

MİKROİŞLEMCİLER VE ASSEMBLY PROGRAMLAMA



Mikroişlemciler

İlk mikroişlemci, 1971 yılında Intel tarafından piyasaya sürülen 4004'dür.

- Saat Hızı 740 kHz
- 0.06 MIPS (Million instructions per second, saniyede milyon işlem)
- Yol genişliği 4 bit
- Transistör sayısı 2 300
- Adreslenebilir Hafıza 640 byte
- Program Hafızası 4 KB
- PMOS (positive-channel metal oxide semiconductor)

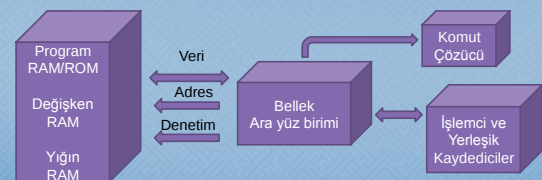
Mikroişlemciler

Üretici	İşlemci	Saat Frekansı	Çekirdek Adeti	Ortalama MIPS
Intel	Core 2 Duo E6600	2400 MHz	2	4660
AMD	Athlon 64X24600+	2400 MHz	2	4162
Intel	Pentium Pro	200 MHz	1	440
AMD	K5-PR	166 MHz	1	260
Intel	Pentium	133 MHz	1	240
Intel	486DX4	100 MHz	1	60
Motorola	68030	40 MHz	1	10
Intel	386DX	33 MHz	1	10
Motorola	68000	8 MHz	1	0,7
Intel	8080	2 MHz	1	0,5
Intel	4004	0,1 MHz	1	0,06

Mikroişlemci Mimarileri

Bellek Yönetimi Açısından

Von Neuman:



Mikroişlemci Mimarileri

Bellek Yönetimi Açısından

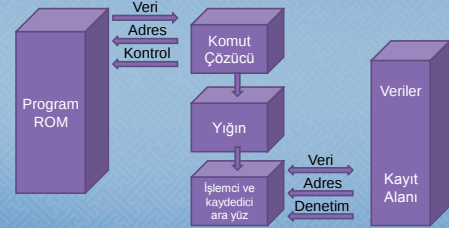
Von Neuman:

- Basit ve ekonomik organizasyon sunar.
- Tek adres alanı vardır.
- Program kodları ve veriler aynı bellek alanında olmasından dolayı karışma ihtimali vardır ve performansı düşürür.
- Kod, adres ve veriler aynı alanda olduğundan gerçek kapasitenin yarısını kullanır.

Mikroişlemci Mimarileri

Bellek Yönetimi Açısından

Harvard:



Mikroişlemci Mimarileri

Bellek Yönetimi Açısından

Harvard:

- Kod okuma ve veri okuma/yazma aynı anda yapılabilenlerinden ayrı birimlerde olmasından dolayı başarımlı yüksektir.
- Adres alanı iki kat daha fazladır.
- Tasarım ve maliyeti düşüktür.

Mikroişlemci Mimarileri

Komut İşleme Tekniği Açısından

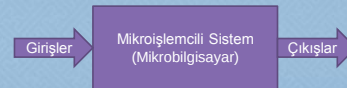
- Karmaşık Komut Kümeli Bilgisayarlar (CIS)
- Azaltılmış Komut Kümeli Bilgisayarlar (RISC)

Mikroişlemci Mimarileri

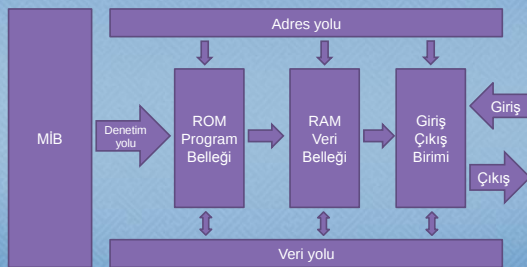
	RISC	CISC
Çevrim süresi	+Mikro-kod çevrim aşamalarından geçmek zorunda olmadığından komutlar hızlı işletilir.	- Uzun süren kod çevrim aşamaları
Programlama	+Anlaşılır ve basit kod kullanımı -Aynı işlev için daha uzun program kodu	-Karmaşık assembly komutu +Daha az saat çevriminde daha çok iş
Derleme performansı	+Tek düze komutlardan dolayı basit kod çevirimi	+Kısa sürede kod çevirimi
Donanım	+Basit komutlar ve basit kod çözücüler, daha az donanım	+ Karmaşık komutlar ve karmaşık kod çözücüler, daha çok donanım

Mikroişlemcilere Donanımsal Bakış

- Mikroişlemciler tek başına kullanılmazlar, çevre birimlerle kullanılırlar.



Mikroişlemcilere Donanımsal Bakış



Mikroişlemcilere Donanımsal Bakış

Mikroişlemciler çevresel birimlere ulaşmak için üç yol kullanır;

- Veri yolu: hem işlemcinin içinde hem de çevre birimlerden işlemciye veri akışını sağlar. Kullanılmadığı durumlarda yüksek empedans gösterir ve böylece yalıtım sağlanmış olur.
- Adres yolu: birimlere adres aracılığı ile ulaşılır. Bu ulaşım ya direkt adres bilgisinden ya da kod çözücü ile oluşur.
- Denetim yolu

Mikroişlemci ve Uygulama Programı İlişkisi

Bilgisayarda çalıştırılacak bir uygulama kullandığı **mikroişlemci ve işletim sistemi** ile uyumlu olmalıdır. Bunun sebebi derleyici ve yorumlayıcı tarafından üretilen makine kodlarının hedef işletim sistemi veya işlemci tarafından çözümlenmesidir.

Yazılım, donanım üzerine kurulabilmektedir ve bit ölçüsü bakımından aşağıdaki gibi olmalıdır;

Mikroişlemci>=İşletim sistemi>=Derleyici>=Uygulama

Assembly

Yazım sırasına göre;

- Etiket: komut satırının ilk bilgisidir ve simgesel isimlerden oluşur.
- İşlem kodu: komut kümesi içinde mikroişlemcinin belirli işi yapmasını sağlayan tanımlanmış kısaltmalardır.
- İşlenenler: işlemciye işlenecek veriyi veya verinin nere de olduğunu gösterir. Tek başına anlamsızdır. Komutun etki edeceği hedef ve kaynak bilgisini içerir.
- Açıklama: noktalı virgülden sonra başlayan satırlar açıklama içerir.

Yönergeler

Dil kapsamı içinde değildir, programın kolay çalışma sını sağlar.

ORG: kod bellek içerisinde programın başlangıç adresini belirtmek için kullanılan adres konumlandırma talimatıdır.

END: programın bittiğini gösterir.

DB: kod bellek içerisinde sayı ve stringleri tanımlar.

EQU: sayısal değerın simgeye atanmasını sağlar.

Adresleme Yöntemleri

Mikroişlemcide birden çok bellek çeşidi vardır ve her bellek için aynı yöntemleri kullanarak erişmek mümkün değildir ve verimsizdir.

Her belleğin adreslenebilecek alanı farklı boyutta adres hat sayısına ihtiyaç vardır.

Adresleme yöntemi işlenen kısımda bulunan değişkenlere ait adreslerin verimli bir şekilde çözümlenmesi işlemidir.

Adresleme Yöntemleri

- Saklayıcı adresleme
- Doğrudan adresleme
- Dolaylı adresleme
- İvedi adresleme
- Bağlı adresleme