

# LOJİK DEVRE TASARIMI

Prof. Dr. Ahmet SERTBAŞ  
asertbas@istanbul.edu.tr

İstanbul Üniversitesi Bilgisayar Mühendisliği Bölümü  
2017-2018 Güz Yarıyılı

## DEĞERLENDİRME ÖLÇÜTÜ

Vize Sınavı	1	%30
Proje	1	%20
Yılıçi		%50
Final	1	%50

## Dersin Kaynakları:

1. ***'Digital Design'***, 5. baskı, M. Morris Mano - Micheal D. Ciletti,  
Pearson Yayınevi,2012. ***'Türkçe Baskı: Sayısal Tasarım'***

# DERSİN İÇERİĞİ:

1. Sayısal İşaret ve Sistemlere Giriş  
Sayı sistemleri ve Sayı Sistemi Dönüşümleri
2. Boolean Cebir Tanımlamaları, Boolean Cebirinin temel özellikleri,  
aksiyomları, eşitlikleri vs..
3. Boolean Fonksiyonları ve Özellikleri, Kanonik Fonksiyon Gösterimleri
4. Boolean Fonksiyonlarını Sadeleştirme Yöntemleri: Karnaugh Haritası ve Tablo (Quin-Mc Cluskey) Yöntemleri
5. Temel Sayısal Mantık Kapıları tanımlamaları ve devreleri
6. Belleksiz Mantık (Kombinezonsal) Devresi Tasarım Prosedürü ve örnek tasarımlar  
Toplayıcılar, Çıkarıcı, Çoğullayıcılar, Kodçözücüler, Kodlayıcılar
7. Programlanabilir Mantık Devreleri: PAL, PLA, PROM, EEPROM
8. Saatli Bellekli Mantık (Ardışıl) devrelerinin Tanımlanması ve Analizi  
Durum Graf ve Tablolarının Elde Edilmesi, Durum Tabloları ve Durum Kodlamaları
9. Bellekli Mantık (Ardışıl) Devre Tasarım prosedürü ve örnek tasarımlar
10. Tutucular, Flip-Floplar
11. Sayıcı Tasarımları
12. Yazmaç (Register) Tasarımı, Ötelemeli yazmaçlar.....

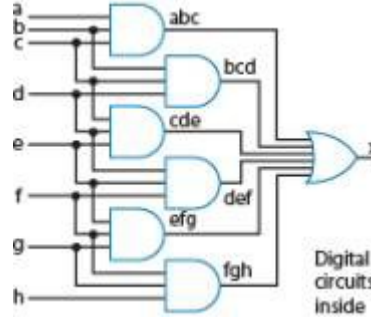
# GİRİŞ: Temel Kavramlar

- 1.1 Sayısal Sistemler
- 1.2 İkili Sayılar
- 1.3 Sayı Taban Dönüşümleri
- 1.4 Sekizli (Octal) ve Onaltılı (Hexadecimal ) Sayılar
- 1.5 Tümleme
- 1.6 İşaretli İkili Sayılar
- 1.7 İkili Kodlar
- 1.8 İkili Bilgi Saklama ve Yazmaçlar (Registers)
- 1.9 İkili Mantık

# Sayısal Sistemler

- Sayısal Çağ ve Bilgi Çağı
- Sayısal Bilgisayarlar
  - ◆ Genel amaçlı
  - ◆ Çoğu bilimsel, sanayi ve ticari uygulamalar
- Sayısal Sistemler
  - ◆ Telefon anahtarlama sistemleri
  - ◆ Sayısal Kamera
  - ◆ Elektronik hesaplayıcılar
  - ◆ Sayısal TV

# Sayısal Sistemler

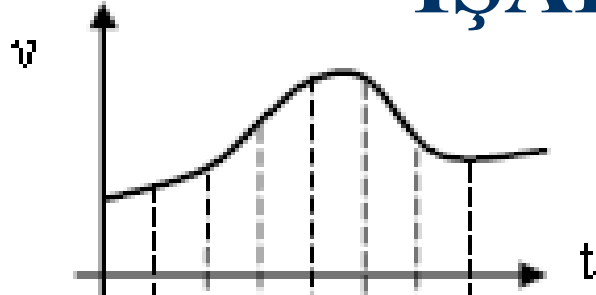


- 1-) Daha güvenilirdir.
- 2-) Devreler ve sistemler aynen tekrarlanabilir (Her benzer sistem tıpatıp aynen çalışır).
- 3-) Sinyal kalitesi değişmez. Bu kalite istenildiği kadar iyi yapılabilir.
- 4-) Gürültü ve dış etkilerden çok az etkilenir.
- 5-) Daha ucuzdur (Pek çok uygulamada).
- 6-) Kopyalama ve iletim sırasında bozulmaz. (İlk kopya ile yüzüncü kopyanın kalitesi aynıdır)

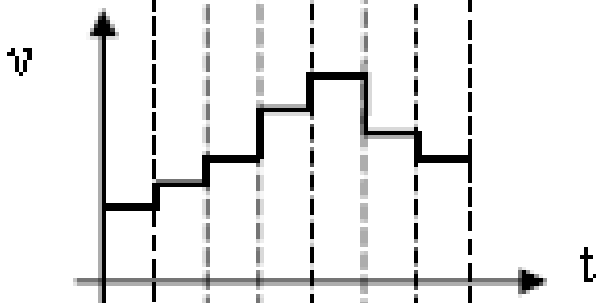


*Elektronik elemanlar gittikçe sayısal eleman haline gelmektedir.  
Donanımı bilmeden etkin yazılım becerisi kazanılması zordur.*

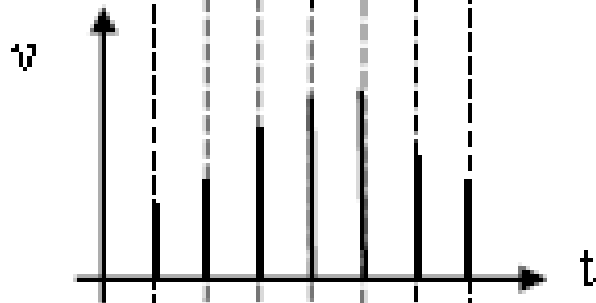
# İŞARET TÜRLERİ



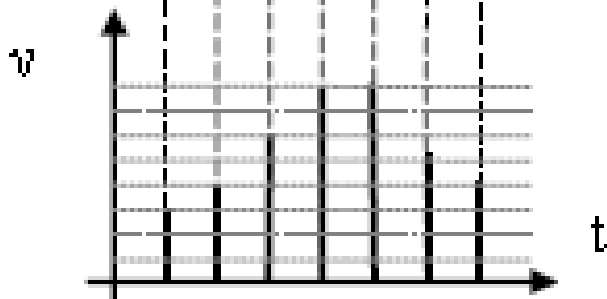
Sürekli Zaman - Analog İşaret



Sürekli Zaman - Kuantalanmış İşaret



Ayrık Zaman - Örneklenmiş İşaret

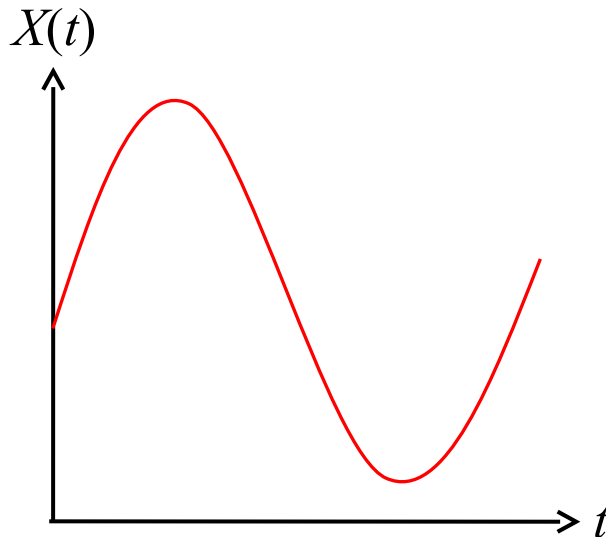


Ayrık Zaman - Sayısal İşaret  
(Örneklenmiş ve Kuantalanmış)

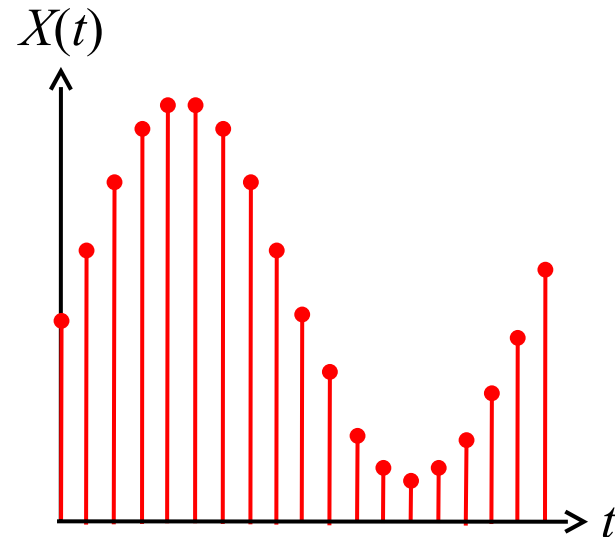


# Analog ve Sayısal İşaretler

- Analog sistem
  - ◆ Fiziksel büyüklükler ve işaretler belirli bir bölgede sürekli olarak değişebilirler.
- Sayısal sistem
  - ◆ Fiziksel büyüklükler ve işaretler sadece ayırık değerlerde var oldukları kabul edilir.



Analog işaret



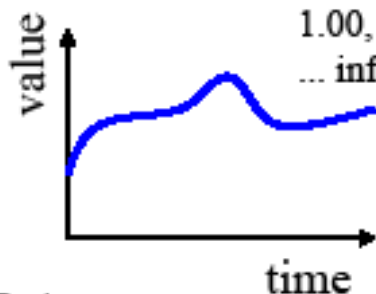
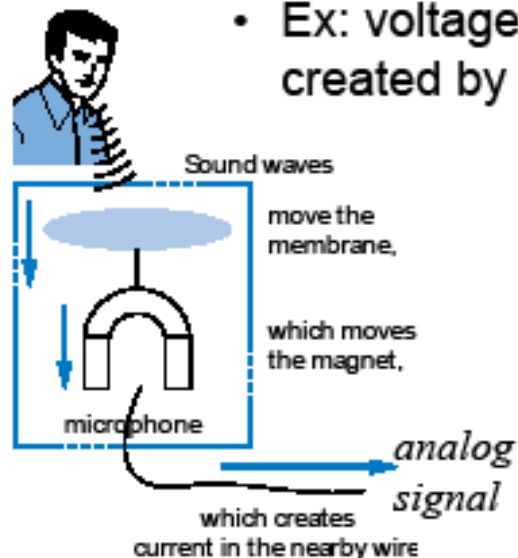
Sayısal İşaret

# What Does “Digital” Mean?

- Analog signal

- Infinite possible values

- Ex: voltage on a wire created by microphone

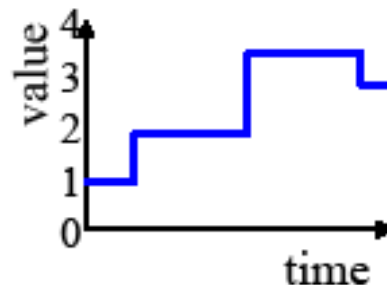
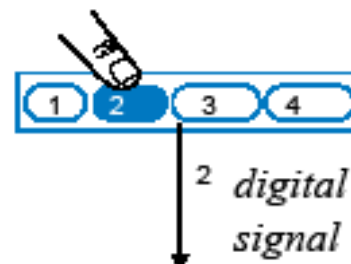


Possible values:  
1.00, 1.01, 2.0000009,  
... infinite possibilities

- Digital signal

- Finite possible values

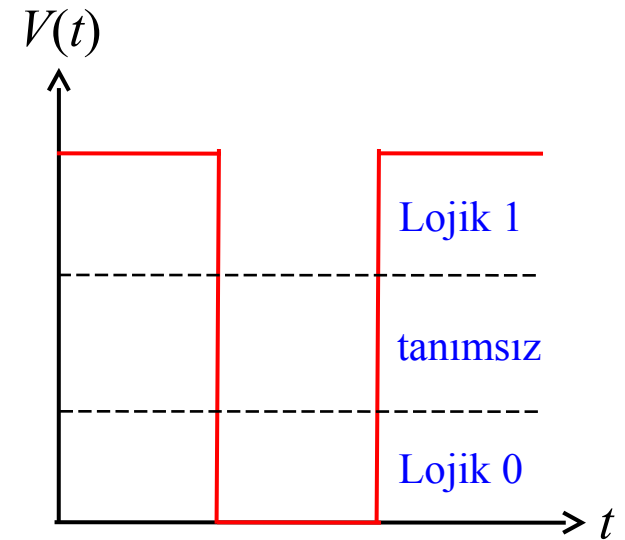
- Ex: button pressed on a keypad



Possible values:  
0, 1, 2, 3, or 4.  
That's it.

# İkili Sayısal İşaret

- Sayısal Sistemlerde değişkenler ayrık değerler alabilir.  
İkili seviye, ikili değişken
- İkili değerler kısa olarak aşağıdaki gibi simgelenir:
  - ◆ Dijit 0 ve Dijit 1
  - ◆ Kelime (sembol) : False (F) ve True (T)
  - ◆ Kelime (sembol) : Low (L) ve High (H)
  - ◆ On ve Off



İkili sayısal işaret

# Onlu Sayı Sistemi

- Base (also called radix) = 10
  - ◆ 10 digits { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }
- Digit Position
  - ◆ Integer & fraction
- Digit Weight
  - ◆  $\text{Weight} = (\text{Base})^{\text{Position}}$
- Magnitude
  - ◆ Sum of “*Digit x Weight*”
- Formal Notation

2	1	0		-1	-2
5	1	2	.	7	4
100	10	1		0.1	0.01
			.		
500	10	2		0.7	0.04

$$d_2 * B^2 + d_1 * B^1 + d_0 * B^0 + d_{-1} * B^{-1} + d_{-2} * B^{-2}$$

$$(512.74)_{10}$$



# Sekizli Sayı Sistemi

- Base = 8
  - ◆ 8 digits { 0, 1, 2, 3, 4, 5, 6, 7 }
- Weights
  - ◆  $\text{Weight} = (\text{Base})^{\text{Position}}$
- Magnitude
  - ◆ Sum of “*Digit x Weight*”
- Formal Notation

64	8	1		1/8	1/64
<div style="border: 1px solid blue; padding: 5px; display: inline-block; background-color: yellow;">5</div>	<div style="border: 1px solid blue; padding: 5px; display: inline-block; background-color: yellow;">1</div>	<div style="border: 1px solid blue; padding: 5px; display: inline-block; background-color: yellow;">2</div>	●	<div style="border: 1px solid blue; padding: 5px; display: inline-block; background-color: yellow;">7</div>	<div style="border: 1px solid blue; padding: 5px; display: inline-block; background-color: yellow;">4</div>
2	1	0		-1	-2

$$5 * 8^2 + 1 * 8^1 + 2 * 8^0 + 7 * 8^{-1} + 4 * 8^{-2}$$
$$=(330.9375)_{10}$$
$$(512.74)_8$$

# İkili Sayı Sistemi

- Base = 2
  - ◆ 2 digits { 0, 1 }, called *binary digits* or “*bits*”

- Weights
  - ◆ Weight =  $(Base)^{Position}$

- Magnitude
  - ◆ Sum of “*Bit x Weight*”

- Formal Notation

- Groups of bits
  - 4 bits = *Nibble*
  - 8 bits = *Byte*

4	2	1		1/2	1/4
<b>1</b>	<b>0</b>	<b>1</b>	•	<b>0</b>	<b>1</b>
2	1	0		-1	-2

$$1 * 2^2 + 0 * 2^1 + 1 * 2^0 + 0 * 2^{-1} + 1 * 2^{-2}$$
$$=(5.25)_{10}$$
$$(101.01)_2$$

**1 0 1 1**

**1 1 0 0 0 1 0 1**

# Onaltılı Sayı Sistemi

- Base = 16
  - ◆ 16 digits { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F }
- Weights
  - ◆ Weight =  $(Base)^{Position}$
- Magnitude
  - ◆ Sum of “*Digit x Weight*”
- Formal Notation

256	16	1		1/16	1/256
<b>1</b>	<b>E</b>	<b>5</b>	•	<b>7</b>	<b>A</b>
2	1	0		-1	-2

$$1 * 16^2 + 14 * 16^1 + 5 * 16^0 + 7 * 16^{-1} + 10 * 16^{-2}$$

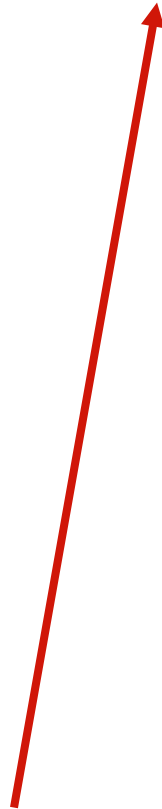
$$=(485.4765625)_{10}$$

$$(1E5.7A)_{16}$$



# 2 nin katları

n	$2^n$
0	$2^0=1$
1	$2^1=2$
2	$2^2=4$
3	$2^3=8$
4	$2^4=16$
5	$2^5=32$
6	$2^6=64$
7	$2^7=128$



n	$2^n$
8	$2^8=256$
9	$2^9=512$
10	$2^{10}=1024$
11	$2^{11}=2048$
12	$2^{12}=4096$
20	$2^{20}=1M$
30	$2^{30}=1G$
40	$2^{40}=1T$

**Kilo**

**Mega**

**Giga**

**Tera**





# Toplama

- Onlu Toplama

$$\begin{array}{r} \text{1} \quad \text{1} \quad \quad \leftarrow \text{Carry} \\ \quad 5 \quad 5 \\ + \quad 5 \quad 5 \\ \hline 1 \quad 1 \quad 0 \end{array}$$

# İkili Toplama

- Kolon Toplama

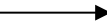
	1	1	1	1	1	1		
		1	1	1	1	0	1	= 61
+			1	0	1	1	1	= 23
<hr/>								
	1	0	1	0	1	0	0	= 84



# İkili Çıkarma

- İhtiyaç olduğunda borç alınır.

	1		2					= (10) <sub>2</sub>
	0	<del>2</del>	2	0	0	2		
	<del>1</del>	0	0	<del>1</del>	<del>1</del>	0	1	= 77
-			1	0	1	1	1	= 23
<hr/>								
	0	1	1	0	1	1	0	= 54



# İkili Çarpma

- Bit bit çarpma

$$\begin{array}{r} \phantom{00000}10111 \\ \times \phantom{000}1010 \\ \hline \phantom{00000}00000 \\ \phantom{000}10111 \\ \phantom{00}00000 \\ 10111 \\ \hline 11100110 \end{array}$$


# Onludan İkiliye Dönüşüm

- Tabanla (=2) sayıyı bölme
- Kalanı bir katsayı olarak alma (0 veya 1)
- Bölüm alınır ve bölme işlemine devam edilir

Örnek: **(13)**<sub>10</sub>

	Quotient Bölüm	Remainder Kalan	Coefficient Katsayı
<b>13</b> / 2 =	<b>6</b>	<b>1</b>	<b>a<sub>0</sub> = 1</b>
<b>6</b> / 2 =	<b>3</b>	<b>0</b>	<b>a<sub>1</sub> = 0</b>
<b>3</b> / 2 =	<b>1</b>	<b>1</b>	<b>a<sub>2</sub> = 1</b>
<b>1</b> / 2 =	<b>0</b>	<b>1</b>	<b>a<sub>3</sub> = 1</b>

Sonuç: **(13)**<sub>10</sub> = (**a<sub>3</sub> a<sub>2</sub> a<sub>1</sub> a<sub>0</sub>**)<sub>2</sub> = **(1101)**<sub>2</sub>

  
MSB                      LSB

# Onludan (Kesirli) İkiliye Dönüşüm

- Taban ile (=2) sayı çarpılır
- Tamsayı kısmı katsayı olarak alınır (0 ve 1)
- Sonucun kesir kısmı alınır ve bölmeye devam edilir

Örnek:  $(0.625)_{10}$

		Integer Tamsayı	Fraction Kesir	Coefficient Katsayı
$0.625$	$* 2 =$	$1$	$. 25$	$a_{-1} = 1$
$0.25$	$* 2 =$	$0$	$. 5$	$a_{-2} = 0$
$0.5$	$* 2 =$	$1$	$. 0$	$a_{-3} = 1$

Sonuç:  $(0.625)_{10} = (0.a_{-1}a_{-2}a_{-3})_2 = (0.101)_2$

↑  
MSB

↑  
LSB

# Onludan Sekizliye Dönüşüm

Örnek:  $(175)_{10}$

	Quotient	Remainder	Coefficient
$175 / 8 =$	<b>21</b>	<b>7</b>	$a_0 = 7$
$21 / 8 =$	<b>2</b>	<b>5</b>	$a_1 = 5$
$2 / 8 =$	<b>0</b>	<b>2</b>	$a_2 = 2$

Sonuç:  $(175)_{10} = (a_2 a_1 a_0)_8 = (257)_8$

Örnek:  $(0.3125)_{10}$

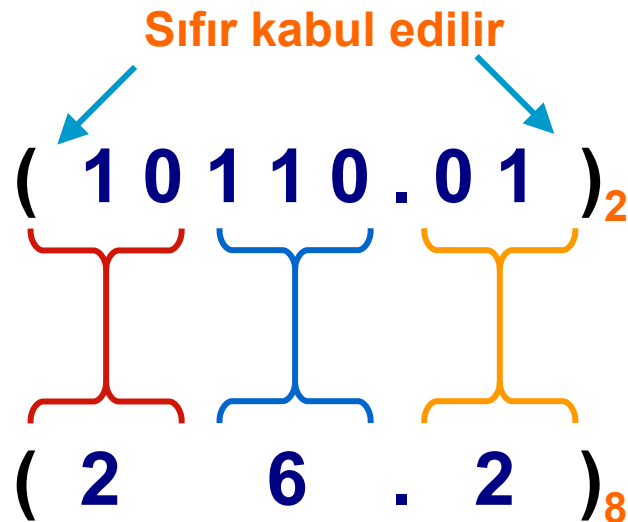
	Integer	Fraction	Coefficient
$0.3125 * 8 =$	<b>2</b>	<b>5</b>	$a_{-1} = 2$
$0.5 * 8 =$	<b>4</b>	<b>0</b>	$a_{-2} = 4$

Sonuç:  $(0.3125)_{10} = (0.a_{-1} a_{-2} a_{-3})_8 = (0.24)_8$

# İkili-Sekizli Dönüşüm

- $8 = 2^3$
- Her 3 bitlik grup bir sekizli (octal) digit belirler..

**Örnek:**



Sekizli	İkili
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

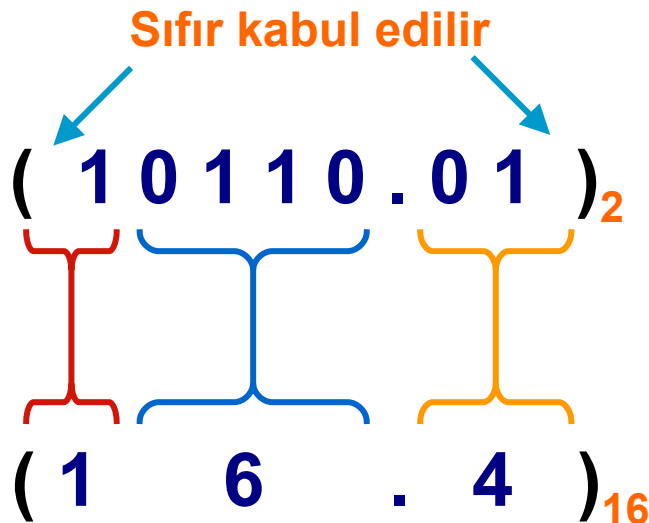
**Her iki yönde de dönüşüm çalışır. (İkili-Sekizli ve Sekizli-İkili)**



# İkili-Onaltılı Dönüşüm

- $16 = 2^4$
- Each group of 4 bits represents a hexadecimal digit

**Example:**



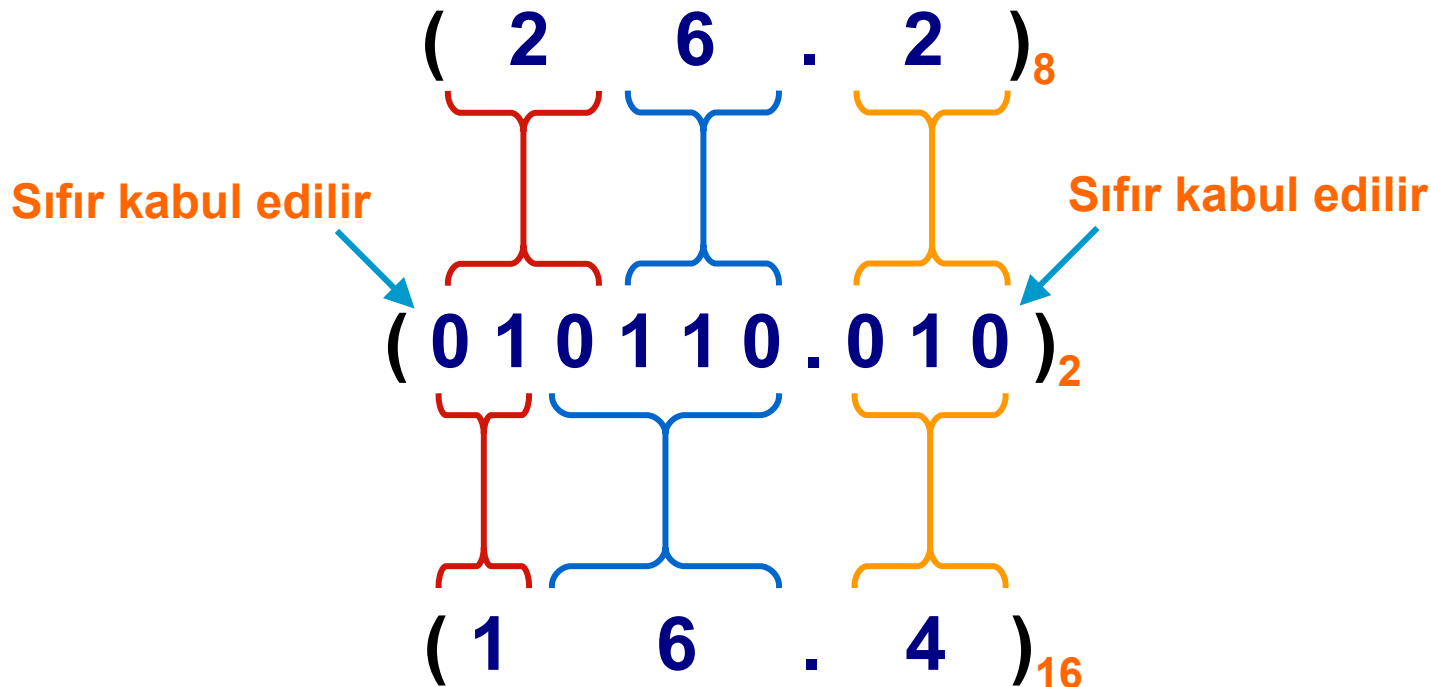
Hex	Binary
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
A	1 0 1 0
B	1 0 1 1
C	1 1 0 0
D	1 1 0 1
E	1 1 1 0
F	1 1 1 1

**Her iki yönde de dönüşüm çalışır. (İkili-Onaltılı ve Onaltılı-İkili)**

# Sekizli-Onaltılı Dönüşüm

- Ara bir adım olarak İkili Dönüşüm gerçekleşir.

**Örnek:**



**Her iki yönde de dönüşüm çalışır. (Onaltılı-Sekizli ve Sekizli-Onaltılı)**

# Sayı Sistemleri

Decimal	Binary	Octal	Hex
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F



# 1.5 Tümlleme- Complements

- Her  $r$  tabanı için 2 farklı tümlleme tanımı yapılabilir:
- Taban Tümlleme -the radix complement ve Azaltılmış Taban Tümlleme- diminished radix complement.
- **Azaltılmış Taban Tümlleme-  $(r-1)$ 'e göre Tümlleme**
  - ◆  $N$   $r$  tabanında verilen  $n$  dijitle bir sayı olsun.  
 $(r-1)$ 'e göre tümlleme işlemi :  $(r^n - 1) - N$
- **Örnek : 6-dijit onlu sayılar:**
  - ◆ 9'a tümlleme:  $(r^n - 1) - N = (10^6 - 1) - N = 999999 - N$
  - ◆ 9'a tümlleme (546700 sayısı) :  $999999 - 546700 = 453299$
- **Örnek : 7-dijit ikili sayılar:**
  - ◆ 1'e tümlleme:  $(r^n - 1) - N = (2^7 - 1) - N = 1111111 - N$
  - ◆ 1'e tümlleme (1011000):  $1111111 - 1011000 = 0100111$
- **Sonuç:**
  - ◆  $(r^n - 1)$  den çıkarma asla bir borç getirmez
  - ◆ Azaltılmış Taban Tümlleme dijitle düzeyinde gerçekleştirilebilir.

# Tümleme

- 1'e Tümleme (*Diminished Radix Complement*)
  - ◆ Bütün '0' lar tüm '1' ler haline dönüşür.
  - ◆ Bütün '1' ler tüm '0' lar haline dönüşür.

Örnek  $(10110000)_2$

$\Rightarrow (01001111)_2$

...

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

# Tümleme

- Taban Tümleme: Radix Complement

$N$ ,  $r$  tabanında verilen  $n$  dijitli bir sayı olsun

$$r^n - N \quad N \neq 0 \text{ ve}$$

$$0 \quad N = 0.$$

$$r^n - N = [(r^n - 1) - N] + 1. \quad (r-1)'e \text{ tümleyen} + 1$$

- Örnek: Taban-10

The 10's complement of 012398 is 987602

The 10's complement of 246700 is 753300

- Örnek: Taban-2

The 2's complement of 1101100 is 0010100

The 2's complement of 0110111 is 1001001

# Tümleme

- 2'e Tümleme (*Radix Complement*)
  - ◆ 1 e tümleyeni bul ve 1 ilave et..

**veya**

*Sağdan ilk 1 in solundaki bitleri tümleyenini al*

Number:        **1** **0** **1** **1** **0** **0** **0** **0**

**1** **0** **1** **1** **0** **0** **0** **0**

1's Comp.:    **0** **1** **0** **0** **1** **1** **1** **1**

      +                    **1**  
      —————  
      **0** **1** **0** **1** **0** **0** **0** **0**

**0** **1** **0** **1** **0** **0** **0** **0**

# Tümleme

- Tümleme Yardımıyla Çıkarma

- ◆  *$r$  tabanında  $n$ -dijitli 2 işaretli sayının farkı  $M - N$  aşağıdaki gibi bulunur:*

1. Add the minuend  $M$  to the  $r$ 's complement of the subtrahend  $N$ . Mathematically,  $M + (r^n - N) = M - N + r^n$ .
2. If  $M \geq N$ , the sum will produce an end carry  $r^n$ , which can be discarded; what is left is the result  $M - N$ .
3. If  $M < N$ , the sum does not produce an end carry and is equal to  $r^n - (N - M)$ , which is the  $r$ 's complement of  $(N - M)$ . To obtain the answer in a familiar form, take the  $r$ 's complement of the sum and place a negative sign in front.



# Tümleme

- Örnek

- ◆ 10'a tümleme kullanarak,  $72532 - 3250$  çıkarma işlemi

	$M =$	72532
10's complement of	$N =$	<u>+96750</u>
	Sum =	169282
	Discard end carry $10^5 =$	<u>-100000</u>
	Answer =	69282

- Örnek

- ◆ 10'a tümleme kullanarak,  $3250 - 72532$  çıkarma işlemi.

	$M =$	03250
10's complement of	$N =$	<u>+27468</u>
	Sum =	30718



Sonlandırma eldesi  
yok



Bu yüzden , sonuç:  $-(10 \text{ a tümleyeni } 30718) = -69282$ .

# Tümleme

- Örnek

♦  $X = 1010100$  ve  $Y = 1000011$ ,

(a)  $X - Y$ ; (b)  $Y - X$ , 2'e tümleme kullanarak

(a)	$X =$	1010100
	2's complement of $Y =$	<u>+0111101</u>
	Sum =	10010001
	Discard end carry $2^7 =$	<u>-10000000</u>
	Answer. $X - Y =$	0010001

(b)	$Y =$	1000011
	2's complement of $X =$	<u>+ 0101100</u>
	Sum =	1101111

Sonlandırma eldesi yok  
 $Y - X = -$  (2's complement  
of 1101111) = - 0010001.

# Tümleme

- İşaretsiz sayıların çıkarımı  $(r - 1)$ 'e tümleme yardımıyla yapılabilir:  
 $(r - 1)$  'e tümleme  $r$  tümlemeden 1 çıkararak bulunabilir.
- Örnek
  - ♦ 1'e tümleme kullanarak

$$\begin{array}{r} \text{(a) } X - Y = 1010100 - 1000011 \\ X = 1010100 \\ \text{1's complement of } Y = \pm 0111100 \\ \text{Sum} = 10010000 \\ \text{End-around carry} = \underline{+ \quad 1} \\ \text{Answer. } X - Y = 0010001 \end{array}$$

$$\begin{array}{r} \text{(b) } Y - X = 1000011 - 1010100 \\ Y = 1000011 \\ \text{1's complement of } X = \underline{+ 0101011} \\ \text{Sum} = 1101110 \end{array}$$



End carry yok  
 $Y - X = - (1\text{'s complement of } 1101110) = - 0010001.$

# 1.6 İşaretli İkili Sayılar

- Negatif Tamsayıları simgelemek için, negatif işareti gösterimine ihtiyaç vardır.
- Genelde sayının en solunda bulunan bit işareti göstermesi en çok kullanılan bir yoldur.
- İkili sayıların simgelenmesinde, Pozitif için 0 Negatif için 1 gösterimi kullanımı daha yaygındır.
- Örnek:

Signed-magnitude representation:	10001001
Signed-1's-complement representation:	11110110
Signed-2's-complement representation:	11110111

- Tabl 1.3 4 bitlik işaretli ikili sayıların üç farklı gösterimdeki karşılıklarını içermektedir.

# İşaretili İkili Sayılar

**Table 1.3**  
*Signed Binary Numbers*

<b>Decimal</b>	<b>Signed-2's Complement</b>	<b>Signed-1's Complement</b>	<b>Signed Magnitude</b>
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0	—	1111	1000
-1	1111	1110	1001
-2	1110	1101	1010
-3	1101	1100	1011
-4	1100	1011	1100
-5	1011	1010	1101
-6	1010	1001	1110
-7	1001	1000	1111

# 1.7 İkili Kodlar

- BCD Code

- ◆ k basamaklı bir sayı için 4k adet BCD basamağı gerekir.
- ◆ 10 tabanında gösterilen 396 sayısı BCD kodlamada 12 bit olacak şekilde 0011 1001 0110 olarak ifade edilir ve her bir 4'lü bit grubu bir rakamı temsil eder.
- ◆ Bir rakamın BCD kodlamadaki karşılığı ikili sayı sistemindeki gibidir.
- ◆ 1010'den 1111 e kadar olan BCD kodlarının herhangi bir karşılığı yoktur.

**Table 1.4**

*Binary-Coded Decimal (BCD)*

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

# İkili Kod

- Örnek:

- ♦ 10 tabanında 185 sayısının BCD kodlamada ve ikili sistemdeki karşılığı:



$$(185)_{10} = (0001\ 1000\ 0101)_{\text{BCD}} = (10111001)_2$$

- BCD toplama:

4	0100	4	0100	8	1000
<u>+ 5</u>	<u>+ 0101</u>	<u>+ 8</u>	<u>+ 1000</u>	<u>+ 9</u>	<u>+ 1001</u>
9	1001	12	1100	17	10001
			<u>+ 0110</u>		<u>+ 0110</u>
			10010		10111

# İkili Kod

- Örnek:

- ◆  $184 + 576 = 760$  işleminin BCD kullanılarak yapılması:

BCD	1	1		
	0001	1000	0100	184
	<u>+ 0101</u>	<u>0111</u>	<u>0110</u>	+576
Binary sum	0111	10000	1010	
Add 6	<u>        </u>	<u>0110</u>	<u>0110</u>	<u>        </u>
BCD sum	0111	0110	0000	760



# İkili Kodlar

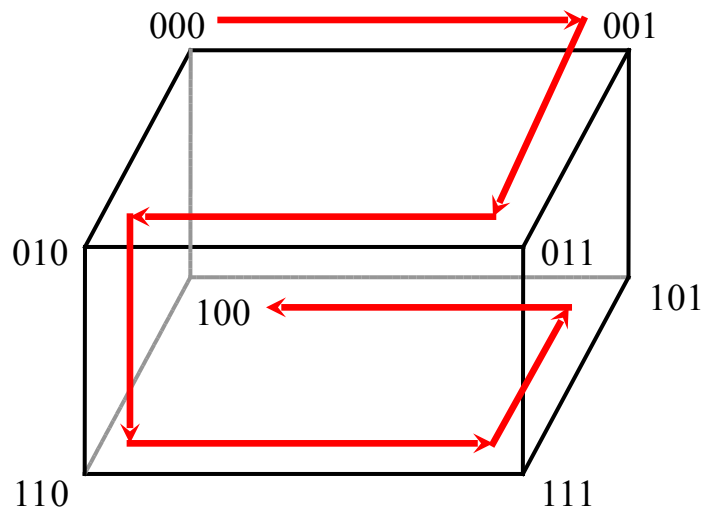
- Diğer Onlu Sistem Kodlamaları

**Table 1.5**  
*Four Different Binary Codes for the Decimal Digits*

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, -2, -1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
Unused bit combinations	1010	0101	0000	0001
	1011	0110	0001	0010
	1100	0111	0010	0011
	1101	1000	1101	1100
	1110	1001	1110	1101
	1111	1010	1111	1110

# İkili Kodlar

- Gray Code
  - ◆ The advantage is that only bit in the code group changes in going from one number to the next.
    - » Error detection.
    - » Representation of analog data.
    - » Low power design.



1-1 and onto!!

**Table 1.6**  
*Gray Code*

Gray Code	Decimal Equivalent
0000	0
0001	1
0011	2
0010	3
0110	4
0111	5
0101	6
0100	7
1100	8
1101	9
1111	10
1110	11
1010	12
1011	13
1001	14
1000	15

# İkili Kodlar

- American Standard Code for Information Interchange (ASCII) Character Code

**Table 1.7**

*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

# İkili Kodlar

- ASCII Character Code

## Control characters

---

NUL	Null	DLE	Data-link escape
SOH	Start of heading	DC1	Device control 1
STX	Start of text	DC2	Device control 2
ETX	End of text	DC3	Device control 3
EOT	End of transmission	DC4	Device control 4
ENQ	Enquiry	NAK	Negative acknowledge
ACK	Acknowledge	SYN	Synchronous idle
BEL	Bell	ETB	End-of-transmission block
BS	Backspace	CAN	Cancel
HT	Horizontal tab	EM	End of medium
LF	Line feed	SUB	Substitute
VT	Vertical tab	ESC	Escape
FF	Form feed	FS	File separator
CR	Carriage return	GS	Group separator
SO	Shift out	RS	Record separator
SI	Shift in	US	Unit separator
SP	Space	DEL	Delete

---

# ASCII Karakter Kodları

- American Standard Code for Information Interchange (Refer to Table 1.7)
- A popular code used to represent information sent as character-based data.
- It uses 7-bits to represent:
  - ◆ 94 Graphic printing characters.
  - ◆ 34 Non-printing characters.
- Some non-printing characters are used for text format (e.g. BS = Backspace, CR = carriage return).
- Other non-printing characters are used for record marking and flow control (e.g. STX and ETX start and end text areas).

# ASCII Özellikleri

- ASCII has some interesting properties:
  - ◆ Digits 0 to 9 span Hexadecimal values  $30_{16}$  to  $39_{16}$
  - ◆ Upper case A-Z span  $41_{16}$  to  $5A_{16}$
  - ◆ Lower case a-z span  $61_{16}$  to  $7A_{16}$ 
    - » Lower to upper case translation (and vice versa) occurs by flipping bit 6.

# İkili Kodlar

- Error-Detecting Code
  - ◆ To detect errors in data communication and processing, an eighth bit is sometimes added to the ASCII character to indicate its parity.
  - ◆ A **parity bit** is an extra bit included with a message to make the total number of 1's either even or odd.
- Example:
  - ◆ Consider the following two characters and their even and odd parity:

	With even parity	With odd parity
ASCII A = 1000001	01000001	11000001
ASCII T = 1010100	11010100	01010100

# İkili Kodlar

- Error-Detecting Code
  - ◆ **Redundancy** (e.g. extra information), in the form of extra bits, can be incorporated into binary code words to detect and correct errors.
  - ◆ A simple form of redundancy is **parity**, an extra bit appended onto the code word to make the number of 1's odd or even. Parity can detect all single-bit errors and some multiple-bit errors.
  - ◆ A code word has **even parity** if the number of 1's in the code word is even.
  - ◆ A code word has **odd parity** if the number of 1's in the code word is odd.
  - ◆ Example:

Message A: 10001001**1** (even parity)

Message B: 10001001**0** (odd parity)



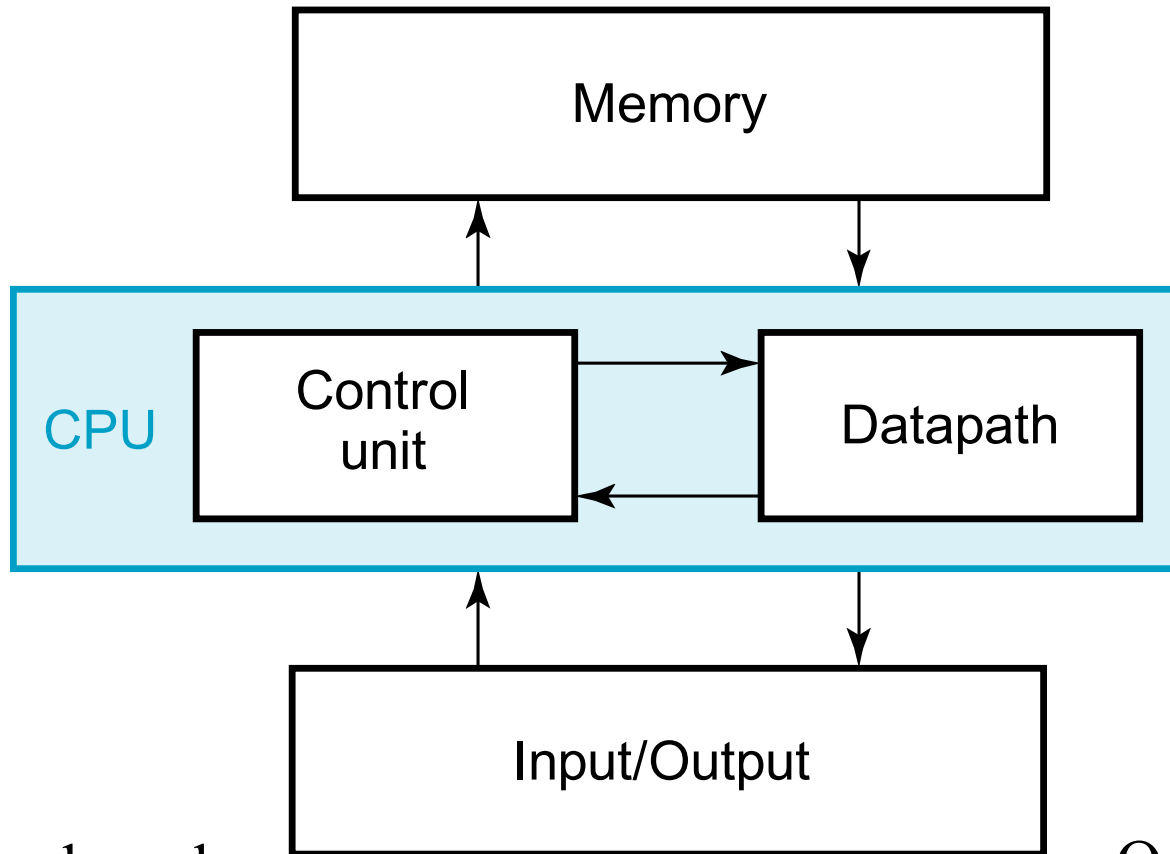
# 1.8 İkili Saklama ve Yazmaçlar

- Yazmaçlar (Registers)
  - ◆ İkili hücre iki kararlı durumdan birini saklama yeteneği olan elemandır.
  - ◆ Yazmaç bir grup ikili hücreden oluşur. Bir  $n$  hücreli yazmaç,  $n$ -bitlik ayrık bilgi miktarı saklayabilir.

$n$  hücre   $2^n$  mümkün durum

- İkili hücre
  - ◆ 2 kararlı durum
  - ◆ 1-bitlik bilgi saklama
  - ◆ Örnekler: flip-flop devreleri, ferrite nüveler, kapasitör
- Yazmaç
  - ◆ Bir grup ikili hücre
  - ◆ x86 CPU içindeki AX yazmacı
- Yazmaç Transferi
  - ◆ Bir yazmaçtan diğerine bilginin iletilmesi.
  - ◆ Sayısal Sistemde en önemli işlemlerden biridir.

# Sayısal Bilgisayar Örneği

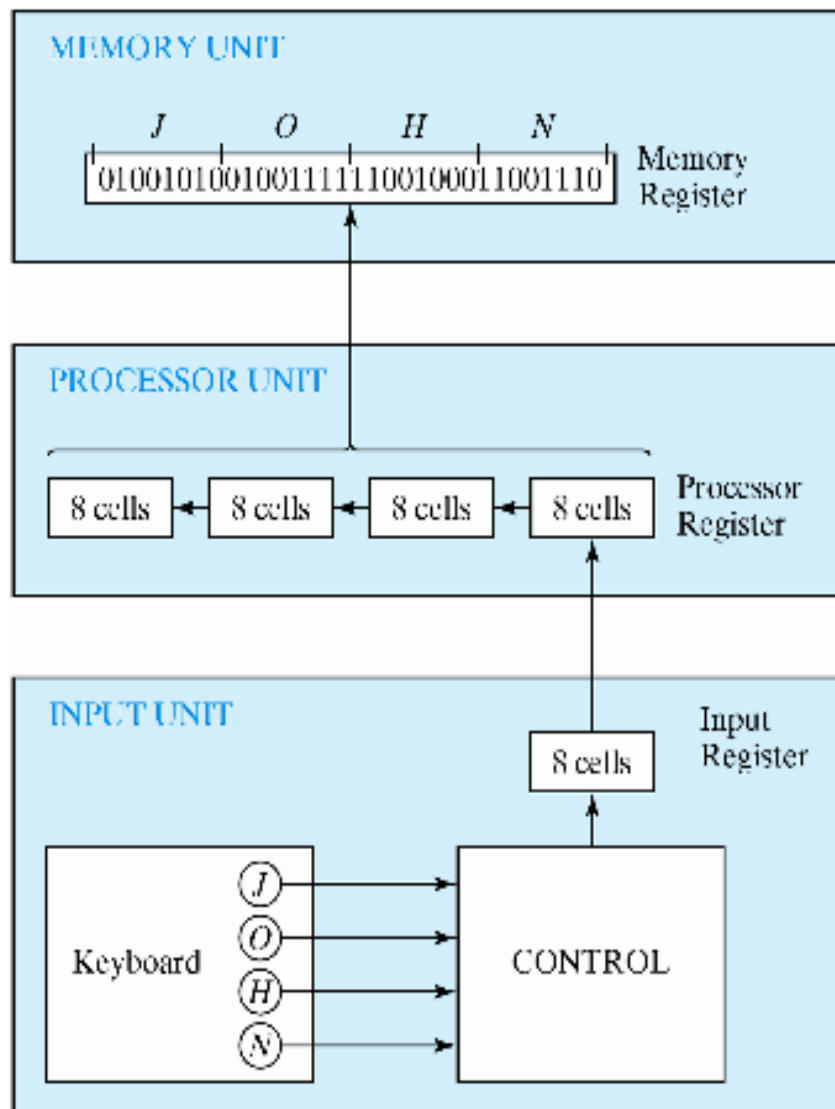


Inputs: Keyboard,  
mouse, modem,  
microphone

Outputs: CRT,  
LCD, modem,  
speakers

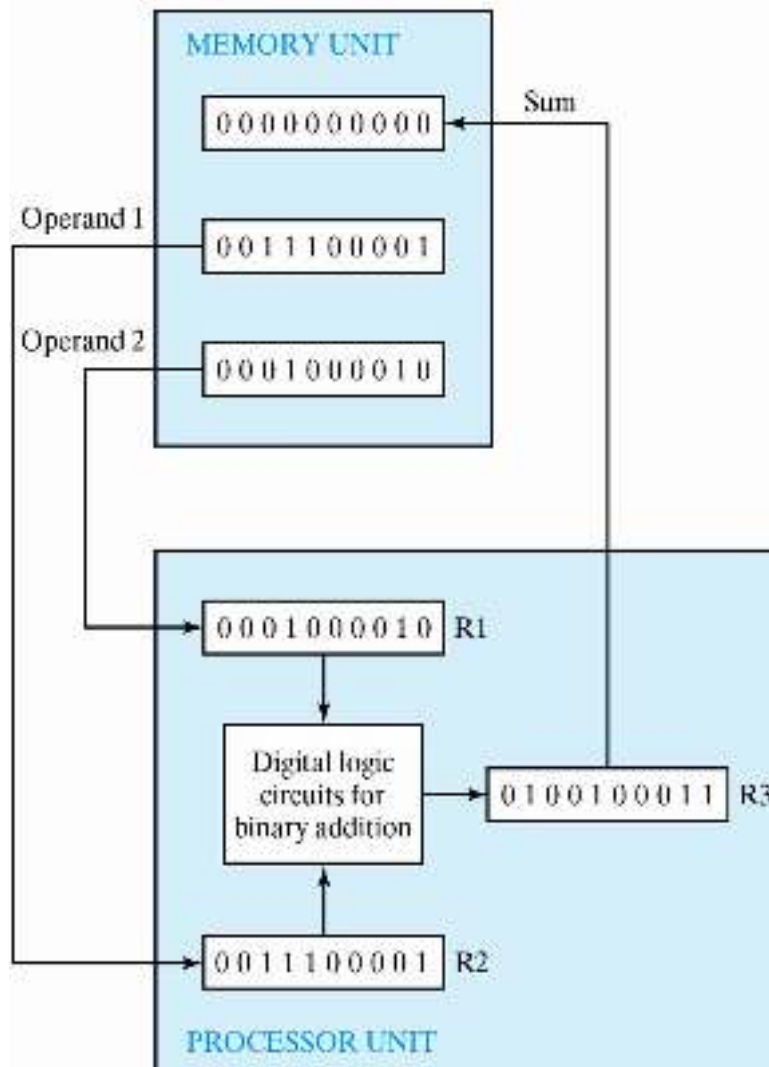
Synchronous or  
Asynchronous?

# Bilginin Transferi



Yazmaçlar arası bilgi transferi

# Bilginin Transferi



## ◆ Load-store işlemler

LD R1;

LD R2;

ADD R3, R2, R1;

SD R3;

İkili Bilgi İşleme Örneği

# 1.9 İkili Lojik

- İkili Mantık tanımı
  - ◆ İkili Mantık ikili değişkenler ve lojik işlem dizini içerir.
  - ◆ Değişkenler harflerle simgelenir  $A, B, C, x, y, z$ , vs...herbir değişken sadece 0 veya 1 değerini alabilir.
  - ◆ 3 temel lojik işlem: AND, OR, ve NOT.

1. AND: This operation is represented by a dot or by the absence of an operator. For example,  $x \cdot y = z$  or  $xy = z$  is read “ $x$  AND  $y$  is equal to  $z$ ,” The logical operation AND is interpreted to mean that  $z = 1$  if only  $x = 1$  and  $y = 1$ ; otherwise  $z = 0$ . (Remember that  $x, y$ , and  $z$  are binary variables and can be equal either to 1 or 0, and nothing else.)
2. OR: This operation is represented by a plus sign. For example,  $x + y = z$  is read “ $x$  OR  $y$  is equal to  $z$ ,” meaning that  $z = 1$  if  $x = 1$  or  $y = 1$  or if both  $x = 1$  and  $y = 1$ . If both  $x = 0$  and  $y = 0$ , then  $z = 0$ .
3. NOT: This operation is represented by a prime (sometimes by an overbar). For example,  $x' = z$  (or  $\bar{x} = z$ ) is read “not  $x$  is equal to  $z$ ,” meaning that  $z$  is what  $x$  is not. In other words, if  $x = 1$ , then  $z = 0$ , but if  $x = 0$ , then  $z = 1$ . The NOT operation is also referred to as the complement operation, since it changes a 1 to 0 and a 0 to 1.

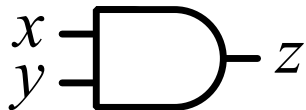
# İkili Lojik

- Doğruluk Tablosu, Boolean İfadeleri, ve Lojik Kapılar

## AND

$x$	$y$	$z$
0	0	0
0	1	0
1	0	0
1	1	1

$$z = x \cdot y = x y$$



## OR

$x$	$y$	$z$
0	0	0
0	1	1
1	0	1
1	1	1

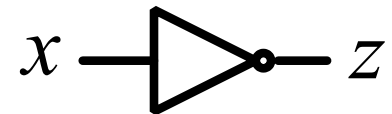
$$z = x + y$$



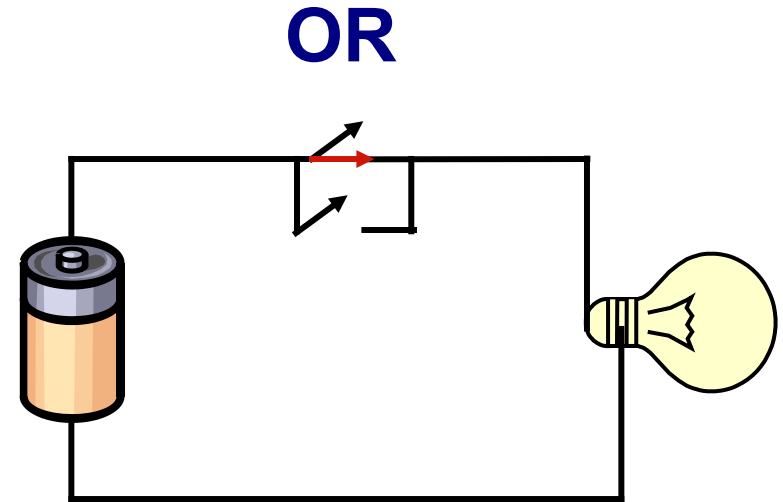
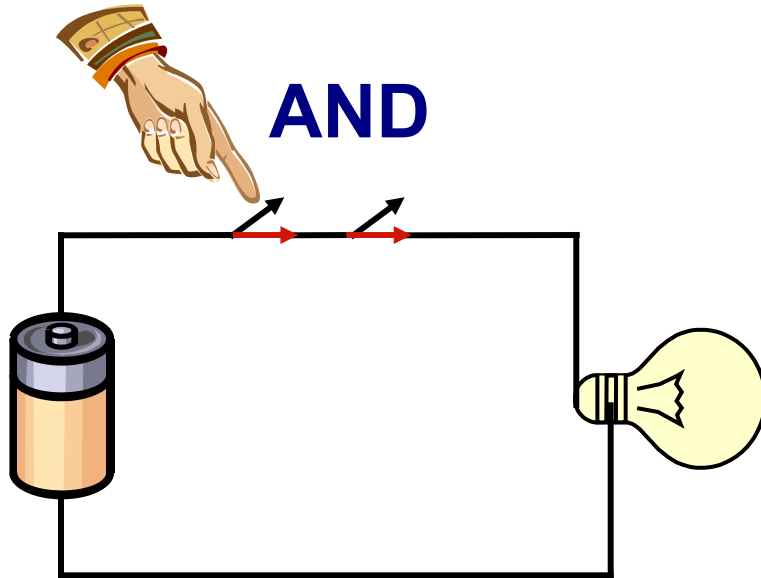
## NOT

$x$	$z$
0	1
1	0

$$z = \overline{x} = x'$$

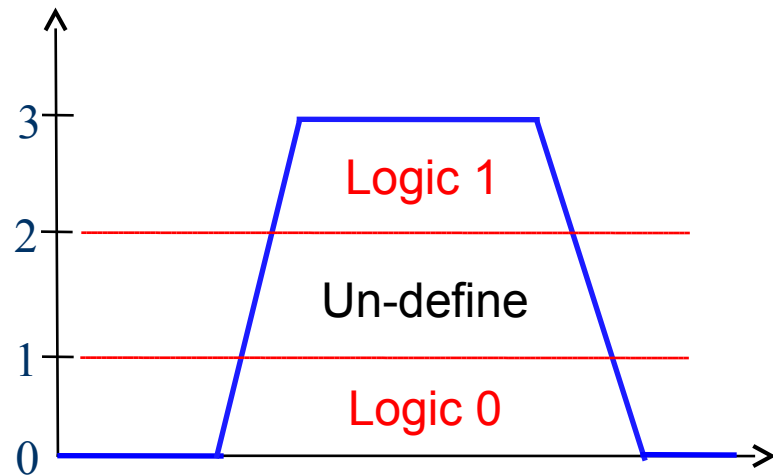
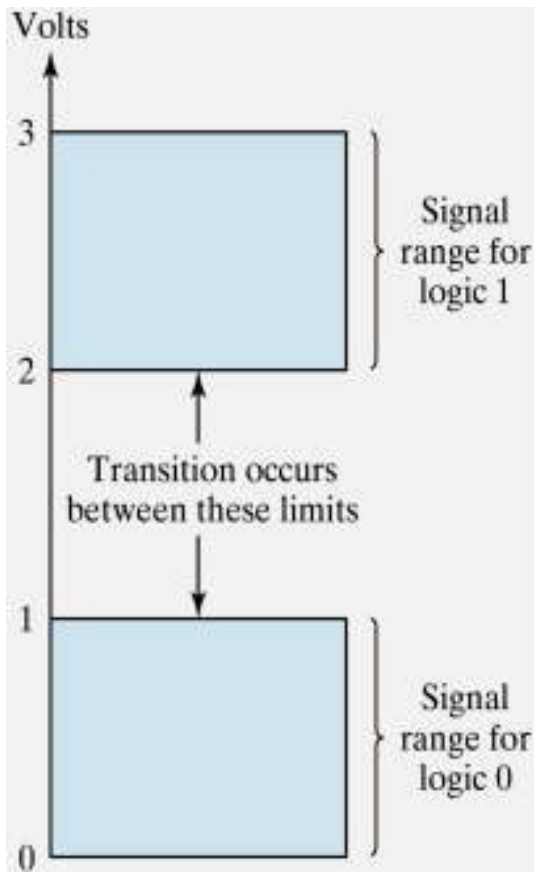


# Anahtarlama Devreleri



# İkili Lojik

- Lojik Kapılar
  - ◆ İkili İşaret Örnekleri

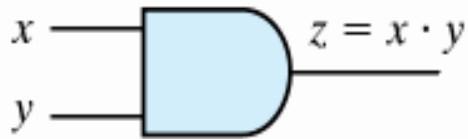




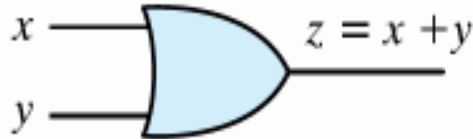
# İkili Lojik

- Lojik Kapılar

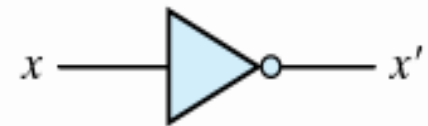
Grafik Semboller ve Giriş-Çıkış İşaretleri



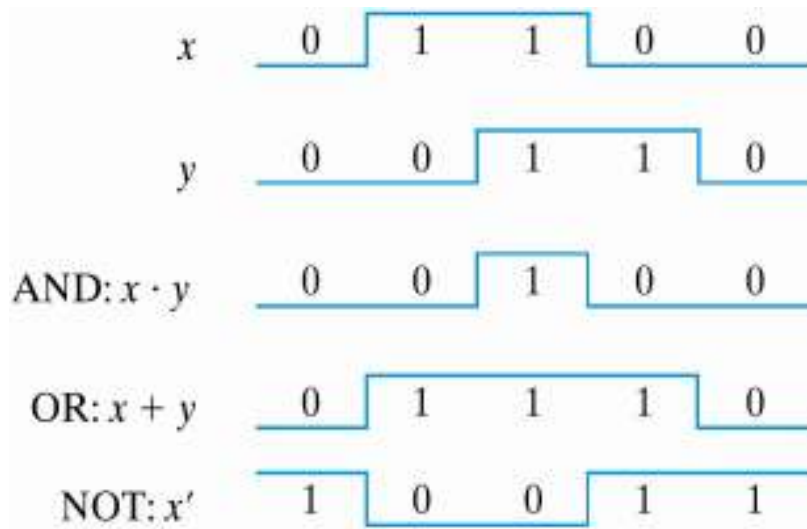
(a) Two-input AND gate



(b) Two-input OR gate



(c) NOT gate or inverter



# İkili Lojik

- Lojik Kapılar

- Grafik Semboller ve Giriş-Çıkış İşaretleri:

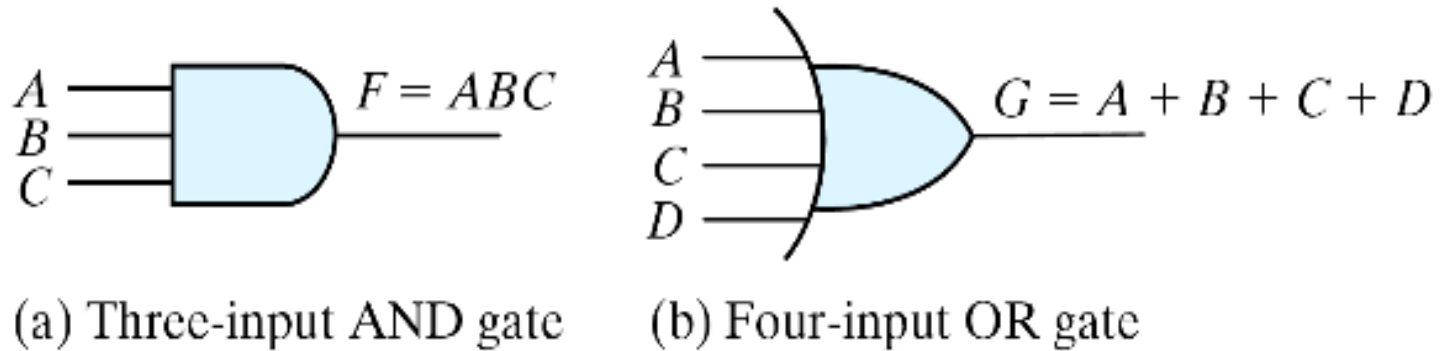


Fig. 1.6 Çok girişli Kapılar