

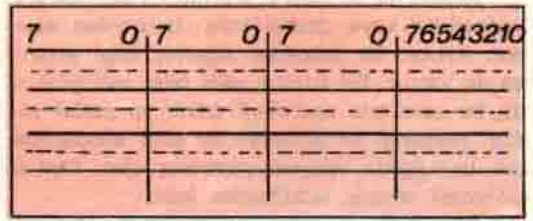
# BİLGİSAYAR BELLEĞİ VE MERKEZİ İŞLEM BİRİMİ

Elekt. Müh. Emrehan HALICI

**B**ilgisayarlarda işlem komutları ve bunlara göre üzerinde işlem yapılacak veriler aynı yerde, sistem belleğinde sakanırlar. Komutların ve verilerin, birbirleriyle karıştırılmadan bilgisayar tarafından kullanılabilmesi önemli bir konudur.

Çevre birimleri dikkate alınmaksızın bilgisayar, iki "kutu" dan oluşmuş bir şekilde düşünülebilir; Ana bellek (tüm bilginin saklandığı yer) ve merkezi işlem birimi (bilginin işlem gördüğü yer). Ekran, klavye, yazıcı ve disket bellek gibi diğer birimler bu iki kutuya bağlıdır.

Ana bellek, içinde "0" veya "1" değeri bulunan "bit"lerden oluşur. Bu bitler ana bellekte rastgele bir biçimde bulunmazlar. Belli sayıda bit yan yana düşünülür ve bunların oluşturduğu bit kümesine "sözcük" denir. Her bilgisayarın sözcük uzunluğu sabit bir sayıdır. Sözcük uzunlukları değişik bilgisayarlar için değişik değerler alabilir. Mikrobilgisayarlar, mi-



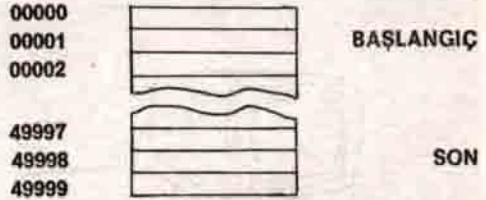
8 Bitlik bir bilgisayarın BELLEK yapısı

nibilgisayarlar ve büyük boy bilgisayarlar için genellikle kullanılan sözcük uzunlukları tabloda gösterilmiştir.

8 bitlik bir bilgisayarın bellek yapısı şekildedeki gibidir.

Her kutucukta 0'dan 7'ye kadar olmak üzere 8 bit bulunur. Bu 8 bit, bilgisayarın bir sözcüğünü oluşturur. 8 bitlik bir veriye "bayt" adı verilir. Böylece, sözcük uzunluğu 8 bit olan bilgisayarlar da bayt ve sözcük aynı anlama gelir.

Bellekte, tek tek bitlere ulaşmak yerine, sözcüklere ulaşmak tercih edildiğinden, her sözcüğün bellekte belirli bir yeri vardır. Bu yere sözcüğün adresi denir. Sözcük adresleri 0'dan başlar ve birer birer artarak belleğin üst sınırına kadar gelir. Örneğin, 50.000 sözcüklü bir belleğin adreslenmesi şekildeki gibi yapılabilir:



Sözcük uzunluğu	Mikrobilgisayarlar	Minbilgisayarlar	Büyük bilgisayarlar
4	Çok	Yok	Yok
6	Yok	Bazı	Yok
8	En yaygın	Bazı	Yok
12	Bazı	Bazı	Yok
16	Bazı (ve yeniler)	En yaygın	Bazı
18	Yok	Bazı	Bazı
24	Yok	Bazı	Bazı
32	Yok	Bazı	En yaygın
64	Yok	Yok	Daha büyükler için en yaygın

Bilgiler bu adreslerin aracılığıyla kutucuklara yerleştirilir veya istendiğinde buralardan okunur. Ayrıca, bir adresten diğerine bilgi aktarımı da yapılır. Bir adrese yeni bilgi yollandığında, bu adresteki eski bilgi silinir ve yenisi yazılır. Herhangi bir adresten bir bilgi alındığında ise, bu bilginin kopyesi çıkarılmış olur. Yani o adresteki sözcük bozulmadan kalır.

Bilgi üzerinde işlem yapılan yerin, merkezi işlem birimi olduğunu söylemiştik. Merkezi işlem biriminde kayıtlayıcılar "register" bulunur. Ana bellekte olduğu gibi, kayıtlayıcılarda da bilgiler saklanabilir ve her türlü aktarma işlemi yapılabilir. Buralarda fazladan aritmetik işlemler de yapılır.

Şimdi hayali bir mikrobilgisayar sistemi varsayalım ve bu bilgisayarın basit bir problemi nasıl çözdüğünü görmeye çalışalım. Ona söylemek istediklerimizi şöyle sıralayalım :

1. Sana iki sayı vereceğiz,
2. Bu iki sayıyı topla,
3. Sonucu yazıcıya yaz,

(Bu örnekte kolaylık olsun diye desimal sayılar kullanılacaktır. Bilindiği gibi aslında bilgisayar içinde tüm sayılar ikili sayı sistemindedir.)

İşleme başlamadan önce mikrobilgisayar sistemi, bellekten bir bölüm ve kayıtlayıcının şekilindeki gibi olduğunu varsayalım.



Hayali mikrobilgisayarımızda komutlar iki bölümden oluşmaktadır : İşlem kodu (iki rakam) ve bu işlemden etkilenecek bellekteki herhangi bir sözcüğün adresi (dört rakam)

79	63	00	95
----	----	----	----

**Komutun bellekteki adresi                      İşlem kodu                      Sözcük adresi**

Bilgisayar içinde yapılabilecek bütün işlemlerin bir kod numarası bulunur. Mikrobilgisayarımızda 63 kodu şu anlama gelmektedir : "klavyeden girilen sayıyı ana belleğe yerleştir". Daha sonraki 0095 sayısı ise, yerleştirme işleminin hangi adresteki sözcüğe yapılacağını göstermektedir. İstedığımız toplama işlemini bilgisayara yaptıracak komutlar, bellekte 79'uncu adresten 84'üncü adrese kadar olan yerdedir. Bu komutların belleğe nasıl yerleştirildiği konusunu burada dikkate almadan, tek tek komutları inceleyelim.

- **79 63 00 95** : Klavyeden girilen sayı, bellekte 95 nolu adresteki sözcüğe yerleştirilir. Eğer o an klavyeden 23 sayısı girilmişse, bu komutun sonunda 95 nolu adresteki sözcük, aşağıda görüldüğü gibi olur :

95	00	00	23
----	----	----	----

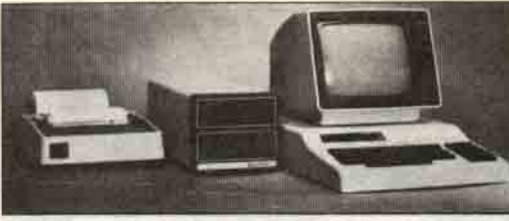
- **80 63 00 96** : Yukarıdaki komutun aynıysa olan bu komutun sonunda klavyeden okunan sayı 96 nolu adrese yerleştirilir. Şimdi de klavyeden 100 sayısının girildiğini varsayalım, bu durumda 96 nolu adresteki sözcük, alta görüldüğü gibi olacaktır :

96	00	01	00
----	----	----	----

- **81 19 00 95** : 19 nolu işlem kodu şu anlama gelmektedir : "Bellekten bir sayı al ve kayıtlayıcıya kopyala Adres olarak verilen 95 sayısı ise, belleğin hangi adresindeki sayının alınacağını göstermektedir. 95 nolu adreste 23 sayısı bulunduğundan, kayıtlayıcıya da **00 00 23** sayısı yerleştirilecektir.

- **82 29 00 96** : 29 nolu işlem kodu, "bellekten bir sayı al ve kayıtlayıcıdaki sayının üstüne ekle" demektir. Önceki komutlarda olduğu gibi, burada da 96 sayısı hangi adresteki sayının alınacağını göstermektedir. 96 nolu adreste 100 olduğundan, bu komutun sonunda kayıtlayıcı şöyle olacaktır :





**Tipik bir mikrobilgisayar sistemi :**  
Ekran, klavye, yazıcı ve disket bellek

00 01 23

● **83 40 00 97** : Bu kod, "kayıtlayıcıdan sayıyı al ve belleğe kopyala" anlamındadır. 97 sayısı, kopyalamanın yapılacağı adrestir. Komutun sonu da 97 nolu adreste, daha önceden bulunan sayı silinecek ve 123 sayısı yerleştirilecektir :

97 00 01 23

● **84 64 00 97** : 64 nolu işlem kodu, "bellekten bir sayı al ve yazıcıya yolla" anlamına gelmektedir. Burada da 97 sayısı, bellekteki hangi adreste bulunan sayının alınacağını göstermektedir. Bu komutun sonunda, 123 sayısı yazıcıya yazılacak ve istediğimiz toplama işlemi gerçekleşmiş olacaktır.

Üstteki toplama işlemi yaptıran komutlar ana bellekte 79 nolu adresten başlamakta ve diğer komutlar sırayla bunu takip etmektedir. Peki bilgisayar, ilk olarak hangi adresteki komutu dikkate alacak ve daha sonra hangi adresteki komutlara geçecektir? Bu sıralama işlemi, merkezi işlem biriminde bulunan "sıralama kayıtlayıcısı" tarafından gerçekleştirilir. Sıralama kayıtlayıcısında her zaman için, bilgisayarın uy-

gulayacağı komutun adresi bulunur. Bir komut yapıldıktan sonra, sıralama kayıtlayıcısı yeni komutun adresini gösterir. O halde, toplama işleminin başlaması için sıralama kayıtlayıcısında ilk olarak 79 sayısı bulunmalıdır. Böylece, bilgisayar işleme başlamak için 79 nolu adresi referans alacaktır. Dikkate değer bir başka konu da, 84 nolu adresteki komuttan sonra bilgisayarın ne yapacağıdır. Bu komuttan sonra sıralama kayıtlayıcısında 85 sayısı bulunacaktır. O halde işlemin bitmesi isteniyorsa, 85 nolu adrese işlemin bittiği anlamına gelen "dur" komutu konulmalı veya program yeni toplamalar yapmak üzere başa döndürülmelidir. Programı başa döndürmek şöyle gerçekleştirilebilir :

**85 51 00 79**

: Buradaki 51 nolu işlem kodu şu anlama gelmektedir: "Bu komuttaki sayıyı al ve sıralama kayıtlayıcısına koy". Komutun sonunda 79 sayısı, sıralama kayıtlayıcısına yüklenecek ve bir sonraki komut 86 değilde 79 nolu adresten alınacaktır. Yani program yeniden işleyecek, klavyeden gelen iki sayı toplanarak yazıcıya yazdırılacaktır. Bu toplama işlemi durmadan devam edecektir.

Tüm karışık ve uzun problemler, bilgisayar içinde temel bazı işlemlerle gerçekleşir. Bu işlemler sırasında kayıtlayıcılar, bellek ve diğer çevre birimleri kullanılır. Kullanıcıya düşen görev, probemin çözülmesi için gereken işlemleri bilgisayara yaptıracak olan komutları, sıralı bir biçimde bilgisayara aktarmaktır.

**Bir şeyin nedenini öğrenmeyi  
kral olmaya yeğ tutarım.**

**DEMOKRİTUS**

Tartışırken, ileri sürülen fikre güç kazandırmak için otoriter davranışlara ağırlık verilmemelidir. Çünkü otoriter davranışlar, öğrenmek isteyenlerin öğrenmelerini engelleyen bir ortam yaratır, kendi yargılarını kullanmaktan alıko/ ar. Böylece bu kişiler, karşılarındakinin her sözünü sorunu çözümlleyici bir yargı olarak kabul ederler.

Yargısı önceden verilmiş bir düşünce yaratmak ise, aklın desteğinden yoksun bir otorite kurmak demektir.

**ÇİCERO**