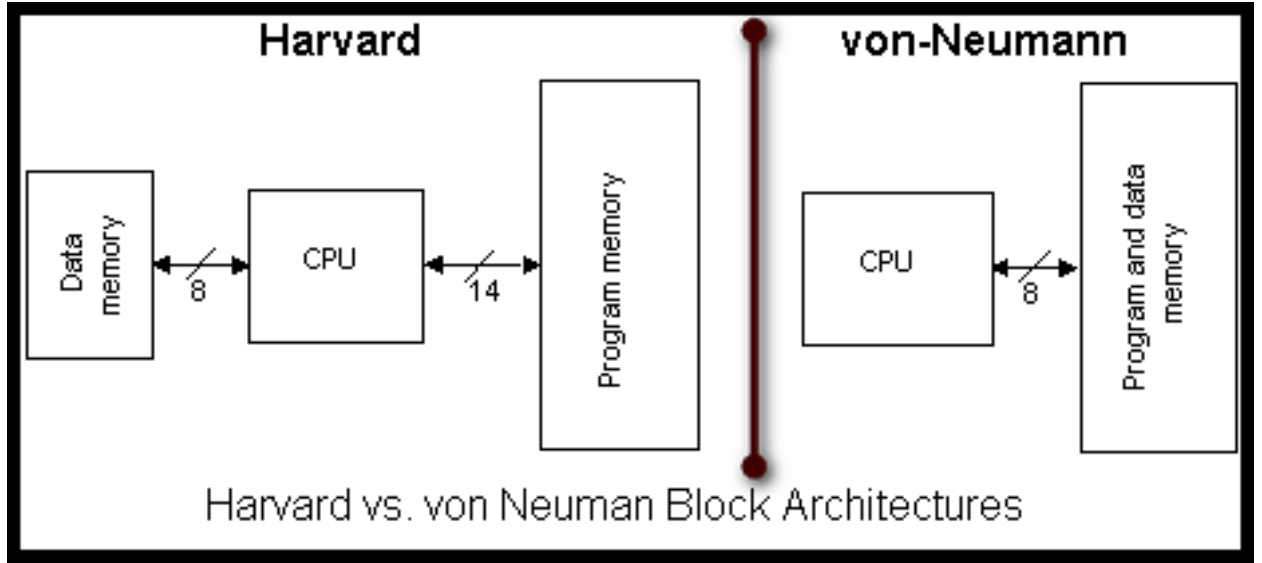


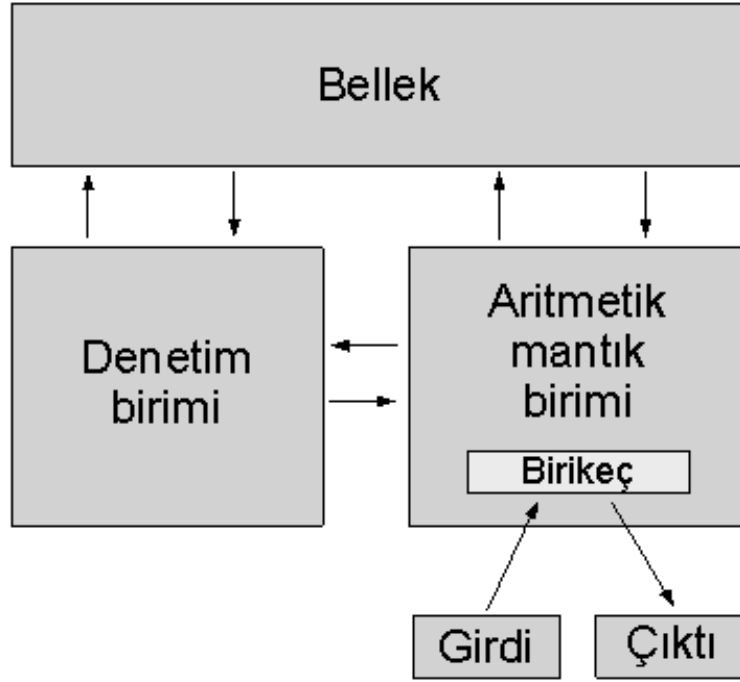
Harvard & Von Neumann Mimarileri

Harvard mimarisi, veri ve komutların Merkezi İşlem Birimine (*MİB* veya *CPU*) giden kanallarının **ayrılması** ile oluşturulmuş bilgisayar tasarımıdır. Von Neumann mimarisi, veri ve komutları **tek bir yığın** (*depolama*) biriminde bulunduran bilgisayar tasarımıdır.

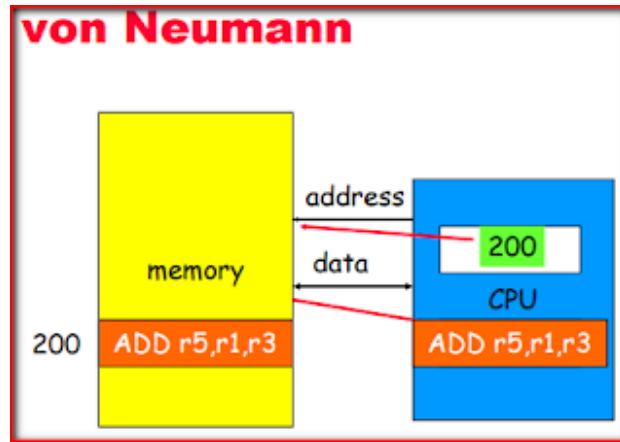


Von Neumann mimarisi

Verilerin ve program kodlarının aynı hafıza birimi üzerinde bulunduran tasarımıdır. Von Neumann mimarisinin temel tasarımı aşağıdaki gibidir:



"**von Neumann mimarisi**" isim olarak John von Neumann'ın 1945 tarihli makalesine dayanır. Bellek ile Merkezi işlem biriminin (*MİB*) ayrılması **von Neumann dar geçidi** olarak bilinen soruna yol açmıştır. Bu sorun MİB ile bellek arası veri taşıma hızının, bellek miktarına göre çok düşük olmasından kaynaklanmaktadır. Bu nedenle, CPU zamanının büyük çoğunluğunu bellekten istenilen verinin gelmesini beklemekle geçirir. Son yıllarda CPU'ların hızları ile bellek erişim hızlarının arasındaki farkın açılması ile bu sorun daha da büyümüştür. Sorunu hafifletmek adına *cache memory* ve *branch prediction* geliştirilmiştir. von Neumann mimarisinin dar geçit sorunu dışında, en olumsuz yanı ise hatalı yazılımların (*buffer overflow gibi*) kendilerine, işletim sistemine ve hatta diğer yazılımlara zarar verebilme olasılığıdır.

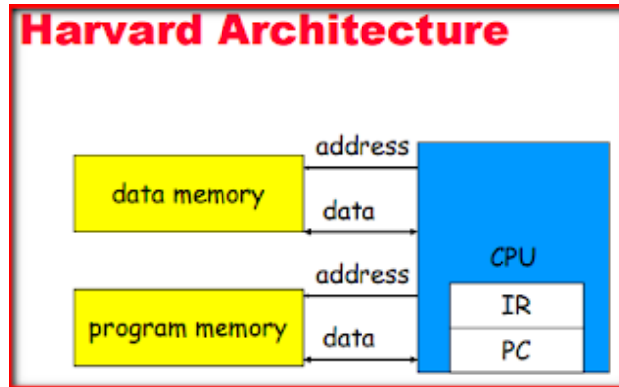


Harvard mimarisi

Harvard mimarisi, ismini ilk kez bu mimariyi kullanan bilgisayar **Harvard Mark I**'den almıştır. Bu mimariyi kullanan makinalar, veriler ile komutlar arasında herhangi bir köprü bulundurmazlar. Veri adresi ile program (*komut*) adresinin adresleme boyutları farklıdır. Harvard mimarisinin temel tasarımı aşağıdaki gibidir:



Günümüz bilgisayarlarında tam anlamıyla kullanıldığı söylenilemez. Yine de Von Neumann mimarisi ve Harvard mimarisinden ortak özellikler günümüz teknolojisinde kullanılmaktadır.



Von Neumann - Harvard karşılaştırması:

- Von Neumann mimarisinde, işlemcinin doğası gereği ya **komutlarla** ya da **verilerle** uğraşmaktadır. Çünkü ikisi de aynı belleği paylaşmaktadır. İkisinin aynı anda olması durumu söz konusu değildir.
- Harvard mimarisini kullanan bir bilgisayarda ise komut ve veriler ayrı tutulduğu için, işlemci aynı esnada hem komutları değerlendirip hem verileri işleyebilir. Bir önbelleğe de gerek yoktur. Bu Harvard mimarisine bir avantaj sağlasa da; Von Neumann mimarisinde komutlar verilerle bir tutulduğundan, program kendi kendine değişim gösterebilir.
- Harvard mimarisinde komutlar ile veriler arasında bir kanal yoktur bu yüzden kodların içine veri gömülmüş programlar çalıştırılırken veya kendi kendine değişim gösterilecek programlar için Von Neumann mimarisi temel alınır.
- Bellek adresleri açısından Harvard mimarisi iki ayrı adres kullandığından; boş komut adresi boş veri adresinden de farklı olacaktır. Von Neumann mimarisinde ise ikisinde aynı adresi paylaşır.

Değiştirilmiş Harvard mimarisi

Değiştirilmiş Harvard mimarisi, Harvard mimarisindeki **veri/komut** bağlantısının eksikliğini gidermesi amacıyla yapılan tasarımsal düzenlemelere verilen isimdir. İşlemci halen veri ve komut erişimine de sahip olsa da aralarında bağlantılar mevcuttur. Bu düzenlemelerden en önemlisi aynı bellek adresi tarafından desteklenen iki ayrı önbellek kullanmasıdır. Biri komutları, biri verileri tutar. **Önbellekte işlem** yapılırken Harvard mimarisinin, **asıl bellekte işlem** yapılırken Von Neumann mimarisinin uygulandığını söyleyebiliriz. Günümüzde kullanılan yeni bilgisayar teknolojilerinde buna benzer mimariler sıkça kullanılmaktadır fakat isimlendirilmesinde ne Von Neumann ne de Harvard demek doğru değildir. Ayrıca komutları, okunabilir veriler gibi göstermek de değiştirilmiş Harvard mimarisine örnek gösterilebilir.

CISC ve RISC Mimarileri

Giriş

Komut Kümesi: Programcının makinayı programlarken kullanabileceği ilkel emirleri veya makina komutlarının tamamının oluşturduğu kümeyi belirtir. Komut setinin karmaşıklığı aşağıdaki faktörlere bağlıdır:

- Komut ve veri formatlarına
- Adresleme modlarına
- Genel amaçlı kaydedicilere
- Op-code tanımlamalarına
- Kullanılan akış kontrol mekanizmalarına bağlıdır.

Mimari Modeller

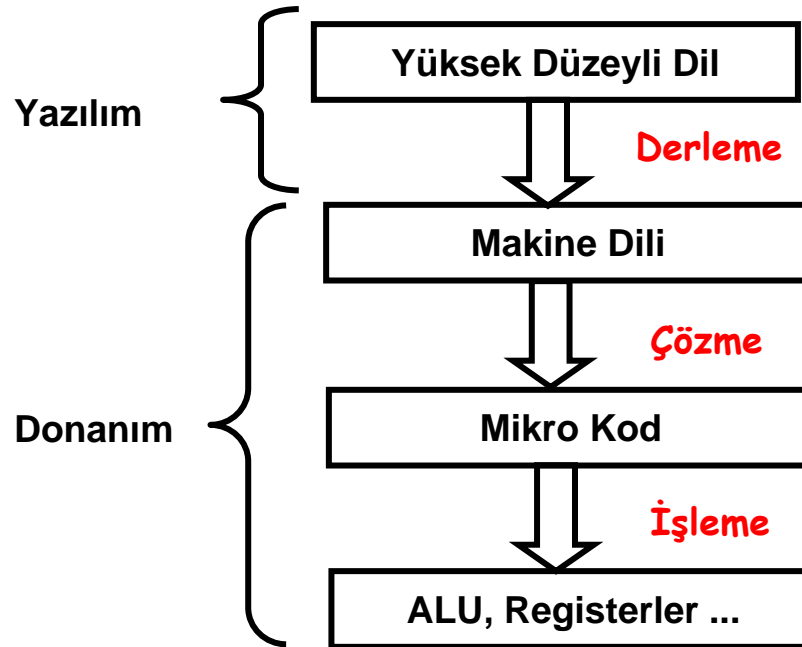
İşlemci tasarımları 2 farklı mimaride şekillenir:

- i) CISC (Complex Instruction Set Computer)
 - Intel, AMD vb...
- ii) RISC (Reduced Instruction Set Computer)
 - Power PC, Sun Sparc vb...

CISC Mimarisi

- 70'li yıllarda geliştirilen bu mimari, programlanması kolay ve etkin bellek kullanımı sağlayan tasarım felsefesinin ürünüdür.
- CPU karmaşık hale gelmekte fakat bu CPU üzerinde koşacak programları daha basitleştirmektedir.
- **En önemli özelliği** değişken uzunlukta ve karmaşık yapıda olan komutları barındırmasıdır. Böylece bellek tasarrufu sağlanır.

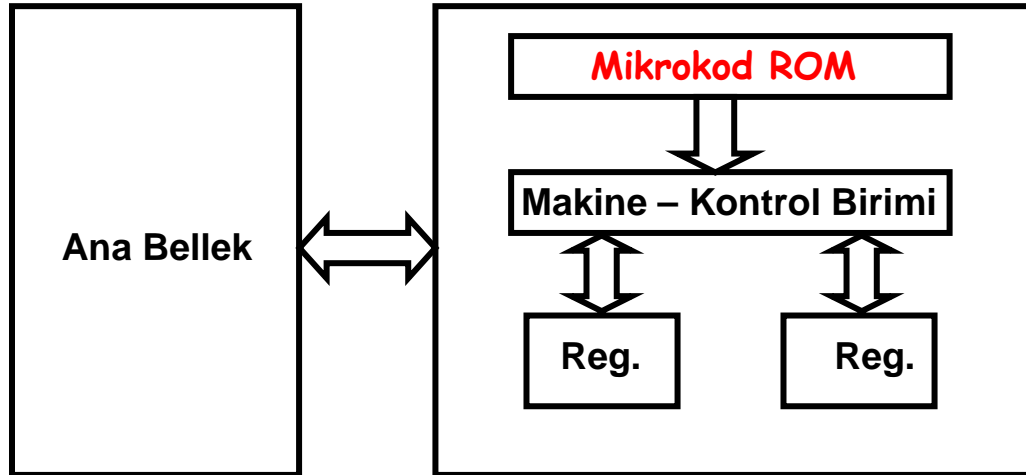
CISC Mimarisi (Devam)



CISC tabanlı bir işlemcinin çalışma biçimi

Mikrokod

İşlemcinin, komut kodlarının her birine karşılık gelen mikrokod komut gruplarını içeren ROM belleği vardır. Bir makine kodu işlemciye eriştiğinde, işlemci kodun daha basit komutlara ayrılmış parçalarını yürütür.



Mikrokod (Devam)

Bu yapıda, komut kümesi aynı kaldığı için programlar, farklı sistemler üzerinde yeniden derlemeye gerek kalmaksızın çalıştırılabilirler.

Geliştirilen yeni komutlar eskilerinin üzerine eklenerek geriye doğru olan uyumluluk sağlanır.

CISC Mimarisinin Avantajları

- Mikroprogramların yürütülmesi kolaydır.
- Geriye doğru uyumludur.
- Mikrokod ROM'A eklenen her bir komut ile CPU daha yetenekli olmaya başlamakta ve verilen bir görevi yürütmek için daha az zaman harcamaktadır.
- Bu tip mimaride yapılan derleyiciler karmaşık olmak zorunda değildir.

CISC Mimarisinin Dezavantajları

- CPU yapısı her kuşak işlemci ile beraber daha karmaşıklaşmıştır.
- Özel olarak tasarlanmış olan komutlar yeteri kadar sık kullanılmamaktadır. Tipik bir programda mevcut komutların yaklaşık olarak %20'si kullanılmaktadır.
- Farklı komutlar farklı sayıda saat çevrimine gerek duyacaklarından performans düşmesi gözlenir.

RISC Mimarisi

- CISC mimarisi ile sistem daha karmaşık hale gelmekte ve basitleştirmek için yeni bir yaklaşım olarak RISC mimarisi doğmuştur.

RISC mimarisi üç temel prensibe dayanmaktadır :

- Bütün komutlar tek bir çevrimde çalıştırılmalıdır.
- Belleğe sadece “Load” ve “Store” komutlarıyla erişilmelidir.
- Bütün icra birimleri mikrokod kullanmadan donanımdan çalıştırılmalıdır.

RISC Mimarisi Özellikleri

- Küçültülen komut kümesi ve azaltılan adresleme modlarına sahiptir.
- Bütün icra birimleri mikrokod kullanılmadan donanımsal olarak çalışmaktadır.
- Yüksek seviyeli dilleri destekleme
- Çok sayıda kaydediciye sahip olması

RISC Mimarisinin Avantajları

- Hız: Azaltılmış komut kümesi sayesinde daha hızlı çalışırlar.
- Basit donanım: RISC işlemcilerin komut kümeleri basitleştirildiklerinden çok az yonga kullanırlar.
- Kısa tasarım zamanı: CISC işlemcilere göre daha çabuk tasarlanabilirler.