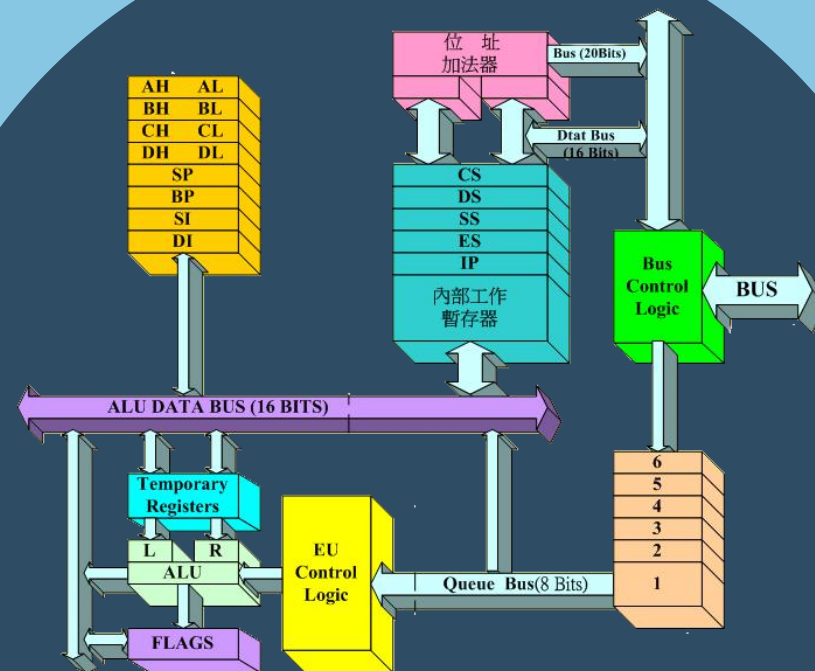


用DS和[address]实现字的传送

贺利坚 主讲



汇编语言程序设计
Assembly Language

要解决的问题：CPU从内存单元中要读取数据

📁 要求

📁 CPU要读取一个内存单元的时候，必须先给出这个内存单元的地址；

📁 原理

📁 在8086PC中，内存地址由段地址和偏移地址组成（段地址:偏移地址）

📁 解决方案：DS和[address]配合

📁 用 DS寄存器存放要访问的数据的段地址

📁 偏移地址用[...]形式直接给出

📁 例1

```
mov bx,1000H
mov ds,bx
mov al, [0]
```

将10000H(1000:0)
中的数据读到al中

📁 例2

```
mov bx,1000H
mov ds,bx
mov [0],al
```

将al中的数据写到
10000H(1000:0)中

📁 将段地址送入DS的两种方式

(1) `mov ds, 1000H`



(2) `mov bx, 1000H`

`mov ds, bx`



📁 8086CPU不支持将数据直接送入段寄存器
（硬件设计的问题）

📁 套路：数据 → 一般的寄存器 → 段寄存器

字的传送

 8086CPU可以一次性传送一个字(16位的数据)

 例

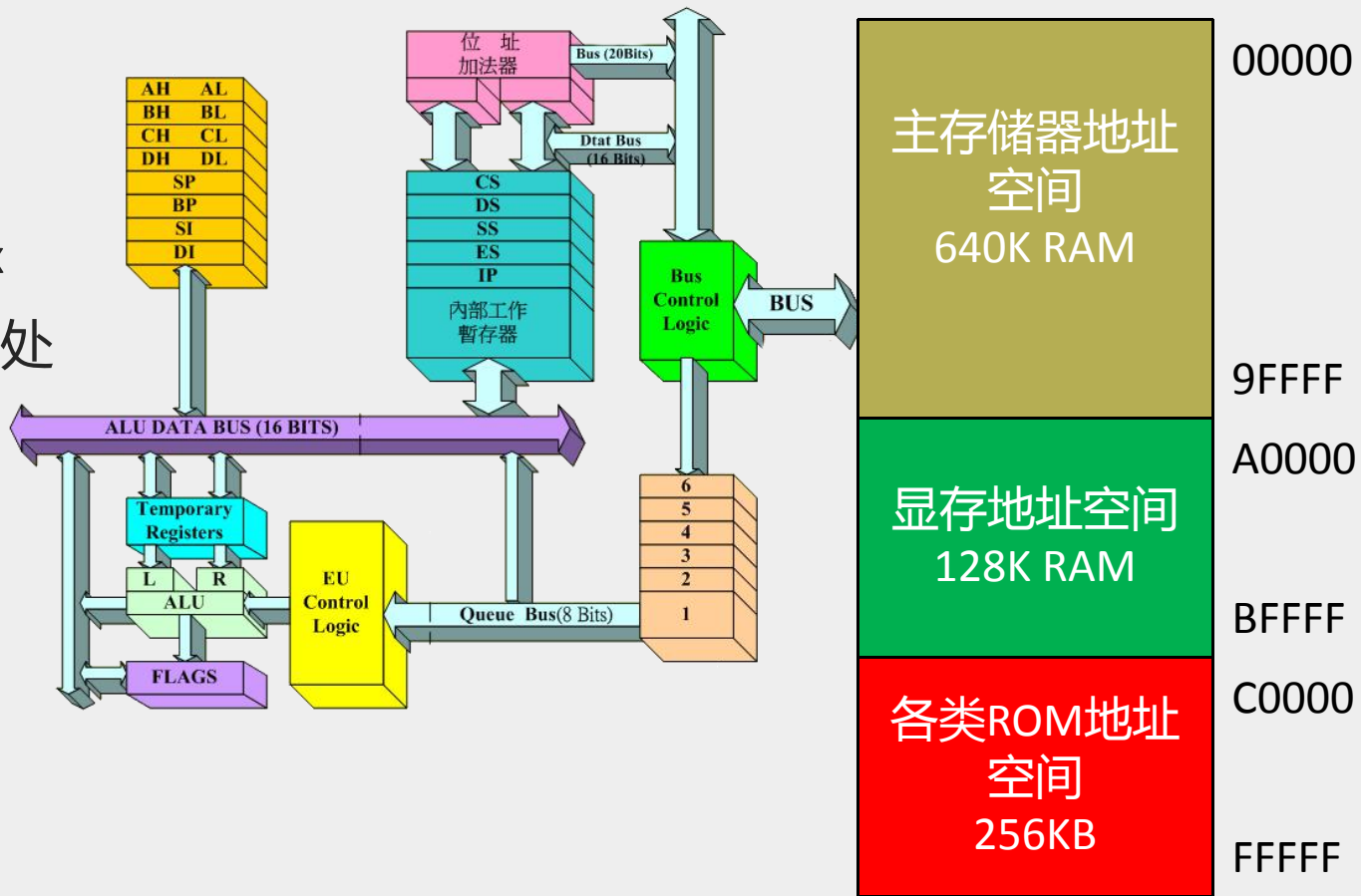
```
mov bx, 1000H
```

```
mov ds, bx
```

mov ax, [0] ;1000:0处的字型数据送入ax

mov [0], cx ;cx中的16位数据送到1000:0处

10000H	23
10001H	11
10002H	22
10003H	66



案例1



内存

10000H	23
10001H	11
10002H	22
10003H	66



指令

```
mov ax, 1000H
mov ds, ax
mov ax, [0]
mov bx, [2]
mov cx, [1]
add bx, [1]
add cx, [2]
```

AX=	BX=	CX=	DX=	SP=	BP=	SI=	DI=					
DS=	ES=	SS=	CS=	IP=	NV	UP	EI	NG	NZ	NA	PO	NC

案例2

内存

10000H	23
10001H	11
10002H	22
10003H	11

指令

```
mov ax, 1000H
mov ds, ax
mov ax, 2C31
mov [0], ax
mov bx, [0]
sub bx, [2]
mov [2], bx
```

AX=	BX=	CX=	DX=	SP=	BP=	SI=	DI=					
DS=	ES=	SS=	CS=	IP=	NV	UP	EI	NG	NZ	NA	PO	NC