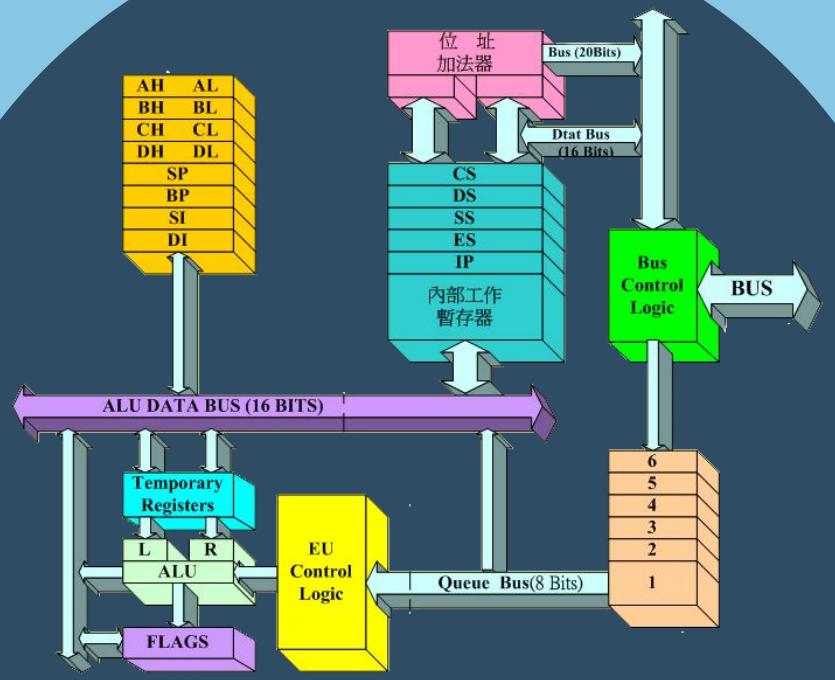
[bx+idata]方式寻址

贺利坚　主讲

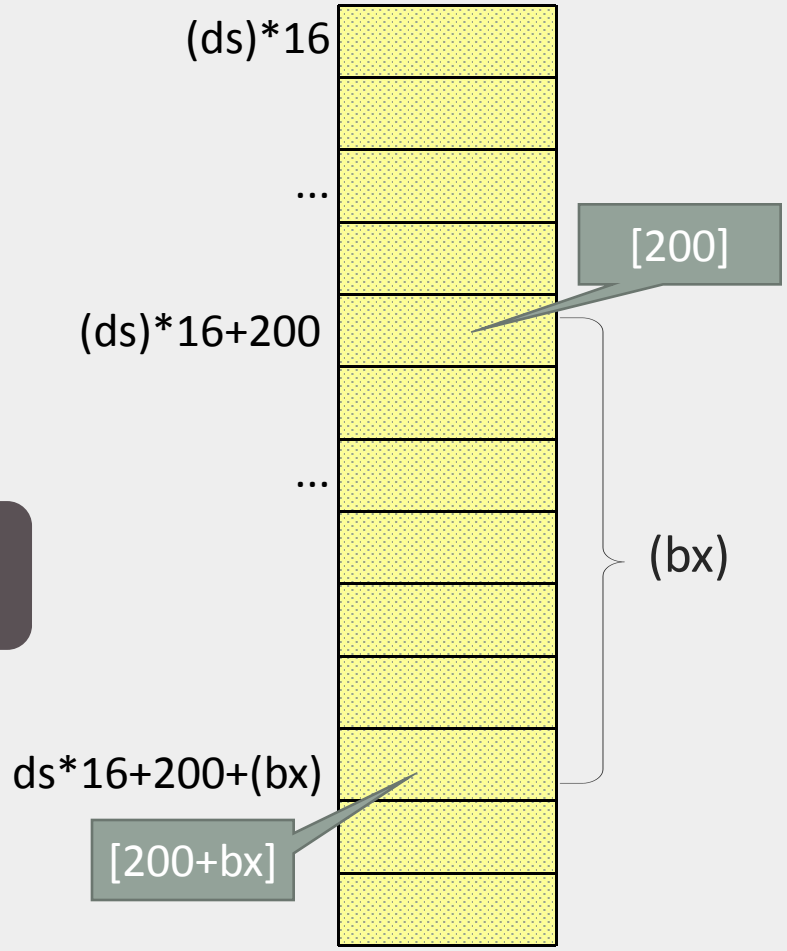


汇编语言程序设计

Assembly Language

[bx+idata]的含义

- [bx+idata]表示一个内存单元，它的偏移地址为(bx)+idata（bx中的数值加上idata）。
- mov ax,[bx+200] / mov ax, [200+bx] 的含义
  - 将一个内存单元的内容送入ax
  - 这个内存单元的长度为2字节（字单元），存放一个字
  - 内存单元的段地址在ds中，偏移地址为200加上bx中的数值
  - 数学化的描述为：(ax)=((ds)*16+200+(bx))
- 指令mov ax,[bx+200]的其他写法（常用）
  - mov ax,[200+bx]
  - mov ax,200[bx]
  - mov ax,[bx].200



示例

```
C:\>debug
-a
073F:0100 mov ax, 2000
073F:0103 mov ds, ax
073F:0105 mov bx, 1000
073F:0108 mov ax, [bx]
073F:010A mov cx, [bx+1]
073F:010D add cx, [bx+2]
073F:0110
-e 2000:1000 BE 00 06 00 00 00
```

```
-t

AX=2000  BX=1000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=2000  ES=073F  SS=073F  CS=073F  IP=0108   NV UP EI PL NZ NA PO NC
073F:0108 8B07          MOV     AX,[BX]                         DS:1000=00BE
-t

AX=00BE  BX=1000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=2000  ES=073F  SS=073F  CS=073F  IP=010A   NV UP EI PL NZ NA PO NC
073F:010A 8B4F01        MOV     CX,[BX+01]                      DS:1001=0600
-t

AX=00BE  BX=1000  CX=0600  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=2000  ES=073F  SS=073F  CS=073F  IP=010D   NV UP EI PL NZ NA PO NC
073F:010D 034F02        ADD     CX,[BX+02]                      DS:1002=0006
-t

AX=00BE  BX=1000  CX=0606  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=2000  ES=073F  SS=073F  CS=073F  IP=0110   NV UP EI PL NZ NA PE NC
073F:0110 00F0          ADD     AL,DH
```

应用：用[bx+idata]的方式进行数组的处理

- 问题：在codesg中填写代码，将datasg中定义的
  - 第一个字符串，转化为大写
  - 第二个字符串转化为小写。

```
assume cs:codesg,ds:datasg
datasg segment
    db 'BaSiC'
    db 'MinIX'
datasg ends
codesg segment
  start:
    ……
codesg ends
end start
```

是否可以对两个同长度的字符串“同步”操作？

```
    mov ax,datasg
    mov ds,ax
    mov bx,0
    mov cx,5
 s: mov al,[bx]
    and al,11011111b
    mov [bx],al
    inc bx
    loop s
    mov bx,5
    mov cx,5
 s0: mov al,[bx]
    or al,00100000b
    mov [bx],al
    inc bx
    loop s0
```

```
mov ax,datasg
    mov ds,ax
    mov bx,0
    mov cx,5
s: mov al,[bx]
    and al,11011111b
    mov [bx],al
    mov al,[5+bx]
    or al,00100000b
    mov [5+bx],al
    inc bx
    loop s
```

在Debug中执行

```
1  assume cs:codesg,ds:datasg
2  datasg segment
3          db 'BaSiC'
4          db 'MinIX'
5  datasg ends
6  codesg segment
7  start: mov ax,datasg
8          mov ds,ax
9
10         mov bx,0
11         mov cx,5
12 s:      mov al,[bx]
13         and al,11011111b
14         mov [bx],al
15
16         mov al,[5+bx]
17         or al,00100000b
18         mov [5+bx],al
19         inc bx
20         loop s
21
22         mov ax, 4c00h
23         int 21h
24 codesg ends
25 end start
```

```
C:\>debug p7-3.exe
-r
AX=FFFF  BX=0000  CX=0031  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076B  IP=0000   NV UP EI PL NZ NA PO NC
076B:0000 B86A07        MOV     AX,076A
-d 076a:0 f
076A:0000  42 61 53 69 43 4D 69 6E-49 58 00 00 00 00 00 00   BaSiCMinIX......
-g

Program terminated normally
-d 076a:0 f
076A:0000  42 41 53 49 43 6D 69 6E-69 78 00 00 00 00 00 00   BASICminix......
-
```

```
char a[5]="BaSiC";
char b[5]=""MinIX";
main(){
   int i;
   i=0;
   do{
      a[i]=a[i]&0xDF;
      b[i]=b[i]|0x20;
      i++;
   }
   while(i<5);
}
```

味道有点像！

[bx+idata]的方式为高级语言实现数组提供了便利机制。