

BİL 362 Mikroişlemciler: İkili İşlem Komutları

Ahmet Burak Can
abc@hacettepe.edu.tr

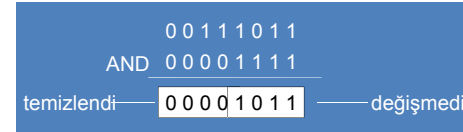
1

AND Komutu

- İki işlenenin karşılıklı bitleri arasında, Boole AND işlemini gerçekleştirir.

- Sözdizimi:

AND hedef, kaynak



AND

x	y	$x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

2

AND Komutu: Uygulamalar - 1

- Görev: AL yazmacı içindeki karakteri büyük harfe çevirmek
- Çözüm: 5. biti temizlemek için AND komutunu kullan.

```
mov al, 'a' ; AL = 01100001b  
and al, 11011111b ; AL = 01000001b
```

3

AND Komutu: Uygulamalar - 2

- Görev: Tamsayı çiftse etikete atlamak
- Çözüm: Sayının en duyarsız (en sağdaki) bitini 1 ile AND'le; sonuç sıfırsa sayı çifttir.

```
mov ax, wordVal  
and ax, 1 ; en sağdaki bit 1 mi?  
cmp ax, 0 ; hayır ise (ZF=0) sayı çift
```

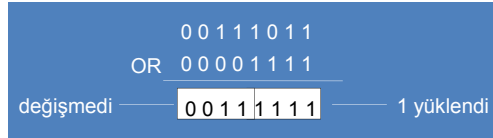
4

OR Komutu

- İki işlenenin karşılıklı bitleri arasında, Boole OR işlemini gerçekleştirir.

- Sözdizimi:

OR hedef, kaynak



OR

x	y	x ∨ y
0	0	0
0	1	1
1	0	1
1	1	1

5

OR Komutu: Uygulamalar - 1

- Görev: İkili sistemde gösterilen onlu bir sayıyı, ASCII karşılığı onlu rakama çevirmek

- Çözüm: 4. ve 5. biti 1 yapmak için OR komutunu kullan.

```
mov al,6 ; AL = 00000110b  
or al,00110000b ; AL = 00110110b
```



'6' rakamının ASCII kodu = 00110110b

6

OR Komutu: Uygulamalar - 2

- Görev: Klavyenin "CapsLock" tuşunu aktif hale getirmek
- Çözüm: BIOS veri alanının 0040:0017h adresinde yer alan klavye bayrağının (1 bayt) 6. bitini OR komutu ile 1 yapın.

```
mov ax,40h ; BIOS bölütü  
mov ds,ax  
mov bx,17h ; klavye bayragi (1 bayt)  
or BYTE PTR [bx],01000000b ; "CapsLock" aktif
```

NOT: Sadece gerçek adres modunda çalışır.

7

OR Komutu: Uygulamalar - 3

- Görev: AL'nin içindeki değer sıfırdan farklıysa etikete atlamak
- Çözüm: AL'nin bitlerini kendisiyle OR'la; sonra JNZ ("jump if not zero") komutunu kullan.

```
or al,al  
jnz IsNotZero ; sifir degilse etikete atla
```

Bir sayıyı kendisiyle OR'lamak, sayının değerini değiştirmez.

8

XOR Komutu

- İki işlenenin karşılıklı bitleri arasında, Boole XOR işlemini gerçekleştirir.

- Sözdizimi:

XOR hedef, kaynak

```
      0 0 1 1 1 0 1 1
XOR  0 0 0 0 1 1 1 1
-----
değişmedi 0 0 1 1 0 1 0 0 ters çevrildi
```

XOR

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Bir işlenenin bitlerini tersine çevirmek için kullanışlı bir yoldur.

9

XOR Komutu: Uygulamalar

Aşağıdaki kod, bir dizideki her karakteri yeni bir değere çevirmek için, XOR komutunu kullanır.

```
KEY EQU 239 ; herhangi bir deger olabilir.
BUFMAX EQU 128
.data
buffer BYTE BUFMAX+1 DUP(0)
bufSize DWORD BUFMAX

.code
mov cx,bufSize ; dongu sayaci
mov si,0 ; buffer icin dizin=0
L1:
xor buffer[si],KEY ; bayti çevir
inc si ; bir sonraki bayta geç
loop L1
```

10

NOT Komutu

- Tek bir işlenenin bitleri üzerinde Boole NOT işlemini gerçekleştirir.

- Sözdizimi:

NOT hedef

```
NOT 0 0 1 1 1 0 1 1
    1 1 0 0 0 1 0 0 Bitler ters çevrildi
```

NOT

X	$\neg X$
F	T
T	F

11

İkili Komutlar ve Bayrakların Durumu

- AND, OR, XOR
 - Taşıma (OF) ve Elde (CF) bayraklarını temizler.
 - İşaret (SF), Sıfır (ZF) ve Eşlik (PF) bayraklarını hedef işlenenin değerine göre etkiler.
- NOT
 - Bayrakları etkilemez.

12

TEST Komutu

- İki işlenenin karşılıklı bitleri arasında AND işlemini gerçekleştirir.
- İşlenenler değişmez, ancak Sıfır ("Zero") bayrağı etkilenir.

- Örnek:

- AL'deki 0 veya 1 numaralı bit 1 ise, etikete atla.

```
test al,00000011b
jnz ValueFound
```

- Örnek:

- AL'deki 0 veya 1 numaralı bit 1 değil ise, etikete atla.

```
test al,00000011b
jz ValueNotFound
```

13

Bayrakların Değerlerini Belirlemek - 1

- Sıfır bayrağı (ZF)

```
test al,0           ; ZF = 1
and al,0           ; ZF = 1
or al,1            ; ZF = 0
```

- İşaret bayrağı (SF)

```
or al,80h           ; SF = 1
and al,7Fh          ; SF = 0
```

14

Bayrakların Değerlerini Belirlemek - 2

- Elde bayrağı (CF)

```
stc                 ; CF = 1
clc                 ; CF = 0
```

- Taşma Bayrağı (OF)

```
mov al,7Fh          ; al = +127
inc al              ; al = 80h (-128), OF = 1
or ax,0             ; OF = 0
```

15

CMP Komutu

- Hedef işleneni kaynak işleneni karşılaştırır.
 - Kaynağı hedeften çıkarma işlemini yapar, ancak hedef değişmez.
- Sözdizimi: **CMP hedef, kaynak**
- Örnekler:

hedef == kaynak

```
mov al,5
cmp al,5           ; ZF = 1
```

hedef < kaynak

```
mov al,4
cmp al,5           ; CF = 1
```

hedef > kaynak

```
mov al,6
cmp al,5           ; ZF = 0, CF = 0
```

16

BT (“Bit Test”) Komutu

- Bir işlenenin n. bitini Elde bayrağına (CF) kopyalar.
- Sözdizimi:
 - BT *işlenen, n*
- Örnek: AX yazmacının 9. biti 1 ise L1 etiketine atla.

```
bt AX,9          ; CF = 9. bit  
jc L1           ; “jump if Carry”
```