

MİKROİŞLEMCİ SİSTEMLERİ

Yrd. Doç. Dr. Şule Gündüz Öğüdücü
<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Merkezi İşlem Birimi

- Merkezi İşlem Birimi (MİB): Bilgisayarın temel birimi
 - Hız
 - Sözcük uzunluğu
 - Buyruk kümesi
 - Adresleme yeteneği
 - Adresleme kapasitesi

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

MİB Yapısı

- Bellek Adres Kütüğü (BAK)
- Bellek Veri Kütüğü (BVK)
- Aritmetik Lojik Birim (ALB)
- Akümülatör (ACC)
- Durum Kütüğü (DK)
- Yardımcı Kütükler (YK)
- Program Sayacı (PS)
- Buyruk Kütüğü (BK)
- Buyruk Çözücü (BÇ)
- Yığın Göstergesi (YG)
- Sıralama Kütüğü (SK)
- Denetçi (DEN)

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

MİB Altbirimleri

- Bellek Adres Kütüğü (BAK): MİB'den giden veya MİB'e gelen verilerin gideceği veya geldiği yerin adresini belirtir. Adres yolunu bağlıdır.
- Bellek Veri Kütüğü (BVK): MİB'den giden veya MİB'e gelen verilerin alınıp verildiği bir iskele. Veri yoluna bağlıdır.
- Akümülatör (ACC): BVK üzerinden gelen veriler akümülatöre alınır. Aritmetik ve lojik işlemler sırasında verinin üzerinde bulunduğu yer.
- Aritmetik Lojik Birim (ALB): ACC'deki veri üzerinde yapılacak işlemler ALB tarafından yürütülür.
 - Aritmetik işlemler
 - Mantıksal işlemler
 - Karşılaştırma ve karar verme işlemleri

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Durum Kütüğü

- DK'nın içeriği karar verme işlemlerinde kullanılır.
- ALB tarafından gerçekleştirilen işlemlerin sonunda ortaya çıkan durumlar
 - Elde
 - Borç
 - Yarım Elde ve Yarım Borç
 - Taşma
- ACC'nin yüklenmesi sonucunda ortaya çıkan durumlar
 - Sıfır
 - Negatif

				T	S	N	Y	E
--	--	--	--	---	---	---	---	---

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Durum Kütüğü

- Örnek:
A= 01001000 B= 01111001
A+B:
A 01001000
B +01111001
 11000001 T=1 S=0 N=1 Y=1 E=0

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Yardımcı Kütükler

- MİB içindeki işlemlere hız kazandırır.
- Üzerinde sık işlem yapılacak veriler için kullanılırlar.
- Veriler BVK üzerinden belleğe ya da G/Ç arabirimine gönderilebilirler veya bu birimlerden veri alabilirler.
- Aritmetik lojik işlemlerde birinci işlenen olamazlar.

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

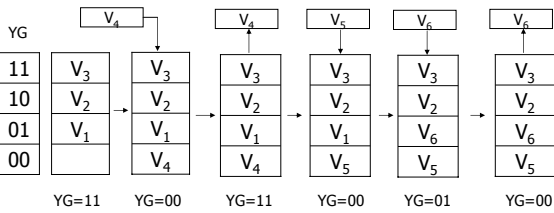
MİB Altbirimleri

- Program Sayacı: Bir sonraki buyruğun adresini içerir. Boyu BAK'nın boyuna eşittir.
- Buyruk Kütüğü: O anda işlenen buyruğu içerir.
- Buyruk Çözücü: BK'da bulunan buyruğun taşıdığı anlamın çözüldüğü yer.

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Yığın

Verileri üst üste yığmaya ve gerektiğinde verileri yığından teker teker almaya yığın işlemi denir.
Örnek: MİB içinde yığın için 4 göz bulunuyor.



<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Yığın Göstergesi

Bellek içinde istenen bir bölge yığın için ayrılıyor. Yığının nerde kurulduğunu belirtmek için yığın göstergesi kullanılıyor. Yığın göstergesi yığının başlangıç adresi ile yüklenir. Yığına her yeni veri eklendiğinde yığın göstergesinin değeri bir azalır.

Adres	YG
\$A000	\$ 9FFF
\$9FFF	\$9FFE
\$9FFE	\$9FFD
\$9FFD	\$9FFC
\$9FFC	\$9FFB
\$9FFB	\$9FFA

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Yığın Göstergesi

Bellek içinde istenen bir bölge yığın için ayrılıyor. Yığının nerde kurulduğunu belirtmek için yığın göstergesi kullanılıyor. Yığın göstergesi yığının başlangıç adresi ile yüklenir. Yığına her yeni veri eklendiğinde yığın göstergesinin değeri bir azalır.

Adres	YG
\$A000	\$ 9FFF
\$9FFF	\$9FFE
\$9FFE	\$9FFD
\$9FFD	\$9FFC
\$9FFC	\$9FFB
\$9FFB	
\$9FFA	

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Sıralama Kütüğü

Sıralı verilerin belleğe yazılması veya bellekte sıralı olarak bulunan verilerin okunması için kullanılır. Boyu BAK boyuna eşittir.

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Denetçi

- Hem MİB içindeki hem de bilgisayar içindeki çalışmayı düzenler.

Girişler

SAAT

Albaştan

Bekle

Kesme

Kesme İsteği

Yalıtım

Çıkışlar

Oku/Yaz

Yalıtıldı

Saat çıkışı

Bellek ya da G/Ç

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Denetim Birimi Girişleri

- Saat: Bilgisayar içindeki her işlem saat işaretine bağlı olarak gerçekleşir.
- Albaştan: Bilgisayarı ilk açıldığı duruma getirmek için kullanılır.
- Bekle: MİB bir süre durmasını sağlar.
- Kesme: MİB o anda yürütmekte olduğu programı keser ve daha önce belirtilmiş olan kesme hizmet programını yürütür.
- Kesme İsteği: Bu girişe gelen kesme isteğine, kesme isteği karar bayrağına verilecek değer ile izin verilir ya da verilmez.
- Yalıtım: MİB adres ve veri yolu çıkışları üçüncü konuma getirilir.

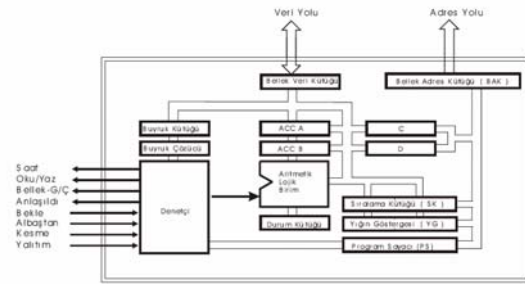
<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Denetim Birimi Çıkışları

- Oku/Yaz: Verilerin akış yönünü belirler.
- Yalıtıldı: Veri ve adres yolunun yalıtıldığını belirtir.
- Saat Çıkışı: Bilgisayar içindeki diğer birimlere gerekli olan saat işaretini sağlar.
- Bellek ya da G/Ç: Adres yolu üzerinde bulunan adres bilgisinin bellek adresi mi yoksa G/Ç arabirimi adresi mi olduğunu belirtir.

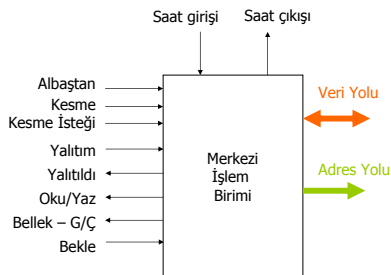
<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

MİB'in İç Yapısı



<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

MİB Dış Görünümü



<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

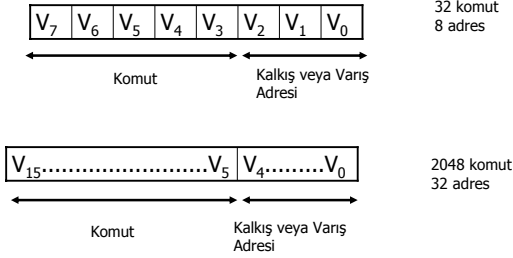
Buyruklar

- MİB'nin her adımda yapması gereken işlemleri belirten açıklama
 - Tek Alanlı Buyruk Yazma Kalıbı
 - Çok Alanlı Buyruk Yazma Kalıbı

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Tek Alanlı Buyruk Yazma Kalıbı

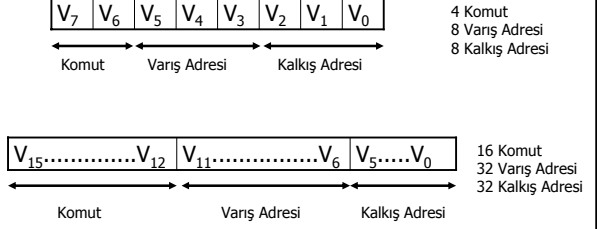
■ Bir Adresli Buyruk Yazım Kalıbı



<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Tek Alanlı Buyruk Yazma Kalıbı

■ İki Adresli Buyruk Yazımı

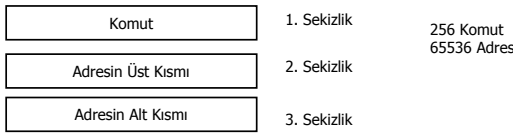


<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Çok Alanlı Buyruk Yazma Kalıbı

■ Çok Sözcük İçine Yazılmış Buyruklar

■ 1 Adresli Buyruk Yazımı



<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Buyruk Kümesi

■ MİB'nin buyruk kümesinin özellikleri

- Buyruk kümesi (hangi işlemler yapılabilir?)
- Buyruk yapısı (Buyruklar nasıl belirtiliyor?)
- Veri saklama (Veri nerde saklanıyor?)
- Adresleme yöntemleri (Veriye nasıl erişiliyor?)
- Hata durumu (Hata oluştuğunda ne oluyor?)
- Bilgisayar mimarisini anlamak işletim sistemi tasarımcıları, derleyici tasarımcıları ve programcılar için önemli.

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

MİB Buyruk Kümesi

- Bilgisayar donanım olarak gerçekleştirildiğinde sadece buyruk kümesini biliyor.
 - **CISC** (Complex Instruction Set Computer - karmaşık komut kümesi bilgisayarı): Çok sayıda komut destekler
 - **RISC** (Reduced Instruction Set Computer - azaltılmış komut kümesi bilgisayarı): Bu tasarımda gerekli komut sayısı azaltılarak donanım yükü yazılıma kaydırılmıştır.
- MİB sadece ikili verilerden oluşan buyruklarla işlem yapabiliyor.
 - Bütün buyruklar 0 ve 1 dizilerinden oluşuyor.
 - Bütün veriler yine 0 ve 1 dizilerinden oluşuyor.

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Makine Dili

- MİB'nin yapabildiği işlemlere karşılık düşen ikili sayılardan oluşan buyruk kümesi
 - 0 ve 1 bitlerinin belirli kod düzenine göre sıralanmasından oluşur.
- Her MİB'nin kendine özgü buyruk kümesi vardır.

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Asembler Dili

- Makine dilinde ikili kodlanmış buyrukların herbiri bu buyrukları çağrıştırarak kısaltmalar kullanılarak isimlendirilmiştir.
- Kısaltmalar kullanılarak yazılan programlara asembler dilinde ya da simgesel dilde yazılmış olur.

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Örnek

Örnek MİB için simgesel dil ve makine dili

Etiket Komut İşlenen Açıklama

Adres	İçerik
0010	00 20 00 80 BAŞLA YÜK A, <\$0080>
0014	03 20 00 81 TOP A, <\$0081>
0018	01 20 00 82 YAZ A, <\$0082>

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Buyruğun Yürütülmesi

```
I:=0
REPEAT
  Bellekte I adresinden bir buyruk al
  Buyruğu yürüt
  I:=I+1
FOREVER
```

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Getirme Çevrimi Aşamaları

- Bellekte \$2000 adresinde bulunan bir buyruk yürütülmek isteniyor. PS değeri \$2000 olarak ayarlanıyor. Buyruğun getirilmesi için gerekli adımlar:
 - Adres yoluna PS bulunan adres bilgisini konulur ve denetçi OKU sinyali etkin duruma getirir
 - Adres kod çözücü girişine gelen adres bilgisi çözülerek hangi bellek gözüne erişileceği belirlenir.
 - Bellek gözündeki veri veri yoluna aktarılır
 - Veri yolundaki veri BVK üzerinden MİB içindeki BÇ aktarılır
 - Buyruk çözüldükten sonra buyruğun yürütülmesi için denetçi uygun sinyalleri oluşturur

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

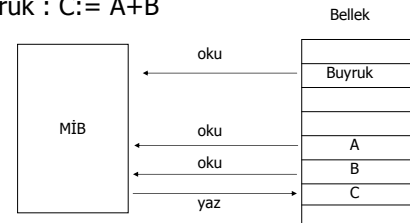
Buyruğun Yürütülmesi

```
I:=0
REPEAT
  Bellekte I adresinden bir buyruk al
  Getirme çevrimi aşamaları
  Buyruğu yürüt:
    Eğer veri bellekteyse veriyi al
    Buyrukta belirtilen işlemi yap
    Eğer sonucu belleğe yazmak gerekiyorsa yaz
  I:=I+1
FOREVER
```

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

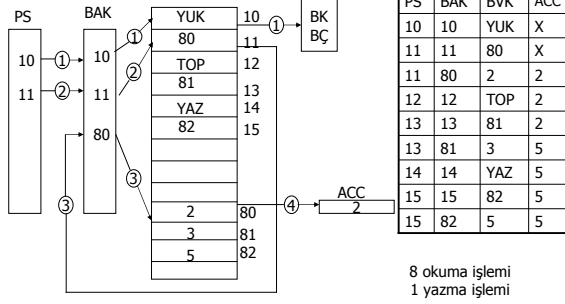
Örnek

Buyruk : C:= A+B



<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>

Örnek



8 okuma işlemi
1 yazma işlemi

<http://www.ninova.itu.edu.tr/EgitimDetay.aspx?eId=30>