汇编语言程序设计
第三章 寻址方式与指令系统 I

张华平 副教授 博士
Email: kevinzhang@bit.edu.cn
Website: http://www.nlpir.org/
@ICTCLAS张华平博士
大数据搜索挖掘实验室（wSMS@BIT）

基本概念

指令集和指令系统

指令的构成：操作码 操作数

如何给出操作数或操作数地址？

操作数域的表示比较复杂，这就是寻址方式要解决的问题。

寻址方式/指令系统

寻址方式：

与数据有关的寻址方式

与转移地址有关的寻址方式

指令系统：

数据传送/算术运算/逻辑

程序控制/串操作/···

3.1 与数据有关的寻址方式

与数据有关的寻址方式与操作数有关。

格式：MOV 目标,源

功能：源→目标

目标和源是操作数域，各自可以有不同的寻址方式。下边以源操作数的寻址方式为例说明。

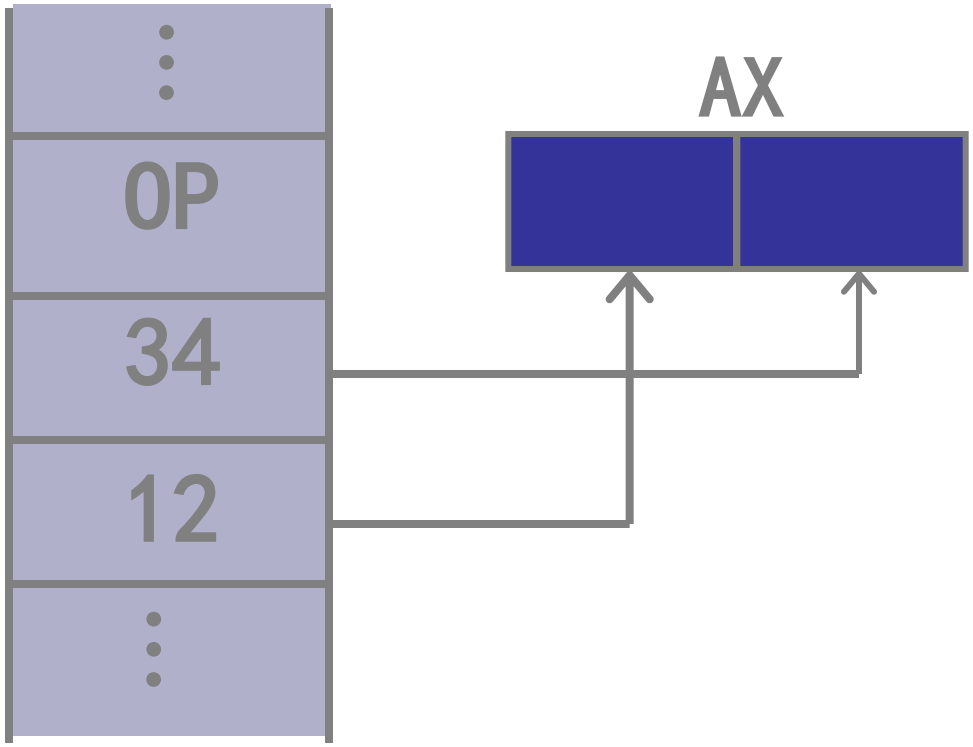
1. 立即寻址方式

操作数直接包含在指令中，紧跟在操作码之后的寻址方式称为立即寻址方式，把该操作数称为立即数。

注意：立即寻址方式只能出现在源操作数的位置。其它寻址方式既可以出现在源也可以出现在目标操作数位置。

例1.

```
MOV   AX,1234H
;(AX)＝1234H
```



立即寻址方式

2. 寄存器寻址方式

操作数直接包含在寄存器中，由指令指定寄存器号的寻址方式。

寄存器可以是8位、16位、32位通用寄存器或16位段寄存器(但CS不能用于目标)。

```
MOV   BX,AX      ;（BX）＝（AX）
MOV   DI,5678H       ;（DI）＝ 5678H
MOV   AL,78H         ;（AL）＝ 78H
MOV   ECX,7890ABCDH
                     ;（ECX）＝ 7890ABCDH
```

其中BX、AX、DI、AL、ECX均为寄存器寻址方式。

除以上两种寻址方式外，以下各种寻址方式的操作数都在存储器中，其操作数称为存储器操作数。

由于80X86对内存采用分段管理，因此由以下寻址方式得到的只是有效地址（简写为EA，在IBM PC中就是操作数地址的偏移量部分），

物理地址=段基址*10H+有效地址

段基址与段寄存器有关，选用段寄存器的情况参阅表2-3。

表2-3 段基址和偏移量的约定情况

操作类型	约定段寄存器	允许指定的段寄存器	偏移量
1. 指令	CS	无	IP
2. 堆栈操作	SS	无	SP
3. 普通变量	DS	ES、SS、CS	EA
4. 字符串指令的源串地址	DS	ES、SS、CS	SI
5. 字符串指令的目标串地址	ES	无	DI
6. BP用作基址寄存器	SS	DS、ES、CS	EA

若工作在实模式，段基址为段寄存器中的内容乘以16的值。

若工作在保护模式，段基址通过段寄存器中的段选择子从描述符中得到。

本章例中的物理地址按实模式计算。

当访内操作类型允许指定段寄存器时，可以使用段超越前缀指定。

功能：明确指出本条指令所要寻址的内存单元在哪个段中。

格式：段寄存器名：

例. ES:、CS:、SS:等。

```
MOV   AX,ES:VER
```

3. 直接寻址方式

操作数的有效地址直接包含在指令中的寻址方式。

有效地址存放在代码段的指令操作码之后，但操作数本身在存储器中，所以必须先求出操作数的物理地址。普通变量缺省情况是存放在DS所指向的数据段，但允许使用段超越前缀指定为其它段。

这种寻址方式常用于存取简单变量。

例7. MOV AL,[78H]

[78H]为直接寻址方式。由于没有使用段超越前缀,因此缺省使用DS段寄存器。

在实模式中，若DS ＝ 3000H

（30078H）＝ 12H

则DS:78H表示物理地址30078H

该指令的执行结果是（AL）＝ 12H

由于在汇编语言中用符号表示地址,所以指令“MOV　AL,VAR”中的源操作数寻址方式是直接寻址，VAR是内存的符号地址。实际上在汇编语言源程序中所看到的直接寻址方式都是用符号表示的,只有在DEBUG环境下，才有[78H]这样的表示。

例8. MOV AL, ES:[78H]

(ES:78H)→AL

该指令的源操作数前使用了段超越前缀“ES:”，明确表示使用附加数据段中的变量。

注意[78H]与ES:[78H]表示不同的物理地址，只是段内偏移量相同而已。

4. 寄存器间接寻址方式

操作数有效地址在基址寄存器BX、BP或变址寄存器SI、DI中，而操作数在存储器中的寻址方式。

对于386以上CPU，这种寻址方式允许使用任何32位通用寄存器。

若指令中使用的是BX、SI、DI、EAX、EBX、ECX、EDX、ESI、EDI，则缺省情况操作数在数据段，即它们默认与DS段寄存器配合。

若使用的是BP、EBP、ESP,则缺省情况默认与SS段寄存器配合。

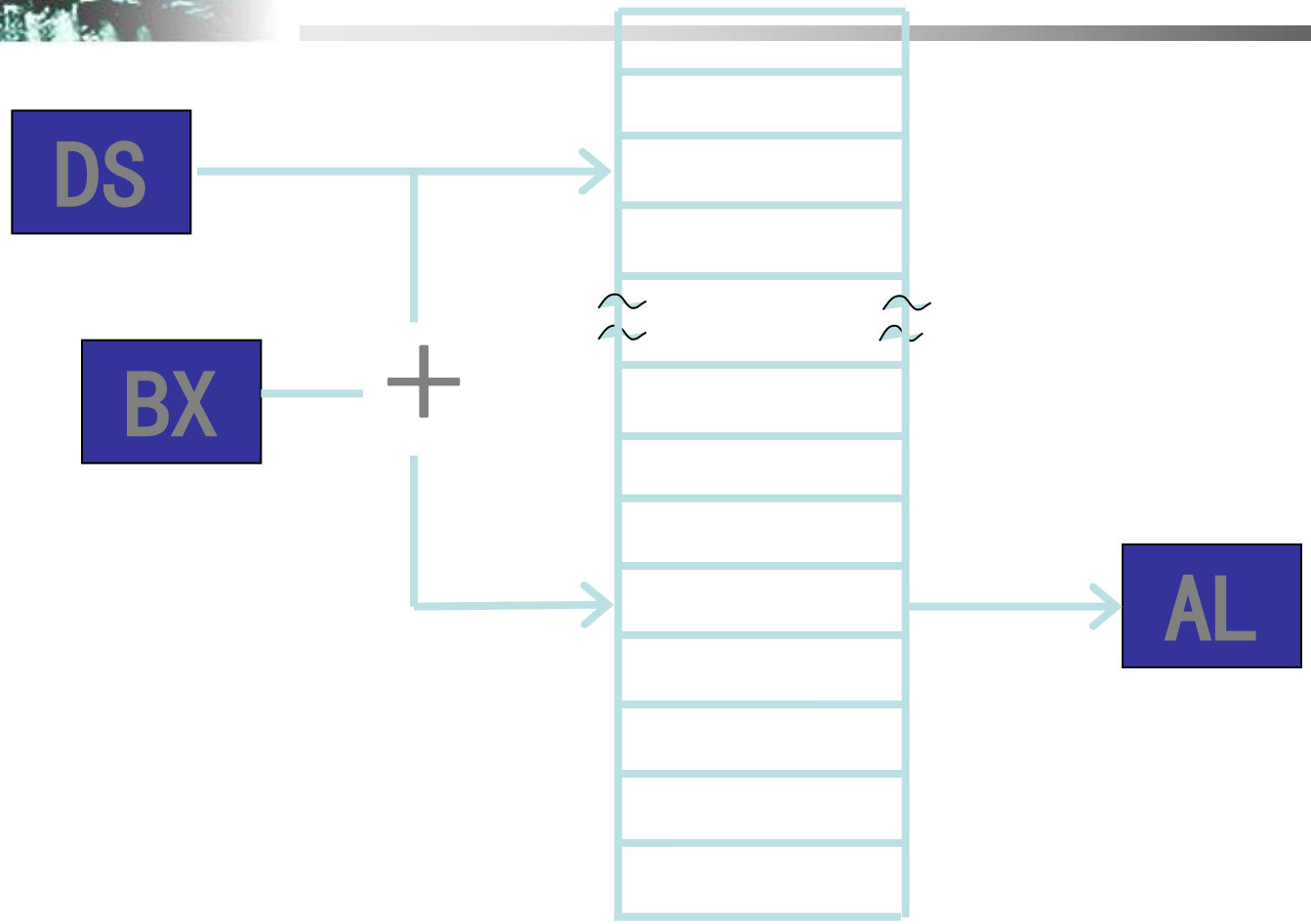
均允许使用段超越前缀。

例9. MOV AL,[BX]

;（DS:[BX]）→AL

;如下图所示

其中[BX]为寄存器间接寻址方式，注意它与寄存器寻址方式在汇编格式上的区别。



寄存器间接寻址方式执行情况

对于MOV　AL,[BX]

若：（DS）＝ 3000H,　（BX）＝ 78H

（30078H）＝ 12H

则：

物理地址＝10H×(DS)＋(BX)＝30078H

该指令的执行结果是（AL）＝ 12H

例10. MOV AX,[BP] ;（SS:[BP]）→AX

若（SS）＝ 2000H, （BP）＝ 80H

（20080H）＝ 12H, （20081H）＝ 56H

物理地址：10H ×（SS）＋（BP）＝ 20080H

执行结果：（AX）＝ 5612H。

利用这种寻址方式再配合修改寄存器内容的指令可以方便地处理一维数组。

5. 寄存器相对寻址方式

操作数的有效地址是一个基址(BX、BP)或变址寄存器(SI、DI)的内容和指令中给定的一个位移量（disp）之和。

386以上允许使用任何32位通用寄存器。位移量可以是一个字节、一个字、一个双字（386以上）的带符号数。

EA＝（基址＜或变址＞寄存器）＋disp

或：EA＝（32位通用寄存器）＋disp

与段寄存器的配合情况：若指令中寄存器相对寻址方式使用BP、EBP、ESP，则默认与SS段寄存器配合。使用其它通用寄存器，则默认与DS段寄存器配合。均允许使用段超越前缀。

例11. MOV AL,8 [BX]

可以表示为： MOV AL,[BX+8]

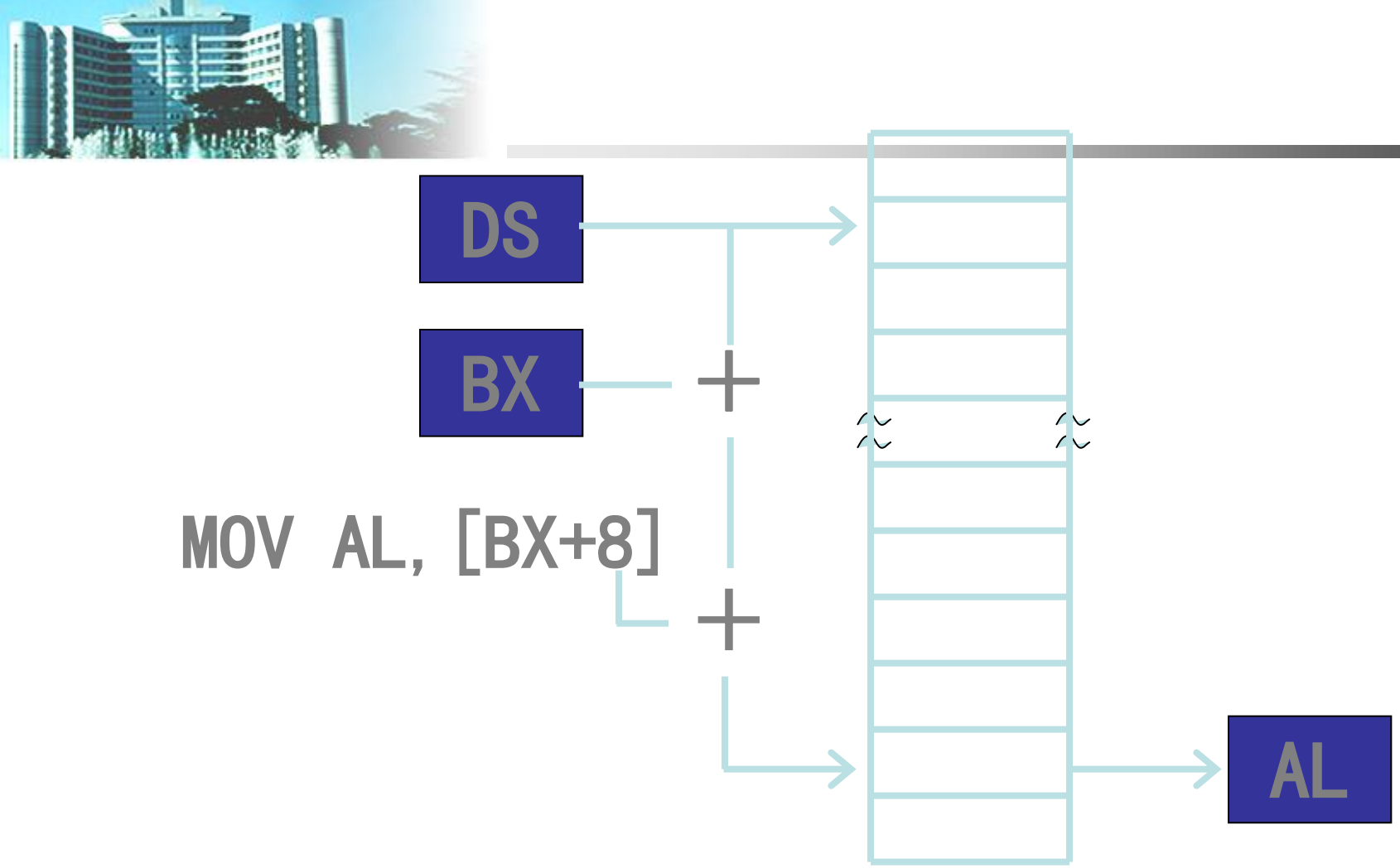
若 （DS）＝ 3000H, （BX）＝ 70

（30078H）＝ 12H

则物理地址＝30078H

该指令的执行结果是（AL）＝12H。

其中8 [BX]为寄存器相对寻址方式,该指令执行情况如下图所示。



寄存器相对寻址执行情况

例12. MOV AL,TABLE[BX]

也可以表示为：MOV AL,[BX+TABLE]

其中TABLE为位移量的符号表示，指令执行结果是（DS:[BX+TABLE]）→AL。

使用这种寻址方式可以访问一维数组，其中TABLE是数组起始地址的偏移量，寄存器中是数组元素的下标乘以元素的长度（一个元素占用的字节数），下标从0开始计数。

6. 基址变址寻址方式

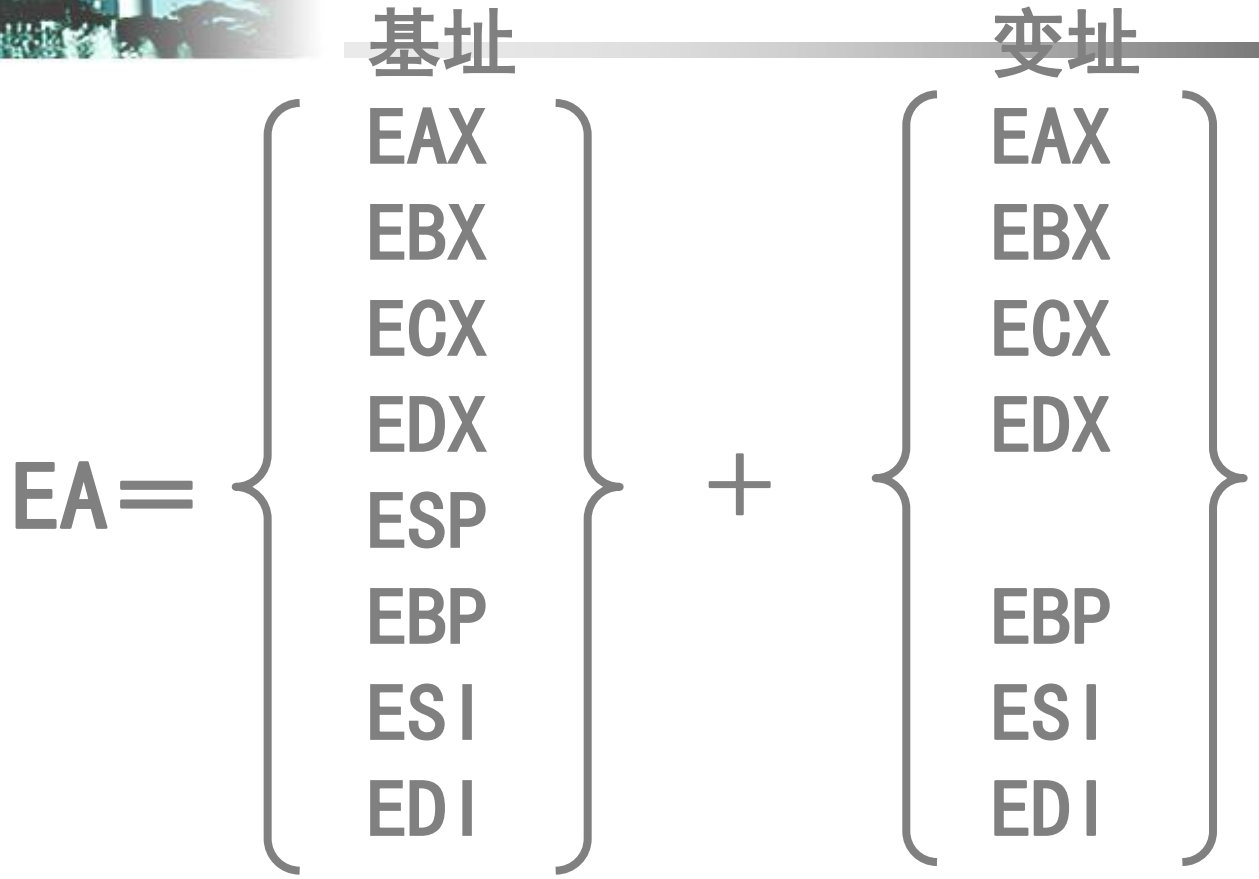
操作数的有效地址是一个基址寄存器和一个变址寄存器的内容之和。

386以上允许使用变址部分除ESP以外的任何两个32位通用寄存器组合。

缺省使用段寄存器的情况由基址寄存器决定。若使用BP、ESP或EBP，缺省与SS配合；若使用BX或其它32位通用寄存器，则缺省与DS配合。允许使用段超越前缀。

即：EA＝(基址寄存器)＋(变址寄存器)

80386以上支持的32位基址变址寻址方式组合如下图。



80386以上32位基址变址寻址方式组合

例13. MOV AL, [BX] [SI]

; (DS:[BX+SI]) →AL

该指令可以表示为:

MOV AL,[BX+SI]

动画演示

执行情况如下图所示。

使用这种寻址方式可以访问一维数组,其中BX存放数组起始地址的偏移量，SI存放数组元素的下标乘以元素的长度，下标从0开始计数。

7. 相对基址变址寻址方式

操作数的有效地址是一个基址和一个变址寄存器的内容和指令中给定的一个位移量之和。

386以上CPU允许使用变址部分除ESP以外的任何两个32位通用寄存器组合。位移量可以是一个字节、一个字、一个双字（386以上）的带符号数。

缺省使用段寄存器的情况由基址寄存器决定。

若使用BP、EBP或ESP，缺省与SS配合；若使用BX或其它32位通用寄存器，缺省与DS配合。

允许使用段超越前缀。

即：EA＝（基址寄存器）＋（变址寄存器）＋disp

80386以上支持的32位相对基址变址寻址方式组合如下图。

例14. MOV AL, ARY[BX] [SI]

; (DS:[BX+SI+ARY]) →AL

可表示为:

MOV AL, ARY [BX+SI]

或 MOV AL,[BX+SI+ARY]

动画演示

执行情况如下图所示。

使用这种寻址方式可以访问形如ARY[3][3]的二维数组，下标从0开始计数。

8. 比例变址寻址方式

这种寻址方式是80386以上的微处理器才提供的。

操作数的有效地址由以下几部分相加组成：基址部分(8个32位通用寄存器)、变址部分（除ESP以外的32位通用寄存器）乘以比例因子、位移量（disp）。

比例因子可以是1(缺省值)、2、4或8，1可用来寻址字节数组，2可用来寻址字数组，4可用来寻址双字数组,8可用来寻址4字数组。位移量可以是一个字节、一个双字的带符号数。缺省使用段寄存器的情况由基址寄存器决定。

即：EA＝（基址寄存器）＋（变址寄存器）×比例因子＋disp

比例变址寻址方式组合如下图所示。可以看出，它实际上是386以上CPU存储器操作数寻址方式的通用公式。除比例因子不能单独使用外，其它各项都可以独立存在或以组合形式出现。

例如若只含有第一列或第二列，就变成寄存器间接寻址方式。若含有第一列和第二列，就变成基址变址寻址方式。

例15. MOV EAX, ARY[EBX] [ESI]

; (DS:[ARY+EBX+ESI]) → EAX

例16. MOV CX,[EAX+2*EDX]

; (DS:[EAX+2*EDX]) → CX

例17. MOV EBX,[EBP+ECX*4+10H]

; (SS:[EBP+ECX*4+10H]) → EBX

例18. MOV EDX, ES:ARY[4*EBX]

; (ES:[ARY+4*EBX]) → EDX

使用这种寻址方式可以方便地访问数组，其中变址寄存器的内容等于数组下标，比例因子为元素长度。

小结

- 立即寻址方式：MOV AX,5
- 寄存器寻址方式：MOV AX,5
- 直接寻址方式：MOV AX,[78H]
- 寄存器间接寻址方式 MOV AX,[BX]
- 寄存器相对寻址 MOV AX,8[BX]
- 基址变址寻址MOV AX,[BX][SI]
- 相对基址变址： MOV AX,ARRY[BX][SI]
- 比例变址寻址方式： MOV EAX,ES:ARRY[4*BX]

3.2 与转移地址有关的寻址方式

本节讨论的寻址方式是用来确定转移及调用指令的转向地址。

以8086／8088的无条件转移指令为例。

无条件转移指令格式：JMP 目标

功能：无条件转移到目标处。

其中的目标有各种寻址方式。这些寻址方式可以被分为段内转移和段间转移两类。段内转移只影响指令指针IP值；段间转移既要影响IP值，也要影响代码段寄存器CS的值。

一、段内直接寻址方式

转向的有效地址是当前指令指针寄存器的内容和指令中指定的8位、16位位移量之和，该位移量是一个相对于指令指针的带符号数。

即要转向的有效地址为：

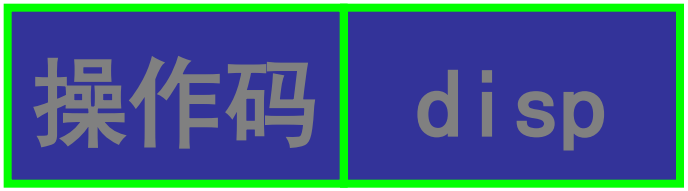
$$EA=(IP)+\left\{\begin{array}{c}8位\\16位\end{array}\right\}disp$$

EA就是要转向的本代码段内指令地址的偏移量。它是通过把IP的当前值加上指令中给出的位移量disp得到的，从而使IP指向下一条要执行的指令，实现段内转移。

若位移量是8位的，则称为短转移，可以实现在距离下条指令的＋127~－128字节范围之内转移

段内无条件短转移指令格式：

JMP　SHORT　LAB



段内短转移方式

若位移量是16位的，则称为近转移，可以实现在距离下条指令的＋32767~－32768(以下简称±32K)字节范围之内转移。

段内无条件近转移指令格式：JMP　LAB

或　　JMP　NEAR　PTR　LAB

该格式的机器指令

其位移量占两个字节

字节0	字节1	字节2
操作码	disp低	disp高

段内直接寻址方式

二、段内间接寻址方式

转向的有效地址在一个寄存器或内存单元中，该寄存器号或内存地址按上节介绍的与操作数有关的寻址方式（立即寻址方式除外）获得。所得到的有效地址送给IP，于是实现转移。指令格式及举例见下表。

表3-1　段内间接寻址方式

格　式	举　例	注　　释
JMP 通用寄存器	JMP　BX	16位转向地址在BX中
JMP 内存单元	JMP WORD PTR VAR JMP WORD PTR[BX]	16位转向地址在VAR字型内存变量中 16位转向地址在BX所指向的内存变量中

具体说明。

设：（DS）＝ 2000H，（BX）＝0300H

（IP）＝0100H，（20300H）＝000BH

（20301H）＝0005H

则：JMP BX

;执行后（IP）＝（BX）＝ 0300H

动画演示

例2. JMP WORD PTR [BX]

说明：式中WORD PTR [BX]表示BX指向一个字型内存单元。

1. 这条指令执行时，先按照操作数寻址方式得到存放转移地址的内存单元：

$$10H \times (DS) + (BX) = 20306H$$

2. 再从该单元中得到转移地址：

$$EA = (20306H) = 050BH$$

3. 于是，(IP)＝EA＝050BH，下一次便执行CS:50BH处的指令，实现了段内间接转移。

$$10H \times (CS) + (050bh) = 20306H$$

动画演示

三、段间直接寻址方式

指令中直接给出转向的4字节的偏移量和段基址,只需把它们分别送给IP和CS后即可实现段间转移。

指令格式:

JMP FAR PTR LAB

下图给出的是段间直接转移指令机器码示意图。执行时把偏移量送给IP，段基址送给CS，即可实现段间直接转移。

图3-11　段间直接寻址方式

动画演示

四、段间间接寻址方式

用一个双字内存变量中的低16位取代IP值，高16位取代CS值，从而实现段间转移。该双字变量的地址可以由除立即寻址和寄存器寻址方式以外的其它与数据有关的寻址方式获得。

具体数据说明：

设：（DS）＝ 2000H，（BX）＝ 0300H

（IP）＝ 0100H，（20300H）＝ 0

（20301H）＝ 05H

（20302H）＝ 10H

（20303H）＝ 60H

则：JMP DWORD PTR [BX]

说明：式中DWORD PTR [BX]表示BX指向一个双字变量。

这条指令执行时，先按照与操作数有关的寻址方式得到存放转移地址的内存单元：

10H×（DS）＋（BX）＝ 20300H

再把该单元中的低字送给IP，高字送给CS，即0500H →IP，6010H→CS，下一次便执行6010:0500H处的指令，实现了段间间接转移。

动画演示

上边介绍的与转移地址有关的寻址方式完全适用于386以上的实模式环境。保护模式下的虚拟8086方式转移地址的形成与此类似，只是32位的偏移量送给EIP，但高16位清0，这是因为段长不能超过64K的缘故。

对于32位的保护模式，其段内转移的目标地址的形成与以上所介绍的16位机相比较没有太大变化，只是偏移量为32位、并把该偏移量送给EIP指令指针寄存器而已。

而段间转移的目标地址采用48位全指针形式，即32位的偏移量和16位的段选择子。段间转移的目标地址的形成比较复杂，需要涉及到任务门、调用门等知识。

感谢关注聆听！

张华平
Email: kevinzhang@bit.edu.cn
微博：@ICTCLAS张华平博士
实验室官网：
http://www.nlpir.org

大数据千人会