

3.1 什么叫寻址方式？8086/8088 CPU 共有哪几种寻址方式？

解：寻址方式主要是指获得操作数所在地址的方法。8086/8088 CPU 具有：立即寻址、直接寻址、寄存器寻址、寄存器间接寻址、寄存器相对寻址、基址—变址寻址、基址—变址—相对寻址以及隐含寻址等 8 种寻址方式。

3.2 设 $DS = 6000H$, $ES = 2000H$, $SS = 1500H$, $SI = 00A0H$, $BX = 0800H$, $BP = 1200H$, 字符常数 VAR 为 $0050H$ 。请分别指出下列各条指令源操作数的寻址方式？并计算除立即寻址外的其他寻址方式下源操作数的物理地址。

- | | |
|---------------------|----------------------------|
| (1) $MOV\ AX, BX$ | (2) $MOV\ DL, 80H$ |
| (3) $MOV\ AX, VAR$ | (4) $MOV\ AX, VAR[BX][SI]$ |
| (5) $MOV\ AL, 'B'$ | (6) $MOV\ DI, ES: [BX]$ |
| (7) $MOV\ DX, [BP]$ | (8) $MOV\ BX, 20H[BX]$ |

解：(1) 寄存器寻址。因源操作数是寄存器，故寄存器 BX 就是操作数的地址。

(2) 立即寻址。操作数 $80H$ 存放于代码段中指令码 MOV 之后。

(3) 立即寻址。

(4) 基址—变址—相对寻址。

$$\begin{aligned}\text{操作数的物理地址} &= DS \times 16 + SI + BX + VAR \\ &= 60000H + 00A0H + 0800H + 0050H \\ &= 608F0H\end{aligned}$$

(5) 立即寻址。

(6) 寄存器间接寻址。

$$\text{操作数的物理地址} = ES \times 16 + BX = 20000H + 0800H = 20800H$$

(7) 寄存器间接寻址。

$$\text{操作数的物理地址} = SS \times 16 + BP = 15000H + 1200H = 16200H$$

(8) 寄存器相对寻址。

$$\text{操作数的物理地址} = DS \times 16 + BX + 20H = 60000H + 0800H + 20H = 60820H$$

3.3 假设 $DS = 212AH$, $CS = 0200H$, $IP = 1200H$, $BX = 0500H$, 位移量 $DATA = 40H$, $[217A0H] = 2300H$, $[217E0H] = 0400H$, $[217E2H] = 9000H$, 试确定下列转移指令的转移地址。

- (1) JMP BX
- (2) JMP WORD PTR[BX]
- (3) JMP DWORD PTR[BX+DATA]

解：转移指令分为段内转移和段间转移，根据其寻址方式的不同，又有段内的直接转移和间接转移，以及段间的直接转移和间接转移地址。对直接转移，其转移地址为当前指令的偏移地址（即 IP 的内容）加上位移量或由指令中直接得出；对间接转移，转移地址等于指令中寄存器的内容或由寄存器内容所指向的存储单元的内容。

- (1) 段内间接转移。转移目标的物理地址 = $CS \times 16 + BX$

$$= 02000H + 0500H = 02500H$$
- (2) 段内间接转移。转移目标的物理地址 = $CS \times 16 + [BX]$

$$= CS + [217A0H]$$

$$= 02000H + 2300H = 04300H$$
- (3) 段间间接转移。转移目标的物理地址 = $[BX + DATA]$

$$= [217E2H] \times 16 + [217E0H]$$

$$= 90000H + 0400H = 90400H$$

3.4 试说明指令 MOV BX,5[BX]与指令 LEA BX,5[BX]的区别。

解：前者是数据传送类指令，表示将数据段中以 $(BX+5)$ 为偏移地址的 16 位数据送寄存器 BX。后者是取偏移地址指令，执行的结果是 $BX = BX + 5$ ，即操作数的偏移地址为 $BX + 5$ 。

3.5 设堆栈指针 SP 的初值为 2300H，AX = 50ABH，BX = 1234H。执行指令 PUSH AX 后，SP = ?，再执行指令 PUSH BX 及 POP AX 之后，SP = ? AX = ? BX = ?

解：堆栈指针 SP 总是指向栈顶，每执行一次 PUSH 指令 SP - 2，执行一次 POP 指令 SP + 2。所以，执行 PUSH AX 指令后，SP = 22FEH；再执行 PUSH BX 及 POP AX 后，SP = 22FEH，AX = BX = 1234H。

3.6 判断下列指令是否正确，若有错误，请指出并改正。

- (1) MOV AH,CX
- (2) MOV 33H,AL
- (3) MOV AX,[SI][DI]
- (4) MOV [BX],[SI]
- (5) ADD BYTE PTR[BP],256
- (6) MOV DATA[SI],ES:AX
- (7) JMP BYTE PTR[BX]
- (8) OUT 230H,AX
- (9) MOV DS,BP
- (10) MUL 39H

解：(1) 指令错。两操作数字长不相等。

(2) 指令错。MOV 指令不允许目标操作数为立即数。

(3) 指令错。在间接寻址中不允许两个间址寄存器同时为变址寄存器。

(4) 指令错。MOV 指令不允许两个操作数同时为存储器操作数。

(5) 指令错。ADD 指令要求两操作数等字长。

(6) 指令错。源操作数形式错，寄存器操作数不加段重设符。

(7) 指令错。转移地址的字长至少应是 16 位的。

(8) 指令错。对输入输出指令，当端口地址超出 8 位二进制数的表达范围时（即寻址的端口数超出 256 时），必须采用间接寻址。

(9) 指令正确。

(10) 指令错。MUL 指令不允许操作数为立即数。

3.7 已知 AL=7BH,BL=38H,试问执行指令 ADD AL,BL 后,AF、CF、OF、PF、SF 和 ZF 的值各为多少?

解: AF=1,CF=0,OF=1,PF=0,SF=1,ZF=0。

3.8 试比较无条件转移指令、条件转移指令、调用指令和中断指令有什么异同?

解: 无条件转移指令的操作是无条件地使程序转移到指定的目标地址,并从该地址开始执行新的程序段,其转移的目标地址既可以是在当前逻辑段,也可以是在不同的逻辑段;条件转移指令是在满足一定条件下使程序转移到指定的目标地址,其转移范围很小,只能当前逻辑段的-128~+127 地址范围内。

调用指令是用于调用程序中常用到的功能子程序,是在程序设计中就设计好的。根据所调用过程入口地址的位置可将调用指令分为段内调用(入口地址在当前逻辑段内)和段间调用。在执行调用指令后,CPU 要保护断点。对段内调用是将其下一条指令的偏移地址压入堆栈,对段间调用则要保护其下一条指令的偏移地址和段基地址,然后将子程序入口地址赋给 IP(或 CS 和 IP)。

中断指令是因一些突发事件而使 CPU 暂时中止它正在运行的程序,转去执行一组专门的中断服务程序,并在执行完后返回原被中止处继续执行原程序,它是随机的。在响应中断后 CPU 不仅要保护断点(即 INT 指令下一条指令的段地址和偏移地址),还要将标志寄存器 FLAGS 压入堆栈保存。

3.9 试判断下列程序执行后,BX 中的内容。

```
MOV CL,3
MOV BX,0B7H
ROL BX,1
ROR BX,CL
```

解: 该程序段是首先将 BX 内容不带进位位循环左移 1 位,再循环右移 3 位。即相当于将原 BX 内容不带进位位循环右移 2 位,故结果为: BX=0C02DH。

3.10 按下列要求写出相应的指令或程序段:

- (1) 写出两条使 AX 内容为 0 的指令;
- (2) 使 BL 寄存器中的高 4 位和低 4 位互换;
- (3) 屏蔽 CX 寄存器的 b11、b7 和 b3 位;
- (4) 测试 DX 中的 b0 和 b8 位是否为 1。

解: (1)

```
MOV AX,0
XOR AX,AX           ;AX 寄存器自身相异或,可使其内容清 0
```

(2)

```
MOV CL,4
ROL BL,CL           ;将 BX 内容循环左移 4 位,可实现其高 4 位和低 4 位的互换
```

(3)

AND CX,0F777H ;将 CX 寄存器中需屏蔽的位"与"0。也可用"或"指令实现

(4)

AND DX,0101H ;将需测试的位"与"1,其余"与"0 屏蔽掉

CMP DX,0101H ;与 0101H 比较

JZ ONE ;若相等则表示 b0 和 b8 位同时为 1

⋮

3.11 分别指出以下两个程序段的功能:

(1)

MOV CX,10

LEA SI,FIRST

LEA DI,SECOND

STD

REP MOVSB

(2)

CLD

LEA DI,[1200H]

MOV CX,00FFH

XOR AX,AX

REP STOSW

解: (1) 该段程序的功能是: 将数据段中 FIRST 为首地址的 10 个字节数据按减地址方向传送到附加段 SECOND 为首址的单元中。

(2) 将附加段中偏移地址为 1200H 单元开始的 00FFH×2 个单元清 0。

3.12 执行以下两条指令后,标志寄存器 FLAGS 的 6 个状态位各为什么状态?

MOV AX,84A0H

ADD AX,9460H

解: 执行 ADD 指令后,6 个状态标志位的状态分别为: CF=1, ZF=0, SF=0, OF=1, PF=1, AF=0。

3.13 将+46 和-38 分别乘以 2,可用什么指令来完成? 如果除以 2 呢?

解: 因为对二进制数,每左移一位相当于乘以 2,右移一位相当于除以 2。所以,将+46 和-38 分别乘以 2,可分别用逻辑左移指令(SHL)和算术左移指令(SAL)完成。SHL 指令针对无符号数,SAL 指令针对有符号数。

当然,也可以分别用无符号数乘法指令 MUL 和有符号数乘法指令 IMUL 完成。

如果是除以 2,则进行相反操作,即用逻辑右移指令 SHR 或无符号数除法指令 DIV 实现+46 除以 2 的运算,用算术右移指令 SAR 或有符号数除法指令 IDIV 实现-38 除以 2 的运算。

3.14 已知 $AX=8060H$, $DX=03F8H$, 端口 $PORT1$ 的地址是 $48H$, 内容为 $40H$; $PORT2$ 的地址是 $84H$, 内容为 $85H$ 。请指出下列指令执行后的结果。

- (1) `OUT DX,AL`
- (2) `IN AL,PORT1`
- (3) `OUT DX,AX`
- (4) `IN AX,48H`
- (5) `OUT PORT2,AX`

解: (1) 将 $60H$ 输出到地址为 $03F8H$ 的端口中;
(2) 从 $PORT1$ 读入一个字节数据, 执行结果: $(AL)=40H$;
(3) 将 $AX=8060H$ 从地址为 $03F8H$ 的端口输出;
(4) 由 $48H$ 端口读入 16 位二进制数;
(5) 将 $8060H$ 从地址为 $85H$ 的端口输出。

3.15 试编写程序, 统计 $BUFFER$ 为起始地址的连续 200 个单元中 0 的个数。

解: 将 $BUFFER$ 为首地址的 200 个单元的数依次与 0 进行比较, 若相等则表示该单元数为 0, 统计数加 1; 否则再取下一个数比较, 直到 200 个单元数全部比较完毕为止。程序如下:

```
LEA SI,BUFFER      ;取 BUFFER 的偏移地址
MOV CX,200          ;数据长度送 CX
XOR BX,BX           ;存放统计数寄存器清 0
AGAIN: MOV AL,[SI]   ;取一个数
        CMP AL,0     ;与 0 比较
        JNE GOON     ;不为 0 则准备取下一个数
        INC BX       ;为 0 则统计数加 1
GOON: INC SI         ;修改地址指针
        LOOP AGAIN   ;若未比较完则继续比较
HLT
```

3.16 写出完成下述功能的程序段:

- (1) 从地址为 $DS:0012H$ 的存储单元中传送一个数据 $56H$ 到 AL 寄存器;
- (2) 将 AL 中的内容左移两位;
- (3) AL 的内容与字节单元 $DS:0013H$ 中的内容相乘;
- (4) 乘积存入字单元 $DS:0014H$ 中。

解: (1)

```
MOV BYTE PTR[0012H],56H
MOV AL,[0012H]
```

(2)

```
MOV CL,2
SHL AL,CL
```

(3)

```
MUL  BYTE PTR[0013H]
```

(4)

```
MOV  [0014H],AX
```

3.17 若 $AL=96H$, $BL=12H$, 在分别执行指令 `MUL` 和 `IMUL` 后, 其结果是多少?
 $OF=?$ $CF=?$

解: `MUL` 是无符号数的乘法指令, 它将两操作数视为无符号数; `IMUL` 是有符号数的乘法指令, 此时, 两操作数被看作有符号数。在该题中, $(AL)=96H$, 其最高位为 1, 是负数。 `IMUL` 指令的执行原理是先求出它的真值(即对它求补), 再做乘法运算。

执行 `MUL BL` 指令后, $AX=0A8CH$, $CF=OF=1$ 。执行 `IMUL BL` 指令后, $AX=F88CH$, $CF=OF=1$ 。