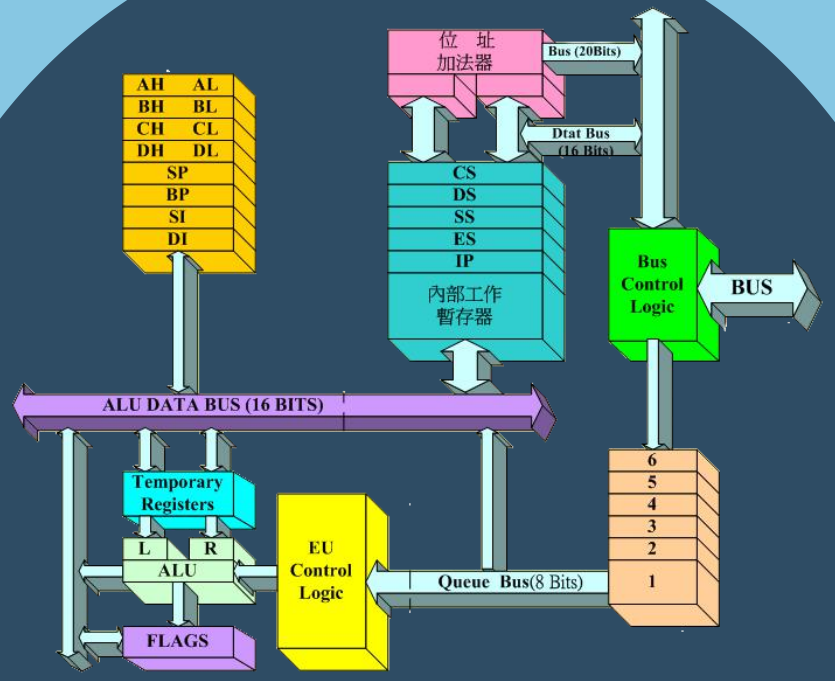
在哪里？有多长？

贺利坚　主讲



汇编语言程序设计

Assembly Language

两个基本问题


```
mov ax, 0
mov ax, [0]
mov ax, [di]
mov ax, [bx+8]
mov ax, [bx+si]
mov ax, [bx+si+8]
mov ax, [bp]
mov ax, [bp+8]
mov ax, [bp+si]
mov ax, [bp+si+8]
…
```

（1）处理的数据在什么地方？

（2）要处理的数据有多长？

汇编语言中数据位置的表达

1、立即数（idata）

对于直接包含在机器指令中的数据,称为立即数（idata），数据包含在指令中

```
mov ax,1
add bx,2000h
or bx,00010000b
mov al,'a'
```

```
-a
073F:0100 mov ax, 1
073F:0103
-u 073f:0100
073F:0100 B80100    MOV     AX,0001
```

2、寄存器

指令要处理的数据在寄存器中，在汇编指令中给出相应的寄存器名。

```
mov ax,bx
mov ds,ax
push bx
mov ds:[0],bx
push ds
mov ss,ax
mov sp,ax
```

3、内存：段地址（SA）和偏移地址（EA）

指令要处理的数据在内存中，由SA:EA确定内存单元。

```
mov ax,[0]
mov ax,[di]
mov ax,[bx+8]
mov ax,[bx+si]
mov ax,[bx+si+8]
```

段地址默认在ds中

```
mov ax,[bp]
mov ax,[bp+8]
mov ax,[bp+si]
mov ax,[bp+si+8]
```

段地址默认在ss中

```
mov ax,ds:[bp]          ：(ax)=((ds)*16+(bp))
mov ax,es:[bx]          ：(ax)=((es)*16+(bx))
mov ax,ss:[bx+si]       ：(ax)=((ss)*16+(bx)+(si))
mov ax,cs:[bx+si+8]     ：(ax)=((cs)*16+(bx)+(si)+8)
```

显性的给出存放段地址的寄存器

指令要处理的数据有多长?

字word操作	字节byte操作	用word ptr或byte ptr指明

mov ax,1
mov bx,ds:[0]
mov ds,ax
mov ds:[0],ax
inc ax
add ax,1000
	mov al,1
	mov al,bl
	mov al,ds:[0]
	mov ds:[0],al
	inc al
	add al,100
		mov word ptr ds:[0],1
		inc word ptr [bx]
		inc word ptr ds:[0]
		add word ptr [bx],2
		
		mov byte ptr ds:[0],1
		inc byte ptr [bx]
		inc byte ptr ds:[0]
		add byte ptr [bx],2
		
		在没有寄存器参与的内存单元访问指令中，用word ptr或byte ptr显性地指明所要访问的内存单元的长度是很必要的，否则，CPU无法得知所要访问的单元是字单元，还是字节单元。

```
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=2000  ES=073F  SS=073F  CS=073F  IP=0100   NV UP EI PL NZ NA PO NC
073F:0100 B80020        MOV     AX,2000
-a 073f:0100
073F:0100 mov byte ptr [1000], 1
073F:0105
-e 2000:1000 FF FF FF FF FF FF FF FF
-T

AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=2000  ES=073F  SS=073F  CS=073F  IP=0105   NV UP EI PL NZ NA PO NC
073F:0105 C606001001    MOV     BYTE PTR [1000],01                 DS:10
-d 2000:1000
2000:1000  01 FF FF FF FF FF FF FF-00 00 00 00 00 00 00 00   ..........
```

```
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=2000  ES=073F  SS=073F  CS=073F  IP=0100   NV UP EI PL NZ NA PO NC
073F:0100 C70600100100  MOV     WORD PTR [1000],0001               DS:10
-a 073f:0100
073F:0100 mov word ptr [1000], 1
073F:0106
-e 2000:1000 FF FF FF FF FF FF FF FF
-t

AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=2000  ES=073F  SS=073F  CS=073F  IP=0106   NV UP EI PL NZ NA PO NC
073F:0106 06            PUSH    ES
-d 2000:1000
2000:1000  01 00 FF FF FF FF FF FF-00 00 00 00 00 00 00 00   ..........
```