

İ.T.Ü
Elektrik-Elektronik Fakültesi
Bilgisayar Mühendisliği Bölümü

MİKROBİLGİSAYAR
LABORATUVARI
DENEY RAPORU

Deney No: 1

Deney Adı: İTÜ-Eğit'in Tanıtımı ve Program Yazımı

Deney Tarihi: 05.10.2005

Grup:

Deneyi Yapanlar: Semih Önder 040010418

Deneyi Yaptıran Öğretim Elemanı:

İTÜ-Eğit'in Tanıtımı ve Program Yazımı

Deneyin Amacı:

Simülasyon yazılımı kullanarak program hazırlanması ve hazırlanan bu programın İTÜ-Eğitte yazılması ve çalıştırılması.

Deneyde ilk olarak İTÜ-Eğit'in tanıtımı yapıldı. İTÜ-Eğit in işlevleri, fonksiyonları üzerinde duruldu. Bunun ardından MC6000 mikroişlemcisi tanıtıldı. Bu işlemcinin dilinin nasıl okunacağı ve kataloglar tanıtıldı.

Deneyin Yapılışı:

Deneyde hazır olarak verilen program Simulator yazılımı üzerinde MC6800 mikroişlemcisi dili kullanılarak derlenmesi sağlandı. Program çalıştırıldı ve adım adım izlendi.

Program;

Bellekte bulunan 10 adet sayıyı toplayıp sonucu bir bellek gözüne yazacaktır. Sayılar \$4400 adresinden başlayarak yazılacak ve program \$4000 adresinden başlayarak yazılacaktır.

```
BAŞLA    SİL A
          SİL B
          YÜK SK, $4400
DEVAM    TOP A, <SK+00>
          ART SK
          ART B
          KAR B, 10
          DEK DEVAM
          YAZ A, $4440
          DUR
```

Program şekilde verildiği gibi olduğundan bunun makine dilinde pek bir anlamı yoktu. Bu nedenle MC6000 mikroişlemcisi diline çevrildi. Verilen kataloglardan bu kodların işlemci dilinde karşılıkları bulundu. Bu durumda aşağıdaki hale geldi. Program;

```
4F      → CLRA (SİL A)
5F      → CLRB (SİL B)
CE 44 00 → LDX (YÜK SK, $4400) (Burada sıralama kütüğünü yükleme yaptığımız için Load Index Reg komutuyla işlem yapmak zorundayız. MC6000 mikroişlemcisi 2 sekizlik bellek gözlerine sahip olduğundan $4400 sayısını ancak 2 bellek gözüne yerleştirebiliyoruz. bu nedenle 44 ve 00 olarak yükleme yapılmıştır)
AB 00   → ADDA (TOP A, <SK+00>) (Toplama yaptığımız için ADDA işlemini seçiyoruz. Ancak burada sıralama kütüğü üzerinden işlem yaptığımızdan Index sütunundaki kodu seçiyoruz. Çünkü sıralama kütüğü üzerinden işlem yapılacağı zaman Index sütunundaki komutlarla çalışıyoruz. 00 adres bilgisi de sıralama kütüğünün işaret ettiği ilk bellek gözünü belirtiyor)
08      → INX (ART SK) (Sıralama kütüğünü arttırdığımız için Increment Index Reg 1 seçiyoruz)
5C      → INCB (ART B)
C1 0A   → CMPB (KAR B, 10) (Onluk tabandaki 10 sayısını 16lık düzene çevirdiğimiz için 0A yazıyoruz)
```

2D F8 → BLT (DEK DEVAM) (Branch if < Zero seçeneğini seçiyoruz.0 ile karşılaştırdığımızdan 0dan küçük olup olmadığını sorguluyoruz.F8 ile de 8 adım geri gidiyoruz)

B7 44 40 → STAA(YAZ A, \$4440)(Store(yaz) \$4440 bellek gözüne yaz)

3F → SWI(DUR)

Programın Çalışması:

İlk önce A ve B akümülatörlerini silme işlemi yapılıyor daha sonra bellekteki \$4400 adresini sıralama kütüğüne atanıyor. Daha sonra A akümülatörü sıralama kütüğünün işaret ettiği bellek gözündeki sayıyla toplanıyor ve sıralama kütüğünün işaret ettiği adres 1 artırılıyor. B sayısı artırılıp 10 (16lık düzende 0A) sayısı ile karşılaştırma yapılıyor. Bunun sebebi ise maksimum 10 tane sayıyı toplamak istediğimizden kaynaklanıyor. Eğer 10 sayıya ulaşılmamışsa programdaki 'DEVAM' bölümüne dallanma yapıp tekrar aynı toplama işlemleri tekrarlanıyor. 10 sayıya ulaşılmışsa A akümülatörünü \$4440 adres satırına yazılıyor. En son olarak da DUR komutuyla sonlandırma işlemi gerçekleştiriliyor.

Sayı Yükleme:

Bellekteki \$4400 adres gözünden başlayarak 10 tane sayı yazıldı. \$4400 adres gözüne ilk olarak '00' sayısı aktarıldı İTÜ-Eğit'teki M tuşuyla bu işlem gerçekleştirilmiş ve G tuşuyla bir sonraki adres gözüne geçme işlemi gerçekleştirilmiştir. Sırayla yazılan sayılar; (00, 01, 02, 03, 04, 05, 06, 07, 08, 09)

Sayılar yüklendikten sonra program eksiksiz bir biçimde çalıştırılmıştır. Sonuç olarak A akümülatörü \$4440 adres satırına yazıldığından bu adres satırına bakıldı. Sonuç olarak $(0+1+2+3+4+5+6+7+8+9 = 45)$ onluk tabandaki 45 sayısının eşdeğeri 16lık tabandaki 2D) sonucu bulunmuştur.

Programın test aşamasında da bellek gözünde bulunan bir sayı silinip sonuç kontrol edilmiştir. Bellekteki 07 sayısı silinip sonuca bakılmış ve bu seferde karşımıza 16lık tabandaki 26 sayısı çıkmıştır. Buda programımızın sorunsuz çalıştığının göstergesidir.

Programın adım adım çalıştırılması:

Programın adım adım çalıştırılmasında ilk 8 adım incelenmiştir.Bu ilk 8 adımdaki değerler tek tek aşağıda Tablo-1de verilmiştir. Adım adım ilerleme işlemi İTÜ-Eğit'teki T tuşuyla yapılmaktadır. Programı adım adım çalıştırmamız esnasında programın bir noktasına duraksama noktası koymak zorundaydık ve bu noktada DEK DEVAM komutunun olduğu işlem satırı olmak zorundaydı. Yani 2D komutunun bulunduğu adres gözüne duraksama noktası koyduk. Bu işlemi F₈ tuşuyla yapıyoruz. Buradan sonra gelen değerde de dallanma işlemi gerçekleştiriliyor. F8 değeriyle 8 adım geriye gidilmesi gerektiğini anlıyoruz. Bu sayma işlemini de bir sonraki satırdan başlayarak sayma yaparsak sağlıklı bir biçimde gerçekleştirebiliriz. Bu durumda da dallanma işlemi 8 adım gerideki toplama komutunun olduğu adres gözüne oluyor.

Tablo-1

Program adı	PS	ACCA	ACCB	SK	DK

Bağıl adres hesabı:

Daha önce bahsettiğimiz duraksama noktası üzerinden bu işlemleri gerçekleştiriyoruz. Belirlediğimiz bir duraksama noktasından uzaklığını ölçmek istediğimiz adres gözünün uzaklığını ölçmemiz İTÜ-Eğit ile mümkün olabilmektedir. Programımızdaki duraksama noktasını değiştirmeden işlemlerimizi yaptık. İlk önce adresini bildiğimiz TOP komutunun olduğu adres gözüyle duraksama noktası arasındaki uzaklığı ölçtük. Bulduğumuz sonuç F8 değeri idi. Daha sonra Tablo-2 de verilen değerlerle olan uzaklığı ölçtük. Bu değerler Tablo-2 de verilmiştir.

Tablo-2

Dallanılacak adres	Kayıklık değeri
\$4050	43
\$3FF0	E3
\$5000	CIS

\$5000 adres gözüne uzaklık değerinin CIS çıkması bu adres gözünün duraksama noktasına olan uzaklığının büyük olmasından kaynaklanmaktadır. Çünkü İTÜ-Eğit kapsamında iki 8lik ölçümler yapılabildiğinden bulunan uzaklığın büyük olması durumunda ölçüm yapılamamaktadır.