1. Yer değiştirme işi;

MOV CX,AX

MOVAX,BX

MOV BX,CX

kodlari ile gerçekleştirilir.

-a 100

13B6:0100 mov cx,ax

13B6:0102 mov ax,bx

13B6:0104 mov bx,cx

13B6:0106

-r

AX=1300 BX=1500 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0100 NV UP EI PL NZ NA PO NC

13B6:0100 89C1 MOV CX,AX

-t

AX=1300 BX=1500 CX=1300 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0102 NV UP EI PL NZ NA PO NC

13B6:0102 89D8 MOV AX,BX

-t

AX=1500 BX=1500 CX=1300 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0104 NV UP EI PL NZ NA PO NC

13B6:0104 89CB MOV BX,CX

-t

AX=1500 BX=1300 CX=1300 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0106 NV UP EI PL NZ NA PO NC

13B6:0106 0000 ADD [BX+SI],AL DS:1300=00

-

1. Aynı işlemi

XCHG AX,BX

Kodu ile gerçekleştirebiliriz.

-r

AX=1300 BX=1500 CX=1500 DX=0000 SP=FFEE BP=0000 SI=0002 DI=0002

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0100 NV UP EI PL ZR NA PE NC

13B6:0100 89C1 MOV CX,AX

-a

13B6:010A xchg ax,bx

13B6:010C

-t

AX=1500 BX=1300 CX=1300 DX=0000 SP=FFEE BP=0000 SI=0002 DI=0002

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0106 NV UP EI PL ZR NA PE NC

13B6:0106 87C3 XCHG AX,BX

-t

AX=1300 BX=1500 CX=1300 DX=0000 SP=FFEE BP=0000 SI=0002 DI=0002

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0108 NV UP EI PL ZR NA PE NC

13B6:0108 87C3 XCHG A

1. MOV AX,[0300]

XCHG [0400],AX

MOV [0300],AX

Komutları ile gerçekleştirebiliriz.

1. Komutları programda yazarak deneyecek olursak,

AX=1300 BX=1500 CX=1300 DX=0000 SP=FFEE BP=0000 SI=0002 DI=0002

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=010C NV UP EI PL ZR NA PE NC

13B6:010C 050100 ADD AX,0001

-t

AX=1301 BX=1500 CX=1300 DX=0000 SP=FFEE BP=0000 SI=0002 DI=0002

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=010F NV UP EI PL NZ NA PO NC

13B6:010F 40 INC AX

-t

AX=1302 BX=1500 CX=1300 DX=0000 SP=FFEE BP=0000 SI=0002 DI=0002

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0110 NV UP EI PL NZ NA PO NC

13B6:0110 0000 ADD [BX+SI],AL DS:1502=00

-

\* İki kod da aynı işlemi yapar ancak ilk kod 3 byte yer kaplar iken “inc ax” komutu 1 byte lık yer kaplar.

* Add komutu ekleme işlemi yapar. İnc komutu ise sadece 1 artırır.
* İnc komutu 8,16 bitlik yazmaçlar ve hafızalarda da işlem yapar.

LEA AX,[BX] = Bu komutla BX in efektif adresini AX e aktarır. Lea komutu pointer gibi çalışır.

MOV AX,[BX] = Bu komutla ise adres içeriğini AX e aktarır.

“ADC”

1. “ADC “ komutu aynı “ADD” gibi toplama işlemi yapar ancak “ADC” komutunda carry flagde toplamaya katılır.

Örneğin;

Carry flag=1 ise,

AX=BX+AX+CF

Aşağdaki örnekte görüleceği üzere AX=3B08 , BX=1304 ve CF=1 ( flaglerde CY mevcut)

Sonuç beklentiği gibi AX=BX+AX+CF=3B08+1304+1=4E0D çıktı. Aynı işlemi ADD komutu işle yapsaydık sonuç 4E0C çıkacaktı.

-r

AX=3B08 BX=1304 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0100 NV UP EI PL NZ NA PO CY

13B6:0100 11D8 ADC AX,BX

-a 13b6:0100

13B6:0100 adc ax,bx

13B6:0102

-r

AX=3B08 BX=1304 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0100 NV UP EI PL NZ NA PO CY

13B6:0100 11D8 ADC AX,BX

-t

AX=4E0D BX=1304 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0102 NV UP EI PL NZ NA PO NC

13B6:0102 89D8 MOV AX,BX

7)

Bu komutla sayıların ikinci complimentini bulabiliriz.

Örneğin rasgele bir sayı seçelim. Sayımız AX=1BC3 olsun.

1BC3 sayınısı bit lerine ayırırsak = 0001 1011 1100 0011 elde ederiz.

2. complimenti sağ dan ilk 1 sayısına kadar aynı sonrasını tam tersi ni yazarak elde ederiz yani 1BC3 ün kompimenti = 1110 0100 0011 1101**= E43D** dir.

Deneyerek görecek olursak;

-r

AX=**1BC3** BX=1304 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0100 NV UP EI PL NZ NA PO NC

13B6:0100 11D8 ADC AX,BX

-a 13b6:0100

13B6:0100 **neg ax**

13B6:0102

-t

AX=**E43D** BX=1304 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0102 NV UP EI NG NZ AC PO CY

13B6:0102 89D8 MOV AX,BX

8)

MOV AL, [0503]

SUB [0502], AL

MOV AL, [0502]

MUL [0501]

ADD AL,[0500]

MOV [0504], AL

Komutları ile Z=A+B\*(C-D) işlemini gerçekleştirebiliriz.

9) mul ile imul da aynı değere sahip registerlerin farklı sonuçlar vermesinin sebebi; MUL komutuyla AL ile CL yi çarpıp direk AX e aktarabiliyoruz. Soruda işaretsiz iki hez sayının çarpımı 1000H çıkmıştır. Flaglerden ise overflow ve carry carry flag etkilenir. IMUL komutu ile de verilen iki değer çarpılır, çıkan sonucun 2 komplimenti alınarak AX e aktarılır. Etkilenecek flagler ise bir öncekinde olduğu gibi OV ve CF dir.

10)

-r

AX=0040 BX=0041 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0100 NV UP EI NG NZ AC PE CY

13B6:0100 F7D8 NEG AX

-a 13b6:0100

13B6:0100 shr ax,1 // ax i 2 y bölme

13B6:0102 shr bx,1 // bx i 2 ye bölme

13B6:0104 shl ax,1 // ax i 2 ile çarpma

13B6:0106 shl bx,1 // bx i 2 ile çarpma

13B6:0108

-t

AX=0020 BX=0041 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0102 NV UP EI PL NZ AC PO **NC**

13B6:0102 D1EB SHR BX,1

-t

AX=0020 BX=0020 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0104 NV UP EI PL NZ AC PO **CY**

13B6:0104 D1E0 SHL AX,1

-t

AX=0040 BX=0020 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0106 NV UP EI PL NZ AC PO **NC**

13B6:0106 D1E3 SHL BX,1

-t

AX=0040 BX=0040 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0108 NV UP EI PL NZ AC PO **NC**

13B6:0108 87C3 XCHG AX,BX

-

**İlk bölme işleminin sonunda ax i ikiye böldüğümüzde carry flag 0 dır yani NC olarak görülüyor. Bx i ikiye böldüğümüzde ise sonuç 20 elde 1 vardır, carry flag in NC den CY ye yani 0 dan 1 e geçtiğini görebiliriz.**

**Çarpma işleminde ise ikisinde de 20 den 40 a çıktık. İki işlemde de elde olmadığı için carry flag 0 a yani NC ye geçti.**

**4 işlemde de sonuç sıfır olmadığı için zero flag hiç değişmemiş NZ olarak kalmıştır.**

-a

13B6:0108 inc ax

13B6:0109

-t

**AX=0041 BX=0040** CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0109 NV UP EI PL **NZ** NA PE **NC**

13B6:0109 C3 RET

**Sadece ax i artırdığımız için AX 40 dan 41 e çıktı. BX de ise beklendiği gibi hiçbir değişiklik olmadı.**

**Zero flag NZ Carry flag de NC olarak kaldı. Hiçbir değişiklik olmadı.**

-a

13B6:0109 xchg ax,bx

13B6:010B

-t

**AX=0040 BX=0041** CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=010B NV UP EI PL **NZ** NA PE **NC**

13B6:010B C3 RET

**İşlem sonunda ax ile bx in değerleri bir biri ile yer değiştirdi. Ax 41 ilken 40, bx ise 40 iken 41 oldu.**

**Zero flag ve carry flag de de herhangi bir değişiklik olmadı çünkü sonucu sıfıra giden bir işlem ve işlem sonunda elde kalacak bir hesaplama yapmadık.**

10) -16 sayısı hex olarak FFEA ya karşılık gelmektedir.

-a 13b6:0100

13B6:0100 shr ax,1

13B6:0102

-r

AX=FFEA BX=0041 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0100 NV UP EI PL NZ NA PE NC

13B6:0100 D1E8 SHR AX,1

-t

AX=7FF5 BX=0041 CX=1304 DX=0000 SP=FFEE BP=0000 SI=0004 DI=0004

DS=13B6 ES=13B6 SS=13B6 CS=13B6 IP=0102 OV UP EI PL NZ NA PE NC

13B6:0102 D1EB SHR BX,1