

第3章 指令集



3.1 填空题

1. 若 8086/8088 CPU 各寄存器的内容为: $AX=0000H$, $BX=0127H$, $SP=FFC0H$, $BP=FFBEH$, $SS=18A2H$ 。现执行以下 3 条指令:

- ① PUSH BX
- ② MOV AX,[BP]
- ③ PUSH AX

在执行完第①条指令后, $SP=(FFBEH)$ 。在执行完指令③后, $AX=(0127H)$, $BX=(0127H)$, $SP=(FFBCH)$ 。

解: 每执行一次 PUSH 指令, 堆栈指针 SP-2。当 BP 作为简址寄存器时, 默认指向堆栈段。

2. 从中断服务子程序返回时应使用(IRET)指令。

3. 数据段中 28A0H 单元的符号地址为 VAR, 若该单元中内容为 8C00H, 则执行指令“LEA AX,VAR”后, AX 的内容为(28A0H)。

解: 28A0H 单元的符号地址为 VAR, 即表示该单元的偏移地址是 28A0H。

4. 下列程序执行后, BX 中的内容为(C02DH)。

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

解: ROL 是不带 CF 的循环左移指令, RORL 是不带 CF 的循环右移指令。改程序的功能就是将 $BX=00B7H$ 循环右移 2 位。

5. 由 ARM 指令基本格式可以看出, ARM 指令通常会有(3 个)操作数。

解: 详见主教材 3.5.1 节。

3.2 简答题

1. 设 $DS=6000H$, $ES=2000H$, $SS=1500H$, $SI=00A0H$, $BX=0800H$, $BP=1200H$ 。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 AX,4[BX][SI]
- ③ MOV AL,'B'
- ④ MOV DI,ES:[BX]
- ⑤ MOV DX,[BP]

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 基址—变址—相对寻址。

操作数的物理地址=DS×16+SI+BX+4=60000H+00A0H+0800H+4=608A4H

- ③ 立即寻址。
- ④ 寄存器间接寻址。

操作数的物理地址=ES×16+BX=20000H+0800H=20800H

- ⑤ 寄存器间接寻址。

操作数的物理地址=SS×16+BP=15000H+1200H=16200H

2. 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. 设 DS=202AH,CS=6200H,IP=1000H,BX=1200H,位移量 DATA=2,内存数据段 BX 指向的各单元内容如图 3-1 所示。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,又有段内直接转移和间接转移,以及段间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。

转移目标的偏移地址=BX=1200H,段地址=CS=6200H

- ② 段内间接转移。

转移目标的偏移地址为 BX 所指字单元的内容=2211H,段地址=CS=6200H

- ③ 段间间接转移。

转移目标为 BX+2 所指 4 个字节单元的内容,即偏移地址=4433H,段地址=6655H

4. 比较无条件转移指令、条件转移指令、调用指令和中断指令的异同。

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在当前逻辑段,也可以在不同的逻辑段;条件

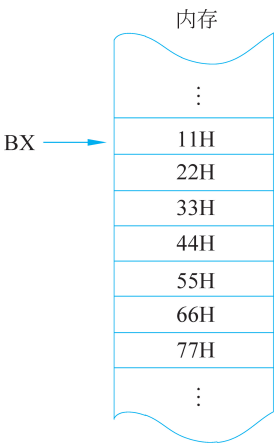


图 3-1 内存数据段

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,只能当前在逻辑段的 $-128\sim+127$ 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。

5. 说明以下程序段的功能:

```
STD
LEA DI, [1200H]
MOV CX, 0F00H
XOR AX, AX
REP STOSW
```

解: 按减地址方向,将附加段中偏移地址为 1200H 单元开始的 F00H 个字单元清 0。

6. x86 处理器主要新增了哪些类型指令?

解: x86 的 32 位指令集在 16 位指令集基础上,除增强了部分 8086 指令的功能外,还新增了包括串输入输出、字节交换、条件传送等指令(详见主教材表 3-8)。

x86-64 指令集是 x86-32 指令集的超集,它将操作数扩展为 64 位,并陆续增加了多媒体扩展指令集 MMX、SSE、SSE2 等扩展指令集,增强了处理器的多媒体、图形图像和 Internet 等的处理能力。

7. 比较 Intel 指令集与 ARM 指令集指令在格式上的主要区别。

解: 相比于 Intel x86 指令集,ARM 指令集在格式上允许 2 个或 3 个操作数,并增加了条件码和后缀两个可选项。

3.3 编程题

1. 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 11 位、7 位和 3 位。
- (4) 测试 DX 中的 0 位和 8 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
(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,此时 ZF=1,若 ZF=1
                    转向 ONE
    :

```

2. 编写程序,实现将+46 和-38 分别乘以 2。

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。可以用移位指令或乘、除运算指令实现。为方便理解,这里先将十进制数 46 和-38 转换为十六进制数,并对-38 取补码。

46=2EH, -38=-26H, $[-26H]_{补} = 11011010B = DAH$

注: 第 4 章学习完变量定义后,可以直接定义变量为十进制数,并可以直接定义为负数,如-38。数制转换和求补码的工作由汇编程序(汇编语言的编译程序)自动完成。

程序代码如下:

```

MOV AL, 2EH
SHL AL, 1          ;AL×2→AL
MOV BL, 0DAH
SAR BL, 1          ;BL×2→BL。用算术移位指令实现符号数移位

```

3. 编写程序,统计 BUFFER 为首地址的连续 200 个字节单元中 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

```

4. 写出完成下述功能的程序段。

(1) 从地址 DS: 1200H 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: 1201H 中的内容相乘。

(4) 乘积存入字单元 DS: 1202H 中。

解: (1) MOV DS: BYTE PTR[0012H], 56H

```
MOV AL,[0012H]
```

```
(2) MOV CL,2
```

```
SHL AL,CL
```

```
(3) MUL DS: BYTE PTR[1201H]
```

```
(4) MOV DS:[1202H],AX
```

5. 设内存数据段中以 M1 为首地址的字节单元中存放了 3 个无符号字节数,编写程序,求这 3 个数之和及这 3 个数的乘积,并将结果分别存放在 M2 和 M3 单元中。

解:

3 个数求和:

```
LEA SI,M1
MOV AL,[SI]
CLC
MOV CX,2
L1: INC SI
    ADC AL,[SI]
    LOOP
    MOV M2,AL
```

3 个数求乘积:

```
LEA SI,M1
XOR BX,BX
MOV AL,[SI]
MUL BYTE PTR[SI+1]
MOV BL,[SI+2]
MUL BX
MOV M3,AX
MOV M3+2,DX
```

6. 编写程序,利用串操作指令,实现按减地址方向将数据段 1000H~1010H 中的内容传送到附加段从 2000H 开始的区域中。

解:

参考代码如下:

```
MOV SI,1010H
MOV DI,2010H
STD
MOV CX,10H
REP MOVSB
HLT
```

第 4 章 汇编语言程序设计

4.1 填空题

1. 将汇编语言源程序转换为机器代码的过程称为(汇编),而要使其能够在计算机上运行,还需要通过(链接)生成可执行文件。

解: 所有源程序都需要经过编译和链接才能被 CPU 执行。汇编语言源程序的编译称为汇编。

2. 执行下列指令后,AX 寄存器中的内容是(1E00H)。

```
TABLE DW 10,20,30,40,50
ENTRY DW 3
:
MOV BX,OFFSET TABLE
ADD BX,ENTRY
MOV AX,[BX]
```

解: OFFSET 是运算符,表示取变量的偏移地址。

3. 已知:

```
ALPHA EQU 100
BETA EQU 25
```

则表达式 $ALPHA \times 100 + BETA$ 的值为(10025)。

解: EQU 是符号定义伪指令,含义是用其前边的符号名取代其后表达式的值。

4. 执行如下指令后,AX=(0004)H,BX=(0152)H。

```
DSEG SEGMENT
ORG 100H
ARY DW 3,4,5,6
CNT EQU 33
DB 1,2,CNT+5,3
DSEG ENDS
:
MOV AX,ARY+2
MOV BX,ARY+10
```