

用汇编编写DOS下的内存驻留程序(2) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/259/2021_2022__E7_94_A8_E6_B1_87_E7_BC_96_E7_c98_259408.htm 基本原理 2.1

8086/8088 IBM PC中央处理单元(Central Processing Unit)是微处理器Inter 8088,8088是8086是小的版本.对于编写程序而言,两者几乎完全相同.两者之间的差别是在于:它们对外的沟通.8086和外界沟通时是经由16位的输入输出通道,内存存取也是每次以16位为单位,8088和8086极为相似,但是它和外界沟通时就必须经由16位的通道.

2.1.1 寄存器

8086/8088的结构简单,其中包含了一组一般用途的16位寄存器.AX,BX,CX,DX,BP,SI,DI.其中AX,BX,CX,DX还可以分成8位的寄存器,譬如:AX可分为AH,AL.BX可分为BH,BL.CX可分为CH,CL.DX可分为DH,DL.寄存器BP,SI,DI的用途也没有特别的限制,但是却不能分成两个字节.另外寄存器SP主要是用来当做堆栈指针.除此之外,还有四个非常重要的段寄存器(Segment Register):CS,DS,SS,ES.指令指针(Instruction pointer)IP是用来控制目前CPU执行到哪一个指令.8086设计时考虑到要和8位的CPU8080兼容.8位的计算机是使用两个字节(亦即16位)来定址,因此其定址空间可以达64K字节.16位的CPU在地址设定上选择了完全不同的方法.CPU以段(Segment)为单位,每一段范围内包括64K字节,而内存中则可以包含许多段.所以,操作系统可以在一个段内执行.而使用者的程序则可以在另一个段内执行.在一个段内,程序包可以把计算机视为只有64K字节内存空间.因此原先8位计算机上执行的程序就可以很容易地移植到16位计算机上.除此之外,内存段也可以彼此重叠,因而两个不同的程序就可以共用某一

块内存.段值是以寄存器来设定的,而实际的地址值则是把段值(16位)往左移4位,然后再加上16位的位移(Offset),因此构成20位的地址值.所以8086可以直接做20位的地址,也就是可能存取到一兆字节的内存.在这一兆字节的内存中,IBM PC保留了最前面的320K字节给系统的ROM BIOS和显示内存,因此使用者最多也就能使用640K字节.

2.1.2 寻址方式

寻址方式(Addressing mode)是一台计算机上许多复杂操作的关键所在.8086提供了以下几种寻址方法:立即寻址,内存间接寻址,寄存器间接寻址等.立即寻址,直接使用数字.内存间接寻址,数值存放在数据段中的某个位置. `mov bx,foo` `foo dw 5` 寄存器间接寻址.有两种寄存器可以使用在这种寻址方式下:基址寄存器(Base Register)和索引寄存器(Index Register).基址寄存器分别是BX和BP,索引寄存器则是SI和DI.在这种寻址方式下,寄存器存放了数据段中的地址值. `mov ax,0F000h` `mov es,ax` `mov si,0FFFEh` `mov dl,byte ptr es:[si]` 上面的程序使用间接寻址方式,由寄存器SI读出位于F000:FFFE位置的数据.寄存器间接存取时,最多只能使用一个基址寄存器各一个索引寄存器.以上的寻址方式可以做不同的结合,因此组合后的结果很多.

2.1.3 标志

8086有9个一位的标志(Flag),它们可以用指示CPU的各种状态.以下是9个标志的简介: CF(Carry Flag):CF为1时就表示算术运算的结果超出正确的长度. PF(Parity Flag):PF为1就表示使用偶校验,PF为0就表示使用奇校验. AF(Auxiliary Carry Flag):和CF相同,只是它使用在低4位的结果.AF通常都使用在20位的地址计算上. ZF(Zero Flag):ZF为1就表示运算结果是0,否则ZF就为0. SF(Sign Flag):SF为1就表示运算结果的最高位是1,否则SF就为0. TF(Trap Flag):TF为1,CPU就单步地执行,在这种模式下每完成一个指令

就发生一个特殊的中断. IF(Interrupt Enable Flag):IF为1,允许CPU接收外界的中断,否则IF就为0. DF(Direction Flag):这个标志使用在循环指令,譬如:MOVS,MOVSB,MOVSW,CMPS,CMPSB和CMPSW.如果DF为1,循环运行时就使地址值往前增加.如果DF为0,则使地址往后减少. OF(Over Flag):OF为1,表示一个考虑正负号的运算超出了正确的字节的长度.

2.1.4 循环

所有的循环指令都是以CX作为计数器.一个循环会反复地执行直到CX等于某一特定值为止.以下的程序就是利用反复地相加,完成两个数的相乘.

```
mov ax,0
mov cx,4
next: add ax,6
loop next
```

在上面的程序中,LOOP指令执行时会把CX减1,并且检查CX的内容.如果CX等于0,就转移到下一条指令,否则就跳到NEXT标示的地方执行.也可以用下面的程序完成相同的功能:

```
mov ax,0
mov cx,4
next: add ax,6
dec cx
cmp cx,0
jne next
```

2.1.5 内存的数据结构

8088是以字节为存取数据的基本单位.计算机的存储结构是8位的字节,但是CPU本身处理数据则是以16位为单位.在内存中,都遵循一个原则,即:高高低低的存储方式.高字节对应高地址,低字节对应低地址.下面是一个简单程序,在AX中放入一个字节的內容并显示:

```
cseg segment org 100h
assume cs:cseg,ds:cseg
start: mov bx,cs
mov ds,bx
mov ah,H
mov al,L
mov test,ax
mov al,[si]
.First byte of test
call dchar
mov al,[si+1]
.Second byte of test
call dchar
ret
.Display the character contained in AL
dchar proc
push ax
push bx
mov bh,1
mov ah,0eh
int 10h
pop bx
pop ax
ret
dchar endp
test dw ?
cseg ends
end start
```

100Test 下载频道开通,各类考试题目直接下载。详细请访问 www.100test.com