

第 2 章 中断接口技术

本章学习目标

本章介绍中断技术原理及实现。重点介绍中断控制器 8259A，给出该芯片详细的编程结构。通过对本章的学习，读者应该掌握以下主要内容：

- 8086 中断系统的分类
- 8259A 主要功能与结构特点
- 8259A 的中断响应过程
- 8259A 的控制字和命令字

2.1 8086 的中断系统

中断（interrupt）技术是现代计算机系统中一种十分重要的功能，是计算机与外部设备之间进行信息交换的一种方式，其最初的引入是为了解决高速运算的 CPU 与低速外部设备之间信息交换的速度匹配问题。随着计算机技术的发展，中断技术不断被赋予新的功能，例如 CPU 外部与内部紧急事件的处理、机器故障的处理、实时控制等多个方面，并产生了软件方法进入中断处理的概念，即软中断。

2.1.1 中断与中断系统功能

1. 中断的概念

中断是一个过程，是 CPU 在正常执行程序的过程中，当遇到外部或内部的紧急事件需要处理时，暂时中断（中止）当前程序的执行，而转去为紧急事件服务，当为这个紧急事件服务处理完毕之后，再返回到暂停点（中断点）继续执行原来的程序，如图 2-1 所示。由于 CPU 正在执行的源程序被暂停执行，所以称为中断，被中断的源程序也叫主程序；

引起中断的原因或发出中断请求的设备称为中断源；为中断事件服务的程序称为中断服务子程序或中断处理程序。此外，还可以用软件方法来引起中断，即事先在程序中安排特殊的指令，CPU 在执行到该指令时，转去执行相应的一段预先安排好的程序，在执行完这个程序后，再返回去执行原来的程序，这称为软中断。主程序被中止的地方，称为断点，即下一条指令所在内存的地址。中断服务子程序一般存放在内存中一个固定的区域内，其起始地址称为中断服务子程序的入口地址。

2. 中断系统的功能

为实现中断而设置的各种中断控制逻辑和管理相应中断的指令有机组合后称为中断系统。中断系统为了实现规定的中断控制，一般应具有以下功能：

- (1) 中断处理与返回。当某个中断源发出中断请求时，CPU 要根据当前条件来决定是否

响应该中断请求，若 CPU 正在处理的工作比发出请求的中断源更重要时，可暂时不响应中断；如果需要响应该中断请求，CPU 必须在执行完主程序当前指令后，保护好断点和现场，然后转去执行中断服务程序，中断处理完毕后再恢复现场和断点，实现中断返回，使 CPU 回到主程序继续执行下一条指令。

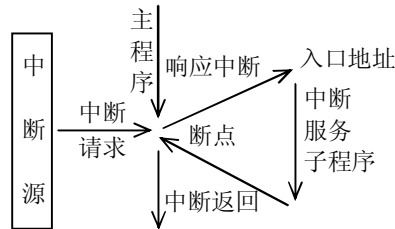


图 2-1 中断过程示意图

(2) 中断优先权判断。如果出现多个中断源同时提出中断请求时，CPU 要确定先为哪一个中断源服务。采用的方法是给每个中断源确定一个中断优先级别，中断系统能够自动地对多个中断源进行排队判优，首先处理优先级别高的中断请求，处理完毕后，再响应中断级别较低的中断请求。

(3) 中断嵌套。中断嵌套是指当 CPU 正在响应某一中断请求并为其服务时，若有优先权更高的中断源发出了中断请求，CPU 会终止当前正在执行的中断服务程序，而响应级别更高的中断请求；当处理完高级别的中断请求服务之后，再返回当前被终止的中断服务程序继续执行。

2.1.2 中断分类

在微机系统中中断可分为两类：硬件中断和软件中断，如图 2-2 所示。

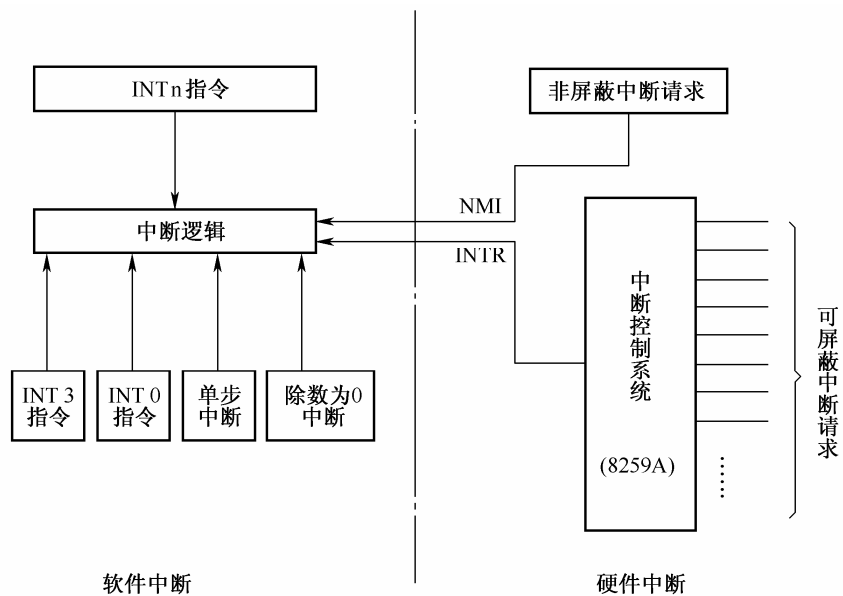


图 2-2 中断分类

1. 硬件中断

硬件中断是指通过外部硬件产生的中断，如打印机、键盘等，有时也称为外部中断。硬件中断又可分为两类：可屏蔽中断和不可屏蔽中断。

(1) 不可屏蔽中断：由 NMI 引脚引入，不受中断允许标志的影响，主要用来处理紧急情况，例如掉电处理。这种中断一旦发生，系统会立即响应。

(2) 可屏蔽中断：由 INTR 引脚引入，受中断允许标志的影响，也就是说，只有当 IF=1 时，可屏蔽中断才能进入；反之则不允许进入。可屏蔽中断可有多个，一般是通过优先级排队，从多个中断源中选出一个进行处理。

2. 软件中断（内部中断）

根据某条指令或对标志寄存器中某个标志的设置而产生，它与硬件电路无关。在 8086 系统中，内部中断主要有以下几种：

(1) 除法出错中断：执行除法指令时若发现除数为 0 或商超过了目的寄存器所能表达的范围，CPU 会立即产生一个中断类型码为 0 的内部中断，该中断称为除法出错中断，该中断的服务处理由操作系统安排。

(2) INT 0 溢出中断：若算术运算结果使溢出标志位 OF=1，则执行 INT0 指令后立即产生一个中断类型码为 4 的内部中断。

(3) INT n 指令：这是 8086 指令系统中的中断指令，其中 n 为中断类型码（范围为 0~255）。CPU 每执行一条这种指令就会发生一次中断。用户可以用 INT n 指令方便地调用不同类型码所代表的中断服务程序。

(4) 断点中断：8086CPU 的指令系统中有一条用于程序调试的中断指令 INT3。该指令会产生一个中断类型码为 3 的内部中断，当程序执行到该指令时，CPU 会转去执行断点中断服务程序，以进行某些特定的检查和处理。

(5) 单步中断：标志寄存器中的 TF=1 时，CPU 就处于单步工作方式。CPU 在每条指令执行完后自动产生中断类型码为 1 的内部中断，把标志寄存器的内容和断点压入堆栈，然后将 TF 和 IF 清零。单步中断过程结束时，由中断返回指令 IRET 从堆栈中将原来保存的标志内容弹出堆栈，恢复到标志寄存器中，此时 TF=1，使 CPU 返回单步工作方式，当下一条指令执行后又产生新的单步中断。单步中断是一种常用的调试工具。

2.1.3 中断向量和中断向量表

系统处理中断的方法很多，处理中断的步骤中最主要的一步就是如何根据不同的中断源进入相应的中断服务子程序，目前用得最多的就是中断向量的方法。

中断向量：把各个中断服务子程序的入口都称为一个中断向量。

中断向量表：将这些中断向量按一定的规律排列成一张表，就是所谓的中断向量表，当中断源发出中断请求时，即可查找该表，找出其中断向量，就可转入相应的中断服务子程序。

8086/8088 中断系统中的中断向量表是位于 0 段的 0~3FFFH 的存储区内，每个中断向量占四个单元，其中前两个单元存放中断处理子程序的入口地址的偏移地址（IP），低位在前，高位在后；后两个单元存放中断处理子程序入口地址的段地址（CS），也是低位在前，高位在后，整个中断向量的排列是按中断类型号进行的，如图 2-3 所示。

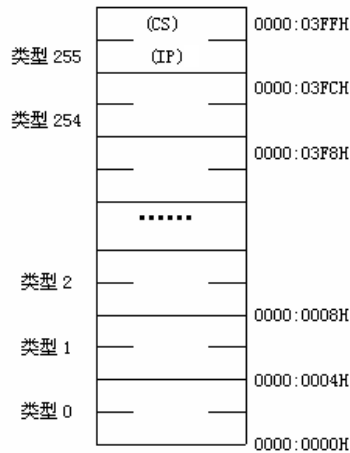


图 2-3 中断向量表

图 2-3 给出了中断类型码与中断向量所在位置之间的对应关系。其中 00H~04H 为专用中断；05H~3FH 为系统保留中断，用户一般是不能对它们定义的（这里面有一些为固定的用途，如 INT 21H 即为 MS-DOS 的系统调用）；40H~FFH 为用户定义的中断。详细分类如下所示：

00H~04H——系统专用	10H~1FH——BIOS 用	40H~FFH——用户用
08H~0FH——硬件中断	20H~3FH——DOS 用	

中断类型号的值乘以 4 即可计算出某个中断类型的中断向量在整个中断向量表中的位置。如类型号为 20H，则中断向量的存放位置为 $20H \times 4 = 80H$ ，设中断服务子程序的入口地址为 4030:2010H，则在 0000:0080H~0000:0083H 单元中就应该顺序放入 10H、20H、30H 和 40H。当系统响应 20H 号中断时，会自动查找中断向量表，找出对应的中断向量装入 CS 和 IP，即转入该中断服务子程序。

2.1.4 中断响应过程与时序

8086/8088 对软件中断和硬件中断响应的过程是不同的，这是由于软件中断和硬件中断所产生的原因不同，下面主要讨论硬件中断的情况。

1. 硬件中断的响应过程

硬件中断指的是由 NMI 引脚进入的非屏蔽中断或由 INTR 引脚进入的可屏蔽中断。下面以可屏蔽中断为例来说明硬件中断的响应过程。

CPU 在 INTR 引脚上接到一个中断请求信号，如果此时 $IF=1$ ，CPU 就会在当前指令执行完之后开始响应外部的中断请求，这时，CPU 在 \overline{INTA} 引脚连续发两个负脉冲，外设接到第二个负脉冲以后，在数据线上发送中断类型码，接到这个中断类型码后，CPU 做如下动作：

- (1) 将中断类型码放入暂存器保存。
- (2) 将标志寄存器内容压入堆栈，以保护中断时的状态。

(3) 将 IF 和 TF 标志清 0。将 IF 清 0 的目的是防止在中断响应的同时又有去响应新的中断请求；而将 TF 清 0 是为了防止 CPU 以单步方式执行中断处理子程序。这里要特别说明的是当 CPU 在中断响应时自动关闭了 IF 标志，因此用户如要进行中断嵌套时，必须在自己的中断处理子程序中用开中断指令来重新设置 IF。

(4) 保护断点。断点指的是在响应中断时，主程序当前指令下面的一条指令的地址。因此保护断点的动作就是将当前的 IP 和 CS 的内容压入堆栈，其目的是为了以后能够正确地返回主程序。

(5) 根据取到的中断类型码，在中断向量表中找出对应的中断向量，将其装入 IP 和 CS，即自动转向中断服务子程序。

对 NMI 进入的中断请求，由于其类型码固定为 2，因此 CPU 不用从外设读取类型码，也不需计算中断向量表的地址，只要将中断向量表中 0000:0008H~0000:000BH 单元内容分别装入 IP 和 CS 即可。

图 2-4 给出了 8086/8088 中断响应过程的流程图，下面做几点说明：

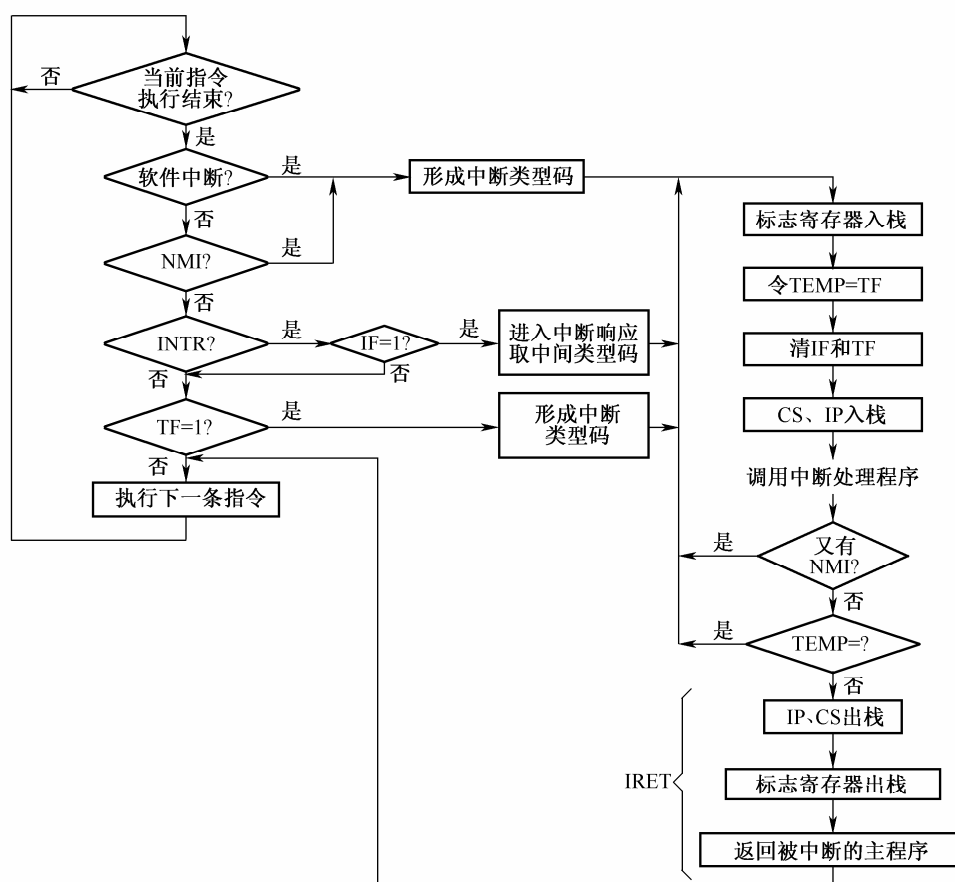


图 2-4 中断响应流程图

(1) 8086/8088 除软件中断外，内部“非屏蔽中断”、“可屏蔽中断”均设有优先级，其中内部中断（除单步外）——即 0、1、3 和 4 号中断的优先级高于非屏蔽中断，非屏蔽中断高于可屏蔽中断，单步中断优先级最低。

(2) 只有在可屏蔽中断的情况下才判断 IF=1 是否成立，才取中断类型码，其余的没有这个动作。

(3) 单步中断是每执行一条指令中断一次，显示出当时各寄存器的内容，供用户参考；

当进入单步中断响应时，CPU 自动清除 TF，在中断返回后，由于恢复了响应时的标志寄存器的值，因此 TF=1，执行完一条指令后又进入单步中断，直到程序将 TF 改为 0 为止。

(4) 关于中断的嵌套，NMI 总是可以响应的，若在中断处理子程序中设立了开中断指令，INTR 的请求也能响应。

(5) 弹出 IP、CS 和各标志位的值，并返回到断点的动作由 IRET 指令完成。

(6) 有些情况下，即使条件满足，CPU 也不能马上响应中断，必须执行完下一条指令（而不是当前指令）才行。例如：

1) 正好执行 LOCK 指令。

2) 执行向 SS 寄存器赋值的传送指令，因为一般要求连续用两条指令对 SS 和 SP 寄存器赋值，以保证堆栈指针的正确性。

(7) 当遇到等待指令或串操作指令时，允许在指令执行的过程中进入中断。这时需要在中断处理子程序中保护现场，以保证中断返回后能继续正确地执行这些指令。

2. 硬件中断的时序

如图 2-5 所示给出了 8086/8088 中断响应总线周期时序，其说明如下：

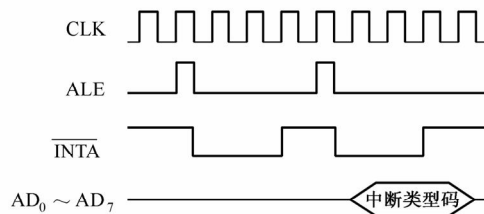


图 2-5 8086/8088 中断响应总线周期

(1) 要求 INTR 信号是一个高电平信号，并且维持 2 个时钟周期，因为 CPU 是在一条指令的最后一个时钟周期采样 INTR，进入中断响应后，它在第一个总线周期的 T1 仍需采样 INTR。

(2) 当 8086 工作在最小模式时，从 $\overline{\text{INTA}}$ 发出中断响应信号；工作在最大模式时，则是通过 $\overline{\text{S2}}$ 、 $\overline{\text{S1}}$ 、 $\overline{\text{S0}}$ 的组合完成的。

(3) 第一个总线周期用来通知外设，CPU 准备响应中断，第二个总线周期接收外设发回的中断类型码，该类型码必须通过 16 位数据总线的低 8 位传送。

(4) 在中断响应总线周期， $\overline{\text{M}}/\overline{\text{IO}}$ 为低，数据/地址线悬空， $\overline{\text{BHE}}/\text{S7}$ 和地址/状态线均悬空，中间的两个空闲周期也可为一个或没有（8088）。

3. 软件中断

软件中断是指通过中断指令来使 CPU 执行中断处理子程序的方法，其特点如下：

(1) 中断类型码由指令提供，不需执行中断响应总线周期，也不受 IF 标志的影响。

(2) 正在执行软件中断时，若有通过 NMI 发出的中断请求，则立即响应；若发生可屏蔽中断请求，只要条件允许（如 IF=1，当前指令执行完）即可响应中断。

(3) 由于其处理程序的入口是通过中断向量表来定位，所以可以方便地用 $\text{INT } n$ 指令进行调用，因此在使用中和一般的子程序没有两样，并且原则上 0~255 种软件中断类型均可使用。

2.1.5 中断程序设计举例

1. 中断程序

已知 PC 机中键盘中断的类型号为 09H, 此中断类型号当按键压下和提起时均会产生中断。下面这个中断程序调用实例主要完成修改 09H 号中断向量 (用 C 语言编写), 使其指向用户自定义的中断服务程序, 在自定义的中断服务程序中读取键码 (扫描码), 需要说明的是读取键盘扫描码的 I/O 地址为 60H; 当用户自定义程序退出的时候, 恢复系统 09H 原来中断向量的入口地址。实现该功能的 C 语言程序代码及说明如下:

```

/*例 2-1.c*/
#include <stdio.h>           //支持标准输入输出, 如 printf()
#include <conio.h>          //控制台输入输出
#include <dos.h>            //支持接口访问、中断向量操作函数
void interrupt (*sys_vect09)(); //声明一个函数为中断函数, 该中断函数的中断号为
sys_vect09
unsigned int key;           //定义全局变量 key (无符号整型), 记录 60H 端口的值
unsigned int flag;         //定义标志 flag, 用来判断是否进行过新的中断
void interrupt my_vect09() //自定义的中断函数
{
    key=inportb(0x60);      //从 60H 端口读取一个字节的的数据送变量 key
    flag=1;                //标志 flag 置 1, 说明变量 key 从端口读取一个新值
    (*sys_vect09)();       //执行原来系统中的中断函数
}

void main()
{
    flag=0;                //初始化标志 flag, 表示还没有发生新的中断
    sys_vect09 = getvect(0x09); //读取系统 9 号调用的中断向量地址, 并存入 sys_vect09
    setvect(0x09,my_vect09); //设置系统新的 9 号调用入口地址是 my_vect09
    while(1) {
        if(flag) {
            printf("%02x",key); //输出中断从 60H 端口读取的值到屏幕
            flag=0;             //标志清 0, 准备接收新的 key 值
            if(key == 0x0010) break; //当输入的值是 0AH 时, 退出 while 循环
            if((key&0x0080) == 0x0080) printf("\n");
                //当字节变量 key 的最高位是 1 时, 屏幕上输出换行符
        }
    }
    setvect(0x09,sys_vect09); //恢复系统 9H 中断的中断入口地址
}

```

2. 相关函数说明

(1) inportb() 函数。Inportb() 函数的原型是:

```
int inportb(int port)
```

该函数是从指定的输入端口 port 读入一个字节, 并返回这个字节。

(2) getvect() 与 setvect() 函数。getvect() 函数是用来获取中断处理程序的入口地址的; setvect() 函数则用来设置某个中断处理程序的入口地址。通过使用这两个函数, 可以自定义中断处理程序 (如 DOS 中断 int 21H 等), 实现诸如内存驻留程序等功能。

(3) interrupt()函数。定义一个中断类型函数“void interrupt (*sys_vect09);”,在用 setvect 改写中断 09H 时就是“setvect(0x09,my_vect09);”,在改写之前应该先保存原来中断 09h 的中断向量地址,这就用到了 getvect()函数,“sys_vect09 = getvect(0x09);”,其中 sys_vect09 是个声明为“void interrupt (*sys_vect09);”型的指针。

2.2 可编程中断控制器 Intel 8259A

可编程中断控制器 Intel 8259A 是 Intel 公司专门为 80x86CPU 控制外部中断而设计的芯片。可编程中断控制器把中断源优先级排队、中断源识别、中断向量提供和中断屏蔽等功能电路集于一身,因而中断系统无需附加任何电路,只需对 8259A 进行编程即可管理 8 级优先中断,中断请求方式和优先级模式也可以通过编程设定或变更,中断入口地址也可由用户任意指定。另外,可将多片 8259A 进行级联,最多可构成 64 级优先中断管理系统。当 8259A 得到 CPU 的中断响应之后,能自动提供中断向量号,从而快速得到中断服务的入口地址。在查询中断方式下,优先级的选定与向量中断方式不一样,但软件将对 8259A 查询,而不是对外设查询。查询时,8259A 送回数据字,指出请求服务的最高优先级,然后软件根据这个数据字转移到适当的服务程序。Intel 8259A 的主要性能包括:

- (1) 具有 8 级中断优先控制,通过级联可以扩展至 64 级优先权控制。
- (2) 每一级中断都可以通过初始设置为允许或屏蔽状态。
- (3) 8259A 的工作方式可以通过编程进行设置。
- (4) 8259A 采用 NMOS 制造工艺,只需要单一的+5V 电源。

2.2.1 8259A 的内部结构和工作原理

8259A 的内部结构主要有数据总线缓冲器、读/写控制逻辑、级联缓冲器、中断控制逻辑电路、中断请求寄存器和优先级比较器,如图 2-6 所示。

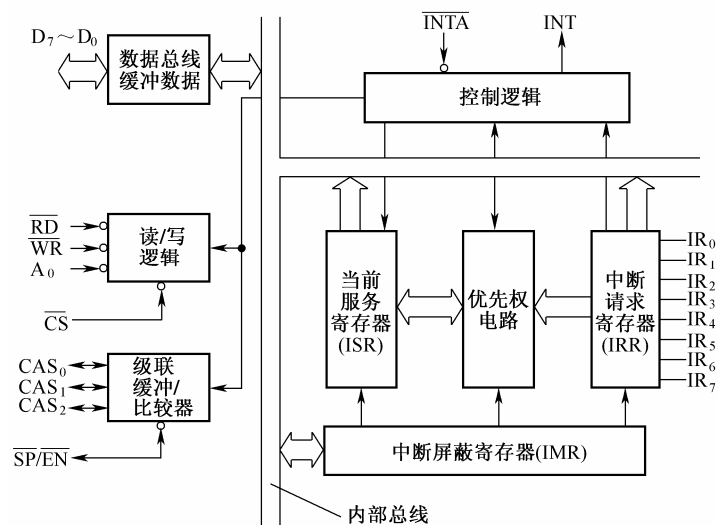


图 2-6 8259A 的内部结构

1. 数据总线缓冲器

数据总线缓冲器是 8259A 与系统数据总线的接口，是 8 位双向三态缓冲器。数据线 $D_7 \sim D_0$ 与 CPU 系统总线相连，构成 CPU 与 8259A 之间的信息传送通道。CPU 与 8259A 之间的控制命令信息、状态信息以及中断类型信息，都是通过该缓冲器传送的。

2. 读/写控制逻辑

读/写控制逻辑是用于接收 CPU 的读/写命令，一方面把来自 CPU 的初始化命令字 ICW 和操作命令字 OCW 存入 8259A 内部相应的端口寄存器，用以规定 8259A 的工作方式和控制模式；另一方面也可使 CPU 通过读/写电路读 8259A 内部有关端口寄存器的状态信息。

3. 级联缓冲器

8259A 既可以工作于单片方式，也可以工作于多片级联方式。当用于实现 8259A 芯片之间的级联时，可把中断源由 8 级扩展至 64 级；在多片级联时，只有一个为主片，其余为从片。

4. 中断控制逻辑电路

中断控制逻辑电路的作用是根据 CPU 对 8259A 编程设定的工作方式产生 8259A 内部控制信号，在适当时候对 CPU 发送中断请求信号 INT，接收来自 CPU 的中断响应信号 \overline{INTA} ，并将 \overline{INTA} 转换为内部所需的各种控制信号。

5. 中断请求寄存器 IRR

中断请求寄存器 IRR 是一个 8 位寄存器，用于记录外部的中断请求。 $IR_0 \sim IR_7$ 为 8 根中断请求输入线，可连接 8 个 I/O 设备的中断请求信号。当 $IR_0 \sim IR_7$ 中的任何一根线上升为高电平时，IRR 中相应的位就置 1；当中断响应信号 \overline{INTA} 有效时，IRR 的相应位被清除。

6. 中断屏蔽寄存器 IMR

中断屏蔽寄存器 IMR 是一个 8 位寄存器，对应 8 个中断源的中断请求信号进行屏蔽控制。当其中某位设置为“0”时，则相应的中断请求可以向 CPU 提出；否则，相应的中断请求被屏蔽，即不允许向 CPU 发出中断请求。该寄存器的内容为 8259A 的操作命令字 OCW1，可以由程序设置或改变。

7. 中断服务寄存器 ISR

中断服务寄存器 ISR 是一个 8 位寄存器，当 CPU 正在处理某个中断源的中断请求时，ISR 寄存器中的相应位设置为 1。

8. 优先级比较器 PR

优先级比较器 PR 用于比较正在处理的中断和刚刚进入的中断请求之间的优先级别，以决定是否产生多重中断或中断嵌套。

2.2.2 8259A 的外部引脚

8259A 是具有 28 个引脚的集成电路芯片，其引脚图及外观示意图如图 2-7 所示。

下面对 28 个引脚进行说明：

(1) $D_7 \sim D_0$ ：双向数据输入/输出引脚，与 CPU 系统总线连接，用以与 CPU 进行信息交换。

(2) $IR_0 \sim IR_7$ ：8 级中断请求信号输入引脚，默认的优先级为 IR_0 最高， IR_7 最低，当有多片 8259A 形成级联时，从片的 INT 与主片的 IR_i 相连。

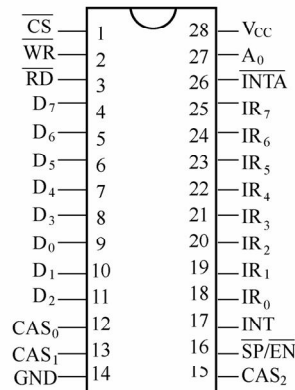


图 2-7 8259A 的引脚信号

(3) $\overline{\text{INT}}$: 中断请求信号输出引脚, 高电平有效, 用以向 CPU 发送中断请求, 应接在 CPU 的 $\overline{\text{INTR}}$ 输入端。

(4) $\overline{\text{INTA}}$: 中断响应应答信号输入引脚, 低电平有效, 在 CPU 发出第二个 $\overline{\text{INTA}}$ 时, 8259A 将其中最高级别的中断请求的中断类型码送出。该引脚应接在 CPU 的 $\overline{\text{INTA}}$ 中断应答信号输出端。

(5) $\overline{\text{RD}}$: 读控制信号输入引脚, 低电平有效, 实现对 8259A 内部有关寄存器内容的读操作。

(6) $\overline{\text{WR}}$: 写控制信号输入引脚, 低电平有效, 实现对 8259A 内部有关寄存器的写操作。

(7) $\overline{\text{CS}}$: 片选信号输入引脚, 低电平有效, 一般由系统地址总线的高位经译码后形成, 决定了 8259A 的端口地址范围。

(8) A_0 : 8259A 两组内部寄存器的选择信号输入引脚, 决定 8259A 的端口地址。其工作方式如下:

- 当 $\text{A}_0=0$ 时, 选择 ICW_1 、 OCW_2 和 OCW_3 。
- 当 $\text{A}_0=1$ 时, 选择 $\text{ICW}_2\sim\text{ICW}_4$ 和 OCW_1 。

(9) $\text{CAS}_2\sim\text{CAS}_0$: 级联信号引脚, 当 8259A 为主片时是输出引脚; 否则为输入引脚。通过与 $\overline{\text{SP/EN}}$ 引脚信号配合, 实现芯片的级联, 这三个引脚信号的不同组合 000~111, 刚好对应于 8 个从片。

(10) $\overline{\text{SP/EN}}$: $\overline{\text{SP}}$ 为级联管理信号的输入引脚, 在非缓冲方式下, 若 8259A 被当作主片使用时, 则 $\overline{\text{SP}}=1$ 接高电平; 当作从片使用时, $\overline{\text{SP}}=0$ 接低电平。在缓冲方式下, $\overline{\text{EN}}$ 作为输出, 用作 8259A 外部数据总线缓冲器的启动信号。

(11) +5V、GND: 电源引脚和接地引脚。

2.2.3 8259A 的工作过程

8259A 应用于 8086/8088 微型计算机系统时, 其进入工作状态的先决条件是必须按要求对其初始化, 使其处于准备就绪状态。当完成初始化后, 8259A 对外部中断请求的响应和处理过程如下:

(1) 当有一条或若干条中断请求输入 ($\text{IR}_7\sim\text{IR}_0$) 有效时, 则使中断请求寄存器的 IRR

的相应位设置为 1。

(2) 若 CPU 处于开中断状态，则在当前指令执行完之后，响应中断，并且从 $\overline{\text{INTA}}$ 发应答信号（两个连续的 $\overline{\text{INTA}}$ 负脉冲）。

(3) 第一个 $\overline{\text{INTA}}$ 负脉冲到达时，IRR 的锁存功能失效，对于 $\text{IR}_7 \sim \text{IR}_0$ 上发来的中断请求信号不予理睬。

(4) 将 ISR 寄存器的相应位置 1，以便为中断优先级比较器的工作做好准备。

(5) 将寄存器的相应位复位，即清除中断请求。

(6) 第二个 $\overline{\text{INTA}}$ 负脉冲到达时，将中断类型寄存器中的内容 ICW_2 送到数据总线的 $\text{D}_7 \sim \text{D}_0$ 上，CPU 以此作为相应中断的类型码。

(7) 若 ICW_4 中的中断结束位为 1，那么第二个 $\overline{\text{INTA}}$ 负脉冲结束时，8259A 将 ISR 寄存器的相应位清零；否则，直至中断服务程序执行完毕，才能通过输出操作命令字 EOI 使该位复位。

2.2.4 8259A 的工作方式

8259A 具有中断优先权管理、中断嵌套、中断屏蔽和中断结束等多种管理工作方式，这些方式可以通过编程设置或改变。

1. 中断优先权管理方式

8259A 中断优先权的管理方式有全嵌套方式、特殊全嵌套方式、优先级自动循环方式和优先级特殊循环方式 4 种。下面分别进行说明：

(1) 全嵌套方式。这是 8259A 默认的优先权设置方式，在全嵌套方式下，8259A 所管理的 8 级中断优先权是固定不变的，其中 IR_0 的中断优先权最高， IR_7 的中断优先权最低。

CPU 响应中断后，请求中断的中断源中优先权最高的中断源在中断服务寄存器 ISR 中的相应位置位，而且把它的中断矢量送至系统数据总线。在此中断源的中断服务完成之前，与其同级或优先级低的中断源的中断请求被屏蔽，只有优先级比它高的中断源的中断请求才能响应，从而出现中断嵌套。

(2) 特殊全嵌套方式。特殊全嵌套方式与全嵌套方式基本相同，所不同的是，当 CPU 处理某一级中断时，如果有同级中断请求，那么 CPU 也会进行响应，从而形成了对同一级中断的特殊嵌套。

特殊全嵌套方式通常应用在有 8259A 级联的系统中，在这种情况下，对主 8259A 编程时，通常使其工作在特殊全嵌套方式下。这样，一方面，CPU 对于优先级别较高的主片的中断输入是允许的；另一方面，CPU 对于来自同一从片的优先级别较高（但对于主片来讲，优先级别是相同的）的中断也是允许并能够响应的。

(3) 优先级自动循环方式。在实际应用中，中断源优先权的情况是比较复杂的，要求 8 级中断的优先权在系统工作过程中可以动态改变，即一个中断源的中断请求被响应之后，其优先权自动降为最低。系统启动时，8 级中断优先权默认为 $\text{IR}_0 \sim \text{IR}_7$ ，假设 IR_4 发出中断请求，若 8259A 工作在优先级自动循环方式下，在 CPU 响应该中断请求之后，中断优先权将自动变为 IR_5 、 IR_6 、 IR_7 、 IR_0 、 IR_1 、 IR_2 、 IR_3 、 IR_4 。

(4) 优先级特殊循环方式。优先级特殊循环方式与自动循环方式相比，只有一点不同，即初始化的优先权是由程序控制的，而不是默认的 $\text{IR}_0 \sim \text{IR}_7$ 。

2. 中断源的屏蔽方式

CPU 对于 8259A 提出的中断请求，都可以加以屏蔽控制。屏蔽控制有普通屏蔽方式和特殊屏蔽方式两种。说明如下：

(1) 普通屏蔽方式。8259A 的每个中断请求输入都要受到屏蔽寄存器中相应位的控制。若相应位为“1”，则中断请求不能向 CPU 发送中断请求。屏蔽是通过屏蔽寄存器 IMR 的编程（操作命令字 OCW_1 ）来加以设置和改变的。

(2) 特殊屏蔽方式。有些场合下，希望在一个中断服务程序的运行过程中能动态地改变系统中的中断优先级结构，即在中断处理的一部分禁止低级中断，而在中断处理的另一部分又能够允许低级中断，于是引入了对中断的特殊屏蔽方式。

设置了特殊屏蔽方式后，用 OCW_1 对屏蔽寄存器中的某一位复位时，同时也会使中断服务寄存器 ISR 中的相应位复位，这样就不只屏蔽了正在处理的等级中断，而且真正开放了其他优先级别较低的中断请求。特殊屏蔽是在中断处理程序中使用的，用了这种方式之后，尽管系统正在处理高级中断，但对外界来讲，只有自己的中断被屏蔽，而允许其他任何级别的中断请求。

3. 结束中断处理的方式

按照对中断结束（复位中断响应寄存器 ISR 中相应位）的不同处理，8259A 有两种工作方式，即是自动结束方式（AEI）和非自动结束方式，而非自动结束方式又可进一步分为一般的中断结束方式和特殊的中断结束方式。

(1) 中断自动结束方式。这种方式仅适用于只有单片 8259A 的场合，在这种方式下，系统一旦响应中断，那么 CPU 在发第二个 \overline{INTA} 脉冲时，就会使中断响应寄存器 ISR 中相应位复位，这样一来，虽然系统在进行中断处理，但对于 8259A 来讲，ISR 没有相应的指示，就如同中断处理结束返回主程序之后一样。CPU 可以再次响应任何级别的中断请求。

(2) 一般的中断结束方式。一般的中断结束方式适合用在全嵌套的情况下，当 CPU 用输出指令向 8259A 发一般中断结束命令 OCW_2 时，8259A 才会使中断响应寄存器 ISR 中优先级最高的位复位。

(3) 特殊的中断结束方式。在特殊全嵌套模式下，系统无法确定哪一级中断为最后处理的中断，也就是说，CPU 无法确定当前所处理的是哪级中断，这时就要采用特殊的中断结束方式。特殊的中断结束方式是指在 CPU 结束中断处理之后，向 8259A 发送一个特殊的 EOI 中断结束命令，这个特殊的中断结束 EOI 命令，明确指出了中断响应寄存器 ISR 中需要复位的位。

注意：在级联方式下，一般不用自动中断结束方式，而需要用非自动结束中断方式，一个中断处理程序结束时，都必须发两个中断结束 EOI 命令，一个发往主片，另一个发往从片。

4. 系统总线的连接方式

按照 8259A 与系统总线的连接方式来分，有下列两种方式。

(1) 缓冲方式。在多片 8259A 级联的系统中，8259A 通过外部总线驱动器和数据总线相连，这就是缓冲方式。在缓冲方式下，8259A 的 $\overline{SP}/\overline{EN}$ 作为输出信号连接到缓冲器，用来启动总线驱动器，在 8259A 与 CPU 之间进行信息交换。

(2) 非缓冲方式。当系统中只有一片或几片 8259A 芯片时，可以将数据总线直接与系统数据总线相连，这时 8259A 处于非缓冲方式；在这种方式下，8259A 的 $\overline{SP}/\overline{EN}$ 作为输入端设置，主片应接高电平，从片应接低电平。

5. 引入中断请求的方式

按照引入中断请求的方式，8259A 有下列几种工作方式。

(1) 边沿触发方式。8259A 将中断请求输入端出现的上升沿作为中断请求信号，上升沿后相应引脚可以一直保持高电平。

(2) 电平触发方式。8259A 将中断请求输入端出现的高电平作为中断请求信号。需要强调说明的是，在这种方式下，中断响应之后高电平必须及时撤除；否则，在 CPU 响应中断之后，如果打开中断，而此时输入端的高电平还没有消除，此时会引起第二次不应该有的中断。

(3) 中断查询方式。当系统中的中断源超过 64 个时，则可以使 8259A 工作在中断查询方式下。中断查询方式的特点是：中断源仍然向 8259A 发中断请求，但 8259A 却不使用 INT 信号向 CPU 发中断请求信号，CPU 内部的中断允许标志复位，所以 CPU 对 INT 引脚上出现的中断请求呈禁止状态。

CPU 用软件查询的方法来确定中断源，从而实现对设备的中断服务，可见，中断查询方式既有中断的特点，又有查询的特点。从外设的角度来看，是靠中断的方式来请求服务；但从 CPU 的角度来看，是用查询方式来确定发中断请求的中断源。

查询是通过 CPU 向 8259A 发查询命令来实现的，查询命令字由 OCW₃ 构成的，其格式如下：

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
X	0	0	0	1	1	0	0

其中，D₂=1 是查询命令的特征位。

8259A 在接到 CPU 发来的上述格式的查询命令之后，立即组成状态字，等待 CPU 来读取，状态字的格式如下：

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
I	X	X	X	X	W ₂	W ₁	W ₀

当 I=0 时，表示该 8259A 芯片没有中断请求；当 I=1 时，表示有中断请求，W₂、W₁ 和 W₀ 即为本片中中断请求优先级别最高的中断源的编码。

2.2.5 8259A 的编程

1. 8259A 的端口地址

由 8259A 的外部结构可知，与寻址 8259A 内部寄存器组有关的信号包括 \overline{CS} 、A₀、 \overline{AD} 、 \overline{WR} 等，其中 CS 是由地址译码器形成的芯片选择信号，只有该引脚为低电平时，相应的 8259A 芯片才能进行工作。

若 8259A 与 8088CPU 配合使用，可直接将 A₀ 与 CPU 的地址信号输出引脚 A₀ 相连，8259A 的两个端口地址是连续的；若 8259A 与 8086CPU 配合使用，如将 8259A 的 D₇~D₀ 接到 16 位数据总线的低 8 位，则 A₀ 应与 CPU 的地址信号输出引脚 A₁ 相连，此时地址码 A₀ 应取 0，8259A 的两个端口地址都是偶地址，若除以 2 之后仍为偶数，则为偶地址；若除以 2 之后为奇数，则为奇地址，即：

A ₁	A ₀	使用寄存器组
0	0	ICW ₁ 、OCW ₂ 和 OCW ₃
1	0	ICW ₂ ~ICW ₄ 和 OCW ₁

地址线 A_1 用以区分 8259A 芯片中的不同寄存器组, 由于 8259A 内部寄存器的寻址只使用一位地址信号, 所以一片 8259A 芯片占用系统中的两个端口地址, 即偶地址和奇地址, 并且规定偶地址小于奇地址。

注意: 8259A 内部并不只有两个寄存器, 为了区别不同的寄存器, 需采用在有关信息中加特征位或规定有关操作顺序的方法来区分不同的输入/输出信息。

2. 8259A 的初始化编程

在使用 8259A 之前, 必须对其进行初始化编程, 以规定其各种工作方式, 并明确其所处的硬件环境。

若 CPU 用一条输出指令向 8259A 的偶地址端口写入一个命令字, 而且当 $D_4=1$ 时, 则被解释为初始化命令字 ICW_1 , 输出 ICW_1 启动了 8259A 的初始化操作, 8259A 的内、外部自动产生下列操作:

- (1) 边沿敏感电路复位, 中断请求的上升沿有效。
- (2) 中断屏蔽器 IMR 清零, 即对所有的中断呈现允许状态。
- (3) 中断优先级自动按 $IR_0 \sim IR_7$ 排列。
- (4) 清除特殊屏蔽方式。

8259A 的初始化编程需要 CPU 向其输出一个 2~4 字节的初始化命令字, 输出初始化命令字的流程如图 2-8 所示, 其中 ICW_1 和 ICW_2 是必须的, 而 ICW_3 和 ICW_4 需根据具体的情况来加以选择。各初始化命令字的安排与作用如下:

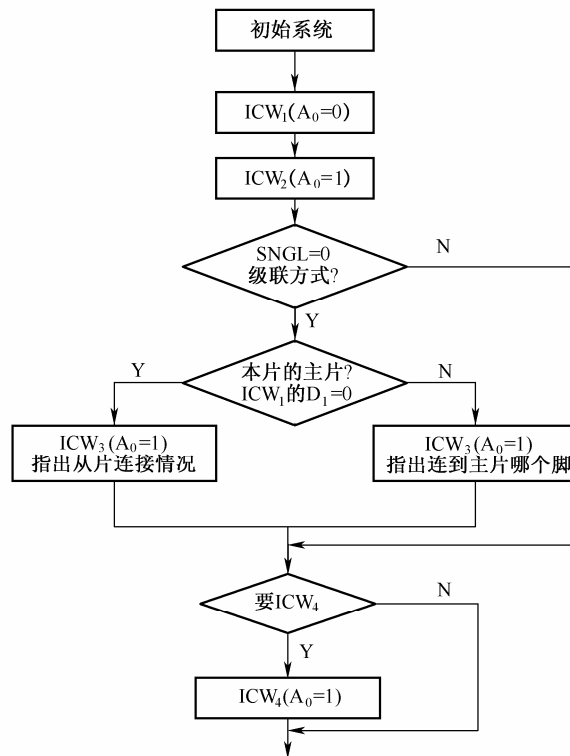


图 2-8 8259A 初始化流程图

(1) ICW₁的格式。ICW₁的格式如图 2-9 所示。初始化命令字 1，写入 8259A 偶地址端口，其各位的功能及含义如下：

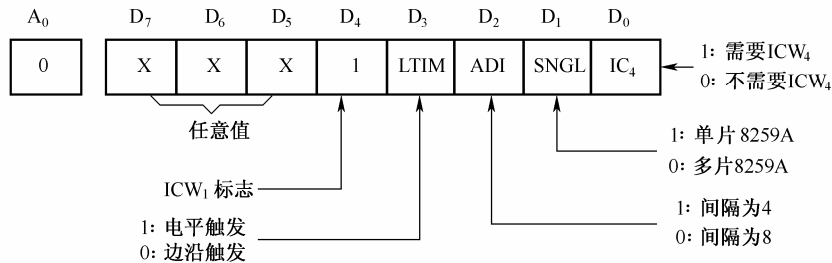


图 2-9 ICW₁的格式

1) IC₄: 用以决定是否需要输出 ICW₄，若 D₀=1，说明必须输出 ICW₄；若 D₀=0，说明不需输出 ICW₄。若 ICW₄ 的各位都为 0，则说明不需要输出 ICW₄。

2) SNGL: 取决于 8259A 芯片是单片工作还是多片级联工作。若 8259A 在单片工作模式下，则 D₁=1；若 8259A 工作在多片级联模式下，则 D₁=0。

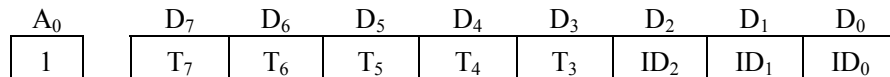
3) ADI: 只用于 MCS80/85 系统中，规定 CALL 地址的间隔，在 8088/8086 系统中，该位无意义。

4) LTIM: 规定中断请求信号的引入方式。若 D₃=1，则表示中断请求信号为高电平有效；若 D₃=0，则表示中断请求信号为上升沿有效。

D₄: 恒定为 1，为 ICW₁ 的特征位。

D₅~D₇: 未用，通常设置为 0。

(2) ICW₂的格式。ICW₂用于设置中断类型号，初始化命令字 2，写入 8259A 奇地址端口。



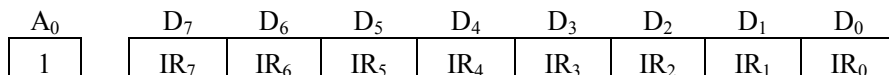
当 8259A 用于 MCS80/85 系统中时，ICW₂用于确定中断入口地址的高 8 位 (A₁₅~A₈)；当 8259A 用于 8088/8086 系统中时，ICW₂的 D₇~D₃为编程设置位，作为本芯片所管理 8 级中断类型码的高 5 位；D₂~D₀位为 8 级中断源所对应的编码 (其中，000 对应 IR₀，111 对应 IR₇)，编程设置对其无影响，它由芯片自动生成。

例如，当 ICW₂=45H 时，8 级中断源的中断类型码分别为 IR₀ 为 40H，…，IR₇ 为 47H。

(3) ICW₃的格式。初始化命令字 3，写入相应 8259A 的奇地址端口。

ICW₃用于 8259A 的级联，8259A 最多允许有一片主片和 8 片从片级联，使能够管理的中断源扩充至 64 个。若系统中只有一片 8259A，则不用 ICW₃；若由多片 8259A 级联，则主、从 8259A 芯片都必须使用 ICW₃，主、从 8259A 芯片中的 ICW₃的使用方式不同。

1) 对于主 8259A 芯片，ICW₃的格式如下：



其中每一位对应于一从片 8259A 芯片，若相应引脚上接有从 8259A 芯片，则相应位为 1；否则，若相应引脚上未接从 8259A 芯片，则相应位为 0。

例如，主 8259A 芯片的 $ICW_3=1110\ 0010$ ，则说明 IR_7 、 IR_6 、 IR_5 、 IR_1 上连有从片。

2) 对于从 8259A 芯片， ICW_3 的格式如下：

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	1/0	1/0	1/0	1/0	1/0	ID ₂	ID ₁	ID ₀

从 8259A 芯片中的 ICW_3 ，只用其中的低 3 位来设置该芯片的标识符，高 5 位全为 0。在中断响应时，主 8259A 芯片通过级联线 $CS_2\sim CS_0$ 依次向各个从 8259A 芯片发送中断请求的优先级最高的源所对应的标识符，每个从 8259A 芯片收到这个标识符之后，与自己在初始化编程时由 ICW_3 设置的标识符进行比较，当两者相符合时，则该从 8259A 芯片在第二个中断响应周期向 CPU 提供由 ICW_2 设置的 8 位中断类型码。

如果本从片的 INT 引脚接在主片的 IR_1 引脚上，则 $ICW_3=0000\ 0001$ 。

(4) ICW_4 的格式。初始化命令字 4，写入 8259A 奇地址端口。只有当 ICW_1 中的 $D_0=1$ 时才需要设置，其各位的功能及含义如下：

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	SFNM	BUF	M/S	AEOI	μ PM

1) D_0 : μ PM 位，取决于系统中所采用微处理器的类型，若系统中的微处理器为 MCS80/85，则 $D_0=0$ ；反之，若系统中的微处理器为 8088/8086，则 $D_0=1$ 。

2) D_1 : AEOI 位，规定结束中断的方式。若 $D_1=1$ ，则为自动中断结束方式；若 $D_1=0$ ，则需要用中断结束命令来结束中断。

3) D_2 : M/S 位，缓冲方式下使用。若 $D_2=1$ ，则表示为主 8259A 芯片；若 $D_2=0$ ，则表示为从 8259A 芯片。

4) D_3 : BUF 位，若 8259A 工作于缓冲方式，则 $D_3=1$ ；否则 $D_3=0$ 。

5) D_4 : SFNM 位：若 $D_4=1$ ，则规定特殊的全嵌套模式；若 $D_4=0$ ，则规定普通的全嵌套模式。

6) $D_5\sim D_7$: 恒定为 000。

3. 8259A 的编程操作

对 8259A 按照上述流程进行初始化编程之后，相应芯片就做好了接收中断的准备，若中断源发生了中断请求，则 8259A 按照初始化编程所规定的各种方式来处理这种请求。在 8259A 的工作期间，CPU 也可以通过操作命令字实现对 8259A 的操作控制，或者改变工作方式，或者实时读取 8259A 中某些寄存器的内容。8259A 有三个操作命令字，其格式及含义如下：

(1) OCW_1 : 中断屏蔽字，必须写入相应 8259A 芯片的奇地址端口，其格式如下：

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	M ₇	M ₆	M ₅	M ₄	M ₃	M ₂	M ₁	M ₀

OCW_1 的每一位可以对相应的中断请求输入进行屏蔽，若 OCW_1 的某一位为 1，则相应的中断请求输入被屏蔽；反之，则相应的中断请求输入呈现允许状态。即若 $M_i=1$ ，则表示 8259A 对 IR_i 的中断请求呈屏蔽状态；否则若 $M_i=0$ ，则表示 8259A 对 IR_i 的中断请求呈允许状态。

(2) OCW_2 : 必须写入相应 8259A 芯片的偶地址端口，其格式如下：

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	R	SL	EOI	0	0	L ₂	L ₁	L ₀

其中 D_4 、 D_3 位恒定为 0，是 OCW_2 的特征位，而 R、SL、EOI 三位的不同组合，可以组成 7 种不同的操作命令，用于改变 8259A 的工作方式。其中三种操作命令字要用到 OCW_2 的低三位，这三位所形成的编码指出操作所涉及的中断源。其中：

- 1) R: 用于表示优先级是否采用循环方式。
- 2) SL: 用于确定是否需要使用 L_2 、 L_1 、 L_0 来指定中断源。
- 3) EOI: 用于指示 OCW_2 是否作为中断结束命令。
- 4) L_2 、 L_1 、 L_0 : 当 $SL=1$ 时，三位的编码用以指示 8 个中断源之一。

R、SL、EOI 共有 8 种不同的组合形式，其中有 7 种是相应的控制命令，其详细说明如下：

- 0、0、0: 取消自动 EOI 循环命令。
- 1、0、0: 设置自动 EOI 循环命令。
- 0、0、1: 普通的 EOI 命令，它适用于完全嵌套方式，在中断服务程序结束时，用于清除 ISR 中最后被置位的相应位。显然，只有在 ICW_4 中的 $AEIOI=0$ 时，才需要在中断服务子程序中向 8259A 发普通的 EOI 命令。
- 0、1、1: 特殊的 EOI 命令，与普通的 EOI 命令的差别在于，它需要利用 L_2 、 L_1 、 L_0 位明确指出 ISR 寄存器中需要被复位的位。
- 1、0、1: 普通循环的 EOI 命令，它在中断服务程序结束时使用，它使已置位的 ISR 寄存器中优先级最高的那一位复位，同时赋予刚刚结束中断处理的中断源的中断优先级最低。
- 1、1、1: 特殊的 EOI 循环命令，它一方面复位 ISR 寄存器中由 L_2 、 L_1 、 L_0 位明确指出的那一位；另一方面，使 L_2 、 L_1 、 L_0 位明确指出的那个中断源的中断优先级最低。
- 1、1、0: 置位优先权命令，用以设置优先级特殊循环方式，即利用 L_2 、 L_1 、 L_0 位明确指出中断优先级最低的中断源。
- 0、1、0: 非操作命令，无实际意义。

(3) OCW_3 : 必须写入相应 8259A 芯片的偶地址端口。其格式如下：

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	ESMM	SMM	0	1	P	RR	RIS

1) D_0 : RIS 位，用以决定下一个读操作所对应的寄存器。若 $D_0=1$ ，则下一个读操作读取中断服务寄存器 ISR 的内容；否则，读取中断请求寄存器 IRR 的内容。

2) D_1 : RR 位，决定下一个操作是否为读操作。若 $D_1=1$ ，则下一个操作是读操作；否则，下一个操作不是读操作。

3) D_2 : P 位，用于 8259A 的查询中断方式。若 $D_2=1$ ，表示为查询命令；否则，表示不是查询命令。

4) D_3 、 D_4 : 恒定为 10，是 OCW_3 的特征位。

5) D_5 、 D_6 : 决定 8259A 是否为设置特殊屏蔽模式命令，若 D_6 、 D_5 为 11，则为设置特殊屏蔽模式命令；若 D_6 、 D_5 为 01，则为撤消特殊屏蔽模式、返回普通屏蔽模式命令；若 $D_6=0$ ，则 D_5 无意义。

6) D_7 : 无关。

2.2.6 8259A 的级联

所谓级联就是在微型计算机系统以 1 片 8259A 的 INT 引脚与 CPU 的 INTR 引脚相连, 称为主片; 再将最多 8 片 8259A 的 INT 引脚, 分别与主 8259A 的 $IR_0 \sim IR_7$ 相连, 称为从片。显然, 在主一从式 8259 级联的微机系统中, 系统能够管理的中断源可由 8 级扩展至 64 级。

主一从式 8259 级联系统的连接, 需要注意以下几点:

(1) 主片的 INT 引脚接 CPU 的 INTR 引脚, 从片的 INT 引脚分别接主片的 IR_i 引脚, 使得由从片输入的中断请求, 能够通过主片向 CPU 发出。

(2) 主片的 3 条级联线与各从片的同名级联线引脚对接, 主片为输出, 从片为输入。主片用以向各从片发出优先级别最高的中断请求的从片代码, 各从片用该代码与本片的代码进行比较, 符合则将本片 ICW_2 中预先设定中断类型码, 送至数据总线。

(3) 主片的 $\overline{SP}/\overline{EN}$ 接 +5V, 从片的 $\overline{SP}/\overline{EN}$ 接地。

如图 2-10 所示为 1 片主片与 3 片从片级联的情况。

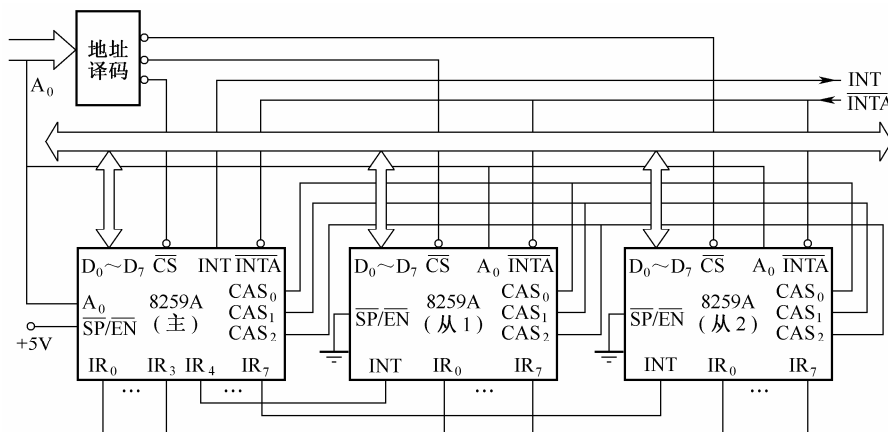


图 2-10 8259A 的级联

级联系统中的所有 8259A 都必须进行各自独立的编程, 作为主片的 8259A 必须设置为特殊的全嵌套方式, 以避免相同从片中优先级较高的中断请求被屏蔽的情况发生。与一般的全嵌套方式相比, 有以下两点需要注意:

(1) 当来自某个从设备的中断请求被响应之后, 主片的优先权逻辑不封锁这个从片, 从而可以使来自从设备的较高优先级的中断请求能被主片正常接受, 并向 CPU 发出。

(2) 中断服务结束时, 必须用软件来检查被服务的中断是否为该从片中唯一的 interrupt request。为此, 须先向从片发一个一般的中断结束命令, 清除已完成服务的 ISR 中的相应位, 然后, 再读出 ISR 的内容, 检查是否为全 0, 若为全 0, 则向主片发一个中断结束命令, 清除与从设备相应的 ISR 中的位; 反之, 则不向主片发中断结束命令, 因为同一从片中还有其他中断请求正在处理。

2.2.7 8259A 的应用举例

例 2-1 CPU 与 8259A 连接如图 2-11 所示。向 8259A 的中断屏蔽寄存器 IMR 写入控制

命令，使系统时钟、键盘、硬盘、软盘开放中断，并使其他中断关闭，其控制语句如下：

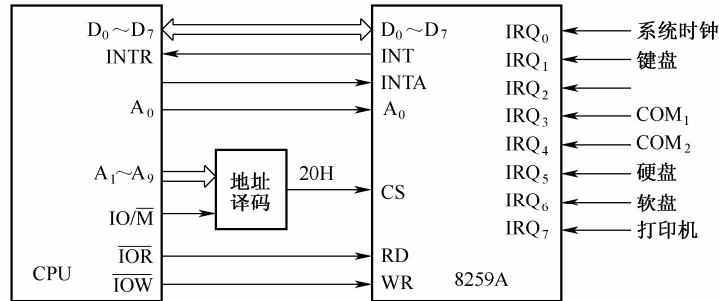


图 2-11 CPU 与 8259 连接示意图

```
MOV    AL, 01100011B      ; 汇编语言
OUT    20H, AL
或：
outportb(0x20, 0x63);    // C 语言程序
```

例 2-2 IBM-PC 机中，只有一片 8259A，可接受外部 8 级中断。在 I/O 地址中，分配 8259A 的端口地址为 20H 和 21H，初始化为：边沿触发、缓冲连接、中断结束采用 EOI 命令、中断优先级采用完全嵌套方式，8 级中断源的中断类型分别为 08H~0FH，初始化程序使用汇编语言实现，其具体语句如下所示：

```
MOV    DX, 20H
MOV    AL, 00010011B
OUT    DX, AL             ; 写入 ICW1
MOV    DX, 21H
MOV    AL, 08H
OUT    DX, AL             ; 写入 ICW2
MOV    AL, 00001101B
OUT    DX, AL             ; 写入 ICW4
XOR    AL, AL
OUT    DX, AL             ; 写入 OCW1
.....
STI
```

例 2-3 进入和退出特殊屏蔽方式的流程图。

假定初始化之后，8259A 工作于完全嵌套方式，要求对于 IR₃ 的中断级能够允许任何级别的中断去中断其服务程序，即 8259A 按特殊屏蔽方式工作。因而在响应 IR₃ 而执行 IR₃ 的中断服务程序时，在服务程序的某个语句之后写入 OCW₁ 以屏蔽 IR₃，然后写入 OCW₃ 使 ESMM=SMM=1，于是从该处开始，8259A 进入特殊屏蔽方式，此后继续执行 IR₃ 的中断服务程序；在中断服务结束之前，再向 8259A 写入 OCW₃ 使 ESMM=1，SMM=0，结束特殊屏蔽方式，返回到完全嵌套方式；接着写入 OCW₁，撤消对 IR₃ 的屏蔽，最后写入 OCW₂，向 8259A 发出 EOI 命令。此例说明在 IR₃ 的中断服务程序的某个地方允许任何级别的中断源中断 IR₃ 的服务程序（除本身之外）。

```
... ; IR3 中断服务程序入口
```

```

STI                ;保护现场
...                ;STI 开中断
MOV AL,00001000B   ;服务程序
OUT 21H,AL         ;OCW1 写入 OCW1, 使 IM3=1
MOV AL,01101000B   ;写入 OCW3, 使 ESMM=SMM=1
OUT 20H,AL         ;OCW3 继续服务
...                ;写入 OCW3, 使 ESMM=1, SMM=0
MOV AL,01001000B   ;写入 OCW1, 使 IM3=0
OUT 20H,AL         ;OCW3 写入 OCW2, 普通的 EOI 命令
MOV AL,00H         ;中断返回
OUT 21H,AL         ;OCW1
MOV AL,00100111B
OUT 20H,AL         ;OCW3
OUT 21H,AL         ;OCW3 EOI 命令

```

例 2-4 读取 8259A 相关寄存器的内容。

设 8259A 的端口地址为 20H、21H，请读入 IRR、ISR、IMR 寄存器的内容，并相继保存在数据段 2000H 开始的内存单元中；若该 8259A 为主片，用查询方式查询哪个从片有中断请求。

```

MOV AL,00001010B   ;发 OCW3, 读取 IRR 的内容
OUT 20H, AL
IN AL, 20H         ;读入并保存 IRR 的内容
MOV [2000H], AL
MOV AL,00001011B   ;发 OCW3, 读取 ISR 的内容
OUT 20H, AL
IN AL, 20H         ;读入并保存 ISR 的内容
MOV [2001H], AL
IN AL, 21H         ;读入并保存 ISR 的内容
MOV [2002H], AL
MOV AL, 0000110xB   ;发 OCW3, 查询是否有中断请求
OUT 20H
IN AL, 20H         ;读入相应状态, 并判断最高位是否为 1
TEST AL, 80H
JZ DONE
AND AL, 07H        ;判断中断源的编码
.....
DONE: HLT

```

本章小结

本章介绍了中断技术原理及实现，重点介绍中断控制器 8259A，主要包括可编程中断控制器 8259A 的主要功能、8259A 的基本组成部分、8259A 的初始化编程格式与方法，以及在 8259A 的工作期间，CPU 也可以通过三个操作命令字，实现对 8259A 的操作控制，或者改变工作方式，或者实时读取 8259A 中某些寄存器的内容。

习题二

一、填空题

1. 现代微机的中断有_____、_____、_____和_____四种类型。
2. 可屏蔽硬件中断的处理过程可分为_____、_____、_____和_____四个阶段。
3. PC 系列微机的存储器的 0000~03FE 共 1024 个地址单元作为中断向量存储初始化区，每个中断向量需占用_____个字节的地址空间，所以可容纳_____个中断向量。
4. 1 片 8259A 可接收_____个中断源的中断输入，2 片 PIC8259A 级联可扩充至_____级中断源。
5. 可屏蔽中断从 CPU 的_____引脚进入，只有当中断允许标志 IF 为_____时，该中断才能得到响应。
6. 设 8086 系统中采用单片 8259A，其 8259A 的 ICW₂=32H，则对应 IR₅ 的中断类型为_____H，它的中断入口地址在中断向量表中的地址为_____H。
7. 在 8086/8088PC 机系统中使用的中断控制器是_____芯片，该中断控制器可接受_____级硬件外中断。
8. PC 机系统中，中断优先级别最高的为_____中断，而级别最低的为_____中断。
9. 在中断响应期间，CPU 要做的工作为_____、_____、_____。
10. 当中断控制器 8259A 的 A₀ 接向地址总线 A₁ 时，若其中一个端口地址为 62H，则另一个口地址为_____H；若某外设的中断类型码为 86H，则该中断应加到 8259A 的中断请求寄存器 IRR 的_____输入端。
11. 8086 最大方式下，两个中断响应周期 $\overline{\text{INTA}}$ 内，微处理器发出_____信号以通知其他_____不能再企图控制总线。
12. 最大方式下，若第二个 $\overline{\text{INTA}}$ 响应周期下，8259A 把_____送到数据总线上。
13. 8259A 应用于 8086 微处理器时，8259A 引脚 A₀ 与地址总线_____相连；8259A 应用于 8088 微处理器时，A₀ 地址总线与_____相连。
14. 8259A 工作于缓冲方式时， $\overline{\text{SP}}/\overline{\text{EN}}$ 是_____方向；当工作于非缓冲方式时， $\overline{\text{SP}}/\overline{\text{EN}}$ 是_____方向。
15. 若 8259A 是主片时，引脚 $\overline{\text{SP}}/\overline{\text{EN}}$ 是_____电平；若 8259A 是从片时，则引脚 $\overline{\text{SP}}/\overline{\text{EN}}$ 是_____电平。
16. 三片可编程中断控制器 8259A 主从连接时最多可供_____个中断源。
17. 8259A 的初始化控制字 ICW 有_____个，其中_____和_____为必写控制字，_____和_____为选写控制字。
18. _____控制字应在 8259A 启动时设置；_____控制字在 8259A 工作过程中

调整。

19. 若 ICW₁ 中的值是 10H, IR₀~IR₇ 的触发方式为_____。
20. 某 8259A 的 IR₇ 对应的中断类型号 8FH, ICW₂ 应写入的值为_____。
21. 在主从 8259A 结构中, 它们的 ICW₃ 写入的内容为_____, 若从片的 INT 接主片的 IR₂, 则主片 ICW₃ 的值为_____; 从片 ICW₃ 的值为_____。
22. 若 OCW₁ 中的值是 28H, IR₀~IR₇ 的_____和_____中断屏蔽。
23. 若 8259A 的功能设置为特殊优先级且 IR₅ 有最高优先级, OCW₂ 应写入_____。
24. 8259A 的寄存器 IRR 读操作前应先对_____进行写控制字操作且写入字为_____。

二、判断题

- () 1. 可编程中断控制器 8259A 仅能单片使用。
- () 2. 8259A 的输入中断源 IR₀~IR₇ 的中断请求信号仅为高电平有效。
- () 3. 8259A 的固定优先级序为 IR₇→IR₀。
- () 4. 8259A 的中断屏蔽寄存器 IMR 的级别低于 8086 的中断控制标志 IF。
- () 5. 8259A 的等优先级的初始优先级序为 IR₀→IR₇。
- () 6. 两片 8259A 主从连接时, 最多可提供 16 个中断源。
- () 7. 8259A 初始化控制字 ICW₄ 时必写控制字。
- () 8. 8259A 的初始化控制字 ICW₁ 可随时根据需要写入。
- () 9. 8259A 的中断类型号在 OCW₂ 中写入。
- () 10. 主从 8259A 的控制字 ICW₃ 的写入内容不同。
- () 11. 8259A 的初始化控制字 ICW 的写入方法为同址顺序写入法。
- () 12. 8259A 的工作控制字 OCW 的写入方法为同址位识别写入法。
- () 13. 8259A 的 IRR 寄存器的读操作与控制字 OCW₃ 有关。
- () 14. 8259A 的 IMR 寄存器的读操作与控制字 OCW₃ 有关。
- () 15. 寄存器 IMR、IRR、ISR 的读操作地址相同。
- () 16. 寄存器 IRR、ISR 的读操作地址相同。
- () 17. 内部中断的优先级总是高于外部中断。
- () 18. 8259A 所管理的中断源中, 优先级低的中断源不可能中断优先级高的中断服务子程序。
- () 19. 若 8259A 中断屏蔽字 OCW₁ 为 00H, 则 8259A 所管理的 8 级中断全被屏蔽。
- () 20. 只要 8259A 所管理的中断源没有被屏蔽, 则任何中断源的中断请求都能得到 CPU 的响应和服务。

三、选择题

1. 8259A 固定优先级方式的中断请求信号 IR₀~IR₇ 的优先级顺序为 ()。
 - A. IR₀→IR₇
 - B. IR₇→IR₀
 - C. IR₀→IR₇→IR₁
 - D. 每个中断请求信号等优先级
2. 8259A 的中断屏蔽寄存器为 ()。

- A. IRR B. IMR C. ISR D. PR
3. CPU 可访问 8259A 的端口地址数为 () 个。
A. 1 B. 2 C. 4 D. 8
4. 8259A 工作在边沿触发, 单片使用, 需对 ICW₄ 操作, ICW₁ 的值为 ()。
A. 00H B. 11H C. 13H D. 1DH
5. 若 8259A 的 ICW₂ 中值为 08H, 中断源 IR₀~IR₇ 对应的中断号为 ()。
A. 00H~07H B. 08H~0FH C. 80H~87H D. 88H~8FH
6. 若外设有 64 个中断源, 需用 () 片 8259A。
A. 1 B. 8 C. 9 D. 64
7. 在 8259A 应用中需对 IR₅、IR₃ 进行屏蔽, 操作命令 0CW₁ 应写入 ()。
A. 53H B. 35H C. 00H D. 28H
8. 8259A 操作命令字 0CW₂ 写入值为 20H, 功能为 ()。
A. 常规 EOI 中断结束 B. 特殊 EOI 中断结束
C. 在自动 EOI 时循环 D. 在特殊 EOI 时循环
9. 用 8259A 构成主从两级中断控制, 2 片 8259A 级联可提供 () 个中断源。
A. 1 B. 8 C. 15 D. 16
10. 在 PC/XT 机中键盘的中断类型码是 09H, 则键盘中断矢量存储在 ()。
A. 36H~39H B. 24H~27H C. 18H~21H D. 18H~1BH
11. 3 片 8259A 级联起来, 可管理 () 级中断。
A. 24 B. 23 C. 22 D. 20
E. 16 F. 15
12. 若 8259A 工作在优先级自动循环方式, 则 IRQ₄ 的中断请求被响应并且服务完毕后, 优先权最高的中断源是 ()。
A. IRQ₃ B. IRQ₅ C. IRQ₀ D. IRQ₄
13. 已知某系统共有 4 台外设, 即 A、B、C 和 D, 每台外设都可能发生中断, 系统规定中断优先级为 A 最高, D 最低, B、C 同级介于 A、D 之间。当 CPU 在执行 C 设备的中断服务程序时, 其余三台外设同时发生中断请求, 此时, IF=1, CPU 响应 () 外设请求。
A. B 设备 B. A 设备
C. 都不响应 D. 依 A、B、D 次序都响应
14. 已知某系统共带 3 台外设, 即 X、Y、Z, 每台外设都能发生中断, 它们的中断优先级为 X>Y>Z, 当前在执行 Z 中断服务程序时, X 和 Y 同时发出中断请求, 若此时 IF=0, 问 CPU 响应 () 外设请求。
A. X 设备 B. Y 设备
C. 先响应 X, 后响应 Y D. 都不响应
15. 在 8088/8086 系统中使用 8259A 作为中断控制器, 试问 8259A 中的中断屏蔽寄存器 (IMR) 中的屏蔽位与 CPU 的标志寄存器中的 IF 位对中断影响是 () 关系。
A. 屏蔽位与 IF 位都用来控制某一个中断源的中断请求
B. 屏蔽位只使对应中断起开中断或关中断的功能
C. 屏蔽位和 IF 位功能完全一样, 可互相替代

- D. 屏蔽位只对一个中断源屏蔽, 而 IF 位对所有中断源的是中断屏蔽
16. IBM-PC 系统中规定的中断优先级次序, 下面正确的是 ()。
- A. 内中断 (除法错、INTO, INT) >NMI>INTR>单步中断
 B. NMI>内中断>INTR>单步中断
 C. 内中断>单步中断>NMI>INTR
 D. NMI>内中断>单步中断>INTR
17. 中断过程与子程序之间用过程比较, 下面错误的是 ()。
- A. 中断过程与子程序调用过程都需保护断点
 B. 中断过程与子程序调用过程都需保护标志寄存器
 C. 中断过程与子程序调用过程都要改变程序执行方向
 D. 中断过程是随机的, 而调用子程序是事先安置好的

四、简答题

1. 说明 AEOI 和 EOI 的区别, 一般 EOI 和特殊 EOI 的区别, 并说明它们各适合在什么情况下使用?

2. 8259A 的主要功能有哪些?

3. 8259A 对中断优先权的管理方式有哪几种? 各是什么含义?

4. 8259A 内部实现优先权管理的核心部件是哪些?

5. 8086CPU 的中断源是如何分类的? 并写出 8086 CPU 中断优先级由高到低的次序。

6. 设 8259A 的端口地址为 20H 和 21H, 若执行下面初始化程序段:

```
MOV AL, 13H
OUT 20H, AL
MOV AL, 40H
OUT 21H, AL
MOV AL, 01H
OUT 21H, AL
```

问: (1) 中断请求信号 IR 的触发方式为_____。

(2) 中断级 IR₂ 的中断类型码为_____。

(3) 中断结束的方式为_____。

(4) 设置优先级的方式为_____。

7. 8259A 的主要功能是什么? 它内部的可读写寄存器完成什么功能?

8. 8259A 分别有哪些初始化命令字和操作命令字? 它们的使用场合有什么不同?

9. 8259A 的中断屏蔽寄存器 IMR 与 8086 中断允许标志 IF 有什么区别?

10. 若系统采用单片 8259A, 中断类型码定义为 20H~27H, 某中断源的请求线与 8259A 的 IR₄ 相连, 试问: 对应中断源的中断向量表入口地址是多少? 若该中断服务程序入口地址为 4FE24H, 则对应该中断源的中断向量表内容是什么, 如何定位?