

第二章 微型计算机指令系统

题 2-1 试分别说明以下各指令的源操作数属于何种寻址方式。

- 答：1、MOV AX, [SP] ——寄存器间接寻址
2、MOV DS, AX ——寄存器寻址
3、MOV DI, 0FF00H ——立即数寻址
4、MOV BX, [2100H] ——直接寻址
5、MOV CX, [SI+5] ——变址寻址
6、MOV AX, TABLE[BP][DI] ——基址加变址寻址
7、MOV DX, COUNT[BX] ——基址寻址

题 2-2 已知有关寄存器中的内容为, (DS) =0F100H, (SS) =0A100H, (SI) =1000H, (DI) =2000H, (BX) =3000H, (BP) =4000H。偏移量 TABLE=0AH, COUNT=0BH。说明题 2-1 中第 1 小题和第 4~7 小题指令源操作数的物理地址。

- 答：1、物理地址=SS×16+SP=0A1000H+4000H=A5000H
4、物理地址=DS×16+2100H=F3100H
5、物理地址=DS×16+SI+5H=F1000H+1000H+5H=F2005H
6、EA=BP+DI+TABLE=600AH
物理地址=DS×16+EA=F1000H+600AH=F700AH
7、EA=BX+COUNT=300BH
物理地址=DS×16+EA=F1000H+300BH=F400BH

题 2-3 某一个存储单元的段地址为 ABCDH, 偏移地址为 ABCDH, 试说明其物理地址是什么; 而另一个存储单元的物理地址为 F1000H, 偏移地址为 FFF0H, 试说明其段地址是什么。

- 答：ABCD0H+ABCDH=B689DH——物理地址
F1000H-FFF0H=E1010H, 所以段地址为:E101H。

题 2-4 分别采用三种不同寻址方式的指令将偏移地址为 5000H 的存储单元的一个字传送到 6000H 单元, 要求源操作数和目标操作数分别采用以下寻址方式:

- 1、直接寻址
- 2、寄存器间接寻址
- 3、变址寻址

- 答：1、MOV AX, [5000H]
MOV [6000H], AX
2、MOV SI, 5000H
MOV DI, 6000H
MOV BX, [SI]
MOV [DI], BX
3、MOV AX, [SI+5000H]
MOV [SI+6000H], AX

题 2-5 说明以下 8086/8088 指令是否正确, 如果不正确, 简述理由。

- 1、MOV AL, SI
- 2、MOV [1001H], [1000H]
- 3、MOV DS, 2000H
- 4、MOV CS, AX
- 5、PUSH 5000H
- 6、POP DL
- 7、IN AX, 1234H
- 8、XCHG BX, 0F000H

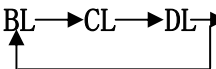
答：1、错。源数据和目标数据类型不匹配。
2、错。不能用一条 MOV 指令完成两个存储单元的数据传送。
3、错。不能直接向 DS 中送立即数。
4、错。CX 寄存器不能作为目的操作数。
5、错。数据类型不符合语法。
6、错。POP 指令以字为操作数。
7、错。端口地址 8 位，超出范围。
8、错。源数据类型错误。

题 2-6 阅读以下程序段，说明其运行结果：

- 1、MOV DX, 0F100H
MOV DH, [DX]
MOV AH, [DX]
初值：(F100H) = 'A', (4100H) = 'B'
结果：(AH) = 41H
- 2、MOV SI, 2100H
MOV [2800H], SI
MOV SP, 2800H
POP DI
结果：(DI) = 2100H, (SP) = 2802H。
- 3、MOV SI, 2000H
MOV DI, 3000H
MOV SP, 0FF00H
PUSH SI
PUSH DI
XCHG SI, DI
MOV AL, [SI]
MOV BL, [DI]
POP SI
POP DI
初值：(2000H) = '1', (3000H) = 'a'
结果：(SI) = 3000H, (DI) = 2000H,
(AL) = 61H, (BL) = 31H,
(SP) = FF00H。

题 2-7 试用 8086/8088 指令编写程序段，分别实现以下要求：

- 1、将 AL 和 BL 寄存器的内容互换
- 2、将 1000H 和 1001H 内存单元中的字互换
- 3、将 2000H 和 2100H 内存单元中的字互换
- 4、将 BL, CL 和 DL 三个寄存器的内容互换, 即



答: 1、 XCHG AL , BL
 2、 MOV BL , [1000H]
 MOV AL , [1001H]
 XCHG AL , BL
 3、 MOV BX, [2000H]
 MOV AX, [2100H]
 XCHG AX, BX
 4、 XCHG BL , CL
 XCHG CL , DL
 XCHG DL , BL

题 2-8 将首地址为 3000H 的 100 个存储单元的内容传送到首地址为 3100H 的内存区, 要求分别使用以下指令:

- 1、一般传送指令 MOV
- 2、串操作指令 MOVS, 但不加重重复前缀 REP
- 3、加重重复前缀的串操作指令 REP MOVS

答: 1、 MOV SI , 3000H
 MOV DI , 3100H
 MOV CX , 100
 CIRCLE: MOV AL , [SI]
 MOV [DI] , AL
 INC SI
 INC DI
 LOOP CIRCLE
 2、 MOV SI , 3000H
 MOV DI , 3100H
 MOV CX , 100
 CIRCLE: MOVSB
 LOOP CIRCLE
 3、 MOV SI , 3000H
 MOV DI , 3100H
 MOV CX , 100
 CLD
 REP MOVSB

题 2-9 说明以下 8086/8088 指令是否正确, 如果不正确, 简述理由

- 1、ADD BL , F100H
- 2、SUB ES , 20H
- 3、AND 0FH, AL
- 4、CMP [SI], [DI]

- 5、INC 2000H
 6、MUL BL , CL
 7、DIV 08H
 8、SAL AX , 5

答:

- 1、错。源操作数和目标操作数的数据类型不匹配。
- 2、错。段寄存器不能参加算术运算。
- 3、错。源操作数和目标操作数位置颠倒。
- 4、错。不能同时为存储器。
- 5、错。操作数类型错误。
- 6、错。其中一个操作数是隐含的。
- 7、错。必须是寄存器或存储器操作数。
- 8、错。超过 1 的循环次数要通过赋值 CL 来完成。

题 2-10 已知寄存器 AL 和 BL 的内容分别如下, 试分析执行 ADD AL , BL 指令后, 寄存器 AL 以及标志位 CF, ZF, SF, AF, OF 和 PF 的内容。

- 1、(AL) = 45H , (BL) = 31H
- 2、(AL) = 7AH , (BL) = 56H
- 3、(AL) = F2H , (BL) = 8DH
- 4、(AL) = B1H , (BL) = F8H
- 5、(AL) = 37H , (BL) = C9H

答:

题号	AL	CF	ZF	SF	AF	OF	PF
1、	76H	0	0	0	0	0	0
2、	D0H	0	0	1	1	1	0
3、	7FH	1	0	0	0	1	0
4、	A9H	1	0	1	0	1	1
5、	00H	1	1	0	1	0	1

(注: 考虑 CF 时要当操作数为无符号数, 考虑 OF 时要当操作数为有符号数, 下同)

题 2-11 已知寄存器 AL 和 BL 的内容分别如下, 试分析执行 SUB AL , BL 指令以后, 寄存器 AL 以及标志位 CF, ZF, SF, AF, OF 和 PF 的内容。

- 1、(AL) = 96H , (BL) = 42H
- 2、(AL) = 27H , (BL) = 38H
- 3、(AL) = 6CH , (BL) = A1H
- 4、(AL) = B4H , (BL) = E7H
- 5、(AL) = 1DH , (BL) = E5H

答:

题号	AL	CF	ZF	SF	AF	OF	PF
1、	54H	0	0	0	0	1	0
2、	EFH	1	0	1	1	0	0
3、	CBH	1	0	1	0	1	0
4、	CDH	1	0	1	1	0	0
5、	38H	1	0	0	0	0	0

题 2-12 阅读以下程序段：

```
MOV    SI , 4000H
MOV    DI , 4100H
MOV    AL , [SI]
ADD    AL , [DI]
DAA
MOV    [DI] , AL
MOV    AL , [SI+1]
MOV    AL , [DI+1]
DAA
MOV    [DI+1] , AL
```

1、根据给定的初值，说明运行结果。

初值： (4000H) = 63H , (4001H) = 54H
(4100H) = 88H , (4101H) = 29H

结果： (4100H) = 51H , (4101H) = 84H 。

2、在以上程序段中，当执行 ADD AL, [DI] 指令，但尚未执行 DAA 指令时，有关寄存器和标志位的内容为：

(AL) = EBH , (CF) = 0 , (ZF) = 0 , (SF) = 1 ,
(AF) = 0 , (OF) = 0 , (PF) = 1 。

3、在以上程序段中，当第一个执行 DAA 指令后，有关寄存器和标志位的内容为：

(AL) = 51H , (CF) = 1 , (ZF) = 0 , (SF) = 0 ,
(AF) = 1 , (OF) = 0 , (PF) = 0 。

4、如果在以上程序段中去掉两条 DAA 指令，但初值不变，则运行结果为：
(4100H) = EBH , (4101H) = 7DH 。

题 2-13 阅读以下程序段：

```
START:  LEA    BX , TABLE
        MOV    CL , [BX]
LOOPER: INC    BX
        MOV    AL , [BX]
        CMP    AL , 0AH
        JNC    X1
        ADD    AL , 30H
        JMP    NEXT
X1:      ADD    AL , 37H
NEXT:    MOV    [BX] , AL
        DEC    CL
        JNZ    LOOPER
```

1、假设从地址 TABLE 开始，10 个存储单元的内容依次为：05H, 01H, 09H, 0CH, 00H, 0FH, 03H, 0BH, 08H, 0AH

依次写出运行以上程序段以后，从地址 TABLE 开始的 10 个存储单元的内容。

2、简单扼要说明以上程序段的功能。

答：1、

35H
31H
39H
43H
30H
46H
33H
42H
38H
41H

2、此程序段的功能是将这些存储单元的数据（十六进制）转化为相应的 ASCII 码。

题 2-14 用一条或几条 8086/8088 指令实现以下要求：

- 1、将 AL 寄存器清零，但进位标志 CF 不变。
- 2、将 AL 寄存器和进位标志 CF 同时清零。
- 3、将进位标志位清零，但 AL 寄存器内容不变。
- 4、将 AL 寄存器中内容的第 0，2，4，6 位取反，其余位不变。
- 5、将 AL 寄存器中内容的高 4 位清零，低 4 位保留。
- 6、将 AL 寄存器中的带符号数（以补码形式存放）取绝对值后存入 CL 寄存器。
- 7、统计 AX 寄存器的内容中 ‘1’ 的个数，将统计结果存入 CL 寄存器。
- 8、两个不压缩的 BCD 码分别存放在 3000H 和 3001H 单元的低 4 位，高 4 位均为零。试将两个不压缩的 BCD 码组合成为一个压缩的 BCD 码，前者放在低 4 位，后者放在高 4 位，存放到 3002H 单元。例如：
初值位： (3000H) = 07H, (3001H) = 05H
要求结果为：(3002H) = 57H
- 9、将 4000H 和 4001H 单元中的两个 ASCII 码分别转化成相应的十六进制数，然后共同存放到 4002H 单元，前者放在低 4 位，后者放在高 4 位。例如：
初值位： (4000H) = 42H = ‘B’ , (4001H) = 36H = ‘6’
要求结果为：(4002H) = 6BH
- 10、将 BL 寄存器中的无符号数乘 128，高位放在 BH 寄存器，低位放在 BL 寄存器，要求执行速度尽量快。
- 11、将 CL 寄存器中的带符号数乘 16，高位放在 CH 寄存器，低位放在 CL 寄存器，要求执行速度尽量快。
- 12、一个 4 位十进制数以压缩 BCD 码的形式存放在偏移地址为 DATA 和 DATA + 1 的内存单元中，DATA 单元存放个位和十位，DATA + 1 单元存放百位和千位。要求将其转换为相应的 ASCII 码，存放到以 BUFF 为首地址的 4 个内存单元。例如：

初值为： (DATA) = 47H , (DATA + 1) = 92H
要求结果为：(BUFF) = ‘7’ , (BUFF+1) = ‘4’
(BUFF+2) = ‘2’ , (BUFF+3) = ‘9’

答：1、 MOV AL, 0
2、 XOR AL, AL
3、 AND AL, AL

第二章 微型计算机指令系统

```
4、 XOR    AL, 55H
5、 AND    AL, 0FH
6、        TEST    AL, 80H      ; 判断是否为负数
          JNZ     NEXT          ; 如果是负数则跳转 NEXT
          MOV     BL, AL
NEXT:     NEG     AL              ; 负数求补, 即绝对值
          MOV     BL, AL
7、        XOR     BL, BL
          MOV     CL, 0FH ; 循环次数
Circle:   SHL     AX, 1          ; 左移一位, 首位移进 CF, ROL 也可以实现
          JC      NEXT1         ; 判断 CF 是否为 1
          JMP     NEXT2
NEXT1:    INC     BL
NEXT2:    LOOP    Circle
          MOV     CL, BL
8、 MOV     BL, [3000H]
          MOV     AL, [3001H]
          MOV     CL, 04H
          SHL     AL, CL
          ADD     AL, BL
          MOV     [3002H], AL
9、        XOR     AX, AX
          XOR     BX, BX
          MOV     AL, [4001H]
          MOV     BL, [4000H]
          CMP     AL, 39H
          JA      L1
          SUB     AL, 30H
X1:       CMP     BL, 39H
          JA      L2
          SUB     BL, 30H
X2:       MOV     CL, 04H
          SHL     AL, CL
          ADD     AL, BL
          MOV     [4002H], AL
          MOV     AH, 4CH
          INT     21H
L1:       SUB     AL, 37H
          JMP     X1
L2:       SUB     BL, 37H
          JMP     X2
10、      XOR     AX, AX
          XOR     BH, BH
          MOV     CL, 07H
```

```

11、    SHL     BX, CL
        XOR     AX, AX
        XOR     CH, CH
        MOV     AX, CX
        MOV     CL, 04H
        SHL     AX, CL
        MOV     CX, AX

12、    LEA     SI, DATA
        LEA     DI, BUFF
        MOV     CL, 02H
LOAD:   MOV     AL, [SI]
        MOV     BL, AL
        AND     AL, 0FH
        ADD     AL, 30H
        MOV     [DI], AL
        INC     DI
        MOV     AL, BL
        PUSH    CX
        MOV     CL, 04H
        ROR     AL, CL
        POP     CX
        AND     AL, 0FH
        ADD     AL, 30H
        MOV     [DI], AL
        INC     DI
        INC     SI
        LOOP    LOAD
    
```

题 2-15 假设 AL 寄存器的内容为 FEH, BL 寄存器的内容为 04H, 试问:

1、执行 MUL BL 指令后, AX 寄存器的内容是什么? 标志位 OFC 和 CF 的值是什么?

2、执行 IMUL BL 指令后, AX 寄存器的内容是什么? 标志位 OFC 和 CF 的值是什么?

答: 1、AX=3F8H ; CF=0F=1

2、AX=FFF8H ; CF=0F=0

题 2-16 假设 AX 寄存器的内容为 0101H, BL 寄存器的内容为 0AH, 试问执行 DIV BL 指令后, AL 寄存器的内容是什么? AH 寄存器的内容是什么?

答: AL=19H ; AH=07H

题 2-17 假设 AX 寄存器的内容为 FFF5H, BL 寄存器的内容为 FEH, 试问执行 IDIV BL 指令后, AL 寄存器的内容是什么? AH 寄存器的内容是什么?

答: AL=05H ; AH=0FFH

题 2-18 已知被减数和减数均为包括 6 个字节的十六进制数, 分别存在首地址为

DATA1 和 DATA2 的内存区，低位在前，高位在后。试用 8086/8088 指令编写减法的程序段，要求相减以后得到的结果存放在首地址为 DATA3 的内存区。

答：

```
MOV     CX, 6
MOV     SI, 0
CLD
LOOPER: MOV     AL, DATA1[SI]
        SBB     AL, DATA2[SI]
        MOV     DATA3[SI], AL
        INC     SI
        DEC     CX
        LOOP    LOOPER
```

题 2-19 如果题 2-18 中的被减数和减数均为包括 6 个字节的压缩 BCD 码(相当于 12 位十进制数)，试重新编写减法的程序段。

答：

```
MOV     CX, 6
MOV     SI, 0
CLD
LOOPER: MOV     AL, DATA1[SI]
        SBB     AL, DATA2[SI]
        DAS
        MOV     DATA3[SI], AL
        INC     SI
        DEC     CX
        LOOP    LOOPER
```

题 2-20 有一个数据块中存放了若干 8 位无符号数，数据块的长度存放在 BLOCK 单元，数据块本身从 BLOCK + 1 单元开始存放，编写程序段找出数据块中最大的无符号数，存放到 MAX 单元。

答：

```
XOR     CX, CX
MOV     CX, [BLOCK]
LEA     SI, BLOCK+1
MOV     AL, [SI]
P1:     INC     SI
        CMP     AL, [SI]
        JAE     P2
        MOV     AL, [SI]
P2:     DEC     CX
        JNZ     P1
        LEA     SI, MAX
        MOV     [SI], AL
```

题 2-21 以 BUFFER 为首地址的内存区存放了 100 个 16 位带符号数，编写程序段比较它们的大小，找出其中最小的带符号数，存进 MIN 和 MIN + 1 单元。

答：

```
LEA     SI, BUFFER
```

```

        MOV     CX, 100
        CLD
        LODSW
        MOV     MIN, AX
        DEC     CX
NEXT:    LODSW
        CMP     AX, MIN
        JL      LESS
        JMP     GOON
LESS:    MOV     MIN, AX
GOON:    LOOP    NEXT
    
```

题 2-22 两个字符串的长度均为 100，首地址分别为 STRING1 和 STRING2，比较两个字符串是否完全相同，如果相同，将 BL 寄存器置为 00H；如果不完全相同，将 BL 寄存器置为 FFH，并将第一个字符串中的第一个不相同字符的地址放在 SI 寄存器。

答：

```

        LEA     SI, STRING1
        LEA     DI, STRING2
        MOV     CX, 100
        CLD
        REPE    CMPSB
        JCXZ    MATCH
        DEC     SI
        MOV     BL, FFH
MATCH:    MOV     BL, 00H
    
```

题 2-23 一个数据块的首地址为 DATA，结束符为 ‘\$’，编写程序段统计数据块中分别等于正值、负值和零的数据个数，分别存入 PLUS、MINUS 和 ZERO 单元。

答：

```

        XOR     AL, AL
        MOV     PLUS, AL
        MOV     MINUS, AL
        MOV     ZERO, AL
        LEA     SI, DATA
        CLD
CHECK:    LODSB
        OR      AL, AL
        JS      X1
        JZ      X2
        CMP     AL, 24H      ; “$” 的 ASCII 码是 24H
        JNZ     X3
        JMP     END
X1:       INC     MINUS
        JMP     CHECK
X2:       INC     ZERO
    
```

	JMP	CHECK	
X3:	INC	PLUS	
	JMP	CHECK	
END:	...		;后续处理程序
	...		
	...		
	...		