

全面、准确地理解和掌握每条汇编指令的功能和用法,是利用指令编写汇编语言程序的关键。在学习每条指令的时候,应该关注指令的功能、寻址方式、是否影响标志位等。本章主要介绍 8086 CPU 指令系统及常用的寻址方式,重点是与数据有关的寻址方式,并结合 DEBUG 上机验证。

3.1 指令和指令系统

3.1.1 汇编指令

1. 指令

通常一条汇编指令对应着一条机器指令,它是计算机能够直接识别并执行的二进制代码,代表一种基本操作,例如:加法、减法、乘法、除法等。计算机中的指令由操作码和操作数两部分组成。操作码说明计算机要执行的操作,而操作数是在指令执行过程中所需要的操作对象。操作数部分可以是操作数本身,也可以是操作数地址或是与操作数有关的其他信息。计算机所能执行的全部指令构成了计算机的指令系统。每种计算机都有各自的指令系统,80x86 系列的指令系统是向上兼容的。

2. 代码指令的格式

当一条汇编指令译成代码指令时,由汇编指令中的操作码和操作数共同决定代码指令的格式。对 8086 CPU,代码指令占 1~6B 不等。第一个字节固定为操作码(OPCODE),其余字节为操作数,操作数可能有 0 个、1 个、2 个或 3 个,操作数字节的确定与指令的寻址方式有关。代码指令的一般格式如图 3-1 所示。

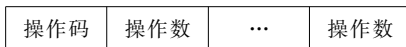


图 3-1 代码指令的一般格式

在汇编语言中,用助记符表示操作码,用符号或符号地址表示操作数或操作数所在的地址,与机器指令一一对应。8086 CPU 的指令系统中,主要包括数据传送指令、算术指令、逻辑指令、串操作指令、控制转移指令、处理机控制与杂项操作指令。为了使学习能够循序渐进地学习汇编指令,本书把指令系统分散到后续各章介绍,使指令系统的学习与汇编语言程序设计有效地结合,避免过多地学习指令系统,令学生感到厌倦。

学习指令系统时,应重点关注指令的汇编格式、指令的基本功能、指令支持的寻址方式、指令的执行对标志位的影响、指令的特殊要求等,切忌死记硬背。因此,本节不再详细介绍

具体指令的使用。

3.1.2 汇编指令的书写形式

1. 格式

```
[name] operation operand [;comment]
```

名字项(name)：可以是标号或变量，表示本语句的符号地址。标号在代码段中定义，后跟冒号。变量在数据段或附加段中定义，后面不跟冒号。

名字项组成：以字母打头的字符串组成，包含字符(A~Z, 0~9, _, ?, \$, @等)，如MainLoop, Calc_long_sum。单独的\$或“?”有特殊含义，不能做符号名。保留字不能用在名字项。

操作项(operation)：可以是指令、伪指令、宏指令的助记符。

操作数项(operand)：可以是一个或多个表达式组成，多个操作数之间用逗号分隔。操作数可能是一个或多个，也可能没有。表达式中涉及的操作符如下：

- 算术操作符：+，-，*，/，MOD；
- 逻辑操作符：AND, OR, XOR, NOT, SHL, SHR；
- 关系操作符：EQ, NE, LT, GT, LE, GE；
- 其他操作符：TYPE, LENGTH, SIZE, OFFSET, SEG, PTR, THIS等。

注释项(comment)：可选项，用来说明程序或指令的功能。一般由分号“;”开始直至回车键结束均视为注释部分。

2. 操作数的主要类型

立即操作数：也称立即数，可采用二、八、十、十六进制，后缀分别为B、Q或O、D、H，若十六进制以字母开头，必须加前导“0”。立即数是作为指令代码的一部分出现在指令中，它通常作为源操作数使用，给寄存器、地址赋初值、循环数等。

寄存器操作数：寄存器的内容参加运算或存放结果。寄存器操作数是把操作数存放在CPU中的寄存器内，即用寄存器存放源或目的操作数。在汇编指令中给出寄存器的名称。在双操作数指令中，可以做源也可以做目的操作数。

存储器操作数：指内存某地址的字节、字、双字等是指令的处理对象，这时必须把处理对象取出或送入相应地址。存储器操作数所在的存储器地址应该是物理地址，但在汇编指令中，通常只给出有效地址EA(它是以各种寻址方式给出的)，而段基值(在段寄存器中)是通过隐含方式(或段超越)使用的。

3. 有效地址和段超越

存储单元的物理地址由两部分组成：段寄存器保存的段基值和偏移地址。在8086的寻址方式中，存储单元所需的偏移地址，称为有效地址，用EA表示。不同的寻址方式，组成有效地址EA的各部分内容也不一样，寻址方式主要是EA如何计算与寻找的问题。

一般情况下，使用寻址方式中规定的默认段寄存器来确定段的基地址。指令中的操作数若不在原默认的区段内，必须在指令中指定段寄存器，这称为段超越。一般操作数可以存放在代码段、数据段、堆栈段或附加段中，而指令必须在代码段中，堆栈也只能在堆栈段中。段地址的基本规定以及允许段超越的情况如表3-1所示。

表 3-1 段地址的基本约定以及允许的段超越情况

访问存储器的方式	默 认 段	允许段超越	偏 移 地 址
取指令	CS	无	IP
堆栈操作	SS	无	SP
一般数据访问	DS	CS,ES,SS	有效地址 EA
BP 作为基址的寻址	SS	CS,ES,DS	BP
串操作的源操作数	DS	CS,ES,SS	SI
串操作的目的操作数	ES	无	DI

3.2 寻址方式

寻址方式是汇编语言中非常重要的内容,只有掌握好各种寻址方式的使用,才能真正理解汇编语言的编程方法。寻址方式有两大类,一类是与数据有关的寻址方式,另一类是与转移地址有关的寻址方式。这里介绍的基本的寻址方式都是与数据有关的,简单地说寻址方式就是寻找操作数的方法。操作数作为操作对象,有时直接给出,也有时不直接给出,而是给出操作数的有效地址或者给出得到操作数有效地址的计算方法。如何得到操作数,这就是寻址方式的主要内容。

本节重点介绍与数据有关的 8086 寻址方式,为便于理解寻址方式的实例,这里先给出 MOV 指令的一般用法。

格式: MOV 目的操作数,源操作数

功能:(目的操作数) \leftarrow (源操作数),即数据从源操作数的副本传送到目的操作数中,它类似于复制操作。

3.2.1 立即寻址方式

在立即寻址方式中,操作数在指令中直接给出,紧跟在操作码的后面存储在代码段中。立即数可以是 8 位或 16 位,对于 16 位立即数,存储时仍然遵循“低位在前,高位在后”的原则。

指令格式: MOV 目的操作数,立即数

例如:

```
MOV AL,5           ;(AL) $\leftarrow$ 5
MOV AH,0FFH       ;(AH) $\leftarrow$ FFH
MOV AX,3064H      ;(AX) $\leftarrow$ 3064H
MOV AL,'A'        ;(AL) $\leftarrow$ 字符'A'的 ASCII 码
```

说明:

- (1) 立即数存储在代码段中。
- (2) 立即数只能用于源操作数字段,如“MOV 40H, AL”错误。
- (3) “源”和“目的”的字长一致,如“MOV AH, 3064H”错误。

【例 3-1】 指令“MOV AX,3864H”执行后,AX=?

指令执行后,立即数 3864H 存入寄存器 AX,操作示意图如图 3-2 所示。



图 3-2 立即寻址操作示意图

在 DEBUG 中,先用 A 命令输入汇编指令,然后用 T 命令单步执行这条指令,指令中的源操作数 3864H 就传递到了 AX 寄存器中。执行结果如下:

```
- A
136E:0100 MOV AX, 3864
136E:0103
- T
AX = 3864  BX = 0000  CX = 0000  DX = 0000  SP = FFEE  BP = 0000  SI = 0000  DI = 0000
DS = 136E  ES = 136E  SS = 136E  CS = 136E  IP = 0103  NV UP EI PL NZ NA PO NC
136E:0103 0000          ADD     [BX + SI], AL          DS:0000 = CD
-
```

3.2.2 寄存器寻址方式

寄存器寻址方式是指操作数存放在处理器内部的寄存器中。在汇编语言中,直接利用寄存器名来表达操作数所在的寄存器。操作数可以是 8 位或 16 位寄存器,但要注意字长的匹配。

说明:

- (1) 字节寄存器只有: AH, AL, BH, BL, CH, CL, DH, DL。
- (2) 16 位寄存器: AX, BX, CX, DX, SI, DI, SP, BP。
- (3) “源”和“目的”的字长一致,如“MOV AH, BX”错误。
- (4) CS 不能用 MOV 指令改变,如“MOV CS, AX”错误。

【例 3-2】 若指令执行前, AX = 3864H, BX = 1234H, 则执行下列两条指令后, 寄存器内容如何变化?

```
MOV AX, BX           ;寄存器 BX 的内容传送给 AX, AX = 1234H
MOV AL, BH           ;寄存器 BH 的内容传送给 AL, AL = 12H
```

在 DEBUG 中验证结果如下:

```
- R AX
AX 0000
:3864
- R BX
BX 0000
:1234
- R
AX = 3864  BX = 1234  CX = 0000  DX = 0000  SP = FFEE  BP = 0000  SI = 0000  DI = 0000
DS = 136E  ES = 136E  SS = 136E  CS = 136E  IP = 0100  NV UP EI PL NZ NA PO NC
```

```

136E:0100 0000      ADD     [BX + SI], AL           DS:1234 = 00
- A
136E:0100 MOV AX, BX
136E:0102 MOV AL, BH
136E:0104
- T
AX = 1234  BX = 1234  CX = 0000  DX = 0000  SP = FFEE  BP = 0000  SI = 0000  DI = 0000
DS = 136E  ES = 136E  SS = 136E  CS = 136E  IP = 0102  NV UP EI PL NZ NA PO NC
136E:0102 88F8      MOV     AL, BH
- T
AX = 1212  BX = 1234  CX = 0000  DX = 0000  SP = FFEE  BP = 0000  SI = 0000  DI = 0000
DS = 136E  ES = 136E  SS = 136E  CS = 136E  IP = 0104  NV UP EI PL NZ NA PO NC
136E:0104 0000      ADD     [BX + SI], AL           DS:1234 = 00
-

```

【例 3-3】 说明下列指令采用的寻址方式和执行结果：

```

MOV AX, 0ABCDH
MOV BX, AX
MOV AH, AL
(AH) = ? (BX) = ?

```

解析：第一条指令，采用立即寻址，将立即数 0ABCDH 存入寄存器 AX，则 (AX) = 0ABCDH；第二条指令，采用寄存器寻址，执行后，(BX) = 0ABCDH，AX 内容不变；第三条指令，采用寄存器寻址，执行后，(AH) = 0CDH，(AL) = 0CDH。最终结果，(AX) = 0CDCDH，(BX) = 0ABCDH。

DEBUG 下验证结果：

```

- A
136E:0100 MOV AX, ABCD
136E:0103 MOV BX, AX
136E:0105 MOV AH, AL
136E:0107
- T
AX = ABCD  BX = 0000  CX = 0000  DX = 0000  SP = FFEE  BP = 0000  SI = 0000  DI = 0000
DS = 136E  ES = 136E  SS = 136E  CS = 136E  IP = 0103  NV UP EI PL NZ NA PO NC
136E:0103 89C3      MOV     BX, AX
- T
AX = ABCD  BX = ABCD  CX = 0000  DX = 0000  SP = FFEE  BP = 0000  SI = 0000  DI = 0000
DS = 136E  ES = 136E  SS = 136E  CS = 136E  IP = 0105  NV UP EI PL NZ NA PO NC
136E:0105 88C4      MOV     AH, AL
- T
AX = CDCD  BX = ABCD  CX = 0000  DX = 0000  SP = FFEE  BP = 0000  SI = 0000  DI = 0000
DS = 136E  ES = 136E  SS = 136E  CS = 136E  IP = 0107  NV UP EI PL NZ NA PO NC
136E:0107 0000      ADD     [BX + SI], AL           DS:ABCD = 00
-

```

3.2.3 直接寻址方式

在直接寻址方式的指令中直接包含了操作数的有效地址，即 EA 由指令直接给出。EA

就在指令的操作码后面的操作数字段中,且按 EA 的低字节在前,高字节在后的形式存放。直接寻址方式直接给出了操作数在主存中的偏移地址,实际的物理地址应由段基值与这个直接给出的有效地址 EA 的组合来决定。一般形式为[偏移地址],或者为符号变量,但在 DEBUG 中不能使用符号变量这种形式。

说明:

(1) 隐含的段为数据段 DS,有效地址 EA 存储在代码段中。

(2) 可使用段跨越前缀,如“MOV AX, ES: [2000H]”。

【例 3-4】 说明“MOV AX, [2000H]”指令的执行情况。

在指令中给出了 EA=2000H,假设(DS)=3000H,那么物理地址 PA=32000H。在该地址单元中的内容是操作数,如图 3-3 所示。执行结果为(AH)=3050H。

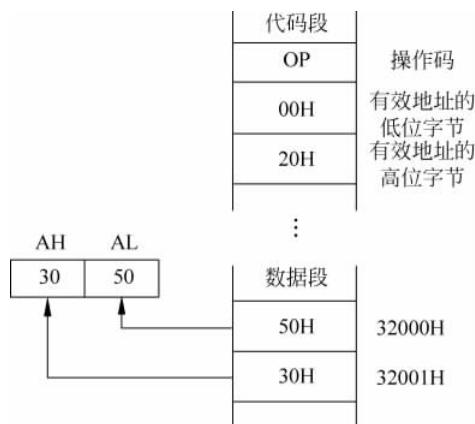


图 3-3 直接寻址示意图

在 DEBUG 中验证时,先用 R 命令设置初始值(DS)=3000H,再用 E 命令设置(3000H:2000H)单元内容为 50H,(3000H:2001H)单元内容为 30H,然后执行汇编指令 A,执行指令后,可以看到(AH)的内容为 3050H。运行结果如下:

```
- R DS
DS 136E
:3000
- E DS:2000 50 30
- D DS:2000 2001
3000:2000 50 30                                PO
- A
136E:0100 MOV AX,[2000]
136E:0103
- T
AX = 3050  BX = 0000  CX = 0000  DX = 0000  SP = FFEE  BP = 0000  SI = 0000  DI = 0000
DS = 3000  ES = 136E  SS = 136E  CS = 136E  IP = 0103  NV UP EI PL NZ NA PO NC
136E:0103 0000          ADD     [BX + SI],AL          DS:0000 = 00
-
```

3.2.4 寄存器间接寻址方式

寄存器间接寻址方式:有效地址 EA 存放在基址寄存器(BX/BP)或变址寄存器

(SI/DI)中,再根据有效地址访问到存储单元中的操作数。它与寄存器寻址不同,这种寻址方式中,寄存器的内容是操作数的有效地址。

说明:

- (1) 有效地址 EA 只能放在 SI、DI、BX、BP,不允许使用其他寄存器。
- (2) “源”和“目的”的字长要匹配,例如:

```
MOV DL, [ BX ]      ;[BX]指示一个字节单元
MOV DX, [ BX ]      ;[BX]指示一个字单元
```

- (3) 适于数组、字符串、表格的处理。

寄存器间接寻址的地址形成,可概括为表 3-2。

表 3-2 寄存器间接寻址的地址形成

变址器	书写形式	EA	默认段寄存器	物理地址
SI	[SI]	(SI)	DS	(DS) * 10H + (SI)
DI	[DI]	(DI)	DS	(DS) * 10H + (DI)
BX	[BX]	(BX)	DS	(DS) * 10H + (BX)
BP	[BP]	(BP)	SS	(SS) * 10H + (BP)

【例 3-5】 说明“MOV AX,[BX]”指令的执行情况。

若(DS)=2000H,(BX)=1000H,存储器中数据如图 3-4 所示。寄存器 BX 的内容不是操作数,而是操作数在存储器中的偏移地址,即有效地址 EA=1000H,物理地址为 DS 寄存器内容左移四位再加上 EA,即 PA=21000H。执行时,从低地址单元 21000H 中取出的内容 A0H 送寄存器 AL,从高地址单元 21001H 中取出的内容 50H 送寄存器 AH。执行结果,寄存器(AX)=50A0H。

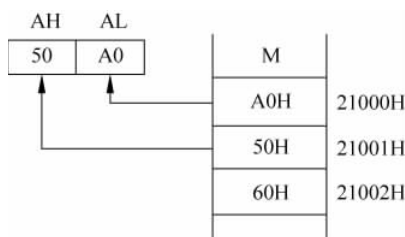


图 3-4 寄存器间接寻址示意图

在 DEBUG 中,上机验证过程如下:

```
- R DS
DS 136E
:2000
- R BX
BX 0000
:1000
- E DS:1000 A0 50 60
- D DS:1000 L3
2000:1000 A0 50 60
- A
```

. P`

```

136E:0100 MOV AX, [BX]
136E:0102
- T
AX = 50A0  BX = 1000  CX = 0000  DX = 0000  SP = FFEE  BP = 0000  SI = 0000  DI = 0000
DS = 2000  ES = 136E  SS = 136E  CS = 136E  IP = 0102  NV UP EI PL NZ NA PO NC
136E:0102 0000      ADD      [BX + SI], AL          DS:1000 = A0
-

```

3.2.5 寄存器相对寻址方式

寄存器相对寻址方式采用寄存器与位移量的组合,有效地址是寄存器内容与位移量之和。寄存器名只能为 BX、BP、SI、DI 之一,位移量为立即数或符号常量、符号变量,但 DEBUG 中不能使用后两种形式。即:

有效地址 = (BX/BP) 或 (SI/DI) + 位移 disp

寄存器相对寻址方式的操作数的书写形式,一般有以下几种:

```

[BX] + disp
disp[BX]
[BX + disp]

```

这种寻址方式适于数组、字符串、表格的处理,其地址形成概括为表 3-3。

表 3-3 寄存器相对寻址的地址形成

变址器	书写形式	EA	默认段寄存器	物理地址
SI	disp[SI]	(SI) + disp	DS	(DS) * 10H + (SI) + disp
DI	disp[DI]	(DI) + disp	DS	(DS) * 10H + (DI) + disp
BX	disp[BX]	(BX) + disp	DS	(DS) * 10H + (BX) + disp
BP	disp[BP]	(BP) + disp	SS	(SS) * 10H + (BP) + disp

【例 3-6】 说明“MOV AX,COUNT[SI]”或“MOV AX,[COUNT+SI]”指令的执行情况。

假设(DS) = 3000H, (SI) = 2000H, COUNT = 10H, 则 PA = 32010H; 若(32010H) = 3EH, (32011H) = 1AH 那么(AX) = 1A3EH。相对寻址示意图如图 3-5 所示。

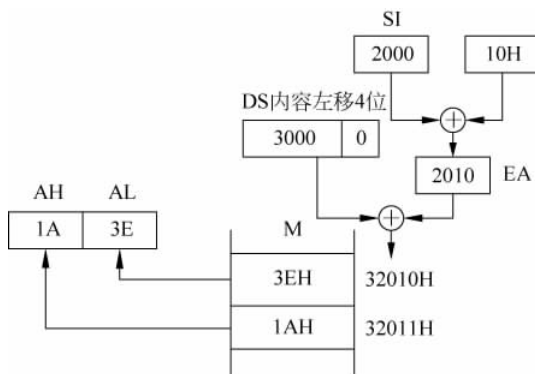


图 3-5 寄存器相对寻址示意图

在 DEBUG 中,进行该指令的验证,上机操作过程如下:

```

- R DS
DS 136E
:3000
- R SI
SI 0000
:2000
- E DS:2010 3E 1A
- D DS:2010 2011
3000:2010 3E 1A >
- A
136E:0100 MOV AX,10[SI]
136E:0103
- T
AX = 1A3E BX = 0000 CX = 0000 DX = 0000 SP = FFEE BP = 0000 SI = 2000 DI = 0000
DS = 3000 ES = 136E SS = 136E CS = 136E IP = 0103 NV UP EI PL NZ NA PO NC
136E:0103 0000 ADD [BX + SI],AL DS:2000 = 00
-

```

3.2.6 基址变址寻址方式

基址变址寻址方式是利用基址寄存器(BX/BP)与变址寄存器(SI/DI)的组合,有效地址是基址寄存器内容与变址寄存器内容之和。一般形式为[BX] [SI]、[BX] [DI]、[BP] [SI]或者[BP] [DI]。两个寄存器也可以放在一个[]内,用加号连接,如[BX + SI]等。

注意,这种寻址方式只能是一个基址寄存器和一个变址寄存器的组合,即

有效地址 = 基址寄存器(BX/BP) + 变址寄存器(SI/DI)

【例 3-7】 说明“MOV DX,[BP+DI]”或“MOV DX,[BP][DI]”指令的执行情况。

若 SS=3000H,DS=2000H,BP=2000H,DI=0500H,则有效地址为

EA = BP + DI = 2000H + 0500H = 2500H;

由于基址寄存器采用 BP,因此默认段寄存器是 SS,这样源操作数的物理地址为
 $SS \times 16 + EA = 30000H + 2500H = 32500H$ 。

若指令执行前: DX=1234H,(32500H)=78H,(32501H)=56H;

则指令执行后: DX=5678H,(32500H)=78H,(32501H)=56H,SS=3000H,DS=2000H,BP=2000H,DI=0500H。

3.2.7 相对基址变址寻址方式

相对基址变址寻址方式又称基址变址相对寻址,它使用基址寄存器、变址寄存器和相对位移量三个组成部分,有效地址是基址寄存器(BX/BP)、变址寄存器(SI/DI)与位移量之和。一般形式是基址变址寻址形式中再增加一个位移量,如[BX][SI] + 位移量的形式。

有效地址 = 基址寄存器(BX/BP) + 变址寄存器(SI/DI) + 位移 disp

书写形式有以下几种:

[BX][SI] + disp
 disp[BX][SI]
 [BX + SI] + disp
 [BX + SI + disp]

当位移量为 0 时,就成为了基址变址寻址方式。其地址的形成过程概括为表 3-4。

表 3-4 相对基址变址寻址的地址形成

基址	变址	书写形式	EA	默认段寄存器	物理地址
BX	SI	disp[BX][SI]	(BX) + (SI) + disp	DS	(DS) * 10H + EA
BX	DI	disp[BX][DI]	(BX) + (DI) + disp	DS	(DS) * 10H + EA
BP	SI	disp[BP][SI]	(BP) + (SI) + disp	SS	(SS) * 10H + EA
BP	DI	disp[BP][DI]	(BP) + (DI) + disp	SS	(SS) * 10H + EA

【例 3-8】 说明“MOV AX,DISP[BX][SI]”指令的执行情况。

假如 DS=4000H, BX=3000H, SI=2000H, DISP=0600H, 则源操作数的有效地址 EA=BX+SI+DISP=3000H+2000H+0600H=5600H;

其物理地址=DS×16+EA=40000H+5600H=45600H。

若执行前: AX=673AH, (45600H)=83H, (45601H)=6AH;

则执行后: AX=6A83H, (45600H)=83H, (45601H)=6AH。

3.2.8 寻址方式小结

表 3-5 概括了上述几种寻址方式,供复习使用。在 DEBUG 下,学习和理解寻址方式有助于深入理解汇编指令及其寻址方式的内涵,真正做到理论与实践相结合。

表 3-5 寻址方式小结

寻址方式	操作数地址	指令格式举例
立即寻址	操作数由指令给出	MOV DX, 100H
寄存器寻址	操作数在寄存器中	MOV AX, BX
直接寻址	操作数有效地址由指令直接给出	MOV AX, [2000H]
寄存器间接寻址	见表 3-2	MOV AX, [BX]
寄存器相对寻址	见表 3-3	MOV AL, BUF[SI]
基址变址寻址	见表 3-4	MOV AX, [BX+SI]
相对基址变址寻址	见表 3-4	MOV AX, DISP[BX+SI]

3.3 实验内容

【实验目的】 通过上机操作,理解 8086 CPU 常用的几种寻址方式,特别是存储单元逻辑地址的表示及指令的执行结果,熟练掌握 D、E、R、A、T 等 DEBUG 命令的用法。

【实验 3-1】 现有 DS=2000H, BX=0100H, SI=0002H, (20100H)=12H, (20101H)=34H, (20102H)=56H, (20103H)=78H, (21200H)=2AH, (21201H)=4CH, (21202H)=0B7H, (21203H)=65H, 试说明下列各条指令的寻址方式及执行后 AX 寄存器的内容。要

求通过 DEBUG 上机验证,比较不同寻址方式的特点,说明上机操作方法和运行结果。

- (1) MOV AX, 2000H ; _____ 寻址方式; AX = _____
 (2) MOV AX, BX ; _____ 寻址方式; AX = _____
 (3) MOV AX, [1200H] ; _____ 寻址方式; AX = _____
 (4) MOV AX, [BX] ; _____ 寻址方式; AX = _____
 (5) MOV AX, 1100H[BX] ; _____ 寻址方式; AX = _____
 (6) MOV AX, [BX + SI] ; _____ 寻址方式; AX = _____
 (7) MOV AX, 1100H[BX][SI] ; _____ 寻址方式; AX = _____

【实验 3-2】 已知(6000H)=12H, (6001H)=34H, (6002H)=56H, (6003H)=78H, (7000H)=0ABH, (7001H)=0CDH, (7002H)=0EFH, (7003H)=0DH。分别执行下列指令后,填入指定寄存器的当前内容,并通过 DEBUG 上机验证,说明上机操作方法和运行结果。

```
MOV AX, 6000H
MOV DS, AX
MOV AX, 7000H
MOV SS, AX
MOV BX, 0
MOV BP, 0
MOV SI, 2
MOV AX, BX ; AX = _____
MOV AX, [0000H] ; AX = _____
MOV AL, [0000H] ; AL = _____
MOV AX, [BX] ; AX = _____
MOV AL, [BX + 1] ; AL = _____
MOV AX, [BX + 1] ; AX = _____
MOV AL, [BX + 2] ; AL = _____
MOV AX, [BX + 2] ; AX = _____
MOV AX, [BX + SI] ; AX = _____
MOV AX, [BP + SI] ; AX = _____
```

习 题

1. 写出下列指令中存储器操作数的物理地址表达式。

- (1) MOV AL, [BX + 5]
 (2) MOV 3[BP], AX
 (3) MOV AX, BUFFER[BX + DI]
 (4) MOV DL, ES:[SI]

2. 8086 系统有哪几种基本的寻址方式? 立即寻址和直接寻址有什么不同? 寄存器寻址和寄存器间接寻址有什么不同?

3. 已知 DS=2000H, SS=3000H, BX=1234H, BP=0045H, SI=1030H, DISP=25H, 说明下列指令中采用的寻址方式,并计算出存储器操作数的物理地址。

- (1) MOV AX, [BX]
- (2) MOV AX, [BX + DISP]
- (3) MOV AX, [BP][SI]
- (4) MOV AX, DISP[BX][SI]

4. 下列指令执行后,寄存器 AX 和 BX 内容各是什么? 要求通过 DEBUG 上机验证。

```
MOV AX, 1234H
MOV BX, AX
MOV AH, AL
MOV [BX], AX
```