IBM PC机的指令系统

- 8086/8088指令系统分成下列六大类：
  - 数据传送指令
  - 算术运算指令
  - **逻辑运算和移位指令**
  - 控制转移指令
  - CPU控制指令
  - 串操作指令

逻辑运算指令

逻辑与指令：AND DST, SRC
执行操作：(DST) ← (DST) ∧ (SRC)
用途：用于屏蔽一个数的某些位。
逻辑或指令：OR DST, SRC
执行操作：(DST) ← (DST) ∨ (SRC)
用途：用于置位一个数的某些位。
异或指令：XOR DST, SRC
执行操作：(DST) ← (DST) ∀ (SRC)
用途：将一个数的某些位取反。
测试指令：TEST OPR1, OPR2
执行操作：(OPR1) ∧ (OPR2)
用途：用于测试一个数的某些位。

CF OF SF ZF PF AF
0 0 * * * 无定义
根据运算结果设置

逻辑运算指令

逻辑非指令：NOT OPR

执行操作：(OPR) ← ¬ (OPR)

功能：按位取反

* OPR不能为立即数
* 不影响标志位

AND VS TEST

- AND具有破坏性，TEST没有
  - AL = 0FFH
  - AND AL，0
  - TEST AL，0
- 同SUB和CMP

例子

例：屏蔽AL的0、1两位

```
AND  AL, 0FCH
```

```
      * * * *  * * * *
AND   1 1 1 1  1 1 0 0
      * * * *  * * 0 0
```

例：置AL的第5位为1

```
OR  AL, 20H
```

```
      * * * *  * * * *
OR    0 0 1 0  0 0 0 0
      * * 1 *  * * * *
```

例：使AL的0、1位变反

```
XOR  AL, 3
```

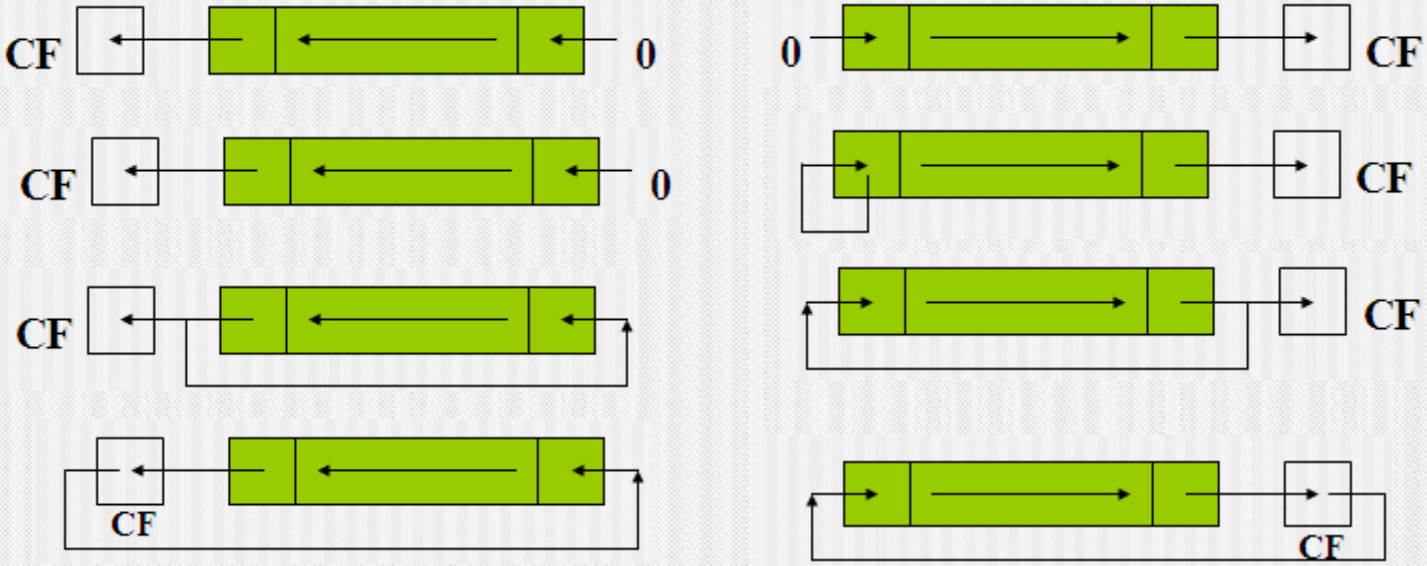
```
      * * * *  * *  * *
XOR   0 0 0 0  0 0  1 1
      * * * *  * *  * *
```

例：测试某些位是0是1

```
TEST  AL, 1
JZ  EVEN
```

移位指令

- 分类：
  - 逻辑SHL/SHR　算术SAL/SAR
  - 循环ROL/ROR　带进位循环RCL/RCR

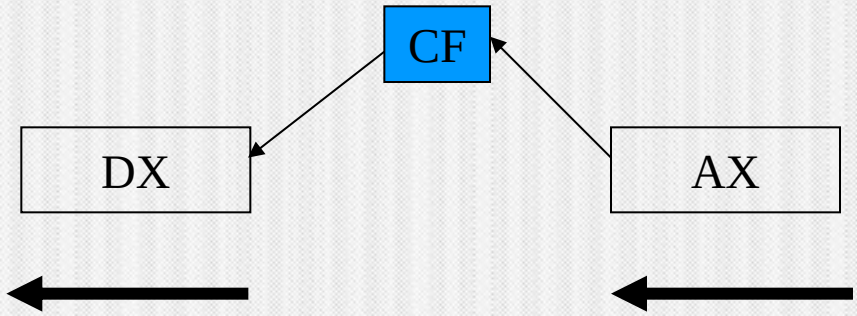


移位指令

- 共同特点
  - 都是按位进行
  - 当移动的位数为一位时，用立即数1；当移动二位或二位以上时，要预先将移动的位数存放在CL中。
  - SHL AL, 2 →MOV CL,2 ; SHL AL,CL;

例子（一）

- RCL和RCR常用在多字节数的移位。
  - 在DX和AX中存放着一个32位数据，试将其左移1位。
  - SHL AX,1
  - RCL DX,1
  - 右移如何处理?



例子（二）

- 把(BL)中的8位数高低4位互换
  - MOV CL,4
  - ROL/ROR BL, CL

  - MOV DL,BL
  - MOV CL,4
  - SHR BL, CL
  - SHL DL, CL
  - OR BL, DL

IBM PC机的指令系统

- 80x86指令系统分成下列六大类：
  - 数据传送指令
  - 算术运算指令
  - 逻辑运算和移位指令
  - **控制转移指令**
  - CPU控制指令
  - 串操作指令

指令执行过程

- 根据CS：IP，取指令
- 分析指令
  - 指令长度L：IP+L
  - 执行哪种操作
- 执行指令：
  - 取值运算或
  - 修改CS：IP

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

无条件转移指令

- 格式：JMP 地址表达式
- 功能：使程序的流程无条件跳到转移地址所指的地方。
  - 转移目的地址 = (CS)×16+(IP)
  - 段内转移：改变IP的内容，CS的内容不变。
  - 段间转移：IP、CS的内容都改变。

- 段内直接转移(相对寻址)：指令中直接给出到达的目标地址
  - 例如：JMP PROG1　　；转移范围：－32768～＋32767
- 段内间接转移(间接寻址)：指定某个寄存器的内容或某个字单元的内容作为转移地址的偏移地址。
  - 例如：JMP BX　　；(BX) → IP
  - JMP WORD PTR[1000H]　；(DS:1000H) → IP
  - JMP WORD PTR[SI+2]　；(DS:SI+2) → IP
  - JMP TABLE[BX]　　；(DS:TABLE+(BX)) → IP

- 段间直接转移(直接寻址)：通过标号直接给出转移地址
  - JMP NEXTP1　　；NEXTP1的段址→ CS，偏址→ IP
- 段间间接转移(间接寻址)：指定一个4字节的单元内容作为转移地址，其中低二字节内容→IP，高二字节内容→CS。
  - 例如：JMP DWORD PTR[100H]
  - JMP DWORD PTR[BX]

条件转移指令

- 标志位条件转移指令
- 二个无符号数比较转移指令
- 二个带符号数比较转移指令

标志位条件转移指令

- JC 标号；当(CF)=1，则转移。
  - JNC 标号；当(CF)=0，则转移。
- JZ/JE 标号；当(ZF)=1，则转移。
  - JNZ/JNE 标号；当(ZF)=0，则转移。
- JS 标号；当(SF)=1，则转移。
  - JNS 标号；当(SF)=0，则转移。
- JO 标号；当(OF)=1，则转移。
  - JNO 标号；当(OF)=0，则转移。
- JP 标号；当(PF)=1，则转移。
  - JNP 标号；当(PF)=0，则转移。

二个无符号数比较转移指令

- 设A为被减数，B为减数。

  CMP A,B

- JA 标号；当A>B时转移；
- JAE 标号；当A≥B时转移；
- JB 标号；当A<B时转移；
- JBE 标号；当A≤B时转移。

  CF ZF

二个带符号数比较转移指令

- JG 标号；当被减数大转移；
- JGE 标号；当被减数大于等于减数转移；
- JL 标号；当被减数小转移；
- JLE 标号；当被减数小于等于减数转移

SF OF ZF

真值表

A	B	OF	SF	大小
+	-	1	1	>
+	-	0	0	>
+	+	0	0	>
+	+	0	1	<
-	-	0	0	>
-	-	0	1	<

例子1

- 完成分段函数

$$AH = \begin{cases} -1 & AL < 0 \\ 0 & AL = 0 \\ 1 & AL > 0 \end{cases}$$

例子1

```
        CMP AL,0
        JG    L1
        JL    L2
        MOV  AH,0
L1: MOV AH,1
L2:MOV AH,-1
```

```
        CMP AL,0
        JG    L1
        JL    L2
        MOV  AH,0
        JMP  DONE
L1: MOV AH,1
        JMP  DONE
L2:MOV AH,-1
DONE:  …
```

循环控制指令 LOOP/LOOPE/LOOPNE

- 格式：LOOP 标号；
  - 功能：(CX)-1→ CX，若(CX)≠0，则转移
  - 功能等价
    - DEC CX
    - JNZ 标号
- 格式：LOOPE/LOOPNE
  - (CX<>0) and (ZF == 1/0)

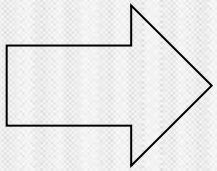
JCXZ条件转移指令

- 格式：JCXZ 标号
  - 功能：当(CX) = 0时，转向标号

Notice！

- 除无条件转移指令只能使用标号；
- 只能是段内直接短转移，即偏移量为-128~127；

JCC label



```
JNCC skipnext
JMP label
Skipnext：
…
Label:
```

- 使用LOOP指令，注意初始值是否为0。

习题

- 找出100个有符号字节数中间的最大数。
  - 设有符号数首地址是ARR

```
        MOV  AL, 80H
        MOV  CX，100
        MOV  SI, 0
AGAIN：CMP  AL，ARR[SI]
        JGE  NEXT
        MOV  AL, ARR[SI]
NEXT:
        INC  SI
        LOOP  AGAIN
```

过程调用和返回指令

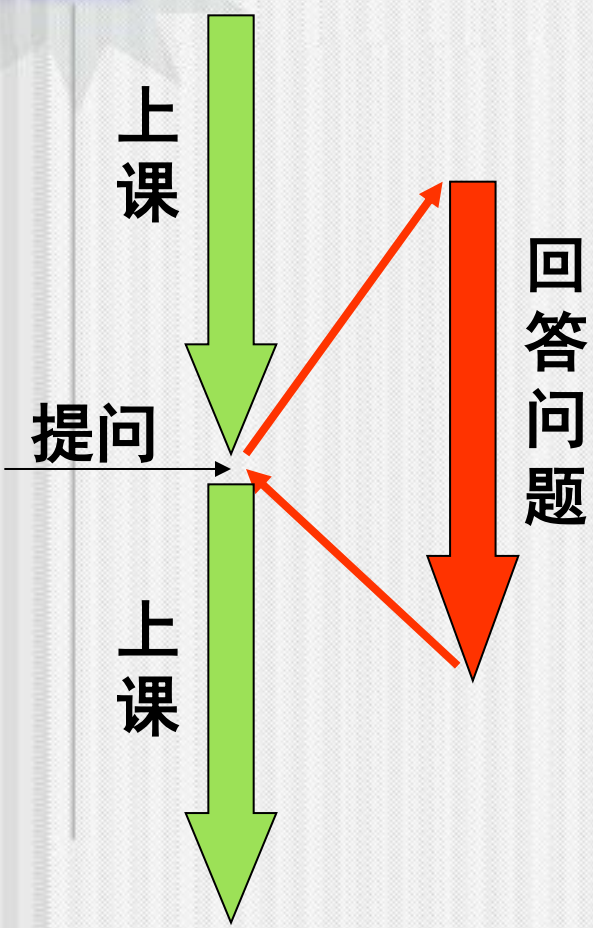
- 调用指令 CALL
  - 格式：CALL 子程序/地址表达式
  - 功能：
    - 保护断点——将当前断点压入堆栈；
    - 转入子程序——将子程序段的入口地址送入IP（/CS）；

- 段内直接调用 (相对寻址)
  - CALL Subx 执行过程：PUSH IP ; Subx入口地址→IP
- 段内间接调用(间接寻址)： 子程序的入口偏移地址存放在Reg或者Mem中
  - CALL BX
  - CALL WORD PTR [1000H]
  - 执行过程：PUSH IP； 子程序入口地址→ IP

- 段间直接调用(直接寻址)
  - CALL Subf
  - 执行过程：PUSH CS;PUSH IP;子程序入口地址→ CS，IP
- 段间间接调用(间接寻址)：子程序入口段地址和偏移地址存在DWORD中
  - CALL DWORD PTR ADDR
  - 执行过程：PUSH CS;PUSH IP;ADDR → IP;ADDR+2 → CS

- 子程序返回指令 RET
  - 格式：RET [n]
  - 功能：返回主程序。根据子程序的属性Near、Far决定：
    - 段内返回：POP IP
    - 段间返回：POP IP；POP CS
  - RET n：返回；并执行SP = SP + n;
  - CALL与RET类型必须一致

中断调用与返回指令



- 中断: 计算机暂停现行程序的运行，转去执行另一程序以处理发生的事件，处理完毕后又自动返回原来的程序继续运行
- 中断类型:
  - 外部中断:可/不可屏蔽中断
  - 内部中断:除法错/溢出(INTO)/单步调试/INT n

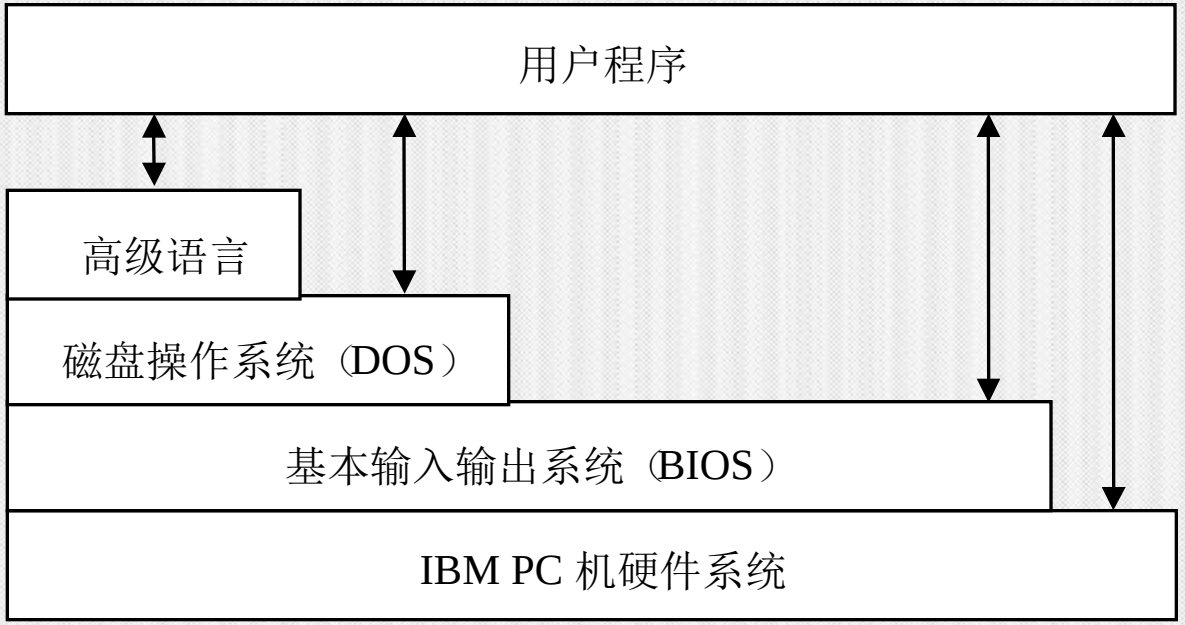
相关概念

- 中断：数据传输方式；软中断和硬中断
- 中断服务程序：处理中断的子程序；
- 中断向量：中断子程序的入口地址
- 中断向量表：存放在00000H地址的四字节表格
- 中断类型码：给中断向量的一个编号
- (中断向量表查看：debug)

- 中断调用指令 INT
  - 格式：INT n
  - 功能：调用n号中断子程序
  - 操作：
    - PUSHF；PUSH CS；PUSH IP
    - 取得中断向量，转入
- 中断返回指令 IRET
  - 格式：IRET
  - 操作：POP IP；POP CS；POPF

DOS 中断调用

- MS-DOS “API” & System “API”
- AH为功能号
- DOS INT部分使用AL/AX作为返回值— 0-成功; 1-失败



DOS 21H号中断调用

- 1. 从键盘读入一个字符
  - MOV AH，1/8
    [回显/不回显]
  - INT 21H ;
  - 键入字符的ASCII存入AL中
- 2. 显示一个字符到屏幕
  - MOV AH，2
  - MOV DL, ASCII
  - INT 21H ;

DOS 21H号中断调用

- 3. 显示一个字符串到屏幕
  - MOV AH，9
  - LEA DX, STRING
  - INT 21H ;
  - ;字符串要求以”$”结束
- 4. 从键盘读入一个字符串到屏幕
  - MOV AH，0AH
  - LEA DX，STRING
  - INT 21H
  - STRING第一个字节为定义的最大长度（含回车）；第二个为实际输入的长度（不含回车）。

DOS 21H号中断调用

- 5. 返回DOS
  - MOV AH,4CH
  - INT 21H

IBM PC机的指令系统

- 80x86指令系统分成下列六大类：
  - 数据传送指令
  - 算术运算指令
  - 逻辑运算和移位指令
  - 控制转移指令
  - 串操作指令
  - **CPU控制指令**

NOP

- 空操作 (机器码: 90H)
- 与XCHG AX,AX相同
- 用途:
  - Timer
    - 1个时钟周期；DSP, C51
  - Place Holder
    - 一个字节；

HLT

- 暂停指令
- 功能：
  - 使CPU进入暂停状态，直到系统复位或发生外部中断
  - 应用程序一般不使用

LOCK

- 封锁前缀
- 用途：
  - 用于多处理器系统，使当前处理器锁住总线，以保证当前指令为原子操作；
  - 当目的操作数为内存操作数时，为了完成“读-修改-写内存” 的操作不被打断；
- 示例：Lock add [bx],ax