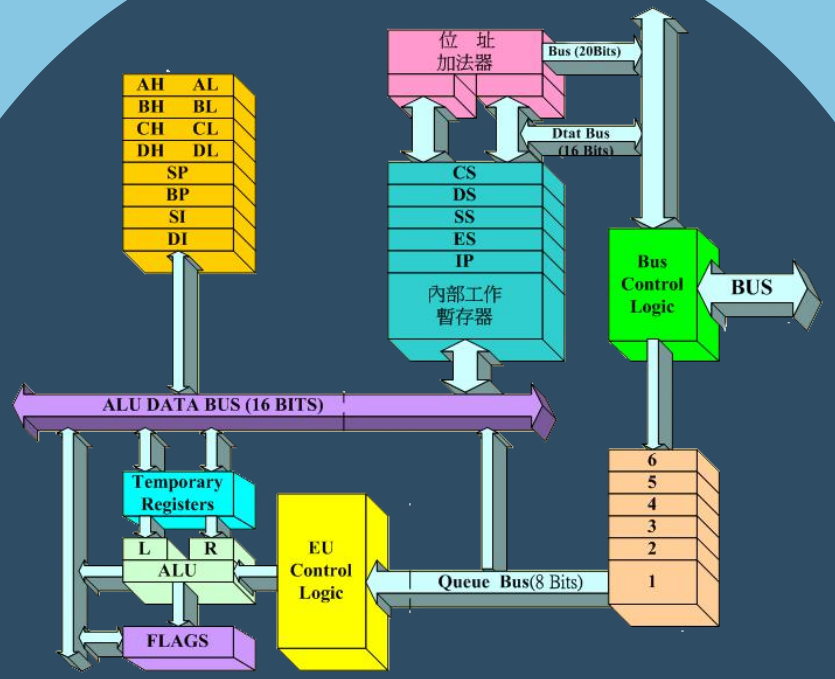
Loop指令

贺利坚　主讲



汇编语言程序设计

Assembly Language

Loop指令

- 功能：实现循环（计数型循环）
- 指令的格式

loop 标号

- CPU 执行loop指令时要进行的操作
  - ① (cx)=(cx)-1；
  - ② 判断cx中的值

    不为零则转至标号处执行程序

    如果为零则向下执行。
- 要求
  - cx 中要提前存放循环次数，因为(cx)影响着loop指令的执行结果
  - 要定义一个标号

```
1  ; loop指令示例程序
2  assume cs:code
3  code segment
4      mov ax,2
5      mov cx,11
6   s: add ax,ax
7      loop s
8
9      mov ax,4c00h
10     int 21h
11 code ends
12 end
```

本程序功能：2 -> 4 -> 8 -> 16 -> 32 -> ...

用loop指令编程实例

任务1：编程计算2^2

```
assume cs:code
code segment
        mov ax,2
        ; 用2+2 实现2*2
        add ax,ax

        mov ax,4c00h
        int 21h
code ends
end
```

任务2：编程计算2^3

```
assume cs:code
code segment
        mov ax,2
        add ax,ax
        add ax,ax

        mov ax,4c00h
        int 21h
code ends
end
```

任务3：编程计算2^12

```
assume cs:code
code segment
        mov ax,2
        ; 做11次add ax,ax

        mov ax,4c00h
        int 21h
code ends
end
```

```
 1  assume cs:code
 2  code segment
 3          mov ax,2
 4          mov cx,11
 5     s:   add ax,ax
 6          loop s
 7
 8          mov ax,4c00h
 9          int 21h
10  code ends
11  end
```

用cx和loop 指令相配合实现循环功能的三个要点：

（1）在cx中存放循环次数；

（2）用标号指定循环开始的位置；

（3）在标号和loop 指令的中间，写上要循环执行的程序段（循环体）。

用Debug执行程序

```
C:\>masm p5-1;
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.


  51798 + 464746 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\>link p5-1;

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

LINK : warning L4021: no stack segment
```

```
C:\>debug p5-1.exe
-u
076A:0000 B80200        MOV     AX,0002
076A:0003 B90B00        MOV     CX,000B
076A:0006 03C0          ADD     AX,AX
076A:0008 E2FC          LOOP    0006
076A:000A B8004C        MOV     AX,4C00
076A:000D CD21          INT     21
076A:000F 26FA          JBE     000B
```

```
C:\>debug p5-1.exe
-g

Program terminated normally
-
```

```
-g 0006

AX=0002  BX=0000  CX=000B  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0006   NV UP EI PL NZ NA PO NC
076A:0006 03C0          ADD     AX,AX
-g 000D

AX=4C00  BX=0000  CX=0000  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000D   NV UP EI PL NZ NA PE NC
076A:000D CD21          INT     21
```

```
-r
AX=FFFF  BX=0000  CX=000F  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0000   NV UP EI PL NZ NA PO NC
076A:0000 B80200        MOV     AX,0002
-t

AX=0002  BX=0000  CX=000F  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0003   NV UP EI PL NZ NA PO NC
076A:0003 B90B00        MOV     CX,000B
-t

AX=0002  BX=0000  CX=000B  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0006   NV UP EI PL NZ NA PO NC
076A:0006 03C0          ADD     AX,AX
-t

AX=0004  BX=0000  CX=000B  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0008   NV UP EI PL NZ NA PO NC
076A:0008 E2FC          LOOP    0006
-t

AX=0004  BX=0000  CX=000A  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0006   NV UP EI PL NZ NA PO NC
076A:0006 03C0          ADD     AX,AX
-
```

t命令和p命令的区别

```
-r
AX=FFFF  BX=0000  CX=000F  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0000   NV UP EI PL NZ NA PO NC
076A:0000 B80200        MOV     AX,0002
-t

AX=0002  BX=0000  CX=000F  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0003   NV UP EI PL NZ NA PO NC
076A:0003 B90B00        MOV     CX,000B
-t

AX=0002  BX=0000  CX=000B  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0006   NV UP EI PL NZ NA PO NC
076A:0006 03C0          ADD     AX,AX
-t

AX=0004  BX=0000  CX=000B  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0008   NV UP EI PL NZ NA PO NC
076A:0008 E2FC          LOOP    0006
-t

AX=0004  BX=0000  CX=000A  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0006   NV UP EI PL NZ NA PO NC
076A:0006 03C0          ADD     AX,AX
-
```

```
-g 0006

AX=0002  BX=0000  CX=000B  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0006   NV UP EI PL NZ NA PO NC
076A:0006 03C0          ADD     AX,AX
-g 000D

AX=4C00  BX=0000  CX=0000  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000D   NV UP EI PL NZ NA PE NC
076A:000D CD21          INT     21
```

```
-p

AX=0002  BX=0000  CX=000F  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0003   NV UP EI PL NZ NA PO NC
076A:0003 B90B00        MOV     CX,000B
-p

AX=0002  BX=0000  CX=000B  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0006   NV UP EI PL NZ NA PO NC
076A:0006 03C0          ADD     AX,AX
-p

AX=0004  BX=0000  CX=000B  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0008   NV UP EI PL NZ NA PO NC
076A:0008 E2FC          LOOP    0006
-p

AX=1000  BX=0000  CX=0000  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000A   NV UP EI PL NZ NA PE NC
076A:000A B8004C        MOV     AX,4C00
```

- 继续命令P(Proceed)：类似T命令，逐条执行指令、显示结果。但遇子程序、中断等时，直接执行，然后显示结果。
- 运行命令G(Go)：从指定地址处开始运行程序，直到遇到断点或者程序正常结束；G命令还可以指定执行到的代码地址。

例：用Loop指令编程

- 问题：计算123x236，结果存储在ax中
- 方法：用加法实现乘法，将123连加236次

```
1   assume cs:code
2   code segment
3       mov ax,0
4       mov cx,236
5    s: add ax,123
6       loop s
7
8       mov ax,4c00h
9       int 21h
10  code ends
11  end
```