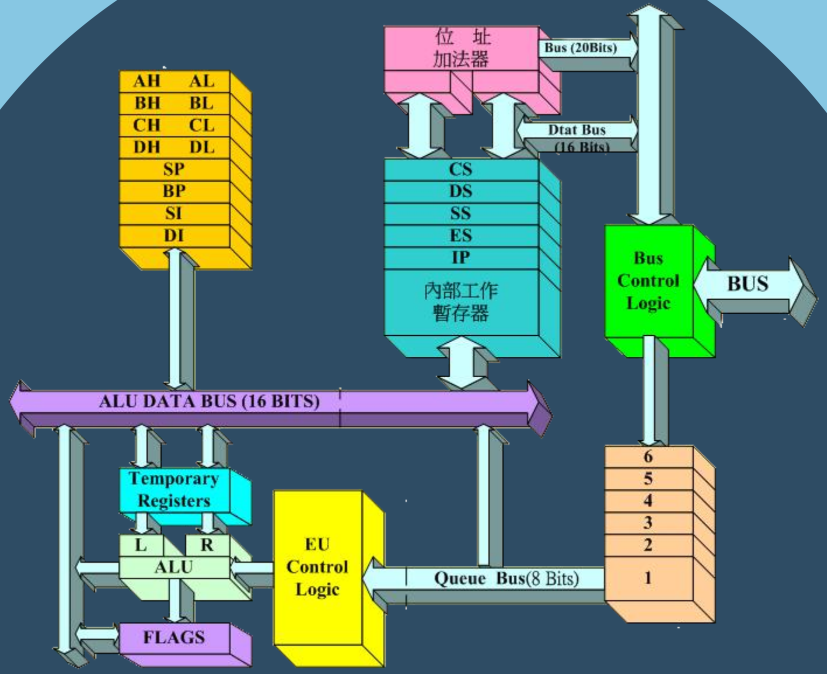
描述内存单元的标号

贺利坚　主讲



汇编语言程序设计

Assembly Language

关于标号

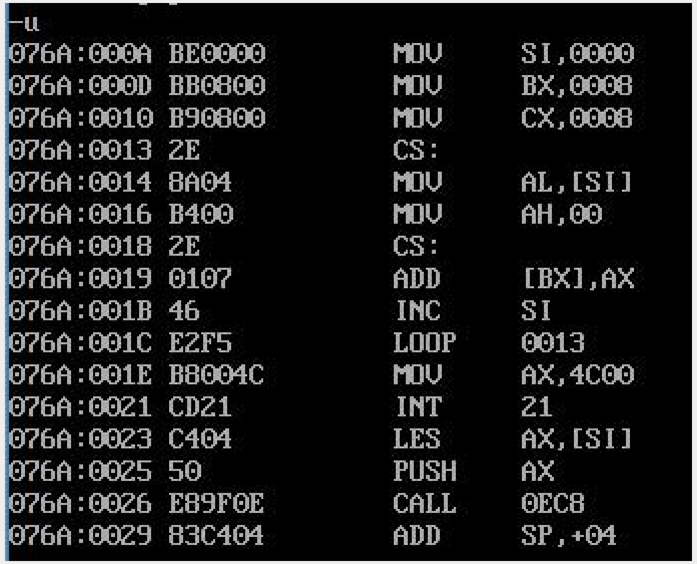
- 代码段中的标号可以用来标记指令、段的起始地址。
- 代码段中的数据也可以用标号

求2^8

```
assume cs:code
code segment
start :mov ax, 1
       mov cx,8
   s : add ax, ax
       loop s
       mov ax,4c00h
       int 21h
code ends
end start
```

将a 标号处的8个字节数据累加，结果存储到b标号处的字中。

```
assume cs:code
code segment
        a : db 1,2,3,4,5,6,7,8
        b : dw 0
start :mov si,offset a
        mov bx,offset b
        mov cx,8
     s : mov al,cs:[si]
        mov ah,0
        add cs:[bx],ax
        inc si
        loop s
        mov ax,4c00h
        int 21h
code ends
end start
```

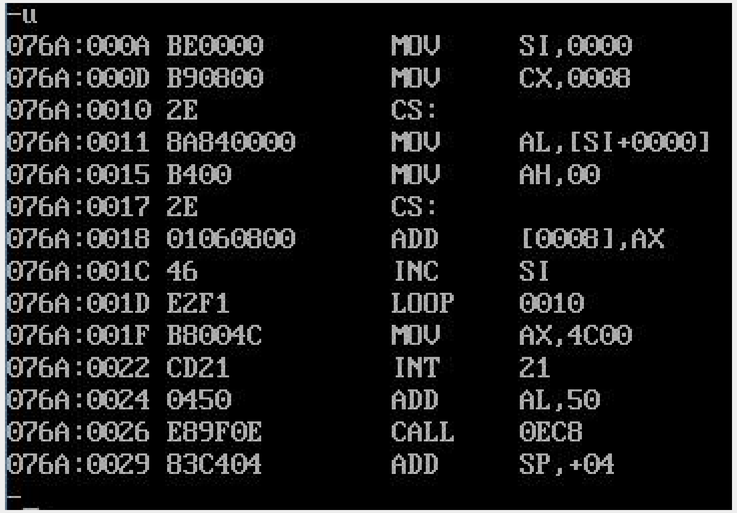


```
-u
076A:000A BE0000        MOV     SI,0000
076A:000D BB0800        MOV     BX,0008
076A:0010 B90800        MOV     CX,0008
076A:0013 2E            CS:
076A:0014 8A04          MOV     AL,[SI]
076A:0016 B400          MOV     AH,00
076A:0018 2E            CS:
076A:0019 0107          ADD     [BX],AX
076A:001B 46            INC     SI
076A:001C E2F5          LOOP    0013
076A:001E B8004C        MOV     AX,4C00
076A:0021 CD21          INT     21
076A:0023 C404          LES     AX,[SI]
076A:0025 50            PUSH    AX
076A:0026 E89F0E        CALL    0EC8
076A:0029 83C404        ADD     SP,+04
```

offset、cs: ……
能简单点吗？

去了冒号的数据标号

```
assume cs:code
code segment
        a db 1,2,3,4,5,6,7,8
        b dw 0
start :  mov si,0
        mov cx,8
   s :   mov al,a[si]
        mov ah,0
        add b,ax
        inc si
        loop s
        mov ax,4c00h
        int 21h
code ends
end start
```



```
-u
076A:000A BE0000        MOV     SI,0000
076A:000D B90800        MOV     CX,0008
076A:0010 2E            CS:
076A:0011 8A840000      MOV     AL,[SI+0000]
076A:0015 B400          MOV     AH,00
076A:0017 2E            CS:
076A:0018 01060800      ADD     [0008],AX
076A:001C 46            INC     SI
076A:001D E2F1          LOOP    0010
076A:001F B8004C        MOV     AX,4C00
076A:0022 CD21          INT     21
076A:0024 0450          ADD     AL,50
076A:0026 E89F0E        CALL    0EC8
076A:0029 83C404        ADD     SP,+04
-
```

- 数据标号
  - 数据标号标记了存储数据的单元的地址和长度。
  - 数据标号不同于仅仅表示地址的地址标号。

- 我们在code 段中使用的标号a、b后面没有“:”，它们同时描述内存地址和单元长度的标号。
- 标号a
  - 地址code:0
  - 以后的内存单元都是字节
- 标号b
  - 地址code:8
  - 以后的内存单元都是字

数据标号同时描述内存地址和单元长度

```
assume cs:code
code segment
        a db 1,2,3,4,5,6,7,8
        b dw 0
start :
        ......


code ends
end start
```

a 代表地址为code:0 ，长度为字节的内存

b 代表地址为code:8 ，长度为字的内存单元

mov al,a [si]

↳ mov al,cs:0[si]

mov al,a[3]

↳ mov al,cs:0[3]

mov al,a[bx+si+3]

↳ mov al,cs:0[bx+si+3]

mov ax,b

↳ mov ax,cs:[8]

mov b,2

↳ mov word ptr cs:[8],2

inc b

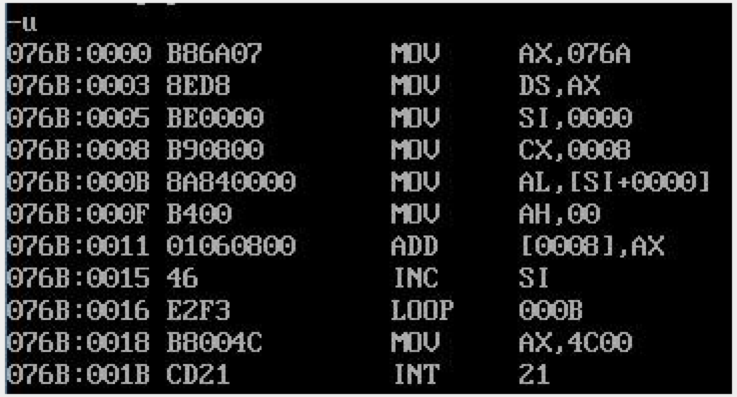
↳ inc word ptr cs:[8]

mov al,b

↳ error!

更常见的方式：数据段中的数据标号

```
assume cs:code,ds:data
data segment
        a db 1,2,3,4,5,6,7,8
        b dw 0
data ends
code segment
start:  mov ax,data
        mov ds,ax
        mov si,0
        mov cx,8
s:      mov al,a[si]
        mov ah,0
        add b,ax
        inc si
        loop s
        mov ax,4c00h
        int 21h
code ends
end start
```



```
-u
076B:0000 B86A07         MOV     AX,076A
076B:0003 8ED8           MOV     DS,AX
076B:0005 BE0000         MOV     SI,0000
076B:0008 B90800         MOV     CX,0008
076B:000B 8A840000       MOV     AL,[SI+0000]
076B:000F B400           MOV     AH,00
076B:0011 01060800       ADD     [0008],AX
076B:0015 46             INC     SI
076B:0016 E2F3           LOOP    000B
076B:0018 B8004C         MOV     AX,4C00
076B:001B CD21           INT     21
```

对比：地址标号只能在代码段中使用

扩展用法：将标号当作数据来定义

```
data segment
     a db 1,2,3,4,5,6,7,8
     b dw 0
     c dw a,b
data ends
```

```
data segment
     a db 1,2,3,4,5,6,7,8
     b dw 0
     c dw offset a, offset b
data ends
```

```
data segment
     a db 1,2,3,4,5,6,7,8
     b dw 0
     c dd a,b
data ends
```

```
data segment
     a db 1,2,3,4,5,6,7,8
     b dw 0
     c dw offset a, seg a, offset b, seg b
data ends
```

seg操作符——取段地址