# 外中断

PC系统的接口卡和主板上,装有各种接口芯片.这些外设接口芯片的内部有若干寄存器,CPU将这些寄存器当作端口来访问.

外设的输入不直接送入内存和CPU,而是送入相关的接口芯片的端口中.

CPU向外设输出也不是直接送入外设,而是先送入端口中,再由相关的芯片送到外设.

## 外中断信息

外设的输入到达,相关芯片将向CPU发出相应的中断信息.CPU在执行完当前指令后,可以检测到发送过来的中断信息,引发中断过程,处理外设的输入.

### 可屏蔽中断

可屏蔽中断是CPU可以不响应的外中断.当CPU检测到可屏蔽中断信息时,如果IF=1,则CPU在执行完当前指令后响应中断,引发中断过程;如果IF=0,则不响应可屏蔽中断.

内中断所引发的中断过程:

(1) 取中断类型码n;

(2) 标志寄存器入栈, IF=0, TF=0;

(3) CS、IP入栈;

(4) (IP)=(n\*4), (CS)=(n\*4+2)

可屏蔽中断所引发的中断过程,除在第1步的实现上有所不同外,基本上和内中断的中断过程相同.

可屏蔽中断信息来自于CPU外部,中断类型码是通过数据总线送入CPU的;而内中断的中断类型码是在CPU内部产生的.

将IF置0的原因是在进入中断处理程序后,禁止其他的可屏蔽中断.

如果在中断处理程序中需要处理可屏蔽中断,可以用指令将IF置1.8086CPU提供的设置IF的指令如下:

sti, 设置IF=1;

cli, 设置IF=0.

### 不可屏蔽中断

不可屏蔽中断是CPU必须响应的外中断.当CPU检测到不可屏蔽中断信息时,则在执行完当前指令后,立即响应,引发中断过程.

对于8086CPU,不可屏蔽中断的中断类型码固定为2,所以中断过程中,不需要取中断类型码.则不可屏蔽中断的中断过程为:

(1) 标志寄存器入栈, IF=0, TF=0;

(2) CS、IP入栈;

(3) (IP)=(8), (CS)=(0AH).

几乎所有由外设引发的外中断,都是可屏蔽中断.

### PC机键盘的处理过程

#### 1.键盘输入

键盘上的每一个键相当于一个开关,键盘中有一个芯片对键盘上的每一个键的开关状态进行扫描.

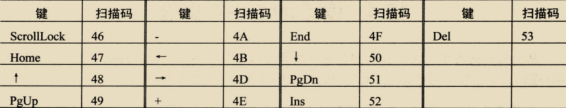
按下一个键时,开关接通,该芯片就产生一个扫描码,扫描码说明了按下的键在键盘上的位置.扫描码被送入主板上的相关接口芯片的寄存器中,该寄存器的端口地址为60h.

松开按下的键时,也产生一个扫描码,扫描码说明了松开的键在键盘上的位置.松开按键时产生的扫描码也被送入60h端口中.

一般将按下一个键时产生的扫描码称为通码,松开一个键产生的扫描码称为断码.扫描码长度为一个字节,通码的第7位为0,断码的第7位为1,即:

**断码=通码+80h**





#### 2.引发9号中断

键盘的输入到达60h端口时,相关的芯片就会向CPU发出中断类型码为9的可屏蔽中断信息.

#### 3.执行int 9中断例程

BIOS提供了int 9中断例程, 用来进行基本的键盘输入处理,主要工作如下:

(1)读出60h端口中的扫描码;

(2)如果是字符键的扫描码,将该扫描码和它所对应的字符码(即ASCII码)送入内存中的BIOS键盘缓冲区;如果是控制键(比如Ctrl)和切换键(比如CapsLock)的扫描码,则将其转变为状态字节(用二进制位记录控制键和切换键状态的字节)写入内存中存储状态字节的单元;

(3)对键盘系统进行相关的控制,比如说,向相关芯片发出应答信息.

BIOS键盘缓冲区是系统启动后,BIOS用于存放int 9中断例程所接收的键盘输入的内存区.该内存区可以存储15个键盘输入.在BIOS键盘缓冲区中,一个键盘输入用一个字单元存放,高位字节存放扫描码,低位字节存放字符码.

0040:17单元存储键盘状态字节,该字节记录了控制键和切换键的状态.键盘状态字节各位记录的信息如下.

0: 右shift状态,置1表示按下右shift键;

1: 左shift状态,置1表示按下左shift键;

2: Ctrl状态,置1表示按下Ctrl键;

3: Alt状态,置1表示按下Alt键;

4: ScrollLock状态,置1表示Scroll指示灯亮;

5: NumLock状态,置1表示小键盘输入的是数字;

6: CapsLock状态,置1表示输入大写字母;

7: Insert状态,置1表示处于删除态.