

用户手册

78K0/LF2

**8 位单片微控制器
带 LCD 控制器/驱动器**

μ PD78F0372

μ PD78F0373

μ PD78F0374

μ PD78F0375

μ PD78F0376

μ PD78F0376D

μ PD78F0382

μ PD78F0383

μ PD78F0384

μ PD78F0385

μ PD78F0386

μ PD78F0386D

[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} (MAX) 和 V_{IH} (MIN) 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} (MAX) 过渡到 V_{IH} (MIN) 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VDD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, 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, 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。

EEPROM 是 **NEC** 电子公司的注册商标。

Windows 和 **Windows NT** 是美国及其它国家微软公司的注册商标。

PC/AT 是 **IBM** 公司的注册商标。

HP9000 系列 **700** 和 **HP-UX** 是 **HP** 公司的注册商标。

SPARC 工作站是 **SPARC** 公司的注册商标。

Solaris 和 **SunOS** 是 **Sun** 公司的注册商标。

SuperFlash 是 **Silicon Storage Technology, Inc (SST 超捷)** 的注册商标,在包括美国日本的多个国家注册。

注意事项：该产品使用的 SuperFlash[®] 技术获得了 Silicon Storage Technology, Inc. 公司的授权。

- 本档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。本文件所登载内容的错误，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），交通信号控制设备，防灾装置，防止犯罪装置，各种安全装置以及医疗设备（不包括专门为维持生命而设计的设备）。

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

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

（注）

（1）本声明中的“本公司”是指日本电气电子株式会社（NEC Electronics Corporation）及其控股公司。

（2）本声明中的“本公司产品”是指所有由日本电气电子株式会社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

本手册适用于那些希望了解 78K0/LF2 功能，并设计开发应用系统和程序的工程师。
目标产品如下。

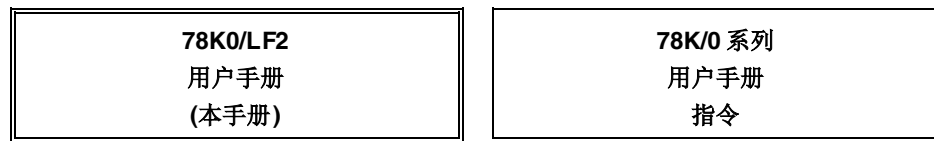
78K0/LF2: μ PD78F0372, μ PD78F0382, μ PD78F0373, μ PD78F0383, μ PD78F0374,
 μ PD78F0384, μ PD78F0375, μ PD78F0385, μ PD78F0376, μ PD78F0386,
 μ PD78F0376D, μ PD78F0386D

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

78K0/LF2 手册主要分为两个部分：手册和指令（与 78K/0 系列通用）。



- 引脚功能
- 内部模块功能
- 中断
- 其它内置外设功能
- 电气特性
- 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。标识“<R>”处是主要修改的地方。在 PDF 文件中可以使用查找“<R>”，来很容易的找到修改的地方。
- 如何解释寄存器格式：
→ 括号中的位名在 RA78K0 中被定义为保留字，并且在 CC78K0 中直接通过 # pragma sfr 定义为一个 sfr 变量。
- 如果读者知道寄存器名字并想验证这个寄存器的详细信息：
→ 可参阅 **附录 B 寄存器索引**。
- 如果读者希望了解 78K/0 系列指令的详细信息：
→ 可参阅 **78K/0 系列指令用户手册 (U12326E)**。

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:	$\overline{\text{xxx}}$ (在引脚和信号名称上划一条线)
注:	文中用 注 标注的相关术语的脚注
注意事项:	需要特别关注的信息
备注:	补充信息
数的表示法:	二进制 ... xxxx 或 xxxx B
	十进制 ... xxxx
	十六进制 ... xxxx H

相关文档

本手册中指出的相关文档可能包括了初级的版本, 但未注明。

设备文档

文档名称	文档编号
78K0/LF2 用户手册	本手册
78K/0 系列指令用户手册	U12326E

开发工具相关文档 (软件)(用户手册)

文档名称	文档编号	
RA78K0 Ver. 3.80 汇编包	操作	U17199E
	语言	U17198E
	结构化汇编语言	U17197E
CC78K0 Ver. 3.70 C 编译器	操作	U17201E
	语言	U17200E
ID78K0-QB Ver. 2.90 集成调试器	操作	U17437E
PM + Ver. 5.20		U16934E

开发工具相关文档 (硬件)(用户手册)

文档名称	文档编号
QB-78K0LX2 在线仿真器	U17468E
QB-78K0MINI 片上调试仿真器	U17029E

与 Flash 存储器编程相关文档 (用户手册)

文档名称	文档编号
PG-FP4 存储器编程器用户手册	U15260E
PG-FPL3 存储器编程器用户手册	U17454E

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

其他文档

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

注 浏览“半导体设备安装手册”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

第一章 概述.....	17
1.1 特性	17
1.2 应用	18
1.3 订购信息.....	19
1.4 引脚配置(俯视图).....	20
1.5 78K0/Lx2 系列产品	23
1.6 框图	25
1.7 功能概述(μ PD78F037x)	26
1.8 功能概述(μ PD78F038x)	28
第二章 引脚功能	30
2.1 引脚功能列表.....	30
2.2 引脚功能描述.....	34
2.2.1 P00, P01 (端口 0).....	34
2.2.2 P10 ~ P17 (端口 1).....	34
2.2.3 P20 ~ P27 (端口 2) (仅限于 μ PD78F037x)	35
2.2.4 P30 ~ P33 (端口 3).....	35
2.2.5 P60, P61 (端口 6).....	36
2.2.6 P70 ~ P76 (端口 7).....	37
2.2.7 P120 ~ P124 (端口 12).....	37
2.2.8 AV _{REF} (仅限于 μ PD78F037x)	38
2.2.9 AV _{SS} (仅限于 μ PD78F037x).....	38
2.2.10 S0 ~ S25	38
2.2.11 S26 ~ S35 (仅限于 μ PD78F038x)	38
2.2.12 COM0 ~ COM3.....	38
2.2.13 LV _{DD}	38
2.2.14 LV _{SS}	38
2.2.15 V _{LC0} ~ V _{LC2}	38
2.2.16 CAPH, CAPL	38
2.2.17 $\overline{\text{RESET}}$	38
2.2.18 REGC	38
2.2.19 V _{DD}	39
2.2.20 V _{SS}	39
2.2.21 FLMD0	39
2.3 引脚 I/O 电路和未使用引脚的建议连接方式	40
第三章 CPU 结构	44
3.1 存储器空间	44
3.1.1 内部程序存储器空间.....	51
3.1.2 内部数据存储器空间.....	52
3.1.3 特殊功能寄存器 (SFR) 区域.....	53
3.1.4 数据存储器寻址.....	53
3.2 处理器寄存器.....	59
3.2.1 控制寄存器.....	59

3.2.2 通用寄存器	63
3.2.3 特殊功能寄存器(SFR)	64
3.3 指令地址寻址	69
3.3.1 相对寻址	69
3.3.2 立即寻址	70
3.3.3 表间接寻址	71
3.3.4 寄存器寻址	71
3.4 操作数地址寻址	72
3.4.1 隐含寻址	72
3.4.2 寄存器寻址	73
3.4.3 直接寻址	74
3.4.4 短直接寻址	75
3.4.5 特殊功能寄存器 (SFR) 寻址	76
3.4.6 寄存器间接寻址	77
3.4.7 基址寻址	78
3.4.8 基址变址寻址	79
3.4.9 堆栈寻址	80
3.5 存储器 bank (仅限于μPD78F0376, 78F0376D, 78F0386 和 78F0386D)	81
3.6 存储器 bank 选择寄存器 (BANK) (仅限于μPD78F0376, 78F0376D, 78F0386 和 78F0386D)	82
3.7 选择存储器 bank (仅限于μPD78F0376, 78F0376D, 78F0386 和 78F0386D)	83
3.7.1 存储器 bank 之间的参考值	83
3.7.2 存储器 bank 间的跳转指令	85
3.7.3 存储器 bank 间的子程序调用	87
3.7.4 通过中断指令跳转到 bank 区域	89
第四章 端口功能	91
4.1 端口功能	91
4.2 端口配置	93
4.2.1 端口 0	94
4.2.2 端口 1	96
4.2.3 端口 2 (仅限于 μ PD78F037x)	101
4.2.4 端口 3	102
4.2.5 端口 6	104
4.2.6 端口 7	105
4.2.7 端口 12	106
4.3 控制端口功能的寄存器	108
4.4 端口功能操作	112
4.4.1 写入 I/O 端口	112
4.4.2 读取 I/O 端口	112
4.4.3 I/O 端口的操作	112
4.5 使用复用功能时端口模式寄存器和输出锁存器的设置	113
第五章 时钟发生器	115
5.1 时钟发生器的功能	115
5.2 时钟发生器的配置	116
5.3 控制时钟发生器的寄存器	118
5.4 系统时钟振荡器	127
5.4.1 X1 振荡器	127

5.4.2	XT1 振荡器.....	127
5.4.3	不使用副系统时钟.....	130
5.4.4	内部高速振荡器.....	130
5.4.5	内部低速振荡器.....	130
5.4.6	预分频器.....	130
5.5	时钟发生器的操作.....	131
5.6	时钟控制.....	134
5.6.1	高速系统时钟控制示例.....	134
5.6.2	控制内部高速振荡时钟示例.....	137
5.6.3	控制副系统时钟示例.....	139
5.6.4	控制内部低速振荡时钟示例.....	141
5.6.5	CPU 和外围硬件所采用的时钟.....	141
5.6.6	CPU 时钟状态转换图.....	142
5.6.7	CPU 时钟转换之前的状况与转换之后的处理.....	147
5.6.8	CPU 时钟和主系统时钟切换所需的时间.....	148
5.6.9	时钟振荡停止前的状况.....	149
5.6.10	外围硬件与源时钟.....	150
第六章	16 位定时器/事件计数器 00 和 01.....	151
6.1	16 位定时器/事件计数器 00 和 01 的功能.....	151
6.2	16 位定时器/事件计数器 00 和 01 的配置.....	152
6.3	控制 16 位定时器/事件计数器 00 和 01 的寄存器.....	158
6.4	16 位定时器/事件计数器 00 和 01 的操作.....	166
6.4.1	间隔定时器的操作.....	166
6.4.2	PPG 输出操作(仅限于定时器 00).....	169
6.4.3	脉冲宽度测量操作(仅限于定时器 00).....	172
6.4.4	外部事件计数器的操作(仅限于定时器 00).....	180
6.4.5	方波输出操作(仅限于定时器 00).....	183
6.4.6	单脉冲输出操作(仅限于定时器 00).....	185
6.5	16 位定时器/事件计数器 00 和 01 的注意事项.....	190
第七章	8 位定时器/事件计数器 50 和 51.....	193
7.1	8 位定时器/事件计数器 50 和 51 的功能.....	193
7.2	8 位定时器/事件计数器 50 和 51 的配置.....	193
7.3	控制 8 位定时器/事件计数器 50 和 51 的寄存器.....	196
7.4	8 位定时器/事件计数器 50 和 51 的操作.....	201
7.4.1	用作间隔定时器.....	201
7.4.2	用作外部事件计数器.....	203
7.4.3	方波输出操作.....	204
7.4.4	PWM 输出操作.....	205
7.5	使用 8 位定时器/事件计数器 50 和 51 的注意事项.....	209
第八章	8 位定时器 H0 和 H1.....	210
8.1	8 位定时器 H0 和 H1 的功能.....	210
8.2	8 位定时器 H0 和 H1 的配置.....	210
8.3	控制 8 位定时器 H0 和 H1 的寄存器.....	214
8.4	8 位定时器 H0 和 H1 的操作.....	219

8.4.1 间隔定时器/方波输出操作.....	219
8.4.2 PWM 输出操作	222
8.4.3 载波发生器操作(仅限于 8 位定时器 H1)	228
第九章 钟表定时器.....	235
9.1 钟表定时器的功能.....	235
9.2 钟表定时器的配置.....	236
9.3 控制钟表定时器的寄存器.....	237
9.4 钟表定时器操作	239
9.4.1 钟表定时器操作	239
9.4.2 间隔定时器操作	239
9.5 钟表定时器使用注意事项.....	240
第十章 看门狗定时器	241
10.1 看门狗定时器的功能	241
10.2 看门狗定时器的配置	242
10.3 控制看门狗定时器的寄存器	243
10.4 看门狗定时器的操作	244
10.4.1 看门狗定时器操作的控制.....	244
10.4.2 看门狗定时器溢出时间的设置.....	245
10.4.3 看门狗定时器窗口打开周期的设置	246
第十一章 时钟输出控制器	248
11.1 时钟输出控制器的功能.....	248
11.2 时钟输出控制器的配置.....	249
11.3 控制时钟输出控制器的寄存器.....	249
11.4 时钟输出控制器的操作.....	251
第十二章 A/D 转换器(仅限于μPD78F037x).....	252
12.1 A/D 转换器的功能.....	252
12.2 A/D 转换器的配置.....	253
12.3 A/D 转换器使用的寄存器.....	255
12.4 A/D 转换器的操作.....	263
12.4.1 A/D 转换器的基本操作.....	263
12.4.2 输入电压和转换结果.....	265
12.4.3 A/D 转换器操作模式.....	266
12.5 A/D 转换器特征表的阅读方法.....	268
12.6 A/D 转换器使用注意事项.....	270
第十三章 串行接口 UART0	274
13.1 串行接口 UART0 的功能.....	274
13.2 串行接口 UART0 的配置.....	275
13.3 控制串行接口 UART0 的寄存器.....	278
13.4 串行接口 UART0 的操作.....	283
13.4.1 操作停止模式.....	283
13.4.2 异步串行接口(UART)模式	284

13.4.3 专用波特率发生器	290
第十四章 串行接口 UART6.....	295
14.1 串行接口 UART6 的功能	295
14.2 串行接口 UART6 的配置	299
14.3 控制串行接口 UART6 的寄存器	302
14.4 串行接口 UART6 的操作	311
14.4.1 操作停止模式	311
14.4.2 异步串行接口(UART)模式	312
14.4.3 专用波特率发生器	325
第十五章 串行接口 CSI10.....	332
15.1 串行接口 CSI10 的功能.....	332
15.2 串行接口 CSI10 的配置.....	332
15.3 控制串行接口 CSI10 的寄存器	334
15.4 串行接口 CSI10 的操作.....	337
15.4.1 操作停止模式	337
15.4.2 3 线串行 I/O 模式.....	338
第十六章 串行接口 IIC0.....	349
16.1 串行接口 IIC0 的功能.....	349
16.2 串行接口 IIC0 的配置.....	352
16.3 控制串行接口 IIC0 的寄存器.....	355
16.4 I²C 总线模式功能.....	368
16.4.1 引脚配置	368
16.5 I²C 总线定义和控制方法.....	369
16.5.1 起始条件	369
16.5.2 地址.....	370
16.5.3 传送方向指示	370
16.5.4 应答信号($\overline{\text{ACK}}$).....	371
16.5.5 停止条件	372
16.5.6 等待.....	373
16.5.7 取消等待	375
16.5.8 中断请求(INTIIC0)产生时序和等待控制.....	375
16.5.9 地址相等的检测方法.....	376
16.5.10 错误检测	376
16.5.11 扩展码.....	377
16.5.12 仲裁.....	378
16.5.13 唤醒功能	379
16.5.14 通信预约	380
16.5.15 其他注意事项	383
16.5.16 通信操作	385
16.5.17 I ² C 中断请求(INTIIC0)产生时序.....	392
16.6 时序图.....	413
16.7 与 LCD 控制器/驱动器的通信	420
16.7.1 系统配置	420
16.7.2 写操作	421

16.7.3 读操作	424
第十七章 LCD 控制器/驱动器	428
17.1 LCD 控制器/驱动器的功能	428
17.2 LCD 控制器/驱动器的配置	430
17.3 控制 LCD 控制器/驱动器的寄存器	433
17.4 设置 LCD 控制器/驱动器	439
17.5 LCD 显示数据存储	441
17.6 Common 和 Segment 信号	442
17.7 显示模式	446
17.7.1 静态显示示例	446
17.7.2 2 分时显示示例	449
17.7.3 3 分时显示示例	452
17.7.4 4 分时显示示例	456
17.8 LCD 驱动提供电压 V_{LC0} , V_{LC1} 和 V_{LC2}	459
17.8.1 内部电阻分压模式	459
17.8.2 外部电阻分压模式	461
17.8.3 内部电压泵模式	462
第十八章 乘法器/除法器(仅限于 μPD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D)	463
18.1 乘法器/除法器的功能	463
18.2 乘法器/除法器的配置	463
18.3 控制乘法器/除法器的寄存器	467
18.4 乘法器/除法器的操作	468
18.4.1 乘法操作	468
18.4.2 除法操作	470
第十九章 中断功能	472
19.1 中断功能类型	472
19.2 中断源及配置	472
19.3 控制中断功能的寄存器	476
19.4 中断服务操作	484
19.4.1 可屏蔽的中断响应	484
19.4.2 软件中断请求响应	486
19.4.3 中断嵌套	487
19.4.4 保持中断请求	490
第二十章 按键中断功能	491
20.1 按键中断的功能	491
20.2 按键中断的配置	491
20.3 控制按键中断的寄存器	492
第二十一章 待机功能	493
21.1 待机功能及配置	493
21.1.1 待机功能	493

21.1.2 控制待机功能的寄存器	493
21.2 待机功能的操作	496
21.2.1 HALT 模式	496
21.2.2 STOP 模式	501
第二十二章 复位功能	506
22.1 确认复位源的寄存器	514
第二十三章 上电清零电路	515
23.1 上电清零电路的功能	515
23.2 上电清零电路的配置	516
23.3 上电清零电路的操作	516
23.4 上电清零电路使用注意事项	519
第二十四章 低电压检测电路	521
24.1 低电压检测电路的功能	521
24.2 低电压检测电路的配置	521
24.3 控制低电压检测电路的寄存器	522
24.4 低电压检测电路的操作	525
24.4.1 用于复位	526
24.4.2 用于中断	531
24.5 低电压检测电路的注意事项	536
第二十五章 选项字节	539
25.1 选项字节的功能	539
25.2 选项字节的格式	540
第二十六章 Flash 存储器	543
26.1 内部存储器容量切换寄存器	543
26.2 内部扩展 RAM 容量切换寄存器	544
26.3 用 Flash 编程器写入数据	544
26.4 编程环境	550
26.5 通信模式	550
26.6 在线方式的引脚处理	552
26.6.1 FLMD0 引脚	552
26.6.2 串行接口引脚	552
26.6.3 RESET 引脚	554
26.6.4 端口引脚	554
26.6.5 REGC 引脚	554
26.6.6 其它信号引脚	554
26.6.7 供电电压	554
26.7 编程方法	555
26.7.1 控制 Flash 存储器	555
26.7.2 Flash 存储器编程模式	555
26.7.3 选择通信模式	556
26.7.4 通信命令	557

26.8 安全性设置	558
26.9 通过自写入进行 Flash 存储器编程	560
26.9.1 引导交换功能	563
第二十七章 片上调试功能(仅限于 μ PD78F0376D 和 78F0386D)	565
27.1 片上调试安全 ID	566
第二十八章 指令集	567
28.1 操作列表使用规则	567
28.1.1 操作数标识符和标识方法	567
28.1.2 操作栏描述	568
28.1.3 标志操作栏的描述	568
28.2 操作列表	569
28.3 按寻址类型列出指令	577
第二十九章 电气特性	580
第三十章 封装图	601
第三十一章 等待注意事项	603
31.1 等待注意事项	603
31.2 产生等待的外围硬件	604
附录 A 开发工具	605
A.1 软件包	608
A.2 语言处理软件	608
A.3 控制软件	609
A.4 Flash 存储器写入工具	609
A.5 调试工具(硬件)	610
A.5.1 当使用在线仿真器 QB-78K0LX2 时	610
A.5.2 当使用片上调试仿真器 QB-78K0MINI 时	611
A.6 调试工具(软件)	611
附录 B 寄存器索引	612
B.1 寄存器索引 (按寄存器名称的字母顺序排列)	612
B.2 寄存器索引 (按寄存器符号的字母顺序排列)	616
附录 C 修订历史	620
C.1 本版本的主要修订	620

第一章 概述

1.1 特征

- 指令最短执行时间可以在高速 (0.1 μ s: @ 高速系统时钟的操作频率为 20 MHz) 和超低速 (122 μ s: @ 副系统时钟的操作频率为 32.768 kHz) 之间改变
- 通用寄存器: 8 位 x 32 个寄存器 (8 位 x 8 个寄存器 x 4 bank)
- ROM, RAM 的容量

产品型号	程序存储器 (ROM)		数据存储器		
			内部高速 RAM ^注	内部扩展 RAM ^注	LCD 显示 RAM
μ PD78F0372, 78F0382	Flash 存储器 ^注	24 KB	1 KB	-	26 x 4 位 (μ PD78F037x) 36 x 4 位 (μ PD78F038x)
μ PD78F0373, 78F0383		32 KB			
μ PD78F0374, 78F0384		48 KB		1 KB	
μ PD78F0375, 78F0385		60 KB		2 KB	
μ PD78F0376, 78F0386, 78F0376D, 78F0386D		96 KB		4 KB	

注 通过使用内部存储器容量切换寄存器(IMS)和内部扩展 RAM 容量切换寄存器 (IXS)，可以改变内部 Flash 存储器、内部高速 RAM 的容量和内部扩展 RAM 的容量。

- 内置单电源 Flash 存储器
- 自编程(具有启动交换功能)
- 片上调试功能(仅限于 μ PD78F0376D 和 78F0386D)
- 内置上电清零 (POC) 电路和低电压检测器 (LVI)
- 内置看门狗定时器(使用内置的内部低速振荡时钟进行操作)
- LCD 控制器/驱动器(内部电压泵，外部电阻分压和内部电阻分压可切换使用)
 μ PD78F037x: Segment 信号: 26, Common 信号: 4
 μ PD78F038x: Segment 信号: 36, Common 信号: 4
- 内置乘法器/除法器 (16 位 x 16 位, 32 位 \div 16 位)
(仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386, 78F0386D)
- 内置按键中断功能: 8 通道
- 内置时钟输出控制器
- I/O 端口
 μ PD78F037x: 34
 μ PD78F038x: 26
- 定时器
 μ PD78F0372, 78F0373, 78F0382, 78F0383: 7 通道
 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386, 78F0386D: 8 通道
 - 16 位定时器/事件计数器: 2 通道^注
 - 8 位定时器/事件计数器: 2 通道
 - 8 位定时器: 2 通道
 - 钟表定时器: 1 通道
 - 看门狗定时器: 1 通道

注 μ PD78F0372, 78F0373, 78F0382 和 78F0383 仅有 1 通道。

- 串行接口: 3 通道
 - UART (LIN (本地互联网)总线支持): 1 通道
 - CSI/UART^注: 1 通道
 - I²C: 1 通道

注 可选择复用功能引脚的任意一种功能。

- 10 位分辨率 A/D 转换器: 8 通道(仅限于 μ PD78F037x)
- 电源电压: $V_{DD} = 1.8 \sim 5.5 \text{ V}$
- 运行工作环境温度: $TA = -40 \sim +85 \text{ }^\circ\text{C}$

1.2 应用

APS 照相机, 数字照相机, AV 设备, 家用电器, 等。

<R>

1.3 订购信息

• Flash 存储器版本 (无铅产品)

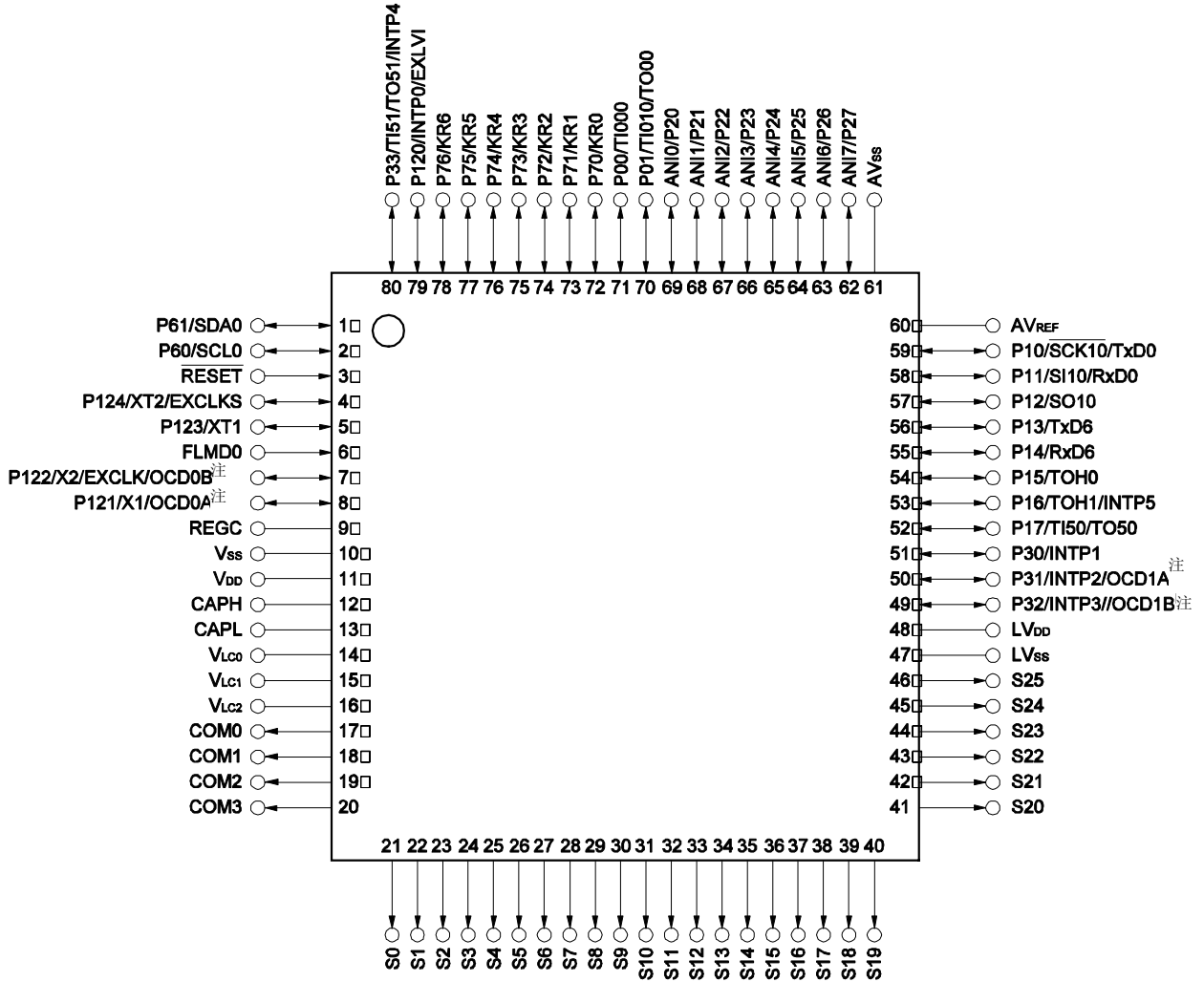
产品型号	封装
μPD78F0372GC-UBT-A	80 引脚塑封 LQFP (14 × 14)
μPD78F0373GC-UBT-A	80 引脚塑封 LQFP (14 × 14)
μPD78F0374GC-UBT-A	80 引脚塑封 LQFP (14 × 14)
μPD78F0375GC-UBT-A	80 引脚塑封 LQFP (14 × 14)
μPD78F0376GC-UBT-A	80 引脚塑封 LQFP (14 × 14)
μPD78F0376DGC-UBT-A ^注	80 引脚塑封 LQFP (14 × 14)
μPD78F0372GK-8EU-A	80 引脚塑封 LQFP (细脚距) (12 × 12)
μPD78F0373GK-8EU-A	80 引脚塑封 LQFP (细脚距) (12 × 12)
μPD78F0374GK-8EU-A	80 引脚塑封 LQFP (细脚距) (12 × 12)
μPD78F0375GK-8EU-A	80 引脚塑封 LQFP (细脚距) (12 × 12)
μPD78F0376GK-8EU-A	80 引脚塑封 LQFP (细脚距) (12 × 12)
μPD78F0376DGK-8EU-A ^注	80 引脚塑封 LQFP (细脚距) (12 × 12)
μPD78F0382GC-UBT-A	80 引脚塑封 LQFP (14 × 14)
μPD78F0383GC-UBT-A	80 引脚塑封 LQFP (14 × 14)
μPD78F0384GC-UBT-A	80 引脚塑封 LQFP (14 × 14)
μPD78F0385GC-UBT-A	80 引脚塑封 LQFP (14 × 14)
μPD78F0386GC-UBT-A	80 引脚塑封 LQFP (14 × 14)
μPD78F0386DGC-UBT-A ^注	80 引脚塑封 LQFP (14 × 14)
μPD78F0382GK-8EU-A	80 引脚塑封 LQFP (细脚距) (12 × 12)
μPD78F0383GK-8EU-A	80 引脚塑封 LQFP (细脚距) (12 × 12)
μPD78F0384GK-8EU-A	80 引脚塑封 LQFP (细脚距) (12 × 12)
μPD78F0385GK-8EU-A	80 引脚塑封 LQFP (细脚距) (12 × 12)
μPD78F0386GK-8EU-A	80 引脚塑封 LQFP (细脚距) (12 × 12)
μPD78F0386DGK-8EU-A ^注	80 引脚塑封 LQFP (细脚距) (12 × 12)

注 μPD78F0376D 和 78F0386D 具有片上调试功能。从重复写入 Flash 存储器的次数有限看，在使用片上调试功能后它的可靠性并不能得到保证，所以大规模生产时不要使用本产品。NEC 电子不接受对于本产品的任何投诉。

1.4 引脚配置(俯视图)

(1) μ PD78F0372, 78F0373, 78F0374, 78F0375, 78F0376, 78F0376D

- 80 引脚塑封 LQFP (14 x14)
- 80 引脚塑封 LQFP (细脚距) (12 x 12)



注 仅限于(PD78F0376D (具有片上调试功能的产品))。

注意事项 1. 使 AV_{SS} 与 V_{SS} 连接。

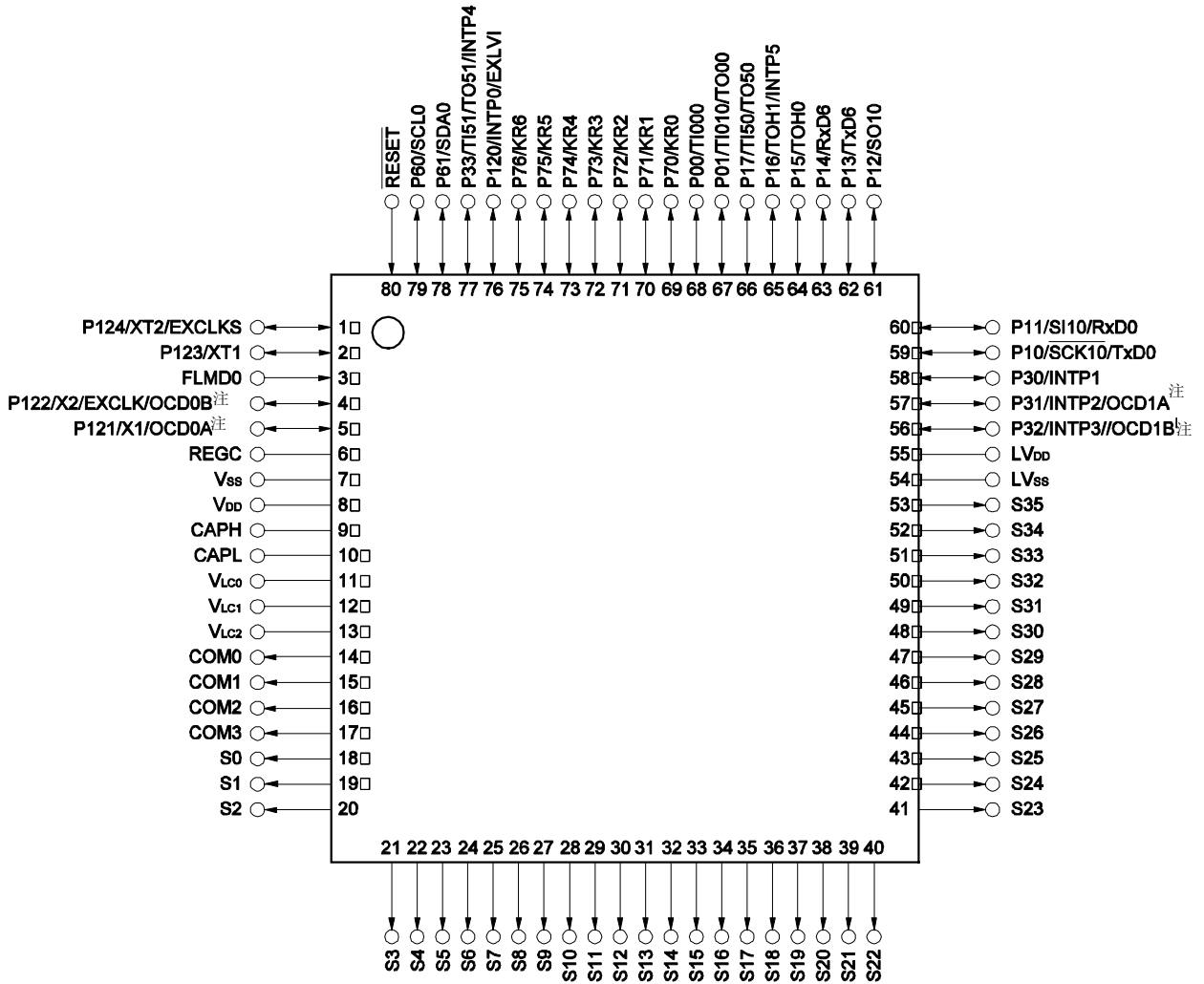
2. 通过一个电容(0.47 μ F ~ 1 μ F: 推荐)将 REGC 引脚和 VSS 连接到一起。

3. 复位后, ANI0/P20 到 ANI7/P27 设置为模拟输入模式。

<R>

(2) μ PD78F0382, 78F0383, 78F0384, 78F0385, 78F0386, 78F0386D

- 80 引脚塑封 LQFP (14 × 14)
- 80 引脚塑封 LQFP (细脚距) (12 × 12)



注 仅限于 μ PD78F0386D (具有片上调试功能的产品)。

<R> 注意事项 通过一个电容(0.47 μ F ~ 1 μ F: 推荐)将 REGC 引脚和 V_{SS}连接到一起。

引脚标识

ANIO ~ ANI7 ^{注1} :	模拟输入	REGC	调校电容
AV _{REF} ^{注1} :	模拟参考电压	RESET:	复位
AV _{SS} ^{注1} :	模拟地	RxD0, RxD6:	接收数据
CAPH, CAPL:	LCD 电源电容控制	S0 ~ S25:	Segment 输出
COM0 ~ COM3:	Common 输出	S26 ~ S35 ^{注2} :	Segment 输出
EXCLK:	外部时钟输入 (主系统时钟)	SCK10:	串行时钟输入/输出
EXCLKS:	外部时钟输入 (副系统时钟)	SCL0:	串行时钟输入/输出
EXLVI:	外部电压输入 用于低电压检测器	SDA0:	串行数据输入/输出
FLMD0:	Flash 编程模式	SI10:	串行数据输入
INTP0 ~ INTP5:	外部中断输入	SO10:	串行数据输出
KR0 ~ KR6:	按键返回	TI000, TI010:	定时器输入
LV _{DD} :	LCD 控制器/驱动器的电源	TI50, TI51:	定时器输入
LV _{SS} :	LCD 控制器/驱动器的地	TO00:	定时器输出
OCD0A ^{注3} , OCD0B ^{注3} :	片上调试输入/输出	TO50, TO51:	定时器输出
OCD1A ^{注3} , OCD1B ^{注3} :	片上调试输入/输出	TOH0, TOH1:	定时器输出
P00, P01:	端口 0	TxD0, TxD6:	发送数据
P10 ~ P17:	端口 1	V _{DD} :	电源电压
P20 ~ P27 ^{注1} :	端口 2	V _{SS} :	地
P30 ~ P33:	端口 3	V _{LC0} ~ V _{LC2} :	LCD 电源
P60, P61:	端口 6	X1, X2:	晶体振荡器 (主系统时钟)
P70 ~ P76:	端口 7	XT1, XT2:	晶体振荡器 (副系统时钟)
P120 ~ P124:	端口 12		

- 注
1. 仅限于 μ PD78F037x。
 2. 仅限于 μ PD78F038x。
 3. 仅限于 μ PD78F0376D 和 78F0386D。

<R>

1.5 78K0/Lx2 系列介绍

ROM	RAM	78K0/LE2	78K0/LF2	78K0/LG2
		64 引脚	80 引脚	100 引脚
128 KB	7 KB	–	–	μ PD78F0397D ^注 μ PD78F0397
96 KB	5 KB	–	μ PD78F0386D 注 μ PD78F0376D 注 μ PD78F0386 μ PD78F0376	μ PD78F0396
60 KB	3 KB	–	μ PD78F0385 μ PD78F0375	μ PD78F0395
48 KB	2 KB	–	μ PD78F0384 μ PD78F0374	μ PD78F0394
32 KB	1 KB	μ PD78F0363D 注 μ PD78F0363	μ PD78F0383 μ PD78F0373	μ PD78F0393
24 KB	1 KB	μ PD78F0362	μ PD78F0382 μ PD78F0372	–
16 KB	768 B	μ PD78F0361	–	–

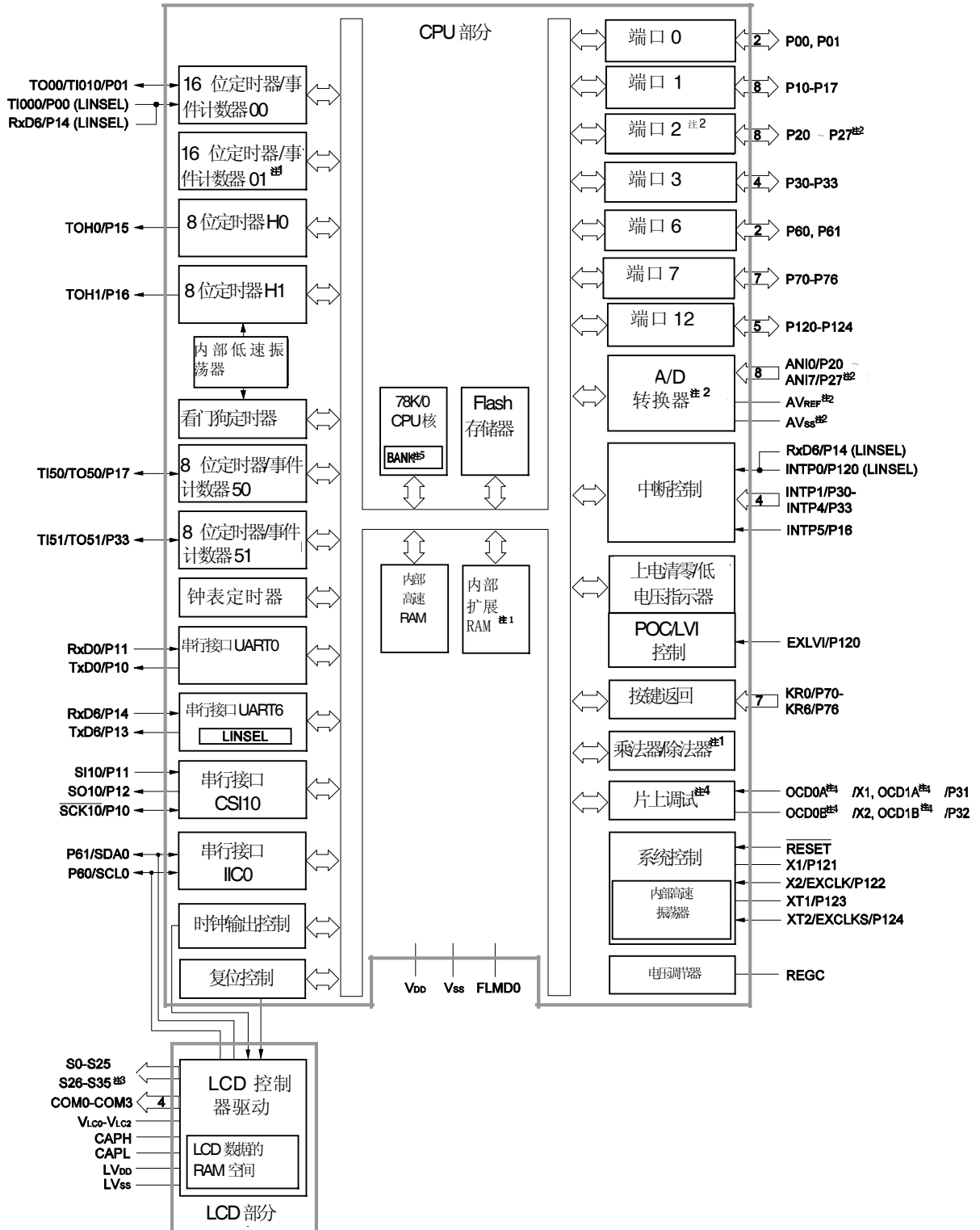
注 具有片上调试功能的产品。

78K0/Lx2 系列功能列表如下:

项目		78K0/LE2			78K0/LF2									78K0/LG2						
		μ PD78F036x			μ PD78F037x				μ PD78F038x					μ PD78F039x						
		64 引脚			80 引脚									100 引脚						
Flash 存储器 (KB)		16	24	32	24	32	48	60	96	24	32	48	60	96	32	48	60	96	128	
RAM (KB)		0.7 5	1	1	1	1	2	3	5	1	1	2	3	5	1	2	3	5	7	
Bank (flash 存储器)		-							4	-					4	-			4	6
电源电压		$V_{DD} = 1.8 \sim 5.5 V$																		
稳压器		提供																		
指令最短执行时间		0.1 μ s (20 MHz: $V_{DD} = 4.0 \sim 5.5 V$)/0.2 μ s (10 MHz: $V_{DD} = 2.7 \sim 5.5 V$)/ 0.4 μ s (5 MHz: $V_{DD} = 1.8 \sim 5.5 V$)																		
时钟	高速系统时钟	20 MHz: $V_{DD} = 4.0 \sim 5.5 V$ /10 MHz: $V_{DD} = 2.7 \sim 5.5 V$ /5 MHz: $V_{DD} = 1.8 \sim 5.5 V$																		
	内部高速振荡时钟	8 MHz (TYP.): $V_{DD} = 1.8 \sim 5.5 V$																		
	副时钟	32.768 kHz (TYP.): $V_{DD} = 1.8 \sim 5.5 V$																		
	内部低速振荡时钟	240 kHz (TYP.): $V_{DD} = 1.8 \sim 5.5 V$																		
端口	总计	24			34				26					40						
	16 位 (TM0)	1 通道			2 通道				1 通道		2 通道			1 通道	2 通道					
定时器	8 位 (TM5)	2 通道																		
	8 位 (TMH)	2 通道																		
	钟表	1 通道																		
	WDT (看门狗定时器)	1 通道																		
	3 线 CSI	-															1 通道			
串行接口	3 线 CSI/UART 注	1 通道																		
	UART 支持 LIN 总线	1 通道																		
	I ² C 总线	1 通道																		
	类型	内部升压, 外部电阻分压和内部电阻分压可切换使用																		
LCD	Segment 信号	20			26				36					40						
	Common 信号	4																		
	10 位 A/D	5 通道			8 通道				-					8 通道						
中断	外部	6			7															
	内部	16			18				15		17			16	19					
按键中断		-			7 通道									8 通道						
复位	RESET 引脚	提供																		
	POC	1.59 V \pm 0.15 V (上升到 1.8 V 所用时间 : 3.6 ms (MAX.))																		
	LVI	可选择 16 个电源电压检测级别																		
	WDT	提供																		
时钟输出		提供																		
乘法器/除法器		-			提供				-					提供		-			提供	
片上调试功能		仅限于 μ PD78F0363D			仅限于 μ PD78F0376D				仅限于 μ PD78F0386D					仅限于 μ PD78F0397D						
工作环境温度		$TA = -40 \sim +85^{\circ}C$																		

注 可选择复用功能引脚的任意一种功能。

<R> 1.6 框图



- 注
1. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386, 78F0386D。
 2. 仅限于 μ PD78F037x。
 3. 仅限于 μ PD78F038x。
 4. 仅限于 μ PD78F0376D 和 78F0386D。
 5. 仅限于 μ PD78F0376, 78F0376D, PD78F0386, 78F0386D。

1.7 功能概述(μPD78F037x)

(1/2)

项目		μPD78F0372	μPD78F0373	μPD78F0374	μPD78F0375	μPD78F0376	μPD78F0376 D
内部存储器	Flash 存储器 (支持自编程) ^{注1}	24 KB	32 KB	48 KB	60 KB	96 KB	
	存储器 bank ^{注2}	-				4 bank	
	高速 RAM ^{注1}	1 KB					
	扩展 RAM ^{注1}	-		1 KB	2 KB	4 KB	
	LCD 显示 RAM	26 × 4 位					
存储器空间		64 KB					
主系统时钟 (振荡频率)	高速系统时钟	X1 (晶体/陶瓷) 振荡器, 外部主系统时钟输入(EXCLK)					
	内部高速振荡时钟	内部振荡器 8 MHz (TYP.): V _{DD} = 1.8 ~ 5.5 V					
副系统时钟 (振荡频率)		XT1 (晶体) 振荡器, 外部副时钟输入(EXCLKS) 32.768 kHz (TYP.): V _{DD} = 1.8 ~ 5.5 V					
内部低速振荡时钟(用于 TMH1, WDT)		内部振荡器 240 kHz (TYP.): V _{DD} = 1.8 ~ 5.5 V					
通用寄存器		8 位 × 32 寄存器(8 位 × 8 寄存器 × 4 bank)					
指令最短执行时间		0.1 μs (高速系统时钟: @ f _{XH} = 20 MHz 工作)					
		0.25 μs (内部高速振荡时钟: @ f _{RH} = 8 MHz (TYP.) 工作)					
		122 μs (副系统时钟: @ f _{SUB} = 32.768 kHz 工作)					
指令集		<ul style="list-style-type: none"> • 8 位操作, 16 位操作 • 乘法/除法 (8 位 × 8 位, 16 位 ÷ 8 位) • 位操作 (置位, 复位, 测试和布尔操作) • BCD 调整, 等 					
I/O 端口		CMOS I/O: 34					
定时器		• 16 位定时器/事件计数器: 1 通道		• 16 位定时器/事件计数器: 2 通道			
		• 8 位定时器/事件计数器: 2 通道		• 8 位定时器: 2 通道			
		• 钟表定时器: 1 通道		• 看门狗定时器: 1 通道			
	定时器输出	5 (PWM 输出: 4, PPG 输出: 1)		6 (PWM 输出: 4, PPG 输出: 2)			
时钟输出		<ul style="list-style-type: none"> • 156.25 kHz, 312.5 kHz (外围硬件时钟: @ f_{PRS} = 20 MHz 工作) • 32.768 kHz (副时钟: @ f_{SUB} = 32.768 kHz 工作) 					
A/D 转换器		10 位分辨率 × 8 通道(AV _{REF} = 2.3 ~ 5.5 V)					

注 1. 通过使用内部存储器容量切换寄存器(IMS)和内部扩展 RAM 容量切换寄存器 (IXS), 可以改变内部 Flash 存储器、内部高速 RAM 的容量和内部扩展 RAM 的容量。

2. 使用 bank 选择寄存器(BANK)可改变 bank。

项目		μPD78F0372	μPD78F0373	μPD78F0374	μPD78F0375	μPD78F0376	μPD78F0376 D
串行接口		<ul style="list-style-type: none"> • UART 支持 LIN 总线: 1 通道 • 3 线串行 I/O/UART[※]: 1 通道 • I²C 总线: 1 通道 					
LCD 控制器/驱动器		<ul style="list-style-type: none"> • 内部电压泵, 外部电阻分压和内部电阻分压可切换使用 • Segment 信号输出: 26 • Common 信号输出: 4 					
乘法器/除法器		- <ul style="list-style-type: none"> • 16 位 x 16 位 = 32 位 (乘法) • 32 位 ÷ 16 位 = 32 位 16 位余数 (除法) 					
向量中断源	内部	16		18			
	外部	7					
按键中断		按键输入引脚 (KR0 ~ KR6)的下降沿产生按键中断(INTKR)					
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 POC 电路内部复位 • 使用低电压检测电路内部复位 					
片上调试功能		-					提供
电源电压		V _{DD} = 1.8 ~ 5.5 V					
工作环境温度		TA = -40 ~ +85°C					
封装形式		<ul style="list-style-type: none"> • 80 引脚塑封 LQFP (14 × 14) • 80 引脚塑封 LQFP (细脚距) (12 × 12) 					

注 可选择复用功能引脚的任意一种功能。

定时器概要说明

		16 位定时器/ 事件计数器 00 和 01 ^{注1}		8 位定时器/ 事件计数器 50 和 51		8 位定时器 H0 和 H1		钟表定时器	看门狗定 时器
		TM00	TM01 ^{注1}	TM50	TM51	TMH0	TMH1		
功能	间隔定时器	1 通道	1 通道	1 通道	1 通道	1 通道	1 通道	1 通道 ^{注2}	(
	外部事件计数器	1 通道	1 通道	1 通道	1 通道	((((
	PPG 输出	1 输出	1 输出	-	-	-	-	-	-
	PWM 输出	-	-	1 输出	1 输出	1 输出	1 输出	-	-
	脉冲宽度测量	2 输入	2 输入	-	-	-	-	-	-
	方波输出	1 输出	1 输出	1 输出	1 输出	1 输出	1 输出	-	-
	载波发生器	-	-	-	-	-	1 输出 ^{注3}	-	-
	钟表定时器	-	-	-	-	-	-	1 通道 ^{注2}	-
看门狗定时器	-	-	-	-	-	-	-	1 通道	
中断源		2	2	1	1	1	1	1	-

注 1. 仅限于μPD78F0374, 78F0375, 78F0376 和 78F0376D。

2. 在钟表定时器中, 看门狗定时器功能和间隔定时器功能同时可用。

3. TM51 和 TMH1 可联合使用, 作为载波发生器模式。

<R> 1.8 功能概述(μPD78F038x)

(1/2)

项目		μPD78F0382	μPD78F0383	μPD78F0384	μPD78F0385	μPD78F0386	μPD78F0386 D
内部存储器	Flash 存储器 (支持自编程) ^{注1}	24 KB	32 KB	48 KB	60 KB	96 KB	
	存储器 bank ^{注2}	-				4 bank	
	高速 RAM ^注	1 KB					
	扩展 RAM ^注	-		1 KB	2 KB	4 KB	
	LCD 显示 RAM	36 × 4 位					
存储器空间		64 KB					
主系统时钟 (振荡频率)	高速系统时钟	X1 (晶体/陶瓷) 振荡器, 外部主系统时钟输入(EXCLK)					
	内部高速振荡时钟	内部振荡器 8 MHz (TYP.): V _{DD} = 1.8 ~ 5.5 V					
副系统时钟 (振荡频率)		XT1 (晶体) 振荡器, 外部副时钟输入(EXCLKS) 32.768 kHz (TYP.): V _{DD} = 1.8 ~ 5.5 V					
内部低速振荡时钟(用于 TMH1, WDT)		内部振荡器 240 kHz (TYP.): V _{DD} = 1.8 ~ 5.5 V					
通用寄存器		8 位 × 32 寄存器(8 位 × 8 寄存器 × 4 bank)					
指令最短执行时间		0.1 μs (高速系统时钟: @ f _{XH} = 20 MHz 工作)					
		0.25 μs (内部高速振荡时钟: @ f _{RH} = 8 MHz (TYP.) 工作)					
		122 μs (副系统时钟: @ f _{SUB} = 32.768 kHz 工作)					
指令集		<ul style="list-style-type: none"> • 8 位操作, 16 位操作 • 乘法/除法 (8 位 × 8 位, 16 位 ÷ 8 位) • 位操作 (置位, 复位, 测试和布尔操作) • BCD 调整, 等 					
I/O 端口		CMOS I/O: 26					
定时器		• 16 位定时器/事件计数器: 1 通道		• 16 位定时器/事件计数器: 2 通道			
		• 8 位定时器/事件计数器: 2 通道		• 8 位定时器: 2 通道			
定时器输出		• 钟表定时器: 1 通道		• 看门狗定时器: 1 通道			
		• 5 (PWM 输出: 4, PPG 输出: 1)		• 6 (PWM 输出: 4, PPG 输出: 2)			
时钟输出		<ul style="list-style-type: none"> • 156.25 kHz, 312.5 kHz (外围硬件时钟: @ f_{PRS} = 20 MHz 工作) • 32.768 kHz (副时钟: @ f_{SUB} = 32.768 kHz 工作) 					
A/D 转换器		-					

注 1. 通过使用内部存储器容量切换寄存器(IMS)和内部扩展 RAM 容量切换寄存器 (IXS), 可以改变内部 Flash 存储器、内部高速 RAM 的容量和内部扩展 RAM 的容量。

2. 使用 bank 选择寄存器(BANK)可改变 bank。

(2/2)

项目		μPD78F0382	μPD78F0383	μPD78F0384	μPD78F0385	μPD78F0386	μPD78F0386 D
串行接口		<ul style="list-style-type: none"> • UART 支持 LIN 总线: 1 通道 • 3 线串行 I/O/UART^注: 1 通道 • I²C 总线: 1 通道 					
LCD 控制器/驱动器		<ul style="list-style-type: none"> • 内部电压泵, 外部电阻分压和内部电阻分压可切换使用 • Segment 信号输出: 36 • Common 信号输出: 4 					
乘法器/除法器		- <ul style="list-style-type: none"> • 16 位 x 16 位 = 32 位 (乘法) • 32 位 ÷ 16 位 = 32 位 16 位余数 (除法) 					
向量中断源	内部	15		17			
	外部	7					
按键中断		按键输入引脚 (KR0 ~ KR6) 的下降沿产生按键中断(INTKR)					
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 POC 电路内部复位 • 使用低电压检测电路内部复位 					
片上调试功能		-					提供
电源电压		VDD = 1.8 ~ 5.5 V					
工作环境温度		TA = -40 ~ +85°C					
封装形式		<ul style="list-style-type: none"> • 80 引脚塑封 LQFP (14 × 14) • 80 引脚塑封 LQFP (细脚距) (12 × 12) 					

注 可选择复用功能引脚的任意一种功能。

定时器概要说明

		16 位定时器/ 事件计数器 00 和 01 ^{注1}		8 位定时器/ 事件计数器 50 和 51		8 位定时器 H0 和 H1		钟表定时器	看门狗定 时器
		TM00	TM01 ^{注1}	TM50	TM51	TMH0	TMH1		
功能	间隔定时器	1 通道	1 通道	1 通道	1 通道	1 通道	1 通道	1 通道 ^{注2}	-
	外部事件计数器	1 通道	1 通道	1 通道	1 通道	-	-	-	-
	PPG 输出	1 输出	1 输出	-	-	-	-	-	-
	PWM 输出	-	-	1 输出	1 输出	1 输出	1 输出	-	-
	脉冲宽度测量	2 输入	2 输入	-	-	-	-	-	-
	方波输出	1 输出	1 输出	1 输出	1 输出	1 输出	1 输出	-	-
	载波发生器	-	-	-	-	-	1 输出 ^{注3}	-	-
	钟表定时器	-	-	-	-	-	-	1 通道 ^{注2}	-
看门狗定时器	-	-	-	-	-	-	-	1 通道	
中断源		2	2	1	1	1	1	1	-

- 注
1. 仅限于 μPD78F0384, 78F0385, 78F0386 和 78F03786D。
 2. 在钟表定时器中, 看门狗定时器功能和间隔定时器功能同时可用。
 3. TM51 和 TMH1 可联合使用, 作为载波发生器模式。

第二章 引脚功能

2.1 引脚功能列表

有三种类型的引脚 I/O 缓冲器电源：AV_{REF}^{注1}，LV_{DD}和V_{DD}。下表显示了这些供电电源与引脚之间的关系。

表 2-1. 引脚 I/O 缓冲器供电电源

电源电压	对应引脚
AV _{REF} ^{注1}	P20 ~ P27 ^{注1}
LV _{DD}	CAPH, CAPL, COM0 ~ COM3, S0 ~ S25, S26 ~ S35 ^{注2} , V _{LC0} ~ V _{LC2}
V _{DD}	其它引脚

- 注
1. 仅限于 μ PD78F037x。
 2. 仅限于 μ PD78F038x。

(1) 端口引脚 (1/2)

引脚名称	I/O	功能	复位后	复用功能
P00	I/O	端口 0 2 位 I/O 端口。 可以位选输入/输出模式。 通过软件设置，可以定义内置上拉电阻的使用。	输入	TI000
P01				TI010/TO00
P10	I/O	端口 1 8 位 I/O 端口。 可以位选输入/输出模式。 通过软件设置，可以定义内置上拉电阻的使用。	输入	SCK10/TxD0
P11				SI10/RxD0
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50
P20 ~ P27 ^{注1}	I/O	端口 2 8 位 I/O 端口。 可以位选输入/输出模式。	输入	ANI0 ~ ANI7 ^{注1}
P30	I/O	端口 3 4 位 I/O 端口。 可以位选输入/输出模式。 通过软件设置，可以定义内置上拉电阻的使用。	输入	INTP1 ~ INTP3
P31				INTP2/OCD1A ^{注2}
P32				INTP3/OCD1B ^{注2}
P33				INTP4/TI51/TO51
P60	I/O	端口 6 2 位 I/O 端口。 输出为 N 沟开漏输出 (6V 耐压)。 可以位选输入/输出模式。	输入	SCL0
P61				SDA0

- 注
1. 仅限于 μ PD78F037x。
 2. 仅限于 μ PD78F0376D 和 78F0386D。

(1) 端口引脚 (2/2)

引脚名称	I/O	功能	复位后	复用功能
P70 ~ P76	I/O	端口 7 7 位 I/O 端口。 可以位选输入/输出模式。 通过软件设置，可以定义内置上拉电阻的使用。	输入	KR0 ~ KR6
P120	I/O	端口 12 5 位 I/O 端口。 可以位选输入/输出模式。 只有 P120 引脚可以通过软件设置，可以定义内置上拉电阻的使用。	输入	INTP0/EXLVI
P121				X1/OCD0A ^注
P122				X2/EXCLK/OCD0B ^注
P123				XT1
P124				XT2/EXCLKS

注 仅限于 μ PD78F0376D 和 78F0386D。

(2) 非端口引脚 (1/2)

引脚名称	I/O	功能	复位后	复用功能
INTP0	输入	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。	输入	P120/EXLVI
INTP1				P30
INTP2				P31/OCD1A ^{※1}
INTP3				P32/OCD1B ^{※1}
INTP4				P33/TI51/TO51
INTP5				P16/TOH1
SI10	输入	串行接口的串行数据输入	输入	P11/RxD0
SO10	输出	串行接口的串行数据输出	输入	P12
SDA0	I/O	串行接口的串行数据 I/O	输入	P61
SCK10	I/O	用于串行接口的时钟输入/输出	输入	P10/TxD0
SCL0				P60
RxD0	输入	串行数据输入到异步串行接口	输入	P11/SI10
RxD6				P14
TxD0	输出	从异步串行接口的串行数据输出	输入	P10/SCK10
TxD6				P13
TI000	输入	16 位定时器/事件计数器 00 的外部计数时钟输入 16 位定时器/事件计数器 00 的捕捉寄存器(CR000、CR010)的捕捉触发输入	输入	P00
TI010		16 位定时器/事件计数器 00 的捕捉寄存器(CR000)的捕捉触发输入		P01/TO00
TO00	输出	16 位定时器/事件计数器 00 输出	输入	P01/TI010
TI50	输入	8 位定时器/事件计数器 50 的外部计数时钟输入	输入	P17/TO50
TI51		8 位定时器/事件计数器 51 的外部计数时钟输入		P33/TO51/INTP4
TO50	输出	8 位定时器/事件计数器 50 输出	输入	P17/TI50
TO51		8 位定时器/事件计数器 51 输出		P33/TI51/INTP4
TOH0		8 位定时器 H0 输出		P15
TOH1		8 位定时器 H1 输出		P16/INTP5
ANI0 ~ ANI7 ^{※2}	输入	A/D 转换器模拟输入	输入	P20 ~ P27 ^{※2}
AV _{REF} ^{※2}	输入	A/D 转换器的参考电压输入和端口 2 的正向供电电源	-	-
AV _{SS} ^{※2}	-	A/D 转换器的地电位。应与 V _{SS} 的电位相同	-	-
S0 ~ S25	输出	LCD 控制器/驱动器段信号输出	输出	-
S26 ~ S35 ^{※3}				
COM0 ~ COM3	输出	LCD 控制器/驱动器共用端子信号输出	输出	-
LV _{DD}	-	LCD 控制器/驱动器的正向电源	-	-
LV _{SS}	-	LCD 控制器/驱动器的电势地	-	-
V _{LC0} ~ V _{LC2}	-	LCD 驱动电压	-	-

- 注
1. 仅限于 μ PD78F0376D 和 78F0386D。
 2. 仅限于 μ PD78F037x。
 3. 仅限于 μ PD78F038x。

(2) 非端口引脚 (2/2)

<R>

引脚名称	I/O	功能	复位后	复用功能
CAPH	-	LCD 驱动电压调节电容连接	-	-
CAPL	-			
KR0 ~ KR6	输入	按键中断输入	输入	P70 ~ P76
REGC	-	连接至用于内部操作的稳压器输出稳定电容 通过一个电容器 (0.47 μF ~ 1 μF : 推荐) 连接至 V_{SS} 。	-	-
RESET	输入	系统复位输入	-	-
EXLVI	输入	用于外部低电压检测的电压输入	输入	P120/INTP0
X1	输入	连接主系统时钟振荡器	输入	P121/OCD0A ^注
X2	-			P122/EXCLK/OCD0B ^注
EXCLK	输入	主系统时钟的外部时钟输入	输入	P122/X2/OCD0B ^注
XT1	输入	连接副系统时钟振荡器	输入	P123
XT2	-			P124/EXCLKS
EXCLKS	输入	副系统时钟的外部时钟输入	输入	P124/XT2
V_{DD}	-	正向电源	-	-
V_{SS}	-	地电位	-	-
FLMD0	-	Flash 存储器编程模式设置	-	-
OCD0A ^注	输入	用于设置片上调试模式	输入	P121/X1
OCD1A ^注				P31/INTP2
OCD0B ^注	-			P122/X2/EXCLK
OCD1B ^注				P32/INTP3

注 仅限于 $\mu\text{PD78F0376D}$ 和 $78F0386D$ 。

2.2 引脚功能描述

2.2.1 P00, P01 (端口 0)

P00 和 P01 作为 2 位 I/O 端口使用。这些引脚也可用于定时器 I/O。

以下操作模式可以用位选指定。

(1) 端口模式

P00 和 P01 作为 2 位 I/O 端口使用。通过使用端口模式寄存器 0 (PM0)，可按位设置 P00 和 P01 为输入或输出端口。由上拉电阻选择寄存器 0 (PU0) 规定内置上拉电阻的使用。

(2) 控制模式

P00 和 P01 作为定时器 I/O 的功能。

(a) TI000

将外部计数时钟输入到 16 位定时器/事件计数器 00，也可将捕捉触发信号输入到 16 位定时器/事件计数器 00 的捕捉寄存器 (CR000、CR010) 中。

(b) TI010

将捕捉触发信号输入到 16 位定时器/事件计数器 00 的捕捉寄存器 (CR000) 中。

(c) T000

定时器输出引脚。

2.2.2 P10 ~ P17 (端口 1)

P10 ~ P17 作为 8 位 I/O 端口使用。这些引脚也可用于外部中断请求输入、串行接口数据 I/O、时钟 I/O 以及定时器 I/O。

以下操作模式可以用位选指定。

(1) 端口模式

P10 ~ P17 作为 8 位 I/O 端口使用。通过使用端口模式寄存器 1 (PM1)，可按位设置 P10 ~ P17 为输入输出端口。由上拉电阻选择寄存器 1 (PU1) 定义内置上拉电阻的使用。

(2) 控制模式

P10 ~ P17 可用于外部中断请求输入、串行接口数据 I/O、时钟 I/O 和定时器 I/O。

(a) SI10

串行接口串行数据输入引脚。

(b) SO10

串行接口串行数据输出引脚。

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

串行接口串行时钟 I/O 引脚。

(d) RxD0, RxD6

异步串行接口的串行数据输入引脚。

(e) TxD0, TxD6

异步串行接口的串行数据输出引脚。

(f) TI50

用于输入一个外部计数时钟到 8 位定时器/事件计数器 50。

(g) TO50, TOH0, 和 TOH1

定时器输出引脚。

(h) INTP5

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

2.2.3 P20 ~ P27 (端口 2) (仅限于 μ PD78F037x)

P20 ~ P27 作为 8 位 I/O 端口使用，也可用于 A/D 转换器模拟输入。

以下操作模式可以用位选指定。

(1) 端口模式

P20 ~ P27 作为 8 位 I/O 端口使用。通过使用端口模式寄存器 2 (PM2)，可按位设置 P20 ~ P27 为输入/输出端口。

(2) 控制模式

P20 ~ P27 用于 A/D 转换器模拟输入引脚 (ANI0 ~ ANI7)，当使用这些引脚作为模拟输入引脚时，可参见 **12.6 A/D 转换器注意事项**中的**(5) ANI0/P20 ~ ANI7/P27**。

注意事项 复位后，ANI0/P20 ~ ANI7/P27 被设置为模拟输入模式。

2.2.4 P30 ~ P33 (端口 3)

P30 ~ P33 作为 4 位 I/O 端口使用，也可用于外部中断请求输入和定时器 I/O。

以下操作模式可以用位选指定。

(1) 端口模式

P30 ~ P33 作为 4 位 I/O 端口使用。通过使用端口模式寄存器 3 (PM3)，可按位设置 P30 ~ P33 为输入/输出端口。由上拉电阻选择寄存器 3 (PU3)定义内置上拉电阻的使用。

(2) 控制模式

P30 ~ P33 用于外部中断请求输入和定时器 I/O。

(a) INTP1 ~ INTP4

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

(b) TI51

将外部计数时钟输入到 8 位定时器/事件计数器 51 中。

(c) TO51

定时器输出引脚。

注意事项 在使用 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386, 和 78F0386D 时, 为了防止故障, 在复位后确保将 P31 引脚电平下拉。

备注 在使用片上调试功能时, μ PD78F0376D 和 78F0386D 的 P31 和 P32 能够用于片上调试模式设定引脚 (OCD1A, OCD1B)。详情请参看 第二十七章 片上调试功能 (仅限于 μ PD78F0376D 和 78F0386D)。

<R> 2.2.5 P60, P61 (端口 6)

P60 和 P61 作为 2 位 I/O 端口使用, 也可作为串行接口时钟 I/O 和数据 I/O 使用。
可以用位选设定以下工作模式。

(1) 端口模式

P60 和 P61 作为 2 位 I/O 端口使用。通过使用端口模式寄存器 6 (PM6), 可按位设置 P60 和 P61 为输入/输出端口。

P60 和 P61 输出为 N-ch 漏极开路输出 (6 V 耐压)。

(2) 控制模式

P60 和 P61 作为串行接口时钟 I/O 和数据 I/O 使用。

(a) SCL0

用于串行接口 IIC0 的串行时钟 I/O。
确保此引脚外部上拉。

(b) SDA0

用于串行接口 IIC0 的串行数据 I/O。
确保此引脚外部上拉。

注意事项 使用 78K0/LF2 时, 请确保与说明一致, 分别使用 P60/SCL0 和 P61/SDA0 作为串行时钟 I/O 引脚和串行数据 I/O 引脚。

2.2.6 P70 ~ P76 (端口 7)

P70 ~ P76 作为 7 位 I/O 端口使用。这些引脚也可作为按键中断输入引脚。
可以用位选设定以下工作模式。

(1) 端口模式

P70 ~ P76 作为 7 位 I/O 端口使用。通过使用端口模式寄存器 7 (PM7)，可按位设置 P70 ~ P76 为输入/输出端口。由上拉电阻选择寄存器 7 (PU7)定义内置上拉电阻的使用。

(2) 控制模式

P70 ~ P76 作为按键中断输入引脚。

2.2.7 P120 ~ P124 (端口 12)

P120 ~ P124 作为 5 位 I/O 端口使用。也可以作为外部中断请求输入、外部低电压检测的电压输入、连接主系统时钟振荡器、连接副时钟振荡器和外部时钟输入。以下操作模式可以位选设定。

(1) 端口模式

P120 ~ P124 作为 5 位 I/O 端口使用，可由端口模式寄存器 12(PM12)，将 P120 ~ P124 设置为输入/输出端口。仅对于 P120，可由上拉电阻选择寄存器 12(PU12)定义内置上拉电阻的使用。

(2) 控制模式

P120 ~ P124 可作为外部中断请求输入、外部低电压检测的电压输入、连接主系统时钟的振荡器、连接副时钟的振荡器和外部时钟输入。

(a) INTPO

通过定义有效沿（上升沿、下降沿，或兼有上升沿和下降沿），它可作为外部中断请求输入(INTPO)使用。

(b) EXLVI

用于外部低电压检测的电压输入。

(c) X1, X2

用于连接主系统时钟振荡器。

(d) EXCLK

用于主系统时钟的外部时钟输入。

(e) XT1, XT2

用于连接副时钟振荡器。

(f) EXCLKS

用于副时钟的外部时钟输入。

备注 在使用片上调试功能时， μ PD78F0376D 和 78F0386D 的 X1 和 X2 能够用于片上调试模式设定引脚 (OCD0A, OCD0B)。详情请参看 第二十七章 片上调试功能 (仅限于 μ PD78F0376D 和 78F0386D)。

2.2.8 AV_{REF} (仅限于 μ PD78F037x)

用于 A/D 转换器参考电压输入引脚。

不使用 A/D 转换器时，将该引脚直接连到 V_{DD}^注。

注 当端口 2 作为数字端口使用时，应使 AV_{REF} 引脚电平与 V_{DD} 引脚的电平相同。

2.2.9 AV_{SS} (仅限于 μ PD78F037x)

作为 A/D 转换器的地引脚。即使在不使用 A/D 转换器时，该引脚电平也始终应与 V_{SS} 相同。

2.2.10 S0 ~ S25

这些引脚作为 LCD 控制器/驱动器的 segment 信号输出引脚。

2.2.11 S26 ~ S35 (仅限于 μ PD78F038x)

这些引脚作为 LCD 控制器/驱动器的 segment 信号输出引脚。

2.2.12 COM0 ~ COM3

这些引脚作为 LCD 控制器/驱动器的 common 信号输出引脚。

2.2.13 LV_{DD}

此引脚作为 LCD 控制器/驱动器的正向电源引脚。

2.2.14 LV_{SS}

此引脚作为 LCD 控制器/驱动器的地电势引脚。

2.2.15 VL_{C0} ~ VL_{C2}

这些引脚作为驱动 LCD 的电源电压引脚。

2.2.16 CAPH, CAPL

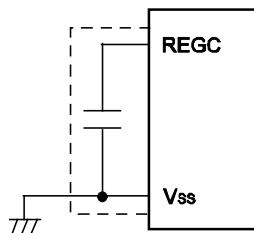
这些引脚作为驱动 LCD 的电容连接引脚。

2.2.17 RESET

低有效的系统复位输入引脚。

2.2.18 REGC

<R> 用于内部操作的调节器输出(2.5 V)稳定电容的连接。通过一个电容 (0.47 ~ 1 μ F : 推荐) 将此引脚连接到 V_{SS} 。



注意事项 在上图虚线部分，用尽可能短的配线连接。

2.2.19 V_{DD}

V_{DD} 为正向供电电源引脚。

2.2.20 V_{SS}

V_{SS} 为地引脚。

2.2.21 FLMD0

用于 Flash 编程模式设置。

在正常操作模式下将 FLMD0 连接至 V_{SS}。

在 Flash 编程模式下，将该引脚连接至 Flash 编程器。

2.3 引脚 I/O 电路和未使用引脚的建议连接方式

表 2-2 为引脚 I/O 电路类型和未使用引脚的建议连接方式。

参考 图 2-1 显示每一种类型的 I/O 电路结构。

表 2-2. 引脚 I/O 电路类型 (1/2)

引脚名称	I/O 电路类型	I/O	未使用引脚的建议连接方式	
P00/TI000	5-AH	I/O	输入： 通过电阻单独连接至 V_{DD} 或 V_{SS} 。 输出： 保持开路。	
P01/TI010/TO00				
P10/SCK10/TxD0				
P11/SI10/RxD0				
P12/SO10	5-AG			
P13/TxD6	5-AH			
P14/RxD6				
P15/TOH0	5-AG			
P16/TOH1/INTP5	5-AH			
P17/TI50/TO50				
P20/ANI0 ~ P27/ANI7 ^{注1,2}	11-G			<模拟设置> 连接至 AV_{REF} 或 AV_{SS} 。 <数字设置> 输入： 通过电阻单独连接至 V_{DD} 或 V_{SS} 。 输出： 保持开路。
P30/INTP1	5-AH			输入： 通过电阻单独连接至 V_{DD} 或 V_{SS} 。 输出： 保持开路。
P31/INTP2				
P32/INTP3				
P33/TI51/TO51/INTP4				
P60/SCL0	13-AD	输入： 连接至 V_{SS} 。 输出： 保持开路。		
P61/SDA0				
P70/KR0 ~ P76/KR6	5-AH	输入： 通过电阻单独连接至 V_{DD} 或 V_{SS} 。 输出： 保持开路。		
P120/INTP0/EXLVI				

- 注
1. 仅限于 μ PD78F037x。
 2. P20/ANI0 ~ P27/ANI7 在复位释放后处于模拟输入模式。

表 2-2. 引脚 I/O 电路类型 (2/2)

引脚名称	I/O 电路类型	I/O	未使用引脚的建议连接方式
P121/X1 ^{注1}	37	I/O	输入： 通过电阻单独连接至 V _{DD} 或 V _{SS} 。 输出： 保持开路。
P122/X2/EXCLK ^{注1}			
P123/XT1 ^{注1}			
P124/XT2/EXCLKS ^{注1}			
S0 ~ S25	17	输出	保持开路
S26 ~ S35 ^{注2}			
COM0 ~ COM3	18		
V _{LC0} ~ V _{LC2}	-	-	
CAPH, CAPL			
RESET	2	输入	-
FLMD0	38	输入	连接到 V _{SS} .
AV _{REF} ^{注3}	-	-	直接连接到 V _{DD} .
AV _{SS} ^{注3}			直接连接到 V _{SS} .

- 注
1. 当这些引脚并不使用时，在 I/O 端口模式下使用以上推荐的连接（参看 图 5-2 时钟操作模式选择寄存器 (OSCCTL) 的格式）。
 2. 仅限于 μ PD78F038x。
 3. 仅限于 μ PD78F037x。

图 2-1. 引脚 I/O 电路列表 (1/2)

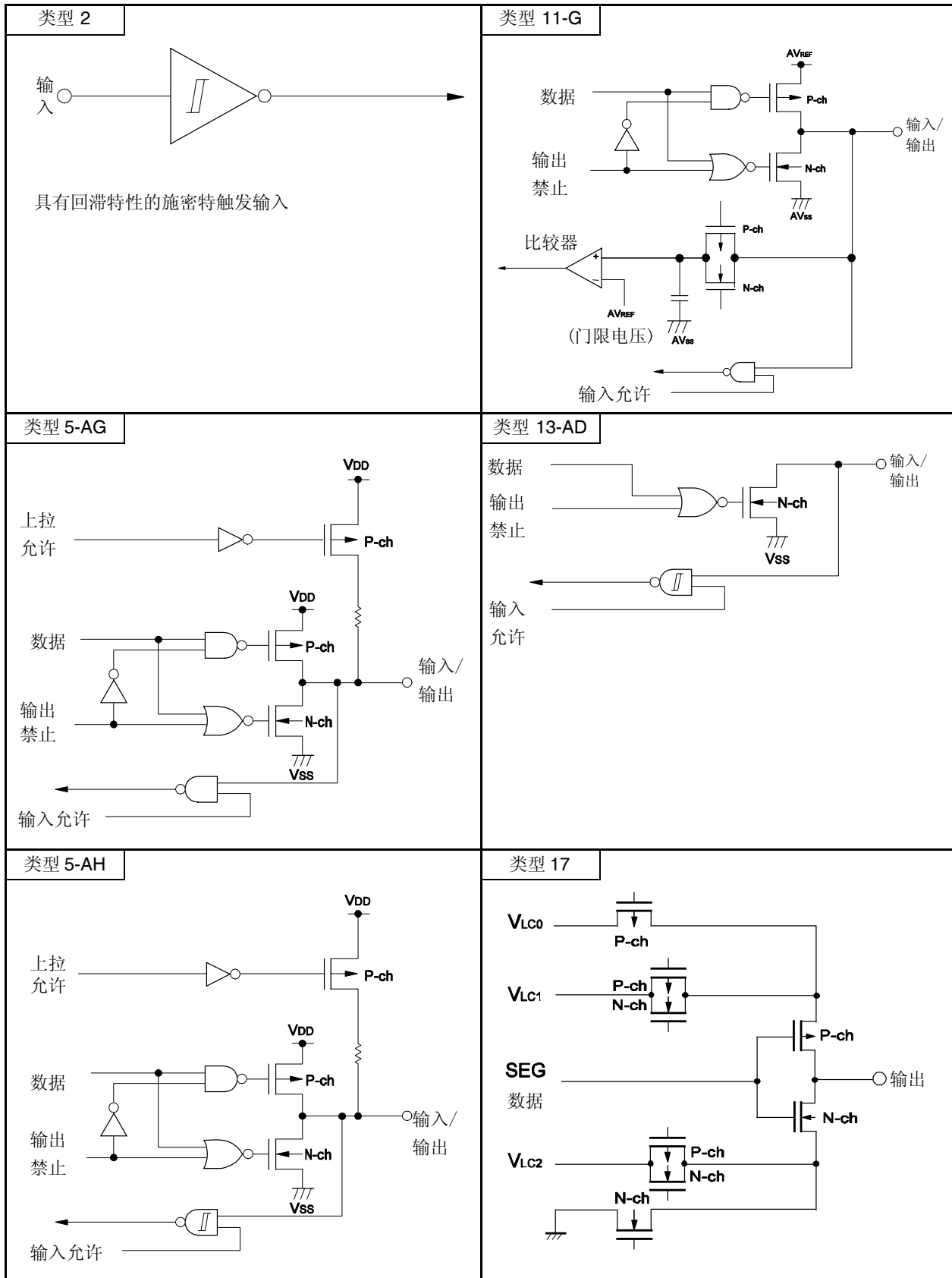
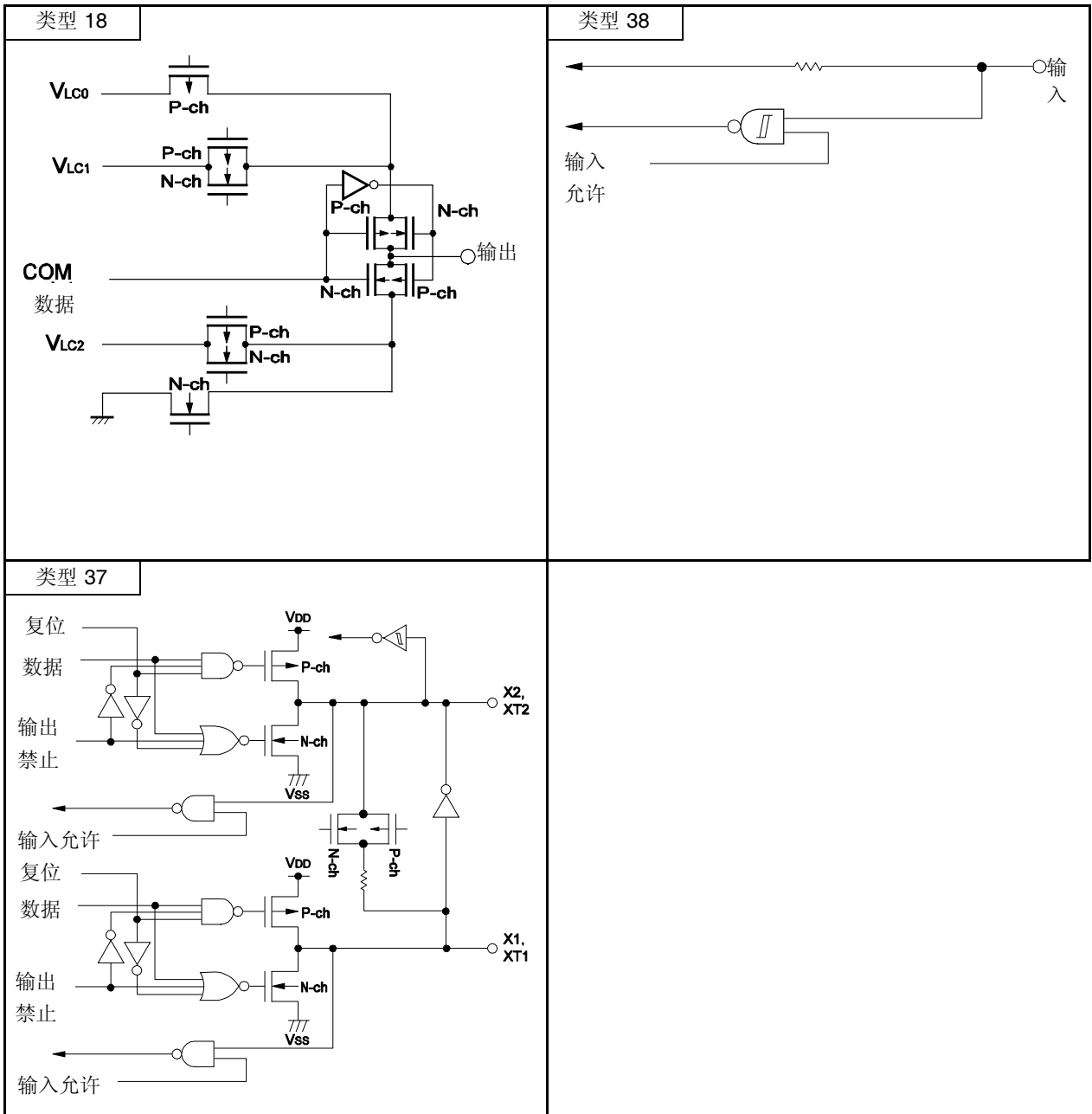


图 2-1. 引脚 I/O 电路列表 (2/2)



第三章 CPU 结构

3.1 存储器空间

78K0/LF2 产品可以访问大小为 64KB 的存储器空间。图 3-1 ~ 3-6 显示了存储器空间映射图。

- 注意事项
1. 不管内部存储器容量有多少，所有 78K0/LF2 产品的内部存储器容量切换寄存器 (IMS) 和内部扩展 RAM 容量切换寄存器 (IXS) 的初始值都是固定的 (IMS = CFH, IXS = 0CH)。因此每种产品设置值如下所示。
 2. 要设置存储器容量，应先设置 IMS，然后 IXS。设置存储器容量以便内部 ROM 和内部扩展 RAM 区域不会重叠。

表 3-1. 设置内部存储器容量切换寄存器 (IMS) 和内部扩展 RAM 容量切换寄存器 (IXS) 的值

<R>

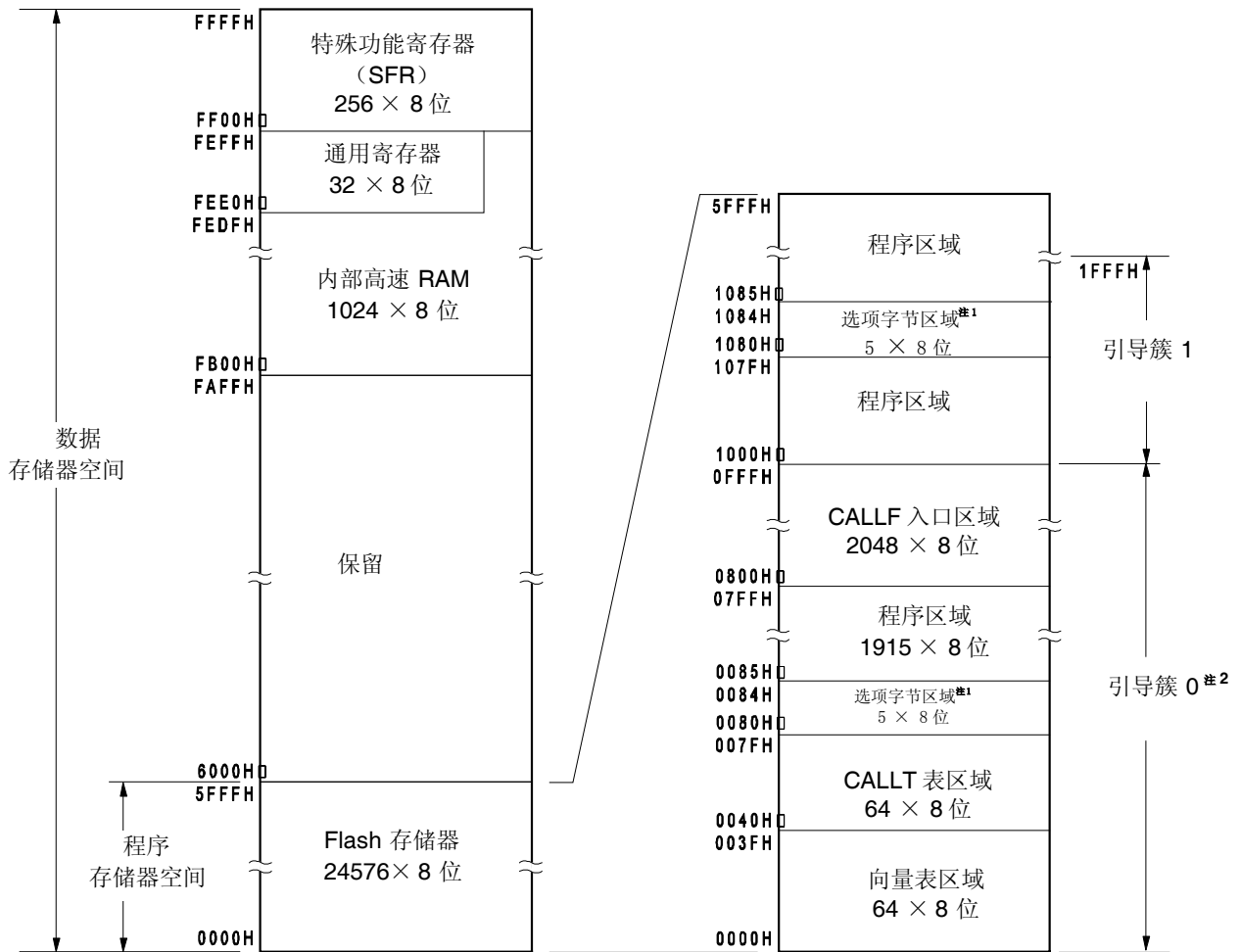
Flash 存储器版本 (78K0/LF2)	IMS	IXS	ROM 容量	内部高速 RAM 容量	内部扩展 RAM 容量
μPD78F0372, 78F0382	C6H	0CH	24 KB	1 KB	-
PD78F0373, 78F0383	C8H		32 KB		
PD78F0374, 78F0384	CCH	0AH	48 KB		1 KB
PD78F0375, 78F0385	CFH	08H	60 KB		2 KB
PD78F0376, 78F0386, 78F0376D, 78F0386D ^{注1}	CCH ^{注2}	04H	96 KB ^{注2}		4 KB

注 1. 具有片上调试功能的产品其 ROM 和 RAM 的容量可根据调试目标产品进行调整，根据调试目标产品设置 IMS 和 IXS 值。

<R>

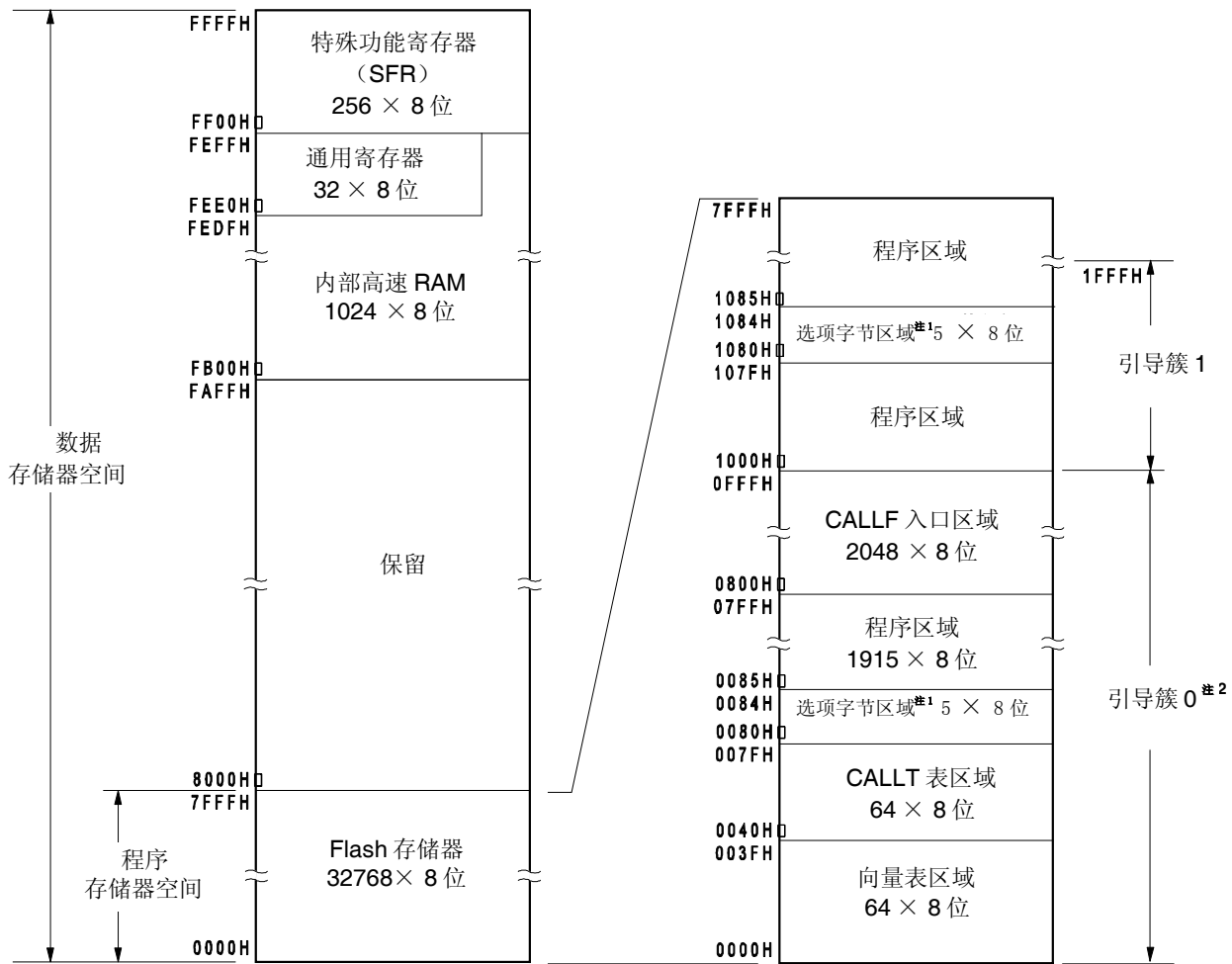
2. μPD78F0376, 78F0376D, 78F0386 和 78F0386D 分别拥有 96KB 的内部 ROM。因为使用存储器 bank，这些器件的 IMS 的设定值与 48KB 产品是相同的。如何设置存储器 bank，请参看 3.6 存储器 Bank 选择寄存器(BANK) (仅限于 μPD78F0376, 78F0376D, 78F0386 和 78F0386D)。

图 3-1. 存储器空间映射图 (μ PD78F0372, 78F0382)



- 注
1. 当不使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H。
当使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H 和 1080H ~ 1084H。
 2. 根据安全设置可以禁止写入引导簇 0 (参见 26.8 安全设置)。

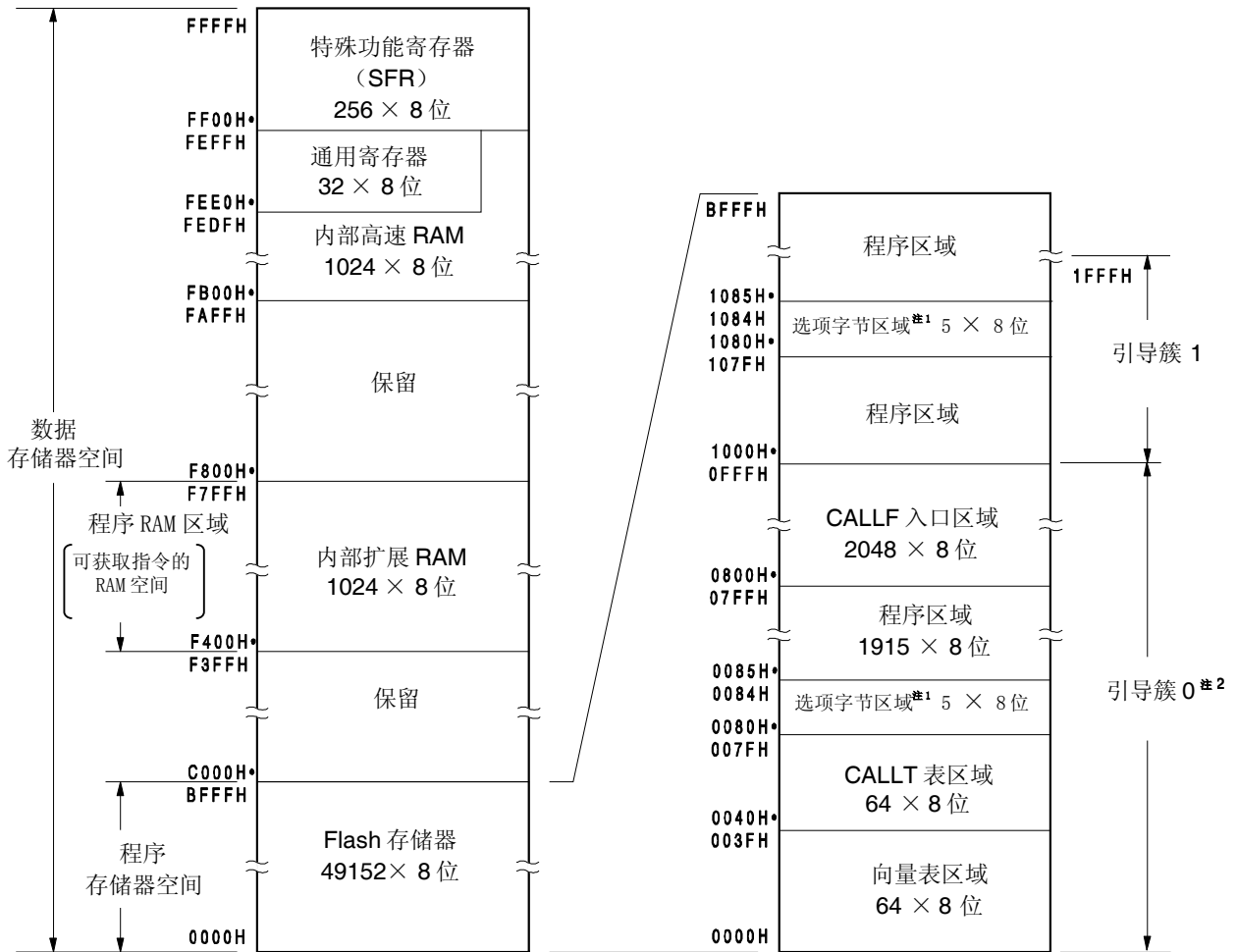
图 3-2. 存储器空间映射图 (μ PD78F0373, 78F0383)



- 注
1. 当不使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H。
当使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H 和 1080H ~ 1084H。
 2. 根据安全设置可以禁止写入引导簇 0 (参见 26.8 安全设置)。

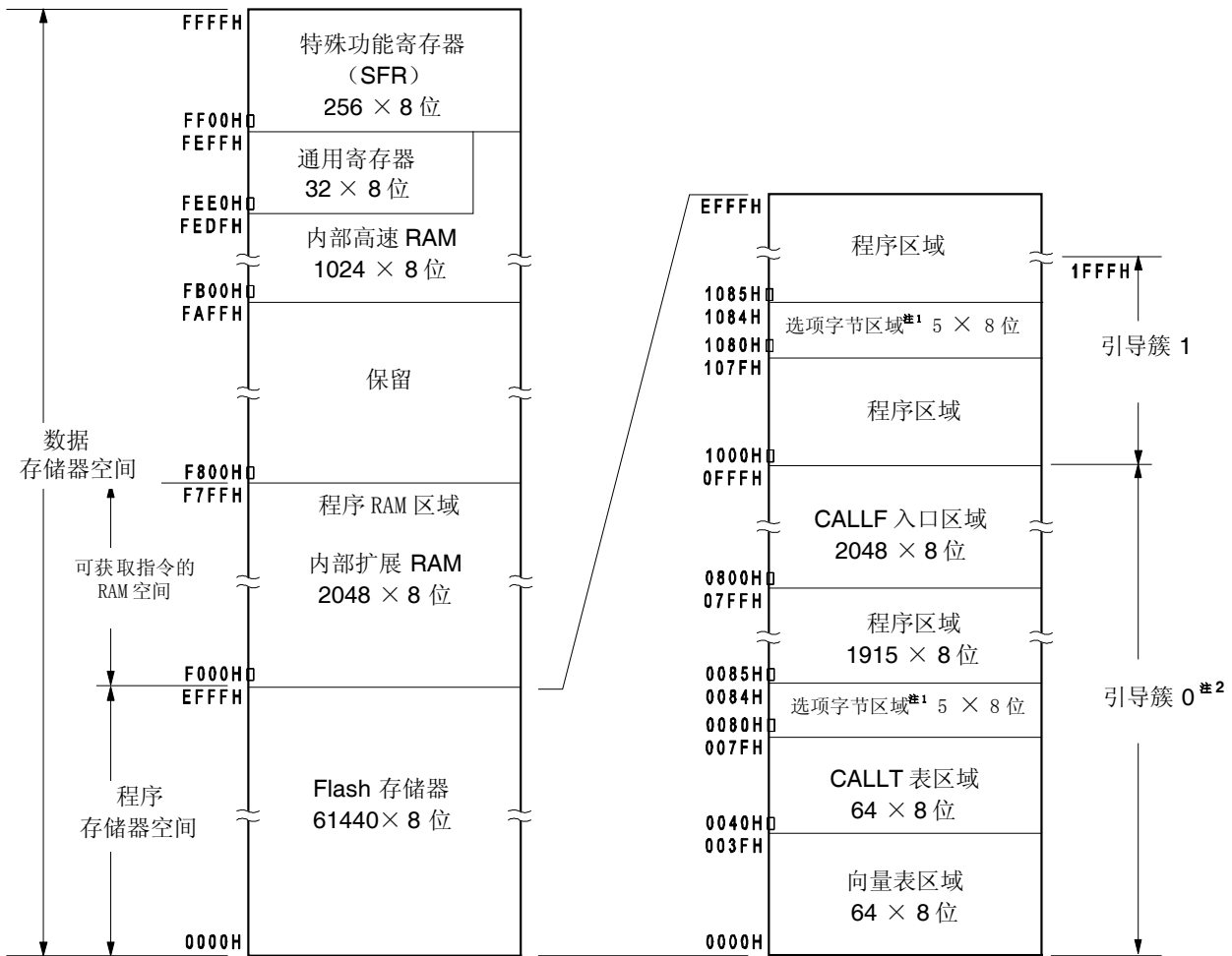
<R>

图 3-3. 存储器空间映射图 (μ PD78F0374, 78F0384)



- 注
1. 当不使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H。
当使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H 和 1080H ~ 1084H。
 2. 根据安全设置可以禁止写入引导簇 0 (参见 26.8 安全设置)。

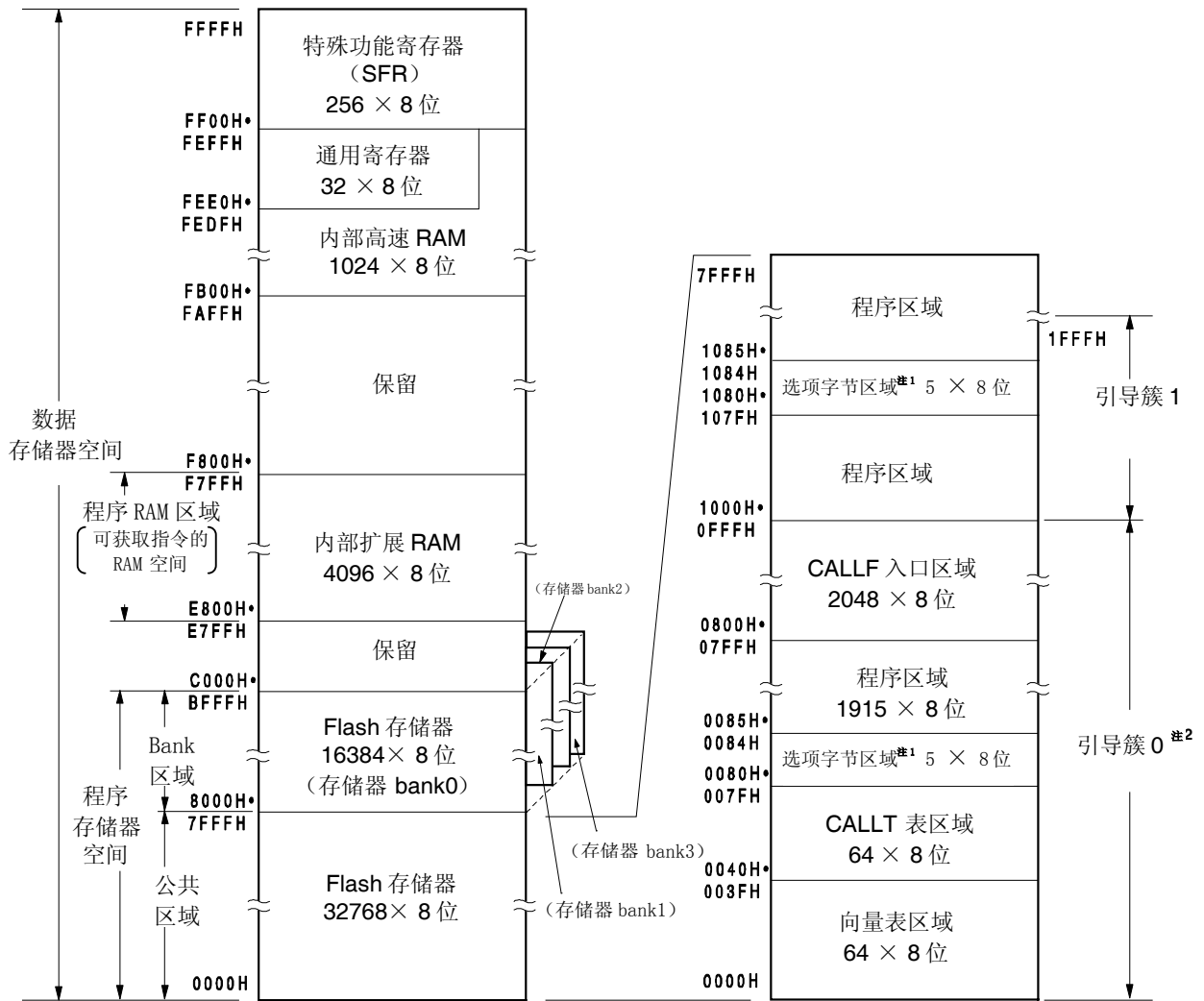
图 3-4. 存储器空间映射图 (μ PD78F0375, 78F0385)



- 注
1. 当不使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H。
当使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H 和 1080H ~ 1084H。
 2. 根据安全设置可以禁止写入引导簇 0 (参见 26.8 安全设置)。

<R>

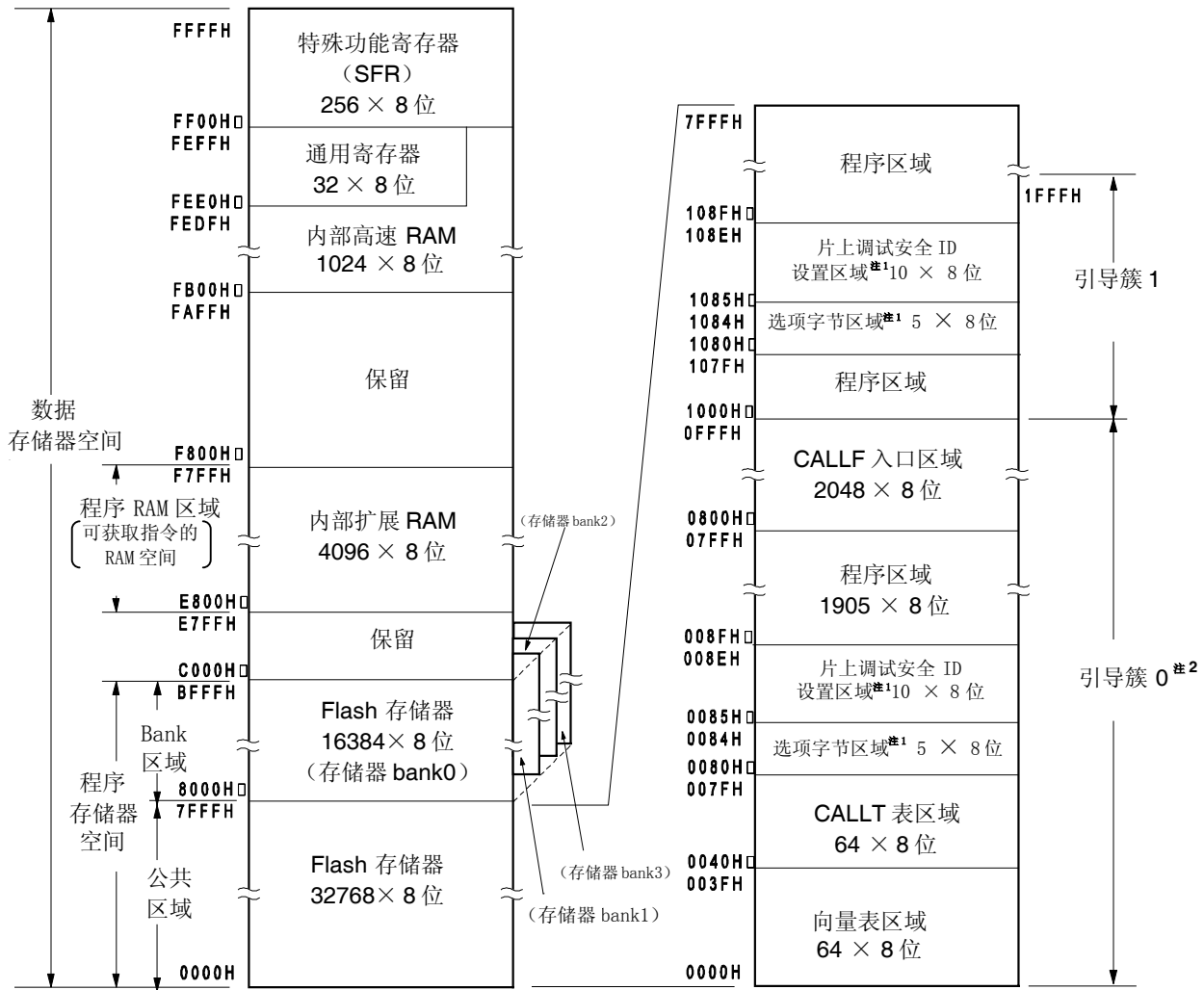
图 3-5. 存储器空间映射图 (μPD78F0376, 78F0386)



- 注
1. 当不使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H。
当使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H 和 1080H ~ 1084H。
 2. 根据安全设置可以禁止写入引导簇 0 (参见 26.8 安全设置)。

<R>

图 3-6. 存储器空间映射图 (μPD78F0376D, 78F0386D)



- 注
1. 当不使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H，和片上调试安全 ID 区域：0085H ~ 008EH。
当使用引导交换功能的时候，设置选项字节区域：0080H ~ 0084H 和 1080H ~ 1084H，和片上调试安全 ID 区域：0085H ~ 008EH 和 1085H ~ 108EH。
 2. 根据安全设置可以禁止写入引导簇 0 (参见 26.8 安全设置)。

3.1.1 内部程序存储器空间

内部程序存储器空间用于存储程序和表数据，一般通过程序计数器 (PC) 来寻址。

78K0/LF2 产品内部 ROM (Flash 存储器) 的情况如下表所示。

<R>

表 3-2. 内部 ROM 容量

产品型号	内部 ROM	
	结构	容量
μ PD78F0372, 78F0382	Flash 存储器	24576 \times 8 位 (0000H \sim 5FFFH)
μ PD78F0373, 78F0383		32768 \times 8 位 (0000H \sim 7FFFH)
μ PD78F0374, 78F0384		49152 \times 8 位 (0000H \sim BFFFH)
μ PD78F0375, 78F0385		61440 \times 8 位 (0000H \sim EFFFH)
μ PD78F0376, 78F0386, 78F0376D, 78F0386D		98304 \times 8 位 (0000H \sim 7FFFH (通用区域) + 8000H \sim BFFFH (bank 区域) \times 4)

内部程序存储器空间主要分为以下几个区域。

(1) 向量表区域

从 0000H 到 003FH 总共 64 字节作为向量表区域。在向量表中存放的是根据复位信号输入或每个中断请求的产生进行转移的程序的起始地址。

在 16 位地址中，低 8 位是偶地址，高 8 位是奇地址。

表 3-3. 向量表

向量表地址	中断源	向量表地址	中断源
0000H	RESET 输入, POC, LVI, WDT	001EH	INTTM50
0004H	INTLVI	0020H	INTTM000
0006H	INTP0	0022H	INTTM010
0008H	INTP1	0024H ^{注1}	INTAD ^{注1}
000AH	INTP2	0026H	INTSR0
000CH	INTP3	0028H	INTWTI
000EH	INTP4	002AH	INTTM51
0010H	INTP5	002CH	INTKR
0012H	INTSRE6	002EH	INTWT
0014H	INTSR6	0034H	INTIIC0/INTDMU ^{注2}
0016H	INTST6	0038H ^{注2}	INTTM001 ^{注2}
0018H	INTCSI10/INTST0	003AH ^{注2}	INTTM011 ^{注2}
001AH	INTTMH1	003EH	BRK
001CH	INTTMH0		

注 1. 仅限于 μ PD78F037x。

注 2. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D。

(2) CALLT 指令表区域

0040H 至 007FH 共 64 字节的区域，可存放 1 字节调用指令(CALLT)的子程序入口地址。

(3) 选项字节区域

0080H ~ 0084H 和 1080H ~ 1084H 各 5 字节的区域可以用作选项字节区域。当不使用引导交换功能时，在 0080H ~ 0084H 设置选项字节；而当使用引导交换功能时，在 0080H ~ 0084H 和 1080H ~ 1084H 设置选项字节。详细信息参见 **第二十五章 选项字节**。

(4) CALLF 指令入口区域

0800H ~ 0FFFH 的区域，用于子程序的直接调用，通过一个 2 字节的调用指令(CALLF)实现。

(5) 片上调试安全 ID 设置区域 (仅限于 μ PD78F0376D, 78F0386D)

0085H ~ 008EH 和 1085H ~ 108EH 各 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 0085H ~ 008EH 区域设置片上调试安全 ID；而当使用引导交换功能时，在 0085H ~ 008EH 和 1085H ~ 108EH 区域设置。详细情况请参见 **第二十七章 片上调试功能 (仅限于 μ PD78F0376D 和 78F0386D)**。

3.1.2 内部数据存储器空间

78K0/LF2 产品包括以下几种 RAM。

(1) 内部高速 RAM

<R>

表 3-4. 内部高速 RAM 容量

产品型号	内部高速 RAM
μ PD78F0372, 78F0382	1024 × 8 位 (FB00H ~ FEFFH)
μ PD78F0373, 78F0383	
μ PD78F0374, 78F0384	
μ PD78F0375, 78F0385	
μ PD78F0376, 78F0386 78F0376D, 78F0386D	

FEE0H ~ FEFFH 共 32 字节的区域分配给 4 个通用寄存器 bank，每个 bank 都由 8 个 8 位寄存器组成。

这个区域不可以用作写入和执行指令的程序区域。

内部高速 RAM 还可以作为堆栈存储器使用。

(2) 内部扩展 RAM

<R>

表 3-5. 内部扩展 RAM 容量

产品型号	内部扩展 RAM
μ PD78F0372, 78F0382	-
μ PD78F0373, 78F0383	
μ PD78F0374, 78F0384	1024 × 8 位 (F400H ~ F7FFH)
μ PD78F0375, 78F0385	2048 × 8 位 (F000H ~ F7FFH)
μ PD78F0376, 78F0386, 78F0376D, 78F0386D	4096 × 8 位 (E800H ~ F7FFH)

内部扩展 RAM 可以像内部高速 RAM 一样作为普通数据区域使用，也可作为写入和执行指令的程序区域。内部扩展 RAM 不能用于堆栈存储器。

(3) LCD 显示 RAM

LCD 显示 RAM 是与 LCD 控制器/驱动器整合在一起的 (参看 图 17-4 LCD 显示 RAM)。

表 3-6. LCD 显示 RAM 容量

产品型号	LCD 显示 RAM
PD78F037x	26 × 4 位 (LCDSEG 的 00H ~ 19H)
PD78F038x	36 × 4 位 (LCDSEG 的 00H ~ 23H)

3.1.3 特殊功能寄存器 (SFR) 区域

片内外围硬件的特殊功能寄存器 (SFR) 被分配在 FF00H ~ FFFFH 的区域 (参见 3.2.3 特殊功能寄存器 (SFR) 中表 3-7 特殊功能寄存器列表)。

注意事项 不要访问那些未分配特殊功能寄存器的地址区域。

3.1.4 数据存储器寻址

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

基于可操作性和其他考虑，在 78K0/LF2 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，故可使用一些特殊的寻址方式，具有特殊功能寄存器(SFR)和通用寄存器的功能。图 3-7 ~ 3-11 显示了数据存储区域与寻址方式的对应关系。如需了解每种寻址方式的详细内容，参见 3.4 操作数地址寻址。

图 3-7. 数据存储器和寻址方式的对应关系 (μ PD78F0372, 78F0382)

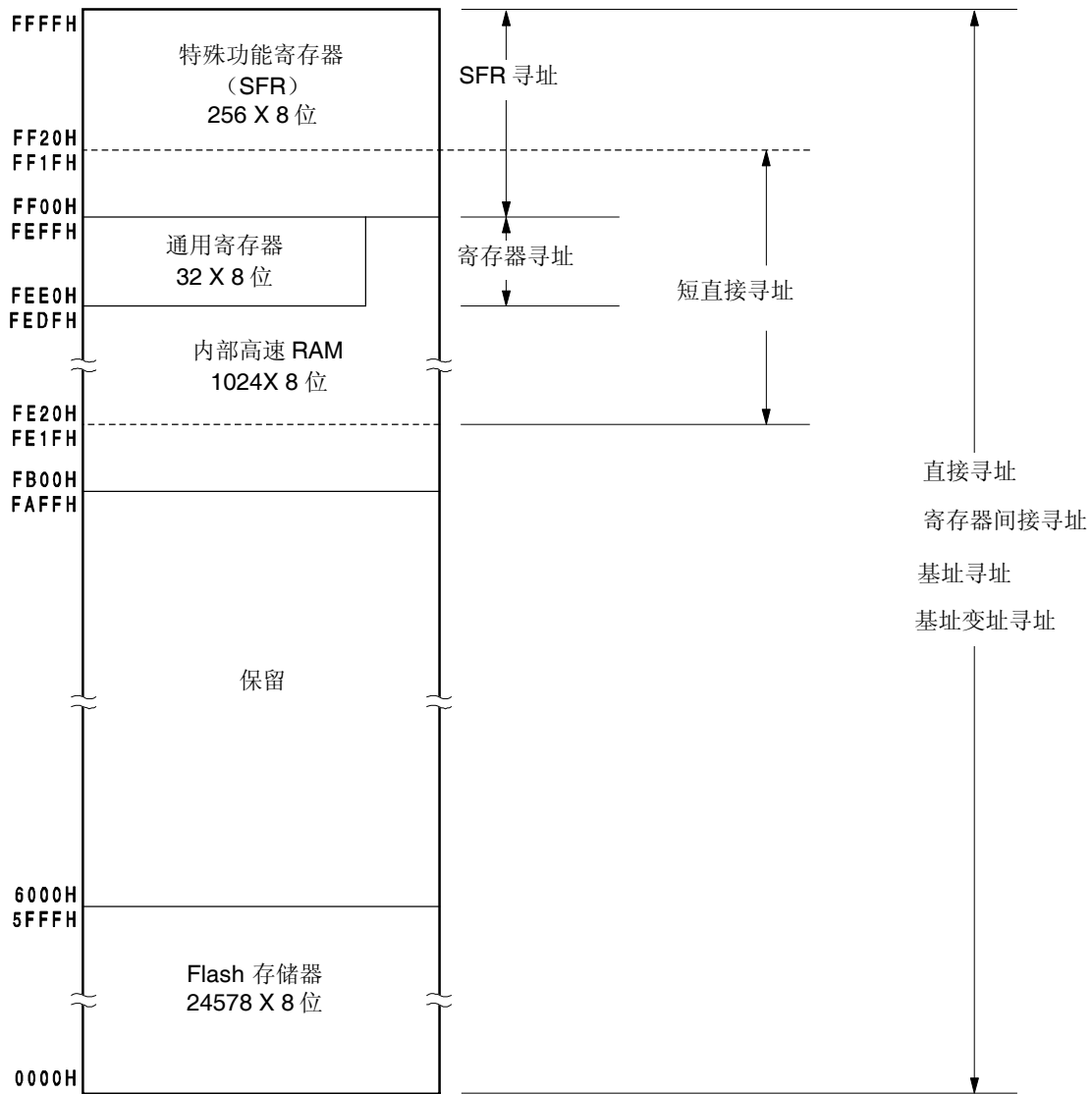
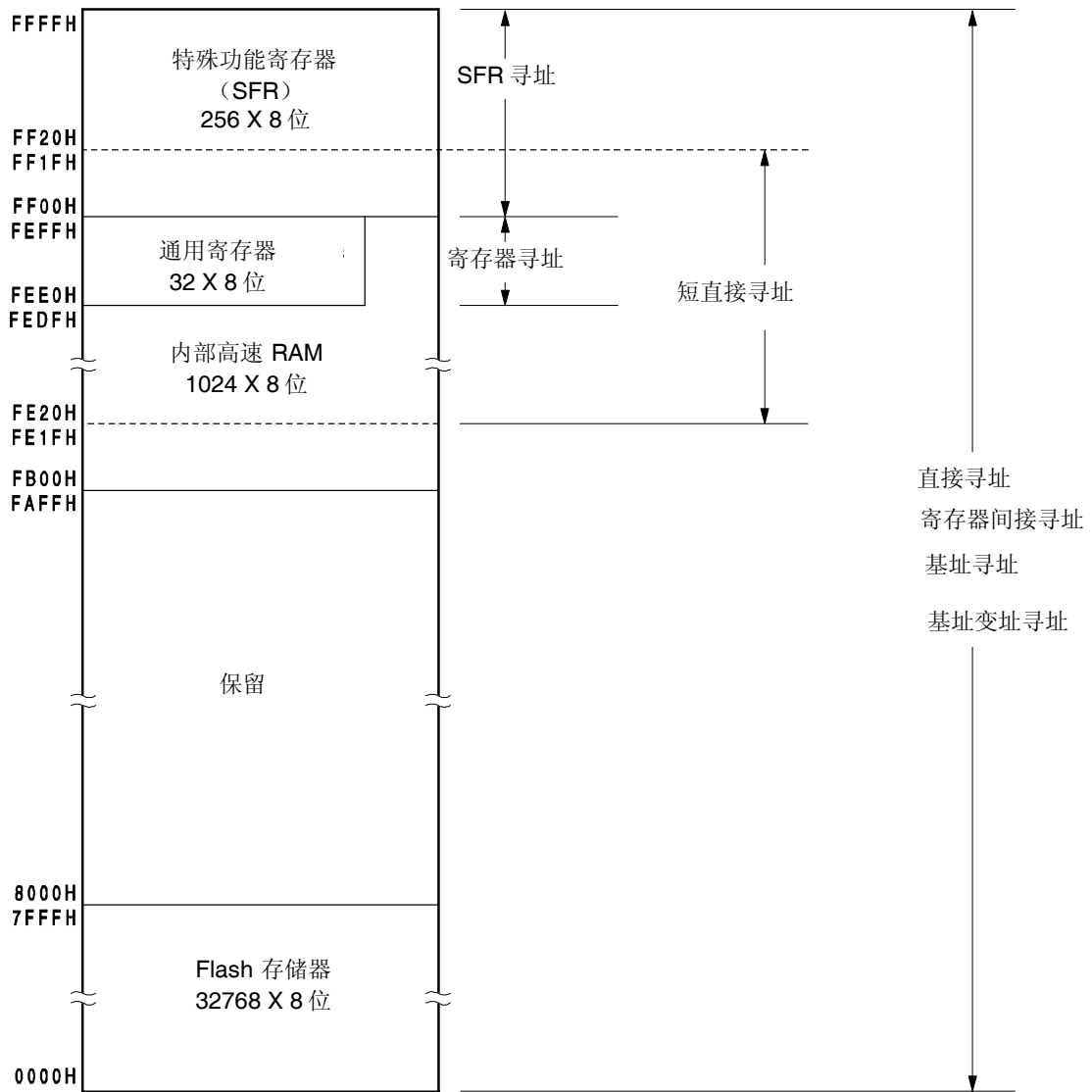


图 3-8. 数据存储器与寻址方式的对应关系 (μ PD78F0373, 78F0383)



<R>

图 3-9. 数据存储器和寻址方式的对应关系 (μ PD78F0374, 78F0384)

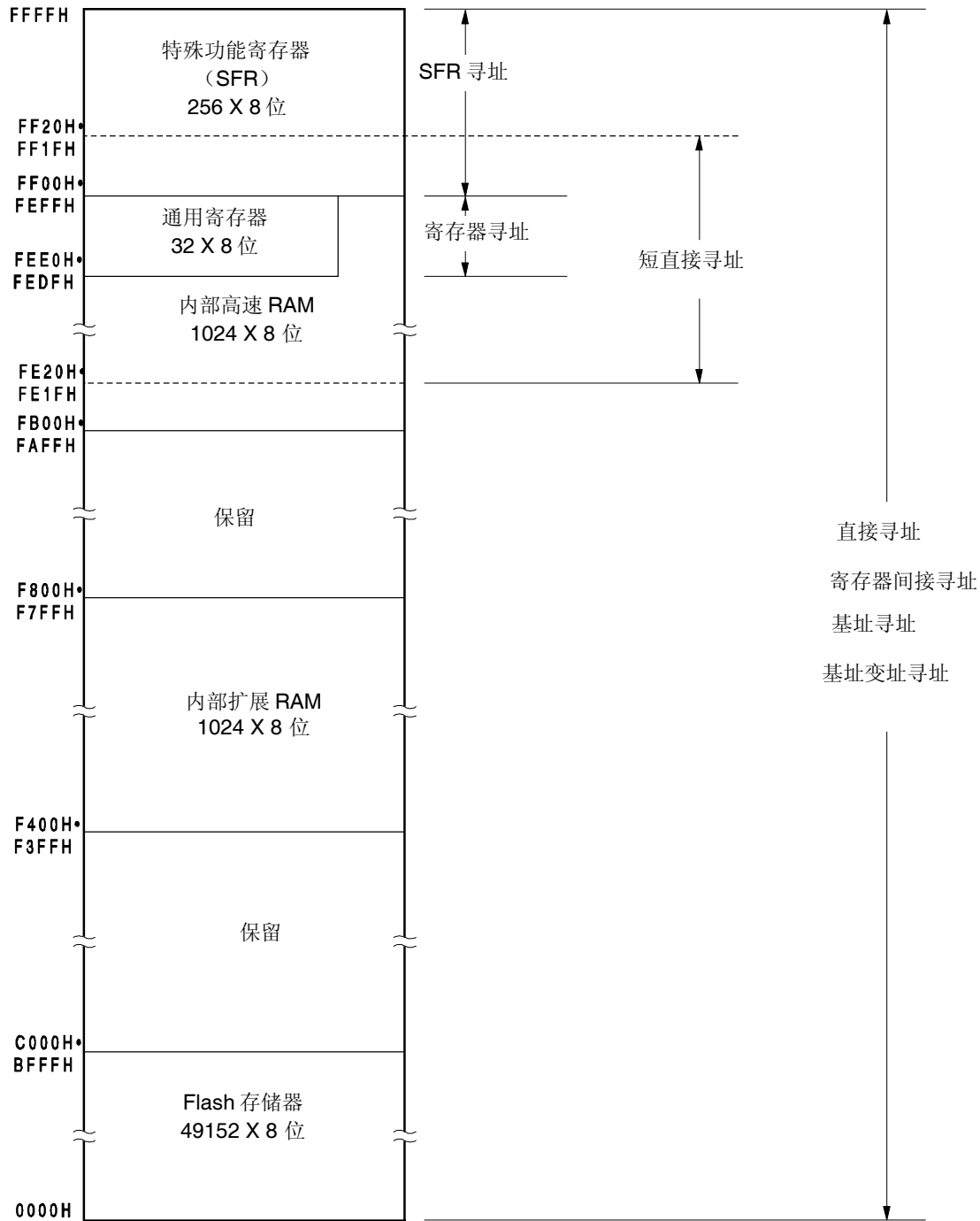
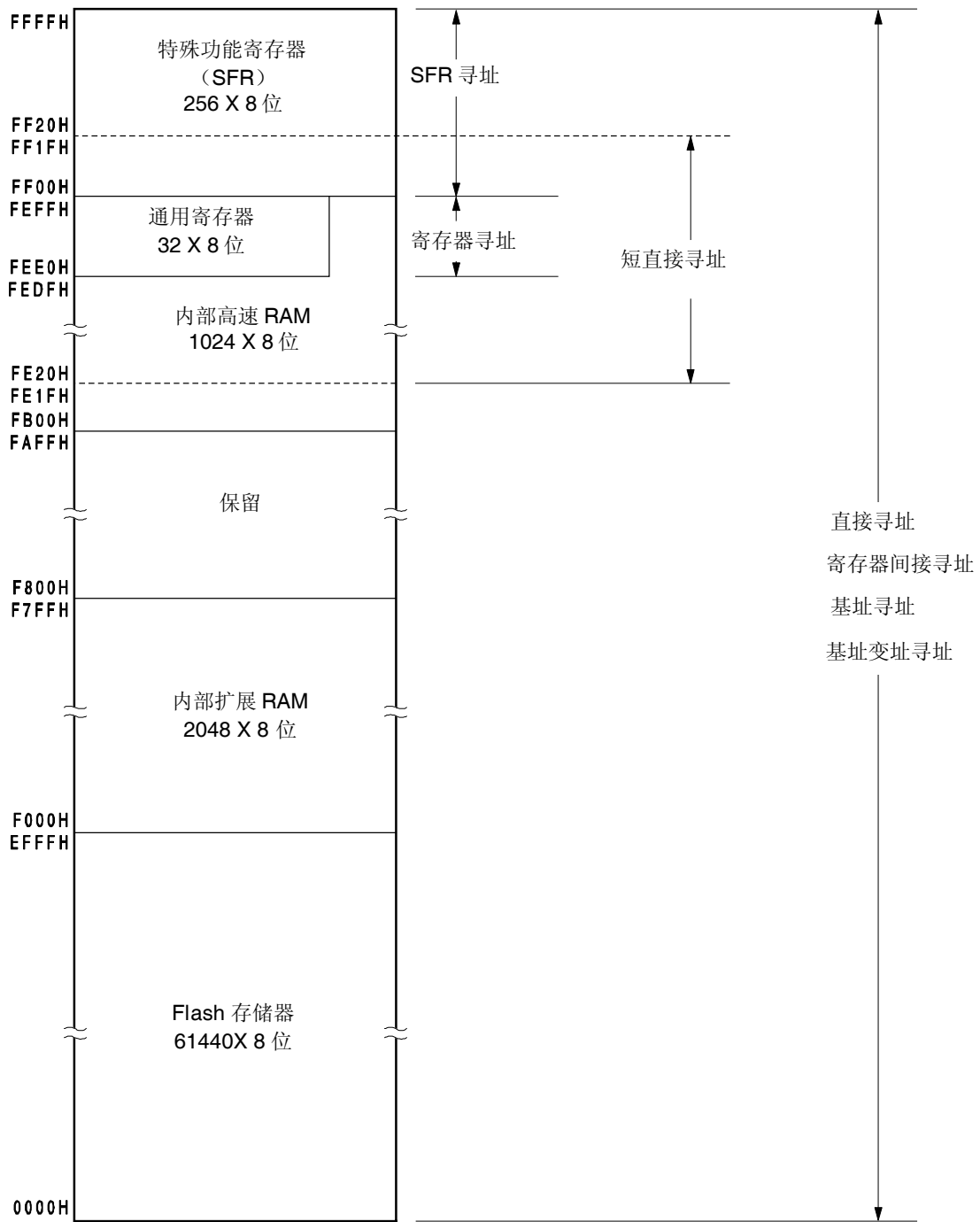
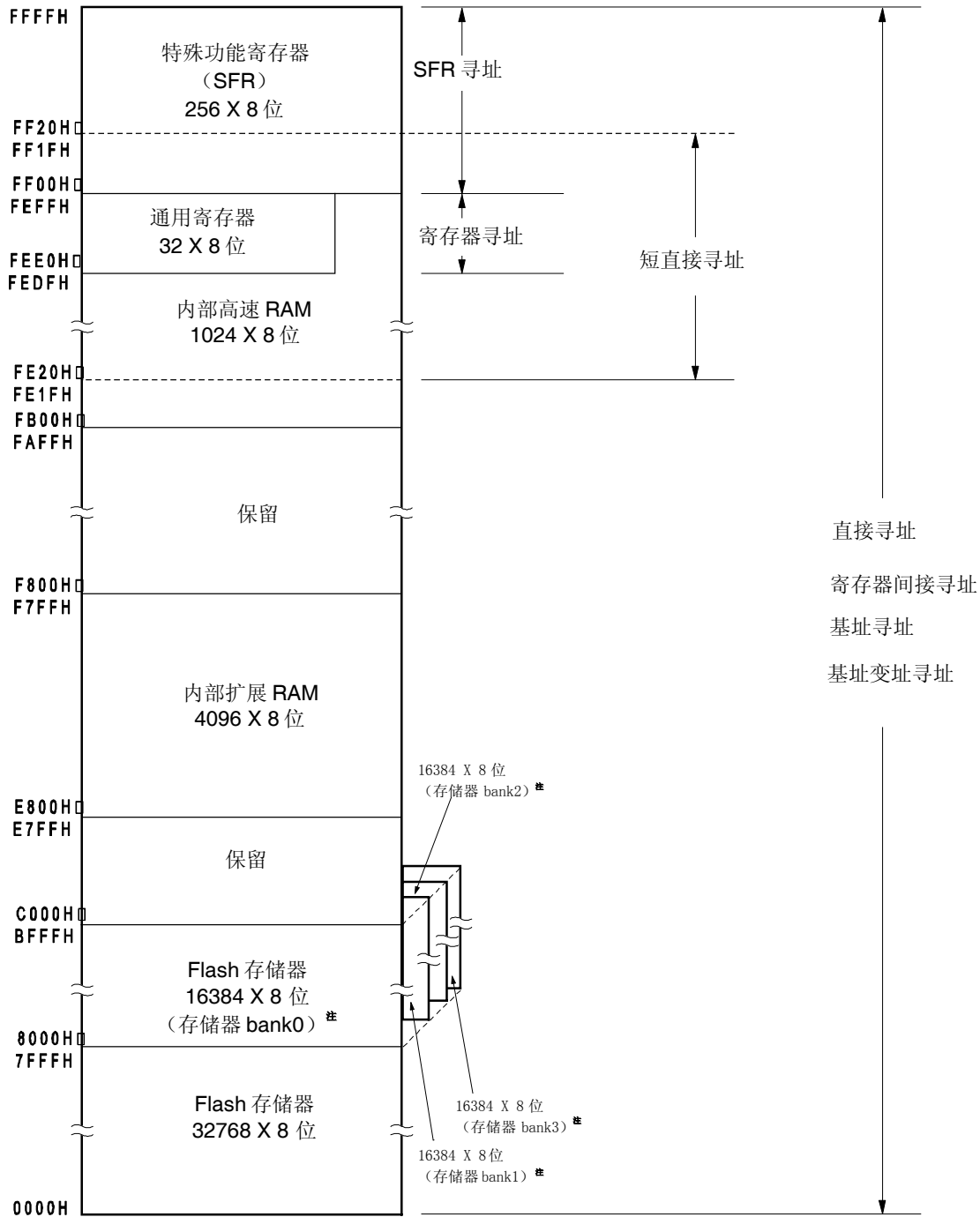


图 3-10. 数据存储器和寻址方式的对应关系 (μ PD78F0375, 78F0385)



<R>

图 3-11. 数据存储器与寻址方式的对应关系
(μ PD78F0376, 78F0376D, 78F0386, 78F0386D)



注 为了跳转到或寻址到某一没有通过存储器 bank 选择寄存器 (BANK) 设置的存储器 bank, 可以通过利用 BANK 来改变这个存储器 bank 的设置。

3.2 处理器寄存器

78K0/LF2 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。程序计数器(PC)、程序状态字(PSW)和堆栈指针(SP)都属于控制寄存器。

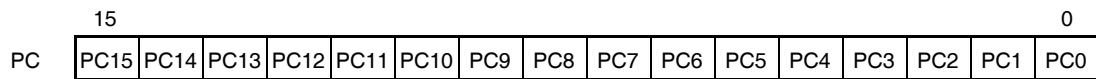
(1) 程序计数器 (PC)

程序计数器是一个 16 位寄存器，用于存放下一条即将要执行的指令的地址。

在正常情况下，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 3-12. 程序计数器格式



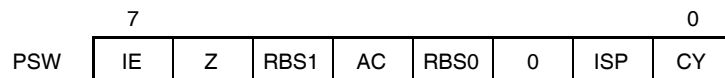
(2) 程序状态字 (PSW)

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

根据中断请求的产生或 PUSH PSW 指令执行，程序状态字的内容自动入栈；通过执行 RETB, RETI 和 POP PSW 指令，程序状态字的值自动恢复。

复位信号的产生将程序状态字的内容设置为 02H。

图 3-13. 程序状态字格式



(a) 中断允许标志 (IE)

该标志用于控制 CPU 响应中断请求操作。

当 IE 为 0 时，表示不允许中断(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)，通过优先服务标志(ISP)、用于各种中断源的中断屏蔽标志以及优先级规定标志来完成响应中断请求的控制。

当执行 DI 指令或中断请求得到响应时，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)

当操作结果为 0 时，该标志置 1，其他情况置 0。

(c) 寄存器组选择标志 (RBS0 和 RBS1)

寄存器组选择标志有两位，用于选择四组寄存器中的一组。

标志位中存储的这 2 位信息用来指明执行 SEL RBn 指令时所选择的寄存器组。

(d) 半进位标志 (AC)

如果操作结果中第 3 位有进位或是在第 3 位上有借位，则该标志置 1。其他情况该标志置 0。

(e) 优先服务标志 (ISP)

该标志用来管理可屏蔽向量中断响应的优先级。当 ISP 为 0 时，由优先级指定标志寄存器 (PR0L, PR0H, PR1L, PR1H) (参见 19.3 (3) 优先级指定标志寄存器 (PR0L, PR0H, PR1L, PR1H)) 指定为低优先级的向量中断请求被禁止响应。对请求的实际响应是由中断允许标志 (IE) 的状态控制的。

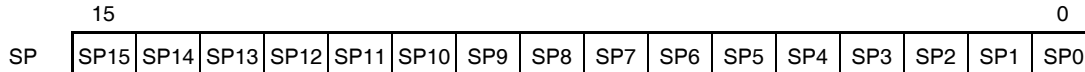
(f) 进位标志 (CY)

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转移值，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 3-14. 堆栈指针格式



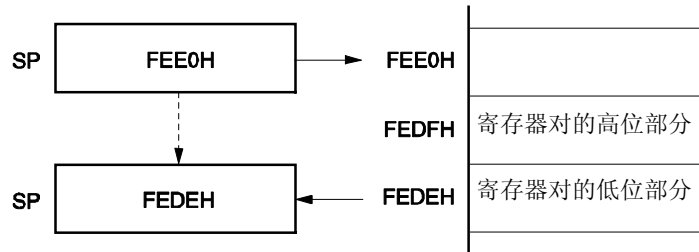
在向堆栈写(保存)数据时，堆栈指针 SP 递减，而从堆栈中读出(恢复)数据时，堆栈指针累加。

堆栈的数据存储/恢复操作过程如图 3-15 和 3-16 所示。

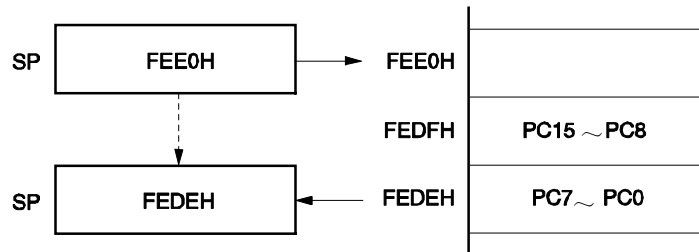
注意事项 由于复位信号产生时，SP 的内容不确定，所以在使用堆栈前必须先对 SP 初始化。

图 3-15. 将数据存入堆栈

(a) PUSH rp 指令 (当 SP = FEE0H 时)



(b) CALL, CALLF, CALLT 指令 (当 SP = FEE0H 时)



(c) 中断, BRK 指令 (当 SP = FEE0H 时)

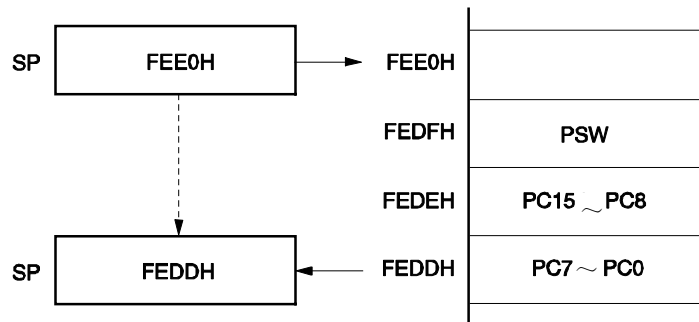
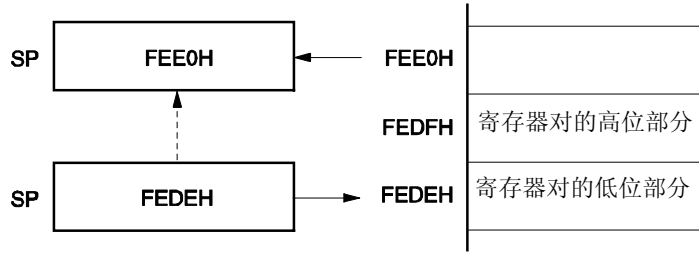
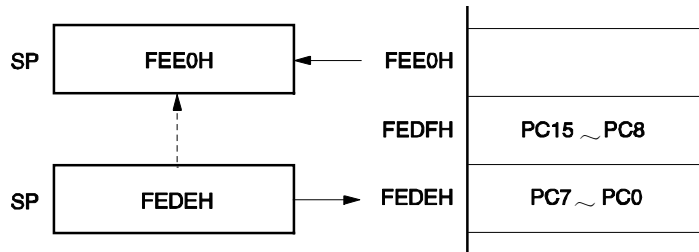


图 3-16. 从堆栈读出数据

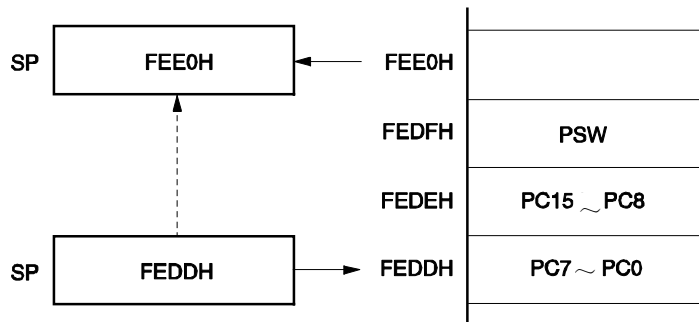
(a) POP rp 指令 (当 SP = FEDEH 时)



(b) RET 指令 (当 SP = FEDEH 时)



(c) RETI, RETB 指令 (当 SP = FEDDH 时)



3.2.2 通用寄存器

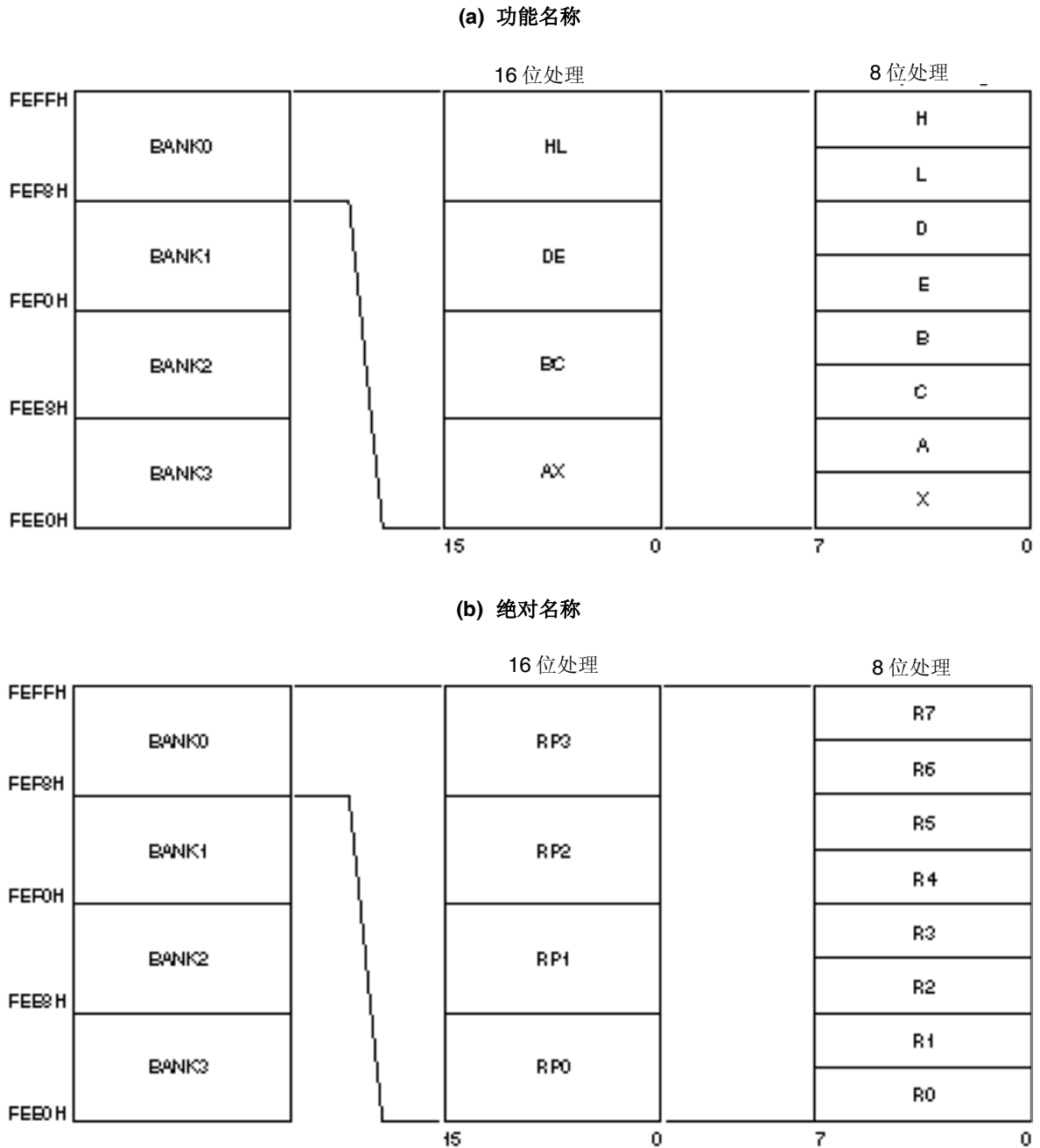
通用寄存器映射到数据存储器特定的地址空间为 FEE0H ~ FEFFH。通用寄存器共有四个 bank，每一个 bank 由 8 个 8 位寄存器(X, A, C, B, E, D, L 和 H)组成。

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, RP0 ~ RP3)。

用于指令执行的寄存器 bank 由 CPU 控制指令(SEL RBn)来设置。由于 4 个寄存器 bank 的结构，通过一个用于正常处理的寄存器和另一个用于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。

图 3-17. 通用寄存器结构



3.2.3 特殊功能寄存器 (SFR)

与通用寄存器不同，每个特殊功能寄存器都有特定的功能。

特殊寄存器(SFR)分配在CPU的在FF00H ~ FFFFH区域，和LCD控制器/驱动器LCDCTL的00H ~ 03H区域。

CPU的特殊功能寄存器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 (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。

备注 对于 LCD 控制器/驱动器中的特殊寄存器(SFR)的操作方法，请参看 16.7 与 LCD 控制器/驱动器的通讯。

表 3-7 为特殊功能寄存器列表。表中术语的含义如下。

- 符号
符号表示特殊功能寄存器的地址。它在 RA78K0 中是保留字，并使用 CC78K0 中的 #pragma sfr 指令定义为一个 sfr 变量。在使用 RA78K0, ID78K0-QB 和 SM+ 时，可以将符号作为指令操作数执行写操作。
- R/W
表示特殊功能寄存器可读或可写。
R/W: 可读/写
R: 只读
W: 只写
- 可操作的位单元
表示可操作的位单元 (1, 8 或 16)。“-”表示不可操作的位单元。
- 复位后
表示复位信号产生后每个寄存器的状态。

表 3-7. 特殊功能寄存器列表 (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	
FF00H	端口寄存器 0	P0	R/W	√	√	–	00H
FF01H	端口寄存器 1	P1	R/W	√	√	–	00H
FF02H	端口寄存器 2 ^注	P2	R/W	√	√	–	00H
FF03H	端口寄存器 3	P3	R/W	√	√	–	00H
FF06H	端口寄存器 6	P6	R/W	√	√	–	00H
FF07H	端口寄存器 7	P7	R/W	√	√	–	00H
FF08H	10 位 A/D 转换结果寄存器 ^注	ADCR	R	–	–	√	0000H
FF09H	8 位 A/D 转换结果寄存器 ^注	ADCRH	R	–	√	–	00H
FF0AH	接收缓冲寄存器 6	RXB6	R	–	√	–	FFH
FF0BH	发送缓冲寄存器 6	TXB6	R/W	–	√	–	FFH
FF0CH	端口寄存器 12	P12	R/W	√	√	–	00H
FF0DH	端口寄存器 13	P13	R/W	√	√	–	00H
FF0FH	串行 I/O 移位寄存器 10	SIO10	R	–	√	–	00H
FF10H	16 位定时器计数器 00	TM00	R	–	–	√	0000H
FF11H							
FF12H	16 位定时器捕捉/比较寄存器 000	CR000	R/W	–	–	√	0000H
FF13H							
FF14H	16 位定时器捕捉/比较寄存器 010	CR010	R/W	–	–	√	0000H
FF15H							
FF16H	8 位定时器计数器 50	TM50	R	–	√	–	00H
FF17H	8 位定时器比较寄存器 50	CR50	R/W	–	√	–	00H
FF18H	8 位定时器 H 比较寄存器 00	CMP00	R/W	–	√	–	00H
FF19H	8 位定时器 H 比较寄存器 10	CMP10	R/W	–	√	–	00H
FF1AH	8 位定时器 H 比较寄存器 01	CMP01	R/W	–	√	–	00H
FF1BH	8 位定时器 H 比较寄存器 11	CMP11	R/W	–	√	–	00H
FF1FH	8 位定时器计数器 51	TM51	R	–	√	–	00H
FF20H	端口模式寄存器 0	PM0	R/W	√	√	–	FFH
FF21H	端口模式寄存器 1	PM1	R/W	√	√	–	FFH
FF22H	端口模式寄存器 2 ^注	PM2	R/W	√	√	–	FFH
FF23H	端口模式寄存器 3	PM3	R/W	√	√	–	FFH
FF26H	端口模式寄存器 6	PM6	R/W	√	√	–	FFH
FF27H	端口模式寄存器 7	PM7	R/W	√	√	–	FFH
FF28H	A/D 转换器模式寄存器 ^注	ADM	R/W	√	√	–	00H
FF29H	模拟输入通道选择寄存器 ^注	ADS	R/W	√	√	–	00H
FF2CH	端口模式寄存器 12	PM12	R/W	√	√	–	FFH
FF2EH	端口模式寄存器 14	PM14	R/W	√	√	–	FFH
FF2FH	A/D 端口配置寄存器 ^注	ADPC	R/W	√	√	–	00H
FF30H	上拉电阻选择寄存器 0	PU0	R/W	√	√	–	00H
FF31H	上拉电阻选择寄存器 1	PU1	R/W	√	√	–	00H
FF33H	上拉电阻选择寄存器 3	PU3	R/W	√	√	–	00H
FF37H	上拉电阻选择寄存器 7	PU7	R/W	√	√	–	00H

注 仅限于 μ PD78F037x。

表 3-7. 特殊功能寄存器列表 (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	
FF3CH	上拉电阻选择寄存器 12	PU12	R/W	√	√	-	00H
FF40H	时钟输出选择寄存器	CKS	R/W	√	√	-	00H
FF41H	8 位定时器比较寄存器 51	CR51	R/W	-	√	-	00H
FF43H	8 位定时器模式控制寄存器 51	TMC51	R/W	√	√	-	00H
FF48H	外部中断上升沿允许寄存器	EGP	R/W	√	√	-	00H
FF49H	外部中断下降沿允许寄存器	EGN	R/W	√	√	-	00H
FF4FH	输入切换控制寄存器	ISC	R/W	√	√	-	00H
FF50H	异步串行接口操作模式寄存器 6	ASIM6	R/W	√	√	-	01H
FF53H	异步串行接口接收错误状态寄存器 6	ASIS6	R	-	√	-	00H
FF55H	异步串行接口发送状态寄存器 6	ASIF6	R	-	√	-	00H
FF56H	时钟选择寄存器 6	CKSR6	R/W	-	√	-	00H
FF57H	波特率发生器控制寄存器 6	BRGC6	R/W	-	√	-	FFH
FF58H	异步串行接口控制寄存器 6	ASICL6	R/W	√	√	-	16H
FF60H	余数寄存器 0 ^注	SDR0	R	SDR0L	√	√	00H
FF61H				SDR0H	-	√	-
FF62H	乘法/除法数据寄存器 A0 ^注	MDA0L	R/W	MDA0LL	√	√	00H
FF63H				MDA0LH	-	√	-
FF64H		MDA0H	R/W	MDA0HL	√	√	00H
FF65H				MDA0HH	-	√	-
FF66H	乘法/除法数据寄存器 B0 ^注	MDB0	R/W	MDB0L	√	√	00H
FF67H				MDB0H	-	√	-
FF68H	乘法/除法控制寄存器 0 ^注	DMUC0	R/W	√	√	-	00H
FF69H	8 位定时器 H 模式寄存器 0	TMHMD0	R/W	√	√	-	00H
FF6AH	定时器时钟选择寄存器 50	TCL50	R/W	√	√	-	00H
FF6BH	8 位定时器模式控制寄存器 50	TMC50	R/W	√	√	-	00H
FF6CH	8 位定时器 H 模式寄存器 1	TMHMD1	R/W	√	√	-	00H
FF6DH	8 位定时器 H 载波控制寄存器 1	TMCYC1	R/W	√	√	-	00H
FF6EH	按键返回模式寄存器	KRM	R/W	√	√	-	00H
FF6FH	钟表定时器操作模式寄存器	WTM	R/W	√	√	-	00H
FF70H	异步串行接口操作模式寄存器 0	ASIM0	R/W	√	√	-	01H
FF71H	波特率发生器控制寄存器 0	BRGC0	R/W	-	√	-	1FH
FF72H	接收缓冲寄存器 0	RXB0	R	-	√	-	FFH
FF73H	异步串行接口接收错误状态寄存器 0	ASIS0	R	-	√	-	00H
FF74H	发送移位寄存器 0	TXS0	W	-	√	-	FFH

注 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D。

表 3-7. 特殊功能寄存器列表 (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	
FF80H	串行操作模式寄存器 10	CSIM10	R/W	√	√	–	00H
FF81H	串行时钟选择寄存器 10	CSIC10	R/W	√	√	–	00H
FF84H	发送缓冲寄存器 10	SOTB10	R/W	–	√	–	00H
FF8CH	定时器时钟选择寄存器 51	TCL51	R/W	√	√	–	00H
FF99H	看门狗定时器允许寄存器	WDTE	R/W	–	√	–	1AH/9AH ^{注2}
FF9FH	时钟操作模式选择寄存器	OSCCTL	R/W	√	√	–	00H
FFA0H	内部振荡模式寄存器	RCM	R/W	√	√	–	80H ^{注3}
FFA1H	主时钟模式寄存器	MCM	R/W	√	√	–	00H
FFA2H	主 OSC 控制寄存器	MOC	R/W	√	√	–	80H
FFA3H	振荡稳定时间计数器的状态寄存器	OSTC	R	√	√	–	00H
FFA4H	振荡稳定时间选择寄存器	OSTS	R/W	–	√	–	05H
FFA5H	IIC 移位寄存器 0	IIC0	R/W	–	√	–	00H
FFA6H	IIC 控制寄存器 0	IICC0	R/W	√	√	–	00H
FFA7H	从设备地址寄存器 0	SVA0	R/W	–	√	–	00H
FFA8H	IIC 时钟选择寄存器 0	IICCL0	R/W	√	√	–	00H
FFA9H	IIC 功能扩展寄存器 0	IICX0	R/W	√	√	–	00H
FFAAH	IIC 状态寄存器 0	IICS0	R	√	√	–	00H
FFABH	IIC 标志寄存器 0	IICF0	R/W	√	√	–	00H
FFACH	复位控制标志寄存器	RESF	R	–	√	–	00H ^{注4}
FFB0H	16 位定时器计数器 01 ^{注1}	TM01	R	–	–	√	0000H
FFB1H							
FFB2H	16 位定时器捕捉/比较寄存器 001 ^{注1}	CR001	R/W	–	–	√	0000H
FFB3H							
FFB4H	16 位定时器捕捉/比较寄存器 011 ^{注1}	CR011	R/W	–	–	√	0000H
FFB5H							
FFB6H	16 位定时器模式控制寄存器 01 ^{注1}	TMC01	R/W	√	√	–	00H
FFB7H	预分频模式寄存器 01 ^{注1}	PRM01	R/W	√	√	–	00H
FFB8H	捕捉/比较控制寄存器 01 ^{注1}	CRC01	R/W	√	√	–	00H
FFB9H	16 位定时器输出控制寄存器 01 ^{注1}	TOC01	R/W	√	√	–	00H
FFBAH	16 位定时器模式控制寄存器 00	TMC00	R/W	√	√	–	00H
FFBBH	预分频模式寄存器 00	PRM00	R/W	√	√	–	00H
FFBCH	捕捉/比较控制寄存器 00	CRC00	R/W	√	√	–	00H
FFBDH	16 位定时器输出控制寄存器 00	TOC00	R/W	√	√	–	00H

- 注
1. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D。
 2. 通过设置选项字节决定了 WDTE 复位后的值。
 3. 这个寄存器在复位释放后立即为 00H，但是内部高速振荡器的振荡稳定时间已经过去之后，该寄存器会自动改变为 80H。
 4. RESF 复位后的值依复位源而变化。

表 3-7. 特殊功能寄存器列表 (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	
FFBEH	低电压检测寄存器	LVIM		R/W	√	√	–	00H ^{注1}
FFBFH	低电压检测等级选择寄存器	LVIS		R/W	√	√	–	00H ^{注1}
FFE0H	中断请求标志寄存器 0L	IF0	IF0L	R/W	√	√	√	00H
FFE1H	中断请求标志寄存器 0H		IF0H	R/W	√	√		00H
FFE2H	中断请求标志寄存器 1L	IF1	IF1L	R/W	√	√	√	00H
FFE3H	中断请求标志寄存器 1H		IF1H	R/W	√	√		00H
FFE4H	中断屏蔽标志寄存器 0L	MK0	MK0L	R/W	√	√	√	FFH
FFE5H	中断屏蔽标志寄存器 0H		MK0H	R/W	√	√		FFH
FFE6H	中断屏蔽标志寄存器 1L	MK1	MK1L	R/W	√	√	√	FFH
FFE7H	中断屏蔽标志寄存器 1H		MK1H	R/W	√	√		FFH
FFE8H	优先级规范标志寄存器 0L	PR0	PR0L	R/W	√	√	√	FFH
FFE9H	优先级规范标志寄存器 0H		PR0H	R/W	√	√		FFH
FFEAH	优先级规范标志寄存器 1L	PR1	PR1L	R/W	√	√	√	FFH
FFEBH	优先级规范标志寄存器 1H		PR1H	R/W	√	√		FFH
FFF0H	内部存储器容量切换寄存器 ^{注2}	IMS		R/W	–	√	–	CFH
<R> FFF3H	存储器 bank 选择寄存器	BANK		R/W	–	√	–	00H
FFF4H	内部扩展 RAM 容量切换寄存器 ^{注2}	IXS		R/W	–	√	–	0CH
FFFBH	处理器时钟控制寄存器	PCC		R/W	√	√	–	01H
LCDCTL's 00H	LCD 模式设定寄存器	LCDMD		R/W	–	√	–	00H
LCDCTL's 01H	LCD 显示模式寄存器	LCDM		R/W	–	√	–	00H
LCDCTL's 02H	LCD 时钟控制寄存器	LCDC		R/W	–	√	–	00H
LCDCTL's 03H	LCD 升压控制寄存器 0	VLCG0		R/W	–	√	–	00H

注 1. LVIM 和 LVIS 的复位值，根据不同的复位源而变化。

2. 不管内部存储器容量有多大，所有 78K0/LF2 产品的内部存储器容量切换寄存器(IMS)和内部扩展 RAM 容量切换寄存器(IXS)的初始值都是固定的(IMS = CFH, IXS = 0CH)。因此对应于每种产品的设置值如下所示。

Flash 存储器版本(78K0/LF2)	IMS	IXS	ROM 容量	内部高速 RAM 容量	内部扩展 RAM 容量
μPD78F0372, 78F0382	C6H	0CH	24 KB	1 KB	–
PD78F0373, 78F0383	C8H		32 KB		
PD78F0374, 78F0384	CCH	0AH	48 KB		1 KB
PD78F0375, 78F0385	CFH	08H	60 KB		2 KB
PD78F0376, 78F0386, 78F0376D, 78F0386D ^{注3}	CCH	04H	96 KB		4 KB

<R> 3. 具有片上调试功能的产品，其 ROM 和 RAM 的容量可根据调试目标产品进行调整，根据调试目标产品设置 IMS 和 IXS 值。

3.3 指令地址寻址

一条指令的地址是由程序计数器(PC)决定的。根据执行指令时所获取的下一条指令字节数，程序计数器(PC)的内容自动增加(每个字节加 1)。在执行转移指令时，将程序计数器(PC)的内容设置为转移目的地址，并按以下寻址方式确定地址。(要了解每条指令的详细信息，请参阅 **78K/0 系列指令用户手册(U12326E)**)。

3.3.1 相对寻址

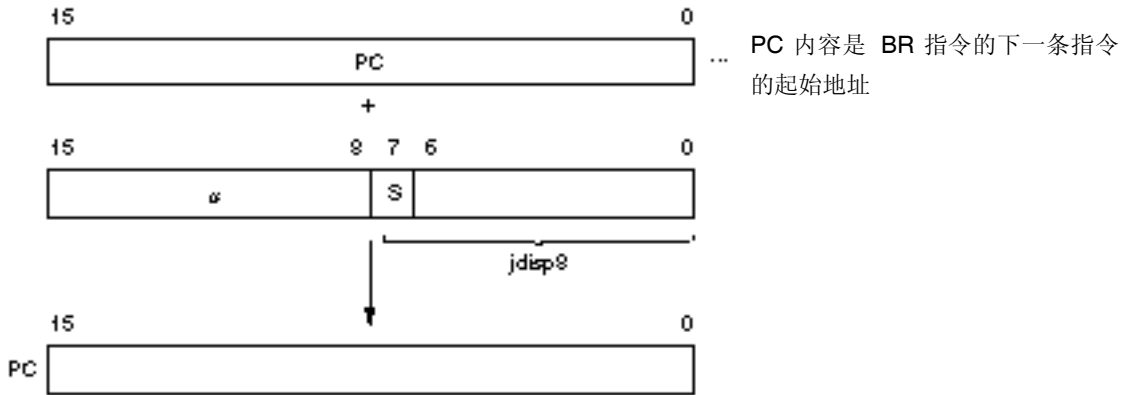
[功能]

将一条指令的 8 位立即数（偏移量：jdisp8）与下一条指令的起始地址相加，结果赋给程序计数器(PC)，然后转向相加结果指向的地址。这个偏移量是带符号数的补码(-128 ~ +127)，其中第 7 位是符号位。

换句话说，在相对寻址中，分支的范围是从下一条指令起始地址的-128 到+127 之间。

当执行“BR \$addr16”指令或条件转移指令时，将执行相对寻址功能。

[图示]



当 S=0 时，α 的所有位都为 0

当 S=1 时，α 的所有位都为 1

3.3.2 立即寻址

[功能]

将指令中的立即数赋给程序计数器(PC)，然后转向该地址。

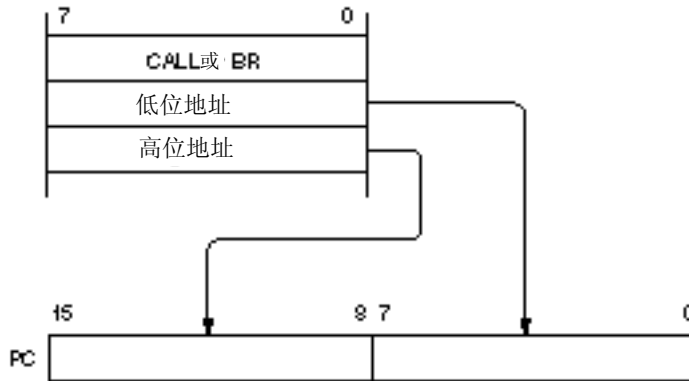
在执行“CALL !addr16”指令、“BR !addr16”指令或“CALLF !addr11”指令时，将执行立即寻址功能。

CALL !addr16 和 BR !addr16 指令的转移地址范围是所有存储空间。

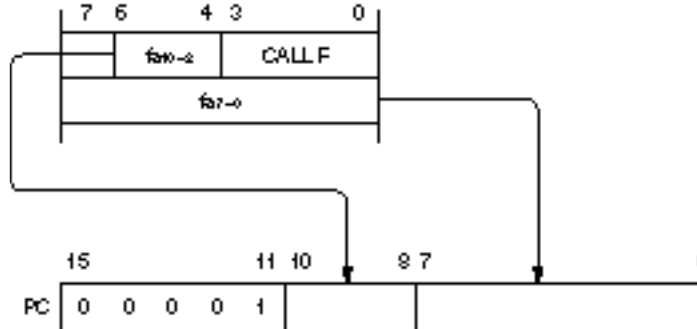
CALLF !addr11 指令的转移地址范围在 0800H 与 0FFFH 之间。

[图示]

CALL !addr16 和 BR !addr16 指令



CALLF !addr11 指令



3.3.3 表间接寻址

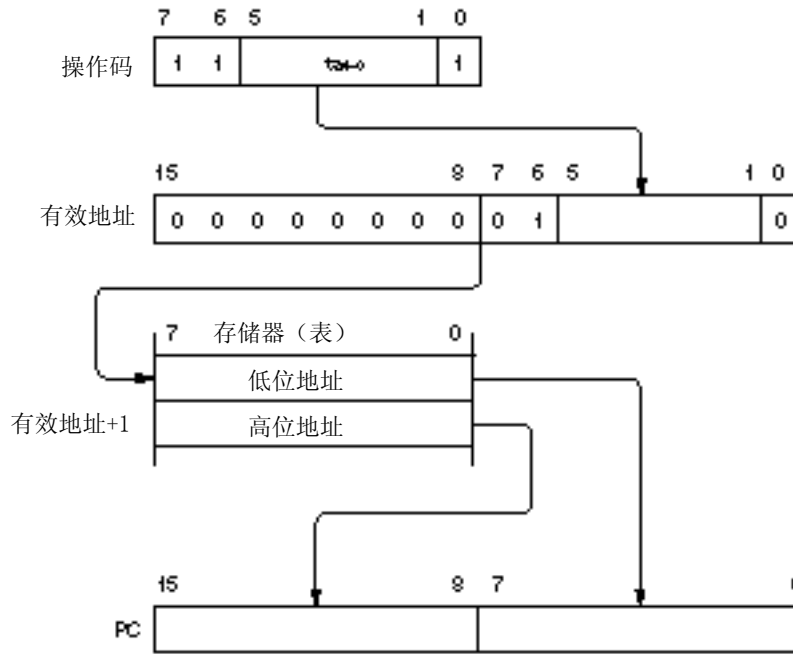
[功能]

通过指令码第 1 位到第 5 位的立即数，访问特定存储区中表的内容(转移目的地址)，并将表的内容赋给程序计数器(PC)，然后转向该地址执行程序。

在执行 CALLT [addr5]指令时，进行表间接寻址。

该指令访问的地址范围是表 40H~7FH 中所存储的地址，转移地址范围可以是整个存储器空间。

[图示]



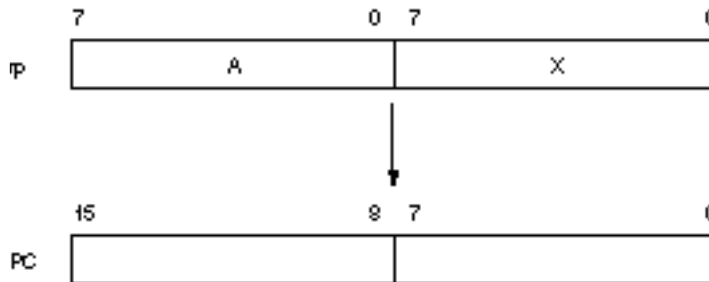
3.3.4 寄存器寻址

[功能]

将寄存器对(AX)的内容赋给程序计数器(PC)，然后转向该地址。

“BR AX”指令将执行寄存器寻址功能。

[图示]



3.4 操作数地址寻址

以下方法用来规定指令执行期间寄存器寻址和存储器寻址所进行的操作。

3.4.1 隐含寻址

[功能]

这种寻址方式自动寻址通用寄存器中作为累加器(A 和 AX)使用的寄存器。
在 78K0/LF2 系列指令中下列指令采用隐含寻址方式。

指令	隐含寻址所指定的寄存器
MULU	A 寄存器存放被乘数, AX 寄存器存放运算结果
DIVUW	AX 寄存器用于存放被除数和商
ADJBA/ADJBS	存放进行十进制调整后的数据
ROR4/ROL4	存放用于数字循环的数字数据

[操作数格式]

由于指令自动采用隐含寻址方式, 所以没有特定的操作数格式。

[举例]

以 MULU X 指令为例, 这是一条 8 位乘 8 位的乘法运算指令, A 寄存器与 X 寄存器相乘的结果存放在 AX 中。在这个例子中 A 寄存器与 AX 寄存器均由隐含寻址方式指定。

3.4.2 寄存器寻址

[功能]

寄存器寻址方式将通用寄存器作为操作数进行访问，并由寄存器组选择标志(RBS0 ~ RBS1)和指令中的寄存器标识码(Rn 和 RPn)，来指定需要访问的通用寄存器。

当具有下列操作数格式的指令执行时，采用寄存器寻址方式。如果使用 8 位寄存器，则指令码中有 3 位用来表示一个 8 位寄存器。

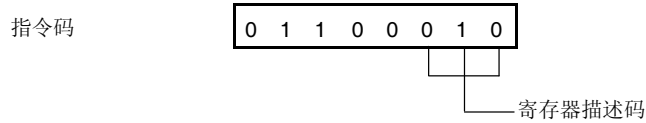
[操作数格式]

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

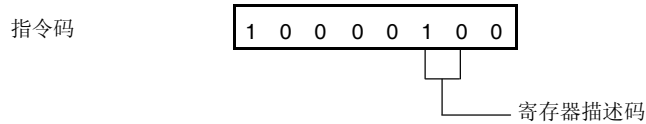
‘r’ 和 ‘rp’可用绝对名称(R0 ~ R7 以及 RP0 ~ RP3)和功能名称(X, A, C, B, E, D, L, H, AX, BC, DE 以及 HL)来描述。

[举例]

MOV A, C; 当选择通用寄存器 C 为 “r” 时



INCW DE; 当选择通用寄存器组 DE 为 “rp” 时



3.4.3 直接寻址

[功能]

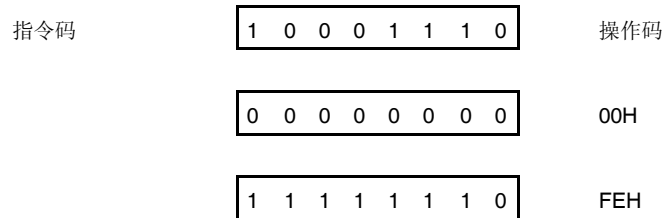
存储器会根据指令字中的操作数地址进行直接寻址操作。

[操作数格式]

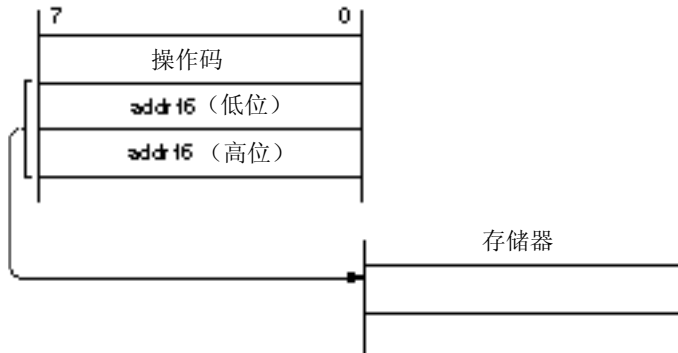
标识符	描述
addr16	标号或 16 位立即数

[举例]

MOV A, !0FE00H; 将!addr16 设置为 FE00H 时



[图示]



3.4.5 特殊功能寄存器 (SFR) 寻址

[功能]

通过指令中的 8 位立即数对存储器的特殊功能寄存器 (SFR) 区域进行寻址。

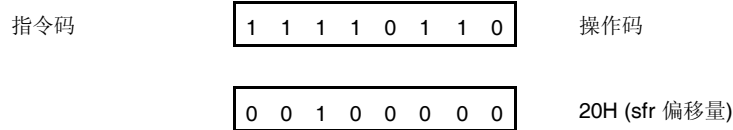
寻址区间为 FF00H ~ FFCFH 以及 FFE0H ~ FFFFH，共 240 字节。而映射在 FF00H ~ FF1FH 区间的特殊功能寄存器 (SFR) 则采用短直接寻址方式。

[操作数格式]

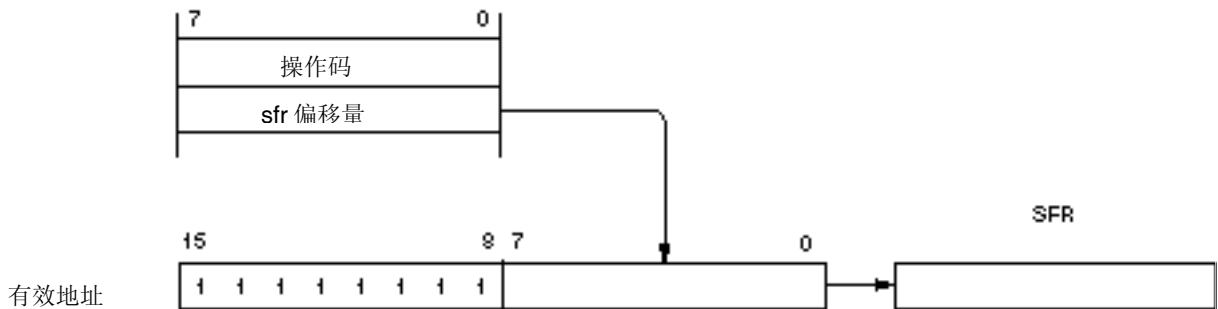
标识符	描述
sfr	特殊功能寄存器名
sfrp	16 位可操作特殊功能寄存器名 (仅使用偶地址)

[举例]

MOV PM0, A; 选择 PM0(FF20H)作为 sfr



[图示]



3.4.6 寄存器间接寻址

[功能]

根据寄存器对的内容进行寻址。该寄存器对由寄存器组选择标志(RBS0 和 RBS1)和指令字中的寄存器对指定码指定。

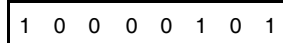
[操作数格式]

标识符	描述
-	[DE], [HL]

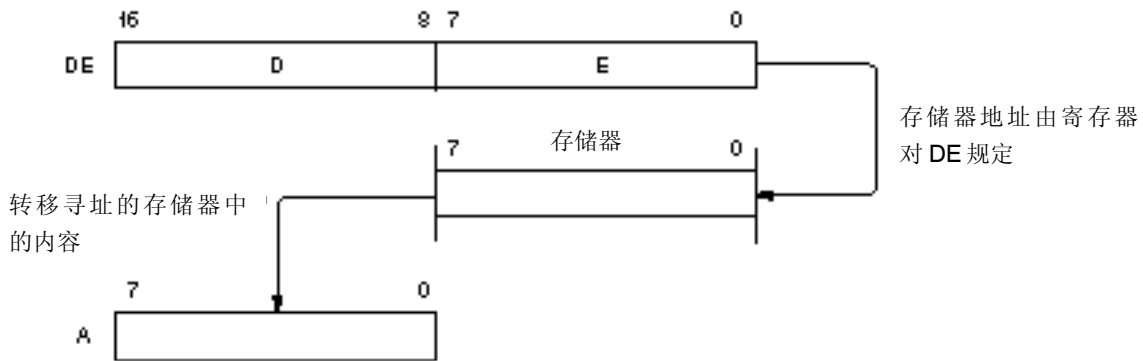
[举例]

MOV A, [DE]; 选择 DE 寄存器对作为操作数时

指令码



[图示]



3.4.7 基址寻址

[功能]

将 8 位立即数加到 HL 寄存器对中，HL 寄存器对作为基地址寄存器。根据相加结果寻址。需要访问的 HL 寄存器对属于由寄存器组选择标志(RBS0 和 RBS1)确定的寄存器组。通过将偏移量扩展为 16 位正数，来完成加法操作，第 16 位的进位忽略不计。

[操作数格式]

标识符	描述
-	[HL + byte]

[举例]

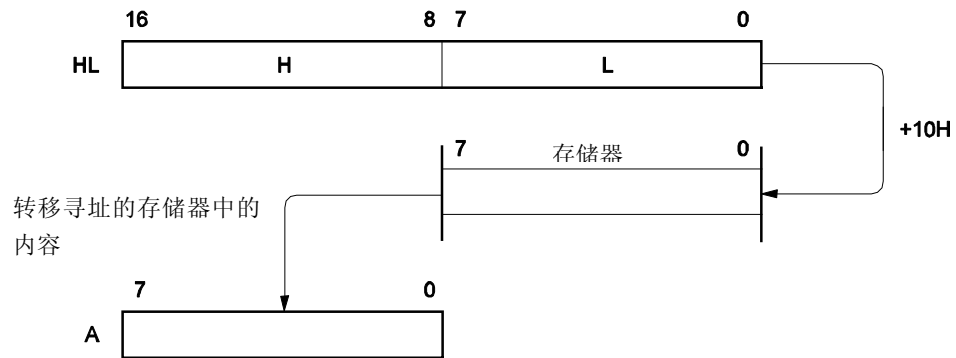
MOV A, [HL + 10H]; byte 的值为 10H 时

指令码

1 0 1 0 1 1 1 0

0 0 0 1 0 0 0 0

[图示]



3.4.8 基址变址寻址

[功能]

将 B 或 C 寄存器的内容加到 HL 寄存器中，HL 寄存器作为基址寄存器，并根据相加结果去寻址。需要访问的 HL、B 和 C 寄存器属于由寄存器组选择标志(RBS0 和 RBS1)确定的寄存器组。通过将 B 或 C 寄存器扩展为一个 16 位的正数来完成加法运算，第 16 位的进位忽略不计。

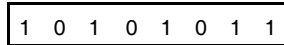
[操作数格式]

标识符	描述
-	[HL + B], [HL + C]

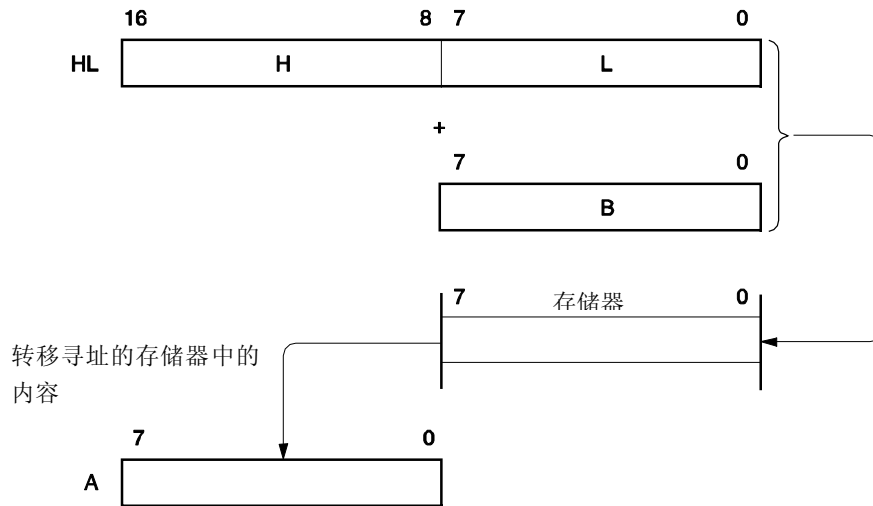
[举例]

MOV A, [HL + B]; 选择 B 寄存器

指令码



[图示]



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。

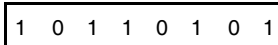
当执行 **PUSH**, **POP**, 子程序调用和返回指令时, 或者产生中断请求时保存或恢复寄存器操作时, 将自动采用这种寻址方式。

该方式仅对内部高速 **RAM** 区域进行寻址。

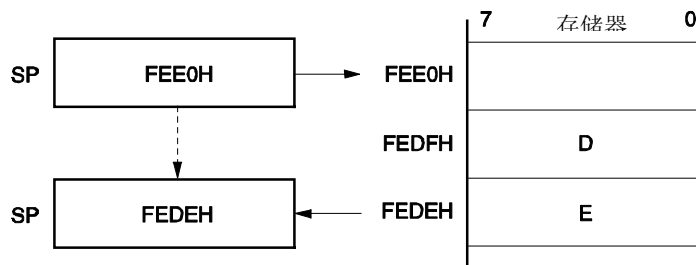
[举例]

PUSH DE; 保存在 DE 寄存器中

指令码



[图示]



<R>

3.5 存储器 Bank (仅限于 μ PD78F0376, 78F0376D, 78F0386 和 78F0386D)

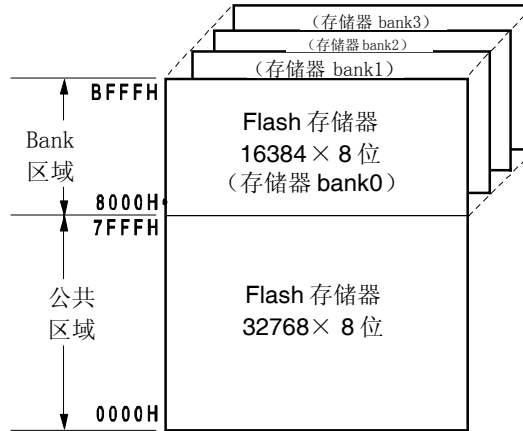
The μ PD78F0376, 78F0376D, 78F0386 和 78F0386D 可以通过选择使用从 8000H ~ BFFFH 的存储器空间的存储器 bank 来实现 96KB 的 ROM 容量。

μ PD78F0376, 78F0376D, 78F0386 和 78F0386D 有如下的存储器 bank 0 ~ 3。

存储器 bank 通过使用存储器 bank 选择寄存器(BANK)来选择。

- 注意事项**
1. 在不同存储器 bank 中，不能调用指令。
 2. 不同的存储器 bank，跳转和访问不能够直接执行。执行不同存储器 bank 的跳转和访问，需要通过通用区域。
 3. 在通用区域中分配中断服务。
 4. 从 7FFFH ~ 8000H 扩展的指令仅可在存储器 bank 0 中执行。

图 3-18. 内部 ROM (Flash 存储器) 结构



<R> 3.6 存储器 Bank 选择寄存器 (BANK) (仅限于 μ PD78F0376, 78F0376D, 78F0386 和 78F0386D)

存储器 Bank 选择寄存器 (BANK) 用来选择使用的存储器 bank。

BANK 可以通过 8 位寄存器操作指令来设置。

复位信号使 BANK 清零 00H。

图 3-19. 存储器 Bank 选择寄存器 (BANK) 的格式

地址: FFF3H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
BANK	0	0	0	0	0	BANK2	BANK1	BANK0

BANK2	BANK1	BANK0	Bank 设定
0	0	0	通用区域 (32 K) + 存储器 bank 0 (16 K)
0	0	1	通用区域 (32 K) + 存储器 bank 1 (16 K)
0	1	0	通用区域 (32 K) + 存储器 bank 2 (16 K)
0	1	1	通用区域 (32 K) + 存储器 bank 3 (16 K)
其他			禁止设置

注意事项 一定要在通用区域中 (0000H ~ 7FFFH) 改变 BANK 寄存器的值。

如果在 bank 区域 (8000H ~ BFFFH) 范围内改变 BANK 寄存器的值, 将在 CPU 中产生一个被忽略的程序循环。因此, 永远不要在 bank 区域中改变 BANK 寄存器的值。

<R> 3.7 选择存储器 Bank (仅限 μ PD78F0376, 78F0376D, 78F0386 和 78F0386D)

通过存储器 bank 选择寄存器 (BANK) 设定存储器 bank, 它被反映在 bank 区域, 并且可以被寻址。因此, 要访问一个不同于当前选择的存储器 bank, 必须用 BANK 寄存器来选择存储器 bank。

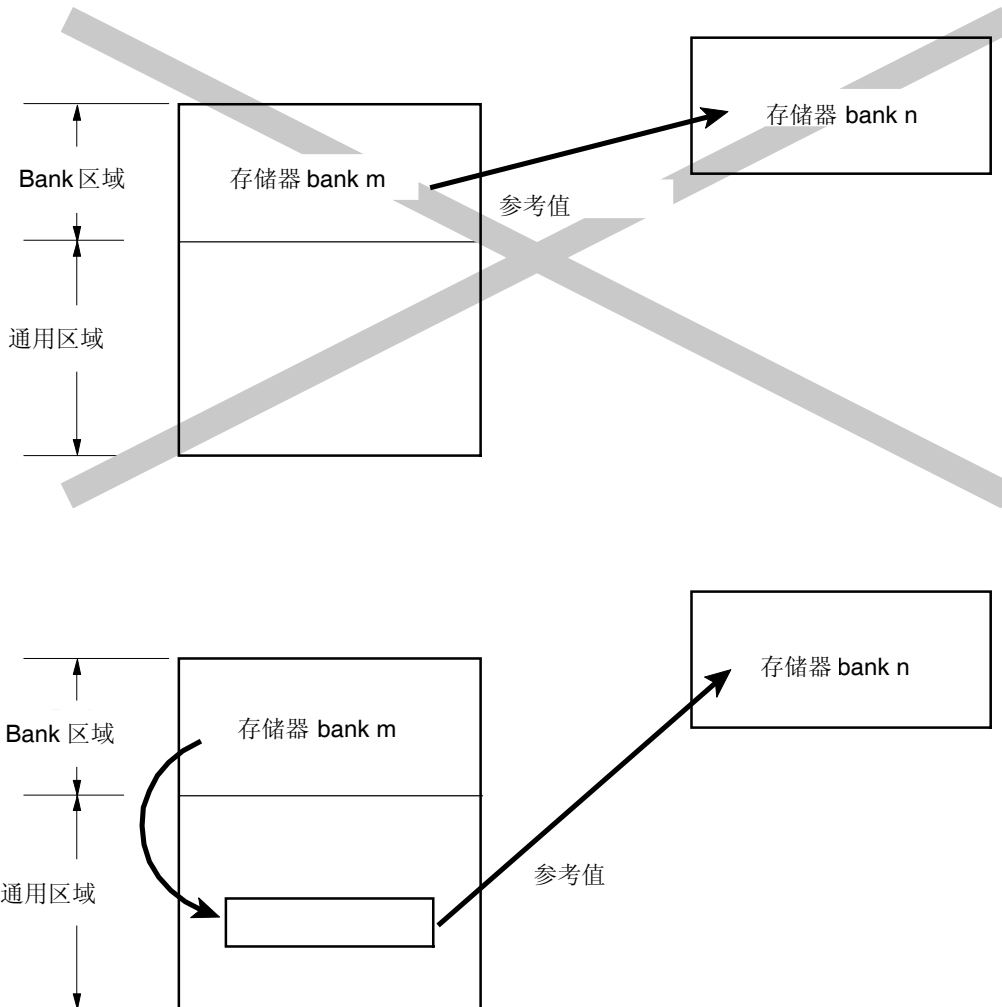
BANK 寄存器的值在 (8000H ~ BFFFH) 范围内一定不能被改变。因此, 为了改变存储器 bank, 可通过指令转移至通用区域 (0000H ~ 7FFFH), 并在通用区域中改变 BANK 寄存器的值。

- 注意事项**
1. 在不同存储器 bank 中, 不能调用指令。
 2. 在不同的存储器 bank 中, 跳转和访问不能够直接执行。执行不同存储器 bank 之间的跳转和访问, 需要通过通用区域。
 3. 在通用区域中分配中断服务。
 4. 从 7FFFH ~ 8000H 扩展的指令仅可在存储器 bank 0 中执行。

3.7.1 存储器 bank 之间的参考值

值不能直接从一个存储器 bank 参考到另一个。

从一个存储器 bank 访问另一个存储器 bank, 先跳转到通用区域 (0000H ~ 7FFFH), 在通用区域改变 BANK 寄存器的设置值, 然后可以被参考。



- 软件例程 (寄存器 A 中存储一个参考值)

```

RAMD   DSEG   SADDR
R_BNKA: DS     2           ; Secures RAM for specifying an address at the reference destination.
R_BNKN: DS     1           ; Secures RAM for specifying a memory bank number at the reference destination.
R_BNKRN: DS    1           ; Secures RAM for saving a memory bank number at the reference source.

-----

ETRC    CSEG   UNIT
ENTRY:

        MOV    R_BNKN,#BANKNUM DATA1 ; Stores the memory bank number at the reference destination.
        MOVW  R_BNKA,#DATA1           ; Stores the address at the reference destination.
        CALL  !BNKRD                   ; Calls a subroutine for referencing between memory banks.
        :
        :

-----

BNKC    CSEG   AT      7000H

BNKRD:  ; Subroutine for referencing between memory banks.
        PUSH  HL                       ; Saves the contents of the HL register.
        MOV   A,R_BNKN                 ; Acquires the memory bank number at the reference destination.
        XCH  A,BANK                    ; Swaps the memory bank number at the reference source for that at the reference
        ; destination
        MOV   R_BNKRN,A                ; Saves the memory bank number at the reference source.
        XCHW AX,HL                     ; Saves the contents of the X register.
        MOVW AX,R_BNKA                 ; Acquires the address at the reference destination.
        XCHW AX,HL                     ; Specifies the address at the reference destination.
        MOV   A,[HL]                   ; Reads the target value.
        XCH  A,R_BNKRN                 ; Acquires the memory bank number at the reference source.
        MOV   BANK,A                   ; Specifies the memory bank number at the reference source.
        MOV   A,R_BNKRN                 ; Write the target value to the A register.
        POP  HL                         ; Restores the contents of the HL register.

        RET                             ; Return

-----

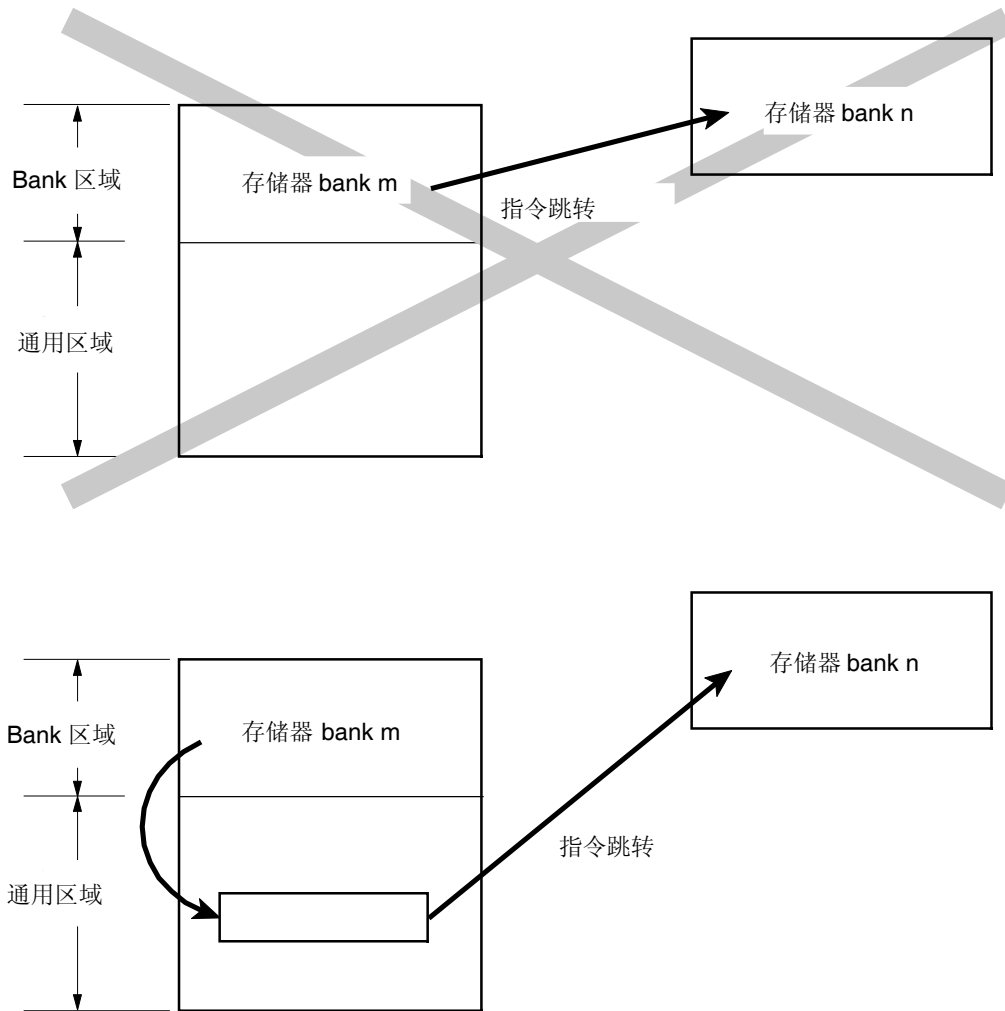
DATA    CSEG   BANK3
DATA1:  DB     0AAH

END
    
```

3.7.2 存储器 bank 间的跳转指令

指令不能从一个存储器 bank 直接跳转到另一个。

为了从一个存储器 bank 访问另一个存储器 bank，先跳转到通用区域 (0000H ~ 7FFFH)，在通用区域改变 BANK 寄存器的设置值，然后再执行跳转指令。



• 软件例程 1 (从所有区域跳转)

```

RAMD   DSEG   SADDR
R_BNKA: DS     2           ; Secures RAM for specifying a memory bank at the branch destination.
R_BNKN: DS     1           ; Secures RAM for specifying a memory bank number at the branch destination.
RSAVEAX: DS    2           ; Secures RAM for saving the AX register.

-----

ETRC    CSEG    UNIT
ENTRY:  MOV     R_BNKN,#BANKNUM TEST ; Stores the memory bank number at the branch destination in RAM.
        MOVW   R_BNKA,#TEST         ; Stores the address at the branch destination in RAM.
        BR     !BNKBR               ; Branches to inter-memory bank branch processing.
        :
        :

-----

BNKC    CSEG    AT      7000H      ;
BNKBR:  MOVW   RSAVEAX,AX           ; Saves the AX register.
        MOV   A,R_BNKN             ; Acquires the memory bank number at the branch destination.
        MOV   BANK,A               ; Specifies the memory bank number at the branch destination.
        MOVW  AX,R_BNKA            ; Specifies the address at the branch destination.
        PUSH  AX                   ; Sets the address at the branch destination to stack.
        MOVW  AX,RSAVEAX           ; Restores the AX register.
        RET                               ; Branch

-----

BN3     CSEG    BANK3
TEST:   MOV --
        :
        :
END
    
```

• 软件例程 2 (从通用区域跳转到其他 bank 区域)

```

ETRC    CSEG    AT      2000H
ENTRY:  MOV     R_BNKN,#BANKNUM TEST ; Stores the memory bank number at the branch destination in RAM.
        BR     !TEST               ; Stores the address at the branch destination in RAM.

-----

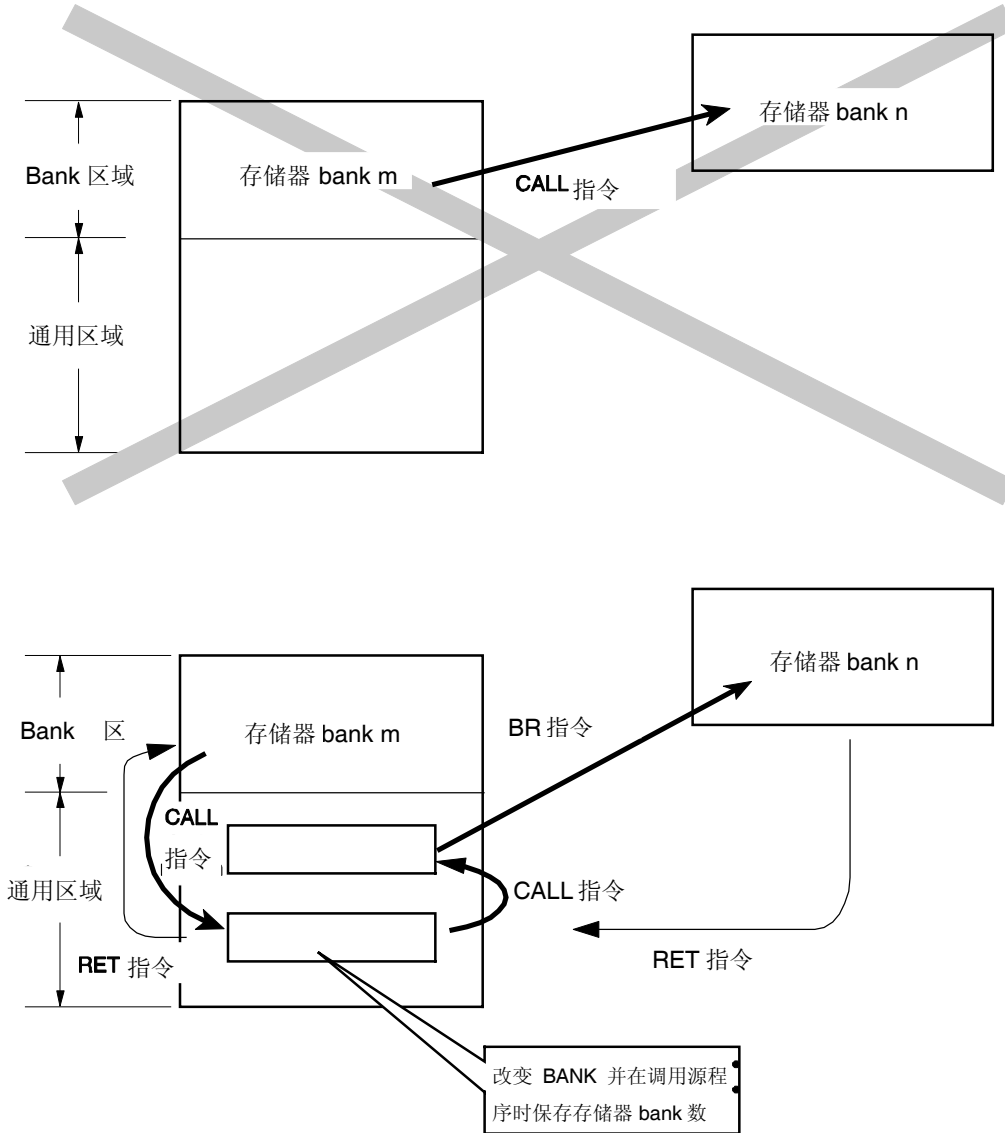
BN3     CSEG    BANK3
TEST:   MOV --
        :
        :
END
    
```

3.7.3 存储器 bank 间的子程序调用

在存储器 bank 间子程序不能被直接调用。

为了在不同存储器 bank 间调用子程序，先跳转到通用区域 (0000H ~ 7FFFH)，在通用区域通过使用 BANK 寄存器指定存储器 bank 的调用目标，执行 CALL 指令，并且通过这个指令跳转到所调用的目标文件。

这时，保存 BANK 寄存器的当前值到 RAM 中，在执行 RET 指令前恢复 BANK 寄存器的值。



• 软件例程

RAMD	DSEG	SADDR		
R_BNKA:	DS	2		; Secures RAM for specifying an address at the calling destination.
R_BNKN:	DS	1		; Secures RAM for specifying a memory bank number at the calling destination.
R_BNKRN:	DS	1		; Secures RAM for saving a memory bank number at the calling source.
RSAVEAX:	DS	2		; Secures RAM for saving the AX register.
<hr/>				
ETRC	CSEG	UNIT		
ENTRY:				
	MOV	R_BNKN,#BANKNUM	TEST	; Store the memory bank number at the calling destination in RAM.
	MOVW	R_BNKA,#TEST		; Stores the address at the calling destination in RAM.
	CALL	!BNKCAL		; Branches to an inter-memory bank calling processing routine.
		:		
		:		
<hr/>				
BNKC	CSEG	AT	7000H	
BNKCAL:				; Inter-memory bank calling processing routine
	MOVW	RSAVEAX,AX		; Saves the AX register.
	MOV	A,R_BNKN		; Acquires the memory bank number at the calling destination.
	XCH	A,BANK		; Changes the bank and acquires the memory bank number at the calling source.
	MOV	R_BNKRN,A		; Saves the memory bank number at the calling source to RAM.
	CALL	!BNKCAL		; Calls a subroutine to branch to the calling destination.
	MOVW	RSAVEAX,AX		; Saves the AX register.
	XCH	A,R_BNKRN		; Acquires the memory bank number at the calling source.
	MOV	BANK,A		; Specifies the memory bank number at the calling source.
	MOVW	RSAVEAX,AX		; Restores the AX register.
	RET			; Returns to the calling source.
BNKCAL:				
	MOVW	AX,R_BNKA		; Specifies the address at the calling destination.
	PUSH	AX		; Sets the address at the calling destination to stack.
	MOVW	AX,RSAVEAX		; Restores source AX register.
	RET	AX		; Branches to the calling destination.
<hr/>				
BN3	CSEG	BANK3		
TEST:				
	MOV ...			
	:			
	:			
	RET			
END				

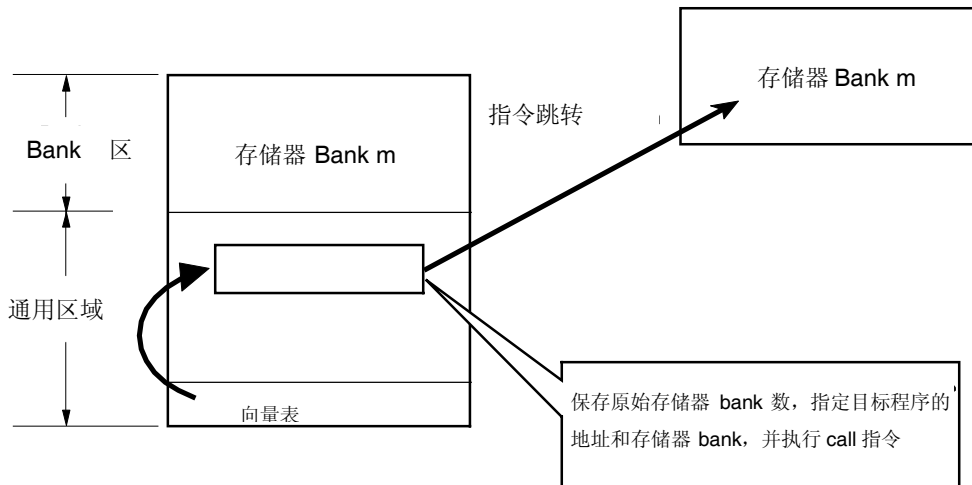
备注 在以上的软件例程中，并不支持多元处理。

3.7.4 通过中断指令跳转到 bank 区域

当中断发生时，指令可以通过矢量表跳转到 BANK 寄存器设定的存储器 bank，但是在中断产生时标识 BANK 寄存器是困难的。

因此，在通用区域 (0000H to 7FFFH) 中通过矢量表指定跳转目标地址，在通用区域中通过 BANK 寄存器指定跳转目标的存储器 bank，并且执行 CALL 指令。此时，在改变 RAM 之前保存 BANK 寄存器的值，并且在执行 RETI 指令之前，恢复 BANK 寄存器的值。

备注 在通用区域中分配要求快速响应的中断服务。



- 软件例程 (当使用 16 位定时器/事件计数器 00 的中断请求)

```

VCTBL   CSEG   AT      0020H
         DW     BNKTM000           ; Specifies an address at the timer interrupt destination.

RAMD    DSEG   SADDR
R_BNKRN: DS      1                ; Secures RAM for saving the memory bank number before the interrupt occurs.

-----
BNKC    CSEG   AT      7000H

BNKTM000:
PUSH    AX                        ; Inter-memory bank interrupt servicing routine
                                       ; Saves the contents of the AX register.

        MOV    A,BANK
        MOV    R_BNKRN,A           ; Saves the memory bank number before the interrupt to RAM.
        MOV    BANK,#BANKNUM TEST  ; Specifies the memory bank number of the interrupt routine.
        CALL   !TEST               ; Calls the interrupt routine.
        MOV    A,R_BNKRN          ; Restores the memory bank number before the interrupt.
        MOV    BANK,A

        POP    AX                  ; Restores the contents of the AX register.

        RETI

-----
BN3     CSEG   BANK3
TEST:                                     ; Interrupt servicing routine
        MOV ...
        :
        :
        RET

END
    
```

备注 注意以下几点可高效的使用存储器 **bank** 选择功能。

- 分配通用区域中经常使用的程序。
- 如果计划参照的值放入 **RAM** 中，它可以被所有区域引用。
- 如果参照的目标程序和存储器 **bank** 中跳转的目标程序被放置在同一个存储器 **bank** 中，则编码大小和处理都是更有效率的。
- 在通用区域中分配要求快速响应的中断服务。

第四章 端口功能

4.1 端口功能

有两种类型的引脚 I/O 缓冲电源： AV_{REF} ^注和 V_{DD} 。这些电源和引脚之间的关系显示如下。

表 4-1. 引脚 I/O 缓冲器电源

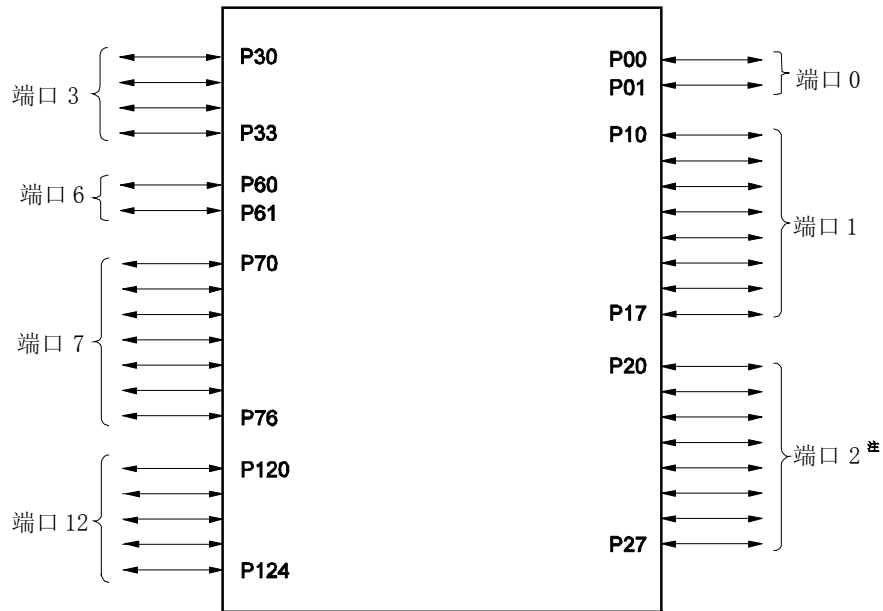
电源	相关引脚
AV_{REF} ^注	P20 ~ P27 ^注
V_{DD}	除 P20 ~ P27 之外的其它引脚

注 仅限于 μ PD78F037x。

78K0/LF2 产品中提供的端口如图 4-1 所示，这些端口可以支持多种控制操作。每个端口的功能如表 4-2 所示。

除了作为数字 I/O 端口功能，这些端口还有几个复用功能。如需了解这些端口复用功能的详细信息，请参见第二章引脚功能。

图 4-1. 端口类型



注 仅限于 μ PD78F037x。

表 4-2. 端口功能

引脚名称	I/O	功能	复位后	复用功能
P00	I/O	端口 0 2 位 I/O 端口 可以位选输入输出模式 通过软件设置, 可以定义内部上拉电阻的使用	输入	TI000
P01				TI010/TO00
P10	I/O	端口 1 8 位 I/O 端口 可以位选输入输出模式 通过软件设置, 可以定义内部上拉电阻的使用	输入	$\overline{\text{SCK10}}/\text{TxD0}$
P11				SI10/RxD0
P12				SO10
P13				TxD6
P14				RxD6
P15				TOH0
P16				TOH1/INTP5
P17				TI50/TO50
P20 ~ P27 ^{注 1}				I/O
P30	I/O	端口 3 4 位 I/O 端口 可以位选输入输出模式 通过软件设置, 可以定义内部上拉电阻的使用	输入	INTP1 ~ INTP3
P31				INTP2/OC1A ^{注 2}
P32				INTP3/OC1B ^{注 2}
P33				INTP4/TI51/TO51
P60	I/O	端口 6 2 位 I/O 端口 N-ch 漏极开路输出(6 V 耐压) 可以位选输入输出模式	输入	SCL0
P61				SDA0
P70 ~ P76	I/O	端口 7 7 位 I/O 端口 可以位选输入输出模式 通过软件设置, 可以定义内部上拉电阻的使用	输入	KR0 ~ KR6
P120	I/O	端口 12 5 位 I/O 端口 可以位选输入输出模式 只有 P120, 可以通过软件设置, 定义内部上拉电阻的使用	输入	INTP0/EXLVI
P121				X1/OC0A ^{注 2}
P122				X2/EXCLK/OC0B ^{注 2}
P123				XT1
P124				XT2/EXCLKS

- 注 1. 仅限于 $\mu\text{PD78F037x}$ 。
 2. 仅限于 $\mu\text{PD78F0376D}$ 和 $78F0386D$ 。

4.2 端口配置

端口包括如下硬件。

表 4-3. 端口配置

项目	配置
控制寄存器	端口模式寄存器 (PM0 ~ PM3, PM6, PM7, PM12) 端口寄存器 (P0 ~ P3, P6, P7, P12) 上拉电阻选择寄存器 (PU0, PU1, PU3, PU7, PU12) A/D 端口配置寄存器 (ADPC)
端口	μ PD78F037x: 34 μ PD78F038x: 26
上拉电阻	22

4.2.1 端口 0

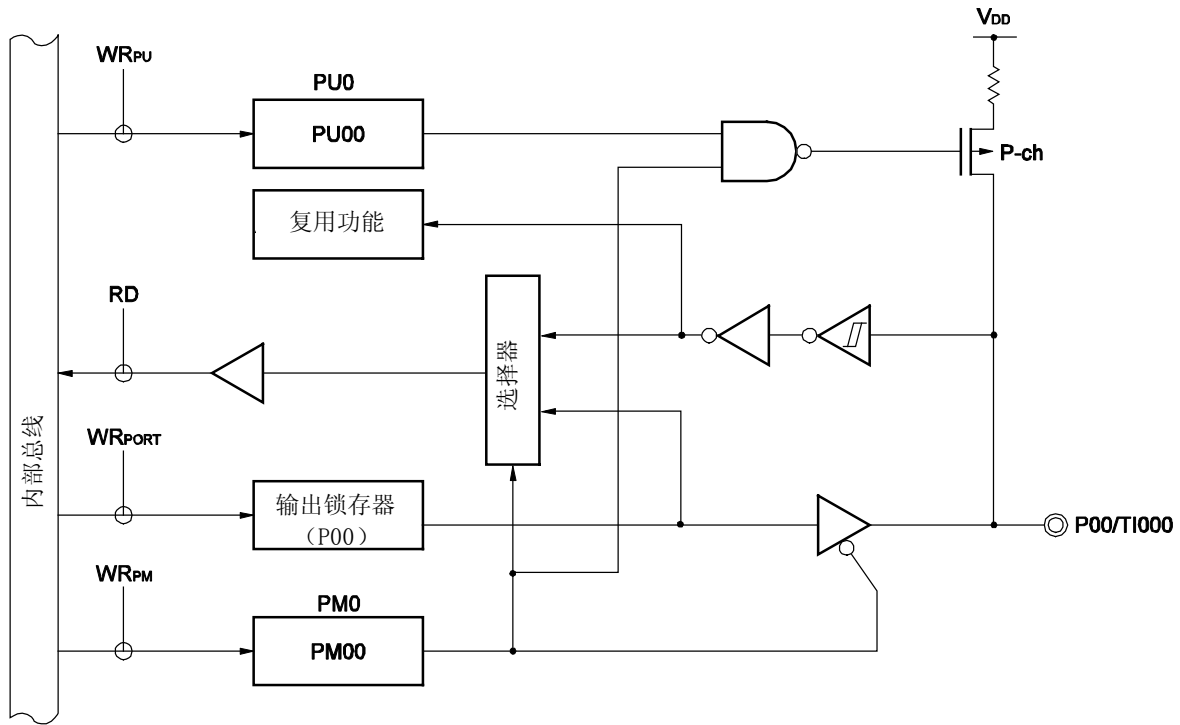
端口 0 是具有输出锁存功能的 2 位 I/O 端口。通过使用端口模式寄存器 0(PM0)，可以位选端口 0 为输入或输出模式。如果 P00 和 P01 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 0(PU0)以 1 位单元的方式指定。

这个端口还可以用于定时器 I/O。

复位信号的产生可将端口 0 设置为输入模式。

图 4-2 和 4-3 显示了端口 0 的框图。

图 4-2. P00 的框图



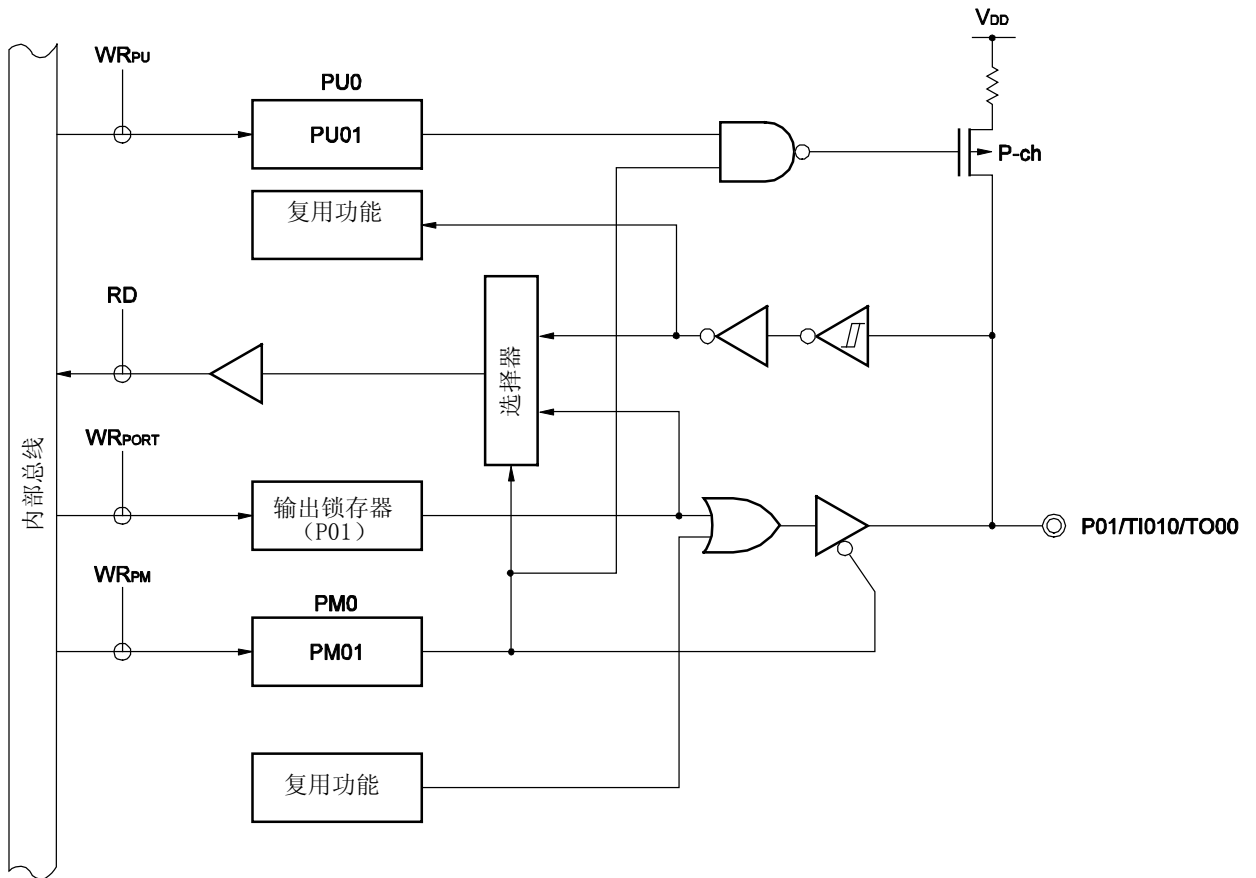
PU0: 上拉电阻选择寄存器 0

PM0: 端口模式寄存器 0

RD: 读信号

WR \times : 写信号

图 4-3. P01 的框图



- PU0: 上拉电阻选择寄存器 0
- PM0: 端口模式寄存器 0
- RD: 读信号
- WR_{xx}: 写信号

4.2.2 端口 1

端口 1 是具有输出锁存功能的 8 位 I/O 端口。通过使用端口模式寄存器 1(PM1)，可以位选端口 1 为输入或输出模式。如果 P10 ~ P17 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 1(PU1)以 1 位单元的方式指定。

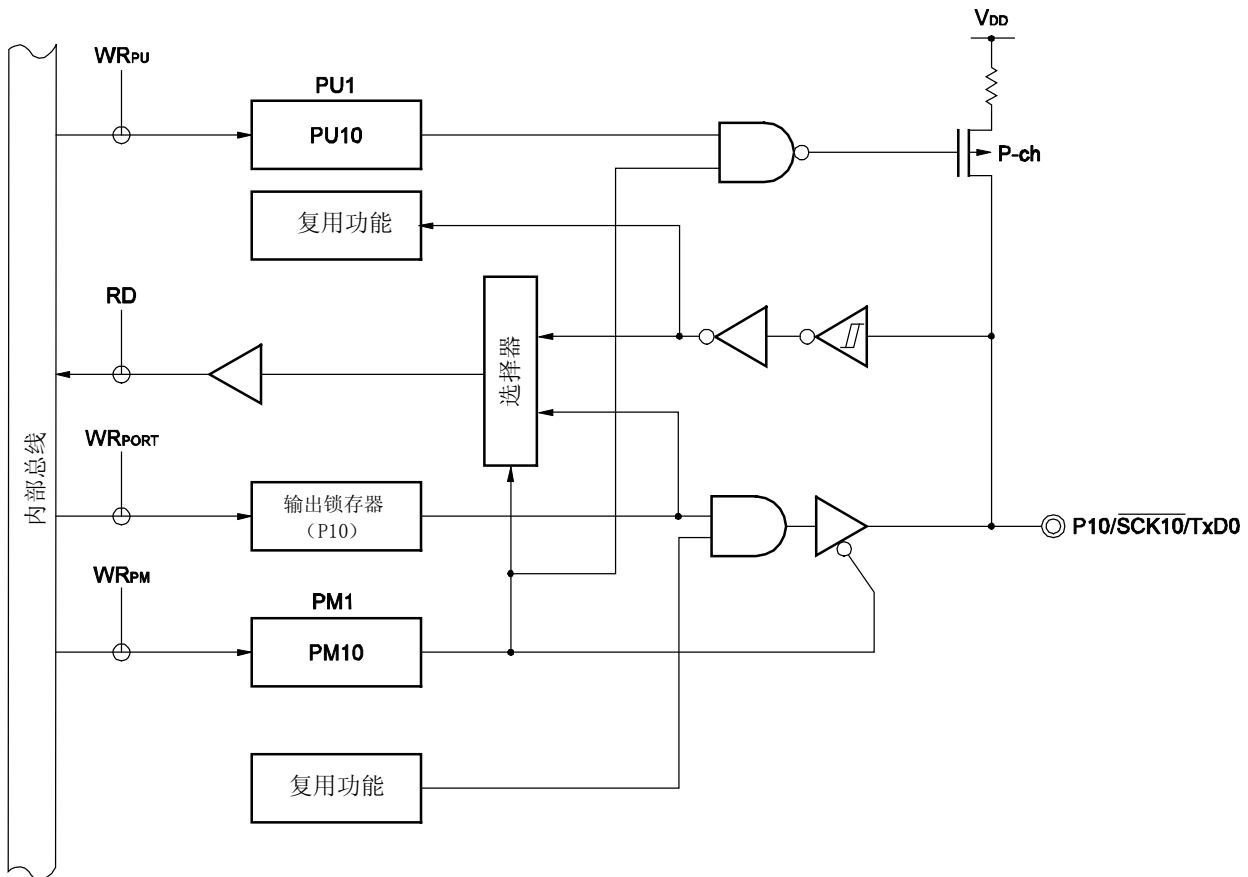
这个端口还可以用于外部中断请求输入、串行接口数据 I/O、时钟 I/O 和定时器 I/O。

复位信号的产生可将端口 1 设置为输入模式。

图 4-4 ~ 4-8 显示了端口 1 的框图。

注意事项 如果 P10/SCK10/TxD0 和 P12/SO10 作为通用端口使用，则将串行操作模式寄存器 10(CSIM10)和串行时钟选择寄存器 10(CSIC10)设置为初始设置状态 (00H)。

图 4-4. P10 的框图



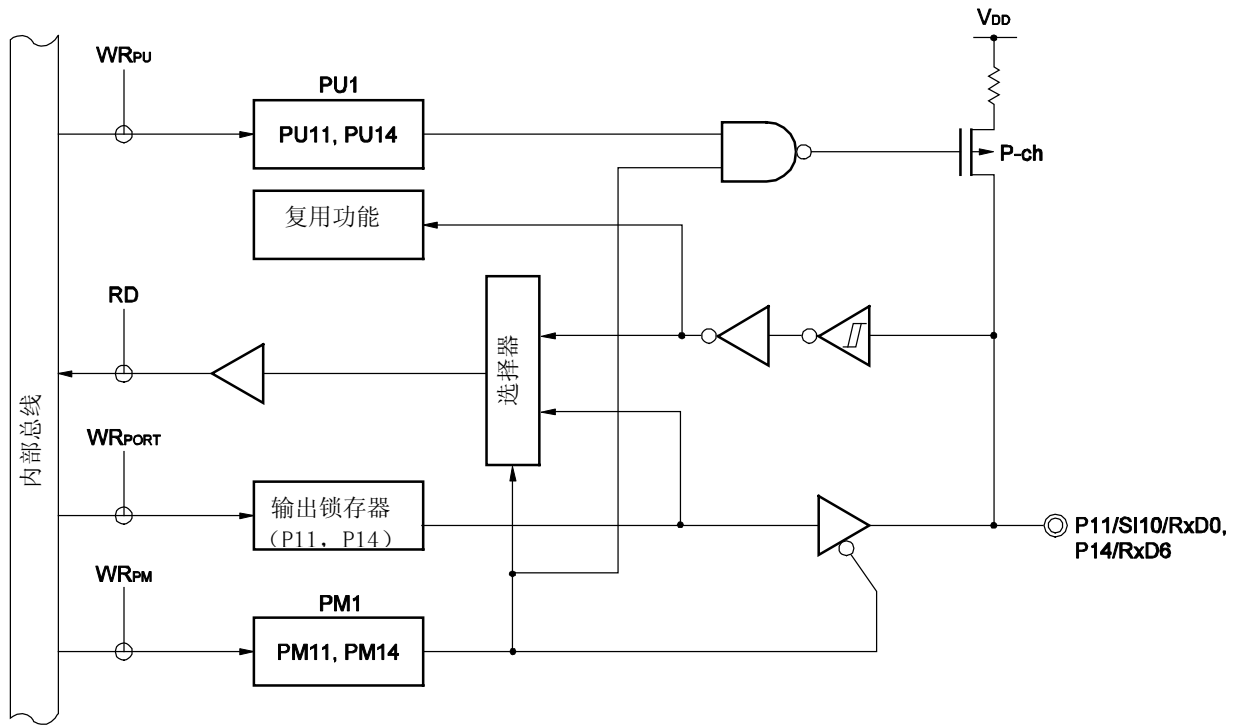
PU1: 上拉电阻选择寄存器 1

PM1: 端口模式寄存器 1

RD: 读信号

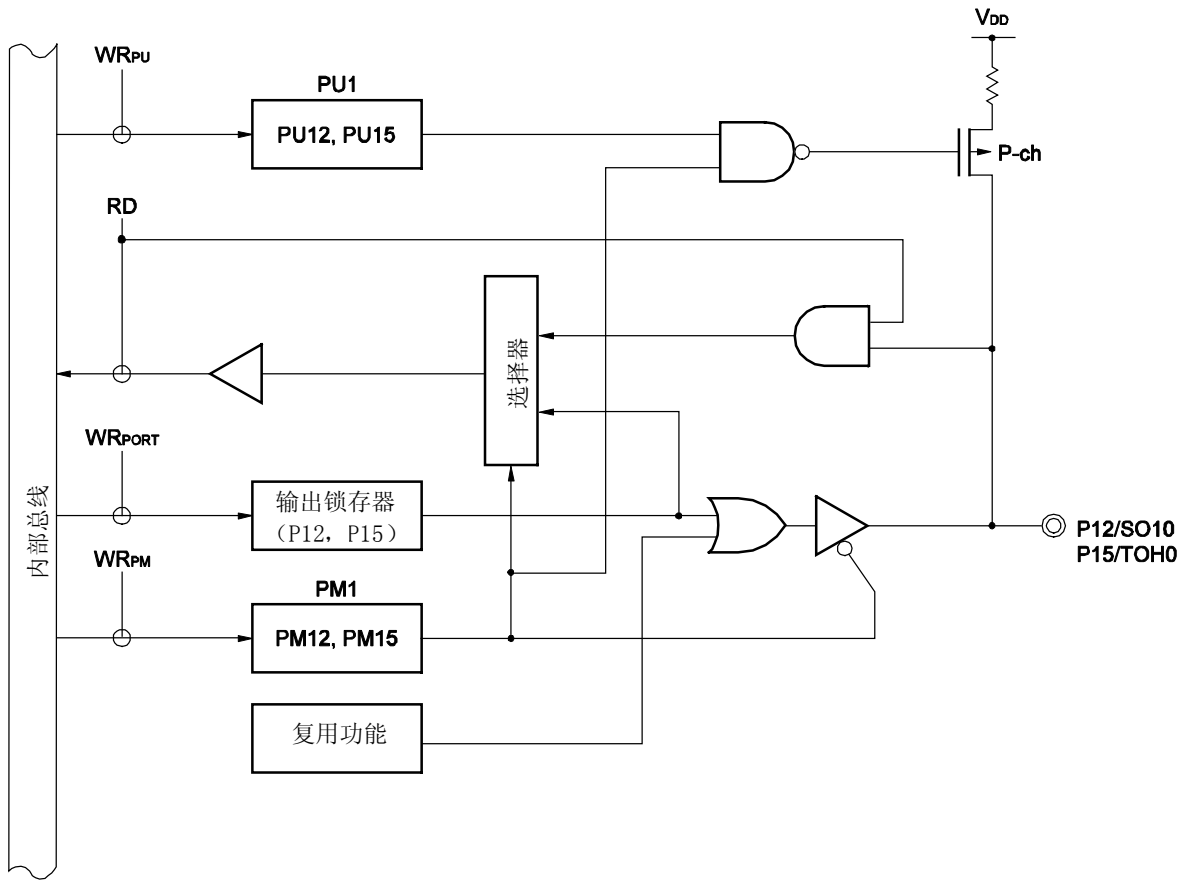
WRxx: 写信号

图 4-5. P11 和 P14 的框图



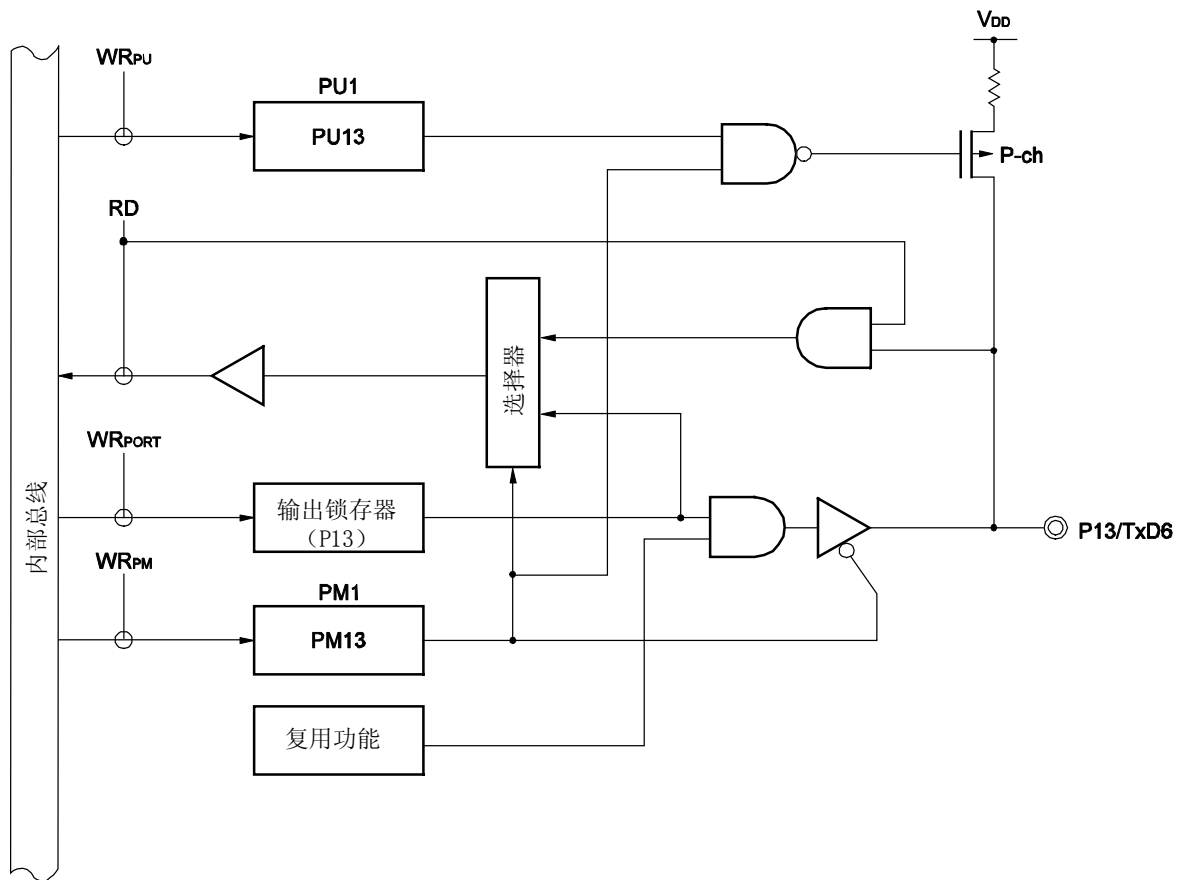
- PU1: 上拉电阻选择寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读信号
- WR_{xx}: 写信号

图 4-6. P12 和 P15 的框图



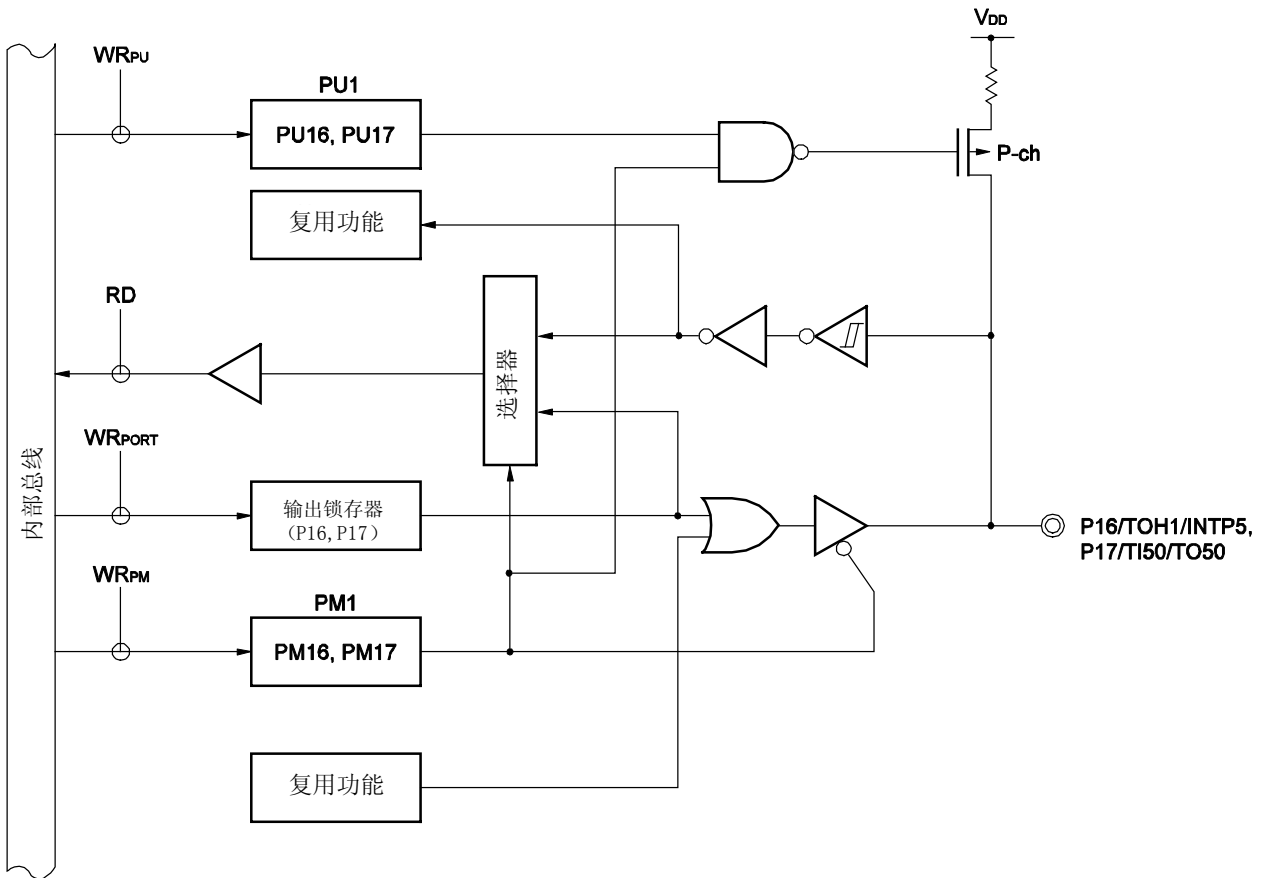
- PU1: 上拉电阻选择寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读信号
- WR_{xx}: 写信号

图 4-7. P13 的框图



- PU1: 上拉电阻选择寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读信号
- WR_{xx}: 写信号

图 4-8. P16 和 P17 的框图



- PU1: 上拉电阻选择寄存器 1
- PM1: 端口模式寄存器 1
- RD: 读信号
- WR_{xx}: 写信号

4.2.3 端口 2 (仅限于 μ PD78F037x)

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)，可以位选端口 2 为输入或输出模式。

这个端口还可以用于 A/D 转换器模拟输入。

如果要将 P20/ANI0 ~ 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 设置为输入模式，从低位开始使用这些引脚。

如果要将 P20/ANI0 ~ P27/ANI7 作为数字输出引脚，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 设置为输出模式。

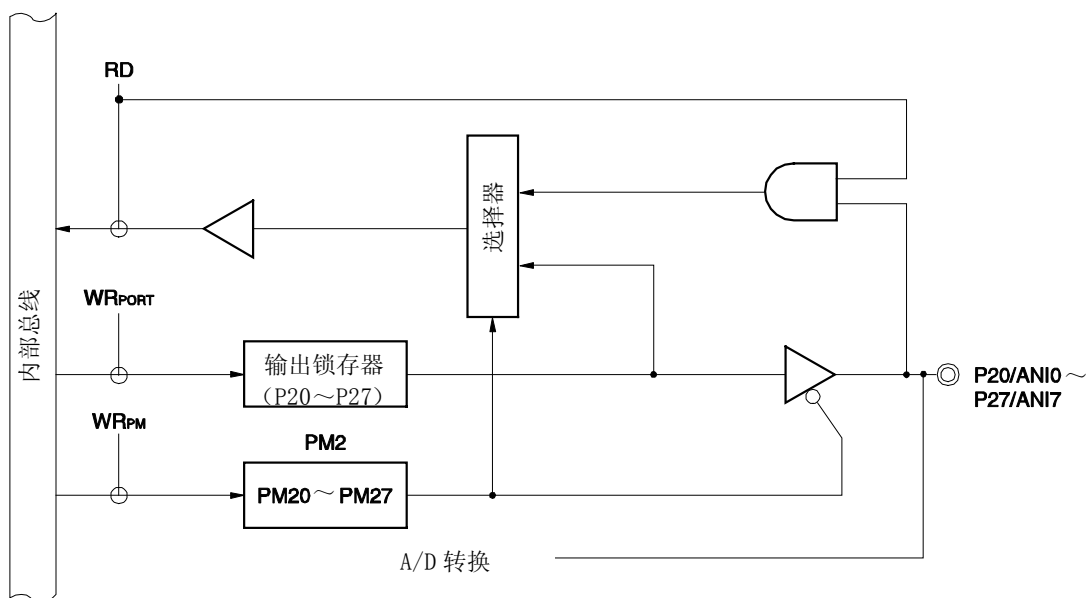
表 4-4. P20/ANI0 ~ P27/ANI7 引脚的功能设置

ADPC 设置	PM2 设置	ADS 设置	P20/ANI0 ~ P27/ANI7 引脚
数字 I/O 选择	输入模式	选择 ANI.	禁止设置
		不选择 ANI.	数字输入
	输出模式	选择 ANI.	禁止设置
		不选择 ANI.	数字输出
模拟输入选择	输入模式	选择 ANI.	模拟输入 (被转换)
		不选择 ANI.	模拟输入 (不被转换)
	输出模式	选择 ANI.	禁止设置
		不选择 ANI.	禁止设置

在复位信号的产生时，P20/ANI0 ~ P27/ANI7 都被设置为模拟输入模式。

图 4-9 显示了端口 2 的框图。

图 4-9. P20 ~ P27 的框图



PM2: 端口模式寄存器 2

RD: 读信号

WR_{xx}: 写信号

4.2.4 端口 3

端口 3 是具有输出锁存功能的 4 位 I/O 端口。通过使用端口模式寄存器 3(PM3)，可以位选端口 3 为输入或输出模式。如果 P30 ~ P33 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 3(PU3)以 1 位单元的方式指定。

这个端口还可以用于外部中断请求输入和定时器 I/O。

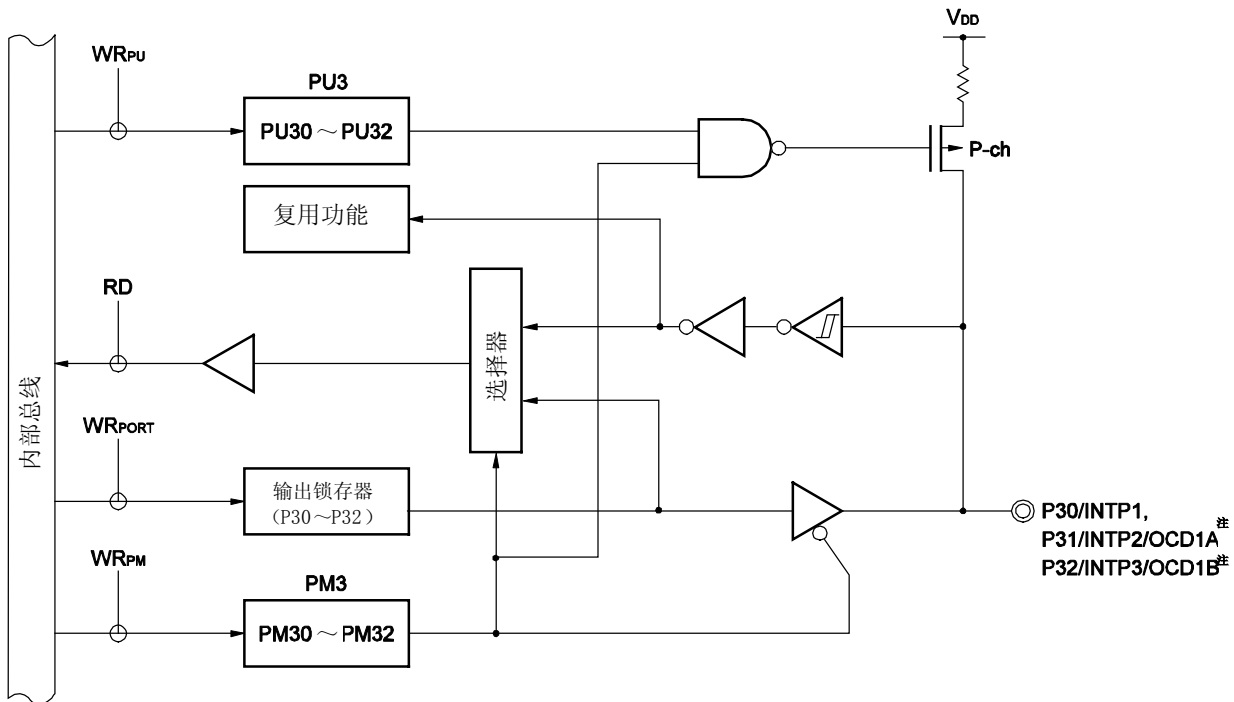
复位信号的产生可将端口 3 设置为输入模式。

图 4-10 和 4-11 显示端口 3 的框图。

注意事项 对于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D, 应确保在复位后将引脚 P31 引脚下拉, 防止故障发生。

备注 在使用片上调试功能时, μ PD78F0376D 和 78F0386D 的 P31 和 P32 引脚能够用于片上调试模式设定引脚 (OCD1A, OCD1B)。详情请参看 第二十七章 片上调试功能 (仅限于 μ PD78F0376D 和 78F0386D)。

图 4-10. P30 ~ P32 的框图



PU3: 上拉电阻选择寄存器 3

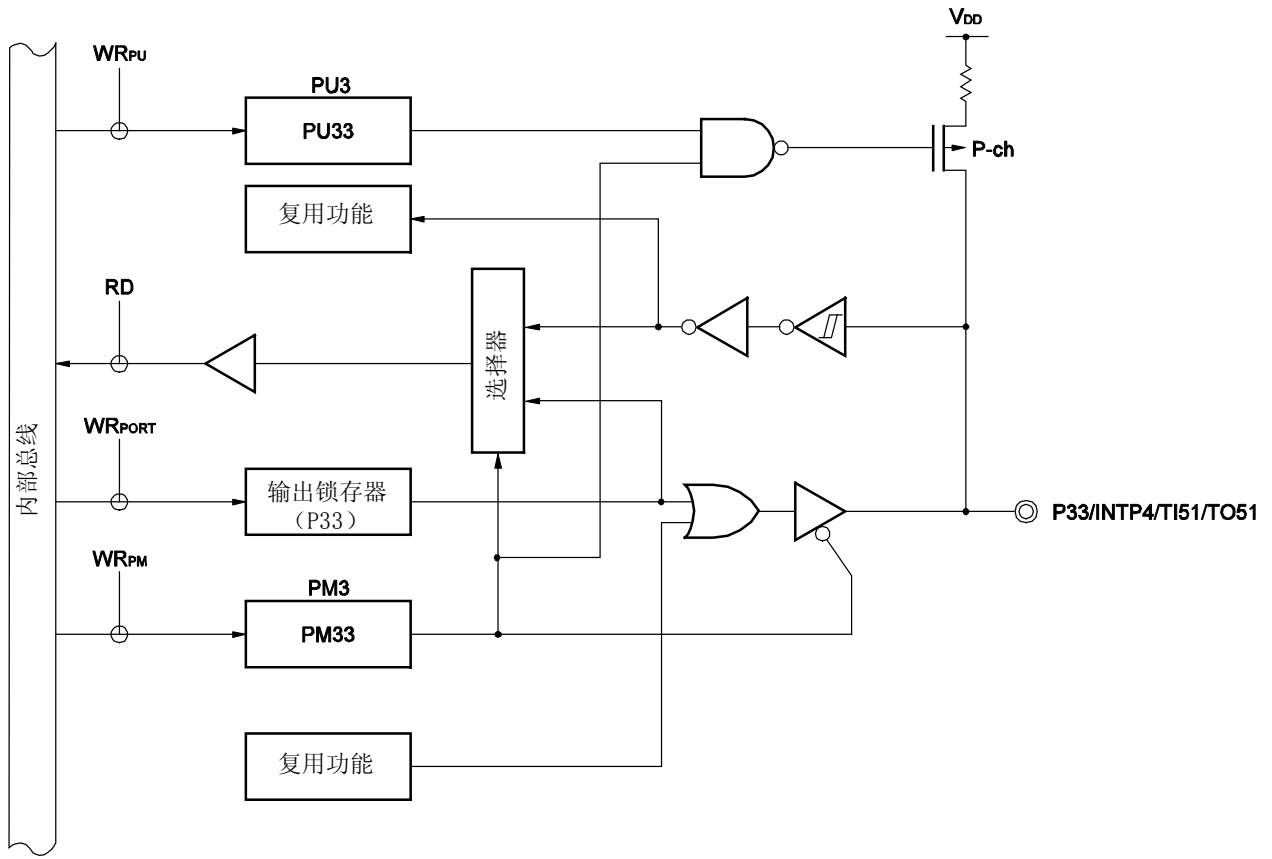
PM3: 端口模式寄存器 3

RD: 读信号

WR_{xx}: 写信号

注 仅限于 μ PD78F0376D 和 78F0386D。

图 4-11. P33 的框图



- PU3: 上拉电阻选择寄存器 3
- PM3: 端口模式寄存器 3
- RD: 读信号
- WR_{xx}: 写信号

<R> 4.2.5 端口 6

端口 6 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 6(PM6)，可以位选端口 6 为输入或输出模式。当 P60 和 P61 引脚被用作输入端口时，使用上拉电阻选择寄存器 6 (PU6) 可位选内置上拉电阻的使用。

P60 和 P61 引脚的输出是 N-ch 漏极开路输出 (6V 耐压)。

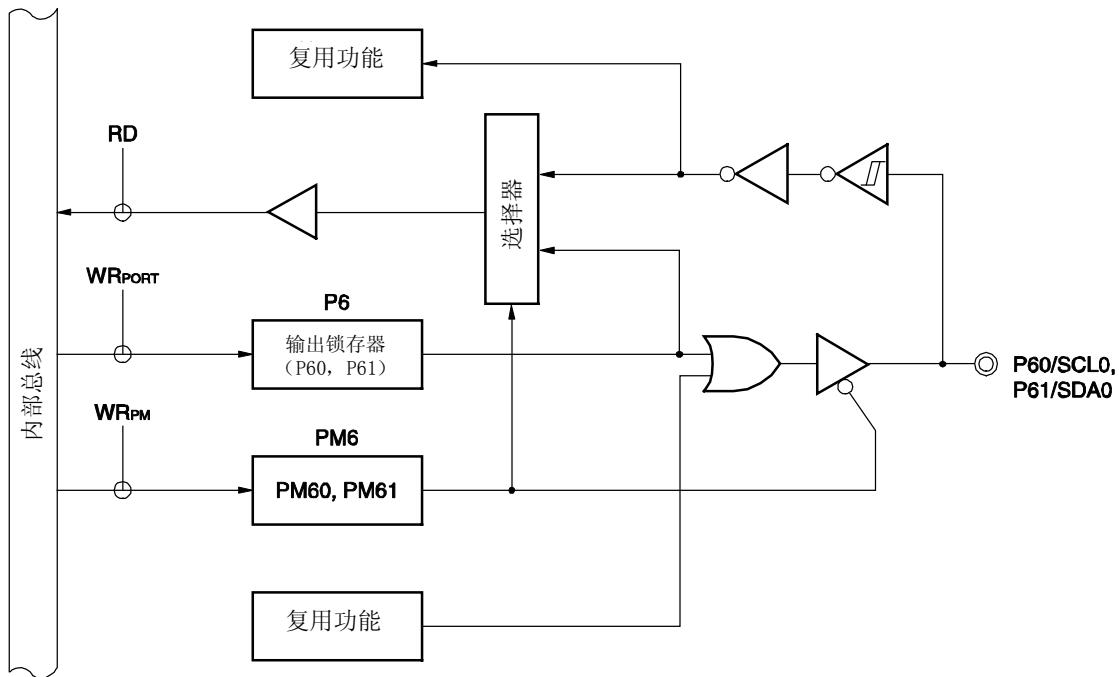
这个端口还可以用作串行接口数据 I/O、时钟 I/O。

复位信号的产生可将端口 6 设置为输入模式。

图 4-12 显示端口 6 的框图。

注意事项 对于 78K0/LF2，可依照规范，分别使用 P60/SCL0 和 P61/SDA0 作为串行时钟 I/O 引脚和串行数据 I/O 引脚。

图 4-12. P60 和 P61 的框图



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

4.2.6 端口 7

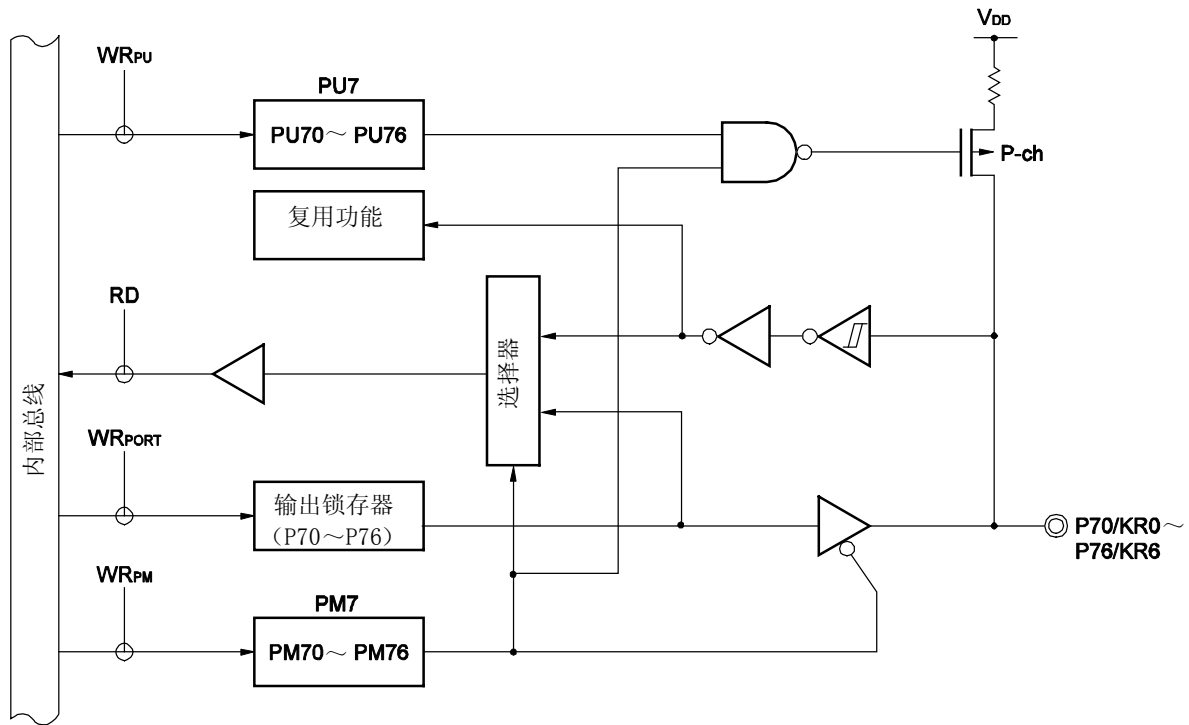
端口 7 是具有输出锁存功能的 7 位 I/O 端口。通过使用端口模式寄存器 7(PM7)，可以位选端口 7 为输入或输出模式。如果 P70 ~ P76 作为输入端口，则内部上拉电阻的使用可以通过上拉电阻选择寄存器 7(PU7)以 1 位单元的方式指定。

这个端口还可以被用作按键返回输入。

复位信号的产生可将端口 7 设置为输入模式。

图 4-13 显示端口 7 的框图。

图 4-13. P70 ~ P76 的框图



PU7: 上拉电阻选择寄存器 7

PM7: 端口模式寄存器 7

RD: 读信号

WR_{xx}: 写信号

4.2.7 端口 12

端口 12 是具有输出锁存功能的 5 位 I/O 端口。通过使用端口模式寄存器 12(PM12)，可以位选端口 12 为输入或输出模式。当 P120 只用作输入端口时，内部上拉电阻的使用可以通过上拉电阻选择寄存器 12(PU12)指定。

这个端口还可以用于外部中断请求输入、外部低电压检测的电压输入、主系统时钟振荡器的连接、外部时钟输入和副时钟振荡器的连接。

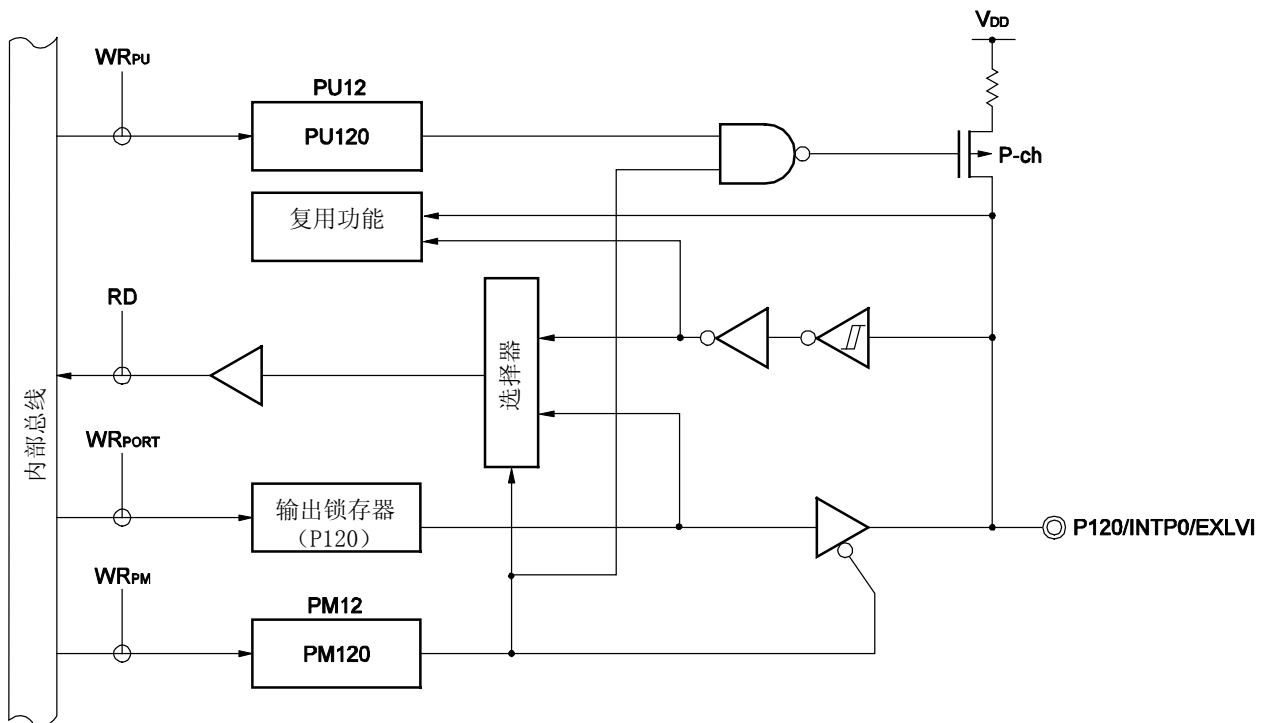
复位信号的产生可将端口 12 设置为输入模式。

图 4-14 和 4-15 显示端口 12 的框图。

注意事项 当使用 P121 ~ P124 引脚连接主系统时钟 (X1,X2) 或副时钟 (XT1,XT2) 的振荡器，或者输入该主系统时钟的外部时钟 (EXCLK) 或副时钟的外部时钟 (EXCLKS) 时，必须通过使用时钟操作模式选择寄存器 (OSCCTL) 对 X1 振荡模式、XT1 振荡模式或者外部时钟输入模式进行设置 (如需了解详细信息，可参见 5.3 (1) 时钟操作模式选择寄存器 (OSCCTL) 和 (3) 副时钟引脚操作模式设置)。OSCCTL 的复位值为 00H (P121 ~ P124 都用作 I/O 端口引脚)。在这种情况下，不必对 PM121 ~ PM124 和 P121 ~ P124 引脚进行设置。

备注 在使用片上调试功能时， μ PD78F0376D 和 78F0386D 的 X1 和 X2 被用于片上调试模式设定引脚 (OCD0A, OCD0B)。详情请参看 第二十七章 片上调试功能 (仅限于 μ PD78F0376D 和 78F0386D)。

图 4-14. P120 的框图



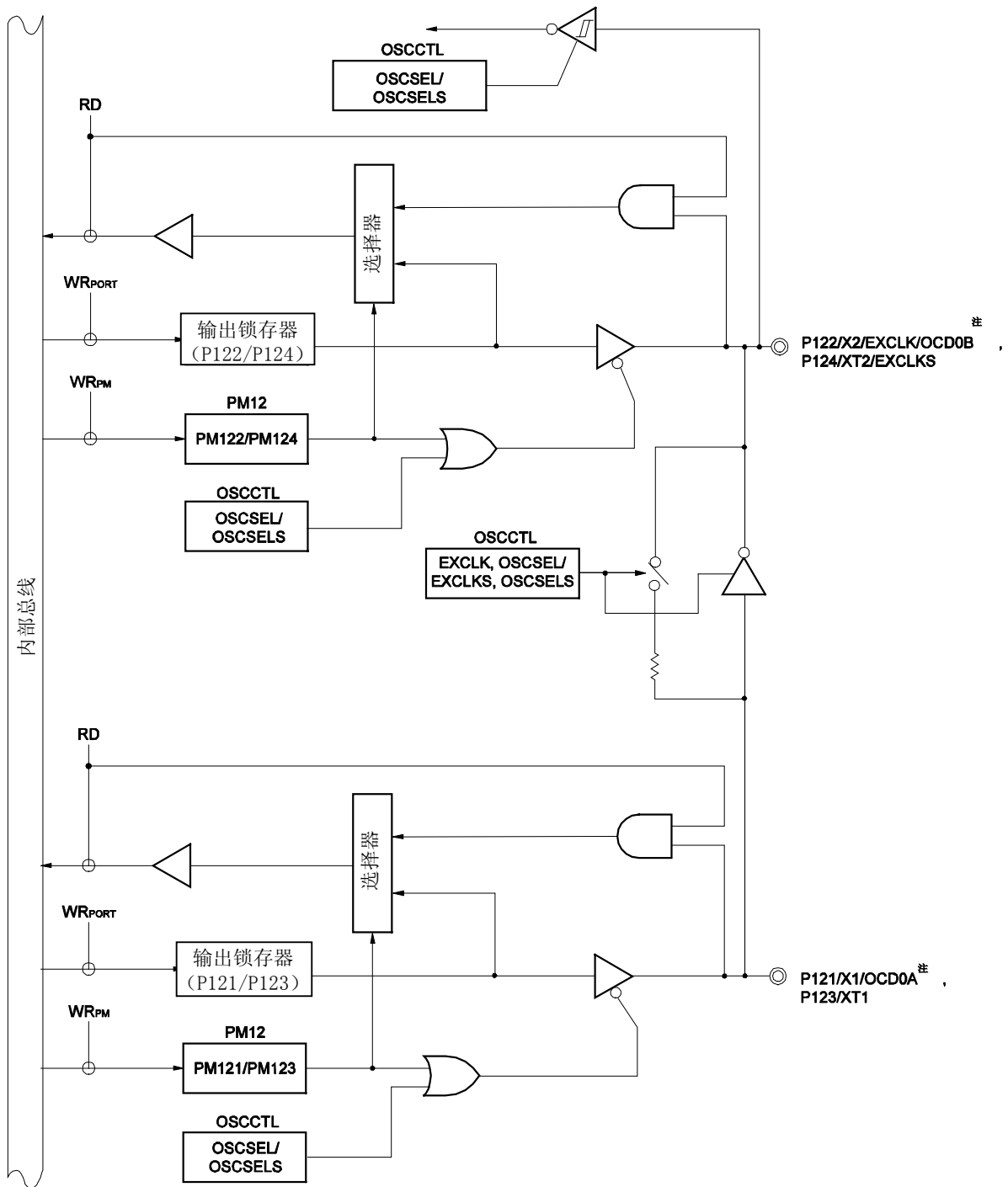
PU12: 上拉电阻选择寄存器 12

PM12: 端口模式寄存器 12

RD: 读信号

WR_{xx}: 写信号

图 4-15. P121 ~ P124 的框图



PU12: 上拉电阻选择寄存器 12

PM12: 端口模式寄存器 12

OSCCTL: 时钟操作模式选择寄存器

RD: 读信号

WR_x: 写信号

注 仅限于 μ PD78F0376D 和 78F0386D。

4.3 控制端口功能的寄存器

端口功能由如下四种类型寄存器控制：

- 端口模式寄存器 (PM0, PM1, PM2^注, PM3, PM6, PM7, PM12)
- 端口寄存器 (P0, P1, P2^注, P3, P6, P7, P12)
- 上拉电阻选择寄存器 (PU0, PU1, PU3, PU7, PU12)
- A/D 端口配置寄存器 (ADPC)

注 仅限于 μ PD78F037x。

(1) 端口模式寄存器 (PM0, PM1, PM2^注, PM3, PM6, PM7, 和 PM12)

这类寄存器以 1 位单元的方式定义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。

复位信号的产生可将这些寄存器的内容设置为 FFH。

当端口引脚使用复用功能时，需要参考表 4.5 使用复用功能时，端口模式寄存器和输出锁存器的设置。

图 4-16. 端口模式寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PM0	1	1	1	1	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
PM2 ^{Note}	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FF22H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
<R> PM6	1	1	1	1	1	PM62	PM61	PM60	FF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FF27H	FFH	R/W
PM12	1	1	1	PM124	PM123	PM122	PM121	PM120	FF2CH	FFH	R/W

PMmn	Pmn 引脚 I/O 模式选择 (m = 0 ~ 3, 6, 7, 12; n = 0 ~ 7)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

注 仅限于 μ PD78F037x。

注意事项 复位释放后，确保设定 PM02、PM03、PM62 和 PM77 为 0。

备注 对于 PM14，参看 11.3 控制时钟输出控制器的寄存器或 17.3 控制 LCD 控制器/驱动器的寄存器

(2) 端口寄存器 (P0, P1, P2^注, P3, P6, P7, P12)

这类寄存器用于设置芯片端口要输出的数据。

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。

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寄存器清零(00H)。

图 4-17. 端口寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
P0	0	0	0	0	0	0	P01	P00	FF00H	00H (输出锁存)	R/W
P1	P17 P16 P15 P14 P13 P12 P11 P10								FF01H	00H (输出锁存)	R/W
P2 ^注	P27 P26 P25 P24 P23 P22 P21 P20								FF02H	00H (输出锁存)	R/W
P3	0	0	0	0	P33	P32	P31	P30	FF03H	00H (输出锁存)	R/W
P6	0	0	0	0	0	0	P61	P60	FF06H	00H (输出锁存)	R/W
P7	0	P76	P75	P74	P73	P72	P71	P70	FF07H	00H (输出锁存)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FF0CH	00H (输出锁存)	R/W

Pmn	m = 0 ~ 3, 6, 7, 12; n = 0 ~ 7	
	输出数据控制 (输出模式下)	输入数据读取 (输入模式下)
0	输出 0	输入低电平
1	输出 1	输入高电平

注 仅限于 μ PD78F037x。

备注 对于 PM13，参看 17.3 控制 LCD 控制器/驱动器的寄存器。

(3) 上拉电阻选择寄存器 (PU0, PU1, PU3, PU7, PU12)

这类寄存器指定是否使用 P00, P01, P10 ~ P17, P30 ~ P33, P70 ~ P76 和 P120 的内部上拉电阻。内部上拉电阻可以以 1 位单元的方式用于那些已设置为输入模式的引脚，并且这些引脚已通过 PU0, PU1, PU3, PU7 和 PU12 指定为使用内部上拉电阻。不论是否设置了 PU0, PU1, PU3, PU7 和 PU12, 设置为输出模式的引脚和用作复用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 4-18. 上拉电阻选择寄存器的格式

符号	7	6	5	4	3	2	1	0	地址	复位后	R/W
PU0	0	0	0	0	0	0	PU01	PU00	FF30H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	0	0	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU7	0	PU76	PU75	PU74	PU73	PU72	PU71	PU70	FF37H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	FF3CH	00H	R/W

PUmn	Pmn 引脚内部上拉电阻选择 (m = 0, 1, 3, 7, 12; n = 0 ~ 7)
0	不连接内部上拉电阻
1	连接内部上拉电阻

(4) A/D 端口配置寄存器 (ADPC) (仅限于 μ PD78F037x)

这个寄存器将 P20/ANI0 ~ P27/ANI7 引脚切换为数字 I/O 端口 或 A/D 转换器的模拟输入端口。

可由 1 位或者 8 位存储器操作指令对 ADPC 进行设置。

复位信号的产生可将寄存器清零(00H)。

图 4-19. A/D 端口配置寄存器 (ADPC) 的格式

地址 : FF2FH 复位后 : 00H RW

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

ADPC3	ADPC2	ADPC1	ADPC0	模拟输入 (A) / 数字输入 (D) 转换							
				P27/ ANI7	P26/ ANI6	P25/ ANI5	P24/ ANI4	P23/ ANI3	P22/ ANI2	P21/ ANI1	P20/ ANI0
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	A	A	A	A	A	A	A	D
0	0	1	0	A	A	A	A	A	A	D	D
0	0	1	1	A	A	A	A	A	D	D	D
0	1	0	0	A	A	A	A	D	D	D	D
0	1	0	1	A	A	A	D	D	D	D	D
0	1	1	0	A	A	D	D	D	D	D	D
0	1	1	1	A	D	D	D	D	D	D	D
1	0	0	0	D	D	D	D	D	D	D	D
除此之外				禁止设置							

- 注意事项**
1. 通过使用端口模式寄存器 2(PM2)将 A/D 转换通道设置为输入模式。
 2. 不要同时使用 ADPC 和 ADS 设置某一引脚为数字 I/O 引脚。
 3. 如果将数据写入 ADPC, 则产生一个等待周期。当 CPU 使用副系统时钟并且外部硬件时钟停止时, 不要将数据写入 ADPC。详细情况可参见 第三十一章 等待注意事项。

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。

注意事项 在使用 1 位操作指令的情况下，尽管只操作了一个信号位，这个端口也要作为 8 位单元来访问。因此，一个可作为输入和输出引脚的端口，指定为输入功能引脚的输出锁存器的内容是不确定的，即使对于未操作的位也一样。

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，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。

4.4.2 读取 I/O 端口

(1) 输出模式

使用传送指令读取的是输出锁存器的内容。输出锁存器的内容不会改变。

(2) 输入模式

使用传送指令读取的是引脚状态。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。

当复位信号的产生时输出锁存器的内容被清零。

(2) 输入模式

读取引脚电平，对它的内容执行操作，操作的结果写入输出锁存器，因为输出缓冲器处于关闭状态，所以引脚状态不会改变。

4.5 使用复用功能时端口模式寄存器和输出锁存器的设置

当使用端口引脚的复用功能时，应按表 4-5 所示对端口模式寄存器和输出锁存器进行设置。

表 4-5. 使用复用功能时，端口模式寄存器和输出锁存器的设置

引脚名称	复用功能		PM _{xx}	P _{xx}
	功能名称	I/O		
P00	TI000	输入	1	×
P01	TI010	输入	1	×
	TO00	输出	0	0
P10	$\overline{\text{SCK10}}$	输入	1	×
		输出	0	1
	TxD0	输出	0	1
P11	SI10	输入	1	×
	RxD0	输入	1	×
P12	SO10	输出	0	0
P13	TxD6	输出	0	1
P14	RxD6	输入	1	×
P15	TOH0	输出	0	0
P16	TOH1	输出	0	0
	INTP5	输入	1	×
P17	TI50	输入	1	×
	TO50	输出	0	0
P20 ~ P27 ^{注 1, 2}	ANI0 ~ ANI7 ^{注 1, 2}	输入	1	×
P30 ~ P32	INTP1 ~ INTP3	输入	1	×
P33	INTP4	输入	1	×
	TI51	输入	1	×
	TO51	输出	0	0
P60	SCL0	I/O	0	0
P61	SDA0	I/O	0	0
P70 ~ P76	KR0 ~ KR6	输入	1	×
P120	INTP0	输入	1	×
	EXLVI	输入	1	×
P121	X1 ^{注 3}	—	×	×
P122	X2 ^{注 3}	—	×	×
	EXCLK ^{注 3}	输入	×	×
P123	XT1 ^{注 3}	—	×	×
P124	XT2 ^{注 3}	—	×	×
	EXCLKS ^{注 3}	输入	×	×

(参考在下页列出注和备注。)

- 注
1. 仅限于 μ PD78F037x。
 2. 通过使用 A/D 端口配置寄存器 (ADPC)、模拟输入通道规格寄存器(ADS)和 PM2, 可以选择 ANI0/P20 ~ ANI7/P27 引脚的功能。

表 4-6. ANI0/P20 ~ ANI7/P27 引脚功能的设置

ADPC 设置	PM2 设置	ADS 设置	ANI0/P20 ~ ANI7/P27 引脚
模拟输入选择	输入模式	选择 ANI.	模拟输入 (转换)
		不选择 ANI.	模拟输入 (不转换)
	输出模式	选择 ANI.	禁止设置
		不选择 ANI.	
数字 I/O 选择	输入模式	选择 ANI.	禁止设置
		不选择 ANI.	数字输入
	输出模式	选择 ANI.	禁止设置
		不选择 ANI.	数字输出

3. 当使用 P121 ~ P124 引脚连接主系统时钟 (X1,X2) 或副时钟 (XT1,XT2) 的振荡器, 或者输入该主系统时钟的外部时钟 (EXCLK) 或副时钟的外部时钟 (EXCLKS) 时, 必须通过使用时钟操作模式选择寄存器 (OSCCTL) 对 X1 振荡模式、XT1 振荡模式或者外部时钟输入模式进行设置 (如需了解详细信息, 可参见 5.3 (1) 时钟操作模式选择寄存器 (OSCCTL) 和 (3) 副时钟引脚操作模式设置)。OSCCTL 的复位值为 00H (P121 ~ P124 都用作 I/O 端口引脚)。在这种情况下, 不必对 PM121 ~ PM124 和 P121 ~ P124 引脚进行设置。

- 备注
1. x: 不必考虑
PMxx: 端口模式寄存器
Pxx: 端口输出锁存器
 2. 在使用片上调试功能时, μ PD78F0376D 和 78F0386D 的 X1、X2、P31 和 P32 能够用于片上调试模式设置引脚 (OCD0A, OCD0B, OCD1A, OCD1B)。详情请参看 第二十七章 片上调试功能 (仅限于 μ PD78F0376D 和 78F0386D)。

5.1 时钟发生器的功能

时钟发生器用于产生时钟提供给 CPU 和外围硬件设备。
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 $f_x = 1 \sim 20\text{MHz}$ 的时钟。
通过执行 STOP 指令或设置主 OSC 控制寄存器(MOC), 可以停止振荡。

<2> 内部高速振荡器

这个振荡电路产生一个 $f_{RH} = 8\text{MHz}$ (TYP.)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 指令或者使用内部振荡模式寄存器(RCM)停止其振荡。

外部主系统时钟($f_{EXCLK} = 1 \sim 20\text{MHz}$)也可以通过 EXCLK/X2/P122 引脚提供。可以通过执行 STOP 指令或者使用 RCM 禁止外部主系统时钟输入。

可以通过使用主时钟模式寄存器(MCM)选择高速系统时钟(X1 时钟或者外部主系统时钟)或内部高速振荡时钟, 作为主系统时钟。

(2) 副系统时钟

• 副系统时钟振荡器

通过在 XT1 和 XT2 之间连接一个 32.768kHz 的振荡器, 该电路以 $f_{XT} = 32.768\text{kHz}$ 的频率进行振荡。通过使用处理器时钟控制寄存器(PCC)和时钟操作模式选择寄存器(OSCCTL), 可以停止振荡。

还可以通过 EXCLKS/XT2/P124 引脚提供一个外部副系统时钟 ($f_{EXCLKS} = 32.768\text{kHz}$)。可以通过设置 PCC 和 OSCCTL 禁止外部副系统时钟输入。

备注	1. f_x :	X1 时钟振荡频率
	2. f_{RH} :	内部高速振荡时钟频率
	3. f_{EXCLK} :	外部主系统时钟频率
	4. f_{XT} :	XT1 时钟振荡频率
	5. f_{EXCLKS} :	外部副系统时钟频率

(3) 内部低速振荡时钟(看门狗定时器时钟)

- 内部低速振荡器

该电路以 $f_{RL} = 240 \text{ kHz}$ (TYP.)的时钟振荡。复位释放后，内部低速振荡时钟总是启动操作。

当通过选项字节设置“内部低速振荡器可由软件停止”时，可以通过使用内部振荡模式寄存器(RCM)停止其振荡。

内部低速振荡时钟不能作为 CPU 时钟。如下硬件使用内部低速振荡时钟。

- 看门狗定时器
- TMH1 (选择 f_{RL} , $f_{RL}/2^7$ 或 $f_{RL}/2^9$ 时)

备注 f_{RL} : 内部低速振荡时钟频率

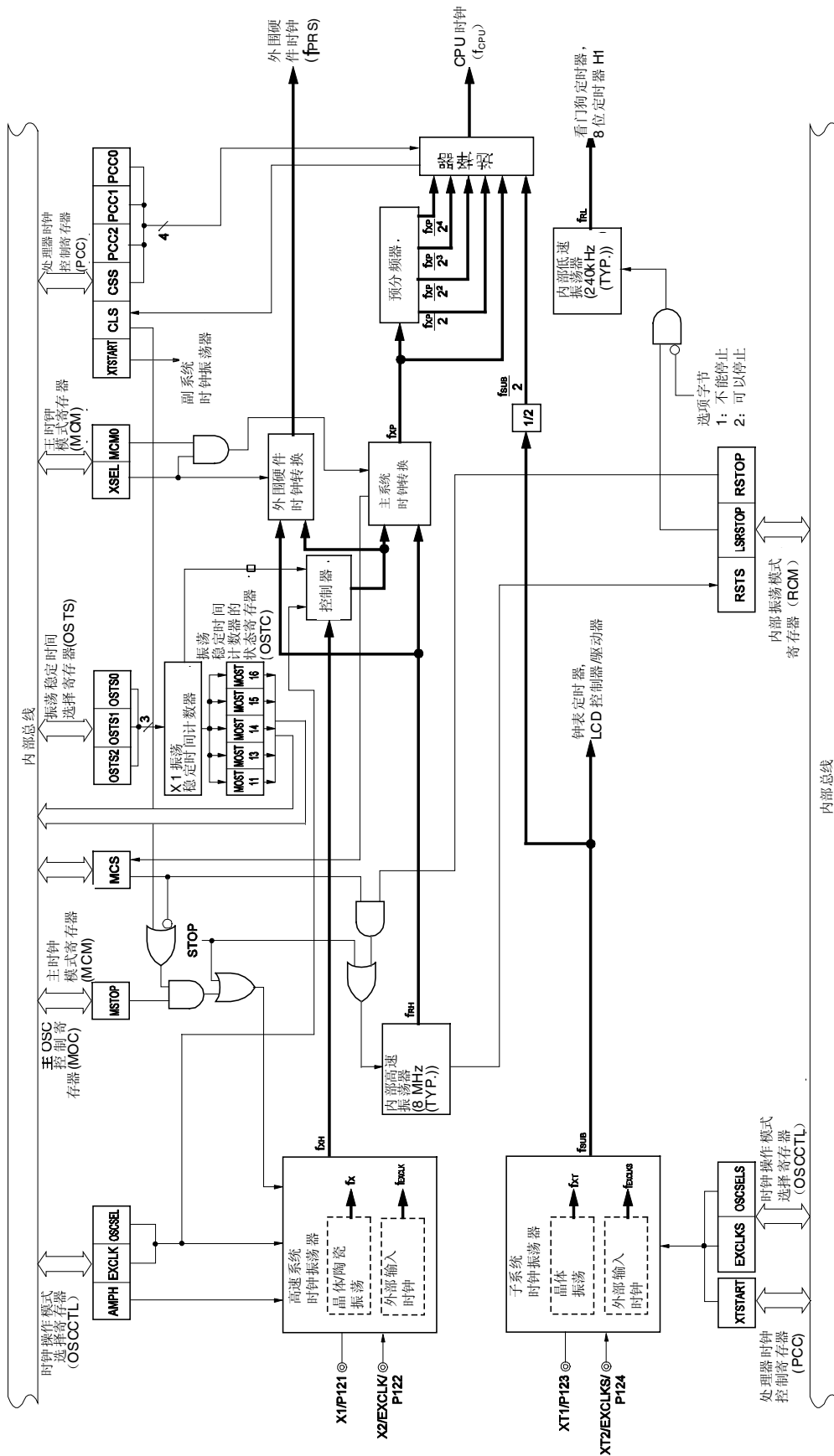
5.2 时钟发生器的配置

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	时钟操作模式选择寄存器(OSCCTL) 处理器时钟控制寄存器(PCC) 内部振荡模式寄存器(RCM) 主 OSC 控制寄存器(MOC) 主时钟模式寄存器(MCM) 振荡稳定时间计数器的状态寄存器(OSTC) 振荡稳定时间选择寄存器(OSTS)
振荡器	X1 振荡器 XT1 振荡器 内部高速振荡器 内部低速振荡器

图 5-1. 时钟发生器的框图



备注	1. f _X :	X1 时钟振荡频率
	2. f _{RH} :	内部高速振荡时钟频率
	3. f _{EXCLK} :	外部主系统时钟频率
	4. f _{XH} :	高速系统时钟振荡频率
	5. f _{XP} :	主系统时钟振荡频率
	6. f _{PRS} :	外围硬件时钟振荡频率
	7. f _{CPU} :	CPU 时钟振荡频率
	8. f _{XT} :	XT1 时钟振荡频率
	9. f _{EXCLKS} :	外部副系统时钟频率
	10. f _{SUB} :	副系统时钟振荡频率
	11. f _{RL} :	内部低速振荡时钟频率

5.3 控制时钟发生器的寄存器

以下七种寄存器用于控制时钟发生器。

- 时钟操作模式选择寄存器 (OSCCTL)
- 处理器时钟控制寄存器 (PCC)
- 内部振荡模式寄存器 (RCM)
- 主 OSC 控制寄存器(MOC)
- 主时钟模式寄存器 (MCM)
- 振荡稳定时间计数器的状态寄存器 (OSTC)
- 振荡稳定时间选择寄存器 (OSTS)

(1) 时钟操作模式选择寄存器 (OSCCTL)

该寄存器用于选择高速系统和副系统时钟的操作模式，以及内置振荡器的获取方式。

可由一个 1 位或 8 位存储器操作指令设置 OSCCTL。

复位信号的产生可将这个寄存器设置为 00H。

图 5-2. 时钟操作模式选择寄存器 (OSCCTL) 的格式

地址: FF9FH 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	3	2	1	<0>
OSCCTL	EXCLK	OSCSEL	EXCLKS [#]	OSCSLS [#]	0	0	0	AMPH
EXCLK	OSCSEL	高速系统时钟引脚操作模式		P121/X1 引脚		P122/X2/EXCLK 引脚		
0	0	I/O 端口模式		I/O 端口				
0	1	X1 振荡模式		晶体/陶瓷振荡器连接				
1	0	I/O 端口模式		I/O 端口				
1	1	外部时钟输入模式		I/O 端口	外部时钟输入			
AMPH	操作频率控制							
0	1 MHz ≤ f _{XH} ≤ 10 MHz							
1	10 MHz < f _{XH} ≤ 20 MHz							

注 EXCLKS 和 OSCSLS 与 XTSTART(处理器时钟控制寄存器 (PCC)的第 6 位)结合使用。
参见 (3) 副系统时钟引脚操作模式的设置。

- 注意事项
1. 如果高速系统时钟振荡频率超过 10MHz, 则必须将 AMPH 设置为 1。
 2. 复位释放后外部功能设置之前应设置 AMPH。复位释放后 AMPH 的值只能被修改一次。在 AMPH 被设置为 1 后, 供应给 CPU 的时钟应停止 5 μs (MIN.)。
 3. 当内部高速振荡时钟或外部主系统时钟被用作 CPU 时钟使用且 AMPH=1 时, 执行 STOP 指令, 则在 STOP 模式释放后, 要提供给 CPU 的时钟需停止 5 μs (MIN.)。如果 X1 时钟用作 CPU 时钟, 振荡稳定时间从 STOP 模式释放后开始计算。
 4. 若要修改 EXCLK 和 OSCSEL, 必须确保主 OSC 控制寄存器(MOC)的第 7 位(MSTOP)=1(X1 振荡器停止或禁止使用来自 EXCLK 引脚的外部时钟)。

备注 f_{XH}: 高速系统时钟振荡频率

(2) 处理器时钟控制寄存器(PCC)

PCC 用于选择 CPU 时钟、分频比和副系统时钟的操作模式。

由 1 位或 8 位存储器操作指令设置 PCC。

复位信号的产生可将 PCC 设置为 01H。

图 5-3. 处理器时钟控制寄存器 (PCC) 的格式

地址: FFFBH 复位后: 01H R/W^{※1}

符号	7	6	<5>	<4>	3	2	1	0
PCC	0	XTSTART ^{※2}	CLS	CSS	0	PCC2	PCC1	PCC0
CLS	CPU 时钟状态							
0	主系统时钟							
1	副系统时钟							
CSS	PCC2	PCC1	PCC0	CPU 时钟 (f _{CPU}) 选择				
0	0	0	0	f _{XP}				
	0	0	1	f _{XP} /2 (默认)				
	0	1	0	f _{XP} /2 ²				
	0	1	1	f _{XP} /2 ³				
	1	0	0	f _{XP} /2 ⁴				
1	0	0	0	f _{SUB} /2				
	0	0	1					
	0	1	0					
	0	1	1					
	1	0	0					
其它				禁止设置				

注 1. 第 5 位只读。

2. XTSTART 与 EXCLKS 和 OSCSELS (时钟操作模式选择寄存器 (OSCCTL) 的第 5 位和第 4 位)结合使用。参见 (3) 副系统时钟引脚操作模式的设置。

注意事项 必须将第 3 位和第 7 位清零。

备注 1. f_{XP}: 主系统时钟振荡频率
2. f_{SUB}: 副系统时钟振荡频率

在 78K0/LF2 中执行速度最快的指令执行时间在 2 个 CPU 时钟以内。因此, CPU 时钟(f_{CPU})与指令最短执行时间的关系如表 5-2 所示。

表 5-2. CPU 时钟与指令最短执行时间的关系

CPU 时钟(f_{CPU})	指令最短执行时间: $2/f_{CPU}$			
	主系统时钟			副系统时钟
	高速系统时钟 ^注		内部高速振荡时钟 ^注	
	操作频率: 10 MHz	操作频率: 20 MHz	操作频率: 8 MHz (TYP.)	操作频率: 32.768 kHz
f_{XP}	0.2 μ s	0.1 μ s	0.25 μ s (TYP.)	–
$f_{XP}/2$	0.4 μ s	0.2 μ s	0.5 μ s (TYP.)	–
$f_{XP}/2^2$	0.8 μ s	0.4 μ s	1.0 μ s (TYP.)	–
$f_{XP}/2^3$	1.6 μ s	0.8 μ s	2.0 μ s (TYP.)	–
$f_{XP}/2^4$	3.2 μ s	1.6 μ s	4.0 μ s (TYP.)	–
$f_{SUB}/2$	–		–	122.1 μ s

注 主时钟模式寄存器 (MCM) 用于设置提供给 CPU 的主系统时钟 (高速系统时钟/内部高速振荡时钟) (参见图 5-6)。

(3) 副系统时钟引脚操作模式的设置

使用处理器时钟控制寄存器 (PCC) 的第 6 位 (XTSTART) 和时钟操作模式选择寄存器 (OSCCTL) 的第 5 位和第 4 位 (EXCLKS, OSCSELS) 共同设置副系统时钟引脚操作模式。

表 5-3. 副系统时钟引脚操作模式的设置

PCC	OSCCTL		副系统时钟引脚操作模式	P123/XT1 引脚	P124/XT2/EXCLKS 引脚
	第 5 位	第 4 位			
XTSTART	EXCLKS	OSCSELS			
0	0	0	I/O 端口模式	I/O 端口	
0	0	1	XT1 振荡模式	晶体振荡器连接	
0	1	0	I/O 端口模式	I/O 端口	
0	1	1	外部时钟输入模式	I/O 端口	外部时钟输入
1	x	x	XT1 振荡模式	晶体振荡器连接	

注意事项 改变 XTSTART、EXCLKS 和 OSCSELS 的当前值时，应确保处理器时钟控制寄存器 (PCC) 的第 5 位 (CLS) = 0 (CPU 使用主系统时钟)。

备注 x: 不必考虑

(4) 内部振荡模式寄存器 (RCM)

内部振荡模式寄存器用于设置内部振荡器操作模式。

由 1 位或 8 位存储器操作指令设置 RCM。

复位信号的产生可将该寄存器设置为 80H^{※1}。

图 5-4. 内部振荡模式寄存器 (RCM) 的格式

地址: FFA0H 复位后: 80H^{※1} R/W^{※2}

符号	<7>	6	5	4	3	2	<1>	<0>
RCM	RSTS	0	0	0	0	0	LSRSTOP	RSTOP

RSTS	内部高速振荡器状态
0	等待内部高速振荡器精确稳定
1	内部高速振荡器的稳定操作

LSRSTOP	内部低速振荡器振荡/停止
0	内部低速振荡器振荡
1	内部低速振荡器停止

RSTOP	内部高速振荡器振荡/停止
0	内部高速振荡器振荡
1	内部高速振荡器停止

- 注
1. 复位释放后寄存器的值立即变为 00H，但在内部高速振荡器稳定后又自动变为 80H。
 2. 第 7 位为只读。

注意事项 当设置 RSTOP = 1 时，必须确保 CPU 使用的不是内部高速振荡时钟。特别地，在下列任一条件下，可以设置 RSTOP = 1。

- 当 MCS = 1 (当 CPU 使用高速系统时钟时)
- 当 CLS = 1 (当 CPU 使用副系统时钟时)

此外，在将 RSTOP 设置为 1 之前应停止正在使用内部高速振荡时钟操作的外围硬件。

(5) 主 OSC 控制寄存器 (MOC)

MOC 用于选择高速系统时钟的操作模式。

当 CPU 不使用高速系统时钟时，该寄存器用于停止 X1 振荡器或禁止 EXCLK 引脚的外部时钟输入。

可以用 1 位或 8 位存储器操作指令设置 MOC。

复位信号的产生可将该寄存器设置为 80H。

图 5-5. 主 OSC 控制寄存器 (MOC) 的格式

地址: FFA2H 复位后: 80H R/W

符号	<7>	6	5	4	3	2	1	0
MOC	MSTOP	0	0	0	0	0	0	0

MSTOP	高速系统时钟操作控制	
	X1 振荡模式	外部时钟输入模式
0	X1 振荡器操作	允许使用来自 EXCLK 引脚的外部时钟
1	X1 振荡器停止	禁止使用来自 EXCLK 引脚的外部时钟

- 注意事项**
1. 设置 **MSTOP = 1** 时，必须确保 CPU 使用的不是高速系统时钟。特别地，在下列任一条件下，可以设置 **MSTOP = 1**。
 - 当 **MCS = 0** (当 CPU 使用内部高速振荡时钟时)
 - 当 **CLS = 1** (当 CPU 使用副系统时钟时)
 此外，在设置 **MSTOP** 为 1 之前应停止正在使用高速系统时钟操作的外围硬件。
 2. 当时钟操作模式选择寄存器 (**OSCCTL**) 的第 6 位 (**OSCSEL**)= 0 (I/O 端口模式) 时，不要将 **MSTOP** 清零。
 3. 外围硬件时钟停止时外围硬件不能操作。外围硬件时钟停止后，若要恢复外围硬件的操作，则必须初始化外围硬件。

(6) 主时钟模式寄存器 (MCM)

MCM 用于选择提供给 CPU 的主系统时钟时钟和外围硬件的时钟。

由 1 位或 8 位存储器操作指令设置 MCM。

复位信号的产生可对寄存器清零(00H)。

图 5-6. 主时钟模式寄存器 (MCM) 的格式

地址: FFA1H 复位后: 00H R/W[≠]

符号	7	6	5	4	3	<2>	<1>	<0>
MCM	0	0	0	0	0	XSEL	MCS	MCM0

XSEL	MCM0	主系统和外围硬件的时钟选择	
		主系统时钟 (f _{XP})	外围硬件时钟 (f _{PRS})
0	0	内部高速振荡时钟 (f _{RH})	内部高速振荡时钟 (f _{RH})
0	1		高速系统时钟 (f _{XH})
1	0	高速系统时钟 (f _{XH})	
1	1		高速系统时钟 (f _{XH})

MCS	主系统时钟状态
0	使用内部高速振荡时钟
1	使用高速系统时钟

注 第 1 位只读。

注意事项 1.XSEL 在复位释放后只能被修改一次。

2.无论 XSEL 和 MCM0 如何设置, 除了 f_{PRS} 以外的一个时钟将被提供给如下外围硬件功能

- 看门狗定时器 (使用内部低速振荡时钟)
- 选择“f_{RL}”, “f_{RL}/2⁷”, 或者 “f_{RL}/2⁹”作为 8 位定时器 H1 的计数时钟 (使用内部低速振荡时钟)
- 选择外部时钟作为外围硬件的时钟源
(以下情况除外: 选择 TM0n (n = 0, 1)的外部计数时钟 (TI00n 有效沿) 时)

(7) 振荡稳定时间计数器的状态寄存器 (OSTC)

该寄存器用于指示 X1 时钟振荡稳定时间计数器的计数状态。当 X1 时钟振荡启动时，使用内部高速振荡时钟或副系统时钟作为 CPU 时钟，X1 时钟振荡稳定时间可以被检测。

可以由 1 位或 8 位存储器操作指令读取 OSTC 的内容。

复位释放后(由 $\overline{\text{RESET}}$ 输入、POC、LVI 和 WDT 进行复位)，通过 STOP 指令，以及设置 MSTOP (MOC 寄存器的第 7 位) = 1，可以对 OSTC 清零(00H)。

图 5-7. 振荡稳定时间计数器的状态寄存器 (OSTC) 的格式

地址: FFA3H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	振荡稳定时间的状态	
					fx = 10 MHz	fx = 20 MHz
1	0	0	0	0	$2^{11}/f_x \text{ min.}$	204.8 $\mu\text{s min.}$
1	1	0	0	0	$2^{13}/f_x \text{ min.}$	819.2 $\mu\text{s min.}$
1	1	1	0	0	$2^{14}/f_x \text{ min.}$	1.64 ms min.
1	1	1	1	0	$2^{15}/f_x \text{ min.}$	3.27 ms min.
1	1	1	1	1	$2^{16}/f_x \text{ min.}$	6.55 ms min.

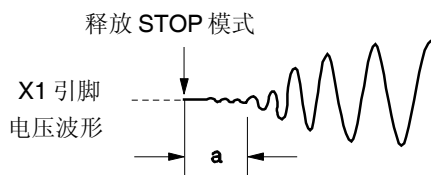
注意事项 1. 在经过上述稳定时间后，从 MOST11 开始的位依次被设置为 1，并一直保持。

2. 振荡稳定时间计数器值计数到由 OSTC 设置的振荡稳定时间。当内部高速振荡时钟作为 CPU 时钟，如果已进入 STOP 模式，在释放该模式时，按如下方式设置振荡稳定时间。

- 预期的 OSTC 振荡稳定时间 \leq 由 OSTC 设置的振荡稳定时间。

注意，在 STOP 模式释放后，仅将达到振荡稳定时间 (由 OSTC 设置) 的状态赋给 OSTC。

3. X1 时钟振荡稳定等待时间不包括时钟振荡开始之前的时间(下图“a”表示的部分)。



备注 fx: X1 时钟振荡频率

(8) 振荡稳定时间选择寄存器 (OSTS)

OSTS 用于选择 X1 时钟振荡稳定等待时间(当 STOP 模式释放时)。

当 CPU 使用 X1 时钟且 STOP 模式释放后, 等待由 OSTS 设置的时间。

当 CPU 使用内部高速振荡时钟时, 应使用 OSTC 确认 STOP 模式释放后已经经历了所要求的振荡稳定时间。可以使用 OSTC 检测振荡稳定时间。

可以由 8 位存储器操作指令设置 OSTS。

复位信号的产生将 OSTS 内容设置为 05H。

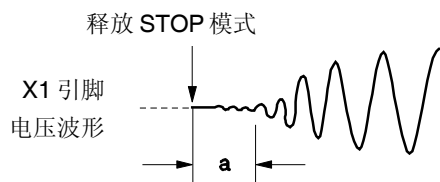
图 5-8. 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFA4H 复位后: 05H R/W

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

OSTS2	OSTS1	OSTS0		振荡稳定时间的选择	
				$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$
0	0	1	$2^{11}/f_x$	204.8 μs	102.4 μs
0	1	0	$2^{13}/f_x$	819.2 μs	409.6 μs
0	1	1	$2^{14}/f_x$	1.64 ms	819.2 μs
1	0	0	$2^{15}/f_x$	3.27 ms	1.64 ms
1	0	1	$2^{16}/f_x$	6.55 ms	3.27 ms
其它			禁止设置		

- 注意事项**
1. 当 CPU 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. 在 X1 时钟振荡稳定时间内不要改变 OSTS 的值。
 3. 振荡稳定时间计数器值增加到由 OSTS 设置的振荡稳定时间。CPU 使用内部高速振荡时钟时, 如果已进入 STOP 模式, 在释放该模式时, 按如下方式设置振荡稳定时间。
 - 预期的 OSTC 振荡稳定时间 \leq 由 OSTS 设置的振荡稳定时间。
 注意, 在 STOP 模式释放后, 仅将达到振荡稳定时间的状态(由 OSTS 设置)赋给 OSTC。
 4. X1 时钟振荡稳定等待时间不包括时钟振荡开始之前的时间(下图“a”表示的部分)。



备注 f_x : X1 时钟振荡频率

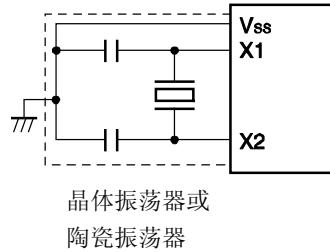
5.4 系统时钟振荡器

5.4.1 X1 振荡器

X1 振荡器采用晶体振荡器或陶瓷振荡器(1 ~ 20MHz)，连接到 X1 和 X2 引脚。

图 5-9 为 X1 振荡器的外部电路示例。

图 5-9. X1 振荡器的外部电路示例 (晶体或陶瓷振荡器)



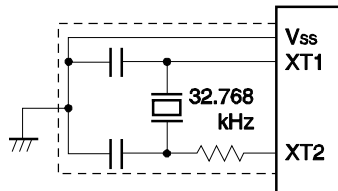
使用时的注意事项在下页列出。

5.4.2 XT1 振荡器

XT1 振荡器采用晶体振荡器(标准值为：32.768 kHz)，连接到 XT1 和 XT2 引脚。

图 5-10 为 XT1 振荡器的外部电路示例。

图 5-10. XT1 振荡器的外部电路示例 (晶体振荡器)



使用时的注意事项在下页列出。

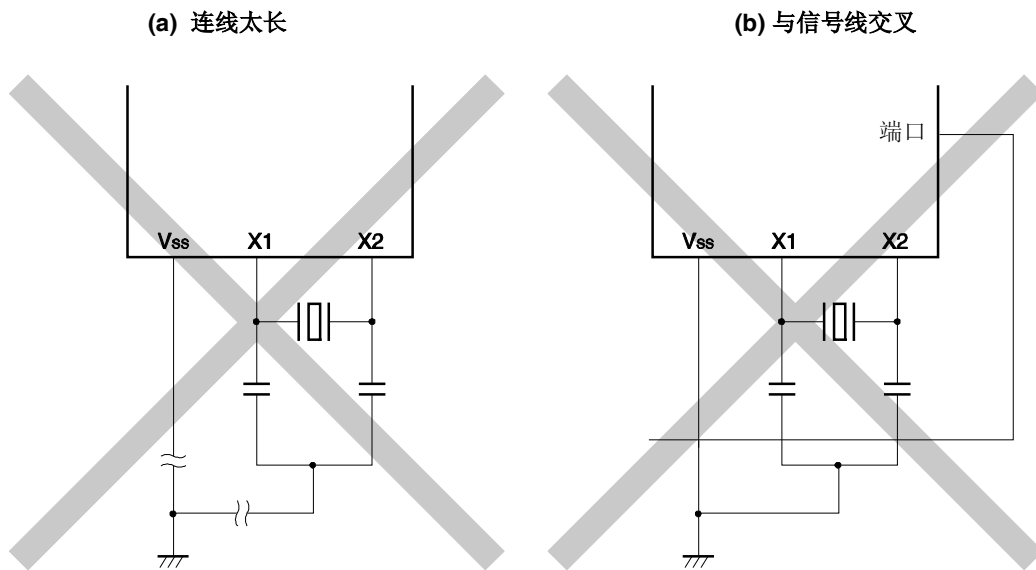
注意事项 1. 在使用 X1 振荡器和 XT1 振荡器时，图 5-9 和图 5-10 中被虚线包围的部分的配线应按照如下配线方法配线，以防止连接线电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。流经的电流变化较大的信号线不要在振荡器周围布线。
- 要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 5-11 为不正确的振荡器连接示例。

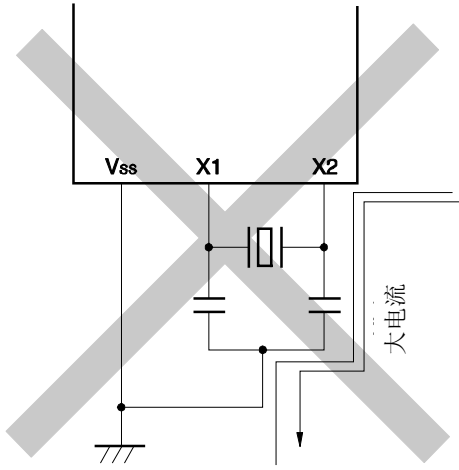
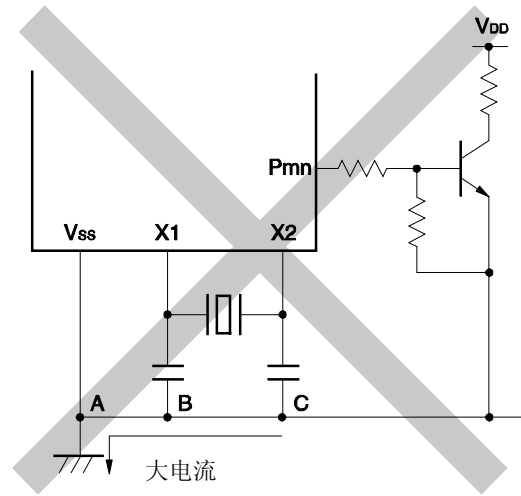
图 5-11. 不正确的振荡器连接示例 (1/2)



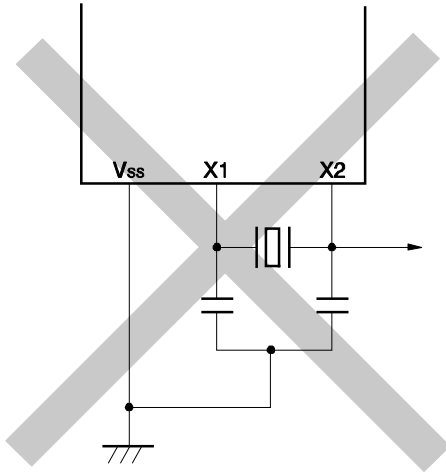
备注 在使用副系统时钟时，分别用 XT1 和 XT2 代替 X1 和 X2。串联电阻也插在 XT2 这边。

图 5-11. 不正确的振荡器连接示例 (2/2)

(c) 连接线周围有高变化电流

(d) 大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。串联电阻也插在 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，从而产生错误。

5.4.3 不使用副系统时钟

如果无需使用副系统时钟进行低功耗操作，或者不使用副系统时钟作为 I/O 端口，则可以设置 XT1 和 XT2 引脚为 I/O 模式(OSCSELS = 0)，并按下列方式进行连接。

输入 (PM123/PM124 = 1): 通过一个电阻独立连接到 V_{DD} 或 V_{SS}。

输出 (PM123/PM124 = 0): 保持开路。

备注 OSCSELS: 时钟操作模式选择寄存器的第 4 位 (OSCCTL)

PM123, PM124: 端口模式寄存器 12 的第 3 位和第 4 位 (PM12)

5.4.4 内部高速振荡器

78K0/LF2 产品中包含内部高速振荡器。可以通过内部振荡模式寄存器(RCM)控制振荡。

复位释放后，内部高速振荡器自动开始振荡(8 MHz (TYP.))。

5.4.5 内部低速振荡器

78K0/LF2 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和 8 位定时器 H1 的时钟使用。内部低速振荡时钟不能用作 CPU 时钟。

可以通过选项字节选择内部低速振荡器“可由软件停止”或“不能停止”。如果设置“可由软件停止”，则可由内部振荡模式寄存器(RCM)控制振荡。

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 (240 kHz (TYP.))。

5.4.6 预分频器

当 CPU 使用主系统时钟时，通过分频主系统时钟，预分频器可以产生多种时钟。

5.5 时钟发生器的操作

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

- 主系统时钟 f_{XP}
 - 高速系统时钟 f_{XH}
 - X1 时钟 f_X
 - 外部主系统时钟 f_{EXCLK}
 - 内部高速振荡时钟 f_{RH}
- 副系统时钟 f_{SUB}
 - XT1 时钟 f_{XT}
 - 外部副系统时钟 f_{EXCLKS}
- 内部低速振荡时钟 f_{RL}
- CPU 时钟 f_{CPU}
- 外围硬件时钟 f_{PRS}

在 78K0/LF2 中，当复位释放后内部高速振荡器输出时，CPU 开始操作，因此具有以下特点。

(1) 增强安全功能

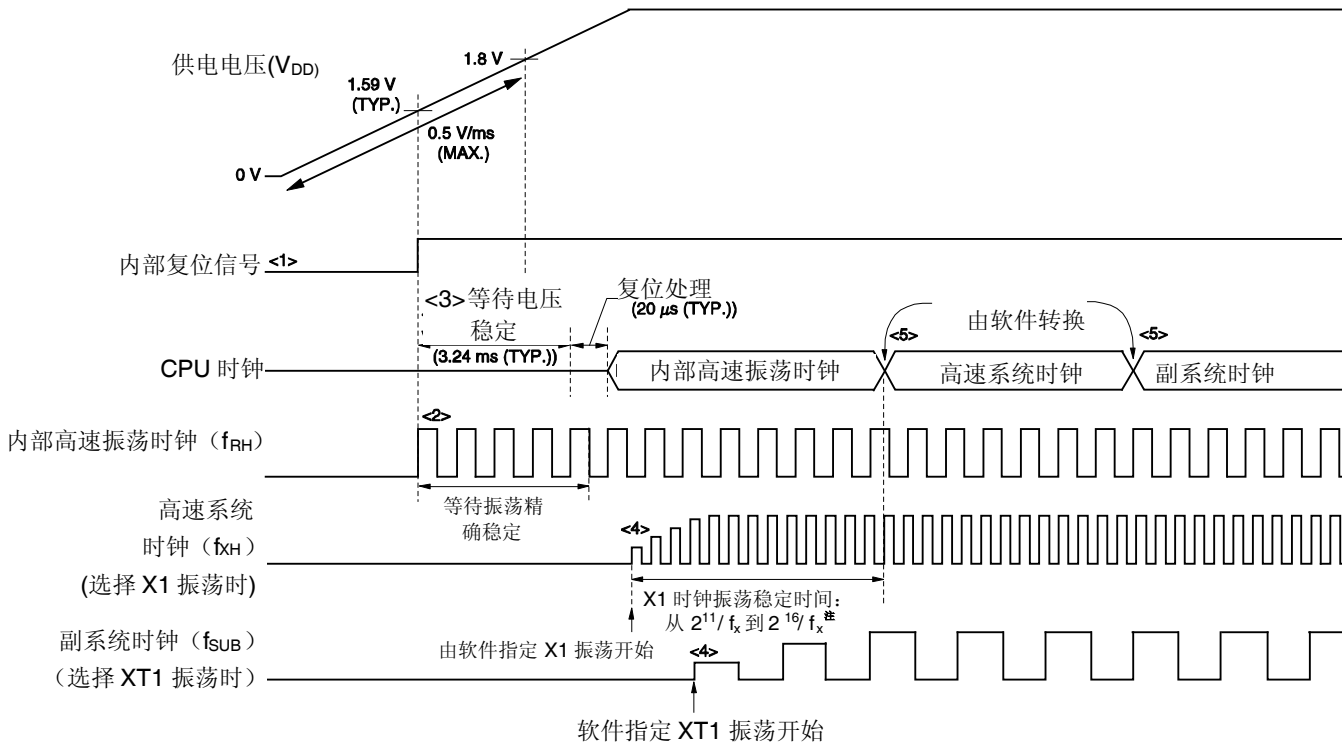
如果 X1 时钟被默认设置为 CPU 时钟，在 X1 时钟遭到损坏或连接错误时设备不能操作，因此复位释放后也不能操作。但是，如果 CPU 的初始时钟是内部高速振荡时钟，则在复位释放后，由内部高速振荡时钟启动设备。这样，系统只需执行最少操作(如由软件确认复位源或在出现故障时执行安全处理)，便可以安全关闭。

(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 5-12 为上电时，时钟发生器的操作图。

图 5-12. 上电时的时钟发生器操作
(设置 1.59 V POC 模式 (选项字节: POCMODE = 0))



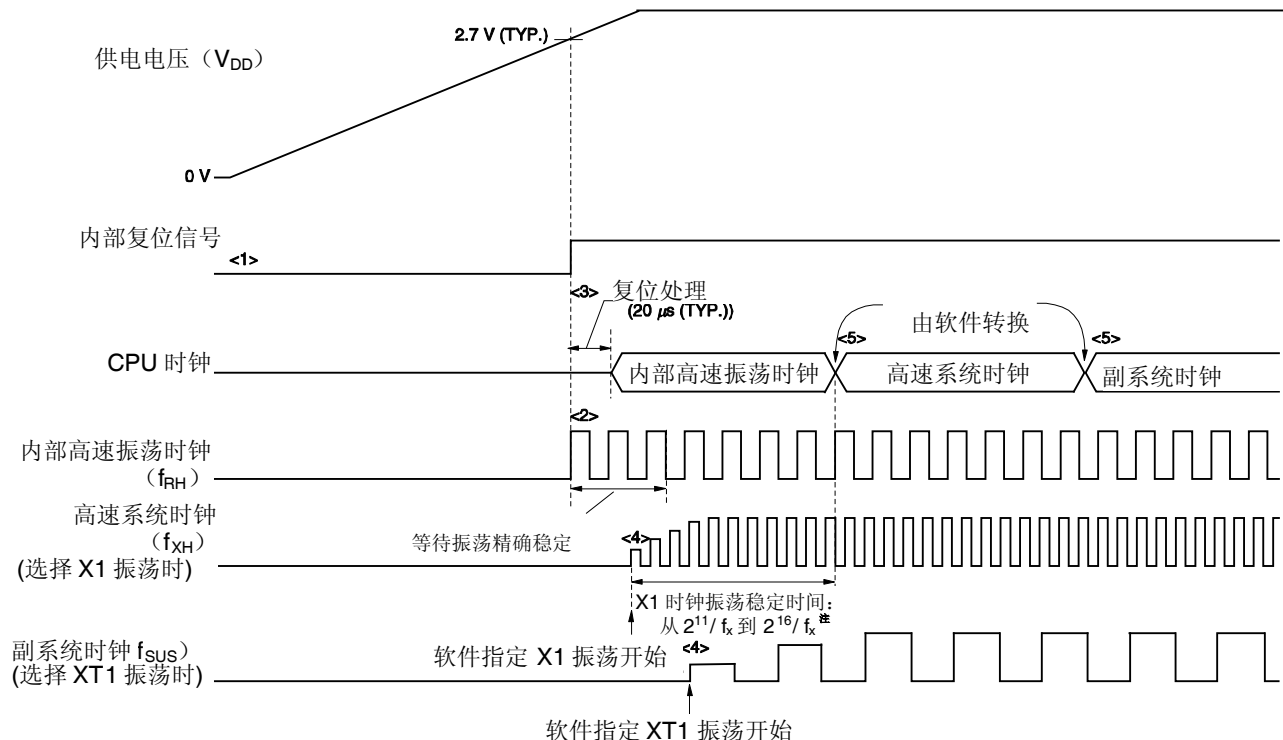
- <1> 当电源开启时，通过上电清零 (POC) 电路产生一个内部复位信号。
- <2> 当供电电压超过 1.59 V (TYP.)，复位释放并且内部高速振荡器自动开始振荡。
- <3> 当供电电压上升斜率为 0.5 V/ms (MAX.) 时，复位释放并且在经历了供电电压和稳压器的稳定时间后，CPU 开始使用内部高速振荡时钟，然后进行复位处理。
- <4> 通过软件设置 X1 或 XT1 时钟的振荡启动(参见 5.6.1 高速系统时钟控制示例中的(1)和 5.6.3 副系统时钟控制示例中的(1))。
- <5> 当 CPU 时钟转换到 X1 或 XT1 时钟时，等待时钟振荡稳定，然后通过软件设置转换(参见 5.6.1 高速系统时钟控制示例中的(3)和 5.6.3 副系统时钟控制示例中的(3))。

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

- 注意事项**
1. 在供电电压达到 1.8V 之前，如果电压上升斜率小于 0.5 V/ms (MAX.)，则输入一个低电平到电源的 **RESET** 引脚直至电压达到 1.8V，或者通过使用选项字节 (POCMODE = 1) 设置 2.7 V/1.59 V POC 模式(见图 5-13)。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 5-12 所示。
 2. 当使用 **EXCLK** 和 **EXCLKS** 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(4))。

图 5-13. 上电时的时钟发生器操作
(设置 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1))



- <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 2.7 V (TYP.)，复位释放并且内部高速振荡器自动开始振荡。
- <3> 复位释放并进行了复位处理后，CPU 开始使用内部高速振荡时钟操作。
- <4> 通过软件设置 X1 或 XT1 时钟的振荡开始(参见 5.6.1 高速系统时钟控制示例中的(1)和 5.6.3 副系统时钟控制示例中的(1))。
- <5> 当 CPU 时钟转换到 X1 或 XT1 时钟时，等待时钟振荡稳定，然后通过软件设置转换(参见 5.6.1 高速系统时钟控制示例中的(3)和 5.6.3 副系统时钟控制示例中的(3))。

注 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 当使用 EXCLK 和 EXCLKS 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(4))。

5.6 时钟控制

5.6.1 高速系统时钟控制示例

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 I/O 端口引脚。

注意事项 复位释放后，X1/P121 和 X2/EXCLK/P122 引脚为 I/O 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) X1 时钟振荡
- (2) 使用外部主系统时钟
- (3) 将高速系统时钟作为 CPU 时钟和外围硬件时钟
- (4) 停止高速系统时钟

(1) X1 时钟振荡时设置过程示例

<1> 设置频率(OSCCTL 寄存器)

使用 AMPH，根据使用频率设置内置振荡器的增益。

AMPH [#]	操作频率控制
0	$1 \text{ MHz} \leq f_{XH} \leq 10 \text{ MHz}$
1	$10 \text{ MHz} < f_{XH} \leq 20 \text{ MHz}$

注 复位释放后设置外部功能前设置 AMPH。复位释放后 AMPH 的值只能被修改一次。当 AMPH=1 时，CPU 时钟停止 5 μs (MIN)。

备注 f_{XH} : 高速系统时钟振荡频率

<2> 设置 P121/X1 和 P122/X2/EXCLK 引脚，并选择 X1 时钟或外部时钟(OSCCTL 寄存器)，当 EXCLK 清零且 OSCSEL 置 1 时，从端口模式转换到 X1 振荡模式

EXCLK	OSCSEL	高速系统时钟引脚操作模式	P121/X1 引脚	P122/X2/EXCLK 引脚
0	1	X1 振荡模式	晶体/陶瓷振荡器连接	

<3> 控制 X1 时钟振荡 (MOC 寄存器)

若 MSTOP 被清零，则 X1 振荡器开始振荡。

<4> 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. 在 X1 时钟运行时，不要改变 EXCLK 和 OSCSEL 的值。

2. 当供电电压达到所用时钟的操作电压后，设置 X1 时钟(参见 第二十九章 电气特性)。

(2) 使用外部主系统时钟时设置过程示例

<1> 设置频率(OSCCTL 寄存器)

使用 AMPH, 设置使用频率。

AMPH ^注	操作频率控制
0	$1 \text{ MHz} \leq f_{XH} \leq 10 \text{ MHz}$
1	$10 \text{ MHz} < f_{XH} \leq 20 \text{ MHz}$

注 复位释放后设置外部功能前设置 AMPH。复位释放后 AMPH 的值只能被修改一次。当 AMPH=1 时，CPU 时钟停止 5 μs (最小)。

备注 f_{XH} : 高速系统时钟振荡频率

<2> 设置 P121/X1 和 P122/X2/EXCLK 引脚并选择操作模式(OSCCTL 寄存器)

当 EXCLK 和 OSCSEL 置 1 时，从端口模式转换到外部时钟输入模式。

EXCLK	OSCSEL	高速系统时钟引脚操作模式	P121/X1 引脚	P122/X2/EXCLK 引脚
1	1	外部时钟输入模式	I/O 端口	外部时钟输入

<3> 控制外部主系统时钟输入(MOC 寄存器)

MSTOP 清零时，允许外部主系统时钟输入。

注意事项 1. 在外部主系统时钟运行时，不要改变 EXCLK 和 OSCSEL 的值。

2. 当供电电压达到所用时钟的操作电压后，设置外部主系统时钟(参见 第二十九章 电气特性)。

(3) 使用高速系统时钟作为 CPU 时钟和外围硬件时钟时设置过程示例<1> 设置高速系统时钟振荡^注

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

注 当高速系统时钟已经运行时，不需要设置<1>。

<2> 设置高速系统时钟作为主系统时钟(MCM 寄存器)

XSEL 和 MCM0 置 1 时，高速系统时钟作为主系统时钟和外围硬件时钟。

XSEL	MCM0	主系统时钟和外围硬件时钟的选择	
		主系统时钟 (f _{XP})	外围硬件时钟 (f _{PRS})
1	1	高速系统时钟 (f _{XH})	高速系统时钟 (f _{XH})

注意事项 如果选择高速系统时钟作为主系统时钟，则只有高速系统时钟可以被设置为外围硬件时钟。

<3> 设置主系统时钟作为 CPU 时钟并选择分频比(PCC 寄存器)

CSS 清零时，主系统时钟提供给 CPU。若要选择 CPU 时钟分频比，应使用 PCC0、PCC1 和 PCC2。

CSS	PCC2	PCC1	PCC0	CPU 时钟 (f _{CPU}) 选择
0	0	0	0	f _{XP}
	0	0	1	f _{XP} /2 (默认)
	0	1	0	f _{XP} /2 ²
	0	1	1	f _{XP} /2 ³
	1	0	0	f _{XP} /2 ⁴
	其它			

(4) 停止高速系统时钟时设置过程示例

高速系统时钟可以用如下两种方式停止。

- 执行 STOP 指令来设置 STOP 模式
- MSTOP 置 1 并停止 X1 振荡(如果使用外部时钟则禁止时钟输入)

(a) 执行 STOP 指令

<1> 停止外围硬件的设置

停止那些不能在 STOP 模式下使用的外围硬件(需要了解不能在 STOP 模式下使用的外围硬件，可参见第二十一章 待机功能)

<2> 释放待机模式后设置 X1 时钟振荡稳定时间

当 CPU 使用 X1 时钟时，在 STOP 指令执行前设置 OSTTS。

<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> 确认 CPU 时钟状态(PCC 和 MCM 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用高速系统时钟。

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

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

<2> 停止高速系统时钟(MOC 寄存器)

MSTOP 置 1 时, X1 振荡停止(禁止外部时钟输入)。

注意事项 当 MSTOP= 1 时, 必须确认 MCS = 0 或 CLS = 1。此外, 必须停止正在使用高速系统时钟的外围硬件。

5.6.2 控制内部高速振荡时钟示例

下面描述的示例是以下情况时的设置过程示例。

(1) 内部高速振荡时钟重新开始振荡时

(2) 使用内部高速振荡时钟作为 CPU 时钟, 并且内部高速振荡时钟或高速系统时钟用作外围硬件时钟时

(3) 停止内部高速振荡时钟时

(1) 内部高速振荡时钟重新振荡时设置过程示例^{#1}

<1> 内部高速振荡时钟重新振荡的设置(RCM 寄存器)

RSTOP 清零时, 内部高速振荡时钟开始运行。

<2> 等待内部高速振荡时钟的振荡精确稳定时间(RCM 寄存器)。

等待直到 RSTS = 1^{#2}。

注 1. 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 时钟使用。

2. 如果 CPU 时钟和外围硬件时钟不需要很高的精度, 则无需等待时间。

(2) 使用内部高速振荡时钟作为 CPU 时钟且内部高速振荡时钟或高速系统时钟作为外围硬件时钟时设置过程示例

- <1> • 内部高速振荡时钟重新振荡^注
(参见 5.6.2 (1) 内部高速振荡时钟重新振荡时设置过程示例)。
- 高速系统时钟振荡^注
(使用高速系统时钟作为外围硬件时钟时所需的设置。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 当内部高速振荡时钟或高速系统时钟已经运行时，不需要设置<1>。

- <2> 选择主系统时钟和外围硬件时钟(MCM 寄存器)
使用 XSEL 和 MCM0 设置主系统时钟和外围硬件时钟。

XSEL	MCM0	主系统时钟和外围硬件时钟的选择	
		主系统时钟 (f _{XP})	外围硬件时钟 (f _{PRS})
0	0	内部高速振荡时钟 (f _{RH})	内部高速振荡时钟 (f _{RH})
0	1		
1	0		高速系统时钟(f _{XH})

- <3> 选择 CPU 时钟分频比(PCC 寄存器)
CSS 清零时，主系统时钟提供给 CPU。若要选择 CPU 时钟分频比，应使用 PCC0、PCC1 和 PCC2。

CSS	PCC2	PCC1	PCC0	CPU 时钟 (f _{CPU}) 选择
0	0	0	0	f _{XP}
	0	0	1	f _{XP} /2 (默认)
	0	1	0	f _{XP} /2 ²
	0	1	1	f _{XP} /2 ³
	1	0	0	f _{XP} /2 ⁴
	其它			禁止设置

(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 STOP 模式
- 将 RSTOP 置 1 且停止内部高速振荡时钟

(a) 执行 STOP 指令

- <1> 设置外围硬件
停止那些不能在 STOP 模式下使用的外围硬件(如需了解不能在 STOP 模式下使用的外围硬件，可参见第二十一章 待机功能)。
- <2> 释放待机模式后 X1 时钟振荡稳定时间的设置
CPU 使用 X1 时钟时，在 STOP 指令执行前设置 OSTs。
- <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 RSTOP 置 1 停止内部高速振荡时钟

<1> 确认 CPU 时钟状态(PCC 和 MCM 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用内部高速振荡时钟。

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

CLS	MCS	CPU 时钟状态
0	0	内部高速振荡时钟
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(RCM 寄存器)

RSTOP 置 1 时, 停止内部高速振荡时钟。

注意事项 当 RSTOP= 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围硬件。

5.6.3 控制副系统时钟示例

可用如下两种副系统时钟。

- XT1 时钟: 晶体/陶瓷振荡器连接到 XT1 和 XT2 引脚。
- 外部副系统时钟: 外部时钟输入到 EXCLKS 引脚。

不使用副系统时钟时, XT1/P123 和 XT2/EXCLKS/P124 引脚可用作 I/O 端口引脚。

注意事项 复位释放后 XT1/P123 和 XT2/EXCLKS/P124 引脚工作在 I/O 端口模式下。

下面描述的示例是以下情况时的设置过程示例。

- (1) XT1 时钟振荡
- (2) 使用外部副系统时钟
- (3) 使用副系统时钟作为 CPU 时钟
- (4) 停止副系统时钟

(1) XT1 时钟振荡时设置过程示例

<1> 设置 XT1 和 XT2 引脚并选择操作模式 (PCC 和 OSCCTL 寄存器)

当 XTSTART、EXCLKS 和 OSCSELS 设置为如下模式时, 将从端口模式转换到 XT1 振荡模式。

XTSTART	EXCLKS	OSCSELS	副系统时钟引脚操作模式	P123/XT1 引脚	P124/XT2/EXCLKS 引脚
0	0	1	XT1 振荡模式	连接晶体/陶瓷振荡器	
1	x	x			

备注 x: 不必考虑

<2> 等待副系统时钟振荡稳定

使用定时器功能, 通过软件等待副系统时钟振荡稳定时间。

注意事项 在副系统时钟运行时, 不要改变 XTSTART, EXCLKS 和 OSCSELS 的值。

(2) 使用外部副系统时钟时设置过程示例

<1> 设置 XT1 和 XT2 引脚, 选择 XT1 时钟/外部时钟和控制振荡 (PCC 和 OSCCTL 寄存器)

当 XTSTART 清零并且 EXCLKS 和 OSCSELS 置 1 时, 将从端口模式转换到外部时钟输入模式。在这种情况下, 将外部时钟输入到 EXCLKS/XT2/P124 引脚。

XTSTART	EXCLKS	OSCSELS	副系统时钟引脚操作模式	P123/XT1 引脚	P124/XT2/EXCLKS 引脚
0	1	1	外部时钟输入模式	I/O 端口	外部时钟输入

注意事项 在副系统时钟运行时, 不要改变 XTSTART, EXCLKS 和 OSCSELS 的值。

(3) 使用副系统时钟作为 CPU 时钟时设置过程示例

<1> 设置副系统时钟振荡^{*}

(参见 5.6.3 (1) XT1 时钟振荡时设置过程示例和 (2) 使用外部副系统时钟时设置过程示例。)

注 当副系统时钟已经运行时, 不需要设置<1>。

<2> 转换 CPU 时钟(PCC 寄存器)

CSS 置 1 时, 副系统时钟提供给 CPU。

CSS	PCC2	PCC1	PCC0	CPU 时钟 (f _{cpu}) 选择
1	0	0	0	f _{sub} /2
	0	0	1	
	0	1	0	
	0	1	1	
	1	0	0	禁止设置
其它				禁止设置

(4) 停止副系统时钟时设置过程示例

<1> 确认 CPU 时钟状态(PCC 和 MCM 寄存器)

根据 CLS 和 MCS 确认 CPU 没有使用副系统时钟。

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

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

<2> 停止副系统时钟 (OSCCTL 寄存器)

OSCSELS 清零时, 停止 XT1 振荡(禁止外部时钟输入)。

注意事项 1. 清零 OSCSEL 时, 必须确认 CLS = 0。此外, 如果钟表定时器使用副系统时钟, 则应停止其操作。

2. 副系统时钟振荡不能通过使用 STOP 指令来停止。

5.6.4 控制内部低速振荡时钟示例

内部低速振荡时钟不能用作 CPU 时钟。

只有如下外围硬件可以使用这个时钟。

- 看门狗定时器
- 8 位定时器 H1(选择 f_{RL} 作为计数时钟时)

此外，可以通过选项字节选择如下操作模式。

- 内部低速振荡器不能停止
- 内部低速振荡器可由软件停止

在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(240 kHz (TYP.))。

(1) 停止内部低速振荡时钟时设置过程示例

<1> 将 LSRSTOP 置 1 (RCM 寄存器)

LSRSTOP 置 1 时，停止内部低速振荡时钟。

(2) 内部低速振荡时钟重新振荡时设置过程示例

<1> LSRSTOP 清零 (RCM 寄存器)

LSRSTOP 清零时，内部低速振荡时钟重新振荡。

注意事项 如果通过选项字节选择“不能停止内部低速振荡器”，则不能控制内部低速振荡时钟的振荡。

5.6.5 CPU 和外围硬件所采用的时钟

下表显示了 CPU 和外围硬件采用的时钟之间的关系及寄存器的设置。

表 5-4. CPU 和外围硬件采用的时钟及寄存器设置

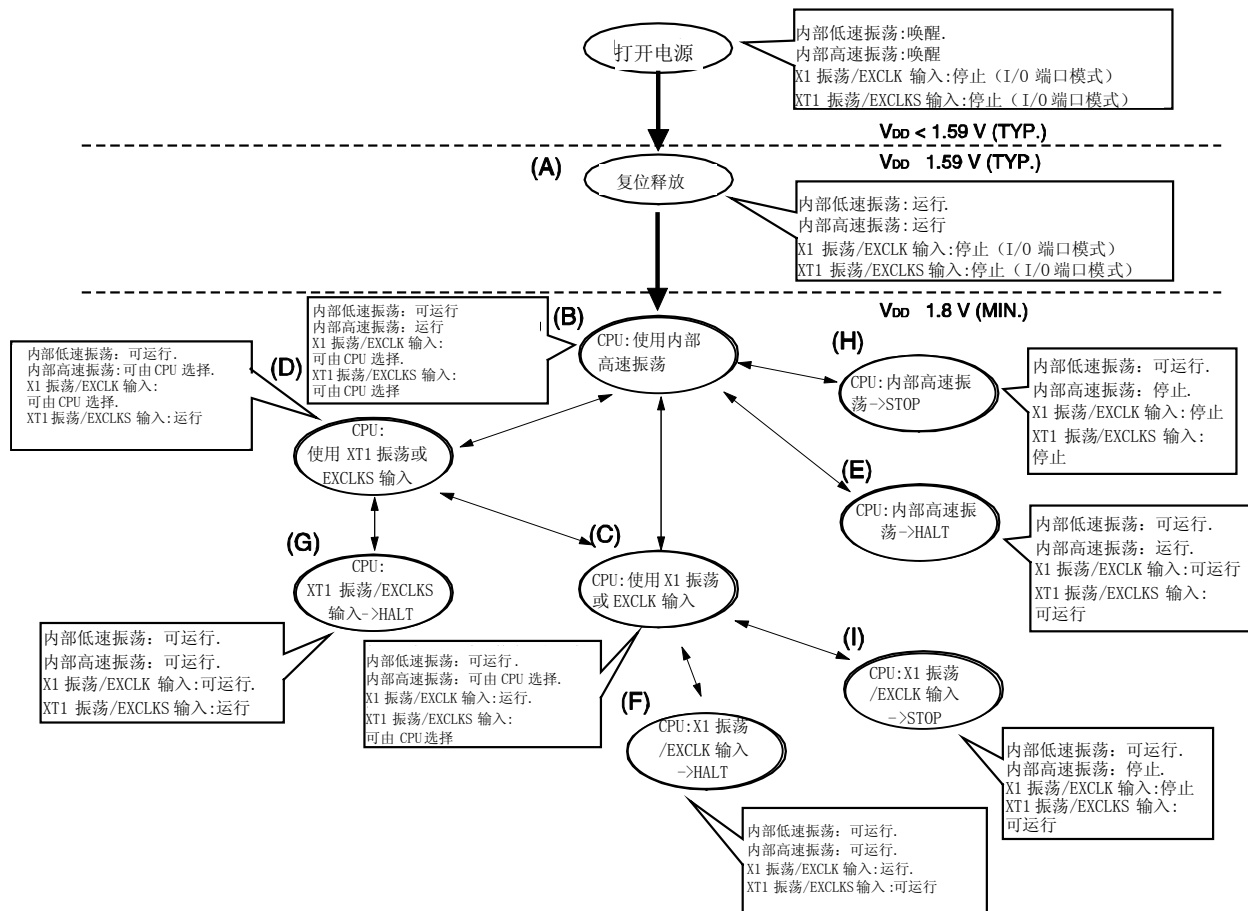
时钟		XSEL	CSS	MCM0	EXCLK
CPU 时钟	外围硬件时钟				
内部高速振荡时钟		0	0	×	×
内部高速振荡时钟	X1 时钟	1	0	0	0
	外部主系统时钟	1	0	0	1
X1 时钟		1	0	1	0
外部主系统时钟		1	0	1	1
副系统时钟	内部高速振荡时钟	0	1	×	×
	X1 时钟	1	1	0	0
		1	1	1	0
	外部主系统时钟	1	1	0	1
		1	1	1	1

- 备注**
1. XSEL: 主时钟模式寄存器(MCM)的第 2 位
 2. CSS: 处理器时钟控制寄存器(PCC)的第 4 位
 3. MCM0: MCM 的第 0 位
 4. EXCLK: 时钟操作模式选择寄存器(OSCCTL)的第 7 位
 5. ×: 不必考虑

5.6.6 CPU 时钟状态转换图

图 5-14 显示了该产品 CPU 时钟状态转换图。

图 5-14. CPU 时钟状态转换图
(设置 1.59 V POC 模式 (选项字节: POCMODE = 0))



备注 在 2.7 V/1.59 V POC 模式下 (选项字节: POCMODE = 1), 当供电电压超过 2.7V(TYP.)时 CPU 时钟状态转换为上图所示的(A), 而在复位处理后(20 μ s (TYP.))变到(B)。

表 5-5 显示了 CPU 时钟的转换过程与 SFR 寄存器设置示例

表 5-5. CPU 时钟转换与 SFR 寄存器设置示例(1/4)

(1) 复位释放(A)后 CPU 使用内部高速振荡时钟(B)

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

(2) 复位释放(A)后 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 寄存器设置顺序) →

SFR 寄存器设置标志	AMPH	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL	MCM0
状态转换							
(A) → (B) → (C) (X1 时钟: $1 \text{ MHz} \leq f_{\text{XH}} \leq 10 \text{ MHz}$)	0	0	1	0	必须检测	1	1
(A) → (B) → (C) (外部主时钟: $1 \text{ MHz} \leq f_{\text{XH}} \leq 10 \text{ MHz}$)	0	1	1	0	不必检测	1	1
(A) → (B) → (C) (X1 时钟: $10 \text{ MHz} < f_{\text{XH}} \leq 20 \text{ MHz}$)	1	0	1	0	必须检测	1	1
(A) → (B) → (C) 外部主时钟: $10 \text{ MHz} < f_{\text{XH}} \leq 20 \text{ MHz}$)	1	1	1	0	不必检测	1	1

注意事项 供电电压达到所用时钟的操作电压后, 设置时钟(参见 第二十九章 电气特性)。

(3) 复位释放(A)后 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 寄存器设置顺序) →

SFR 寄存器设置标志	XTSTART	EXCLKS	OSCSLS	等待振荡稳定	CSS
状态转换					
(A) → (B) → (D) (XT1 时钟)	0	0	1	必须	1
	1	×	×		
(A) → (B) → (D) (外部副系统时钟)	0	1	1	不必要	1

备注

- 表 5-5 的(A) 到 (I)对应图 5-14 的(A)到(I)。
- EXCLK, OSCSEL, EXCLKS, OSCSELS, AMPH:
时钟操作模式选择寄存器(OSCCTL)的第 7 位~第 4 位与第 0 位
- MSTOP: 主 OSC 控制寄存器(MOC)的第 7 位
- XSEL, MCM0: 主时钟模式寄存器(MCM)的第 2 位与第 0 位
- XTSTART, CSS: 处理器时钟控制寄存器(PCC)的第 6 位与第 4 位
- ×: 不必考虑

表 5-5. CPU 时钟转换与 SFR 寄存器设置示例(2/4)

(4) CPU 时钟从内部高速振荡时钟(B)转换到高速系统时钟(C)

(SFR 寄存器的设置顺序) ➔

SFR 寄存器的设置标志	AMPH ^注	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL ^注	MCM0
状态转换							
(B) → (C) (X1 时钟: $1 \text{ MHz} \leq f_{\text{XH}} \leq 10 \text{ MHz}$)	0	0	1	0	必须检测	1	1
(B) → (C) (外部主时钟: $1 \text{ MHz} \leq f_{\text{XH}} \leq 10 \text{ MHz}$)	0	1	1	0	不必检测	1	1
(B) → (C) (X1 时钟: $10 \text{ MHz} < f_{\text{XH}} \leq 20 \text{ MHz}$)	1	0	1	0	必须检测	1	1
(B) → (C) (外部主时钟: $10 \text{ MHz} < f_{\text{XH}} \leq 20 \text{ MHz}$)	1	1	1	0	不必检测	1	1

如果这些寄存器已设置则不必要 如果 CPU 使用高速系统时钟则不必要

注 复位释放后该标志只能被修改一次。如果已经设置了该项，则无需再设置。

注意事项 供电电压达到所用时钟的操作电压后，设置时钟(参见 第二十九章 电气特性)。

(5) CPU 时钟从内部高速振荡时钟(B)转换到副系统时钟(D)

(SFR 寄存器的设置顺序) ➔

SFR 寄存器的设置标志	XTSTART	EXCLKS	OSCSELS	等待振荡稳定	CSS
状态转换					
(B) → (D) (XT1 时钟)	0	0	1	必须	1
	1	×	×		
(B) → (D) (外部副系统时钟)	0	1	1	不必要	1

如果 CPU 使用副系统时钟则不必要

- 备注**
- 表 5-5 的(A) 到 (I)对应图 5-14 的(A)到(I)。
 - EXCLK, OSCSEL, EXCLKS, OSCSELS, AMPH:
时钟操作模式选择寄存器(OSCCTL)的第 7 位~第 4 位与第 0 位
 - MSTOP: 主 OSC 控制寄存器(MOC)的第 7 位
 - XSEL, MCM0: 主时钟模式寄存器(MCM)的第 2 位与第 0 位
 - XTSTART, CSS: 处理器时钟控制寄存器(PCC)的第 6 位与第 4 位
 - ×: 不必考虑

表 5-5. CPU 时钟转换与 SFR 寄存器设置示例 (3/4)

(6) CPU 时钟从高速系统时钟(C)转换到内部高速振荡时钟(B)

(SFR 寄存器的设置顺序) →

状态转换	SFR 寄存器的设置标志	RSTOP	RSTS	MCM0
(C) → (B)		0	确认该标志为 1	0

如果 CPU 使用内部高速振荡时钟则不必要

(7) CPU 时钟从高速系统时钟(C)转换到副系统时钟(D)

(SFR 寄存器的设置顺序)

状态转换	SFR 寄存器的设置标志	XTSTART	EXCLKS	OSCSLS	等待振荡稳定	CSS
(C) → (D) (XT1 时钟)		0	0	1	必须	1
		1	×	×		
(C) → (D) (外部副系统时钟)		0	1	1	不必要	1

如果 CPU 使用副系统时钟则不必要

(8) CPU 时钟从副系统时钟(D)转换到内部高速振荡时钟(B)

(SFR 寄存器的设置顺序) →

状态转换	SFR 寄存器的设置标志	RSTOP	RSTS	MCM0	CSS
(D) → (B)		0	确认该标志为 1	0	0

如果 CPU 使用内部高速振荡时钟则不必要 如果 XSEL=0 则不必要

备注

- 表 5-5 的(A) 到 (I) 对应图 5-14 的(A)到(I)。
- MCM0: 主时钟模式寄存器(MCM)的第 0 位
EXCLKS, OSCSLS: 时钟操作模式选择寄存器(OSCCTL)的第 5 位与第 4 位
RSTS, RSTOP: 内部振荡模式寄存器(RCM)的第 7 位和第 0 位
XTSTART, CSS: 处理器时钟控制寄存器(PCC)的第 6 位与第 4 位
×: 不必考虑

表 5-5. CPU 时钟转换与 SFR 寄存器设置示例 (4/4)

(9) CPU 时钟从副系统时钟(D)转换到高速系统时钟(C)

(SFR 寄存器的设置顺序) →

SFR 寄存器的设置标志	AMPH [‡]	EXCLK	OSCSEL	MSTOP	OSTC 寄存器	XSEL [‡]	MCM0	CSS
状态转换								
(D) → (C) (X1 时钟: $1 \text{ MHz} \leq f_{XH} \leq 10 \text{ MHz}$)	0	0	1	0	必须检测	1	1	0
(D) → (C) (外部主时钟: $1 \text{ MHz} \leq f_{XH} \leq 10 \text{ MHz}$)	0	1	1	0	不必检测	1	1	0
(D) → (C) (X1 时钟: $10 \text{ MHz} < f_{XH} \leq 20 \text{ MHz}$)	1	0	1	0	必须检测	1	1	0
(D) → (C) (外部主时钟: $10 \text{ MHz} < f_{XH} \leq 20 \text{ MHz}$)	1	1	1	0	不必检测	1	1	0

如果这些寄存器已设置则不必要
如果 CPU 使用高速系统时钟则不必要
如果寄存器已设置则不必要

注 复位释放后该标志只能被修改一次。如果已经设置了该项，则无需再设置。

注意事项 供电电压达到所用时钟的操作电压后，设置时钟(参见 第二十七章 电气特性)。

(10) • 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。

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

(11) • CPU 使用内部高速振荡时钟(B)时 STOP 模式(H)的设置。

- CPU 使用高速系统时钟(C)时 STOP 模式(I)的设置。

(设置顺序) →

状态转换	设置	
(B) → (H) (C) → (I)	停止那些不能在 STOP 模式下使用的外部功能	执行 STOP 指令

备注

1. 表 5-5 的(A) 到 (I)对应图 5-14 的(A)到(I)。
2. EXCLK, OSCSEL, AMPH: 时钟操作模式选择寄存器(OSCCTL)的第 7 位,第 6 位和第 0 位
MSTOP: 主 OSC 控制寄存器(MOC)的第 7 位
XSEL, MCM0: 主时钟模式寄存器(MCM)的第 2 位和第 0 位
CSS: 处理器时钟控制寄存器(PCC) 的第 4 位

5.6.7 CPU 时钟转换之前的状况与转换之后的处理

CPU 时钟转换之前的状况与转换之后的处理显示如下。

表 5-6. 转换 CPU 时钟

CPU 时钟		转换前的状况	转换后的处理
转换前	转换后		
内部高速振荡时钟	X1 时钟	X1 振荡稳定 • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • 经历了振荡稳定时间	<ul style="list-style-type: none"> • 可以停止内部高速振荡器 (RSTOP = 1). • AMPH 被设置为 1 后 CPU 时钟停止 5 μs (MIN.)。
	外部主系统时钟	允许来自 EXCLK 引脚的外部时钟输入 • MSTOP = 0, OSCSEL = 1, EXCLK = 1	
X1 时钟	内部高速振荡时钟	内部高速振荡器振荡 • RSTOP = 0	可以停止 X1 振荡(MSTOP = 1).
外部主系统时钟			可以禁止外部主系统时钟输入(MSTOP = 1).
内部高速振荡时钟	XT1 时钟	XT1 振荡稳定 • XTSTART = 0, EXCLKS = 0, OSCSELS = 1 或 XTSTART = 1 • 经历了振荡稳定时间	通过停止内部高速振荡器可以降低操作电流 (RSTOP = 1)
X1 时钟			可以停止 X1 振荡 (MSTOP = 1)
外部主系统时钟			可以禁止外部主系统时钟输入(MSTOP = 1)
内部高速振荡时钟	外部副系统时钟	允许来自 EXCLK 引脚的外部时钟输入 • XTSTART = 0, EXCLKS = 1, OSCSELS = 1	通过停止内部高速振荡器可以降低操作电流 (RSTOP = 1).
X1 时钟			可以停止 X1 振荡(MSTOP = 1).
外部主系统时钟			可以禁止外部主系统时钟输入(MSTOP = 1).
XT1 时钟, 外部副系统时钟	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 • RSTOP = 0, MCS = 0	可停止 XT1 振荡或者禁止外部副系统时钟输入(OSCSELS = 0).
	X1 时钟	X1 振荡稳定且选择高速系统时钟作为主系统时钟 • MSTOP = 0, OSCSEL = 1, EXCLK = 0 • 经历了振荡稳定时间 • MCS = 1	<ul style="list-style-type: none"> • 可停止 XT1 振荡或者禁止外部副系统时钟输入(OSCSELS = 0). • AMPH 被设置为 1 后 CPU 时钟停止 5 μs (MIN.)。
	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 • MSTOP = 0, OSCSEL = 1, EXCLK = 1 • MCS = 1	

5.6.8 CPU 时钟和主系统时钟转换所需的时间

通过设置处理器时钟控制器(PCC)的第 0 位~第 2 位(PCC0~PCC2)以及第 4 位(CSS)，可以转换 CPU 时钟(在主系统时钟和副系统时钟之间)，同时可以改变主系统时钟的分频比。

修改 PCC 后，实际的转换操作不会立即执行；使用转换前的时钟继续操作几个时钟(见 表 5-7)。

可以通过 PCC 寄存器的第 5 位(CLS)来确定 CPU 使用主系统时钟还是副系统时钟。

表 5-7. CPU 时钟转换所需时间与主系统时钟周期分频因子

转换前设置值				转换后设置值																																																																															
CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0	CSS	PCC2	PCC1	PCC0																																																								
				0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1	×	×	×																																																								
0	0	0	0	16 个时钟				16 个时钟				16 个时钟				16 个时钟				$2f_{XP}/f_{SUB}$ 个时钟																																																															
	0	0	1																					8 个时钟				8 个时钟				8 个时钟				8 个时钟				f_{XP}/f_{SUB} 个时钟																																											
	0	1	0																																									4 个时钟				4 个时钟				4 个时钟				4 个时钟				$f_{XP}/2f_{SUB}$ 个时钟																							
	0	1	1																																																													2 个时钟				2 个时钟				2 个时钟				2 个时钟				$f_{XP}/4f_{SUB}$ 个时钟			
	1	0	0																																																																																
1	×	×	×	2 个时钟				2 个时钟				2 个时钟				2 个时钟				2 个时钟																																																															

注意事项 主系统时钟周期分频因子(PCC0~PCC2)的选择，以及从主系统时钟到副系统时钟的转换(CSS 从 0 变到 1)不能同时设置。

但对于主系统时钟周期分频因子(PCC0~PCC2)的选择，以及副系统时钟到主系统时钟的转换(CSS 从 1 变到 0)可以同时设置。

备注

1. 表 5-7 列出的时钟数是转换前的 CPU 时钟数。
2. 将 CPU 时钟从主系统时钟转换到副系统时钟，可通过舍入一个时钟并舍弃小数部分来计算时钟数量(如下所示)。

示例： 将 CPU 时钟从 $f_{SUB}/2$ 转换到 $f_{XP}/2$ (@ 振荡频率 $f_{SUB} = 32.768 \text{ kHz}$, $f_{XP} = 10 \text{ MHz}$)
 $f_{XP}/f_{SUB} = 10000/32.768 \cong 305.1 \rightarrow 306$ 个时钟

通过设置主时钟模式寄存器(MCM)的第 0 位 (MCM0)，可以转换主系统时钟(在内部高速振荡时钟和高速系统时钟之间)。

修改 MCM0 后，实际的转换操作不会立即执行；使用转换前的时钟继续操作几个时钟(见 表 5-8)。

可以通过 MCM 寄存器的第 1 位(MCS)来确定 CPU 使用内部高速振荡时钟还是高速系统时钟。

表 5-8. 主系统时钟转换所需的最长时间

转换前设置值	转换后设置值	
MCM0	MCM0	
	0	1
0		$1 + 2f_{RH}/f_{XH}$ 时钟
1	$1 + 2f_{XH}/f_{RH}$ 时钟	

注意事项 当内部高速振荡时钟转换到高速系统时钟时，MCM 的第 2 位(XSEL)必须预先被设置为 1。XSEL 位在复位释放后只能被修改一次。

备注

1. 表 5-8 所列时钟数是转换之前的主系统时钟数。
2. 通过舍去小数部分，计算表 5-8 的时钟数。

示例： 将主系统时钟从内部高速振荡时钟转换到高速系统时钟(@振荡频率 $f_{RH} = 8$ MHz, $f_{XH} = 10$ MHz)

$$1 + 2f_{RH}/f_{XH} = 1 + 2 \times 8/10 = 1 + 2 \times 0.8 = 1 + 1.6 = 2.6 \rightarrow 2 \text{ 个时钟}$$

5.6.9 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 5-9. 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 不使用内部高速振荡时钟)	RSTOP = 1
X1 时钟	MCS = 1 或 CLS = 1 (CPU 不使用高速系统时钟)	MSTOP = 1
外部主系统时钟		
XT1 时钟	CLS = 0 (CPU 不使用副系统时钟)	OSCSELS = 0
外部副系统时钟		

5.6.10 外围硬件与源时钟

下表列出了 78K0/LF2 中包含的外围硬件与源时钟。

表 5-10. 外围硬件与源时钟

源时钟 外围硬件		外围硬件时钟 (fPRS)	副系统时钟 (fSUB)	内部低速振荡时钟 (fRL)	TM50 输出	来自外围硬件引脚 的外部时钟
16 位定时器/ 事件计数器 00	00	Y	N	N	N	Y (TI000 引脚) ^注
	01	Y	N	N	N	N
8 位定时器/ 事件计数器	50	Y	N	N	N	Y (TI50 引脚) ^注
	51	Y	N	N	N	Y (TI51 引脚) ^注
8 位定时器	H0	Y	N	N	Y	N
	H1	Y	N	Y	N	N
钟表定时器		Y	Y	N	N	N
看门狗定时器		N	N	Y	N	N
时钟输出		Y	Y	N	N	N
A/D 转换器		Y	N	N	N	N
串行接口	UART0	Y	N	N	Y	N
	UART6	Y	N	N	Y	N
	CSI10	Y	N	N	N	Y (SCK10 引脚) ^注
	IIC0	Y	N	N	N	Y (SCL0 引脚) ^注
LCD 控制/驱动器		Y	Y	N	N	N

注 当 CPU 使用副系统时钟且内部高速振荡时钟已经停止时，不要启动依赖由外围硬件引脚输入外部时钟的这些功能。

备注 Y: 可选, N: 不可选

第六章 16 位定时器/事件计数器 00 和 01

μ PD78F0372, 78F0373, 78F0382 和 78F0383 有 16 位定时器/事件计数器 00, μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D 有 16 位定时器/事件计数器 00 和有 16 位定时器 01。

6.1 16 位定时器/事件计数器 00 和 01 的功能

16 位定时器/事件计数器 00 和 01^注 有如下功能。

- 间隔定时器
- PPG 输出
- 脉冲宽度测量
- 外部事件计数器
- 方波输出
- 单脉冲输出

(1) 间隔定时器

16 位定时器/事件计数器 00 和 01 以预置的时间间隔产生中断请求。

(2) PPG 输出 (仅定时器 00)

16 位定时器/事件计数器 00 和 01 可以输出矩形波, 矩形波的频率和输出宽度可以自由设置。

(3) 脉冲宽度测量 (仅定时器 00)

16 位定时器/事件计数器 00 和 01 可以测量外部输入信号的脉冲宽度。

(4) 外部事件计数器 (仅定时器 00)

16 位定时器/事件计数器 00 和 01 可以测量外部输入信号的脉冲数。

(5) 方波输出 (仅定时器 00)

16 位定时器/事件计数器 00 和 01 可以输出任选频率的方波。

(6) 单脉冲输出 (仅定时器 00)

16 位定时器/事件计数器 00 和 01 可以输出任意脉冲宽度的单脉冲。

注 仅在 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D 中可用。

6.2 16 位定时器/事件计数器 00 和 01 的配置

16 位定时器/事件计数器 00 和 01 包括以下硬件。

表 6-1. 16 位定时器/事件计数器 00 和 01 的配置

项目	配置
定时器/计数器	16 位定时器计数器 0n (TM0n)
寄存器	16 位定时器捕捉/比较寄存器 00n, 01n (CR00n, CR01n)
定时器输入	TI000, TI010 引脚
定时器输出	TO00 引脚, 输出控制器
控制寄存器	16 位定时器模式控制寄存器 0n (TMC0n) 16 位定时器捕捉/比较控制寄存器 0n (CRC00) 16 位定时器输出控制寄存器 0n (TOC00) 预分频器模式寄存器 0n (PRM0n) 端口模式寄存器 0 (PM0) 端口寄存器 0 (P0)

备注 n = 0: μ PD78F0372, 78F0373, 78F0382 和 78F0383

n = 0, 1: μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D

图 6-1 和 6-2 显示了 16 位定时器/事件计数器的框图。

图 6-1 16 位定时器/事件计数器 00 的框图

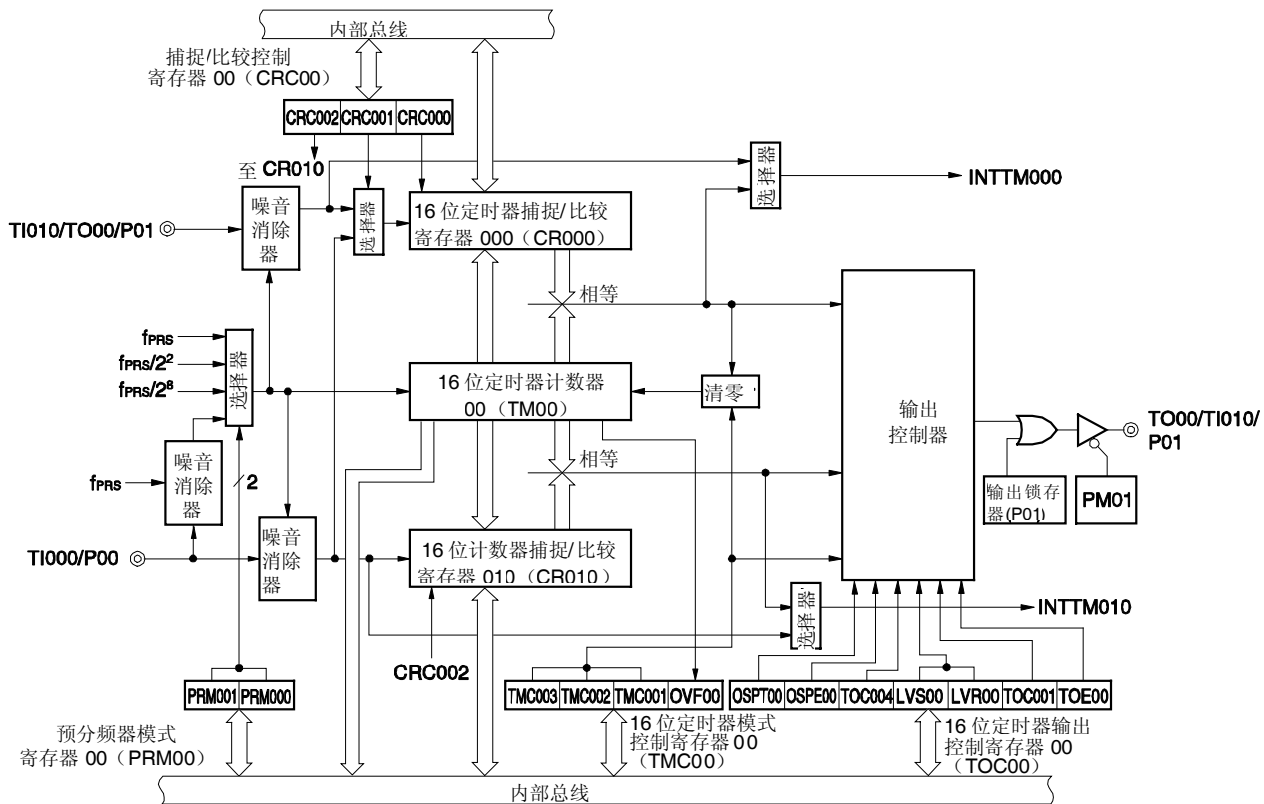
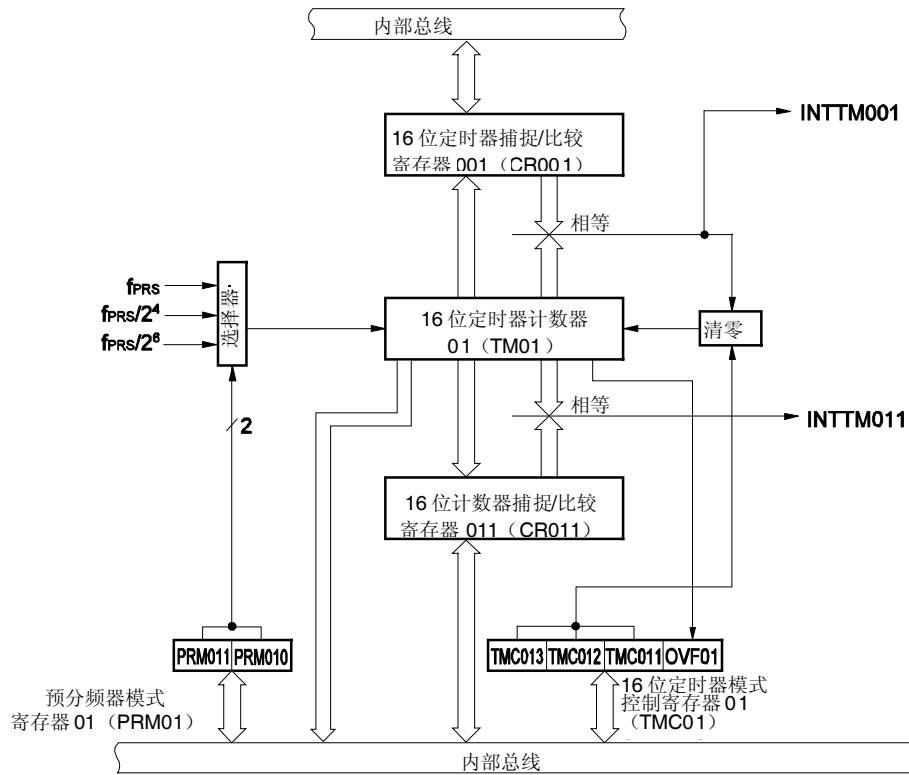


图 6-2 16 位定时器/事件计数器 01 的框图
 (仅 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D)

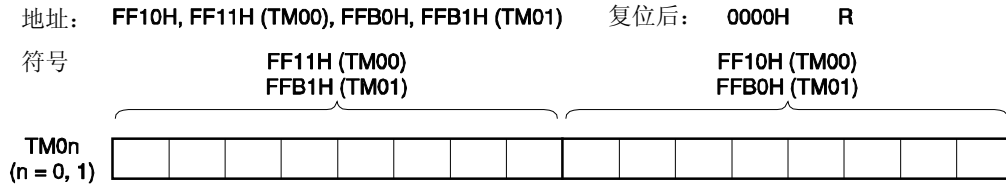


(1) 16 位定时器计数器 0n (TM0n)

TM0n 是 16 位只读寄存器，用于脉冲计数。

计数器随计数时钟的上升沿次数同步增加。

图 6-3. 16 位定时器计数器 0n (TM0n) 的格式



如出现以下情况，计数值复位为 0000H。

- <1> 复位信号的产生
- <2> TMC0n3 和 TMC0n2 被清零
- <3> 在 TI00n 引脚输入有效沿产生清零和启动模式下，TI000 引脚输入有效沿时
- <4> 在 TM0n 和 CR00n 相等时产生清零和启动模式下，如果 TM0n 和 CR00n 相等时
- <5> 在单脉冲输出模式下设置 OSPT00

备注 n = 0: μ PD78F0372, 78F0373, 78F0382 和 78F0383
 n = 0, 1: μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D

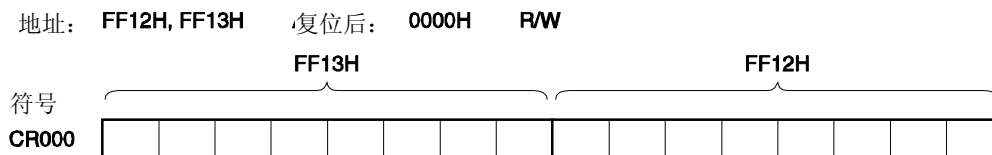
(2) 16 位定时器捕捉/比较寄存器 000 (CR000)

CR000 是 16 位寄存器，具有捕捉或比较功能，通过捕捉/比较控制寄存器 00 (CRC00) 的第 0 位 (CRC000) 设置为捕捉寄存器或比较寄存器。

可用 16 存储器指令操作 CR000。

复位信号的产生可将这些寄存器设置为 0000H。

图 6-4. 16 位定时器捕捉/比较寄存器 000 (CR000) 的格式



- **CR000 用作比较寄存器**

CR000 的设置值与 16 位定时器计数器 00 (TM00) 的计数值频繁地相比较，如果相等将产生一个中断请求 (INTTM000)。在 CR000 被重写前一直保持这个值。

- **CR000 用作捕捉寄存器**

可以选择 TI000 或 TI010 引脚的有效沿作为捕捉触发。通过使用预分频模式寄存器 00 (PRM00) 设置 TI000 或 TI010 引脚的有效沿 (参看 表 6-2)。

表 6-2. CR000 捕捉触发器和 TI000 与 TI010 引脚的有效沿

(1) TI000 引脚作为捕捉触发器的有效沿选择 (CRC001 = 1, CRC000 = 1)

CR000 捕捉触发器	TI000 引脚有效沿		
	ES001	ES000	
下降沿	上升沿	0	1
上升沿	下降沿	0	0
无捕捉操作	上升沿和下降沿	1	1

(2) TI010 引脚作为捕捉触发器的有效沿选择 (CRC001 = 0, CRC000 = 1)

CR000 捕捉触发器	TI010 引脚有效沿		
	ES101	ES100	
下降沿	下降沿	0	0
上升沿	上升沿	0	1
上升沿和下降沿	上升沿和下降沿	1	1

- 备注**
1. 禁止设置 ES001, ES000 = 1, 0 和 ES101, ES100 = 1, 0。
 2. ES001, ES000: 预分频模式寄存器 00 (PRM00) 的第 5 和 4 位
ES101, ES100: 预分频模式寄存器 00 (PRM00) 的第 7 和 6 位
CRC001, CRC000: 捕捉/比较控制寄存器 00 (CRC00) 的第 1 和 0 位

- 注意事项**
1. TM00 和 CR000 相等时产生清零&启动，在此模式下设置 CR000 为 0000H 以外的值。
 2. 如果 CR000 寄存器在自由运行模式下被清零为 0000H，并且在清零模式下使用 TI000 引脚的有效沿，则在 TM00 溢出 (FFFFH) 之后当 CR000 的值从 0000H 变为 0001H 时产生一个中断请求 (INTTM00)。另外，在 TM00 和 CR000 相等后，产生 INTTM00，检测 TI010 引脚的有效边沿后，通过单脉冲清零这个定时器。
 3. 当 P01 作为 TI010 引脚的有效沿使用时，它不能用于定时器输出 (TO00)。此外，当 P01 作为 TO00 时，它不能用于 TI010 引脚的有效沿输入。
 4. 当 CR000 作为捕捉寄存器，如果寄存器读取时序和捕捉触发冲突则所读的数据是不确定的 (捕捉数据本身是正确的)。
如果计数停止输入和捕捉触发相冲突，则捕捉值不确定。
 5. 在 TM00 操作时，不能重写 CR000。

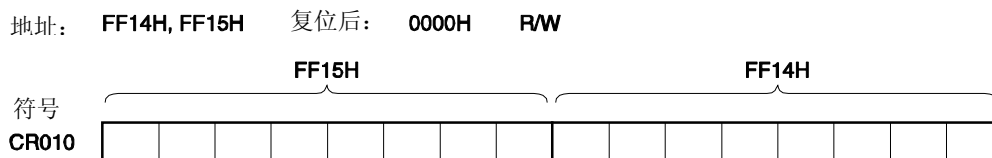
(3) 16 位定时器捕捉/比较寄存器 010 (CR010)

CR010 是 16 位寄存器，具有捕捉寄存器和比较寄存器的功能，通过捕捉/比较控制寄存器 CRC00 的第 2 位 (CRC002) 选择其捕捉或比较寄存器的功能。

可用 16 存储器指令操作 CR010。

复位信号的产生可将这些寄存器设置为 0000H。

图 6-5. 16 位定时器捕捉/比较寄存器 010 (CR010) 的格式



- **CR010 用作比较寄存器**

CR010 的设置值与 16 位定时器计数器 TM00 的计数值频繁地相比较，如果相等将产生一个中断请求 (INTTM010)。这个值直到 CR010 重写前都不会改变。

- **CR010 用作捕捉寄存器**

可以选择 TI000 引脚的有效沿作为捕捉触发。TI000 有效沿由预分频模式寄存器 PRM00 来设置(参看表 6-3)。

表 6-3. CR010 捕捉触发器和 TI000 引脚的有效边沿 (CRC002 = 1)

CR010 捕捉触发器	TI000 引脚有效沿		
	ES001	ES000	
下降沿	下降沿	0	0
上升沿	上升沿	0	1
上升沿和下降沿	上升沿和下降沿	1	1

备注

1. 禁止设置 ES001, ES000 = 1, 0 和 ES101, ES100 = 1, 0。
2. ES001, ES000: 预分频模式寄存器 00 (PRM00) 的第 5 位和第 4 位
CRC002: 捕捉/比较控制寄存器 00 (CRC00) 的第 2 位

- 注意事项
1. 如果 CR010 寄存器清零为 0000H，则在 TM00 溢出 (FFFFH) 以后，当 CR010 的值从 0000H 变为 0001H 时，产生一个中断请求 (INTTM010)。另外，在 TM00 和 CR010 相等后，产生 INTTM010，检测 TI000 引脚的有效边沿后，通过单脉冲清零这个定时器。
 2. 当 CR010 作为捕捉寄存器时，如果寄存器读取时序和捕捉触发输入相冲突 (捕捉数据本身是正确的)，则读取数据不确定。
如果计数停止输入和捕捉触发输入相冲突，则捕捉数据不确定。
 3. TM00 运行时 CR010 可以重写。详细信息请参看图 6-20 的注意事项 2。

(4) 16 位定时器捕捉/比较寄存器 001 (CR001)^注

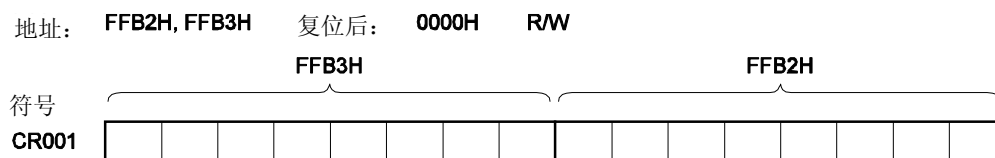
CR001 是一个 16 位比较寄存器。

CR001 可通过 16 位存储器操作指令来设置。

复位后此寄存器为 0000H。

注 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D。

图 6-6. 16 位定时器捕捉/比较寄存器 001 (CR001) 的格式



CR001 的设置值与 16 位定时器计数器 01 (TM01) 的计数值频繁比较, 如果相等则产生中断请求 (INTTM001)。直到重写 CR001 前, 这个值一直被保存。

- 注意事项
1. 如果在自由运行模式下 CR001 被清零为 0000H, 则当 TM01 溢出 (FFFFH) 后 CR001 从 0000H 变为 0001H 时产生一个中断请求 (INTTM001)。
 2. 在 TM01 运行时, 不能重写 CR001。
 3. CR001 不能作为捕捉寄存器使用。

(5) 16 位定时器捕捉/比较寄存器 011 (CR011)^注

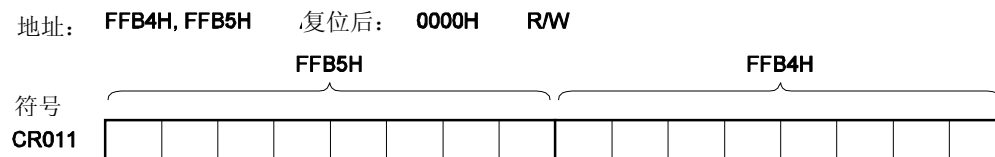
CR011 是一个 16 位比较寄存器。

CR011 可通过 16 位存储器操作指令来设置。

复位后此寄存器为 0000H。

注 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D。

图 6-7. 16 位定时器捕捉/比较寄存器 011 (CR011) 的格式



CR011 的设置值与 16 位定时器计数器 01 (TM01) 的计数值频繁比较, 如果相等则产生中断请求 (INTTM011)。直到重写 CR011 前, 这个值一直被保存。

- 注意事项
1. 如果在自由运行模式下 CR011 被清零为 0000H, 则当 TM01 溢出 (FFFFH) 后 CR011 从 0000H 变为 0001H 时产生一个中断请求 (INTTM011)。
 2. 在 TM01 运行时, 可以重写 CR011。详细信息请参看图 6-20 的注意事项 2。
 3. CR011 不能作为捕捉寄存器使用。

6.3 控制 16 位定时器/事件计数器 00 和 01 的寄存器

控制 16 位定时器/事件计数器 00 和 01 的 6 个寄存器如下。

- 16 位定时器模式控制寄存器 0n (TMC0n)
- 捕捉/比较控制寄存器 0n (CRC00)
- 16 位定时器输出控制寄存器 0n (TOC00)
- 预分频器模式寄存器 0n (PRM0n)
- 端口模式寄存器 0 (PM0)
- 端口寄存器 0 (P0)

(1) 16 位定时器模式控制寄存器 0n (TMC0n)

TMC0n 是一个 8 位寄存器，用于设置 16 位定时器操作模式、16 位定时器计数器 0n (TM0n) 清零模式和输出定时，及溢出检测。

可以由 1 位或 8 位存储器操作指令设置。

复位信号的产生将 TMC0n 清零(00H)。

注意事项 当设置 TMC0n2 与 TMC0n3 为 0, 0 (操作停止模式) 以外的值时，16 位定时器计数器 0n (TM0n) 开始计数。若要停止操作，则设置 TMC0n2 和 TMC0n3 为 0, 0。

备注 n = 0: μ PD78F0372, 78F0373, 78F0382 和 78F0383

n = 0, 1: μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D

图 6-8. 16 位定时器模式控制寄存器 00 (TMC00) 的格式

地址: FFBAH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	TMC001	操作模式和清零模式选择	TO00 反转时钟选择	产生中断请求
0	0	0	操作停止 (TM00 清零)	不改变	不产生
0	0	1			
0	1	0	自由运行模式	TM00 与 CR000 相等或者 TM00 与 CR010 相等	<当作为比较寄存器使用时> TM00 与 CR000 相等或者 TM00 与 CR010 相等时产生 <当作为捕捉寄存器使用时> 通过输入 CR000 捕捉触发来产生
0	1	1		TM00 与 CR000 相等或者 TM00 与 CR010 相等, 或者 TI000 有效沿	
1	0	0	清零 & 启动发生在 TI000 引脚 有效沿	-	
1	0	1			
1	1	0	清零 & 启动发生在 TM00 与 CR000 相等时	TM00 与 CR000 相等或者 TM00 与 CR010 相等	
1	1	1		TM00 与 CR000 相等或者 TM00 与 CR010 相等, 或者 TI000 有效沿	

OVF00	16 位定时器计数器 00 (TM00) 溢出检测
0	溢出不检测
1	溢出检测

- 注意事项**
1. 在写不同于 OVF00 的标志位以前, 定时器操作要停止。
 2. 使用预分频模式寄存器 00 (PRM00) 来设置 TI000 引脚的有效沿。
 3. 如果有下列任何模式: 可选择在 TM00 和 CR000 相等时产生的清零&启动模式, 在 TI000 引脚有效沿产生的清零&启动模式, 或自由运行模式, 当 CR000 的设置值为 FFFFH 并且 TM00 的值从 FFFFH 变化到 0000H 时, OVF00 标志被设置为 1。

备注

TO00: 16 位定时器/事件计数器 00 输出引脚
 TI000: 16 位定时器/事件计数器 00 输入引脚
 TM00: 16 位定时器计数器 00
 CR000: 16 位定时器捕捉/比较计数器 000
 CR010: 16 位定时器捕捉/比较计数器 010

图 6-9 16 位定时器模式控制寄存器 01 (TMC01)的格式

地址: FFB6H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	<0>
TMC01	0	0	0	0	TMC013	TMC012	TMC011	OVF01

TMC013	TMC012	操作模式和清零模式选择	产生中断请求
0	0	停止运行 (TM01 清零)	不产生
0	1	自由运行模式	TM01 与 CR011 相等或者 TM01 与 CR001 相等时产生
1	0	禁止设置	
1	1	清零 & 启动发生在 TM01 与 CR001 相等时	TM01 与 CR011 相等或者 TM01 与 CR001 相等时产生

OVF01	16 位定时器计数器 01 (TM01) 溢出检测
0	溢出不检测
1	溢出检测

- 注意事项**
1. 在写不同于 OVF01 的标志位以前，定时器操作要停止。
 2. 如果有下列任何模式：可选择在 TM01 和 CR001 相等时产生的清零&启动模式，或自由运行模式，当 CR001 的设置值为 FFFFH 并且 TM01 的值从 FFFFH 变化到 0000H 时，OVF01 标志被设置为 1。

备注

TM01: 16 位定时器计数器 01
 CR001: 16 位定时器捕捉/比较计数器 001
 CR011: 16 位定时器捕捉/比较计数器 011

(2) 捕捉/比较控制寄存器 00 (CRC00)

CRC00 用于控制 16 位定时器捕捉/比较寄存器 CR000 与 CR010 的操作。

可以由 1 位或 8 位存储器操作指令设置 CRC00。

复位信号的产生将 CRC00 清零(00H)。

图 6-10. 捕捉/比较控制寄存器 00 (CRC00) 的格式

地址: FFBCH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000
CRC002	CR010 操作模式的选择							
0	用作比较寄存器							
1	用作捕捉寄存器							
CRC001	CR000 捕捉触发选择							
0	在 TI010 引脚的有效沿捕捉							
1	在 TI000 引脚的有效沿反向 [*] 时捕捉							
CRC000	CR000 操作模式选择							
0	用作比较寄存器							
1	用作捕捉寄存器							

注 指定上升沿或下降沿作为 TI000 引脚的有效沿时不执行捕捉操作。

注意事项 1. 在设置 CRC00 以前，定时器必须停止操作。

2. 当用 16 位定时器模式控制寄存器 00 (TMC00) 来选择在 TM00 和 CR000 相等时产生的清零 & 启动模式，CR000 不必指定为捕捉寄存器。

3. 为确保捕捉操作正确执行，捕捉触发需要的脉冲应大于所选时钟周期的两个计数时钟，该计数时钟用预分频器模式寄存器 00 (PRM00) 选择。

(3) 16 位定时器输出控制寄存器 00 (TOC00)

本寄存器控制 16 位定时器/事件计数器 00 输出控制器的操作。它可设置/复位定时器输出 F/F(LV00)，允许/禁止输出反向和 16 位定时器/事件计数器 00 定时器输出，允许/禁止单脉冲输出操作，和通过软件设置单脉冲输出触发。

可以由 1 位或 8 位存储器操作指令设置 TOC00。

复位信号的产生将 TOC00 清零(00H)。

图 6-11. 16 位定时器输出控制寄存器 00 (TOC00) 的格式

地址: FFBDH 复位后: 00H R/W

符号	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
OSPT00	通过软件进行单脉冲输出触发							
0	无单脉冲输出							
1	单脉冲输出							
OSPE00	单脉冲输出操作控制							
0	连续脉冲输出模式							
1	单脉冲输出模式 ^注							
TOC004	CR010 与 TM00 相等时定时器输出 F/F 控制							
0	禁止反转操作							
1	允许反转操作							
LVS00	LVR00	定时器输出 F/F 状态设置						
0	0	无变化						
0	1	定时器输出 F/F 复位(0)						
1	0	定时器输出 F/F 设置(1)						
1	1	禁止设置						
TOC001	CR000 与 TM00 相等时定时器输出 F/F 控制							
0	禁止反转操作							
1	允许反转操作							
TOE00	定时器输出控制							
0	禁止输出 (输出恒为低电平)							
1	允许输出							

注 只有在自由运行模式和在 TI000 引脚有效沿产生的清零&启动模式下单脉冲输出模式操作才正确。在 TM00 和 CR000 寄存器相等时产生的清零&启动模式下，因为溢出不会发生所以单脉冲不可能输出。

- 注意事项**
1. 在设置除 TOC004 位之外的位时，定时器操作必须被停止。
 2. 读取 LVS00 和 LVR00，则只能读出 0 来。
 3. 在数据设置之后，OSPT00 会自动清零，所以只能读出 0。
 4. 除了在单脉冲输出模式下，不要设置 OSPT00 为 1。
 5. 连续写 OSPT00 时要求两个或更多的计数时钟周期的时钟间隔，该计数时钟周期通过预分频模式寄存器 00 (PRM00) 选择。
 6. 不要在设置 TOE00 以前设置 LVS00 为 1，不要将 LVS00 和 TOE00 同时设置为 1。
 7. 应按以下次序执行 <1> 和 <2>，不要同时执行。
 - <1> 设置 TOC001, TOC004, TOE00, OSPE00: 设置定时器输出操作
 - <2> 设置 LVS00, LVR00: 设置定时器输出 F/F

(4) 预分频器模式寄存器 0n (PRM0n)

PRM0n 用于设置 16 位定时器计数器 0n(TM0n)的计数时钟以及 TI000 与 TI010 引脚输入的有效沿。

可由 1 位或 8 位存储器操作指令设置 PRM0n。

复位信号的产生将 PRM0n 清零(00H)。

备注 n = 0: μ PD78F0372, 78F0373, 78F0382 和 78F0383

n = 0, 1: μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D

图 6-12. 预分频器模式寄存器 00 (PRM00) 的格式

地址: FFBBH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PRM00	ES101	ES100	ES001	ES000	0	0	PRM001	PRM000

ES101	ES100	TI010 引脚有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	兼有上升沿和下降沿

ES001	ES000	TI000 引脚有效沿选择
0	0	下降沿
0	1	上升沿
1	0	禁止设置
1	1	兼有上升沿和下降沿

PRM001	PRM000	计数时钟选择				
			$f_{PRS} = 2 \text{ MHz}$	$f_{PRS} = 5 \text{ MHz}$	$f_{PRS} = 10 \text{ MHz}$	$f_{PRS} = 20 \text{ MHz}$
0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz	78.12 kHz
1	1	TI000 有效沿 ^{**}				

注 外部时钟脉冲宽度要求大于内部时钟(f_{PRS})的两个周期宽度。

- 注意事项**
1. 在停止定时器操作后总是设置数据到 **PRM00**。
 2. 如果 **TI000** 引脚有效沿被设置用于计数时钟，不要使用 **TI000** 引脚有效沿和捕捉触发器设置清零&启动模式。
 3. 如果系统复位后 **TI000** 或 **TI010** 立即为高电平，**TI000** 或 **TI010** 引脚的有效沿为上升沿或兼有两种脉冲沿时可立即检测到这个上升沿并允许 **16 位定时器/事件计数器 00(TM00)** 的操作。当 **TI000** 或 **TI010** 引脚被上拉时要注意。然而一旦定时器操作已经被停止然后再次允许时则不检测上升沿。
 4. 当使用 **TI010** 引脚的有效沿，**P01** 不能用做定时器输出 (**TO00**)，并且当 **TO00** 引脚使用时，**TI010** 引脚不能使用有效沿。

备注 f_{PRS} : 外围硬件时钟振荡频率

图 6-13. 预分频器模式寄存器 01 (PRM01) 的格式

地址: FFB7H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
PRM01	0	0	0	0	0	0	PRM011	PRM010

PRM011	PRM010		计数时钟选择			
			$f_{PRS} = 2 \text{ MHz}$	$f_{PRS} = 5 \text{ MHz}$	$f_{PRS} = 10 \text{ MHz}$	$f_{PRS} = 20 \text{ MHz}$
0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	1	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	$f_{PRS}/2^6$	31.25 kHz	78.125 kHz	156.25 kHz	312.5 kHz
1	1	禁止设置				

注意事项 在停止定时器操作后总是设置数据到 **PRM01**。

备注 f_{PRS} : 外围硬件时钟振荡频率

(5) 端口模式寄存器 0 (PM0)

该寄存器按位设置端口 0 为输入/输出模式。

如果使用 P01/TO00/TI010 引脚用于定时器输出，则需要将 PM01 以及 P01 的输出锁存器清零。

如果使用 P00/TI000 和 P01/TO00/TI010 引脚用于定时器输入，则需要将 PM00 和 PM01 设置为 1，此时 P00 和 P01 的输出锁存器可以为 0 也可以为 1。

可由 1 位或 8 位存储器操作指令设置 PM0。

复位信号的产生将 PM0 设置为 FFH。

图 6-14. 端口模式寄存器 0 (PM0) 的格式

地址: **FF20H** 复位后: **FFH** R/W

符号 7 6 5 4 3 2 1 0

PM0	1	1	1	1	PM03	PM02	PM01	PM00
-----	---	---	---	---	------	------	------	------

PM0n	P0n 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 复位释放后，一定要设置第 2 位和第 3 位为 0。

6.4 16 位定时器/事件计数器 00 和 01 的操作

6.4.1 间隔定时器的操作

通过图 6-15 设置 16 位定时器模式控制寄存器 0n (TMC0n) 和捕捉/比较控制寄存器 00 (CRC00)，可设置间隔定时器操作。

设置

如下为基本操作步骤。

- <1> 当使用定时器 00 时设置 CRC00 寄存器 (参看 图 6-15 设置值)。
- <2> 设置任意值到 CR00n 寄存器。
- <3> 通过使用 PRM0n 寄存器设置计数时钟。
- <4> 设置 TMC0n 寄存器以开始操作 (参看 图 6-15 设置值)。

注意事项 在 TM0n 操作期间，CR00n 不能被重写。

备注 对于怎样允许 INTTM00n 中断，参看 第十九章 中断功能。

使用事先使用 16 位定时器捕捉/比较寄存器 00n(CR00n) 设置的计数值作为时间间隔，中断请求可频繁产生。

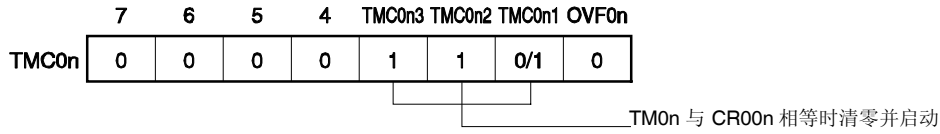
当 16 位定时器计数器 0n (TM0n) 的计数值和 CR00n 的值相等时，则 TM0n 清零并继续计数运行，并产生中断请求信号(INTTM00n)。

16 位定时器/事件计数器 0n 的计数时钟可以用预分频模式寄存器 0n (PRM0n) 的第 0 位和第 1 位(PRM0n0, PRM0n1) 来选择。

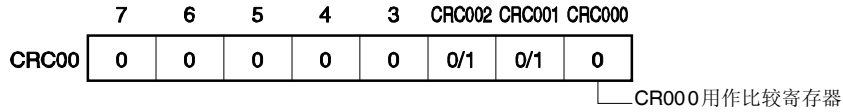
备注 n = 0: μ PD78F0372, 78F0373, 78F0382 和 78F0383
n = 0, 1: μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D

图 6-15. 间隔定时器操作的控制寄存器设置

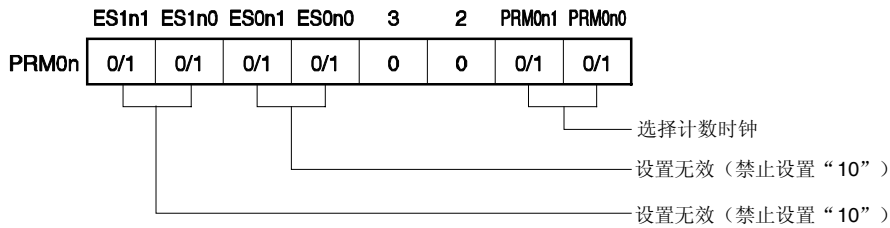
(a) 16 位定时器模式控制寄存器 0n (TMC0n)



(b) 捕捉/比较控制寄存器 00 (CRC00)

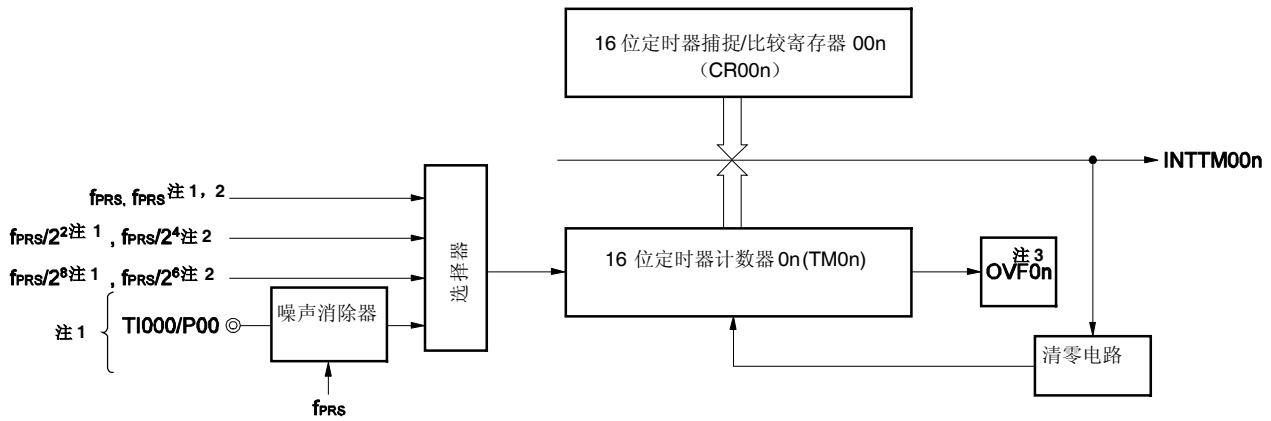


(c) 预分频器模式寄存器 0n (PRM0n)



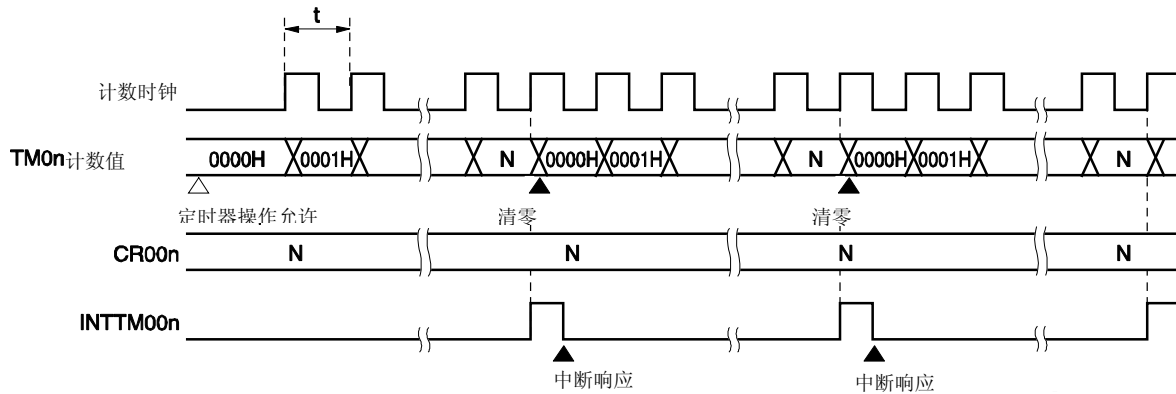
- 备注**
1. 0/1: 间隔定时器可以通过设置 0 或 1 来允许其它功能同时使用。详细情况请参看各自控制寄存器的描述。
 2. n = 0: μ PD78F0372, 78F0373, 78F0382 和 78F0383
 n = 0, 1: μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D

图 6-16. 间隔定时器配置框图



- 注
1. 当使用定时器 00。
 2. 当使用定时器 01。
 3. 仅当 16 位定时器捕捉/比较寄存器 00n (CR00n) 被设置为 FFFFH 时，将 OVF0n 设置为 1。

图 6-17. 间隔定时器操作的时序



- 备注
- 间隔定时器 = $(N + 1) \times t$
- $N = 0001H \sim FFFFH$ (可设置范围)
- $n = 0$: μ PD78F0372, 78F0373, 78F0382 和 78F0383
- $n = 0, 1$: μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D

6.4.2 PPG 输出操作 (仅限于定时器 00)

按图 6-18 设置 16 位定时模式控制寄存器 00 (TMC00) 和捕捉/比较控制寄存器 00 (CRC00)，允许作为 PPG(可编程脉冲发生器) 输出操作。

设置

如下为基本操作步骤。

- <1> 设置 CRC00 寄存器 (对于设置值参看 图 6-18)。
- <2> 设置 CR000 寄存器的任意值作为周期。
- <3> 设置 CR010 寄存器的任意值作为占空比。
- <4> 设置 TOC00 寄存器 (对于设置值参看 图 6-18)。
- <5> 通过使用 PRM00 寄存器设置计数时钟。
- <6> 设置 TMC00 寄存器开始操作 (对于设置值参看 图 6-18)。

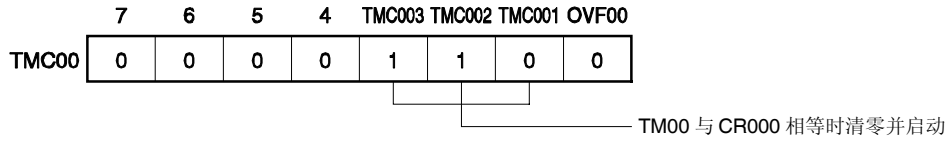
注意事项 在操作期间要改变占空比值 (CR010 寄存器的值)，请参看 图 6-20 PPG 输出操作时序的注意事项 2。

- 备注**
1. 了解 IO00 引脚的设置，可参见 6.3 (5) 端口模式寄存器 0 (PM0)。
 2. 要了解如何允许 INTTM00n 中断，参见 第十九章 中断功能。

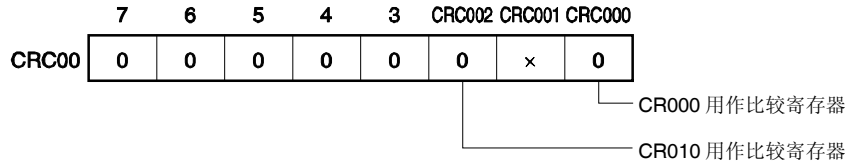
在 PPG 输出操作中，从 TO00 引脚输出的矩形波的脉宽和周期分别对应于 16 位定时器捕捉/比较寄存器 010(CR010) 和 16 位定时器捕捉/比较寄存器 000(CR000)中的预设值。

图 6-18. PPG 输出操作的控制寄存器设置

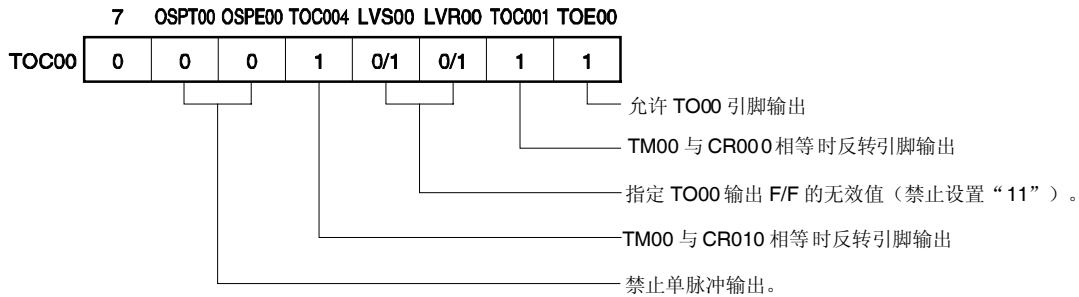
(a) 16 位定时器模式控制寄存器 00 (TMC00)



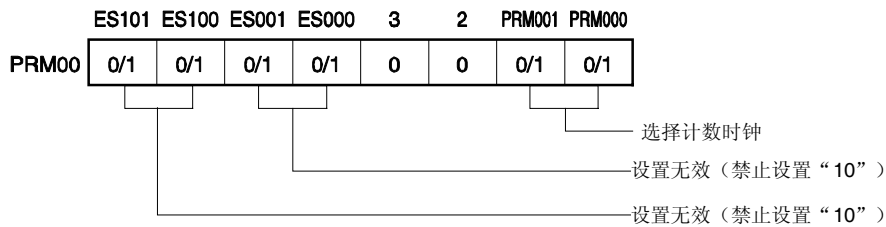
(b) 捕捉/比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)



(d) 预分频器模式寄存器 00 (PRM00)



注意事项 1. 以下为 CR000 和 CR010 的设置范围：

$$0000H \leq CR010 < CR000 \leq FFFFH$$

2. 通过 PPG 输出的脉冲周期 (CR000 设置值 + 1) 具有占空比 (CR010 设置值 + 1)/(CR000 设置值 + 1)

备注 ×: 不必考虑

图 6-19. PPG 输出的配置框图

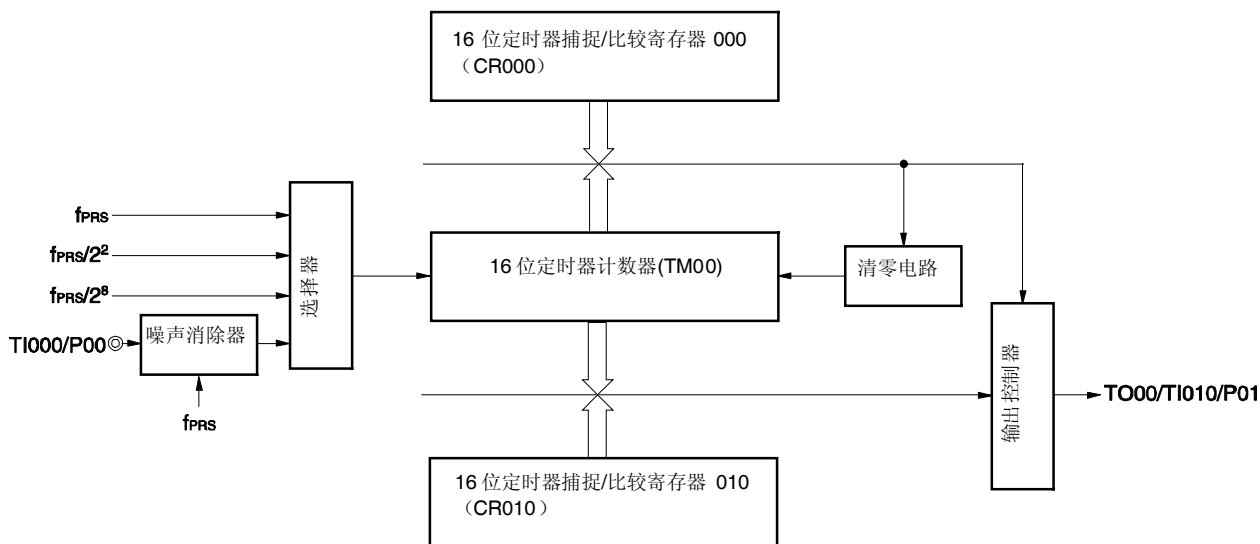
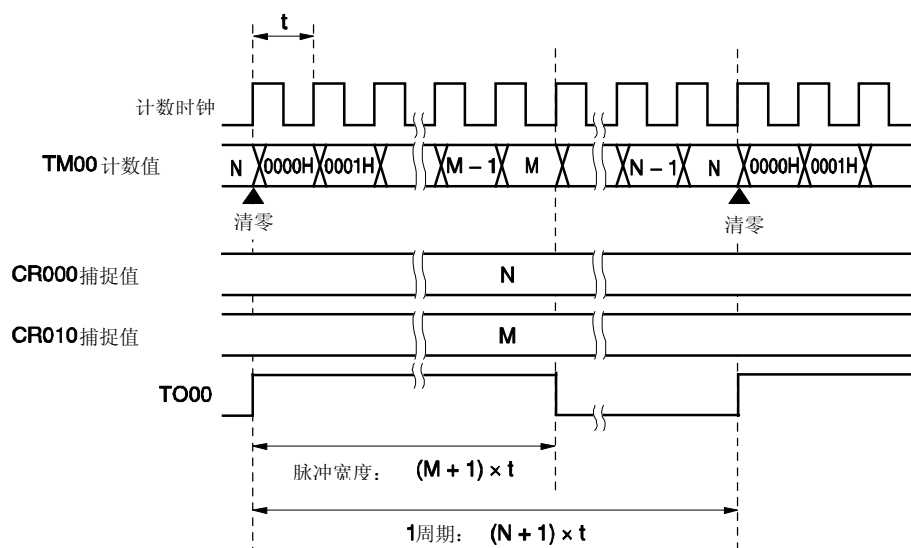


图 6-20. PPG 输出操作时序



- 注意事项
1. 当 TM00 操作时 CR000 不能够重写。
 2. 在 PPG 输出操作中，当 TM00 操作时使用以下步骤改变脉冲宽度 (重写 CR010)。
 - <1> 通过 TM00 和 CR010 相等禁止定时器输出的反向操作 (TOC004 = 0)
 - <2> 禁止 INTTM010 中断 (TMMK010 = 1)
 - <3> 重写 CR010
 - <4> 等待 TM00 计数时钟的 1 个周期
 - <5> 通过 TM00 和 CR010 相等允许定时器输出的反向操作 (TOC004 = 1)
 - <6> 将 INTTM010 (TMIF010 = 0) 中断请求标志清零
 - <7> 允许 INTTM010 中断 (TMMK010 = 0)

备注 0000H ≤ M < N ≤ FFFFH

6.4.3 脉冲宽度测量操作 (仅限于定时器 00)

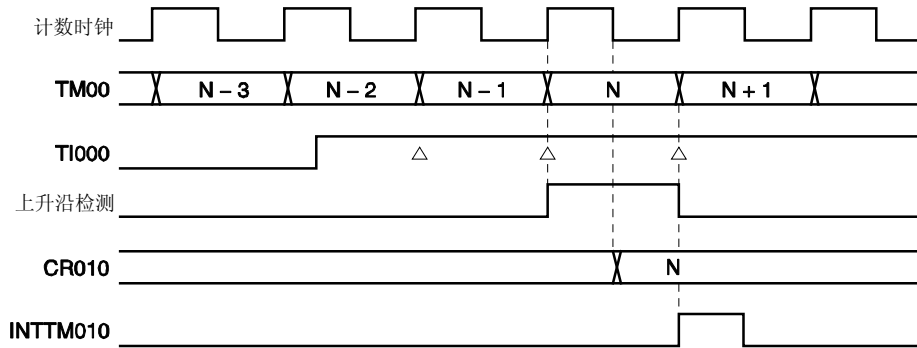
16 位定时计数器 TM00 可用于测量输入到 TI000 与 TI010 引脚的信号脉冲宽度。

有两种测量方法：在自由运行模式下使用 TM00 进行测量，以及通过同步使用 TI000 引脚输入信号的脉冲沿重新启动定时器来进行测量。

当产生中断时，读取捕捉寄存器的有效值，验证溢出标志，然后计算必须的脉冲宽度。在验证后清除溢出标志。

捕捉操作并不执行，直到在计数时钟内信号脉宽被取样，此由预分频模式寄存器 00(PRM00)选择并被 TI000 或 TI010 有效电平检测两次，如此可消除单脉冲宽度的噪音。

图 6-21. 指定上升沿的 CR010 捕捉操作



设置

如下为基本操作步骤。

- <1> 设置 CRC00 寄存器 (设置值请参看 图 6-22, 6-25, 6-27 和 6-29)。
- <2> 通过使用 PRM00 寄存器设置计数时钟。
- <3> 设置 TMC00 寄存器启动操作 (设置值参看 图 6-22, 6-25, 6-27 和 6-29)。

注意事项 要使用两个捕捉寄存器，需要设置 TI000 和 TI010 引脚。

- 备注**
1. 了解 TIO00(或 TI010)引脚的设置，可参见 6.3 (5) 端口模式寄存器 0 (PM0)。
 2. 要了解如何开启 INTTM000(或 INTTM010)中断，参见 第十九章 中断功能。

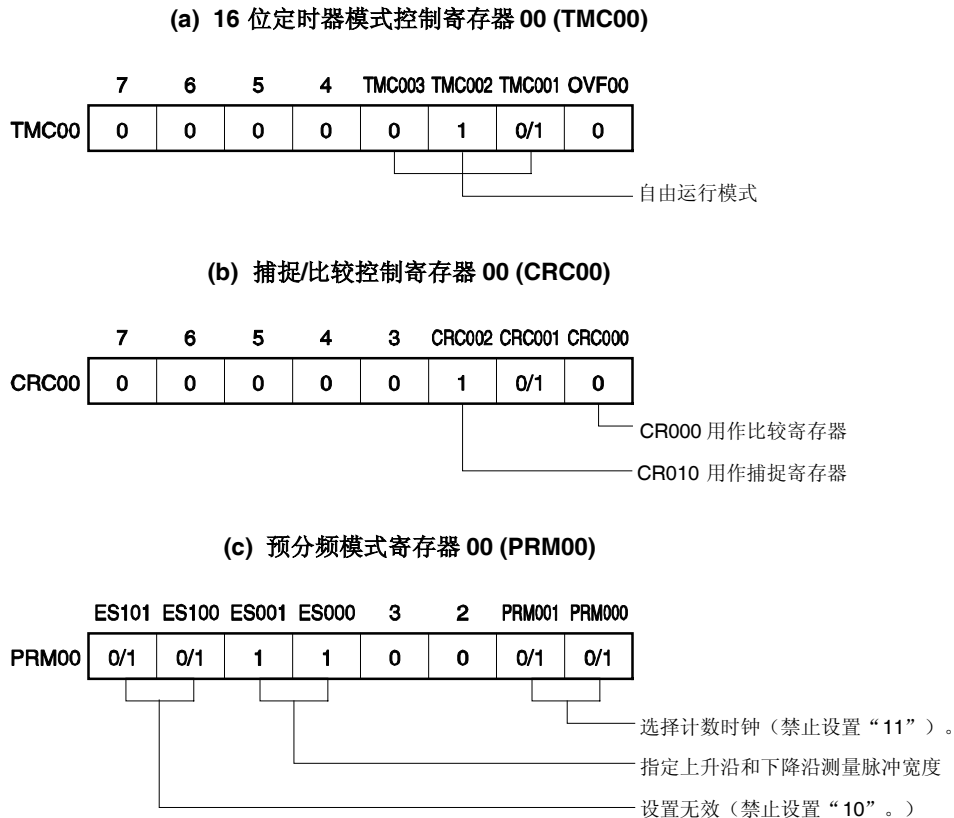
(1) 通过使用自由运行计数器和一个捕捉寄存器测量脉冲宽度

当 16 位定时器计数器 00 (TM00) 运行在自由运行模式下时，通过预分频模式寄存器 00 (PRM00) 指定的边沿输入的 TI000 引脚，TM00 的值被输入至 16 位定时器捕捉/比较寄存器 010 (CR010)，并且一个外部中断请求信号 (INTTM010) 产生。

通过使用 PRM00 的第 4 位和第 5 位 (ES000 和 ES001) 可设定 TI000 的上升和下降沿。

使用由 PRM00 选择的计数时钟进行采样，并且仅当 TI000 引脚的有效电平被检测两次才执行捕捉操作，以此来执行采样，才能消除短脉冲宽度的噪音。

图 6-22. 通过使用自由运行计数器和一个捕捉寄存器测量脉冲宽度的控制寄存器设置 (当使用 TI000 和 CR010 时)



备注 0/1: 设置 0 或 1 允许其它功能与脉宽测量同时使用。详细信息请参看各个控制寄存器的描述。

图 6-23. 自由运行计数器的测量脉冲宽度配置框图

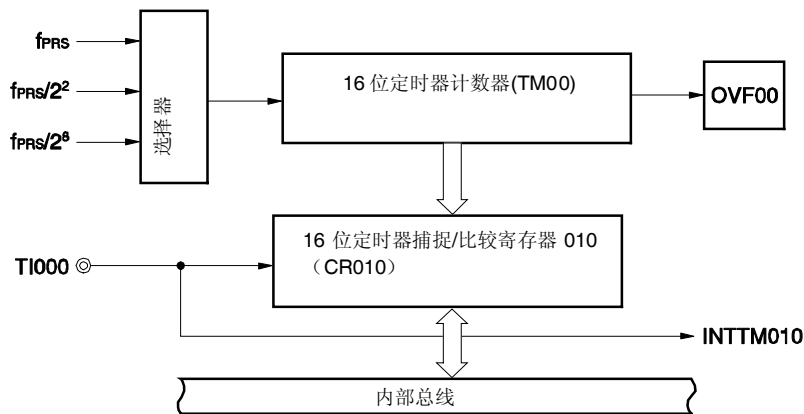
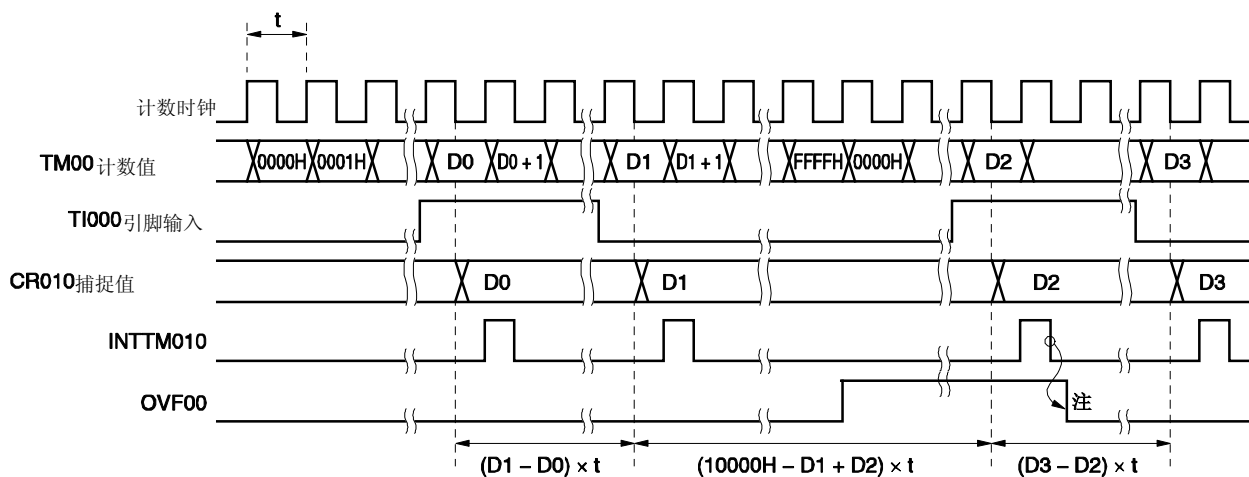


图 6-24. 自由运行计数器和一个捕捉寄存器的测量脉冲宽度操作的时序 (指定双沿)



注 通过软件清 OVF00。

(2) 通过自由运行计数器测量两个脉冲宽度

当在自由运行模式下操作 16 位定时器计数器 00 (TM00)，可以同时测量两个输入到 TI000 和 TI010 引脚的信号的脉冲宽度。

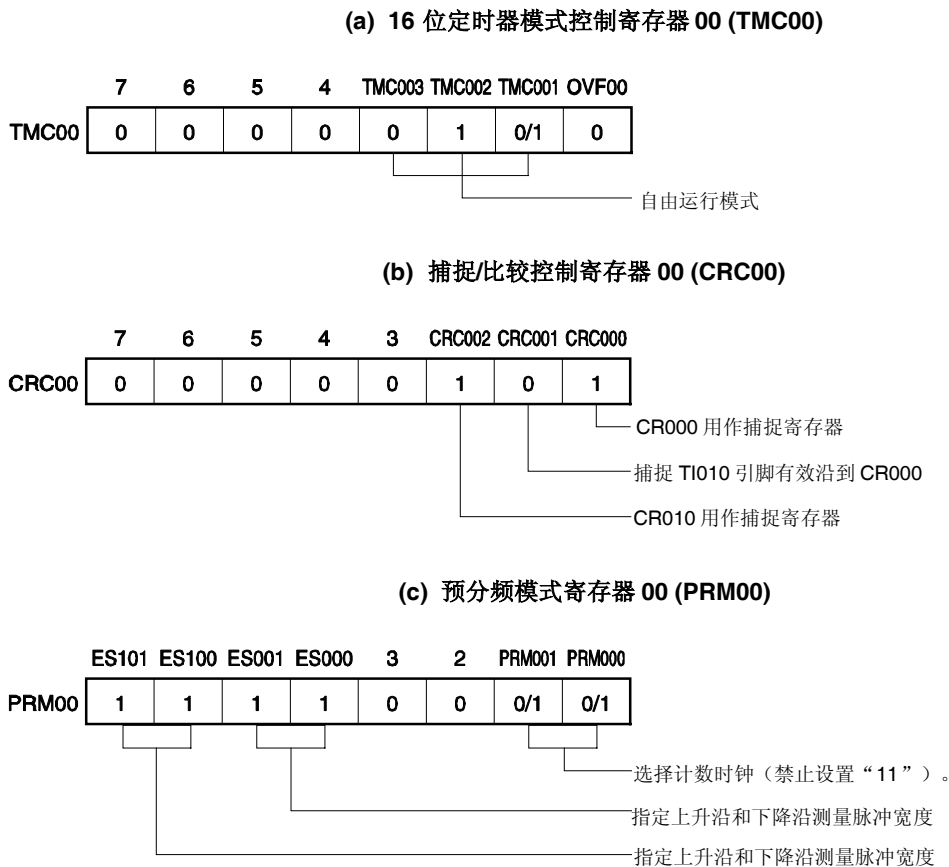
通过使用预分频模式寄存器 00 (PRM00) 的第 4 位和第 5 位 (ES000 和 ES001) 可设定 TI000 的上升和下降沿，TM00 的值输入 16 位定时器捕捉/比较寄存器 010 (CR010)，并且触发一个中断请求信号 (INTTM010)。

而且，当使用预分频模式寄存器 00 (PRM00) 的第 6 位和第 7 位 (ES100 和 ES101) 可设定 TI010 的上升和下降沿，TM00 的值输入 16 位定时器捕捉/比较寄存器 000 (CR000)，并且触发一个中断请求信号 (INTTM000)。

通过使用预分频模式寄存器 00 (PRM00) 的第 4 位和第 5 位 (ES000 和 ES001) 和第 6 位和第 7 位 (ES100 和 ES101) 指定 TI000 和 TI010 引脚的上升和下降沿。

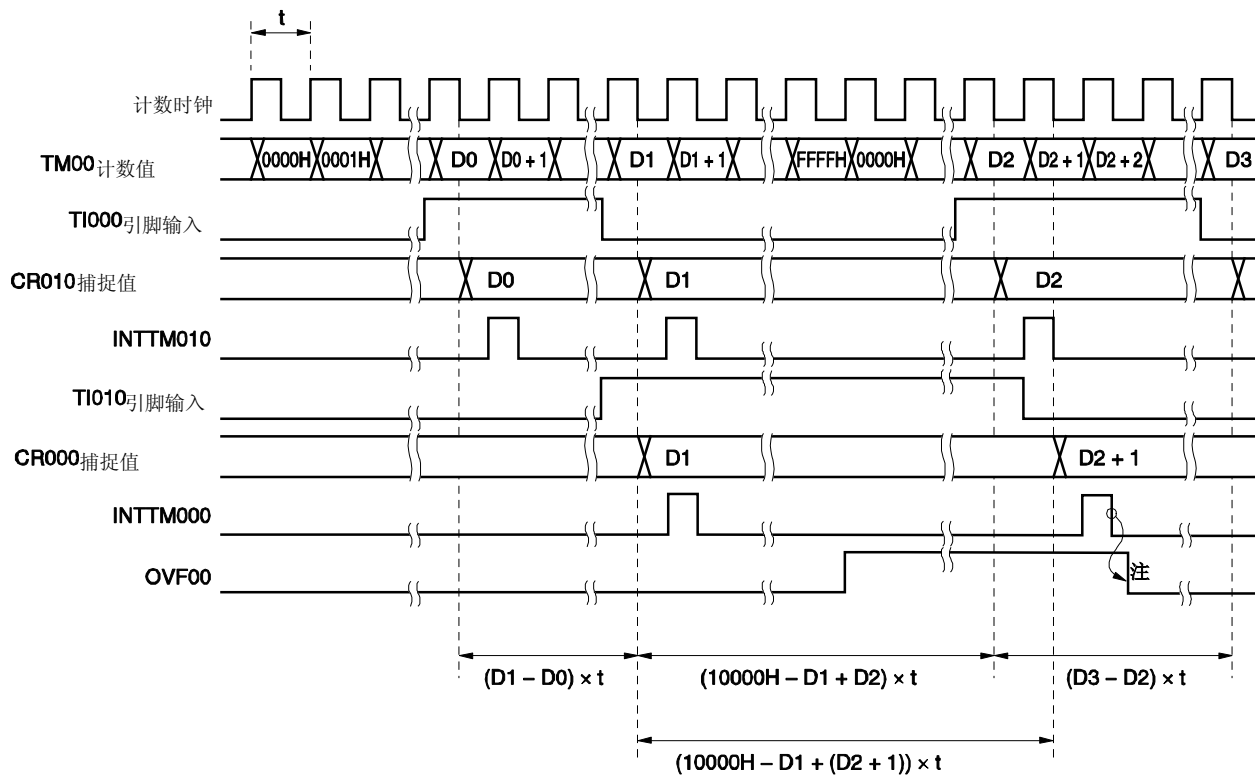
使用由 PRM00 选择的计数时钟周期，并且仅当 TI000 或 TI010 引脚的有效电平被检测两次才执行捕捉操作，以此来执行采样，才能消除短脉冲宽度的噪音。

图 6-25. 通过自由运行计数器测量两个脉冲宽度的控制寄存器设置



备注 0/1: 设置 0 或 1 允许其它功能与脉宽测量同时使用。详细信息请参看各个控制寄存器的描述。

图 6-26. 自由运行计数器测量脉冲宽度操作的时序
(指定双沿)

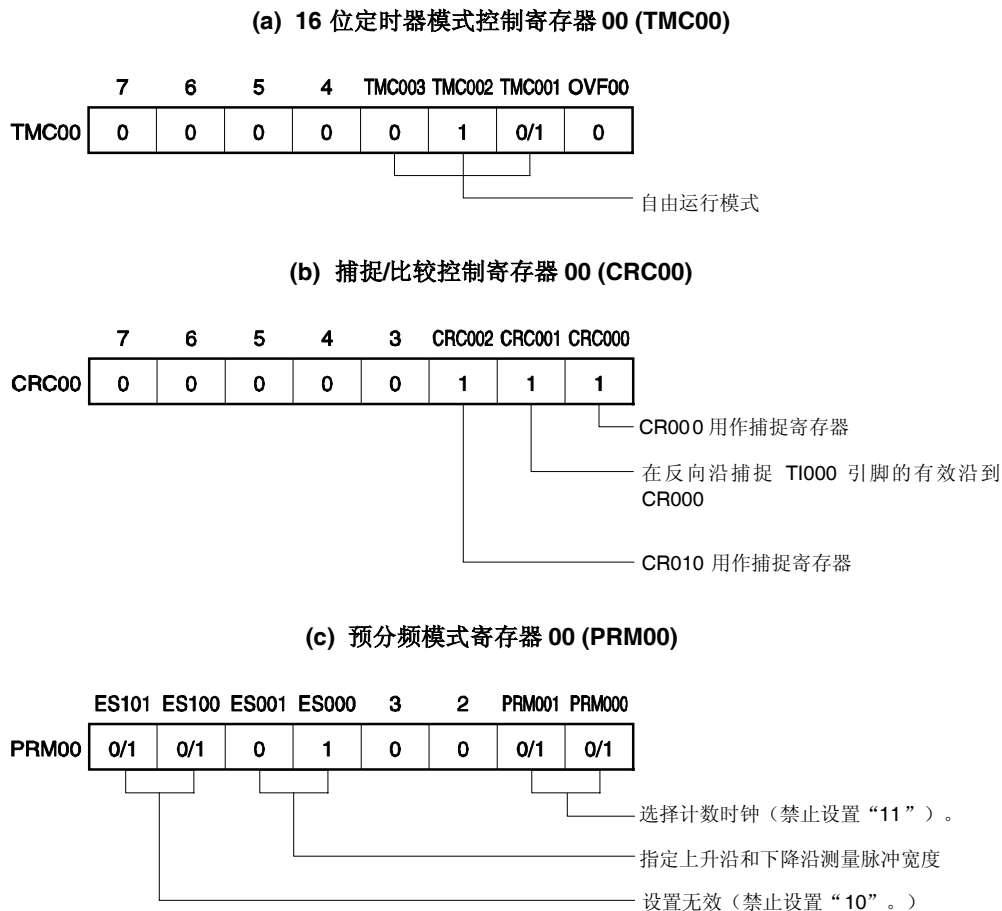


注 通过软件清 OVF00。

(3) 通过使用自由运行计数器和两个捕捉寄存器测量脉冲宽度

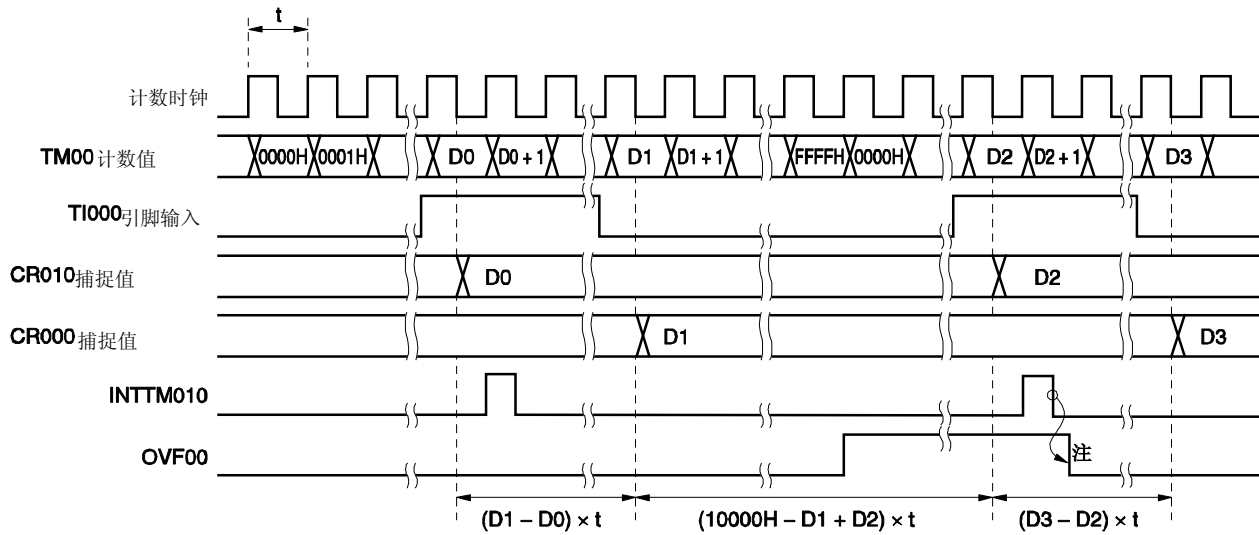
当在自由运行模式下操作 16 位定时器计数器 00 (TM00)，可以测量输入到 TI000 引脚的信号脉冲宽度。通过使用预分频模式寄存器 00 (PRM00) 的第 4 位和第 5 位 (ES000 和 ES001) 可设定 TI000 的上升或下降沿，TM00 的值输入 16 位定时器捕捉/比较寄存器 010 (CR010)，并且触发一个中断请求信号 (INTTM010)。当捕捉操作的反向沿输入到 CR010，TM00 的值输入到 16 位定时器捕捉/比较寄存器 000 (CR000) 中。使用由 PRM00 选择的计数时钟周期，并且仅当 TI000 引脚的有效电平被检测两次才执行捕捉操作，以此来执行采样，才能消除短脉冲宽度的噪音。

图 6-27. 通过使用自由运行计数器和两个捕捉寄存器测量脉冲宽度的控制寄存器的设置
(指定上升沿)



备注 0/1: 设置 0 或 1 允许其它功能与脉宽测量同时使用。详细信息请参看各个控制寄存器的描述。

图 6-28. 通过使用自由运行计数器和两个捕捉寄存器测量脉冲宽度的时序
(指定上升沿)



注 通过软件清 OVF00。

(4) 使用重启来测量脉冲宽度

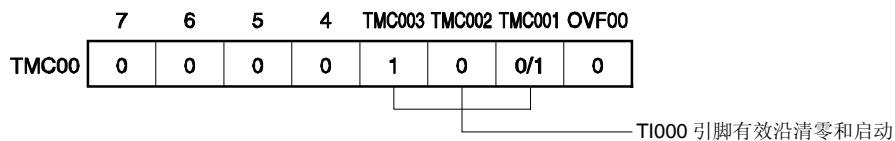
当判断 TI000 引脚的有效边沿输入，16 位定时器计数器 00 (TM00) 的计数值输入到 16 位定时器捕捉/比较寄存器 010 (CR010) 中，并且通过清零 TM00 和重启计数操作来测量输入到 TI000 引脚信号的脉冲宽度。

两个边沿中的任何一个 — 上升沿或下降沿 — 能够使用预分频模式寄存器 00 (PRM00) 的第 4 位和第 5 位 (ES000 和 ES001) 来选择。

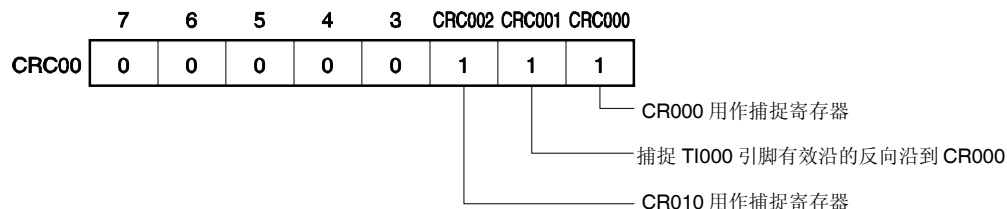
使用由 PRM00 选择的计数时钟周期，并且仅当 TI000 引脚的有效电平被检测两次才执行捕捉操作，以此来执行采样，才能消除短脉冲宽度的噪音。

图 6-29. 使用重启来测量脉冲宽度的控制寄存器设置
(指定上升沿)

(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉/比较控制寄存器 00 (CRC00)



(c) 预分频模式寄存器 00 (PRM00)

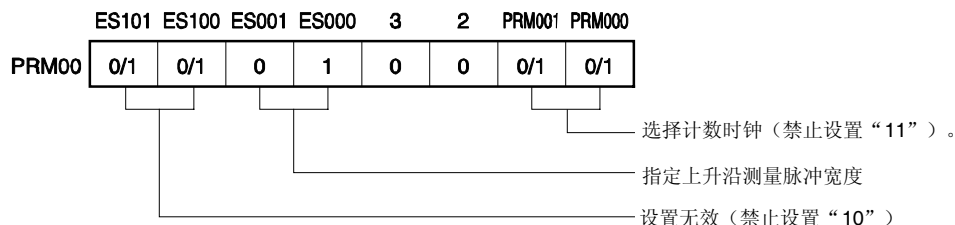
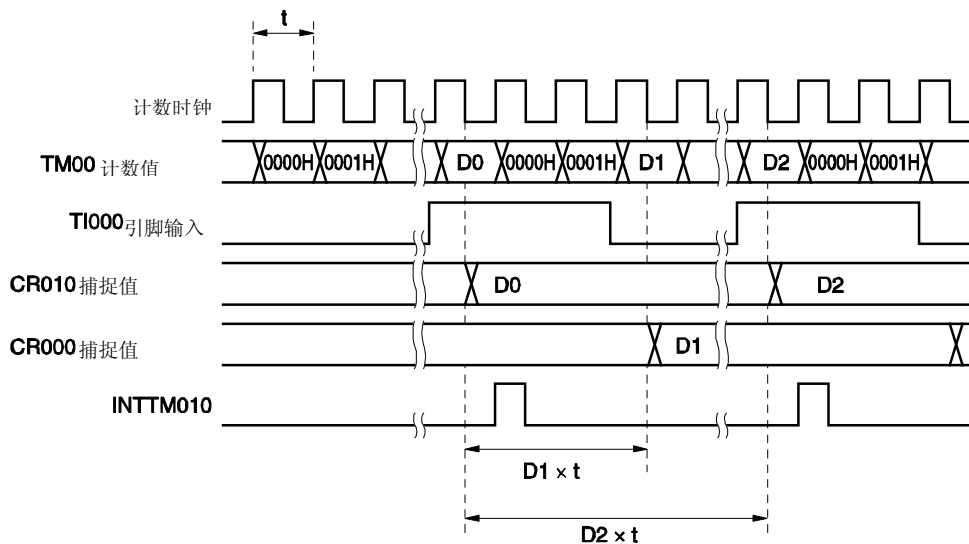


图 6-30. 使用重启来测量脉冲宽度的的时序
(指定上升沿)



6.4.4 外部事件计数器的操作 (仅限于定时器 00)

设置

如下为基本操作步骤。

- <1> 设置 CRC00 寄存器 (设置值参看 图 6-31)。
- <2> 通过使用 PRM00 寄存器设置计数时钟。
- <3> 对 CR000 设置任意值 (0000H 不能被设置)。
- <4> 设置 TMC00 以启动操作 (设置值参看 图 6-31)。

- 备注**
- 1. 了解 TIO00 引脚的设置, 可参见 6.3 (5) 端口模式寄存器 0 (PM0)。
 - 2. 要了解如何允许 INTTM000 中断, 参见 第十九章 中断功能。

使用 16 位定时器计数器 00 (TM00) 对外部时钟脉冲输入到 TIO00 引脚的脉冲数量进行计数。

由预分频模式寄存器 00 (PRM00) 指定的有效沿输入, 则 TM00 逐次记录。

当 TM00 计数值与 16 为定时器捕捉/比较寄存器 000 (CR000) 的值相等时, TM00 清零, 产生中断请求信号 (INTTM000)。

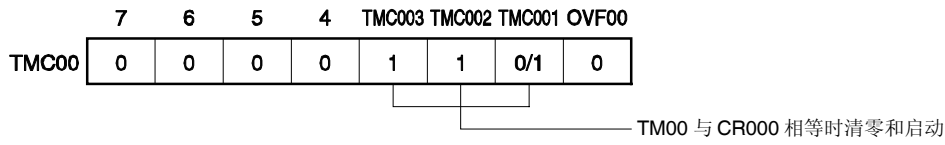
输入 CR000 的值不能为 0000H (不能执行出 1 位脉冲的计数操作)。

三种边沿中的任意一种 — 上升, 下降, 或者双沿 — 可以通过使用预分频模式寄存器 00 (PRM00) 的第 4 位和第 5 位 (ES000 和 ES001) 来选择。

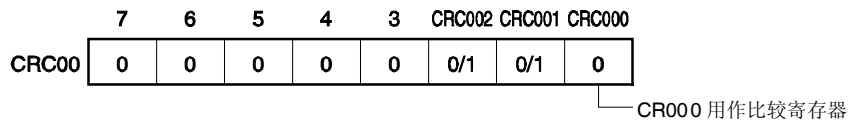
使用内部时钟 (fPRS), 并且仅当 TIO00 引脚的有效电平被检测两次才执行捕捉操作, 以此来执行采样, 才能消除短脉冲宽度的噪音。

图 6-31. 外部事件计数器模式的控制寄存器设置
(指定上升沿)

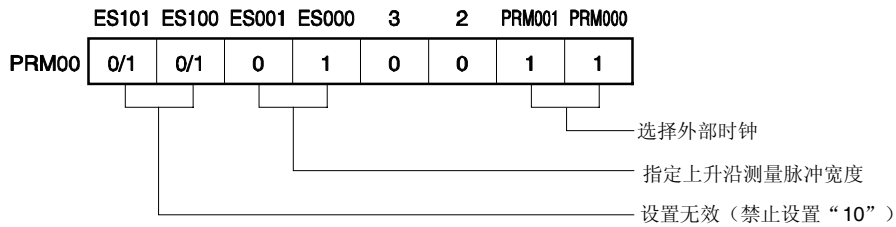
(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉/比较控制寄存器 00 (CRC00)

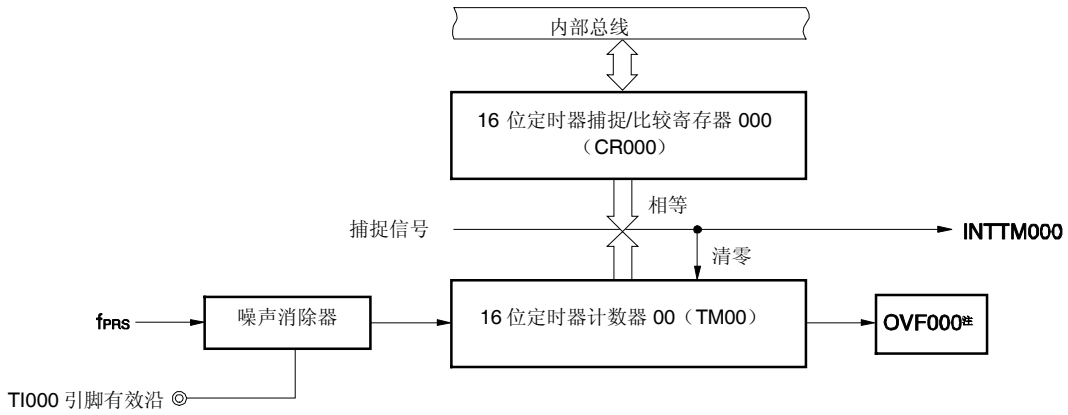


(c) 预分频模式寄存器 00 (PRM00)



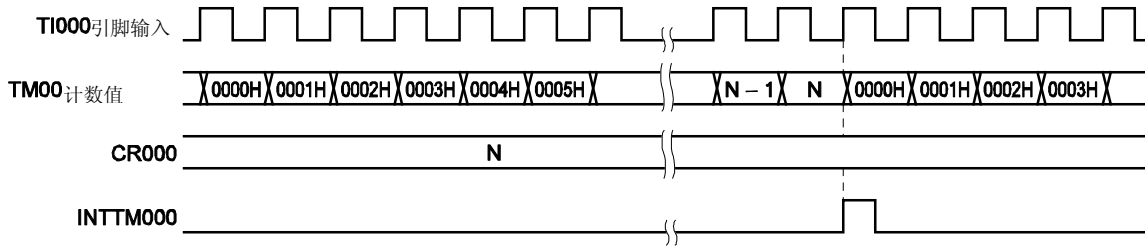
备注 0/1: 设置 0 或 1 允许其它功能与外部事件计数器同时使用。详细信息请参看各个控制寄存器的描述。

图 6-32. 外部事件计数器的配置框图



注 仅当 CR000 设置为 FFFFH 时 OVF00 被设置为 1。

图 6-33. 外部事件计数器操作时序 (指定上升沿)



注意事项 当读取外部时间计数器计数值时，TM00 可以被读取。

6.4.5 方波输出操作 (仅限于定时器 00)

设置

如下为基本操作步骤。

- <1> 通过使用 PRM00 寄存器设置计数时钟。
- <2> 设置 CRC00 寄存器 (设置值参看 图 6-34)。
- <3> 设置 TOC00 寄存器 (设置值参看 图 6-34)。
- <4> 设置任意值到 CR000 寄存器 (0000H 除外)。
- <5> 设置 TMC00 以启动操作 (设置值参看 图 6-34)。

注意事项 在 TM00 操作过程中，CR000 不能被重写。

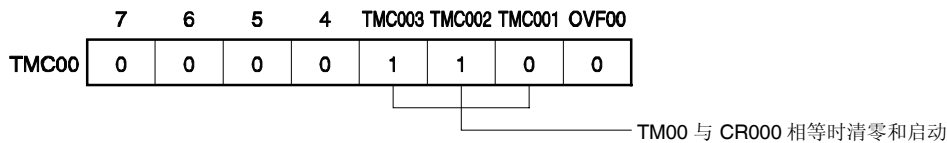
- 备注**
- 1. 了解 TO00 引脚的设置，可参见 6.3 (5) 端口模式寄存器 0 (PM0)。
 - 2. 要了解如何允许 INTTM000 中断，参见 第十九章 中断功能。

通过预先设置 16 位定时器捕捉/比较寄存器 000 (CR000) 的计数值，可控制间隔时间，以输出任意频率的方波。

通过设置 16 位定时器输出控制寄存器 00 (TOC00) 的第 0 位 (TOE00) 和第 1 位 (TOC001) 预先设置计数值 CR000 + 1 作为间隔判断，使 TO00 引脚输出状态反转。如此可以设置任意频率的方波输出。

图 6-34. 方波输出模式的控制寄存器设置 (1/2)

(a) 16 位定时器模式控制寄存器 00 (TMC00)



(b) 捕捉/比较控制寄存器 00 (CRC00)

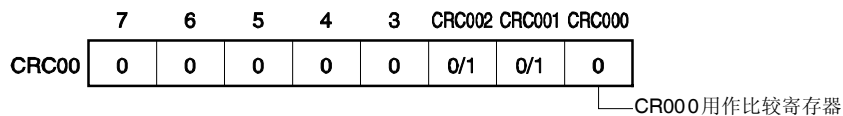
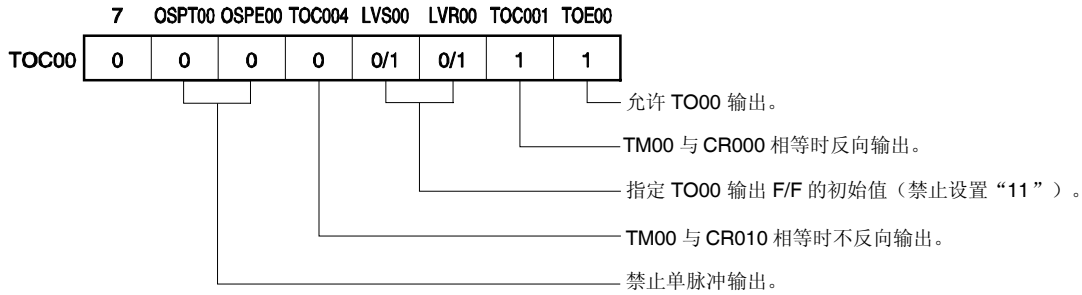
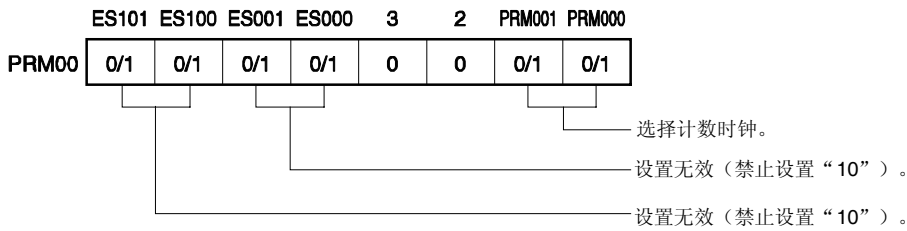


图 6-34. 方波输出模式的控制寄存器设置(2/2)

(c) 16 位定时器输出控制寄存器 00 (TOC00)

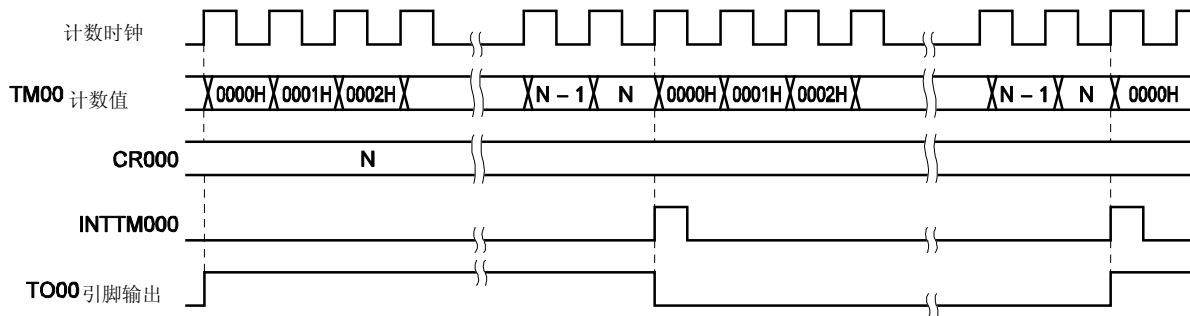


(d) 预分频模式寄存器 00 (PRM00)



备注 0/1: 设置 0 或 1 允许其它功能与方波输出同时使用。详细信息请参看各个控制寄存器的描述。

图 6-35. 方波输出操作时序



6.4.6 单脉冲输出操作 (仅限于定时器 00)

16 位定时器/事件计数器 00 能够和软件触发器或外部触发器 (TI000 引脚输入) 同步输出单脉冲。

设置

如下为基本操作步骤。

- <1> 通过使用 PRM00 寄存器设置计数时钟。
- <2> 设置 CRC00 寄存器 (设置值参看图 6-36 和 6-38)。
- <3> 设置 TOC00 寄存器 (设置值参看图 6-36 和 6-38)。
- <4> 设置任意值到 CR000 和 CR010 寄存器 (0000H 除外)。
- <5> 设置 TMC00 以启动操作 (设置值参看图 6-36 和 6-38)。

- 备注**
1. 了解 TO00 引脚的设置, 可参见 6.3 (5) 端口模式寄存器 0 (PM0)。
 2. 要了解如何允许 INTTM000 (如果必要, INTTM010) 中断, 参见第十九章 中断功能。

(1) 软件触发的单脉冲输出

通过软件设置 16 位定时器模式控制寄存器 00 (TMC00)、捕捉/比较控制寄存器 00 (CRC00)、16 位定时器输出控制寄存器 00 (TOC00), 如图 6-36, 并设置 TOC00 寄存器的第 6 位 (OSPT00) 为 1, 来使 TO00 引脚输出一个单脉冲。

通过设置 OSPT00 位为 1, 16 位定时器/事件计数器 00 被清零和启动, 并且在计数值 (N) 时输出变为有效, 该计数值 (N) 由 16 位定时器捕捉/比较寄存器 010 (CR010) 设置。在这之后, 其输出在计数值 (M) 时变得无效, 该计数值 (M) 由 16 位定时器捕捉/比较寄存器 000 (CR000) 设置。

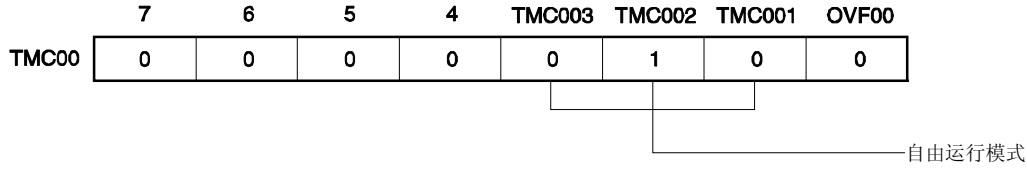
即使单脉冲已经被输出, TM00 寄存器仍然继续运行。要停止 TM00 寄存器, TMC00 寄存器的 TMC003 和 TMC002 位必须被设置为 00。

注 N < M 的示例在描述于此。当 N > M 时, 根据 CR000 寄存器输出变为有效并且根据 CR010 寄存器输出变为无效。不要将 N 设置给 M。

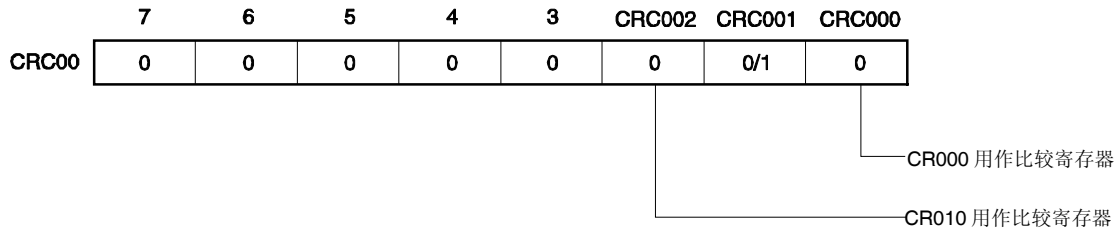
- 注意事项**
1. 当单脉冲正在输出时不要再次设置 OSPT00 位为 1。为了再次输出单脉冲, 等待直到当前的单脉冲输出已完成。
 2. 当使用软件触发 16 位定时器/事件计数器 00 的单脉冲输出时, 不要改变 TI000 引脚或其复用功能端口引脚的电平。
因为在此示例中外部触发也是有效的, 则在 TI000 引脚或其复用功能端口引脚的电平有效时定时器被清零并启动, 这会导致以不期望的时序输出脉冲。

图 6-36. 软件触发的单脉冲输出控制寄存器的设置

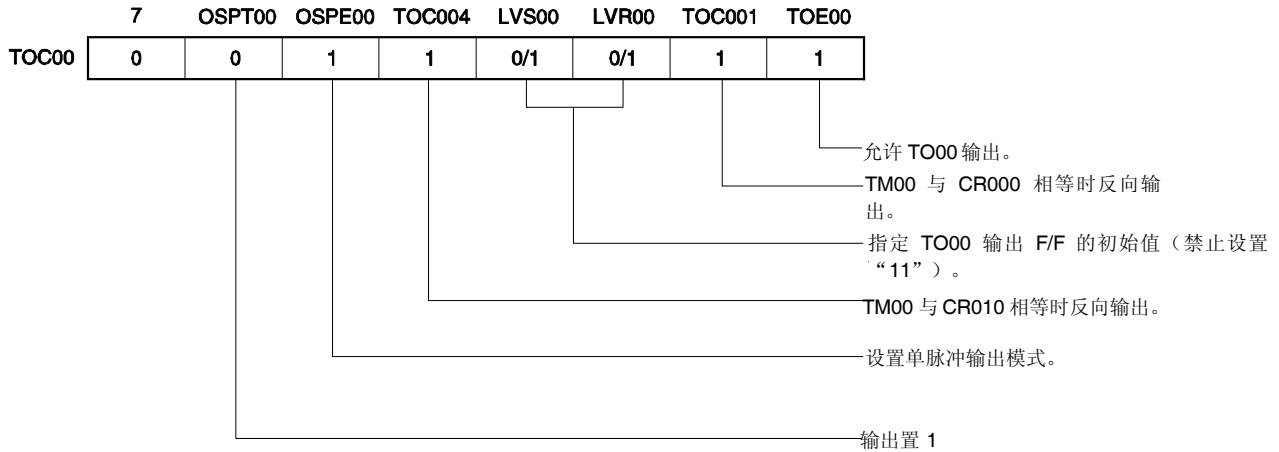
(a) 16 位定时器模式控制寄存器 00 (TMC00)



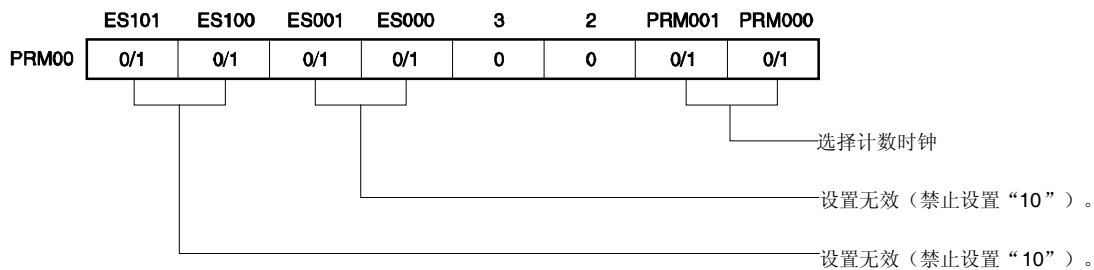
(b) 捕捉/比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)

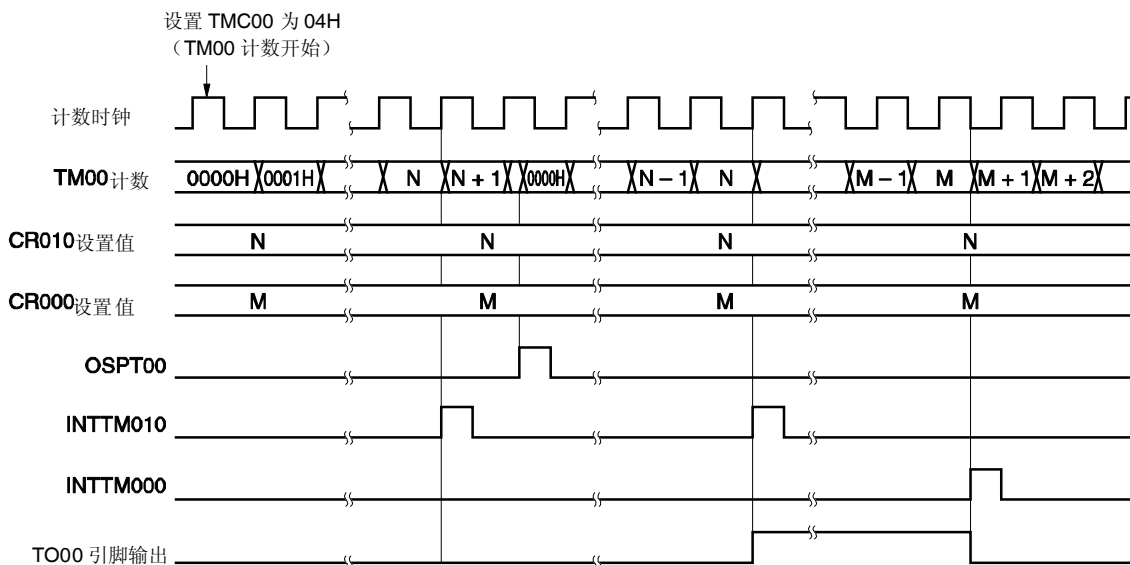


(d) 预分频模式寄存器 00 (PRM00)



注意事项 不要对 CR000 和 CR010 寄存器设置 0000H。

图 6-37. 软件触发的单脉冲输出操作的时序



注意事项 只要给 TMC003 和 TMC002 位的设置值不为 00（操作停止模式）时，16 位定时器计数器 00 开始操作。

备注 $N < M$

(2) 外部触发的单脉冲输出

通过设置 16 位定时器模式控制寄存器 00（TMC00）、捕捉/比较控制寄存器 00（CRC00）、16 位定时器输出控制寄存器 00（TOC00），如图 6-38，并且使用 TI000 引脚的有效边沿作为外部触发器，来使 TO00 引脚输出一个单脉冲。

通过预分频模式寄存器 00（PRM00）的第 4 位和第 5 位（ES000, ES001）指定 TI000 引脚的有效沿。可指定上升沿，下降沿，或兼有上升沿和下降沿。

当检测到 TI000 引脚的有效沿时，16 位定时器/事件计数器被清零并启动，在预先设置在 16 位定时器捕捉/比较寄存器 010（CR010）中的计数值处输出变为有效。在这之后，在预先设置在 16 位定时器捕捉/比较寄存器 000（CR000）处输出变为无效^注。

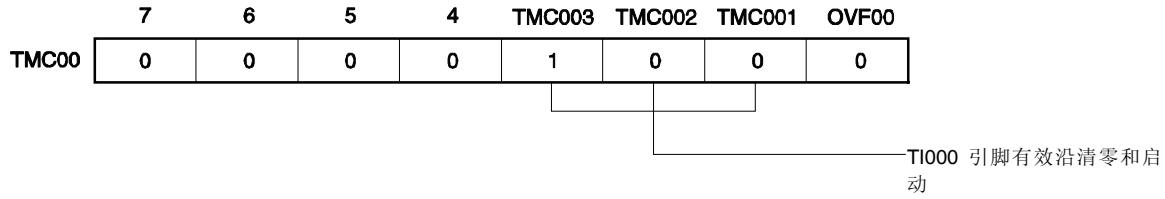
注 $N < M$ 的示例在描述于此。当 $N > M$ 时，根据 CR000 寄存器输出变为有效并且根据 CR010 寄存器输出变为无效。不要将 N 设置给 M。

注意事项 当单脉冲正在输出时不要再次输入外部触发。

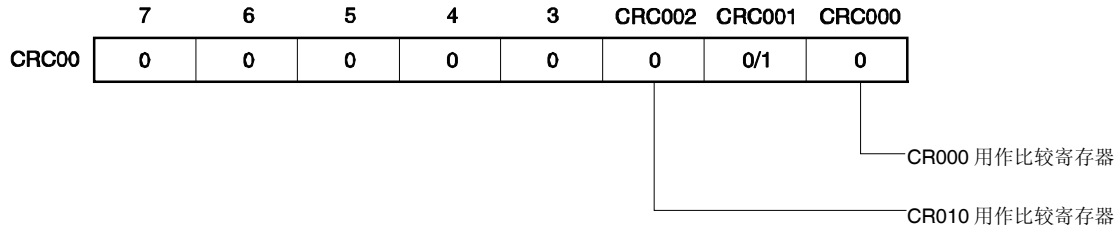
为了再次输出单脉冲，等待直到当前的单脉冲输出完成。

图 6-38. 外部触发的单脉冲输出控制寄存器的设置
(指定上升沿)

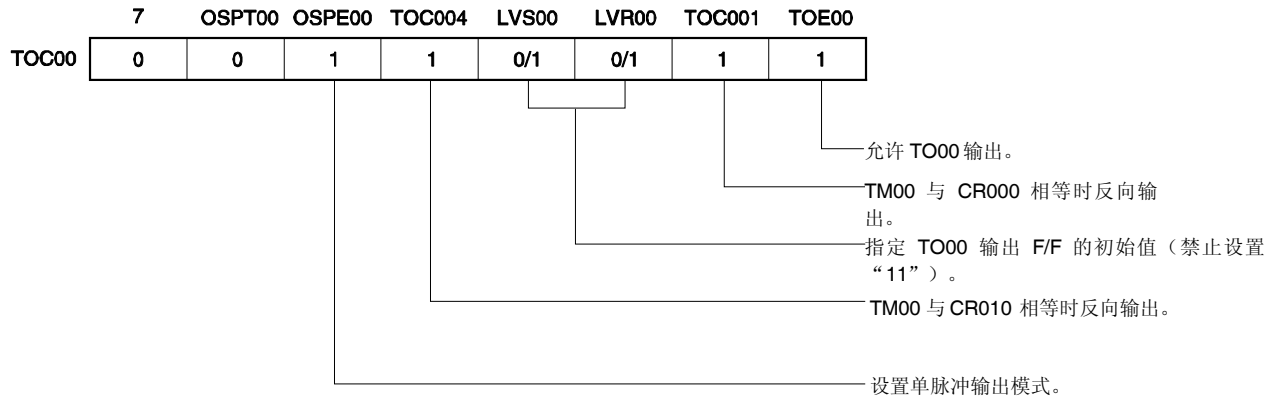
(a) 16 位定时器模式控制寄存器 00 (TMC00)



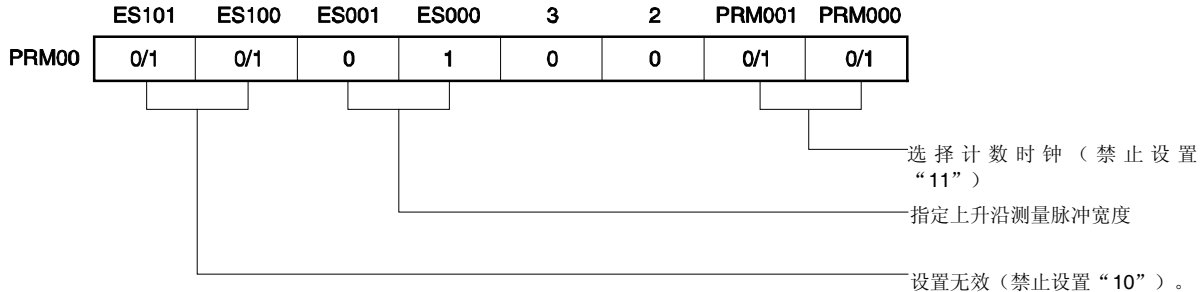
(b) 捕捉/比较控制寄存器 00 (CRC00)



(c) 16 位定时器输出控制寄存器 00 (TOC00)

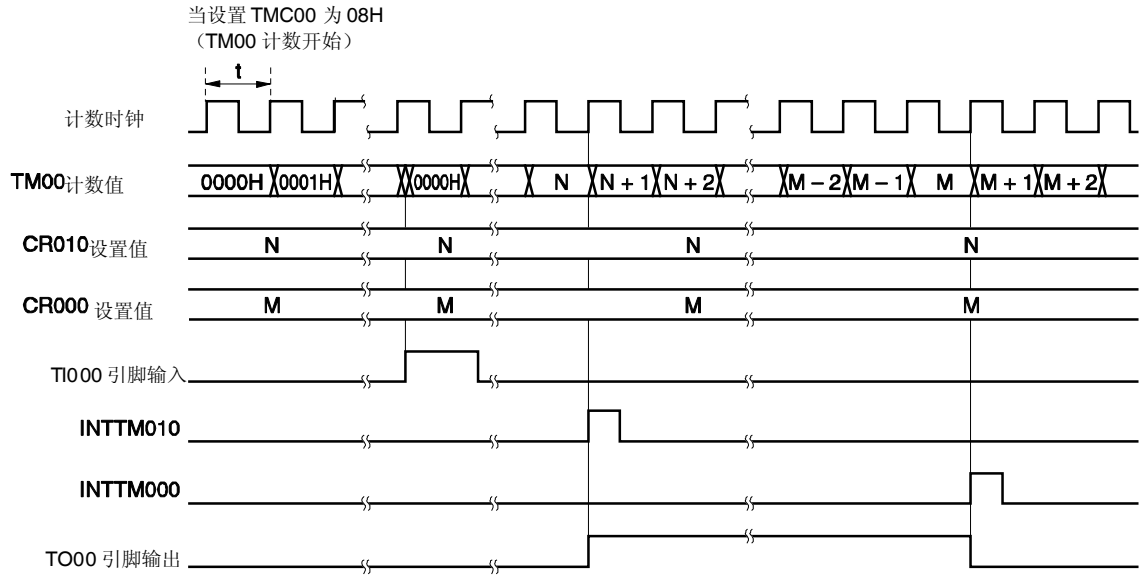


(d) 预分频模式寄存器 00 (PRM00)



注意事项 不要对 CR000 和 CR010 寄存器设置 0000H。

图 6-39. 外部触发的单脉冲输出操作的时序
(指定上升沿)



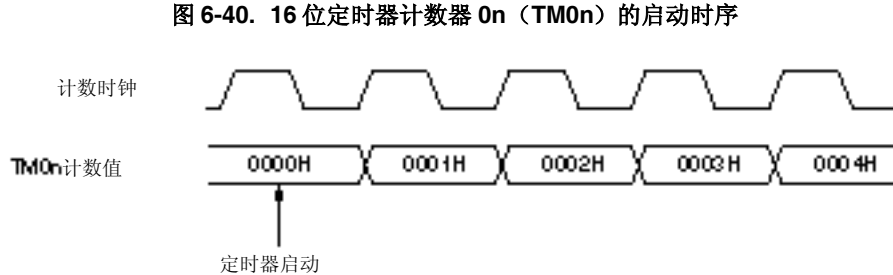
注意事项 只要给 TMC003 和 TMC002 位的设置值不为 00（操作停止模式）时，16 位定时器计数器 00 开始操作。

备注 N < M

6.5 16 位定时器/事件计数器 00 和 01 的注意事项

(1) 定时器启动误差

定时器启动后，在产生相等信号之前可能发生将近一个时钟周期的误差。这是由于启动 TM0n 计数操作与计数脉冲不同步引起的。



(2) 16 位定时器捕捉/比较寄存器 00n 设置

在此模式中，TM0n 和 CR00n 相等时产生清零&启动，设置 16 位定时器捕捉/比较寄存器 00n (CR00n) 为不同于 0000H 的值。这意味着当 16 位定时器/事件计数器 0n 用于外部事件计数器时不能执行单脉冲计数操作。

(3) 捕捉寄存器数据保持时序

在 16 位定时器/事件计数器 00 停止以后，16 位定时器捕捉/比较寄存器 000 和 010 (CR000 和 CR010) 中的值不会被保证 s。

(4) 有效边沿设置

在分别设置 16 为定时器模式控制寄存器 00 (TMC00) 的第 2 位和第 3 位 (TMC002 和 TMC003) 为 0, 0 时，设置 TI000 引脚的有效沿，然后停止定时器操作。使用预分频模式寄存器 00 (PRM00) 的第 4 位和第 5 位 (ES000 和 ES001) 设置有效沿。

(5) 重复触发单脉冲

(a) 软件控制的单脉冲输出

在单脉冲输出时，不能再次设置 OSPT00 位为 1。
为了再次输出单脉冲，应等到当前单脉冲输出完成之后。

(b) 外部触发控制的单脉冲输出

在单脉冲输出时，不能再次输入外部触发。
为了再次输出单脉冲，应等到当前单脉冲输出完成之后。

(c) 单脉冲输出功能

当使用具有软件触发的 16 位定时器/事件计数器 00 的单脉冲输出时，不要改变 TI000 引脚或其复用功能端口引脚的电平。

因为在此示例中外外部触发也是有效的，则在 TI000 引脚或其复用功能端口引脚的电平有效时定时器被清零并启动，这会导致以不期望的时序输出脉冲。

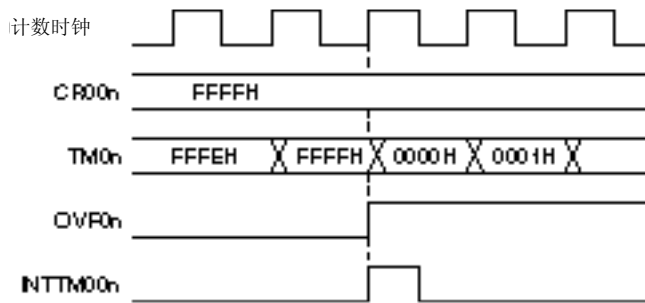
(6) OVF0n 标志的操作

<1> OVF0n 标志在下列中被设置为 1

当在以下模式中时：TM0n 和 CR00n 相等时产生清零&启动模式，TI000 有效边沿产生清零&启动模式，或者自由运行模式。

↓
CR00n 设置为 FFFFH
↓
TM0n 计数从 FFFFH 到 0000H.

图 6-41. OVF0n 标志的操作时序



<2> 即使在 TM0n 溢出之后，下一个计数时钟计数之前（TM0n 变为 0001H 以前）OVF0n 标志被清零，OVF0n 标志重新设置并且禁止被清零。

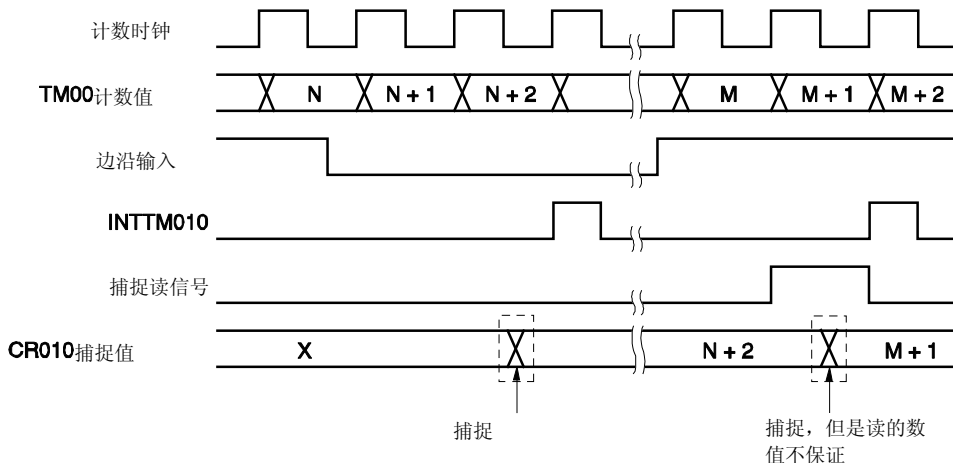
备注 n = 0: μ PD78F0372, 78F0373, 78F0382 和 78F0383
n = 0, 1: μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D

(7) 冲突操作

16 位定时器捕捉/比较寄存器（CR000/CR010）的读取时期和捕捉触发器输入（CR000/CR010 被用作捕捉寄存器）之间的冲突。

捕捉触发输入具有高优先级。从 CR000/CR010 读取的数据不确定。

图 6-42. 捕捉寄存器数据保持时序



(8) 定时器操作

- <1> 即使读取 16 位定时器计数器 00 (TM00)，也不能够通过 16 位定时器捕捉/比较寄存器 010 (CR010) 来捕捉这个值。
- <2> 不论 CPU 是什么运行模式，当定时器停止时，TI000/TI010 引脚的输入信号是不确定的。
- <3> 仅在自由运行模式和在 TI000 有效边沿触发清零&启动模式下单脉冲输出模式操作正确。在 TM00 和 CR000 寄存器值相等时产生的清零&启动模式中，因为不发生溢出，则单脉冲输出无法执行。

(9) 捕捉操作

- <1> 当指定 TI000 的有效沿作为计数时钟时，通过捕捉寄存器指定的捕捉操作(指定 TI000 作为触发)不能正确操作。
- <2> 为了确保捕捉操作的可靠性，捕捉触发要求脉冲宽度必须大于两个计数时钟(由预分频模式寄存器 00 (PRM00) 选择)。
- <3> 在计数时钟的下降沿执行捕捉操作，但中断请求输入(INTTM000/INTTM010)则在下一个计数时钟的上升沿产生。

(10) 比较操作

如果 CR000/CR010 设置为比较模式时，即使一个捕捉触发信号输入了，也不能执行捕捉操作。

(11) 边沿检测

- <1> 如果系统复位后 TI000 或 TI010 引脚处于高电平，并且当指定上升沿或兼有两种脉冲沿作为 TI000 或 TI010 引脚的有效沿时，允许 16 位定时器/事件计数器 00 (TM00) 的操作，在允许操作后一个上升沿会立即被检测到。上拉 TI000 或 TI010 引脚时要特别注意。但如果操作停止之后又重启操作，则不检测上升沿。
- <2> 根据 TI000 的有效沿是用作计数时钟还是用作捕捉触发，采样时钟(用于消除噪音)会发生变化。在前一种情况中采样时钟恒为 f_{PRS} 。而在后一种情况中则将由预分频模式寄存器 00 (PRM00) 选择的计数时钟用于采样。仅当通过采样检测到两次有效沿时才启动捕捉操作，这样可以消除窄脉冲宽度的噪音。

第七章 8位定时器/事件计数器 50 和 51

7.1 8位定时器/事件计数器 50 和 51 的功能

8位定时器/事件计数器 50 和 51 具有以下功能。

- 间隔定时器
- 外部事件计数器
- 方波输出
- PWM 输出

7.2 8位定时器/事件计数器 50 和 51 的配置

8位定时器/事件计数器 50 和 51 包括以下硬件。

表 7-1. 8位定时器/事件计数器 50 和 51 的配置

项目	配置
定时器寄存器	8位定时器计数器 5n (TM5n)
寄存器	8位定时器比较寄存器 5n (CR5n)
定时器输入	TI5n
定时器输出	TO5n
控制寄存器	定时器时钟选择寄存器 5n (TCL5n) 8位定时器模式控制寄存器 5n (TMC5n) 端口模式寄存器 1 (PM1)或端口模式寄存器 3 (PM3) 端口寄存器 1 (P1)或端口寄存器 3 (P3)

图 7-1 与图 7-2 显示 8位定时器/事件计数器 50 和 51 的框图

图 7-1. 8 位定时器/事件计数器 50 的框图

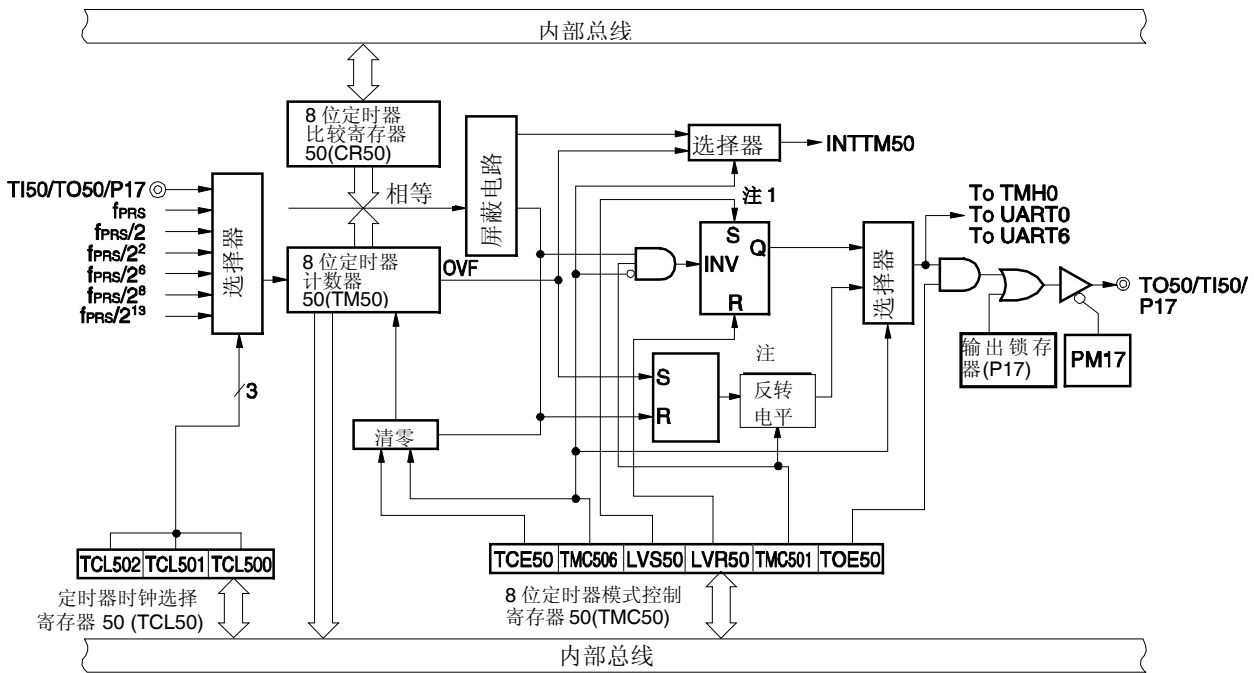
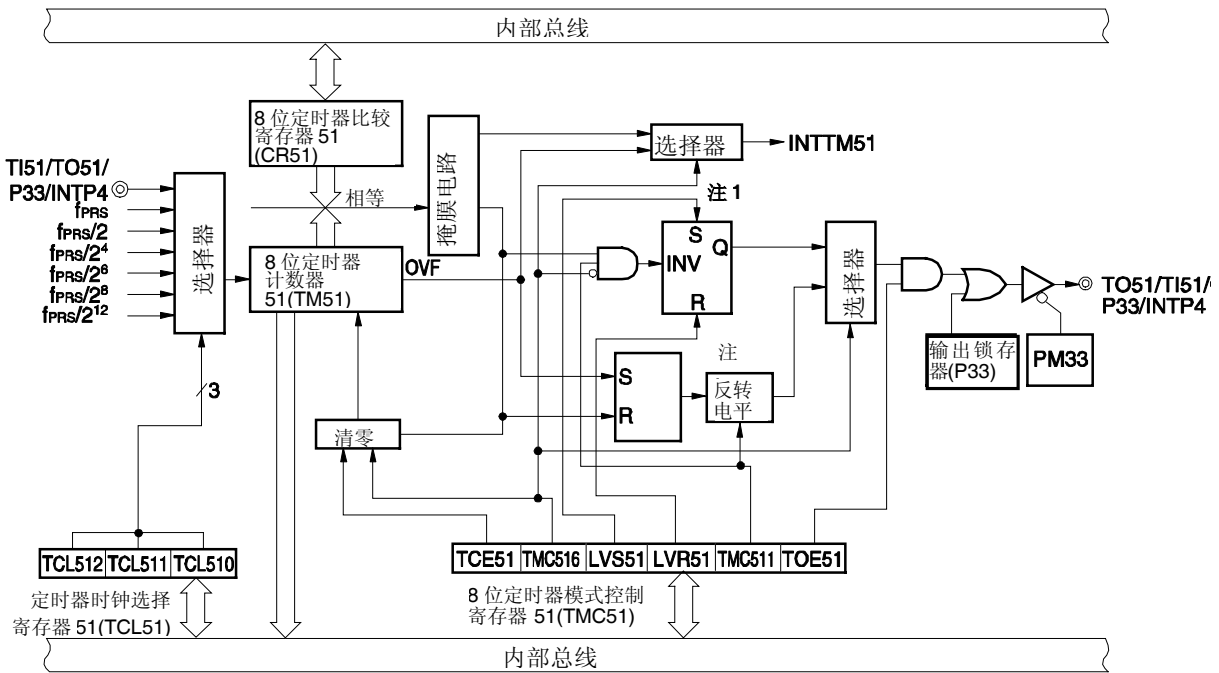


图 7-2. 8 位定时器/事件计数器 51 的框图

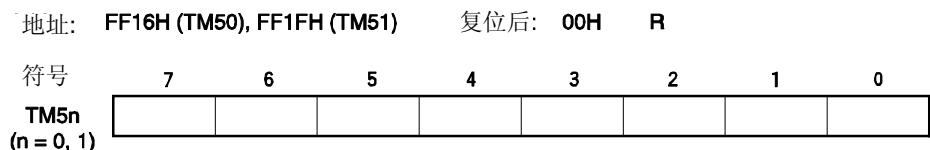


- 注 1. 定时器输出 F/F
2. PWM 输出 F/F

(1) 8 位定时器计数器 5n (TM5n)

TM5n 是一个 8 位寄存器，用于对计数脉冲进行计数，它是只读的。
该计数器的计数值随计数时钟的上升沿同步增加。

图 7-3. 8 位定时器计数器 5n (TM5n) 的格式



在以下几种情况下将计数值清零(00H)。

- <1> 产生复位信号
- <2> TCE5n 清零
- <3> 在 TM5n 与 CR5n 相等时清零和启动模式中，当 TM5n 与 CR5n 相等时。

(2) 8 位定时器比较寄存器 5n (CR5n)

可以由 8 位存储器操作指令对 CR5n 进行读写。

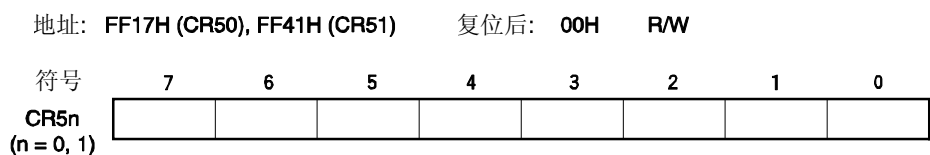
除了 PWM 模式外，CR5n 的值将与 8 位定时器计数器 5n (TM5n) 的计数值频繁相比较，如果相等将产生中断请求 (INTTM5n)。

在 PWM 模式下，TM5n 与 CR5n 的值相等时，TO5n 输出非有效电平，但不产生中断信号。

CR5n 的取值范围为：00H ~ FFH。

复位信号的产生将 CR5n 清零(00H)。

图 7-4. 8 位定时器比较寄存器 5n (CR5n) 的格式



- 注意事项**
1. 在 TM5n 与 CR5n (TMC5n6 = 0)相等时清零和启动模式下，在操作过程中不要将其它值写入 CR5n。
 2. 在 PWM 模式中，以至少 3 个计数时钟 (由 TCL5n 选择时钟) 的间隔对 CR5n 进行重写。

备注 n = 0, 1

7.3 控制 8 位定时器/事件计数器 50 和 51 的寄存器

以下四种寄存器用于控制 8 位定时器/事件计数器 50 和 51。

- 定时器时钟选择寄存器 5n (TCL5n)
- 8 位定时器模式控制寄存器 5n (TMC5n)
- 端口模式寄存器 1 (PM1)或端口模式寄存器 3 (PM3)
- 端口寄存器 1 (P1)或端口寄存器 3 (P3)

(1) 定时器时钟选择寄存器 5n (TCL5n)

该寄存器设置 8 位定时器/事件计数器 5n 的计数时钟，和 TI5n 引脚输入的有效沿。

可由 1 位或 8 位存储器操作指令设置 TCL5n。

复位信号的产生将 TCL5n 清零(00H)。

备注 n = 0, 1

图 7-5. 定时器时钟选择寄存器 50 (TCL50) 的格式

地址: FF6AH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	计数时钟选择				
			f _{PRS} = 2 MHz	f _{PRS} = 5 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz	
0	0	0	TI50 引脚下降沿				
0	0	1	TI50 引脚上升沿				
0	1	0	f _{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	1	1	f _{PRS} /2	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	0	f _{PRS} /2 ²	500 kHz	1.25 MHz	2.5 MHz	5 MHz
1	0	1	f _{PRS} /2 ⁶	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	1	0	f _{PRS} /2 ⁸	7.81 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	1	1	f _{PRS} /2 ¹³	0.24 kHz	0.61 kHz	1.22 kHz	2.44 kHz

注意事项 1. 如果要向 TCL50 写入其它值，则必须提前停止定时器操作。

2. 必须将第 3 位 ~ 第 7 位清零。

备注 f_{PRS}: 外围硬件时钟频率

图 7-6. 定时器时钟选择寄存器 51 (TCL51) 的格式

地址: FF8CH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	计数时钟选择				
			$f_{PRS} =$ 2 MHz	$f_{PRS} =$ 5 MHz	$f_{PRS} =$ 10 MHz	$f_{PRS} =$ 20 MHz	
0	0	0	TI51 引脚下降沿				
0	0	1	TI51 引脚上升沿				
0	1	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	1	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	1	0	$f_{PRS}/2^8$	7.81 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	1	1	$f_{PRS}/2^{12}$	0.49 kHz	1.22 kHz	2.44 kHz	4.88 kHz

- 注意事项**
1. 如果要向 **TCL51** 写入其它值，则必须提前停止定时器操作。
 2. 必须将第 3 位 ~ 第 7 位清零。

备注 f_{PRS} : 外围硬件时钟频率

(2) 8位定时器模式控制寄存器 5n (TMC5n)

TMC5n 主要执行以下五种设置操作。

- <1> 8位定时器计数器 5n (TM5n)计数操作控制
- <2> 8位定时器计数器 5n (TM5n)操作模式选择
- <3> 定时器输出 F/F (flip flop 触发器)状态设置
- <4> 定时器 F/F 控制或 PWM(自由运行)模式中有效电平的选择
- <5> 定时器输出控制

可以由 1 位或 8 位存储器操作指令设置 TMC5n。

复位信号的产生将该寄存器清零(00H)。

备注 n = 0, 1

图 7-7. 8位定时器模式控制寄存器 50 (TMC50) 的格式

地址: FF6BH 复位后: 00H R/W*

符号	<7>	6	5	4	<3>	<2>	1	<0>
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50

TCE50	TM50 计数操作控制
0	清零后, 禁止计数操作 (计数器停止操作)
1	开始计数

TMC506	TM50 操作模式选择
0	TM50 与 CR50 相等时清零&启动模式
1	PWM (自由运行)模式

LVS50	LVR50	定时器输出 F/F 状态设置
0	0	无变化
0	1	定时器输出 F/F 清零(0) (TO50 默认输出: 低电平)
1	0	定时器输出 F/F 设置(1) (TO50 默认输出: 高电平)
1	1	禁止设置

TMC501	其它模式(TMC506 = 0)	PWM 模式(TMC506 = 1)
	定时器 F/F 控制	有效电平选择
0	禁止反转操作	高电平有效
1	允许反转操作	低电平有效

TOE50	定时器输出控制
0	禁止输出 (TM50 输出为低电平)
1	允许输出

注 第 2 位和第 3 位为只写操作位。

(参见下页的注意事项和备注)

图 7-8. 8 位定时器模式控制寄存器 51 (TMC51) 的格式

 地址: FF43H 复位后: 00H R/W ^注

符号	<7>	6	5	4	<3>	<2>	1	<0>
TMC51	TCE51	TMC516	0	0	LVS51	LVR51	TMC511	TOE51
TCE51	TM51 计数操作控制							
0	清零后禁止计数操作 (计数器停止操作)							
1	开始计数							
TMC516	TM51 操作模式选择							
0	TM51 与 CR51 相等时清零 & 启动模式							
1	PWM (自由运行) 模式							
LVS51	LVR51	定时器输出 F/F 状态设置						
0	0	无变化						
0	1	定时器输出 F/F 清零 (0) (TO51 默认输出: 低电平)						
1	0	定时器输出 F/F 设置 (1) (TO51 默认输出: 高电平)						
1	1	禁止设置						
TMC511	其它模式 (TMC516 = 0)				PWM 模式 (TMC516 = 1)			
	定时器 F/F 控制				有效电平选择			
0	禁止反转操作				高电平有效			
1	允许反转操作				低电平有效			
TOE51	定时器输出控制							
0	禁止输出 (TM51 输出为低电平)							
1	允许输出							

注 第 2 位和第 3 位为只写操作位。

注意事项 1. 除 PWM 模式外, 对 LVS5n 与 LVR5n 的设置有效。

2. 按下列顺序执行 <1> ~ <4>, 注意不能同时执行。

- <1> 设置 TMC5n1, TMC5n6: 操作模式设置
- <2> 设置 TOE5n, 允许输出: 允许定时器输出
- <3> 设置 LVS5n, LVR5n (见注意事项 1): 定时器 F/F 设置
- <4> 设置 TCE5n

3. 在重写 TMC5n6 之前停止操作。

- 备注
1. 在 PWM 模式中, 将 TCE5n 清零可禁止 PWM 输出。
 2. 读取 LVS5n 与 LVR5n 时, 读取的值为 0。
 3. 无论 TCE5n 取何值, TMC5n6、LVS5n、LVR5n、TMC5n1 和 TOE5n 的值都与 TO5n 引脚有关。
 4. n = 0, 1

(3) 端口模式寄存器 1 和 3 (PM1, PM3)

这两个寄存器用于按位设置端口 1 和 3 的输入/输出操作模式。

当 P17/TO50/TI50 和 P33/TO51/TI51/INTP4 引脚用于定时器输出时，将 PM17 和 PM33 以及 P17 与 P33 的输出锁存器清零。

当 P17/TO50/TI50 和 P33/TO51/TI51/INTP4 引脚用于定时器输入时，将 PM17 和 PM33 置 1，P17 与 P33 的输出锁存值可以为 0 也可以为 1。

可以由 1 位或 8 位存储器操作指令设置 PM1 和 PM3。

复位信号的产生将寄存器的内容设置为 FFH。

图 7-9. 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10
PM1n	P1n 引脚 I/O 模式选择 (n = 0 ~ 7)							
0	输出模式 (输出缓冲器打开)							
1	输入模式 (输出缓冲器关闭)							

图 7-10. 端口模式寄存器 3 (PM3) 的格式

地址: FF23H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30
PM3n	P3n 引脚 I/O 模式选择 (n = 0 ~ 3)							
0	输出模式 (输出缓冲器打开)							
1	输入模式 (输出缓冲器关闭)							

7.4 8 位定时器/事件计数器 50 和 51 的操作

7.4.1 用作间隔定时器

8 位定时器/事件计数器 5n 用作间隔定时器，可以在间隔时间内不断产生中断请求，该间隔时间由 8 位定时器比较寄存器 5n (CR5n) 的预置值决定。

当 8 位定时器计数器 5n (TM5n) 中的计数值与 CR5n 中的值相等时，将 TM5n 清零并继续计数，同时产生中断请求信号 (INTTM5n)。

根据定时器时钟选择寄存器 5n (TCL5n) 的第 0 位~第 2 位 (TCL5n0 ~ TCL5n2) 选择 TM5n 的计数时钟。

设置

<1> 设置寄存器

- TCL5n: 设置计数时钟
- CR5n: 比较值
- TMC5n: 停止计数操作，选择 TM5n 与 CR5n 相等时清零&启动模式。
(TMC5n = 0000xxx0B, x = 不必考虑)

<2> 将 TCE5n 置 1 后，开始计数。

<3> 如果 TM5n 与 CR5n 值相等时，产生中断 INTTM5n (将 TM5n 清零(00H))。

<4> 以相同的时间间隔重复产生 INTTM5n。

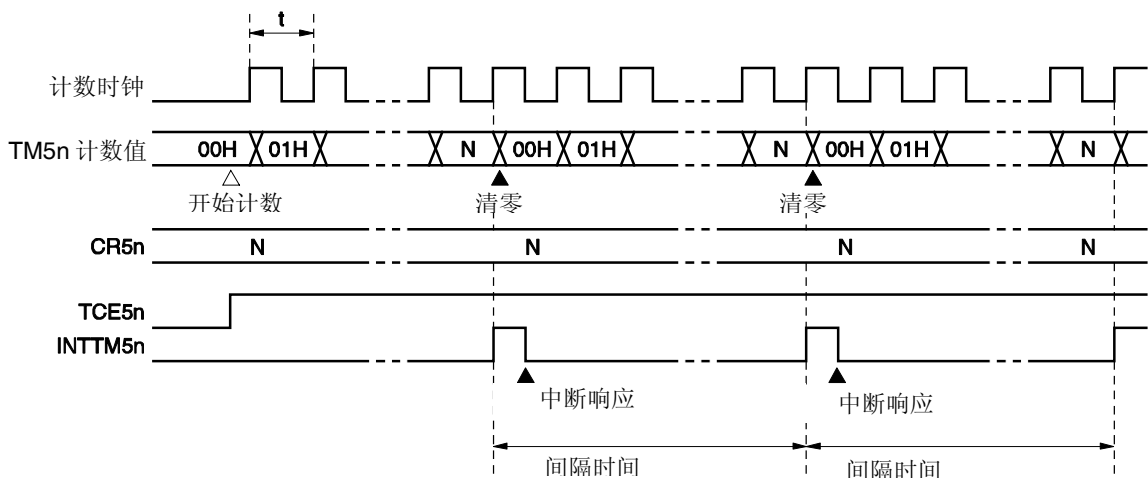
将 TCE5n 清零，停止计数操作。

注意事项 操作期间不要向 CR5n 写入其它值。

- 备注**
1. 如需了解如何允许 INTTM5n 中断，请参见 第十九章 中断功能。
 2. n = 0, 1

图 7-11. 间隔定时器操作的时序 (1/2)

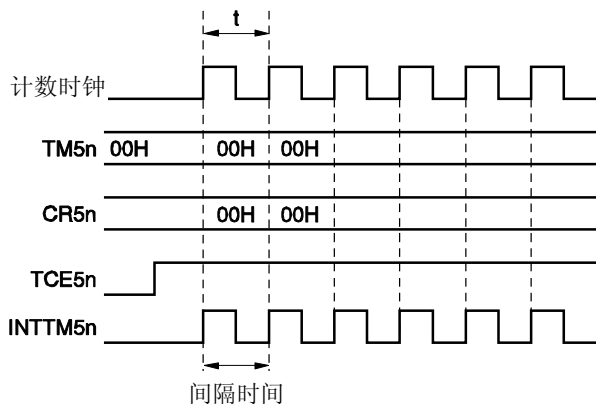
(a) 基本操作



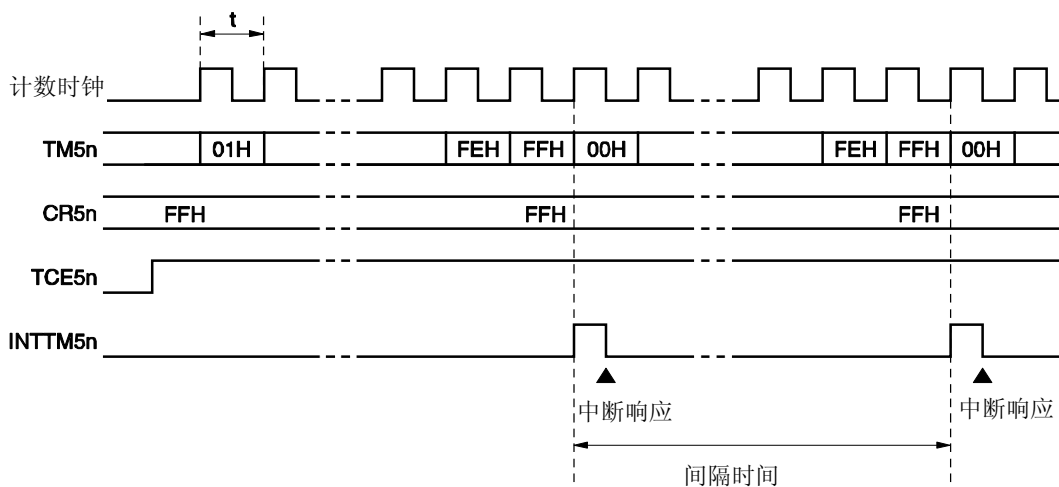
- 备注**
- 间隔时间 = $(N + 1) \times t$
 $N = 01H \sim FFH$
 $n = 0, 1$

图 7-11. 间隔定时器操作的时序 (2/2)

(b) 当 CR5n = 00H



(c) 当 CR5n = FFH



备注 n = 0, 1

7.4.2 用作外部事件计数器

外部事件计数器通过 8 位定时器计数器 5n (TM5n) 对输入到 TI5n 引脚的外部时钟脉冲计数。

TM5n 随输入的有效沿次数递增。有效沿由定时器时钟选择寄存器 5n (TCL5n) 规定, 可以选择上升沿或下降沿。

当 TM5n 中的计数值与 8 位定时器比较寄存器 5n (CR5n) 中的值相等时, 将 TM5n 清零, 并产生中断请求信号 (INTTM5n)。

一旦 TM5n 中的值与 CR5n 的值相等, 将产生 INTTM5n 中断请求。

设置

<1> 设置寄存器

- 设置端口模式寄存器 (PM17 或 PM33)^{*} 的值为 1。
- TCL5n: 选择 TI5n 引脚输入脉冲沿
 TI5n 引脚下降沿 → TCL5n = 00H
 TI5n 引脚上升沿 → TCL5n = 01H
- CR5n: 比较值
- TMC5n: 停止计数, 选择 TM5n 与 CR5n 相等时清零&启动模式, 禁止定时器 F/F 反相操作, 禁止定时器输出。

(TMC5n = 0000x00B x =不必考虑)

<2> 当 TCE5n = 1 时, 开始对从 TI5n 引脚输入的脉冲计数。

<3> 当 TM5n 与 CR5n 的值相等时, 将产生 INTTM5n (TM5n 清零 (00H))。

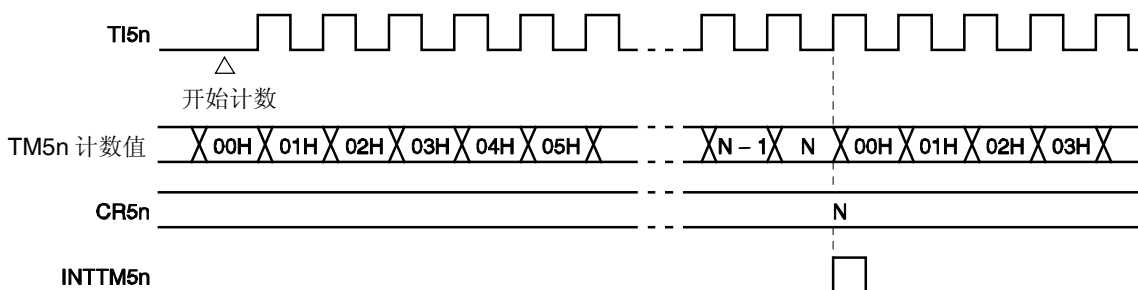
<4> 在经过以上设置后, 每当 TM5n 与 CR5n 的值相等时都会产生 INTTM5n 中断请求。

注 8 位定时器/事件计数器 50: PM17

8 位定时器/事件计数器 51: PM33

备注 如需了解如何允许 INTTM5n 中断, 请参见 第十九章 中断功能。

图 7-12. 外部事件计数器操作的时序 (规定上升沿有效)



备注 N = 00H ~ FFH

n = 0, 1

7.4.3 方波输出操作

以 8 位定时器比较寄存器 5n (CR5n) 的预置值决定的间隔时间输出任选频率的方波。

通过将 8 位定时器模式控制寄存器 5n (TMC5n) 的第 0 位 (TOE5n) 置 1, 可以在规定的间隔时间内反转 TO5n 引脚输出的状态。这样, 可以输出任选频率的方波 (占空比=50%)。

设置

<1> 设置寄存器

- 端口输出锁存器 (P17 或 P33)^注 和端口模式寄存器 (PM17 或 PM33)^注 清零。
- TCL5n: 选择计数时钟
- CR5n: 比较值
- TMC5n: 停止计数, 选择 TM5n 与 CR5n 相等时清零&启动模式

LVS5n	LVR5n	定时器输出 F/F 状态设置
1	0	定时器输出 F/F 清零 (0) (TO5n 引脚的默认输出值: 低电平)
0	1	定时器输出 F/F 置 (1) (TO5n 引脚的默认输出值: 高电平)

允许定时器输出

(TMC5n = 00001011B 或 00000111B)

<2> 一旦 TCE5n = 1, 便开始计数。

<3> 通过 TM5n 与 CR5n 的相等, 反转定时器输出 F/F。产生 INTTM5n 后, 将 TM5n 清零 (00H)。

<4> 在经过以上设置后, 在相同间隔内反转定时器输出 F/F, 并从 TO5n 输出方波。

频率如下。

- 频率 = $1/2t(N + 1)$
(N: 00H ~ FFH)

注 8 位定时器/事件计数器 50: P17, PM17

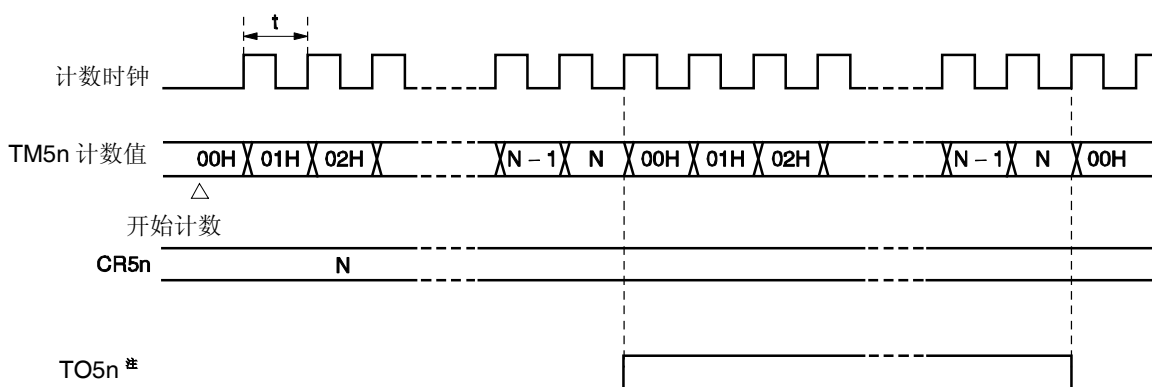
8 位定时器/事件计数器 51: P33, PM33

注意事项 操作期间不要向 CR5n 写入其它值。

备注 1. 如需了解如何允许 INTTM5n 中断, 请参见 第十九章 中断功能。

2. n = 0, 1

图 7-13. 方波输出操作的时序



注 由 8 位定时器模式控制寄存器 5n (TMC5n) 的第 2 位、第 3 位 (LVR5n, LVS5n) 设置 TO5n 的初始值。

7.4.4 PWM 输出操作

当 8 位定时器模式控制寄存器 5n (TMC5n) 的第 6 位 (TMC5n6) 置 1 时，8 位定时器/事件计数器 5n 作为 PWM 输出使用。

占空比由 8 位定时器比较寄存器 5n (CR5n) 的值决定，并从 TO5n 引脚输出。

设置 CR5n 为 PWM 脉冲的有效电平宽度；可以由 TMC5n 的第 1 位 (TMC5n1) 选择有效电平。

由定时器时钟选择寄存器 5n (TCL5n) 的第 0 位~第 2 位 (TCL5n0 ~ TCL5n2) 选择计数时钟。

可以由 TMC5n 的第 0 位 (TOE5n) 决定 PWM 输出允许/禁止。

注意事项 在 PWM 模式中，当对 CR5n 进行重写时，需要至少 3 个计数时钟 (该计数时钟由 TCL5n 选择) 周期的写间隔。

备注 n = 0, 1

(1) PWM 输出的基本操作

设置

<1> 设置寄存器

- 将端口输出锁存器 (P17 或 P33)[※] 和端口模式寄存器 (PM17 或 PM33)[※] 清零。
- TCLK5n: 选择计数时钟
- CR5n: 比较值
- TMC5n: 停止计数操作, 选择 PWM 模式
定时器输出 F/F 不改变。

TMC5n1	有效电平选择
0	高电平有效
1	低电平有效

允许定时器输出

(TMC5n = 01000001B 或 01000011B)

- <2> 当 TCE5n = 1 时, 计数开始。
TCE5n 清零可以停止计数操作。

注 8 位定时器/事件计数器 50: P17, PM17
8 位定时器/事件计数器 51: P33, PM33

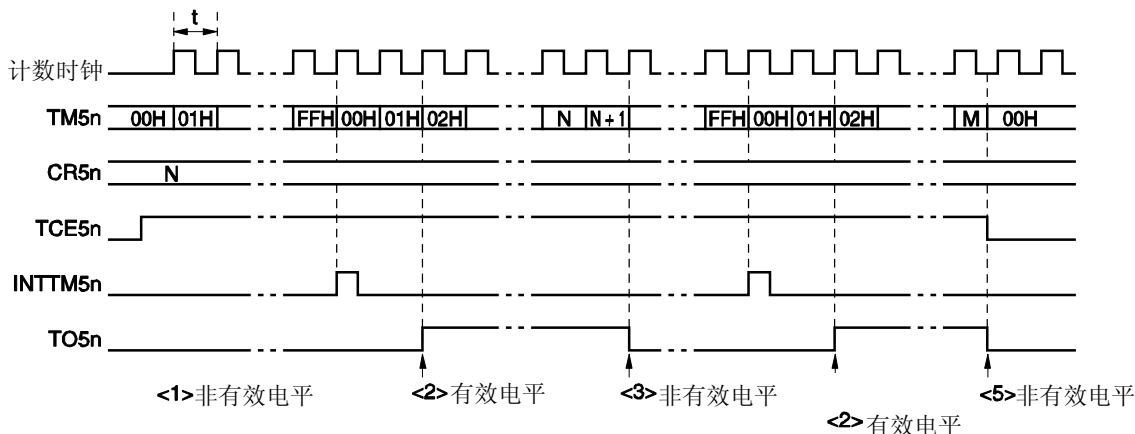
PWM输出操作

- <1> PWM 输出 (从 TO5n 输出) 一直输出非有效电平, 直至产生溢出。
 <2> 当发生溢出时, 输出有效电平。直至 CR5n 与 8 位定时器计数器 5n (TM5n) 的计数值相等时才停止输出有效电平。
 <3> 若 CR5n 与计数值相等, 将输出非有效电平直至溢出再次发生。
 <4> 重复 <2>和 <3>的操作, 直至计数停止。
 <5> 当 TCE5n=0 停止计数时, PWM 输出非有效电平。
 需要了解时序的详细信息, 参见图 7-14 和 7-15。
 周期、有效电平宽度和占空比如下。
- 周期 = $2^8 t$
 - 有效电平宽度 = Nt
 - 占空比 = $N/2^8$
(N = 00H ~ FFH)

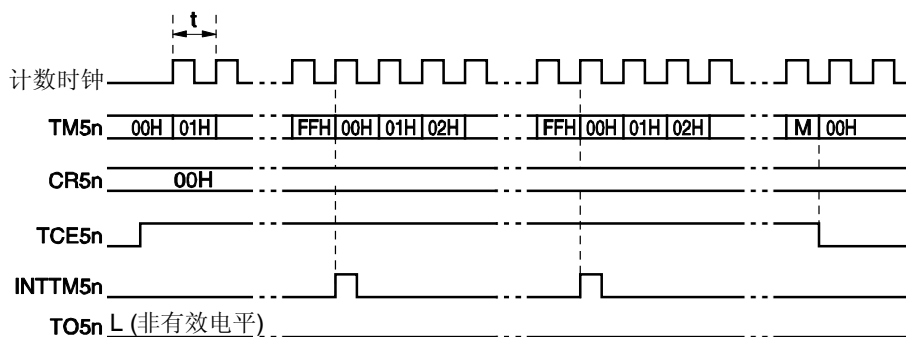
备注 n = 0, 1

图 7-14. PWM 输出操作的时序

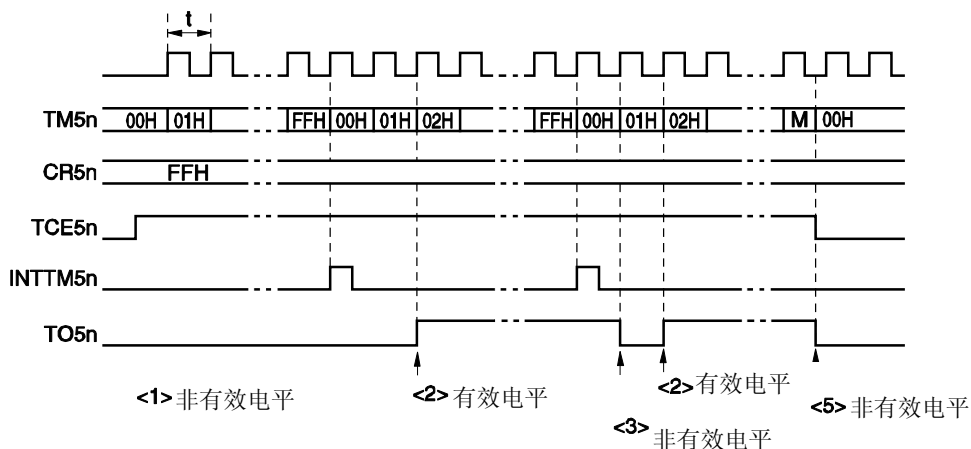
(a) 基本操作 (有效电平 = H)



(b) CR5n = 00H



(c) CR5n = FFH

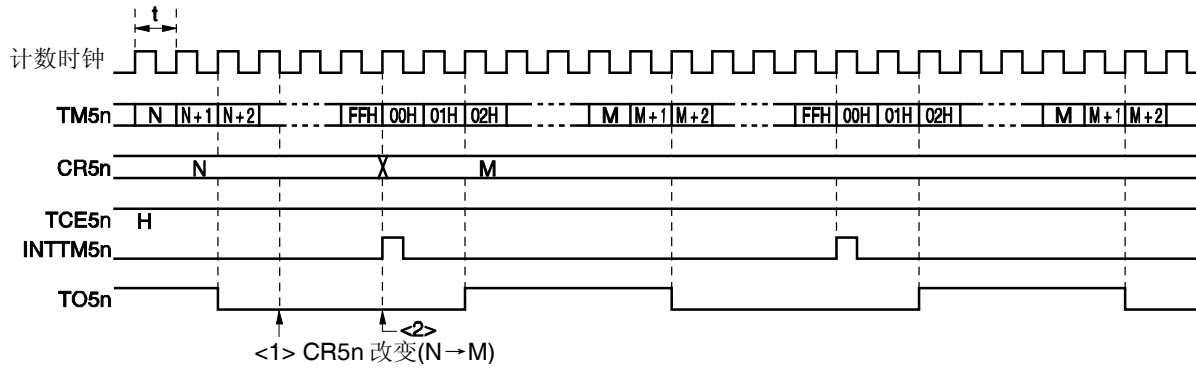


- 备注
1. 图 7-14 (a) 中的 <1> ~ <3> 和 <5> 与 7.4.4 (1) PWM 输出基本操作的 PWM 输出操作中 <1> ~ <3> 和 <5> 相对应。
 2. $n = 0, 1$

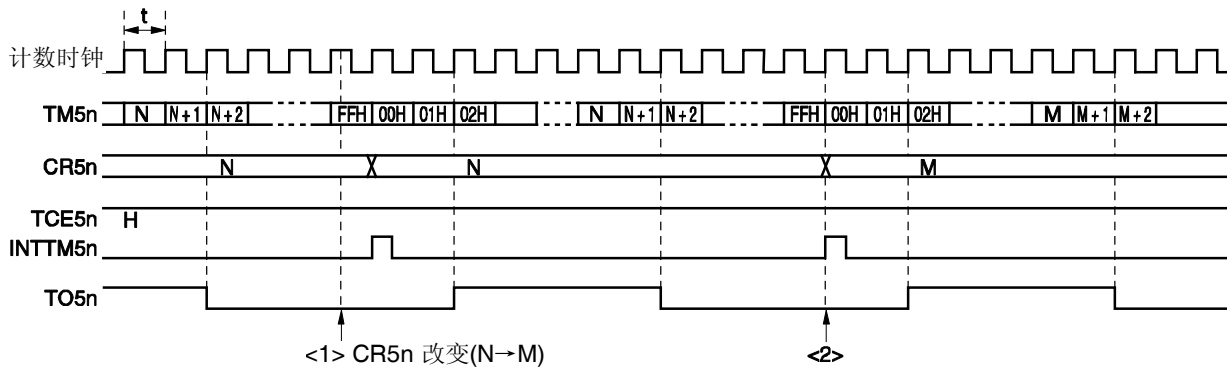
(2) 当 CR5n 改变时的操作

图 7-15. 改变 CR5n 时的操作时序

- (a) 在计数值为 FFH 时，时钟上升沿出现之前，CR5n 的值由 N 变为 M
 → 改变后，溢出时立即将值传送至 CR5n。



- (b) 在计数值为 FFH 时，时钟上升沿出现之后，CR5n 的值由 N 变为 M
 → 当出现第 2 次溢出时将值传送至 CR5n。



注意事项 图 7-15 <1>与<2> 之间读取 CR5n 时，读取的值与实际值不同(读取的值：M，CR5n 实际值：N)。

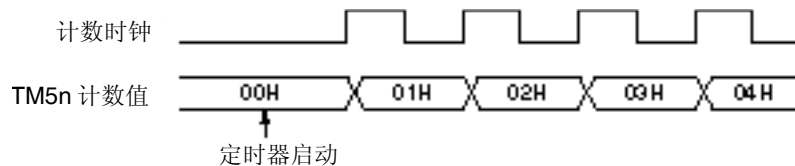
7.5 使用 8 位定时器/事件计数器 50 和 51 的注意事项

(1) 定时器启动误差

定时器启动后，在产生相等信号之前可能发生将近一个时钟周期的误差。

这是由于 8 位定时器计数器 50 和 51(TM50, TM51)的启动与计数时钟不同步引起的。

图 7-16. 8 位定时器计数器 5n 启动的时序



备注 n = 0, 1

第八章 8 位定时器 H0 和 H1

8.1 8 位定时器 H0 和 H1 的功能

8 位定时器 H0 和 H1 的功能如下。

- 间隔定时器
- 方波输出
- PWM 输出模式
- 载波发生器模式 (仅用于 8 位定时器 H1)

8.2 8 位定时器 H0 和 H1 的配置

8 位定时器 H0 和 H1 包括以下硬件

表 8-1. 8 位定时器 H0 和 H1 的配置

项目	配置
定时器寄存器	8 位定时器计数 Hn
寄存器	8 位定时器 H 比较寄存器 0n (CMP0n) 8 位定时器 H 比较寄存器 1n (CMP1n)
定时器输出	TOHn, 输出控制器
控制寄存器	8 位定时器 H 模式寄存器 n (TMHMDn) 8 位定时器 H 载波控制寄存器 1 (TMCYC1) ^注 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1)

注 仅用于 8 位定时器 H1

备注 n = 0, 1

图 8-1 和 8-2 为定时器框图。

图 8-1. 8 位定时器 H0 的框图

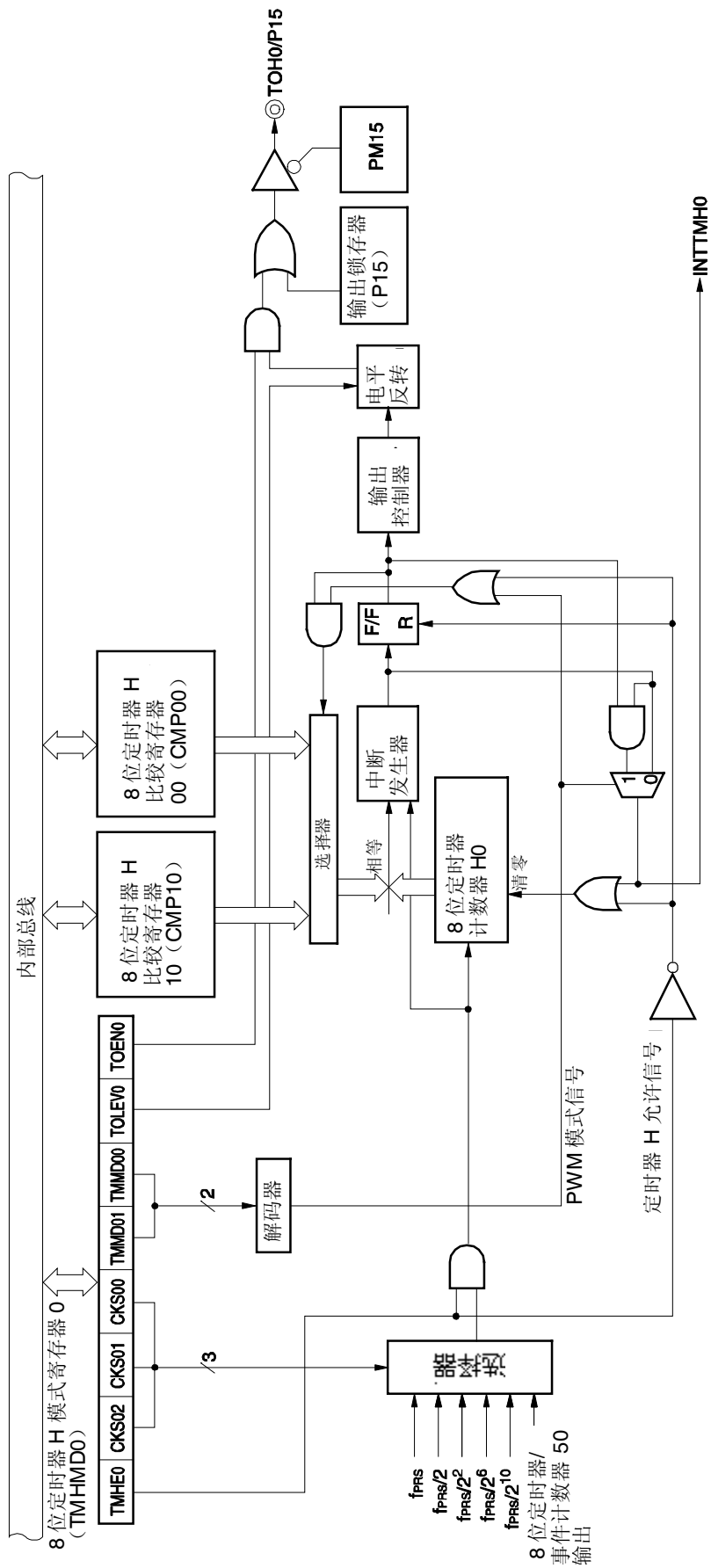
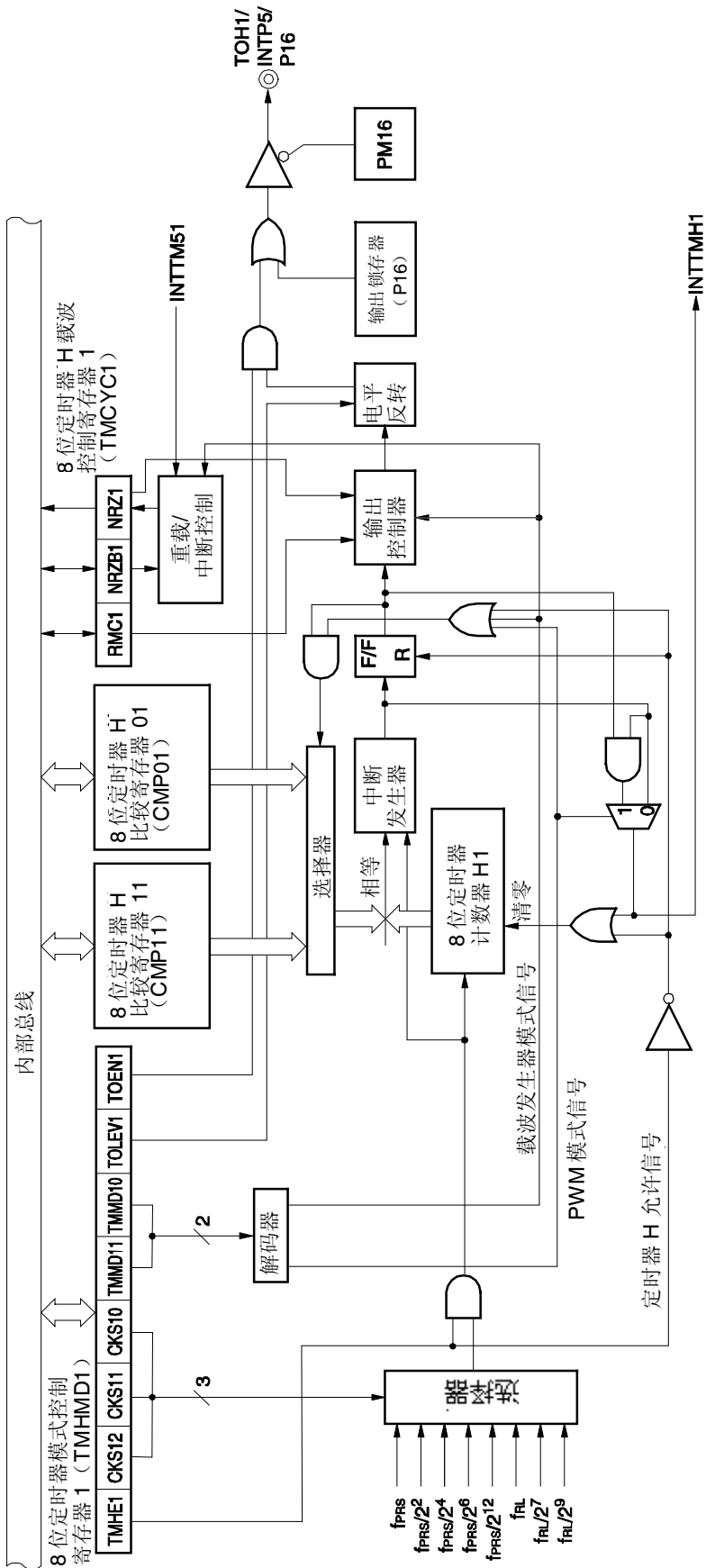


图 8-2. 8 位定时器 H1 的框图



(1) 8 位定时器 H 比较寄存器 0n (CMP0n)

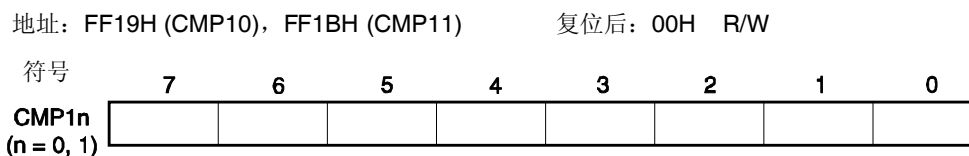
该寄存器可由 8 位存储器操作指令进行读写。该寄存器用于此定时器的所有操作模式。

该寄存器不断地将 CMP0n 的值和 8 位定时器计数器 Hn 的计数值进行比较，并且当两个值相等时，产生一个中断请求信号 (INTTMHn) 并反转 TOHn 的输出。

当定时器停止时 (TMHEn = 0) 重写 CMP0n 的值。

复位信号的产生将该寄存器设置为 00H。

图 8-3. 8 位定时器 H 比较寄存器 0n (CMP0n) 的格式



注意事项 CMP0n 在定时器计数操作期间不能被重写。

(2) 8 位定时器 H 比较寄存器 1n (CMP1n)

可由 8 位存储器操作指令读写该寄存器。该寄存器用于 PWM 输出模式和载波发生器模式。

在 PWM 输出模式中，该寄存器不断地将 CMP1n 的设置值和 8 位定时器计数器 Hn 的计数值进行比较，当两个值相等时，反转 TOHn 的输出。不产生中断请求信号。

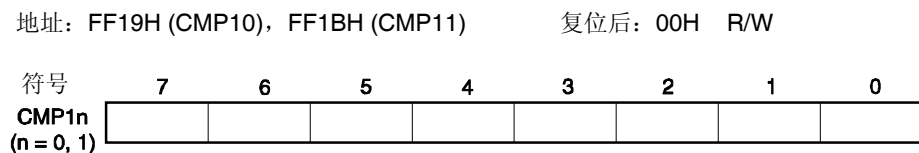
在载波发生器模式中，CMP1n 寄存器不断地将 CMP1n 的设置值和 8 位定时器计数器 Hn 的计数值进行比较，如果两个值相等，则产生一个中断请求信号 (INTTMHn)。同时，计数值清零。

可以在定时器计数期间对 CMP1n 进行重写。

如果在定时器操作期间重写 CMP1n 的值，则当定时器计数器的值和 CMP1n 以前的值相等时，锁存新的值并传输给 CMP1n，然后 CMP1n 的值用新值取代。如果计数值和 CMP1n 值的相等操作与 CMP1n 的写操作发生冲突，那么 CMP1n 的值将保持不变。

复位信号的产生对该寄存器清零(00H)。

图 8-4. 8 位定时器 H 比较寄存器 1n (CMP1n) 的格式



注意事项 在 PWM 输出模式和载波发生器模式下，当停止定时器计数操作 (TMHEn = 0) 后再启动定时器计数操作 (TMHEn = 1) 时，必须设置 CMP1n (即使对 CMP1n 设置相同的值也要再设置一次)。

备注 n = 0, 1

8.3 控制 8 位定时器 H0 和 H1 的寄存器

以下四个寄存器用于控制 8 位定时器 H0 和 H1。

- 8 位定时器 H 模式寄存器 n (TMHMDn)
- 8 位定时器 H 载波控制寄存器 1 (TMCYC1)^注
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

注 仅用于 8 位定时器 H1

(1) 8 位定时器 H 模式寄存器 n (TMHMDn)

该寄存器用于控制定时器 H 的模式。

可以由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号的产生对寄存器清零 (00H)。

备注 n = 0, 1

图 8-5. 8 位定时器 H 模式寄存器 0 (TMHMD0) 的格式

地址: FF69H

复位后: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	是否允许定时器操作
0	停止定时器计数操作 (计数器清零)
1	允许定时器计数 (由输入时钟启动计数操作)

CKS02	CKS01	CKS00	计数时钟选择				
			f _{PRS} = 2 MHz	f _{PRS} = 5 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz	
0	0	0	f _{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	1	f _{PRS} /2	1 MHz	2.5 MHz	5 MHz	10 MHz
0	1	0	f _{PRS} /2 ²	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	1	1	f _{PRS} /2 ⁶	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	0	0	f _{PRS} /2 ¹⁰	1.95 kHz	4.88 kHz	9.77 kHz	19.54 kHz
1	0	1	TM50 输出 ^注				
其它			禁止设置				

TMMD01	TMMD00	定时器操作模式
0	0	间隔定时器模式
1	0	PWM 输出模式
其它		禁止设置

TOLEV0	定时器输出电平控制 (默认模式)
0	低电平
1	高电平

TOEN0	定时器输出控制
0	禁止输出
1	允许输出

注 在选择 TM50 输出作为计数时钟时, 应注意以下几点。

- TM50 和 CR50 的值相等时产生清零&启动模式 (TMC506 = 0),
先启动 8 位定时器/事件计数器 50, 然后允许定时器进行 F/F 反转操作 (TMC501 = 1)
- PWM 模式 (TMC506 = 1)
先启动 8 位定时器/事件计数器 50, 然后设置计数时钟, 使占空比= 50%
在任何模式中都不必将 TO50 作为定时器输出引脚。

- 注意事项
1. 当 **TMHE0=1** 时，禁止设置 **TMHMD0** 的其他位。
 2. 在 **PWM** 输出模式中，当定时器停止计数后 (**TMHE0 = 0**) 再次启动计数操作时 (**TMHE0 = 1**)，必须设置 8 位定时器 H 比较寄存器 10 (**CMP10**) 的值 (即使设置的是相同的值，也必须再次设置)。

- 备注
1. f_{PRS} : 外围硬件时钟频率
 2. **TMC506**: 8 位定时器模式控制寄存器 50 (**TMC50**) 的第 6 位
TMC501: **TMC50** 的第 1 位

图 8-6. 8 位定时器 H 模式寄存器 1 (TMHMD1) 的格式

地址: FF6CH 复位后: 00H R/W

	<7>	6	5	4	3	2	<1>	<0>
TMHMD1	TMHE1	CKS12	CKS11	CKS10	TMMD11	TMMD10	TOLEV1	TOEN1

TMHE1	是否允许定时器操作
0	停止定时器计数操作 (计数器清零)
1	允许定时器计数 (由输入时钟启动计数操作)

CKS12	CKS11	CKS10		计数时钟选择			
				f _{PRS} = 2 MHz	f _{PRS} = 5 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz
0	0	0	f _{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	1	f _{PRS} /2 ²	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	1	0	f _{PRS} /2 ⁴	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	1	f _{PRS} /2 ⁶	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
1	0	0	f _{PRS} /2 ¹²	0.49 kHz	1.22 kHz	2.44 kHz	4.88 kHz
1	0	1	f _{RL} /2 ⁷	1.88 kHz (TYP.)			
1	1	0	f _{RL} /2 ⁹	0.47 kHz (TYP.)			
1	1	1	f _{RL}	240 kHz (TYP.)			

TMMD11	TMMD10	定时器操作模式
0	0	间隔定时器模式
0	1	载波发生器模式
1	0	PWM 输出模式
1	1	禁止设置

TOLEV1	定时器输出电平控制 (默认模式)
0	低电平
1	高电平

TOEN1	定时器输出控制
0	禁止输出
1	允许输出

- 注意事项**
1. 当 TMHE1=1 时, 禁止设置 TMHMD1 的其他位。
 2. 在 PWM 输出模式和载波发生模式中, 当定时器停止计数后 (TMHE1=0) 再次启动计数器操作时 (TMHE1=1), 必须设置 8 位定时器 H 比较寄存器 11 (CMP11) 的值 (即使设置的是相同的值, 也必须再次设置)。
 3. 当使用载波发生器模式时, 设置 TMH1 的计数时钟频率至少是 TM51 计数时钟频率的 6 倍。

- 备注**
1. f_{PRS}: 外围硬件时钟频率
 2. f_{RL}: 内部低速振荡时钟频率

(2) 8 位定时器 H 载波控制寄存器 1 (TMCYC1)

该寄存器用于控制 8 位定时器 H1 的遥控输出和载波脉冲输出状态。

可以由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号的产生对寄存器清零 (00H)。

图 8-7. 8 位定时器 H 载波控制寄存器 1 (TMCYC1) 的格式

地址: FF6DH 复位后: 00H R/W[≠]

	7	6	5	4	3	2	1	<0>
TMCYC1	0	0	0	0	0	RMC1	NRZB1	NRZ1

RMC1	NRZB1	遥控输出
0	0	低电平输出
0	1	高电平输出
1	0	低电平输出
1	1	载波脉冲输出

NRZ1	载波脉冲输出状态标志
0	禁止载波输出状态 (低电平状态)
1	允许载波输出状态 (RMC1=1: 载波脉冲输出, RMC1=0: 高电平状态)

注 第 0 位只读。

(3) 端口模式寄存器 1 (PM1)

该寄存器按位设置端口 1 为输入/输出操作模式。

在将 P15/TOH0 和 P16/TOH1/INTP5 引脚用于定时器输出时, 需对 PM15 和 PM16 以及 P15 和 P16 的输出锁存器清零(0)。

可以由 1 位或 8 位存储器操作指令设置 PM1。

复位信号的产生将寄存器置 FFH。

图 8-8. 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 复位后: FFH R/W

符号

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

PM1n	P1n 引脚 I/O 模式 选择 (n = 0 ~ 7)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

8.4 8 位定时器 H0 和 H1 的操作

8.4.1 间隔定时器/方波输出操作

当 8 位定时器计数器 H_n 与比较寄存器 0_n (CMP0n) 相等时, 将产生中断请求信号 (INTTMHn) 并对 8 位定时器计数器 H_n 清零 (00H)。

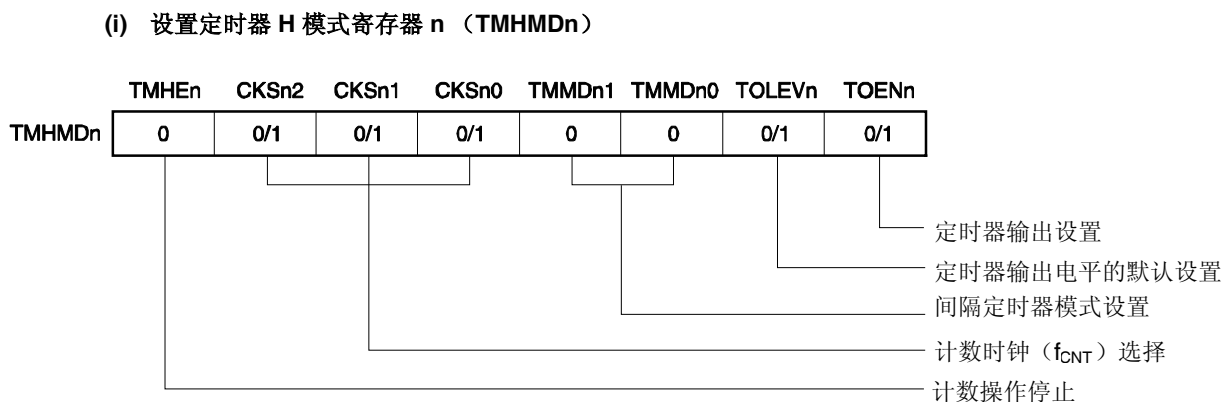
在间隔定时器模式中不使用比较寄存器 1_n (CMP1n)。即使 CMP1n 寄存器已赋值, 也不会检测 8 位定时器计数器 H_n 与 CMP1n 寄存器的相等情况, 所以不影响定时器的输出。

通过将定时器 H 模式寄存器 n (TMHMDn) 的第 0 位 (TOENn) 置 1, 可以从 TOHn 输出任意频率的方波 (占空比 = 50%)。

设置

<1> 寄存器设置。

图 8-9. 间隔定时器/方波输出操作期间的寄存器设置



(ii) CMP0n 寄存器设置

如果 N 作为比较值, 则时间间隔如下:

- 时间间隔 = $(N + 1)/f_{CNT}$

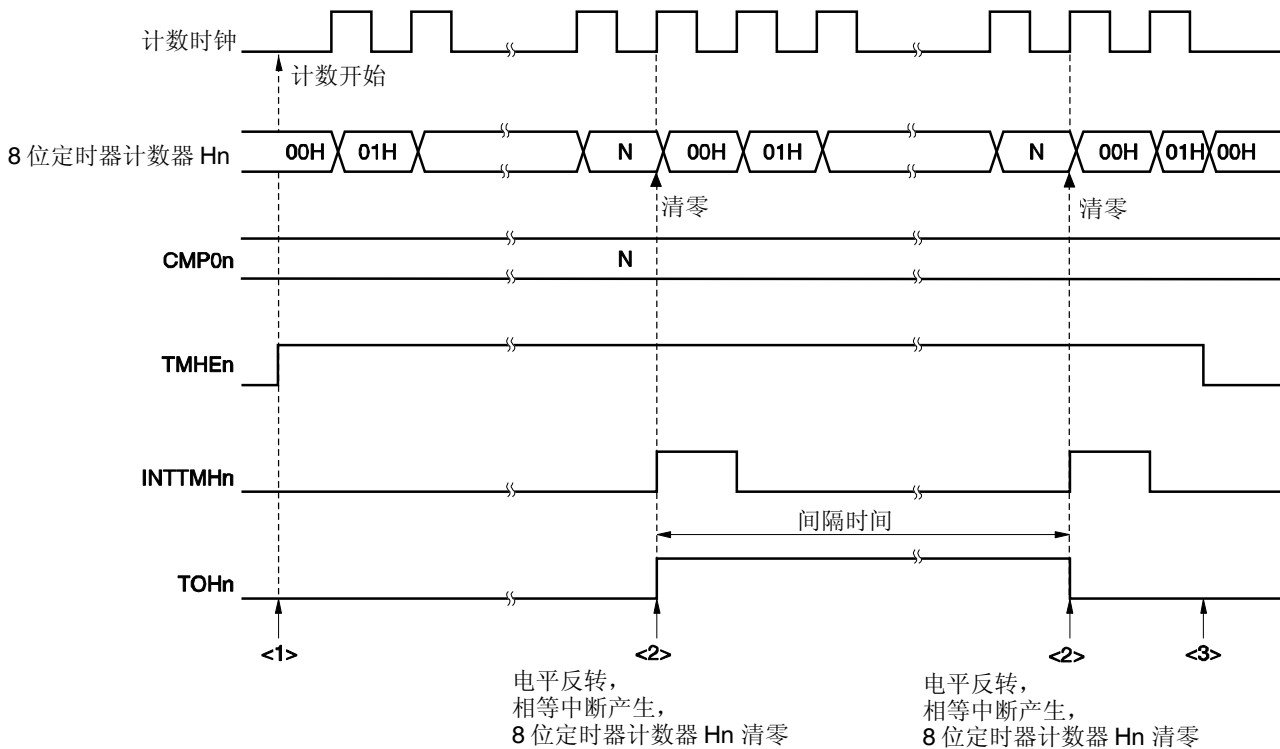
<2> 当 TMHEn = 1 时, 开始计数。

<3> 当 8 位定时器计数器 H_n 与 CMP0n 的值相等时, 将产生 INTTMHn 信号, 并对 8 位定时器计数器 H_n 清零 (00H)。

<4> 随后, 以相同的间隔时间重复产生 INTTMHn 信号。若要停止计数操作, 则将 TMHEn 清零(0)。

- 备注
1. 如需了解输出引脚的设置情况, 参见 8.3 (3) 端口模式寄存器 1 (PM1)。
 2. 如需了解允许 INTTMHn 信号中断的情况, 可参见 第十九章 中断功能。
 3. $n = 0, 1$

图 8-10. 间隔定时器/方波输出操作时序 (1/2)

(a) 基本操作 (当 $01H \leq CMP0n \leq FEH$ 时的操作)

<1> 通过将 TMHEn 置 1, 允许计数操作。一旦允许计数, 则在不到 1 个时钟周期的时间内启动计数操作。

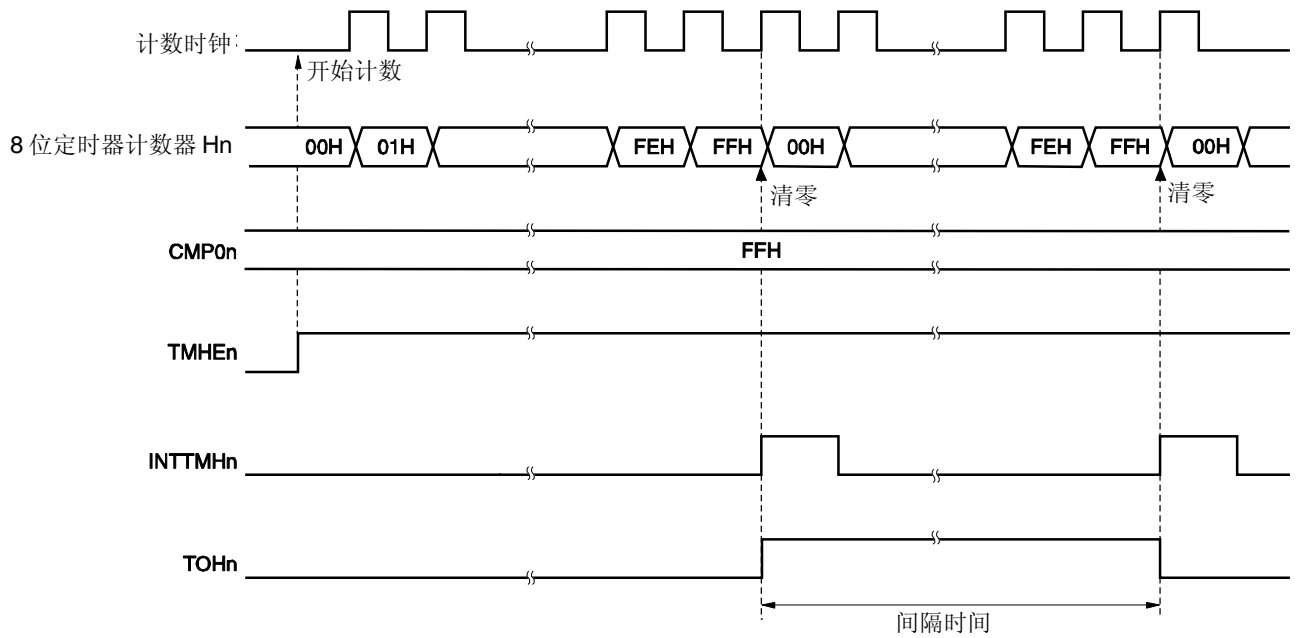
<2> 当 8 位定时器计数器 Hn 与 CMP0n 寄存器的值相等时, 将该定时器计数器清零并反转 TOHn 的输出电平。此外在计数时钟的上升沿输出 INTTMHn 信号。

<3> 在定时器 Hn 操作期间将 TMHEn 位清零, 设置 INTTMHn 信号和 TOHn 的输出为默认电平。如果在将 TMHEn 位清零 (0) 前它们已经处于默认电平, 则电平保持不变。

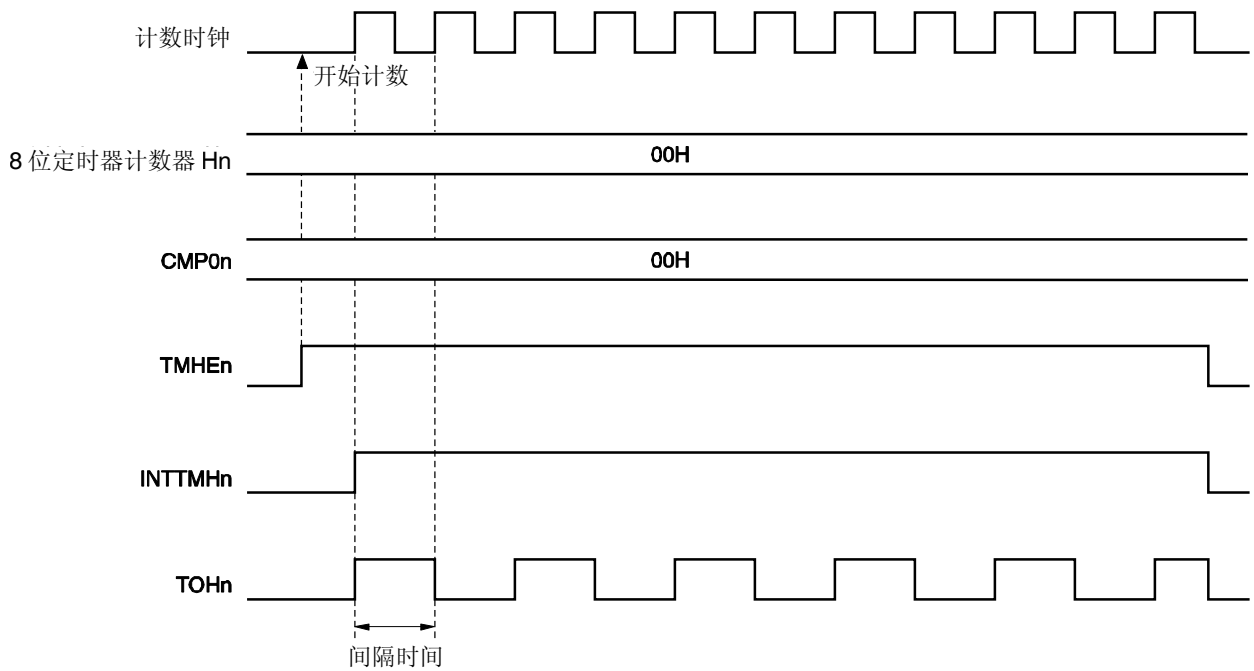
备注 n = 0, 1
 $01H \leq N \leq FEH$

图 8-10. 间隔定时器/方波输出操作时序 (2/2)

(b) 当 CMP0n = FFH 时的操作



(c) 当 CMP0n = 00H 时的操作



备注 n = 0, 1

8.4.2 PWM 输出操作

在 PWM 输出模式中，可以输出具有任意占空比和周期的脉冲。

8 位定时器比较寄存器 0n (CMP0n) 用于控制定时器输出 (TOHn) 的周期。禁止在定时器操作期间修改 CMP0n 寄存器的值。

8 位定时器比较寄存器 1n (CMP1n) 用于控制定时器输出 (TOHn) 的占空比。允许在定时器操作期间修改 CMP1n 寄存器的值。

PWM 输出模式中的操作如下所示。

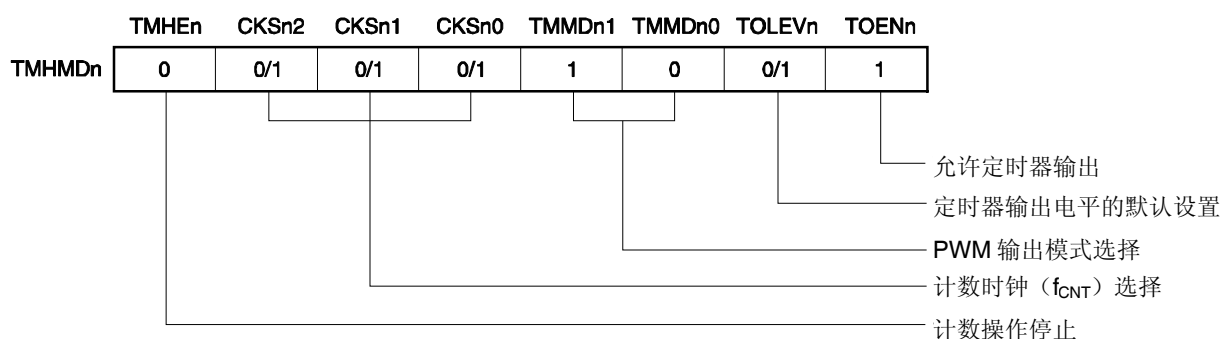
在定时器开始计数后，当 8 位定时器计数器 Hn 与 CMP0n 寄存器的值相等时，反转 TOHn 的输出电平且清零 8 位定时器计数器 Hn。当 8 位定时器计数器 Hn 与 CMP1n 寄存器的值相等时，反转 TOHn 的输出电平。

设置

<1> 寄存器设置。

图 8-11. PWM 输出模式下的寄存器设置

(i) 设置定时器 H 模式寄存器 n (TMHMDn)



(ii) 设置寄存器 CMP0n

- 比较值(N): 周期设置

(iii) 设置寄存器 CMP1n

- 比较值(M): 占空比设置

备注

1. n = 0, 1
2. 00H ≤ CMP1n (M) < CMP0n (N) ≤ FFH

<2> 当 TMHEn = 1 时，开始计数。

<3> CMP0n 是在允许计数操作后首次被比较的比较寄存器。当 8 位定时器计数器 Hn 与 CMP0n 寄存器的值相等时，将 8 位定时器计数器 Hn 清零、产生中断请求信号 (INTTMHn)，并且反转 TOHn 输出电平。同时切换比较寄存器（与 8 位定时器计数器 Hn 比较的寄存器）：由 CMP0n 改为 CMP1n。

<4> 当 8 位定时器计数器 Hn 与 CMP1n 寄存器相等时，反转 TOHn 输出电平，并切换比较寄存器（与 8 位定时器计数器 Hn 比较的寄存器）：由 CMP1n 改为 CMP0n。此时不对 8 位定时器计数器 Hn 清零，也不产生 INTTMHn 信号。

<5> 重复执行过程<3> 和 <4>, 可以获取具有任意占空比的脉冲。

<6> 若要停止计数操作, 则设置 $TMHE_n = 0$ 。

若 $CMP0_n$ 的值为 N 、 $CMP1_n$ 的值为 M , 计数时钟频率为 f_{CNT} , 则 PWM 脉冲的输出周期和占空比如下所示。

- PWM 脉冲输出周期 = $(N + 1)/f_{CNT}$
- 占空比 = $(M + 1)/(N + 1)$

注意事项

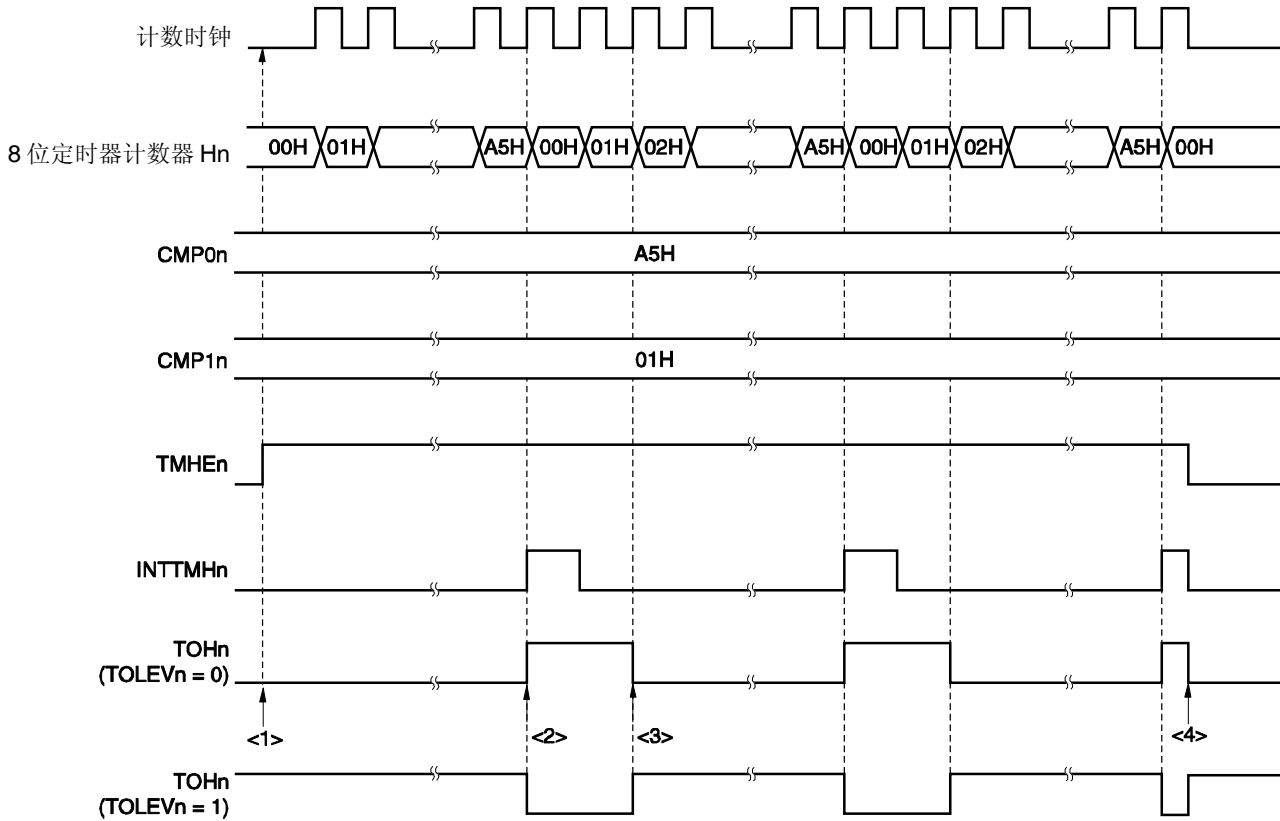
1. 可以在定时器计数操作时改变 $CMP1_n$ 的设置值。但从改变 $CMP1_n$ 寄存器的值到这个值被传输给寄存器需要经历三个操作时钟（通过 $TMHMD_n$ 寄存器的 $CKSn_2 \sim CKSn_0$ 位选择的时钟信号）
2. 在定时器计数操作停止后（ $TMHE_n=0$ ），再次启动定时器操作时（ $TMHE_n=1$ ）,必须对 $CMP1_n$ 进行设置（即使是设置相同的值，也必须重新设置 $CMP1_n$ 寄存器）
3. $CMP1_n$ 寄存器的设置值（ M ）和 $CMP0_n$ 寄存器的设置值（ N ）必须在以下范围内取值。
 $00H \leq CMP1_n (M) < CMP0_n (N) \leq FFH$

备注

1. 如需了解输出引脚的设置, 参见 8.3 (3) 端口模式寄存器 1 (PM1)。
2. 要了解如何允许 $INTTMH_n$ 信号中断的情况, 可参见 第十九章 中断功能。
3. $n = 0, 1$

图 8-12. PWM 输出模式下的操作时序 (1/4)

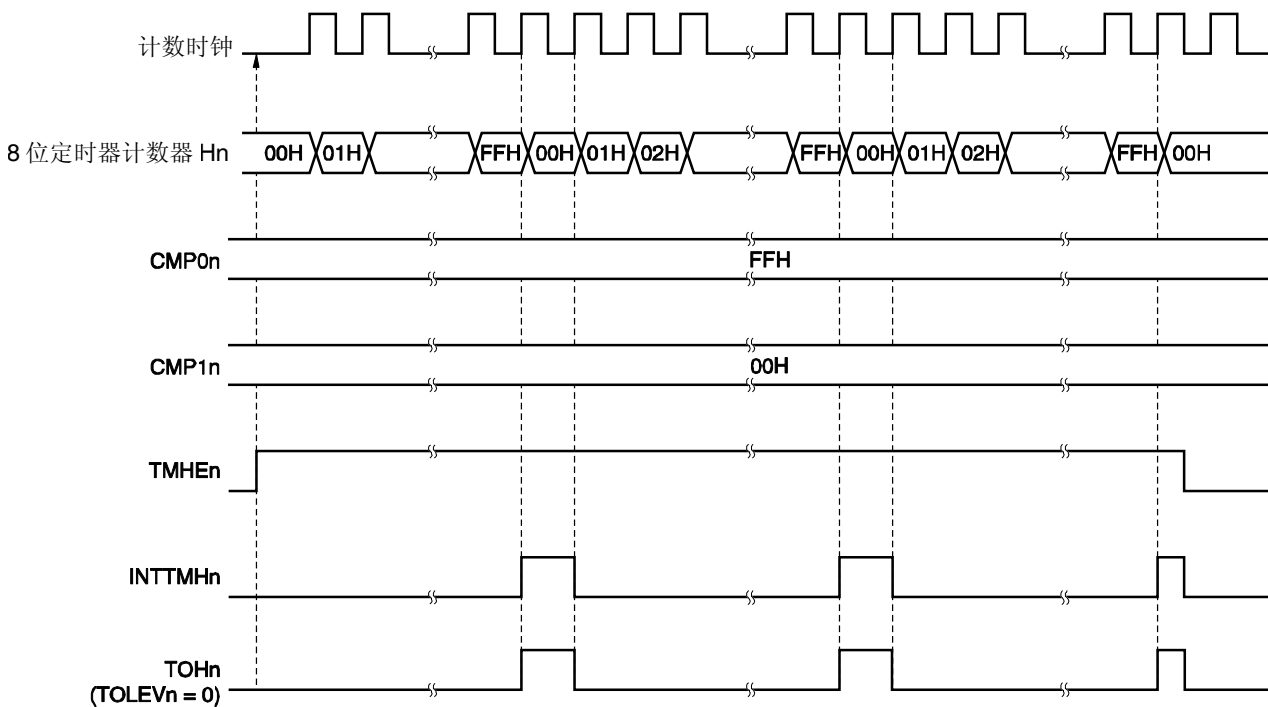
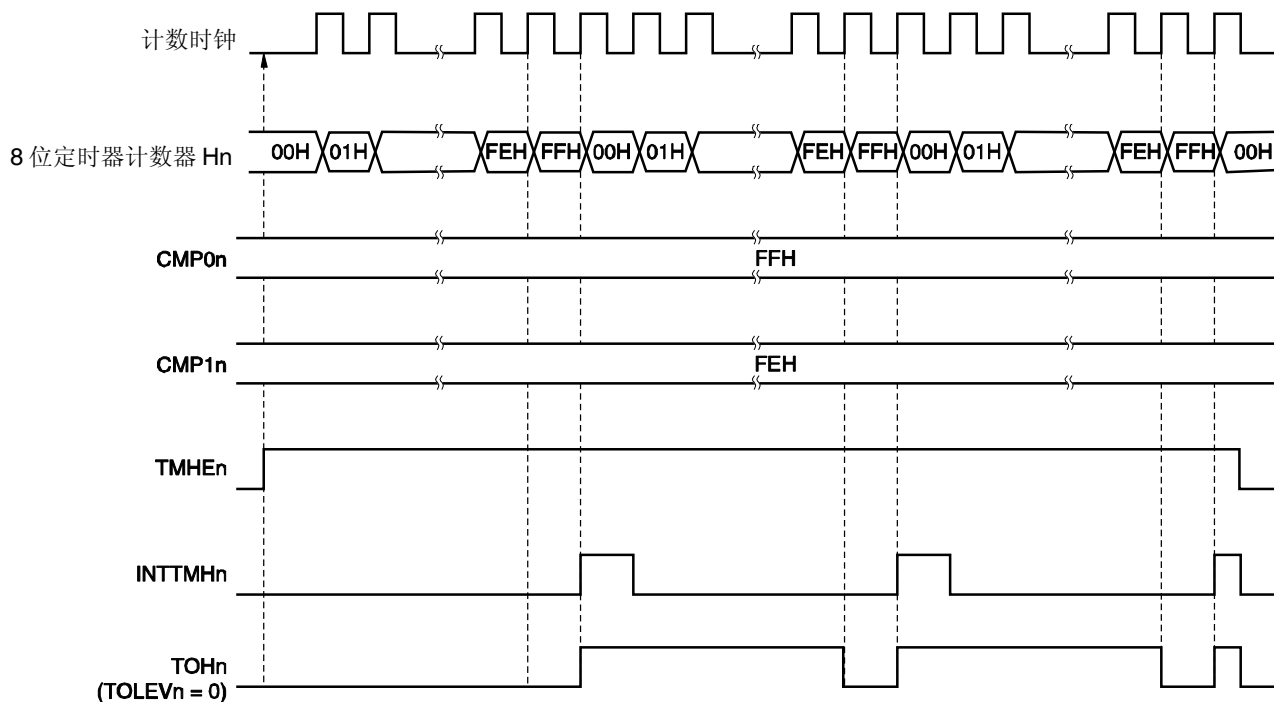
(a) 基本操作



- <1> 通过设置 $TMHEn = 1$ ，允许计数操作。通过屏蔽一个计数时钟，启动 8 位定时器计数器 H_n 。此时， TOH_n 输出保持默认电平状态。
- <2> 当 8 位定时器计数器 H_n 与 $CMP0_n$ 相等时，将 TOH_n 的输出电平反转，对 8 位定时器计数器 H_n 清零，并输出 $INTTMH_n$ 中断信号。
- <3> 当 8 位定时器计数器 H_n 与 $CMP1_n$ 相等时，反转 TOH_n 的输出电平。而此时并不对 8 位定时器计数器 H_n 清零，也不输出 $INTTMH_n$ 中断信号。
- <4> 在定时器 H_n 操作期间若将 $TMHEn$ 位清零，可以设置 $INTTMH_n$ 信号和 TOH_n 输出电平为默认值。

备注 $n = 0, 1$

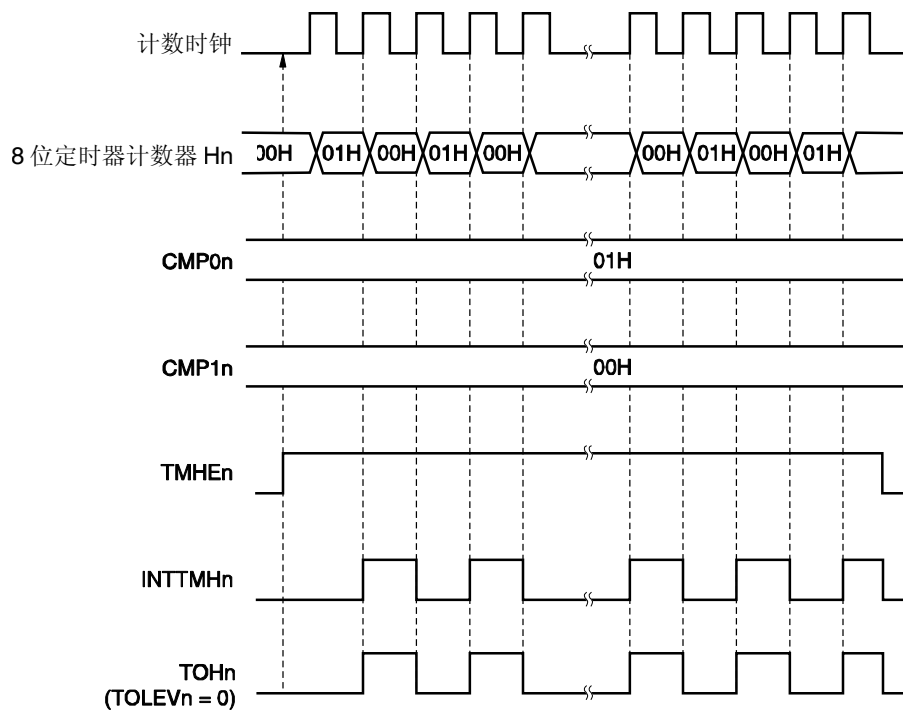
图 8-12. PWM 输出模式下的操作时序 (2/4)

(b) 当 $CMP0n = FFH$, $CMP1n = 00H$ 时的操作(c) 当 $CMP0n = FFH$, $CMP1n = FEH$ 时的操作

备注 n = 0, 1

图 8-12. PWM 输出模式下的操作时序 (3/4)

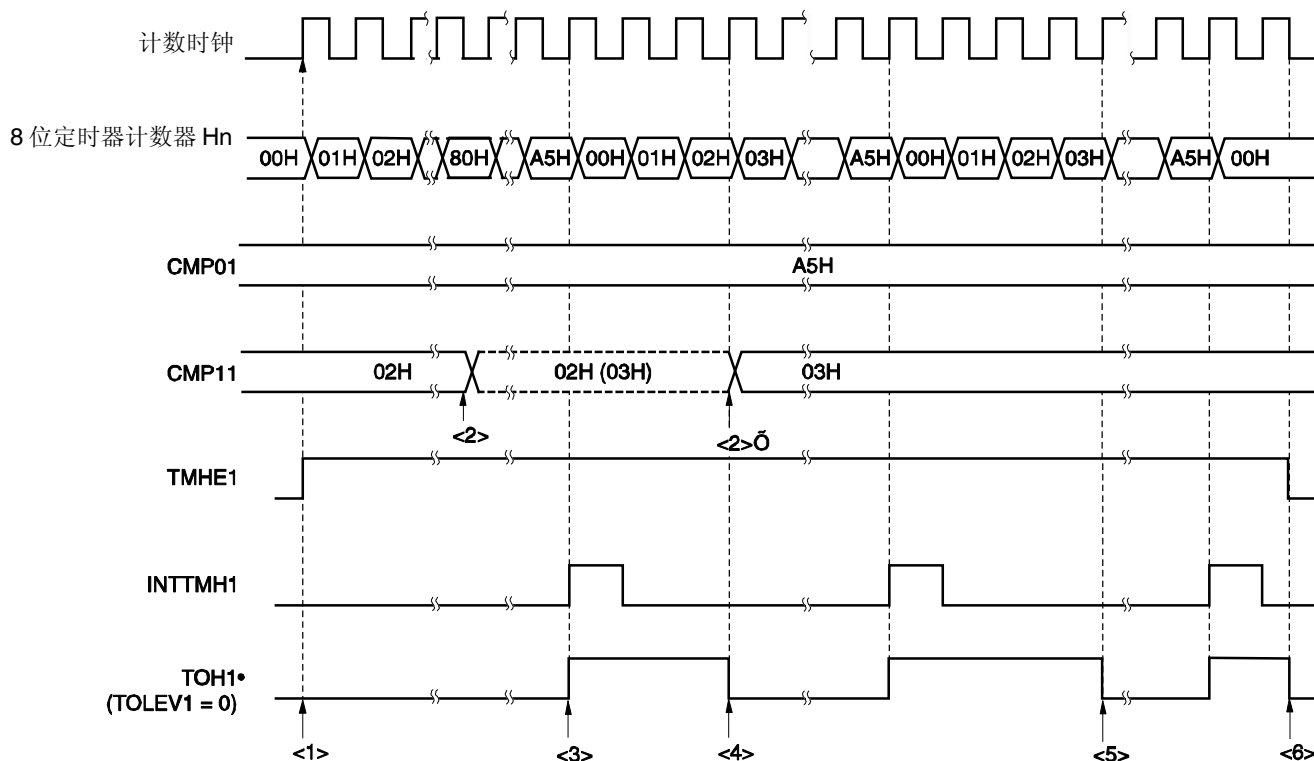
(d) 当 $CMP0n = 01H$, $CMP1n = 00H$ 时的操作



备注 n = 0, 1

图 8-12. PWM 输出模式下的操作时序 (4/4)

(e) 修改 CMP1n 的时序操作 (CMP1n = 02H → 03H, CMP0n = A5H)



- <1> 通过设置 TMHE_n = 1 允许计数操作。通过屏蔽 1 个计数时钟，启动 8 位定时器计数器 H_n。此时 TOH_n 输出保持默认电平状态。
- <2> 在定时器计数器操作期间可以修改 CMP1_n 的值。修改操作与计数时钟不同步。
- <3> 当 8 位定时器计数器 H_n 与 CMP0_n 的值相等时，将 8 位定时器计数器 H_n 的值清零、反转 TOH_n 输出电平，并输出 INTTMH_n 信号。
- <4> 如果 CMP1_n 的值被修改，则修改后的值被锁存，且不传送到寄存器。当 8 位定时器计数器 H_n 的值与修改前 CMP1_n 寄存器的值相等时，则将值传送到 CMP1_n，这样 CMP1_n 的值就被修改了 (<2>)。但从 CMP1_n 的值被修改到将值传送到寄存器至少需要 3 个计数时钟。如果在 3 个计数时钟内产生一个相等信号，则不能将修改后的值传送到寄存器。
- <5> 当 8 位定时器计数器 H_n 的值与修改后的 CMP1_n 的值相等时，反转 TOH_n 输出电平。此时不对 8 位定时器计数器 H_n 清零，也不产生 INTTMH_n 信号。
- <6> 在定时器 H_n 操作期间若将 TMHE_n 位清零，可以设置 INTTMH_n 信号和 TOH_n 输出电平为默认值。

备注 n = 0, 1

8.4.3 载波发生器操作（仅限于 8 位定时器 H1）

在载波发生器模式中 8 位定时器 H1 用于产生红外遥控器的载波信号，8 位定时器/事件计数器 51 用于红外遥控信号的产生（计时）

以 8 位定时器/事件计数器 51 设置的周期输出载波时钟。该载波时钟由 8 位定时器 H1 产生。

在载波发生器模式中，8 位定时器 H1 载波脉冲由 8 位定时器/事件计数器 51 控制输出，并从 TOH1 输出。

(1) 载波的发生

在载波发生器模式中，8 位定时器 H 比较寄存器 01（CMP01）产生低电平宽度的载波脉冲波形，而 8 位定时器 H 比较寄存器 11（CMP11）产生高电平宽度的载波脉冲波形。

在 8 位定时器 H1 操作期间允许修改 CMP11，但禁止修改 CMP01。

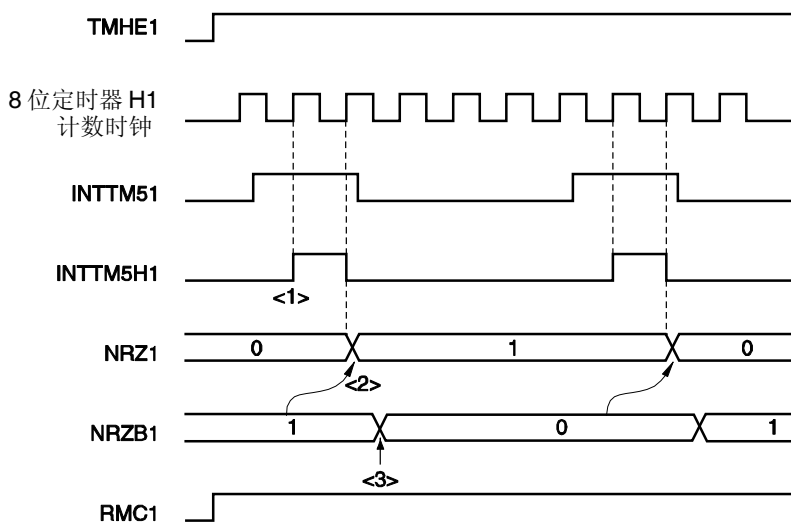
(2) 载波输出控制

载波输出由 8 位定时器/事件计数器 51 的中断请求信号（INTTM51）和 8 位定时器 H 载波控制寄存器（TMCYC1）的 NRZB1 和 RMC1 位控制。输出之间的关系如下所示。

RMC1 位	NRZB1 位	输出
0	0	低电平输出
0	1	高电平输出
1	0	低电平输出
1	1	载波脉冲输出

为了在计数操作期间控制载波脉冲输出，TMCYC1 寄存器的 NRZ1 和 NRZB1 位有 1 个主位和从位配置。NRZ1 位只读，而 NRZB1 位可读可写。INTTM51 信号与 8 位定时器 H1 计数时钟同步，且作为 INTTM5H1 信号输出。INTTM5H1 信号作为 NRZ1 位的数据传送信号，且 NRZB1 位的值被传送至 NRZ1 位。从 NRZB1 位传送至 NRZ1 位的时序如下所示。

图 8-13. 传送时序



- <1> INTTM51 信号与 8 位定时器 H1 的计数时钟同步，并作为 INTTM5H1 信号输出。
- <2> 在 INTTM5H1 信号上升沿的第 2 个时钟处将 NRZB1 位的值传送至 NRZ1 位。
- <3> 在中断服务程序中将下一个值写入 NRZB1 位。该中断服务程序已由 INTTM5H1 中断启动或通过轮询中断请求标志检测到时序后启动。将下一次要计数的值写入 CR51 中。

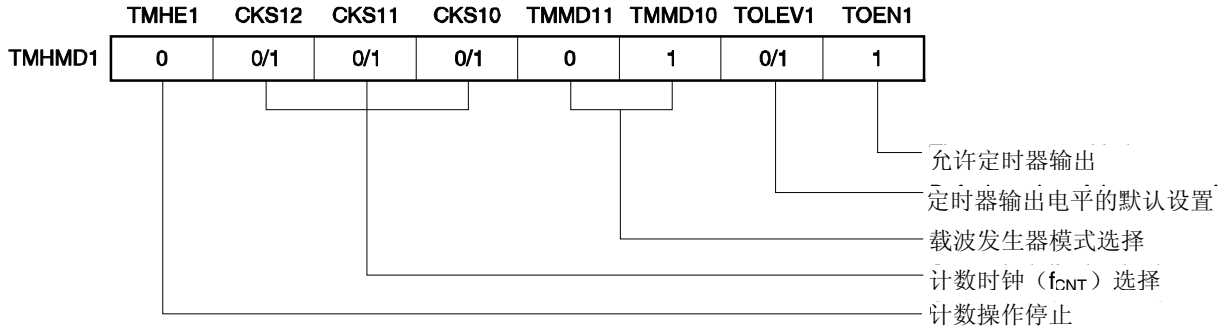
- 注意事项**
1. NRZB1 位被重写后至少到第 2 个时钟时才能被再次重写，否则，从 NRZB1 位到 NRZ1 位的传送不能得到保证。
 2. 在载波发生器模式中使用 8 位定时器/事件计数器 51 时，在<1>时刻将产生中断。如果在其他模式中使用 8 位定时器/事件计数器 51，中断产生的时序则不相同。

设置

<1> 寄存器设置

图 8-14. 在载波发生器模式下的寄存器设置

(i) 设置 8 位定时器 H 模式寄存器 1 (TMHMD1)



(ii) CMP01 寄存器设置

- 比较值

(iii) CMP11 寄存器设置

- 比较值

(iv) TMCYC1 寄存器设置

- RMC1 = 1 ... 遥控输出允许位
- NRZB1 = 0/1 ... 载波输出允许位

(v) TCL51 和 TMC51 寄存器设置

- 参见 7.3 控制 8 位定时器/事件计数器 50 和 51 的寄存器。

<2> 当 TMHE1 = 1 时，8 位定时器 H1 开始计数。

<3> 当 8 位定时器模式控制寄存器 51 (TMC51) 的 TCE51=1 时，8 位定时器/事件计数器 51 开始计数。

<4> 允许计数后，第 1 个用于比较的比较寄存器是 CMP01。当 8 位定时器计数器 H1 的计数值与 CMP01 的值相等时，将产生 INTTMH1 信号，并对 8 位定时器计数器 H1 清零，而此时与 8 位定时器计数器 H1 比较的比较寄存器从 CMP01 被切换为 CMP11。

<5> 当 8 位定时器计数器 H1 的计数值与 CMP11 寄存器的值相等时，将产生 INTTMH1 信号，并对 8 位定时器计数器 H1 清零，而此时与 8 位定时器计数器 H1 比较的比较寄存器从 CMP11 被切换为 CMP01。

<6> 重复执行过程<4>和<5>，将产生 1 个载波时钟。

<7> INTTM51 信号与 8 位定时器 H1 的计数时钟同步，并作为 INTTM5H1 信号输出。INTTM5H1 信号作为 NRZB1 位的数据传输信号，并将 NRZB1 位的值传送至 NRZ1 位。

<8> 在中断服务程序中将下一个值写入 NRZB1 位。该中断服务程序已由 INTTM5H1 中断启动或通过轮询中断请求标志检测到时序后启动。将下一次要计数的值写入 CR51 中。

<9> 当 NRZ1 位是高电平时，将从 TOH1 引脚输出载波时钟。

<10> 通过执行上述过程，可以获得任意形式的载波时钟。若要停止计数操作，可以将 TMHE1 清零。

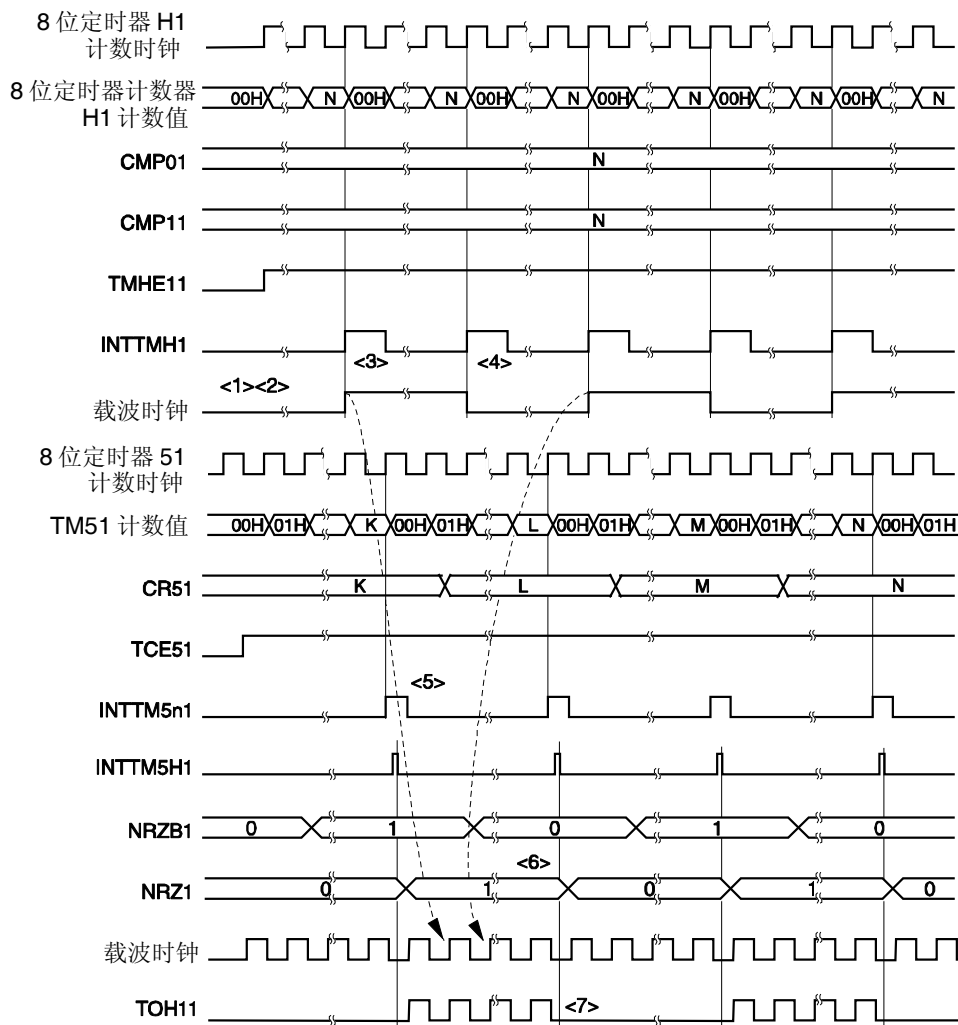
如果 CMP01 的值为 N、CMP11 的值为 M、计数时钟频率为 f_{CNT} ，则载波时钟输出周期与占空比如下所示。

- 载波时钟输出周期 = $(N + M + 2)/f_{CNT}$
- 占空比=高电平宽度/载波时钟输出宽度 = $(M + 1) / (N + M + 2)$

- 注意事项**
1. 在定时器计数操作停止后（TMHE1=0），再启动定时器操作（TMHE1=1）时，必须设置 CMP11（即使是相同的值，也必须对 CMP11 重新设置）
 2. 设置 TMH1 的计数时钟频率至少是 TM51 的计数时钟频率的 6 倍。
 3. CMP01 和 CMP11 的值必须在 01H 和 FFH 之间。
 4. 定时器计数期间可以改变 CMP11 寄存器的值。但是，从改变 CMP11 的值到将该值传输到寄存器需要三个操作时钟（由 TMHMD1 寄存器的 CKS12~CKS10 位选择的时钟信号）
 5. 在计时操作开始前必须设置 RMC1。

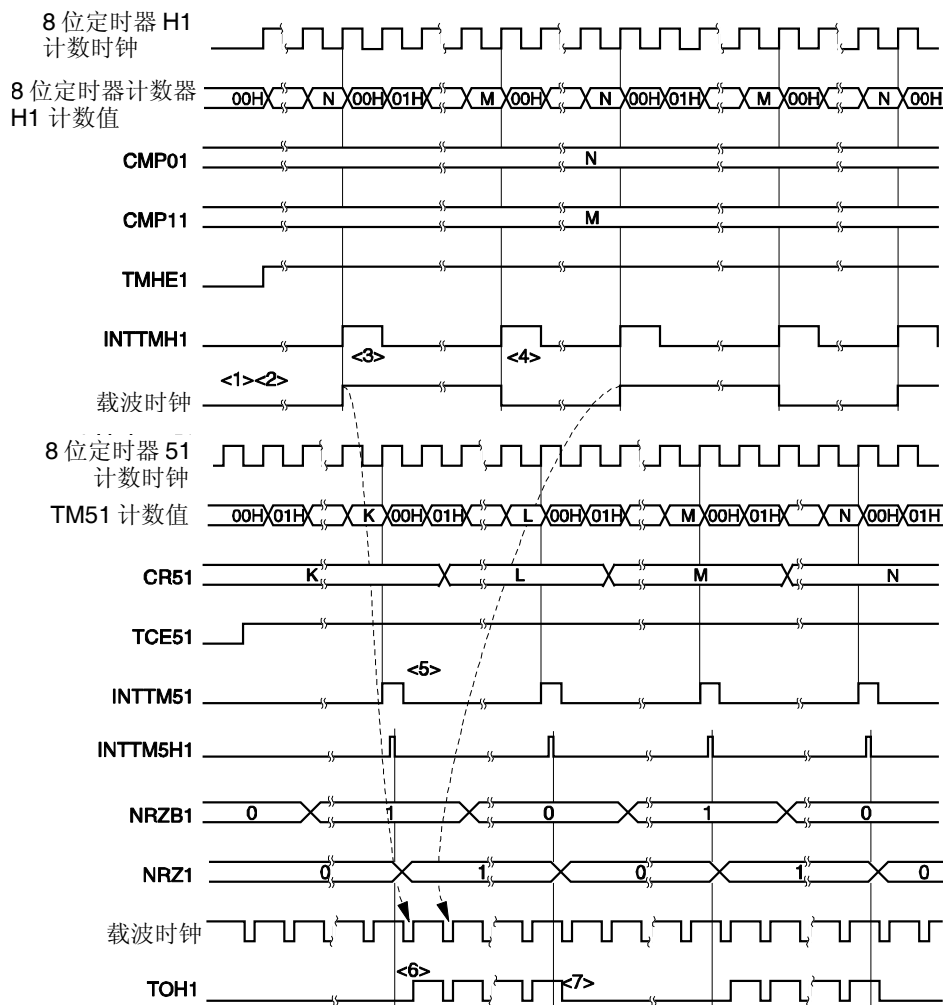
- 备注**
1. 如需了解输出引脚的设置，参见 8.3 (3) 端口模式寄存器 1 (PM1)
 2. INTTMH1 中断信号的使用参见 第十九章 中断功能。

图 8-15. 载波发生器模式操作时序 (1/3)

(a) 当 $CMP01 = N$, $CMP11 = N$ 时的操作

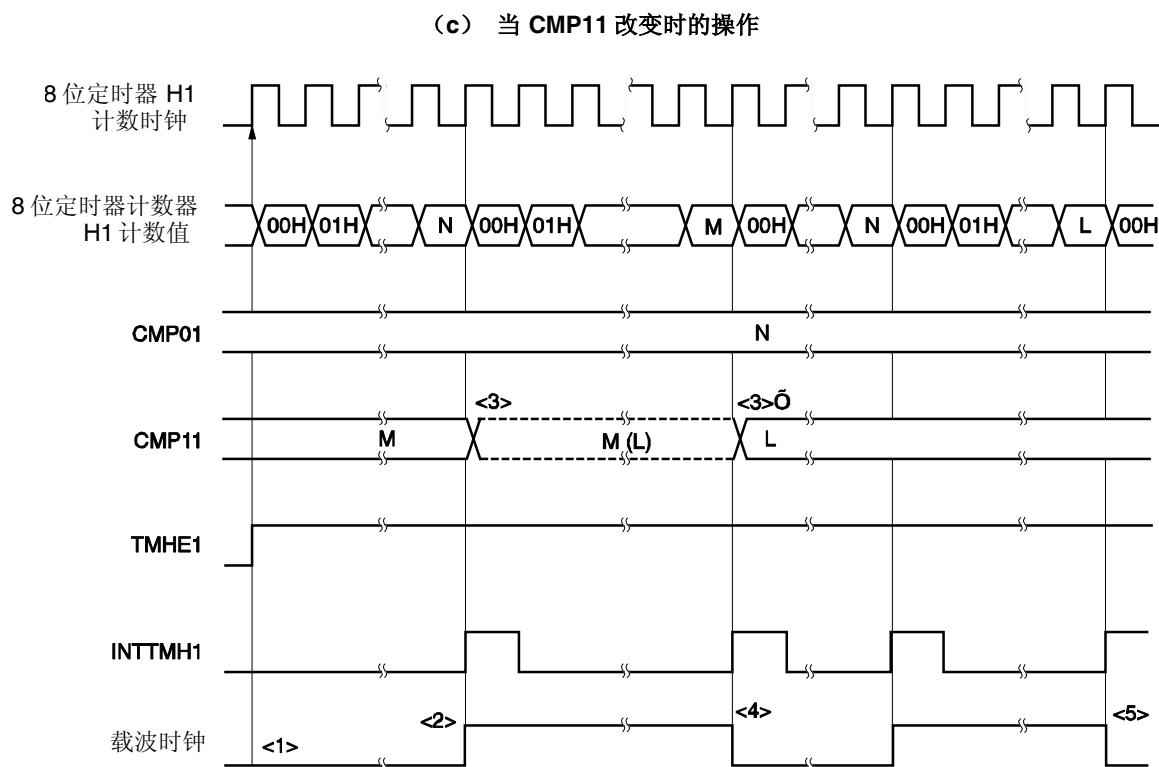
- <1> 当 $TMHE1 = 0$ 且 $TCE51 = 0$ 时, 停止 8 位定时器计数器 H1 的操作。
- <2> 当 $TMHE1 = 1$ 时, 8 位定时器计数器 H1 开始计数。此时载波时钟保持默认电平状态。
- <3> 当 8 位定时器计数器 H1 的计数值与 $CMP01$ 的值相等时, 产生第 1 个 $INTTMH1$ 信号并反转载波时钟信号, 同时与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP01$ 切换为 $CMP11$ 。8 位定时器计数器 H1 的值被清零 ($00H$)。
- <4> 当 8 位定时器计数器 H1 的计数值与 $CMP11$ 的值相等时, 产生 $INTTMH1$ 信号并反转载波时钟信号, 同时与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP11$ 切换为 $CMP01$ 。8 位定时器计数器 H1 的值被清零 ($00H$)。重复执行过程<3>和<4>, 将产生 1 个载波时钟, 且占空比恒为 50%。
- <5> 当 $INTTM51$ 信号产生时, 它与 8 位定时器 H1 计数时钟同步, 并作为 $INTTM5H1$ 信号输出。
- <6> $INTTM5H1$ 信号用作 $NRZB1$ 位的数据传输信号, 并将 $NRZB1$ 位的值传送至 $NRZ1$ 位。
- <7> 设置 $NRZ1 = 0$ 时, $TOH11$ 输出低电平。

图 8-15. 载波发生器模式操作的时序 (2/3)

(b) 当 $CMP01 = N$, $CMP11 = M$ 时的操作

- <1> 当 $TMHE1 = 0$ 且 $TCE51 = 0$ 时，停止 8 位定时器计数器 H1 的操作。
- <2> 当 $TMHE1 = 1$ 时，8 位定时器计数器 H1 开始计数。此时载波时钟保持默认电平状态。
- <3> 当 8 位定时器计数器 H1 的计数值与 $CMP01$ 的值相等时，产生第 1 个 $INTTMH1$ 信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP01$ 切换为 $CMP11$ 。8 位定时器计数器 H1 的值被清零 (00H)。
- <4> 当 8 位定时器计数器 H1 的计数值与 $CMP11$ 的值相等时，产生 $INTTMH1$ 信号并反转载波时钟信号，同时与 8 位定时器计数器 H1 比较的比较寄存器从 $CMP11$ 切换为 $CMP01$ 。8 位定时器计数器 H1 的值被清零 (00H)。重复执行过程<3>和<4>，将产生 1 个载波时钟，且占空比不等于 50%。
- <5> 当 $INTTM51$ 信号产生时，它与 8 位定时器 H1 计数时钟同步，并作为 $INTTM5H1$ 信号输出。
- <6> 若 $NRZ1=1$ ，在载波时钟的第 1 个上升沿处输出 1 个载波信号。
- <7> 当 $NRZ1 = 0$ 时， $TOH1$ 输出保持高电平状态，在载波时钟为高电平时（在过程<6>和<7>中，保证载波时钟波形的高电平宽度）不会转变为低电平。

图 8-15. 载波发生器模式操作的时序 (3/3)



- <1> 当 $TMHE1 = 1$ 时，8 位定时器 H1 开始计数。此时载波时钟保持默认电平状态。
- <2> 当 8 位定时器计数器 H1 的计数值与 CMP01 的值相等时，输出 INTTMH1 信号、反转载波信号，并将定时器计数器清零(00H)。同时与 8 位定时器计数器 H1 比较的比较寄存器从 CMP01 切换为 CMP11。
- <3> CMP11 与计数时钟不同步，在 8 位定时器 H1 操作期间可以修改 CMP11 的值，但修改后的新值(L)被锁存。当 8 位定时器计数器 H1 的计数值与 CMP11 修改前的值(M)相等(<3>)时，修改 CMP11 的值(<3>)。
但从改变 CMP11 的值到将该值传送到寄存器需要 3 个计数时钟。即使在 3 个计数时钟内产生相等信号，新值不会被传送到寄存器
- <4> 当 8 位定时器计数器 H1 的计数值与 CMP11 修改前的值(M)相等时，输出 INTTMH1 信号、反转载波信号，并将定时器计数器清零(00H)。同时与 8 位定时器计数器 H1 比较的比较寄存器从 CMP11 切换为 CMP01。
- <5> 由修改后的值(L)确定 8 位定时器计数器 H1 的计数值与 CMP11 的值再次相等时的时序。

9.1 钟表定时器的功能

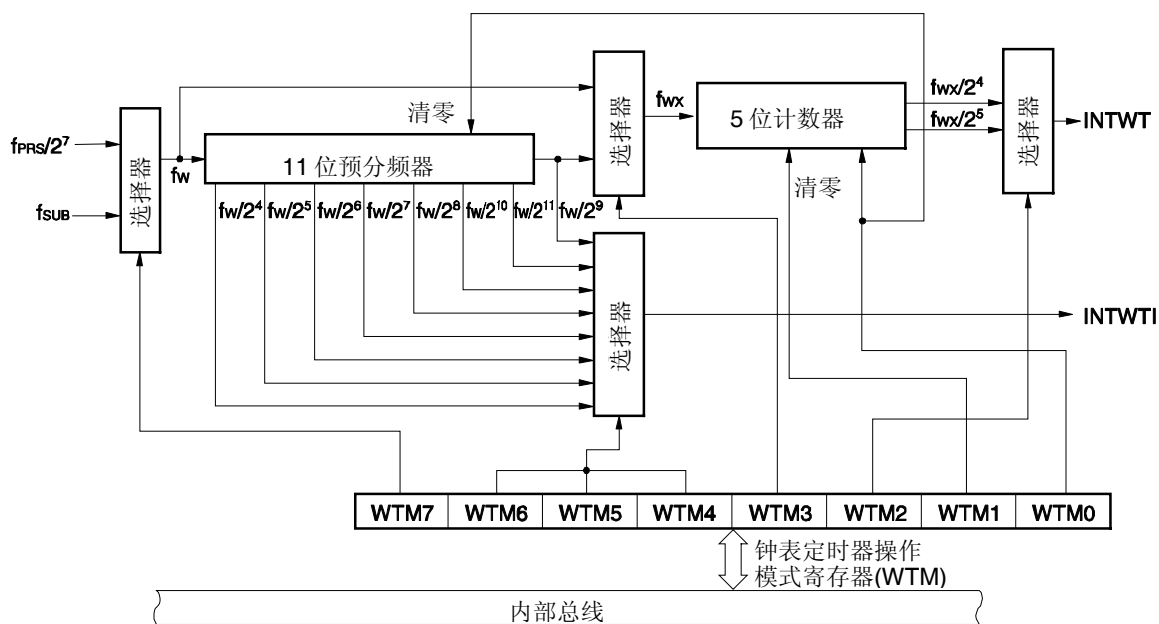
钟表定时器有以下功能。

- 钟表定时器
- 间隔定时器

钟表定时器和间隔定时器可以同时使用。

图 9-1 为钟表定时器框图。

图 9-1. 钟表定时器框图



备注

- f_{PRS} : 外围硬件时钟频率
- f_{SUB} : 副系统时钟频率
- f_w : 钟表定时器时钟频率 ($f_{PRS}/2^7$ 或 f_{SUB})
- f_{wx} : f_w 或 $f_w/2^9$

(1) 钟表定时器

当使用外围硬件时钟或副系统时钟时，以预设的时间间隔产生中断请求信号 (INTWT)。

表 9-1. 钟表定时器中断时间

中断时间	操作频率 $f_{SUB} = 32.768 \text{ kHz}$	操作频率 $f_{PRS} = 2 \text{ MHz}$	操作频率 $f_{PRS} = 5 \text{ MHz}$	操作频率 $f_{PRS} = 10 \text{ MHz}$	操作频率 $f_{PRS} = 20 \text{ MHz}$
$2^4/f_w$	488 μs	1.02 ms	410 μs	205 μs	102 μs
$2^5/f_w$	977 μs	2.05 ms	819 μs	410 μs	205 μs
$2^{13}/f_w$	0.25 s	0.52 s	0.210 s	0.105 s	52.5 ms
$2^{14}/f_w$	0.5 s	1.05 s	0.419 s	0.210 s	0.105 s

备注 f_{PRS} : 外围硬件时钟频率
 f_{SUB} : 副系统时钟频率
 f_w : 钟表定时器时钟频率 ($f_{PRS}/2^7$ 或 f_{SUB})

(2) 间隔定时器

以预设的时间间隔产生中断请求信号 (INTWTI)。

表 9-2. 间隔定时器间隔时间

间隔时间	操作频率 $f_{SUB} = 32.768 \text{ kHz}$	操作频率 $f_{PRS} = 2 \text{ MHz}$	操作频率 $f_{PRS} = 5 \text{ MHz}$	操作频率 $f_{PRS} = 10 \text{ MHz}$	操作频率 $f_{PRS} = 20 \text{ MHz}$
$2^4/f_w$	488 μs	1.02 ms	410 μs	205 μs	102 μs
$2^5/f_w$	977 μs	2.05 ms	820 μs	410 μs	205 μs
$2^6/f_w$	1.95 ms	4.10 ms	1.64 ms	820 μs	410 μs
$2^7/f_w$	3.91 ms	8.20 ms	3.28 ms	1.64 ms	820 μs
$2^8/f_w$	7.81 ms	16.4 ms	6.55 ms	3.28 ms	1.64 ms
$2^9/f_w$	15.6 ms	32.8 ms	13.1 ms	6.55 ms	3.28 ms
$2^{10}/f_w$	31.3 ms	65.5 ms	26.2 ms	13.1 ms	6.55 ms
$2^{11}/f_w$	62.5 ms	131.1 ms	52.4 ms	26.2 ms	13.1 ms

备注 f_{PRS} : 外围硬件时钟频率
 f_{SUB} : 副系统时钟频率
 f_w : 钟表定时器时钟频率 ($f_{PRS}/2^7$ 或 f_{SUB})

9.2 钟表定时器的配置

钟表定时器包括以下硬件。

表 9-3. 钟表定时器的配置

项目	配置
计数器	5 位 \times 1
预分频器	11 位 \times 1
控制寄存器	钟表定时器操作模式寄存器 (WTM)

9.3 控制钟表定时器的寄存器

由钟表定时器操作模式寄存器 (WTM) 控制钟表定时器。

- 钟表定时器操作模式寄存器 (WTM)

该寄存器用于设置钟表定时器计数时钟、允许/禁止操作、预分频器间隔时间和 5 位计数器操作控制。

可以由 1 位或 8 位存储器操作指令设置 WTM。

复位信号的产生将 WTM 清零 (00H)。

图 9-2. 钟表定时器操作模式寄存器 (WTM) 的格式

地址: FF6FH 复位后: 00H R/W

符号	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	钟表定时器计数时钟选择 (fw)					
		f _{SUB} = 32.768 kHz	f _{PRS} = 2 MHz	f _{PRS} = 5 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz
0	f _{PRS} /2 ⁷	–	15.625 kHz	39.062 kHz	78.125 kHz	156.25 kHz
1	f _{SUB}	32.768 kHz	–			

WTM6	WTM5	WTM4	预分频器间隔时间选择
0	0	0	2 ⁴ /fw
0	0	1	2 ⁵ /fw
0	1	0	2 ⁶ /fw
0	1	1	2 ⁷ /fw
1	0	0	2 ⁸ /fw
1	0	1	2 ⁹ /fw
1	1	0	2 ¹⁰ /fw
1	1	1	2 ¹¹ /fw

WTM3	WTM2	钟表定时器中断时间选择
0	0	2 ¹⁴ /fw
0	1	2 ¹³ /fw
1	0	2 ⁵ /fw
1	1	2 ⁴ /fw

WTM1	5 位计数器操作控制
0	操作停止后清零
1	启动操作

WTM0	钟表定时器操作允许
0	停止操作(预分频器和 5 位计数器清零)
1	允许操作

注意事项 在钟表定时器操作期间不要修改计数时钟和间隔时间（通过设置 WTM 的第 4 位 ~ 第 7 位 (WTM4 ~ WTM7) 实现）。

- 备注**
1. f_W : 钟表定时器时钟频率($f_{PRS} / 2^7$ 或 f_{SUB})
 2. f_{PRS} : 外围硬件时钟频率
 3. f_{SUB} : 副系统时钟频率

9.4 钟表定时器操作

9.4.1 钟表定时器操作

通过使用外围硬件时钟或副系统时钟，钟表定时器以指定的时间间隔产生中断请求信号 (INTWT)。

当钟表定时器操作模式寄存器 (WTM) 的第 0 位 (WTM0) 和第 1 位 (WTM1) 为 1 时，开始计数操作。当这些位被设置为零时，将 5 位计数器清零，同时停止计数操作。

在间隔定时器操作的同时，通过将 WTM1 清零，钟表定时器的零秒启动被激活。但在这种情况下，不对 11 位预分频器清零。因此，在零秒启动后的第 1 次溢出 (INTWT) 时会产生 $1 \times 2^9 \times 1/f_w$ 秒的误差。

按以下时间间隔产生中断请求。

表 9-4. 钟表定时器中断时间

WTM3	WTM2	中断时间选择	操作频率				
			$f_{SUB} = 32.768 \text{ kHz}$ (WTM7 = 1)	$f_{PRS} = 2 \text{ MHz}$ (WTM7 = 0)	$f_{PRS} = 5 \text{ MHz}$ (WTM7 = 0)	$f_{PRS} = 10 \text{ MHz}$ (WTM7 = 0)	$f_{PRS} = 20 \text{ MHz}$ (WTM7 = 0)
0	0	$2^{14}/f_w$	0.5 s	1.05 s	0.419 s	0.210 s	0.105 s
0	1	$2^{13}/f_w$	0.25 s	0.52 s	0.210 s	0.105 s	52.5 ms
1	0	$2^5/f_w$	977 μs	2.05 ms	819 μs	410 μs	205 μs
1	1	$2^4/f_w$	488 μs	1.02 ms	410 μs	205 μs	102 μs

- 备注
1. f_w : 钟表定时器时钟频率($f_{PRS}/2^7$ 或 f_{SUB})
 2. f_{PRS} : 外围硬件时钟频率
 3. f_{SUB} : 副系统时钟频率

9.4.2 间隔定时器操作

钟表定时器用作间隔定时器时，以预设的时间间隔重复产生中断请求 (INTWTI)。

可以由钟表定时器操作模式寄存器 (WTM) 的第 4 位 ~ 第 6 位 (WTM4 ~ WTM6) 选择间隔时间。

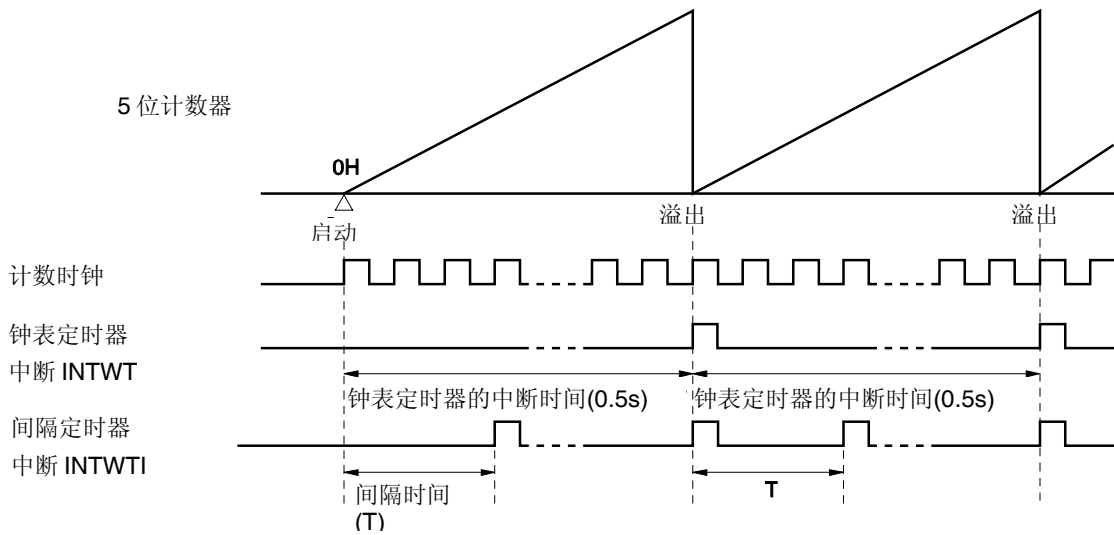
当 WTM 的第 0 位 (WTM0) 为 1 时，开始计数。而当该位被设置为零时，停止计数。

表 9-5. 间隔定时器间隔时间

WTM6	WTM5	WTM4	间隔时间	操作频率				
				$f_{SUB} = 32.768 \text{ kHz}$ (WTM7 = 1)	$f_{PRS} = 2 \text{ MHz}$ (WTM7 = 0)	$f_{PRS} = 5 \text{ MHz}$ (WTM7 = 0)	$f_{PRS} = 10 \text{ MHz}$ (WTM7 = 0)	$f_{PRS} = 20 \text{ MHz}$ (WTM7 = 0)
0	0	0	$2^4/f_w$	488 μs	1.02 ms	410 μs	205 μs	102 μs
0	0	1	$2^5/f_w$	977 μs	2.05 ms	820 μs	410 μs	205 μs
0	1	0	$2^6/f_w$	1.95 ms	4.10 ms	1.64 ms	820 μs	410 μs
0	1	1	$2^7/f_w$	3.91 ms	8.20 ms	3.28 ms	1.64 ms	820 μs
1	0	0	$2^8/f_w$	7.81 ms	16.4 ms	6.55 ms	3.28 ms	1.64 ms
1	0	1	$2^9/f_w$	15.6 ms	32.8 ms	13.1 ms	6.55 ms	3.28 ms
1	1	0	$2^{10}/f_w$	31.3 ms	65.5 ms	26.2 ms	13.1 ms	6.55 ms
1	1	1	$2^{11}/f_w$	62.5 ms	131.1 ms	52.4 ms	26.2 ms	13.1 ms

- 备注
1. f_w : 钟表定时器时钟频率($f_{PRS}/2^7$ 或 f_{SUB})
 2. f_{PRS} : 外围硬件时钟频率
 3. f_{SUB} : 副系统时钟频率

图 9-3. 钟表定时器/间隔定时器的操作时序



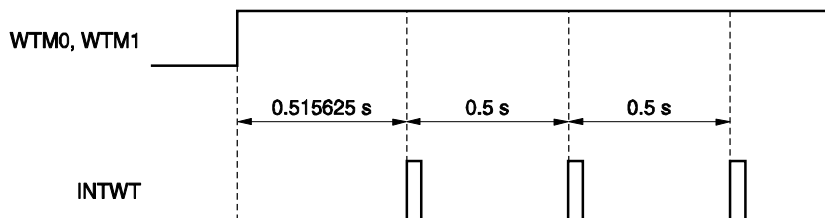
备注 f_w : 钟表定时器时钟频率
 图中括号里的值为 $f_w = 32.768 \text{ kHz}$ ($WTM7 = 1, WTM3, WTM2 = 0, 0$) 时的操作结果

9.5 钟表定时器使用注意事项

通过钟表定时器模式控制寄存器 (WTM) (WTM 的第 0 位 (WTM0) 和第 1 位 (WTM1) 置 1) 允许钟表定时器和 5 位计数器操作时, 从寄存器被设置后到第 1 个中断请求 (INTWT) 产生的这段间隔时间与 WTM 的第 2 位和第 3 位 (WTM2 和 WTM3) 指定的间隔时间并不完全相等。但其后的 INTWT 信号会以指定的时间间隔产生。

图 9-4. 钟表定时器中断请求信号 (INTWT) 产生示例 (当中断周期 = 0.5 s)

经过 0.515625 秒产生第 1 个 INTWT ($2^9 \times 1/32768 = 0.015625 \text{ s}$)。
 随后 INTWT 每 0.5 秒产生 1 次。



第十章 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允许寄存器 (WDTE) 执行 1 位操作指令。
- 如果将“ACH”以外的数据写入 WDTE
- 如果在窗口关闭期间将数据写入 WDTE
- 如果从没有通过 IMS 和 IXS 寄存器设置的区域获取指令 (在 CPU 挂起时检查是否有无效校验)
- 如果因为执行 1 条读/写指令，CPU 访问了没有通过 IMS 和 IXS 寄存器设置的区域 (FB00H ~ FFFFH 除外) (CPU 程序循环期间检测是否有非法访问)

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 (WDTRF) 置 1。需要了解 RESF 的详细信息，可以参阅 **第二十二章 复位功能**。

10.2 看门狗定时器的配置

看门狗定时器包含以下硬件。

表 10-1. 看门狗定时器的配置

项目	配置
控制寄存器	看门狗定时器允许寄存器 (WDTE)

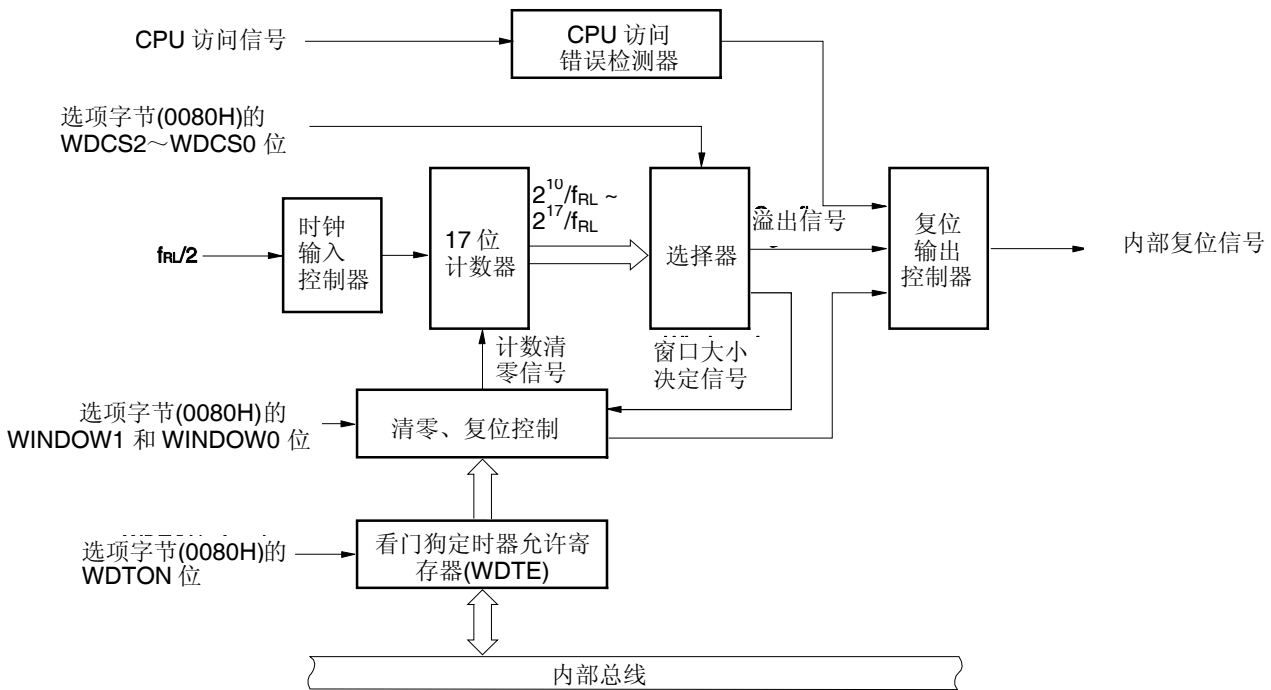
由选项字节设置如何控制计数器的操作、溢出时间和窗口打开周期。

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 (0080H)
窗口打开周期	第6位和第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位 ~ 第1位 (WDCS2 ~ WDCS0)

备注 需要了解选项字节的内容, 可参见 第二十五章 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允许寄存器 (WDTE) 控制。

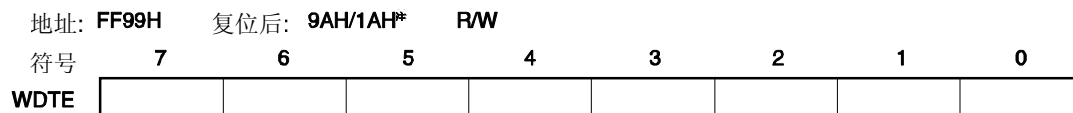
(1) 看门狗定时器允许寄存器 (WDTE)

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. 看门狗定时器允许寄存器 (WDTE) 的格式



注 WDTE 的复位值随选项字节 (0080H) 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 (允许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。如果看门狗定时器的时钟源被停止，则当看门狗定时器的时钟源恢复操作时，将产生内部复位信号。
 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。如果看门狗定时器的源时钟被停止，当提供给看门狗定时器的源时钟恢复操作时，将产生内部复位信号。
 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 看门狗定时器操作的控制

- 当使用看门狗定时器时，由选项字节 (0080H) 指定其操作。
 - 通过设置选项字节 (0080H) 的第 4 位 (WDTON) =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，可参见 第二十五章)。

WDTON	看门狗定时器计数器的操作控制/非法访问检测
0	禁止计数器操作 (复位后停止计数)，禁止非法访问检测操作
1	允许计数器操作 (复位后开始计数)，允许进行非法访问检测操作

- 通过使用选项字节 (0080H) 的第 3 位~第 1 位 (WDACS2 ~ WDACS0) 设置溢出时间 (如需了解详细信息，参见 10.4.2 和 第二十五章)。
 - 通过使用选项字节 (0080H) 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 第二十五章)。
- 复位释放后，看门狗定时器开始计数。
 - 在看门狗定时器开始计数后且在达到选项字节设置的溢出时间之前，将“ACH”写入 WDTE，看门狗定时器将被清零并再次开始计数。
 - 经过上述步骤后对 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。
 - 如果已经历了溢出时间还没有将“ACH”写入 WDTE，则会产生内部复位信号。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允许寄存器 (WDTE) 执行位操作指令时。
 - 如果将“ACH”以外的数据写入 WDTE。
 - 如果从没有通过 IMS 和 IXS 寄存器设置的区域获取指令 (在 CPU 程序循环期间检查是否有无效校验)
 - 如果因为执行 1 条读/写指令，CPU 访问了没有通过 IMS 和 IXS 寄存器设置的区域 (FB00H ~ FFFFH 除外) (CPU 程序循环期间检测是否有非法访问)

- 注意事项**
- 复位释放后对 WDTE 进行第一次写操作，将清零看门狗定时器，如果这一操作是在到达溢出时间之前进行的而不管写操作时序，看门狗定时器再次开始计数。
 - 如果通过将“ACH”写入 WDTE 来对看门狗定时器清零，则实际的溢出时间可能与由选项字节设置的溢出时间不同，最高可达 $2/f_{RL}$ 秒。
 - 在计数值溢出 (FFFFH) 之前，可将看门狗定时器立即清零。

注意事项 4. 在 **HALT** 和 **STOP** 模式下看门狗定时器的操作依据选项字节第 0 位 (**LSROSC**) 的设置值的不同而不同，如下所示。

	LSROSC = 0 (可由软件停止内部低速振荡器)	LSROSC = 1 (不能停止内部低速振荡器)
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 **LSROSC = 0**，则在 **HALT** 或 **STOP** 模式释放后看门狗定时器恢复计数。此时，不对计数器清零，而从计数器停止时的值开始计数。

如果在 **LSROSC = 0** 时通过设置 **LSRSTOP** (内部振荡模式寄存器 (**RCM**) 的第 1 位=1) 停止内部低速振荡器的振荡，则看门狗定时器停止操作。此时不对计数器清零 (**0**)。

5. 在 **flash** 存储器自编程和 **EEPROM™** 仿真期间看门狗定时器不停止操作。在处理期间中断响应被延迟。设置溢出时间和窗口大小时，应考虑延迟。

11.4.2 看门狗定时器溢出时间的设置

通过使用选项字节 (0080H) 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，则将产生内部复位信号。在到达溢出时间之前且在窗口打开周期内通过将“**ACH**”写入 **WDTE**，可对当前计数值清零，且看门狗定时器再次开始计数。

溢出时间设置如下所示。

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
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)

注意事项 1. 禁止出现 **WDCS2 = WDCS1 = WDCS0 = 0** 且 **WINDOW1 = WINDOW0 = 0** 的情况。

2. 在 **flash** 存储器自编程和 **EEPROM** 仿真期间看门狗定时器不停止操作。在处理期间中断响应被延迟。设置溢出时间和窗口大小时，应考虑延迟。

备注 1. f_{RL} : 内部低速振荡时钟频率

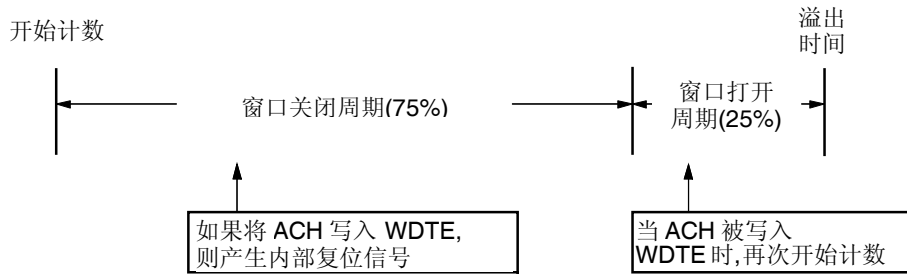
2. (): $f_{RL} = 264 \text{ kHz (MAX.)}$

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 (0080H) 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。

- 如果在窗口打开期间将“ACH”写入 WDTE, 则对看门狗定时器清零并再次开始计数。
- 在窗口关闭期间即使将“ACH”写入 WDTE, 也会检测到异常情况, 并产生内部复位信号。

示例: 如果窗口打开周期为 25%



注意事项 复位释放后对 WDTE 进行第一次写操作, 将清零看门狗定时器, 如果这一操作是在到达溢出时间之前进行的而不管写操作时序, 看门狗定时器再次开始计数。

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. 禁止出现 $WDCS2 = WDCS1 = WDCS0 = 0$ 且 $WINDOW1 = WINDOW0 = 0$ 的情况。
 2. 在 flash 存储器自编程和 EEPROM 仿真期间看门狗定时器不停止操作。在处理阶段中断响应被延迟。设置溢出时间和窗口大小时, 应考虑延迟。

备注 如果将溢出时间设置为 $2^{10}/f_{RL}$ ，则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 ~ 3.56 ms	0 ~ 2.37 ms	0 ~ 0.119 ms	无
窗口打开时间	3.56 ~ 3.88 ms	2.37 ~ 3.88 ms	0.119 ~ 3.88 ms	0 ~ 3.88 ms

<当窗口打开周期为 25%>

- 溢出时间:

$$2^{10}/f_{RL} (\text{MAX.}) = 2^{10}/264 \text{ kHz} (\text{MAX.}) = 3.88 \text{ ms}$$

- 窗口关闭时间:

$$0 \sim 2^{10}/f_{RL} (\text{MIN.}) \times (1 - 0.25) = 0 \sim 2^{10}/216 \text{ kHz} (\text{MIN.}) \times 0.75 = 0 \sim 3.56 \text{ ms}$$

- 窗口打开时间:

$$2^{10}/f_{RL} (\text{MIN.}) \times (1 - 0.25) \sim 2^{10}/f_{RL} (\text{MAX.}) = 2^{10}/216 \text{ kHz} (\text{MIN.}) \times 0.75 \sim 2^{10}/264 \text{ kHz} (\text{MAX.}) \\ = 3.56 \sim 3.88 \text{ ms}$$

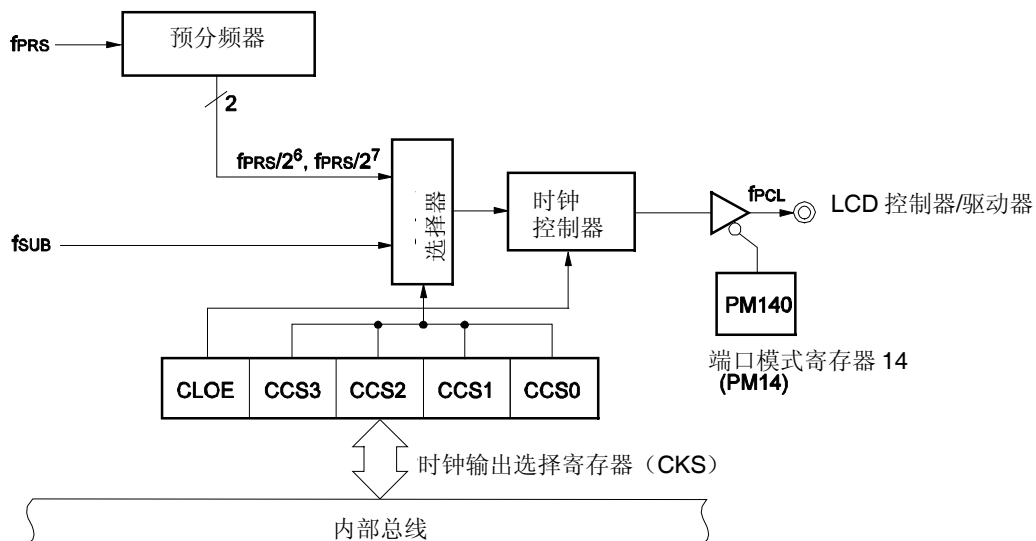
第十一章 时钟输出控制器

11.1 时钟输出控制器的功能

78K0/LF2 的时钟输出控制器可为 LCD 控制器/驱动器提供时钟输出。时钟输出选择寄存器 (CKS) 选择的时钟可支持 LCD 控制器/驱动器。

图 11-1 显示了时钟输出控制器的框图。

图 11-1. 时钟输出控制器的框图



11.2 时钟输出控制器的配置

时钟输出控制器包括以下硬件。

表 11-1. 时钟输出控制器的配置

项目	配置
控制寄存器	时钟输出选择寄存器 (CKS) 端口模式寄存器 14 (PM14)

11.3 控制时钟输出控制器的寄存器

以下两个寄存器用于控制时钟输出控制器。

- 时钟输出选择寄存器 (CKS)
- 端口模式寄存器 14 (PM14)

(1) 时钟输出选择寄存器 (CKS)

该寄存器用于设置允许/禁止时钟输出给 LCD 控制器/驱动器，并设置输出时钟。
 可以由 1 位或 8 位存储器操作指令设置 CKS。
 复位信号的产生将 CKS 清零 (00H)。

<R>

图 11-2. 时钟输出选择寄存器 (CKS) 的格式

地址: FF40H 复位后: 00H R/W

符号	7	6	5	<4>	3	2	1	0
CKS	0	0	0	CLOE	CCS3	CCS2	CCS1	CCS0

CLOE	PM140	指定允许/禁止时钟输出给 LCD 控制器/驱动器 ^注
1	0	允许时钟输出给 LCD 控制器/驱动器
其它		禁止时钟输出给 LCD 控制器/驱动器

CCS3	CCS2	CCS1	CCS0	PCL 输出时钟选择			
				f _{SUB} = 32.768 kHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz	
0	1	1	0	f _{PRS} /2 ⁶	–	156.25 kHz	312.5 kHz
0	1	1	1	f _{PRS} /2 ⁷		78.125 kHz	156.25 kHz
1	0	0	0	f _{SUB}	32.768 kHz	–	
其它				禁止设置			

注 通过结合 PM140 的设置可指定允许/禁止 PCL 时钟输出 (参看 17.3 (7) 端口模式寄存器 14 (PM14))。

- 注意事项 1. 当时钟输出操作停止时 (CLOE = 0)，设置 CCS3 ~ CCS0。
 2. 第 5 位 ~ 第 7 位必须设置为 0。

- 备注 1. f_{PRS}: 外围硬件时钟振荡频率
 2. f_{SUB}: 副系统时钟振荡频率

(2) 端口模式寄存器 14 (PM14)

由 PM14 控制时钟输出给 LCD 控制器/驱动器。
 当使用本寄存器作为时钟输出功能时，设置 PM140 为 0。
 可以由 1 位或 8 位存储器操作指令设置 PM14。
 复位信号产生将 PM14 设置为 FFH。

图 11-3. 端口模式寄存器 14 (PM14) 的格式

地址: FF2EH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM14	1	1	1	1	1	1	1	PM140

PM140	时钟输出给 LCD 控制器/驱动器
0	允许时钟输出给 LCD 控制器/驱动器
1	禁止时钟输出给 LCD 控制器/驱动器

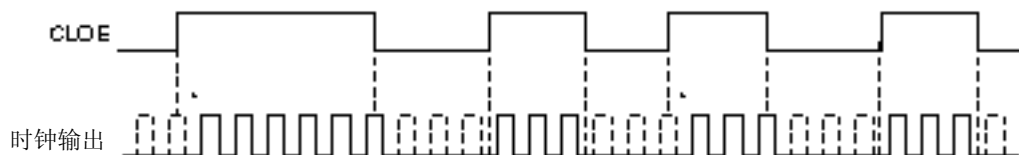
11.4 时钟输出控制器的操作

时钟脉冲输出包含以下过程。

- <1> 通过时钟输出选择寄存器 (CKS) 的第 0 位 ~ 第 3 位 (CCS0 ~ CCS3) 选择时钟脉冲输出频率 (时钟脉冲输出为禁止状态)。
- <2> 将 CKS 的第 4 位 (CLOE) 置 1, 允许时钟输出。

备注 在时钟输出的输出允许/禁止切换阶段, 时钟输出控制器不能输出小宽度的脉冲。如图 11-4 所示, 必须确保在时钟的低电平区 (图中标 * 处) 启动输出。当停止输出时, 也必须在时钟高电平之后停止输出。

图 11-4. 时钟输出应用示例



第十二章 A/D 转换器 (仅限于 μ PD78F037x)

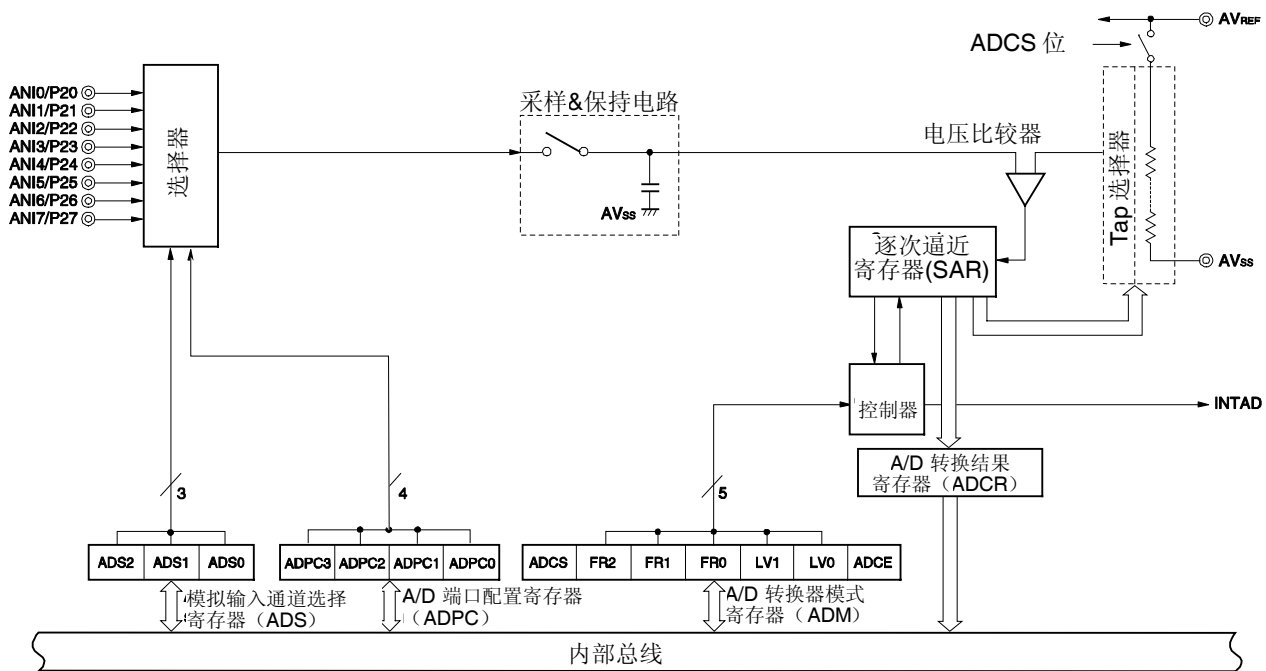
12.1 A/D 转换器的功能

A/D 转换器用于将模拟输入信号转换为数字信号，最多可由 8 个通道 (ANI0 ~ ANI7) 组成，具有 10 位分辨率。
A/D 转换器有以下功能。

- 10 位分辨率 A/D 转换

从模拟输入 ANI0~ANI7 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，都可以产生一个中断请求 (INTAD)。

图 12-1. A/D 转换器的框图



12.2 A/D 转换器的配置

A/D 转换器包括以下硬件。

(1) ANI0 ~ ANI7 引脚

这些是 8 通道 A/D 转换器的模拟输入引脚。用于输入模拟信号，以便转换成数字信号。除了那些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。

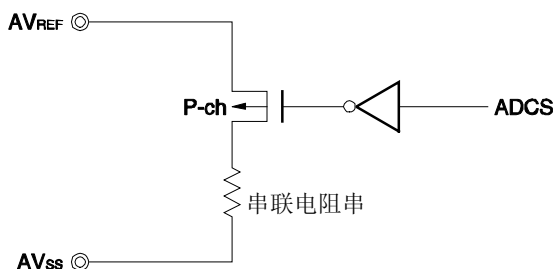
(2) 采样&保持电路

采样&保持电路在 A/D 转换器启动时用于采样由选择器选择的模拟输入引脚的输入电压，并在 A/D 转换期间保持采样到的电压值。

(3) 串联电阻串

串联电阻串用于连接 AV_{REF} 和 AV_{SS} ，并产生一个电压，与采样到的电压值进行比较。

图 12-2. 串联电阻串的电路配置



(4) 电压比较器

电压比较器用于比较采样到的电压值与串联电阻串的输出电压。

(5) 逐次逼近寄存器 (SAR)

该寄存器用于转换电压比较器所比较的结果，并从最高有效位(MSB)开始转换结果。

当电压被转换成数字值，并写入最低有效位 (LSB) 时 (A/D 转换结束)，将 SAR 的内容传送到 A/D 转换结果寄存器 (ADCR) 中。

(6) 10 位 A/D 转换结果寄存器 (ADCR)

每次 A/D 转换结束时，将 A/D 转换结果从逐次逼近寄存器 (SAR) 传送至该寄存器中。并由该寄存器将转换结果保存在它的高 10 位中 (低 6 位恒为 0)。

(7) 8 位 A/D 转换结果寄存器 (ADCRH)

每次 A/D 转换结束时, 将 A/D 转换结果从逐次逼近寄存器传送到该寄存器中。并将转换结果保存在 ADCRH 的高 8 位中。

注意事项 当从 ADCR 和 ADCRH 读取数据时, 会产生一个等待周期。当 CPU 工作在副系统时钟, 且外围硬件时钟处于停止状态时, 不要从 ADCR 和 ADCRH 读取数据。需要了解详细信息, 可参阅第二十九章等待注意事项。

(8) 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。

(9) AV_{REF} 引脚

该引脚为 A/D 转换器输入一个模拟供电/参考电压。当端口 2 被用作数字端口时, 该引脚的电压应与 V_{DD} 的电压保持一致。

根据 AV_{REF} 和 AV_{SS} 的电压, 将输入到 ANI0 ~ ANI7 的信号转换成数字信号。

(10) AV_{SS} 引脚

这是 A/D 转换器的信号地引脚。即便没有使用 A/D 转换器, 该引脚的电压也应与 V_{SS} 的电压保持一致。

(11) A/D 转换器模式寄存器(ADM)

该寄存器用于设置被转换的模拟输入信号的转换时间, 并启动或停止转换操作。

(12) A/D 端口配置寄存器(ADPC)

该寄存器用于将 ANI0/P20 ~ ANI7/P27 引脚切换成 A/D 转换器的模拟输入或端口的数字 I/O。

(13) 模拟输入通道选择寄存器 (ADS)

该寄存器用来选择输入模拟电压 (将被转换成数字信号) 的端口。

(14) 端口模式寄存器 2 (PM2)

该寄存器用于将 ANI0/P20 ~ ANI7/P27 引脚切换为输入或输出。

12.3 A/D转换器使用的寄存器

A/D转换器使用以下六个寄存器。

- A/D转换器模式寄存器 (ADM)
- A/D端口配置寄存器 (ADPC)
- 模拟输入通道选择寄存器 (ADS)
- 端口模式寄存器 2 (PM2)
- 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) 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。

可由 1 位或 8 位存储器操作指令设置 ADM。

复位信号产生将该寄存器清零(00H)。

图 12-3. A/D转换器模式寄存器 (ADM) 的格式

地址: FF28H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	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

ADCE	比较器操作控制 ^{注2}
0	停止比较器的操作
1	允许比较器的操作 (比较器: 1/2AVREF 操作)

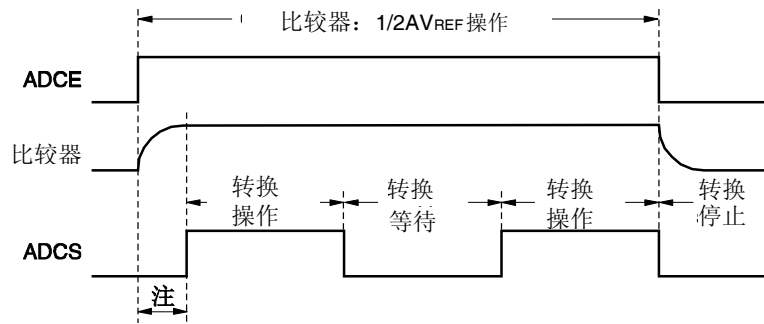
- 注
1. 如需了解 FR2~FR0、LV1、LV0 和 A/D 转换的详细信息，可参见表 12-2 A/D 转换时间的选择。
 2. 比较器的的操作是由 ADCS 和 ADCE 控制的，从操作开始到操作稳定需要 1 μ s。因此，当 ADCE 被设置为 1 后，至少经过 1 μ s 的时间，再将 ADCS 置 1 时，此时的转换结果优先于第一次的转换结果。否则，忽略第一次转换的数据。

表 12-1. ADCS 和 ADCE 的设置

ADCS	ADCE	A/D 转换操作
0	0	停止状态 (不存在直流功耗路径)
0	1	转换等待模式 (比较器: 1/2AVREF 操作, 只有比较器产生功耗)
1	0	转换模式 (停止比较器的操作 ^注)
1	1	转换模式 (比较器: 1/2AVREF 操作)

注 因为第一次转换的结果不确定，因此忽略第 1 次转换的数据。

图 12-4. 使用比较器时的时序图



注 从 ADCE 位上升到 ADCS 位下降所需时间必须至少为 $1\ \mu\text{s}$ ，以便稳定内部电路。

- 注意事项
1. 在将 FR0 ~ FR2 位, LV1 和 LV0 位修改成其他值之前 (非相同值), A/D 转换必须停止。
 2. 如果对 ADM 赋值, 会产生 1 个等待周期。当 CPU 工作在副系统时钟, 且外围硬件时钟处于停止状态时, 不要对 ADM 赋值。需要了解详细信息, 可参阅 第三十一章 等待注意事项。

表 12-2. A/D 转换时间的选择

(1) $2.7\text{ V} \leq \text{AV}_{\text{REF}} \leq 5.5\text{ V}$

A/D 转换器模式寄存器 (ADM)					转换时间的选择			转换时钟 (f_{AD})	转换时间配置				
FR2	FR1	FR0	LV1	LV0	$f_{\text{PRS}} = 2\text{ MHz}$	$f_{\text{PRS}} = 10\text{ MHz}$	$f_{\text{PRS}} = 20\text{ MHz}$ 注		SAR 清 零	采样	连续转换 时间	ADCR 转 换, INTAD 发生器	
0	0	0	0	0	$264/f_{\text{PRS}}$	禁止设置	$26.4\ \mu\text{s}$	$13.2\ \mu\text{s}$ 注	$f_{\text{PRS}}/12$	$2/f_{\text{AD}}$	$6/f_{\text{AD}}$	$12/f_{\text{AD}}$	$2/f_{\text{AD}}$
0	0	1	0	$176/f_{\text{PRS}}$	$17.6\ \mu\text{s}$		$8.8\ \mu\text{s}$ 注	$f_{\text{PRS}}/8$					
0	1	0	0	$132/f_{\text{PRS}}$	$13.2\ \mu\text{s}$		$6.6\ \mu\text{s}$ 注	$f_{\text{PRS}}/6$					
0	1	1	0	$88/f_{\text{PRS}}$	$8.8\ \mu\text{s}$ 注		禁止设置	$f_{\text{PRS}}/4$					
1	0	0	0	$66/f_{\text{PRS}}$	$33.0\ \mu\text{s}$			$6.6\ \mu\text{s}$ 注	$f_{\text{PRS}}/3$				
1	0	1	0	$44/f_{\text{PRS}}$	$22.0\ \mu\text{s}$			禁止设置	$f_{\text{PRS}}/2$				
其它					禁止设置								

注 只有当 $4.0\text{ V} \leq \text{AV}_{\text{REF}} \leq 5.5\text{ V}$ 时, 才能设置。

(2) $2.3\text{ V} \leq \text{AV}_{\text{REF}} < 2.7\text{ V}$

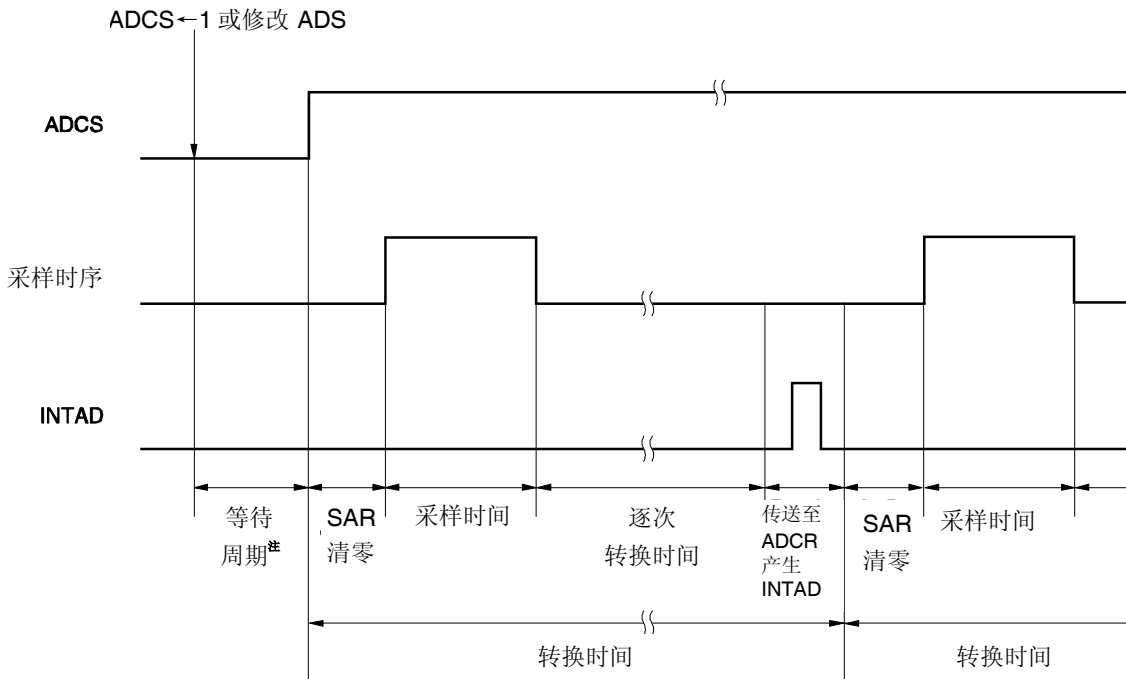
A/D 转换器模式寄存器 (ADM)					转换时间的选择			转换时钟 (f_{AD})	转换时间配置			
FR2	FR1	FR0	LV1	LV0	$f_{\text{PRS}} = 2\text{ MHz}$	$f_{\text{PRS}} = 5\text{ MHz}$	SAR 清 零		采样	连续转换 时间	ADCR 转 换, INTAD 发生器	
0	0	0	0	1	$480/f_{\text{PRS}}$	禁止设置	禁止设置	$f_{\text{PRS}}/12$	$2/f_{\text{AD}}$	$24/f_{\text{AD}}$	$12/f_{\text{AD}}$	$2/f_{\text{AD}}$
0	0	1	0	$320/f_{\text{PRS}}$	$64.0\ \mu\text{s}$		$f_{\text{PRS}}/8$					
0	1	0	0	$240/f_{\text{PRS}}$	$48.0\ \mu\text{s}$		$f_{\text{PRS}}/6$					
0	1	1	0	$160/f_{\text{PRS}}$	$32.0\ \mu\text{s}$		$f_{\text{PRS}}/4$					
1	0	0	0	1	$120/f_{\text{PRS}}$	$60.0\ \mu\text{s}$	禁止设置	$f_{\text{PRS}}/3$				
1	0	1	0	1	$80/f_{\text{PRS}}$	$40.0\ \mu\text{s}$	禁止设置	$f_{\text{PRS}}/2$				
其它					禁止设置							

注意事项 1. 根据以下条件设置转换时间

- $4.0\text{ V} \leq \text{AV}_{\text{REF}} \leq 5.5\text{ V}$: 采样 + 连续转换时间 = $5 \sim 30\ \mu\text{s}$
($f_{\text{AD}} = 0.6 \sim 3.6\text{ MHz}$)
 - $2.7\text{ V} \leq \text{AV}_{\text{REF}} < 4.0\text{ V}$: 采样 + 连续转换时间 = $10 \sim 30\ \mu\text{s}$
($f_{\text{AD}} = 0.6 \sim 1.8\text{ MHz}$)
 - $2.3\text{ V} \leq \text{AV}_{\text{REF}} < 2.7\text{ V}$: 采样 + 连续转换时间 = $25 \sim 62\ \mu\text{s}$
($f_{\text{AD}} = 0.6 \sim 1.48\text{ MHz}$)
2. 当修改 FR2 ~ FR0, LV1 和 LV0 为其他值时 (非相同值), 要预先停止 A/D 转换一次 (ADCS = 0)。
 3. 当 $2.3\text{ V} \leq \text{AV}_{\text{REF}} < 2.7\text{ V}$ 时, 修改 LV1 和 LV0 的默认值。
 4. 上述的转换时间不包括时钟频率误差。在选择转换时间的时候, 要将时钟频率误差考虑进去。

备注 f_{PRS} : 外围硬件时钟频率

图 12-5. A/D 转换器采样和 A/D 转换时序



注 等待周期的详细信息参见 第三十一章 等待注意事项。

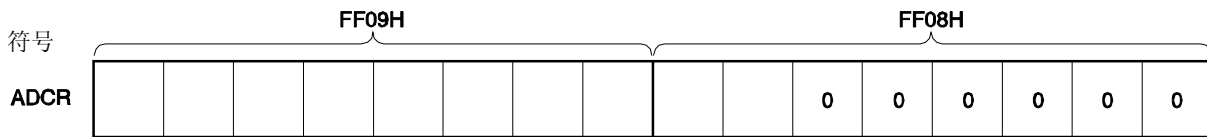
(2) 10 位 A/D 转换结果寄存器 (ADCR)

这是一个 16 位寄存器，用来存储 A/D 转换结果。寄存器的低 6 位恒为 0。每次 A/D 转换结束时，将转换结果从逐次逼近寄存器传送至 ADCR 中。转换结果的高 8 位存储在 FF09H 中，而转换结果的低 2 位存储在 FF08H 的高 2 位中。

可由 16 位存储器操作指令读取 ADCR。
复位信号产生将该寄存器清零 (0000H)。

图 12-6. 10 位 A/D 转换结果寄存器 (ADCR) 的格式

地址:FF08H, FF09H 复位后: 0000H R



- 注意事项
- 在对 A/D 转换器模式寄存器 (ADM) 和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。转换结束后先读取转换结果，再对 ADM, ADS 和 ADPC 赋值。如果没有按照上述时序，可能会读取到不正确的转换结果。
 - 如果从 ADCR 读取数据，会产生一个等待周期。当 CPU 工作在副系统时钟，且外围硬件时钟处于停止状态时，不要从 ADCR 读取数据。需要了解详细信息，可参阅 第三十一章 等待注意事项。

(3) 8位 A/D 转换结果寄存器 (ADCRH)

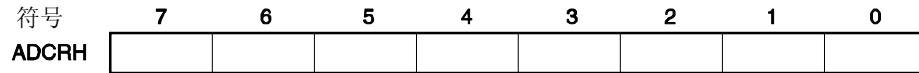
这是一个 8 位寄存器，用来存储 A/D 转换结果。可以存储 10 位分辨率的高 8 位。

可由 8 位存储器操作指令读取 ADCRH。

复位信号产生将该寄存器清零(00H)。

图 12-7. 8 位 A/D 转换结果寄存器 (ADCRH) 的格式

地址: FF09H 复位后: 00H R



- 注意事项**
1. 在对 A/D 转换器模式寄存器 (ADM)、模拟输入通道选择寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 赋值时，ADCRH 的内容可能不确定。转换结束后先读取转换结果，再对 ADM、ADS 和 ADPC 赋值。如果没有按照上述时序，可能会读取到不正确的转换结果。
 2. 如果从 ADCRH 读取数据，会产生一个等待周期。当 CPU 工作在副系统时钟，且外围硬件时钟处于停止状态时，不要从 ADCRH 读取数据。需要了解详细信息，可参阅 第三十一章 等待注意事项。

(4) 模拟输入通道选择寄存器 (ADS)

该寄存器用来选择被 A/D 转换的模拟电压的输入通道。

可由 1 位或 8 位存储器操作指令设置 ADS。

复位信号产生将该寄存器清零(00H)。

图 12-8. 模拟输入通道选择寄存器 (ADS) 的格式

地址: FF29H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
ADS	0	0	0	0	0	ADS2	ADS1	ADS0

ADS2	ADS1	ADS0	模拟输入通道的选择
0	0	0	ANI0
0	0	1	ANI1
0	1	0	ANI2
0	1	1	ANI3
1	0	0	ANI4
1	0	1	ANI5
1	1	0	ANI6
1	1	1	ANI7

- 注意事项
1. 第 3 位 ~ 第 7 位必须清零。
 2. 使用端口模式寄存器 2 (PM2) 设置一个用于 A/D 转换通道的端口为输入模式。
 3. 不要同时使用 ADPC 和 ADS 设置一个引脚作为数字 I/O 引脚。
 4. 如果对 ADS 赋值, 会产生一个等待周期。当 CPU 工作在副系统时钟, 且外围硬件时钟处于停止状态时, 不要对 ADS 赋值。需要了解详细信息, 可参阅 第三十一章 等待注意事项。

(5) A/D 端口配置寄存器 (ADPC)

这个寄存器用于将 ANI0/P20 ~ ANI7/P27 引脚切换为 A/D 转换器的模拟输入或者数字 I/O 端口。
 可由 1 位或 8 位存储器操作指令设置 ADPC。
 复位信号产生将该寄存器清零(00H)。

图 12-9. A/D 端口配置寄存器 (ADPC) 的格式

地址: FF2FH 复位后: 00H R/W

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

ADPC3	ADPC2	ADPC1	ADPC0	模拟输入(A)/数字 I/O(D)的切换							
				ANI7/ P27	ANI6/ P26	ANI5/ P25	ANI4/ P24	ANI3/ P23	ANI2/ P22	ANI1/ P21	ANI0/ P20
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	A	A	A	A	A	A	A	D
0	0	1	0	A	A	A	A	A	A	D	D
0	0	1	1	A	A	A	A	A	D	D	D
0	1	0	0	A	A	A	A	D	D	D	D
0	1	0	1	A	A	A	D	D	D	D	D
0	1	1	0	A	A	D	D	D	D	D	D
0	1	1	1	A	D	D	D	D	D	D	D
1	0	0	0	D	D	D	D	D	D	D	D
其它				禁止设置							

- 注意事项
1. 用端口模式寄存器 2 (PM2) 设置一个用于 A/D 转换通道的端口为输入模式。
 2. 不要同时使用 ADPC 和 ADS 设置一个引脚作为数字 I/O 引脚。
 3. 如果将数据写入 ADPC, 会产生一个等待周期。当 CPU 工作在副系统时钟, 且外围硬件时钟处于停止状态时, 不要将数据写入 ADPC。需要了解详细信息, 可参阅 第三十一章 等待注意事项。

(6) 端口模式寄存器 2 (PM2)

在使用 ANI0/P20 ~ ANI7/P27 引脚作为模拟输入端口时, 将 PM20 ~ PM27 设为 1。此时 P20 ~ P27 的输出锁存器的内容为 0 或 1。

如果将 PM20 ~ PM27 设为 0, 则它们不能用作模拟输入端口引脚。

可由 1 位或 8 位的存储器操作指令来设置 PM2。

复位信号产生将该寄存器设置为 FFH。

图 12-10. 端口模式寄存器 2 (PM2) 的格式

地址: FF22H 复位后: FFH R/W

	7	6	5	4	3	2	1	0
符号	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

PM2n	P2n 引脚 I/O 模式的选择(n = 0~7)
0	输出模式(输出缓冲器打开)
1	输入模式(输出缓冲器关闭)

根据 ADPC、ADS 和 PM2 的设置, ANI0/P20 ~ ANI7/P27 引脚功能如下所示。

图 12-3. ANI0/P20 ~ ANI7/P27 引脚功能的设置

ADPC	PM2	ADS	ANI0/P20 ~ ANI7/P27 引脚
模拟输入的选择	输入模式	选择 ANI	模拟输入 (被转换)
		不选择 ANI	模拟输入 (不被转换)
	输出模式	选择 ANI	禁止设置
		不选择 ANI	
数字 I/O 的选择	输入模式	选择 ANI	禁止设置
		不选择 ANI	数字输入
	输出模式	选择 ANI	禁止设置
		不选择 ANI	数字输出

12.4 A/D转换器的操作

12.4.1 A/D转换器的基本操作

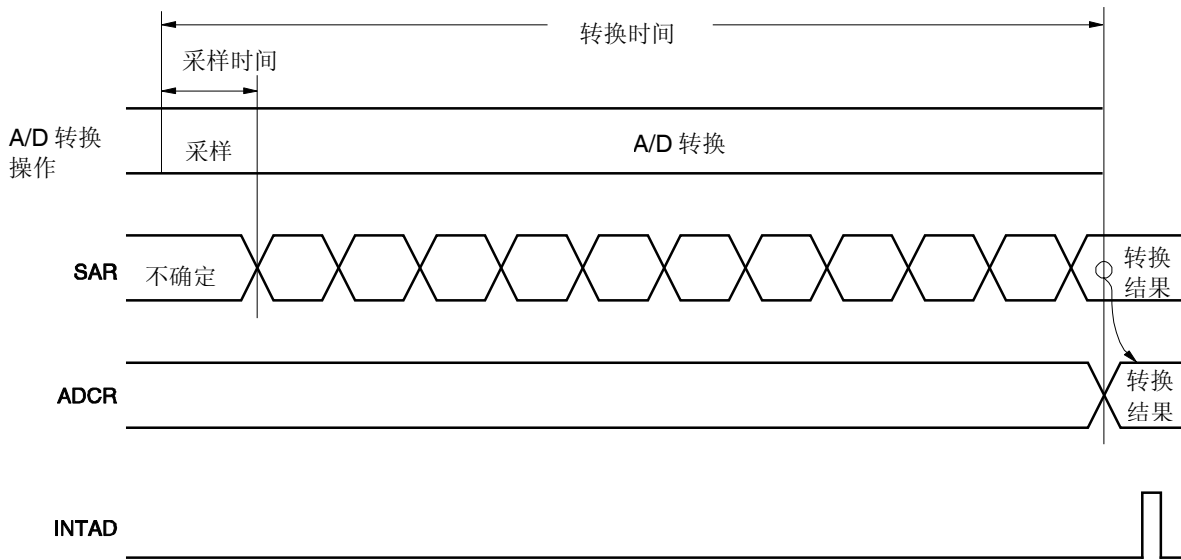
- <1> 把A/D转换器模式寄存器的第0位(ADCE)置1启动比较器的操作。
- <2> 通过A/D端口配置寄存器(ADPC)设置通道,用于A/D转换的模拟输入,并且通过端口模式寄存器2(PM2)设置为输入模式。
- <3> 根据ADM的第5位~第1位(FR2~FR0、LV1和LV0)设置A/D转换时间。
- <4> 通过模拟输入通道选择寄存器(ADS)选择一个通道用于A/D转换
- <5> 设置ADM的第7位(ADCS)为1,启动转换操作。
(<6>~<12>的操作由硬件完成)
- <6> 由采样&保持电路对输入到已选中的模拟输入通道的电压进行采样。
- <7> 在经过一定时间的采样后,采样&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。
- <8> 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串的分接电压置为 $(1/2)AV_{REF}$ 。
- <9> 由电压比较器比较串联电阻串的分接电压与采样电压。如果模拟输入电压高于 $(1/2)AV_{REF}$,则SAR的MSB=1;如果模拟输入电压低于 $(1/2)AV_{REF}$,则SAR的MSB=0。
- <10> 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。
 - 第9位=1: $(3/4)AV_{REF}$
 - 第9位=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。
 - 模拟输入电压 \geq 分接电压: 第8位=1
 - 模拟输入电压 $<$ 分接电压: 第8位=0
- <11> 按此方式继续进行比较,直至SAR的第0位。
- <12> 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)。
- <13> 反复执行步骤<6>~<12>,直至ADCS被清零(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<5>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<5>开始操作。如要改变A/D转换的通道,则从步骤<4>开始。

注意事项 必须确保<1>~<5>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 12-11. A/D 转换器的基本操作



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)，A/D 转换操作将会连续执行。
 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，并且若 ADCS 被设置为 1，则转换操作重新开始。
 复位信号产生将 A/D 转换结果寄存器 (ADCR, ADCRH) 的内容设置为 0000H 或 00H。

12.4.2 输入电压和转换结果

输入到模拟输入引脚 (ANIO ~ ANI7) 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。

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

$$ADCR = SAR \times 64$$

或

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

其中，INT(): 该函数返回括号中值的整数部分。

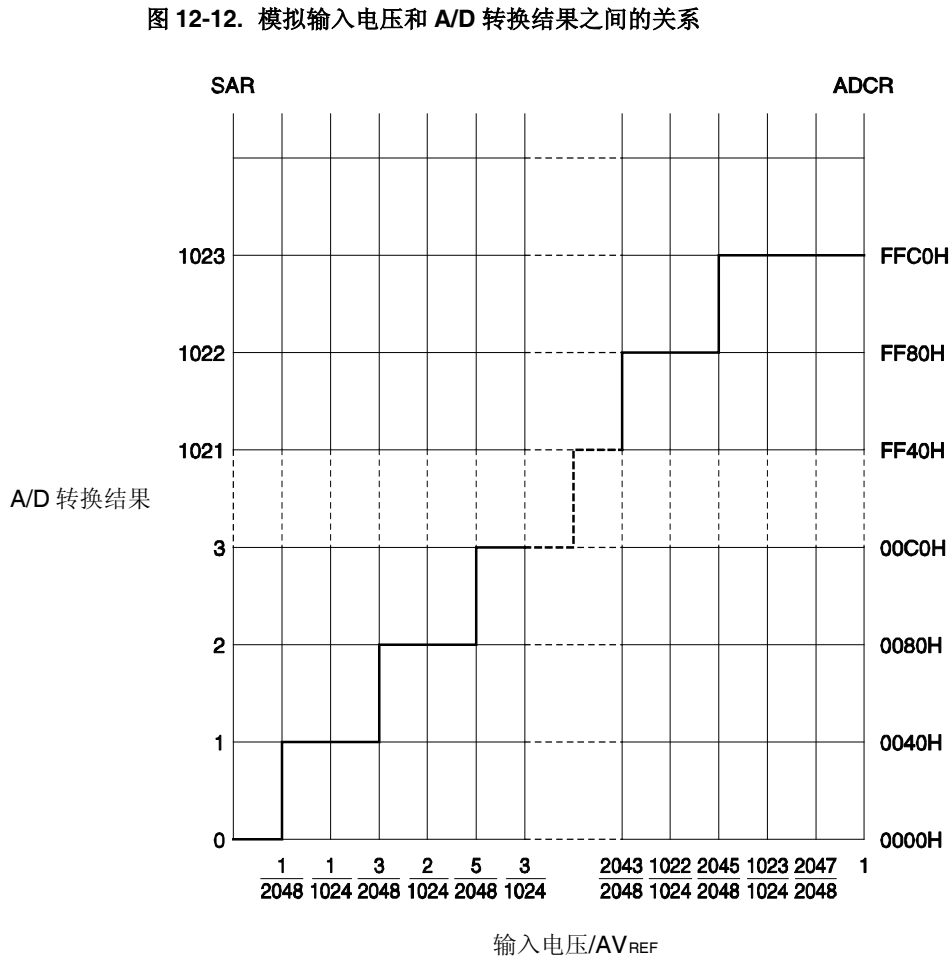
V_{AIN} : 模拟输入电压

AV_{REF} : AV_{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 12-12 显示模拟输入电压与 A/D 转换结果之间的关系。



12.4.3 A/D 转换器操作模式

A/D 转换器的操作模式为选择模式。由模拟输入通道选择寄存器 (ADS) 从 ANI0 ~ ANI7 中选择一个模拟输入通道，并执行 A/D 转换。

(1) A/D 转换操作

通过将 A/D 转换器模式寄存器 (ADM) 的第 7 位 (ADCS) 置 1，可启动电压的 A/D 转换操作，该电压由模拟输入通道选择寄存器 (ADS) 选择的模拟输入引脚输入。

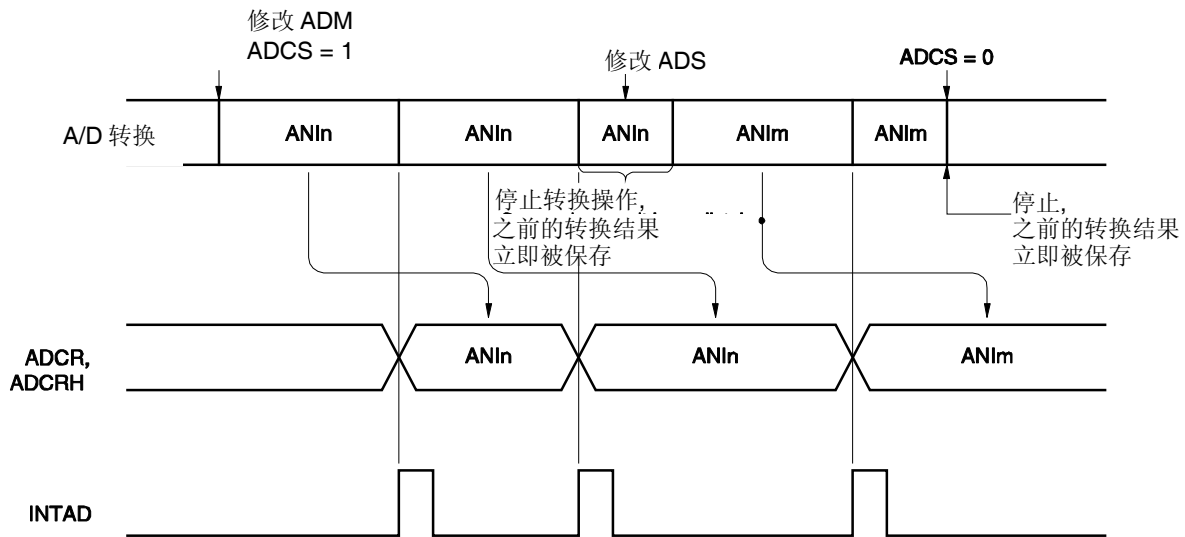
当 A/D 转换完成时，将 A/D 转换结果存于 A/D 转换结果寄存器 (ADCR) 中，并产生一个中断请求信号 (INTAD)。

当一个 A/D 转换结束时，下一个 A/D 转换操作将立即开始。

如果在 A/D 转换期间 ADS 的值被重写，则停止当前正在执行的 A/D 转换操作，重新开始。

如果在 A/D 转换期间 ADCS 被置 0，则 A/D 转换立即停止。此时，之前的转换结果立即被保存。

图 12-13. A/D 转换操作



- 备注
1. $n = 0 \sim 7$
 2. $m = 0 \sim 7$

设置方法描述如下。

- <1> 设置 A/D 转换器模式寄存器 (ADM) 的第 0 位 (ADCE) 为 1。
- <2> 根据 A/D 端口配置寄存器的第 3 位 ~ 第 0 位 (ADPC3 ~ ADPC0) 和端口模式寄存器 2 (PM2) 的第 7 位 ~ 第 0 位 (PM27 ~ PM20)，设置在模拟输入模式下使用的通道。
- <3> 根据 ADM 的第 5 位 ~ 第 1 位 (FR2 ~ FR0、LV1 和 LV0) 选择转换时间。
- <4> 使用模拟输入通道选择寄存器 (ADS) 的第 2 位 ~ 第 0 位 (ADS2 ~ ADS0)，选择通道。
- <5> 将 ADM 的第 7 位 (ADCS) 置 1，启动 A/D 转换操作。
- <6> 当 A/D 转换结束，产生一个中断请求信号 (INTAD)。
- <7> 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>

- <8> 根据 ADS 的第 2 位 ~ 第 0 位 (ADS2 ~ ADS0) 改变通道，启动 A/D 转换操作。
- <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。

<完成 A/D 转换>

- <11> 将 ADCS 清零。
- <12> 将 ADCE 清零。

- 注意事项**
1. 必须确保步骤 <1> ~ <5> 的操作时间至少为 1 μ s。
 2. <1> 可以在 <2> 和 <4> 之间进行。
 3. <1> 可以被省略。但在这种情况中 (<5> 之后) 忽略第一个转换结果。
 4. 步骤 <6> ~ <9> 所经历的时间与使用 ADM 的第 5 位 ~ 第 1 位 (FR2 ~ FR0, LV1, LV0) 设置的转换时间不同。步骤 <8> ~ <9> 所经历的时间为 FR2 ~ FR0, LV1, LV0 设置的转换时间。

12.5 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 12-14. 总误差

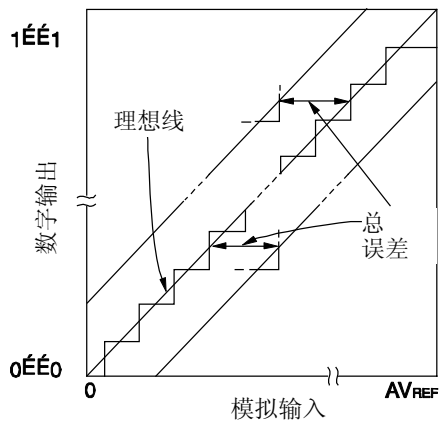
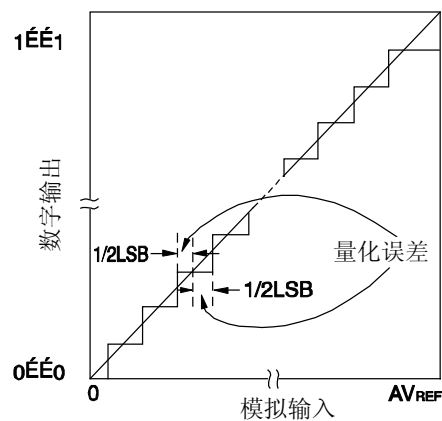


图 12-15. 量化误差



(4) 零度误差

零度误差表示当数字输出范围在 $0\text{.....}000 \sim 0\text{.....}001$ 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 $0\text{.....}001 \sim 0\text{.....}010$ 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 $1\dots\dots110 \sim 1\dots\dots111$ 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 12-16. 零度误差

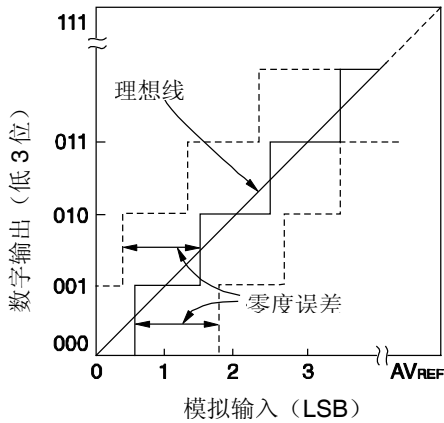


图 12-17. 满度误差

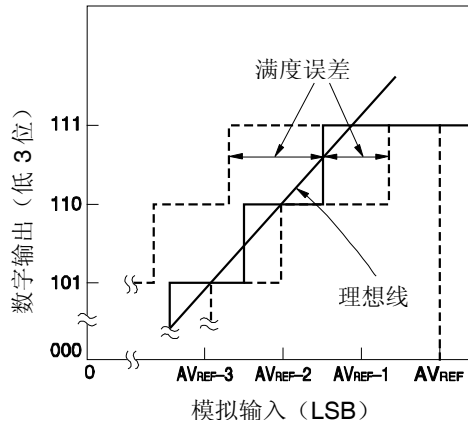


图 12-18. 积分线性误差

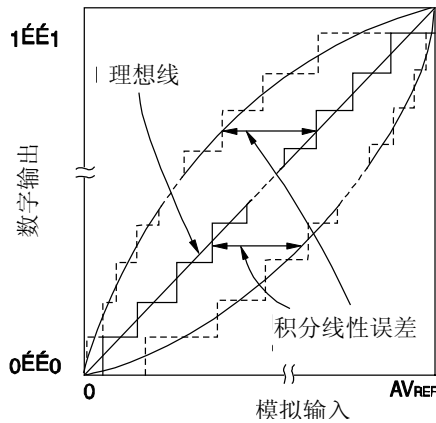
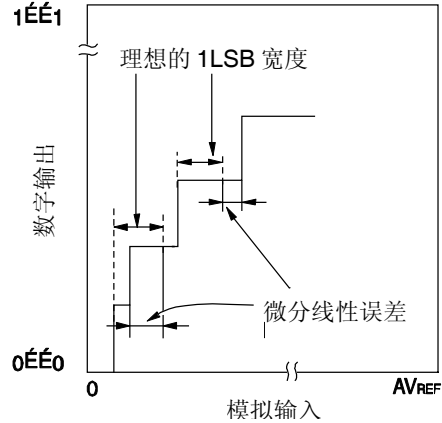


图 12-19. 微分线性误差

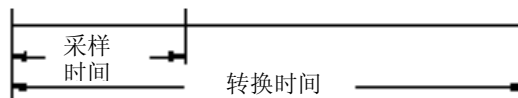


(8) 转换时间

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。



12.6 A/D 转换器使用注意事项

(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 ~ ANI7 的输入范围

观察 ANI0 ~ ANI7 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> 转换结束后，通过指令对 A/D 转换结果寄存器 (ADCR, ADCRH) 的写操作和 ADCR 或 ADCRH 的写操作之间的冲突。

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、A/D 转换器模式寄存器 (ADM) 的写操作以及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 ~ ANI7 引脚的噪音。

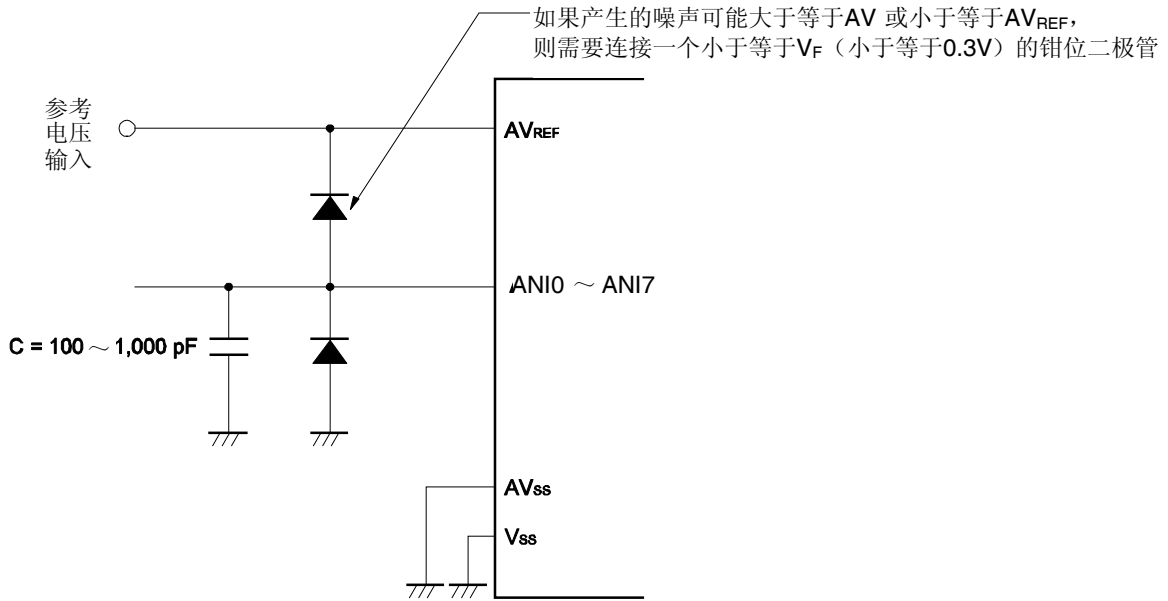
<1> 连接一个低等效电阻和优质频率响应的电容到电源上

<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 12-20 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 12-20. 模拟输入引脚连接



(5) ANI0/P20 ~ ANI7/P27

- <1> 模拟输入引脚 (ANI0 ~ ANI7) 也可用作输入端口引脚 (P20 ~ P27)。当选择 ANI0 ~ ANI7 中的任意一个通道执行 A/D 转换时, 转换过程中不要访问 P20 ~ P27; 否则转换分辨率可能会降低。建议从 ANI0/P20 (离 AV_{REF} 最远) 开始, 选择引脚用作 P20 ~ P27。
- <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 ~ ANI7 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, 而在采样期间则有为电容充电的电流, 因此根据是否进行采样, 输入阻抗会波动, 且无法解决。为了使采样有效, 建议模拟输入源的输出阻抗小于等于 $10\text{ k}\Omega$, 并将一个 100 pF 左右的电容连接到 ANI0 ~ ANI7 引脚上(参见图 12-20)。

(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, 当它串联连接到 AV_{REF} 与 AV_{SS} 引脚之间的串联电阻串时, 会导致较大的参考电压误差。

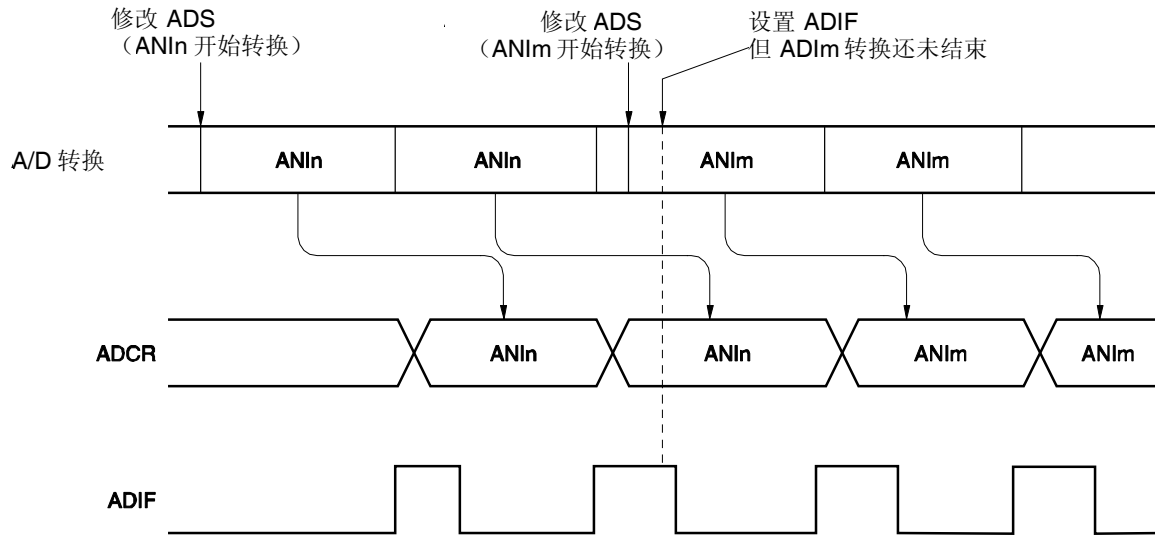
(8) 中断请求标志(ADIF)

即使模拟输入通道选择寄存器 (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 转换停止后又重新开始时, 在启动前先对 ADIF 清零。

图 12-21. A/D 转换结束中断请求产生的时序



- 备注
1. $n = 0 \sim 7$
 2. $m = 0 \sim 7$

(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, 如悬挂 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)的读取操作

当对 A/D 转换器模式寄存器 (ADM)、模拟输入通道选择寄存器 (ADS) 和 A/D 端口配置寄存器 (ADPC) 进行写操作时, ADCR 和 ADCRH 的内容可能不确定。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 12-22. ANIn 引脚的内部等效电路

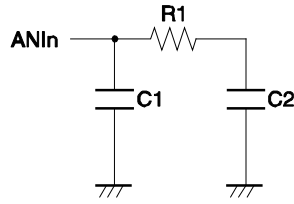


表 12-4. 等效电路的电阻和电容 (参考值)

AV_{REF}	R1	C1	C2
$4.0\text{ V} \leq AV_{REF} \leq 5.5\text{ V}$	8.1 k Ω	8 pF	5 pF
$2.7\text{ V} \leq AV_{REF} < 4.0\text{ V}$	31 k Ω	8 pF	5 pF
$2.3\text{ V} \leq AV_{REF} < 2.7\text{ V}$	381 k Ω	8 pF	5 pF

- 备注
1. 表 12-4 所显示的电阻和电容值仅为参考值。
 2. $n = 0 \sim 7$

13.1 串行接口 UART0 的功能

串行接口 UART0 有以下两种模式。

(1) 操作停止模式

这个模式在不进行串口通信时使用，以降低功耗。

需要了解详细信息，请参阅 **13.4.1 操作停止模式**。

(2) 异步串行接口 (UART) 模式

该模式的功能概括如下。

需要了解详细信息，请参阅 **13.4.2 异步串行接口 (UART) 模式** 和 **13.4.3 专用波特率发生器**。

- 最大传输率： 312.5 kbps
- 双引脚配置 TXD0: 发送数据输出引脚
RXD0: 接收数据输入引脚
- 可选择通信数据的宽度为 7 或 8 位。
- 专用内置 5 位波特率发生器，允许设置任意大小的波特率。
- 可独立的执行发送和接收操作 (全双工操作)。
- 恒为 LSB-first 通信

注意事项 1. 如果串行接口 UART0 使用的时钟未停止(例如处于 HALT 模式下)，则可继续正常操作。如果时钟已停止(例如处于 STOP 模式下)，则每个寄存器都停止操作，并且在时钟停止之前及时保存数据。TxD0 引脚在时钟停止之前立即保存数据并输出。但在时钟恢复使用后的操作不能得到保证。因此必须复位电路，使 POWER0 = 0、RXE0 = 0 以及 TXE0 = 0。

2. 先设置 POWER0 = 1，然后设置 TXE0 = 1(发送)或 RXE0 = 1(接收)开始通信。

3. 由 BRGC0 设置基本时钟 (f_{XCLK0})，可使 TXE0 与 RXE0 同步。为了能够再次发送或接收，在 TXE0 或 RXE0 被清零后，至少要经历两个基本时钟再将 TXE0 或 RXE0 置 1。如果在不到两个基本时钟时设置 TXE0 或 RXE0，则发送或接收电路可能不被初始化。

4. 设置 TXE0=1 之后经过至少一个基本时钟(f_{XCLK0})才能发送数据到 TXS0。

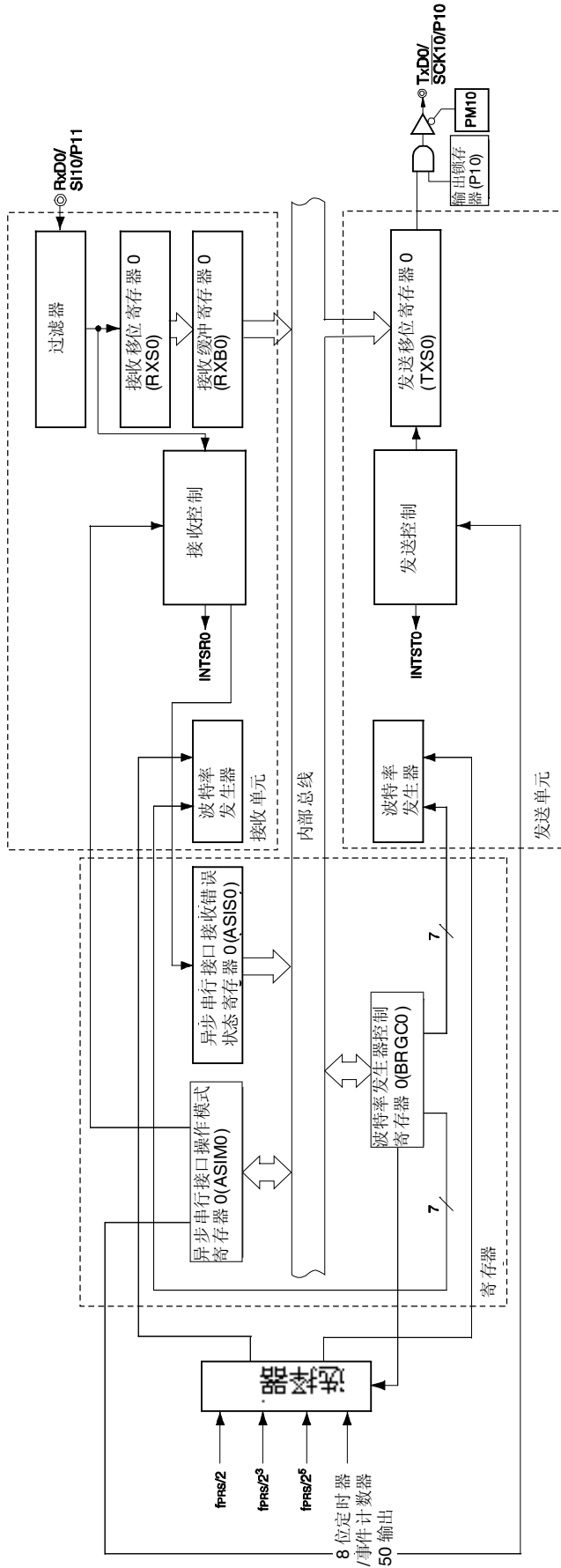
13.2 串行接口 UART0 的配置

串行接口 UART0 包括以下硬件。

表 13-1. 串行接口 UART0 的配置

项目	配置
寄存器	接收缓冲寄存器 0 (RXB0) 接收移位寄存器 0 (RXS0) 发送移位寄存器 0 (TXS0)
控制寄存器	异步串行接口操作模式寄存器 0 (ASIM0) 异步串行接口接收错误状态寄存器 0 (ASIS0) 波特率发生器控制寄存器 0 (BRGC0) 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1)

图 13-1. 串行接口 UART0 的框图



(1) 接收缓冲寄存器 0 (RXB0)

该 8 位寄存器用于存储由接收移位寄存器 0 (RXS0) 转换的并行数据。

每接收到 1 字节的数据，新的接收数据就从接收移位寄存器 0 (RXS0) 被传送到该寄存器中。

如果数据宽度被设定为 7 位，则将接收到的数据传送到 RXB0 的第 0 位 ~ 第 6 位，而 MSB 位恒为 0。

如果出现溢出错误(OVE0)，则不把接收数据传送到 RXB0 中。

可由 8 位存储器操作指令读取 RXB0 的内容。不能将数据写入该寄存器中。

复位信号的产生和 POWER0 = 0 将该寄存器设置为 FFH。

(2) 接收移位寄存器 0 (RXS0)

该寄存器将输入到 RxD0 引脚的串行数据转换成并行数据。

程序不能直接使用 RXS0。

(3) 发送移位寄存器 0 (TXS0)

该寄存器用于设置发送数据。当数据写入 TXS0 时启动发送操作，串行数据从 TxD0 引脚输出。

可由 8 位存储器操作指令对 TXS0 进行写操作。该寄存器的内容不能被读取。

复位信号的产生、POWER0 = 0 和 TXE0 = 0 将该寄存器设置为 FFH。

- 注意事项**
1. 设置 TXE0=1 之后经过至少一个基本时钟 (f_{xCLK0}) 才能发送数据到 TXS0。
 2. 在发送完成中断信号 (INTST0) 产生之前，不要把下一个发送数据写入 TXS0。

13.3 控制串行接口 UART0 的寄存器

串行接口 UART0 由以下 5 个寄存器控制。

- 异步串行接口操作模式寄存器 0 (ASIM0)
- 异步串行接口接收错误状态寄存器 0 (ASIS0)
- 波特率发生器控制寄存器 0 (BRGC0)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

(1) 异步串行接口操作模式寄存器 0 (ASIM0)

该 8 位寄存器用于控制串行接口 UART0 的串行通信操作。

该寄存器可由 1 位或 8 位存储器操作指令设置。

复位信号的产生将该寄存器设置为 01H。

图 13-2. 异步串行接口操作模式寄存器 0 (ASIM0) 的格式 (1/2)

地址: FF70H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1

POWER0	内部操作时钟的允许/禁止操作
0 ^{注1}	禁止内部操作时钟的操作 (时钟恒为低电平) 并异步复位内部电路 ^{注2}
1	允许内部操作时钟的操作

TXE0	允许/禁止发送
0	禁止发送 (同步复位发送电路)
1	允许发送

RXE0	允许/禁止接收
0	禁止接收 (同步复位接收电路)
1	允许接收

- 注
1. 当 POWER0 = 0 时, 从 RxD0 引脚输入的信号恒为高电平。
 2. 对异步串行接口接收错误状态寄存器 0 (ASIS0)、发送移位寄存器 0 (TXS0)和接收缓冲寄存器 0 (RXB0) 进行复位。

图 13-2. 异步串行接口操作模式寄存器 0 (ASIM0) 的格式 (2/2)

PS01	PS00	发送操作	接收操作
0	0	不输出校验位	无校验接收
0	1	输出零校验	零校验接收 ^注
1	0	输出奇校验	按奇校验判断
1	1	输出偶校验	按偶校验判断

CL0	指定发送/接收数据的字符宽度
0	数据的字符宽度 = 7 位
1	数据的字符宽度 = 8 位

SL0	指定发送数据停止位的个数
0	停止位个数 = 1
1	停止位个数 = 2

注 如果选择“按零校验接收”，则无法判断校验。因此不要设置异步串行接口接收错误状态寄存器 0 (ASIS0) 的第 2 位 (PE0)，这样就不会产生错误中断。

- 注意事项**
1. 启动发送时，对 POWER0 置 1，然后对 TXE0 置 1。停止发送时，先将 TXE0 清零，然后将 POWER0 清零。
 2. 启动接收时，对 POWER0 置 1，然后对 RXE0 置 1。停止接收时，先将 RXE0 清零，然后将 POWER0 清零。
 3. 当 RxD0 引脚输入为高电平时，先对 POWER0 置 1，再对 RXE0 置 1。当输入为低电平，POWER0 置 1，RXE0 置 1 时，启动接收操作。
 4. 通过 BRGC0 设置的基本时钟 (f_{XCLK0}) 同步 TXE0 与 RXE0。为了能够再次发送或接收，在 TXE0 或 RXE0 被清零后，至少要经历两个基本时钟再将 TXE0 或 RXE0 置 1。如果在不到两个基本时钟的时间内设置 TXE0 或 RXE0，则发送或接收电路可能不被初始化。
 5. 设置 TXE0=1 之后经过至少一个基本时钟 (f_{XCLK0}) 才能设置发送数据到 TXS0。
 6. 在对 PS01、PS00 和 CL0 位重写之前，先将 TXE0 和 RXE0 位清零。
 7. 在重写 SL0 位时，必须确保 TXE0 = 0。总是以“停止位的个数=1”的格式执行接收操作，因此不会受到 SL0 设置值的影响。
 8. 第 0 位必须设置为 1。

(2) 异步串行接口接收错误状态寄存器 0 (ASIS0)

该寄存器用于指示串行接口 UART0 在接收操作完成时出现的错误状态，包括 3 个错误标志位 (PE0, FE0, OVE0)。可由 8 位存储器操作指令对该寄存器进行只读操作。

如果 ASIM0 的第 5 位 (RXE0) 或第 7 位 (POWER0) = 0，则复位信号将使 ASIS0 寄存器清零为 00H。读取寄存器的值为 00H。如果产生一个接收错误，则先读取 ASIS0 再读取接收缓冲寄存器 0 (RXB0) 以便将出错标志位清零。

图 13-3. 异步串行接口接收错误状态寄存器 0 (ASIS0) 的格式

地址: FF73H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIS0	0	0	0	0	0	PE0	FE0	OVE0

PE0	指示校验错误的状态标志位
0	如果 POWER0 = 0 且 RXE0 = 0、或者如果读取 ASIS0
1	如果发送数据的校验位与接收完成时的校验位不相等

FE0	指示帧错误的状态标志位
0	如果 POWER0 = 0 或 RXE0 = 0、或者如果读取 ASIS0
1	如果接收完成时未检测到停止位

OVE0	指示溢出错误的状态标志位
0	如果 POWER0 = 0 且 RXE0 = 0、或者如果读取 ASIS0
1	如果接收数据被设置在 RXB0 寄存器中且在读取该数据前下一个接收操作已完成

- 注意事项**
1. 根据异步串行接口操作模式寄存器 0 (ASIM0) 的 PS01 和 PS00 位的内容的不同，对 PE0 位的操作也有所不同。
 2. 无论停止位的个数是多少，只能将接收数据的第 1 位作为停止位进行检测。
 3. 如果出现溢出错误，则下一个接收数据不写入接收缓冲寄存器 0 (RXB0) 中，而是被忽略。
 4. 如果从 ASIS0 读取数据，会产生一个等待周期。当 CPU 采用副系统时钟操作而停止外围硬件时钟时，不要从 ASIS0 读取数据。需要了解详细信息，可参见 第三十一章 等待注意事项。

(3) 波特率发生器控制寄存器 0 (BRGC0)

该寄存器用于选择串行接口 UART0 的基本时钟和 5 位计数器的分频值。

可由 8 位存储器操作指令设置 BRGC0。

复位信号的产生将该寄存器设置为 1FH。

图 13-4. 波特率发生器控制寄存器 0 (BRGC0) 的格式

地址: FF71H 复位后: 1FH R/W

符号	7	6	5	4	3	2	1	0
BRGC0	TPS01	TPS00	0	MDL04	MDL03	MDL02	MDL01	MDL00

TPS01	TPS00	基本时钟 (f _{CLK0}) 的选择				
			f _{PRS} = 2 MHz	f _{PRS} = 5 MHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz
0	0	TM50 输出 ^{#2}				
0	1	f _{PRS} /2	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	f _{PRS} /2 ³	250 kHz	625 kHz	1.25 MHz	2.5 MHz
1	1	f _{PRS} /2 ⁵	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz

MDL04	MDL03	MDL02	MDL01	MDL00	k	5 位计数器输出时钟的选择
0	0	×	×	×	×	禁止设置
0	1	0	0	0	8	f _{CLK0} /8
0	1	0	0	1	9	f _{CLK0} /9
0	1	0	1	0	10	f _{CLK0} /10
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
•	•	•	•	•	•	•
1	1	0	1	0	26	f _{CLK0} /26
1	1	0	1	1	27	f _{CLK0} /27
1	1	1	0	0	28	f _{CLK0} /28
1	1	1	0	1	29	f _{CLK0} /29
1	1	1	1	0	30	f _{CLK0} /30
1	1	1	1	1	31	f _{CLK0} /31

注 选择 TM50 输出作为基本时钟时，应注意以下几点。

- TM50 与 CR50 相等时计数时钟清零并启动模式(TMC506 = 0)
先启动 8 位定时器/事件计数器 50，然后允许定时器 F/F 进行反转操作(TMC501 = 1)。

- PWM 模式 (TMC506 = 1)

先启动 8 位定时器/事件计数器 50，然后设置计数时钟，使占空比 = 50%。

不论何种模式，都不必将 TO50 作为定时器输出引脚。

- 注意事项**
1. 在重写 MDL04 ~ MDL00 位时，必须确保 ASIM0 的第 6 位(TXE0)和第 5 位(RXE0) = 0。
 2. 波特率为 5 位计数器输出时钟的 1/2。

- 备注**
1. f_{CLK0}: 根据 TPS01 和 TPS00 位选择的基本时钟的频率
 2. f_{PRS}: 外围硬件时钟频率
 3. k: 由 MDL04 ~ MDL00 位设置的值 (k = 8, 9, 10, ..., 31)
 4. x: 不必考虑
 5. TMC506: 8 位定时器模式控制寄存器 50 (TMC50) 的第 6 位
TMC501: TMC50 的第 1 位

(4) 端口模式寄存器 1 (PM1)

该寄存器按位设置端口 1 为输入/输出模式。

当 P10/TxD0/SCK10 引脚用于串行接口数据输出时，将 PM10 清零且对 P10 的输出锁存器置 1。

当 P11/RxD0/SI10 引脚用于串行接口数据输入时，将 PM11 置 1。此时 P11 的输出锁存器的值可以等于 0 或 1。
可由 1 位或 8 位存储器操作指令设置 PM1。

复位信号的产生将该寄存器设置为 FFH。

图 13-5. 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10
PM1n	P1n 引脚 I/O 模式选择 (n = 0~7)							
0	输出模式 (输出缓冲器打开)							
1	输入模式 (输出缓冲器关闭)							

13.4 串行接口 UART0 的操作

串行接口 UART0 有以下两种模式。

- 操作停止模式
- 异步串行接口(UART)模式

13.4.1 操作停止模式

在此模式下，不能进行串行通信，这样可降低功耗。此外，引脚可作为通用端口引脚使用。通过将 ASIM0 的第 7 位、第 6 位和第 5 位 (POWER0、TXE0 和 RXE0) 清零，可设置操作停止模式。

(1) 使用的寄存器

由异步串行接口操作模式寄存器 0 (ASIM0) 设置操作停止模式。

可由 1 位或 8 位存储器操作指令设置 ASIM0。

复位信号的产生将该寄存器设置为 01H。

地址: FF70H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM0	POWER0	TXE0	RXE0	PS01	PS00	CL0	SL0	1
POWER0	内部操作时钟的允许/禁止操作							
0 ^{注1}	禁止内部操作时钟的操作(时钟恒为低电平)且对内部电路进行异步复位 ^{注2}							
TXE0	允许/禁止发送							
0	禁止发送 (同步复位发送电路)							
RXE0	允许/禁止接收							
0	禁止接收 (同步复位接收电路)							

- 注
1. 当 POWER0 = 0 时，从 RxD0 引脚输入的信号恒为高电平。
 2. 对异步串行接口接收错误状态寄存器 0 (ASIS0)、发送移位寄存器 0 (TXS0) 和接收缓冲寄存器 0 (RXB0) 进行复位。

注意事项 TXE0 和 RXE0 清零后再对 POWER0 清零，可设置操作停止模式。

如果要启动通信，设置 POWER0=1，然后设置 TXE0 或 RXE0=1。

备注 如果要将 RxD0/SI10/P11 和 TxD0/SCK10/P10 作为通用端口引脚使用，可参见 第四章 端口功能。

13.4.2 异步串行接口 (UART)模式

此模式下，在起始位后的 1 字节数据被发送/接收，可执行全双工操作。
可使用专用 UART 波特率发生器，这样通信选择的波特率范围较大。

(1) 使用的寄存器

- 异步串行接口操作模式寄存器 0 (ASIM0)
- 异步串行接口接收错误状态寄存器 0 (ASIS0)
- 波特率发生器控制寄存器 0 (BRGC0)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

在 UART 模式中设置操作的基本过程如下。

- <1> 设置 BRGC0 寄存器 (参见 图 13-4)。
- <2> 设置 ASIM0 的第 1 位~第 4 位 (SL0、CL0、PS00 和 PS01) (参见 图 13-2)。
- <3> 将 ASIM0 的第 7 位 (POWER0) 置 1。
- <4> 将 ASIM0 的第 6 位 (TXE0) 置 1。 → 允许发送。
将 ASIM0 的第 5 位 (RXE0) 置 1。 → 允许接收。
- <5> 对 TXS0 寄存器赋值。 → 开始发送数据。

注意事项 在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。

寄存器设置与引脚之间的关系如下所示。

表 13-2. 寄存器设置与引脚之间的关系

POWER0	TXE0	RXE0	PM10	P10	PM11	P11	UART0 操作	引脚功能	
								TxD0/SCK10/P10	RxD0/SI10/P11
0	0	0	x ^注	x ^注	x ^注	x ^注	停止	SCK10/P10	SI10/P11
1	0	1	x ^注	x ^注	1	x	接收	SCK10/P10	RxD0
	1	0	0	1	x ^注	x ^注	发送	TxD0	SI10/P11
	1	1	0	1	1	x	发送/ 接收	TxD0	RxD0

注 能够作为端口或串行接口 CSI10 使用。

备注

x: 不必考虑

POWER0: 异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位

TXE0: ASIM0 的第 6 位

RXE0: ASIM0 的第 5 位

PM1x: 端口模式寄存器

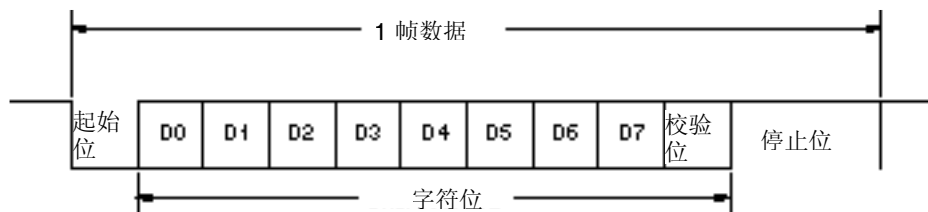
P1x: 端口输出锁存器

(2) 通信操作

(a) 正常发送/接收数据的格式和波形示例

图 13-6 和 13-7 为正常发送/接收数据的格式和波形示例。

图 13-6. UART 正常发送/接收数据的格式



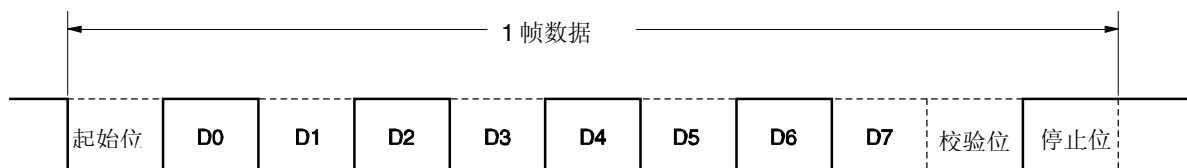
一个数据帧由以下各位组成。

- 起始位 ... 1 位
- 字符位 ... 7 或 8 位 (LSB-first)
- 校验位 ... 偶校验、奇校验、零校验或无校验
- 停止位 ... 1 或 2 位

由异步串行接口操作模式寄存器 0 (ASIM0) 规定一个数据帧中字符位的宽度、校验方式和停止位的宽度。

图 13-7. UART 正常发送/接收数据波形示例

1. 数据宽度: 8 位, 校验位: 偶校验, 停止位: 1 位, 通信数据: 55H



2. 数据宽度: 7 位, 校验位: 奇校验, 停止位: 2 位, 通信数据: 36H



3. 数据宽度: 8 位, 校验位: 无, 停止位: 1 位, 通信数据: 87H



(b) 校验方式与操作

校验位用于检测数据通信中的位错误。通常在发送和接收两端可采用相同的校验方式。当采用偶校验和奇校验时，可检测到 1 位(奇数)错误。而采用零校验和无校验时，则不能检测到错误。

(i) 偶校验

• 发送

控制发送数据，包括校验位，使得数据中“1”的个数为偶数。

校验位的取值如下。

如果发送数据有奇数个“1”：1

如果发送数据有偶数个“1”：0

• 接收

计算接收数据(包括校验位)中“1”的个数。如果结果为奇数，则产生校验错误。

(ii) 奇校验

• 发送

与偶校验方式不同，奇校验用于控制发送数据，包括校验位，使得数据中“1”的个数为奇数。

校验位取值如下。

如果发送数据有奇数个“1”：0

如果发送数据有偶数个“1”：1

• 接收

计算接收数据(包括校验位)中“1”的个数。如果结果为偶数，则产生校验错误。

(iii) 零校验

无论发送数据为何值，发送时校验位被清零。

接收数据时不检测校验位。因此无论校验位为何值，都不会产生校验错误。

(iv) 无校验

发送数据中没有校验位。

进行接收操作时认为接收数据中没有校验位。因此也不会产生校验错误。

(c) 发送

当异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位 (POWER0)=1 且 ASIM0 的第 6 位 (TXE0)=1 时, 允许发送。将待发送数据写入发送移位寄存器 0 (TXS0), 可启动发送操作。起始位、校验位和停止位会自动添加到数据中。

启动发送操作时, 先从 TxD0 引脚输出起始位, 然后从 LSB 开始依次发送数据其它位。当发送结束时, 由 ASIM0 设置的校验位和停止位被添加到数据中, 并产生发送完成中断请求 (INTST0)。

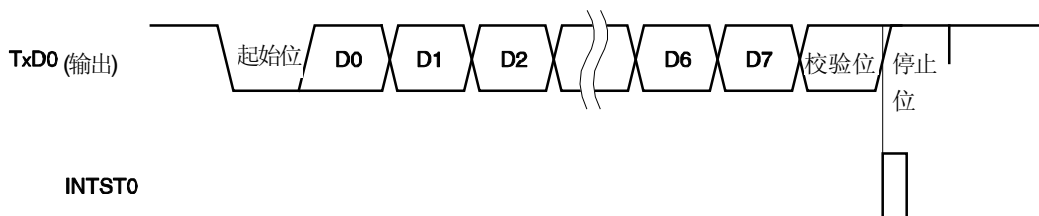
当下一个待发送数据被写入 TXS0 时, 发送操作才结束。

图 13-8 显示了发送完成中断请求(INTST0)的时序。最后一个停止位一输出就产生该中断。

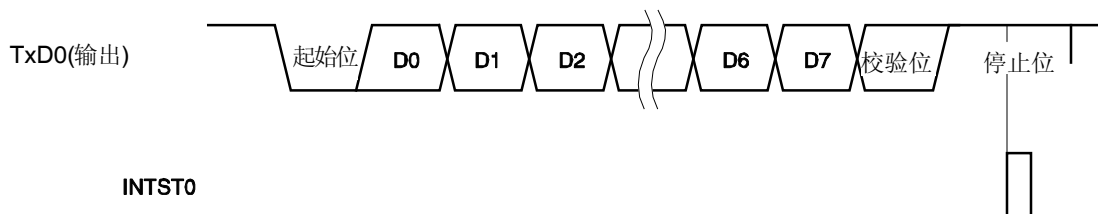
注意事项 在将待发送数据写入 TXS0 后, 在发送完成中断信号 (INTST0) 产生之前, 不要将下一个发送数据写入 TXS0。

图 13-8. 发送完成中断请求的时序

1. 停止位宽度: 1



2. 停止位宽度: 2



(d) 接收

当异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位 (POWER0)=1, 然后将 ASIM0 的第 5 位 (RXE0) 置 1 时, 允许接收并对 RxD0 引脚输入进行采样。

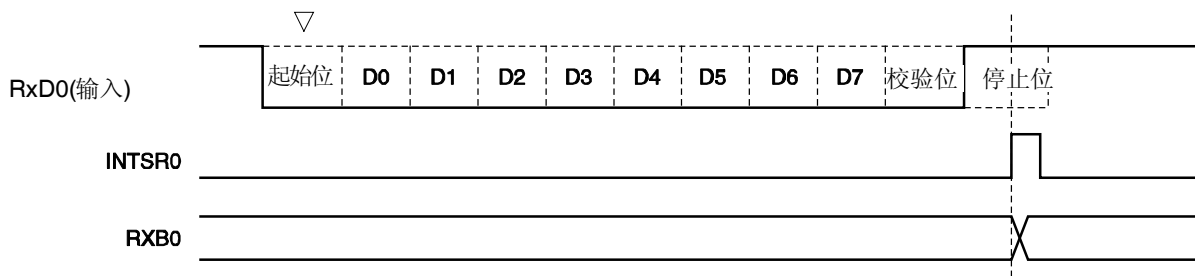
当检测到 RxD0 引脚输入的下降沿时波特率发生器的 5 位计数器开始计数。当计数值等于波特率发生器控制寄存器 0 (BRGC0) 的设置值时, 再次采样 RxD0 的输入信号 (如图 14-9 中的 ∇)。如果此时 RxD0 为低电平, 则认为输入信号为起始位。

当检测到起始位时, 开始接收, 以设置的波特率将串行数据依次存入接收移位寄存器 0 (RXS0)。当接收到停止位时, 会产生接收完成中断信号 (INTSR0), 并且将 RXS0 的数据写入接收缓冲寄存器 0 (RXB0)。但如果出现溢出错误 (OVE0), 则接收到的数据不写入 RXB0。

即使在接收过程中出现校验错 (PE0), 接收仍然继续, 直至接收到停止位, 而在接收完成后会产生一个接收错误中断 (INTSR0)。

当接收操作完成且存在接收错误, 则产生 INTSR0。

图 13-9. 接收完成中断请求时序



- 注意事项**
1. 如果出现接收错误, 必须先读取异步串行接口接收错误状态寄存器 0 (ASIS0) 再读取接收缓冲寄存器 0 (RXB0) 的内容, 才能对出错标志清零。否则, 当接收到下一个数据时会产生溢出错误, 而且接收错误状态保持不变。
 2. 接收始终按“停止位的个数 = 1”的情况执行, 第 2 个停止位被忽略。

(e) 接收错误

在接收期间有三种出错情况: 校验错误、帧错误或溢出错误。如果异步串行接口接收错误状态寄存器 0 (ASIS0) 的错误标志位被设置, 作为数据接收的结果, 则将会产生一个接收错误中断请求信号 (INTSR0)。

在执行接收错误中断服务 (INTSR0) 时读取 ASIS0 的内容, 可确定接收期间出现的是何种错误 (参见图 13-3)。

读取 ASIS0 后, 将其清 0。

表 13-3. 接收错误产生的原因

接收错误	原因
校验错误	发送数据的校验位与接收数据的校验位不相等
帧错误	未检测到停止位
溢出错误	在从接收缓冲寄存器 0 (RXB0)中读取数据前已完成下一个数据的接收操作

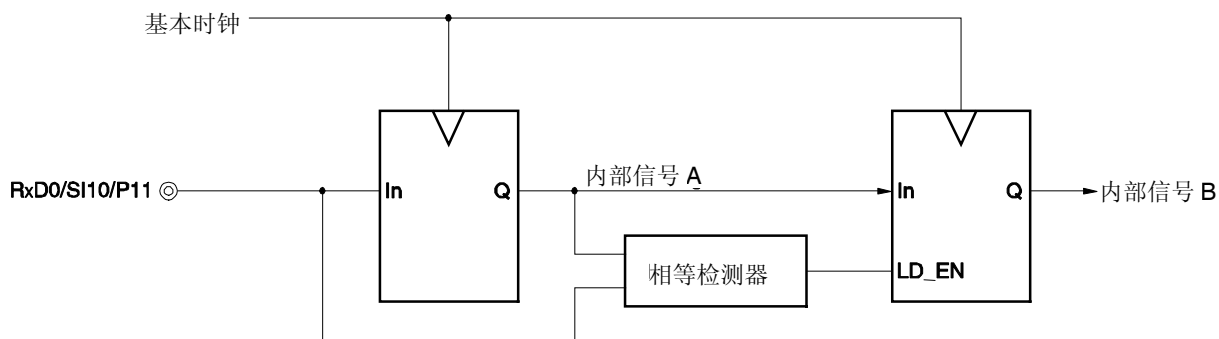
(f) 接收数据的噪声过滤器

通过预分频器, 使用基本时钟输出, 采样 RxD0 信号。

如果两次采样值相同, 则相等检测器的输出会发生变化, 并把采样的数据作为输入数据。

噪声过滤器的电路结构如图 13-10 所示, 接收操作的内部过程与外部信号状态相比被延迟两个时钟执行。

图 13-10. 噪声过滤器电路



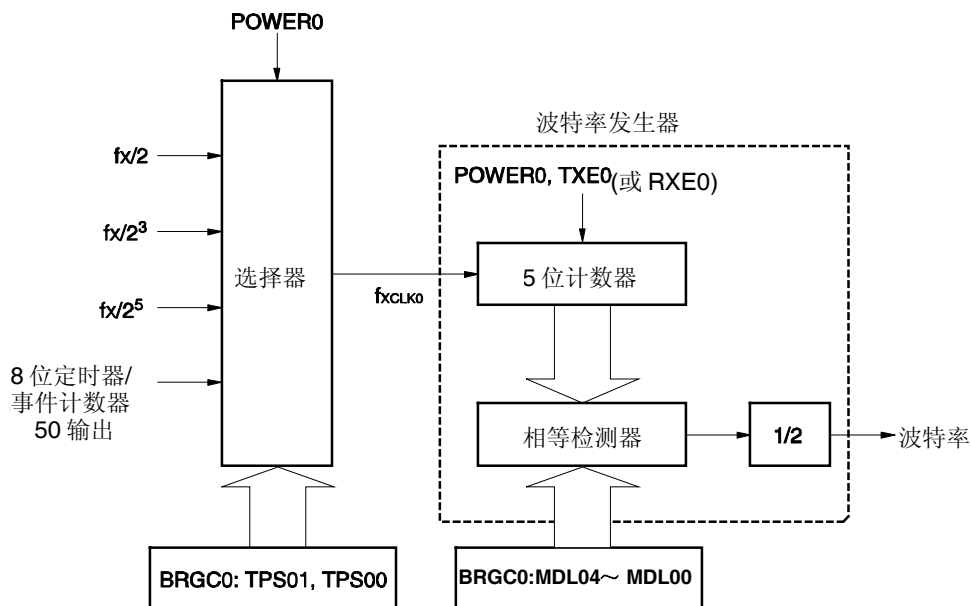
13.4.3 专用波特率发生器

专用波特率发生器由一个源时钟选择器和一个 5 位可编程计数器组成，用于产生 UART0 发送/接收的串行时钟。分别使用不同的 5 位计数器用于发送和接收。

(1) 波特率发生器的配置

- 基本时钟**
 该时钟由波特率发生器控制寄存器 0 (BRGC0) 的第 7 位和第 6 位 (TPS01 和 TPS00) 选择。当异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位 (POWER0)=1 时，将该时钟提供给每个模块。该时钟称为基本时钟，它的频率用 f_{CLK0} 表示。当 $POWER0 = 0$ 时，基本时钟恒为低电平。
- 发送计数器**
 当异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位 (POWER0) 或第 6 位 (TXE0) = 0 时，该计数器停止计数，并被清零。
 当 $POWER0 = 1$ 且 $TXE0 = 1$ 时计数器开始计数。
 当第 1 个发送数据被写入发送移位寄存器 0 (TXS0) 时，计数器被清零。
- 接收计数器**
 当异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位 (POWER0) 或第 5 位 (RXE0) = 0 时，该计数器停止计数，并被清零。
 当检测到起始位时，计数器开始计数。
 在收到一帧数据后计数器停止计数，直到下一个起始位被检测到时再重新开始计数。

图 13-11. 波特率发生器的配置



备注

- POWER0: 异步串行接口操作模式寄存器 0 (ASIM0) 的第 7 位
- TXE0: ASIM0 的第 6 位
- RXE0: ASIM0 的第 5 位
- BRGC0: 波特率发生器控制寄存器 0

(2) 串行时钟的发生器

使用波特率发生器控制寄存器 0 (BRGC0)可产生串行时钟。

根据 BRGC0 的第 7 位和第 6 位 (TPS01 和 TPS00)，可选择输入给 5 位计数器的时钟。

根据 BRGC0 的第 4 位 ~ 第 0 位 (MDL04 ~ MDL00) 选择 5 位计数器的分频值 ($f_{CLK0}/8 \sim f_{CLK0}/31$)。

表 13-4. TPS01 和 TPS00 的设置值

TPS01	TPS00	基本时钟 (f_{CLK0}) 的选择				
		$f_{PRS} = 2 \text{ MHz}$	$f_{PRS} = 5 \text{ MHz}$	$f_{PRS} = 10 \text{ MHz}$	$f_{PRS} = 20 \text{ MHz}$	
0	0	TM50 输出				
0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
1	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz

(a) 波特率

通过下列公式计算波特率。

$$\bullet \text{ 波特率} = \frac{f_{CLK0}}{2 \times k} \text{ [bps]}$$

f_{CLK0} : 根据 BRGC0 的 TPS01 和 TPS00 位选择的基本时钟的频率

k: 根据 BRGC0 的 MDL04 ~ MDL00 位设置的值 ($k = 8, 9, 10, \dots, 31$)

(b) 波特率误差

通过下列公式计算波特率误差。

$$\bullet \text{ 误差 (\%)} = \left[\frac{\text{实际波特率 (有误差的波特率)}}{\text{预期波特率 (正确的波特率)}} - 1 \right] \times 100 \text{ [\%]}$$

- 注意事项**
1. 在发送过程中必须保持波特率误差在接收方允许的误差范围内。
 2. 接收期间的波特率误差必须满足“(4)接收期间允许的波特率范围”中所描述的范围。

示例:

基本时钟的频率 = 2.5 MHz = 2,500,000 Hz
 BRGC0 的 MDL04 ~ MDL00 位 = 10000B ($k = 16$)
 目标波特率 = 76,800 bps

$$\begin{aligned} \text{波特率} &= 2.5 \text{ M}/(2 \times 16) \\ &= 2,500,000/(2 \times 16) = 78,125 \text{ [bps]} \end{aligned}$$

$$\begin{aligned} \text{误差} &= (78,125/76,800 - 1) \times 100 \\ &= 1.725 \text{ [\%]} \end{aligned}$$

(3) 波特率设置举例

表 13-5. 波特率发生器的数据设置

波特率 [bps]	f _{PRS} = 2.0 MHz				f _{PRS} = 5.0 MHz				f _{PRS} = 10.0 MHz				f _{PRS} = 20.0 MHz			
	TPS01, TPS00	k	计算值	ERR [%]	TPS01, TPS00	k	计算值	ERR [%]	TPS01, TPS00	k	计算值	ERR [%]	TPS01, TPS00	k	计算值	ERR [%]
4800	2	26	4808	0.16	3	16	4883	1.73	-	-	-	-	-	-	-	-
9600	2	13	9615	0.16	3	8	9766	1.73	3	16	9766	1.73	-	-	-	-
10400	2	12	10417	0.16	2	30	10417	0.16	3	15	10417	0.16	3	30	10417	0.16
19200	1	26	19231	0.16	2	16	19531	1.73	3	8	19531	1.73	3	16	19531	1.73
24000	1	21	23810	-0.79	2	13	24038	0.16	2	26	24038	0.16	3	13	24038	0.16
31250	1	16	31250	0	2	10	31250	0	2	20	31250	0	3	10	31250	0
33660	1	15	33333	-0.79	2	9	34722	3.34	2	18	34722	3.34	3	9	34722	3.34
38400	1	13	38462	0.16	2	8	39063	1.73	2	16	39063	1.73	3	8	39063	1.73
56000	1	9	55556	-0.79	1	22	56818	1.46	2	11	56818	1.46	2	22	56818	1.46
62500	1	8	62500	0	1	20	62500	0	2	10	62500	0	2	20	62500	0
76800	-	-	-	-	1	16	78125	1.73	2	8	78125	1.73	2	16	78125	1.73
115200	-	-	-	-	1	11	113636	-1.36	1	22	113636	-1.36	2	11	113636	-1.36
153600	-	-	-	-	1	8	156250	1.73	1	16	156250	1.73	2	8	156250	1.73

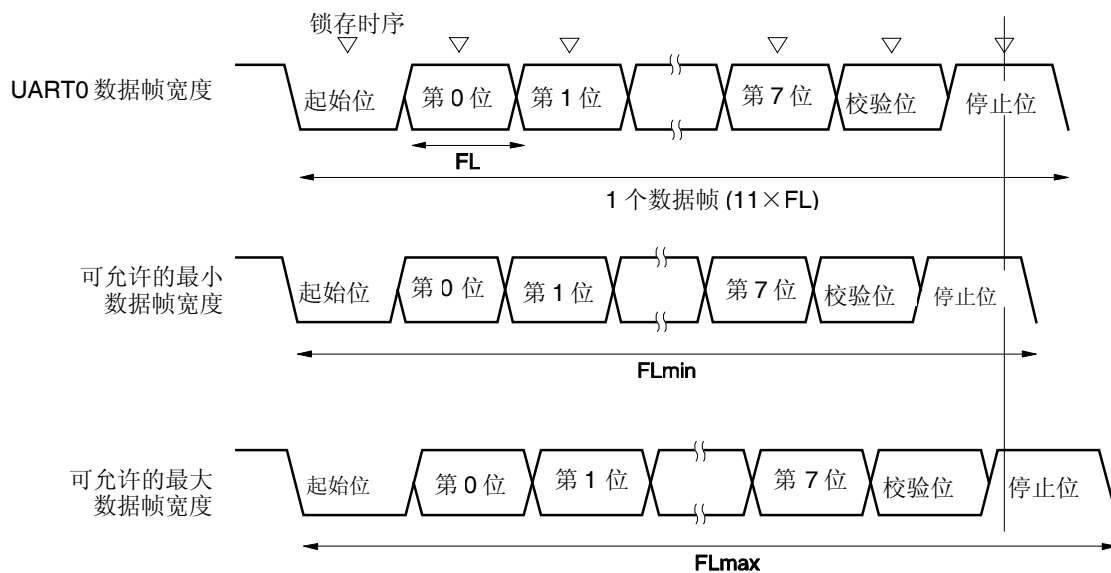
备注 TPS01, TPS00: 波特率发生器控制寄存器 0 (BRGC0) 的第 7 位和第 6 位 (用于设置基本时钟(f_{CLK0}))
k: 根据 BRGC0 的 MDL04 ~ MDL00 位设置的值 (k = 8, 9, 10, ..., 31)
f_{PRS}: 外围硬件时钟频率
ERR: 波特率误差

(4) 接收期间允许的波特率范围

接收期间来自发送端的波特率可允许的误差范围如下所示。

注意事项 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。

图 13-12. 接收期间允许的波特率范围



如图 13-12 所示，当检测到起始位后，接收数据的锁存时序由波特率发生器控制寄存器 0 (BRGC0) 设置的计数器来确定。如果数据的最后一位(停止位)满足该锁存时序，则该数据可被正确接收。

假定接收数据为 11 位，各项理论值计算如下。

$$FL = (\text{Brate})^{-1}$$

Brate: UART0 的波特率

k: BRGC0 设置的值

FL: 1 位数据宽度

锁存时序的极限值: 2 个时钟

可允许的最小数据帧宽度: $FL_{min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

因此在发送端设置的可接收的最大波特率如下所示。

$$BR_{max} = (FL_{min}/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

同样，可计算允许的最大数据帧宽度。

$$\frac{10}{11} \times FL_{max} = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FL_{max} = \frac{21k-2}{20k} FL \times 11$$

因此在发送端设置的可接收的最小波特率如下所示。

$$BR_{min} = (FL_{max}/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

可通过上述最小和最大波特率公式计算在 UART0 与发送端之间允许的波特率误差，如下所示。

表 13-6. 允许的最大/最小波特率误差

分频比 (k)	允许的最大波特率误差	允许的最小波特率误差
8	+3.53%	-3.61%
16	+4.14%	-4.19%
24	+4.34%	-4.38%
31	+4.44%	-4.47%

- 备注**
1. 接收时允许的误差与每帧的位数、输入时钟频率和分频比(k)有关。输入时钟频率和分频比(k)越高，允许的误差就越大。
 2. k: BRGC0 设置的值。

14.1 串行接口 UART6 的功能

串行接口 UART6 有以下两种模式。

(1) 操作停止模式

在不进行串行通信时可采用此模式，以降低功耗。

需要了解详细信息，可参见 **14.4.1 操作停止模式**。

(2) 异步串行接口(UART) 模式

该模式支持 LIN (本地互联网) 总线。该模式的功能概括如下。

需要了解详细信息，可参见 **14.4.2 异步串行接口(UART)模式** 和 **14.4.3 专用波特率发生器**。

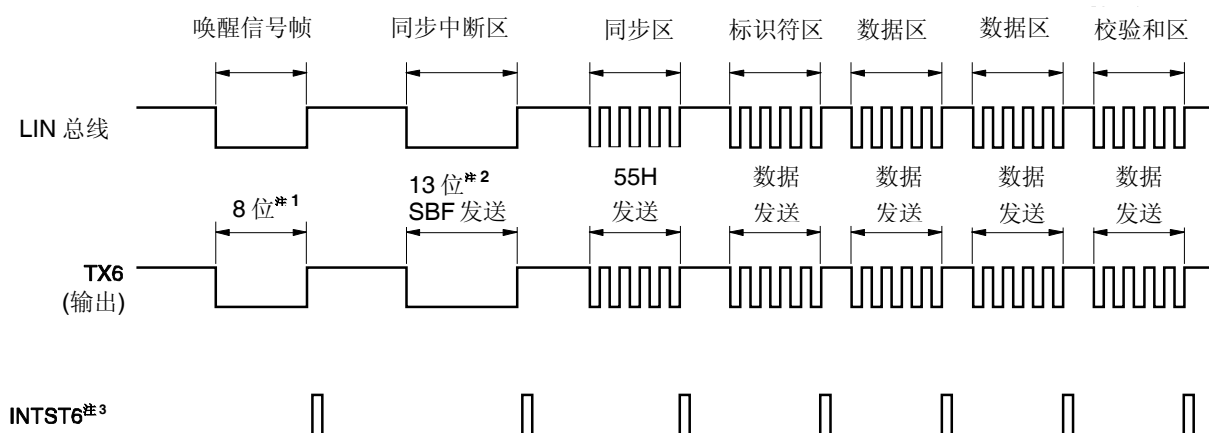
- 最大传输率: 312.5 kbps
- 双引脚配置 TxD6: 发送数据输出引脚
RxD6: 接收数据输入引脚
- 可选择通信数据的数据宽度为 7 或 8 位。
- 专用内置 8 位波特率发生器允许设置任意大小的波特率。
- 可独立执行发送和接收操作(全双工操作)。
- 可选 MSB-first 或 LSB-first 通信。
- 反向发送操作
- 同步中断区域发送长度为 13 ~ 20 位
- 可规定多于 11 位的同步中断区域接收 (提供 SBF 接收标志位)。

- 注意事项**
1. **TxD6** 输出反向操作仅在发送端进行而不在接收端进行。为了能够使用该功能，接收端应准备好接收已反向的数据。
 2. 如果串行接口 **UART6** 使用的时钟未被停止(例如，处于 **HALT** 模式下)，则继续正常操作。如果串行接口 **UART6** 使用的时钟已被停止(例如，处于 **STOP** 模式下)，则寄存器停止操作，并且在时钟停止之前及时保存数据。**TXD6** 引脚也在时钟停止之前立即保存数据并输出。但在时钟恢复使用后正常操作不能得到保证。因此必须复位电路，使 **POWER6 = 0**、**RXE6 = 0**、**TXE6 = 0**。
 3. 设置 **POWER6 = 1** 然后设置 **TXE6 = 1** (发送) 或 **RXE6 = 1** (接收)，以启动通信。
 4. 通过基本时钟 (**f_{CLK6}**) (由 **CKSR6** 设置) 对 **TXE6** 与 **RXE6** 进行同步。若要再次允许发送或接收，必须在 **TXE6** 或 **RXE6** 清零 (0) 后至少两个基本时钟时将 **TXE6** 或 **RXE6** 设置为 1。如果在两个基本时钟内设置 **TXE6** 或 **RXE6**，则发送电路或接收电路可能不被初始化。
 5. 在设置 **TXE6 = 1** 后至少经历一个基本时钟 (**f_{CLK6}**) 再为 **TXB6** 设置发送数据。
 6. 如果连续发送数据，从停止位到下一个起始位的通信时序将延长两个宏操作时钟。但这不会影响到通信结果，因为接收端会在检测到起始位时对时序初始化。如果 **UART6** 用于 **LIN** 模式，则不能使用连续发送功能。

备注 LIN 表示本地互联网，是一个低速(1 ~ 20 kbps)串行通信协议，可用于降低网络费用。
 LIN 是一种单主设备 (single-master) 通信方式，一个主设备上最多可以连接 15 个从设备。
 LIN 的从设备用于控制开关、制动器和传感器，这些设备通过 LIN 网与 LIN 主设备相连。
 通常，LIN 主设备与一个网络，如 CAN (控制器局域网 Controller Area Network) 相连。
 此外，LIN 总线采用单线方式，通过收发器(符合 ISO9141)与各节点相连。
 在 LIN 协议中，主设备发送带有波特率信息的一帧数据，从设备接收数据并校正波特率误差。因此当从设备端的波特率误差在 $\pm 15\%$ 范围内时，可以进行通信。

图 14-1 和 14-2 概括了 LIN 的发送和接收操作。

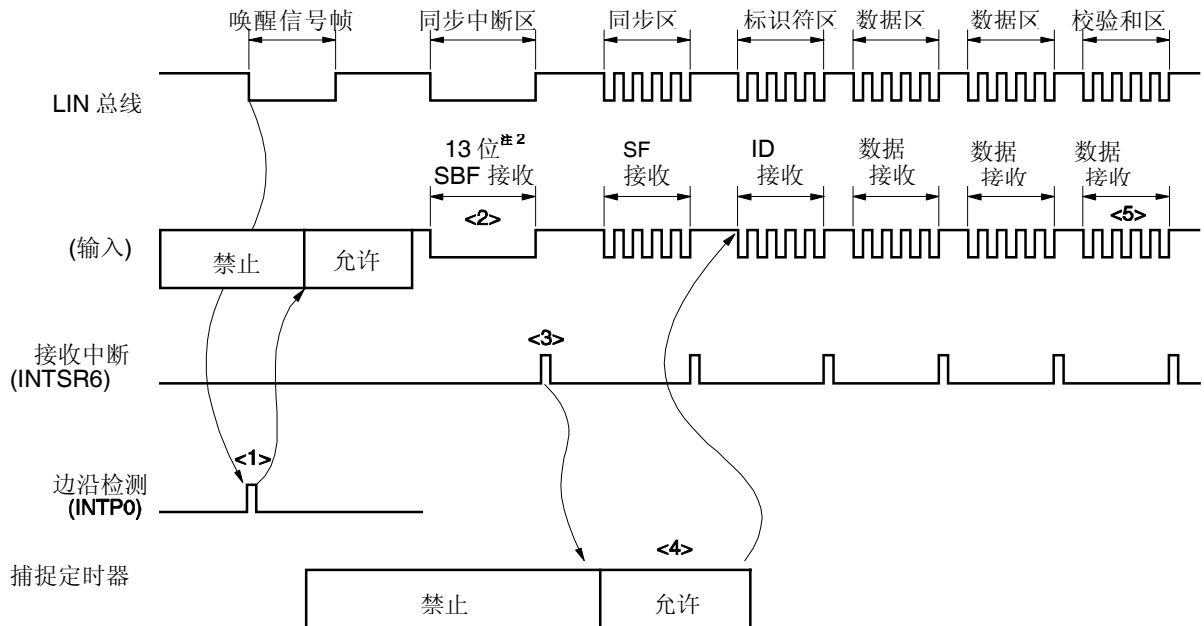
图 14-1. LIN 发送操作



- 注**
1. 在 8 位发送模式中，可通过发送 80H 替代唤醒信号帧。
 2. 使用硬件输出同步中断区。可通过异步串行接口控制寄存器 6 (ASICL6) 的第 4 位 ~ 第 2 位 (SBL62 ~ SBL60) 设置输出宽度 (等于位长度) (参见 14.4.2 (2) (h) SBF 发送)。
 3. 每次发送完成时输出 INTST6。在发送 SBF 时也输出 INTST6。

备注 区域之间的间隔由软件控制。

图 14-2. LIN 接收操作



接收处理过程如下。

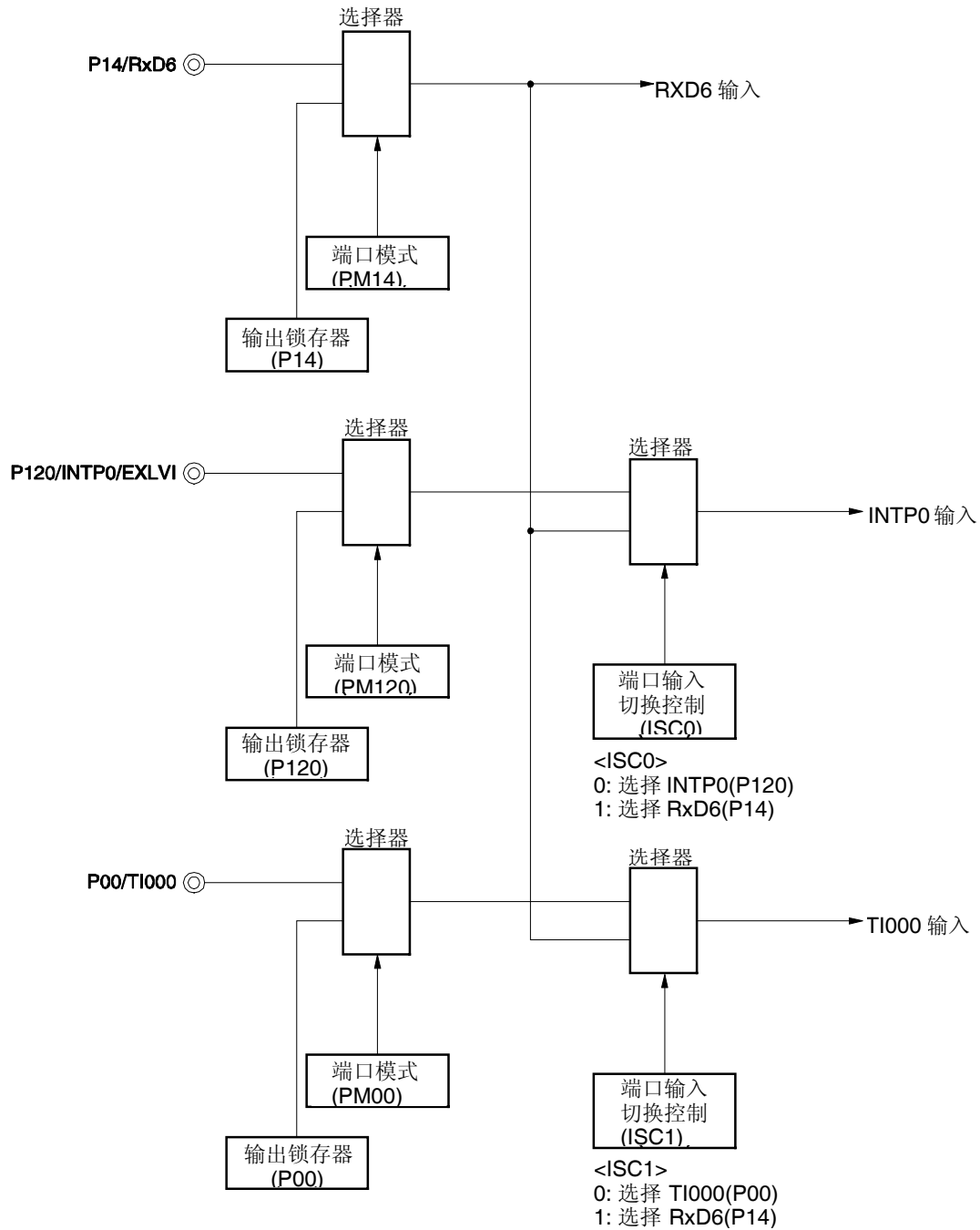
- <1> 在引脚的边沿处检测唤醒信号，允许使用 UART6，并设置 SBF 接收模式。
- <2> 当检测到停止位时才停止接收操作。在检测到至少 11 位低电平数据的 SBF 时，认为 SBF 已被正确接收，并输出一个中断信号。在检测到少于 11 位低电平数据的 SBF 时，认为出现 SBF 接收错误。此时不输出中断信号，而是恢复 SBF 接收模式。
- <3> 如果 SBF 被正确接收，将输出一个中断信号。用这个 SBF 接收完成中断服务来启动 16 位定时器/事件计数器 00 并测量同步区的位长度（脉冲宽度）（参见 6.4.3 脉冲宽度测量操作（定时器 00））。禁止检测 OVE6、PE6 和 FE6 错误，不进行 UART 通信的错误检测，也不执行移位寄存器和 RXB6 之间的数据传送。移位寄存器中保存复位值 FFH。
- <4> 根据同步区的位长度计算波特率误差，在 SBF 接收后禁止使用 UART6，然后对波特率发生器控制寄存器 6 (BRGC6) 重新设置。
- <5> 用软件区分校验和区。在接收到校验和区后通过软件对 UART6 进行初始化，并再次设置 SBF 接收模式。

图 14-3 显示了 LIN 接收操作的端口配置。

通过检测外部中断信号 (INTP0) 脉冲沿，接收从 LIN 主设备发出的唤醒信号。根据 16 位定时器/事件计数器 00 的外部事件捕捉操作来测量 LIN 主设备发送的同步区的位长度，并计算波特率误差。

不进行外部连接 RxD6 和 INTP0/TI000，而由端口输入切换控制 (ISC0/ISC1) 可将接收端口的 (RxD6) 的输入源输入给外部中断 (INTP0) 和 16 位定时器/事件计数器 00。

图 14-3. LIN 接收操作的端口配置



备注 ISC0, ISC1: 输入切换控制寄存器(ISC)的第0位和第1位 (参见 图 14-11)

以下是 LIN 通信中用到的外部功能。

<用到的外部功能>

- 外部中断(INTP0): 唤醒信号检测
用途: 检测唤醒信号的脉冲沿和通信的开始。
- 16位定时器/事件计数器 00 (TI000): 波特率误差检测
用途: 通过检测同步区(SF)的位长度并按位分频, 来检测波特率误差(在捕捉模式下测量 TI000 输入脉冲的间隔)。
- 串行接口 UART6

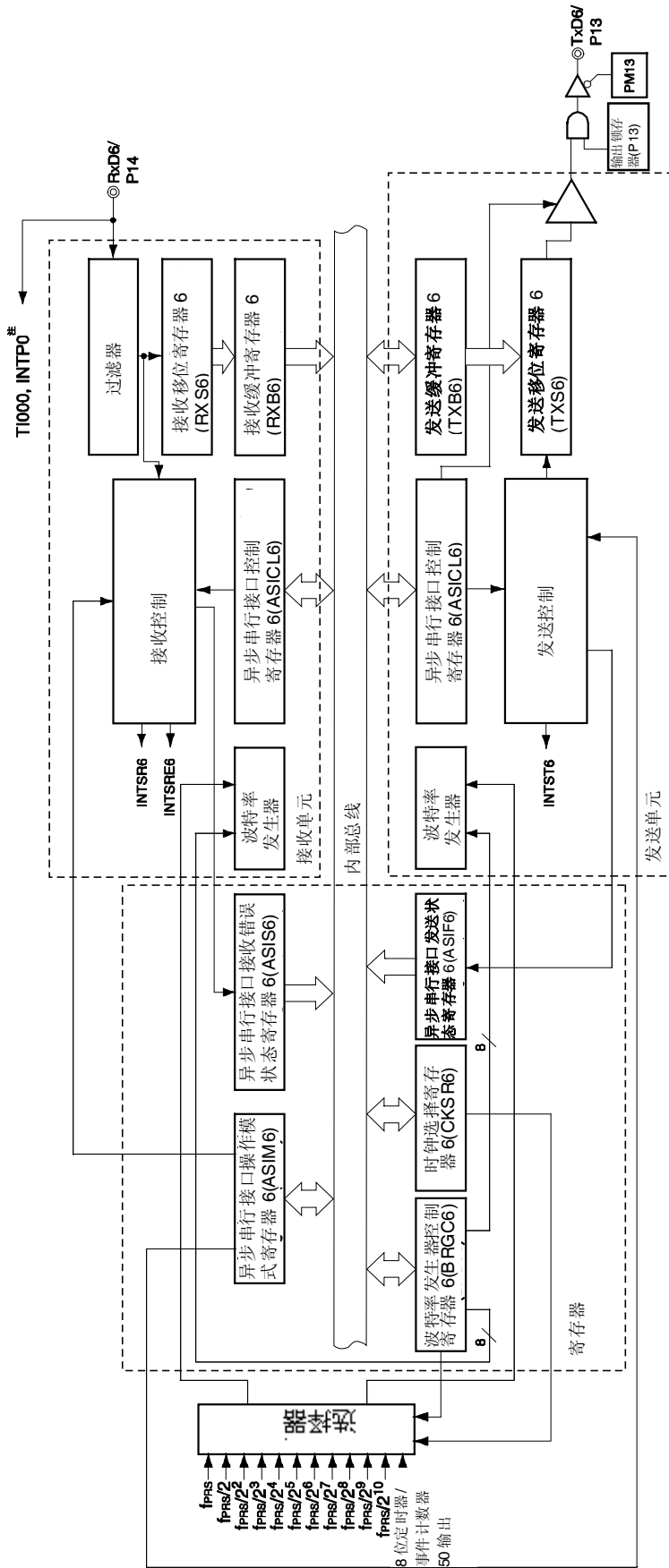
14.2 串行接口 UART6 的配置

串行接口 UART6 包括以下硬件。

表 14-1. 串行接口 UART6 的配置

项目	配置
寄存器	接收缓冲寄存器 6 (RXB6) 接收移位寄存器 6 (RXS6) 发送缓冲寄存器 6 (TXB6) 发送移位寄存器 6 (TXS6)
控制寄存器	异步串行接口操作模式寄存器 6 (ASIM6) 异步串行接口接收错误状态寄存器 6 (ASIS6) 异步串行接口发送状态寄存器 6 (ASIF6) 时钟选择寄存器 6 (CKSR6) 波特率发生器控制寄存器 6 (BRGC6) 异步串行接口控制寄存器 6 (ASICL6) 输入切换控制寄存器 (ISC) 端口模式寄存器 1 (PM1) 端口寄存器 1 (P1)

图 14-4. 串行接口 UART6 的框图



注 使用输入切换控制寄存器 (ISC) 进行选择。

(1) 接收缓冲寄存器 6 (RXB6)

该 8 位寄存器用于存储由接收移位寄存器(RXS6)转换的并行数据。

每接收到 1 字节的数据，新的接收数据就从 RXS6 传送到该寄存器中。如果数据宽度被设定为 7 位，则按如下方式发送数据。

- 在 LSB-first 接收模式下，将接收到的数据传送到 RXB6 的第 0 位 ~ 第 6 位，而 MSB 位恒为 0。
- 在 MSB-first 接收模式下，将接收到的数据传送到 RXB6 的第 1 位 ~ 第 7 位，而 LSB 位恒为 0。

如果出现溢出错误(OVE6)，则不把接收数据传送到 RXB6 中。

可由 8 位存储器操作指令读取 RXB6 的内容。不能将数据写入该寄存器中。

复位信号产生将该寄存器设置为 FFH。

(2) 接收移位寄存器 6 (RXS6)

该寄存器将输入到 RxD6 引脚的串行数据转换成并行数据。

程序不能直接使用 RXS6。

(3) 发送缓冲寄存器 6 (TXB6)

该寄存器用于设置发送数据。当数据写入 TXB6 时开始发送。

可由 8 位存储器操作指令对 TXB6 进行读或写操作。

复位信号产生将该寄存器设置为 FFH。

注意事项 1. 当异步串行接口发送状态寄存器 6 (ASIF6) 的第 1 位 (TXBF6)=1 时不要对 TXB6 进行写操作。

2. 在通信过程中(当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 和第 6 位 (TXE6) 均为 1，或 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 均为 1)，不要通过软件刷新 (写入相同值) TXB6。

3. 在设置 TXE6 = 1 后，至少经过一个基本时钟(f_{XCLK6})，将发送数据写入 TXB6。

(4) 发送移位寄存器 6 (TXS6)

该寄存器将 TXB6 中要传送的数据作为串行数据从 TxD6 移出。在第 1 次发送过程中，当数据写入 TXB6 后立即被传送。或者在持续发送过程中，当发出 1 帧数据后，INTST6 中断产生之前数据立即被传送。利用基本时钟的下降沿从 TxD6 引脚传送来自 TXB6 的数据。

程序不能直接对 TXS6 操作。

14.3 控制串行接口 UART6 的寄存器

串行接口 UART6 由以下 9 个寄存器控制。

- 异步串行接口操作模式寄存器 6 (ASIM6)
- 异步串行接口接收错误状态寄存器 6 (ASIS6)
- 异步串行接口发送状态寄存器 6 (ASIF6)
- 时钟选择寄存器 6 (CKSR6)
- 波特率发生器控制寄存器 6 (BRGC6)
- 异步串行接口控制寄存器 6 (ASICL6)
- 输入切换控制寄存器 (ISC)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

(1) 异步串行接口操作模式寄存器 6 (ASIM6)

该 8 位寄存器用于控制串行接口 UART6 的串行通信。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号产生将该寄存器设置为 01H。

备注 在通信过程中 (当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 和第 6 位 (TXE6) 均为 1, 或 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 均为 1), 可以通过软件刷新 (写入相同值) ASIM6。

图 14-5. 异步串行接口操作模式寄存器 6 (ASIM6) 的格式 (1/2)

地址: FF50H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6
POWER6	允许/禁止内部操作时钟的操作							
0 ^{#1}	禁止内部操作时钟的操作 (时钟恒为低电平) 并对内部电路进行异步复位 ^{#2} 。							
1	允许内部操作时钟的操作							
TXE6	允许/禁止发送							
0	禁止发送 (同步复位发送电路)							
1	允许发送							
RXE6	允许/禁止接收							
0	禁止接收 (同步复位接收电路)							
1	允许接收							

- 注**
1. 发送期间当 POWER6 = 0 时, TxD6 引脚的输出变为高电平, 而来自 RxD6 引脚的输入恒为高电平。
 2. 对异步串行接口接收错误状态寄存器 6 (ASIS6)、异步串行接口发送状态寄存器 6 (ASIF6)、异步串行接口控制寄存器 6 (ASICL6) 的第 7 位 (SBRF6) 和第 6 位 (SBRT6), 以及接收缓冲寄存器 6 (RXB6) 进行复位。

图 14-5. 异步串行接口操作模式寄存器 6 (ASIM6) 的格式 (2/2)

PS61	PS60	发送操作	接收操作
0	0	不输出校验位	无校验接收
0	1	输出零校验	按零校验接收 ^注
1	0	输出奇校验	按奇校验判断
1	1	输出偶校验	按偶校验判断

CL6	指定发送/接收数据的字符宽度
0	数据的字符宽度 = 7 位
1	数据的字符宽度 = 8 位

SL6	指定发送数据停止位的个数
0	停止位的个数 = 1
1	停止位的个数 = 2

ISRM6	允许/禁止出错时接收完成中断的产生
0	出错时产生“INTSRE6”(此时, 不产生 INTSR6)
1	出错时产生“INTSR6”(此时, 不产生 INTSRE6)

注 如果选择“按零校验接收”，则无法判断校验。因此不要设置异步串行接口接收错误状态寄存器 0 (ASIS6) 的第 2 位 (PE6)，这样也不会产生出错中断。

- 注意事项**
1. 启动发送操作时，对 POWER6 置 1，然后对 TXE6 置 1。停止发送操作时，先将 TXE6 清零，然后将 POWER6 清零。
 2. 启动接收操作时，对 POWER6 置 1，然后对 RXE6 置 1。停止接收操作时，先将 RXE6 清零，然后将 POWER6 清零。
 3. 当 RxD6 引脚输入为高电平时，先对 POWER6 置 1，再对 RXE6 置 1。当输入为低电平时，POWER6 置 1，RXE6 置 1 时，启动接收操作。
 4. 通过基本时钟 (f_{XCLK6}) (由 CKSR6 设置) 对 TXE6 与 RXE6 进行同步。若要再次允许发送或接收，必须在 TXE6 或 RXE6 清零 (0) 后至少两个基本时钟时将 TXE6 或 RXE6 设置为 1。如果在两个基本时钟内设置 TXE6 或 RXE6，则发送电路或接收电路可能不被初始化。
 5. 设置 TXE6 = 1 后，至少经过一个基本时钟 (f_{XCLK6})，将发送数据写入 TXB6。
 6. 在对 PS61、PS60 和 CL6 位重写之前，先将 TXE6 和 RXE6 位清零。
 7. 工作在 LIN 模式时，PS61 与 PS60 恒为 0。
 8. 在重写 SL6 位时，必须确保 TXE6 = 0。总是以“停止位的个数=1”的格式执行接收操作，因此接收不会受到 SL6 设置值的影响。
 9. 当重写 ISRM6 时，RXE6 必须为 0。

(2) 异步串行接口接收错误状态寄存器 6 (ASIS6)

该寄存器用于指示串行接口 UART6 的接收操作完成时出现的错误状态，包括 3 个出错标志位 (PE6, FE6, OVE6)。

可由 8 位存储器操作指令对该寄存器进行只读操作。

如果 ASIM6 的第 5 位 (RXE6) 或第 7 位 (POWER6) = 0，则复位信号产生将该寄存器设置为 00H。读取该寄存器的值为 00H。如果产生接收错误，读取 ASIS6 再读取接收缓冲寄存器 6 (RXB6)，可将出错标志清零。

图 14-6. 异步串行接口接收错误状态寄存器 6 (ASIS6) 的格式

地址: FF53H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIS6	0	0	0	0	0	PE6	FE6	OVE6

PE6	指示校验错误的状态标志位
0	如果 POWER6 = 0 且 RXE6 = 0，或者如果读取 ASIS6
1	如果发送数据的校验位与接收完成时的校验位不相等

FE6	指示帧错误的状态标志位
0	如果 POWER6 = 0 且 RXE6 = 0、或者如果读取 ASIS6
1	如果接收完成时未检测到停止位

OVE6	指示溢出错误的状态标志位
0	如果 POWER6 = 0 且 RXE6 = 0、或 ASIS6 被读取
1	如果将接收数据存于 RXB6 寄存器中且在读取该数据前下一个接收操作已完成

- 注意事项**
1. 根据异步串行接口操作模式寄存器 6 (ASIM6) 的 PS61 和 PS60 位的内容的不同，对 PE6 位的操作也有所不同。
 2. 无论停止位的个数是多少，只将接收数据的第 1 位作为停止位进行检测。
 3. 如果出现溢出错误，则下一个接收数据不写入接收缓冲寄存器 6 (RXB6) 中，而是被忽略。
 4. 如果从 ASIS6 读取数据，会产生一个等待周期。当 CPU 采用副系统时钟而停止外部硬件时钟操作时，不要从 ASIS6 读取数据。需要了解详细信息，可参见 第三十四章 等待注意事项。

(3) 异步串行接口发送状态寄存器 6 (ASIF6)

该寄存器指示串行接口 UART6 的发送状态，包括两个状态标志位(TXBF6 和 TXSF6)。

当数据从 TXB6 传送到 TXS6 中后，通过将下一个发送数据写入 TXB6，使得发送能够连续进行而不被中断，即便是在中断产生期间也不受影响。

可由 8 位存储器操作指令对该寄存器进行只读操作。

如果 ASIM6 的第 7 位 (POWER6) 或第 6 位 (TXE6) 置 0，则复位信号产生将该寄存器设置为 00H。

图 14-7. 异步串行接口发送状态寄存器 6 (ASIF6) 的格式

地址: FF55H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
ASIF6	0	0	0	0	0	0	TXBF6	TXSF6

TXBF6	发送缓冲数据标志位
0	如果 POWER6 = 0 或 TXE6 = 0，或如果数据被传送到发送移位寄存器 6 (TXS6) 中
1	当数据被写入发送缓冲寄存器 6 (TXB6) 中 (如果数据已在 TXB6 中)

TXSF6	发送移位寄存器数据标志位
0	如果 POWER6 = 0 或 TXE6 = 0，或如果发送完成后下一个数据没有从发送缓冲寄存器 6 (TXB6) 发出
1	如果数据从发送缓冲寄存器 6 (TXB6) 发出 (如果数据仍在发送过程中)

- 注意事项**
1. 为了能够连续发送数据，应将第一个发送的数据 (第一个字节) 写入 TXB6 中。之后，检查 TXBF6 标志位，必须为“0”。如果正确，将下一个发送数据 (第二个字节) 写入 TXB6 中。如果 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。
 2. 如果要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查 TXSF6 标志位，应为“0”。然后执行初始化操作。如果 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。

(4) 时钟选择寄存器 6 (CKSR6)

该寄存器用于选择串行接口 UART6 的基本时钟。

可由 8 位存储器操作指令设置 CKSR6。

复位信号产生将该寄存器清零(00H)。

备注 通信过程中 (当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 和第 6 位 (TXE6) 均为 1, 或 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 均为 1), 可由软件刷新 (写入相同值) CKSR6。

图 14-8. 时钟选择寄存器 6 (CKSR6) 的格式

地址: FF56H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CKSR6	0	0	0	0	TPS63	TPS62	TPS61	TPS60

TPS63	TPS62	TPS61	TPS60		基本时钟(f_{CLK6}) 的选择					
					$f_{PRS} =$ 2 MHz	$f_{PRS} =$ 5 MHz	$f_{PRS} =$ 10 MHz	$f_{PRS} =$ 20 MHz		
0	0	0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz		
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz		
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz		
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz		
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz		
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz		
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz		
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	78.13 kHz	156.25 kHz		
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	39.06 kHz	78.13 kHz		
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	19.53 kHz	39.06 kHz		
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	9.77 kHz	19.53 kHz		
1	0	1	1	TM50 输出 ^注						
其它				禁止设置						

注 选择 TM50 输出作为基本时钟时, 应注意以下几点。

- TM50 与 CR50 相等时计数时钟清零并启动的模式(TMC506 = 0)
先启动 8 位定时器/事件计数器 50, 然后允许定时器 F/F 反转操作 (TMC501 = 1)。
- PWM 模式 (TMC506 = 1)
先启动 8 位定时器/事件计数器 50, 然后设置计数时钟, 使占空比 = 50%。
不论何种模式, 都不必将 TO50 作为定时器输出引脚。

注意事项 在重写 TPS63 ~ TPS60 位时, 必须确保 POWER6 = 0。

- 备注**
1. f_{PRS} : 外围硬件时钟频率
 2. TMC506: 8 位定时器模式控制寄存器 50 (TMC50) 的第 6 位
TMC501: TMC50 的第 1 位

(5) 波特率发生器控制寄存器 6 (BRGC6)

该寄存器用于设置串行接口 UART6 的 8 位计数器的分频值。

可由 8 位存储器操作指令设置 BRGC6。

复位信号产生将该寄存器设置为 FFH。

备注 通信过程中 (当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 和第 6 位 (TXE6) 均为 1, 或 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 均为 1), 可由软件刷新 (写入相同值) BRGC6。

图 14-9. 波特率发生器控制寄存器 6 (BRGC6) 的格式

地址: FF57H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
BRGC6	MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60

MDL67	MDL66	MDL65	MDL64	MDL63	MDL62	MDL61	MDL60	k	8 位计数器输出时钟选择
0	0	0	0	0	×	×	×	×	禁止设置
0	0	0	0	1	0	0	0	8	$f_{CLK6}/8$
0	0	0	0	1	0	0	1	9	$f_{CLK6}/9$
0	0	0	0	1	0	1	0	10	$f_{CLK6}/10$
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	0	252	$f_{CLK6}/252$
1	1	1	1	1	1	0	1	253	$f_{CLK6}/253$
1	1	1	1	1	1	1	0	254	$f_{CLK6}/254$
1	1	1	1	1	1	1	1	255	$f_{CLK6}/255$

- 注意事项**
1. 在重写 MDL67 ~ MDL60 位时, 必须确保 ASIM6 的第 6 位 (TXE6) 和第 5 位 (RXE6) = 0。
 2. 波特率为 8 位计数器输出时钟的 1/2。

- 备注**
1. f_{CLK6} : 根据 CKSR6 寄存器的 TPS63 ~ TPS60 位选择基本时钟的频率
 2. k: 由 MDL67 ~ MDL60 位设置的值 (k = 8, 9, 10, ..., 255)
 3. ×: 不必考虑

(6) 异步串行接口控制寄存器 6 (ASICL6)

该寄存器用于控制串行接口 UART6 的串行通信操作。

可由 1 位或 8 位存储器操作指令设置 ASICL6。

复位信号产生将该寄存器设置为 16H。

注意事项 通信过程中(当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 和第 6 位 (TXE6) 均为 1, 或 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 均为 1), 可由软件刷新 (写入相同值) ASICL6。但是, 由于可能会重新触发 SBF 接收或 SBF 发送, 因此在 SBF 接收期间 (SBRT6 = 1) 或 SBF 发送期间 (直到由于 SBTT6=1 而产生 INTST6 为止) 通过刷新操作不要将 SBRT6 与 SBTT6 都设置为 1。

图 14-10. 异步串行接口控制寄存器 6 (ASICL6) 的格式(1/2)

地址: FF58H 复位后: 16H R/W*

符号	<7>	<6>	5	4	3	2	1	0
ASICL6	SBRF6	SBRT6	SBTT6	SBL62	SBL61	SBL60	DIR6	TXDLV6
SBRF6	SBF 接收状态标志							
0	如果 POWER6 = 0 且 RXE6 = 0, 或者如果 SBF 已正确接收							
1	SBF 接收操作正在进行中							
SBRT6	SBF 接收触发							
0	-							
1	SBF 接收触发							
SBTT6	SBF 发送触发							
0	-							
1	SBF 发送触发							

注 第 7 位是只读的。

图 15-10. 异步串行接口控制寄存器 6 (ASICL6) 的格式 (2/2)

SBL62	SBL61	SBL60	SBF 发送输出宽度控制
1	0	1	SBF 输出宽度: 13 位
1	1	0	SBF 输出宽度: 14 位
1	1	1	SBF 输出宽度: 15 位
0	0	0	SBF 输出宽度: 16 位
0	0	1	SBF 输出宽度: 17 位
0	1	0	SBF 输出宽度: 18 位
0	1	1	SBF 输出宽度: 19 位
1	0	0	SBF 输出宽度: 20 位

DIR6	首位说明
0	MSB
1	LSB

TXDLV6	允许/禁止反向 TxD6 输出
0	TxD6 正常输出
1	TxD6 反向输出

- 注意事项**
1. 如果出现 SBF 接收错误, 则返回到 SBF 接收模式, 并保持 SBRF6 标志位状态(1)不变。
 2. 在设置 SBRT6 之前, 必须确保 ASIM6 的第 7 位 (POWER6) 和第 5 位 (RXE6) 为 1。在设置 SBRT6=1 后, SBF 接收完成 (产生中断请求信号之前) 之前不要将 SBRT6 清零。
 3. SBRT6 的读取值恒为 0。在 SBF 正确接收后 SBRT6 自动清零。
 4. 在设置 SBTT6=1 之前, 必须确保 ASIM6 的第 7 位 (POWER6) 和第 6 位 (TXE6) 为 1。在设置 SBTT6=1 之后, SBF 接收完成 (产生中断请求信号之前) 之前不要将 SBTT6 清零。
 5. SBTT6 的读取值恒为 0。并在 SBF 发送结束后自动清零。
 6. 在接收期间不要设置 SBRT6=1, 而在发送期间也不要设置 SBTT6=1。
 7. 在重写 DIR6 和 TXDLV6 之前, 应对 TXE6 和 RXE6 清零。

(7) 输入切换控制寄存器 (ISC)

输入切换控制寄存器 (ISC) 在 LIN (Local Interconnect Network 本地互连网络) 接收期间用于接收从主设备发送的状态信号。

当 ISC0 与 ISC1=1 时, 选择 P14/RxD6 引脚的信号输入作为 INTP0 与 TI000 的输入源。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号产生将该寄存器清零 (00H)。

图 14-11. 输入切换控制寄存器 (ISC) 的格式

地址: FF4FH 复位后: 00H R/W

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

ISC1	TI000 输入源的选择
0	TI000 (P00)
1	RxD6 (P14)

ISC0	INTP0 输入源的选择
0	INTP0 (P120)
1	RxD6 (P14)

(8) 端口模式寄存器 1 (PM1)

该寄存器可以按位设置端口 1 的输入/输出模式。

当 P13/TxD6 引脚用于串行接口数据输出时, 将 PM13 清零且 P13 的输出锁存器的值设置为 1。

当 P14/RxD6 引脚用于串行接口数据输入时, 将 PM14 置 1。此时 P14 的输出锁存器的值= 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM1。

复位信号产生将该寄存器设置为 FFH。

图 14-12. 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 复位后: FFH R/W

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

PM1n	P1n 引脚 I/O 模式选择 (n = 0 ~ 7)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

14.4 串行接口 UART6 的操作

串行接口 UART6 有以下两种模式。

- 操作停止模式
- 异步串行接口(UART) 模式

14.4.1 操作停止模式

在此模式下，不能进行串行通信，因此，可降低功耗。此外，引脚可作为一般的端口引脚使用。通过将 ASIM6 的第 7 位、第 6 位和第 5 位 (POWER6、TXE6 和 RXE6) 清零，可设置操作停止模式。

(1) 使用的寄存器

由异步串行接口操作模式寄存器 6 (ASIM6) 设置操作停止模式。

可由 1 位或 8 位存储器操作指令设置 ASIM6。

复位信号产生将该寄存器设置为 01H。

地址: FF50H 复位后: 01H R/W

符号	<7>	<6>	<5>	4	3	2	1	0
ASIM6	POWER6	TXE6	RXE6	PS61	PS60	CL6	SL6	ISRM6
POWER6	允许/禁止内部操作时钟的操作							
0 ^{注1}	禁止内部操作时钟的操作 (时钟恒为低电平) 且对内部电路进行异步复位 ^{注2}							
TXE6	允许/禁止发送							
0	禁止发送 (同步复位发送电路)							
RXE6	允许/禁止接收							
0	禁止接收 (同步复位接收电路)							

- 注
1. 发送期间当 POWER6 = 0 时，TxD6 的输出变成高电平而来自 RxD6 引脚的输入恒为高电平。
 2. 对异步串行接口接收错误状态寄存器 6 (ASIS6)、异步串行接口发送状态寄存器 6 (ASIF6)、异步串行接口控制寄存器 (ASICL6) 的第 7 位 (SBRF6) 和第 6 位 (SBRT6) 和接收缓冲寄存器 6 (RXB6) 进行复位。

注意事项 TXE6 和 RXE6 清零后再对 POWER6 清零，可设置操作停止模式。

如果要启动，需先设置 POWER6=1，然后设置 TXE6=1 或 RXE6=1。

备注 如果将 RxD6/P14 和 TxD6/P13 作为通用端口引脚使用，可参见 第四章 端口功能。

14.4.2 异步串行接口 (UART) 模式

此模式下，在传送了起始位后，紧接着发送/接收 1 字节数据，然后可执行全双工操作。
可使用专用 UART 波特率发生器，这样通信选择的波特率范围较大。

(1) 使用的寄存器

- 异步串行接口操作模式寄存器 6 (ASIM6)
- 异步串行接口接收错误状态寄存器 6 (ASIS6)
- 异步串行接口发送状态寄存器 6 (ASIF6)
- 时钟选择寄存器 6 (CKSR6)
- 波特率发生器控制寄存器 6 (BRGC6)
- 异步串行接口控制寄存器 6 (ASICL6)
- 输入切换控制寄存器 (ISC)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

在 UART 模式中设置操作的基本过程如下。

- <1> 设置 CKSR6 寄存器 (参见图 14-8)
- <2> 设置 BRGC6 寄存器 (参见图 14-9)
- <3> 设置 ASIM6 的第 0 位~第 4 位 (ISRM6、SL6、CL6、PS60、PS61) (参见图 14-5)
- <4> 设置 ASICL6 的第 0 位和第 1 位 (TXDLV6、DIR6) (参见图 14-10)
- <5> 设置 ASIM6 的第 7 位 (POWER6) = 1
- <6> 设置 ASIM6 的第 6 位 (TXE6) = 1。 → 允许发送
设置 ASIM6 的第 5 位 (RXE6) = 1。 → 允许接收
- <7> 将数据写入发送缓冲寄存器 6 (TXB6) → 开始发送数据

注意事项 在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。

寄存器设置与引脚之间的关系如下所示。

表 14-2. 寄存器设置与引脚之间的关系

POWER6	TXE6	RXE6	PM13	P13	PM14	P14	UART6 操作	引脚功能	
								TxD6/P13	RxD6/P14
0	0	0	x [#]	x [#]	x [#]	x [#]	停止	P13	P14
1	0	1	x [#]	x [#]	1	x	接收	P13	RxD6
	1	0	0	1	x [#]	x [#]	发送	TxD6	P14
	1	1	0	1	1	x	发送/ 接收	TxD6	RxD6

注 可以用作端口寄存器。

- 备注**
- x: 不必考虑
 - POWER6: 异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位
 - TXE6: ASIM6 的第 6 位
 - RXE6: ASIM6 的第 5 位
 - PM1x: 端口模式寄存器
 - P1x: 端口输出锁存器

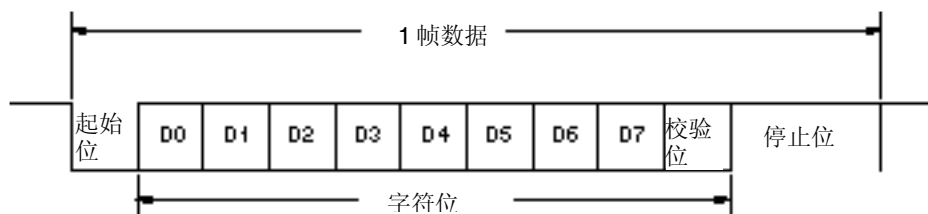
(2) 通信操作

(a) 正常发送/接收数据的格式和波形示例

图 14-13 和 14-14 为正常发送/接收数据的格式和波形示例。

图 14-13. 正常 UART 发送/接收数据的格式

1. LSB-first 发送/接收



2. MSB-first 发送/接收



一帧数据由以下各位组成。

- 起始位 ... 1 位
- 字符位 ... 7 或 8 位
- 校验位 ... 偶校验、奇校验、零校验或无校验
- 停止位 ... 1 或 2 位

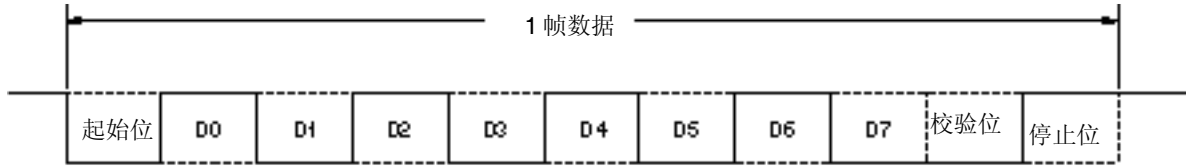
由异步串行接口操作模式寄存器 6 (ASIM6) 规定一个数据帧中字符位的宽度、校验方式和停止位的宽度。

由异步串行接口控制寄存器 6 (ASICL6) 的第 1 位 (DIR6) 规定数据通信方式 (LSB-first 或 MSB-first)。

由 ASICL6 的第 0 位 (TXDLV6) 规定 TxD6 引脚输出的是正常数据还是反向数据。

图 14-14. 正常 UART 发送/接收数据波形示例

1. 数据宽度: 8 位、LSB-first, 校验位: 偶校验, 停止位: 1 位, 通信数据: 55H



2. 数据宽度: 8 位、MSB-first, 校验位: 偶校验, 停止位: 1 位, 通信数据: 55H



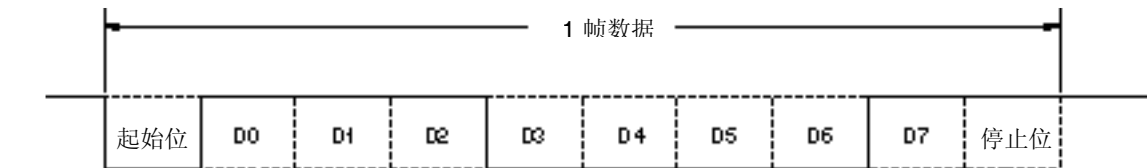
3. 数据宽度: 8 位、MSB-first, 校验位: 偶校验, 停止位: 1 位, 通信数据: 55H, Tx/D6 引脚输出反向数据



4. 数据宽度: 7 位、LSB-first, 校验位: 奇校验, 停止位: 2 位, 通信数据: 36H



5. 数据宽度: 8 位、LSB-first, 校验位: 无, 停止位: 1 位, 通信数据: 87H



(b) 校验方式与操作

校验位用于检测数据通信中的位错误。通常在发送和接收两端可采用相同的校验方式。当采用偶校验和奇校验时，可检测到 1 位(奇数位)错误。而采用零校验和无校验时，则不能检测到错误。

注意事项 在 LIN 模式下，PS61 与 PS60 恒为 0。

(i) 偶校验

• 发送

控制发送数据，包括校验位，使得数据中“1”的个数为偶数。

校验位的取值如下。

如果发送数据有奇数个“1”：1

如果发送数据有偶数个“1”：0

• 接收

计算接收数据(包括校验位)中“1”的个数。如果“1”的个数为奇数，则产生校验错。

(ii) 奇校验

• 发送

控制发送数据，包括校验位，使得数据中“1”的个数为奇数。

如果发送数据有奇数个“1”：0

如果发送数据有偶数个“1”：1

• 接收

计算接收数据(包括校验位)中“1”的个数。如果“1”的个数为偶数，则产生校验错。

(iii) 零校验

无论发送数据为何值，发送时校验位清零。

接收数据时不检测校验位。因此无论校验位为何值，都不会产生校验错。

(iv) 无校验

发送数据中没有校验位。

进行接收操作时认为接收数据中没有校验位。因此也不会产生校验错。

(c) 正常发送

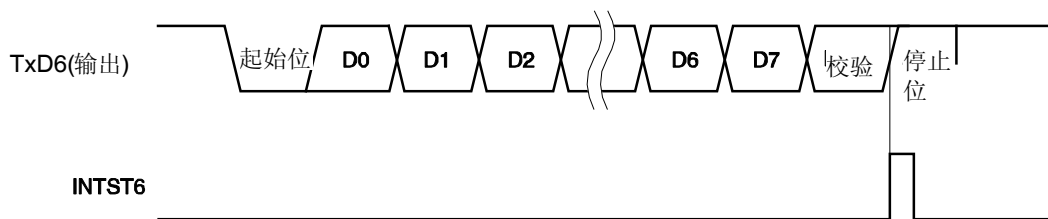
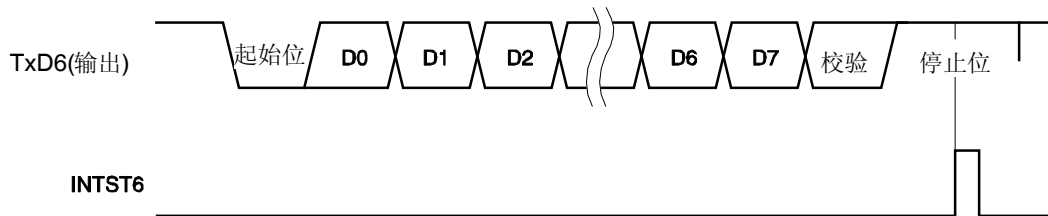
在设置异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6)=1 然后设置 ASIM6 的第 6 位 (TXE6)=1 时, 则允许发送。将待发送的数据写入发送缓冲寄存器 6 (TXB6) 中, 可启动发送操作。起始位、校验位和停止位会自动被添加到数据中。

启动发送操作时, TXB6 中的数据被传送到发送移位寄存器 6 (TXS6) 中, 然后数据依次从 TXS6 被输出到 TxD6 引脚。当发送结束时, 由 ASIM6 设置的校验位和停止位被添加到数据中, 并产生发送完成中断请求 (INTST6)。

当下一个发送数据写入 TXB6 时, 发送操作才结束。

图 14-15 显示了发送完成中断请求 (INTST6) 的时序。最后一个停止位一输出就产生该中断。

图 14-15. 正常的发送完成中断请求时序

1. 停止位宽度: 1**2. 停止位宽度: 2**

(d) 连续发送

当发送移位寄存器 6 (TXS6) 已经启动移位操作，就可以把下一个发送数据写入发送缓冲寄存器 6 (TXB6) 中。因此，当发送完一帧数据后，即使在执行 INTST6 中断服务时也能够连续发送数据，从而实现高效率的通信速率。此外，当产生发送完成中断时，通过读取异步串行接口发送状态寄存器 ASIF6 的第 0 位 (TXSF6)，可以对 TXB6 进行两次有效的写操作 (2 个字节) 而无须等待发送一帧数据的时间。

为了能够连续发送数据，必须根据 ASIF6 寄存器检查发送状态，判断是否可以对 TXB6 进行写操作，如果可以，则写入数据。

- 注意事项**
1. 在连续发送期间，ASIF6 寄存器的 TXBF6 和 TXSF6 标志从“10”变成“11”，再变成“01”。因此检查此状态时，不能通过 TXBF6 和 TXST6 来判断。在连续发送期间只读 TXBF6 标志。
 2. 在 LIN 模式下，不能使用连续发送功能。在将发送数据写入发送缓冲寄存器 6 (TXB6) 之前，异步串行接口发送状态寄存器 6 (ASIF6) 的值必须为 00H。

TXBF6	对 TXB6 的写操作
0	允许写操作
1	禁止写操作

注意事项 为了能够连续发送数据，应将第一个发送的数据 (第一个字节) 写入 TXB6 中。之后，检查 TXBF6 标志位，必须为“0”。如果正确，将下一个发送数据 (第二个字节) 写入 TXB6 中。如果 TXBF6 标志为“1”时将数据写入 TXB6，则无法保证发送数据的正确性。

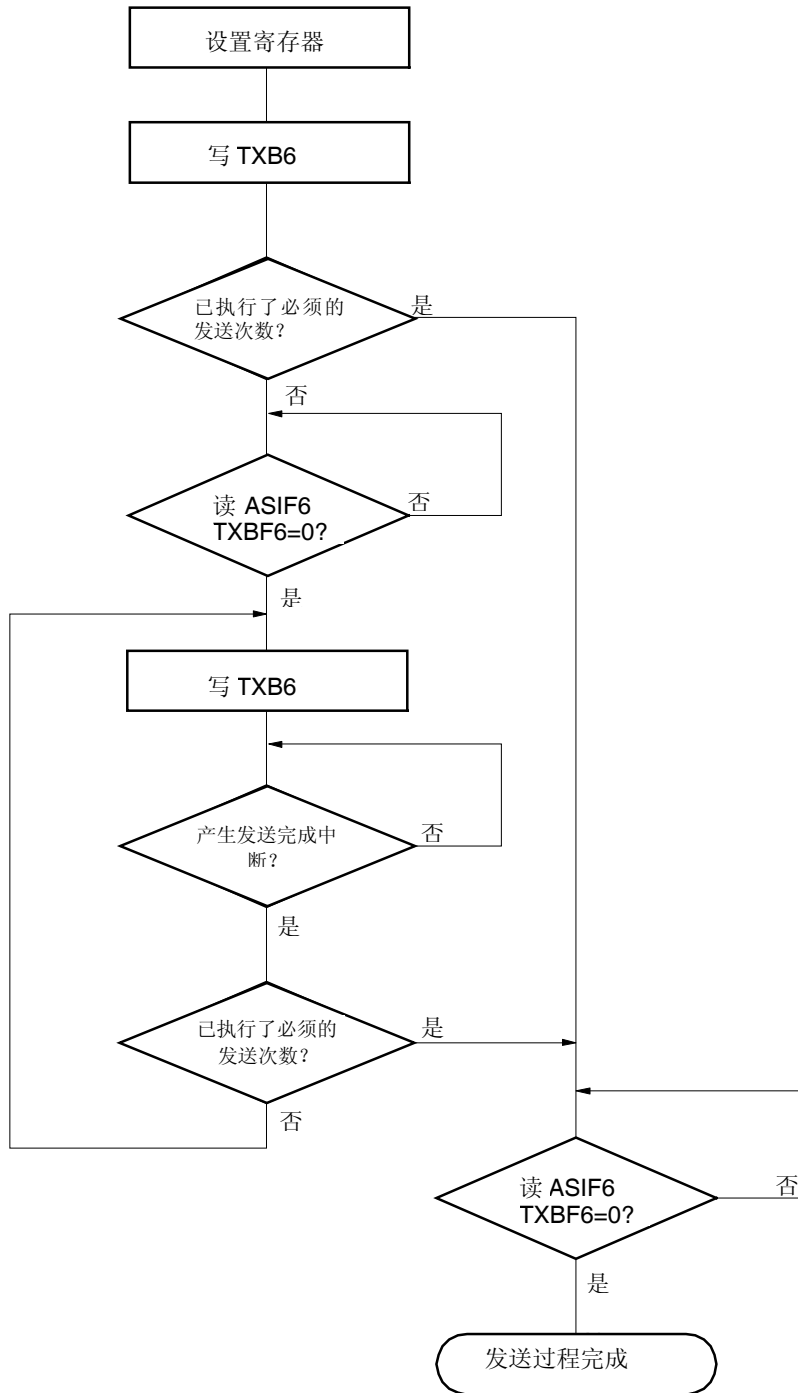
可用 TXSF6 标志位检测通信状态。

TXSF6	发送状态
0	发送操作结束
1	发送操作正在进行中

- 注意事项**
1. 如果要在连续发送结束时初始化发送单元，则在产生发送完成中断后必须检查 TXSF6 标志位，应为“0”。然后执行初始化操作。如果 TXSF6 标志为“1”时执行初始化操作，则无法保证发送数据的正确性。
 2. 在连续发送期间，当发送完一帧数据后并在执行 INTST6 中断服务之前下一个发送操作可能已经完成。计数测量时，可通过开发一个可以计算发送数据的个数的程序或者通过参考 TXSF6 标志来实现检测。

图 14-16 为连续发送操作流程示例。

图 14-16. 连续发送操作流程示例

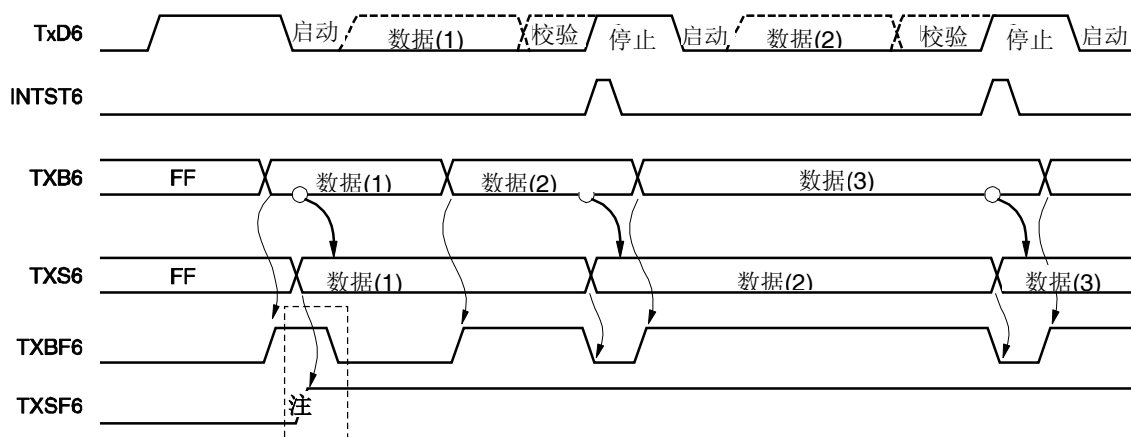


备注

- TXB6: 发送缓冲寄存器 6
- ASIF6: 异步串行接口发送状态寄存器 6
- TXBF6: ASIF6 的第 1 位(发送缓冲器数据标志位)
- TXSF6: ASIF6 的第 0 位(发送移位寄存器数据标志位)

图 14-17 显示了启动连续发送操作的时序，而图 14-18 显示了结束连续发送操作的时序。

图 14-17. 启动连续发送操作的时序

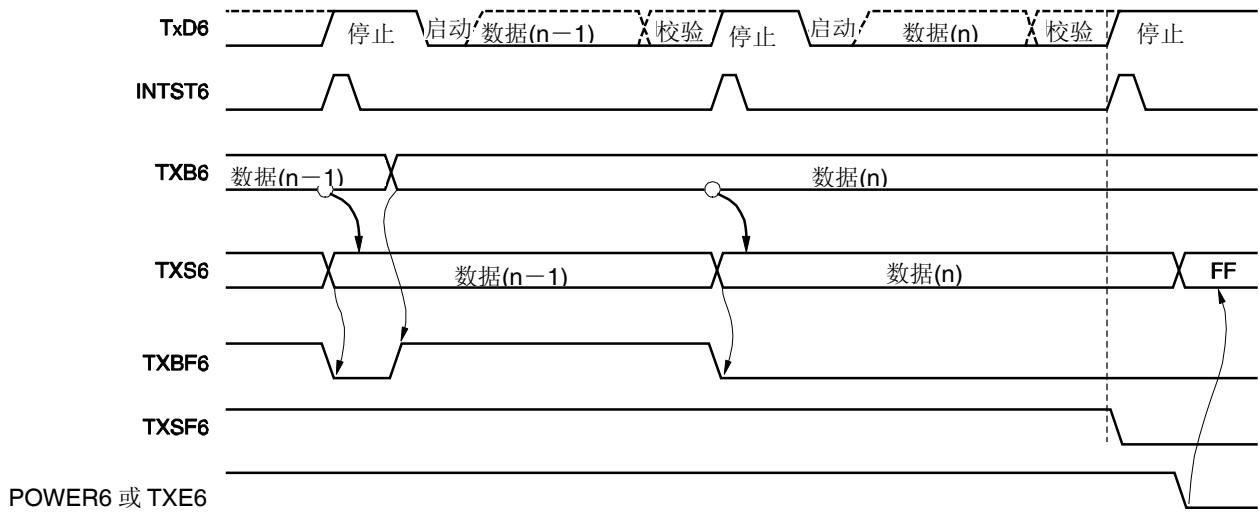


注 在读取 ASIF6 时，存在 TXBF6 和 TXSF6 = 1, 1 的情况。因此，仅用 TXBF6 就可以判断是否能进行写操作。

备注

- TxD6: TxD6 引脚 (输出)
- INTST6: 中断请求信号
- TXB6: 发送缓冲寄存器 6
- TXS6: 发送移位寄存器 6
- ASIF6: 异步串行接口发送状态寄存器 6
- TXBF6: ASIF6 的第 1 位
- TXSF6: ASIF6 的第 0 位

图 14-18. 结束连续发送操作的时序



- 备注
- TxD6: TxD6 引脚 (输出)
 - INTST6: 中断请求信号
 - TXB6: 发送缓冲寄存器 6
 - TXS6: 发送移位寄存器 6
 - ASIF6: 异步串行接口发送状态寄存器 6
 - TXBF6: ASIF6 的第 1 位
 - TXSF6: ASIF6 的第 0 位
 - POWER6: 异步串行接口操作模式寄存器 (ASIM6) 的第 7 位
 - TXE6: 异步串行接口操作模式寄存器 (ASIM6) 的第 6 位

(e) 正常接收

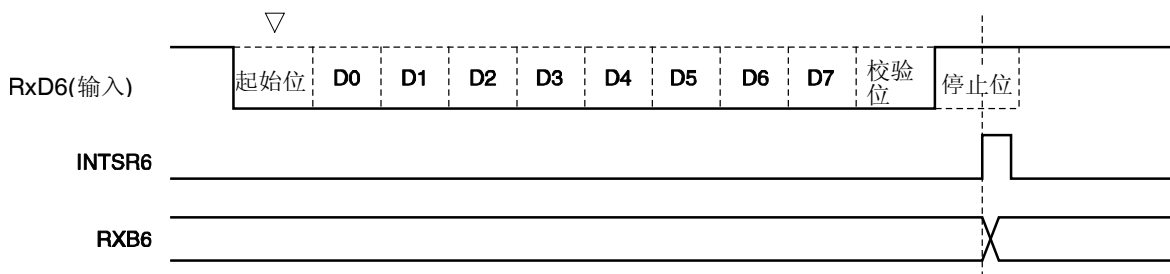
当设置异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6)=1, 然后将 ASIM6 的第 5 位 (RXE6) 置 1 时, 允许接收并对 RxD6 引脚的输入进行采样。

当检测到 RxD6 引脚输入的下降沿时, 波特率发生器的 8 位计数器开始计数。当计数值达到波特率发生器控制寄存器 6 (BRGC6) 的设置值时, 再次采样 RxD6 的输入信号(如图 14-19 中 ▽)。如果此时 RxD6 为低电平, 则认为输入信号为起始位。

当检测到起始位时, 开始接收, 以设置的波特率将串行数据依次存入接收移位寄存器 6 (RXS6) 中。当接收到停止位时, 会产生接收完成中断信号 (INTSR6), 并且 RXS6 的数据被写入接收缓冲寄存器 6 (RXB6) 中。但如果出现溢出错误 (OVE6), 则接收到的数据不写入 RXB6。

即使在接收过程中出现校验错 (PE6), 接收仍然继续, 直至接收到停止位, 而在接收完成后会产生一个接收错误中断 (INTSR6/INTSRE6)。

图 14-19. 接收完成中断请求时序



- 注意事项**
1. 如果出现接收错误, 则先后读取 ASIS6 和 RXB6, 可将出错标志清零。否则, 当接收到下一个数据时会产生溢出错误, 而且接收错误状态保持不变。
 2. 接收始终按“停止位的个数 = 1”的情况执行, 第 2 个停止位被忽略。
 3. 在读取 RXB6 之前, 必须先读取异步串行接口接收错误状态寄存器 6 (ASIS6)。

(f) 接收错误

在接收期间有三种出错情况：校验错误、帧错误或溢出错误。作为数据接收的结果，如果异步串行接口接收错误状态寄存器 6 (ASIS6) 的出错标志位被置位，则将会产生一个接收错误中断请求信号 (INTSR6/INTSRE6)。

在执行接收错误中断服务 (INTSR6/INTSRE6) 时读取 ASIS6 的内容，可确定接收期间出现的是何种错误 (参见图 14-6)。

读取 ASIS6 后，其内容清零(0)。

表 14-3. 产生接收错误的原因

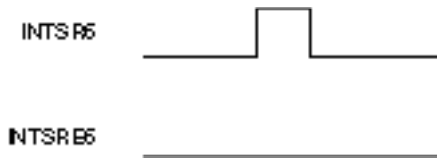
接收错误	产生原因
校验错误	发送数据的校验位与接收到的数据校验位不相等
帧错误	未检测到停止位
溢出错误	在读取接收缓冲寄存器 6 (RXB6) 的数据前已完成下一个数据的接收操作

通过将异步串行接口操作模式寄存器 6 (ASIM6) 的第 0 位 (ISRM6) 清零，可以把接收错误中断划分为接收完成中断 (INTSR6) 和错误中断 (INTSRE6) 两部分。

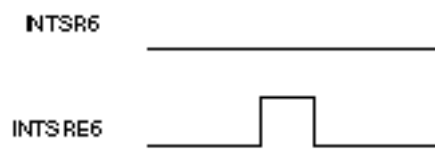
图 14-20. 接收错误中断

1. 如果 ISRM6 被清零(接收完成中断 (INTSR6) 和接收错误中断 (INTSRE6) 是分开的)

(a) 接收期间无错误

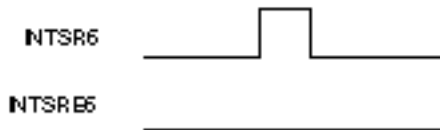


(b) 接收期间产生错误

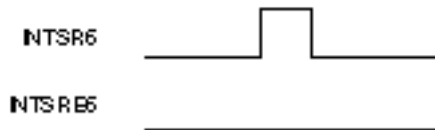


2. 如果 ISRM6=1 (接收错误中断包含在 INTSR6 中)

(a) 接收期间无错误



(b) 接收期间产生错误



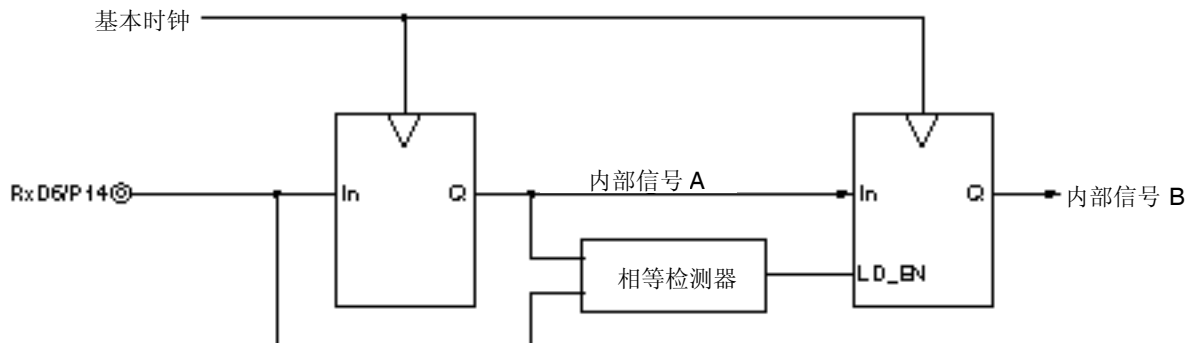
(g) 接收数据的噪声过滤器

使用由预分频模块输出的基准时钟采样 **RxD6** 信号。

如果两次采样值相同，则相等检测器的输出会发生变化，并把采样的数据作为输入数据。

噪声过滤器的电路结构如图 14-21 所示，接收操作的内部过程从外部信号状态开始延迟两个时钟才执行。

图 14-21. 噪声过滤器电路

**(h) SBF 发送**

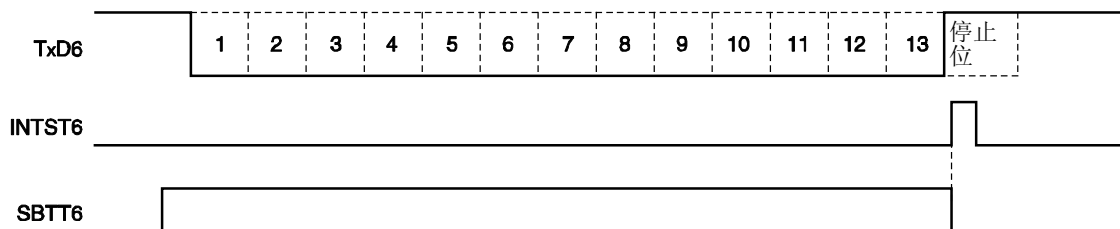
在 LIN 模式中，可使用 **SBF** (同步中断区) 发送控制功能进行发送。LIN 的发送操作参见图 14-1 LIN 的发送操作。

当异步串行接口模式寄存器 6 (**ASIM6**) 的第 7 位 (**POWER6**) 被设置为 1 时，**TxD6** 引脚输出高电平。接下来 **ASIM6** 的第 6 位 (**TXE6**) 被设置为 1 时，进入允许发送状态。且通过设置异步串行接口控制寄存器 6 (**ASICL6**) 的第 5 位 (**SBTT6**) 为 1，可启动 **SBF** 发送操作。

开始发送后，第 13 位~第 20 位 (由 **ASICL6** 的第 4 位~第 2 位(**SBL62** ~ **SBL60**) 设置) 输出低电平。**SBF** 发送完毕后，产生发送完成中断请求 (**INTST6**)，且 **SBTT6** 被自动清零。接着，恢复到正常发送模式。

直到下一次数据被发送到缓冲寄存器 6 (**TXB6**) 或 **SBTT6** 被设置为 1 时，才停止 **SBF** 发送。

图 14-22. SBF 发送



备注 TxD6: TxD6 引脚 (输出)
INTST6: 发送完成中断请求
SBTT6: 异步串行接口控制寄存器 6 (**ASICL6**) 的第 5 位

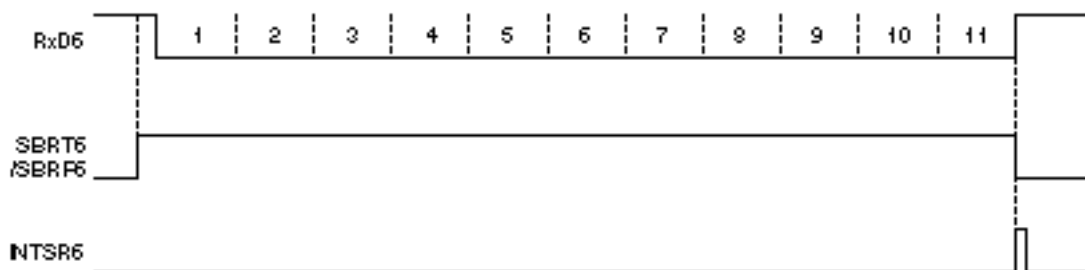
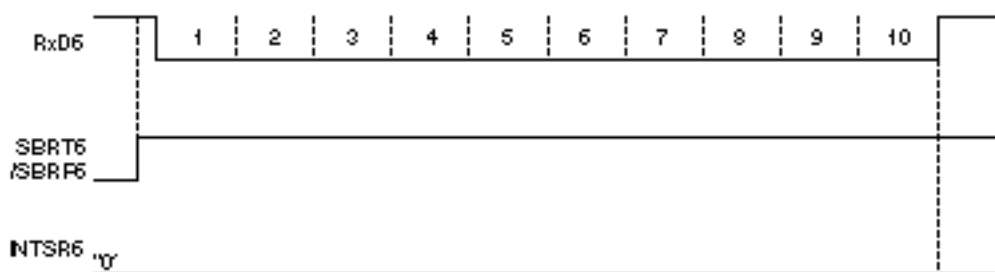
(i) SBF 接收

在 LIN 模式下，SBF (同步中断区) 接收控制功能用于接收操作。需要了解 LIN 的接收操作，可参见图 14-2 LIN 接收操作。

将异步串行接口模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 设置为 1，然后再将 ASIM6 的第 5 位 (RXE6) 设置为 1，允许接收。当异步串行接口控制寄存器 6 (ASICL6) 的第 6 位 (SBRT6) 被设置为 1 时，允许 SBF 接收。在 SBF 接收允许状态下，对 RxD6 引脚进行采样，并以与正常接收允许状态下同样的方式检测起始位。

当检测到起始位时，启动接收操作，按照设置的波特率将串行数据依次存入接收移位寄存器 6 (RXS6) 中。在接收到停止位且 SBF 的宽度大于或等于 11 位时，与正常接收操作相同，将产生接收完成中断请求 (INTSR6)。此时 SBRF6 和 SBRT6 被自动清零，SBF 接收操作结束。禁止检测 OVE6、PE6 和 FE6 (异步串行接口接收错误状态寄存器 6 (ASIS6) 的第 0 位 ~ 第 2 位) 错误，不进行 UART 通信的错误检测。此外不执行接收移位寄存器 6 (RXS6) 和接收缓冲寄存器 6 (RXB6) 之间的数据传送，并保存复位值 FFH。如果 SBF 的宽度小于或等于 10 位，则在接收到停止位后不会产生出错中断，并恢复 SBF 接收模式。此时不对 SBRF6 和 SBRT6 位清零。

图 14-23. SBF 接收

1. 正常 SBF 接收 (检测到停止位，且 SBF 宽度大于 10.5 位)**2. SBF 接收错误 (检测到停止位，且 SBF 宽度小于或等于 10.5 位)**

备注

- RxD6: RxD6 引脚 (输入)
- SBRT6: 异步串行接口控制寄存器 6 (ASICL6) 的第 6 位
- SBRF6: ASICL6 的第 7 位
- INTSR6: 接收完成中断请求

14.4.3 专用波特率发生器

专用波特率发生器由一个源时钟选择器和一个 8 位可编程计数器组成，用于产生 UART6 发送/接收的串行时钟。分别使用不同的 8 位计数器用于发送和接收。

(1) 波特率发生器的配置

- 基本时钟

该时钟由时钟选择寄存器 6 (CKSR6) 的第 3 位 ~ 第 0 位 (TPS63 ~ TPS60) 选择。当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6)=1 时，将该时钟提供给每个模块。该时钟称为基本时钟，它的频率用 f_{CLK6} 表示。当 POWER6 = 0 时，基本时钟恒为低电平。

- 发送计数器

当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 或第 6 位 (TXE6) = 0 时，该计数器停止计数，并被清零。

当 POWER6 = 1 且 TXE6 = 1 时计数器开始计数。

当第 1 个发送数据写入到发送缓冲寄存器 6 (TXB6) 时，计数器被清零。

如果连续发送数据，则当发送完 1 帧数据时计数器再次被清零。如果后面没有数据要发送，则计数器不被清零，而是继续计数直至 POWER6 或 TXE6 = 0。

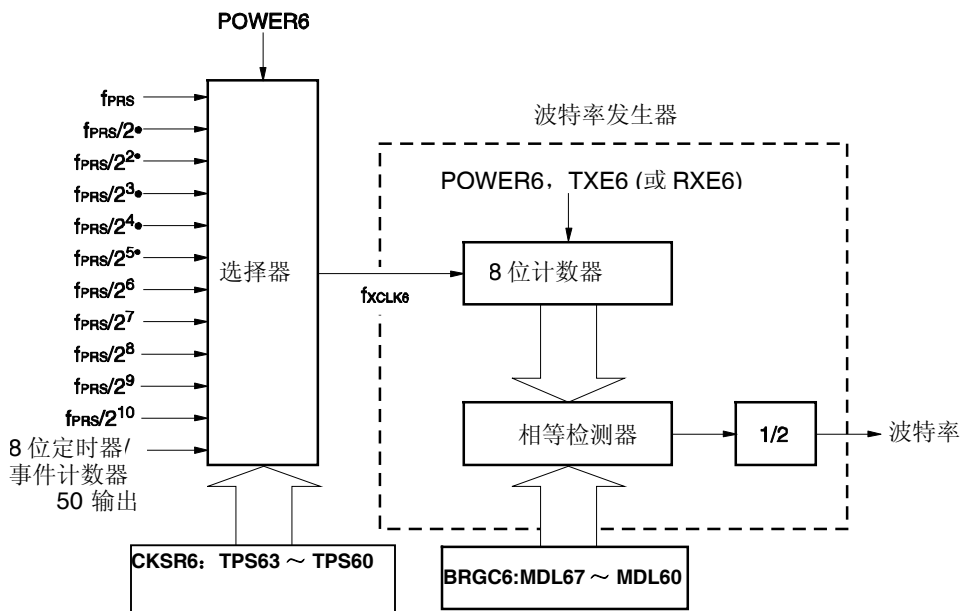
- 接收计数器

当异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位 (POWER6) 或第 5 位 (RXE6)= 0 时，该计数器停止计数，并被清零。

当检测到起始位时，计数器开始计数。

在收到一帧数据后计数器停止计数，直到下一个起始位被检测到时再重新开始计数。

图 14-24. 波特率发生器的配置



- 备注
- POWER6: 异步串行接口操作模式寄存器 6 (ASIM6) 的第 7 位
 - TXE6: ASIM6 的第 6 位
 - RXE6: ASIM6 的第 5 位
 - CKSR6: 时钟选择寄存器 6
 - BRGC6: 波特率发生器控制寄存器 6

(2) 串行时钟的产生

使用时钟选择寄存器 6 (CKSR6) 和波特率发生器控制寄存器 6 (BRGC6) 可产生串行时钟。

使用 CKSR6 的第 3 位 ~ 第 0 位 (TPS63 ~ TPS60)，可以设置输入到 8 位计数器的时钟。使用 BRGC6 的第 7 位 ~ 第 0 位 (MDL67 ~ MDL60) 设置 8 位计数器的分频值 ($f_{XCLK6}/8 \sim f_{XCLK6}/255$)。

表 14-4. TPS63 ~ TPS60 的设定值

TPS63	TPS62	TPS61	TPS60		基本时钟 (f_{XCLK6}) 选择			
					$f_{PRS} = 2 \text{ MHz}$	$f_{PRS} = 5 \text{ MHz}$	$f_{PRS} = 10 \text{ MHz}$	$f_{PRS} = 20 \text{ MHz}$
0	0	0	0	f_{PRS}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	78.13 kHz	156.25 kHz
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	39.06 kHz	78.13 kHz
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	19.53 kHz	39.06 kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	9.77 kHz	19.53 kHz
1	0	1	1		TM50 输出			
其它					禁止设置			

(a) 波特率

通过下列公式计算波特率。

$$\bullet \text{ 波特率} = \frac{f_{\text{CLK6}}}{2 \times k} \text{ [bps]}$$

f_{CLK6} : 根据 CKSR6 寄存器的 TPS63 ~ TPS60 位选择的基本时钟的频率

k: 根据 BRGC6 寄存器的 MDL67 ~ MDL60 位设置的值 ($k = 8, 9, 10, \dots, 255$)

(2) 波特率误差

通过下列公式计算波特率误差。

$$\bullet \text{ 误差 (\%)} = \left(\frac{\text{实际波特率 (有误差的波特率)}}{\text{预期波特率 (正确的波特率)}} \right) - 1 \times 100 \text{ [\%]}$$

注意事项 1. 发送期间必须保持波特率误差在接收端允许的误差范围内。

2. 接收期间，波特率误差必须满足“(4)接收期间允许的波特率范围”中所描述的范围。

示例: 基本时钟频率 = 10 MHz = 10, 000, 000 Hz
BRGC6 寄存器的 MDL67 ~ MDL60 位 = 00100001B ($k = 33$)
目标波特率 = 153600 bps

$$\begin{aligned} \text{波特率} &= 10 \text{ M} / (2 \times 33) \\ &= 10000000 / (2 \times 33) = 151,515 \text{ [bps]} \end{aligned}$$

$$\begin{aligned} \text{误差} &= (151515/153600 - 1) \times 100 \\ &= -1.357 \text{ [\%]} \end{aligned}$$

(3) 波特率设置示例

表 14-5. 波特率发生器的设置数据

波特率 [bps]	f _{PRS} = 2.0 MHz				f _{PRS} = 5.0 MHz				f _{PRS} = 10.0 MHz				f _{PRS} = 20.0 MHz			
	TPS63- TPS60	k	计算值	ERR [%]	TPS63- TPS60	k	计算值	ERR [%]	TPS63- TPS60	k	计算值	ERR [%]	TPS63- TPS60	k	计算值	ERR [%]
300	8H	13	301	0.16	7H	65	301	0.16	8H	65	301	0.16	9H	65	301	0.16
600	7H	13	601	0.16	6H	65	601	0.16	7H	65	601	0.16	8H	65	601	0.16
1200	6H	13	1202	0.16	5H	65	1202	0.16	6H	65	1202	0.16	7H	65	1202	0.16
2400	5H	13	2404	0.16	4H	65	2404	0.16	5H	65	2404	0.16	6H	65	2404	0.16
4800	4H	13	4808	0.16	3H	65	4808	0.16	4H	65	4808	0.16	5H	65	4808	0.16
9600	3H	13	9615	0.16	2H	65	9615	0.16	3H	65	9615	0.16	4H	65	9615	0.16
19200	2H	13	19231	0.16	1H	65	19231	0.16	2H	65	19231	0.16	3H	65	19231	0.16
24000	1H	21	23810	-0.79	3H	13	24038	0.16	4H	13	24038	0.16	5H	13	24038	0.16
31250	1H	4	31250	0	4H	5	31250	0	5H	5	31250	0	6H	5	31250	0
38400	1H	13	38462	0.16	0H	65	38462	0.16	1H	65	38462	0.16	2H	65	38462	0.16
48000	0H	21	47619	-0.79	2H	13	48077	0.16	3H	13	48077	0.16	4H	13	48077	0.16
76800	0H	13	76923	0.16	0H	33	75758	-1.36	0H	65	76923	0.16	1H	65	76923	0.16
115200	0H	9	111111	-3.55	1H	11	113636	-1.36	0H	43	116279	0.94	0H	87	114943	-0.22
153600	-	-	-	-	1H	8	156250	1.73	0H	33	151515	-1.36	1H	33	151515	-1.36
312500	-	-	-	-	0H	8	312500	0	1H	8	312500	0	2H	8	312500	0

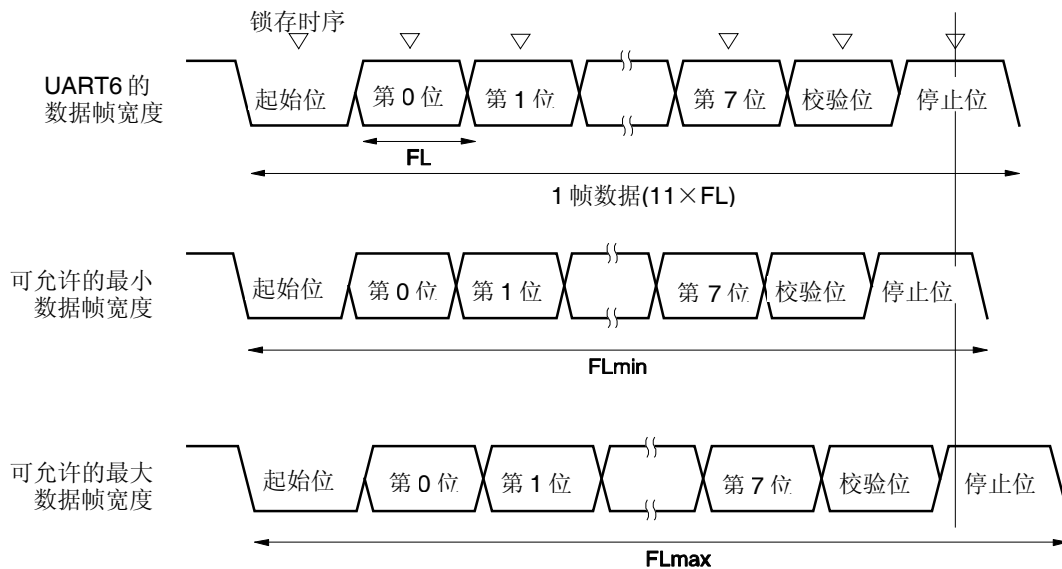
备注 TPS63 ~ TPS60: 时钟选择寄存器 6 (CKSR6) 的第 3 位 ~ 第 0 位 (用于设置基本时钟(f_{XCLK6}))
k: 使用波特率发生器控制寄存器 6 (BRGC6) 的 MDL67 ~ MDL60 位设置的值
 (k = 8, 9, 10, ..., 255)
f_{PRS}: 外围硬件时钟频率
ERR: 波特率误差

(4) 接收期间允许的波特率范围

接收期间来自发送端的波特率可允许的误差如下所示。

注意事项 必须确保接收期间波特率误差在允许的误差范围内，可用以下公式计算。

图 14-25. 接收期间允许的波特率范围



如图 14-25 所示，当检测到起始位后，接收数据的锁存时序由波特率发生器控制寄存器 6 (BRGC6) 设置的计数器确定。如果数据的最后一位(停止位)满足该锁存时序，则数据可被正确接收。

假定接收数据为 11 位，各项理论值计算如下。

$$FL = (\text{Brate})^{-1}$$

Brate: UART6 的波特率
 k: BRGC6 的设置值
 FL: 1 位数据宽度
 锁存时序的极限: 2 个时钟

$$\text{可允许的最小数据帧宽度: } FL_{\min} = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

因此在发送端设定的可接收的最大波特率如下所示。

$$BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k+2} \text{ Brate}$$

同样，可计算允许的最大数据帧宽度。

$$\frac{10}{11} \times FL_{\max} = 11 \times FL - \frac{k+2}{2 \times k} \times FL = \frac{21k-2}{2 \times k} FL$$

$$FL_{\max} = \frac{21k-2}{20k} FL \times 11$$

因此在发送端设定的可接收的最小波特率如下所示。

$$BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k-2} \text{ Brate}$$

可通过上述最小和最大波特率公式计算在 UART6 与发送端之间允许的波特率误差，如下所示。

表 14-6. 允许的最大/最小波特率误差

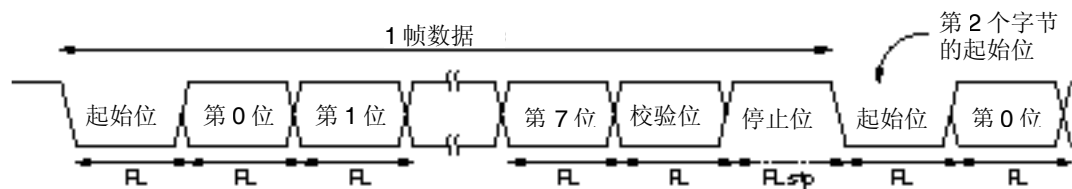
分频比(k)	允许的最大波特率误差	允许的最小波特率误差
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

- 备注**
1. 接收时允许的误差与每帧的位数、输入时钟频率和分频比(k)有关。输入时钟频率和分频比(k)越高，允许的误差就越大。
 2. k: BRGC6 设置的值。

(5) 连续发送期间数据帧的宽度

当连续发送数据时，从停止位到下一个起始位的数据帧宽度比正常值延长了两个基本时钟。不过，由于在检测到起始位时接收端的时序被初始化，因此通信结果不会受到影响。

图 14-26. 连续发送期间数据帧的宽度



当 1 位数据宽度为 FL 、停止位宽度为 FL_{stp} ，基本时钟频率为 f_{xCLK6} 时，有以下公式成立。

$$FL_{stp} = FL + 2/f_{xCLK6}$$

因此，连续发送期间的数据帧宽度为：

$$\text{数据帧宽度} = 11 \times FL + 2/f_{xCLK6}$$

第十五章 串行接口 CSI10

15.1 串行接口 CSI10 的功能

串行接口 CSI10 有以下两种模式。

- 操作停止模式
- 3 线串行 I/O 模式

(1) 操作停止模式

在不进行串行通信时可采用此模式，以降低功率消耗。

需要了解详细信息，可参见 15.4.1 操作停止模式。

(2) 3 线串行 I/O 模式（可选择 MSB-first / LSB-first）

该模式采用三条线（一条串行时钟线（ $\overline{\text{SCK10}}$ ）和两条串行数据线（SI10 和 SO10））传输 8 位数据。

在 3 线串行 I/O 模式中数据通信时间可以缩短，因为发送和接收操作可以同步进行。

此外，可以指定采用 MSB-first 还是 LSB-first 传送 8 位数据，因此该接口可以连接任何设备。

3 线串行 I/O 模式用来连接采用时钟串行接口的外部 IC 和显示控制器。

需要了解详细信息，可参见 15.4.2 3 线串行 I/O 模式。

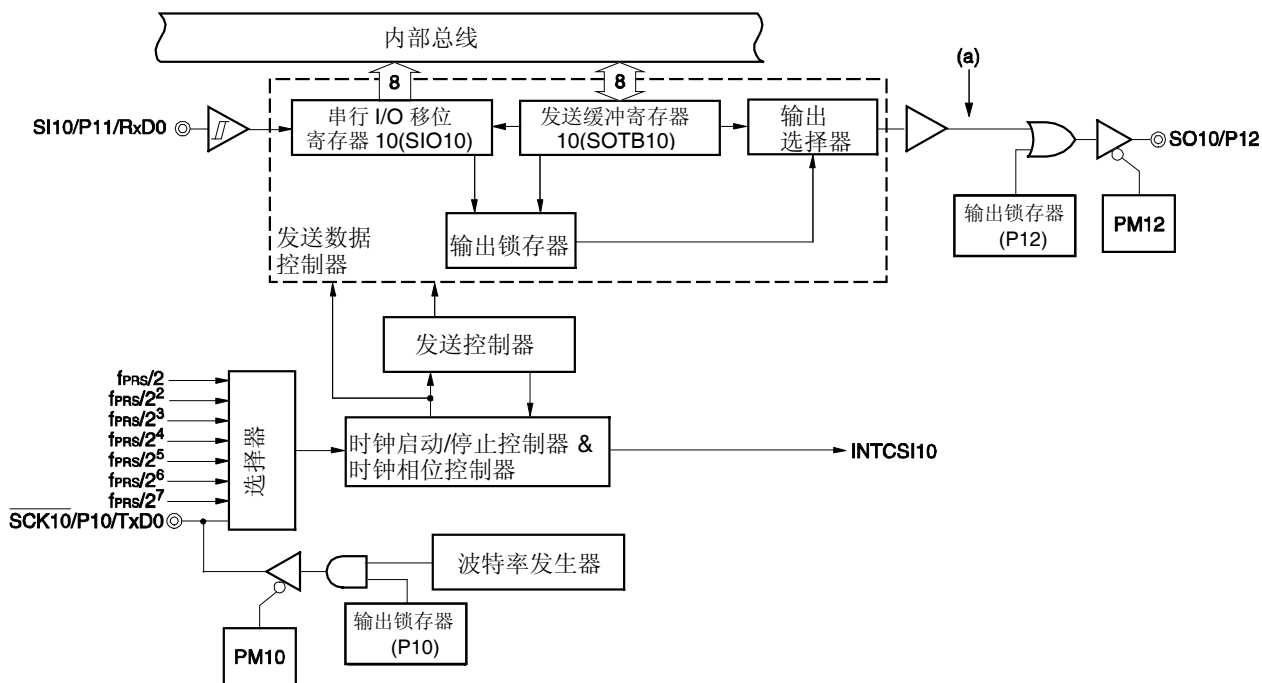
15.2 串行接口 CSI10 的配置

串行接口 CSI10 包括以下硬件。

表 15-1. 串行接口 CSI10 的配置

项目	配置
控制器	发送控制器 时钟启动/停止控制器 & 时钟相位控制器
寄存器	发送缓冲寄存器 10（SOTB10） 串行 I/O 移位寄存器 10（SIO10）
控制寄存器	串行操作模式寄存器 10（CSIM10） 串行时钟选择寄存器 10（CSIC10） 端口模式寄存器 1（PM1） 端口寄存器 1（P1）

图 15-1. 串行接口 CSI10 的框图



备注 (a): SO10 输出

(1) 发送缓冲寄存器 10 (SOTB10)

该寄存器用于设置发送数据。

当串行操作模式寄存器 10 (CSIM10) 的第 7 位 (CSIE10) 和第 6 位 (TRMD10) 均为 1 时, 通过将数据写入 SOTB10 来启动发送/接收操作。

使用串行 I/O 移位寄存器 10 将写入 SOTB10 的并行数据转换成串行数据, 并把数据输出到串行输出引脚 (SO10)。

可由 8 位存储器操作指令对 SOTB10 进行读写。

复位信号的产生将该寄存器清零 (00H)。

注意事项 当 CSOT10 = 1 (在串行通信期间) 时不要访问 SOTB10。

(2) 串行 I/O 移位寄存器 10 (SIO10)

该 8 位寄存器可将并行数据转换成串行数据, 反之亦然。

可由 8 位存储器操作指令对寄存器进行读操作。

如果串行操作模式寄存器 10 (CSIM10) 的第 6 位 (TRMD10) = 0, 则从 SIO10 读取数据可启动接收操作。

接收期间, 将数据从串行输入引脚 (SI10) 读到 SIO10 中。

复位信号的产生将该寄存器清零 (00H)。

注意事项 当 CSOT10 = 1 (在串行通信期间) 时不要访问 SIO10。

15.3 控制串行接口 CSI10 的寄存器

串行接口 CSI10 由以下四个寄存器控制。

- 串行操作模式寄存器 10 (CSIM10)
- 串行时钟选择寄存器 10 (CSIC10)
- 端口模式寄存器 1 (PM1)
- 端口寄存器 1 (P1)

(1) 串行操作模式寄存器 10 (CSIM10)

CSIM10 用于选择操作模式、允许或禁止操作。

可由 1 位或 8 位存储器操作指令设置 CSIM10。

复位信号的产生将该寄存器清零 (00H)。

图 15-2. 串行操作模式寄存器 10 (CSIM10) 的格式

地址: FF80H 复位后: 00H R/W^{※1}

符号	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10
CSIE10	3 线串行 I/O 模式中的操作控制							
0	禁止操作 ^{※2} 且异步复位内部电路 ^{※3}							
1	允许操作							
TRMD10 ^{※4}	发送/接收模式控制							
0 ^{※5}	接收模式 (禁止发送)							
1	发送/接收模式							
DIR10 ^{※6}	规定起始位							
0	MSB							
1	LSB							
CSOT10	通信状态标志							
0	通信停止							
1	通信正在进行中							

- 注
1. 第 0 位只读。
 2. 若将 P10/SCK10/TxD0 和 P12/SO10 用作通用端口, 设置 CSIC10 为默认状态(00H)。
 3. 对 CSIM10 的第 0 位 (CSOT10) 和串行 I/O 移位寄存器 10 (SIO10) 进行复位。
 4. 当 CSOT10 = 1 (在串行通信期间) 时不要重写 TRMD10。
 5. 当 TRMD10 = 0 时, SO10 输出 (参见图 15-1(a)) 恒为低电平。当从 SIO10 读取数据时, 启动接收操作。
 6. 当 CSOT10 = 1 (在串行通信期间) 时不要重写 DIR10。

注意事项 第 5 位必须清零。

(2) 串行时钟选择寄存器 10 (CSIC10)

该寄存器规定数据发送/接收的时序，并设置串行时钟。

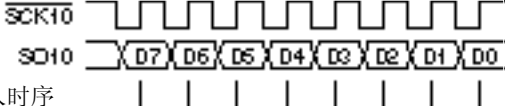
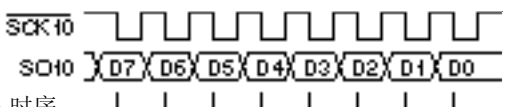
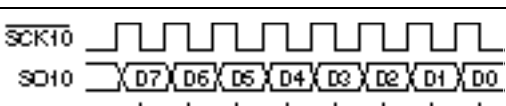
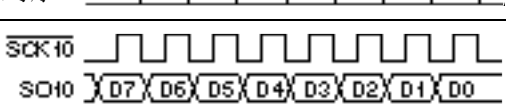
可由 1 位或 8 位存储器操作指令设置 CSIC10。

复位信号的产生将寄存器清零 (00H)。

图 15-3. 串行时钟选择寄存器 10 (CSIC10) 的格式

地址: FF81H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
CSIC10	0	0	0	CKP10	DAP10	CKS102	CKS101	CKS100

CKP10	DAP10	规定数据发送/接收的时序		类型
0	0			1
0	1			2
1	0			3
1	1			4

CKS102	CKS101	CKS100	CSI10 串行时钟的选择				模式	
			$f_{PRS} = 2 \text{ MHz}$	$f_{PRS} = 5 \text{ MHz}$	$f_{PRS} = 10 \text{ MHz}$	$f_{PRS} = 20 \text{ MHz}$		
0	0	0	$f_{PRS}/2$	1 MHz	2.5 MHz	5 MHz	禁止设置	主设备模式
0	0	1	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	
0	1	0	$f_{PRS}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	
0	1	1	$f_{PRS}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz	
1	0	0	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	312.5 kHz	625 kHz	
1	0	1	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	156.25 kHz	312.5 kHz	
1	1	0	$f_{PRS}/2^7$	15.63 kHz	39.06 kHz	78.13 kHz	156.25 kHz	
1	1	1	输入到 SCK10 的外部时钟				从设备模式	

<R>

- 注意事项
1. 当 CSIE10=1 时 (允许操作)，不要对 CSIC10 进行写操作。
 2. 若将 P10/SCK10/TxD0 和 P12/SO10 用作通用端口，设置 CSIC10 为默认状态(00H)。
 3. 复位后数据时钟的相位类型为类型 1。

备注 f_{PRS} : 外围硬件时钟频率

(3) 端口模式寄存器 1 (PM1)

该寄存器用于按位设置端口 1 为输入/输出操作模式。

当 P10/SCK10 和作为串行接口的时钟输出引脚时，对 PM10 清零，并将 P10 的输出锁存器的值设置为 1。

当 P12/SO10 作为串行接口的数据输出引脚时，则对 PM12 清零，并将 P12 的输出锁存器的值设置为 0。

当 P10/SCK10 作为串行接口的时钟输入引脚、P11/SI10/RxD0 作为数据输入引脚时，将 PM10、PM11 设置为 1。此时，P10 和 P11 的输出锁存值可能是 0 或 1。

可由 1 位或 8 位存储器操作指令设置 PM1。

复位信号的产生将该寄存器的值设置为 FFH。

图 15-4. 端口模式寄存器 1 (PM1) 的格式

地址: FF21H 复位后: FFH R/W

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

PM1n	P1n 引脚 I/O 模式选择 (n=0~7)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

15.4 串行接口 CSI10 的操作

串行接口 CSI10 可在以下两种模式中使用。

- 操作停止模式
- 3 线串行 I/O 模式

15.4.1 操作停止模式

在此模式下，不能进行串行通信，因此，可降低功率消耗。此外，P10/ $\overline{\text{SCK10}}$ /TxD0, P11/SI10/RxD0, P12/SO10 在此模式下可作为通用 I/O 端口引脚使用。

(1) 使用的寄存器

由串行操作模式寄存器 10 (CSIM10) 设置操作停止模式。
通过将 CSIM10 的第 7 位 (CSIE10) 清零，设置该模式。

(a) 串行操作模式寄存器 10 (CSIM10)

可由 1 位或 8 位存储器操作指令设置 CSIM10。

复位信号的产生将 CSIM10 清零 (00H)。

地址: FF80H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10
	CSIE10	3 线串行 I/O 模式中的操作控制						
	0	禁止操作 ^{注1} ，并异步复位内部电路 ^{注2}						

- 注
1. 若将 P10/ $\overline{\text{SCK10}}$ /TxD0 和 P12/SO10 用作通用端口，设置 CSIC10 为默认状态(00H)。
 2. 对 CSIM10 的第 0 位 (CSOT10) 和串行 I/O 移位寄存器 10 (SIO10) 进行复位。

15.4.2 3 线串行 I/O 模式

3 线串行 I/O 模式用来连接外部 IC 和具有时钟串行接口的显示控制器。

该模式中使用三条线进行通信：串行时钟（ $\overline{\text{SCK10}}$ ）、串行输出（SO10）和串行输入（SI10）线。

(1) 使用的寄存器

- 串行操作模式寄存器 10（CSIM10）
- 串行时钟选择寄存器 10（CSIC10）
- 端口模式寄存器 1（PM1）
- 端口寄存器 1（P1）

3 线串行 I/O 模式设置操作的基本步骤如下。

- <1> 设置 CSIC10 寄存器（参见图 15-3）
- <2> 设置 CSIM10 寄存器的第 0 位，第 4 位和第 6 位（CSOT10, DIR10 和 TRMD10）（参见图 15-2）。
- <3> CSIM10 的第 7 位（CSIE1n）置 1。 → 允许发送/接收。
- <4> 将数据写入发送缓冲寄存器 10（SOTB10）。 → 启动发送/接收操作。
从串行 I/O 移位寄存器 10（SIO10）中读取数据。 → 启动数据接收操作。

注意事项 在设置端口模式寄存器和端口寄存器时，要考虑与通信另一方的关系。

寄存器设置与引脚之间的关系如下所示。

表 15-2. 寄存器设置与引脚之间的关系

CSIE10	TRMD10	PM11	P11	PM12	P12	PM10	P10	CSI10 操作	引脚功能		
									SI10/RxD0/ P11	SO10/P12	SCK10/ TxD0/P10
0	x	x ^{注1}	x ^{注1}	x ^{注1}	x ^{注1}	x ^{注1}	x ^{注1}	停止	RxD0/P11	P12	TxD0/ P10 ^{注2}
1	0	1	x	x ^{注1}	x ^{注1}	1	x	从设备接收 ^{注3}	SI10	P12	SCK10 (输入) ^{注3}
1	1	x ^{注1}	x ^{注1}	0	0	1	x	从设备发送 ^{注3}	RxD0/P11	SO10	SCK10 (输入) ^{注3}
1	1	1	x	0	0	1	x	从设备发送/接收 ^{注3}	SI10	SO10	SCK10 (输入) ^{注3}
1	0	1	x	x ^{注1}	x ^{注1}	0	1	主设备接收	SI10	P12	SCK10 (输出)
1	1	x ^{注1}	x ^{注1}	0	0	0	1	主设备发送	RxD0/P11	SO10	SCK10 (输出)
1	1	1	x	0	0	0	1	主设备发送/接收	SI10	SO10	SCK10 (输出)

- 注
1. 可用作端口引脚。
 2. 如果 P10/SCK10/TxD0 用作端口引脚，则将 CKP10 清零。
 3. 如果使用从设备模式，则须将 CKS102、CKS101 和 CKS100 设置为 1、1 和 1。

备注

x: 不必考虑

CSIE10: 串行操作模式寄存器 10 (CSIM10) 的第 7 位

TRMD10: CSIM10 的第 6 位

CKP10: 串行时钟选择寄存器 10 (CSIC10) 的第 4 位

CKS102, CKS101, CKS100: CSIC10 的第 0 位 ~ 第 2 位

PM1x: 端口模式寄存器

P1x: 端口输出锁存器

(2) 通信操作

在 3 线串行 I/O 模式中，是以 8 位为单元发送或接收数据。每位数据与串行时钟同步地发送或接收。

当串行操作模式寄存器 10 (CSIM10) 的第 6 位 (TRMD10) =1 时，允许发送或接收数据。当数据被写入到发送缓冲寄存器 10 (SOTB10) 时，开始进行发送/接收。此外，串行操作模式寄存器 10 (CSIM10) 的第 6 位 (TRMD10) = 0 时，允许接收数据。

当数据从串行 I/O 移位寄存器 10 (SIO10) 被读出时，启动接收操作。

通信开始后，CSIM10 的第 0 位 (CSOT10) =1。当 8 位数据通信结束时，设置通信完成中断请求标志 (CSIIF10)，且对 CSOT10 清零。这样允许进行下一个通信。

注意事项 当 CSOT10 = 1 时 (串行通信期间)，不要访问控制寄存器和数据寄存器。

图 15-5. 3 线串行 I/O 模式中的时序 (1/2)

(a) 发送/接收时序 (类型 1: TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 0)

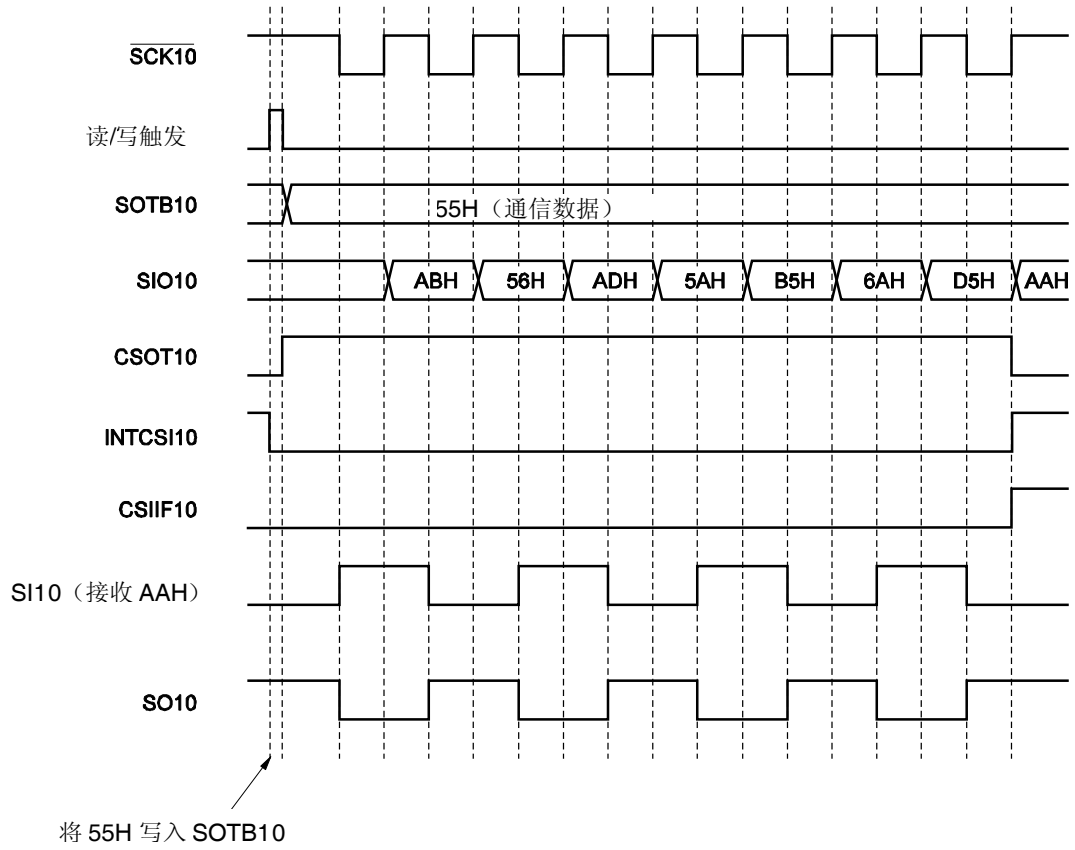


图 15-5. 3 线串行 I/O 模式中的时序 (2/2)

(b) 发送/接收时序 (类型 2: TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 1)

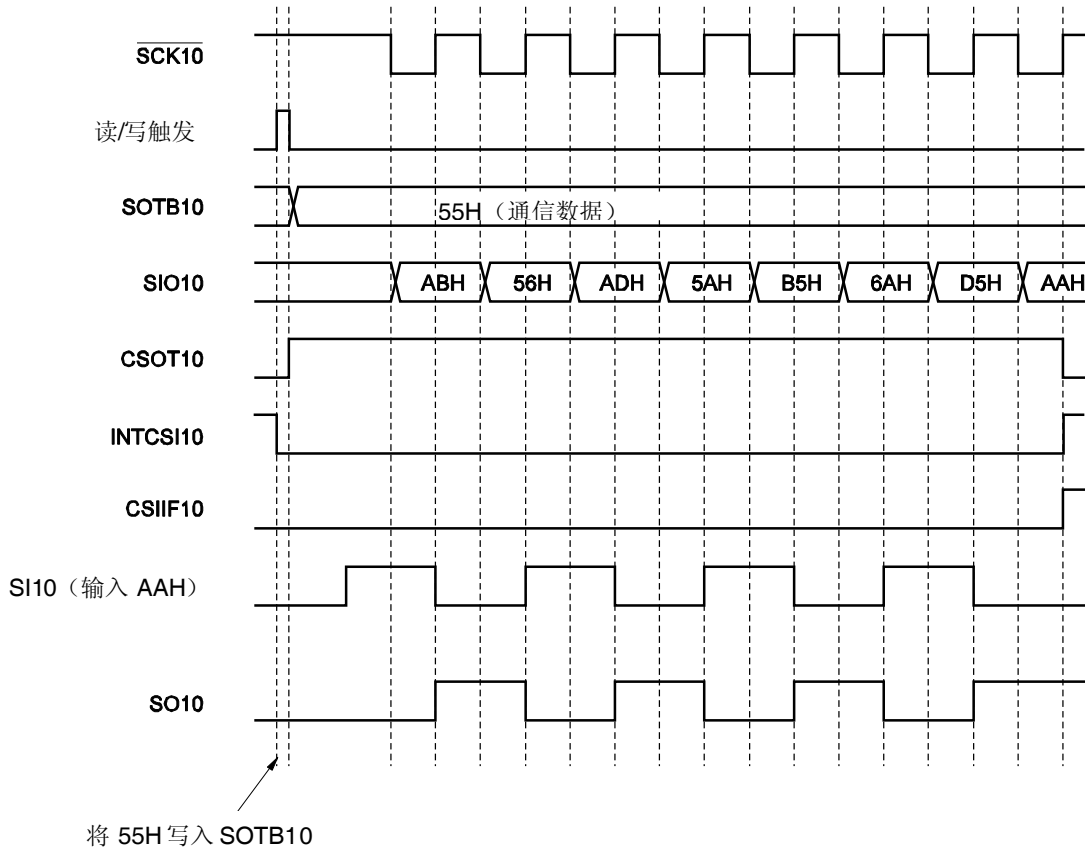
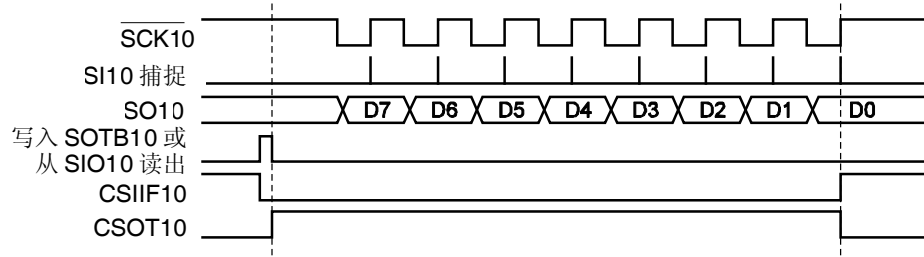
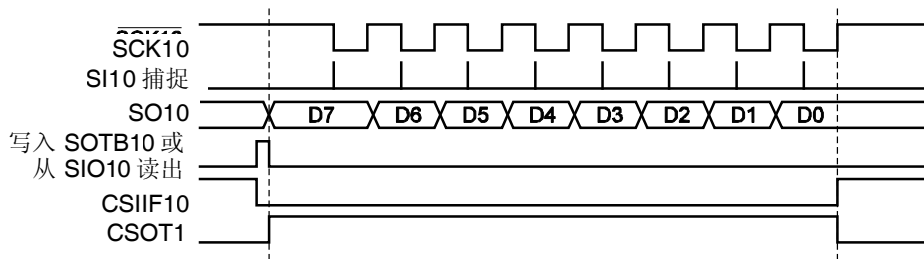


图 15-6. 时钟/数据相位的时序

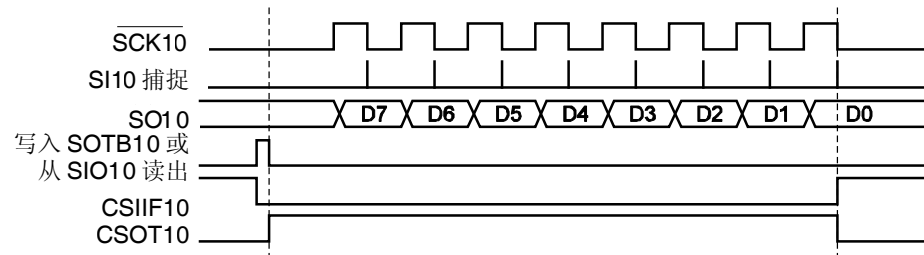
(a) 类型 1: CKP10 = 0, DAP10 = 0, DIR10 = 0



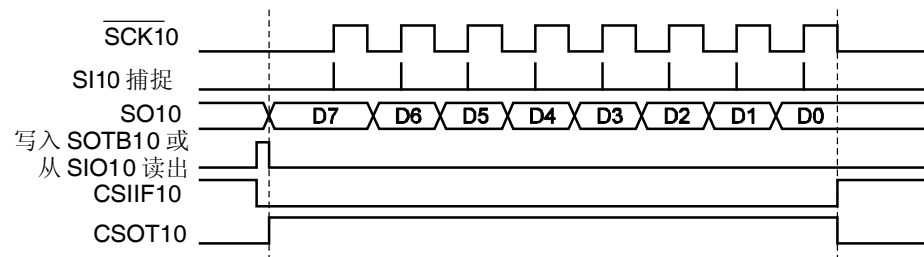
(b) 类型 2: CKP10 = 0, DAP10 = 1, DIR10 = 0



(c) 类型 3: CKP10 = 1, DAP10 = 0, DIR10 = 0



(d) 类型 4: CKP10 = 1, DAP10 = 1, DIR10 = 0

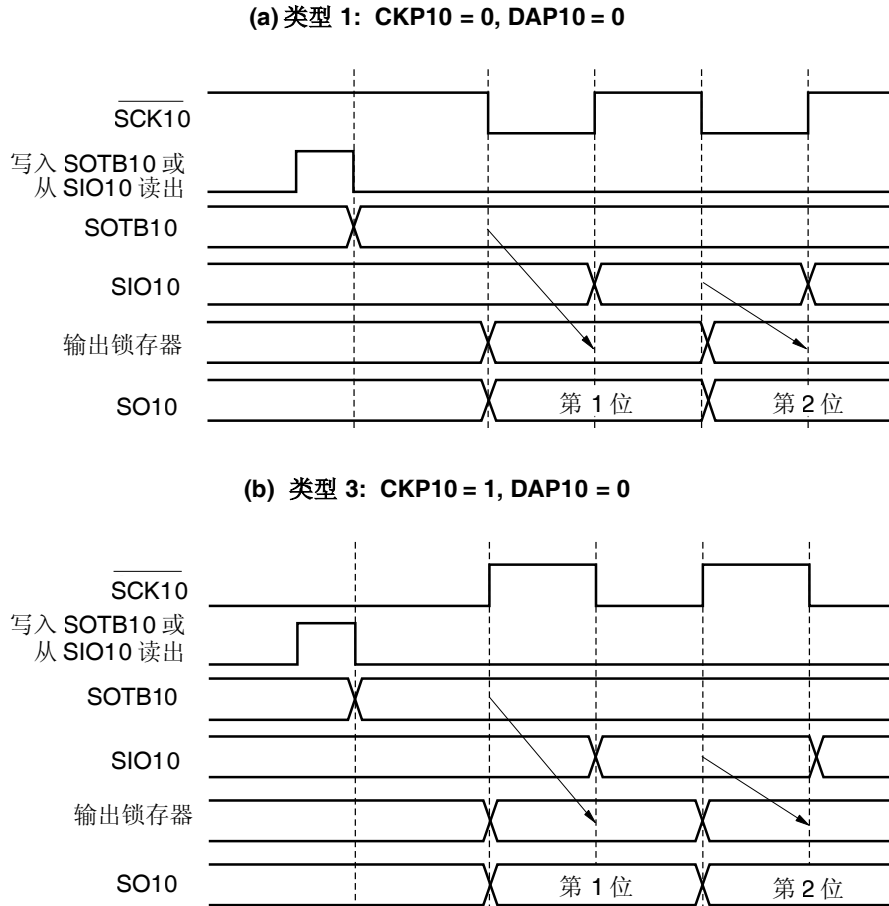


备注 上图显示了一个通信操作过程，即采用 MSB first 发送数据。

(3) 输出到 SO10 引脚的时序（起始位）

通信开始后，将发送缓冲寄存器 10（SOTB10）的值从 SO10 引脚输出。
此时起始位的输出操作如下所示。

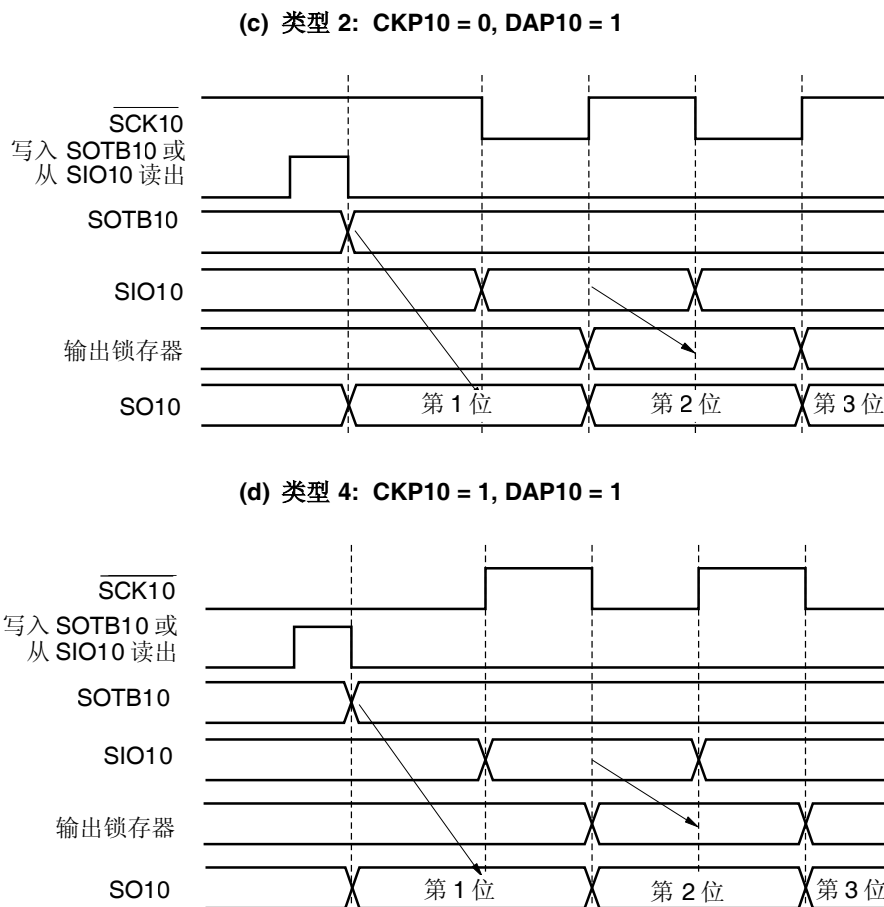
图 15-7. 起始位的输出操作 (1/2)



在 $\overline{\text{SCK10}}$ 的下降沿（或上升沿），起始位数据通过 SOTB10 寄存器被直接锁存到输出锁存器，并通过输出选择器从 SO10 引脚输出。然后在 $\overline{\text{SCK10}}$ 下一个上升沿（或下降沿）将 SOTB10 寄存器的内容传送到 SIO10 中，并移出 1 位。同时，通过 SIO10 引脚将接收数据的起始位存储到 SIO10 寄存器中。

在下一个 $\overline{\text{SCK10}}$ 的下降沿（或上升沿），通过 SIO10 寄存器将第 2 位以及随后各位锁存到输出锁存器，然后从 SO10 引脚输出数据。

图 15-7. 起始位的输出操作 (2/2)



在 SOTB10 的写信号或 SIO10 的读信号的下降沿，起始位数据通过 SOTB10 寄存器被直接锁存，并通过输出选择器从 SO10 引脚输出。然后在 SCK10 下一个下降沿（或上升沿）将 SOTB10 寄存器的内容传送到 SIO10 中，并移出 1 位。同时，将接收数据的起始位通过 SIO10 引脚存储到 SIO10 寄存器。在下一个 SCK10 的上升沿（或下降沿），通过 SIO10 寄存器将第 2 位以及随后各位锁存到输出锁存器，然后从 SO10 引脚输出数据。

(4) SO10 引脚的输出值 (终止位)

在通信完成后, SO10 引脚保持终止位的输出值。

图 15-8. SO10 引脚的输出值 (终止位) (1/2)

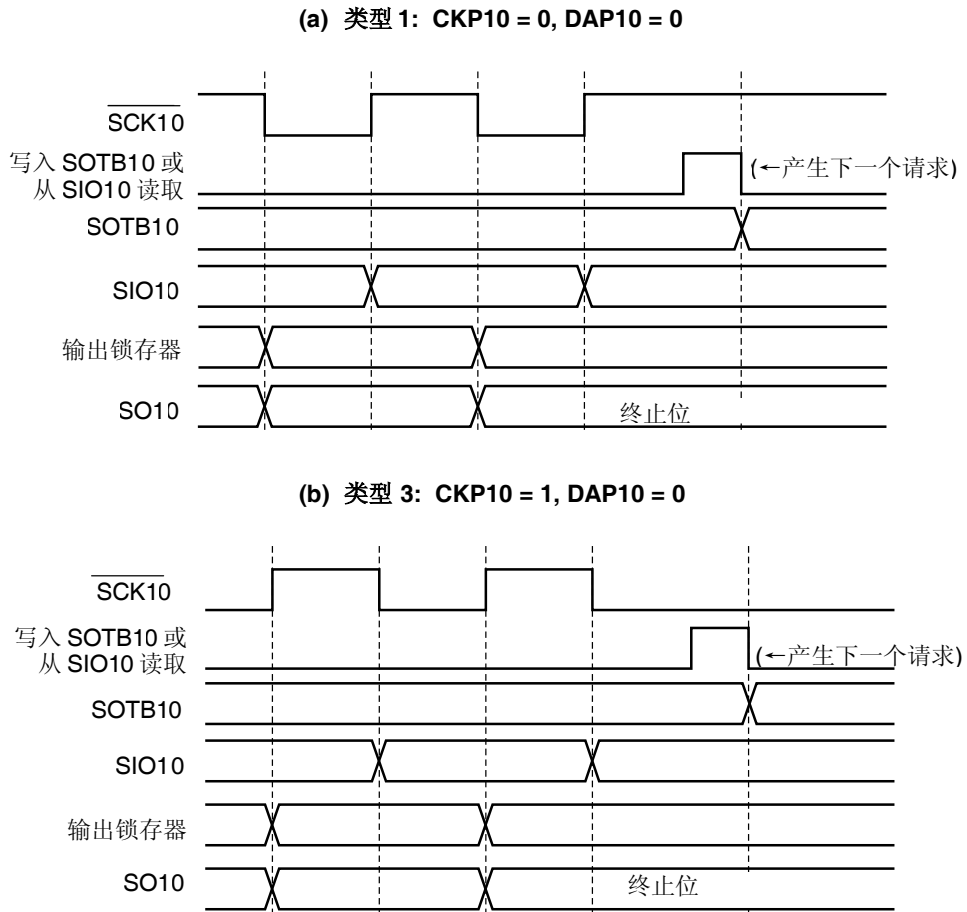
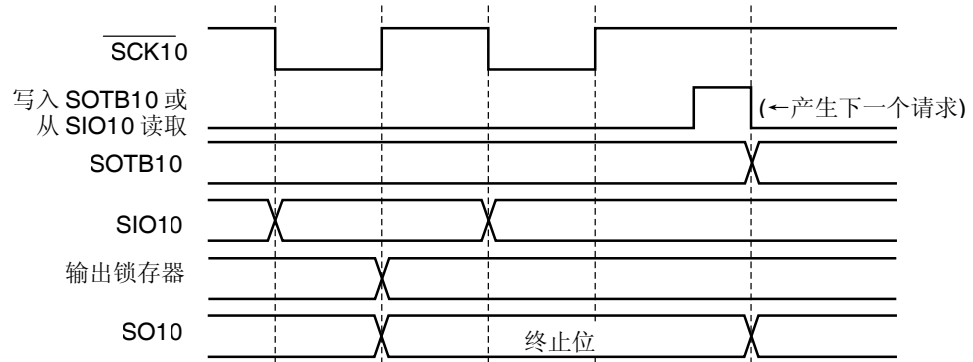
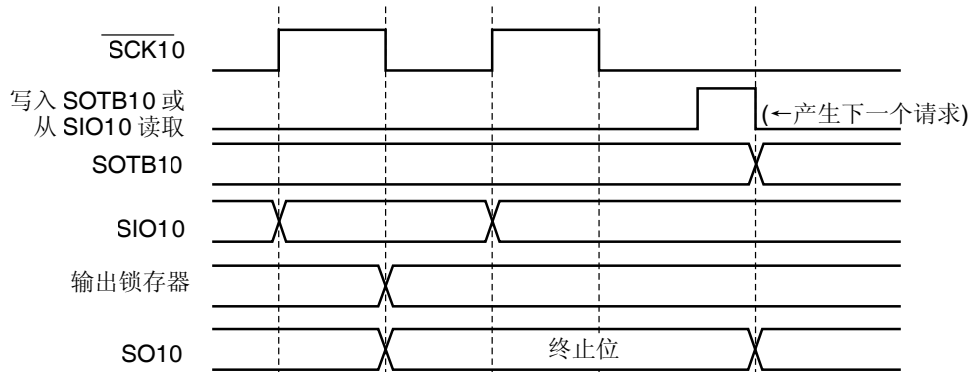


图 15-8. SO10 引脚的输出值 (终止位) (2/2)

(c) 类型 2: CKP10 = 0, DAP10 = 1



(d) 类型 4: CKP10 = 1, DAP10 = 1



(5) SO10 输出 (参看 图 15-1 中的 (a))

如果串行操作模式寄存器 10 (CSIM10) 的第 7 位 (CSIE10) 被清零, 则 SO10 输出状态如下。

表 15-3. SO10 输出状态

TRMD10	DAP10	DIR10	SO10 输出 ^{注1}
TRMD10 = 0 ^{注2}	-	-	输出低电平 ^{注2}
TRMD10 = 1	DAP10 = 0	-	SO10 锁存值 (低电平输出)
	DAP10 = 1	DIR10 = 0	SOTB10 第 7 位的值
		DIR10 = 1	SOTB10 第 0 位的值

- 注
1. 根据 PM12 和 P12 以及 SO10 的输出, 决定 SO10/P12 引脚的实际输出。
 2. 复位后的状态

注意事项 如果对 TRMD10、DAP10 和 DIR10 进行写操作, 则 SO10 的输出值将发生改变。

16.1 串行接口 IIC0 的功能

串行接口 IIC0 有以下两种模式。

(1) 操作停止模式

在不进行串行传送时可采用此模式，以降低功耗。

(2) I²C 总线模式 (支持多个主设备)

该模式采用两条线(一条串行时钟(SCL0)总线和一条串行数据总线(SDA0))进行多设备的 8 位数据传送。

该模式遵循 I²C 总线格式并且主设备能够通过串行数据总线，产生“起始条件”，“地址”，“传送方向指示”，“数据”和“停止条件”等信号传送至从设备。从设备能够通过硬件自动地检测这些接收到的状态和数据。这个功能可以简化控制 I²C 总线的应用程序部分。

由于 SCL0 和 SDA0 引脚是开漏输出，IIC0 总线的串行时钟线和串行数据总线需要连接上拉电阻。

图 16-1 是串行接口 IIC0 的框图。

图 16-1. 串行接口 IIC0 框图

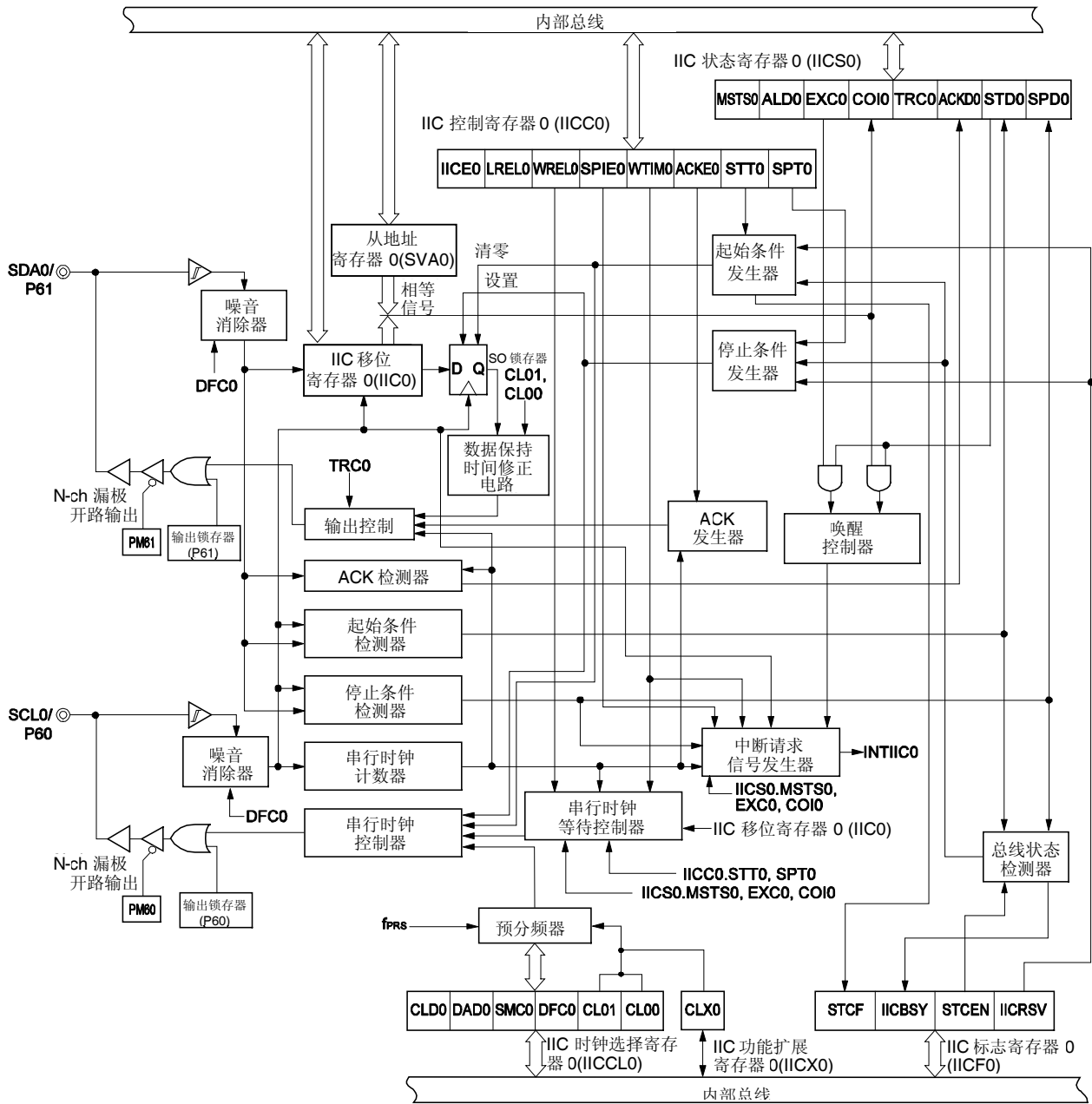
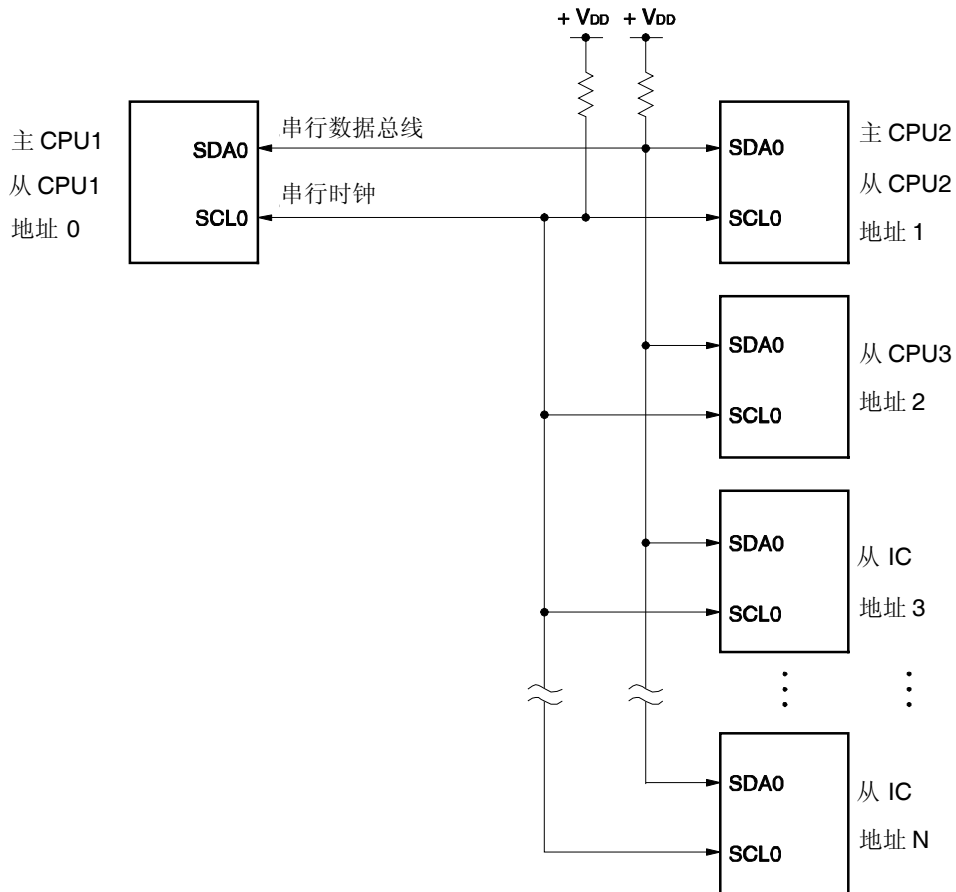


图 16-2 是串行总线配置示例。

图 16-2. 使用 I²C 总线的串行总线配置示例



16.2 串行接口 IIC0 的配置

串行接口 IIC0 包括下列硬件。

表 16-1. 串行接口 IIC0 配置

项目	配置
寄存器	IIC 移位寄存器 0(IIC0) 从设备地址寄存器 0(SVA0)
控制寄存器	IIC 控制寄存器 0(IICC0) IIC 状态寄存器 0(IICS0) IIC 标志寄存器 0(IICF0) IIC 时钟选择寄存器 0(IICCL0) IIC 功能扩展寄存器 0(IICX0) 端口模式寄存器 6(PM6) 端口寄存器 6(P6)

(1) IIC 移位寄存器 0(IIC0)

IIC0 用于 8 位串行数据和 8 位并行数据的相互转换，并且和串行时钟同步。IIC0 可以用来发送和接收数据。

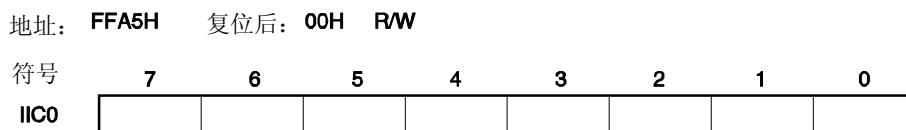
对 IIC0 的读写操作可以控制实际的发送和接收操作。

在等待期间通过对 IIC0 写入数据，可以取消等待状态并启动数据传送操作。

可由 8 位存储器操作指令设置 IIC0。

复位信号的产生将 IIC0 设置为 00H。

图 16-3. IIC 移位寄存器 0 (IIC0) 的格式



注意事项 1. 在数据传送期间不要对 IIC0 进行写操作。

2. 仅在等待期间才能对 IIC0 进行读写操作。除等待期间外，禁止在其他的通信状态下访问 IIC0。然而，作为主设备时，在通信触发位(STT0)被设为 1 之后 IIC0 只能被写入一次。

(2) 从设备地址寄存器 0(SVA0)

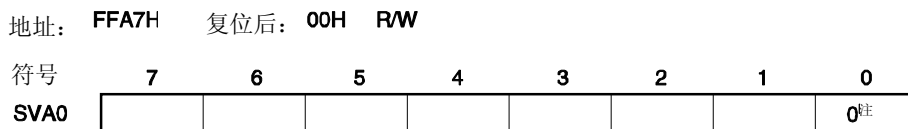
该寄存器用于在从设备模式下存储本地地址。

可由 8 位存储器操作指令设置 SVA0。

但是当 STD0=1 时，禁止重写该寄存器(当检测到起始条件时)。

复位信号将 SVA0 设置为 00H。

图 16-4. 从设备地址寄存器 0 (SVA0) 的格式



注 第 0 位恒为 0。

(3) SO 锁存器

SO 锁存器用来保持 SDA0 引脚的输出电平。

(4) 唤醒控制器

当该寄存器接收的地址与从设备地址寄存器 0(SVA0)设置的地址相等，或者当接收到一个扩展代码时，这个电路将产生一个中断请求信号(INTIIC0)。

(5) 预分频器

用于选择采样时钟。

(6) 串行时钟计数器

该计数器对发送/接收操作期间输出或输入的串行时钟进行计数，同时用于验证 8 位数据是否已发送或接收。

(7) 中断请求信号发生器

此电路用于控制中断请求信号(INTIIC0)的产生。

一个 I²C 中断请求由下列两个触发条件产生：

- 串行时钟的第 8 个或第 9 个下降沿(通过 WTIM0 位设置)
- 当检测到停止条件(通过 SPIE0 位设置)时

备注 WTIM0 位：IIC 控制寄存器 0 (IICC0) 的第 3 位
 SPIE0 位：IIC 控制寄存器 0 (IICC0) 的第 4 位

(8) 串行时钟控制器

在主设备模式下，此电路通过 SCL0 引脚，从采样时钟产生时钟输出。

(9) 串行时钟等待控制器

用于控制等待时序。

(10) ACK 信号发生器，停止条件检测器，起始条件检测器和 ACK 信号检测器

这些电路用于产生和检测每种状态。

(11) 数据保持时间修正电路

此电路根据串行时钟的下降沿产生数据保持时间。

(12) 起始条件发生器

当 STT0 位为 1 时，该电路产生一个起始条件信号。

但是，在禁止通信预约的状态下(IICRSV 位 = 1)，只要总线尚未被释放(IICBSY 位 = 1)，起始条件请求将被忽略并且 STCF 被置为 1。

(13) 停止条件发生器

当 SPT0 位为 1 时，此电路产生一个停止条件。

(14) 总线状态检测器

此电路通过检测起始条件和停止条件来确定总线是否被释放。

然而，如果操作后总线状态无法被立即检测到，则初始状态由 **STCEN** 位来设置。

备注	STT0 位:	IIC 控制寄存器 0 (IICC0) 的第 1 位
	SPT0 位:	IIC 控制寄存器 0 (IICC0) 的第 0 位
	IICRSV 位:	IIC 标志寄存器 0 的第 0 位
	IICBSY 位:	IIC 标志寄存器 0 的第 6 位
	STCF 位:	IIC 标志寄存器 0 的第 7 位
	STCEN 位:	IIC 标志寄存器 0 的第 1 位

16.3 控制串行接口 IIC0 的寄存器

串行接口 IIC0 由下列 7 个寄存器控制：

- IIC 控制寄存器 0 (IICC0)
- IIC 标志寄存器 0 (IICF0)
- IIC 状态寄存器 0 (IICS0)
- IIC 时钟选择寄存器 0 (IICCL0)
- IIC 功能扩展寄存器 0 (IICX0)
- 端口模式寄存器 6 (PM6)
- 端口寄存器 6 (P6)

(1) IIC 控制寄存器 0 (IICC0)

这个寄存器用于允许/停止 I²C 的操作、设置等待时间和设置 I²C 的其他操作。

可由 1 位或 8 位存储器操作指令对 IICC0 进行设置。但必须在 IICE0 = 0 或者在等待期间设置 SPIE0，WTIM0 和 ACKE0 位。当 IICE0 位从“0”变为“1”时，这些位可以同时被设置。

复位信号的产生将 IICC0 设置为 00H。

图 16-5. IIC 控制寄存器 0 (IICC0) 的格式 (1/4)

地址: FFA6H 复位后: 00H R/W

符号 <7> <6> <5> <4> <3> <2> <1> <0>

IICC0	IICE0	LRELO	WRELO	SPIE0	WTIMO	ACKE0	STT0	SPT0
-------	-------	-------	-------	-------	-------	-------	------	------

IICE0	I ² C 操作允许
0	停止操作。复位 IIC 状态寄存器 0(IICS0) ^{注1} 。停止内部操作。
1	允许操作
当 SCL0 和 SDA0 在高电平时，必须将该位设置为 1。	
清零条件 (IICE0 = 0)	设置条件 (IICE0 = 1)
<ul style="list-style-type: none"> • 通过指令清零 • 复位 	<ul style="list-style-type: none"> • 通过指令设置

LRELO ^{注2}	退出通信
0	正常操作
1	此时从当前的通信模式退出，进入待机模式。当执行此操作后设置的值自动清零。 用于那些已接收了本地无关扩展码的情况。 SCL0 和 SDA0 线被设置为高阻抗状态。 下列 IIC 控制器 0 (IICC0) 和 IIC 状态寄存器 0 (IICS0) 的标志位被清零。 • STT0 • SPT0 • MSTS0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0
从通信模式退出后一直保持待机模式，直到满足下列通信条件。	
<ul style="list-style-type: none"> • 检测到停止条件后，在主设备模式下重新启动。 • 启动后，发生地址相等或接收到扩展码。 	
清零条件 (LRELO = 0)	设置条件 (LRELO = 1)
<ul style="list-style-type: none"> • 执行后自动清零 • 复位 	<ul style="list-style-type: none"> • 通过指令设置

WRELO ^{注2}	等待取消
0	不取消等待
1	取消等待。在等待取消后，此设置被自动清除。
在第 9 个时钟脉冲的下降沿有效启动的发送状态 (TRC0 = 1) 下，在等待期间设置 WRELO(等待取消) 时，SDA0 将进入高阻抗状态 (TRC0 = 0)。	
清零条件 (WRELO = 0)	设置条件 (WRELO = 1)
<ul style="list-style-type: none"> • 执行后自动清零 • 复位 	<ul style="list-style-type: none"> • 通过指令设置

- 注
1. 对 IICS0 寄存器、IICF0 寄存器的 STCF0 位与 IICBSY 位和 IICCL0 寄存器的 CLD0 位与 DAD0 位进行复位。
 2. 当 IICE0 = 0 时，本标志信号无效。

注意事项 允许 I²C 操作 (IICE0 = 1) 后，在 SCL0 为高电平并且 SDA0 为低电平时，立即检测到起始条件。允许 I²C 操作 (IICE0 = 1) 后，立即通过 1 位存储器操作指令设置 LRELO (1)。

图 16-5. IIC 控制寄存器 0 (IICC0) 的格式 (2/4)

SPIE0 ^{※1}	当检测到停止条件时，允许/禁止产生中断请求	
0	禁止	
1	允许	
清零条件 (SPIE0 = 0)		设置条件 (SPIE0 = 1)
<ul style="list-style-type: none"> 通过指令清零 复位 		<ul style="list-style-type: none"> 通过指令进行设置

WTIMO ^{※1}	对等待和中断请求产生的控制	
0	<p>在第 8 个时钟的下降沿产生中断请求。</p> <p>主设备模式：输出 8 个时钟之后，将时钟输出设置为低电平且设置等待状态。</p> <p>从设备模式：输入 8 个时钟之后，将时钟设置为低电平且为主设备设置等待状态。</p>	
1	<p>在第 9 个时钟的下降沿产生中断请求。</p> <p>主设备模式：输出 9 个时钟之后，将时钟输出设置为低电平且设置等待状态。</p> <p>从设备模式：输入 9 个时钟之后，将时钟设置为低电平且设置主设备等待状态。</p>	
<p>地址传送期间，在第 9 个时钟的下降沿产生中断，而与该位的设置无关。地址传送完成之后该位设置才有效。主设备模式下，在地址传送期间的第 9 个时钟的下降沿插入一个等待信号。对于已接收到本地地址的从设备，在发出应答信号(ACK)之后的第 9 个时钟的下降沿插入一个等待信号。但是，当从设备已经收到一个扩展码时，将在第 8 个时钟的下降沿插入等待信号。</p>		
清零条件 (WTIMO = 0)		设置条件 (WTIMO = 1)
<ul style="list-style-type: none"> 通过指令清零 复位 		<ul style="list-style-type: none"> 通过指令进行设置

ACKE0 ^{※1,2}	应答控制	
0	禁止应答	
1	允许应答。在第 9 个时钟期间，将 SDA0 设置为低电平。但是，在非扩展模式及地址传送期间， $\overline{\text{ACK}}$ 信号无效。	
清零条件 (ACKE0 = 0)		设置条件 (ACKE0 = 1)
<ul style="list-style-type: none"> 通过指令进行清零 复位 		<ul style="list-style-type: none"> 通过指令进行设置

- 注
1. IICE0 = 0 时，此标志信号无效。
 2. 在地址传送期间及编码为非扩展码时，所设的值无效。
在从设备模式下并且地址相等时，无论设置何值都将产生应答信号。

图 16-5. IIC 控制寄存器 0 (IICC0) 的格式 (3/4)

STT0 ^注	起始条件触发	
0	不产生起始条件	
1	总线释放时(STOP 模式下): 产生一个起始条件(作为主设备启动)。SCL0 处于高电平时, SDA0 从高电平变为低电平时产生起始条件。然后, 经过额定时间, SCL0 变为低电平。 有第三方通信时: <ul style="list-style-type: none"> 当允许通信预约功能时 (IICRSV = 0) 该位作为起始条件预约标志。设置为 1 时, 在总线释放之后自动产生一个起始条件。 当禁止通信预约功能时 (IICRSV = 1) 将 STCF 设置为 1, 且 STT0 被清零。不产生起始条件。 等待状态(主设备模式): 退出等待状态之后产生一个重复起始条件。	
设置时序的注意事项 <ul style="list-style-type: none"> 主设备接收: 传送期间不能设置为 1。仅在 ACKE0 被清零且从设备通报结束接收时的等待期间, 才能设为 1。 主设备发送: 应答期间内, 不能正常产生起始条件。在第 9 个时钟输出后的等待期间可以设置为 1。 不能和 SPT0 同时设为 1。 禁止在 STT0 被清零前重复设置为 1。 		
清零条件 (STT0 = 0)		设置条件 (STT0 = 1)
<ul style="list-style-type: none"> 禁止通信预约时, 通过将 SST0 设置为 1 进行清零 由仲裁失败清零 通过主设备产生起始条件后进行清零 由 LRELO=1(退出通信)清零 当 IICE0 = 0 (操作停止)时 复位 		<ul style="list-style-type: none"> 通过指令进行设置

注 IICE0 = 0 时, 该标志信号无效。

- 备注
- 第 1 位 (STT0)在数据设置后再被读取时将会变为 0。
 - IICRSV: IIC 标志寄存器 (IICF0)的第 0 位
STCF: IIC 标志寄存器(IICF0)的第 7 位

图 16-5. IIC 控制寄存器 0 (IICC0) 的格式 (4/4)

SPT0	停止条件触发器	
0	不产生停止条件	
1	产生停止条件 (结束主设备传送) SDA0 变为低电平后, 设置 SCL0 为高电平或等待其变为高电平。然后, 经过额定时间后, SDA0 从低电平变为高电平, 则产生一个停止条件。	
设置时序注意事项 <ul style="list-style-type: none"> 主设备接收: 传送期间不能设为 1。 仅在 ACKE0 被清零且从设备通报结束接收时的等待期间, 才能设为 1。 主设备发送: 应答期间内, 不能正常产生停止条件。在第 9 个时钟输出后的等待期间可以设为 1。 不能和 SST0 同时设置为 1 仅在主设备模式^注下, SPT0 才能设置为 1 WTIMO 清零后, 如果在输出 8 个时钟之后的等待期间将 SPT0 设为 1, 那么注意, 在第 9 个时钟的高电平期间将产生一个停止条件。在输出 8 个时钟后的等待期间 WTIMO 应从 0 变为 1, 并且在输出第 9 个时钟后的等待期间 SPT0 应被设置为 1。 禁止在 SPT0 被清零前重复设置为 1。 		
清零条件 (SPT0 = 0)		设置条件 (SPT0 = 1)
<ul style="list-style-type: none"> 由仲裁失败清零 检测到停止条件后自动清零 可由 LREL0=1(退出通信)清零 IICE0 = 0 时(操作停止) 复位 		<ul style="list-style-type: none"> 通过指令进行设置

注 仅在主设备模式下 SPT0 可设置为 1。但是, 在切换到允许操作状态后检测到第 1 个停止条件之前必须将 SPT0 设置为 1, 并产生停止条件。如需了解详细信息, 可参见 16.5.15 其他注意事项。

注意事项 在第 9 个时钟, 当 IIC 状态寄存器 0(IICS0)的第 3 位(TRC0)被设为 1 和 WREL0 设置为 1 时, 取消等待状态后, TRC0 被清零, SDA0 线被设为高阻抗状态。

备注 第 0 位(SPT0)在设置后再读取时将变为 0。

(2) IIC 状态寄存器 0 (IICS0)

此寄存器用于表示 I²C 的状态。

仅当 STT0 = 1 且在等待期间，才能通过 1 位或 8 位存储器操作指令读取 IICS0。

复位信号的产生将 IICS0 设置为 00H。

注意事项 如果从 IICS0 中读取数据，将产生一个等待周期。当 CPU 使用副系统时钟且外围硬件时钟停止时，不要从 IICS0 中读取数据。详见 第三十一章 等待注意事项。

图 16-6. IIC 状态寄存器 0 (IICS0) 的格式 (1/3)

地址: FFAAH 复位后: 00H R

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	主设备状态	
0	从设备状态或通信待机状态	
1	主设备通信状态	
清零条件 (MSTS0 = 0)		设置条件 (MSTS0 = 1)
<ul style="list-style-type: none"> 检测到停止条件时 ALD0=1 时 (仲裁失败) 由 LREL0 = 1 (退出通信) 清零 IICE0 从 1 变为 0 (操作停止) 时 复位 		<ul style="list-style-type: none"> 起始条件产生时

ALD0	仲裁失败的检测	
0	此状态表示没有仲裁或仲裁结果为“获胜”。	
1	此状态表示仲裁结果为“失败”。MSTS0 被清零。	
清零条件 (ALD0 = 0)		设置条件 (ALD0 = 1)
<ul style="list-style-type: none"> 读取 IICS0 之后自动清零 IICE0 从 1 变为 0 (操作停止) 时 复位 		<ul style="list-style-type: none"> 仲裁结果为“失败”时。

EXC0	扩展码接收的检测	
0	未接收到扩展码	
1	接收到扩展码	
清零条件 (EXC0 = 0)		设置条件 (EXC0 = 1)
<ul style="list-style-type: none"> 检测到起始条件时 检测到停止条件时 由 LREL0 = 1 (退出通信) 清零 IICE0 从 1 变为 0 (操作停止) 时 复位 		<ul style="list-style-type: none"> 当接收到的地址数据的高 4 位为“0000”或“1111”时 (在第 8 个时钟的上升沿设置)

注 即使使用 1 位存储器操作指令对 IICS0 之外的位进行操作，此寄存器也会清零。因此当使用 ALD0 位时，应先读取该位数据。

备注 LREL0: IIC 控制寄存器 0(IICC0) 的第 6 位
IICE0: IIC 控制寄存器 0(IICC0) 的第 7 位

图 16-6. IIC 状态寄存器 0 (IICS0) 的格式 (2/3)

COI0	地址相等的检测	
0	地址不相等	
1	地址相等	
清零条件 (COI0 = 0)		设置条件 (COI0 = 1)
<ul style="list-style-type: none"> • 检测到起始条件时 • 检测到停止条件时 • 由 LREL0 = 1 清零 (退出通信) • IICE0 从 1 变为 0 时 (操作停止) • 复位 		<ul style="list-style-type: none"> • 当接收地址和本地地址相等时 (从设备地址寄存器 0 (SVA0)) (在第 8 个时钟的上升沿设置)

TRC0	发送/接收状态的检测	
0	接收状态 (非发送状态)。SDA0 线被设为高阻抗状态。	
1	发送状态。允许将 SO0 锁存器中的内容输出到 SDA0 线 (在第 1 个字节的第 9 个时钟的下降沿才有效)。	
清零条件 (TRC0 = 0)		设置条件 (TRC0 = 1)
<主设备模式和从设备模式> <ul style="list-style-type: none"> • 检测到停止条件时 • 由 LREL0 = 1 (退出通信) 清零 • IICE0 从 1 变为 0 (操作停止) 时 • 由 WREL0 = 1 注 (取消等待) 清零 • ALD0 从 0 变为 1 (仲裁失败) 时 • 复位 <主设备模式> <ul style="list-style-type: none"> • 当“1”输出到第 1 个字节的 LSB (传送方向指示位) 时 <从设备模式> <ul style="list-style-type: none"> • 检测到起始条件时 • 当“0”输入到第 1 个字节的 LSB (传送方向指示位) 时 <未用于通信时>		<主设备模式> <ul style="list-style-type: none"> • 产生起始条件时 • 当“0”输出到第 1 个字节的 LSB (传送方向指示位) 时 <从设备模式> <ul style="list-style-type: none"> • 当“1”输入到第 1 个字节的 LSB (传送方向指示位) 时

注 当 IIC 状态寄存器 0 (IICS0) 的第 3 位 (TRC0) 为 1 时，通过在第 9 个时钟将 IIC 控制寄存器 0 (IICC0) 的第 5 位 (WREL0) 设置为 1，来取消等待状态，TRC0 被清零且 SDA0 线进入高阻抗状态。

备注 LREL0: IIC 控制寄存器 0 (IICC0) 的第 6 位
IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

图 16-6. IIC 状态寄存器 0 (IICS0) 的格式(3/3)

ACKD0	应答信号的检测 (\overline{ACK})	
0	未检测到应答信号	
1	已检测到应答信号	
清零条件 (ACKD0 = 0)		设置条件 (ACKD0 = 1)
<ul style="list-style-type: none"> • 检测到停止条件时 • 在下一个字节的第一个时钟的上升沿 • 由 LREL0 = 1 (退出通信)清零 • 当 IICE0 从 1 变为 0 (操作停止)时 • 复位 		<ul style="list-style-type: none"> • 在 SCL0 的第 9 个时钟的上升沿将 SDA0 线设为低电平之后

STD0	起始条件的检测	
0	未检测到起始条件	
1	已检测到起始条件。这表示地址传送阶段有效。	
清零条件 (STD0 = 0)		设置条件 (STD0 = 1)
<ul style="list-style-type: none"> • 检测到停止条件时 • 在地址发送之后的下一个字节的第 1 个时钟的上升沿 • 由 LREL0 = 1 (退出通信)清零 • 当 IICE0 从 1 变为 0 (操作停止)时 • 复位 		<ul style="list-style-type: none"> • 检测到起始条件时

SPD0	停止条件的检测	
0	未检测到停止条件	
1	已检测到停止条件。主设备通信结束，释放总线。	
清零条件 (SPD0 = 0)		设置条件 (SPD0 = 1)
<ul style="list-style-type: none"> • 在设置该位并检测到起始条件后的地址发送字节第 1 个时钟的上升沿 • 当 IICE0 从 1 变为 0 (操作停止)时 • 复位 		<ul style="list-style-type: none"> • 检测到停止条件时

备注 LREL0: IIC 控制寄存器 0(IICC0)的第 6 位
IICE0: IIC 控制寄存器 0(IICC0)的第 7 位

(3) IIC 标志寄存器 0 (IICF0)

此寄存器用于设置 I²C 的操作模式，并指示 I²C 总线的状态。

IICF0 可由 1 位或 8 位的存储器操作指令进行设置。但是，STCF 和 IICBSY 位是只读的。

IICRSV 位用于允许/禁止通信预约功能 (详见 16.5.14 通信预约)。

STCEN 用于设置 IICBSY 的初始值 (详见 16.5.15 其他注意事项)。

仅当 I²C 操作被禁止(IIC 控制寄存器 0 (IICC0)的第 7 位(IICE0) = 0)时才可对 IICRSV 和 STCEN 位进行写操作。

当允许操作时，可以读取 IICF0 寄存器。

复位信号的产生将 IICF0 设置为 00H。

图 16-7. IIC 标志寄存器 0 (IICF0) 的格式

地址: FFABH 复位后: 00H R/W^注

符号	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT0 清零标志	
0	产生起始条件	
1	未能产生起始条件: STT0 清零	
清零条件 (STCF=0)		设置条件 (STCF=1)
<ul style="list-style-type: none"> 通过 STT0=1 清零 IICE0=0 时 (操作停止) 复位 		<ul style="list-style-type: none"> 当禁止通信预约时 (IICRSV=1), 未能产生起始条件 且 STT0 被清零

IICBSY	I ² C 总线状态标志	
0	总线释放状态 (当 STCEN0=1 时的通信初始状态)	
1	总线通信状态 (当 STCEN0=0 时的通信初始状态)	
清零条件 (IICBSY=0)		设置条件 (IICBSY=1)
<ul style="list-style-type: none"> 检测到停止条件 IICE0=0 时 (操作停止) 复位 		<ul style="list-style-type: none"> 检测到起始条件 当 STCEN=0 时设置 IICE0

STCEN	允许初始启动触发	
0	允许操作后 (IICE0=1), 可以在检测到停止条件的基础上产生一个起始条件。	
1	允许操作后 (IICE0=1), 可以在不检测到停止条件的基础上产生一个起始条件。	
清零条件 (STCEN=0)		设置条件 (STCEN=1)
<ul style="list-style-type: none"> 检测到停止条件 复位 		<ul style="list-style-type: none"> 通过指令进行设置

IICRSV	通信预约功能禁止位	
0	允许通信预约	
1	禁止通信预约	
清零条件 (IICRSV=0)		设置条件 (IICRSV=1)
<ul style="list-style-type: none"> 通过指令进行清零 复位 		<ul style="list-style-type: none"> 通过指令进行设置

注 第 6 位和第 7 位是只读的。

- 注意事项
1. 仅当操作停止(IICE0 = 0)时才可对 STCEN 进行写操作。
 2. 当 STCEN = 1 时不论总线处于何种状态, 总线释放状态(IICBSY = 0)都可以识别, 因此当第一个起始条件产生(STT0 = 1)时, 必须确保没有第三方通信, 以防通信被破坏。
 3. 仅当操作停止(IICE0 = 0)时才可对 IICRSV 进行写操作。

备注

STT0: IIC 控制寄存器 0 (IICC0) 的第 1 位
 IICE0: IIC 控制寄存器 0 (IICC0) 的第 7 位

(4) IIC 时钟选择寄存器 0 (IICCL0)

此寄存器用来为 I²C 总线设置传送时钟。

可由 1 位或 8 位的存储器操作指令设置 IICCL0。但是，CLD0 和 DAD0 位是只读的。应根据 IIC 功能扩展寄存器 0 (IICX0) 的第 0 位 (CLX0) 设置 SMC0、CL01 和 CL00 位 (详见 16.3 (6) I²C 传送时钟设置方法)。

IIC 控制寄存器 0 (IICCO) 的第 7 位 (IICE0) 为 0 时设置 IICCL0。

复位信号的产生将 IICCL0 设置为 00H。

图 16-8. IIC 时钟选择寄存器 0 (IICCL0) 的格式 (1/2)

地址: FFA8H 复位后: 00H R/W ^注

符号	7	6	<5>	<4>	<3>	<2>	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	CL01	CL00

CLD0	SCL0 引脚电平的检测(仅当 IICE0 = 1 时有效)
0	SCL0 引脚在低电平被检测。
1	SCL0 引脚在高电平被检测。
清零条件 (CLD0 = 0)	
<ul style="list-style-type: none"> • SCL0 引脚为低电平时 • IICE0 = 0 (操作条件) 时 • 复位 	
设置条件 (CLD0 = 1)	
<ul style="list-style-type: none"> • SCL0 引脚为高电平时 	

DAD0	SDA0 引脚电平的检测(仅当 IICE0 = 1 时有效)
0	SDA0 引脚在低电平被检测
1	SDA0 引脚在高电平被检测
清零条件 (DAD0 = 0)	
<ul style="list-style-type: none"> • SDA0 引脚为低电平时 • IICE0 = 0 (操作条件) 时 (复位) 	
设置条件 (DAD0 = 1)	
<ul style="list-style-type: none"> • SDA0 引脚为高电平时 	

注 第 4 位和第 5 位是只读的。

备注 IICE0: IIC 控制寄存器 0 (IICCO) 的第 7 位

图 16-8. IIC 时钟选择寄存器 0 (IICCL0) 的格式 (2/2)

SMC0	操作模式切换
0	标准模式
1	高速模式

DFC0	数字滤波器操作控制
0	关闭数字滤波器
1	打开数字滤波器
数字滤波器只能用于高速模式	
在高速模式下, 不论 DFC0 位如何设置(设置(1)/清除(0)), 传送时钟不会发生变化。	
数字滤波器用于高速模式下的噪声消除	

(5) IIC 功能扩展寄存器 0 (IICX0)

此寄存器用于设置 I²C 的功能扩展。

可由 1 位或 8 位存储器操作指令设置 IICX0。根据 IIC 时钟选择寄存器 0(IICCL0)的第 3 位、第 1 位和第 0 位 (SMC0、CL01 和 CL00)设置 CLX0 (详见 16.3 (6) I²C 传送时钟设置方法)。

IIC 控制寄存器 0(IICC0)的第 7 位(IICE0)为 0 时设置 IICX0。

复位信号的产生将 IICX0 设置为 00H。

图 16-9. IIC 功能扩展寄存器 0 (IICX0) 的格式

地址: FFA9H	复位后: 00H	R/W						
符号	7	6	5	4	3	2	1	<0>
IICX0	0	0	0	0	0	0	0	CLX0

(6) I²C 传送时钟设置方法

I²C 传送时钟的频率 (f_{SCL}) 通过下列表达式计算。

$$f_{SCL} = 1/(m \times T + t_r + t_f)$$

m = 12, 18, 24, 44, 66, 86 (详见表 16-2 选择时钟设置)

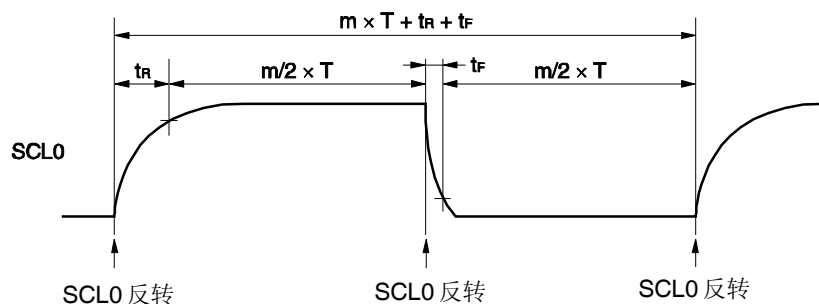
T: 1/f_w

t_r: SCL0 上升时间

t_f: SCL0 下降时间

举例来说, 当 $f_w = f_{PRS}/2 = 4.19 \text{ MHz}$ 、 $m = 86$ 、 $t_r = 200 \text{ ns}$ 和 $t_f = 50 \text{ ns}$ 时, I^2C 传送时钟频率(f_{SCL})通过下列表达式计算。

$$f_{SCL} = 1/(88 \times 238.7 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 48.1 \text{ kHz}$$



通过 IIC 时钟选择寄存器 0(IICCL0)的第 3 位、第 1 位和第 0 位 (SMC0、CL01 和 CL00)与 IIC 功能扩展寄存器 0(IICX0)的第 0 位(CLX0)设置选择时钟。

表 16-2. 选择时钟设置

IICX0 第 0 位 CLX0	IICCL0			选择时钟 (f_w)	传送时钟 (f_w/m)	可设置的 选择时钟(f_w)范围	操作模式
	第 3 位 SMC0	第 1 位 CL01	第 0 位 CL00				
0	0	0	0	$f_{PRS}/2$	$f_w/44$	2.00 ~ 4.19 MHz	正常模式 (SMC0 位 = 0)
0	0	0	1	$f_{PRS}/2$	$f_w/86$	4.19 ~ 8.38 MHz	
0	0	1	0	$f_{PRS}/4$	$f_w/86$		
0	0	1	1	禁止设置			
0	1	0	×	$f_{PRS}/2$	$f_w/24$	4.00 ~ 8.38 MHz	高速模式 (SMC0 位 = 1)
0	1	1	0	$f_{PRS}/4$	$f_w/24$		
0	1	1	1	禁止设置			
1	0	×	×				
1	1	0	×	$f_{PRS}/2$	$f_w/12$	4.00 ~ 4.19 MHz	高速模式 (SMC0 位 = 1)
1	1	1	0	$f_{PRS}/4$	$f_w/12$		
1	1	1	1	禁止设置			

注意事项 允许操作(通过设置 IIC 控制寄存器 0(IICC0)的第 7 位(IICE0)为 1 实现)前, 使用 CLX0、SMC0、CL01 和 CL00 确定 I^2C 的传送时钟频率。若改变传送时钟频率, 必须把 IICE0 清零。

备注

1. ×: 不必考虑
2. f_{PRS} : 外围硬件时钟频率

(7) 端口模式寄存器 6 (PM6)

此寄存器可以按位设置端口 6 的输入/输出操作模式。

当 P60/SCL0 引脚用于时钟 I/O 同时将 P61/SDA0 引脚用于串行数据 I/O 时，将 PM60 和 PM61 清零，且将 P60 和 P61 的输出锁存器清零。

在设置输出模式之前应将 IICE0 (IIC 控制寄存器 0 (IICC0) 的第 7 位) 设为 1，这是因为当 IICE0 为 0 时，P60/SCL0 和 P61/SDA0 引脚输出低电平(恒定)。

可由 1 位或 8 位存储器操作指令设置 PM6。

复位信号的产生将 PM6 设置为 FFH。

图 16-10. 端口模式寄存器 6 (PM6) 的格式

地址: FF26H 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	PM62	PM61	PM60

PM61	SDA0 引脚 I/O 模式选择
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

PM60	SCL0 引脚 I/O 模式选择
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

注意事项 复位释放后，确保第 2 位设置为 0。

16.4 I²C 总线模式功能

16.4.1 引脚配置

串行时钟引脚(SCL0)和串行数据总线引脚(SDA0)配置如下。

(1) SCL0..... 此引脚用于串行时钟的输入和输出。

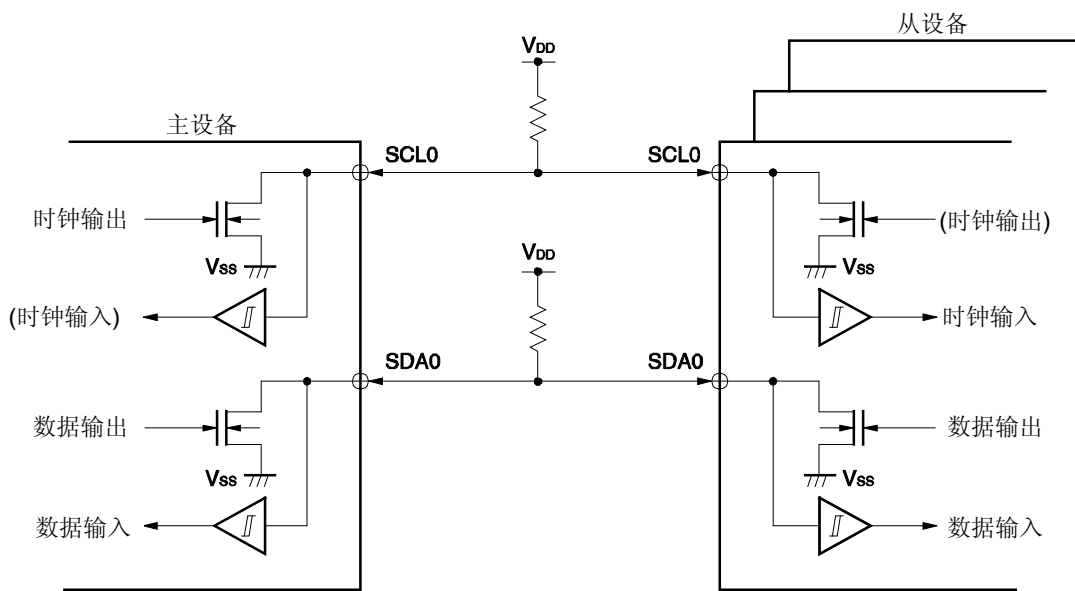
此引脚是主设备和从设备的 N-ch 漏极开路输出。该输入是施密特(Schmitt)输入。

(2) SDA0 此引脚用于串行数据的输入和输出。

此引脚是主设备和从设备的 N-ch 漏极开路输出。该输入是施密特(Schmitt)输入。

由于串行时钟线和串行数据总线的输出都是 N-ch 漏极开路输出，因此需要一个外部上拉电阻。

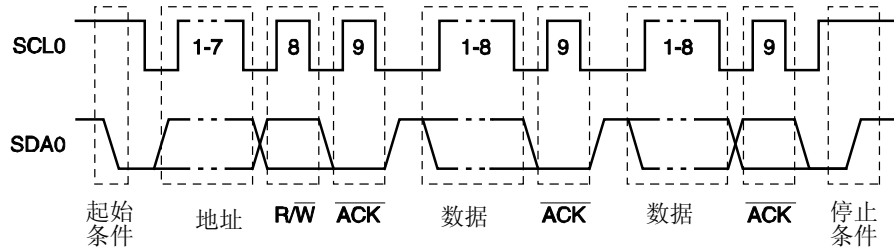
图 16-11. 引脚配置框图



16.5 I²C 总线定义和控制方法

以下描述了 I²C 总线的串行数据通信格式和 I²C 总线使用的信号。图 16-12 显示了通过 I²C 串行数据总线的“起始条件”，“地址”，“数据”和“停止条件”等信号输出时的传送时序。

图 16-12. I²C 总线串行数据传送时序



主设备产生起始条件、从设备地址和停止条件。

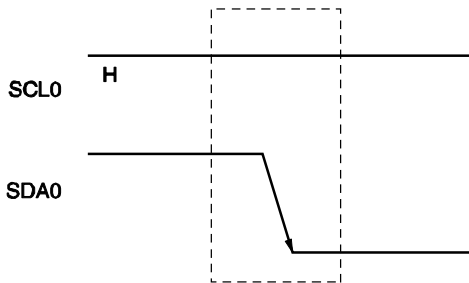
应答信号($\overline{\text{ACK}}$)可以由主设备或者从设备产生(通常,由接收 8 位数据的设备输出)。

主设备持续输出串行时钟 ($\overline{\text{SCL0}}$)。但在从设备中,可以延长 SCL0 的低电平阶段同时可以插入等待时间。

16.5.1 起始条件

SCL0 引脚处于高电平且 SDA0 引脚从高电平变为低电平时,满足起始条件。当启动串行传送时, SCL0 引脚和 SDA0 引脚的起始条件是由主设备产生并发给从设备的信号。当使用从设备时,可以检测到起始条件。

图 16-13. 起始条件



检测到停止条件(SPD0: IIC 状态寄存器 0 (IICS0)的第 0 位=1)后, IIC 控制寄存器 0(IICC0)的第 1 位(STT0)被设置(为 1)时,输出一个起始条件。检测到起始条件时, IICS0 的第 1 位(STD0)被设置(为 1)。

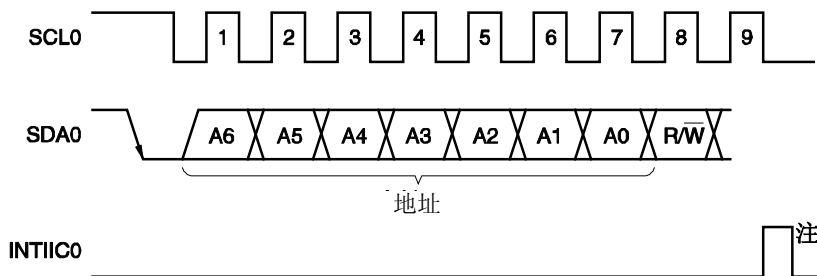
16.5.2 地址

地址由起始条件之后的 7 位数据指定。

一个地址是一个输出的 7 位数据段，用于选择一个通过总线与主设备相连的从设备。因此，通过总线相连的每个从设备必须有一个唯一的地址。

从设备包括具有以下功能的硬件：检测起始条件，检查 7 位地址数据是否与存储在从设备地址寄存器 0(SVA0)中的数据相等。如果地址数据和 SVA0 的值相等，则该从设备将被选中且与主设备通信，直到主设备产生一个起始条件或停止条件。

图 16-14. 地址



注 如果在从设备操作期间接收到的是数据而不是本地地址或扩展码时，则 INTIIC0 不会发出。

将从设备地址和第 8 位(用于指示传送方向(如 16.5.3 传送方向指示中所描述的))一起写入 IIC 移位寄存器 0(IIC0)并且输出。接收到的地址写入 IIC0。

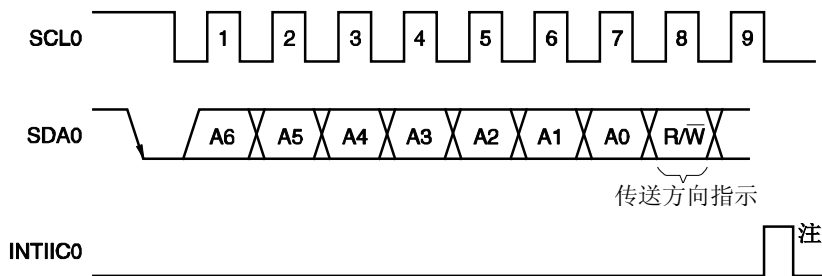
将从设备地址分配给 IIC0 的高 7 位。

16.5.3 传送方向指示

除了 7 位地址数据之外，主设备还发送 1 位数据用于指示传送方向。

当传送方向指示位的值为“0”时，表示主设备正向从设备传送数据。当传送输方向指示位的值为“1”时，表示主设备正接收来自从设备的数据。

图 16-15. 传送方向指示



注 如果在从设备操作期间接收到的是数据而不是本地地址或扩展码时，则 INTIIC0 不会发出。

16.5.4 应答信号 ($\overline{\text{ACK}}$)

$\overline{\text{ACK}}$ 信号用于检测处于发送方和接收方的串行数据的状态。

当接收方收到 8 位数据后将返回一个 $\overline{\text{ACK}}$ 信号。

发送方通常在发送了 8 位数据后接收到 $\overline{\text{ACK}}$ 信号。当 $\overline{\text{ACK}}$ 信号从接收方返回时，认为已正确接收并继续之后的处理。是否检测到 $\overline{\text{ACK}}$ 信号可以通过 IIC 状态寄存器 0(IICS0)的第 2 位(ACKD0)进行检测。

主设备接收到最后一个数据项时，不再返回 $\overline{\text{ACK}}$ 信号而是产生一个停止条件。如果从设备在接收到数据后未返回 $\overline{\text{ACK}}$ 信号，主设备将输出一个停止条件或重启条件并且停止传送。如果未返回 $\overline{\text{ACK}}$ 信号，可能是如下原因。

- <1> 接收操作没有正常进行。
- <2> 接收到最后一个数据项。
- <3> 由地址指示的接收方不存在。

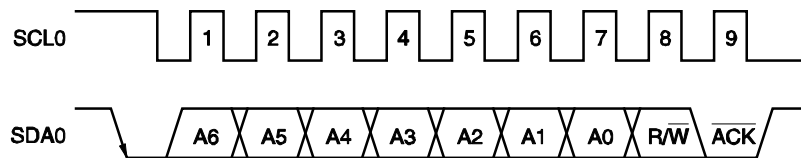
要产生 $\overline{\text{ACK}}$ 信号，接收方必须在第 9 个时钟时使 SDA0 线变为低电平(表示正常接收)。

可通过设置 IIC 控制寄存器 0(IICC0)的第 2 位(ACKE0)为 1，自动产生 $\overline{\text{ACK}}$ 信号。IICS0 寄存器的第 3 位(TRC0)可被数据的第 8 位(紧接着 7 位地址信息)设置。通常，设置 ACKE0=1 用于接收 (TRC0 = 0)。

如果一个从设备在接收(TRC0 = 0)期间不再收到任何数据或者不再需要下一个数据项，则从设备必须通过将 ACKE0 清零通知主设备，这样它就不会再收到任何数据。

接收(TRC0 = 0)期间，当主设备不需要下一个数据项时，必须将 ACKE0 清零，这样就不会产生 $\overline{\text{ACK}}$ 。在这种方式下，主设备通知作为发送方的从设备，不再需要其他数据(发送过程将停止)。

图 16-16. $\overline{\text{ACK}}$ 信号



接收到本地地址时，无论 ACKE0 的值是多少，都将自动产生 $\overline{\text{ACK}}$ 信号。当接收到一个非本地地址时，则不产生 $\overline{\text{ACK}}$ 信号(NACK)。

接收到扩展码时，如果 ACKE0 预先设为 1，则将产生 $\overline{\text{ACK}}$ 信号。

接收到数据时如何产生 $\overline{\text{ACK}}$ 信号，将根据等待时序的设置而变化。

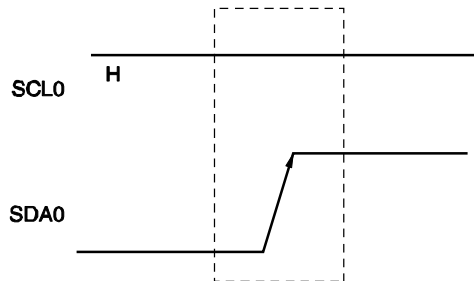
- 当选择 8 时钟等待状态时 (IICC0 寄存器的第 3 位(WTIM0)= 0):
退出等待状态前通过将 ACKE0 设为 1，在 SCL0 引脚第 8 个时钟的下降沿产生 $\overline{\text{ACK}}$ 信号。
- 当选择 9 时钟等待状态时 (IICC0 寄存器的第 3 位(WTIM0)= 1):
通过预先设置 ACKE0 为 1 产生 $\overline{\text{ACK}}$ 信号。

16.5.5 停止条件

SCL0 引脚处于高电平时，将 SDA0 引脚从低电平变为高电平可以产生一个停止条件。

停止条件是一个信号，它在串行传送操作结束时由主设备产生并发送给从设备。当作为从设备使用时，可以检测到停止条件。

图 16-17. 停止条件



IIC 控制寄存器 0(IICC0)的第 0 位(SPT0)设为 1 时，会产生一个停止条件。当检测到停止条件时，IIC 状态寄存器 0(IICS0)的第 0 位(SPD0)被设为 1 且当 IICC0 的第 4 位(SPIE0)被设为 1 时产生 INTIIC0。

16.5.6 等待

等待状态用来通知通信另一方，设备(主设备或从设备)正准备发送或者接收数据(即处于等待状态)。

设置 SCL0 引脚为低电平，通知通信另一方设备已进入等待状态。当主设备和从设备的等待状态被取消时，下一个数据传送才能开始。

图 16-18. 等待 (1/2)

- (1) 当主设备有 9 个时钟等待而从设备有 8 个时钟等待时
(主设备发送，从设备接收，同时 ACKE0 = 1)

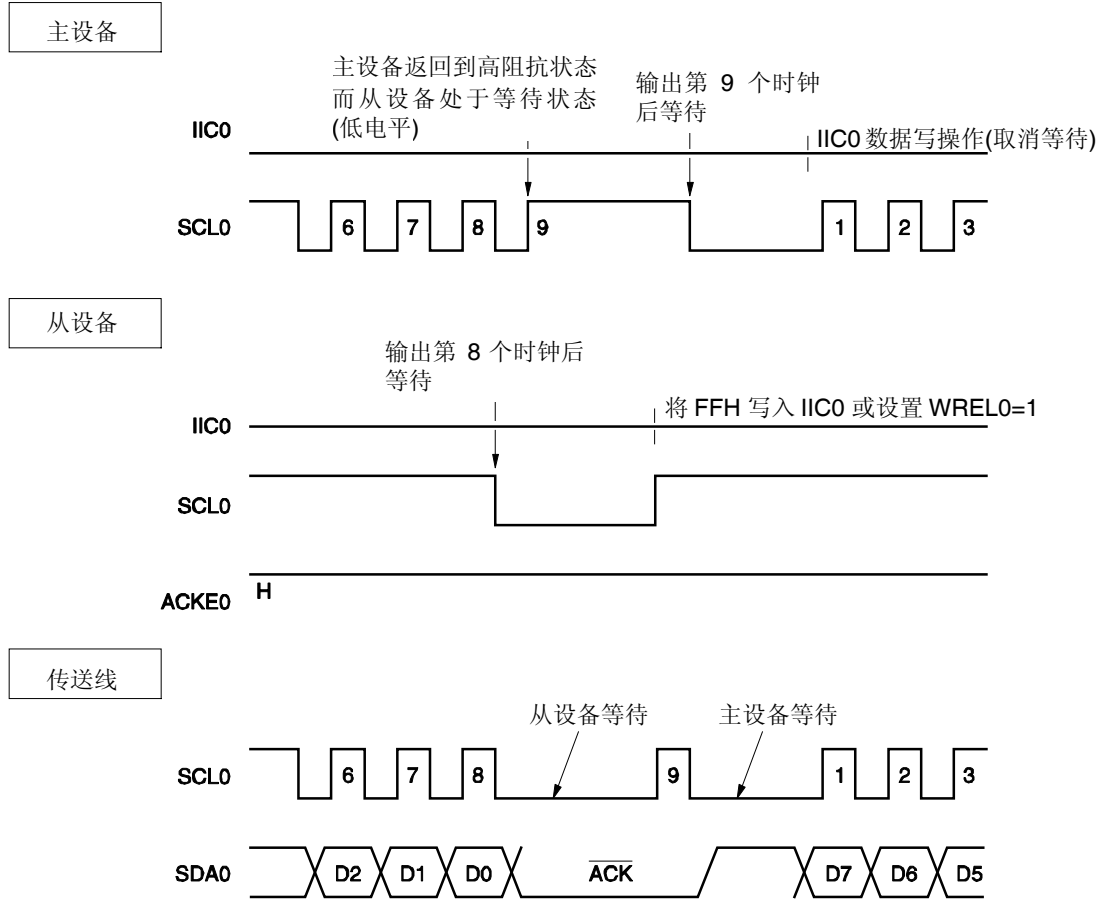
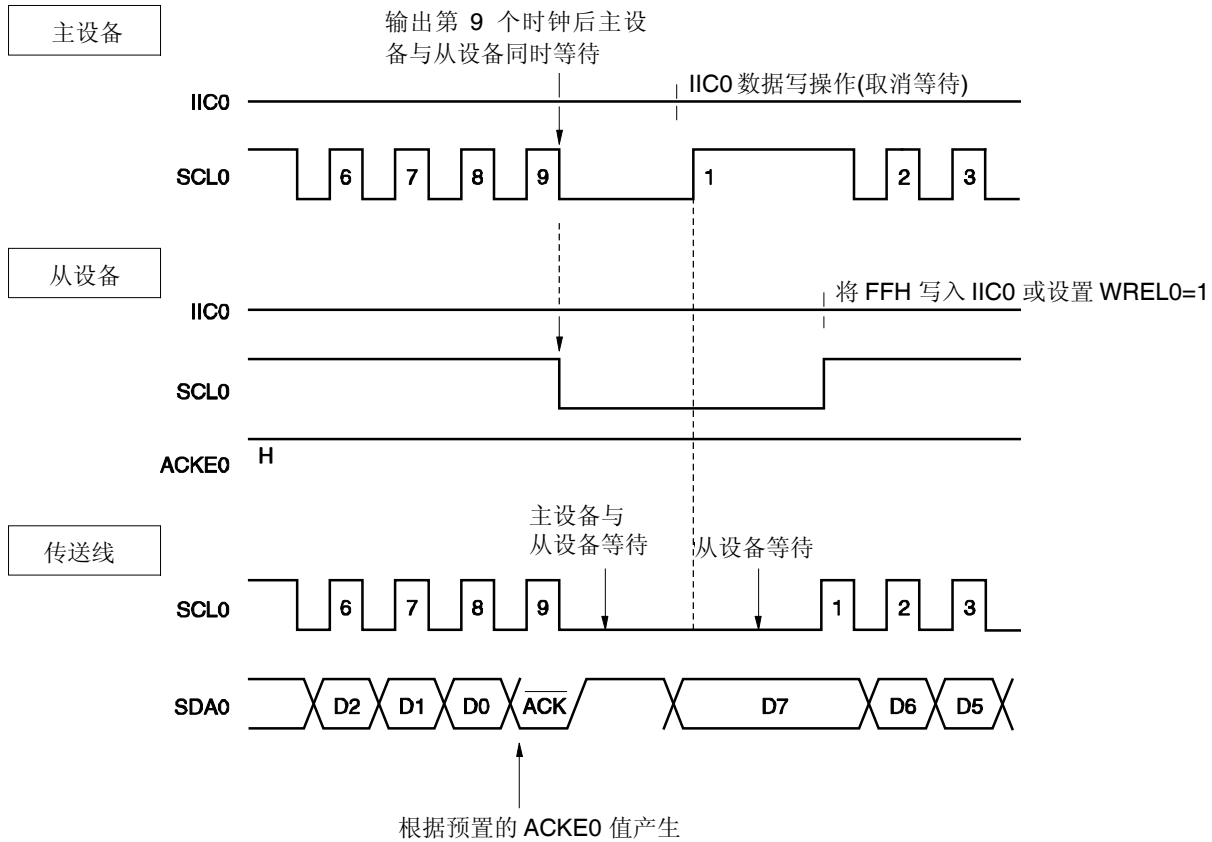


图 16-18. 等待 (2/2)

(2) 当主设备和从设备都有 9 个时钟后等待时
(主设备发送, 从设备接收, 同时 ACKE0 = 1)



备注 ACKE0: IIC 控制寄存器 0(IICC0)的第 2 位
WRELO: IIC 控制寄存器 0(IICC0)的第 5 位

根据 IIC 控制寄存器 0(IICC0)的第 3 位(WTIM0)的设置, 自动产生等待状态。

通常, 当 IICC0 的第 5 位(WRELO)设置为 1 时或者在将 FFH 写入 IIC 移位寄存器 0(IIC0)时, 接收方取消等待状态, 而当数据写入 IIC0 时, 发送方取消等待状态。

主设备也能通过以下任何一种方法取消等待状态。

- 通过把 IICC0 的第 1 位(STT0)设为 1
- 通过把 IICC0 的第 0 位(SPT0)设为 1

16.5.7 取消等待

I²C 通常通过如下处理方式取消一个等待状态。

- 把数据写入 IIC 移位寄存器 0(IIC0)
- 设置 IIC 控制寄存器 0(IICC0)的第 5 位(WRELO) (取消等待)
- 设置 IICC0 寄存器的第 1 位(STT0) (产生起始条件)[#]
- 设置 IICC0 寄存器的第 0 位(SPT0) (产生停止条件)[#]

注 仅用于主设备

当执行以上取消等待处理过程时，I²C 取消等待状态且恢复通信。

通过向 IIC0 写数据来取消等待状态并发送数据 (包括地址)。

通过设置 IIC 控制寄存器 0(IICC0)的第 5 位(WRELO)为 1，可在取消等待状态之后接收数据，或完成数据发送操作。

设置 IICC0 的第 1 位(STT0)为 1，可在取消一个等待状态之后产生一个重启条件。

设置 IICC0 的第 0 位(SPT0)为 1，可在取消一个等待状态之后产生一个停止条件。

对于一个等待状态只能执行一次取消操作。

例如，如果通过设置 WRELO 为 1 在取消等待状态之后将数据写入 IIC0，则可能将一个不正确的值输出到 SDA0，这是因为改变 SDA0 线的时序和 IIC0 写操作时序相冲突。

除此以外，如果通信已放弃时如果 IICE0 被清零，则停止通信，这样可以取消等待状态。

如果 I²C 总线由于噪音产生死锁，则通过设置 IICC0 的第 6 位(LRELO)可以保存通信处理内容，这样等待状态便可以取消。

16.5.8 中断请求 (INTIIC0) 产生时序和等待控制

IIC 控制寄存器 0(IICC0)的第 3 位(WTIM0)的设置决定 INTIIC0 产生时序和相应的等待控制，如表 16-3 所示。

表 16-3. INTIIC0 产生时序和等待控制

WTIM0	从设备操作期间			主设备操作期间		
	地址	数据接收	数据发送	地址	数据接收	数据发送
0	9 ^{#1,2}	8 ^{#2}	8 ^{#2}	9	8	8
1	9 ^{#1,2}	9 ^{#2}	9 ^{#2}	9	9	9

注 1. 仅当有地址与从设备地址寄存器 0(SVA0)设置的地址相等时，在第 9 个时钟的下降沿产生从设备的 INTIIC0 信号和等待周期。

此时，不论 IICC0 的第 2 位(ACKE0)设置为何值，都将产生 $\overline{\text{ACK}}$ 信号。对于一个接收到扩展码的从设备，在第 8 个时钟的下降沿将产生 INTIIC0。

但是，如果重启之后地址不相等，将在第 9 个时钟的下降沿产生 INTIIC0，但不会产生等待周期。

2. 如果接收地址与从设备地址寄存器 0(SVA0)的内容不相等，同时也未接收到扩展码，则 INTIIC0 和等待周期都不会产生。

备注 表中的数字是指串行时钟的时钟信号个数。中断请求和等待控制和这些时钟信号的下降沿同步。

(1) 地址发送/接收期间

- 从设备操作: 中断和等待时序取决于以上注 1、2 中所描述的情况, 与 WTIMO 位无关。
- 主设备操作: 中断和等待时序在第 9 个时钟的下降沿发生, 与 WTIMO 位无关。

(2) 数据接收期间

- 主/从设备操作: 中断和等待时序取决于 WTIMO 位。

(3) 数据发送期间

- 主/从设备操作: 中断和等待时序取决于 WTIMO 位。

(4) 等待取消的方法

四种取消等待方法如下所示。

- 向 IIC 移位寄存器 0(IIC0)写入数据
- 设置 IIC 控制寄存器 0(IICC0)的第 5 位 (WREL0) (取消等待)
- 设置 IICC0 寄存器的第 1 位 (STT0) (产生起始条件)^注
- 设置 IICC0 寄存器的第 0 位 (SPT0) (产生停止条件)^注

注 仅用于主设备。

当选择了一个 8-时钟等待时(WTIMO = 0), 必须先决定是否产生应答信号, 再考虑取消等待。

(5) 停止条件的检测

当检测到一个停止条件时(仅当 SPIE0 = 1), INTIIC0 将会产生。

16.5.9 地址相等的检测方法

在 I²C 总线模式下, 主设备可以通过发送的从设备地址来选择指定的从设备。

可以通过硬件自动检测到地址相等。当将一个本地地址设置到从设备地址寄存器 0(SVA0)中且该地址与主设备发送的从设备地址相等时, 或者接收到扩展码时, 都将产生一个中断请求(INTIIC0)。

16.5.10 错误检测

I²C 总线模式下, 在数据发送期间通过发送设备的 IIC 移位寄存器 (IIC0) 捕获串行数据总线 (SDA0) 的状态, 因此发送前的 IIC0 数据可以用来和已发送的 IIC0 数据比较, 以此来实现对发送错误的检测。当比较值不相等时, 可以判断存在一个发送错误。

16.5.11 扩展码

(1) 当接收地址的高 4 位为“0000”或“1111”时，代表接收到扩展码的扩展码接收标志(EXC0)被设为 1，并且在第 8 个时钟的下降沿产生中断请求(INTIIC0)。而存放在从设备地址寄存器 0(SVA0)中的本地地址不受影响。

(2) 如果由一个 10 位地址传送设置 SVA0 为“11110xx0”且主设备传送的值为“11110xx0”，则其结果如下所示。注意在第 8 个时钟的下降沿产生 INTIIC0。

- 数据的高 4 位相等: EXC0 = 1
- 7 位数据相等: COI0 = 1

备注 EXC0: IIC 状态寄存器 0(IICS0)的第 5 位
 COI0: IIC 状态寄存器 0(IICS0)的第 4 位

(3) 由于中断请求发生之后的处理过程随扩展码之后的数据而有所不同，因此这样的处理过程由软件来执行。

如果当一个从设备正在操作时接收到扩展码，那么该从设备(即使它的地址不匹配)也将参与通信。

例如，接收到扩展码后，如果不希望目标设备作为从设备来操作，则将 IIC 控制寄存器 0(IICC0)的第 6 位(LREL0)设为 1，为下一个通信操作设置待机模式。

表 16-4. 扩展码位定义

从设备地址	R/W 位	描述
0 0 0 0 0 0 0	0	通用调用地址
0 0 0 0 0 0 0	1	起始字节
0 0 0 0 0 0 1	×	C 总线地址
0 0 0 0 0 1 0	×	为不同总线格式预约的地址
1 1 1 1 0 X X	×	10 位从设备地址规范

16.5.12 仲裁

当多个主设备同时产生起始条件信号时 (在 STD0 设为 1 之前将 STT0 设为 1)，调整时钟数目直至传送数据改变，然后执行主设备间的通信。这种操作称为仲裁。

当有一个主设备仲裁失败时，可通过仲裁失败产生的时序设置 IIC 状态寄存器 0(IICS0)中的仲裁失败标志(ALD0)为 1，同时 SCL0 和 SDA0 线变为高阻抗，从而释放总线。

基于下一个中断请求的时序(第 8 个或第 9 个时钟，检测到停止条件时等)以及由软件设置 ALD0=1 来检测仲裁失败。有关中断请求时序，详见 16.5.17 I²C 中断请求(INTIIC0)产生时序。

备注 STD0: IIC 状态寄存器 0 (IICS0) 的第 1 位
 STT0: IIC 控制寄存器 0 (IICC0) 的第 1 位

图 16-19. 仲裁时序示例

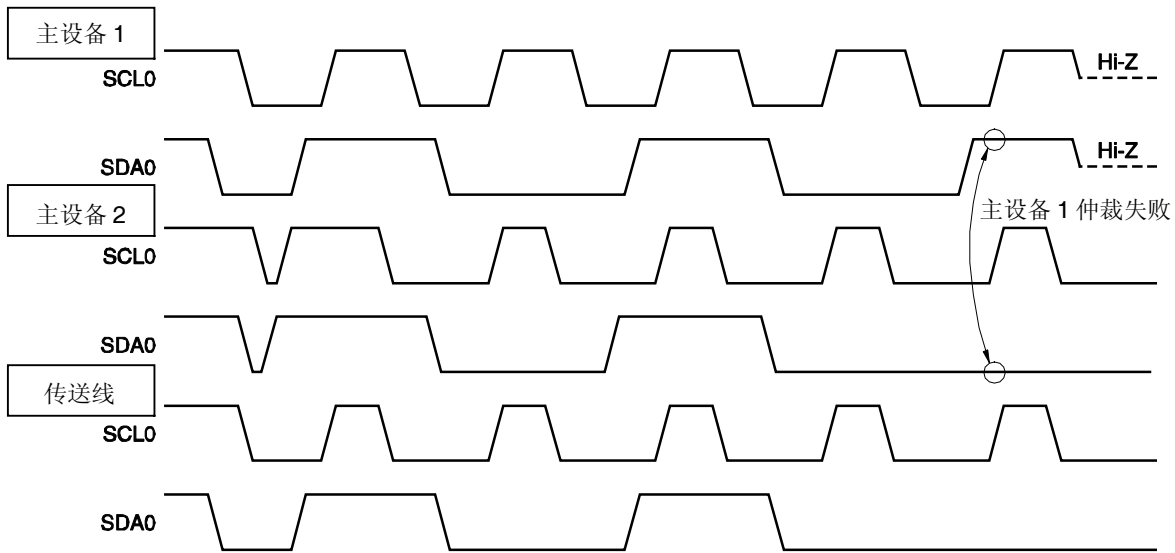


表 16-5. 仲裁期间的状态和中断请求产生时序

仲裁期间的状态	中断请求产生时序
地址发送期间	字节传送后第 8 个或第 9 个时钟的下降沿 ^{注1}
地址发送后读/写数据	
扩展码发送期间	
扩展码发送后读/写数据	
数据发送期间	
数据发送之后的 ACK 信号传送期间	
数据传送期间检测到重启条件时	
数据传送期间检测到停止条件时	停止条件产生时 (当 SPIE0 = 1) ^{注2}
准备产生重启条件时数据为低电平	字节传送后第 8 个或第 9 个时钟的下降沿 ^{注1}
准备产生重启条件时检测到停止条件	
准备产生停止条件时数据为低电平	停止条件产生时 (当 SPIE0 = 1) ^{注2}
准备产生停止条件时 SCL0 为低电平	
准备产生重启条件时 SCL0 为低电平	字节传送后第 8 个或第 9 个时钟的下降沿 ^{注1}

- 注**
1. 当 WTIM0 (IIC 控制寄存器 0(IICC0)的第 3 位) = 1 时,将在第 9 个时钟的下降沿产生一个中断请求。当 WTIM0 = 0 且扩展码的从设备地址已被接收时, 将在第 8 个时钟的下降沿产生一个中断请求。
 2. 如果可以产生仲裁, 则设置 SPIE0 = 1, 进行主设备操作。

备注 SPIE0: IIC 控制寄存器 0 (IICC0)的第 4 位

16.5.13 唤醒功能

I²C 总线从设备唤醒功能是, 在接收到本地地址及扩展码时产生一个中断请求信号(INTIIC0)。

此功能通过在地址不相等时禁止产生不必要的 INTIIC0 信号, 使处理更加有效。

当检测到一个起始条件时, 设置唤醒待机模式。仲裁失败可能将主设备(已产生启动条件)转变为从设备, 如果因此而发送地址时, 则该唤醒待机模式有效。

可是, 当检测到停止条件时, 无论是否有唤醒功能, 都设置 IIC 控制寄存器 0 (IICC0) 的第 4 位(SPIE0), 这将决定是否允许或禁止中断请求。

16.5.14 通信预约

(1) 允许使用通信预约功能时(IIC 标志寄存器 0(IICF0)的第 0 位(IICRSV) = 0)

如果要启动当前不使用总线的主设备通信，可以使用通信预约功能，在总线释放时允许发送一个启动条件信号。有两种不使用总线时的模式。

- 当仲裁结果不是主/从设备操作时
- 当接收到扩展码且禁止从设备操作时(不返回 ACK 信号且 IIC 控制寄存器 0 (IICC0) 的第 6 位(LRELO)被设置为 1 时释放总线)。

如果总线未使用时(检测到停止条件后)IICC0 的第 1 位(STT0)被设为 1，则自动产生一个起始条件并且设置等待状态。

如果在 IICC0 的第 4 位(SPIE0)被设为 1 之后将地址写入 IIC 移位寄存器 0(IIC0)，并且通过总线释放(检测停止状态)产生中断请求信号(INTIIC0)来检测该地址，则设备将以主设备方式自动开始通信。在检测到停止条件前写入 IIC0 的数据是无效的。

当 STT0 已被设为 1 时，操作模式(作为起始条件或作为通信预约)取决于总线状态。

- 如果总线已被释放..... 产生一个起始条件
- 如果总线未被释放 (待机模式)..... 通信预约

STT0 设为 1 且经历了等待时间后，使用 MSTSO(IIC 状态寄存器 0 (IICS0)的第 7 位) 检查是否进行通信预约操作。

应通过软件设置的等待周期见表 16-6。

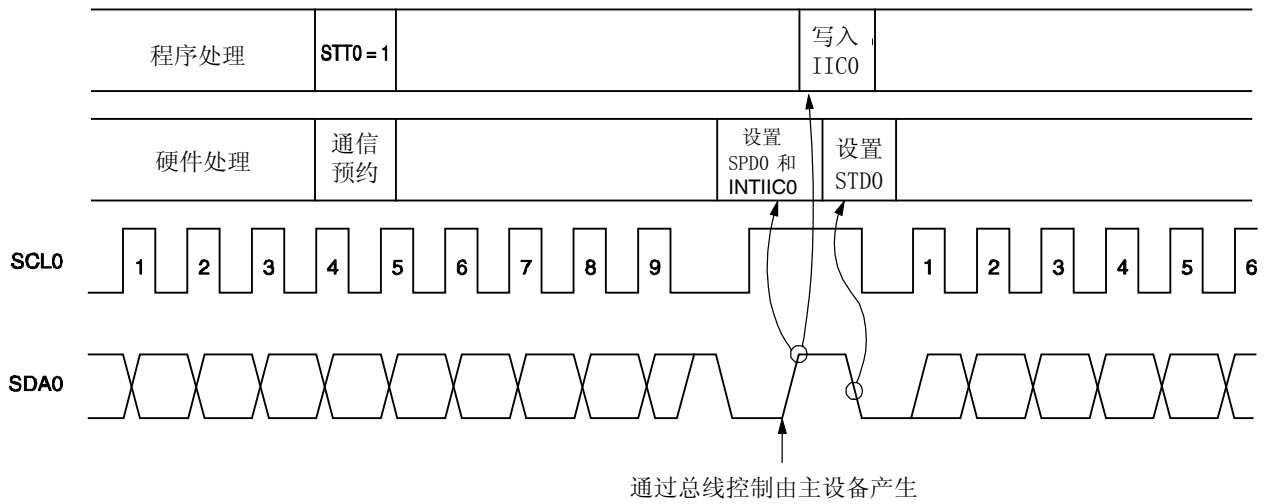
★

表 16-6. 等待周期

CLX0	SMC0	CL01	CL00	等待周期
0	0	0	0	46 个时钟
0	0	0	1	86 个时钟
0	0	1	0	172 个时钟
0	0	1	1	34 个时钟
0	1	0	0	30 个时钟
0	1	0	1	
0	1	1	0	60 个时钟
0	1	1	1	12 个时钟
1	1	0	0	18 个时钟
1	1	0	1	
1	1	1	0	36 个时钟

图 16-20 显示通信预约时序。

图 16-20. 通信预约时序



- 备注
- IIC0: IIC 移位寄存器 0
 - STT0: IIC 控制寄存器 0 (IICC0) 的第 1 位
 - STD0: IIC 状态寄存器 0 (IICS0) 的第 1 位
 - SPD0: IIC 状态寄存器 0 (IICS0) 的第 0 位

通过以下时序接受通信预约。IIC 状态寄存器 0(IICS0)的第 1 位(STD0)设为 1 之后，可以通过在检测到停止条件之前设置 IIC 控制寄存器 0(IICC0)的第 1 位(STT0)为 1 来进行通信预约。

图 16-21. 接受通信预约的时序

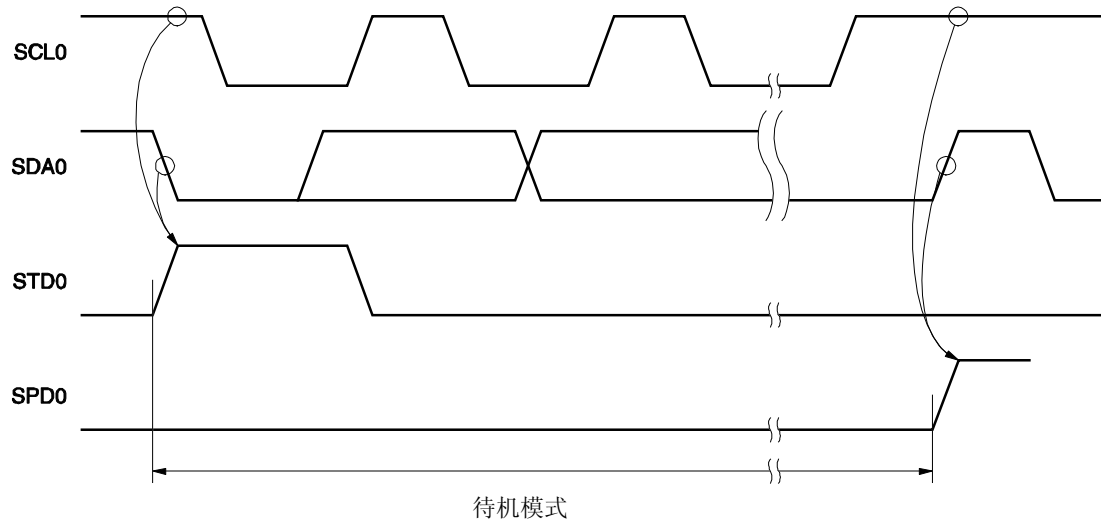
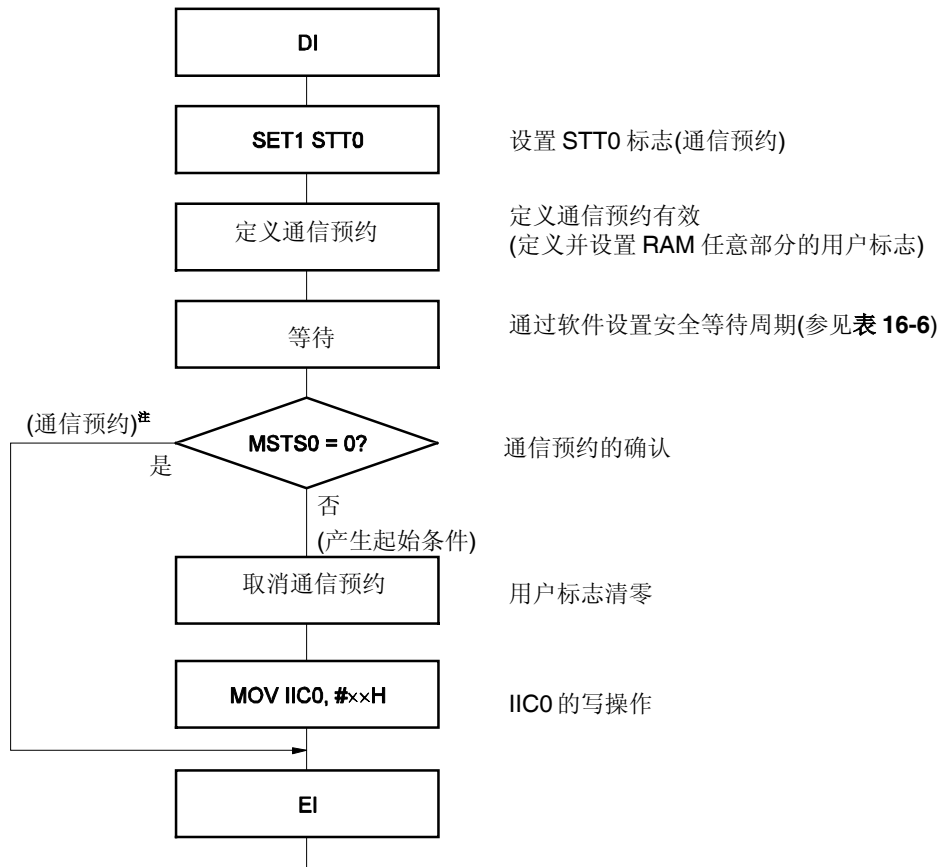


图 16-22 显示通信预约协议。

图 16-22. 通信预约协议



注 当产生一个停止条件中断请求时，通信预约执行对 IIC 移位寄存器 0(IIC0)的写操作。

备注 STT0: IIC 控制寄存器 0 (IICC0) 的第 1 位
 MSTS0: IIC 状态寄存器 0 (IICS0) 的第 7 位
 IIC0: IIC 移位寄存器 0

(2) 当禁止使用通信预约功能时 (IIC 标志寄存器 0(IICF0)的第 0 位(IICRSV) = 1)

如果在总线通信期间，总线未使用时将 IIC 控制寄存器 0 (IICC0) 的第 1 位(STT0)设置为 1，则该请求会被拒绝并且不产生起始条件。下列两种状态包括了总线未使用时的情况。

- 当仲裁结果不是主/从设备操作时
- 接收到一个扩展码且禁止从设备操作时 (不返回 $\overline{\text{ACK}}$ 信号且 IIC 控制寄存器 0 (IICC0) 的第 6 位(LREL0)被设置为 1 时释放总线)

通过检查 STCF(IICF0 的第 7 位)来确认是产生了起始条件还是请求被拒绝。设置 STT0 = 1 后，直到 STCF 被设置为 1 时，必须满足表 16-7 所示的时间。因此，必须通过软件确保等待时间。

表 16-7. 等待周期

CL01	CL00	等待周期
0	0	6 个时钟
0	1	6 个时钟
1	0	12 个时钟
1	1	3 个时钟

16.5.15 其他注意事项

(1) 当 STCEN (IIC 标志寄存器 0(IICF0)的第 1 位) = 0 时

在允许 I²C 操作(IICE0 = 1)后, 无论总线的实际状态如何都可以识别总线通信状态(IICBSY (IICF0 的第 6 位) = 1)。从检测到非停止条件模式变为主设备通信模式时, 首先产生一个停止条件以释放总线, 然后进行主设备通信。

当使用多个主设备时, 如果总线未被释放(没有检测到停止条件), 则不能进行主设备通信。
按下列步骤产生停止条件。

- <1> 设置 IIC 时钟选择寄存器 0(IICCL0)。
- <2> 设置 IIC 控制寄存器 0(IICC0) 的第 7 位(IICE0)为 1。
- <3> 设置 IICC0 的第 0 位(SPT0)为 1。

(2) 当 STCEN = 1

在允许 I²C 操作(IICE0 = 1)后, 无论总线的实际状态如何都可以识别总线释放状态(IICBSY (IICF0 的第 6 位) = 1)。要产生第一个起始条件 (STT0 (IIC 控制寄存器 0 (IICC0)的第 1 位) = 1), 必须确认总线已被释放, 以免干扰其他通信。

(3) 如果其他 I²C 通信已经进行

如果 SDA0 引脚为低电平同时 SCL0 引脚为高电平时允许 I²C 操作, 且设备已进行通信, 则 I²C 的宏可以识别 SDA0 引脚已经变为低电平(检测一个起始条件)。如果此时总线上的值被识别为一个扩展码, 则返回 ACK 信号, 但这会干扰其他 I²C 通信。为了避免这种情况, 可按下列步骤启动 I²C。

- <1> 当检测到停止条件时, 将 IICC0 的第 4 位(SPIE0)清零, 禁止产生中断请求信号(INTIIC0)。
- <2> 设置 IICC0 的第 7 位(IICE0)为 1, 允许 I²C 的操作。
- <3> 等待对启动条件的检测。
- <4> ACK 信号返回之前(设置 IICE0 为 1 之后的 4 到 80 个时钟)设置 IICC 的第 6 位 (LRELO) 为 1, 强行禁止检测。

(4) 允许操作之前(IICE0 = 1)通过使用 SMC0, CL01, CL00 (IICL0 的第 3 位, 第 1 位和第 0 位)和 CLX0 (IICX0 的第 0 位)决定传送时钟频率。要改变传送时钟频率, 必须清零 IICE0。

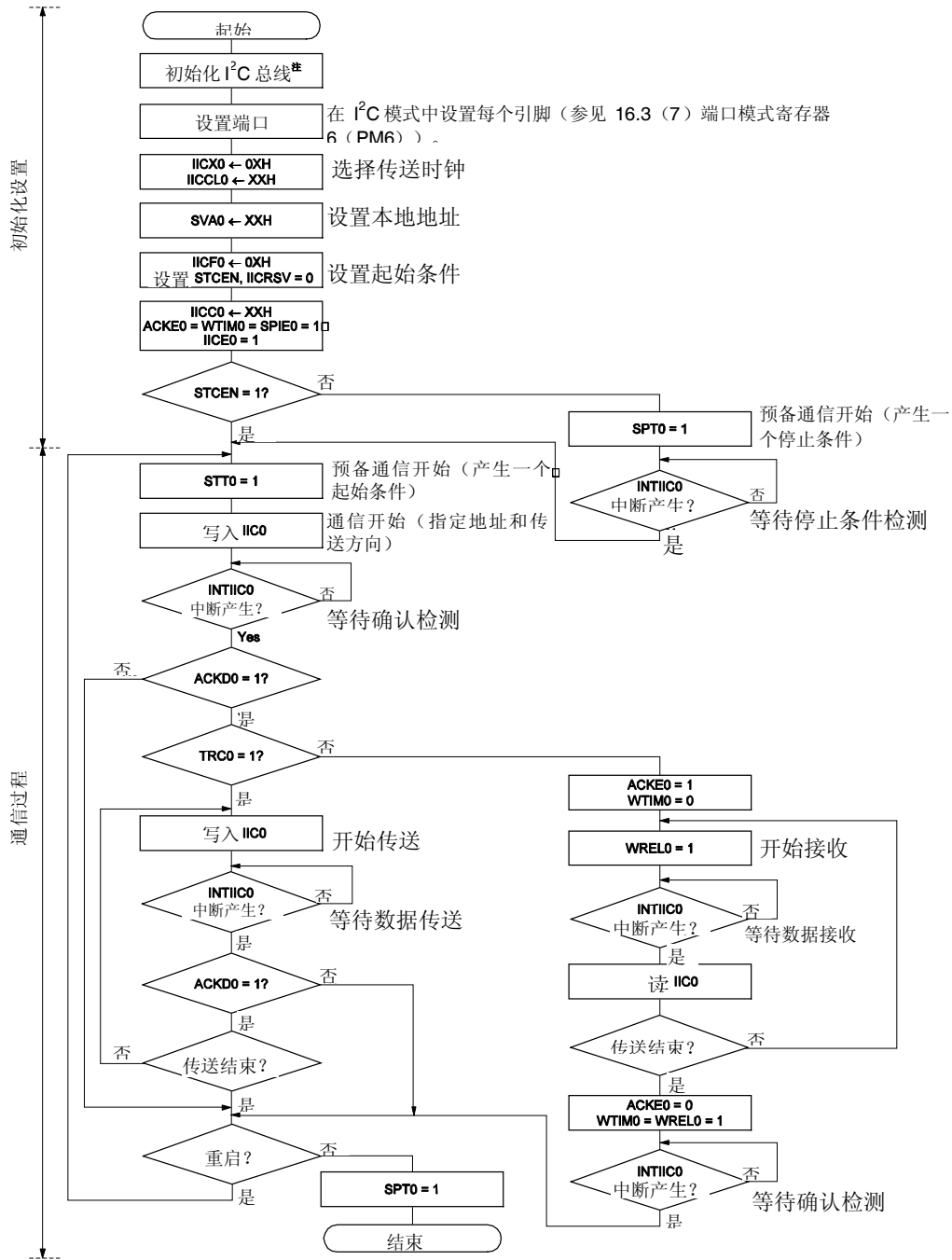
- (5) 在对 **STT0** 和 **SPT0** (**IICC0** 的第 1 位和第 0 位)设置之后和清零之前, 禁止再次设置。

- (6) 当预约发送时, 设置 **SPIE0** (**IICL0** 的第 4 位)为 1, 以允许在检测到停止条件时产生中断请求。中断请求产生后, 将通信数据写入 **IIC0**, 传送过程开始。如果检测到停止条件时不产生中断请求, 设备将处于等待状态, 因为开始通信时不产生中断请求。但是, 由软件检测 **MSTS0** (**IICS0** 的第 7 位)时不必设置 **SPIE0** 为 1。

16.5.16 通信操作

(1) 主设备操作 (单主设备系统)

图 16-23. 主设备操作流程 (单主设备系统)

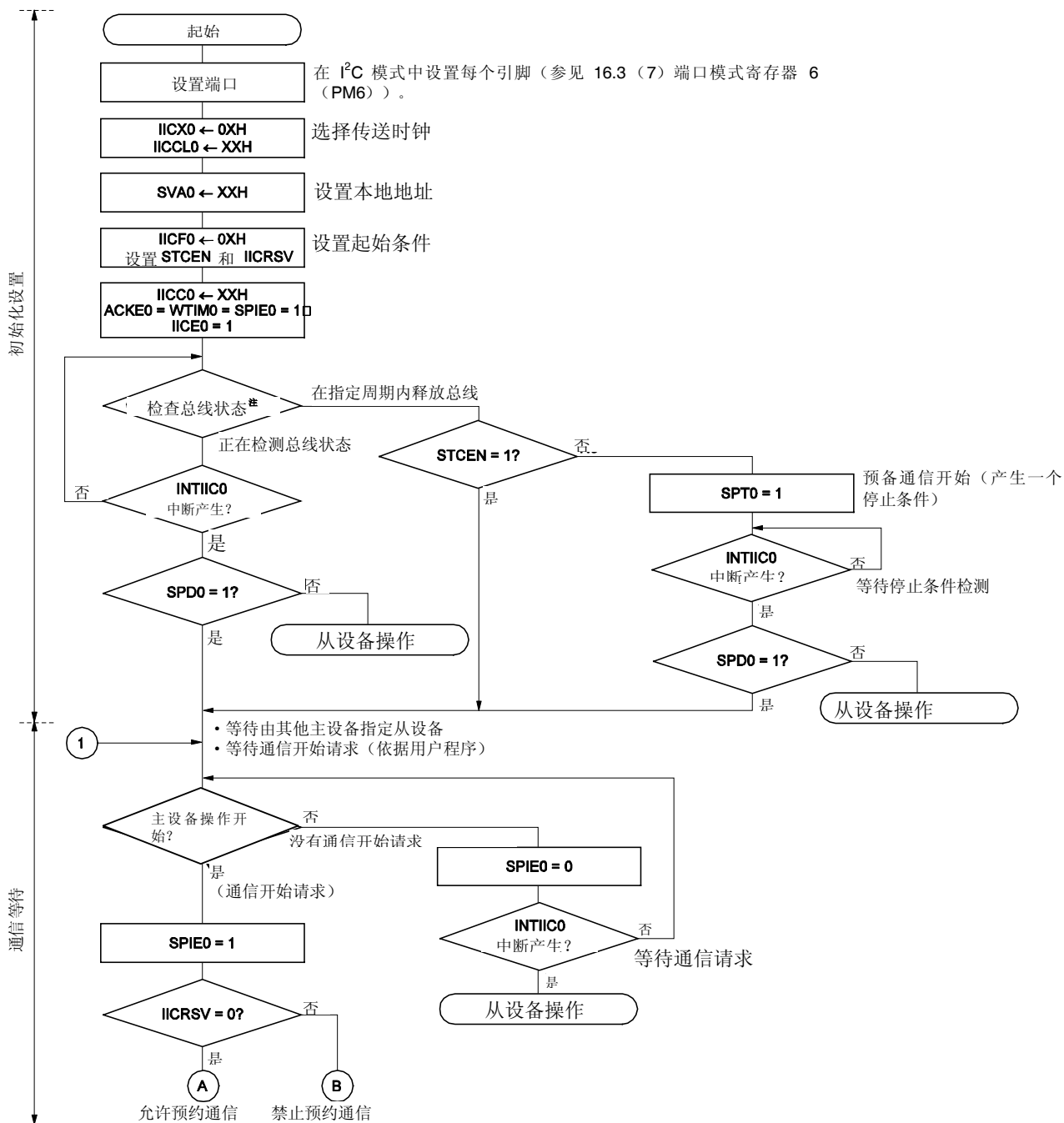


注 根据正在通信的产品规范释放(SCL0 和 SDA0 引脚 = 高电平)I²C 总线。如果 EEPROM 向 SDA0 引脚输出一个低电平，例如，设置 SCL0 引脚为输出端口模式，且从输出端口输出一个时钟脉冲直到 SDA0 引脚持续处于高电平。

备注 遵照正在通信的产品规范，注意发送和接收的格式。

(2) 主设备操作 (多主设备系统)

图 16-24. 主设备操作流程 (多主设备系统) (1/3)



注 确认总线已释放了(CLD0 位 = 1, DAD0 位 = 1)指定的一段时间(如, 一帧的周期)。如果 SDA0 引脚持续处于低电平, 可以根据正在使用的产品规范决定是否释放(SCL0 和 SDA0 引脚 = 高电平)I²C 总线。

图 16-24. 主设备操作流程 (多主设备系统) (2/3)

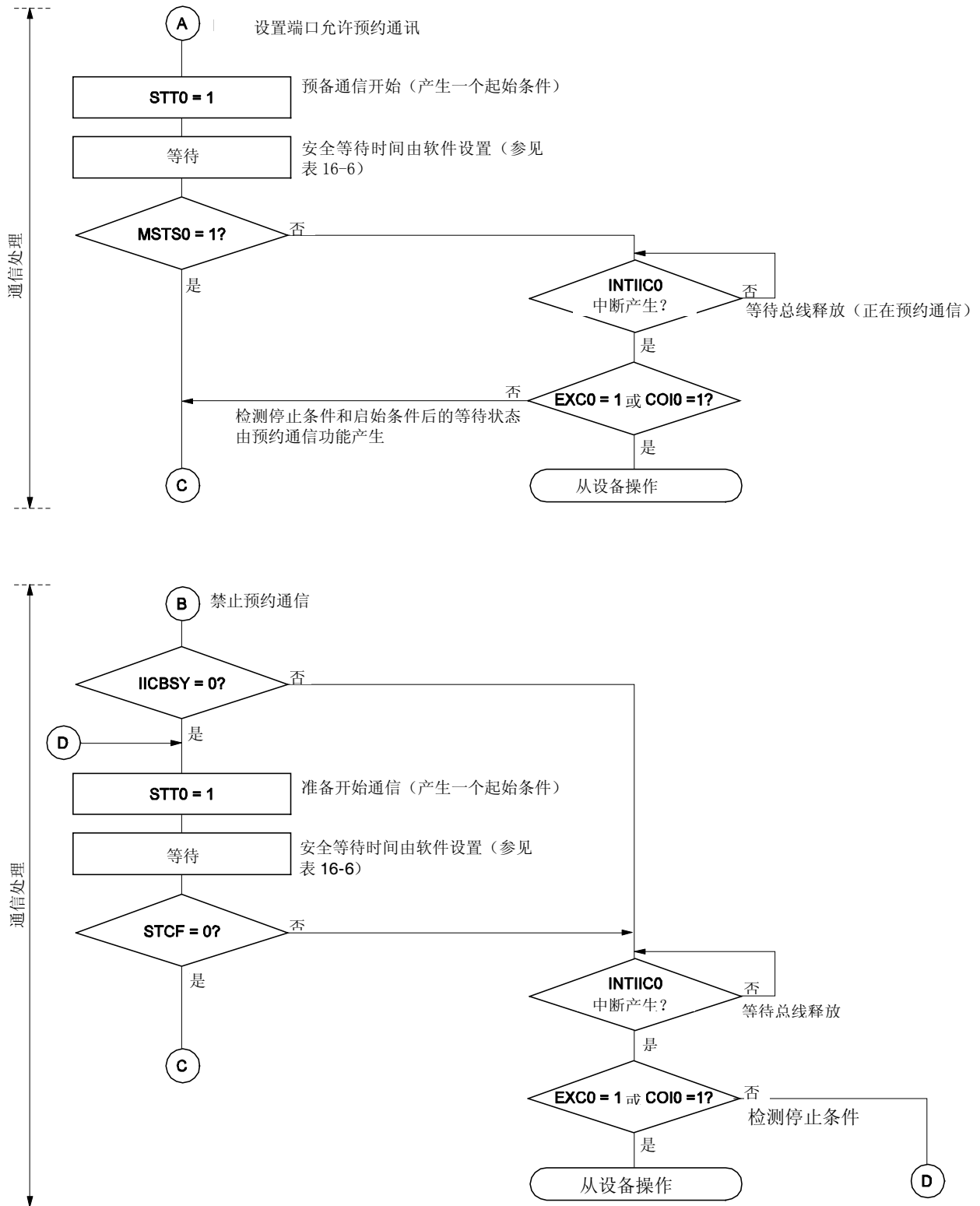
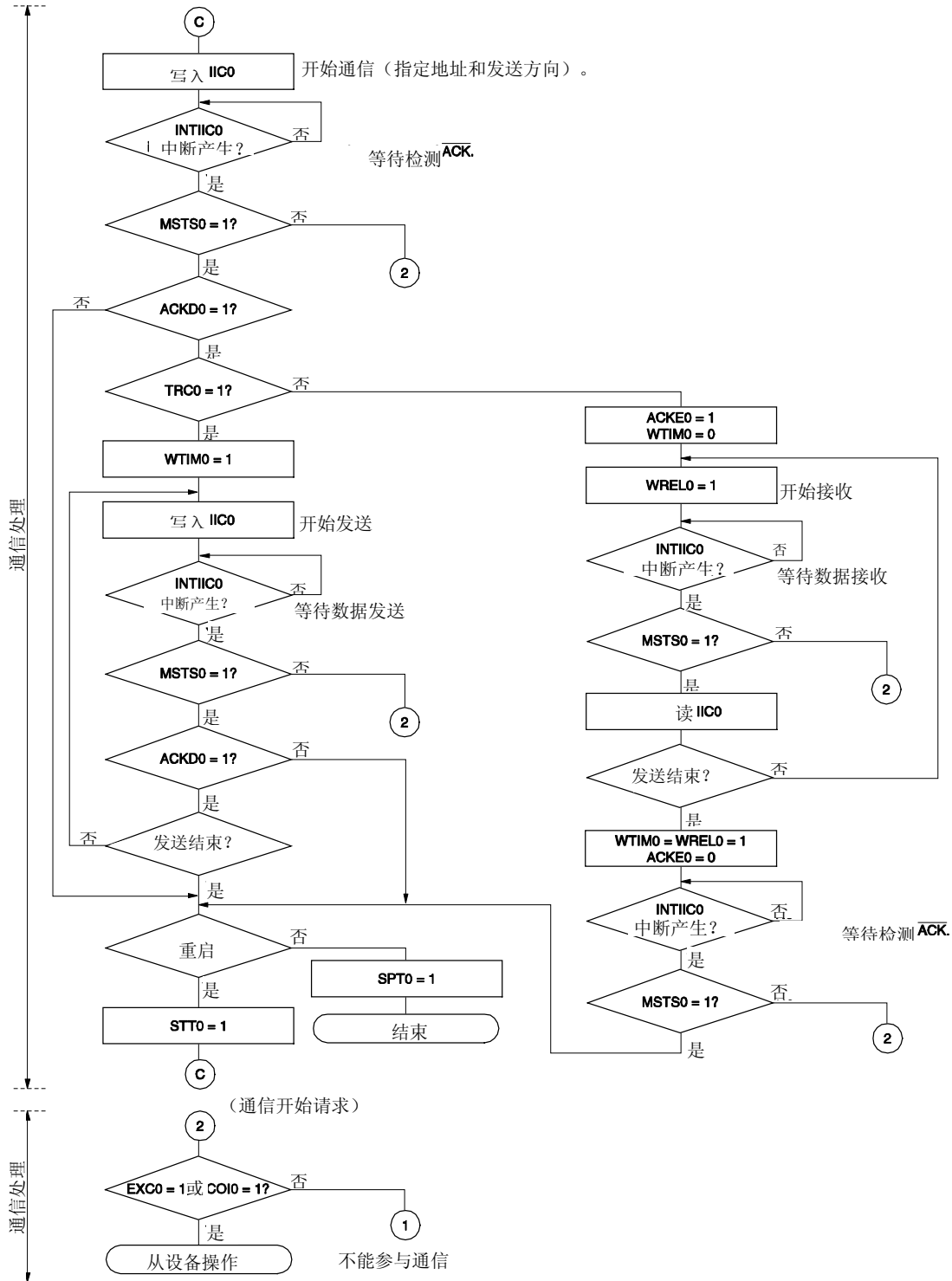


图 16-24. 主设备操作流程 (多主设备系统) (3/3)



备注

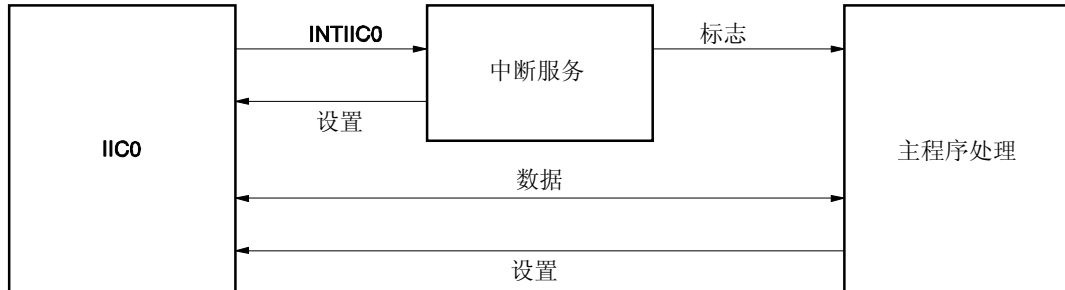
1. 遵照正在通信的产品规范，注意发送和接收的格式。
2. 如果在多主设备系统中作为一个主设备使用时，则应在每次中断 INTIIC0 发生后读取 MSTS0 位来检查仲裁结果。
3. 如果在多主设备系统中作为一个从设备使用时，则通过在每次中断 INTIIC0 发生后使用 IICS0 和 IICF0 寄存器来检查状态，并决定下一步的处理。

(3) 从设备操作

从设备操作过程如下所示。

基本上，从设备操作是由事件驱动的。因此，必须执行 INTIIC0 中断处理过程(该中断处理过程必须完全改变操作状态例如通信时的停止条件检测)。

在下列说明中，假定数据通信不支持扩展码。同时假定 INTIIC0 中断服务只执行状态转换处理过程，且实际数据通信过程由主程序执行。



因此，设置以下 3 个标志，并传递给主程序而不是 INTIIC0，这样可以执行数据通信处理过程。

<1> 通信模式标志

此标志指示下列两种通信状态。

- 清除模式: 未进行数据通信
- 通信模式: 进行数据通信(从有效地址检测到停止条件检测，不检测来自主设备的 $\overline{\text{ACK}}$ 信号，地址不相等)

<2> 准备标志

此标志表示允许进行数据通信。它的功能和普通数据通信的 INTIIC0 中断相同。通过中断服务可以设置此标志，通过主程序可以将该标志清零。通信启动时，可由中断服务将此标志清零。但是，在发送第一个数据时不能由中断服务设置准备标志。因此，发送第一个数据时该标志未清零 (认为地址匹配是下一个数据发送请求)。

<3> 通信方向标志

此标志指示通信方向。它的值与 TRC0 相同。

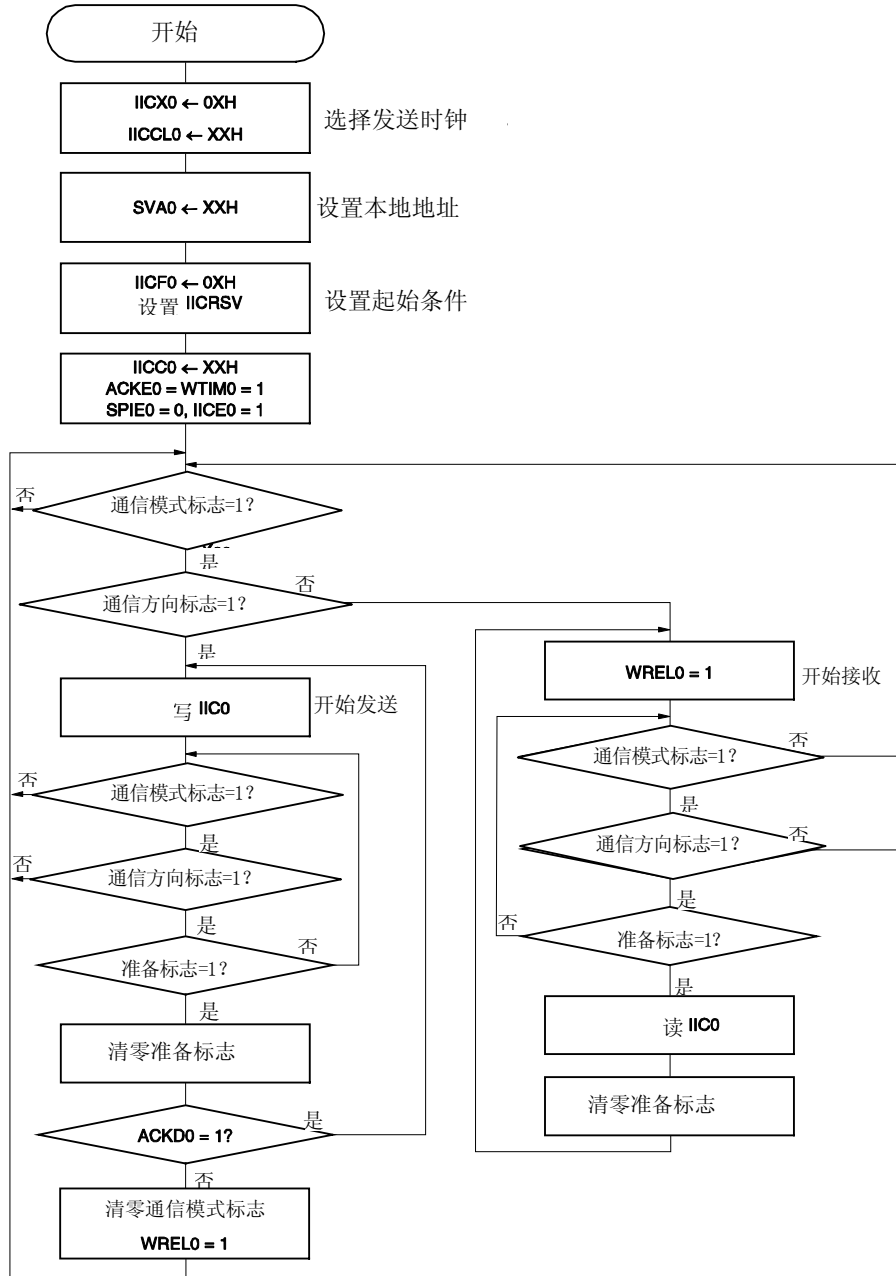
从设备操作的主程序处理过程说明如下。

启动串行接口 IIC0 且等待直到允许通信。当允许通信时，通过使用通信模式标志和准备标志来执行通信 (由中断来执行停止条件和启动条件的处理。这里，通过标志来检查状态)。

重复发送操作直到主设备不再返回 ACK 信号。如果没有从主设备返回 ACK 信号，则通信结束。

对于接收操作，则要接收一定量的数据。通信完成时，ACK 信号不作为下一个数据返回。此后，主设备产生一个停止条件或重启条件。使用这种方法可从通信状态退出。

图 16-25. 从设备操作流程图 (1)



备注 遵照正在通信的产品规范，注意发送和接收的格式。

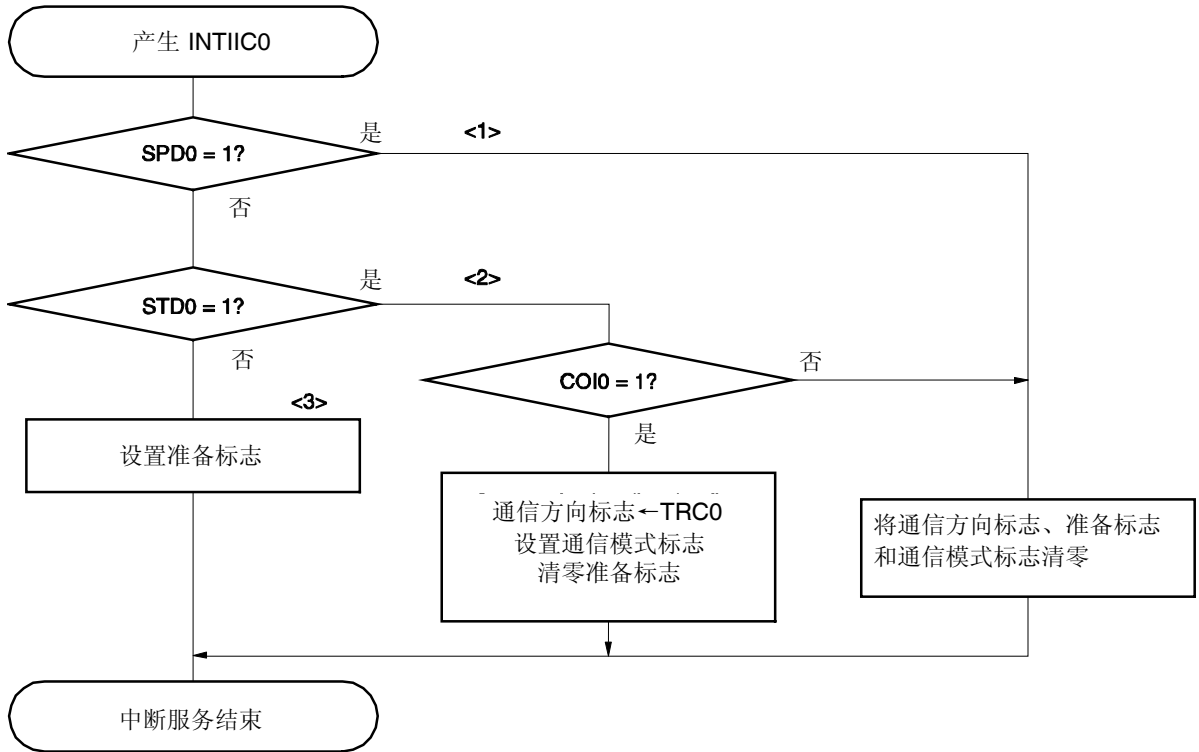
以下示例说明了从设备 INTIIC0 中断处理过程 (处理中假设不使用扩展码)。INTIIC0 中断检查状态, 同时执行下列操作。

- <1> 若发出停止条件则通信停止。
- <2> 若发出起始条件, 检查地址并在地址不相等时结束通信。如果地址相等, 则设置通信模式, 取消等待, 且处理过程从中断返回(准备标志清零)。
- <3> 对于数据发送/接收, 仅需设置准备标志。处理过程从中断返回, 同时 I²C 总线保持等待状态。

备注 以上<1> ~ <3> 与图 16-26 从设备操作流程(2)中<1> ~ <3> 相对应。

★

图 16-26. 从设备操作流程(2)



16.5.17 I²C 中断请求 (INTIIC0) 产生时序

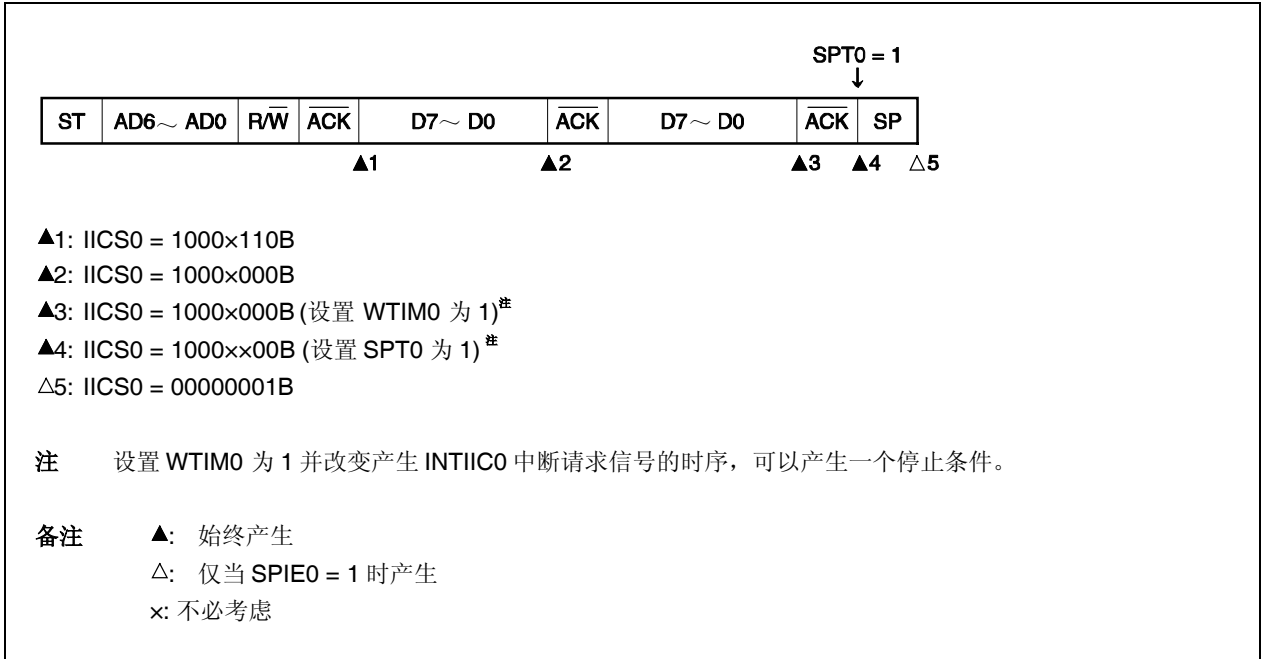
以下显示了发送或接收数据的时序和中断请求信号 INTIIC0 产生时序，以及 INTIIC0 信号产生时 IICS0 寄存器的值。

备注	ST:	起始条件
	AD6 ~ AD0:	地址
	$\overline{R/W}$:	传送方向指示
	\overline{ACK} :	应答
	D7 ~ D0:	数据
	SP:	停止条件

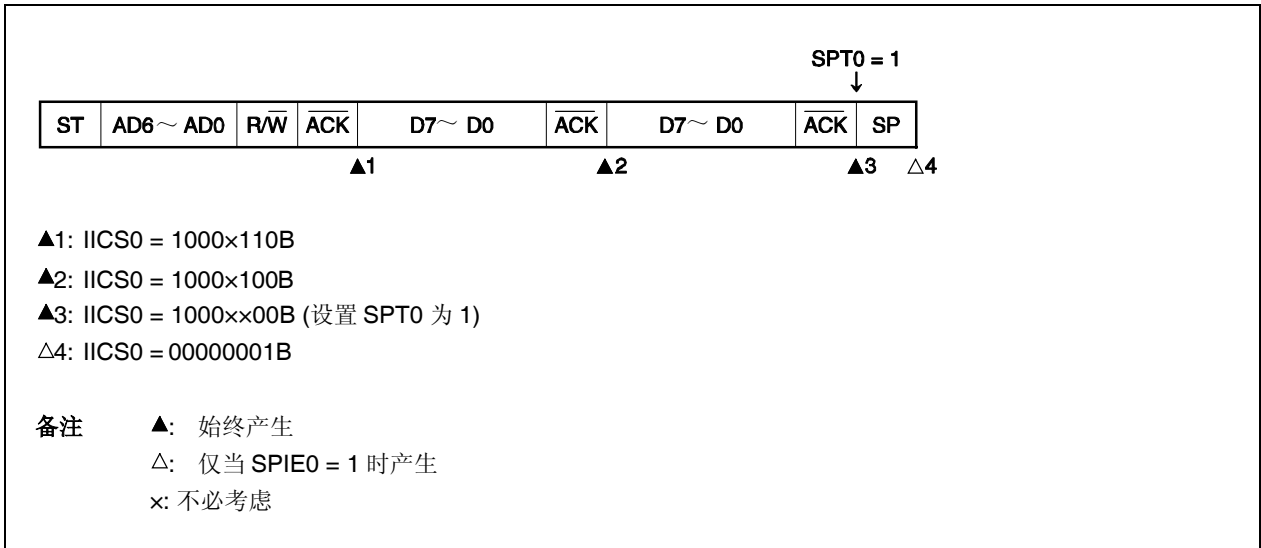
(1) 主设备操作

(a) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止(发送/接收)

(i) 当 $WTIM0 = 0$

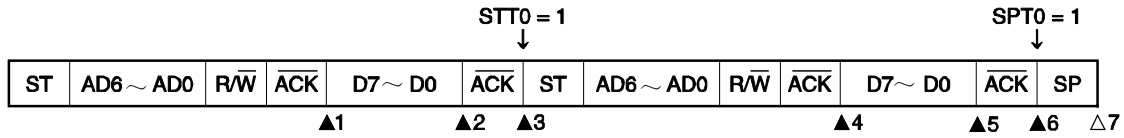


(ii) 当 $WTIM0 = 1$



(b) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止 (重启)

(i) 当 $WTIM0 = 0$

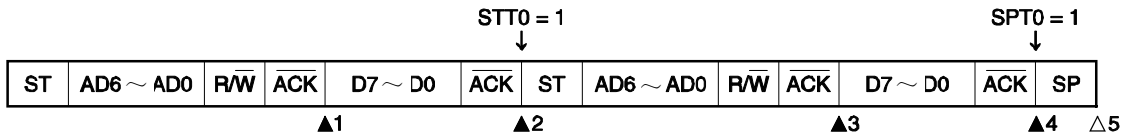


- ▲1: IIC0 = 1000x110B
- ▲2: IIC0 = 1000x000B (设置 $WTIM0$ 为 1)^{※1}
- ▲3: IIC0 = 1000xx00B (将 $WTIM0$ 清零^{※2}, 设置 $STT0$ 为 1)
- ▲4: IIC0 = 1000x110B
- ▲5: IIC0 = 1000x000B (设置 $WTIM0$ 为 1)^{※3}
- ▲6: IIC0 = 1000xx00B (设置 $SPT0$ 为 1)
- △7: IIC0 = 00000001B

- 注
1. 设置 $WTIM0$ 为 1 并改变产生 $INTIIC0$ 中断请求信号的时序, 可以产生一个起始条件。
 2. 设置 $WTIM0=0$ 保存原始设置。
 3. 设置 $WTIM0$ 为 1 并改变产生 $INTIIC0$ 中断请求信号的时序, 可以产生一个停止条件。

- 备注
- ▲: 始终产生
 - △: 仅当 $SPIE0 = 1$ 时产生
 - x: 不必考虑

(ii) 当 $WTIM0 = 1$

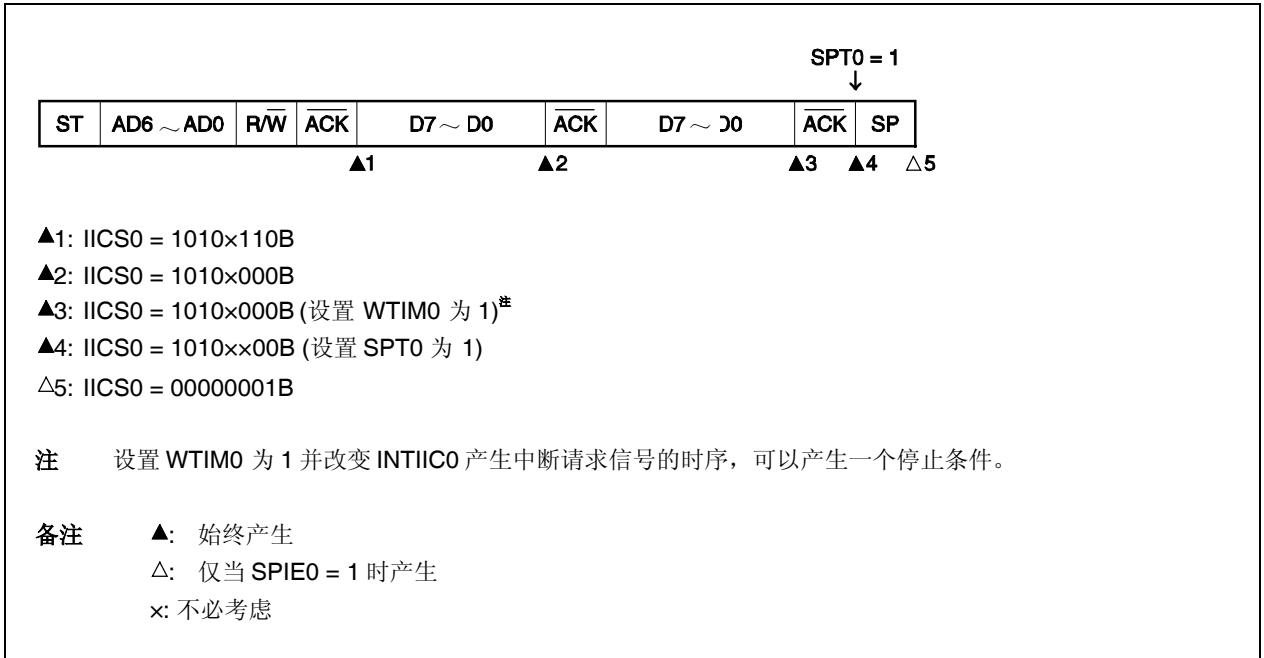


- ▲1: IIC0 = 1000x110B
- ▲2: IIC0 = 1000xx00B (设置 $STT0$ 为 1)
- ▲3: IIC0 = 1000x110B
- ▲4: IIC0 = 1000xx00B (设置 $SPT0$ 为 1)
- △5: IIC0 = 00000001B

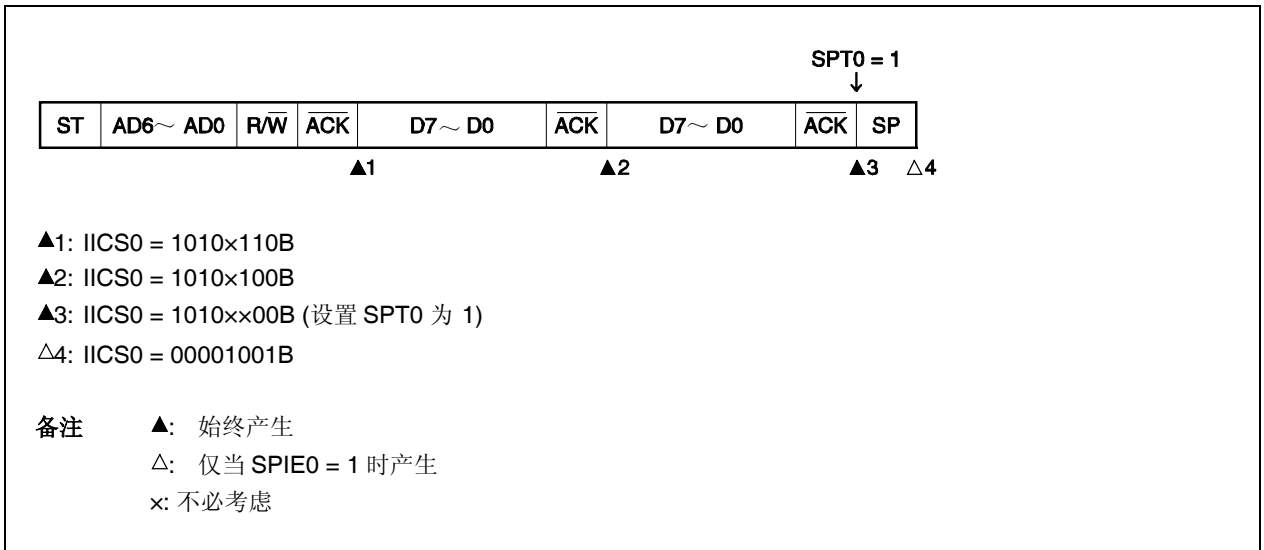
- 备注
- ▲: 始终产生
 - △: 仅当 $SPIE0 = 1$ 时产生
 - x: 不必考虑

(c) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止 (发送扩展码)

(i) 当 WTIMO = 0



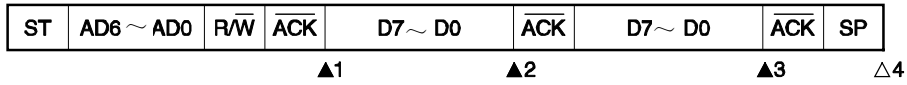
(ii) 当 WTIMO = 1



(2) 从设备操作 (接收从设备地址数据)

(a) 开始 ~ 地址 ~ 数据 ~ 数据 ~ 停止

(i) 当 $WTIM0 = 0$



▲1: IICS0 = 0001x110B

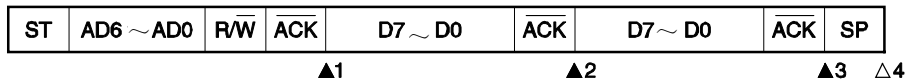
▲2: IICS0 = 0001x000B

▲3: IICS0 = 0001x000B

△4: IICS0 = 00000001B

备注 ▲: 始终产生
 △: 仅当 $SPIE0 = 1$ 时产生
 x: 不必考虑

(ii) 当 $WTIM0 = 1$



▲1: IICS0 = 0001x110B

▲2: IICS0 = 0001x100B

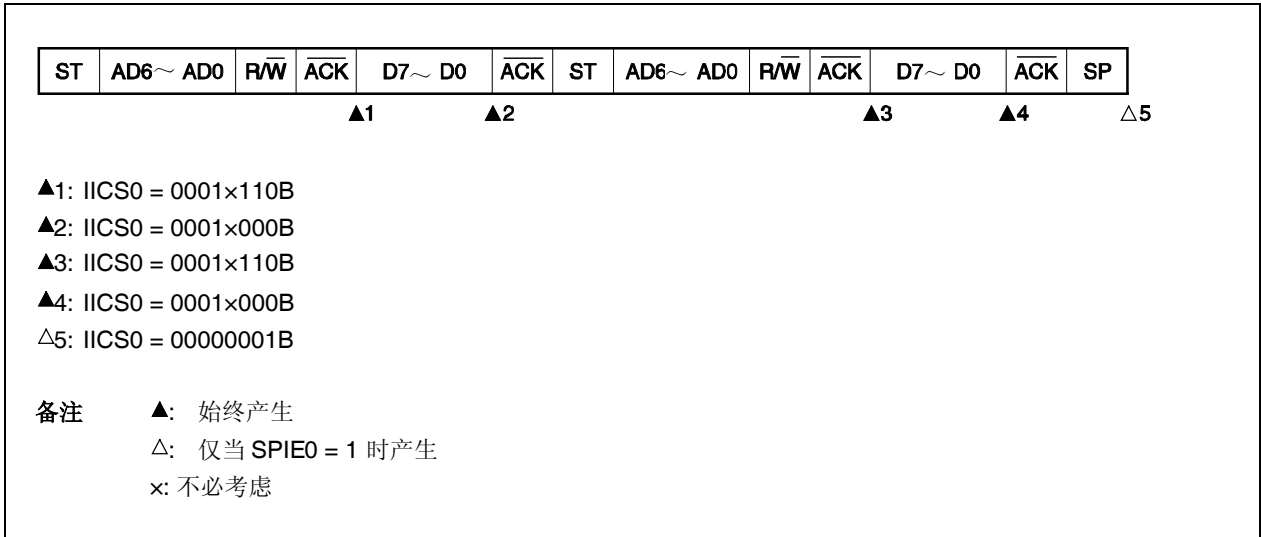
▲3: IICS0 = 0001xx00B

△4: IICS0 = 00000001B

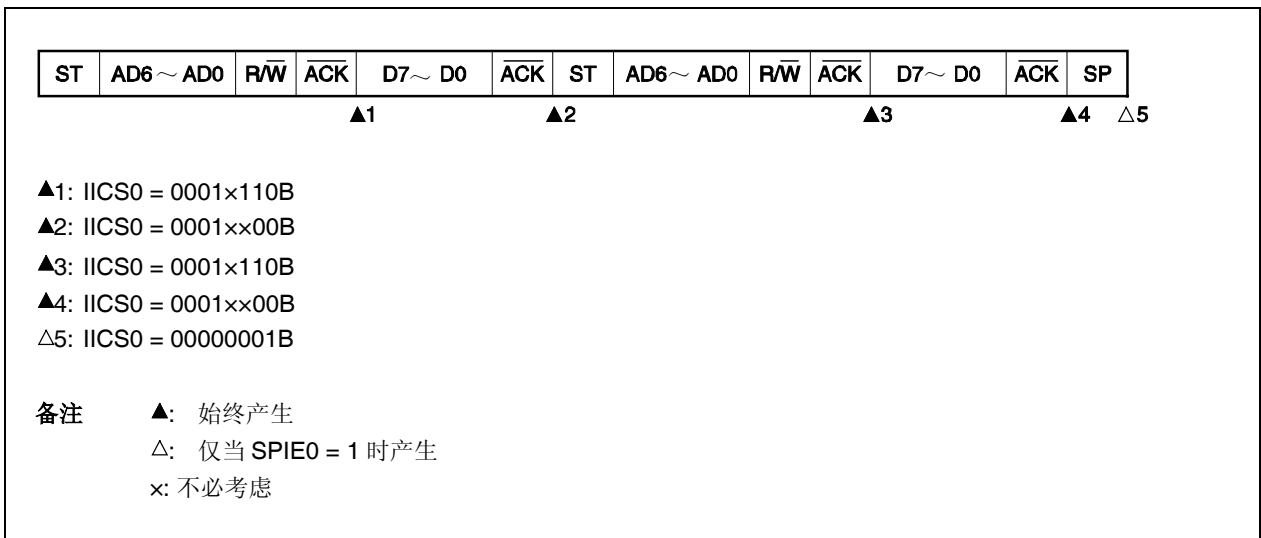
备注 ▲: 始终产生
 △: 仅当 $SPIE0 = 1$ 时产生
 x: 不必考虑

(b) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

(i) 当 $WTIMO = 0$ (重启后, 与 $SVA0$ 相等)

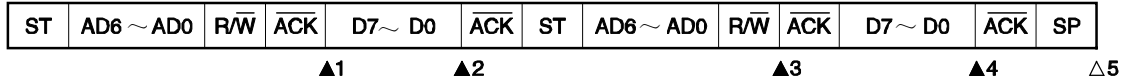


(ii) 当 $WTIMO = 1$ (重启后, 与 $SVA0$ 相等)



(c) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 代码 ~ 数据 ~ 停止

(i) 当 $WTIMO = 0$ (重启后, 地址不相等 (= 扩展码))



▲1: IICSO = 0001x110B

▲2: IICSO = 0001x000B

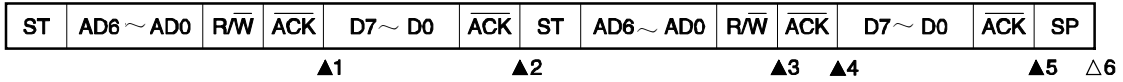
▲3: IICSO = 0010x010B

▲4: IICSO = 0010x000B

△5: IICSO = 00000001B

备注 ▲: 始终产生
 △: 仅当 $SPIE0 = 1$ 时产生
 x: 不必考虑

(ii) 当 $WTIMO = 1$ (重启后, 地址不相等 (= 扩展码))



▲1: IICSO = 0001x110B

▲2: IICSO = 0001xx00B

▲3: IICSO = 0010x010B

▲4: IICSO = 0010x110B

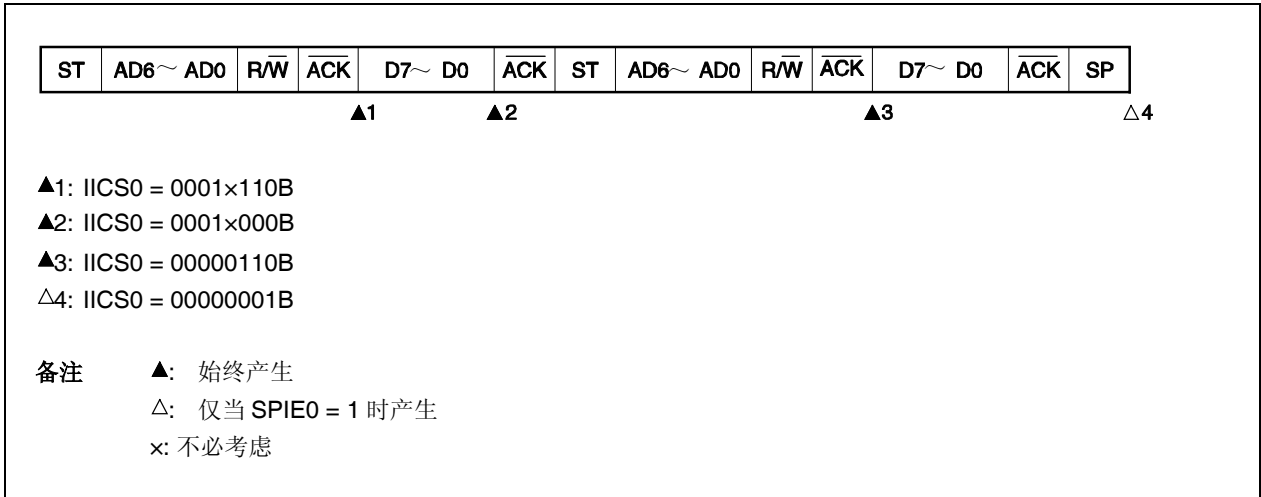
▲5: IICSO = 0010xx00B

△6: IICSO = 00000001B

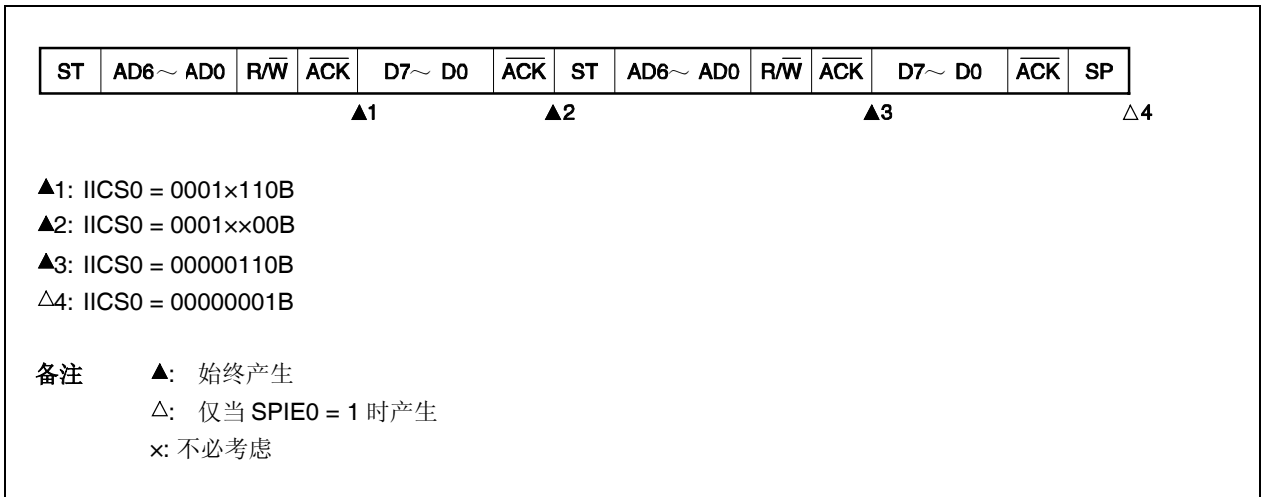
备注 ▲: 始终产生
 △: 仅当 $SPIE0 = 1$ 时产生
 x: 不必考虑

(d) 开始 ~ 地址 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

(i) 当 $WTIMO = 0$ (重启后, 地址不相等 (= 非扩展码))



(ii) 当 $WTIMO = 1$ (重启后, 地址不相等 (= 非扩展码))

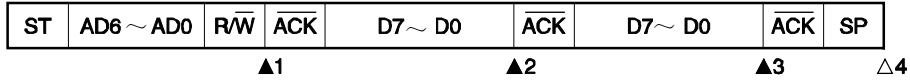


(3) 从设备操作 (当接收到扩展码时)

接收到扩展码时，设备总是参与通信。

(a) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止

(i) 当 $WTIM0 = 0$



▲1: IICS0 = 0010x010B

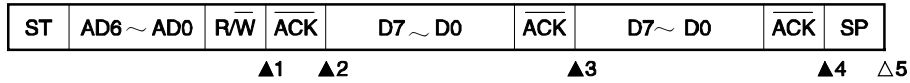
▲2: IICS0 = 0010x000B

▲3: IICS0 = 0010x000B

△4: IICS0 = 00000001B

备注 ▲: 始终产生
 △: 仅当 SPIE0 = 1 时产生
 x: 不必考虑

(ii) 当 $WTIM0 = 1$



▲1: IICS0 = 0010x010B

▲2: IICS0 = 0010x110B

▲3: IICS0 = 0010x100B

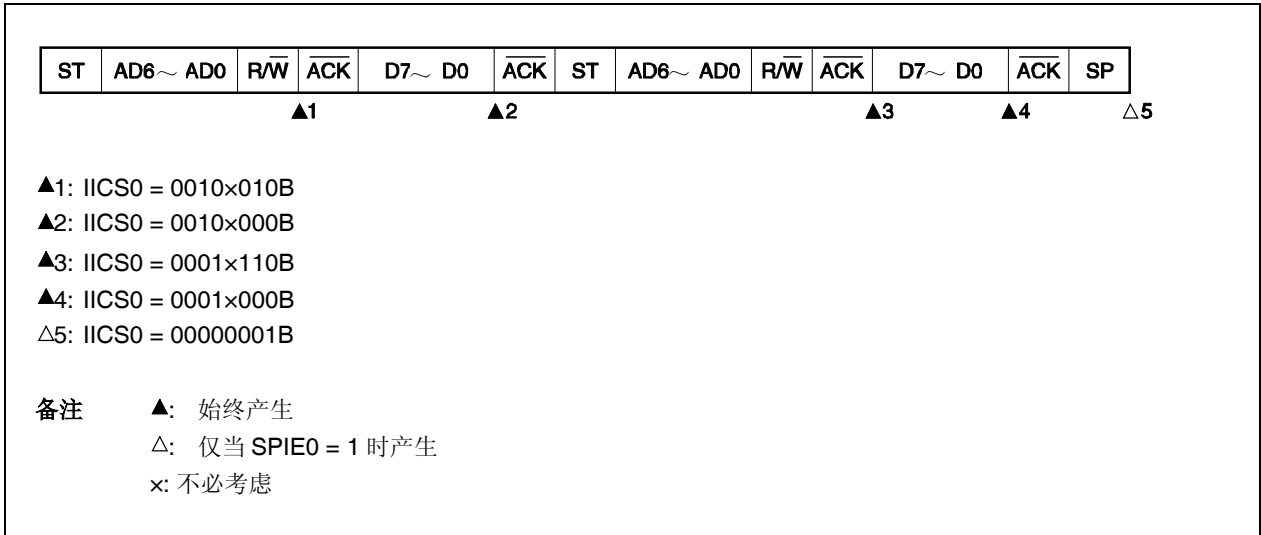
▲4: IICS0 = 0010xx00B

△5: IICS0 = 00000001B

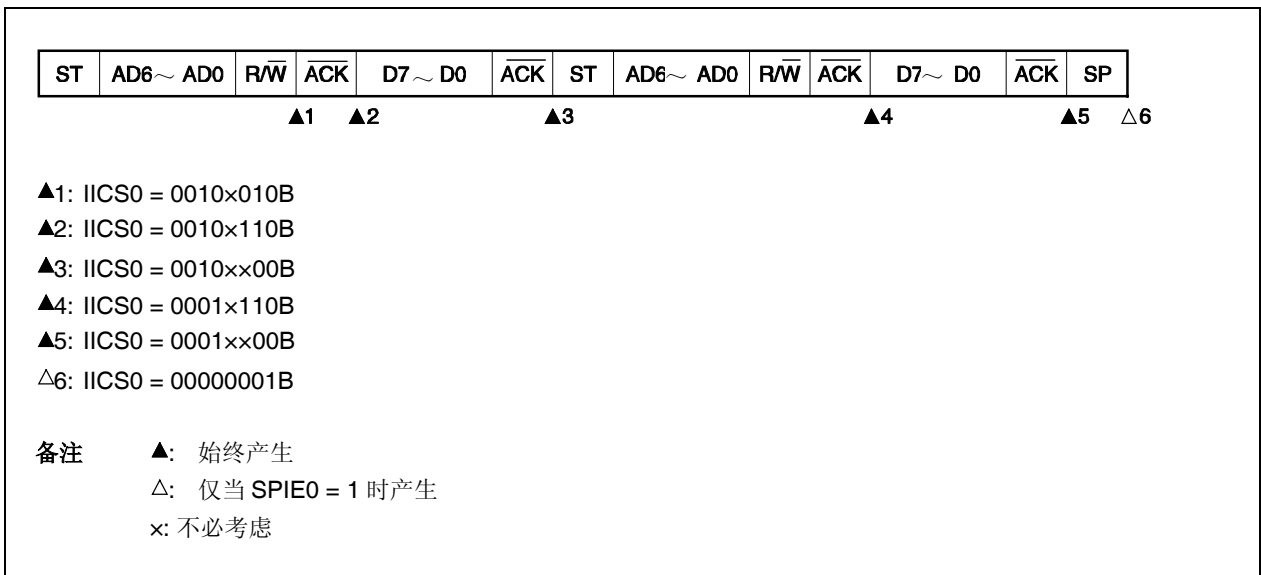
备注 ▲: 始终产生
 △: 仅当 SPIE0 = 1 时产生
 x: 不必考虑

(b) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

(i) 当 $WTIMO = 0$ (重启后, 与 $SVA0$ 相等)



(ii) 当 $WTIMO = 1$ (重启后, 与 $SVA0$ 相等)



(c) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 代码 ~ 数据 ~ 停止

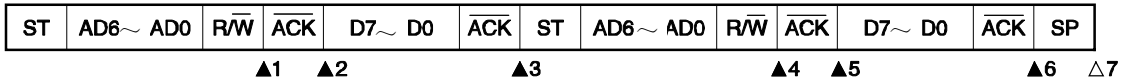
(i) 当 $WTIMO = 0$ (重启后, 接收到扩展码)



- ▲1: IICS0 = 0010x010B
- ▲2: IICS0 = 0010x000B
- ▲3: IICS0 = 0010x010B
- ▲4: IICS0 = 0010x000B
- △5: IICS0 = 00000001B

备注 ▲: 始终产生
 △: 仅当 $SPIE0 = 1$ 时产生
 x: 不必考虑

(ii) 当 $WTIMO = 1$ (重启后, 接收到扩展码)

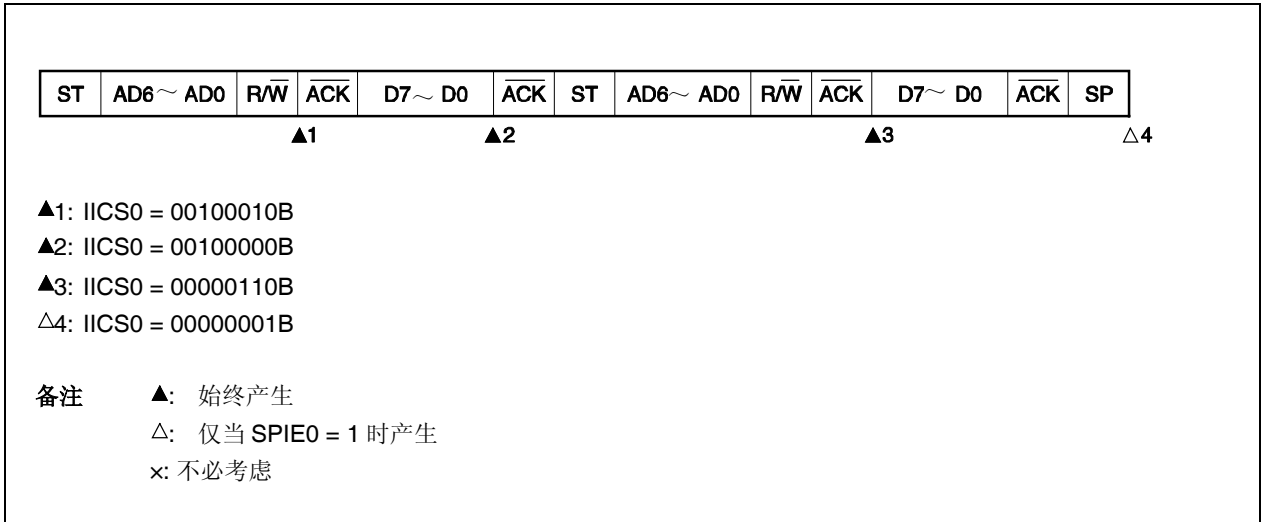


- ▲1: IICS0 = 0010x010B
- ▲2: IICS0 = 0010x110B
- ▲3: IICS0 = 0010xx00B
- ▲4: IICS0 = 0010x010B
- ▲5: IICS0 = 0010x110B
- ▲6: IICS0 = 0010xx00B
- △7: IICS0 = 00000001B

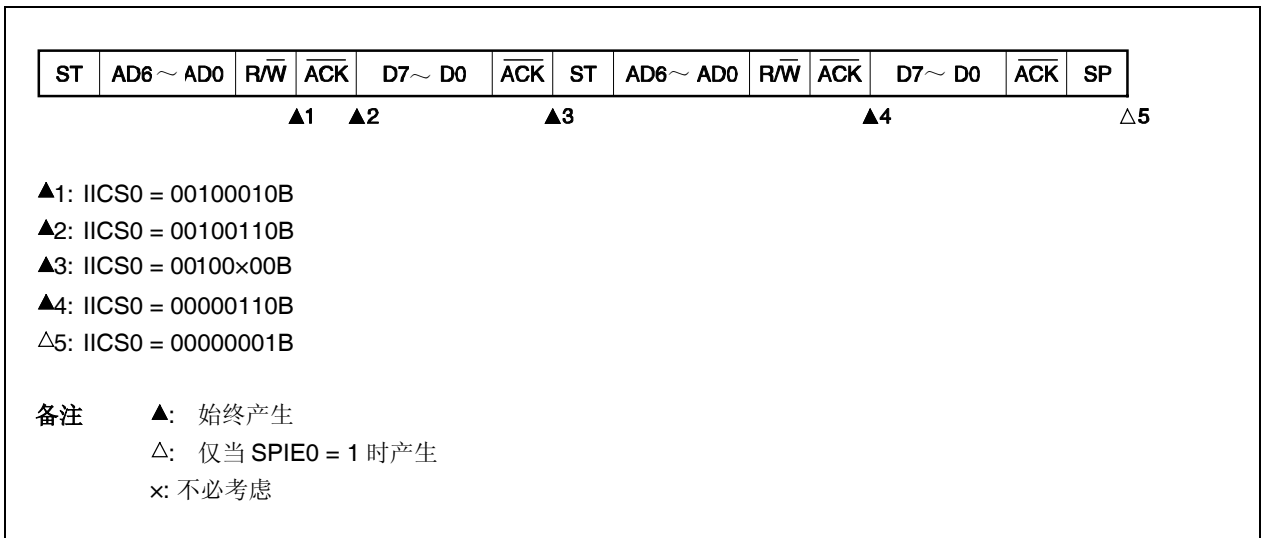
备注 ▲: 始终产生
 △: 仅当 $SPIE0 = 1$ 时产生
 x: 不必考虑

(d) 开始 ~ 代码 ~ 数据 ~ 开始 ~ 地址 ~ 数据 ~ 停止

(i) 当 $WTIMO = 0$ (重启后, 地址不相等 (= 非扩展码))

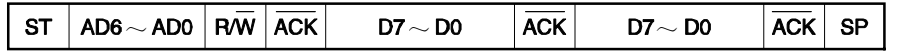


(ii) 当 $WTIMO = 1$ (重启后, 地址不相等 (= 非扩展码))



(4) 未进行通信时的操作

(a) 开始 ~ 代码 ~ 数据 ~ 数据 ~ 停止



△1

△1: IICS0 = 00000001B

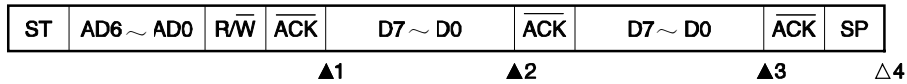
备注 △: 仅当 SPIE0 = 1 时产生

(5) 发生仲裁失败时的操作 (仲裁失败后作为从设备操作)

在多主设备系统中当一个设备作为主设备使用时，可在每次中断请求信号 INTIIC0 产生时读取 MSTS0 位来检查仲裁结果。

(a) 发送从设备地址数据期间发生仲裁失败时

(i) 当 WTIM0 = 0



▲1: IICS0 = 0101x110B

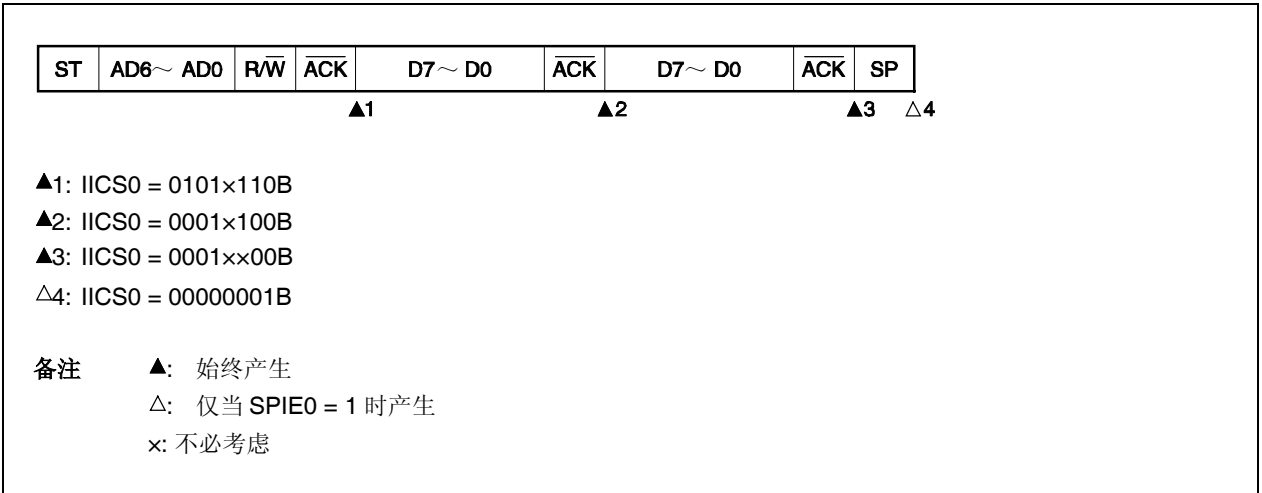
▲2: IICS0 = 0001x000B

▲3: IICS0 = 0001x000B

△4: IICS0 = 00000001B

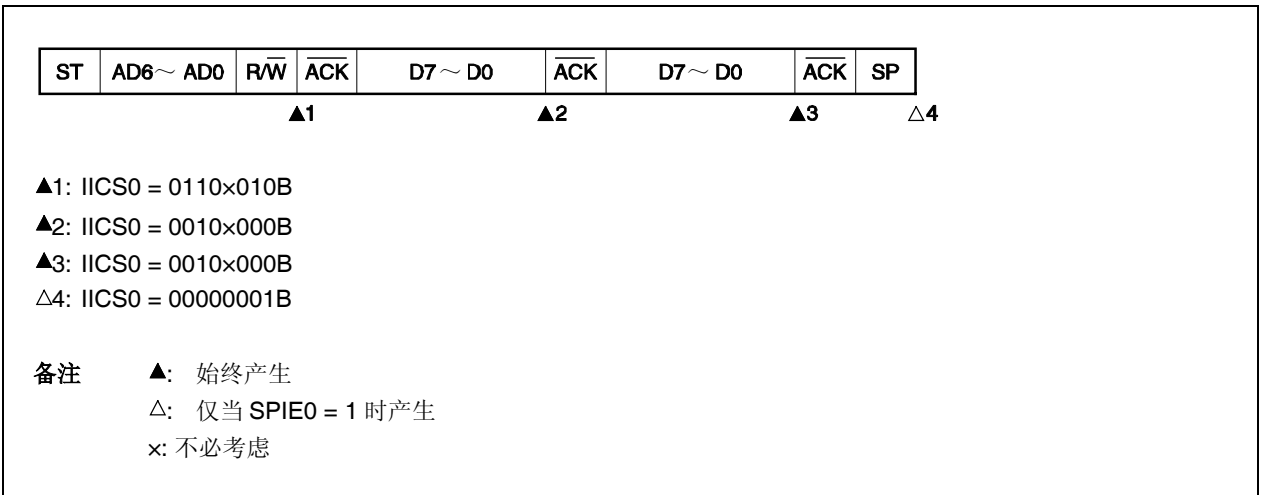
备注 ▲: 始终产生
 △: 仅当 SPIE0 = 1 时产生
 x: 不必考虑

(ii) 当 $WTIM0 = 1$

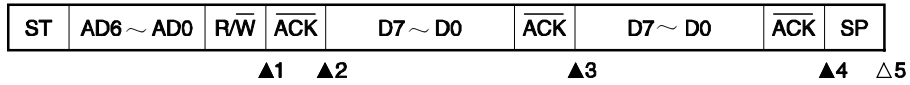


(b) 发送扩展码期间发生仲裁失败时

(i) 当 $WTIM0 = 0$



(ii) 当 $WTIM0 = 1$



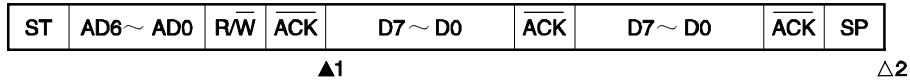
- ▲1: IICSO = 0110x010B
- ▲2: IICSO = 0010x110B
- ▲3: IICSO = 0010x100B
- ▲4: IICSO = 0010xx00B
- △5: IICSO = 00000001B

备注 ▲: 始终产生
 △: 仅当 SPIE0 = 1 时产生
 x: 不必考虑

(6) 发生仲裁失败时的操作 (仲裁失败后不进行通信)

在多主设备系统中当一个设备作为主设备使用时，可在每次中断请求信号 INTIIC0 产生时读取 MSTSO 位来检查仲裁结果。

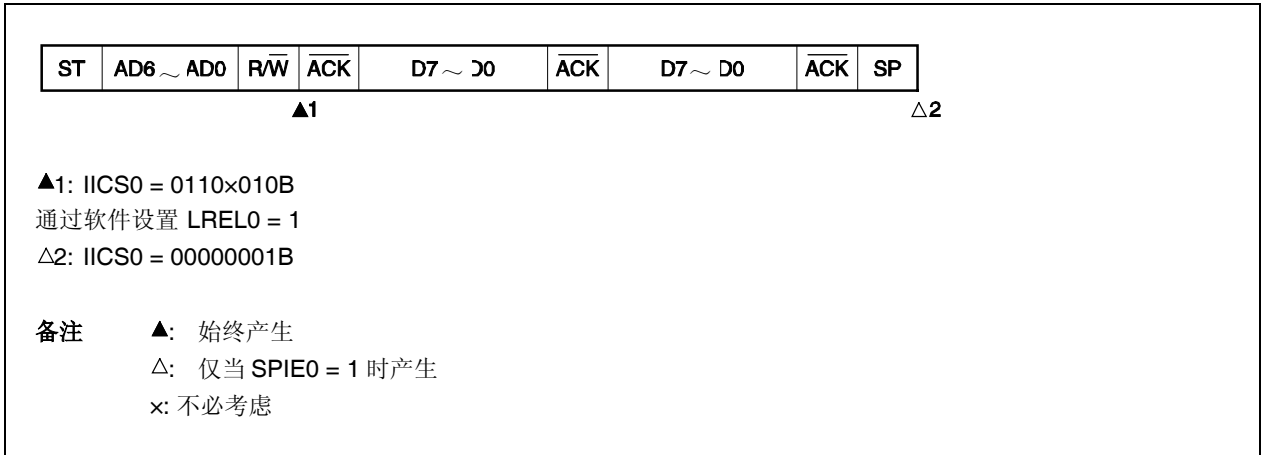
(a) 发送从设备地址数据期间发生仲裁失败时 (当 $WTIM0 = 1$)



- ▲1: IICSO = 01000110B
- △2: IICSO = 00000001B

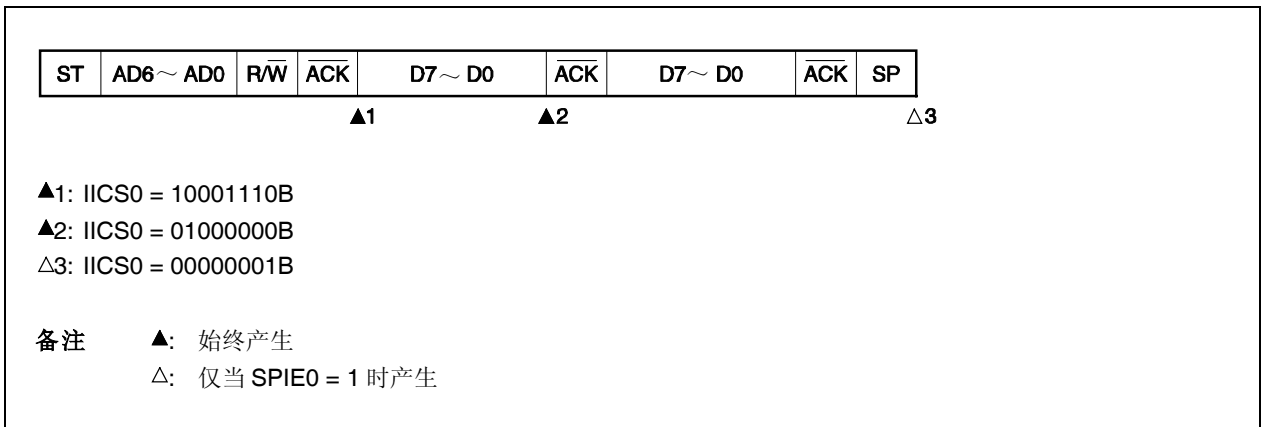
备注 ▲: 始终产生
 △: 仅当 SPIE0 = 1 时产生

(b) 发送扩展码期间发生仲裁失败时

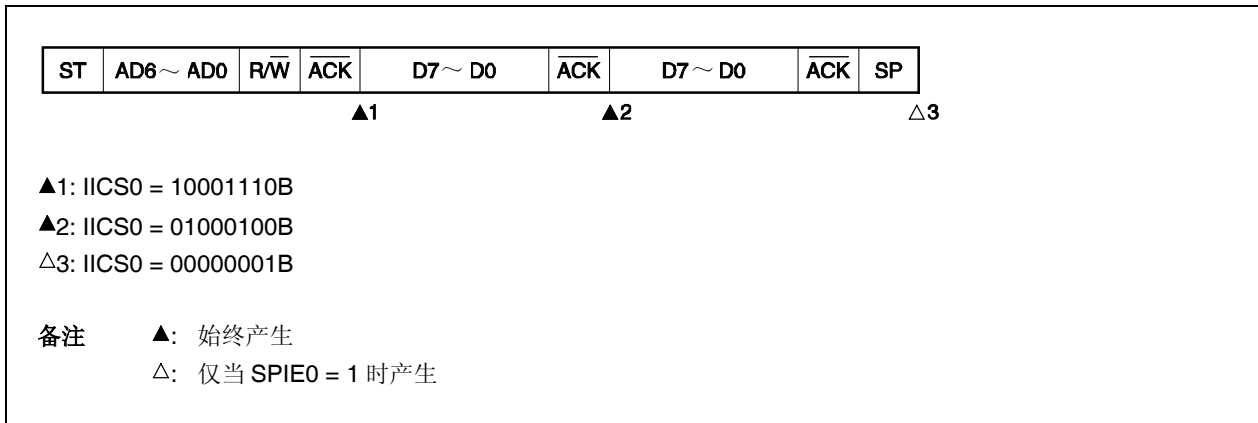


(c) 发送数据期间发生仲裁失败时

(i) 当 WTIMO = 0

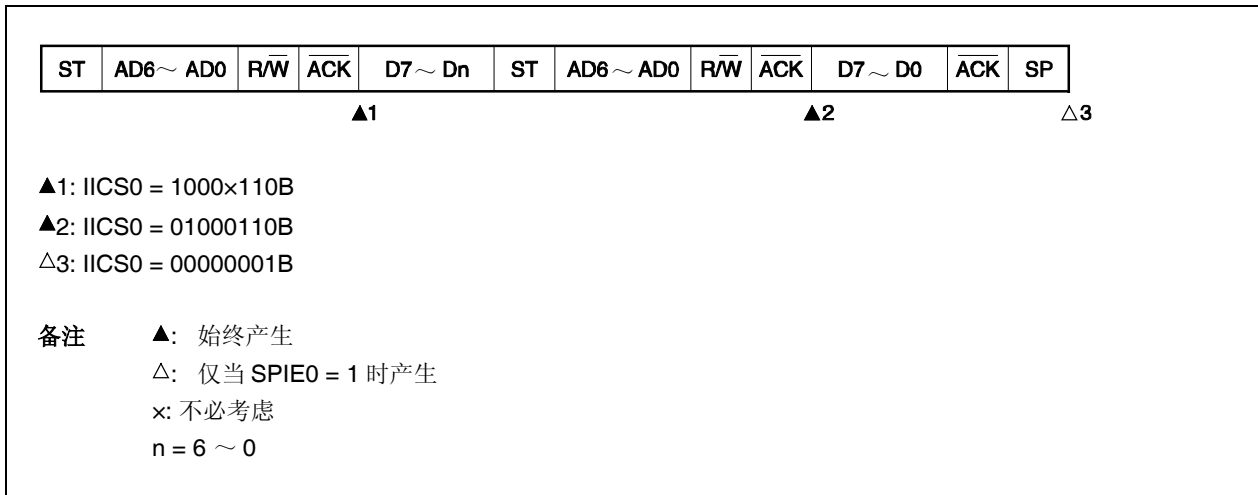


(ii) 当 WTIMO = 1

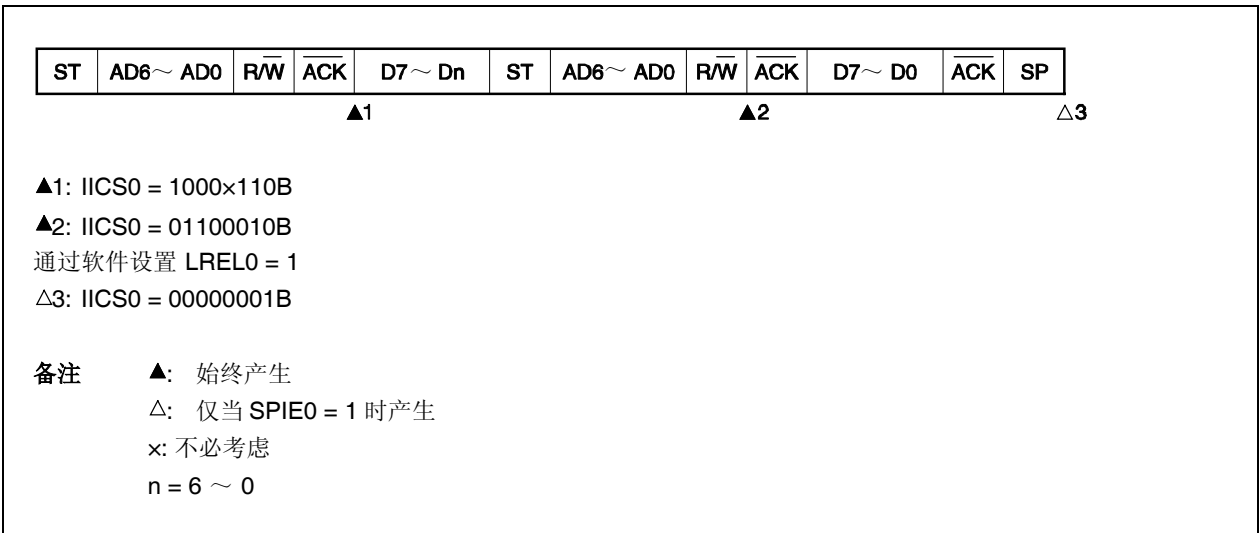


(d) 数据传输期间由于起始条件导致仲裁失败时

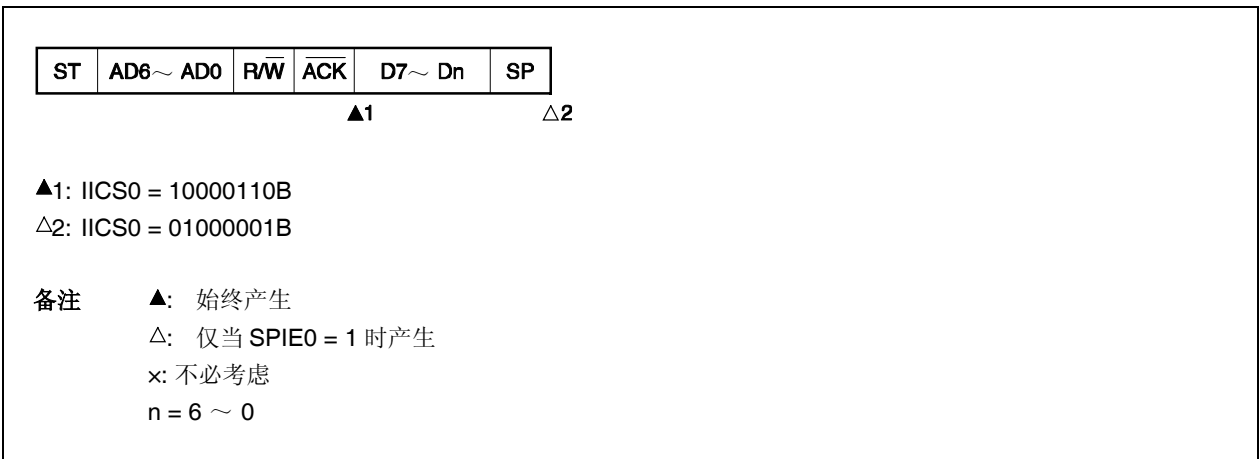
(i) 非扩展码 (例: 与 SVA0 不相等)



(ii) 扩展码

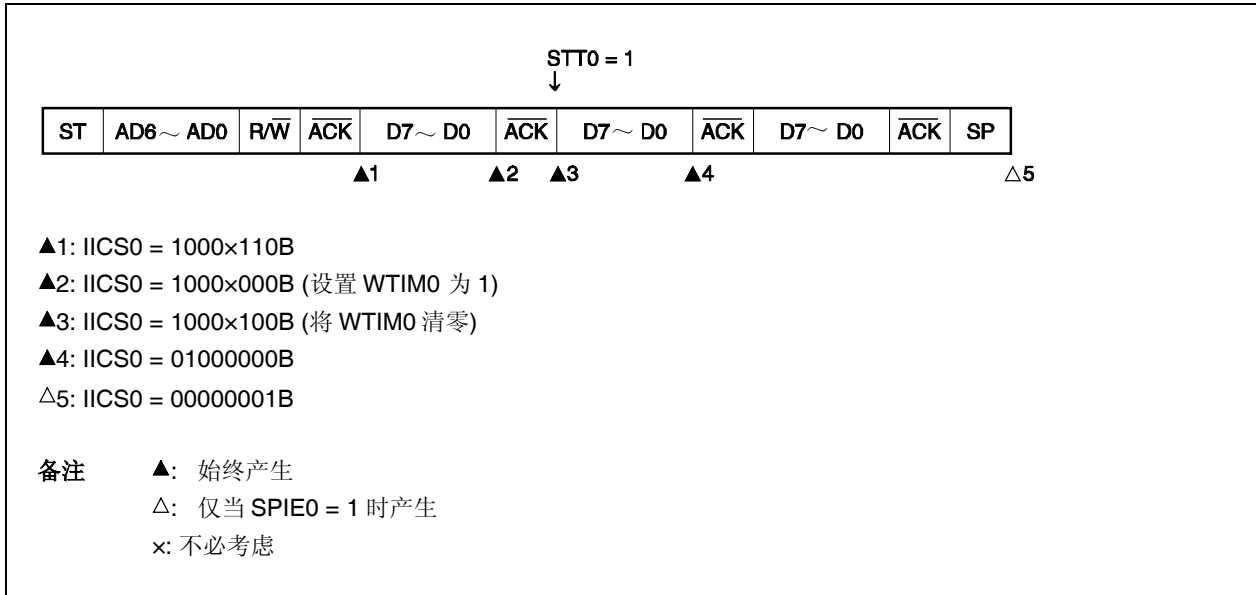


(e) 数据传送期间由于停止条件导致仲裁失败时

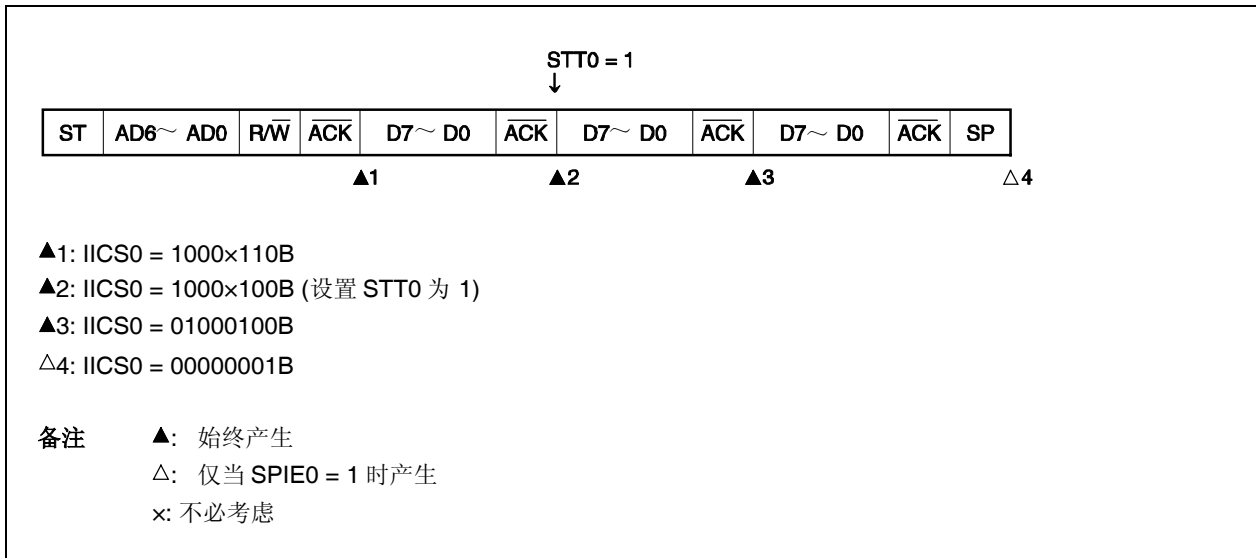


(f) 要产生重启条件时，由于低电平数据导致仲裁失败时

(i) 当 $WTIM0 = 0$

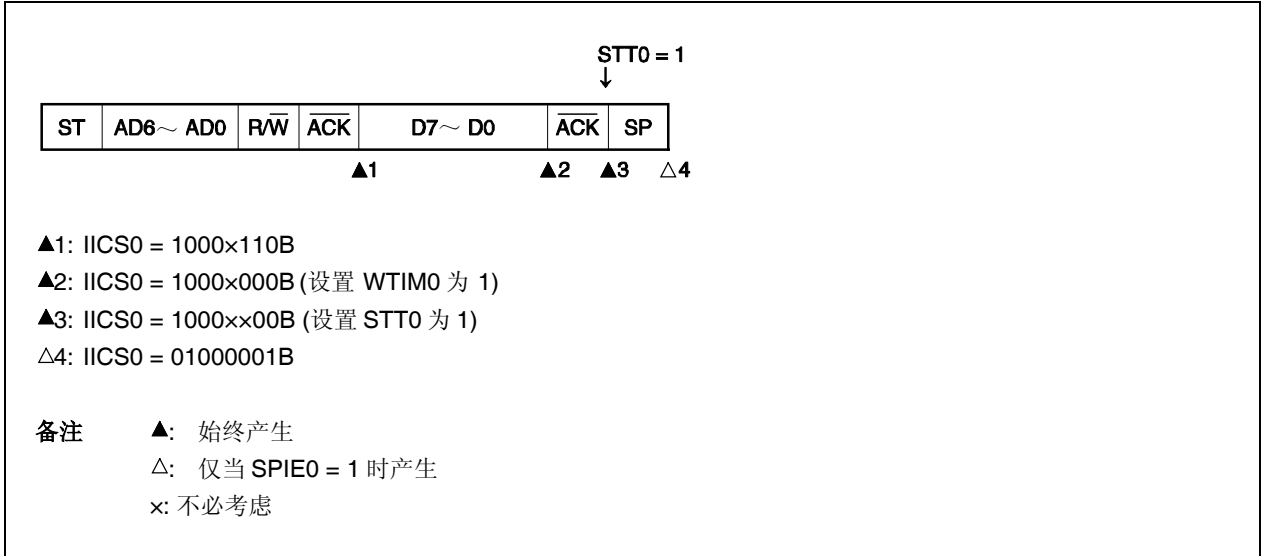


(ii) 当 $WTIM0 = 1$

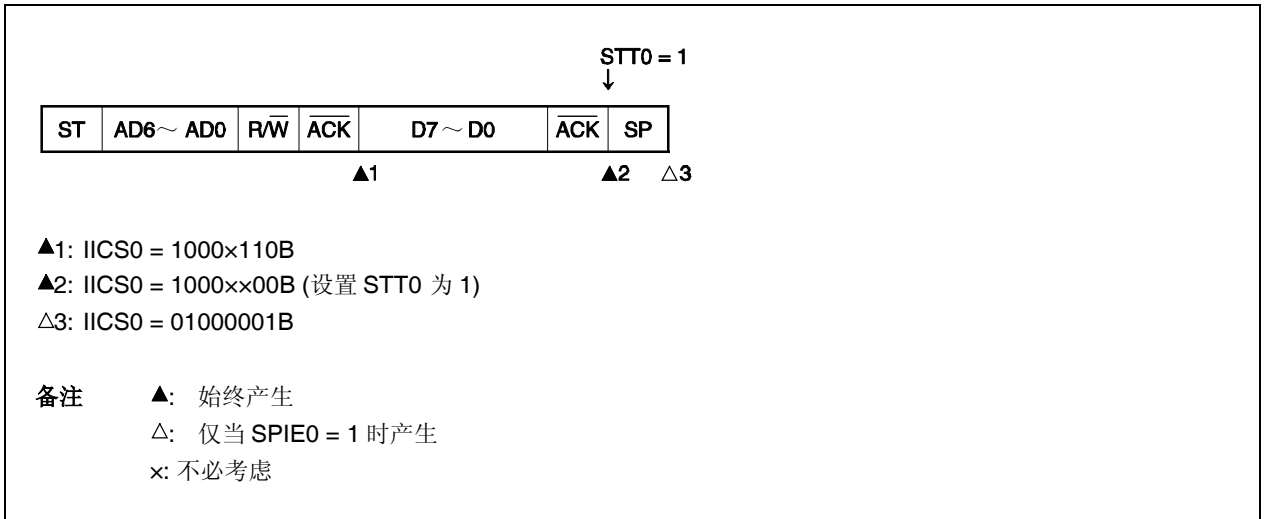


(g) 要产生重启条件时，由于停止条件导致仲裁失败时

(i) 当 $WTIM0 = 0$

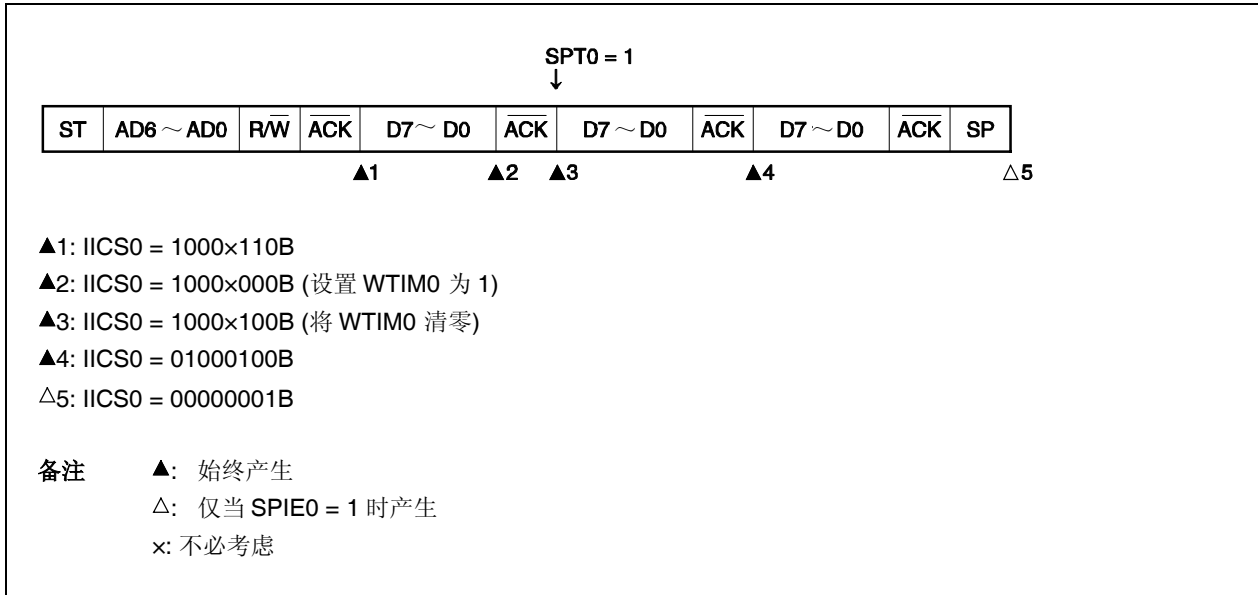


(ii) 当 $WTIM0 = 1$

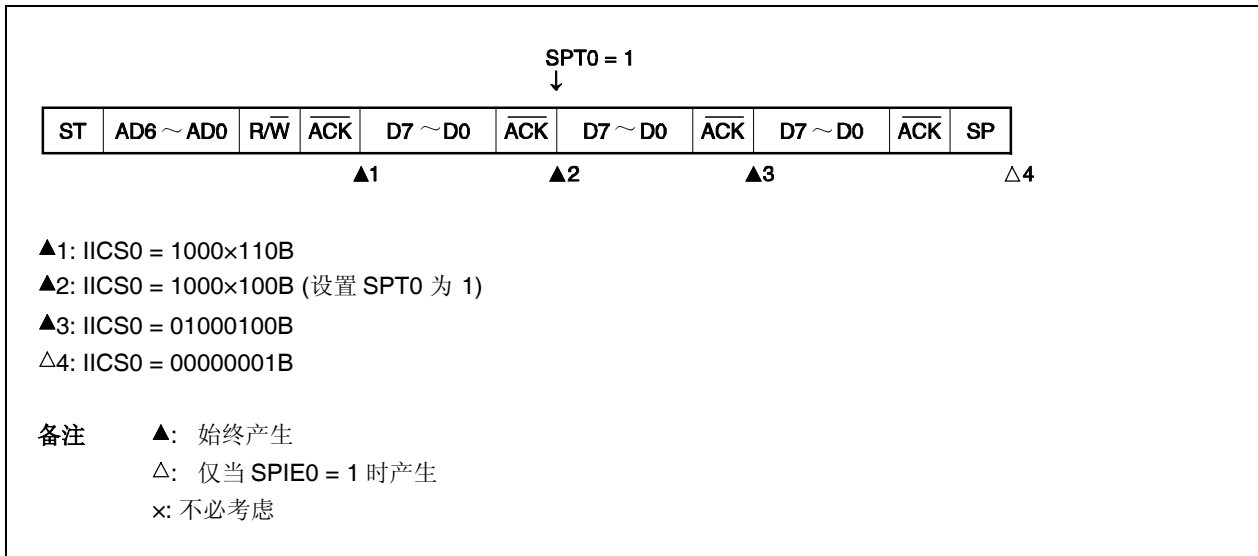


(h) 要产生停止条件时，由于低电平数据导致仲裁失败时

(i) 当 $WTIM0 = 0$



(ii) 当 $WTIM0 = 1$



16.6 时序图

使用 I²C 总线模式时，主设备通过串行总线输出一个地址来选择一个从设备作为它的通信伙伴。

输出从设备地址后，主设备发送 TRC0 位(IIC 状态寄存器 0(IICS0)的第 3 位)，用于指定数据传送方向，然后开始与从设备进行串行通信。

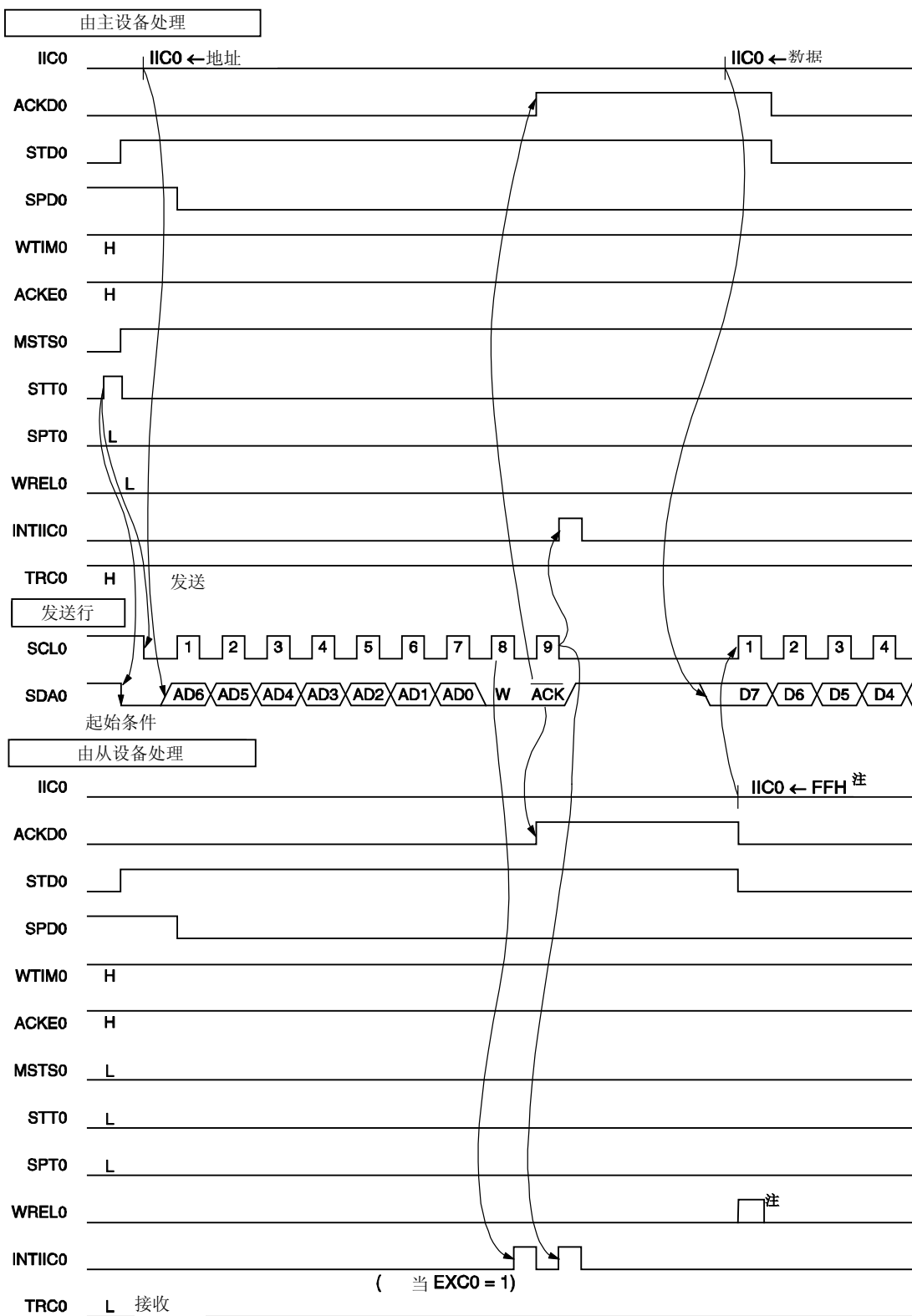
图 16-27 和 16-28 表示了数据通信的时序图。

IIC 移位寄存器 0(IIC0)的移位操作和串行时钟(SCL0)的下降沿同步。将发送数据传入 SO0 锁存器并通过 SDA0 引脚输出(MSB first)。

在 SCL0 的上升沿通过 SDA0 引脚输入的数据被捕捉到 IIC0 寄存器。

图 16-27. 主设备至从设备的通信示例
(当主设备和从设备都选择 9 时钟等待时) (1/3)

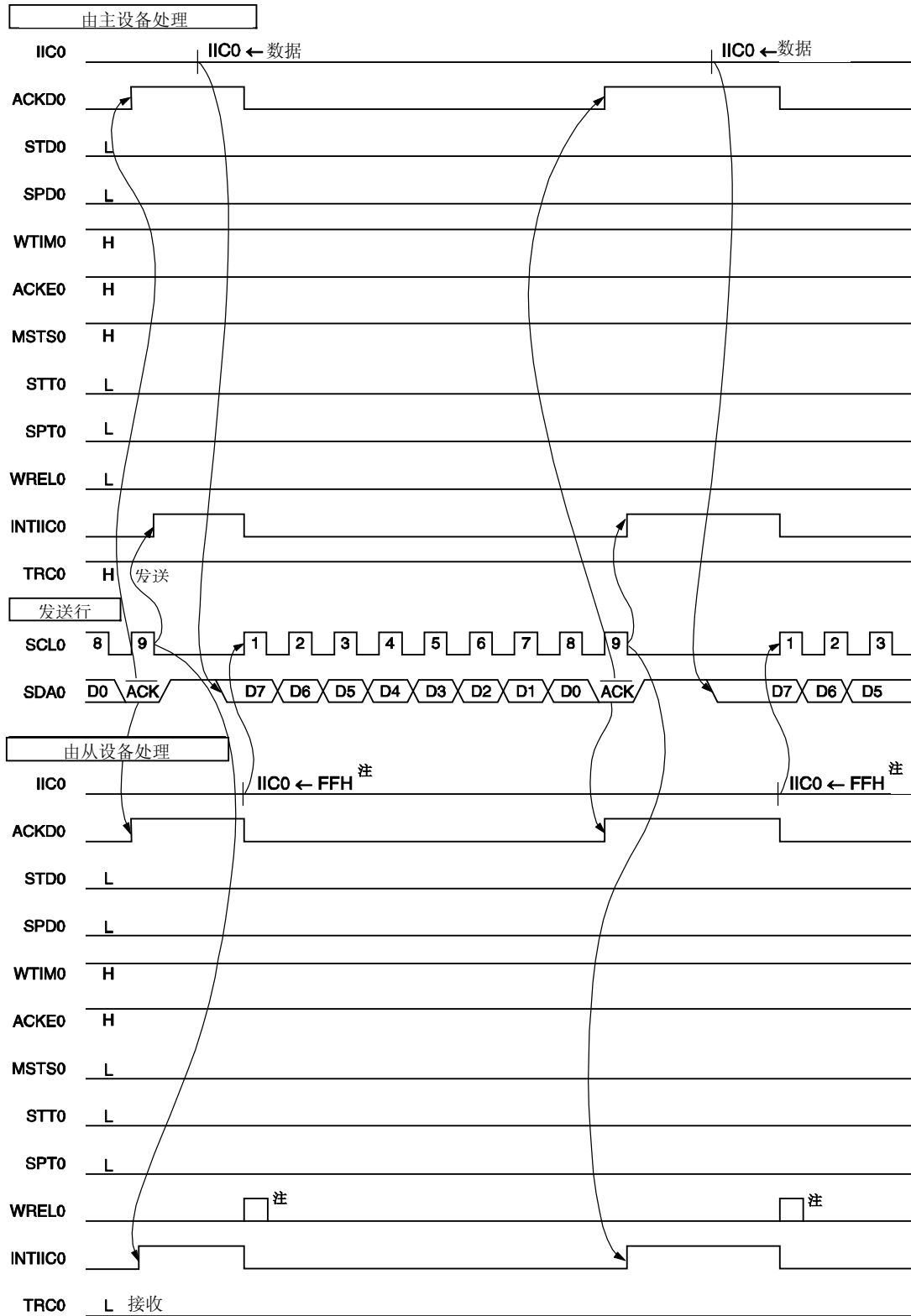
(1) 起始条件 ~ 地址



注 将“FFH”写入 IIC0 或设置 WREL0，可以取消从设备的等待状态。

图 16-27. 主设备至从设备的通信示例
(当主设备和从设备都选择 9 时钟等待时) (2/3)

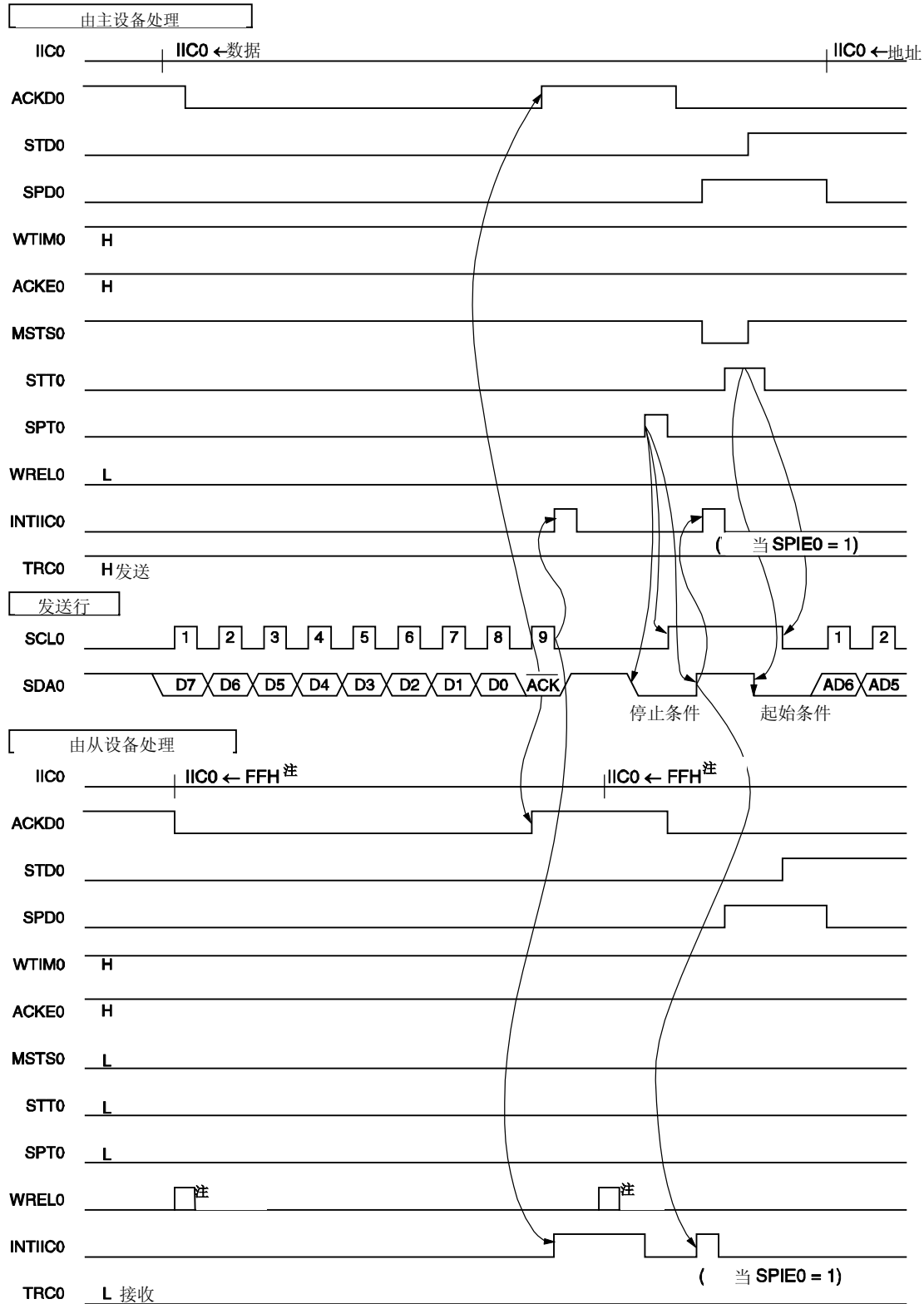
(2) 数据



注 将“FFH”写入 IIC0 或设置 WRELO，可以取消从设备的等待状态。

图 16-27. 主设备至从设备的通信示例
(当主设备和从设备都选择 9 时钟等待时) (3/3)

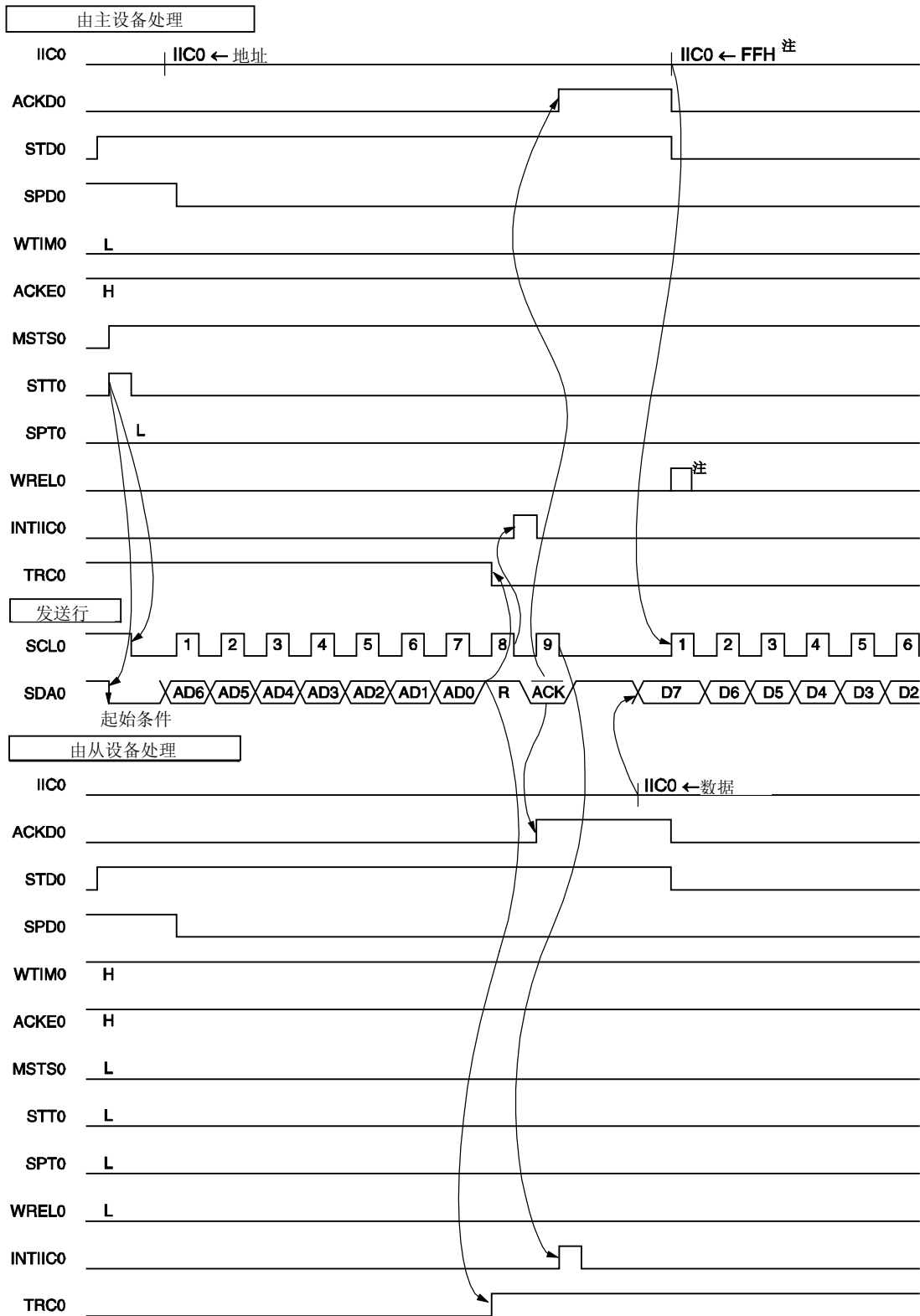
(3) 停止条件



注 将“FFH”写入 IIC0 或设置 WRELO，可以取消从设备的等待状态。

图 16-28. 从设备至主设备通信示例
(当主设备选择 8 时钟等待, 从设备选择 9 时钟等待时) (1/3)

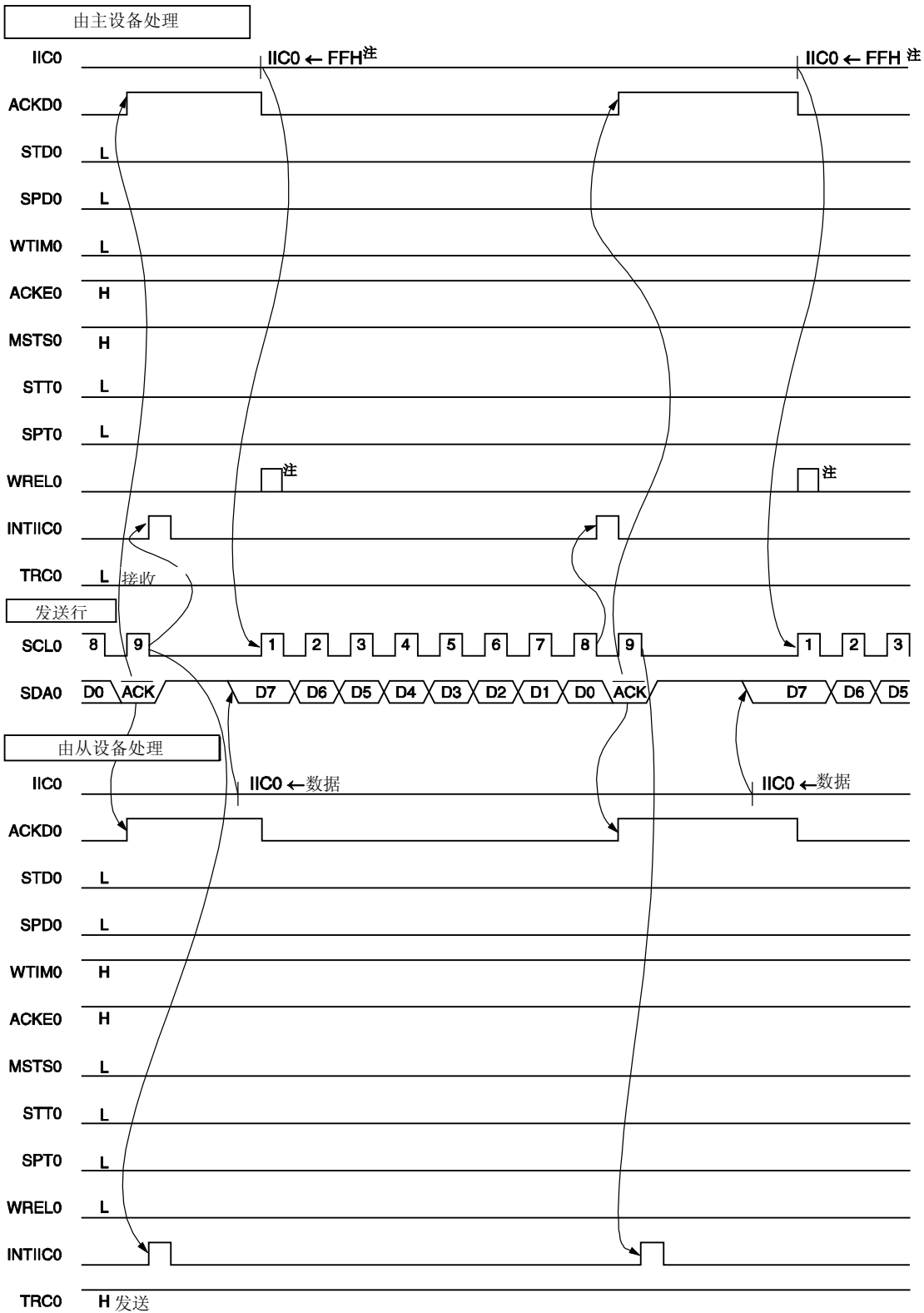
(1) 起始条件 ~ 地址



注 将“FFH”写入 IIC0 或设置 WRELO，可以取消主设备的等待状态。

图 16-28. 从设备至主设备通信示例
(当主设备选择 8 时钟等待, 从设备选择 9 时钟等待时) (2/3)

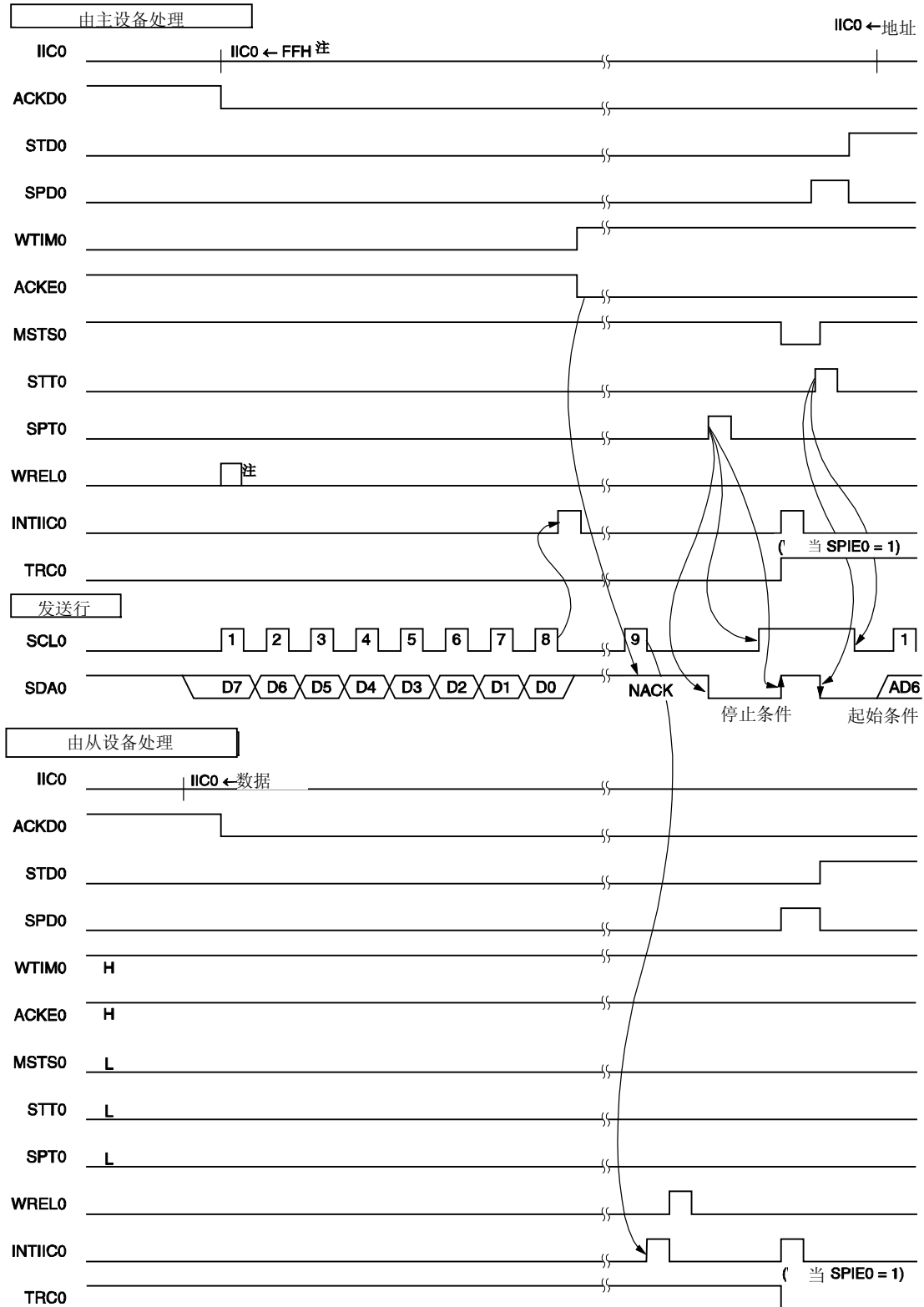
(2) 数据



注 将“FFH”写入 IIC0 或设置 WREL0, 可以取消主设备的等待状态。

图 16-28. 从设备至主设备通信示例
(当主设备选择 8 时钟等待, 从设备选择 9 时钟等待时) (3/3)

(3) 停止条件



注 将“FFH”写入 IIC0 或设置 WRELO, 可以取消主设备的等待状态。

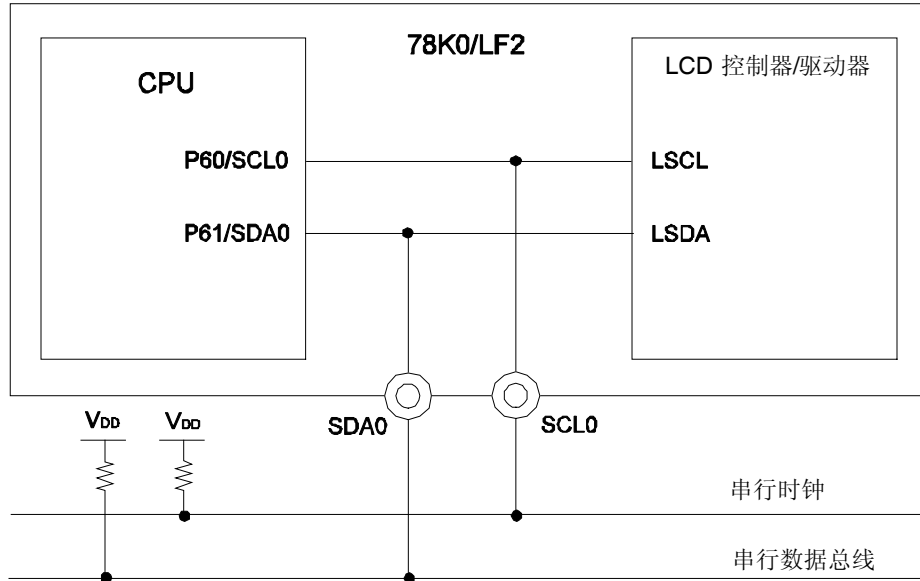
16.7 与 LCD 控制器/驱动器的通信

在 78K0/LF2 中，需经 I²C 总线接口设置 LCD 控制器/驱动器。因此，可以对 LCD 控制器/驱动器的寄存器进行读写。

16.7.1 系统配置

78K0/LF2 的 LCD 控制器/驱动器系统结构如图 16-29 所示。

图 16-29. 系统结构

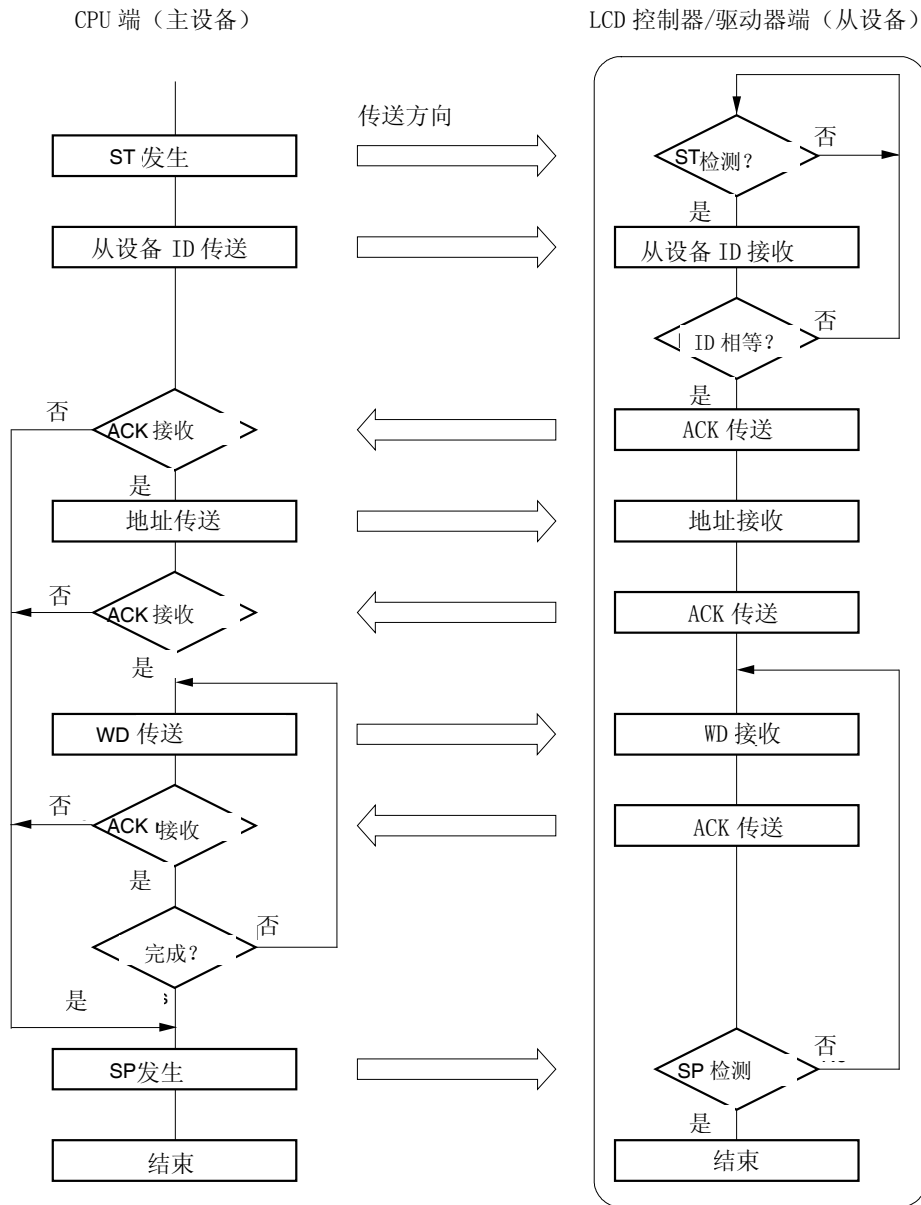


16.7.2 写操作

经由 I²C 总线接口对 LCD 控制器/驱动器写入的处理流程、格式和操作，说明如下。
 通过从设备 ID 和设备地址对 LCD 控制器/驱动器寄存器进行访问(参见图 17-3)。

(1) 处理程序

图 16-30. 写操作的处理程序



备注 ST: 起始条件
 RST: 重启条件
 SP: 停止条件

(2) 通信格式

给每个 LCD 控制器/驱动器的寄存器写入数据，都通过启始条件、从设备 ID、地址、写入数据、停止条件这种顺序。

图 16-31. 写操作（写入两次）通信格式

访问目标	<1> ST	<2> 从设备 ID							<3> R/W	<4> ACK	<5> 地址								<6> ACK
LCDCTL	ST	0	1	1	1	0	0	0	0	ACK	A7	A6	A5	A4	A3	A2	A1	A0	ACK
LCDSEG	ST	0	1	1	1	0	0	1	0	ACK	A7	A6	A5	A4	A3	A2	A1	A0	ACK

<7> 写入数据 1								<8> ACK	<9> 写入数据 2								<10> ACK	<11> SP
D7	D6	D5	D4	D3	D2	D1	D0	ACK	D7	D6	D5	D4	D3	D2	D1	D0	ACK	SP
D7	D6	D5	D4	D3	D2	D1	D0	ACK	D7	D6	D5	D4	D3	D2	D1	D0	ACK	SP

地址

LCDCTL : A7, A6, A5, A4, A3, A2, A1, A0

LCDSEG: A7, A6, A5, A4, A3, A2, A1, A0

地址[※]

LCDCTL : (A7. A6. A5. A4. A3. A2. A1. A0) + 1

注 在 78K0/LF2 中，基于寄存器读/写的开始地址，通过启始条件到停止条件期间不断地传送读/写访问，地址自动增加。基于这个功能，不必每次都对地址进行设置。

注意事项 1. 出现以下访问情况产生停止条件。

- 非指定格式的访问
- 非指定从设备 ID 的访问

2. 为了减小噪声 SDA0 恒为低电平输出，P130 输入为 0(端口寄存器 13 的第 0 位)时，将复位 LCD 控制器/驱动器。

备注

ST: 启始条件

SP: 停止条件

A7 ~ A0: LCDCTL 或 LCDSEG 的地址

(3) 操作

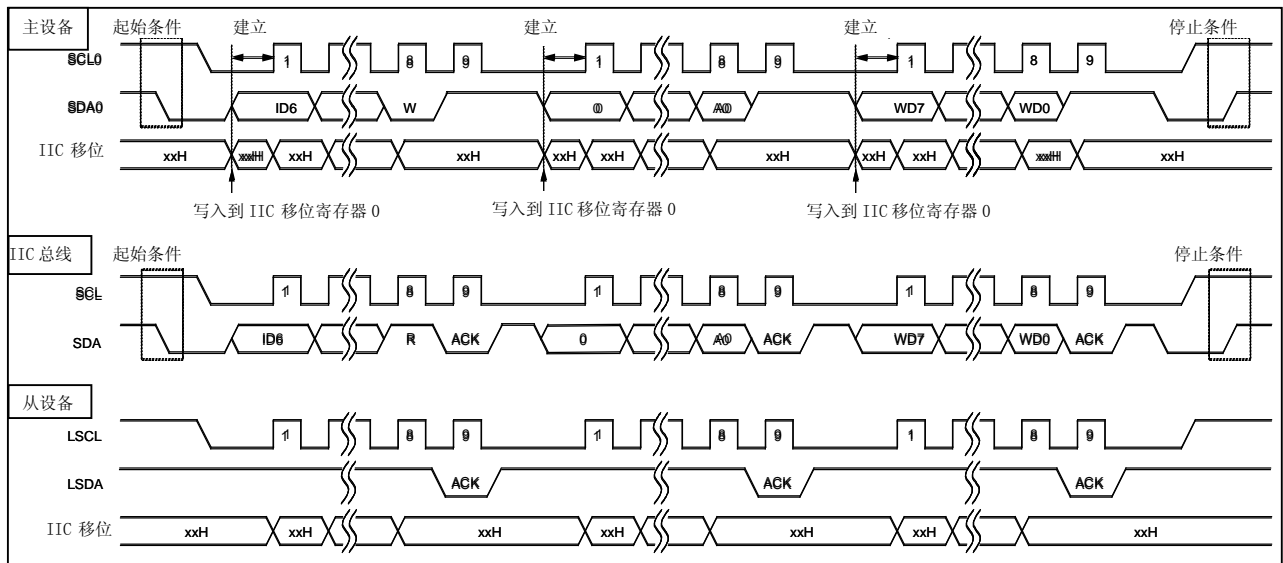
传送两次写入数据的操作流程如下。

<1> ~ <11> 与图 16-31 中 <1> ~ <11> 相对应。

- <1> 传送起始条件。
- <2> 传送从设备 ID (第 1 个 ~ 第 7 个时钟)。
- <3> 传送 R/W 信息 (0) (第 8 个时钟)。
- <4> 接收应答信号(第 9 个时钟的上升沿)。
- <5> 传送写入开始地址 (第 1 个 ~ 第 7 个时钟在 <4>之后)。
- <6> 接收应答信号 (第 9 个时钟的上升沿)。
- <7> 传送写入数据 (第 1 次) (第 1 个 ~ 第 8 个时钟在 <6>之后)。
- <8> 接收应答信号 (第 9 个时钟的上升沿)。
- <9> 传送写入数据 (第 2 次) (第 1 个 ~ 第 8 个时钟在 <8>之后)。
(地址自动加 1)
- <10> 接收应答信号 (第 9 个时钟的上升沿)。
- <11> 传送停止条件。

图 16-32 显示了写入操作的时序图。

图 16-32. 写入操作时序图

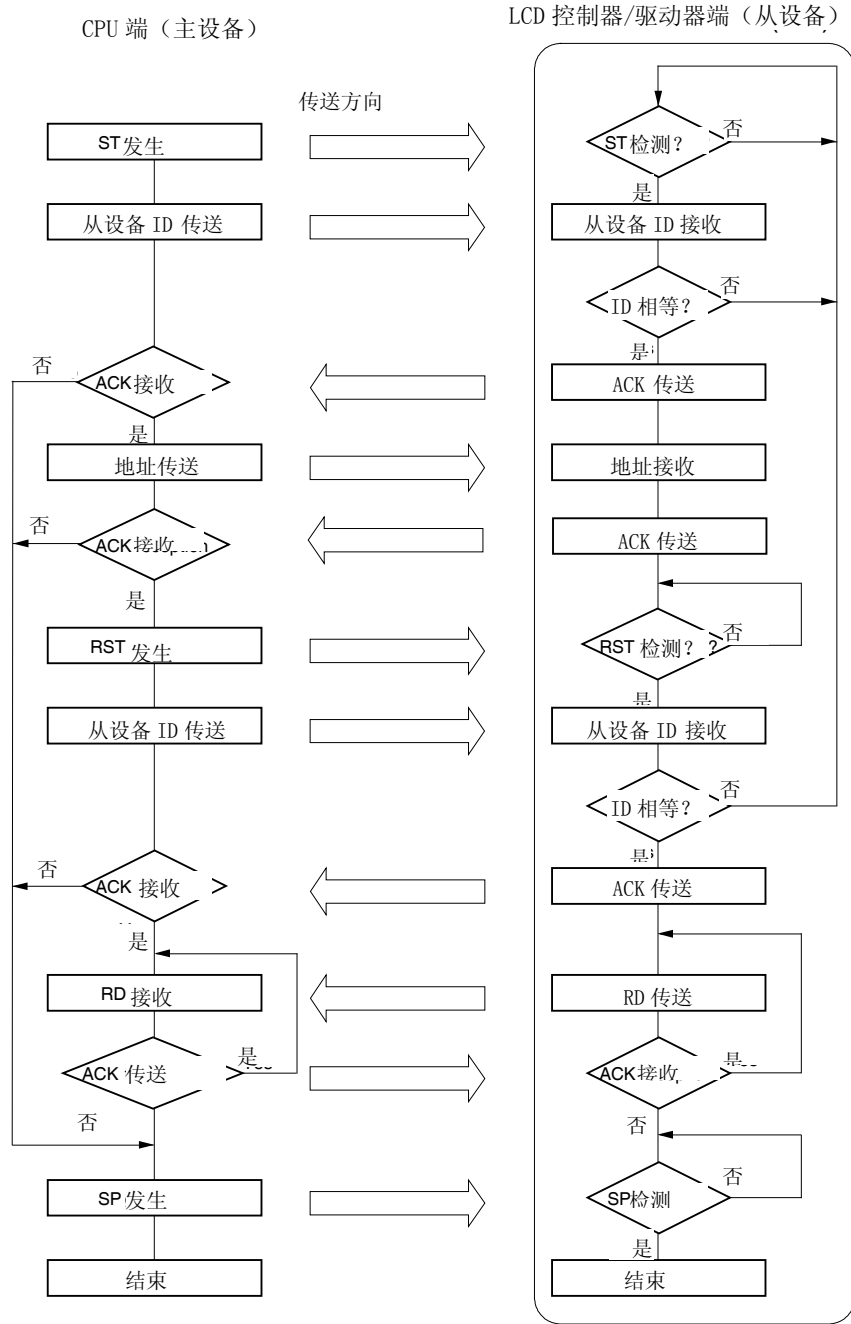


16.7.3 读操作

经由 I²C 总线接口对 LCD 控制器/驱动器读出的处理流程、格式和操作，说明如下。
 通过从设备 ID 和地址对 LCD 控制器/驱动器寄存器进行访问(参见图 17-3)。

(1) 处理流程

图 16-33. 读操作的处理流程



备注 ST: 起始条件
 RST: 重启条件
 SP: 停止条件

(2) 通信格式

从每个 LCD 控制器/驱动器的寄存器中读出数据，都通过启始条件、从设备 ID、地址、读粗数据、停止条件这种顺序。

图 16-34. 读操作(读两次)通信格式

访问目标	<1> ST	<2> 从设备 ID						<3> R/W	<4> ACK	<5> 地址								<6> ACK	
LCDCTL	ST	0	1	1	1	0	0	0	0	ACK	A7	A6	A5	A4	A3	A2	A1	A0	ACK
LCDSEG	ST	0	1	1	1	0	0	1	0	ACK	A7	A6	A5	A4	A3	A2	A1	A0	ACK

<7> RST	<8> 从设备 ID						<9> R/W	<10> ACK	<11> 读入数据 1								<12> ACK	
RST	0	1	1	1	0	0	0	1	ACK	D7	D6	D5	D4	D3	D2	D1	D0	ACK
RST	0	1	1	1	0	0	1	1	ACK	D7	D6	D5	D4	D3	D2	D1	D0	ACK

地址
LCDCTL : A7, A6, A5, A4, A3, A2, A1, A0
LCDSEG: A7, A6, A5, A4, A3, A2, A1, A0

<13> 读入数据 2								<14> ACK	<15> SP
D7	D6	D5	D4	D3	D2	D1	D0	NAK	SP
D7	D6	D5	D4	D3	D2	D1	D0	NAK	SP

地址^注
LCDCTL : (A7. A6. A5. A4. A3. A2. A1. A0) + 1

注 在 78K0/LF2 中，基于寄存器读/写的开始地址，通过启始条件到停止条件期间不断地传送读/写访问，地址自动增加。基于这个功能，不必每次都对地址进行设置。

注意事项 1. 出现以下访问情况产生停止条件。

- 非指定格式的访问
- 非指定从设备 ID 的访问

2. 为了减小噪声 SDA0 恒为低电平输出，P130 输入为 0 (端口寄存器 13 的第 0 位) 时，将复位 LCD 控制器/驱动器。

备注 ST: 启始条件
RST: 重启条件
SP: 停止条件
A7 ~ A0: LCDCTL 或 LCDSEG 的地址

(3) 操作

接收两次读出数据的操作流程如下。

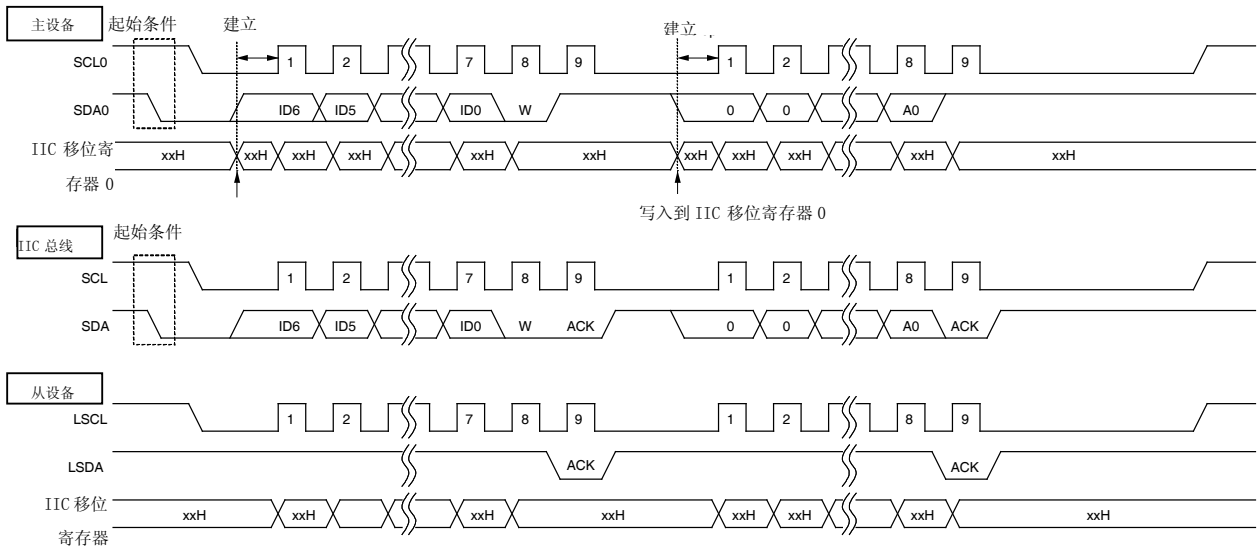
<1> ~ <15> 与图 16-34 中 <1> ~ <15> 相对应。

- <1> 传送起始条件。
- <2> 传送从设备 ID (第 1 次) (第 1 个 ~ 第 7 个时钟)。
- <3> 传送 R/W 信息 (0) (第 8 个时钟)。
- <4> 接收应答信号 (第 9 个时钟的上升沿)。
- <5> 传送读出开始地址 (第 1 个 ~ 第 8 个时钟在<4>之后)。
- <6> 接收应答信号 (第 9 个时钟的上升沿)。
- <7> 传送起始条件。
- <8> 传送从设备 ID (第 2 次) (第 1 个 ~ 第 7 个时钟在<7>之后)。
- <9> 传送 R/W 信息 (1) (第 8 个时钟)。
- <10> 接收应答信号 (第 9 个时钟的上升沿)。
- <11> 接收读出数据 (第 1 次) (第 1 个 ~ 第 8 个时钟在<10>之后)。
- <12> 接收应答信号 (第 8 个时钟的下降沿到第 9 个时钟的上升沿)。
- <13> 接收读出数据 (第 2 次) (第 1 个 ~ 第 8 个时钟在<12>之后)。
(地址自动加 1)
- <14> 传送停止应答信号^注。
- <15> 传送停止条件。

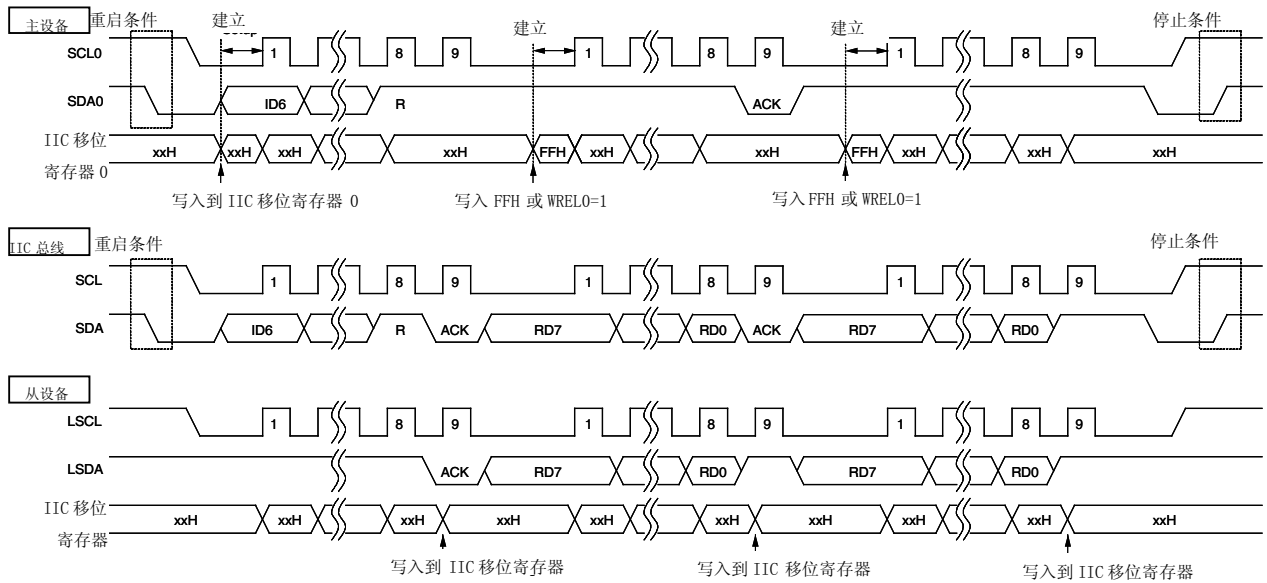
注 当接收数据完成时不要传送应答信号。

图 16-35 显示了读操作时序图。

图 16-35. 读操作时序图



(继续)



第十七章 LCD控制器/驱动器

17.1 LCD 控制器/驱动器的功能

78K0/LF2 的 LCD 控制器/驱动器的功能如下。

- (1) LCD 驱动器的参考电压发生器可转换内部电压泵、外部电阻分压和内部电阻分压。
- (2) 基于自动读取存储器显示数据，自动输出 segment 和 common 信号。
- (3) 5 种不同的显示模式：
 - 静态
 - 1/2 占空比(1/2 偏压)
 - 1/3 占空比(1/2 偏压)
 - 1/3 占空比(1/3 偏压)
 - 1/4 占空比(1/3 偏压)
- (4) 4 种不同的帧频率，选择各自的显示模式。
- (5) μ PD78F037x: Segment 信号输出: 26 (S0 ~ S25), Common 信号输出: 4 (COM0 ~ COM3)
 μ PD78F038x: Segment 信号输出: 36 (S0 ~ S35), Common 信号输出: 4 (COM0 ~ COM3)

表 17-1 列出了每种显示模式的最大显示像素。

表 17-1. 像素列表

(a) μ PD78F037x

LCD 驱动器的参考电压发生器	偏压模式	分时	使用的 Common 信号	Segment 数	最大像素	
•外部电阻分压 •内部电阻分压	-	静态	COM0 (COM1~ COM3)	26	26 (26 路 segment 信号, 1 路 common 信号) ^{注 1}	
		1/2	2		COM0, COM1	52 (26 路 segment 信号, 2 路 common 信号) ^{注 2}
			3		COM0 ~ COM2	78 (26 路 segment 信号, 3 路 common 信号) ^{注 3}
•内部电压泵 •外部电阻分压 •内部电阻分压	1/3	3	COM0 ~ COM2		104 (26 路 segment 信号, 4 路 common 信号) ^{注 4}	
		4	COM0 ~ COM3			

- 注
1. 3-数字 LCD 面板, 每个 digit 有 8-segment 构造。
 2. 6-数字 LCD 面板, 每个 digit 有 4-segment 构造。
 3. 9-数字 LCD 面板, 每个 digit 有 3-segment 构造。
 4. 13-数字 LCD 面板, 每个 digit 有 2-segment 构造。

(b) μ PD78F038x

LCD 驱动器的参考电压发生器	偏压模式	分时	使用的 Common 信号	Segment 数	最大像素	
•外部电阻分压 •内部电阻分压	-	静态	COM0 (COM1~ COM3)	36	36 (36 路 segment 信号, 1 路 common 信号) ^{注 1}	
		1/2	2		COM0, COM1	72 (36 路 segment 信号, 2 路 common 信号) ^{注 2}
			3		COM0 ~ COM2	108 (36 路 segment 信号, 3 路 common 信号) ^{注 3}
•内部电压泵 •外部电阻分压 •内部电阻分压	1/3	3	COM0 ~ COM2		144 (36 路 segment 信号, 4 路 common 信号) ^{注 4}	
		4	COM0 ~ COM3			

- 注
1. 4-数字 LCD 面板, 每个 digit 有 8-segment 构造。
 2. 9-数字 LCD 面板, 每个 digit 有 4-segment 构造。
 3. 13-数字 LCD 面板, 每个 digit 有 3-segment 构造。
 4. 18-数字 LCD 面板, 每个 digit 有 2-segment 构造。

17.2 LCD 控制器/驱动器的配置

LCD 控制器/驱动器包括以下硬件。

LCD 控制器/驱动器包括两模块：控制模块 LCDSEG 和控制 LCD 寄存器设置和模式设置的模块 LCDCTL。

表 17-2. LCD 控制器/驱动器的配置

项目	配置
LCD 控制器/驱动器	显示输出 (LCDSEG) μ PD78F037x: 26 路 segment 信号, 4 路 common 信号 (COM0 ~ COM3) μ PD78F038x: 36 路 segment 信号, 4 路 common 信号 (COM0 ~ COM3)
	控制寄存器 (LCDCTL) LCD 模式设置寄存器(LCDMD) LCD 显示模式寄存器(LCDM) LCD 时钟控制寄存器(LCDC) LCD 升压控制寄存器 0 (VLCG0)
CPU	控制寄存器 时钟输出选择寄存器(CKS) 端口寄存器 13 (P13) 端口模式寄存器 14 (PM14)

图 17-1. LCD 控制器/驱动器的硬件配置

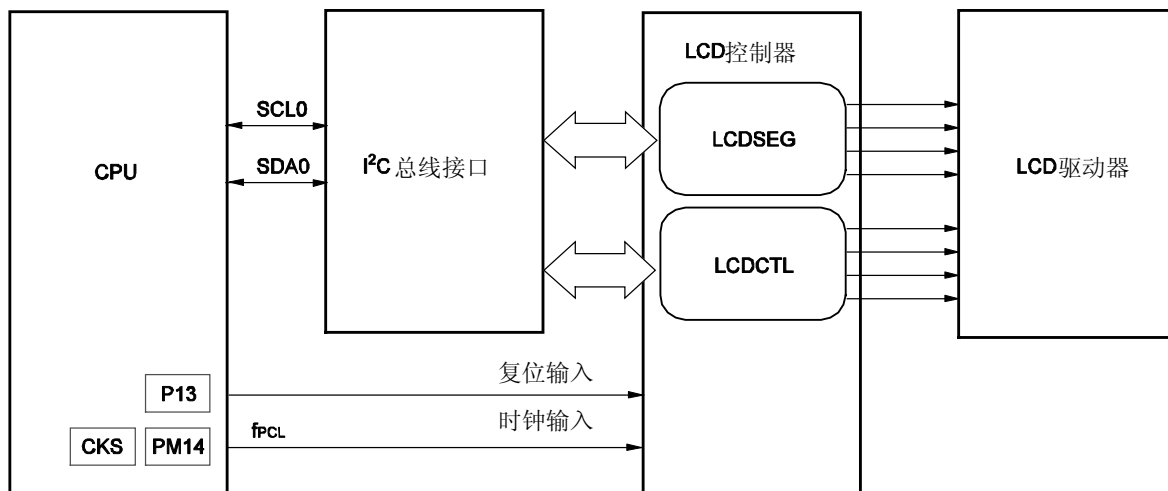
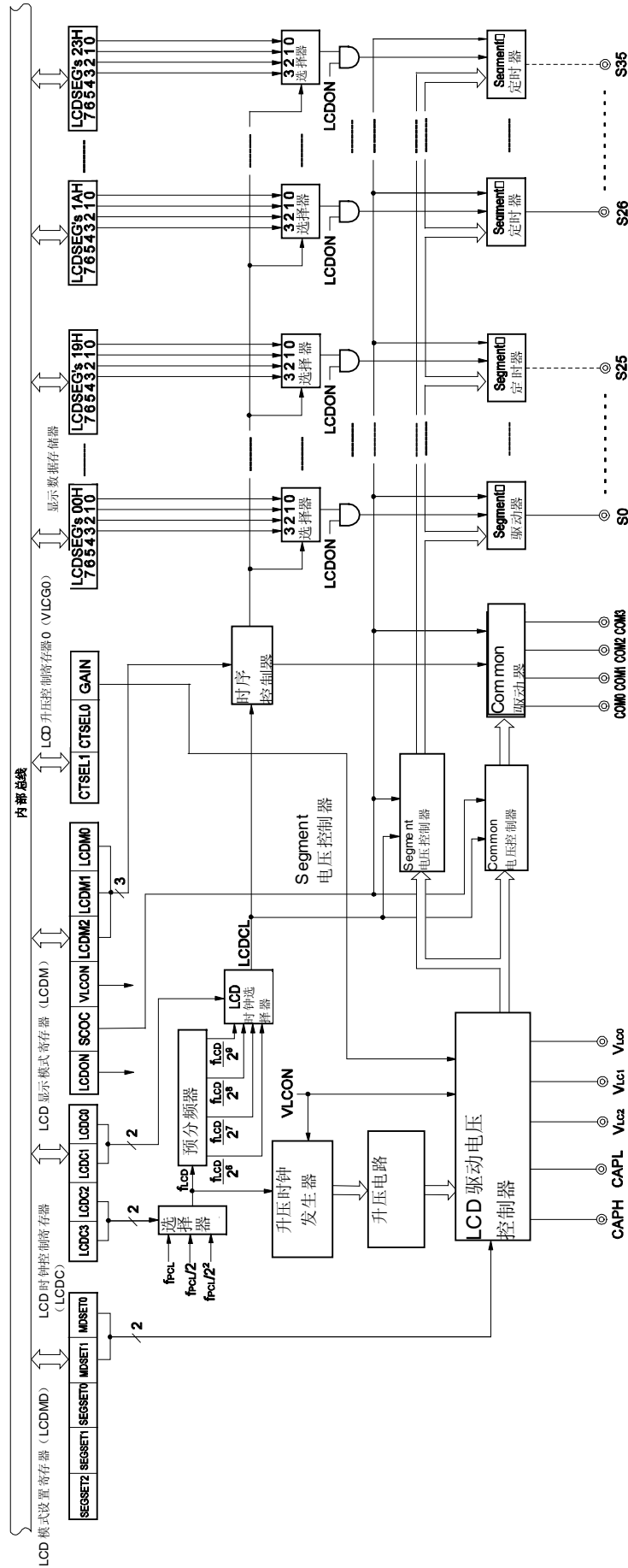


图 17-2 LCD 控制器/驱动器方框图



备注 f_{pcl}: 由时钟输出控制器产生的时钟

只适用于 μ PD78F036x

图 17-3 显示了 LCD 控制器/驱动器的控制寄存器和图 17-4 显示了 LCD 显示 RAM。

图 17-3. LCD 控制器/驱动器的控制寄存器

地址	位								寄存器
	7	6	5	4	3	2	1	0	
LCDCTL's 03H	CTSEL1	CTSEL0	0	0	0	0	0	GAIN	→ VLCOG0
02H	0	0	0	0	LCDC3	LCDC2	LCDC1	LCDC0	→ LCDC
01H	LCDON	SCOC	VLCON	0	0	LCDM2	LCDM1	LCDM0	→ LCDM
LCDCTL's 00H	SEGSET2	SEGSET1	SEGSET0	0	0	0	MDSET1	MDSET0	→ LCDMD

图 17-4. LCD 显示 RAM

地址	位								Segment
	7	6	5	4	3	2	1	0	
LCDSEG's 23H ^注	0	0	0	0					→ S35 ^注
22H ^注	0	0	0	0					→ S34 ^注
21H ^注	0	0	0	0					→ S33 ^注
20H ^注	0	0	0	0					→ S32 ^注
1FH ^注	0	0	0	0					→ S31 ^注
1EH ^注	0	0	0	0					→ S30 ^注
1DH ^注	0	0	0	0					→ S29 ^注
1CH ^注	0	0	0	0					→ S28 ^注
1BH ^注	0	0	0	0					→ S27 ^注
1AH ^注	0	0	0	0					→ S26 ^注
19H	0	0	0	0					→ S25
18H	0	0	0	0					→ S24
17H	0	0	0	0					→ S23
16H	0	0	0	0					→ S22
15H	0	0	0	0					→ S21
14H	0	0	0	0					→ S20
13H	0	0	0	0					→ S19
12H	0	0	0	0					→ S18
11H	0	0	0	0					→ S17
10H	0	0	0	0					→ S16
0FH	0	0	0	0					→ S15
0EH	0	0	0	0					→ S14
0DH	0	0	0	0					→ S13
0CH	0	0	0	0					→ S12
0BH	0	0	0	0					→ S11
0AH	0	0	0	0					→ S10
09H	0	0	0	0					→ S9
08H	0	0	0	0					→ S8
07H	0	0	0	0					→ S7
06H	0	0	0	0					→ S6
05H	0	0	0	0					→ S5
04H	0	0	0	0					→ S4
03H	0	0	0	0					→ S3
02H	0	0	0	0					→ S2
01H	0	0	0	0					→ S1
LCDSEG's 00H	0	0	0	0					→ S0

↑ ↑ ↑ ↑
 Common COM3 COM2 COM1 COM0

注 仅限于μPD78F038x

备注 第 4 位 ~ 第 7 位恒为 0。

17.3 控制 LCD 控制器/驱动器的寄存器

LCD 控制器/驱动器使用以下 7 个寄存器进行控制。

- LCD 模式设置寄存器 (LCDMD)
- LCD 显示模式寄存器 (LCDM)
- LCD 时钟控制寄存器 (LCDC)
- LCD 升压控制寄存器 0 (VLCG0)
- 时钟输出选择寄存器 (CKS)
- 端口寄存器 13 (P13)
- 端口模式寄存器 14 (PM14)

(1) LCD 模式设置寄存器 (LCDMD)

LCDMD 设置 segment 使用个数和 LCD 参考电压发生器。

可由 8 位存储器操作指令设置 LCDMD。

复位信号将 LCDMD 设置为 00H。

图 17-5. LCD 模式设置寄存器的格式

地址: LCDCTL's 00H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
LCDMD	SEGSET2	SEGSET1	SEGSET0	0	0	0	MDSET1	MDSET0

SEGSET2	SEGSET1	SEGSET0	设置Segment个数
0	1	0	26 (μ PD78F037x)
0	1	1	36 (μ PD78F038x)
其它			禁止设置

MDSET1	MDSET0	选择LCD 参考电压发生器
0	0	外部电阻分压方式
0	1	内部电阻分压方式
1	×	内部电压泵方式

- 注意事项**
1. 第 2 位 ~ 第 4 位必须为 0。
 2. 复位后 LCDMD 仅可设置一次。

(2) LCD 显示模式寄存器 (LCDM)

LCDM 可以选择液晶是否显示。还可以选择是否允许 segment 引脚/common 引脚输出、升压电路操作和显示模式。

可由 8 位存储器操作指令设置 LCDM。

复位信号将 LCDM 设置为 00H。

图 17-6. LCD 显示模式寄存器格式

地址: LCDCTL's 01H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
LCDM	LCDON	SCOC	VLCON	0	0	LCDM2	LCDM1	LCDM0

LCDON	LCD 显示允许/禁止
0	显示关 (取消全部segment 输出)
1	显示

SCOC	Segment 引脚/common 引脚输出控制 ^注
0	输出低电平给segment/common引脚
1	输出非选择电压给segment引脚和输出 LCD 波形给common引脚

VLCON	升压电路操作允许/禁止 ^注
0	无内部升压
1	允许内部升压

LCDM2	LCDM1	LCDM0	LCD 控制器/驱动器显示模式选择			
			电阻分压方式		升压方式	
			分时	偏压模式	分时	偏压模式
0	0	0	4	1/3	4	1/3
0	0	1	3	1/3	3	1/3
0	1	0	2	1/2	4	1/3
0	1	1	3	1/2	3	1/3
1	0	0	静态		禁止设置	
其它			禁止设置			

注 当未使用 LCD 显示时, SCOC 和 VLCON 必须设置为 0 以降低功耗。

注意事项 1. 第 3 位和第 4 位必须设置为 0。

2. 按以下步骤操作 VLCON。

A. 转换显示状态从开到关后停止升压：

1) LCDON = 0 设置显示关状态。

2) SCOC = 0 禁止所有的 segment 缓冲器和 common 缓冲器输出。

3) VLCON = 0 停止升压。

B. 在显示开状态期间停止升压：

禁止设置。确保显示关后停止升压。

C. 升压停止状态设置显示开：

1) VLCON = 1 开始升压，需等待升压等待时间(t_{VWAIT}) (参见第二十九章 电气特性)。

2) SCOC = 1 设置所有的 segment 缓冲器和 common 缓冲器在非显示输出状态。

3) LCDON = 1 显示开。

(3) LCD 时钟控制寄存器 (LCDC)

LCDC 指定 LCD 源时钟和 LCD 时钟。

依据 LCD 时钟和分时来决定帧频率。

可由 8 位存储器操作指令设置 LCDC。

复位信号将 LCDC 设置为 00H。

图 17-7. LCD 时钟控制寄存器的结构

地址: LCDCTL's 02H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
LCDC	0	0	0	0	LCDC3	LCDC2	LCDC1	LCDC0

LCDC3	LCDC2	LCD 源时钟(f_{LCD}) 选择 ^注
0	x	f_{PCL} (时钟输出控制器产生时钟)
1	0	$f_{PCL}/2$
1	1	$f_{PCL}/2^2$

LCDC1	LCDC0	LCD 时钟 (LCDCL) 选择
0	0	$f_{LCD}/2^6$
0	1	$f_{LCD}/2^7$
1	0	$f_{LCD}/2^8$
1	1	$f_{LCD}/2^9$

注 指定 LCD 源时钟 (f_{LCD}) 频率最小为 32 kHz。

注意事项 1. 第 4 位 ~ 第 7 位必须为 0。

2. 在改变 LCDC 设置前，要确保停止升压 (VLCON = 0)。

3. 设置帧频率要低于 128 Hz。

(4) LCD 升压控制寄存器 0 (VLCG0)

VLCG0 在升压操作期间控制升压电平。

可由 8 位存储器操作指令设置 VLCG0。

复位信号将 VLCG0 设置为 00H。

<R>

图 17-8. LCD 升压控制寄存器 0 的格式

地址: LCDCTL's 03H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
VLCG0	CTSEL1	CTSEL0	0	0	0	0	0	GAIN

GAIN	参考电压 (V _{LC2}) 选择 ^{注1}
0	1.5 V (使用4.5V LCD 面板的规范)
1	1.0 V (使用3V LCD 面板的规范)

CTSEL1	CTSEL0	对照调节器 (TYP.) ^{注2}					
		V _{LC0}		V _{LC1}		V _{LC2}	
		GAIN = 0	GAIN = 1	GAIN = 0	GAIN = 1	GAIN = 0	GAIN = 1
1	0	4.89 V	3.39 V	3.27 V	2.27 V	1.63 V	1.13 V
1	1	4.71 V	3.21 V	3.13 V	2.13 V	1.57 V	1.07 V
0	0	4.50 V	3.00 V	3.00 V	2.00 V	1.50 V	1.00 V
0	1	4.29 V	2.79 V	2.87 V	1.87 V	1.43 V	0.93 V

- 注 1. 依据使用的 LCD 面板规范进行选择。
2. 设置这几位使 LV_{DD} 升压后变成 2.0 ~ 5.5 V。

- 注意事项 1. 第 1 位 ~ 第 5 位必须设置为 0。
2. 在改变 VLCG0 设置前, 必须停止升压 (VLC_{CON} = 0)。

(5) 时钟输出选择寄存器 (CKS)

CKS 可以控制时钟(PCL)输出到 LCD 控制器/驱动器并设置输出时钟。

可由 1 位或 8 位存储器操作指令设置 CKS。

复位信号将 CKS 设置为 00H。

图 17-9. 时钟输出选择寄存器的格式

地址: FF40H 复位后: 00H R/W

符号	7	6	5	<4>	3	2	1	0
CKS	0	0	0	CLOE	CCS3	CCS2	CCS1	CCS0

CLOE	PM140	时钟输出到LCD控制器/驱动器规范 ^注
1	0	允许时钟输出到LCD控制器/驱动器
其它		禁止时钟输出到LCD控制器/驱动器

CCS3	CCS2	CCS1	CCS0	PCL 输出时钟选择			
				f _{SUB} = 32.768 kHz	f _{PRS} = 10 MHz	f _{PRS} = 20 MHz	
0	1	1	0	f _{PRS} /2 ⁶	-	156.25 kHz	312.5 kHz
0	1	1	1	f _{PRS} /2 ⁷		78.125 kHz	156.25 kHz
1	0	0	0	f _{SUB}	32.768 kHz	-	
其它				禁止设置			

注 PCL 时钟输出与否要结合 PM140 设置 (参见 (7) 端口模式寄存器 14 (PM14))。

<R>

注意事项 1.当时钟操作停止(CLOE = 0)时设置 CCS3 ~ CCS0。

2.第 5 位 ~ 第 7 位必须设置为 0。

备注

1. f_{PRS}: 外围硬件时钟振荡频率

2. f_{SUB}: 副系统时钟振荡频率

(6) 端口寄存器 13 (P13)

P13 控制 LCD 控制器/驱动器的复位。

当使用 LCD 控制器/驱动器时，设置 P130 为 1。

可由 1 位或 8 位存储器操作指令设置 P13。

复位信号将 P13 设置为 00H。

图 17-10. 端口寄存器 13 的格式

地址: FF0DH 复位后: 00H (输出锁存) R/W

符号	7	6	5	4	3	2	1	0
P13	0	0	0	0	0	0	0	P130

P130	LCD 控制器/驱动器的复位控制
0	复位状态设置
1	复位状态释放

(7) 端口寄存器 14 (PM14)

PM14 控制时钟输出到 LCD 控制器/驱动器。

当使用 LCD 控制器/驱动器时，设置 P140 为 0。

可由 1 位或 8 位存储器操作指令设置 PM14。

复位信号将 P14 设置为 FFH。

图 17-11. 端口寄存器 14 的格式

地址: FF2EH 复位后: FFH R/W

符号	7	6	5	4	3	2	1	0
PM14	1	1	1	1	1	1	1	PM140

PM140	控制时钟输出到LCD 控制器/驱动器
0	允许时钟输出到LCD 控制器/驱动器
1	禁止时钟输出到LCD 控制器/驱动器

17.4 设置 LCD 控制器/驱动器

设置 LCD 控制器/驱动器按以下步骤。

(1) 升压方式

- 按以下流程转换 LCD 控制器/驱动器由复位状态到显示状态

- <1> 设置 P130 = 1，释放复位状态。
- <2> 使用时钟输出寄存器(CKS)设置输出时钟。
- <3> 设置 PM140 = 0，将其设置为输出模式。
- <4> 设置 CLOE (CKS 的第 4 位)为 1 允许时钟输出。
- <5> 设置 MDSET1(LCDMD 的第 1 位)为 1，设置升压方式。
(初始设置：外部电阻分压方式)
- <6> 设置 LCD 显示 RAM (第 0 位 ~ 第 3 位) 中的 LCD 显示数据初值。
- <7> 使用 LCDM0, LCDM1, 和 LCDM2 (LCD 显示模式寄存器(LCDM)的第 0 位,第 1 位和第 2 位) 来设置显示模式。(不能设置 1/2 偏压模式和静态模式)。
- <8> 使用 LCD 时钟控制寄存器 (LCDC)设置 LCD 时钟。
- <9> 使用升压控制寄存器 0(VLCG0)设置升压电平和对照。
GAIN = 0: $V_{LC0} = 4.5\text{ V}$, $V_{LC1} = 3\text{ V}$, $V_{LC2} = 1.5\text{ V}$
GAIN = 1: $V_{LC0} = 3\text{ V}$, $V_{LC1} = 2\text{ V}$, $V_{LC2} = 1\text{ V}$
- <10> 设置 VLCON (LCDM 的第 5 位) 为 1 允许升压。
- <11> 设置 VLCON 后等待升压等待时间 (t_{VWAIT}) (参见第二十九章 电气特性)。
- <12> 设置 SCOC (LCDM 的第 6 位)为 1 输出非选择的电压。
- <13> 设置 LCDON (LCDM 第 7 位)为 1 和设置数据存储器与显示内容相一致的数据，开始输出相应的数据。

以后的流程同上，在数据存储器设置显示数据。

备注 由于使用 I²C 总线可对寄存器进行位操作。

(2) 电阻分压方式

•按以下流程转换 LCD 控制器/驱动器由复位状态到显示状态

<1> 设置 P130 = 1 释放复位状态。

<2> 使用时钟输出寄存器(CKS)设置输出时钟。

<3> 设置 PM140 = 0 将其设置为输出模式。

<4> 设置 CLOE (CKS 的第 4 位)为 1 允许时钟输出。

<5> 使用 MDSET0 和 MDSET1 (LCDMD 的第 0 位和第 1 位) 设置内部升压方式。

(MDSET0, MDSET1 = 0, 0: 外部电阻分压方式,
MDSET0, MDSET1 = 0, 1: 内部电阻分压方式)

<6> 设置 LCD 显示 RAM (第 0 位 ~ 第 3 位) 中的 LCD 显示数据初值。

<7> 使用 LCDM0, LCDM1, 和 LCDM2 (LCD 显示模式寄存器(LCDM)的第 0 位,第 1 位和第 2 位) 来设置显示模式。

<8> 使用 LCD 时钟控制寄存器 (LCDC)设置 LCD 时钟。

<9> 设置 SCOC (LCDM 的第 6 位)为 1 输出非选择的电压。

<10> 设置 LCDON (LCDM 第 7 位)为 1 和设置数据存储器与显示内容相一致的数据, 开始输出相应的数据。

以后的流程同上, 在数据存储器设置显示数据。

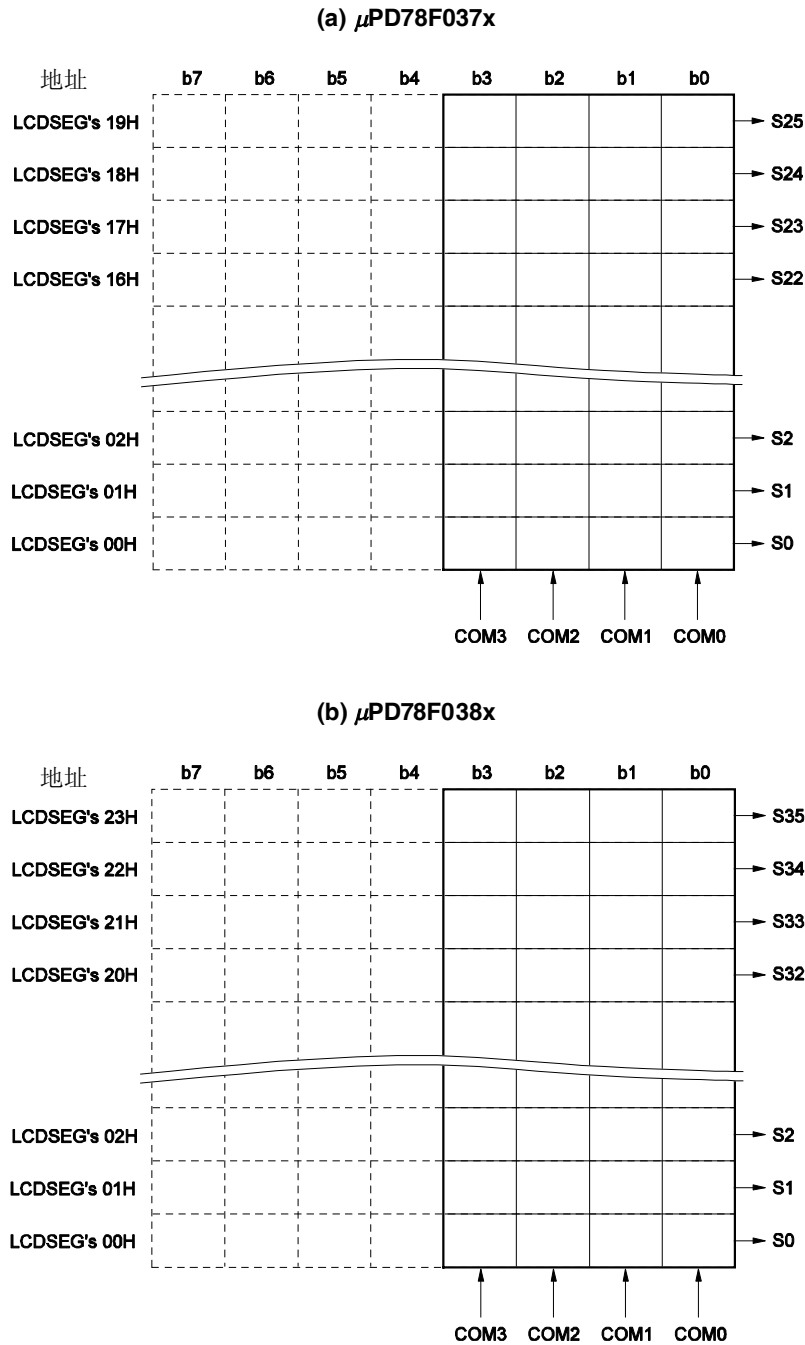
备注 由于使用 I²C 总线可对寄存器进行位操作。

17.5 LCD 显示数据存储器

LCD 显示数据存储器是 LCDSEG 地址 00H ~ 27H 的映射。使用 LCD 控制器/驱动器，LCD 显示数据存储器中的数据在 LCD 面板上。

图 17-12 显示了 LCD 显示数据存储器 and segment/common 输出的关系。

图 17-12. LCD 显示数据存储器内容和 segment/common 输出的关系



注意事项 LCD 显示数据存储器的高 4 位未分配，确保其值为 0。

17.6 Common 和 Segment 信号

当相应的 common 和 segment 信号产生高于指定电压(LCD 驱动电压, V_{LCD})的位差, 点亮 LCD 面板上的每个像素。当产生低于 V_{LCD} 位差时关闭像素。

应用 DC 电压到 LCD 面板驱动 common 和 segment 信号, 可能引发坏点。为了避免这个问题, 使用 AC 电压驱动 LCD 面板。

(1) Common 信号

依照表 17-3 分时列表列出的分时选择 common 信号。在静态显示模式中, COM0 ~ COM3 输出相同的信号。

在 2 分时模式, COM2 和 COM3 引脚悬空。3 分时模式, COM3 引脚悬空。

表 17-3. COM 信号

COM信号 \ 分时数	COM0	COM1	COM2	COM3
静态显示模式				
2 分时模式			悬空	悬空
3 分时模式				悬空
4 分时模式				

(2) Segment 信号

(a) μ PD78F037x

segment 信号符合 26 字节的 LCD 显示数据存储器 (LCDSEG 的 00H ~ 19H)。每个字节的第 0 位, 第 1 位, 第 2 位和第 3 位分别是同步读出的 COM0, COM1, COM2 和 COM3。如果某一位为 1, 则将其转换为选择电压, 如果为 0, 则将其转换为非选择电压。转换结果由 segment (S0 ~ S25) 引脚输出。

(a) μ PD78F037x

segment 信号符合 36 字节的 LCD 显示数据存储器 (LCDSEG 的 00H ~ 23H)。每个字节的第 0 位, 第 1 位, 第 2 位和第 3 位分别是同步读出的 COM0, COM1, COM2 和 COM3。如果某一位为 1, 则将其转换为选择电压, 如果为 0, 则将其转换为非选择电压。转换结果由 segment (S0 ~ S35) 引脚输出。

利用上述所给出的信息, 核对 front-surface 电极 (对应于 segment 信号) 和 rear-surface 电极 (对应于 common 信号) 的什么组合可以在 LCD 显示数据存储器内组成显示模式, 并以一对一的原则写入符合所期望的显示模式的位数据。

LCD 显示数据存储器的第 1 位 ~ 第 3 位、第 2 位和第 3 位、第 3 位分别不可用于 LCD 静态显示、2 分时模式和 3 分时模式。这些位用于非显示功能。

LCD 显示数据存储器第 4 位 ~ 第 7 位恒为 0。

(3) common 和 segment 信号的输出波形

表 17-4 列出了 common 和 segment 信号的输出电压。

当 common 和 segment 信号都为选择电压，将获得显示器开显示电压 $\pm V_{LCD}$ 。其他情况为关显示电压。

表 17-4. LCD 驱动电压

(a) 静态显示模式

Segment 信号		选择信号电平	非选择信号电平
		V_{LCO}/V_{LCO}	V_{LCO}/V_{SS}
Common 信号		V_{LCO}/V_{LCO}	V_{LCO}/V_{SS}
V_{LCO}/V_{SS}		$-V_{LCD}/+V_{LCD}$	0 V/0 V

(b) 1/2 偏压模式

Segment 信号		选择信号电平	非选择信号电平
		V_{LCO}/V_{LCO}	V_{LCO}/V_{SS}
Common 信号		V_{LCO}/V_{LCO}	V_{LCO}/V_{SS}
选择信号电平	V_{LCO}/V_{SS}	$-V_{LCD}/+V_{LCD}$	0 V/0 V
非选择信号电平	$V_{LC1} = V_{LC2}$	$-V_{LCD}/+V_{LCD}$	$+V_{LCD}/-V_{LCD}$

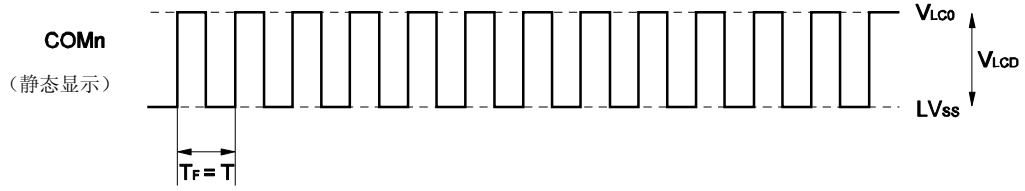
(c) 1/3 偏压模式

Segment 信号		选择信号电平	非选择信号电平
		V_{LCO}/V_{LCO}	V_{LC1}/V_{LC2}
Common 信号		V_{LCO}/V_{LCO}	V_{LC1}/V_{LC2}
选择信号电平	V_{LCO}/V_{SS}	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{3}V_{LCD}/+V_{LCD}$
非选择信号电平	V_{LC2}/V_{LC1}	$-V_{LCD}/+V_{LCD}$	$-\frac{1}{3}V_{LCD}/+V_{LCD}$

图 17-13 显示了 common 信号波形，图 17-14 显示了 common 和 segment 信号的电压和相位。

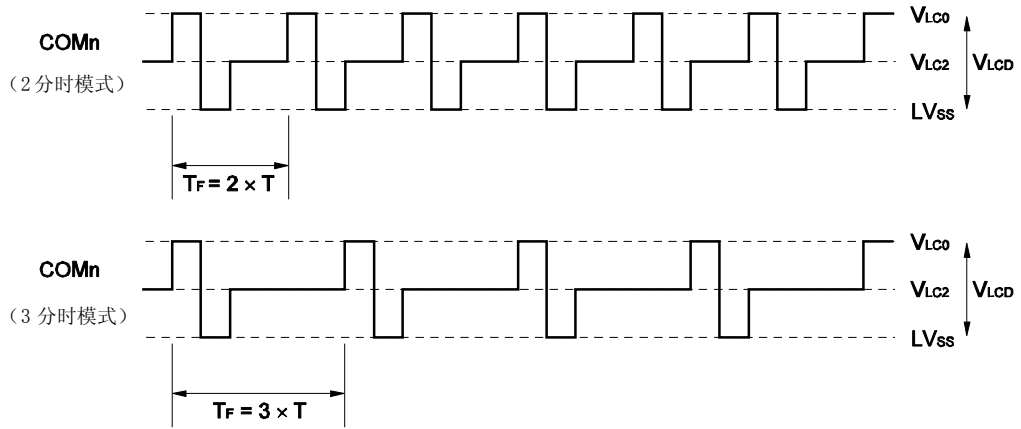
图 17-13. Common 信号波形

(a) 静态显示模式



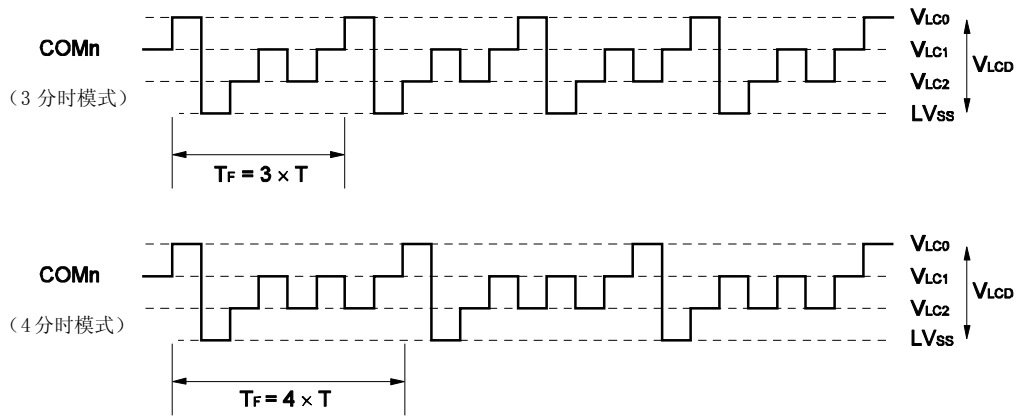
T: 1 个 LCD 时钟周期 T_F : 帧频率

(b) 1/2 偏压模式



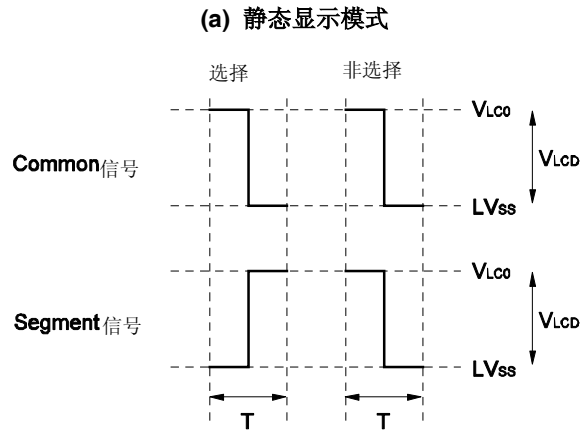
T: 1 个 LCD 时钟周期 T_F : 帧频率

(b) 1/3 偏压模式

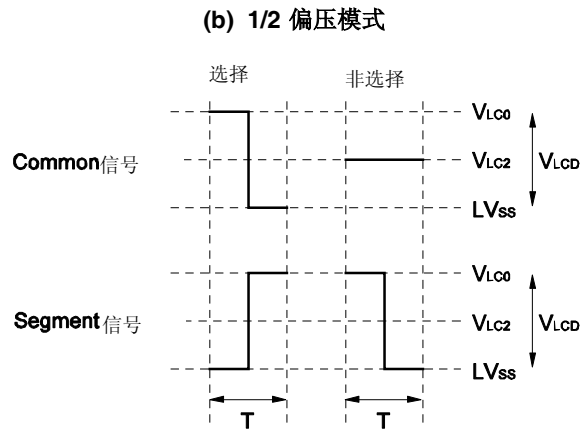


T: 1 个 LCD 时钟周期 T_F : 帧频率

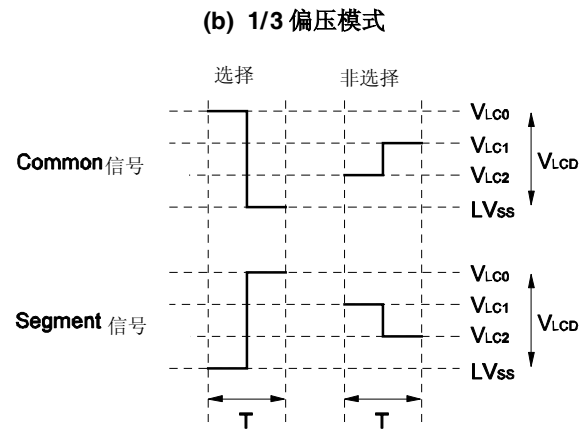
图 17-14. Common 和 Segment 信号的电压和相位



T: 1 个 LCD 时钟周期



T: 1 个 LCD 时钟周期



T: 1 个 LCD 时钟周期

17.7 显示模式

17.7.1 静态显示示例

图 17-16 显示了如何把 3 数字 LCD 面板显示形成图形，按图 17-15 连接 78K0/LF2 芯片 segment 信号(S0 ~ S23) 和 common 信号(COM0)。这个例子在 LCD 面板上显示数字"12.3"。显示数据存储器(LCDSEG 的 00H ~ 17H 地址)的内容符合这个显示。


以下描述集中在数字"2"( 显示，显示在第二个字段。为了在 LCD 面板显示"2"，必须按照 common 信号 COM0 的时序，按照表 17-5 所示从 S8 ~ S15 引脚输出选择信号或非选择信号电压；参见图 17-15 segment 信号和 LCD segment 的关系。

表 17-5. 选择和非选择电压(COM0)

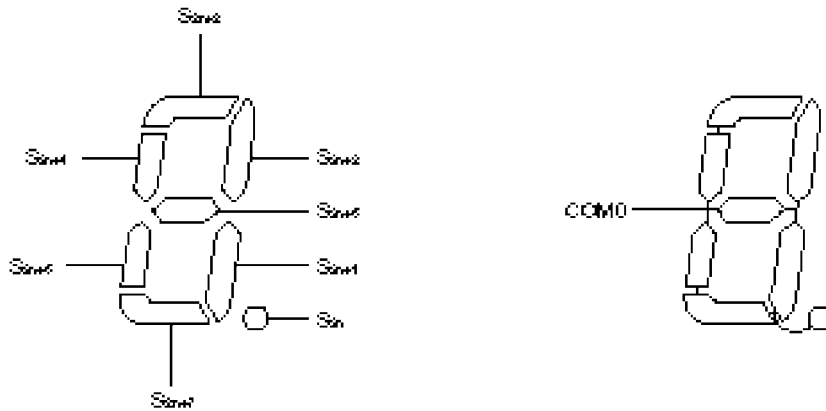
Segment \ Common	S8	S9	S10	S11	S12	S13	S14	S15
COM0	选择	非选择	选择	选择	非选择	选择	选择	选择

根据表 17-5，它决定显示数据存储器区域的第 0 位的模式(LCDSEG 的 08H ~ 0FH)必须是 10110111。

图 17-17 显示了 S11 和 S12, 和 COM0 的 LCD 驱动波形。当选择信号电压以 COM0 的时序应用到 S11 时，LCD segment 产生一个相应的交替矩形波形，+VLCD/-VLCD。

COM1 ~ COM3 和 COM0 的波形相同。所以，COM0 ~ COM3 可以连接在一起增加驱动能力。

图 17-15. 静态 LCD 显示图形和电极连接



备注 n = 0 ~ 2

图 17-16. 静态 LCD 面板连接示例

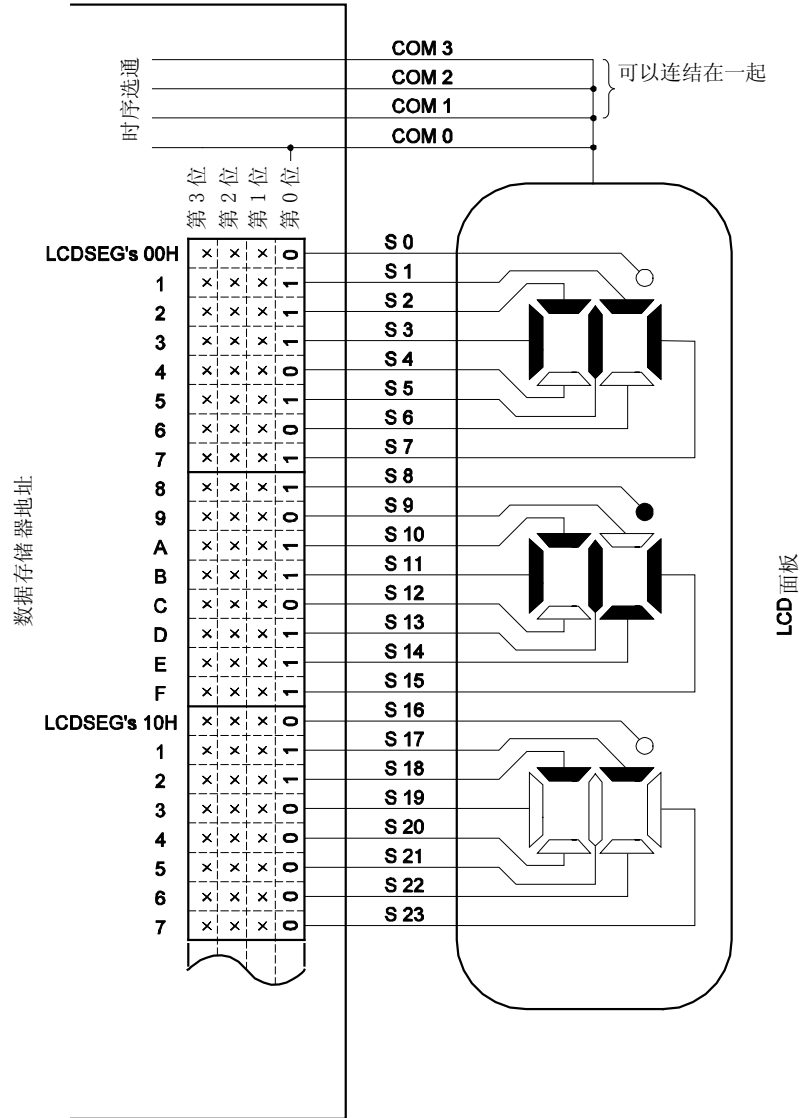
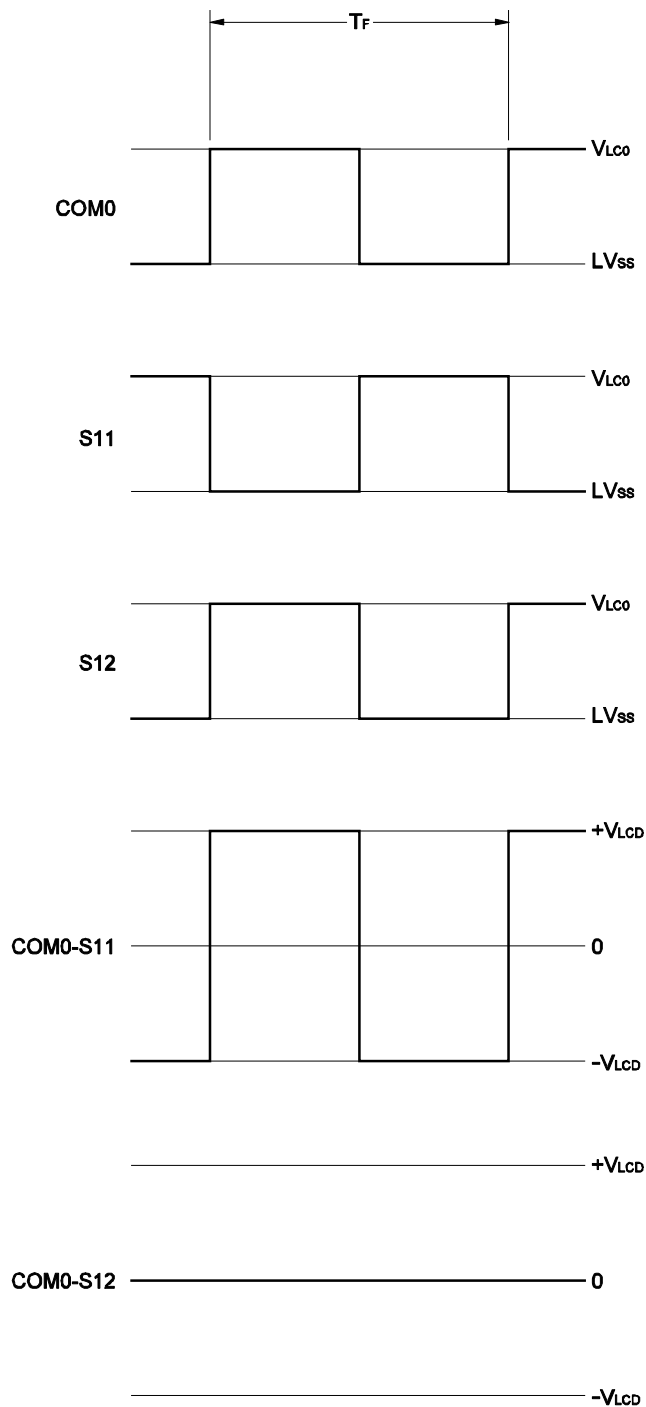


图 17-17. 静态 LCD 驱动波形示例



17.7.2 2 分时显示示例

图 17-19 显示了如何把 6 数字 LCD 面板显示形成图形，按图 17-18 连接 78K0/LF2 芯片 segment 信号(S0 ~ S23) 和 common 信号(COM0 和 COM1)。这个例子在 LCD 面板上显示数字"12345.6"。显示数据存储区(LCDSEG 的 00H ~ 17H 地址)的内容符合这个显示。

以下描述集中在数字"3"()显示，显示在第四个字段。为了在 LCD 面板显示"3"，必须按照 common 信号 COM0 和 COM1 的时序，按照表 17-6 所示从 S12 ~ S15 引脚输出选择信号或非选择信号电压；参见图 17-18 segment 信号和 LCD segment 的关系。

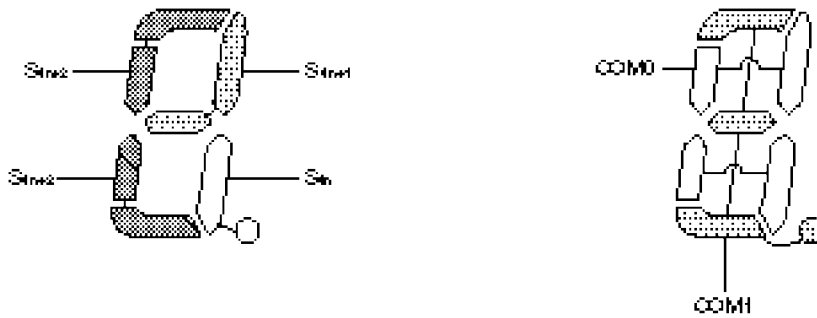
表 17-6. 选择和非选择电压 (COM0 和 COM1)

Segment \ Common	S12	S13	S14	S15
COM0	选择	选择	非选择	非选择
COM1	非选择	选择	选择	选择

依照表 17-6，它决定了 S15 相应的显示数据存储区(LCDSEG 的 0FH)必须是 xx10。

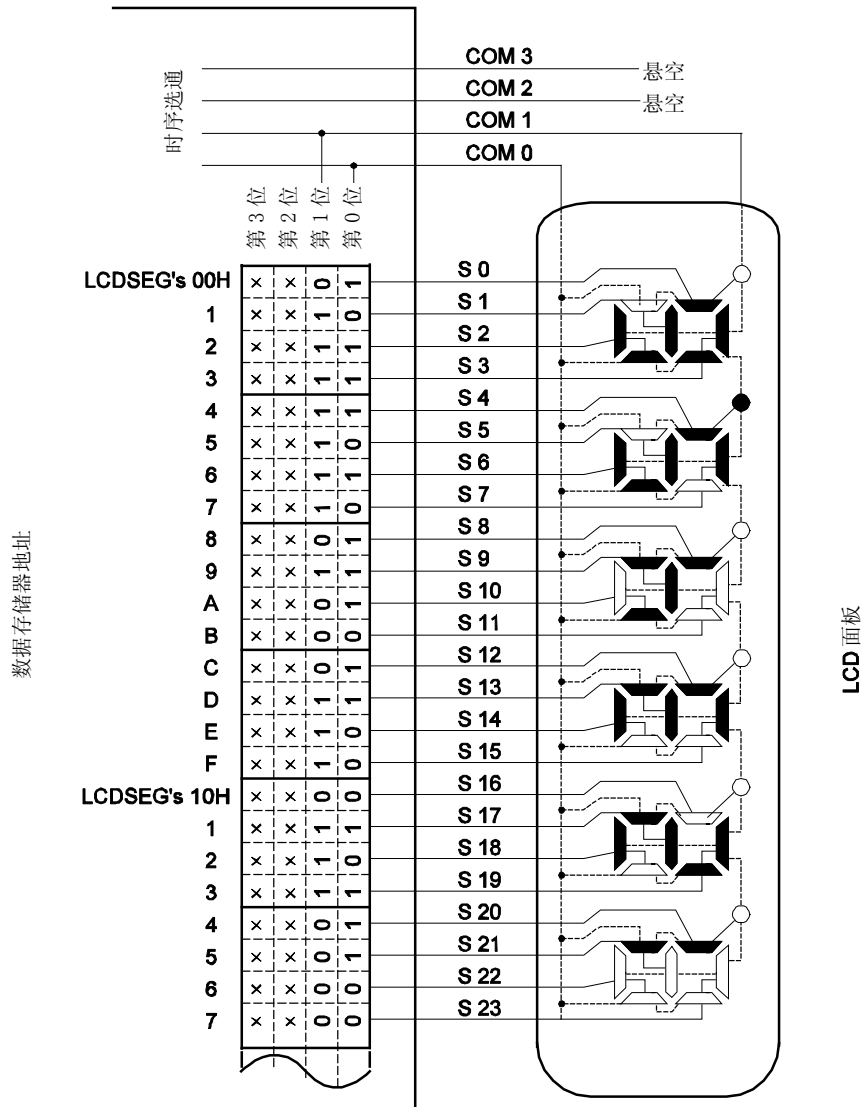
图 17-20 显示了 S15 信号和每个 common 信号之间的 LCD 驱动波形。当选择电压以 COM1 的时序被应用于 S15 时，LCD segment 产生一个相应的交替矩形波形，+VLCD/-VLCD。

图 17-18. 2 分时 LCD 显示图形和电极连接



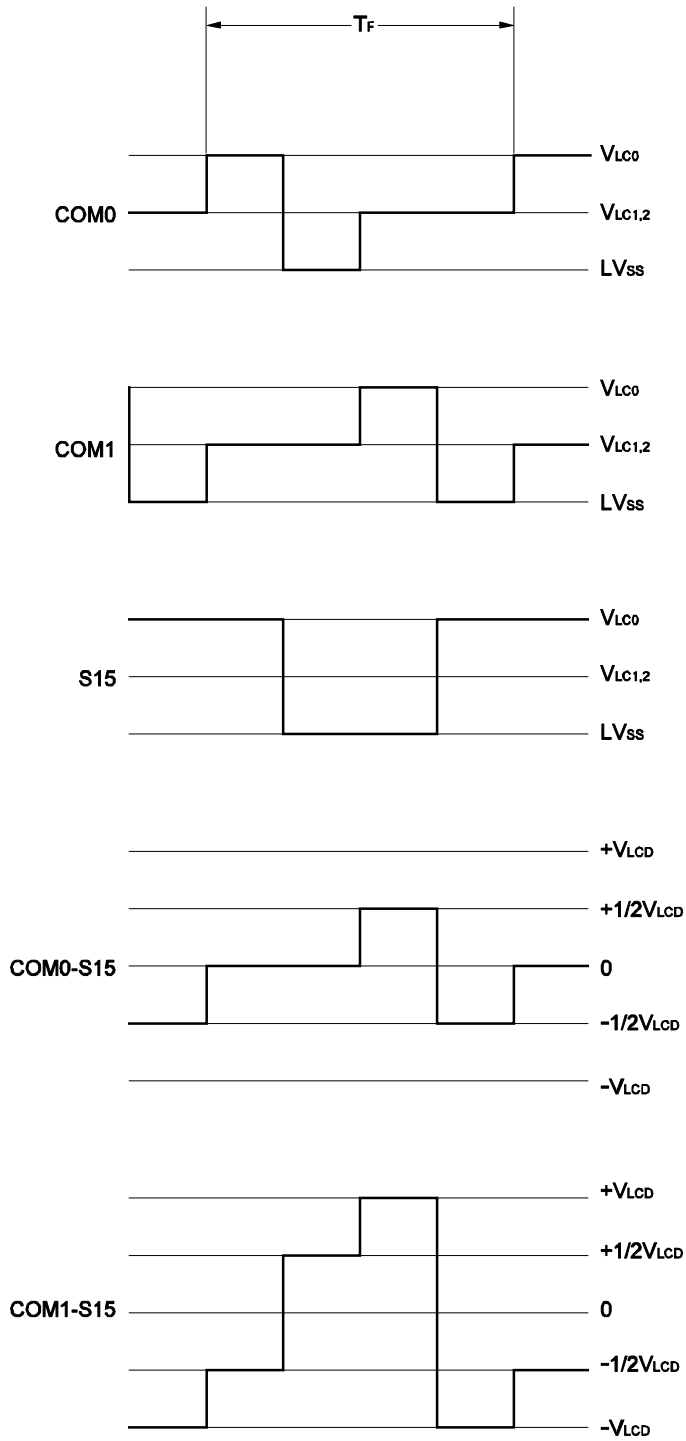
备注 n = 0 ~ 5

图 17-19. 2 分时 LCD 面板连接示例



x: 因为使用 2 分时, 可以始终被用于存储任意数据。

图 17-20. 2 分时 LCD 驱动波形示例 (1/2 偏压模式)



17.7.3 3 分时显示示例

图 17-22 显示了如何把 8 数字 LCD 面板显示形成图形，按图 17-21 连接 78K0/LF2 芯片 segment 信号(S0 ~ S23) 和 common 信号(COM0 ~ COM2)。这个例子在 LCD 面板上显示数字"123456.78"。显示数据存储器(LCDSEG 的 00H ~ 17H 地址)的内容符合这个显示。

以下描述集中在数字"6"()显示，显示在第三个字段。为了在 LCD 面板显示"6"，必须按照 common 信号 COM0 ~ COM2 的时序，按照表 17-7 所示从 S6 ~ S8 引脚输出选择信号或非选择信号电压；参见图 17-21 segment 信号和 LCD segment 的关系。

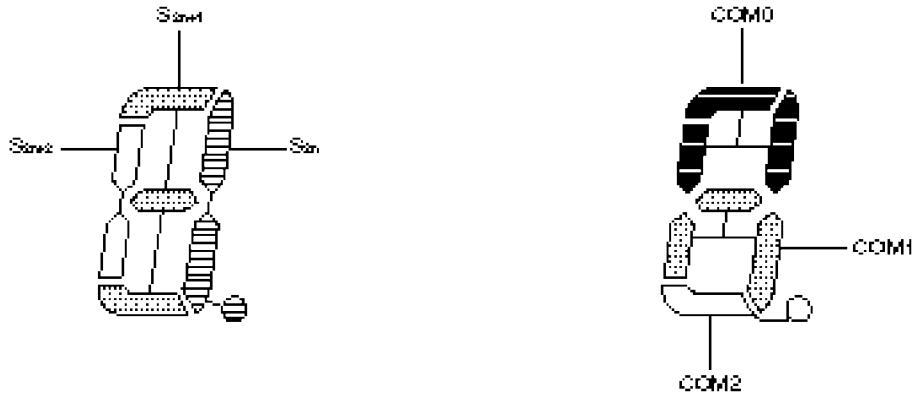
表 17-7. 选择和非选择电压 (COM0 ~ COM2)

Segment \ Common	S6	S7	S8
COM0	非选择	选择	选择
COM1	选择	选择	选择
COM2	选择	选择	-

依照表 17-7，它决定了 S6 相应的显示数据存储器区域(LCDSEG 的 06H)必须是 x110。

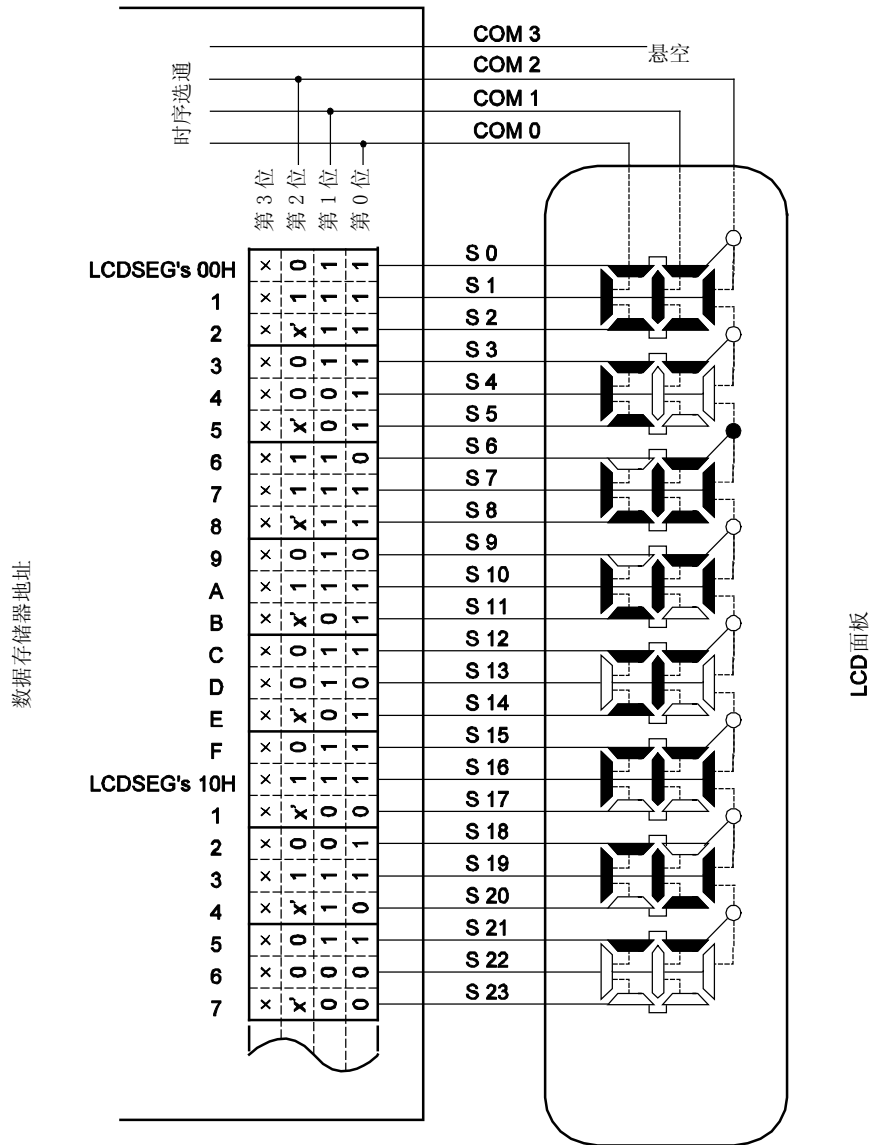
图 17-23 和 17-24 显示了 S6 信号和每个 common 信号在 1/2 和 1/3 偏压模式之间的 LCD 驱动波形。当选择电压以 COM1 或 COM2 的时序被应用在 S6 时，LCD segment 产生一个相应的交替矩形波形， $+V_{LCD}/-V_{LCD}$ 。

图 17-21. 3 分时 LCD 显示图形和电极连接



备注 n = 0 ~ 7

图 17-22. 3 分时 LCD 面板连接示例



x': 因为在 LCD 面板内没有相应的 segment, 可以被用于存储任意数据。
 x : 因为使用 3 分时模式, 可以始终被用于存储任意数据。

图 17-23. 3 分时 LCD 驱动波形示例 (1/2 偏压模式)

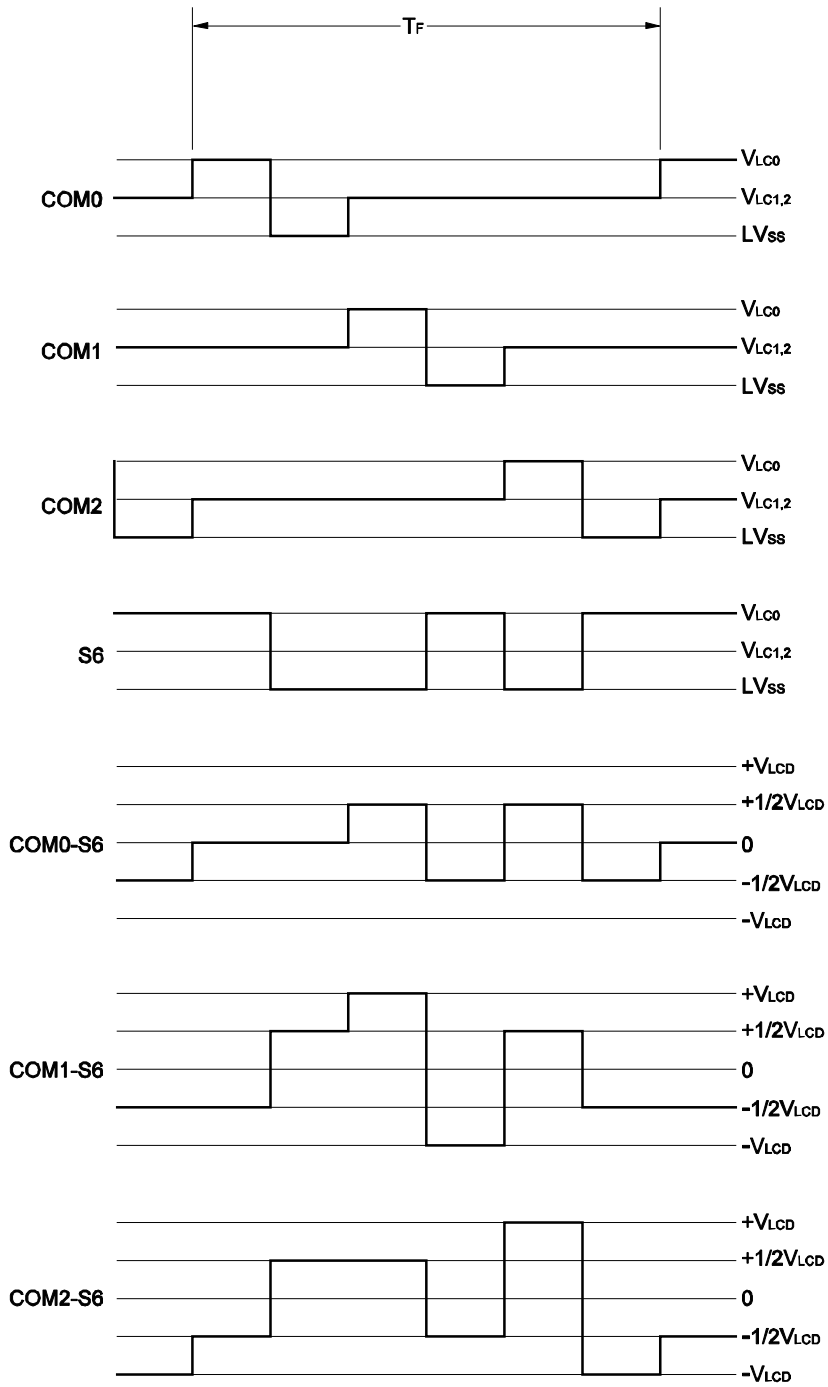
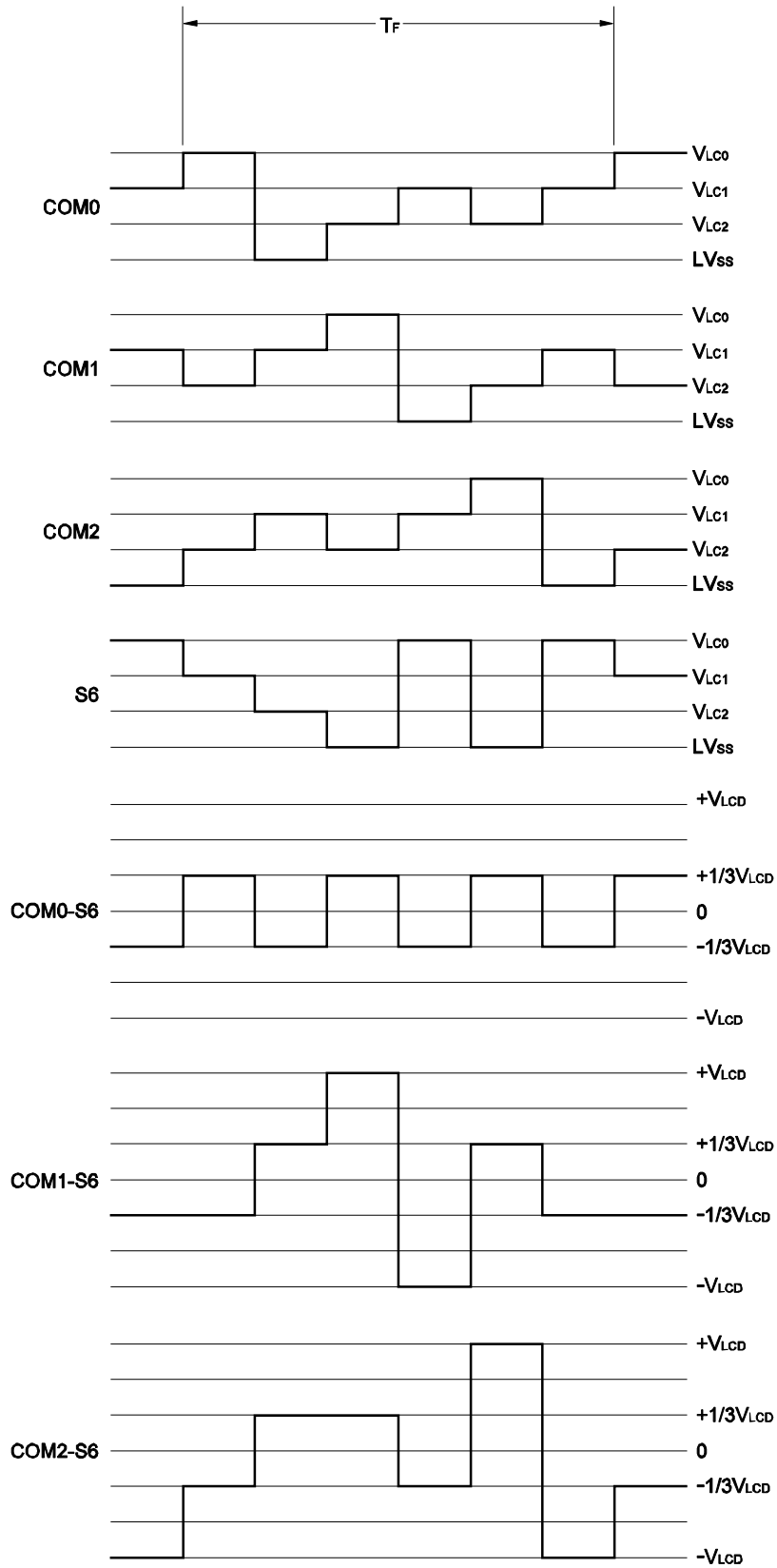


图 17-24. 3 分时 LCD 驱动波形示例 (1/3 偏压模式)



17.7.4 4 分时显示示例

图 17-26 显示了如何把 12 数字 LCD 面板显示形成图形，按图 17-25 连接 78K0/LF2 芯片 segment 信号(S0 ~ S23) 和 common 信号(COM0 ~ COM3)。这个例子在 LCD 面板上显示数字"123456.789012"。显示数据存储区(LCDSEG 的 00H ~ 17H 地址)的内容符合这个显示。

以下描述集中在数字"6"的显示，显示在第七个字段。为了在 LCD 面板显示"6"，必须按照 common 信号 COM0 ~ COM3 的时序，按照表 17-8 所示从 S12 和 S13 引脚输出选择信号或非选择信号电压；参见图 17-25 segment 信号和 LCD segment 的关系。

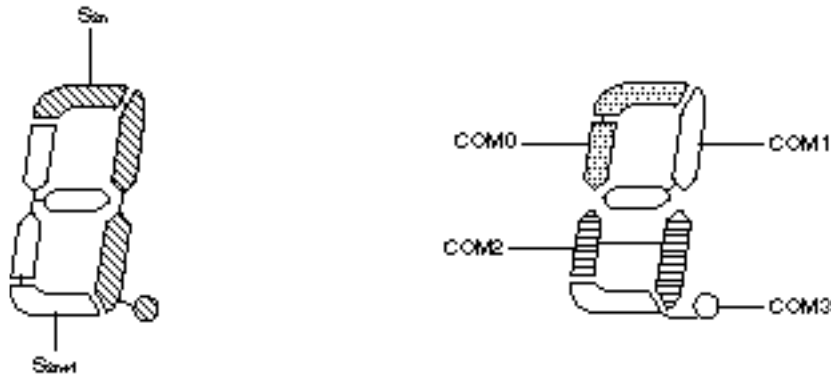
表 17-8. 选择和非选择电压 (COM0 ~ COM3)

Segment \ Common	S12	S13
COM0	选择	选择
COM1	非选择	选择
COM2	选择	选择
COM3	选择	选择

依照表 17-8，它决定了 S12 相应的显示数据存储区(LCDSEG 的 0CH)必须是 1101。

图 17-27 显示了 S12 信号和每个 common 信号的 LCD 驱动波形。当选择电压以 COM0 的时序被应用到 S12 时，LCD segment 产生一个相应的交替矩形波形， $+V_{Lcd}/-V_{Lcd}$ 。

图 17-25. 4 分时 LCD 显示图形和电极连接



备注 n = 0 ~ 11

图 17-26. 4 分时 LCD 面板连接示例

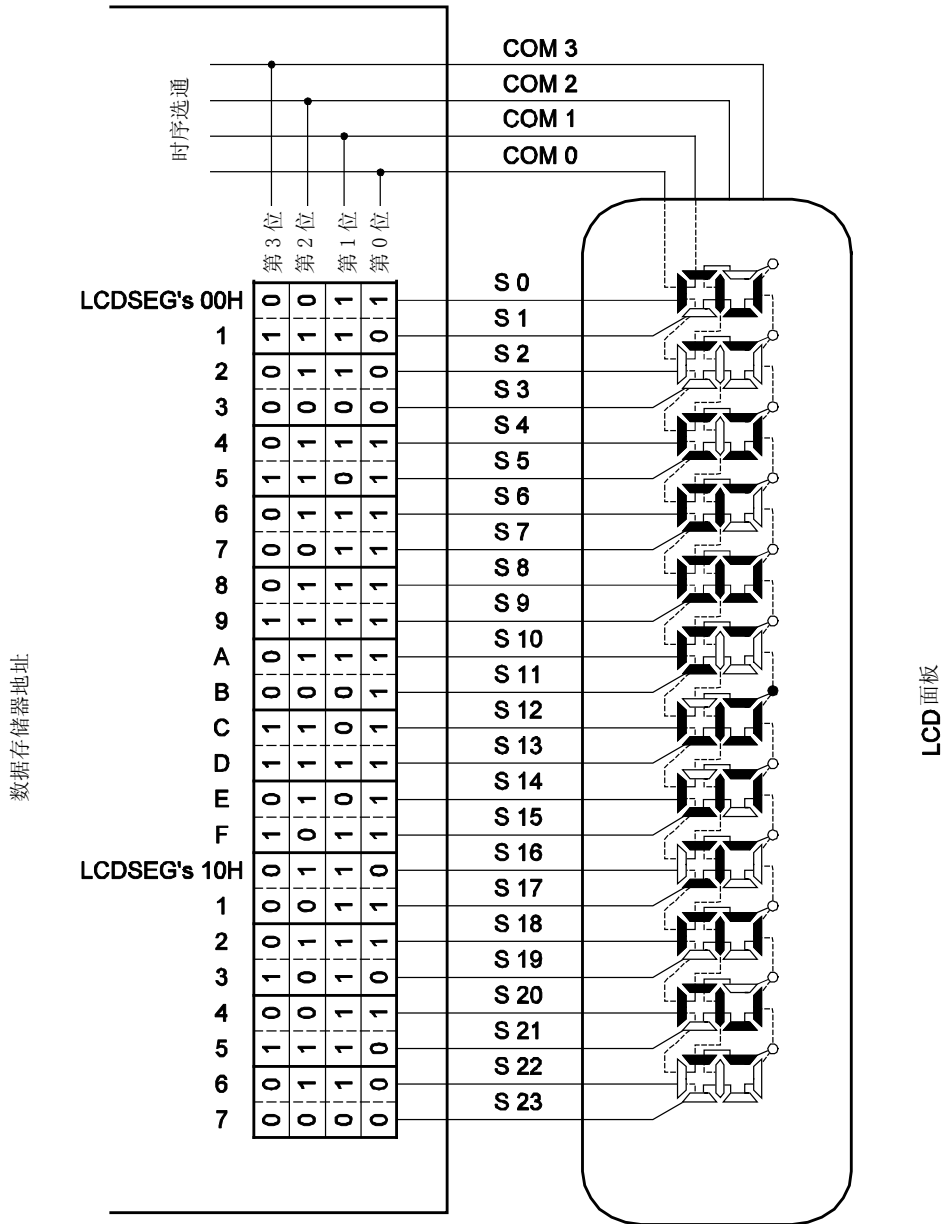
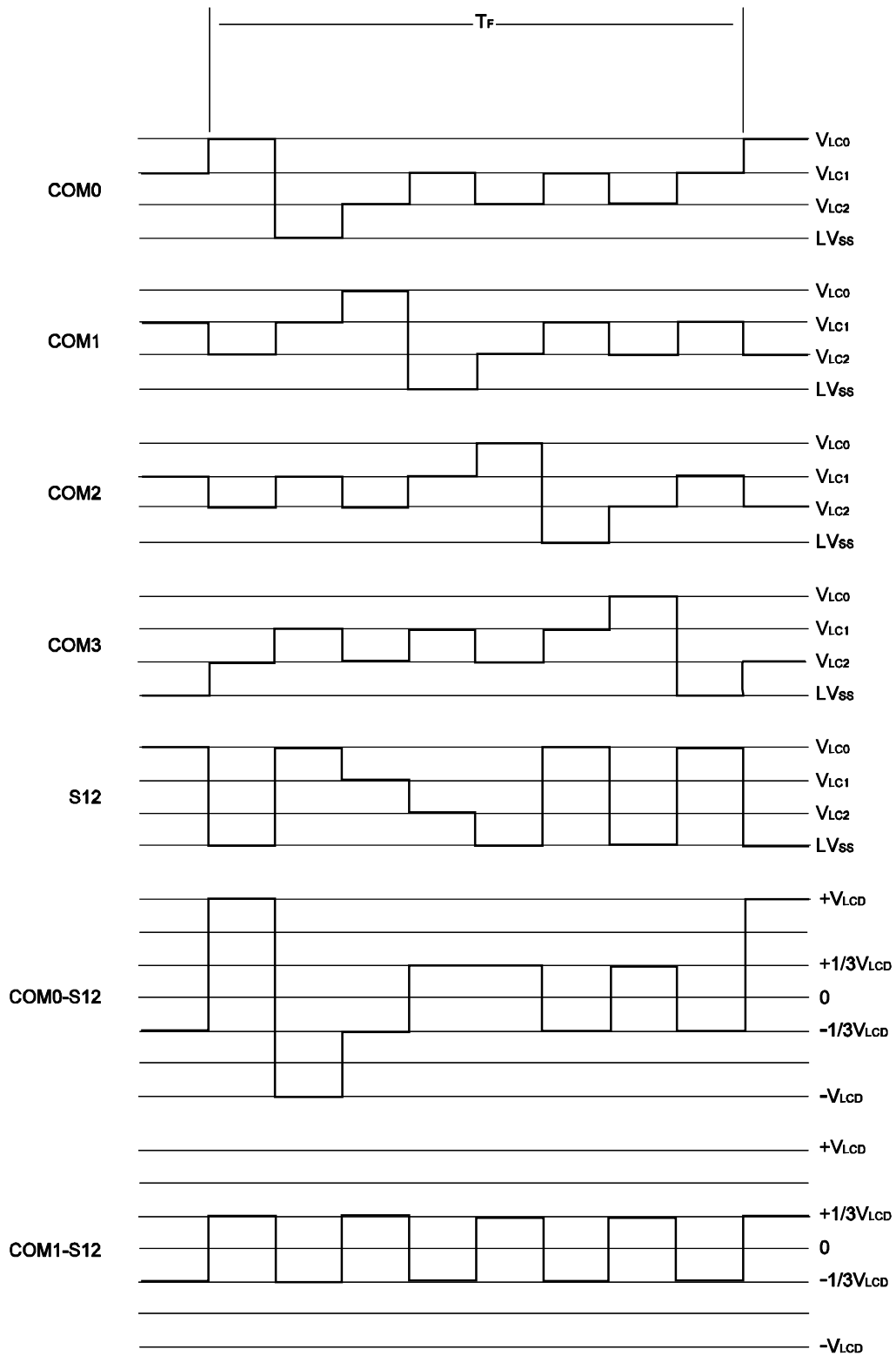


图 17-27. 4 分时 LCD 驱动波形示例 (1/3 偏压模式)



备注 忽略 COM2至 S12 和 COM3至 S12 波形。

17.8 LCD 驱动电压 V_{LC0} , V_{LC1} 和 V_{LC2}

78K0/LF2 中 LCD 驱动电源提供可由 3 种类型产生：内部电阻分压模式、外部电阻分压模式或内部电压泵模式。

17.8.1 内部电阻分压模式

78K0/LF2 内有分压电阻器提供产生 LCD 驱动电源。使用内部分压电阻器，LCD 驱动电源提供表 17-9 列出的偏压模式，不必使用外部分压电阻器。

表 17-9. LCD 驱动电压 (含片内电压驱动器电阻)

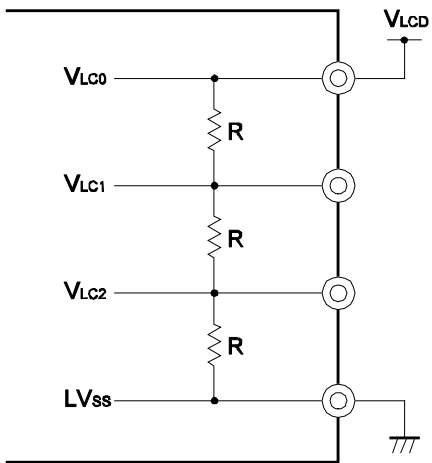
LCD 驱动电压引脚 \ 偏压模式	无偏压 (静态)	1/2 偏压模式	1/3 偏压模式
V_{LC0}	V_{LCD}	V_{LCD}	V_{LCD}
V_{LC1}	$\frac{2}{3} V_{LCD}$	$\frac{1}{2} V_{LCD}^{\#}$	$\frac{2}{3} V_{LCD}$
V_{LC2}	$\frac{1}{3} V_{LCD}$		$\frac{1}{3} V_{LCD}$

注 1/2 偏压模式，必须连接 V_{LC1} 和 V_{LC2} 引脚。

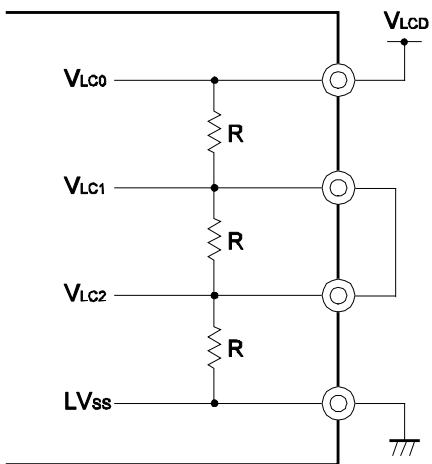
图 17-28 显示按表 17-9 产生 LCD 驱动电压的示例。

图 17-28 LCD 驱动电源连接示例 (内部电阻分压模式)

(a) 1/3 偏压模式和静态显示模式



(b) 1/2 偏压模式

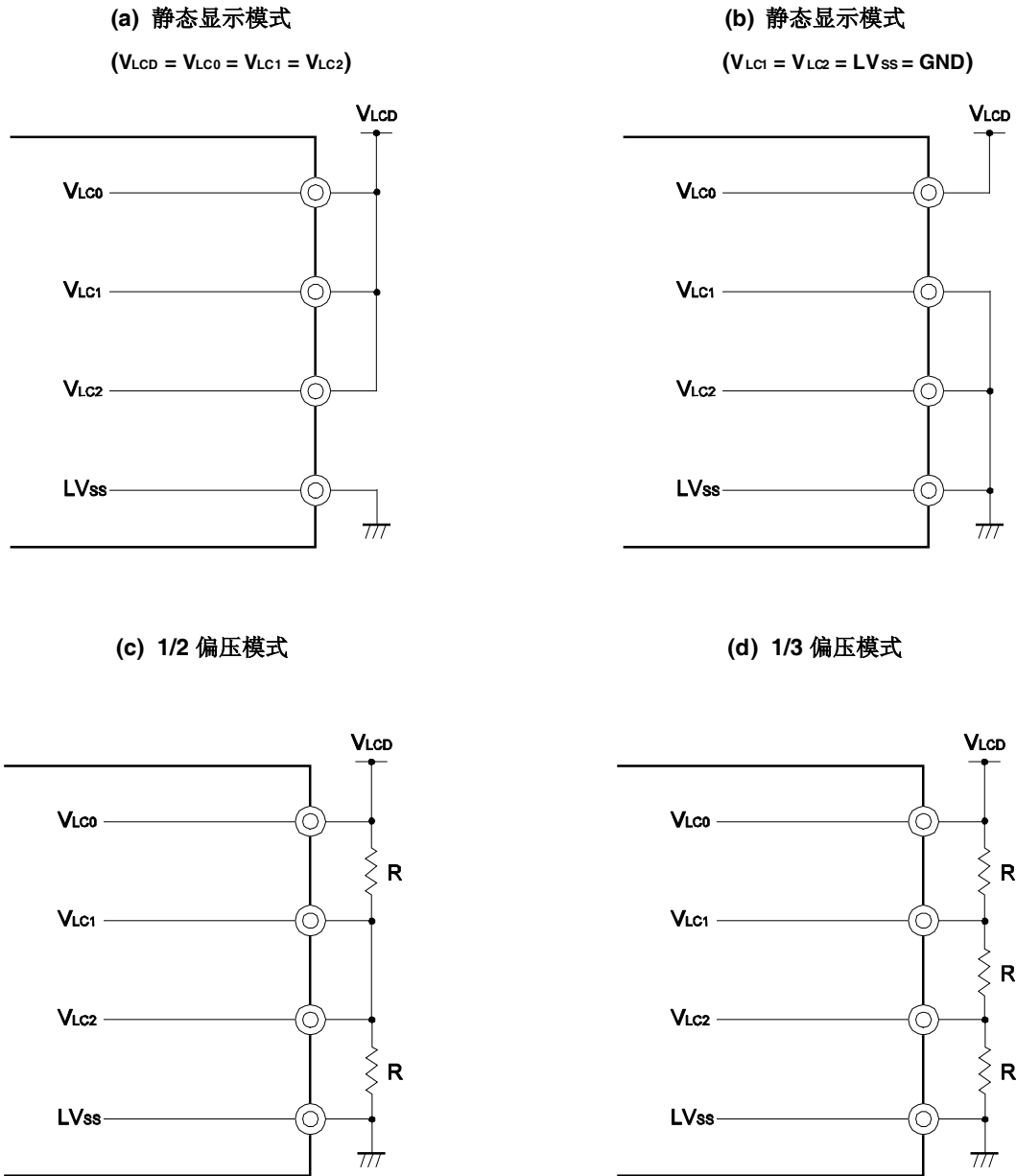


17.8.2 外部电阻分压模式

78K0/LF2 也可使用外边分压电阻器产生 LCD 驱动电源，不使用内部电阻器。图显示了相应各种偏压模式下 LCD 驱动电压连接示例。

<R>

图 17-29. LCD 驱动电压连接示例 (外部电阻分压模式)



备注 (a) 和 (b) 都可用于静态显示模式连接。

17.8.3 内部电压泵模式

78K0/LF2 包含电压泵电路(仅×3)可提供 LCD 驱动电压。内部 LCD 参考电压从 V_{LC2} 引脚输出。 V_{LC1} 引脚输出高于 V_{LC2} 2 倍的电压, V_{LC0} 引脚输出高于 V_{LC2} 3 倍的电压。

LCD 参考电压 (V_{LC2}) 通过 LCD 升压控制寄存器 0 (VLCG0)指定。

<R>

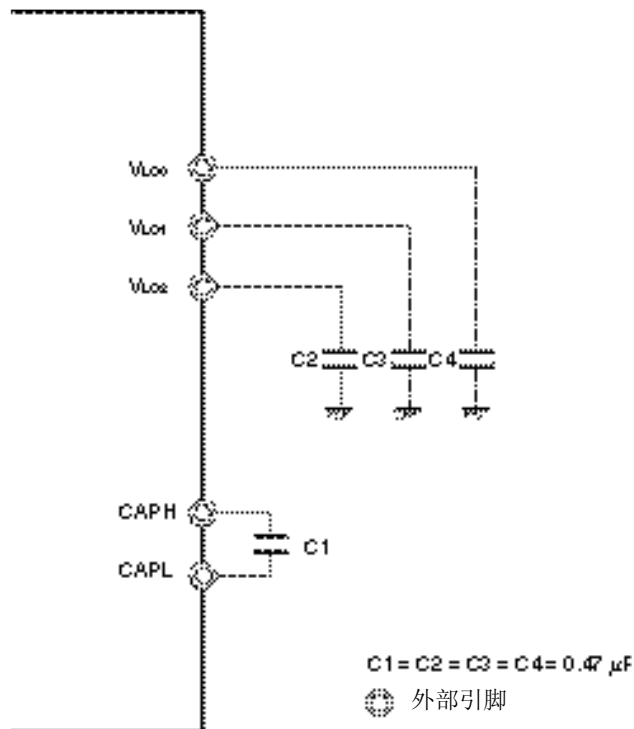
78K0/LF2 选择内部电压泵模式时, 需要一个外部电容 (0.47 ~ 1 μ F: 推荐)。

表 17-10. $V_{LC0} \sim V_{LC2}$ 引脚输出电压

LCD 驱动电源引脚	VLCG0	GAIN = 0	GAIN = 1
V_{LC0}		4.5 V	3.0 V
V_{LC1}		3.0 V	2.0 V
V_{LC2} (LCD 参考电压)		1.5 V	1.0 V

- 注意事项
1. 使用 LCD 功能时, 不能让 V_{LC0} , V_{LC1} 和 V_{LC2} 引脚悬空。参考图 17-30 连接。
 2. 当 LCD 驱动电压从主电源供应中分离, 不管 V_{DD} 和 LV_{DD} 是否波动都将提供恒定的电压。

图 17-30. LCD 驱动器引脚连接示例



备注 使用电容会产生微小的漏电流。
另外, C1 使用单极电容。

第十八章 乘法器/除法器

(仅限于 μ PD78F0374,78F0375,78F0376,78F0376D,78F0384,78F0385,78F0386 和 78F0386D)

仅限于 μ PD78F0374,78F0375,78F0376,78F0376D,78F0384,78F0385,78F0386 和 78F0386D 提供乘法器/除法器。

18.1 乘法器/除法器的功能

乘法器/除法器有以下功能。

- 16 位 \times 16 位 = 32 位(乘法)
- 32 位 \div 16 位 = 32 位, 16 位余数(除法)

18.2 乘法器/除法器的配置

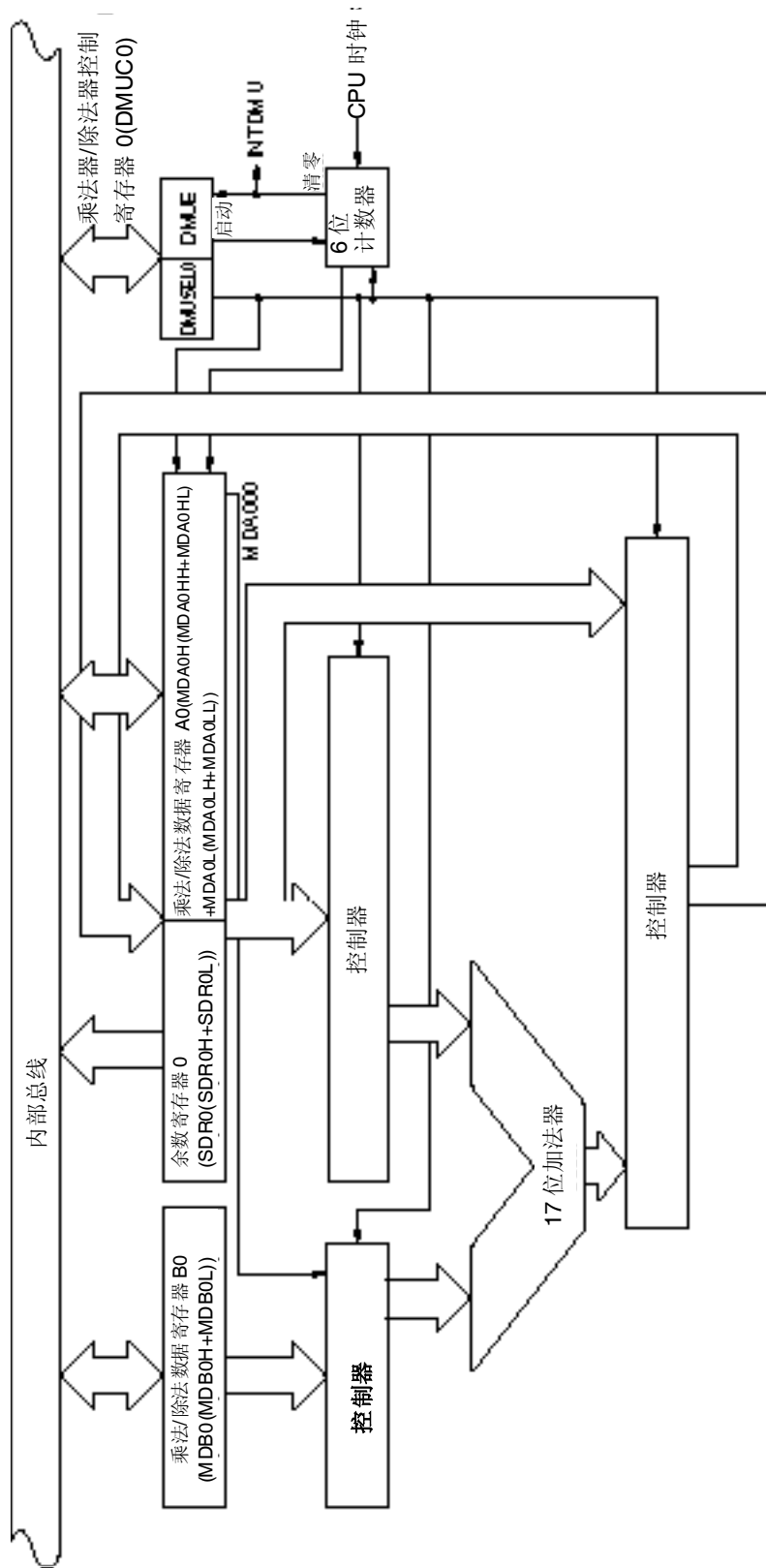
乘法器/除法器包括以下硬件。

表 18-1. 乘法器/除法器的配置

项目	配置
寄存器	余数寄存器 0 (SDR0) 乘法/除法数据寄存器 A0 (MDA0H、MDA0L) 乘法/除法数据寄存器 B0 (MDB0)
控制寄存器	乘法器/除法器控制寄存器 0 (DMUC0)

图 18-1 为乘法器/除法器的框图。

图 18-1. 乘法器/除法器的框图



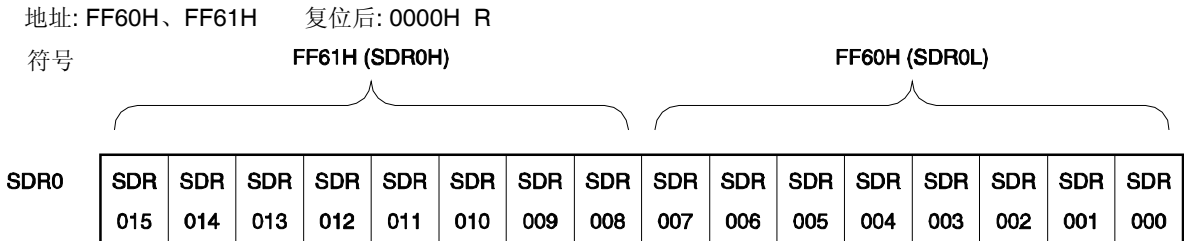
(1) 余数寄存器 0 (SDR0)

SDR0 是一个 16 位寄存器，用于存放余数。在乘法模式中该寄存器存储 0，而在除法模式中该寄存器存放操作结果的余数。

可由 8 位或 16 位存储器操作指令读取 SDR0。

复位信号产生将 SDR0 清零(0000H)。

图 18-2 余数寄存器 0 (SDR0) 的格式

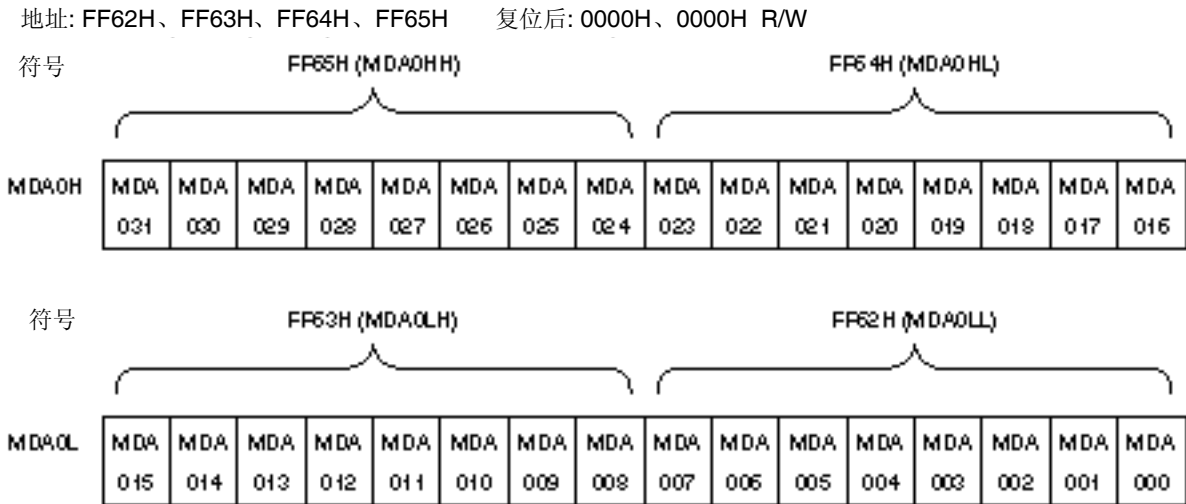


- 注意事项**
1. 在操作期间(乘法器/除法器控制寄存器 0(DMUC0)的第 7 位(DMUE)= 1)，从 SDR0 中读取的值不确定。
 2. 当启动操作时(当 DMUE=1 时)，SDR0 被复位。

(2) 乘法/除法数据寄存器 A0 (MDA0H、MDA0L)

MDA0 是一个 32 位寄存器，用于在乘法模式中设置 16 位乘数 A 和在除法模式中设置一个 32 位被除数，并存储 32 位的操作结果 (高 16 位: MDA0H、低 16 位: MDA0L)。

图 18-3 乘法/除法数据寄存器 A0 (MDA0H、MDA0L)的格式



- 注意事项**
1. 在乘法模式中(乘法器/除法器控制寄存器 0 (DMUC0) = 81H)启动操作时，MDA0H 被清零。
 2. 在操作期间(乘法器/除法器控制寄存器 0(DMUC0)的第 7 位(DMUE)= 1)，不要修改 MDA0 的值。此时即使执行修改操作，但结果不确定。
 3. 在操作期间(当 DMUE = 1)，从 MDA0 读取的值不确定。

18.3 控制乘法器/除法器的寄存器

乘法器/除法器由乘法器/除法器控制寄存器 0(DMUC0)控制。

(1) 乘法器/除法器控制寄存器 0 (DMUC0)

DMUC0 是一个 8 位寄存器，用于控制乘法器/除法器的操作。

可用 1 位或 8 位存储器操作指令设置 DMUC0。

复位信号产生将 DMUC0 清零(00H)。

图 18-5 乘法器/除法器控制寄存器 0 (DMUC0) 的格式

地址: FF68H 复位后: 00H R/W

符号	<7>	6	5	4	3	2	1	0
DMUC0	DMUE	0	0	0	0	0	0	DMUSEL0
	DMUE [*]	启动/停止操作						
	0	停止操作						
	1	启动操作						
	DMUSEL0	操作模式(乘法/除法)的选择						
	0	除法模式						
	1	乘法模式						

注 当 DMUE =1 时，启动操作。操作结束后 DMUE 被自动清零。

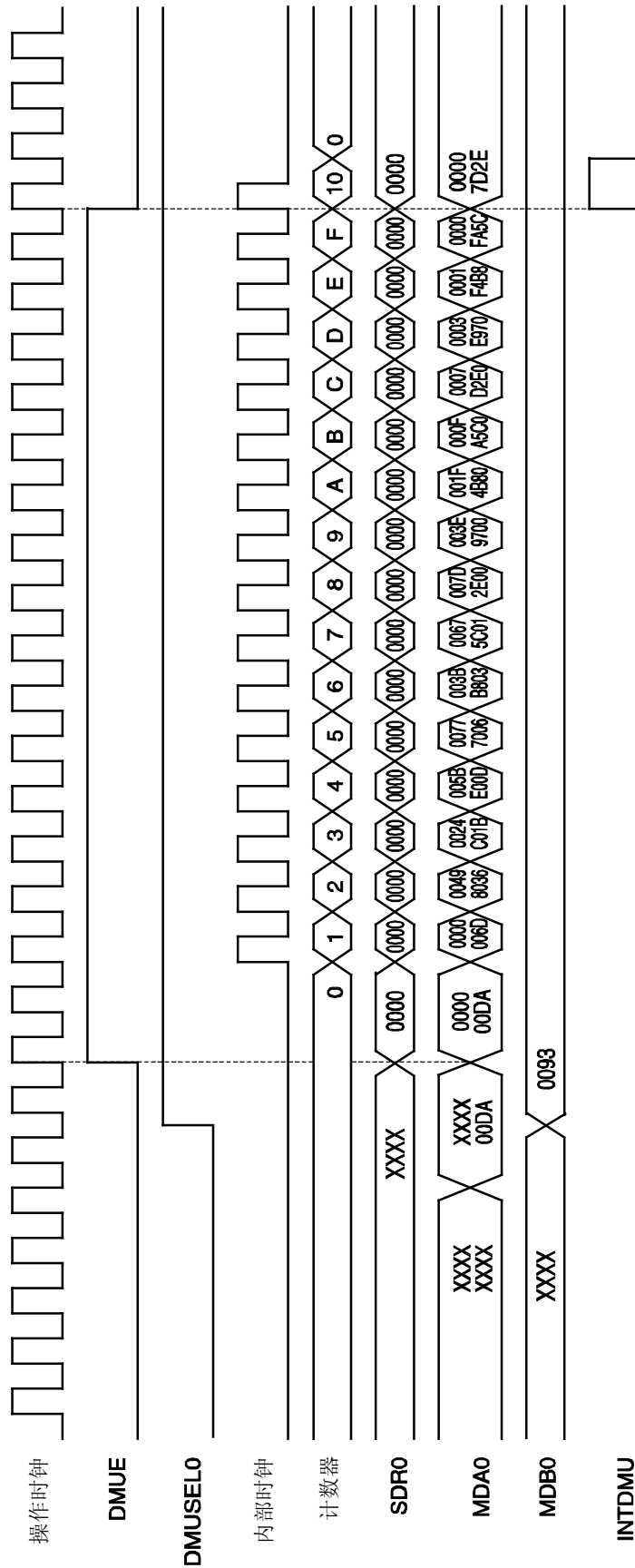
- 注意事项 1. 如果操作期间(当 DMUE = 1)DMUE 被清零，则操作结果不确定。如果设置了中断标志，在执行清零指令时操作完成，操作结果可以得到保证。
2. 在操作期间(当 DMUE = 1)时，不要修改 DMUSEL0 的值。如果修改了，则存放在乘法/除法数据寄存器 A0 (MDA0)和余数寄存器 0 (SDR0)中的操作结果不确定。
3. 如果在操作期间(当 DMUE = 1)将 DMUE 清零，则停止操作。若要再次执行操作，应对乘法/除法数据寄存器 A0 (MDA0)、乘法/除法数据寄存器 B0 (MDB0)和乘法器/除法器控制寄存器 0 (DMUC0)进行设置，然后启动操作(通过设置 DMUE = 1)。

18.4 乘法器/除法器的操作

18.4.1 乘法操作

- 初始化设置
 1. 将操作数赋给乘法/除法数据寄存器 A0L (MDA0L)和乘法/除法数据寄存器 B0 (MDB0)。
 2. 设置乘法器/除法器控制寄存器 0 (DMUC0)的第 0 位和第 7 位(DMUSEL0 和 DMUE)=1。启动操作。
- 操作期间
 3. 启动操作后经历 16 个内部时钟时, 操作完成(操作期间中间数据存于 MDA0L 和 MDA0H 中, 因此读取这些寄存器得到的值不确定)。
- 操作结束
 4. 操作结果存于 MDA0L 和 MDA0H 中。
 5. DMUE 被清零 (操作结束)。
 6. 操作结束后, 产生中断请求信号(INTDMU)。
- 下一次操作
 7. 执行下一次乘法操作, 要从 **18.4.1 乘法操作**中的初始化设置开始。
 8. 执行下一次除法操作, 要从 **18.4.2 除法操作**中的初始化设置开始。

图 18-6. 乘法操作(00DAH × 0093H)的时序



18.4.2 除法操作

- 初始化设置
 1. 将操作数赋给乘法/除法数据寄存器 A0(MDA0L 和 MDA0H)和乘法/除法数据寄存器 B0(MDB0)。
 2. 分别设置乘法器/除法器控制寄存器 0(DMUC0)的第 0(DMUSEL0)和第 7 位(DMUE)为 0 和 1。启动操作。
- 操作期间
 3. 启动操作后经历 32 个内部时钟时, 操作完成(操作期间中间数据存于 MDA0L 和 MDA0H 以及余数寄存器 0(SDR0)中, 因此读取这些寄存器得到的值不确定)。
- 操作结束
 4. 操作结果存于 MDA0L、MDA0H 和 SDR0 中。
 5. DMUE 被清零(操作结束)。
 6. 操作结束后, 产生中断请求信号(INTDMU)。
- 下一次操作
 7. 执行下一次乘法操作, 要从 **18.4.1 乘法操作**中的初始化设置开始。
 8. 执行下一次除法操作, 要从 **18.4.2 除法操作**中的初始化设置开始。

第十九章 中断功能

19.1 中断功能的类型

可分为以下两种类型。

(1) 可屏蔽中断

这类中断可进行屏蔽控制。通过设置优先级指定标志寄存器(PR0L, PR0H, PR1L, PR1H)将可屏蔽中断分为高优先级中断组和低优先级中断组。

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,请参见表 19-1 产生一个退出待机信号,并退出 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。

- μ PD78F0372, 78F0373
外部: 7, 内部: 16
- μ PD78F0374, 78F0375, 78F0376, 78F0376D
外部: 7, 内部: 18
- μ PD78F0382, 78F0383
外部: 7, 内部: 15
- μ PD78F0384, 78F0385, 78F0386, 78F0386D
外部: 7, 内部: 17

(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。

19.2 中断源及配置

μ PD78F0372 和 78F0373, 共有 24 种中断源。 μ PD78F0374, 78F0375, 78F0376 和 78F0376D, 共有 26 种中断源。 μ PD78F0382 和 78F0383, 共有 23 种中断源。 μ PD78F0384, 78F0385, 78F0386 和 78F0386D, 共有 25 种中断源。都包括可屏蔽的中断和软件中断。另外,还具有 4 种复位源(参见表 19-1)。

表 19-1. 中断源列表(1/2)

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	0	INTLVI	低压检测 ^{注3}	内部	0004H	(A)
	1	INTP0	引脚输入脉冲沿检测	外部	0006H	(B)
	2	INTP1			0008H	
	3	INTP2			000AH	
	4	INTP3			000CH	
	5	INTP4			000EH	
	6	INTP5			0010H	
	7	INTSRE6	UART6 产生接收错误	内部	0012H	(A)
	8	INTSR6	UART6 接收结束		0014H	
	9	INTST6	UART6 发送结束		0016H	
	10	INTCSI10/ INTST0	CSI10 通信结束/ UART0 通信结束		0018H	
	11	INTTMH1	TMH1 与 CMP01 相等 (指定比较寄存器)		001AH	
	12	INTTMH0	TMH0 与 CMP00 相等 (指定比较寄存器)		001CH	
	13	INTTM50	TM50 与 CR50 相等 (指定比较寄存器)		001EH	
	14	INTTM000	TM00 与 CR000 相等 (指定比较寄存器), TI010 引脚有效沿检测 (指定捕捉寄存器)		0020H	
	15	INTTM010	TM00 与 CR010 相等 (指定比较寄存器), TI000 引脚有效沿检测 (指定捕捉寄存器)		0022H	
	16	INTAD ^{注4}	A/D 转换结束		0024H	
	17	INTSR0	UART0 接收结束或产生接收错误		0026H	
	18	INTWTI	钟表定时器参考时间间隔信号		0028H	
	19	INTTM51	TM51 与 CR51 相等 (指定比较寄存器)		002AH	
	20	INTKR	按键中断检测		外部	
21	INTWT	钟表定时器溢出	内部	002EH	(A)	

- 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 24 的优先级别最低。
 2. 基本配置类型 (A) ~ (D) 与图 19-1 中 (A) ~ (D) 相对应。
 3. 当低电压检测寄存器(LVIM)的第 1 位(LVIMD)被清零时。
 4. 仅限于 μ PD78F037x。

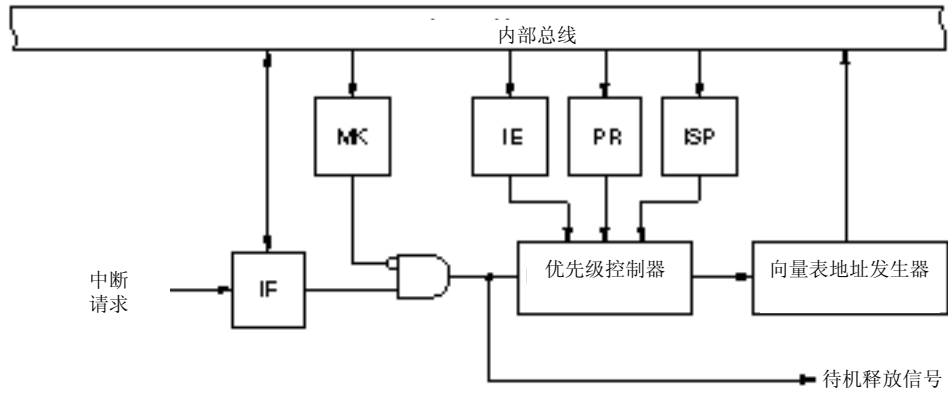
表 19-1. 中断源列表(2/2)

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	22	INTIIC0/ INTDMU ^{注3}	IIC0 通信结束/乘法/除法操作结束	内部	0034H	(A)
	23	INTTM001 ^{注3}	TM01 与 CR001 相等		0038H	
	24	INTTM011 ^{注3}	TM01 与 CR011 相等		003AH	
软件	-	BRK	执行 BRK 指令	-	003EH	(D)
复位	-	RESET	复位输入	-	0000H	-
		POC	上电清零			
		LVI	低压检测 ^{注4}			
		WDT	WDT 溢出			

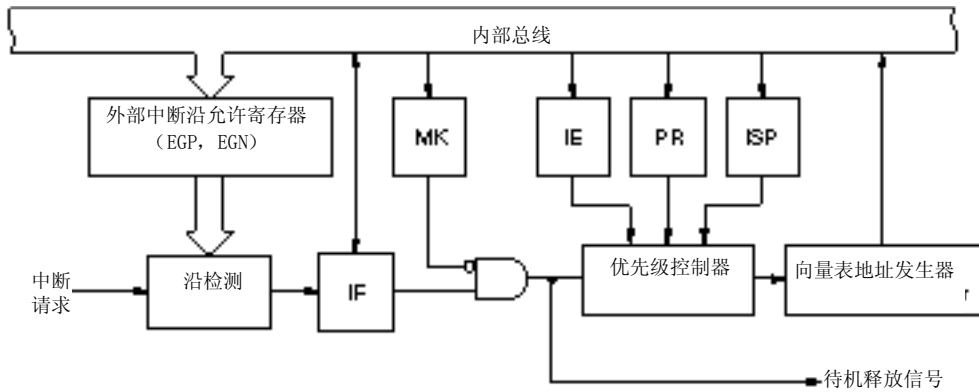
- 注
1. 当两个或两个以上可屏蔽中断同时产生时，采用默认的优先级来决定向量中断的处理顺序。优先级 0 的优先级别最高，而优先级 24 的优先级别最低。
 2. 基本配置类型 (A) ~ (D) 与图 19-1 中 (A) ~ (D) 相对应。
 3. 仅在 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D 中包括中断源 INTDMU, INTTM001 和 INTTM011。
 4. 当低电压检测寄存器 (LVIM) 的第 1 位 (LVIMD) 设置为 1 时。

图 19-1. 中断功能的基本配置(1/2)

(A) 内部可屏蔽中断



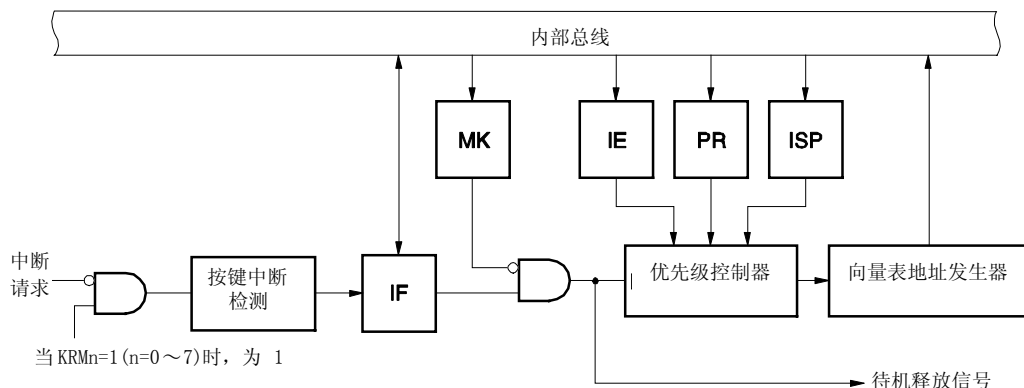
(B) 外部可屏蔽中断 (INTP0 ~ INTP5)



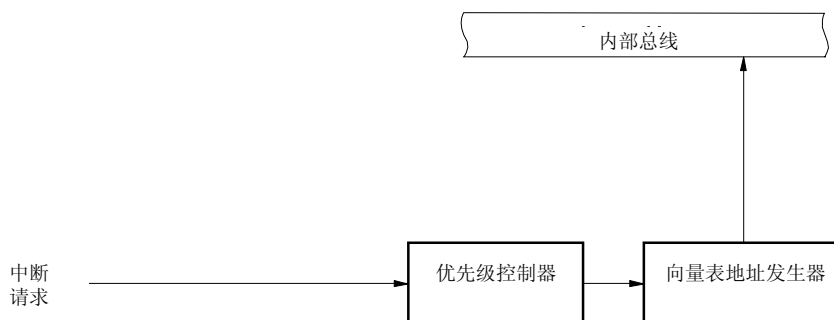
- IF: 中断请求标志
- IE: 中断允许标志
- ISP: 服务优先级标志
- MK: 中断屏蔽标志
- PR: 优先级指定标志

图 19-1. 中断功能的基本配置(2/2)

(C) 外部可屏蔽中断 (INTKR)



(D) 软件中断



- IF: 中断请求标志
- IE: 中断允许标志
- ISP: 服务优先级标志
- MK: 中断屏蔽标志
- PR: 优先级指定标志
- KRM: 按键返回模式寄存器

19.3 控制中断功能的寄存器

以下 6 种寄存器用于控制中断功能。

- 中断请求标志寄存器(IF0L、IF0H、IF1L、IF1H)
- 中断屏蔽标志寄存器(MK0L、MK0H、MK1L、MK1H)
- 优先级指定标志寄存器(PR0L、PR0H、PR1L、PR1H)
- 外部中断上升沿允许寄存器(EGP)
- 外部中断下降沿允许寄存器(EGN)
- 程序状态字(PSW)

表 19-2 显示了对应于中断请求源的中断请求标志、中断屏蔽标志和优先级指定标志列表。

表 19-2. 相应中断请求源的标志

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			
INTLVI	LVIF	IF0L	LVIMK	MK0L	LVIPR	PROL			
INTP0	PIF0		PMK0		PPR0				
INTP1	PIF1		PMK1		PPR1				
INTP2	PIF2		PMK2		PPR2				
INTP3	PIF3		PMK3		PPR3				
INTP4	PIF4		PMK4		PPR4				
INTP5	PIF5		PMK5		PPR5				
INTSRE6	SREIF6		SREMK6		SREPR6				
INTSR6	SRIF6	IF0H	SRMK6	MK0H	SRPR6	PROH			
INTST6	STIF6		STMK6		STPR6				
INTCSI10	CSIIF10		DUALIF0 注 1		CSIMK10		DUALMK0 注 2	CSIPR10	DUALPR0 注 2
INTST0	STIF0				STMK0			STPR0	
INTTMH1	TMIFH1		TMMKH1		TMPRH1				
INTTMH0	TMIFH0		TMMKH0		TMPRH0				
INTTM50	TMIF50		TMMK50		TMPR50				
INTTM000	TMIF000		TMMK000		TMPR000				
INTTM010	TMIF010		TMMK010		TMPR010				
INTAD 注 3	ADIF 注 3		IF1L		ADMK 注 3		MK1L	ADPR 注 3	PR1L
INTSR0	SRIF0	SRMK0		SRPR0					
INTWTI	WTIF	WTIMK		WTIPR					
INTTM51	TMIF51	TMMK51		TMPR51					
INTKR	KRIF	KRMK		KRPR					
INTWT	WTIF	WTMK		WTPR					
INTIIC0	IICIF0	IF1H		IICMK0	MK1H	IICPR0		PR1H	
INTDMU 注 4	DMUIF 注 4		DMUMK 注 4	DMUPR 注 4					
INTTM001 注 4	TMIF001 注 4		TMMK001 注 4	TMPR001 注 4					
INTTM011 注 4	TMIF011 注 4		TMMK011 注 4	TMPR011 注 4					

- 注
1. 如果产生 INTCSI10 或 INTST0 中的任意一种，则这些位设置为(1)。
 2. 支持中断源 INTCSI10 和 INTST0。
 3. 仅限于 μ PD78F037x。
 4. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D。

(1) 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H)

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 和 IF1H。当 IF0L 与 IF0H、IF1L 与 IF1H 组合起来形成 16 位寄存器 IF0 和 IF1 时，可由 16 位存储器操作指令设置这些寄存器。

复位信号的输入将这些寄存器清零(00H)。

图 19-2. 中断请求标志寄存器 (IF0L, IF0H, IF1L, IF1H) 的格式

地址: FFE0H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	SREIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIF

地址: FFE1H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF010	TMIF000	TMIF50	TMIFH0	TMIFH1	DUALIF0 CSIIF10 STIF0	STIF6	SRIF6

地址: FFE2H 复位后: 00H R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	0	0	WTIF	KRIF	TMIF51	WTIIF	SRIF0	ADIF ^{‡1}

地址: FFE3H 复位后: 00H R/W

符号	7	6	5	4	<3>	<2>	1	<0>
IF1H	0	0	0	0	TMIF011 ^{‡2}	TMIF001 ^{‡2}	0	IICIF0 DMUIF ^{‡2}

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 1. 仅限于 μ PD78F037x。
2. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D。

- 注意事项 1. 对于 μ PD78F037x，必须对 IF1L 的第 6 位和第 7 位清零。对于 μ PD78F038x，必须对 IF1L 的第 0 位、第 6 位和第 7 位清零。
2. 对于 μ PD78F0372, 78F0373, 78F0382 和 78F0383，必须对 IF1H 的第 1 位 ~ 第 7 位清零。对于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D，必须对 IF1H 的第 1 位和第 4 位 ~ 第 7 位清零。
3. 当退出待机模式后要使用定时器、串行接口或 A/D 转换器时，在将中断请求标志清零后对这些部件操作一次。噪音可能会设置中断请求标志。

注意事项 4. 当使用中断请求标志寄存器中的标志时，可使用一位存储器操作指令 (CLR1)。当用 C 语言描述时，由于编译后的汇编指令必须是一位存储器操作指令 (CLR1)，所以应该使用一位操作指令，如“IF0L.0 = 0;”或“_asm(“clr1 IF0L, 0”);”。

如果用 C 语言描述一条 8 位存储器操作指令如“IF0L &= 0xfe;”，则编译后将被转换为三条汇编指令：

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

这种情况下，当处于“mov a, IF0L”和“mov IF0L, a”之间的时序时，即使同一中断请求标志寄存器 (IF0L) 的其他请求标志被设置为 1，则该请求标志也将被“mov IF0L, a”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 (MKOL, MKOH, MK1L, MK1H)

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 MKOL、MKOH、MK1L 和 MK1H。当 MKOL 与 MKOH、MK1L 与 MK1H 组合起来形成 16 位寄存器 MK0 与 MK1 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的内容设置为 FFH。

图 19-3. 中断屏蔽标志寄存器的格式 (MKOL, MKOH, MK1L, MK1H)

地址: FFE4H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MKOL	SREMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK

地址: FFE5H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MKOH	TMMK010	TMMK000	TMMK50	TMMKH0	TMMKH1	DUALMK0 CSIMK0 STMK0	STMK6	SRMK6

地址: FFE6H 复位后: FFH R/W

符号	7	6	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	1	1	WTMK	KRMK	TMMK51	WTIMK	SRMK0	ADMK ^{‡1}

地址: FFE7H 复位后: FFH R/W

符号	7	6	5	4	<3>	<2>	1	<0>
MK1H	1	1	1	1	TMMK011 ^{‡2}	TMMK001 ^{‡2}	1	IICMK0 DMUMK ^{‡2}

XXMKX	中断服务控制							
0	允许中断服务							
1	禁止中断服务							

- 注 1. 仅限于 μ PD78F037x。
 2. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D。

- 注意事项 1. 对于 μ PD78F037x，必须将 MK1L 的第 6 位和第 7 位置 1。对于 μ PD78F038x，必须将 MK1L 的第 0 位、第 6 位和第 7 位置 1。
 2. 对于 μ PD78F0372, 78F0373, 78F0382 和 78F0383，必须将 MK1H 的第 1 位 ~ 第 7 位置 1。对于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D，必须将 MK1H 的第 1 位和第 4 位 ~ 第 7 位置 1。

(3) 优先级指定标志寄存器(PR0L, PR0H, PR1L, PR1H)

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。

可由 1 位或 8 位存储器操作指令设置 PR0L、PR0H、PR1L 和 PR1H。当 PR0L 与 PR0H、PR1L 与 PR1H 组合起来形成 16 位寄存器 PR0 与 PR1 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内容设置为 FFH。

图 19-4. 优先级指定标志寄存器 (PR0L, PR0H, PR1L, PR1H) 的格式

地址: FFE8H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0L	SREPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	LVIPR

地址: FFE9H 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	TMPR010	TMPR000	TMPR50	TMPRH0	TMPRH1	DUALPR0 CSIPR10 STPR0	STPR6	SRPR6

地址: FFEAH 复位后: FFH R/W

符号	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR1L	1	1	WTPR	KRPR	TMPR51	WTIPR	SRPR0	ADPR ^{注1}

地址: FFE BH 复位后: FFH R/W

符号	7	6	5	4	<3>	<2>	1	<0>
PR1H	1	1	1	1	TMPR011 ^{注2}	TMPR001 ^{注2}	1	IICPR0 DMUPR ^{注2}

XXPRX	优先级选择
0	高优先级
1	低优先级

- 注 1. 仅限于 μ PD78F037x。
 2. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D。

- 注意事项 1. 对于 μ PD78F037x, 必须将 PR1L 的第 6 位和第 7 位置 1。对于 μ PD78F038x, 必须将 PR1L 的第 0 位、第 6 位和 7 位置 1。
 2. 对于 μ PD78F0372, 78F0373, 78F0382 和 78F0383, 必须将 PR1H 的第 1 位 ~ 第 7 位置 1。对于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D, 必须将 PR1H 的第 1 位和第 4 位 ~ 第 7 位置 1。

(4) 外部中断上升沿允许寄存器(EGP)、外部中断下降沿允许寄存器(EGN)

这两个寄存器用于指定 INTP0 ~ INTP5 的有效沿。

可由 1 位或 8 位存储器操作指令设置 EGP 和 EGN。

复位信号的产生将这些寄存器清零 (00H)。

图 19-5 外部中断上升沿允许寄存器(EGP)和外部中断下降沿允许寄存器(EGN)的格式

地址: FF48H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGP	0	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

地址: FF49H 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
EGN	0	0	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn 引脚有效沿的选择 (n = 0 ~ 5)
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 19-3 显示了与 EGPn 和 EGNn 相关的端口。

表 19-3. 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	P30	INTP1
EGP2	EGN2	P31	INTP2
EGP3	EGN3	P32	INTP3
EGP4	EGN4	P33	INTP4
EGP5	EGN5	P16	INTP5

注意事项 在从外部中断功能切换到端口功能时可能会检测到脉冲沿，因此通过将 EGPn 和 EGNn 清零可选择端口模式。

备注 n = 0 ~ 5

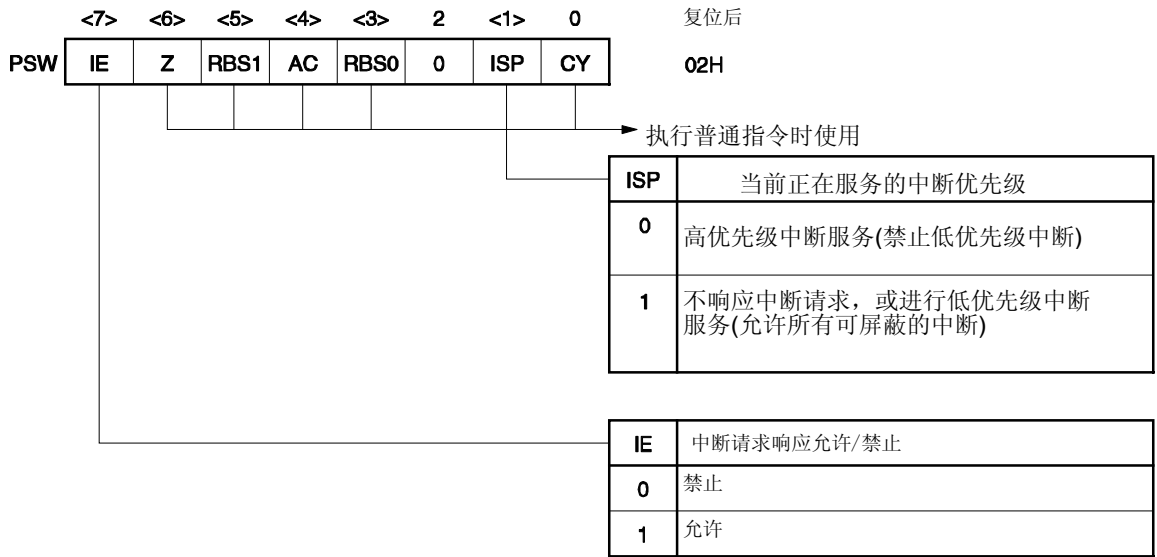
(5) 程序状态字(PSW)

PSW 用于保存指令执行结果和中断请求的当前状态。在 PSW 中包含 IE 标志（设置允许/禁止可屏蔽中断）和 ISP 标志（控制中断嵌套）。

除了 8 位读/写操作指令，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转移到 ISP 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。

RESET 输入将 PSW 设置为 02H。

图 19-6. 程序状态字的格式



19.4 中断服务操作

19.4.1 可屏蔽的中断响应

当中断请求标志被设置为 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 =1)，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（ISP 标志复位为 0），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 19-4 所示。

中断请求响应时序可参见图 19-8 和 19-9。

表 19-4. 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
When $\times\times PR = 0$	7 个时钟	32 个时钟
When $\times\times PR = 1$	8 个时钟	33 个时钟

注 如果是在除法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/fCPU (fCPU: 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。

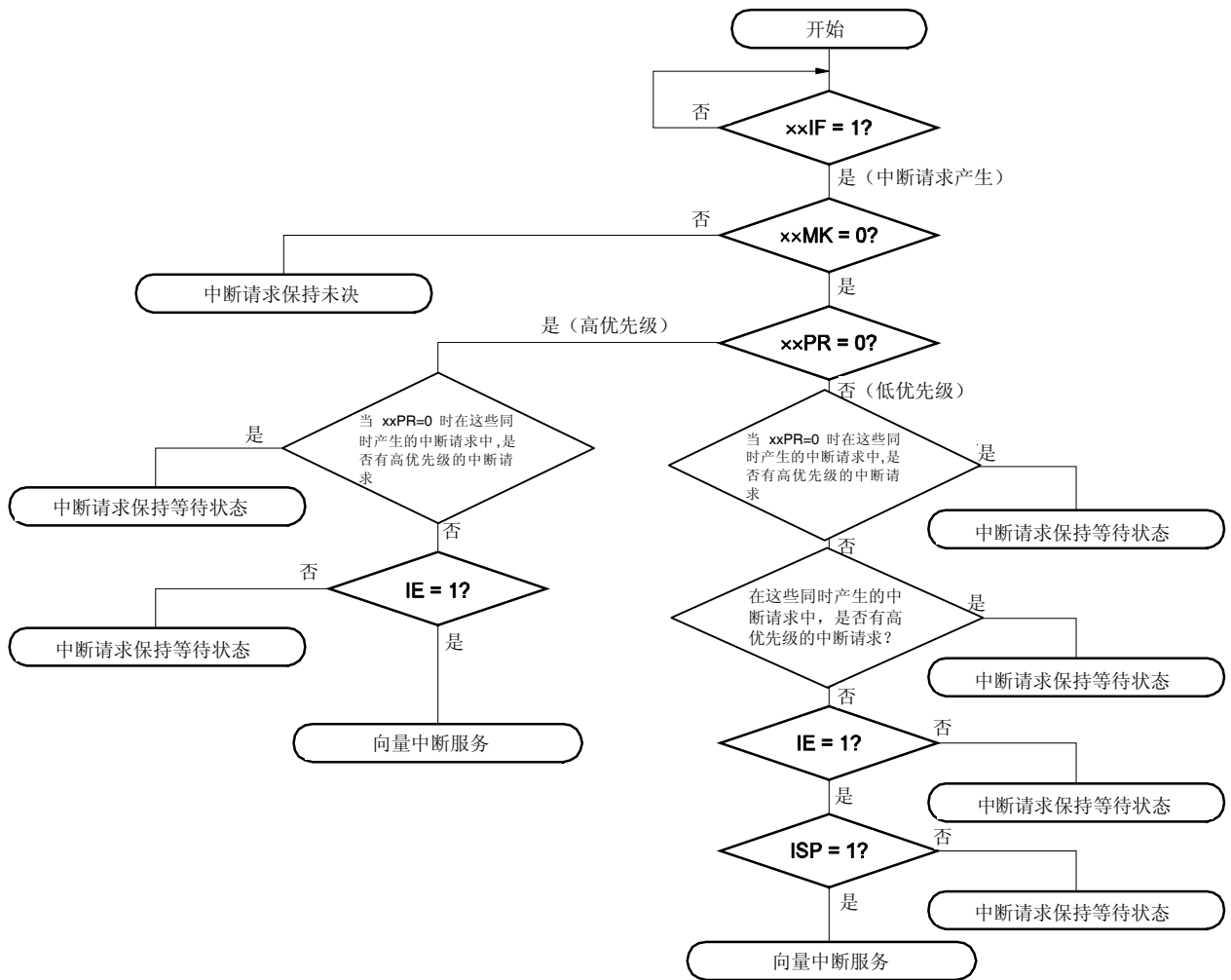
当允许响应中断请求时，处于等待状态的中断请求被响应。

图 19-7 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 ISP 中。将用于中断请求的向量表数据传送到 PC 中，并转移。

可通过执行 RETI 指令从中断返回。

图 19-7. 中断请求响应处理算法



xxIF: 中断请求标志

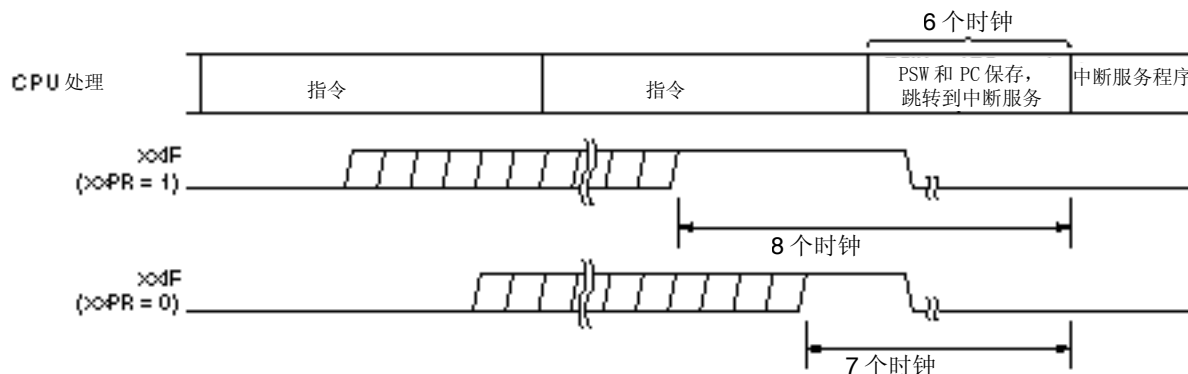
xxMK: 中断屏蔽标志

xxPR: 优先级指定标志

IE: 控制可屏蔽中断请求响应的标志 (1 = 允许、0 = 禁止)

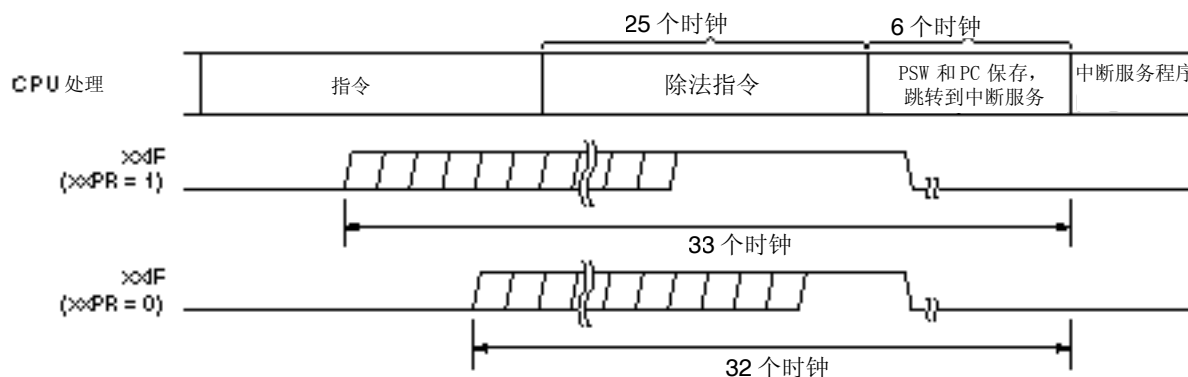
ISP: 指示正在服务的中断优先级别的标志 (0 = 高优先级的中断服务, 1 = 没有中断请求被响应或低优先级的中断服务)

图 19-8. 中断请求响应时序 (最短时间)



备注 1 个时钟: 1/fCPU (fCPU: CPU 时钟)

图 19-9. 中断请求响应时序 (最长时间)



备注 1 个时钟: 1/fCPU (fCPU: CPU 时钟)

19.4.2 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 (003EH、003FH) 传送到 PC 中，然后转移。

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。

19.4.3 中断嵌套

在执行一个中断服务程序时，又响应了其他中断，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 19-5 显示了允许中断嵌套的中断请求之间的关系，图 19-10 为中断嵌套示例。

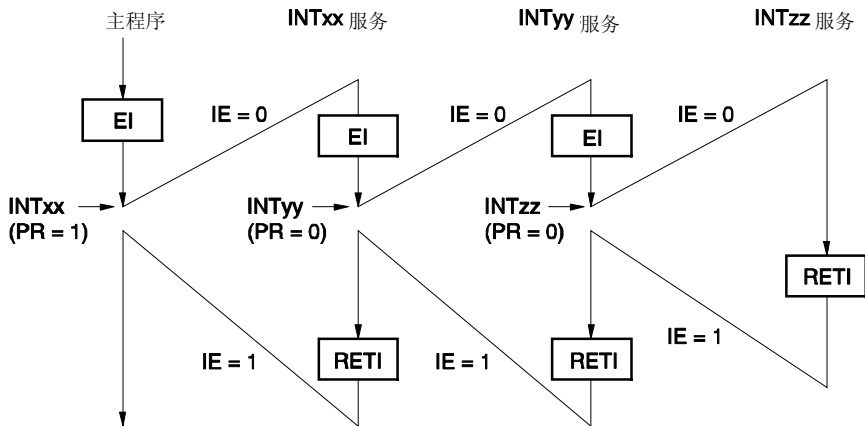
表 19-5. 中断服务期间允许进行中断嵌套的中断请求之间的关系

中断嵌套		可屏蔽中断请求				软件中断请求
		PR = 0		PR = 1		
		IE = 1	IE = 0	IE = 1	IE = 0	
正在服务的中断						
可屏蔽的中断	ISP = 0	○	×	×	×	○
	ISP = 1	○	×	○	×	○
软件中断		○	×	○	×	○

- 备注**
- : 允许中断嵌套
 - ×: 禁止中断嵌套
 - ISP 和 IE 为 PSW 中的标志位。
 ISP = 0: 正在服务一个高优先级中断。
 ISP = 1: 无中断请求被响应，或正在服务一个低优先级中断。
 IE = 0: 禁止响应中断请求。
 IE = 1: 允许响应中断请求。
 - PR 为 PR0L、PR0H、PR1L 和 PR1H 中的标志位。
 PR = 0: 高优先级
 PR = 1: 低优先级

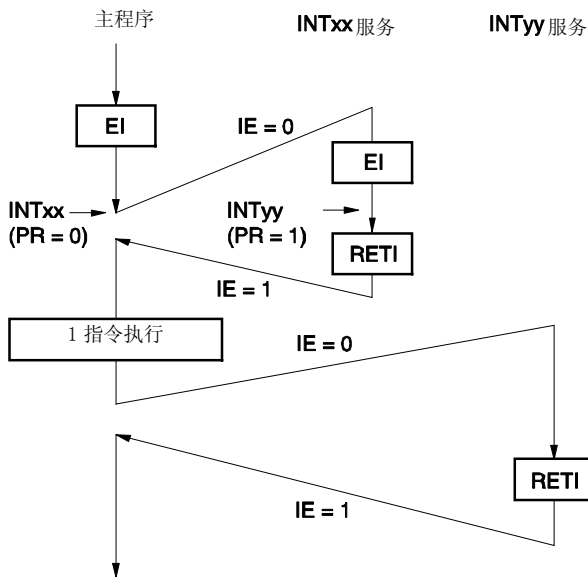
图 19-10. 中断嵌套示例 (1/2)

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

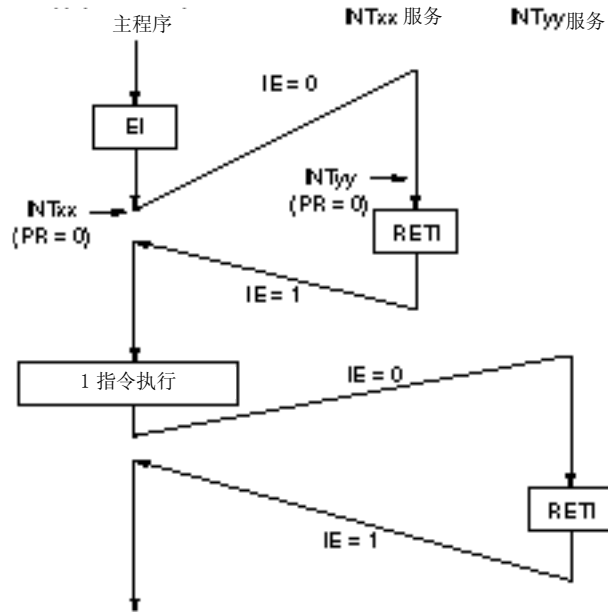


在 INTxx 中断服务期间不响应中断请求 INTyy，因为 INTyy 的优先级低于 INTxx，这样就不会产生中断嵌套。中断请求 INTyy 处于等待状态，在执行一条主程序指令后响应该中断请求。

- PR = 0: 高优先级
- PR = 1: 低优先级
- IE = 0: 禁止响应中断请求

图 19-10. 中断嵌套示例 (2/2)

例 3. 由于不允许中断没有产生中断嵌套



在 INTxx 中断服务期间不允许中断（不执行 EI 指令），因此不响应中断请求 INTyy，这样也不会产生中断嵌套。中断请求 INTyy 处于等待状态，在执行一条主程序指令后响应该中断请求。

PR = 0: 高优先级

IE = 0: 禁止响应中断请求

19.4.4 保持中断请求

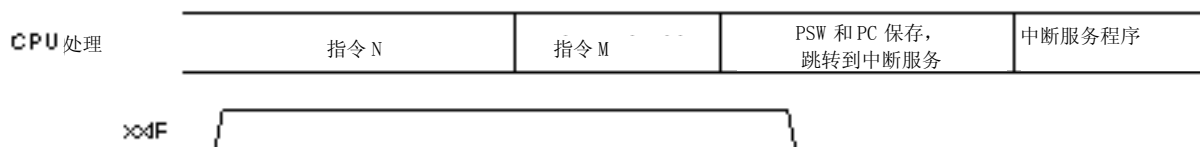
在某些指令执行期间，即使出现中断请求，请求响应也要保持等待状态，直到下一条指令执行结束。以下列出这类指令（中断请求保持指令）。

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW. bit, CY
- MOV1 CY, PSW. bit
- AND1 CY, PSW. bit
- OR1 CY, PSW. bit
- XOR1 CY, PSW. bit
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW. bit, \$addr16
- BF PSW. bit, \$addr16
- BTCLR PSW. bit, \$addr16
- EI
- DI
- 用于 IF0L, IF0H, IF1L, IF1H, MK0L, MK0H, MK1L, MK1H, PR0L, PR0H, PR1L 和 PR1H 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 19-11 显示了处于等待状态的中断请求时序。

图 19-11. 中断请求保持时序



- 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（优先级别）的值不会影响xxIF（中断请求）的操作。

第二十章 按键中断功能

20.1 按键中断的功能

可通过设置按键返回模式寄存器(KRM)和向按键中断输入引脚(KR0~KR6)输入一个下降沿产生按键中断 (INTKR)。

表 20-1. 按键中断检测引脚的分配

标志	描述
KRM0	以 1 位单元控制 KR0 信号。
KRM1	以 1 位单元控制 KR1 信号。
KRM2	以 1 位单元控制 KR2 信号。
KRM3	以 1 位单元控制 KR3 信号。
KRM4	以 1 位单元控制 KR4 信号。
KRM5	以 1 位单元控制 KR5 信号。
KRM6	以 1 位单元控制 KR6 信号。

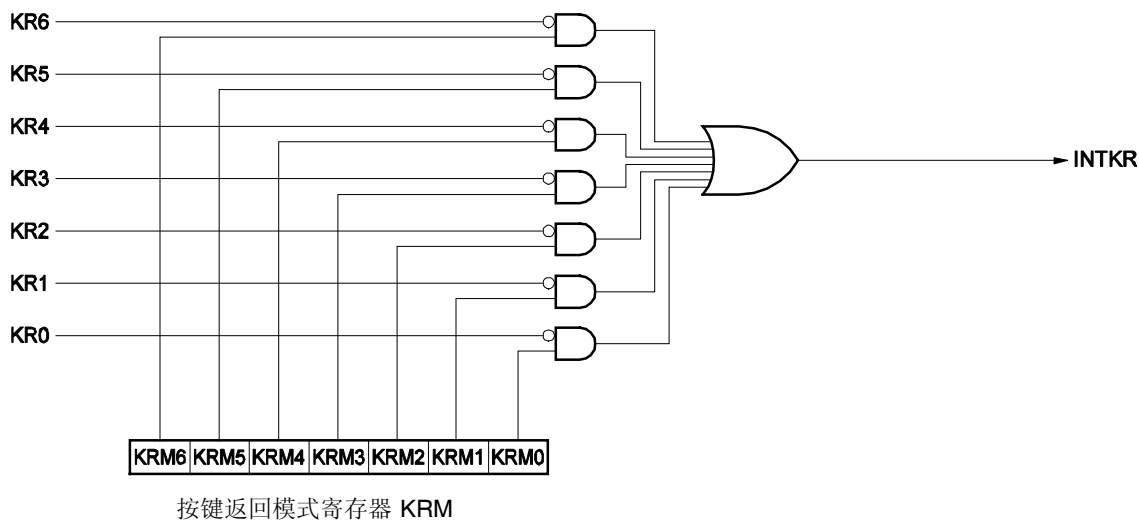
20.2 按键中断的配置

按键中断包括以下硬件。

表 20-2 按键中断的配置

项目	配置
控制寄存器	按键返回模式寄存器 (KRM)

图 20-1. 按键中断的框图



20.3 控制按键中断的寄存器

(1) 按键返回模式寄存器(KRM)

该寄存器使用 KR0 ~ KR6 信号分别控制 KRM0 ~ KRM6 位。

KRM 可由 1 位或 8 位存储器操作指令设置。

复位信号的产生将 KRM 清零(00H)。

图 20-2. 按键返回模式寄存器 (KRM) 的格式

地址: FF6EH 复位后: 00H R/W

符号	7	6	5	4	3	2	1	0
KRM	0	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	按键中断模式控制
0	不检测按键中断信号
1	检测按键中断信号

- 注意事项**
1. 如果 KRM0 ~ KRM6 中任何一位被设置为 1，则将上拉电阻寄存器 7(PU7)的第 0 位~第 6 位 (PU70 ~ PU76)的相应位设置为 1。
 2. 如果 KRM 发生变化，则中断请求标志可能被设置。因此可以先禁止中断，再修改 KRM。先将中断请求标志清零，再允许中断。
 3. 在按键中断模式中没有被使用的位可用作通用端口。

第二十一章 待机功能

21.1 待机功能及配置

21.1.1 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。如果设置 HALT 模式前，高速系统时钟振荡器、内部高速振荡器、内部低速振荡器或副系统时钟振荡器正在使用，则设置后每种时钟的振荡继续。在此模式中，工作电流不如 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的操作非常有效。

(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。

可通过中断请求释放该模式，这样，被中断的操作可以继续执行。由于选择 X1 时钟时，在释放 STOP 模式后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。不能停止副系统时钟的振荡。而当 CPU 使用主系统时钟或副系统时钟时，均可以使用 HALT 模式。
 2. 当切换到 STOP 模式时，在执行 STOP 指令前必须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。

21.1.2 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的状态寄存器（OSTC）
- 振荡稳定时间选择寄存器（OSTS）

备注 有关这些寄存器启动、停止以及时钟选择的情况，可参见第五章 时钟发生器。

(1) 振荡稳定时间计数器的状态寄存器 (OSTC)

这是 X1 时钟振荡稳定时间计数器的状态寄存器。如果 CPU 使用 X1 振荡时钟、内部高速振荡时钟或副系统时钟，则可以检测 X1 时钟振荡稳定时间。

可由 1 位或 8 位存储器操作指令读取 OSTC。

复位释放（通过 $\overline{\text{RESET}}$ 输入、POC、LVI 和 WDT）、STOP 指令以及 MSTOP（MOC 寄存器的第 7 位）= 1 可以将 OSTC 清零（00H）。

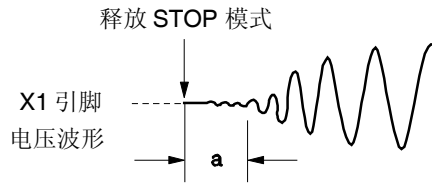
图 21-1. 振荡稳定时间计数器的状态寄存器 (OSTC) 的格式

地址: FFA3H 复位后: 00H R

符号	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	振荡稳定时间的状态		
					$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$	
1	0	0	0	0	$2^{11}/f_x \text{ min.}$	204.8 $\mu\text{s min.}$	102.4 $\mu\text{s min.}$
1	1	0	0	0	$2^{13}/f_x \text{ min.}$	819.2 $\mu\text{s min.}$	409.6 $\mu\text{s min.}$
1	1	1	0	0	$2^{14}/f_x \text{ min.}$	1.64 ms min.	819.2 $\mu\text{s min.}$
1	1	1	1	0	$2^{15}/f_x \text{ min.}$	3.27 ms min.	1.64 ms min.
1	1	1	1	1	$2^{16}/f_x \text{ min.}$	6.55 ms min.	3.27 ms min.

- 注意事项**
1. 在经历上述时间后，从 MOST11 起各位被设置为 1，并保持。
 2. 振荡稳定时间计数器的计数达到 OSTC 设置的振荡稳定时间。如果已进入 STOP 模式并使用内部高速振荡时钟作为 CPU 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 OSTC 设置的振荡稳定时间
 因此需要注意，在释放 STOP 模式后，只有通过 OSTC 设置的振荡稳定时间期间的状态被设置到 OSTC。
 3. X1 时钟振荡稳定等待时间不包括从释放 STOP 模式到时钟振荡启动这段时间（即下图“a”所示的部分）。



备注 fx: X1 时钟振荡频率

(2) 振荡稳定时间选择寄存器(OSTS)

该寄存器用于选择释放 STOP 模式后 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，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。振荡稳定时间可根据 OSTC 设置的时间来检测。

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 05H。

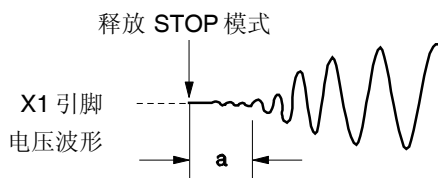
图 21-2. 振荡稳定时间选择寄存器 (OSTS) 的格式

地址: FFA4H 复位后: 05H R/W

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

OSTS2	OSTS1	OSTS0		振荡稳定时间的选择	
				$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$
0	0	1	$2^{11}/f_x$	204.8 μs	102.4 μs
0	1	0	$2^{13}/f_x$	819.2 μs	409.6 μs
0	1	1	$2^{14}/f_x$	1.64 ms	819.2 μs
1	0	0	$2^{15}/f_x$	3.27 ms	1.64 ms
1	0	1	$2^{16}/f_x$	6.55 ms	3.27 ms
其它			禁止设置		

- 注意事项**
- 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。
 - 在 X1 时钟振荡稳定期间不要修改 OSTS 寄存器的值。
 - 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，并使用内部高速振荡时钟时作为 CPU 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，只有通过 OSTS 设置的振荡稳定时间期间的状态被设置到 OSTC。
 - X1 时钟振荡等待时间不包括从释放 STOP 模式到时钟振荡启动这段时间（即下图“a”所示的部分）。



备注 f_x : X1 时钟振荡频率

21.2 待机功能的操作

21.2.1 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 使用的是高速系统时钟、内部高速振荡时钟还是副系统时钟，都可以设置 HALT 模式。

HALT 模式中的操作状态如下所示。

表 21-1. HALT 模式中的操作状态 (1/2)

HALT 模式设置		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_{RH})时	当 CPU 使用 X1 时钟(f_x)时	当 CPU 使用外部主系统时钟(f_{EXCLK})时
项目				
系统时钟		停止 CPU 时钟		
主系统时钟	f_{RH}	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)	保持设置 HALT 模式前的状态
	f_{EXCLK}	由外部时钟输入进行操作或停止		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_{EXCLKS}	由外部时钟输入进行操作或停止		
f_{RL}		保持设置 HALT 模式前的状态		
CPU		操作停止		
Flash 存储器		操作停止		
RAM		保持设置 HALT 模式前的状态		
端口 (锁存器)		保持设置 HALT 模式前的状态		
16 位定时器/事件计数器	00	可操作		
	01 ^{※1}			
8 位定时器/事件计数器	50			
	51			
8 位定时器	H0			
	H1			
钟表定时器				
看门狗定时器		可操作。当通过可选字节设置“可以由软件停止内部低速振荡器”时，停止看门狗定时器的时钟。		
时钟输出		可操作		
A/D 转换器 ^{※2}				
串行接口	UART0			
	UART6			
	CSI10			
	IIC0			
LCD 控制器/驱动器				
乘法器/除法器 ^{※1}				
上电清零 (POC) 功能				
低电压检测功能				
外部中断				

注 1. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386, 78F0386D。

2. 仅限于 μ PD78F037x。

备注

f_{RH} : 内部高速振荡时钟
 f_x : X1 时钟
 f_{EXCLK} : 外部主系统时钟
 f_{XT} : XT1 时钟
 f_{EXCLKS} : 外部副系统时钟
 f_{RL} : 内部低速振荡时钟

表 21-1. HALT 模式中的操作状态(2/2)

HALT 模式设置		当 CPU 使用副系统时钟并执行 HALT 指令时	
		当 CPU 使用 XT1 时钟(f_{XT})时	当 CPU 使用外部副系统时钟(f_{EXCLKS})时
项目			
系统时钟		停止 CPU 时钟	
主系统时钟	f_{RH}	保持设置 HALT 模式前的状态	
	f_x		
	f_{EXCLK}	由外部时钟输入进行操作或停止	
副系统时钟	f_{XT}	操作继续 (不能停止)	保持设置 HALT 模式前的状态
	f_{EXCLKS}	由外部时钟输入进行操作或停止	操作继续 (不能停止)
f_{RL}		保持设置 HALT 模式前的状态	
CPU		操作停止	
Flash 存储器		操作停止	
RAM		保持设置 HALT 模式前的状态	
端口(锁存器)		保持设置 HALT 模式前的状态	
16 位定时器/事件计数器	00 ¹	可操作	
	01 ^{1,2}		
8 位定时器/事件计数器	50 ¹		
	51 ¹		
8 位定时器	H0		
	H1		
钟表定时器			
看门狗定时器		可操作。当通过可选字节设置“可以由软件停止内部低速振荡器”时，停止看门狗定时器的时钟。	
时钟输出		可操作	
A/D 转换器 ³		可操作，但是当外围硬件时钟 (f_{PRS}) 停止时，操作停止。	
串行接口	UART0	可操作	
	UART6		
	CSI10 ¹		
	IIC0 ¹		
LCD 控制器/驱动器			
乘法器/除法器 ²			
上电清零 (POC) 功能			
低电压检测功能			
外部中断			

- 注意**
1. 当 CPU 使用副系统时钟且内部高速振荡时钟已经停止时，不要启动使用外部时钟的外围硬件的功能。
 2. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386, 78F0386D。
 3. 仅限于 μ PD78F037x。

备注

f_{RH} : 内部高速振荡时钟
 f_x : X1 时钟
 f_{EXCLK} : 外部主系统时钟
 f_{XT} : XT1 时钟
 f_{EXCLKS} : 外部副系统时钟
 f_{RL} : 内部低速振荡时钟

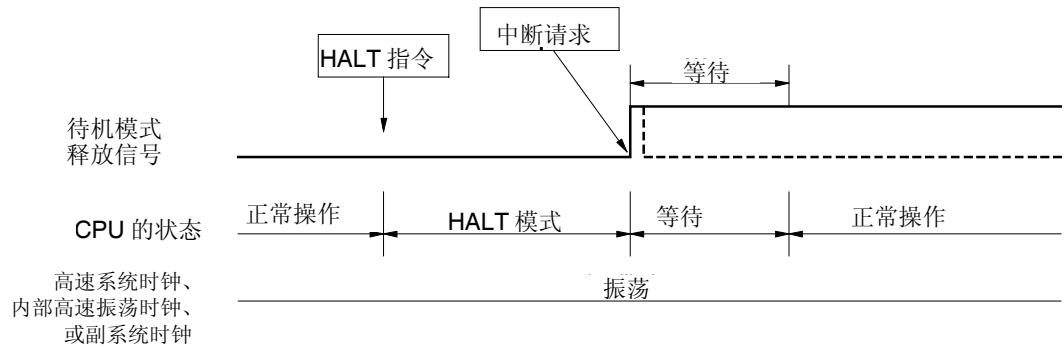
(2) 释放 HALT 模式

以下两种中断源可以释放 HALT 模式。

(a) 由没有被屏蔽的中断请求释放

当产生一个没有被屏蔽的中断时，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 21-3. 通过产生中断请求释放 HALT 模式



- 备注**
1. 虚线表示释放待机模式的中断请求被响应的情况。
 2. 等待时间如下所示：
 - 执行向量中断服务程序时：8 或 9 个 时钟
 - 不执行向量中断服务程序时：2 或 3 个 时钟

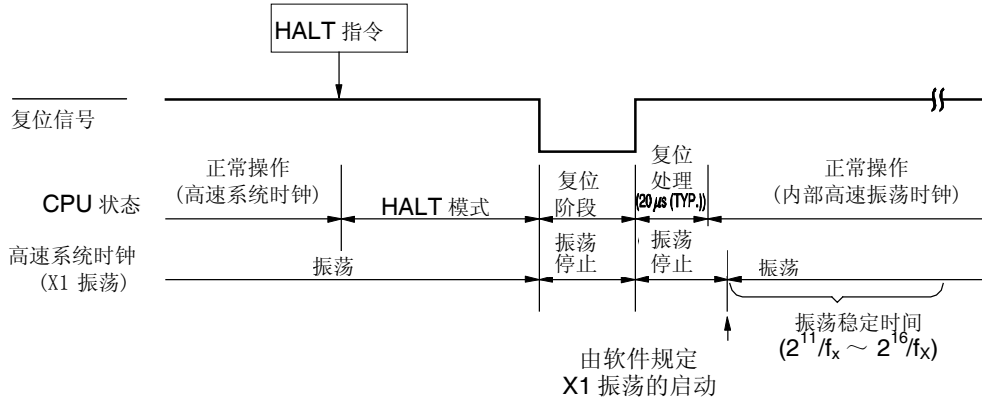
(b) 通过产生的复位信号释放

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。

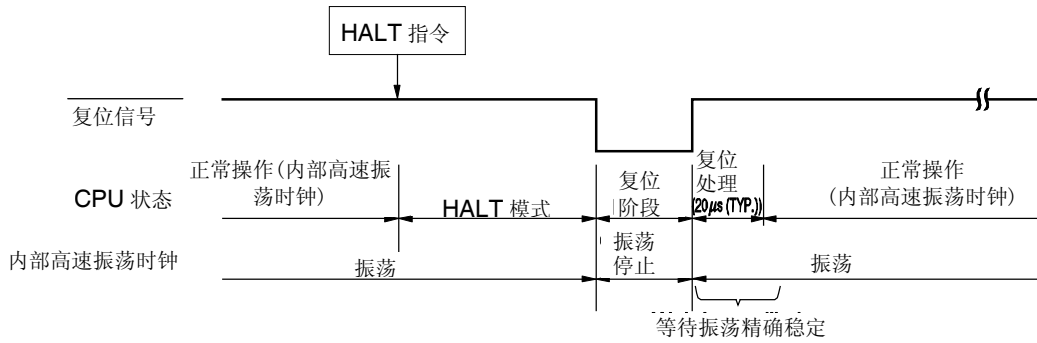
<R>

图 21-4. 通过复位释放 HALT 模式

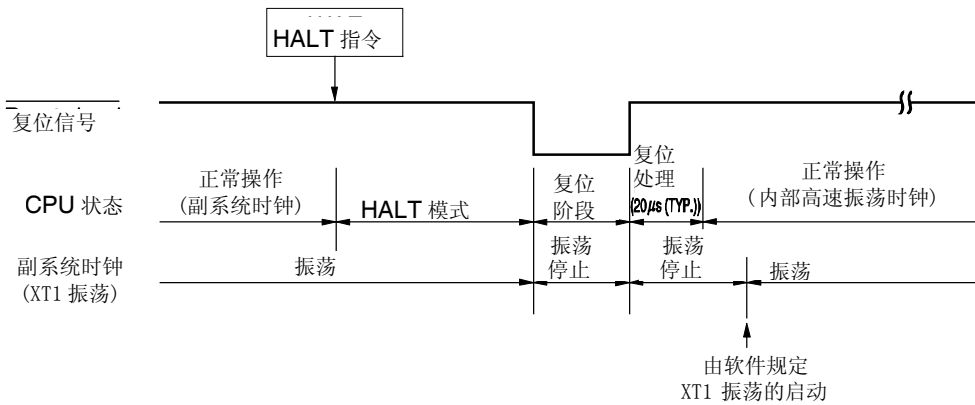
(1) CPU 使用高速系统时钟



(2) CPU 使用内部高速振荡时钟



(3) CPU 使用副系统时钟



备注 fx: X1 时钟振荡频率

表 21-2. 在 HALT 模式中中断请求对应的操作

释放源	MK _{xx}	PR _{xx}	IE	ISP	操作
可屏蔽中断请求	0	0	0	×	执行下一个地址的指令
	0	0	1	×	执行中断服务程序
	0	1	0	1	执行下一个地址的指令
	0	1	×	0	
	0	1	1	1	执行中断服务程序
	1	×	×	×	保持 HALT 模式
复位	—	—	×	×	复位处理

×: 不必考虑

21.2.2 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。仅当设置 STOP 模式前 CPU 使用主系统时钟时才可以设置 STOP 模式。

注意事项 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清零时，则立即释放待机模式。因此，在执行 STOP 指令后，立即将 STOP 模式复位到 HALT 模式，并且在经历了 OSTS 设置的等待时间后，系统返回操作模式。

STOP 模式中的操作状态如下所示。

表 21-3. STOP 模式中的操作状态

STOP 模式设置		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_{RH})时	当 CPU 使用 X1 时钟(f_x)时	当 CPU 使用外部主系统时钟(f_{EXCLK})时
项目				
系统时钟		停止 CPU 时钟		
主系统时钟	f_{RH}	停止		
	f_x			
	f_{EXCLK}	输入无效		
副系统时钟	f_{XT}	保持设置 STOP 模式前的状态		
	f_{EXCLKS}	由外部时钟输入进行操作或停止		
f_{RL}		保持设置 STOP 模式前的状态		
CPU		操作停止		
Flash 存储器		操作停止		
RAM		保持设置 STOP 模式前的状态		
端口 (锁存器)		保持设置 STOP 模式前的状态		
16 位定时器/事件计数器	00	操作停止		
	01 ^{注1}			
8 位定时器/事件计数器	50	仅当选择 TI50 作为计数时钟时可操作		
	51	仅当选择 TI51 作为计数时钟时可操作		
8 位定时器	H0	在 8 位定时器/事件计数器 50 操作期间, 仅当选择 TM50 输出作为计数时钟时可操作		
	H1	仅当选择 f_{RL} , $f_{RL}/2^7$, $f_{RL}/2^9$ 作为计数时钟时可操作		
钟表定时器		仅当选择副系统时钟作为计数时钟时可操作		
看门狗定时器		可操作。当可选字节设置“可由软件停止内部低速振荡器”时, 停止看门狗定时器的时钟。		
时钟输出		仅当选择副系统时钟作为计数时钟时可操作		
A/D 转换器 ^{注2}		操作停止		
串行接口	UART0	在 8 位定时器/事件计数器 50 操作期间, 仅当选择 TM50 输出作为计数时钟时可操作		
	UART6			
	CSI10	仅当选择外部时钟作为串行时钟时可操作		
	IIC0			
LCD 控制器/驱动器		仅当选择副系统时钟作为计数时钟时可操作		
乘法器/除法器 ^{注1}		操作停止		
上电清零 (POC) 功能		可操作		
低电压检测功能				
外部中断				

注 1. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386, 78F0386D。

2. 仅限于 μ PD78F037x。

备注

f_{RH} : 内部高速振荡时钟

f_x : X1 时钟

f_{EXCLK} : 外部主系统时钟

f_{XT} : XT1 时钟

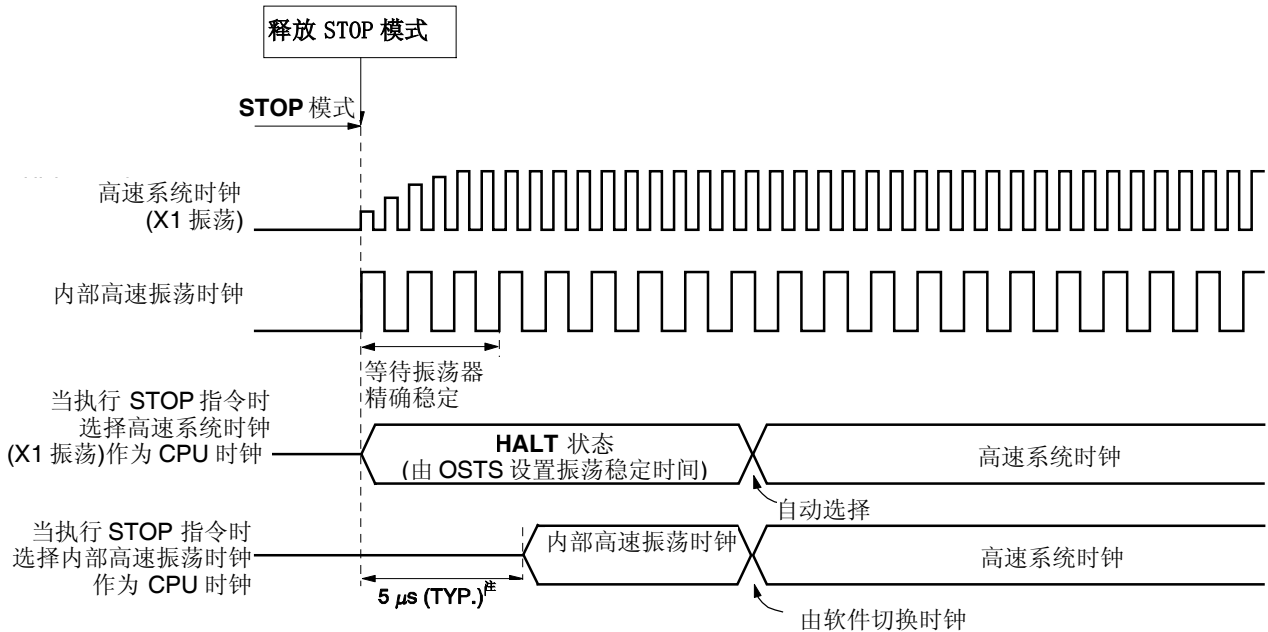
f_{EXCLKS} : 外部副系统时钟

f_{RL} : 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 模式下时钟停止操作的外围硬件，必须重新启动外围硬件。
 2. 即使通过选项字节选择“可由软件停止内部低速振荡器”，在 STOP 模式下，内部低速振荡时钟仍继续振荡，并保持 STOP 模式设置前的状态。要在 STOP 模式下停止内部低速振荡器的振荡，应通过软件停止该振荡器，然后执行 STOP 指令。
 3. 当 CPU 使用高速系统时钟(X1 振荡)时，为了在 STOP 模式释放后缩短振荡稳定时间，可在执行 STOP 指令前将 CPU 时钟暂时切换到内部高速振荡时钟。STOP 模式释放后，在将 CPU 时钟从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 AMPH 设置为 1 时执行 STOP 指令，当内部高速振荡时钟或外部主系统时钟作为 CPU 时钟时，在 STOP 模式释放后，内部高速振荡时钟或外部主系统时钟提供给 CPU 时钟 5 μ s (最小)。

(2) 释放 STOP 模式

图 21-5. 释放 STOP 模式时的操作时序



注 当 AMPH = 1

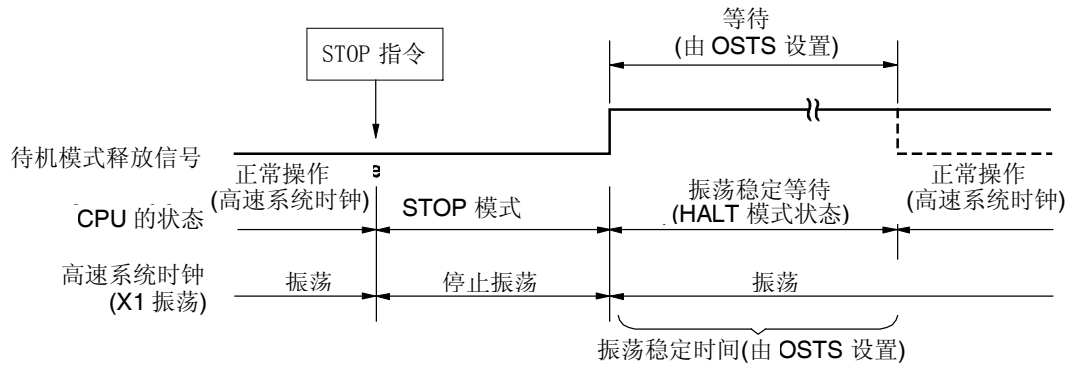
STOP 模式可通过以下两种方式释放。

(a) 由没有被屏蔽的中断请求释放

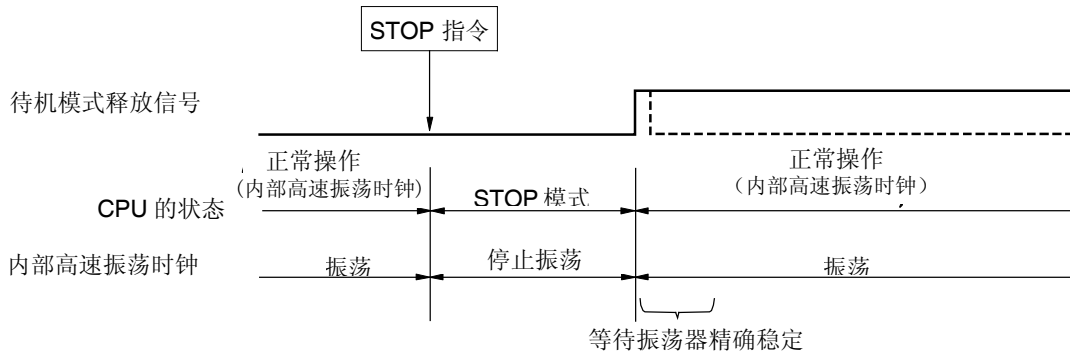
当产生一个没有被屏蔽的中断请求时，释放 STOP 模式。经历振荡稳定时间后，如果允许响应中断，则执行向量中断服务程序。如果禁止响应中断，则执行下一个地址的指令。

图 21-6. 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)



(2) 当 CPU 使用内部高速振荡频率

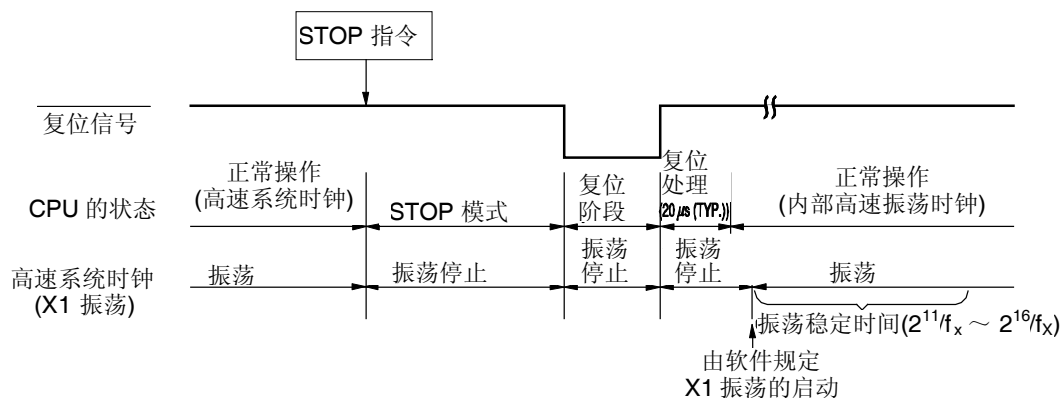


备注 虚线表示释放待机模式的中断请求被响应的情况。

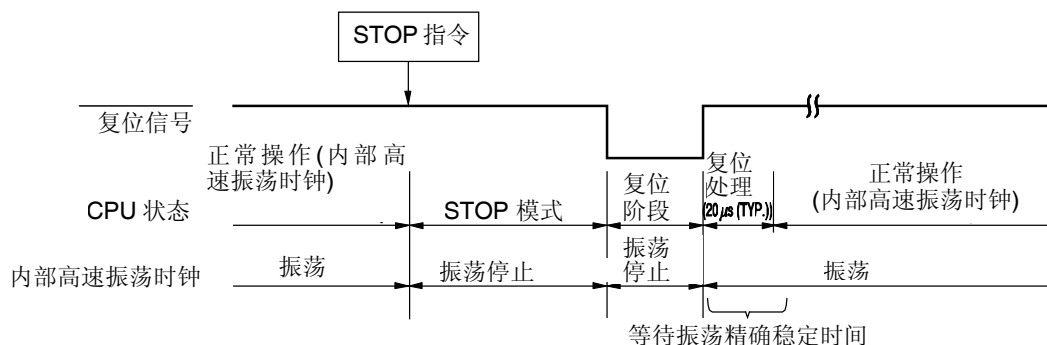
(b) 由复位信号的产生释放

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后执行之。

图 21-7. 由复位释放 STOP 模式



(2) 当 CPU 使用内部高速振荡时钟



备注 fx: X1 时钟振荡频率

表 21-4. 在 STOP 模式中 与中断请求对应的操作

释放源	MK _{xx}	PR _{xx}	IE	ISP	操作
可屏蔽的中断请求	0	0	0	×	执行下一个地址的指令
	0	0	1	×	执行中断服务程序
	0	1	0	1	执行下一个地址的指令
	0	1	×	0	
	0	1	1	1	执行中断服务程序
	1	×	×	×	保持 STOP 模式
复位	-	-	×	×	复位处理

x: 不必考虑

第二十二章 复位功能

以下 4 种操作用于产生复位信号。

- (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) 通过比较低电压检测电路 (LVI) 的检测电压和供电电压引起的内部复位

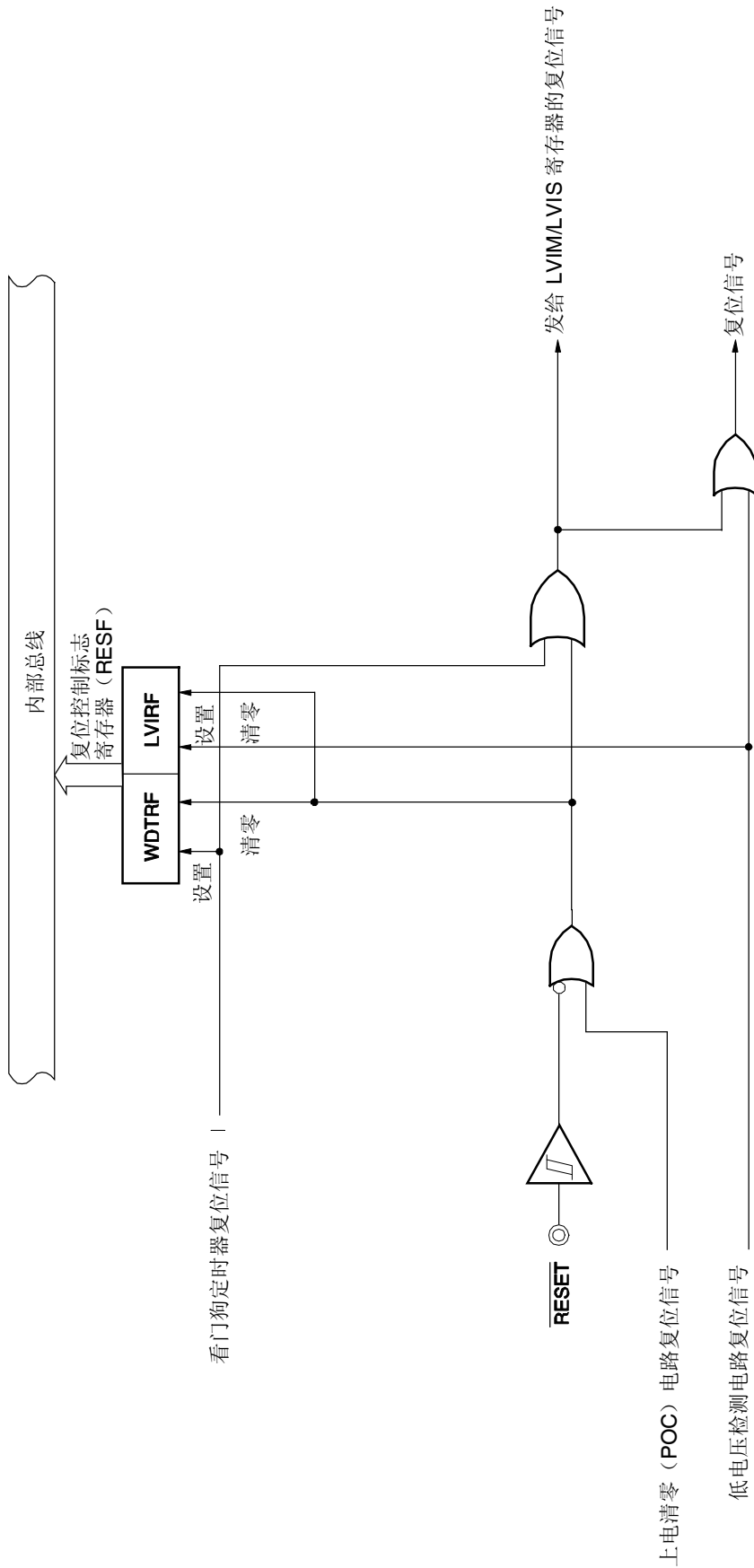
外部复位与内部复位在功能上没有什么区别。在这两种情况中，当产生复位信号时，程序都是从地址 0000H 和 0001H 处开始执行。

当 $\overline{\text{RESET}}$ 引脚输入为低电平、看门狗定时器溢出或 POC 和 LVI 电路的电压检测都可以引起复位，每项硬件的设置状态如表 22-1 和 22-2 所示。在复位信号产生期间或在复位释放后的振荡稳定时间内，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 22-2 ~ 22-4)。由 POC 和 LVI 电路供电电压检测引起的复位，在复位后，当 $V_{DD} \geq V_{POC}$ 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（参见第二十三章 上电清零电路 和 第二十四章 低电压检测电路）。

- 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 。
 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和外部副系统时钟输入无效。
 3. 在通过复位释放 STOP 模式时，复位输入期间保持 STOP 模式的内容。但端口引脚变为高阻态。

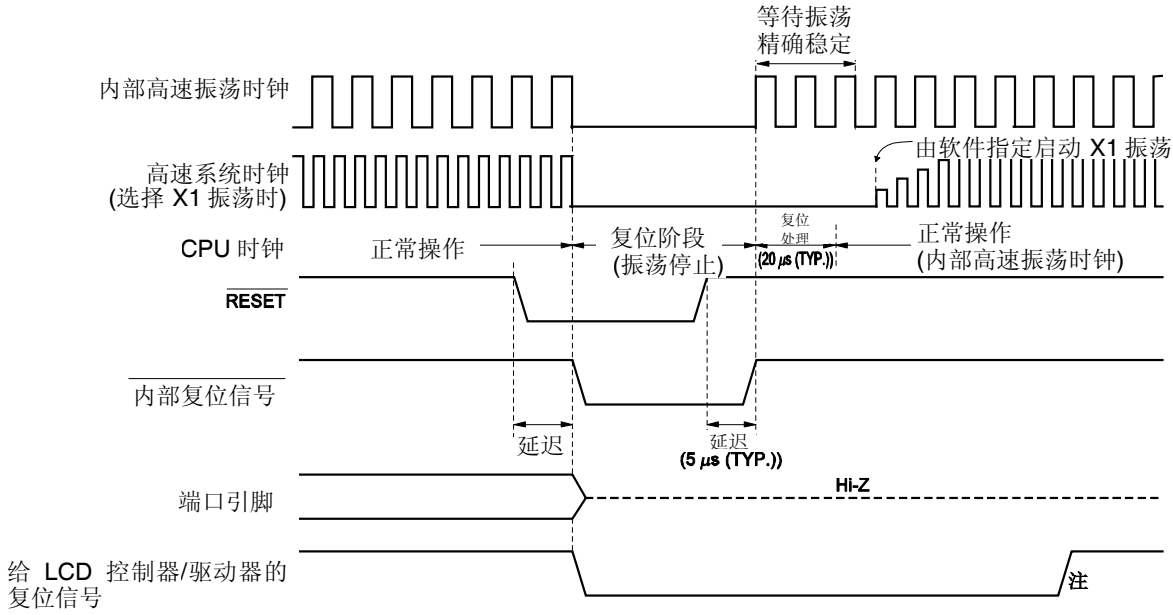
图 22-1. 复位功能的框图



注意事项 LVI 电路内部复位功能不能复位 LVI 电路。

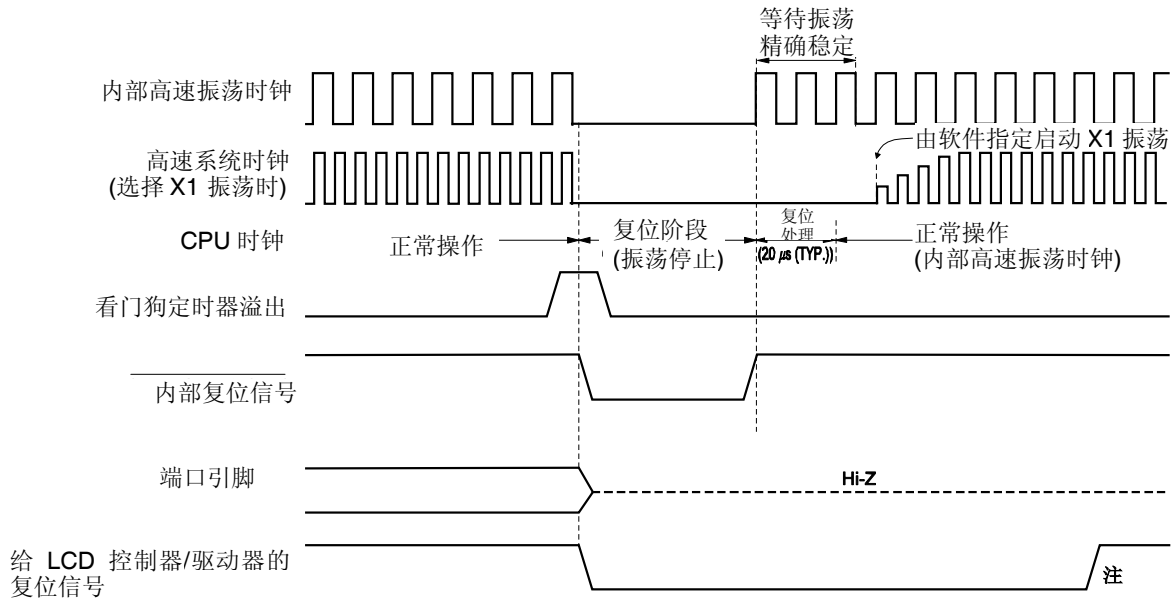
- 备注**
- 1. LVIM: 低电压检测寄存器
 - 2. LVIS: 低电压检测等级选择寄存器

图 22-2. 由 RESET 输入进行复位的时序



注 通过软件设置 P130 为 1 (端口模式寄存器 13 的第 0 位)。

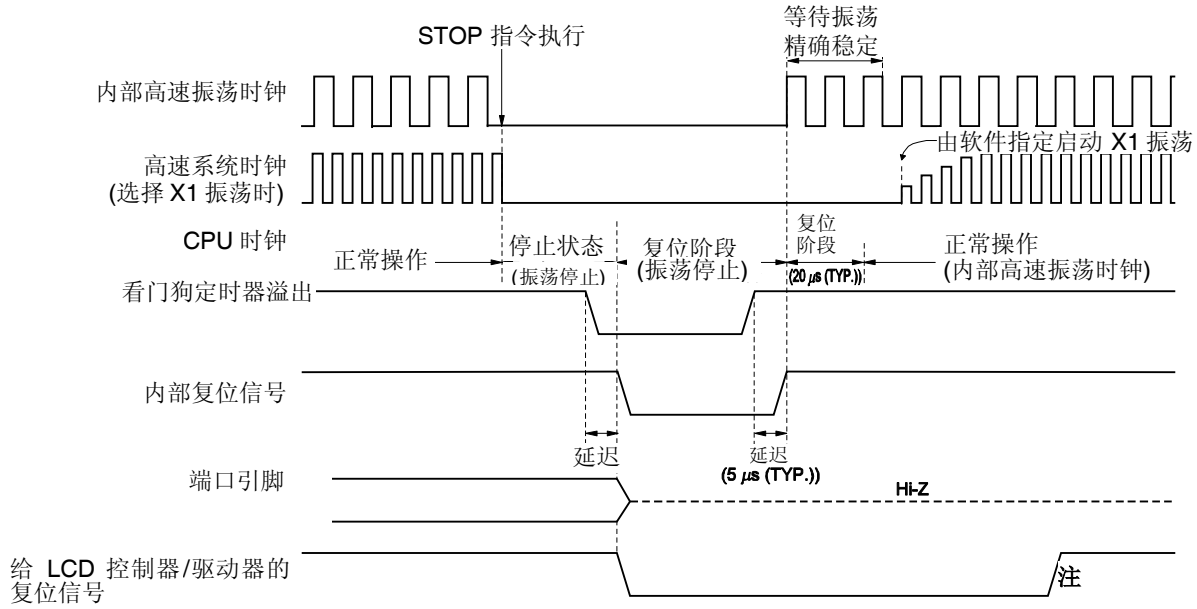
图 22-3. 由于看门狗定时器溢出进行复位的时序



注 通过软件设置 P130 为 1 (端口模式寄存器 13 的第 0 位)。

注意事项 看门狗定时器内部复位功能也可以复位看门狗定时器

图 22-4. 在 STOP 模式下通过 RESET 输入进行复位的时序



注 通过软件设置 P130 为 1 (端口模式寄存器 13 的第 0 位)。

备注 如需了解上电清零 (POC) 电路和低电压检测电路的复位时序, 可参见第二十三章 上电清零电路 和 第二十四章 低电压检测电路。

表 22-1. 复位期间的操作状态

项目		复位期间	
系统时钟		停止 CPU 时钟	
主系统时钟	f _{RH}	停止操作	
	f _X	停止操作(引脚为 I/O 端口模式)	
	f _{EXCLK}	时钟输入无效(引脚为 I/O 端口模式)	
副系统时钟	f _{XT}	停止操作(引脚为 I/O 端口模式)	
	f _{EXCLKS}	时钟输入无效(引脚为 I/O 端口模式)	
f _{RL}		停止操作	
CPU			
Flash 存储器			
RAM			
端口 (锁存器)			
16 位定时器/事件计数器	00		
	01 ^{注1}		
8 位定时器/事件计数器	50		
	51		
8 位定时器	H0		
	H1		
钟表定时器			
看门狗定时器			
时钟输出			
A/D 转换器 ^{注2}			
串行接口	UART0		
	UART6		
	CSI10		
	IIC0		
LCD 控制器/驱动器			
乘法器/除法器 ^{注1}			
上电清零功能			可操作
低电压检测功能			停止操作
外部中断			

注 1. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386, 78F0386D。

2. 仅限于 μ PD78F037x。

备注

- f_{RH}: 内部高速振荡时钟
- f_X: X1 振荡时钟
- f_{EXCLK}: 外部主系统时钟
- f_{XT}: XT1 振荡时钟
- f_{EXCLKS}: 外部副系统时钟
- f_{RL}: 内部低速振荡时钟

表 22-2. 复位响应后的硬件状态 (1/3)

硬件		响应复位后的状态 ^{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)		不确定
程序状态字(PSW)		02H
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 (P0 ~ P3, P7, P12, P13) (输出锁存器)		00H
端口模式寄存器(PM0 ~ PM3, PM6, PM7, PM12, PM14)		FFH
上拉电阻选择寄存器(PU0, PU1, PU3, PU7, PU12)		00H
内部扩展 RAM 容量切换寄存器(IXS)		0CH ^{注3}
内部存储器容量切换寄存器(IMS)		CFH ^{注3}
存储器 bank 选择寄存器 (BANK)		00H
时钟操作模式选择寄存器 (OSCCTL)		00H
处理器时钟控制寄存器(PCC)		01H
内部振荡模式寄存器(RCM)		80H
主 OSC 控制寄存器(MOC)		80H
主时钟模式寄存器(MCM)		00H
振荡稳定时间计数器的状态寄存器(OSTC)		00H
振荡稳定时间选择寄存器(OSTS)		05H
16 位定时器/事件计数器 00, 01 ^{注4}	定时器计数器 00, 01 (TM00, TM01)	0000H
	捕捉/比较寄存器 000, 010, 001, 011 (CR000, CR010, CR001, CR011)	0000H
	模式控制寄存器 00, 01 (TMC00, TMC01)	00H
	预分频器模式寄存器 00, 01 (PRM00, PRM01)	00H
	捕捉/比较控制寄存器 00 (CRC00)	00H
	定时器输出控制寄存器 00 (TOC00)	00H

<R>

- 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. 当在待机模式中进行复位时，即使复位后也保持复位前的状态。
 3. 在所有 78K0/LF2 产品中，无论内部存储器容量如何，当复位释放后内部存储器容量切换寄存器(IMS)和内部扩展 RAM 容量切换寄存器(IXS)的初值为常数(IMS = CFH, IXS = 0CH)。因此复位释放后必须为每个产品设置以下值。

Flash 存储器版本(78K0/LF2)	IMS	IXS
μPD78F0372, 78F0382	C6H	0CH
μPD78F0373, 78F0383	C8H	
μPD78F0374, 78F0384	CCH	0AH
μPD78F0375, 78F0385	CFH	08H
μPD78F0376, 78F0386, 78F0376D, 78F0386D ^{注5}	CCH	04H

4. 仅限于μPD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386 和 78F0386D 有 16 位定时器/事件计数器 01。
5. 带有片上调试功能产品的 ROM 和 RAM 容量可根据调试目标产品进行调试。根据调试目标产品设置 IMS 和 IXS。

<R>

表 22-2. 复位响应后的硬件状态 (2/3)

硬件		响应复位后的状态 ^{注1}
8 位定时器/事件计数器 50, 51	定时器计数器 50, 51 (TM50, TM51)	00H
	比较寄存器 50, 51 (CR50, CR51)	00H
	定时器时钟选择寄存器 50, 51 (TCL50, TCL51)	00H
	模式控制寄存器 50, 51 (TMC50, TMC51)	00H
8 位定时器 H0, H1	比较寄存器 00, 10, 01, 11 (CMP00, CMP10, CMP01, CMP11)	00H
	模式寄存器 (TMHMD0, TMHMD1)	00H
	载波控制寄存器 1 (TMCYC1) ^{注2}	00H
钟表定时器	操作模式寄存器 (WTM)	00H
时钟输出控制器	时钟输出选择寄存器 (CKS)	00H
看门狗定时器	允许寄存器 (WDTE)	1AH/9AH ^{注3}
A/D 转换器 ^{注4}	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
	模拟输入通道选择寄存器 (ADS)	00H
	A/D 端口配置寄存器 (ADPC)	00H
串行接口 UART0	接收缓冲寄存器 0 (RXB0)	FFH
	发送移位寄存器 0 (TXS0)	FFH
	异步串行接口操作模式寄存器 0 (ASIM0)	01H
	异步串行接口接收错误状态寄存器 0 (ASIS0)	00H
	波特率发生器控制寄存器 0 (BRGC0)	1FH
串行接口 UART6	接收缓冲寄存器 6 (RXB6)	FFH
	发送缓冲寄存器 6 (TXB6)	FFH
	异步串行接口操作模式寄存器 6 (ASIM6)	01H
	异步串行接口接收错误状态寄存器 6 (ASIS6)	00H
	异步串行接口发送状态寄存器 6 (ASIF6)	00H
	时钟选择寄存器 6 (CKSR6)	00H
	波特率发生器控制寄存器 6 (BRGC6)	FFH
	异步串行接口控制寄存器 6 (ASICL6)	16H
	输入切换控制寄存器 (ISC)	00H
串行接口 CSI10	发送缓冲寄存器 10 (SOTB10)	00H
	串行 I/O 移位寄存器 10 (SIO10)	00H
	串行操作模式寄存器 10 (CSIM10)	00H
	串行时钟选择寄存器 10 (CSIC10)	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. 仅限于 8 位定时器 H1。
 3. 通过设置选项字节确定 WDTE 的复位值。
 4. 仅限于 μ PD78F037x。

表 22-2. 复位响应后的硬件状态 (3/3)

硬件		响应复位后的状态 ^{注 1}
串行接口 IIC0	移位寄存器 0 (IIC0)	00H
	控制寄存器 0 (IIC00)	00H
	从设备地址寄存器 0 (SVA0)	00H
	时钟选择寄存器 0 (IICCL0)	00H
	功能扩展寄存器 0 (IICX0)	00H
	状态寄存器 0 (IICCS0)	00H
	标志寄存器 0 (IICF0)	00H
LCD 控制器/驱动器	LCD 模式设置寄存器 (LCDMD)	00H
	LCD 显示模式寄存器 (LCDM)	00H
	LCD 时钟控制寄存器(LCDC)	00H
	LCD 升压控制寄存器 0 (VLCG0)	00H
乘法器/除法器 ^{注 3}	余数寄存器 0 (SDR0)	0000H
	乘法/除法数据寄存器 A0 (MDA0H, MDA0L)	0000H
	乘法/除法数据寄存器 B0 (MDB0)	0000H
	乘法器/除法器控制寄存器 0 (DMUC0)	00H
按键中断	按键返回模式寄存器 (KRM)	00H
复位功能	复位控制标志寄存器 (RESF)	00H ^{注 3}
低电压检测电路	低电压检测寄存器 (LVIM)	00H ^{注 3}
	低电压检测等级选择寄存器 (LVIS)	00H ^{注 3}
中断	请求标志寄存器 0L, 0H, 1L, 1H (IF0L, IF0H, IF1L, IF1H)	00H
	屏蔽标志寄存器 0L, 0H, 1L, 1H (MK0L, MK0H, MK1L, MK1H)	FFH
	优先级指定标志寄存器 0L, 0H, 1L, 1H (PR0L, PR0H, PR1L, PR1H)	FFH
	外部中断上升沿允许寄存器 (EGP)	00H
	外部中断下降沿允许寄存器 (EGN)	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. 仅限于 μ PD78F0374, 78F0375, 78F0376, 78F0376D, 78F0384, 78F0385, 78F0386, 78F0386D。
3. 这些值依据复位源的不同而变化。

复位源		RESET 输入	由 POC 进行复位	由 WDT 进行复位	由 LVI 进行复位
寄存器	RESF	清零(0)	清零(0)	设置(1)	保持
	WDTRF 位			保持	设置(1)
	LVIRF 位				
	LVIM	清零(00H)	清零(00H)	清零(00H)	保持
	LVIS				

22.1 确认复位源的寄存器

78K0/LF2 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。

$\overline{\text{RESET}}$ 输入、由 POC 电路引起的复位输入以及读取 RESF 可将 RESF 清零 (00H)。

图 22-5. 复位控制标志寄存器 (RESF) 的格式

地址: FFACH 复位后: 00H^注 R

符号	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	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 复位后的值依据复位源而变化。

注意事项 不能通过 1 位存储器操作指令读取数据。

产生复位请求时的 RESF 状态如表 22-3 所示。

表 22-3. 产生复位请求时的 RESF 状态

复位源 标志	$\overline{\text{RESET}}$ 输入	由 POC 引起的 复位	由 WDT 引起的 复位	由 LVI 引起的复 位
WDTRF	清零(0)	清零(0)	设置(1)	保持
LVIRF			保持	设置 (1)

23.1 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。
在 1.59 V POC 模式下(选项字节: POCMODE = 0)，当供电电压(V_{DD})超过 $1.59\text{ V} \pm 0.15\text{ V}$ 时，释放复位信号。
在 2.7 V/1.59 V POC 模式下(选项字节: POCMODE = 1)，当供电电压(V_{DD})超过 $2.7\text{ V} \pm 0.2\text{ V}$ 时，释放复位信号。
- 比较供电电压 (V_{DD}) 和检测电压 ($V_{POC} = 1.59\text{ V} \pm 0.15\text{ V}$)，并在 $V_{DD} < V_{POC}$ 时产生内部复位信号，在 $V_{DD} \geq V_{POC}$ 释放复位。

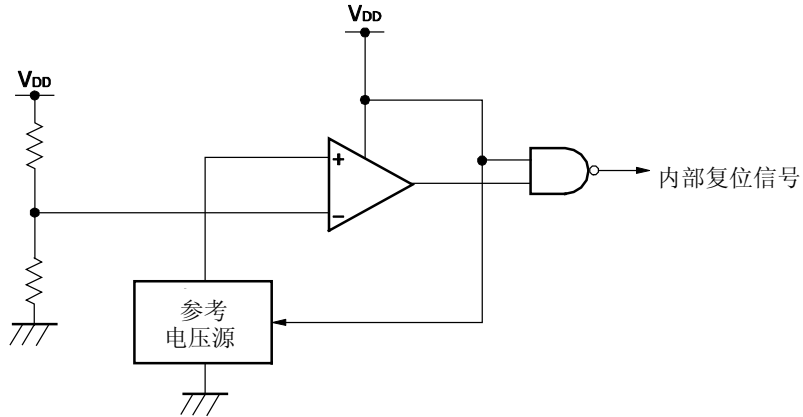
注意事项 如果由 POC 电路产生内部复位信号，则将复位控制标志寄存器（RESF）清零（00H）。

备注 本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）或低电压检测电路（LVI）产生的内部复位信号。当 WDT 或 LVI 产生内部复位信号时，不对 RESF 清零（00H），标志位被设置为 1。
如需了解 RESF 的详细内容，参见第二十二章 复位功能。

23.2 上电清零电路的配置

上电清零电路的框图如图 23-1 所示。

图 23-1. 上电清零电路的框图



23.3 上电清零电路的操作

(1) 1.59 V POC 模式 (选项字节: POCMODE = 0)

- 上电时产生内部复位信号。当供电电压(V_{DD})超过检测电压($V_{POC} = 1.59\text{ V} \pm 0.15\text{ V}$)时, 释放复位状态。
- 比较供电电压(V_{DD})和检测电压($V_{POC} = 1.59\text{ V} \pm 0.15\text{ V}$)。当 $V_{DD} < V_{POC}$ 时, 产生内部复位信号。当 $V_{DD} \geq V_{POC}$, 释放内部复位信号。

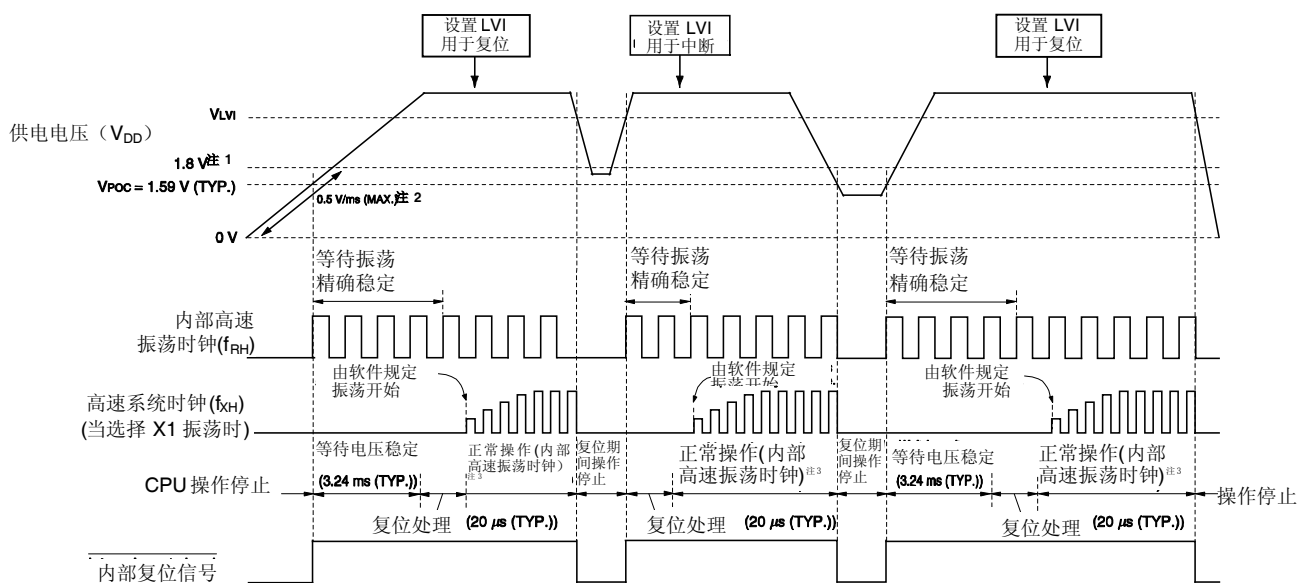
(2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)

- 上电时产生内部复位信号。当供电电压(V_{DD})超过检测电压($V_{DDPOC} = 2.7\text{ V} \pm 0.2\text{ V}$)时, 释放复位状态。
- 比较供电电压(V_{DD})和检测电压($V_{POC} = 1.59\text{ V} \pm 0.15\text{ V}$)。当 $V_{DD} < V_{POC}$ 时, 产生内部复位信号。当 $V_{DD} \geq V_{POC}$ 时, 释放内部复位信号。

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如下所示。

图 23-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序 (1/2)

(1) 1.59 V POC 模式 (选项字节: POCMODE = 0)



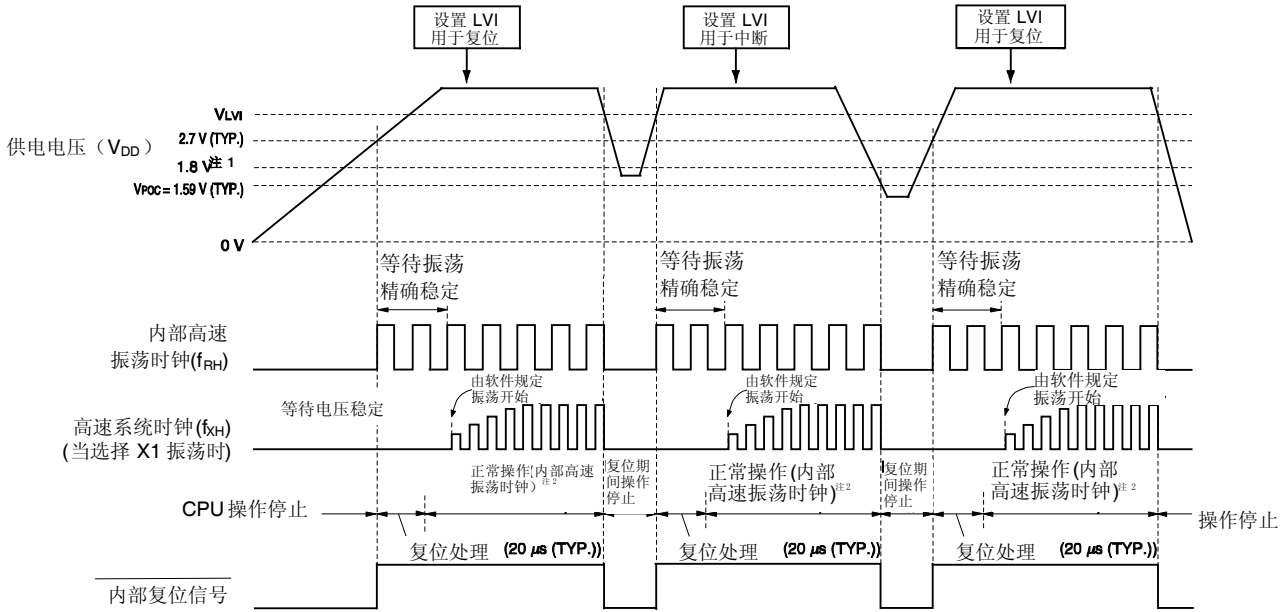
- 注
1. 保证操作的电压范围： $1.8V \leq V_{DD} \leq 5.5V$ 。当供电电压下降时，为使状态低于 1.8V 复位状态，可以使用低电压检测电路的复位功能，或通过 **RESET** 引脚输入低电平。
 2. 如果上电后电压上升到 1.8V 的速率小于 0.5 V/ms (MIN.)，则在上电后且电压抵达 1.8V 以前输入一个低电平到 **RESET** 引脚，或者通过使用选项字节 ($POCMODE = 1$) 设置 2.7 V/1.59 V POC 模式。
 3. 可以选择内部高速振荡时钟、高速系统时钟或副系统时钟作为 CPU 时钟。如果要使用 X1 时钟，可以通过 **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。

注意事项 在释放复位状态后通过软件设置低电压检测电路 (参见第二十四章 低电压检测电路)。

备注 V_{LVI} : LVI 检测电压
 V_{POC} : POC 检测电压

图 23-2. 由上电清零电路和低电压检测电路产生内部复位信号的时序(2/2)

(2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)



- 注
1. 保证操作的电压范围： $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ 。当供电电压下降时，为使状态低于 1.8 V 复位状态，可以使用低电压检测电路的复位功能，或向 **RESET** 引脚输入低电平。
 2. 可以选择内部高速振荡时钟、高速系统时钟或子系统时钟作为 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。

注意事项 在释放复位状态后通过软件设置低电压检测电路 (参见第二十四章 低电压检测电路)。

备注 V_{LVI} : LVI 检测电压
 V_{POC} : POC 检测电压

23.4 上电清零电路使用注意事项

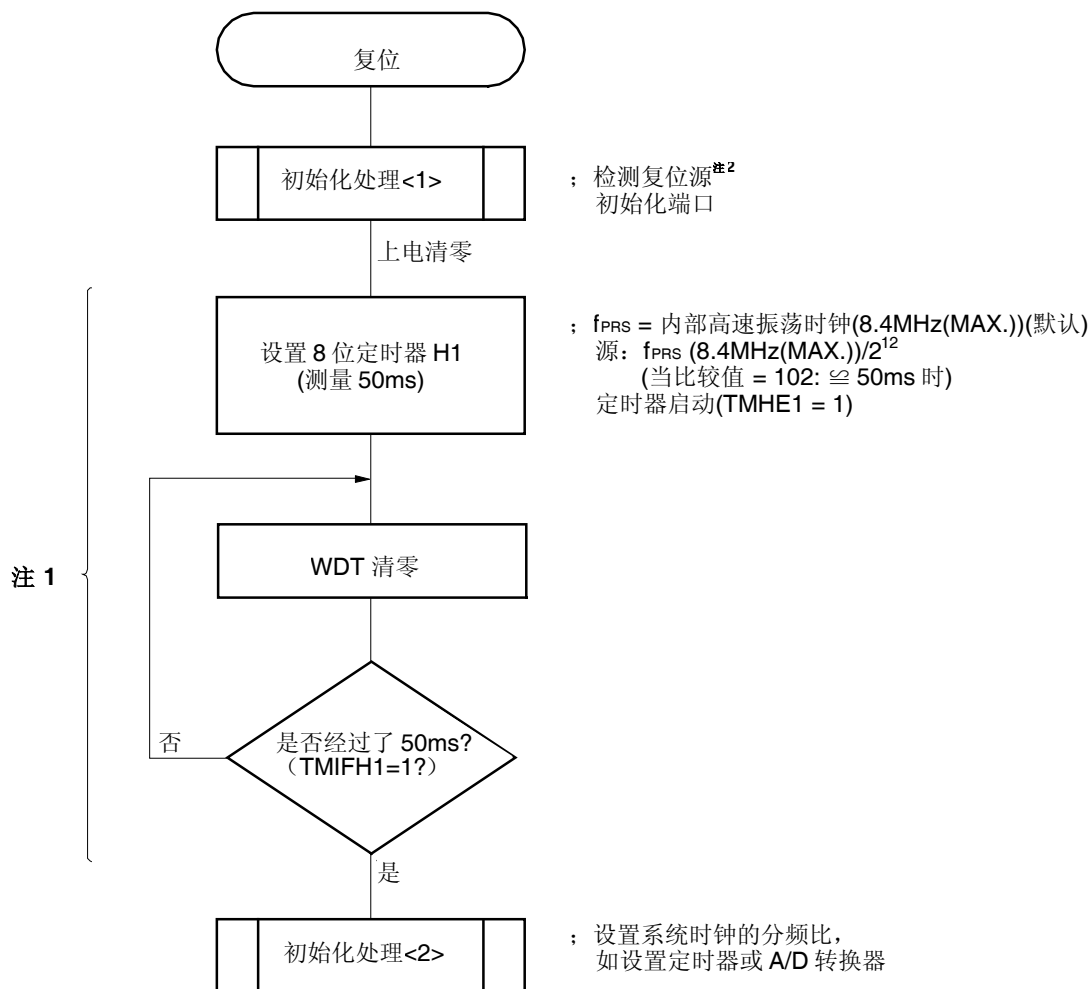
在系统中，如果供电电压 (VDD) 接近 POC 检测电压 (VPOC) 时的一段时间内产生波动，则系统可能会反复复位和释放复位。在这种情况下，可采用下列方法任意设置从复位释放到微控制器的启动所需要的时间。

<方法>

在释放复位信号后，通过软件计数器（使用一个定时器）等待系统供电电压的波动期，然后对端口初始化。

图 23-3. 复位释放后软件处理过程示例 (1/2)

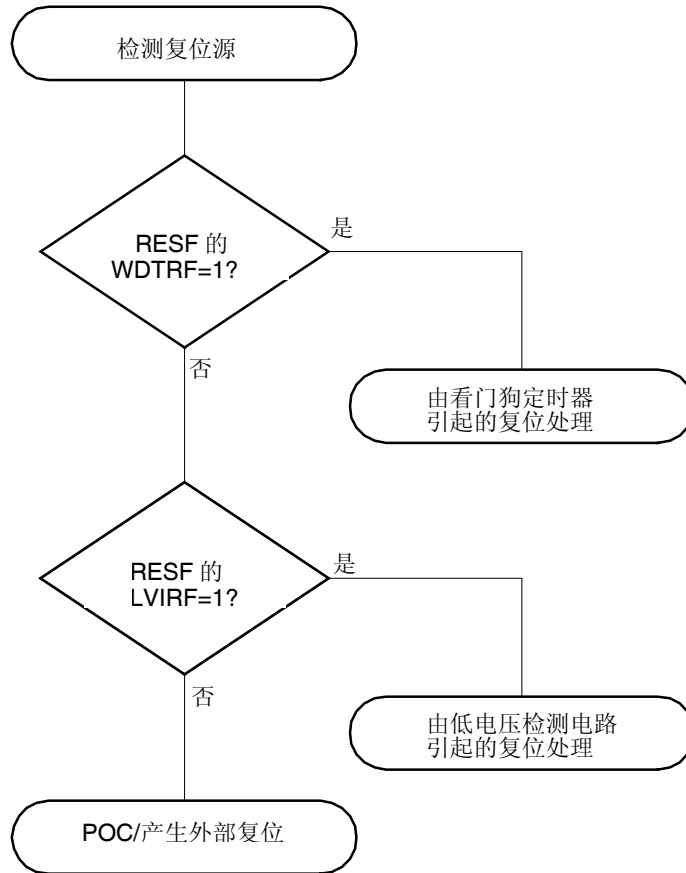
- 如果供电电压接近 POC 检测电压，其波动时间 ≤ 50 ms



- 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。

图 23-3. 复位释放后软件处理过程示例 (2/2)

• 检测复位源



第二十四章 低电压检测电路

24.1 低电压检测电路的功能

低电压检测电路 (LVI) 有以下功能。

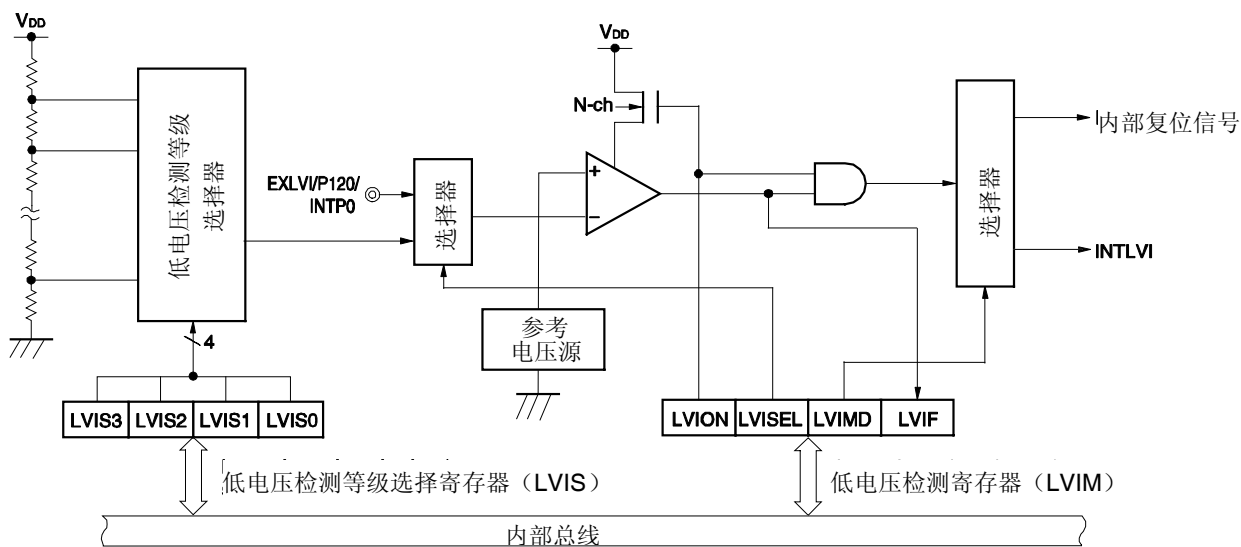
- 比较供电电压 (V_{DD}) 和检测电压 (V_{LVI})，并在 $V_{DD} < V_{LVI}$ 时产生内部中断信号或内部复位信号。通过软件改变供电电压的检测等级 (16 等级)。
- 比较外部输入引脚的输入电压 (EXLVI) 与检测电压 ($V_{EXLVI} = 1.21 \text{ V (TYP.)}$)，并在 $EXLVI < V_{EXLVI}$ 时产生内部中断信号或内部复位信号。
- 可由软件选择供电电压 (V_{DD}) 或外部输入引脚的输入电压 (EXLVI)。
- 可由软件选择中断或复位功能。
- 在 STOP 模式下可操作。

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，请参见 第二十二章 复位功能。

24.2 低电压检测电路的配置

低电压检测电路的框图如图 24-1 所示。

图 24-1. 低电压检测电路的框图



24.3 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)
- 低电压检测等级选择寄存器 (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。

复位信号的产生将 LVIM 设置为 00H。

图 24-2. 低电压检测寄存器 (LVIM) 的格式

地址: FFBEH 复位后: 00H R/W ^{※1}

符号	<7>	6	5	4	3	<2>	<1>	<0>
LVIM	LVION	0	0	0	0	LVISEL	LVIMD	LVIF
LVION ^{※2,3}	允许低电压检测操作							
0	禁止操作							
1	允许操作							
LVISEL ^{※2}	电压检测的选择							
0	检测供电电压(V _{DD})的等级							
1	检测外部输入引脚的输入电压(EXLVI)的等级							
LVIMD ^{※2}	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}) < 检测电压 (V_{LVI}), 产生中断信号 • LVISEL = 1: 当外部输入引脚的输入电压(EXLVI) < 检测电压 (V_{LVI}), 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}), 产生内部复位信号 • LVISEL = 1: 当外部输入引脚的输入电压(EXLVI) < 检测电压 (V_{LVI}), 产生内部复位信号 							
LVIF ^{※4}	低电压检测标志							
0	<ul style="list-style-type: none"> • LVISEL = 0: 供电电压 (V_{DD}) ≥ 检测电压 (V_{LVI}) 或当禁止操作时 • LVISEL = 1: 外部输入引脚的输入电压(EXLVI) ≥ 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
1. 第 0 位只读。
 2. 在产生除 LVI 复位以外的复位时, LVION、LVIMD 和 LVISEL 被清零(0)。在产生 LVI 复位时这些位不被清零。
 3. 当 LVION=1 时, LVI 电路中的比较器开始操作。从设置 LVION=1 到 LVIF 确定电压, 使用软件等待操作稳定时间(10 μ S (MAX.))。
 4. 当 LVION = 1 且 LVIMD = 0 时, 输出 LVIF 的值作为中断请求信号 INTLVI。

- 注意事项
1. 在执行以下任一过程后, 停止 LVI。
 - 当使用 8 位操作指令时: 00H → LVIM。
 - 当使用 1 位存储器操作指令时: 对 LVION 清零。
 2. 外部输入引脚的输入电压(EXLVI)必须为: EXLVI < V_{DD}。

(2) 低电压检测等级选择寄存器 (LVIS)

该寄存器用于选择低电压检测等级。

可由 1 位或 8 位存储器操作指令设置该寄存器。

复位信号将 LVIS 设置为 00H。

图 24-3. 低电压检测等级选择寄存器 (LVIS) 的格式

地址: FFBFH 复位后: 00H 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	检测等级
0	0	0	0	$V_{LV10} (4.24 V \pm 0.1 V)$
0	0	0	1	$V_{LV11} (4.09 V \pm 0.1 V)$
0	0	1	0	$V_{LV12} (3.93 V \pm 0.1 V)$
0	0	1	1	$V_{LV13} (3.78 V \pm 0.1 V)$
0	1	0	0	$V_{LV14} (3.62 V \pm 0.1 V)$
0	1	0	1	$V_{LV15} (3.47 V \pm 0.1 V)$
0	1	1	0	$V_{LV16} (3.32 V \pm 0.1 V)$
0	1	1	1	$V_{LV17} (3.16 V \pm 0.1 V)$
1	0	0	0	$V_{LV18} (3.01 V \pm 0.1 V)$
1	0	0	1	$V_{LV19} (2.85 V \pm 0.1 V)$
1	0	1	0	$V_{LV110} (2.70 V \pm 0.1 V)$
1	0	1	1	$V_{LV111} (2.55 V \pm 0.1 V)$
1	1	0	0	$V_{LV112} (2.39 V \pm 0.1 V)$
1	1	0	1	$V_{LV113} (2.24 V \pm 0.1 V)$
1	1	1	0	$V_{LV114} (2.08 V \pm 0.1 V)$
1	1	1	1	$V_{LV115} (1.93 V \pm 0.1 V)$

- 注意事项
1. 第 4 位 ~ 第 7 位必须清零。
 2. LVI 操作期间不要修改 LVIS 的值。
 3. 当检测到外部输入引脚(EXLVI)的输入电压时, 检测电压($V_{EXLVI} = 1.21 V (TYP.)$)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 PM12 设置为 FFH。

图 24-4. 端口模式寄存器 12 (PM12)的格式

地址: FF2CH	复位后: FFH R/W							
符号:	7	6	5	4	3	2	1	0
PM12	1	1	1	PM124	PM123	PM122	PM121	PM120

PM12n	P12n 引脚 I/O 模式的选择 (n = 0 ~ 4)
0	输出模式 (输出缓冲器打开)
1	输入模式 (输出缓冲器关闭)

24.4 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

- 如果 LVISEL = 0，比较供电电压(V_{DD})与检测电压(V_{LVI})，当 V_{DD} < V_{LVI} 时产生内部复位信号，而当 V_{DD} ≥ V_{LVI} 时释放内部复位。
- 如果 LVISEL = 1，比较外部输入引脚的输入电压(EXLVI)和检测电压(V_{EXLVI} = 1.21 V (TYP.))，在 EXLVI < V_{EXLVI} 时产生内部复位信号，而在 EXLVI ≥ V_{EXLVI} 时释放内部复位。

(2) 用于中断

- 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} = 1.21 V (TYP.))，当 EXLVI < V_{EXLVI} 时产生中断信号(INTLVI)。

备注 LVISEL: 低电压检测寄存器(LVIM)的第 2 位

24.4.1 用于复位

(1) 当检测供电电压(V_{DD})等级时

- 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})的等级)(默认值)。
 - <3> 使用低电压检测等级选择寄存器 (LVIS) 的第 3 位~第 0 位 (LVIS3 ~ LVIS0) 设置检测电压。
 - <4> 设置 LVIM 的第 7 位 (LVION) =1 (允许 LVI 的操作)。
 - <5> 使用软件等待操作稳定时间(10 μ s (MAX.))。
 - <6> 等待直至根据 LVIM 的第 0 位(LVIF)检测到供电电压(V_{DD}) \geq 检测电压(V_{LVI})。
 - <7> 设置 LVIM 的第 1 位 (LVIMD) =1 (当供电电压 (V_{DD}) < 检测电压 (V_{LVI}) 时产生内部复位信号)。

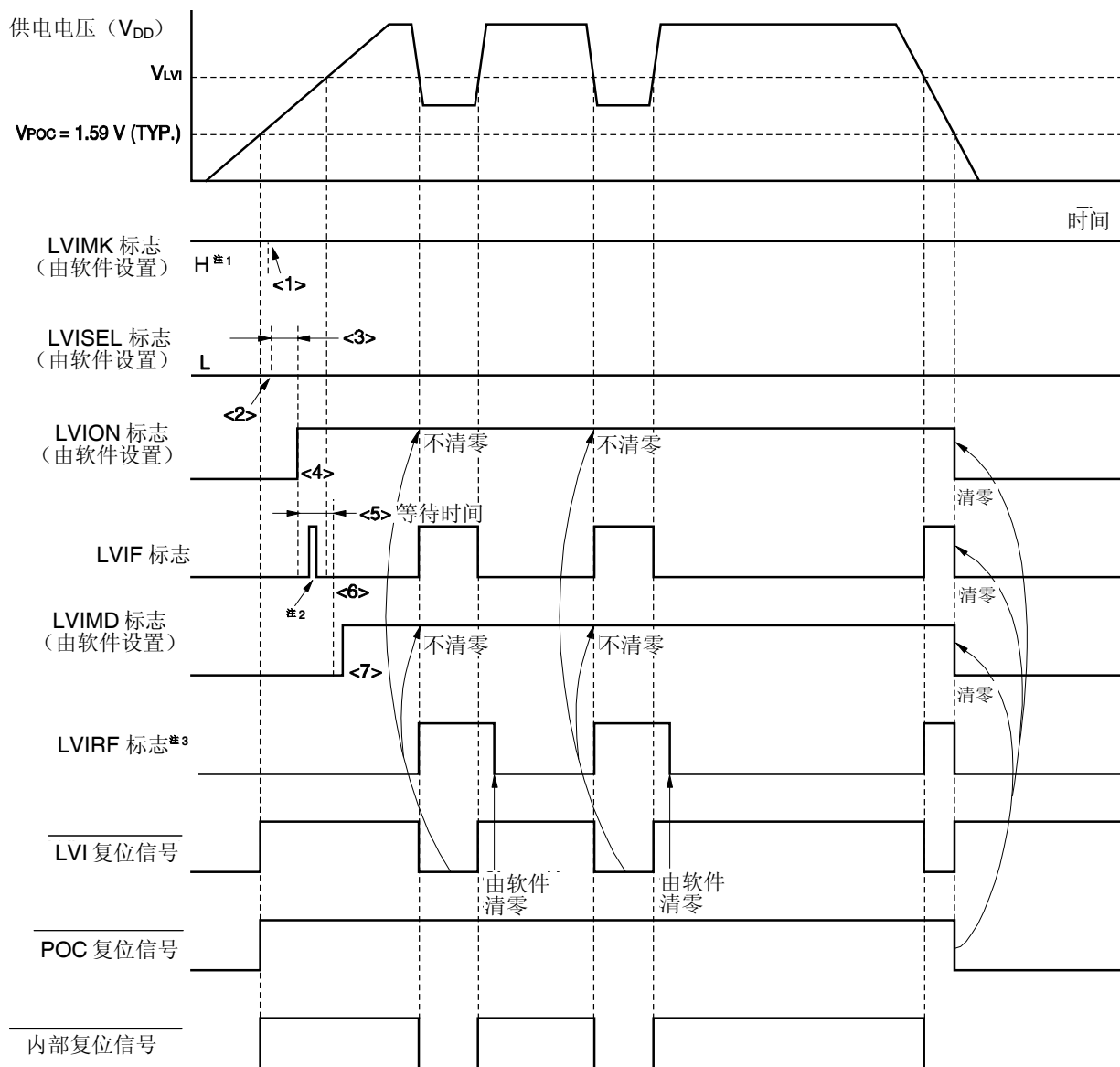
图 24-5 显示了由低电压检测电路产生的内部复位信号的时序。该时序图中的标号与上面的<1> ~ <7>对应。

- 注意事项**
1. 必须执行过程<1>。当 LVIMK = 0 时，在过程<4>之后可能立即会产生中断。
 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 位存储器操作指令时:
00H \rightarrow LVIM
 - 当使用 1 位存储器操作指令时:
先对 LVIMD 清零(0)，再对 LVION 清零(0)。

图 24-5. 低电压检测电路内部复位信号产生时序
(检测供电电压(V_{DD})等级) (1/2)

(1) 1.59 V POC 模式 (选项字节: POCMODE = 0)

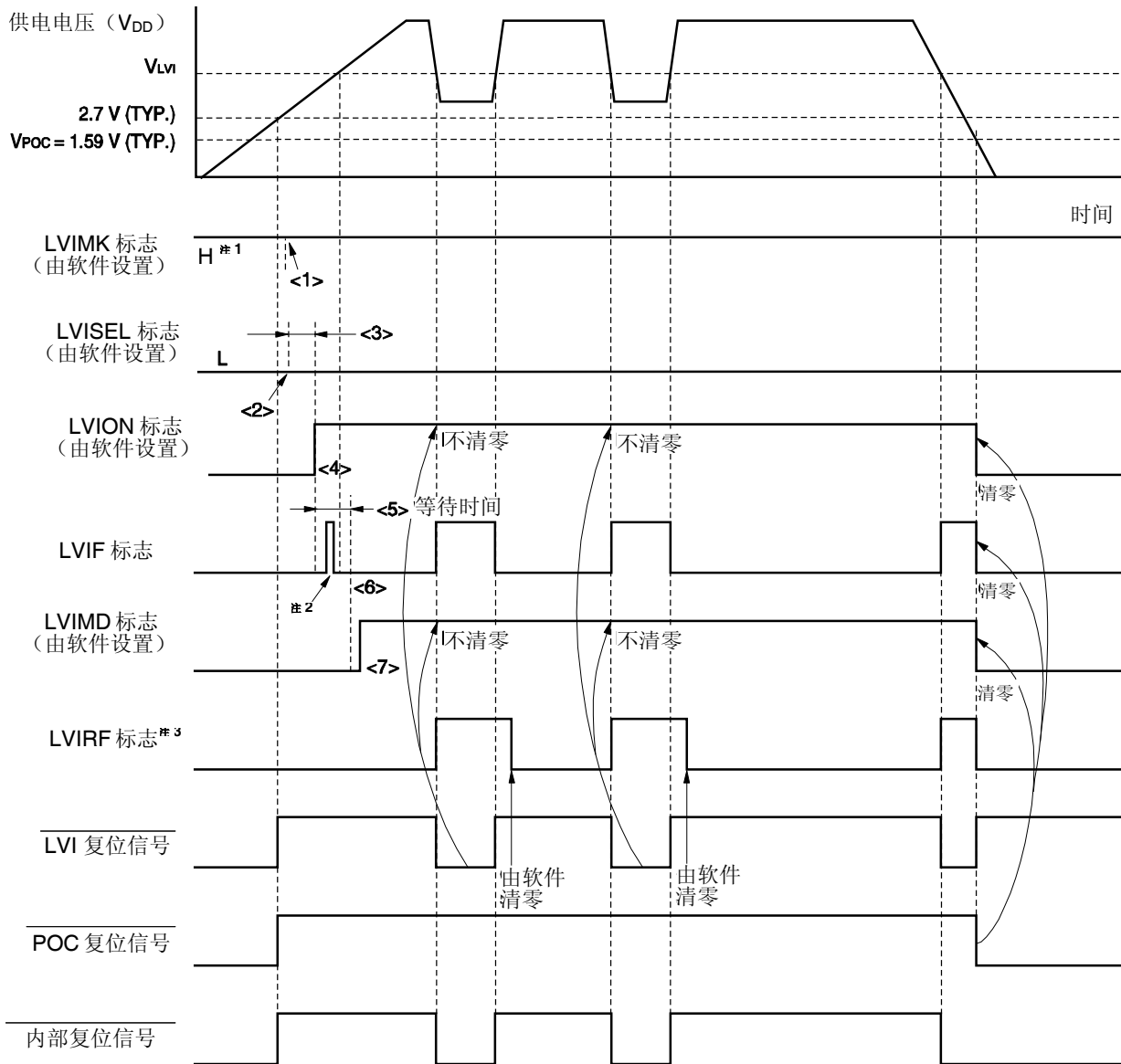


- 注
1. 由复位信号产生将 LVIMK 标志设置为“1”。
 2. 可以设置 LVIF 标志为(1)。
 3. LVIRF 是复位控制标志寄存器 (RESF) 的第 0 位。需要了解 RESF 的详细内容, 可参见第二十二章 复位功能。

备注 图 24-5 中的<1> ~ <7>与 24.4.1(1) 当检测供电电压(V_{DD})等级时中“当启动操作时”的描述<1> ~ <7>相对应。

图 24-5. 低电压检测电路内部复位信号产生时序
(检测供电电压(V_{DD})等级) (2/2)

(2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)



- 注
1. 由复位信号产生将 LVIMK 标志设置为“1”。
 2. 可以设置 LVIF 标志为(1)。
 3. LVIRF 是复位控制标志寄存器 (RESF) 的第 0 位。需要了解 RESF 的详细内容, 可参见第二十二章 复位功能。

备注 图 24-5 中的<1> ~ <7>与 24.4.1(1) 当检测供电电压(V_{DD})等级时中“当启动操作时”的描述<1> ~ <7>相对应。

(2) 当检测外部输入引脚的输入电压(EXLVI)等级时

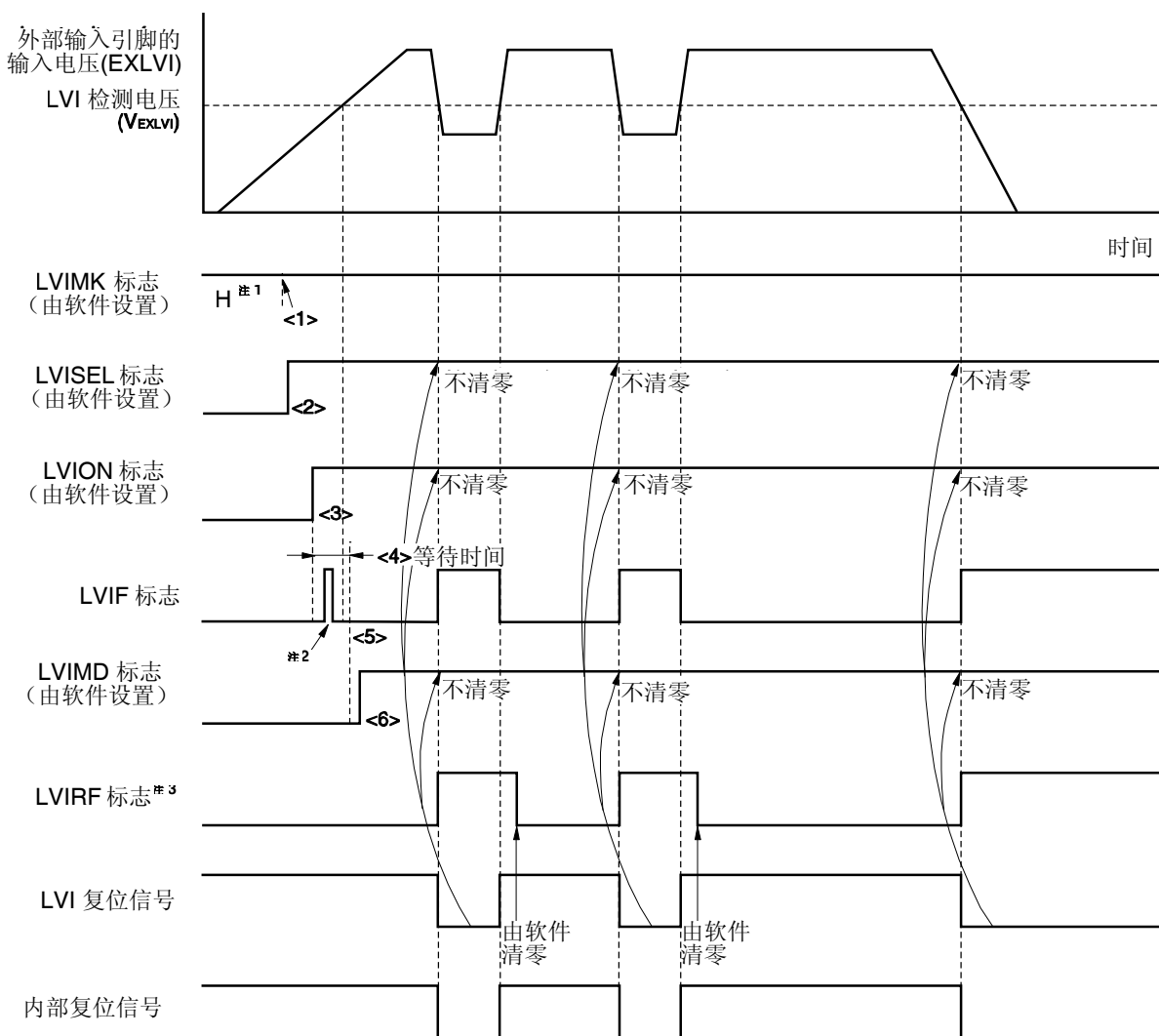
- 当启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)。
 - <2> 设置低电压检测寄存器(LVIM)的第 2 位(LVISEL)=1 (检测外部输入引脚(EXLVI) 的输入电压等级)。
 - <3> 设置 LVIM 的第 7 位 (LVION) =1 (允许 LVI 的操作)。
 - <4> 使用软件等待操作稳定时间(10 μ s (MAX.)^注)。
 - <5> 等待直至根据 LVIM 的第 0 位(LVIF)检测到: 外部输入引脚的输入电压(EXLVI) \geq 检测电压($V_{EXLVI} = 1.21$ V (TYP.))
 - <6> 设置 LVIM 的第 1 位(LVIMD)=1(当外部输入引脚的输入电压(EXLVI) < 检测电压($V_{EXLVI} = 1.21$ V (TYP.))时产生内部复位信号)。

图 24-6 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. 必须执行<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 (TYP.)), 则不会产生内部复位信号。
 3. 必须保证外部输入引脚的输入电压(EXLVI) : $EXLVI < V_{DD}$

- 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:
00H \rightarrow LVIM
 - 当使用 1 位存储器操作指令:
先对 LVIMD 清零(0), 再对 LVION 清零(0)。

图 24-6. 低电压检测电路内部复位信号产生时序
(检测外部输入引脚的输入电压(EXLVI)的等级)



- 注
1. 由复位信号的产生设置 LVIMK 标志为“1”。
 2. 可以设置 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，可参见第二十二章 复位功能。

备注 图 24-6 中的<1> ~ <6>与 24.4.1 (2) 当检测外部输入引脚的输入电压(EXLVI)等级时中“当启动操作时”的描述<1> ~ <6>相对应

24.4.2 用于中断

(1) 当检测供电电压(V_{DD})等级时

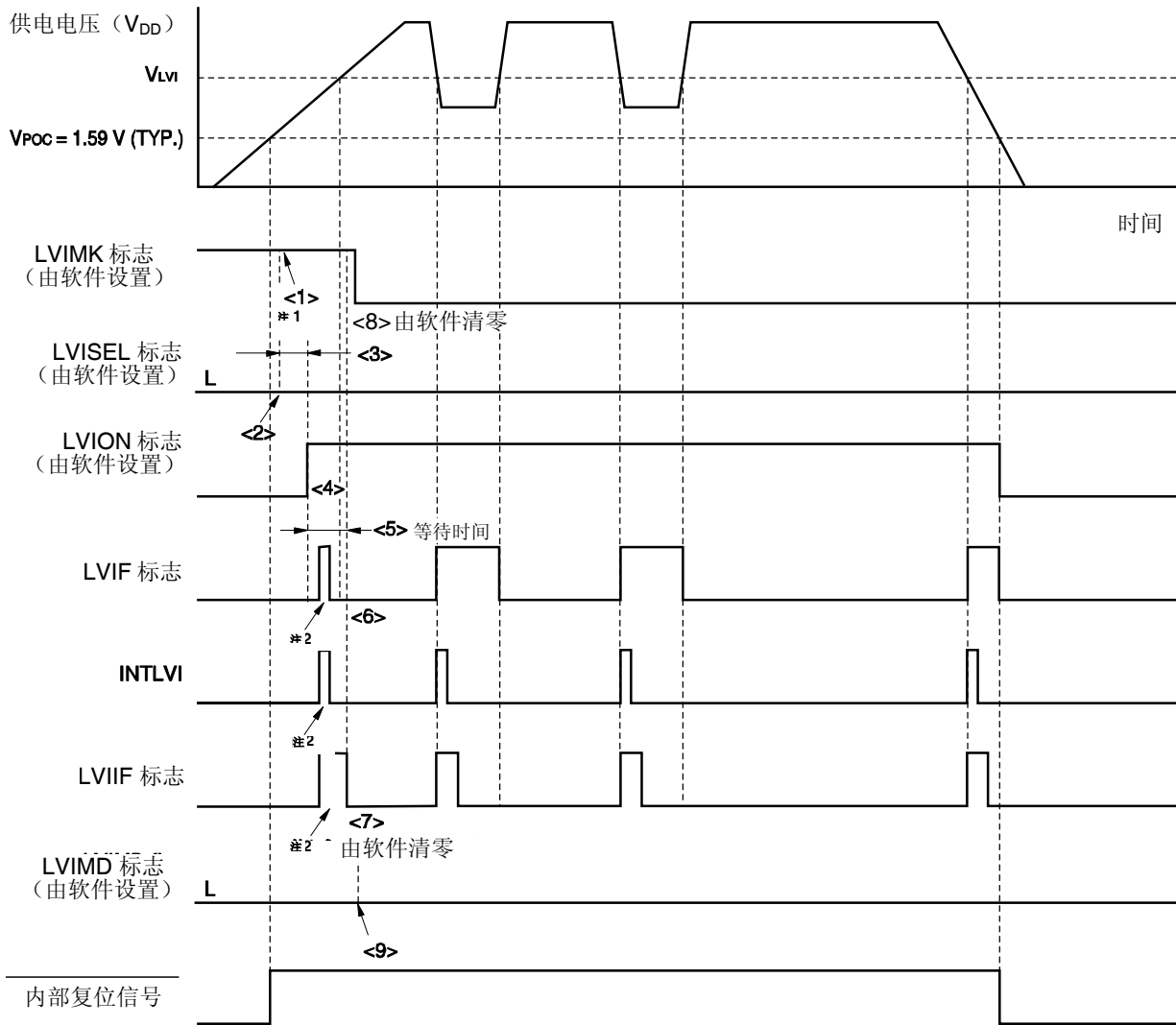
- 当启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)
 - <2> 将低电压检测寄存器(LVIM)的第 2 位(LVISEL)清零(0)(检测供电电压(V_{DD})等级)(默认值)。
 - <3> 使用低电压检测等级选择寄存器 (LVIS) 的第 3 位~第 0 位 (LVIS3 ~ LVIS0) 设置检测电压。
 - <4> 设置 LVIM 的第 7 位 (LVION) =1 (允许 LVI 的操作)。
 - <5> 使用软件等待操作稳定时间(10 μ s (MAX.))。
 - <6> 在 LVIM 的第 0 位(LVIF), 确定“供电电压(V_{DD}) \geq 检测电压(V_{LVI})”。
 - <7> 将 LVI 的中断请求标志 (LVIF) 清零(0)。
 - <8> 释放 LVI 的中断屏蔽标志 (LVIMK)。
 - <9> 将 LVIM 的第 1 位(LVIMD)清零(0)(当供电电压(V_{DD}) < 检测电压(V_{LVI})时产生中断信号)(默认值)。
 - <10> 执行 EI 指令 (当使用向量中断时)。

图 24-7 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> ~ <9>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:
00H \rightarrow LVIM
 - 当使用 1 位存储器操作指令:
对 LVION 清零 (0)

图 24-7. 低电压检测电路中断信号产生时序
(检测供电电压(V_{DD})的等级) (1/2)

(1) 1.59 V POC 模式 (选项字节: POCMODE = 0)

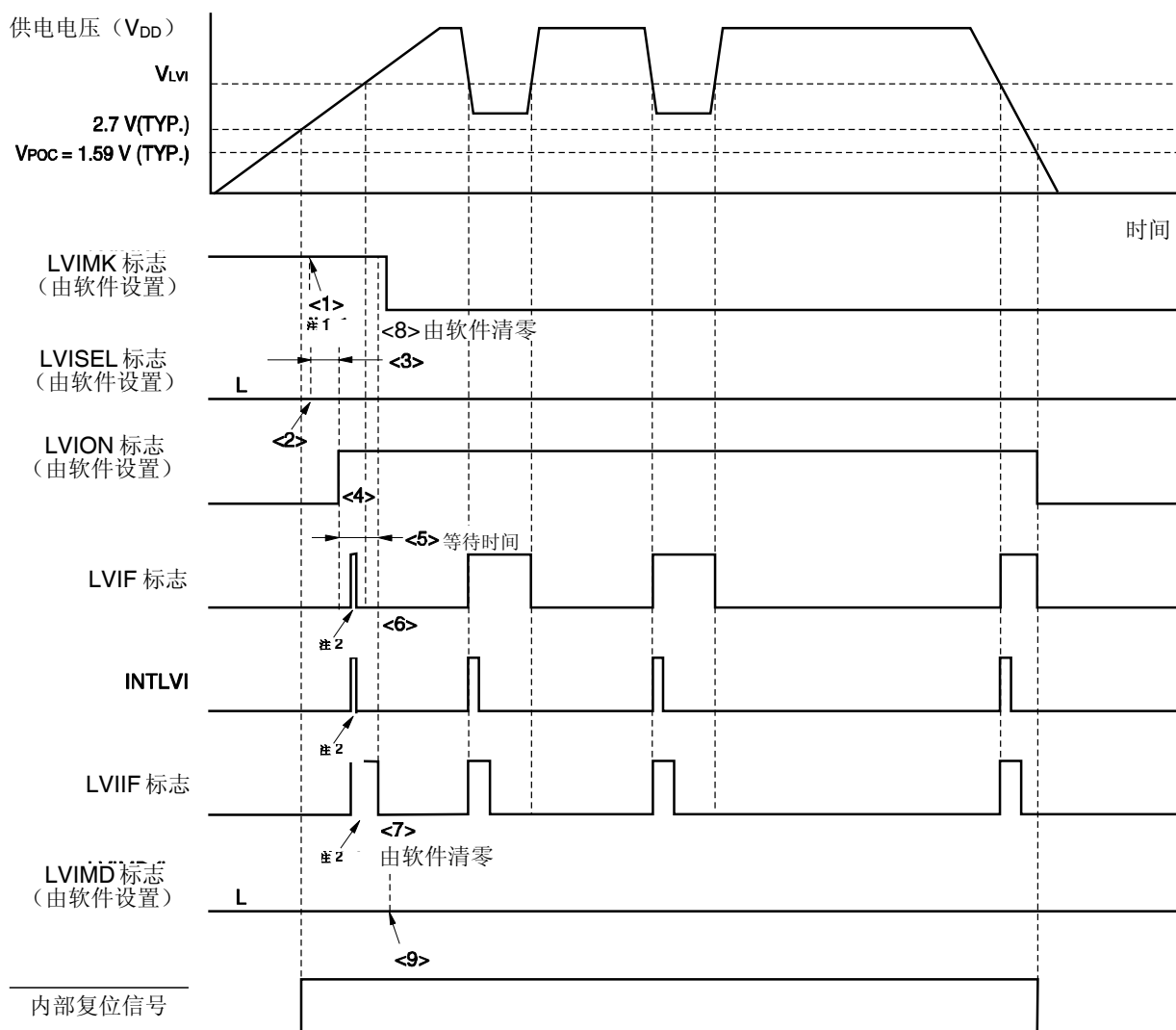


- 注
1. 可由复位信号的产生设置 LVIMK 标志为“1”。
 2. 产生中断请求信号(INTLVI)，且可以设置 LVIF 和 LVIIF 标志为(1)。

备注 图 24-7 中的<1> ~ <9>与 24.4.2 (1) 当检测供电电压(V_{DD})等级时中“当启动操作时”的描述<1> ~ <9>相对应。

图 24-7. 低电压检测电路中中断信号产生时序
(检测供电电压(V_{DD})的等级) (2/2)

(2) 2.7 V/1.59 V POC 模式 (选项字节: POCMODE = 1)



- 注
1. 可由复位信号的产生设置 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)。

备注 图 24-7 中的<1> ~ <9>与 24.4.2 (1) 当检测供电电压(V_{DD})等级时中“当启动操作时”的描述<1> ~ <9>相对应。

(2) 当检测外部输入引脚的输入电压(EXLVI)等级时

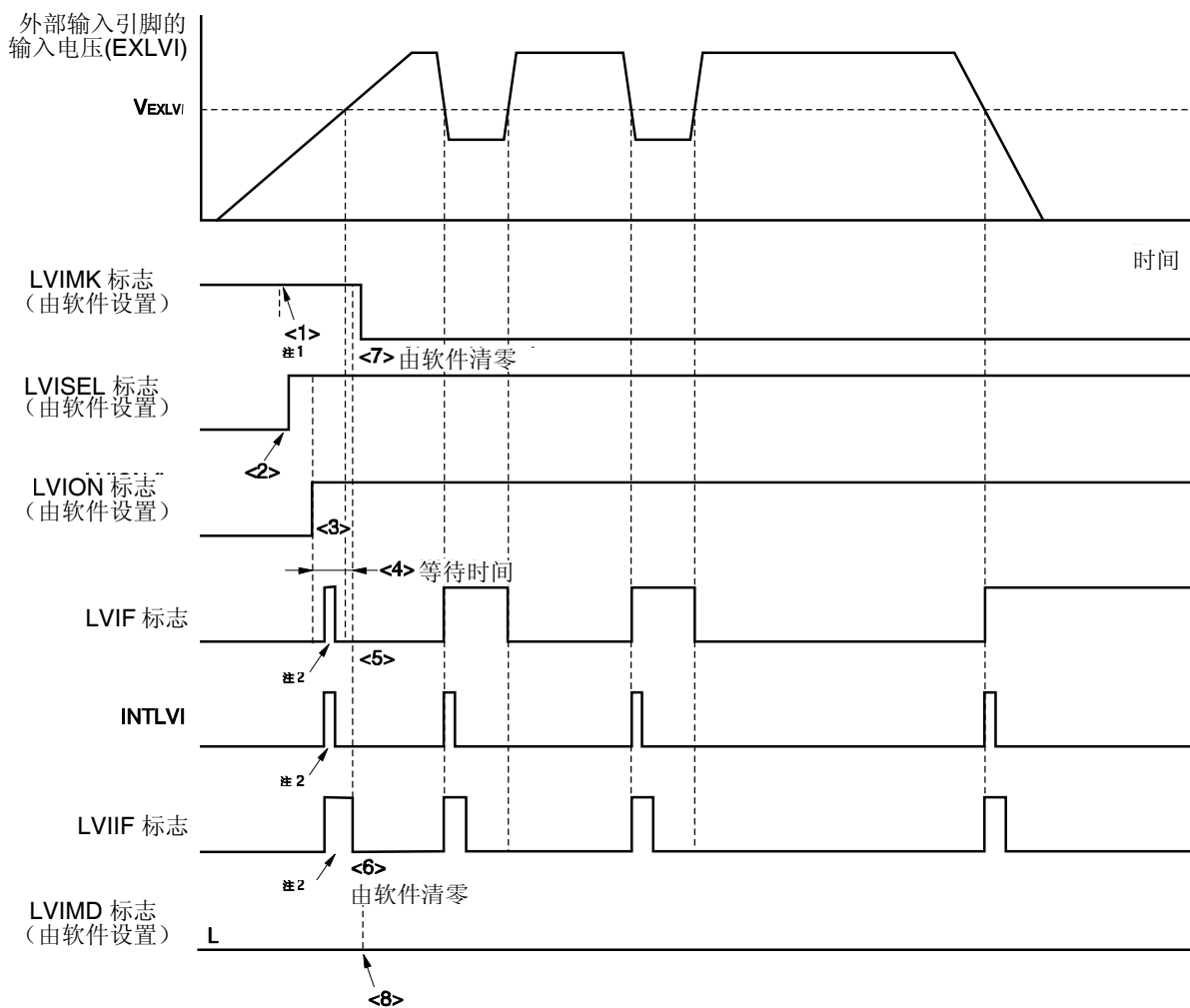
- 当启动操作时
 - <1> 屏蔽 LVI 中断 (LVIMK = 1)
 - <2> 设置低电压检测寄存器(LVIM)的第 2 位(LVISEL)=1(检测外部输入引脚的输入电压(EXLVI)等级)。
 - <3> 设置 LVIM 的第 7 位 (LVION) =1 (允许 LVI 的操作)。
 - <4> 使用软件等待操作稳定时间(10 μ s (MAX.))。
 - <5> 在 LVIM 的第 0 位(LVIF), 确定“外部输入引脚的输入电压(EXLVI) \geq 检测电压($V_{EXLVI} = 1.21 \text{ V (TYP.)}$)”。
 - <6> 将 LVI 的中断请求标志 (LVIIF) 清零(0)。
 - <7> 释放 LVI 的中断屏蔽标志 (LVIMK)。
 - <8> 将 LVIM 的第 1 位(LVIMD)清零(0)(当供电电压(V_{DD}) < 检测电压(V_{LVI})时产生中断信号)(默认值)。
 - <9> 执行 EI 指令 (当使用向量中断时)。

图 24-8 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。

注意事项 外部输入引脚的输入电压(EXLVI)必须满足以下条件: $EXLVI < V_{DD}$

- 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:
00H \rightarrow LVIM
 - 当使用 1 位存储器操作指令:
对 LVION 清零 (0)

图 24-8. 低电压检测电路中中断信号产生时序
(检测外部输入引脚的输入电压(EXLVI)等级)



- 注 1. 可由复位信号产生设置 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)。

备注 图 24-8 中的<1> ~ <8>与 24.4.2 (1) 当检测供电电压(V_{DD})等级时<1> ~ <8>中“当启动操作时”的描述<1> ~ <8>相对应。

24.5 低电压检测电路的注意事项

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。

(1) 用于复位

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（1）任意设置从复位释放到微控制器的启动所经历的时间。

(2) 用于中断

可能会频繁产生中断请求。可采用方法（2）中的(b)。

本系统按以下动作。

<方法>

(1) 用于复位

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 24-9)。

(2) 用于中断

(a) 在 LVI 中断的服务程序中通过使用低电压检测寄存器（LVIM）的第 0 位(LVIF)，检测“供电电压(V_{DD}) \geq 检测电压(V_{Lvi})”。把中断请求标志寄存器 0L(IF0L)的第 0 位(LVIIF)清零。

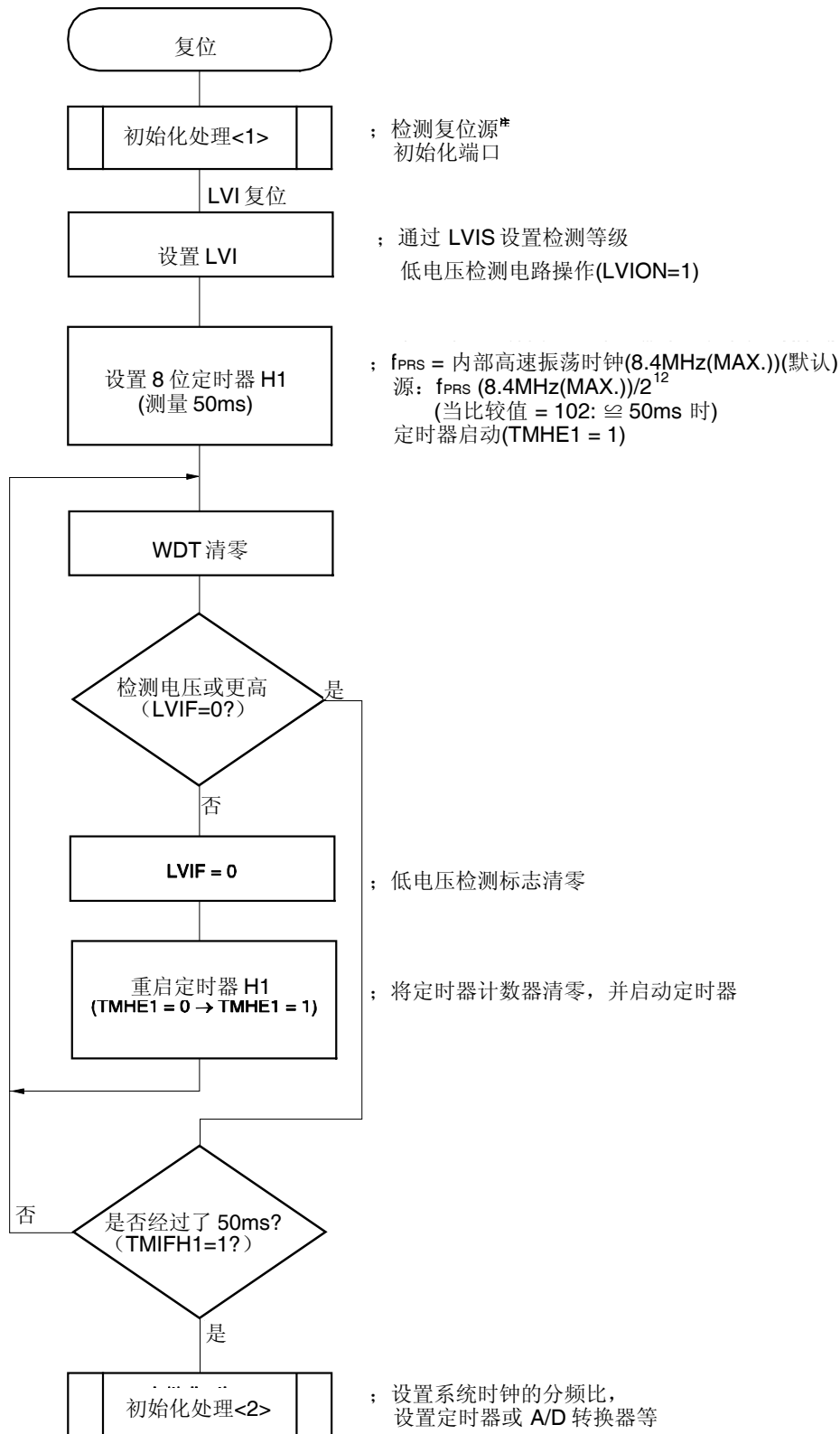
(b) 在系统中，如果供电电压（ V_{DD} ）在接近 LVI 检测电压（ V_{Lvi} ）的一段时间内产生波动，等待供电电压波动期间，使用 LVIF 标志检测“供电电压(V_{DD}) \geq 检测电压(V_{Lvi})”，并且把 LVIIF 标志清零。

备注 如果设置低电压检测寄存器(LVIM)的第 2 位(LVISEL)为“1”，则对上述方法作如下修改。

- 供电电压(V_{DD}) \rightarrow 外部输入引脚的输入电压($EXLVI$)
- 检测电压(V_{Lvi}) \rightarrow 检测电压($V_{EXLVI} = 1.21 V^{*}$)

图 24-9. 复位释放后软件处理过程示例(1/2)

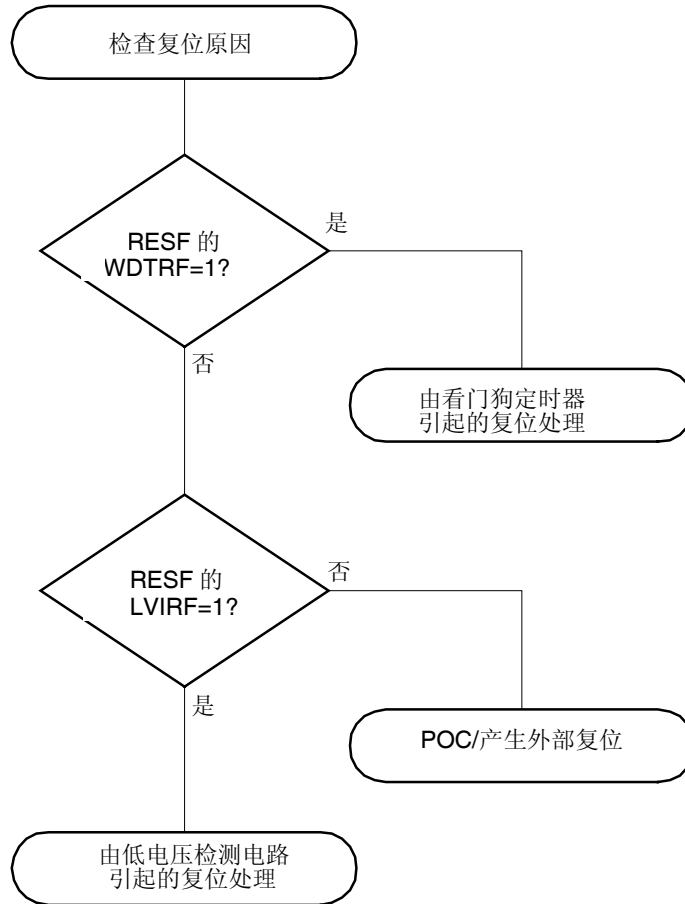
- 如果供电电压接近 LVI 检测电压，其波动时间 ≤ 50 ms



注 流程图显示在下页。

图 24-9. 复位释放后软件处理过程示例(2/2)

•检查复位源



第二十五章 选项字节

25.1 选项字节的功能

78K0/LF2 中 Flash 存储器的 0080H ~ 0084H 区域为选项字节区域。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 0080H ~ 0084H 切换到 1080H ~ 1084H。因此，事先将 0080H ~ 0084H 和 1080H ~ 1084H 设置为相同的值。

(1) 0080H/1080H

- 内部低速振荡器的操作
 - 可由软件停止
 - 不能停止
- 看门狗定时器间隔时间的设置
- 看门狗定时器计数器的操作
 - 允许计数器的操作
 - 禁止计数器的操作
- 看门狗定时器窗口打开周期的设置

(2) 0081H/1081H

- 选择 POC 模式
 - 2.7 V/1.59 V POC 模式操作期间 (POCMODE = 1)
上电时该设备处于复位状态，直至供电电压达到 2.7 V (TYP.)。当电压超过 2.7 V (TYP.)时设备从复位状态释放。此后在 2.7 V 不检测 POC，但在 1.59 V (TYP.)检测 POC。
如果上电后当速率低于 0.5 V/ms (MAX.)时供电电压上升到 1.8 V，建议使用 2.7 V/1.59 V POC 模式。
 - 1.59 V POC 模式操作期间 (POCMODE = 0)
上电时该设备处于复位状态，直至供电电压达到 1.59 V (TYP.)。当电压超过 1.59 V (TYP.)时设备从复位状态释放。此后采用与上电时相同的方法在 1.59 V (TYP.)检测 POC。

(3) 0084H/1084H

- 片上调试操作控制
 - 禁止片上调试操作
 - 一旦证明片上调试安全 ID 失效时，允许片上调试操作和擦除 Flash 存储器数据。
 - 一旦证明片上调试安全 ID 失效时，允许片上调试操作，但不擦除 Flash 存储器数据。

- 注意事项**
1. 对于没有配备片上调试功能的产品 (μ PD78F0372, 78F0373, 78F0374, 78F0375, 78F0376, 78F0382, 78F0383, 78F0384, 78F0385 和 78F0386)，必须将 00H 赋给 0084H 单元(禁止片上调试操作)。由于在引导交换时 0084H 和 1084H 要相互切换，因此也要对 1084H 设置 00H。
 2. 对于具有片上调试功能的产品 (μ PD78F0376D 和 78F0386D)，如果要使用它的片上调试功能，需要对 0084H 单元赋值 02H 或 03H。由于在引导交换时 0084H 和 1084H 要相互切换，因此也要对 1084H 设置与 0084H 相同的值。

注意事项 确保将地址 0082H 和 0083H 的内容清零 (0082H/1082H 和 0083H/1083H，当使用启动交换功能时)。

25.2 选项字节的格式

选项字节格式如下所示。

图 25-1. 选项字节的格式 (1/2)

地址: 0080H/1080H^注

7	6	5	4	3	2	1	0
0	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	LSROSC

WINDOW1	WINDOW0	看门狗定时器窗口打开周期
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/非法访问检测
0	禁止计数器操作 (复位后停止计数), 禁止非法访问检测操作
1	允许计数器操作 (复位后开始计数), 允许非法访问检测操作

WDCS2	WDCS1	WDCS0	看门狗定时器溢出时间
0	0	0	$2^{10}/f_{RL}$ (3.88 ms)
0	0	1	$2^{11}/f_{RL}$ (7.76 ms)
0	1	0	$2^{12}/f_{RL}$ (15.52 ms)
0	1	1	$2^{13}/f_{RL}$ (31.03 ms)
1	0	0	$2^{14}/f_{RL}$ (62.06 ms)
1	0	1	$2^{15}/f_{RL}$ (124.12 ms)
1	1	0	$2^{16}/f_{RL}$ (248.24 ms)
1	1	1	$2^{17}/f_{RL}$ (496.48 ms)

LSROSC	内部低速振荡器操作
0	可由软件停止 (当将 1 写入 RCM 寄存器的第 0 位(LSRSTOP)时停止操作)
1	不能停止 (即使将 1 写入 LSRSTOP 位也不停止操作)

注 由于在引导交换时 0080H 和 1080H 要相互切换, 因此要对 0080H 设置与 1080H 相同的值。

- 注意事项
1. 禁止设置: $WDCS2 = WDCS1 = WDCS0 = 0$ 且 $WINDOW1 = WINDOW0 = 0$ 。
 2. Flash 存储器自编程和模拟 EEPROM 期间看门狗定时器继续工作。处理过程中中断响应时间将被延迟。设置溢出时间和窗口大小时应考虑到这一延迟情况。
 3. 如果 $LSROSC = 0$ (可由软件停止振荡), 则无论内部振荡模式寄存器(RCM)的第 0 位(LSRSTOP)为何值, 在 HALT 和 STOP 模式下对看门狗定时器不提供计数时钟。
当 8 位定时器 H1 使用内部低速振荡时钟时, 即使在 HALT/STOP 模式下也对 8 位定时器 H1 提供计数时钟。
 4. 必须对第 7 位清零(0)。

- 备注
1. f_{RL} : 内部低速振荡时钟频率
 2. (): $f_{RL} = 264$ kHz (最大)

图 25-1. 选项字节的格式 (2/2)

地址: 0081H/1081H^{※1,2}

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

POCMODE	POC 模式选择
0	1.59 V POC 模式 (默认)
1	2.7 V/1.59 V POC 模式

- 注
1. 使用专用 Flash 存储器编程器只能对 POCMODE 进行写操作。在自编程期间不能设置 POCMODE，或者自编程引导交换(此时，设置 1.59 V POC 模式(默认))操作时也不能设置。但由于引导交换时 1081H 单元的值被复制到 0081H 单元，建议在使用引导交换功能时为 1081H 单元设置与 0081H 相同的值。
 2. 要改变 POC 模式的设置，可在对 Flash 存储器进行一次性擦除(片擦除)后将该值再次写入 0081H 中。在对 Flash 存储器指定 block 擦除后不能修改设置。

注意事项 必须将第 7 位 ~ 第 1 位清零。

地址: 0082H/1082H, 0083H/1083H[※]

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

- 注 由于 0082H 与 0083H 单元为保留区域，因此必须对这两个存储单元赋值 00H。由于在引导交换时 0082H 与 0083H 要和 1082H 与 1083H 相互切换，因此也要对 1082 与 1083H 设置 00H。

地址: 0084H/1084H^{※1,2}

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCDEN1	OCDEN0

OCDEN1	OCDEN0	片上调试操作控制
0	0	禁止操作
0	1	禁止设置
1	0	允许操作。在确定片上调试安全 ID 失效时不要擦除 Flash 存储器数据。
1	1	允许操作。在确定片上调试安全 ID 失效时可以擦除 Flash 存储器数据。

- 注
1. 对于没有配备片上调试功能的产品(μ PD78F0372, 78F0373, 78F0374, 78F0375, 78F0376, 78F0382, 78F0383, 78F0384, 78F0385 和 78F0386)，必须将 00H 赋给 0084H 单元(禁止片上调试操作)。由于在引导交换时 0084H 和 1084H 要相互切换，因此也要对 1084H 设置 00H。
 2. 对于具有片上调试功能的产品(μ PD78F0376D 和 78F0386D)，如果要使用它的片上调试功能，需要对 0084H 单元赋值 02H 或 03H。由于在引导交换时 0084H 和 1084H 要相互切换，因此也要对 1084H 设置与 0084H 相同的值。

备注 如需了解片上调试安全 ID 的详细信息，参见第二十七章 片上调试功能 (仅适用于 μ PD78F0376D 和 78F0386D)。

以下是设置选项字节的软件描述示例。

OPT	CSEG	AT 0080H	
OPTION:	DB	30H	; 允许看门狗定时器的操作 (对非法访问检测的操作), ; 看门狗定时器窗口打开周期: 50%, ; 看门狗定时器的溢出时间: $2^{10}/f_{RL}$, ; 可由软件停止内部低速振荡器
	DB	00H	; 1.59 V POC 模式
	DB	00H	; 保留区域
	DB	00H	; 保留区域
	DB	00H	; 禁止片上调试

备注 复位处理期间参考选项字节。如需了解复位处理的时序，可参见**第二十二章 复位功能**。

第二十六章 Flash 存储器

78K0/LF2 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。

26.1 内部存储器容量切换寄存器

可以使用内部存储器容量切换寄存器（IMS）选择内存容量。

可使用 8 位存储器操作指令设置 IMS。

复位信号的产生将 IMS 设置为 CFH。

注意事项 复位释放后必须按照表 26-1 对每个产品进行设置。

图 26-1. 内部存储器容量切换寄存器（IMS）的格式

地址： FFF0H 复位后： CFH R/W

符号	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0
	RAM2	RAM1	RAM0	内部高速 RAM 容量选择				
	1	1	0	1024 字节				
	其它			禁止设置				
	ROM3	ROM2	ROM1	ROM0	内部 ROM 容量选择			
	0	1	1	0	24 KB			
	1	0	0	0	32 KB			
	1	1	0	0	48 KB			
	1	1	1	1	60 KB			
	其它				禁止设置			

注意事项 如果要设置存储器容量，应先设置 IMS 再设置 IXS。设置存储器容量时应使内部存储器 ROM 和内部扩展 RAM 区域不要产生重叠。

表 26-1. 内部存储器容量切换寄存器的设置

Flash 存储器版本(78K0/LF2)	IMS 设置
μPD78F0372, 78F0382	C6H
μPD78F0373, 78F0383	C8H
μPD78F0374, 78F0384	CCH
μPD78F0375, 78F0385	CFH
μPD78F0376, 78F0386, 78F0376D, 78F0386D ^{※1}	CCH ^{※2}

- <R> 注
- 具有片上调试功能的产品，其 ROM 和 RAM 容量根据调试目标产品来进行调试。根据调试目标产品设置 IMS 和 IXS。
 - μPD78F0376, 78F0376D, 78F0386 和 78F0386D 分别有 96 KB 的内部 ROM。但是，由于使用存储器 bank，这些设备中的 IMS 设置值与 48 KB 产品的相同。如需了解存储器 bank 的设置方法，可参见 3.6 存储器 Bank 选择寄存器 (BANK) (仅限于μPD78F0376, 78F0376D, 78F0386 和 78F0386D)。

26.2 内部扩展 RAM 容量切换寄存器

使用内部扩展 RAM 容量切换寄存器(IXS)可以选择内部扩展 RAM 的容量。

可使用 8 位存储器操作指令设置 IXS。

复位信号的产生将 IXS 设置为 0CH。

注意事项 在复位释放后必须如表 26-2 所示设置每种产品的值。

图 26-2 内部扩展 RAM 容量切换寄存器 (IXS) 的格式

地址: FFF4H 复位后: 0CH R/W

符号	7	6	5	4	3	2	1	0
IXS	0	0	0	IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0

IXRAM4	IXRAM3	IXRAM2	IXRAM1	IXRAM0	内部扩展 RAM 容量选择
0	1	1	0	0	0 字节
0	1	0	1	0	1024 字节
0	1	0	0	0	2048 字节
0	0	1	0	0	4096 字节
其它					禁止设置

注意事项 如果要设置存储器容量，应先设置 IMS，再设置 IXS。设置存储器容量时应使内部 ROM 和内部扩展 RAM 区域不要产生重叠。

表 26-2 内部扩展 RAM 容量切换寄存器 (IXS) 的设置

Flash 存储器版本 (78K0/LF2)	IXS 设置
μPD78F0372, 78F0382	0CH
μPD78F0373, 78F0383	
μPD78F0374, 78F0384	0AH
μPD78F0375, 78F0385	08H
μPD78F0376, 78F0386, 78F0376D, 78F0386D	04H

26.3 用 Flash 编程器写入数据

可使用专用 Flash 编程器将数据写入 Flash 存储器，可以 On-board 写入，也可以 Off-board 写入。

(1) On-board 编程

当目标系统上安装了 78K0/LF2 系列产品后，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

在将 78K0/LF2 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.的产品。

表 26-3. 78K0/LF2 系列产品与专用 Flash 编程器之间的写操作(μ PD78F037x)

专用 Flash 编程器的引脚配置			使用 CSI10		使用 UART6	
信号名称	I/O	引脚功能	引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	接收信号	SO10/P12	57	TxD6/P13	56
SO/TxD	输出	发送信号	SI10/RxD0/P11	58	RxD6/P14	55
SCK	输出	传输时钟	SCK10/TxD0/P10	59	–	–
CLK	输出	78K0/LF2 的时钟	$\text{--}_{\text{注}1}$	–	EXCLK/X2/P122 $\text{注}2$	7
/RESET	输出	复位信号	RESET	3	RESET	3
FLMD0	输出	模式信号	FLMD0	6	FLMD0	6
V _{DD}	I/O	V _{DD} 电压产生/ 电压监测	V _{DD}	11	V _{DD}	11
			LV _{DD}	48	LV _{DD}	48
			AV _{REF}	60	AV _{REF}	60
GND	–	地	V _{SS}	10	V _{SS}	10
			LV _{SS}	47	LV _{SS}	47
			AV _{SS}	61	AV _{SS}	61

- 注
1. 在使用 CSI10 时，只能使用内部高速振荡时钟 (f_{RH})。
 2. 在使用 UART6 时，只能使用 X1 时钟(f_X)或外部主系统时钟(f_{EXCLK})。若要使用 Flash 编程器的时钟，连接编程器的 CLK 到 EXCLK。

表 26-4. 78K0/LF2 系列产品与专用 Flash 编程器之间的写操作(μ PD78F038x)

专用 Flash 编程器的引脚配置			使用 CSI10		使用 UART6	
信号名称	I/O	引脚功能	引脚名称	引脚编号	引脚名称	引脚编号
SI/RxD	输入	接收信号	SO10/P12	61	TxD6/P13	62
SO/TxD	输出	发送信号	SI10/RxD0/P11	60	RxD6/P14	63
SCK	输出	传输时钟	SCK10/TxD0/P10	59	–	–
CLK	输出	78K0/LF2 的时钟	$\text{--}_{\text{注}1}$	–	EXCLK/X2/P122 $\text{注}2$	4
/RESET	输出	复位信号	RESET	80	RESET	80
FLMD0	输出	模式信号	FLMD0	3	FLMD0	3
V _{DD}	I/O	V _{DD} 电压产生/ 电压监测	V _{DD}	8	V _{DD}	8
			LV _{DD}	55	LV _{DD}	55
GND	–		V _{SS}	7	V _{SS}	7
			LV _{SS}	54	LV _{SS}	54

- 注
1. 在使用 CSI10 时，只能使用内部高速振荡时钟 (f_{RH})。
 2. 在使用 UART6 时，只能使用 X1 时钟(f_X)或外部主系统时钟(f_{EXCLK})。若要使用 Flash 编程器的时钟，连接编程器的 CLK 到 EXCLK。

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 26-3. 3 线串行 I/O (CSI10) 模式下使用适配器对 Flash 存储器进行写操作示例(μ PD78F037x)

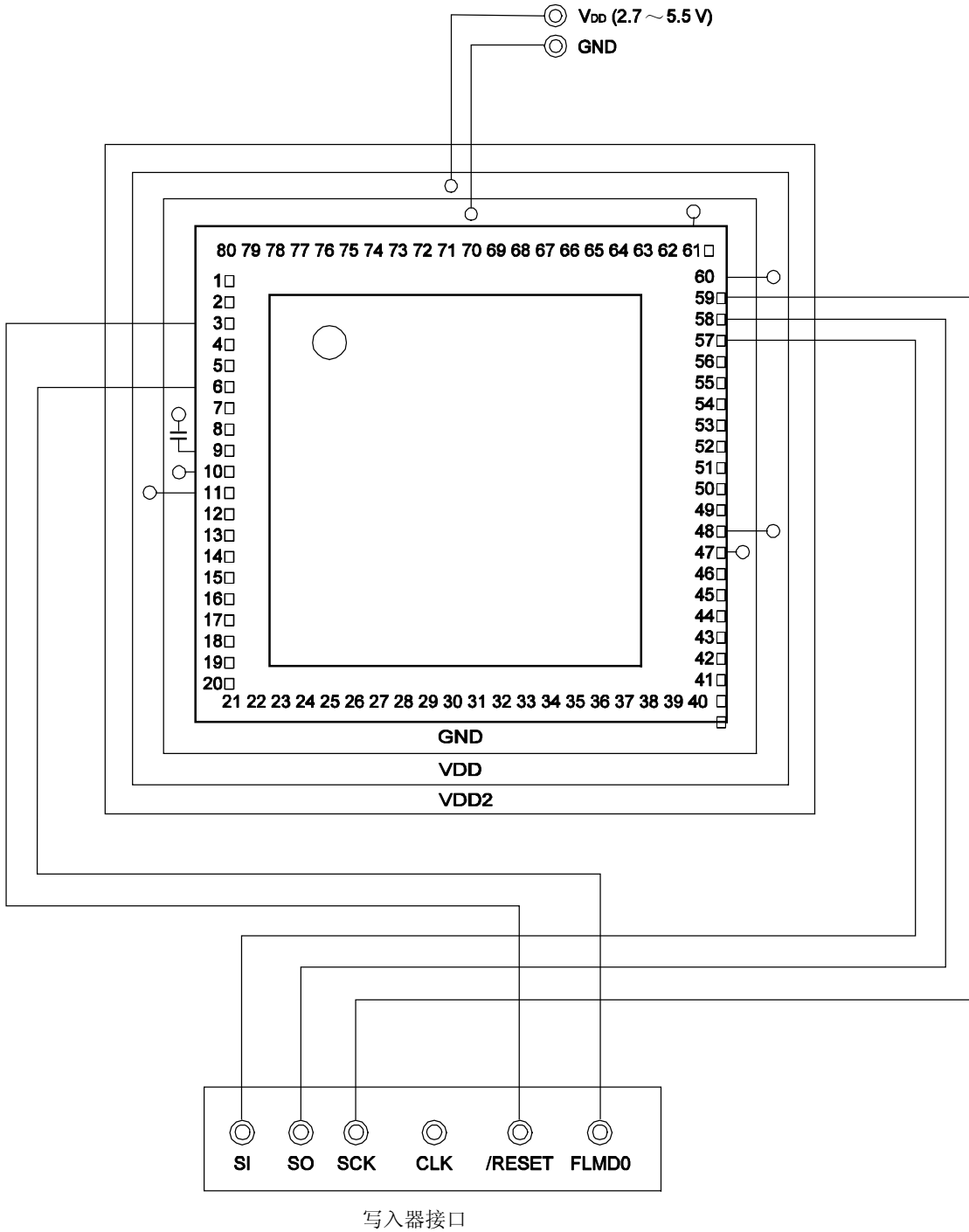


图 26-4. UART (UART6) 模式下使用适配器对 Flash 存储器进行写操作示例(μ PD78F037x)

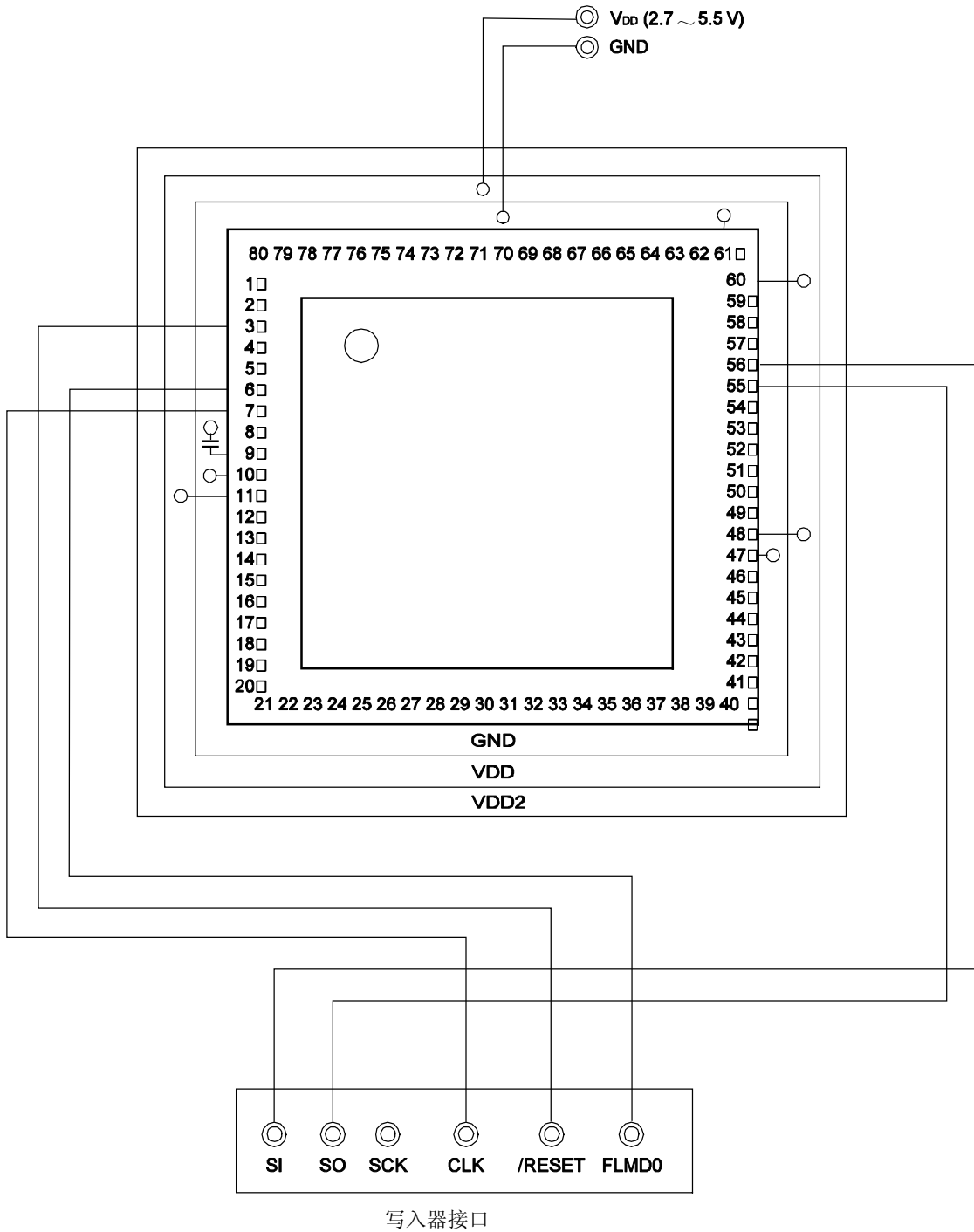


图 26-5. 3 线串行 I/O (CSI10) 模式下使用适配器对 Flash 存储器进行写操作示例(μ PD78F038x)

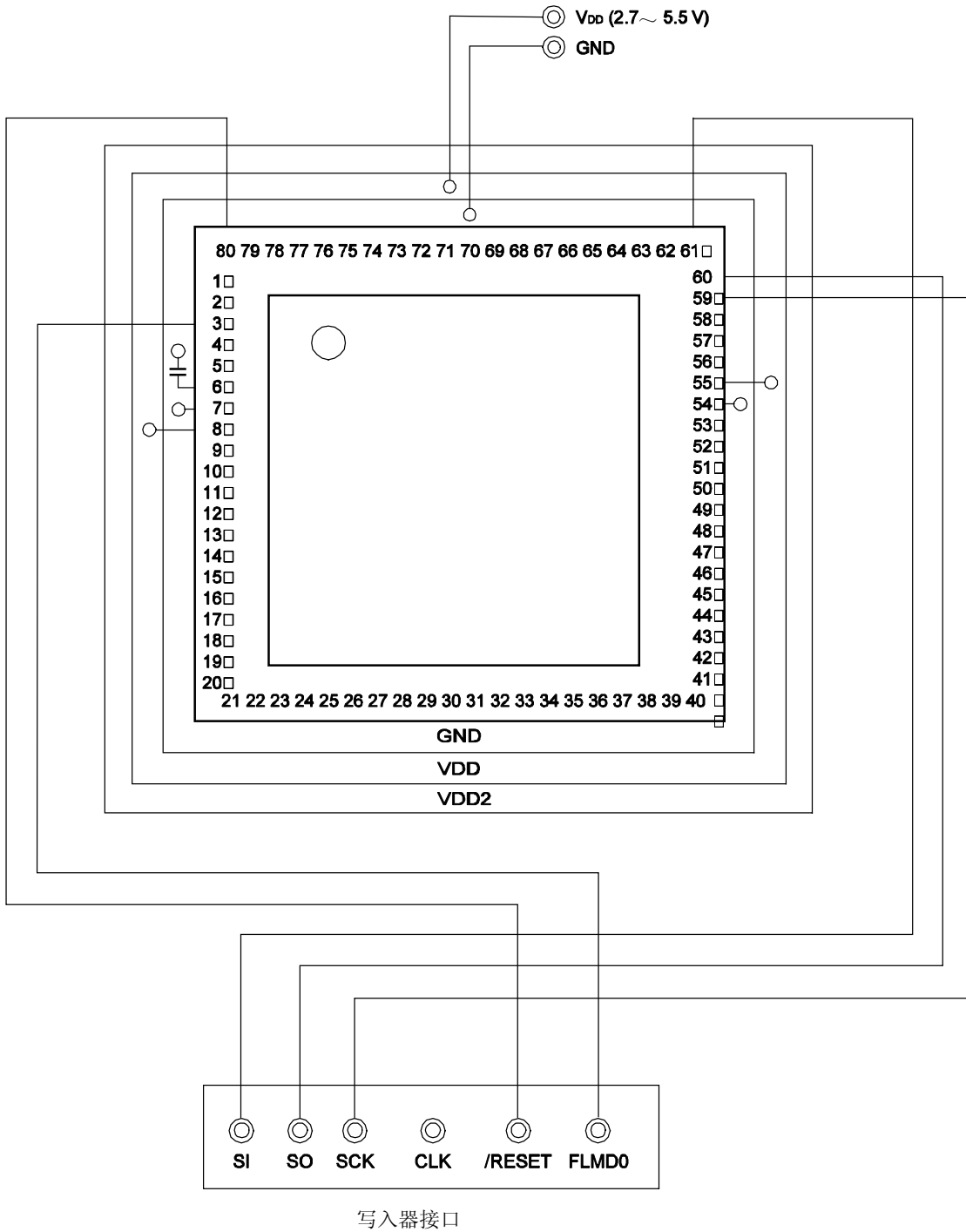
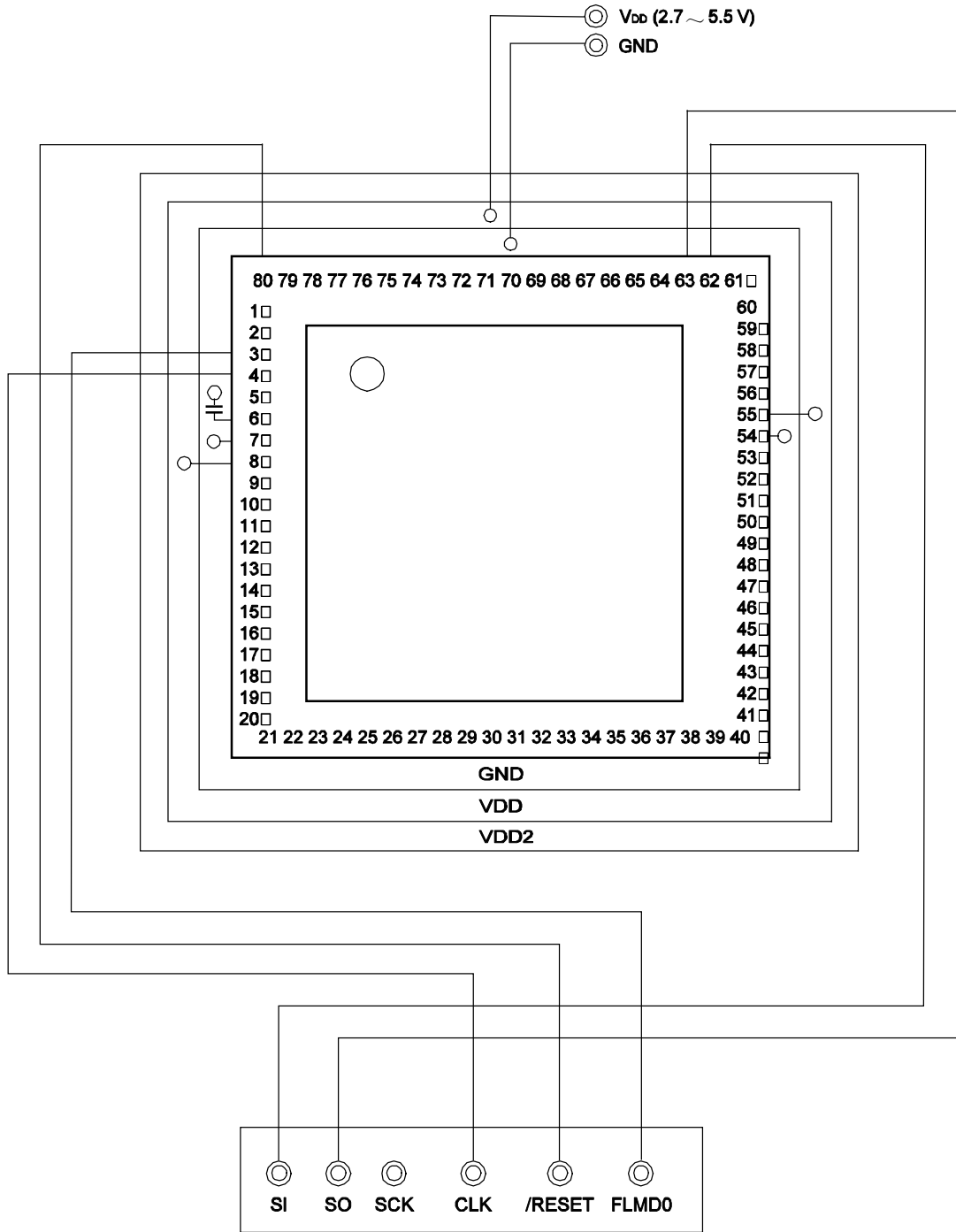


图 26-6. UART (UART6) 模式下使用适配器对 Flash 存储器进行写操作示例(μ PD78F038x)



写入器接口

26.4 编程环境

以下是 78K0/LF2 Flash 存储器所需的编程环境。

图 26-7. Flash 存储器编程环境



需要有一个控制专用 Flash 编程器的主机。

专用 Flash 编程器与 78K0/LF2 的接口使用 CSI10 或 UART6 进行写和擦除等操作。Flash 存储器离线写操作时必须使用一个专用程序适配器（FA 系列）。

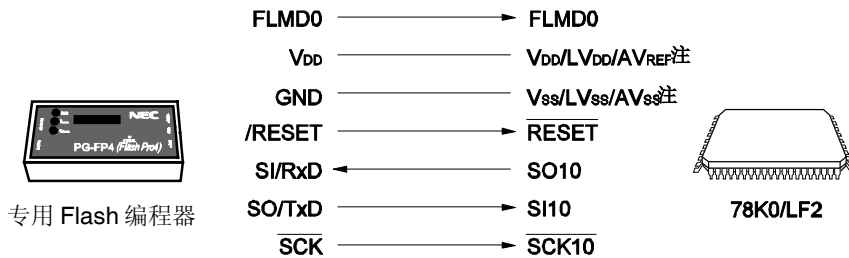
26.5 通信模式

通过 78K0/LF2 的 CSI10 或 UART6，在专用 Flash 编程器与 78K0/LF2 之间建立串行通信。

(1) CSI10

传输速率： 2.4 kHz ~ 2.5 MHz

图 26-8. 专用 Flash 编程器的通信 (CSI10)

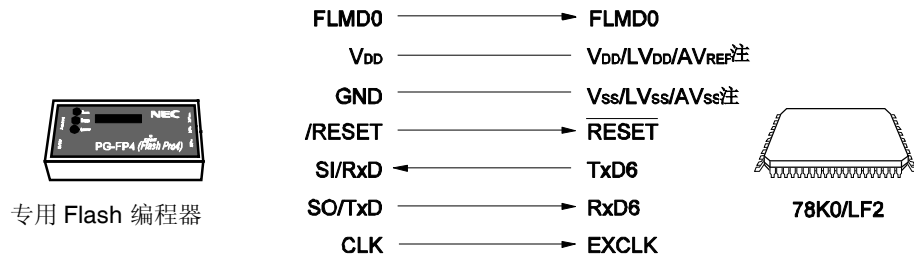


注 仅限于 μ PD78F037x。

(2) UART6

传输速率: 115200 bps

图 26-9. 专用 Flash 编程器的通信(UART6)

注 仅限于 μ PD78F037x。

如果将 FlashPro4 作为 78K0/LF2 专用 flash 编程器，它产生以下信号。详细情况，请参看 FlashPro4 的用户手册。

表 26-5. 引脚连接

FlashPro4			78K0/LF2	连接	
信号名称	I/O	引脚功能	引脚名称	CSI10	UART6
FLMD0	输出	模式信号	FLMD0	□	□
V _{DD}	I/O	V _{DD} 电压/电源监测	V _{DD} , LV _{DD} , AV _{REF} ^{注1}	□	□
GND	-	地	V _{SS} , LV _{SS} , AV _{SS} ^{注1}	□	□
CLK	输出	78K0/LF2 时钟输出	EXCLK	x ^{注2}	○ ^{注3}
/RESET	输出	复位信号	RESET	□	□
SI/RxD	输入	接收信号	SO10/TxD6	□	□
SO/TxD	输出	发送信号	SI10/RxD6	□	□
SCK	输出	传输时钟	SCK10	□	x

注 1. 仅限于 μ PD78F037x。2. 在使用 CSI10 时，只能使用内部高速振荡时钟 (f_{RH})。3. 在使用 UART6 时，只能使用 X1 时钟 (f_X) 或外部主系统时钟 (f_{EXCLK})。若要使用 Flash 编程器的时钟，连接编程器的 CLK 到 EXCLK。

备注 □: 确保连接引脚。

○: 如果在目标板上产生信号，则此引脚不需要连接。

x: 此引脚不需要连接。

26.6 在线方式的引脚连接

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。

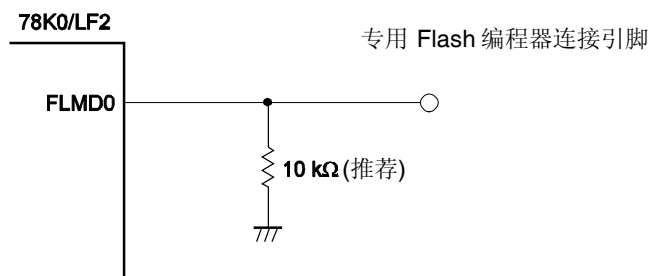
当设置 Flash 存储器编程模式时，那些不用于 Flash 存储器编程的引脚状态与复位后的状态相同。因此如果外部设备不能立即识别复位后的状态，则必须采用如下方式连接引脚。

26.6.1 FLMD0 引脚

在普通操作模式下，FLMD0 引脚的输入电压为 0V。在 Flash 存储器编程模式中，将 V_{DD} 写电压提供给 FLMD0 引脚。FLMD0 引脚的连接示意图如下所示。

<R>

图 26-10. FLMD0 引脚连接示例



26.6.2 串行接口引脚

串行接口使用的引脚如下所示。

表 26-6. 串行接口使用的引脚

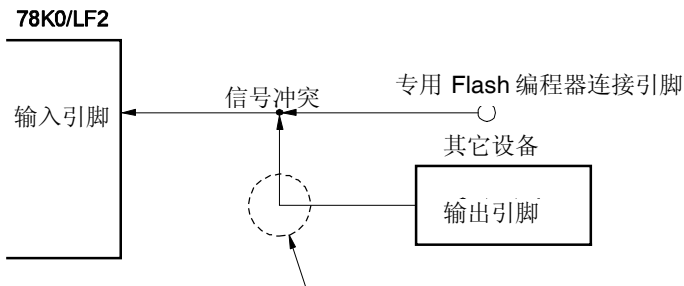
串行接口	使用的引脚
CSI10	SO10, SI10, SCK10
UART6	TxD6, RxD6

在将专用 Flash 编程器和串行接口 (已与板上其它设备连接) 的引脚相连时，必须特别注意：信号之间不能冲突，并且另一个设备不能出现误操作。

(1) 信号冲突

如果专用 Flash 编程器 (输出端) 与串行接口 (已连接到另一个设备的输出端) 的一个输入引脚相连, 这时会产生信号冲突。为了避免这种情况, 应隔离与另一个设备的连接, 或者使另一个设备处于高阻抗状态。

图 26-11. 信号冲突 (串行接口的输入引脚)

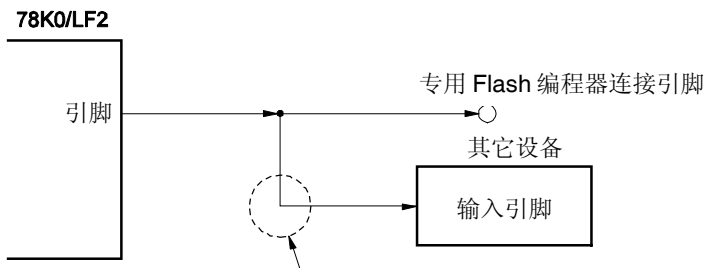


在 Flash 存储器编程模式下, 设备的输出信号与专用 Flash 编程器发送的信号发生冲突。因此必须隔离该设备的信号。

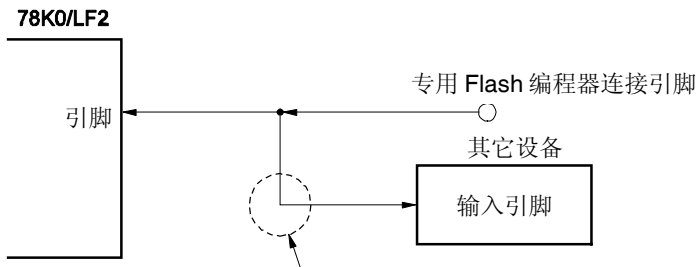
(2) 其它设备的故障

如果专用 Flash 编程器的输出或输入端与串行接口 (已连接到另一个设备的输入端) 的一个输入或输出引脚相连, 则信号可能会输出到另一个设备, 从而引起该设备的故障。为了避免这种情况, 应隔离与该设备的连接。

图 26-12. 其它设备的故障



在 Flash 存储器编程模式下, 如果 78K0/LF2 的输出信号影响到另外一个设备, 那么隔离另一设备的信号



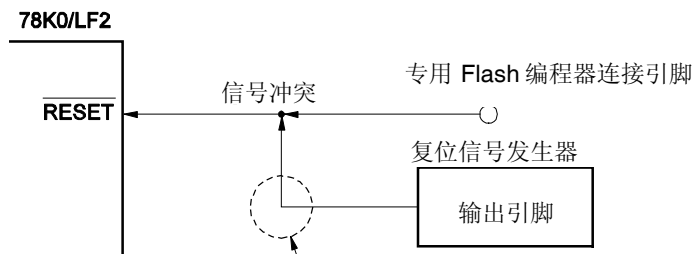
在 Flash 存储器编程模式下, 如果专用 Flash 编程器的输出信号影响到另外一个设备, 那么隔离另一设备的信号

26.6.3 RESET 引脚

如果将专用 Flash 编程器的复位信号连接到 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 26-13. 信号冲突(RESET 引脚)



在 Flash 存储器编程模式中，复位信号发生器输出的信号与专用 Flash 存储器编程器的输出信号发生冲突。因此必须隔离复位信号发生器的信号。

26.6.4 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 V_{DD} 或 V_{SS}。

26.6.5 REGC 引脚

<R> 以正常操作相同的方式，通过一个电容 (0.47 ~ 1 μF: 推荐) 连接 REGC 引脚到 GND。

26.6.6 其它信号引脚

在正常操作模式下使用 on-board 时钟时，以相同的状态连接 X1 和 X2。

为了从编程器输入操作时钟，要连接编程器的 CLK 到 EXCLK。

- 注意事项**
1. 在使用 CSI10 时，只能使用内部高速振荡时钟(f_{RH})。
 2. 在使用 UART6 时，只能使用 X1 时钟(f_X)或外部主系统时钟(f_{EXCLK})。

26.6.7 供电电压

要使用 Flash 编程器输出的供电电压，则将 V_{DD} 引脚与 Flash 编程器的 V_{DD} 相连，将 V_{SS} 引脚与 Flash 编程器的 GND 相连。

然而，如果要使用 Flash 编程器的电压监视功能，必须将 V_{DD} 和 V_{SS} 引脚分别与 Flash 编程器的 V_{DD} 和 GND 相连。

要使用板上供电电压，要按正常操作模式连接。

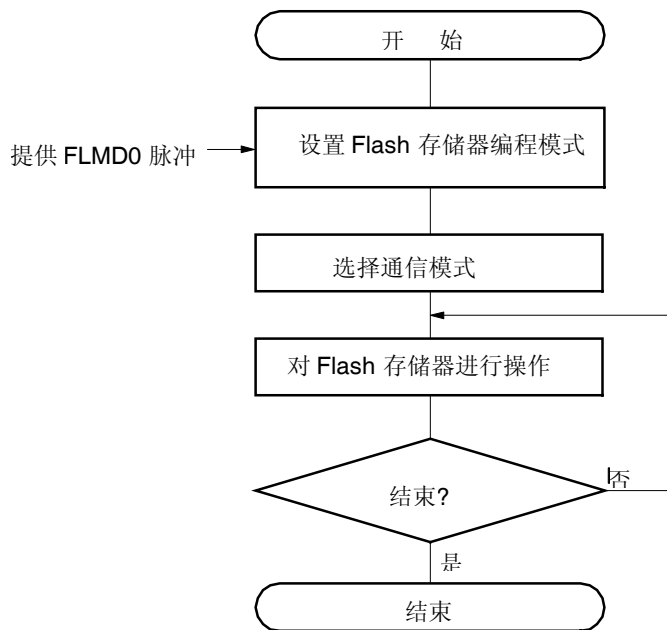
其它的供电电压 (LV_{DD}, LV_{SS}, AV_{REF} 和 AV_{SS}) 与正常操作模式中的相同。

26.7 编程方法

26.7.1 控制 Flash 存储器

下图显示了 Flash 存储器操作过程。

图 26-14. Flash 存储器操作过程



26.7.2 Flash 存储器编程模式

如果通过专用 Flash 编程器重写 Flash 存储器的内容，则必须将 78K0/LF2 设置为 Flash 存储器编程模式。而要设置该模式，必须将 FLMD0 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 26-15. Flash 存储器编程模式

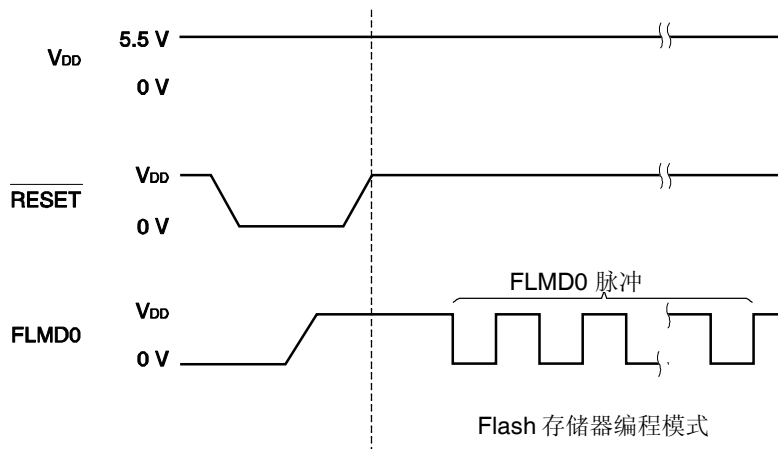


表 26-7. 复位释放后 FLMD0 引脚与操作模式之间的关系

FLMD0	操作模式
0	正常操作模式
V _{DD}	Flash 存储器编程模式

26.7.3 选择通信模式

在 78K0/LF2 中，进入专用 Flash 存储器编程模式后，通过将脉冲（可多达 11 个脉冲）输入到 FLMD0 引脚，来选择通信模式。由 Flash 编程器产生这些 FLMD0 脉冲。

下表显示了脉冲个数与通信模式之间的关系。

表 26-8. 通信模式

通信模式	标准设置 ^{注1}					使用的引脚	外部时钟	FLMD0 脉冲数
	端口	速率	使用目标	频率	倍率			
UART (UART6)	UART-ch0	115200 bps ^{注3}	可选择的	1 ~ 20 MHz ^{注2}	1.0	TxD6, RxD6	f _x	0
							f _{EXCLK}	3
3 线串行 I/O (CSI10)	SIO-ch0	2.4 kHz ~ 2.5 MHz				SO10, SI10, SCK10	f _{RH}	8

- 注
1. FlashPro4 应选择标准设置项目。
 2. 电压不同设置范围也不一样。如需了解详细内容，可参见电气特性章节。
 3. 波特率误差以外的因素，如信号波形瞬变，也会影响 UART 通信，因此必须象测量波特率误差一样严格测量瞬变。

注意事项 当选择 UART6 时，在接收到 FLMD0 脉冲后，根据专用 Flash 编程器发送的复位命令计算接收时钟。

备注

f_x: X1 时钟

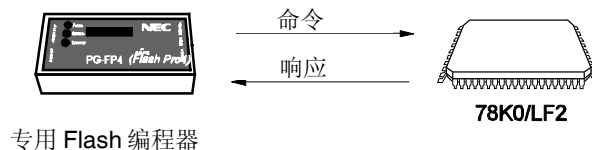
f_{EXCLK}: 外部主系统时钟

f_{RH}: 内部高速振荡时钟

26.7.4 通信命令

78K0/LF2 使用命令与专用 Flash 编程器进行通信。从 Flash 编程器发往 78K0/LF2 的信号称为命令，从 78K0/LF2 发往专用 Flash 编程器的信号称为响应。

图 26-16. 通信命令



78K0/LF2 的 Flash 存储器控制命令列表如下。所有这些命令都从编程器发出，78K0/LF2 根据相应的命令进行处理。

表 26-9. Flash 存储器控制命令

类别	命令名称	功能
校验	一次性校验命令	比较整个存储器内容和输入数据内容。
擦除	一次性擦除命令	擦除整个存储器内容。
空白检测	一次性空白检测命令	检查整个存储器的擦除状态。
数据写	高速写命令	写入数据到指定写入地址和写入字节，并执行校验检查。
	连续写命令	高速写入指令立即执行前给地址写入数据，并执行校验检查。
系统设置，控制	状态读命令	获得操作状态
	振荡频率设置命令	设置振荡频率
	擦除时间设置命令	设置一次性擦除的擦除时间
	写时间设置命令	设置写入数据的写入时间
	波特率设置命令	设置使用 UART 时的波特率
	硅标记命令	读取硅标记信息
	复位命令	逃离各个状态

78K0/LF2 根据专用 Flash 编程器发出的命令返回一个响应。78K0/LF2 发出的响应名称列表如下。

表 26-10. 响应命令

响应名称	功能
ACK	响应命令/数据
NAK	响应非法命令/数据

26.8 安全设置

以下显示使用安全设置命令能够执行的操作。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)

该项设置禁止对 Flash 存储器整个所有 block 进行 block 擦除和一次性擦除(片擦除)。一旦禁止执行一次性擦除(片擦除)命令，所有禁止设置项不能被取消。

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 block 擦除

该项设置禁止对 Flash 存储器中指定 block 进行 block 擦除。使用一次性擦除（片擦除）命令时可取消禁止设置。

- 禁止写

该项设置禁止对 Flash 存储器整个 block 进行写和 block 擦除。使用一次性擦除（片擦除）命令时可取消禁止设置。

- 禁止重写引导簇 0

该项设置禁止对 Flash 存储器中引导簇 0(0000H ~ 0FFFH)进行一次性擦除(片擦除)、block 擦除和写操作。

注意事项 如果对重写引导簇 0 进行了安全性设置，则不能再对该设备的引导簇 0 进行重写。

Flash 存储器在默认设置下，允许一次性擦除(片擦除)、block 擦除、写和重写引导簇 0 功能。以上安全性设置仅用于 On-board/Off-board 编程。各项安全性设置可以结合使用。

表 26-11 显示了 78K0/LF2 允许使用安全功能时擦除命令与写命令之间的关系。

表 26-11. 允许使用安全功能时命令之间的关系

安全设置 \ 命令	一次性擦除(片擦除)命令	Block 擦除命令	写入命令
禁止一次性擦除(片擦除)	无效	无效	有效 ^注
禁止 block 擦除	有效		有效
禁止写入			无效
禁止重写引导簇 0	无效		

注 由于禁止使用擦除命令，所以与已经写到 Flash 存储器中的数据不同的数据不能被写入。

表 26-12 显示了安全设置和各种编程模式操作之间的关系。

表 26-12. 安全设置和各种编程模式操作之间的关系

安全设置 \ 编程模式	On-board/Off-board 编程		自编程	
	安全设置	安全操作	安全设置	安全操作
禁止一次性擦除(片擦除)	允许	有效 ¹	禁止	无效 ²
禁止 block 擦除				
禁止写入				
禁止重写引导簇 0			允许	有效

- 注
1. 安全设置时禁止执行各种命令。
 2. 执行自编程命令不必考虑安全设置。

26.9 通过自写入进行 Flash 存储器编程

78K0/LF2 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 78K0/Kx2 自编程示例库重写 flash 存储器，因此可用于对程序升级。

如果在自编程期间产生中断，可以暂时停止自编程操作，而去执行中断服务程序。要执行中断服务程序，应在停止自编程操作后恢复正常操作模式，并执行 EI 指令。而在恢复自编程模式后，可以恢复自编程操作。

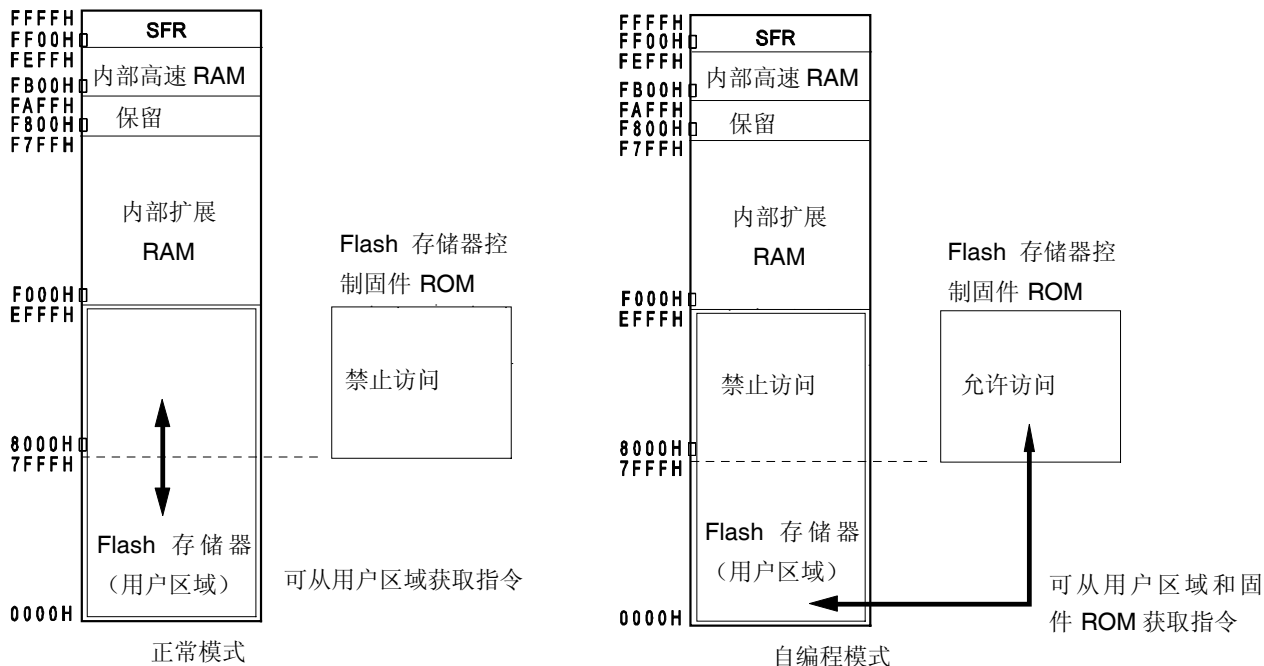
备注 如需了解自编程功能和 78K0/Kx2 自编程库的详细信息，可参见 **78K0/Kx2 Flash 存储器自编程用户手册 (U17516E)**。

- 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. 自编程期间 FLMD0 引脚应输入高电平。
 3. 在启动自编程之前必须执行 DI 指令。
自编程功能检查中断请求标志 (IF0L, IF0H, IF1L 和 IF1H)。如果产生了中断请求，则停止自编程操作。
 4. 即使在 DI 状态下也可以通过未屏蔽的中断请求停止自编程。要避免这种情况，可以通过使用中断屏蔽标志寄存器(MK0L, MK0H, MK1L 和 MK1H)屏蔽该中断。
 5. 在内部高速振荡时钟下执行自编程。如果 CPU 使用 X1 时钟或外部主系统时钟，则内部高速振荡时钟的振荡稳定等待时间应在自编程期间。

(注意事项 6 在下一页列出)

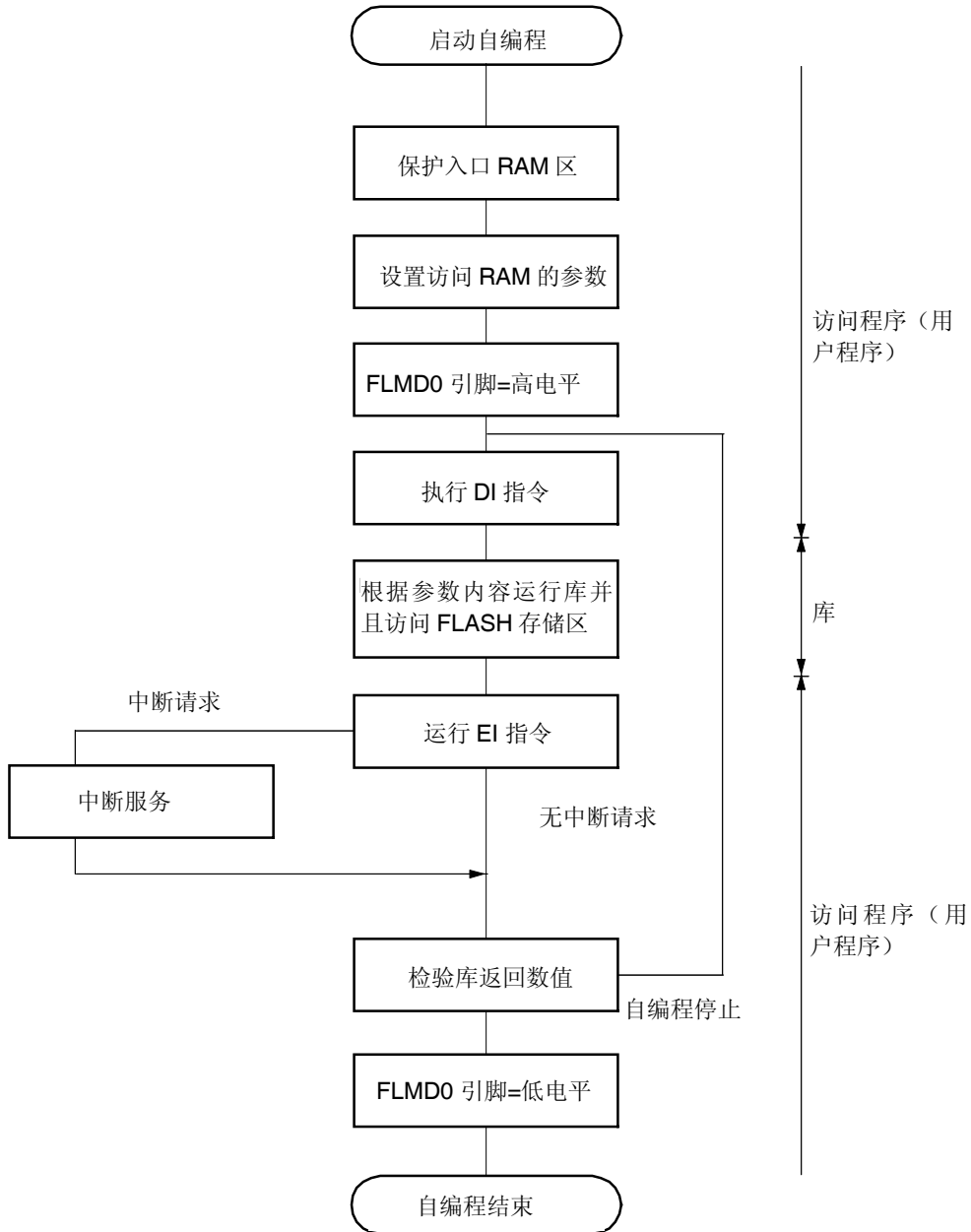
注意事项 6. 将用于自编程的入口程序分配在公共区域(0000H ~ 7FFFH)。

图 26-17. 用于自编程的操作模式和存储器映射图(μPD78F0375, 78F0385)



自编程过程显示如下。

图 26-18. 自编程流程



26.9.1 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导致重写引导区域的操作失败，则引导区域的数据可能会丢失且不能通过复位重启程序。

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 78K0/LF2 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。之后对初始引导程序区域，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。

如果已经将程序正确写入引导簇 0，则通过使用 78K0/LF2 固件的设置信息功能恢复初始引导区域。

注 一个引导簇的容量为 4 KB，通过引导交换功能交换引导簇 0 与引导簇 1 的内容。

引导簇 0 (0000H ~ 0FFFH): 初始引导程序区域

引导簇 1 (1000H ~ 1FFFH): 用于引导交换的区域

图 26-19. 引导交换功能

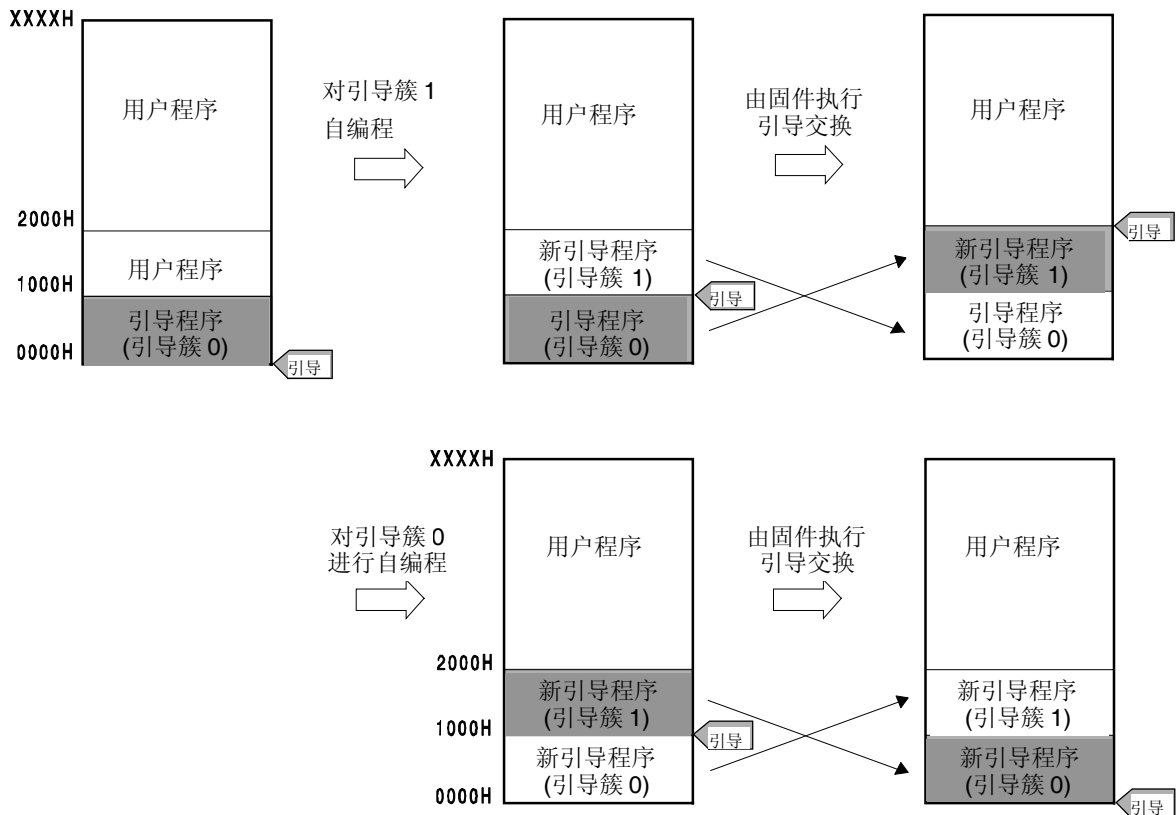
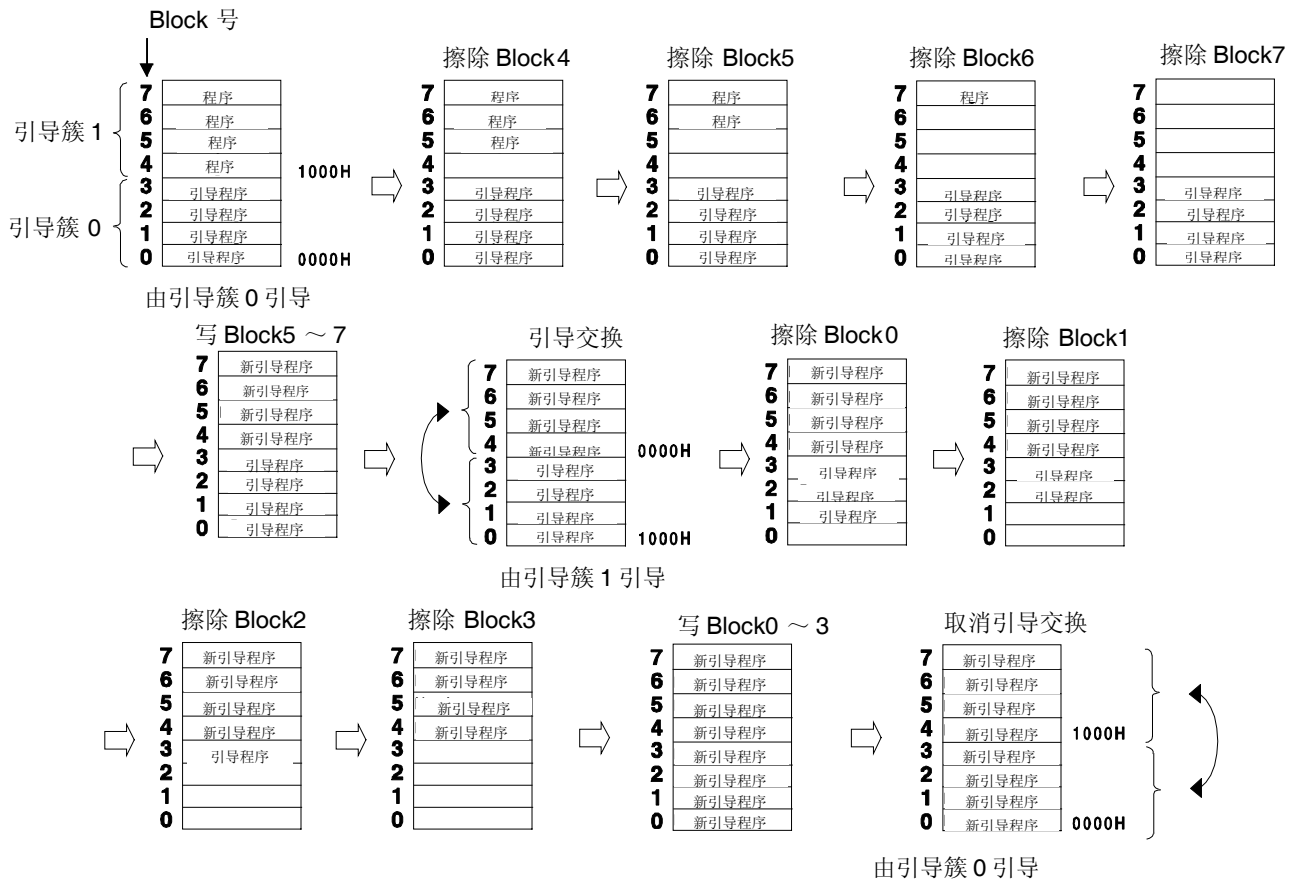


图 26-20. 引导交换执行示例

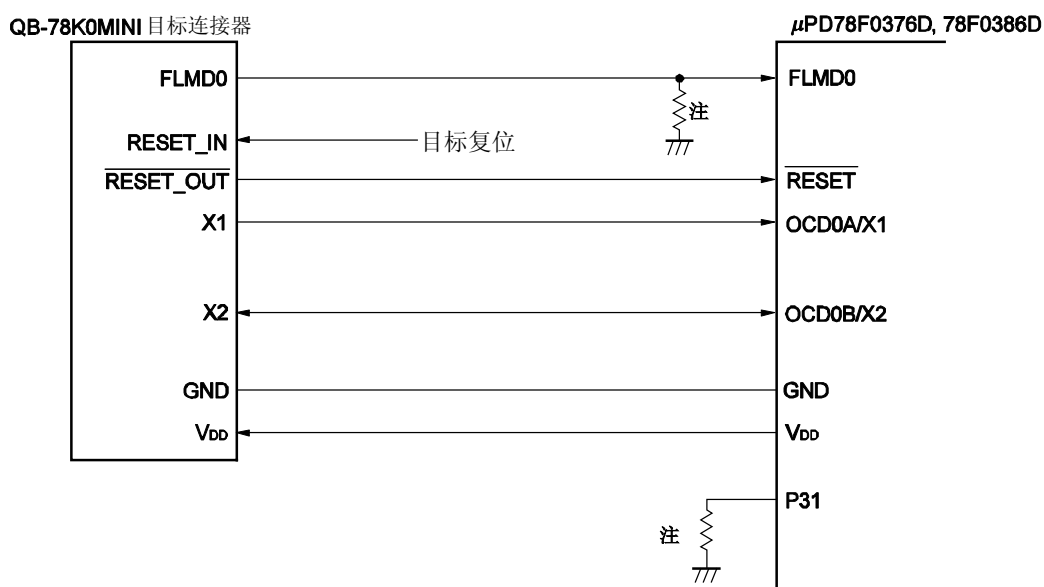


第二十七章 片上调试功能(仅限于 μ PD78F0376D 和 78F0386D)

μ PD78F0376D 和 78F0386D 使用 V_{DD} 、 $\overline{FLMD0}$ 、RESET、OCD0A/X1 (或 OCD1A/P31)、OCD0B/X2 (或 OCD1B/P32)和 V_{SS} 引脚, 通过片上调试仿真器 (QB-78K0MINI) 与主机通信, 进行片上调试。可以选择 OCD0A/X1 与 OCD1A/P31 或 OCD0B/X2 与 OCD1B/P32。

注意事项 μ PD78F0376D 和 78F0386D 具有片上调试功能。由于在使用片上调试功能后产品的稳定性不能得到保证, 因此在给定 flash 存储器可重写次数的情况下不要将该产品用于大规模生产。NEC 电子不接受有关该产品的投诉

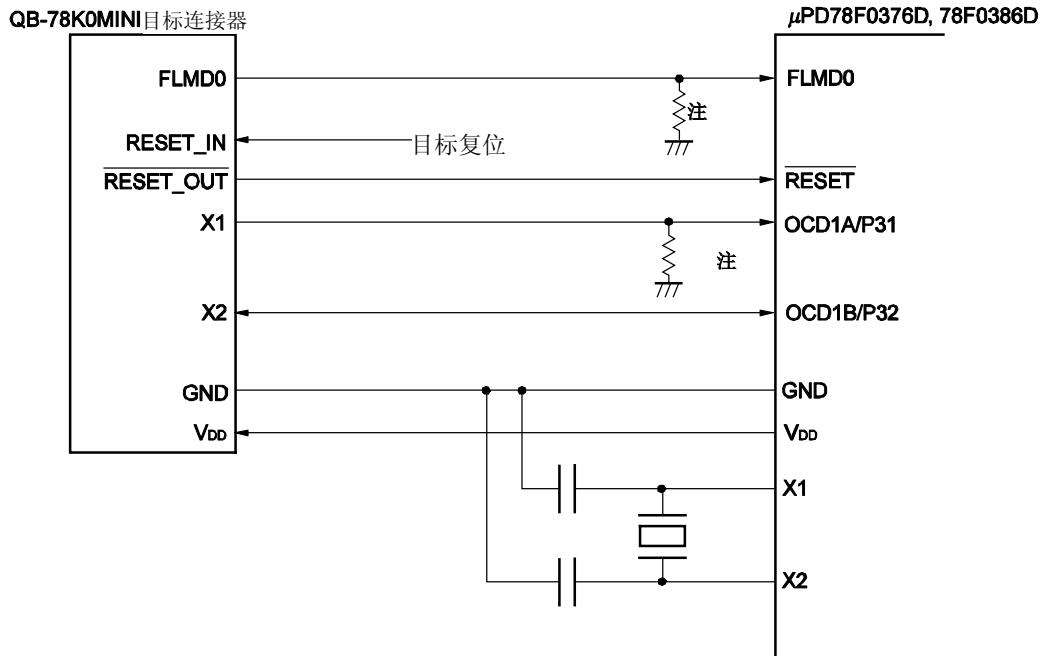
图 27-1. QB-78K0MINI 与 μ PD78F0376D 和 78F0386D 的连接示例
(当使用 OCD0A/X1 和 OCD0B/X2 时)



注 下拉电阻值至少应为 470 Ω 。

- 注意事项**
1. 在片上调试期间从 OCD0A/X1 引脚输入时钟。
 2. 通过外部下拉 OCD1A/P31 引脚控制 OCD0A/X1 和 OCD0B/X2 引脚。

图 27-2. QB-78K0MINI 与 μ PD78F0376D 和 78F0386D 的连接示例
(当使用 OCD1A 和 OCD1B 时)



注 下拉电阻值至少应为 470 Ω 。

27.1 片上调试安全 ID

在 μ PD78F0376D 和 78F0386D 中，一个片上调试操作控制标志存放在 flash 存储器地址 0084H 中 (参见第二十五章选项字节)，片上调试安全 ID 设置在地址 0085H ~ 008EH 中。

当使用引导交换功能时，由于 0084H、0085H ~ 008EH 与 1084H 以及 1085H ~ 108EH 的内容被交换，因此预先设置与 1084H 以及 1085H ~ 108EH 的内容相同的值。

如需了解片上调试安全 ID 的详细内容，可参见 QB-78K0MINI 用户手册 (U17029E)。

表 27-1. 片上调试安全 ID

地址	片上调试安全 ID
0085H ~ 008EH	任何 10 字节的 ID 代码
1085H ~ 108EH	

第二十八章 指令集

本章以表格方式列出了 78K0/LF2 的指令集。如需了解每种指令的操作和操作代码，可参阅 **78K/0 系列指令用户手册 (U12326E)**。

28.1 操作列表使用规则

28.1.1 操作数标识符和标识方法

根据规范确定的指令操作数标识方法（详情可参见汇编程序编程规范），在每种指令的“操作数”栏列出操作数。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[]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
- !: 绝对地址标识
- \$: 相对地址标识
- [: 间接地址标识

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[]。

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 28-1. 操作数标识符和标识方法

标识符	标识方法
<i>r</i> <i>rp</i> <i>sfr</i> <i>sfrp</i>	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) 特殊功能寄存器符号 ^注 特殊功能寄存器符号（仅用于 16 位可操作寄存器偶地址） ^注
<i>saddr</i> <i>saddrp</i>	FE20H ~ FF1FH 立即数或标号 FE20H ~ FF1FH 立即数或标号（仅用于偶地址）
<i>addr16</i> <i>addr11</i> <i>addr5</i>	0000H ~ FFFFH 立即数或标号 （仅用于 16 位数据传送指令的偶地址） 0800H ~ 0FFFH 立即数或标号 0040H ~ 007FH 立即数或标号（仅用于偶地址）
<i>word</i> <i>byte</i> <i>bit</i>	16 位立即数或标号 8 位立即数或标号 3 位立即数或标号
<i>RBn</i>	RB0 ~ RB3

注 不能使用这些操作数访问地址 FFD0H ~ FFD7H。

备注 特殊功能寄存器符号参见表 3-7 特殊功能寄存器列表

28.1.2 操作栏描述

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
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 位寄存器的高 8 位和低 8 位
^:	逻辑与 (AND)
∨:	逻辑或 (OR)
⊕:	逻辑异或 (exclusive OR)
—:	数据取反
addr16:	16 位立即数或标号
jdisp8:	带符号的 8 位数据 (偏移量)

28.1.3 标志操作栏的描述

(空):	不受影响
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

28.2 操作列表

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	4	-	r ← byte	
		saddr, #byte	3	6	7	(saddr) ← byte	
		sfr, #byte	3	-	7	sfr ← byte	
		A, r ^{注3}	1	2	-	A ← r	
		r, A ^{注3}	1	2	-	r ← A	
		A, saddr	2	4	5	A ← (saddr)	
		saddr, A	2	4	5	(saddr) ← A	
		A, sfr	2	-	5	A ← sfr	
		sfr, A	2	-	5	sfr ← A	
		A, laddr16	3	8	9	A ← (addr16)	
		laddr16, A	3	8	9	(addr16) ← A	
		PSW, #byte	3	-	7	PSW ← byte	x x x
		A, PSW	2	-	5	A ← PSW	
		PSW, A	2	-	5	PSW ← A	x x x
		A, [DE]	1	4	5	A ← (DE)	
		[DE], A	1	4	5	(DE) ← A	
		A, [HL]	1	4	5	A ← (HL)	
		[HL], A	1	4	5	(HL) ← A	
		A, [HL + byte]	2	8	9	A ← (HL + byte)	
		[HL + byte], A	2	8	9	(HL + byte) ← A	
	A, [HL + B]	1	6	7	A ← (HL + B)		
	[HL + B], A	1	6	7	(HL + B) ← A		
	A, [HL + C]	1	6	7	A ← (HL + C)		
	[HL + C], A	1	6	7	(HL + C) ← A		
	XCH	A, r ^{注3}	1	2	-	A ↔ r	
		A, saddr	2	4	6	A ↔ (saddr)	
		A, sfr	2	-	6	A ↔ (sfr)	
		A, laddr16	3	8	10	A ↔ (addr16)	
		A, [DE]	1	4	6	A ↔ (DE)	
		A, [HL]	1	4	6	A ↔ (HL)	
		A, [HL + byte]	2	8	10	A ↔ (HL + byte)	
		A, [HL + B]	2	8	10	A ↔ (HL + B)	
A, [HL + C]	2	8	10	A ↔ (HL + C)			

- 注
1. 当访问内部高速 RAM 时或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

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	rp, #word	3	6	–	rp ← word			
		saddrp, #word	4	8	10	(saddrp) ← word			
		sfrp, #word	4	–	10	sfrp ← word			
		AX, saddrp	2	6	8	AX ← (saddrp)			
		saddrp, AX	2	6	8	(saddrp) ← AX			
		AX, sfrp	2	–	8	AX ← sfrp			
		sfrp, AX	2	–	8	sfrp ← AX			
		AX, rp ^{注3}	1	4	–	AX ← rp			
		rp, AX ^{注3}	1	4	–	rp ← AX			
		AX, !addr16	3	10	12	AX ← (addr16)			
		!addr16, AX	3	10	12	(addr16) ← AX			
	XCHW	AX, rp ^{注3}	1	4	–	AX ↔ rp			
8 位操作	ADD	A, #byte	2	4	–	A, CY ← A + byte	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	x	x	x
		A, r ^{注4}	2	4	–	A, CY ← A + r	x	x	x
		r, A	2	4	–	r, CY ← r + A	x	x	x
		A, saddr	2	4	5	A, CY ← A + (saddr)	x	x	x
		A, !addr16	3	8	9	A, CY ← A + (addr16)	x	x	x
		A, [HL]	1	4	5	A, CY ← A + (HL)	x	x	x
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	x	x	x
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	x	x	x
	A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	x	x	x	
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	x	x	x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	x	x	x
		A, r ^{注4}	2	4	–	A, CY ← A + r + CY	x	x	x
		r, A	2	4	–	r, CY ← r + A + CY	x	x	x
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	x	x	x
		A, !addr16	3	8	9	A, CY ← A + (addr16) + C	x	x	x
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	x	x	x
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	x	x	x
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	x	x	x
A, [HL + C]		2	8	9	A, CY ← A + (HL + C) + CY	x	x	x	

- 注
1. 当访问内部高速 RAM 时或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. 仅当 rp = BC, DE 或 HL
 4. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU}) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

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	SUB	A, #byte	2	4	–	A, CY ← A – byte	x x x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	x x x
		A, r ^{注 3}	2	4	–	A, CY ← A – r	x x x
		r, A	2	4	–	r, CY ← r – A	x x x
		A, saddr	2	4	5	A, CY ← A – (saddr)	x x x
		A, !addr16	3	8	9	A, CY ← A – (addr16)	x x x
		A, [HL]	1	4	5	A, CY ← A – (HL)	x x x
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	x x x
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	x x x
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C)	x x x
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	x x x
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	x x x
		A, r ^{注 3}	2	4	–	A, CY ← A – r – CY	x x x
		r, A	2	4	–	r, CY ← r – A – CY	x x x
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	x x x
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	x x x
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	x x x
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	x x x
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	x x x
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C) – CY	x x x
	AND	A, #byte	2	4	–	A ← A ∧ byte	x
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	x
		A, r ^{注 3}	2	4	–	A ← A ∧ r	x
		r, A	2	4	–	r ← r ∧ A	x
		A, saddr	2	4	5	A ← A ∧ (saddr)	x
		A, !addr16	3	8	9	A ← A ∧ (addr16)	x
		A, [HL]	1	4	5	A ← A ∧ (HL)	x
		A, [HL + byte]	2	8	9	A ← A ∧ (HL + byte)	x
		A, [HL + B]	2	8	9	A ← A ∧ (HL + B)	x
		A, [HL + C]	2	8	9	A ← A ∧ (HL + C)	x

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

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	4	–	$A \leftarrow A \vee \text{byte}$		x	
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$		x	
		A, r ^{注 3}	2	4	–	$A \leftarrow A \vee r$		x	
		r, A	2	4	–	$r \leftarrow r \vee A$		x	
		A, saddr	2	4	5	$A \leftarrow A \vee (\text{saddr})$		x	
		A, !addr16	3	8	9	$A \leftarrow A \vee (\text{addr16})$		x	
		A, [HL]	1	4	5	$A \leftarrow A \vee (\text{HL})$		x	
		A, [HL + byte]	2	8	9	$A \leftarrow A \vee (\text{HL} + \text{byte})$		x	
		A, [HL + B]	2	8	9	$A \leftarrow A \vee (\text{HL} + B)$		x	
		A, [HL + C]	2	8	9	$A \leftarrow A \vee (\text{HL} + C)$		x	
	XOR	A, #byte	2	4	–	$A \leftarrow A \vee \text{byte}$		x	
		saddr, #byte	3	6	8	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$		x	
		A, r ^{注 3}	2	4	–	$A \leftarrow A \vee r$		x	
		r, A	2	4	–	$r \leftarrow r \vee A$		x	
		A, saddr	2	4	5	$A \leftarrow A \vee (\text{saddr})$		x	
		A, !addr16	3	8	9	$A \leftarrow A \vee (\text{addr16})$		x	
		A, [HL]	1	4	5	$A \leftarrow A \vee (\text{HL})$		x	
		A, [HL + byte]	2	8	9	$A \leftarrow A \vee (\text{HL} + \text{byte})$		x	
		A, [HL + B]	2	8	9	$A \leftarrow A \vee (\text{HL} + B)$		x	
		A, [HL + C]	2	8	9	$A \leftarrow A \vee (\text{HL} + C)$		x	
	CMP	A, #byte	2	4	–	$A - \text{byte}$	x	x	x
		saddr, #byte	3	6	8	$(\text{saddr}) - \text{byte}$	x	x	x
		A, r ^{注 3}	2	4	–	$A - r$	x	x	x
		r, A	2	4	–	$r - A$	x	x	x
		A, saddr	2	4	5	$A - (\text{saddr})$	x	x	x
		A, !addr16	3	8	9	$A - (\text{addr16})$	x	x	x
		A, [HL]	1	4	5	$A - (\text{HL})$	x	x	x
		A, [HL + byte]	2	8	9	$A - (\text{HL} + \text{byte})$	x	x	x
		A, [HL + B]	2	8	9	$A - (\text{HL} + B)$	x	x	x
		A, [HL + C]	2	8	9	$A - (\text{HL} + C)$	x	x	x

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时
 3. “r = A” 除外

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器（PCC）选择的 CPU 时钟（f_{cpu}）的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

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	6	–	AX, CY ← AX + word	x x x
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	x x x
	CMPW	AX, #word	3	6	–	AX – word	x x x
乘法/ 除法	MULU	X	2	16	–	AX ← A × X	
	DIVUW	C	2	25	–	AX (商), C (余数) ← AX ÷ C	
递增/ 递减	INC	r	1	2	–	r ← r + 1	x x
		saddr	2	4	6	(saddr) ← (saddr) + 1	x x
	DEC	r	1	2	–	r ← r – 1	x x
		saddr	2	4	6	(saddr) ← (saddr) – 1	x x
	INCW	rp	1	4	–	rp ← rp + 1	
	DECW	rp	1	4	–	rp ← rp – 1	
循环	ROR	A, 1	1	2	–	(CY, A7 ← A0, Am-1 ← Am) × 1 次	x
	ROL	A, 1	1	2	–	(CY, A0 ← A7, Am+1 ← Am) × 1 次	x
	RORC	A, 1	1	2	–	(CY ← A0, A7 ← CY, Am-1 ← Am) × 1 time	x
	ROLC	A, 1	1	2	–	(CY ← A7, A0 ← CY, Am+1 ← Am) × 1 time	x
	ROR4	[HL]	2	10	12	A3-0 ← (HL)3-0, (HL)7-4 ← A3-0, (HL)3-0 ← (HL)7-4	
	ROL4	[HL]	2	10	12	A3-0 ← (HL)7-4, (HL)3-0 ← A3-0, (HL)7-4 ← (HL)3-0	
BCD 调整	ADJBA		2	4	–	加法后的十进制调整	x x x
	ADJBS		2	4	–	减法后的十进制调整	x x x
位操作	MOV1	CY, saddr.bit	3	6	7	CY ← (saddr.bit)	x
		CY, sfr.bit	3	–	7	CY ← sfr.bit	x
		CY, A.bit	2	4	–	CY ← A.bit	x
		CY, PSW.bit	3	–	7	CY ← PSW.bit	x
		CY, [HL].bit	2	6	7	CY ← (HL).bit	x
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY	
		sfr.bit, CY	3	–	8	sfr.bit ← CY	
		A.bit, CY	2	4	–	A.bit ← CY	
		PSW.bit, CY	3	–	8	PSW.bit ← CY	x x
[HL].bit, CY	2	6	8	(HL).bit ← CY			

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (fcpu) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

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	AND1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \wedge (\text{saddr.bit})$			×	
		CY, sfr.bit	3	-	7	$CY \leftarrow CY \wedge \text{sfr.bit}$			×	
		CY, A.bit	2	4	-	$CY \leftarrow CY \wedge A.\text{bit}$			×	
		CY, PSW.bit	3	-	7	$CY \leftarrow CY \wedge \text{PSW.bit}$			×	
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \wedge (\text{HL}).\text{bit}$			×	
	OR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \vee (\text{saddr.bit})$			×	
		CY, sfr.bit	3	-	7	$CY \leftarrow CY \vee \text{sfr.bit}$			×	
		CY, A.bit	2	4	-	$CY \leftarrow CY \vee A.\text{bit}$			×	
		CY, PSW.bit	3	-	7	$CY \leftarrow CY \vee \text{PSW.bit}$			×	
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \vee (\text{HL}).\text{bit}$			×	
	XOR1	CY, saddr.bit	3	6	7	$CY \leftarrow CY \oplus (\text{saddr.bit})$			×	
		CY, sfr.bit	3	-	7	$CY \leftarrow CY \oplus \text{sfr.bit}$			×	
		CY, A.bit	2	4	-	$CY \leftarrow CY \oplus A.\text{bit}$			×	
		CY, PSW.bit	3	-	7	$CY \leftarrow CY \oplus \text{PSW.bit}$			×	
		CY, [HL].bit	2	6	7	$CY \leftarrow CY \oplus (\text{HL}).\text{bit}$			×	
	SET1	saddr.bit	2	4	6	$(\text{saddr.bit}) \leftarrow 1$				
		sfr.bit	3	-	8	$\text{sfr.bit} \leftarrow 1$				
		A.bit	2	4	-	$A.\text{bit} \leftarrow 1$				
		PSW.bit	2	-	6	$\text{PSW.bit} \leftarrow 1$		×	×	×
		[HL].bit	2	6	8	$(\text{HL}).\text{bit} \leftarrow 1$				
	CLR1	saddr.bit	2	4	6	$(\text{saddr.bit}) \leftarrow 0$				
		sfr.bit	3	-	8	$\text{sfr.bit} \leftarrow 0$				
		A.bit	2	4	-	$A.\text{bit} \leftarrow 0$				
		PSW.bit	2	-	6	$\text{PSW.bit} \leftarrow 0$		×	×	×
		[HL].bit	2	6	8	$(\text{HL}).\text{bit} \leftarrow 0$				
	SET1	CY	1	2	-	$CY \leftarrow 1$			1	
	CLR1	CY	1	2	-	$CY \leftarrow 0$			0	
	NOT1	CY	1	2	-	$CY \leftarrow \overline{CY}$			×	

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

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	!addr16	3	7	-	$(SP - 1) \leftarrow (PC + 3)_H, (SP - 2) \leftarrow (PC + 3)_L,$ $PC \leftarrow \text{addr16}, SP \leftarrow SP - 2$	
	CALLF	!addr11	2	5	-	$(SP - 1) \leftarrow (PC + 2)_H, (SP - 2) \leftarrow (PC + 2)_L,$ $PC_{15-11} \leftarrow 00001, PC_{10-0} \leftarrow \text{addr11},$ $SP \leftarrow SP - 2$	
	CALLT	[addr5]	1	6	-	$(SP - 1) \leftarrow (PC + 1)_H, (SP - 2) \leftarrow (PC + 1)_L,$ $PC_H \leftarrow (00000000, \text{addr5} + 1),$ $PC_L \leftarrow (00000000, \text{addr5}),$ $SP \leftarrow SP - 2$	
	BRK		1	6	-	$(SP - 1) \leftarrow \text{PSW}, (SP - 2) \leftarrow (PC + 1)_H,$ $(SP - 3) \leftarrow (PC + 1)_L, PC_H \leftarrow (003FH),$ $PC_L \leftarrow (003EH), SP \leftarrow SP - 3, IE \leftarrow 0$	
	RET		1	6	-	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $SP \leftarrow SP + 2$	
	RETI		1	6	-	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $\text{PSW} \leftarrow (SP + 2), SP \leftarrow SP + 3$	R R R
	RETB		1	6	-	$PC_H \leftarrow (SP + 1), PC_L \leftarrow (SP),$ $\text{PSW} \leftarrow (SP + 2), SP \leftarrow SP + 3$	R R R
堆栈操作	PUSH	PSW	1	2	-	$(SP - 1) \leftarrow \text{PSW}, SP \leftarrow SP - 1$	
		rp	1	4	-	$(SP - 1) \leftarrow \text{rp}_H, (SP - 2) \leftarrow \text{rp}_L,$ $SP \leftarrow SP - 2$	
	POP	PSW	1	2	-	$\text{PSW} \leftarrow (SP), SP \leftarrow SP + 1$	R R R
		rp	1	4	-	$\text{rp}_H \leftarrow (SP + 1), \text{rp}_L \leftarrow (SP),$ $SP \leftarrow SP + 2$	
	MOVW	SP, #word	4	-	10	$SP \leftarrow \text{word}$	
		SP, AX	2	-	8	$SP \leftarrow \text{AX}$	
AX, SP		2	-	8	$\text{AX} \leftarrow \text{SP}$		
无条件转移	BR	!addr16	3	6	-	$PC \leftarrow \text{addr16}$	
		\$addr16	2	6	-	$PC \leftarrow PC + 2 + \text{jdisp8}$	
		AX	2	8	-	$\text{PCH} \leftarrow \text{A}, \text{PCL} \leftarrow \text{X}$	
条件转移	BC	\$addr16	2	6	-	如果 $\text{CY} = 1$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	BNC	\$addr16	2	6	-	如果 $\text{CY} = 0$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	BZ	\$addr16	2	6	-	如果 $\text{Z} = 1$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	
	BNZ	\$addr16	2	6	-	如果 $\text{Z} = 0$, 则 $PC \leftarrow PC + 2 + \text{jdisp8}$	

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{cpu}) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

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	BT	saddr.bit, \$addr16	3	8	9	如果(saddr.bit) = 1, 则 PC ← PC + 3 + jdisp8		
		sfr.bit, \$addr16	4	-	11	如果 sfr.bit = 1, 则 PC ← PC + 4 + jdisp8		
		A.bit, \$addr16	3	8	-	如果 A.bit = 1, 则 PC ← PC + 3 + jdisp8		
		PSW.bit, \$addr16	3	-	9	如果 PSW.bit = 1, 则 PC ← PC + 3 + jdisp8		
		[HL].bit, \$addr16	3	10	11	如果(HL).bit = 1, 则 PC ← PC + 3 + jdisp8		
	BF	saddr.bit, \$addr16	4	10	11	如果(saddr.bit) = 0, 则 PC ← PC + 4 + jdisp8		
		sfr.bit, \$addr16	4	-	11	如果 sfr.bit = 0, 则 PC ← PC + 4 + jdisp8		
		A.bit, \$addr16	3	8	-	如果 A.bit = 0, 则 PC ← PC + 3 + jdisp8		
		PSW.bit, \$addr16	4	-	11	如果 PSW.bit = 0, 则 PC ← PC + 4 + jdisp8		
		[HL].bit, \$addr16	3	10	11	如果 (HL).bit = 0, 则 PC ← PC + 3 + jdisp8		
	BTCLR	saddr.bit, \$addr16	4	10	12	如果(saddr.bit) = 1, 则 PC ← PC + 4 + jdisp8 然后复位 (saddr.bit)		
		sfr.bit, \$addr16	4	-	12	如果 sfr.bit = 1, 则 PC ← PC + 4 + jdisp8 然后复位 sfr.bit		
		A.bit, \$addr16	3	8	-	如果 A.bit = 1, 则 PC ← PC + 3 + jdisp8 然后复位 A.bit		
		PSW.bit, \$addr16	4	-	12	如果 PSW.bit = 1, 则 PC ← PC + 4 + jdisp8 然后复位 PSW.bit	× × ×	
		[HL].bit, \$addr16	3	10	12	如果(HL).bit = 1, 则 PC ← PC + 3 + jdisp8 然后复位 (HL).bit		
	DBNZ	B, \$addr16	2	6	-	B ← B - 1, 然后 如果 B ≠ 0, 则 PC ← PC + 2 + jdisp8		
		C, \$addr16	2	6	-	C ← C - 1, 然后 如果 C ≠ 0, 则 PC ← PC + 2 + jdisp8		
		saddr, \$addr16	3	8	10	(saddr) ← (saddr) - 1, 然后 如果(saddr) ≠ 0, 则 PC ← PC + 3 + jdisp8		
	CPU 控制	SEL	RBn	2	4	-	RBS1, 0 ← n	
		NOP		1	2	-	无操作	
EI			2	-	6	IE ← 1 (允许中断)		
DI			2	-	6	IE ← 0 (禁止中断)		
HALT			2	6	-	设置 HALT 模式		
STOP			2	6	-	设置 STOP 模式		

- 注
1. 当访问内部高速 RAM 或针对无数据访问的指令
 2. 当访问内部高速 RAM 以外的区域时

- 备注
1. 一个指令时钟周期是指由处理器时钟控制寄存器 (PCC) 选择的 CPU 时钟 (f_{CPU}) 的一个周期。
 2. 该时钟周期用于内部 ROM 程序。

28.3 按寻址类型列出指令

(1) 8 位指令

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

第 2 操作数 第 1 操作数	#byte	A	R [#]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	\$addr16	1	无
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

注 “r = A” 除外

(2) 16 位指令

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

第 2 操作数 第 1 操作数	#word	AX	Rp [※]	sfrp	saddrp	laddr16	SP	无
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW [※]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
laddr16		MOVW						
SP	MOVW	MOVW						

注 仅当 rp = BC, DE, HL 时使用

(3) 位操作指令

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

第 2 操作数 第 1 操作数	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	\$addr16	无
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) 调用指令 / 转移指令

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

第 2 操作数 第 1 操作数	AX	laddr16	laddr11	[addr5]	\$addr16
基本指令	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
复合指令					BT BF BTCLR DBNZ

(5) 其它指令

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

第二十九章 电气特性

注意事项 μ PD78F0376D 和 78F0386D 具有片上调试功能。不要将该产品用于大规模生产，因为在给定 Flash 重写次数的条件下，使用片上调试功能后，产品的可靠性得不到保证。NEC 电子不接受有关该产品的任何投诉。

最大额定值 (T_A = 25°C) (1/2)

参数	符号	条件		额定值	单位
<R> 供电电压	V _{DD}	V _{DD} = LV _{DD}		-0.3 ~ +6.5	V
<R>	LV _{DD}	V _{DD} = LV _{DD}		-0.3 ~ +6.5	V
<R>	V _{SS}	V _{SS} = LV _{SS}		-0.3 ~ +0.3	V
<R>	LV _{SS}	V _{SS} = LV _{SS}		-0.3 ~ +0.3	V
<R>	AV _{REF}			-0.3 ~ V _{DD} + 0.3 [※]	V
<R>	AV _{SS}			-0.3 ~ +0.3	V
<R> 输入电压	V _{I1}	P00, P01, P10 ~ P17, P20 ~ P27, P30 ~ P33, P70 ~ P76, P120 ~ P124, X1, X2, XT1, XT2, FLMD0, RESET		-0.3 ~ V _{DD} + 0.3 [※]	V
	V _{I2}	SCL0, SDA0 (N-ch 漏极开路)		-0.3 ~ +6.5	V
<R> 输出电压	V _{O1}	P00, P01, P10 ~ P17, P20 ~ P27, P30 ~ P33, P70 ~ P76, P120 ~ P124, X1, X2, XT1, XT2, RESET		-0.3 ~ V _{DD} + 0.3 [※]	V
	V _{O2}	S0 ~ S35, COM0 ~ COM3		-0.3 ~ V _{LC0} + 0.3 [※]	
模拟输入电压	V _{AN}	ANI0 ~ ANI7		-0.3 ~ AV _{REF} + 0.3 [※] 和 -0.3 ~ V _{DD} + 0.3 [※]	V
<R> 输出电流, 高	I _{OH}	每个引脚	P00, P01, P10 ~ P17, P30 ~ P33, P70 ~ P76, P120	-10	mA
		所有引脚总和 -80 mA	P00, P01, P120	-25	mA
			P10 ~ P17, P30 ~ P33, P70 ~ P76	-55	mA
<R>		每个引脚	P20 ~ P27	-0.5	mA
<R>		所有引脚总和		-2	mA
<R>		每个引脚	P121 ~ P124	-1	mA
<R>		所有引脚总和		-4	mA

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的相同。

最大额定值 ($T_A = 25^\circ\text{C}$) (2/2)

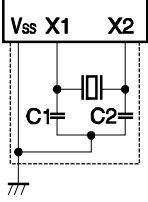
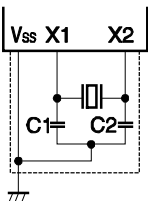
参数	符号	条件		额定值	单位
输出电流, 低	I _{oL}	每个引脚	P00, P01, P10 ~ P17, P30 ~ P33, P70 ~ P76, P120, SCL0, SDA0	30	mA
		所有引脚总和 200 mA	P00, P01, P120	60	mA
			P10 ~ P17, P30 ~ P33, P70 ~ P76, SCL0, SDA0	140	mA
		每个引脚	P20 ~ P27	1	mA
		所有引脚总和		5	mA
		每个引脚	P121 ~ P124	4	mA
		所有引脚总和		10	mA
工作环境温度	T _A	正常操作模式	-40 ~ +85	°C	
		flash 存储器编程模式			
储藏温度	T _{stg}		-40 ~ +125	°C	

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的相同。

X1 振荡器特性

(T_A = -40 ~ +85°C, 1.8 V ≤ V_{DD} = LV_{DD} ≤ 5.5 V, V_{SS} = LV_{SS} = AV_{SS} = 0 V)

振荡器	建议使用的电路	参数	条件	MIN.	TYP.	MAX.	单位
陶瓷振荡器		X1 时钟振荡频率 (f _x) ^{注 1}	4.0 V ≤ V _{DD} ≤ 5.5 V	1.0 ^{注 2}		20.0	MHz
			2.7 V ≤ V _{DD} < 4.0 V	1.0 ^{注 2}		10.0	
			1.8 V ≤ V _{DD} < 2.7 V	1.0		5.0	
晶体振荡器		X1 时钟振荡频率 (f _x) ^{注 1}	4.0 V ≤ V _{DD} ≤ 5.5 V	1.0 ^{注 2}		20.0	MHz
			2.7 V ≤ V _{DD} < 4.0 V	1.0 ^{注 2}		10.0	
			1.8 V ≤ V _{DD} < 2.7 V	1.0		5.0	

注 1. 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**。

<R> 2. 当使用 UART6 在线写时，2.0 MHz (MIN.)。

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，以防止连接电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 流经的电流变化较大的信号线不要在振荡器周围布线。
- 要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 振荡器的选择和常数，使用者可选择任一参数的振荡器或由振荡器制造厂商推荐。

内部振荡器特性

(TA = -40 ~ +85°C, 1.8 V ≤ VDD = LVDD ≤ 5.5 V, VSS = LVSS = AVSS = 0 V)

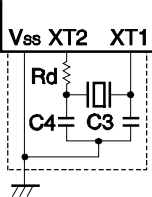
振荡器	参数	条件	MIN.	TYP.	MAX.	单位	
<R> 8 MHz 内部振荡器	内部高速振荡时钟频率 (fRH) [※]	RSTS = 1	2.7 V ≤ VDD ≤ 5.5 V	7.6	8.0	8.4	MHz
			1.8 V ≤ VDD < 2.7 V	7.6	8.0	10.4	MHz
		RSTS = 0		2.48	5.6	9.86	MHz
<R> 240 kHz 内部振荡器	内部低速振荡时钟频率 (fRL)	2.7 V ≤ VDD ≤ 5.5 V	216	240	264	kHz	
		1.8 V ≤ VDD < 2.7 V	192	240	264	kHz	

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**。

备注 RSTS: 内部振荡模式寄存器(RCM) 的第 7 位

XT1 振荡器特性

(TA = -40 ~ +85°C, 1.8 V ≤ VDD = LVDD ≤ 5.5 V, VSS = LVSS = AVSS = 0 V)

振荡器	建议使用的电路	参数	条件	MIN.	TYP.	MAX.	单位
晶体振荡器		XT1 时钟振荡频率 (fxt) [※]		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，以防止连接线电容产生不利影响。

- 连接线越短越好。
- 连接线不应与其他信号线交叉。
- 流经的电流变化较大的信号线不要在振荡器周围布线。
- 要保持振荡器电容器的接地点电压与 VSS 相同。
- 不要将电容的地信号接入大电流地。
- 不要从振荡器获取信号。

2. XT1 振荡器是一个低振幅电路，用于降低功耗，但由于比 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 振荡器的选择和常数，使用者可选择任一参数的振荡器或由振荡器制造厂商推荐。

DC 特性 (1/5)

(T_A = -40 ~ +85°C, 1.8 V ≤ V_{DD} = LV_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = LV_{SS} = AV_{SS} = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位			
输出电流, 高 ^{注1}	IOH1	P00, P01, P10 ~ P17, P30 ~ P33, P70 ~ P76, P120 的每个引脚	4.0 V ≤ V _{DD} ≤ 5.5 V			-3.0	mA		
			2.7 V ≤ V _{DD} < 4.0 V			-2.5	mA		
			1.8 V ≤ V _{DD} < 2.7 V			-1.0	mA		
		P00, P01, P120 的所有引脚 ^{注3}	4.0 V ≤ V _{DD} ≤ 5.5 V				-20.0	mA	
			2.7 V ≤ V _{DD} < 4.0 V				-10.0	mA	
			1.8 V ≤ V _{DD} < 2.7 V				-5.0	mA	
		P10 ~ P17, P30 ~ P33, P70 ~ P76 的所有引脚 ^{注3}	4.0 V ≤ V _{DD} ≤ 5.5 V				-30.0	mA	
			2.7 V ≤ V _{DD} < 4.0 V				-19.0	mA	
			1.8 V ≤ V _{DD} < 2.7 V				-10.0	mA	
		所有引脚 ^{注3}	4.0 V ≤ V _{DD} ≤ 5.5 V				-50.0	mA	
			2.7 V ≤ V _{DD} < 4.0 V				-29.0	mA	
			1.8 V ≤ V _{DD} < 2.7 V				-15.0	mA	
	IOH2	P20 ~ P27 的每个引脚	AV _{REF} = V _{DD}			-0.1	mA		
	IOH3	P121 ~ P124 的每个引脚				-0.1	mA		
输出电流, 低 ^{注2}	IOL1	P00, P01, P10 ~ P17, P30 ~ P33, P70 ~ P76, 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			5.0	mA		
			1.8 V ≤ V _{DD} < 2.7 V			2.0	mA		
		SCL0, SDA0 的每个引脚	4.0 V ≤ V _{DD} ≤ 5.5 V				15.0	mA	
			2.7 V ≤ V _{DD} < 4.0 V				3.0	mA	
			1.8 V ≤ V _{DD} < 2.7 V				0.6	mA	
		P00, P01, P120 的所有引脚 ^{注3}	4.0 V ≤ V _{DD} ≤ 5.5 V				20.0	mA	
			2.7 V ≤ V _{DD} < 4.0 V				15.0	mA	
			1.8 V ≤ V _{DD} < 2.7 V				9.0	mA	
		P10 ~ P17, P30 ~ P33, P70 ~ P76, SCL0, SDA0 的所有引脚 ^{注3}	4.0 V ≤ V _{DD} ≤ 5.5 V				45.0	mA	
			2.7 V ≤ V _{DD} < 4.0 V				35.0	mA	
			1.8 V ≤ V _{DD} < 2.7 V				20.0	mA	
		所有引脚 ^{注3}	4.0 V ≤ V _{DD} ≤ 5.5 V				65.0	mA	
			2.7 V ≤ V _{DD} < 4.0 V				50.0	mA	
			1.8 V ≤ V _{DD} < 2.7 V				29.0	mA	
			IOL2	P20 ~ P27 的每个引脚	AV _{REF} = V _{DD}			0.4	mA
			IOL3	P121 ~ P124 的每个引脚				0.4	mA

- 注
- 即使电流从 V_{DD} 流向输出引脚, 设备操作时的电流值也可以得到保证。
 - 即使电流从输出引脚流向 GND, 设备操作时的电流值也可以得到保证。
 - 占空比=70%时 (输出电流的时间为 0.7 × t, 不输出电流的时间为 0.3 × t, 其中 t 为指定时间)的规定。可通过以下表达式计算占空比不等于 70%时的引脚总的输出电流。
 - 当 I_{OH} 的占空比为 n%: 引脚总的输出电流 = (I_{OH} × 0.7)/(n × 0.01)
 - <示例> 当占空比为 50%, I_{OH} = 20.0 mA
 引脚总的输出电流 = (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。

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

DC 特性 (2/5)

(T_A = -40 ~ +85°C, 1.8 V ≤ V_{DD} = LV_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = LV_{SS} = AV_{SS} = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
<R> 输入电压, 高	V _{IH1}	P12, P13, P15, P121 ~ P124	0.7V _{DD}		V _{DD}	V
	V _{IH2}	P00, P01, P10, P11, P14, P16, P17, P30 ~ P33, P70 ~ P76, P120, RESET	0.8V _{DD}		V _{DD}	V
	V _{IH3}	P20 ~ P27	AV _{REF} = V _{DD}		AV _{REF}	V
	V _{IH4}	SCL0, SDA0	0.7V _{DD}		6.0	
<R> 输入电压, 低	V _{IL1}	P12, P13, P15, P121 ~ P124, SCL0, SDA0	0		0.3V _{DD}	V
	V _{IL2}	P00, P01, P10, P11, P14, P16, P17, P30 ~ P33, P70 ~ P76, P120, RESET	0		0.2V _{DD}	V
	V _{IL3}	P20 ~ P27	AV _{REF} = V _{DD}		0.3AV _{REF}	V
输出电压, 高	V _{OH1}	P00, P01, P10 ~ P17, P30 ~ P33, P70 ~ P76, P120	4.0 V ≤ V _{DD} ≤ 5.5 V, I _{OH1} = -3.0 mA		V _{DD} - 0.7	V
			2.7 V ≤ V _{DD} < 4.0 V, I _{OH1} = -2.5 mA		V _{DD} - 0.5	V
			1.8 V ≤ V _{DD} < 2.7 V, I _{OH1} = -1.0 mA		V _{DD} - 0.5	V
	V _{OH2}	P20 ~ P27	AV _{REF} = V _{DD} , I _{OH2} = -0.1 mA		V _{DD} - 0.5	V
			P121 ~ P124	I _{OH2} = (0.1 mA)	V _{DD} (0.5)	V
输出电压, 低	VOL1	P00, P01, P10 ~ P17, P30 ~ P33, P70 ~ P76, P120	4.0 V (V _{DD} (5.5 V, I _{OL1} = 8.5 mA		0.7	V
			2.7 V (V _{DD} < 4.0 V, I _{OL1} = 5.0 mA		0.7	V
			1.8 V ≤ V _{DD} < 2.7 V, I _{OL1} = 2.0 mA		0.5	V
			1.8 V ≤ V _{DD} < 2.7 V, I _{OL1} = 0.5 mA		0.4	V
	VOL2	P20 ~ P27	AV _{REF} = V _{DD} , I _{OL2} = 0.4 mA		0.4	V
			P121 ~ P124	I _{OL2} = 0.4 mA		0.4
	VOL3	SCL0, SDA0	4.0 V ≤ V _{DD} ≤ 5.5 V, I _{OL3} = 15 mA		2.0	V
			4.0 V ≤ V _{DD} ≤ 5.5 V, I _{OL3} = 3.0 mA		0.4	V
			2.7 V ≤ V _{DD} < 4.0 V, I _{OL3} = 3.0 mA		0.6	V
			2.7 V ≤ V _{DD} < 4.0 V, I _{OL3} = 2.0 mA		0.4	V
1.8 V ≤ V _{DD} < 2.7 V, I _{OL3} = 0.6 mA				0.5	V	

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

DC 特性 (3/5)

(T_A = -40 ~ +85°C, 1.8 V ≤ V_{DD} = LV_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = LV_{SS} = AV_{SS} = 0 V)

参数	符号	条件		MIN.	TYP.	MAX.	单位	
输入漏电流, 高	I _{LIH1}	P00, P01, P10 ~ P17, P30 ~ P33, P70 ~ P76, P120, SCL0, SDA0, FLMD0, RESET	V _I = V _{DD}			1	μA	
	I _{LIH2}	P20 ~ P27	V _I = AV _{REF} = V _{DD}			1	μA	
	I _{LIH3}	P121 ~ 124 (X1, X2, XT1, XT2)	V _I = V _{DD}	I/O 端口模式			1	μA
				OSC 模式			20	μA
输入漏电流, 低	I _{LIL1}	P00, P01, P10 ~ P17, P30 ~ P33, P70 ~ P76, P120, SCL0, SDA0, FLMD0, RESET	V _I = V _{SS}			-1	μA	
	I _{LIL2}	P20 ~ P27	V _I = V _{SS} , AV _{REF} = V _{DD}			-1	μA	
	I _{LIL3}	P121 ~ 124 (X1, X2, XT1, XT2)	V _I = V _{SS}	I/O 端口模式			-1	μA
				OSC 模式			-20	μA
上拉电阻	R _U	V _I = V _{SS}		10	20	100	kΩ	
FLMD0 供电电压	V _{IL}	正常操作模式		0		0.2V _{DD}	V	
	V _{IH}	自编程模式		0.8V _{DD}		V _{DD}	V	

备注 除非另外说明, 复用功能引脚的特性与端口引脚的相同。

DC 特性 (4/5)

(TA = -40 ~ +85°C, 1.8 V ≤ VDD = LVDD ≤ 5.5 V, AVREF ≤ VDD, VSS = LVSS = AVSS = 0 V)

参数	符号	条件		MIN.	TYP.	MAX.	单位				
<R>	IDD1	操作模式	fXH = 20 MHz ^{※2} , VDD = 5.0 V	方波输入		3.2	5.5	mA			
				振荡器连接		4.5	6.9				
			fXH = 10 MHz ^{※2,3} , VDD = 5.0 V	方波输入		1.6	2.8	mA			
				振荡器连接		2.3	3.9				
			fXH = 10 MHz ^{※2,3} , VDD = 3.0 V	方波输入		1.5	2.7	mA			
				振荡器连接		2.2	3.2				
			fXH = 5 MHz ^{※2,3} , VDD = 3.0 V	方波输入		0.9	1.6	mA			
				振荡器连接		1.3	2.0				
			fXH = 5 MHz ^{※2,3} , VDD = 2.0 V	方波输入		0.7	1.4	mA			
				振荡器连接		1.0	1.6				
			fRH = 8 MHz, VDD = 5.0 V					1.4	2.5	mA	
			fSUB = 32.768 kHz ^{※4} , VDD = 5.0 V	方波输入		6	25	μA			
				振荡器连接		15	30				
			<R>	IDD2	HALT 模式	fXH = 20 MHz ^{※2} , VDD = 5.0 V	方波输入		0.8	2.6	mA
振荡器连接		2.0					4.4				
fXH = 10 MHz ^{※2,3} , VDD = 5.0 V	方波输入					0.4	1.3	mA			
	振荡器连接					1.0	2.4				
fXH = 5 MHz ^{※2,3} , VDD = 3.0 V	方波输入					0.2	0.65	mA			
	振荡器连接					0.5	1.1				
fRH = 8 MHz, VDD = 5.0 V							0.4	1.2	mA		
fSUB = 32.768 kHz ^{※4} , VDD = 5.0 V	方波输入					3.0	22	μA			
	振荡器连接					12	25				
<R>	IDD3 ^{※5}	STOP 模式				VDD = 5.0 V			1	20	μA
						VDD = 5.0 V, TA = -40 ~ +70°C			1	10	μA

注 1. 流入内部供电电源的总电流(VDD, AVREF), 包括外部操作电流和当输入引脚固定为 VDD 或 VSS 输入漏电流。不包括流入上拉电阻的电流和端口的输出电流。

2. 不包括 8 MHz 内部振荡器的操作电流和流入 A/D 转换器、看门狗定时器、LVI 电路和 LCD 控制器/驱动器的电流。

3. 当 AMPH (时钟操作模式选择寄存器(OSCCTL)的第 0 位) = 0。

4. 不包括 X1 振荡、8 MHz 内部振荡器和 240 kHz 内部振荡器的操作电流, 和流入 A/D 转换器、看门狗定时器、LVI 电路和 LCD 控制器/驱动器的电流。

5. 不包括 240 kHz 内部振荡器和 XT1 振荡的操作电流, 和流入 A/D 转换器、看门狗定时器、LVI 电路和 LCD 控制器/驱动器的电流。

- 备注
- fXH: 高速系统时钟频率 (X1 时钟振荡频率或外部主系统时钟频率)
 - fRH: 内部高速振荡时钟频率
 - fSUB: 副系统时钟频率 (XT1 时钟振荡频率或外部副系统时钟频率)

DC 特性 (5/5)

(T_A = -40 ~ +85°C, 1.8 V ≤ V_{DD} = LV_{DD} ≤ 5.5 V, AV_{REF} ≤ V_{DD}, V_{SS} = LV_{SS} = AV_{SS} = 0 V)

参数	符号	条件		MIN.	TYP.	MAX.	单位
A/D 控制器操作电流	I _{ADC} ^{註1}	以最大速率转换期间	2.3 V ≤ AV _{REF} ≤ V _{DD}		0.86	1.9	mA
看门狗定时器操作电流	I _{WDT} ^{註2}	240 kHz 内部低速振荡时钟操作期间			5	10	μA
LVI 操作电流	I _{LVI} ^{註3}				9	18	μA
LCD 操作电流	I _{LCD1} ^{註4}	当 LCD (包括升压电路) 停止和 IIC 操作时	LV _{DD} = 5.0 V		150	330	μA
			LV _{DD} = 3.0 V		75	160	μA
	I _{LCD2} ^{註4}	当仅有 LCD 升压电路和 IIC 处于后备状态时	LV _{DD} = 5.0 V		2	36	μA
			LV _{DD} = 3.0 V		1.5	16	μA
	I _{LCD3} ^{註4}	当 LCD 显示和 IIC 处于后备状态时	LV _{DD} = 5.0 V		5	45	μA
			LV _{DD} = 3.0 V		4	22	μA
	I _{LCD4} ^{註4}	当 LCD (包括倍压电路) 停止和 IIC 处于后备状态时	LV _{DD} = 5.0 V		0.1	5	μA
			LV _{DD} = 3.0 V		0.05	3	μA

- 注
1. 仅包括流过 A/D 转换器的电流。当 A/D 转换器运行在操作模式或 HALT 模式，78K0/LF2 的电流值通过 I_{ADC} 加 I_{DD1} 或 I_{DD2} 获得。
 2. 仅包括流过看门狗定时器（包括 240 kHz 内部振荡器的操作电流）的电流。当看门狗定时器运行在 HALT 模式或 STOP 模式，78K0/LF2 的电流值通过 I_{WDT} 加 I_{DD2} 或 I_{DD3} 获得。
 3. 仅包括流过 LVI 电路的电流。当 LVI 电路运行在 HALT 模式或 STOP 模式，78K0/LF2 的电流值通过 I_{LVI} 加 I_{DD2} 或 I_{DD3} 获得。
 4. 仅包括流过 LCD 控制器/驱动器的电流。78K0/LF2 的电流值通过供电电压(I_{DD1}, I_{DD2} 或 I_{DD3}) 加 LCD 工作电流 (I_{LCD1}, I_{LCD2}, I_{LCD3} 或 I_{LCD4}) 获得。

AC 特性

(1) 基本操作

 $(T_A = -40 \sim +85^\circ\text{C}, 1.8\text{ V} \leq V_{DD} = LV_{DD} \leq 5.5\text{ V}, AV_{REF} \leq V_{DD}, V_{SS} = LV_{SS} = AV_{SS} = 0\text{ V})$

参数	符号	条件	MIN.	TYP.	MAX.	单位	
指令周期 (指令最短执行时间)	T_{CY}	主系统时钟 (f_{XP}) 操作	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.1		32	μs
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	0.2		32	μs
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$0.4^{\#1}$		32	μs
		副系统时钟 (f_{SUB}) 操作		114	122	125	μs
外部主系统时钟频率	f_{EXCLK}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$1.0^{\#2}$		20.0	MHz	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	$1.0^{\#2}$		10.0	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	1.0		5.0	MHz	
<R> 外部主系统时钟输入高电平宽度, 低电平宽度	$t_{EXCLKH},$ t_{EXCLKL}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	24		500	ns	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	48		500	ns	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	96		500	ns	
外部副系统时钟频率	f_{EXCLKS}		32	32.768	35	kHz	
<R> 外部副系统时钟输入高电平宽度、低电平宽度	$t_{EXCLKSH},$ $t_{EXCLKSL}$		12			ns	
<R> TI000, TI010 输入高电平宽度、低电平宽度	$t_{TIH0},$ t_{TIL0}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$2/f_{sam} + 0.1^{\#3}$			μs	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	$2/f_{sam} + 0.2^{\#3}$			μs	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$2/f_{sam} + 0.5^{\#3}$			μs	
TI50, TI51 输入频率	f_{TI5}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			10	MHz	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			10	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			5	MHz	
TI50, TI51 输入高电平宽度、低电平宽度	$t_{TIH5},$ t_{TIL5}	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	50			ns	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	50			ns	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	100			ns	
中断输入高电平宽度、低电平宽度	$t_{INTH},$ t_{INTL}		1			μs	
按键返回输入低电平宽度	t_{KR}		250			ns	
RESET 低电平宽度	t_{RSL}		10			μs	

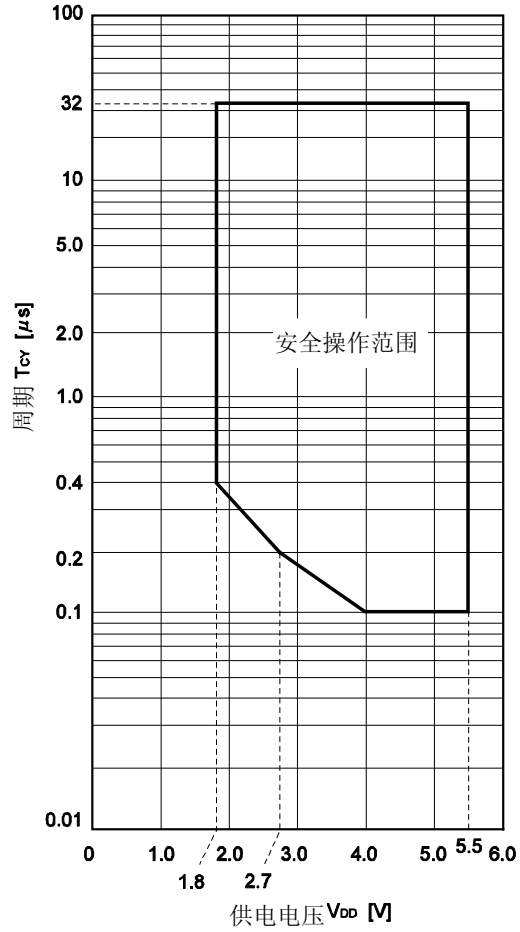
注 1. 当使用 8 MHz 内部振荡器时为 0.38 μs 。

2. 当使用 UART6 在线写时, 2.0 MHz (MIN.)。

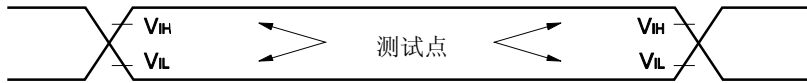
3. 使用预分频模式寄存器 00 (PRM00) 的第 0 位和第 1 位 (PRM000、PRM001) 可以选择 $f_{sam} = f_{PRS}$, $f_{PRS}/4$, $f_{PRS}/256$ 。注意, 当选择 TI000 的有效沿作为计数时钟时, $f_{sam} = f_{PRS}$ 。

<R>

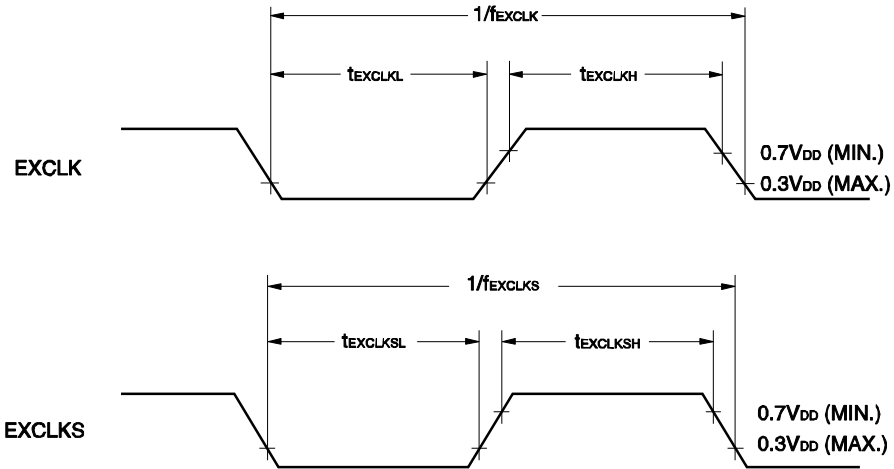
T_{CY} vs. V_{DD} (主系统时钟操作)



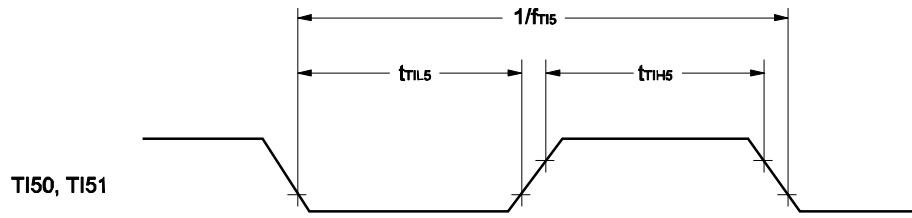
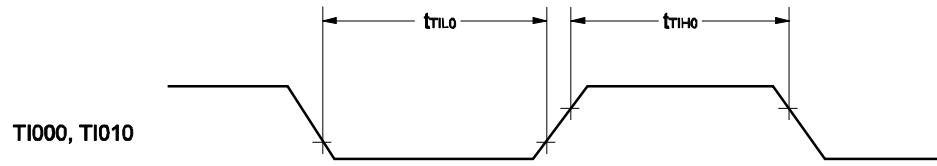
<R> AC 时序测试点(不包括外部主系统时钟和副系统时钟)



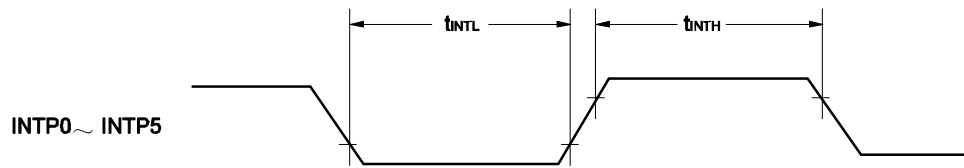
外部主系统时钟时序, 外部副系统时钟时序



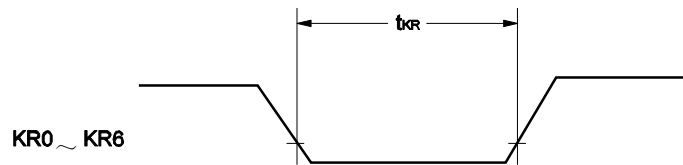
TI 时序



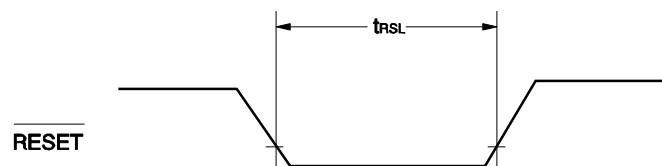
中断请求输入时序



按键中断输入时序



RESET 输入时序



(2) 串行接口

(TA = -40 ~ +85°C, 1.8 V ≤ VDD = LVDD ≤ 5.5 V, VSS = LVSS = AVSS = 0 V)

(a) UART6 (专用波特率发生器输出)

参数	符号	条件	MIN.	TYP.	MAX.	单位
传送速率					625	kbps

(b) UART0 (专用波特率发生器输出)

参数	符号	条件	MIN.	TYP.	MAX.	单位
传送速率					625	kbps

(c) IIC0

参数	符号	标准模式		高速模式		单位
		MIN.	MAX.	MIN.	MAX.	
SCL0 时钟频率	fSCL	0	100	0	400	kHz
启动/重新启动条件的设置时间 ^{注1}	tSU:STA	4.8	–	0.7	–	μs
保持时间	tHD:STA	4.1	–	0.7	–	μs
当 SCL0 = “L”时的保持时间	tLOW	5.0	–	1.25	–	μs
当 SCL0 = “H”时的保持时间	tHIGH	5.0	–	1.25	–	μs
数据建立时间 (接收)	tSU:DAT	0	–	0	–	μs
数据保持时间 (发送) ^{注2}	tHD:DAT	0.47	4.0	0.23	1.00	μs

- 注
1. 当检测到启动/重启条件时在经历了该周期时间后产生第一个时钟脉冲。
 2. tHD:DAT的最大值(MAX.)是在正常传送期间, 在 $\overline{\text{ACK}}$ (应答)时序中插入一个等待状态。

(d) CSI10 (主设备模式, $\overline{\text{SCK10}}$... 内部时钟输出)

	参数	符号	条件	MIN.	TYP.	MAX.	单位
<R>	$\overline{\text{SCK10}}$ 周期时间	t_{KCY1}	$4.0 \text{ V} \leq V_{\text{DD}} \leq 5.5 \text{ V}$	200			ns
<R>			$2.7 \text{ V} \leq V_{\text{DD}} < 4.0 \text{ V}$	400			ns
<R>			$1.8 \text{ V} \leq V_{\text{DD}} < 2.7 \text{ V}$	600			ns
<R>	$\overline{\text{SCK10}}$ 高/低电平宽度	t_{KH1} , t_{KL1}	$4.0 \text{ V} \leq V_{\text{DD}} \leq 5.5 \text{ V}$	$t_{\text{KCY1}}/2 - 20$ #1			ns
<R>			$2.7 \text{ V} \leq V_{\text{DD}} < 4.0 \text{ V}$	$t_{\text{KCY1}}/2 - 30$ #1			ns
<R>			$1.8 \text{ V} \leq V_{\text{DD}} < 2.7 \text{ V}$	$t_{\text{KCY1}}/2 - 60$ #1			ns
<R>	SI10 设置时间 (到 $\overline{\text{SCK10}}\uparrow$)	t_{SIK1}	$4.0 \text{ V} \leq V_{\text{DD}} \leq 5.5 \text{ V}$	70			ns
<R>			$2.7 \text{ V} \leq V_{\text{DD}} < 4.0 \text{ V}$	100			ns
<R>			$1.8 \text{ V} \leq V_{\text{DD}} < 2.7 \text{ V}$	100			ns
	SI10 保持时间 (从 $\overline{\text{SCK10}}\uparrow$)	t_{KSI1}		30			ns
	从 $\overline{\text{SCK10}}\downarrow$ 到 SO10 输出的延迟时间	t_{KSO1}	$C = 50 \text{ pF}$ #2			40	ns

- 注
1. 使用高速系统时钟 (f_{XH}) 时的取值。
 2. C 是 $\overline{\text{SCK10}}$ 和 SO10 输出线的负载电容。

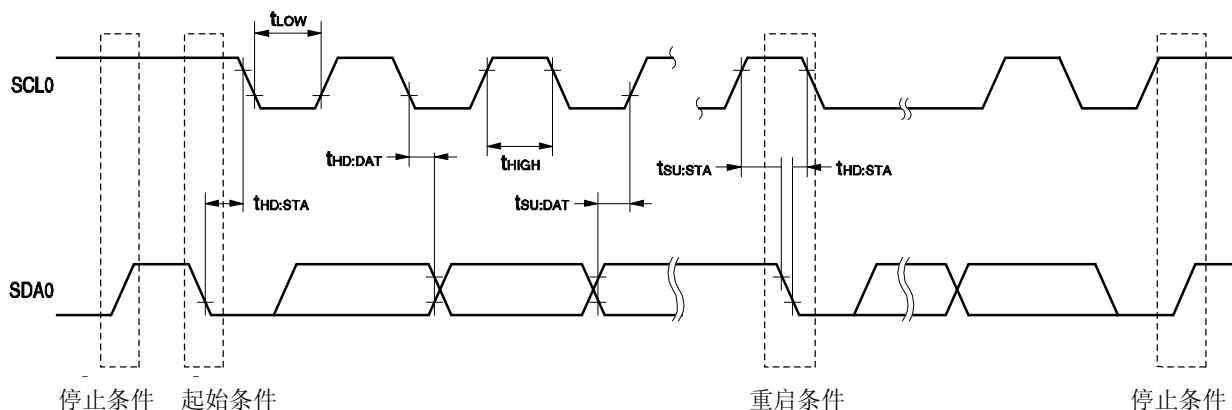
(e) CSI10 (从设备模式, $\overline{\text{SCK10}}$... 外部时钟输入)

	参数	符号	条件	MIN.	TYP.	MAX.	单位
	$\overline{\text{SCK10}}$ 周期	t_{KCY2}		400			ns
	$\overline{\text{SCK10}}$ 高/低电平宽度	t_{KH2} , t_{KL2}		$t_{\text{KCY2}}/2$			ns
	SI10 设置时间 (至 $\overline{\text{SCK10}}\uparrow$)	t_{SIK2}		80			ns
	SI10 保持时间 (从 $\overline{\text{SCK10}}\uparrow$)	t_{KSI2}		50			ns
<R>	从 $\overline{\text{SCK10}}\downarrow$ 到 SO10 输出的延迟时间	t_{KSO2}	$C = 50 \text{ pF}$ #	$4.0 \text{ V} \leq V_{\text{DD}} \leq 5.5 \text{ V}$		120	ns
<R>				$2.7 \text{ V} \leq V_{\text{DD}} < 4.0 \text{ V}$		120	ns
<R>				$1.8 \text{ V} \leq V_{\text{DD}} < 2.7 \text{ V}$		180	ns

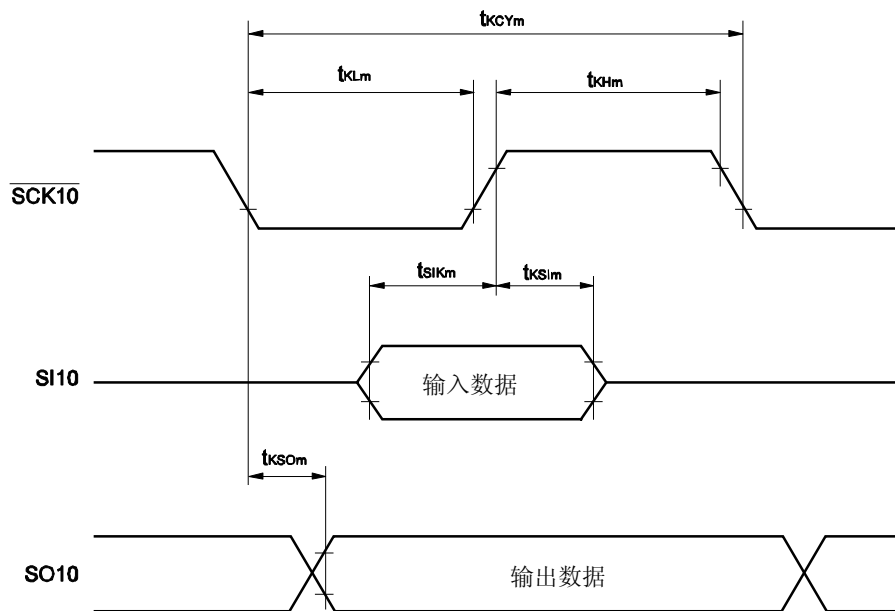
- 注 C 是 SO10 输出线的负载电容。

串行传送时序

IIC0:



CSI10:



备注 $m = 1, 2$

A/D 转换器特性(仅限于 $\mu\text{PD78F037x}$)(T_A = -40 ~ +85°C, 1.8 V ≤ V_{DD} = LV_{DD} ≤ 5.5 V, 2.3 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = LV_{SS} = AV_{SS} = 0 V)

参数	符号	条件	MIN.	TYP.	MAX.	单位
分辨率	R _{ES}				10	位
总误差 ^{注1,2}	A _{INL}	4.0 V ≤ AV _{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV _{REF} < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV _{REF} < 2.7 V			±1.2	%FSR
转换时间	t _{CONV}	4.0 V ≤ AV _{REF} ≤ 5.5 V	6.1		36.7	μs
		2.7 V ≤ AV _{REF} < 4.0 V	12.2		36.7	μs
		2.3 V ≤ AV _{REF} < 2.7 V	27		66.6	μs
零度误差 ^{注1,2}	E _{ZS}	4.0 V ≤ AV _{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV _{REF} < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV _{REF} < 2.7 V			±0.6	%FSR
满度误差 ^{注1,2}	E _{FS}	4.0 V ≤ AV _{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV _{REF} < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV _{REF} < 2.7 V			±0.6	%FSR
积分非线性误差 ^{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
		2.3 V ≤ AV _{REF} < 2.7 V			±6.5	LSB
微分非线性误差 ^{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
		2.3 V ≤ AV _{REF} < 2.7 V			±2.0	LSB
模拟输入电压	V _{AIN}		AV _{SS}		AV _{REF}	V

- 注 1. 不包括量化误差 (±1/2 LSB)
 2. 该值表示满度的百分比 (%FSR)

LCD 特性($T_A = -40 \sim +85^\circ\text{C}$, $2.0\text{ V} \leq \text{LV}_{\text{DD}} \leq 5.5\text{ V}$)

(1) 电阻分压模式

(a) 静态显示模式 ($2.0\text{ V} \leq \text{LV}_{\text{DD}} \leq 5.5\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
LCD 驱动电压	V_{LCD}		2.0		LV_{DD}	V
LCD 驱动电阻 ^{※1}	R_{LCD}		60	100	150	$\text{k}\Omega$
LCD 输出电阻 ^{※2} (Common)	R_{ODC}				40	$\text{k}\Omega$
LCD 输出电阻 ^{※2} (Segment)	R_{ODS}				200	$\text{k}\Omega$
LV_{DD} 和 V_{LC0} 上拉电阻 ^{※3}	R_{LU}	$\text{LV}_{\text{DD}} = 5.0\text{ V}$, $V_{\text{LC0}} = 3.0\text{ V}$		7.3		$\text{k}\Omega$

(b) 1/3 偏压模式 ($2.5\text{ V} \leq \text{LV}_{\text{DD}} \leq 5.5\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
LCD 驱动电压	V_{LCD}		2.5		LV_{DD}	V
LCD 驱动电阻 ^{※1}	R_{LCD}		60	100	150	$\text{k}\Omega$
LCD 输出电阻 ^{※2} (Common)	R_{ODC}				40	$\text{k}\Omega$
LCD 输出电阻 ^{※2} (Segment)	R_{ODS}				200	$\text{k}\Omega$
LV_{DD} 和 V_{LC0} 上拉电阻 ^{※3}	R_{LU}	$\text{LV}_{\text{DD}} = 5.0\text{ V}$, $V_{\text{LC0}} = 3.0\text{ V}$		7.3		$\text{k}\Omega$

(c) 1/2 偏压模式 ($2.7\text{ V} \leq \text{LV}_{\text{DD}} \leq 5.5\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
LCD 驱动电压	V_{LCD}		2.7		LV_{DD}	V
LCD 驱动电阻 ^{※1}	R_{LCD}		60	100	150	$\text{k}\Omega$
LCD 输出电阻 ^{※2} (Common)	R_{ODC}	$T_A = -10 \sim +85^\circ\text{C}$			40	$\text{k}\Omega$
		$T_A = -40 \sim -10^\circ\text{C}$			60	$\text{k}\Omega$
LCD 输出电阻 ^{※2} (Segment)	R_{ODS}				200	$\text{k}\Omega$
LV_{DD} 和 V_{LC0} 上拉电阻 ^{※3}	R_{LU}	$\text{LV}_{\text{DD}} = 5.0\text{ V}$, $V_{\text{LC0}} = 3.0\text{ V}$		7.3		$\text{k}\Omega$

- 注
1. 当仅连接内部电阻时。
 2. 输出电阻是 V_{LC0} , V_{LC1} , V_{LC2} 和 V_{SS} 引脚与任一 SEG 和 COM 引脚间的电阻。
 3. 当 LCD 模式由 LCD 模式设置寄存器(LCDMD)开始设置时, 需分离。

备注 上述表中所述的值含 $V_{\text{LC0}} \sim V_{\text{LC2}}$ 和 GND 间连接 $0.47\mu\text{F}$ 电容。

(2) 内部升压模式($1.8\text{ V} \leq \text{LVDD} \leq 5.5\text{ V}$)

参数	符号	条件		MIN.	TYP.	MAX.	单位	
LCD 输出电压波动范围	V_{LCD2}	$C1 \sim C4^{\#1}$ $= 0.47\ \mu\text{F}^{\#2}$	GAIN = 0	CTSEL1 = 0, CTSEL0 = 1	1.35	1.43	1.51	V
				CTSEL1 = 0, CTSEL0 = 0	1.42	1.50	1.58	V
				CTSEL1 = 1, CTSEL0 = 1	1.48	1.57	1.66	V
				CTSEL1 = 1, CTSEL0 = 0	$1.54^{\#3}$	$1.63^{\#3}$	$1.72^{\#3}$	V
		GAIN = 1	CTSEL1 = 0, CTSEL0 = 1	0.87	0.93	1.00	V	
			CTSEL1 = 0, CTSEL0 = 0	0.94	1.00	1.06	V	
			CTSEL1 = 1, CTSEL0 = 1	1.00	1.07	1.14	V	
			CTSEL1 = 1, CTSEL0 = 0	1.06	1.13	1.20	V	
两倍输出电压	V_{LCD1}	$C1 \sim C4^{\#1} = 0.47\ \mu\text{F}^{\#2}$			$2 V_{\text{LCD2}}$		V	
三倍输出电压	V_{LCD0}	$C1 \sim C4^{\#1} = 0.47\ \mu\text{F}^{\#2}$			$3 V_{\text{LCD2}}$		V	
<R> 升压等待时间 ^{#4}	t_{VWAIT}	GAIN = 1	$4.5\text{ V} \leq \text{LVDD} \leq 5.5\text{ V}$	4			s	
			$1.8\text{ V} \leq \text{LVDD} < 4.5\text{ V}$	0.5			s	
<R>		GAIN = 0		0.5			s	
LCD 输出电阻 ^{#5} (Common)	R_{ODC}					40	k Ω	
LCD 输出电阻 ^{#5} (Segment)	R_{ODS}					200	k Ω	

注 1. LCD 驱动需使用电压引脚连接电容。

C1: CAPH 和 CAPL 间电容

C2: V_{LC0} 和 GND 间电容

C3: V_{LC1} 和 GND 间电容

C4: V_{LC2} 和 GND 间电容

2. 当频率小于等于 128 Hz 时, SEG 和 COM 引脚开路, 并且(LCDON, SCOC, VLCON) = 111B.

3. 当操作电压范围为 $2.0\text{ V} \leq \text{LVDD} < 5.5\text{ V}$ 时。

4. 等待时间包括从升压开始直到显示为止。

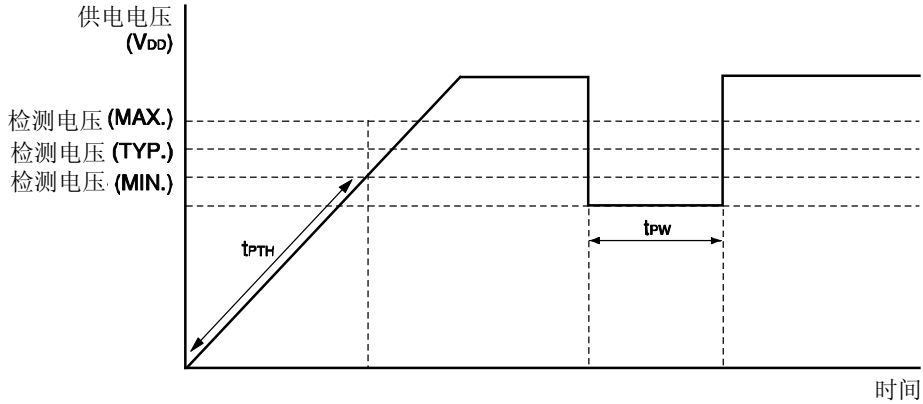
(LCDON = 1)

5. 输出电阻是 V_{LC0} , V_{LC1} , V_{LC2} 和 V_{SS} 引脚与任一 SEG 和 COM 引脚间的电阻。

1.59 V POC 电路特性 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{SS} = LV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
检测电压	V_{POC}		1.44	1.59	1.74	V
供电电压的上升斜率	t_{PTH}	$V_{DD}: 0\text{ V} \rightarrow$ 变到 V_{POC} 的指定电压	0.5			V/ms
最小脉冲宽度	t_{PW}		200			μs

POC 电路时序

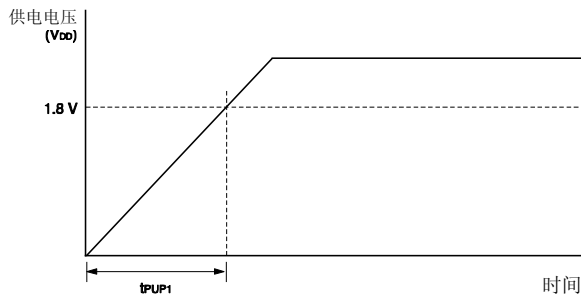


供电电压上升时间($T_A = -40 \sim +85^\circ\text{C}$, $V_{SS} = LV_{SS} = 0\text{ V}$)

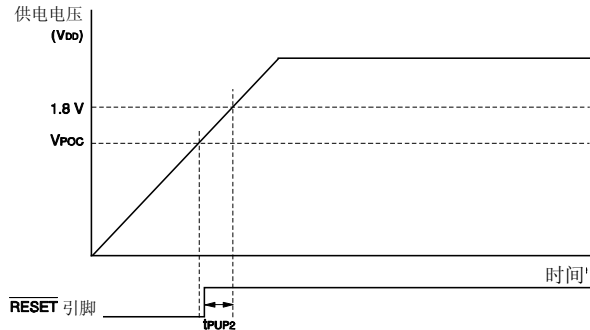
参数	符号	条件	MIN.	TYP.	MAX.	单位
上升至 1.8 V ($V_{DD}(\text{MIN.})$)所需的最长时间 ($V_{DD}: 0\text{ V} \rightarrow 1.8\text{ V}$)	t_{PUP1}	当没有使用 $\overline{\text{RESET}}$ 输入时, $\text{POCMODE}(\text{选项字节}) = 0$			3.6	ms
上升至 1.8 V ($V_{DD}(\text{MIN.})$)所需的最长时间 (释放 $\overline{\text{RESET}}$ 输入 $\rightarrow V_{DD}: 1.8\text{ V}$)	t_{PUP2}	当使用 $\overline{\text{RESET}}$ 输入时, $\text{POCMODE}(\text{选项字节}) = 0$			1.9	ms

供电电压上升时间时序

•当没有使用 $\overline{\text{RESET}}$ 引脚输入时



•当使用 $\overline{\text{RESET}}$ 引脚输入时



2.7 V POC 电路特性 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{SS} = LV_{SS} = 0\text{ V}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
供电电压应用中的检测电压	V_{DDPOC}	$\text{POCMODE}(\text{选项字节}) = 1$	2.50	2.70	2.90	V

LVI 电路特性 ($T_A = -40 \sim +85^\circ\text{C}$, $V_{POC} \leq V_{DD} = LV_{DD} \leq 5.5\text{ V}$, $V_{SS} = LV_{SS} = 0\text{ V}$)

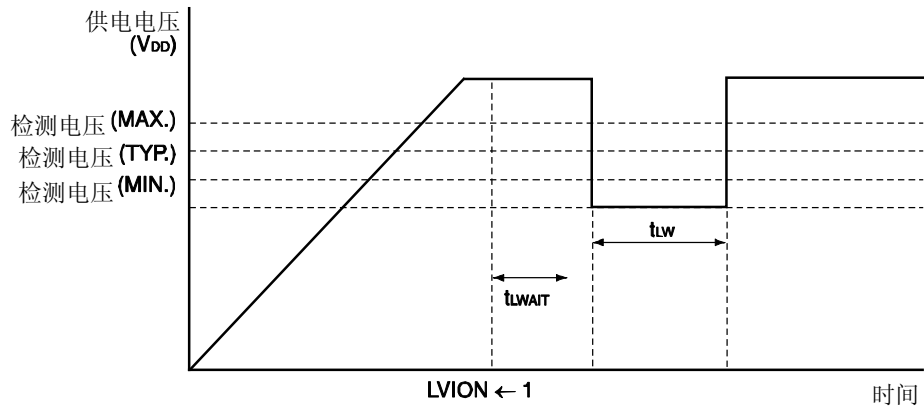
参数	符号	条件	MIN.	TYP.	MAX.	单位	
检测电压	供电电压电平	V_{LV10}		4.14	4.24	4.34	V
		V_{LV11}		3.99	4.09	4.19	V
		V_{LV12}		3.83	3.93	4.03	V
		V_{LV13}		3.68	3.78	3.88	V
		V_{LV14}		3.52	3.62	3.72	V
		V_{LV15}		3.37	3.47	3.57	V
		V_{LV16}		3.22	3.32	3.42	V
		V_{LV17}		3.06	3.16	3.26	V
		V_{LV18}		2.91	3.01	3.11	V
		V_{LV19}		2.75	2.85	2.95	V
		V_{LV10}		2.60	2.70	2.80	V
		V_{LV11}		2.45	2.55	2.65	V
		V_{LV12}		2.29	2.39	2.49	V
		V_{LV13}		2.14	2.24	2.34	V
		V_{LV14}		1.98	2.08	2.18	V
		V_{LV15}		1.83	1.93	2.03	V
外部输入引脚 ^{注 1}	EXLVI	$EXLVI < V_{DD}$, $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	1.11	1.21	1.31	V	
最短脉冲宽度	t_{LW}		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。

注 2. 从设置低电压检测寄存器(LVIM)的第 7 位(LVION)=1 到操作稳定所需的时间。

备注 $V_{LV1(n-1)} > V_{LV1n}$: $n = 1 \sim 15$

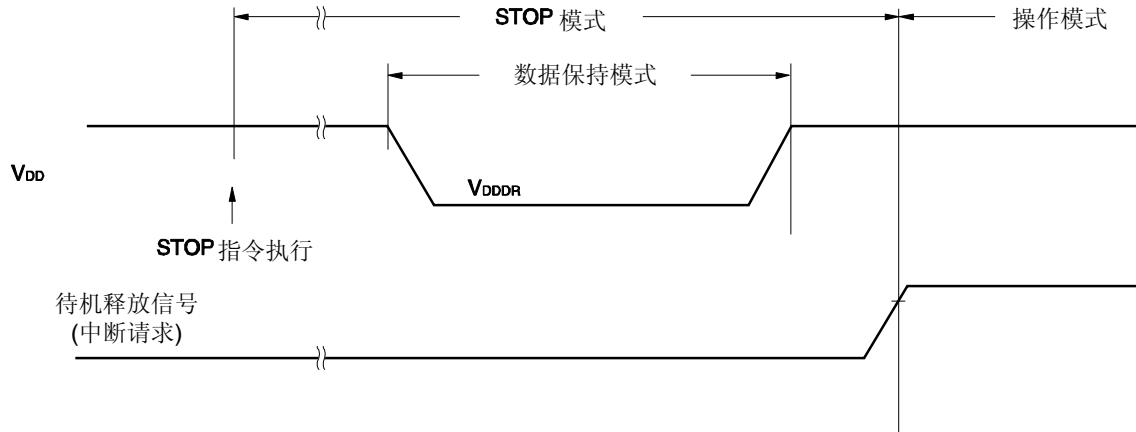
LVI 电路时序



数据存储器在 STOP 模式下低供电电压时的数据保持特性 ($T_A = -40 \sim +85^\circ\text{C}$)

参数	符号	条件	MIN.	TYP.	MAX.	单位
数据保持供电电压	V_{DDDR}		1.44 ^注		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

($T_A = -40 \sim +85^\circ\text{C}$, $2.7\text{ V} \leq V_{DD} = LV_{DD} \leq 5.5\text{ V}$, $V_{SS} = LV_{SS} = AV_{SS} = 0\text{ V}$)

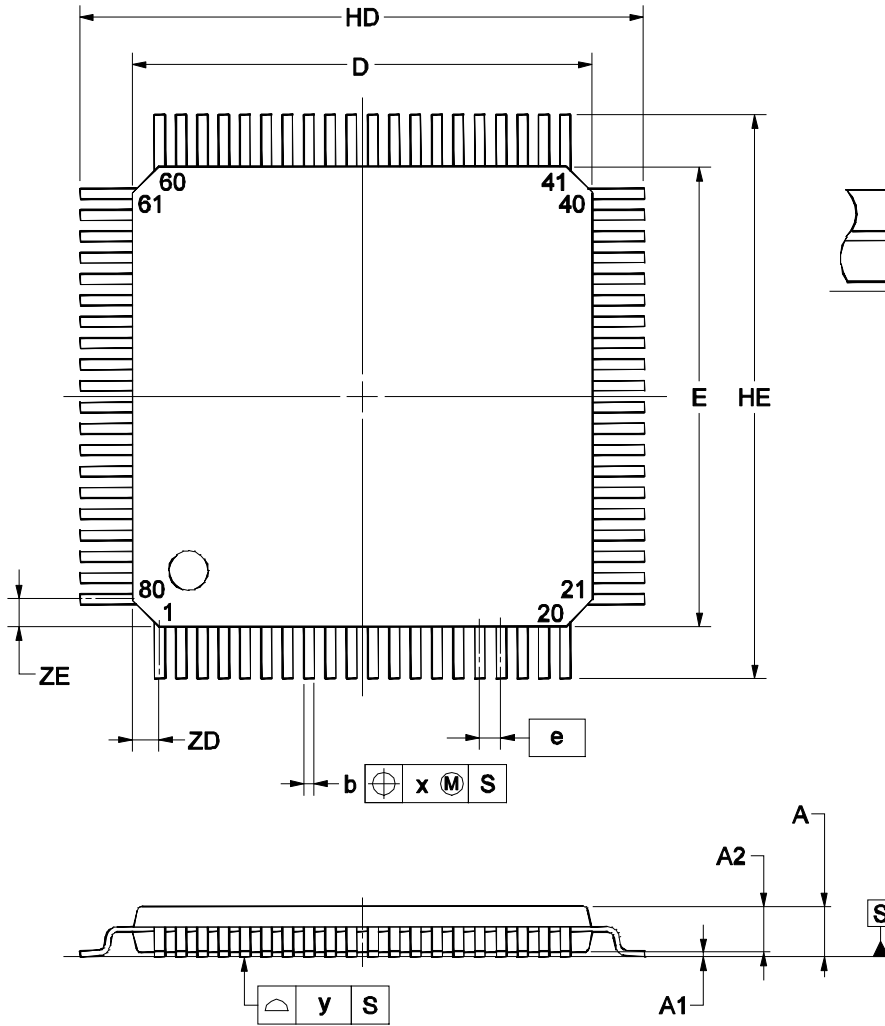
• 基本特性

参数	符号	条件	MIN.	TYP.	MAX.	单位
V_{DD} 供电电流	I_{DD}	$f_{XP} = 10\text{ MHz (TYP.)}, 20\text{ MHz (MAX.)}$		4.5	11.0	mA
擦除时间 ^{注1}	所有 block	T_{eraca}		20	200	ms
	Block 单元	T_{erasa}		20	200	ms
写时间(8 位)	T_{wrwa}			10	100	μs
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 ^{注3}	100			次数

注 1. 不包括擦除前的预写时间和擦除验证时间 (写回时间)。
2. 在出货后第一次对产品进行写操作时，“擦除 → 写”和“只写”作为一次重写。

备注 1. f_{XP} : 主系统时钟振荡频率
2. 串行写入操作特性，参阅 **78K0/Lx2 Flash 存储器编程(编程器)应用笔记(U18204E)**。

80 引脚塑封 LQFP(14×14)



引脚端详图

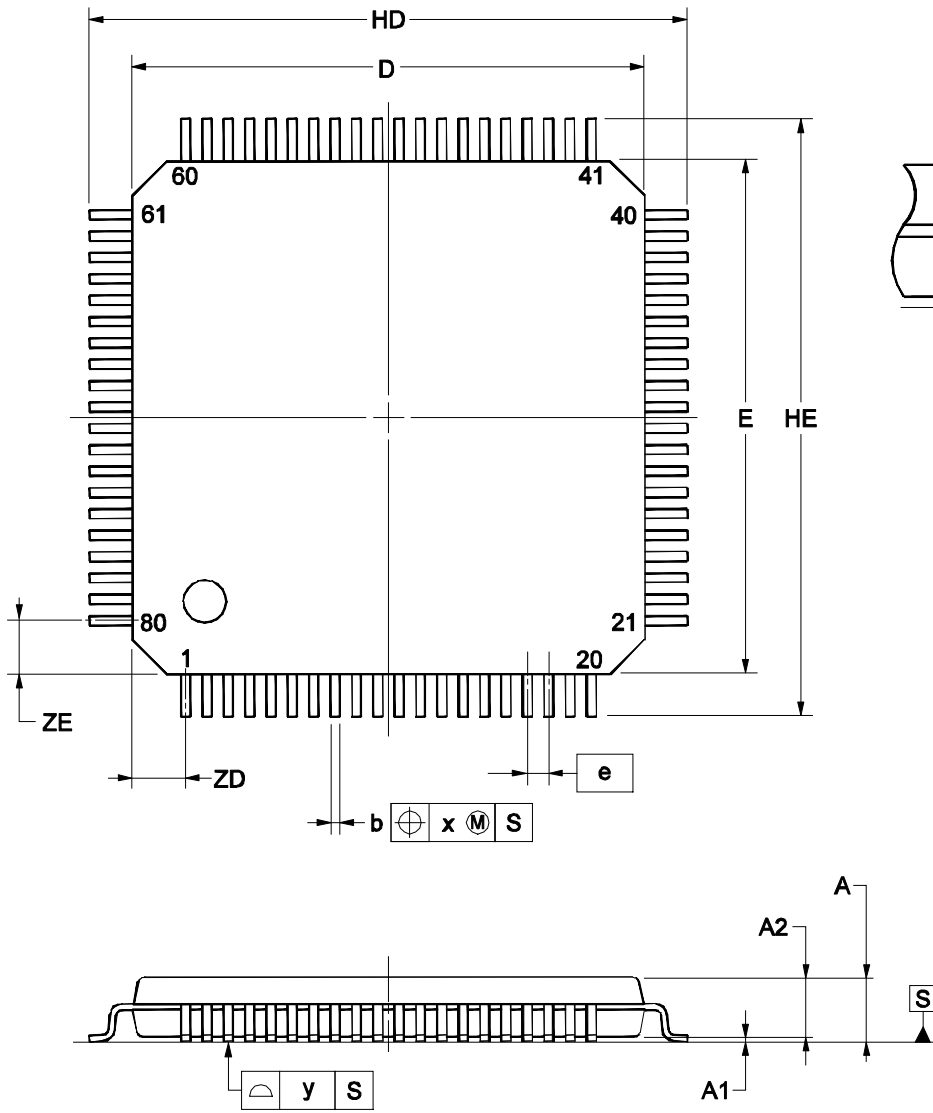
(单位: mm)

项目	尺寸
D	14.00±0.20□
E	14.00±0.20□
HD	17.20±0.20□
HE	17.20±0.20□
A	1.70 MAX.□
A1	0.125±0.075□
A2	1.40±0.05
A3	0.25
b	0.32±0.06
c	0.17 ^{+0.03} _{-0.06}
L	0.80
Lp	0.888±0.15
L1	1.60±0.20
θ	3° ^{+5°} _{-3°}
e	0.65
x	0.13□
y	0.10□
ZD	0.825□
ZE	0.825
P80GC-65-UBT	

注

该结构在最大使用材料情况下，每条引脚的中心线位于其实际位置的 0.13mm 内

80 引脚塑封 LQFP(密脚距)(12×12)



引脚端详图

(单位: mm)

项目	尺寸
D	12.00±0.20□
E	12.00±0.20□
HD	14.00±0.20□
HE	14.00±0.20□
A	1.60 MAX.□
A1	0.10±0.05□
A2	1.40±0.05
A3	0.25
b	0.22±0.05
c	0.145 ^{+0.055} _{-0.045}
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	3° ^{+5°} _{-3°}
e	0.50
x	0.08□
y	0.08□
ZD	1.25□
ZE	1.25

P80GK-50-8EU-1

注

该结构在最大使用材料情况下，每条引脚的中心线位于其实际位置的 0.08mm 内

第三十一章 等待注意事项

31.1 等待注意事项

本产品有两种内部系统总线。

一种是 CPU 总线，另一种是连接低速外围硬件的外设总线。

由于 CPU 总线时钟与外设总线时钟是异步的，因此如果对 CPU 的访问与对外围硬件的访问冲突时，可能会有不期望的非法数据产生。

因此当访问可能会产生冲突的外围硬件时，CPU 反复执行处理，直至出现正确数据。

这样，CPU 并不执行下一条指令，而是等待。如果出现这种情况，指令执行所需的时钟数会以等待的时钟数而增加（如需了解等待时钟数的详细内容，参见表 31-1）。在进行实时操作时必须要注意这一点。

31.2 产生等待的外围硬件

表 31-1 列出了当被 CPU 访问时发出等待请求的寄存器和 CPU 等待的时钟数。

表 31-1. 产生等待的寄存器和 CPU 等待时钟数

外围硬件	寄存器	访问方式	等待时钟的个数
串行接口 UART0	ASIS0	读	1 个时钟 (恒定)
串行接口 UART6	ASIS6	读	1 个时钟 (恒定)
串行接口 IIC0	IICS0	读	1 个时钟 (恒定)
A/D 转换器	ADM	写	1 ~ 5 个时钟 (当选择 $f_{AD} = f_{PRS}/2$ 时)
	ADS	写	1 ~ 7 个时钟 (当选择 $f_{AD} = f_{PRS}/3$ 时)
	ADPC	写	1 ~ 9 个时钟 (当选择 $f_{AD} = f_{PRS}/4$ 时)
	ADCR	读	2 ~ 13 个时钟 (当选择 $f_{AD} = f_{PRS}/6$ 时) 2 ~ 17 个时钟 (当选择 $f_{AD} = f_{PRS}/8$ 时) 2 ~ 25 个时钟 (当选择 $f_{AD} = f_{PRS}/12$ 时)
<p>上述时钟个数是指在选择相同的源时钟作为 f_{CPU} 和 f_{PRS} 时的值。可以根据以下表达式和条件计算等待时钟的个数。</p> <p><计算等待时钟数></p> <ul style="list-style-type: none"> 等待时钟数 = $\{(1/f_{AD}) \times 2 / (1/f_{CPU})\} + 1$ <ul style="list-style-type: none"> * 如果等待时钟的个数 ≤ 0.5 则将小数点以后的部分舍去，而如果等待时钟的个数 > 0.5，则进行四舍五入 <p>f_{AD}: A/D 转换时钟的频率 ($f_{PRS}/2 \sim f_{PRS}/12$)</p> <p>$f_{CPU}$: CPU 时钟频率</p> <p>$f_{PRS}$: 外围硬件时钟频率</p> <p>$f_{XP}$: 主系统时钟频率</p> <p><等待时钟的最大/最小个数所需的条件></p> <ul style="list-style-type: none"> 最大个数: CPU 的最高速度(f_{XP}), A/D 转换时钟的最低速度 ($f_{PRS}/12$) 最小个数: CPU 的最低速度($f_{SUB}/2$), A/D 转换时钟的最高速度($f_{PRS}/2$) 			

注意事项 当 CPU 工作在副系统时钟和外围硬件时钟停止的状态下，不要使用需等待请求的访问模式访问上述寄存器。

备注 这里的时钟是指 CPU 时钟 (f_{CPU})。

附录 A 开发工具

78K0/LF2 的系统开发中可以使用以下开发工具。

图 A-1 显示了开发工具的组成。

- **支持 PC98-NX 系列**

除非特别说明，IBM PC/AT™ 兼容机支持的产品与 PC98-NX 系列计算机是兼容的。当使用 PC98-NX 系列计算机时，参阅 IBM PC/AT 兼容机的使用说明。

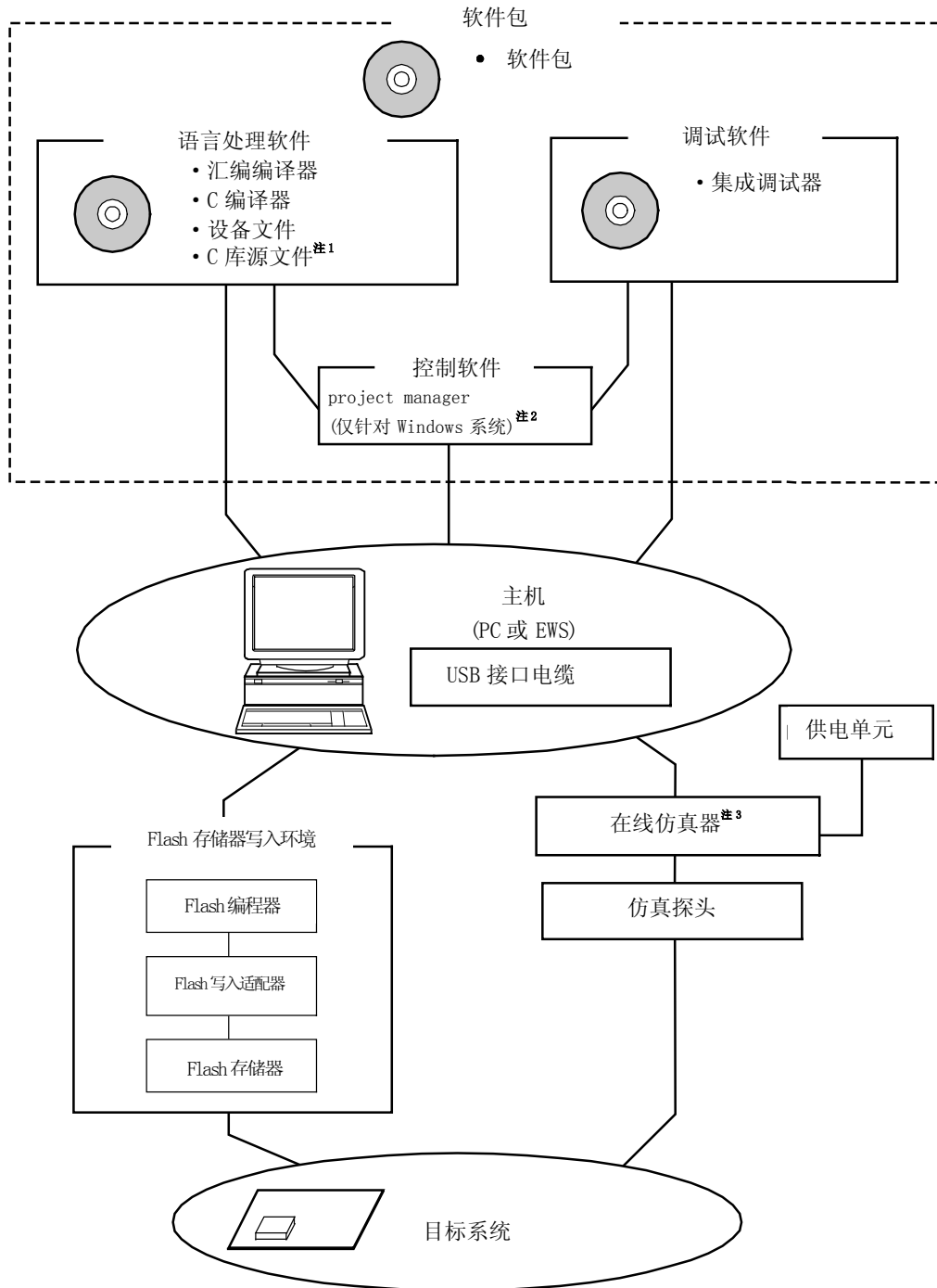
- **Windows™**

除非特别说明，“Windows”是指以下几种操作系统。

- Windows 98
- Windows NT™
- Windows 2000
- Windows XP

图 A-1 开发工具的组成(1/2)

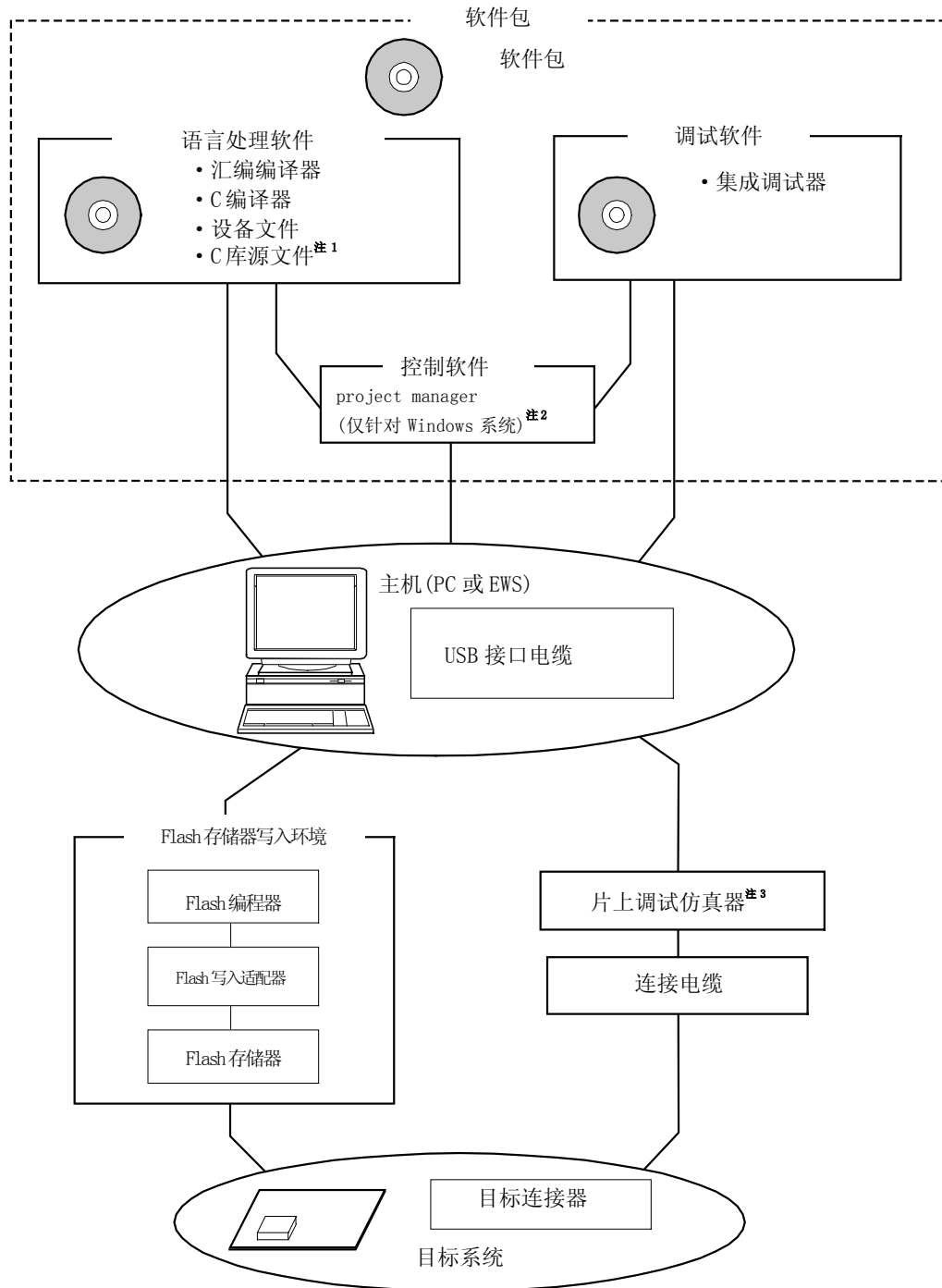
(1) 当使用在线仿真器 QB-78K0LX2 时



- 注
1. 软件包中不包括 C 库源文件。
 2. 汇编编译器中包括 project manager PM+ 。
PM+仅用于 Windows。
 3. 购买在线仿真器 QB-78K0LX2 时还提供集成调试器 ID78K0-QB、简易 flash 存储器编程器 PG-FPL3，
电源和 USB 接口电缆。其它产品均独立销售。

图 A-1 开发工具的组成(2/2)

(2) 当使用片上调试仿真器 QB-78K0MINI 时



- 注
1. 软件包中不包括 C 库源文件。
 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-78K0MINI 时还提供集成调试器 ID78K0-QB、USB 接口电缆和连接电缆。其它产品均独立销售。

A.1 软件包

SP78K0 78K/0 系列软件包	该软件包中的开发工具（软件）与 78k/0 系列相同 产品型号： μ SxxxxSP78K0
-----------------------	---

备注 产品型号中的 xxxx 随主机和使用的 OS 而变化。

μ SxxxxSP78K0

xxxx	主机	OS	存储介质
AB17	PC-9800 系列	Windows (日文版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英文版)	

A.2 语言处理软件

RA78K0 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。 该汇编编译器可以自动建立符号表和对转移指令进行优化。 该汇编编译器应当和设备文件 (DF780397) (独立销售)一起使用。</p> <p><PC 环境中使用 RA78K0 的注意事项> 该汇编编译器是基于 DOS 应用的。它也可以在 Windows 环境下使用，但必须在 Windows 环境下使用 Project Manager (包含在汇编编译器软件包中)。</p> <p>产品型号： μSxxxxRA78K0</p>
CC78K0 C 编译器	<p>该编译器将 C 程序转换成微控制器可执行的目标代码。 该编译器应当和汇编程序、设备文件(二者均独立销售)结合使用。</p> <p><PC 环境中使用 CC78K0 的注意事项> 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> <p>产品型号： μSxxxxCC78K0</p>
DF780397 ^{注 1} 设备文件	<p>该文件包含设备特有的信息。 该设备文件应当结合工具 (RA78K0、CC78K0 和 ID78K0-QB) (均为独立销售)一起使用。 相应的 OS 和主机随使用工具而变化。</p> <p>产品型号： μSxxxxDF780397</p>
CC78K0-L ^{注 2} C 库源文件	<p>这是一个函数的源文件，用来配置 C 编译器中的目标库。 该文件需要和用户规范说明中的目标库(包含在 C 编译器中)相匹配。</p> <p>产品型号： μSxxxxCC78K0-L</p>

- 注
1. DF780397 可以与 RA78K0、CC78K0 和 ID78K0-QB 一起使用。
 2. 软件包(SP78K0)不包括 CC78K0-L。

备注 产品型号中的(((随主机和使用的 OS 而变化。

μSxxxxRA78K0
μSxxxxCC78K0
μSxxxxCC78K0-L

xxxx	主机	OS	存储介质
AB17	PC-9800 系列	Windows (日文版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英文版)	
3P17	HP9000 系列 700™	HP-UJ™ (Rel. 10.10)	
3K17	SPARC 工作站™	SunOS™ (Rel. 4.1.4) Solaris™ (Rel. 2.5.1)	

μSxxxxDF780397

xxxx	主机	OS	存储介质
AB13	PC-9800 系列	Windows (日文版)	3.5 英寸 2HD FD
BB13	IBM PC/AT 兼容机	Windows (英文版)	

A.3 控制软件

PM+ Project manager	这是一个控制软件，可使用户在 Windows 环境下有效地进行软件开发。用户程序开发中进行的任何操作，如启动编辑器、构造程序和启动调试器，都可以由 project manager 执行。 <注意事项> project manager 包含在汇编编译器软件包 (RA78K0) 中。 仅在 Windows 中使用。
------------------------	---

<R>

A.4 Flash 存储器写入工具

PG-FP4, FL-PR4 Flash 存储器编程器	Flash 存储器编程器，专用于内置 Flash 存储器的微控制器。
PG-FPL3, FP-LITE3 简易 Flash 存储器编程器	简易 Flash 存储器编程器，专用于内置 Flash 存储器的微控制器。
FA-80GC-8BT-A FA-78F0376GC-UBT-MX FA-78F0386GC-UBT-MX FA-80GK-9EU-A FA-78F0376GK-8EU-MX FA-78F0386GK-8EU-MX Flash 存储器写入适配器	Flash 存储器写入适配器用于连接 Flash 存储器编程器 <ul style="list-style-type: none"> FA-80GC-8BT-A, FA-78F0376GC-UBT-MX, FA-78F0386GC-UBT-MX : 用于 80 引脚塑封 LQFP (GC-UBT 型) FA-80GK-9EU-A, FA-78F0376GK-8EU-MX, FA-78F0386GK-8EU-MX : 用于 80 引脚塑封 LQFP (GK-8EU 型)

备注 1. FL-PR4, FP-LITE3, FA-80GC-8BT-A, FA-78F0376GC-UBT-MX, FA-78F0386GC-UBT-MX, FA-80GK-9EU-A, FA-78F0376GK-8EU-MX 和 FA-78F0386GK-8EU-MX 都是 Naito Densei Machida Mfg. Co., Ltd 的产品。

电话: +81-45-475-4191 Naito Densei Machida Mfg. Co., Ltd.

2. 使用 Flash 存储器写入适配器的最新版本。

A.5 调试工具 (硬件)

A.5.1 当使用在线仿真器 QB-78K0LX2 时

QB-78K0LX2 ^注 在线仿真器	在使用 78K0/LX2 开发应用系统时，在线仿真器用于调试硬件和软件。它支持集成调试器(ID78K0-QB)。该仿真器应当与电源、仿真头及用于连接仿真器和主机的 USB 电缆结合使用。
QB-144-CA-01 引脚检测适配器	该适配器用于示波器等波形检测。
QB-144-EP-01S 仿真探头	这是一种可以灵活使用的仿真探头，用于连接在线仿真器和目标系统。
QB-80GC-EA-04T, QB-80GC-EA-05T, QB-80GK-EA-04T, QB-80GK-EA-05T 交换适配器	该适配器用于实现从在线仿真器到目标连接器的引脚转换。 <ul style="list-style-type: none"> • QB-80GC-EA-04T: 80 引脚塑封 LQFP (μPD78F037x GC-UBT 型) • QB-80GC-EA-05T: 80 引脚塑封 LQFP (μPD78F038x GC-UBT 型) • QB-80GK-EA-04T: 80 引脚塑封 LQFP (μPD78F037x GK-8EU 型) • QB-80GK-EA-05T: 80 引脚塑封 LQFP (μPD78F038x GK-8EU 型)
QB-80GC-YS-01T, QB-80GK-YS-01T 空间适配器	该适配器用于调整目标系统和在线仿真器之间的高度。 <ul style="list-style-type: none"> • QB-80GC-YS-01T: 80 引脚塑封 LQFP (GC-UBT 型) • QB-80GK-YS-01T: 80 引脚塑封 LQFP (GK-8EU 型)
QB-80GC-YQ-01T, QB-80GK-YQ-01T YQ 连接器	该连接器用于连接目标连接器和交换适配器。 <ul style="list-style-type: none"> • QB-80GC-YQ-01T: 80 引脚塑封 LQFP (GC-UBT 型) • QB-80GK-YQ-01T: 80 引脚塑封 LQFP (GK-8EU 型)
QB-80GC-HQ-01T, QB-80GK-HQ-01T 安装适配器	该适配器用于安装目标设备。 <ul style="list-style-type: none"> • QB-80GC-HQ-01T: 80 引脚塑封 LQFP (GC-UBT 型) • QB-80GK-HQ-01T: 80 引脚塑封 LQFP (GK-8EU 型)
QB-80GC-NQ-01T, QB-80GK-NQ-01T 目标连接器	该连接器被安装到目标系统上。 <ul style="list-style-type: none"> • QB-80GC-NQ-01T: 80 引脚塑封 LQFP (GC-UBT 型) • QB-80GK-NQ-01T: 80 引脚塑封 LQFP (GK-8EU 型)

注 使用 QB-78K0LX2 时同时提供集成调试器 ID78K0-QB、简易 Flash 存储器编程器 PG-FPL3、电源和 USB 接口电缆。

备注 封装内容随产品型号而变化，如下所示：

产品型号 \ 封装内容	在线仿真器	仿真头	交换适配器	YQ 连接器	目标连接器
QB-78K0LX2-ZZZ	QB-78K0LX2	无			
QB-78K0LX2-T80GC04		QB-144-EP-01S	QB-80GC-EA-04T	QB-80GC-YQ-01T	QB-80GC-NQ-01T
QB-78K0LX2-T80GC05			QB-80GC-EA-05T		
QB-78K0LX2-T80GK04		QB-144-EP-01S	QB-80GK-EA-04T	QB-80GK-YQ-01T	QB-80GK-NQ-01T
QB-78K0LX2-T80GK05			QB-80GK-EA-05T		

A.5.2 当使用片上调试仿真器 QB-78K0MINI 时

QB-78K0MINI ^注 片上调试仿真器	在使用 78K0/LX2 开发应用系统时，片上调试仿真器用于调试硬件和软件。它支持与 QB-78K0MINI 一起提供的集成调试器 (ID78K0-QB)。该仿真器使用一条连接电缆和一条用于连接主机的 USB 接口电缆。
目标连接器规范说明	10 引脚通用连接器 (2.54 mm 脚距)

注 使用 QB-78K0MINI 时同时提供集成调试器 ID78K0-QB，USB 接口电缆和连接电缆。

A.6 调试工具(软件)

ID78K0-QB 集成调试器	该调试器支持 78K/0 系列的在线仿真器。ID78K0-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区显示)来显示源程序的跟踪结果。该调试器应当结合设备文件(独立销售)一起使用。 产品型号： μS_{xxxx} ID78K0-QB
--------------------	--

备注 产品型号中的 xxxx 随主机和使用的 OS 不同而变化。

μS_{xxxx} ID78K0-QB

xxxx	主机	OS	存储介质
AB17	PC-9800 系列	Windows (日文版)	CD-ROM
BB17	IBM PC/AT 兼容机	Windows (英文版)	

附录 B 寄存器索引

B.1 寄存器索引 (按寄存器名称的字母顺序排列)

[A]

A/D 转换器模式寄存器(ADM)	255
A/D 端口配置寄存器(ADPC)	111, 261
模拟输入通道选择寄存器(ADS)	260
异步串行接口控制寄存器 6 (ASICL6)	308
异步串行接口操作模式寄存器 0 (ASIM0)	278
异步串行接口操作模式寄存器 6 (ASIM6)	302
异步串行接口接收错误状态寄存器 0 (ASIS0)	280
异步串行接口接收错误状态寄存器 6 (ASIS6)	304
异步串行接口发送状态寄存器 6 (ASIF6)	305

[B]

波特率发生器控制寄存器 0 (BRGC0)	281
波特率发生器控制寄存器 6 (BRGC6)	307

[C]

捕捉/比较控制寄存器 00 (CRC00)	161
时钟操作模式选择寄存器(OSCCTL)	118
时钟输出选择寄存器(CKS)	250, 437
时钟选择寄存器 6 (CKSR6)	306

[E]

8 位 A/D 转换结果寄存器(ADCRH)	259
8 位定时器比较寄存器 50 (CR50)	195
8 位定时器比较寄存器 51 (CR51)	195
8 位定时器计数器 50 (TM50)	195
8 位定时器计数器 51 (TM51)	195
8 位定时器 H 载波控制寄存器 1 (TMCYC1)	218
8 位定时器 H 比较寄存器 00 (CMP00)	213
8 位定时器 H 比较寄存器 01 (CMP01)	213
8 位定时器 H 比较寄存器 10 (CMP10)	213
8 位定时器 H 比较寄存器 11 (CMP11)	213
8 位定时器 H 模式寄存器 0 (TMHMD0)	214
8 位定时器 H 模式寄存器 1 (TMHMD1)	214
8 位定时器模式控制寄存器 50 (TMC50)	198
8 位定时器模式控制寄存器 51 (TMC51)	198
外部中断下降沿允许寄存器(EGN)	482
外部中断上升沿允许寄存器(EGP)	482

[I]

IIC 时钟选择寄存器 0 (IICCL0)	364
IIC 控制寄存器 0 (IICC0)	355
IIC 标志寄存器 0 (IICF0)	362

IIC 功能扩展寄存器 0 (IICX0)	365
IIC 移位寄存器 0 (IIC0)	352
IIC 状态寄存器 0 (IICS0)	360
输入切换控制寄存器(ISC)	310
内部扩展 RAM 容量切换寄存器(IXS).....	544
内部存储器容量切换寄存器(IMS).....	543
内部振荡模式寄存器(RCM)	122
中断屏蔽标志寄存器 0H (MK0H).....	480
中断屏蔽标志寄存器 0L (MK0L).....	480
中断屏蔽标志寄存器 1H (MK1H).....	480
中断屏蔽标志寄存器 1L (MK1L).....	480
中断请求标志寄存器 0H (IF0H)	478
中断请求标志寄存器 0L (IF0L)	478
中断请求标志寄存器 1H (IF1H)	478
中断请求标志寄存器 1L (IF1L)	478
[K]	
按键返回模式寄存器(KRM)	492
[L]	
LCD 时钟控制寄存器(LCDC).....	435
LCD 显示模式寄存器(LCDM)	434
LCD 模式设置寄存器(LCDMD).....	433
LCD 升压控制寄存器 0 (VLCG0).....	436
低电压检测等级选择寄存器(LVIS).....	524
低电压检测寄存器(LVIM)	523
[M]	
主时钟模式寄存器(MCM)	124
主 OSC 控制寄存器(MOC)	123
存储 bank 选择寄存器(BANK)	82
乘法/除法数据寄存器 A0 (MDA0H, MDA0L).....	465
乘法/除法数据寄存器 B0 (MDB0).....	466
乘法器/除法器控制寄存器 0 (DMUC0).....	467
[O]	
振荡稳定时间计数器的状态寄存器(OSTC).....	125, 494
振荡稳定时间选择寄存器(OSTS).....	126, 495
[P]	
端口模式寄存器 0 (PM0).....	108, 165
端口模式寄存器 1 (PM1).....	108, 200, 218, 282, 310, 336
端口模式寄存器 2 (PM2).....	108, 262
端口模式寄存器 3 (PM3).....	108, 200
端口模式寄存器 6 (PM6).....	108, 367
端口模式寄存器 7 (PM7).....	108
端口模式寄存器 12 (PM12)	108, 525
端口模式寄存器 14 (PM14)	251, 439

端口寄存器 0 (P0).....	109
端口寄存器 1 (P1).....	109
端口寄存器 2 (P2).....	109
端口寄存器 3 (P3).....	109
端口寄存器 6 (P6).....	109
端口寄存器 7 (P7).....	109
端口寄存器 12 (P12).....	109
端口寄存器 13 (P13).....	439
预分频器模式寄存器 00 (PRM00).....	163
预分频器模式寄存器 01 (PRM01).....	163
优先级指定标志寄存器 0H (PROH).....	481
优先级指定标志寄存器 0L (PROL).....	481
优先级指定标志寄存器 1H (PR1H).....	481
优先级指定标志寄存器 1L (PR1L).....	481
处理器时钟控制寄存器(PCC).....	120
上拉电阻选择寄存器 0 (PU0).....	110
上拉电阻选择寄存器 1 (PU1).....	110
上拉电阻选择寄存器 3 (PU3).....	110
上拉电阻选择寄存器 7 (PU7).....	110
上拉电阻选择寄存器 12 (PU12).....	110
[R]	
接收缓冲寄存器 0 (RXB0).....	277
接收缓冲寄存器 6 (RXB6).....	301
余数数据寄存器 0 (SDR0).....	465
复位控制标志寄存器(RESF).....	514
[S]	
串行时钟选择寄存器 10 (CSIC10).....	335
串行 I/O 移位寄存器 10 (SIO10).....	333
串行操作模式寄存器 10 (CSIM10).....	334
16 位定时器捕捉/比较寄存器 000 (CR000).....	154
16 位定时器捕捉/比较寄存器 001 (CR001).....	157
16 位定时器捕捉/比较寄存器 010 (CR010).....	156
16 位定时器捕捉/比较寄存器 011 (CR011).....	157
16 位定时器计数器 00 (TM00).....	154
16 位定时器计数器 01 (TM01).....	154
16 位定时器模式控制寄存器 00 (TMC00).....	158
16 位定时器模式控制寄存器 01 (TMC01).....	158
16 位定时器输出控制寄存器 00 (TOC00).....	161
从地址寄存器 0 (SVA0).....	352
[T]	
10 位 A/D 转换结果寄存器(ADCR).....	258
定时器时钟选择寄存器 50 (TCL50).....	196
定时器时钟选择寄存器 51 (TCL51).....	196
发送缓冲寄存器 10 (SOTB10).....	333

发送缓冲寄存器 6 (TXB6).....	301
发送移位寄存器 0 (TXS0).....	277

[W]

钟表定时器操作模式寄存器(WTM).....	237
看门狗定时器允许寄存器(WDTE).....	243

B.2 寄存器索引(按寄存器符号的字母顺序排列)**[A]**

ADCR: 10 位 A/D 转换结果寄存器.....	258
ADCRH: 8 位 A/D 转换结果寄存器.....	259
ADM: A/D 转换器模式寄存器.....	255
ADPC: A/D 端口配置寄存器.....	111, 261
ADS: 模拟输入通道选择寄存器.....	260
ASICL6: 异步串行接口控制寄存器 6.....	308
ASIF6: 异步串行接口发送状态寄存器 6.....	305
ASIM0: 异步串行接口操作模式寄存器 0.....	278
ASIM6: 异步串行接口操作模式寄存器 6.....	302
ASIS0: 异步串行接口接收错误状态寄存器 0.....	280
ASIS6: 异步串行接口接收错误状态寄存器 6.....	304

[B]

BANK: 存储 bank 选择寄存器.....	82
BRGC0: 波特率发生器控制寄存器 0.....	281
BRGC6: 波特率发生器控制寄存器 6.....	307

[C]

CKS: 时钟输出选择寄存器.....	250, 437
CKSR6: 时钟选择寄存器 6.....	306
CMP00: 8 位定时器 H 比较寄存器 00.....	213
CMP01: 8 位定时器 H 比较寄存器 01.....	213
CMP10: 8 位定时器 H 比较寄存器 10.....	213
CMP11: 8 位定时器 H 比较寄存器 11.....	213
CR000: 16 位定时器捕捉/比较寄存器 000.....	154
CR001: 16 位定时器捕捉/比较寄存器 001.....	157
CR010: 16 位定时器捕捉/比较寄存器 010.....	156
CR011: 16 位定时器捕捉/比较寄存器 011.....	157
CR50: 8 位定时器比较寄存器 50.....	195
CR51: 8 位定时器比较寄存器 51.....	195
CRC00: 捕捉/比较控制寄存器 00.....	161
CSIC10: 串行时钟选择寄存器 10.....	335
CSIM10: 串行操作模式寄存器 10.....	334

[D]

DMUC0: 乘法器/除法器控制寄存器 0.....	467
----------------------------	-----

[E]

EGN:	外部中断下降沿允许寄存器	482
EGP:	外部中断上升沿允许寄存器	482

[I]

IF0H:	中断请求标志寄存器 0H.....	478
IF0L:	中断请求标志寄存器 0L	478
IF1H:	中断请求标志寄存器 1H.....	478
IF1L:	中断请求标志寄存器 1L	478
IIC0:	IIC 移位寄存器 0.....	352
IICC0:	IIC 控制寄存器 0.....	355
IICCL0:	IIC 时钟选择寄存器 0.....	364
IICF0:	IIC 标志寄存器 0.....	362
IICS0:	IIC 状态寄存器 0.....	360
IICX0:	IIC 功能扩展寄存器 0.....	365
IMS:	内部存储器容量切换寄存器	543
ISC:	输入切换控制寄存器	310
IXS:	内部扩展 RAM 容量切换寄存器	544

[K]

KRM:	按键返回模式寄存器	492
------	-----------------	-----

[L]

LCDC:	LCD 时钟控制寄存器	435
LCDM:	LCD 显示模式寄存器	434
LCDMD:	LCD 模式设置寄存器	433
LVIM:	低电压检测寄存器.....	523
LVIS:	低电压检测等级选择寄存器	524

[M]

MCM:	主时钟模式寄存器.....	124
MDA0H:	乘法/除法数据寄存器 A0.....	465
MDA0L:	乘法/除法数据寄存器 A0.....	465
MDB0:	乘法/除法数据寄存器 B0.....	466
MK0H:	中断屏蔽标志寄存器 0H.....	480
MK0L:	中断屏蔽标志寄存器 0L	480
MK1H:	中断屏蔽标志寄存器 1H.....	480
MK1L:	中断屏蔽标志寄存器 1L	480
MOC:	主 OSC 控制寄存器	123

[O]

OSCCTL:	时钟操作模式选择寄存器	118
OSTC:	振荡稳定时间计数器的状态寄存器.....	125, 494
OSTS:	振荡稳定时间选择寄存器	126, 495

[P]

P0:	端口寄存器 0.....	109
P1:	端口寄存器 1.....	109
P2:	端口寄存器 2.....	109

P3:	端口寄存器 3	109
P6:	端口寄存器 6	109
P7:	端口寄存器 7	109
P12:	端口寄存器 12	109
P13:	端口寄存器 13	439
PCC:	处理器时钟控制寄存器	120
PM0:	端口模式寄存器 0	108, 165
PM1:	端口模式寄存器 1	108, 200, 218, 282, 310, 336
PM2:	端口模式寄存器 2	108, 262
PM3:	端口模式寄存器 3	108, 200
PM6:	端口模式寄存器 6	108, 367
PM7:	端口模式寄存器 7	108
PM12:	端口模式寄存器 12	108, 525
PM14:	端口模式寄存器 14	251, 439
PR0H:	优先级指定标志寄存器 0H	481
PR0L:	优先级指定标志寄存器 0L	481
PR1H:	优先级指定标志寄存器 1H	481
PR1L:	优先级指定标志寄存器 1L	481
PRM00:	预分频器模式寄存器 00	163
PRM01:	预分频器模式寄存器 01	163
PU0:	上拉电阻选择寄存器 0	110
PU1:	上拉电阻选择寄存器 1	110
PU3:	上拉电阻选择寄存器 3	110
PU7:	上拉电阻选择寄存器 7	110
PU12:	上拉电阻选择寄存器 12	110
[R]		
RCM:	内部振荡模式寄存器	122
RESF:	复位控制标志寄存器	514
RXB0:	接收缓冲寄存器 0	277
RXB6:	接收缓冲寄存器 6	301
[S]		
SDR0:	余数数据寄存器 0	465
SIO10:	串行 I/O 移位寄存器 10	333
SOTB10:	发送缓冲寄存器 10	333
SVA0:	从地址寄存器 0	352
[T]		
TCL50:	定时器时钟选择寄存器 50	196
TCL51:	定时器时钟选择寄存器 51	196
TM00:	16 位定时器计数器 00	154
TM01:	16 位定时器计数器 01	154
TM50:	8 位定时器计数器 50	195
TM51:	8 位定时器计数器 51	195
TMC00:	16 位定时器模式控制寄存器 00	158
TMC01:	16 位定时器模式控制寄存器 01	158

TMC50:	8 位定时器模式控制寄存器 50	198
TMC51:	8 位定时器模式控制寄存器 51	198
TMCYC1:	8 位定时器 H 载波控制寄存器 1	218
TMHMD0:	8 位定时器 H 模式寄存器 0	214
TMHMD1:	8 位定时器 H 模式寄存器 1	214
TOC00:	16 位定时器输出控制寄存器 00	161
TXB6:	发送缓冲寄存器 6	301
TXS0:	发送移位寄存器 0	277
[V]		
VLCG0:	LCD 升压控制寄存器 0	436
[W]		
WDTE:	看门狗定时器允许寄存器	243
WTM:	钟表定时器操作模式寄存器	237

C.1 本版本的主要修订

(1/2)

页码	描述	分类
全篇	增加了下列产品 PD78F0374, 78F0376, 78F0376D, 78F0384, 78F0386, 78F0386D 删除了下列产品 PD78F0375D, 78F0385D	(d)
	增加 P60 和 P61 引脚, 端口模式寄存器 6 (PM6) 和端口寄存器 6 (P6)	(b)
	扩展了电容的范围 (“0.47 μ F: 目标” \rightarrow “0.47 ~ 1 μ F: 推荐)	(b)
	增加了存储器 bank 选择寄存器	(b)
第一章 概述		
p. 18	从 1.1 特性中删除了关于产品处理部分管理的描述	(d)
p. 19	更改 1.3 订购信息	(d)
p. 23	更改 1.5 78K0/Lx2 系列介绍	(d)
p. 25	更改 1.6 框图	(d)
p. 26	更改了 1.7 功能概述 (μ PD78F037x)	(d)
p. 28	更改了 1.8 功能概述 (μ PD78F038x)	(d)
第三章 CPU 结构		
p. 44	更改和增加注 1 到表 3-1 内部存储器容量切换寄存器 (IMS) 和内部扩展 RAM 容量切换寄存器 (IXS) 的设定值	(c, d)
p. 47	增加了图 3-3 存储空间映射图 (μ PD78F0374, 78F0384)	(d)
p. 49	增加了图 3-5 存储空间映射图 (μ PD78F0376, 78F0386)	(d)
p. 50	增加了图 3-6 存储空间映射图 (μ PD78F0376D, 78F0386D)	(d)
p. 51	更改了表 3-2 内部 ROM 容量	(d)
p. 52	更改了表 3-4 内部高速 RAM 容量	(d)
p. 53	更改了表 3-2 内部扩展 RAM 容量	(d)
p. 56	增加了图 3-9 数据存储器和地址之间的关系 (μ PD78F0374, 78F0384)	(d)
p. 58	增加了图 3-11 数据存储器和地址之间的关系 (μ PD78F0376, 78F0376D, 78F0386, 78F0386D)	(d)
p. 68	增加了注 3 到表 3-7 特殊功能寄存器表 (4/4)	(c)
p. 81	增加了 3.5 存储器 bank (仅用于 μ PD78F0376, 78F0376D, 78F0386 和 78F0386D)	(b)
p. 82	增加了 3.6 存储器 bank 选择寄存器 (BANK) (仅用于 μ PD78F0376, 78F0376D, 78F0386, 和 78F0386D)	(b)
p. 83	增加了 3.7 选择存储器 bank (仅用于 μ PD78F0376, 78F0376D, 78F0386 和 78F0386D)	(b)
第十一章 时钟输出控制器		
p. 250	更改了图 11-2 时钟输出选择寄存器 (CKS) 的格式	(b)
第十五章 串行接口 CSI10		
p. 335	更改了图 15-3 串行时钟选择寄存器 10 (CSIC10) 的格式	(b)

备注

在上表中的“分类”按如下区分。

(a): 错误修正。(b): 增加/更改说明。(c): 增加/更改描述或注。

(d): 增加/更改封装, 产品型号, 或管理分类。(e): 增加/更改相关文档。

页码	描述	分类
第十七章 LCD 控制器/驱动器		
p. 436	更改了 图 17-8 LCD 电压升压控制寄存器 0 的格式	(a)
p. 437	增加了 注意事项 1 到 图 17-9 时钟输出选择寄存器的格式	(c)
p. 461	更改了 图 17-29 LCD 驱动电路(外部电阻分压模式)的连接示例	(c)
第二十二章 复位功能		
p. 511	增加了 注 5 到 表 22-2 复位响应后的硬件状态(1/3)	(c)
第二十六章 Flash 存储器		
p. 543	更改了 图 26-1 内部存储器容量切换寄存器 (IMS) 的格式	(d)
p. 543	更改和增加了 注 1 到 表 26-1 内部存储器容量切换寄存器的设置	(c, d)
p. 544	更改了 图 26-2 内部扩展 RAM 容量切换寄存器 (IXS) 的格式	(d)
p. 544	更改了 表 26-2 内部扩展 RAM 容量切换寄存器的设置	(d)
p. 552	更改了 图 26-10 FLMD0 引脚连接示例	(b)
第二十九章 电气特性		
p. 580	更改目标规范为正式规范	(b)
附录 A 开发工具		
p. 609	增加了 FA-78F0376GC-UBT-MX, FA-78F0386GC-UBT-MX, FA-78F0376GK-8EU-MX 和 FA-78F0386GK-8EU-MX 到 A.4 Flash 存储器写入工具	(d)
附录 C 版本历史		
p. 620	增加了此章	(c)

备注 在上表中的“分类”按如下区分。

(a): 错误修正。(b): 增加/更改说明。(c): 增加/更改描述或注。

(d): 增加/更改封装, 产品型号, 或管理分类。(e): 增加/更改相关文档。

详细信息请联系:

(中国区)

网址:

<http://www.cn.necel.com/>

<http://www.necel.com/>

[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