

Başkent Üniversitesi
Elektrik – Elektronik Mühendisliği
Bölümü

EEM 332 – Mikroişlemciler

Ön Çalışma

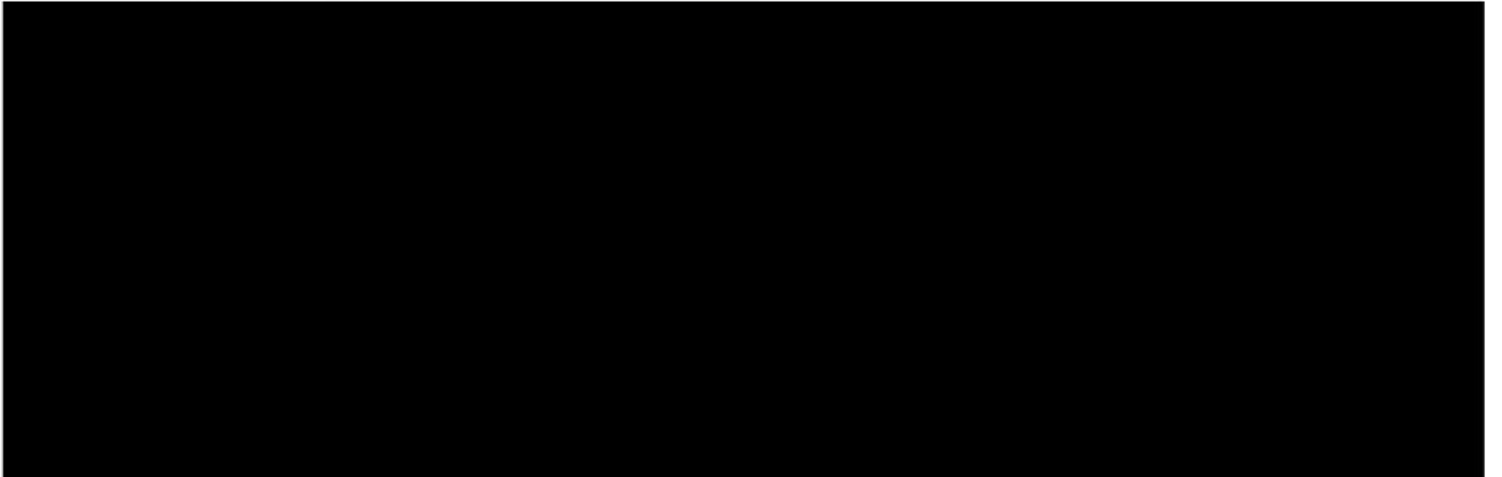
Deney No: 3

Deney Adı: Aritmetik ve Mantık İşlemleri

Deney Tarihi: 14 / 03 / 2007

☐ Salı ☐ Çarşamba

Deney Yeri: ☐ A408 ☐ B407



Bu çalışma,kapak sayfası hariç 3 sayfadan oluşmaktadır.

Ç1.

```
mov cx,ax
mov ax,bx
mov bx,cx
```

Ç2.

```
xchg ax,bx
```

Ç3.

```
mov ax,[300]
xchg ax,[400]
mov [300],ax
```

Ç4.

add ax,1 komutu ax register'ı ile 1 sayısını toplar sonucu ax register'ında saklar.

inc ax komutu ax registerının içindeki değeri 1 artırır.Yani burada 2 komut aynı işlevi görmektedir.

Ç5.

LEA ofset adres yükleme komutudur. Bu adresi 16 bitlik yazmaçlara (AX, BX, CX, DX) yükler.

LEA AX,[BX] burada ofset sadece BX olduğu için BX değerini AX'e yükler. (effective address)

MOV AX,[BX] burada hafıza bölgesinin BX ofsetindeki değer AX'e atanır. (AX'E DS:500'deki veri aktarılır.)

Ç6.

ADC AX,BX $AX=AX+BX+CF$ işlemini gerçekleştirir.

a. $CF=1$ ise $AX=AX+BX+1$

b. $CF=0$ ise $AX=AX+BX+0$

Ç7.

NEG BX komutu BX registerının tümleyenini alır.(2's complement binary olarak).

Bu işlemden BX registerının içindeki değer ve işaret bayrağı etkilenir.

Ç8.

```
mov al,[502]
sub al,[503]
mov bh,[501]
mul bh
mov bh,[500]
add bh,al
mov [504],bh
```

Ç9.

AL=20H
CL=80H

MUL CL → unsigned olarak işlem yaptığı için $20H \times 80H = 1000H$ sonucunu verir.

IMUL CL → signed olarak işlem yapar $80_{16} = 1000\ 0000_2$ 'dir. İlk bit sign biti olduğu için sayı negatiftir. Dolayısıyla sonuç negatif olacaktır. F000H (1'in 2's complementi F tir sonuç negatif olduğu içinde 2's complement formdadır).

Bu işlem sonunda Sign flag 1 olacaktır.

Ç10.

a. SHR AX,1
 SHR BX,1

 SHL AX,1
 SHL BX,1

SHR BX,1 işleminden sonra cf 'in değeri 1 olmuştur sebebi ise:

$0041_{16} = 0000\ 0000\ 0100\ 0001_2$ sağa kaydırıldığında en sağdaki 1 cf'e gitmektedir.

AX ve BX yazmaçlarının yeni değerleri eşittir. Bunun sebebi 2 ye bölme işlemi sonucu AX yazmacının değeri 0020, BX yazmacının değeri de 0020 olmuştur. ($41_{16} / 2 = 20_{16}$)

Karşılaştırma işleminden sonra (CMP AX,BX) zero flag 1 olmuştur.

AX yazmacındaki değeri 1 arttırdıktan sonra (INC AX) karşılaştırma işleminde zero flag ve carry flag 0 olmuştur.

AX ve BX yazmaçlarının değerleri birbiriyle değiştirildiğinde (XCHG AX,BX), zero ve carry flaglerinde bir değişiklik olmamıştır.

b. $16_{10} = 0010_{16}$, $-16_{10} = \text{FFF0}_{16}$
 $8_{10} = 0008_{16}$, $-8_{10} = \text{FFF8}_{16}$

$\text{FFF0 H} = 1111\ 1111\ 1111\ 0000$

$\text{FFF8 H} = 1111\ 1111\ 1111\ 1000$

SAR AX,1 komutu -16 (FFF0H) sayısını ikiye böler, -8 (FFF8).

```
-a cs:100
1627:0100 sar ax,1
1627:0102
-r ax
AX 0000
:fff0
-r
```

AX=FFF0 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1627 ES=1627 SS=1627 CS=1627 IP=0100 NV UP EI PL NZ NA PO NC
1627:0100 D1F8 SAR AX,1

```
-r ip
IP 0100
:
-t
```

AX=FFF8 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1627 ES=1627 SS=1627 CS=1627 IP=0102 NV UP EI NG NZ NA PO NC
1627:0102 3AC4 CMP AL,AH