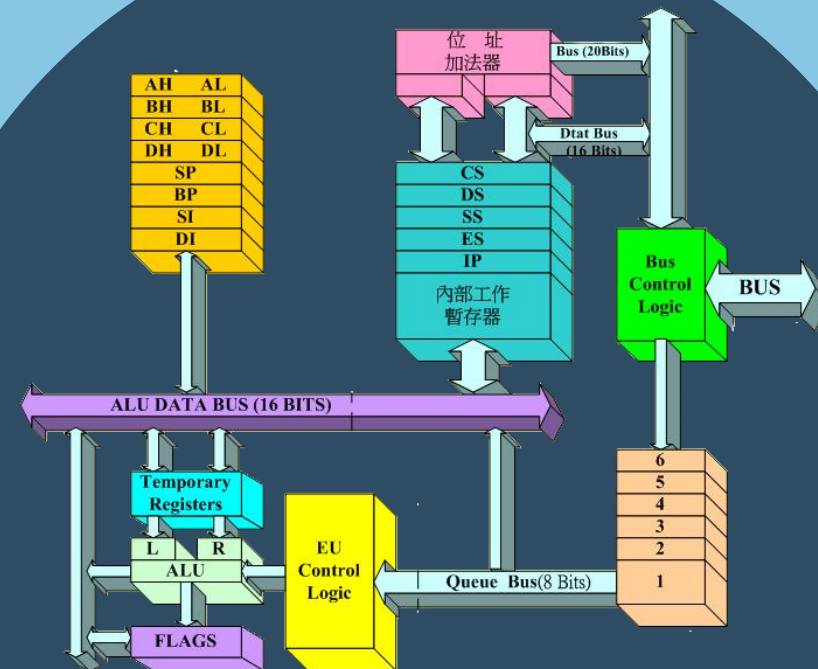


移位指令

贺利坚 主讲



汇编语言程序设计
Assembly Language

移位指令

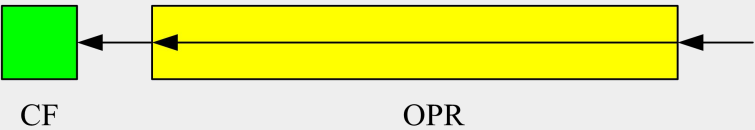
```
mov al,01001000b
shl al,1
```

0
CF

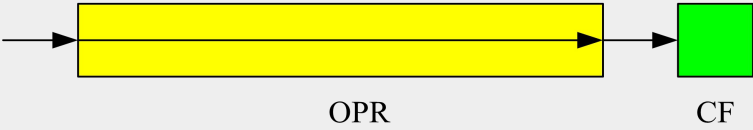
0	1	0	0	1	0	0	0
1	0	0	1	0	0	0	0

结果：(al)=10010000b ,
CF=0

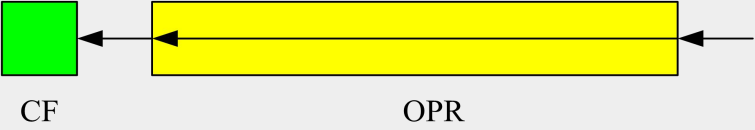
逻辑左移 SHL OPR, CNT



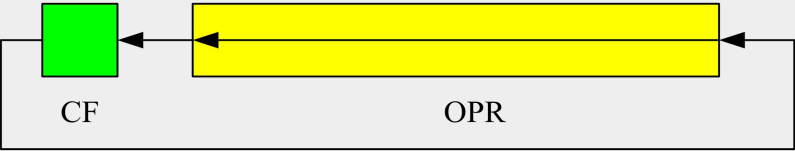
逻辑右移 SHR OPR, CNT



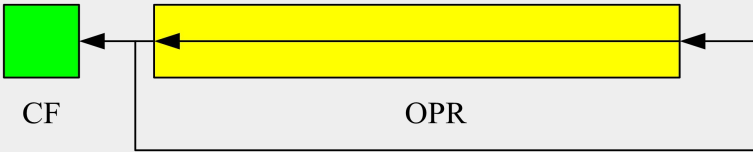
算术左移 SAL OPR, CNT



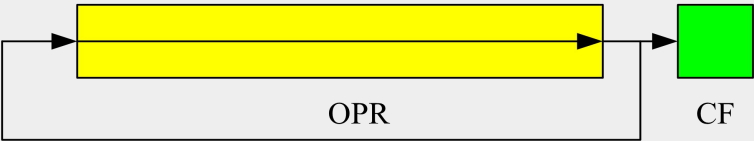
带进位循环左移 RCL OPR, CNT



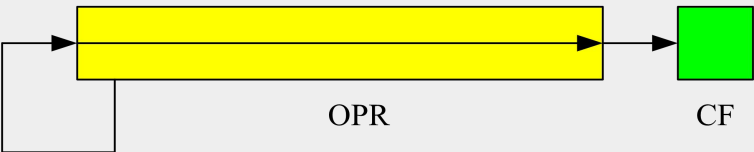
循环左移 ROL OPR, CNT



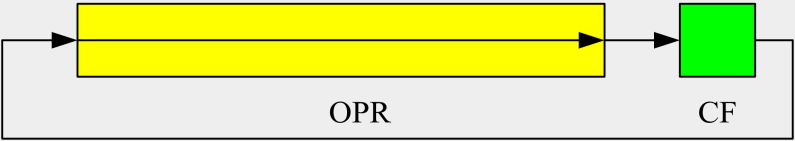
循环右移 ROR OPR, CNT



算术右移 SAR OPR, CNT



带进位循环右移 RCR OPR, CNT



S,SH-Shift L-Left R-Right A-Arithmetic R,RO-Rotate C-Carry

示例：逻辑移位指令shl和shr

 SHL OPR, CNT，将OPR逻辑左移CNT位

- (1) 将寄存器或内存单元中的数据向左移位
- (2) 将最后移出的一位写入CF中
- (3) 最低位用0补充

 shl指令操作示例

```
mov al,01010001b
```


```
mov cl,3
```

```
shl al,cl
```

移动位数
大于1时，
必须用cl

结果：(al)=10001000b，
CF=0

	0	1	0	1	0	0	0	1
CF	1	0	0	0	1	0	0	0

 惯常用法：将X逻辑左移一位，相当于执行 $X=X*2$ ；右移一位，相当于执行 $X=X/2$

```
mov al, 00000001b ;执行后(al)=00000001=1
```

```
shl al, 1 ;执行后(al)=00000010=2
```

```
shl al, 1 ;执行后(al)=00000100=4
```

```
shl al, 1 ;执行后(al)=00001000=8
```

```
mov cl, 3
```

```
shl al, cl ;执行后(al)=01000000=64
```

```
mov cl, 2
```

```
shr al, cl ;执行后(al)=00010000=16
```