

# Von Neumann Mimarisi

# Sayısal Bilgisayarın Tarihsel Gelişim Süreci

- Babage'in analitik makinası (1833)
- Vakumlu lambanın bulunuşu (1910)
- İlk elektronik sayısal bilgisayar (1946)
- Transistörün bulunuşu (1947)
- İlk transistörlü sayısal bilgisayar (1960)
- Entegre devrenin bulunuşu (1963)
- İlk mikroişlemci (1970)
- 10.000 transistörlü entegre devreler (1981)

# Saklı Program Kavramı

- Bilgisayardan istenilen işlerin gerçekleştirilebilmesi için gereken işlem dizisi bilgisayarın içinde saklanabilmektedir.
- İstenilen bir işi yerine getiren işlemlerin listesine program denir. Programlar bilgisayarın içinde saklanırlar.
- Program kavramının iki avantajı vardır; Birincisi işlemler için gereken zaman kullanıcıya bağlı değildir. İkincisi sadece veriler girilerek aynı işlemler tekrar tekrar yaptırılabilir.



# Temel Yapı

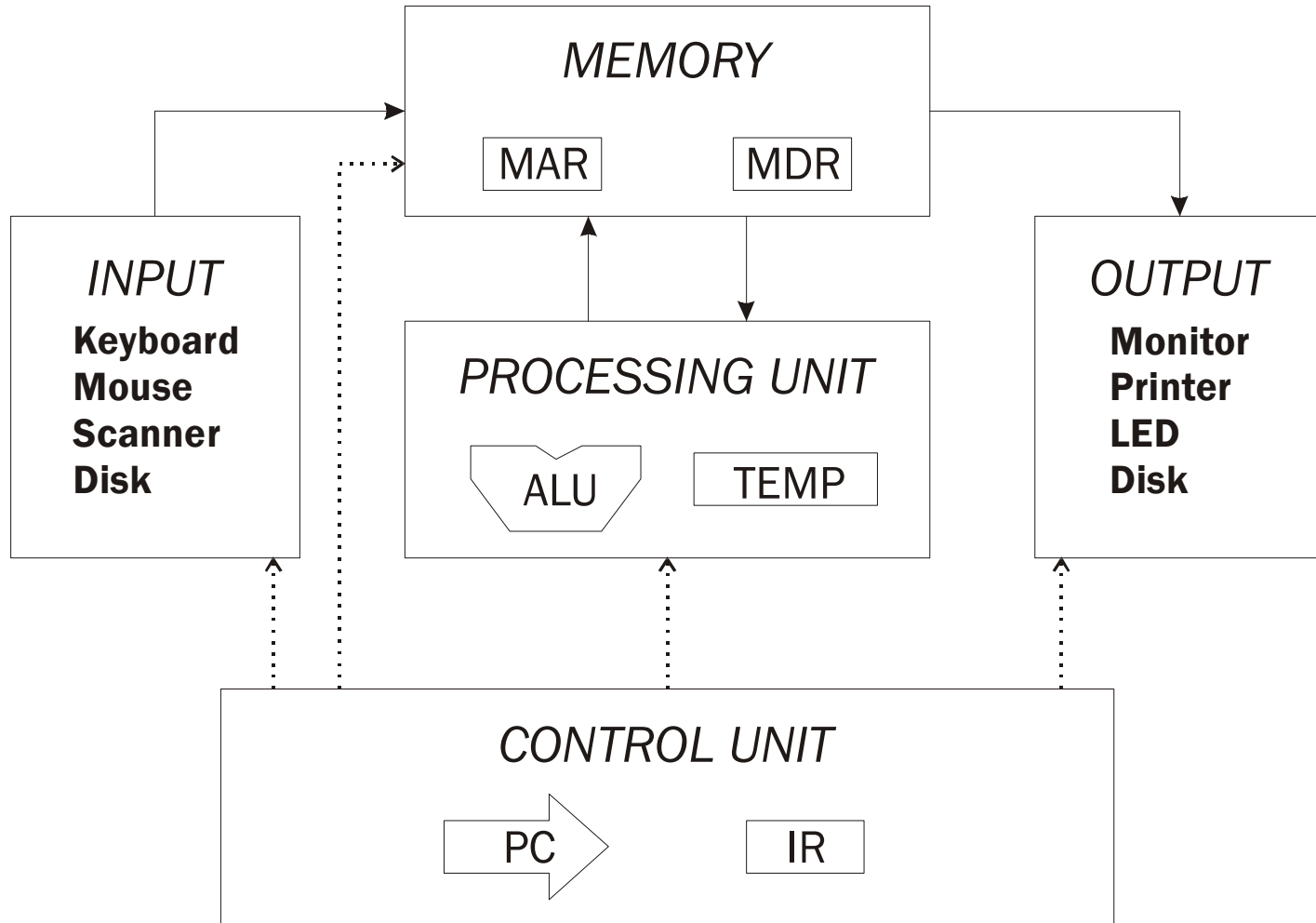
Sayısal bir bilgisayar 3 bileşenden oluşmaktadır.

- i) Hem programı oluşturan komutların listesini, hem de işlemeyi bekleyen veri değerlerini tutan bir bellek birimi
- ii) Her bir program komutunu yorumlayıp yürüten merkezi işlem birimi (MİB & CPU)
- iii) Bir veya daha çok giriş ve çıkış portu sağlayan giriş-çıkış (I/O) arabirim ünitesi






Mikroişlemci, ROM, RAM ve I/O birimlerinden oluşan adreslenebilir saklayıcılar topluluğudur.

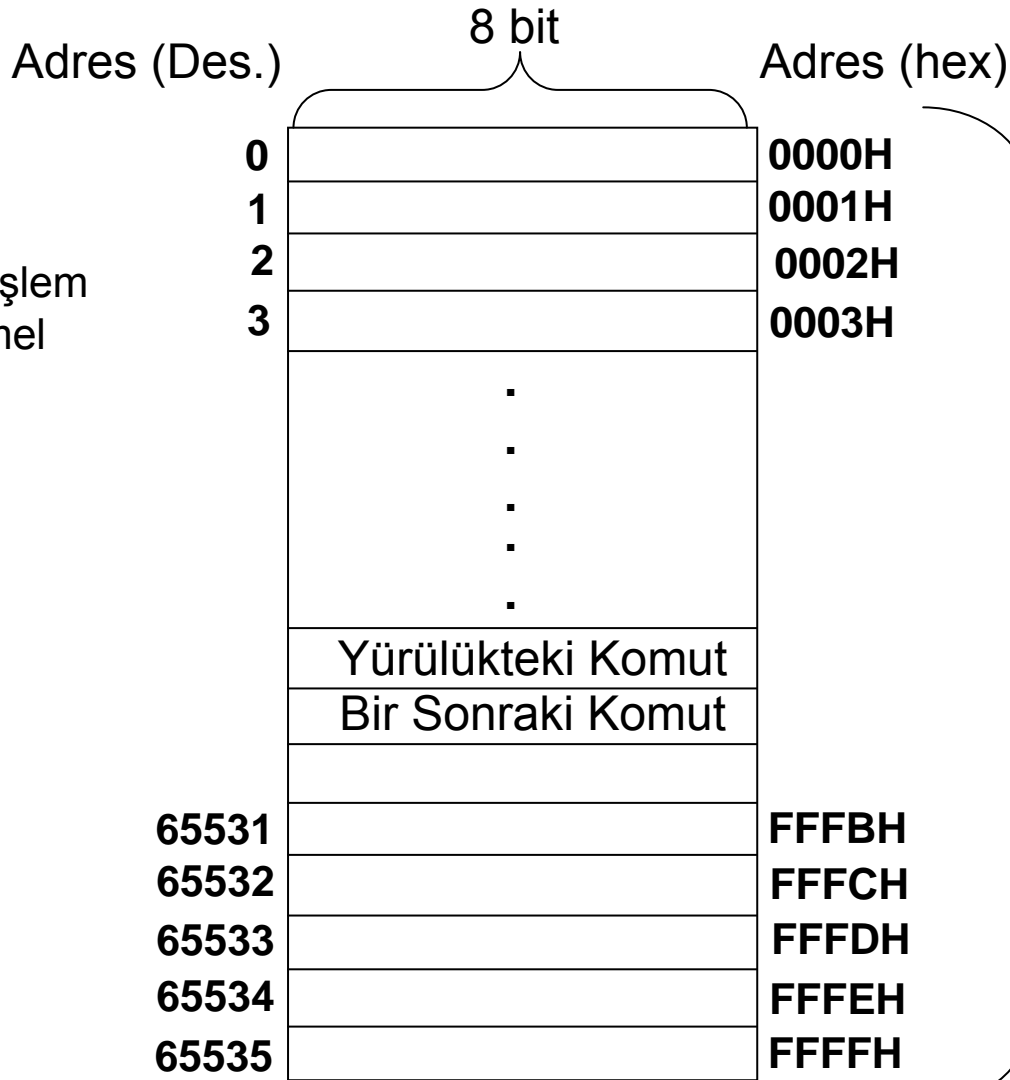
# Von Neumann Modeli



# Bellek (Memory)

Bellek tümdevresinin kapasitesi arttıkça kullanılacak uç sayısı da **logaritmik** olarak artmaktadır.

-  Temel bellek birimi gözedir.
-  Gözelerin yanyana gelmesiyle gözler oluşur.
-  Gözelerin bir araya gelmesiyle de bellek oluşur.



Bellek üzerinde işlem yapan temel komutlar LOAD ve STORE

16 bit adres hattına sahip bir sistemin bellek yapısı



# Bellek Arayüzü (Memory Interface)



Mikroişlemci, bellek ile MAR (Memory Address Register) ve MDR (Memory Data Register) olarak bilinin iki yapı yardımıyla iletişim kurar.



 Bellekteki X konumundan bir okuma işlemi gerçekleştirileceği zaman:

- Adres X ilk önce MAR kaydedicisine yerleştirilir.
- Belleğe bir okuma sinyali(RD) gönderilir.
- X konumunun içerdiği veri değeri MDR kaydedicisinde elde edilir.

 Bellekteki X konumuna T değerinin yazılması gerektiği durumda

- T verisi önce, MDR kaydedicisinin içerisine yerleştirilir.
- X bellek adresi MAR kaydedicisine yerleştirilir.
- Ardından mikroişlemci denetim birimi tarafından üretilen yazma sinyali (WR) ile bu konuma T verisi yazılmış olunur.

# Merkezi İşlem Birimi (CPU)



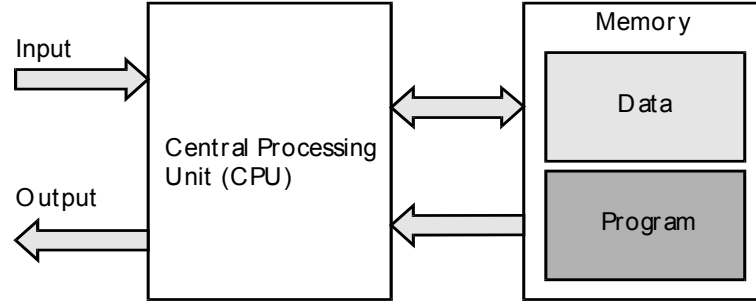
Mikroişlemci, bellekten komutları okuyan, çalıştıran ve giriş-çıkış cihazlarını kontrol eden birimdir. Bilgisayarın beynidir denilebilir.



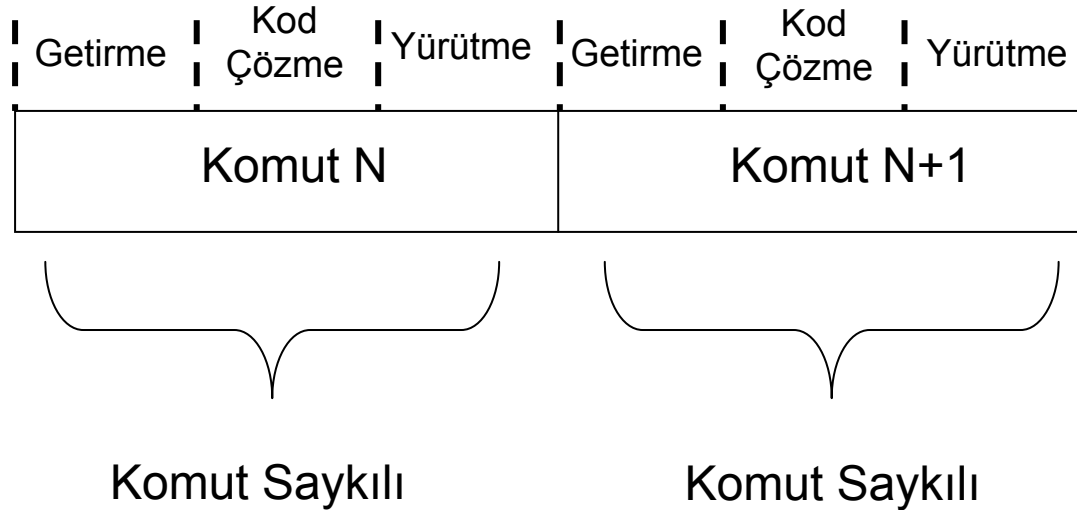
Bilgisayar içerisinde olan işlem akışlarını kontrol eder ve düzenler.



Bütün matematiksel ve mantıksal işlemleri gerçekleştirir.

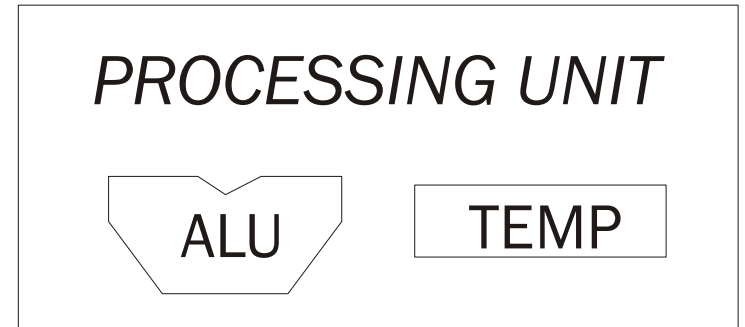


Mikroişlemci, bellekten komutları okuyan, yürüten ve giriş-çıkış cihazlarını kontrol eden birimdir.



## ✓ Arithmetical and Logic Unit (ALU)

Temel aritmatiksel (+,-,x,/) ve mantıksal (AND,OR,NOT...) işlemleri gerçekleştirir.



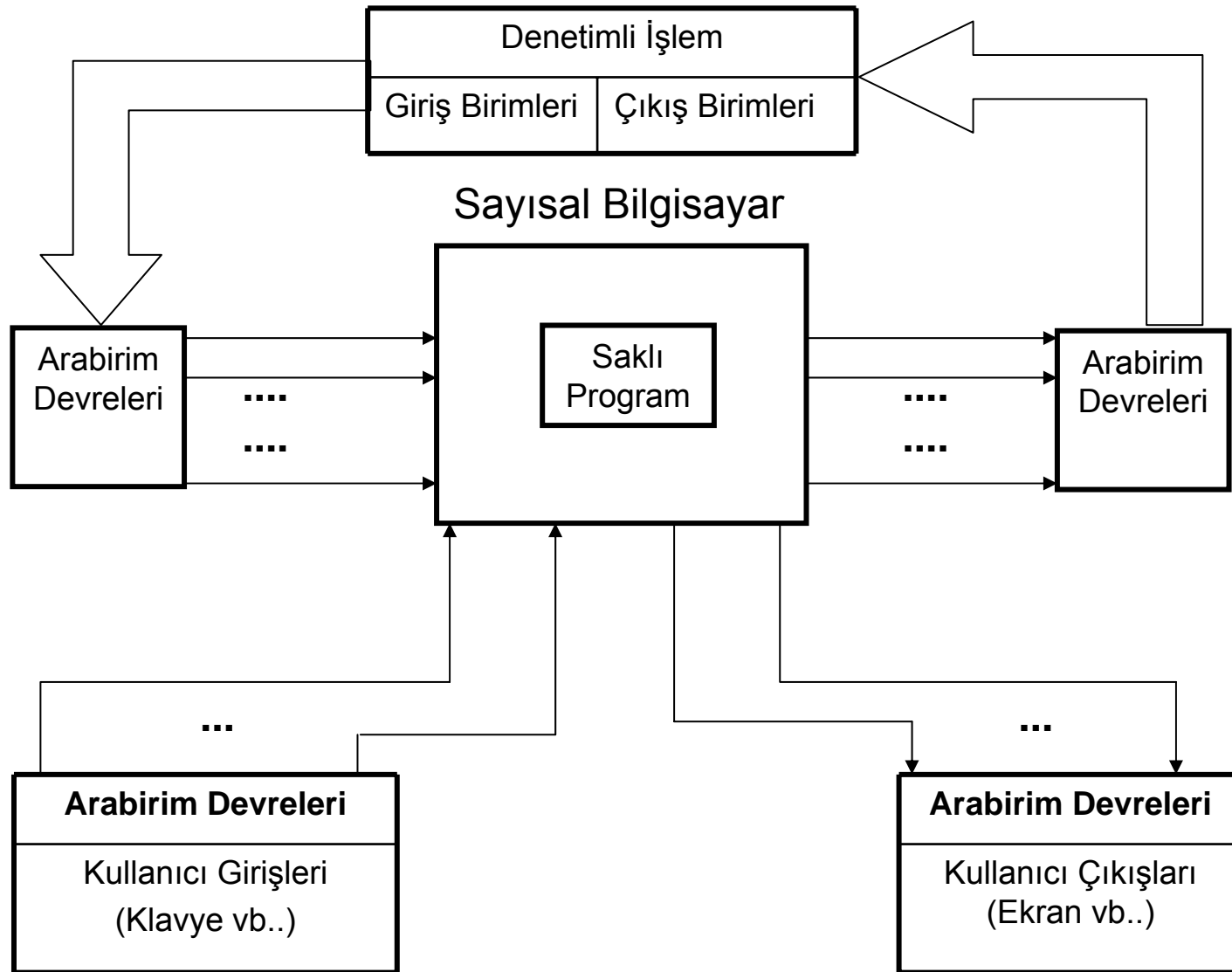
## ✓ Kaydediciler (Registers)

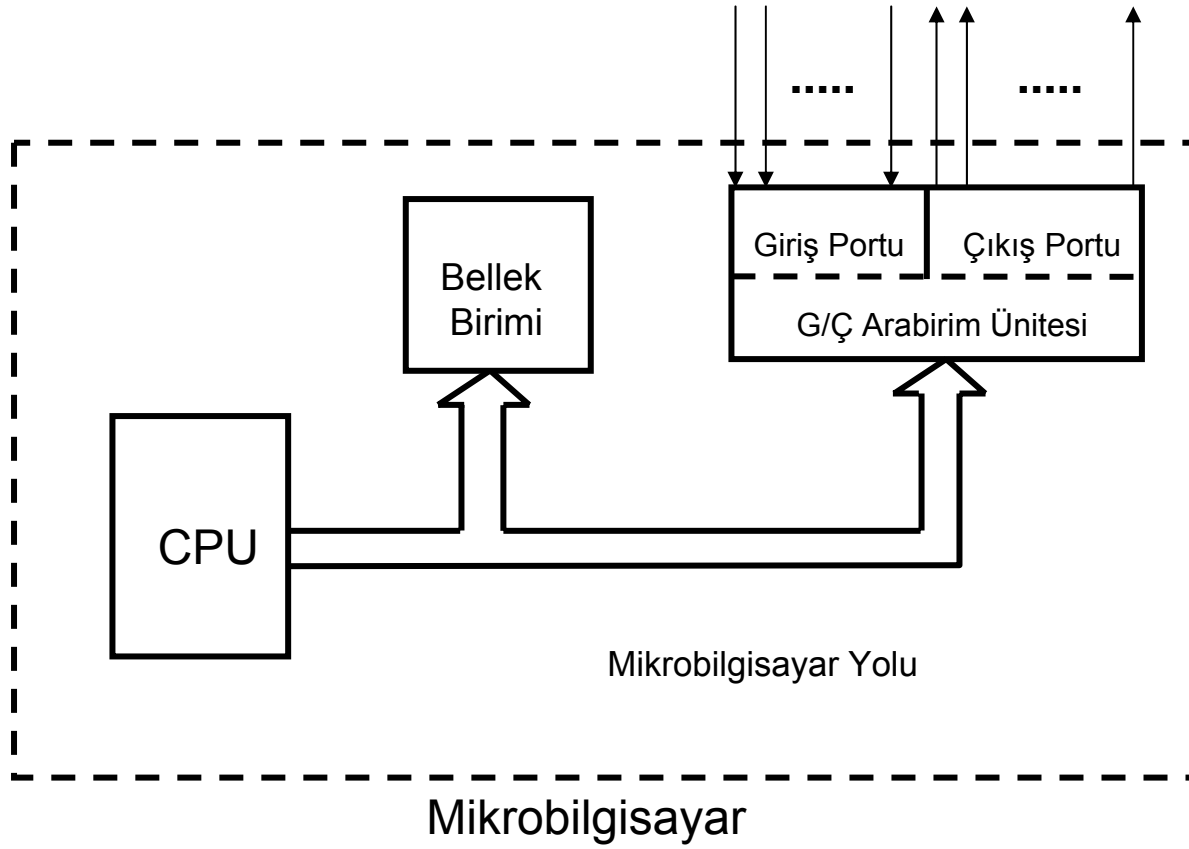
Üzerinde işlem yapılacak veri ve/veya komut parçacıklarının mikroişlemci içersinde saklayan gözlerdir. Bellek gözleri ile benzerlik taşır.

## ✓ Kelime Uzunluğu (Word Size)

Paralel olarak işlenen veri bitlerinin sayısıdır. Kelime, işlemcideki genel amaçlı kaydedicilerin büyüklüğü ve aynı zamanda her bir bellek alanı kapasitesidir.

# Giriş/Çıkış (Input&Output) Arabirimleri





# Kontrol Birimi (Control Unit)



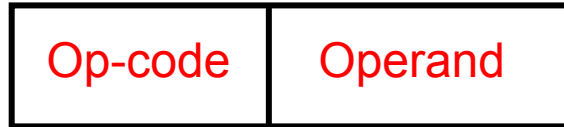
- ✓ Komutların yürütümünü gerçekleştirir. Sistem üzerindeki senkronizasyonu sağlar. Sistemdeki diğer birimlere ne yapmaları gerektiğini söyler.
- ✓ Komut Kaydedicisi (Intruction Register) : O an yürürlükte olan komutun tutulduğu kaydedicidir.
- ✓ Program Sayıcısı (Program Counter) : Bellekte yürütülecek bir sonraki komutun adresini tutar.



# Komut



Komut, yapılan bir işin en temel parçasıdır.



**Op-code (İşlem Kodu):** CPU'ya hangi işlemin gerçekleştirileceğini söyler.

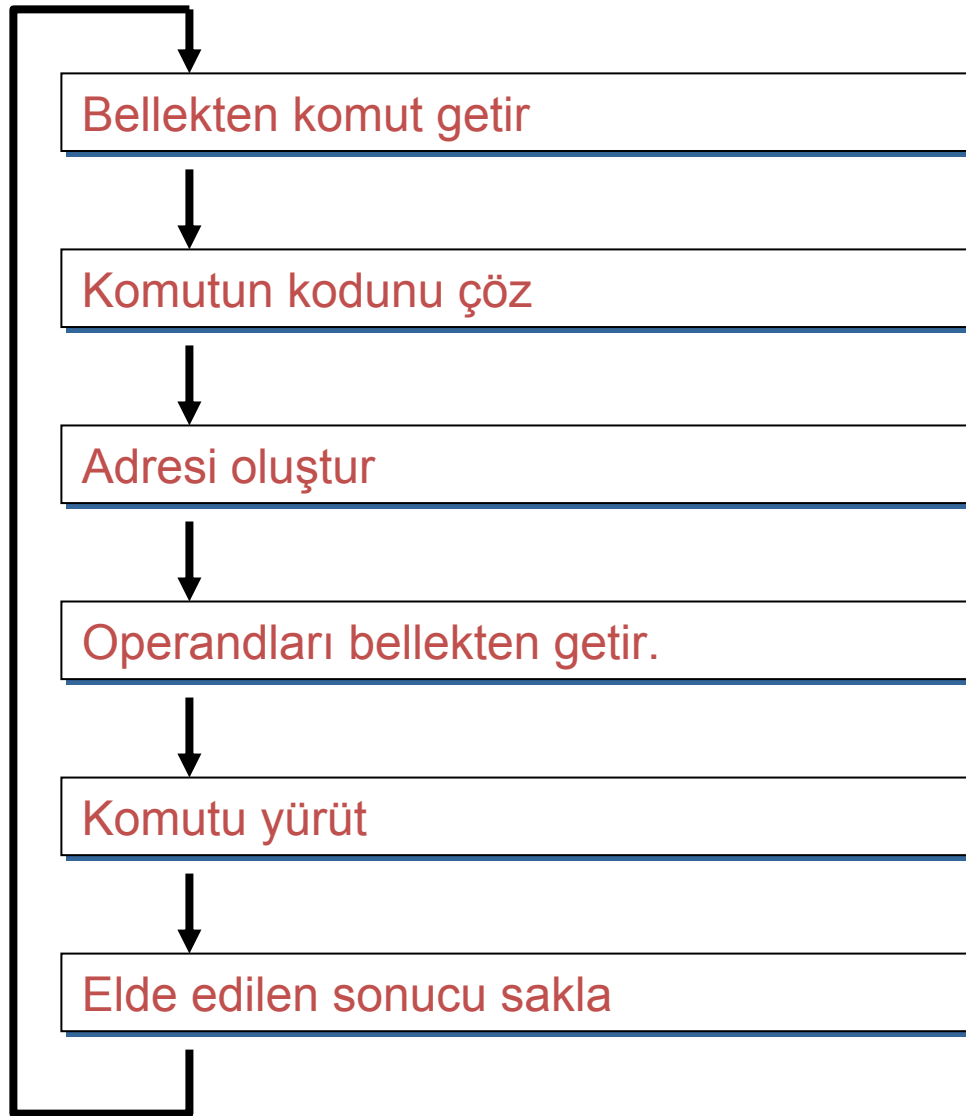
**Operand (İşlenen):** Op-code tarafından kullanılacak verinin adresini belirtir.

Mikroişlemci içindeki komutlar birden fazla operanda sahip olabileceği gibi hiç bir operandı olmayan komutlarda mevcuttur.



Bilgisayarın komutları ve formatları, komut seti mimarisi olarak bilinir (**Instruction Set Architecture-ISA**)

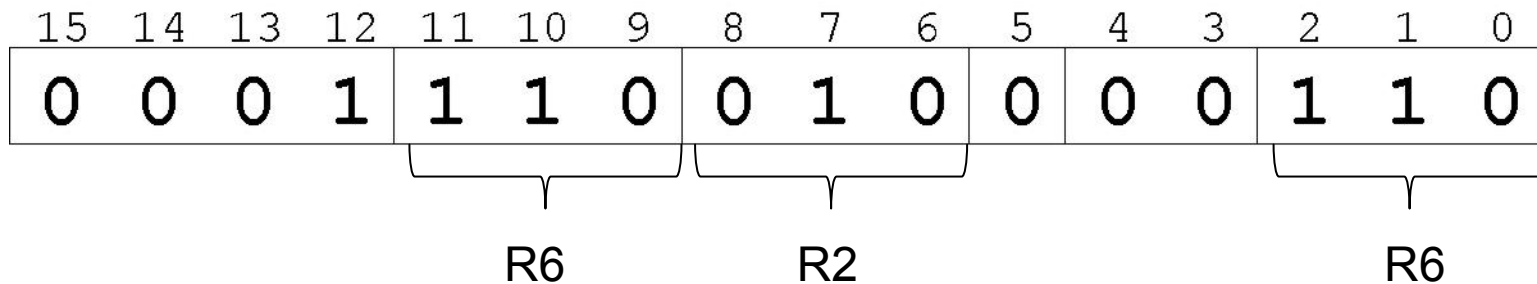
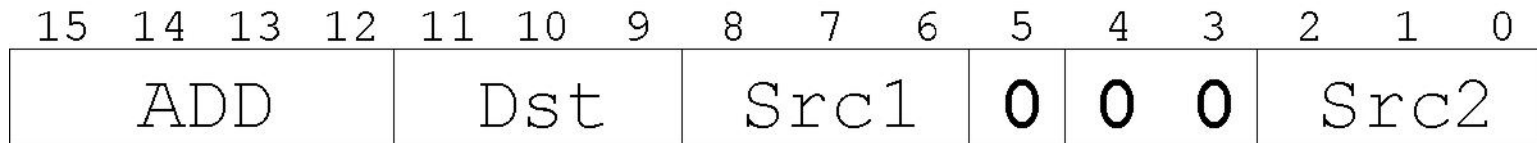
# Komut İşleme



## Örn: T mikroişlemcisinde ADD komutunun yürütülmesi

✓ T mikroişlemcisi 16 bitlik komut setine sahiptir. 16 bitlik komut setinin 4 biti işlem kodu (op-code) için ayrılmıştır. [12-15]

✓ T mikroişlemcisinin R0,...,R7 olmak üzere 8 adet kaydedicisi vardır.



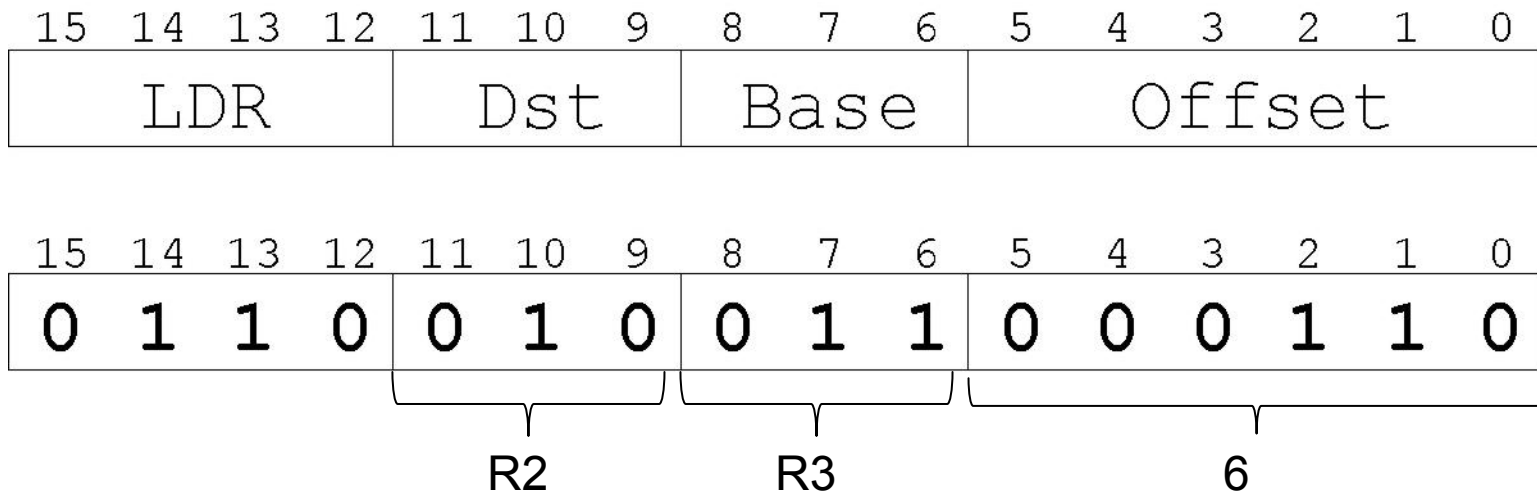
Bu komut, R2'nin içeriğini R6'nın içeriğine ekler ve sonucu R6'da saklar.

## Örn: T mikroişlemcisinde LDR komutunun yürütülmesi

- ✓ Veriyi bellekten okur ve okunan değeri belirtilen kaydedicilerden birisinde saklar.

$$\text{Offset Address} + \text{Base Address} = \text{Memory Address}$$

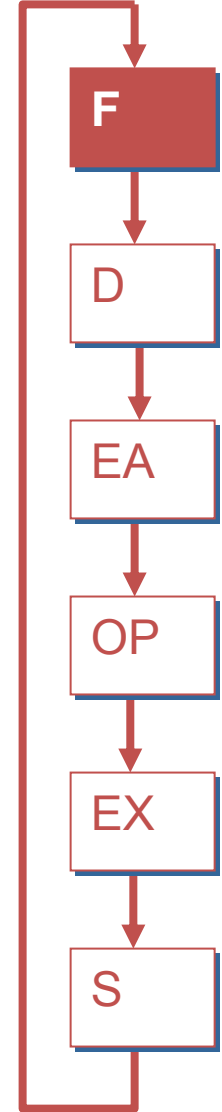
Memory Address ile belirtilen adresteki değeri “0 1 0” ile belirtilen hedef kaydedicide saklar.



Bellek adresini oluşturmak için 6 değerini R3 kaydedicisinin içeriğine ekler. Ardından oluşturduğu bellek adresinin içeriğini R2 kaydedicisinde saklar.

# Komut İşleme: Getir (Fetch)

- ✓ PC'nin içeriğini bellek adres kaydedicisine yükle (MAR)
- ✓ Bellekten bir okuma işlemi gerçekleştirir.
- ✓ Bellek veri kaydedicisinin (MDR) içeriğini oku ve bunu daha sonra komut kaydedicisine (IR) yerleştir.
- ✓ PC'nin içeriğini bir sonraki komutu gösterecek şekilde 1 artır.



# Komut İşleme: Kod Çözme(Decode)

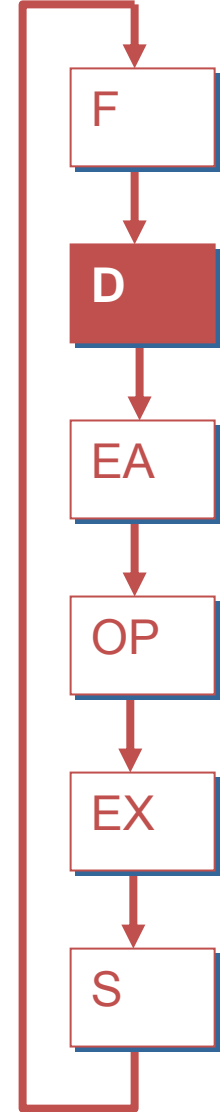
- ✓ Komutun işlem parçasını (op-code) tanımla.
  - T mikroişlemcisinde bu komutun en yüksek değerlikli ilk 4 bitidir.

- ✓ Mikroişlemci komutun op-code kısmına bakarak operand kısmı için kaç byte getireceğini bilir.

Örn:

LDR komutunun son 6 biti offset adresini gösterir.

ADD komutunun son 3 biti işlem parçasının ikinci kısmını (operand 2'yi gösterir)



# Komut İşleme: Adresin Hesaplanması (Evaluate Address)



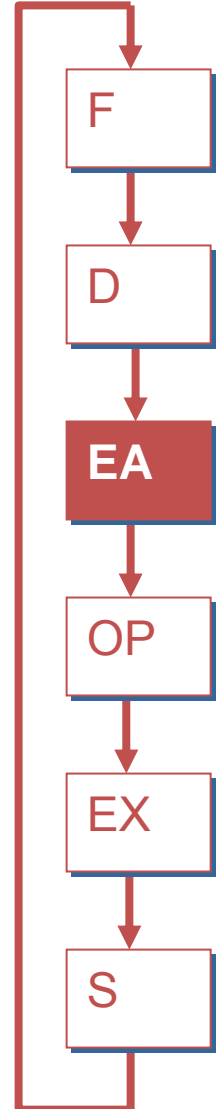
Bellek erişimine ihtiyaç duyan bir komut, erişim için gerekli olan adresi hesaplar.

Örn:

Offset Address + Base Address = Memory Address

Offset Address + PC = Memory Address

Offset Address + 0 = Memory Address



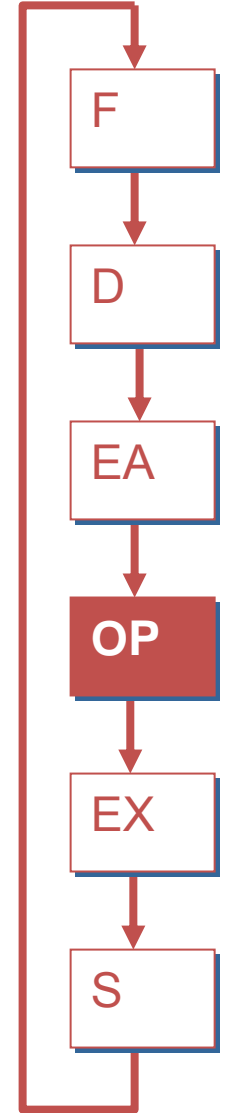
# Komut İşleme: İşlenen Getir (Fetch Operand)



Komutun üzerinde işlem yapacağı parçaları getirir.

Örn:

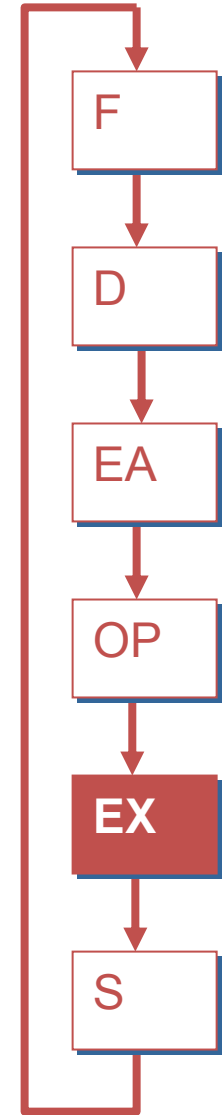
- Bellekten verinin yüklenmesi
- Kaydediciden bir değerin okunması





# Komut İşleme: Yürütme (Execution)

- ✓ Mikroişlemci önceden okumuş olduğu op-code ve operandları kullanarak ilgili işlemi yerine getirir.



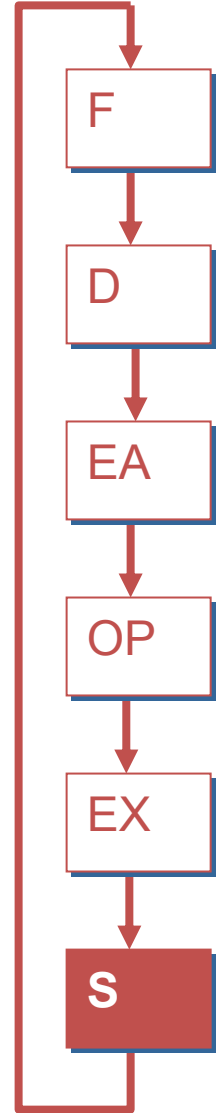
# Komut İşleme: Sakla (Store)

✓ Komut sonucunda, elde edilen değer saklanır. Bu bir bellek konumu olabileceği gibi, bir kaydedici veya bir çıkış portu da olabilir.




Örn:

Bellekteki bir konuma sonucun saklanacağı durumda

- Önce ilgili konumun adresi MAR kaydedicisine yerleştirilir
- Kaydedilecek veri MDR kaydedicisine yerleştirilir.
- Ardından belleğe yazma işlemi gerçekleştirilir.



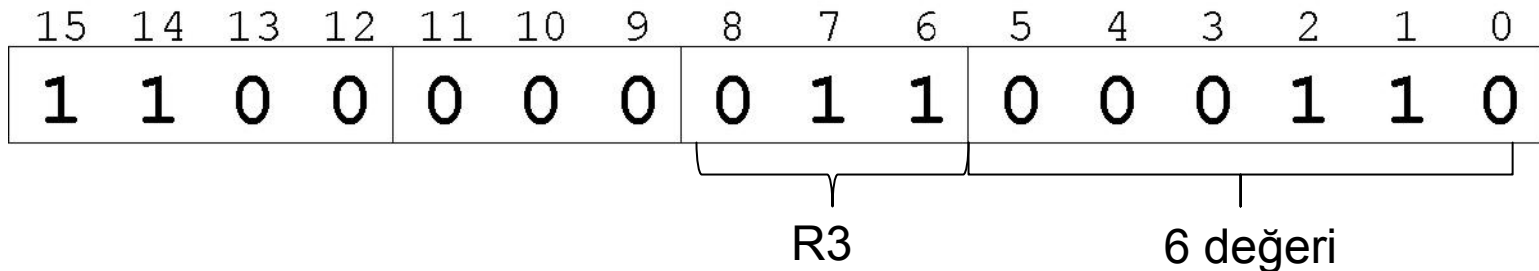
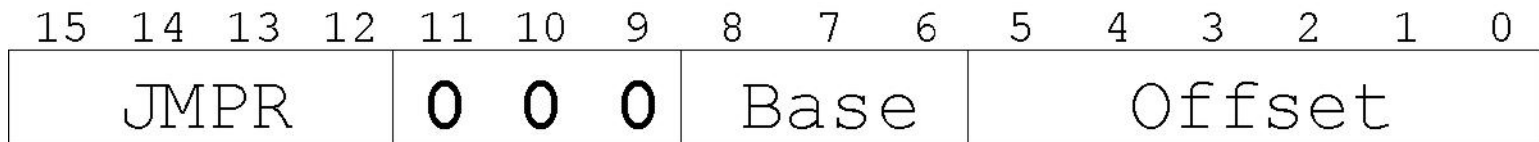
# Komut İşlenme Sırasını Değiştirme

-  Buraya kadar belirtilen işlemlerde bellekteki komutlar sırasıyla işlenmekteydi. Bu durumda PC'in içeriği sırasıyla birer birer artmaktaydı.
-  Bazı durumlarda bu tür kullanımın dışına çıkılarak program akışının yönünü değiştirmek mümkündür. **Örn:** Döngüler(loop), Şartlı yapılar (if-then) ve alt yordamlar (sub-routines) gibi.
-  Bu işlemi gerçekleştirmek için bazı komutlara gereksinim duyulur. Bunlar sıçırma(jump) ve şartlı dallanma(branch) komutlarıdır.

## Örn: T mikroişlemcisinin belleğinde sıçrama



Offset değeri bir kaydedicinin değerine eklenerek yürütülecek bir sonraki komutun adresini oluşturabilir. Bu adres PC'nin içeriğine yüklenerek programın bu adresten devam etmesi sağlanabilir.



R3 kaydedicisinin değerini taban olarak alıp bunun üzerine offset değerinin eklenmesi ile yürütülecek bir sonraki komutun bulunduğu bellek adresi elde edilir ve elde edilen bu değer PC'nin içerisine yerleştirilir.

# Komut Türleri



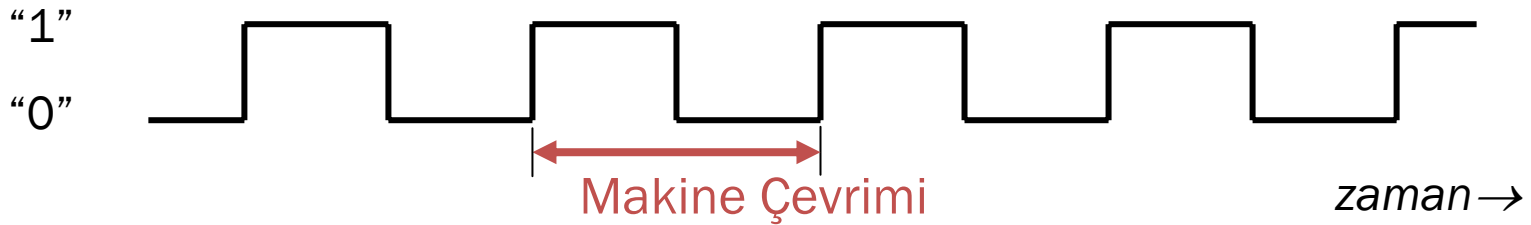
Mikroişlemci tarafından yürütülen komut türleri genel olarak aşağıdaki sınıflara ayrılabilir.

- Veri Taşıma Komutları
- Veri İşleme Komutları
- Program Kontrol Komutları
- Giriş / Çıkış Komutları

# Sistem Saati (System Clock)

✓ Mikroişlemcide kontrol biriminin çalışmasını sağlayan sistem saatidir.

Saat darbesinin her bir kenarında kontrol birimi bir sonraki makine çevrimine girer. Bu çevrim yeni bir komut olabileceği gibi yürürlükteki komutun farklı bir evresi de olabilir.



Sistem saati, kristal bir osilatör devre yardımıyla yürütülür. Sistem saati düzenli olarak "1" ve "0" lar üretir.

# Mhz ve MIPS

MIPS = millions of instructions per second

MHz = millions of clock cycles per second

 Bu iki kavram birbirine benzer olmasına karşı farklıdırlar. **Neden?**

# Von Neumann Modeli

