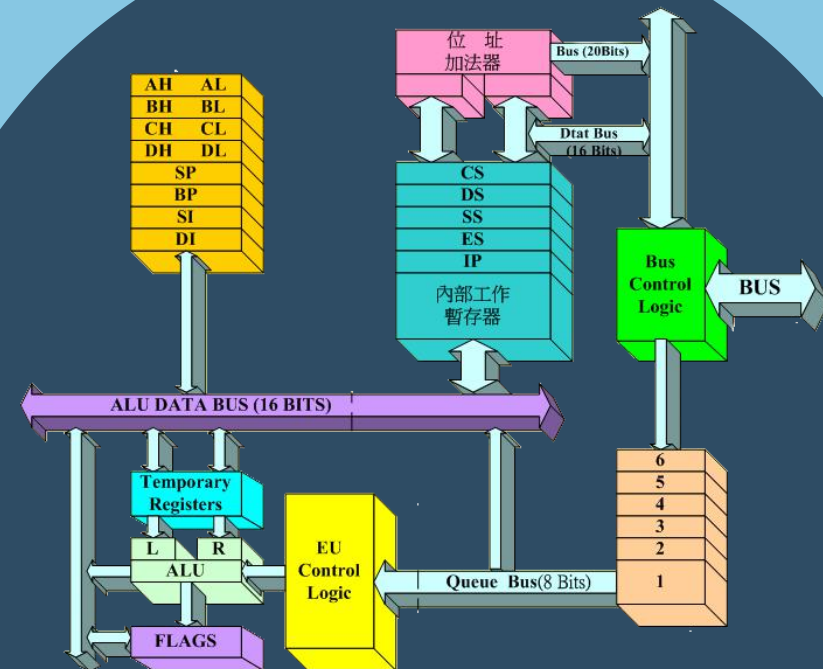


[...]和(...)

贺利坚 主讲

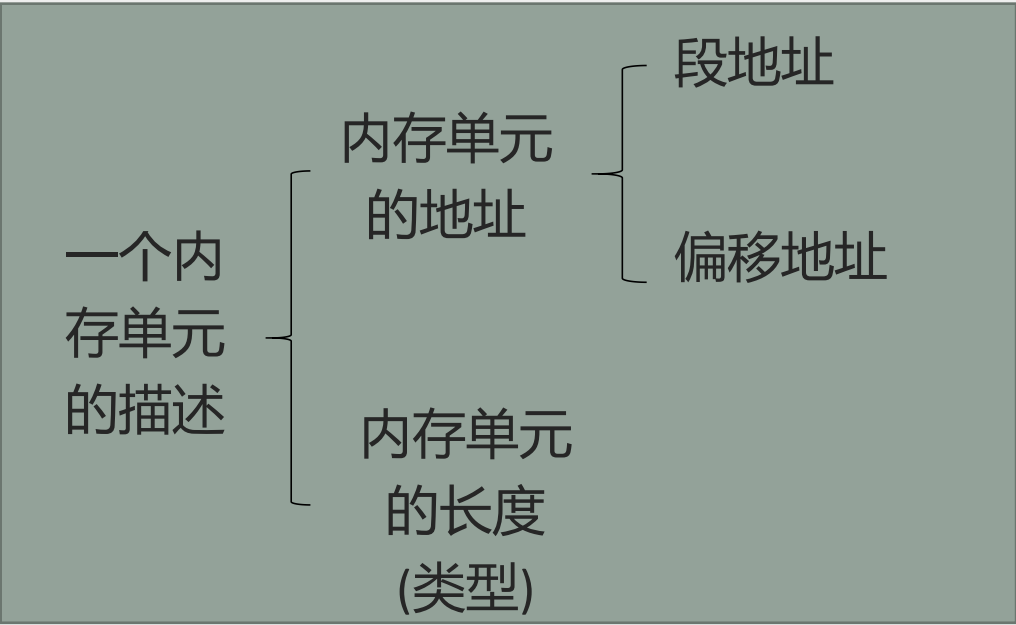



汇编语言程序设计
Assembly Language

[...]的规定与(...)的约定

 [...]——(汇编语法规则)表示一个内存单元

指令	段地址	偏移地址	操作单位
mov ax, [0]	在DS中	在[0]中	字
mov al, [0]	在DS中	在[0]中	字节
mov ax,[bx]	在DS中	在[bx]中	字
mov al,[bx]	在DS中	在[bx]中	字节



 (...)——(为学习方便做出的约定)表示一个内存单元或寄存器中的内容

只能用寄存器
及物理地址

描述对象	描述方法	描述对象	描述方法
ax中的内容为0010H	(ax)=0010H	2000:1000 处的内容为0010H	(21000H)=0010H
mov ax,[2]的功能	(ax)=((ds)*16+2)	mov [2], ax的功能	((ds)*16+2)=(ax)
add ax,2 的功能	(ax)=(ax)+2	add ax,bx的功能	(ax)=(ax)+(bx)
push ax的功能	(sp) = (sp)-2 ((ss)*16 + (sp))=(ax)	pop ax 的功能	(ax)=((ss)*16+(sp)) (sp)=(sp)+2

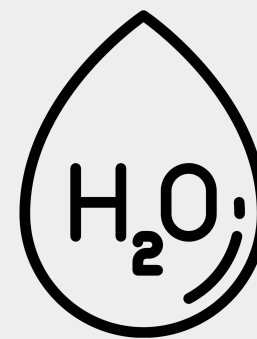
再约定：符号idata表示常量

🖥️例：

📁mov ax,[idata] : 代表mov ax,[1]、 mov ax,[2]、 mov ax,[3]...

📁mov bx,idata : 代表mov bx,1、 mov bx,2、 mov bx,3...

📁mov ds,idata : 代表mov ds,1、 mov ds,2...(都是非法指令)



案例分析

```
mov ax,2000H
mov ds,ax
mov bx,1000H
mov ax,[bx]
inc bx
inc bx
mov [bx],ax
inc bx
inc bx
mov [bx],ax
inc bx
mov [bx],al
inc bx
mov [bx],al
```

AX=	BX=
DS=	ES=

```
mov ax,[bx] --- (ax)=((ds) *16 +(bx))
mov [bx],ax --- ((ds)*16 +(bx))=(ax)
```

BE	21000H
00	21001H
	21002H
	21003H
	21004H
	21005H
	21006H
	21007H