第 3 章 8086的寻址方式和指令系统

第 3 章　8086的寻址方式和指令系统

- 3-1　8086汇编语言及寻址方式
- 3-2　数据传送类指令
- 3-3　算术运算指令
- 3-4　逻辑运算指令
- 3-5　串（数据块）处理指令
- 3-6　控制转移指令
- 3-7　处理机控制指令

学习目的

通过对本章的学习，应该能够达到下列要求：

- 了解8086的寻址方式
- 掌握8086的指令系统

学习目的

重点

- 8086寻址方式
- 数据传送类指令
- 算术运算指令
- 逻辑运算类指令
- 控制转移指令

难点

- 8086寻址方式

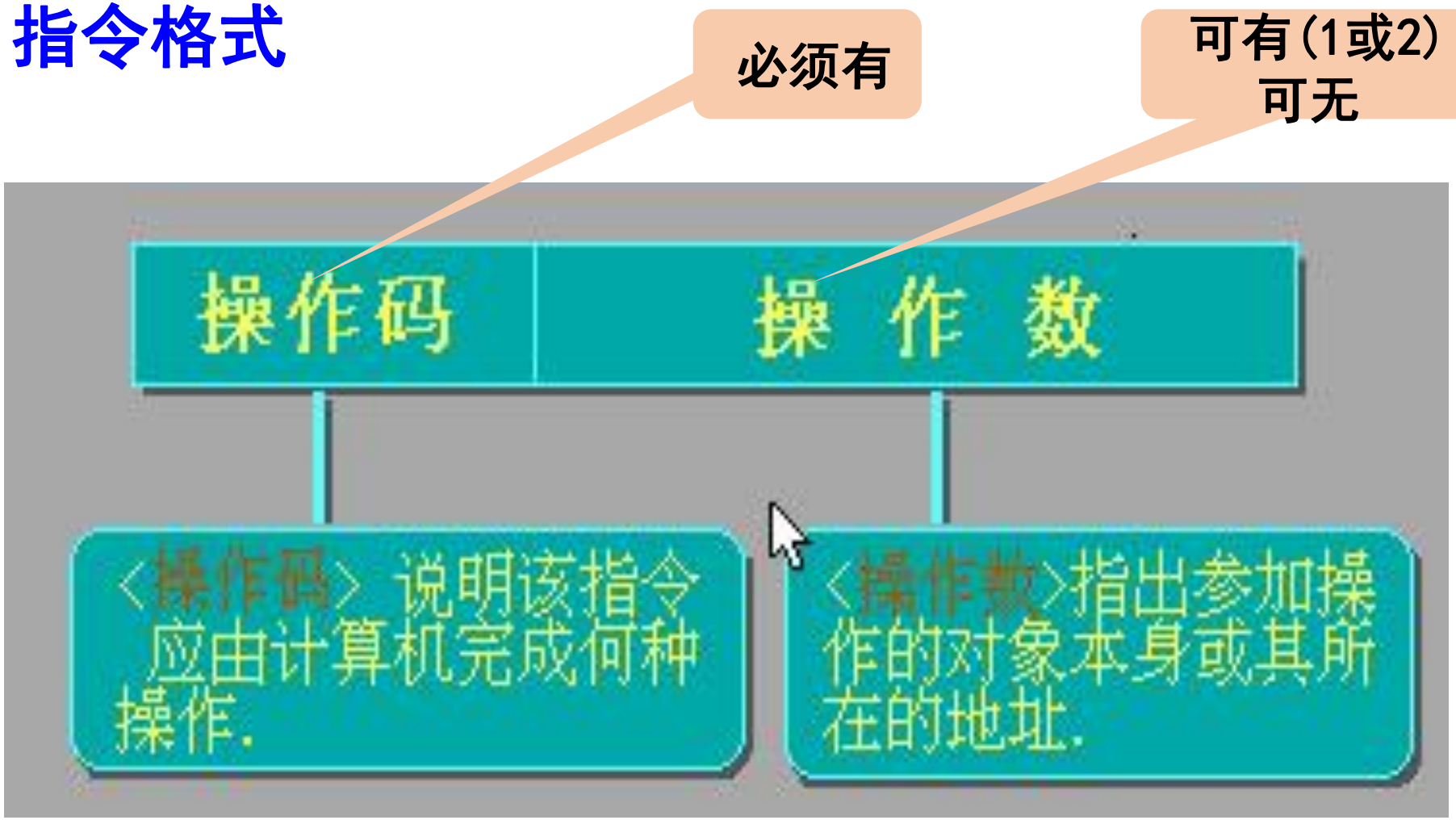
第 3 章　8086的寻址方式和指令系统

- 3-1　8086汇编语言及寻址方式
- 3-2　数据传送类指令
- 3-3　算术运算指令
- 3-4　逻辑运算指令
- 3-5　串（数据块）处理指令
- 3-6　控制转移指令
- 3-7　处理机控制指令

3-1 8086汇编语言及寻址方式

- 掌握8086基本指令的格式、功能及应用方法。

一、指令格式



3-1 8086汇编语言及寻址方式

1、操作码（指令助记符）

反映了指令的功能（或操作的性质），用英文缩写表示，如：

MOV 传送

CMP 比较

ADD 加法

OUT 输出

3-1 8086汇编语言及寻址方式

2、操作数（学习指令系统应特别注意）

指令操作的对象（所要处理的数据或数据的位置信息）。

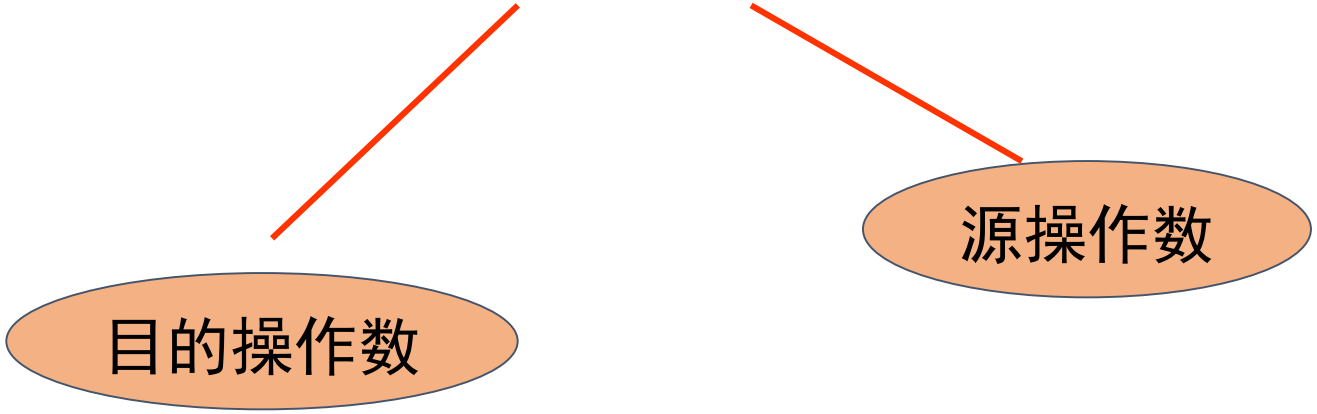
三种类型：立即数；　寄存器；　存储器（地址信息）

3-1 8086汇编语言及寻址方式

二、寻址方式

1、**立即寻址**——**操作数**由指令给出，**源操作数**为“立即数”。

如：　MOV　AX, 3064H　　；功能将立即数3064H送AX。



3-1 8086汇编语言及寻址方式

2、寄存器寻址 —— 操作数在指定的寄存器中

◇ 操作数存放在CPU的内部寄存器reg中：

- 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

例：MOV AX, BX；AX←BX

注意：* 源操作数和目的操作数的字长要一致。 MOV AH, BX

* CS不能用MOV指令改变 MOV CS, AX

3-1 8086汇编语言及寻址方式

3、存储器操作数的寻址方式

- 程序设计时，8086采用逻辑地址表示主存地址
  - 段地址在默认的或用段超越前缀指定的段寄存器中
  - 指令中只需给出操作数的偏移地址（有效地址EA）
- 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

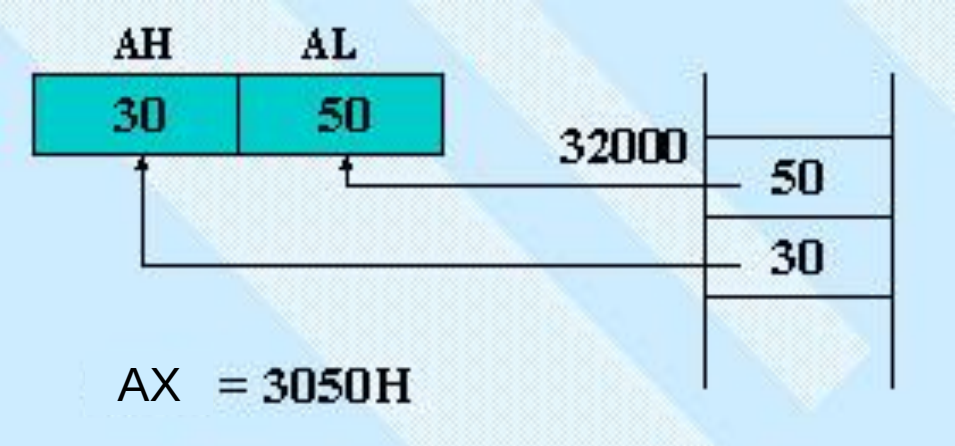
3-1 8086汇编语言及寻址方式

(1) **直接寻址**：**有效地址EA（操作数的偏移地址）由指令直接给出。**

物理地址PA = 16× (DS) + EA

例： MOV AX, DS: [2000H] 或 MOV AX, [2000H]

EA=2000H, 假设DS=3000H, 那么PA=32000H



低对低
高对高

注意：* 隐含的段为数据段 DS

*** 可使用段超越(加前缀)。 MOV AX, ES :[2000H]**

3-1 8086汇编语言及寻址方式

(2)寄存器间接寻址——EA只能在 BX、BP、SI、DI中

如：　MOV　AX，[BX]

若 **DS=3000H，BX=2000H**，则物理地址 **PA=32000H**

指令的功能为：　**AL** ← （32000H）

AH ← （32001H）

规定：　BX, SI, DI → 隐含的段寄存器 DS

BP → 隐含的段寄存器 SS

3-1 8086汇编语言及寻址方式

(2)寄存器间接寻址——EA只能在 BX、BP、SI、DI中

注意：

* 源操作数和目的操作数的字长一致

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

MOV     DX, [BX]      ;[BX]指示一个字单元[BX]->DL,  [BX+1]->DH
```

* 适于数组、字符串、表格的处理

3-1 8086汇编语言及寻址方式

（3）寄存器相对寻址

有效地址**EA** = { BX / BP / SI / DI } + { 8位 / 16位 } **位移量**

例： MOV AX, COUNT[SI] 或 MOV AX, [COUNT+SI]

假设**DS=3000H, SI=2000H, COUNT=3000H**

那么 **PA = 35000H**

指令功能：将**35000H**开始的两个单元内容送AX

3-1 8086汇编语言及寻址方式

（4）基址变址寻址

有效地址EA = (BX / BP) + (SI / DI)

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

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

* 必须是一个基址寄存器和一个变址寄存器的组合

~~MOV AX, [BX][BP]~~

~~MOV AX, [SI][DI]~~

3-1 8086汇编语言及寻址方式

（5）相对基址变址寻址

有效地址EA = (BX / BP) + (SI / DI) + (8位 / 16位) 位移量

```
MOV AX, [BX+DI+56]        ; AX←DS:[BX+DI+56]
```

↓

```
MOV AX, 56[BX+DI]
```

↓

```
MOV AX, 56[BX][DI]
```

3-1 8086汇编语言及寻址方式

4、固定寻址（隐含寻址）

指令中已经默认对微处理器中的某寄存器进行操作，不用在指令中指明使用的寄存器。

- **DAA**
- **LOOP　N1**

3-1 8086汇编语言及寻址方式

5、I/O端口操作数的寻址方式

- 直接端口寻址：端口地址由指令直接提供，是**8位立即数**，访问端口号00~FFH，共**256个**端口。

  例：**IN AL，63H**

- 间接端口寻址：寻址的端口号由**寄存器DX**提供，访问端口号0000~FFFFH，共**64K个**端口。

  例：**MOV DX，213H**

  IN AL，DX

寻 址 方 式 小 结

寻址方式	操作数地址(PA)	指令格式举例
立即寻址	操作数由指令给出	MOV DX, 100H ;DX←100H
寄存器寻址	操作数在寄存器中	ADD AX, BX ; AX←AX+BX
直接寻址	操作数的有效地址由指令直接给出	MOV AX, [2000H] ;AX←(2000H)(2001H)
寄存器间接寻址	仅允许 BX、BP、SI、DI 及其组合	
寄存器 相对寻址		
基址变址寻址		
相对基址 变址寻址		

寻 址 方 式 小 结

学习寻址方式时要注意：

(1) 正确书写各种寻址方式的汇编格式

(2) 清楚各种寻址方式所指定的操作数或操作数地址在何处?

(指令中 / 寄存器中 / 存储器单元中)

(3) 各种存储器寻址方式是如何形成操作数的**物理地址**的?

(段寄存器 / 基址、变址寄存器 / 偏移量的组合关系)

(4) 各种寻址方式限定使用的寄存器

第 3 章　8086的寻址方式和指令系统

- 3-1　8086汇编语言及寻址方式
- 3-2　数据传送类指令
- 3-3　算术运算指令
- 3-4　逻辑运算指令
- 3-5　串（数据块）处理指令
- 3-6　控制转移指令
- 3-7　处理机控制指令

第 3 章　8086的寻址方式和指令系统

8086指令系统的分类

8086微处理器指令系统中有133条指令，根据指令的操作性质可分为六大类。

8086指令系统

1、传送类指令
2、运算类指令
3、逻辑类指令
4、转移类指令
5、串操作指令
6、控制类指令

注意：1. 指令的基本功能

2. 指令的执行对标志位的影响

3. 对寻址方式或寄存器使用的限制和隐含使用的情况

第 3 章　8086的寻址方式和指令系统

- 3-1　8086汇编语言及寻址方式
- 3-2　数据传送类指令
- 3-3　算术运算指令
- 3-4　逻辑运算指令
- 3-5　串（数据块）处理指令
- 3-6　控制转移指令
- 3-7　处理机控制指令

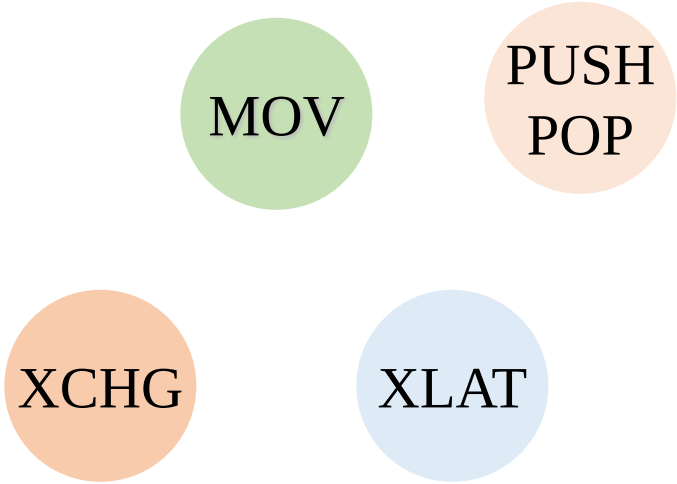
3-2 数据传送类指令

- 通用数据传送指令
- 地址传送指令
- 标志寄存器传送指令

3-2 数据传送类指令

一、通用数据传送指令

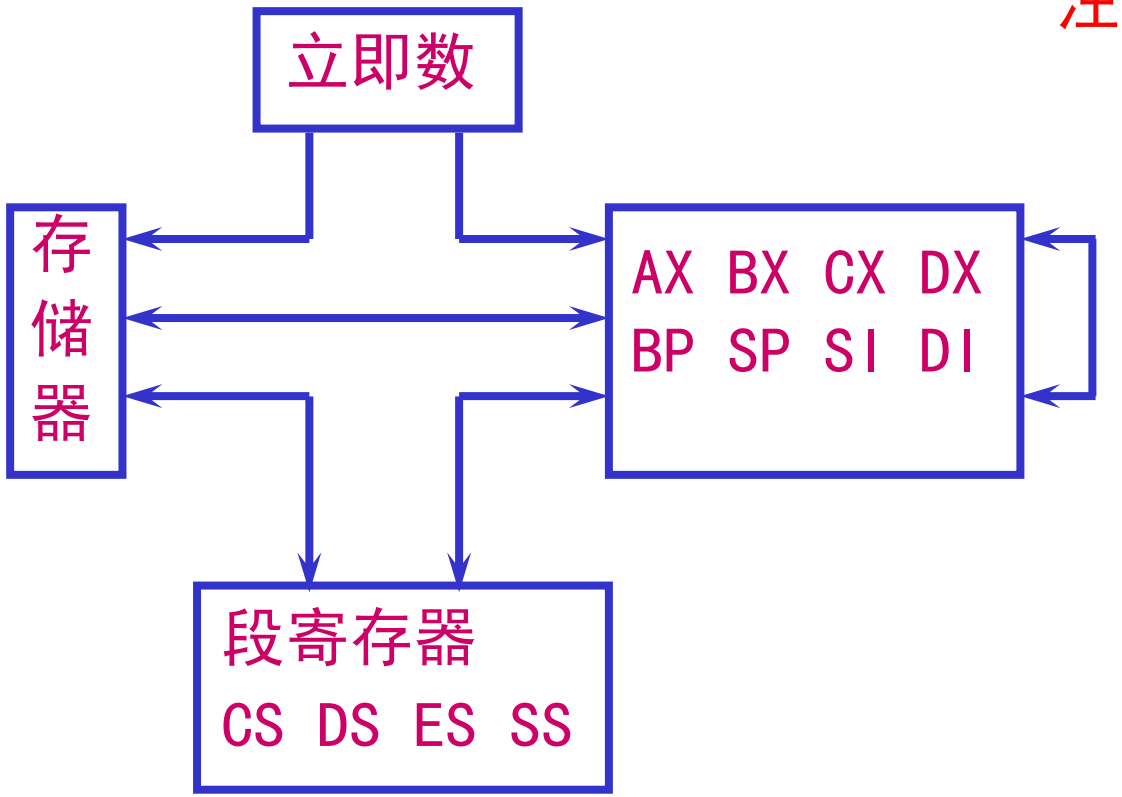
- 提供方便灵活的通用传送操作
- 有4种指令



3-2 数据传送类指令

1、传送指令：

MOV　　DST, SRC　　；　DST ← SRC



注意：
* DST不能是CS、立即数
* 不影响标志位
* DST、SRC不能同时为段寄存器

例： MOV DS, ES ×

* 立即数不能直接送段寄存器

例： MOV DS, 2000H ×

3-2 数据传送类指令

例：将AX、BX、SI寄存器清零， 将数FF11H送CX、DX寄存器。

法1:

```
MOV     AX，0000H
MOV     BX，0
MOV     SI，  0
MOV     CX，0FF11H
MOV     DX，0FF11H
```

送的数第一个为字母时，要用0隔开。

法2:

```
MOV     AX，0
MOV     BX，AX
MOV     SI，  AX
MOV     CX，0FF11H
MOV     DX，CX
```

3-2 数据传送类指令

当一个操作数的操作类型无法确定时，需要利用汇编语言的操作符显式指明。

如：　　MOV　　[BX+SI], 54H　　；非法指令，

修正：

MOV　　BYTE PTR [BX+SI], 54H ；　BYTE　　PTR 说明是字节操作

MOV　　WORD PTR [BX+SI], 54H；WORD　　PTR　说明是字操作

3-2 数据传送类指令

例：将50H、51H分别存入存储器21000H、21006H单元

法1:

```
MOV     AX，2000H
MOV     DS，AX
MOV     SI，1000H
MOV     BYTE PTR[SI]，50H
MOV     BYTE PTR[SI+6]，51H
```

法2:

```
MOV     AX，2000H
MOV     DS，AX
MOV     BL，50H
MOV     BH，51H
MOV     DS: [1000H]，BL
MOV     DS: [1006H]，BH
```

3-2 数据传送类指令

例： 交换**BUF1**和**BUF2**两单元的内容。

```
MOV     SI，OFFSET BUF1
MOV     DI，OFFSET BUF2
MOV     AL，[SI]
MOV     AH，[DI]
MOV     [SI]，AH
MOV     [DI]，AL
```

说明BUF1和BUF2为地址的偏移量

3-2 数据传送类指令

MOV指令并非任意传送！非法指令的主要现象：

- **两个操作数的类型不一致**

  MOV CX,AL

- **无法确定是字节量还是字量操作**

  MOV [1004],05H

正确: MOV SI, 1004H

MOV AL, 5

MOV [SI], AL

3-2 数据传送类指令

MOV指令并非任意传送!非法指令的主要现象：

- **两个操作数都是存储器**

  MOV [2000H], [3000H]

- **段寄存器的操作有一些限制**

  MOV DS, 2000H 不允许立即数传送给段寄存器

  MOV DS, ES 不允许段寄存器之间的直接数据传送

  MOV CS, [SI] 不允许直接改变CS值

3-2 数据传送类指令

2、栈传送指令

◇堆栈是一个“**先进后出**”的主存区域，位于堆栈段中；

SS——段地址

◇堆栈指针**永远 →栈顶**

◇堆栈只有两种基本操作：　**进栈 PUSH**

出栈 POP

3-2 数据传送类指令

进栈指令：　**PUSH　SRC**

例：PUSH　AX　　；(SP-1) ← AH

；(SP-2) ← AL, SP ← SP -2

出栈指令：　**POP　DST**

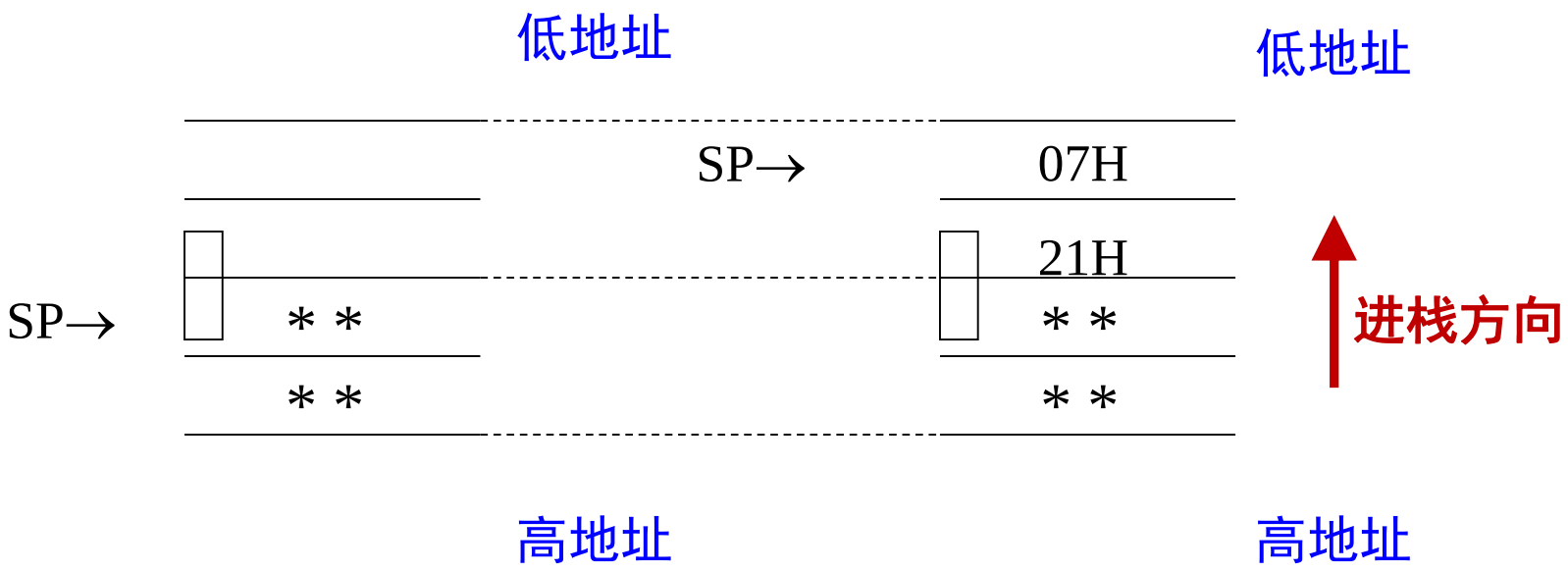
例：POP　BX　　；BL ←(SP)，BH ← (SP+1)

；SP ← SP + 2

3-2 数据传送类指令

例： 假设 AX = 2107 H，执行 PUSH AX

注意：高对高，低对低



3-2 数据传送类指令

堆栈操作的特点

- 堆栈操作必须以**字**为单位。　　PUSH　AL ×
- 字量数据从栈顶压入和弹出时，都是**低地址字节送低字节，高地址字节送高字节**。（**高对高，低对低**）
- 堆栈操作遵循**先进后出**原则。
- 不能用**立即寻址方式**。　　PUSH　2600H ×
- 不影响标志位。
- 出栈指令DST**不能是CS** 。　　POP　CS ×

3-2 数据传送类指令

- 堆栈段是程序中不可缺少的一个内存区，常用来：
  - 临时存放数据
  - 传递参数
  - 保存和恢复寄存器

3-2 数据传送类指令

```
例：    PUSH  AX        ；保护现场
        PUSH  BX
                ...     ;此期间用到AX和BX寄存器
                ...
        POP  BX         ；恢复现场
        POP  AX
        RET
```

3-2 数据传送类指令

3、交换指令： XCHG OPR1, OPR2 ;OPR1 ↔ OPR2

- 允许寄存器与寄存器之间交换数据
- 允许寄存器与存储器之间交换数据

注意:

*** 不影响标志位**

*** 不允许在存储器与存储器之间交换数据**

*** 不允许使用段寄存器**

例：XCHG BX, [BP+SI]

XCHG [BX], [BP+SI] ×

XCHG AL, BH

XCHG BX, DS ×

3-2 数据传送类指令

4、 换码指令(将一个字节从一种代码换成另一种代码)

```
XLAT                ; AL←DS:[BX+AL]
```

- 将BX指定的缓冲区中、AL指定的位移处的**一个字节**数据取出赋给AL。
- **换码指令执行前：**
  * 造表
  * 表格**首地址**---->BX
  * 相对表格首地址的**位移量**----> AL
- **换码指令执行后：**

  AL中的值已转换为内存表格中的某一值。

3-2 数据传送类指令

例：试用XLAT指令求3的平方。

1：构造表

2：设置BX和AL的初始值

3：使用XLAT指令

```
DATA SEGMENT
    PF DB 0，1，4，9，16，25，36，49，64，81
DATA ENDS
CODE SEGMENT
……
MOV BX,OFFSET PF  或 LEA BX,PF
MOV AL,03H
XLAT
……
RET
```

3-2 数据传送类指令

二、目标地址传送指令

1、有效地址送寄存器指令：

例： MOV SI, 0

LEA AX, [SI+2728H] ； AX ← SI+2728H 有效地址，不是存储值

LEA BX, [BP+SI] ； BX ← BP+SI

3-2 数据传送类指令

二、目标地址传送指令

2、地址指针送寄存器和DS指令：

注意：取的是存储值

例： LDS DI, [SI+2728H]； DI ← (SI+2728H 和 SI+2728H+1)

；DS←(SI+ 2728H+2和SI+ 2728H+3）

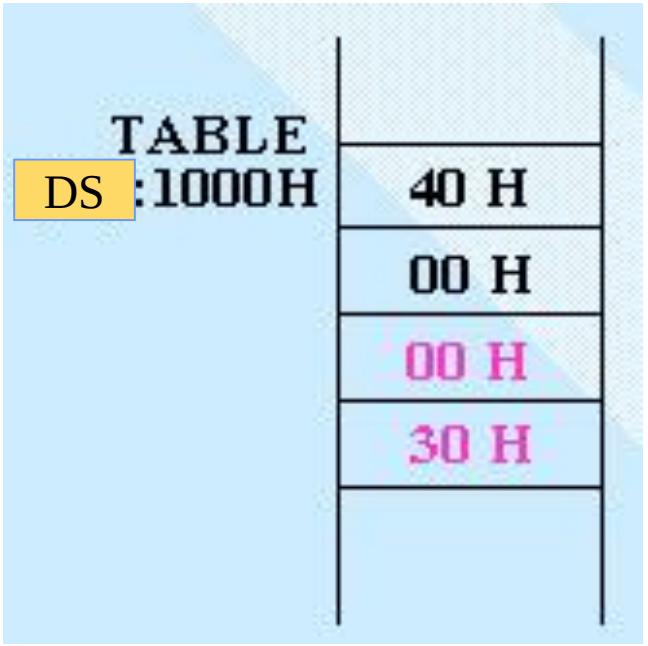
3、指针送寄存器和ES指令：

注意：取的是存储值

例：LES DI, [SI+2728H]；即：4个相继字节 → 寄存器DI、ES

3-2 数据传送类指令

例：




```
MOV     BX, OFFSET TABLE    ; BX=1000H
LEA     BX, TABLE           ; BX=1000H
MOV     SI, 0
LEA     BX,  [SI+1000H]     ;BX=1000H

LDS     BX, [SI+1000H]      ; BX=0040H
                            ; DS=3000H
LES     BX, [SI+1000H]      ; BX=0040H
                            ; ES=3000H
```

注意：
* 不影响标志位
* 目的操作数不能是段寄存器
* 源操作数必须为存储器寻址方式

3-2 数据传送类指令

三、标志寄存器传送指令

标志送AH指令：　　LAHF ；　AH←标志寄存器的低字节

AH送标志寄存器指令：　SAHF ；　PSW的低字节 ← AH

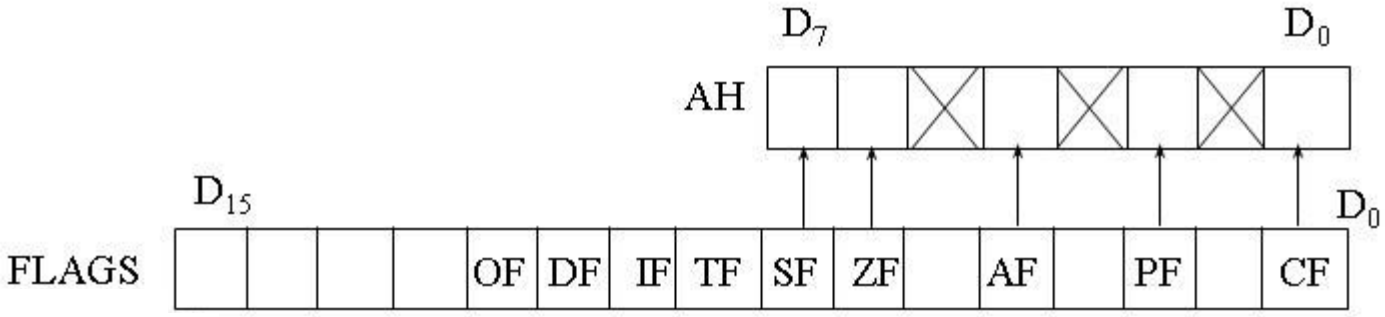
标志进栈指令：　　PUSHF ；　PSW入栈

标志出栈指令：　　POPF ；　PSW出栈

* SAHF和 POPF影响标志位

* LAHF和PUSHF不影响标志位。

传送类指令不影响标志位
（除了往PSW送数的指令）



第 3 章　8086的寻址方式和指令系统

- 3-1　8086汇编语言及寻址方式
- 3-2　数据传送类指令
- 3-3　算术运算指令
- 3-4　逻辑运算指令
- 3-5　串（数据块）处理指令
- 3-6　控制转移指令
- 3-7　处理机控制指令

3-3 算数运算指令

- 加法指令
- 减法指令
- 乘法指令
- 除法指令
- 十进制调整指令

3-3 算数运算指令

一、加法指令

不带进位加法指令：ADD DST, SRC ； DST ← SRC + DST

带进位加法指令：ADC DST, SRC ； DST ← SRC + DST + CF

加1指令：INC OPR ； OPR ← OPR + 1

二、减法指令

不带借位的减法指令：SUB DST, SRC ； DST ← DST - SRC

带借位的减法指令：SBB DST, SRC ； DST ← DST - SRC - CF

减1指令：DEC OPR ； OPR ← OPR - 1

求补指令：NEG OPR ； OPR ← 0 - OPR

比较指令：CMP OPR1, OPR2 ； OPR1 - OPR2

3-3 算数运算指令

例： 无符号双字加法和减法

8234 7856H＋1234 8998H－8000 4491H＝?

```
MOV   AX,   7856H      ;  AX＝7856H
MOV   DX,   8234H      ;  DX＝8234H
ADD   AX,   8998H      ;  AX＝01EEH，CF＝1
ADC   DX,   1234H      ;  DX＝9469H，CF＝0
SUB   AX,   4491H      ;  AX＝BD5DH，CF＝1
SBB   DX,   8000H      ;  DX＝1468H，CF＝0
```

8234 7856H＋1234 8998H－8000 4491H ＝**1468 BD5DH**

3-3 算数运算指令

比较指令CMP（compare）

```
CMP     AX，BX
CMP     AL，100
```

- 做减法运算：**DST－SRC**
- **CMP**指令将目的操作数减去源操作数，**但差值不回送目的操作数**
- 比较指令通过减法运算**影响状态标志**，用于比较两个操作数的大小关系

3-3 算数运算指令

增量和减量指令

```
INC     SI                  ;  SI←SI＋1
DEC     BYTE PTR [SI]       ;  [SI]←[SI]－1
```

- INC指令和DEC指令是**单操作数指令**
- 与加法和减法指令实现的加1和减1不同的是：**INC和DEC不影响CF标志。**

例：

```
ADD     AL, 1
INC     AL
```

3-3 算数运算指令

求补指令NEG（negtive）

- NEG指令对操作数执行求补运算，即用零减去操作数，然后结果返回操作数。
- 求补运算也可以表达成：将操作数**按位取反后加1。**
- NEG指令对标志的影响与用零作减法的SUB指令一样。

```
MOV    AL,  64H
NEG    AL          ;   AL＝0－64H＝9CH
                   ;   CF＝1  SF＝1
                   ;   ZF＝0   PF＝1
```

3-3 算数运算指令

例：将22000H和23000H开始的4个字节相加，和送存24000H开始的单元。（高位对应高地址，低位对应低地址）

```
MOV    AX，2000H     ；置段地址
MOV    DS，AX
MOV    ES，AX
MOV    SI，2000H     ；置被加数首址
MOV    DI，3000H     ；置加数首址
MOV    BX，4000H     ；置和首址
MOV    AX，[SI]      ；取被加数低16位
ADD    AX，[DI]      ；低16位部分和
MOV    [BX]，AX      ；存低16位部分和
```

```
MOV    AX，[SI+2]；高16位计算
ADC    AX，[DI+2]
MOV    [BX+2]，AX
MOV    AX，0         ；取高位进位
ADC    AX，0
MOV    [BX+4]，AX
```

3-3 算数运算指令

三、 乘法指令

无符号数乘法指令：**MUL　　SRC**

；字节操作数　AX ← AL × SRC

；字操作数　　DX, AX ← AX × SRC

带符号数乘法指令：**IMUL　　SRC**

；字节操作数、字操作数同上。

注意： * AL(AX)为隐含的乘数寄存器。

* AX(DX,AX)为隐含的乘积寄存器。

* **SRC不能为立即数**。

3-3 算数运算指令

例： AX = 16A5H，BX = 0611H

(1) MUL BX ; DX, AX ← AX ×BX

```
        1 6 A 5 H
  ×     0 6 1 1 H
-------------------
        1 6 A 5
    1 6 A 5
+ 8 7 D E
-------------------
0 0 8 9 5 E F 5 H
```

3-3 算数运算指令

四、除法指令

无符号数除法指令：　　DIV　　SRC

带符号数除法指令：　　IDIV　　SRC

执行操作：　**字节操作**　AL ← AX / SRC 的商

AH ← AX / SRC 的余数

字操作　AX ← DX, AX / SRC的商

DX ← DX, AX / SRC的余数

注意:　*除数必须为被除数的一半字长。

*做无符号数除法，扩展很简单，清零。而做带符号数除法，则需用扩展指令CBW或CWD　。

***SRC不能为立即数。**

*对所有条件标志位均**无意义**。

3-3 算数运算指令

★符号扩展指令：

CBW ；AL → AX

执行操作：若AL的最高有效位为0，则AH= 00H

若AL的最高有效位为1，则AH= FFH

CWD ；AX → DX, AX

执行操作：若AX的最高有效位为0，则DX= 0000H

若AX的最高有效位为1，则DX= FFFFH

例：AX=BA45H

```
CBW          ; AX=0045H
CWD          ; DX=FFFFH ， AX=BA45H
```

3-3 算数运算指令

五、十进制调整指令

压缩的BCD码：用4位二进制数表示1位十进制数

例：$59 = (0101\ 1001)_{BCD}$

非压缩的BCD码：用8位二进制数表示1位十进制数。（即低4位二进制数表示1位十进制数，高4位二进制数为0）

例：$59 = (0000\ 0101 \quad 0000\ 1001)_{BCD}$

压缩BCD码加法的调整指令：**DAA**

压缩BCD码减法的调整指令：**DAS**

非压缩BCD码加、减、乘、除法调整指令

AAA　AAS　AAM　AAD

3-3 算数运算指令

五、十进制调整指令

压缩BCD码**加法**的调整指令：DAA

如果AL的**低4位大于9或AF=1**，则AL的内容加06H，并将AF置1；然后如果AL的**高4位大于9或CF=1**，则AL的内容加60H，且将CF置1

压缩BCD码**减法**的调整指令：DAS

非压缩BCD码加、减、乘、除法调整指令

AAA　AAS　AAM　AAD

3-3 算数运算指令

注意:

* 隐含的操作寄存器为AL
* 紧接在加减指令之后使用
* 影响状态标志位（对OF无意义）

```
(1) MOV    AL, 34H        ;AL=34H
    ADD    AL, 89H        ;AL=34H+89H=BDH
    DAA                   ; AL=BDH+60H+06H=23H  AF=CF=1
(2) MOV    AL, 34H        ;AL=34H
    SUB    AL, 89H        ;AL=34H-89H=ABH
    DAS                   ;AL=ABH-60H-06H=45H  AF=CF=1
```

第 3 章　8086的寻址方式和指令系统

- 3-1　8086汇编语言及寻址方式
- 3-2　数据传送类指令
- 3-3　算术运算指令
- 3-4　逻辑运算指令
- 3-5　串（数据块）处理指令
- 3-6　控制转移指令
- 3-7　处理机控制指令

3-4 逻辑运算指令

- 逻辑运算指令
- 循环移位指令

3-4 逻辑运算指令

一、逻辑运算指令

逻辑非指令：NOT OPR ；OPR ← OPR取反

逻辑与指令：AND DST, SRC ； DST ← DST ∧ SRC

逻辑或指令：OR DST, SRC ； DST ← DST ∨ SRC

异或指令： XOR DST, SRC ； DST ← DST ∀ SRC

测试指令： TEST OPR1, OPR2；OPR1 ∧ OPR2

（测试是否为空）TEST AX, AX

注意： * AND、OR、XOR、TEST对标志位产生下列影响

CF	OF	SF	ZF	PF	AF
0	0	.	.	.	无定义

* NOT 不影响标志位。

3-4 逻辑运算指令

例：

```
AND    BL,11110110B    ;BL中D0和D3清0，其余位不变
OR     BL,00001001B    ;BL中D0和D3置1，其余位不变
XOR    BL,00001001B    ;BL中D0和D3求反，其余位不变
```

- AND指令可用于某些位的清位（同0相与），不影响其他位
- OR指令可用于某些位的置位 （同1相或），不影响其他位
- XOR指令可用于某些位的求反（同1相异或），不影响其他位

3-4 逻辑运算指令

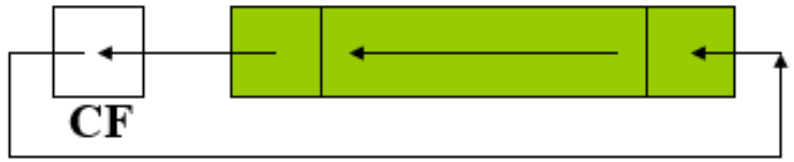
二、循环移位指令

循环左移 **ROL** OPR, CNT

循环右移 **ROR** OPR, CNT

小循环

带进位循环左移 **RCL** OPR, CNT



带进位循环右移 **RCR** OPR, CNT

大循环

3-4 逻辑运算指令

二、循环移位指令

循环左移 **ROL OPR, CNT**

```
mov al, 40h      ; AL = 01000000b
rol al, 1        ; AL = 10000000b, CF = 0
rol al, 1        ; AL = 00000001b, CF = 1
rol al, 1        ; AL = 00000010b, CF = 0
```

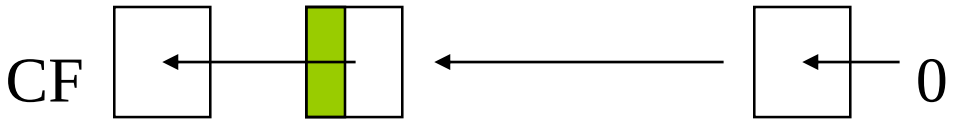
带进位循环左移 **RCL OPR, CNT**

```
clc              ; CF = 0
mov bl, 88h      ; CF, BL = 0 10001000b
rcl bl, 1        ; CF, BL = 1 00010000b
rcl b1, 1        ; CF, BL = 0 00100001b
```

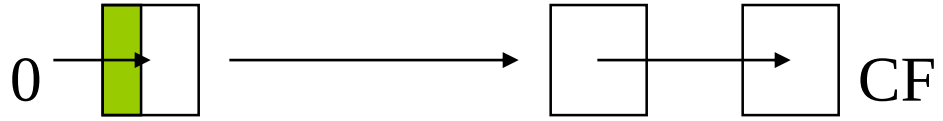
3-4 逻辑运算指令

三、移位指令

逻辑左移 **SHL OPR, CNT**

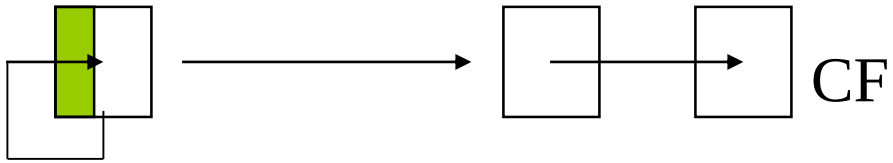


逻辑右移 **SHR OPR, CNT**



算术左移 **SAL OPR, CNT**（**同逻辑左移**）

算术右移 **SAR OPR, CNT** (**最高位填充符号位。正数填充0，负数填充1**)



3-4 逻辑运算指令

例： 将DX中的非压缩BCD码合并为压缩BCD,存DL。

（如：DX=0507H，结果DL=57H)

```
MOV     CL, 4
SHL     DH, CL              ；低4位移到高4位 DH=50H
ADD     DL, DH              ；合并到DL=57H
```

例：AX= 0012H，BX= 0034H，把它们装配成AX= 1234H

```
MOV  CL, 8
ROL  AX, CL          ；AX=1200H
ADD  AX, BX
```

3-4 逻辑运算指令

注意:

* OPR可用立即数以外的任何寻址方式

* CNT=1，SHL　OPR, 1

CNT>1，MOV　CL, CNT

SHL　OPR,　CL　　; 以SHL为例

* 状态标志位：

CF = 移入的数值

移位指令：　　SF、ZF、PF 根据移位结果设置， AF无定义

循环移位指令：　不影响 SF、ZF、PF、AF

3-4 逻辑运算指令

例： 将AL寄存器中的无符号数乘以10

- 逻辑左移一位相当于无符号数乘以2
- 逻辑右移一位相当于无符号数除以2

```
XOR   AH,AH        ; 实现AH＝0，同时使CF＝0
                     AL--→AX
SHL   AX,1         ; AX←2×AX
MOV   BX,AX        ; BX←AX＝2×AX
SHL   AX,1         ; AX←4×AX
SHL   AX,1         ; AX←8×AX
ADD   AX,BX        ; AX←8×AX＋2×AX＝10×AX
```

第 3 章　8086的寻址方式和指令系统

- 3-1　8086汇编语言及寻址方式
- 3-2　数据传送类指令
- 3-3　算术运算指令
- 3-4　逻辑运算指令
- 3-5　串（数据块）处理指令
- 3-6　控制转移指令
- 3-7　处理机控制指令

3-5 串（数据块）处理指令

- 串传送指令
- 存入串指令
- 从串取指令
- 串比较指令
- 串扫描指令

3-5 串（数据块）处理指令

配合使用的前缀有：

REP　　重复

REPE/REPZ　　相等/为零则重复

REPNE/REPNZ　　不相等/不为零则重复

3-5 串（数据块）处理指令

一、 串传送指令MOVS

MOVSB （字节）

MOVSW （字）

执行操作：(1) (DI) ← (SI)

(2) 字节操作：SI←SI±1，DI←DI±1

字操作： SI←SI±2，DI←DI±2

方向标志： DF=0时用 +，DF=1时用 - 。

3-5 串（数据块）处理指令

如：REP　　MOVSB　　;将数据段中的整串数据传送到附加段中。

源串（数据段）➔ 目的串（附加段）

注意：执行REP　　MOVS之前，应先做好：

（1）源串首地址（末地址）→ SI

（2）目的串首地址（末地址）→ DI

（3）串长度 → CX

（4）建立方向标志（CLD使DF=0，STD使DF=1）

•源串必须在*数据段DS*中，目的串必须在*附加段ES*中。

• 不影响条件标志位

3-5 串（数据块）处理指令

例：将DS段2000H开始的数据串（100个字节）传送到ES段3000H开始的单元。

```
MOV    SI, 2000H
MOV    DI, 3000H
MOV    CX, 100
CLD                    ; 增址传送
MOVSB
MOVSB
MOVSB
.
MOVSB
MOVSB
```

100 条= REP MOVSB

3-5 串（数据块）处理指令

二、存入串指令 STOS

STOSB ； [DI]←AL，DI←DI±1

STOSW ； [DI+1]、[DI] ←AX，DI←DI±2

* 目的串必须在附加段中，

* 不影响条件标志位

3-5 串（数据块）处理指令

二、存入串指令 STOS

例：把附加段中2000H开始的100个字节缓冲区清零。

```
MOV     DI, 2000H
MOV     AL, 0
MOV     CX, 100          ;  MOV  CX, 50
CLD
REP     STOSB            ;  REP  STOSW
```

3-5 串（数据块）处理指令

三、从串取指令LODS

LODSB ；AL←[SI]，SI←SI±1

LODSW ；AX←[SI]、[SI+1]，SI←SI±2

注意:

* LODS指令一般不与REP联用
* 不影响状态标志位

3-5 串（数据块）处理指令

四、串比较指令CMPS

CMPSB （字节）/ **CMPSW** （字）

执行操作： (1) [SI] - [DI]

根据比较结果设置条件标志位：相等 ZF=1

不等 ZF=0

(2) 字节操作：SI←SI±1，DI←DI±1

字操作： SI←SI±2，DI←DI±2

由DF确定“+”或“-”。

(3)可配合使用的前缀: **REPZ （REPNZ）**

3-5 串（数据块）处理指令

五、串扫描指令SCAS

SCASB （字节）/ SCASW （字）

执行操作： 字节操作：AL - [DI]，DI←DI±1

字操作： AX - [DI][DI+1]，DI←DI±2

可配合使用的前缀: REPZ （REPNZ）

执行操作： (1) 如CX=0或ZF=0 (ZF=1)则退出，否则转(2)

(2) CX←CX-1

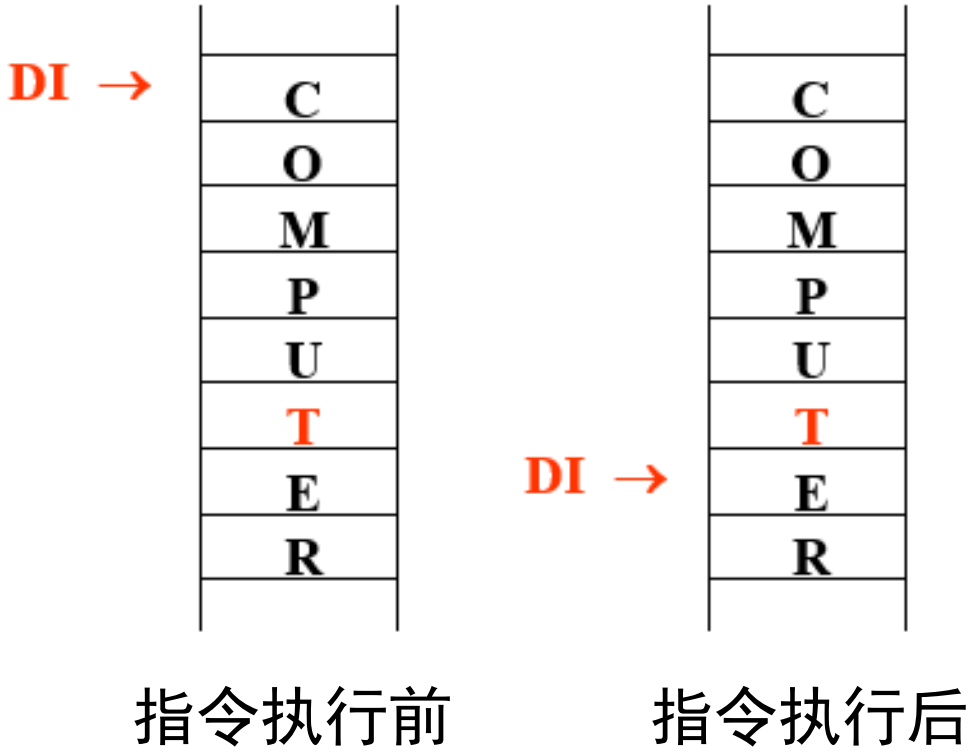
(3) 执行CMPS / SCAS

(4) 重复(1) ~ (3)

3-5 串（数据块）处理指令

例： 从一个字符串中查找一个指定的字符。

```
MESS    DB  'COMPUTER'
LEA     DI, [MESS]
MOV     AL, 'T'
MOV     CX, 8
CLD
REPNE   SCASB
```



指令执行前　　指令执行后

DI：相匹配字符的下一个地址
CX：剩下还未比较的字符个数

第 3 章　8086的寻址方式和指令系统

- 3-1　8086汇编语言及寻址方式
- 3-2　数据传送类指令
- 3-3　算术运算指令
- 3-4　逻辑运算指令
- 3-5　串（数据块）处理指令
- 3-6　控制转移指令
- 3-7　处理机控制指令

3-6 控制转移指令

- 无条件转移指令
- 条件转移指令
- 循环指令
- 子程序调用和返回指令
- 中断指令

3-6 控制转移指令

一、无条件转移指令 JMP OPR

例：

```
        ...
        ...
        JMP    HELLO
        ...
        ...
HELLO:  MOV   AL, 3
```

二、条件转移指令

1、根据单个条件标志的设置情况转移：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				O	D	I	T	S	Z		A		P		C

3-6 控制转移指令

格式		测试条件	
JZ(JE)	OPR	ZF=1	
JNZ(JNE)	OPR	ZF=0	
JS	OPR	SF=1	
JNS	OPR	SF=0	
JO	OPR	OF=1	溢出
JNO	OPR	OF=0	
JP	OPR	PF=1	位字节1的个数为偶数
JNP	OPR	PF=0	
JC	OPR	CF=1	
JNC	OPR	CF=0	

3-6 控制转移指令

2、比较两个无符号数，并根据比较结果转移：

	格式		测试条件
低于	JB(JNAE,JC)	OPR	CF=1
高于等于	JNB(JAE,JNC)	OPR	CF=0
低于等于	JBE(JNA)	OPR	CF∨ZF=1
高于	JNBE(JA)	OPR	CF∨ZF=0

3-6 控制转移指令

3、比较两个带符号数，并根据比较结果转移：

	格式		测试条件
小于<	JL(JNGE)	OPR	SF∀OF=1
大于等于≥	JNL(JGE)	OPR	SF∀OF=0
小于等于≤	JLE(JNG)	OPR	(SF∀OF)∨ZF=1
大于>	JNLE(JG)	OPR	(SF∀OF)∨ZF=0

4、测试CX的值为0则转移：

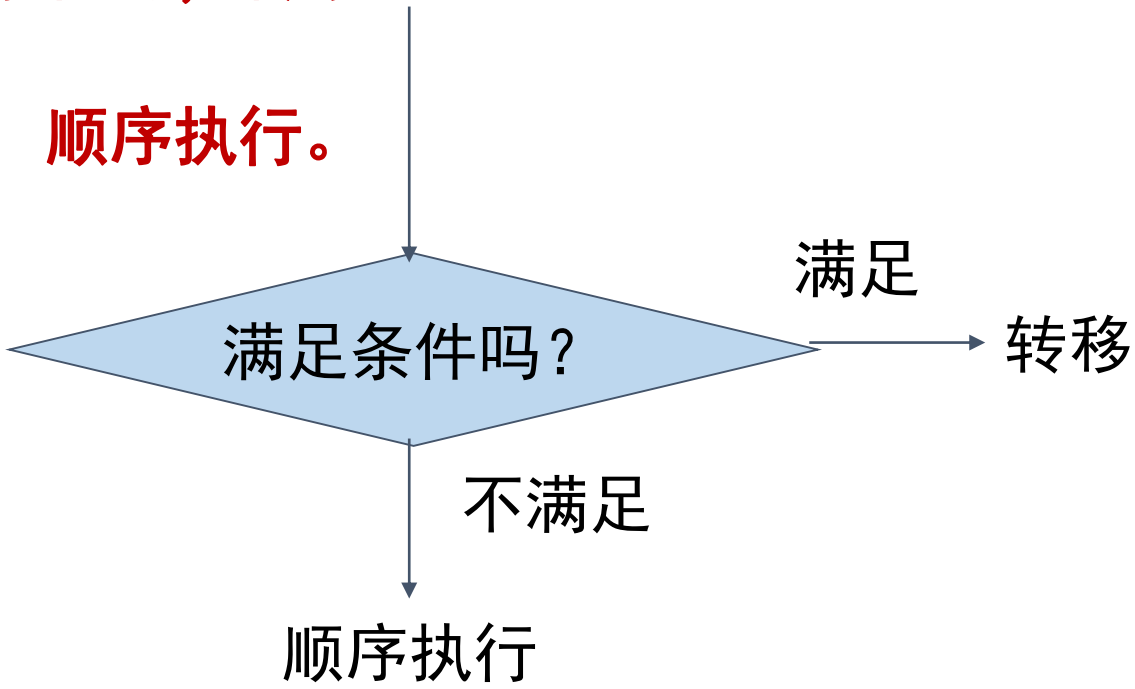
格式		测试条件
JCXZ	OPR	CX=0

3-6 控制转移指令

注意：条件转移指令共同的特点是：

满足条件时，转移。

否则，顺序执行。



3-6 控制转移指令

例： 求两无符号数中较大值，存wmax。

```
        cmp ax,  bx    ；比较AX和BX
        jae  next      ；若AX≥BX，转移next
        x  ng ax,  bx  ；若AX＜BX，交换
next:       wmax,  ax
```

如果AX和BX存放的是有符号数，
则条件转移指令应采用JGE指令

3-6 控制转移指令

三、循环指令

LOOP

LOOPZ / LOOPE

LOOPNZ / LOOPNE

执行步骤：(1) CX←CX -1

(2) 检查是否满足测试条件，如满足则

IP←IP+8位位移量，实行循环；

不满足则IP不变，退出循环。

注意：* CX中存放循环次数

* 只能使用段内直接寻址的**8位位移量**（-128~127）

3-6 控制转移指令

例： 数据块传送

```
    MOV CX, 400H              ；设置循环次数400H
    MOV SI, OFFSET  sbuf      ；SI指向数据段源缓冲区开始
    MOV DI, OFFSET dbuf       ；DI指向附加段目的缓冲区开始
AG:
    MOV AL, [SI]              ；循环主体（实现数据传送）
    MOV ES:[DI], AL           ；每次传送一个字节
    INC SI                    ；SI和DI指向下一个单元
    INC DI
    LOOP AG                   ；循环次数CX减1，不为0继续循环
    ……
```

3-6 控制转移指令

四、调用指令

调用指令：　CALL　　DST

五、RET返回指令　　RET

六、中断指令：　　INT　　n

从中断返回指令：　IRET

注意：
* n= (0 ~ 255) 是中断类型号。
* INT(INTO)指令执行完，把IF和TF置0，但不影响其它标志位。
* IRET指令执行完，标志位由堆栈中取出的值确定。

第 3 章　8086的寻址方式和指令系统

- 3-1　8086汇编语言及寻址方式
- 3-2　数据传送类指令
- 3-3　算术运算指令
- 3-4　逻辑运算指令
- 3-5　串（数据块）处理指令
- 3-6　控制转移指令
- 3-7　处理机控制指令

3-7 处理机控制指令

标志处理指令：

CLC　CF←0

CMC　CF←CF

STC　CF←1

CLD　DF←0

STD　DF←1

CLI　IF←0

STI　IF←1

注意：　* 只影响本指令指定的标志

其它指令：

NOP　无操作（机器码占一个字节）

HLT　暂停机（等待一次外中断，之后继续执行程序）