 1	DLS0 n11	DLS0n0 0/1

(f) 串行数据寄存器 0n (SDR0n) (低 8 位: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	设置波特率							0	设置发送数据							

备注

n: 通道编号 (n=0~2)

p: CSI 数 (p = 00、01、10)

□: 在 CSI 主发送模式下固定设置, ■: 禁止设置 (设为初始值)

X: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1

SIOp

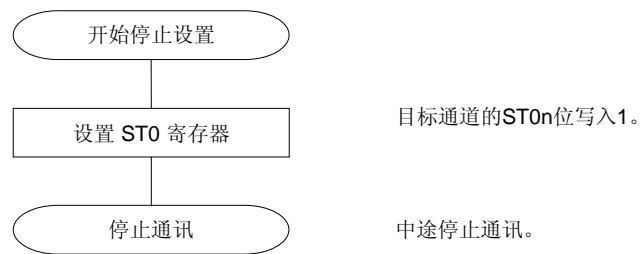
(2) 操作程序

图 12-46. 从发送的初始化程序



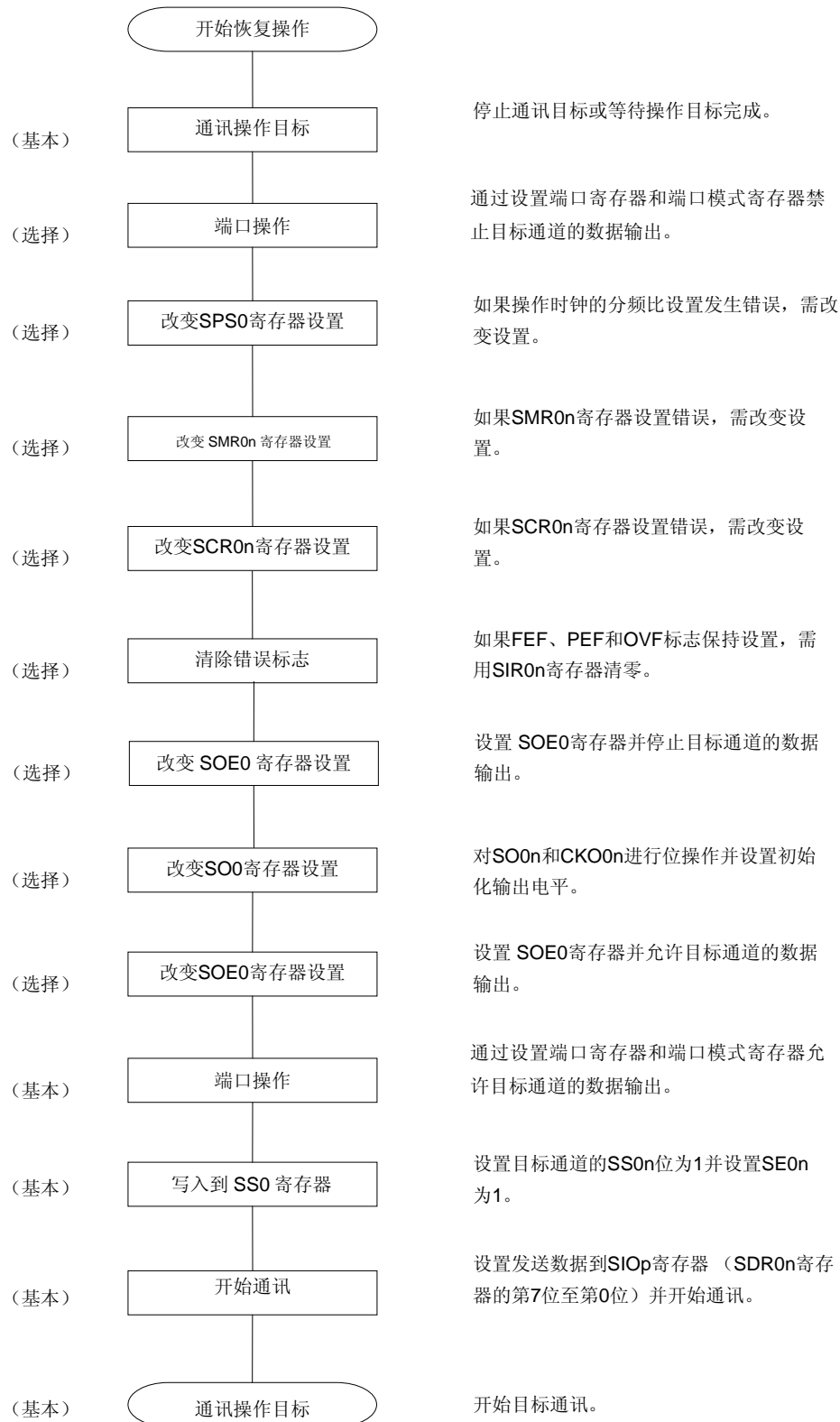
注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

图 12-47. 从发送的停止程序



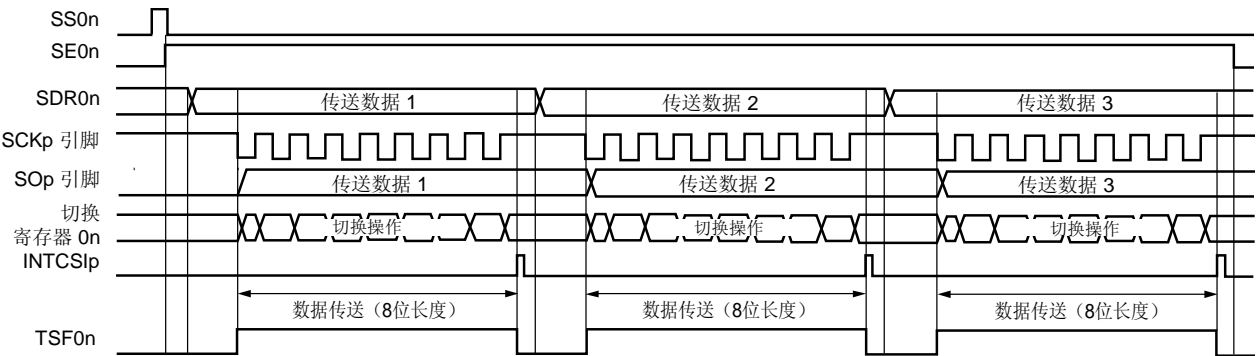
备注 即使通信停止后，引脚电平依然保持。要恢复操作，重新设置 SO0 寄存器（参见图 12-48 恢复从发送的程序）。

图 12-48. 恢复从发送的程序



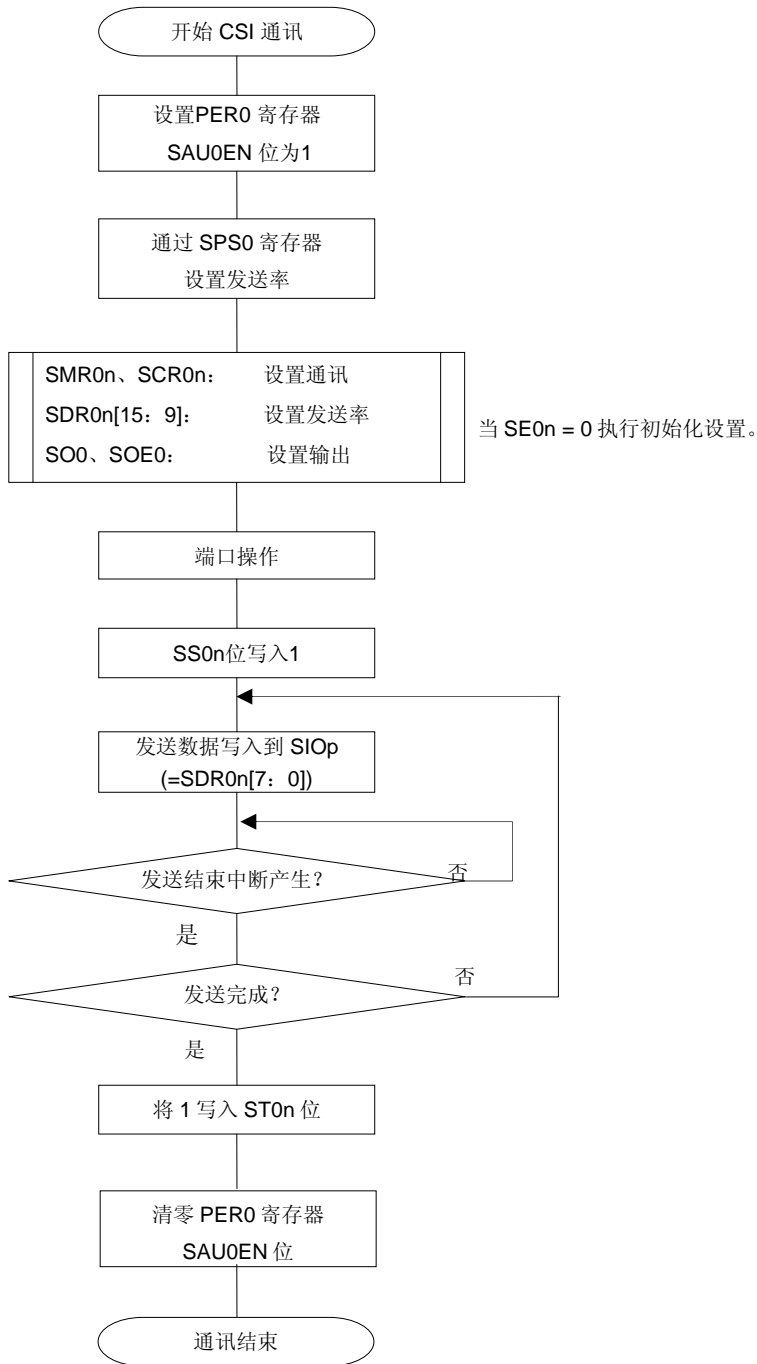
(3) 处理流程（在单发送模式下）

图 12-49. 从发送的时序图（在单发送模式下）



备注 n: 通道编号 (n = 0~2)
 p: CSI 编号 (p = 00、01、10)

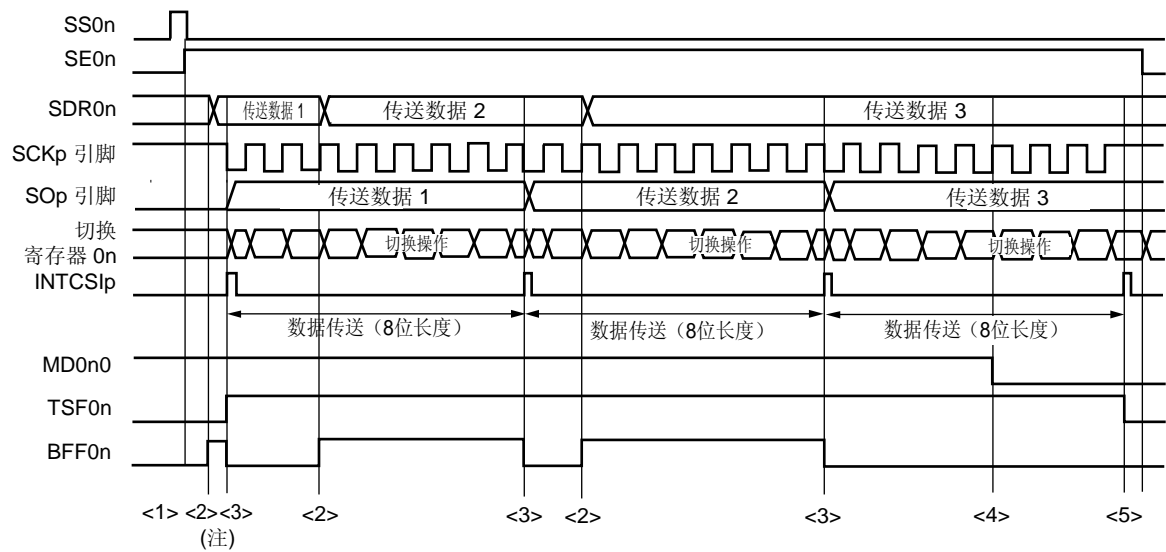
图 12-50. 从发送的流程图（在单发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

(4) 处理流程（在连续发送模式下）

图 12-51. 从发送的时序图（在连续发送模式下）

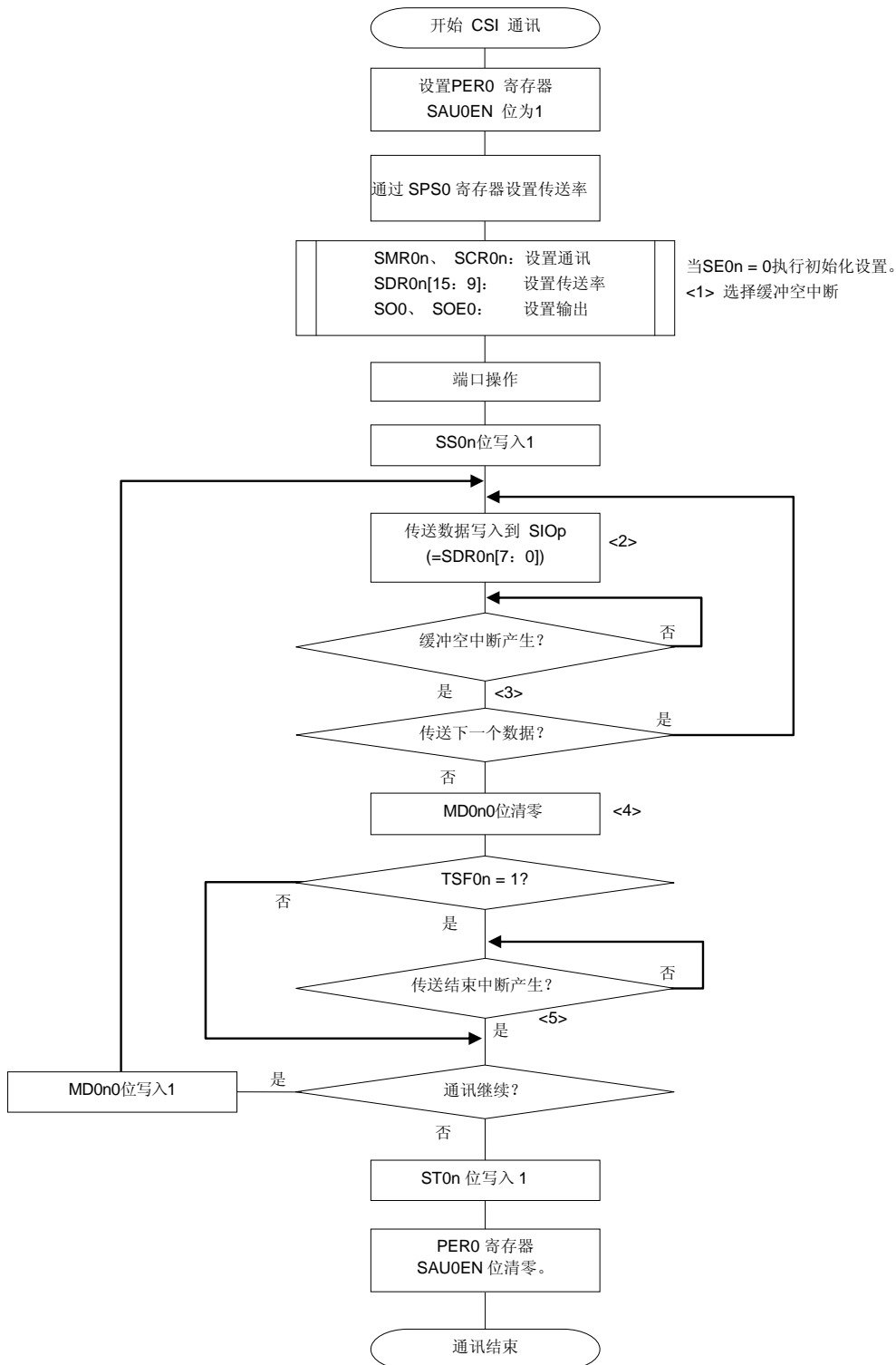


注 在 BFF0n = 1 的情况下，当发送数据写入 SDR0n 寄存器时，发送数据可以被重写。

注意事项 即使在操作过程中，MD0n0 位也可以被重写。但是必须在最后一位开始发送之前重写它。

备注 n: 通道编号 (n = 0~2)
p: CSI 编号 (p = 00、01、10)

图 12-52. 从发送的流程图（在连续发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

备注 图中的<1>~<5> 对应图 12-52 从发送的时序图（在连续发送模式下）中的<1>~<5>。

12.5.5 从接收

从接收是 μ PD79F9211 接收另一个设备的数据，发送时钟由通讯对象输入。

3 线串行 I/O	CSI00	CSI01	CSI10
目标通道	SAU 的通道 0	SAU 的通道 1	SAU 的通道 2
使用引脚	SCK00, SI00	SCK01, SI01	SCK10, SI10
中断	INTCSI00	INTCSI01	INTCSI10
	仅发送结束中断（禁止设置缓冲空中断。）		
错误检测标志	只有溢出错误检测标志 (OVF0n)		
发送数据长度	7 或 8 位		
发送速率	最大值 $f_{MCK}/6$ [MHz] ^{注 1,2}		
数据相位	通过 DAP0n 位可选择 <ul style="list-style-type: none"> • DAP0n = 0: 从串行时钟开始操作时,数据输入开始。 • DAP0n = 1: 在串行时钟操作开始之前半个时钟，数据输入开始。 		
时钟相位	通过 CKP0n 位选择 <ul style="list-style-type: none"> • CKP0n = 0: 正向 • CKP0n = 1: 反向 		
数据方向	MSB 或者 LSB		

- 注**
1. 因为外部串行时钟输入到引脚 SCK00、SCK01 和 SCK10 是内部采样并使用，最大发送速率为 $f_{CLK}/6$ [MHz]。
 2. 请在满足上面条件和电气特性中的 AC 特性的范围内，使用这个操作（参见 第 26 章 电气特性（目标））。

- 备注**
1. f_{MCK} : 目标通道的工作时钟(MCK)频率
n: 通道编号 (n=0~2)

(1) 寄存器设置

图 12-53. 用于 3 线串行 I/O (CSI00、CSI01、CSI10) 的从接收的寄存器内容的示例

(a) 串行输出寄存器 0 (SO0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 ×	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 ×	SO01 ×	SO00 ×

(b) 串行输出使能寄存器 0 (SOE0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	SOE01 0/1	SOE00 0/1

当用于 CSI 从接收时，设置这些位为 0。

(c) 串行通道启动寄存器 0 (SS0)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 0/1	SS00 0/1

(d) 串行模式寄存器 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 1	0	0	0	0	0	STS0n 0	0	SIS0n0 0	1	0	0	MD0n2 0	MD0n1 0	MD0n0 0

通道 n 的工作模式
0: 发送结束中断

(e) 串行通信操作设置寄存器 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 0	RXE0n 1	DAP0n 0/1	CKP0n 0/1	0	EOC0n 0	PTC0 n10	PTC0 n00	DIR0n 0/1	0	SLC0n1 0	SLC0n0 0	0	DLS0n2 1	DLS0 n11	DLS0n0 0/1

(f) 串行数据寄存器 0n (SDR0n) (低 8 位: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	0000000 (设置波特率)								0	接收数据寄存器						

备

n: 通道编号 (n=0~2)

p: CSI 数 (p = 00、01、10)

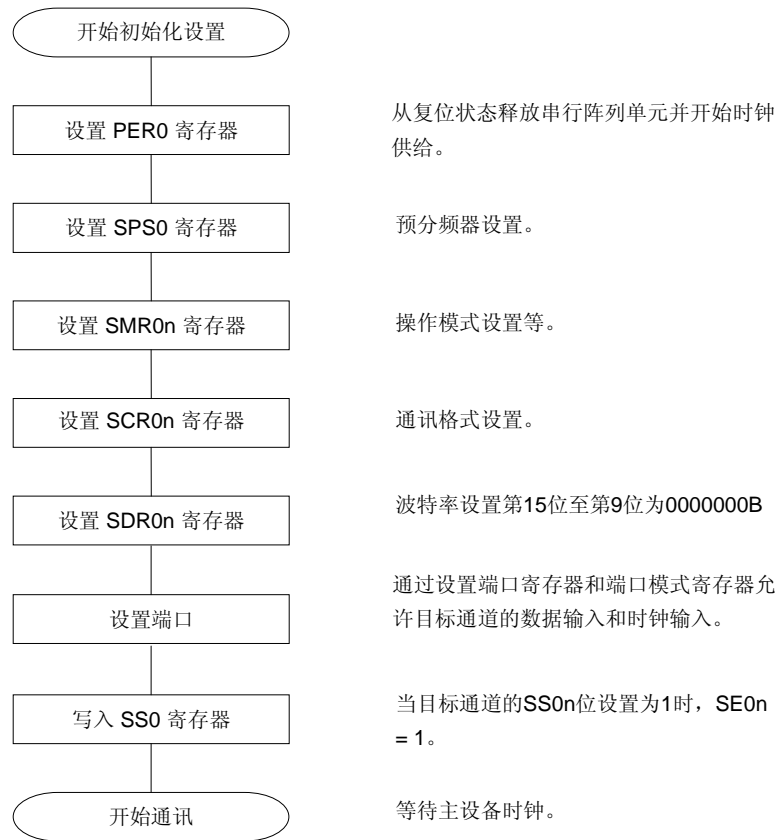
□: 在 CSI 主发送模式下固定设置, ■: 禁止设置 (设为初始值)

X: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1

(2) 操作程序

图 12-54. 从接收的初始化程序

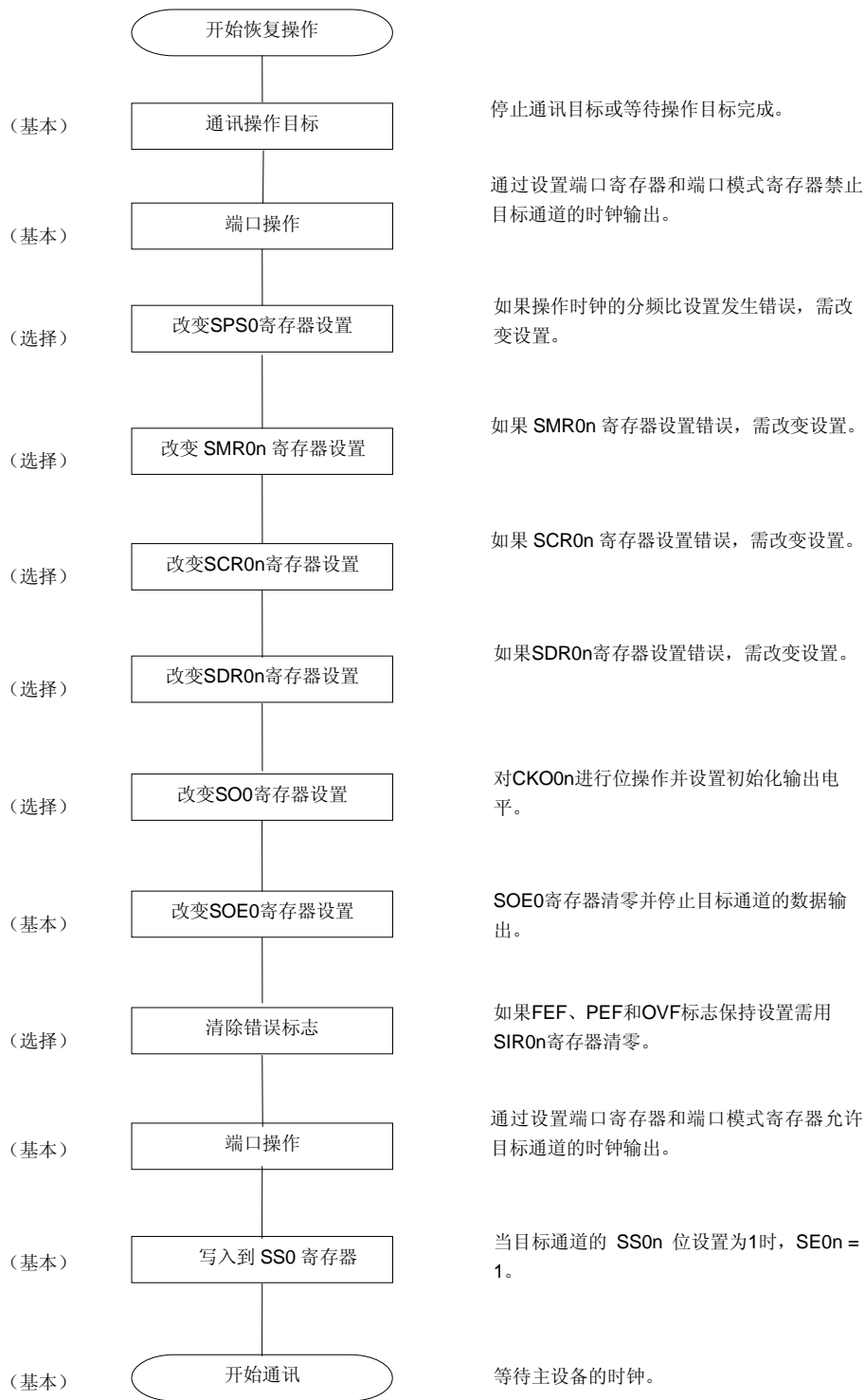


意事项在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

图 12-55. 从接收的停止程序

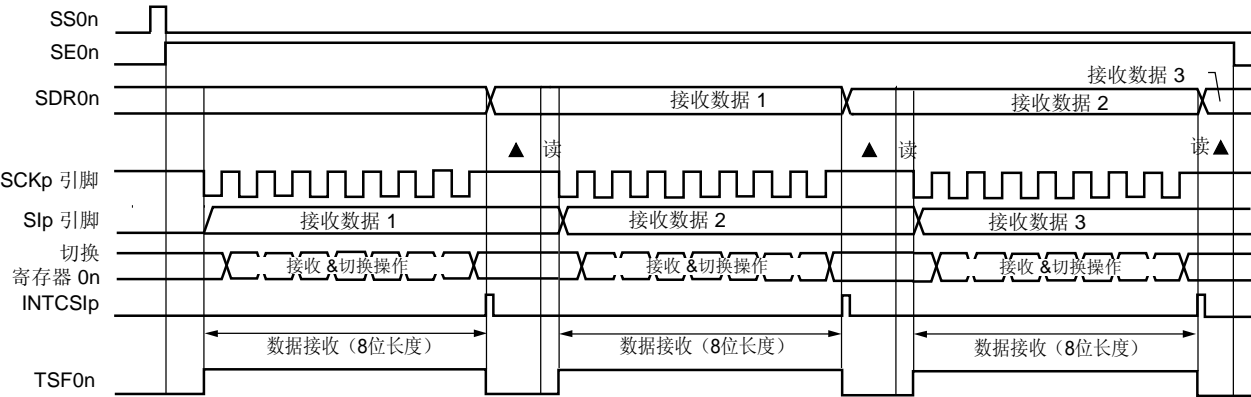


图 12-56. 恢复从接收的程序



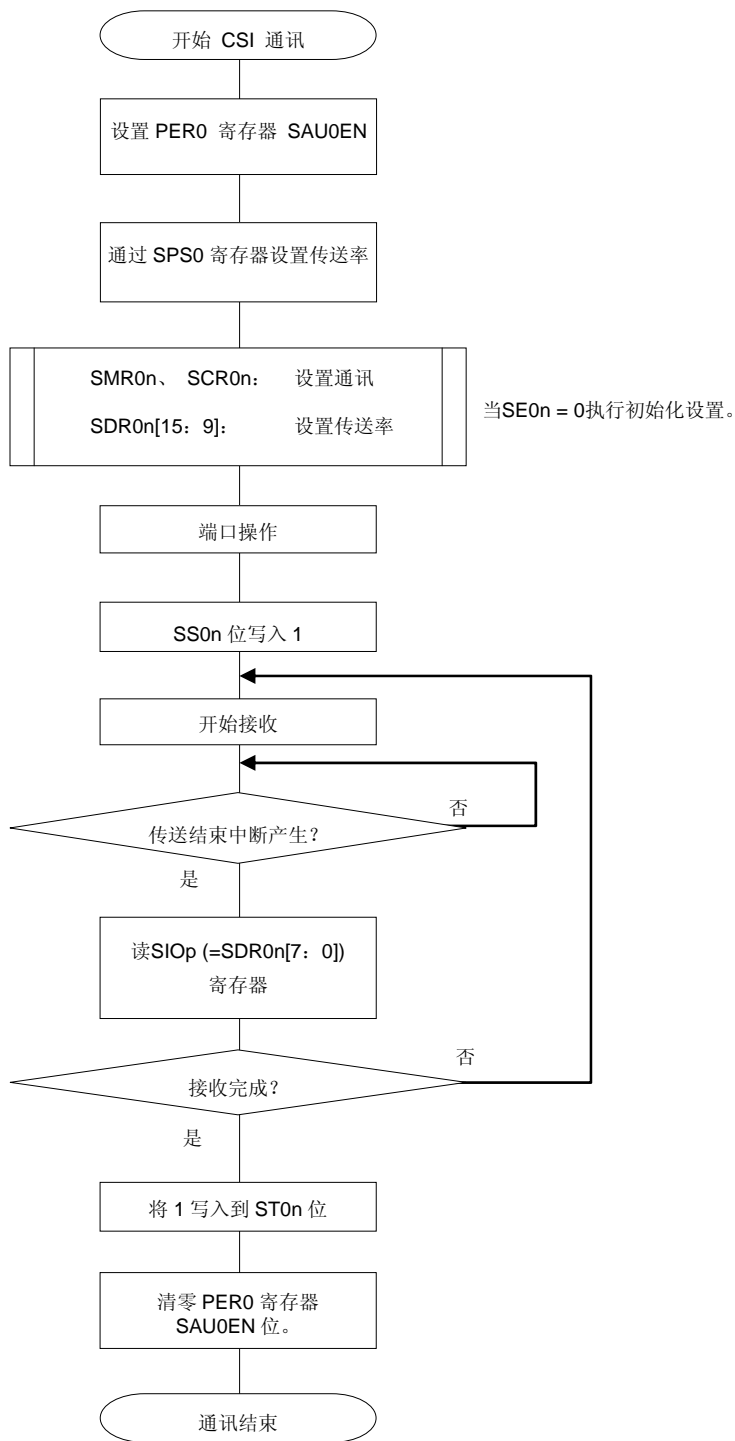
(3) 处理流程（在单接收模式下）

图 12-57. 从接收的时序图（在单接收模式下）



备注 n: 通道编号 (n = 0~2)
 p: CSI 编号 (p = 00、01、10)

图 12-58. 从接收的流程图 (在单接收模式下)



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

12.5.6 从发送/接收

从发送/接收是μPD79F9211 在从另外一个设备输入一个发送时钟状态下发送数据到另一个设备/从另一个设备接收数据。

<R>

3 线串行 I/O	CSI00	CSI01	CSI10
目标通道	SAU 的通道 0	SAU 的通道 1	SAU 的通道 2
使用引脚	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10
中断	INTCSI00	INTCSI01	INTCSI10
	可以选择发送结束中断（在单发送模式下）或者缓冲空中断（在连续发送模式下）		
错误检测标志	只有溢出错误检测标志 (OVF0n)		
发送数据长度	7 或 8 位		
发送速率	最大值 f _{MCK} /6 [MHz] ^{注 1,2}		
数据相位	通过 DAP0n 位可选择 <ul style="list-style-type: none">• DAP0n = 0: 从串行时钟操作开始时,数据 I/O 开始。• DAP0n = 1: 在串行时钟操作开始之前半个时钟, 数据 I/O 开始。		
时钟相位	通过 CKP0n 位选择 <ul style="list-style-type: none">• CKP0n = 0: 正向• CKP0n = 1: 反向		
数据方向	MSB 或者 LSB		

- 注 1. 因为外部串行时钟输入到引脚 SCK00、SCK01 和 SCK10 是内部采样并使用, 最大发送速率为 f_{CLK}/6 [MHz]。
2. 请在满足上面条件和电气特性中的 AC 特性的范围内, 使用这个操作（参见 第 26 章 电气特性（目标））。

备注 1. f_{MCK}: 目标通道的工作时钟(MCK)频率

(1) 寄存器设置

图 12-59. 用于 3 线串行 I/O (CSI00、CSI01、CSI10) 的从发送/接收的寄存器内容的示例

(a) 串行输出寄存器 0 (SO0)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 ×	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 0/1	SO01 0/1	SO00 0/1

(b) 串行输出使能寄存器 0 (SOE0)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	SOE01 0/1	SOE00 0/1

(c) 串行通道启动寄存器 0 (SS0)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 0/1	SS00 0/1

(d) 串行模式寄存器 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 1	0	0	0	0	0	STS0n 0	0	SIS0n0	1	0	0	MD0n2 0	MD0n1 0	MD0n0 0/1

通道 n 的工作模式

0: 发送结束中断

1: 缓冲空中断

(e) 串行通信操作设置寄存器 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 1	RXE0n 1	DAP0n 0/1	CKP0n 0/1	0	EOC0n 0	PTC0 n10	PTC0 n00	DIR0n 0/1	0	SLC0n1 0	SLC0n0 0	0	DLS0n2 1	DLS0 n11	DLS0n0 0/1

(f) 串行数据寄存器 0n (SDR0n) (低 8 位: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	0000000 (设置波特率)							0	设置发送数据/接收数据寄存器							

SIOp

备注

n: 通道编号 (n=0~2)

p: CSI 数 (p = 00、01、10)

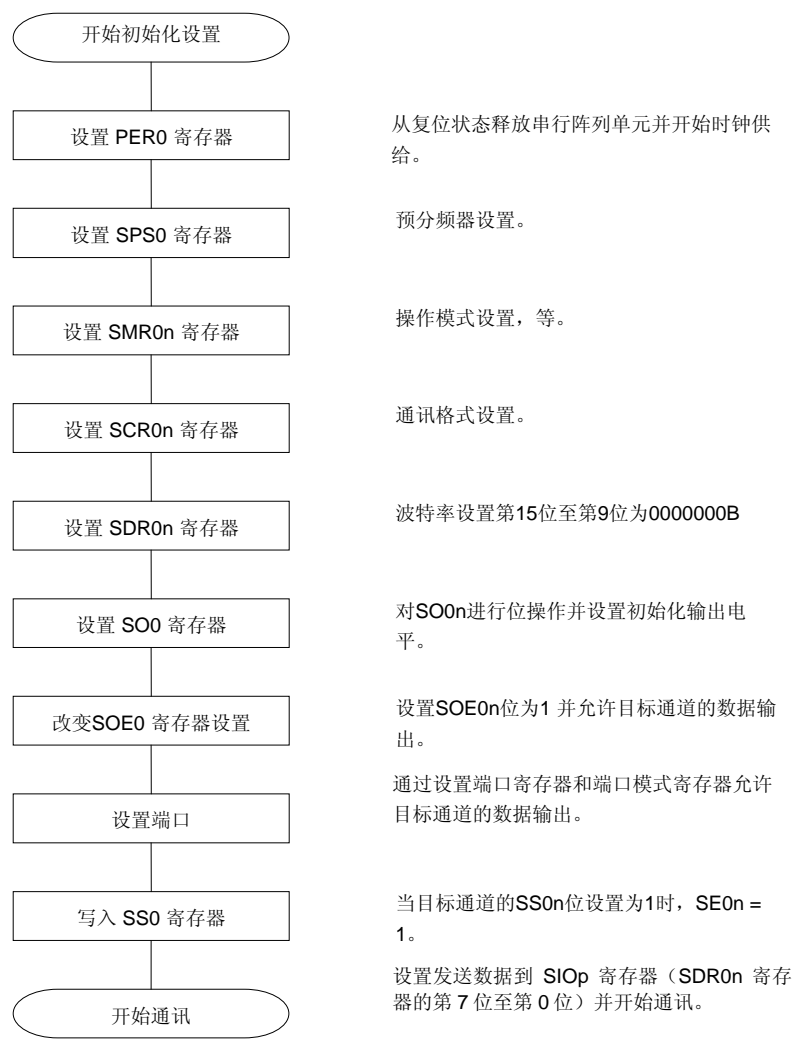
□: 在 CSI 主发送模式下固定设置, ■: 禁止设置 (设为初始值)

X: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1

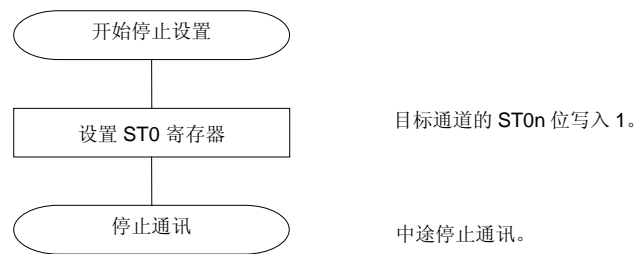
(2) 操作程序

图 12-60. 从发送/接收的初始化程序



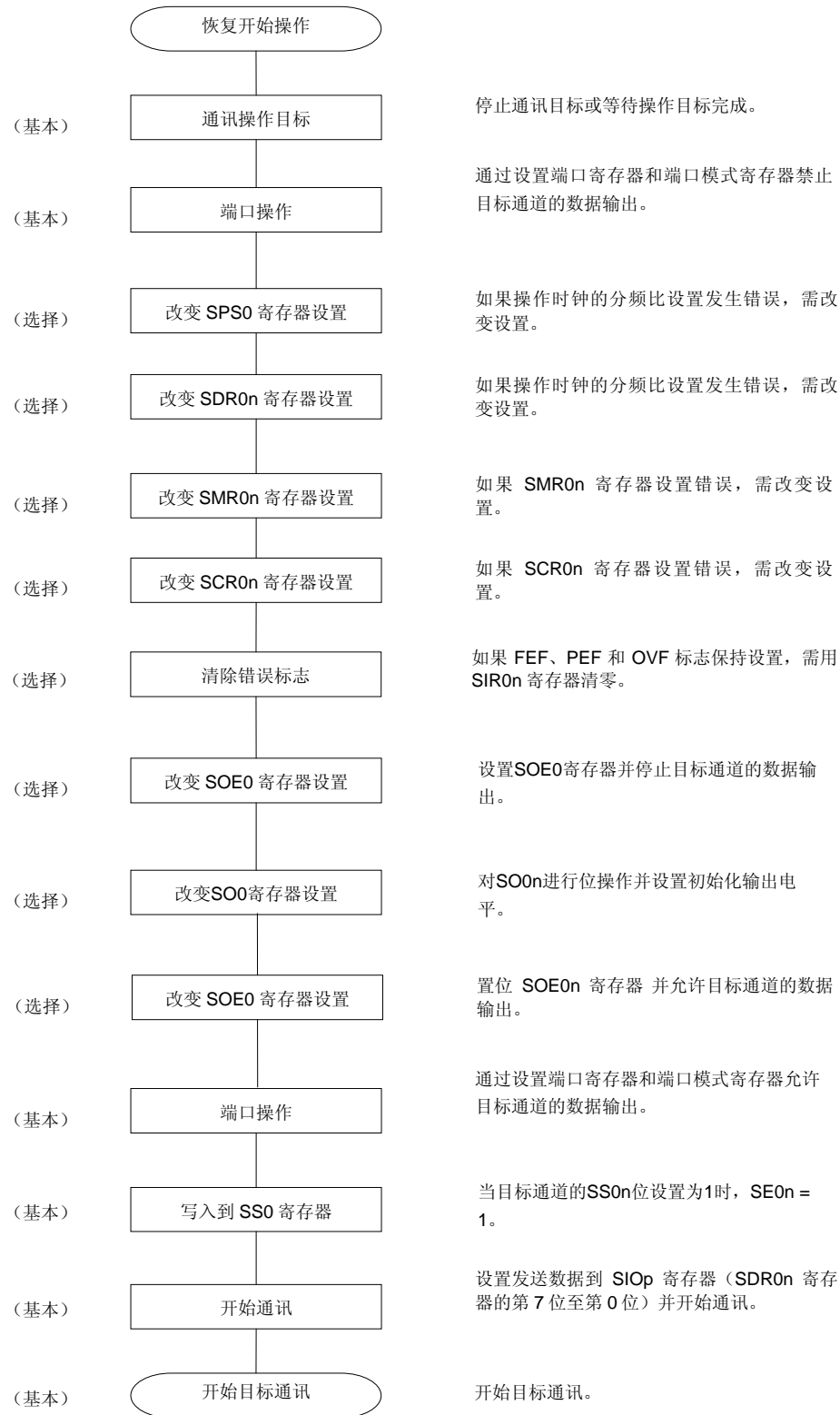
注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

图 12-61. 从发送/接收的停止程序



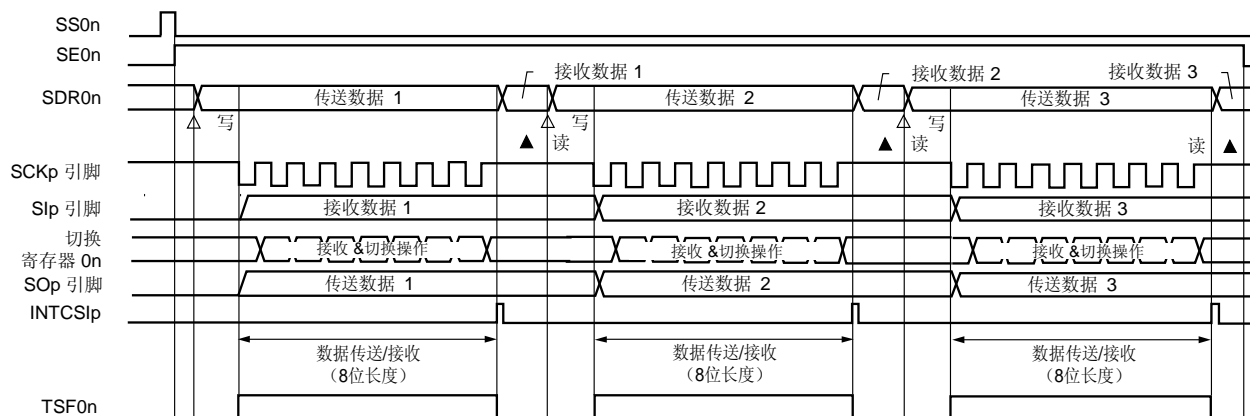
备注 即使通信停止后，引脚电平依然保持。要恢复操作，重新设置 SO0 寄存器（参见图 12-62 恢复从发送/接收的程序）。

图 12-62. 恢复从发送/接收的程序



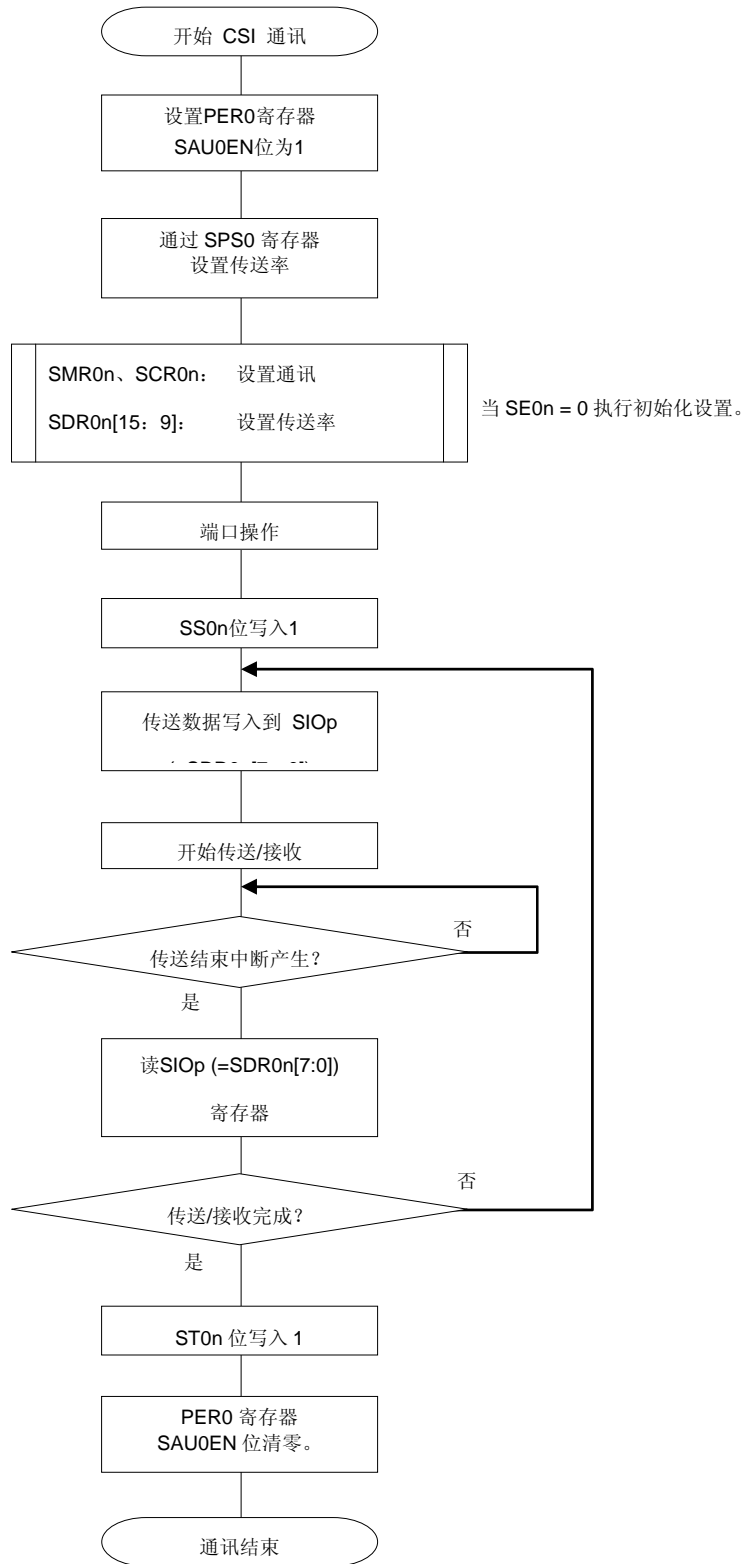
(3) 处理流程（在单发送/接收模式下）

图 12-63. 从发送/接收的时序图（在单发送/接收模式下）



备注 n: 通道编号 (n = 0~2)
p: CSI 编号 (p = 00、01、10)

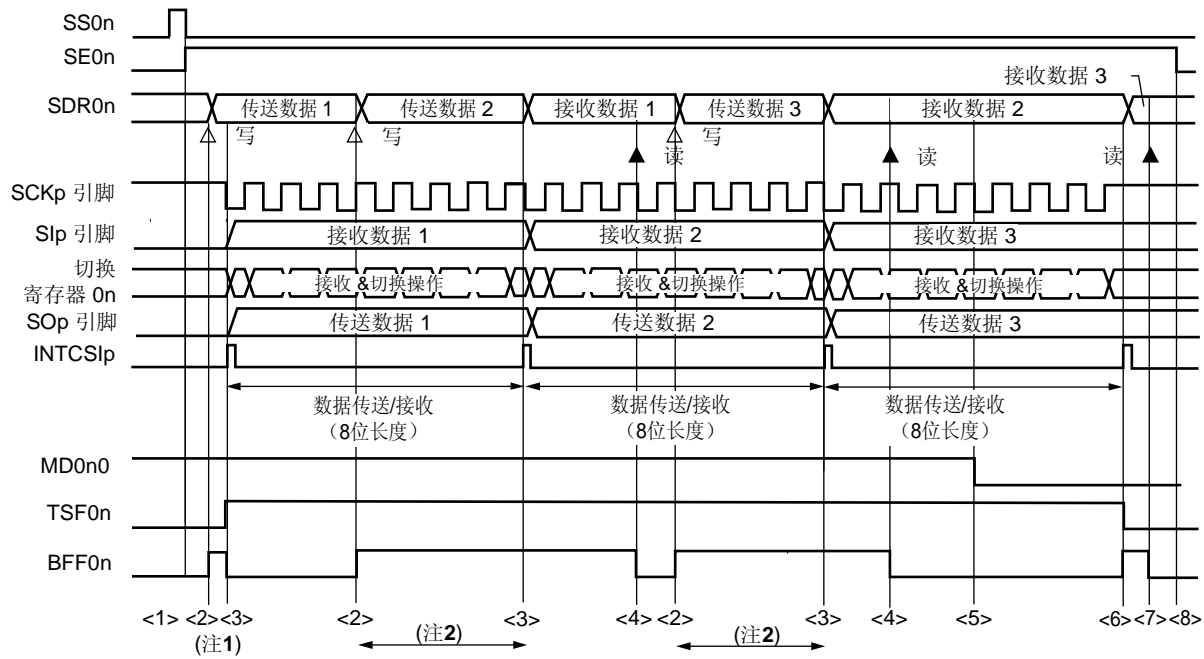
图 12-64. 从发送/接收的流程图（在单发送/接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

(4) 处理流程（在连续发送/发送模式下）

图 12-65. 从发送/发送的时序图（在连续发送/发送模式下）



注 1. 在 BFF0n = 1 的情况下，当发送数据写入 SDR0n 寄存器时，发送数据可以被重写。

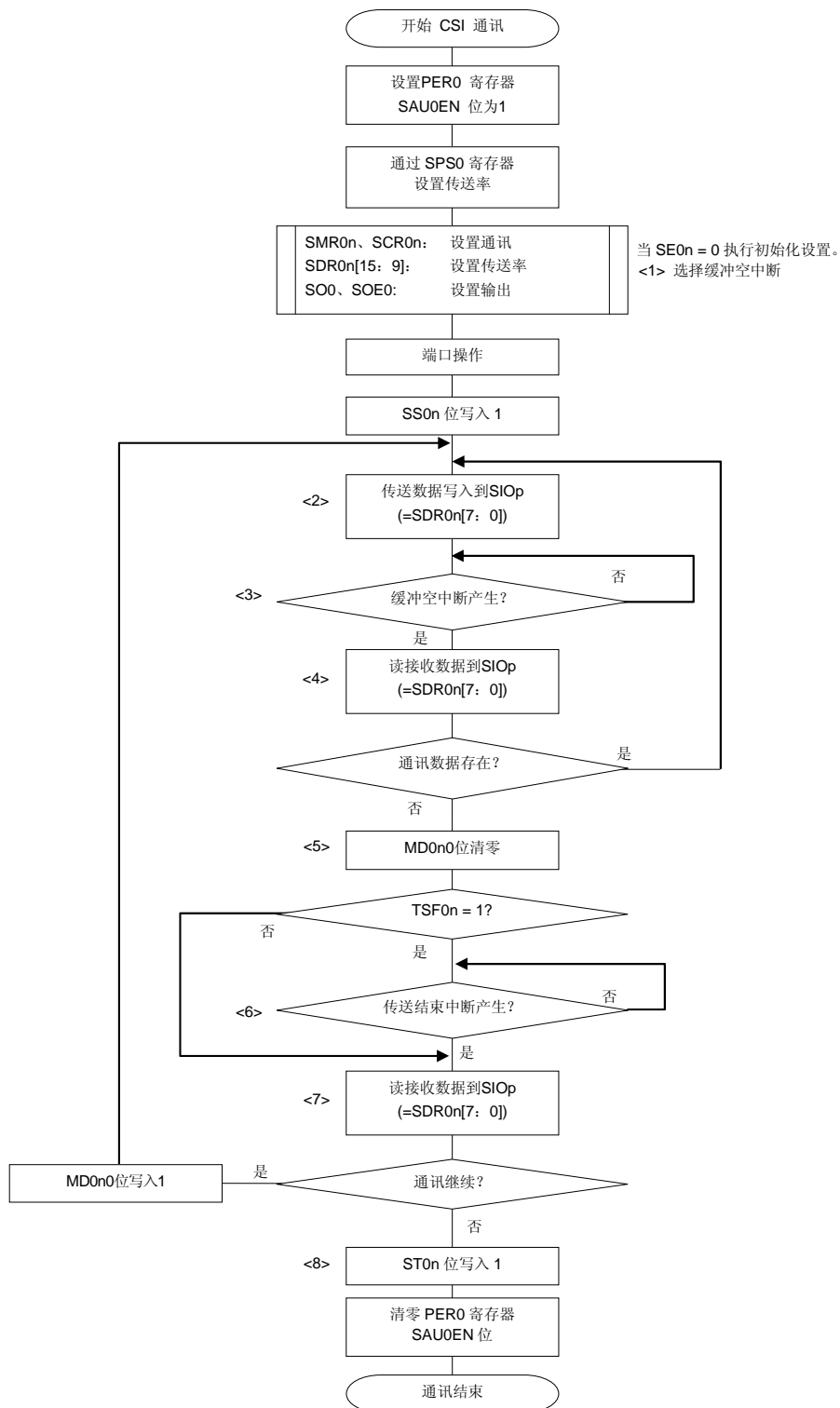
2. 在这期间，发送数据可以通过读取 SDR0n 寄存器来读取。此时发送操作不受影响。

注意事项 即使在操作过程中，MD0n0 位也可以被重写。

但是，要在最后一位开始发送之前重写它，这样它可以在最后一个发送数据的发送结束中断之前被重写。

- 备注**
1. 图中的<1>~ <8> 对应图 12-66 从发送/接收的流程图（在连续发送/接收模式下）中的<1>~ <8>。
 2. n: 通道编号 (n = 0~2)
p: CSI 编号 (p = 00、01、10)

图 12-66. 从发送/接收的流程图（在连续发送/接收模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

备注 图中的<1> ~<8> 对应图 12-64 从发送/接收的时序图（在连续发送/接收模式下）中的<1>~<8>。

12.5.7 计算发送时钟频率

用于 3 线串行 I/O（CSI00、CSI01、CSI10）通信的发送时钟频率，可以通过下面公式计算。

(1) 主

$$(\text{发送时钟频率}) = \{ \text{目标通道的工作时钟(MCK)频率} \} \div (\text{SDR0n} [15:9] + 1) \div 2 [\text{Hz}]$$

(2) 从

$$(\text{发送时钟频率}) = \{ \text{从主提供的串行时钟 (SCK) 频率} \}^{\#} [\text{Hz}]$$

<R> 注 允许的最大发送时钟频率是 $f_{\text{MCK}} / 6$ [MHz] 。

备注 1. SDR0n[15:9]的值是 SDR0n 寄存器的第 15 位～第 9 位的值（0000000B～1111111B），因此它的取值范围为 0～127。

2. n: 通道编号（n=0～2）

工作时钟（MCK）由串行时钟选择寄存器 0 (SPS0)和串行模式寄存器 0n (SMR0n)的第 15 位（CKS0n）来确定。

表 12-2. 工作时钟的选择

SMR0n 寄存器	SPS0 寄存器								工作时钟(MCK) ^{※1}	
CKS0n	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000		f _{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f _{CLK}	20 MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	10 MHz
	X	X	X	X	0	0	1	0	f _{CLK} /22	5 MHz
	X	X	X	X	0	0	1	1	f _{CLK} /23	2.5 MHz
	X	X	X	X	0	1	0	0	f _{CLK} /24	1.25 MHz
	X	X	X	X	0	1	0	1	f _{CLK} /25	625 kHz
	X	X	X	X	0	1	1	0	f _{CLK} /26	313 kHz
	X	X	X	X	0	1	1	1	f _{CLK} /27	156 kHz
	X	X	X	X	1	0	0	0	f _{CLK} /28	78.1 kHz
	X	X	X	X	1	0	0	1	f _{CLK} /29	39.1 kHz
	X	X	X	X	1	0	1	0	f _{CLK} /210	19.5 kHz
	X	X	X	X	1	0	1	1	f _{CLK} /211	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 ^{※2}	
1	0	0	0	0	X	X	X	X	f _{CLK}	20 MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	10 MHz
	0	0	1	0	X	X	X	X	f _{CLK} /22	5 MHz
	0	0	1	1	X	X	X	X	f _{CLK} /23	2.5 MHz
	0	1	0	0	X	X	X	X	f _{CLK} /24	1.25 MHz
	0	1	0	1	X	X	X	X	f _{CLK} /25	625 kHz
	0	1	1	0	X	X	X	X	f _{CLK} /26	313 kHz
	0	1	1	1	X	X	X	X	f _{CLK} /27	156 kHz
	1	0	0	0	X	X	X	X	f _{CLK} /28	78.1 kHz
	1	0	0	1	X	X	X	X	f _{CLK} /29	39.1 kHz
	1	0	1	0	X	X	X	X	f _{CLK} /210	19.5 kHz
	1	0	1	1	X	X	X	X	f _{CLK} /211	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 ^{※2}	
其它值									设置禁止	

- 注
1. 当改变用于 f_{CLK} 的时钟（通过改变系统时钟控制寄存器 (CKC) 的值）时，在停止串行阵列单元 (SAU) 的操作 (ST0 = 000FH) 之后，才这样做。当选择 INTTM02 为工作时钟时，同样停止定时器阵列单元 TAUS (TT0 = 00FFH)。
 2. 利用 SPS0 寄存器，通过设置 TAUS 的 TIS0 寄存器的 TIS02 位为 1，为输入时钟选择 f_{SUB}/4，并且选择 INTTM02，可以在副系统时钟的一个固定分频比时操作 SAU，无论 f_{CLK} 频率（主系统时钟、副系统时钟）如何。但是，当改变 f_{CLK}，必须按照上面注 1 描述的停止 SAU 和 TAUS。

- 备注
1. X: 不必考虑
 2. n: 通道编号 (n=0~2)

12.6 UART (UART0, UART1) 通信的操作

使用 2 条线的起始停止同步功能：串行数据发送(TxD)和串行数据接收(RxD)线。它和通信部分（通过使用内部波特率）同步发送或者接收数据。全双工 UART 通信可以被看作是使用两个通道，一个专门发送（偶通道），一个专门接收（奇通道）。

[数据发送/接收]

- 5, 7, 或者 8 位的数据长度
- 选择 MSB/LSB
- 发送/接收数据的电平设置和反向选择
- 附加校验位和校验检测功能
- 附加停止位

[中断功能]

- 发送结束中断/缓冲空中断
- 帧错误、校验错误、或者溢出错误时产生错误中断

[错误检测标志]

- 帧错误、校验错误、或者溢出错误

UART0（单元的 0 和 1 通道）支持 LIN 总线

[LIN 总线功能]

- 唤醒信号检测
- 同步中断域 (SBF)检测
- 同步域测量、波特率测量

} 使用外部中断 (INTP0)或定时器阵列单元 TAUS。

UART0 使用 SAU 的通道 0 和 1。

UART1 使用 SAU 的通道 2 和 3。

通道	用作 CSI	用作 UART	用作简易 I ² C
0	CSI00	UART0（支持 LIN 总线）	—
1	CSI01		—
2	CSI10	UART1	IIC10
3	—		—

UART 执行下面 4 种通信操作。

- UART 发送 (参见 12.6.1.)
- UART 接收 (参见 12.6.2.)
- LIN 发送（仅 UART0） (参见 12.6.3.)
- LIN 接收（仅 UART0） (参见 12.6.4.)

12.6.1 UART 发送

UART 发送是 μ PD79F9211 从发送数据到另外一个设备的异步操作（异步起始停止）

UART 使用的两个通道，偶通道用于 UART 发送。

UART	UART0	UART1
目标通道	SAU 的通道 0	SAU 的通道 2
使用引脚	TxD0	TxD1
中断	INTST0	INTST1
	可以选择发送结束中断（在单发送模式下）或者缓冲空中断（在连续发送模式下）	
错误检测标志	无	
发送数据长度	5, 7, 或 8 位	
发送速率	$\text{Max } f_{\text{MCK}}/6 \text{ [bps]} \text{ (SDR0n [15:9] = 2 或更大), Min } f_{\text{CLK}}/(2 \times 2^{11} \times 128) \text{ [bps]}^{\#}$	
数据相位	正向输出（默认：高电平） 反向输出（默认：低电平）	
校验位	以下可选 <ul style="list-style-type: none"> • 无校验位 • 附加 0 校验 • 附加偶校验 • 附加奇校验 	
停止位	以下可选 <ul style="list-style-type: none"> • 附加 1 位 • 附加 2 位 	
数据方向	MSB 或者 LSB	

注 请在满足上面条件和电气特性中的 AC 特性的范围内，使用这个操作（参见 第 26 章 电气特性（目标））。

备注

1. f_{MCK} : 目标通道的工作时钟（MCK）频率
 f_{CLK} : 系统时钟频率
2. n : 通道编号（ $n = 0, 2$ ）

(1) 寄存器设置

图 12-67. 用于 UART (UART0, UART1) 的 UART 发送的寄存器内容的示例 (1/2)

(a) 串行输出寄存器 0 (SO0)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 ×	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 0/1 ^注	SO01 ×	SO00 0/1 ^注

(b) 串行输出使能寄存器 0 (SOE0)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	SOE01 ×	SOE00 0/1

(c) 串行通道启动寄存器 0 (SS0)...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 0/1

(d) 串行输出电平寄存器 0 (SOL0)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL02 0/1	0	SOL00 0/1

0: 正向（正常）发送
1: 反向发送

(e) 串行模式寄存器 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 0	0	0	0	0	0	STS0n 0	0	SIS0n0 0	1	0	0	MD0n2 0	MD0n1 1	MD0n0 0/1

通道 n 的工作模式

0: 发送结束中断

1: 缓冲空中断

注 在启动发送前，确保当目标通道的 SOL0n 位被设置为 0 时，该位设置为 1，当目标通道的 SOL0n 位被设置为 1 时，该位设置为 0。这个值根据通信期间数据的变化而变化。

备注 n: 通道数 (n = 0, 2)

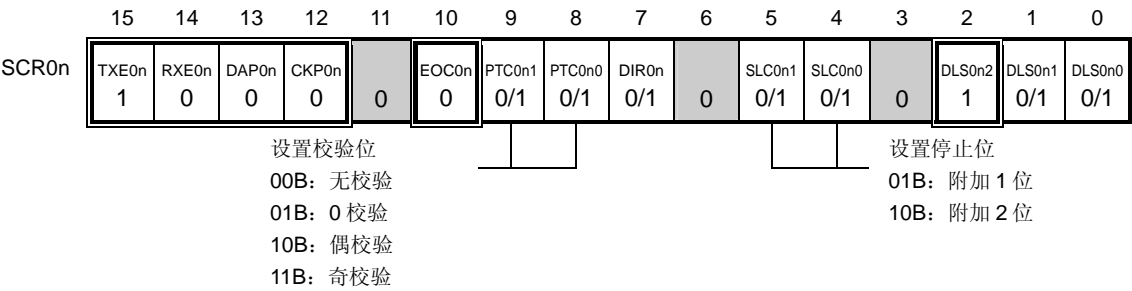
□: 在 UART 发送模式下固定设置，■: 禁止设置（由硬件给出固定值）

×: 此模式下不能使用的位（不在任何模式下使用时，设为初始值）

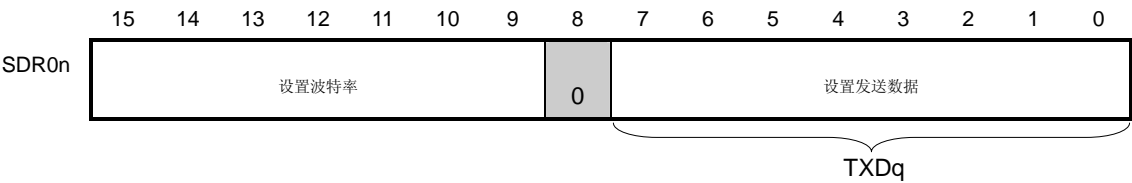
0/1: 根据用户的使用设置为 0 或 1

图 12-67. 用于 UART (UART0, UART1) 的 UART 发送的寄存器内容的示例 (2/2)

(f) 串行通信操作设置寄存器 0n (SCR0n)



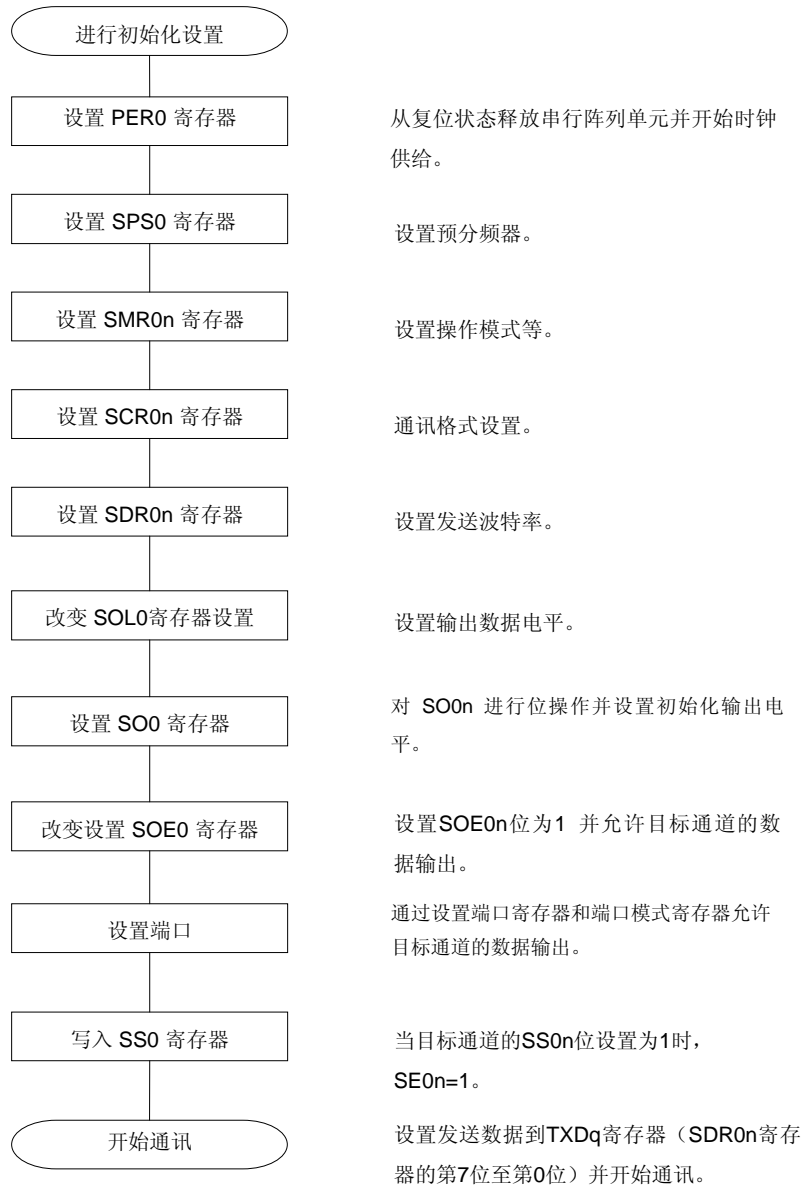
(g) 串行数据寄存器 0n (SDR0n) (低 8 位: TXDq)



备注 n: 通道数 (n = 0, 2), q: UART 数 (q = 0, 1)
□: 在 UART 发送模式下固定设置, ■: 禁止设置 (设为初始值)
0/1: 根据用户的使用设置为 0 或 1

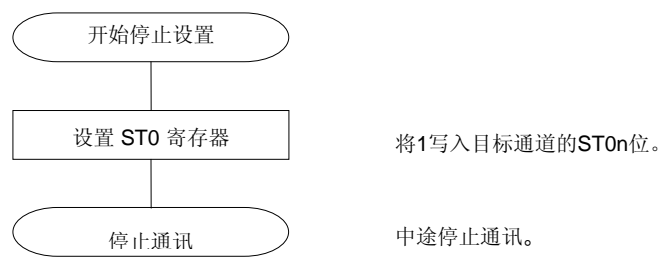
(2) 操作程序

图 12-68. UART 发送的初始化程序



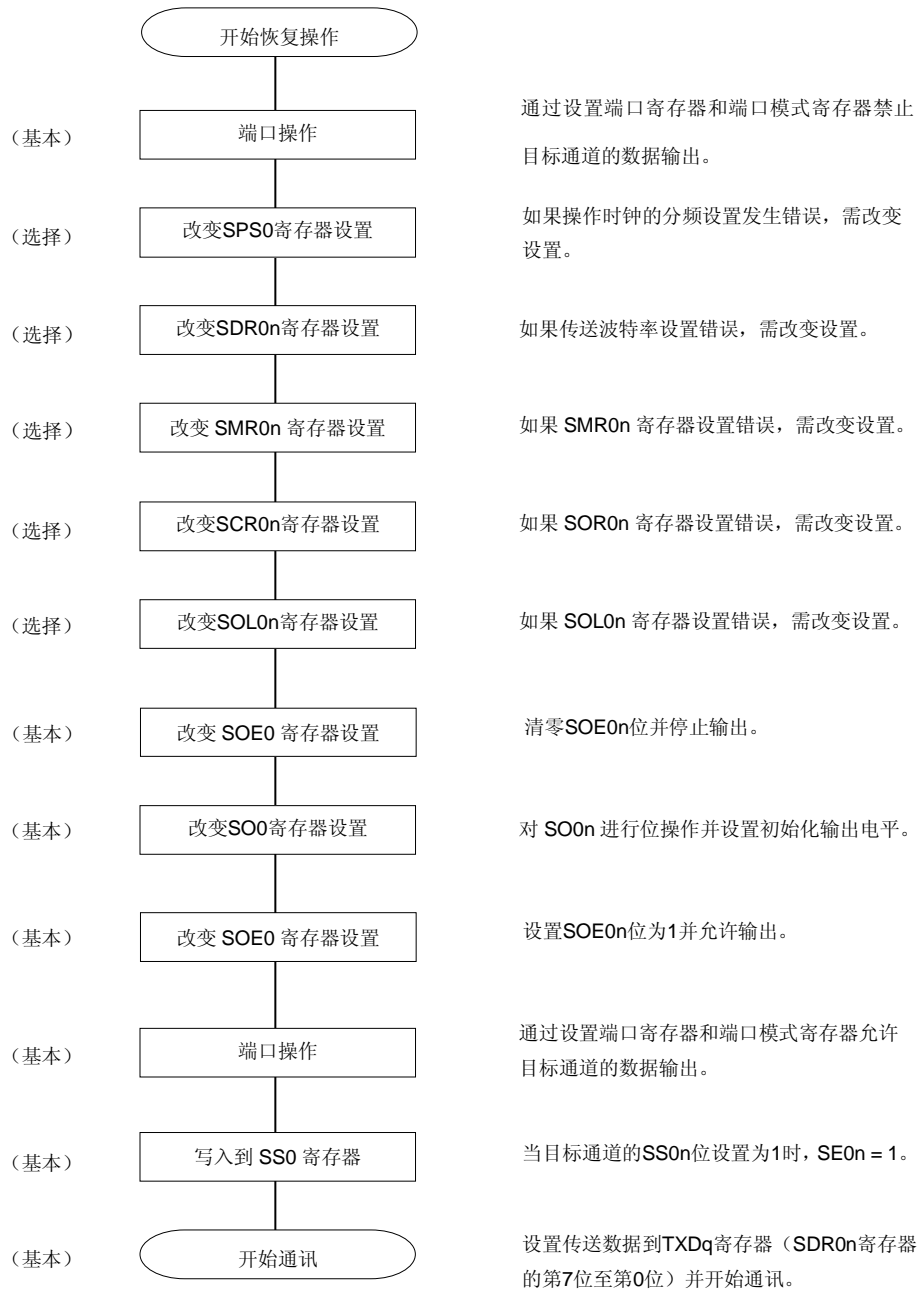
注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

图 12-69. UART 发送的停止程序



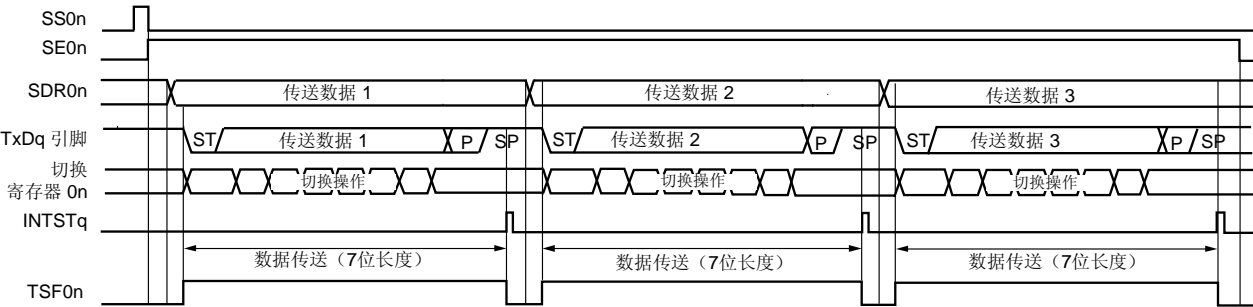
备注 即使通信停止后，引脚电平依然保持。要恢复操作，重新设置 SO0 寄存器（参见 图 12-70 恢复 UART 发送的程序）。

图 12-70. 恢复 UART 发送的程序



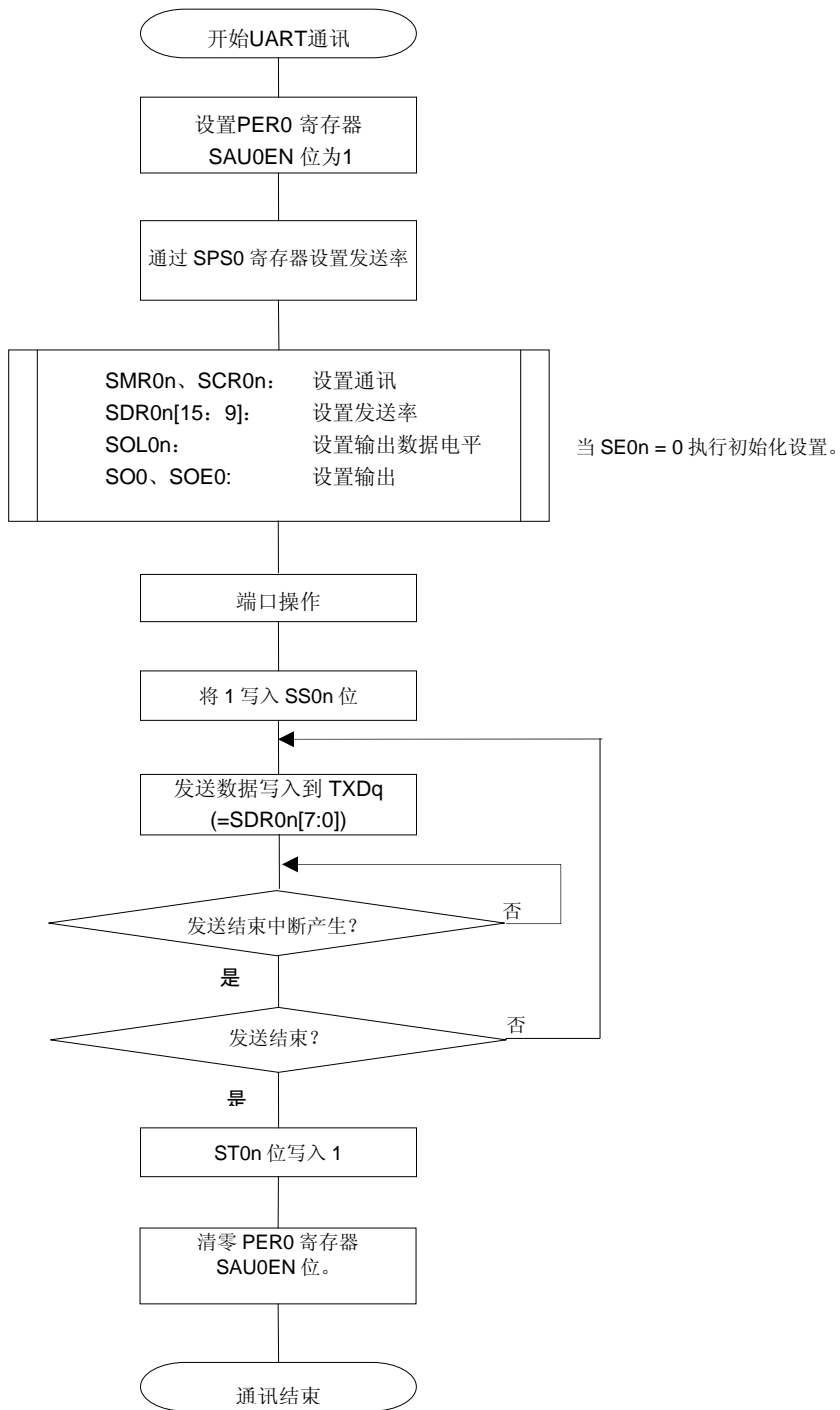
(3) 处理流程（在单发送模式下）

图 12-71. UART 发送的时序图（在单发送模式下）



备注 n: 通道数 (n = 0, 2), q: UART 数 (q = 0, 1)

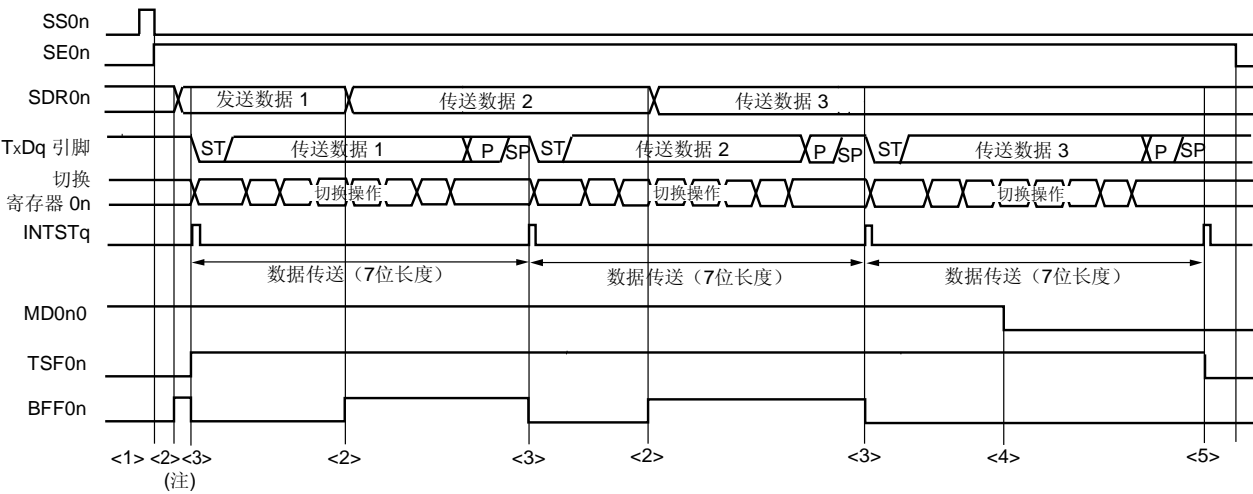
图 12-72. UART 发送的流程图（在单发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

(4) 处理流程（在连续发送模式下）

图 12-73. UART 发送的时序图（在连续发送模式下）

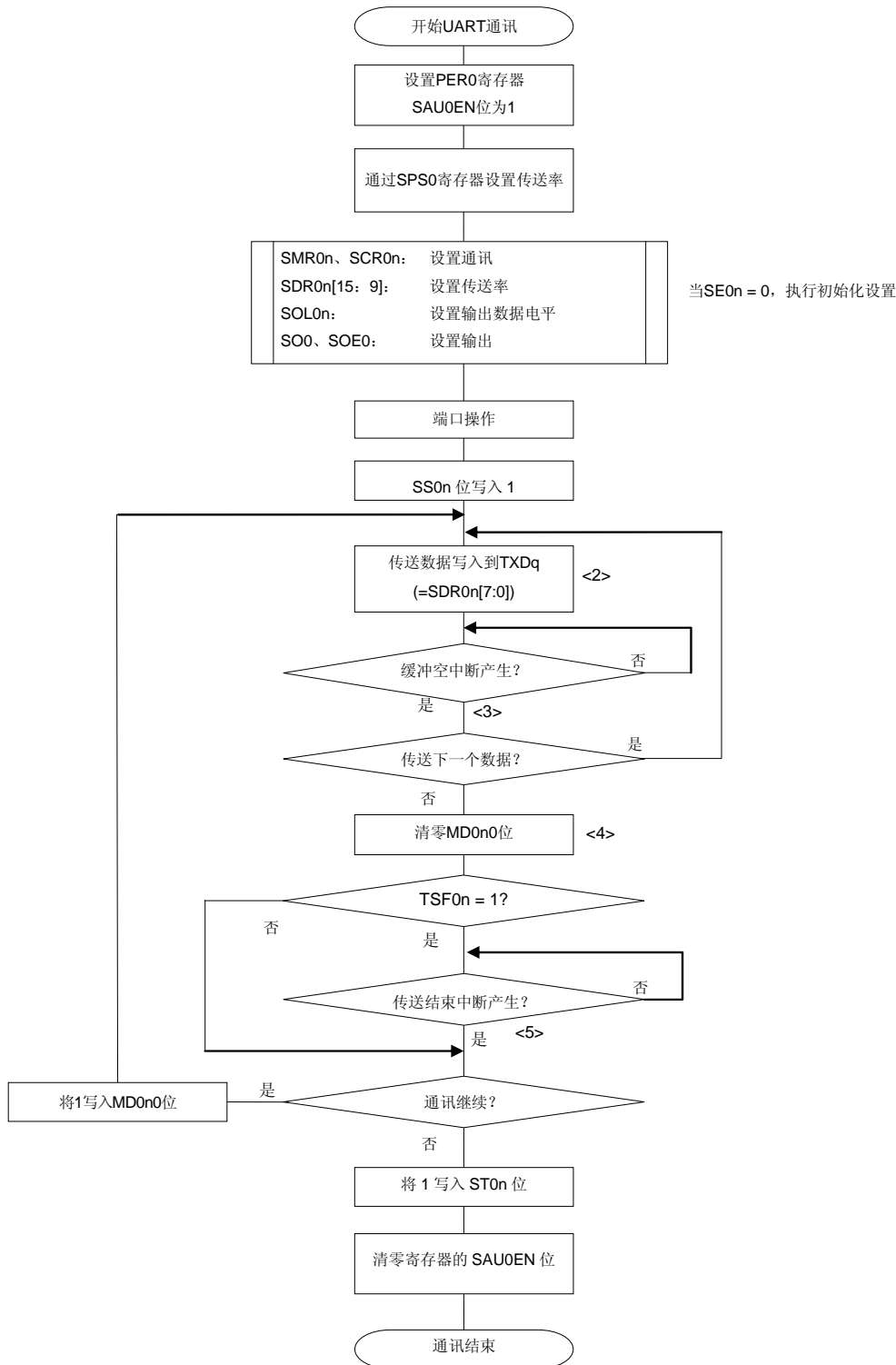


注 在 BFF0n = 1 的情况下，当发送数据写入 SDR0n 寄存器时，发送数据可以被重写。

注意事项 即使在操作过程中，MD0n0 位也可以被重写。
但是，要在最后一位开始发送之前重写它，这样它可以在最后一个发送数据的发送结束中断之前被重写。

备注 n: 通道数 (n = 0, 2), q: UART 数 (q = 0, 1)

图 12-74. UART 发送的流程图（在连续发送模式下）



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

备注 图中的<1>~<5> 对应图 12-73 UART 发送的时序图（在连续发送模式下）中的<1>~<5>。

12.6.2 UART接收

UART 接收是 μ PD79F9211 从另外一个设备同步接收数据的一个操作（同步起始停止）。

对于 UART 接收，使用用于 UART 两个通道中的奇通道。

UART	UART0	UART1
目标通道	SAU 的通道 1	SAU 的通道 3
使用引脚	RxD0	RxD1
中断	INTSR0	INTSR1
	仅发送结束中断（禁止设置缓冲空中断。）	
错误中断	INTSRE0	INTSRE1
错误检测标志	<ul style="list-style-type: none"> 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) 	
发送数据长度	5、7 或 8 位	
发送速率	$\text{Max } f_{\text{MCK}}/6 \text{ [bps]} \text{ (SDR0n [15:9] = 2 或更大), Min } f_{\text{CLK}}/(2 \times 2^{11} \times 128) \text{ [bps]}^{\#}$	
数据相位	正向输出（默认：高电平） 反向输出（默认：低电平）	
校验位	以下可选 <ul style="list-style-type: none"> 无校验位（无校验检查） 附加 0 校验（无校验检查） 附加偶校验 附加奇校验 	
停止位	附加 1 位	
数据方向	MSB 或者 LSB	

注 请在满足上面条件和电气特性中的 AC 特性的范围内，使用这个操作（参见 第 26 章 电气特性（目标））。

备注

1. f_{MCK} : 目标通道的工作时钟(MCK)频率
 f_{CLK} : 系统时钟频率
2. n : 通道编号 ($n = 1, 3$)

(1) 寄存器设置

图 12-75. 用于 UART(UART0, UART1)的 UART 接收寄存器的内容示例 (1/2)

(a) 串行输出寄存器 0 (SO0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 ×	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 ×	SO01 ×	SO00 ×

(b) 串行输出使能寄存器 0 (SOE0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 ×	SOE01 0/1	SOE00 ×

当用于 UART0 接收时，设置此位为 0。

(c) 串行通道启动寄存器 0 (SS0) ... 只设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 0/1	SS02 ×	SS01 0/1	SS00 ×

(d) 串行模式寄存器 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 0	0	0	0	0	0	STS0n 1	0	SIS0n0 0/1	1	0	0	MD0n2 0	MD0n1 1	MD0n0 0

0: 正向（正常）接收
1: 反向接收通道 n 的工作模式
0: 发送结束中断

(e) 串行模式寄存器 0r (SMR0r)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0r	CKS0r 0/1	CCS0r 0	0	0	0	0	0	STS0r 0	0	SIS0r0 0	1	0	0	MD0r2 0	MD0r1 1	MD0r0 0/1

与 CKS0n 的设置相同

通道 r 的工作模式:
0: 发送结束中断
1: 缓冲空中断**注意事项** 对于 UART 接收，确保设置通道 r 的 SMR0r，通道 r 是与通道 n 成对使用的偶数通道。**备注** n: 通道数 (n = 1, 3), r: 通道数 (r = n - 1)

□: 在 UART 接收模式下固定设置，■: 禁止设置（设为初始值）

x: 此模式下不能使用的位（不在任何模式下使用时，设为初始值）

0/1: 根据用户的使用设置为 0 或 1

图 12-75. 用于 UART(UART0, UART1)的 UART 接收寄存器的内容示例 (2/2)

(f) 串行通信操作设置寄存器 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0	DIR0n		SLC0n1	SLC0n0		DLS0n2	DLS0n1	DLS0n0
	0	1	0	0	0	1	0/1	0/1	0/1	0	0	1	0	1	0/1	0/1

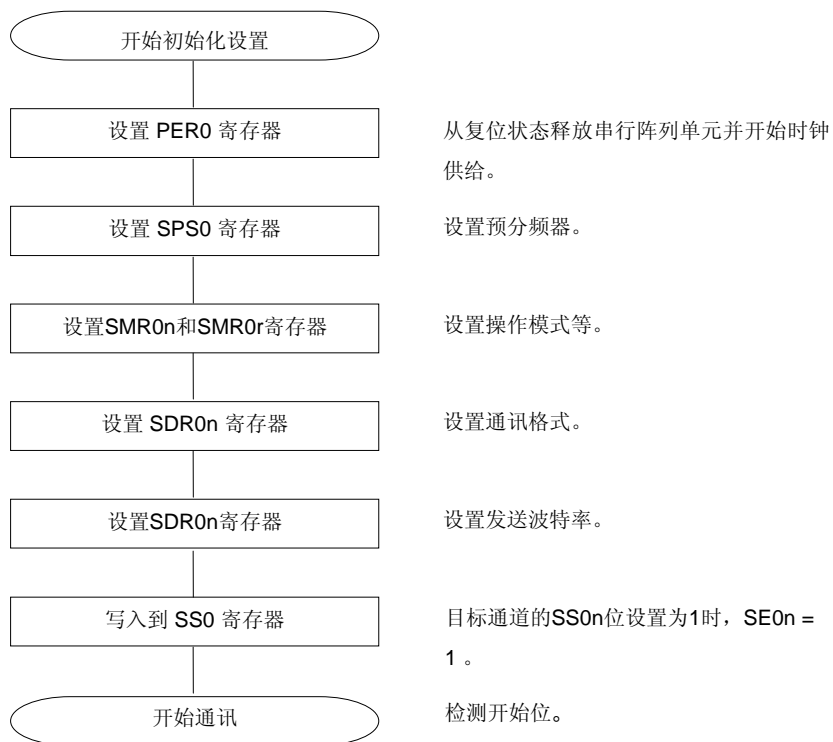
(g) 串行数据寄存器 0n (SDR0n) (低 8 位: RXDq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	设置波特率							0	接收数据寄存器							
								RXDq								

备注 n: 通道数 (n = 1, 3), q: UART 数 (q = 0, 1)
□: 在 UART 接收模式下固定设置, ■: 禁止设置 (设为初始值)
0/1: 根据用户的使用设置为 0 或 1

(2) 操作程序

图 12-76. UART 接收的初始化程序



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

图 12-77. UART 接收的停止程序

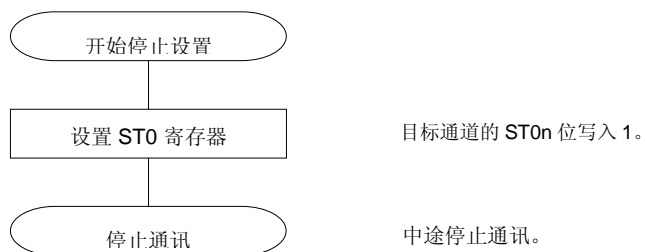
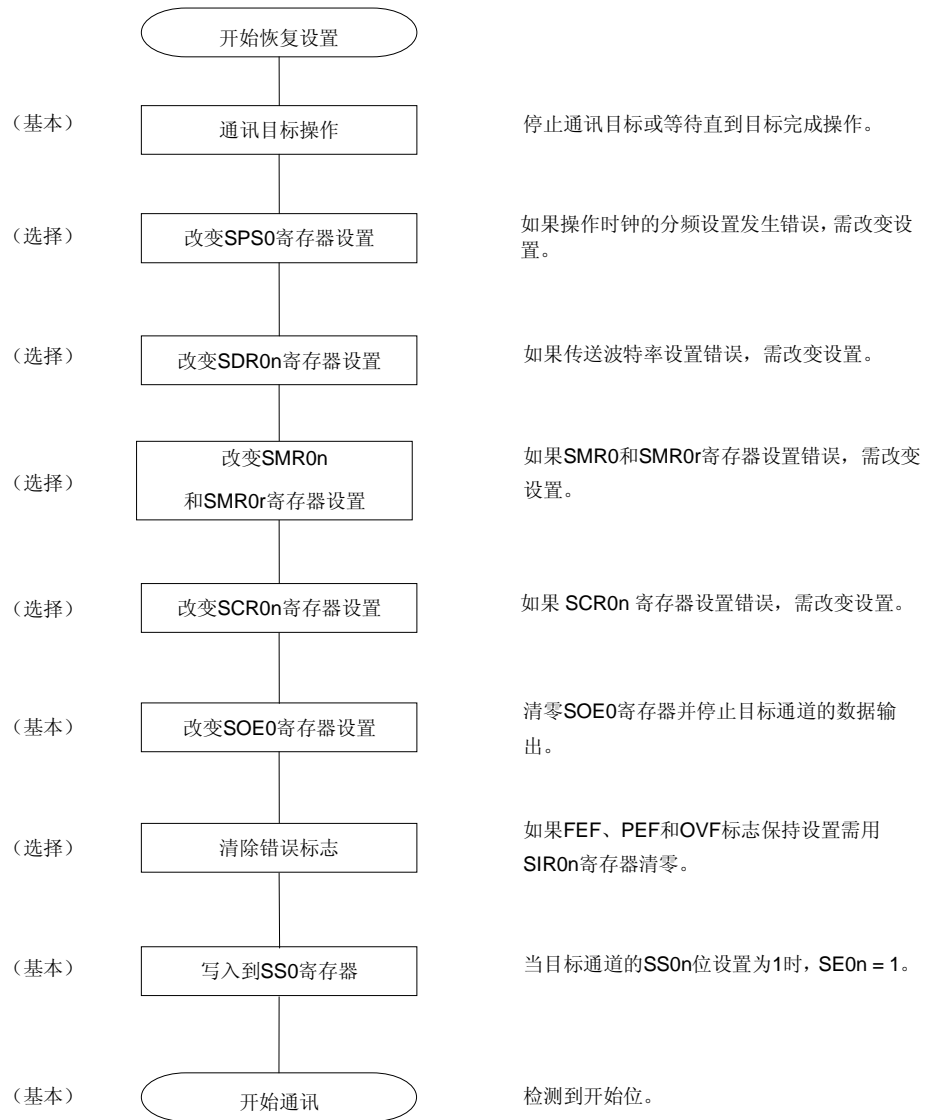
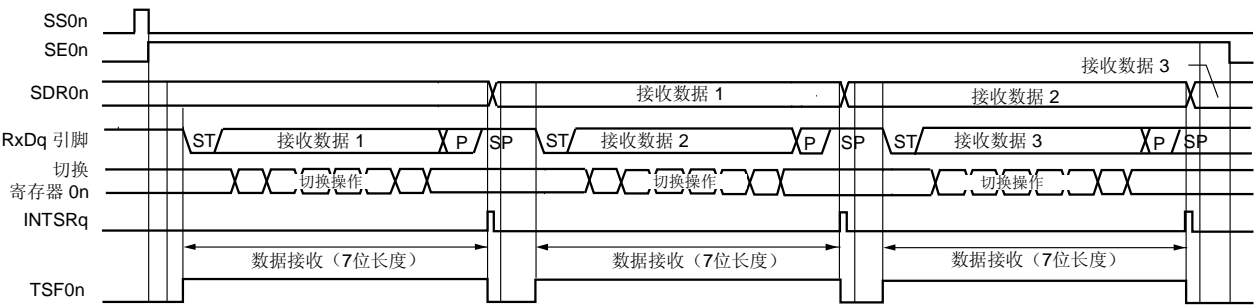


图 12-78. 恢复 UART 接收的程序



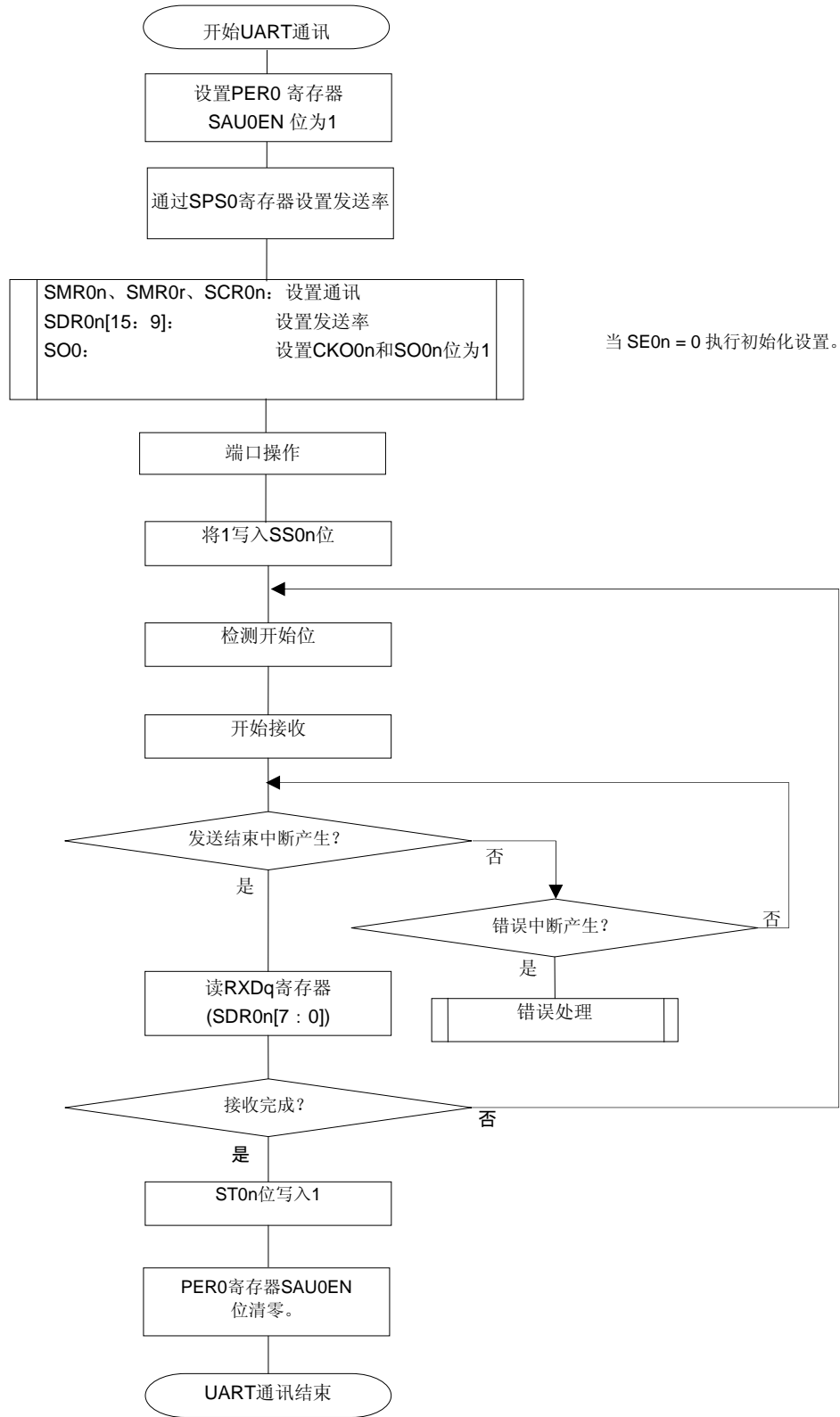
(3) 处理流程

图 12-79. UART 接收的时序图



备注 n: 通道数 (n = 1, 3), q: UART 数 (q = 0, 1)

图 12-80. UART 接收的流程图



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

12.6.3 LIN发送

在 UART 发送中，UART0 支持 LIN 通信。

关于 LIN 发送，使用单元(SAU)的通道 0。

UART	UART0	UART1
支持 LIN 的通信	支持	不支持
目标通道	SAU 的通道 0	—
使用引脚	TxD0	—
中断	INTST0	—
	可以选择发送结束中断（在单发送模式下）或者缓冲空中断（在连续发送模式下）	
错误检测标志	无	
发送数据长度	8 位	
发送速率	最大 $f_{MCK}/6$ [bps] (SDR00 [15:9] = 2 或更大, 最小 $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] [※]	
数据相位	正向输出（默认：高电平） 反向输出（默认：低电平）	
校验位	以下可选 <ul style="list-style-type: none"> • 无校验位 • 附加 0 校验 • 附加偶校验 • 附加奇校验 	
停止位	以下可选 <ul style="list-style-type: none"> • 附加 1 位 • 附加 2 位 	
数据方向	MSB 或者 LSB	

注 请在满足上面条件和电气特性中的 AC 特性的范围内，使用这个操作（参见 第 26 章 电气特性（目标））。

备注 f_{MCK} : 目标通道的工作时钟（MCK）频率

f_{CLK} : 系统时钟频率

LIN 代表本地互连网络并且是一个计划用于降低汽车网络费用的低速（1~20 kbps）串行通信协议。

LIN 的通信是单主通信，可以连接 15 个从设备到一个主设备。

这些从设备用于控制通过 LIN 连接到主设备的切换器，制动器和传感器。

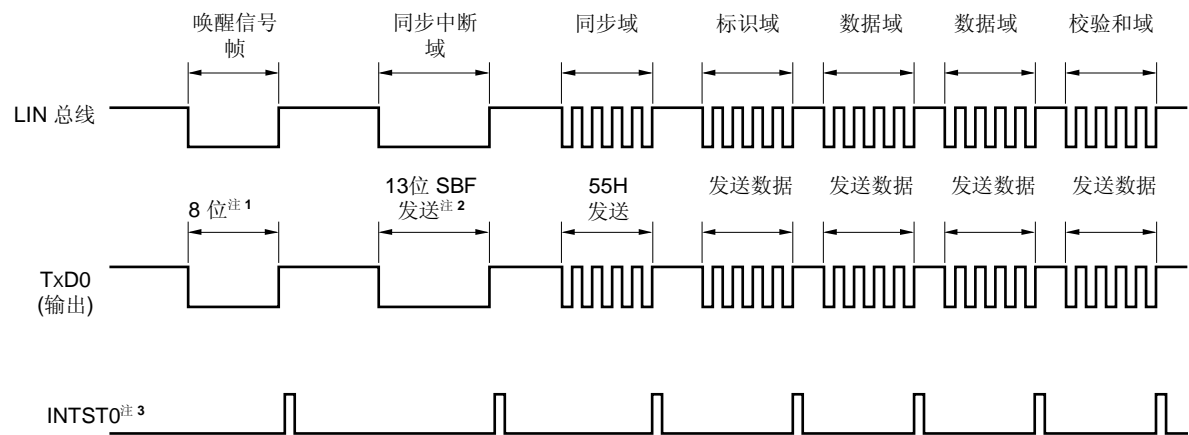
通常，主设备连接到例如 CAN(控制区域网络)的一个网络。

LIN 总线是一个单线总线，它的网络节点通过收发器连接到 ISO9141。

通过 LIN 协议，主设备发送一个含有波特率信息的帧。从设备从主设备接收这个帧，并纠正波特率误差。如果从设备的波特率误差在±15%，则可以建立这个通信。

图 12-81 描述了发送 LIN 的操作。

图 12-81. LIN 的操作的发送

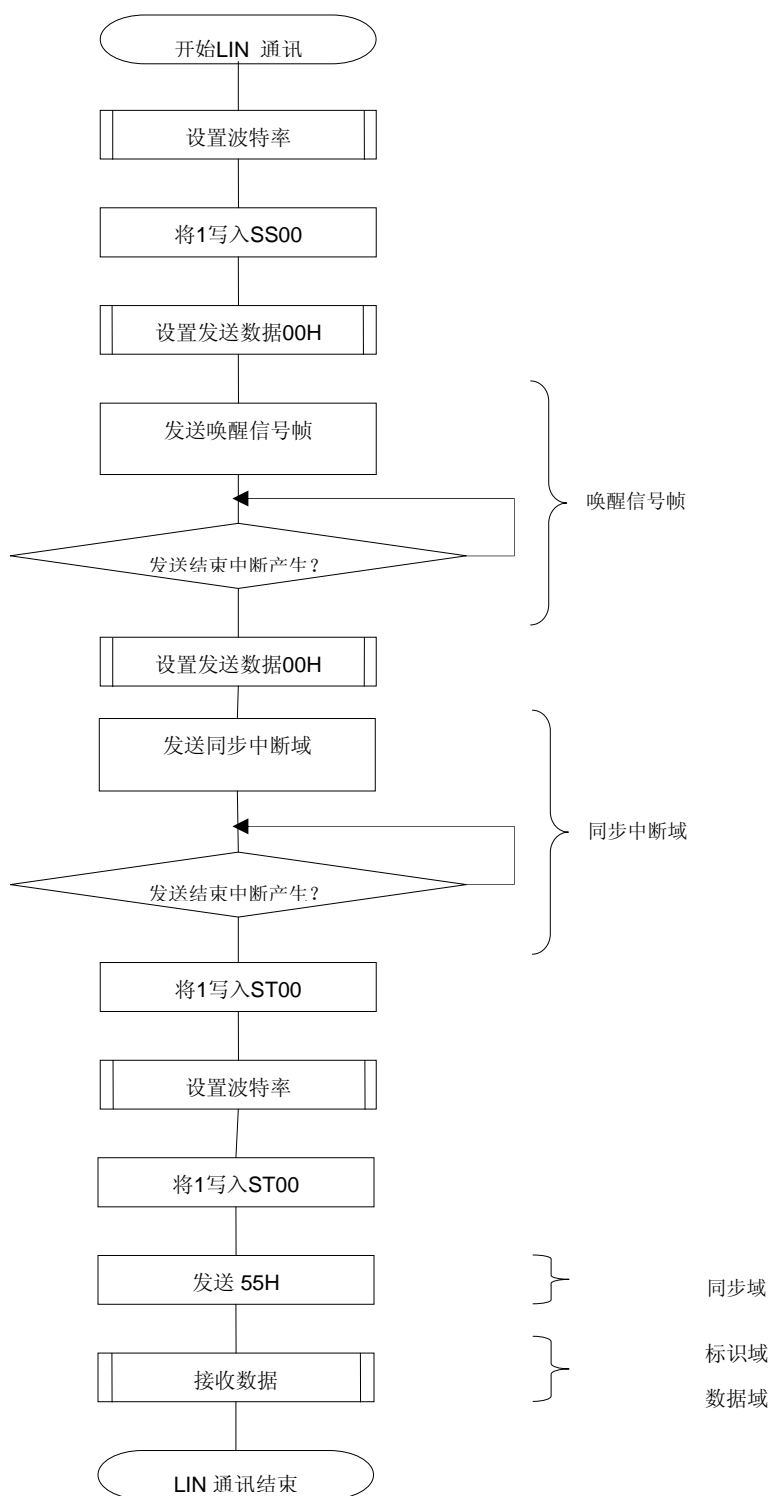


- 注
1. 设置波特率满足唤醒信号的标准并且发送数据 00H。
 2. 同步中断域定义为长度为 13 位并且输出低电平。因此，当主发送的波特率为 N [bps]时，同步中断区域的波特率计算如下。
$$(\text{同步中断区域的波特率}) = 9/13 \times N$$

通过在这个波特率下发送数据 00H，产生一个同步中断域
 3. INTST0 是发送完成后的输出。当执行 SBF 发送时，同样也输出 INTST0。

备注 域的间隔是通过软件控制的。

图 12-82. LIN 发送的流程图



12.6.4 LIN 接收

在 UART 接收中，UART0 支持 LIN 通信。

对于 LIN 接收，使用单元(SAU)的通道 1。

UART	UART0	UART1
支持 LIN 的通信	支持	不支持
目标通道	SAU 的通道 1	—
使用引脚	RxD0	—
中断	INTSR0	—
	仅发送结束中断（禁止设置缓冲空中断。）	
错误中断	INTSRE0	—
错误检测标志	<ul style="list-style-type: none"> 帧错误检测标志 (FEF01) 奇偶校验错误检测标志 (PEF01) 溢出错误检测标志 (OVF01) 	
发送数据长度	8 位	
发送速率	最大 $f_{MCK}/6$ [bps] (SDR01 [15:9] = 2 或更大), 最小 $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] [※]	
数据相位	正向输出（默认：高电平） 反向输出（默认：低电平）	
校验位	以下可选 <ul style="list-style-type: none"> 无校验位 附加 0 校验 附加偶校验 附加奇校验 	
停止位	以下可选 <ul style="list-style-type: none"> 附加 1 位 附加 2 位 	
数据方向	MSB 或者 LSB	

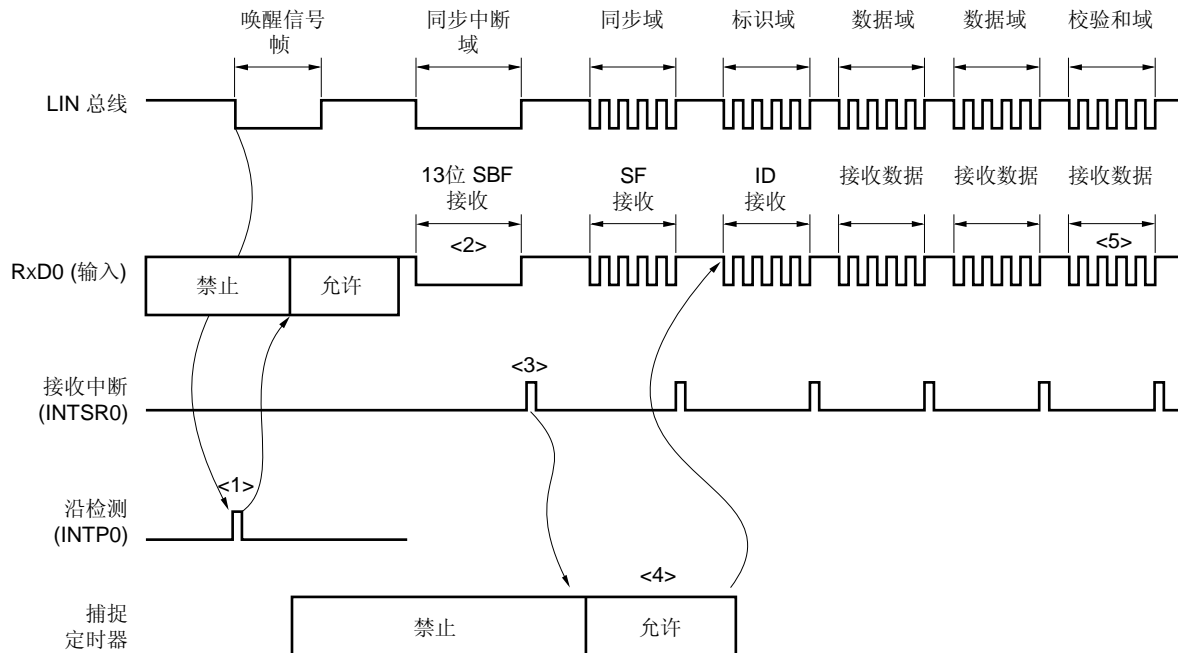
注 请在满足上面条件和电气特性中的 AC 特性的范围内，使用这个操作（参见第 26 章 电气特性（目标））。

备注 f_{MCK} : 目标通道的工作时钟(MCK)频率

f_{CLK} : 系统时钟频率

图 12-83 描述了 LIN 的接收操作。

图 12-83. LIN 的接收操作



这是下面信号处理。

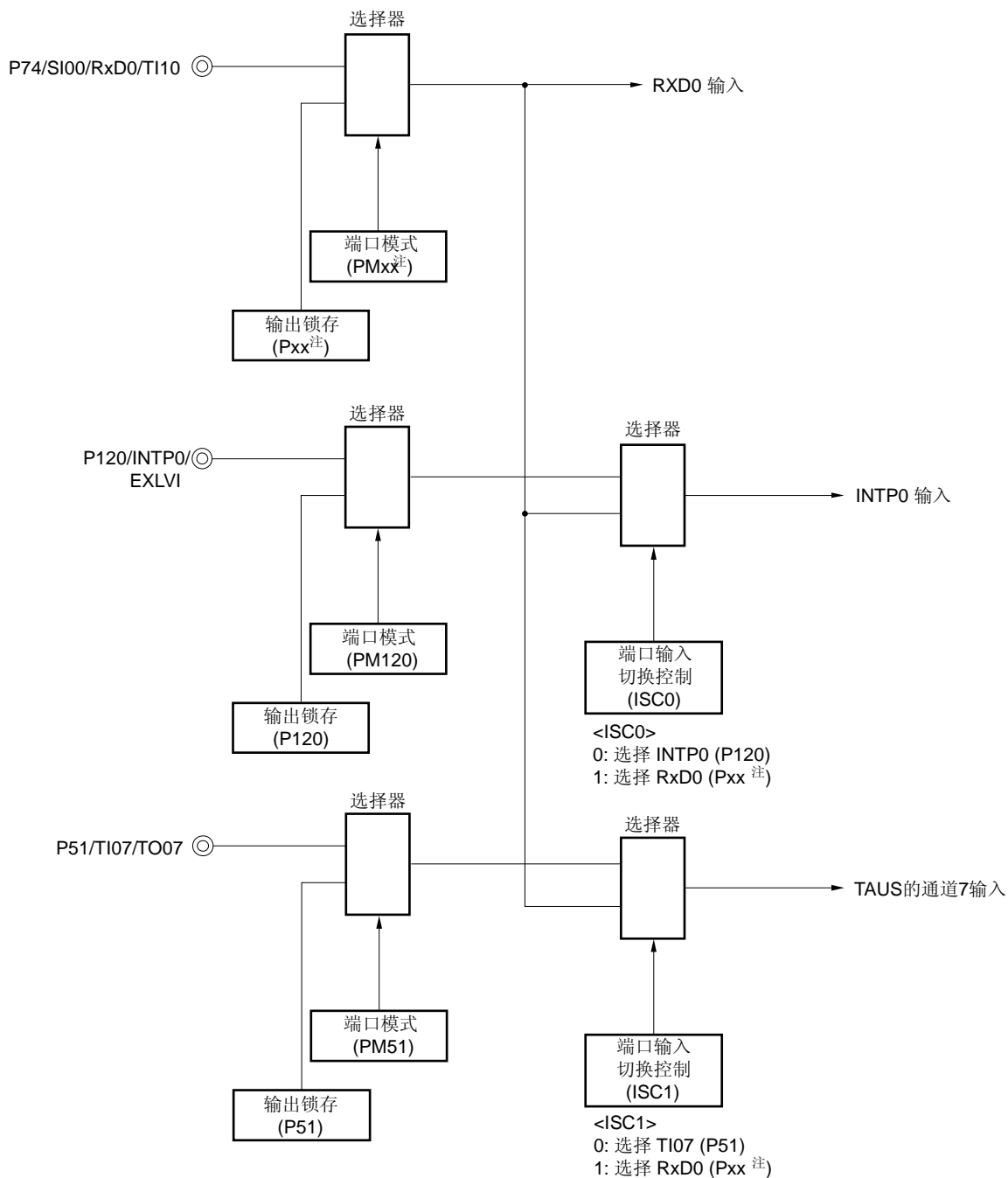
- <1> 通过检测一个引脚的中断沿(INTPO)来检测唤醒信号。 当检测到唤醒信号，允许 UART0 的接收(RXE01 = 1)并等待 SBF 接收。
- <2> 当检测到 SBF 的起始位时，接收以设置的波特率开始且产生的串行数据保存在 RXD0 寄存器中 (= 串行数据寄存器 01 (SDR01)的第 7 位~第 0 位)。当检测到停止位，产生接收结束中断请求 (INTSR0)。当 11 位或者更多的低电平的数据被检测为 SBF，则判断 SBF 接收正确结束。如果少于 11 位的低电平的数据被检测为 SBF，则产生接收错误，并且系统返回到 SBF 接收等待状态。
- <3> 当 SBF 正确接收完成时，开始定时器阵列单元 TUAS 的通道 7 并测量同步中断区域的位间隔（脉冲宽度）（见 6.7.5 用作输入信号高/低电平宽度测量的操作）。
- <4> 从同步域(SF)的位间隔计算波特率误差。停止 UART0 一次并且调节（重新设置）波特率。
- <5> 校验和区域可以通过软件来辨别。另外，在接收到校验和区域之后，执行初始化 UART0，并且等待 SBF 的接收可以通过软件来执行。

图 12-84 展示了执行 LIN 接收的端口的配置。

从 LIN 的主设备发送的唤醒信号通过检测外部中断（INTP0）的有效沿来接收。从主通道发送的同步域的长度，可以利用定时器阵列单元 TAUS 的外部事件捕捉操作来测量，来计算波特率误差。

通过控制端口输入（ISC0/ISC1）的切换，用于接收的端口输入（RxD0）的输入源可以输入到外部中断引脚（INTP0）和定时器阵列单元 TAUS。

图 12-84. LIN 操作接收的端口配置



注 xx = 74

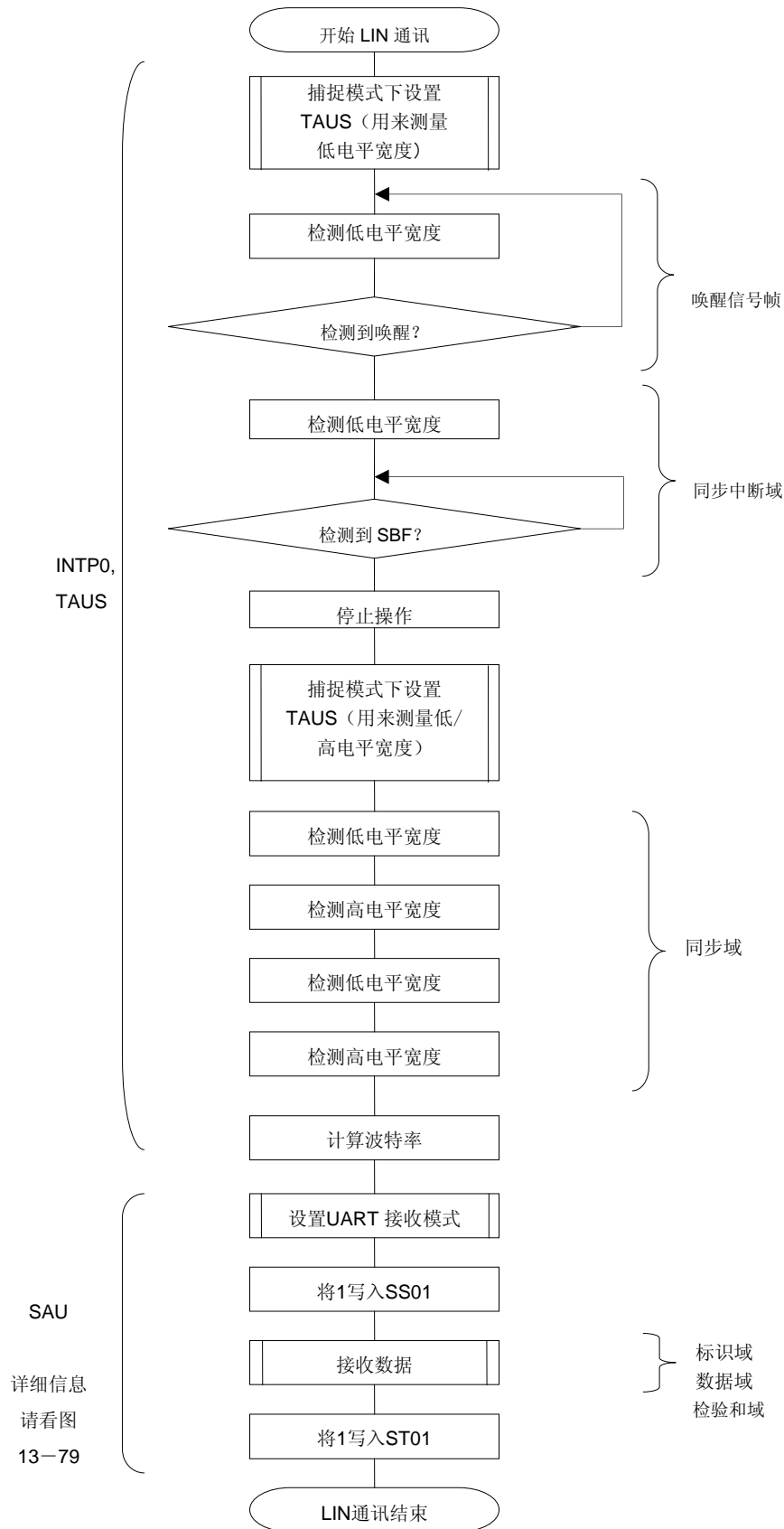
备注 ISC0、ISC1：输入切换控制寄存器（ISC）的第 0 位和第 1 位（参阅 图 12-16。）

用于 LIN 通信的外围功能如下。

<使用的外围功能>

- 外部中断（INTP0）；唤醒信号检测
用途： 检测唤醒信号边沿和通信开始
- 定时器阵列单元 TAUS 的通道 7；波特率错误检测。
用途： 检测同步区域（SF）的长度并除以位数从而检测错误（输入到 RxD0 的边沿间隔在捕捉模式中测量。）
- 串行阵列单元（SAU）的通道 0 和 1（UART0）

图 12-85. LIN 接收的流程图



12.6.5 计算波特率

(1) 波特率计算公式

UART (UART0, UART1) 通信的波特率可以按照下列表达式来计算。

$$(\text{波特率}) = \{\text{目标通道的工作时钟(MCK)频率}\} \div (\text{SDR0n}[15:9] + 1) \div 2 [\text{bps}]$$

注意事项 禁止设置 $\text{SDR0n}[15:9] = (0000000\text{B}, 0000001\text{B})$ 。

- 备注**
1. 当使用 UART 时, $\text{SDR0n}[15:9]$ 的值是 SDR0n 寄存器的第 15 位~第 9 位的值 (0000010B ~1111111B), 因此它的取值为 2~127。
 2. n: 单元编号 (n = 0~3)

工作时钟 (MCK) 由串行时钟选择寄存器 0 (SPS0) 和串行模式寄存器 0n (SMR0n) 的第 15 位 (CKS0n) 来确定。

表 12-3. 工作时钟的选择

SMR0n 寄存器		SPS0 寄存器							工作时钟(MCK) ^{※1}	
CKS0n	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000		f _{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f _{CLK}	20 MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	10 MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	5 MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	2.5 MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	1.25 MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	625 kHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	313 kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	156 kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	78.1 kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	39.1 kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	19.5 kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 ^{※2}	
1	0	0	0	0	X	X	X	X	f _{CLK}	20 MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	10 MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	5 MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	2.5 MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	1.25 MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	625 kHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	313 kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	156 kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	78.1 kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	39.1 kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	19.5 kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 ^{※2}	
其它值									设置禁止	

- 注 1. 当改变用于 f_{CLK} 的时钟 (通过改变系统时钟控制寄存器 (CKC) 的值) 时, 在停止串行阵列单元 (SAU) 的操作 (ST0 = 000FH) 之后, 才这样做。当选择 INTTM02 为工作时钟时, 同样停止定时器阵列单元 TAUS (TT0 = 00FFH)。
2. 利用 SPS0 寄存器, 通过设置 TAUS 的 TIS0 寄存器的 TIS02 位为 1, 为输入时钟选择 f_{SUB}/4, 并且选择 INTTM02, 可以在副系统时钟的一个固定分频比时操作 SAU, 无论 f_{CLK} 频率 (主系统时钟、副系统时钟) 如何。但是, 当改变 f_{CLK}, 必须按照上面注 1 描述的停止 SAU 和 TAUS。

- 备注 1. X: 不必考虑
2. n: 单元编号 (n = 0~3)

(2) 发送期间的波特率误差

发送期间的 UART (UART0, UART1)通信的波特率误差可以通过下面公式计算。 确保发送方的波特率在接收方的允许波特率范围之内。

$$(\text{波特率出错}) = (\text{计算的波特率数值}) \div (\text{目标波特率}) \times 100 - 100 [\%]$$

这在 $f_{\text{CLK}} = 20 \text{ MHz}$ 下设置 UART 波特率的例子。

UART 波特率 (目标波特率)	$f_{\text{CLK}} = 20 \text{ MHz}$			
	工作时钟(MCK)	SDR0n[15:9]	计算的波特率	目标波特率的误差
300 bps	$f_{\text{CLK}}/2^9$	64	300.48 bps	+0.16 %
600 bps	$f_{\text{CLK}}/2^8$	64	600.96 bps	+0.16 %
1200 bps	$f_{\text{CLK}}/2^7$	64	1201.92 bps	+0.16 %
2400 bps	$f_{\text{CLK}}/2^6$	64	2403.85 bps	+0.16 %
4800 bps	$f_{\text{CLK}}/2^5$	64	4807.69 bps	+0.16 %
9600 bps	$f_{\text{CLK}}/2^4$	64	9615.38 bps	+0.16 %
19200 bps	$f_{\text{CLK}}/2^3$	64	19230.8 bps	+0.16 %
31250 bps	$f_{\text{CLK}}/2^3$	39	31250.0 bps	$\pm 0.0 \%$
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	$\pm 0.0 \%$

备注 n: 通道 (n = 0, 2)

(3) 接收允许的波特率范围

在 UART (UART0, UART1) 通信期间，用于接收允许的波特率范围可以通过下面公式计算。确保发送方的波特率在接收方的允许波特率范围之内。

(最大接收波特率) = $\frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$

(最小接收波特率) = $\frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$

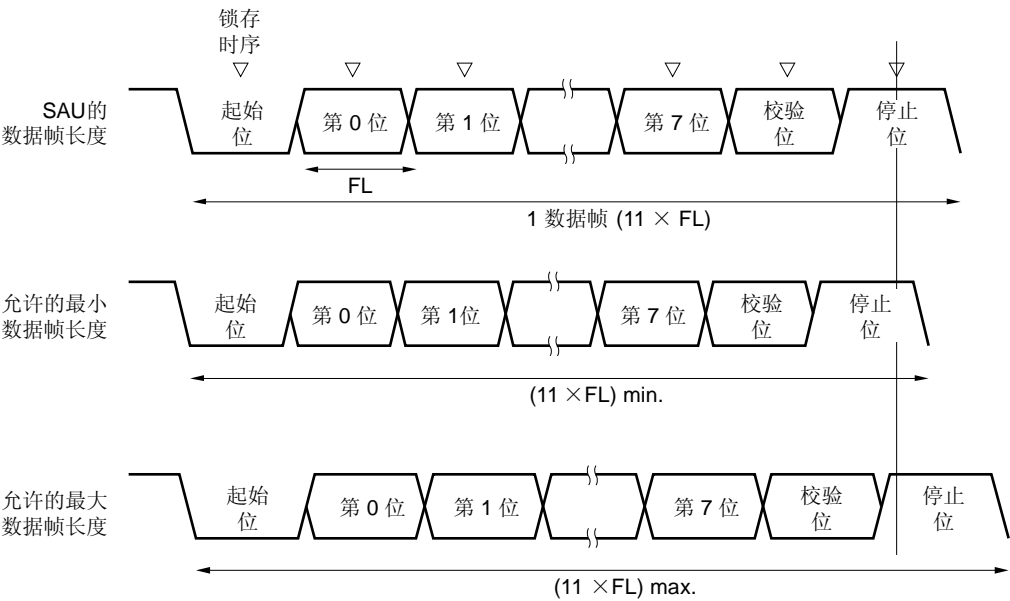
Brate: 在接收方计算的波特率（参见 12.6.5 (1) 波特率计算公式。）

k: SDR0n[15:9] + 1

Nfr: 1 数据帧长度[位]
= (起始位) + (数据长度) + (校验位) + (停止位)

备注 n: 通道 (n = 1, 3)

图 12-86. 接收允许的波特率范围 (1 数据帧长度 = 11 位)



如图 12-86 所示，在检测到起始位之后，锁存接收数据的时序由分频比决定，而分频比是通过设置串行数据寄存器 0n (SDR0n) 的第 15 位~第 9 位获得。如果在获得这个时序之前接收到最后一个数据（停止位），可以正确接收这个数据。

12.7 简易I²C (IIC10)通信操作

通过使用 2 线用于和两个或者更多设备通信的时钟通信功能：串行时钟(SCL)和串行数据(SDA)。该通信功能计算执行具有例如 EEPROM, flash 存储器, 和 A/D 转换器设备的单通信, 因此, 仅可以通过主使用并且不具有等待检测功能。确保通过使用软件, 和操作控制寄存器, 可以读取起始停止条件的 AC 特性。

[数据发送/接收]

- <R>
- 主发送、主接收（仅具有主功能的主设备）
 - ACK 输出功能*和 ACK 检测功能
 - 8 位数据长度
（当发送地址, 高 7 位描述地址, 最后一位用于 R/W 控制。）
 - 手动产生启动条件和停止条件

[中断功能]

- 发送结束中断

[错误检测标志]

- 校验错误（ACK 错误）

* [简易 I²C 不支持的功能]

- 从发送、从接收
- 仲裁丢失检测功能
- 等待检测功能

<R>

注

当接收最后一个数据时, 如果 0 写入 SOE02 （SOE0 寄存器）位且串行通信数据输出被停止, 则 ACK 不会输出。详细信息请参阅 12.7.3 (2)中的处理流程。

支持简易 I²C(IIC10)的通道是 SAU 的通道 2。

通道	用作 CSI	用作 UART	用作简易 I ² C
0	CSI00	UART0（支持 LIN 总线）	—
1	CSI01		—
2	CSI10	UART1	IIC10
3	—		—

简易 I²C（IIC10）执行以下 4 种类型的通信操作。

- 发送地址区域（见 12.7.1.）
- 发送数据（见 12.7.2.）
- 接收数据（见 12.7.3.）
- 产生停止条件（见 12.7.4.）

12.7.1 地址域发送

发送地址区域是一个第一次执行 I²C 通信来定义主（从）目标的发送操作。产生启动条件后，在一帧里发送一个地址（7 位）和发送方式（1 位）。

简易 I ² C	IIC10
目标通道	SAU 的通道 2
使用引脚	SCL10, SDA10
中断	INTIIC10
	仅发送结束中断（禁止设置缓冲空中断。）
错误检测标志	奇偶校验错误检测标志（PEF02）
发送数据长度	8 位（高 7 位作为地址，最后一位用于 R/W 控制）
发送速率	最大 f _{CLK} /4 MHz f _{CLK} : 系统时钟频率 然而，I ² C 的每个模式下必须满足下列条件。 <ul style="list-style-type: none"> • 最大 400 kHz（第一模式） • 最大 100 kHz（标准模式）
数据电平	正向输出（默认：高电平）
校验位	无检验位
停止位	附加 1 位（用于 ACK 接收时序）
数据方向	MSB

(1) 寄存器设置

图 12-87. 简易 I²C (IIC10) 地址域发送寄存器内容的举例

(a) 串行输出寄存器 0 (SO0)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 0/1	SO01 ×	SO00 ×

通过操作 SO02 位，产生启动条件。

(b) 串行输出允许寄存器 0 (SOE0)...仅设置目标通道的位。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	SOE01 ×	SOE00 ×

直到产生启动条件，SOE02 = 0，并且在产生后 SOE02 = 1。

(c) 串行通道启动寄存器 0 (SS0) ...仅设置目标通道的位为 1。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 ×

(d) 串行模式寄存器 02 (SMR02)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR02	CKS02 0/1	CCS02 0	0	0	0	0	0	STS02 0	0	SIS020 0	1	0	0	MD022 1	MD021 0	MD020 0

通道 2 的工作模式
0: 发送结束中断

(e) 串行通信操作设置寄存器 02 (SCR02)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR02	TXE02 1	RXE02 0	DAP02 0	CKP02 0	0	EOC02 0	PTC02 1	PTC02 0	DIR02 0	0	SLC02 1	SLC020 1	0	DLS02 2	DLS02 1	DLS020 1

设置校验位
00B: 无校验设置停止位
01B: 附加 1 位(ACK)

(f) 串行数据寄存器 02(SDR02) (低 8 位: SIO10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR02	设置波特率							0	设置发送数据 (地址 + R/W)							

SIO10

备注 □: 在 IIC 模式下固定设置, ■.: 禁止设置 (设为初始值)
 ×: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)
 0/1: 根据用户的使用设置为 0 或 1

(2) 操作程序

图 12-88. 地址域发送的初始化程序



注意事项 在设置 PER0 寄存器为 1 后，确保在 4 个或更多时钟周期之后对 SPS0 寄存器进行设置。

(3) 处理流程

图 12-89. 地址域发送的时序图

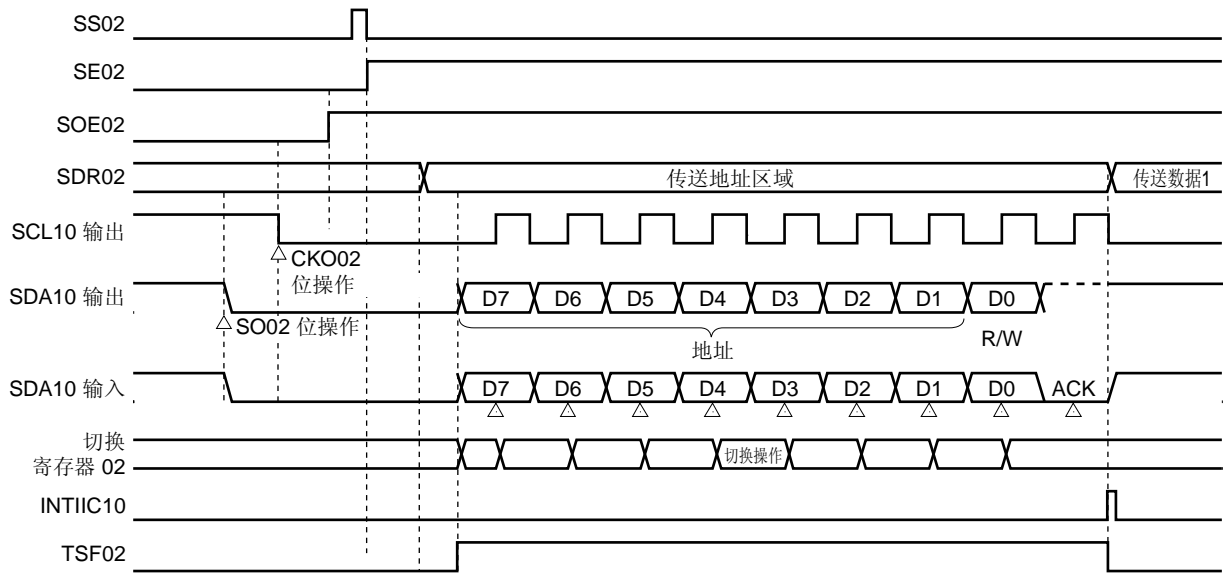
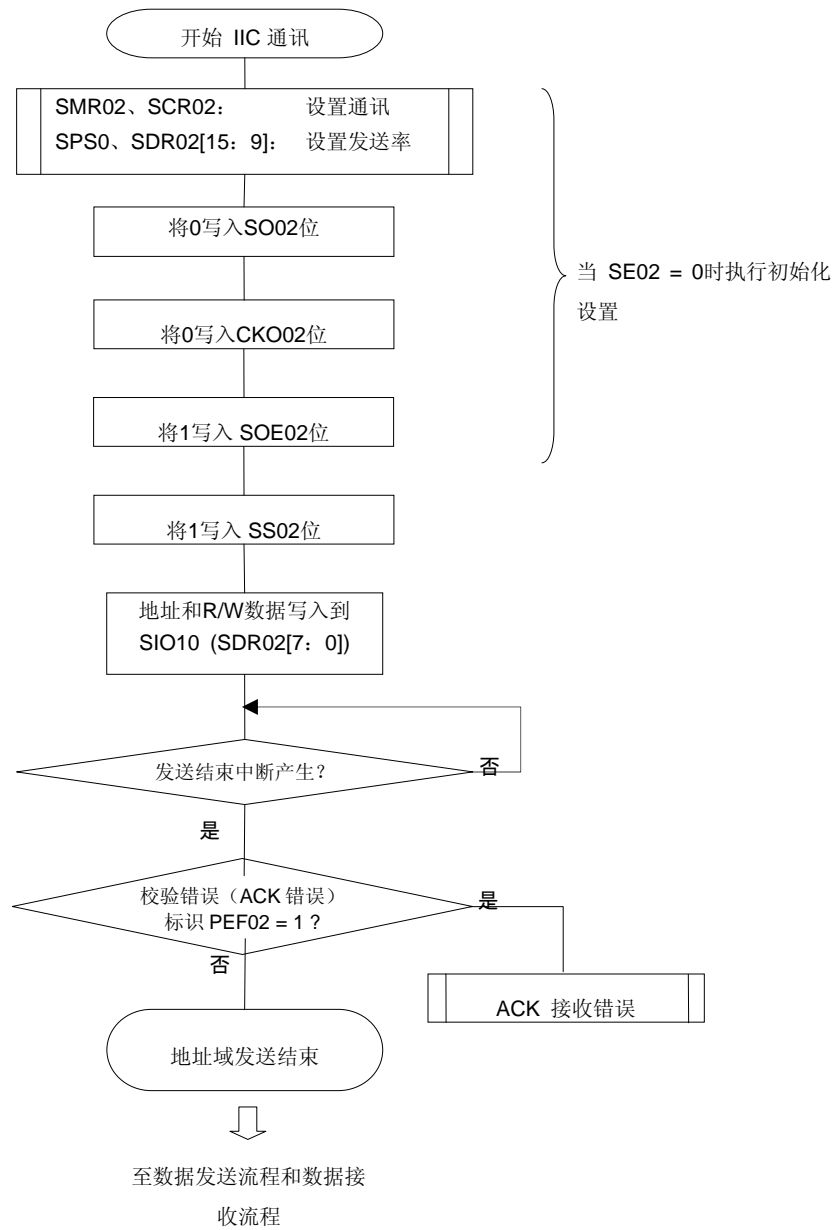


图 12-90. 地址域发送的流程图



12.7.2 数据发送

数据发送是一个在接收到地址域后发送（从）目标数据的操作。 全部数据发送到从设备后，产生一个停止条件并且总线释放。

简易 I ² C	IIC10
目标通道	SAU 的通道 2
使用引脚	SCL10, SDA10
中断	INTIIC10
	仅发送结束中断（禁止设置缓冲空中断。）
错误检测标志	奇偶校验错误检测标志（PEF02）
发送数据长度	8 位
发送速率	最大 $f_{CLK}/4$ MHz f_{CLK} : 系统时钟频率 然而，I ² C 的每个模式下必须满足下列条件。 <ul style="list-style-type: none"> • 最大 400 kHz（第一模式） • 最大 100 kHz（标准模式）
数据电平	正向输出（默认：高电平）
校验位	无检验位
停止位	附加 1 位（用于 ACK 接收时序）
数据方向	MSB

(1) 寄存器设置

图 12-91. 简易 I²C (IIC10) 的数据发送寄存器内容示例

(a) 串行输出寄存器 0 (SO0)...

不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0						CKO02 0/1 ^注	CKO01 ×	CKO00 ×						SO02 0/1 ^注	SO01 ×	SO00 ×
	0	0	0	0	1		×	×	0	0	0	0	1		×	×

(b) 串行输出使能寄存器 0 (SOE0)...

不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0														SOE02	SOE01	SOE00
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	×	×

(c) 串行通道启动寄存器 0 (SS0)...

不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0														SS03	SS02	SS01
	0	0	0	0	0	0	0	0	0	0	0	0	0	×	0/1	×
															×	×

(d) 串行模式寄存器 02 (SMR02)...

不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR02	CKS02	CCS02						STS02		SIS020				MD022	MD021	MD020
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(e) 串行通信操作设置寄存器 02 (SCR02)...

不要操作该寄存器的位，

除了位 TXE02 和

RXE02，在数据

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR02	TXE02	RXE02	DAP02	CKP02		EOC02	PTC02	PTC02	DIR02		SLC02	SLC020		DLS02	DLS02	DLS020
	1	0	0	0	0	0	1	0	0	0	1	1	0	2	1	1
							0	0			0			1	1	1

(f) 串行数据寄存器 02 (SDR02) (低 8 位: SIO10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR02	设置波特率							0	设置发送数据							
									SIO10							

注 该值由通信操作过程中的通信数据决定。

备注 ☐: 在 IIC 模式下固定设置, ☐: 禁止设置 (设为初始值)

×: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)

0/1: 根据用户的使用设置为 0 或 1

(2) 处理流程

图 12-92. 数据发送的时序图

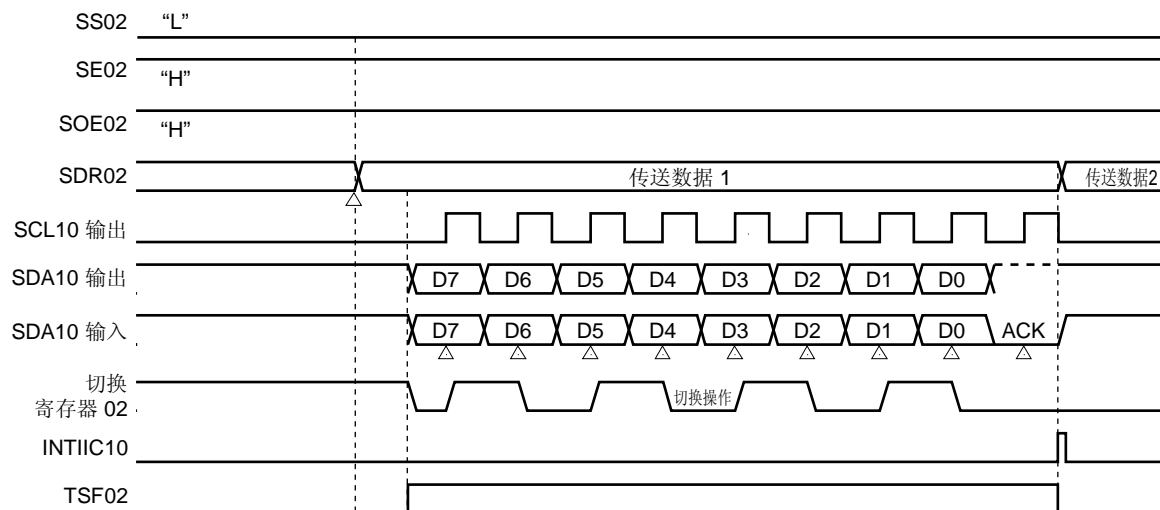
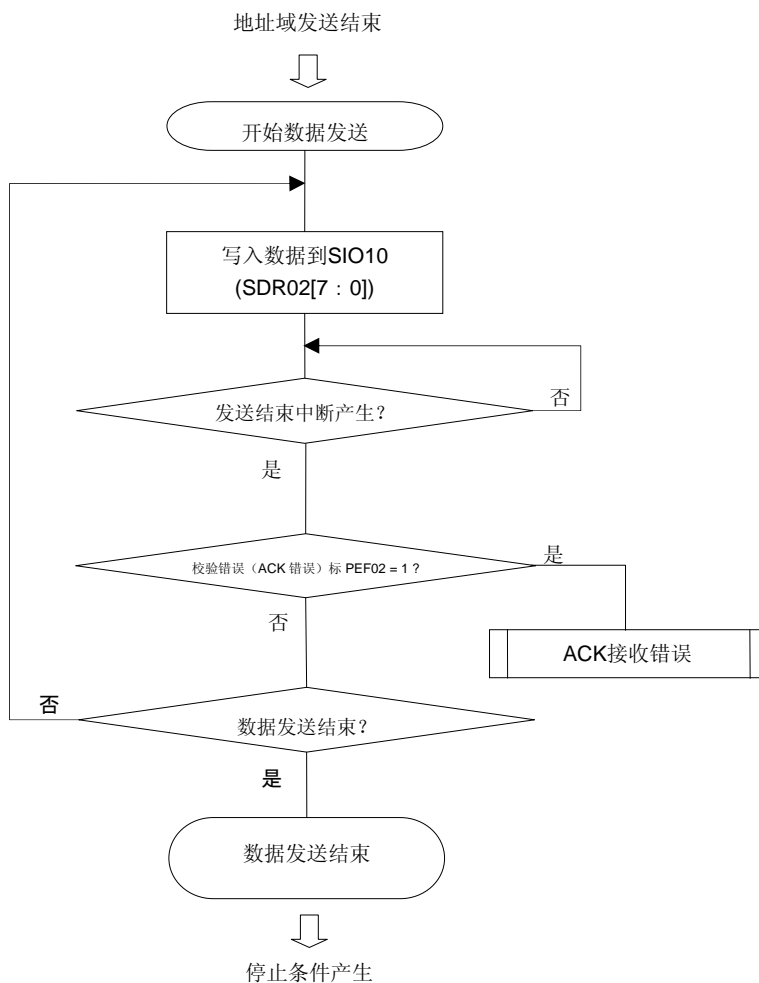


图 12-93. 数据发送的流程图



12.7.3 数据接收

数据接收是在接收到地址域后接收（从）目标数据的操作。从设备的数据全部接收后，产生一个停止条件并且总线释放。

简易 I ² C	IIC10
目标通道	SAU 的通道 2
使用引脚	SCL10, SDA10
中断	INTIIC10
	仅发送结束中断（禁止设置缓冲空中断。）
错误检测标志	无
发送数据长度	8 位
发送速率	最大 f _{CLK} /4 MHz f _{CLK} : 系统时钟频率 然而，I ² C 的每个模式下必须满足下列条件。 <ul style="list-style-type: none"> • 最大 400 kHz（第一模式） • 最大 100 kHz（标准模式）
数据电平	正向输出（默认：高电平）
校验位	无检验位
停止位	附加 1 位（用于 ACK 接收时序）
数据方向	MSB

(1) 寄存器设置

图 12-94. 简易 I²C (IIC10) 的数据接收寄存器内容示例

(a) 串行输出寄存器 0 (SO0)...

不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1 ^注	CKO01 ×	CKO00 ×	0	0	0	0	1	SO02 0/1 ^注	SO01 ×	SO00 ×

(b) 串行输出使能寄存器 0 (SOE0)...

不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 1	SOE01 ×	SOE00 ×

(c) 串行通道启动寄存器 0(SS0)...

不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 ×

(d) 串行模式寄存器 02 (SMR02)...

不要在数据发送/接收过程中修改该寄存器。

发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR02	CKS02 0/1	CCS02 0	0	0	0	0	0	STS02 0	0	SIS020 0	1	0	0	MD022 1	MD021 0	MD020 0

(e) 串行通信操作设置寄存器 02 (SCR02)...

不要操作该寄存器的位，
除了位 TXE02 和
RXE02，在数据
发送/接收。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR02	TXE02 0	RXE02 1	DAP02 0	CKP02 0	0	EOC02 0	PTC02 1 0	PTC02 0 0	DIR02 0	0	SLC02 1 0	SLC020 1	0	DLS02 2 1	DLS02 1 1	DLS020 1

(f) 串行数据寄存器 02(SDR02) (低 8 位: SIO10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR02	设置波特率							0	设置伪发送数据 (FFH)							
SIO10																

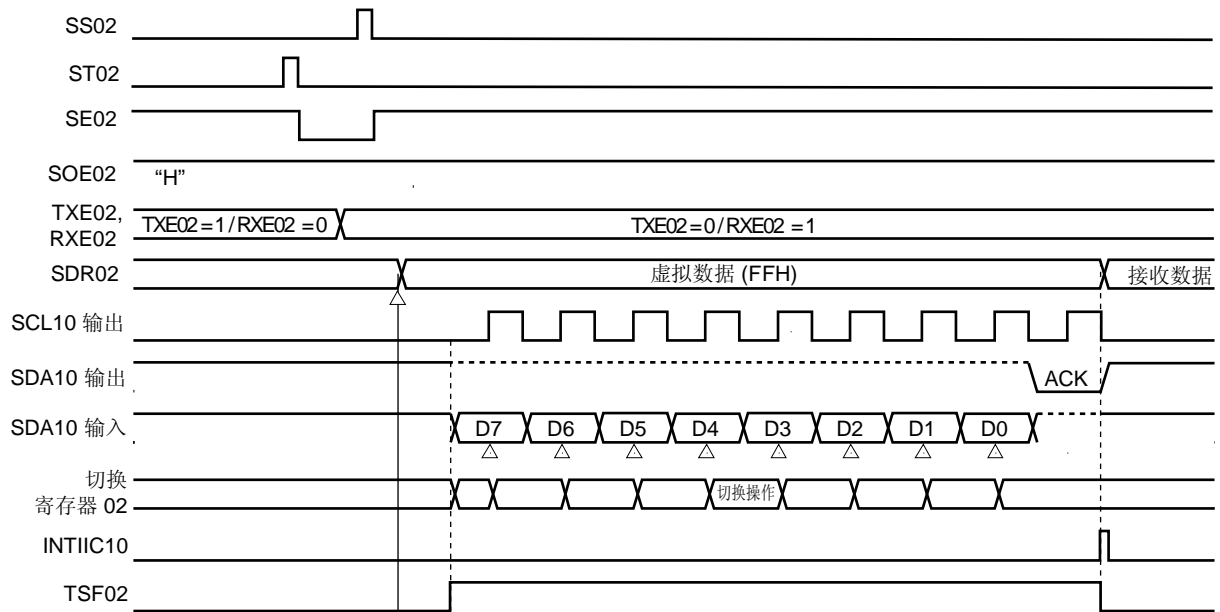
注 该值由通信操作过程中的通信数据决定。

备注 □: 在 IIC 模式下固定设置, ■.: 禁止设置 (设为初始值)
 ×: 此模式下不能使用的位 (不在任何模式下使用时, 设为初始值)
 0/1: 根据用户的使用设置为 0 或 1

(2) 处理流程

图 12-95. 数据接收的时序图

(a) 启动数据接收



<R>

(b) 接收最后数据

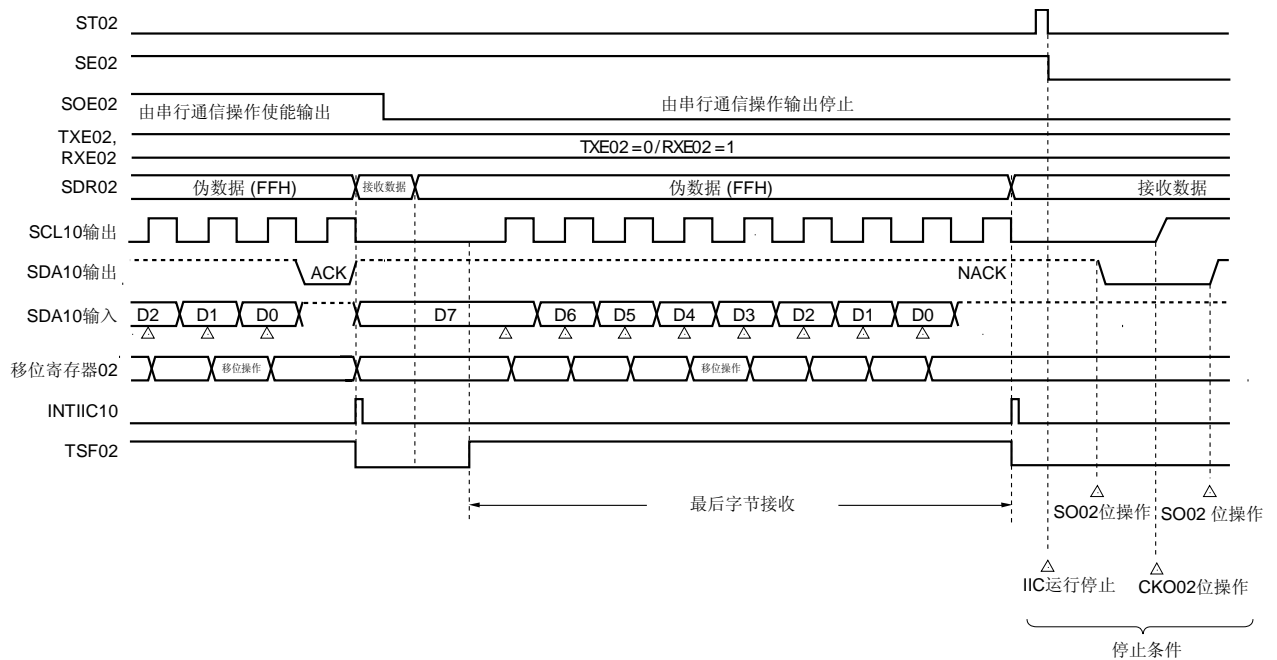
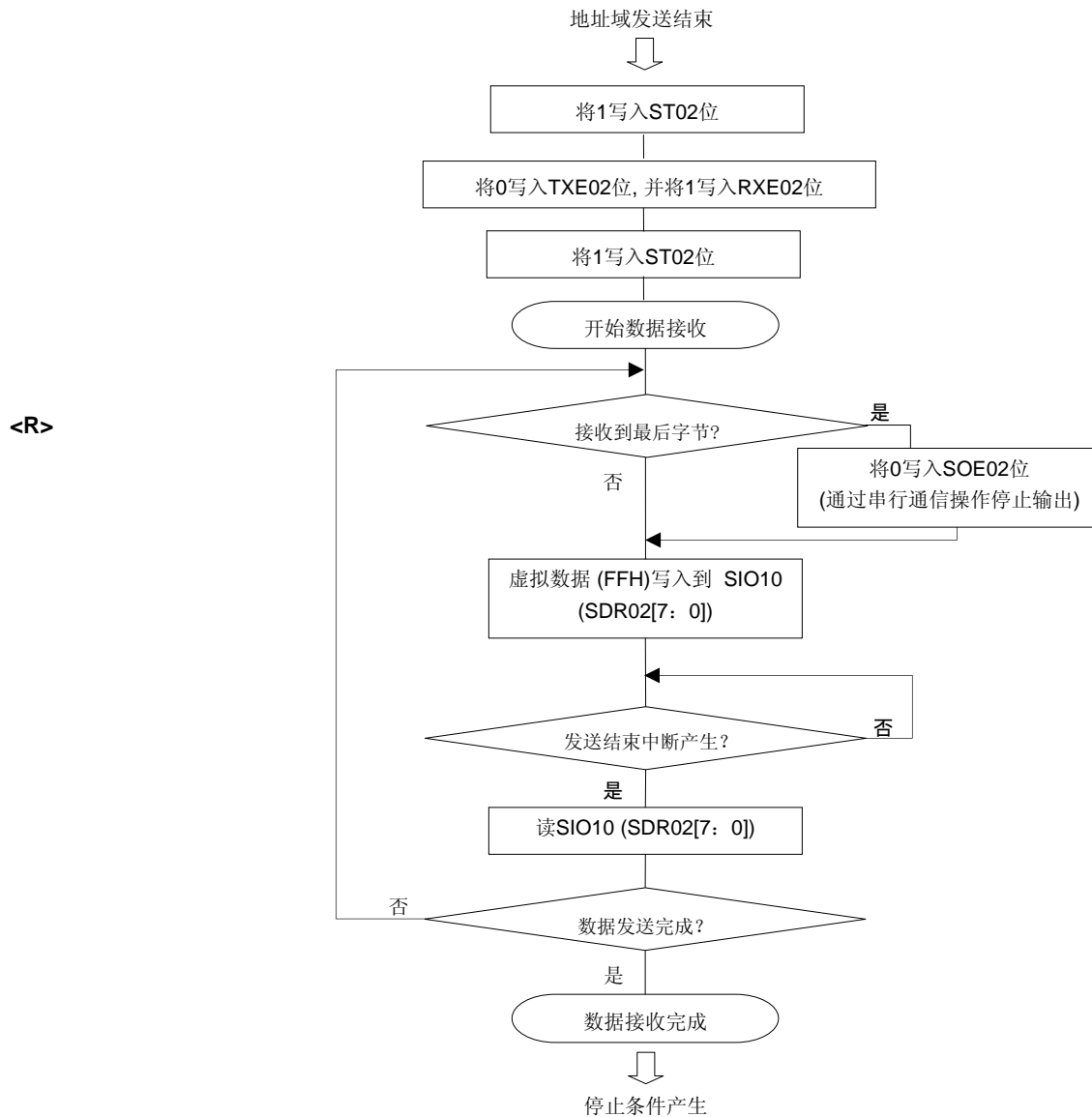


图 12-96. 数据接收的流程图



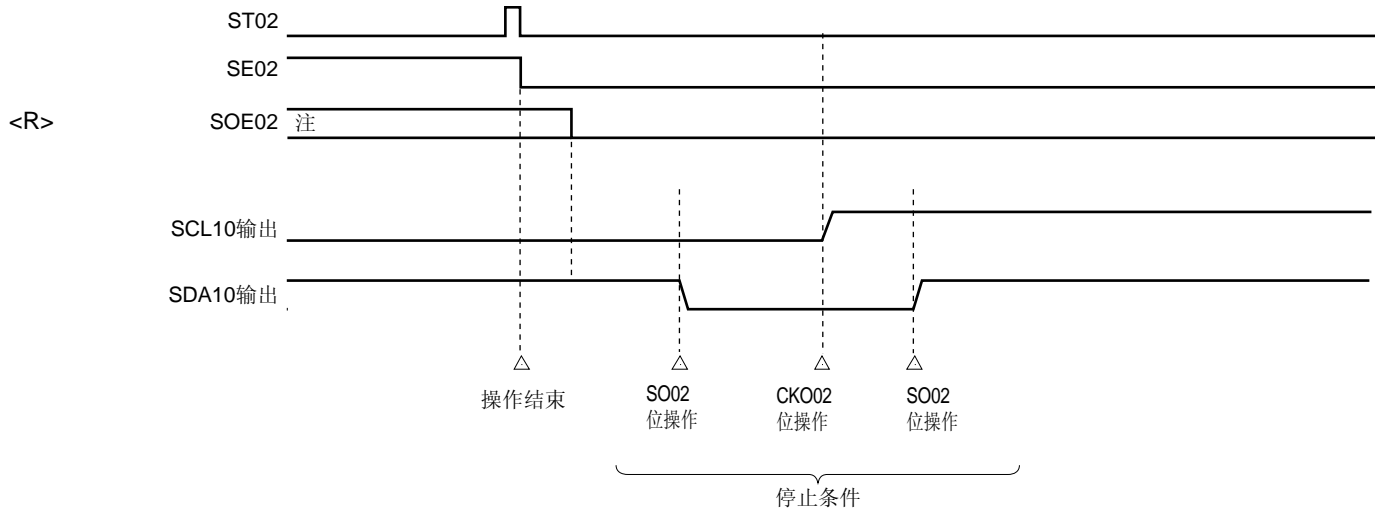
<R> **注意事项** 当接收到最后数据时, **ACK** 不输出(**NACK**)。通过设置“1”至 **ST02** 位来停止操作并产生停止条件, 从而完成通信。

12.7.4 停止条件的产生

全部数据发送到从设备或者从从设备读取到全部数据后，产生一个停止条件并且总线释放。

(1) 处理流程

图 12-97. 停止条件的产生的时序图



注 接收操作期间，接收最后数据之前，SOE02 位被清零。

图 12-98. 停止条件的产生的流程图



12.7.5 计算发送速率

简易 I²C (IIC10) 通信的发送速率可以按照下列表达式来计算。

$$(\text{传输率}) = \{\text{目标通道的工作时钟(MCK)频率}\} \div (\text{SDR02}[15:9] + 1) \div 2$$

备注 SDR02[15:9]的值是 SDR02 寄存器的第 15 位～第 9 位的值 (0000000B～1111111B)，因此它的取值范围为 0～127。

工作时钟 (MCK) 由串行时钟选择寄存器 0 (SPS0) 和串行模式寄存器 02 (SMR02) 的第 15 位 (CKS02) 来确定。

表 12-4. 工作时钟的选择

SMR02 寄存器	SPS0 寄存器								工作时钟(MCK) ^❶	
CKS02	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000		f _{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f _{CLK}	20 MHz
	X	X	X	X	0	0	0	1	f _{CLK} /2	10 MHz
	X	X	X	X	0	0	1	0	f _{CLK} /2 ²	5 MHz
	X	X	X	X	0	0	1	1	f _{CLK} /2 ³	2.5 MHz
	X	X	X	X	0	1	0	0	f _{CLK} /2 ⁴	1.25 MHz
	X	X	X	X	0	1	0	1	f _{CLK} /2 ⁵	625 kHz
	X	X	X	X	0	1	1	0	f _{CLK} /2 ⁶	313 kHz
	X	X	X	X	0	1	1	1	f _{CLK} /2 ⁷	156 kHz
	X	X	X	X	1	0	0	0	f _{CLK} /2 ⁸	78.1 kHz
	X	X	X	X	1	0	0	1	f _{CLK} /2 ⁹	39.1 kHz
	X	X	X	X	1	0	1	0	f _{CLK} /2 ¹⁰	19.5 kHz
	X	X	X	X	1	0	1	1	f _{CLK} /2 ¹¹	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 ^❷	
1	0	0	0	0	X	X	X	X	f _{CLK}	20 MHz
	0	0	0	1	X	X	X	X	f _{CLK} /2	10 MHz
	0	0	1	0	X	X	X	X	f _{CLK} /2 ²	5 MHz
	0	0	1	1	X	X	X	X	f _{CLK} /2 ³	2.5 MHz
	0	1	0	0	X	X	X	X	f _{CLK} /2 ⁴	1.25 MHz
	0	1	0	1	X	X	X	X	f _{CLK} /2 ⁵	625 kHz
	0	1	1	0	X	X	X	X	f _{CLK} /2 ⁶	313 kHz
	0	1	1	1	X	X	X	X	f _{CLK} /2 ⁷	156 kHz
	1	0	0	0	X	X	X	X	f _{CLK} /2 ⁸	78.1 kHz
	1	0	0	1	X	X	X	X	f _{CLK} /2 ⁹	39.1 kHz
	1	0	1	0	X	X	X	X	f _{CLK} /2 ¹⁰	19.5 kHz
	1	0	1	1	X	X	X	X	f _{CLK} /2 ¹¹	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 ^❷	
其它值									设置禁止	

- 注**
1. 当改变用于 f_{CLK} 的时钟 (通过改变系统时钟控制寄存器 (CKC) 的值) 时, 在停止串行阵列单元 (SAU) 的操作 (ST0 = 000FH) 之后, 才这样做。当选择 INTTM02 为工作时钟时, 同样停止定时器阵列单元 TAUS (TT0 = 00FFH)。
 2. 利用 SPS0 寄存器, 通过设置 TAUS 的 TIS0 寄存器的 TIS02 位为 1, 为输入时钟选择 f_{SUB}/4, 并且选择 INTTM02, 可以在副系统时钟的一个固定分频比时操作 SAU, 无论 f_{CLK} 频率 (主系统时钟、副系统时钟) 如何。但是, 当改变 f_{CLK}, 必须按照上面注 1 描述的停止 SAU 和 TAUS。

备注 X: 不必考虑

这有一个 $MCK = f_{CLK} = 20\text{ MHz}$ 下设置 IIC 发送速率的例子。

IIC 发送模式 (预期发送速率)	$f_{CLK} = 20\text{ MHz}$			
	工作时钟(MCK)	SDR02[15:9]	计算出来的发送速率	预期发送速率的误差
100 kHz	f_{CLK}	99	100 kHz	0.0%
400 kHz	f_{CLK}	24	400 kHz	0.0%

12.8 错误的处理程序

下面图 12-99 ~12-101 是如果发生错误的处理程序。

图 12-99. 校验错误或者溢出错误的处理程序

软件操作	硬件状态	备注
读 SDR0n 寄存器	→ BFF = 0, 并且通道 n 能够接收数据。	在错误处理中如果完成下一次接收防止溢出错误。
读 SSR0n 寄存器		定义错误类型并且读取用于清除错误标志的值。
写入 SIR0n 寄存器。	→ 清除错误标志。	只能在读取过程中, 通过将 SSR0n 寄存器中读出的值直接写入 SIR0n 寄存器来清除错误。

图 12-100. 帧错误时的处理程序

软件操作	硬件状态	备注
读 SDR0n 寄存器	→ BFF = 0, 并且通道 n 能够接收数据。	在错误处理中如果完成下一次接收防止溢出错误。
读 SSR0n 寄存器		定义错误类型并且读取用于清除错误标志的值。
写入 SIR0n 寄存器。	→ 清除错误标志。	只能在读取过程中, 通过将 SSR0n 寄存器中读出的值直接写入 SIR0n 寄存器来清除错误。
设置 ST0n 位为 1。	→ SE0n = 0, 通道 n 停止操作。	
和其它部分通信同步		由于起始位发生移位的帧错误, 和其它部分通信的同步将重建, 并恢复通信。
设置 SS0n 位为 1。	→ SE0n = 1, 允许通道 n 操作。	

备注 n: 单元编号 (n = 0~3)

图 12-101. 简易 I²C 模式中校验错误(ACK 错误)的处理程序

软件操作	硬件状态	备注
读 SDR02 寄存器	→ BFF = 0, 允许通道 2 接收数据。	在错误处理中如果完成下一次接收防止溢出错误。
读 SSR02 寄存器		定义错误类型并且读取用于清除错误标志的值。
写入 SIR02 寄存器。	→ 清除错误标志。	只能在读取过程中, 通过将 SSR02 寄存器中读出的值直接写入 SIR02 寄存器来清除错误。
设置 ST02 位为 1。	→ SE02 = 0, 通道 2 停止操作。	因为没有返回 ACK 从设备没有准备好接收。因此, 产生一个停止条件, 总线释放, 通信从启动条件重新开始。或者产生一个复位条件并且可以重新从地址发送开始。
产生停止条件		
产生启动条件		
设置 SS02 位为 1。	→ SE02 = 1, 允许通道 2 操作。	

12.9 寄存器设置和引脚之间的关系

表 12-5～表 12-8 所示为串行阵列单元 0 和 1 的每个通道的寄存器设置和引脚之间的关系。

12-5. 寄存器设置和引脚之间的关系（通道 0：CSI00、UART0 发送）

SE0 0 ^{注 1}	MD 002	MD0 01	SOE 00	SO0 0	CKO 00	TXE 00	RXE 00	PM 75	P75	PM7 4 ^{注 2}	P74 ^{注 2}	PM 73	P73	操作模式	引脚功能		
															SCK00/TI 11/P75	SI00/RxD0/T I10/P74 ^{注 2}	SO00/ TxD0/TO10/ P73
0	0	0	0	1	1	0	0	x 注 3	x 注 3	x 注 3	x 注 3	x 注 3	x 注 3	操作停止模式	TI11/P75	TI10/P74 RxD0/TI10/P 74	TO10/P73
	0	1															
1	0	0	0	1	1	0	1	1	x	1	x	x 注 3	x 注 3	CSI00 从机接收	SCK00 (输入)	SI00	TO10/P73
			1	0/1 注 4	1	1	0	1	x	x 注 3	x 注 3	0	1	CSI00 从机发送	SCK00 (输入)	TI10/P74	SO00
			1	0/1 注 4	1	1	1	1	x	1	x	0	1	从机 CSI00 发送 /接收	SCK00 (输入)	SI00	SO00
			0	1	0/1 注 4	0	1	0	1	1	x	x 注 3	x 注 3	CSI00 主机接收	SCK00 (输出)	SI00	TO10/P73
			1	0/1 注 4	0/1 注 4	1	0	0	1	x 注 3	x 注 3	0	1	CSI00 主机发送	SCK00 (输出)	TI10/P74	SO00
			1	0/1 注 4	0/1 注 4	1	1	0	1	1	x	0	1	主机 CSI00 发送 /接收	SCK00 (输出)	SI00	SO00
	0	1	1	0/1 注 4	1	1	0	x 注 3	x 注 3	x 注 3	x 注 3	0	1	UART0 发送 注 5	TI11/P75	RxD0/TI10/P 74	TxD0

- 注
1. SE0 寄存器是只读寄存器，使用 SS0 和 ST0 寄存器来设置它。
 2. 当通道 1 被设置成 UART0 接收，该引脚变成 RxD0 功能引脚（参见表 12-8）。在这种情况下，通道 0 必须选择为操作停止模式或 UART0 发送。
 3. 该引脚可以被设置为端口功能引脚。
 4. 取决于通信操作，该项为 0 或 1。详情请参阅 12.3 (12) 串行输出寄存器 0 (SO0)。
 5. 当成对使用 UART0 发送和接收时，设置通道 1 为 UART0 接收（参阅表 12-8）。

备注 X: 不必考虑

12-6. 寄存器设置和引脚之间的关系（通道 1: CSI01、UART0 接收）

SE0 1 ^{注 1}	MD 012	MD0 11	SOE 01	SO 01	CKO 01	TXE 01	RXE 01	PM 72	P72	PM 71	P71	PM 70	P70	PM7 4 ^{注 2}	P74 ^{注 2}	操作模式	引脚功能			
																	SCK01/ INTP6/ P72	SI01/IN TP5/P7 1	SO01/I NTP4/ P70	SI00/RxD0/ TI10/P74 ^{注 2}
0	0	0	0	1	1	0	0	x ^{注 3}	x ^{注 3}	x ^{注 3}	x ^{注 3}	x ^{注 3}	x ^{注 3}	x ^{注 3}	x ^{注 3}	操作停止模 式	INTP6/ P72	INTP5/ P71	INTP4/ P70	SI00/TI10/P 74
	0	1																		
1	0	0	0	1	1	0	1	1	x	1	x	x ^{注 3}	x ^{注 3}	x ^{注 3}	x ^{注 3}	CSI01 从机 接收	SCK01 (输入)	SI01	INTP4/ P70	SI00/TI10/P 74
			1	0/1 ^{注 4}	1	1	0	1	x	x ^{注 3}	x ^{注 3}	0	1	x ^{注 3}	x ^{注 3}	CSI01 从机 发送	SCK01 (输入)	INTP5/ P71	SO01	SI00/TI10/P 74
			1	0/1 ^{注 4}	1	1	1	1	x	1	x	0	1	x ^{注 3}	x ^{注 3}	CSI01 从机 发送/接收	SCK01 (输入)	SI01	SO01	SI00/TI10/P 74
			0	1	0/1 ^{注 4}	0	1	0	1	1	x	x ^{注 3}	x ^{注 3}	x ^{注 3}	x ^{注 3}	CSI01 主机 接收	SCK01 (输出)	SI01	INTP4/ P70	SI00/TI10/P 74
			1	0/1 ^{注 4}	0/1 ^{注 4}	1	0	0	1	x ^{注 3}	x ^{注 3}	0	1	x ^{注 3}	x ^{注 3}	CSI01 主机 发送	SCK01 (输出)	INTP5/ P71	SO01	SI00/TI10/P 74
			1	0/1 ^{注 4}	0/1 ^{注 4}	1	1	0	1	1	x	0	1	x ^{注 3}	x ^{注 3}	CSI01 主机 发送/接收	SCK01 (输出)	SI01	SO01	SI00/TI10/P 74
	0	1	0	1	1	0	1	x ^{注 3}	x ^{注 3}	x ^{注 3}	x ^{注 3}	x ^{注 3}	x ^{注 3}	1	x	UART0 接 收 ^{注 5, 6}	INTP6/ P72	INTP5/ P71	INTP4/ P70	RxD0

- 注**
1. SE0 寄存器是只读寄存器，使用 SS0 和 ST0 寄存器来设置它。
 2. 当通道 1 被设置成 UART0 接收，该引脚变成 RxD0 功能引脚。在这种情况下，将通道 0 设置为操作停止模式或 UART0 发送（参见表 12-6）。
当通道 0 被设置为 CSI00，该引脚就不能用作 RxD0 功能引脚。在这种情况下，将通道 1 设置为操作停止模式或 CSI01。
 3. 该引脚可以被设置为端口功能引脚。
 4. 取决于通信操作，该项为 0 或 1。详情请参阅 12.3 (12) 串行输出寄存器 0 (SO0)。
 5. 当成对使用 UART0 发送和接收时，设置通道 0 为 UART0 发送（参阅表 12-6）。
 6. UART0 接收过程中，也必须设置通道 0 的 SMR00 寄存器。详细信息，请参阅 12.6.2 (1) 寄存器设置。

备注 X: 不必考虑

表 12-7. 寄存器设置和引脚之间的关系 (通道 2: CSI10、UART1 发送、IIC10)

SE0 ^{注1}	MD022	MD021	SOE02	SO02	CKO02	TXE02	RXE02	PM32	P32	PM3 ^{注2}	P31 ^{注2}	PM30	P30	操作模式	引脚功能		
															SCK10/ SCL10/ INTP2/P32	SI10/SDA10/ RxD1/INTP1/ TI09/P31 ^{注2}	SO10/ TxD1/ TO11/P30
0	0	0	0	1	1	0	0	x ^{注3}	x ^{注3}	x ^{注3}	x ^{注3}	x ^{注3}	x ^{注3}	操作停止模式	INTP2/P32	INTP1/TI09/P31	TO11/P30
	0	1														RxD1/INTP1/ TI09/P31	
	1	0														INTP1/TI09/P31	
1	0	0	0	1	1	0	1	1	x	1	x	x ^{注3}	x ^{注3}	CSI10 从机接收	SCK10 (输入)	SI10	TO11/P30
			1	0/1 ^{注4}	1	1	0	1	x	x ^{注3}	x ^{注3}	0	1	CSI10 从机发送	SCK10 (输入)	INTP1/TI09/P31	SO10
			1	0/1 ^{注4}	1	1	1	1	x	1	x	0	1	CSI10 从机发送/ 接收	SCK10 (输入)	SI10	SO10
			0	1	0/1 ^{注4}	0	1	0	1	1	x	x ^{注3}	x ^{注3}	CSI10 主机接收	SCK10 (输出)	SI10	TO11/P30
			1	0/1 ^{注4}	0/1 ^{注4}	1	0	0	1	x ^{注3}	x ^{注3}	0	1	CSI10 主机发送	SCK10 (输出)	INTP1/TI09/P31	SO10
			1	0/1 ^{注4}	0/1 ^{注4}	1	1	0	1	1	x	0	1	CSI10 主机发送/ 接收	SCK10 (输出)	SI10	SO10
	0	1	1	0/1 ^{注4}	1	1	0	x ^{注3}	x ^{注3}	x ^{注3}	x ^{注3}	0	1	UART1 传输 ^{注5}	INTP2/P32	RxD1/INTP1/ TI09/P31	TxD1
0	1	0	0	0/1 ^{注6}	0/1 ^{注6}	0	0	0	1	0	1	x ^{注3}	x ^{注3}	IIC10 启动条件	SCL10	SDA10	TO11/P30
						1	0										
						0	1										
			1	0/1 ^{注4}	0/1 ^{注4}	1	0	0	1	0	1	x ^{注3}	x ^{注3}	IIC10 地址域发送	SCL10	SDA10	TO11/P30
			1	0/1 ^{注4}	0/1 ^{注4}	1	0	0	1	0	1	x ^{注3}	x ^{注3}	IIC10 数据发送	SCL10	SDA10	TO11/P30
1			1	0/1 ^{注4}	0/1 ^{注4}	0	1	0	1	0	1	x ^{注3}	x ^{注3}	IIC10 数据接收	SCL10	SDA10	TO11/P30
			0	0/1 ^{注7}	0/1 ^{注7}	0	0	0	1	0	1	x ^{注3}	x ^{注3}	IIC10 停止条件	SCL10	SDA10	TO11/P30
						1	0										
						0	1										

- 注
1. SE0 寄存器是只读寄存器，使用 SS0 和 ST0 寄存器来设置它。
 2. 当通道 3 被设置成 UART1 接收，该引脚变成 RxD1 功能引脚（参见表 12-10）。在这种情况下，通道 2 必须选择为操作停止模式或 UART1 发送。
 3. 该引脚可以被设置为端口功能引脚。

4. 取决于通信操作，该项为 0 或 1。详情请参阅 12.3 (12) 串行输出寄存器 0 (SO0)。
5. 当成对使用 UART1 发送和接收时，设置通道 3 为 UART1 接收（参阅表 12-10）。
6. 在产生启动条件之前，将 CKO02 位设为 1。当启动条件产生时，将 SO02 位由 1 清除为 0。
7. 在产生停止条件之前，将 CKO02 位设为 1。当停止条件产生时，将 SO02 位由 0 设为 1。

备注 X: 不必考虑

表 12-8. 寄存器设置和引脚之间的关系（通道 3: UART1 接收）

SE03 ^{注1}	MD032	MD031	TXE03	RXE03	PM31 ^{注2}	P31 ^{注2}	操作模式	引脚功能
								SI10/SDA10/RxD1/INTP1/TI09/P31 ^{注2}
0	0	1	0	0	x ^{注3}	x ^{注3}	操作停止模式	SI10/SDA10/INTP1/TI09/P31 ^{注2}
1	0	1	0	1	1	x	UART1 接收 ^{注4, 5}	RxD1

- 注
1. SE0 寄存器是只读寄存器，使用 SS0 和 ST0 寄存器来设置它。
 2. 当通道 3 设置成 UART1 接收，该引脚变成 RxD1 功能引脚。在这种情况下，将通道 2 设置为操作停止模式或 UART1 发送（参见表 12-9）。
当通道 2 被设置为 CSI10 或 IIC10，该引脚就不能当作 RxD1 功能引脚使用。在这种情况下，将通道 3 设置为操作停止模式。
 3. 该引脚可以设置为端口功能引脚。
 4. 当成对使用 UART1 发送和接收时，设置通道 2 为 UART1 发送（参阅表 12-9）。
 5. UART1 接收过程中，也必须设置通道 2 的 SMR02 寄存器。详细信息，请参阅 12.6.2 (1) 寄存器设置。

备注 X: 不必考虑

13.1 乘法器/除法器的功能

乘法器/除法器的功能如下。

- 16 位 × 16 位 = 32 位 (乘法)
- 32 位 ÷ 32 位 = 32 位, 32 位余数 (除法)

13.2 乘法器/除法器的配置

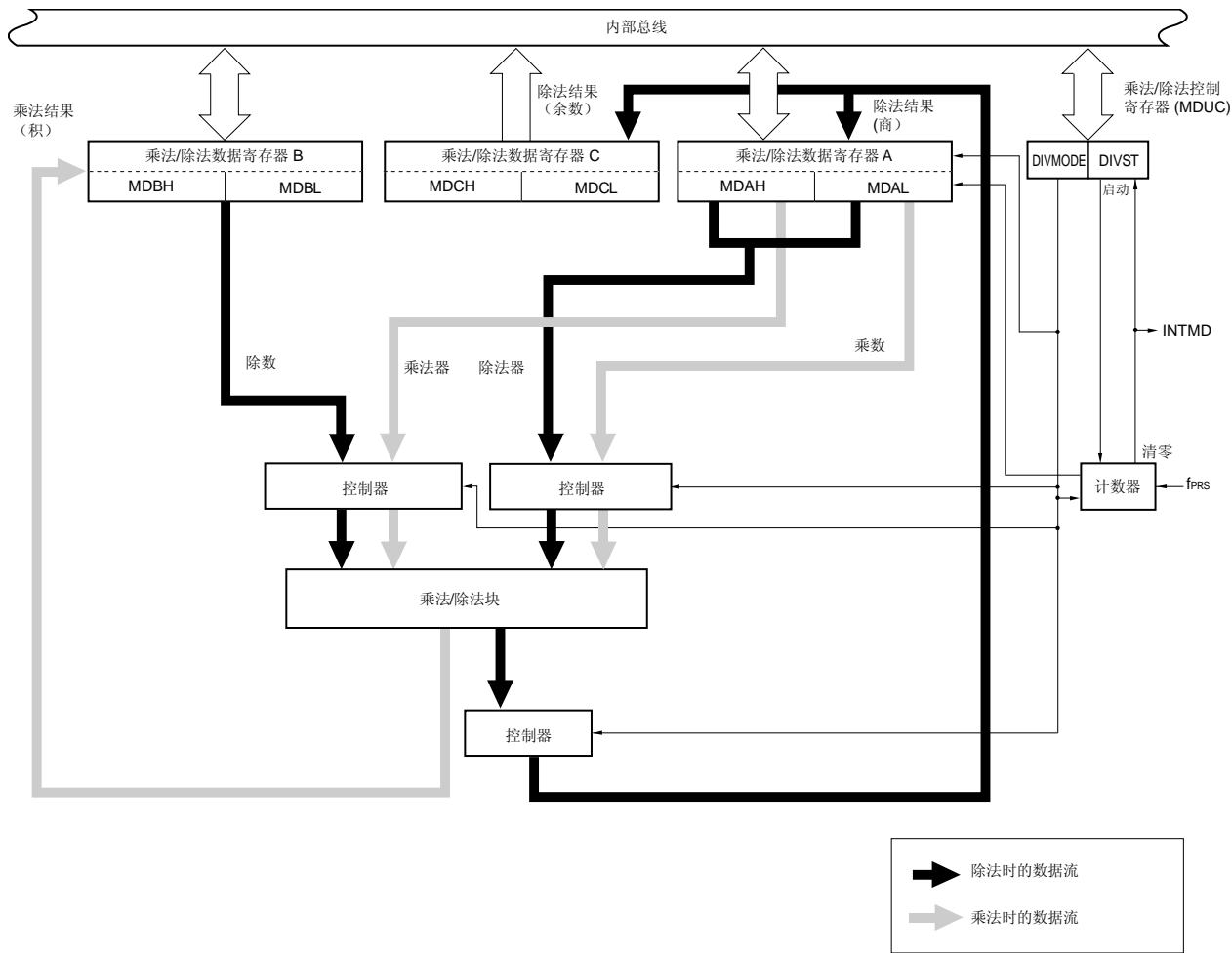
乘法器/除法器包含下列硬件。

表 13-1. 乘法器/除法器的配置

项目	配置
寄存器	乘法/除法数据寄存器 A (L) (MDAL)
	乘法/除法数据寄存器 A (H) (MDAH)
	乘法/除法数据寄存器 B (L) (MDBL)
	乘法/除法数据寄存器 B (H) (MDBH)
	乘法/除法数据寄存器 C (L) (MDCL)
	乘法/除法数据寄存器 C(H)(MDCH)
控制寄存器	乘法/除法控制寄存器 (MDUC)

图 13-1 所示为乘法器/除法器的框图。

图 13-1. 乘法器/除法器的框图



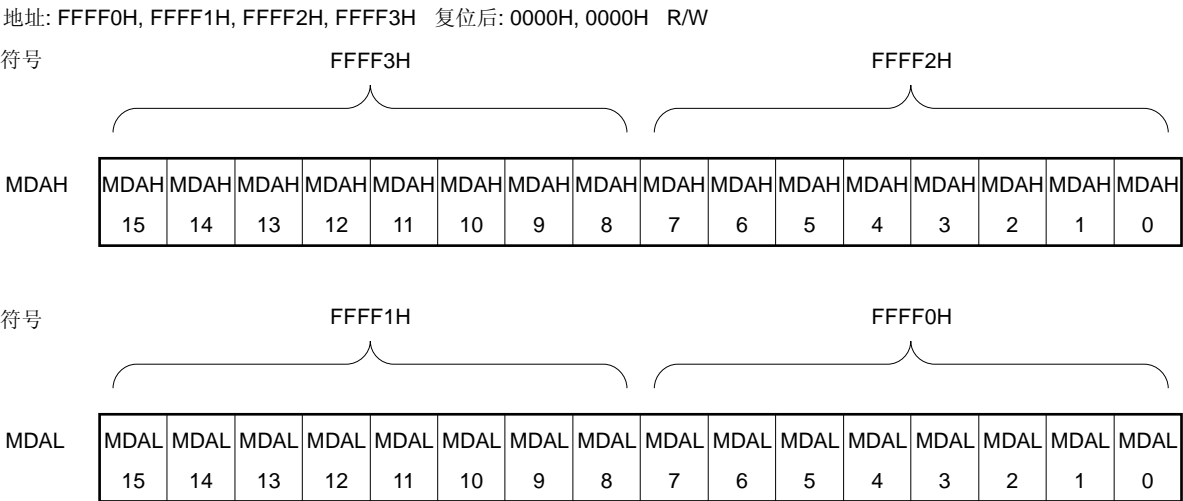
(1) 乘法/除法数据寄存器 A (MDAH, MDAL)

MDAH 和 MDAL 寄存器设置的值用于乘法或除法运算并存放运算结果。在乘法模式下设置乘数和被乘数，在除法模式下设置被除数。另外，在除法模式下运算数据（商）存放在 MDAH 和 MDAL 寄存器中。

MDAH 和 MDAL 可以通过 16 位操作指令来设置。

复位信号将这些寄存器清零为 0000H。

图 13-2. 乘法/除法数据寄存器 A(MDAH, MDAL)的格式



- 注意事项
- 1. 在除法运算过程中不要重写 MDAH 和 MDAL（当乘法/除法控制寄存器（MDUC）为 81H 时）。此时运算将被执行，但是运算结果将是一个不确定值。
 - 2. 不保证在除法运算过程中（当 MDUC 为 81H 时）读取的 MDAH 和 MDAL 的值。

下表所示为执行运算时 MDAH 和 MDAL 的功能。

表 13-2. 执行运算时 MDAH 和 MDAL 的功能

	操作模式	设置	操作结果
0	乘法模式	MDAH: 乘数 MDAL: 被乘数	—
1	除法模式	MDAH: 除数（高16位） MDAL: 被除数（低16位）	MDAH: 除法结果（商）高16位 MDAL: 除法结果（商）低16位

备注 DIVMODE: 乘法/除法控制寄存器（MDUC）的第 7 位

(2) 乘法/除法数据寄存器 B (MDBL, MDBH)

MDBH 和 MDBL 寄存器设置的值用于乘法或除法运算并存放运算结果。在乘法模式下存放运算结果（积），在除法模式下设置除数。

MDBH 和 MDBL 可以通过 16 位操作指令来设置。

复位信号将这些寄存器清零为 0000H。

图 13-3. 乘法/除法数据寄存器 B(MDBH,MDBL)的格式

地址: FFFF4H, FFFF5H, FFFF6H, FFFF7H 复位后: 0000H, 0000H R/W



- 注意事项
1. 在除法运算过程中不要重写 MDBH 和 MDBL (当乘法/除法控制寄存器 (MDUC) 为 81H 时)。运算结果将不确定。
 2. 在除法模式下不要将 MDBH 和 MDBL 设置为 0000H。否则运算结果将会是个不确定值。

下表所示为执行运算时 MDBH 和 MDBL 的功能。

表 13-3. 运算执行中 MDBH 和 MDBL 的功能

	操作模式	设置	操作结果
0	乘法模式	—	MDBH: 乘法结果（积） 高16位
			MDBL: 乘法结果（积） 低16位
1	除法模式	MDBH: 除数（高16位）	—
		MDBL: 被除数（低16位）	

备注 DIVMODE: 乘法/除法控制寄存器 (MDUC) 的第 7 位

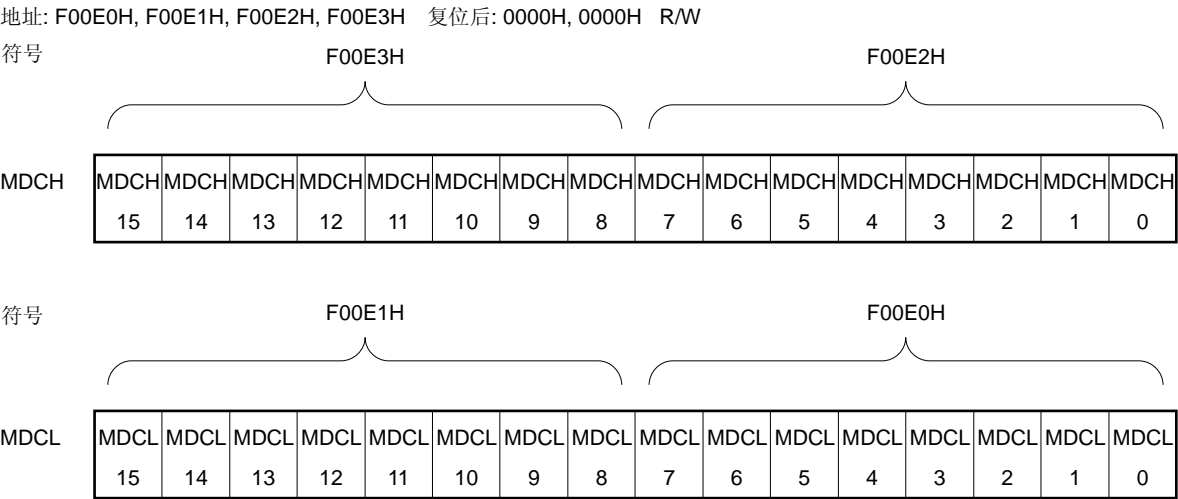
(3) 乘法/除法数据寄存器 C (MDCL, MDCH)

在除法模式下，MDCH 和 MDCL 寄存器存放运算结果的余数值。它们不可在乘法模式中使用。

MDCH 和 MDCL 可由 16 位操作指令读取。

复位信号将这些寄存器清零为 0000H。

图 13-4. 乘法/除法数据寄存器 C(MDCH, MDCL)的格式



注意事项 不保证在除法运算过程中（当乘法/除法控制寄存器 (MDUC) 为 81H 时）读取的 MDCH 和 MDCL 的值。

表 13-4. 执行运算时 MDCH 和 MDCL 的功能

	操作模式	设置	操作结果
0	乘法模式	—	—
1	除法模式	—	MDCH: 余数（高16位） MDCL: 余数（低16位）

备注 DIVMODE: 乘法/除法控制寄存器 (MDUC) 的第 7 位

如下，根据执行乘法和执行除法不同寄存器的配置也有所不同。

- 进行乘法时的配置寄存器
 <乘数 A> <乘数 B> <积>
 MDAL (第 15~0 位) × MDAH (第 15~0 位) = [MDBH (第 15~0 位), MDBL (第 15~0 位)]
- 进行除法时的配置寄存器
 <被除数> <除数>
 [MDAH (第 15~0 位), MDAL (第 15 位~第 0 位)] ÷ [MDBH (第 15~0 位), MDBL (第 15~0 位)] =
 <商> <余数>
 [MDAH (位 15~0), MDAL (位 15~0)] ... [MDCH (位 15~0), MDCL (位 15~0)]

13.3 乘法器/除法器的寄存器

乘法器/除法器由乘法/除法控制寄存器 (MDUC) 进行控制。

(1) 乘法/除法控制寄存器 (MDUC)

MDUC 是一个控制乘法器/除法器运算的 8 位寄存器。

MDUC 可以由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 13-5. 乘法/除法控制寄存器 (MDUC) 的格式

地址: F00E8H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	<0>
MDUC	DIVMODE	0	0	0	0	0	0	DIVST
DIVMODE	工作模式（乘法 / 除法）选择							
0	乘法模式							
1	除法模式							
DIVST ^注	除法操作开始 / 停止							
0	除法操作处理完成							
1	除法运算开始 / 除法运算处理中							

注 DIVST 在除法模式中只能设为(1)。在除法模式中，将 DIVST 设为(1)开始除法运算。当运算结束后，DIVST 自动清(0)。在乘法模式下，将乘数和被乘数分别设置到 MDAH 和 MDAL 后，运算自动开始。

- 注意事项 1. 在运算过程中不要重写 DIVMODE（当 DIVST 为 1 时）。如果被重写，运算结果将会是个不确定的值。
2. 在除法运算过程中（当 DIVST 为 1 时），DIVST 不能通过软件清(0)。

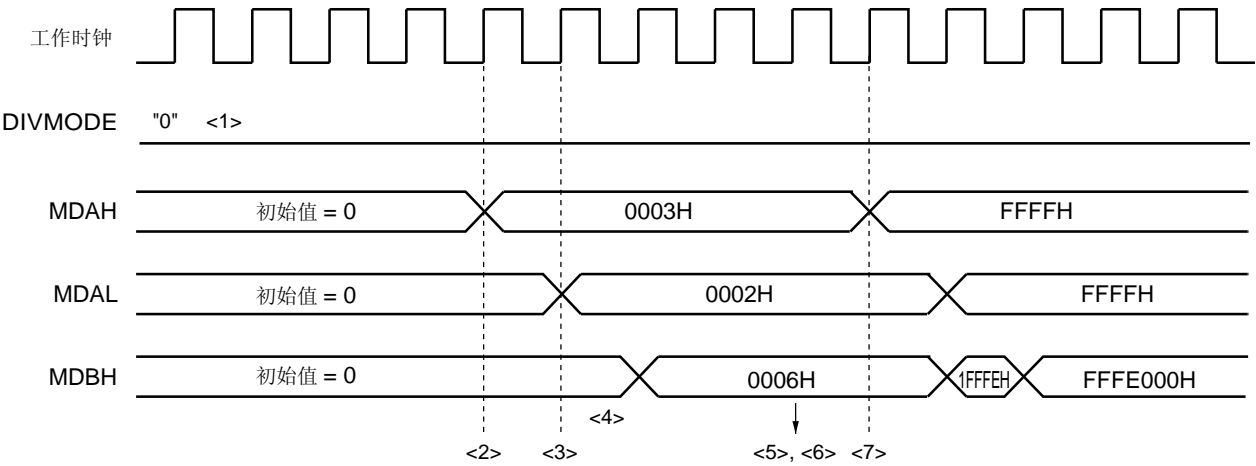
13.4 乘法器/除法器的操作

13.4.1 乘法运算

- 初始设置
 - <1> 设置乘法 / 除法控制寄存器 (MDUC) 的第 7 位 (DIVMODE) 为 0。
 - <2> 设置被乘数到乘法 / 除法数据寄存器 A (L) (MDAL)。
 - <3> 设置乘数到乘法 / 除法数据寄存器 A (H) (MDAH)。
(执行步骤<2>和<3>的顺序时, 没有优先顺序。当乘数和被乘数分别设置到 MDAH 和 MDAL 后, 乘法运算自动开始。)
- 操作处理期间
 - <4> 等待至少一个时钟。一个时钟发布后, 操作将结束。
- 运算结束
 - <5> 从乘法 / 除法数据寄存器 B (L) (MDBL)中读取乘积 (较低的 16 位)。
 - <6> 从乘法 / 除法数据寄存器 B (H) (MDBH)中读取乘积 (较高的 16 位)。
(执行步骤<5>和<6>的顺序没有任何优先顺序)
- 下一操作
 - <7> 要执行下一步乘法运算, 从**乘法运算**的"初始设置"开始。
 - <8> 要执行下一步除法操作, 从 **13.4.2 除法运算**的"初始设置"开始。

备注 步骤<1>~<7>与图 13-6 中的<1>~<7>相对应。

图 13-6. 乘法运算 (0003H × 0002H) 的时序图

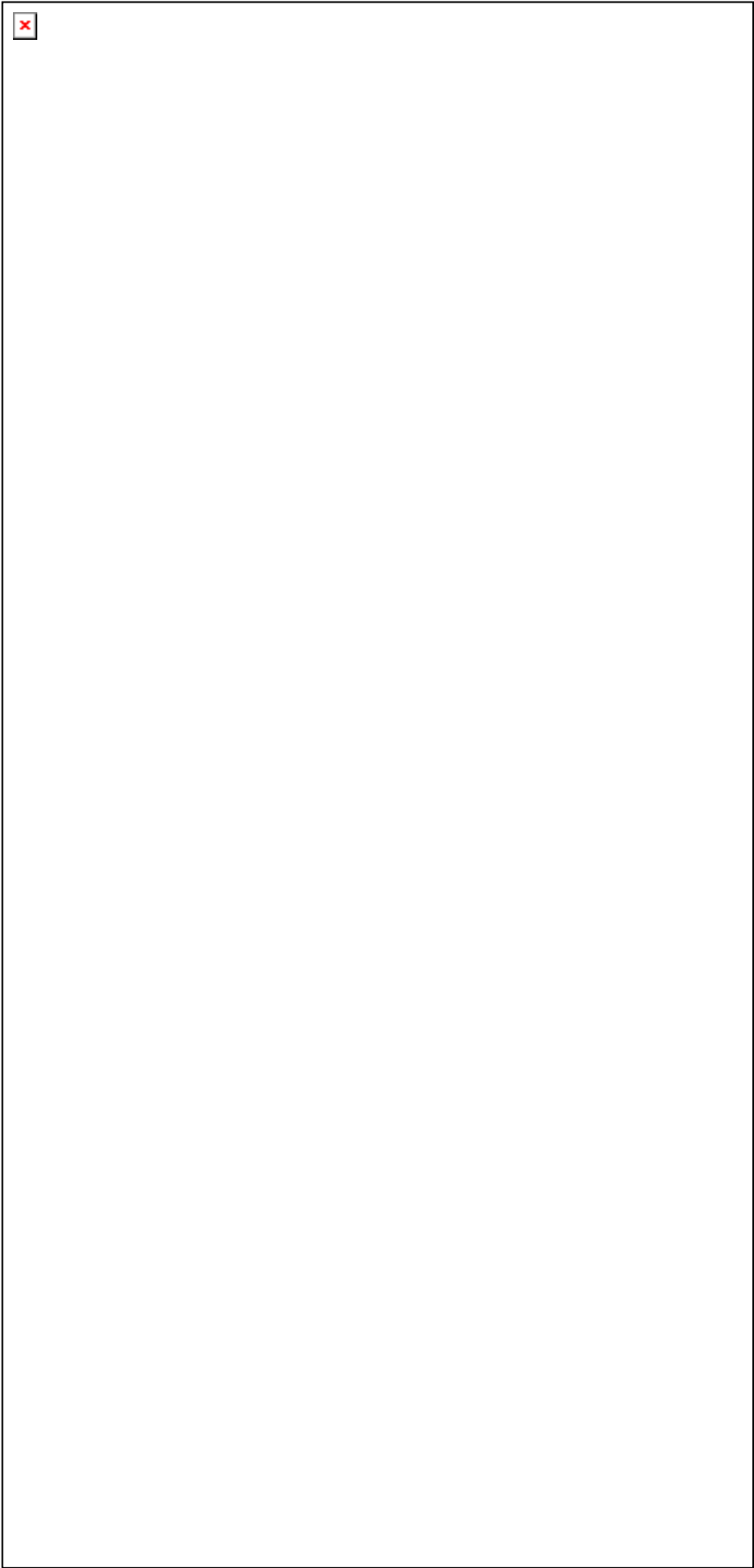


13.4.2 除法操作

- 初始设置
 - <1> 设置乘法 / 除法控制寄存器 (MDUC) 的第 7 位 (DIVMODE) 为 1。
 - <2> 设置被除数 (较高的 16 位) 到乘法 / 除法数据寄存器 A (H) (MDAH)。
 - <3> 设置被除数 (较低的 16 位) 到乘法 / 除法数据寄存器 A (L) (MDAL)。
 - <4> 设置除数 (较高的 16 位) 到乘法 / 除法数据寄存器 B (H) (MDBH)。
 - <5> 设置除数 (较低的 16 位) 到乘法 / 除法数据寄存器 B (L) (MDBL)。
 - <6> 设置 MDUC 的位 0 (DIVST) 为 1。
(执行步骤<2>~<5>的顺序时, 没有优先顺序)
- 操作处理期间
 - <7> 下列过程之一完成后, 运算将结束。
 - 至少等待 16 个时钟(当 16 个时钟发布后, 运算将结束。)
 - 检查 DIVST 是否被清除
 - 产生一个除法完成中断(INTMD)
 (运算过程中, MDBL、MDBH、MDCH 和 MDCL 的读取数值不被保证)
- 运算结束
 - <8> DIVST 被清 (0) 且产生一个中断请求信号(INTMD) (运算结束)。
 - <9> 从 MDAL 中读取商(较低的 16 位)。
 - <10> 从 MDAH 中读取商(较高的 16 位)。
 - <11> 从乘法 / 除法数据寄存器 C (L) (MDCL) 中读取余数(较低的 16 位)。
 - <12> 从乘法 / 除法数据寄存器 C (H) (MDCH) 中读取余数(较高的 16 位)。
(执行步骤<9>~<12>的顺序时, 没有优先顺序。)
- 下一操作
 - <13> 要执行下一步乘法运算, 从 13.4.1 乘法运算的"初始设置"开始。
 - <14> 要执行下一步除法运算, 从除法运算的"初始设置"开始。

备注 步骤<1>~<12>与图 13-7 中的<1>~<12>相对应

图 13-7. 除法运算(示例: $35 \div 6 = 5$, 余数 5)的时序图



第 14 章 DMA 控制器

μPD79F9211 具有内部 DMA（存储器直接访问）控制器。

数据能够在支持 DMA、SFRs 的外围硬件和内部 RAM 之间自动传输，而不通过 CPU。

这样，CPU 的正常内部操作和数据传输能够与 SFR 和内部 RAM 之间的传输并行执行，因此能够处理大容量的数据。另外，使用通信的实时控制、定时器和 A/D 也可以实现。

14.1 DMA 控制器

- DMA 通道个数：2
- 传输单位：8 或 16 位
- 最大传输单元：1024 次
- 传输类型：2 周期传输（传输期间，一次传输在 2 个时钟内完成且 CPU 停止处理。）
- 传输模式：单次传输模式
- 传输请求：可从下列外围硬件中断中选择
 - A/D 转换器
 - 串行接口（CSI00, CSI01, CSI10, UART0, UART1, 或 IIC10）
 - 定时器（通道 0, 1, 4, 或 5）
- 传输目标：在 SFR 内部 RAM 之间

这里是使用 DMA 的实例。

- 串行接口的连续传输
- 模拟数据的批传输
- 固定间隔捕捉 A/D 转换结果
- 固定间隔捕捉端口数值

14.2 DMA控制器的配置

DMA 控制器包括下列硬件。

表 14-1. DMA 控制器的配置

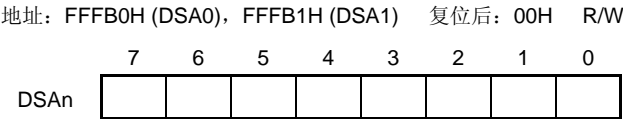
项目	配置
地址寄存器	• DMA SFR 地址寄存器 0, 1 (DSA0, DSA1) • DMA RAM 地址寄存器 0, 1 (DRA0, DRA1)
计数寄存器	• DMA 字节计数寄存器 0, 1 (DBC0, DBC1)
控制寄存器	• DMA 模式控制寄存器 0, 1 (DMC0, DMC1) • DMA 操作模式控制寄存器 0, 1 (DRC0, DRC1)

(1) DMA SFR 地址寄存器 n (DSAn)

此为 8 位寄存器，用来设置 DMA 通道 n 的传输源或者目的地的 SFR 地址。
SFR 地址的低 8 位设置在 FFF00H~FFFFFH[※]之间。
此寄存器并不自动增加，但固定为一个特定的数值。
16 位传输模式下，忽略最低位，并当作偶地址。
DSAn 可以进行 8 位读写。但是，DMA 传输期间不能写入。
复位后这个寄存器的值为 00H。

注 除了地址 FFFFEH，因为 PMC 寄存器分配在此。

图 14-1. DMA SFR 地址寄存器 n (DSAn)的格式

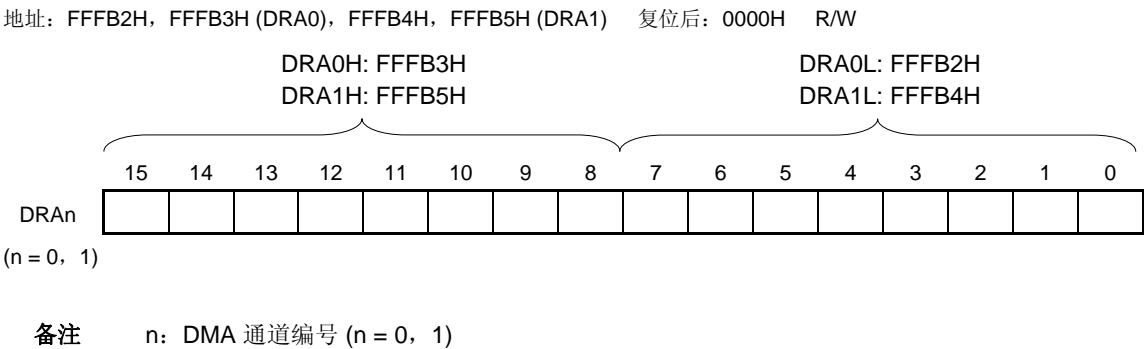


备注 n: DMA 通道编号 (n = 0, 1)

(2) DMA RAM 地址寄存器 n (DRAn)

此为 16 位寄存器，用来设置 DMA 通道 n 的传输源或目的地的 RAM 地址。
除通用寄存器外，内部 RAM 区域的地址，可被设置到该寄存器。
设置 RAM 地址的低 16 位。
此寄存器在 DMA 传输开始时自动增加。在 8 位传输模式下，+1 增加；在 16 位传输模式下，+2 增加。DMA 传输从设置在 DRAn 寄存器中的地址开始。当最后地址的数据传输完成时，DRAn 停止，其内部的值在 8 位传输模式下为最后地址+1，在 16 位传输模式下为最后地址+2。
16 位传输模式下，忽略最低位，并当作偶地址。
DRAn 可以进行 8 位或 16 位读写。但是，DMA 传输期间不能写入。
复位后这个寄存器的值为 0000H。

图 14-2. DMA RAM 地址寄存器 n (DRAn)的格式



(3) DMA 字节计数寄存器 n (DBCn)

此为 10 位寄存器，用来设置 DMA 通道 n 执行传输的次数值。执行 DMA 传输之前，确保将传输的次数值设置在此 DBCn 寄存器中（最多 1024 次）。

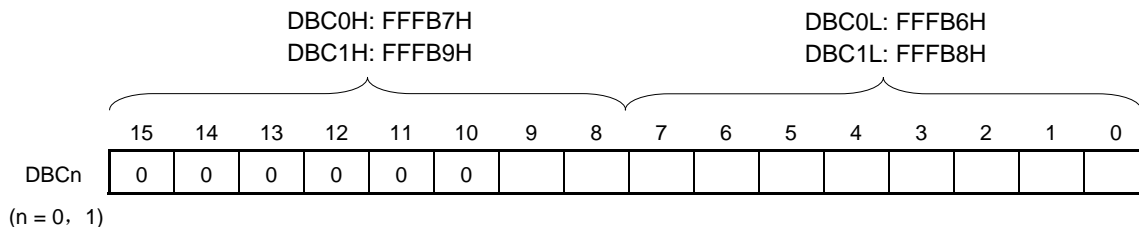
每一次 DMA 传输执行完成，此寄存器自动减少。在 DMA 传输期间读取该 DBCn 寄存器，可知道传输的剩余次数。

DBCn 可以进行 8 位或 16 位读写。但是，DMA 传输期间不能写入。

复位后这个寄存器的值为 0000H。

图 14-3. DMA 字节计数寄存器 n (DBCn) 的格式

地址: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) 复位后: 0000H R/W



DBCn[9:0]	传输次数值 (DBCn 写入时)	剩下的次数值 (DBCn 读取时)
000H	1024	完成传输或者等待 1024 次 DMA 传输
001H	1	等待剩下的 1 次 DMA 传输
002H	2	等待剩下的 2 次 DMA 传输
003H	3	等待剩下的 3 次 DMA 传输
⋮	⋮	⋮
3FEH	1022	等待剩下的 1022 次 DMA 传输
3FFH	1023	等待剩下的 1023 次 DMA 传输

注意事项 1. 确保将位 15~10 清“0”。

2. 如果指定了通用寄存器或者连续传输超出了内部 RAM 空间，通用寄存器或者 SFR 空间将被写入或者读取，导致这些空间的数据丢失。确保在内部 RAM 空间设置传输次数值。

备注 n: DMA 通道编号 (n = 0, 1)

14.3 控制DMA控制器的寄存器

DMA 控制器由下列寄存器控制。

- DMA 模式控制寄存器 n (DMCn)
- DMA 操作控制寄存器 n (DRCn)

备注 n: DMA 通道编号 (n = 0, 1)

(1) DMA 模式控制寄存器 n (DMCn)

DMCn 寄存器用来设置 DMA 通道 n 的传输模式。它用来选择传输方向、数据大小、挂起设置和启动源。第 7 位 (STGn) 是启动 DMA 的软件触发。

操作期间 (当 DSTn = 1 时), DMCn 的第 6 位、第 5 位和第 3~第 0 位禁止重写。

DMCn 可由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 14-4. DMA 模式控制寄存器 n (DMCn) (1/2)的格式

地址: FFFBAH (DMC0), FFFBBH (DMC1) 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn [※]	DMA 传输开始软件触发
0	没有触发操作
1	当 DMA 操作允许时 (DENn = 1), DMA 传输开始。
当 DMA 操作允许时 (DENn = 1), 通过在此位写入 1, 开始 DMA 传输。 当读取此位时, 一直为 0。	

DRSn	选择 DMA 传输方向
0	SFR 到内部 RAM
1	内部 RAM 到 SFR

DSn	制定 DMA 传输的传输数据大小
0	8 位
1	16 位

DWAITn	DMA 传输的挂起
0	DMA 开始请求之后, 执行 DMA 传输 (不保持挂起)。
1	如果有, 保持 DMA 启动请求挂起状态。
挂起的 DMA 传输可以通过清除 DWAITn 数值为 0 来开始。当 DWAITn 数值设置为 1 时, 有效的保持 DMA 传输挂起需要 2 个时钟。	

注 不管 IFCn0~IFCn3 为何数值, 软件触发 (STGn) 都可以使用。

备注 n: DMA 通道编号 (n = 0, 1)

图 14-4. DMA 模式控制寄存器 n (DMCn) (2/2)的格式

地址: FFFBAH (DMC0), FFFBBH (DMC1) 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

IFCn 3	IFCn 2	IFCn 1	IFCn 0	选择 DMA 启动源 [※]	
				触发信号	触发内容
0	0	0	0	—	通过中断禁止 DMA 传输。 (只允许软件触发。)
0	0	1	0	INTTM00	定时器通道 0 中断
0	0	1	1	INTTM01	定时器通道 1 中断
0	1	0	0	INTTM04	定时器通道 4 中断
0	1	0	1	INTTM05	定时器通道 5 中断
0	1	1	0	INTST0/INTCSI00	UART0 传输结束中断/CSI00 传输结束 中断
0	1	1	1	INTSR0/INTCSI01	UART0 接收结束中断/CSI01 传输结束 中断
1	0	0	0	INTST1/INTCSI10/INTIIC10	UART1 传输结束中断/ CSI10 传输结束中断/ IIC10 传输结束中断
1	0	0	1	INTSR1	UART1 接收结束中断
1	1	0	0	INTAD	A/D 转换结束中断
其它值				设置禁止	

注 不管 IFCn0~IFCn3 为何数值, 软件触发 (STGn) 都可以使用。

备注 n: DMA 通道编号 (n = 0, 1)

(2) DMA 操作控制寄存器 n (DRCn)

DRCn 寄存器用来禁止或允许 DMA 通道 n 的传输。

操作期间（当 DSTn = 1 时），此寄存器的第 7 位 (DENn) 禁止重写。

DRCn 可由 1 位或 8 位存储器操作指令设置。

复位后这个寄存器的值为 00H。

图 14-5. DMA 操作控制寄存器 n (DRCn) 的格式

地址：FFFBCH (DRC0)，FFFB DH (DRC1) 复位后：00H R/W

符号	<7>	6	5	4	3	2	1	<0>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA 操作允许标志
0	禁止 DMA 通道 n 的操作（停止 DMA 的工作时钟）。
1	允许 DMA 通道 n 的操作。
DMA 操作允许后 (DENn = 1)，当 DSTn = 1 时，DMAC 等待 1 个 DMA 触发。	

DSTn	DMA 传输模式标志
0	DMA 通道 n 的 DMA 传输完成。
1	DMA 通道 n 的 DMA 传输没有完成（仍在执行中）。
DMA 操作允许后 (DENn = 1)，当 DSTn = 1 时，DMAC 等待 1 个 DMA 触发。 当由 IFCn3 ~ IFCn0 设置的软件触发 (STGn) 或开始源触发输入时，DMA 传输开始。 当 DMA 传输完成之后，此位自动清零 0。执行期间将此位置 0，以强制中止 DMA 传输。	

注意事项 1. 当 DMA 传输完成时，DSTn 标志自动清除为 0。

只有当 DSTn = 0 时，允许写入 DENn 标志。当没有等待 DMA 中断 (INTDMA n) 产生而中止 DMA 传输时，要设置 DSTn 为 0，然后设置 DENn 为 0（详细信息，参考 14.5.5 软件强行中止）。

2. 当 OSMC 寄存器的 FSEL 位被设为 1 时，在设置后的至少三个时钟内，不要允许 (DENn = 1) DMA 操作。

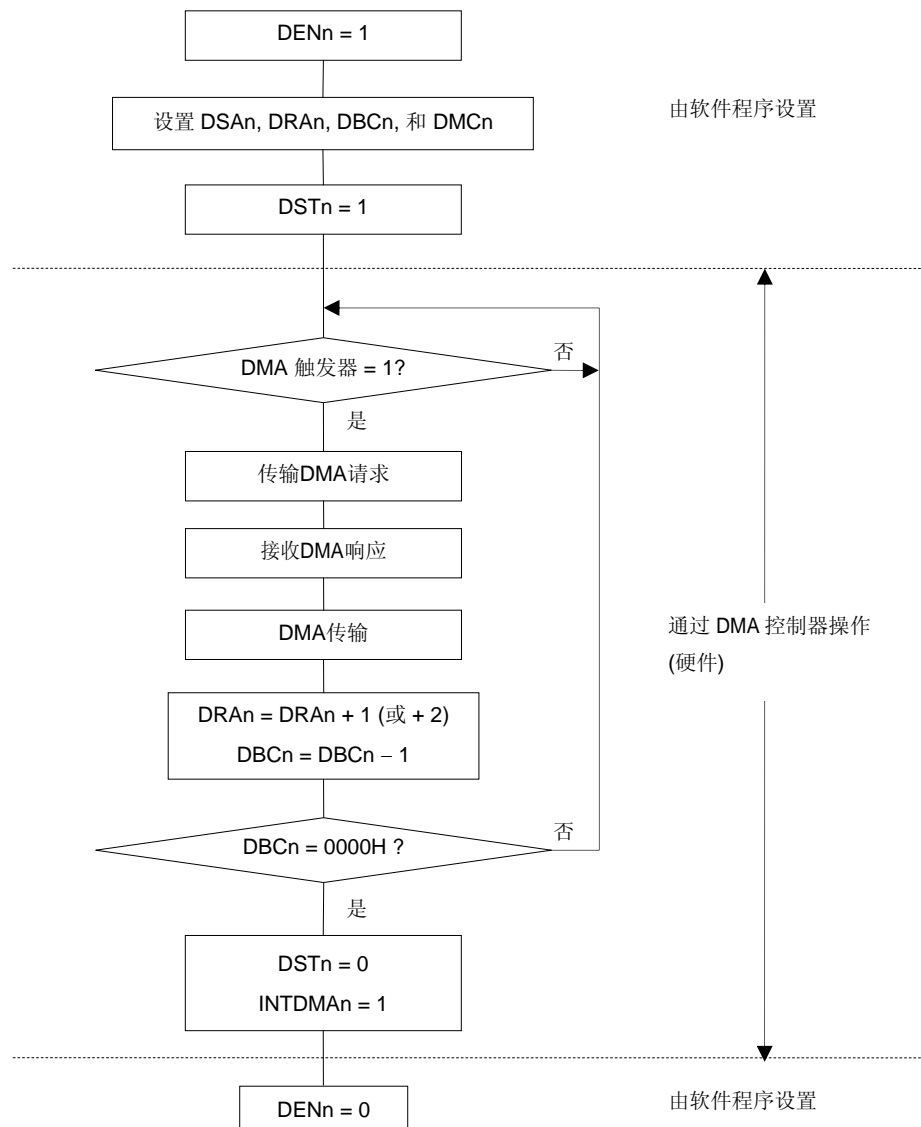
备注 n: DMA 通道编号 (n = 0, 1)

14.4 DMA控制器的操作

14.4.1 操作顺序

- <1> 当 $DEN_n = 1$ 时，DMA 控制器操作允许。其它寄存器写入之前，确保将 DEN_n 设置为 1。使用 8 位操作指令写入 80H。
- <2> 设置 1 个 SFR 地址，1 个 RAM 地址，传输次数值，并且在 $DSAn$ 、 $DRAn$ 、 $DBCn$ 和 $DMCn$ 寄存器中设置 DMA 传输的传输模式。
- <3> 当 $DSTn = 1$ 时，DMA 控制器等待 DMA 触发。使用 8 位操作指令写入 81H。
- <4> 当由 $IFCn3 \sim IFCn0$ 指定的软件触发 ($STGn$) 或启动源触发输入时，DMA 传输开始。
- <5> 当由 $DBCn$ 寄存器设置的传输次数值变为 0 时，传输完成，并且产生一个中断 ($INTDMA_n$)，传输自动停止。
- <6> 当 DMA 控制器不使用时，通过将 DEN_n 清除为 0，来停止 DMA 控制器的操作。

图 14-6. 操作顺序



备注 n: DMA 通道编号 (n = 0, 1)

14.4.2 传输模式

下面 4 种模式通过使用 DMCn 寄存器的第 6 位和第 5 位 (DRSn 和 DS_n) 选择 DMA 传输的模式。

DRSn	DS _n	DMA传输模式
0	0	从1字节数据的SFR（固定地址）到RAM（地址通过+1增加）的传输
0	1	从2字节数据的SFR（固定地址）到RAM（地址通过+2增加）的传输
1	0	从1字节数据的RAM（地址通过+1增加）到SFR（固定地址）的传输
1	1	从2字节数据的RAM（地址通过+2增加）到SFR（固定地址）的传输

通过这些传输模式，最多 1024 个字节的数据可以通过串行接口连续传输，A/D 转换的结果数据也可以连续传输，并且可以通过使用定时器在固定的时间间隔内扫描端口数据。

14.4.3 DMA传输中止

DBCn = 00H 并且 DMA 传输完成时，DSTn 位自动清除为 0。中断请求 (INTDMA_n) 产生，并且传输中止。

当 DSTn 位清除为 0 以强制中止 DMA 传输，DBCn 和 DRAn 寄存器在传输中止时保持其值。

如果传输强行中止，中断请求 (INTDMA_n) 不产生。

备注 n: DMA 通道编号 (n = 0, 1)

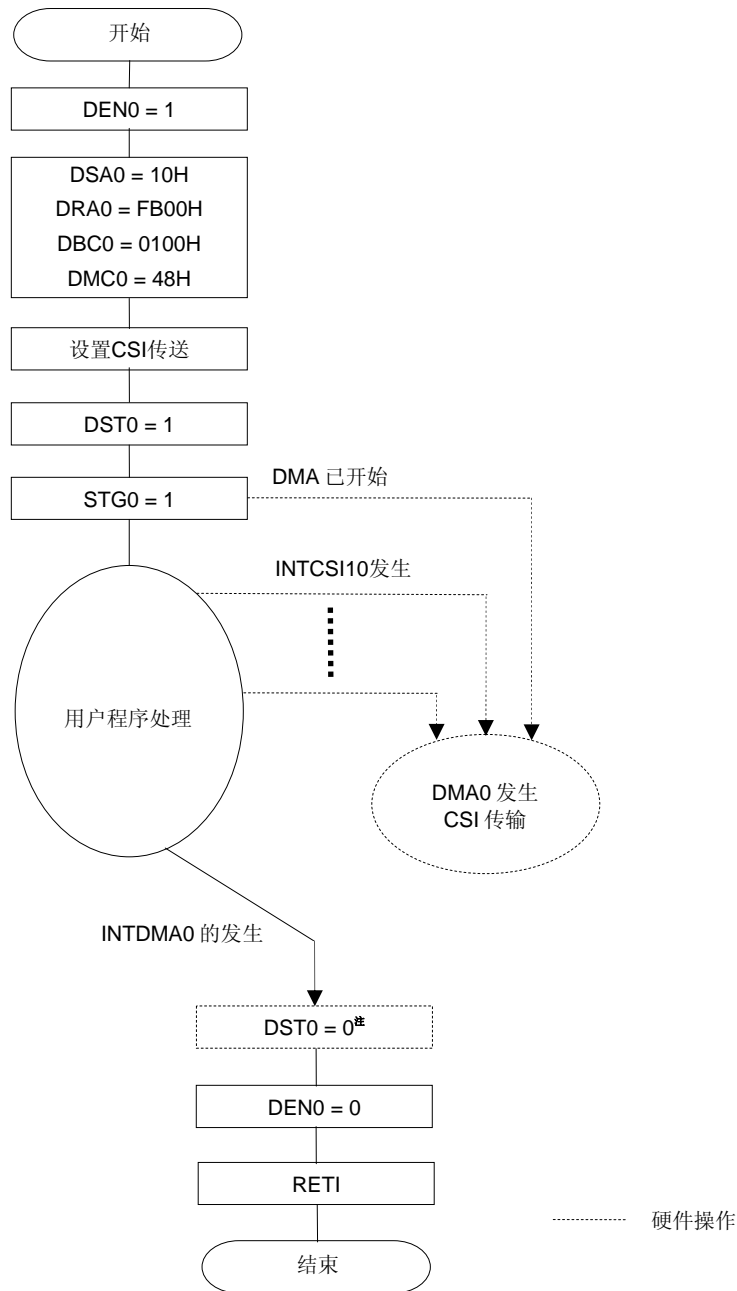
14.5 DMA控制器设置的实例

14.5.1 CSI连续传输

用于 CSI 连续传输的设置实例的流程图如下所示。

- CSI10 的连续传输
- DMA 通道 0 用于 DMA 传输。
- DMA 启动源：INTCSI10（软件触发（STG0）只用于第一个启动源）
- CSI10 的中断由 IFC03~IFC00 指定（DMC0 寄存器的第 3 位~第 0 位）= 1000B。
- 传输 RAM 的 FFB00H~FFBFFH（256 字节）到 CSI 的传输缓冲器（SIO10）的 FFF10H。

图 14-7. CSI 连续传输的设置实例



注 当 DMA 传输完成时，DST0 标志自动清除为 0。

只有当 DST0 = 0 时，DEN0 标志允许。为了中止 DMA 传输，而不等待 DMA0 (INTDMA0) 中断产生，把 DST0 设置为 0，然后把 DEN0 也设置为 0（详细信息，参考 14.5.5 软件强行中断）。

连续传输的第一次触发并不由 CSI 中断启动。要由软件触发启动。

第二次的 CSI 传输和以后的传输自动执行。

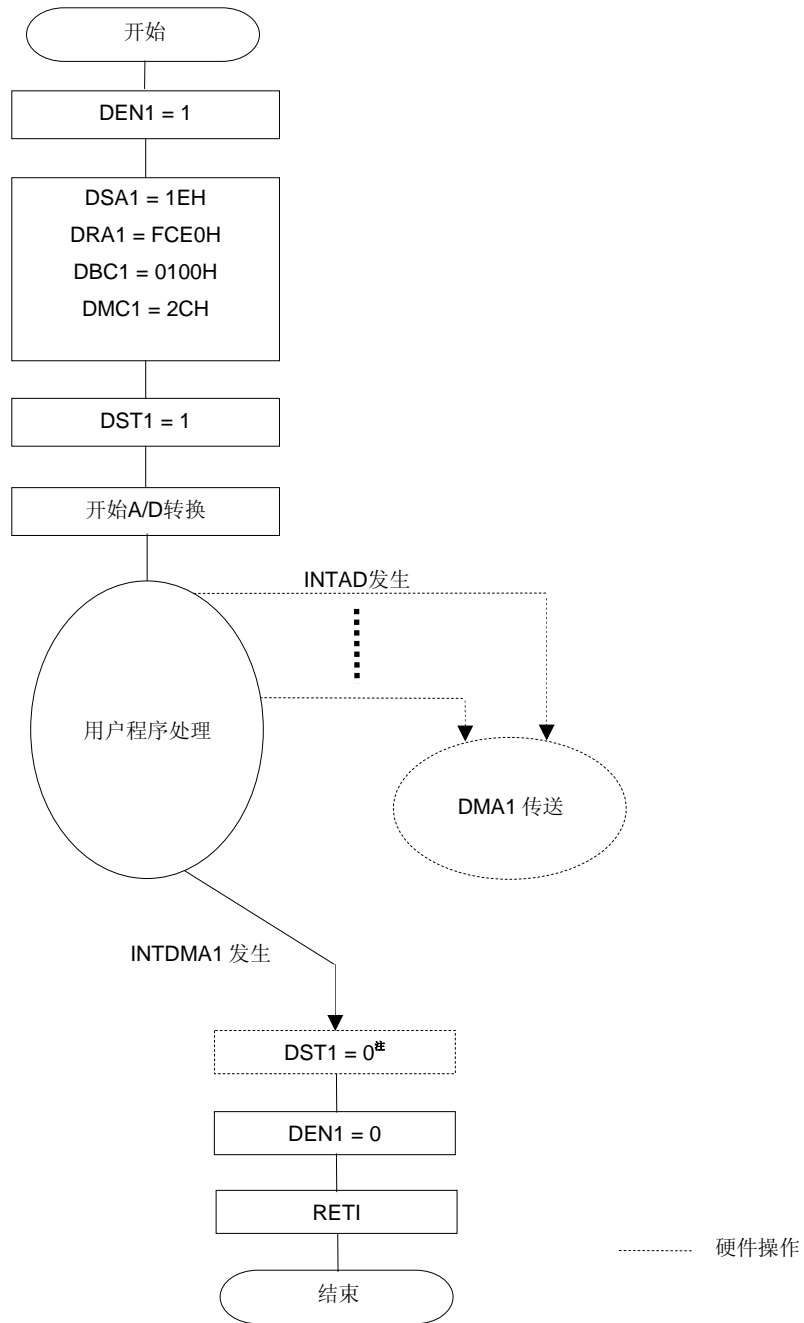
一旦最后的数据写入传输缓冲器，DMA 中断 (INTDMA0) 马上产生。此时，CSI 的最后数据正在传输。为了再次开始 DMA 传输，因此要一直等到 CSI 传输完成。

14.5.2 A/D转换结果的连续捕捉

表示 A/D 转换结果连续捕捉设置实例的流程图如下所示。

- A/D 转换结果的连续捕捉。
- DMA 通道 1 用于 DMA 传输。
- DMA 启动源：INTAD
- A/D 中断由 IFC13~IFC10（DMC1 寄存器的第 3 位~第 0 位）= 1100B 来指定。
- 把 10 位 A/D 转换结果寄存器的 FFF1EH 和 FFF1FH（2 个字节）传输到 RAM 的 FFCE0H~FFEDFH 的 512 个字节中。

图 14-8. A/D 转换结果连续捕捉设置的实例



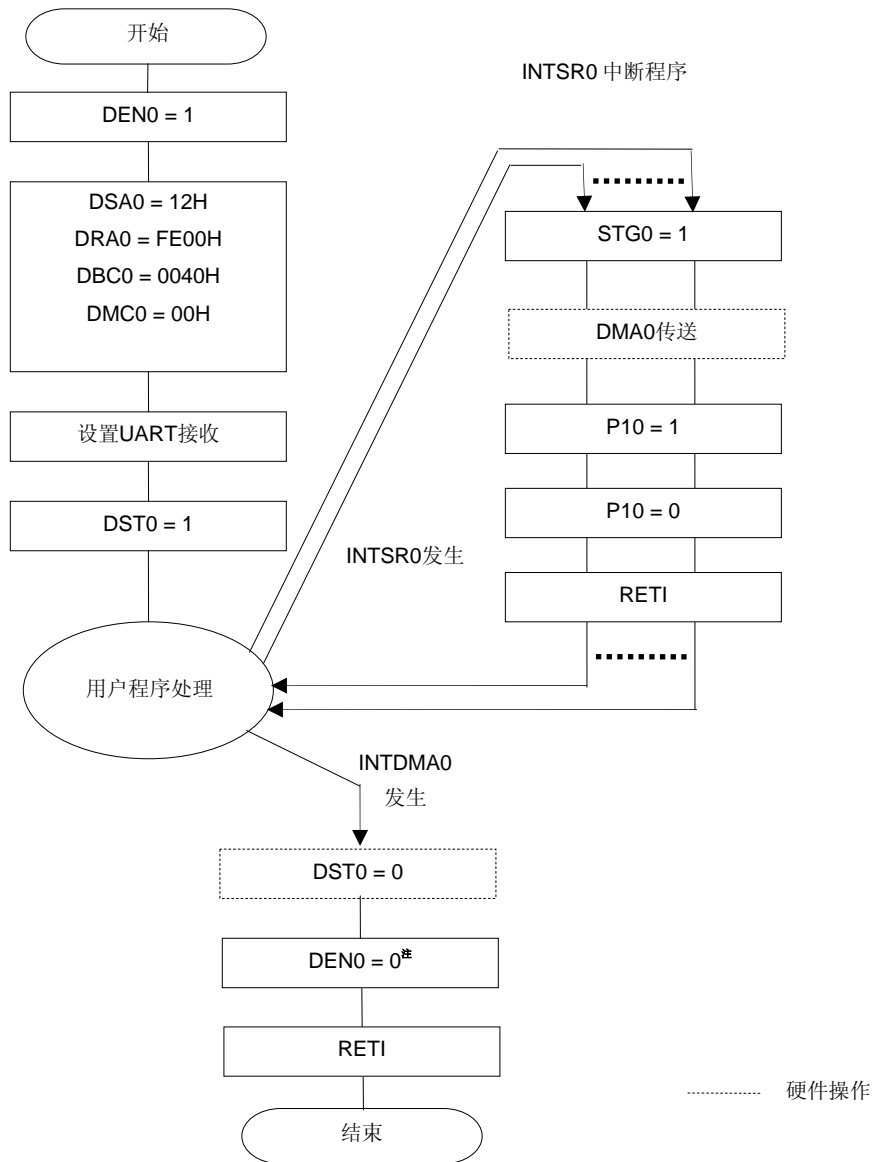
注 当 DMA 传输完成时，DST1 标志自动清除为 0。
 只有当 DST1 = 0 时，DEN1 标志允许。为了中止 DMA 传输，而不等待 DMA1 (INTDMA1) 中断产生，把 DST1 设置为 0，然后把 DEN1 也设置为 0（详细信息，参考 14.5.5 软件强行中止）。

14.5.3 UART连续接收+ACK 传输

表示 UART 连续接收+ACK 传输设置实例的流程图如下所示。

- 从 UART0 连续接收数据，并在接收完成之后输出 ACK 给 P10。
- DMA 通道 0 用于 DMA 传输。
- DMA 启动源：软件触发（禁止中断产生时的 DMA 传输。）
- 传输 UART 接收数据寄存器 0 (RXD0) 的 FFF12H 到 RAM 的 FFE00H~FFE3FH 的 64 字节。

图 14-9. UART 连续接收+ACK 传输设置的实例



注 当 DMA 传输完成时，DST0 标志自动清除为 0。
只有当 DST0 = 0 时，DEN0 标志允许。为了中止 DMA 传输，而不等待 DMA0 (INTDMA0) 中断产生，把 DST0 设置为 0，然后把 DEN0 也设置为 0（详细信息，参考 14.5.5 软件强行中断）。

备注 此为一个软件触发用作 DMA 启动源的实例。
如果不传输 ACK，并且只有数据从 UART 中连续接收，那么 UART 接收结束中断 (INTSR0) 可以用于启动 DMA 用于数据接收。

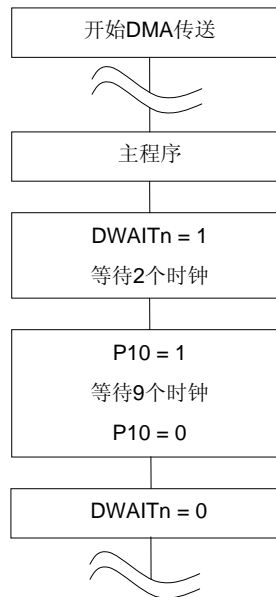
14.5.4 通过DWAITn保持DMA传输挂起

启动 DMA 传输时，指令执行期间传输进行。此时，CPU 的操作停止，并且延迟 2 个时钟。如果这对设定系统的操作造成了问题，DMA 传输可以通过设置 DWAITn 为 1 来保持挂起。

例如，为了从 P10 引脚输出脉宽为 10 个操作频率的时钟的脉冲，如果 DMA 传输中途开始，那么脉冲宽度将增加到 12。这种情况下，DMA 传输可以通过设置 DWAITn 为 1 来保持挂起。

设置 DWAITn 为 1 后，保持 DMA 传输挂起需要 2 个时钟。

图 14-10. 通过 DWAITn 保持 DMA 传输挂起设置的实例



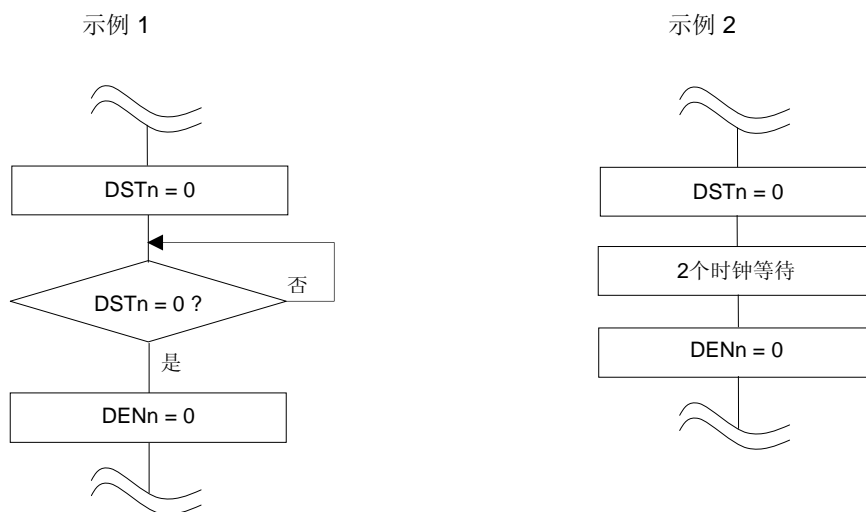
- 备注
1. n: DMA 通道编号 (n = 0, 1)
 2. 1 个时钟: $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

14.5.5 软件强行中止

DSTn 由软件设置为 0 之后，DMA 传输需要 2 个时钟才能真正停止并且 DSTn 位被设置为 0。为了由软件强行中止 DMA 传输，而不等待 DMA 中断 (INTDMA_n) 的产生，需要执行下列的操作。

- 由软件设置 DSTn 为 0（用 8 位操作指令写入 DRCn = 80H），通过查询确认 DSTn 已清除为 0，然后设置 DENn 为 0（使用 DRCn = 00H 以 8 位操作指令写入）。
- 由软件设置 DSTn 为 0（用 8 位操作指令写入 DRCn = 80H），然后在 2 个或者更多个时钟之后设置 DENn 为 0（使用 DRCn = 00H 以 8 位操作指令写入）。

图 14-11. DMA 传输的强行中止



- 备注
1. n: DMA 通道编号 (n = 0, 1)
 2. 1 个时钟: $1/f_{CLK}$ (f_{CLK} : CPU 时钟)

(3) 待机模式下的操作

DMA 控制器的待机模式按照下列所示进行。

表 14-3. 待机模式下 DMA 的操作

状态	DMA操作
HALT模式	正常操作
STOP模式	停止操作。 如果DMA传输和STOP指令执行冲突，那么DMA传输可能被破坏。因此，执行STOP指令之前，要停止DMA。

(4) DMA 挂起指令

下列指令之后，即使 DMA 请求产生，DMA 传输也要立即保持挂起。

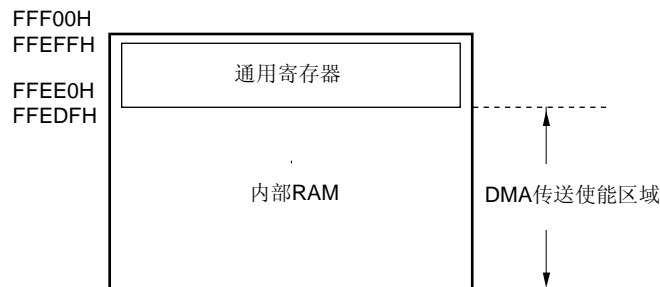
- CALL !addr16
- CALL &!addr16
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- 分别用于寄存器 IF0L、IF0H、IF1L、IF1H、IF2L、IF2H、MK0L、MK0H、MK1L、MK1H、MK2L、MK2H、PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H 和 PSW 的位操作指令，以及以 ES 寄存器为操作数的 8 位操作指令

(5) 如果指定了通用寄存器区域或者内部 RAM 之外的区域时的操作

在 DMA 传输期间，由 DRA0n 指示的地址增加。如果地址增加到通用寄存器区域，或者超过了内部 RAM 区域，下列操作将被执行。

- 在从 SFR 到 RAM 传输的模式下
地址数据丢失。
- 在从 RAM 到 SFR 传输的模式下
不确定数据传输到 SFR。

任何一种情况下，都会发生误操作或者对系统造成损坏。因此，确保地址在内部 RAM 区域而不是在通用寄存器区域。



第 15 章 中断函数

15.1 中断函数的类型

可分为以下两种类型。

(1) 可屏蔽中断

这类中断可进行屏蔽控制。通过设置优先级指定标志寄存器（PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H）将可屏蔽中断分为四个优先级组。

支持中断嵌套，即在低优先级的中断服务子程序中可以响应高优先级的中断请求。如果同时产生两个或两个以上具有相同优先级的中断请求，则根据向量中断服务的优先级进行处理。优先级顺序，参见表 15-1。

产生一个待机释放信号，并释放 STOP 和 HALT 模式。

可屏蔽中断包括外部中断请求和内部中断请求。

外部：8，内部：33

(2) 软件中断

这是通过执行 BRK 指令产生的一类向量中断。即使禁止中断时也可以响应这类中断。软件中断不受中断优先级控制。

15.2 中断源和配置

μPD79F9211 系列中，共有 43 种中断源，包括可屏蔽的中断和软件中断。另外，还具有 5 种复位源（参见表 15-1）。

表 15-1. 中断源列表(1/2)

中断类型	默认优先级 ^{註 1}	中断源		内部/ 外部	向量表 地址	基本配置类型 ^{註 2}
		名称	触发器			
可屏蔽的	0	INTWDTI	看门狗定时器间隔定时 ^{註 3} (溢出时间的 75%)	内部	0004H	(A)
	1	INTLVI	低电压检测 ^{註 4}		0006H	
	2	INTP0	引脚输入沿检测	外部	0008H	(B)
	3	INTP1			000AH	
	4	INTP2			000CH	
	5	INTP3/INTTMOFF0	引脚输入沿检测/定时器高阻控制中断 0		000EH	
	6	INTP4	引脚输入沿检测		0010H	
	7	INTP5			0012H	
	8	INTTMAD	A/D 转换定时器触发	内部	0014H	(A)
	9	INTCMP0	CMP0 检测		0016H	
	10	INTCMP1	CMP1 检测		0018H	
	11	INTDMA0	DMA0 发送结束		001AH	
	12	INTDMA1	DMA1 发送结束		001CH	
	14	INTSR0	UART0 接收结束/ CSI01 通信结束		0020H	
	15	INTSRE0	UART0 发生通信错误		0022H	
	16	INTST1 /INTCSI10 /INTIIC10	UART1 发送结束/ CSI10 通信结束/ IIC10 通信结束		0024H	
	17	INTSR1	UART1 接收结束		0026H	
	18	INTSRE1	UART1 发生通信错误		0028H	
	20	INTTM00	定时器通道 0 计数或捕捉结束		002CH	
	21	INTTM01	定时器通道 1 计数或捕捉结束		002EH	
	22	INTTM02	定时器通道 2 计数或捕捉结束		0030H	
	23	INTTM03	定时器通道 3 计数或捕捉结束		0032H	
	24	INTAD	A/D 转换结束		0034H	

- 注
1. 当两个或两个以上可屏蔽中断同时产生时，采用默认的优先级来决定向量中断的处理顺序。优先级 0 的优先级别最高，而优先级 41 的优先级别最低。
 2. 基本配置类型 (A)~(C) 与图 15-1 中 (A)~(C) 相对应。
 3. 当选项字节 (000C0H) 第 7 位 (WDTINT) 被置 1 时。
 4. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 被清零时。

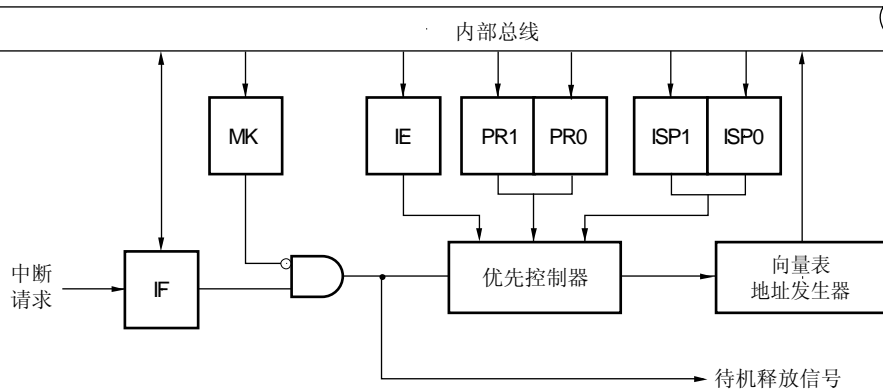
表 15-1. 中断源列表(2/2)

中断类型	默认优先级 ^{注1}	中断源		内部/ 外部	向量表 地址	基本配置类型 ^{注2}
		名称	触发器			
可屏蔽的	25	INTRTC	实时计数器的固定周期信号/报警匹配检测	内部	0036H	(A)
	26	INTRTCI	实时计数器的间隔信号检测		0038H	
	27	INTTMM0	定时器阵列单元山型中断信号检测 0		003CH	
	28	INTTMV0	定时器阵列单元谷型中断信号检测 0		003EH	
	29	INTMD	除法操作结束		0040H	
	30	INTTM04	定时器通道 4 计数或捕捉结束		0042H	
	31	INTTM05	定时器通道 5 计数或捕捉结束		0044H	
	32	INTTM06	定时器通道 6 计数或捕捉结束		0046H	
	33	INTTM07	定时器通道 7 计数或捕捉结束		0048H	
	34	INTP6	引脚输入沿检测	外部	004AH	(B)
	35	INTP7/INTTMOFF1	引脚输入沿检测/定时器高阻控制中断 0		004CH	
	36	INTTMM1	定时器阵列单元山型中断信号检测 1	内部	004EH	(A)
	37	INTTMV1	定时器阵列单元谷型中断信号检测 1		0050H	
	38	INTTM08	定时器通道 8 计数或捕捉结束		0052H	
	39	INTTM09	定时器通道 9 计数或捕捉结束		0054H	
	40	INTTM10	定时器通道 10 计数或捕捉结束		0056H	
	41	INTTM11	定时器通道 11 计数或捕捉结束		0058H	
软件	—	BRK	BRK 指令执行	—	007EH	(C)
复位	—	RESET	RESET 引脚输入	—	0000H	—
		POC	上电清零			
		LVI	低电压检测 ^{注3}			
		WDT	看门狗定时器溢出			
		TRAP	非法指令执行 ^{注4}			

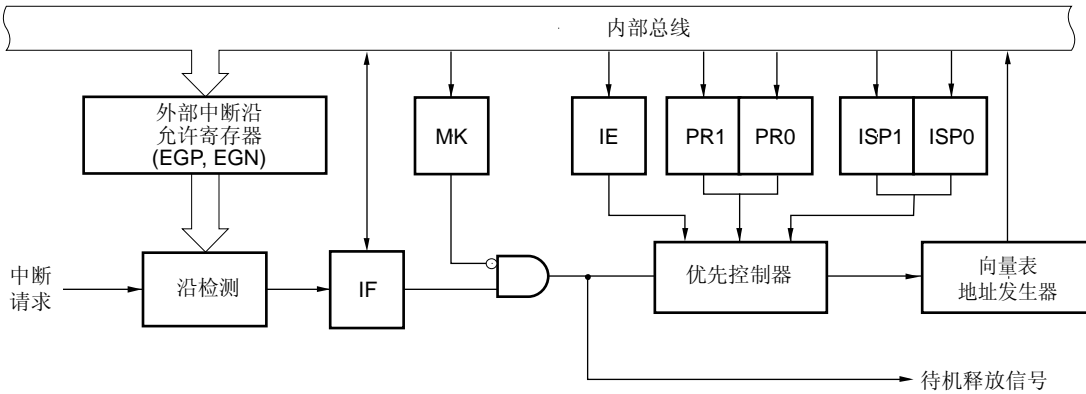
- 注
1. 当两个或两个以上可屏蔽中断同时产生时，采用默认的优先级来决定向量中断的处理顺序。优先级 0 的优先级别最高，而优先级 41 的优先级别最低。
 2. 基本配置类型 (A)~(C) 与图 15-1 中 (A)~(C) 相对应。
 3. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 被设置为 1 时。
 4. 当执行 FFH 中的指令代码时。
由非法指令执行引起的复位，不能通过在线仿真器或片上调试仿真器终止。

图 15-1. 中断函数的基本配置

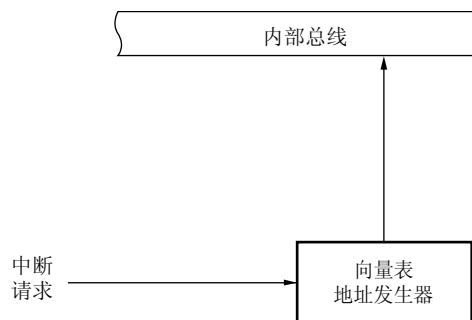
(A) 内部可屏蔽中断



(B) 外部可屏蔽中断



(C) 软件中断



- IF: 中断请求标志
- IE: 中断允许标志
- ISP0: 服务优先级标志 0
- ISP1: 服务优先级标志 1
- MK: 中断屏蔽标志
- PR0: 优先级指定标志 0
- PR1: 优先级指定标志 1

15.3 控制中断函数的寄存器

以下 6 种寄存器用于控制中断函数。

- 中断请求标志寄存器（IF0L、IF0H、IF1L、IF1H、IF2L、IF2H）
- 中断屏蔽标志寄存器（MK0L、MK0H、MK1L、MK1H、MK2L、MK2H）
- 优先级指定标志寄存器（PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H）
- 外部中断上升沿使能寄存器（EGP0）
- 外部中断下降沿使能寄存器（EGN0）
- 程序状态字（PSW）

表 15-2 列出了与中断请求源相对应的中断请求标志、中断屏蔽标志和优先级指定标志。

表 15-2. 与中断请求源相对应的标志 (1/2)

中断源	中断请求标志		中断屏蔽标志		优先级指定标志	
		寄存器		寄存器		寄存器
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1	
INTP0	PIF0		PMK0		PPR00, PPR10	
INTP1	PIF1		PMK1		PPR01, PPR11	
INTP2	PIF2		PMK2		PPR02, PPR12	
INTP3 ^{注1}	PIF3 ^{注1}		PMK3 ^{注1}		PPR03、PPR13 ^{注1}	
INTTMOFF0 ^{注1}	TMOFFIF0 ^{注1}		TMOFFMK0 ^{注1}		TMOFFPR00、 TMOFFPR10 ^{注1}	
INTP4	PIF4		PMK4		PPR04, PPR14	
INTP5	PIF5		PMK5		PPR05, PPR15	
INTTMAD	TMADIF	IF0H	TMADMK	MK0H	TMADPR0、 TMADPR1	PR00H, PR10H
INTCMP0	CMPIF0		CMPMK0		CMPPR00, CMPPR10	
INTCMP1	CMPIF1		CMPMK1		CMPPR01, CMPPR11	
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10	
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11	
INTST0 ^{注2}	STIF0 ^{注2}		STMK0 ^{注2}		STPR00、STPR10 ^{注2}	
INTCSI00 ^{注2}	CSIIF00 ^{注2}		CSIMK00 ^{注2}		CSIPR000、CSIPR100 ^{注2}	
INTSR0 ^{注3}	SRIF0 ^{注3}		SRMK0 ^{注3}		SRPR00、SRPR10 ^{注3}	
INTCSI01 ^{注3}	CSIIF01 ^{注3}		CSIMK01 ^{注3}		CSIPR001、CSIPR101 ^{注3}	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	

- 注
1. 不要同时使用 INTP3 和 INTTMOFF0，因为它们分享中断请求源的标志。如果产生 INTP3 和 INTTMOFF0 中断源中的一个产生中断，IF0L 的第 5 位设置为 1。MK0L，PR00L，和 PR10L 的第 5 位对应这 2 个中断源。
 2. 不要同时使用 UART0 和 CSI00，因为它们共享中断请求源标志。如果中断源 INST0 和 INTCSI00 中的一个产生中断，则 IF0H 的第 5 位设置为 1。MK0H、PR00H、和 PR10H 的第 5 位支持这 2 个中断源。
 3. 不要同时使用 UART0 和 CSI01，因为它们共享中断请求源标志。如果中断源 INSR0 和 INTCSI01 中的一个产生中断，则 IF0H 的第 6 位设置为 1。MK0H、PR00H、和 PR10H 的第 6 位支持这 2 个中断源。

表 15-2. 与中断请求源相对应的标志 (2/2)

中断源	中断请求标志		中断屏蔽标志		优先级指定标志	
		寄存器		寄存器		寄存器
INTST1 ^{注1}	STIF1 ^{注1}	IF1L	STMK1 ^{注1}	MK1L	STPR01、STPR11 ^{注1}	PR01L, PR11L
INTCSI10 ^{注1}	CSIIF10 ^{注1}		CSIMK10 ^{注1}		CSIPR010、CSIPR110 ^{注1}	
INTIIC10 ^{注1}	IICIF10 ^{注1}		IICMK10 ^{注1}		IICPR010、IICPR110 ^{注1}	
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11	
INTSRE1	SREIF1		SREMK1		SREPR01, SREPR11	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102	
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1	
INTRTCI	RTCIF		RTCIMK		RTCIPR0, RTCIPR1	
INTTMM0	TMMIF0		TMMMK0		TMMPR00、TMMPR10	
INTTMV0	TMVIF0		TMVMK0		TMVPR00、TMVPR10	
INTMD	MDIF		MDMK		MDPR0, MDPR1	
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106	
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
INTP6	PIF6		PMK6		PPR06, PPR16	
INTP7 ^{注2}	PIF7 ^{注2}		PMK7 ^{注2}		PPR07、PPR17 ^{注2}	
INTTMOFF1 ^{注2}	TMOFFIF1 ^{注2}		TMOFFMK1 ^{注2}		TMOFFPR01、 TMOFFPR11 ^{注2}	
INTTMM1	TMMIF1		TMMMK1		TMMPR01、TMMPR11	
INTTMV1	TMVIF1		TMVMK1		TMVPR01、TMVPR11	
INTTM08	TMIF08		TMMK08		TMPR008、TMPR108	
INTTM09	TMIF09	IF2H	TMMK09	MK2H	TMPR009、TMPR109	PR02H 、 PR12H
INTTM10	TMIF10		TMMK10		TMPR010、TMPR110	
INTTM11	TMIF11		TMMK11		TMPR011、TMPR111	

- 注**
1. 不要同时使用 UART1、CSI10、和 IIC10，因为它们共享中断请求源标志。如果中断源 INTST1、INTCSI10、和 INTIIC10 中的 1 个产生，IF1L 的第 0 位设置为 1。MK1L、PR01L、和 PR11L 的第 0 位支持这 3 个中断源。
 2. 不要同时使用 INTP7 和 INTTMOFF1，因为它们共享中断请求源的标志。如果中断源 INTP7 和 INTTMOFF1 中的一个产生中断，IF2L 的第 4 位设置为 1。MK2L、PR02L、和 PR12L 的第 4 位支持这 2 个中断源。

(1) 中断请求标志寄存器 (IF0L、IF0H、IF1L、IF1H、IF2L、IF2H)

当产生相关的中断请求或执行指令时，这些中断请求标志被置 1。当执行的指令是响应中断请求或复位输入时，这些标志被清零。

当响应中断时，中断请求标志自动清零，然后进入中断服务程序。

可由 1 位或 8 位存储器操作指令设置 IF0L、IF0H、IF1L、IF1H、IF2L 和 IF2H。当 IF0L 与 IF0H、IF1L 与 IF1H 和 IF2L 与 IF2H 组合起来形成 16 位寄存器 IF0、IF1 和 IF2 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的产生可将这些寄存器清零(00H)。

备注 如果执行写入此寄存器数据的指令，指令执行的时钟个数增加 2。

图 15-2. 中断请求标志寄存器的格式 (IF0L、IF0H、IF1L、IF1H、IF2L、IF2H) (1/2)

地址: FFFE0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3 TMOFFIF0	PIF2	PIF1	PIF0	LVIIIF	WDTIIF

地址: FFFE1H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0	SRIF0 CSIIIF01	STIF0 CSIIIF00	DMAIF1	DMAIF0	CMPIF1	CMPIF0	TMADIF

地址: FFFE2H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	0	SREIF1	SRIF1	STIF1 CSIIIF10 IICIF10

地址: FFFE3H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
IF1H	TMIF04	MDIF	TMVIF0	TMMIF0	0	RTCIIF	RTCIF	ADIF

地址: FFFD0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	TMIF08	TMVIF1	TMMIF1	PIF7 TMOFFIF1	PIF6	TMIF07	TMIF06	TMIF05

图 15-2. 中断请求标志寄存器的格式（IF0L、IF0H、IF1L、IF1H、IF2L、IF2H）（2/2）

地址：FFFD1H 复位后：00H R/W

符号	7	6	5	4	3	<2>	<1>	<0>
IF2H	0	0	0	0	0	TMIF11	TMIF10	TMIF09

XXIFX	中断请求标志
0	没有产生中断请求信号
1	产生中断请求，中断请求状态

注意事项

1. 确保对 IF1H 的第 3 位、IF2H 的第 3 位～第 7 位清零。

2. 当退出待机模式后并要使用定时器、串行接口或 A/D 转换器时，先将中断请求标志清零后再对这些部件进行操作。噪音可能会设置中断请求标志。

3. 当使用中断请求标志寄存器中的标志时，使用一位存储器操作指令 (CLR1)。在 C 语言中描述时，应使用诸如“IF0L.0 = 0;”或“_asm(“clr1 IF0L, 0”);”的位操作指令，因为编译的汇编程序必须为 1 位存储器操作指令(CLR1)。

如果用 C 语言描述的程序使用 8 位存储器操作指令如“IF0L &= 0xfe;”并编译的话，将变成 3 个指令的汇编程序。

mov a, IF0L

and a, #0FEH

mov IF0L, a

在这种情况下，即使相同中断请求标志寄存器(IF0L)中的另一位的请求标志在“mov a, IF0L”和“mov IF0L, a”之间时序上设为 1 时，该标志在“mov IF0L, a”时清零。因此在 C 语言中使用 8 位存储器操作指令时必须小心。

(2) 中断屏蔽标志寄存器（MK0L、MK0H、MK1L、MK1H、MK2L、MK2H）

这些中断屏蔽标志用于允许/禁止相关的可屏蔽中断服务。

可由 1 位或 8 位存储器操作指令设置 MK0L、MK0H、MK1L、MK1H、MK2L、和 MK2H。当 MK0L 与 MK0H，MK1L 与 MK1H，MK2L 与 MK2H 组合起来形成 16 位寄存器 MK0、MK1 和 MK2 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的产生将这些寄存器设置为 FFH。

备注 如果执行写入此寄存器数据的指令，指令执行的时钟个数增加 2。

图 15-3. 中断屏蔽标志寄存器的格式（MK0L、MK0H、MK1L、MK1H、MK2L、MK2H）

地址：FFFE4H 复位后：FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3 TMOFFMK0	PMK2	PMK1	PMK0	LVIMK	WDTIMK

地址：FFE5H 复位后：FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0	SRMK0 CSIMK01	STMK0 CSIMK00	DMAMK1	DMAMK0	CMPMK1	CMPMK0	TMADMK

地址：FFFE6H 复位后：FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	0	SREMK1	SRMK1	STMK1 CSIMK10 IICMK10

地址：FFFE7H 复位后：FFH R/W

符号	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
MK1H	TMMK04	MDMK	TMVMK0	TMMMK0	1	RTCIMK	RTCMK	ADMK

地址：FFFD4H 复位后：FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	TMMK08	TMVMK1	TMMMK1	PMK7 TMOFFMK1	PMK6	TMMK07	TMMK06	TMMK05

地址：FFFD5H 复位后：FFH R/W

符号	7	6	5	4	3	<2>	<1>	<0>
MK2H	1	1	1	1	1	TMMK11	TMMK10	TMMK09

XXMKX	中断服务控制
0	允许中断服务
1	禁止中断服务

注意事项 确保对 MK1H 的第 3 位、MK2H 的第 3 位～第 7 位设置为 1。

(3) 优先级指定标志寄存器 (PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H)

这些优先级指定标志寄存器用于设置相关的可屏蔽中断优先级次序。

优先级可通过组合使用 PR0xy 和 PR1xy 寄存器设置 (xy = 0L、0H、1L、1H、2L 或 2H)。

可由 1 位或 8 位存储器操作指令设置 PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、和 PR12H。当 PR00L 与 PR00H、PR01L 与 PR01H、PR02L 与 PR02H、PR10L 与 PR10H、PR11L 与 PR11H、和 PR12L 与 PR12H 组合起来形成 16 位寄存器 PR00、PR01、PR02、PR10、PR11 和 PR12 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的产生将这些寄存器设置为 FFH。

备注 如果执行写入此寄存器数据的指令，指令执行的时钟个数增加 2。

图 15-4. 优先级指定标志寄存器的格式

(PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H)
(1/3)

地址: FFFE8H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03 TMOFFPR00	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

地址: FFECH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13 TMOFFPR10	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

地址: FFFE9H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00	SRPR00 CSIPR001	STPR00 CSIPR000	DMAPR01	DMAPR00	CMPPR01	CMPPR00	TMADPR0

地址: FFFEDH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10	SRPR10 CSIPR101	STPR10 CSIPR100	DMAPR11	DMAPR10	CMPPR11	CMPPR10	TMADPR1

图 15-4. 优先级指定标志寄存器的格式

(PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H)
(2/3)

地址: FFFEAH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	0	SREPR01	SRPR01	STPR01 CSIPR010 IICPR010

地址: FFFEEH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	0	SREPR11	SRPR11	STPR11 CSIPR110 IICPR110

地址: FFFEBH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR01H	TMPR004	MDPR0	TMVPR00	TMMPR00	1	RTCIPR0	RTCPR0	ADPR0

地址: FFFEFH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR11H	TMPR104	MDPR1	TMVPR10	TMMPR10	1	RTCIPR1	RTCPR1	ADPR1

地址: FFFD8H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	TMPR008	TMVPR01	TMMPR01	PPR07 TMOFFPR0 1	PPR06	TMPR007	TMPR006	TMPR005

地址: FFFDCH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	TMPR108	TMVPR11	TMMPR11	PPR17 TMOFFPR1 1	PPR16	TMPR107	TMPR106	TMPR105

地址: FFFD9H 复位后: FFH R/W

符号	7	6	5	4	3	<2>	<1>	<0>
PR02H	1	1	1	1	1	TMPR011	TMPR010	TMPR009

地址: FFFDDH 复位后: FFH R/W

符号	7	6	5	4	3	<2>	<1>	<0>
PR12H	1	1	1	1	1	TMPR111	TMPR110	TMPR109

图 15-4. 优先级指定标志寄存器的格式
(PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L、PR12H)
(3/3)

XXPR1X	XXPR0X	优先级选择
0	0	指定优先级 0 (高优先级)
0	1	指定优先级 1
1	0	指定优先级 2
1	1	指定优先级 3 (低优先级)

注意事项 确保要把 PR01H 和 PR11H 的第 3 位、PR02H 和 PR12H 的第 3 位～第 7 位设置为 1。

- (4) 外部中断上升沿使能寄存器 (EGP0)，外部中断下降沿使能寄存器(EGN0)，
这两个寄存器用于指定 INTP0~INTP7 的有效沿。
EGP0 和 EGN0 可以通过 1 位或 8 位存储器操作指令设置。
复位信号的产生可将这些寄存器清零(00H)。

图 15-5. 外部中断上升沿使能寄存器 (EGP0)
和外部中断下降沿使能寄存器 (EGN0)的格式

地址: FFF38H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

地址: FFF39H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn 引脚有效沿的选择 (n = 0 ~ 7)
0	0	禁止脉冲沿检测
0	1	下降沿
1	0	上升沿
1	1	兼有上升沿和下降沿

表 15-3 显示了与 EGPn 和 EGNn 相关的端口。

表 15-3. EGPn 和 EGNn 相关的端口

检测允许寄存器		脉冲沿检测端口	中断请求信号
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P31	INTP1
EGP2	EGN2	P32	INTP2
EGP3	EGN3	P80	INTP3
EGP4	EGN4	P70	INTP4
EGP5	EGN5	P71	INTP5
EGP6	EGN6	P72	INTP6
EGP7	EGN7	P82	INTP7

注意事项 在从外部中断函数切换到端口功能时，可能会检测到脉冲沿，因此通过将 EGPn 和 EGNn 清零，可选择端口模式。

备注 n = 0~7

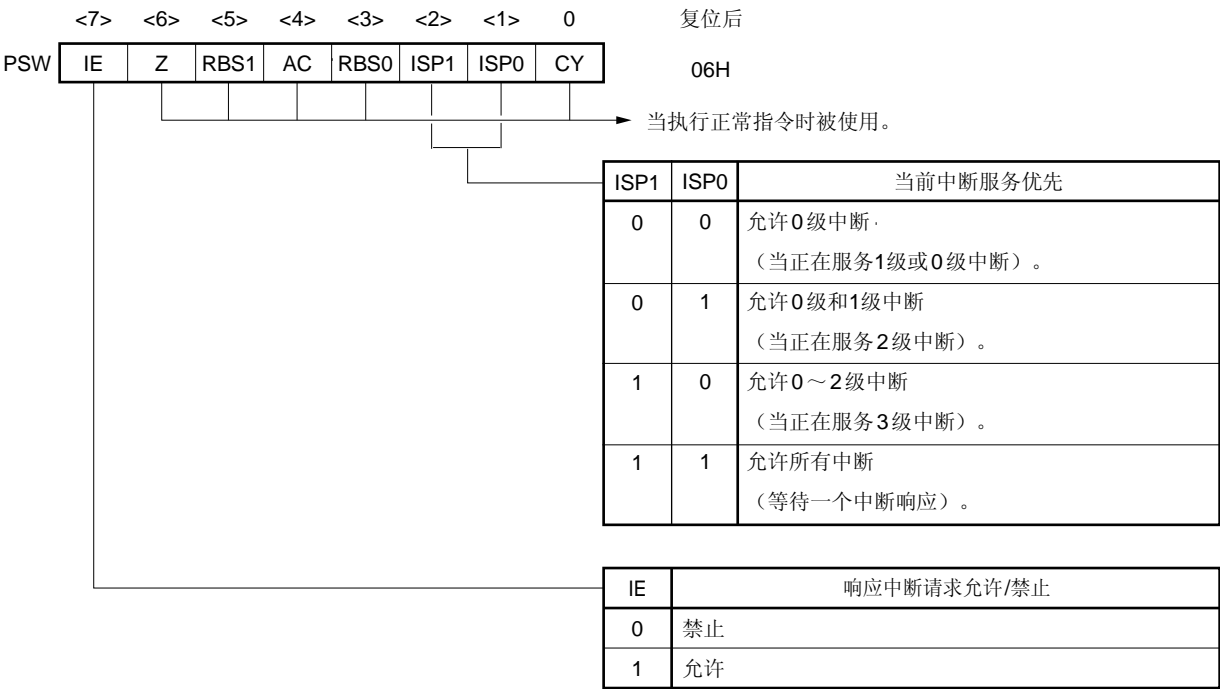
(5) 程序状态字 (PSW)

PSW 是用于保存指令执行结果和中断请求的当前状态的寄存器。在 PSW 中包含 IE 标志（设置允许/禁止可屏蔽中断）和 ISP0 与 ISP1 标志（控制中断嵌套）。

除了 8 位读/写操作指令外，还可使用位操作指令和专用指令 (EI 和 DI) 对该寄存器进行操作。在响应向量中断请求时，如果执行 BRK 指令，则将 PSW 的内容自动保存到堆栈中，并且将 IE 标志复位为 0。如果响应可屏蔽中断请求，则将被响应中断的优先级指定标志的内容传送到 ISP0 和 ISP1 标志中。执行 PUSH PSW 指令将 PSW 的内容保存到堆栈中。而执行 RETI、RETB 和 POP PSW 指令可将这些内容从堆栈中恢复。

复位信号的产生将程序状态字的内容设置为 06H。

图 15-6. 程序状态字的格式



15.4 中断服务操作

15.4.1 可屏蔽的中断响应

当中断请求标志=1 且与该中断请求相关的屏蔽标志（MK）被清零时，可以响应这个可屏蔽中断请求。如果处于中断允许状态（IE 标志=1），则可以响应向量中断请求。但在一个较高优先级中断请求服务期间，不响应低优先级的中断请求。

从一个可屏蔽中断请求产生，到中断服务执行所经历的时间，如表 15-4 所示。

中断请求响应时序可参见图 15-8 和 15-9。

表 15-4. 从可屏蔽中断产生到执行中断服务所需要的时间

	最短时间	最长时间 ^注
时间	9 个时钟	14 个时钟

注 如果是在 RET 指令执行之前产生一个中断请求，则等待时间会更长。

备注 1 个时钟：1/fCLK (fCLK：CPU 时钟)

如果同时产生两个或两个以上的可屏蔽中断请求，则首先响应优先级指定标志中优先级别高的请求。如果两个或两个以上的中断请求具有相同的优先级别，则首先响应具有最高默认优先级的中断请求。

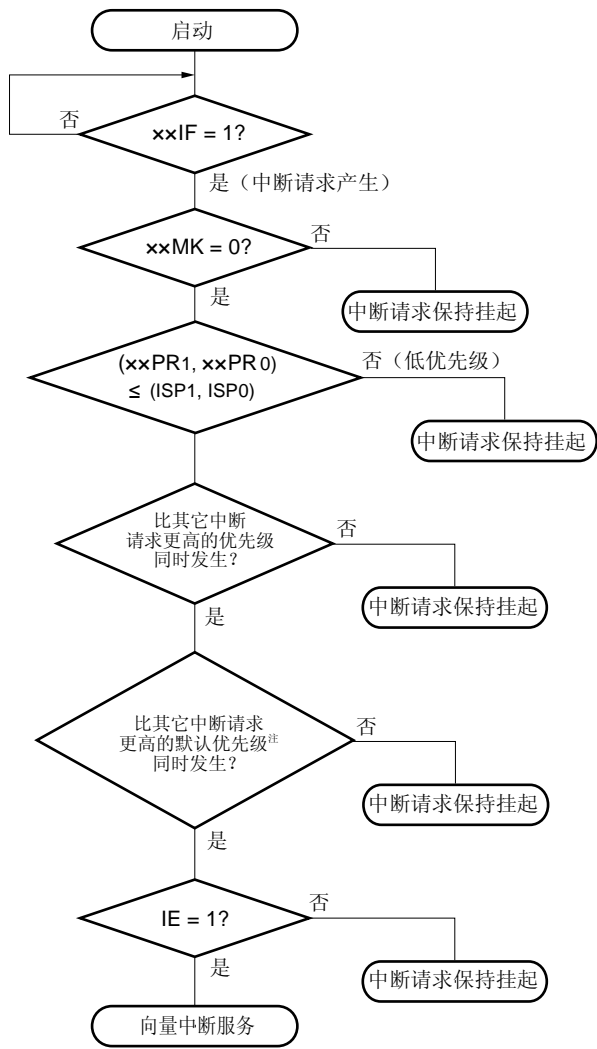
当允许响应中断请求时，处于等待状态的中断请求被响应。

图 15-7 显示了中断请求响应算法。

如果响应了一个可屏蔽中断请求，则将 PSW、PC 的内容依次保存到堆栈中，然后将 IE 标志复位为 0，并将与被响应的中断相关的优先级指定标志的内容传送到 ISP1 和 ISP0 中。将用于每个中断请求的向量表数据传送到 PC 中并分支。

可通过执行 RETI 指令从中断返回。

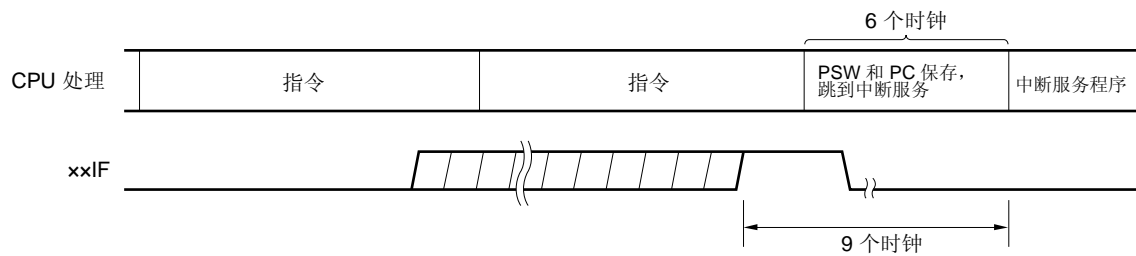
图 15-7. 中断请求响应处理算法



xxIF: 中断请求标志
xxMK: 中断屏蔽标志
xxPR0: 优先级指定标志 0
xxPR1: 优先级指定标志 1
IE: 控制可屏蔽中断请求响应的标志 (1 = 允许、0 = 禁止)
ISP0, ISP1: 指示正在服务的中断优先级别的标志 (参见 图 15-6)

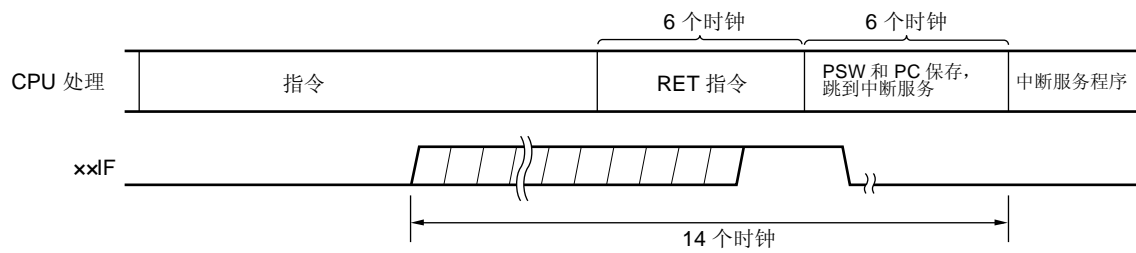
注 关于默认优先级，参考 表 15-1 中断源列表

图 15-8. 中断请求响应时序（最短时间）



备注 1 个时钟：1/fCLK (fCLK：CPU 时钟)

图 15-9. 中断请求响应时序（最长时间）



备注 1 个时钟：1/fCLK (fCLK：CPU 时钟)

15.4.2 软件中断请求响应

执行 **BRK** 指令可响应软件中断。软件中断不能被禁止。

如果响应了一个软件中断请求，则将程序状态字 (PSW) 和程序计数器 (PC) 的内容依次保存到堆栈中，然后将 **IE** 标志复位为 0，并将向量表的内容 (0007EH, 0007FH) 传送到 PC 中并分支。

可通过执行 **RETB** 指令从软件中断返回。

注意事项 不能使用 **RETI** 指令从软件中断返回。

15.4.3 中断嵌套

在执行一个中断服务程序时，又响应了其它中断，这时就产生了中断嵌套。

除非选择允许中断请求响应状态 (**IE = 1**)，否则不会产生中断嵌套。在响应一个中断请求时，禁止响应其它中断请求 (**IE = 0**)。因此，如果要允许中断嵌套，必须在中断服务期间执行 **EI** 指令，将 **IE** 标志置 1，从而允许响应其它中断请求。

此外，即使允许中断，也不一定允许中断嵌套，这是因为受到中断优先级控制的限制。可使用两类优先级控制方式：默认优先级控制和可编程优先级控制。可编程优先级控制用于中断嵌套。

在中断允许状态中，如果产生的中断请求的优先级与正在服务的中断优先级相等或高于它，则响应该中断请求，从而产生中断嵌套。如果产生的中断请求的优先级低于正在服务的中断优先级，则不响应该中断请求。由于禁止中断或中断请求的优先级别较低，这些不被允许的中断请求处于等待状态。若当前的中断服务已结束，则在执行至少一条主程序指令后才可响应处于等待状态的中断请求。

表 15-5 显示了允许中断嵌套的中断请求之间的关系，图 15-10 为中断嵌套示例。

表 15-5. 中断服务期间允许进行中断嵌套的中断请求之间的关系

正在服务的中断		中断嵌套	可屏蔽中断请求								软件中断请求
			优先级 0 (PR = 00)		优先级 1 (PR = 01)		优先级 2 (PR = 10)		优先级 3 (PR = 11)		
			IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
可屏蔽的中断	ISP1 = 0 ISP0 = 0	○	×	×	×	×	×	×	×	○	
	ISP1 = 0 ISP0 = 1	○	×	○	×	×	×	×	×	○	
	ISP1 = 0 ISP0 = 1	○	×	○	×	○	×	×	×	○	
	ISP1 = 1 ISP0 = 1	○	×	○	×	○	×	○	×	○	
软件中断			○	×	○	×	○	×	○	×	○

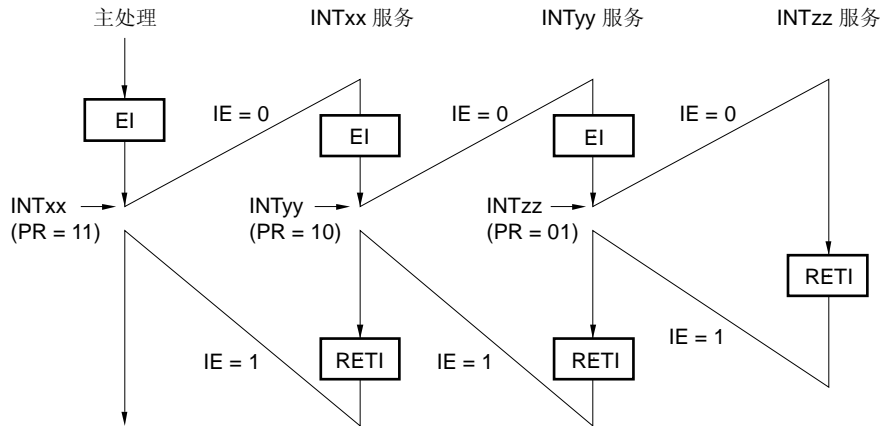
备注

- ：允许中断嵌套
- ×：禁止中断嵌套
- ISP0, ISP1, 和 IE 为 PSW 中的标志位。
 ISP1 = 0, ISP0 = 0：正在服务优先级 1 或 0 的中断。
 ISP1 = 0, ISP0 = 1：正在服务优先级 2 的中断。
 ISP1 = 1, ISP0 = 0：正在服务优先级 3 的中断。
 ISP1 = 1, ISP0 = 1：等待中断响应。
 IE = 0：禁止响应中断请求。
 IE = 1：允许响应中断请求。
- PR 为 PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L 和 PR12H 中的标志位。
 PR = 00：通过 $\times\times PR1\times = 0$, $\times\times PR0\times = 0$, 指定等级 0（较高优先级）
 PR = 01：通过 $\times\times PR1\times = 0$, $\times\times PR0\times = 1$, 指定等级 1
 PR = 10：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）

<R>

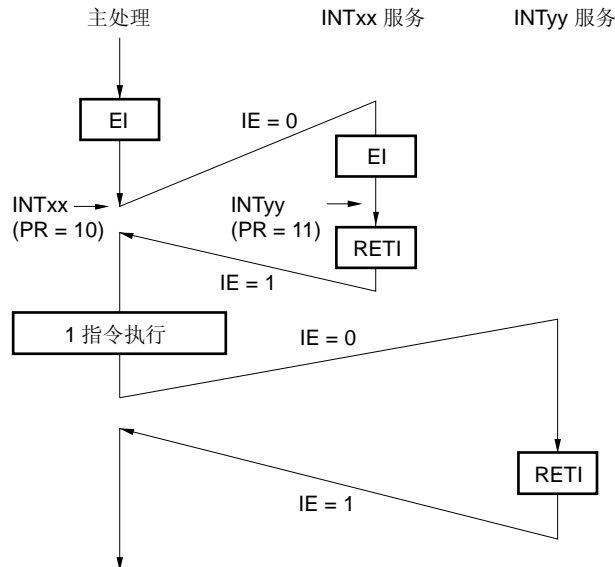
图 15-10. 中断嵌套示例(1/2)

例 1. 产生两次中断嵌套



在进行 INTxx 中断服务期间，响应了两个中断请求 INTyy 和 INTzz，这时就产生了中断嵌套。在响应每个中断之前，必须先执行 EI 指令才能允许响应中断请求。

例 2. 由于优先级控制没有产生中断嵌套



在 INTxx 中断服务期间不响应中断请求 INTyy，因为 INTyy 的优先级低于 INTxx，这样就不会产生中断嵌套。中断请求 INTyy 处于等待状态，在执行一条主程序指令后响应该中断请求。

PR = 00: 通过 $\times\times PR1\times = 0$, $\times\times PR0\times = 0$ (较高优先级)，指定等级 0

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 $\times\times PR1\times = 1$, $\times\times PR0\times = 1$ (较低优先级)，指定等级 3

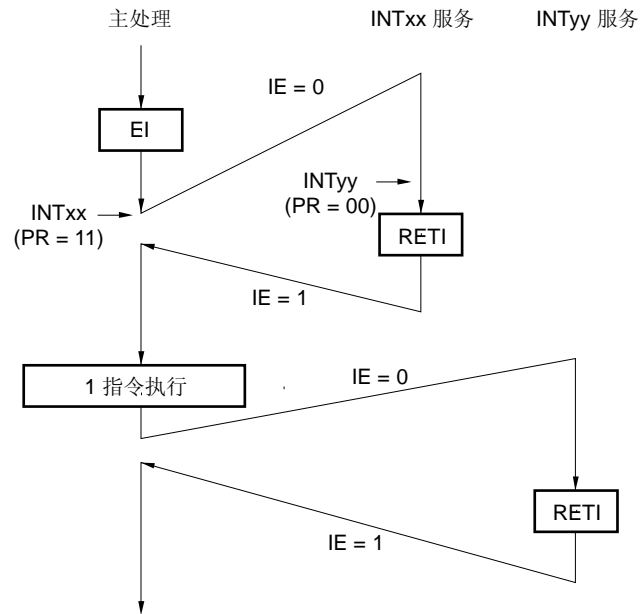
IE = 0: 禁止响应中断请求

IE = 1: 允许响应中断请求。

<R>

图 15-10. 中断嵌套示例(2/2)

例 3. 由于不允许中断没有产生中断嵌套



在 INT_{xx} 中断服务期间不允许中断（不执行 EI 指令），因此不响应中断请求 INT_{yy} ，这样也不会产生中断嵌套。中断请求 INT_{yy} 处于等待状态，在执行一条主程序指令后响应该中断请求。

PR = 00: 通过 $\times \times PR1 \times = 0$, $\times \times PR0 \times = 0$, 指定等级 0（较高优先级）

PR = 01: 通过 $\times \times PR1 \times = 0$, $\times \times PR0 \times = 1$, 指定等级 1

PR = 10: 通过 $\times \times PR1 \times = 0$, $\times \times PR0 \times = 1$, 指定等级 2

<R> 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: 禁止响应中断请求

IE = 1: 允许响应中断请求。

15.4.4 保持中断请求

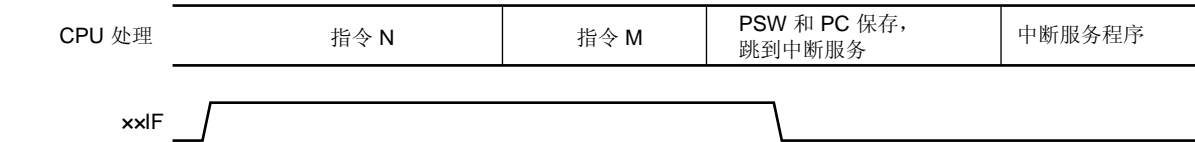
在某些指令执行期间，即使出现中断请求，请求响应也要保持等待状态，直到下一条指令执行结束。以下列出这类指令（中断请求保持指令）。

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, \$addr8
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- 用于 IF0L、IF0H、IF1L、IF1H、IF2L、IF2H、MK0L、MK0H、MK1L、MK1H、MK2L、MK2H、PR00L、PR00H、PR01L、PR01H、PR02L、PR02H、PR10L、PR10H、PR11L、PR11H、PR12L 和 PR12H 寄存器的操作指令。

注意事项 **BRK** 指令不属于上述列出的中断请求保持指令。但通过执行 **BRK** 指令激活的软件中断会将 **IE** 标志清零。因此，即使在执行 **BRK** 指令期间产生可屏蔽中断请求，该中断请求也不会被响应。

图 15-11 显示了处于等待状态的中断请求时序。

图 15-11. 中断请求保持



- 备注**
1. 指令 N：中断请求保持指令
 2. 指令 M：除中断请求保持指令之外的指令
 3. xxPR（优先级）的值不会影响xxIF（中断请求）的操作。

16.1 待机功能及配置

16.1.1 待机功能

待机功能用于减少系统的工作电流，有以下两种模式。

(1) HALT 模式

通过执行 HALT 指令设置 HALT 模式。在 HALT 模式中，CPU 操作时钟停止。HALT 模式设置之前，如果高速系统时钟振荡器、内部高速振荡器、40 MHz 内部高速振荡器、或副系统时钟振荡器正在运行，则每种时钟的振荡继续。在此模式下，工作电流的下降在不如 STOP 模式中下降得多，但 HALT 模式对于中断请求产生后立即重启操作和频繁进行间断操作非常有效。

(2) STOP 模式

通过执行 STOP 指令设置 STOP 模式。在 STOP 模式中，高速系统时钟振荡器和内部高速振荡器停止操作，整个系统的操作停止，这样 CPU 的工作电流将会大幅下降。
因为可通过中断请求清除该模式，所以可以执行中断操作。但是，选择 X1 时钟时，在释放 STOP 模式后，因为需要一段等待时间以确保振荡器振荡稳定，因此如果需要在产生中断请求后立即进行处理，则应选择 HALT 模式。

在这两种模式中，寄存器、标志和数据存储器的内容将会保持进入待机模式前的内容。I/O 端口输出锁存器和输出缓冲器的状态也将被保持。

- 注意事项
- 1. 仅当 CPU 使用主系统时钟时，才能使用 STOP 模式。当 CPU 使用副系统时钟时，不能设置 STOP。而当 CPU 无论是使用主系统时钟还是副系统时钟，均可以使用 HALT 模式。
 - 2. 当切换到 STOP 模式时，确保在执行 STOP 指令前，停止使用主系统时钟的外围硬件的操作。
 - 3. 当使用待机功能时，建议采用以下步骤降低 A/D 转换器的操作电流：首先将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 和第 0 位 (ADCE) 清零以停止 A/D 转换操作，然后执行 STOP 指令。
 - 4. 当使用待机功能时，建议采用以下步骤降低比较器的操作电流：首先将比较器 n 控制寄存器 (CnCTL) 的第 7 位 (CnEN) 和比较器 n 内部参考电压选择寄存器的第 7 位 (CnVRE) 清零，从而停止比较器的操作，然后执行 STOP 指令。

<R>

<R> 备注 n = 0, 1

<R>

- 注意事项
5. 当使用待机功能时，建议采用以下步骤降低可编程增益放大器的操作电流：首先将可编程增益放大器控制寄存器 (OAM) 的第 7 位 (OAEN) 清零，从而停止可编程增益放大器的操作，然后执行 STOP 指令。
 6. 在 HALT 或 STOP 模式下，可通过选项字节选择内部低速振荡器是继续还是停止。详细信息参考第 21 章 选项字节。
 7. 当 CPU 使用 40 MHz 内部高速振荡时钟时，不能执行 STOP 指令。确保在变换到内部高速振荡时钟工作后执行 STOP 指令。

16.1.2 控制待机功能的寄存器

待机功能由以下两个寄存器控制。

- 振荡稳定时间计数器状态寄存器(OSTC)
- 振荡稳定时间选择寄存器(OSTS)

备注 有关启动、停止以及选择时钟的寄存器的情况，可参考第 5 章 时钟发生器。

(1) 振荡稳定时间计数器状态寄存器(OSTC)

这是表明 X1 时钟振荡稳定时间计数器的计数状态的寄存器。

X1 时钟振荡稳定时间在如下情况可以检测。

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

可由 1 位或 8 位存储器操作指令读取 OSTC。

当复位释放（通过 **RESET** 输入、POC、LVI、WDT 和执行非法指令复位）、STOP 指令以及 MSTOP（CSC 寄存器的第 7 位）=1 时，将此寄存器清零（00H）。

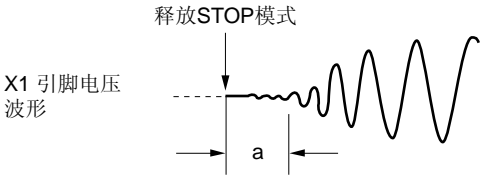
图 16-1. 振荡稳定时间计数器状态寄存器（OSTC）的格式

地址：FFFA2H 复位后：00H R

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

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

- 注意事项
- 在经历上述时间后，各个位从 **MOST8** 起按顺序被设置为 1，并保持为 1。
 - 振荡稳定时间计数器的计数达到 **OSTS** 设置的振荡稳定时间。如果已进入 **STOP** 模式，并在 CPU 使用内部高速振荡时钟时释放 **STOP** 模式，则可以按以下方式设置振荡稳定时间。
 - 预期的 **OSTC** 振荡稳定时间 \leq **OSTS** 设置的振荡稳定时间因此需要注意，在释放 **STOP** 模式后，只有达到 **OSTS** 设置的振荡稳定时间的状态被设置到 **OSTC**。
 - X1** 时钟振荡稳定等待时间不包含直到时钟振荡开始的时间（下为“a”）。



备注 f_x : X1 时钟振荡频率

(2) 振荡稳定时间选择寄存器(OSTS)

此寄存器用于选择 STOP 模式释放时的 X1 时钟振荡稳定等待时间。

在释放 STOP 模式后，且 CPU 使用 X1 时钟时，操作等待由 OSTS 设置的时间。

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

可由 8 位存储器操作指令设置 OSTS。

复位信号的产生将 OSTS 设置为 07H。

图 16-2. 振荡稳定时间选择寄存器(OSTS)的格式

地址: FFFA3H 复位后: 07H R/W

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

OSTS2	OSTS1	OSTS0	振荡稳定时间选择		
				f _x = 10 MHz	f _x = 20 MHz
0	0	0	2 ⁸ /f _x	25.6 μs	设置禁止
0	0	1	2 ⁹ /f _x	51.2 μs	25.6 μs
0	1	0	2 ¹⁰ /f _x	102.4 μs	51.2 μs
0	1	1	2 ¹¹ /f _x	204.8 μs	102.4 μs
1	0	0	2 ¹³ /f _x	819.2 μs	409.6 μs
1	0	1	2 ¹⁵ /f _x	3.27 ms	1.64 ms
1	1	0	2 ¹⁷ /f _x	13.11 ms	6.55 ms
1	1	1	2 ¹⁸ /f _x	26.21 ms	13.11 ms

- 注意事项
1.

如果在 X1 时钟用作 CPU 时钟时要设置 STOP 模式，则应在执行 STOP 指令之前设置 OSTS。
2.

禁止设置振荡稳定时间为 20 μs 或更短。
3.

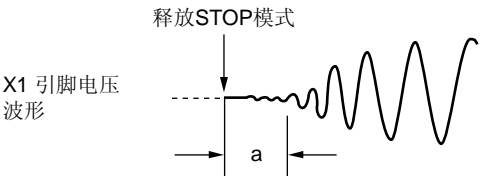
改变 OSTS 寄存器的设置之前，确认 OSTC 寄存器的计数操作已经完成。
4.

在 X1 时钟振荡稳定期间，不要修改 OSTS 寄存器的值。
5.

振荡稳定时间计数器的计数达到 OSTS 设置的振荡稳定时间。如果已进入 STOP 模式，并在 CPU 使用内部高速振荡时钟时释放 STOP 模式，则可以按以下方式设置振荡稳定时间。

• 预期的 OSTC 振荡稳定时间 ≤ OSTS 设置的振荡稳定时间
6.

X1 时钟振荡稳定等待时间不包含直到时钟振荡开始的时间（下为“a”）。
- 因此需要注意，在释放 STOP 模式后，只有达到 OSTS 设置的振荡稳定时间的状态被设置到 OSTC。



备注 f_x: X1 时钟振荡频率

16.2 待机功能的操作

16.2.1 HALT模式

(1) HALT 模式

通过执行 HALT 指令设置 HALT 模式。无论设置前 CPU 时钟是高速系统时钟、内部高速振荡时钟、40MHz 内部高速振荡时钟、还是副系统时钟，都可以设置为 HALT 模式。

HALT 模式中的操作状态如下所示。

表 16-1. HALT 模式的操作状态 (1/2)

HALT 模式设置 项目			当 CPU 使用主系统时钟并执行 HALT 指令时		
			当 CPU 使用内部高速振荡时钟 (f _{IH})或 40 MHz 内部高速振荡时钟 (f _{DSC})时，	当 CPU 使用 X1 时钟 (f _X) 时	当 CPU 使用外部主系统时钟 (f _{EX}) 时
系统时钟			停止 CPU 时钟		
	主系统时钟	f _{IH} , f _{DSC}	操作继续 (不能停止)		保持设置 HALT 模式前的状态
		f _X	保持设置 HALT 模式前的状态	操作继续 (不能停止)	不能操作
		f _{EX}		不能操作	操作继续 (不能停止)
	副系统时钟	f _{XT}	保持设置 HALT 模式前的状态		
	f _{IL}		通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 来设置 <ul style="list-style-type: none">• WTON = 0: 停止• WTON = 1 且 WDSTBYON = 1: 振荡• WTON = 1 且 WDSTBYON = 0: 停止		
CPU			操作停止		
Flash 存储器			操作停止		
RAM			值被保持		
端口(锁存器)			保持设置 HALT 模式前的状态		
定时器阵列单元 TAUS			可操作		
变频器控制功能					
实时计数器 (RTC)					
看门狗定时器			通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 来设置 <ul style="list-style-type: none">• WTON = 0: 停止，• WTON = 1 且 WDSTBYON = 1: 操作，• WTON = 1 且 WDSTBYON = 0: 停止		
A/D 转换器			可操作		
可编程增益放大器					
比较器					
串行阵列单元 (SAU)					
乘法器/除法器					
DMA 控制器					
上电清零功能					
低电压检测功能					
外部中断					

备注

f_{IH}: 内部高速振荡时钟

f_{DSC}: 40 MHz 内部高速振荡时钟

f_X: X1 时钟

f_{EX}: 外部主系统时钟

f_{XT}: XT1 时钟

f_{IL}: 内部低速振荡时钟

表 16-1. HALT 模式的操作状态 (2/2)

HALT 模式设置			当 CPU 使用副系统时钟并执行 HALT 指令时
项目			当 CPU 使用 XT1 时钟(f_{xt})时
系统时钟			停止 CPU 时钟
主系统时钟	f_{IH}, f_{DSC}	保持设置 HALT 模式前的状态	
	f_X		
		f_{EX}	操作或由外部时钟输入停止
副系统时钟		f_{XT}	操作继续 (不能停止)
f_{IL}		通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 来设置 <ul style="list-style-type: none">• WTON = 0: 停止• WTON = 1 且 WDSTBYON = 1: 振荡• WTON = 1 且 WDSTBYON = 0: 停止	
CPU			操作停止
Flash 存储器			操作停止 (低功耗电流模式下的等待状态)
RAM			值被保持
端口(锁存器)			保持设置 HALT 模式前的状态
定时器阵列单元 TAUS			可操作
变频器控制功能			不能操作
实时计数器 (RTC)			可操作
看门狗定时器			通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 来设置 <ul style="list-style-type: none">• WTON = 0: 停止• WTON = 1 且 WDSTBYON = 1: 操作• WTON = 1 且 WDSTBYON = 0: 停止
A/D 转换器			不能操作
可编程增益放大器			可操作
比较器			
串行阵列单元 (SAU)			
乘法器/除法器			
DMA 控制器			可操作
上电清零功能			
低电压检测功能			
外部中断			

备注

f_{IH} : 内部高速振荡时钟

f_{DSC} : 40 MHz 内部高速振荡时钟

f_X : X1 时钟

f_{EX} : 外部主系统时钟

f_{XT} : XT1 时钟

f_{IL} : 内部低速振荡时钟

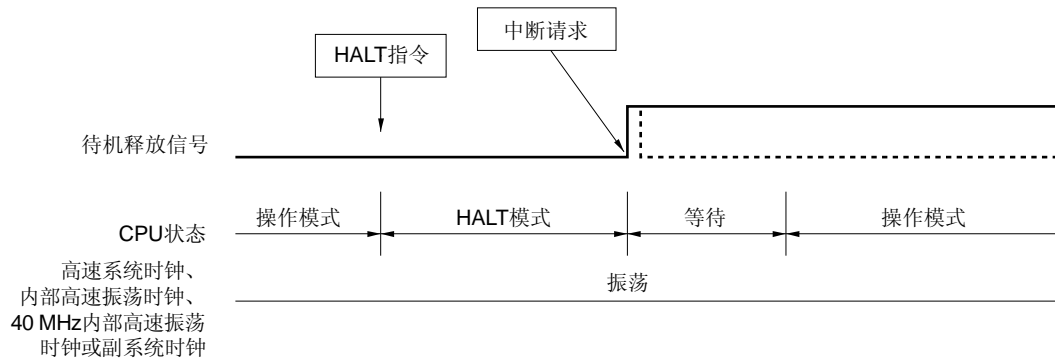
(2) 释放 HALT 模式

以下两种方式可以释放 HALT 模式。

(a) 由未被屏蔽的中断请求释放

当产生一个未被屏蔽的中断时，释放 HALT 模式。如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

图 16-3. 通过产生中断请求释放 HALT 模式



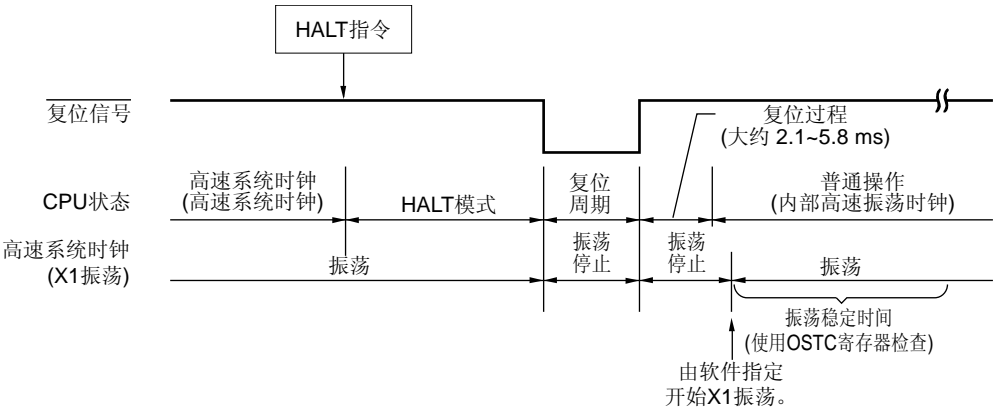
备注 虚线表示释放待机模式的中断请求被响应的情况。

- (b) 由复位信号的产生释放
- 当产生复位信号时，释放 HALT 模式，然后在进行正常复位操作后，程序从复位向量指向的地址处开始执行。

图 16-4. 通过复位释放 HALT 模式

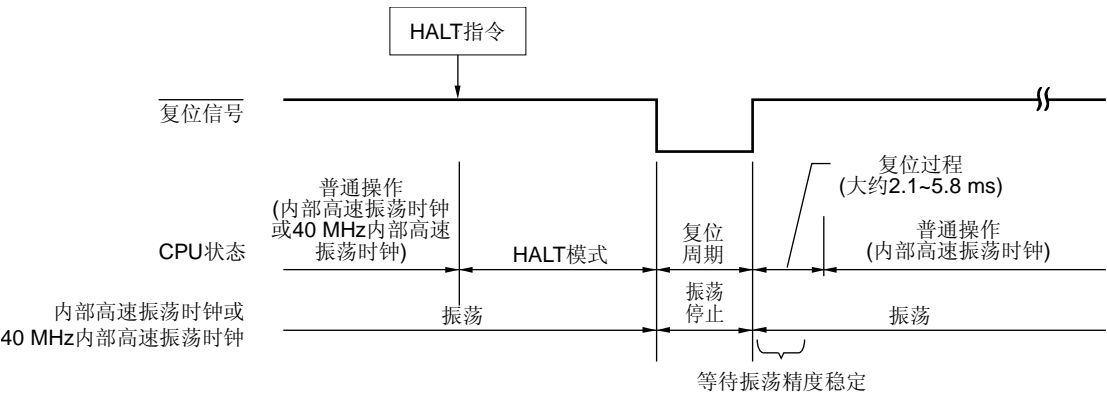
<R>

(1) 当 CPU 使用高速系统时钟时



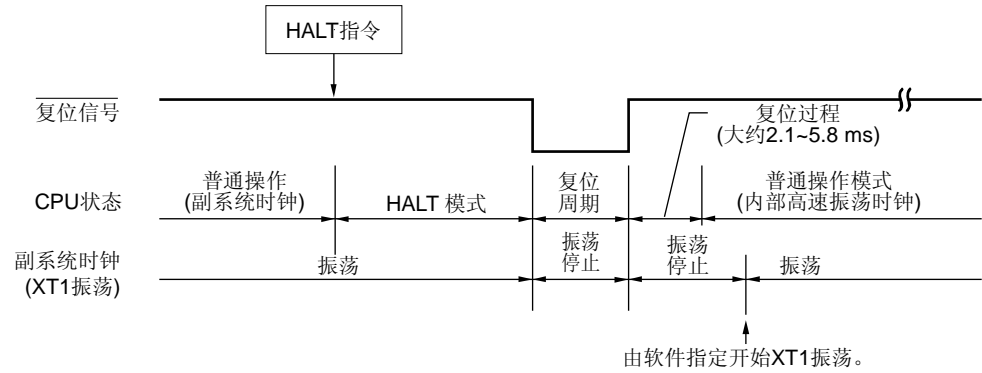
(2) CPU 使用内部高速振荡时钟或 40 MHz 内部高速振荡时钟
高速振荡时钟作为 CPU 时钟使用

<R>



(3) 当 CPU 使用副系统时钟

<R>



备注 fx: X1 时钟振荡频率

16.2.2 STOP模式

(1) STOP 模式设置及操作状态

通过执行 STOP 指令，设置 STOP 模式。而且只有在设置前 CPU 使用内部高速振荡时钟、X1 时钟、或者外部主系统时钟，才可以设置 STOP 模式。

- 注意事项
1. 由于中断请求信号用于释放待机模式，因此如果一个中断源的中断请求标志被设置且中断屏蔽标志被复位时，则立即清除待机模式。因此，在执行 STOP 指令后，STOP 模式立即被复位到 HALT 模式，并且在经历了振荡稳定时间选择寄存器 (OSTS) 设置的等待时间后，系统返回操作模式。
 2. 当 CPU 使用 40 MHz 内部高速振荡时钟时，不能执行 STOP 指令。确保在变换到内部高速振荡时钟工作后执行 STOP 指令。

STOP 模式中的操作状态如下所示。

表 16-2. STOP 模式的操作状态

STOP 模式设置 项目			当 CPU 使用主系统时钟并执行 STOP 指令时		
			当 CPU 使用内部高速振荡时钟 (f _{IH}) 时	当 CPU 使用 X1 时钟 (f _x) 时	当 CPU 使用外部主系统时钟 (f _{EX}) 时
系统时钟			停止 CPU 时钟		
主系统时钟	f _{IH}	停止			
	f _x				
	f _{EX}				
副系统时钟	f _{XT}	保持设置 STOP 模式前的状态			
f _{IL}		通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 来设置 <ul style="list-style-type: none">• WTON = 0: 停止• WTON = 1 且 WDSTBYON = 1: 振荡• WTON = 1 且 WDSTBYON = 0: 停止			
CPU			操作停止		
Flash 存储器			操作停止		
RAM			值被保持		
端口(锁存器)			保持设置 STOP 模式前的状态		
定时器阵列单元 TAUS			禁止操作		
变频器控制功能					
实时计数器 (RTC)			可操作		
看门狗定时器			通过选项字节 (000C0H) 的第 0 位 (WDSTBYON) 和第 4 位 (WTON) 来设置 <ul style="list-style-type: none">• WTON = 0: 停止• WTON = 1 且 WDSTBYON = 1: 操作• WTON = 1 且 WDSTBYON = 0: 停止		
A/D 转换器			禁止操作		
可编程增益放大器					
比较器					
串行阵列单元 (SAU)					
乘法器/除法器			禁止操作		
DMA 控制器					
上电清零功能			可操作		
低电压检测功能					
外部中断					

备注

f_{IH} : 内部高速振荡时钟

f_x : X1 时钟

f_{EX} : 外部主系统时钟

f_{XT} : XT1 时钟

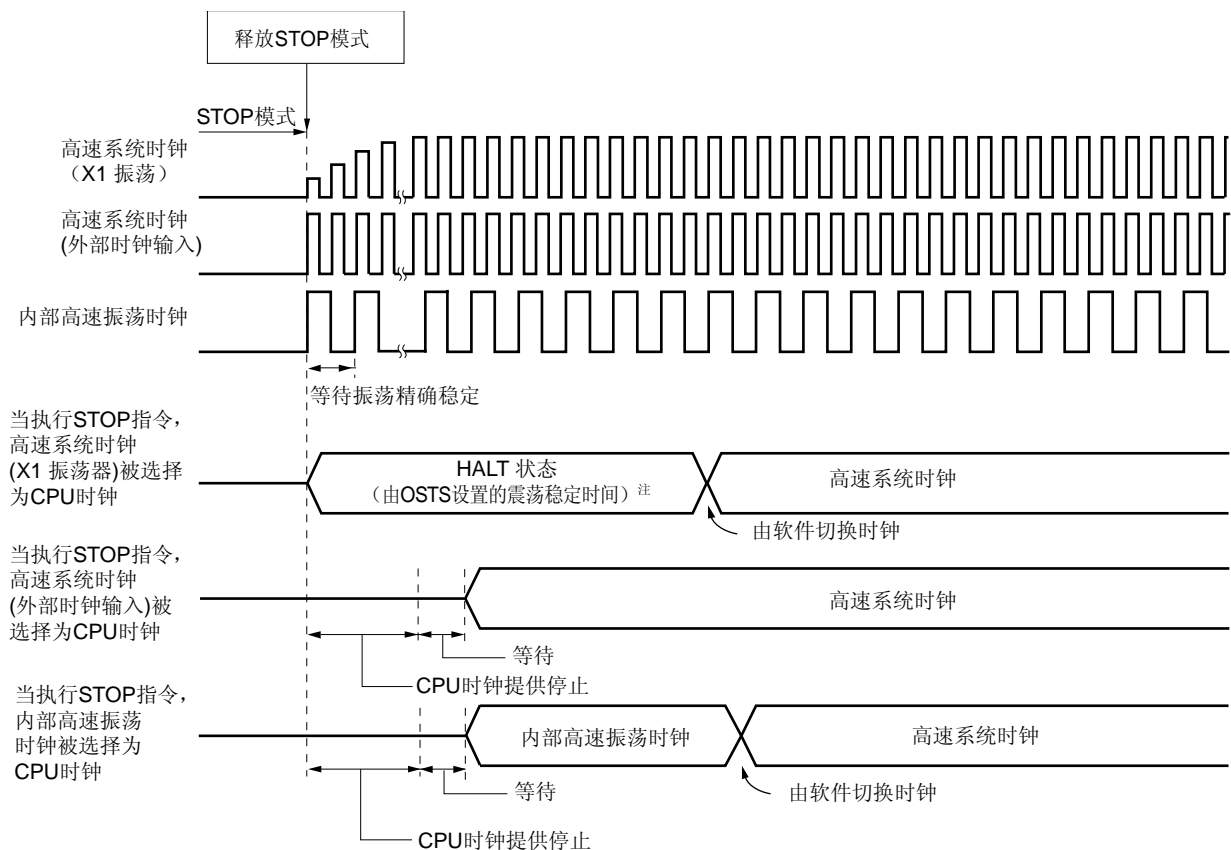
f_{IL} : 内部低速振荡时钟

- 注意事项
1. 在 STOP 模式释放后，要使用在 STOP 模式下停止操作的外围硬件，以及在 STOP 模式下时钟停止振荡的外围硬件，必须重新启动外围硬件。
 2. 要在 STOP 模式下停止内部低速振荡时钟，可在 HALT/STOP 模式下使用选项字节停止看门狗定时器的操作 (000C0H 的第 0 位(WDSTBYON) = 0)，然后执行 STOP 指令。
 3. 当 CPU 使用高速系统时钟 (X1 振荡) 时，为了在 STOP 模式释放后缩短振荡稳定时间，可在执行下一个 STOP 指令前，将 CPU 时钟暂时切换到内部高速振荡时钟。STOP 模式释放后，在将 CPU 时钟从内部高速振荡时钟切换到高速系统时钟 (X1 振荡) 之前，应使用振荡稳定时间计数器的状态寄存器 (OSTC) 检测振荡稳定时间。
 4. 当 CPU 使用 40 MHz 内部高速振荡时钟时，不能执行 STOP 指令。确保在变换到内部高速振荡时钟工作后执行 STOP 指令。

<R>

(2) 释放 STOP 模式

图 16-5. 释放 STOP 模式时的操作时序



注 当由 OSTS 设置的振荡稳定时间等于或小于 $61\mu\text{s}$ 时，HALT 状态最大保持为“ $61\mu\text{s} + \text{等待时间}$ ”。

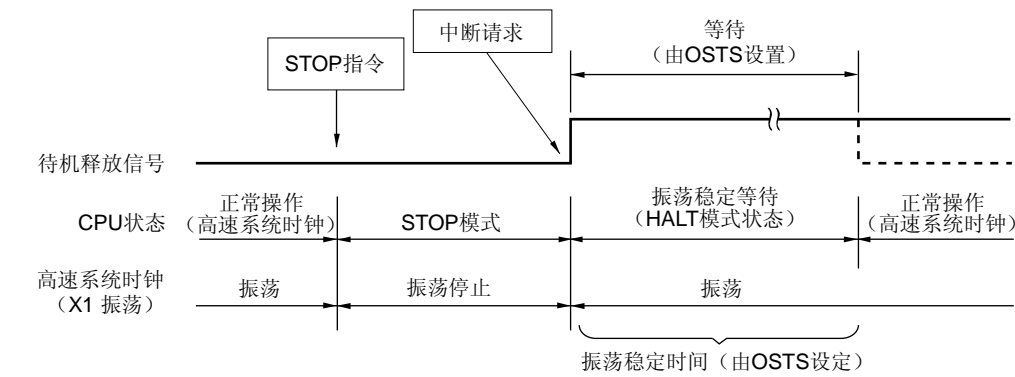
STOP 模式可通过以下两种方式释放。

(a) 由未被屏蔽的中断请求释放

当产生一个未被屏蔽的中断请求时，释放 STOP 模式。经历振荡稳定时间后，如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

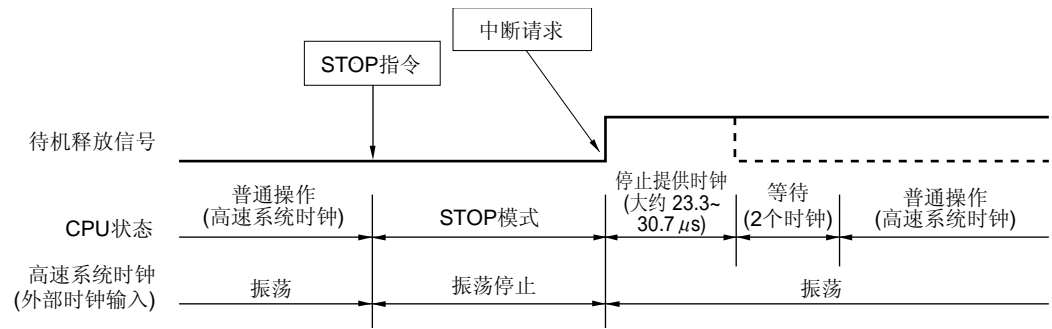
图 16-6. 通过产生中断请求释放 STOP 模式(1/2)

(1) 当 CPU 使用高速系统时钟（X1 振荡器）时



<R>

(2) 当 CPU 使用高速系统时钟（外部时钟输入）时

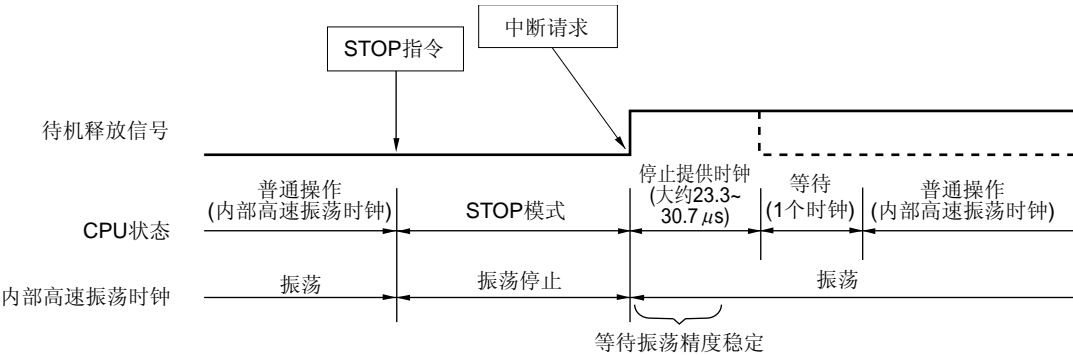


备注 虚线表示释放待机模式的中断请求被响应的情况。

图 16-6. 通过产生中断请求释放 STOP 模式(2/2)

<R>

(3) 当 CPU 使用内部高速振荡时钟时



备注 虚线表示释放待机模式的中断请求被响应的情况。

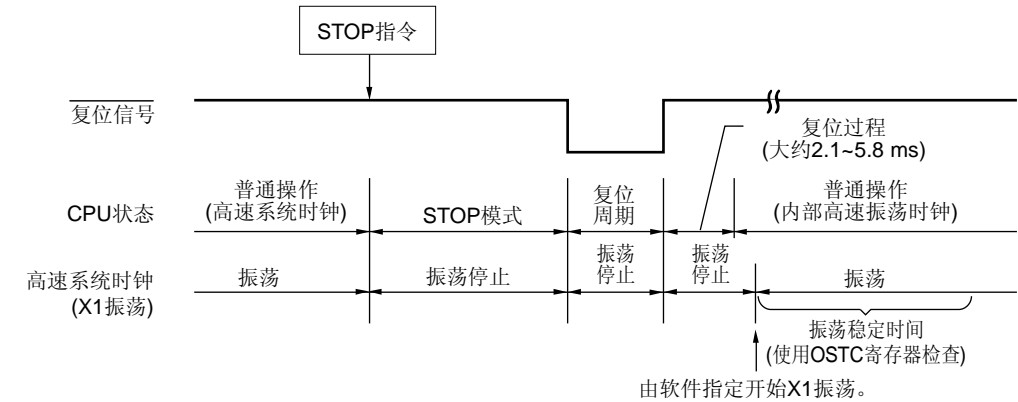
(b) 由复位信号的产生释放

当产生复位信号时，释放 STOP 模式，然后在正常复位操作情况下，程序转向复位向量地址然后执行。

图 16-7. 由复位释放 STOP 模式

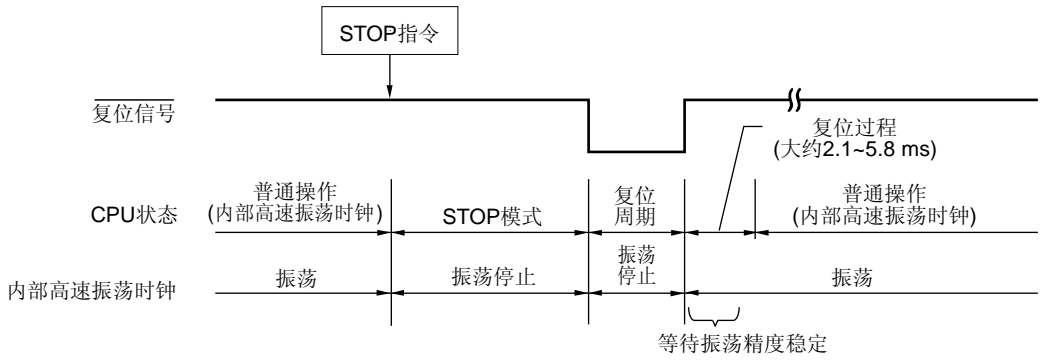
<R>

(1) 当 CPU 使用高速系统时钟时



(2) 当 CPU 使用内部高速振荡时钟时

<R>



备注 f_x : X1 时钟振荡频率

第 17 章 复位功能

以下 5 种操作可产生一个复位信号。

- (1) 由 $\overline{\text{RESET}}$ 引脚输入的外部复位信号
- (2) 由看门狗定时器程序循环检测引起的内部复位
- (3) 通过比较上电清零 (POC) 电路的检测电压和供电电压引起的内部复位
- (4) 通过低电压检测电路 (LVI) 的供电电压或外部输入引脚的输入电压 (EXLVI) 和检测电压的比较引起的内部复位
- (5) 由执行非法指令引起的内部复位[※]

外部复位与内部复位一样，当产生复位信号时，程序都是从地址 0000H 和 0001H 处开始执行。

当低电平输入到 $\overline{\text{RESET}}$ 引脚、看门狗定时器溢出、POC 和 LVI 电路的电压检测或执行非法指令[※]，都可以使复位有效，且硬件的各项设置状态如表 17-1 和 17-2 所示。在复位信号产生期间或在复位释放后振荡稳定时间期间，每个引脚均为高阻抗。

当 $\overline{\text{RESET}}$ 引脚输入低电平时，设备被复位。当 $\overline{\text{RESET}}$ 引脚输入高电平时，设备从复位状态释放，并在复位处理后使用内部高速振荡时钟执行程序。由看门狗定时器引起的复位自动释放，并在复位处理后使用内部高速振荡时钟执行程序 (参见图 17-2~17-4)。由 POC 和 LVI 电路复位 由 POC 和 LVI 电路供电电压检测引起的复位，在复位后，当 $V_{DD} \geq V_{POR}$ 或 $V_{DD} \geq V_{LVI}$ 时自动释放，并使用内部高速振荡时钟执行程序 (参见第 18 章 上电清零电路 和 第 19 章 低电压检测电路)。

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

注意事项 1. 对于外部复位，输入 $\overline{\text{RESET}}$ 引脚的低电平的时间至少应为 10 μs 。

(要实现电源应用上的一个外部复位，那么必须持续一个至少 10 μs 的低电平，在此期间供电电压在操作范围内 ($V_{DD} \geq 2.7 \text{ V}$)。)

2. 复位输入期间，X1 时钟、XT1 时钟、内部高速振荡时钟和内部低速振荡时钟停止振荡。外部主系统时钟输入无效。

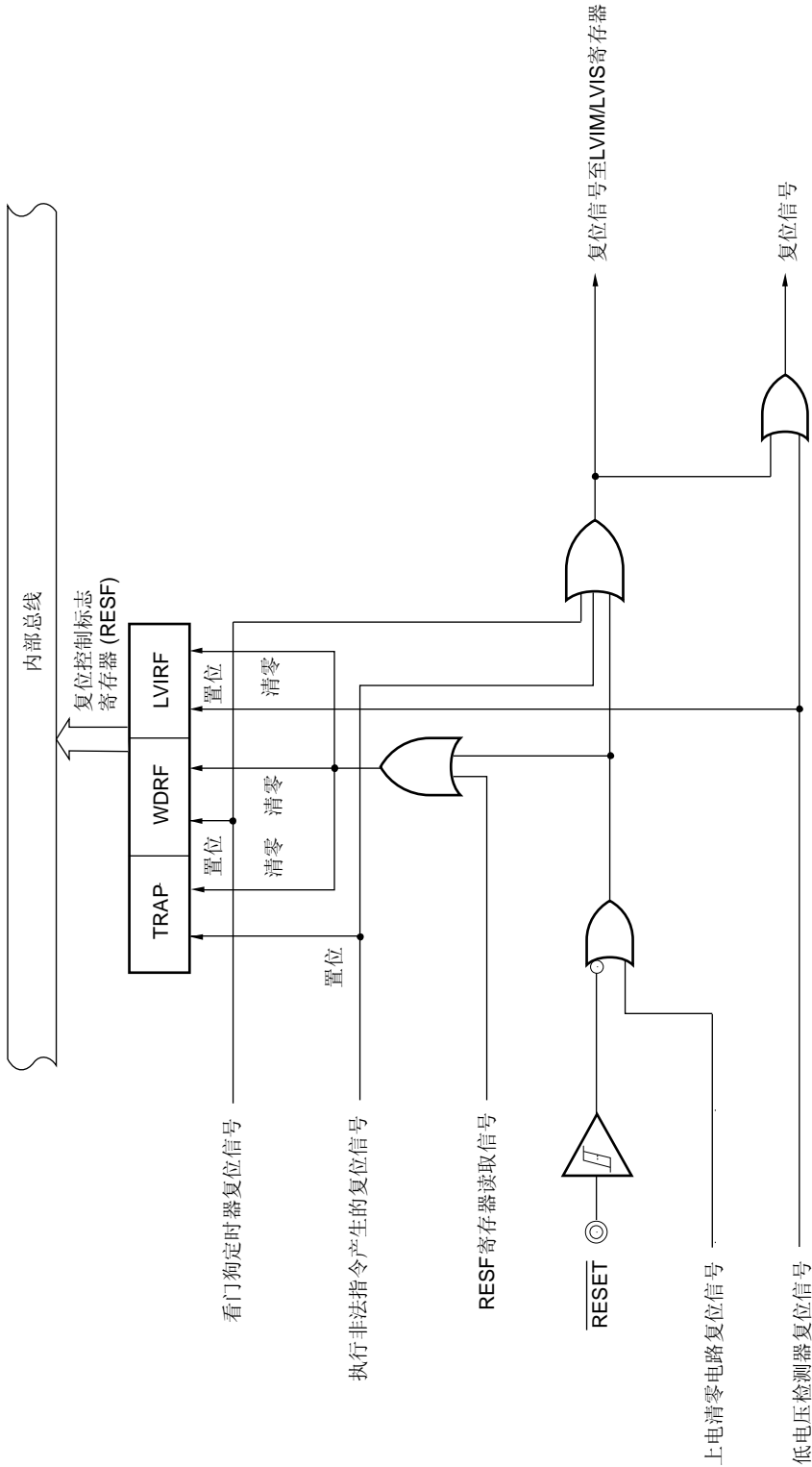
3. 在通过复位释放 STOP 模式时，复位输入期间保持 STOP 模式下的 RAM 内容。

4. 当复位有效时，端口引脚都变成高阻态，因为 SFR 和第二个 SFR 被初始化。

备注 V_{POR} : POC 供电上升检测电压

图 17-1. 复位功能的框图

<R>



注意事项 LVI 电路内部复位功能不能复位 LVI 电路。

- 备注
- 1. LVIM: 低电压检测寄存器
 - 2. LVIS: 低电压检测电平选择寄存器

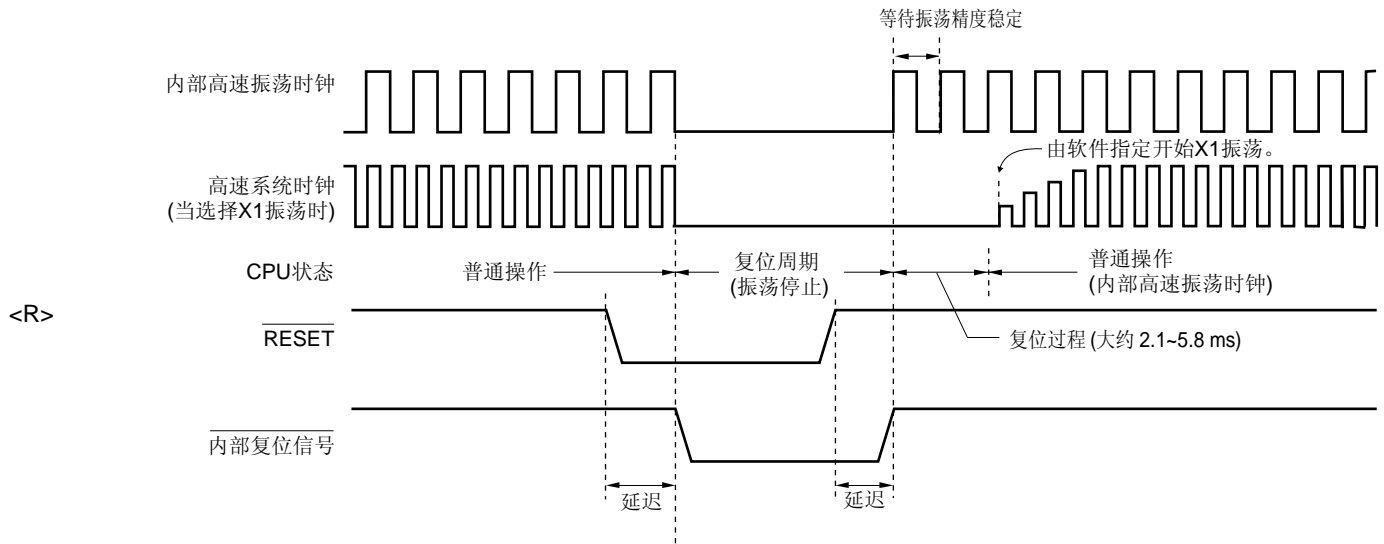
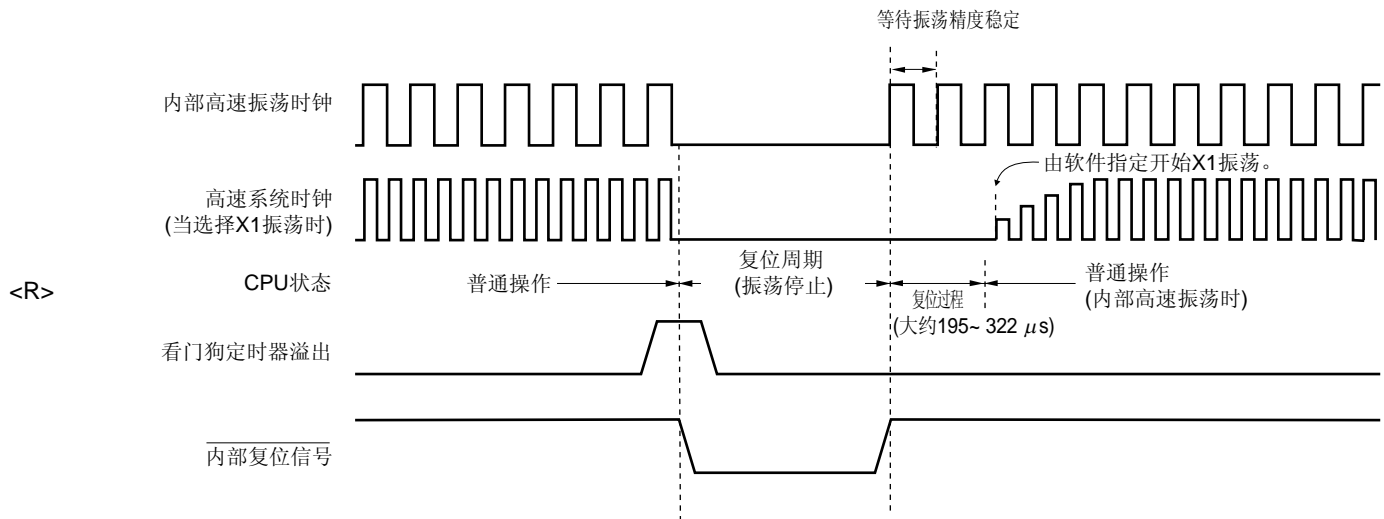
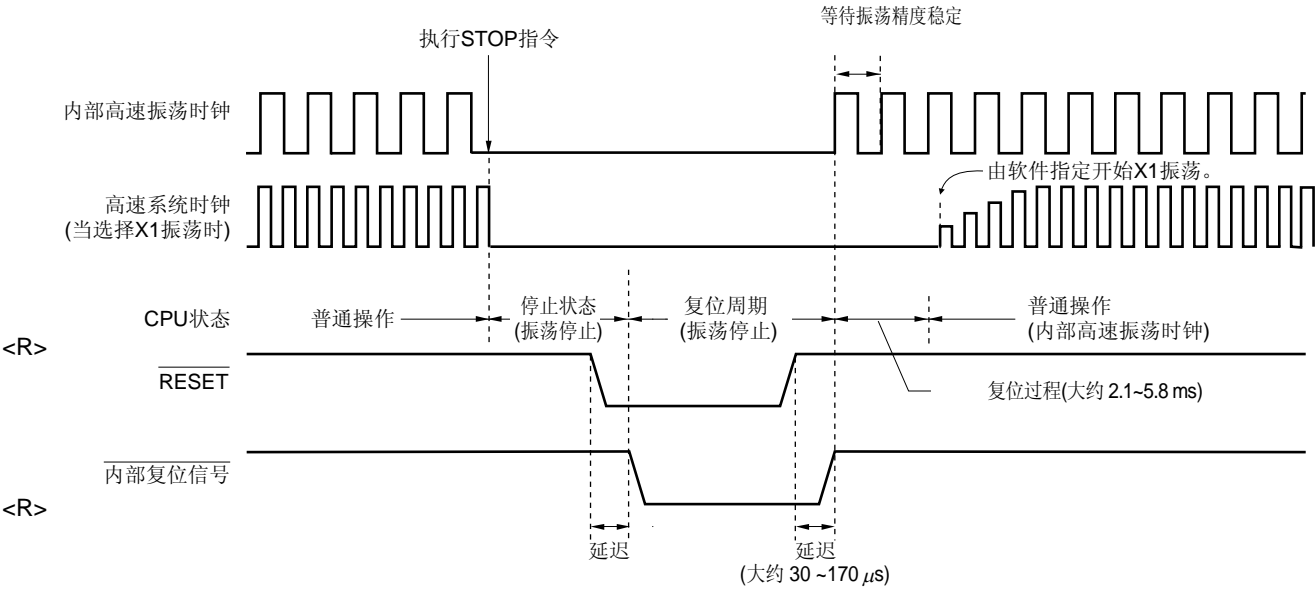
图 17-2. 由 $\overline{\text{RESET}}$ 输入进行复位的时序

图 17-3. 由于看门狗定时器溢出进行复位的时序



注意事项 看门狗定时器内部复位看门狗定时器。

图 17-4. 在 STOP 模式下通过 RESET 输入进行复位的时序



备注 关于上电清零电路和低电压检测器的复位时序，可参见第 18 章 上电清零电路和第 19 章 低电压检测电路。

表 17-1. 复位期间的操作状态

<R>

项目			复位期间
系统时钟			CPU 时钟 停止。
主系统时钟	f _{IH} , f _{DSC}		操作停止
		f _X	操作停止(X1 和 X2 引脚为输入端口模式)
		f _{EX}	时钟输入无效(引脚为输入端口模式)
	副系统时钟	f _{XT}	操作停止(XT1 和 XT2 引脚为输入端口模式)
	f _{IL}		操作停止
CPU			操作停止(然而, 当电压至少为上电清零检测电压时, 数值仍被保留。)
Flash 存储器			
RAM			
定时器阵列单元 TAUS			
实时计数器 (RTC)			
看门狗定时器			
A/D 转换器			
可编程增益放大器			
比较器			
串行阵列单元 (SAU)			
乘法器/除法器			
DMA 控制器			
上电清零功能			
低电压检测功能			
外部中断			

备注

f_{IH}: 内部高速振荡时钟
 f_{DSC}: 40 MHz 内部高速振荡时钟
 f_X: X1 振荡时钟
 f_{EX}: 外部主系统时钟
 f_{XT}: XT1 振荡时钟
 f_{IL}: 内部低速振荡时钟

表 17-2. 复位响应后的硬件状态(1/4)

硬件		复位响应后 ^{注1}
程序计数器 (PC)		设置复位向量表的内容 (0000H、0001H)
堆栈指针 (SP)		未定义
程序状态字 (PSW)		06H
RAM	数据存储器	不确定 ^{注2}
	通用寄存器	不确定 ^{注2}
端口寄存器 (P1~P5、P7、P8、P12、P15 ^{注3}) (输出锁存器)		00H
端口模式寄存器 (PM1~PM5、PM6 ^{注3} 、PM7、PM8、PM12、PM15 ^{注3})		FFH
端口输入模式寄存器(PIM3、PIM7、PIM8)		00H
端口输出模式寄存器(POM3、POM7)		00H
上拉电阻选项寄存器 (PU1、PU3~PU5、PU7、PU12)		00H
时钟操作模式控制寄存器(CMC)		00H
时钟操作状态控制寄存器(CSC)		C0H
处理器模式控制寄存器 (PMC)		00H
系统时钟控制寄存器(CKC)		09H
40 MHz 内部高速振荡控制寄存器(DSCCTL)		00H
振荡稳定时间计数器状态寄存器(OSTC)		00H
振荡稳定时间选择寄存器(OSTS)		07H
噪声滤波使能寄存器 0, 1, 2 (NFEN0, NFEN1, NFEN2)		00H
外围设备使能寄存器 0, 1, 2(PER0, PER1, PER2)		00H
操作速度模式控制寄存器(OSMC)		00H
定时器阵列单元 (TAUS)	定时器数据寄存器 00、01、02、03、04、05、06、07、08、09、10、11 (TDR00、TDR01、TDR02、TDR03、TDR04、TDR05、TDR06、TDR07、TDR08、TDR09、TDR10、TDR11)	0000H
	定时器模式寄存器 00、01、02、03、04、05、06、07、08、09、10、11 (TMR00、TMR01、TMR02、TMR03、TMR04、TMR05、TMR06、TMR07、TMR08、TMR09、TMR10、TMR11)	0000H
	定时器状态寄存器 00、01、02、03、04、05、06、07、08、09、10、11 (TSR00、TSR01、TSR02、TSR03、TSR04、TSR05、TSR06、TSR07、TSR08、TSR09、TSR10、TSR11)	0000H
	定时输入选择寄存器 0(TIS0)	00H
	定时器计数寄存器 00、01、02、03、04、05、06、07、08、09、10、11 (TCR00、TCR01、TCR02、TCR03、TCR04、TCR05、TCR06、TCR07、TCR08、TCR09、TCR10、TCR11)	FFFFH
	定时器通道使能状态寄存器 0(TE0)	0000H
	定时器通道启动寄存器 0(TS0)	0000H
	定时器通道停止寄存器 0(TT0)	0000H
	定时器时钟选择寄存器 0(TPS0)	0000H
	定时器输出寄存器 0(TO0)	0000H
	定时器输出使能寄存器 0(TOE0)	0000H
	定时器输出电平寄存器 0(TOL0)	0000H

- 注 1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。
2. 当在待机模式下执行复位时，即使在复位后预复位状态也将保持。

表 17-2. 复位响应后的硬件状态(2/4)

硬件		复位响应后 ^{※1}
定时器阵列单元(TAUS)	定时器输出模式寄存器 0(TOM0)	0000H
	定时器三角波输出模式寄存器 0(TOT0)	0000H
	定时器实时输出使能寄存器 0(TRE0)	0000H
	定时器实时输出寄存器 0(TRO0)	0000H
	定时器实时控制寄存器 0(TRC0)	0000H
	定时器调制输出使能寄存器 0(TME0)	0000H
	定时器死区时间输出使能寄存器 0(TDE0)	0000H
	TAU 可选模式寄存器(OPMR)	0000H
	TAU 可选状态寄存器(OPSR)	0000H
	TAU 可选高阻启动触发寄存器(OPHS)	0000H
	TAU 可选高阻停止触发寄存器(OPHT)	0000H
实时计数器	副计数寄存器 (RSUBC)	0000H
	秒计数寄存器 (SEC)	00H
	分钟计数寄存器 (MIN)	00H
	小时计数寄存器 (HOUR)	12H
	星期计数寄存器 (WEEK)	00H
	日计数寄存器 (DAY)	01H
	月计数寄存器 (MONTH)	01H
	年计数寄存器 (YEAR)	00H
	钟表错误修正寄存器 (SUBCUD)	00H
	分钟报警寄存器(ALARMWM)	00H
	小时报警寄存器(ALARMWH)	12H
	星期报警寄存器 ALARMWW)	00H
	实时计数器控制寄存器 0 (RTCC0)	00H
	实时计数器控制寄存器 1 (RTCC1)	00H
	实时计数器控制寄存器 2 (RTCC2)	00H
看门狗定时器	使能寄存器(WDTE)	1AH/9AH ^{※2}
A/D 转换器	10 位 A/D 转换结果寄存器 (ADCR)	0000H
	8 位 A/D 转换结果寄存器 (ADCRH)	00H
	模式寄存器 (ADM)	00H
	模式寄存器 1(ADM1)	00H
	模式输入通道指定寄存器(ADS)	00H
	A/D 端口配置 寄存器 (ADPC)	10H

- 注
1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。
 2. WDTE 的复位数值由选项字节设置确定。

表 17-2. 复位响应后的硬件状态(3/4)

硬件		复位响应后 ^注
串行阵列单元 (SAU)	串行数据寄存器 00, 01, 02, 03 (SDR00, SDR01, SDR02, SDR03)	0000H
	串行状态寄存器 00, 01, 02, 03 (SSR00, SSR01, SSR02, SSR03)	0000H
	串行标志清零触发寄存器 00, 01, 02, 03 (SIR00, SIR01, SIR02, SIR03)	0000H
	串行模式寄存器 00, 01, 02, 03 (SMR00, SMR01, SMR02, SMR03)	0020H
	串行通信操作设置寄存器 00, 01, 02, 03 (SCR00, SCR01, SCR02, SCR03)	0087H
	串行通道使能状态寄存器 0 (SE0)	0000H
	串行通道启动触发寄存器 0 (SS0)	0000H
	串行通道停止触发寄存器 0 (ST0)	0000H
	串行时钟选择寄存器 0 (SPS0)	0000H
	串行输出寄存器 0 (SO0)	0F0FH
	串行输出使能寄存器 0 (SOE0)	0000H
	输入切换控制寄存器 (ISC)	00H
乘法器/除法器	乘法/除法数据寄存器 A (L) (MDAL)	0000H
	乘法/除法数据寄存器 A (H) (MDAH)	0000H
	乘法/除法数据寄存器 B (L) (MDBL)	0000H
	乘法/除法数据寄存器 B (H) (MDBH)	0000H
	乘法/除法数据寄存器 C (L) (MDCL)	0000H
	乘法/除法数据寄存器 C (H) (MDCH)	0000H
	乘法/除法控制寄存器 (MDUC)	00H

注 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。

表 17-2. 复位响应后的硬件状态(4/4)

硬件		复位响应后 ^{註 1}
<R>	复位功能	不确定 ^{註 2}
	低电压检测器	低电压检测寄存器 (LVIM) 00H ^{註 3} 低电压检测电平选择寄存器 (LVIS) 0EH ^{註 2}
DMA 控制器	SFR 地址寄存器 0, 1 (DSA0, DSA1)	00H
	RAM 址寄存器 0L, 0H, 1L, 1H (DRA0L, DRA0H, DRA1L, DRA1H)	00H
	字节计数寄存器 0L, 0H, 1L, 1H (DBC0L, DBC0H, DBC1L, DBC1H)	00H
	模式控制寄存器 0, 1 (DMC0, DMC1)	00H
	操作控制寄存器 0, 1 (DRC0, DRC1)	00H
中断	请求标志寄存器 0L、0H、1L、1H、2L、2H (IF0L、IF0H、IF1L、IF1H、IF2L、IF2H)	00H
	屏蔽标志寄存器 0L、0H、1L、1H、2L、2H (MK0L、MK0H、MK1L、MK1H、MK2L、MK2H)	FFH
	优先级指定标志寄存器 00L、00H、01L、01H、02L、02H、10L、10H、11L、11H、12L、12H (PR00L、PR00H、PR01L、PR01H、PR10L、PR10H、PR11L、PR11H、PR02L、PR02H、PR12L、PR12H)	FFH
	外部中断上升沿使能寄存器 0 (EGP0)	00H
	外部中断下降沿使能寄存器 0 (EGN0)	00H
可编程增益放大器	可编程增益放大器控制寄存器(OAM)	00H
比较器	比较器 0 控制寄存器(C0CTL)	00H
	比较器 0 内部参考电压设置寄存器(C0RVM)	00H
	比较器 1 控制寄存器(C1CTL)	00H
	比较器 1 内部参考电压设置寄存器(C1RVM)	00H

- 注 1. 在复位信号产生或振荡稳定时间等待期间，硬件状态中只有 PC 内容不确定。复位后其它硬件状态保持不变。
2. 这些值依据复位源的不同而变化。

复位源		RESET 输入	由 POC 复位	由执行非法指令复位	由 WDT 复位	由 LVI 复位
RESF	TRAP 位	清零 (0)	清零 (0)	置位 (1)	保持	保持
	WDRF 位			保持	置位 (1)	保持
	LVIRF 位			保持	保持	置位 (1)
LVIS		清零 (0EH)	清零 (0EH)	清零 (0EH)	清零 (0EH)	保持

3. 此数值依据复位源和选项字节而不同。

17.1 确认复位源的寄存器

μPD79F9211 中有许多内部复位源。复位控制标志寄存器（RESF）用于存储产生复位请求的复位源。
可由 8 位存储器操作指令读取 RESF。

<R> RESET 输入、由上电清零电路(POC)电路复位、以及读取 RESF，可将 TRAP、WDRF 和 LVIRF 清零。

<R> 图 17-5. 复位控制标志寄存器（RESF）的格式

地址：FFFA8H 复位后：未定义 R

符号	7	6	5	4	3	2	1	0
RESF	TRAP ^{※1}	未定义	未定义	WDRF ^{※1}	未定义	未定义	未定义	LVIRF ^{※1}
TRAP	由执行非法指令引起内部复位请求 ^{※2}							
0	不产生内部复位请求，或对 RESF 清零。							
1	产生内部复位请求。							
WDRF	看门狗定时器产生的内部复位请求(WDT)							
0	不产生内部复位请求，或对 RESF 清零。							
1	产生内部复位请求。							
LVIRF	由低电压检测电路（LVI）产生的内部复位请求							
0	不产生内部复位请求，或对 RESF 清零。							
1	产生内部复位请求。							

- 注
- 1. 复位后的值依据复位源而变化。
 - 2. 当指令码 FFH 执行时产生非法指令。
由非法指令执行引起的复位，不能通过在线仿真器或片上调试仿真器终止。

注意事项 1. 不要用 1 位存储器操作指令读取数据。

- <R>
- 2. 不要仅仅基于 RESF 寄存器 8 位数据的读取值就作出判断，因为 TRAP、WDRF 和 LVIRF 之外的位都是不确定的。
 - 3. 当使用 LVI 默认开始功能 (000C1H 的第 0 位(LVIOFF) = 0) 时，依据上电波形，LVIRF 标志可能一开始就变成 1。

产生复位请求时的 RESF 状态如表 17-3 所示。

表 17-3. 产生复位请求时的 RESF 状态

复位源 标志	RESET 输入	由 POC 复位	由执行非法指令复位	由 WDT 复位	由 LVI 复位
TRAP	清零 (0)	清零 (0)	置位 (1)	保持	保持
WDRF			保持	置位 (1)	保持
LVIRF			保持	保持	置位 (1)

18.1 上电清零电路的功能

上电清零电路 (POC) 有以下功能。

- 在上电时产生内部复位信号。
当供电电压 (V_{DD}) 超过 $1.61\text{ V} \pm 0.09\text{ V}^{\text{註}}$ 时，释放复位信号。

注意事项 如果低电压检测器 (LVI) 由选项字节默认设置为 ON，那么复位信号直到供电电压 (V_{DD}) 超过 $2.07\text{ V} \pm 0.2\text{ V}^{\text{註}}$ 才释放。

- 比较供电电压 (V_{DD}) 和检测电压 ($V_{PDR} = 1.59\text{ V} \pm 0.09\text{ V}^{\text{註}}$)，并在 $V_{DD} < V_{PDR}$ 时产生内部复位信号。

注 这些均为初步数据，可能会改变。

<R>

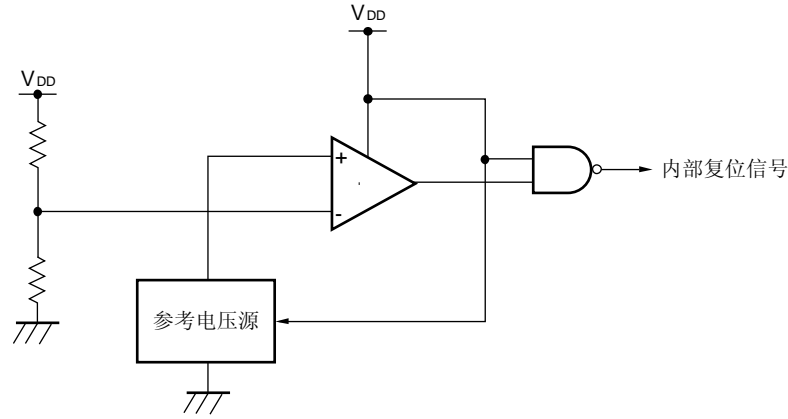
注意事项 如果在 POC 电路产生一个内部复位信号，则复位控制标志寄存器 (RESF) 的 TRAP、WDRF 和 LVIRF 被清零。

备注 此产品集成了多种可以产生内部复位信号的硬件功能。复位控制标志寄存器 (RESF) 中有一个标志用于指示复位源，用于看门狗定时器 (WDT)、低电压检测电路 (LVI)、或者执行非法指令产生内部复位信号时。当 WDT 或 LVI 产生内部复位信号时，不对 RESF 清零 (00H)，标志位设置为 1。
需要了解 RESF 的详细信息，可参见第 17 章 复位功能。

18.2 上电清零电路的配置

上电清零电路的框图如图 18-1 所示。

图 18-1. 上电清零电路的框图



18.3 上电清零电路的操作

- 在上电时产生一个内部复位信号。当供电电压 (V_{DD}) 超过检测电压 ($V_{PDR} = 1.61\text{ V} \pm 0.09\text{ V}^{\text{注}}$) 时，释放复位信号。

注意事项 如果低电压检测器 (LVI) 由选项字节默认设置为 ON，那么复位信号直到供电电压 (V_{DD}) 超过 $2.07\text{ V} \pm 0.2\text{ V}^{\text{注}}$ 才释放。

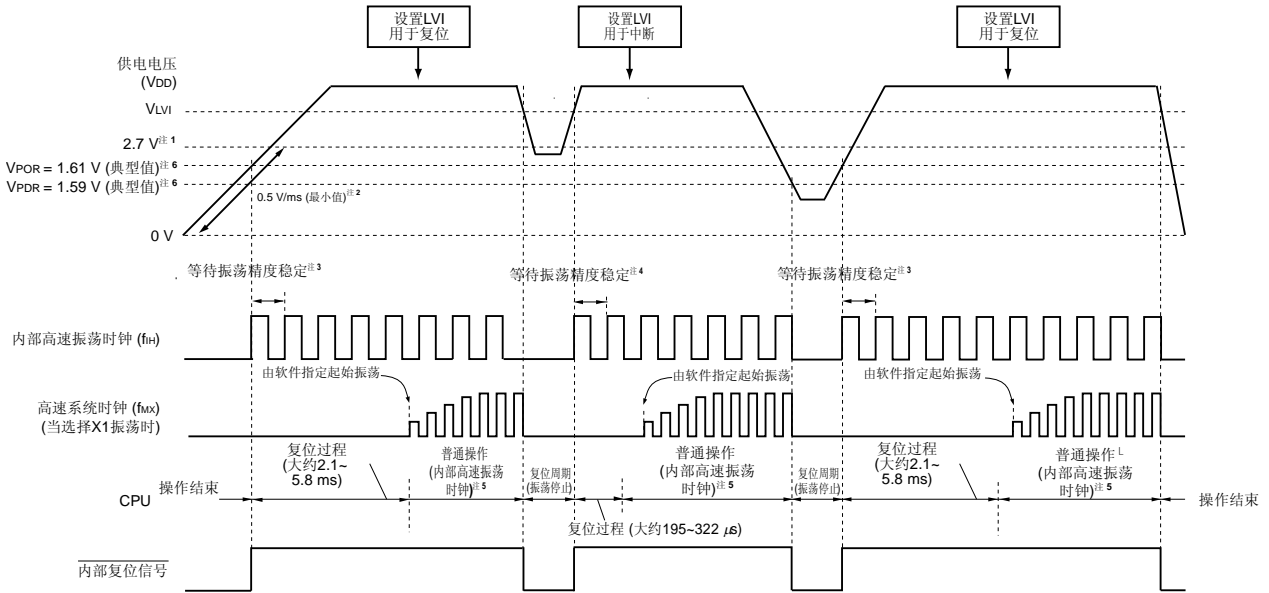
- 比较供电电压 (V_{DD}) 和检测电压 ($V_{PDR} = 1.59\text{ V} \pm 0.09\text{ V}^{\text{注}}$)。在 $V_{DD} < V_{PDR}$ 时，产生内部复位信号。

注 这些均为初步数据，可能会改变。

由上电清零电路和低电压检测电路产生的内部复位信号的时序如下所示。

图 18-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序 (1/2)

<R> (1) 电源应用时 LVI 为 OFF(选项字节: LVIOFF = 1)



- 注
1. 可靠操作的电压范围是 $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ 。确认电源电压变为 2.7 V 后能正常运行。为了在供电电压下降时，使低于 2.7 V 的状态为复位状态，可以使用低电压检测电路的复位功能，或通过 **RESET** 引脚输入低电平。
 2. 如果上电后电压升至 2.7 V 期间的速率低于 0.5 V/ms （最小值），在电压达到 2.7 V 之前向 **RESET** 引脚输入一个低电平，或者通过选项字节将 LVI 设置为 ON（选项字节: LVIOFF = 0）。
 3. 复位处理时间，比如等待内部电压稳定的时间，包括内部高速振荡时钟的振荡精确稳定时间。
 4. 内部复位处理时间包括内部高速振荡时钟的振荡精确稳定时间。
 5. 可以选择内部高速振荡时钟和高速系统时钟或副系统时钟作为 CPU 时钟。如果要使用 X1 时钟，可以通过 OSTC 确认是否经历了振荡稳定时间。而如果要使用 XT1 时钟，可以通过定时器功能来确认是否经历了振荡稳定时间。
 6. 这只是初步数据，可能会变化。

注意事项 复位状态释放后，通过软件设置低电压检测电路（参考第 19 章 低电压检测电路）。

备注

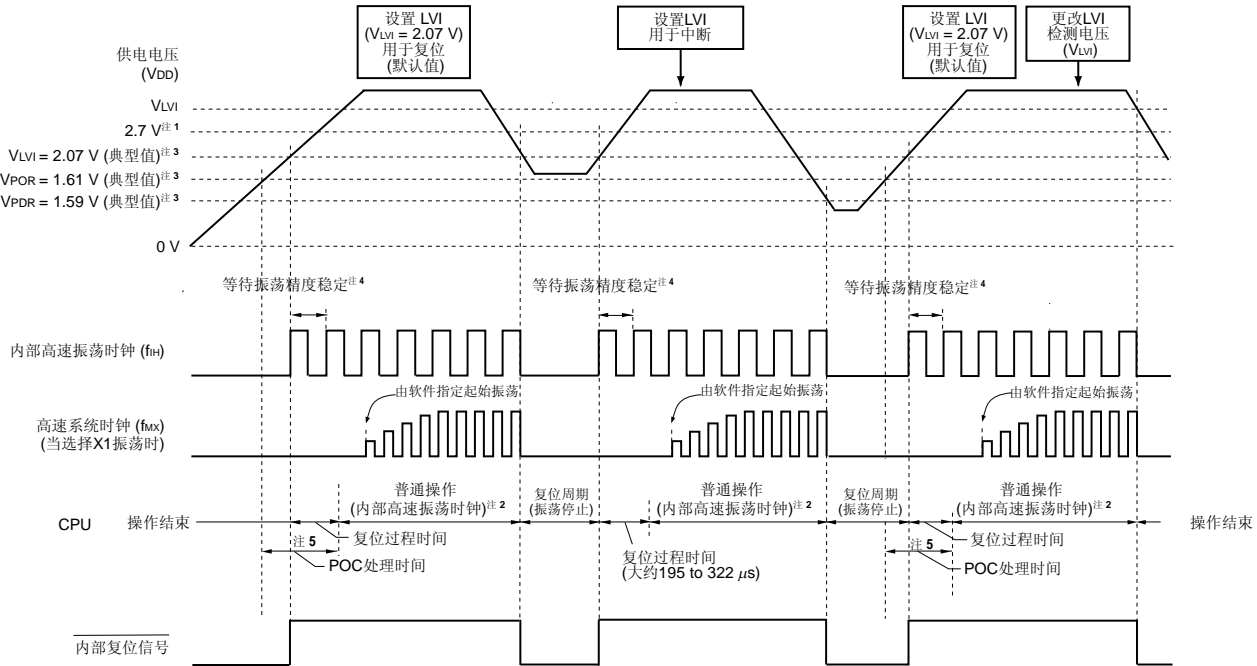
VLVI: LVI 检测电压

VPOR: POC 供电上升检测电压

VPDR: POC 供电下降检测电压

图 18-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序 (2/2)

<R> (2) 当上电之后 LVI 为 ON 时 (选项字节: LVIOFF = 0)



- 注
1. 可靠操作的电压范围是 $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ 。确认电源电压变为 2.7 V 后能正常运行。为了在供电电压下降时，使低于 2.7 V 的状态为复位状态，可以使用低电压检测电路的复位功能，或通过 $\overline{\text{RESET}}$ 引脚输入低电平。
 2. 可以选择内部高速振荡时钟和高速系统时钟或副系统时钟作为 CPU 时钟。如果要使用 X1 时钟，可以通过 OSTC 确认是否经历了振荡稳定时间。而如果要使用 XT1 时钟，可以通过定时器功能来确认是否经历了振荡稳定时间。
 3. 这些均为初步数据，可能会改变。
 4. 内部复位处理时间包括内部高速振荡时钟的振荡精确稳定时间。

<R>

5. 下面的时间是到达 POC 检测电压 (1.59 V (典型值)) 和开始正常操作之间需要的时间。
 - 当从 1.59 V (典型值) 达到 2.07 V (典型值) 的时间小于 5.8 ms :
在 1.59 V (典型值) 和启动正常操作之间需要 $2.2 \sim 6.0\text{ ms}$ 的 POC 处理时间。
 - 当从 1.59 V (典型值) 达到 2.07 V (典型值) 的时间大于 5.8 ms :
从 2.07 V (典型值) 到开始正常操作之间，需要 $125 \sim 153\text{ μs}$ 的复位处理时间。

注意事项 复位状态释放后，通过软件设置低电压检测电路 (参考第 19 章 低电压检测电路)。

备注

VLVI: LVI 检测电压

VPOR: POC 供电上升检测电压

VPDR: POC 供电下降检测电压

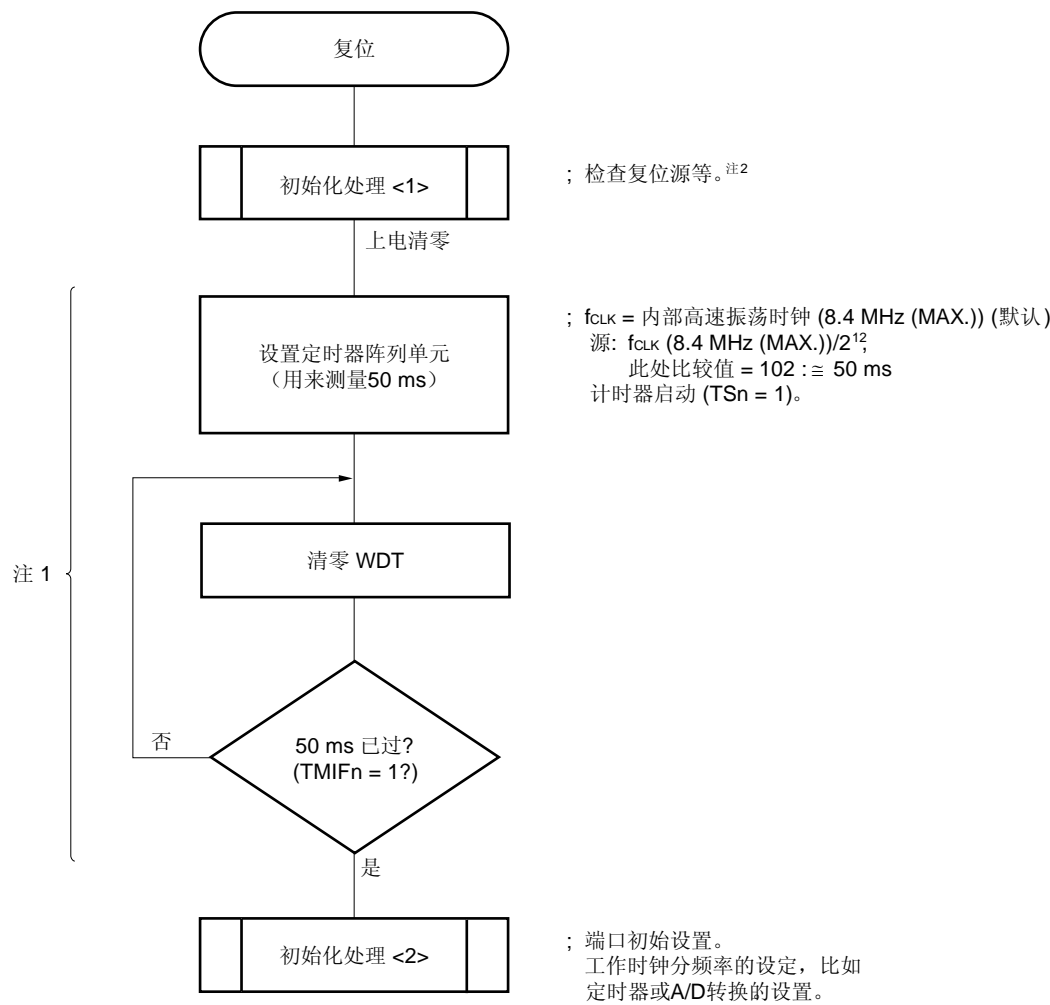
18.4 上电清零电路使用注意事项

在系统中，如果供电电压 (V_{DD}) 在接近 POC 检测电压 (V_{POR}, V_{PDR}) 时的一段时间内产生波动，则系统可能会反复复位和从复位状态释放。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动所需要的时间。

<方法>
在释放复位信号后，通过一个使用定时器的软件计数器，等待系统供电电压的波动期，然后对端口初始化。

图 18-3. 复位释放后软件处理过程示例 (1/2)

- 如果供电电压在接近 POC 检测电压时的波动时间为 50 ms 或更少

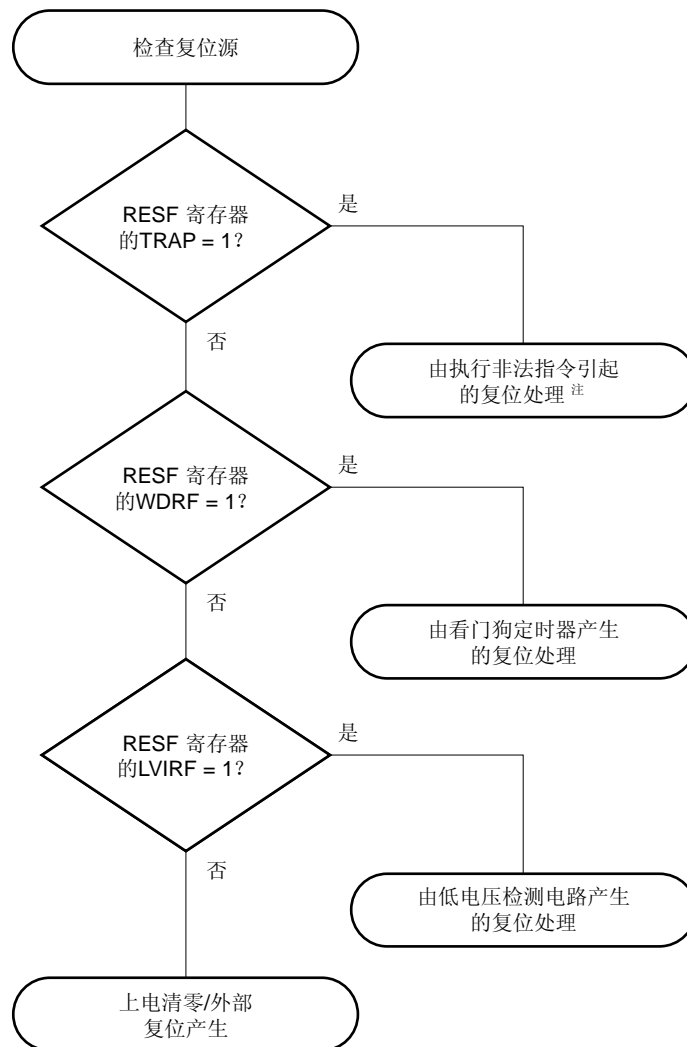


- 注 1. 如果在此期间再次产生复位，则不启动初始化操作<2>。
2. 流程图显示在下页。

备注 n = 00~11

图 18-3. 复位释放后软件处理过程示例 (2/2)

• 检查复位源



注 当执行 FFH 中的指令码时，产生非法操作。
由执行非法指令引起的复位不能通过在线仿真器或片上调试仿真器调试终止。
调试仿真器。

第 19 章 低电压检测电路

19.1 低电压检测电路的功能

低电压检测电路 (LVI) 有以下功能。

- LVI 电路比较供电电压 (V_{DD}) 和检测电压 (V_{LVI})，或比较从外部输入引脚 (EXLVI) 的输入电压和检测电压 (V_{EXLVI} = 1.21 V ±0.1 V[※])，并产生内部复位信号或内部中断信号。
- 低电压检测电路 (LVI) 可由选项字节默认设置为 ON。如果设置为 ON，供电电压从 POC 检测电压(V_{POR}=1.61V (典型值))或更低电压开始上电时，当供电电压 (V_{DD}) < 检测电压 (V_{LVI}=2.07V±0.2V[※]) 时，会产生内部复位信号。之后，当供电电压 (V_{DD}) < 检测电压 (V_{LVI}= 2.07 V ±0.1 V[※]) 时，产生内部复位信号。
- 可由软件选择检测供电电压 (V_{DD}) 或外部输入引脚的输入电压 (EXLVI)。
- 可由软件选择复位或中断操作。
- 可通过软件改变供电电压的检测电平 (V_{LVI}，16 电平)。
- 在 STOP 模式下可操作。

注 这只是初步数据，可能会变化。

依据软件设置，可产生以下复位和中断信号。

供电电压 (V _{DD}) 的电平检测选择 (LVISEL = 0)		从外部输入引脚(EXLVI) 的输入电压检测电平选择 (LVISEL = 1)	
选择复位 (LVIMD = 1). 当 V _{DD} < V _{LVI} 时，产生内部复位信号，并且当 V _{DD} ≥ V _{LVI} 时，释放复位信号。	选择中断 (LVIMD = 0). 当 V _{DD} 降低至低于 V _{LVI} (V _{DD} < V _{LVI}) 时，产生内部中断信号，或者当 V _{DD} 变为 V _{LVI} 或更高时 (V _{DD} ≥ V _{LVI})。	选择复位 (LVIMD = 1). 当 EXLVI < V _{EXLVI} 时，产生内部复位信号，并且当 EXLVI ≥ V _{EXLVI} 时，释放复位信号。	选择中断 (LVIMD = 0). 当 EXLVI 降低至低于 V _{EXLVI} (EXLVI < V _{EXLVI}) 时，产生内部中断信号，或者当 EXLVI 变为 V _{EXLVI} 或更高时 (EXLVI ≥ V _{EXLVI})。

备注 LVISEL: 低电压检测寄存器 (LVIM) 的第 2 位
LVIMD: LVIM 的第 1 位。

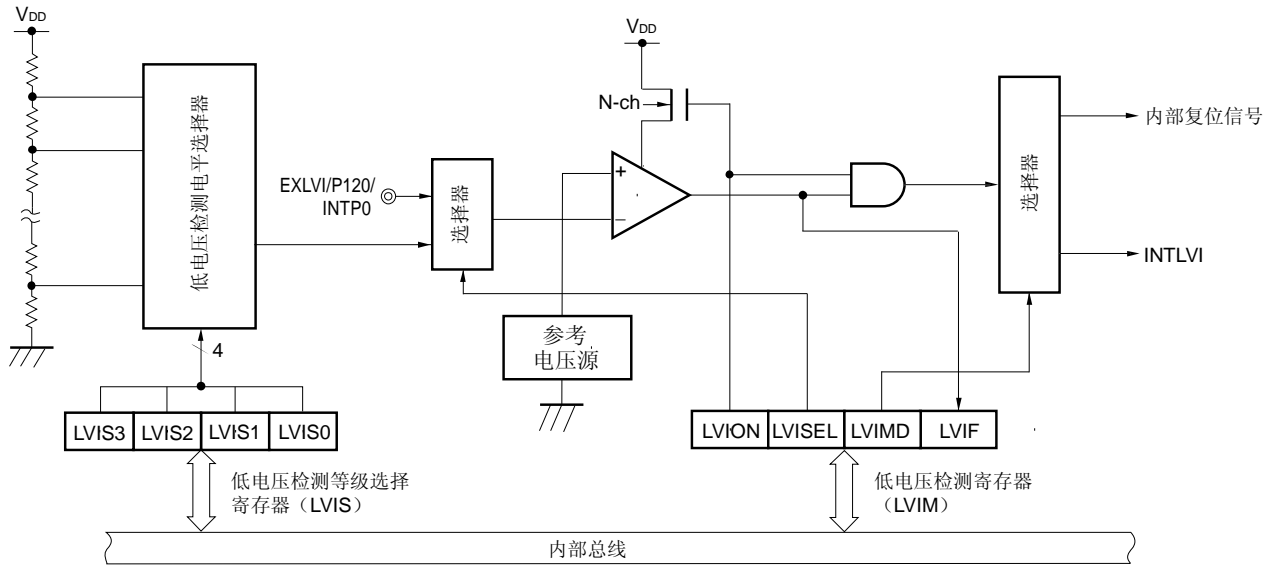
当低电压检测电路运行时，供电电压或者外部输入引脚的输入电压可以通过读取低电压检测标志（LVIM 的位 0: LVIF）来检测是否高于或低于检测电平。

当低电压检测电路用作产生复位时，如果复位产生，复位控制标志寄存器 (RESF) 的第 0 位 (LVIRF) 被置 1。需要了解 RESF 的详细信息，可参见第 17 章 复位功能。

19.2 低电压检测电路的配置

低电压检测电路的框图如图 19-1 所示。

图 19-1. 低电压检测电路的框图



19.3 控制低电压检测电路的寄存器

低电压检测电路由以下寄存器控制。

- 低电压检测寄存器 (LVIM)
- 低电压检测电平选择寄存器 (LVIS)
- 端口模式寄存器 12 (PM12)

(1) 低电压检测寄存器 (LVIM)

该寄存器用于设置低电压检测和操作模式。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位后这个寄存器的值为 00H。

图 19-2. 低电压检测寄存器 (LVIM) 的格式

地址: FFFA9H 复位后: 00H^{注1} R/W^{注2}

符号	<7>	6	5	4	3	<2>	<1>	<0>
LVIM	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION ^{注3,4}	允许低电压检测操作
0	禁止操作
1	允许操作

LVISEL ^{注3}	电压检测的选择
0	检测供电电压 (V_{DD}) 的电平
1	检测外部输入引脚的输入电压 ($EXLVI$) 的电平

LVIMD	低电压检测操作模式(中断/复位)的选择
0	<ul style="list-style-type: none"> LVISEL = 0: 当供电电压 V_{DD} 降至低于检测电压 V_{LVI} ($V_{DD} < V_{LVI}$) 时, 产生内部中断信号, 或者当 V_{DD} 变为 V_{LVI} 或更高时 ($V_{DD} \geq V_{LVI}$)。 LVISEL = 1: 当外部输入引脚($EXLVI$)的输入电压下降低于检测电压(V_{EXLVI}) ($EXLVI < V_{EXLVI}$)或当 $EXLVI$ 成为 V_{EXLVI} 或更高($EXLVI \geq V_{EXLVI}$)时将产生中断信号。
1	<ul style="list-style-type: none"> LVISEL = 0: 当供电电压 $V_{DD} <$ 检测电压 V_{LVI} 时, 产生内部复位信号, 并且当 $V_{DD} \geq V_{LVI}$ 时, 释放复位信号。 LVISEL = 1: 当从外部输入引脚来的输入电压 $EXLVI <$ 检测电压 V_{EXLVI} 时, 产生内部复位信号, 并且当 $EXLVI \geq V_{EXLVI}$ 时, 释放复位信号。

LVIF	低电压检测标
0	<ul style="list-style-type: none"> LVISEL = 0: 供电电压(V_{DD}) \geq 检测电压(V_{LVI})或禁止 LVI 操作时 LVISEL = 1: 外部输入引脚的输入电压($EXLVI$) \geq 检测电压(V_{EXLVI}), 或当禁止操作时
1	<ul style="list-style-type: none"> LVISEL = 0: 供电电压 (V_{DD}) $<$ 检测电压 (V_{LVI}) LVISEL = 1: 外部输入引脚的输入电压($EXLVI$) $<$ 检测电压(V_{EXLVI})

- 注
- 复位数值依据复位源和选项字节的设置而改变。
LVI 复位此寄存器不清零(00H)。
当除去 LVI 的复位信号应用时, 如果选项字节 LVIOFF = 0 则设置为“82H”, 如果选项字节 LVIOFF = 1 则设置为“00H”。
 - 第 0 位只读。
 - 在产生除 LVI 复位以外的复位时, LVION、LVIMD 和 LVISEL 被清零。LVI 复位时, 这些位不被清零。

注 4. 当 LVION 设置为 1 时，LVI 电路中比较器的操作就开始。在 LVION 设置为 1 和当用 LVIF 确认电压之间，使用软件进行下列时间段的等待。

- 运行稳定时间 (10 μs (最大值))
- 最小脉冲宽度 (200 μs (最小值))
- 检测延迟时间 (200 μs (最大值))

在此期间不管电压数值为何，LVIF 的数值可以设置/清零，因此就不能使用。同时，在此期间 LVIIF 中断请求标志设置为 1。

注意事项 1. 在执行以下任一过程后，停止 LVI。

- 当使用 8 位存储器操作指令：向 LVIM 写入 00H。
- 当使用 1 位存储器操作指令：对 LVION 清零。

2. 外部输入引脚的输入电压 (EXLVI) 必须满足 $\text{EXLVI} < V_{\text{DD}}$

3. 当 LVI 用于中断模式(LVIMD = 0)且 LVISEL 设为 0 时，将产生用于禁止 LVI 操作(清除 LVION)的中断请求信号(INTLVI)，此时供电电压(V_{DD})小于或等于检测电压(V_{LVI}) (若 LVISEL = 1，则外部输入引脚(EXLVI)的输入电压小于或等于检测电压(V_{EXLVI}))，并可能设置 LVIIF 至 1。

(2) 低电压检测电平选择寄存器 (LVIS)

该寄存器用于选择低电压检测电平。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号产生输入，设置此寄存器为 0EH。

图 19-3. 低电压检测电平选择寄存器 (LVIS) 的格式

地址: FFFAAH 复位后: 0EH^{注1} R/W

符号	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	检测电平
0	0	0	0	V _{LV10} (4.22 ±0.1 V) ^{注2}
0	0	0	1	V _{LV11} (4.07 ±0.1 V) ^{注2}
0	0	1	0	V _{LV12} (3.92 ±0.1 V) ^{注2}
0	0	1	1	V _{LV13} (3.76 ±0.1 V) ^{注2}
0	1	0	0	V _{LV14} (3.61 ±0.1 V) ^{注2}
0	1	0	1	V _{LV15} (3.45 ±0.1 V) ^{注2}
0	1	1	0	V _{LV16} (3.30 ±0.1 V) ^{注2}
0	1	1	1	V _{LV17} (3.15 ±0.1 V) ^{注2}
1	0	0	0	V _{LV18} (2.99 ±0.1 V) ^{注2}
1	0	0	1	V _{LV19} (2.84 ±0.1 V) ^{注2}
1	0	1	0	V _{LV110} (2.68 ±0.1 V) ^{注2}
1	0	1	1	V _{LV111} (2.53 ±0.1 V) ^{注2}
1	1	0	0	V _{LV112} (2.38 ±0.1 V) ^{注2}
1	1	0	1	V _{LV113} (2.22 ±0.1 V) ^{注2}
1	1	1	0	V _{LV114} (2.07 ±0.1 V) ^{注2}
1	1	1	1	V _{LV115} (1.91 ±0.1 V) ^{注2}

注 1. 复位数值依据复位源而改变。

如果 LVIS 寄存器被 LVI 设置，它不会复位而是保持当前值。若非 LVI 的复位有效，则该寄存器的值复位至“0EH”。

2. 这些均为初步数据，可能会改变。

注意事项 1. 确保将位 4~7 清“0”。

注意事项 2. 按照下列的方法之一可改变 LVIS 的数值。

- 当停止 LVI 之后改变数值
 - <1> 停止 LVI (LVION = 0)。
 - <2> 改变 LVIS 寄存器。
 - <3> 设置中断模式 (LVIMD = 0)。
 - <4> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <5> 允许 LVI 操作 (LVION = 1)。
 - <6> 在取消 LVI 中断屏蔽 (LVIMK = 0) 之前，用软件清零。因为当 LVI 操作允许时，会设置 LVIIIF 标志。
- 当设置用作中断模式 (LVIMD = 0) 后，改变数值
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 设置中断模式 (LVIMD = 0)。
 - <3> 改变 LVIS 寄存器。
 - <4> 在取消 LVI 中断屏蔽 (LVIMK = 0) 之前，用软件清零。因为当改变 LVIS 寄存器时，会设置 LVIIIF 标志。

3. 当检测到外部输入引脚的输入电压 (EXLVI) 时，检测电压 (V_{EXLVI}) 恒定。因此，不必设置 LVIS。

(3) 端口模式寄存器 12 (PM12)

当使用 P120/EXLVI/INTP0 引脚进行外部低电压检测电压输入时，设置 PM120=1。此时，P120 的输出锁存器的值可以为 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM12。

复位信号的产生将此寄存器设置为 FFH。

图 19-4. 端口模式寄存器 12 (PM12) 的格式

地址: FFF2CH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM12	1	1	1	1	1	1	1	PM120

PM120	P120 引脚 I/O 模式选择
0	输出模式（输出缓冲器开）
1	输入模式（输出缓冲器关）

19.4 低电压检测电路的操作

低电压检测电路能够用于以下两种模式。

(1) 用于复位 (LVIMD = 1)

- 如果 LVISEL = 0，比较供电电压 (V_{DD}) 与检测电压 (V_{LVI})，当 $V_{DD} < V_{LVI}$ 时产生内部复位信号，而当 $V_{DD} \geq V_{LVI}$ 时释放内部复位。
- 如果 LVISEL = 1，比较外部输入引脚的输入电压 ($EXLVI$) 和检测电压 (V_{EXLVI})，在 $EXLVI < V_{EXLVI}$ 时产生内部复位信号，而在 $EXLVI \geq V_{EXLVI}$ 时释放内部复位。

备注 低电压检测电路 (LVI) 可由选项字节默认设置为 ON。如果设置为 ON，供电电压从 POC 检测电压 ($V_{POR}=1.61V$ (典型值)) 或更低电压开始上电时，当供电电压 (V_{DD}) < 检测电压 ($V_{LVI}=2.07V \pm 0.2V^{(*)}$) 时，会产生内部复位信号。之后，当供电电压 (V_{DD}) < 检测电压 ($V_{LVI} = 2.07 V \pm 0.1 V^{(*)}$) 时，产生内部复位信号。

(2) 用作中断 (LVIMD = 0)

- 如果 LVISEL = 0，比较供电电压 (V_{DD}) 与检测电压 (V_{LVI})。当 V_{DD} 降至低于 V_{LVI} ($V_{DD} < V_{LVI}$) 时，或者当 V_{DD} 变为 V_{LVI} 或更高时 ($V_{DD} \geq V_{LVI}$)，产生中断信号 (INTLVI)。
- 如果 LVISEL = 1，比较外部输入引脚的输入电压 ($EXLVI$) 和检测电压 ($V_{EXLVI} = 1.21V \pm 0.1V^{(*)}$)。当 $EXLVI$ 降至低于 V_{EXLVI} ($EXLVI < V_{EXLVI}$) 时，或者当 $EXLVI$ 变为 V_{EXLVI} 或更高时 ($EXLVI \geq V_{EXLVI}$)，产生中断信号 (INTLVI)。

注 这只是初步数据，可能会变化。

当低电压检测电路运行时，供电电压或者外部输入引脚的输入电压可以通过读取低电压检测标志 (LVIM 的位 0: LVIF) 来检测是否高于或低于检测电平。

备注 LVIMD: 低电压检测寄存器 (LVIM) 的第 1 位
LVISEL: LVIM 的第 2 位。

19.4.1 当用于复位

(1) 当检测供电电压 (V_{DD}) 等级时

(a) 当 LVI 默认启动功能 (LVIOFF = 1) 设置停止时

- 当启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 把低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清零 (检测供电电压的电平 (V_{DD})) (缺省值)。
 - <3> 使用低电压检测电平选择寄存器 (LVIS) 的位 3~0 (LVIS3~LVIS0) 位设置检测电压。
 - <4> 设置 LVIM 的第 7 位 (LVION) = 1 (允许 LVI 的操作)。
 - <5> 使用软件等待下列时间段 (总计 $410\mu s$)。
 - 运行稳定时间 ($10\mu s$ (最大值))
 - 最小脉冲宽度 ($200\mu s$ (最小值))
 - 检测延迟时间 ($200\mu s$ (最大值))
 - <6> 等到 LVIM 的第 0 位 (LVIF) 检查出 (供电电压 (V_{DD}) \geq 检测电压 (V_{LVI}))
 - <7> 设置 LVIM 的第 1 位 (LVIMD) 为 1 (当检测到电平时, 产生复位)。

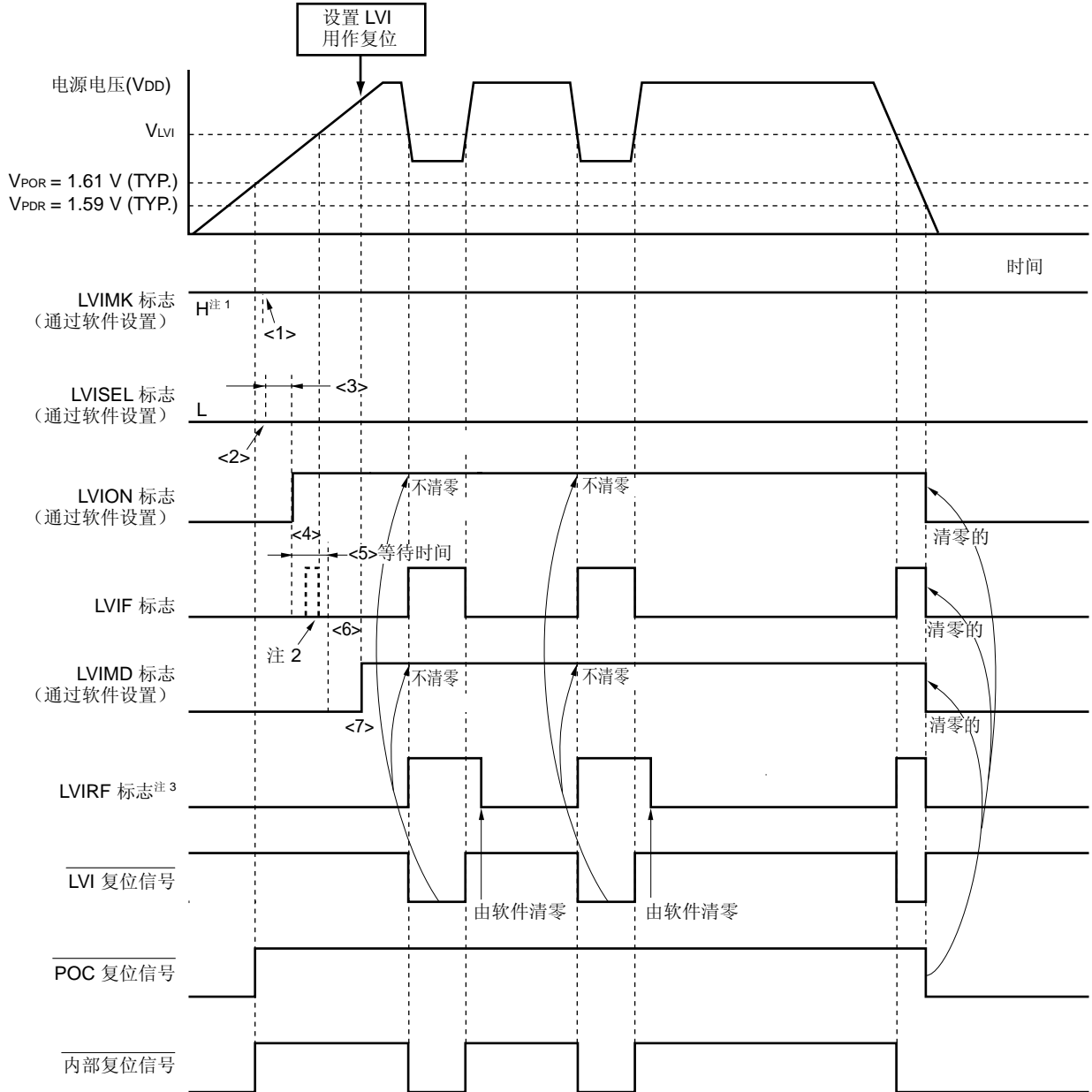
图 19-5 显示了由低电压检测电路产生的内部复位信号的时序。该时序图中的标号与上面的<1>~<7>对应。

注意事项 1. 务必执行<1>。当 LVIMK = 0 时, 可能在<4>之后立即发生中断。

2. 当 LVIMD=1 时, 如果供电电压 (V_{DD}) \geq 检测电压 (V_{LVI}), 则不会产生内部复位信号。

- 当停止操作时
 - 必须执行以下过程之一。
 - 当使用 8 位存储器操作指令:
 - 向 LVIM 写入 00H。
 - 当使用 1 位存储器操作指令:
 - 先对 LVIMD 清零, 再对 LVION 清零。

图 19-5. 低电压检测电路内部复位信号产生时序
(位: LVISEL = 0, 选项字节: LVIOFF = 1)



- 注
1. 通过复位信号发生设置 LVIMK 标志为“1”。
 2. 可以设置中断请求标志寄存器的 LVIIF 标志和 LVIF 标志为 (1)。
 3. LVIRF 为复位控制标志寄存器 (RESF) 的第 0 位。需要了解 RESF 的详细信息, 可参见第 17 章 复位功能。

- 备注
1. 上述图 19-5 中的<1> 至 <7>对应于 19.4.1 (1)(a) 当 LVI 默认启动功能 (LVIOFF = 1) 设置停止时中的“当启动操作”中描述的<1> 至 <7>。
 2. V_{POR}: POC 供电上升检测电压
V_{PDR}: POC 供电下降检测电压

(b) 当 LVI 默认启动功能设置允许时 (LVIOFF = 0)

- 当启动操作时

在下列初始设置状态下开启。

- 设置 LVIM 的第 7 位 (LVION) 为 1(允许 LVI 操作)零 0(检测供电电压的电平(V_{DD}))
- 设置低电压检测电平选择寄存器 (LVIS) 为 0EH(默认数值: $V_{LVI} = 2.07V \pm 0.1V$)
- 设置 LVIM 的第 1 位 (LVIMD) 为 1(当检测到电平时产生复位)
- 设置 LVIM 的位 0(LVIF)至 0 (“供电电压(V_{DD}) \geq 检测电压(V_{LVI})”)

图 19-6 显示了由低电压检测电路产生的内部复位信号的时序。

- 当停止操作时

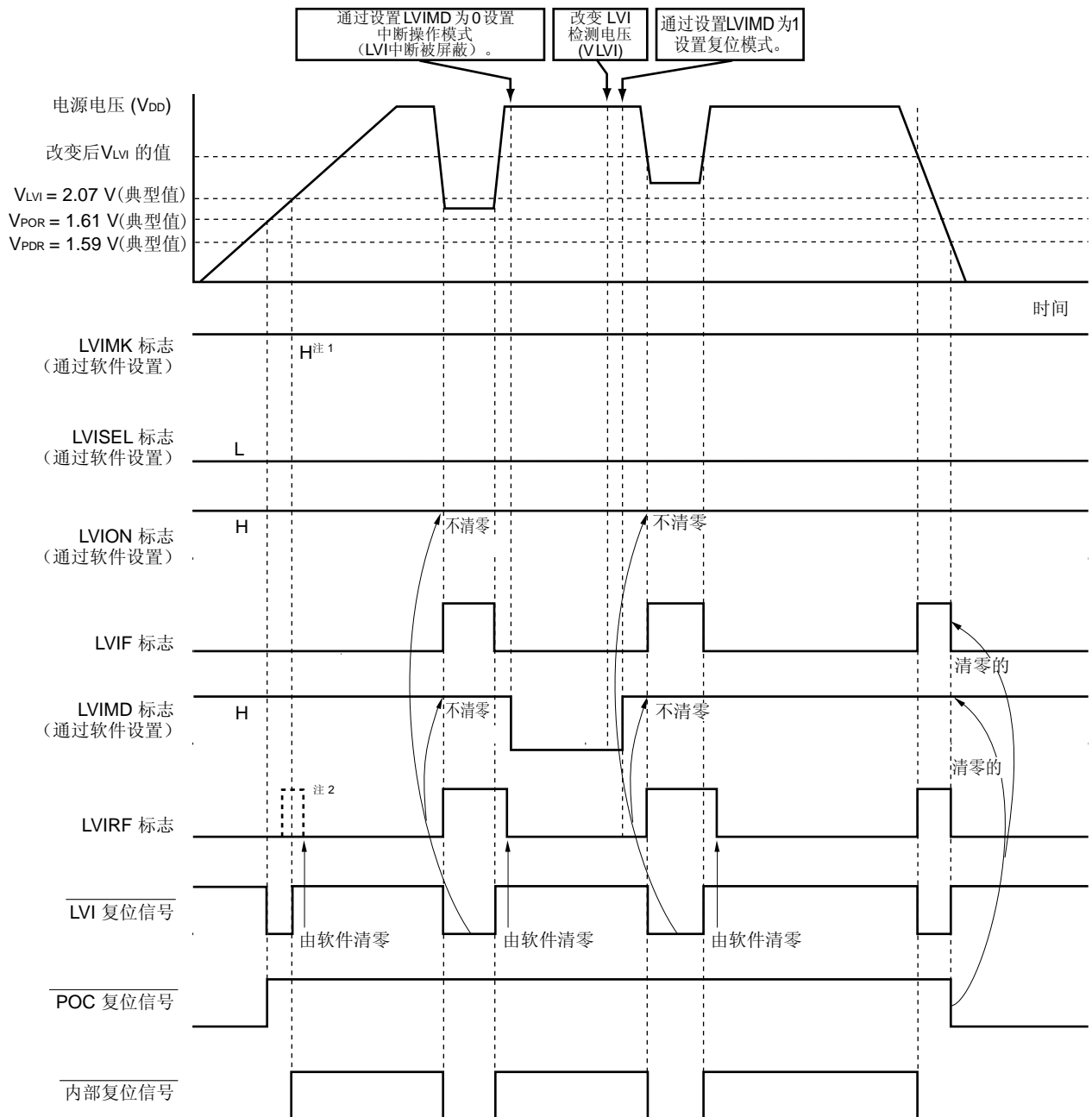
必须执行以下过程之一。

- 当使用 8 位存储器操作指令:
向 LVIM 写入 00H。
- 当使用 1 位存储器操作指令:
先对 LVIMD 清零, 再对 LVION 清零。

注意事项 当 LVI 默认开始功能使用时, 如果通过软件设置 LVI 操作禁止, 则操作如下:

- 当 LVION = 0 时不执行低电压检测。
- 如果当 LVION = 0 时发生复位, 在复位释放后 CPU 启动时 LVION 将被重置为 1。有一个阶段低电压检测不能正常执行, 但由于 WDT 和非法指令执行产生一个复位。
通过 LVI 进行脉冲宽度检测时最大必须不超过 200 μs 。复位发生时设置 LVION = 1, CPU 开始操作, 而不需要等待 LVI 稳定时间。

图 19-6. 低电压检测电路内部复位信号产生时序
(位: LVISEL = 0, 选项字节: LVIOFF = 0)



注 1. 通过复位信号发生设置 LVIMK 标志为“1”。

2. LVIRF 为复位控制标志寄存器 (RESF) 的第 0 位。

当使用 LVI 的默认开始功能 (000C1H 的第 0 位 (LVIOFF) = 0), LVIRF 标志可能会由于上电波形而从开始就变为 1。

需要了解 RESF 的详细信息, 可参见第 17 章 复位功能。

备注 V_{POR}: POC 供电上升检测电压

V_{PDR}: POC 供电下降检测电压

(2) 当检测外部输入引脚的输入电压 (EXLVI) 电平时

- 当启动操作时
 - <1> 屏蔽 LVI 中断 ($LVIMK = 1$)。
 - <2> 设置低电压检测寄存器 (LVIM) 的第 2 位 ($LVISEL = 1$) (检测外部输入引脚的输入电压 (EXLVI) 电平)。
 - <3> 设置 LVIM 的第 7 位 ($LVION = 1$) (允许 LVI 的操作)。
 - <4> 使用软件等待下列时间段 (总计 $410\mu s$)。
 - 运行稳定时间 ($10\mu s$ (最大值))
 - 最小脉冲宽度 ($200\mu s$ (最小值))
 - 检测延迟时间 ($200\mu s$ (最大值))
 - <5> 等到 LVIM 的第 0 位 ($LVIF$) 检测到 (外部输入引脚的输入电压 (EXLVI) \geq 检测电压 ($V_{EXLVI} = 1.21 V$ (典型值)))。
 - <6> 设置 LVIM 的第 1 位 ($LVIMD = 1$) (当检测到电平时产生复位信号)。

图 19-7 显示了由低电压检测电路产生的内部复位信号的时序。该时序图中的标号与上面的<1>~<6>对应。

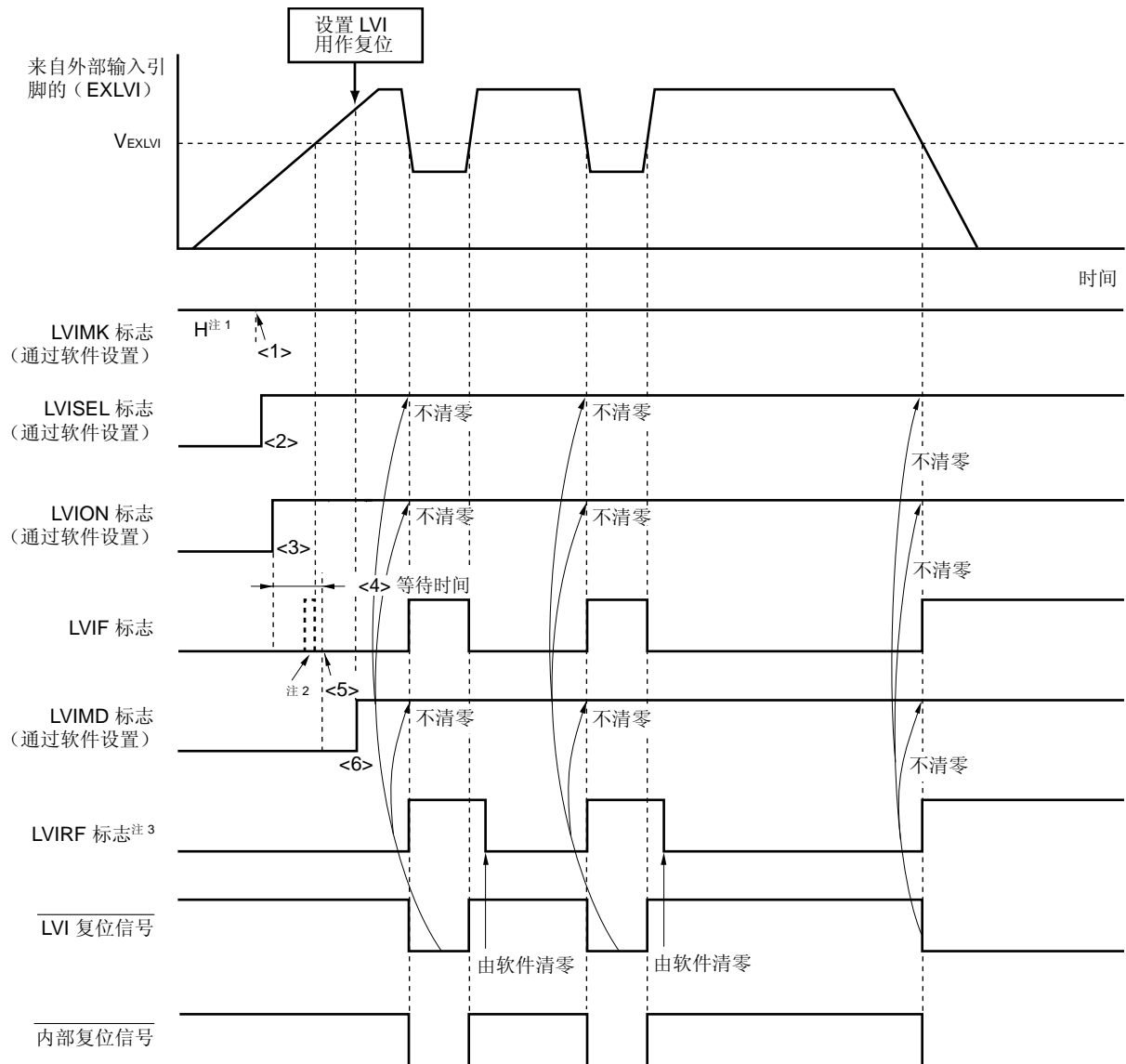
注意事项 1. 务必执行<1>。当 $LVIMK = 0$ 时，在完成<3>后可能立即会产生中断。

2. 如果在 $LVIMD=1$ 时，外部输入引脚的输入电压 (EXLVI) \geq 检测电压 ($V_{EXLVI} = 1.21 V$ (典型值))，则不会产生内部复位信号。

3. 外部输入引脚的输入电压 (EXLVI) 必须满足 $EXLVI < V_{DD}$

- 当停止操作时
 - 必须执行以下过程之一。
 - 当使用 8 位存储器操作指令：
 - 向 LVIM 写入 00H。
 - 当使用 1 位存储器操作指令：
 - 先对 LVIMD 清零，再对 LVION 清零。

图 19-7. 低电压检测电路内部复位信号产生时序
(位: LVISEL = 1)



- 注
1. 通过复位信号发生设置 LVIMK 标志为“1”。
 2. 可以设置中断请求标志寄存器的 LVIF 标志和 LVIF 标志为 (1)。
 3. LVIRF 为复位控制标志寄存器 (RESF) 的第 0 位。需要了解 RESF 的详细信息, 可参见第 17 章 复位功能。

备注 上述图 19-7 中的<1> 至 <6>对应于 19.4.1 (2)当检测外部输入引脚的输入电压 (EXLVI) 电平时中的“当启动操作”中描述的<1> 至 <6>。

19.4.2 用作中断时

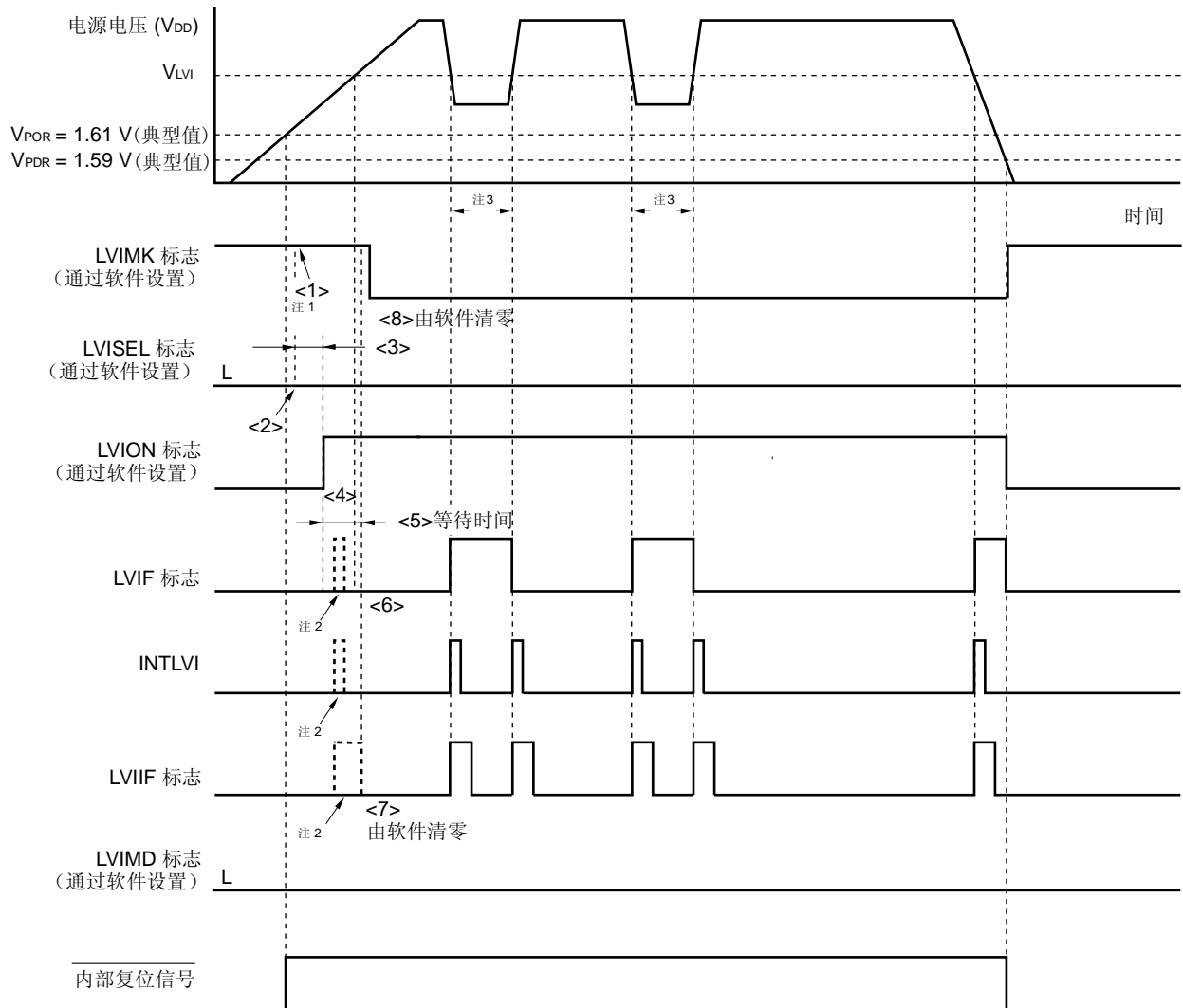
(1) 当检测供电电压 (V_{DD}) 等级时(a) 当 LVI 默认启动功能 ($LVIOFF = 1$) 设置停止时

- 当启动操作时
 - <1> 屏蔽 LVI 中断 ($LVIMK = 1$)。
 - <2> 将低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清零 (检测供电电压 (V_{DD}) 等级) (默认值)。
将 LVIM 的第 1 位 (LVIMD) 清零 (当检测到电平时产生中断信号) (默认值)。
 - <3> 使用低电压检测电平选择寄存器 (LVIS) 的第 3~0 ($LVIS3 \sim LVIS0$) 位设置检测电压。
 - <4> 设置 LVIM 的第 7 位 ($LVION$) = 1 (允许 LVI 的操作)。
 - <5> 使用软件等待下列时间段 (总计 $410\mu s$)。
 - 运行稳定时间 ($10\mu s$ (最大值))
 - 最小脉冲宽度 ($200\mu s$ (最小值))
 - 检测延迟时间 ($200\mu s$ (最大值))
 - <6> 在 LVIM 的位 0 ($LVIF$) 处, 当检测 V_{DD} 的下降沿时, 确认“供电电压 (V_{DD}) \geq 检测电压 (V_{LVI})”, 或当检测 V_{DD} 的上升沿时, 确认“供电电压 (V_{DD}) $<$ 检测电压 (V_{LVI})”。
 - <7> 将 LVI 的中断请求标志 ($LVIIF$) 清零。
 - <8> 释放 LVI 的中断屏蔽标志 ($LVIMK$)。
 - <9> 执行 EI 指令 (当使用向量中断时)。

图 19-8 显示了由低电压检测电路产生的中断信号的时序。该时序图中的标号与上面的<1>~<8>对应。

- 当停止操作时
必须执行以下过程之一。
 - 当使用 8 位存储器操作指令:
向 LVIM 写入 00H。
 - 当使用 1 位存储器操作指令:
对 $LVION$ 清零。

图 19-8. 低电压检测电路中断信号产生时序
(位: LVISEL = 0, 选项字节: LVIOFF = 1)



- 注
1. 通过复位信号发生设置 LVIMK 标志为“1”。
 2. 产生中断请求信号 (INTLVI)，且可以设置 LVIF 和 LVIIF 标志为 (1)。
 3. 若禁止 LVI 操作时，当供电电压 (V_{DD}) 小于或等于检测电压 (V_{LVI}) 时，将产生中断请求信号 (INTLVI) 并可能设置 LVIIF 为 1。

- 备注
1. 上述图 19-8 中的 <1> 至 <8> 对应于 19.4.2 (1)(a) 当 LVI 默认启动功能 (LVIOFF = 1) 设置停止时的“当启动操作”中描述的 <1> 至 <8>。
 2. V_{POR} : POC 供电上升检测电压
 V_{PDR} : POC 供电下降检测电压

(b) 当 LVI 默认启动功能设置允许时 (LVIOFF = 0)

- 当启动操作时

<1> 在下列初始设置状态下开启。

- 设置 LVIM 的第 7 位 (LVION) 为 1(允许 LVI 操作)
- 把低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) 清零 0(检测供电电压的电平 (V_{DD}))
- 设置低电压检测电平选择寄存器 (LVIS) 为 0EH(默认数值: V_{LVI} = 2.07V ± 0.1V)
- 设置 LVIM 的第 1 位 (LVIMD) 为 1(当检测到电平时产生复位)
- 设置 LVIM 的位 0(LVIF)至 0 (检测下降沿“供电电压(V_{DD}) ≥ 检测电压(V_{LVI})”)

<2> 将 LVIM 的位 1 (LVIMD) 清零(当检测到电平时产生中断信号)(默认值)。

<3> 释放 LVI 的中断屏蔽标志 (LVIMK) 。

<4> 执行 EI 指令(当使用向量中断时)。

图 19-9 显示了由低电压检测电路产生的中断信号的时序。该时序图中的标号与上面的<1>~<3>对应。

- 当停止操作时

必须执行以下过程之一。

- 当使用 8 位存储器操作指令:

向 LVIM 写入 00H。

- 当使用 1 位存储器操作指令:

对 LVION 清零。

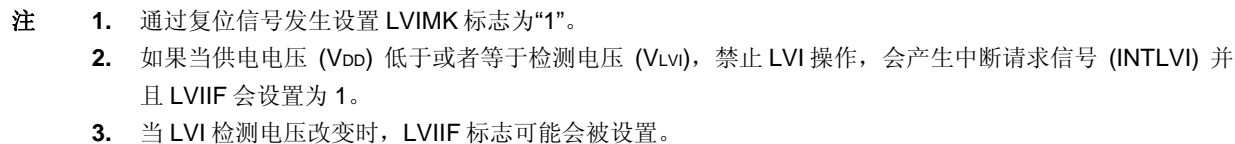
注意事项 1. 当 LVI 默认开始功能使用时, 如果通过软件设置 LVI 操作禁止, 则操作如下:

- 当 LVION = 0 时不执行低电压检测。
- 如果当 LVION = 0 时发生复位, 在复位释放后 CPU 启动时 LVION 将被重置为 1。有一个阶段低电压检测不能正常执行, 但由于 WDT 和非法指令执行产生一个复位。
通过 LVI 进行脉冲宽度检测时最大必须不超过 200 μs。复位发生时设置 LVION = 1, CPU 开始操作, 而不需要等待 LVI 稳定时间。

2. 当使用 LVI 的默认开始功能 (000C1H 的第 0 位 (LVIOFF) = 0), LVIRF 标志可能会由于上电波形而从开始就变为 1。

需要了解 RESF 的详细信息, 可参见第 17 章 复位功能。

(位: LVISEL = 0, 选项字节: LVIOFF = 0)



2. V_{POR} : POC 供电上升检测电压
 V_{PDR} : POC 供电下降检测电压

(2) 当检测外部输入引脚的输入电压 (EXLVI) 电平时

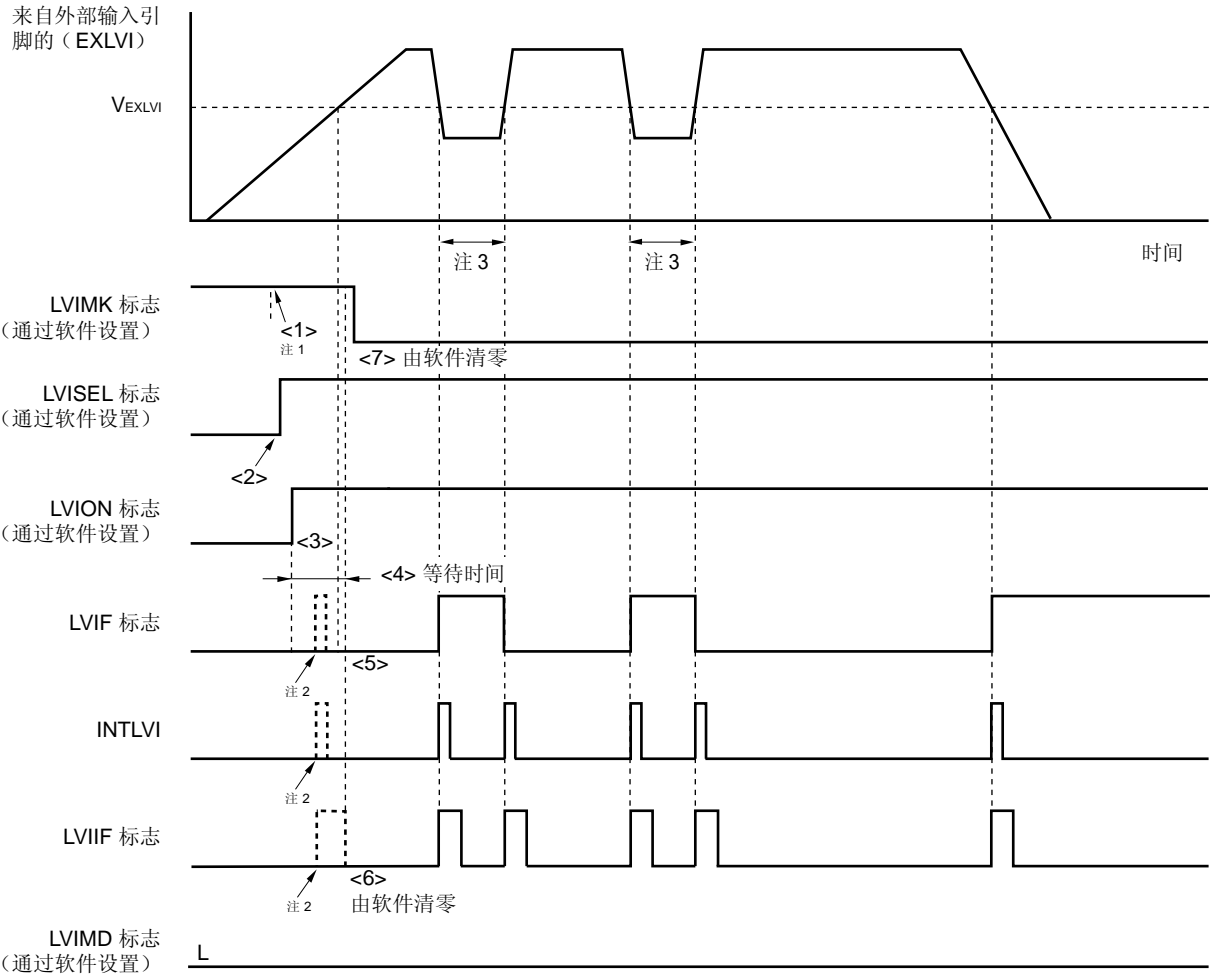
- 当启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 设置低电压检测寄存器 (LVIM) 的第 2 位 (LVISEL) = 1(检测外部输入引脚的输入电压 (EXLVI) 电平)。将 LVIM 的第 1 位 (LVIMD) 清零(当检测到电平时产生中断信号)(默认值)。
 - <3> 设置 LVIM 的第 7 位 (LVION) = 1(允许 LVI 的操作)。
 - <4> 使用软件等待下列时间段(总计 410 μ s)。
 - 运行稳定时间 (10 μ s (最大值))
 - 最小脉冲宽度 (200 μ s (最小值))
 - 检测延迟时间 (200 μ s (最大值))
 - <5> 在 LVIM 的位 0(LVIF)处, 当检测 EXLVI 的下降沿时, 确认“来自外部输入引脚的输入电压(EXLVI) \geq 检测电压($V_{EXLVI} = 1.21$ V (典型值))”, 或当检测 EXLVI 的上升沿时, 确认“来自外部输入引脚的输入电压 (EXLVI) < 检测电压 ($V_{EXLVI} = 1.21$ V (典型值))”。
 - <6> 将 LVI 的中断请求标志 (LVIF)清零。
 - <7> 释放 LVI 的中断屏蔽标志 (LVIMK) 。
 - <8> 执行 EI 指令(当使用向量中断时)。

图 19-10 显示了由低电压检测电路产生的中断信号的时序。该时序图中的标号与上面的<1>~<7>对应。

注意事项 外部输入引脚的输入电压 (EXLVI) 必须满足 $EXLVI < V_{DD}$

- 当停止操作时
必须执行以下过程之一。
- 当使用 8 位存储器操作指令:
向 LVIM 写入 00H。
- 当使用 1 位存储器操作指令:
对 LVION 清零。

图 19-10. 低电压检测电路中断信号产生时序
(位: LVISEL = 1)



- 注
1. 通过复位信号发生设置 LVIMK 标志为“1”。
 2. 产生中断请求信号 (INTLVI)，且可以设置 LVIF 和 LVIIF 标志为 (1)。
 3. 若禁止 LVI 操作时，当来自外部输入引脚的输入电压(EXLVI)小于或等于检测电压(V_{EXLVI})时，将产生中断请求信号(INTLVI)并可能设置 LVIIF 为 1。

备注 上述图 19-10 中的<1> 至 <7>对应于 19.4.2 (2)当检测外部输入引脚的输入电压 (EXLVI) 电平时中的“当启动操作”中描述的<1> 至 <7>。

19.5 低电压检测电路的注意事项

(1) 当供电电压 (V_{DD}) 在接近 LVI 检测电压 (V_{LVI}) 周围而频繁波动时采取的方法

在系统中，如果供电电压 (V_{DD}) 在接近 LVI 检测电压 (V_{LVI}) 的一段时间内产生波动，这时根据低电压检测电路的使用情况进行如下操作。

操作示例 1：用作复位时

系统可能会反复进行复位和释放复位。

在这种情况下，可采用下面的方法任意设置从复位解除到微控制器的启动所经历的时间。

<方法>

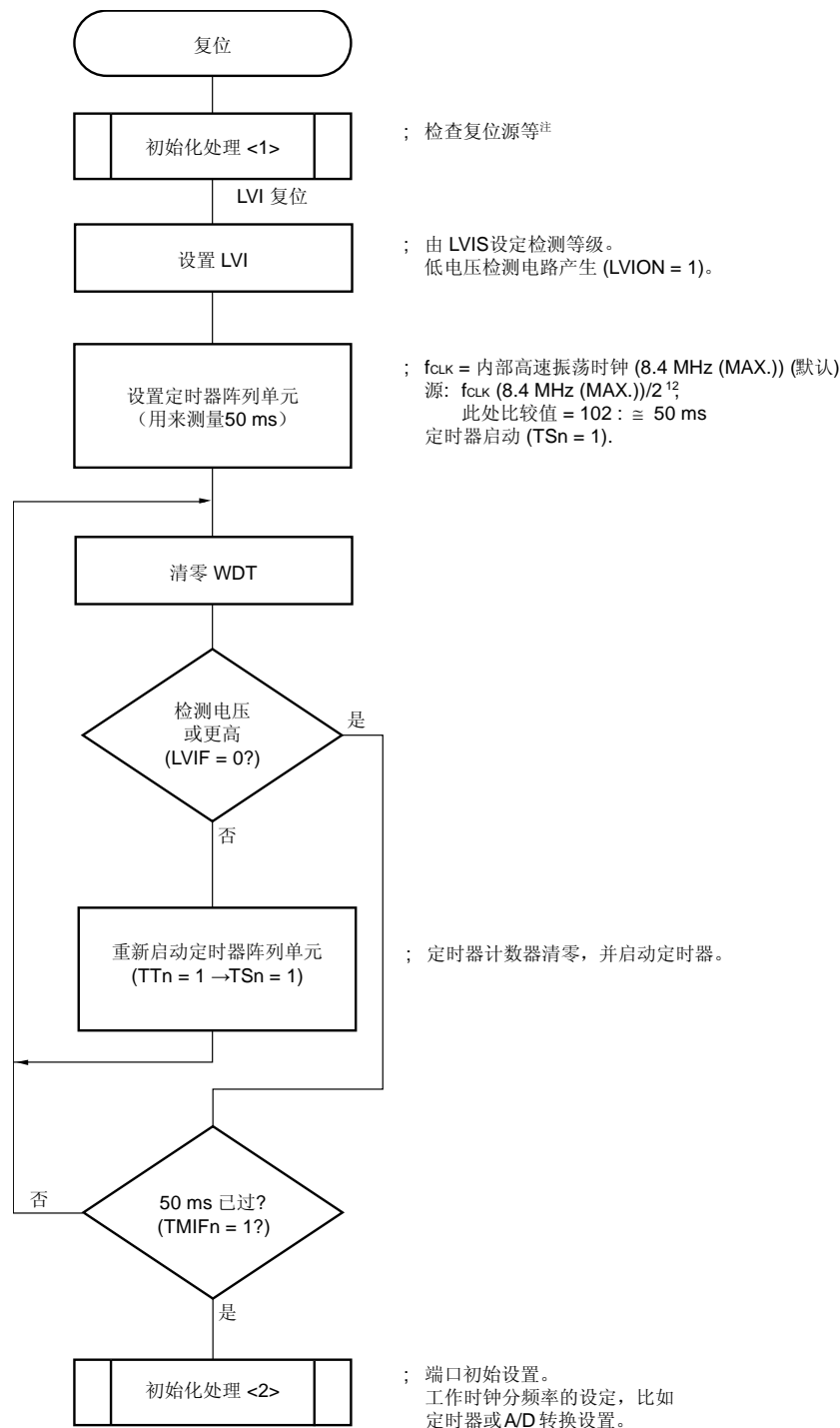
在释放复位信号后，通过软件计数器（使用一个定时器）等待系统的供电电压的波动期，然后对端口初始化（参见图 19-11）。

备注 如果低电压检测寄存器(LVIM)的位 2(LVISEL)设为“1”，上述意思改变如下。

- 电源电压 (V_{DD}) → 来自外部引脚的输入电压 ($EXLVI$)
- 检测电压 (V_{LVI}) → 检测电压 ($V_{EXLVI} = 1.21V$)

图 19-11. 复位释放后软件处理过程示例 (1/2)

- 如果供电电压在接近 LVI 检测电压时的波动时间为 50 ms 或更少

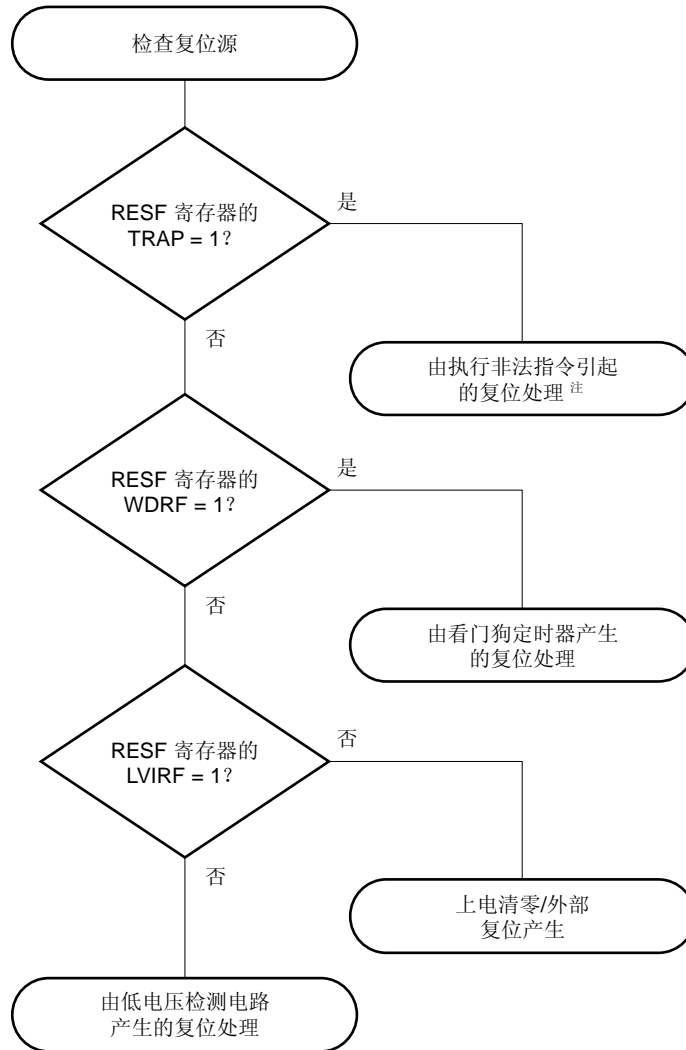


注 流程图显示在下页

- 备注
1. 如果低电压检测寄存器(LVIM)的位 2(LVISEL)设为“1”，上述意思改变如下。
 - 电源电压 (V_{DD}) → 来自外部引脚的输入电压 (EXLVI)
 - 检测电压 (V_{LVI}) → 检测电压 (V_{EXLVI} = 1.21V)
 2. n = 00~11

图 19-11. 复位释放后软件处理过程示例 (2/2)

• 检查复位源



注 当指令码 FFH 执行时产生非法指令。
由执行非法指令引起的复位不能通过在线仿真器或片上调试仿真器调试终止。
调试仿真器。

备注 如果低电压检测寄存器(LVIM)的位 2(LVISEL)设为“1”，上述意思改变如下。

- 电源电压 (VDD) → 来自外部引脚的输入电压 (EXLVI)
- 检测电压 (VLVI) → 检测电压 (VEXLVI = 1.21V)

操作示例 2：用作中断时

系统可能会频繁产生中断。

可采取下列方法。

<方法>

通过使用低电压检测寄存器(LVIM)的位 0(LVIF)进行 LVI 中断例程时，当检测 V_{DD} 的下降沿时，确认“供电电压(V_{DD}) \geq 检测电压(V_{LVI})”，或当检测 V_{DD} 的上升沿时，确认“供电电压 (V_{DD}) $<$ 检测电压 (V_{LVI})”。将中断请求标志寄存器 0L (IF0L) 的第 1 位 (LVIIF) 清零。

对于供电电压在 LVI 检测电压附近比较长时间的波动的系统，在等待供电电压波动时间之后采取以上方法。

备注 如果低电压检测寄存器(LVIM)的位 2(LVISEL)设为“1”，上述意思改变如下。

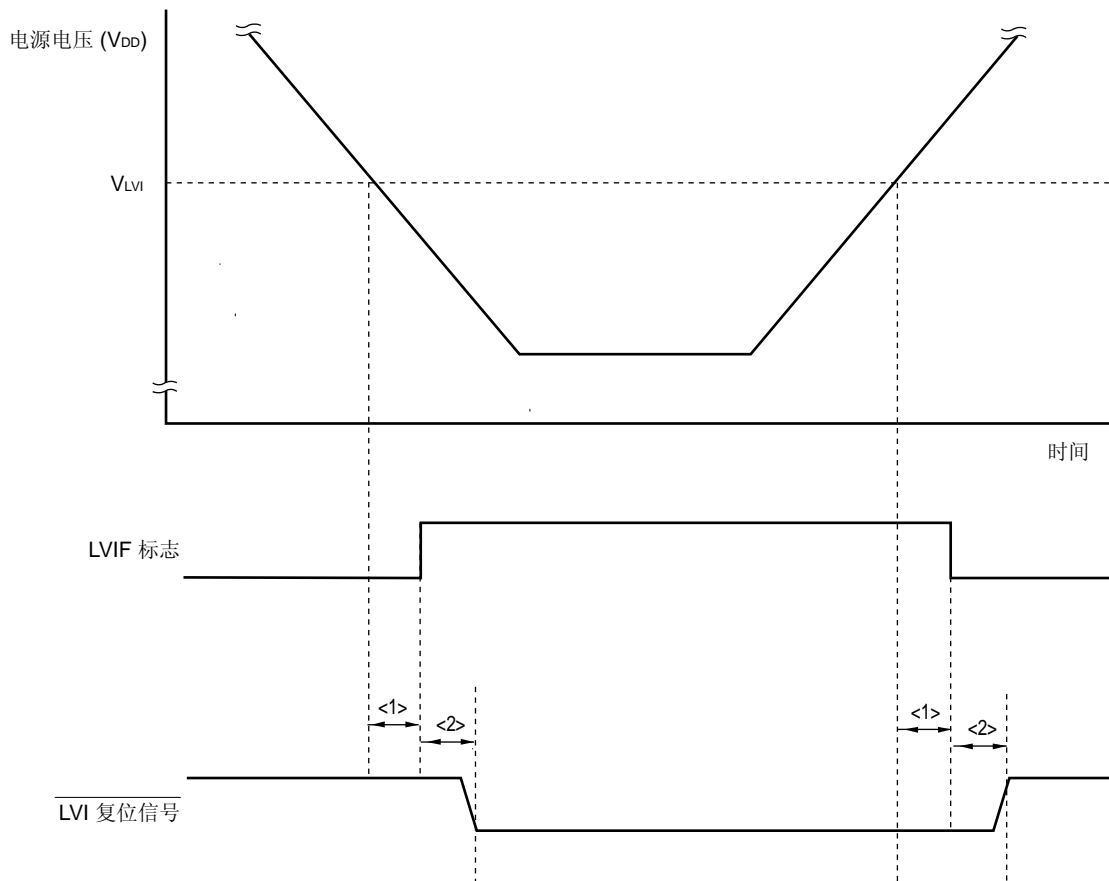
- 电源电压 (V_{DD}) \rightarrow 来自外部引脚的输入电压 (EX_{LVI})
- 检测电压 (V_{LVI}) \rightarrow 检测电压 ($V_{EX_{LVI}} = 1.21V$)

(2) 从 LVI 复位源产生时开始直到 LVI 复位产生或释放时的延时

从供电电压 (V_{DD}) $<$ LVI 检测电压 (V_{LVI}) 时到 LVI 复位产生时，会有一些延时。

同样，从 LVI 检测电压 (V_{LVI}) \leq 供电电压 (V_{DD}) 到 LVI 复位释放时，也会有一些延时（见图 19-12）。

图 19-12. 从 LVI 复位源产生时开始到 LVI 复位产生或释放时的延时



<1>: 最小脉冲宽度 (200 μs (最小值))

<2>: 检测延迟时间 (200 μs (最大值))

20.1 稳压器概述

μPD79F9211 包含一个在恒电压下操作设备的电路。此时，为了稳定稳压器的输出电压，把 REGC 引脚通过一个电容（0.47 至 1μF：目标值）连接到 Vss 上。然而，当使用的 STOP 模式从内部高速振荡时钟和外部主系统时钟运行起已经进入时，推荐使用 0.47 mF。同时，使用性能良好的电容器，因为它用来稳定内部电压的。

稳压器的输出电压正常为 2.4V（典型值），在低功耗电流模式下为 1.8V（典型值）。

20.2 控制稳压器的寄存器

(1) 稳压器模式控制寄存器 (RMC)

此寄存器设置稳压器的输出电压。

RMC 可用 8 位存储器操作指令设置。

复位输入设置此寄存器为 00H。

图 22-1. 稳压器模式控制寄存器 (RMC) 的格式

地址: F00F4H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
RMC								

RMC[7:0]	控制稳压器的输出电压
5AH	固定为低功耗电流模式(1.8 V)
00H	根据条件切换正常模式(2.4 V) 和低功耗电流模式(1.8 V) (参考 表 22-1)
其它值	设置禁止

注意事项 1. RMC 寄存器只有在低功耗电流模式下才可以重写(参考表 22-1)。换句话说, 当高速系统时钟(f_{MX})、高速内部振荡时钟(f_{IH})和 40 MHz 内部高速振荡时钟(f_{DSC})都停止时, CPU 操作使用副系统时钟(f_{XT})期间, 重写此寄存器。

2. 当设置固定为低功耗电流模式时，RMC 寄存器可在下列情况下使用。

<R> < CPU 时钟选择 X1 时钟时> $f_x \leq 5 \text{ MHz}$ 并且 $f_{CLK} \leq 1 \text{ MHz}$

<R> <当选择高速内部振荡时钟、外部输入时钟或副系统时钟 作为 CPU 时钟时> $f_{CLK} \leq 1 \text{ MHz}$

3. 在低功耗电流模式下，自编程功能禁止。

表 22-1. 稳压器输出电压条件

	模式	输出电压	条件
<R>	低功耗电流模式	1.8 V	<u>RESET</u> 引脚复位中
			STOP 模式下(OCD 模式除外)
			CPU 操作使用副系统时钟 (fxt) 期间, 高速系统时钟 (fmx)、高速内部振荡时钟 (fin) 和 40 MHz 内部高速振荡时钟 (fdsc) 都停止时。
			HALT 模式下设置为 CPU 操作使用副系统时钟 (fxt) 期间, 当高速系统时钟 (fmx)、高速内部振荡时钟 (fin) 和 40 MHz 内部高速振荡时钟 (fdsc) 都停止时。
	普通电流模式	2.4 V	其它值

第 21 章 选项字节

21.1 选项字节的功能

μPD79F9211 的 flash 存储器的地址 000C0H~000C3H 组成选项字节区域。

选项字节由用户选项字节 (000C0H~000C2H) 和片上调试选项字节 (000C3H)组成。

当打开电源或从复位状态重启设备时，设备自动参考选项字节，并设置指定的功能。使用该产品时，必须使用选项字节设置以下几项功能。

在自编程期间使用引导交换功能时，将 000C0H~000C3H 切换到 010C0H~010C3H。因此，事先将 000C0H~000C3H 和 010C0H~010C3H 设置为相同的值。

注意事项 必须将 FFH 设置为 000C2H (000C2H/ 010C2H，当使用引导交换操作时)。

21.1.1 用户选项字节 (000C0H~ 000C2H/ 010C0H ~010C2H)

(1) 000C0H/ 010C0H

- 看门狗定时器的操作
 - 在 HALT 或 STOP 模式下操作停止或允许。
- 看门狗定时器的间隔时间的设置
- 看门狗定时器的操作
 - 操作停止或允许。
- 看门狗定时器的窗口打开期间的设置
- 看门狗定时器的间隔中断的设置
 - 使用的或未使用的

注意事项 由于在引导交换时 000C0H 要被 010C0H 替换，因此要对 010C0H 设置与 000C0H 相同的值。

(2) 000C1H/010C1H

- 复位释放时的 LVI 设置（上电时）
 - LVI 由复位释放默认开或关（仅由 $\overline{\text{RESET}}$ 引脚复位，不包括 LVI、POC、WDT、或非法指令）。

注意事项 由于在引导交换时 000C1H 要被 010C1H 替换，因此要对 010C1H 设置与 000C1H 相同的值。

(3) 000C2H/ 010C2H

- 必须设置 FFH，因为这些地址为保留区域。

注意事项 当使用引导交换操作时，设置 FFH 到 010C2H，因为 000C2H 要被 010C2H 替换。

21.1.2 片上调试选项字节 (000C3H/ 010C3H)

- 片上调试操作的控制
 - 片上调试操作禁止或允许。
- 当片上调试安全 ID 验证失败时，flash 存储器数据的处理
 - 如果片上调试安全 ID 验证失败，flash 存储器的数据被擦除或不擦除。

注意事项 由于在引导交换时 000C3H 要被 010C3H 替换，因此要对 010C3H 设置与 000C3H 相同的值。

21.2 用户选项字节的格式

用户选项字节的格式如下所示。

图 21-1. 用户选项字节 (000C0H/010C0H) 的格式(1/2)

地址: 000C0H/010C0H^{※1}

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON
WDTINIT	看门狗定时器的间隔中断的使用						
0	间隔中断未使用。						
1	当达到溢出时间的 75%时，间隔中断产生。						
WINDOW1	WINDOW0	看门狗定时器窗口打开周期 ^{※2}					
0	0	25%					
0	1	50%					
1	0	75%					
1	1	100%					
WDTON	看门狗定时器计数器的操作控制						
0	计数器操作禁止（复位后计数停止）						
1	计数器操作允许（复位后计数开始）						
WDCS2	WDCS1	WDCS0	看门狗定时器溢出时间 (f _{IL} = 33 kHz (最大.))				
0	0	0	2 ⁷ /f _{IL} (3.88 ms)				
0	0	1	2 ⁸ /f _{IL} (7.76 ms)				
0	1	0	2 ⁹ /f _{IL} (15.52 ms)				
0	1	1	2 ¹⁰ /f _{IL} (31.03 ms)				
1	0	0	2 ¹² /f _{IL} (124.12 ms)				
1	0	1	2 ¹⁴ /f _{IL} (496.48 ms)				
1	1	0	2 ¹⁵ /f _{IL} (992.97 ms)				
1	1	1	2 ¹⁷ /f _{IL} (3971.88 ms)				

<R>

图 21-1. 用户选项字节 (000C0H/010C0H) 的格式(2/2)

地址: 000C0H/010C0H^{注1}

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON
WDSTBYON	看门狗定时器计数器的操作控制 (HALT/STOP 模式)						
0	计数器操作在 HALT/STOP 模式下停止 ^{注2}						
1	计数器操作在 HALT/STOP 模式下允许						

注 1. 由于在引导交换时 000C0H 要被 010C0H 替换, 因此要对 010C0H 设置与 000C0H 相同的值。

2. 当 WDSTBYON = 0 时, 窗口打开周期是 100%, 不论 WINDOW1 和 WINDOW0 为何值。

注意事项 看门狗定时器在 Flash 存储器自编程和 EEPROM 仿真期间继续操作。处理期间, 中断响应时间被延迟。设置溢出时间和窗口大小时应考虑延迟。

备注 f_{IL}: 内部低速振荡时钟频率

图 21-2. 选项字节 (000C1H/010C1H)的格式

地址: 000C1H/010C1H^注

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	LVIOFF
LVIOFF	LVI 在电源应用时的设置						
0	复位释放 (上电时) 时, LVI 默认打开 (允许 LVI 默认启动功能)						
1	复位释放 (上电时) 时, LVI 默认关闭 (停止 LVI 默认启动功能)						

注 由于在引导交换时 000C1H 要被 010C1H 替换, 因此要对 010C1H 设置与 000C1H 相同的值。

注意事项 1. 确保设置位 7~1 为“1”。

2. 当 LVI 默认开始功能使用时, 如果通过软件设置 LVI 操作禁止, 则操作如下:

- 当 LVION = 0 时不执行低电压检测。
- 如果当 LVION = 0 时发生复位, 在复位释放后 CPU 启动时 LVION 将被重置为 1。有一个阶段低电压检测不能正常执行, 但由于 WDT 和非法指令执行产生一个复位。
通过 LVI 进行脉冲宽度检测时最大必须不超过 200 μ s。复位发生时设置 LVION = 1, CPU 开始操作, 而不需要等待 LVI 稳定时间。

图 21-3. 选项字节 (000C2H/010C2H) 的格式

地址: 000C2H/010C2H[※]

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

注 必须将 FFH 设置为 000C2H，因为这些地址在保留区域。当使用引导交换操作时也要设置 FFH 为 010C2 因为 000C2H 要被 010C2H 替换。

21.3 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 21-4. 片上调试选项字节 (000C3H/010C3H) 的格式

地址: 000C3H/010C3H[※]

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	片上调试操作的控制
0	0	禁止片上调试操作。
0	1	设置禁止
1	0	如果允许片上调试和验证片上调试安全 ID 失败，擦除 flash 存储器的数据。
1	1	如果允许片上调试和验证片上调试安全 ID 失败，不擦除 flash 存储器的数据。

注 由于在引导交换时 000C3H 要被 010C3H 替换，因此要对 010C3H 设置与 000C3H 相同的值。

注意事项 第 7 位和第 0 位（OCDENSET 和 OCDERSD）只能被指定一个值。
000010B 的第 6 位必须设置为 1。

备注 当使用片上调试功能时，第 3 位～第 1 位的值被写入，设置后其值变得不确定。
但是，设置时必须将第 3 位～第 1 位设置为默认值（0、1、和 0）。

<R> 21.4 选项字节的设置

选项字节设置的软件描述示例如下所示。

OPT	CSEG	OPT_BYTE
	DB	10H ; 不要使用看门狗定时器的间隔中断, ; 允许看门狗定时器操作, ; 看门狗定时器的窗口打开周期是 25%, ; 看门狗定时器的溢出时间是 $2^{10}/f_{IL}$, ; HALT/STOP 模式期间, 停止看门狗定时器的操作
	DB	0FFH ; 停止 LVI 默认启动功能
	DB	0FFH ; 保留区域
	DB	85H ; 允许片上调试操作, 不要擦除 Flash 存储器。 ; 当安全 ID 认证失败时的数据

在自编程期间, 使用引导交换功能时, 将 000C0H~000C3H 切换到 010C0H~010C3H。因此, 描述至 010C0H~010C3H, 设置为如下与 000C0H~000C3H 相同的值。

OPT2	CSEG	AT	010C0H
	DB		10H ; 不要使用看门狗定时器的间隔中断, ; 允许看门狗定时器操作, ; 看门狗定时器的窗口打开周期是 25%, ; 看门狗定时器的溢出时间是 $2^{10}/f_{IL}$, ; HALT/STOP 模式期间, 停止看门狗定时器的操作
	DB		0FFH ; 停止 LVI 默认启动功能
	DB		0FFH ; 保留区域
	DB		85H ; 允许片上调试操作, 不要擦除 Flash 存储器。 ; 当安全 ID 认证失败时的数据

注意 要通过汇编语言指定选项字节, 应使用 OPT_BYTE 作为 CSEG 伪指令的重定位属性名。要为使用引导交换功能而在 010C0H~010C3H 设置选项字节, 应使用重定位属性 AT 来指定一个绝对地址。

备注 用户选项字节和片上调试选项字节, 也可以通过 RA78K0R 或 PM+的链接器选项来设置。当这样做时, 通过链接器选项设置的内容优先, 即使源中有如上文提到的描述存在。
关于如何设置链接器选项, 参见 RA78K0R 汇编程序包用户手册。

第 22 章 FLASH 存储器

μPD79F9211 包含 Flash 存储器，可以安装在电路板上实现在线写入、擦除和覆盖操作。

22.1 用Flash存储器编程器写入

通过使用专用 Flash 存储器编程器，数据可以 on-board 或 off-board 写入到 Flash 存储器。

(1) On-board 编程

当μPD79F9211 系列产品安装到目标系统上以后，Flash 存储器的内容可以被重写。必须将连接专用 Flash 编程器的连接器安装在目标系统中。

(2) Off-board 编程

在μPD79F9211 系列产品安装到目标系统之前，可使用专用程序适配器（FA 系列）将数据写入 Flash 存储器。

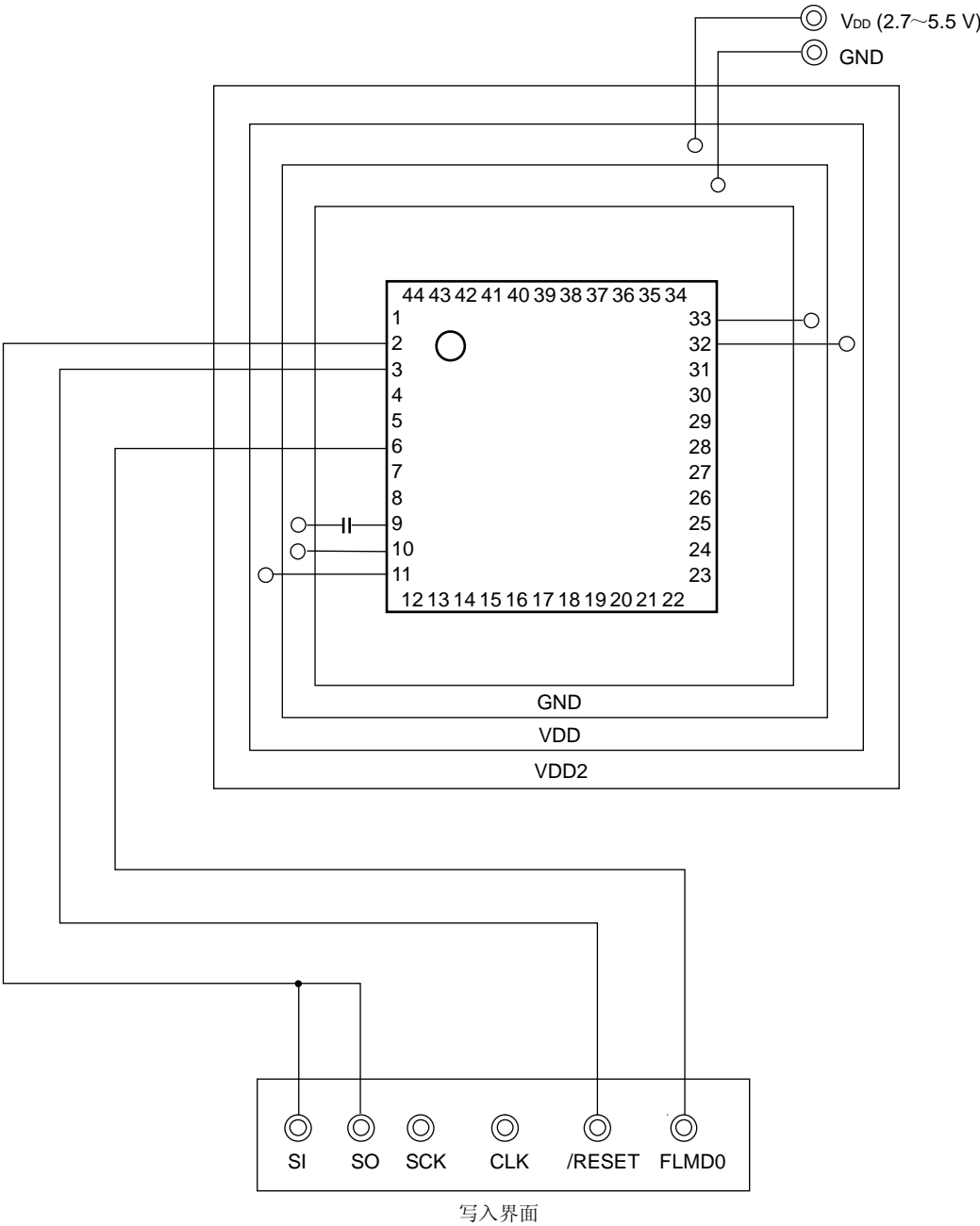
备注 FA 系列是 Naito Densei Machida Mfg. Co., Ltd 公司的产品。.

表 22-1. μPD79F9211 和专用 Flash 存储器编程器之间的写操作

专用 Flash 存储器编程器的引脚配置			44-引脚 GB 封装	
信号名称	I/O	引脚功能	引脚名称	引脚编号
SI/RxD	输入	接收信号	TOOL0/P40	2
SO/TxD	输出	传送信号		
SCK	输出	传送时钟	—	—
CLK	输出	时钟 输出	—	—
/RESET	输出	复位信号	$\overline{\text{RESET}}$	3
FLMD0	输出	模式信号	FLMD0	6
V _{DD}	I/O	V _{DD} 电压产生/ 电压监测	V _{DD}	11
			AV _{REF}	32
GND	—	地	V _{SS}	10
			AV _{SS}	33

在使用适配器对 Flash 存储器进行写操作时，可采用如下所示的连接方式。

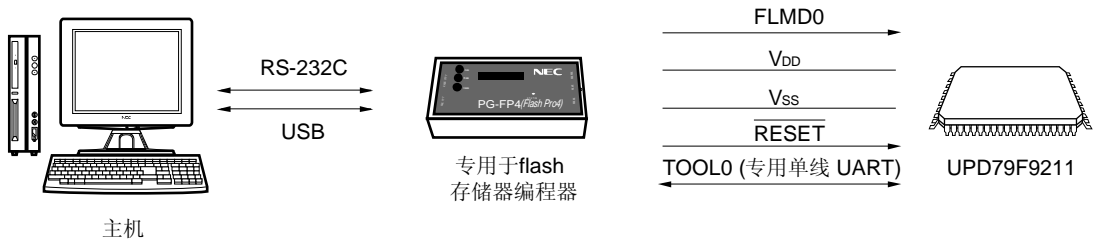
图 22-1. 使用适配器对 Flash 存储器进行写操作示例(GB 封装)



22.2 编程环境

以下是 μ PD79F9211 Flash 存储器写入所需的编程环境。

图 22-2. 向 Flash 存储器写入的环境



需要有一个控制专用 Flash 编程器的主机。

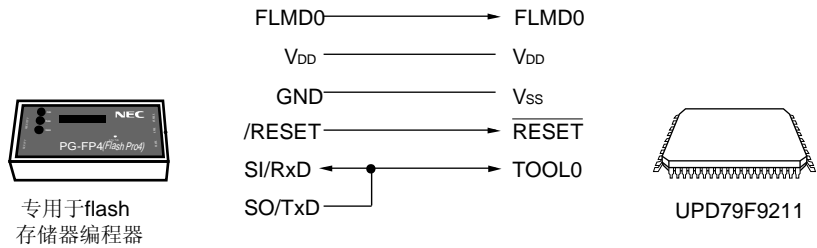
专用 Flash 存储器编程器与 μ PD79F9211 的接口 TOOL0 引脚通过专用单线 UART 进行写入和擦除等操作。Flash 存储器 off-board 写操作时必须使用一个专用编程适配器（FA 系列）。

22.3 通信模式

专用 Flash 存储器编程器和 μ PD79F9211 之间的通信由串行通信连接，使用 TOOL0 引脚通过 μ PD79F9211 的专用单线 UART 进行通讯。

传送速率：115,200 bps～ 1,000,000 bps

图 22-3. 专用 Flash 存储器编程器的通信



当使用 FlashPro4 作为专用 Flash 存储器编程器时，FlashPro4 给μPD79F9211 产生如下信号。详细情况，参考 FlashPro4 的用户手册。

表 22-2. 引脚连接

FlashPro4			μPD79F9211	连接
信号名称	I/O	引脚功能	引脚名称	
FLMD0	输出	模式信号	FLMD0	⊙
V _{DD}	I/O	V _{DD} 电压产生/电源监测	V _{DD} , AV _{REF}	⊙
GND	—	地	V _{SS} , AV _{SS}	⊙
CLK	输出	时钟 输出	—	×
/RESET	输出	复位信号	RESET	⊙
SI/RxD	输入	接收信号	TOOL0	⊙
SO/TxD	输出	传送信号		
SCK	输出	传送时钟	—	×

备注 ⊙: 确保连接引脚。
×: 引脚不必连接。

22.4 On Board方式的引脚连接

对 Flash 存储器进行 on-board 写入操作时，目标系统必须有连接专用 Flash 编程器的连接器。电路板上首先要提供一个选择功能，可以选择正常操作模式或 Flash 存储器编程模式。

当设置 Flash 存储器编程模式时，那些不用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此如果外部设备不能立即识别复位后的状态，则必须采用如下方式连接引脚。

22.4.1 FLMD0 引脚

(1) 在 Flash 存储器编程模式下

当通过 Flash 存储器编程器写入数据时，将此引脚直接连接到 Flash 存储器编程器。这提供了一个 V_{DD} 电平的写入电压到 FLMD0 引脚。
FLMD0 引脚不必由外部进行下拉，因为复位时会被内部下拉。要外部下拉，使用 1kΩ到 200kΩ的电阻。

(2) 在普通操作模式下

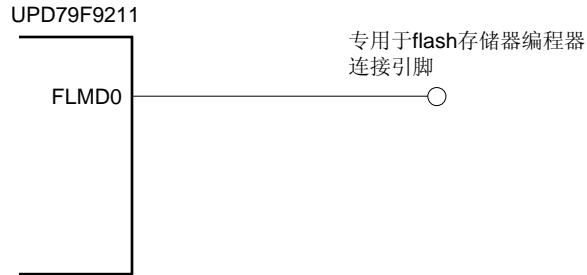
普通操作期间，建议此引脚开路。
复位释放前 FLMD0 引脚必须保持 V_{SS} 电位，但是不必由外部进行下拉，因为复位时会被内部下拉。但是，必须保持选择下拉（如 FLMDPUP = “0”，缺省值），通过使用背景事件控制寄存器（BECTL）中的位 7（FLMDPUP）（见 22.5（1）背景事件控制寄存器）。要处于外部下拉状态，需要使用 200kΩ或更小值的电阻。
自编程功能和使用编程器对 Flash 存储器进行重写的功能可以通过硬件来禁止，将该引脚直接连接到 V_{SS} 引脚即可。

(3) 在自编程模式下

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

在自编程模式下，设置在自编程库切换到上拉状态。

图 22-4. FLMD0 引脚连接示例

**22.4.2 TOOL0 引脚**

在 Flash 存储器编程模式下，该引脚直接连接到专用 Flash 存储器编程器，或通过外部电阻连接到 V_{DD} 进行上拉。

当在普通操作模式下允许片上调试功能时，经外部电阻连接到 V_{DD} 对该引脚进行上拉，必须在复位释放前确保输入 V_{DD} 到 TOOL0 引脚（禁止下拉此引脚）。

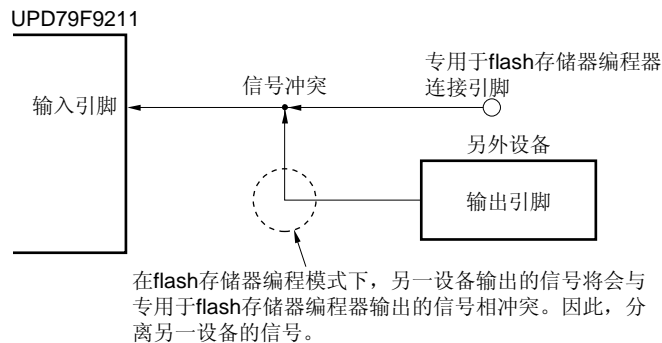
备注 因为使用了单线 UART、SAU 引脚不能用于 μPD79F9211 和专用 Flash 存储器编程器之间的通信。

22.4.3 RESET 引脚

如果将专用 Flash 编程器的复位信号连接到 $\overline{\text{RESET}}$ 引脚（已连接到板上的复位信号发生器），则会产生信号冲突。为了避免这种情况，应隔离与复位信号发生器的连接。

在 Flash 存储器编程模式下，如果从用户系统输入复位信号，则不能对 Flash 存储器进行正确编程。因此除了专用 Flash 编程器的复位信号外，不要输入其它信号。

图 22-5. 信号冲突($\overline{\text{RESET}}$ 引脚)



22.4.4 端口引脚

当设置 Flash 存储器编程模式时，那些不用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此，如果外部设备（与端口相连）不能立即识别复位后的状态，则必须通过一个电阻将端口引脚连接到 VDD 或 VSS。

22.4.5 REGC 引脚

与普通操作期间相同，通过一个电容（0.47 至 1μF：推荐值）连接 REGC 引脚到 GND。然而，当使用的 STOP 模式从内部高速振荡时钟和外部主系统时钟运行起已经进入的话，推荐使用 0.47μF。同时，使用性能良好的电容器，因为它是用来稳定内部电压的。

22.4.6 X1 和X2 引脚

以与普通操作模式相同的状态连接 X1 和 X2。

备注 在 Flash 存储器编程模式下，使用内部高速振荡时钟 (f_{IH})。

22.4.7 供电电压

要使用 Flash 编程器输出的供电电压，则将 VDD 引脚与 Flash 编程器的 VDD 相连，将 VSS 引脚与 Flash 编程器的 GND 相连。

要使用 on-board 供电电压，要按正常操作模式连接。

然而，当 on-board 供电电压使用时，如果要使用 Flash 编程器的电压监视功能，必须将 VDD 和 VSS 引脚分别与 Flash 编程器的 VDD 和 GND 相连。

其他的供电电压(AVREF 和 AVSS)那与正常模式中的相同 。

22.5 控制Flash 存储器的寄存器

(1) 背景事件控制寄存器(BECTL)

即使 FLMD0 引脚不由外部控制，可通过 BECTL 寄存器设置为自编程模式从而由软件控制。


但是，根据 FLMD0 引脚的处理，可能不能由软件设置到自编程模式。使用 BECTL 时，建议 FLMD0 引脚悬空。当通过外部将其下拉时，使用大于 100 kΩ 的电阻。另外，在普通操作模式下，使用 BECTL 和下拉选项。在自编程模式下，在自编程库中设置被切换到上拉。

BECTL 寄存器可由 1 位或 8 位存储器操作指令设置。

复位输入设置此寄存器为 00H。

图 22-6. 背景事件控制寄存器 (BECTL) 的格式

地址: FFFBEH 复位后: 00H R/W

符号		6	5	4	3	2	1	0
BECTL	FLMDPUP	0	0	0	0	0	0	0

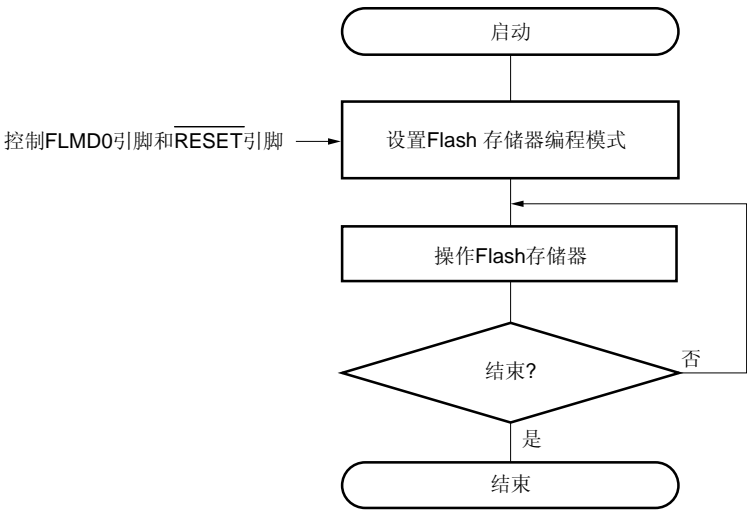
FLMDPUP	FLMD0引脚的软件控制
0	选择下拉
1	选择上拉

22.6 编程方法

22.6.1 控制Flash存储器

下图显示了操作 Flash 存储器的过程。

图 22-7. Flash 存储器操作过程



22.6.2 Flash 存储器编程模式

要使用专用 Flash 存储器编程器重写 Flash 存储器的内容，设置μPD79F9211 为 Flash 存储器编程模式。而要设置该模式，必须将 FLMD0 引脚和 TOOL0 引脚连接到 V_{DD}，并对复位信号清零。

当 on-board 写 Flash 存储器时，使用跳线改变模式。

图 22-8. Flash 存储器编程模式

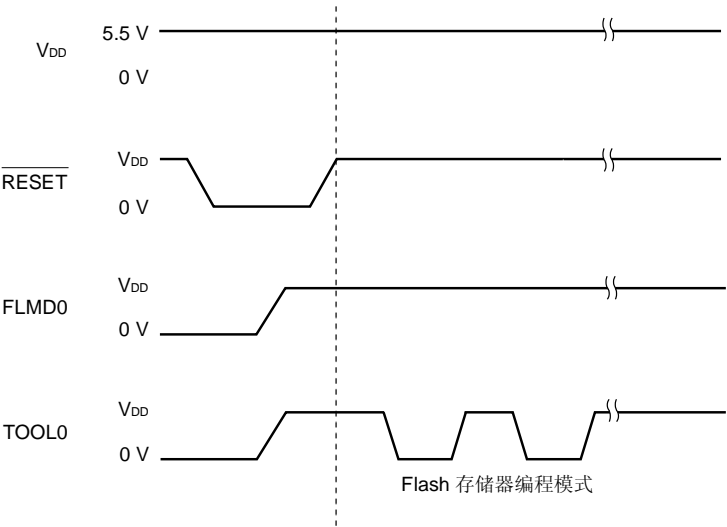


表 22-3. 复位释放后 FLMD0 引脚与操作模式之间的关系

FLMD0	操作模式
0	普通操作模式
V _{DD}	Flash 存储器编程模式

22.6.3 选择通信模式

μPD79F9211 的通信模式如下。

表 22-4. 通信模式

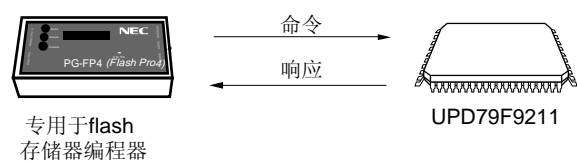
通信模式	标准设置 ^{註 1}				使用的引脚
	端口	速度	频率	倍率	
1 线模式 (专用单线 UART)	UART-ch0	1 Mbps ^{註 2}	—	—	TOOL0

- 注
- 1. Flash 存储器编程器的 GUI 上的标准设置选项。
 - 2. 波特率误差以外的因素，如信号波形瞬变，也会影响 UART 通信，因此必须像测量波特率误差一样严格测量瞬变。

22.6.4 通信命令

μPD79F9211 使用命令与专用 Flash 存储器编程器进行通信。从 Flash 存储器编程器发往μPD79F9211 的信号称为命令，从μPD79F9211 发往专用 Flash 存储器编程器的信号称为响应。

图 22-9. 通信命令



μPD79F9211 的 Flash 存储器控制命令列表如下。所有这些命令都由编程器发出，μPD79F9211 根据接收到的命令进行相应的处理。

表 22-5. Flash 存储器控制命令

分类	命令名称	功能
校验	校验	将 Flash 存储器指定区域的内容和从编程器发送的数据比较。
擦除	芯片擦除	擦除全部 Flash 存储器内容。
	块擦除	擦除 Flash 存储器的指定区域。
空白检测	块空白检测	检查 Flash 存储器的指定块擦除状态。
写入	编程	在 Flash 存储器的指定区域写入数据。
获取信息	硅标记	获取μPD79F9211 信息 (比如产品型号和 flash 存储器配置)。
	版本获得	获取μPD79F9211 硬件版本。
	校验和	获得指定区域的校验和数据。
安全	安全设置	设置安全信息。
其它	复位	用于检查通信的同步状态。
	波特率设置	当选择 UART 通信模式时设置波特率。

μPD79F9211 根据专用 Flash 编程器发出的命令返回一个响应。μPD79F9211 发出的响应列表如下。

表 22-6. 响应名称

响应名称	功能
ACK	响应命令/数据
NAK	响应非法命令/数据

22.7 安全设置

μPD79F9211 支持安全性功能，该功能禁止修改已经写入 Flash 存储器内的用户程序，所以没有授权的话，无法改变程序。

使用安全性设置命令可以执行如下操作。在设置下一个编程模式时，安全性设置有效。

- 禁止一次性擦除（片擦除）

在 on-board/off-board 编程期间，该项设置禁止对 Flash 存储器的所有块执行块擦除和一次性擦除（片擦除）命令。一旦禁止一次性擦除（片擦除）命令，所有的禁止设置项（包括禁止一次性擦除（片擦除））不能被取消。

注意事项 在对一次性擦除进行安全性设置后，不能再对该设置执行擦除操作。此外，由于禁止执行擦除命令，即使执行了写命令，与已经写到 Flash 存储器中的数据不同的数据也不能被写入。

- 禁止块擦除

在 on-board/off-board 编程期间，该项设置禁止对 Flash 存储器中的指定块进行擦除。但可通过自编程进行块擦除。

- 禁止写入

在 on-board/off-board 编程期间，该项设置禁止对 Flash 存储器所有块进行写入和块擦除。但可通过自编程进行块写入。

- 禁止重写引导簇 0

这项设置禁止在 Flash 存储器的引导簇 0（00000H~00FFFH）执行块擦除和写命令。并且执行一次性擦除（片擦除）命令。

<R>

Flash 存储器在默认设置下，允许使用一次性擦除（片擦除）、块擦除、写和重写引导簇 0 命令。以上安全性设置仅用于在 on-board/off-board 编程和自编程。各项安全性设置可以结合使用。

通过执行批量擦除（片擦除）命令，可以取消禁止擦除块和写命令。

表 22-7 表示了当 μPD79F9211 允许使用安全功能时，擦除命令和写命令之间的关系。

备注 要在自编程期间禁止写和擦除，使用 Flash 密封窗口功能（详见 22.8.2）。

表 22-7. 允许安全性功能和命令之间的关系

(1) on-board/off-board 编程期间

有效的安全性设置	执行命令		
	批量擦除（片擦除）	块擦除	写入
禁止批量擦除（片擦除）	不能批量擦除	块不能被擦除。	有效 ^註 。
禁止块擦除	可以批量擦除。		有效。
禁止写			无效。
禁止重写引导簇 0	不能批量擦除	引导簇 0 不能擦除。	引导簇 0 不能被写入。

注 确认没有数据已写在被写的区域。因为批量擦除（片擦除）禁止后，数据不能被擦除，所以在还有数据没被擦除的情况下不要写数据。

(2) 自编程期间

有效的安全性设置	执行命令	
	块擦除	写入
禁止批量擦除（片擦除）	块可被擦除。	有效
禁止块擦除		
禁止写		
禁止重写引导簇 0	引导簇 0 不能擦除。	引导簇 0 不能被写入。

备注 要在自编程期间禁止写和擦除，使用 Flash 密封窗口功能（详见 22.8.2）。

表 22-8. 在每个编程模式下设置安全性

(1) On-board/off-board 编程

安全	安全设置	如何禁止安全设置
禁止批量擦除（片擦除）	通过专用 Flash 存储器编程器的 GUI 设置等。	复位后不能禁止。
禁止块擦除		执行批量擦除（片擦除）命令
禁止写		
禁止重写引导簇 0		复位后不能禁止。

(2) 自编程

安全	安全设置	如何禁止安全设置
禁止批量擦除（片擦除）	通过信息库设置。	复位后不能禁止
禁止块擦除		在 on-board/off-board 编程期间（在自编程期间不能禁止）执行批量擦除（片擦除）命令
禁止写		
禁止重写引导簇 0		

22.8 通过自编程进行Flash存储器编程

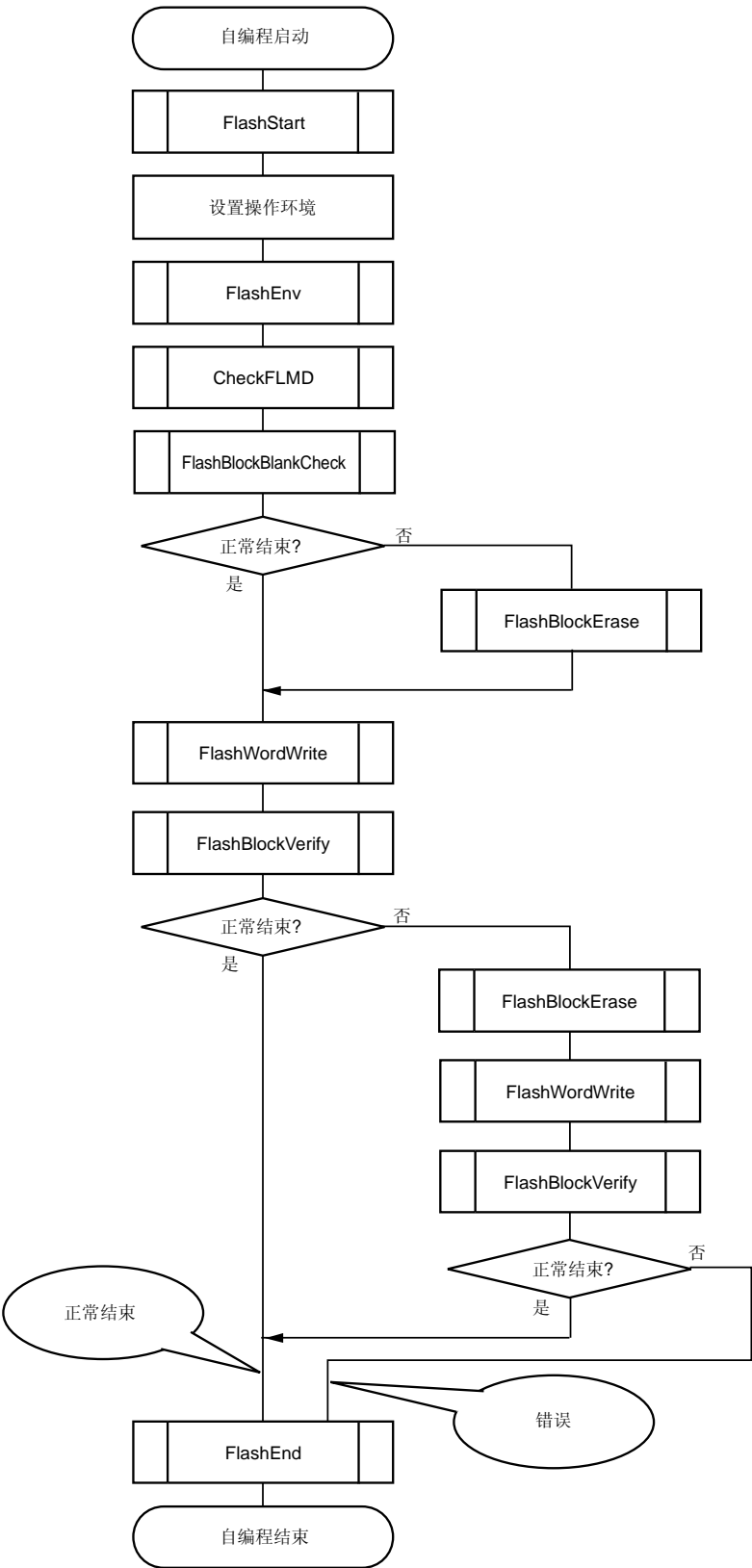
μPD79F9211 支持自编程功能，即可以通过用户程序重写 Flash 存储器。由于该功能允许用户应用程序使用 μPD79F9211 自编程库重写 flash 存储器，因此可用于对程序升级。

如果自编程期间产生中断，可以暂时停止自编程操作，而执行中断服务程序。如果在 EI 状态一个未屏蔽中断请求产生，则请求分支直接从自编程库转向中断程序。而在其后恢复自编程模式后，自编程可以继续进行。但是，中断响应时间与在普通模式下不同。

- 注意事项**
1. 当 CPU 使用副系统时钟时，不能使用自编程功能。
 2. 在自编程模式下，调用自编程开始库 (FlashStart)。
 3. 在自编程期间要禁止一个中断，与在普通操作模式下方法相同，在通过 DI 指令执行将 IE 标志清零的状态下，执行自编程库。要允许一个中断，将中断屏蔽标志清零，通过 EI 指令设置 IE 标志为（1），然后执行自编程库。
 4. 在低功耗电流模式下，自编程功能禁止。低功耗电流模式的详细内容，参见第 20 章 稳压器。

下图所示为通过使用自编程库重写 Flash 存储器的流程图。

图 22-10. 自编程的流程图（重写 Flash 存储器）



22.8.1 引导交换功能

如果在自编程期间由于电源失效或其它原因而导致引导区域的重写失败，则由于引导区域的数据损失，不能通过复位或覆盖重启程序。

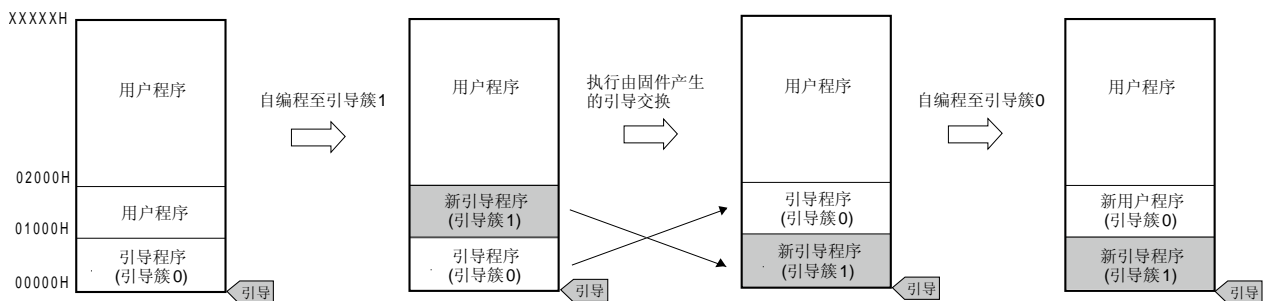
使用引导交换功能可以避免这种情况。

在擦除引导簇 0（这是一个引导程序区域^{*}）之前，先通过自编程将一个新的引导程序写入引导簇 1。当将该程序正确写入引导簇 1 后，通过使用μPD79F9211 固件的设置信息功能将引导簇 1 与引导簇 0 的内容交换，这样引导簇 1 就用作引导区域。之后对源引导程序区域，即引导簇 0，进行擦除或写操作。

这样，即使在重写引导程序区域时出现电源失效，也能够正确执行程序，这是因为程序在复位并再次执行时实行了从引导簇 1 的引导交换。

注 引导簇是一个 4 KB 区域，引导簇 0 和 1 通过引导交换功能进行交换。

图 22-11. 引导交换功能

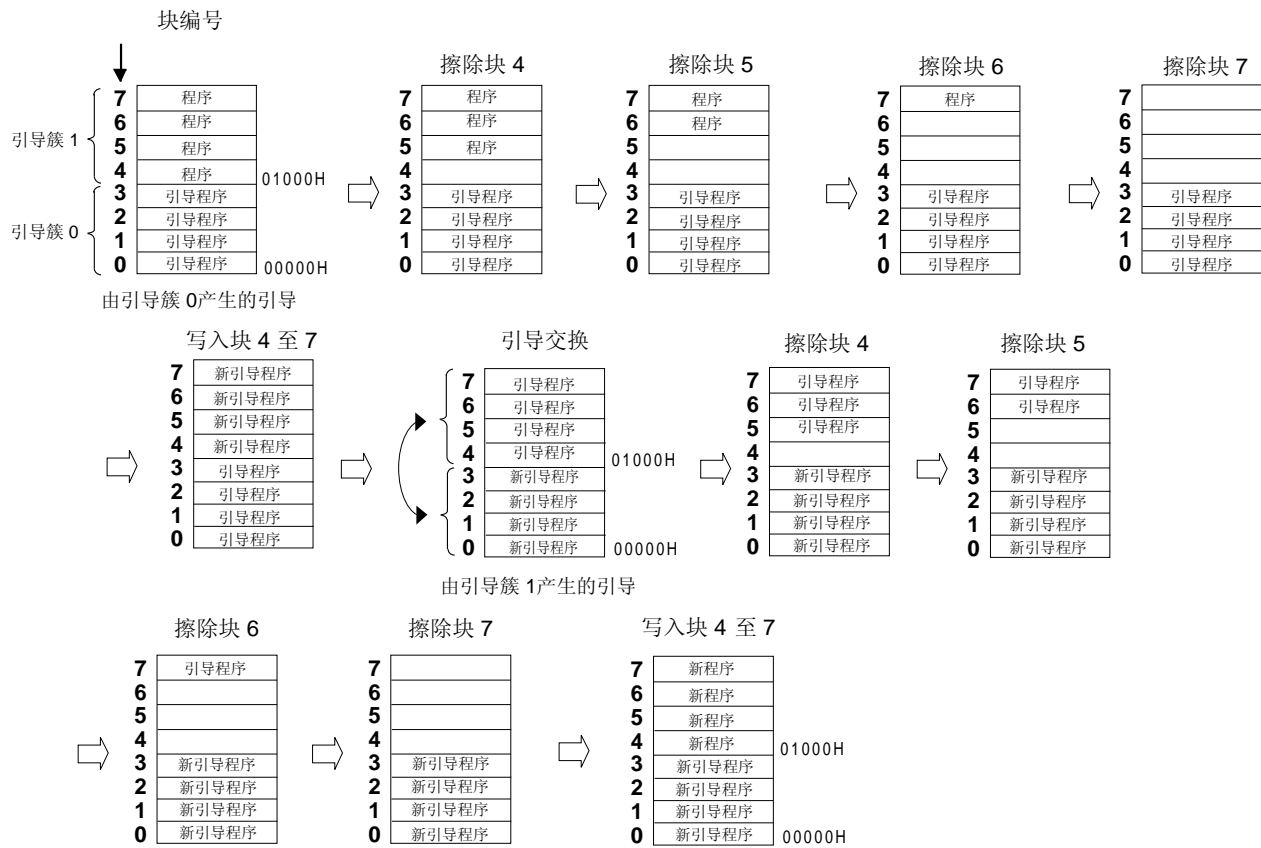


上图的示例如下。

引导簇 0：引导交换前的引导程序区域

引导簇 1：引导交换后的引导程序区域

图 22-12. 引导交换执行示例



22.8.2 Flash保护窗口功能

Flash 保护窗口功能作为一种为自编程提供的安全功能。

在自编程期间，在窗口指定范围内对 Flash 存储器进行写入和擦除允许，在指定范围以外对 Flash 存储器写入和擦除则禁止。

通过在 on-board/off-board 编程和自编程中的设置和修改，窗口范围可以扩大或缩小。但是，保护功能只有在自编程期间有效。在 on-board/off-board 编程时，允许在窗口范围以外对 Flash 存储器进行写入和擦除操作。

注意事项 如果引导簇 0 的重写禁止区域超出 Flash 保护窗口范围，则禁止重写引导簇 0 有优先权。

表 22-9. Flash保护窗口功能设置/修改方法和命令之间的关系

编程条件	窗口范围设置/修改方法	执行命令	
		块擦除	写入
自编程	通过设置信息库指定开始和结束块。	只有在窗口范围内允许块擦除。	只有在窗口范围内允许写入。
On-board/Off-board 编程	在专用 Flash 存储器编程器的 GUI 指定开始和结束块，等。	超出窗口范围块擦除也允许。	超出窗口范围写入也允许。

备注 参见 22.7 安全设置，禁止在 on-board/off-board 编程时写入/擦除。

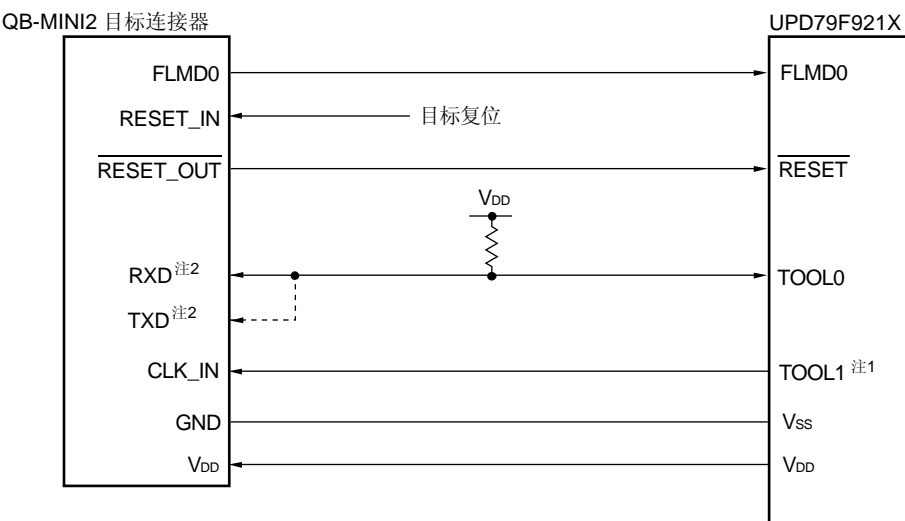
第 23 章 片上调试功能

23.1 连接QB-MINI2 到μPD79F9211

μPD79F9211 使用 V_{DD}、FLMD0、RESET、TOOL0、TOOL1^{注1} 和 V_{SS} 引脚，通过片上调试仿真器(QB-MINI2) 实现与主机的通信。

注意事项 μPD79F9211 具有片上调试功能，可用于开发和评价。在量产时不要使用该产品中的片上调试功能，因为 flash 存储器中保证的可重写次数可能会超出，从而不能保证产品的可靠性。对于因使用片上调试功能而产生的问题，NEC 电子不承担任何责任。

图 23-1. QB-MINI2 和μPD79F9211 的连接示例



- 注**
1. 在 1 线模式下通讯不需要连接，但在 2 线模式下通信需要连接。此时，因为当不连接 QB-MINI2 时 TOOL1 为未使用引脚，根据表 2-2 未使用引脚的连接执行必要的连接。
 2. 由于 QB-MINI2 内 RXD 和 TXD 短接，因此虚线连接不是必须的。当使用其它 Flash 存储器编程器时，在编程器内 RXD 和 TXD 可能不会短接。在这种情况下，他们必须在目标系统上短接。

备注 自编程在片上调试时，建议 FLMD0 引脚悬空。要通过外部下拉，使用大于 100 kΩ 的电阻。

1 线模式（单线 UART）使用 TOOL0 引脚，或 2 线模式使用 TOOL0 和 TOOL1 引脚，用于串行通信。对于 Flash 存储器编程，使用 1 线模式。1 线模式或 2 线模式被用于片上调试。表 23-1 所列为 1 线模式和 2 线模式之间的区别。

表 23-1. 1 线模式和 2 线模式之间的区别列表

通信模式	Flash 存储器编程功能	调试功能
1 线模式	可用	<ul style="list-style-type: none"> 不支持伪实时 RAM 监控 (RRM) 功能。 不支持 DMM 功能（在 RUN 时重写存储器）。 调试器的速度比 2 线模式慢 2 到 4 倍。
2 线模式	无	<ul style="list-style-type: none"> 支持伪实时 RAM 监控 (RRM) 功能。 支持 DMM 功能（在 RUN 时重写存储器）。

备注 2 线模式不用于 Flash 编程，因此，即使 TOOL1 引脚连接到 QB-MINI2 的 CLK_IN，写入操作没有问题，正常执行。

23.2 片上调试安全 ID

在 μ PD79F9211 中，一个片上调试操作控制标志存放在 flash 存储器地址 000C3H 中 (参见第 21 章 选项字节)，片上调试安全 ID 设置在地址 000C4H~000CDH 中，以防止第三方读取存储器内容。

当使用引导交换功能时，由于 000C3H、000C4H~000CDH 与 010C3H 及 010C4H~010CDH 的内容被交换，因此预先设置与 010C3H 以及 010C4H~010CDH 的内容相同的值。

关于片上调试安全 ID，请参考 QB-MINI2 片上调试仿真器用户手册 (U18371E)。

表 23-2. 片上调试安全 ID

地址	片上调试安全 ID
000C4H~000CDH	任何 10 字节的 ID 代码
010C4H~010CDH	

23.3 用户资源的确保

要执行 μ PD79F9211 和 QB-MINI2 之间的串行通信，也就是每个调试功能，必须预先保留存储器空间。

如果使用日本电器株式会社的汇编程序 RA78K0R 或编译器 CC78K0R，可以通过链接选项设置。

(1) 存储空间的确保

图 23-2 中的阴影部分是保留给调试监控程序的区域，所以用户的程序或数据不能被分配在这些空间。当使用片上调试功能时，这些空间必须是安全的，不能被用户程序使用。而且，这些区域也不能被用户程序修改。

产品 ()：内部 ROM	注 1 的地址
μPD79F9211 (16 KB)	03C00H-03FFFH

- 更多确保存储器空间的方法的详细内容，请参考 **QB-MINI2 片上调试仿真器用户手册 (U18371E)**。

24.1 BCD 修正电路功能

通过此电路，可将 BCD（二进制码十进数）码与 BCD 码的加/ 减的结果，生成 BCD 码。
将 A 寄存器作为操作数，加/减 BCDADJ 寄存器，通过执行加/减运算，可获得十进数修正操作结果。

24.2 BCD修正电路使用的寄存器

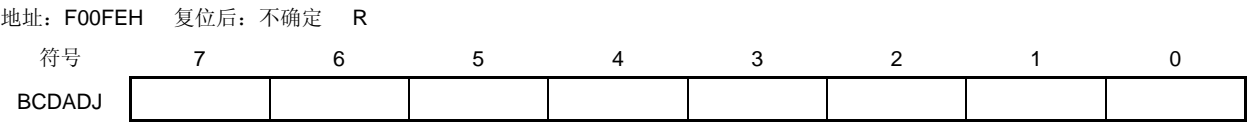
BCD 修正电路使用如下寄存器。

- BCD 修正结果寄存器 (BCDADJ)

(1) BCD 修正结果寄存器 (BCDADJ)

通过将 A 寄存器作为操作数的加/减指令，获得的加/减结果，其修正值以 BCD 码保存在 BCDADJ 寄存器中。
从 BCDADJ 寄存器读取的值，根据 A 寄存器读取时的值和 CY 与 AC 标志的不同而不同。
BCDADJ 可以由 8 位存储器操作指令读取。
复位输入后此寄存器的值不确定。

图 26-1. BCD 修正结果寄存器 (BCDADJ) 的格式



24.3 BCD 修正电路操作

BCD 修正电路的基本操作如下。

(1) 加法： 通过使用 BCD 码值，计算 BCD 码值与另一个 BCD 码值相加的结果 BCD 代码值。

<1> 要执行加法的 BCD 码值存储在 A 寄存器中。

<2> 通过 A 寄存器与第二操作数（待加 BCD 码的值）的值以二进制相加，二进制运行结果存储在 A 寄存器中，修正值存储在 BCDADJ 寄存器中。

<3> 通过把 A 寄存器（二进制相加结果）的二进制码的值加到 BCDADJ 寄存器（修正值）上，完成十进制数的修正，并且修正结果存储在 A 寄存器和 CY 寄存器中。

注意事项 从 BCDADJ 寄存器读取的值，根据 A 寄存器读取时的值和 CY 与 AC 标志的不同而不同。因此，在指令 <2> 后执行指令 <3>，而不执行任何其它指令。要在中断允许状态下执行 BCD 修正，需要在中断功能中保存和恢复 A 寄存器。PSW（CY 标志和 AC 标志）通过 RETI 指令恢复。

下面所示为例子。

例 1: $99 + 89 = 188$

指令	A 寄存器	CY 寄存器	AC 标志	BCDADJ 寄存器
MOV A, #99H ; <1>	99H	—	—	—
ADD A, #89H ; <2>	22H	1	1	66H
ADD A, !BCDADJ ; <3>	88H	1	0	—

例 2: $85 + 15 = 100$

指令	A 寄存器	CY 寄存器	AC 标志	BCDADJ 寄存器
MOV A, #85H ; <1>	85H	—	—	—
ADD A, #15H ; <2>	9AH	0	0	06H
ADD A, !BCDADJ ; <3>	00H	1	1	—

例 3: $80 + 80 = 160$

指令	A 寄存器	CY 寄存器	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：通过使用 BCD 码值，计算从 BCD 码值中减去另一个 BCD 码值的结果使用 BCD 代码值

- <1> 被减的 BCD 码值存储在 A 寄存器中。
- <2> 从 A 寄存器中按照二进制减去第二个操作数（减数的 BCD 码值），二进制减法结果存储在 A 寄存器中，修正值存储在 BCDADJ 寄存器中。
- <3> 从 A 寄存器（二进制相减结果）中减去 BCDADJ 寄存器的值（修正值）完成十进制数修正，并将修正结果存储在 A 寄存器和 CY 寄存器中。

注意事项 从 BCDADJ 寄存器读取的值，根据 A 寄存器读取时的值和 CY 与 AC 标志的不同而不同。因此，在指令 <2> 后执行指令 <3>，而不执行任何其它指令。要在中断允许状态下执行 BCD 修正，需要在中断功能中保存和恢复 A 寄存器。PSW（CY 标志和 AC 标志）通过 RETI 指令恢复。

下面所示为例子。

示例：91 – 52 = 39

指令	A 寄存器	CY 寄存器	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 25 章 指令集

本章以表格方式列出了 78K0R 微控制器的指令集。如需了解每种指令的操作和操作代码，可参阅 **78K0R 微控制器指令用户手册 (U17792E)**。

备注 表 25-5 操作列表中的阴影部分表示操作或指令格式是 78K0R 微控制器新添加的。

25.1 操作列表使用规则

25.1.1 操作数标识符和指定方法

操作数在每个指令的“操作数”栏中进行描述，与指令操作数标识符的描述方法一致（详细内容请参考汇编程序规范）。如果有两种或两种以上的标识方法，可选其中之一。大写字母和符号#，！，！！，\$，\$！，[]，和 ES：是关键字，必须按其原样书写。每种符号的含义如下所示。

- #：立即数标识
- !：16 位绝对地址标识
- !!：20 位绝对地址标识
- \$：8 位相对地址标识
- \$!：16 位相对地址标识
- []：间接地址标识
- ES：扩展地址标识

立即数用来描述一个数值型数据或标号。当使用标号时，注意必须加上符号#，！，！！，\$，\$！，[]和 ES：。

对应操作数寄存器标识符 r 和 rp，功能名称（X，A，C，等）或绝对名称（下表括号中的名称：R0，R1，R2 等）都可用于标识。

表 25-1. 操作数标识符和描述方法

标识符	描述方法
r	X (R0)，A (R1)，C (R2)，B (R3)，E (R4)，D (R5)，L (R6)，H (R7)
rp	AX (RP0)，BC (RP1)，DE (RP2)，HL (RP3)
sfr	特殊功能寄存器符号（SFR 符号）
sfrp	特殊功能寄存器符号（16 位可操作表 SFR 符号。仅用于偶地址 [※] ）
saddr	FFE20H～FFF1FH 立即数或标号
saddrp	FFE20H～FF1FH 立即数或标号（仅用于偶地址 [※] ）
addr20	00000H～FFFFFH 立即数或标号
addr16	0000H～FFFFH 立即数或标号（仅用于 16 位数据传送指令的偶地址 [※] ）
addr5	0080H～00BFH 立即数或标号（仅用于偶地址）
word	16 位立即数或标号
byte	8 位立即数或标号
bit	3 位立即数或标号
RBn	RB0～RB3

注 当奇地址指定时，第0位 = 0。

备注 特殊功能寄存器符号，参见表 3-5 SFR 列表和表 3-6 扩展 SFR（第二 SFR）列表。

25.1.2 操作栏描述

指令执行时的运算显示在“运算”栏中，并使用下列符号。

表 25-2. “运算”栏中的符号

符号	功能
A	A 寄存器；8 位累加器
X	X 寄存器
B	B 寄存器
C	C 寄存器
D	D 寄存器
E	E 寄存器
H	H 寄存器
L	L 寄存器
ES	ES 寄存器
CS	CS 寄存器
AX	AX 寄存器对；16 位累加器
BC	BC 寄存器对
DE	DE 寄存器对
HL	HL 寄存器对
PC	程序计数器
SP	堆栈指针
PSW	程序状态字
CY	进位标志
AC	辅助进位标志
Z	零标志
RBS	寄存器 bank 选择标志
IE	中断请求允许标志
()	括号中的地址或寄存器所指的存储单元的内容
X _H , X _L	16 位寄存器: XH = 高 8 位, XL = 低 8 位
X _S , X _H , X _L	20 位寄存器: XS = (位 19~16), XH = (位 15~8), XL = (位 7~0)
^	逻辑与 (AND)
∨	逻辑或 (OR)
⊕	逻辑异或 (异或)
—	数据取反
addr5	16 位立即数 (只在 0080H~00BFH 的偶地址)
addr16	16 位立即数
addr20	20 位立即数
jdisp8	带符号的 8 位数据 (偏移量)
jdisp16	带符号的 16 位数据 (偏移量)

25.1.3 标志操作栏的描述

指令执行时的标志值的改变显示在“标志”栏中，并使用下列符号。

表 25-3. “标志”栏中的符号

标志	标志值的改变
(空白)	不变
0	清零
1	设置为 1
×	根据结果进行设置/清零
R	恢复先前保存的值

25.1.4 PREFIX指令

带“ES:”的指令具有 PREFIX 操作码作为前缀以扩展可访问数据区域至 1MB 空间(00000H 至 FFFFFH)，通过将 ES 寄存器的值添加到 F0000H 至 FFFFFH 的 64KB 区域。当 PREFIX 操作码作为前缀附加到目标指令时，在 PREFIX 操作码后只会立即执行一条指令作为添加 ES 寄存器值的地址。

表 25-4. PREFIX 操作码的使用示例

指令	操作代码				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	–
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	–	–	–	–
MOV A, ES:[HL]	11H	8BH	–	–	–

注意事项 在执行 PREFIX 指令前，用 MOV ES、A 等设置 ES 寄存器的值。

25.2 操作列表

表 25-5. 操作列表 (1/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据 传送	MOV	r, #byte	2	1	–	$r \leftarrow \text{byte}$			
		saddr, #byte	3	1	–	$(\text{saddr}) \leftarrow \text{byte}$			
		sfr, #byte	3	1	–	$\text{sfr} \leftarrow \text{byte}$			
		!addr16, #byte	4	1	–	$(\text{addr16}) \leftarrow \text{byte}$			
		A, r ^{注 3}	1	1	–	$A \leftarrow r$			
		r, A ^{注 3}	1	1	–	$r \leftarrow A$			
		A, saddr	2	1	–	$A \leftarrow (\text{saddr})$			
		saddr, A	2	1	–	$(\text{saddr}) \leftarrow A$			
		A, sfr	2	1	–	$A \leftarrow \text{sfr}$			
		sfr, A	2	1	–	$\text{sfr} \leftarrow A$			
		A, !addr16	3	1	4	$A \leftarrow (\text{addr16})$			
		!addr16, A	3	1	–	$(\text{addr16}) \leftarrow A$			
		PSW, #byte	3	3	–	$\text{PSW} \leftarrow \text{byte}$	×	×	×
		A, PSW	2	1	–	$A \leftarrow \text{PSW}$			
		PSW, A	2	3	–	$\text{PSW} \leftarrow A$	×	×	×
		ES, #byte	2	1	–	$\text{ES} \leftarrow \text{byte}$			
		ES, saddr	3	1	–	$\text{ES} \leftarrow (\text{saddr})$			
		A, ES	2	1	–	$A \leftarrow \text{ES}$			
		ES, A	2	1	–	$\text{ES} \leftarrow A$			
		CS, #byte	3	1	–	$\text{CS} \leftarrow \text{byte}$			
		A, CS	2	1	–	$A \leftarrow \text{CS}$			
		CS, A	2	1	–	$\text{CS} \leftarrow A$			
		A, [DE]	1	1	4	$A \leftarrow (\text{DE})$			
		[DE], A	1	1	–	$(\text{DE}) \leftarrow A$			
		[DE + byte], #byte	3	1	–	$(\text{DE} + \text{byte}) \leftarrow \text{byte}$			
		A, [DE + byte]	2	1	4	$A \leftarrow (\text{DE} + \text{byte})$			
		[DE + byte], A	2	1	–	$(\text{DE} + \text{byte}) \leftarrow A$			
		A, [HL]	1	1	4	$A \leftarrow (\text{HL})$			
		[HL], A	1	1	–	$(\text{HL}) \leftarrow A$			
		[HL + byte], #byte	3	1	–	$(\text{HL} + \text{byte}) \leftarrow \text{byte}$			

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. r = A 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。
 2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (2/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据 传送	MOV	A, [HL + byte]	2	1	4	$A \leftarrow (HL + \text{byte})$			
		[HL + byte], A	2	1	–	$(HL + \text{byte}) \leftarrow A$			
		A, [HL + B]	2	1	4	$A \leftarrow (HL + B)$			
		[HL + B], A	2	1	–	$(HL + B) \leftarrow A$			
		A, [HL + C]	2	1	4	$A \leftarrow (HL + C)$			
		[HL + C], A	2	1	–	$(HL + C) \leftarrow A$			
		word[B], #byte	4	1	–	$(B + \text{word}) \leftarrow \text{byte}$			
		A, word[B]	3	1	4	$A \leftarrow (B + \text{word})$			
		word[B], A	3	1	–	$(B + \text{word}) \leftarrow A$			
		word[C], #byte	4	1	–	$(C + \text{word}) \leftarrow \text{byte}$			
		A, word[C]	3	1	4	$A \leftarrow (C + \text{word})$			
		word[C], A	3	1	–	$(C + \text{word}) \leftarrow A$			
		word[BC], #byte	4	1	–	$(BC + \text{word}) \leftarrow \text{byte}$			
		A, word[BC]	3	1	4	$A \leftarrow (BC + \text{word})$			
		word[BC], A	3	1	–	$(BC + \text{word}) \leftarrow A$			
		[SP + byte], #byte	3	1	–	$(SP + \text{byte}) \leftarrow \text{byte}$			
		A, [SP + byte]	2	1	–	$A \leftarrow (SP + \text{byte})$			
		[SP + byte], A	2	1	–	$(SP + \text{byte}) \leftarrow A$			
		B, saddr	2	1	–	$B \leftarrow (\text{saddr})$			
		B, !addr16	3	1	4	$B \leftarrow (\text{addr16})$			
		C, saddr	2	1	–	$C \leftarrow (\text{saddr})$			
		C, !addr16	3	1	4	$C \leftarrow (\text{addr16})$			
		X, saddr	2	1	–	$X \leftarrow (\text{saddr})$			
		X, !addr16	3	1	4	$X \leftarrow (\text{addr16})$			
		ES:!addr16, #byte	5	2	–	$(ES, \text{addr16}) \leftarrow \text{byte}$			
		A, ES:!addr16	4	2	5	$A \leftarrow (ES, \text{addr16})$			
		ES:!addr16, A	4	2	–	$(ES, \text{addr16}) \leftarrow A$			
		A, ES:[DE]	2	2	5	$A \leftarrow (ES, DE)$			
		ES:[DE], A	2	2	–	$(ES, DE) \leftarrow A$			
		ES:[DE + byte], #byte	4	2	–	$((ES, DE) + \text{byte}) \leftarrow \text{byte}$			
		A, ES:[DE + byte]	3	2	5	$A \leftarrow ((ES, DE) + \text{byte})$			
		ES:[DE + byte], A	3	2	–	$((ES, DE) + \text{byte}) \leftarrow A$			

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。
 2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (3/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据 传送	MOV	A, ES:[HL]	2	2	5	$A \leftarrow (ES, HL)$			
		ES:[HL], A	2	2	—	$(ES, HL) \leftarrow A$			
		ES:[HL + byte], #byte	4	2	—	$((ES, HL) + \text{byte}) \leftarrow \text{byte}$			
		A, ES:[HL + byte]	3	2	5	$A \leftarrow ((ES, HL) + \text{byte})$			
		ES:[HL + byte], A	3	2	—	$((ES, HL) + \text{byte}) \leftarrow A$			
		A, ES:[HL + B]	3	2	5	$A \leftarrow ((ES, HL) + B)$			
		ES:[HL + B], A	3	2	—	$((ES, HL) + B) \leftarrow A$			
		A, ES:[HL + C]	3	2	5	$A \leftarrow ((ES, HL) + C)$			
		ES:[HL + C], A	3	2	—	$((ES, HL) + C) \leftarrow A$			
		ES:word[B], #byte	5	2	—	$((ES, B) + \text{word}) \leftarrow \text{byte}$			
		A, ES:word[B]	4	2	5	$A \leftarrow ((ES, B) + \text{word})$			
		ES:word[B], A	4	2	—	$((ES, B) + \text{word}) \leftarrow A$			
		ES:word[C], #byte	5	2	—	$((ES, C) + \text{word}) \leftarrow \text{byte}$			
		A, ES:word[C]	4	2	5	$A \leftarrow ((ES, C) + \text{word})$			
		ES:word[C], A	4	2	—	$((ES, C) + \text{word}) \leftarrow A$			
		ES:word[BC], #byte	5	2	—	$((ES, BC) + \text{word}) \leftarrow \text{byte}$			
		A, ES:word[BC]	4	2	5	$A \leftarrow ((ES, BC) + \text{word})$			
		ES:word[BC], A	4	2	—	$((ES, BC) + \text{word}) \leftarrow A$			
		B, ES:!addr16	4	2	5	$B \leftarrow (ES, \text{addr16})$			
		C, ES:!addr16	4	2	5	$C \leftarrow (ES, \text{addr16})$			
		X, ES:!addr16	4	2	5	$X \leftarrow (ES, \text{addr16})$			
	XCH	A, r ^{#3}	1 (r = X) 2 (除 r = X)	1	—	$A \longleftrightarrow r$			
		A, saddr	3	2	—	$A \longleftrightarrow (\text{saddr})$			
		A, sfr	3	2	—	$A \longleftrightarrow \text{sfr}$			
		A, !addr16	4	2	—	$A \longleftrightarrow (\text{addr16})$			
		A, [DE]	2	2	—	$A \longleftrightarrow (DE)$			
		A, [DE + byte]	3	2	—	$A \longleftrightarrow (DE + \text{byte})$			
		A, [HL]	2	2	—	$A \longleftrightarrow (HL)$			
		A, [HL + byte]	3	2	—	$A \longleftrightarrow (HL + \text{byte})$			
		A, [HL + B]	2	2	—	$A \longleftrightarrow (HL + B)$			
		A, [HL + C]	2	2	—	$A \longleftrightarrow (HL + C)$			

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. r = A 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。
 2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (4/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位数据 传送	XCH	A, ES:laddr16	5	3	–	$A \longleftrightarrow (ES, \text{addr16})$			
		A, ES:[DE]	3	3	–	$A \longleftrightarrow (ES, DE)$			
		A, ES:[DE + byte]	4	3	–	$A \longleftrightarrow ((ES, DE) + \text{byte})$			
		A, ES:[HL]	3	3	–	$A \longleftrightarrow (ES, HL)$			
		A, ES:[HL + byte]	4	3	–	$A \longleftrightarrow ((ES, HL) + \text{byte})$			
		A, ES:[HL + B]	3	3	–	$A \longleftrightarrow ((ES, HL) + B)$			
		A, ES:[HL + C]	3	3	–	$A \longleftrightarrow ((ES, HL) + C)$			
	ONEB	A	1	1	–	$A \leftarrow 01H$			
		X	1	1	–	$X \leftarrow 01H$			
		B	1	1	–	$B \leftarrow 01H$			
		C	1	1	–	$C \leftarrow 01H$			
		saddr	2	1	–	$(saddr) \leftarrow 01H$			
		laddr16	3	1	–	$(\text{addr16}) \leftarrow 01H$			
		ES:laddr16	4	2	–	$(ES, \text{addr16}) \leftarrow 01H$			
	CLRB	A	1	1	–	$A \leftarrow 00H$			
		X	1	1	–	$X \leftarrow 00H$			
		B	1	1	–	$B \leftarrow 00H$			
		C	1	1	–	$C \leftarrow 00H$			
		saddr	2	1	–	$(saddr) \leftarrow 00H$			
		laddr16	3	1	–	$(\text{addr16}) \leftarrow 00H$			
		ES:laddr16	4	2	–	$(ES, \text{addr16}) \leftarrow 00H$			
	MOVS	[HL + byte], X	3	1	–	$(HL + \text{byte}) \leftarrow X$	×		×
		ES:[HL + byte], X	4	2	–	$(ES, HL + \text{byte}) \leftarrow X$	×		×
16 位数据 传送	MOVW	rp, #word	3	1	–	$rp \leftarrow \text{word}$			
		saddrp, #word	4	1	–	$(saddrp) \leftarrow \text{word}$			
		sfrp, #word	4	1	–	$sfrp \leftarrow \text{word}$			
		AX, saddrp	2	1	–	$AX \leftarrow (saddrp)$			
		saddrp, AX	2	1	–	$(saddrp) \leftarrow AX$			
		AX, sfrp	2	1	–	$AX \leftarrow sfrp$			
		sfrp, AX	2	1	–	$sfrp \leftarrow AX$			
		AX, rp ^{注 3}	1	1	–	$AX \leftarrow rp$			
		rp, AX ^{注 3}	1	1	–	$rp \leftarrow AX$			

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. rp = AX 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。
 2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (5/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位数据 传送	MOVW	AX, !addr16	3	1	4	AX ← (addr16)			
		!addr16, AX	3	1	–	(addr16) ← AX			
		AX, [DE]	1	1	4	AX ← (DE)			
		[DE], AX	1	1	–	(DE) ← AX			
		AX, [DE + byte]	2	1	4	AX ← (DE + byte)			
		[DE + byte], AX	2	1	–	(DE + byte) ← AX			
		AX, [HL]	1	1	4	AX ← (HL)			
		[HL], AX	1	1	–	(HL) ← AX			
		AX, [HL + byte]	2	1	4	AX ← (HL + byte)			
		[HL + byte], AX	2	1	–	(HL + byte) ← AX			
		AX, word[B]	3	1	4	AX ← (B + word)			
		word[B], AX	3	1	–	(B + word) ← AX			
		AX, word[C]	3	1	4	AX ← (C + word)			
		word[C], AX	3	1	–	(C + word) ← AX			
		AX, word[BC]	3	1	4	AX ← (BC + word)			
		word[BC], AX	3	1	–	(BC + word) ← AX			
		AX, [SP + byte]	2	1	–	AX ← (SP + byte)			
		[SP + byte], AX	2	1	–	(SP + byte) ← AX			
		BC, saddrp	2	1	–	BC ← (saddrp)			
		BC, !addr16	3	1	4	BC ← (addr16)			
		DE, saddrp	2	1	–	DE ← (saddrp)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		HL, saddrp	2	1	–	HL ← (saddrp)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	–	(ES, addr16) ← AX			
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)			
		ES:[DE], AX	2	2	–	(ES, DE) ← AX			
		AX, ES:[DE + byte]	3	2	5	AX ← ((ES, DE) + byte)			
		ES:[DE + byte], AX	3	2	–	((ES, DE) + byte) ← AX			
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)			
		ES:[HL], AX	2	2	–	(ES, HL) ← AX			

注 1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。

2. 当程序存储器区域被访问时。

备注 1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。

2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (6/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位数据 传送	MOVW	AX, ES:[HL + byte]	3	2	5	$AX \leftarrow ((ES, HL) + \text{byte})$			
		ES:[HL + byte], AX	3	2	–	$((ES, HL) + \text{byte}) \leftarrow AX$			
		AX, ES:word[B]	4	2	5	$AX \leftarrow ((ES, B) + \text{word})$			
		ES:word[B], AX	4	2	–	$((ES, B) + \text{word}) \leftarrow AX$			
		AX, ES:word[C]	4	2	5	$AX \leftarrow ((ES, C) + \text{word})$			
		ES:word[C], AX	4	2	–	$((ES, C) + \text{word}) \leftarrow AX$			
		AX, ES:word[BC]	4	2	5	$AX \leftarrow ((ES, BC) + \text{word})$			
		ES:word[BC], AX	4	2	–	$((ES, BC) + \text{word}) \leftarrow AX$			
		BC, ES:!addr16	4	2	5	$BC \leftarrow (ES, \text{addr16})$			
		DE, ES:!addr16	4	2	5	$DE \leftarrow (ES, \text{addr16})$			
		HL, ES:!addr16	4	2	5	$HL \leftarrow (ES, \text{addr16})$			
	XCHW	AX, rp ^{注 3}	1	1	–	$AX \leftrightarrow rp$			
	ONEW	AX	1	1	–	$AX \leftarrow 0001H$			
		BC	1	1	–	$BC \leftarrow 0001H$			
	CLRW	AX	1	1	–	$AX \leftarrow 0000H$			
		BC	1	1	–	$BC \leftarrow 0000H$			
8 位操作	ADD	A, #byte	2	1	–	$A, CY \leftarrow A + \text{byte}$	×	×	×
		saddr, #byte	3	2	–	$(saddr), CY \leftarrow (saddr) + \text{byte}$	×	×	×
		A, r ^{注 4}	2	1	–	$A, CY \leftarrow A + r$	×	×	×
		r, A	2	1	–	$r, CY \leftarrow r + A$	×	×	×
		A, saddr	2	1	–	$A, CY \leftarrow A + (saddr)$	×	×	×
		A, !addr16	3	1	4	$A, CY \leftarrow A + (\text{addr16})$	×	×	×
		A, [HL]	1	1	4	$A, CY \leftarrow A + (HL)$	×	×	×
		A, [HL + byte]	2	1	4	$A, CY \leftarrow A + (HL + \text{byte})$	×	×	×
		A, [HL + B]	2	1	4	$A, CY \leftarrow A + (HL + B)$	×	×	×
		A, [HL + C]	2	1	4	$A, CY \leftarrow A + (HL + C)$	×	×	×
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A + (ES, \text{addr16})$	×	×	×
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A + (ES, HL)$	×	×	×
		A, ES:[HL + byte]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + \text{byte})$	×	×	×
		A, ES:[HL + B]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + B)$	×	×	×
		A, ES:[HL + C]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + C)$	×	×	×

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. rp = AX 除外
 4. r = A 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟(f_{cpu})的一个周期。
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。

表 25-5. 操作列表 (7/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	ADDC	A, #byte	2	1	–	$A, CY \leftarrow A + \text{byte} + CY$	x	x	x
		saddr, #byte	3	2	–	$(saddr), CY \leftarrow (saddr) + \text{byte} + CY$	x	x	x
		A, r ^{注 3}	2	1	–	$A, CY \leftarrow A + r + CY$	x	x	x
		r, A	2	1	–	$r, CY \leftarrow r + A + CY$	x	x	x
		A, saddr	2	1	–	$A, CY \leftarrow A + (saddr) + CY$	x	x	x
		A, !addr16	3	1	4	$A, CY \leftarrow A + (\text{addr16}) + CY$	x	x	x
		A, [HL]	1	1	4	$A, CY \leftarrow A + (HL) + CY$	x	x	x
		A, [HL + byte]	2	1	4	$A, CY \leftarrow A + (HL + \text{byte}) + CY$	x	x	x
		A, [HL + B]	2	1	4	$A, CY \leftarrow A + (HL + B) + CY$	x	x	x
		A, [HL + C]	2	1	4	$A, CY \leftarrow A + (HL + C) + CY$	x	x	x
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A + (ES, \text{addr16}) + CY$	x	x	x
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A + (ES, HL) + CY$	x	x	x
		A, ES:[HL + byte]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + \text{byte}) + CY$	x	x	x
		A, ES:[HL + B]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + B) + CY$	x	x	x
		A, ES:[HL + C]	3	2	5	$A, CY \leftarrow A + ((ES, HL) + C) + CY$	x	x	x
	SUB	A, #byte	2	1	–	$A, CY \leftarrow A - \text{byte}$	x	x	x
		saddr, #byte	3	2	–	$(saddr), CY \leftarrow (saddr) - \text{byte}$	x	x	x
		A, r ^{注 3}	2	1	–	$A, CY \leftarrow A - r$	x	x	x
		r, A	2	1	–	$r, CY \leftarrow r - A$	x	x	x
		A, saddr	2	1	–	$A, CY \leftarrow A - (saddr)$	x	x	x
		A, !addr16	3	1	4	$A, CY \leftarrow A - (\text{addr16})$	x	x	x
		A, [HL]	1	1	4	$A, CY \leftarrow A - (HL)$	x	x	x
		A, [HL + byte]	2	1	4	$A, CY \leftarrow A - (HL + \text{byte})$	x	x	x
		A, [HL + B]	2	1	4	$A, CY \leftarrow A - (HL + B)$	x	x	x
		A, [HL + C]	2	1	4	$A, CY \leftarrow A - (HL + C)$	x	x	x
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A - (ES:\text{addr16})$	x	x	x
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A - (ES:HL)$	x	x	x
		A, ES:[HL + byte]	3	2	5	$A, CY \leftarrow A - ((ES:HL) + \text{byte})$	x	x	x
		A, ES:[HL + B]	3	2	5	$A, CY \leftarrow A - ((ES:HL) + B)$	x	x	x
		A, ES:[HL + C]	3	2	5	$A, CY \leftarrow A - ((ES:HL) + C)$	x	x	x

注 1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。

2. 当程序存储器区域被访问时。

3. r = A 除外

备注 1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。

2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (8/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	SUBC	A, #byte	2	1	–	$A, CY \leftarrow A - \text{byte} - CY$	×	×	×
		saddr, #byte	3	2	–	$(saddr), CY \leftarrow (saddr) - \text{byte} - CY$	×	×	×
		A, r ^{注 3}	2	1	–	$A, CY \leftarrow A - r - CY$	×	×	×
		r, A	2	1	–	$r, CY \leftarrow r - A - CY$	×	×	×
		A, saddr	2	1	–	$A, CY \leftarrow A - (saddr) - CY$	×	×	×
		A, !addr16	3	1	4	$A, CY \leftarrow A - (\text{addr16}) - CY$	×	×	×
		A, [HL]	1	1	4	$A, CY \leftarrow A - (\text{HL}) - CY$	×	×	×
		A, [HL + byte]	2	1	4	$A, CY \leftarrow A - (\text{HL} + \text{byte}) - CY$	×	×	×
		A, [HL + B]	2	1	4	$A, CY \leftarrow A - (\text{HL} + B) - CY$	×	×	×
		A, [HL + C]	2	1	4	$A, CY \leftarrow A - (\text{HL} + C) - CY$	×	×	×
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A - (\text{ES:addr16}) - CY$	×	×	×
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A - (\text{ES:HL}) - CY$	×	×	×
		A, ES:[HL + byte]	3	2	5	$A, CY \leftarrow A - ((\text{ES:HL}) + \text{byte}) - CY$	×	×	×
		A, ES:[HL + B]	3	2	5	$A, CY \leftarrow A - ((\text{ES:HL}) + B) - CY$	×	×	×
		A, ES:[HL + C]	3	2	5	$A, CY \leftarrow A - ((\text{ES:HL}) + C) - CY$	×	×	×
	AND	A, #byte	2	1	–	$A \leftarrow A \wedge \text{byte}$	×		
		saddr, #byte	3	2	–	$(saddr) \leftarrow (saddr) \wedge \text{byte}$	×		
		A, r ^{注 3}	2	1	–	$A \leftarrow A \wedge r$	×		
		r, A	2	1	–	$r \leftarrow r \wedge A$	×		
		A, saddr	2	1	–	$A \leftarrow A \wedge (saddr)$	×		
		A, !addr16	3	1	4	$A \leftarrow A \wedge (\text{addr16})$	×		
		A, [HL]	1	1	4	$A \leftarrow A \wedge (\text{HL})$	×		
		A, [HL + byte]	2	1	4	$A \leftarrow A \wedge (\text{HL} + \text{byte})$	×		
		A, [HL + B]	2	1	4	$A \leftarrow A \wedge (\text{HL} + B)$	×		
		A, [HL + C]	2	1	4	$A \leftarrow A \wedge (\text{HL} + C)$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \wedge (\text{ES:addr16})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \wedge (\text{ES:HL})$	×		
		A, ES:[HL + byte]	3	2	5	$A \leftarrow A \wedge ((\text{ES:HL}) + \text{byte})$	×		
		A, ES:[HL + B]	3	2	5	$A \leftarrow A \wedge ((\text{ES:HL}) + B)$	×		
		A, ES:[HL + C]	3	2	5	$A \leftarrow A \wedge ((\text{ES:HL}) + C)$	×		

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. r = A 除外

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。
 2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (9/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	OR	A, #byte	2	1	–	$A \leftarrow A \vee \text{byte}$	×		
		saddr, #byte	3	2	–	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	×		
		A, r ^{注 3}	2	1	–	$A \leftarrow A \vee r$	×		
		r, A	2	1	–	$r \leftarrow r \vee A$	×		
		A, saddr	2	1	–	$A \leftarrow A \vee (\text{saddr})$	×		
		A, !addr16	3	1	4	$A \leftarrow A \vee (\text{addr16})$	×		
		A, [HL]	1	1	4	$A \leftarrow A \vee (\text{HL})$	×		
		A, [HL + byte]	2	1	4	$A \leftarrow A \vee (\text{HL} + \text{byte})$	×		
		A, [HL + B]	2	1	4	$A \leftarrow A \vee (\text{HL} + B)$	×		
		A, [HL + C]	2	1	4	$A \leftarrow A \vee (\text{HL} + C)$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \vee (\text{ES:addr16})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \vee (\text{ES:HL})$	×		
		A, ES:[HL + byte]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + \text{byte})$	×		
		A, ES:[HL + B]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + B)$	×		
		A, ES:[HL + C]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + C)$	×		
	XOR	A, #byte	2	1	–	$A \leftarrow A \nabla \text{byte}$	×		
		saddr, #byte	3	2	–	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$	×		
		A, r ^{注 3}	2	1	–	$A \leftarrow A \nabla r$	×		
		r, A	2	1	–	$r \leftarrow r \nabla A$	×		
		A, saddr	2	1	–	$A \leftarrow A \nabla (\text{saddr})$	×		
		A, !addr16	3	1	4	$A \leftarrow A \nabla (\text{addr16})$	×		
		A, [HL]	1	1	4	$A \leftarrow A \nabla (\text{HL})$	×		
		A, [HL + byte]	2	1	4	$A \leftarrow A \nabla (\text{HL} + \text{byte})$	×		
		A, [HL + B]	2	1	4	$A \leftarrow A \nabla (\text{HL} + B)$	×		
		A, [HL + C]	2	1	4	$A \leftarrow A \nabla (\text{HL} + C)$	×		
		A, ES:!addr16	4	2	5	$A \leftarrow A \nabla (\text{ES:addr16})$	×		
		A, ES:[HL]	2	2	5	$A \leftarrow A \nabla (\text{ES:HL})$	×		
		A, ES:[HL + byte]	3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + \text{byte})$	×		
		A, ES:[HL + B]	3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + B)$	×		
		A, ES:[HL + C]	3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + C)$	×		

注 1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。

2. 当程序存储器区域被访问时。

3. r = A 除外

备注 1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。

2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (10/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
8 位操作	CMP	A, #byte	2	1	–	A – byte	x	x	x
		saddr, #byte	3	1	–	(saddr) – byte	x	x	x
		A, r ^{注 3}	2	1	–	A – r	x	x	x
		r, A	2	1	–	r – A	x	x	x
		A, saddr	2	1	–	A – (saddr)	x	x	x
		A, !addr16	3	1	4	A – (addr16)	x	x	x
		A, [HL]	1	1	4	A – (HL)	x	x	x
		A, [HL + byte]	2	1	4	A – (HL + byte)	x	x	x
		A, [HL + B]	2	1	4	A – (HL + B)	x	x	x
		A, [HL + C]	2	1	4	A – (HL + C)	x	x	x
		!addr16, #byte	4	1	4	(addr16) – byte	x	x	x
		A, ES:!addr16	4	2	5	A – (ES:addr16)	x	x	x
		A, ES:[HL]	2	2	5	A – (ES:HL)	x	x	x
		A, ES:[HL + byte]	3	2	5	A – ((ES:HL) + byte)	x	x	x
		A, ES:[HL + B]	3	2	5	A – ((ES:HL) + B)	x	x	x
		A, ES:[HL + C]	3	2	5	A – ((ES:HL) + C)	x	x	x
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	x	x	x
	CMP0	A	1	1	–	A – 00H	x	x	x
		X	1	1	–	X – 00H	x	x	x
		B	1	1	–	B – 00H	x	x	x
		C	1	1	–	C – 00H	x	x	x
		saddr	2	1	–	(saddr) – 00H	x	x	x
		!addr16	3	1	4	(addr16) – 00H	x	x	x
		ES:!addr16	4	2	5	(ES:addr16) – 00H	x	x	x
	CMPS	X, [HL + byte]	3	1	4	X – (HL + byte)	x	x	x
		X, ES:[HL + byte]	4	2	5	X – ((ES:HL) + byte)	x	x	x

注 1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。

2. 当程序存储器区域被访问时。

3. r = A 除外

备注 1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。

2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (11/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
16 位操作	ADDW	AX, #word	3	1	–	$AX, CY \leftarrow AX + word$	x	x	x
		AX, AX	1	1	–	$AX, CY \leftarrow AX + AX$	x	x	x
		AX, BC	1	1	–	$AX, CY \leftarrow AX + BC$	x	x	x
		AX, DE	1	1	–	$AX, CY \leftarrow AX + DE$	x	x	x
		AX, HL	1	1	–	$AX, CY \leftarrow AX + HL$	x	x	x
		AX, saddrp	2	1	–	$AX, CY \leftarrow AX + (saddrp)$	x	x	x
		AX, !addr16	3	1	4	$AX, CY \leftarrow AX + (addr16)$	x	x	x
		AX, [HL+byte]	3	1	4	$AX, CY \leftarrow AX + (HL + byte)$	x	x	x
		AX, ES:!addr16	4	2	5	$AX, CY \leftarrow AX + (ES:addr16)$	x	x	x
		AX, ES: [HL+byte]	4	2	5	$AX, CY \leftarrow AX + ((ES:HL) + byte)$	x	x	x
	SUBW	AX, #word	3	1	–	$AX, CY \leftarrow AX - word$	x	x	x
		AX, BC	1	1	–	$AX, CY \leftarrow AX - BC$	x	x	x
		AX, DE	1	1	–	$AX, CY \leftarrow AX - DE$	x	x	x
		AX, HL	1	1	–	$AX, CY \leftarrow AX - HL$	x	x	x
		AX, saddrp	2	1	–	$AX, CY \leftarrow AX - (saddrp)$	x	x	x
		AX, !addr16	3	1	4	$AX, CY \leftarrow AX - (addr16)$	x	x	x
		AX, [HL+byte]	3	1	4	$AX, CY \leftarrow AX - (HL + byte)$	x	x	x
		AX, ES:!addr16	4	2	5	$AX, CY \leftarrow AX - (ES:addr16)$	x	x	x
		AX, ES: [HL+byte]	4	2	5	$AX, CY \leftarrow AX - ((ES:HL) + byte)$	x	x	x
	CMPW	AX, #word	3	1	–	$AX - word$	x	x	x
		AX, BC	1	1	–	$AX - BC$	x	x	x
		AX, DE	1	1	–	$AX - DE$	x	x	x
		AX, HL	1	1	–	$AX - HL$	x	x	x
		AX, saddrp	2	1	–	$AX - (saddrp)$	x	x	x
		AX, !addr16	3	1	4	$AX - (addr16)$	x	x	x
		AX, [HL+byte]	3	1	4	$AX - (HL + byte)$	x	x	x
		AX, ES:!addr16	4	2	5	$AX - (ES:addr16)$	x	x	x
		AX, ES: [HL+byte]	4	2	5	$AX - ((ES:HL) + byte)$	x	x	x
乘	MULU	X	1	1	–	$AX \leftarrow A \times X$			

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{CPU})的一个周期。
 2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (12/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
递增/递减	INC	r	1	1	—	$r \leftarrow r + 1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr) + 1$	×	×	
		laddr16	3	2	—	$(addr16) \leftarrow (addr16) + 1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte) + 1$	×	×	
		ES:laddr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) + 1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) + 1$	×	×	
	DEC	r	1	1	—	$r \leftarrow r - 1$	×	×	
		saddr	2	2	—	$(saddr) \leftarrow (saddr) - 1$	×	×	
		laddr16	3	2	—	$(addr16) \leftarrow (addr16) - 1$	×	×	
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte) - 1$	×	×	
		ES:laddr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) - 1$	×	×	
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) - 1$	×	×	
	INCW	rp	1	1	—	$rp \leftarrow rp + 1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp) + 1$			
		laddr16	3	2	—	$(addr16) \leftarrow (addr16) + 1$			
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte) + 1$			
		ES:laddr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) + 1$			
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) + 1$			
	DECW	rp	1	1	—	$rp \leftarrow rp - 1$			
		saddrp	2	2	—	$(saddrp) \leftarrow (saddrp) - 1$			
		laddr16	3	2	—	$(addr16) \leftarrow (addr16) - 1$			
		[HL+byte]	3	2	—	$(HL+byte) \leftarrow (HL+byte) - 1$			
		ES:laddr16	4	3	—	$(ES, addr16) \leftarrow (ES, addr16) - 1$			
		ES: [HL+byte]	4	3	—	$((ES:HL) + byte) \leftarrow ((ES:HL) + byte) - 1$			
移位	SHR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow 0) \times cnt$			×
	SHRW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow 0) \times cnt$			×
	SHL	A, cnt	2	1	—	$(CY \leftarrow A_7, A_m \leftarrow A_{m-1}, A_0 \leftarrow 0) \times cnt$			×
		B, cnt	2	1	—	$(CY \leftarrow B_7, B_m \leftarrow B_{m-1}, B_0 \leftarrow 0) \times cnt$			×
		C, cnt	2	1	—	$(CY \leftarrow C_7, C_m \leftarrow C_{m-1}, C_0 \leftarrow 0) \times cnt$			×
	SHLW	AX, cnt	2	1	—	$(CY \leftarrow AX_{15}, AX_m \leftarrow AX_{m-1}, AX_0 \leftarrow 0) \times cnt$			×
		BC, cnt	2	1	—	$(CY \leftarrow BC_{15}, BC_m \leftarrow BC_{m-1}, BC_0 \leftarrow 0) \times cnt$			×
	SAR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow A_7) \times cnt$			×
	SARW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow AX_{15}) \times cnt$			×

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(fCPU)的一个周期。
 2. 该时钟周期用于内部 ROM（Flash 存储器）程序。
 3. cnt 表示位移数。

表 25-5. 操作列表 (13/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
循环	ROR	A, 1	2	1	–	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			×
	ROL	A, 1	2	1	–	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			×
	RORC	A, 1	2	1	–	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			×
	ROLC	A, 1	2	1	–	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			×
	ROLWC	AX, 1	2	1	–	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			×
		BC, 1	2	1	–	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			×
位操作	MOV1	CY, saddr.bit	3	1	–	$CY \leftarrow (saddr).bit$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow sfr.bit$			×
		CY, A.bit	2	1	–	$CY \leftarrow A.bit$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow PSW.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow (HL).bit$			×
		saddr.bit, CY	3	2	–	$(saddr).bit \leftarrow CY$			
		sfr.bit, CY	3	2	–	$sfr.bit \leftarrow CY$			
		A.bit, CY	2	1	–	$A.bit \leftarrow CY$			
		PSW.bit, CY	3	4	–	$PSW.bit \leftarrow CY$	×	×	
		[HL].bit, CY	2	2	–	$(HL).bit \leftarrow CY$			
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow (ES, HL).bit$			×
		ES:[HL].bit, CY	3	3	–	$(ES, HL).bit \leftarrow CY$			
	AND1	CY, saddr.bit	3	1	–	$CY \leftarrow CY \wedge (saddr).bit$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \wedge sfr.bit$			×
		CY, A.bit	2	1	–	$CY \leftarrow CY \wedge A.bit$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \wedge PSW.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \wedge (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \wedge (ES, HL).bit$			×
	OR1	CY, saddr.bit	3	1	–	$CY \leftarrow CY \vee (saddr).bit$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \vee sfr.bit$			×
		CY, A.bit	2	1	–	$CY \leftarrow CY \vee A.bit$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \vee PSW.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \vee (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \vee (ES, HL).bit$			×

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{CPU})的一个周期。
 2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (14/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
位操作	XOR1	CY, saddr.bit	3	1	–	$CY \leftarrow CY \nabla (saddr).bit$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \nabla sfr.bit$			×
		CY, A.bit	2	1	–	$CY \leftarrow CY \nabla A.bit$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \nabla PSW.bit$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \nabla (HL).bit$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \nabla (ES, HL).bit$			×
	SET1	saddr.bit	3	2	–	$(saddr).bit \leftarrow 1$			
		sfr.bit	3	2	–	$sfr.bit \leftarrow 1$			
		A.bit	2	1	–	$A.bit \leftarrow 1$			
		!addr16.bit	4	2	–	$(addr16).bit \leftarrow 1$			
		PSW.bit	3	4	–	$PSW.bit \leftarrow 1$	×	×	×
		[HL].bit	2	2	–	$(HL).bit \leftarrow 1$			
		ES:!addr16.bit	5	3	–	$(ES, addr16).bit \leftarrow 1$			
		ES:[HL].bit	3	3	–	$(ES, HL).bit \leftarrow 1$			
	CLR1	saddr.bit	3	2	–	$(saddr).bit \leftarrow 0$			
		sfr.bit	3	2	–	$sfr.bit \leftarrow 0$			
		A.bit	2	1	–	$A.bit \leftarrow 0$			
		!addr16.bit	4	2	–	$(addr16).bit \leftarrow 0$			
		PSW.bit	3	4	–	$PSW.bit \leftarrow 0$	×	×	×
		[HL].bit	2	2	–	$(HL).bit \leftarrow 0$			
		ES:!addr16.bit	5	3	–	$(ES, addr16).bit \leftarrow 0$			
		ES:[HL].bit	3	3	–	$(ES, HL).bit \leftarrow 0$			
	SET1	CY	2	1	–	$CY \leftarrow 1$			1
	CLR1	CY	2	1	–	$CY \leftarrow 0$			0
	NOT1	CY	2	1	–	$CY \leftarrow \overline{CY}$			×

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。
 2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (15/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
调用/ 返回	CALL	rp	2	3	–	$(SP - 2) \leftarrow (PC + 2)_S, (SP - 3) \leftarrow (PC + 2)_H,$ $(SP - 4) \leftarrow (PC + 2)_L, PC \leftarrow CS, rp,$ $SP \leftarrow SP - 4$			
		\$!addr20	3	3	–	$(SP - 2) \leftarrow (PC + 3)_S, (SP - 3) \leftarrow (PC + 3)_H,$ $(SP - 4) \leftarrow (PC + 3)_L, PC \leftarrow PC + 3 +$ $jdisp16,$ $SP \leftarrow SP - 4$			
		!addr16	3	3	–	$(SP - 2) \leftarrow (PC + 3)_S, (SP - 3) \leftarrow (PC + 3)_H,$ $(SP - 4) \leftarrow (PC + 3)_L, PC \leftarrow 0000, addr16,$ $SP \leftarrow SP - 4$			
		!!addr20	4	3	–	$(SP - 2) \leftarrow (PC + 4)_S, (SP - 3) \leftarrow (PC + 4)_H,$ $(SP - 4) \leftarrow (PC + 4)_L, PC \leftarrow addr20,$ $SP \leftarrow SP - 4$			
	CALLT	[addr5]	2	5	–	$(SP - 2) \leftarrow (PC + 2)_S, (SP - 3) \leftarrow (PC + 2)_H,$ $(SP - 4) \leftarrow (PC + 2)_L, PC_S \leftarrow 0000,$ $PC_H \leftarrow (0000, addr5 + 1),$ $PC_L \leftarrow (0000, addr5),$ $SP \leftarrow SP - 4$			
	BRK	–	2	5	–	$(SP - 1) \leftarrow PSW, (SP - 2) \leftarrow (PC + 2)_S,$ $(SP - 3) \leftarrow (PC + 2)_H, (SP - 4) \leftarrow (PC + 2)_L,$ $PC_S \leftarrow 0000,$ $PC_H \leftarrow (0007FH), PC_L \leftarrow (0007EH),$ $SP \leftarrow SP - 4, IE \leftarrow 0$			
	RET	–	1	6	–	$PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1),$ $PC_S \leftarrow (SP + 2), SP \leftarrow SP + 4$			
	RETI	–	2	6	–	$PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1),$ $PC_S \leftarrow (SP + 2), PSW \leftarrow (SP + 3),$ $SP \leftarrow SP + 4$	R	R	R
	RETB	–	2	6	–	$PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1),$ $PC_S \leftarrow (SP + 2), PSW \leftarrow (SP + 3),$ $SP \leftarrow SP + 4$	R	R	R

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。
 2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (16/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
堆栈操作	PUSH	PSW	2	1	–	$(SP - 1) \leftarrow PSW, (SP - 2) \leftarrow 00H,$ $SP \leftarrow SP - 2$			
		rp	1	1	–	$(SP - 1) \leftarrow rp_H, (SP - 2) \leftarrow rp_L,$ $SP \leftarrow SP - 2$			
	POP	PSW	2	3	–	$PSW \leftarrow (SP + 1), SP \leftarrow SP + 2$	R	R	R
		rp	1	1	–	$rp_L \leftarrow (SP), rp_H \leftarrow (SP + 1), SP \leftarrow SP + 2$			
	MOVW	SP, #word	4	1	–	$SP \leftarrow word$			
		SP, AX	2	1	–	$SP \leftarrow AX$			
		AX, SP	2	1	–	$AX \leftarrow SP$			
		HL, SP	3	1	–	$HL \leftarrow SP$			
		BC, SP	3	1	–	$BC \leftarrow SP$			
		DE, SP	3	1	–	$DE \leftarrow SP$			
		ADDW SP, #byte	2	1	–	$SP \leftarrow SP + byte$			
		SUBW SP, #byte	2	1	–	$SP \leftarrow SP - byte$			
无条件转移	BR	AX	2	3	–	$PC \leftarrow CS, AX$			
		\$addr20	2	3	–	$PC \leftarrow PC + 2 + jdisp8$			
		\$!addr20	3	3	–	$PC \leftarrow PC + 3 + jdisp16$			
		!addr16	3	3	–	$PC \leftarrow 0000, addr16$			
		!!addr20	4	3	–	$PC \leftarrow addr20$			
条件转移	BC	\$addr20	2	2/4 ^{注3}	–	$PC \leftarrow PC + 2 + jdisp8$ if CY = 1			
	BNC	\$addr20	2	2/4 ^{注3}	–	$PC \leftarrow PC + 2 + jdisp8$ if CY = 0			
	BZ	\$addr20	2	2/4 ^{注3}	–	$PC \leftarrow PC + 2 + jdisp8$ if Z = 1			
	BNZ	\$addr20	2	2/4 ^{注3}	–	$PC \leftarrow PC + 2 + jdisp8$ if Z = 0			
	BH	\$addr20	3	2/4 ^{注3}	–	$PC \leftarrow PC + 3 + jdisp8$ if $(Z \vee CY) = 0$			
	BNH	\$addr20	3	2/4 ^{注3}	–	$PC \leftarrow PC + 3 + jdisp8$ if $(Z \vee CY) = 1$			
	BT	saddr.bit, \$addr20	4	3/5 ^{注3}	–	$PC \leftarrow PC + 4 + jdisp8$ if saddr.bit = 1			
		sfr.bit, \$addr20	4	3/5 ^{注3}	–	$PC \leftarrow PC + 4 + jdisp8$ if sfr.bit = 1			
		A.bit, \$addr20	3	3/5 ^{注3}	–	$PC \leftarrow PC + 3 + jdisp8$ if A.bit = 1			
		PSW.bit, \$addr20	4	3/5 ^{注3}	–	$PC \leftarrow PC + 4 + jdisp8$ if PSW.bit = 1			
		[HL].bit, \$addr20	3	3/5 ^{注3}	6/8	$PC \leftarrow PC + 3 + jdisp8$ if (HL).bit = 1			
		ES:[HL].bit, \$addr20	4	4/6 ^{注3}	7/9	$PC \leftarrow PC + 4 + jdisp8$ if (ES, HL).bit = 1			

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. 这表示“不满足条件/满足条件”的时钟数。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器（CKC）选择的 CPU 时钟(f_{cpu})的一个周期。
 2. 该时钟周期用于内部 ROM（Flash 存储器）程序。

表 25-5. 操作列表 (17/17)

指令组	助记符	操作数	字节数	时钟		操作	标志		
				注 1	注 2		Z	AC	CY
条件转移	BF	saddr.bit, \$addr20	4	3/5 ^{注3}	—	PC ← PC + 4 + jdisp8 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}	6/8	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, \$addr20	4	4/6 ^{注3}	7/9	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	5/7 ^{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			
条件跳转	SKC	—	2	1	—	如果 CY = 1 跳过下一条指令			
	SKNC	—	2	1	—	如果 CY = 0 跳过下一条指令			
	SKZ	—	2	1	—	如果 Z = 1 跳过下一条指令			
	SKNZ	—	2	1	—	如果 Z = 0 跳过下一条指令			
	SKH	—	2	1	—	如果 (Z ∨ CY) = 0 跳过下一条指令			
	SKNH	—	2	1	—	如果 (Z ∨ CY) = 1 跳过下一条指令			
CPU 控制	SEL	RBn	2	1	—	RBS[1:0] ← n			
	NOP	—	1	1	—	无操作			
	EI	—	3	4	—	IE ← 1 (允许中断)			
	DI	—	3	4	—	IE ← 0 (禁止中断)			
	HALT	—	2	3	—	设置 HALT 模式			
	STOP	—	2	3	—	设置 STOP 模式			

- 注
1. 当内部 RAM 区域或 SFR 区域被访问，或对于无数据访问的指令。
 2. 当程序存储器区域被访问时。
 3. 这表示“不满足条件/满足条件”的时钟数。

- 备注
1. 一个指令时钟周期是指由系统时钟控制寄存器 (CKC) 选择的 CPU 时钟(f_{cpu})的一个周期。
 2. 该时钟周期用于内部 ROM (Flash 存储器) 程序。
 3. n 表示寄存器 bank 的序号 (n = 0~3)。

第 26 章 电气特性（目标值）

注意事项 1. 这些规范所示为目标值，设备评估后可能改变。

2. μ PD79F9211 具有片上调试功能，可用于开发和评价。在量产时不要使用产品中的片上调试功能，因为 flash 存储器中保证的可重写次数可能会超出，从而不能保证产品的可靠性。对于因使用片上调试功能而产生的问题，NEC 电子不承担任何责任。

最大额定值 ($T_A = 25^\circ\text{C}$) (1/2)

参数	符号	条件	额定值	单位
供电电压	V_{DD}		-0.5 至 +6.5	V
	V_{SS}		-0.5 至 +0.3	V
	AV_{REF}		-0.5 至 $V_{DD} + 0.3$ ^{註1}	V
	AV_{SS}		-0.5 至 +0.3	V
REGC 引脚输入电压	V_{IREGC}	REGC	-0.3~3.6 和 -0.3~ $V_{DD} + 0.3$ ^{註2}	V
输入电压	V_{I1}	P10 至 P13、P30 至 P32、P40、P41、P50 至 P52、 P70 至 P75、P120 至 P124、 $\overline{\text{RESET}}$ 、FLMD0	-0.3 至 $V_{DD} + 0.3$ ^{註1}	V
	V_{I3}	P20~P27、P80~P83、P150~P151	-0.3 至 $AV_{REF} + 0.3$ ^{註1} 和 -0.3 至 $V_{DD} + 0.3$ ^{註1}	V
输出电压	V_{O1}	P10~P13、P30 ~ P32、P40、P41、P50 ~ P52、P70 ~ P75、P120	-0.3 至 $V_{DD} + 0.3$ ^{註1}	V
	V_{O2}	P20~P27、P80~P83、P150~P151	-0.3 至 $AV_{REF} + 0.3$	V

注 1. 必须小于等于 6.5 V。

2. 通过一个电容（0.47 ~ 1 mF：推荐）连接 REGC 引脚和 V_{SS} 。该值控制 REGC 引脚的绝对最大额定值。不要将此电压用于这个引脚。

注意事项 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。

备注 除非另外说明，复用功能引脚的特性与端口引脚相同。

最大额定值 ($T_A = ^\circ\text{C}$) (2/2)

参数	符号	条件		额定值	单位
模拟 输入电压	V _{AN}	ANI0~ANI9、PGAI、CMP0M、CMP0P、CMP1M、CMP1P		-0.3 至 AV _{REF} +0.3 ^註 和 -0.3 至 V _{DD} +0.3 ^註	V
输出电流,高	I _{OH1}	每个引脚	P10~P13、P30 ~ P32、P40、P41、P50 ~ P52、P70 ~ P75、P120	-10	mA
		所有引脚总和 -80 mA	P40、P41、P120	-25	mA
			P10~P13、P30~P32、P50~P52、P70~P75	-55	mA
	I _{OH2}	每个引脚	P20~P27、P80~P83、P150~P151	-0.5	mA
		所有引脚总和		-2	mA
	输出电流,低	I _{OL1}	每个引脚	P10~P13、P30 ~ P32、P40、P41、P50 ~ P52、P70 ~ P75、P120	30
所有引脚总和 200 mA			P40、P41、P120	60	mA
			P10~P13、P30~P32、P50~P52、P70~P75	140	mA
I _{OL2}		每个引脚	P20~P27、P80~P83、P150~P151	1	mA
		所有引脚总和		5	mA
工作环境温度		T _A	在普通操作模式下		-40 至 +85
	在 Flash 存储器编程模式下				
存储温度	T _{stg}			-40 至 +125	°C

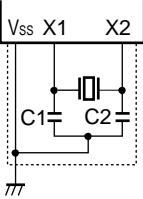
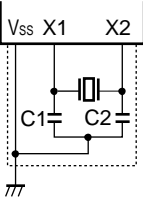
注 必须小于等于 6.5 V。

注意事项 任何一项参数哪怕是在瞬间超过最大额定值，都会使产品质量受到影响。也就是说，最大额定值是产品濒临物理损坏的临界点，因而，必须保证产品在不超过最大额定值的条件下使用。

备注 除非另外说明，复用功能引脚的特性与端口引脚相同。

X1 振荡器特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = AV_{SS} = 0\text{ V}$)

振荡器	建议使用的电路	参数	条件	最小值	典型值	最大值	单位
陶瓷振荡器		X1 时钟振荡频率 (f_x) ^註	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		20.0	MHz
晶体振荡器		X1 时钟振荡频率 (f_x) ^註	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		20.0	MHz

注 仅表示振荡器的特性。如需了解指令执行时间，可参见 AC 特性。

注意事项 1. 在使用 X1 时钟时，上图中被虚线包围的部分的配线应按照如下布线方法布线，以防止连接电容时产生不利影响。

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

2. 由于复位后 CPU 使用内部高速振荡时钟进行操作，因此用户要使用振荡稳定时间计数器的状态寄存器（OSTC）检测 X1 时钟振荡稳定时间。在充分评估了所使用的振荡器的振荡稳定时间后，再确定 OSTC 和振荡稳定时间选择寄存器（OSTS）的振荡稳定时间。

备注 对于振荡器选择和振荡常数，用户需要自己评价振荡特性或请振荡器厂商评价。

内部振荡器特性
($T_A = -40$ 至 $+85^{\circ}\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = AV_{SS} = 0\text{ V}$)

<R>

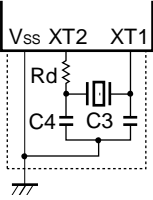
振荡器	参数	条件		最小值	典型值	最大值	单位
内部高速振荡时钟频率 ^{註1}	f _{IH}	2.7 V ≤ V _{DD} ≤ 5.5 V			8		MHz
	f _{DSC}	2.7 V ≤ V _{DD} ≤ 5.5 V			40		MHz
内部低速振荡时钟频率	f _{IL}	普通电流模式	2.7 V ≤ V _{DD} ≤ 5.5 V	27	30	33	kHz
		低功耗电流模式 ^{註2}		15	30	45	kHz

- 注
1. 这只显示了振荡器特性。如需了解指令执行时间，可参见 AC 特性。
 2. 在如下情况下校准器输出设置为低功耗电流模式：
 - 当 RMC 寄存器设置为 5AH。
 - RESET 引脚复位期间。
 - 在 STOP 模式下 (除在 OCD 模式期间外)。
 - CPU 操作使用副系统时钟 (f_{XT}) 期间，高速系统时钟 (f_{MX})、高速内部振荡时钟 (f_{IH})和 40 MHz 内部高速振荡时钟 (f_{DSC}) 都停止时。
 - HALT 模式下当 CPU 操作设置为使用副系统时钟(f_{XT}) 期间，当高速系统时钟 (f_{MX})、高速内部振荡时钟 (f_{IH})和 40 MHz 内部高速振荡时钟(f_{DSC})都停止时。

备注 正常电流模式和依照校准器输出电压的低功耗电流模式的详细情况，参见第 20 章 稳压器。

XT1 振荡器特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = AV_{SS} = 0\text{ V}$)

振荡器	建议使用的电路	项目	条件	最小值	典型值	最大值	单位
晶体振荡器		XT1 时钟振荡频率 (f_{XT}) [※]		32	32.768	35	kHz

注 仅表示振荡器的特性。如需了解指令执行时间，可参见 AC 特性。

注意事项 1. 在使用 XT1 时钟时，上图中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接电容时产生不利影响。

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

2. XT1 振荡器是一个低振幅电路，用于降低功耗，但比 X1 振荡器更容易受到噪音干扰。因此在使用 XT1 时钟时应特别注意连线方式。

备注 对于振荡器选择和振荡常数，用户需要自己评价振荡特性或请振荡器厂商评价。

DC 特性 (1/8)

(TA = -40 至+85°C, 2.7 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ AVREF ≤ VDD, VSS = AVSS = 0 V)

项目	符号	条件		最小值	典型值	最大值	单位
输出电流, 高 ^{註1}	IOH1	P10~P13、P30 ~ P32、P40、P41、P50 ~ P52、P70 ~ P75、P120 的每个引脚	4.0 V ≤ VDD ≤ 5.5 V			-3.0	mA
			2.7 V ≤ VDD < 4.0 V			-1.0	mA
		P40、P41、P120 的总和 (当占空比 = 70% ^{註2})	4.0 V ≤ VDD ≤ 5.5 V			-20.0	mA
			2.7 V ≤ VDD < 4.0 V			-10.0	mA
		P10~P13、P30~P32、P50~P52、P70~P75 的总和 (当占空比 = 70% ^{註2})	4.0 V ≤ VDD ≤ 5.5 V			-30.0	mA
			2.7 V ≤ VDD < 4.0 V			-19.0	mA
	IOH2	所有引脚总和 (当占空比 = 60% ^{註2})	4.0 V ≤ VDD ≤ 5.5 V			-50.0	mA
			2.7 V ≤ VDD < 4.0 V			-29.0	mA
		对 P20~P27、P80~P83、P150~P151 的每个引脚	AVREF ≤ VDD			-0.1	mA

注 1. 即使电流从 VDD 引脚流向输出引脚，设备工作时的电流值也可以得到保证。

2. 占空比为 60% 或 70%的条件下指定。

可通过以下表达式计算已经改变占空比的输出电流值（当占空比从 70%改变为 n%）。

 •引脚的总输出电流 = (IOH × 0.7)/(n × 0.01)

 <举例> n = 50% 并且 IOH = 20.0 mA

 引脚的总输出电流 = (-20.0 × 0.7)/(50 × 0.01) = -28.0 mA

但是，允许流入每个引脚的电流不会随占空比而变化。高于最大额定值的电流不能流入引脚。

注意事项 在 N 沟道漏极开路模式中，P30~P32、P70、P72、P73 和 P75 不输出高电平。

备注 除非另外说明，复用功能引脚的特性与端口引脚相同。

DC 特性 (2/8)

(T_A = -40 至 +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

项目	符号	条件	最小值	典型值	最大值	单位
输出电流, 低 ^{註1}	I _{OL1}	P30 ~ P32、P40、P41、P52、P70 ~ P75、P120 的每个引脚	4.0 V ≤ V _{DD} ≤ 5.5 V		8.5	mA
			2.7 V ≤ V _{DD} < 4.0 V		1.0	mA
		P10 至 P13、P50、P51 的每个引脚	4.0 V ≤ V _{DD} ≤ 5.5 V		10.0	mA
			2.7 V ≤ V _{DD} < 4.0 V		1.0	mA
		P40、P41、P120 的总和 (当占空比 = 70% ^{註2})	4.0 V ≤ V _{DD} ≤ 5.5 V		20.0	mA
			2.7 V ≤ V _{DD} < 4.0 V		15.0	mA
		P10~P13、P30 ~ P32、P50 ~ P52、P70 ~ P75 的所有引脚总和 (当占空比 = 70% ^{註2})	4.0 V ≤ V _{DD} ≤ 5.5 V		45.0	mA
			2.7 V ≤ V _{DD} < 4.0 V		35.0	mA
		所有引脚总和 (当占空比 = 70% ^{註2})	4.0 V ≤ V _{DD} ≤ 5.5 V		65.0	mA
			2.7 V ≤ V _{DD} < 4.0 V		40.0	mA
	I _{OL2}	对 P20~P27、P80~P83、P150~P151 的每个引脚	AV _{REF} ≤ V _{DD}		0.4	mA

注 1. 即使电流从输出引脚流向 V_{SS} 引脚和 AV_{SS} 引脚, 器件工作时的电流值也可以得到保证。

2. 占空比为 60% 或 70% 的条件下指定。

可通过以下表达式计算已经改变占空比的输出电流值 (当占空比从 70% 改变为 n%)。

$$\bullet \text{ 引脚的总输出电流} = (I_{OL} \times 0.7) / (n \times 0.01)$$

$$<\text{举例}> n = 50\% \text{ 并且 } I_{OL} = 20.0 \text{ mA}$$

$$\text{引脚的总输出电流} = (20.0 \times 0.7) / (50 \times 0.01) = 28.0 \text{ mA}$$

但是, 允许流入每个引脚的电流不会随占空比而变化。高于最大额定值的电流不能流入引脚。

备注 除非另外说明, 复用功能引脚的特性与端口引脚相同。

DC 特性 (3/8)

(T_A = -40 至 +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

项目	符号	条件		最小值	典型值	最大值	单位
输入电压, 高	V _{IH1}	P30、P123、P124		0.7V _{DD}		V _{DD}	V
	V _{IH2}	P10 ~ P13、P31、P32、P40、P41、P50 ~ P52、P70 ~ P75、P120~P122、EXCLK、 $\overline{\text{RESET}}$	普通输入缓冲器	0.8V _{DD}		V _{DD}	V
	V _{IH3}	P31, P32, P71, P72, P74, P75	TTL 输入缓冲器 4.0 V ≤ V _{DD} ≤ 5.5 V	2.2		V _{DD}	V
			TTL 输入缓冲器 2.7 V ≤ V _{DD} < 4.0 V	2.0		V _{DD}	V
	V _{IH4}	P20~P27、P81、P83、P150~P151	2.7 V ≤ AV _{REF} ≤ V _{DD}	0.7AV _{REF}		AV _{REF}	V
	V _{IH5}	P80, P82	2.7 V ≤ AV _{REF} ≤ V _{DD}	0.8AV _{REF}		AV _{REF}	V
	V _{IH7}	FLMD0		0.9V _{DD} 注		V _{DD}	V

注 当用在 Flash 存储器编程模式下时, 必须大于等于 0.9V_{DD}。

注意事项 即使在 N 沟道漏极开路模式下, 引脚 P30~P32、P70、P72、P73 和 P75 的 V_{IH} 的最大值也是 V_{DD}。

备注 除非另外说明, 复用功能引脚的特性与端口引脚相同。

DC 特性 (4/8)**($T_A = -40$ 至 $+85^{\circ}\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = AV_{SS} = 0\text{ V}$)**

项目	符号	条件		最小值	典型值	最大值	单位
输入电压, 低	V_{IL1}	P30、P123、P124		0		$0.3V_{DD}$	V
	V_{IL2}	P10 ~ P13、P31、P32、P40、P41、P50 ~ P52、P70 ~ P75、P120~P122、EXCLK、RESET	普通输入缓冲器	0		$0.2V_{DD}$	V
	V_{IL3}	P31, P32, P71, P72, P74, P75	TTL 输入缓冲器 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0		0.8	V
			TTL 输入缓冲器 $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	0		0.5	V
	V_{IL4}	P20~P27、P81、P83、P150~P151	$2.7\text{ V} \leq AV_{REF} \leq V_{DD}$	0		$0.3AV_{REF}$	V
	V_{IL5}	P80, P82	$2.7\text{ V} \leq AV_{REF} \leq V_{DD}$	0		$0.2AV_{REF}$	V
	V_{IL7}	FLMD0 [※]		0		$0.1V_{DD}$	V

注 当禁止 Flash 存储器写入时, 将 FLMD0 引脚直接连接到 V_{SS} , 并且保持电压小于 $0.1V_{DD}$ 。

备注 除非另外说明, 复用功能引脚的特性与端口引脚相同。

DC 特性 (5/8)**($T_A = -40$ 至 $+85^\circ\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = AV_{SS} = 0\text{ V}$)**

项目	符号	条件	最小值	典型值	最大值	单位
输出电压, 高	V _{OH1}	P10~P13、P30 ~ P32、P40、 P41、P50 ~ P52、P70 ~ P75、 P120	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OH1} = -3.0\text{ mA}$	$V_{DD} - 0.7$		V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OH1} = -1.0\text{ mA}$	$V_{DD} - 0.5$		V
	V _{OH2}	P20~P27、P80~P83、P150~P152	$AV_{REF} \leq V_{DD}$, $I_{OH2} = -0.1\text{ mA}$	$AV_{REF} - 0.5$		V
输出电压, 低	V _{OL1}	P10~P13、P30 ~ P32、P40、 P41、P50 ~ P52、P70 ~ P75、 P120	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 8.5\text{ mA}$		0.7	V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 1.0\text{ mA}$		0.5	V
	V _{OL2}	P10 至 P13、P50、P51	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 10.0\text{ mA}$		1.4	V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $I_{OL1} = 1.0\text{ mA}$		0.5	V
	V _{OL3}	P20~P27、P80~P83、P150~P151	$AV_{REF} = V_{DD}$, $I_{OL2} = 0.4\text{ mA}$		0.4	V

注意事项 在 N 沟道漏极开路模式中, P30~P32、P70、P72、P73 和 P75 不输出高电平。

备注 除非另外说明, 复用功能引脚的特性与端口引脚相同。

DC 特性 (6/8)

(T_A = -40 至 +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

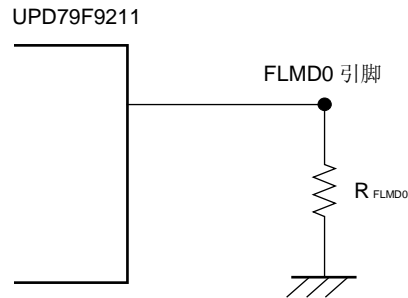
项目	符号	条件	最小值	典型值	最大值	单位
输入漏电流, 高	I _{LIH1}	P10~P13、P30 ~ P32、P40、P41、P50 ~ P52、P70 ~ P75、P120、FLMD0、RESET	V _I = V _{DD}		1	μA
	I _{LIH2}	P20~P27、P80~P83、P150~P151	V _I = AV _{REF} , AV _{REF} = V _{DD}		1	μA
	I _{LIH3}	P121~P124 (X1、X2、XT1、XT2)	V _I = V _{DD}	在输入端口	1	μA
				在振荡器连接	10	μA
输入漏电流, 低	I _{LIL1}	P10~P13、P30 ~ P32、P40、P41、P50 ~ P52、P70 ~ P75、P120、FLMD0、RESET	V _I = V _{SS}		-1	μA
	I _{LIL2}	P20~P27、P80~P83、P150~P151	V _I = V _{SS} , AV _{REF} = V _{DD}		-1	μA
	I _{LIL3}	P121~P124 (X1、X2、XT1、XT2)	V _I = V _{SS}	在输入端口	-1	μA
				在振荡器连接	-10	μA

备注 除非另外说明，复用功能引脚的特性与端口引脚相同。

DC 特性 (7/8)
($T_A = -40$ 至 $+85^{\circ}\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = AV_{SS} = 0\text{ V}$)

项目	符号	条件	最小值	典型值	最大值	单位
片内上拉电阻	R_U	P10~P13、P30 ~ P32、P40、P41、P50 ~ P52、P70 ~ P75、P120、P141 $V_i = V_{SS}$, 在输入端口	10	20	100	$k\Omega$
FLMD0 引脚外部下拉电阻 ^注	R_{FLMD0}	当使用软件允许自编程模式设置时	100			$k\Omega$

注 建议 FLMD0 引脚悬空。如果引脚需要外部下拉，设置 R_{FLMD0} 为 100 $k\Omega$ 或以上。



备注 除非另外说明，复用功能引脚的特性与端口引脚相同。

DC 特性 (8/8)

(T_A = -40 至 +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

参数	符号	条件			最小值	典型值	最大值	单位
供电电流	I _{DD1}	操作模式				T.B.D	T.B.D	mA
	I _{DD2}	HALT 模式				T.B.D	T.B.D	mA
	I _{DD3}	STOP 模式				T.B.D	T.B.D	μA
RTC 工作电流	I _{RTC} ^{注 1, 2}	f _{SUB} = 32.768 kHz		V _{DD} = 3.0 V		0.2	1.0	μA
				V _{DD} = 2.0 V		0.2	1.0	μA
看门狗定时器工作电流	I _{WDT} ^{注 2, 3}	f _{IL} = 30 kHz				5	10	μA
A/D 转换器工作电流	I _{ADC} ^{注 4}	以最大速度转换期间，	高速模式 1	AV _{REF} = V _{DD} = 5.0 V		T.B.D	T.B.D	mA
			高速模式 2	AV _{REF} = V _{DD} = 3.0 V		T.B.D	T.B.D	mA
			普通模式	AV _{REF} = V _{DD} = 5.0 V		T.B.D	T.B.D	mA
可编程增益放大器工作电流	I _{AMP} ^{注 5}					T.B.D	T.B.D	μA
比较器工作电流	I _{CMP} ^{注 6}	当不使用内部参考电压时的每个通道	AV _{REF} = V _{DD} = 5.0 V			T.B.D	T.B.D	μA
			AV _{REF} = V _{DD} = 3.0 V			T.B.D	T.B.D	μA
		当使用内部参考电压时的每个通道	AV _{REF} = V _{DD} = 5.0 V			T.B.D	T.B.D	μA
			AV _{REF} = V _{DD} = 3.0 V			T.B.D	T.B.D	μA
LVI 工作电流	I _{LVI} ^{注 7}					9	18	μA

- <R> 注 1. 电流只经过实时计数器 (XT1 振荡器的工作电流除外)。当实时计数器操作在操作模式或 HALT 模式下, μPD79F9211 电流的典型值, 是 I_{DD1} 或 I_{DD2} 的典型值和 I_{RTC} 的典型值之和。I_{DD1} 和 I_{DD2} 最大值也包括实时计数器工作电流。当实时计数器运行在 f_{CLK} = f_{SUB}/2 期间, I_{DD2} 的典型值包括实时计数器工作电流。
2. 当内部高速振荡器和高速系统时钟停止时。
3. 电流只经过看门狗定时器 (包括 30 kHz 内部振荡器的工作电流)。当看门狗定时器运行在停止模式时, μPD79F9211 的电流值是 f_{CLK} = f_{SUB}/2 时 I_{DD1}、I_{DD2} 或 I_{DD3}、与 I_{WDT} 的和。
4. 电流只流经 A/D 转换器 (AV_{REF} 引脚)。当 A/D 转换器在工作模式或 HALT 模式时, μPD79F9211 的电流值是 I_{DD1} 或 I_{DD2} 和 I_{ADC} 的总和。
5. 电流只流经可编程增益放大器 (AV_{REF} 引脚)。当可编程增益放大器运行在工作模式或 HALT 模式时, μPD79F9211 的电流值是 I_{DD1} 或 I_{DD2} 与 I_{AMP} 的和。
6. 电流只流经运算比较器 (AV_{REF} 引脚)。当比较器运行在工作模式或 HALT 模式时, μPD79F9211 的电流值是 I_{DD1} 或 I_{DD2} 与 I_{CMP} 的和。
7. 电流只经过 LVI 电路。当 LVI 电路在工作中, HALT 或 STOP 模式时, μPD79F9211 的电流值是 I_{DD1}、I_{DD2} 或 I_{DD3} 和 I_{LVI} 的总和。

- 备注 1. f_{IL}: 内部低速振荡时钟频率
2. f_{SUB}: 副系统时钟频率 (XT1 时钟振荡频率)
3. f_{CLK}: CPU/外围硬件时钟频率

AC 特性

(1) 基本操作(1/6)

(TA = -40 至+85°C, 2.7 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ AVREF ≤ VDD, VSS = AVSS = 0 V)

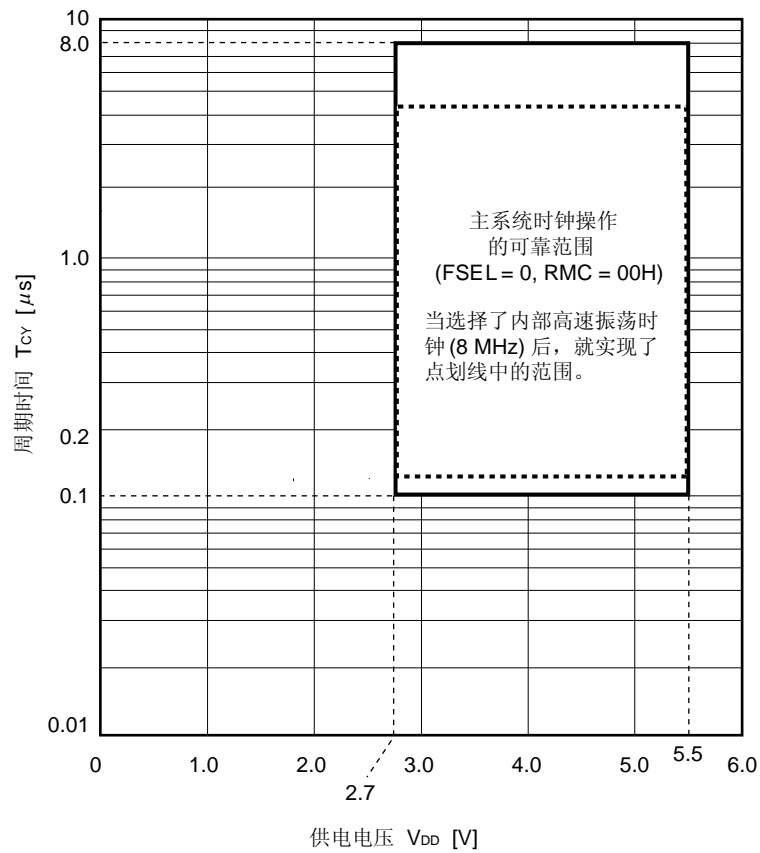
项目	符号	条件	最小值	典型值	最大值	单位
指令周期（最短指令执行时间）	TCY	主系统时钟 (fMAIN)操作	2.7 V ≤ VDD ≤ 5.5 V	0.05		8 μs
		低功耗电流模式		1		8 μs
		副系统时钟 (fSUB) 操作		57.2	61	62.5 μs
		在自编程模式下	普通电流模式 2.7 V ≤ VDD ≤ 5.5 V	0.05		0.5 μs
外部主系统时钟频率	fEX	2.7 V ≤ VDD ≤ 5.5 V	2.0		20.0	MHz
外部主系统时钟输入高电平宽度，低电平宽度	tEXH, tEXL	2.7 V ≤ VDD ≤ 5.5 V	24			ns
TI02 ~ TI07、TI09~TI11、SLTI 输入高电平宽度、低电平宽度	tTIH, tTIL	fMCK = fIH40	2/fMCK+10			ns
		其它值	1/fMCK+10			ns
TO02~TO07、TO10、TO11、SLTO 输出频率	fTO	2.7 V ≤ VDD ≤ 5.5 V			10	MHz
PCLBUZ0 输出频率	fPCL	2.7 V ≤ VDD ≤ 5.5 V			10	MHz
中断输入高电平宽度，低电平宽度	tINTH, tINTL		1			μs
RESET 低电平宽度	tRSL		10			μs

备注

1. fMCK: 宏工作时钟频率
(由 TMRn 寄存器的 CKSn 位设置的工作时钟。n:通道编号 (n = 00 ~ 11))
2. 关于稳压器的正常电流模式和低功耗电流模式的详细信息，请参考第 20 章 稳压器。

(1) 基本操作(2/6)

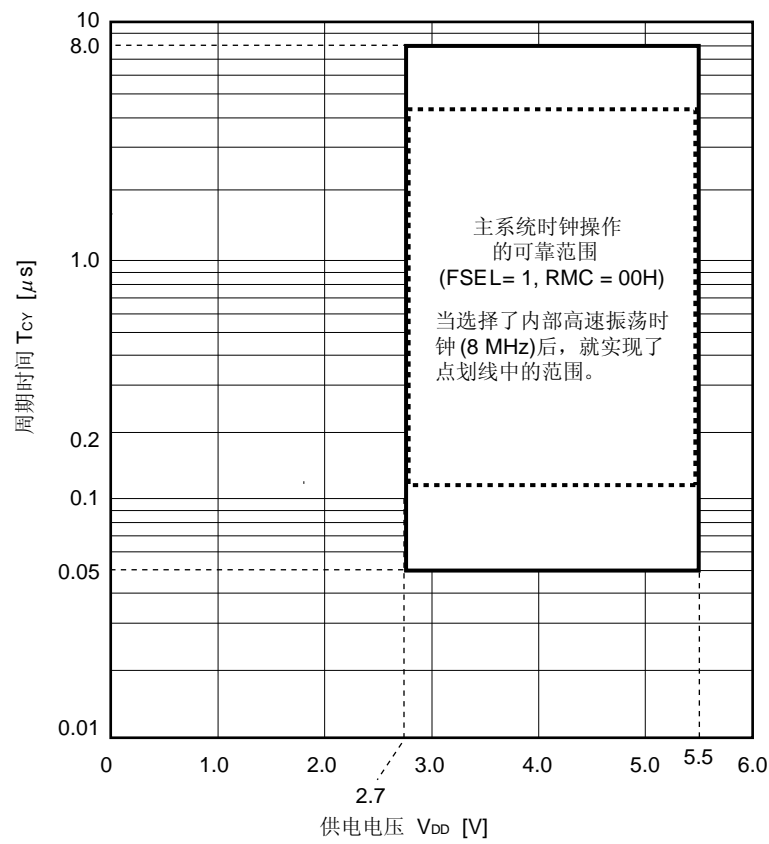
在主系统时钟工作期间的指令最短执行时间 (FSEL = 0, RMC = 00H)



备注 FSEL: 工作速度模式控制寄存器 (OSMC) 的位 0。

(1) 基本操作(3/6)

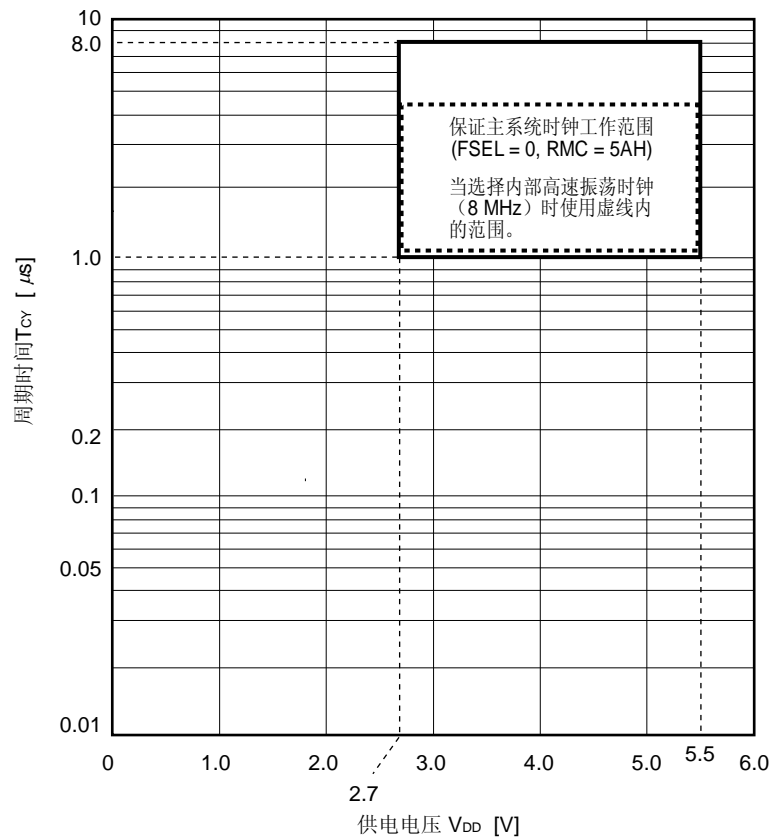
在主系统时钟工作期间的指令最短执行时间（FSEL = 1, RMC = 00H）



备注 FSEL: 工作速度模式控制寄存器 (OSMC) 的位 0。

(1) 基本操作(4/6)

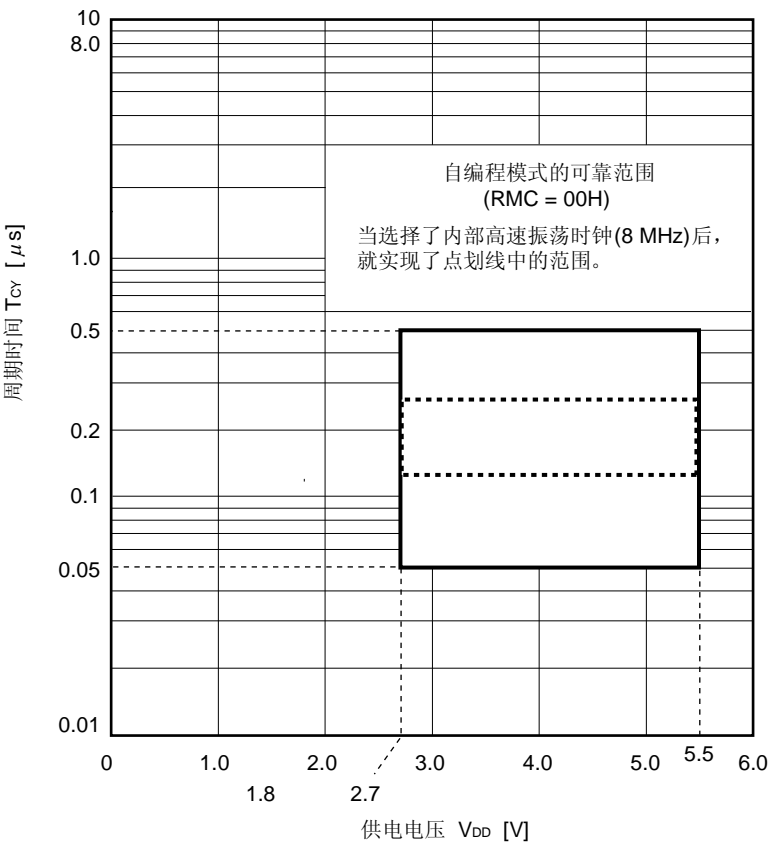
<R> 在主系统时钟工作期间的指令最短执行时间 (FSEL = 0, RMC = 5AH)



- 备注
1. FSEL: 工作速度模式控制寄存器 (OSMC) 的位 0。
 2. 当 RMC 被设置为 5AH 时, 在整个电压范围是 5MHz (最大值)。

(1) 基本操作(5/6)

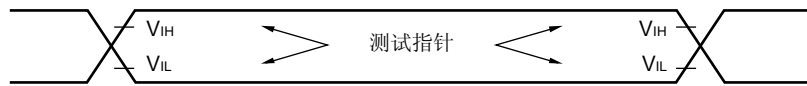
在自编程模式下的指令最短执行时间（RMC = 00H）



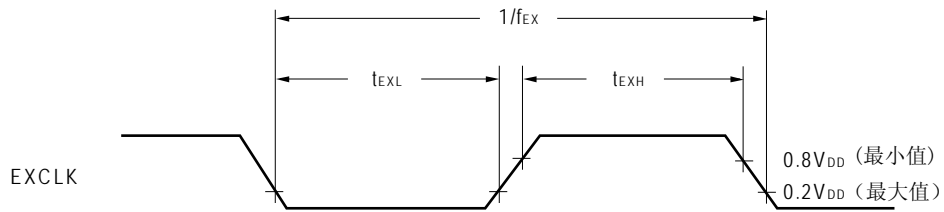
备注 当 RMC 被设置为 5AH 或者 CPU 使用副系统时钟时，自编程功能无法使用。

(1) 基本操作(6/6)

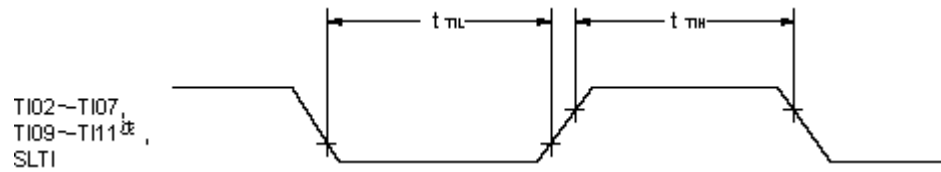
AC 时序测试点



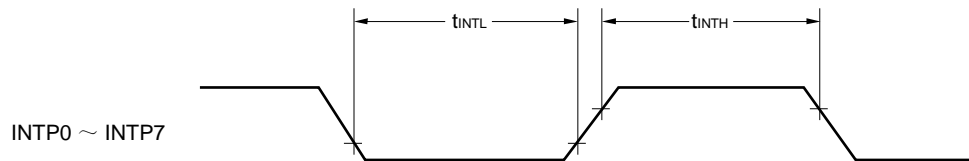
外部主系统时钟时序



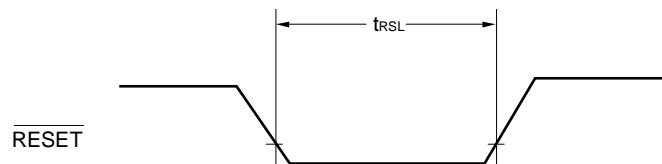
TI 时序



中断请求输入时序



RESET 输入时序

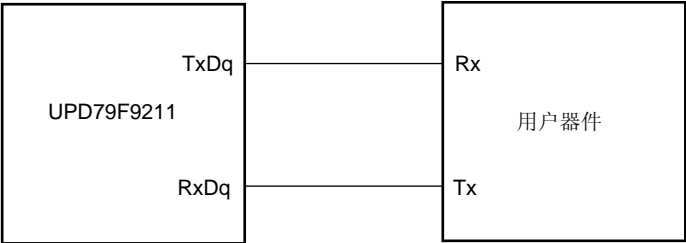


(2) 串行接口：串行阵列单元(1/17)
(TA = -40 至+85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

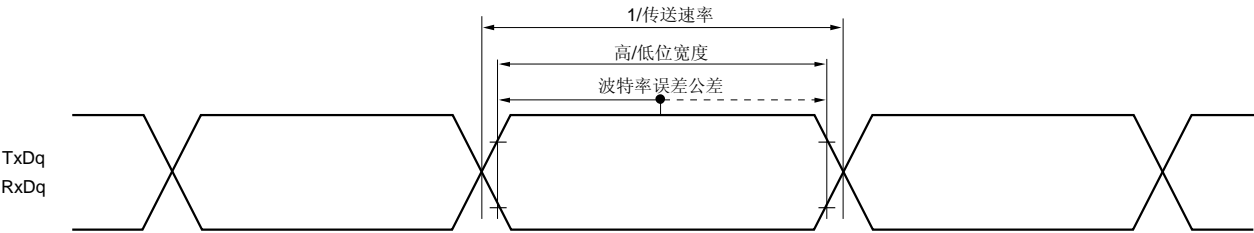
(a) 相同电平下通信（UART 模式）（专用波特率发生器输出）

参数	符号	条件	最小值	典型值	最大值	单位
传送速率					fMCK/6	Mbps
		fCLK = 20 MHz, fMCK = fCLK			3.3	Mbps

UART 模式连接图（相同电平下通信）



UART 模式位宽度（相同电平下通信）（参考）



注意事项 使用 PIMg 和 POMg 寄存器，为 RxDq 选择正常输入缓冲器、为 TxDq 选择正常输出模式。

备注

1. q: UART 编号 (q = 0, 1), g: PIM 和 POM 编号 (g = 3, 7)
2. fMCK: 串行阵列单元工作时钟频率
(由 SMR0n 寄存器的 CKS0n 位设置的工作时钟。n: 通道编号 (n=0~3))

(2) 串行接口：串行阵列单元(2/17)

(TA = -40 至 +85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

(b) 相同电平下通信 (CSI 模式) (主设备模式、 $\overline{\text{SCKp}}$... 内部时钟输出)

参数	符号	条件	最小值	典型值	最大值	单位
$\overline{\text{SCKp}}$ 周期时间	t_{KCY1}	4.0 V ≤ VDD ≤ 5.5 V	200			ns
		2.7 V ≤ VDD < 4.0 V	400			ns
$\overline{\text{SCKp}}$ 高/低电平宽度	$t_{\text{KH1}}, t_{\text{KL1}}$	4.0 V ≤ VDD ≤ 5.5 V	$t_{\text{KCY1}}/2 - 20$			ns
		2.7 V ≤ VDD < 4.0 V	$t_{\text{KCY1}}/2 - 35$			ns
Slp 设置时间 (到 $\overline{\text{SCKp}}\uparrow$) ^{注 1}	t_{SIK1}	4.0 V ≤ VDD ≤ 5.5 V	70			ns
		2.7 V ≤ VDD < 4.0 V	100			ns
Slp 保持时间 (从 $\overline{\text{SCKp}}\uparrow$) ^{注 2}	t_{KSI1}		30			ns
从 $\overline{\text{SCKp}}\downarrow$ 到 SOp 输出的延迟时间 ^{注 3}	t_{KSO1}	C = 50 pF ^{注 4}			40	ns

- 注
1. 当 DAP0n = 0 和 CKP0n = 0, 或 DAP0n = 1 和 CKP0n = 1 时。Slp 设置时间变为“至 $\overline{\text{SCKp}}\downarrow$ ”, 当 DAP0n = 0 和 CKP0n = 1, 或 DAP0n = 1 和 CKP0n = 0。
 2. 当 DAP0n = 0 和 CKP0n = 0, 或 DAP0n = 1 和 CKP0n = 1 时。Slp 保持时间变为“自 $\overline{\text{SCKp}}\downarrow$ ”, 当 DAP0n = 0 和 CKP0n = 1, 或 DAP0n = 1 和 CKP0n = 0。
 3. 当 DAP0n = 0 和 CKP0n = 0, 或 DAP0n = 1 和 CKP0n = 1 时。SOp 输出的延迟时间变为“自 $\overline{\text{SCKp}}\uparrow$ ”, 当 DAP0n = 0 和 CKP0n = 1, 或 DAP0n = 1 和 CKP0n = 0。
 4. C 是 $\overline{\text{SCKp}}$ 和 SOp 输出线的负载电容。

注意事项 使用 PIMg 和 POMg 寄存器, 为 Slp 选择正常输入缓冲器、为 SOp 和 $\overline{\text{SCKp}}$ 选择正常输出模式。

- 备注
1. p: CSI 编号(p = 00、01、10)
g: PIM 和 POM 编号 (g = 3、7)
 2. n: 单元编号 (n = 0~2)

(2) 串行接口：串行阵列单元(3/17)

(TA = -40 至+85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

(c) 相同电平下通信 (CSI 模式) (从设备模式、 $\overline{\text{SCKp}}$... 外部时钟输入) 中

参数	符号	条件	最小值	典型值	最大值	单位
$\overline{\text{SCKp}}$ 周期时间	t _{KCY2}	16 MHz < f _{MCK}	8/f _{MCK}			ns
		f _{MCK} ≤ 16 MHz	6/f _{MCK}			ns
$\overline{\text{SCKp}}$ 高/低电平宽度	t _{KH2} , t _{KL2}		f _{KCY2} /2			ns
Slp 设置时间 (到 $\overline{\text{SCKp}}\uparrow$) ^{註 1}	t _{SIK2}		1/f _{MCK} +80			ns
Slp 保持时间 (从 $\overline{\text{SCKp}}\uparrow$) ^{註 2}	t _{KS2}		50			ns
从 $\overline{\text{SCKp}}\downarrow$ 到 SOp 输出的延迟时间 ^{註 3}	t _{KSO2}	C = 50 pF ^{註 4}	4.0 V ≤ V _{DD} ≤ 5.5 V		1/f _{MCK} +120	ns
			2.7 V ≤ V _{DD} < 4.0 V		1/f _{MCK} +120	ns

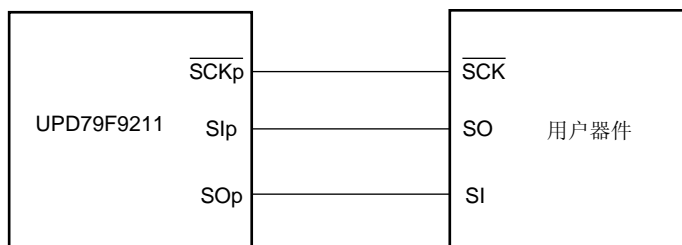
- 注
1. 当 DAP0n = 0 和 CKP0n = 0, 或 DAP0n = 1 和 CKP0n = 1 时。Slp 设置时间变为“至 $\overline{\text{SCKp}}\downarrow$ ”, 当 DAP0n = 0 和 CKP0n = 1, 或 DAP0n = 1 和 CKP0n = 0。
 2. 当 DAP0n = 0 和 CKP0n = 0, 或 DAP0n = 1 和 CKP0n = 1 时。Slp 保持时间变为“自 $\overline{\text{SCKp}}\downarrow$ ”, 当 DAP0n = 0 和 CKP0n = 1, 或 DAP0n = 1 和 CKP0n = 0。
 3. 当 DAP0n = 0 和 CKP0n = 0, 或 DAP0n = 1 和 CKP0n = 1 时。SOp 输出的延迟时间变为“自 $\overline{\text{SCKp}}\uparrow$ ”, 当 DAP0n = 0 和 CKP0n = 1, 或 DAP0n = 1 和 CKP0n = 0。
 4. C 是 $\overline{\text{SCKp}}$ 和 SOp 输出线的负载电容。

注意事项 使用 PIMg 和 POMg 寄存器, 为 Slp 和 $\overline{\text{SCKp}}$ 选择正常输入缓冲器、为 SOp 选择正常输出模式。

- 备注
1. p: CSI 编号 (p = 00、01、10)
g: PIM 和 POM 编号 (g = 3、7)
 2. f_{MCK}: 串行阵列单元工作时钟频率
(由 SMR0n 寄存器的 CKS0n 位设置的工作时钟。n: 通道编号 (n=0~2))

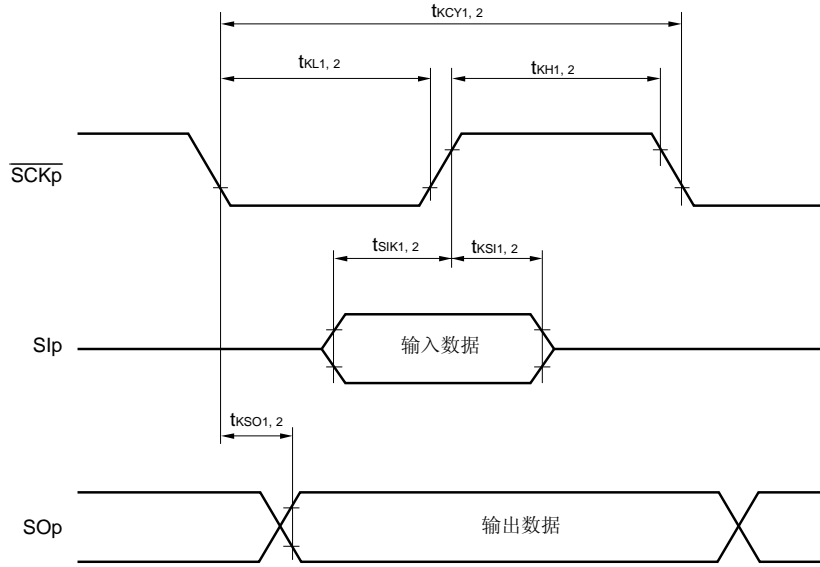
(2) 串行接口：串行阵列单元(4/17)

CSI 模式连接图 (相同电平下通信)



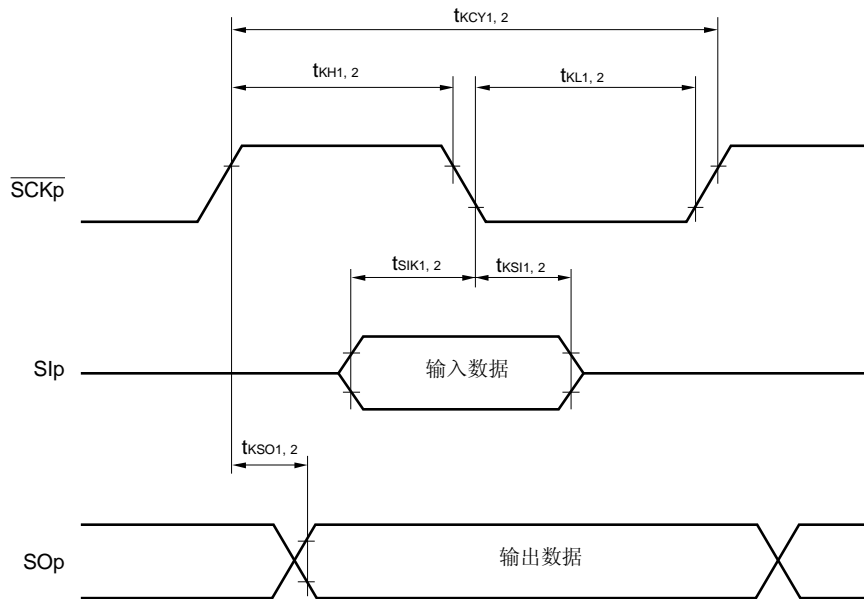
CSI 模式串行传送时序（相同电平下通信）

（当 $DAP0n = 0$ 且 $CKP0n = 0$ ，或 $DAP0n = 1$ 且 $CKP0n = 1$ 。）



CSI 模式串行传送时序（相同电平下通信）

（当 $DAP0n = 0$ 且 $CKP0n = 1$ ，或 $DAP0n = 1$ 且 $CKP0n = 0$ ）



- 备注
1. p: CSI 编号(p = 00、01、10)
 2. n: 单元编号 (n = 0~2)

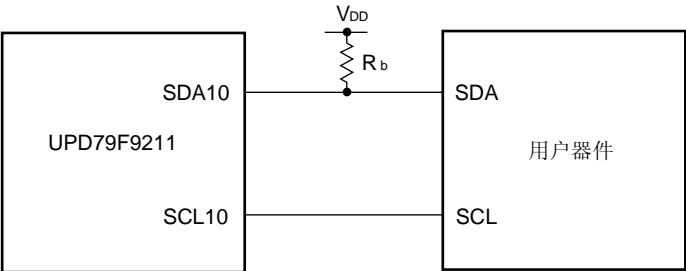
(2) 串行接口：串行阵列单元(5/17)

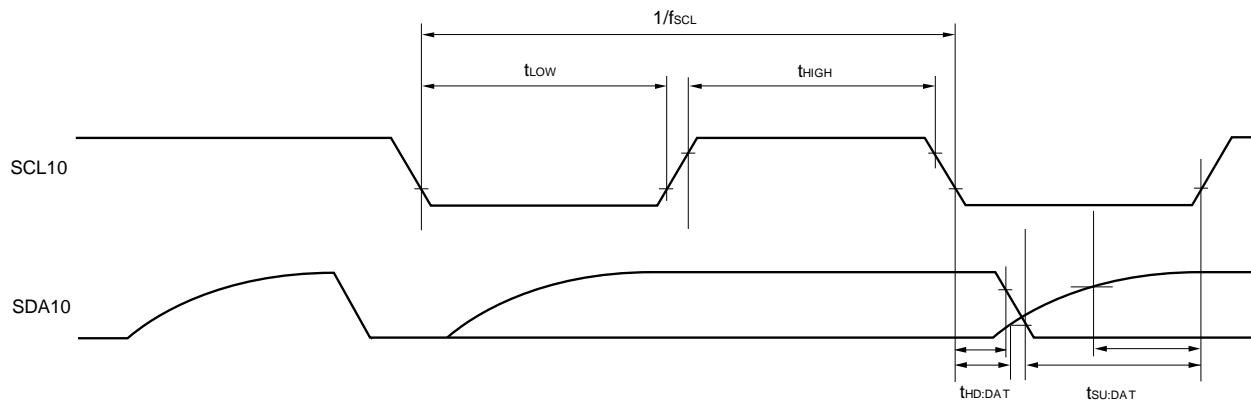
(TA = -40 至+85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

(d) 相同电平下通信（简易 I²C 模式）

参数	符号	条件	最小值	最大值	单位
SCL10 时钟频率	fSCL	4.0 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 1.7 kΩ		1.0	MHz
		2.7 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ		480	kHz
保持时间，当 SCL10 =“L”	tLOW	4.0 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 1.7 kΩ	475		ns
		2.7 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ	995		ns
保持时间，当 SCL10 =“H”	tHIGH	4.0 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 1.7 kΩ	475		ns
		2.7 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ	995		ns
数据设置时间（接收）	tsu:DAT	4.0 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 1.7 kΩ	1/fMCK+85		ns
		2.7 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ	1/fMCK+120		ns
数据保持时间（传送）	thd:DAT	4.0 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 1.7 kΩ	0	130	ns
		2.7 V ≤ VDD ≤ 5.5 V, Cb = 100 pF, Rb = 3 kΩ	0	160	ns

简易 I²C 模式连接图（相同电平下通信）



简易 I²C 模式串行传送时序（相同电平下通信）

注意事项 用 PIM3 和 POM3 寄存器，为 SDA10 选择普通输入缓存器和 N 沟道漏极开路输出（V_{DD} 耐压）模式，为 SCL10 选择普通输出模式输出。

- 备注**
1. $R_b[\Omega]$: 通讯线 (SDA10) 上拉电阻，
 $C_b[F]$: 通信线 (SCL10, SDA10) 负载电容
 2. f_{MCK} : 串行阵列单元工作时钟频率
 （通过 SMR02 寄存器的 CKS02 位，设置工作时钟。）

(2) 串行接口：串行阵列单元(6/17)

(TA = -40 至+85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

(e) 不同电平下通信 (2.5V, 3V) (UART 模式) (专用波特率发生器输出) (1/2)

参数	符号	条件			最小值	典型值	最大值	单位
传送速率		接收	4.0 V ≤ VDD ≤ 5.5 V,				fMCK/6	bps
			2.7 V ≤ Vb ≤ 4.0 V	fCLK = 20 MHz, fMCK = fCLK			3.3	Mbps
			2.7 V ≤ VDD ≤ 4.0 V,				fMCK/6	bps
			2.3 V ≤ Vb ≤ 2.7 V	fCLK = 20 MHz, fMCK = fCLK			3.3	Mbps

注意事项 使用 PIMg 和 POMg 寄存器，为 RxDq 选择 TTL 输入缓冲器，为 TxDq 选择 N 沟道漏极开路输出（VDD 耐压）模式。

- 备注
1. q: UART 编号(q =0, 1), g: PIM 和 POM 编号 (g = 3, 7)
 2. Vb[V]: 通信线电压
 3. fMCK: 串行阵列单元工作时钟频率
(由 SMR0n 寄存器的 CKS0n 位设置的工作时钟。n: 通道编号 (n=0~3))
 4. 以下的 VIH 和 VIL 是在 UART 模式中使用不同电平通信时的串行阵列单元 AC 特性的观察点。
4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V: VIH = 2.2 V, VIL = 0.8 V
2.7 V ≤ VDD ≤ 4.0V, 2.3 V ≤ Vb ≤ 2.7 V: VIH = 2.0 V, VIL = 0.5 V

(2) 串行接口: 串行阵列单元(7/17)

(TA = -40 至 +85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

(e) 不同电平下通信 (2.5V, 3V) (UART 模式) (专用波特率发生器输出) (2/2)

参数	符号	条件	最小值	典型值	最大值	单位
传送速率		传送	4.0 V ≤ VDD ≤ 5.5 V,			注 1
		2.7 V ≤ Vb ≤ 4.0 V	fCLK = 16.8 MHz, fMCK = fCLK, Cb = 50 pF, Rb = 1.4 kΩ, Vb = 2.7 V			2.8 ^{#2}
		2.7 V ≤ VDD ≤ 4.0 V,				注 3
		2.3 V ≤ Vb ≤ 2.7 V	fCLK = 19.2 MHz, fMCK = fCLK, Cb = 50 pF, Rb = 2.7 kΩ, Vb = 2.3 V			1.2 ^{#4}

注 1. 使用 fMCK/6 或下面公式计算的较小的最大传送速率是有效最大传送速率。

当 4.0 V ≤ VDD ≤ 5.5 V 和 2.7 V ≤ Vb ≤ 4.0 V 时, 用于计算发送速率的公式。

$$\text{最大传送速率} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \quad [\text{bps}]$$

$$\text{波特率误差 (理论值)} = \frac{\frac{1}{\text{传输速率} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{传输速率}}) \times \text{传送位数}} \times 100 [\%]$$

* 该值是传输和传送端之间相对差的理论值。

2. 作为示例, 当满足“条件”栏中描述的条件时, 将计算该值。参见上述注 1, 根据用户实际条件来计算最大传送速率。

3. 使用 fMCK/6 或下面公式计算的较小的最大传送速率是有效最大传送速率。

当 2.7 V ≤ VDD ≤ 4.0 V 和 2.3 V ≤ Vb ≤ 2.7 V 时, 用于计算传送速率的公式。

$$\text{最大传送速率} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \quad [\text{bps}]$$

$$\text{波特率误差 (理论值)} = \frac{\frac{1}{\text{传输速率} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{传输速率}}) \times \text{传送位数}} \times 100 [\%]$$

* 该值是传输和传送端之间相对差的理论值。

4. 作为示例, 当满足“条件”栏中描述的条件时, 将计算该值。参见上述注 3, 根据用户实际条件来计算最大传送速率。

注意事项 使用 PIMg 和 POMg 寄存器, 为 RxDq 选择 TTL 输入缓冲器, 为 TxDq 选择 N 沟道漏极开路输出 (VDD 耐压) 模式。

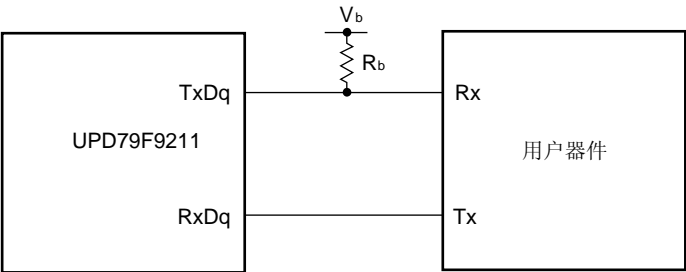
(备注在下页列出。)

(2) 串行接口：串行阵列单元(8/17)

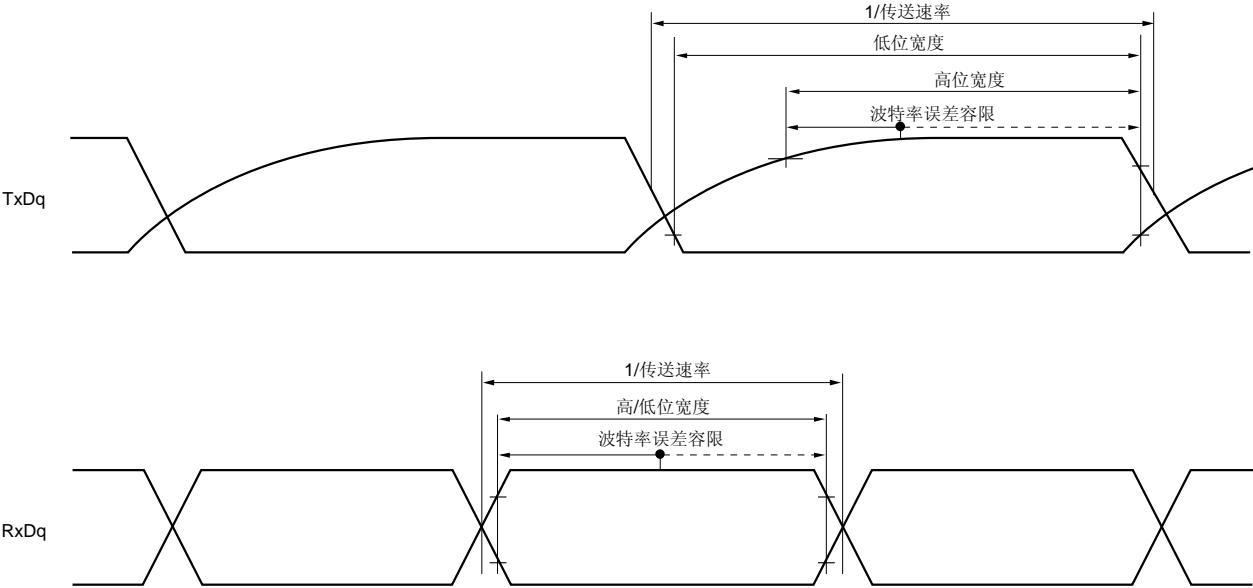
- 备注
- 1. $R_b[\Omega]$: 通信线 (TxDq) 上拉电阻, $C_b[F]$: 通信线 (TxDq) 负载电容, 电容, $V_b[V]$: 通信线电压
 - 2. q: UART 编号(q =0, 1), g: PIM 和 POM 编号 (g = 3, 7)
 - 3. f_{MCK} : 串行阵列单元工作时钟频率
(由 SMR0n 寄存器的 CKS0n 位设置的工作时钟。n: 通道编号 (n=0~3))
 - 4. 以下的 V_{IH} 和 V_{IL} 是在 UART 模式中使用不同电平通信时的串行阵列单元 AC 特性的观察点。
 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}, 2.7\text{ V} \leq V_b \leq 4.0\text{ V}$: $V_{IH} = 2.2\text{ V}, V_{IL} = 0.8\text{ V}$
 $2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}, 2.3\text{ V} \leq V_b \leq 2.7\text{ V}$: $V_{IH} = 2.0\text{ V}, V_{IL} = 0.5\text{ V}$

(2) 串行接口：串行阵列单元(9/17)

UART 模式连接图（不同电平下通信）



UART 模式位宽度（不同电平下通信）（参考）



注意事项 使用 PIMg 和 POMg 寄存器，为 RxDq 选择 TTL 输入缓冲器，为 TxDq 选择 N 沟道漏极开路输出（ V_{DD} 耐压）模式。

- 备注
- 1. $R_b[\Omega]$: 通信线 (TxDq) 上拉电阻, $V_b[V]$: 通信线电压
 - 2. q: UART 编号(q =0, 1), g: PIM 和 POM 编号 (g = 3, 7)

(2) 串行接口: 串行阵列单元(10/17)

(TA = -40 至 +85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

(f) 不同电平下通信 (2.5 V, 3 V) (CSI 模式) (主设备模式, $\overline{\text{SCKp}}$... 内部时钟输出) (1/2)

参数	符号	条件	最小值	典型值	最大值	单位
$\overline{\text{SCKp}}$ 周期时间	t _{KCY1}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 50 pF, R _b = 1.4 kΩ	500			ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b < 2.7 V, C _b = 50 pF, R _b = 2.7 kΩ	1000			ns
$\overline{\text{SCKp}}$ 高电平宽度	t _{KH1}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 50 pF, R _b = 1.4 kΩ	t _{KCY1} /2 - 120			ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b < 2.7 V, C _b = 50 pF, R _b = 2.7 kΩ	t _{KCY1} /2 - 275			ns
$\overline{\text{SCKp}}$ 低电平宽度	t _{KL1}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 50 pF, R _b = 1.4 kΩ	t _{KCY1} /2 - 20			ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b < 2.7 V, C _b = 50 pF, R _b = 2.7 kΩ	t _{KCY1} /2 - 35			ns
Slp 设置时间 (到 $\overline{\text{SCKp}}\uparrow$) ^{※1}	t _{SIK1}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 50 pF, R _b = 1.4 kΩ	195			ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b < 2.7 V, C _b = 50 pF, R _b = 2.7 kΩ	380			ns
Slp 保持时间 (从 $\overline{\text{SCKp}}\uparrow$) [※]	t _{KSI1}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 50 pF, R _b = 1.4 kΩ	30			ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b < 2.7 V, C _b = 50 pF, R _b = 2.7 kΩ	30			ns
从 $\overline{\text{SCKp}}\downarrow$ 到 SOp 输出的延迟时间 [※]	t _{KSO1}	4.0 V ≤ V _{DD} ≤ 5.5 V, 2.7 V ≤ V _b ≤ 4.0 V, C _b = 50 pF, R _b = 1.4 kΩ			165	ns
		2.7 V ≤ V _{DD} < 4.0 V, 2.3 V ≤ V _b < 2.7 V, C _b = 50 pF, R _b = 2.7 kΩ			320	ns

注 当 DAP0n = 0 且 CKP0n = 0, 或 DAP0n = 1 且 CKP0n = 1

注意事项 使用 PIMg 和 POMg 寄存器, 为 Slp 选择 TTL 输入缓冲器, 为 SOp 和 $\overline{\text{SCKp}}$ 选择 N 沟道漏极开路输出 (VDD 耐压) 模式。

备注

1. p: CSI 编号(p = 00、01、10)
g: PIM 和 POM 编号 (g = 3、7)
2. n: 单元编号 (n = 0~2)
3. R_b[Ω]: 通信线 ($\overline{\text{SCKp}}$, SOp) 上拉电阻,
C_b[F]: 通信线 (Slp, SOp, $\overline{\text{SCKp}}$) 负载电容, V_b[V]: 通信线电压
4. 以下的 V_{IH} 和 V_{IL} 是在 CSI 模式下使用不同电平通信时的串行阵列单元 AC 特性的观察点。
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) 串行接口: 串行阵列单元(11/17)

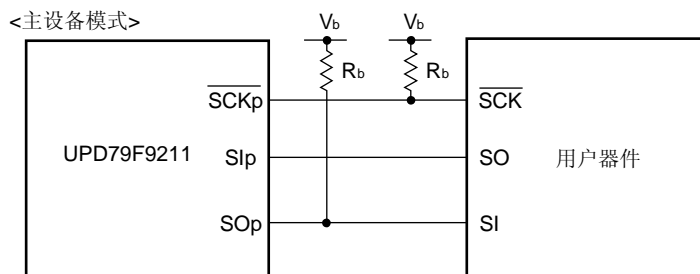
(TA = -40 至 +85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

(f) 不同电平下通信 (2.5 V, 3 V) (CSI 模式) (主设备模式, SCKp... 内部时钟输出) (2/2)

参数	符号	条件	最小值	典型值	最大值	单位
Slp 设置时间 (到 SCKp↓) ^{※1}	tsIK1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	70			ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	100			ns
Slp 保持时间 (从 SCKp↓) [※]	tkSI1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ	30			ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ	30			ns
从 SCKp↑ 到 SOp 输出的延迟时间 [※]	tkSO1	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ			40	ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb < 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ			40	ns

注 当 DAP0n = 0 且 CKP0n = 1, 或 DAP0n = 1 且 CKP0n = 0

CSI 模式连接图 (不同电平下通信)



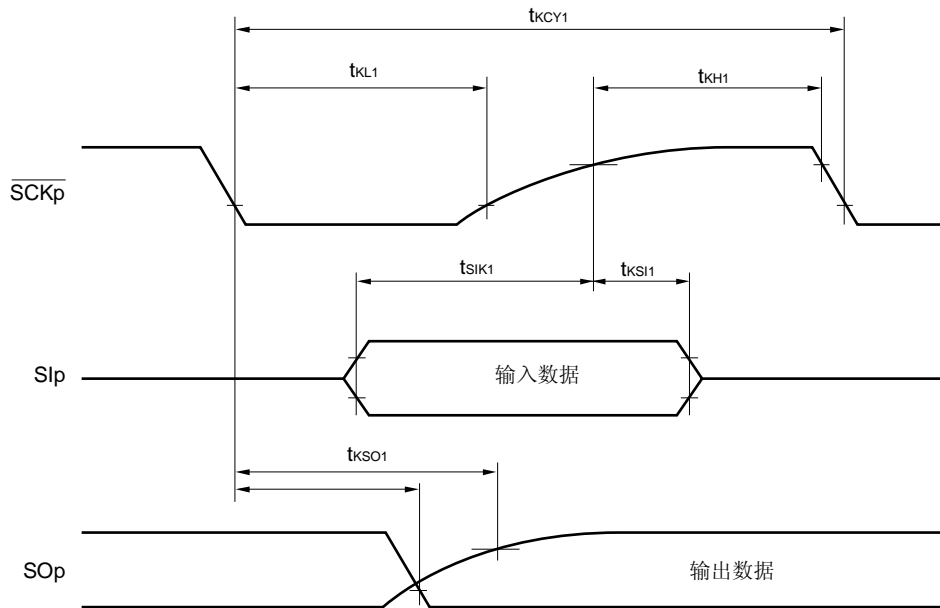
注意事项 使用 PIMg 和 POMg 寄存器, 为 Slp 选择 TTL 输入缓冲器, 为 SOp 和 SCKp 选择 N 沟道漏极开路输出 (VDD 耐压) 模式。

备注

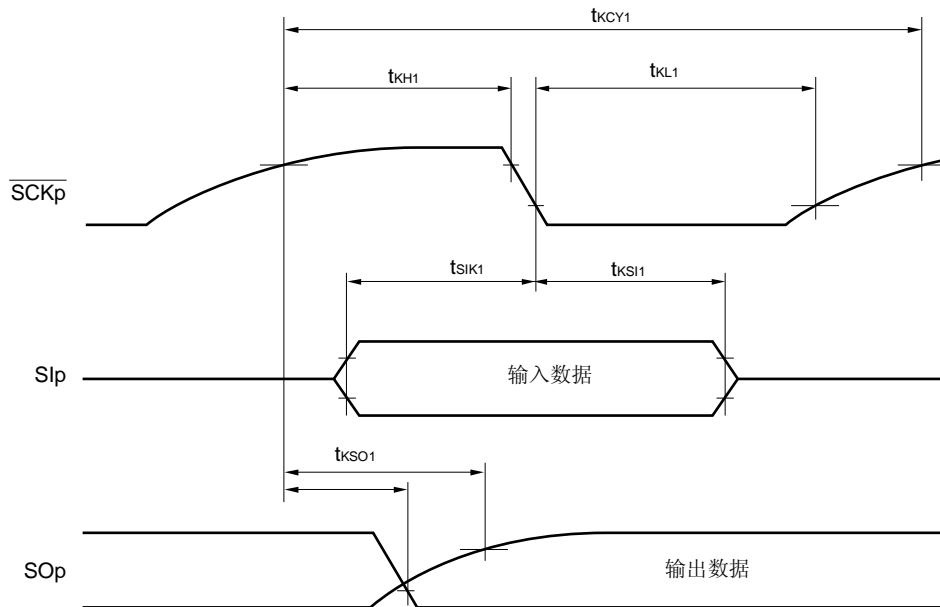
1. p: CSI 编号 (p = 00、01、10)
g: PIM 和 POM 编号 (g = 3、7)
2. n: 单元编号 (n = 0~2)
3. Rb[Ω]: 通信线 (SCKp, SOp) 上拉电阻,
Cb[F]: 通信线 (Slp, SOp, SCKp) 负载电容, Vb[V]: 通信线电压
4. 以下的 VIH 和 VIL 是在 CSI 模式下使用不同电平通信时的串行阵列单元 AC 特性的观察点。
4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V: VIH = 2.2 V, VIL = 0.8 V
2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V: VIH = 2.0 V, VIL = 0.5 V

(2) 串行接口：串行阵列单元(12/17)

CSI 模式传输时序（不同电平下通信）

（当 $DAP0n = 0$ 且 $CKP0n = 0$ ，或 $DAP0n = 1$ 且 $CKP0n = 1$ ）

CSI 模式传输时序（不同电平下通信）

（当 $DAP0n = 0$ 且 $CKP0n = 1$ ，或 $DAP0n = 1$ 且 $CKP0n = 0$ ）

注意事项 使用 PIMg 和 POMg 寄存器，为 SIp 选择 TTL 输入缓冲器，为 SOp 和 SCKp 选择 N 沟道漏极开路输出（V_{DD} 耐压）模式。

- 备注
1. p: CSI 编号(p = 00、01、10)
g: PIM 和 POM 编号 (g = 3、7)
 2. n: 单元编号 (n = 0~2)

(2) 串行接口：串行阵列单元(13/17)

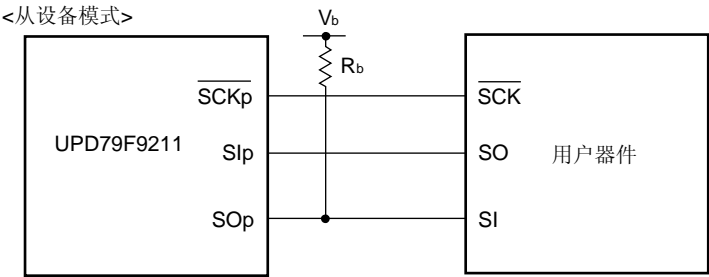
(TA = -40 至+85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

(g) 不同电平下的通信 (2.5 V, 3 V) (CSI 模式) (从设备模式, $\overline{\text{SCKp}}$... 外部时钟输入)

参数	符号	条件	最小值	典型值	最大值	单位
$\overline{\text{SCKp}}$ 周期时间	tKCY2	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V	16.6 MHz < fMCK	12/fMCK		ns
			12.5 MHz < fMCK ≤ 16.6 MHz	10/fMCK		ns
			8.3 MHz < fMCK ≤ 12.5 MHz	8/fMCK		ns
			fMCK ≤ 8.3 MHz	6/fMCK		ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V	17.5 MHz < fMCK	18/fMCK		ns
			15 MHz < fMCK ≤ 17.5 MHz	16/fMCK		ns
			12.5 MHz < fMCK ≤ 15 MHz	14/fMCK		ns
			10 MHz < fMCK ≤ 12.5 MHz	12/fMCK		ns
			7.5 MHz < fMCK ≤ 10 MHz	10/fMCK		ns
			5 MHz < fMCK ≤ 7.5 MHz	8/fMCK		ns
			fMCK ≤ 5 MHz	6/fMCK		ns
$\overline{\text{SCKp}}$ 高/低电平宽度	tKH2, tKL2	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V	fKCY2/2 – 20			ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V	fKCY2/2 – 35			ns
Slp 设置时间 (到 $\overline{\text{SCKp}}\uparrow$) 注 1	tSIK2		1/fMCK + 90			ns
Slp 保持时间 (从 $\overline{\text{SCKp}}\uparrow$) 注 2	tKSI2		50			ns
从 $\overline{\text{SCKp}}\downarrow$ 到 SOp 输出的 延迟时间 注 3	tKSO2	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 50 pF, Rb = 1.4 kΩ			1/fMCK + 245	ns
		2.7 V ≤ VDD < 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 50 pF, Rb = 2.7 kΩ			1/fMCK + 400	ns

- 注
1. 当 DAP0n = 0 和 CKP0n = 0, 或 DAP0n = 1 和 CKP0n = 1 时。Slp 设置时间变为“至 $\overline{\text{SCKp}}\downarrow$ ”, 当 DAP0n = 0 和 CKP0n = 1, 或 DAP0n = 1 和 CKP0n = 0。
 2. 当 DAP0n = 0 和 CKP0n = 0, 或 DAP0n = 1 和 CKP0n = 1 时。Slp 保持时间变为“自 $\overline{\text{SCKp}}\downarrow$ ”, 当 DAP0n = 0 和 CKP0n = 1, 或 DAP0n = 1 和 CKP0n = 0。
 3. 当 DAP0n = 0 和 CKP0n = 0, 或 DAP0n = 1 和 CKP0n = 1 时。SOp 输出的延迟时间变为“自 $\overline{\text{SCKp}}\uparrow$ ”, 当 DAP0n = 0 和 CKP0n = 1, 或 DAP0n = 1 和 CKP0n = 0。

CSI 模式连接图（不同电平下通信）



(注意事项和备注在下页中给出。)

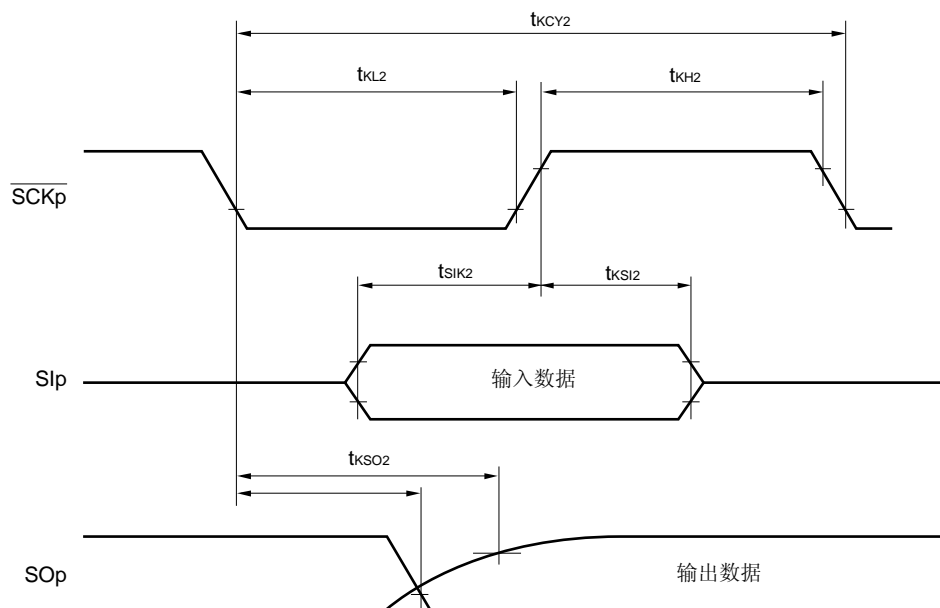
(2) 串行接口：串行阵列单元(14/17)

注意事项 使用 PIMg 和 POMg 寄存器，为 SIp 和 $\overline{\text{SCKp}}$ 选择 TTL 输入缓冲器，为 SOp 选择 N 沟道漏极开路输出 (V_{DD} 耐压) 模式。

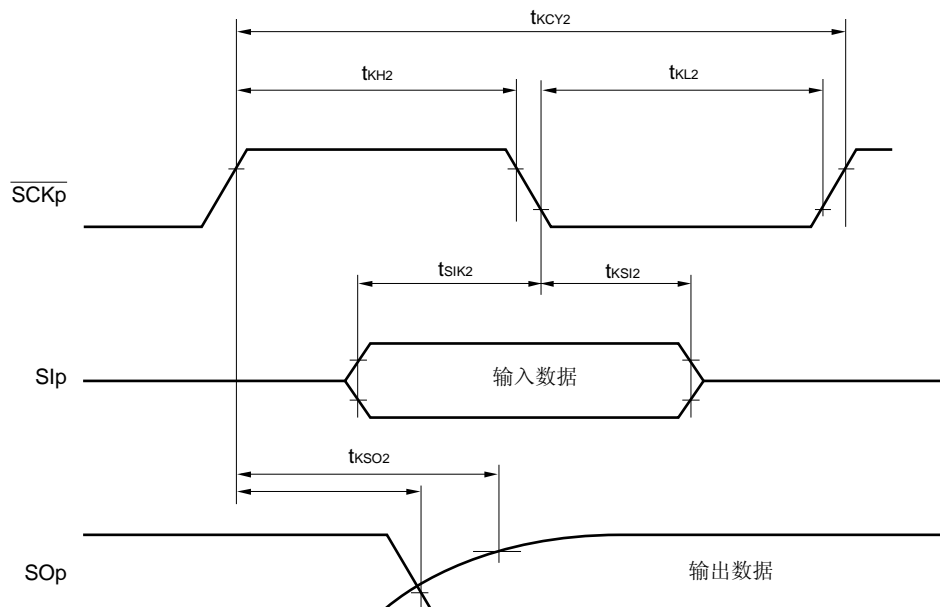
- 备注**
1. p: CSI 编号(p = 00、01、10)
g: PIM 和 POM 编号 (g = 3、7)
 2. $R_b[\Omega]$: 通信线 (SO_p) 上拉电阻,
 $C_b[F]$: 通信线 (SO_p, $\overline{\text{SCKp}}$)负载电容, $V_b[V]$: 通信线电压
 3. f_{MCK} : 串行阵列单元工作时钟频率
(由 SMR0n 寄存器的 CKS0n 位设置的工作时钟。n: 通道编号 (n=0~2))
 4. 以下的 V_{IH} 和 V_{IL} 是在 CSI 模式下使用不同电平通信时的串行阵列单元 AC 特性的观察点。
 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}, 2.7\text{ V} \leq V_b \leq 4.0\text{ V}: V_{IH} = 2.2\text{ V}, V_{IL} = 0.8\text{ V}$
 $2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}, 2.3\text{ V} \leq V_b \leq 2.7\text{ V}: V_{IH} = 2.0\text{ V}, V_{IL} = 0.5\text{ V}$

(2) 串行接口：串行阵列单元(15/17)

CSI 模式传输时序（不同电平下通信）

（当 $DAP0n = 0$ 且 $CKP0n = 0$ ，或 $DAP0n = 1$ 且 $CKP0n = 1$ ）

CSI 模式传输时序（不同电平下通信）

（当 $DAP0n = 0$ 且 $CKP0n = 1$ ，或 $DAP0n = 1$ 且 $CKP0n = 0$ ）

注意事项 使用 $PIMg$ 和 $POMg$ 寄存器，为 Slp 和 \overline{SCKp} 选择 TTL 输入缓冲器，为 SOp 选择 N 沟道漏极开路输出（ V_{DD} 耐压）模式。

- 备注
1. p : CSI 编号 ($p = 00、01、10$)
 g : PIM 和 POM 编号 ($g = 3、7$)
 2. n : 单元编号 ($n = 0 \sim 2$)

(2) 串行接口: 串行阵列单元(16/17)

(TA = -40 至+85°C, 2.7 V ≤ VDD ≤ 5.5 V, VSS = AVSS = 0 V)

(h) 不同电平下通信 (2.5 V, 3 V) (简易 I²C 模式)

参数	符号	条件	最小值	最大值	单位
SCL10 时钟频率	f _{SCL}	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 100 pF, Rb = 1.4 kΩ		1.0	MHz
		2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 100 pF, Rb = 2.7 kΩ		480	kHz
保持时间, 当 SCL10 = "L"	t _{LOW}	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 100 pF, Rb = 1.4 kΩ	525		ns
		2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 100 pF, Rb = 2.7 kΩ	1065		ns
保持时间, 当 SCL10 = "H"	t _{HIGH}	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 100 pF, Rb = 1.4 kΩ	235		ns
		2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 100 pF, Rb = 2.7 kΩ	445		ns
数据设置时间 (接收)	t _{SU:DAT}	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 100 pF, Rb = 1.4 kΩ	1/f _{MCK} + 135		ns
		2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 100 pF, Rb = 2.7 kΩ	1/f _{MCK} + 190		ns
数据保持时间 (传送)	t _{HD:DAT}	4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V, Cb = 100 pF, Rb = 1.4 kΩ	0	140	ns
		2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V, Cb = 100 pF, Rb = 2.7 kΩ	0	160	ns

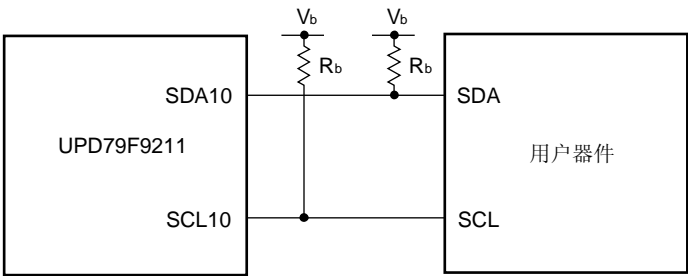
注意事项 通过 PIM3 和 POM3 寄存器, 为 SDA10 选择普通输入缓冲器和 N 沟道漏极开路输出 (VDD 耐压) 模式, 为 SCL10 选择 N 沟道漏极开路输出 (VDD 耐压) 模式。

备注

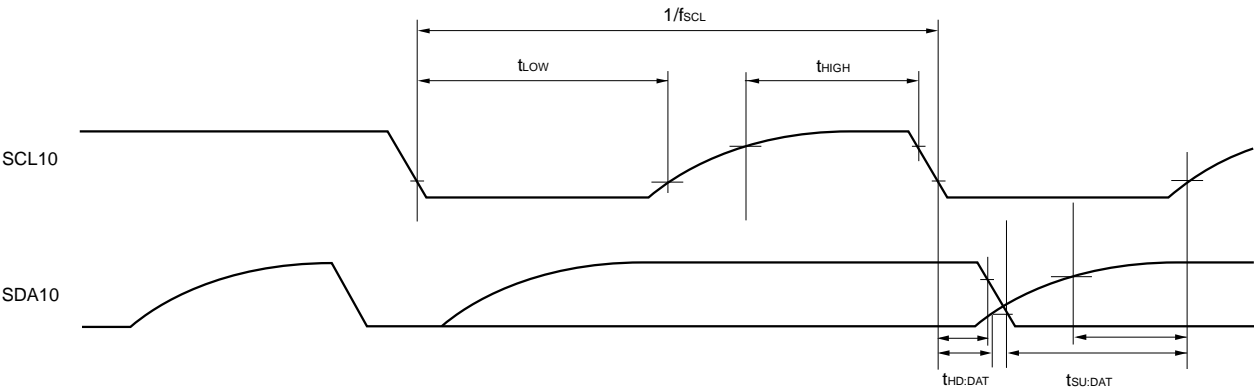
1. Rb[Ω]: 通讯线 (SDA10, SCL10) 上拉电阻,
Cb[F]: 通信线 (SDA10, SCL10) 负载电容, Vb[V]: 通信线电压
2. f_{MCK}: 串行阵列单元工作时钟频率
(通过 SMR02 寄存器的 CKS02 位, 设置工作时钟。)
3. 以下的 VIH 和 VIL 是在简易 I²C 模式下使用不同电平通信时的串行阵列单元 AC 特性。
4.0 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ Vb ≤ 4.0 V: VIH = 2.2 V, VIL = 0.8 V
2.7 V ≤ VDD ≤ 4.0 V, 2.3 V ≤ Vb ≤ 2.7 V: VIH = 2.0 V, VIL = 0.5 V

(2) 串行接口：串行阵列单元(17/17)

简易 I²C 模式连接图（不同电平下通信）



简易 I²C 模式串行传送时序（不同电平下通信）



注意事项 通过 PIM3 和 POM3 寄存器，为 SDA10 选择普通输入缓冲器和 N 沟道漏极开路输出（V_{DD} 耐压）模式，为 SCL10 选择 N 沟道漏极开路输出（V_{DD} 耐压）模式。

备注 R_b[Ω]：通信线 (SDA10, SCL10) 上拉电阻，V_b[V]：通信线电压

(3) 串行接口：片上调试 (UART)

(T_A = -40 至+85°C， 2.7 V ≤ V_{DD} ≤ 5.5 V， V_{SS} = AV_{SS} = 0 V)

(a) 片上调试 (UART)

参数	符号	条件	最小值	典型值	最大值	单位
传送速率			f _{CLK} /212		f _{CLK} /6	bps
		Flash 存储器编程模式			3.33	Mbps
TOOL1 输出频率	f _{TOOL1}	2.7 V ≤ V _{DD} ≤ 5.5 V			10	MHz

A/D 转换器特性

(T_A = -40 至+85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = AV_{SS} = 0 V)

参数	符号	条件	最小值	典型值	最大值	单位
分辨率	RES				10	位
总误差 ^{注 1, 2}	AINL	4.0 V ≤ AV _{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV _{REF} < 4.0 V			±0.6	%FSR
转换时间	t _{CONV}	4.0 V ≤ AV _{REF} ≤ 5.5 V	高速模式 1	2.5		66.6 μs
			普通模式	6.1		66.6 μs
		2.7 V ≤ AV _{REF} < 4.0 V	高速模式 2	4.5		66.6 μs
			普通模式	12.2		66.6 μs
零度误差 ^{注 1, 2}	E _{ZS}	4.0 V ≤ AV _{REF} ≤ 5.5 V	普通模式		±0.4	%FSR
		2.7 V ≤ AV _{REF} < 4.0 V	普通模式		±0.6	%FSR
满度误差 ^{注 1, 2}	E _{FS}	4.0 V ≤ AV _{REF} ≤ 5.5 V	普通模式		±0.4	%FSR
		2.7 V ≤ AV _{REF} < 4.0 V	普通模式		±0.6	%FSR
积分线性误差 ^{注 1}	I _{LE}	4.0 V ≤ AV _{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV _{REF} < 4.0 V			±4.5	LSB
微分线性误差 ^{注 1}	D _{LE}	4.0 V ≤ AV _{REF} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV _{REF} < 4.0 V			±2.0	LSB
模拟 输入电压	V _{AIN}	2.7 V ≤ AV _{REF} ≤ 5.5 V	AV _{SS}		AV _{REF}	V

- 注
1. 不包括量化误差 (±1/2 LSB)。
 2. 此值表示对满度的百分比 (%FSR)。

可编程增益放大器特性

(TA = -40 至+85°C, 2.7 V ≤ VDD ≤ 5.5 V, 2.7 V ≤ AVREF ≤ VDD, VSS = AVSS = 0 V)

参数	符号	条件		最小值	典型值	最大值	单位
输入偏移电压	VIOAMP				T.B.D		mV
输入电压范围	VIAMP	×1 增益		0.1AVREF		0.45AVREF	V
		其它值		0.1AVREF/增益		0.9AVREF/增益	V
最大输出电压	VOAMP			0.1AVREF		0.9AVREF	V
回转速率	SRF	上升沿	4.0 V ≤ AVREF ≤ 5.5 V		T.B.D		V/μs
			2.7 V ≤ AVREF < 4.0 V		T.B.D		V/μs
	SRR	下降沿	4.0 V ≤ AVREF ≤ 5.5 V		T.B.D		V/μs
			2.7 V ≤ AVREF < 4.0 V		T.B.D		V/μs
增益	RG			1 至 12			次
操作稳定等待时间	tAMP					3	μs

备注 慢速率：与输出电压的上升或下降有关的变化

V/μs：每 1μs 电压的变化

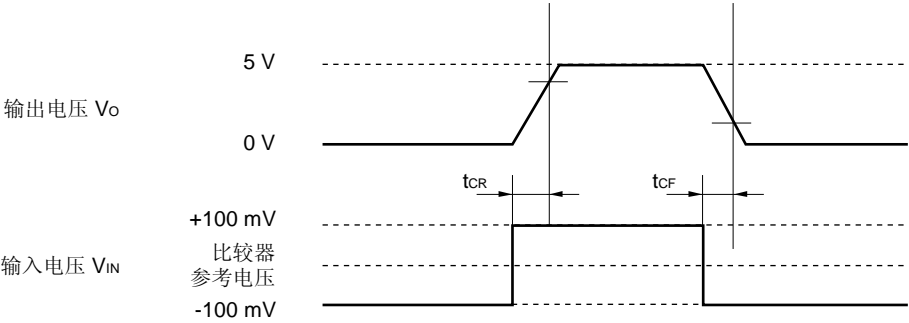
操作稳定等待时间：可编程增益放大器的操作被允许（OAEN = 1）之后，到满足可编程增益放大器的 DC 和 AC 特性的状态输入所需的时间。

比较器特性

($T_A = -40$ 至 $+85^{\circ}\text{C}$, $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$, $2.7\text{ V} \leq AV_{REF} \leq V_{DD}$, $V_{SS} = AV_{SS} = 0\text{ V}$)

参数	符号	条件	最小值	典型值	最大值	单位
输入偏移电压	$V_{IO_{CMP}}$			T.B.D		mV
输入电压范围	$V_{I_{CMP}}$		$0.1AV_{REF}$		$0.9AV_{REF}$	V
偏离内部参考电压	$V_{I_{REF}}$			T.B.D		%
响应时间	t_{CR}	输入幅度 = $\pm 100\text{ mV}$, 在上升沿 ^{注 1}		T.B.D		ns
	t_{CF}	输入幅度 = $\pm 100\text{ mV}$, 在上升沿 ^{注 2}		T.B.D		ns
操作稳定等待时间	t_{CMP}				1	μs
参考电压稳定等待时间	t_{VR}				1	μs

- 注
1. 当 CMP0P 和 CMP1P 的输入或可编程增益放大器的输出从比较器参考电压 -100 mV 变为比较器参考电压 $+100\text{ mV}$ 时的脉冲响应特性。
 2. 当 CMP0P 和 CMP1P 的输入或可编程增益放大器的输出从比较器参考电压 $+100\text{ mV}$ 变为比较器参考电压 -100 mV 时的脉冲响应特性。



备注

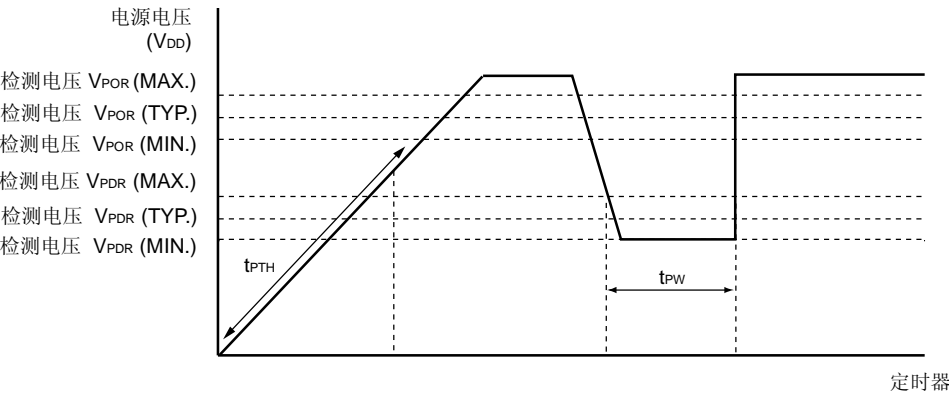
操作稳定等待时间: 在比较器开始工作($CnEN = 1$)后, 需要一定时间使比较器满足 DC 和 AC 规范

稳定等待时间: 内部参考电压被允许($CnREN = 1$)之后, 到内部参考电压电路的电压等级达到理想值的 99%所需的时间。

POC 电路特性 (TA = -40 至 +85°C, Vss = 0 V)

参数	符号	条件	最小值	典型值	最大值	单位
检测电压	V _{POR}	供电电压上升时间	1.52	1.61	1.70	V
	V _{PDR}	供电电压下降时间	1.5	1.59	1.68	V
供电电压上升斜率	t _{PTH}	改变 V _{DD} 的趋势: 0 V → V _{POR}	0.5			V/ms
最小脉冲宽度	t _{PW}	当电压下降时	200			μs
检测延迟时间					200	μs

POC 电路时序



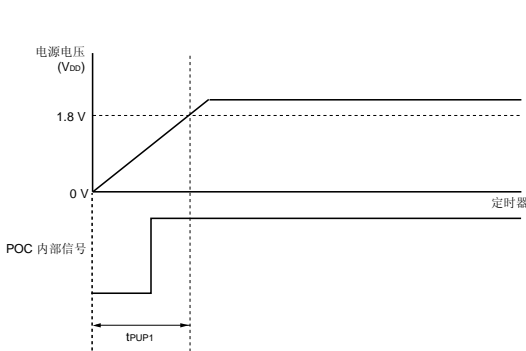
供电电压上升时间 (TA = -40~+85°C, Vss = 0 V)

参数	符号	条件	最小值	典型值	最大值	单位
上升到 1.8 V (V _{DD} (最小值)) [※] 的最大时间 (V _{DD} : 0 V → 1.8 V)	t _{PUP1}	当不使用 $\overline{\text{RESET}}$ 输入时, LVI 默认的启动功能设置为停止 (LVIOFF (选项字节) = 1)。			3.6	ms
上升到 1.8 V (V _{DD} (最小值)) [※] 的最大时间 (释放 $\overline{\text{RESET}}$ 输入 → V _{DD} : 1.8 V)	t _{PUP2}	当使用 $\overline{\text{RESET}}$ 复位输入时, LVI 默认启动功能设置停止 (LVIOFF (选项字节) = 1)。			1.88	ms

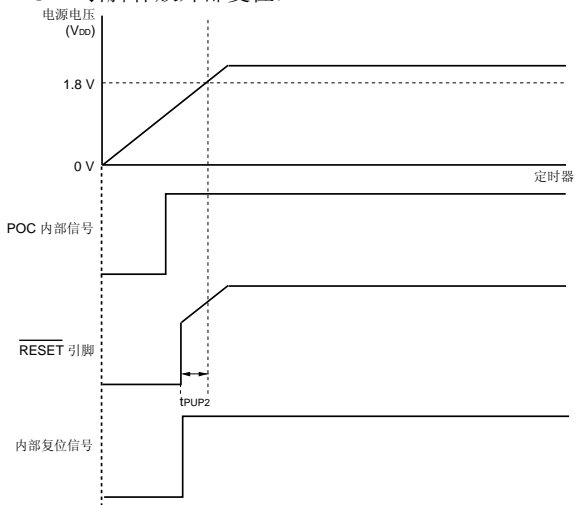
注 确保供电电压上升时间小于此值。

供电电压上升时间时序

- 当不使用 $\overline{\text{RESET}}$ 引脚输入时



- 当使用 $\overline{\text{RESET}}$ 引脚输入时 (在 POC 释放之后, 由 $\overline{\text{RESET}}$ 引脚释放外部复位)



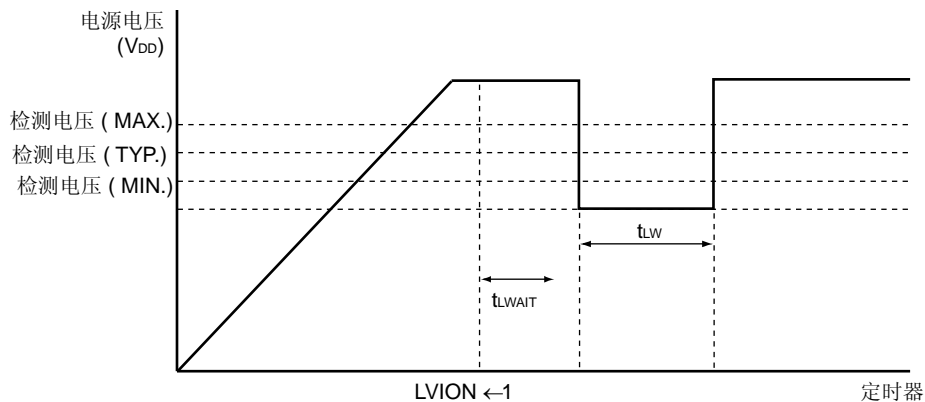
LVI 电路特性($T_A = -40 \sim +85^\circ\text{C}$, $V_{PDR} \leq V_{DD} \leq 5.5\text{ V}$, $V_{SS} = AV_{SS} = 0\text{ V}$)

参数		符号	条件	最小值	典型值	最大值	单位
检测电压	供电电压电平	V _{LVI0}		4.12	4.22	4.32	V
		V _{LVI1}		3.97	4.07	4.17	V
		V _{LVI2}		3.82	3.92	4.02	V
		V _{LVI3}		3.66	3.76	3.86	V
		V _{LVI4}		3.51	3.61	3.71	V
		V _{LVI5}		3.35	3.45	3.55	V
		V _{LVI6}		3.20	3.30	3.40	V
		V _{LVI7}		3.05	3.15	3.25	V
		V _{LVI8}		2.89	2.99	3.09	V
		V _{LVI9}		2.74	2.84	2.94	V
		V _{LVI10}		2.58	2.68	2.78	V
		V _{LVI11}		2.43	2.53	2.63	V
		V _{LVI12}		2.28	2.38	2.48	V
		V _{LVI13}		2.12	2.22	2.32	V
		V _{LVI14}		1.97	2.07	2.17	V
	V _{LVI15}		1.81	1.91	2.01	V	
	外部输入引脚 ^{注 1}	V _{EXLVI}	EXLVI < V _{DD} , 2.7 V ≤ V _{DD} ≤ 5.5 V	1.11	1.21	1.31	V
	电源应用上的供电电压	V _{PUPLVI}	当设置 LVI 默认开始功能允许时	1.87	2.07	2.27	V
最小脉冲宽度		t _{LW}		200			μs
检测延迟时间						200	μs
操作稳定等待时间 ^{注 2}		t _{LWAIT}				10	μs

- 注 1. 使用 EXLVI/P120/INTP0 引脚。
 2. 从设置低电压检测寄存器 (LVIM) 的第 7 位 (LVION) 为 1 到操作稳定所需要的时间。

备注 $V_{LVI(n-1)} > V_{LVin}$: $n = 1 \sim 15$

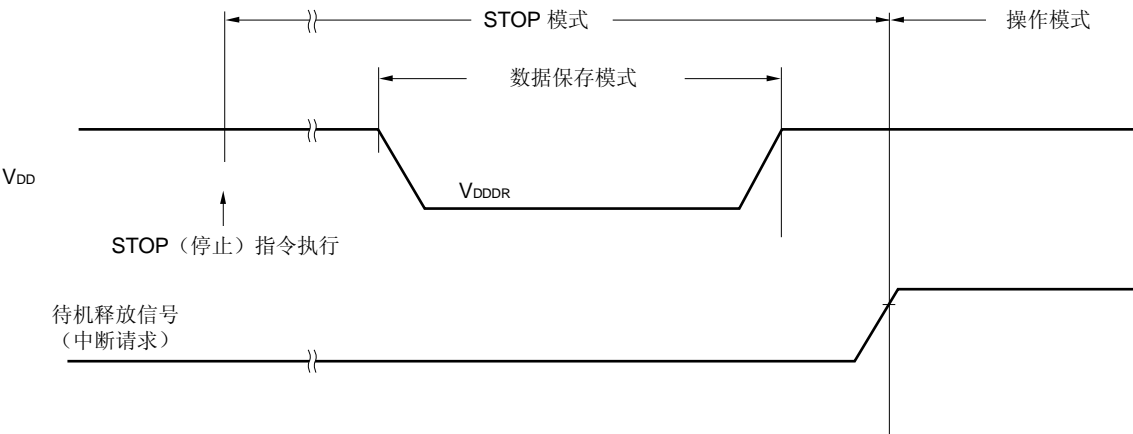
LVI 电路时序



数据存储器 STOP 模式低供电电压数据保持特性(T_A = -40 至 +85C)

参数	符号	条件	最小值	典型值	最大值	单位
数据保持供电电压	V _{DDDR}		1.5 ^註		5.5	V

注 该值根据 POC 检测电压而变化。当电压降低时，一直保持该数据直至 POC 复位有效，但当 POC 复位有效时数据不能保持。



Flash 存储器编程特性

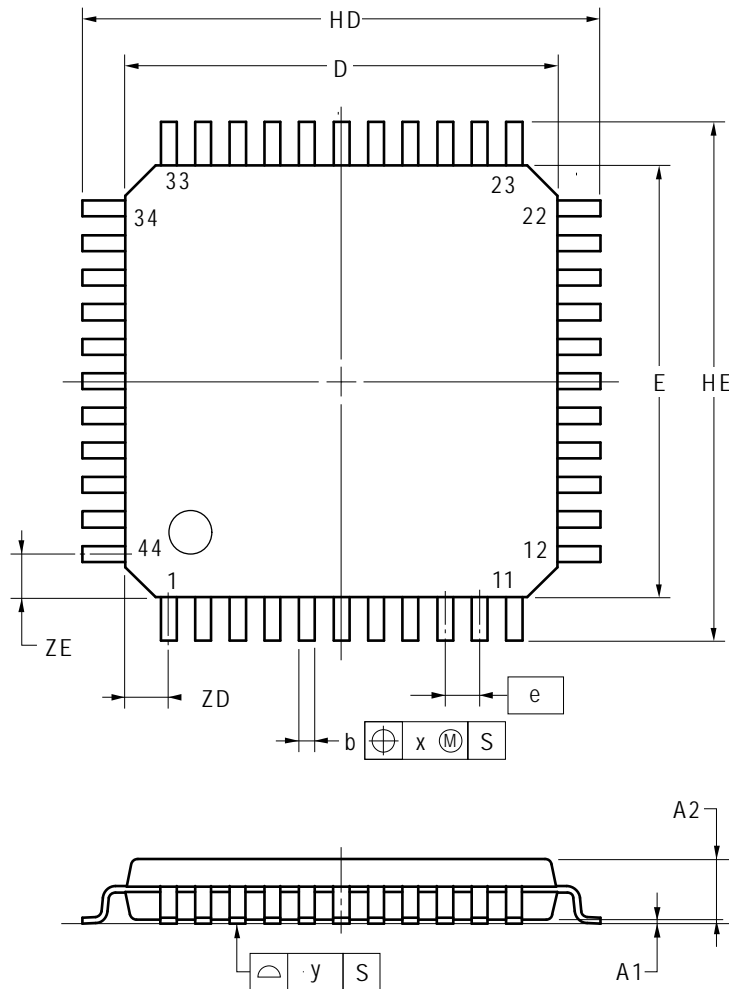
(T_A = -40 至+85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = AV_{SS} = 0 V)

参数	符号	条件	最小值	典型值	最大值	单位
V _{DD} 供电电流	I _{DD}	典型值 = 10 MHz, 最大值 = 20 MHz		6	20	mA
CPU/外围硬件时钟频率	f _{CLK}		2		20	MHz
每片重写次数	C _{erwr}	保持时间: 10 年 擦除 1 次 + 擦除后写 1 次 = 1 次重写 ^註	10			次

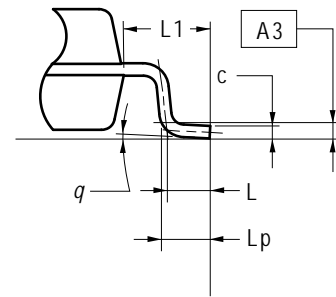
注 当产品运输后首次写入时，“擦→写”和“只写”均作为一次重写。

第 27 章 封装图

44引脚塑封 LQFP (10x10)



铅端详细信息



(单位:mm)

项目	尺寸
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.35 ^{+0.08} _{-0.04}
c	0.125 ^{+0.075} _{-0.025}
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	3° ^{+5°} _{-3°}
e	0.80
x	0.20
y	0.10
ZD	1.00
ZE	1.00

P44GB-80-8ES

© NEC Electronics Corporation 2005

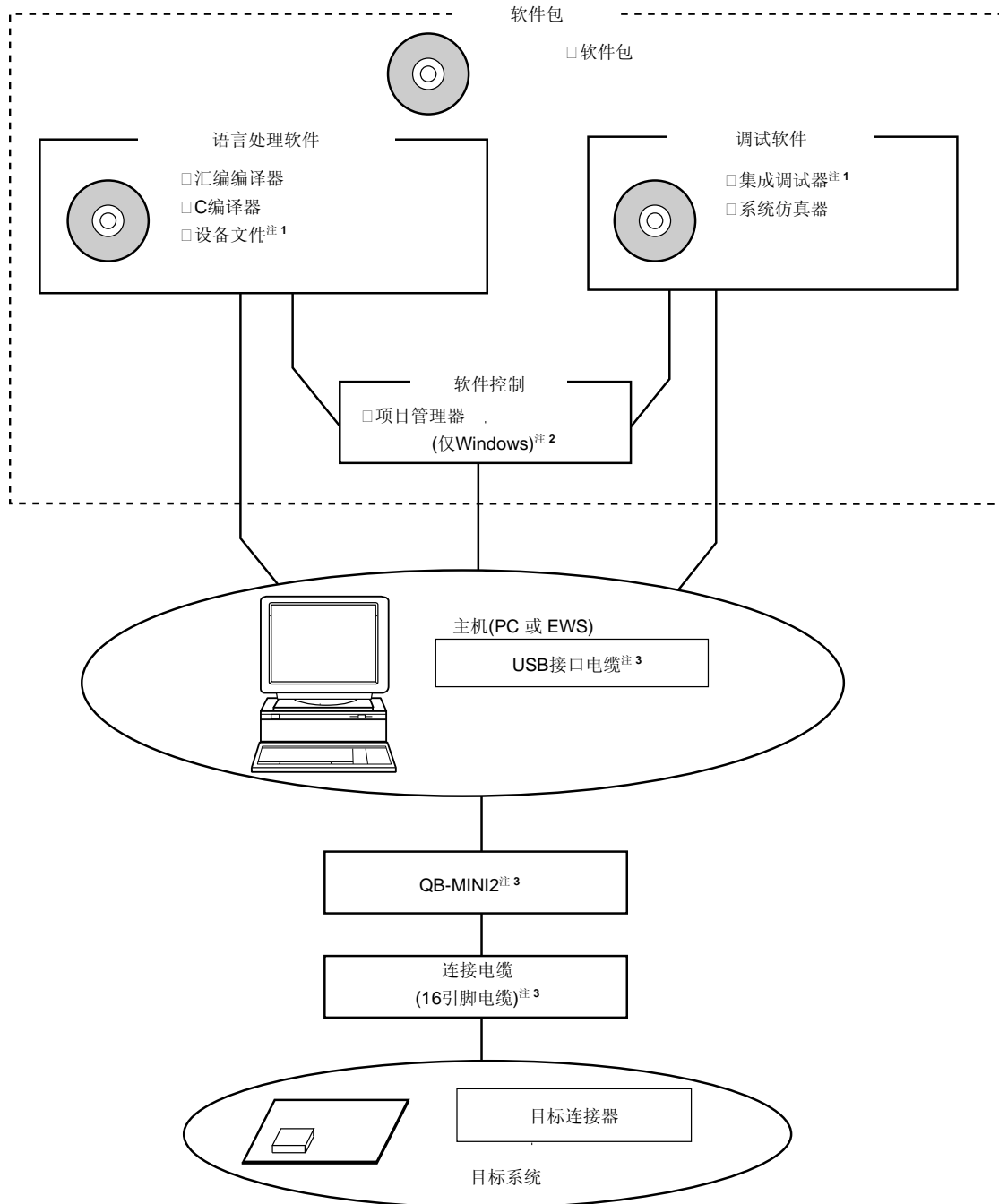
注
在最大材料条件下，每个铅中心线位于其实际位置的 0.20 mm 内。

以下开发工具可用来开发采用 μ PD79F9211 的系统。

图 A-1 显示了开发工具的组成。

图 A-1 开发工具的组成

当使用具有编程功能的片上调试仿真器 QB-MINI2 时



- 注
1. 可从开发工具下载网站（<http://www.necel.com/micro/ods/eng/index.html>）下载 μ PD79F9211（DF799211）的设备文件和集成调试器（ID78K0R-QB）。
 2. 汇编编译器中包括 project manager PM+。
PM+ 仅用于 Windows。
 3. 购买片上调试仿真器 QB-MINI2 时还提供 USB 接口电缆、连接电缆（10 引脚电缆和 16 引脚电缆）和 78K0-OCD 板。其它产品均独立销售。另外，可从 MINICUBE2 的下载网站（<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>）下载操作 QB-MINI2 的软件。

A.1 软件包

SP78K0R 78K0R 系列软件包	开发工具（软件包）适用于 78K0R 微控制器。
------------------------	--------------------------

A.2 语言处理软件

RA78K0R 汇编编译器	<p>该汇编编译器将助记符编写的程序转换为微处理器可执行的目标代码。</p> <p>该汇编编译器可以自动建立符号表和对传送指令进行优化。</p> <p>该汇编编译器应当和设备文件（DF799211）一起使用。</p> <p><在 PC 环境中使用 RA78K0R 的注意事项></p> <p>该汇编编译器是基于 DOS 应用的。它也可以在 Windows 环境下使用，但必须在 Windows 环境下使用 Project Manager(包含在汇编编译器软件包中)。</p>
CC78K0R C 编译器	<p>该编译器将 C 语言编写的程序转换为微控制器可执行的目标代码。</p> <p>该编译器应当结合汇编编译器软件包和设备文件(两者均单独出售)一起使用。</p> <p><在 PC 环境中使用 CC78K0R 的注意事项></p> <p>该 C 编译器是基于 DOS 应用的。它也可以在 Windows 环境下使用，但必须在 Windows 环境下使用 Project Manager(包含在汇编编译器软件包中)。</p>
DF799211 ^注 设备文件	<p>该文件包含设备特有的信息。</p> <p>该设备文件应当结合工具(RA78K0R、CC78K0R、ID78K0R-QB) (均为独立销售)一起使用。</p> <p>相应的 OS 和主机随使用工具而变化。</p>

注 1. DF799211 可以与 RA78K0R、CC78K0R 和 ID78K0R-QB 一起使用。

A.3 Flash存储器编程工具

A.3.1 当使用Flash存储器编程器PG-FP5、PG-FP4 时

PG-FP5、PG-FP4Flash 存储器编程器	Flash 存储器编程器，专用于片上 Flash 存储器的微控制器。
Flash 存储器编程适配器	Flash 存储器编程适配器，用于连接到 Flash 存储器编程器使用。

备注 使用 Flash 存储器编程适配器的最新版本。

A.3.2 当使用具有编程功能的片上调试仿真器QB-MINI2

QB-MINI2 具有编程功能的片上调试仿真器	这是具有片上 Flash 存储器的微控制器专用的 Flash 存储器编程器。当使用 UPD79F9211 微控制器开发应用系统时，它也可以当作片上调试仿真器，用于调试硬件和软件。 当作为 Flash 存储器编程器使用时，它应当和一个连接电缆(16 引脚电缆)和一个 USB 接口电缆一起使用，后者用于连接主机。
----------------------------	---

备注 另外，可从MINICUBE2 的下载网站
(<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>) 下载操作QB-MINI2 的软件。

A.4 调试工具 (硬件)

A.4.2 当使用具有编程功能的片上调试仿真器QB-MINI2 时

QB-MINI2 具有编程功能的片上调试仿真器	当使用 UPD79F9211 微控制器开发应用系统时，该片上调试仿真器用于调试硬件和软件。也可专门当作具有片上 Flash 存储器的微控制器的 Flash 存储器编程器使用。当作为片上调试仿真器使用时，应当与一根连接电缆(16 引脚电缆)和一个 USB 接口电缆一起使用，后者用于连接主机。

备注 另外，可从MINICUBE2 的下载网站
(<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>) 下载操作QB-MINI2 的软件。

ID78K0R-QB 集成调试器	该调试器支持 78K0R 微控制器的在线仿真器。ID78K0R-QB 是基于 Windows 的软件。它改善了 C 兼容的调试功能，并使用集成窗口功能(结合源程序、分开显示和跟踪结果的存储器显示)来显示源程序的跟踪结果。它应当结合设备文件(DF799211)一起使用。
---------------------	--

详细信息请联系:

中国区

MCU 技术支持热线:

电话: +86-400-700-0606 (普通话)

服务时间: 9:00-12:00, 13:00-17:00 (不含法定节假日)

网址:

<http://www.cn.necel.com/> (中文)

<http://www.necel.com/> (英文)

[北京]

日电电子(中国)有限公司

中国北京市海淀区知春路 27 号量子芯座
7, 8, 9, 15 层

电话: (+86) 10-8235-1155

传真: (+86) 10-8235-7679

[深圳]

日电电子(中国)有限公司深圳分公司

深圳市福田区益田路卓越时代广场大厦 39 楼
3901, 3902, 3909 室

电话: (+86) 755-8282-9800

传真: (+86) 755-8282-9899

[上海]

日电电子(中国)有限公司上海分公司

中国上海市浦东新区银城中路 200 号
中银大厦 2409-2412 和 2509-2510 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230

[香港]

香港日电电子有限公司

香港九龙旺角太子道西 193 号新世纪广场
第 2 座 16 楼 1601-1613 室

电话: (+852) 2886-9318

传真: (+852) 2886-9022/ 2886-9044

上海恩益禧电子国际贸易有限公司

中国上海市浦东新区银城中路 200 号
中银大厦 2511-2512 室

电话: (+86) 21-5888-5400

传真: (+86) 21-5888-5230

[成都]

日电电子(中国)有限公司成都分公司

四川省成都市二环路南三段 15 号
天华大厦 608 室

电话: (+86) 28-8512-5224

传真: (+86) 28-8512-5334

[长春]

日电电子(中国)有限公司长春分公司

吉林省长春市朝阳区
西安大路 727 号中银大厦 A 座 1609 室
电话: (+86) 431-8859-7533 / 8859-8533
传真: (+86) 431-8680-2944

[大连]

日电电子(中国)有限公司大连分公司

大连市中山路 88 号天安国际大厦 2701 室
电话: (+86) 411-8230-8815 / 8230-8825
传真: (+86) 411-8230-8835

[青岛]

日电电子(中国)有限公司青岛分公司

中国山东青岛市宁夏路 288 号
G3 楼 607 室
电话: (+86) 532-8872-7900/8872-7901
传真: (+86) 532-8872-7902