第三章
80x86的指令系统和寻址方式

80X86的寻址方式

80X86的指令系统

80X86的寻址方式

- 与数据有关的寻址方式

- 与转移地址有关的寻址方式

指令的组成

- 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哪种操作，如传送、运算、移位、跳转等操作，它是指令中不可缺少的组成部分
- 地址码说明参与运算的操作数从哪里获取，结果存放何处
- 有些指令不需要操作数，通常的指令都有一个或两个操作数，也有个别指令有3个甚至4个操作数

指令的操作码和操作数

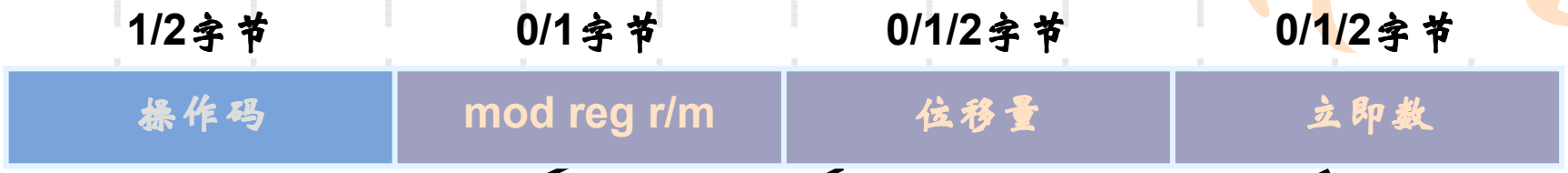
- 每种指令的操作码：
  - 用一个唯一的助记符表示（指令功能的英文缩写）
  - 对应着机器指令的一个二进制编码
- 指令中的地址码：
  - 可以是一个具体的数值
  - 可以是存放数据的寄存器
  - 或指明数据在主存位置的存储器地址

寻址方式

- 指令系统设计了多种操作数的来源
- 寻找操作数的过程就是操作数的寻址
- 操作数采取哪一种寻址方式，会影响机器运行的速度和效率

如何寻址一个操作数对程序设计很重要

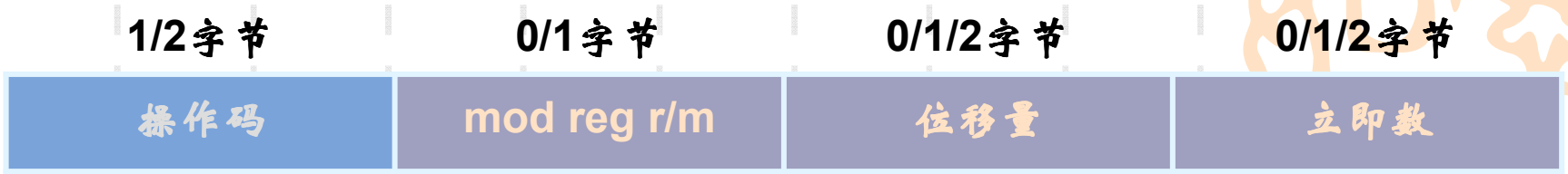
8086的机器代码格式



操作数

- 表明采用的寻址方式
- 给出某些寻址方式需要的对基地址的偏移量
- 给出立即寻址方式需要的数值本身

标准机器代码示例



mov ax,[BP+0]；机器代码是 8B 46 00

- 前一个字节8B是操作码
- 中间一个字节46（01 000 110）是“mod reg r/m”字节
  - reg＝000表示目的操作数为AX
  - mod＝01和r/m＝110表示源操作数为[BP+D8]
- 最后一个字节就是8位位移量［D8＝］00

其它机器代码形式

mov al,05 ；机器代码是B0 05

- 前一个字节B0是操作码（含一个操作数AL），后一个字节05是立即数

mov ax,0102H ；机器代码是B8 02 01

- 前一个字节B8是操作码（含一个操作数AX），后两个字节02 01是16位立即数（低字节02在低地址）

指令的助记符格式

示例

操作码　操作数1,操作数2 ；注释

- 操作数2，称为源操作数 src，它表示参与指令操作的一个对象
- 操作数1，称为目的操作数 dest，它不仅可以作为指令操作的一个对象，还可以用来存放指令操作的结果
- 分号后的内容是对指令的解释

传送指令MOV的格式

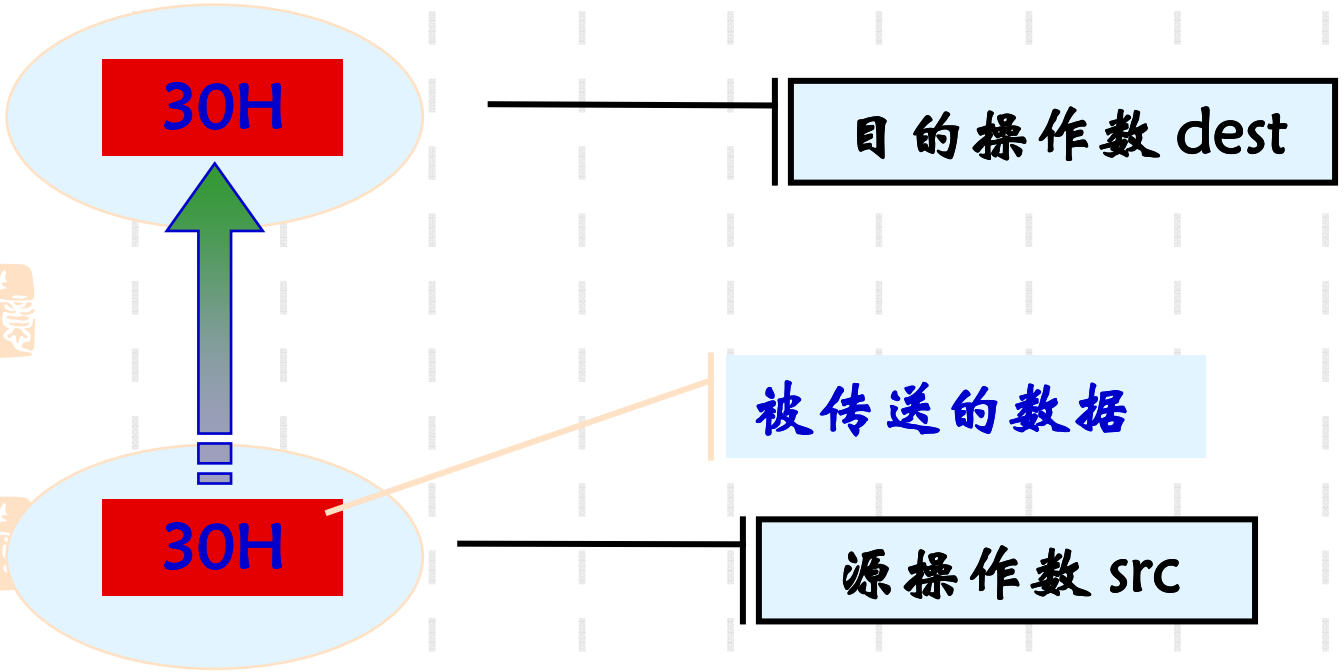
演示

MOV dest,src ; dest←src

- MOV指令的功能是将源操作数src传送至目的操作数dest，例如：

```
MOV AL,05H          ; AL←05H
MOV BX,AX           ; BX←AX
MOV AX,[SI]         ; AX←DS:[SI]
MOV AX,[BP+06H]     ; AX←SS:[BP+06H]
MOV AX,[BX+SI]      ; AX←DS:[BX+SI]
```

传送指令MOV的功能

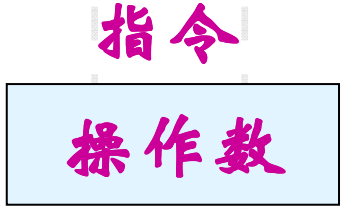


与数据有关的寻址方式

- 立即寻址方式
- 寄存器寻址方式
- 存储器寻址方式

立即数寻址方式

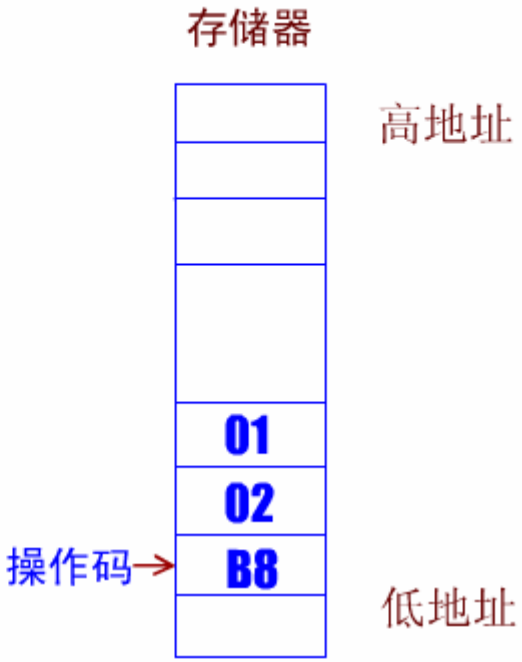
- 指令中的操作数直接存放在机器代码中，紧跟在操作码之后（操作数作为指令的一部分存放在操作码之后的主存单元中）



- 这种操作数被称为立即数imm
  - 它可以是8位数值i8（00H～FFH）
  - 也可以是16位数值i16（0000H～FFFFH）
- 立即数寻址方式常用来给寄存器赋值

```
MOV AL,05H        ; AL←05H
MOV AX,0102H      ; AX←0102H
```

立即数寻址方式



播放

停止

MOV AX,0102H

与数据有关的寻址方式

- 立即寻址方式
- 寄存器寻址方式
- 存储器寻址方式

寄存器寻址方式

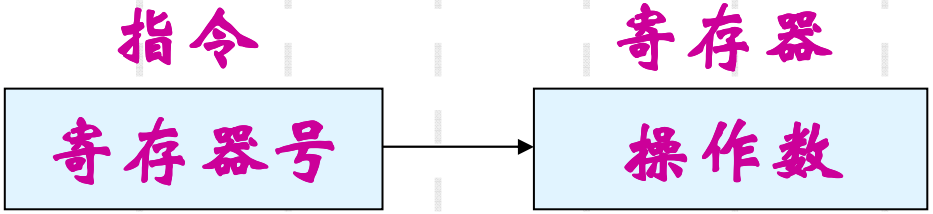
- 操作数存放在CPU的内部寄存器reg中，可以是：
  - 8位寄存器r8：

    AH、AL、BH、BL、CH、CL、DH、DL
  - 16位寄存器r16：

    AX、BX、CX、DX、SI、DI、BP、SP
  - 4个段寄存器seg：

    CS、DS、SS、ES

寄存器寻址方式



MOV AX,1234H；AX←1234H

MOV BX,AX　　；BX←AX

寄存器寻址方式

MOV AX,BX

播放
停止

与数据有关的寻址方式

- 立即寻址方式
- 寄存器寻址方式
- 存储器寻址方式

存储器寻址方式

- 指令中给出操作数的主存地址信息（偏移地址，称之为有效地址EA），而段地址在默认的或用段超越前缀指定的段寄存器中
- 8086设计了多种存储器寻址方式

1、直接寻址方式

2、寄存器间接寻址方式

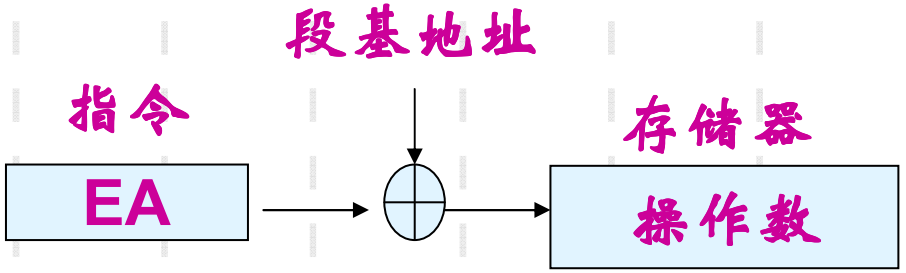
3、寄存器相对寻址方式

4、基址变址寻址方式

5、相对基址变址寻址方式

直接寻址方式

- 有效地址在指令中直接给出
- 默认的段地址在DS段寄存器，可使用段超越前缀改变



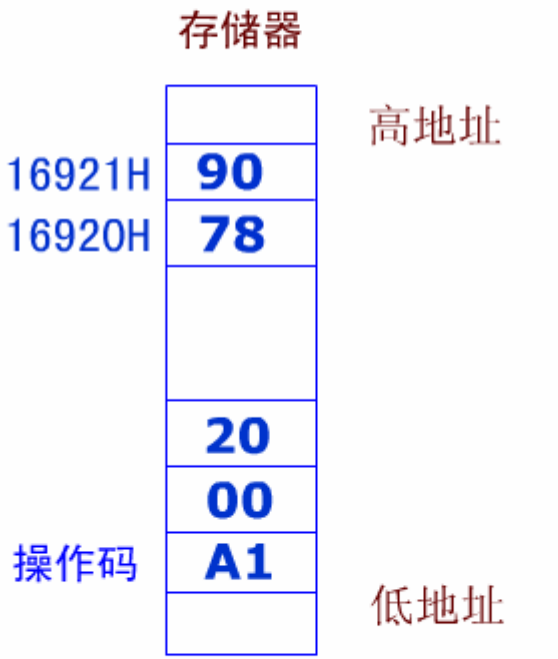
MOV AX,[2000H]

；AX←DS:[2000H]；指令代码：A10020

MOV AX,ES:[2000H]

；AX←ES:[2000H]；指令代码：26A10020

直接寻址方式



MOV AX,[2000H]

播放

停止

直接寻址

存储器寻址方式

1、直接寻址方式

2、寄存器间接寻址方式

3、寄存器相对寻址方式

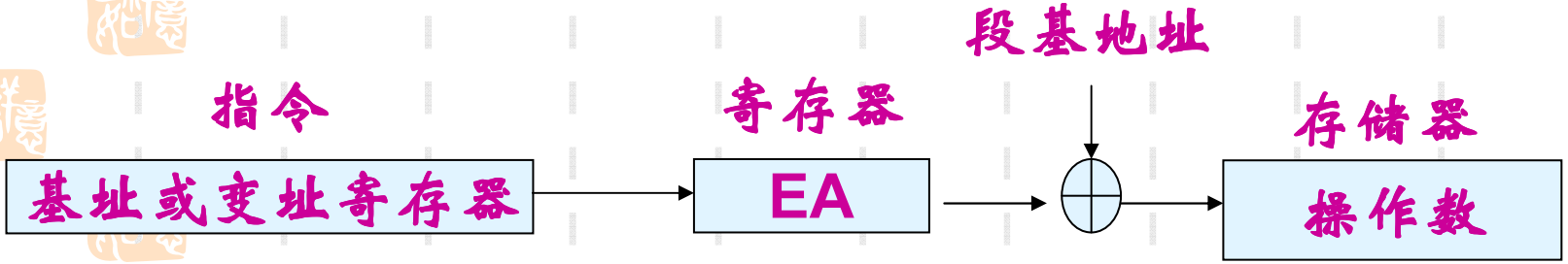
4、基址变址寻址方式

5、相对基址变址寻址方式

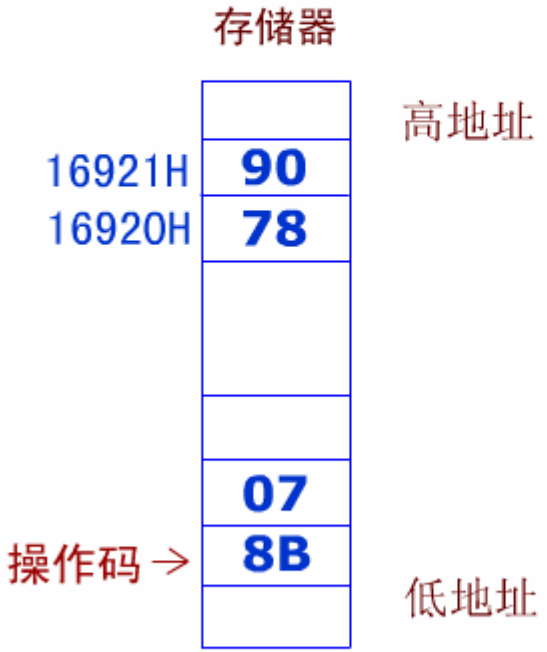
寄存器间接寻址方式

- 有效地址存放在基址寄存器BX或变址寄存器SI、DI中
- 默认的段地址在DS段寄存器，可使用段超越前缀改变

MOV AX,[SI] ；AX←DS:[SI]



寄存器间接寻址方式



2000H BX

MOV AX, [BX]

播放

停止

寄存器间接寻址

存储器寻址方式

1、直接寻址方式

2、寄存器间接寻址方式

3、寄存器相对寻址方式

4、基址变址寻址方式

5、相对基址变址寻址方式

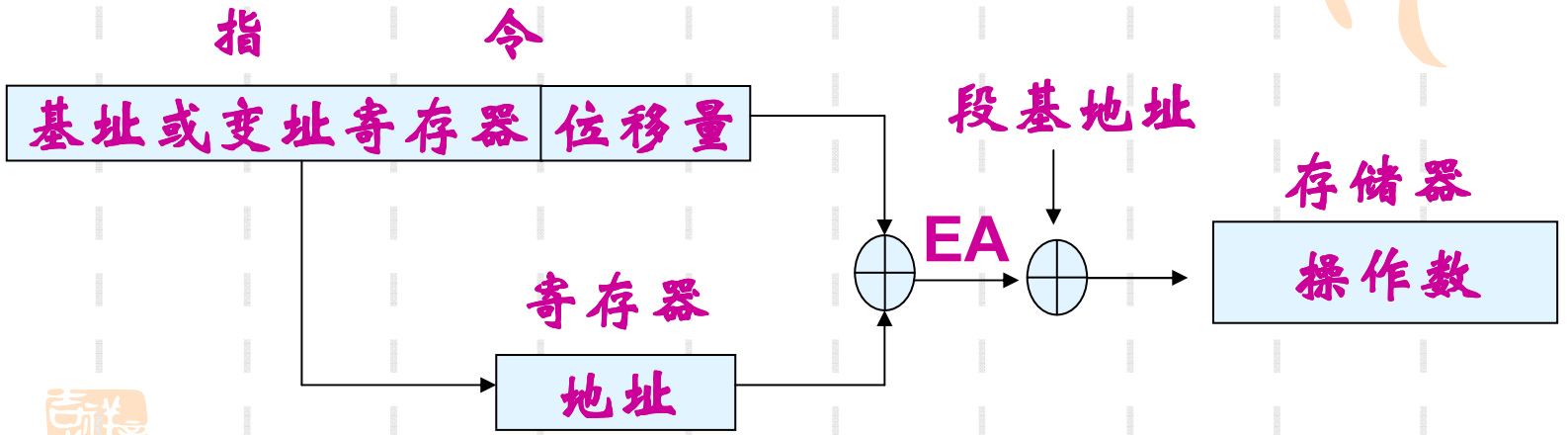
寄存器相对寻址方式

- 有效地址是寄存器内容与有符号8位或16位位移量之和，寄存器可以是BX、BP或SI、DI

  有效地址＝BX/BP/SI/DI＋8/16位位移量

- 段地址对应BX/SI/DI寄存器默认是DS，对应BP寄存器默认是SS；可用段超越前缀改变

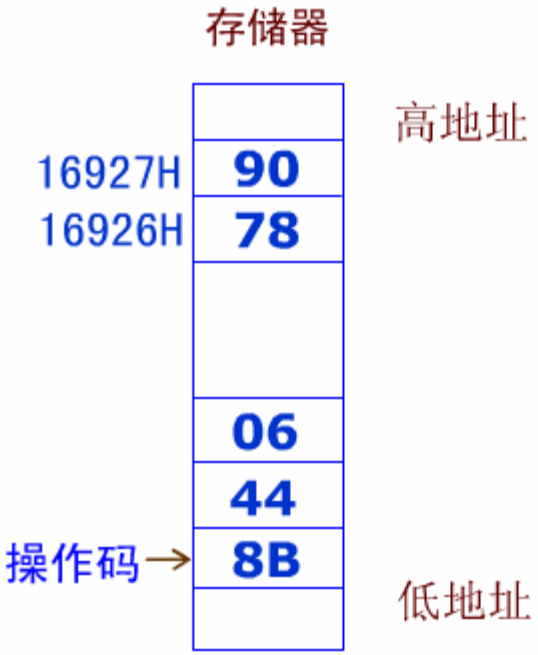
寄存器相对寻址指令



MOV AX,[DI+06H]　　；AX←DS:[DI+06H]

MOV AX,[BP+06H]　　；AX←SS:[BP+06H]

寄存器相对寻址方式



MOV AX, [SI+06H]

播放

停止

寄存器相对寻址

存储器寻址方式

1、直接寻址方式

2、寄存器间接寻址方式

3、寄存器相对寻址方式

4、基址变址寻址方式

5、相对基址变址寻址方式

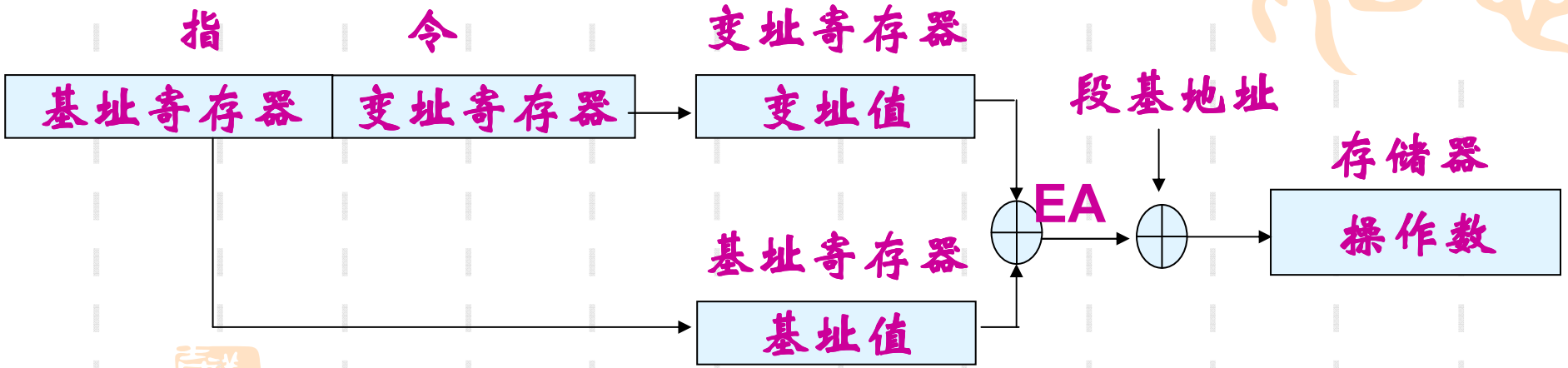
基址变址寻址方式

- 有效地址由基址寄存器（BX或BP）的内容加上变址寄存器（SI或DI）的内容构成：

  有效地址 = BX/BP + SI/DI

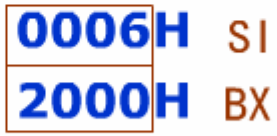
- 段地址对应BX基址寄存器默认是DS，对应BP基址寄存器默认是SS；可用段超越前缀改变

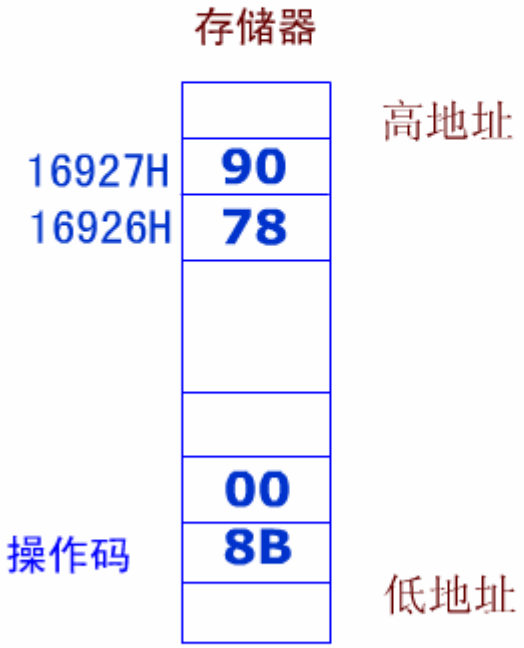
基址变址寻址指令




```
MOV AX,[BX+SI]      ; AX←DS:[BX+SI]
MOV AX,[BP+DI]      ; AX←SS:[BP+DI]
MOV AX,DS:[BP+DI]   ; AX←DS:[BP+DI]
```

基址变址寻址方式





MOV AX, [BX+SI]

播放

停止

存储器寻址方式

1、直接寻址方式

2、寄存器间接寻址方式

3、寄存器相对寻址方式

4、基址变址寻址方式

5、相对基址变址寻址方式

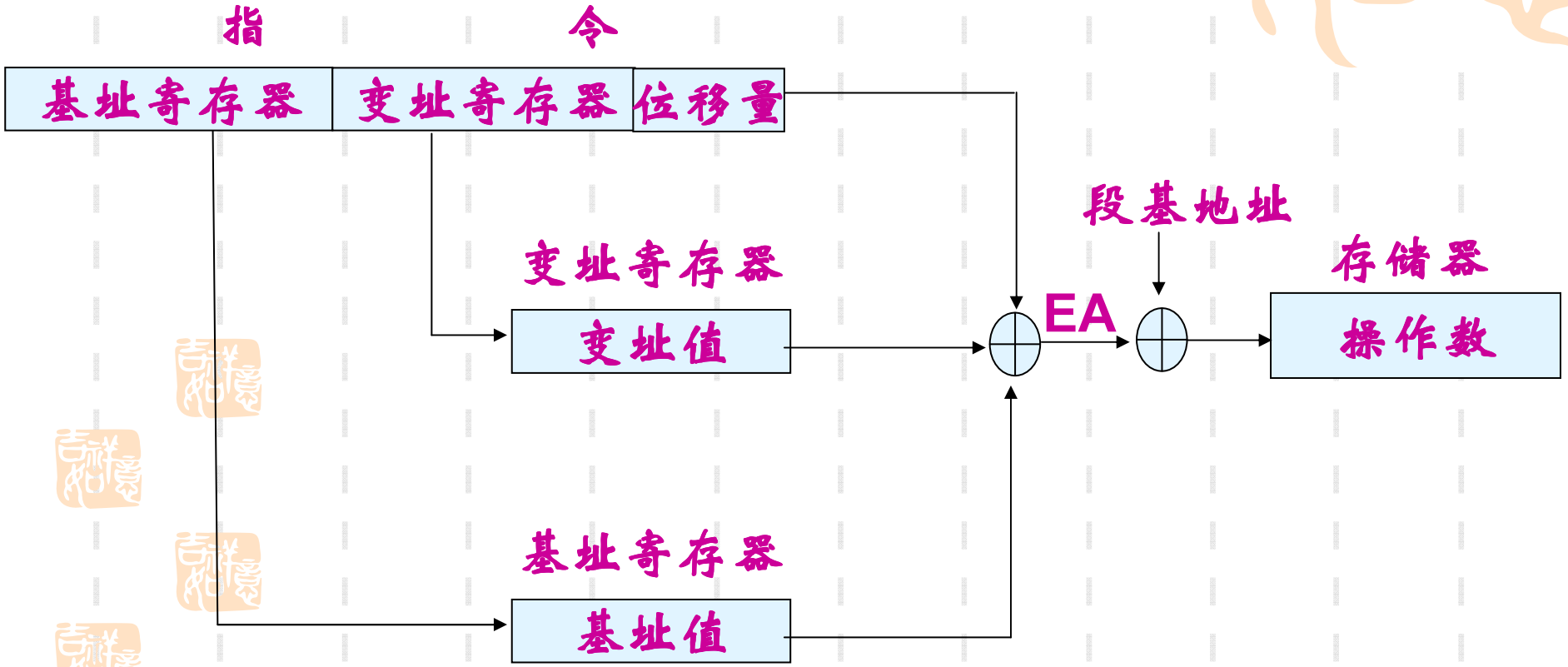
相对基址变址寻址方式

- 有效地址是基址寄存器（BX/BP）、变址寄存器（SI/DI）与一个8位或16位位移量之和：

  有效地址＝BX/BP＋SI/DI＋8/16位位移量

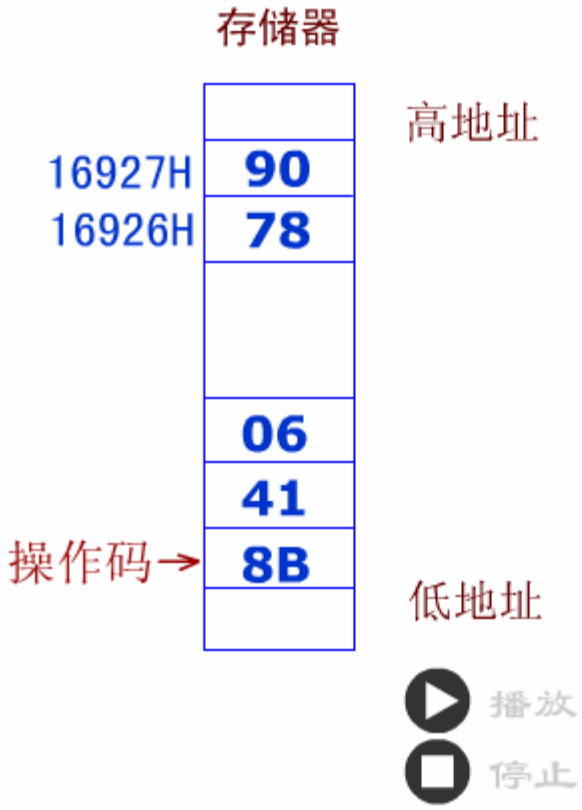
- 段地址对应BX基址寄存器默认是DS，对应BP基址寄存器默认是SS；可用段超越前缀改变

相对基址变址寻址指令



MOV AX,[BX+SI+06H]；AX←DS:[BX+SI+06H]

相对基址变址寻址方式



MOV AX, [BX+DI+06H]

相对基址变址寻址指令

- 位移量可用符号表示

- 同一寻址方式有多种表达形式

用符号表示位移量

- 在寄存器相对寻址或相对基址变址寻址方式中，位移量可用符号表示：

```
MOV AX,[SI+COUNT]
;COUNT是事先定义的变量或常量（就是数值）

MOV AX,[BX+SI+WNUM]
;WNUM也是变量或常量
```

多种表达形式

- 同一寻址方式可以写成不同的形式：

```
MOV AX,[BX][SI]
    ；等同于 MOV AX,[BX+SI]
MOV AX,COUNT[SI]
    ；等同于 MOV AX,[SI+COUNT]
MOV AX,WNUM[BX][SI]
    ；等同于 MOV AX,WNUM[BX+SI]
    ；等同于 MOV AX,[BX+SI+WNUM]
```

指令操作数的表达－寄存器

- r8——任意一个8位通用寄存器

  AH AL BH BL CH CL DH DL
- r16——任意一个16位通用寄存器

  AX BX CX DX SI DI BP SP
- reg——代表r8或r16
- seg——段寄存器 CS/DS/ES/SS

一定要熟悉噢！

指令操作数的表达－存储器

- m8——一个8位存储器操作数单元（所有主存寻址方式）
- m16——一个16位存储器操作数单元（所有主存寻址方式）
- mem——代表m8或m16

一定要熟悉噢！

指令操作数的表达－立即数

- i8——一个8位立即数
- i16——一个16位立即数
- imm——代表i8或i16
- dest——目的操作数
- src——源操作数

一定要熟悉噢！

80X86的寻址方式

- 与数据有关的寻址方式

- 与转移地址有关的寻址方式

与转移地址有关的寻址方式

通常情况下，程序顺序执行。若要改变指令顺序执行方式，就要给指令指出新的段地址(CS)和偏移地址(IP)，这就是与转移地址有关的寻址方式。

四种：段内直接寻址　　段内间接寻址

段间直接寻址　　段间间接寻址

段内直接寻址

段内直接转移的寻址方式。

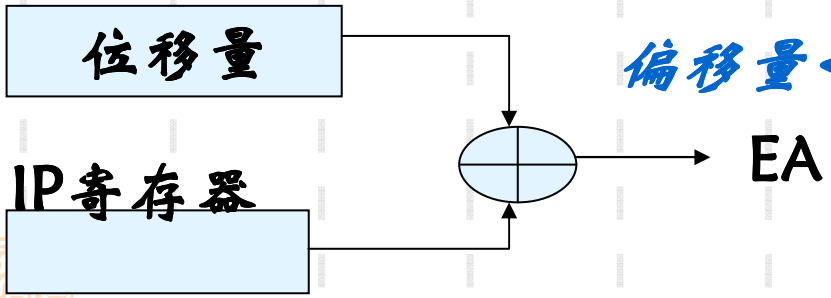
因为在同一段内，CS不变，只变IP。

指令中直接给出了转移地址的偏移量（8位或16位），该值与转移指令的下一条指令的首地址相加，即得IP的新值。

段内直接寻址

偏移量>0 向高地址方向转移

偏移量<0 向低地址方向转移



偏移量8位 短跳转,范围 -128~+127

JMP SHORT QUEST

偏移量16位 近跳转,范围 -32768~+32767

JMP NEAR PTR QUEST

与转移地址有关的寻址方式

段内直接寻址　　段内间接寻址

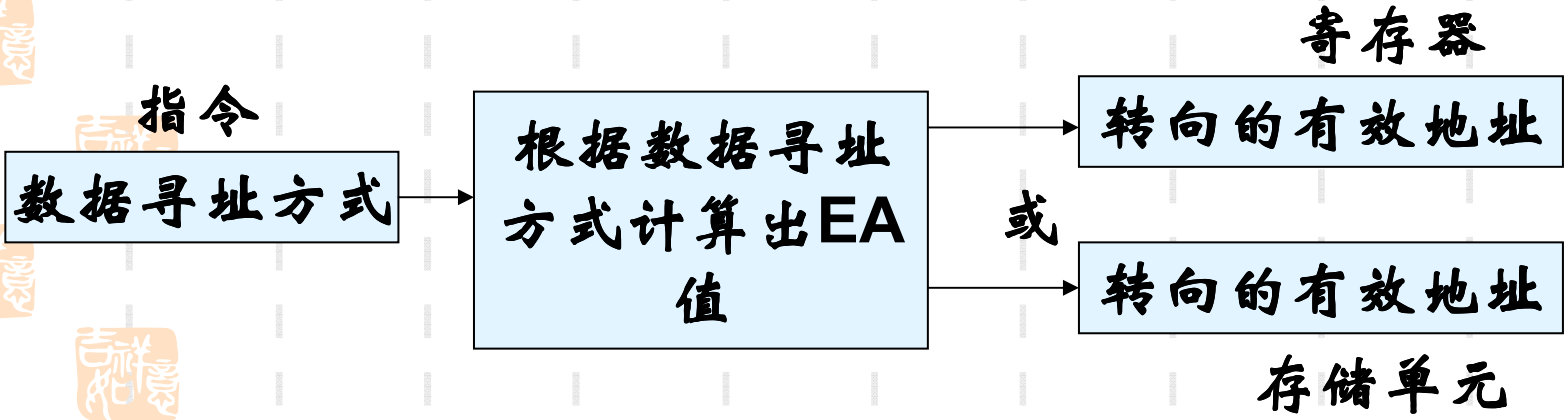
段间直接寻址　　段间间接寻址

段内间接寻址

段内间接转移的寻址方式。

转移的有效地址值在寄存器或存储器中。

指令中直接给出了寄存器名或给出访问存储器的各种寻址方式,以便在存储器中找到转移的有效地址。



段内间接寻址

JMP BX

JMP WORD PTR [BP+TABLE]

设(DS)=2000H, (BX)=1256H, (SI)=528FH

位移量=20A1H, (232F7H)=3280H, (264E5H)=2450H

JMP BX ; (IP)=1256H

JMP TABLE[BX] ; (IP)=3280H

JMP [BX][SI] ; (IP)=(264E5H)=2450H

与转移地址有关的寻址方式

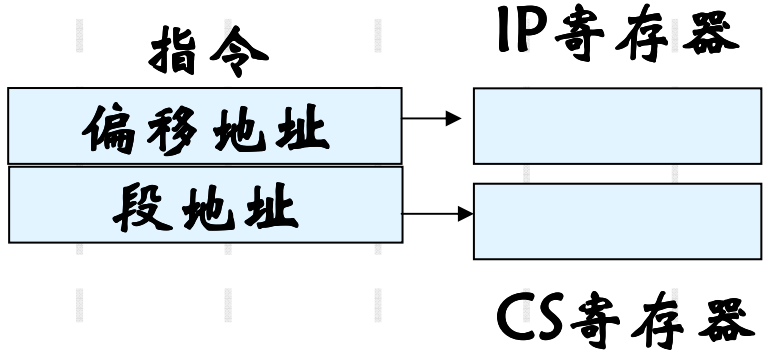
段内直接寻址　　段内间接寻址

段间直接寻址　　段间间接寻址

段间直接寻址

因为不在同一段内，CS改变，IP也变。

段间直接转移,指令中直接给出了新的段地址CS和偏移地址IP。偏移地址（IP）在低地址，段地址在高地址。



指令的汇编语言格式为：

JMP FAR PTR NEXTROUTINT

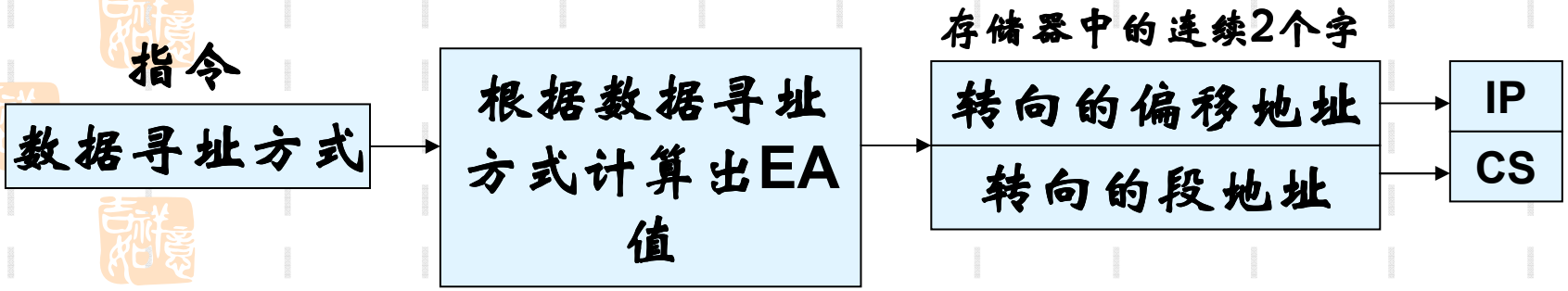
与转移地址有关的寻址方式

段内直接寻址　　段内间接寻址

段间直接寻址　　段间间接寻址

段间间接寻址

新的段地址(CS)和偏移地址(IP)在存储器的连续4个字节单元中。存储器的地址由指令中给出的各种寻址方式（立即寻址方式和寄存器寻址方式除外）求得。其段地址隐含为数据段。



JMP DWORD PTR [INTERS+BX]