5.1 通用机器指令概述

(1) 数据传送指令
(2) 算术运算指令
(3) 逻辑运算指令
(4) 移位指令
(5) 位操作和字节操作指令
(6) 标志位控制指令
(7) I/O 指令
(8) 控制转移指令
(9) 串操作指令
(10) 杂项指令

5.1 通用机器指令概述

华中科技大学

一般情况下，指令的共同要求：

(1) 双操作数的操作数类型必须匹配。

```
MOV  AX , BH
SUB   BL，  CX
MOV  SI , CL
ADD  BUF , AX   ；其中 BUF   DB  1, 2
MOV  ds:[2000] ,2
```

5.1 通用机器指令概述

一般情况下，指令的共同要求：

(1) 双操作数的操作数类型必须匹配。

(2) 目的操作数一定不能是立即操作数

CMP AX，2

功能是：根据 (AX) –2 的结果设置标志位

~~CMP 2，AX~~

5.1 通用机器指令概述

一般情况下，指令的共同要求：

(1) 双操作数的操作数类型必须匹配。

(2) 目的操作数一定不能是立即操作数。

(3) **目的操作数和源操作数不能同时为存储器操作数。** 如果一个操作数在数据存储单元中，另一个一定要是立即数和寄存器操作数。

~~MOV BUF1, BUF2~~
~~ADD WORD PTR [ESI], [EDI]~~
~~SUB BUF1, [ESI]~~

5.2 数据传送指令

华中科技大学

1、一般数据传送指令

MOV、MOVSX、MOVZX、XCHG、XLAT

2、堆栈操作指令

PUSH、POP、PUSHA、PUSHAD、POPA、POPAD

3、标志寄存器传送指令

PUSHF、POPF、PUSHFD、POPFD、LAHF、SAHF

4、地址传送指令

LEA、LDS、LES、LSS

除了SAHF、POPF,POPFD外，其他不影响标志位。

5.2.1 一般数据传送指令


```
MOV      OPD, OPS          ; 数据传送
MOVSX    R16/R32, OPS      ; 符号扩展传送
MOVZX    R16/R32, OPS      ; 0（无符号）扩展传送
XCHG     OPD, OPS          ; 一般数据交换
XLAT                       ; 查表转换
```

MOV EAX, BX——错误的指令

```
MOVSX EAX,  BX
MOVZX EAX,  BX
```

5.2.1 一般数据传送指令

华中科技大学

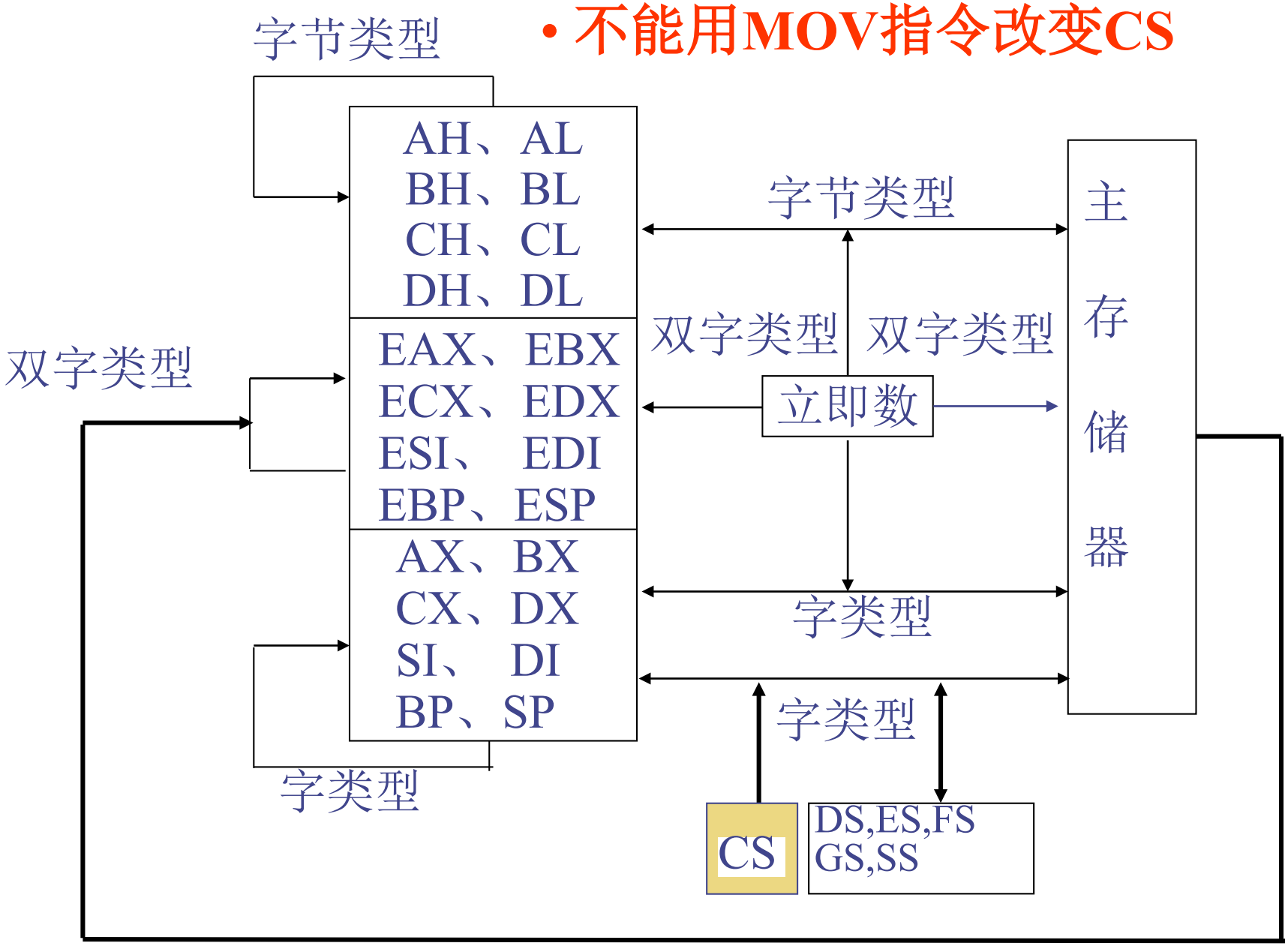
1、MOV指令

语句格式：MOV OPD, OPS

功　　能：(OPS) → OPD

MOV 指令所允许的数据传送路径及类型,见下图。

- 立即数不能直接送段寄存器
- 不能用MOV指令改变CS



5.2.1 一般数据传送指令

2、一般数据交换指令

语句格式：XCHG　OPD，OPS

功　　能：（OPD）→ OPS　（OPS）→ OPD

将源、目的地址指明的单元中的内容互换。

例3：　XCHG AH, AL

执行前：（AX）=1234H　执行后：（AX）= 3412H

Question: 指令XCHG　BUF1，BUF2 是否正确？

5.2.1 一般数据传送指令

3、查表转换指令

语句格式：XLAT

功　　能：([EBX+AL]) → AL

将 (EBX)为首址，(AL)为位移量的字节存储单元中的数据传送给AL。

5.2.1 一般数据传送指令

设有一个16进制数码（0～9，A~F) 在(AL)中，现请将该数码转换为对应的ASCII。

一般的算法：判断(AL)是否小于等于9，

若是：则将(AL)+30H → AL;

否则：　将(AL)+37H → AL;

```
MYTAB   DB '0123456789ABCDEF'
MOV     EBX, OFFSET  MYTAB
MOV     AL，10
XLAT   ；(AL)='A'
```

XLAT 可用来对文本数据进行编码和译码，从而实现简单的加密和解密。

5.2.1 堆栈操作指令

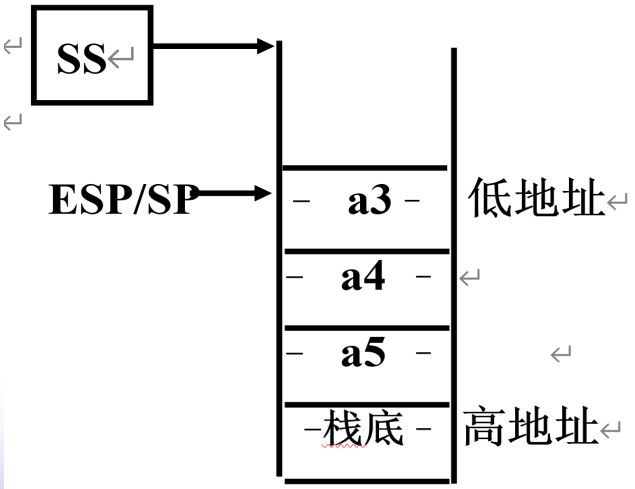
堆栈：是在内存中开辟的一片存贮区，一端固定(栈底)，一端活动(栈顶)，只允许在一端插入或删除数据。

数据存取原则:先进后出，以字/双字为单位。

地址变化情况：存入数据时从高地址向低地址扩展。

堆栈中的数据也称元素或栈项。

元素进栈称压入，出栈称弹出。



5.2.3 堆栈操作指令

华中科技大学

1、进栈指令： PUSH

语句格式：PUSH OPS

功能 ：将立即数、寄存器、段寄存器、存储器中的一个字/双字数据压入堆栈中。

例：PUSH AX

PUSH EAX

PUSH X

PUSH DWORD PTR [EBX]

PUSH 1234H

PUSH AL

5.2.3 堆栈操作指令

1、进栈指令：PUSH OPS

- 字数据入栈
  - ① (ESP) - 2 → ESP
  - ② 字数据 → [ESP]
- 双字数据入栈：
  - ① (ESP) - 4 → ESP
  - ② 双字数据 → [ESP]

记为：(OPS) → ↓ (ESP)

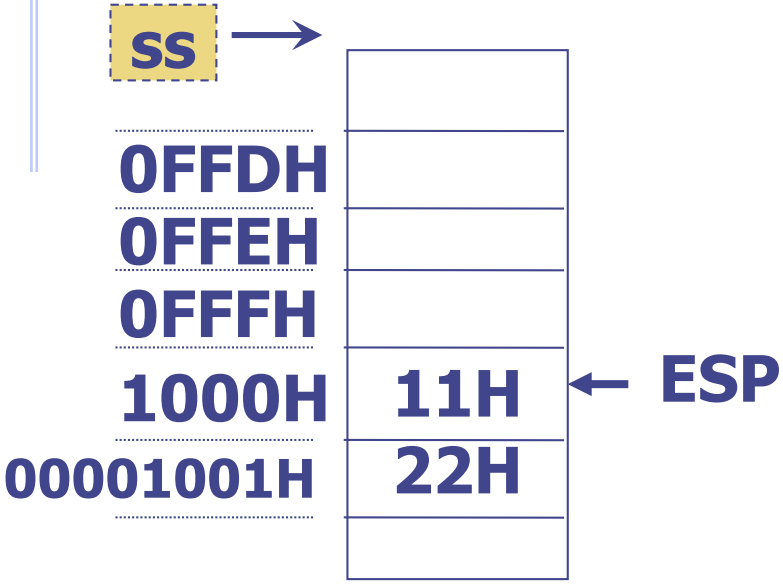
ESP —2、—4 取决于是字还是双字入栈

5.2.3 堆栈操作指令

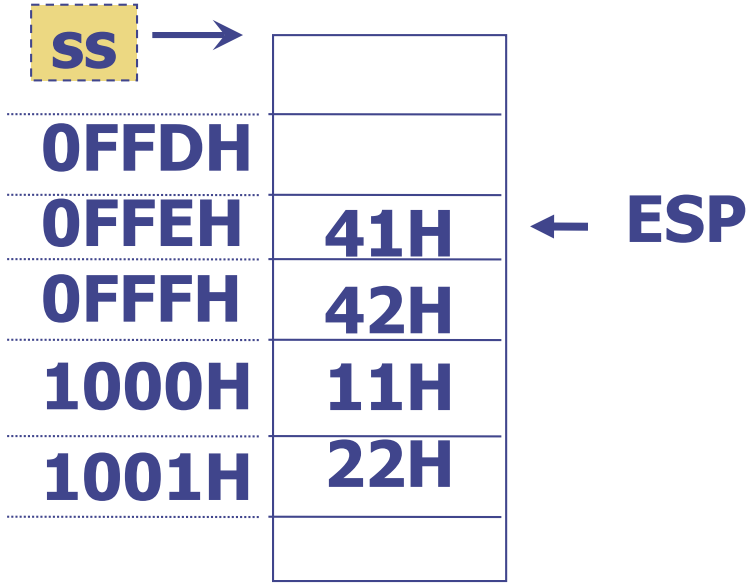
堆栈示意图画法中应注意的问题

PUSH AX

执行前：(AX)=4241H (ESP)=00001000H



指令执行堆栈状态



执行“PUSH AX”后的堆栈状态

ESP 00001000H

ESP 00000FFEH

5.2.3 堆栈操作指令

2、出栈指令： POP

语句格式：POP OPD

功　　能：将栈顶元素弹出送至某一寄存器、段寄存器（CS除外）、存储器中。

字数据出栈

① ([ESP]) → OPD

② (ESP)+2 → ESP

记为：↑(ESP) → OPD

双字数据出栈类似，(ESP)+4->ESP

5.2.5 地址传送指令

1、传送偏移地址指令

语句格式：LEA R32，OPS

功　　能：将OPS的偏移地址送入OPD中。

说明：

- R32 是一个32位的通用寄存器；
- OPS 所提供的一定是一个存储器地址；

5.2.5 地址传送指令

1、传送偏移地址指令

```
MOV   ESI, OFFSET  NUM
LEA   ESI, NUM      与上一行语句等效；

LEA  EDI, [ESI+4]        (ESI)+4 -> EDI

MOV  EDI，  [ESI+4]
            DS: ([ESI]+4) -> EDI
```

如何用一条指令实现(EAX)+ (EBX)*8 -> ECX ？

```
LEA  ECX，  [EAX+EBX*8]
```

5.3 算术运算指令

一般对标志位都有影响

1、加法指令

INC、ADD、ADC

2、减法指令

DEC、NEG、SUB、SBB、CMP

3、乘法指令

IMUL、MUL

4、除法指令

IDIV、DIV

5、符号扩展指令

CBW、CWD、CWDE、CDQ

5.3.2 减法指令

DEC OPD

NEG OPD

SUB OPD, OPS

SBB OPD, OPS

CMP OPD, OPS

DEC对OF,SF,ZF
其它指令对
CF,OF,SF,ZF有影响;

5.3.3 乘法指令

(1) 有符号乘法

■ 双操作数的有符号乘指令

语句格式：IMUL OPD, OPS

功　　能：(OPD) * (OPS) → **OPD**

说　　明：**OPD** 为 **16/32**位寄存器

OPS 为同类型的寄存器、存储器操作数或立即数。

例：**IMUL AX, BX**

IMUL EAX, DWORD PTR[ESI]

IMUL AX, 3

5.3.3 乘法指令

(1) 有符号乘法

■ 3个操作数的有符号乘指令

语句格式：**IMUL OPD, OPS，n**

功　　能：**(OPS) * n → OPD**

说　　明：**OPD 为 16/32位寄存器**

OPS为同类型的寄存器、存储器

操作数或立即数。

例：**IMUL AX, BX, -10**

IMUL EAX, DWORD PTR[ESI],5

IMUL BX, AX, 3

5.3.3 乘法指令

(1) 有符号乘法

■单操作数的有符号乘法

语句格式： IMUL OPS

字节乘法： (AL)*(OPS) → AX

字 乘 法： (AX)*(OPS) → DX , AX

双字乘法： (EAX) *(OPS) →EDX,EAX

说明：OPS不能是立即数 IMUL 100是错误指令

如果乘积的高位不是低位的符号扩展，而是包含有效位，则CF=1, OF=1.

5.3.3 乘法指令

(2) 无符号乘法

语句格式： MUL OPS

功　　能：

字节乘法：　(AL)*(OPS) → AX

字乘法：　　(AX)*(OPS) → DX , AX

双字乘法：　（EAX) *(OPS) →EDX,EAX

说　　明：OPS不能是立即数

如果乘积的高位不是低位的符号扩展，而是包含有效位，则CF=1, OF=1.

5.3.3 乘法指令

华中科技大学

无符号乘法与有符号乘法的比较

```
.code
begin:
    mov  al,10H
    mov  bl, -2      ;  (bl)=FE
    imul  bl
```

(ax)=0FFE0H,结果高字节无有效位，有OF=0,CF=0

```
    mov  al,10H
    mul   bl
```

(ax)=0FE0H, 结果高字节有有效位，有OF=1, CF=1

```
    mov  al,-10h
    mov  bl,2
    imul  bl
```

(ax) = 0FFE0H

```
mov  al,-10h
mov  bl,2
mul   bl
```

(al)=0F0H

(ax) = 01E0H

```
end begin
```

5.3.4 除法指令

(1) 有符号除法

IDIV OPS

字节除法： (AX)/(OPS) → AL(商), AH(余)

字 除 法： (DX,AX)/(OPS) → AX (商), DX (余)

双字除法： (EDX,EAX)/(OPS) → EAX (商), EDX

(2) 无符号除法

DIV OPS

字节除法：(AX)/(OPS) → AL(商),AH(余)

字 除 法：(DX,AX)/(OPS) → AX (商), DX (余)

双字除法：(EDX,EAX)/(OPS) → EAX (商), EDX

5.3.5 符号扩展指令

(1) 将字节转换成字

CBW

将AL中的符号扩展至AH中。

(2) 将字转换成双字

CWD

将AX中的符号扩展至DX中。

5.3.5 符号扩展指令

(3) 将AX中的有符号数扩展为32位送EAX

```
CWDE
```

(4) 将EAX中的有符号数扩展为64位数

```
送  EDX, EAX
CDQ
```

5.4 逻辑运算指令

华中科技大学

求反

NOT OPD ; (OPD)求反→OPD

逻辑乘

AND OPD, OPS ; (OPD)^(OPS) →OPD

测试指令

TEST OPD, OPS ; (OPD)^(OPS)

逻辑加

OR OPD, OPS ; (OPD)V(OPS) →OPD

按位加

XOR OPD, OPS ; (OPD)异或(OPS) →OPD

5.5 移位指令

(1)算术左移 SAL　Shift Arithmetic Left
(2)逻辑左移 SHL　SHift Logical Left
(3)逻辑右移 SHR　SHift Logical Right
(4)算术右移 SAR　Shift Arithmetic Right
(5)循环左移 ROL　Rotate Left
(6)循环右移 ROR　Rotate Right
(7)带进位的循环左移 RCL
Rotate left through Carry
(8)带进位的循环右移 RCR

5.5 移位指令

语句格式：

操作符　OPD，n 或 CL

功能：将(OPD)中的所有位按操作符规定的方式移动，结果存在OPD对应的单元中。

说明：

OPD可以是寄存器，也可以是地址表达式。

5.5 移位指令

(1) 算术左移　SAL OPD，n

(2) 逻辑左移　SHL OPD，n

(OPD)向左移动n位，低位补0

SAL　AX, 3　等价于

SAL　AX,1

SAL　AX,1

SAL　AX,1　；CF为执行最后一次移位时送入的值

5.5 移位指令

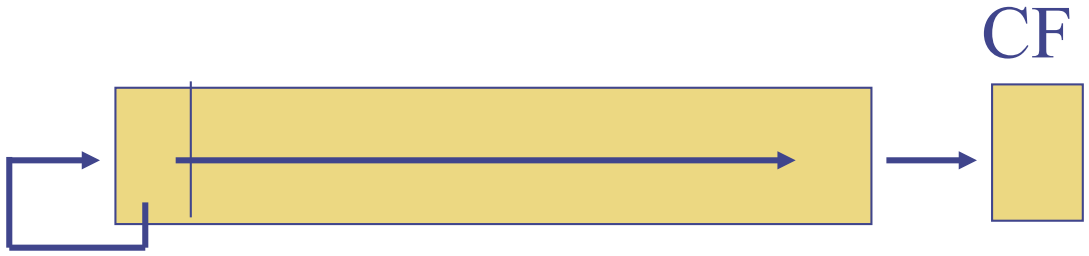
(3)逻辑右移　SHR　OPD，n

(OPD) 向右移动n位，高位补0


```
MOV    AH, 5;      5→  0000 0101
SHR    AH, 1
       ; (AH)= ?          2
       ; (CF)= ?          1
```

(4)算术右移 SAR OPD, n

(OPD)向右移动n位，**最高位不变**。




```
MOV     AH, 0F5H
SHR     AH, 1
    ; (AH)= ?     7AH
    ; (CF)= ?     1
```

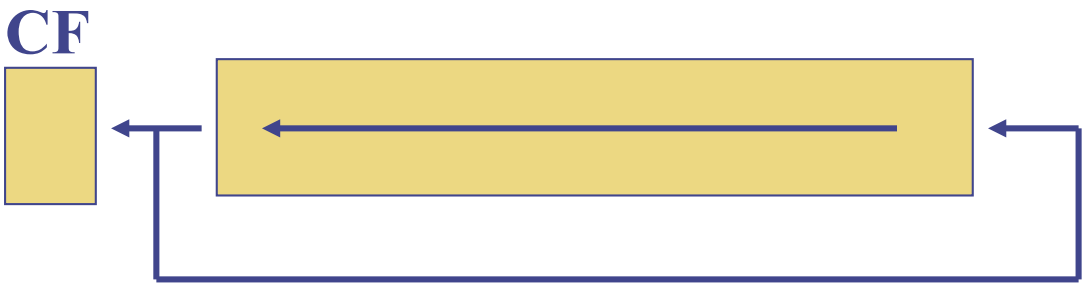
F5→1111 0101

```
MOV     AH, 0F5H
SAR     AH, 1
    ; (AH)= ?     0FAH
    ; (CF)= ?     1
```

比较两种指令，其结果说明什么？

(5) 循环左移 ROL OPD, n

将(OPD)的最高位与最低位连接起来，组成一个环。将环中的所有位一起向左移动n位，CF的内容为最后移入位的值。




```
MOV   DL, 0EAH
ROL   DL,  4
```

(DL) = ?　　CF = ?

0AEH　0

(6) 循环右移 ROR OPD, n

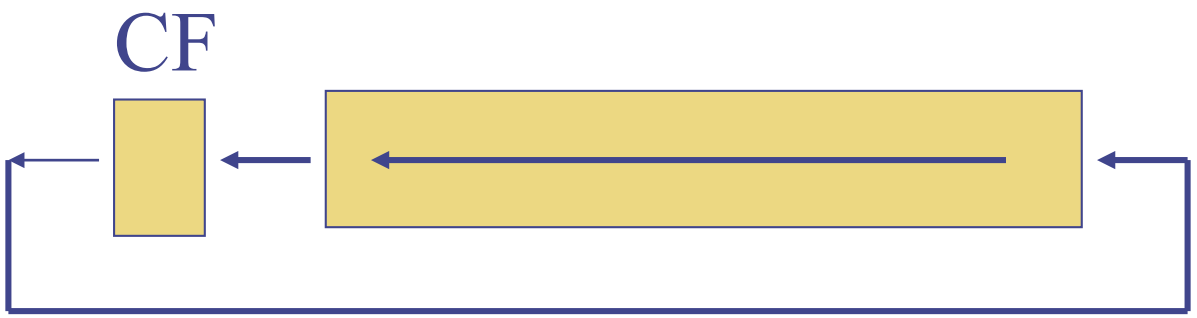
将(OPD)的最高位与最低位连接起来，组成一个环。将环中的所有位一起向右移动n位，CF的内容为最后移入位的值。


```
MOV   DL, 0EAH
ROR   DL, 4
```

(DL) = ? CF = ? 0AEH 1

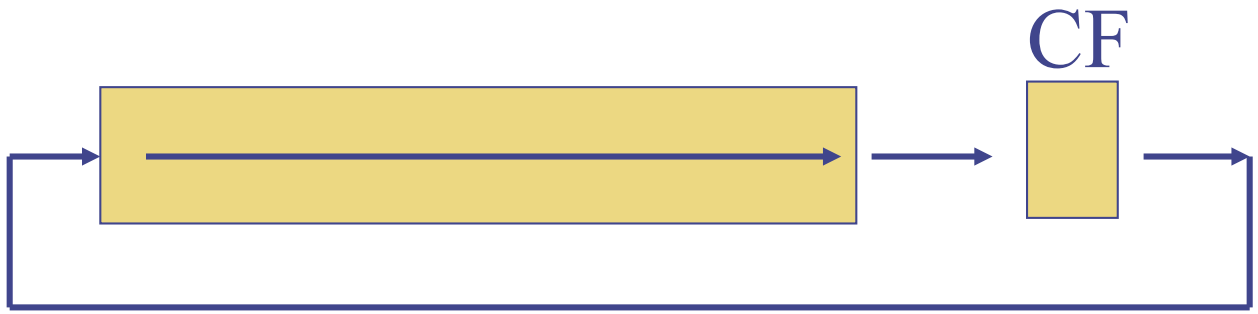
(7) 带进位的循环左移 RCL OPD, n

将（OPD)的最高位、 CF、(OPD)最低位连接起来，组成一个环。将环中的所有位一起向左移动n位，CF的内容为最后移入位的值。



(8) 带进位的循环右移　RCR　OPD,　n

- (OPD)、CF连接组成一个环;
- 将环中的所有位一起向右移动n位;
- CF的内容为最后移入位的值。



5.5 移位指令

SAL、 SHL
SAR、 SHR
ROL、 ROR
RCL、 RCR

移动方向？
CF的摆放位置？
移动规则？

REVIEW

数据传送指令

1、一般数据传送指令

MOV、MOVSX、MOVZX、XCHG、XLAT

2、堆栈操作指令

PUSH、POP、PUSHA、PUSHAD、POPA、POPAD

3、标志寄存器传送指令

PUSHF、POPF、PUSHFD、POPFD、LAHF、SAHF

4、地址传送指令

LEA

5、输入、输出指令

IN、OUT

REVIEW

算术运算指令

1、加法指令

INC、ADD、ADC

2、减法指令

DEC、NEG、SUB、SBB、CMP

3、乘法指令

IMUL (三种形式)、MUL

4、除法指令

IDIV、DIV

5、符号扩展指令

CBW、CWD

REVIEW

华中科技大学

逻辑运算指令

NOT、AND、TEST、OR、XOR

移位指令

SAL、 SHL

SAR、 SHR

ROL、 ROR

RCL、 RCR

试用不同指令将（AX)置0。

```
MOV   AX,  0
SUB    AX,  AX
AND    AX,  0
XOR    AX, AX
SHL    AX, 16
```

试用不同的指令，将AX的高、低字节内容互换。

```
XCHG  AH, AL
ROL    AX,  8
ROR    AX, 8
```