5 Nisan 2022

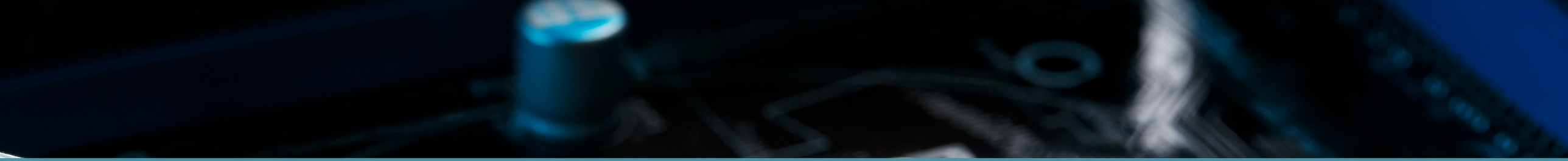
BİLGİSAYAR DONANIMI-BIL110

Öğr. Gör. Buse Yaren TEKİN

İçerikler

İşlemcinin İşleyişi Nasıl Olur?

Bu üniteyi çalıştıktan sonra;
İşlemcinin çalışmasını ve kısımlarını kavrayabileceksiniz.



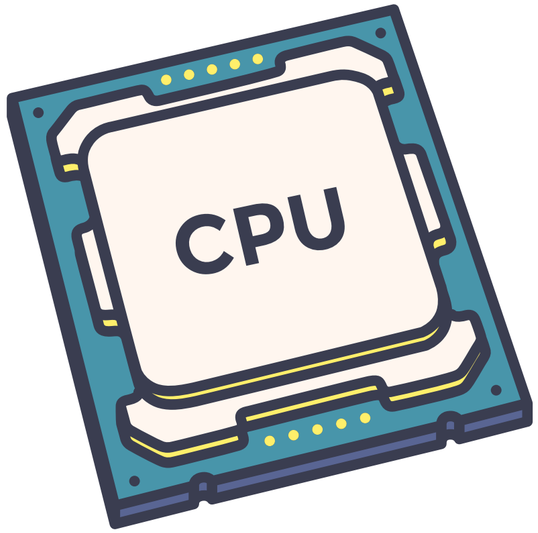
İşlemcinin İşleyişi Nasıl Olur?

Bölüm 1

Hatırlayalım - İşlemci

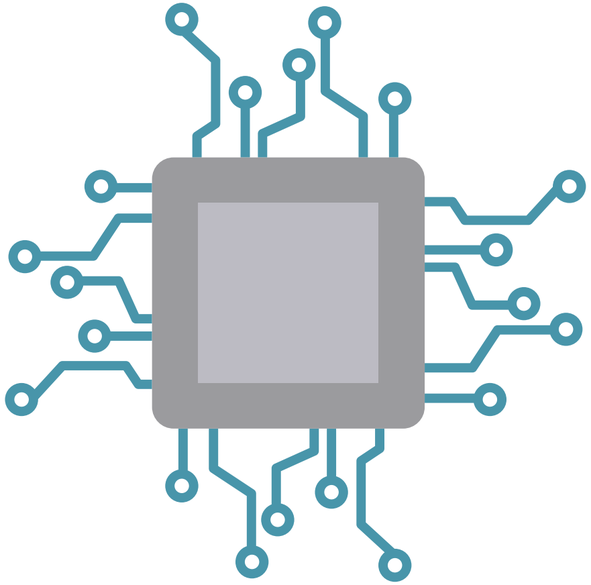
Veriler üzerindeki yaptığı işlemler, temel aritmetik işlemleri kadar ya da çok daha karmaşık gibi çeşitli seviyelerde olabilir.

İşlemciler, klavyeden girilen tuşun ifade ettiği karakteri aynen ekranda gösterme şeklinde bir işlem yaptığı gibi; aldığı verileri değerlendirip yeni veriler de üretebilir.



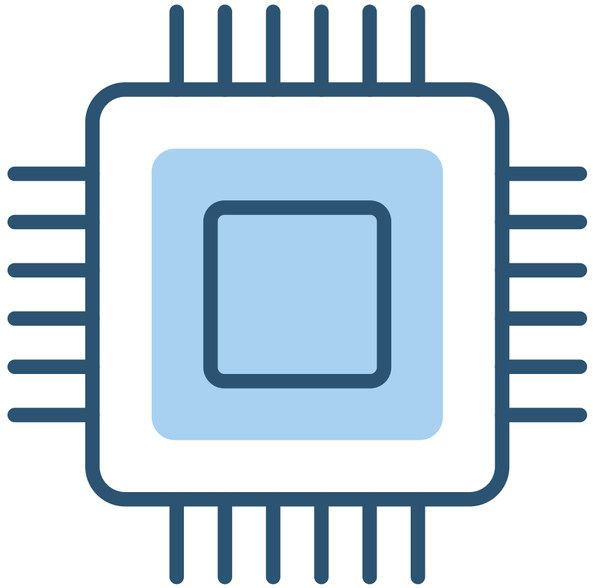
İşlemci

Çalıştırılmakta olan yazılımın içinde bulunan komutları işler. Merkezî işlem birimi, aritmetik ve mantıksal işlem yapma yeteneğine sahiptir. Giriş ve çıkış birimleri arasında verilen yazılım ile uygun çalışmayı sağlar. MİB, makine dili denilen düşük seviyeli kodlama sistemi ile çalışır; bu kodlama sistemi bilgisayarın algılayabileceği işlem kodlarından oluşur. Bir mikroişlemcinin algılayabileceği kodların tamamına o işlemcinin komut kümesi denir.



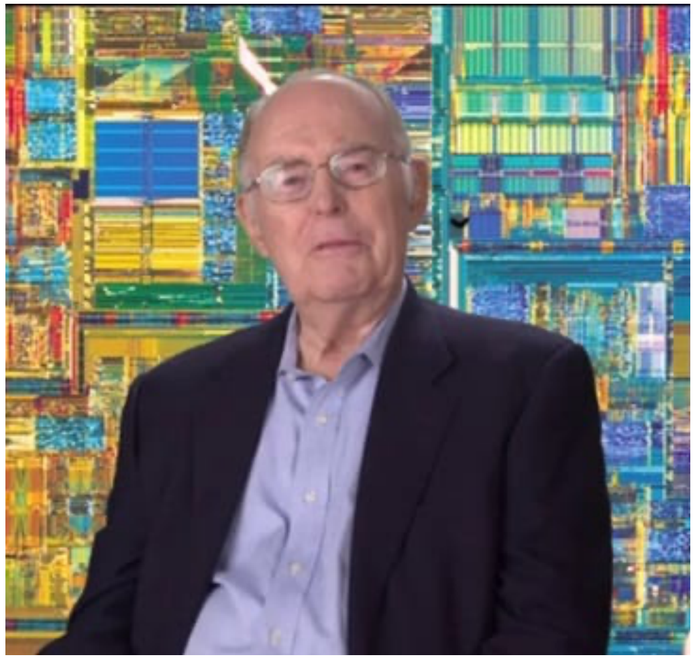
İşlemci

Çeşitli teknolojilerin daha küçük ve daha güvenilir elektronik aygıtlar üretmeye başlamasıyla MİB tasarımlarının kompleks yapıları da artış gösterdi. Bu yoldaki ilk gelişme transistörlerin gelişiyle başladı. 1950'ler ve 1960'lar da MİB'lerın transistörlere geçişi ile vakum tüpü ve elektriksel röle gibi güvensiz ve kırılgan geçiş elementleri artık kullanılmaz hale gelmişti. Bu gelişim sayesinde de, üzerinde ayrık bileşenler bulunan bir veya birden çok baskı devre kartlarına daha kompleks ve daha güvenilir MİB'ler yerleştirildi.



Moore Yasası

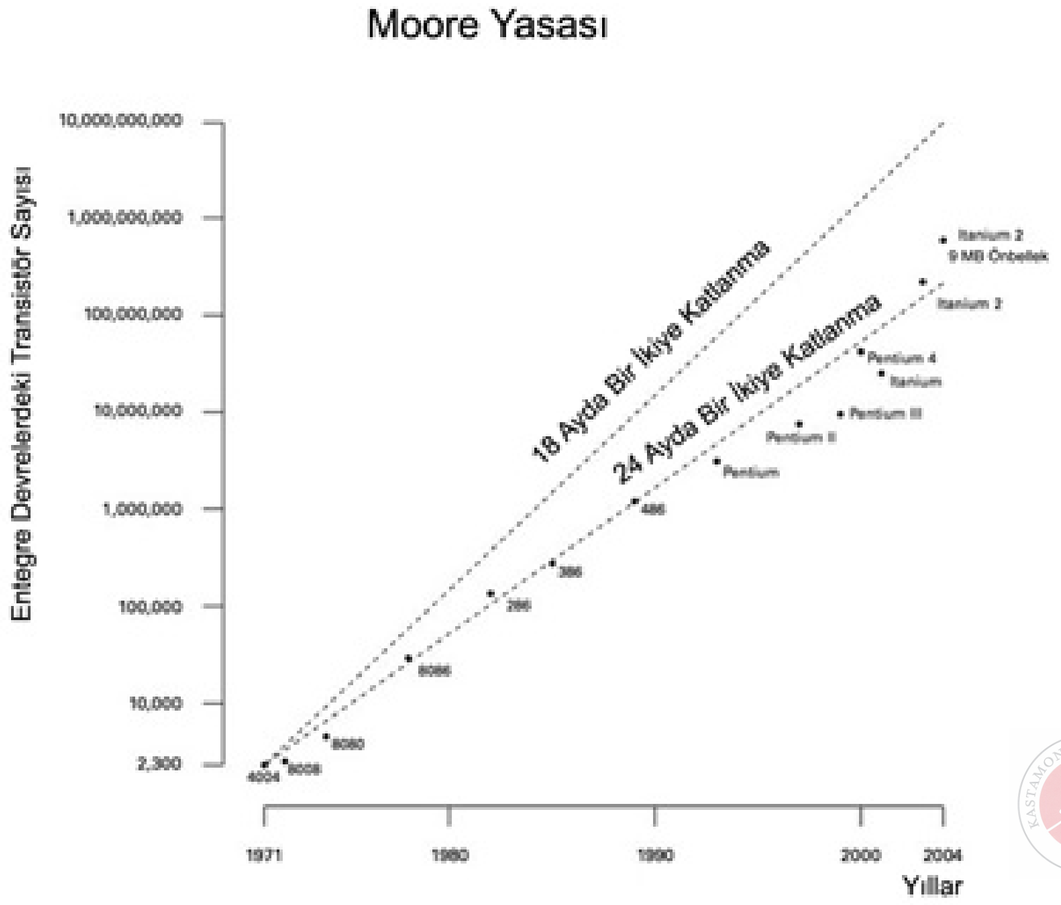
Moore Yasası, Intel şirketinin kurucularından Gordon Moore'un 19 Nisan 1965 yılında Electronics Magazine dergisinde yayınlanan makalesi ile teknoloji tarihine kendi adıyla geçen yasadır.



Her 18 ayda bir tümleşik devre üzerine yerleştirilebilecek bileşen sayısının iki katına çıkacacağını, bunun bilgisayarların işlem kapasitelerinde büyük artışlar yaratacağını, üretim maliyetlerinin ise aynı kalacağını, hatta düşme eğilimi göstereceğini öngören deneysel (ampirik) gözlemdir.

Moore Yasası

Stanford Üniversitesi’nde Bilgisayar Bilimleri bölüm başkanlığı yapan Nvidia baş bilimcisi ve başkan yardımcısı Bill Dally, 2010 yılında Moore Yasası’nın artık geçersiz olduğunu söylemiş ve işlemci hız artışlarının Moore Yasası’nı karşıladığını ama kanunun diğer parçası olan güç tüketimine yönelik ölçeklendirmenin sona erdiğini söylemiştir.



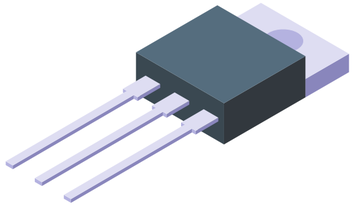
Moore Yasası

Dally'e göre Moore Kanunu'nda yer alan CPU ölçeklendirmesine ilişkin öngörü artık (2010 itibariyle) geçerli değil. Dally şöyle diyor;

Çok çekirdekli işlemci kullanmak, trene kanat takarak uçak yapmaya çalışmak gibi. Geleceğe yönelik ekonomik büyüme ve ticari yenilikler için paralel işlem teknolojilerinin yani GPU'nun (grafik işlem birimlerinin) ilerlemesi gerekiyor. İleri gidebilmek için kritik nokta, enerji verimli sistemler inşa etmektir.

Moore Yasası

Minimum bileşen maliyetleri için karmaşıklık her yıl kabaca 2 prim oranı artmaktadır Kısa vadede artış göstermesedde bu oranın bu şekilde devam etmesi beklenebilir. 1975'e kadar bileşenlerin sayısı minimum maliyetli her bir entegre devre için 65 bin olacaktır. Böyle büyük bir devrin tekbir plaka üzerine sığdırabileceğinie ben inanıyorum.
Moore



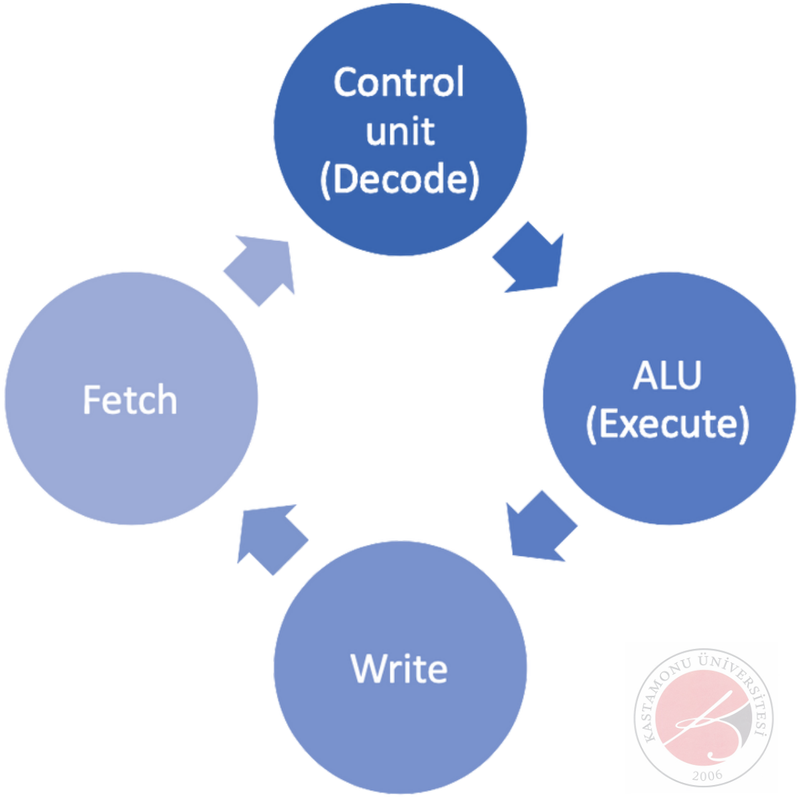
Moore Yasası

- **ALU:** Aritmetiksel ve mantıksal işlemlerin yapıldığı yerdir.
- **Ön Bellek:** İşlemcinin belleğidir.
- **Program sayacı:** İşlemci için sıradaki komutun adresini hafızasında tutar. Önce L1 / L2 / L3 ön belleğe bakar istenilen adresi bulamazsa geçici belleğe sonra da sabit diske bakar.
- **Kod çözücü:** Geçici bellekten gelen kodlar adres bilgisiyle birlikte geldiğinde ALU'nun bu kodu işleye bilmesi için kodun adres kısmından ayrılmasını sağlayan birimdir.
- **Kaydediciler (Registers):** ALU'da işlenmiş bir kodun saklandığı yerdir.
- **Bayraklar (Flags):** ALU'nun yaptığı işlemin sonucuna göre 1 veya 0 sonucunu alır.

Fetch - Decode - Execute

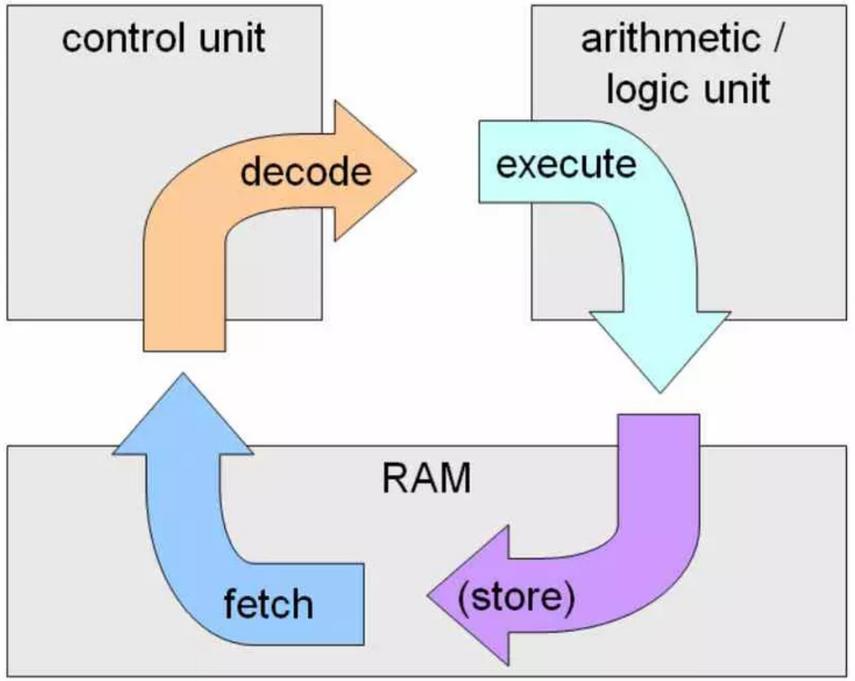
İşlemcilerin en basit sürümlerinde üç farklı aşamada çalışmasıdır, bunlar aşağıdaki gibidir:

- Getirme veya Yakalama - Fetch: Komutun RAM'den yakalandığı ve işlemcinin içine kopyalandığı işlemdir.
- Kod Çözme - Decode: Daha önce yakalanan talimatın kodunun çözüldüğü ve yürütme birimlerine gönderildiği işlemdir.
- Yürüt - Execute: Talimatın çözüldüğü ve sonucun işlemcinin dahili kayıtlarına veya RAM'in bellek adresine yazıldığı yerdir.



Fetch - Decode - Execute

İşlemciler genellikle bilgisayarın beyni olarak adlandırılır. Bunun sebebi de, aslında bilgisayardaki tüm komutları(instructions) yönetmesidir. Bunu yaparken aslında temel bir döngü kullanır:
Getir(Fetch) -> Çözümle(Decode) -> Çalıştır(Execute)



Fetch (Kodu Alma)

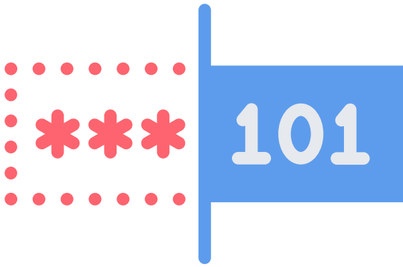
- Bir komutu hafızadan getirmek, onu işlemekten daha uzun sürdüğü için işlemcilerde register denilen bir alan vardır. Bu register da işletilecek olan komutun bazı önemli değişkenleri ve sonuçları gecici olarak tutulur. İşlemcideki register haricinde bilgisayarlar, özel bazı registerlara da sahiptir.
- Bunlardan birincisi program counter adında, işlemci tarafından bir sonraki getirilecek komutun, hafızada bulunduğu adresini tutan bir register dır. Ne zamanki ilgili komut, işlemci tarafından getirilirse(fetch), program counter, bir sonraki getirilecek komut tarafından güncellenir.

Fetch (Kodu Alma)

- Bu evre, program belleğinden komutu almayı içerir. Program belleğindeki yer, programın o andaki yerini bir sayıyla tutan program sayıcı tarafından belirlenir.
- Başka bir deyişle, program sayıcı, MİB'nin o andaki programın hangi kısmında olduğunun yerini tutmaktadır. Bir komut alındıktan sonra program sayıcı, alınan komutun boyunun bellek birim cinsinden değeri kadar artırılır. Bazen getirilmesi gereken komut hızca daha yavaş bir bellekten alınır, böylece MİB'nin komutun geri dönmesini beklerken zaman kazanması sağlanır.

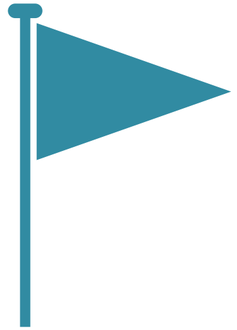
Decode (Kod Çözme)

- MİB'nin bellekten getirdiği komut, MİB'nin ne yapacağını belirlemede kullanılır. İşte bu kod çözme evresinde, komut MİB'deki önem oranına göre parçalara ayrılır. Sayısal kodun değerinin yorumlanması, MİB'nin komut set mimarisi (Instruction Set Architecture) ile tanımlanır.
- Genelde, komuttaki sayıların bir grubu, işlem kodu, hangi işlevin gerçekleştirmesi gerektiğini gösterir. Geri kalan kısımdaki sayılar komut için gerekli bilgileri sağlarlar (örneğin bir toplam işlemi için gereken işlenen değerler)



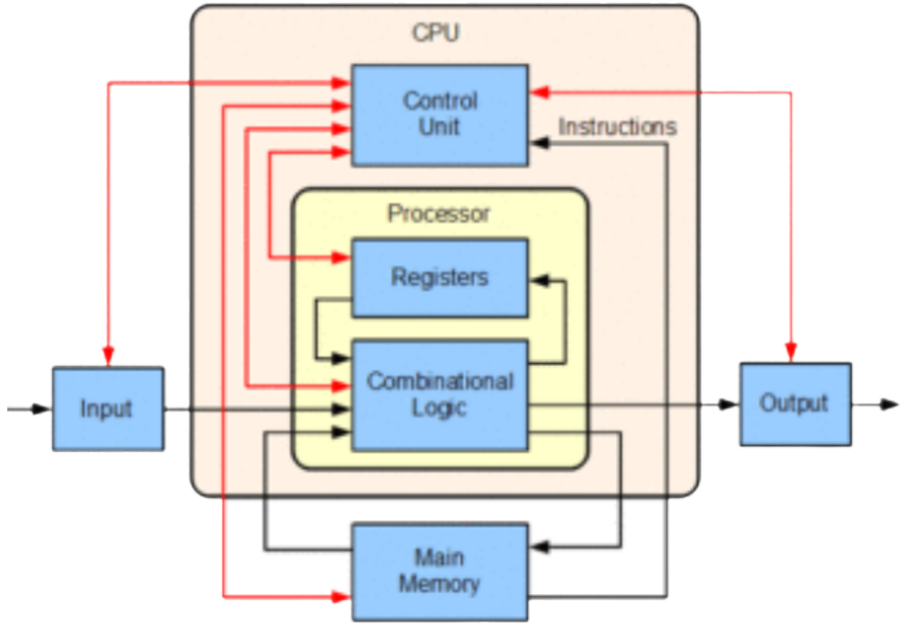
Execute (Yürütme)

- Bu evrede, istenen işin gerçekleşebilmesi için MİB'nin birçok kısmı bağlı haldedir. Örneğin, bir toplama işlemi istendiğinde, aritmetik ve mantık birimi (Arithmetic Logic Unit) bir kısım giriş ve çıkışlara bağlı olacaktır. Girişler toplamada kullanılacak sayıları içerirken, çıkışlar ise sonuç değerini tutacaktır. ALU, girişlerde basit aritmetik ve mantık işlemlerini gerçekleştirecek devre yapılarına sahiptir. Eğer toplama işlemi MİB'nin gerçekleştirebileceğinden çok büyük sonuçlar üretiyorsa, bayrak yazmaçlarındaki aritmetik taşma bayrağı kullanılacaktır.



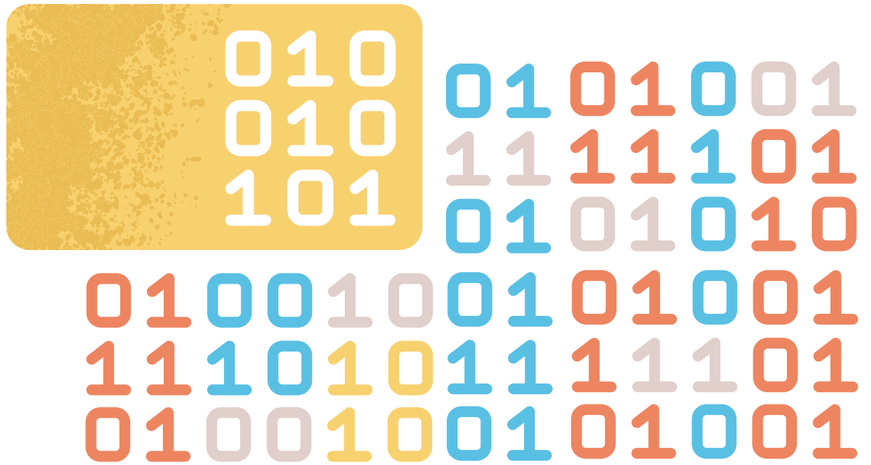
İşlemcinin İşleyişi

- Bilgisayarda herhangi bir program çalışırken, programla ilgili kodlar geçici bellek (ram) üzerine alınır. Kodlar sırasıyla program sayacı tarafından komut kaydedicisine gönderilir. Komut kaydedicisi komutu adresinden ayrılması için komut çözücüye gönderir. Komut çözücüde kod adres kısmından ayrılarak ALU'ya gönderilir. ALU içersinde kod işlendikten sonra kaydedicilere ve geçici belleğe gönderilir.



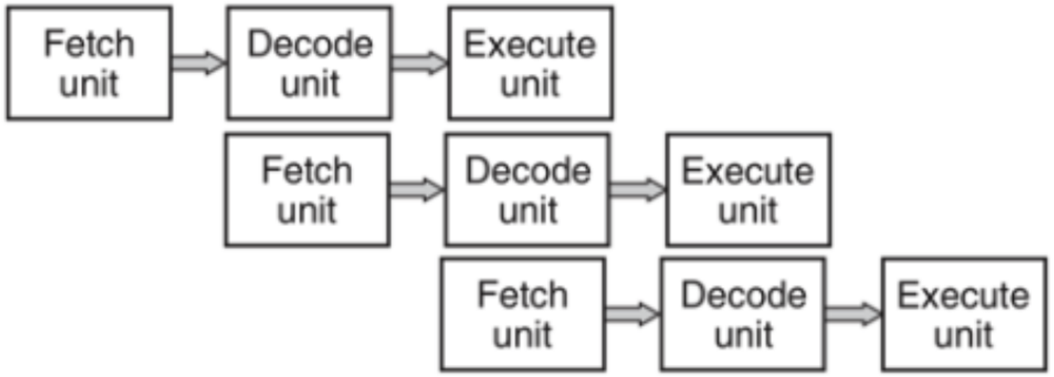
İşlemcinin İşleyişi

- Komut seti, bir işlemcinin yürütebileceği bütün işlemlerin ve bu işlemlere ait varyasyonların listesi, talimatnamesidir. Komut seti veri türlerini, komutları, kayıtları, adresleme türlerini, hafıza mimarisini, kesme, hata yakalama ve harici giriş-çıkışı (I/O) ve makine dilini içerir.
- Makine Dili: İşlemcinin direkt olarak yürüttüğü komutların formuna denir. Bir nevi kök komutlardır. “0 ve 1”lerden oluşurlar çünkü çoğunlukla makine dilinde “ikilik sistem (binary)” kullanılır. Programlama dilindeki komutlar “çeviricilerce” işlemci tarafından yorumlanabilecek bu sayılara dönüştürülürler.



İşlemcinin İşleyişi

- **Pipeline:** işlemci üzerindeki yapıların boşta kalma olmaksızın çalışmasıdır.
- **Hyper Threading:** Intel firmasının işlemcilerinde aslında tek bir işlemci varken işlemcinin boşta kalan kaynakları kullanarak sistemde iki farklı işlemci varmış gibi davranmasını sağlayan sistemdir.



Pipeline Yöntemi

Kaynaklar

Ebubekir Yaşar-Bilgisayar Donanımı

Tuncay Uzun-Mikroişlemci Sistemleri, İlham Tarımer-Mikroişlemciler

Bilgisayar Donanımı, Sinop Üniversitesi, Öğr. Gör. Erhan Sur

Ali Döngel-Bilgisayar Donanımı ve Elektronik