第3章 80X86的指令系统和寻址方式

CHAPTER

本章主要介绍80X86的数据类型、寻址方式、指令格式、指令系统(数据传送指令、算术运算指令、逻辑运算指令、串操作数指令、控制转移指令、处理机指令和高级语言指令)和80486指令等。

本章学习要求如下：

(1) 掌握80X86的数据类型。

(2) 掌握80X86的寻址方式。

(3) 掌握80X86的指令系统及它们的一些使用实例。

(4) 了解保护属性检查指令、高级语言指令以及cache管理指令等。

习题精解

1. 已知DS=2000H,BX=0100H,SI=0002H,存储单元[20100H]~[20103H]依次存放12 34 56 78H,[21200H]~[21203H]依次存放2A 4C B7 65H,说明下列每条指令执行后AX寄存器的内容。

(1) MOV AX,1200H

(2) MOV AX,BX

(3) MOV AX,[1200H]

(4) MOV AX,[BX]

(5) MOV AX,[BX+1100H]

(6) MOV AX,[BX+SI]

(7) MOV AX,[BX][SI+1100H]

参考答案：

根据物理地址形成公式

物理地址=段基址×16+有效地址

得

(1) 1200H

(2) 0100H

(3) 4C2AH

(4) 3412H

(5) 4C2AH

(6) 7856H

(7) 65B7H

分析：本题主要考查点是数据寻址方式，寻址方式就是寻找指令中用于说明操作数所在地址的方法，或者是寻找操作数有效地址的方法。包括数据寻址和转移地址寻址两大类。80X86 的数据寻址方式总结如下：

(1) 立即寻址

操作数作为指令的一部分而直接写在指令中，这种操作数称为立即数，这种寻址方式就称为立即数寻址。

注意：使用立即寻址的指令主要用来给寄存器赋初值，并且只能用于源操作数字段；不能直接给段寄存器和标志寄存器赋予立即数。

例如：

```
MOV  AL,5
MOV  AX,1234H
MOV  EAX,12345678H
```

(2) 寄存器寻址

操作数存放在指令规定的某个寄存器(如：对于 16 位操作数，寄存器可以是 AX、BX、CX、DX、SI、DI、SP 或 BP；而对于 8 位操作数，寄存器可以是 AH、AL、BH、BL、CH、CL、DH 或 DL)中。例如：

```
MOV  AL,BH
MOV  AX,CX
MOV  EAX,EBP
```

(3) 存储器寻址

操作数存放在存储器中，在寻址时要计算存储单元的有效地址，有效地址 EA 可以由以下 4 种成分组成。

- 位移量：是存放在指令中的一个 8 位、16 位或 32 位的数。
- 基址：其值存放在基址寄存器中。BX、BP、任何 32 位通用寄存器都可以作为基址寄存器使用。
- 变址：其值存放在变址寄存器中。SI、DI、除 ESP 外的 32 位通用寄存器都可以作为基址寄存器使用。
- 比例因子：是 80386 及后继机型中新增寻址方式中的术语。其值可为 1、2、4、8。

根据计算有效地址 EA 方法的不同，存储器寻址又分为下列几种。

① 直接寻址(direct addressing)　操作数的有效地址是指令的一部分，它与操作码一起存放在代码段中，默认操作数在数据段(DS)中，如果操作数定义在其他段中，则应在指令中指定段超越前缀。

EA=立即数

例如：

```
MOV  AX,[200H]
```

② 寄存器间接寻址(register indirect addressing)　操作数的有效地址 EA 存放在基址寄存器(BX 或 BP)或变址寄存器(DI 或 SI)中。计算物理地址的默认段规定，当寄存器是 SI、DI 和 BX 时为 DS，当寄存器是 BP 时为 SS。

EA=寄存器中的值

例如：

```
MOV  AX,[BX]
MOV  AX,[BP]
MOV  AX,[SI]
MOV  AX,[DI]
```

③ 寄存器相对寻址(register relative addressing)　操作数的有效地址 EA 由指定的寄存器内容，加上指令中给出的 8 位或 16 位偏移量(当然要由一个段寄存器作为地址基准)作为操作数的偏移地址。计算物理地址的默认段仍然是 SI、DI 和 BX 为 DS，BP 为 SS。

EA=基址或变址寄存器(BX、BP、DI、SI)±8 位或 16 位的位移量

注意：寄存器相对寻址常用于存取表格或一维数组中的元素——把表格的起始地址作为位移量，元素的下标值放在间址寄存器中(或反之)。

例如：

```
MOV  AX,[BX+2]
MOV  AX,[BP+1]
MOV  AX,[SI-1]
MOV  AX,[DI-2]
MOV  AX,[BX+2000H]
MOV  AX,[BP+1000H]
MOV  AX,[SI-1000H]
MOV  AX,[DI-2000H]
```

④ 基址变址寻址(base-plus-index addressing)　有效地址由基址寄存器(BP 或 BX)的内容加上变址寄存器(DI 或 SI)的内容形成。如基址寄存器为 BX 时，与 DS 形成的物理地址指向数据段；如基址寄存器为 BP 时，与 SS 形成的物理地址指向堆栈段。

EA=基址寄存器的内容+变址寄存器的内容

例如：

```
MOV  AX,[BX+SI]
MOV  AX,[BX+DI]
```

```
MOV  AX,[BP+SI]
MOV  AX,[BP+DI]
```

注意：一条指令中同时使用基址寄存器或变址寄存器是错误的。

例如：MOV　CL，[BX＋BP] 或 MOV AX，[SI＋DI] 均为非法指令。

⑤ 基址变址相对寻址(base-plus-index relative addressing)　有效地址由基址寄存器(BP 或 BX)、变址寄存器(DI 或 SI)及相对偏移量形成。默认段的使用仍然是 DS 与 BX 组合，以及 SS 与 BP 组合。

```
EA=基址寄存器的内容+变址寄存器的内容±8位或16位的位移量
```

例如：

```
MOV  AX,[BX+SI+2]
MOV  AX,[BX+DI+1]
MOV  AX,[BP+SI-1]
MOV  AX,[BP+DI-2]
MOV  AX,[BX+SI+2000H]
MOV  AX,[BX+DI+1000H]
MOV  AX,[BP+SI-1000H]
MOV  AX,[BP+DI-2000H]
```

⑥ 比例变址寻址方式　有效地址为变址寄存器的内容乘以指令中指定的比例因子再加上位移量之和。例如：

```
MOV  EAX,[ESI*4+100H]
```

⑦ 基址比例变址寻址方式　有效地址为变址寄存器的内容乘以指令中指定的比例因子再加上基址寄存器的内容之和。例如：

```
MOV  EAX,[ESI*4+EBX]
```

⑧ 相对基址比例变址寻址方式　有效地址为变址寄存器的内容乘以指令中指定的比例因子，加上基址寄存器的内容，再加上位移量之和。例如：

```
MOV  EAX,[ESI*4+EBX+8]
```

(4) 隐含寻址

指令操作数是隐含的，在指令中未显式地指明。例如：

```
MUL BL
```

指令隐含了被乘数 AL 及乘积 AX。

类似的指令还有 DIV、CBW、MOVS 等。

(5) I/O 端口寻址

① 直接端口寻址　端口地址用 8 位立即数(0～255)表示。例如：

```
IN AL,21H;    从 I/O端口地址为 21H 的端口中读取数据送到 AL 中
```

② 间接端口寻址　I/O 端口地址事先存放在规定的 DX 寄存器中(0～65535)。例如：

```
MOV  DX,0FF03H
OUT  DX,AL;    将 AL 中的内容输出到地址由 DX 内容所指定的端口中
```

2. 指出下列指令的错误。

(1) MOV CX,DL　　(2) MOV IP,AX

(3) MOV ES,1234H　　(4) MOV ES,DS

(5) MOV AL,300　　(6) MOV [SP],AX

(7) MOV AX,BX+DI　　(8) MOV 20H,AH

参考答案：

(1) 字长不同。

(2) IP 寄存器既不能作为源操作数，也不能作为目的操作数。

(3) 立即数不能直接赋给段寄存器。

(4) 段寄存器之间不能直接赋值。

(5) 立即数 300 超过 8 位，不能赋给 8 位寄存器 AL。

(6) SP 不能作为间址寄存器寻址。

(7) 两个寄存器不能相加。

(8) 立即数不能作为目标操作数。

分析：本题主要考查在应用通用传送指令时，需要注意以下几项(以 MOV 指令为例)：

(1) 传送指令可传送 8 位数据，也可传送 16 位数据，具体取决于指令中涉及的寄存器是 8 位还是 16 位，也取决于立即数的形式。例如：

```
MOV  AX,[35AH]          ;传送 DS 段中偏移地址为 35AH 的字单元内容至 AX
MOV  BL,[35AH]          ;传送 DS 段中偏移地址为 35AH 的字节单元内容至 BL
MOV  [BP],WORD PTR 18H  ;传送 16 位数据 0018H 至 SS 段中两个单元
MOV  [BP],BYTE PTR 37H  ;传送 8 位数据 37H 至 SS 段中一个单元
```

(2) 传送指令中总是既含源操作数，又含目的操作数，两者之中至少有一个是用寄存器来指出的，这可减少指令长度。因此，一个立即数不能直接送直接寻址的内存单元。交换指令(XCHG)两个操作数都不能是立即数。例如：

```
MOV  WORD PTR [1000H],32A8H    ;错误
MOV  WORD PTR [BX],32A8H       ;正确
```

(3) 传送指令不能在两个内存单元之间直接传送数据。例如：

```
MOV  [2000H],[35AH]          ;错误
MOV  WORD PTR [BX],[8729H]   ;错误
```

(4) 在传送指令中，寄存器既可以作为源操作数，也可以作为目的操作数，但 CS 寄存器不能作为目的操作数，换句话说，这个寄存器的值不能随意修改。而 IP 寄存器既不能

作源操作数，也不能作为目的操作数。例如：

```
MOV  CS,[35AH]          ;错误
MOV  CS,AX              ;错误
MOV  AX,IP              ;错误
MOV  IP,WORD PTR[BX]    ;错误
```

(5) 用 BX、SI、DI 来间接寻址时，默认的段寄存器为 DS，而用 BP 来间接寻址时，默认的段寄存器为 SS。

例如：对于 MOV WORD PTR[BP],1000 和 MOV WORD PTR[BX],2000，设 DS=3000H，SS=4000H，BX=5000H，BP=6000H，则前一条指令将立即数 1000 送到物理地址为 46000H 和 46001H 的两单元中，后一条指令将立即数 2000 送到物理地址为 35000H 和 35001H 的两单元中。

(6) 8086 系统规定，凡是遇到给 SS 寄存器赋值的传送指令时，系统会自动禁止外部中断，等到本条指令和下条指令执行后，又自动恢复对 SS 寄存器赋值前的中断开放状态。这样做是为了允许程序员连续用两条指令分别对 SS 和 SP 寄存器赋值，同时又防止堆栈空间变动过程中出现中断。了解这点后，就应该注意在修改 SS 和 SP 的指令之间不要插入其他指令。例如：

```
MOV  SS,AX
MOV  DL,38H
MOV  AH,2
INT  21H
MOV  SP,BX
```

上述写法是错误的。

(7) 除了一些直接影响 FLAGS 的指令(如 POPF)外，一般传送指令不改变标志寄存器的内容。

(8) 立即数不能直接送段寄存器 DS、ES 以及 SS，要用通用寄存器或存储单元作桥梁。例如：

```
MOV  DS,875BH             ;错误
MOV  AX,875BH
MOV  DS,AX                ;正确
MOV  WORD PTR[BX],32A8H
MOV  ES,[BX]              ;正确
```

实际上，上述几点中有些要求适于其他通用传送型指令，甚至也适合其他种类的指令，读者可在后面的学习中不断去总结和积累。

3. 已知数字 0～9 对应的格雷码依次为：18H、34H、05H、06H、09H、0AH、0CH、11H、12H、14H，它存在于以 TABLE 为首地址(设为 200H)的连续区域中。对如下程序段的每条指令加上注释，说明每条指令的功能和执行结果。

```
LEA      BX,TABLE
MOV      AL,8
```

```
XLAT
```

参考答案：

```
LET    BX,TABLE     ;得到表首地址,放在寄存器 BX 中
MOV    AL,8         ;立即数 8 赋值给 AL,即 AL 中存放的是相对于表首的偏移地址
XLAT                ;利用查表转换指令,实质是查找 8 的格雷码
```

结果是(AL)＝12H。

程序段的功能为：把表首地址即 200H 赋给 BX。

分析：本题主要考查查表转换指令(XLAT)的应用。

指令功能：AL←DS：[BX＋AL]。

用途：用于查表(或对一维数组操作)，表首地址的偏移地址在 BX 中，表长度可达 256 字节。把 BX 的值作为内存字节数组首地址、下标为 AL 的数组元素的值传送给 AL。有两个隐含操作数 BX 和 AL。

4. 什么是堆栈？它的工作原则是什么？它的基本操作有哪两个？对应哪两种指令？

参考答案：

堆栈是一段具有特殊存取规则的数据区，工作原则是先进后出(FILO)，它有两种基本的操作，即进栈和出栈，对应的指令为 PUSH 和 POP。

分析：堆栈的应用很多，应用 8086/8088 系统堆栈时要注意下面几点：

- 堆栈是向下生长的；
- 工作原则是先进后出(FILO)；
- 有两种操作，即进栈和出栈，只能进行字操作，且是对准字；
- 对应的逻辑地址是 SS：SP。

5. 略。

6. 给出下列各条指令执行后 AL 的值，以及 CF、ZF、SF、OF 和 PF 的状态。

```
MOV  AL,89H
ADD  AL,AL
ADD  AL,9DH
CMP  AL,0BCH
SUB  AL,AL
DEC  AL
INC  AL
```

参考答案：

```
MOV  AL, 89H    ;(AL)=89H 各状态标志位不变
ADD  AL, AL     ;(AL)=12H CF=1 ZF=0 SF=0 OF=1 PF=1
ADD  AL, 9DH    ;(AL)=AFH CF=0 ZF=0 SF=1 OF=0 PF=1
CMP  AL, 0BCH   ;(AL)=AFH CF=1 ZF=0 SF=0 OF=0 PF=0
SUB  AL, AL     ;(AL)=0H CF=0 ZF=1 SF=0 OF=0 PF=1
DEC  AL         ;(AL)=FFH CF=0(不影响 CF) ZF=0 SF=0 OF=1 PF=1
INC  AL         ;(AL)=0H CF=0(不影响 CF) ZF=1 SF=0 OF=1 PF=1
```

分析：本题主要考查指令对标志位的影响，总结如下。

(1) 数据传送指令除了几个专门给标志寄存器赋值的指令(如：POPF、SAHF 等)外，对标志位都没有影响。

(2) 算术运算指令中除了 DEC 和 INC 不影响 CF 外，其他算术指令对 6 个状态标志位都有影响。

(3) 逻辑运算指令中，NOT 不影响标志位，其他 4 种指令将使 CF＝OF＝0，AF 无定义，而 SF、ZF 和 PF 则根据运算结果而定。

以上详细情况请参考主教材。

7. 略。

8. 略。

9. 略。

10. 略。

11. 略。

12. 编写程序段完成如下要求：

(1) 用位操作指令实现 AL(无符号数)乘以 10。

(2) 用逻辑运算指令实现数字 0～9 的 ASCII 码与非压缩 BCD 码的互相转换。

(3) 把 DX. AX 中的双字右移 4 位。

参考答案：

(1)

```
MOV  BL, AL
MOV  CL, 3
SHL  AL, CL  ;AL×8
SHL  BL, 1   ;AL×2
ADD  AL, BL  ;AL×8+AL×2=AL×10
```

分析：本题主要考查用移位指令实现乘除法。

① 算术移位(SAL/SAR)——把操作数看作有符号数。

逻辑移位(SHL/SHR)——把操作数看作无符号数。

② 移位位数放在 CL 寄存器中，如果只移 1 位，也可以直接写在指令中。例如：

```
MOV  CL,4
SHR  AL,CL    ;AL 中的内容右移 4 位
```

③ 影响 C、P、S、Z、O 标志。

④ 结果未溢出时，有

左移 1 位＝操作数×2

右移 1 位＝操作数/2

⑤ 用移位操作代替乘除法可提高运算速度。在编写汇编程序的时候，应该注意程序的执行效率，使程序得到最大程度的优化，特别是在处理海量数据的时候，这就变得非常必要。移位指令作为系统指令的一部分，可以在一定程度上帮助实现复杂的数值运算，而

不会增加系统负担，这是非常有意义的。

例如：计算 x * 10。

采用乘法指令：

```
MOV  BL,10
MUL  BL
```

共需 70～77 个 T 周期。

采用移位和加法指令：

```
SAL  AL,1      ;2T
MOV  AH,AL     ;2T
SAL  AL,1      ;2T
SAL  AL,1      ;2T
ADD  AL,AH     ;3T
```

只需 11 个 T 周期，仅相当于乘法指令的 1/7。

(2) ASCII 码转换成非压缩型 BCD 码的指令是：

```
AND  AL, 0FH
```

非压缩型 BCD 转换成 ASCII 码的指令是：

```
OR  AL, 30H
```

分析：根据 ASCII 码与 BCD 码的特点：

① 0～9 的 ASCII 码为 30H～39H。

② 0～9 非压缩型 BCD 码为 0H～9H。

又根据或(OR)指令和与(AND)指令的特点：

① 任何数和 1 相或(OR)结果都得 1。

② 任何数和 0 相与(AND)结果都得 0。

(3)

```
MOV  BX, DX
AND  BX, 0FH
MOV  CL, 12
SHL  BX, CL
MOV  CL, 4
SHR  DX, CL
SHR  AX, CL
OR   AX, BX
```

分析：实现 32 位数据逻辑右移，要注意高字的低位移到低字的高位。

13. 略。

14. 已知数据段 500H～600H 处存放了一个字符串，说明下列程序段执行后的结果。

```
MOV  SI,600H
MOV  DI,601H
```

```
MOV  AX,DS
MOV  ES,AX
MOV  CX,256
STD
REP  MOVSB
```

参考答案：

把500H～600H处的256个字节的字符串顺序存放到偏移地址601H的地方。

分析：本题和题15都在考查串操作指令的应用。

串操作类指令可以用来实现内存区域的数据串操作。这些数据串可以是字节串，也可以是字串。

(1) 重复指令前缀

串操作类指令可以与重复指令前缀配合使用，从而可以使操作得以重复进行，及时停止。重复指令前缀的几种形式见表3-1所示。

表3-1 重复前缀

汇编格式	执行过程	影响指令
REP	(1)若(CX)＝0，则退出；(2)CX＝CX－1；(3)执行后续指令；(4)重复(1)～(3)	MOVS,STOS,LODS
REPE/REPZ	(1)若(CX)＝0或ZF＝0，则退出；(2)CX＝CX－1；(3)执行后续指令；(4)重复(1)～(3)	CMPS,SCAS
REPNE/REPNZ	(1)若(CX)＝0或ZF＝1，则退出；(2)CX＝CX－1；(3)执行后续指令；(4)重复(1)～(3)	CMPS,SCAS

(2) 串操作指令

串操作指令共有5种，具体见表3-2。对串指令要注意以下几个问题：

表3-2 串操作指令

功能	指令格式	执行操作
串传送	MOVS DST,SRC MOVSB MOVSW	由操作数说明是字节或字操作；其余同MOVSB或MOVSW [(ES：DI)]←[(DS：SI)]；SI＝SI±1，DI＝DI±1； [(ES：DI)]←[(DS：SI)]；SI＝SI±2，DI＝DI±2；
串比较	CMPS DST,SRC CMPSB CMPSW	由操作数说明是字节或字操作；其余同CMPSB或CMPSW [(ES：DI)]－[(DS：SI)]；SI＝SI±1，DI＝DI±1； [(ES：DI)]－[(DS：SI)]；SI＝SI±2，DI＝DI±2；
串搜索	SCAS DST SCASB SCASW	由操作数说明是字节或字操作；其余同SCASB或SCASW AL－[(ES：DI)]；DI＝DI±1； AX－[(ES：DI)]；DI＝DI±2；
存串	STOS DST STOSB STOSW	由操作数说明是字节或字操作；其余同STOSB或STOSW AL→[(ES：DI)]；DI＝DI±1； AX→[(ES：DI)]；DI＝DI±2；
取串	LODS SRC LODSB LODSW	由操作数说明是字节或字操作；其余同LODSB或LODSW [(DS：SI)]→AL；SI＝SI±1； [(DS：SI)]→AX；SI＝SI±2；