



“ İkili Sayı Sistemi ”

Dr. Cahit Karakuş, February-2019

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?

- Number systems
 - Decimal
 - Binary
 - Bits
 - bytes
 - Hexadecimal
 - Octal
 - Numbers conversion among different systems
- Ascii code

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

- **Binary** numbers
 - Digits = {0, 1}
 - $(11010.11)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$
 $= (26.75)_{10}$
 - 1 KByte (kilo) = $2^{10} = 1,024$, 1MByta (mega) = $2^{20} = 1,048,576$,
1GByte (giga) = $2^{30} = 1,073,741,824$
- **Octal** numbers
 - Digits = {0, 1, 2, 3, 4, 5, 6, 7}
 - $(127.4)_8 = 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} = (87.5)_{10}$
- **Hexadecimal** numbers
 - Digits = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
 - $(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 = (46,687)_{10}$

Data Representation in Computer

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

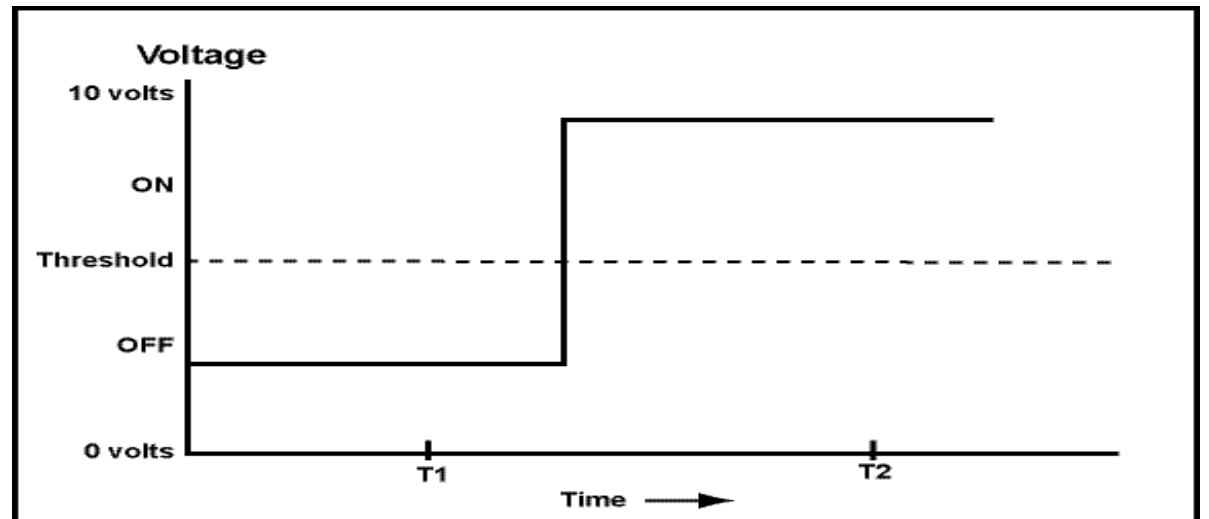
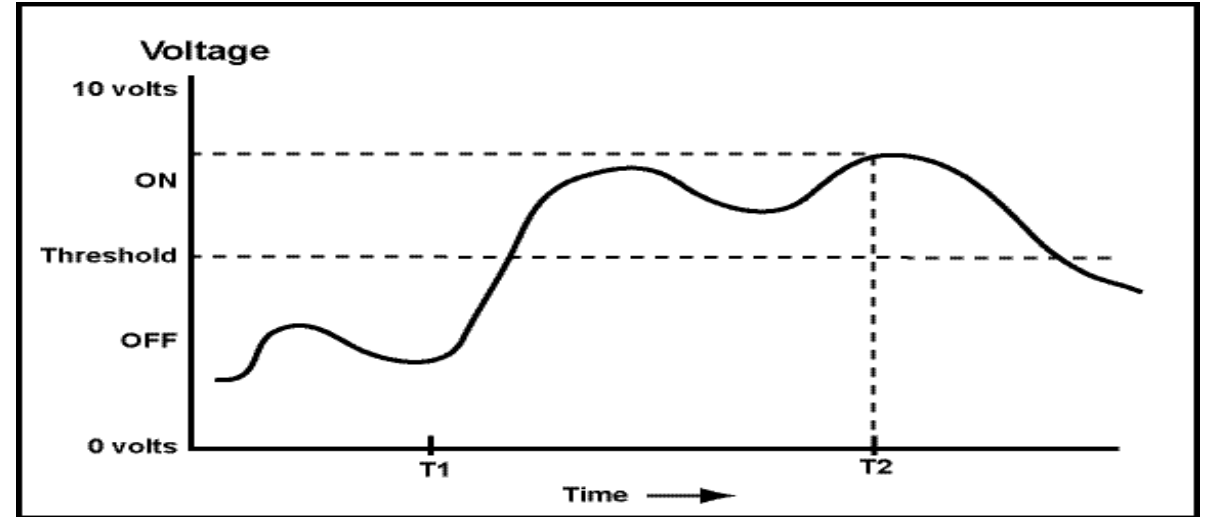
There needs a standard way

- ASCII code: **American Standard Code for Information Interchange**
 - ASCII codes represent [text](#) in [computers](#), [communications](#) equipment, and other devices that use text.
 - 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

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	□

Analog and Binary Signal

- A voltage below the *threshold*
 - off
- A voltage above the *threshold*
 - on



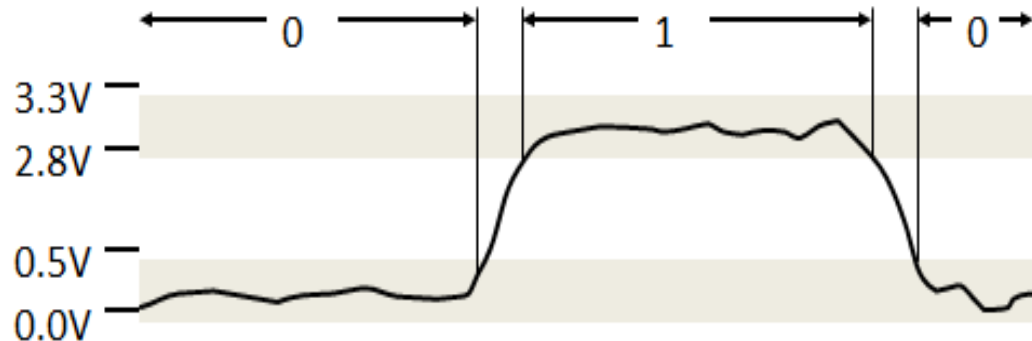
İkili Sinyal- Bit

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

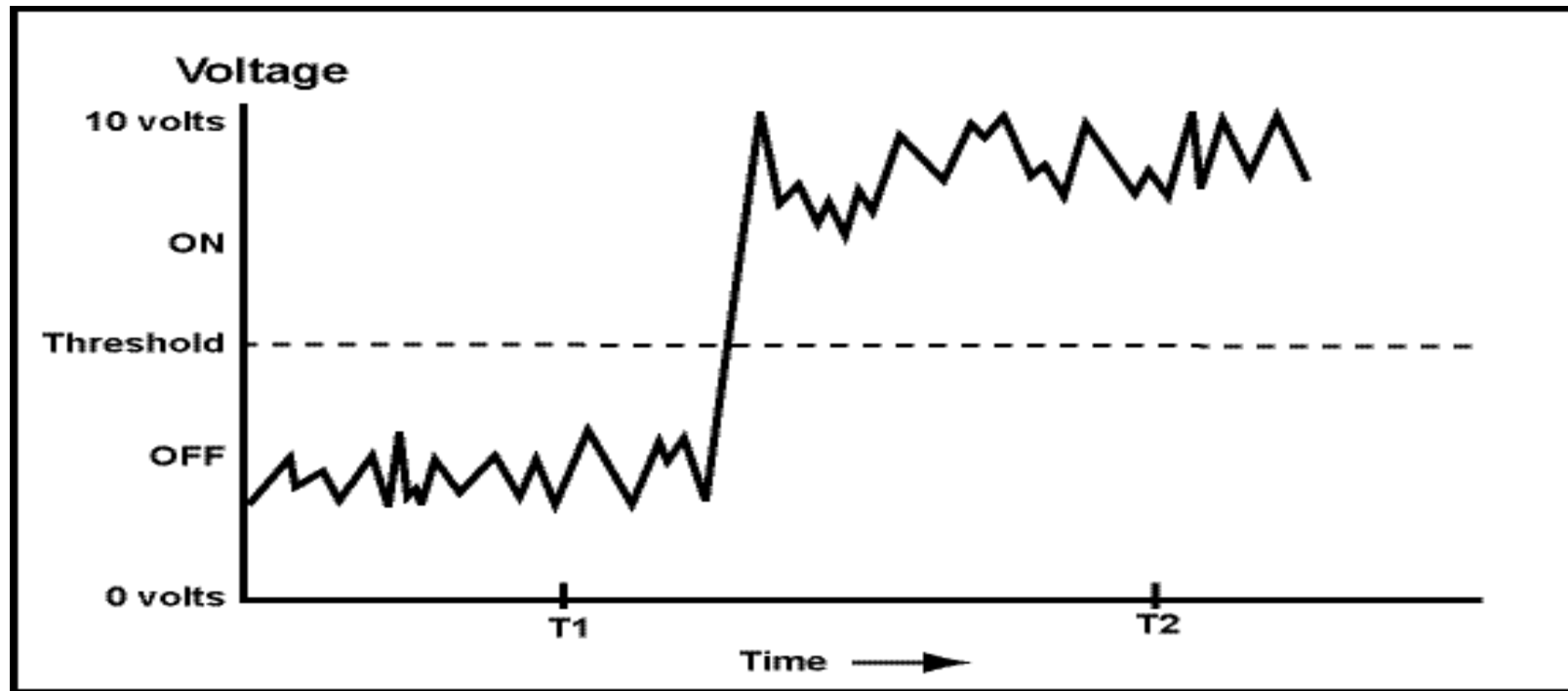
Bit: Sadece matematiksel bir kavram değil, aynı zamanda fiziksel dünyada karşılığı vardır.

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



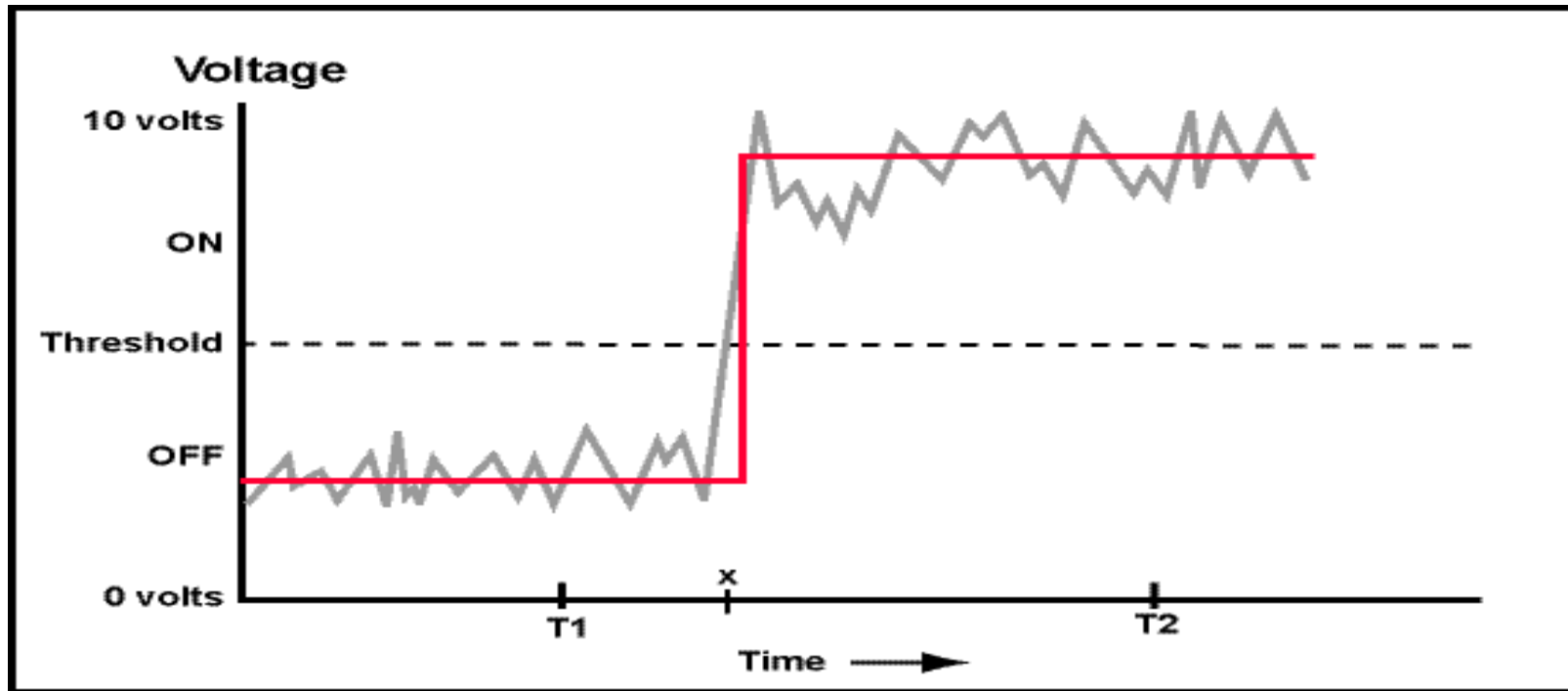
Noise on Transmission

- When the signal is transferred it will pick up noise from the environment



Recovery - Filtering

- Even when the noise is present the binary values are transmitted without error





Binary

Electronic Circuit

A circuit is an inter-connected set of electronic components that perform a function

- Integrated Circuits (ICs)
 - Combinations of thousands of circuits built on tiny pieces of silicon called chips

Why Binary Arithmetic?

$$3 + 5$$



$$= 8$$

$$0011 + 0101$$



$$= 1000$$

İKİLİ (BINARY) SAYI SİSTEMİ

- Bilgisayarlar için sadece ‘var-yok’ ‘evet-hayır’ veya ‘kapalı-açık’ ifadeleri geçerlidir.
- Bilgisayarlar sadece 0 ve 1 rakamlarından oluşan ikili sayı sistemini kullanırlar.
- Bit ismi Binary Digit, ikilik rakamdan türetilmiştir.
- 8 bite bir byte denir.
- 1 varlığı temsil eder, 0 ise yokluğu temsil eder. 1 gerilim var (5Volt), 0 ise gerilim yok demektir.
- Klavyeden yazılan rakamlar ve harfler bilgisayarın anlayacağı şekle çevrilirken ikili sayı sistemi devreye girer.
- Binary sayı sistemi, Türkçesi ikili sayı sistemi anlamına gelen kavramdır. İkili sayılar 2 tabanında yazılarak elde edilir. Dolayısı ile ikilik sistemdeki tüm sayılar 1 ve 0’dan ibarettir.
- Günümüz bilgisayarlarının neredeyse tamamında kullanılmaktadır.
- Günlük hayatımızda kullandığımız rakamlar ise onluk tabanda, bir başka ismiyle decimaldir.
- Decimal sistemi oluşturan rakamlar bildiğimiz gibi 0,1,2,3,4,5,6,7,8,9 şeklindedir.

Why Binary Arithmetic?

- Hardware can only deal with binary digits, 0 and 1.
- 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, on binary numbers.

Bit

- Bir sisteme sorulan, doğru ve yanlış olarak verilen iki yanıttan biri olan bilgidir.
- *Bir bit(0/1) bilginin temel birimidir.*
- Bit, bilgisayar sistemlerinde kullanılan bilginin temel kavramıdır, temel taşıdır.
- Bitler 0 veya 1 lerden oluşur. Birer matematiksel nesnelerdir ve birer fiziksel durumlara karşılık gelirler.
- 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.
- İki tür bit vardır, klasik bir cbit, bit (0,1) ve quantum bit, qubit ($|0\rangle$, $|1\rangle$).

Bit

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

Binary

- In Computer Systems we concentrate on binary (base 2)
 - Why?
 - Because digital components (from which the computer is built) can be in one of two states : on or off
 - We use 1 and 0 to represent these two states
 - We want to develop a method for representing information in binary
 - numbers (positive, negative, integer, floating point, fraction), strings of characters, booleans, images, sounds, programming instructions
 - For unsigned integer values, we can store them directly using binary
 - we convert from one to the other using the conversion algorithms on the previous slide where base = 2
- BIT is a unit of information equivalent to the result of a choice between only 2 possible alternatives in the binary number system.
- BYTE is a sequence of 8 bits (enough to represent one character of alphanumeric data) processed as a single unit for information.

Byte

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	Kb	2^{10} Byte
Megabyte	Mb	2^{20} Byte
Gigabyte	Gb	2^{30} Byte
Terabyte	Tb	2^{40} Byte
Petabyte	Pb	2^{50} Byte
Exabyte	Eb	2^{60} Byte
Zettabyte	Zb	2^{70} Byte
Yottabyte	Yb	2^{80} Byte

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)

Örnekler

16Gbyte kaç Byte'dır?

- $16\text{GByte} = 2^4 \cdot 2^{30} = 2^{34} \text{ Byte}$

16Gbyte kaç KByte'dır?

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

128MByte kaç Byte'dır?

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

256MByte kaç Byte'dır?

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

4KByte kaç Byte'dır?

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

32Tbyte kaç byte'dır?

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

8PByte kaç Byte'dır?

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

Multipliers for Measuring Bytes

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)
 - 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.

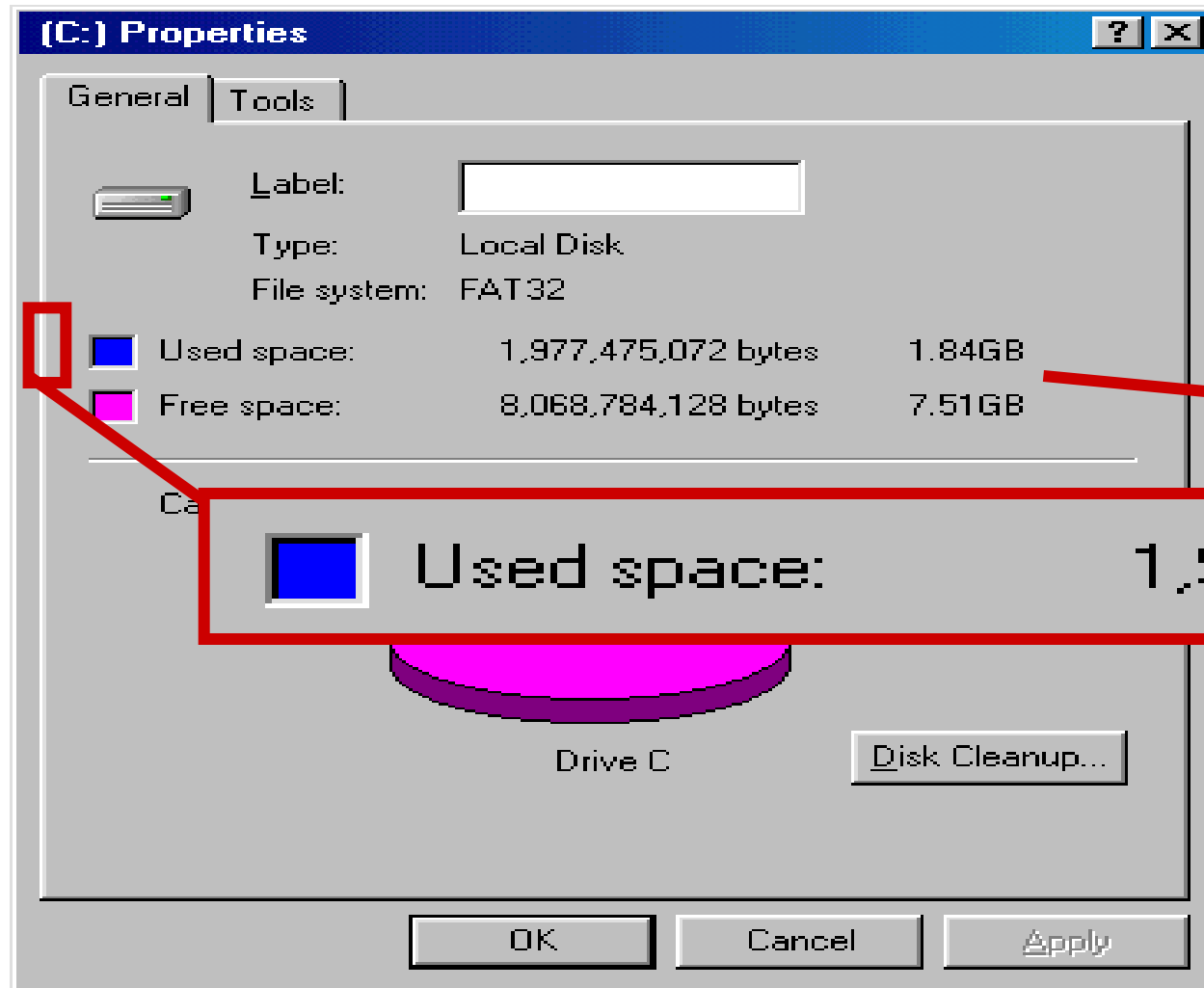
Kilobyte	Kb	2^{10} Byte
Mega Byte	Mb	2^{20} Byte
Giga Byte	Gb	2^{30} Byte
Tera Byte	Tb	2^{40} Byte
Peta Byte	Pb	2^{50} Byte
Exa	Eb	2^{60} Byte
Zetta Byte	Zb	2^{70} Byte
Yotta Byte	Yb	2^{80} Byte

Bit terimi belleğin 8 bitlik bir değerini işaretleyen ya da tanımlayan en küçük birimi olarak tanımlanmıştır. 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.

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

$$/ 2^{30} =$$

Bit, 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.

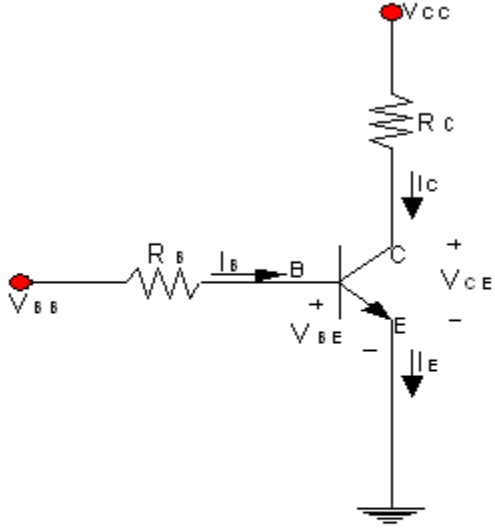
- Seconds are the units used to measure time
 - 1 millisecond (msec) = 1×10^{-3} seconds = 0.001 seconds
 - 1 microsecond (msec) = 1×10^{-6} seconds = 0.000001 seconds
 - 1 nanosecond (nsec) = 1×10^{-9} seconds = 0.000000001 seconds
- Bits per second are the units used to measure channel capacity/bandwidth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

Sayısal Mantığın Temelleri

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

Gates and Transistors



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

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

$$I_C = \beta * I_B$$

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

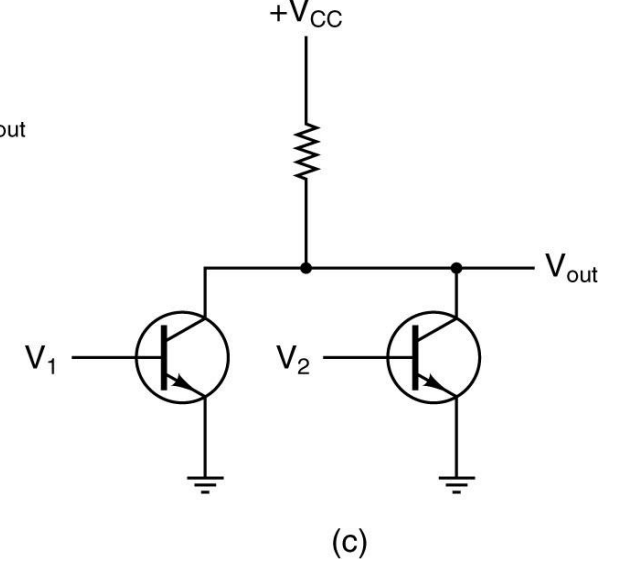
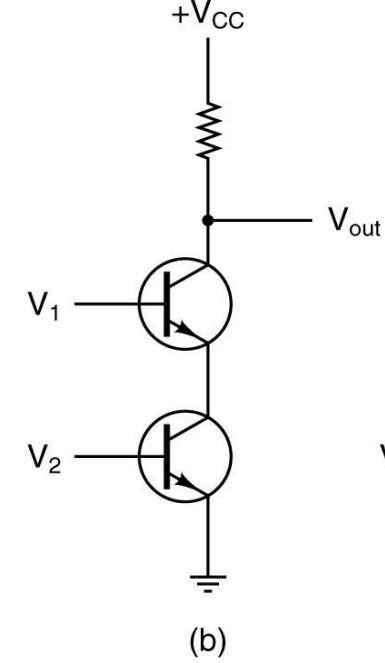
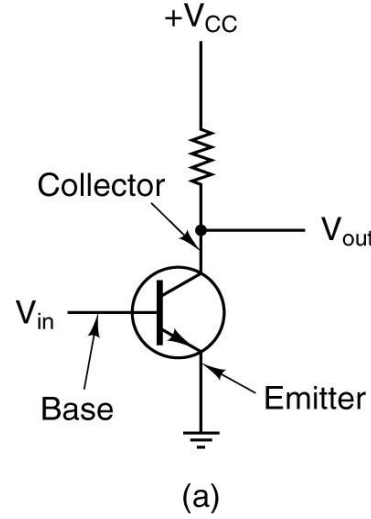
$$I_B = \frac{V_{BB} - V_{BE}}{R_B}$$

$V_{CE} \leq 0\ V$ 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.

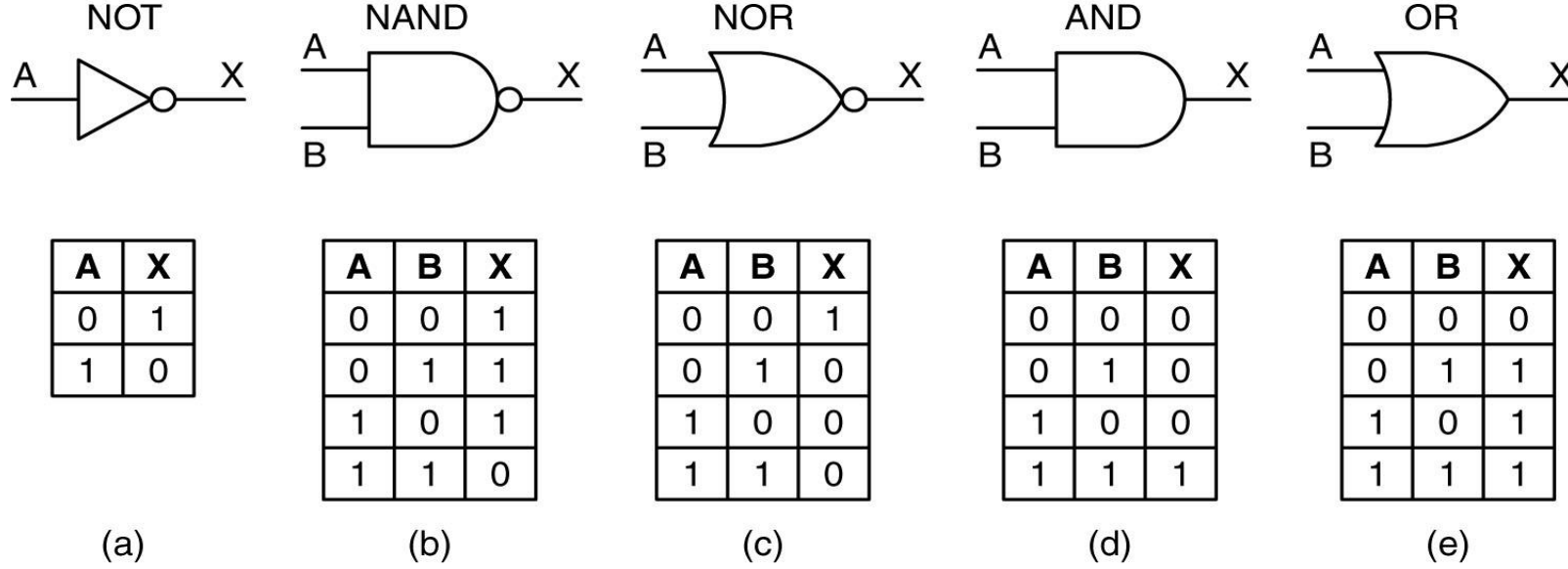


(a) A transistor inverter.

(b) A NAND gate.

(c) A NOR gate.

Mantık Kapıları için semboller ve işlevsel davranış



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

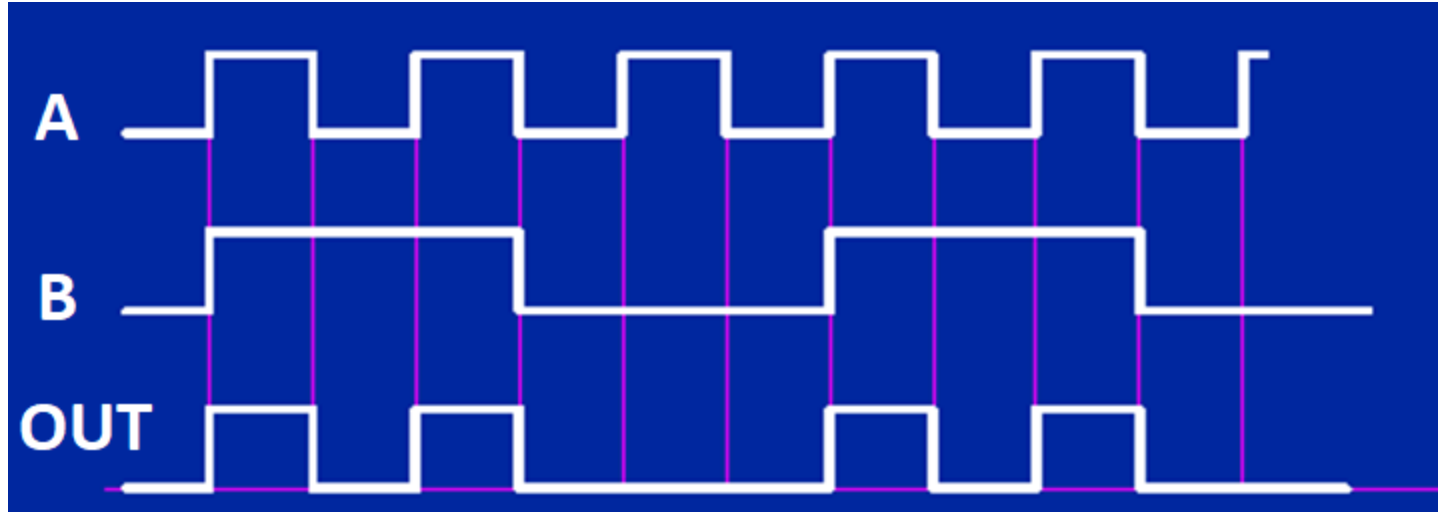
OR Kapısı: Girişlerden herhangi biri 1 ise çıkış 1 dır. Girişlerin tümü 0 ise çıkış 0 dir.

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.

Giriş ve Çıkış mantıksal seviilere bakılarak mantık kapısını belirleme

- Giriş dalga formları A ve B bir mantık kapısının iki girişine uygulandığında çıkış dalga formu belli ise bu kapının türünü belirleyiniz. (AND Kapısı)



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

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.

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

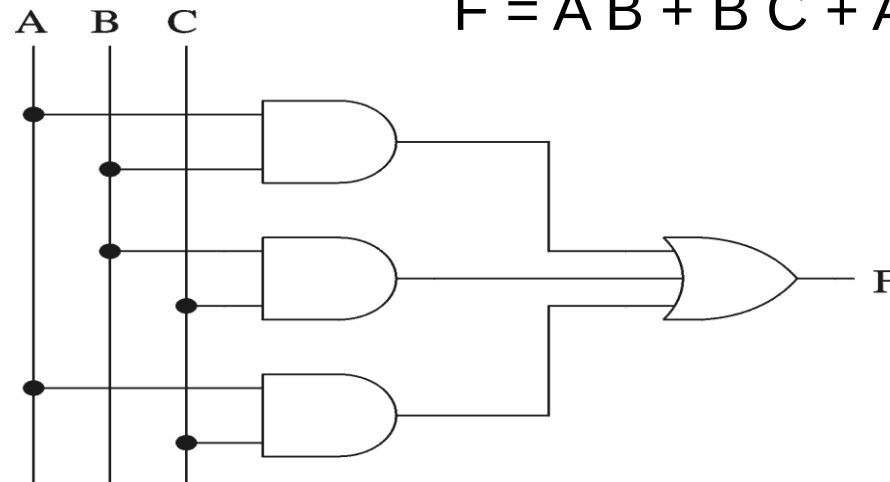
- **Logical functions can be expressed in several ways:**

- 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

Boolean Algebra

- 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. We could also say
 - $x = \text{TRUE} = 0$
 - $x = \text{FALSE} = 1$
- This is called negative logic or low-true logic. Usually we use the positive logic convention.
- 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 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.
- $A + A + A + A + A + \dots + A + 1 = 1$; 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)
- $AAA \dots A = A$
 - 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 matematiği ikili (0/1) sayı sisteminde hesaplama yapılır.

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) $0,1,2, \dots, 9$

d) Her değeri e) hiçbir değeri

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

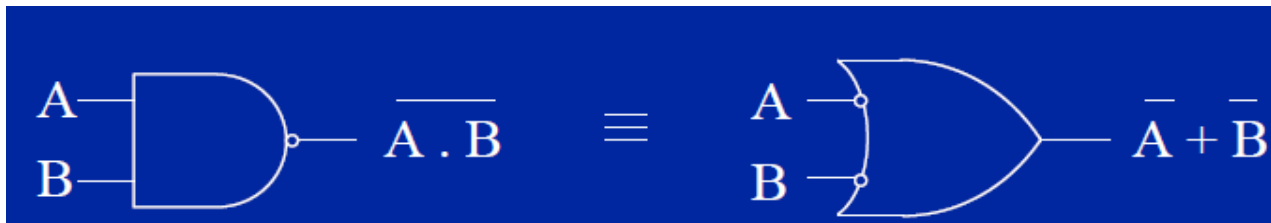
Soru: Boole Cebirinde $A*A*A*A=?$ A) A B) A^4

Boole Cebirinin Kuralları ve Yasaları

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

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

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



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

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) $a+\bar{a}=1$ Sabit Özelliği
b) $a \cdot \bar{a}=0$

9. a) $0+a=a$ Etkisizlik Özelliği
b) $1 \cdot a=a$

10. a) $1+a=1$ Yutan Sabit Özelliği
b) $0 \cdot a=0$

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

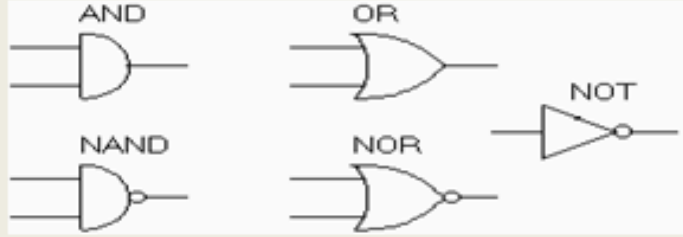
- The 12 Rules of Boolean Algebra

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

Mantıksal işlemlerin sonucu daima 1 ya da 0 dır.

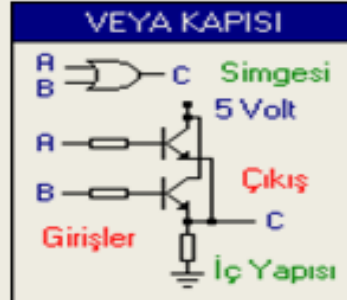
- $a + a + a + a + \dots + a = a$
- $a * a * a * \dots * a = a$
- $1 + a + b + c + \dots + z = 1$
- $ab'c + ab'c = ab'c$; benzer ifadelerin toplamı her zaman benzer bir tanesine eşittir.

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

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

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

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

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

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

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

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

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

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

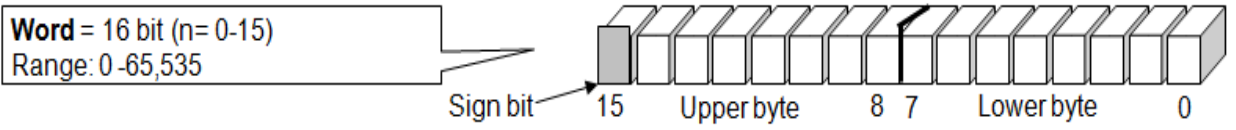
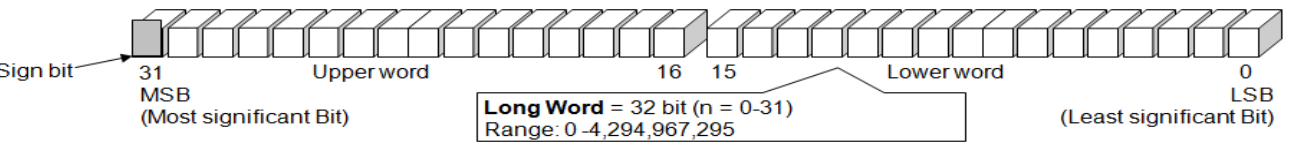


Digital Data Structure

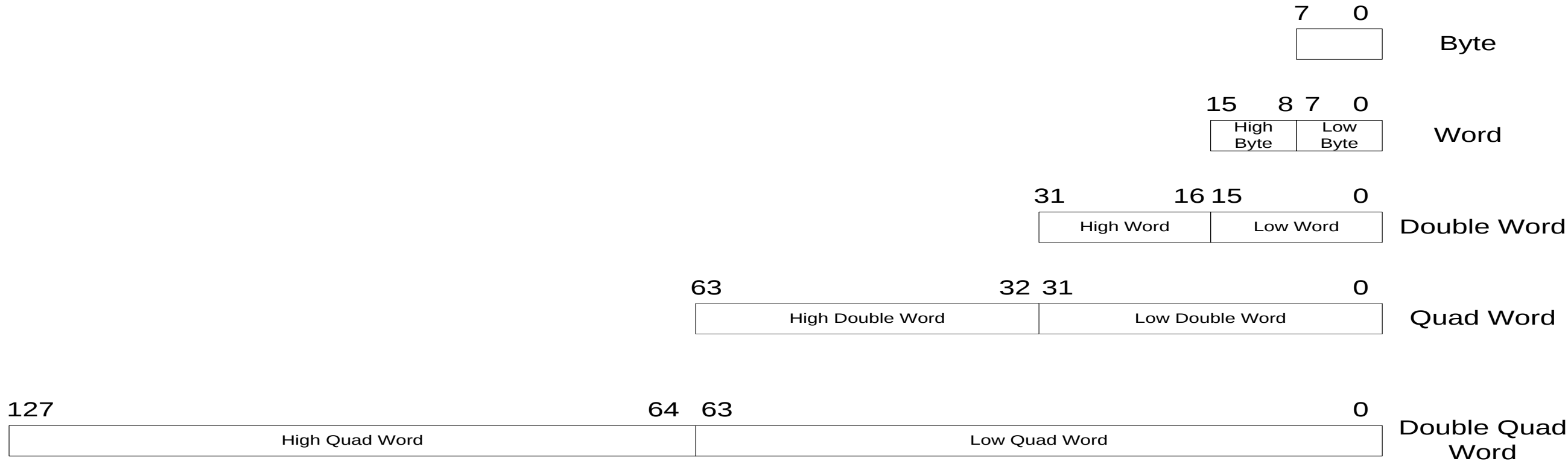
Bits, Bytes and Words

- Bilgisayarlar, mevcut elektronik teknolojimizle uygulanmasını kolaylaştırdığı için temel-2 sistemini kullanır.
- Taban-2'de çalışacak bir bilgisayar inşa etmek şu anda daha ucuz.
- Bit kelimesi, "Binary digit" kelimelerinin kısaltılmış halidir.
- Ondalık basamakların 0 ile 9 arasında değişen 10 olası değeri vardır, bitlerin yalnızca iki olası değeri vardır: 0 ve 1. Bu nedenle, bir ikili sayı, 1011 gibi yalnızca 0'lar ve 1'lerden oluşur.
- We can use the same method as base-10 to determine the value of 1011, but instead of 10 use 2: $(1 * 2^3) + (0 * 2^2) + (1 * 2^1) + (1 * 2^0) = 8 + 0 + 2 + 1 = 11$
- Bit temelinde 0 ya da 1 ile tanımlanan var/yok mantığıyla çalışan işarettir. Bir işaret, farklı sinüs dalgalarının toplamından oluşur. (Genişlik ve frekans) Frekansın temeli titreşimdir. Frekans, bir saniyedeki titreşim sayısıdır. Tersisi ise periyottur. Yani periyot bir tek titreşimin süresidir. Analog işaret, çok sayıda frekans bileşiminden oluştuğundan işareti işlemek zordur. O nedenle frekans domaininde işlenir. Frekans spektrumunda işaretin başladığı ve bittiği frekans aralığı bant genişliğini verir.
- Binary Sayıların Decimal Sayılara Dönüştürülmesi: $(100011)_2 = 2^5 + 2^1 + 2^0 = 32 + 2 + 1 = (35)_{10} = (23)_{16}$
- Ondalık Binary Sayıların Decimal Sayılara Dönüştürülmesi: $(111,101)_2 = 2^2 + 2^1 + 2^0 + 2^{-1} + 2^{-2} + 2^{-3} = 4 + 2 + 1 + 0,5 + 0,25 = 7,75$
- Decimal Sayıların Binary Sayılara Çevrilmesi: $(172)_{10} = (128 + 32 + 8 + 4)_{10} = (2^7 + 2^5 + 2^3 + 2^2)_{10} = (10101100)_2 = (AC)_{16}$
- Ondalık Decimal Sayıların Binary Sayılara Dönüştürülmesi: $(10,75)_{10} = ?$ $(10)_{10} = (1010)_2$, $2^{-1} = 1/2 = 0,5$, $2^{-2} = 1/4 = 0,25$, $(10,75)_{10} = (1010,11)_2$

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!!!!

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

Variables

TYPE	NAME	VALUE	
int	number	1	Stored only Integer
int	sum	500500	Stored only Integer
double	radius	5.5	Stored only floating-point number
double	area	95.0334	Stored only floating-point number
String	greeting	Hello	Stored only texts
String	statusMsg	Game Over	Stored only texts

A variable has a *name*, stores a *value* of the declared *type*.

Intrinsic data types

- **BYTE, SBYTE**
 - 8-bit unsigned integer; 8-bit signed integer
- **WORD, SWORD**
 - 16-bit unsigned & signed integer
- **DWORD, SDWORD**
 - 32-bit unsigned & signed integer
- **QWORD**
 - 64-bit integer
- **TBYTE**
 - 80-bit integer

Variable Declaration

- **Each variable has a type and assigned a memory address.**
- **Data-defining pseudo-ops**
 - DB define byte
 - DW define word
 - DD define double word (two consecutive words)
 - DQ define quad word (four consecutive words)
 - DT define ten bytes (five consecutive words)
- **Each pseudo-op can be used to define one or more data items of given type.**

Byte Variables

■ Assembler directive format defining a byte variable

- name DB initial value
- a question mark (“?”) place in initial value leaves variable uninitialized

■ I DB 4 define variable I with initial value 4

■ J DB ? Define variable J with uninitialized value

■ Name DB “Course” allocate 6 bytes for Name

■ K DB 5, 3, -1 allocates 3 bytes



Değişken tanımında kaç byte? Kaç adet 4 bit?

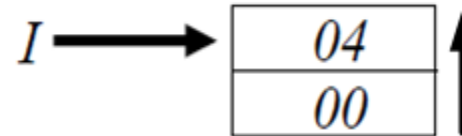
- Hex: 4 bit tanımlar.
- Örnek:
- DW: 16 bit. 2 byte, 4 adet 4 bit.

Word Variables

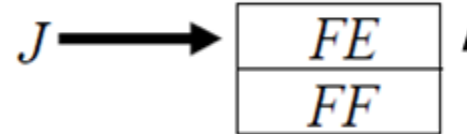
■ Assembler directive format defining a word variable

- name DW initial value

■ I DW 4



■ J DW -2



■ K DW 1ABCH



■ L DW "01"



Değişken tanımında kaç byte? Kaç adet 4 bit?

- Hex: 4 bit tanımlar.
- O yüzden 4 bir bir hex sayı sistemi ile tanımlanır.

• Örnek:

- DW: 16 bit. 2 byte, 4 adet 4 bit.

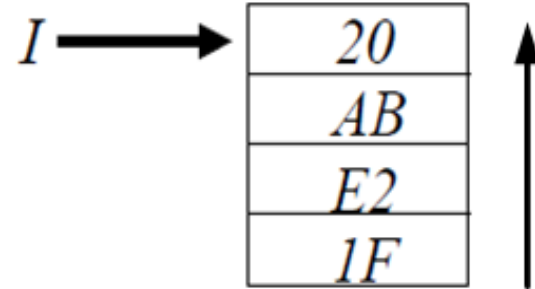
- I DW 4: (0004)h

Double Word Variables

- **Assembler directive format defining a word variable**

- name DD initial value

- **I DD 1FE2AB20H**



- **J DD -4**





Binary Addition

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

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

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

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

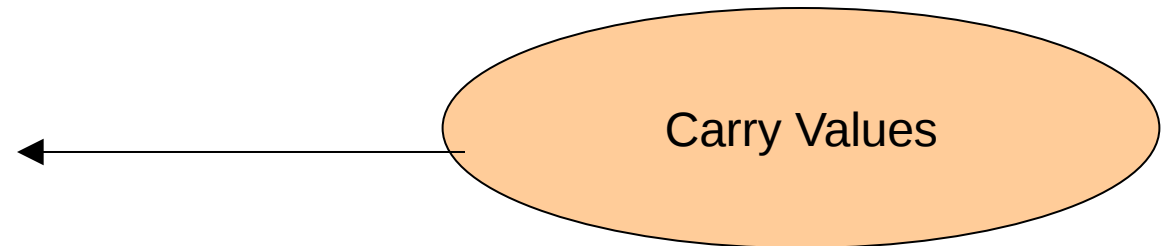
$$\begin{array}{r} 1 \quad 11 \leftarrow \text{carry} \\ 10101010111 \\ + 110000110 \\ \hline 11011011101 \end{array}$$

Arithmetic in Binary

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

1 + 1 is 0 with a carry

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



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

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

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

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

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

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:

+ 6	00000110	- 6	11111010
+13	<u>00001101</u>	+13	<u>00001101</u>
+19	00010011	+ 7	00000111
+ 6	00000110	- 6	11111010
-13	<u>11110011</u>	-13	<u>11110011</u>
- 7	11111001	-19	11101101

- $A=(1110\ 1110\ 0001)_2=(3809)_{10}=(EE1)_{16}$, $B=(1100\ 0100\ 0011)_2=(3139)_{10}=(C43)_{16}$
- $C=A+B$
- $C=(1\ 1011\ 0010\ 0100)_2=(6948)_{10}=(1B24)_{16}$

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.

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

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

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

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



Binary Conversion

Binary Numbers

- A number in a base-r system

$$X = X_{n-1}X_{n-2} \dots X_1X_0 \cdot X_{-1}X_{-2} \dots X_{-(m-1)}X_{-m}$$

$$\text{Value}(x) = x_{n-1} \cdot r^{n-1} + x_{n-2} \cdot r^{n-2} + \dots + x_0 \cdot r^0 + x_{-1} \cdot r^{-1} + x_{-2} \cdot r^{-2} + \dots + x_{-m} \cdot r^{-m}$$

$$(234.26)_6 = 2 \cdot 6^2 + 3 \cdot 6^1 + 4 \cdot 6^0 + 2 \cdot 6^{-1} + 6 \cdot 6^{-2} = (94.5)_{10}$$

$$(45.4)_8 = 4 \cdot 8^1 + 5 \cdot 8^0 + 4 \cdot 8^{-1} = (39.5)_{10}$$

Radix Number System

- Base – 2 (binary numbers)
 - 0 1
- Base – 8 (octal numbers)
 - 0 1 2 3 4 5 6 7
- Base – 16 (hexadecimal numbers)
 - 0 1 2 3 4 5 6 7 8 9 A B C D E F

Binary Number System

- Base 2
- Two Digits: 0, 1
- Example: 1010110_2
- Positional Number System

$$2^{n-1} \cdots 2^4 2^3 2^2 2^1 2^0$$
$$b_{n-1} \cdots b_4 b_3 b_2 b_1 b_0$$

- **Binary Digits** are called Bits
- Bit b_0 is the least significant bit (LSB).
- Bit b_{n-1} is the most significant bit (MSB).

Binary Sayıların Decimal Sayılara Dönüştürülmesi:

- $(100011)_2 = 2^5 + 2^1 + 2^0 = 32 + 2 + 1 = (35)_{10} = (23)_h$
- 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
- $(10,75)_{10} = ?$ $(10)_{10} = (2^3 + 2^1)_{10} = (1010)_2$, $2^{-1} = 1/2 = 0,5$ $2^{-2} = 1/4 = 0,25$, $(10,75)_{10} = (1010,11)_2$

A 4-bit binary number

2^3 (b_3)	2^2 (b_2)	2^1 (b_1)	2^0 (b_0)
8	4	2	1

MSB

Most Significant Bit

LSB

Least Significant Bit

Binary number: 0110 = $(0 \times 8) + (1 \times 4) + (1 \times 2) + (0 \times 1) = 6$

Binary number: 1101 = $8 + 4 + 1 = 13$

Binary representation

- What is the value of the binary representation **1010**?

	1	0	1	0
	↑	↑	↑	↑
position:	3	2	1	0

$$\begin{aligned}\mathbf{1010} &= \mathbf{1} \times 2^3 + \mathbf{0} \times 2^2 + \mathbf{1} \times 2^1 + \mathbf{0} \times 2^0 \\ &= \mathbf{1} \times 8 + \mathbf{0} \times 4 + \mathbf{1} \times 2 + \mathbf{0} \times 1 \\ &= 8 + 0 + 2 + 0 = \mathbf{10}\end{aligned}$$

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

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.

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

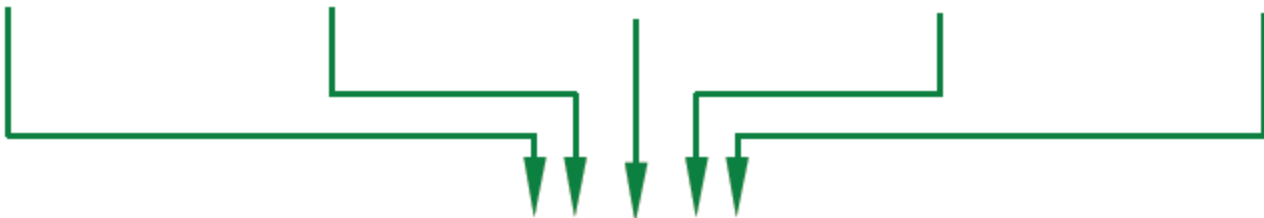
$10110_b = 22_d$

Most Significant Bit (MSB)

Least Significant Bit (LSB)

Decimal Number System

- Most computers count in binary, which we can easily understand from the decimal so ingrained in us

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


The diagram illustrates the expansion of the decimal number 35462 into its place value components. It shows the expression $3 \times 10^4 + 5 \times 10^3 + 4 \times 10^2 + 6 \times 10^1 + 2 \times 10^0$ above the number 35462. Green lines and arrows connect each term of the expression to its corresponding digit in the number: 3×10^4 connects to 3, 5×10^3 connects to 5, 4×10^2 connects to 4, 6×10^1 connects to 6, and 2×10^0 connects to 2.

35462

Decimal to Binary

- Technique
 - Divide by two, keep track of the remainder
 - First remainder is bit 0 (LSB, least-significant bit)
 - Second remainder is bit 1
 - Etc.

$$125_{10} = ?_2$$

$$125_{10} = 1111101_2$$

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 and Decimal Conversion

$2^{(7)}$	$2^{(6)}$	$2^{(5)}$	$2^{(4)}$	$2^{(3)}$	$2^{(2)}$	$2^{(1)}$	$2^{(0)}$
128	64	32	16	8	4	2	1

192.57.30.224
11000000.00111001.00011110.11100000

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

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

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

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

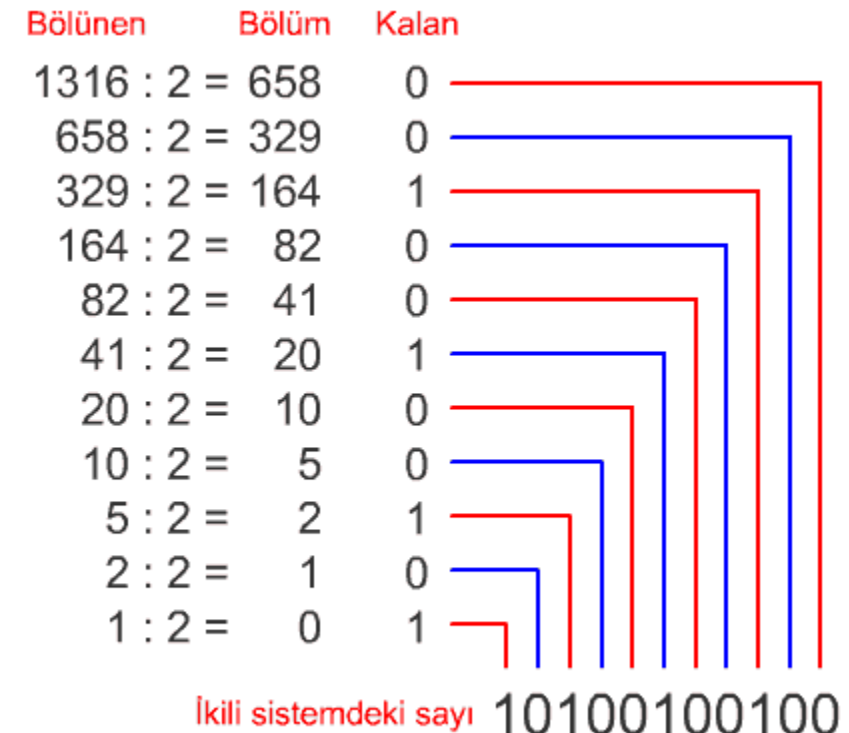
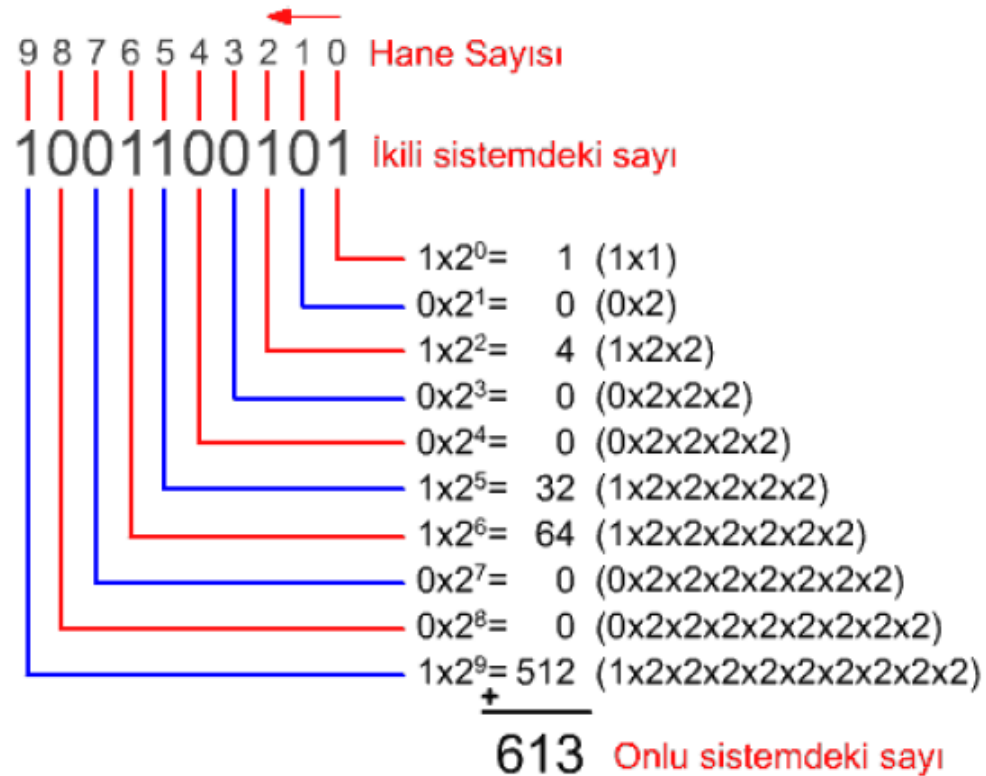
Decimal Sayıların Binary Sayılara Çevrilmesi:

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

Binary to Decimal

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

Binary sayı sisteminden decimal'a çevirme



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

- $(217)_{10} = (?)_2$
- $(217)_{10} = (11011001)_2$
- Sağlaması:
- $128 + 64 + 16 + 8 + 1 = 217$

- $(54)_{10} = (?)_2$
- $32 + 16 + 4 + 2$
- $(54)_{10} = (110110)_2$

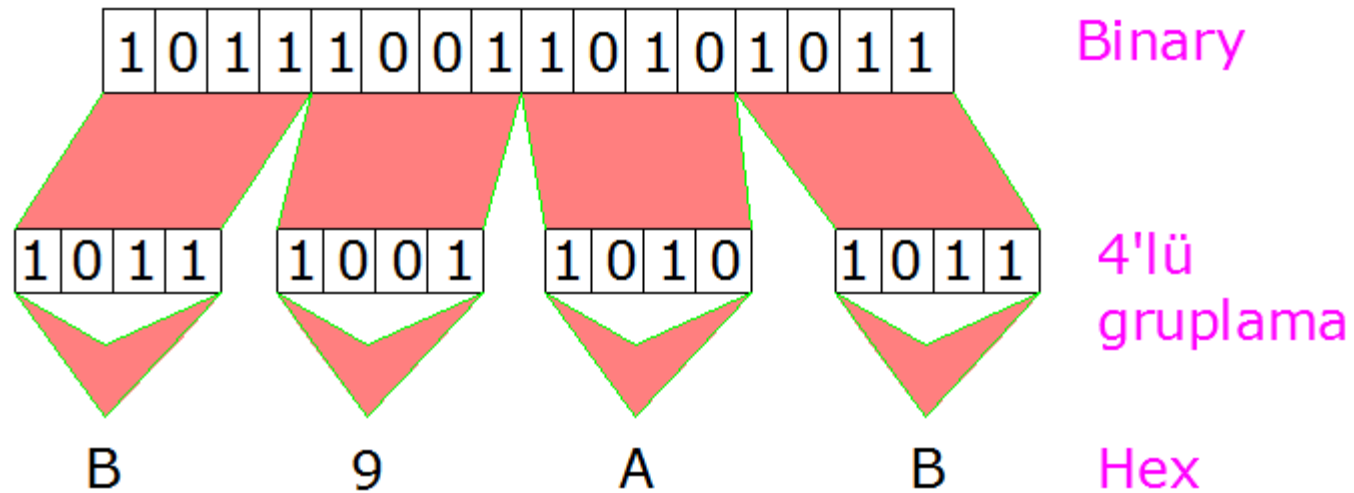


Hexa Numbering System

Heksa desimal Sayı Sistemi Nedir ?

- Binary ve desimal sistemden sonra mantık olarak aynı ancak fark olarak 16 lıktabana sahip sayılar hexa desimal sayılardır.
- Heksa desimal sistemde 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 hekza desimala çevrilip bellekte kodlanması sağlanır.
- Sayılar binary'den heksadesimale çevrilirken sağdan sola doğru dörder basamak olmak üzere gruplandırılır.
- Çünkü hekza decimal sayı sisteminin tabanı 16 dır ve binary sayı sisteminde 0-15 sayıları, 4 bit ile ifade edilebilmektedir.

Binary Sayının Hexa decimal'e Çevrilmesi



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

Hexadecimal	Binary	Decimal
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
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Binary – Hexa Numbering System

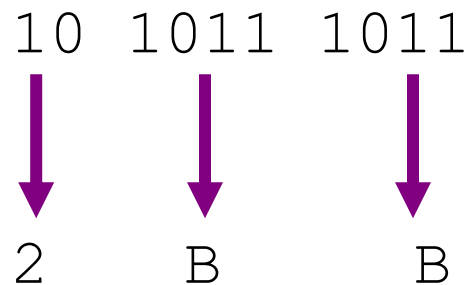
Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

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

Binary to Hexadecimal

- Technique
 - Group bits in fours, starting on right
 - Convert to hexadecimal digits

$$1010111011_2 = ?_{16}$$



$$1010111011_2 = 2BB_{16}$$

Örnek: $(100011)_2 = 2^5 + 2^1 + 2^0 = 32 + 2 + 1 = (35)_{10} = (23)_{16}$

Decimal to Binary to Hexadecimal

- Technique
 - Divide by 16
 - Keep track of the remainder

$$1234_{10} = ?_{16}$$

$$\begin{array}{r|l} 16 & 1234 \\ 16 & 77 \quad 2 \\ 16 & 4 \quad 13 = D \\ & 0 \quad 4 \end{array}$$

$$1234_{10} = 4D2_{16}$$

Hexadecimal to Binary to Decimal

- Technique
 - Multiply each bit by 16^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

$$\begin{array}{lcl} \text{ABC}_{16} \Rightarrow & \text{C} \times 16^0 = 12 \times 1 = & 12 \\ & \text{B} \times 16^1 = 11 \times 16 = & 176 \\ & \text{A} \times 16^2 = 10 \times 256 = & 2560 \\ & & 2748_{10} \end{array}$$

Hexadecimal to Binary

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

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

1	0	A	F
↓	↓	↓	↓
0001	0000	1010	1111

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

Hexa decimal sayıların toplanması

Örnek: $(21A)_{16} + (452)_{16} = (?)_{16}$

- $A + 2 = C$
- $1 + 5 = 6$
- $2 + 4 = 6$
- $(21A)_{16} + (452)_{16} = (66C)_{16}$

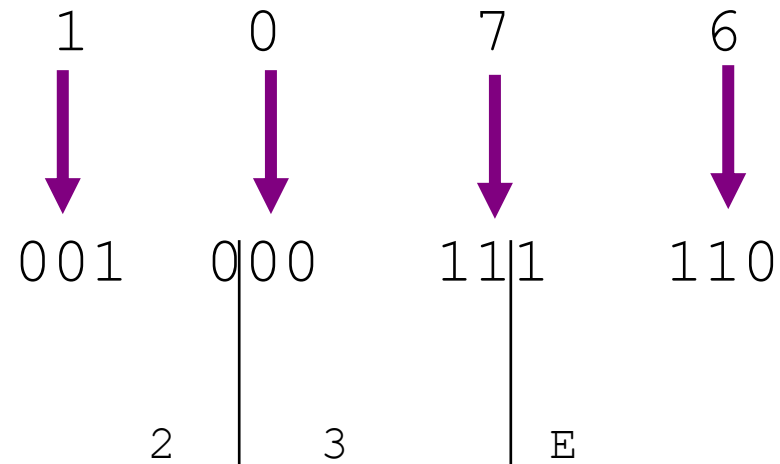
Örnek: $(73C)_{16} + (A2F)_{16} = (?)_{16}$

- $C + F = (1B)_{16}$
- $3 + 2 + 1 = 6$
- $7 + A = (11)_{16}$
- $(73C)_{16} + (A2F)_{16} = (116B)_{16}$

Octal to Hexadecimal

- Technique
 - Use binary as an intermediary

$$1076_8 = ?_{16}$$

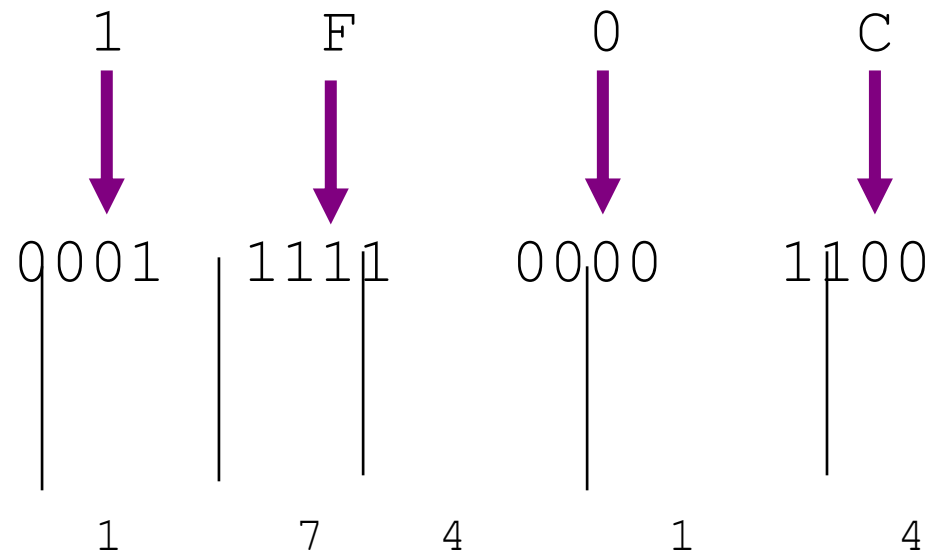


$$1076_8 = 23E_{16}$$

Hexadecimal to Octal

- Technique
 - Use binary as an intermediary

$$1F0C_{16} = ?_8$$

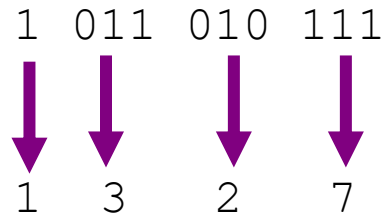


$$1F0C_{16} = 17414_8$$

Binary to Octal

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

$$1011010111_2 = ?_8$$

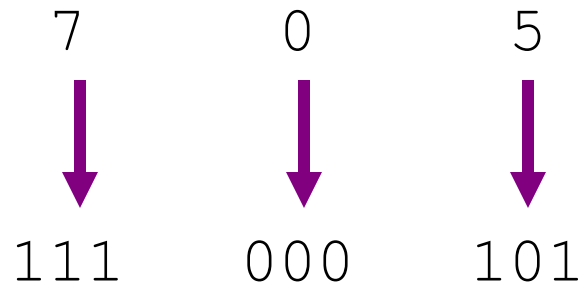


$$1011010111_2 = 1327_8$$

Octal to Binary

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

$$705_8 = ?_2$$



$$705_8 = 111000101_2$$

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.

- | | <u>Kalan</u> |
|------------------|--------------|
| • $121 / 2 = 60$ | 1 |
| • $60 / 2 = 30$ | 0 |
| • $30 / 2 = 15$ | 0 |
| • $15 / 2 = 7$ | 1 |
| • $7 / 2 = 3$ | 1 |
| • $3 / 2 = 1$ | 1 |
| • $1 / 2 = 0$ | 1 |

• Kesirli kısmı dönüştürelim.

- | | |
|-----------------------|---|
| • $0.125 * 2 = 0.250$ | 0 |
| • $0.250 * 2 = 0.500$ | 0 |
| • $0.500 * 2 = 1.000$ | 1 |

• Sonuç = 1111001.001

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
 - $1011.1001 =$
 - $1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 + 1*2^{-1} + 0*2^{-2} + 0*2^{-3} + 1*2^{-4} =$
 - $8 + 2 + 1 + \frac{1}{2} + \frac{1}{16} =$
 - $11 \frac{9}{16} = 11.5625$
 - 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

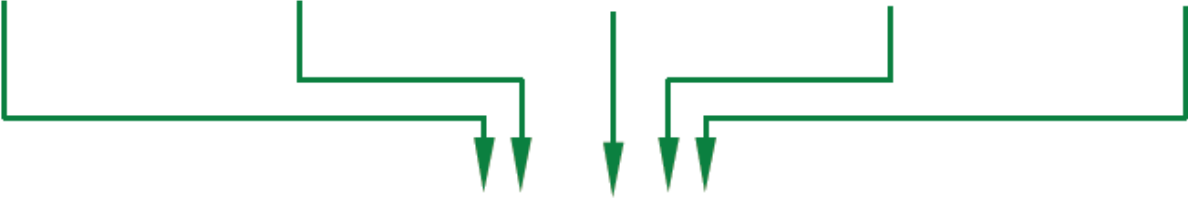
- Decimal to decimal (just for fun)

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

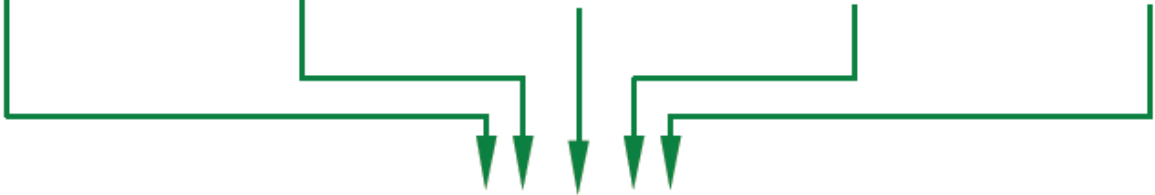
- Binary to decimal

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

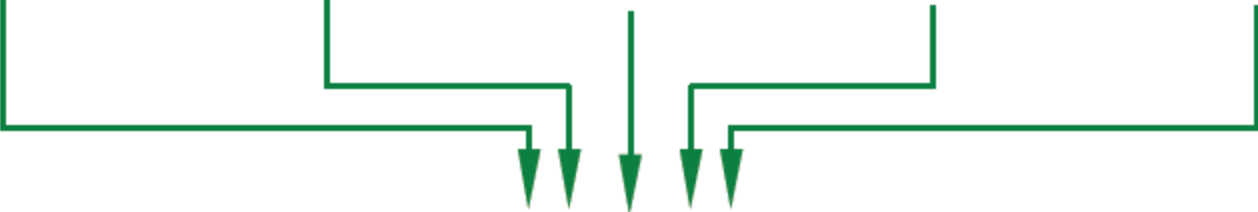
Works for Fractional Numbers too...

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


35.462

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


$10.110_b = 2.75_d$

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

$$11.011_b = 3.375_d$$

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:

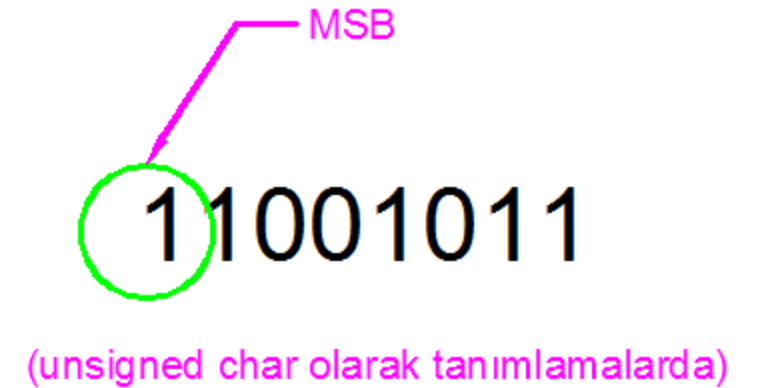
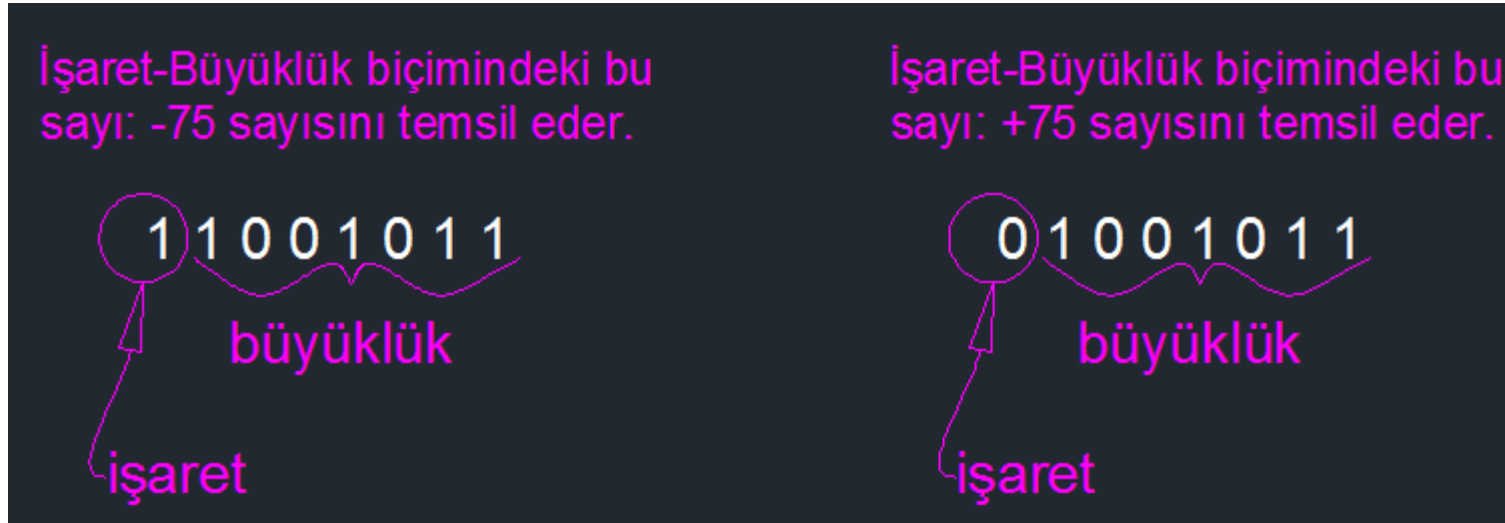
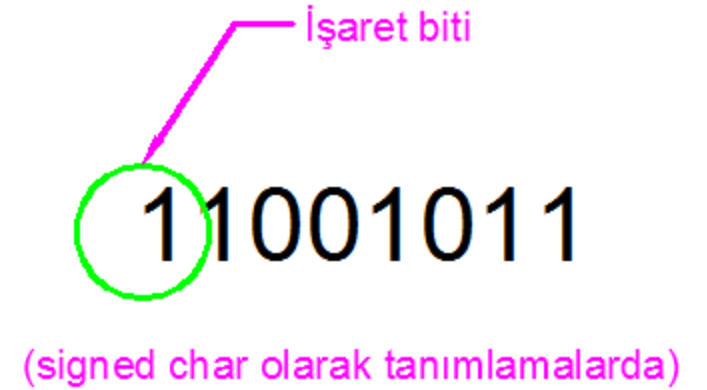
- $(100111,01)_2 = (?)_{10}$

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

Twos Complement Representation

İşaretli binary sayılar

- İkili sayı sistemlerinde 1 byte ile (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.



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

ÖRNEK: +19 ve -19 decimalsayılarını işaret-büyüklik formunda 1 bytebinarysayı ile ifade ediniz.

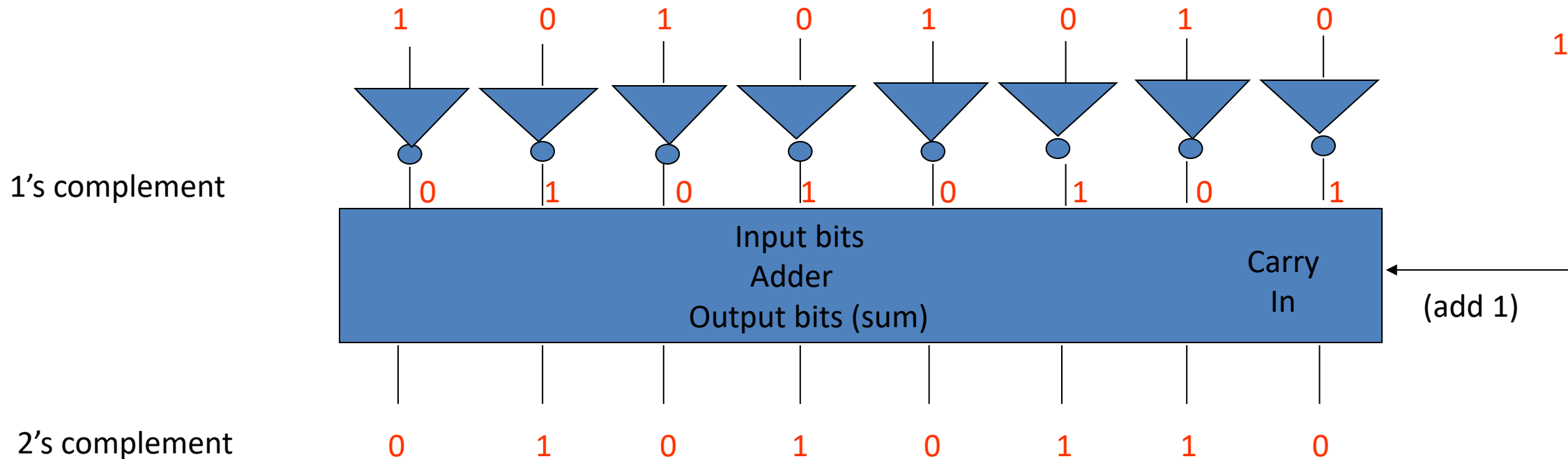
- Çözüm:
- $(19)_{10} = (10011)_2$
- $+19 = 00010011$
- $-19 = 10010011$

ÖRNEK: +35 ve -35 decimalsayılarını işaret-büyüklik formunda 1 bytebinarysayı ile ifade ediniz.

- Çözüm:
- $(35)_{10} = (100011)_2$
- $+35 = 00100011$
- $-35 = 10100011$

Complements of Binary Numbers

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



Negatif Sayıların ikil sayı sistemine çevrilmesi

- Önce negatif işareti olmadan ondalık sayıyı ikili sayı sistemine çevirin.
- En sağdan başlayarak dörterli gruplara ayırın.
- En soldaki bit 1 ise 0'lar ilave ederek dörtlü gruba tamamlayın ya da sıfırlardan oluşan yeni bir dörtlü grup ilave edin.
- Elde ettiğiniz ikili sayı sisteminde 1 yerine 0; 0 yerine 1 yazın.
- Son olarak yeni ikili sayı sistemini +1 ile toplayın.

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

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

Negatif sayılar

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

Negatif sayının ikili sistemde gösterimi

- Data tipine göre (byte, word,...), önce sayı pozitif olark ikili sayı sistemine çevrilir.
- Bitset tersi alınır
- 1 ile toplanır.

Örnek: -5

0000 0101

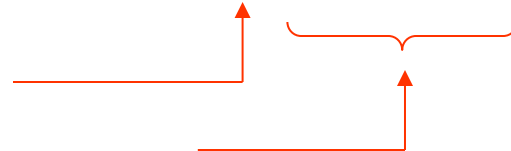
1111 1010 +1 =1111 1011= (FB)h

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

01111101	126
+ 00111010	+ 58
10110111	183

Sign Incorrect
Magnitude Incorrect





Coding

Kodlama

- 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.

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.

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

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)

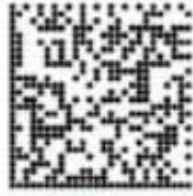
- Karekod, kare veya dikdörtgen biçimlerde basılabilen 2 boyutlu barkodun ismidir. Kare veya dikdörtgen şeklinde olan bu yapının genel adı ise Data Matrixtir.
- Karekodkelimesi ilk olarak, Beşeri Tıbbi Ürünler Barkod Uygulama Kılavuzunda kullanılmıştır.
- Ülkemizdeki ilk uygulama alanı, ilaç sektörüdür.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



QR-Code



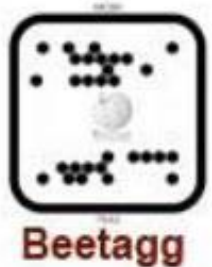
DataMatrix



Cool-Data-Matrix



Aztec



Beetagg



Trillcode



Quickmark



Shotcode



mCode





ASCII Coding

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.

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	+

ASCII Character Codes (1)

Hex	Name	Meaning	Hex	Name	Meaning
0	NUL	Null	10	DLE	Data Link Escape
1	SOH	Start Of Heading	11	DC1	Device Control 1
2	STX	Start Of Text	12	DC2	Device Control 2
3	ETX	End Of Text	13	DC3	Device Control 3
4	EOT	End Of Transmission	14	DC4	Device Control 4
5	ENQ	Enquiry	15	NAK	Negative Acknowledgement
6	ACK	ACKnowledgement	16	SYN	SYNchronous idle
7	BEL	BELl	17	ETB	End of Transmission Block
8	BS	BackSpace	18	CAN	CANcel
9	HT	Horizontal Tab	19	EM	End of Medium
A	LF	Line Feed	1A	SUB	SUBstitute
B	VT	Vertical Tab	1B	ESC	ESCape
C	FF	Form Feed	1C	FS	File Separator
D	CR	Carriage Return	1D	GS	Group Separator
E	SO	Shift Out	1E	RS	Record Separator
F	SI	Shift In	1F	US	Unit Separator

The ASCII Character set: characters 0 – 31.

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.

American Standard Code for Information Interchange (ASCII)

$b_4b_3b_2b_1$	$b_7b_6b_5$							
	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	—	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	—	o	DEL

Characters and Other Codes

- **ASCII** (American Standard Code for Information Interchange)
 - Most widely used character code.
 - See Table 1.11 for 7-bit ASCII code.
 - The eighth bit is often used for error detection (parity bit)
 - **Example:** ASCII code representation of the word **Digital**

Character	Binary Code	Hexadecimal Code
D	1000100	44
i	1101001	69
g	1100111	67
i	1101001	69
t	1110100	74
a	1100001	61
l	1101100	6C



Error Detection Codes and Correction Codes

Error Detection Codes and Correction Codes

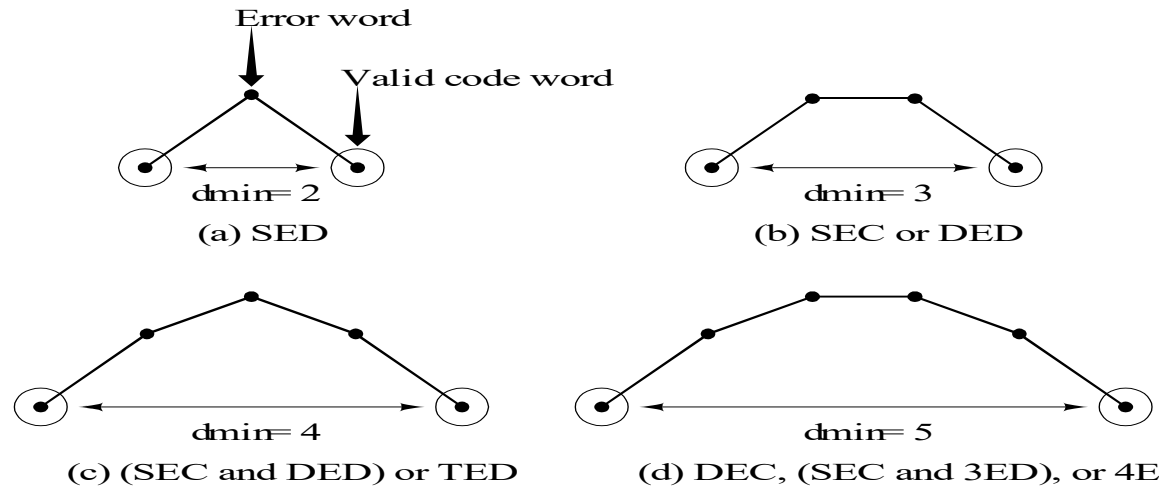
- **An error:** An incorrect value in one or more bits.
- **Single error:** An incorrect value in only one bit.
- **Multiple error:** One or more bits are incorrect.
- Errors are introduced by hardware failures, external interference (noise), or other unwanted events.
- **Error detection/correction code:** 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**

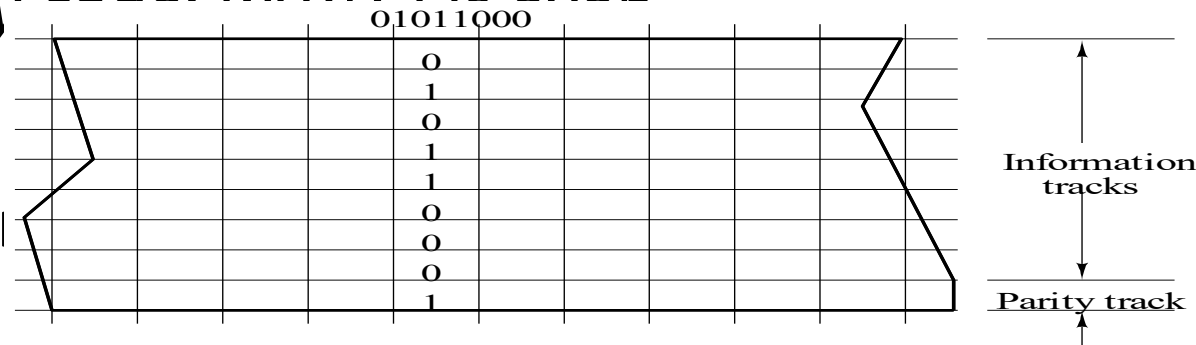
- Concatenate (|) a *parity bit*, P , to each code word of C .



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

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

- Parity code



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:

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

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

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