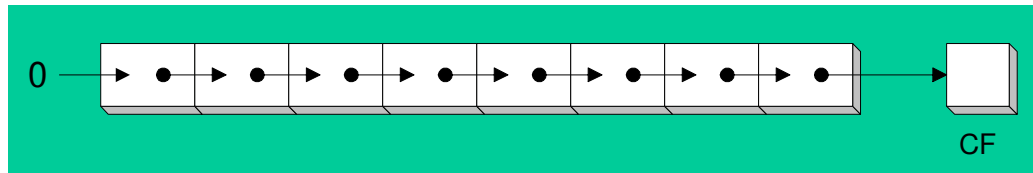
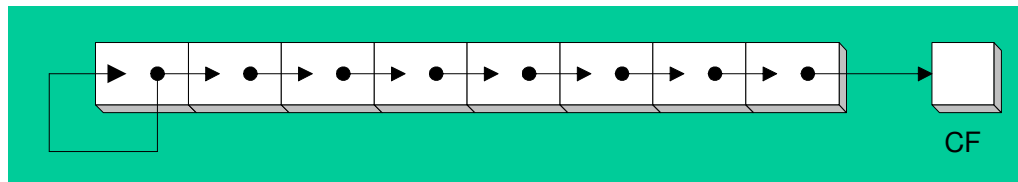


Λογική /αριθμητική ολίσθηση

Η λογική ολίσθηση συμπληρώνει τη νέα θέση bit με 0

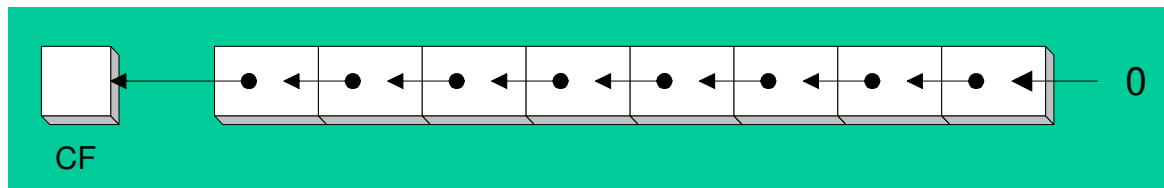


Η αριθμητική ολίσθηση συμπληρώνει τη νέα θέση bit με ήδη υπάρχον περιεχόμενο



Εντολή SHL - SAL

Η **SHL** ή **SAL** μετακινεί κάθε bit του τελεστή μια θέση προς τα αριστερά προς την κατεύθυνση του σημαντικότερου bit. Το σημαντικότερο bit μετακινείται στην CF και στο χαμηλότερο bit εισέρχεται ένα 0. Ισοδυναμεί με ταχύτατο πολλαπλασιασμό του τελεστή επί 2.



Ταχύς πολλαπλασιασμός

Η ολίσθηση μίας θέσης Ισοδυναμεί με ταχύτατο πολλαπλασιασμό του τελεστή επί 2

```
mov dl,5    ; dl = 00000101 = 5  
shl dl,1    ; dl = 00001010 = 10
```

Η ολίσθηση n bit ισοδυναμεί με πολλαπλασιασμό του τελεστή επί 2^n

```
mov dl,5  
shl dl,2    ; DL = 20
```

ΟΛΙΣΘΗΣΕΙΣ

Οι εντολές συντάσσονται με δυο τρόπους. Εφόσον η ολίσθηση είναι μια ο δεύτερος τελεστής είναι 1, αλλιώς ο δεύτερος τελεστής είναι ο καταχωρητής CL ο οποίος και καθορίζει το αριθμό των ολισθήσεων.

Π.χ. με την εντολή:

```
SHL AX, 1
```

Πολλαπλασιάζεται το περιεχόμενο του AX επί 2. Ενώ με τις εντολές :

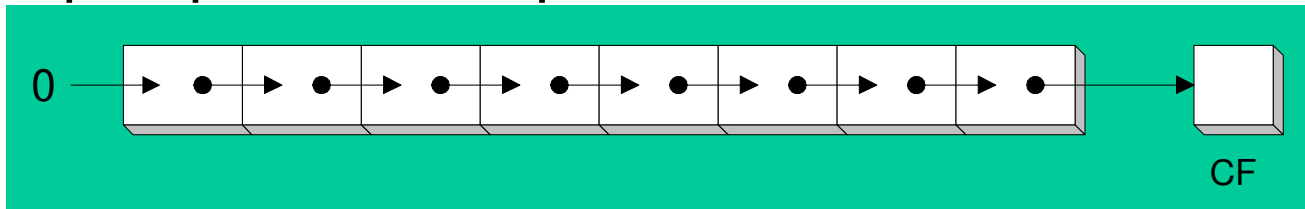
```
MOV CL, 4
```

```
SHL AX, CL
```

Πολλαπλασιάζεται το περιεχόμενο του AX επί 16.

Εντολή SHR

Η **SHR** μετακινεί κάθε bit του τελεστή μια θέση προς τα δεξιά. Το μη σημαντικό bit (υπ' αριθμ. 0) μετακινείται στην CF και στο υψηλότερο bit εισέρχεται ένα 0. Ισοδυναμεί με ταχύτατη μη προσημασμένη διαίρεση του τελεστή δια 2.

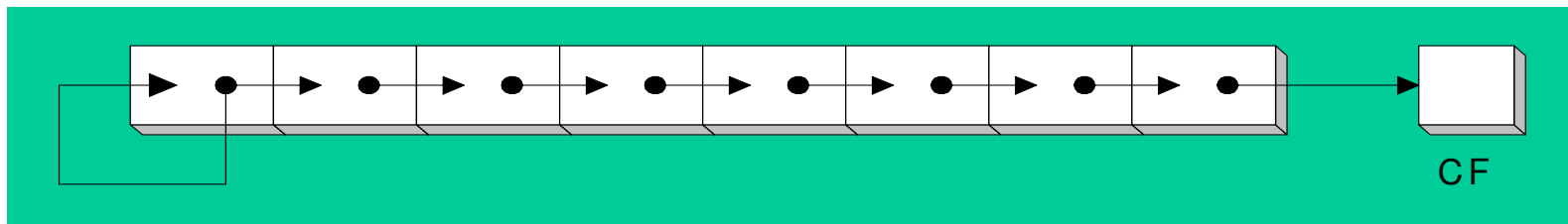


Η ολίσθηση n bit ισοδυναμεί με διαίρεση του τελεστή διά 2^n

```
mov dl,80 01010000
shr dl,1           ; DL = 40
                  00101000
shr dl,2           ; DL = 10
                  00001010
```

Εντολή SAR

Η **SAR** είναι παρόμοια προς την προηγούμενη εντολή δηλαδή μετακινεί κάθε bit του τελεστή μια θέση προς τα δεξιά αλλά το σημαντικότερο bit αφού μετακινηθεί προς τα δεξιά επιστρέφει στην θέση του.



Εξάσκηση

```
mov al,6Bh 01101011
```

```
shr al,1
```

```
shl al,3
```

```
mov al,8Ch 10001100
```

```
sar al,1
```

```
sar al,3
```

a. 35h 00110101

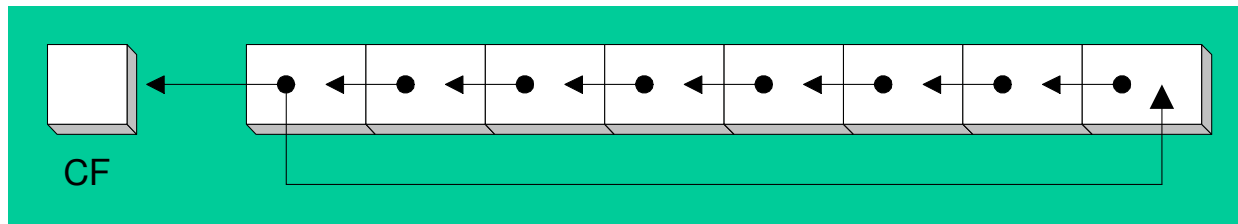
b. A8h 10101000

c. C6h 11000110

d. F8h 11111000

Εντολή ROL

Η **ROL** είναι παρόμοια με την **SHL** με την διαφορά ότι το σημαντικότερο bit μεταφέρεται στο λιγότερο σημαντικό bit καθώς και στην **CF**

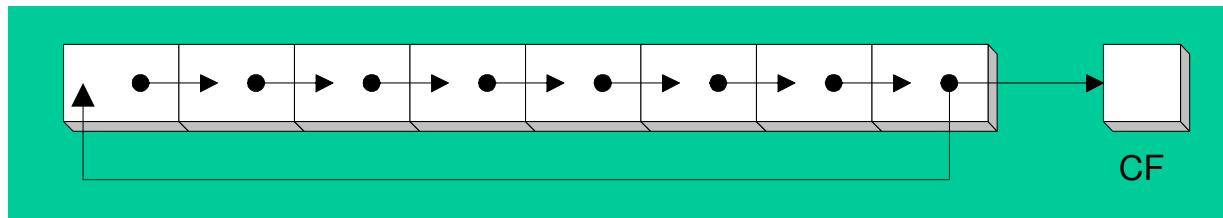


```
mov al,11110000b
rol al,1                ; AL = 11100001b

mov dl,3Fh              ; 00111111
rol dl,4                ; DL = F3h 11110011
```


Εντολή ROR

Η **ROR** είναι παρόμοια με την SHR με την διαφορά ότι το ολιγότερο σημαντικό bit μεταφέρεται στο σημαντικότερο bit καθώς και στην CF



```
mov al,11110000b
ror al,1           ; AL = 01111000b

mov dl,3Fh ;00111111
ror dl,4           ; DL = F3h  11110011
```

εξάσκηση

```
mov al,6Bh 01101011
```

```
ror al,1
```

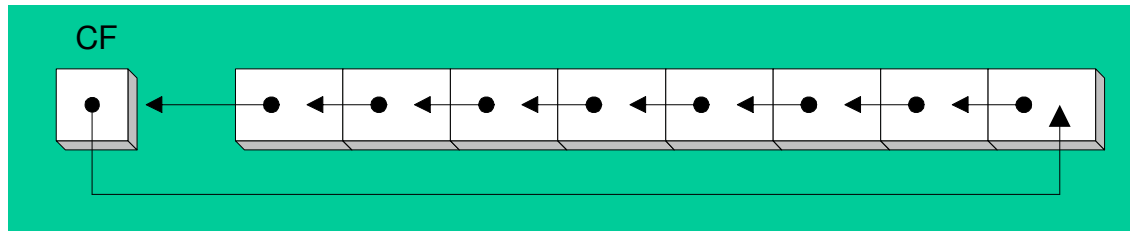
```
rol al,3
```

a. B5h 10110101

b. ADh 10101101

Εντολή RCL

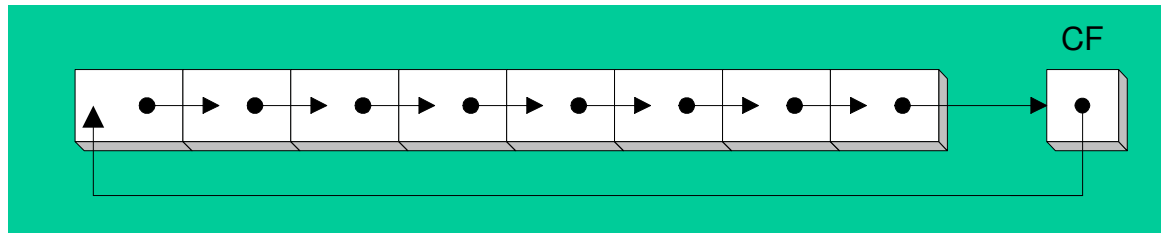
Στην εντολή αυτή κατά την περιστροφή συμμετέχει και η CF. Δηλαδή κατά την RCL τα bits ολισθαίνουν προς τα αριστερά, το σημαντικότερο bit μεταφέρεται στην CF και το περιεχόμενο της CF μεταφέρεται στο υπ' αριθμ. 0 bit.



```
clc                ; CF = 0
mov bl,88h        ; CF=0, BL = 10001000b
rcl bl,1          ; CF=1, BL = 00010000b
rcl bl,1          ; CF=0, BL = 00100001b
```

Εντολή RCR

Στην εντολή αυτή κατά την περιστροφή συμμετέχει και η CF. Δηλαδή κατά την RCR τα bits ολισθαίνουν προς τα δεξιά, το λιγότερο σημαντικό bit μεταφέρεται στην CF και το περιεχόμενο της CF μεταφέρεται στο σημαντικότερο bit.



```
stc                ; CF = 1
mov ah,10h         ; CF = 1 , AH = 00010000b
rcr ah,1           ; CF = 0 , AH = 10001000b
```

Εξάσκηση

```
stc
```

```
mov al,6Bh ; 01101011
```

```
rcr al,1
```

```
rcl al,3
```

a. B5h 10110101 CF =1

b. AEh 10101110 CF =1