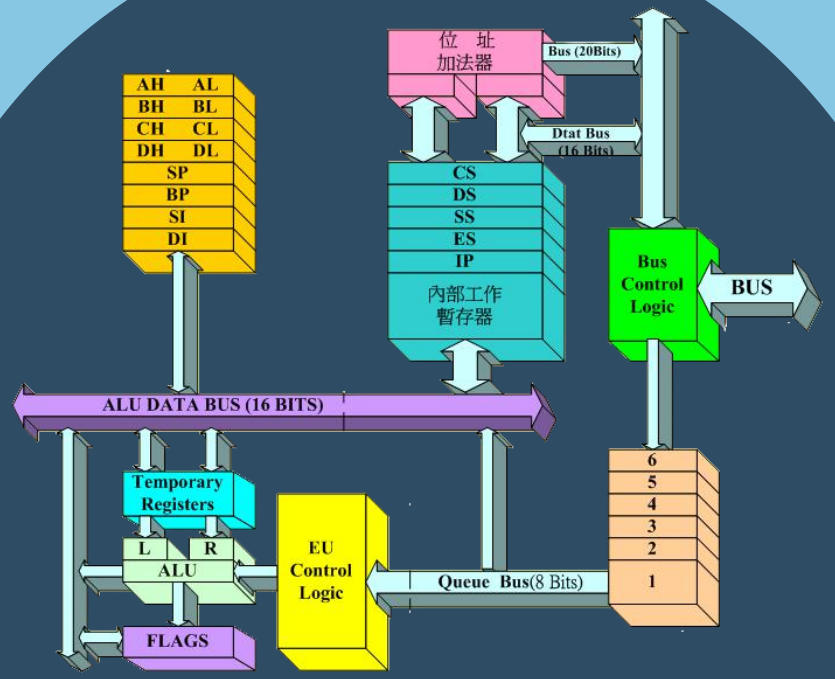
# 其他转移指令

贺利坚　主讲




汇编语言程序设计

Assembly Language

# jcxz指令

- 指令格式：jcxz 标号
- 功能：如果(cx)=0，则转移到标号处执行
  当(cx)≠0时，什么也不做（程序向下执行）
  - 当(cx)=0时，(IP)=(IP)+8位位移）
    - 8位位移=“标号”处的地址-jcxz指令后的第一个字节的地址；
    - 8位位移的范围为-128~127，用补码表示；
    - 8位位移由编译程序在编译时算出。
- jcxz是有条件转移指令
  - 所有的有条件转移指令都是短转移
  - 对IP的修改范围都为-128~127
  - 在对应的机器码中包含转移的位移，而不是目的地址

```
1  assume cs:codesg
2  codesg segment
3  start: mov ax,2000H
4         mov ds, ax
5         mov bx,0
6      s: mov cx, [bx]
7         jcxz ok
8         inc bx
9         inc bx
10        jmp short s
11     ok: mov dx, bx
12        mov ax, 4c00H
13        int 21H
14 codesg ends
15 end start
```

```
C:\>debug p9-7.exe
-u
076A:0000 B80020        MOV     AX,2000
076A:0003 8ED8          MOV     DS,AX
076A:0005 BB0000        MOV     BX,0000
076A:0008 8B0F          MOV     CX,[BX]
076A:000A E304          JCXZ    0010
076A:000C 43            INC     BX
076A:000D 43            INC     BX
076A:000E EBF8          JMP     0008
076A:0010 8BD3          MOV     DX,BX
076A:0012 B8004C        MOV     AX,4C00
076A:0015 CD21          INT     21
```

# loop指令

- 指令格式：loop 标号
- 指令操作

（1）(cx)=(cx)-1;

（2）当(cx)≠0时，则转移到标号处执行

当(cx)=0时，程序向下执行

- 如果(cx)≠0，(IP)=(IP)+8位位移
  - 8位位移=“标号”处的地址-loop指令后的第一个字节的地址
  - 8位位移的范围为-128~127，用补码表示
  - 8位位移由编译程序在编译时算出

```
1  assume cs:codesg
2  codesg segment
3  start: mov cx, 6h
4         mov ax,10h
5      s: add ax, ax
6         loop s
7         mov ax, 4c00H
8         int 21H
9  codesg ends
10 end start
```

```
C:\>debug p9-8.exe
-u
076A:0000 B90600        MOV     CX,0006
076A:0003 B81000        MOV     AX,0010
076A:0006 03C0          ADD     AX,AX
076A:0008 E2FC          LOOP    0006
076A:000A B8004C        MOV     AX,4C00
076A:000D CD21          INT     21
076A:000F 01B85C00      ADD     [BX+SI+005C
```

loop s 在执行时只涉及到 s的位移（-4，前移4个字节，补码表示为FCH）

# 根据位移进行“相对”转移的意义

- 对 IP的修改是根据转移目的地址和转移起始地址之间的位移来进行

  jmp short 标号

  jmp near ptr 标号

  jcxz 标号

  loop 标号

- 在它们对应的机器码中不包含转移的目的地址，而包含的是到目的地址的位移。
  - 如果 loop s 的机器码中包含的是 s 的地址，则就对程序段在内存中的偏移地址有了严格的限制，易引发错误。
  - 当机器码中包含的是转移的位移，无论 s 处的指令的实际地址是多少，loop指令转移的相对位移是不变的。
- 这样的设计，方便了程序段在内存中的浮动装配。

```
C:\>debug p9-7.exe
-u
076A:0000 B80020        MOV     AX,2000
076A:0003 8ED8          MOV     DS,AX
076A:0005 BB0000        MOV     BX,0000
076A:0008 8B0F          MOV     CX,[BX]
076A:000A E304          JCXZ    0010
076A:000C 43            INC     BX
076A:000D 43            INC     BX
076A:000E EBF8          JMP     0008
076A:0010 8BD3          MOV     DX,BX
076A:0012 B8004C        MOV     AX,4C00
076A:0015 CD21          INT     21
```

```
C:\>debug p9-8.exe
-u
076A:0000 B90600        MOV     CX,0006
076A:0003 B81000        MOV     AX,0010
076A:0006 03C0          ADD     AX,AX
076A:0008 E2FC          LOOP    0006
076A:000A B8004C        MOV     AX,4C00
076A:000D CD21          INT     21
076A:000F 01B85C00      ADD     [BX+SI+005C
```