



“ İkili Sayı Sistemi ”

Dr. Cahit Karakuş, February-2019

Veri Türleri

- 1. Aşağıdakilerden hangisi bir veri türü değildir?
- a) Sembolik Veriler
- b) Alfanümerik Veriler
- c) Sayısal Veriler
- d) Alfabetik Veriler
- Cevap: a
- Açıklama: Veri türleri üç temel türdendir: Sayısal, Alfabetik ve Alfanümerik. Sayısal Veriler yalnızca sayılardan oluşur.
- Alfabetik Veriler sadece harflerden ve bir boşluk karakterinden, alfanümerik veriler ise sembollerden oluşur.

Types of data

- Data Types are used to define the type of a variable. It defines what type of data we are going to store in a variable. The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters.
- Numeric - int, float, complex
- String - str
- Sequence - list, tuple, range
- Binary - bytes, bytearray, memoryview
- Mapping - dict
- Boolean - bool
- Set - set, frozenset
- None - NoneType

Information Representation

- Symbols: Metinler, Sayılar, Alfabe, Resim, Video (Symbols ikili sayı sistemine dönüşür (8bit))
- Types of data
 - Text
 - Numbers: Integers, Reals (floating point numbers)
- Text
 - ASCII code (American Standard Committee on Information Interchange)
 - A unique 8-bit binary code for each character:
 - A-Z, a-z, 1-9, ., - ! " £ \$ % ^ & * () _ +
 - Special unprintable characters such as the
 - ENTER key (CR for carriage return)
- Numbers
 - Binary number representation of integers
 - If we save one bit to signify positive (+) or negative (-), then an n-bit binary word can represent integers in the range: $-2^{n-1} - 1 .. +2^{n-1}$

ASCII

- Semboller: harfler, Rakamlar, +, -, /, *, ., :, ;,
- Harfler, sayılar, aritmetik işlemler dahil tüm karakterler 8 bitlik bir veri ile tanımlanır.
- $2^8=256$ adet farklı sembol var. İndis: 0, 1, 2, ... , 255
- (0000 0000)b (1111 1111)b
- Klavyedeki tuşlar

Representing Numbers

- Positive whole numbers
 - We already know one way to represent them: i.e., just use base 2 number system
- All integers, i.e., including negative integers
 - Set aside a bit for storing the sign
 - 1 for +, 0 for –
- Decimal numbers, e.g., 3.1415936, 100.34
 - Floating point representation:
 - $\text{sign} * \text{mantissa} * 2^{\text{exp}}$
 - 64 bits: one for sign, some for mantissa, some for exp.

- Number systems: Ondalıkli sayı sistemi (Decimal) aşağıdaki sayı sistemine dönüştürülür.
 - Binary (İki sayı sistemi 0/1)
 - Bits: 0 ya da 1
 - Bytes: 8bit
 - Hexadecimal: 4bit
 - Octal
- Ascii code

Different types of data

- Numbers
 - Whole number, fractional number, ...
- Text
 - ASCII code, unicode
- Audio
- Image and graphics
- video

How can they all be represented as binary strings?

Strings and Character Numbers

- Bilgisayarın içinde, dizeler tıpkı sayılar gibi 1'ler ve 0'lar dizisi olarak temsil edilir.
- Bir dizede, karakterlerin (sembollerin) herbiri bir sayı olacak şekilde ikili sayılar dizisi olarak saklanır.
- Bilgisayarların ilk günlerinde, her üretici karakterler için kendi sayı kodlamasını kullandı.
- ASCII sistemi (Bilgi Değişimi için Amerikan Standart Kodu) 127 bitlik kodlar kullanır.
- Yazılımlar, Unicode'u destekler (100.000'den fazla sembol)

Sayı Sistemi

- Ondalık sayı sistemi 0 ile 9 arasındaki sembollerden oluşur. İki basamaklı ondalık sayı sistemi oluşturmak için 10 ile çarpılır. Üç basamaklı sayı sistemi oluşturmak için 100 ile çarpılır. Basamaklar 10'nun katları biçiminde devam edilir.
- İkili sayı sistemi 0 ve 1 den oluşur. Transistörün 0 ya da 1 çıkışı ile ifade edilir. Elektriksel sinyal olarak bulunur. Basamaklar 2'nin katları biçiminde ifade edilir.
- Örnek: $(109)_{10} = 1 \cdot 10^2 + 0 \cdot 10^1 + 9 \cdot 10^0$
- Örnek: $(101)_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (5)_{10}$ (İndislenir: 2 1 0); 2 üzeri indisler katsayıları ile çarpılarak toplanır.
- Bilgisayarda herşey 1 veya 0 olarak tutulur. Bit: 071
- $(0100\ 0001)_b$ bu semboldür, ondalık sayı sisteminde 65 değerine, ya da A karakterine karşılık gelir.
- Binary dünyasında herşey 1 ya da 0 ile ifade edilebilir. Bilgisayar hafızasındaki karşılığına bit denir.
- Bitler yanyana gelerek daha fazla bit dizisi tanımlanır. Sözelimi 2 bitlik sistemde 00, 01, 10, 10 dört farklı durum ya da sembol oluşturulabilir. Sayısal olarak gösterilmek istenildiğinde 0, 1, 2, 3 olarak ifade edilebilir.

Sayısal Sayı Sistemi

- Digital Number systems
 - Binary, Bits (0/1)
 - Byte: 8 bit
 - Hexadecimal: 4 bit
 - Octal: 3 bit
 - Numbers conversion among different systems
- Ascii code (Klavyedeki her bir tuş 8 bit yani 1 byte ile temsil edilir). Klavyedeki A tuşuna basıldığında bellekte ne görülür? 8 bit : (41)h = (0100 0001)b

Temel Birimler

- Bilgisayar sistemlerinde temel bilgi birimi bit: (1/0)
- Bit: 0/1; bitler elektriksel sinyaller ile temsil edilir. Elektronik devre elemanı Transistor. Data Bus, Bellek gözü içeriği, Registers
- Byte: 8 bitlik veriyi temsil eder. Ya da 1byte'lık bellek gözünü işaret eder. Belleklerde 2'li tabandaki işlemleri temsil eder. Adres Bus; bellek ve bellek gözü seçer.
- Bit/sec: Transfer edilecek ya da işlenecek bir saniyedeki veri miktarını temsil eder. 10 tabandaki üssel işlemler ile gösterilir.
- Qubit: Quantum hesaplamalarda en küçük veriyi temsil eder.
- Elektron: Qubitler elektronlar ile temsil edilir..



Veri

Veri Bilimi

- Veri Bilimi, bilgiler hakkında sonuçlar çıkarmak amacıyla uygulamalı ya da hesplamalı matematik, istatistik, olasılık ve makine öğrenimi tekniklerini kullanarak ham verileri analiz etme bilimidir.

Veri – Enformasyon - Bilgi

- **Veri:** Anlam kazanmamış, ilişkilendirilmemiş, özümlenmemiş, işlenmemiş gerçekler ya da bilgi parçacıklarıdır. Herhangi bir içerikten yoksun formlardadırlar. Yorum taşımazlar ancak işlenmek için hazırdırlar.
- Bilgisayar, içinde işlemek üzere girdi olarak alır, belleklerde saklanır; işlenir, çıktı olarak üretilir. Üretilen çıktı belleğe kayıt edilir, ekranda görüntülenir ya da I/O birlerinden diğer sistemlere transfer edilir.
- **Enformasyon:** Veriye değer katılarak, verinin anlamlandırılmasıdır. Belli bir amaç için birbiriyle ilişkili verilerin biraraya getirilmesi, düzenlenmesi sonucu oluşur ve bir anlam taşır, haber niteliği vardır. Kurumsal olarak bakıldığında enformasyon, anlamı olan veritabanıdır. Veri, içerik işlemleriyle değer kazandırılarak enformasyona dönüştürülmektedir. Enformasyon veriden doğmaktadır ve enformasyon da bilgiye dönüşmektedir.
- **Bilgi:** Genel olarak, veri ve enformasyonun yorumlanmasıyla ortaya çıkar.
- **Büyük Veri:** Verilerin günümüzde hız, çeşitlilik, kapasite (hacim) açısından büyük artış göstermesi ve bu artışa teknolojinin de destek vererek, yeni çözümler üretmesi ile birlikte “Büyük Veri” kavramı ortaya çıkmıştır.

Sembol - Veri – Bilgi – Yetenek - Bilgelik

Data – Information – Knowledge - Wisdom

- **Semboller (Sinyaller, Resimler, Şekiller, ...):** Giriş aşamasında bilgisayarın belleğine aktarılan rakamlar ve semboller, rakamlar, kelimeler, görüntüler, video ve ses gibi bilgi parçacıklarıdır.
- **Veri:** Anlam kazanmamış, ilişkilendirilmemiş, özümlememiş, işlenmemiş gerçekler ya da bilgi parçacıklarıdır. Herhangi bir içerikten yoksun formlardadırlar. Bazen fiziksel bir olaydır, yorumlanmamış gözlemlerdir. Yorum taşımazlar ancak işlenmek için hazırdırlar. Karar vermede etkili değildirler.
- **Information (Bilgi) :** Cevaplanması gereken ne, kim, ne zaman, nerede sorulardır. Bilgi, işlenmiş, düzenlenmiş, anlamlandırılmış verilerdir. Bilgi, organize, anlamlı ve yararlı verilerdir. Çıktı aşaması sırasında, oluşturulan bilgiler basılı rapor, grafik ve görseller ile sunum formuna sokulur. Bilgiler ileride kullanılmak üzere bilgisayar saklanır.
- **Knowledge (irfan sahibi, Yetenek - Tecrübe, Deneyim):** Karar vermede, kestirim yapmada, doğruyu aramada performansı yükseltmektir.
- **Understand (Bilinçlenmek):** anlayarak, kavrayarak, hissederek bilinçlenmektir.
- **Wisdom (Bilgelik):** Değerlendirilmiş anlayıştır. Sorgulayarak, kestirim yaparak karar vermek ve yorumlamaktır.

Veri Tipleri

Veriler farklı boyutlarda ve ayrıca türlerde gelir:

- Sinyaller: Ses, ısı, elektrik, elektromanyetik, koku, titreşim
- Metinler
- Sayılar
- Tıklama akışları
- Grafikler
- Tablolar
- Görüntüler
- İşlemler
- Videolar

Why Binary Arithmetic?

- Bilgisayarın belleğine tüm veriler 1 ya da 0 olarak kaydedilir.
- Tüm aritmetik, lojik, veri saklama, ver transferi binary matematiğinde yapılır.
- İşlemcide tüm işlemler 1 ve 0 lardan yapılır.
- Donanım sadece 1 ve 0 ikili sayı sisteminde elektriksel sinyaller ile yapılır.
Transistör.
- Must represent all numbers, integers or floating point, positive or negative, by binary digits, called *bits*.
- Can devise electronic circuits to perform arithmetic operations: add, subtract, multiply and divide, and mantıksal işlemler, karşılaştırmalar on binary numbers.

Why Binary Arithmetic?

$$3 + 5$$



$$= 8$$

$$0011 + 0101$$



$$= 1000$$

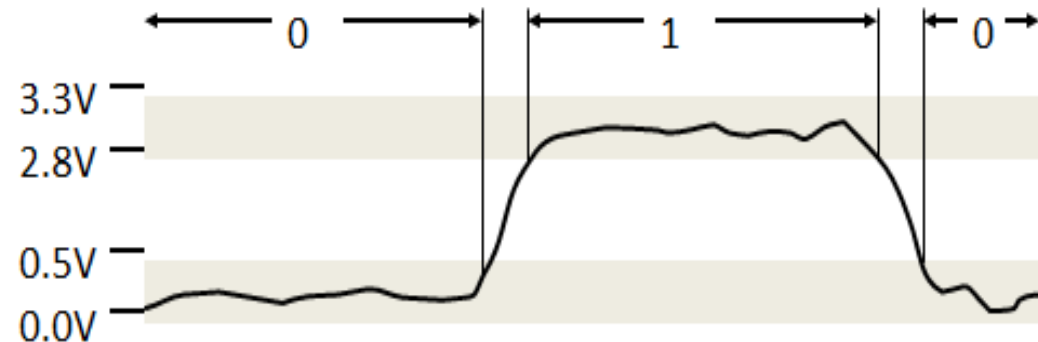
İkili Sinyaller- Bit (0/1)

İkili sinyal (Binary signal), ikili durum sinyali: İki durumlu veriler (0/1).

- off & on
- Elektriksel sinyaller ile taşınır, saklanır. low voltage & high voltage; 0v & 5v

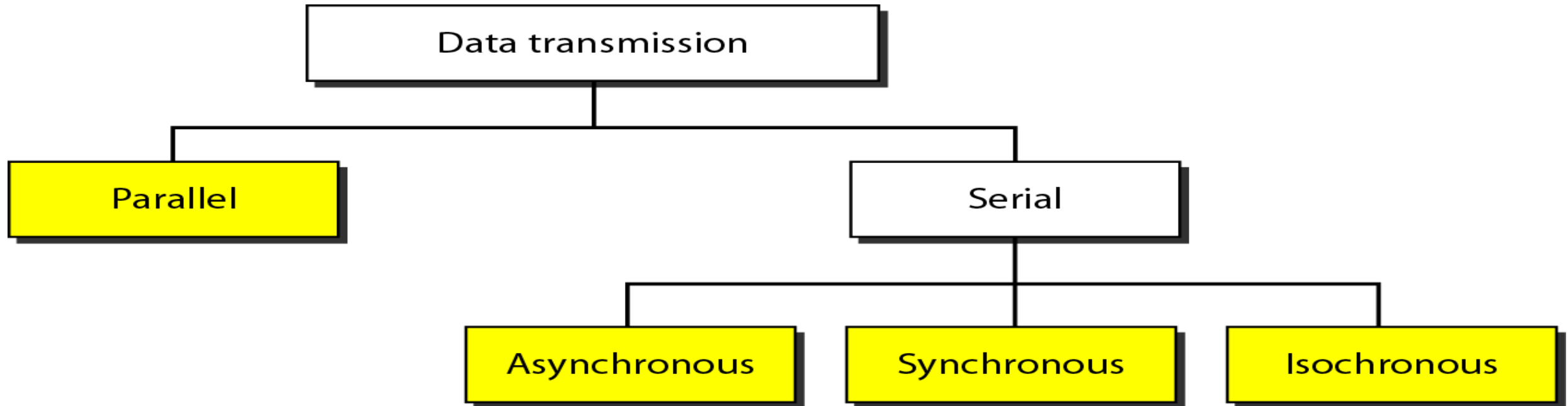
Bit: Sadece matematiksel bir kavram değil, aynı zamanda fiziksel dünyada karşılığı vardır.

- İkili sayı sistemi 0 veya 1 değerine sahip olur ve başka hiçbir şey olmaz.
- **Bir bit, bir bilgisayardaki en küçük bilgi birimidir**



Data Transmission

Data transmission and modes



Serial Transmission

Data is transmitted, on a single channel, one bit at a time one after another
- Much faster than parallel



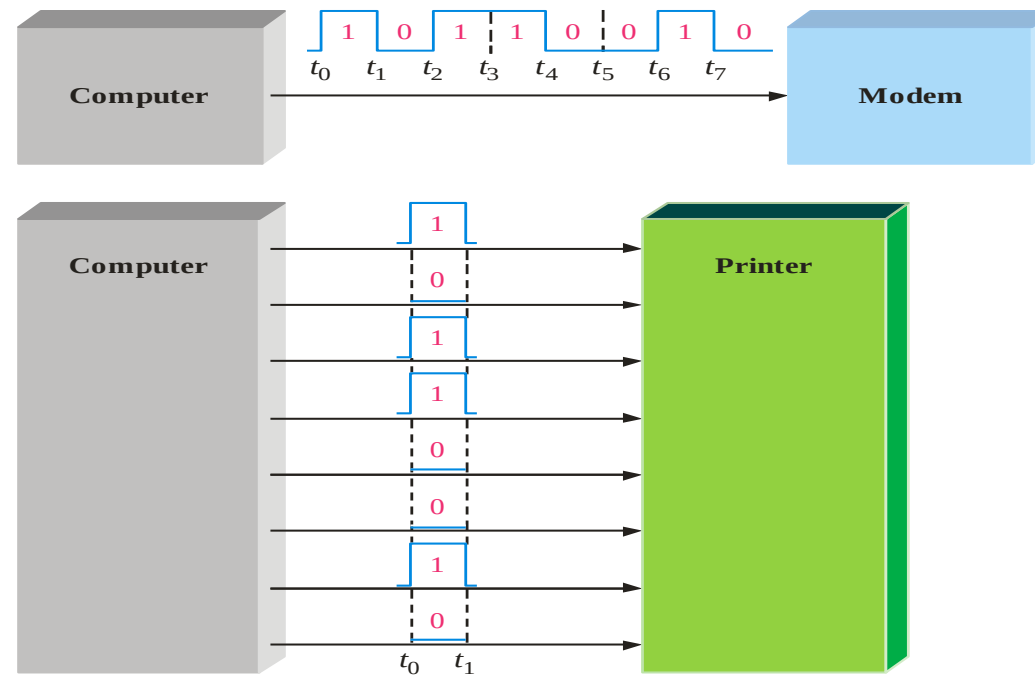
Parallel Transmission

- each bit has it's own piece of wire along which it travels
- often used to send data to a printer

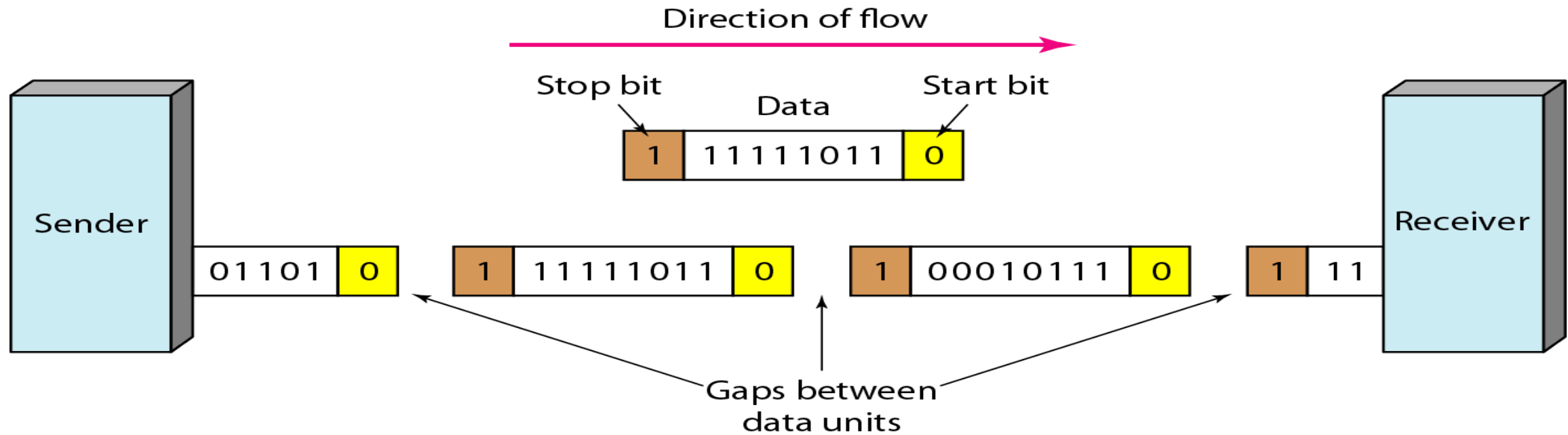


Serial and Parallel Data

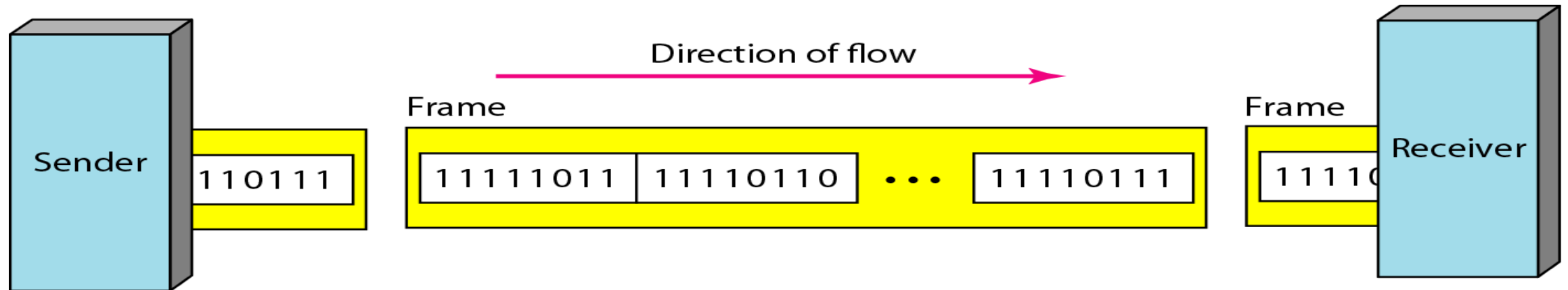
Data can be transmitted by either serial transfer or parallel transfer.



Asynchronous transmission



Synchronous transmission



Analogue and Digital Signals

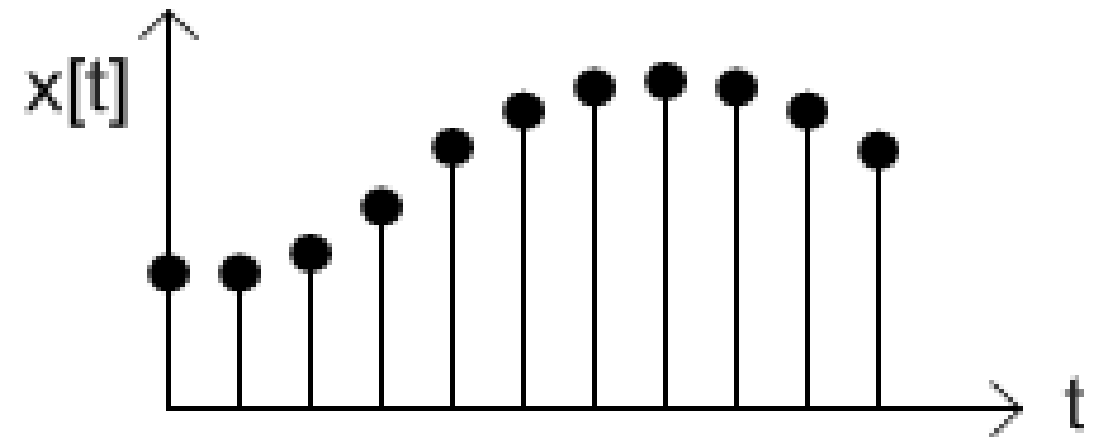
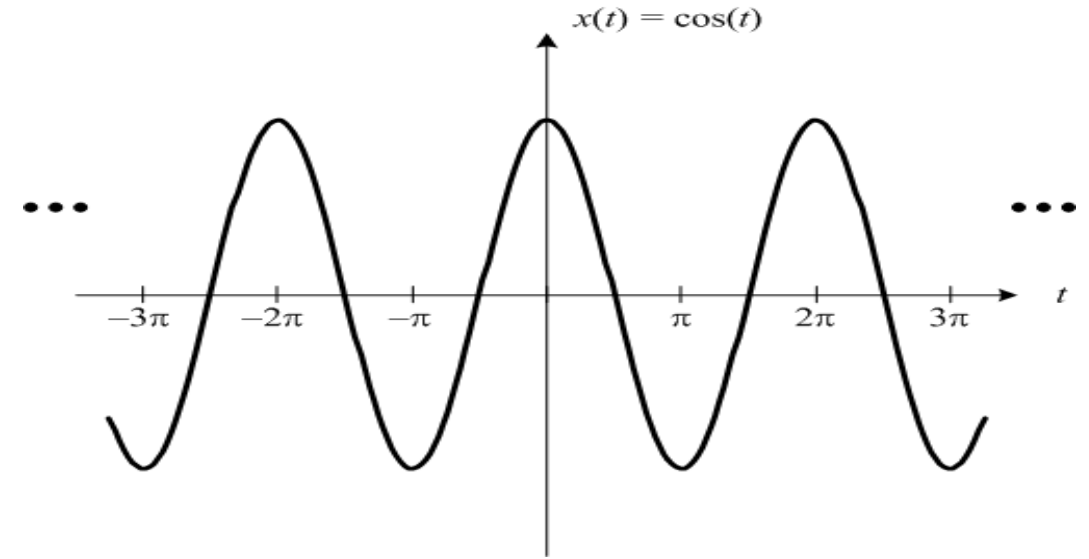
Bilgisayarın içinde veriler nasıl temsil edilir?

- Bilgisayarın içinde, dizeler tıpkı sayılar gibi 1'ler ve 0'lar dizisi olarak temsil edilir.
- Sinyaller uzaklara dalgalar şeklinde yayılırlar. Dalgalar sinyallerdeki mesajları taşırlar. Sinyaller: Akustik sinyaller, Sismik Sinyaller, Elektrik sinyaller, Elektromanyetik sinyaller, Isı, Titreşim, ...
- Mesajlar/Semboller: Ses, Sayılar (pozitif, negatif, tamsayı, kayan nokta, kesir), Karakterler, Metinler, Resim, Görüntü, Klavye tuşları; mesajlar, semboller
- Analog sinyaller: Genliği, frekansı, fazı zamanla değişen sinyallerdir. $X(t_i) = A(t_i) * \sin(wt_i + \phi(t_i))$;
- $w = 2\pi f$, f : frekans (Hz=1/sec), ϕ : faz (Derece ya da radyan), A : Genlik (birim, volt, amper)
- Bilgisayar sisteminde veriler aritmetiksel ve mantıksal işlemler yapılırken, belleklerde saklanırken, ilgili birimler arasında transfer edilirken bit(1/0) olarak yapıldığına göre Analog sinyaller sayısal sinyallere dönüştürülmesi gerekmektedir.

Sinyallerin sınıflandırılması

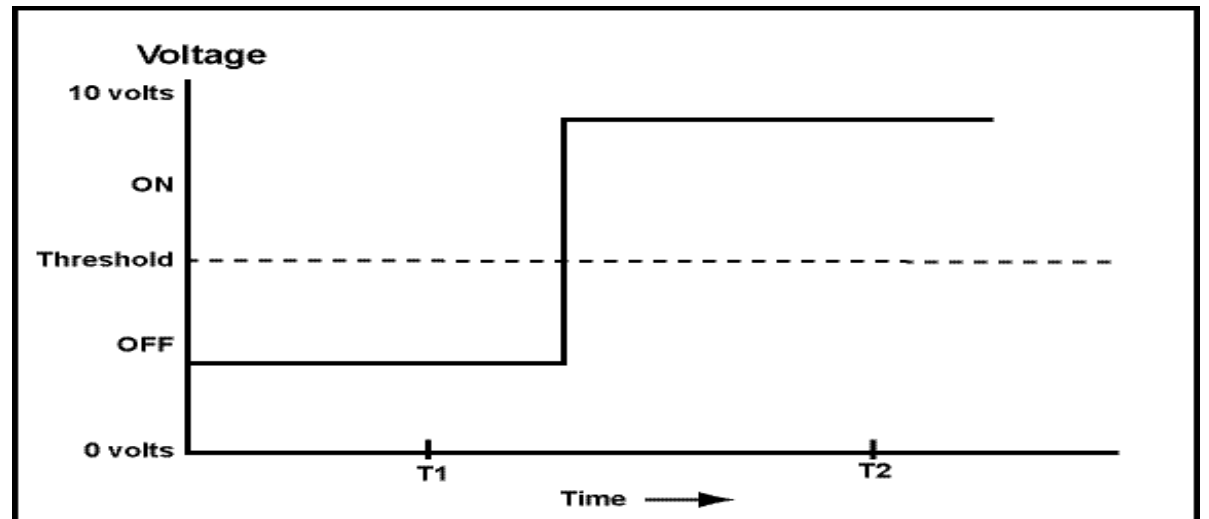
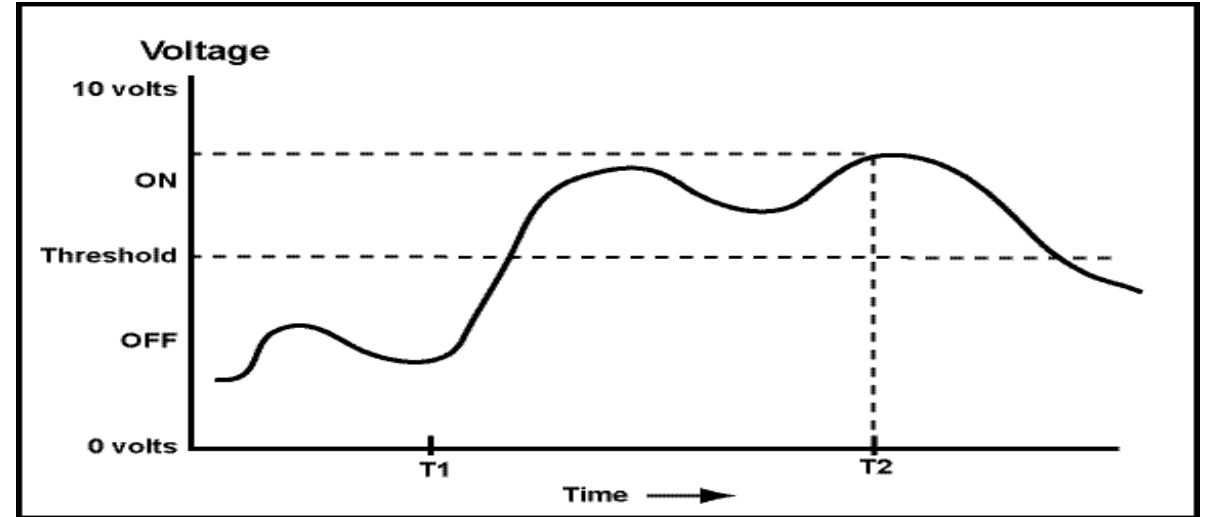
Signals are basically classified into two different types as follows.

- Continuous time signals, Analog sinyaller
- Discrete (Ayrık) time signals: Giriş değerinin belirli aralıklarla veya seviyelerde ölçülmesi ile elde edilen çıktı sinyaline verilen addır. Ayrık zamanlı sinyali, örnekleme işlemi ile giriş sinyalinden türetilir. Ayrık zamanlı sinyalinden alınan örnekler quantalama ile sayısal sinyale dönüştürülür.



Analog to Binary Signal

- A voltage below the *threshold*
 - Off (0)
- A voltage above the *threshold*
 - On (1)



Analogue and Digital Signals

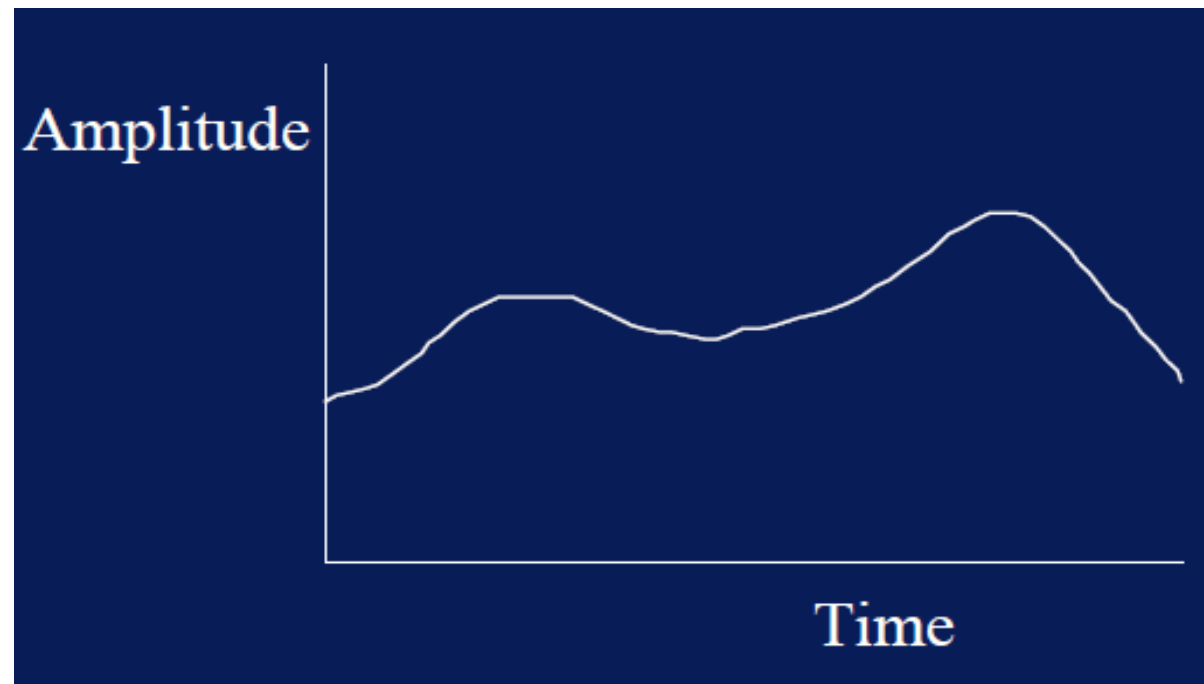
- Calculation with numbers is usually done in base 10 arithmetic
- Easier to effect machine computation in base 2 or binary notation
- We can also use base 2 or binary notation to represent logic values: TRUE and FALSE
- Manipulation of these (digital) logic values is subject to the laws of logic as set out in the formal rules of Boolean algebra

Analogue and Digital Signals

- An analogue signal can have any value within certain operating limits
- For example, in a (common emitter) amplifier, the output (O/P) can have any value between 0v and 10v.
- A digital signal can only have a fixed number of values within certain tolerances

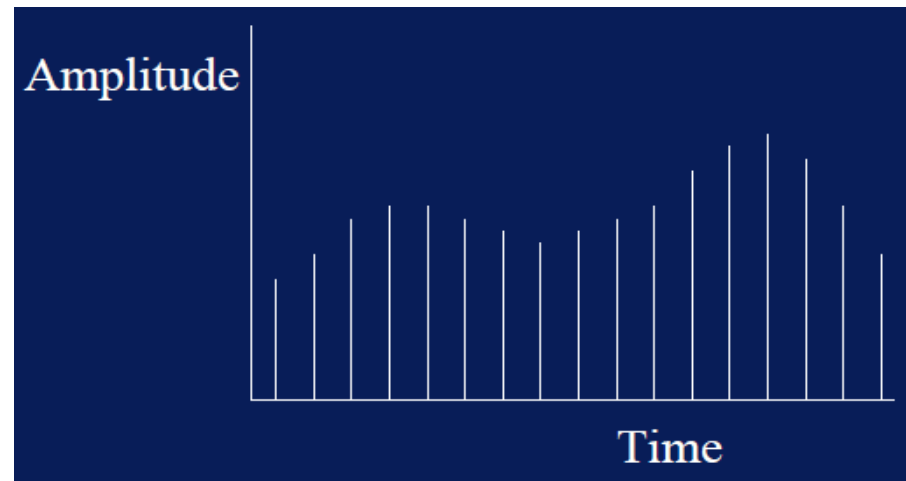
Analogue Signals

- The amplitude is defined at all moments in time



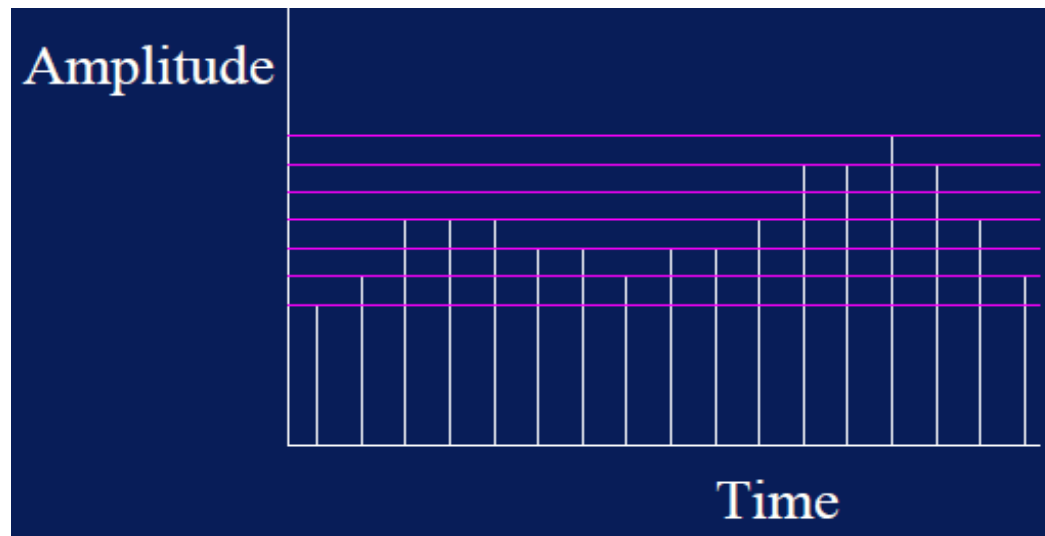
A digital signal

- It is a sampled version of the analogue signal
- Only defined at certain discrete times
- DISCRETE TIME SIGNAL
- A digital signal is a sampled version of the analogue signal
- Analog sinyal örnek alma aralığı= $1/f_{\text{maks}}$, [sec]. Burada f_{maks} =analog sinyalin maksimum frekansıdır.



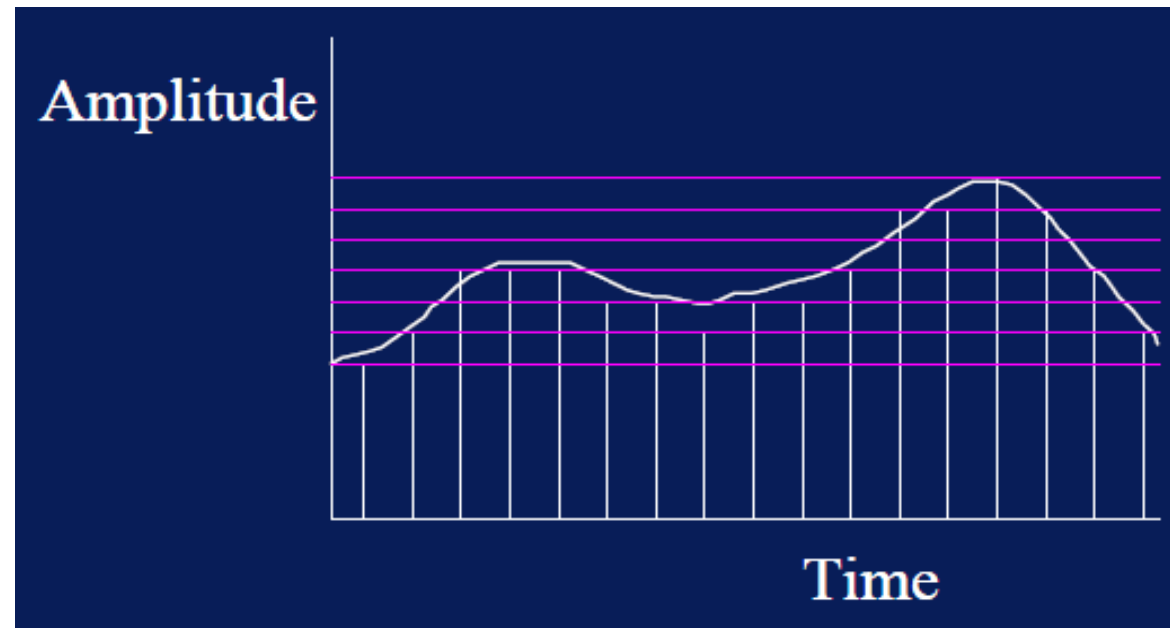
Analogue and Digital Signals

- The amplitude may also be restricted to take on discrete values only
- • In which case it is said to be quantized



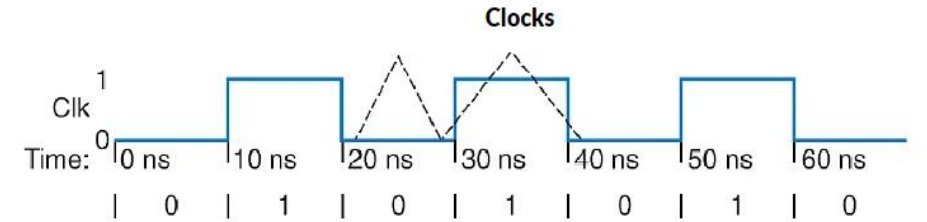
Analogue and Digital Signals

- Quantization introduces errors which depend on the step size or the resolution
- Signals (voltages or currents) which are samples and quantized are said to be DIGITAL
- They can be represented by a sequence of numbers



Dijital Sinyal Tanımları

- **Dijital sinyal:** Bir dizi sayısal ayırık değerlerden oluşan sinyaldir. Bilginin işlendiği ortamlar dijitaldir.
- **Birim Darbe Sinyali:** Bir seviyeden diğerine, darbe genişliği adı verilen süre boyunca değişen sinyaldir, sonrasında orijinal seviyeye döner.
- **Clock Sinyali:** Eşit aralıklı sürekli elektriksel darbe katarı sinyalidir. Verileri temsil eden bitler (0/1) clock darbe katarının yükselen ya da düşen kenar ile tetiklenerek anlam kazanır. Veriler clock sinyalin yüksel ya da düşen kenarı ile tetiklenir. Verinin 1 ya da 0 olduğuna Clock sinyalinin yükselen ya da düşen kenar karar verir. **Verinin uzunluğu clock sinyalinin periyodu süresi kadar olmak zorundadır.**
- **Eşit aralıklarla tekrar eden sinyalin birim uzunluğudur.**
- **Bit:** Bir bilgisayardaki olası en küçük bilgi birimidir (0/1). Elektriksel sinyal olarak taşınır ve işlenir.
- **İkili (Binary) sayı sistemi:** Bit (0/1) ile temsil edilen sayı sistemidir.
- **Byte:** 8 bit veriyi temsil eder ya da 8 bitlik bellek birimini işaret eder.



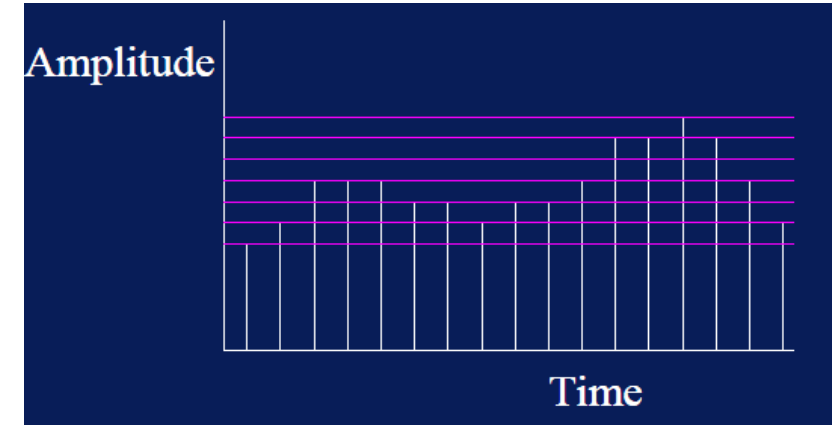
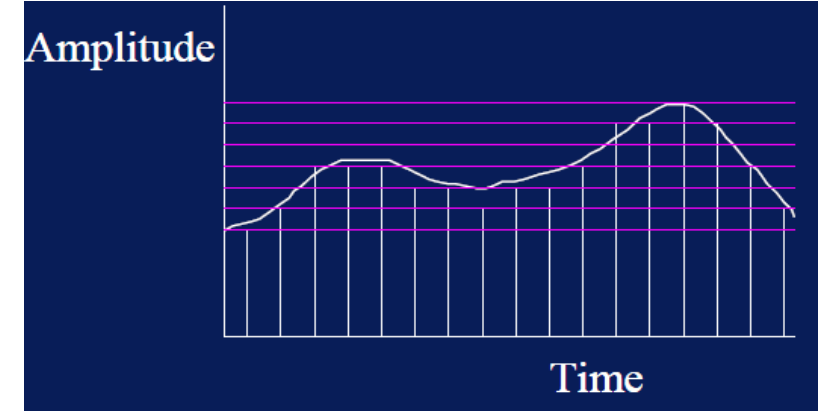
- **Clock period:** time interval between pulses
 - Above signal: period = 20 ns
- **Clock cycle:** one such time interval
 - Above signal shows 3.5 clock cycles
- **Clock frequency:** 1/period
 - Above signal: frequency = $1 / 20 \text{ ns} = 50 \text{ MHz}$
 - $1 \text{ Hz} = 1/\text{s}$

Freq	Period
100 GHz	0.01 ns
10 GHz	0.1 ns
1 GHz	1 ns
100 MHz	10 ns
10 MHz	100 ns

Digital Signals – Sampling Theorem

Analog ve Dijital Sinyaller

- Kainatı oluşturan tüm bileşenler ürettikleri mesajlar, analog sinyaller ile iletilir; bu sayede birbirleri ile etkileşimli iletişim halindedirler.
- Analog sinyal sayısal sinyale dönüştürülürken, genlik ve faz değerlerinden, belirli zaman aralıklarında (Örnekleme frekansı) örnekler alınır. Bu işleme örnek alma işlevi denir. Örnek alma işlemi gerçekleştirilirken örnek alma frekansı, analog sinyalin maksimum frekansının 2 katına eşit ya da büyük olmak zorundadır.
- Analog sinyalden alınan örneklerle ait değerler, genlik ölçeklendirme aralığında ayırık değerlere atanır. Bu işleme quantalama denir. Bu atama işlemi esnasında, örnek alma zaman aralığı, quantalama değerlerine öteleme ve çözünürlüğe bağlı olarak quantalama hataları oluşur.
- Ayırık değerler belirli sayıda ikili sayı sistemi ile temsil edilmesine ikili sayı sisteminde kodlama denir. Herbir ayırık genlik değeri belirli sayıdaki ikili (0/1) sayı sistemi ile temsil edilir. Böylece sayısal (dijital) sinyaller elde edilir.

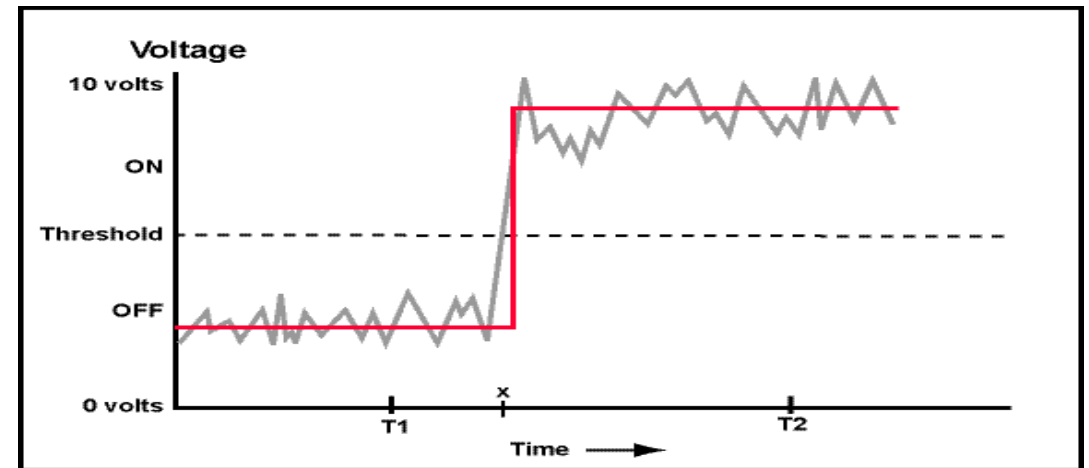
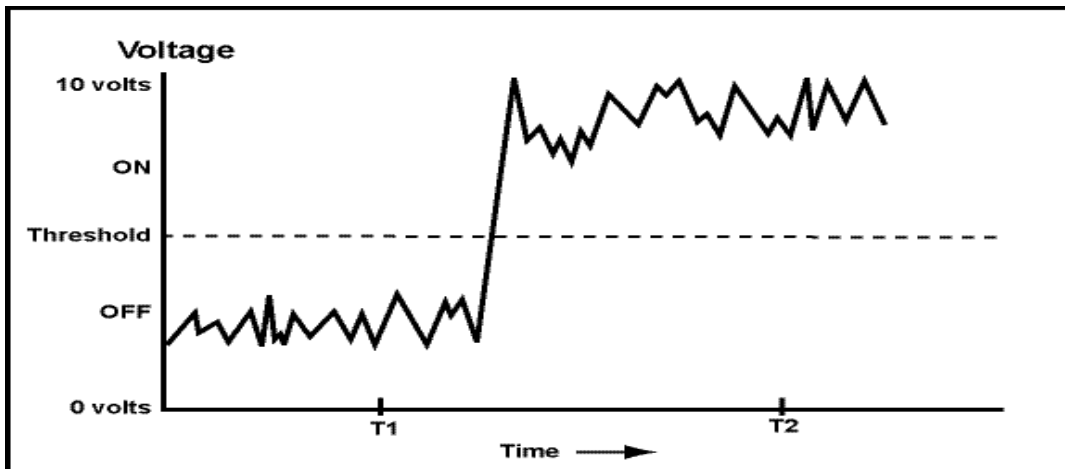


ANALOG-TO-DIGITAL CONVERSION

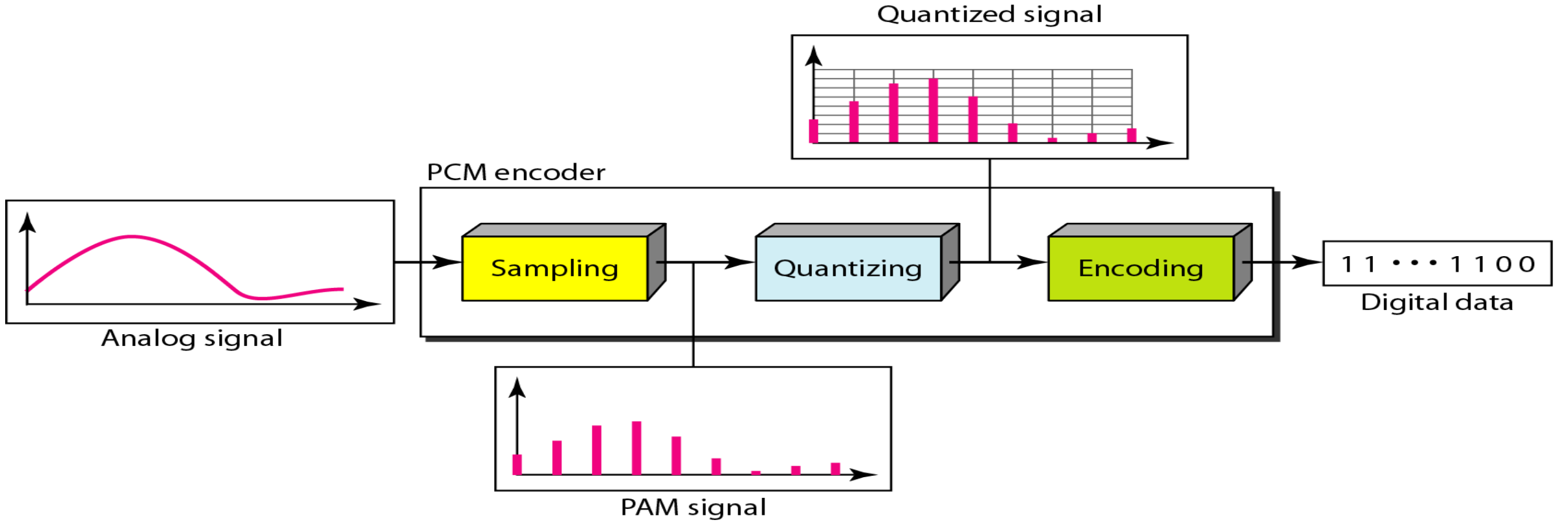
Dijital bir sinyal analog sinyale göre daha üstündür çünkü ekonomiktir, gürültüye karşı daha dayanıklıdır ve kolayca geri kazanılabilir, düzeltilebilir ve yükseltilebilir. Transistör üzerinde geçici ya da sürekli saklanabilmektedir. Bu sebeple, günümüzdeki eğilim bir analog sinyali dijital verilere değiştirmektir. Bilgisayar ya da sayısal sistemlerden bahsedilmektedir. İletim ortamında sayısal sinyal elverişli değildir. Sayısal sinyaller haberleşme kanallarında iletilirken analog sinyallere dönüştürülür.

Noise on Transmission

- When the signal is transferred it will pick up noise from the environment
- Even when the noise is present the binary values are transmitted without error
- Recovery - Filtering



Analog sinyalin sayısallaştırılması



- Analog sinyalden alınan örneklere ait değerler, genlik ölçeklendirme aralığında ayırık değerlere atanır. Bu işleme quantalama denir. Bu atama işlemi esnasında, örnek alma zaman aralığı, quantalama değerlerine öteleme ve çözünürlüğe bağlı olarak quantalama hataları oluşur.
- Ayırık değerler belirli sayıda ikili sayı sistemi ile temsil edilmesine ikili sayı sisteminde kodlama denir. Herbir ayırık genlik değeri belirli sayıdaki ikili (0/1) sayı sistemi ile temsil edilir. Böylece sayısal (dijital) sinyaller elde edilir.

Nyquist Örnekleme Teoremi

Bir analog sinyal örneklenip sayısala dönüştürüldükten sonra yeniden analog işarete dönüştürüldüğünde aynı sinyalin elde edilmesi için örnekleme frekansın sinyalin band genişliğinin iki katına eşit ya da büyük olması gerekmektedir.

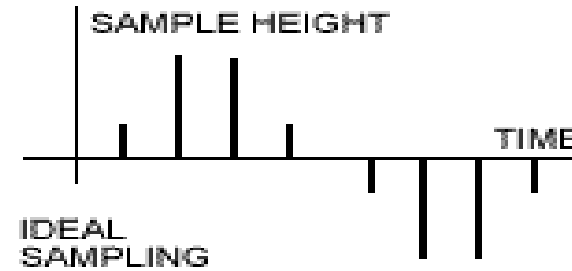
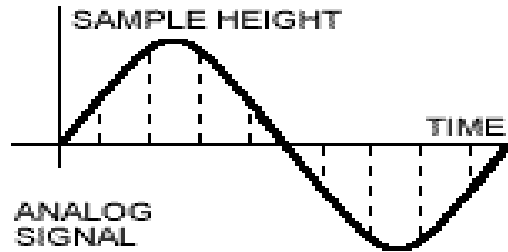
$$f_s \geq 2 * f_m$$

$$f_s \geq 2 * B$$

- burada f_s , örnekleme frekansıdır ve f_m , sinyaldeki maksimum frekanstır; B bant genişliğidir.
- Frekans bir saniyedeki peryod sayısıdır. $f_s = 1/T_s$ dir. T_s : peryod [saniye], f : frekans [Hz=1/saniye]. Örnek alma aralığıdır.
- Telefon haberleşmesinde band genişliği, $B = 4\text{KHz}$ dir (Anlama, Tanıma, Hissetme) ve $f_s = 2 * B$ alınır. O halde örnekleme aralığı $T_s = 1/8\text{KHz} = 125\text{mikrosaniye}$ dir. 125 mikrosaniyede bir analog sinyalde örnek alınır. Alınan örneği ayrık değer denir.

Örnekleme

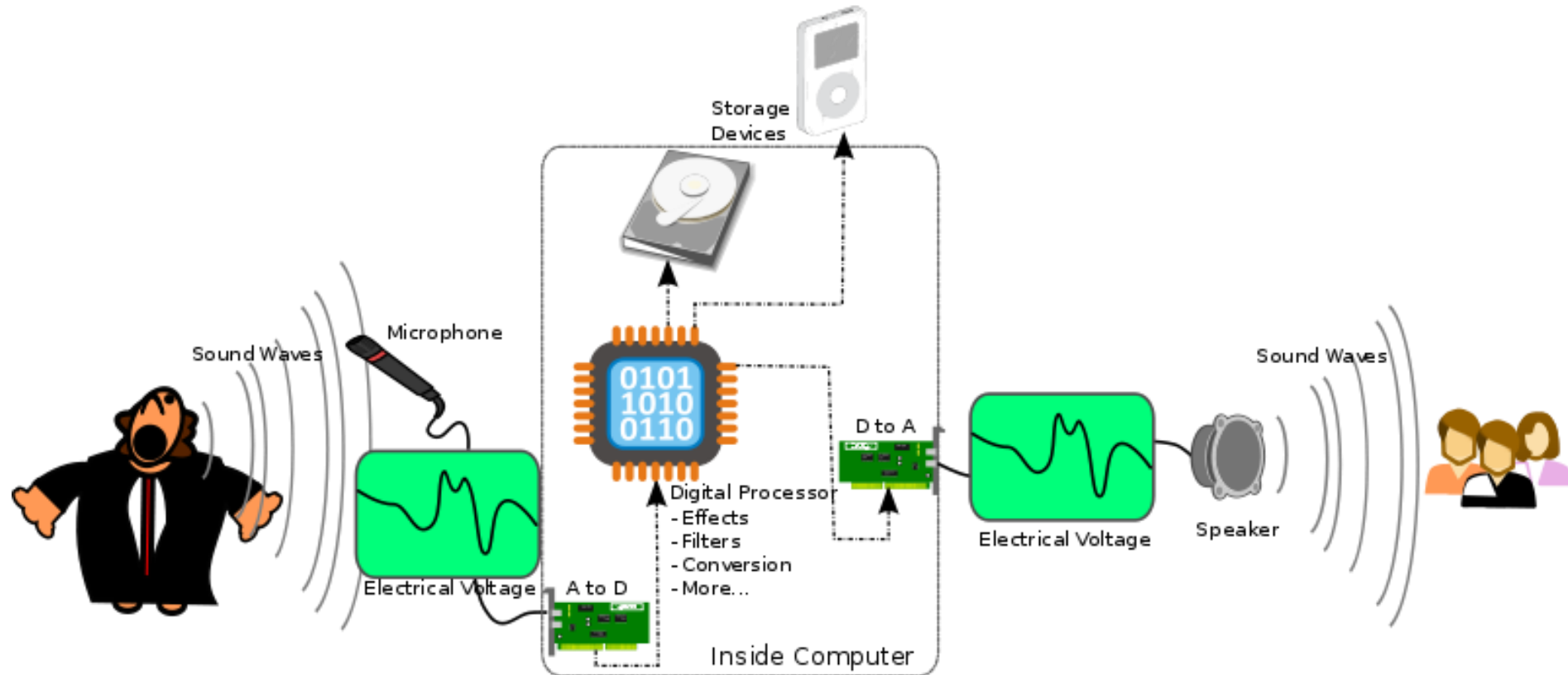
- Örnekleme, sürekli bir zaman sinyalini ayrık bir zaman sinyaline dönüştürmektir
- Örnekleme genellikle eşit zaman aralıklarında yapılır; bu aralığa örnekleme aralığı denir. Örnekleme aralığının karşılığına örnekleme frekansı veya örnekleme oranı denir. Örnekleme oranı birimi Hz dir.
- Örnekleme teoremine uygun olarak, telefon ses sinyalleri frekansı 300 Hz ila 3400 Hz arasındadır. 4KHz olarak alınır, örnekleme frekansı 8000 Hz'e eşit veya daha büyük alınır.
- Bir analog sinyalden saniyede eşit aralıklarla 8000 örnek alınıyor ve bir örnekleme 8 bit ile temsil ediliyorsa saniyede kaç bit örnek alınır? $8000 \times 8 = 64\,000 \text{ bit/sec} = 64\text{Kbit/sec}$.
- Veri haberleşmesinde bir kanalın minumum kapasitesi=64Kbit/s.
- Herbir örnek 8 bit ile temsil ediliyor ise, örnekleme aralık sayısı (Kuantalama) $2^8 = 256$ dır.
- 256 quantalama aralığı (katman) elde edilir.
- Sese dalagaları analog sinyale dönüştürüldükten sonra 125 μsec de bir örnek alınır. Örnekleme aralığı, $T = 1/8000 = 0.000125 \text{ sec.} = 125 \mu\text{s}$, T örnekleme aralığıdır.



Ideal Sampling and Aliasing

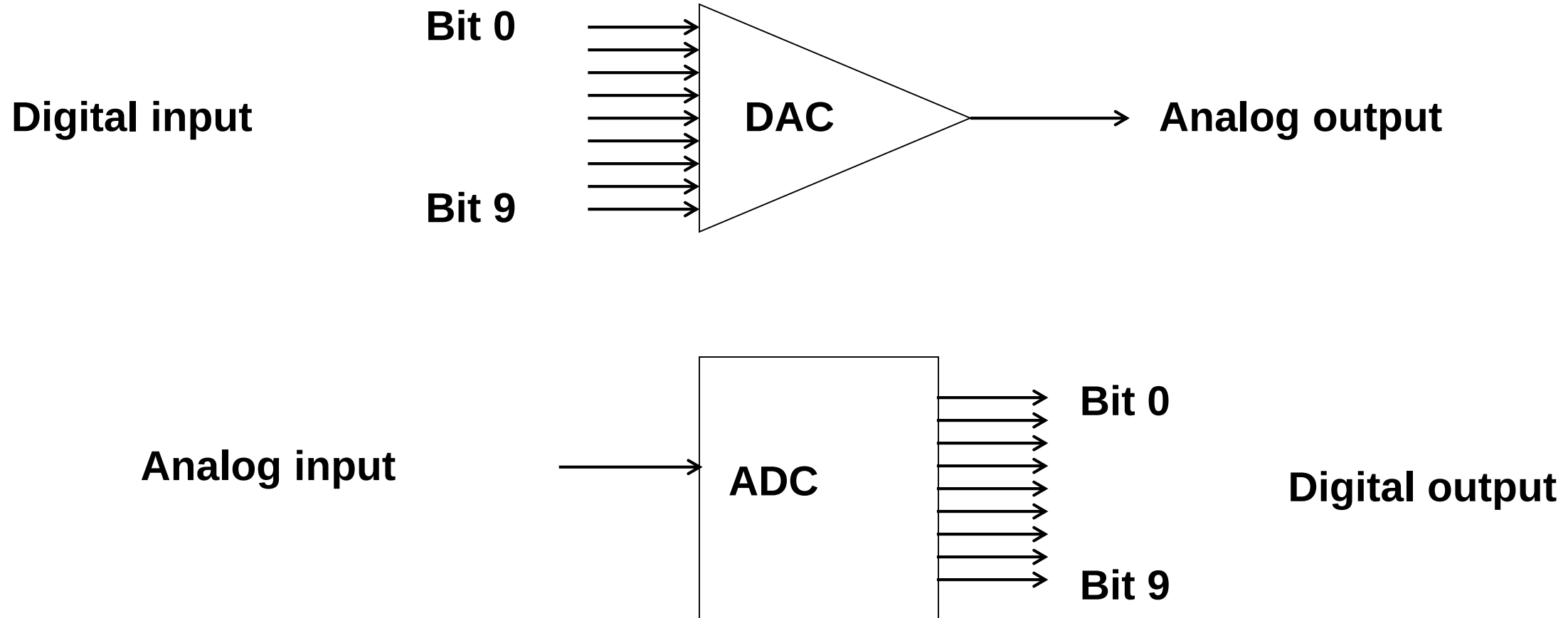
- Sampled signal is discrete in time domain with spacing T_s
- Spectrum will repeat for every f_s Hz
- Aliasing (spectral overlapping – Bir sinüsoidal sinyalin diğer bir sinüsoidal sinyal içerisinde kaybolması) if f_s is too small ($f_s < 2f_m$)
- Nyquist sampling rate $f_s = 2f_m$
- Generally oversampling is done $\rightarrow f_s > 2f_m$

Lifecycle from Sound to Digital to Sound

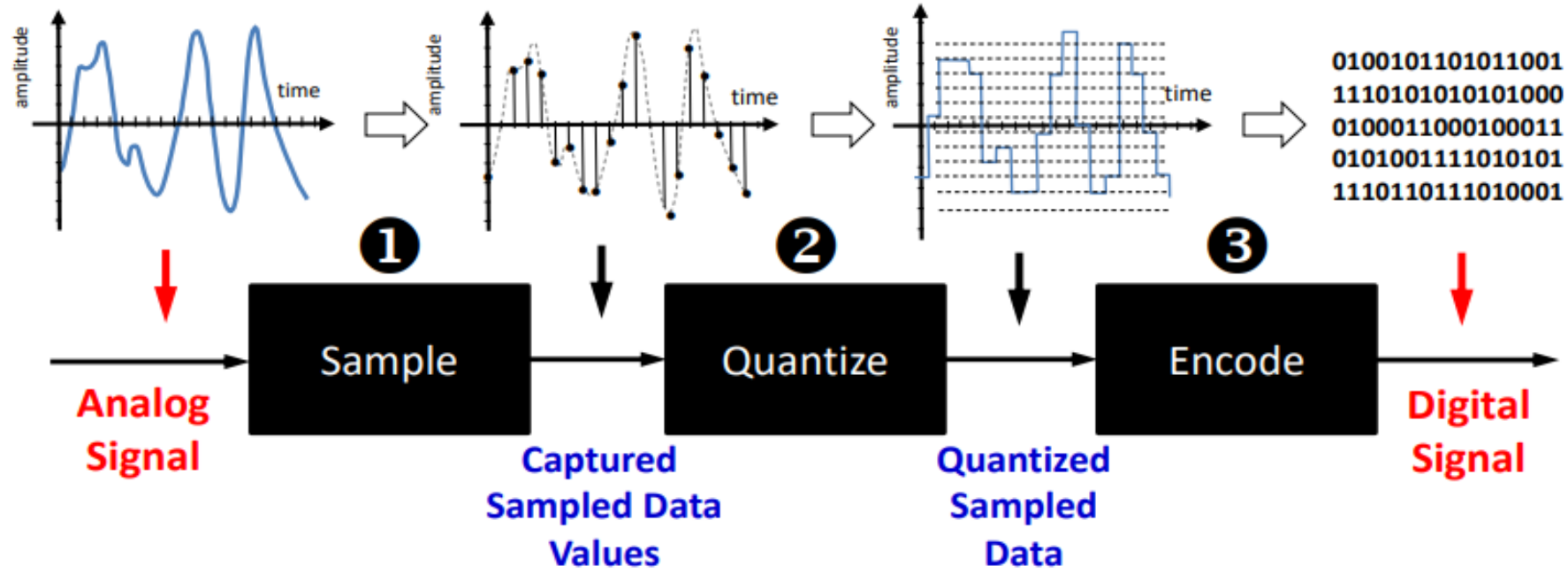


Source: http://en.wikipedia.org/wiki/Digital_audio

Analog – Digital Conversion



Analog to Digital Conversion Process (ADC)



Quantalama: Analog sinyalden alınan örneler bit dizisine dönüştürülürken (sayısala dönüştürülürken) katmanlar arasındaki değerler ya üst katmana ya da alt katmana ötelenir. Bu işleme quantalama denir.

Digital Design ve Boole Cebri

Sayısal Mantığın Temelleri

- Sayısal mantık değerlerini temsil etmek için ikili sayı sistemi kullanılır: 1/0 (Doğru /Yanlış, İyi /Kötü, Gece/Gündüz, 0V /5V)
- Sayısal mantık değerlerinin matematiksel işlemleri, Boole cebirinin kurallarında belirtilen yasalar tabidir.
- Boole cebirinin kurallarında belirtilen yasaların matematiksel girdileri ve çıktıları ikili sayı (1/0) sistemi ile temsil edilir.
- Bilgisayar sistemlerinin donanımını oluşturan mantıksal kapıları
 - AND, OR, NOT, NAND, NOR, XOR, ...
- Mantık kapıları, transistörler kullanılarak oluşturulur.
 - NOT gate can be implemented by a single transistor
 - AND gate requires 3 transistors
- Transistörler bilgisayar sistemlerinin temel devre elemanlarıdır.
 - Pentium consists of 3 million transistors
 - Compaq Alpha consists of 9 million transistors
 - Now we can build chips with more than 100 million transistors

Design Hierarchy

- Many digital systems can be divided into three design levels that form a well-defined hierarchy:
- The Architecture Level: High-level concerned with overall system management
- The Logic Level: Intermediate level concerned with the technical details of the system
- The Physical Level: Low level concerned with the details needed to manufacture or assemble the system
- We have already studied the architecture level
- Now we will address the logic level
- At the logic level, there are two classes of digital system
 - Combinational - digital systems without memory
 - Sequential - digital systems with memory

Boolean Algebra (Mantıksal devre)

- Definition: a logic variable x can have only one of two possible values or states
- $x = \text{TRUE}$
- $x = \text{FALSE}$
- In binary notation, we can say
- $x = \text{TRUE} = 1$
- $x = \text{FALSE} = 0$
- This is called positive logic or high-true logic
- Electrically, 1 is represented by a more positive voltage than zero and 0 is represented by zero volts
- $x = \text{TRUE} = 1 = 5 \text{ volts}$
- $x = \text{FALSE} = 0 = 0 \text{ volts}$

Boole Cebri Teoremleri

1. a) $a+b=b+a$ Değişme Özelliği

b) $a \cdot b = b \cdot a$

2. a) $a+b+c=a+(b+c)$ Birleşme Özelliği

b) $a \cdot b \cdot c = a \cdot (b \cdot c)$

3. a) $a+b \cdot c = (a+b) \cdot (a+c)$ Dağılma Özelliği

b) $a \cdot (b+c) = a \cdot b + a \cdot c$

c) $a(b+c)=ab+ac$

4. a) $a+a=a$ Değişkende Fazlalık Özelliği

b) $a \cdot a = a$

5. a) $a+a \cdot b=a$ Yutma Özelliği

b) $a \cdot (a+b)=a$

6. a) $(a)^n = a$ işlemde Fazlalık Özelliği

b) $(a \times n) = a$

7. a) $\overline{(a+b)} = \bar{a} \cdot \bar{b}$ De Morgan Kuralı

b) $\overline{(a \cdot b)} = \bar{a} + \bar{b}$

8. a) $0+a=a$ Etkisizlik Özelliği

b) $1 \cdot a = a$

9. a) $a+\bar{a}=1$ Sabit Özelliği

b) $a \cdot \bar{a} = 0$

10. a) $1+a=1$ Yutan Sabit Özelliği

b) $0 \cdot a = 0$

11. a) $(a+b) \cdot b = a \cdot b$

b) $a \cdot b + b = a + b$

12. a) $a+b \cdot a+c \cdot b+c = a+b \cdot (a+c)$

b) $a \cdot b + a \cdot c + b \cdot c = a \cdot b + a \cdot c$

13. a) $a+b \cdot a+c = a \cdot c + a \cdot b$

b) $a \cdot b + a \cdot c = a + c \cdot (a+b)$

14. a) $f(a,b,c,d,\dots) = [a+f(0,b,c,d,\dots)] \cdot [a+f(1,b,c,d,\dots)]$

Shannon Teoremi

b) $f(a,b,c,d,\dots) = a \cdot f(1,b,c,d,\dots) + [\bar{a} \cdot f(0,b,c,d,\dots)]$

- The 12 Rules of Boolean Algebra

- $A + 0 = A$
- $A + 1 = 1$
- $A \cdot 0 = 0$
- $A \cdot 1 = A$
- $A + A = A$
- $A + \bar{A} = 1$
- $A \cdot A = A$
- $A \cdot \bar{A} = 0$
- $\bar{\bar{A}} = A$
- $A + AB = A$
- $A + \bar{A}B = A + B$
- $(A + B)(A + C) = A + BC$

- **Boolean Algebra:** rules for rewriting Boolean functions
 - Useful for simplifying Boolean functions
 - Simplifying = reducing gate count, reducing gate “levels”
 - Rules: similar to logic (0/1 = F/T)
 - **Identity:** $A1 = A$, $A+0 = A$
 - **0/1:** $A0 = 0$, $A+1 = 1$
 - **Inverses:** $(A')' = A$
 - **Idempotency:** $AA = A$, $A+A = A$
 - **Tautology:** $AA' = 0$, $A+A' = 1$
 - **Commutativity:** $AB = BA$, $A+B = B+A$
 - **Associativity:** $A(BC) = (AB)C$, $A+(B+C) = (A+B)+C$
 - **Distributivity:** $A(B+C) = AB+AC$, $A+(BC) = (A+B)(A+C)$
 - **DeMorgan's:** $(AB)' = A'+B'$, $(A+B)' = A'B'$

Boole Cebirinin Kuralları ve Yasaları

Tek değişkenli temel kurallar:

- Boole Cebirinin Kurallarının ve Yasalarının her birinin bir kanıtı, değişkenin yalnızca iki bit(0/1) değere sahip olabileceği gerçeğinden yararlanılarak kolayca ispat edilebilir.
- Not: $A=0$ ya da 1 olur.
- $A + A + A + A + \dots + A + 1 = 1$; OR kapısında girişlerden herhangi biri 1 ise çıkış birdir. Diğer ispat etme yöntemi, 1 ve 0 değerleri verilerek doğruluk aranır.
- $A + A + A + \dots + A = A$ (Neden? İki değişkenli 0 ya da 1 girişler mevcuttur)
- $A + A + \dots + A = A$
 - If $A = 0$ then $0 + 1 = 1$
 - If $A = 1$ then $1 + 1 = 1$

$$A + 0 = A$$

$$A + 1 = 1$$

$$A + A = A$$

$$A + \bar{A} = 1$$

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot A = A$$

$$A \cdot \bar{A} = 0$$

It should be noted that $\bar{\bar{A}} = A$

Mantıksal devrelerde ve matematiğinde iki sayı var: 0 ve 1

Soru: Boole cebirini kullanarak aşağıdaki işlemi yapınız. $A=9$, $A+1=?$

a) 0 b) 1 c) 10 d) 8 e) hiçbir

Soru: Boole cebirinde A hangi değerleri alır? a) 0 b) 1 c) $0/1$ d) $0,1,2, \dots, 9$ e) Her değeri f) hiçbir değeri

Soru: Boole cebirinde $A=1$ ise, $A+A+A+A=?$ A) 1 B) 0 C) A D) 5 E) $5A$ F) Hiçbiri

Soru: Boole Cebirinde $A \cdot A \cdot A \cdot A=?$ A) A B) A^4

Rules and Laws of Boolean Algebra

- Operations on Boolean variables are defined by rules and laws, the most important of which are presented here
- Commutative Law
$$A \cdot B = B \cdot A$$
$$A + B = B + A$$
- This states that the order of the variables is unimportant
- Associative Law
$$A \cdot (B \cdot C) = A \cdot (B \cdot C)$$
$$A + (B + C) = A + (B + C)$$
- This states that the grouping of the variables is unimportant
- Distributive Law: $A \cdot (B + C) = A \cdot B + A \cdot C$
- This states that we can remove the parenthesis by ‘multiplying through’
- The above laws are the same as in ordinary algebra, where ‘+’ and ‘.’ are interpreted as addition and multiplication

Rules and Laws of Boolean Algebra

- Basic rules involving one variable:

$$A + 0 = A \quad A \cdot 0 = 0$$

$$A + 1 = 1 \quad A \cdot 1 = A$$

$$A + A = A \quad A \cdot A = A$$

$$A + A' = 1 \quad A \cdot A' = 0$$

- It should be noted that $A'' = A$
- An informal proof of each of these rules is easily accomplished by taking advantage of the fact that the variable can have only two possible values
- For example, rule 2: $A + 1 = 1$
 - If $A = 0$ then $0 + 1 = 1$
 - If $A = 1$ then $1 + 1 = 1$

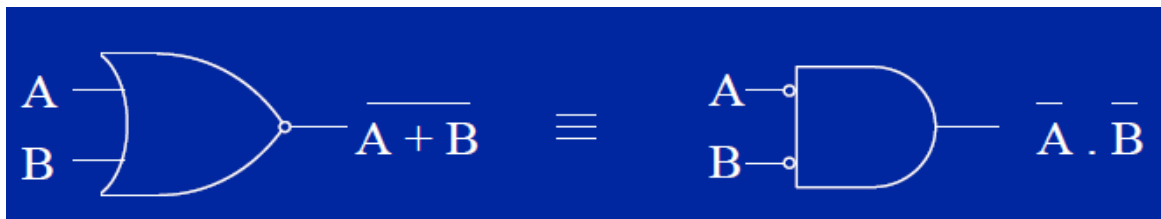
DeMorgan's Laws

$$\overline{A + B} = \bar{A} . \bar{B}$$

$$\overline{A . B} = \bar{A} + \bar{B}$$

A	B	A + B	$\overline{A + B}$	$\bar{A}$	$\bar{B}$	$\bar{A} . \bar{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

A	B	A . B	$\overline{A . B}$	$\bar{A}$	$\bar{B}$	$\bar{A} + \bar{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

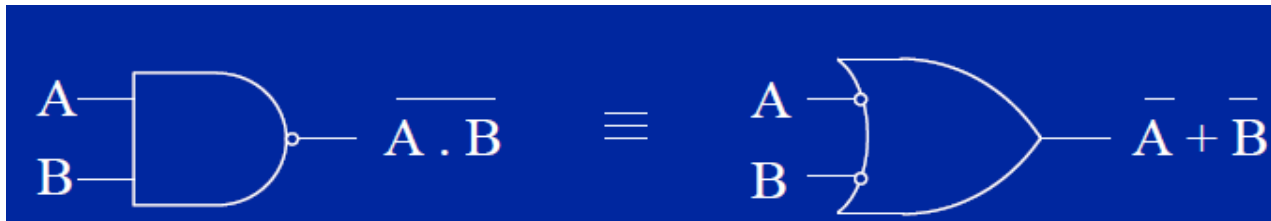


Boole Cebirinin Kuralları ve Yasaları

DeMorgan Yasaları, NAND ve NOR mantığı ile uğraşırken özellikle yararlıdır.

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$



Some useful theorems

- $A + A.B = A.(1+B) = A, \quad 1+B=1$
- $A + A'.B = (A+A').(A+B) = A+B, \quad A+A'=1$
- $A.B + AB' = A(B+B') = A, \quad B+B'=1$
- $A.(A+B) = A.A + AB = A + AB = A(1+B) = A, \quad A.A=A, \quad 1+B=1$
- $A(A'+B) = A.A' + AB = AB; \quad A.A'=0$
- $(A+B)(A+B') = AA + AB' + AB + BB' = A + AB' + AB = A(1+B+B') = A; \quad A.A=1, \quad B.B'=0, \quad 1+B+B'=1$
- $A + A'.B = A+B$

A	B	$\bar{A}$	$\bar{A}.B$	$A + \bar{A}.B$	$A+B$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

Mantıksal işlemlerin sonucu daima 1 ya da 0 dır.

- $a + a + a + a + \dots + a = a$
- $a * a * a * \dots * a = a$
- $1 + a + b + c + \dots + z = 1$
- $ab'c + ab'c = ab'c$; benzer ifadelerin toplamı her zaman benzer bir tanesine eşittir.

Boolean Algebra

- Developed by George Boole in 19th Century
 - Algebraic representation of logic
 - Encode “True” as 1 and “False” as 0

And

- $A \& B = 1$ when both $A=1$ and $B=1$

$\&$	0	1
0	0	0
1	0	1

Not

- $\sim A = 1$ when $A=0$

$\sim$	
0	1
1	0

Or

- $A \mid B = 1$ when either $A=1$ or $B=1$

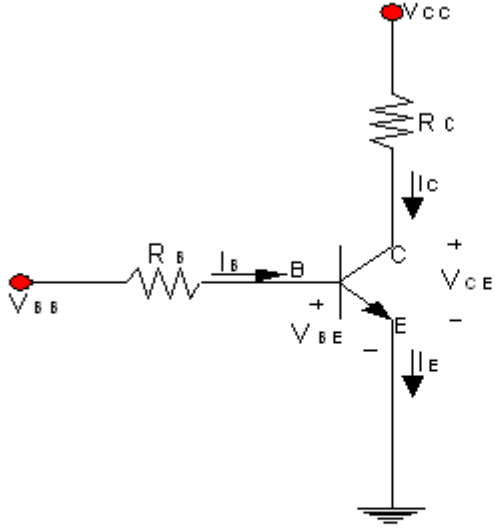
$\mid$	0	1
0	0	1
1	1	1

Exclusive-Or (Xor)

- $A \wedge B = 1$ when either $A=1$ or $B=1$, but not both

$\wedge$	0	1
0	0	1
1	1	0

Gates and Transistors



$$V_{CC} = R_C * I_C + V_{CE}$$

$$V_{BB} = R_B * I_B + V_{BE}$$

$$I_C = \beta * I_B$$

$$I_{C\ SAT} = \frac{V_{CC}}{R_C}$$

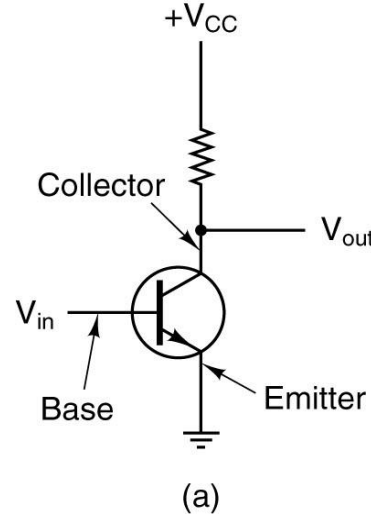
$$I_B = \frac{V_{BB} - V_{BE}}{R_B}$$

$V_{CE} \leq 0\ V$ ya da $I_C \geq I_{C\ SAT}$; Saturasyon $V_{CE} = 0\ V$ Olur

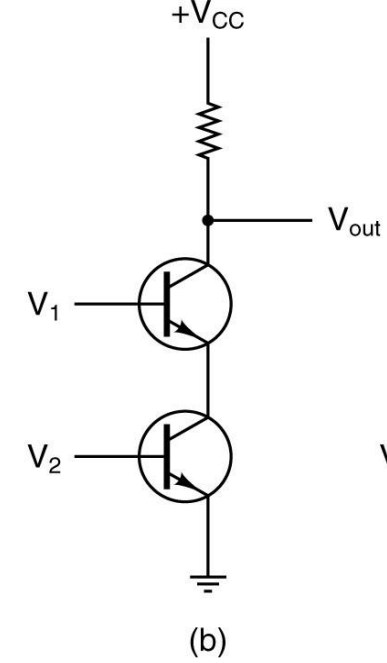
$I_B \leq 0\ A$ ise ; Kesmede $I_B = I_C = 0\ A$ Olur

VCE=VCC kesme durumunda

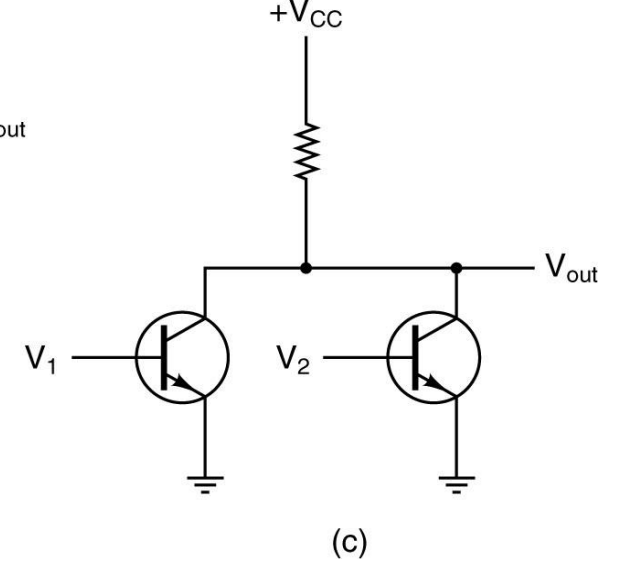
Transistör elektron akışını kontrol eden, yarı iletken teknolojisinde üretilen bir devre elemanıdır.



(a)



(b)



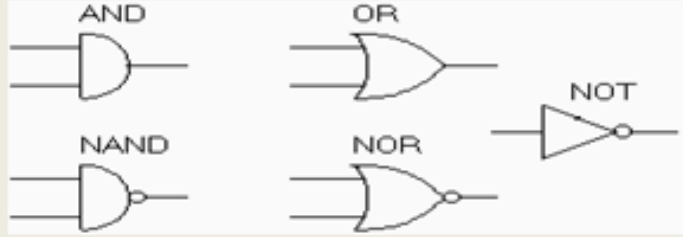
(c)

(a) A transistor inverter.

(b) A NAND gate.

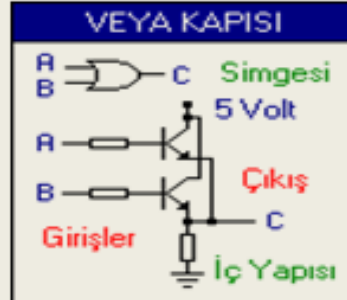
(c) A NOR gate.

Gates



Doğruluk tablosu:

A	B	OR $A+B$	AND $A*B$	NOT A'	NOR $(A+B)'$	NAND $(A*B)'$	EXOR $(A')*B+A*(B')$
0	0	0	0	1	1	1	0
0	1	1	0	1	0	1	1
1	0	1	0	0	0	1	1
1	1	1	1	0	0	0	0



Formüller	0 Değeri Verildiğinde	1 Değeri Verildiğinde
$A \cdot 0 = 0$	$A = 0$ ise, $0 \cdot 0 = 0$	$A = 1$ ise, $1 \cdot 0 = 0$
$A \cdot 1 = A$	$A = 0$ ise, $0 \cdot 1 = 0$	$A = 1$ ise, $1 \cdot 1 = 1$
$A + 0 = A$	$A = 0$ ise, $0 + 0 = 0$	$A = 1$ ise, $1 + 0 = 1$
$A + 1 = 1$	$A = 0$ ise, $0 + 1 = 1$	$A = 1$ ise, $1 + 1 = 1$
$A \cdot A = A$	$A = 0$ ise, $0 \cdot 0 = 0$	$A = 1$ ise, $1 \cdot 1 = 1$
$A + A = A$	$A = 0$ ise, $0 + 0 = 0$	$A = 1$ ise, $1 + 1 = 1$
$A \cdot A' = 0$	$A = 0$ ise, $0 \cdot 1 = 0$	$A = 1$ ise, $1 \cdot 0 = 0$
$A + A' = 1$	$A = 0$ ise, $0 + 1 = 1$	$A = 1$ ise, $1 + 0 = 1$
$(A')' = A$	$A = 0$ ise, $A' = 1, (A')' = 0$	$A = 1$ ise, $A' = 0, (A')' = 1$

Sadeleştirmeler

$$(A + B) = (B + A)$$

$$A + B \cdot C = A + (B \cdot C) = A + B \cdot C$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C) = A \cdot B \cdot C$$

$$(A + B) \cdot (A + C) = A + (B \cdot C)$$

$$(A' \cdot B) + (A \cdot B') = A \oplus B$$

$$(A + B)' = A' \cdot B'$$

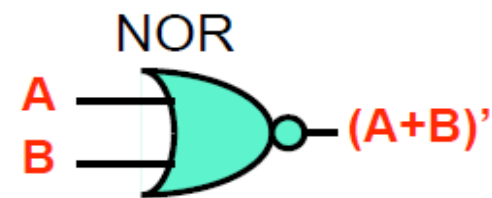
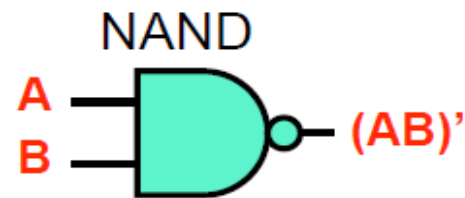
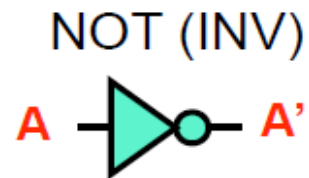
$$(A \cdot B) = (B \cdot A)$$

$$(A' \cdot B') + (A \cdot B) = (A \oplus B)'$$

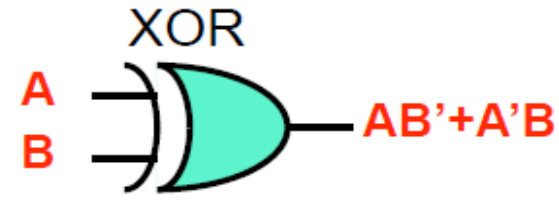
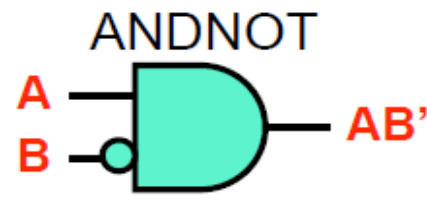
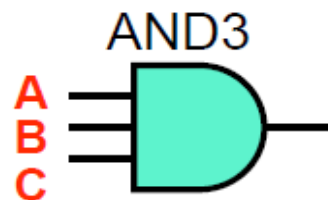
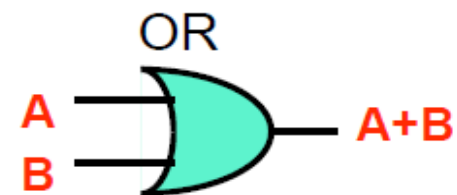
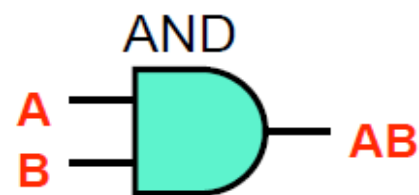
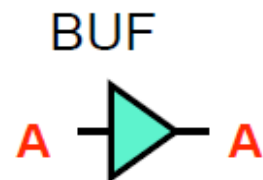
$$(A \cdot B)' = A' + B'$$

Logic Gates

- **Logic gates:** implement Boolean functions
 - Basic gates: NOT, NAND, NOR
 - Underlying CMOS transistors are naturally inverting ($\circ$ = NOT)



- NAND, NOR are "Boolean complete"



Temel Lojik Kapılar -1

- Simple gates
 - AND
 - OR
 - NOT
- Functionality can be expressed by a truth table
 - A truth table lists output for each possible input combination
- Precedence
 - NOT > AND > OR
 - $F = A B + A \overline{B}$
 $= (A (B)) + ((A) \overline{B})$

— — —
— — —

Gate

Symbol

Truth-Table

Expression

NAND



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

$$Z = \overline{X \cdot Y}$$

AND



X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

$$Z = X \cdot Y$$

NOR



X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

$$Z = \overline{X + Y}$$

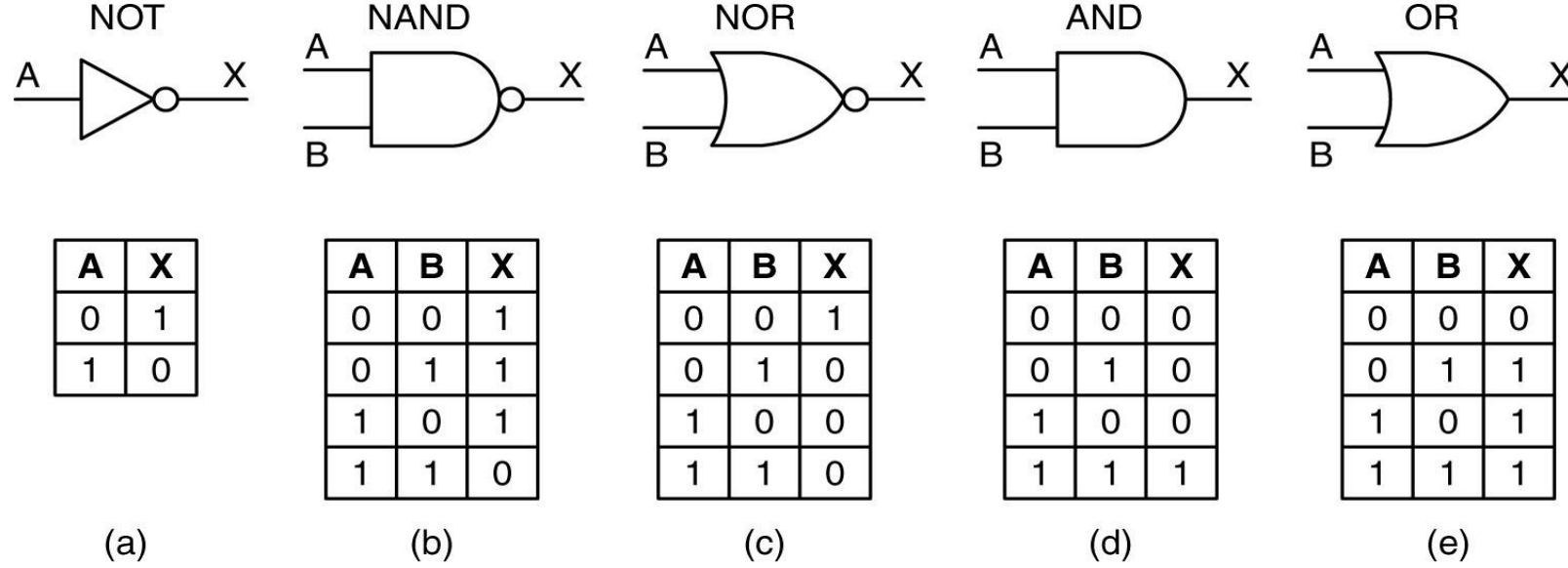
OR



X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

$$Z = X + Y$$

Mantık Kapıları için semboller ve işlevsel davranış



AND Kapısı: Girişlerden herhangi biri 0 ise çıkış 0 dır. Girişlerin tümü 1 ise çıkış 1 dir.

OR Kapısı: Girişlerden herhangi biri 1 ise çıkış 1 dır. Girişlerin tümü 0 ise çıkış 0 dir.

NOT Kapısı: girişin evriğini alır.

Mantık kapısı, bir Boole işlevini uygulayan idealleştirilmiş veya fiziksel bir devredir, yani bir veya daha fazla mantık girişinde mantıksal bir işlem gerçekleştirir ve tek bir mantık çıkışı üretir.

Logic AND Gates

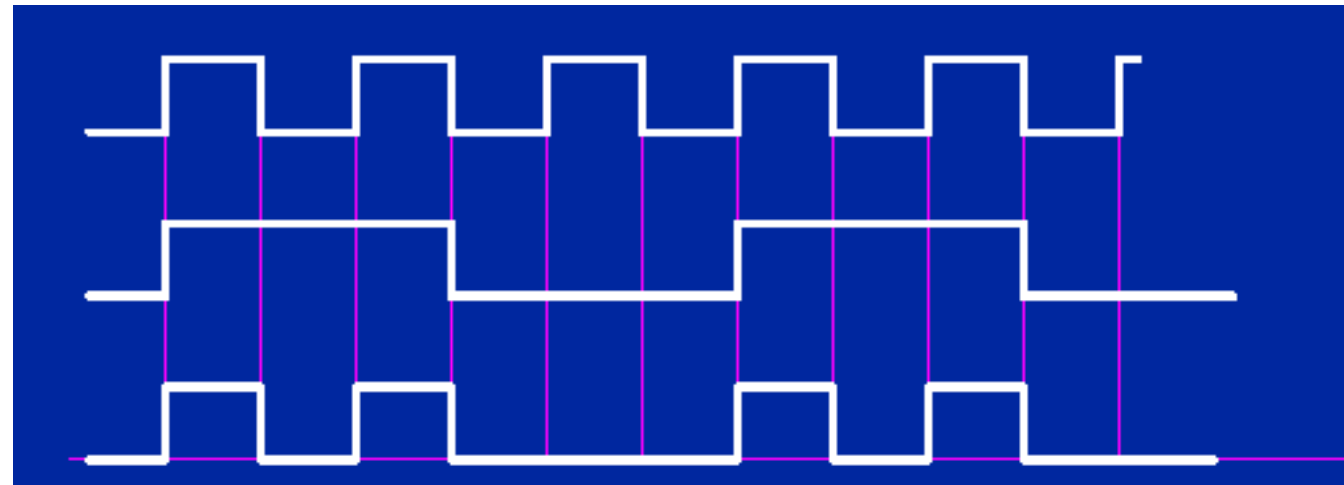
- Logic gates are switching circuits that perform certain simple operations on binary signals
- These operations are chosen to facilitate the implementation of useful functions
 - The AND Gate - Determine the output waveform when the input waveforms A and B are applied to the two inputs of an AND gate
 - A and B are variables and note the use of the . to denote AND
 - Giriş dalga formları A ve B bir mantık kapısının iki girişine uygulandığında çıkış dalga formu belli ise bu kapının türünü belirleyiniz. (AND Kapısı)

The AND Gate



The AND Truth Table

A	B	$f = A \text{ AND } B$
0	0	0
0	1	0
1	0	0
1	1	1



Logic OR Gates

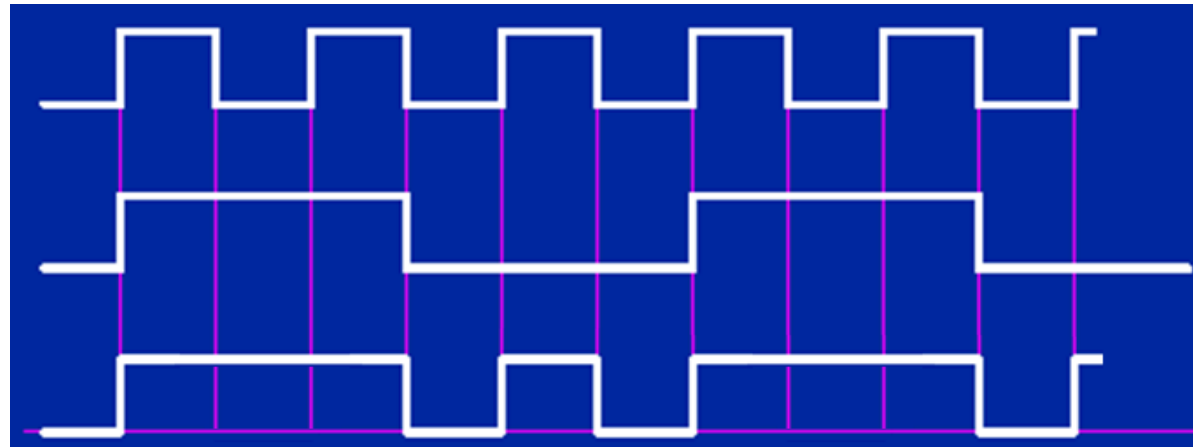
- A and B are variables and note the use of the + to denote OR

The OR Gate



The OR Truth Table

A	B	$f = A \text{ OR } B$
0	0	0
0	1	1
1	0	1
1	1	1



Logic NOT Gates

- Note the use of the bar over the A to denote NOT

The NOT Gate

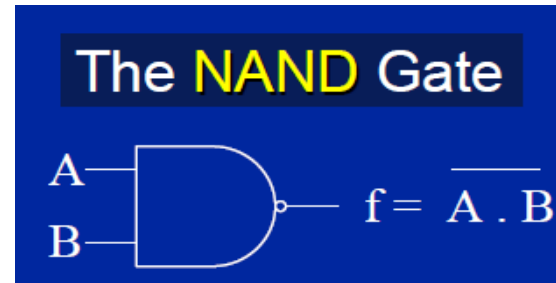
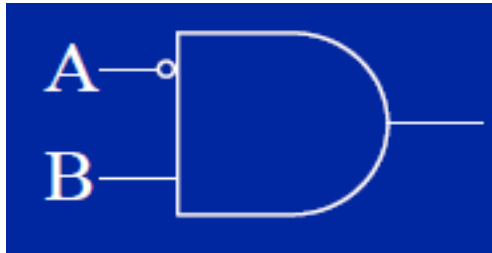


The NOT Truth Table

A	f = NOT A
0	1
1	0

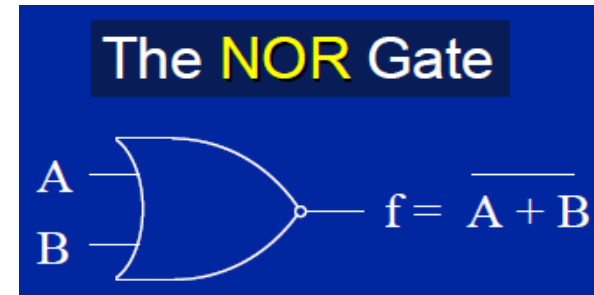
Logic Gates

- Sometimes a 'bubble' is used to indicate Inversion
- In fact it is simpler to manufacture the combination NOT AND and NOT OR than it is to deal with AND and OR
- NOT AND becomes NAND
- NOT OR becomes NOR



The **NAND** Truth Table

A	B	$f = A \text{ NAND } B$
0	0	1
0	1	1
1	0	1
1	1	0



The **NOR** Truth Table

A	B	$f = A \text{ NOR } B$
0	0	1
0	1	0
1	0	0
1	1	0

Logic Gates

Toplama

The EXCLUSIVE OR Truth Table

A	B	f = A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Karşılaştırma

The EXCLUSIVE NOR Truth Table

A	B	f = A XOR B
0	0	1
0	1	0
1	0	0
1	1	1

The XOR Gate



EXCLUSIVE NOR Gate



This is called the equivalence gate

Karşılaştırma ve Aritmetik toplama işlemlerinde XOR kapıları kullanılır.

Girişlerin tümü birbirine eşit (0 ya da 1) çıkış sıfır ise XOR, çıkış 1 ise XNOR kapısıdır.

The XOR – XNOR Gates

- The **EXCLUSIVE OR** Truth Table

$$f = A \text{ XOR } B$$

$$= A \oplus B$$

$$= \overline{A}B + A\overline{B}$$

A	B	f = A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

The **XOR** Gate



The **EXCLUSIVE NOR** Truth Table

$$f = \text{NOT } (A \text{ XOR } B)$$

$$= \overline{A \oplus B}$$

$$= \overline{A}B + AB$$

A	B	f = A XOR B
0	0	1
0	1	0
1	0	0
1	1	1

The **EXCLUSIVE NOR** Gate



Örnek:

:

- A=1 ve B=1 olduğunda ya da C=1 ve D=1 olduğunda alarm çalsın.

$$F=AB+CD$$

AB durumunda, m12, m13, m14, m15

CD durumunda m3,m7,m11, m15

Row	A	B	C	D		Minterm
0	0	0	0	0	$\bar{A}\bar{B}\bar{C}\bar{D}$	m0
1	0	0	0	1	$\bar{A}\bar{B}\bar{C}D$	m1
2	0	0	1	0	$\bar{A}\bar{B}C\bar{D}$	m2
3	0	0	1	1	$\bar{A}\bar{B}CD$	m3
4	0	1	0	0	$\bar{A}B\bar{C}\bar{D}$	m4
5	0	1	0	1	$\bar{A}B\bar{C}D$	m5
6	0	1	1	0	$\bar{A}BC\bar{D}$	m6
7	0	1	1	1	$\bar{A}BCD$	m7
8	1	0	0	0	$A\bar{B}\bar{C}\bar{D}$	m8
9	1	0	0	1	$A\bar{B}\bar{C}D$	m9
10	1	0	1	0	$A\bar{B}C\bar{D}$	m10
11	1	0	1	1	$A\bar{B}CD$	m11
12	1	1	0	0	$AB\bar{C}\bar{D}$	m12
13	1	1	0	1	$AB\bar{C}D$	m13
14	1	1	1	0	$ABC\bar{D}$	m14
15	1	1	1	1	$ABCD$	m15

Logic Functions

Truth Table: 3-input majority function

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

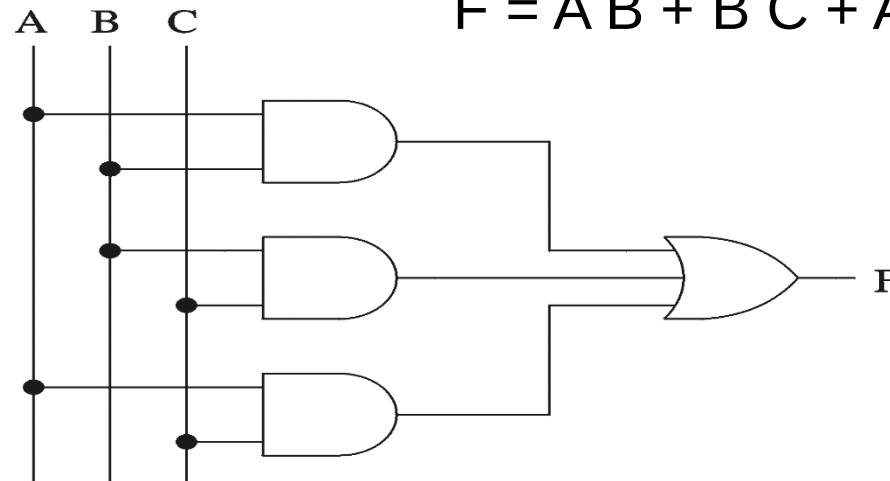
- **Logical functions can be expressed in several ways:**

- Doğruluk Tablosu
- Mantıksal İfade
- Gösterim

- **Logical expression form**

$$F = A'BC + AB'C + ABC' + ABC$$

$$F = AB + BC + AC$$



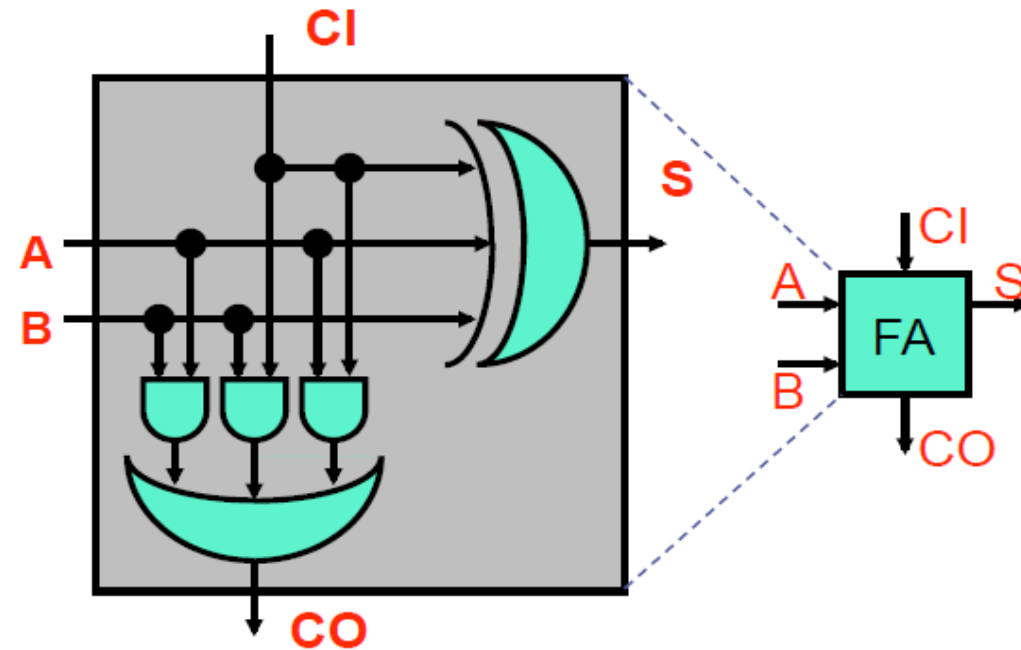
- Verilen denklemlerdeki değişkenlerin toplamı giriş sayısını verir.
- Mantık kapılarında mantıksal denklemin sonucu 1 ya da 0'dır.

Giriş sayısı, m olursa kaç farklı durum vardır? $S = 2^m$ durum vardır. 2 tabanlı olmasının nedeni ikili sayı (binary) sisteminden kaynaklanmaktadır: 0 ya da 1, bit

Full Adder

- What is the logic for a full adder?
 - Look at truth table

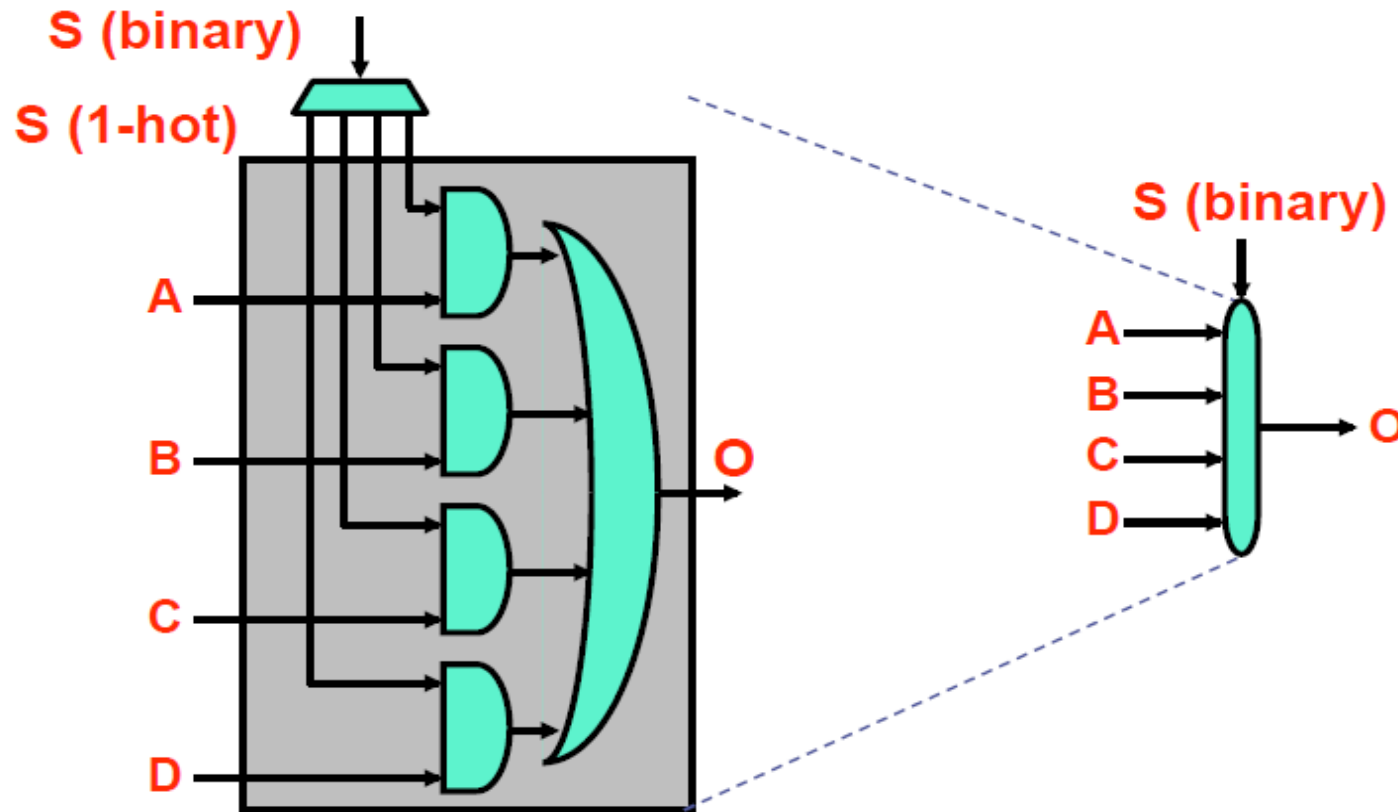
<u>CI</u>	<u>A</u>	<u>B</u>	<u>→</u>	<u>CO</u>	<u>S</u>
0	0	0	→	0	0
0	0	1	→	0	1
0	1	0	→	0	1
0	1	1	→	1	0
1	0	0	→	0	1
1	0	1	→	1	0
1	1	0	→	1	0
1	1	1	→	1	1



- $S = C'A'B + C'AB' + CA'B' + CAB = C \wedge A \wedge B$
- $CO = C'AB + CA'B + CAB' + CAB = CA + CB + AB$

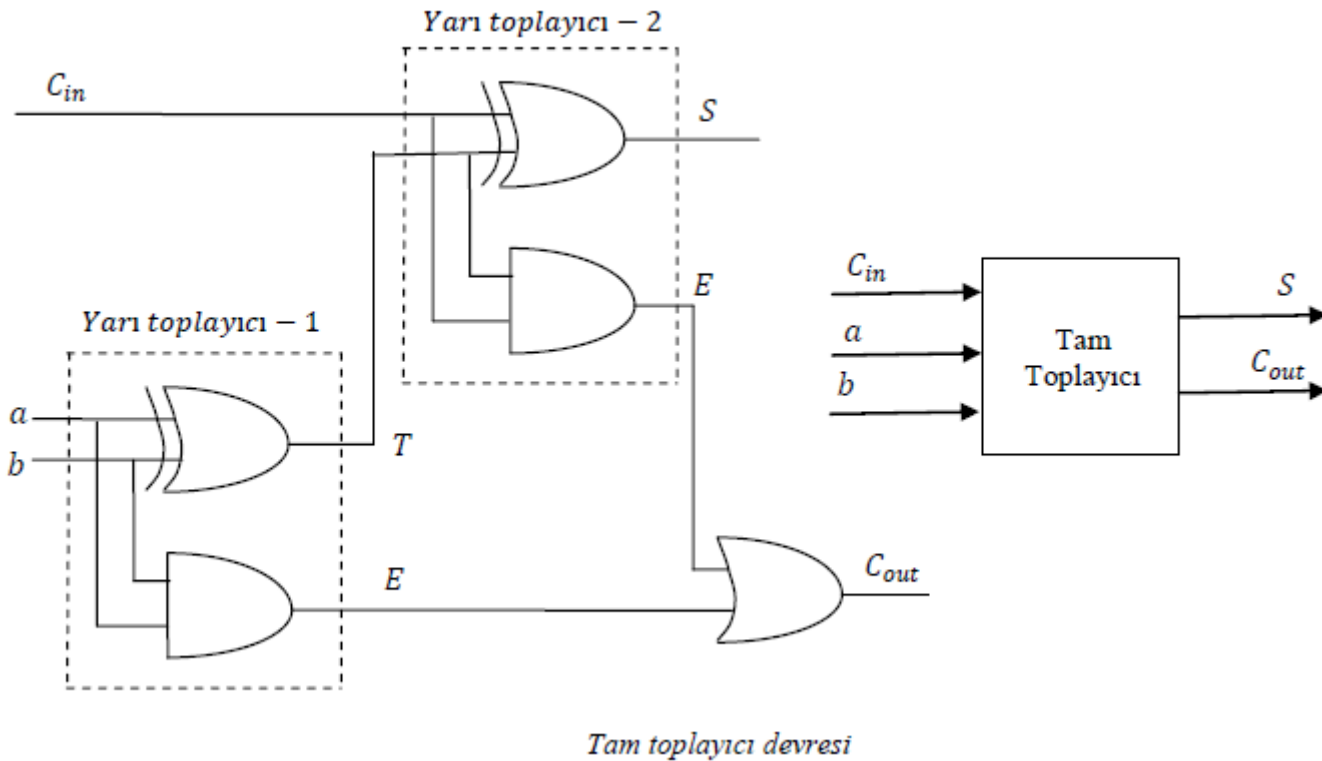
Multiplexer (mux): selects output from N inputs

- Example: 1-bit 4-to-1 mux
- Not shown: N-bit 4-to-1 mux = N 1-bit 4-to-1 muxes + 1 decoder



Tam toplayıcı (Full Adder)

- Girişinde elde bitinin olduğu ve bir bitlik iki sayı ile birlikte toplandığı bir kombinyasyonel devredir.



C_{in}	a	b	Toplam S	Elde C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$ab \backslash C_{in}$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$ab \backslash C_{in}$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

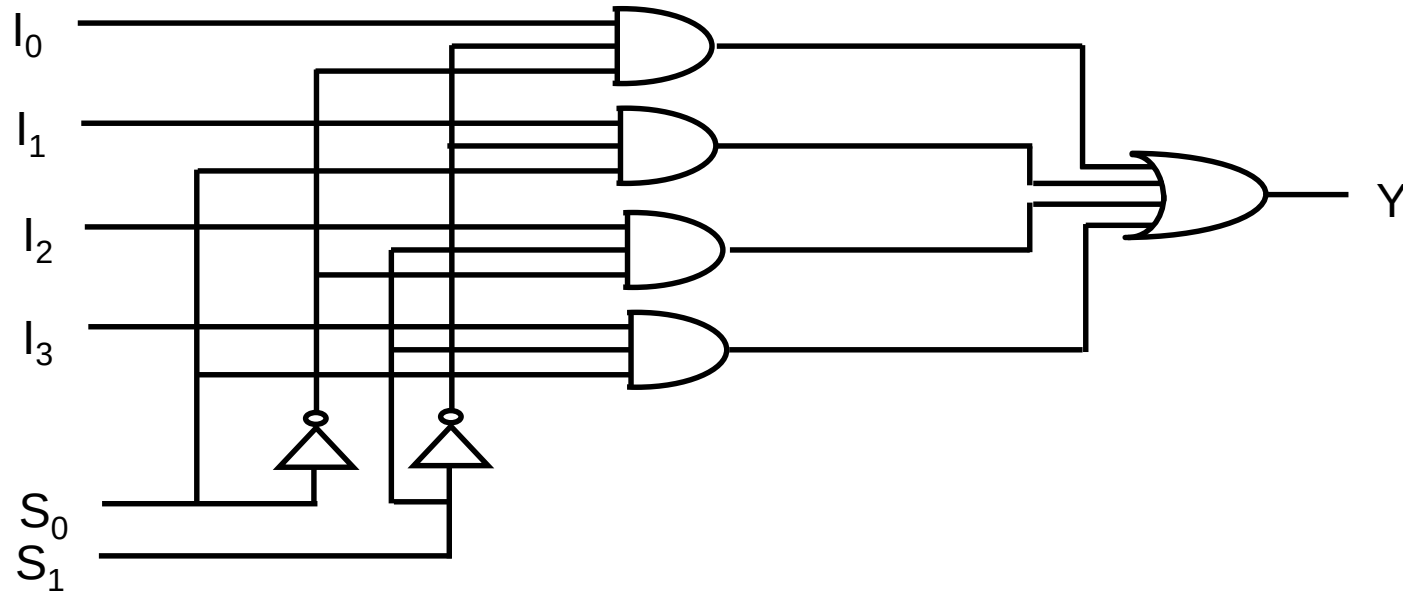
$$T = C_{in} \bar{a} \bar{b} + \bar{C}_{in} \bar{a} b + C_{in} a b + \bar{C}_{in} a \bar{b} = \bar{C}_{in} (\bar{a} b + a \bar{b}) + C_{in} (a b + \bar{a} \bar{b}) \Rightarrow T = a \oplus b \oplus C_{in}$$

$$C_{out} = C_{in} b + C_{in} a + a b$$

MULTIPLEXER

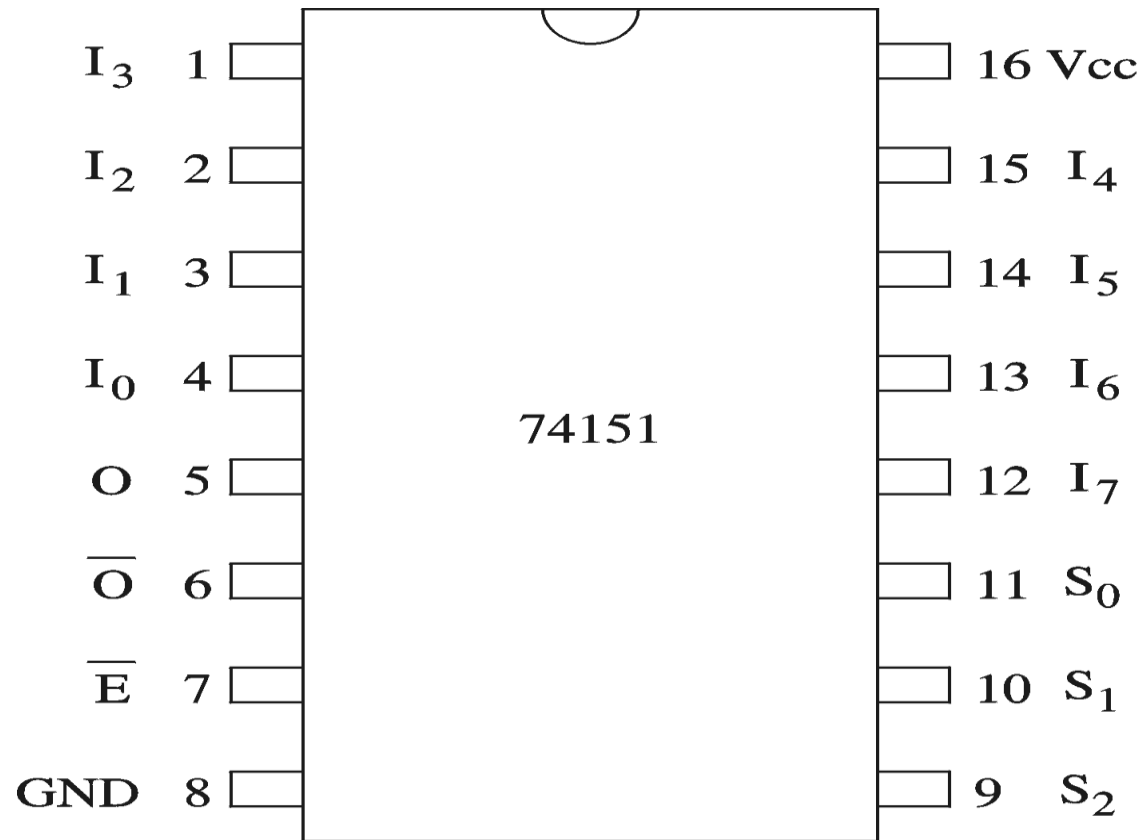
4-to-1 Multiplexer

Select		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

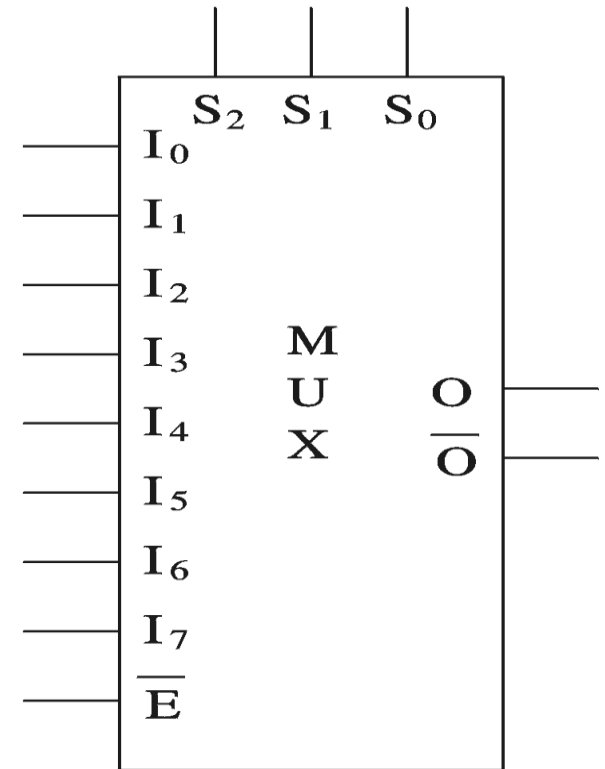


Multiplexers

Example chip: 8-to-1 MUX



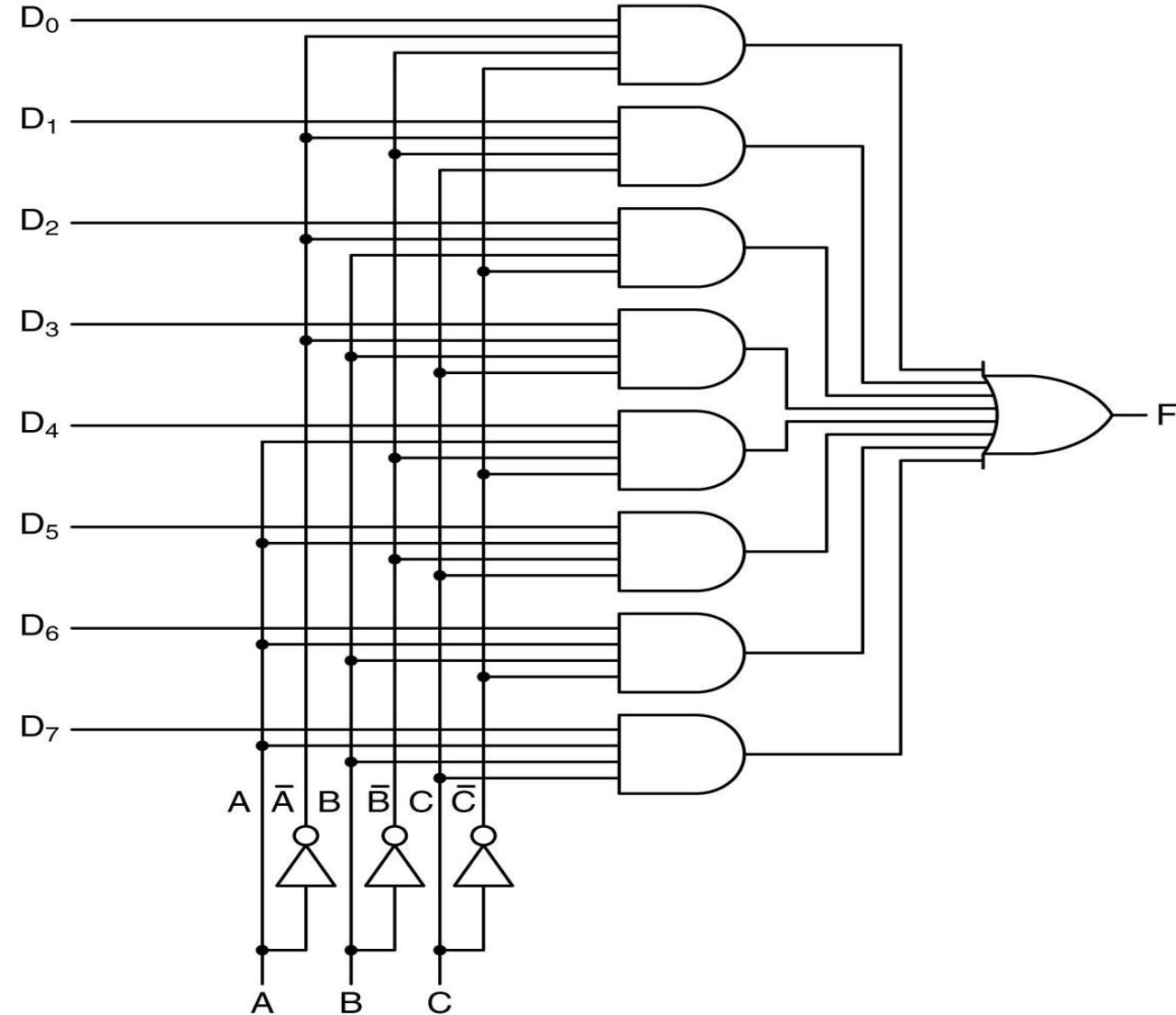
(a) Connection diagram



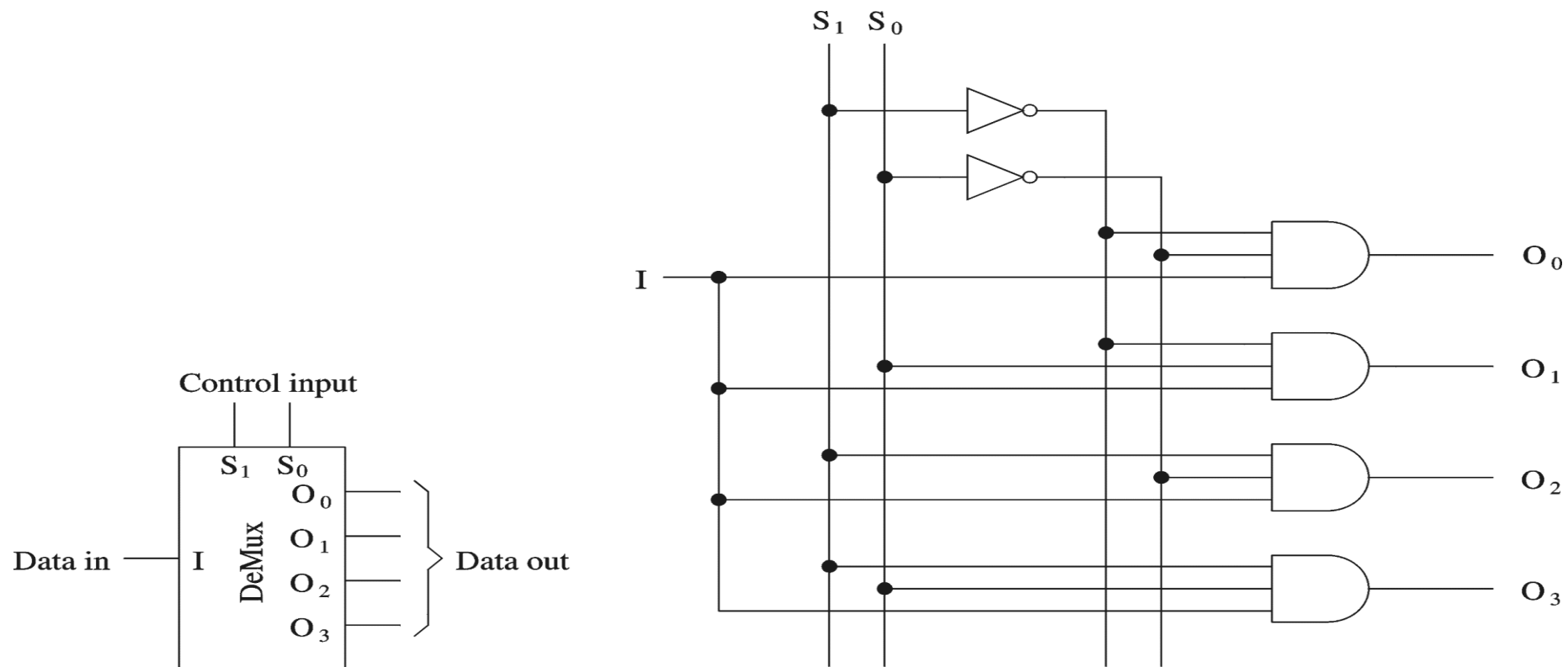
(b) Logic symbol

Multiplexers

An eight-input multiplexer circuit.



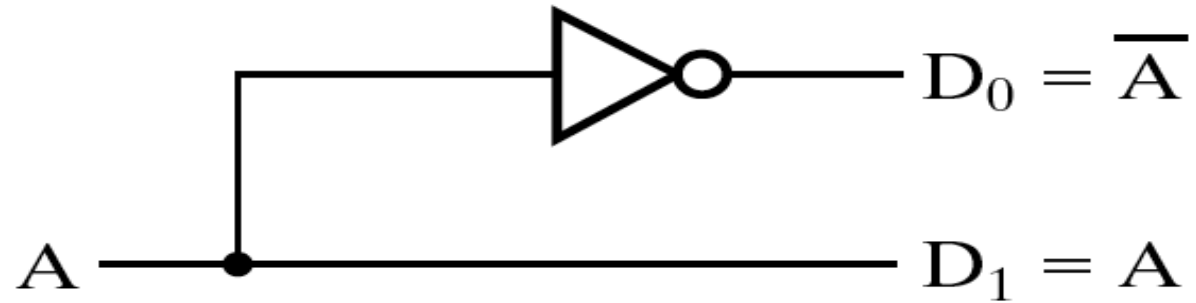
Demultiplexer (DeMUX)



1-2 Decoder

A	D₀	D₁
0	1	0
1	0	1

(a)

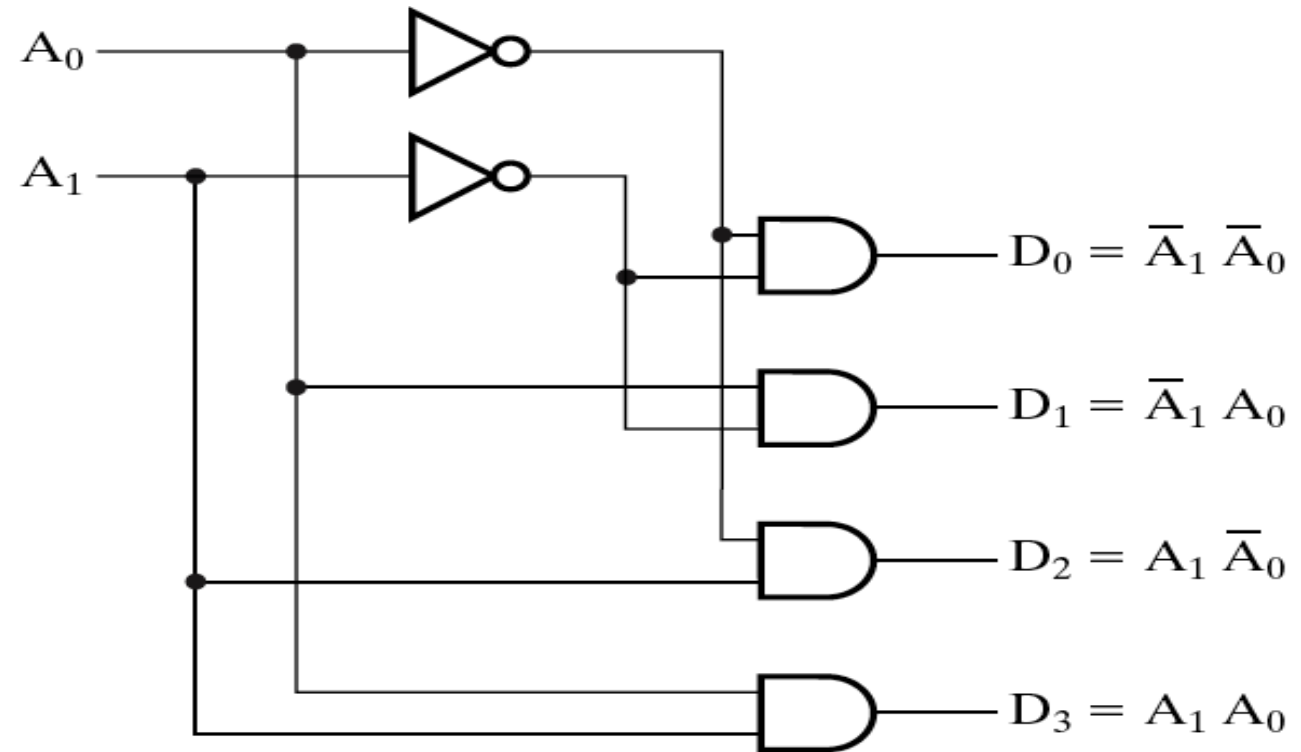


(b)

2-to-4 Decoder

A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

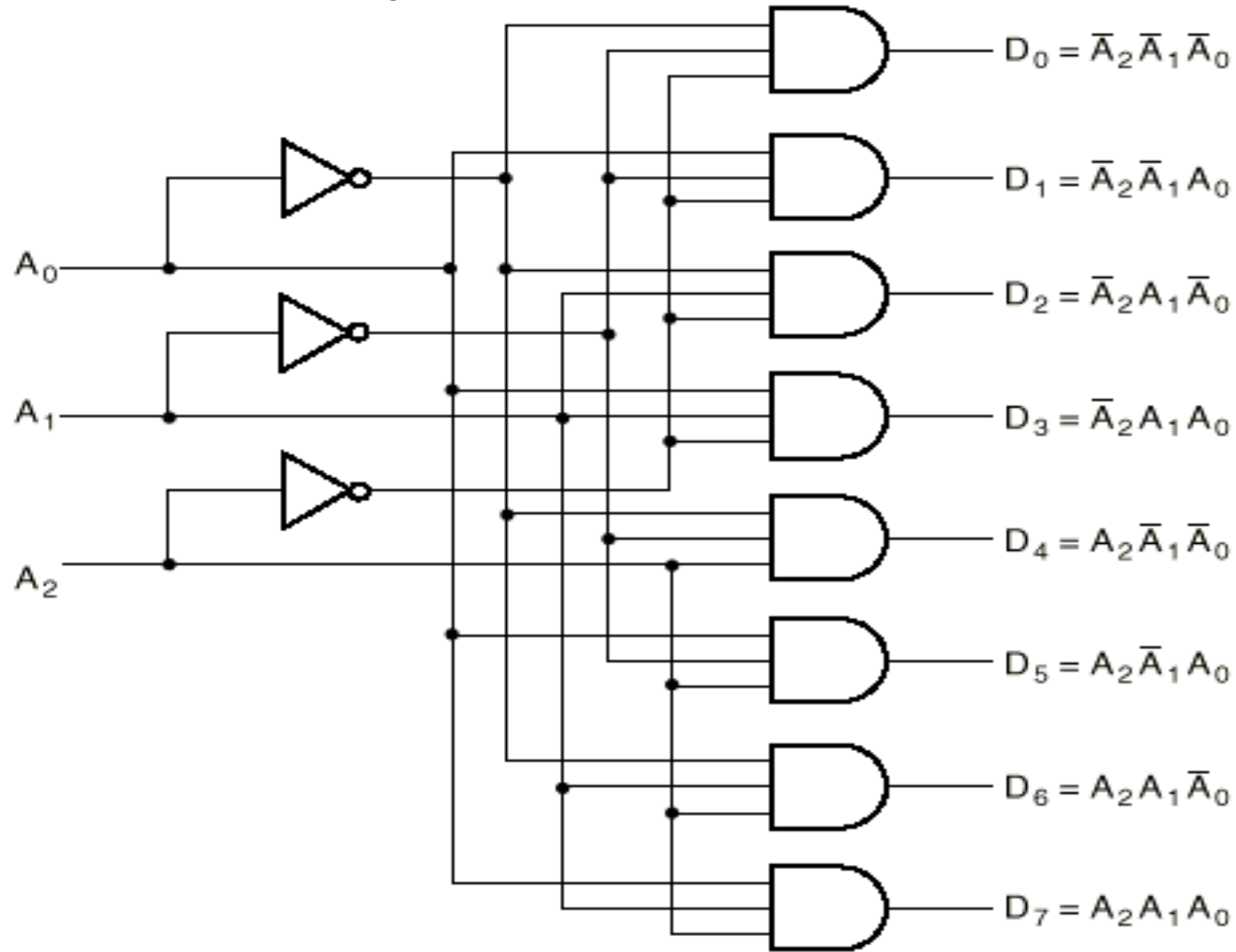
(a)



(b)

3-to-8 Decoder

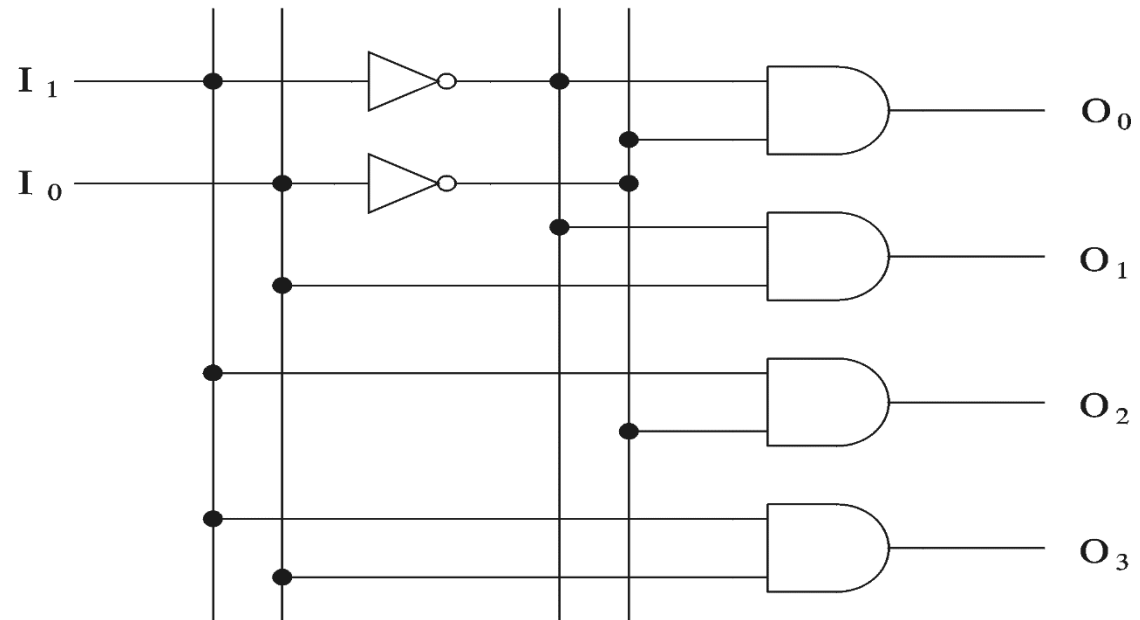
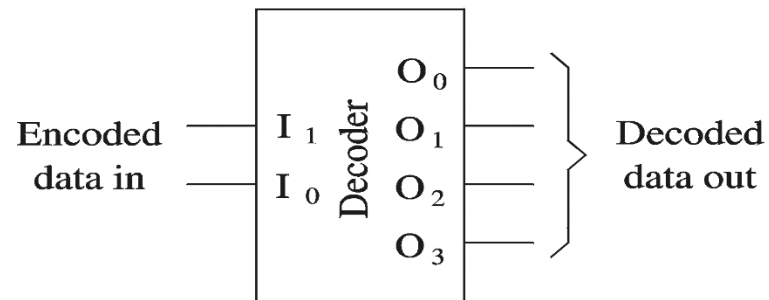
- Decoding devresi bellek seçmede kullanılır.



Decoders

- Decoder selects one-out-of-N inputs

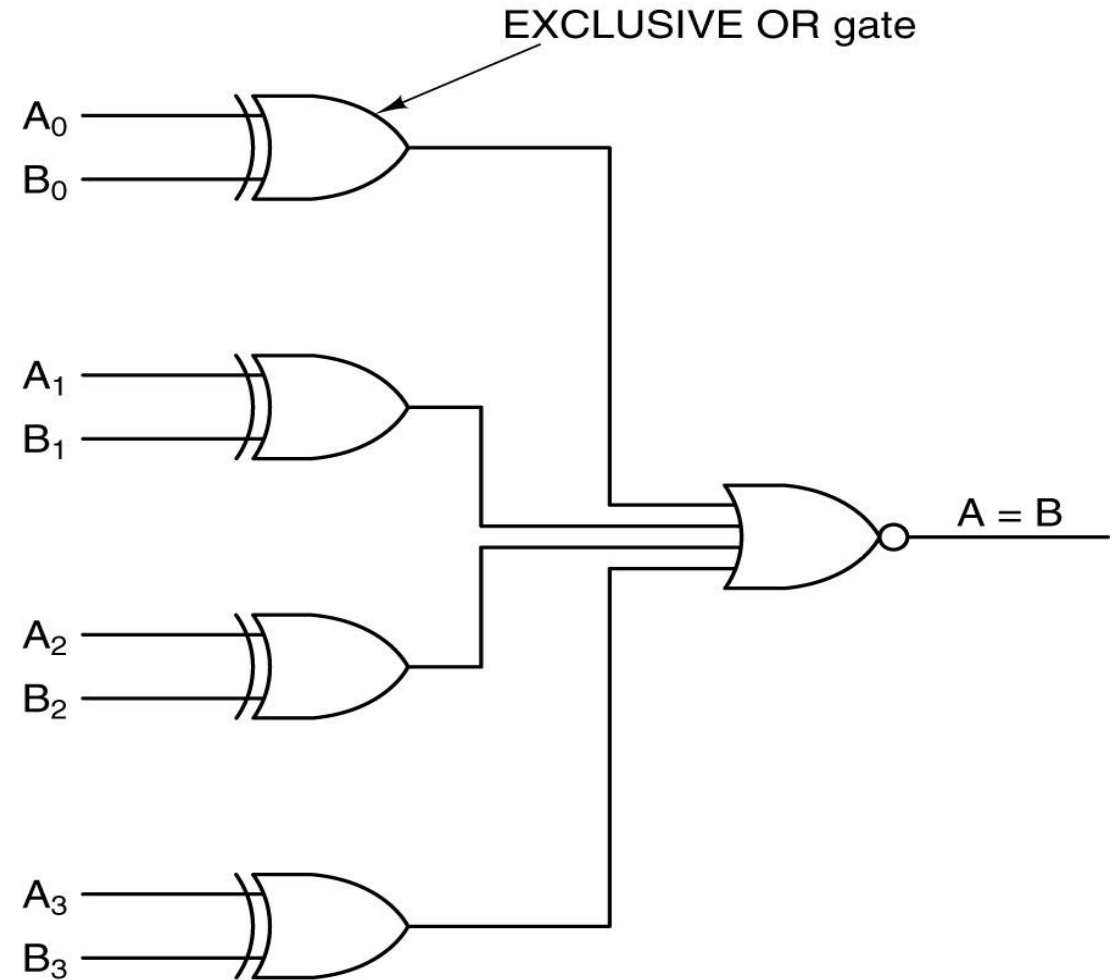
I_1	I_0	O_3	O_2	O_1	O_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



Comparators

A simple 4-bit comparator.

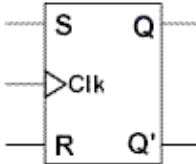
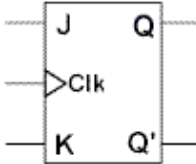
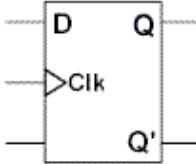
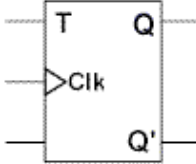
Used to implement comparison operators ($=$, $>$, $<$, $\geq$, $\leq$)



Sıralı Mantık (Sequential Logic)

- Sıralı mantık belleğe sahiptir; devre, önceki giriş setinin sonucunu depolar. Çıkış, geçmişteki girişlerin yanı sıra mevcut girişlere bağlıdır.
- Sıralı mantıktaki temel öge, bir bitlik veri için bellek ögesi olarak işlev gören iki durumlu mandal (latch) veya iki durumlu tetikleme devresidir (flip – flop).

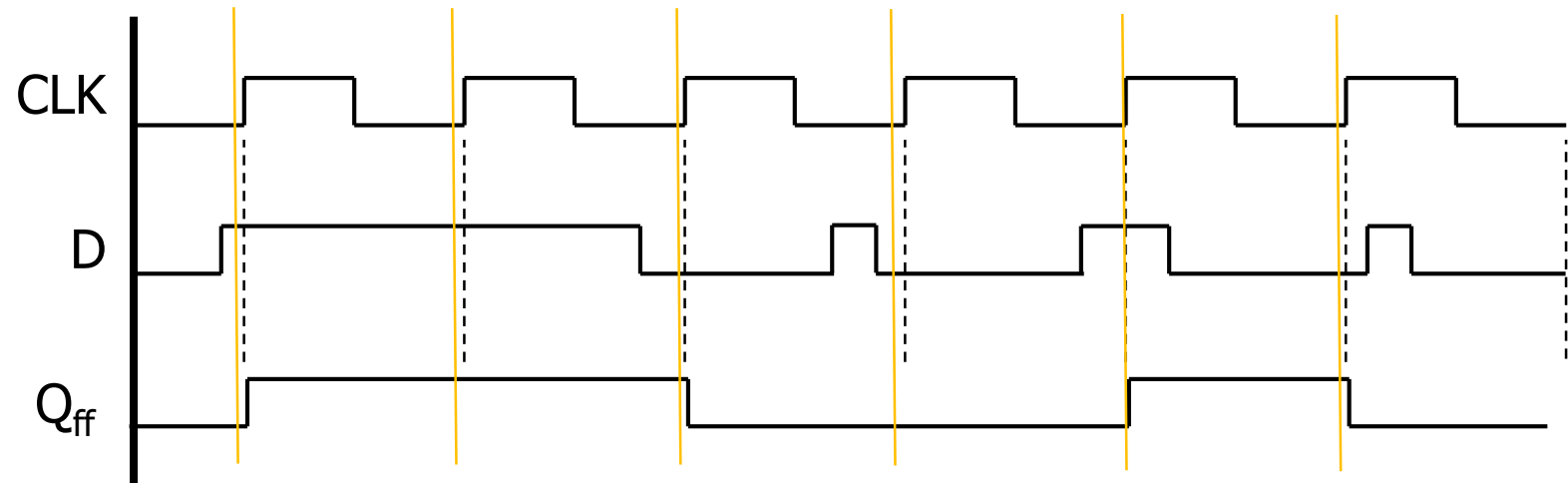
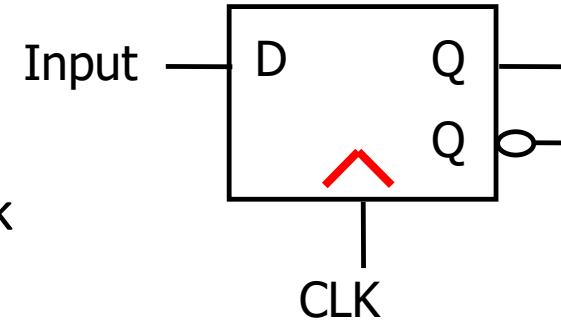
Sequential Logic (Bellek özelliğine sahiptir.)

FLIP-FLOP NAME	FLIP-FLOP SYMBOL	CHARACTERISTIC EQUATION	EXCITATION TABLE																				
SR		$Q_{(next)} = S + R'Q$ $SR = 0$	<table><tr><th>Q</th><th>Q_(next)</th><th>S</th><th>R</th></tr><tr><td>0</td><td>0</td><td>0</td><td>X</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>X</td><td>0</td></tr></table>	Q	Q _(next)	S	R	0	0	0	X	0	1	1	0	1	0	0	1	1	1	X	0
Q	Q _(next)	S	R																				
0	0	0	X																				
0	1	1	0																				
1	0	0	1																				
1	1	X	0																				
JK		$Q_{(next)} = JQ' + K'Q$	<table><tr><th>Q</th><th>Q_(next)</th><th>J</th><th>K</th></tr><tr><td>0</td><td>0</td><td>0</td><td>X</td></tr><tr><td>0</td><td>1</td><td>1</td><td>X</td></tr><tr><td>1</td><td>0</td><td>X</td><td>1</td></tr><tr><td>1</td><td>1</td><td>X</td><td>0</td></tr></table>	Q	Q _(next)	J	K	0	0	0	X	0	1	1	X	1	0	X	1	1	1	X	0
Q	Q _(next)	J	K																				
0	0	0	X																				
0	1	1	X																				
1	0	X	1																				
1	1	X	0																				
D		$Q_{(next)} = D$	<table><tr><th>Q</th><th>Q_(next)</th><th>D</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	Q	Q _(next)	D	0	0	0	0	1	1	1	0	0	1	1	1					
Q	Q _(next)	D																					
0	0	0																					
0	1	1																					
1	0	0																					
1	1	1																					
T		$Q_{(next)} = TQ' + T'Q$	<table><tr><th>Q</th><th>Q_(next)</th><th>T</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	Q	Q _(next)	T	0	0	0	0	1	1	1	0	1	1	1	0					
Q	Q _(next)	T																					
0	0	0																					
0	1	1																					
1	0	1																					
1	1	0																					

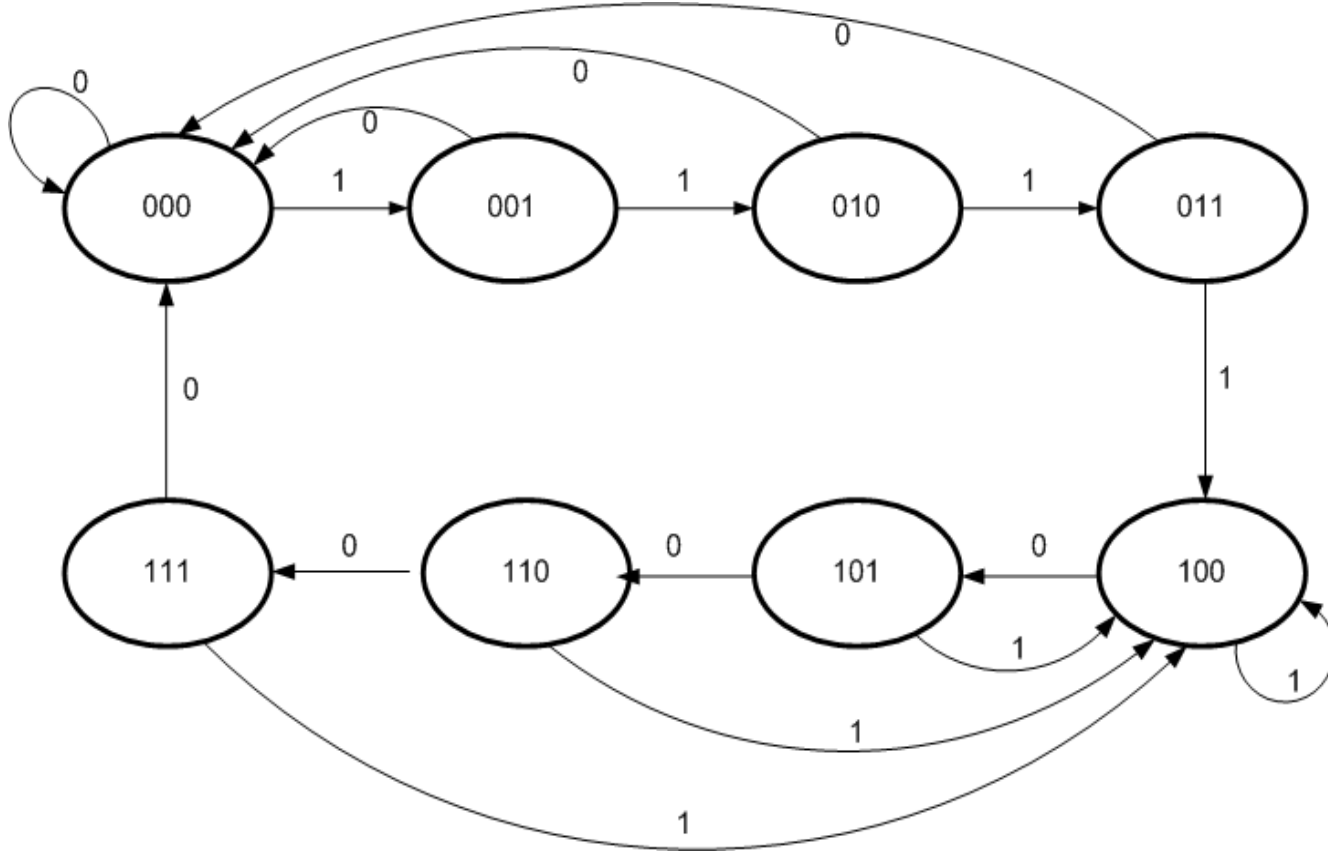
Clok'un yükselen kenarı ile tetiklenir ve çıkış giriş değerini alır. Bir sonraki klok yükselen kenarı gelene kadar durumu değişmez. Buna bellek özelliği denir.

The D flip-flop

- Input sampled at clock edge
 - Rising edge: Input passes to output
 - Otherwise: Flip-flop holds its output
- Flip-flops can be rising-edge triggered or falling-edge triggered
- Clok işaretinin yükselen kenarında çıkış girişe eşit olur ($Q=D$). Clok işaretinin diğer tüm durumlarda çıkış değişmez.
- Şu anki durum (Q) ve bir sonraki durum (D) birlikte göz önüne alınır.



Durum Diyagramı



- Şu anki ve bir sonraki durumlara göre ikili devre sayısı bulunur
- İkili sayısı=3 adettir. Çünkü Durum diyagramında tüm durumlar 0 ile 7 arasında değişmektedir. Toplam durum sayısı=8=2³ dür.
- Şu anki durumlar D-ikili devresini Q çıkışlarında bulunmaktadır. Bir sonraki durum ise D-ikili devresinin D girişlerinde bulunmaktadır.
- Clock'un yükselen kenarı ile D-ikili devresi tetiklendiğinde Q-çıkışları D-girişlerine eşit olur.

Durum Tablosunun oluşturulması ve Karnaugh diyagramı yardımıyla indirgenmesi

Şu anki durum			Giriş	Bir sonraki durum		
Q2	Q1	Q0	C	D2	D1	D0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	0	1	0	1	1
0	1	1	0	0	0	0
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	1	1	0
1	0	1	1	1	0	0
1	1	0	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	0	0	0
1	1	1	1	1	0	0

Q ₂ Q ₁	Q ₀ C			
	00	01	11	10
00				
01			(1)	
11	(1)	(1)	(1)	
10	(1)	(1)	(1)	(1)

D2=

Q ₂ Q ₁	Q ₀ C			
	00	01	11	10
00			(1)	
01		(1)		
11	(1)			
10				(1)

D1=

Q ₂ Q ₁	Q ₀ C			
	00	01	11	10
00		(1)		
01		(1)		
11	(1)			
10	(1)			

D0=

- $D2 = Q_2Q_1' + Q_2Q_0' + Q_1Q_0C$
- $D1 = Q_2Q_1Q_0'C' + Q_2'Q_1Q_0'C + Q_2'Q_1'Q_0C + Q_2Q_1'Q_0C'$
- $D0 = Q_2'Q_0'C' + Q_2Q_0'C$



Data Representation in Computer

Sayısal Sayı Sistemi

- Digital Number systems
 - Binary, Bits (0/1)
 - Byte: 8 bit
 - Hexadecimal: 4 bit
 - Octal: 3 bit
 - Numbers conversion among different systems
- Ascii code (Klavyedeki her bir tuş 8 bit yani 1 byte ile temsil edilir). Klavyedeki A tuşuna basıldığında bellekte ne görülür? 8 bit : (41)h = (0100 0001)b

Digital Number Systems

- **Binary** numbers
 - Digits = {0, 1}
 - $(11010.11)_2 = ?$
 - İndisleme: Noktaddan sola ve sağa doğru ayrı ayrı indislenir.
 - İndisleme: 43210.-1-2
 - Değer yazma= $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$
 $= (26.75)_{10}$

Digital Number Systems

- **Binary** numbers
 - Digits = {0, 1}
 - $(11010)_2 = ?$
 - İndisleme: Noktaddan sola ve sağa doğru ayrı ayrı indislenir.
 - İndisleme: 43210; Üssel isafler toplanırken olanlar 1 ile olmayan 0 ile çarpılır.
 - Değer yazma= $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (26)_d$

Data Representation in Computer

- In all modern computers, all information is represented by using binary values.
- Each storage location (cell - Transistor): has two states
 - low-voltage electrical signal => 0
 - High-voltage electrical signal => 1
 - it can store a binary digit, **bit**
- Eight bits grouped together to form a **byte**
- Two bytes grouped together to form a **word**
 - Word length of a computer, e.g., 32 bits computer, 64 bits computer, 128 bits

Temel Birimler

- Bilgisayar sistemlerinde temel bilgi birimi bit: (1/0)
- Bit: 0/1; bitler elektriksel sinyaller ile temsil edilir. Elektronik devre elemanı Transistor, Data Bus, Bellek gözü içeriği, Registers
- Byte: 8 bitlik veriyi temsil eder. Ya da 1byte'lık bellek gözünü işaret eder. Belleklerde 2'li tabandaki işlemleri temsil eder. Adres Bus; bellek ve bellek gözü seçer.
- Bellek Boyutu: 2^n ile ifade edilir. Burada n: belleğe gelen adres hat sayısıdır. 2 hat gelirse hat üzeri: 00, 01, 10, 11 olur. O halde $2^2=4$ byte
- Bit/sec: Transfer edilecek ya da işlenecek bir saniyedeki veri miktarını temsil eder. 10^n tabandaki üssel işlemler ile gösterilir.
- Qubit: Quantum hesaplamalarda en küçük veriyi temsil eder.
- Elektron: Qubitler elektronlar ile temsil edilir.

İKİLİ (BINARY) SAYI SİSTEMİ

- Türkçesi ikili sayı sistemi anlamına gelen kavramdır. İkili sayılar 2 tabanında yazılarak elde edilir. Dolayısı ile ikili sistemdeki tüm sayılar 1 ve 0'dan ibarettir.
- Bilgisayar Sistemlerinde ikili sayı sistemine odaklanıyoruz (taban 2)
- Niye?
 - Çünkü (bilgisayarı oluşturan) dijital bileşenler (transistor) iki durumdan birinde olabilir: açık veya kapalı
 - Bu iki durumu temsil etmek için 1 ve 0 kullanıyoruz. Bit ismi Binary Digit, ikili sayı sisteminden türetilmiştir.
- Bilgileri ikili olarak temsil etmek için bir yöntem geliştirmek istiyoruz.
 - sayılar (pozitif, negatif, tamsayı, kayan nokta, kesir), karakter dizileri, booleanlar, resimler, sesler, programlama talimatları
 - Klavyeden yazılan rakamlar ve harfler bilgisayarın anlayacağı şekle çevrilirken ikili sayı sistemi devreye girer.
- İşaretsiz tamsayı değerleri için, bunları doğrudan ikili kullanarak saklayabiliriz.
 - taban = 2 olan önceki slayttaki dönüştürme kullanarak birinden diğerine dönüştürüyoruz
- **Bilgisayarlar sadece 0 ve 1 rakamlarından oluşan ikili sayı sistemini kullanırlar.**
- 1 varlığı temsil eder, 0 ise yokluğu temsil eder. 1 gerilim var (5Volt), 0 ise gerilim yok demektir.
- Günlük hayatımızda kullandığımız rakamlar ise onluk tabanda, bir başka isimle decimaldir. Decimal sistemi oluşturan rakamlar bildiğimiz gibi 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 şeklindedir.

İkili Sayı Sistemi

- Sözelimi m , bitlik sistem oluşturulursa 2^m durum elde edilebilir.
- 8 bit yanyana geldiğinde 1 byte elde edilir. $2^8=256$ farklı sembolü temsil eder. 0 dan 255'e kadar sayı sistemi ile gösterilebilir, -128 ... 0 ...127'ye kadar negatif ve pozitif sayılar gösterebilir; ya da karakterleri atayabiliriz.
- Byte'lar yanyana konularak daha fazla karakter ya da sayı sistemi elde edilebilir. Sözelimi 4 byte yana konularak 32 bit elde edilir.

İkili Sinyal- Bit

İkili sinyal (Binary signal), iki durum sinyali: İki durumlu veriler (0/1).

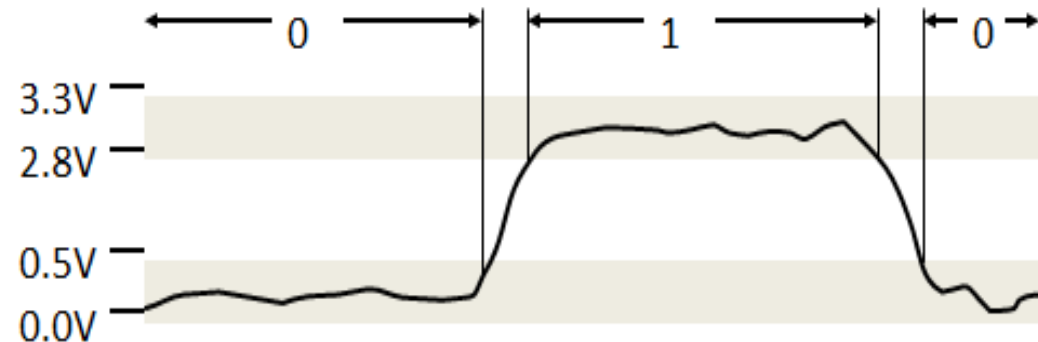
- off & on
- Elektriksel sinyaller ile taşınır, saklanır. low voltage & high voltage; 0v & 5v

Electronic Implementation

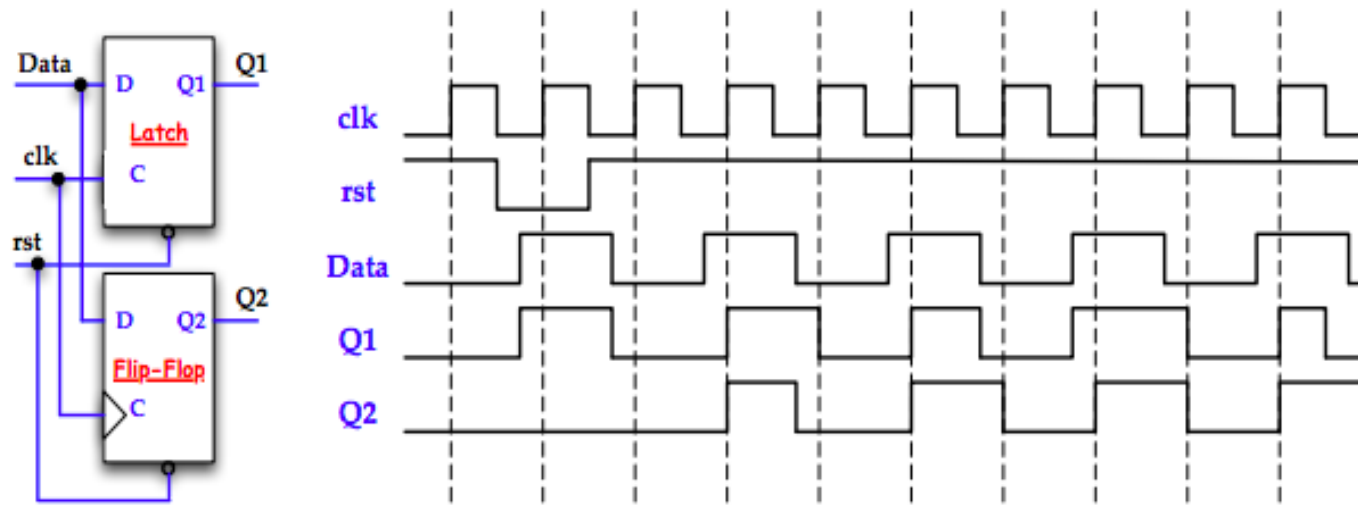
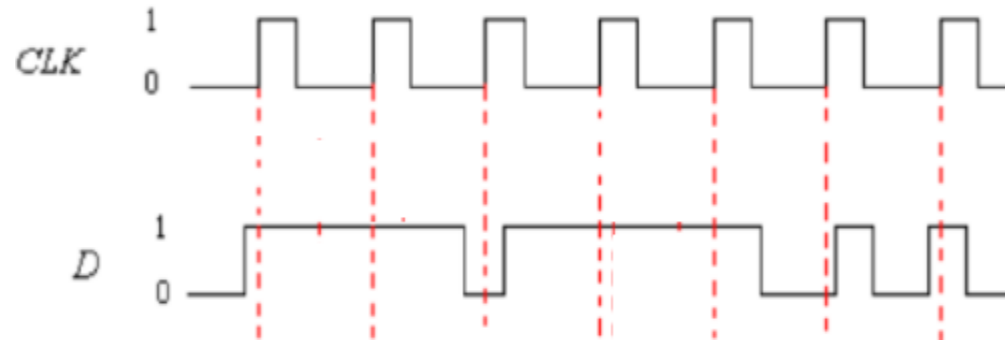
- Easy to store with bistable elements
- Reliably transmitted on noisy and inaccurate wires

Bit: Sadece matematiksel bir kavram değil, aynı zamanda fiziksel dünyada karşılığı vardır.

- Harazmi, Joseph Marie Jacquard , Turing, Shannon
- İkili sayı sistemi 0 veya 1 değerine sahip olur ve başka hiçbir şey olmaz.
- Bir bit, bir bilgisayardaki en küçük bilgi birimidir.



Data and Clock



Bit

- Bit: 0/1; bitler elektriksel sinyaller ile temsil edilir. Data Bus, Bellek gözleri (cells), Registers
- Bitler transistörler ile temsil edilir.
- Byte: 8 bitlik veriyi temsil eder. Ya da 1byte'lık bellek gözünü işaret eder.
- Bit/sec: Transfer edilecek ya da işlenecek bir saniyedeki veri miktarını temsil eder. 10 tabandaki üssel işlemler ile gösterilir.

Bit

- Bit, bilgisayar sistemlerinde kullanılan bilginin temel kavramıdır, temel taşıdır. Bitler, 0 veya 1'lerden oluşur. Birer matematiksel nesnelerdir, ve birer fiziksel durumlara karşılık gelirler.
- *Bir bit(0/1) bilginin temel birimidir.* Bit bilgisayarlar ile bilginin temsil edilmesinde kullanılır. Fiziksel gerçekleştirilmesi ne olursa olsun, bir bit her zaman, bir 0 ya da bir 1 olduğu anlaşılmaktadır. Buna bir benzetme, bir lamba anahtarı ile 0 temsil kapalı konuma ve 1 açık konumda olduğudur.
- Normal veya "klasik" hesaplamada, temel bilgi birimi bittir. Bir bit, 0 veya 1 olmak üzere iki durumdan birinde olabilir.
- Klasik hesaplama cihazlarının yapı taşı, iki durumlu sistem ile temsil edilir (0/1).
- Aslında, aralarında kontrollü geçişlerin olduğu sonlu bir kesikli, kararlı durum kümesine sahip herhangi bir sistem iş görür.

Bit

- Bir bitin alabileceği değerler 0 ya da 1 dir. Durum sayısı= $2^1=2$, Durum: 0,1
- Elimizde iki bitlik bir bilgi var ise kaç durum vardır? $2^2= 4$ durum vardır. Durum: 00, 01, 10,11.
- Üç bitin alabileceği 8 durum vardır: 000, 001, 010, 011, 100, 101, 110, 111. Durum sayısı= $2^3=8$
- n bitin alabileceği durum sayısı= 2^n dir. Durum uzayı olarak tanımlanır.
- Odanın sıcaklığı 25derece, bu bir bilgidir.
- $(25)_d=2^4+2^3+2^0$; İndisleme= 43210
- $(25)_d=(11001)_b$, $(25)_{10}=(11001)_2$
- İkili tabandaki bir sayı onluk tabana nasıl dönüştürülür? Sağdan 0 ile başlanarak indislenir. 1 olan 2 üzeri indis değerleri toplanır. $(11001)_2=2^4+2^3+2^0=25$.
- $(11001)_b$: bu bilgisayarda saklanan bir bilgidir. İkili say sisteminde bir veriyi temsil eder.
- Bilgisayarda tüm fiziksel sistemler ve sinyaller bitler ile temsil edilirler.

Example.....

- If a family has 3 children, find the probability that all 3 children are girls.
- ing for all 3 girls.....

- Sample Space:

BBB

BBG

BGB

GBB

→ **GGG**

GGB

GBG

BGG

$$P(\text{All 3 Girls}) = \frac{1}{8}$$

(2³) 2:kişi sayısı, 3: dizilenme sayısı (Çocuk sayısı)

Bit (0/1)

- BIT is a unit of information equivalent to the result of a choice between only 2 possible alternatives in the binary number system.
- Bitler 0 veya 1 lerden oluşur. Birer matematiksel nesnelerdir ve birer fiziksel durumlara karşılık gelirler.
- Bir sisteme sorulan, doğru ve yanlış olarak verilen iki yanıtta biri olan bilgidir.
- Bit, bilgisayar sistemlerinde kullanılan bilginin temel kavramıdır, temel taşıdır.
- Bit, bilgisayarlar ile bilginin temsil edilmesinde kullanılır.
- Fiziksel gerçekleştirilmesi ne olursa olsun, bir bit her zaman, 0 ya da 1 olduğu anlaşılmaktadır.
- Buna bir benzetme, bir lamba anahtarı ile 0 temsil kapalı konuma ve 1 açık konumda olduğudur.
- İki tür bit vardır, klasik bir cbit, bit (0,1) ve quantum bit, qubit ($|0\rangle$, $|0\rangle$).

Bit/San

Bit: Dijital elektronikte ve binary sayı sisteminde sadece 0 ve 1 değerleri vardır. Tüm işlemler bu iki değer üzerinden yapılır. 0 ya da 1 bilgisinin her birine bit denir. Bit→0/1 den oluşan bilgi

- Bits are the units used to describe an amount of data in a network
 - 1 kilobit (Kbit) = 1×10^3 bits = 1,000 bits
 - 1 megabit (Mbit) = 1×10^6 bits = 1,000,000 bits
 - 1 gigabit (Gbit) = 1×10^9 bits = 1,000,000,000 bits

Bit/Saniye: Bit/sec→1 sn. ye de bir noktadan diğer noktaya iletilen bilgi. Ya da bir saniyede işlen bit yani bilgi miktarıdır. **BPS (Bit Per Second);** Saniyede iletilen bit sayısına bps denir.

- Seconds are the units used to measure time
 - 1 millisecond (msec) = 1×10^{-3} seconds = 0.001 seconds
 - 1 microsecond (msec) = 1×10^{-6} seconds = 0.000001 seconds
 - 1 nanosecond (nsec) = 1×10^{-9} seconds = 0.000000001 seconds
- Bits per second are the units used to measure channel capacity/bandwidth and throughput
 - bit per second (bps)
 - kilobits per second (Kbps)
 - megabits per second (Mbps)

Byte

Byte: Bellek boyutunu verir. Elektronik ve bilgisayar bilimlerinde genellikle 8 bitlik dizilim boyunca 1 veya 0 değerlerini bünyesine alan ve kaydedilen bilgilerin türünden bağımsız bir bellek ölçüm birimidir.

Kilobyte	Kb	2^{10} Byte
Mega Byte	Mb	2^{20} Byte
Giga Byte	Gb	2^{30} Byte
Tera Byte	Tb	2^{40} Byte
Peta Byte	Pb	2^{50} Byte
Exa Byte	Eb	2^{60} Byte
Zetta Byte	Zb	2^{70} Byte
Yotta Byte	Yb	2^{80} Byte

Bit terimi belleğin 8 bitlik bir değerini işaretleyen ya da tanımlayan en küçük birimi olarak tanımlanmıştır. Byte→bellekte 8bitlik adres gözü ya da bellek boyutu tanımlar tanımlanır.

Soru: 1 Terabyte bellek gözünde kaç bit vardır. Bir adet bit 3 adet transistor devresi ile yapılıyorsa kaç adet transistör kullanılmıştır?

1Terabyet= 2^{40} byte; 1 Byte=8 bit= 2^3 bit ise 1Tbyte= $(2^{40}) * (2^3)$ bit= 2^{43} bit

$2^{43} * 3$ transistör kullanılmıştır.

Byte

BYTE is a sequence of 8 bits (enough to represent one character of alphanumeric data) processed as a single unit for information.

Byte: Bellek boyutunu hesaplanırken ya da veri tipini tanımlada kullanılır. Elektronik ve bilgisayar bilimlerinde genellikle 8 bitlik dizilim boyunca 1 veya 0 değerlerini bünyesine alan ve kaydedilen bilgilerin türünden bağımsız bir bellek ölçüm birimidir. Belleğin 8 bitlik bir değerini işaretleyen ya da tanımlayan en küçük birimi olarak tanımlanmıştır.

- Kilobyte (K) $2^{10} = 1,024$ bytes
- Megabyte (M) $2^{20} = 1,048,576$ bytes
- Gigabyte (G) $2^{30} = 1,073,741,824$ bytes
- Terabytes (T) $2^{40} = 1,099,511,627,776$ bytes
- Petabytes (P) $2^{50} = 1,125,899,906,842,624$ bytes
- Exabytes (E) $2^{60} = 1,152,921,504,606,846,976$ bytes
- Zettabytes (Z) $2^{70} = 1,180,591,620,717,411,303,424$ bytes
- Yottabytes (Y) $2^{80} = 1,208,925,819,614,629,174,706,176$ bytes

Bits and Bytes are Slightly Different

“Kilo” or “Mega” have slightly different values when used with bits per second or with bytes.

- When Referring to Bytes (as in computer memory)
 - Kilobyte (KB) $2^{10} = 1,024$ bytes
 - Megabyte (MB) $2^{20} = 1,048,576$ bytes
 - Gigabyte (GB) $2^{30} = 1,073,741,824$ bytes
 - Terabyte (TB) $2^{40} = 1,099,511,627,776$ bytes
- When Referring to Bits Per Second (as in transmission rates): Veri işleme ya da veri transfer hızıdır.
 - Kilobit per second (Kbps) = 1000 bps (thousand)
 - Megabit per second (Mbps) = 1,000,000 bps (million)
 - Gigabit per second (Gbps) = 1,000,000,000 bps (billion)
 - Terabit per second (Tbps) = 1,000,000,000,000 bps (trillion)

Byte, Baud Rate

Byte: Bellek boyutunu verir. Elektronik ve bilgisayar bilimlerinde genellikle 8 bitlik dizilim boyunca 1 veya 0 değerlerini bünyesine alan ve kaydedilen bilgilerin türünden bağımsız bir bellek ölçüm birimidir.

Bit terimi belleğin 8 bitlik bir değerini işaretleyen ya da tanımlayan en küçük birimi olarak tanımlanmıştır. Daha sonra, 1956'da, 6 Bite'tan 8 Bite geliştirilmiştir. Bite, bit ile karıştırılmaması için daha sonra Byte'a çevrilmiştir. Diğer bir kelime açıklamasına göre de, Byte, "by eight" in (Türkçe'de *sekiz kez veya sekiz ile*) kısaltılmış halidir. Byte→bellekte 8bitlik adres gözü ya da bellek boyutu tanımlar tanımlanır.

Baud Rate: Data iletiminde modülatör çıkışında bir saniyede meydana gelen sembol (baud) değişikliğine baud hızı denir. Baud hızı baud/sn ile gösterilir. Baud hızı sinyalin anahtarlama hızını gösterir.

Örnek: Bir veri iletim hattının iletim hızı 4800 baud/sn olsun. Bu iletim her baud 4 bitle kodlanmış bilgi içeriyorsa bps olarak hızımız $4800 \times 4 = 19200$ bps olur. Baud Rate'i kullanmadaki amaç band genişliğini daha verimli kullanmak.

Örnekler

12Gbyte kaç Byte'dır?

- 12 yerine 16 alınır. (2^n olmalıdır.)
- $16\text{GByte} = 2^4 \cdot 2^{30} = 2^{34}$ Byte

16Gbyte kaç KByte'dır?

- $16\text{GByte} = 2^4 \cdot 2^{30} = 2^{34} / 2^{10} = 2^{24}$ KByte
- $16\text{GByte} = 2^4 \cdot 2^{30} = 2^{34} / 2^{20} = 2^{14}$ MByte
- $16\text{GByte} = 2^4 \cdot 2^{30} = 2^{34} / 2^{30} = 2^4$ GByte

128MByte kaç Byte'dır?

- $128\text{MByte} = 2^7 \cdot 2^{20} = 2^{27}$ Byte

256MByte kaç Byte'dır?

- $256\text{MByte} = 2^8 \cdot 2^{20} = 2^{28}$ Byte

4KByte kaç Byte'dır?

- $4\text{KByte} = 2^2 \cdot 2^{10} = 2^{12}$ Byte

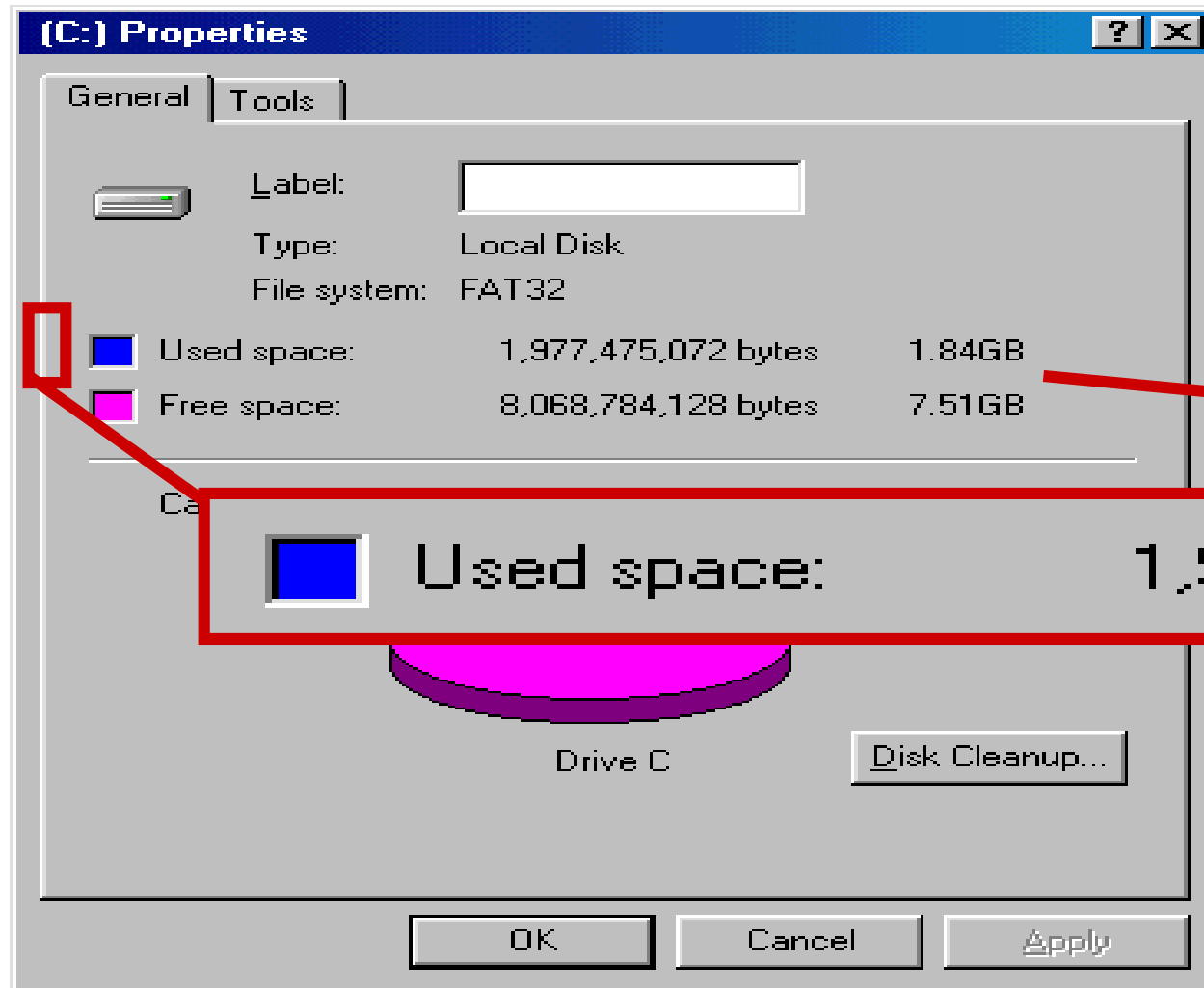
32Tbyte kaç byte'dır?

- $32\text{TByte} = 2^5 \cdot 2^{40} = 2^{45}$ Byte
- $32\text{TByte} = 2^5 \cdot 2^{40} = 2^{45} / 2^{10} = 2^{35}$ KByte
- $32\text{TByte} = 2^5 \cdot 2^{40} = 2^{45} / 2^{20} = 2^{25}$ MByte
- $32\text{TByte} = 2^5 \cdot 2^{40} = 2^{45} / 2^{30} = 2^{15}$ GByte
- $32\text{TByte} = 2^5 \cdot 2^{40} = 2^{45} / 2^{40} = 2^5$ TByte

8PByte kaç Byte'dır?

- $8\text{PByte} = 2^3 \cdot 2^{50} = 2^{53}$ Byte

Example



In the lab...

1. Double click on My Computer
2. Right click on C:
3. Click on Properties

Used space: 1,977,475,072 bytes 1.84GB

$$/ 2^{30} =$$



Data Representations

16 bitten yani 2 bayttan oluşan sayılara 1 word adı verilir.

1100101100011000

8 bitten oluşan sayılara 1 byte adı verilir.

Bir binary sayının en solundaki bit MSB, en sağındaki bit de LSB olarak isimlendirilir.

1100101100011000

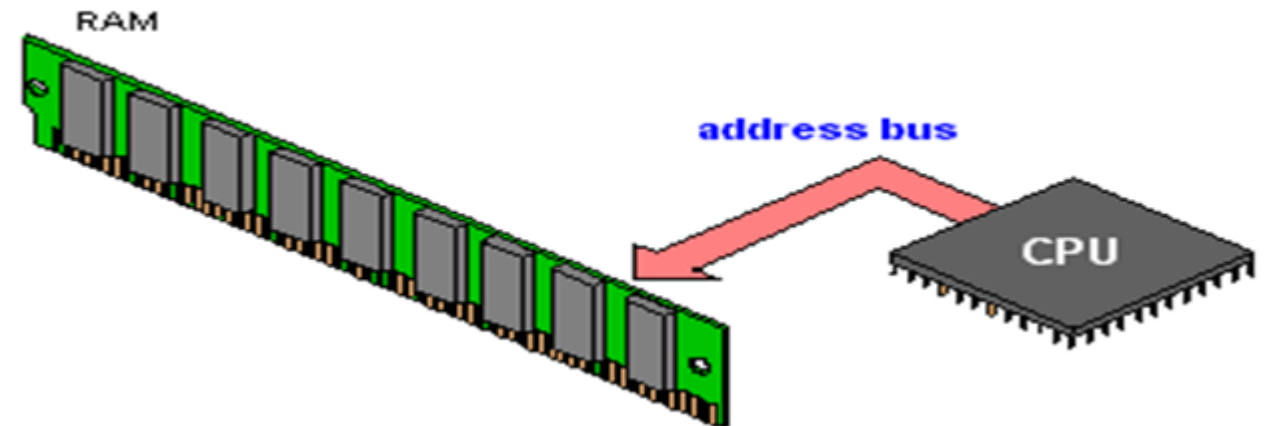
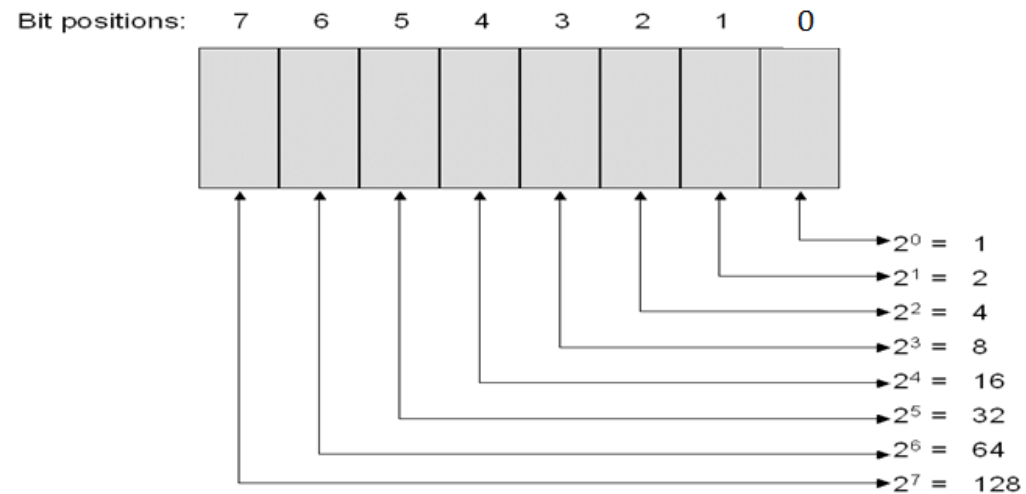
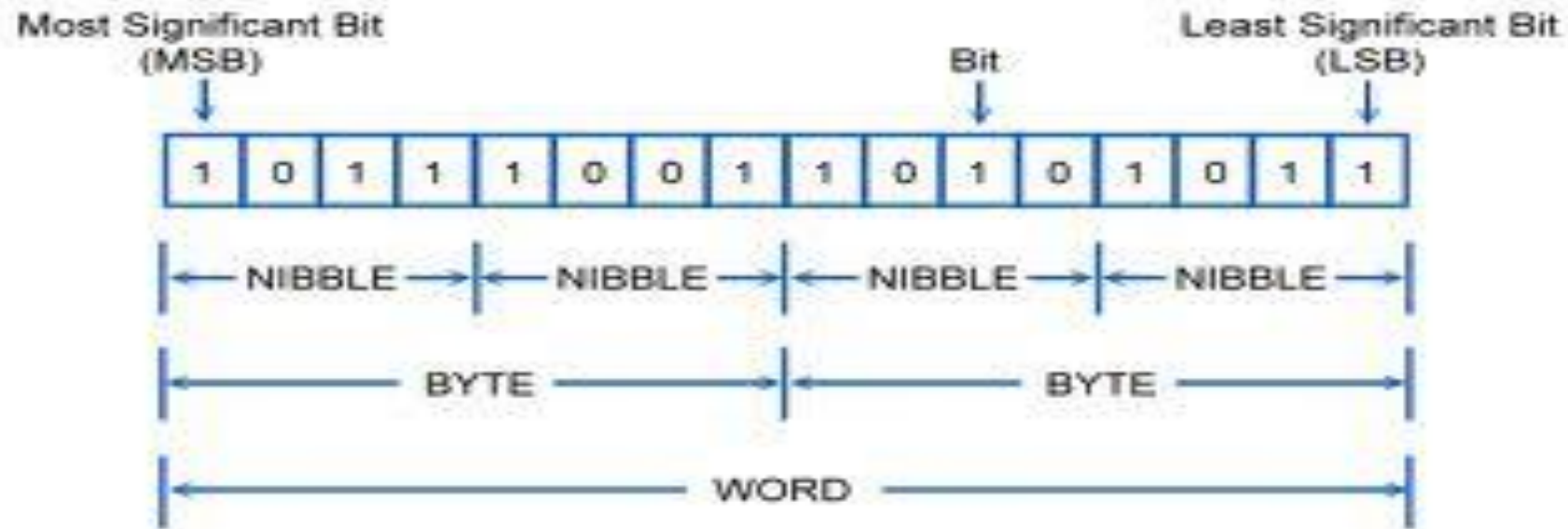
MSB

Most Significant Bit

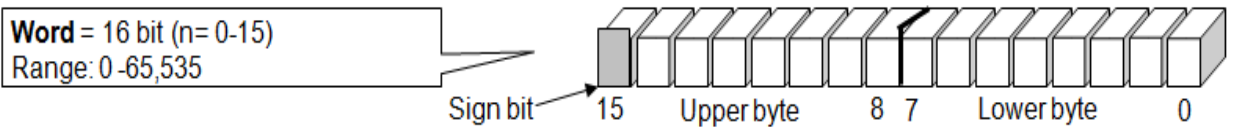
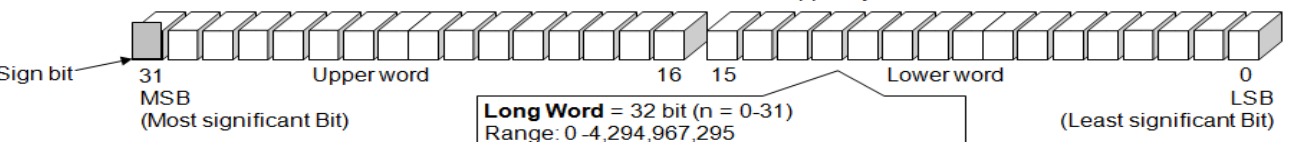
LSB

Least Significant Bit

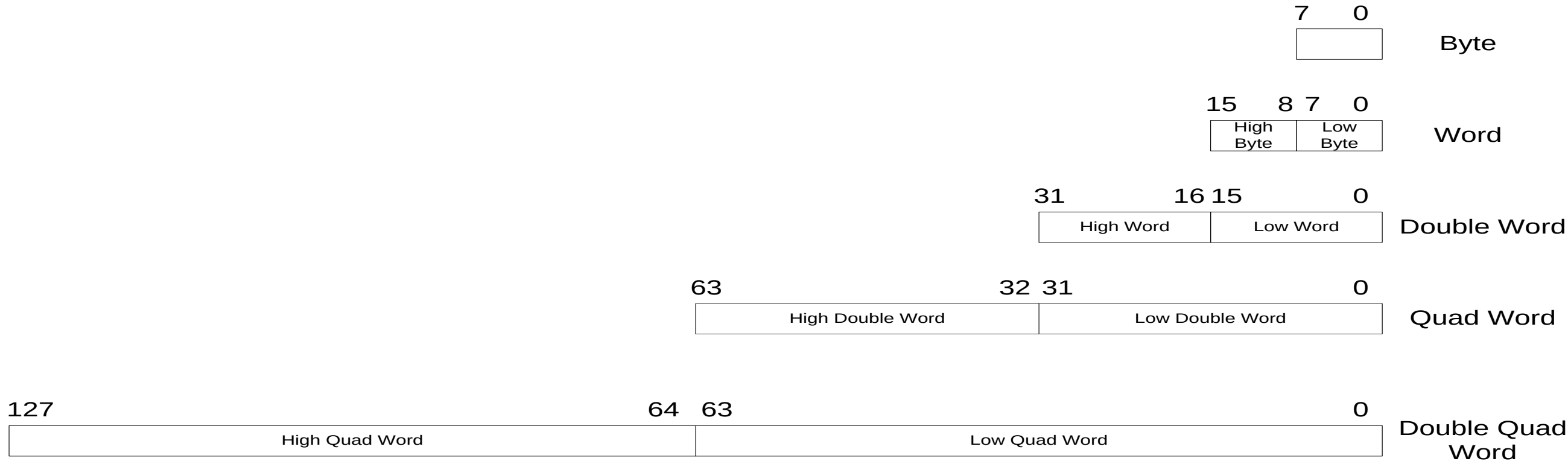
Bits and Bytes



DATA SIZE

Nibble	4 bit	Nibble = 4 bit (n = 0-3) Range: 0 -15 
Byte	8 bit	Byte = 8 bit (n = 0-7) Range: 0 -255 
Word	16 bit	Word = 16 bit (n = 0-15) Range: 0 -65,535 
Long word	32 bit	Long Word = 32 bit (n = 0-31) Range: 0 -4,294,967,295 

Data Types - size



Always take care of the type of data an instruction accesses!!!!

Data Representations

- Sizes (in Bytes)

– Data Type	Compaq Alpha	Typical 32-bit	Intel IA32
• int	4	4	4
• long int	8	4	4
• char	1	1	1
• short	2	2	2
• float	4	4	4
• double	8	8	8
• long double	8	8	10/12
• char *	8	4	4

Binary sisteminde veri tipleri

- Byte: 8 bit
- Word: 16 bit
- Double Word: 32 bit
- Q Word: 64 bit
- Veri boyutu 10^9 bit/sec (Veri işleme ya da veri transfer hızı)



Binary Manuplation

Bilgisayarda İşlemler

- Aritmetik: +, (-, *, /)
- Mantıksal: VE, VEYA, NOT, ...
- Karşılaştırma: <, >, ≥, ≤, ==, ≠, ...
- Mikroişlemci ALU (Aritmetik Logic Unit) sadece toplama işlemi vardır ve çarpma ve bölme işlemlerini ise öteleme ile yapar.
- ROL: Rotate operand1 left (Çarpma)

Addition (binary)

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} {}^1 1 \\ + 1 \\ \hline 10 \end{array}$$

$$\begin{array}{r} 1 \\ 1 \\ 1 \\ + \\ \hline 11 \end{array}$$

İkili Sayılarda Toplama

Binary(İkilik) sayı sistemindeki temel toplama kuralları;

0+0	= 0	→ Elde 0	Toplam 0
0+1	= 1	→ Elde 0	Toplam 1
1+0	= 1	→ Elde 0	Toplam 1
1+1	= 10	→ Elde 1	Toplam 0
1+1+1	= 11	→ Elde 1	Toplam 1

Numerical examples for addition follow:

$$\begin{array}{r} + 6 \quad 00000110 \\ +13 \quad 00001101 \\ \hline +19 \quad 00010011 \end{array}$$

$$\begin{array}{r} + 6 \quad 00000110 \\ -13 \quad 11110011 \\ \hline - 7 \quad 11111001 \end{array}$$

$$\begin{array}{r} - 6 \quad 11111010 \\ +13 \quad 00001101 \\ \hline + 7 \quad 00000111 \end{array}$$

$$\begin{array}{r} - 6 \quad 11111010 \\ -13 \quad 11110011 \\ \hline -19 \quad 11101101 \end{array}$$

Ondalık toplama işlemini düşünün: $3+1=4$

İkili sayı sistemine dönüştürülür.

2 üssü toplamlardan ondalık sayı sistemi eldeilir: $4=2^2$

İndisleme: 2 1 0

Dönüştürme: (100)

(3)d=(011)b

(1)d=(001)b

(011)b+(001)b=(100)b

Binary addition

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$$

- Represent sum of binary numbers as a binary number

decimal addition

$$1+1 = 2$$

$$1+1+1 = 3$$

binary addition

$$1+1 = 10$$

$$1+1+1 = 10+1 = 11$$

Adding binary numbers

Binary addition

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$$

- Represent sum of binary numbers as a binary number

decimal addition

$$1+1 = 2$$

$$1+1+1 = 3$$

binary addition

$$1+1 = 10$$

$$1+1+1 = 10+1 = 11$$

Adding binary numbers

$$\begin{array}{r} 101 \\ + 10 \\ \hline 111 \end{array}$$

$$\begin{array}{r} 11 \leftarrow \text{carry} \\ 101 \\ + 11 \\ \hline 1000 \end{array}$$

$$\begin{array}{r} 11 \leftarrow \text{carry} \\ 111 \\ + 110 \\ \hline 1101 \end{array}$$

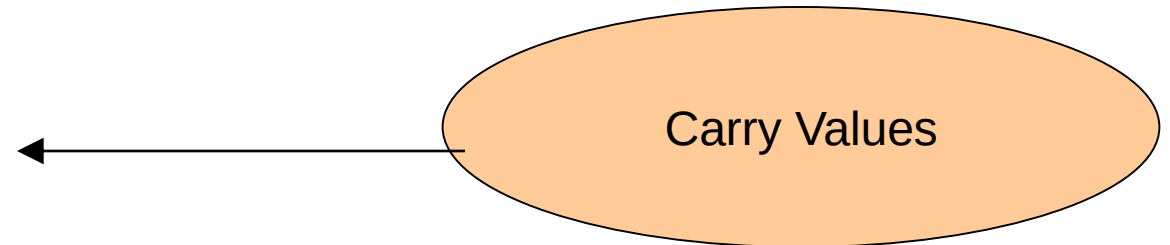
$$\begin{array}{r} 1 \quad 11 \leftarrow \text{carry} \\ 10101010111 \\ + 110000110 \\ \hline 11011011101 \end{array}$$

Arithmetic in Binary

Remember that there are only 2 digit symbols in binary, 0 and 1

1 + 1 is 0 with a carry

```
  1 0 1 1 1 1 1
    1 0 1 0 1 1 1
+  1 0 0 1 0 1 1
-----
  1 0 1 0 0 0 1 0
```



Binary sayılarda toplama-çıkarma işlemleri

Aşağıda verilen toplama işlemlerini gerçekleştirin.

$$\begin{array}{r} \text{a- } (11)_2 \\ + (11)_2 \\ \hline (110)_2 \end{array}$$

$$\begin{array}{r} 3 \\ + 3 \\ \hline 6 \end{array}$$

$$\begin{array}{r} \text{b- } (100)_2 \\ + (11)_2 \\ \hline (111)_2 \end{array}$$

$$\begin{array}{r} \text{c- } (111)_2 \\ + (11)_2 \\ \hline (1010)_2 \end{array}$$

$$\begin{array}{r} \text{d- } (0110)_2 \\ + (1111)_2 \\ \hline (10101)_2 \end{array}$$

$$\begin{array}{r} \text{c- } (11101)_2 \\ (1001)_2 \\ + (111)_2 \\ \hline (101101)_2 \end{array}$$

Aşağıda verilen çıkarma işlemlerini gerçekleştirin.

$$\begin{array}{r} \text{a- } (11)_2 \\ - (10)_2 \\ \hline (01)_2 \end{array}$$

$$\begin{array}{r} \text{b- } (100)_2 \\ - (011)_2 \\ \hline (001)_2 \end{array}$$

$$\begin{array}{r} \text{c- } (101)_2 \\ - (011)_2 \\ \hline (010)_2 \end{array}$$

$$\begin{array}{r} \text{d- } (1010)_2 \\ - (0011)_2 \\ \hline (0111)_2 \end{array}$$

ROL: Rotate operand1 left (Çarpma)

- **ROL Operand1, Operand2**
- Algorithm: Tüm bitleri sola kaydırılır, giden bit CF'ye ayarlanır ve aynı bit en sağdaki konuma eklenir.
- Not: Operand1'i sola döndürülür. Döndürme sayısı Operand2 tarafından belirlenir.

Operand:

- memory, immediate
- REG, immediate
- memory, CL
- REG, CL

- **Example:**

MOV AL, 1Ch ; AL = 00011100b

ROL AL, 1 ; AL = 00111000b, CF=0.

RET

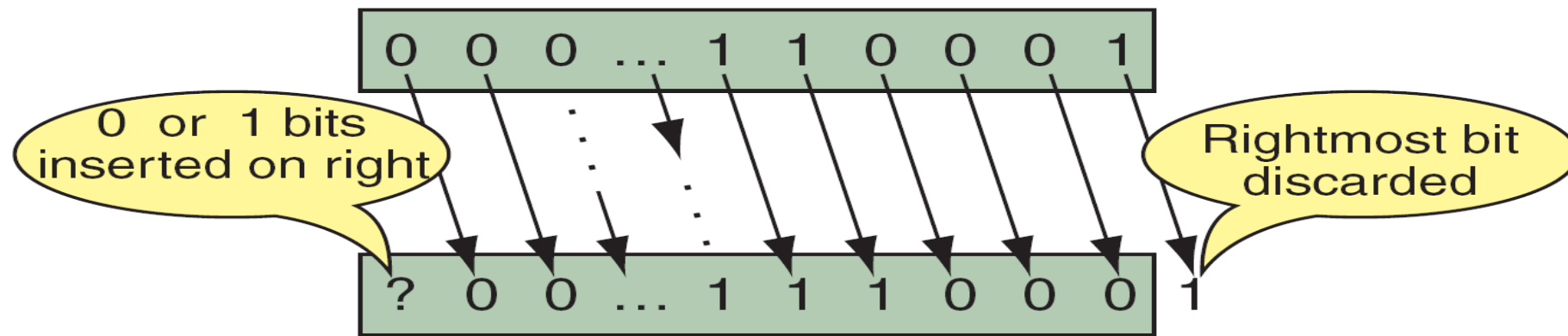


FIGURE Shift-right Operation

$2^{\text{shift value}}$	Divides by	Shift Operator
1	2	$\gg 1$
2	4	$\gg 2$
3	8	$\gg 3$
4	16	$\gg 4$
...	...	...
n	2^n	$\gg n$

Table

Divide by Shift

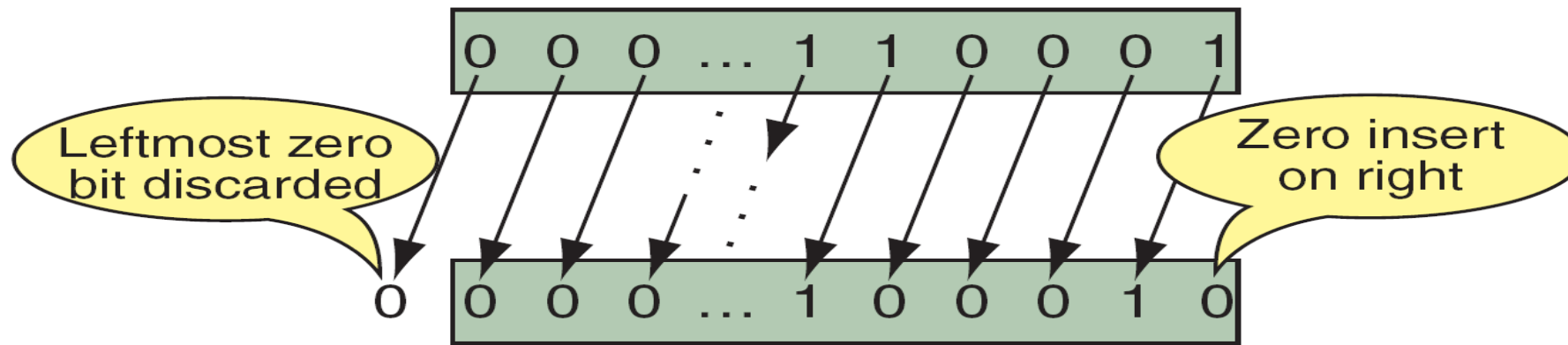


FIGURE Shift-left Operation

$2^{\text{shift value}}$	Multiplies by	Shift Operator
1	2	$\ll 1$
2	4	$\ll 2$
3	8	$\ll 3$
4	16	$\ll 4$
...	...	...
n	2^n	$\ll n$

Table

Multiply by Shift



Binary Conversion

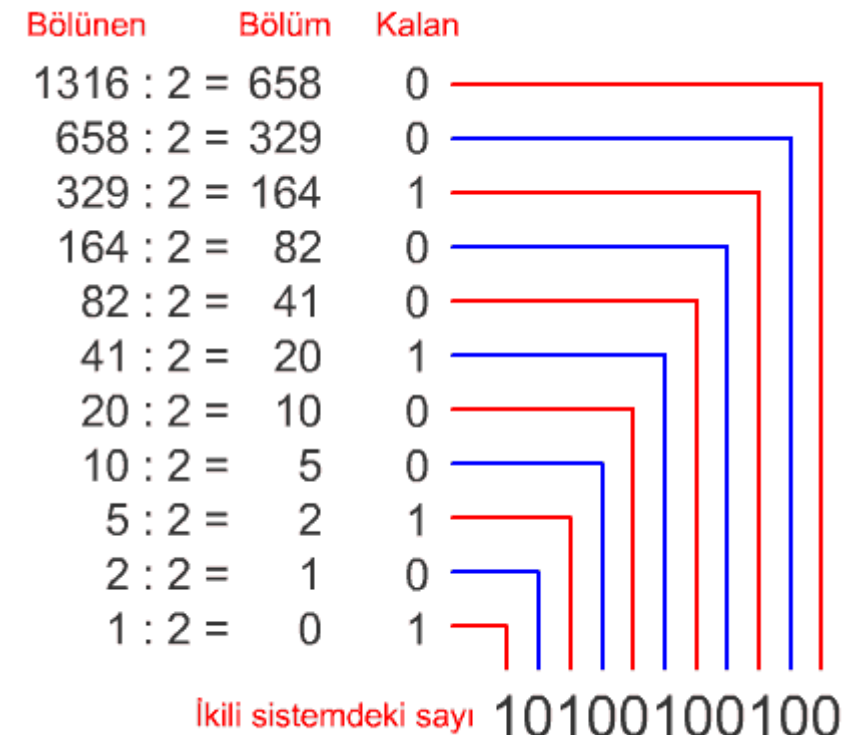
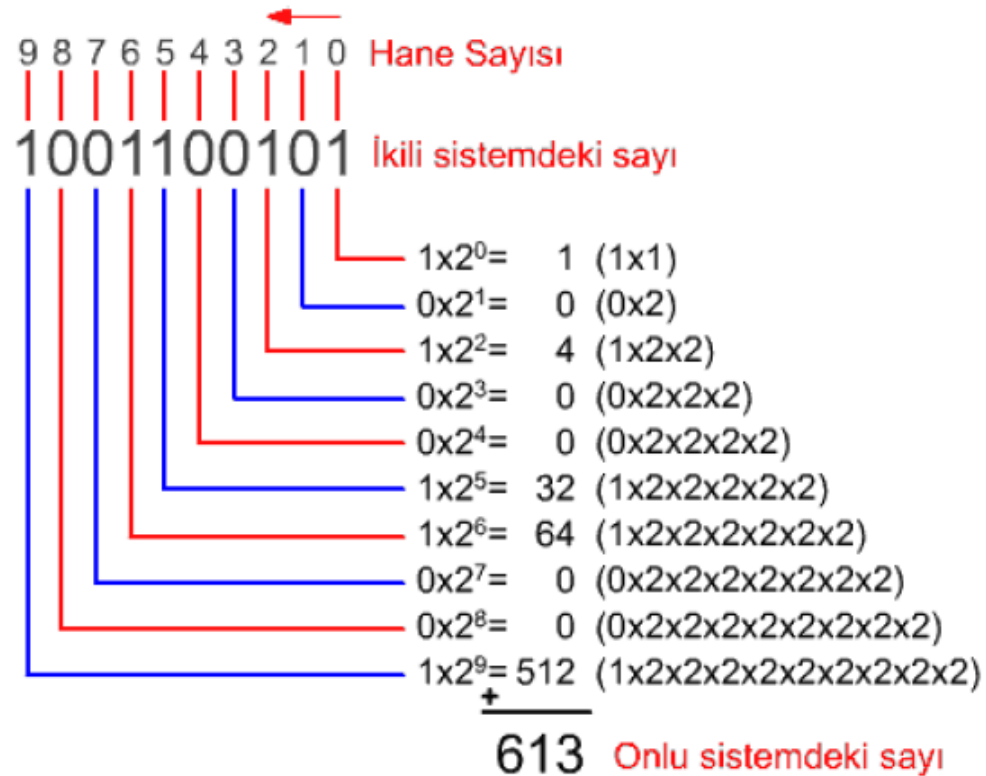
Binary Sayıların Decimal Sayılara Dönüştürülmesi:

- $(100011)_2 = (?)_{10}$
indis: 5,4,3,2,1,0
1 olanların indis karşılığı = 2^{indis}
 $2^5 + 2^1 + 2^0 = 32 + 2 + 1 = (35)_{10} = (23)_h$
- Ondalıklı Binary Sayıların Decimal Sayılara Dönüştürülmesi: $(111,101)_2 = 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-3} = 4 + 2 + 1 + 1/2 + 1/8 = 7,625$
- Decimal Sayıların Binary Sayılara Çevrilmesi: $(172)_{10} = (128 + 32 + 8 + 4)_{10} = (2^7 + 2^5 + 2^3 + 2^2)_{10} = (1010\ 1100)_2 = (AC)_{16}$
- Ondalıklı Decimal Sayıların Binary Sayılara Dönüştürülmesi
- $(10,75)_{10} = ?$ $(10)_{10} = (2^3 + 2^1)_{10} = (1010)_2$, $2^{-1} = 1/2 = 0,5$ $2^{-2} = 1/4 = 0,25$, $(10,75)_{10} = (1010,11)_2$

Numbers

- A 16-bit binary number
- 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
- 0 0 0 0 0 0 0 1 1 0 0 1 1 1 1 1
- $= 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
- $= 256 + 128 + 16 + 8 + 4 + 2 + 1$
- $= 415$

Binary sayı sisteminden decimal'a çevirme



Binary to Decimal

Decimal	→ → conversion → →	Binary
0 =	0×2^0	= 0
1 =	1×2^0	= 1
2 =	$1 \times 2^1 + 0 \times 2^0$	= 10
3 =	$1 \times 2^1 + 1 \times 2^0$	= 11
4 =	$1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$	= 100
5 =	$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$	= 101
6 =	$1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$	= 110
7 =	$1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$	= 111
8 =	$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$	= 1000

Binary

- Binary is exactly the same, only instead of ten digits/states (0 to 9) we have just two, so the base becomes 2:

Bilgisayar sistemlerinde 2 üzeri indeksleme 0,1,2,3,.. Biçimindedir. İndeksleme sağdan başlar.

$$1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$10110_b = 22_d$

Most Significant Bit (MSB)

Least Significant Bit (LSB)

Binary'den decimal'a çevirme örnekleri

- Örnek:

- $(1010)_2 = (?)_{10}$

- $(1010)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

- $(1010)_2 = 8 + 0 + 2 + 0$

- $(1010)_2 = 10$

- Örnek:

- $(11001)_2 = (?)_{10}$

- $(11001)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

- $(11001)_2 = 16 + 8 + 0 + 0 + 1$

- $(11001)_2 = 25$

Soru

- $(1011\ 0010)_b = (?)_d$
- Bir bit sağdan sola öteleyin
- Bir bit soldan sağa öteleyin
- Yorumlayın

Numbers

- If we let the most significant bit (MSB) signify positive or negative numbers (1 for negative; 0 for positive) Word olara tanımlayalım (16 bit)
- Then
- $+9 = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1$
- $-9 = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1$
- $+9 + (-9) = 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0$
- The result is NOT zero. This is a problem!

Numbers

- So we need a different representation for negative numbers
- Called 2s-complement
- Take the 1s-complement of the positive number:
- (+9)₁₀: 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
- 1'ler 0; 0'lar 1 yapılır:
- 1 1 1 1 1 1 1 1 1 1 0 1 1 0
- Note that the addition of the 1s-complement and the original number is not zero. +1 eklenir.
- To get the 2s-complement, add 1
- 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 + 1
- ---
- 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
- Now do the addition:
- 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 +
- 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
- ---
- 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Numbers

- Another example:
- $+1 + (-1)$
- 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 +
- 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
- ---
- 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Örnek: (16bit)

- (-27)_d
- (+27)_d=16+8+2+1=2⁴+2³+2¹+2⁰
- indis: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
- 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1
- (+27)_d=0000 0000 0001 1011 =(001B)_h
- 1111 1111 1110 0100 (1'ler 0; 0'lar 1 yapıldı)
- 1111 1111 1110 0100
- 1
- 1111 1111 1110 0101=(-27)_d

Binary Sayıların Decimal Sayılara Dönüştürülmesi:

- $(100011)_2 = (?)_{10}$
indis: 5,4,3,2,1,0
1 olanların indis karşılığı = 2^{indis}
 $2^5 + 2^1 + 2^0 = 32 + 2 + 1 = (35)_{10} = (23)_h$
- Ondalık Binary Sayıların Decimal Sayılara Dönüştürülmesi: $(111,101)_2 = 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-3} = 4 + 2 + 1 + 1/2 + 1/8 = 7,625$
- Decimal Sayıların Binary Sayılara Çevrilmesi: $(172)_{10} = (128 + 32 + 8 + 4)_{10} = (2^7 + 2^5 + 2^3 + 2^2)_{10} = (1010\ 1100)_2 = (AC)_{16}$
- Ondalık Decimal Sayıların Binary Sayılara Dönüştürülmesi
- $(10,75)_{10} = ?$ $(10)_{10} = (2^3 + 2^1)_{10} = (1010)_2$, $2^{-1} = 1/2 = 0,5$ $2^{-2} = 1/4 = 0,25$, $(10,75)_{10} = (1010,11)_2$

Örnek

$(3)_d = 2^1 + 2^0 = 2 + 1$
 $(1 \ 1)_b$

$(215)_d = (b)_?$ maksimum 2ⁿ
 $= 2^7 + 2^6 + 2^4 + 2^2 + 2^1 + 2^0$ n=0,1,2,...
 128
 $\begin{array}{r} 87 \\ 64 \\ \hline 23 \\ 16 \\ \hline 7 \\ 4 \\ \hline 3 \\ 2 \\ \hline 1 \end{array}$
 $\begin{array}{cccccccc} 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}$
 $(11010111)_b$
 $= 2^7 + 2^6 + 2^4 + 2^3 + 2^1 + 2^0$
 $\begin{array}{ccc} 1 & 1 & 1 \\ +1 & +1 & +1 \\ \hline 10 & 10 & 11 \\ +1 & +1 & +10 \\ \hline 11 & 11 & 10 \end{array}$
 $(101)_b = 2^2 + 2^0 = 5$

$(22)_d = 2^4 + 2^2 + 2^1$ $2 \dots 0000010110$
 $\begin{array}{r} 16 \\ + \\ \hline 06 \end{array}$
 $\begin{array}{cccccc} 4 & 3 & 2 & 1 & 0 \\ (1 & 0 & 1 & 1 & 0) \end{array}_b$

$(10 \ 110)_b \rightarrow$ Öteklem 0 eklenmesi
 $\begin{array}{r} 10110 \\ \downarrow \\ 543210 \end{array}$
 $\begin{array}{r} 10110 \\ \downarrow \\ 3210 \end{array}$
 $2^5 + 2^3 + 2^2 = 32 + 8 + 4 = 44 = (22 \times 2)$

$(101)_b = (5)$
 $\rightarrow (10)_b = (2)$
 $10.111 \times \frac{5}{2}$
 $2^3 + 2^1 + 2^0 = 8 + 2 + 1 = 11 = (\frac{22}{2})$
 $\frac{1}{2} + \frac{1}{4} + \frac{1}{8}$

Binary (ikili): 0/1 $\rightarrow$ Bit

Bilgisayar: Veri, işleyen, I/O

Transistor $\rightarrow$ Yarıiletken teknolojisi,
elektron akışı kontrol ediliyor

$\rightarrow$ Anahtarlama

$\rightarrow$ Kuvvetlendirme

$\rightarrow$ Şu anki durum ve bir sonraki durumu

Sembol (mesaj)

Veri (16)

Karakter (klavye tuşları) $\rightarrow$ ASCII $\rightarrow$ her karakter $\rightarrow$ 8 bit

Renk tonları $\rightarrow$ 8 bit / 16 bit / 32 bit / ...

$$(278)_{10} = (b)_2 = 2^8 + 2^4 + 2^2 + 2^1$$

En yüksek 2^n

maksimum indis

olanlar $\rightarrow$ 1, olmayanlar $\rightarrow$ 0

$$\begin{array}{cccccccc} 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ (1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0)_{10} \end{array}$$

$$(100010110)_2 = (d)_{10} \\ = 2^8 + 2^4 + 2^2 + 2^1$$

Sağdan solara indisle
olanlara 2^n
Topla:

$$\begin{array}{r} 256 \\ 16 \\ 4 \\ 2 \end{array}$$

$$(278)_{10}$$

$$\begin{array}{r} 278 \\ 256 \\ \hline 22 \\ 16 \\ \hline 6 \\ 4 \\ \hline 2 \end{array}$$

OR - Veya (birtanesi 1 ise sonucu 1)

	OR	NOR
0 or 0 = 0	1	
0 or 1 = 1	0	
1 or 0 = 1	0	
1 or 1 = 1	0	

AND (ve) (hepsi 1 ise sonucu 1)

	AND	NAND
0 and 0 = 0	1	
0 and 1 = 0	1	
1 and 0 = 0	1	
1 and 1 = 1	0	

Not Teri

1 → 0
0 → 1

XOR (karşılaştırma)

0 XOR 0 → 0
0 XOR 1 → 1
1 XOR 0 → 1
1 XOR 1 → 0

$0 \text{ or } 0 = 0$
 $0 \text{ or } 1 = 1$
 $1 \text{ or } 0 = 1$
 $1 \text{ or } 1 = 1$

$$(10111011)_2 \text{ OR } (01110001)_2 = ?$$

$$\Downarrow$$

$$(11111011)_2 = (FB)_{16}$$

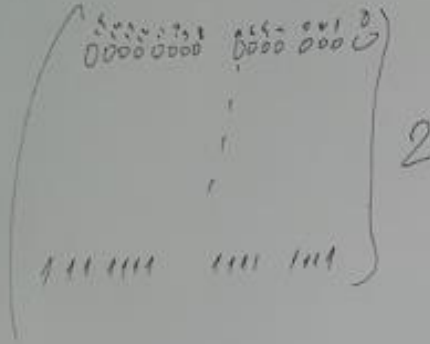
0 and 0 = 0
 0 and 1 = 0
 1 and 0 = 0
 1 and 1 = 1

$$(11001010)_2 \text{ AND } (00001100)_2 = (08)_{16}$$

278 kbyte kaç byte?

i° $278 \leq 2^m$ ile ifade edilmeli.

$$2^9 \text{ kbyte} = 2^9 * 2^{10} = 2^{19} \text{ byte.}$$



$$\begin{array}{r} 2^9 - \text{byte / 8 bitlik} \\ \text{ belleğin} \\ \text{ Adres hatları} \\ \hline 10^3 \\ \text{Adres hatları sayısının} \\ \hline \text{1 bit/sec, bit} \end{array}$$

Veri transferinde
Veri işlemi.

$$(781)_d = (?)_b = (?)_h$$
$$512 = 2^9 + 2^8 + 2^7 + 2^6 + 2^5$$

$$\begin{array}{r} 9 \quad 8 \quad 7 \quad 6 \quad 5 \quad 4 \quad 3 \quad 2 \quad 1 \quad 0 \\ (1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1)_b \end{array}$$

$$= (30D)_h$$

From Base 10 to Base 2: using table

- Input : a decimal number
- Output: the equivalent number in base 2
- Procedure:
 - Write a table as follows
 1. Find the largest two's power that is smaller than the number
 1. Decimal number 234 => largest two's power is 128
 2. Fill in 1 in corresponding digit, subtract 128 from the number => 106
 3. Repeat 1-2, until the number is 0
 4. Fill in empty digits with 0

...	512	256	128	64	32	16	8	4	2	1
			1	1	1	0	1	0	1	0

- Result is 11101010

Converting Binary to Decimal

Binary number system is base 2

- **0, 1**
- **Uses 2 numbers**

$$10010001 = 145$$

Base 2 representation	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal representation	128	64	32	16	8	4	2	1
Base 2 representation	1	0	0	1	0	0	0	1

Binary to Decimal

- Technique
 - Multiply each bit by 2^n , where n is the “weight” of the bit
 - The weight is the position of the bit, starting from 0 on the right
 - Add the results

$$101011_2 \Rightarrow \begin{array}{rclcl} 1 & \times & 2^0 & = & 1 \\ 1 & \times & 2^1 & = & 2 \\ 0 & \times & 2^2 & = & 0 \\ 1 & \times & 2^3 & = & 8 \\ 0 & \times & 2^4 & = & 0 \\ 1 & \times & 2^5 & = & 32 \\ & & & & 43_{10} \end{array}$$

$$\begin{array}{ccc} 1 & 1 & 1 \\ \uparrow & \uparrow & \uparrow \\ \text{position: } 2 & 1 & 0 \end{array}$$

$$\begin{aligned} 111 &= 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 1 \times 4 + 1 \times 2 + 1 \times 1 \\ &= 4 + 2 + 1 = 7 \end{aligned}$$

Decimal Sayıların Binary Sayılara Çevrilmesi:

$$(172)_{10} = (128 + 32 + 8 + 4)_{10} = (2^7 + 2^5 + 2^3 + 2^2)_{10} = (1010 \ 1100)_2 = (AC)_{16}$$

Binary'den decimal'a çevirme örnekleri

- Örnek:

- $(1010)_2 = (?)_{10}$

- $(1010)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

- $(1010)_2 = 8 + 0 + 2 + 0$

- $(1010)_2 = 10$

- Örnek:

- $(11001)_2 = (?)_{10}$

- $(11001)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

- $(11001)_2 = 16 + 8 + 0 + 0 + 1$

- $(11001)_2 = 25$

Decimal sayı sisteminden binary sayı sistemine çevirme örnekleri

- $(217)_d = (?)_b$
 - $(217)_d = (11011001)_b$
 - Sağlaması:
 - Maksimum ne var, $2^7 = 128$
 - $128 + 64 + 16 + 8 + 1 = 217$
 - İndislenir: 7 6 5 4 3 2 1 0 ise 2 üzeri olanlara 1 olmayanlar 0 yazılır.
 - $(11011001)_b$
-
- $(54)_d = (?)_2$
 - $32 + 16 + 4 + 2$
 - $(54)_d = (110110)_b$



Hex Numbering System

Groups of bits

- Groups of bits can represent data or information
 - 1 bit - 2 alternatives
 - 2 bits - 4 alternatives
 - 3 bits - 8 alternatives
 - 4 bits - 16 alternatives
 - n bits - 2^n alternatives
 - 8 bits - $2^8 = 256$ alternatives
 - a group of 8 bits is called a byte

4 Bits

• 0000	• 1000
• 0001	• 1001
• 0010	• 1010
• 0011	• 1011
• 0100	• 1100
• 0101	• 1101
• 0110	• 1110
• 0111	• 1111

Bit Gruplandırma (Bellek)

- n bit var ise gruptama 2^n
- Bit gruplandırma adres hatlarında, bellek ve bellek gözü seçmede. I/O birimi seçmede.
- 64kbyte kaç byte? $64 * 2^{10} \text{ byte} = 2^6 * 2^{10} = 2^{16} \text{ byte}$.
 - Soru: adres hattı kaç adet? 16 adet. Neden? Bit gruplandırma sayısı, $n=16$
 - Açıklama: 64 Kbyte bellek kapasitesini tanımladığı için 16 adet adres bus hattı gelir.
 $2^{16} = 2^6 * 2^{10} = 64 \text{Kbyte}$

Radix Number System

- Base – 2 (binary numbers), bit:0/1
 - 0 1
- Base – 8 (octal numbers)
 - 0 1 2 3 4 5 6 7
- Base – 16 (hexadecimal numbers), 0 ve 1 lerin dizisinden oluşan ikil sayı sisteminde sağdan başlayarak 4'lü gruplar ayrılır.
 - 0 1 2 3 4 5 6 7 8 9 A B C D E F

Binary/Hexidecimal

Decimal	Binary	Hexidecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Conversions:

1000 1110 (binary)

8 E (hex)

Notations for hex:

- 0x8E
- 8Eh
- 8E₁₆

Bilgisayarda, tüm işlevler 1/0 ikili sayı sisteminde. 1 ve 0 lardan oluşan dizi var.

- Giriş ve Çıkış
- Veri Transferi
- Saklama
- Manipülasyon (Aritmetik, Mantıksal; kontrol işlemleri, kesme, ...)

Veri: Sinyal (Ses, EM, Elektriksel, ..),
Görüntü, Resim, Metin; portlar,
klavye, mouse

Hex Sayı Sistemi Nedir ?

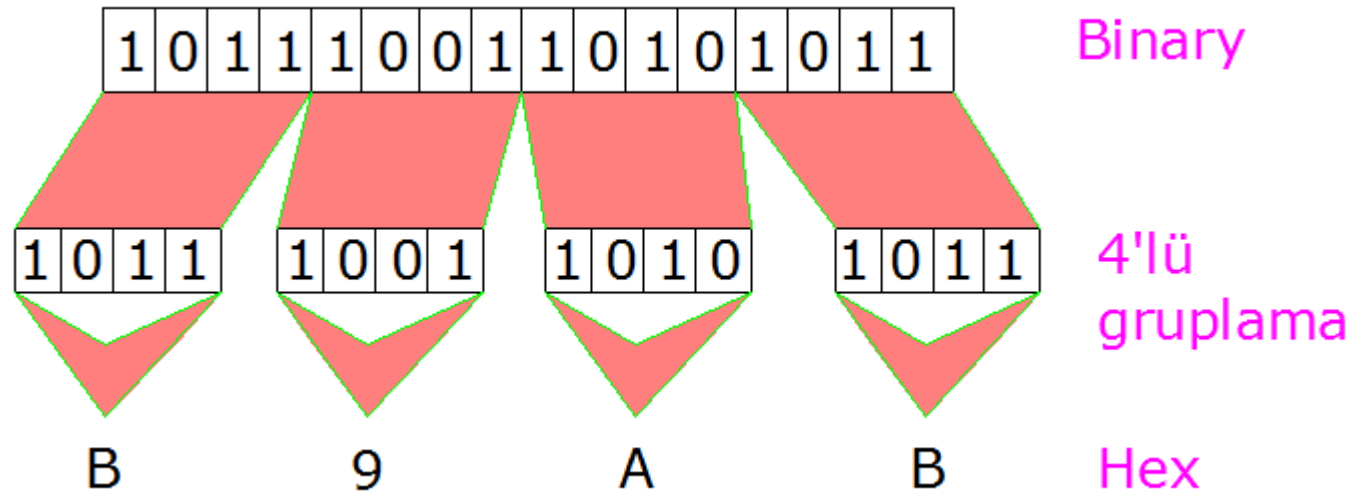
- Binary ve desimal sistemden sonra mantık olarak aynı ancak fark olarak 16 lıkta bana sahip sayılar hexa desimal sayılardır.
- Heks sisteminde kullanabileceğimiz rakamlar benzer mantıkla yine 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 şeklindedir.
- Ancak hexa desimal sisteme göre yazılan sayılar 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F, şeklindedir.
- Dijital Elektronikte binary sayıların heksa desimala çevrilip bellekte kodlanması sağlanır.
- Sayılar binary'den heksadesimale çevrilirken sağdan sola doğru dörder basamak olmak üzere gruplandırılır.
- Çünkü heksa decimal sayı sisteminin tabanı 16 dır ve binary sayı sisteminde 0-15 sayıları, 4 bit ile ifade edilebilmektedir.

Numbers

- Short-hand for all these 1s and 0s
- HEX notation
- Hex is used as a notation for any sequence of bits (e.g. ASCII characters require just two hex digits)
- Each group of 4 bits represents a number in the range 0 – 15
- Thus:
- 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0009
- 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 FFF7
- 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0000

0 0 0 0 =	0	1 0 0 0 =	8
0 0 0 1 =	1	1 0 0 1 =	9
0 0 1 0 =	2	1 0 1 0 =	A
0 0 1 1 =	3	1 0 1 1 =	B
0 1 0 0 =	4	1 1 0 0 =	C
0 1 0 1 =	5	1 0 0 1 =	D
0 1 1 0 =	6	1 1 1 0 =	E
0 1 1 1 =	7	1 1 1 1 =	F

Binary Sayının Hexa decimal'e Çevrilmesi



Hexadecimal to Binary

- Convert each hexadecimal digit to a 4-bit equivalent binary representation

$$10AF_{16} = ?_2$$

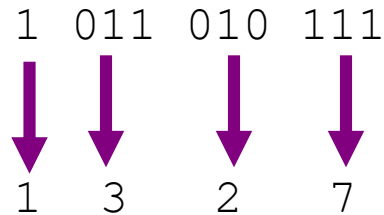
1	0	A	F
↓	↓	↓	↓
0001	0000	1010	1111

$$10AF_{16} = 0001000010101111_2$$

Binary to Octal

- Technique
 - Group bits in threes, starting on right
 - Convert to octal digits

$$1011010111_2 = ?_8$$

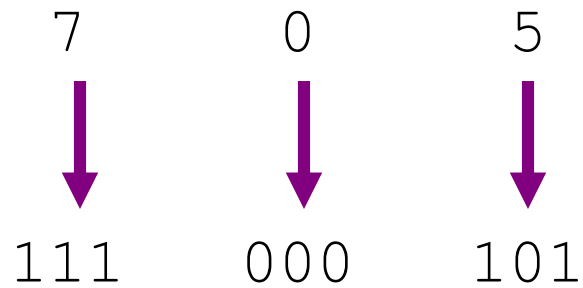


$$1011010111_2 = 1327_8$$

Octal to Binary

- Technique
 - Convert each octal digit to a 3-bit equivalent binary representation

$$705_8 = ?_2$$



$$705_8 = 111000101_2$$



ASCII Coding

There needs a standard way

- ASCII code: **American Standard Code for Information Interchange**
 - ASCII codes represent [text](#) in [computers](#), [communications](#) equipment, and other devices that use text.
 - ASCII kodları, bilgisayarlardaki, klavyedeki tüm tuşların 8 bitlik değerini temsil eder. Böylece klavyedeki tuşlar ile bellek arasındaki iletişim sayısal olarak yapılmış olur.
 - 128 characters:
 - 33 are non-printing [control characters](#) (now mostly obsolete)^[7] that affect how text and space is processed
 - 94 are printable characters
 - [space](#) is considered an invisible graphic

ASCII Coding

- Bilgisayarda 0 ve 1'lerle karakterleri ifade etmek için ikili kodlama sistemleri kullanılır.
- ASCII, karakterleri göstermek için sekiz bit (bir bayt) kullanır. Yeni geliştirilen Unicode ise karakterleri göstermek için onaltı bit kullanır:
- ASCII (American Standard Code for Information Interchange - bilgi değişimi için Amerikan standart kodlaması)
- EBCDIC (Extended Binary Coded Decimal Interchange Code - genişletilmiş ikili kodlamalı onluk sistem değiştirme kodlaması): IBM şirketi tarafından ana bilgisayarlarda kullanılmak için geliştirilmiştir.
- Unicode: Çince ve Japonca gibi dilleri desteklemek için tasarlanmış onaltı bit kullanan kodlamadır. Bu diller sekiz bit kullanan ASCII ve EBCDIC kodlamaları ile gösterilemeyecek kadar çok sayıda karakter kullanırlar. Unicode kodlaması, IBM, Apple ve Microsoft şirketlerinin desteklediği Unicode şirketi tarafından geliştirilmiştir.

■ | <u>Kodlama</u> | <u>Kullanımı</u> | |----------------|------------------| |----------------|------------------|

ASCII	Kişisel bilgisayarlar
-------	-----------------------

EBCDIC	Anabilgisayarlar
--------	------------------

Unicode	Uluslararası diller
---------	---------------------

- Klavyede bir tuşa bastığınız zaman, tuşa karşılık gelen karakter, bilgisayarın anlayabileceği bir dizi bite çevirilir. Örneğin, klavyede A harfine basmak bilgisayar bunu (41)h=(01000001)b ASCII koduna çevirir.
- Dökümanlar değişik bilgisayarlar veya uygulama programları tarafından paylaşıldığı zaman, aynı kodlama sistemi kullanılmalıdır.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

ASCII Character Codes (2)

Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char
20	(Space)	30	0	40	@	50	P	60	'	70	p
21	!	31	1	41	A	51	Q	61	a	71	q
22	"	32	2	42	B	52	R	62	b	72	r
23	#	33	3	43	C	53	S	63	c	73	s
24	\$	34	4	44	D	54	T	64	d	74	t
25	%	35	5	45	E	55	U	65	e	75	u
26	&	36	6	46	F	56	V	66	f	76	v
27	'	37	7	47	G	57	W	67	g	77	w
28	(	38	8	48	H	58	X	68	h	78	x
29	)	39	9	49	I	59	Y	69	i	79	y
2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
2C	,	3C	<	4C	L	5C	\	6C	l	7C	
2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

The ASCII Character set: characters 32 – 127.

Bits as Codes

- Ascii kodu bizim bilgisayarda görsel olarak girdiğimiz karakter, harf ve rakamların bilgisayar dilindeki temsil edilme şeklidir diyebiliriz.
- Yani bilgisayarımızın o karakteri, harfi veya rakamı belleğinde saklama biçimidir,
- Açılımı ASCII (American Standard Code for Information Interchange) olan
- Bu kodlama sistemi ilk olarak telgraf kodlarında kullanılmıştır
- Klavyedeki her tuşun 8 bitlik bir karşılığı vardır.
- İkili Kodlar her harf, rakam ve özel karakteri temsil eder.
- ASCII: Her karakter benzersiz bir 8 bitlik koddur
- ASCII: 26 harf, 10 hane, özel karakterler için 256 benzersiz kod
- Unicode: 100.000'den fazla benzersiz karakteri destekler.
- Klavyeden girilen her karakter bilgisayara ASCII kodu ile giriş yapar.

ASCII	SYMBOL	ASCII	SYMBOL
00110000	0	01001110	N
00110001	1	01001111	O
00110010	2	01010000	P
00110011	3	01010001	Q
00110100	4	01010010	R
00110101	5	01010011	S
00110110	6	01010100	T
00110111	7	01010101	U
00111000	8	01010110	V
00111001	9	01010111	W
01000001	A	01011000	X
01000010	B	01011001	Y
01000011	C	01011010	Z
01000100	D	00100001	!
01000101	E	00100010	"
01000110	F	00100011	#
01000111	G	00100100	\$
01001000	H	00100101	%
01001001	I	00100110	&
01001010	J	00101000	(
01001011	K	00101001	)
01001100	L	00101010	*
01001101	M	00101011	+

Fractions

Virgüllü ondalık sayıyı binarysayıya çevirme

- 10 tabanında verilen sayının tamsayı kısmı sürekli olarak istenen tabana bölünerek kalanlar sağdan sola doğru yazılır.
- 10 tabanında verilen sayının kesir kısmı için ise, bu kısım sürekli olarak istenen taban ile çarpılır ve çarpımın sonucu istenen basamağa erişene kadar (ya da 0 çıkana kadar) işlemin tamsayı kısmı alınır.

• Örnek: $(121.125)_{10}$ sayısını 2 tabanına dönüştürünüz.

- $121 / 2 = 60$
- $60 / 2 = 30$
- $30 / 2 = 15$
- $15 / 2 = 7$
- $7 / 2 = 3$
- $3 / 2 = 1$
- $1 / 2 = 0$

Kalan

1
0
0
1
1
1
1



• Kesirli kısmı dönüştürelim.

- $0.125 * 2 = 0.250$
- $0.250 * 2 = 0.500$
- $0.500 * 2 = 1.000$

0
0
1



• Sonuç = 1111001.001

Ondalık sayı sistemini ikil sayı sistemine dönüştürülürken 2'ye bölünerek en son böüm geröeklene devam edilir. Kalan var ise 1 yok ise 0 yapılır.

Here are some handy facts:

Power of 2	Base 2	Base 10
2^{-4}	.0001	.0625
2^{-3}	.001	.125
2^{-2}	.01	.25
2^{-1}	.1	.5
2^0	1	1
2^1	10	2
2^2	100	4
2^3	1000	8
2^4	10000	16
2^5	100000	32
2^6	1000000	64
2^7	10000000	128
2^8	100000000	256

N	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Fractions

- We can extend our unsigned representational system of binary to include a decimal point
 - After the decimal point, the i exponent, in 2^i , becomes negative
 - So, we now have the $\frac{1}{2}$ column, the $\frac{1}{4}$ column, etc
 - $(1011.1001)_b =$
 - Indisleme: iki grupta yapılır. Noktadan sola ve sağa doğru... 3,2,1,0,-1,-2,-3,-4
 - $1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 + 1*2^{-1} + 0*2^{-2} + 0*2^{-3} + 1*2^{-4} =$
 - $8 + 2 + 1 + \frac{1}{2} + \frac{1}{16} =$
 - $11 \frac{9}{16} = 11.5625$
 - What is .4304? Use 8-bits with 4 fraction bits
 - .4304 has a .25, .125, .03125, .015625, and more fractions, but this exceeds the number of fraction bits so the number is 0000.0110
 - But $0000.0110 = .125 + 0.3125 = .375$, we have a loss in precision!
 - In the fraction representation, our decimal point is typically fixed, so this is often known as *fixed point representation*
 - We will cover a floating point representation later

Fractions

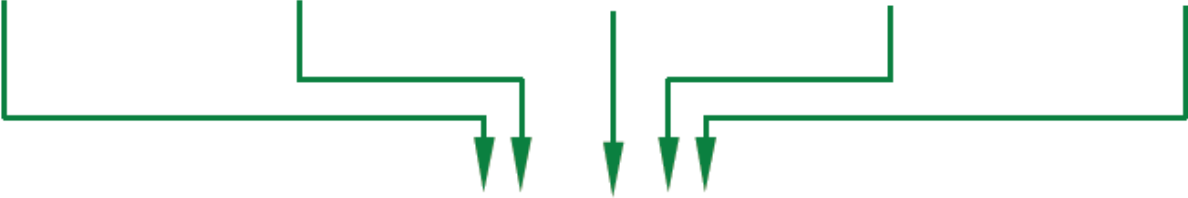
- Decimal to decimal (just for fun)

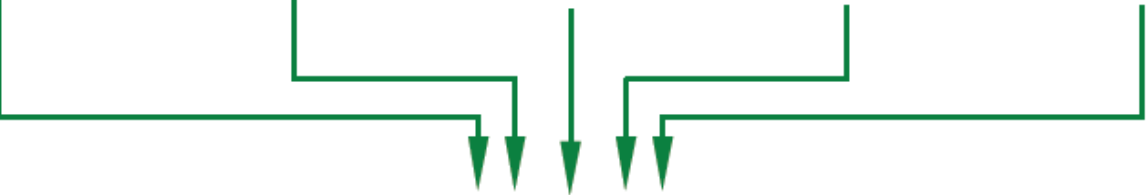
$$\begin{array}{rcl} 3.14 & \Rightarrow & \\ & 4 \times 10^{-2} = & 0.04 \\ & 1 \times 10^{-1} = & 0.1 \\ & 3 \times 10^0 = & 3 \\ & & 3.14 \end{array}$$

- Binary to decimal

$$\begin{array}{rcl} 10.1011 & \Rightarrow & \\ & 1 \times 2^{-4} = & 0.0625 \\ & 1 \times 2^{-3} = & 0.125 \\ & 0 \times 2^{-2} = & 0.0 \\ & 1 \times 2^{-1} = & 0.5 \\ & 0 \times 2^0 = & 0.0 \\ & 1 \times 2^1 = & 2.0 \\ & & 2.6875 \end{array}$$

Works for Fractional Numbers too...

$$3 \times 10^1 + 5 \times 10^0 + 4 \times 10^{-1} + 6 \times 10^{-2} + 2 \times 10^{-3}$$

$$35.462$$

$$1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3}$$

$$10.110_b = 2.75_d$$

$$\begin{array}{ccccccccc} 1 \times 2^1 + & 1 \times 2^0 + & 0 \times 2^{-1} + & 1 \times 2^{-2} + & 1 \times 2^{-3} \\ | & | & | & | & | \\ \hline & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ & & 1 & 1 & 0 & 1 & 1 \\ & & 11.011_b = 3.375_d \end{array}$$

Binary'den decimal'a çevirme örnekleri

- Örnek:

- $(111,101)_2 = (?)_{10}$

- $(111,101)_2 =$

- $1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

- $(111,101)_2 =$

- $1 \times 4 + 1 \times 2 + 1 \times 1 + 1 \times \frac{1}{2} + 0 \times \frac{1}{4} + 1 \times \frac{1}{8}$

- $(111,101)_2 =$

- $4 + 2 + 1 + 0,5 + 0 + 0,125$

- $(111,101)_2 = (7,625)_{10}$

- Örnek:

- $(100111,01)_2 = (?)_{10}$

- $(100111,01)_2 = (39,25)_{10}$

Twos Complement Representation

Unsigned Binary Integers

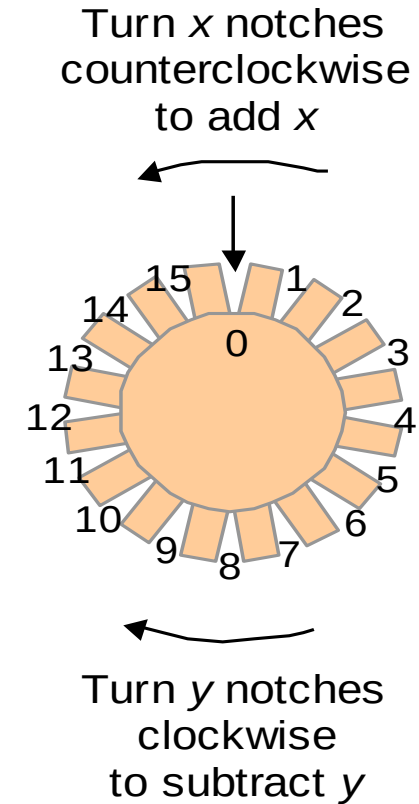
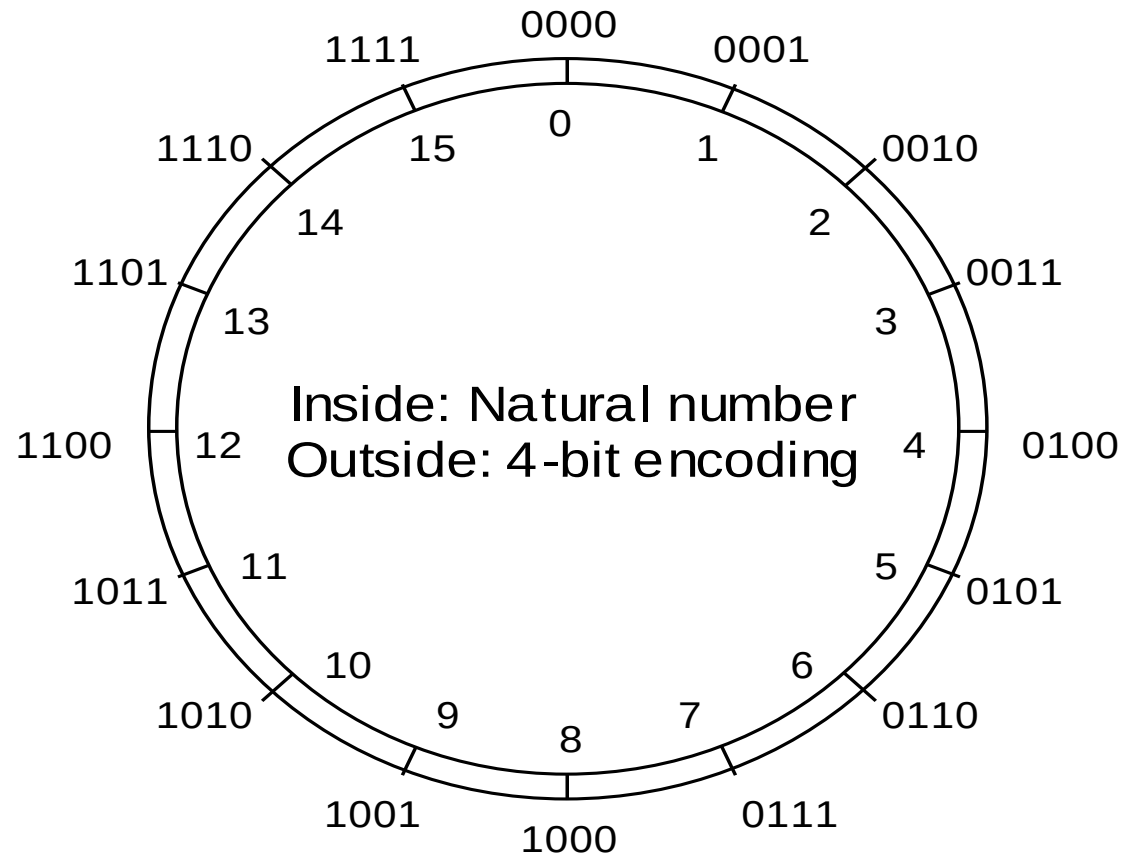


Figure Schematic representation of 4-bit code for integers in $[0, 15]$.

Two's-Complement Representation

With k bits, numbers in the range $[-2^{k-1}, 2^{k-1} - 1]$ represented.
Negation is performed by inverting all bits and adding 1.

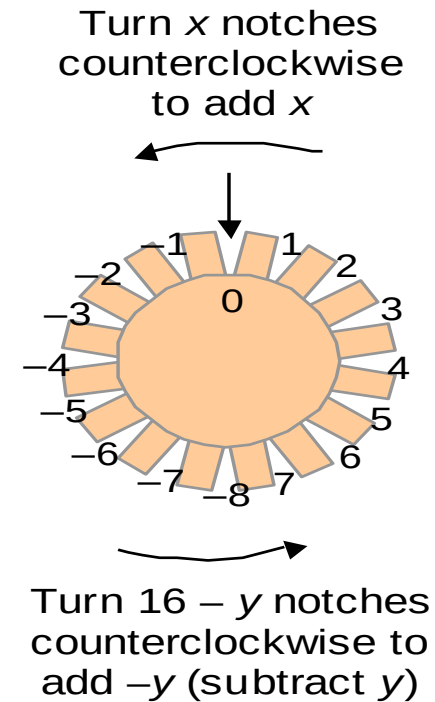
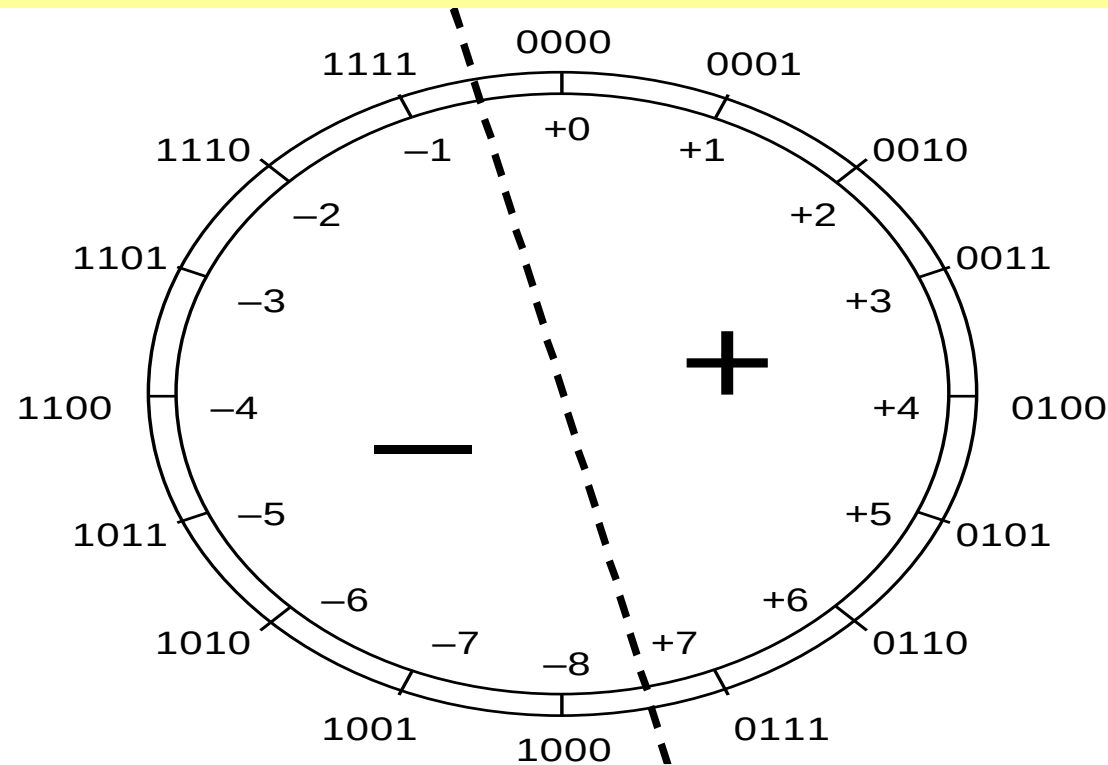


Figure Schematic representation of 4-bit 2's-complement code for integers in $[-8, +7]$.

Fixed-Point 2's-Complement Numbers

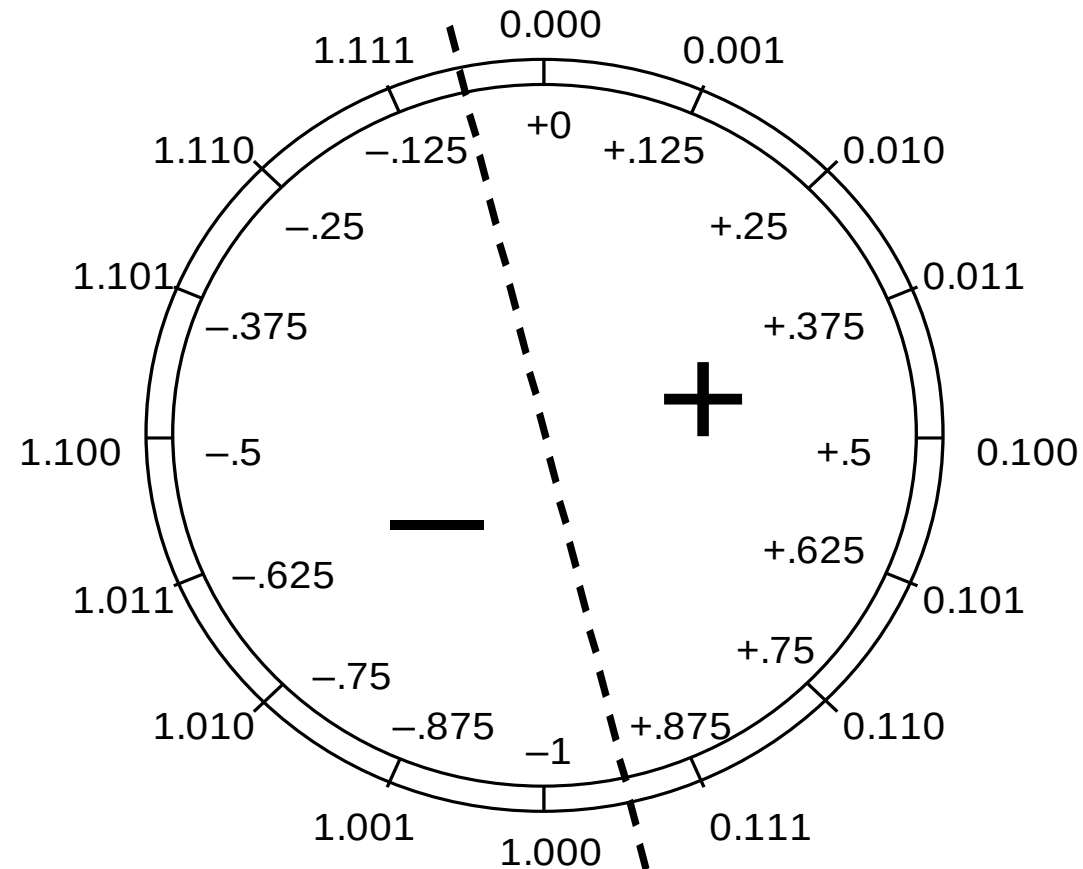
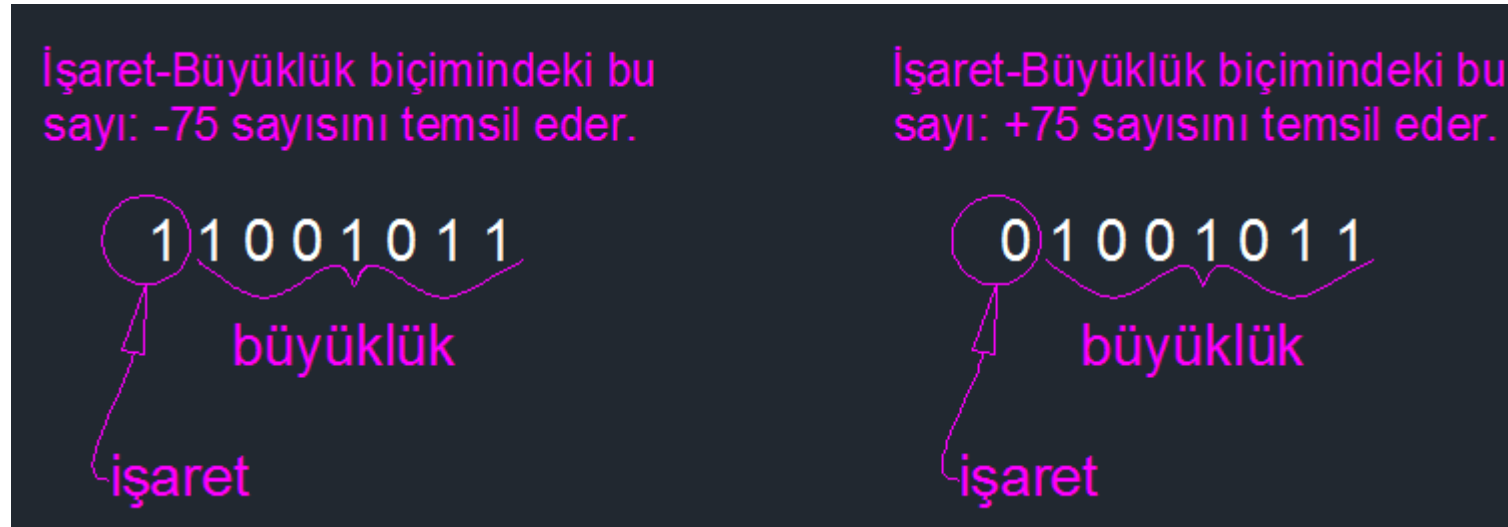
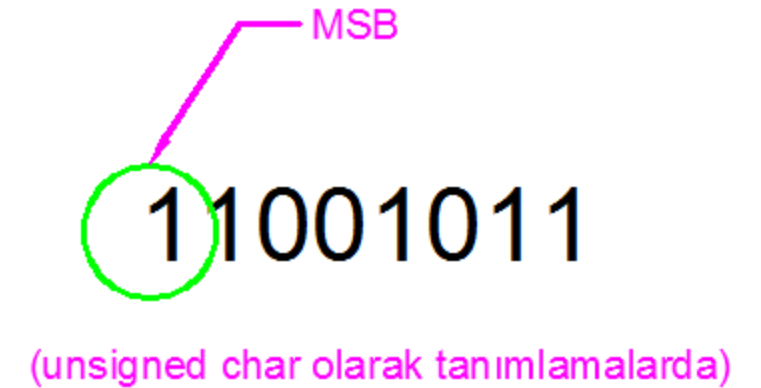
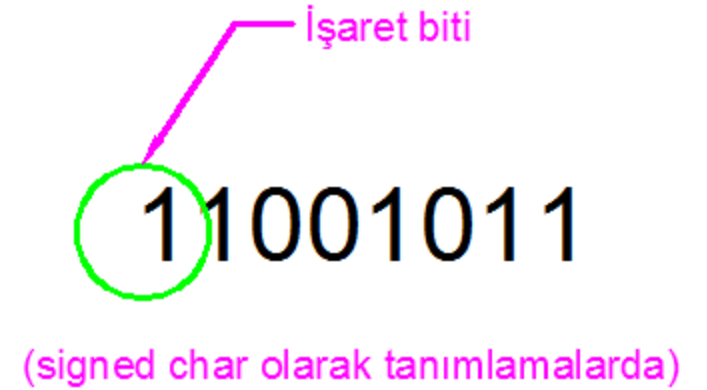


Figure Schematic representation of 4-bit 2's-complement encoding for (1 + 3)-bit fixed-point numbers in the range $[-1, +7/8]$.

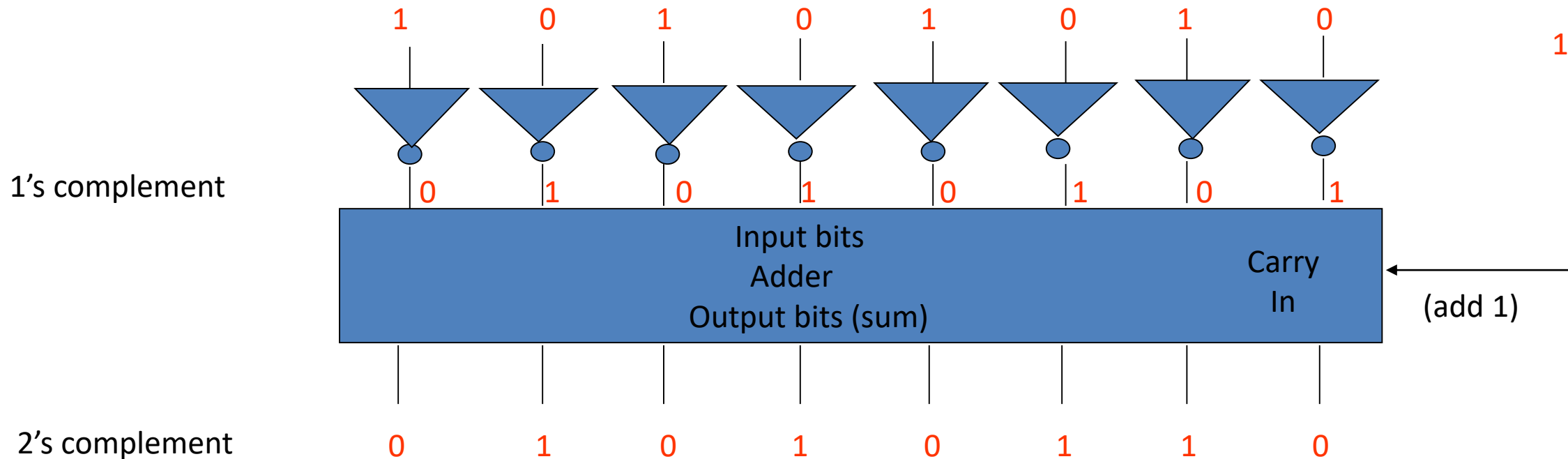
İşaretli binary sayılar

- İkili sayı sistemlerinde 1 byte eile (8 bit) 0-255 arası pozitif sayıları ifade edebiliriz, fakat
- •negatif sayıları ifade etmek için kullanıldığında en soldaki bit işaret bitidir.
- •Eğer işaret biti 1 ise sayı negatif, 0 ise sayı pozitiftir.



Complements of Binary Numbers

- 2's complement
- Find 1's complement and then add 1



İşaretli (signed) binary sayıların bir gösterim şekli de işaret-büyüklik biçiminde gösterimdir.

ÖRNEK: +19 ve -19 desimal sayılarını işaret-büyüklik formunda 1 byte binary sayı ile ifade ediniz.

- Çözüm:
- $(19)_{10} = 16 + 3 = 16 + 2 + 1 = 2^4 + 2^1 + 2^0$;
- İndisleme: 4, 3, 2, 1, 0
- $(10011)_2$
- $(+19)_d = (00010011)_b$
- $(-19)_d = (?)_b$ Not: 8, 16, 32, 64 bit düzeninde işlem yapılır.
- $(+19)_d = (00010011)_b \rightarrow$ ikil sayı sisteminde tersi alınır; 1 $\rightarrow$ 0, 0 $\rightarrow$ 1 konur. Sonra +1 eklenir.
- $(-19)_d = (11101100)_b + 1 = (1110\ 1101)_b$

Negatif Sayıların ikil sayı sistemine çevrilmesi

- Önce negatif işareti olmadan ondalık sayıyı ikili sayı sistemine çevrilir.
- En sağdan başlayarak dörterli gruplara ayrılır.
- En soldaki bit 1 ise 0'lar ilave ederek dörtlü gruba tamamlanır ya da sıfırlardan oluşan yeni bir dörtlü grup ilave edilir. (8, 16, 32, 64, ...)
- Elde ettiğiniz ikili sayı sisteminde 1 yerine 0; 0 yerine 1 yazılır.
- Son olarak yeni ikili sayı sistemi +1 ile toplanır.

$$\begin{aligned} +18 &= 0001\ 0010 \text{ (twos complement)} \\ \text{bitwise complement} &= 1110\ 1101 \\ &+ \quad \quad \quad 1 \\ \hline 1110\ 1110 &= -18 \end{aligned}$$

$$\begin{aligned} -18 &= 1110\ 1110 \text{ (twos complement)} \\ \text{bitwise complement} &= 0001\ 0001 \\ &+ \quad \quad \quad 1 \\ 0001\ 0010 &= +18 \end{aligned}$$

Negatif sayılar

- Byte: 8 bit data tanımlar; 8 bitlik bellek gözünü işaret eder.
- Örnek db -4
 $(4)_d = (0000\ 0100)_b$
 $(-4)_d = \text{Binary } (4) \text{ Tersisi } +1 :$
 $(-4)_d = 1111\ 1011 + 1 = 1111\ 1100 = (\text{FC})_h$
- Örnek dW -4
 $(4)_d = (0000\ 0000\ 0000\ 0100)_b$
 $(-4)_d = \text{Word}(4) \text{ Tersisi } +1$
 $(-4)_d = (1111\ 1111\ 1111\ 1011)_b + 1 = (1111\ 1111\ 1111\ 1100)_b = (\text{FFFC})_h$

Overflow

- Taşma, toplamdaki bit sayısı, toplanan ve artırılan bitlerin sayısını aştığında meydana gelir.
- Taşma, yanlış işaret ile gösterilir. Yalnızca her iki sayı da pozitif olduğunda veya her iki sayı da negatif olduğunda oluşur



Coding

Kodlama

- 0 ile 9 arasındaki ikil sayı sistemi kullanılır.
- Genel olarak kodlama, görülebilen, okunabilen, yazı, sayı ve işaretlerin değiştirilmesi olarak tanımlanır.
- Binary Coded Decimal Kodlaması(BCD) sisteminde her desimalsayı karakteri için, dört bit kullanılır.

<u>Decimal</u>	BCD
25	0010 0101
32	0011 0010
679	0110 0111 1001
2571	0010 0101 0111 0001

<u>Decimal Sayıları</u>	BCD Kodu
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Characters and Other Codes

- To represent information as strings of alpha-numeric characters.
- **Binary Coded Decimal (BCD)**
 - **Used to represent the decimal digits 0 - 9.**
 - 4 bits are used.
 - Each bit position has a weight associated with it (***weighted code***).
 - Weights are: 8, 4, 2, and 1 from MSB to LSB (called 8-4-2-1 code).
 - BCD Codes:

0: 0000	1: 0001	2: 0010	3: 0011	4: 0100
5: 0101	6: 0110	7: 0111	8: 1000	9: 1001
 - Used to encode numbers for output to numerical displays
 - Used in processors that perform decimal arithmetic.
 - **Example:** $(9750)_{10} = (1001011101010000)_{BCD}$

BCD kodlama örnekleri

Örnek: $(10100110)_2$ sayısını BCD koduna çeviriniz.

- Binary sayıyı ilk olarak desimale çevirelim;
- $2+4+32+128=166$
- $(10100110)_2 = (166)_{10}$
- BCD koduna çevirirsek
- $(000101100110)_{BCD}$

Örnek: $(1010110101)_2$ sayısını BCD koduna çeviriniz.

- Binary sayıyı ilk olarak desimale çevirelim;
- $1+4+16+32+128+512=693$
- $(1010110101)_2 = (693)_{10}$
- BCD koduna çevirirsek
- $(011010010011)_{BCD}$ bulunur.

0 -12 Decimal sayıları için Gray kodlaması

- Birinci bit aynen aşağıya indirilir,
- ikinci bit aşağıya indirilen bit ile toplanır ve sonuç hemen aşağıya indirilen birinci bitin sağına yazılır.
- Üçüncü bit, aşağıya yazılan ikinci bit ile toplanır ve ikinci bit 'in yanına yazılır.
- son bite kadar işlem böyle devam ettirilir.

Örnek:(011010111101)₂sayısını graykoduna çeviriniz.

- **İlk rakam aynen alınır,**
- **Diğer basamaklarda sayı değişiyorsa 1, sayı değişmiyorsa 0 alınarak sonuca ulaşılır.**
- (011010111101)₂
- (010111100011)_{gray}

Örnek:(1110001101101)₂sayısını graykoduna çeviriniz.

- İlk rakam aynen alınır,
- Diğer basamaklarda sayı değişiyorsa 1, sayı değişmiyorsa 0 alınarak sonuca ulaşılır.
- (1110001101101)₂
- (1001001011011)_{gray}

Örnek: 297 decimalsayısını graykodu ile kodlayınız?

(297)₁₀= (100101001)₂

(100101001)₂=(110111101)_{gray}

<u>Decimal</u>	<u>Binary</u>	<u>Gray</u>
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010

Gray Kodu

- Gray kodunun, sütun taraması esasına göre çalışan cihazlarda oldukça geniş bir kullanım alanı vardır.
- Geçişler sırasında hatayı minimuma indirmek için geliştirilmiş bir koddur. İsmi mucidi Frank GRAY'den alır.
- Özelliği, Ardarda iki sayı arasında sadece tek bit değişikliğidir.
- Normal Binary kodunda ardışık sayılarda çoğu kez birden fazla bit değişikliği söz konusudur. Devrelerde 0-1 durumları arasındaki değişim sırasında okuyucu farklı değerler okuyabilir.
- Gray kodunda sadece tek bit değiştiğinden, ve binary sayı olduğundan, gray kodunda bu hata ortadan kaldırılmıştır.

Graykodu ile kodlanmış sayıyı tekrar elde etme örnekleri

Örnek: $(01011101)_{gray}$ graykodlu sayıyı binary'e çevirelim.

- $(01011101)_{gray}$
- $(01101001)_2$
- İlk sayı aynen aşağı alınır,
- Sağındaki diğer sayılar
- yukarıdaki sayı 1 olduğunda rakamın değiştiğine,
- 0 olduğunda değişmediğine alamet olarak değerlendirilir.

Örnek: $(10101100111)_{gray}$

- graykodlu sayıyı binary'e çevirelim.
- $(10101100111)_{gray}$
- $(11001000101)_2$

Characters and Other Codes

- ***Gray Code***
 - ***Cyclic code***: A circular shifting of a code word produces another code word.
 - ***Gray code***: A cyclic code with the property that two consecutive code words differ in only 1 bit (the *distance* between the two code words is 1).
 - Gray code for decimal numbers 0 - 15: See Table 1.12



Karekod(barkod)

Karekod(barkod)

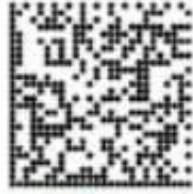
- Karekod, kare veya dikdörtgen biçimlerde basılabilen 2 boyutlu barkodun ismidir. Kare veya dikdörtgen şeklinde olan bu yapının genel adı ise Data Matrixtir.
- Karekodkelimesi ilk olarak, Beşeri Tıbbi Ürünler Barkod Uygulama Kılavuzunda kullanılmıştır.
- Ülkemizdeki ilk uygulama alanı, ilaç sektörüdür.
- Karekodbarkodların başlıcaları; QR Kod, Data MatrixKod, AztekKod



QR-Code



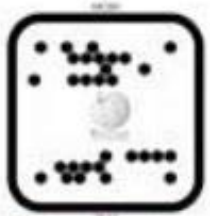
DataMatrix



Cool-Data-Matrix



Aztec



Beetagg



Trillcode



Quickmark



Shotcode



mCode





Error Detection Codes and Correction Codes

Error Detection Codes and Correction Codes

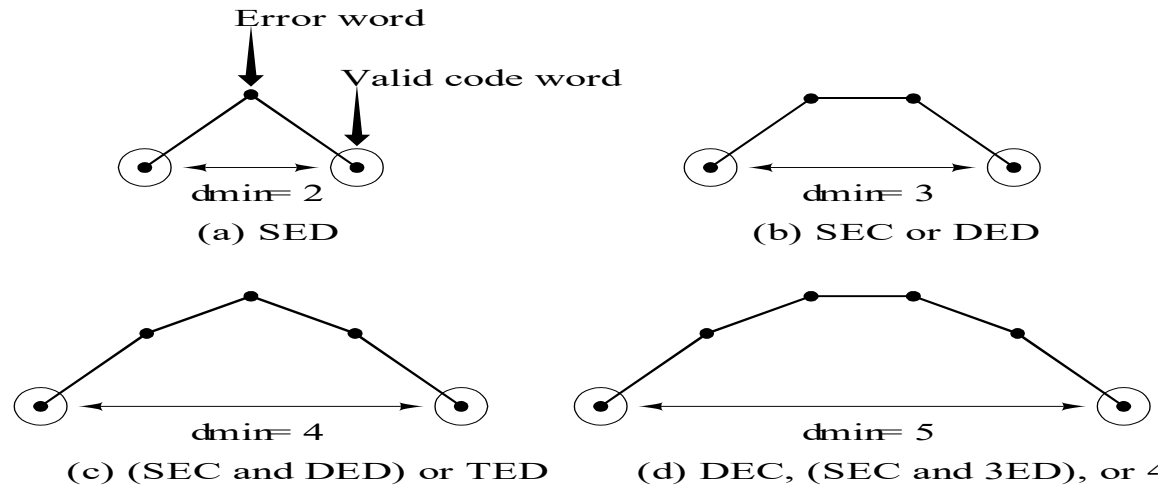
- **An error:** An incorrect value in one or more bits.
- **Single error:** An incorrect value in only one bit.
- **Multiple error:** One or more bits are incorrect.
- Errors are introduced by hardware failures, external interference (noise), or other unwanted events.
- **Error detection/correction code:** Information is encoded in such a way that a particular class of errors can be detected and/or corrected.
- Let I and J be n -bit binary information words
 - $w(I)$: the number of 1's in I (*weight*)
 - $d(I, J)$: the number of bit positions in which I and J differ (*distance*)
- **Example:** $I = (01101100)$ and $J = (11000100)$
 - $w(I) = 4$ and $w(J) = 3$
 - $d(I, J) = 3$.

Error Detection Codes and Correction Codes

- General Properties
 - *Minimum distance*, d_{min} , of a code C: for any two code words I and J in C,
 $d(I, J) \geq d_{min}$
 - A code provides t *error correction* plus *detection of* s additional errors if and only if the following inequality is satisfied.
$$2t + s + 1 \leq d_{min} \quad (1.25)$$
 - **Example:**
 - Single-error detection (SED): $s = 1, t = 0, d_{min} = 2$.
 - Single-error correction (SEC): $s = 0, t = 1, d_{min} = 3$.
 - Single-error correction and double-error detection (SEC and DED):
 $s = t = 1, d_{min} = 4$.

Error Detection Codes and Correction Codes

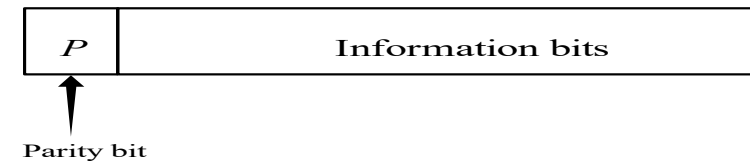
- Relationship between the minimum distance between code words and the ability to detect and correct errors:



Error Detection Codes and Correction Codes

- ***Simple Parity Code***

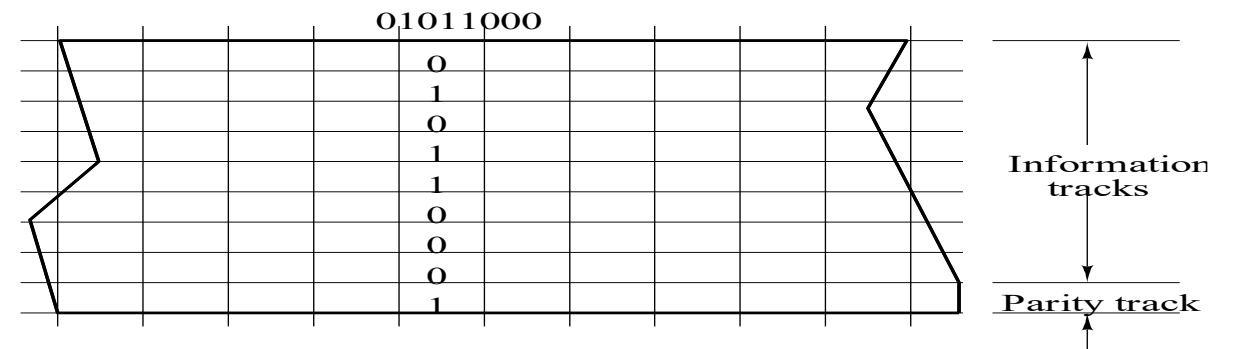
- Concatenate (|) a *parity bit*, P , to each code word of C .



- *Odd-parity code*: $w(P|C)$ is odd.

- *Even-parity code*: $w(P|C)$ is even.

- Parity coding on magnetic tape:



Error Detection Codes and Correction Codes

- **Example:** Odd-parity code for ASCII code characters:

Character	ASCII Code	Odd-parity Code
0	0110000	10110000
X	1011000	01011000
=	0111100	1111100
BEL	0000111	00000111

- Error detection: Check whether a code word has the correct parity.
- Single-error detection code ($d_{\min} = 2$).
- **Two-out-of-Five Code**
 - Each code word has exactly two 1's and three 0's.
 - Detects single errors and multiple errors in adjacent bits.



Hamming Codes

Hamming Codes (1)

- Multiple check bits are employed.
- Each check bit is defined over (or covers) a subset of the information bits.
- Subsets overlap so that each information bit is in at least two subsets.
- d_{min} is equal to the weight of the minimum-weight nonzero code word.
- **Hamming Code 1**
 - $d_{min} = 3$, single error correction code.
 - Let C be the set of all code words:
an error word with single error: c_e
the correct code word for the error word: c
then, $d(c_e, c) = 1$ and $d(c_e, w) > 1$ for all other $w \in C$
 - So, a single error can be detected and corrected by finding out the code word which differs in 1 bit position from the error word.

Hamming Codes (2)

- A code word consists of 4 information bits and 3 check bits:

$$c = (i_3 i_2 i_1 i_0 c_2 c_1 c_0)$$

- Each check bit covers:

$$c_2: i_3, i_2, i_1 \quad c_1: i_3, i_2, i_0 \quad c_0: i_3, i_1, i_0$$

- This relationship is specified by the *generating matrix*, G :

$$G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix} = \begin{bmatrix} 1000 p_{11} p_{12} p_{13} \\ 0100 p_{21} p_{22} p_{23} \\ 0010 p_{31} p_{32} p_{33} \\ 0001 p_{41} p_{42} p_{43} \end{bmatrix}$$

- Encoding of an information word i to produce a code word, c :

$$c = iG$$

Hamming Codes (3)

- Decoding can be done using the *parity-check matrix*, H :

$$H = \begin{bmatrix} p_{11} & p_{21} & p_{31} & p_{41} & 1 & 0 & 0 \\ p_{12} & p_{22} & p_{32} & p_{42} & 0 & 1 & 0 \\ p_{13} & p_{23} & p_{33} & p_{43} & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

- H matrix is can be derived from G matrix.
- An n -tuple c is a code word generated by G if and only if
$$Hc^T = 0$$
- Let d be a data word corresponding to a code word c , which has been corrupted by an error pattern e . Then

$$d = c + e$$

- Decoding:
 - Compute the syndrome, s , of d using H matrix.
 - s tells the position of the erroneous bit.

Hamming Codes (4)

– Computation of the syndrome:

$$s = Hd^T$$

$$= H(c + e)^T$$

$$= Hc^T + He^T$$

$$= 0 + He^T$$

$$= He^T$$

- Note: All computations are performed using *modulo-2 arithmetic*.

Hamming Codes (5)

- **Hamming Code 2)**

- $d_{min} = 4$, single error correction and double-error detection.
- The generator and parity-check matrices are:

$$G = \begin{bmatrix} 10000111 \\ 01001110 \\ 00101101 \\ 00011011 \end{bmatrix}$$

$$H = \begin{bmatrix} 01111000 \\ 11100100 \\ 11010010 \\ 10110001 \end{bmatrix}$$

Odd-weight-column code:

- H matrix has an odd number of ones in each column.
- Example: Hamming Code 2.
- Has many properties; single-error correction, double-error detection, multiple-error detection, low cost encoding and decoding, etc.

Hamming Codes (6)

- Hamming codes are most easily designed by specifying the H matrix.
- For any positive integer $m \geq 3$, there exists an (n, k) SEC Hamming code with the following properties:
 - Code length: $n = 2^m - 1$
 - Number of information bits: $k = 2^m - m - 1$
 - Number of check bits: $n - k = m$
 - Minimum distance: $d_{min} = 3$
- The H matrix is an $n \times m$ matrix with all nonzero m -tuples as its column.
- A possible H matrix for a (15, 11) Hamming code, when $m = 4$:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Hamming Codes (7)

- **Example:** A Hamming code for encoding five ($k = 5$) information bits.
 - Four check bits are required ($m = 4$). So, $n = 9$.
 - A (9, 5) code can be obtained by deleting six columns from the (15,11) code shown above.
 - The H and G matrices are:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Usage Notes

- A lot of slides are adopted from the presentations and documents published on internet by experts who know the subject very well.
- I would like to thank who prepared slides and documents.
- Also, these slides are made publicly available on the web for anyone to use
- If you choose to use them, I ask that you alert me of any mistakes which were made and allow me the option of incorporating such changes (with an acknowledgment) in my set of slides.

Sincerely,

Dr. Cahit Karakuş

cahitkarakus@gmail.com