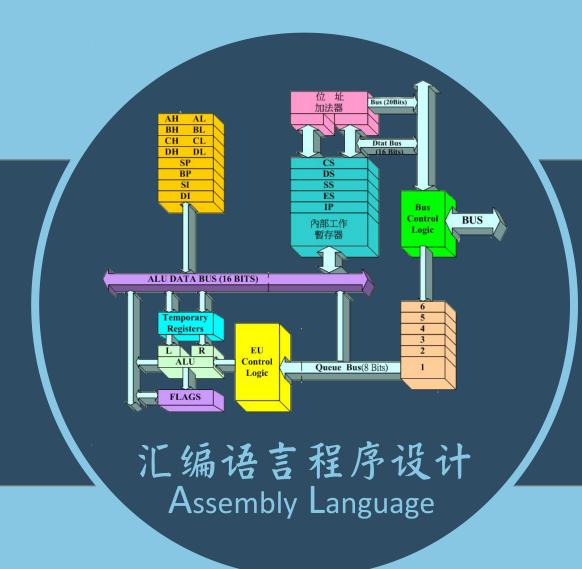
由int指令引发的中断

贺利坚 主讲



int n引起的中断

- □CPU内部产生的中断信息
 - ☞ 除法错误
 - @ 单步执行
 - ^小执行into指令
 - ⚠ 执行int 指令
- □int格式: int n, n为中断类型码
 - ₾ 功能:引发中断过程

assume cs:code

code segment

start:

....

mov ax,4c00h

int 21h

code ends

end start



- □CPU 执行int n指令,相当于引发一个 n号中断的中断过程,执行过程如下:
 - (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).
 - ——从此处转去执行n号中断的中断处理程序。

小小结:

- int 指令的最终功能和call指令相似,都是调用一段程序。
- 一般情况下,系统将一些具有一定功能的 子程序,以中断处理程序的方式提供给应 用程序调用。

编写供应用程序调用的中断例程

- □技术手段:编程时,可以用 int指令调用子程序
 - ⚠ 此子程序即中断处理程序,简称为中断例程
 - 可以自定义中断例程 ,实现特定功能
- □示例:中断7ch的中断例程 的编写和安装
 - 中断例程序需要按中断的运行机制的要求编写
 - ₾ 参考:中断0的中断例程



8086CPU的中断向量表:



示例:中断7ch的中断例程(1)

- □任务:编写程序
 - ─ 写7ch的中断例程,完成特定任务
 - □ 功能:求一个word型数据的平方
 - ◎ 参数: (ax)=要计算的数据
 - □ 返回值: dx, ax中存放结果的高、低16位
 - **编写安装中断例程**
 - △ 测试
 - □ 应用举例:求2*3456^2
- ⊒对中断例程的要求
 - (1)编程实现求平方功能的程序;
 - (2)安装程序,我们将其安装在0:200处;
 - (3)设置中断向量表,将程序的入口地址保存在7ch表项中,使其成为中断7ch的中断例程。

```
assume cs:code
code segment
start:mov ax,cs
    mov ds,ax
    mov si,offset sqr
    mov ax,0
    mov es,ax
                            end start
    mov di,200h
    mov cx,offset sqrend - offset sqr
    cld
    rep movsb
    mov ax.0
    mov es,ax
    mov word ptr es:[7ch*4],200h
    mov word ptr es:[7ch*4+2],0
    mov ax,4c00h
    int 21h
 sgr: mul ax
    iret
sgrend:nop
code ends
```

end start

assume cs:code
code segment
start: mov ax,3456
int 7ch; 计算(ax)^2
add ax,ax
adc dx, dx
mov ax,4c00h
int 21h
code ends
end start

执行int n时 pushf push CS push IP

执行iret时 pop IP pop CS popf

示例:中断7ch的中断例程(2)

■7ch的中断例程的任务

- □ 功能:将以 0结尾的字符串转化为大写。
- ◆参数: ds:si指向字符 串的首地址

```
assume cs:code
data segment
   db 'conversation',0
data ends
code segment
start: mov ax,data
   mov ds,ax
   mov si,0
   int 7ch
   mov ax,4c00h
   int 21h
code ends
end start
```

□中断处理程序的常规的步骤

- (1)保存用到的寄存器。
- (2)处理中断。
- (3)恢复用到的寄存器。
- (4)用 iret 指令返回。

```
capital:
    push cx
    push si
change: mov cl,[si]
    mov ch,0
    icxz ok
    and byte ptr [si],11011111b
    inc si
    imp short change
ok: pop si
    рор сх
    iret
capitalend:nop
```

```
assume cs:code
code segment
start:mov ax,cs
    mov ds,ax
    mov si,offset capital
    mov ax,0
    mov es,ax
    mov di,200h
    mov cx,offset capitalend - offset capital
    cld
    rep movsb
```

```
mov ax,0
mov es,ax
mov word ptr es:[7ch*4],200h
mov word ptr es:[7ch*4+2],0
```

mov ax,4c00h int 21h

code ends end start