

## 第一节 数制转换

+74、-74 的原码、反码与补码表示。

## 第二节 有符号数据的表示（补码）、溢出

计算各条指令执行后 AL 的内容。

1. MOV AL, 66; ADD AL, -52
2. MOV AL, -66; ADD AL, 52
3. MOV AL, 60H, ADD AL, 50H

## 第三节 微型计算机组成、FR 标志位的计算（8086 CPU 内部结构）

1. 冯·诺依曼结构微型计算机的组成。
2. 计算各指令执行后寄存器的内容，ADD 指令执行后 FR 的各状态标志位的值。  
MOV BL, 70H ; ADD BL, 30H
3. 计算各指令执行后寄存器的内容，ADD 指令执行后 FR 的各状态标志位的值。  
MOV AX, -2A8H; ADD AX, 100H
4. 设 AL=57H，则执行 SUB AL, 0F0H 后，OF=\_\_\_\_\_，ZF=\_\_\_\_\_，CF=\_\_\_\_\_，SF=\_\_\_\_\_。

## 第四节 8086CPU 内部结构、引脚、最小系统配置、内存分段结构（逻辑地址→物理地址）

1. 8086CPU 可以分成\_\_\_\_\_和\_\_\_\_\_两部分同时工作。
2. 逻辑地址有两部分构成，分别为\_\_\_\_\_和\_\_\_\_\_。
3. 当引脚 MN/MX 接+5V 时，8086 工作在\_\_\_\_\_模式；在此工作模式构成的系统中，地址锁存器 8282 共有

\_\_\_\_片，需要锁存\_\_\_\_条信号。

4. 8086 复位后从\_\_\_\_逻辑地址开始执行程序，该地址所对应的物理地址是\_\_\_\_\_。

## 第五节 读写时序、内存分体结构

1. 当引脚  $\overline{MN}/\overline{MX}$  接+5V 时, 8086 工作在\_\_\_\_\_模式; 在此工作模式构成的系统中, 地址锁存器 8282 共有\_\_\_\_\_片, 需要锁存\_\_\_\_\_条信号, 其中\_\_\_\_\_用于控制偶存储体是否工作。
2. 设  $DS=2000H$ , 则执行指令 `MOV AL, [100H]` 过程中, 在总线周期的第\_\_\_\_\_个时钟周期 8086 输出地址, 地址总线上将出现如下地址: \_\_\_\_\_, 它由 BIU 中的计算生成。
3. 当执行指令 `OUT 70H, AL` 时,  $M/\overline{IO}=$ \_\_\_\_\_,  $\overline{RD}=$ \_\_\_\_\_;  $\overline{WR}=$ \_\_\_\_\_; AL 中的数据通过数据总线的\_\_\_\_\_ (高、低) 八位传送到 IO 接口电路。
4. 设  $DS=2000H$ ,  $AL=50H$ , 8086CPU 在执行指令 `MOV AL, [11H]` 时, 引脚  $M/\overline{IO}=$ \_\_\_\_\_,  $\overline{BHE}=$ \_\_\_\_\_,  $\overline{RD}=$ \_\_\_\_\_,  $\overline{WR}=$ \_\_\_\_\_,  $AD_0$  在  $T_1$  时钟周期为\_\_\_\_\_电平、在  $T_2$  时钟周期为\_\_\_\_\_, \_\_\_\_\_片地址锁存器 8282 输出的地址总线上信号为\_\_\_\_\_H, 存储器中的数据通过数据总线的\_\_\_\_\_ (高、低) 八位传送到 AL。

## 第六节 指令格式、寻址方式、数据传送类指令（一）

1. 写出下列指令中源操作数的寻址方式。
  - ① MOV AX, 1234H
  - ② MOV SS, AX
  - ③ MOV BX, [BP+100H]
  - ④ MOV AX, [BX+DI]
  - ⑤ MOV CX, [2000H]
2. 判断指令是否有错，如有错则改正。
  - ① MOV AX, CS
  - ② MOV [BX], X ; X 为变量名
  - ③ MOV DS, 2000H
  - ④ MOV AL, ABH
  - ⑤ POP AL
  - ⑥ MOV AL, [SI+DI]
3. 设当前 AX=1234H, BX=5678H, SS= 1000H, SP = 0100H。此时先后执行 PUSH AX 和 POP BX 指令，画出执行指令前后堆栈的变化情况（图中需要注明 SP 指针和内存单元的内容）。

## 第七节 数据传送指令（二）、算术运算指令（一）

1. 内存中偏移地址为 10H 处存储字变量 X=1234H, 试问下列指令执行后目标操作数的内容。

(1) MOV AX, X

(2) LEA AX, X

2. 下列指令序列执行后, AL 及 FR 的值。

MOV AL, 3AH;

ADD AL, 52H;

ADD AL, 74H;

3. 编写指令序列实现十进制运算: 34+19

## 第八节 算术运算指令（二）、逻辑运算指令

1. 已知  $AX=1002H$ ,  $BX=3F0H$ ,  $DS:[3F0H]=2000H$ , 求下列每条指令执行后的结果是什么?  $CF=?$  每条指令执行占多少个总线周期?
  - (1) `SUB BX, [3F0H]`
  - (2) `DEC BYTE PTR[BX]`
  - (3) `NEG AL`
  - (4) `MUL BL`
2. 按如下要求各写出一条指令。
  - (1) 测试  $AL$  寄存器的  $D1$  位是否为 1: \_\_\_\_\_。
  - (2) 将  $BL$  寄存器的值（有符号数）乘以 2: \_\_\_\_\_。
  - (3) 将  $AL$  寄存器的高 4 位清零、低 4 位保持不变: \_\_\_\_\_。
  - (4) 将  $AL$  寄存器的高 4 位取反、低 4 位保持不变: \_\_\_\_\_。
3. 设  $DS=2000H$ ,  $AX=-1000H$ ,  $BX=301H$ , 内存中从  $20300H$  地址开始处依次存储  $10H$ 、 $20H$ 、 $30H$ 、 $40H$ ……则执行指令 `SUB AX, [BX]` 需要占用\_\_\_\_\_个总线周期, 指令执行后  $AX=$ \_\_\_\_\_,  $CF=$ \_\_\_\_\_,  $OF=$ \_\_\_\_\_。
4. 思考: 用移位指令实现  $X*10$ 。

## 第九节 移位指令、(无)条件转移指令

1. 在代码段中偏移地址为 2030H 的单元中存有一条两字节无条件短转移指令代码,若第 2 字节代码为 0D8H,则转移的目标地址偏移量为\_\_\_\_\_。

2. 设有如下指令序列:

```
1234H: 190H      NEXT1: .....  
                                MOV  BL, 0FFH  
                                XOR   AL, AL  
                                CMP   AL, BL  
1234H: 200H      JL     NEXT1  
1234H: 202H      NEXT2: .....
```

当指令 `CMP AL, BL` 执行完时, AL 寄存器的值为\_\_\_\_\_, 标志寄存器 FR 的 OF 位为\_\_\_\_\_, SF 位为\_\_\_\_\_, ZF 位为\_\_\_\_\_, CF 位为\_\_\_\_\_。

3. 上题的指令序列中, 指令 `JL NEXT1` 的代码为\_\_\_\_\_字节代码, 指令代码的第 2 字节为\_\_\_\_\_, `NEXT1` 的逻辑地址是\_\_\_\_\_, 当 `JL NEXT1` 指令执行完后, 下一条应执行的指令的物理地址是\_\_\_\_\_。

## 第十节 LOOP、CALL、串操作指令

1. 设有程序段如下：

```
NEXT:  MOV  AL, [BX]
      .....
      LOOP NEXT
```

汇编时指出：转移距离超过范围（-128~127 字节）。试适当修改程序，使 LOOP 指令能转去执行 NEXT 地址的程序。

2. 设当前 SS= 0C00H，SP = 0200H，说明 CALL DELAY 指令对寄存器和存储器的影响。（注：CALL DELAY 为 3 字节指令。）

```
0C20: 0003          CALL  DELAY

                        DELAY  PROC
0C20: 000B          MOV    CX,100
0C20: 000E          L1: LOOP  L1
0C20: 0010          RET
                        DELAY  ENDP
```

3. 将 10200H 地址开始的 1000 个存储单元清 0。



## 第十一节 伪指令

若定义如下数据段：

```
DATA    SEGMENT

        ORG    100H

DA1      DB    'ABC', 42H
DA2      DW    03H, 'BC', $+2
DA3      DW    DA2
AA1      EQU   $-DA1

        ORG    $+4

BB1      DB    10 DUP(2, 2 DUP(?))

DATA     ENDS
```

写出下列指令单独执行后，目的操作数的值。

```
LEA      BX, DA2
MOV      DI, OFFSET BB1
MOV      AL, TYPE DA1
MOV      AX, AA1
MOV      AL, LENGTH BB1
MOV      AL, BYTE PTR DA2
MOV      AX, DA2+2
MOV      AX, DA3
MOV      AL, SIZE BB1
```

## 第十二节 程序运行过程、DOS 功能调用

1. 编写宏定义实现回车、换行输出。
2. 编写一段程序，实现：键盘输入一个字符，判断是否为“Q”，若是则转移到标号 EXIT，退出程序，否则向下继续运行程序。
3. 对汇编语言源程序进行汇编、连接、调试采用\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_软件。

### 第十三节DOS 功能调用、BIOS 调用、程序举例

1. 设有变量定义及程序如下：

```
X      DB  100
Y      DB  0
.....
      MOV  AL, X
      MOV  AH, 1
NEXT1: INC  AL
      JZ   EXIT
      INC  AH
      JMP  NEXT1
EXIT:   MOV  Y, AH
程序执行完, AL=____, Y=____。
```

2. 设有变量定义及程序如下：

```
X      DB  1,0,1,1,0,0,1,1
.....
      MOV  SI, OFFSET X
      MOV  BL, 80H
      MOV  BH, 0
      MOV  CX, 8
NEXT2: MOV  AL, [SI]
      TEST AL, 1
      JZ   NEXT1
      ADD  BH, BL
NEXT1: INC  SI
      SHR  BL, 1
      LOOP NEXT2
程序执行完, BH=____, BL=____。
```

3. 设有变量定义及程序如下：

```
X      DB  1, 2, 3, 4
...
      MOV  CX, 3
      MOV  SI, (OFFSET X)+3
      MOV  AH, [SI]
NEXT:  DEC  SI
      MOV  AL, [SI]
      MOV  [SI+1], AL
      LOOP NEXT
      MOV  [SI], AH
程序执行完, X 的内容为_____。
```

## 第十四节 程序举例

## 第十五节 接口