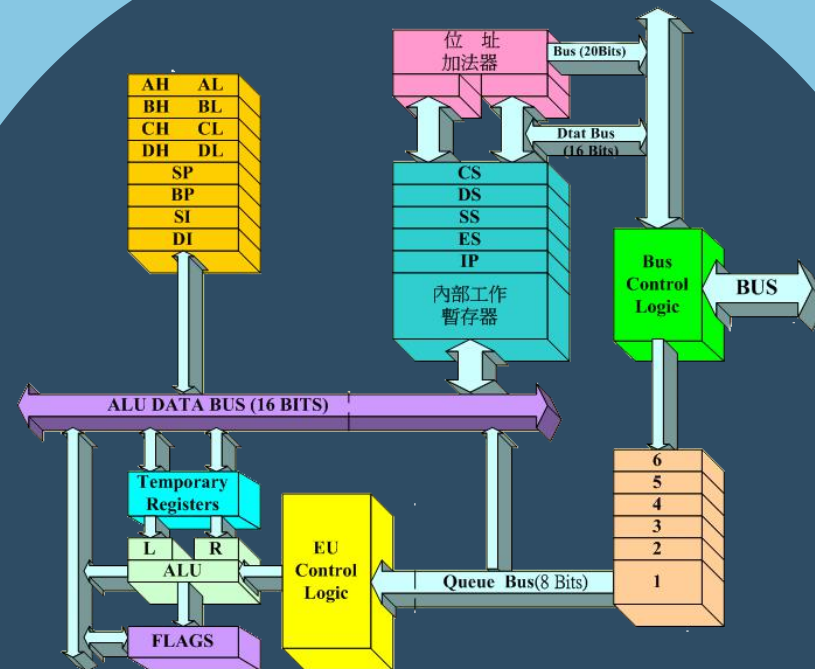


SI和DI寄存器

贺利坚 主讲



汇编语言程序设计
Assembly Language

CPU内部的寄存器

8086CPU有14个寄存器：

通用寄存器：AX、BX、CX、DX

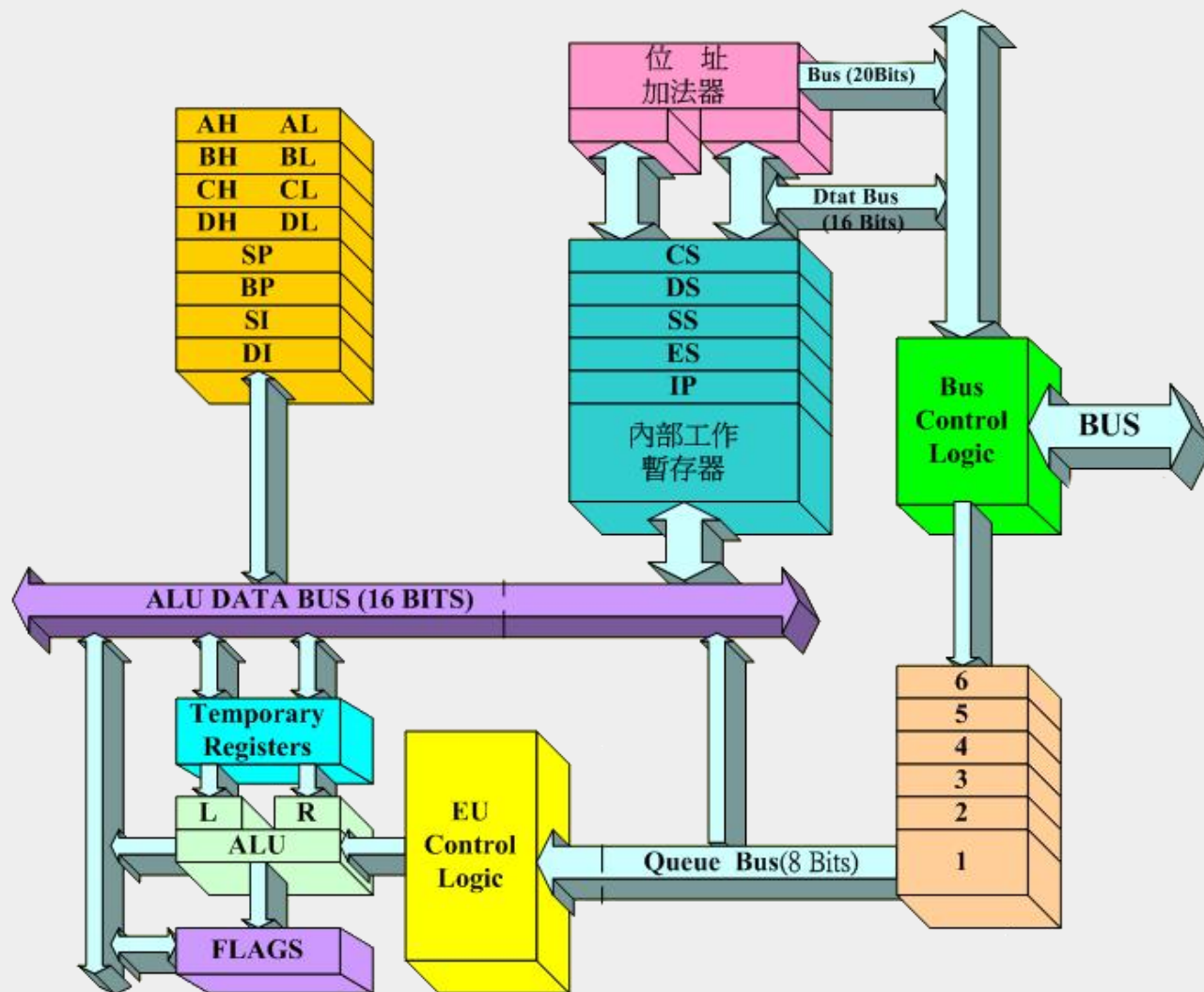
变址寄存器：SI、DI

指针寄存器：SP、BP

指令指针寄存器：IP

段寄存器：CS、SS、DS、ES

标志寄存器：PSW



SI和DI常执行与地址有关的操作

💻 SI和DI是8086CPU中和BX功能相近的寄存器

📁 区别：SI和DI不能够分成两个8位寄存器来使用。

💻 下面的三组指令实现了相同的功能：

(1) mov bx,0

mov ax,[bx]

(2) mov si,0

mov ax,[si]

(3) mov di,0

mov ax,[di]

(1) mov bx,0


mov ax,[bx+123]

(2) mov si,0

mov ax,[si+123]

(3) mov di,0

mov ax,[di+123]



💻 BX：通用寄存器，在计算存储器地址时，常作为基址寄存器用

💻 SI：source index，源变址寄存器

💻 DI：destination index，目标变址寄存器

总会有什么不同吧？



应用SI和DI

问题

用寄存器SI和DI实现将字符串 'welcome to masm!' 复制到它后面的数据区中。

程序定义

```
assume cs:codesg,ds:datasg
datasg segment
    db 'welcome to masm!'
    db '.....'
datasg ends
codesg segment
....
codesg ends
end
```

源数据起始地址：datasg:0

目标数据起始地址：datasg:16

用ds:si 指向要复制的原始字符串

用 ds:di 指向目的空间

然后用一个循环来完成复制。

```
1  assume cs:codesg,ds:datasg
2  datasg segment
3      db 'welcome to masm!'
4      db '.....'
5  datasg ends
6  codesg segment
7  start: mov ax,datasg
8          mov ds,ax
9
10         mov si,0
11         mov di,16
12         mov cx,8
13 s:      mov ax,[si]
14         mov [di],ax
15         add si,2
16         add di,2
17         loop s
18
19         mov ax,4c00h
20         int 21h
21 codesg ends
22 end start
```

程序运行

```
1  assume cs:codesg,ds:datasg
2  datasg segment
3      db 'welcome to masn!'
4      db '.....'
5  datasg ends
6  codesg segment
7  start: mov ax,datasg
8          mov ds,ax
9
10         mov si,0
11         mov di,16
12         mov cx,8
13  s:     mov ax,[si]
14         mov [di],ax
15         add si,2
16         add di,2
17         loop s
18
19         mov ax,4c00h
20         int 21h
21  codesg ends
22  end start
```

```
-d 076A:0 1f
076A:0000  77 65 6C 63 6F 6D 65 20-74 6F 20 6D 61 73 6D 21  welcome to masn!
076A:0010  2E 2E 2E 2E 2E 2E 2E 2E-2E 2E 2E 2E 2E 2E 2E  welcome to masn!
.....
-u
076C:0000 B86A07      MOV     AX,076A
076C:0003 8ED8      MOV     DS,AX
076C:0005 BE0000      MOV     SI,0000
076C:0008 BF1000      MOV     DI,0010
076C:000B B90800      MOV     CX,0008
076C:000E 8B04      MOV     AX,[SI]
076C:0010 8905      MOV     [DI],AX
076C:0012 83C602      ADD     SI,+02
076C:0015 83C702      ADD     DI,+02
076C:0018 E2F4      LOOP    000E
076C:001A B8004C      MOV     AX,4C00
076C:001D CD21      INT     21
076C:001F 0C00      OR      AL,00
-g

Program terminated normally
-d 076A:0 1f
076A:0000  77 65 6C 63 6F 6D 65 20-74 6F 20 6D 61 73 6D 21  welcome to masn!
076A:0010  77 65 6C 63 6F 6D 65 20-74 6F 20 6D 61 73 6D 21  welcome to masn!
```

程序还可以写作——

```
1  assume cs:codesg,ds:datasg
2  datasg segment
3      db 'welcome to masm!'
4      db '.....'
5  datasg ends
6  codesg segment
7  start: mov ax,datasg
8          mov ds,ax
9
10         mov si,0
11         mov di,16
12         mov cx,8
13  s:     mov ax,[si]
14         mov [di],ax
15         add si,2
16         add di,2
17         loop s
18
19         mov ax,4c00h
20         int 21h
21 codesg ends
22 end start
```



```
1  assume cs:codesg,ds:datasg
2  datasg segment
3      db 'welcome to masm!'
4      db '.....'
5  datasg ends
6  codesg segment
7  start: mov ax,datasg
8          mov ds,ax
9
10         mov si,0
11         mov cx,8
12  s:     mov ax,0[si]
13         mov 16[si],ax
14         add si,2
15         loop s
16
17         mov ax,4c00h
18         int 21h
19 codesg ends
20 end start
```

[bx+idata]形式