

王爽汇编第14章,外中断

1. 外中断概述
2. 可屏蔽中断
3. 不可屏蔽中断
4. 编写int 9中断例程

1. 外中断概述

CPU在计算机系统中,除了能够执行指令,进行运算以外,还应该能够对外部设备进行控制,接收它们的输入,向它们进行输出(I/O能力)

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

CPU通过端口和外部设备进行联系

- 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向外设的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。
- CPU还可以向外设输出控制命令,而这些控制命令也是先送到相关芯片的端口中,然后再由相关的芯片根据命令对外设实施控制。

当CPU外部有需要处理的事情发生的时候,比如说,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。

2. 可屏蔽中断

可屏蔽中断是CPU可以不响应的外中断。CPU是否响应可屏蔽中断,要看标志寄存器的IF位的设置。

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。

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。

中断过程中将IF置0的原因就是,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: sti, 设置 IF=1; cli, 设置 IF=0。

3. 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:

- 标志寄存器入栈, IF=0, TF=0;
- CS、IP入栈;
- IP = 8, CS = 0AH

几乎所有由外设引发的外中断，都是可屏蔽中断。当外设需要处理的事件（比如说键盘输入）发生时，相关芯片向CPU发出可屏蔽中断信息。

不可屏蔽中断是在系统中有必须处理的紧急情况发生时用来通知CPU的中断信息。

4. 编写int 9中断例程

```
1 ;编程：在屏幕中间依次显示“a”~“z”，并可以让人看清。在显示的过程中，按下'Esc'键后，改变显示
  的颜色。
2
3 ;完整功能代码：
4
5 assume cs:code
6
7 stack segment
8     db 128 dup (0)
9 stack ends
10
11 data segment
12     dw 0,0
13 data ends
14
15 code segment
16 start:
17     mov ax,stack
18     mov ss,ax
19     mov sp,128
20     mov ax,data
21     mov ds,ax
22     mov ax,0
23     mov es,ax
24
25     push es:[9*4]
26     pop ds:[0]
27     push es:[9*4+2]
28     pop ds:[2] ;将原来的int 9中断例程的入口地址保存在ds:0、ds:2单元中
29
30     mov word ptr es:[9*4], offset int9
31     mov es:[9*4+2], cs ;在中断向量表中设置新的int 9中断例程的入口地址
32
33 ;显示字符串
34     mov ax, 0b800h
35     mov es, ax
36     mov ah, 'a'
37 s:
38     mov es:[160*12+40*2], ah
39     call delay
40     inc ah
41     cmp ah, 'z'
42     jna s
43     mov ax,0
44     mov es,ax
45
46     push ds:[0]
47     pop es:[9*4]
48     push ds:[2]
49     pop es:[9*4+2] ;将中断向量表中int 9中断例程的入口恢复为原来的地址
50
51     mov ax,4c00h
52     int 21h
```

```

53
54 ;将循环延时的程序段写为一个子程序
55 delay:
56     push ax
57     push dx
58     mov dx, 2000h ;用两个16位寄存器来存放32位的循环次数
59     mov ax, 0
60 s1:
61     sub ax, 1
62     sbb dx, 0
63     cmp ax, 0
64     jne s1
65     cmp dx, 0
66     jne s1
67     pop dx
68     pop ax
69     ret
70
71 ;-----以下为新的int 9中断例程-----
72
73 int9:
74     push ax
75     push bx
76     push es
77
78     in al, 60h;从端口60h读出键盘的输入
79
80     pushf ;标志寄存器入栈
81
82     pushf
83     pop bx
84     and bh,11111100b
85     push bx
86     popf ;TF=0,IF=0
87
88     call dword ptr ds:[0] ;对int指令进行模拟,调用原来的int 9中断例程
89
90     cmp al,1
91     jne int9ret
92
93     mov ax,0b800h
94     mov es,ax
95     inc byte ptr es:[160*12+40*2+1] ;属性增加1,改变颜色
96
97 int9ret:
98     pop es
99     pop bx
100    pop ax
101    iret
102
103 code ends
104
105 end start

```

CPU对外设输入的通常处理方法

- (1) 外设的输入送入端口;
- (2) 向CPU发出外中断(可屏蔽中断)信息;
- (3) CPU检测到可屏蔽中断信息,如果IF=1,CPU在执行完当前指令后响应中断,执行相应的中断例程;
- (4) 可在中断例程中实现对外设输入的处理。

端口 和 中断机制 , 是CPU进行 I/O 的基础。

参考文章：

https://blog.csdn.net/qq_39654127/article/details/88698911 王爽《汇编语言》笔记（详细）