

BİL 362 Mikroişlemciler: Aritmetik İşlem Komutları

Ahmet Burak Can
abc@hacettepe.edu.tr

1

Arttırma ve Azaltma Komutları

- Hedef işlenene 1 ekler veya hedef işlenenden 1 çıkarır.

— İşlenen yazmaç veya bellek olmalıdır.

- **INC** *hedef*

• $hedef \leftarrow hedef + 1$

- **DEC** *hedef*

• $hedef \leftarrow hedef - 1$

2

Arttırma ve Azaltma: Örnek

```
.data
myWord WORD 1000h
myDword DWORD 10000000h
.code
    inc myWord           ; 1001h
    dec myWord           ; 1000h
    inc myDword          ; 10000001h

    mov ax,00FFh
    inc ax                ; AX = 0100h
    mov ax,00FFh
    inc al                ; AX = 0000h
```

3

Alıştırma

Aşağıdaki her komut işletildikten sonra, hedef işlenenin değerini gösterin.

```
.data
myByte BYTE 0FFh, 0
.code
    mov al,myByte        ; AL = FFh
    mov ah,[myByte+1]    ; AH = 00h
    dec ah               ; AH = FFh
    inc al               ; AL = 00h
    dec ax               ; AX = FEFF
```

4

Ekleme ve Çıkarma Komutları

- *ADD hedef, kaynak*
 - $hedef \leftarrow hedef + kaynak$
- *SUB hedef, kaynak*
 - $hedef \leftarrow hedef - kaynak$
- Birden çok bellek işleneni kullanılamaz.

5

Ekleme ve Çıkarma: Örnek

```
.data
var1 DWORD 10000h
var2 DWORD 20000h
.code
; ---EAX---
mov eax,var1      ; 00010000h
add eax,var2      ; 00030000h
add ax,0FFFFh     ; 0003FFFFh
add eax,1         ; 00040000h
sub ax,1          ; 0004FFFFh
```

6

NEG (“Negate”) Komutu

Bir işlenenin işaretini değiştirir. İşlenen yazmaç veya bellek işleneni olabilir.

```
.data
valB DB -1
valW DW +32767
.code
mov al,valB      ; AL = -1
neg al           ; AL = +1
neg valW         ; valW = -32767
```

7

NEG Komutu ve Bayraklar

İşlemci NEG komutunu aşağıdaki iç işleme göre işletir:

SUB 0, işlenen

Sıfır olmayan her işlenen Elde (“Carry”) bayrağının 1 olmasına neden olur.

```
.data
valB DB 1,0
valC DB -128
.code
neg valB          ; CF = 1, OF = 0
neg [valB + 1]    ; CF = 0, OF = 0
neg valC          ; CF = 1, OF = 1
```

8

Aritmetik İfadeleri Uygulama

Üst düzey derleyiciler matematiksel ifadeleri Assembly diline çevirir.

Örnek:

Rval = -Xval + (Yval - Zval)

```
Rval SDWORD ?
Xval SDWORD 26
Yval SDWORD 30
Zval SDWORD 40
.code
    mov ax,Xval
    neg ax                ; AX = -26
    mov bx,Yval
    sub bx,Zval           ; BX = -10
    add ax,bx
    mov Rval,ax           ; -36
```

9

Alıştırma

Aşağıdaki matematiksel ifadeyi Assembly diline çevirin. Xval, Yval ve Zval'in güncellenmesine izin vermeyin.

Rval = Xval - (-Yval + Zval)

Not: Tüm değerlerin işaretli çift-sözcük olduğunu varsayın.

```
mov bx,Yval
neg bx
add bx,Zval
mov ax,Xval
sub ax,bx
mov Rval,ax
```

10

Çarpma (MUL) Komutu

- Bir işlenen alır. İşlenen, boyutuna göre, AL veya AX yazmacındaki değerle çarpılır.
 - İşlenen 1 byte ise, işlenen AL yazmacındaki değerle çarpılır, sonuç AX yazmacına yazılır.
 - İşlenen 2 byte ise, işlenen AX yazmacındaki değerle çarpılır, sonucun düşük öncelikli byte'lar AX yazmacında, yüksek öncelikli byte'lar DX yazmacına yazılır.
- Örnek:

```
.data
    count db 3
.code
    mov al, 4
    mov bl, 5
    mul bl
    mul count
    mul bx
```

11

İşaretli Çarpma (IMUL) Komutu

- Çarpma komutunun işaretli sayılarla yapılmasını sağlar.

```
.data
    count db -1
.code
    mov al, 2
    mov bx, -2
    imul count
    imul bx
```

12

Bölme (DIV) Komutu

- Bir işlenen alır.
- İşlenen 1 byte ise, AX yazmacının içeriğini işlenen değerine böler. AL yazmacına bölümü, AH yazmacına kalanı kaydeder.
- İşlenen 2 byte ise, DX ve AX yazmaçlarının ikisinin birlikte oluşturduğu değeri (DX yüksek öncelikli byte'ları tutmak üzere) işlenen değerine böler.

```
mov ax, 43
mov bl, 03
div bl      ;AL = 14  AH = 01
```

13

İşaretli Bölme (IDIV) Komutu

- Bölme komutunun işaretli sayılarla yapılmasını sağlar.

```
mov ax, -13
mov bl, 03
idiv bl      ;AL = -4  AH = -1
```

14

Aritmetik İşlemler ve Etkilenen Bayraklar

15

Aritmetik İşlemlerden Etkilenen Bayraklar

- Aritmetik Mantık Birimi (ALU) aritmetik ve bit işlemlerinin sonuçlarını, durum bayraklarına yansıtır.
 - Hedef işlenenin içeriğine bağlı olarak
- En çok kullanılan bayraklar:
 - Sıfır ("Zero") bayrağı – Hedef sıfıra eşitse 1 olur.
 - İşaret ("Sign") bayrağı – Hedef negatifse 1 olur.
 - Elde ("Carry") bayrağı – İşaretsiz değer aralık ("range") dışıysa 1 olur.
 - Yardımcı Elde ("Auxiliary Carry") bayrağı – En sağdaki baytın üçüncü biti elde veya ödünç oluşturduysa 1 olur.
 - Taşma ("Overflow") flag – İşaretli değer aralık ("range") dışıysa 1 olur.
- MOV komutu bayrakları hiçbir zaman etkilemez.

16

Sıfır Bayrağı (“Zero Flag” - ZF)

Sıfır bayrağı, işlem sonucu hedef işlenende sıfır değerini üretirse 1 olur.

```
mov cx,1
sub cx,1          ; CX = 0, ZF = 1
mov ax,0FFFFh
inc ax            ; AX = 0, ZF = 1
inc ax            ; AX = 1, ZF = 0
```

17

İşaret Bayrağı (“Sign Flag” - SF)

Hedef işlenen değeri;

negatif olduğunda işaret bayrağı 1 olur;

pozitif olduğunda işaret bayrağı 0 olur.

```
mov cx,0
sub cx,1          ; CX = -1, SF = 1
add cx,2          ; CX = 1, SF = 0
```

İşaret bayrağı, hedef işlenenin en duyarlı bitinin kopyasıdır.

```
mov al,0
sub al,1          ; AL = 11111111b, SF = 1
add al,2          ; AL = 00000001b, SF = 0
```

18

Elde Bayrağı (“Carry Flag” - CF)

Elde bayrağı, işlem sonucu işaretsiz, aralık dışı (hedef işlenen için çok büyük veya çok küçük) bir değer üretirse 1 olur.

```
mov al,0FFh
add al,1          ; CF = 1, AL = 00

; Sifirin altına inersek:

mov al,0
sub al,1          ; CF = 1, AL = FF
```

INC ve DEC komutları Elde bayrağını etkilemez.

Sıfır olmayan bir işlenene NEG komutu uygulandığında Elde bayrağı 1 olur.

19

Yardımcı Elde (“Auxiliary Carry Flag” - AC)

Yardımcı Elde bayrağı, hedef işlenenin üçüncü bitinde elde veya ödünç oluştuğunda 1 olur.

```
mov al,0Fh
add al,1          ; AC = 1, AL = 10
```

20

Alıştırma

Aşağıdaki işlemlerin sonucunda; hedef işlenenin değerini ve İşaret, Sıfır ve Elde bayraklarının değerlerini gösterin.

```
mov ax,00FFh
add ax,1          ; AX=0100h SF=0 ZF=0 CF=0
sub ax,1          ; AX=00FFh SF=0 ZF=0 CF=0
add al,1          ; AL=00h SF=0 ZF=1 CF=1
mov bh,6Ch
add bh,95h        ; BH=01h SF=0 ZF=0 CF=1

mov al,2
sub al,3          ; AL=FFh SF=1 ZF=0 CF=1
```

21

Taşma Bayrağı (“Overflow Flag” - OF)

Taşma bayrağı, işlem sonucu işaretli, aralık dışı (hedef işlenen için çok büyük veya çok küçük) bir değer üretirse 1 olur.

```
; Örnek 1
mov al,+127
add al,1          ; OF = 1

; Örnek 2
mov al,-128
sub al,1          ; OF = 1

; Örnek 3
mov al,7Fh
add al,1          ; OF = 1, AL = 80h
```

22

Örnek: EkleCik3.asm

```
.data
Rval WORD ?
Xval WORD 26
Yval WORD 30
Zval WORD 40

.code
main PROC
; Veri bölümü adresini ilklendir
mov ax,@data
mov ds,ax

; INC ve DEC
mov ax,1000h
inc ax ; 1001h
dec ax ; 1000h

; ifade: Rval = -Xval + (Yval - Zval)
mov ax,Xval
neg ax ; -26
mov bx,Yval
sub bx,Zval ; -10
add ax,bx
mov Rval,ax ; -36

; sıfır bayrağı için örnek:
mov cx,1
sub cx,1 ; ZF = 1
mov ax,0FFFFh
inc ax ; ZF = 1

; işaret bayrağı için örnek:
mov cx,0
sub cx,1 ; SF = 1
mov ax,7FFFh
add ax,2 ; SF = 1

; elde bayrağı için örnek:
mov al,0FFh
add al,1 ; CF = 1, AL = 00

; taşma bayrağı için örnek:
mov al,+127
add al,1 ; OF = 1
mov al,-128
sub al,1 ; OF = 1

; Komutların sonu
.exit

main ENDP
END main
```

23