移位实现乘法程序

移位操作

- 移位是以位为单位将数据向左或向右的移动
- 移位操作是最基本的数据处理方法
  - 左移一位相当于数值乘以2
  - 右移一位相当于数值除以2（余数在CF中）
    - 逻辑右移一位是无符号数除以2
    - 算术右移一位是有符号数除以2
- 对数据加减、配合移位操作可以实现乘除法运算

移位可以实现乘法

- 两个整数相乘能用移位、配合加减运算实现
  - 可以提高乘法操作的速度，实现代码优化

$$x \times y = x \times \sum_{k=0}^{n} a_k \times 2^k = \sum_{k=0}^{n} (x \times a_k \times 2^k) \quad , \quad a_k = 1 \text{ or } 0$$

a_k=1，对x左移k位

$x \times 10 = x \times 2^3 + x \times 2^1$

10x = x左移1位+x左移3位

16位数据零位扩展为32位

```
        ;数据段
wvar    word 34000
        ;代码段
        xor eax,eax        ;EAX＝0
        mov ax,wvar        ;AX＝要乘以10的无符号数
```

将16位数据进行零位扩展为32位

32位数据乘10

```
shl eax,1
mov ebx,eax
shl eax,2
add eax,ebx
```

```
mul eax,10
```

;左移1位等于乘2

;EBX＝EAX×2

;再左移2位，EAX＝EAX×8

;EAX＝EAX×10

$$x \times 10 = x \times 2^3 + x \times 2^1$$

十进制数据显示

```
call dispuid        ;显示乘积
call dispcrlf       ;换行，移动到下一行首列位置
mul eax,10          ;EAX＝EAX×10
call dispuid        ;显示乘积
```

子程序名	DISPUID
入口参数	EAX = 32位数据
功能说明	显示无符号十进制整数

子程序名	DISPCRLF
入口参数	无
功能说明	光标回车换行

移位指令实现乘法程序

```
        ;数据段
wvar    word 34000
        ;代码段
        xor eax,eax
        mov ax,wvar
        shl eax,1
        mov ebx,eax
        shl eax,2
        add eax,ebx
        call dispuid
        call dispcrlf
        mul eax,10
        call dispuid
```

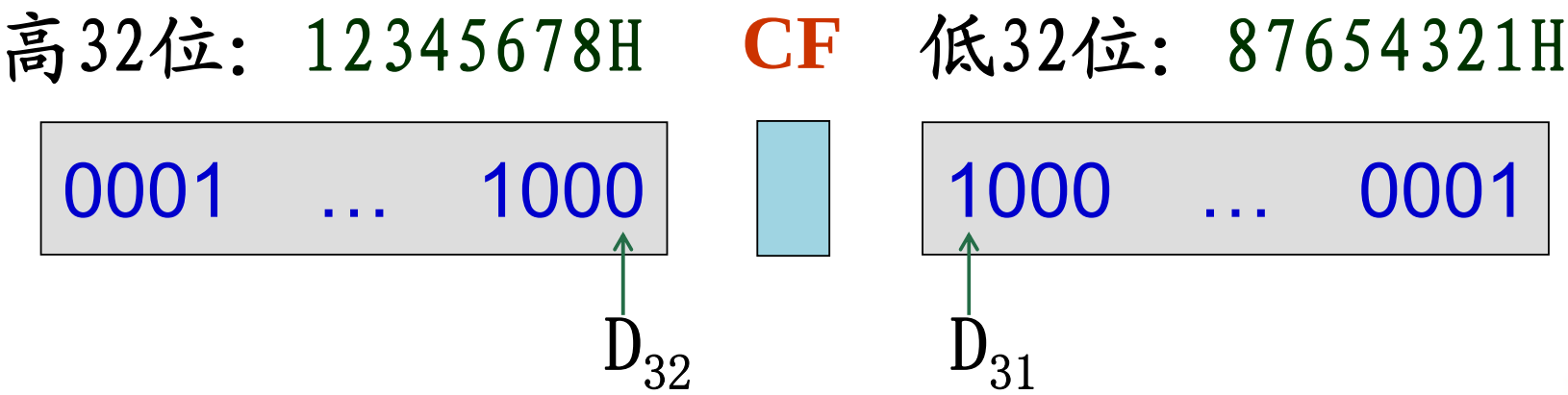
运行结果

```
340000
3400000
```

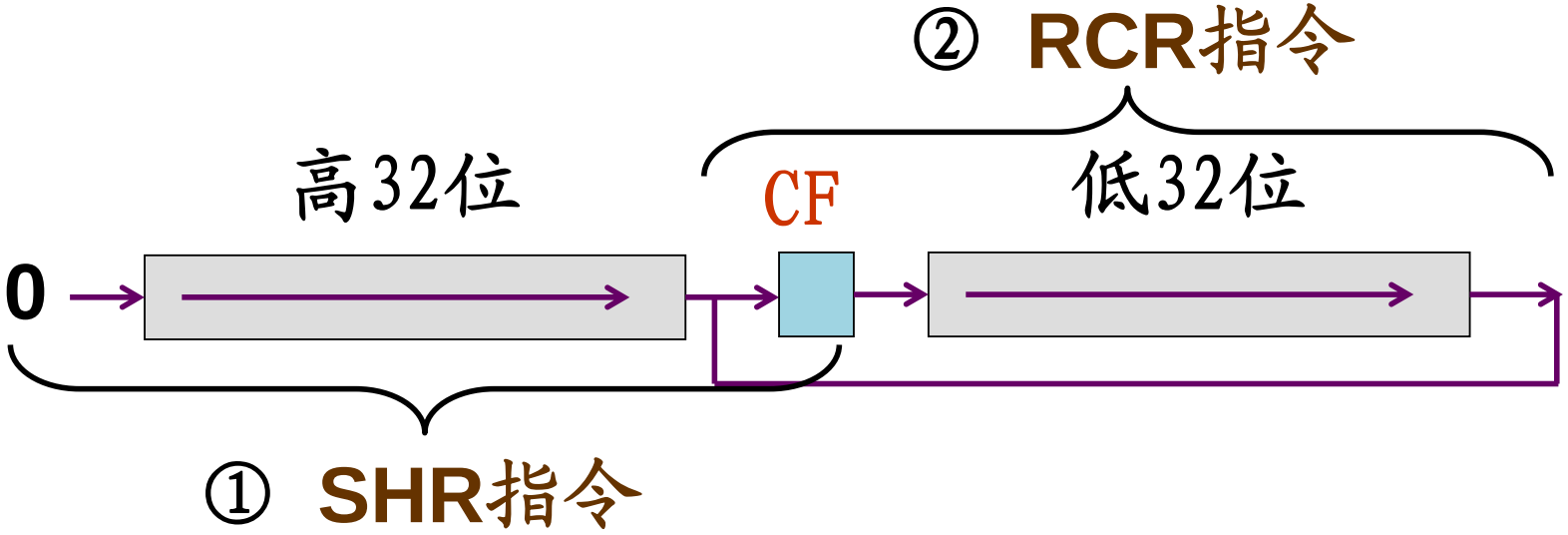
64位数据移位程序

64位数据的移位操作

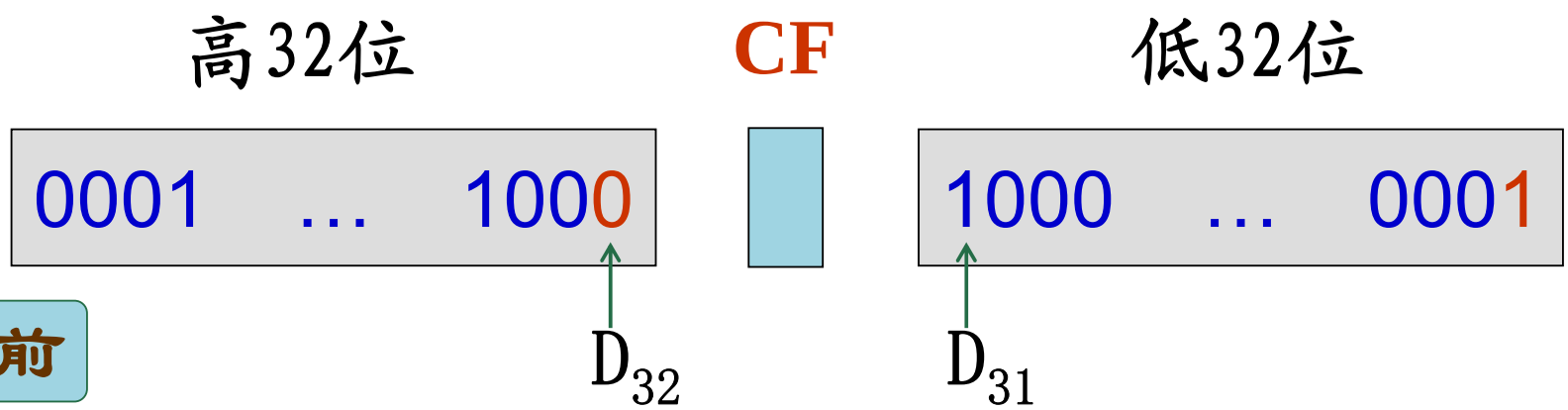
- IA-32支持8、16和32位数据的各种移位操作
- 对于64位数据需要分成高32位、低32位分别移位
  - 高低32位中间的D_{32}和D_{31}位，需要利用CF衔接
  - 不仅要使用移位指令（SHL、SHR和SAR）
  - 必然要使用带进位的循环移位指令（RCL或RCR）



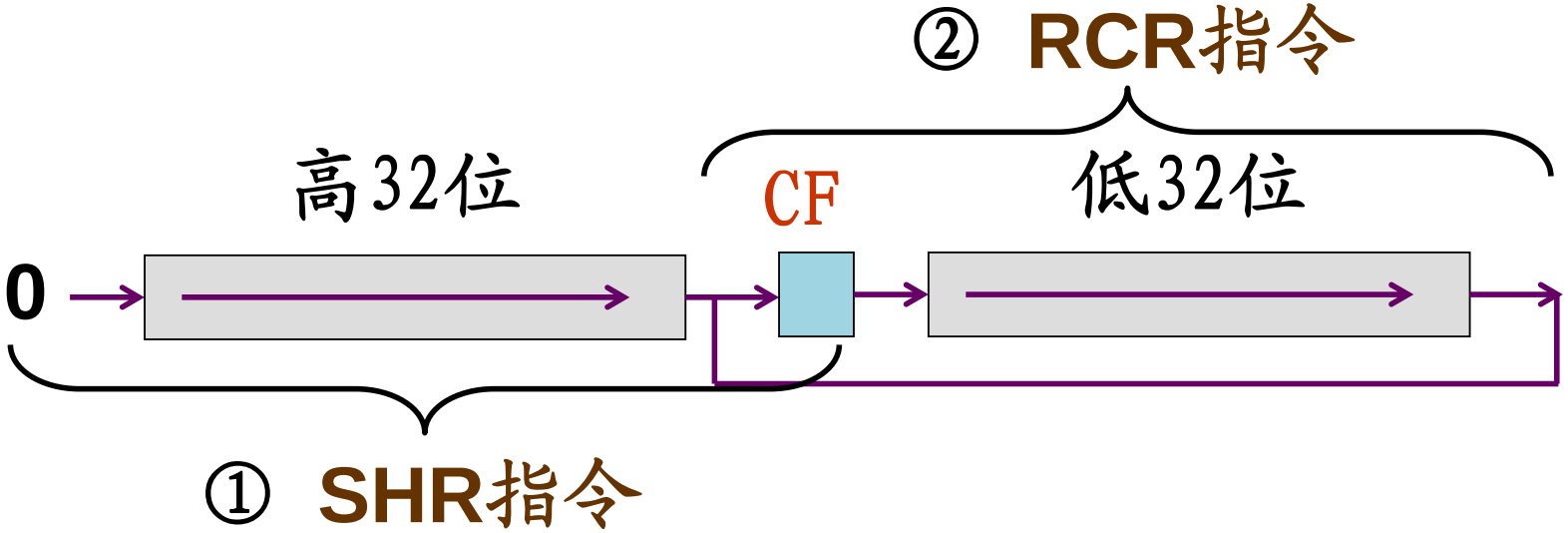
64位数据的逻辑右移

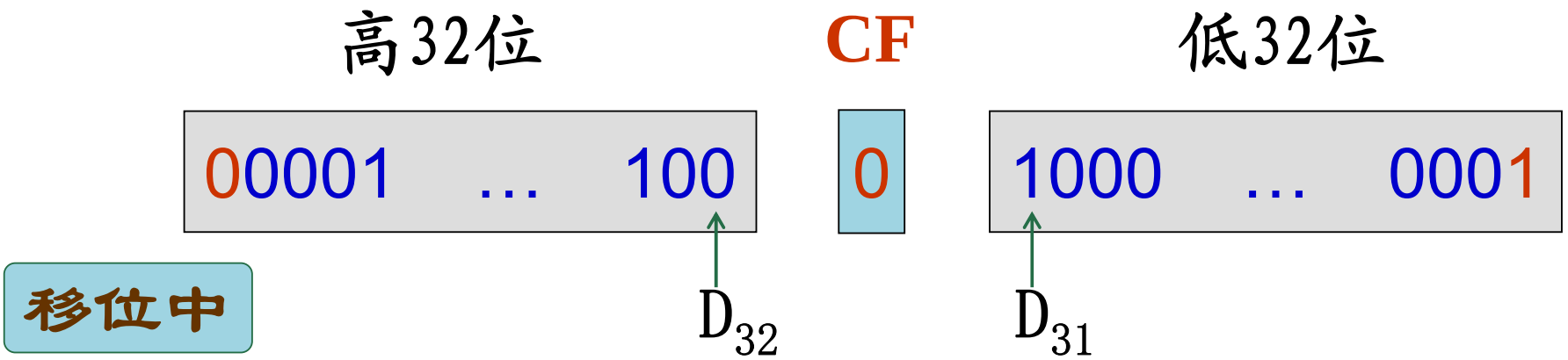


移位前

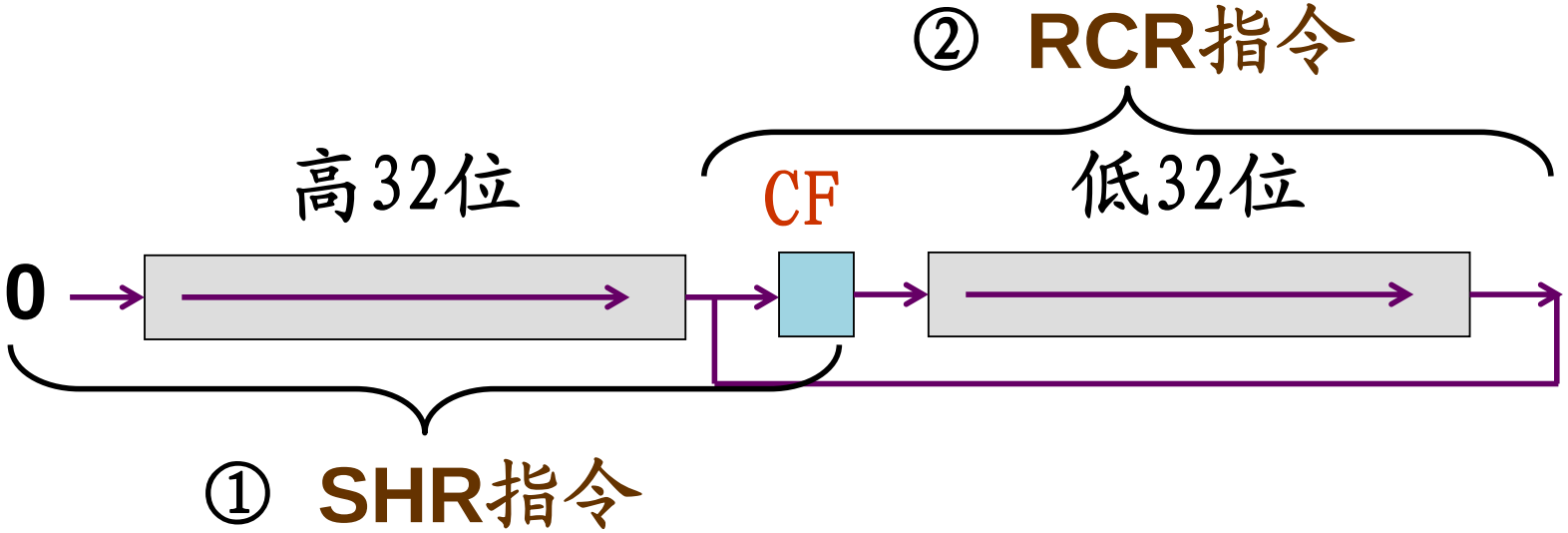


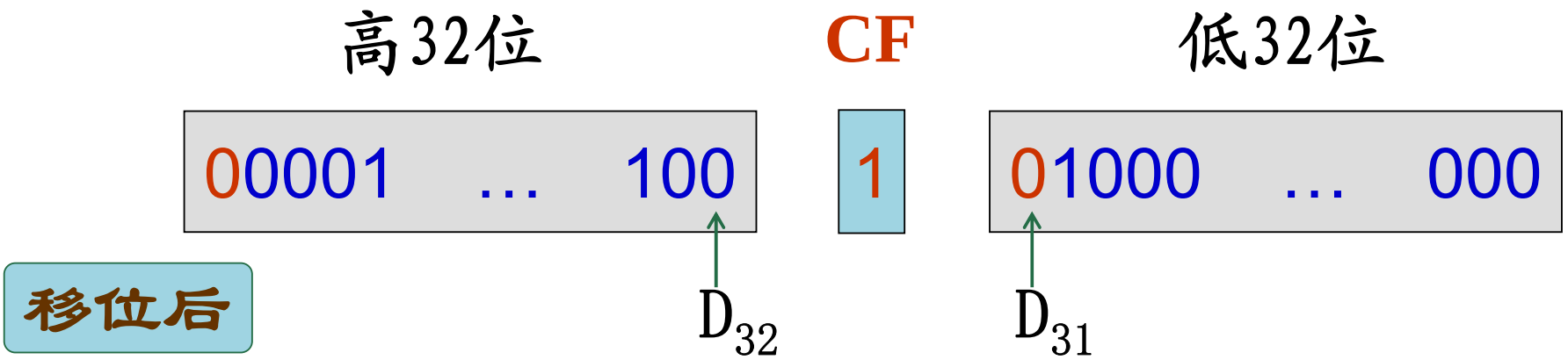
64位数据的逻辑右移（执行SHR指令）





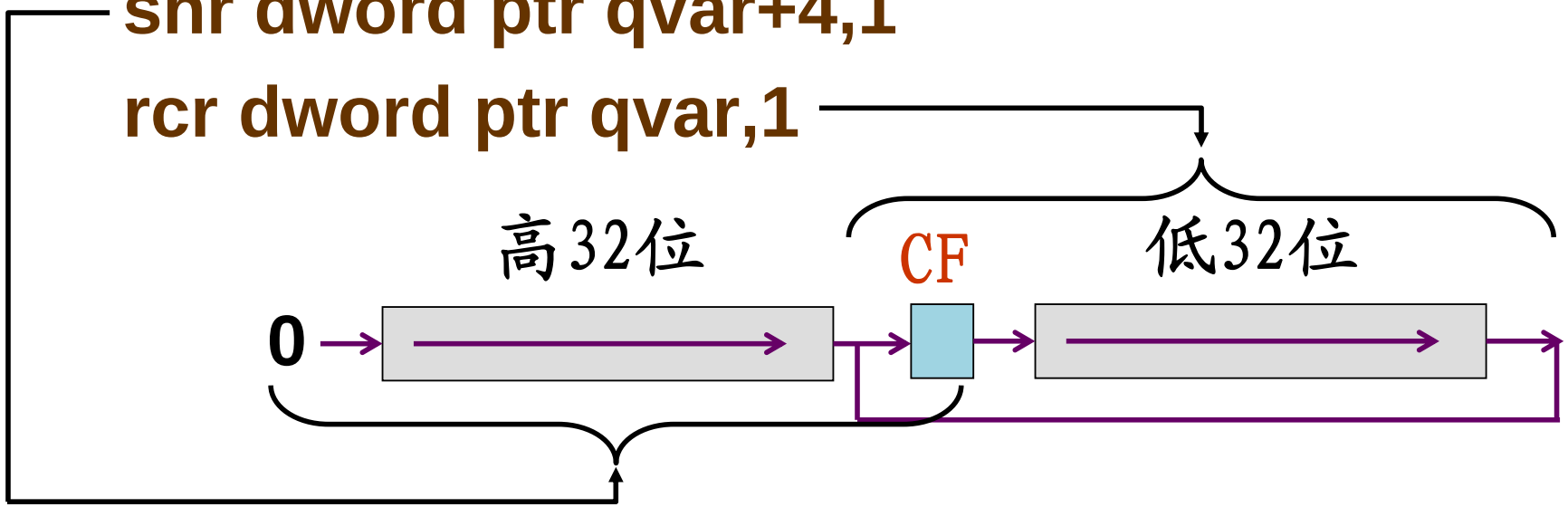
64位数据的逻辑右移（执行RCR指令）



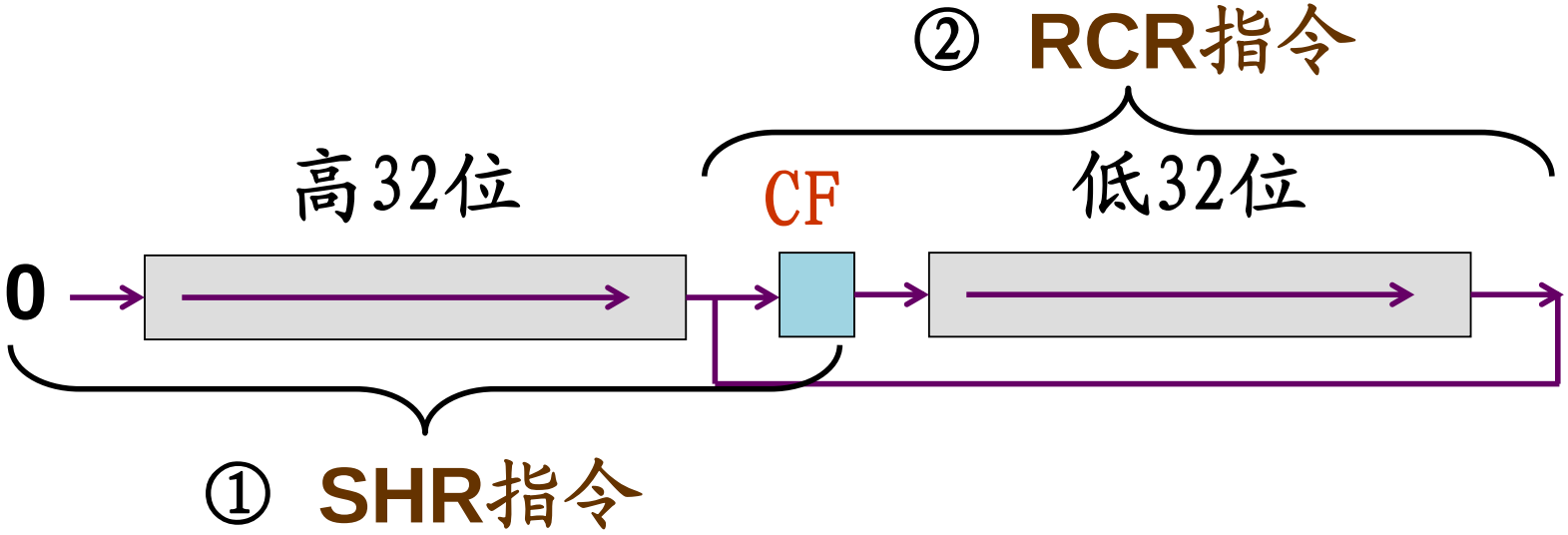


64位数据逻辑右移程序

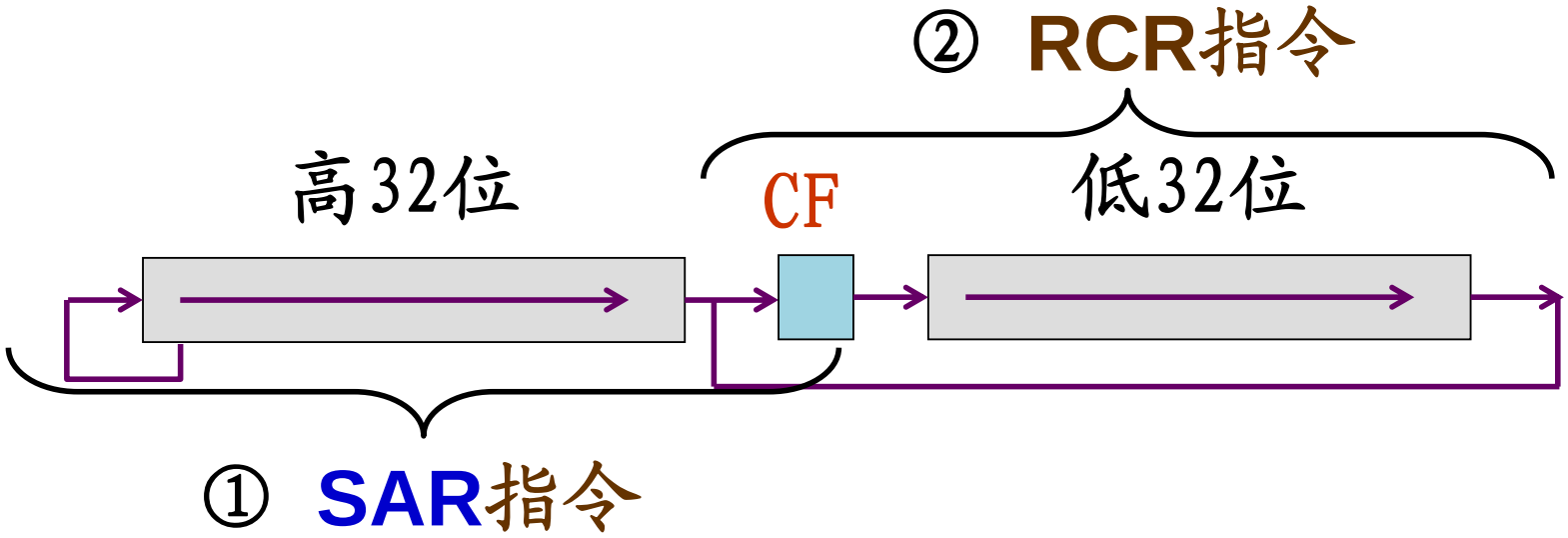
```
        ;数据段
qvar    qword 1234567887654321h
        ;代码段
        shr dword ptr qvar+4,1
        rcr dword ptr qvar,1
```



64位数据的逻辑右移



64位数据的算术右移



64位数据的逻辑（算术）左移