

第4章 操作数的寻址方式

- 4.1 立即寻址方式
- 4.2 寄存器寻址方式
- 4.3 直接寻址方式
- 4.4 寄存器间接寻址方式
- 4.5 寄存器相对寻址方式
- 4.6 基址变址寻址方式
- 4.7 相对基址变址寻址方式

第4章 操作数的寻址方式

- 计算机中的指令由操作码和操作数组成。
- 操作数字段可以有一个、两个或三个，通常称为一地址、二地址或三地址指令。
- 二地址指令中两个操作数分别称为源操作数和目的操作数。
- 所谓寻址方式就是指令中寻找操作数的方式。

- **80x86汇编语言指令的一般格式位：**
[标号:] 指令助记符 [操作数] [;注释]
- **[]中的内容位可选项。**

- 标号：符号地址，表示指令在内存中的位置。
标号后应加冒号：。
- 指令助记符：指令名称，是指令功能的英文缩写。
- 操作数：指令要操作的数据或数据所在的地址。
寄存器，常量，变量，表达式。
- 注释：每行以分号“；”开头，汇编程序不处理。

4.1 立即寻址方式

- 立即寻址方式

操作数就在**指令中**，紧跟在操作码之后，操作数作为指令的一部分存放在代码段。

- 例4.1 **MOV AL, 6H**

执行完此条指令后 (**AL**) = **06H**

- 例4.2 **MOV AX, 12AFH**

AX=12AFH，即**AH=12H**，**AL=AFH**

C:\>DEBUG

-A

0B13:0100 MOV AL,6

0B13:0102 MOV AX,12AF

0B13:0105

-U100L5

0B13:0100 B006 MOV AL,06

0B13:0102 B8AF12 MOV AX,12AF

-T=100

AX=0006 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=0B13 ES=0B13 SS=0B13 CS=0B13 IP=0102 NU UP EI PL NZ NA PO NC

0B13:0102 B8AF12 MOV AX,12AF

-T

AX=12AF BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=0B13 ES=0B13 SS=0B13 CS=0B13 IP=0105 NU UP EI PL NZ NA PO NC

0B13:0105 90 NOP

-

注意：

- 执行时无需去内存取数，因此称为立即数。
- 主要用于寄存器赋初值。
- 立即数只能作为源操作数，并且长度与目的操作数一致。

4.2 寄存器寻址方式

操作数就是**寄存器中**的值。指令中给出寄存器名。

例3.3 **MOV AX, BX**

指令执行后，**AX=BX**，**BX**保持不变。

- 例 **MOV AL, BL**
MOV AX, BX
MOV AL, BX
MOV AX, BL

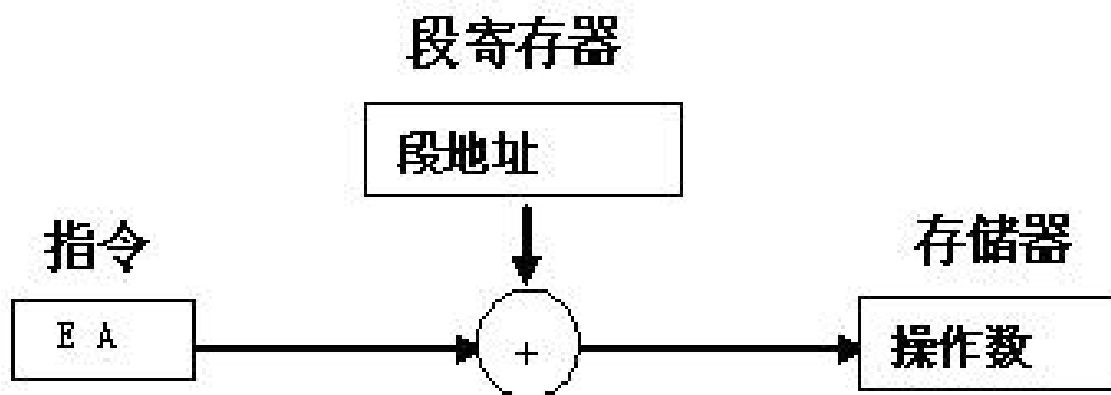
错
错

注意：

- 以上两种寻址方式都与存储器无关。
- 以下各种寻址方式的操作数都在存储器中。
- 偏移地址也称为**有效地址（EA）**。

4.3 直接寻址方式

- 操作数的有效地址EA就在指令中。机器默认段地址在DS中。



4.3 直接寻址方式

- (1) 存储器读操作
- 例4-4 MOV AX, DS: [2000H]
- 该指令表示从数据段的2000H单元读出一个字送入AX。
- 其中(DS)=1500H, (17000H)=31H, (17001H)=65H, (AX)=1020H。则
- 有效地址EA=2000H
- 物理地址=(DS)*10H+EA=15000H+2000H=17000H
- 执行指令后: (AX)=6531H

4.3 直接寻址方式

- 课堂习题
 - **MOV AX, DS: [1008H]**
 - (DS) = 3000H
 - (31007H) = ABH
 - (31008H) = CDH
 - (31009H) = EFH
 - (31010H) = 12H
 - (AX) = 1234H
- 执行后**AX** = ?

4.3 直接寻址方式

- 课堂习题
 - **MOV AX, DS: [1008H]**
 - (DS) = 3000H
 - (31007H) = ABH
 - (31008H) = CDH
 - (31009H) = EFH
 - (31010H) = H
 - (AX) = 1234H
- 执行后**AX = EFCDH**

4.3 直接寻址方式

- (2) 存储器写操作
- 如果要实现CPU写内存操作，只要把MOV指令的目的操作数变为存储单元，源操作数为CPU的寄存器即可。
- 例4-5 MOV DS: [4000H], AX
- 将AX的值写入数据段的4000H单元。已知(DS)=1500H, (AX)=3946H。则有效地址EA=4000H
- 物理地址 $= (DS) * 10H + EA$
- $= 15000H + 4000H = 19000H$
- 执行指令后: (19000H)=46H
- (19001H)=39H

4.3 直接寻址方式

- 课堂习题
 - MOV DS: [1050H], CL
 - (DS) = 2500H
 - (CL) = F2H
 - 初始内存值:
 - (26040H) = 44H
 - (26050H) = 55H
 - (26060H) = 66H
- 执行后哪一行数据会发生改变？改变后的值是

4.3 直接寻址方式

- 课堂习题

- MOV DS: [1050H], CL

- (DS) = 2500H
 - (CL) = F2H
 - 初始内存值:
 - (26040H) = 44H
 - (26050H) = 55H
 - (26060H) = 66H

- 执行后哪一行数据会发生改变？ 改变后的值是

- (26050H) = F2H

4.3 直接寻址方式

- **(3) 符号地址**
- 直接寻址方式除了用数值作为有效地址之外，还可以用符号地址的形式。为存储单元定义一个名字，该名字就是符号地址。如果把存储单元看成变量，该名字也是变量名。

4.3 直接寻址方式

- (3) 符号地址
- 例4-6
- VALUE DW 5678H
- MOV AX, VALUE
- MOV AX, [VALUE]
- 有效地址(EA)=VALUE=1000H
- 物理地址 $= (DS) * 10H + EA = 15000H + 1000H = 16000$

4.3 直接寻址方式

- (4) 段前缀

在与内存有关的寻址方式中，操作数的段地址默认为数据段，80X86规定除了数据段之外，数据还可以存放在其他三种段中。如果操作数在其他段中存放，称为段超越，需要在指令中用段超越前缀指出，即用操作数前加上段寄存器名和冒号表示。

4.3 直接寻址方式

- 例4-7 VALUE DW 1000H

MOV AX, DS: [VALUE]

MOV AX, ES: [VALUE]

若已知(ES)=3600H, EA=VALUE=1000H, 则有段超越
前缀ES的指令源操作数的物理地址计算为:

物理地址=(ES)*10H+EA=36000H+1000H=37000H

若字单元(37000H)=9091H

执行完MOV AX, ES: [VALUE]后: (AX)=9091H

4.4 寄存器间接寻址方式

- 操作数的有效地址在寄存器中，只允许使用BX、BP、SI和DI寄存器。

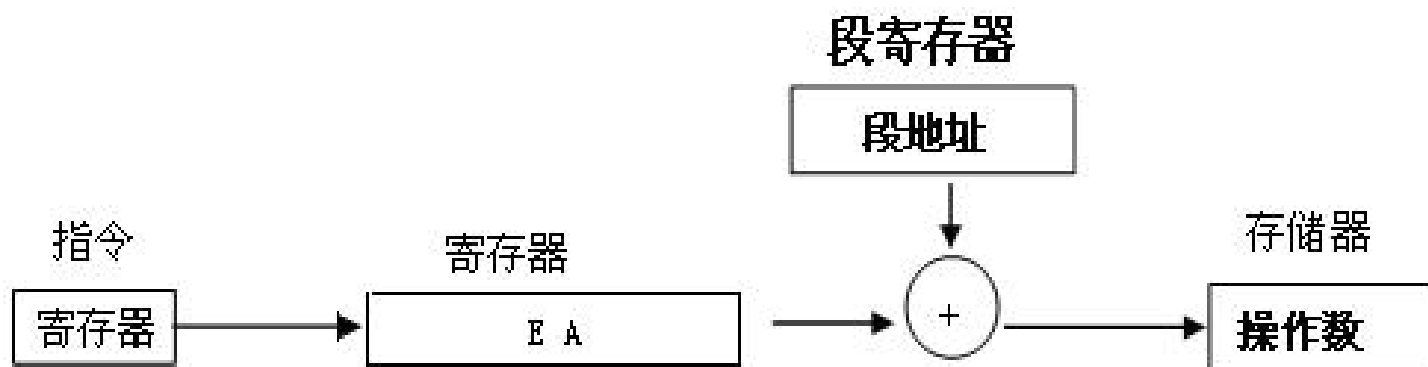
物理地址=10H x (DS) + (BX)

物理地址=10H x (DS) + (SI)

物理地址=10H x (DS) + (DI)

物理地址=10Hx (SS) + (BP)

4.4 寄存器间接寻址方式



4.4 寄存器间接寻址方式

- 例4-8 MOV AX,[BX]
- 已知 (DS) = 1500H, (BX) = 4580H, 则
- $EA = (BX) = 4580H$
- 物理地址 = $(DS) * 10H + EA = 15000H + 4580H = 19580H$
- 若 (19580H) = 2364H
- 执行指令后: (AX) = 2364H

4.4 寄存器间接寻址方式

- 例4-9 MOV SS:[DI],AX
- 已知 (ES) = 2500H, (DI) = 5318H, 则
- EA = (DI) = 5318H
- 物理地址 = (SS) * 10H + EA = 25000H + 5318H = 2A318H
- 若 (AX) = 2468H
- 执行指令后: (2A318H) = 68H
- (2A319H) = 24H

4.4 寄存器间接寻址方式

- 例4-10
- MOV AX, [BX] ; 默认DS寄存器作段地址
- MOV DX, [BP] ; 默认SS寄存器作段地址
- MOV ES:[DI], AX ; 指定ES寄存器作段地址

4.5 寄存器相对寻址方式

- 操作数的**有效地址**是一个**寄存器**和**位移量**之和。

物理地址=10H x (DS) + (BX) + 8(16)位位移量

物理地址=10H x (DS) + (SI) + 8(16)位位移量

物理地址=10H x (DS) + (DI) + 8(16)位位移量

物理地址=10H x (SS) + (BP) + 8(16)位位移量

4.5 寄存器相对寻址方式

- 例4-11 **MOV AX, TOP[SI]**
- 以上指令**TOP**为符号地址，即位移量。
- 已知 **(DS) = 1500H**，**(SI) = 7310H**，**TOP = 25H**，则
- 有效地址 **EA = (SI) + TOP = 7310H + 25H = 7335H**
- 物理地址 = **(DS) * 10H + EA = 15000H + 7335H = 1C335H**
- 若 **(1C335H) = 2428H**，执行指令后，**(AX) = 2428H**

4.5 寄存器相对寻址方式

- 课堂习题
- **MOV CX, ES:OFFSET[DI]**
- **(ES) = 5000H (DI) = 2050H OFFSET = 60H**
- **OFFSET**是符号地址
- 物理地址
 - **(520A0H) = A87BH**
 - **(520B0H) = B87BH**
 - **(520C0H) = C87BH**
- 执行后**CX = ?**

4.5 寄存器相对寻址方式

- 课堂习题
- **MOV CX, ES:OFFSET[DI]**
- **(ES) = 5000H (DI) = 2050H OFFSET = 60H**
- **OFFSET**是符号地址
- 物理地址
 - **(520A0H) = A87BH**
 - **(520B0H) = B87BH**
 - **(520C0H) = C87BH**
- 执行后**CX = B87BH**

4.5 寄存器相对寻址方式

- 课堂习题
- **MOV AX, TABLE[BX][SI]**
- **(DS) = 4000H (BX) = 0100H (SI) = 0020H TABLE = 1000H**
- 物理地址
 - **(41110H) = F0D1H**
 - **(41120H) = F0D2H**
 - **(41130H) = F0D3H**
- 执行后**AX = ?**

4.5 寄存器相对寻址方式

- 课堂习题
- **MOV AX, TABLE[BX][SI]**
- **(DS) = 4000H (BX) = 0100H (SI) = 0020H TABLE = 1000H**
- **OFFSET**是符号地址
- 物理地址
 - **(4110H) = F0D1H**
 - **(41120H) = F0D2H**
 - **(41130H) = F0D3H**
- 执行后**AX = F0D2H**

4.5 寄存器相对寻址方式

- 例4-12 **MOV [BX+2623H], AX**
- 或写成**MOV [BX].2623H, AX**
- 已知 **(DS) = 1500H**, **(BX) = 6854H**, 则
- 有效地址**EA = (BX) + 2623H = 8E77H**
- 物理地址**= (DS) * 10H + EA = 15000H + 8E77H = 1DE77H**
- 若 **(1DE77H) = 3567H**, 执行指令后: **(AX) = 3567H**

4.5 寄存器相对寻址方式

- 例4-13
- `MOV AX, ARRAY[BX]`
- `MOV AX, [ARRAY][BX]`
- `MOV AX, [ARRAY+BX]`
- `MOV AL, BUF[BX]`
- `MOV AL, [BX+8H]`
- `MOV AL, [BX].8H`

4.6 基址变址寻址方式

- 操作数的有效地址是一个基址寄存器和一个变址寄存器的内容之和。

基址寄存器BX和BP，变址寄存器SI和DI。

默认段寄存器搭配和寄存器间接寻址方式一样。

4.6 基址变址寻址方式

- 例4-14 **MOV AX, [BX+DI]**
- 执行前：已知 $(DS) = 2100H$, $(BX) = 0158H$, $(DI) = 10A5H$, $(221FD) = 34H$, $(221FE) = 95H$, $(AX) = 0FFFFH$ 。则
- 有效地址 $EA = (BX) + (DI) = 0158H + 10A5H = 11FDH$
- 物理地址 $= (DS) * 10H + EA = 21000H + 11FDH = 221FDH$
- 执行后, $(AX) = 9534H$

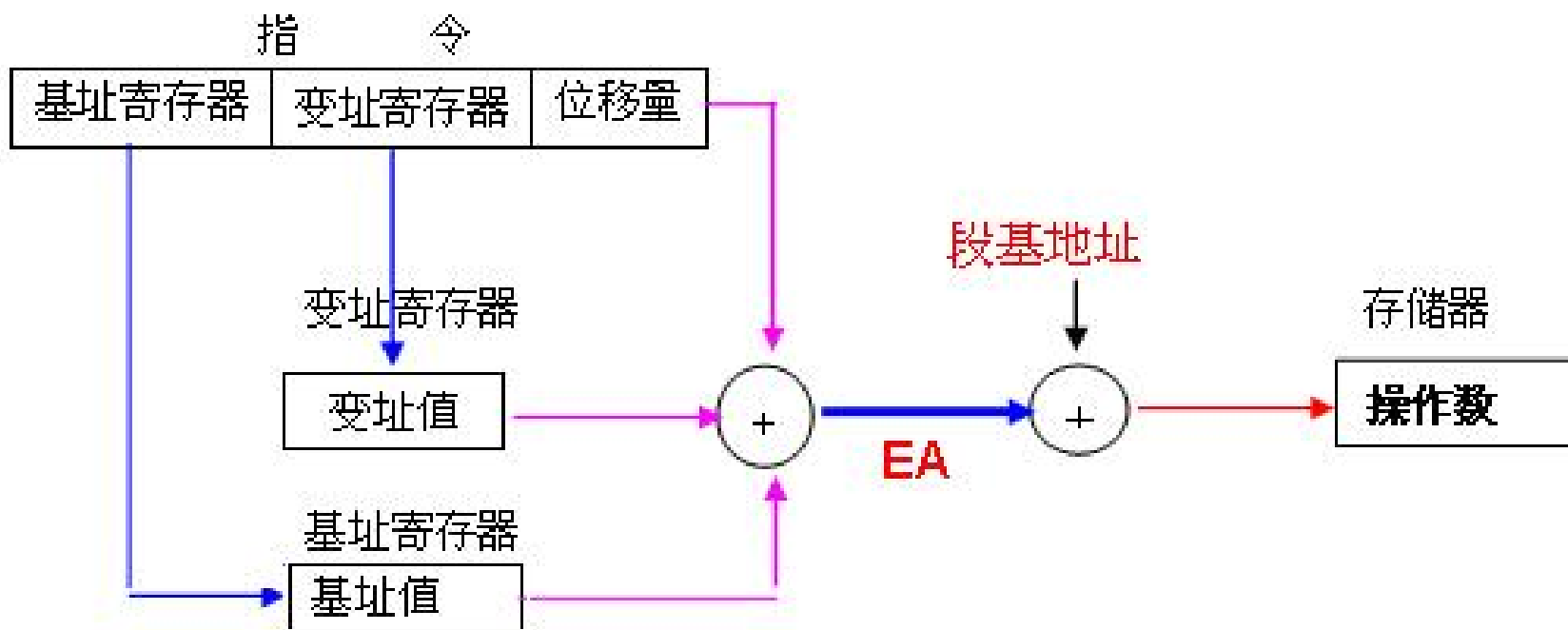
4.6 基址变址寻址方式

- 例4-15
- **MOV AX, [BX][SI]** ; 默认**DS**寄存器作段地址
- **MOV AX, [BP][DI]** ; 默认**SS**寄存器作段地址
- **MOV AX, ES:[BX][DI]** ; 指定**ES**寄存器作段地址
- **MOV DX, [BP][SI]** ; 默认**SS**寄存器作段地址
- **MOV [BX+DI], CX** ; 默认**DS**寄存器作段地址
- **MOV [BP+SI], AL** ; 默认**SS**寄存器作段地址

4.7 相对基址变址寻址方式

- 操作数的有效地址是一个基址寄存器和一个变址寄存器以及一个位移量之和。
基址寄存器BX和BP，变址寄存器SI和DI。
默认段寄存器搭配和寄存器间接寻址方式一样。

4.7 相对基址变址寻址方式



4.7 相对基址变址寻址方式

- 例4-16
- **MOV AX, MASK[BX][SI]** ; 默认**DS**寄存器作段地址
- **MOV AX, [MASK+BX+SI]** ; 默认**DS**寄存器作段地址
- **MOV AX, [BX+SI].MASK** ; 默认**DS**寄存器作段地址
- 以上三种表示形式实现的功能是一样的。其有效地址
EA=MASK+(BX)+(SI); 物理地址=**(DS)*10H+EA**。

本章小结

- 各种寻址方式总结
- 双操作数指令的两点提示：
 - （1） 双操作数指令的两个操作数，长度须匹配。
 - （2） 双操作数指令的两个操作数中，不能两个操作数同为内存单元。

本章小结

- 操作数的有效地址前可以加段跨越前缀，但在以下三种情况下不允许：
 - 串处理指令的目的串必须用**ES**段
 - **PUSH**指令的目的和**POP**指令的源必须用**SS**段
 - 指令必须存放在**CS**段