

## Deney 6: Mikroişlemci Dilinde Program Analizi

**Deneyin Amacı** : Mikroişlemci dilinde yazılmış bir programın nasıl çalıştığını, neyi amaçladığını belirlemek için analizinin yapılmasının öğrenilmesidir.

### Deney Öncesi Yapılacak İşlemler:

Mikroişlemci dilinde yazılmış bir programın analizinin simülatör yazılımı kullanılarak analizinin nasıl yapılacağı araştırılarak incelenecektir. Seçilen simülatör yazılımı bilgisayar üzerinde çalıştırılarak ileri düzeyde kullanımı öğrenilecektir.

### Deneyde Yapılacak İşlemler:

Ders kitabı veya ders notlarında **Bölüm 14 Mikroişlemci Programlama Teknikleri** bölümündeki **Mikroişlemcilerin Gelişmiş Komutları** kısmındaki aşağıda verilen programın analizi simülatör programı kullanılarak yapılacaktır.

1. Örnek Pr. 14-7'de, aşağıdaki programı yazarak BCD toplama İşlemini Toplamadan Sonra Akümülatörü Ondalığa Ayarla Komutunu (DAA) kullanarak analizini dört farklı giriş değeri için yapınız.

```
0000          CPU    "6800.TBL"
0000          HOF    "MOT8"
D019          org    0d019H ; programın başlangıç adresi
D019 CE0000   basla: ldx    #0      ; verilerin başlangıç adresi
D01C 0C       basla2: clc          ; Elde bayrağını sıfırla
D01D A640     ldaa   40h,x      ; A akümülatörüne ilk veriyi yükle
D01F A941     adca   41h,x      ; A aküm. ile sonraki veriyi eldeli topla
D021 19       daa          ; sonucu BCD'ye dönüştür
D022 08       inx
D023 08       inx          ; bir sonraki veri adresi
D024 20F6     bra    basla2     ;işleme devam et
0040          org    0040h      ; veri bölgesi
0040 6332482733   dfb   63h,32h,48h,27h,33h,84h,79h,89h ; veriler
FFFE          org    0fffeh     ; reset vektörü
FFFE D019     dwm    basla      ;reset
0000          end          ; programın sonu
```

Ders kitabı veya ders notlarında **Bölüm 14 Mikroişlemci Programlama Teknikleri** bölümündeki **Mikroişlemci Dilinde Program Tasarımı ve Analizi** kısmındaki, aşağıda verilen programların analizi simülatör programı kullanılarak yapılacaktır.

2. Örnek Pr. 14-8'deki sayı dizisinin toplamını hesaplayan programın kaynak dosyasını yazınız ve çeviricide derleyiniz.

```
0040 =          BOYUT:   EQU    0040H
0042 =          TOPLAM:  EQU    0042H
0044 =          DIZI_ADR: EQU    0044H
0000 CE000A   BASLA:    LDX    #10
0003 DF40          STX    BOYUT
0005 CE0000          LDX    #0
0008 DF42          STX    TOPLAM
000A 9643     TOP1:    LDAA   TOPLAM+1
000C AB44          ADDA   DIZI_ADR,X
000E 2403          BCC    ELD_YOK
0010 7C0042          INC    TOPLAM
0013 9743     ELD_YOK: STAA   TOPLAM+1
0015 08          INX
0016 9C40          CPX    BOYUT
0018 26F0          BNE    TOP1
001A DE42          LDX    TOPLAM
0044          ORG    0044H
```

```
0044 0102030405      DFB 1,2,3,4,5,6,7,8,9,10
FFFE                  ORG 0FFFEH
FFFE 00 00           DWM BASLA      ;RES Reset
0000                  END
```

3. Pr. 14-9'da bellekte 70H ile 78H adresleri arasındaki alandaki baytların sıfır ile doldurulması (temizlenmesi) sağlayan programın kaynak dosyasını yazınız ve çeviricide derleyiniz.

```
2000                  ORG 2000H ; programın başlangıcı
2000 CE0070          LDX #70H ; başlangıç adres değerini X yazmacına yükle
2003 6F00 L1:       CLR 0,X ; X+0 adresinin içeriğini sıfırla
2005 08              INX ; adres değerinin artırılması
2006 8C0079          CPX #79H ; son adres sıfırlandı mı?
2009 26F8            BNE L1 ; gelinmediyse L1'e git
0070                  ORG 0070H
0070 0102030405      DFB 1,2,3,4,5,6,7,8,9
```

4. Aşağıda verilen 6802 mikroişlemci dili programın eksiklerini tamamlayınız ve her satırındaki komutun açıklamasını yanına yazınız. E000h adresinden başlayan Programın kaynak dosyasını yazınız ve çeviricide derleyiniz. PC program sayıcısı, A akümülatörü, X dizin yazmacı, durum yazmacının Z (sıfır) biti ve etkilenen bellek gözleri üzerinde analizini yapınız ve programın toplam çalışma süresini hesaplayınız.

Yazmaçların ilk durumu : PC= E000h SP=006Fh X=0002h A=25h B=FAh CCR=CCh

Bellek gözlerinin ilk durumu (bütün değerler hex olarak verilmiştir.) :

0030	FC	FD	FE	FF	01	02	03	04	05	06	07	08	09	10	11	12
0040	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

```
LDX #30h
STX 40h
LDX #38h
STX 42h
L1: LDX 40h
LDAA 0,X
INX
STX 40h
LDX 42h
STAA 0,X
INX
STX 42h
CPX #39h
BNE L1
NOP
```

### Sorular:

1. Bellekteki 70H ile 78H adresleri arasındaki baytların temizlenmesini, sıfır ile doldurulmasını sağlayan programın azalan adres sayacı ile tasarlayarak yazınız.
2. 0011h:0012h ve 0021h:0022h bellek adreslerinde bulunan iki tane 16-bit ikili sayıyı toplayıp sonucu 0030h:0031h:0032h adreslerinde saklayan programı yazınız.
3. 0011h:0012h ve 0021h:0022h bellek adreslerinde bulunan iki tane 4 basamalı BCD sayıyı toplayıp sonucu 0030h:0031h:0032h adreslerinde saklayan programı yazınız.
4. Bellekteki 0000h ile 007Fh adresleri arasındaki 128 tane 8-bit bilginin, sırasıyla 0000h adresindeki bilgi 0100h adresine, 007Fh adresindeki 017fh adresine gelecek şekilde 0100h-007Fh bellek bölgesine taşınmasını X dizin yazmacını artan adres sayacı olarak, yükleme ve saklama komutlarını kullanarak yapan programın tasarlanması
5. Bellekteki 0000h ile 00FFh adresleri arasındaki 256 tane 8-bit bilginin, sırasıyla 0000h adresindeki bilgi 1000h adresine, 00FFh adresindeki 10FFh adresine gelecek şekilde 1000h-10FFh bellek bölgesine taşınmasını SP yığın işaretçi ve X dizin yazmacını adres sayacı olarak, yükleme ve saklama komutlarını kullanarak yapan programı yazınız.