TEMEL BİLGİSAYAR KAVRAMLARI

Mühendis ve Teknisyen

Mühendis (engineer), ihtiyaç duyulan ürünü; ekonomik (cost), güvenli (safe) ve kullanışlı (functional) olarak zamanında (on time) ortaya koyan kişilerdir. Bunu yapmak için; keşif (invent), tasarım (design), analiz (analysis), uygulama (build) ve sınama (test) yaparlar[1].

Teknisyenler (technician) ise genellikle laboratuvarda teknik ekipmanlarla ilgilenen veya bu ekipmanlar ile pratik çalışmalar yapan kişilerdir. Genel olarak amacı ekipmanın bakım ve kullanımına ilişkindir.

Konumuz yazılım olduğundan, yazılım mühendisleri ise istenilen bilgisayar yazılımını ekonomik, güvenli ve kullanışlı olarak zamanında ortaya koyan kişilerdir. Programcılar ise teknisyenlere karşılık gelir. Yani geliştirme aşamasında bazı yazılım kısımlarının kodlanmasını veya geliştirilen yazılımın çalışır tutulmasını sağlarlar.

Bilgisayara Niçin İhtiyaç Duyulur?

Bir çiftçi niçin traktöre ihtiyaç duyar? Bir berber niçin elektrikli tıraş makinesi kullanır? Benzer şekilde bir mühendis niçin bilgisayara ihtiyaç duyar?

İşte bütün bu soruların cevabı yapılacak işleri daha kısa sürede yapmaktır. Yani bütün araç gereç ve makinalara olan ihtiyacımız işlerimizi daha hızlı (speed) yapabilmek içindir. Bütün makineler gibi bilgisayar da veriden hızlıca bilgi elde etmemizi ve tekrarlanan hesaplamaları hatasız yapmamızı hızlıca sağlayan makinelerdir.

Veri sorumlusu, veri hazırlama kontrol işletmeni gibi unvanlar bu sebeple ortaya çıkmıştır. Bu unvanlara ihtiyaç duyulmasının sebebi verinin belli bir şekle uygun olarak bilgisayara girilmesi ve işlenen verinin saklanması ve raporlanmasının bilgisayar dünyasında önemli bir yer tutmasıdır.

Bilgisayarlar ilk zamanlarda daha çok cebirsel ifadeleri hatasız ve hızlı bir şekilde yapmak için kullanılmışlardır. Bunu daha sonradan ALGOL (ALGOritmic Language) adını alacak IAL (International Algebraic Language) ya da FORTRAN (FORmula TRANslation) dilinden de anlayabiliriz.

Veri ve Bilgi

Veriler (data), genellikle bir ölçüm ya da sayım sonucu elde edilen ve hakkında az şey bilinen varlıklardır. Örneğin ölçülen hava sıcaklığı, derse giren öğrenci sayısı, tartılan sebzenin ağırlığı gibi şeyler.

Bilgi (information) ise verinin işlenerek otaya çıkan anlamlı verilerdir. Örneğin ortalama hava sıcaklığı, ölçülen yerde bir ölçüm yapmaya gerek kalmadan sıcaklığı yaklaşık olarak bilmeye yarayan bir bilgidir. Bir sınıftaki not ortalaması, o sınıftan rastgele seçilen bir öğrencinin notunu yaklaşık belirleyen bir bilgidir.

Kısacası bilgi, işlenmiş veridir.

En Basit Bilgisayar

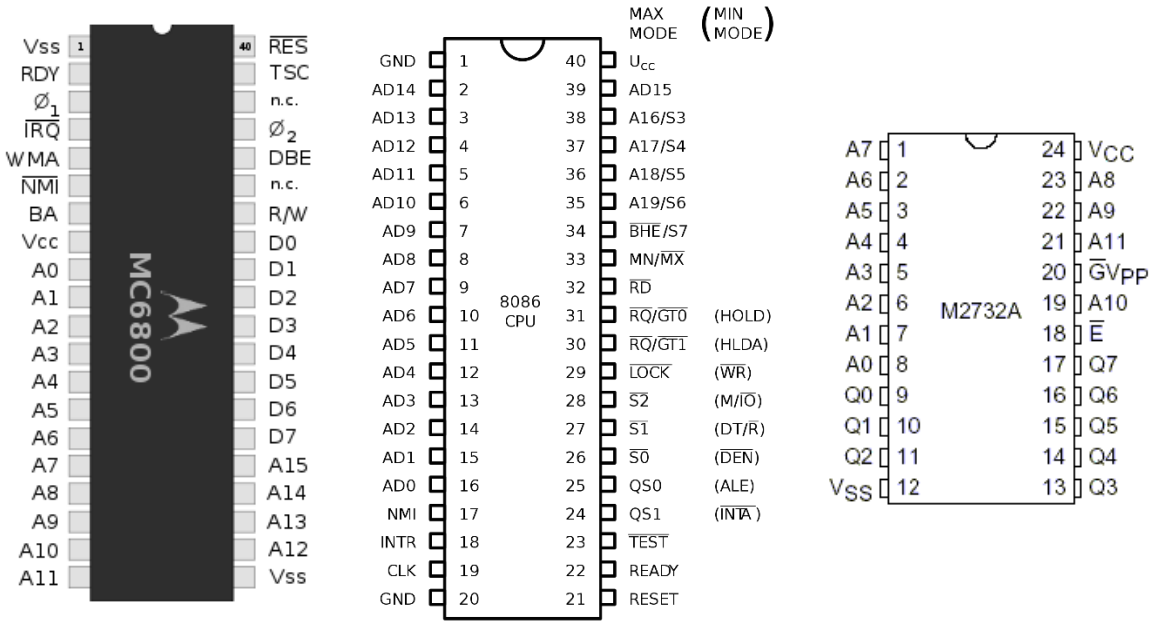
En basit bilgisayar;

- Merkezi işlem birimi (Central Processing Unit-CPU) kısaca işlemci,
- Bellek (memory) ve
- Bunları birbirine bağlayan elektrik yollarından (bus) oluşur.

[1] https://www.bls.gov/

Bellek ise işlemcinin işleyeceği emirleri barındırır. Belleğin hem adres (address) hem de veri (data bus) bacakları bulunur. Benzer şekilde işlemcinin de bellek ile iletişimini sağlayan adres ve veri bacakları bulunur. Belleğin bu bacakları elektrik yolları ile işlemcinin adres ve veri bacaklarına bağlıdır. Bunların yanında hem işlemcinin hem de belleğin okuma (read), yazma (write), kesme (interrupt) veya kristal bağlanan bacakları gibi kontrol (control) bacakları da bulunur.

–



Şekil 1. Motorola 6802, Intel 8086 İşlemciler ile 2732 EPROM Belleği

Şekil 1'de görüldüğü üzere 6800 işlemcisinin 15 adet adres bacağı, 8 adet veri bacağı bulunmaktadır. 8086 işlemcisinin ise 20 adet adres bacağı bulunakta olup bu bacakların 16 adedi aynı zamanda veri bacağı olarak kullanılmaktadır. 2732 belleğin ise 12 adet adres bacağı, Q ile gösterilen 8 adet veri bacağı bulunaktadır.

İşlemci ile bellek arasında bağlantı sağlayan elektrik yolları (bus) adres veya veri taşır. Bu yolların veri taşıyanlarına veri yolu (data bus), adres taşıyanına ise adres yolu (address bus) adı verilir. Veri yolu; hem işlemci tarafından belleğe veri yazmak veya bellekten işlemciye taşınacak veriler için kullanıldığından çift yönlüdür. Adres yolu ise belleğin hangi bölgesindeki veriye ulaşılacağını belirleyen hatlar olup tek yönlüdür.

Bilgisayarlar elektrikle çalışan aletlerdir. İşlemci ile bellek arasında bir hatta elektrik varsa "1" yoksa "0" olarak anlamlandırılır. Bunlar ikili sayı sisteminin rakamlarıdır (BInary DigiT) ve kısaca bit olarak adlandırılır. Dolayısıyla;

– 1 elektrik hattının taşıyacağı veri 0 ya da 1 olabilir.
– 2 elektrik hattının taşıyacağı veri 00, 01, 10 ve 11 olabilir. Yani 2^2 kadar farklı veri taşıyabilir.
– N adet elektrik hattının taşıyacağı veri 2^N farklı alternatifte bilgi olacaktır.

İşlemcinin veri yolunun (data bus) genişliği kelime uzunluğudur ve işlemcinin aynı anda işlem yaptığı bit sayısını da verir. 8 Bitlik veriye bayt (byte) adı verilir ve bellek miktarı genellikle bayt olarak ölçülür. Veri yolu, 8 bit ise 8 bitlik-BYTE işlemci, 16 bit ise 16 bitlik-WORD işlemci, 32 bit ise 32 bitlik-DWORD (Double WORD) işlemci, 64 bit ise 64 bitlik-QWORD (Quad WORD) işlemci olarak adlandırılır. Bu nedenle veri yolu genişliği ile işlemcinin akümülatör ve kaydedicilerin bit sayısı da aynı olur.

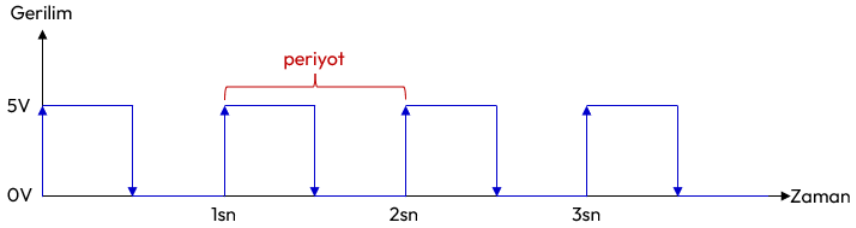
Adreslenebilir bellek miktarı ise işlemcinin adres hatlarının sayısı yani adres yolu (address bus) ile belirlenir. Adres yolu ne kadar geniş ise adreslenebilir bellek miktarı da o kadar artar.

İşlemci	Veri Yolu	Adres Yolu	Adreslenebilir Bellek: bayt	kilobayt	megabayt	gigabayt
Motorola 6802	8 Bit	16 Bit	2^{16}=65.536	64		
Intel 8086	16 Bit	20 Bit	2^{20}=1.048.576	1024	1	

Intel 80286	16 Bit	24 Bit	2^{24}=16.777.216		16	
Intel Pentium	32 Bit	32 Bit	2^{32}=4.294.967.296		4096	4
Intel i7	64 Bit	36 Bit	2^{36}=68.719.476.736			64

Tablo 1. Bazı İşlemcilerin Adres ve Veri Yolu Genişlikleri

Bilgisayarlarda kablodaki gerilimin değeri genelde 5V, 3.3V, 2.8V, 2V, 1.5V veya 1V gibi değerler alır. Kullanıcılar için gerilimin ne olduğu değil varlığı önemlidir. Gerilim değerin yüksek olması fazla akım çekme ve ısınma anlamına gelir. Bu nedenle zamanla daha düşük gerilimle çalışan işlemciler tasarlanmıştır.



Şekil 2. Frekansı 1 olan Kare Dalga İşareti

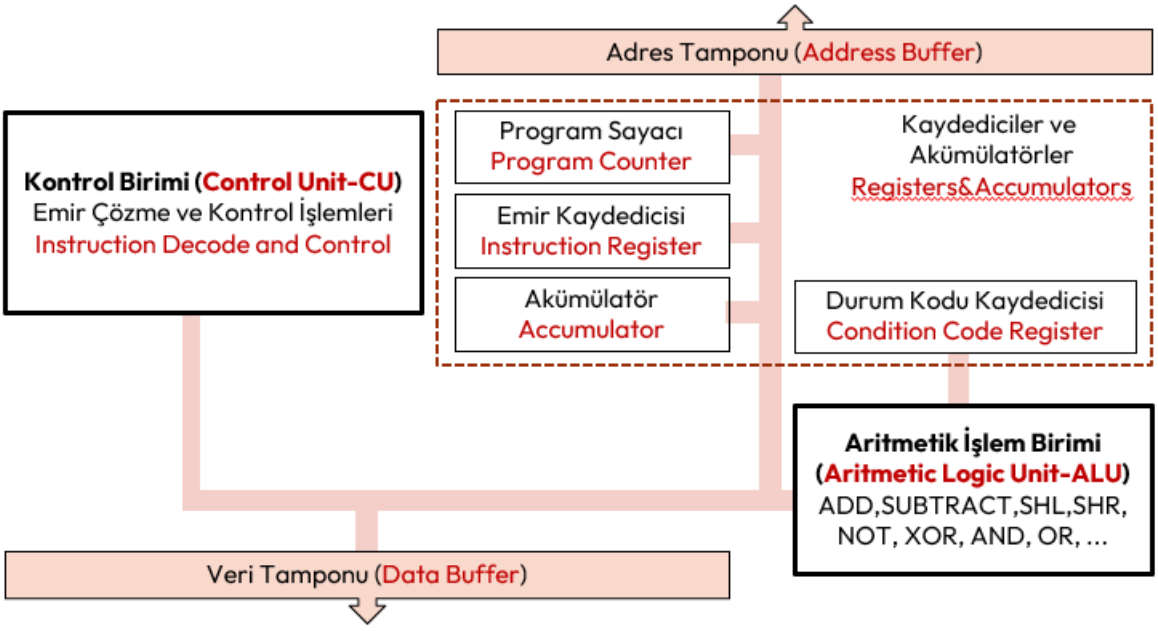
İşlemciler kare dalga olarak oluşturulmuş elektriksel bir işareti saat olarak kullanır. Bu işaret işlemciye Ø ya da CLK bacaklarından uygulanır. Saniyede 1 kez bir dalga üreten işaretin frekansı 1/1sn = 1 Hertz= 1 Hz. olarak hesaplanır. Bir mikro saniyede bir kez dalga üreten işaretin frekansı; 1/1µs = $1/10^{-6}$s = 10^{6} Hertz = 1.000.000 Hertz = 1 MHz olarak hesaplanır.

İşlemciye uygulanan saatin frekansı ne kadar yüksek olursa işlemci o kadar hızlanır. Ancak işlemcilerin de tasarımından ötürü karşılayabileceği belli bir frekans değeri bulunmaktadır. Günümüzde 5 GHz frekansını karşılayan işlemciler bulunmaktadır.

İşlemcinin Emri Alma, Çözme ve İcra Döngüsü

Basit bir işlemcide;

- Kaydediciler (register), işlemci içerisinde adres ve veri saklayan mini belleklerdir.
- Akümülatörler ise bellekten alınan verileri saklamak için en sık kullanılan kaydedicilerdir.
- Program sayacı (program counter), işlemcinin icra edeceği emri (instruction) takip için kullanılır. Getirilecek bir sonraki emrin bellek adresini içerir.
- Emir kaydedicisi (instruction register) işlemcinin o an icra ettiği emirdir.
- Koşul kodu kaydedicisi (condition code register), herhangi bir işlemin durumunu gösteren farklı bayraklar (flag) içerir.



Şekil 3. Basit bir İşlemcinin Yapısı

Bu işlemci tasarımı aynı zamanda depolanmış program (stored program) kavramı olarak adlandırılır ve *Von Neumann Mimarisi* olarak da bilinir[2]. *John Von Neumann* tarafından 1946-1949 yılları arasında tasarlanan bu mimari günümüzde modern bilgisayarlarda hala kullanılmaktadır.

Basit bir bilgisayara elektrik verildiğinde aşağıdaki üç adımlık çevrim elektrik kesilene kadar sürekli tekrarlanır;

1. Emri Alma (fetch): Program sayacının (program counter) tuttuğu adresteki emir (instruction) bellekten okunur.
2. Çözme (decode): İşlemci içerisindeki kontrol birimi (control unit) tarafından emrin ne anlama geldiği ve buna bağlı nelerin yapılacağını (örneğin ulaşılacak bellek adresi değiştirilir) belirler ve program sayacı bir sonraki emir için bir artırılır.
3. İcra (execute): Çözülen emir koduna bağlı olarak gerekirse yine işlemci içinde bulunan aritmetik işlem birimini (Aritmetic Logic Unit-ALU) çalıştırılır ve emrin işlenmesi sonucunda ortaya çıkan değerler ise işlemci içinde bulunan akümülatörlere (accumulator) veya kaydedicilerde (register) saklanır.

Burada çözülecek emirlerin (instruction) her biri bir tamsayıya karşılık gelir. Bu saylar emir kodu (instruction code) veya makine kodu (opcode) olarak da bilinirler. İşlemciler tüm sayıları emir kodu olarak anlamaz. Yani sınırlı sayıda sayı emri tanırlar. Yukarıda verilen Motorola 6802 işlemcisi 72 farklı emri, Intel 8086 işlemcisi ise 116 farklı emri tanır. Çözülebilecek tüm emirlerin oluşturduğu kümeye emir seti (instruction set) adı verilir.

İşlemci Tasarlama Stratejileri

İşlemciler, aşağıda verilen üç strateji ile tasarlanırlar;

1. İndirgenmiş Emir Setli İşlemciler (Reduced Instruction Set Computing-RISC): Az sayıda emir alan yüksek hızlı işlemci tasarımı. RISC işlemciler az sayıda emre sahip olduğundan emrin çözülmesi ve icrası da kısa zaman almaktadır. Bu da çalıştırılacak kodu büyütür.
2. Karmaşık Emir Setli İşlemciler (Complex Instruction Set Computing-CISC): Çok fazla emri anlayan karmaşık nispeten yavaş işlemci tasarımı. CISC işlemciler, çok sayıda emir tanıyabildiğinden emrin çözülmesi ve icra edilmesi nispeten daha uzun sürer. Ancak yüksek düzey dillerin daha kolay makine koduna çevrilmesini kolaylaştırır.
3. Özel İşlemciler: Bunlar grafik kartları ve yapay zekâ gibi belli amaçlar için tasarlanmış işlemcilerdir.

Genel olarak işlemcilerde *"fetch-decode-execute"* çevrimi 1 saat periyodunda tamamlanmaz. RISC işlemciler için bu çevrim, birçok emir için 1 saat periyodunda tamamlanır. CISC işlemciler için ise birden fazla saat periyodu gerekir. Çünkü CISC işlemcilerin emir seti daha geniş olduğundan emrin çözülmesi daha fazla vakit alır. Bu nedenle RISC işlemciler CISC işlemcilere göreceli olarak daha hızlıdır.

Günümüzde ticari olarak satılan işlemcilerin çoğu ortak emir seti kullanmaya başlamışlardır;

– Masaüstü bilgisayarların birçoğu Intel ve AMD Marka işlemciler kullanır ve bunlar IA-16, IA-32, x86-16, x86-64, AMD64, ... emir setleri ve eklentilerini desteklemektedir.
– Bilinen birçok telefon işlemcisi Acorn Risc Machine-ARM tabanlı işlemciler kullanır. Bu işlemcilerde ARMv7, ARMv8, ... emir setleri ve eklentileri kullanılmaktadır.

Program ve Programlama

Belli bir işlemi gerçekleştirmek üzere, işlemciye verilen bir dizi emre (instruction), bilgisayar programı (computer program) denir[3]. Bu bir dizi emrin, işlemcinin çalıştıracağı şekilde hazırlaması işlemine programlama (programming) denir. Programlama emir kodları (instruction code/opcode) ile yapıldığından bu programın dili, makine dili (machine language) olarak adlandırılır.

[2] https://www.youtube.com/watch?v=ByIlwN8q2ss
[3] "ISO/IEC 2382:2015". ISO. 2020-09-03. [Software includes] all or part of the programs, procedures, rules, and associated documentation of an information processing system.

Her işlemci üreticisi, her bir emre farklı makine kodu verdiğinden programlama oldukça zorlu ve teknik bilgi gerektirmekteydi. Burada daha önce yazılmış bir programın ne yaptığını anlamak için her defasında bu listelere defalarca bakmak gerekiyordu.

Sembolik İsim (Mnemonic)	16 Tabanında Emir Kodu (Hexadecimal Opcode)	Tanım
ABA	1B	ADD A to B → A
INCA	4C	INCREMENT A
INCB	5C	INCREMENT B
SBA	10	SUBTRACT B from A→ A
LDA	96	LOAD M to A, M: Memory
LDB	D6	LOAD M to B, M: Memory

Tablo 2. 6802 Emir Seti Örneği ve Sembolik İsim Listesi

Programın okunurluğunu artırmak için her bir emre sembolik isimler (mnemonic) verilmeye başlandı. Bu sembolik isimlerle yazılan program montaj kodu (assembly code) olarak adlandırılır. Montaj kodları bir metin dosyasına yazılarak bir dosyaya saklanır. Bu dosyalar montaj dosyaları olarak adlandırılır. Montaj dosyalarını makine diline çeviren programlar ilk derleyicilerdir (compiler). Bu derleyiciler, sembolik isimlerle yazılmış program kodlarını makine koduna çeviren programlardır.

Aşağıda x86 emir seti kullanan Linux işletim sisteminde örnek montaj (assembly) kodu örneği verilmiştir; Program genel amaçlı CL kaydedicisine koyulan sayıya kadar 2 ve 3'e bölünebilen sayıların toplamını bulup yine genel amaçlı DX kaydedicisine koyar[4].

```
section  .text          ; programı oluşturan emir kodları buradan başlar
         global   _start
_start:
         mov   dx, 0    ; dx 0 olsun. Toplamı tutatcak.
         mov   cl, 99   ; 99 kez tekrarlanacak.
sum:
         mov   ax, cx   ; cx, ax e kopyalanır. Bölme işlemi için.
         mov   bl, 3    ; bl 3 olsun.
         div   bl       ; ax/bl.
         cmp   ah, 0    ; Kalan olup olmadığı kontrol ediliyor.
         jne   cont     ; Kalan yoksa, 'cont' etiketine git.
         mov   bl, 2    ; Sonra bl 2 olsun.
         div   bl       ; ax/bl
         cmp   ah, 0    ; Kalan olup olmadığı kontrol ediliyor.
         jne   cont     ; Kalan yoksa, 'cont' etiketine git.
         add   dx, c    ; Değilse, cx deki sayı 2 ve 3'>e bölünür. Toplama ekle.
cont:
         loop  sum      ; Bir sonraki sayı için 'sum' etiketine dön.
exit:
         mov   eax, 1   ; Artık programdan çıkıyoruz.
         mov   ebx, 0   ; Return 0.
         int   80h      ; linux çekirdeğini çağır. (soft interrupt 80h!).
```

Genel Amaçlı Bilgisayarlar ve İşletim Sistemleri

Genel amaçlı yani herkesin kendi amacına göre kullanacağı bilgisayarlar; Veri girişini sağlayan klavye, verileri görüntüleneceği monitör, işlenen verilerin elektrik kesildiğinde saklanacağı disk ve benzeri donanımlara sahip olmalıdır.

Genel amaçlı bilgisayarın kullanıcı isteklerine cevap verebilmesi için;

1. Çeşitli programlara sahiptirler. İşte bu programlar bütününe işletim sistemi (operating system) adı verilir. DOS, Windows, Unix, Linux, MacOs, ChromeOS, OS/2 bunların en çok bilinenleridir.

[4] https://github.com/armut/x86-Linux-Assembly-Examples

2. Elektrik verildiğinde klavye, monitör ve disk olup olmadığı kontrol eden Basic Input Output System-BIOS programı koşturulur. BIOS gerekli kontrollerden sonra bilgisayara işletim sistemini yükler. İşletim sistemi yüklendikten sonra, işletim sistemi kullanıcının vereceği komutları bekler.
3. Kullanıcı işletim sitemine vereceği konutlarla yeni bir program derleyebilir, derlenmiş bir programı çalıştırabilir.
4. Kullanıcının vereceği komutlar grafik olmayan işletim sistemlerinde (DOS, Unix, Linux gibi) komut yazılarak verilir. Komut yazılan bu ekrana konsol, terminal ya da komut satırı adı verilir. Sunucu işletim sistemlerinde hala tercih edilmektedir.
5. Grafik ara yüzlerine sahip işlettim sistemlerinde (Windows ve MacOs İşletim Sistemleri ile KDE, Mate, GNome, Cinnamon, LXQt, XFce ve Deepin gibi grafik ara yüzleri içeren Linux İşletim Sistemleri) fare tıklamasıyla çoğu komut verilebilmektedir.

İşletim sistemlerinin yaygınlaşması ile kullanıcının bilgisayarı donanım bilmeden yapabilmesi sağlanmıştır. Günümüzdeki işletim sistemlerinin temel programlama dili olarak C dili kullanılmaktadır.