

# BİL 362 Mikroişlemciler: Bit Kaydırma Komutları

Ahmet Burak Can  
abc@hacettepe.edu.tr

1

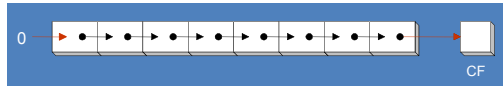
## Kaydırma ve Döndürme Komutları

- Mantıksal ve Aritmetik Kaydırma
- SHL Komutu
- SHR Komutu
- SAL ve SAR Komutları
- ROL Komutu
- ROR Komutu
- RCL ve RCR Komutları
- SHLD/SHRD Komutları

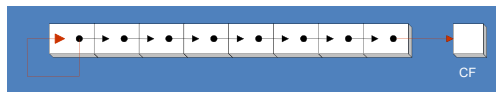
2

## Mantıksal ve Aritmetik Kaydırma

- Mantıksal kaydırma, yeni oluşan bit konumuna 0 kopyalar.



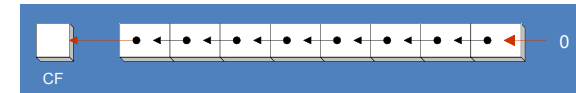
- Aritmetik kaydırma, yeni oluşan bit konumuna, sayının işaret bitini kopyalar.



3

## SHL Komutu

- SHL ("shift left") komutu hedef işlenenin bitlerinde mantıksal sola kaydırma gerçekleştirir ve en az duyarlı bite 0 kopyalar.



4

## Hızlı Çarpma

Bir sayının bitlerini sola kaydırarak sayıyı 2 ile çarpabiliriz.

```
mov dl,5  
shl dl,1
```

Önce: 00000101 = 5  
Sonra: 00001010 = 10

Bitleri n kez sola kaydırırsak sayıyı  $2^n$  ile çarpmış oluruz.

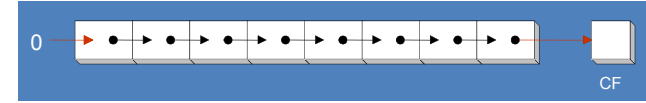
Örnek:  $5 * 2^2 = 20$

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

5

## SHR Instruction

- SHR ("shift right") komutu hedef işlenenin bitlerinde mantıksal sağa kaydırma gerçekleştirir ve en duyarlı bite 0 kopyalar.



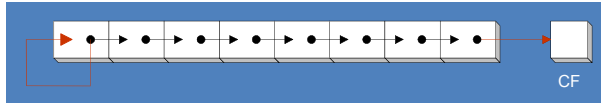
Bitleri n kez sağa kaydırırsak sayıyı  $2^n$ 'e bölmüş oluruz.

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

6

## SAL ve SAR Komutları

- SAL ("shift arithmetic left") komutu, SHL ile aynı işlevi gerçekleştirir.
- SAR ("shift arithmetic right") komutu, hedef işlenin bitleri üzerinde aritmetik sağa kaydırma gerçekleştirir ve işlenenin işaretini korur.



Örnek:

```
mov dl,-80  
sar dl,1      ; DL = -40  
sar dl,2      ; DL = -10
```

7

## Alıştırma

Aşağıdaki her kaydırmadan sonra AL yazmacının onaltılı değerini gösterin.

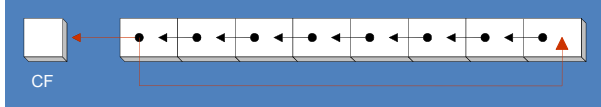
```
mov al,6Bh  
shr al,1      ; a. 35h  
shl al,3      ; b. A8h  
mov al,8Ch  
sar al,1      ; c. C6h  
sar al,3      ; d. F8h
```

8

## ROL Komutu

- ROL (“rotate left”) komutu, bir işlenenin bitlerini sola kaydırır ve en duyarlı biti, hem Elde bayrağına hem de en az duyarlı bite kopyalar.

– Kaybolan bit olmaz.



Örnek:

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

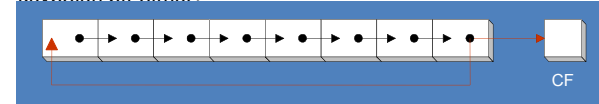
mov dl,3Fh
rol dl,4                ; DL = F3h
```

9

## ROR Instruction

- ROR (“rotate right”) komutu, bir işlenenin bitlerini sağa kaydırır ve en duyarsız biti, hem Elde bayrağına hem de en duyarlı bite kopyalar.

– Kaybolan bit olmaz.



Örnek:

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

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

10

## Alıştırma

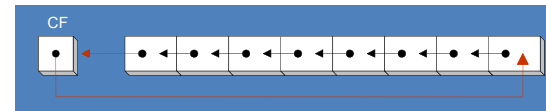
Aşağıdaki her döndürmeden sonra, AL yazmacının onaltılı değerini gösterin.

```
mov al,6Bh
ror al,1                ; a. B5h
rol al,3                ; b. ADh
```

11

## RCL Komutu

- RCL (“rotate carry left”) komutu bir işlenenin bitlerini sola kaydırır:
  - Elde bayrağının değerini en az duyarlı bite kopyalar;
  - En duyarlı biti Elde bayrağına kopyalar.



Örnek:

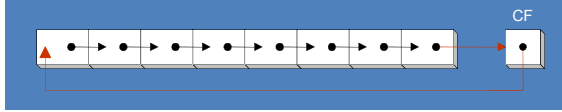
```
clic                    ; CF = 0
mov bl,88h              ; CF=0 ,BL=10001000b
rcl bl,1                ; CF=1 ,BL=00010000b
rcl bl,1                ; CF=0 ,BL=00100001b
```

12

## RCR Komutu

- RCR (“rotate carry right”) komutu bir işlenenin bitlerini sağa kaydırır:

- Elde bayrağının değerini en duyarlı bite kopyalar;
- En az duyarlı biti Elde bayrağına kopyalar.



Örnek:

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

13

## SHLD Komutu

- SHLD (“shift left double”) komutu, bir kaynak işlenenin bitlerini verilen sayı kadar sola kaydırarak hedef işlenene kopyalar.
  - Kaydırma sonucunda hedef işlenende boşalan bite, kaynak işlenenin en duyarlı biti kopyalanır.
  - Kaynak işlenen etkilenmez.

- Sözdizimi:

*SHLD hedef, kaynak, sayı*

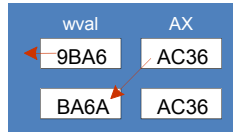
14

## SHLD: Örnek

“wval” veri değerini 4 bit sola kaydırarak en duyarsız bitleri AX yazmacının en duyarlı bitleri ile değiştirmek istersek:

```
.data
wval WORD 9BA6h
.code
mov ax,0AC36h
shld wval,ax,4
```

Önce:



Sonra:



15

## SHRD Komutu

- SHRD (“shift right double”) komutu, bir kaynak işlenenin bitlerini verilen sayı kadar sağa kaydırarak hedef işlenene kopyalar.
  - Kaydırma sonucunda hedef işlenende boşalan bite, kaynak işlenenin en duyarsız biti kopyalanır.
  - Kaynak işlenen etkilenmez.

- Sözdizimi:

*SHRD hedef, kaynak, sayı*

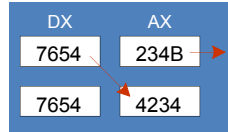
16

## SHRD: Örnek

AX yazmacını 4 bit sağa kaydırarak en duyarlı bitleri DX yazmacının en duyarsız 4 biti ile değiştirmek istersek:

```
mov ax,234Bh
mov dx,7654h
shrd ax,dx,4
```

Önce:



Sonra:

## Alıştırma

Aşağıdaki kaydırma işlemleri sonucunda hedef işlenenlerin onaltılı değerlerini gösterin.

```
mov ax,7C36h
mov dx,9FA6h
shld dx,ax,4      ; DX = FA67h
shrd dx,ax,8      ; DX = 36FAh
```