Al-Mustaqbal University

prosthetic and orthotic Engineering

LOGIC CIRCUIT(Lab)

CREQ 201

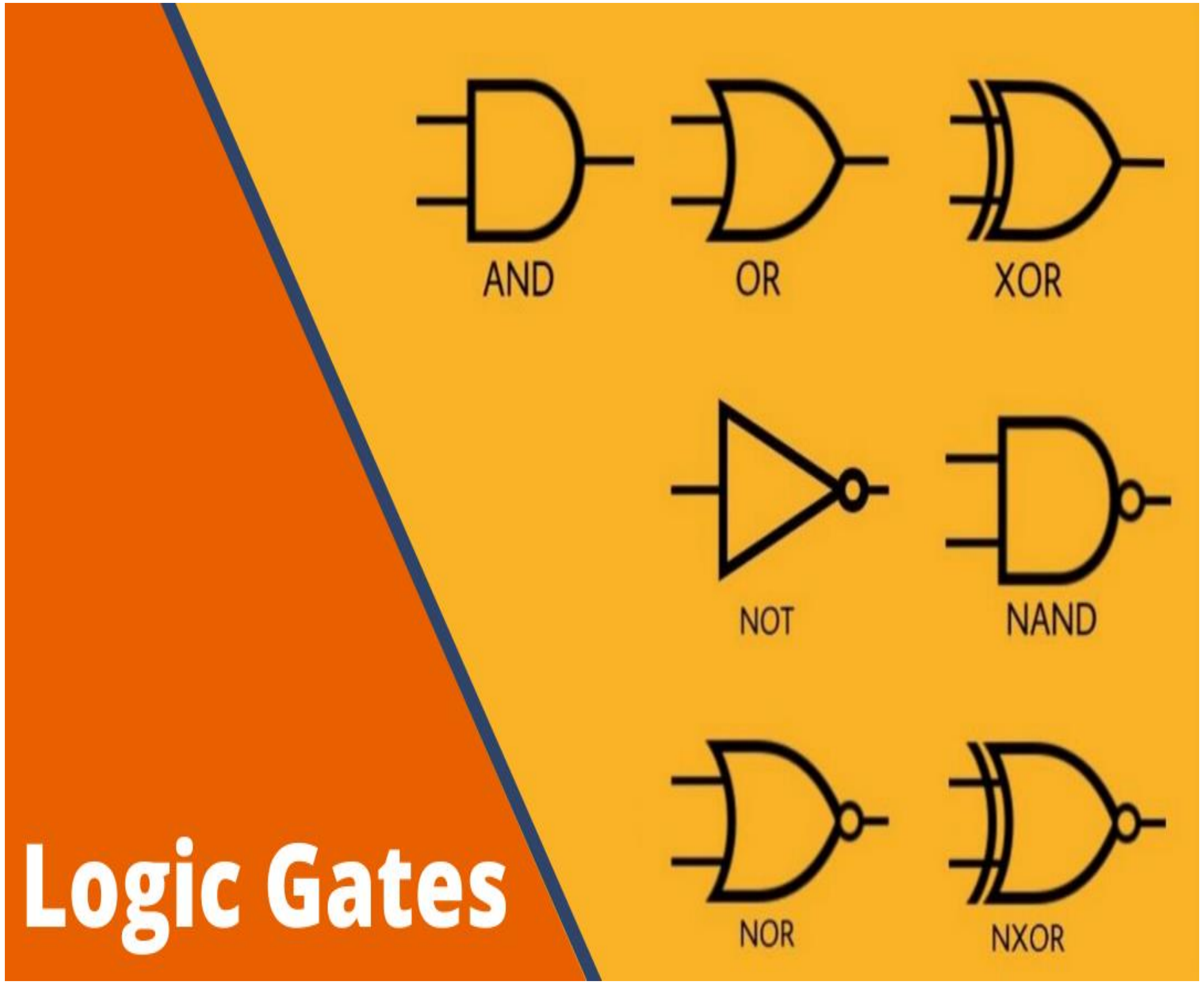
-Lecture 1-

Prof. Dr Mohammed Almaamori
ENG. Noor Adnan Madlool

Logic gates: are idealized or physical devices implementing a Boolean function, which it performs a logical operation on one or more logical inputs and produce a single output.

The main hierarchy for logic gates is as follows:-

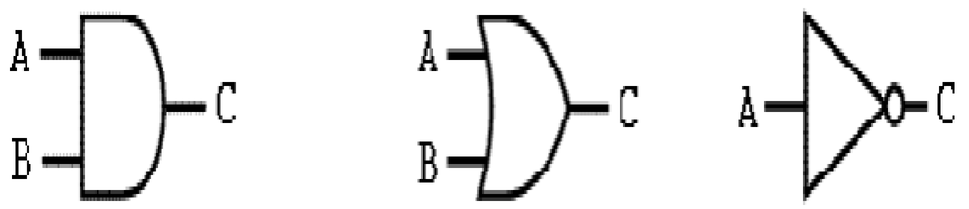
- ✓ Basic Gates
- ✓ Universal Gates
- ✓ Advanced Gates



Basic Gates

- **AND gate**: - Function of AND gate is to give the output true when both the inputs are true. In all the other remaining cases output becomes false.
- **OR gate**: - Function of OR gate is to give output true when one of the either inputs are true .In the remaining case output becomes false.
- **NOT gate**: -Function of NOR gate is to reverse the nature of the input .It converts true input to false and vice versa.

Basic Gates Symbol



AND

Inputs A	Inputs B	Output C
0	0	0
0	1	0
1	0	0
1	1	1

output $C=A.B$

OR

Inputs A	Inputs B	Output C
0	0	0
0	1	1
1	0	1
1	1	1

output $C=A+B$

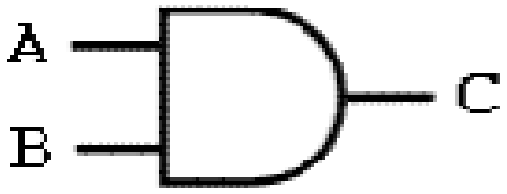
NOT

Input A	Output C
0	1
1	0

output $C=\bar{A}$

Basic Gates Symbol

AND



Inputs		Output
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

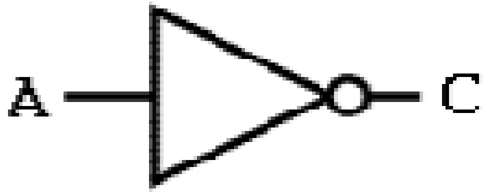
output $C=A.B$

OR

Inputs		Output
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

output $C=A+B$

NOT



Input	Output
A	C
0	1
1	0

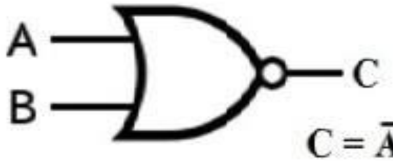
output $C=\overline{A}$

Universal Gates

- **NAND gate**: -Function of NAND gate is to give true output when one of the two provided input are false. In the remaining output is true case
- **NOR gate**: - NOR gate gives the output true when both the two provided input are false. In all the other cases output remains false.

Universal Gates:- NAND & NOR

NOR GATE

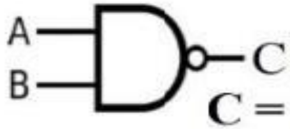


$C = \overline{A + B}$

TRUTH TABLE

INPUT		OUTPUT
A	B	A NCR B
0	0	1
0	1	0
1	0	0
1	1	0

NAND GATE



$C = \overline{A.B}$

Truth Table

INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Universal Gates:- NAND & NOR

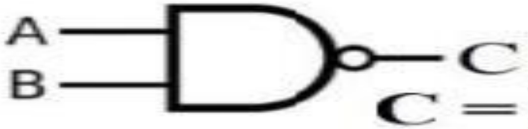
NOR GATE

$C = \overline{A + B}$

TRUTH TABLE

INPUT		OUTPUT
A	B	A NCR B
0	0	1
0	1	0
1	0	0
1	1	0

NAND GATE



$C = \overline{A.B}$

Truth Table

INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Universal Gates

- **NAND gate**: -Function of NAND gate is to give true output when one of the two provided input are false. In the remaining output is true case

- **NOR gate**: - NOR gate gives the output true when both the two provided input are false. In all the other cases output remains false.

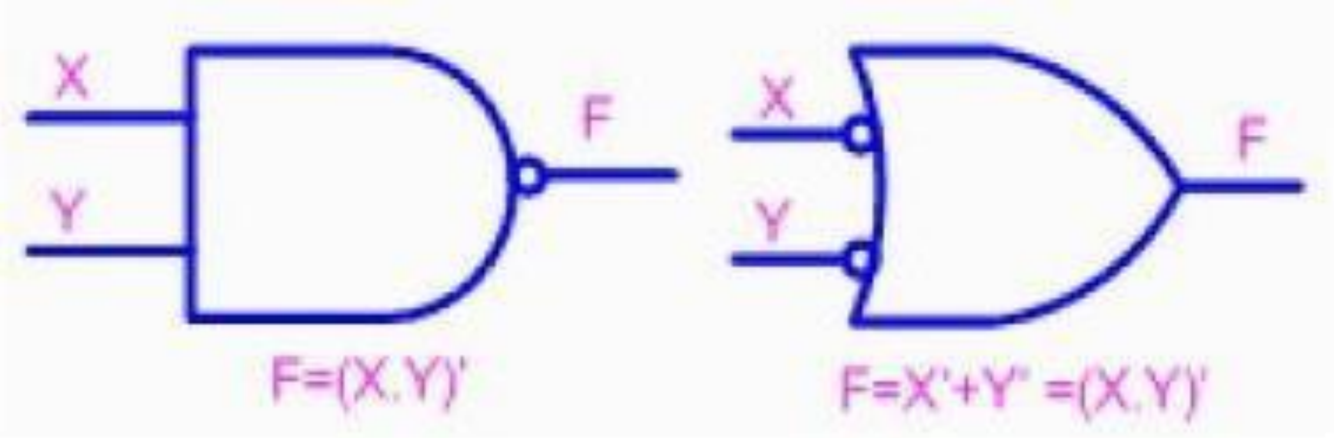


Figure: Another NAND Gate Symbol

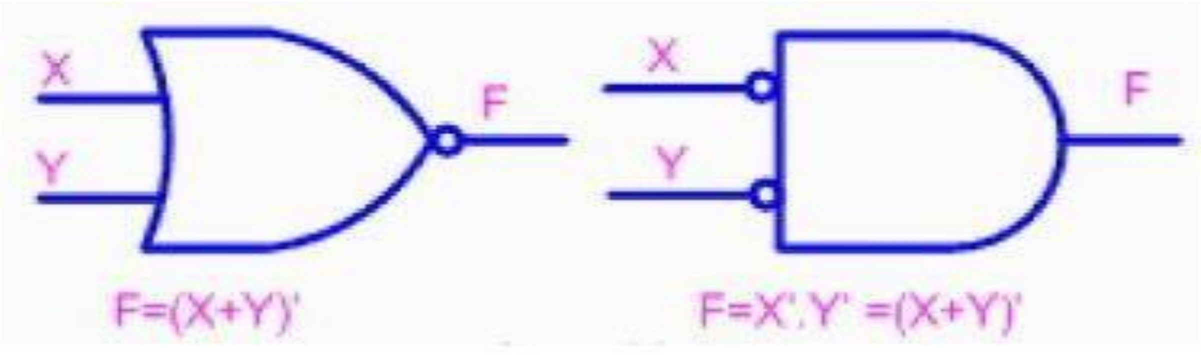
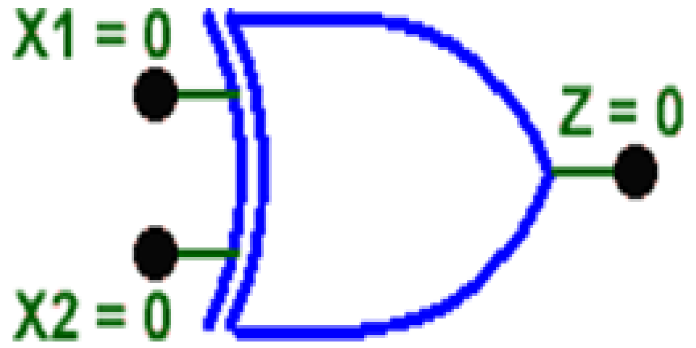


Figure:Another NORGate Symbol

Advanced Gates

- **XOR gate**: - The function of XOR gate is to give output true only when both the inputs are true.

EXCLUSIVE OR

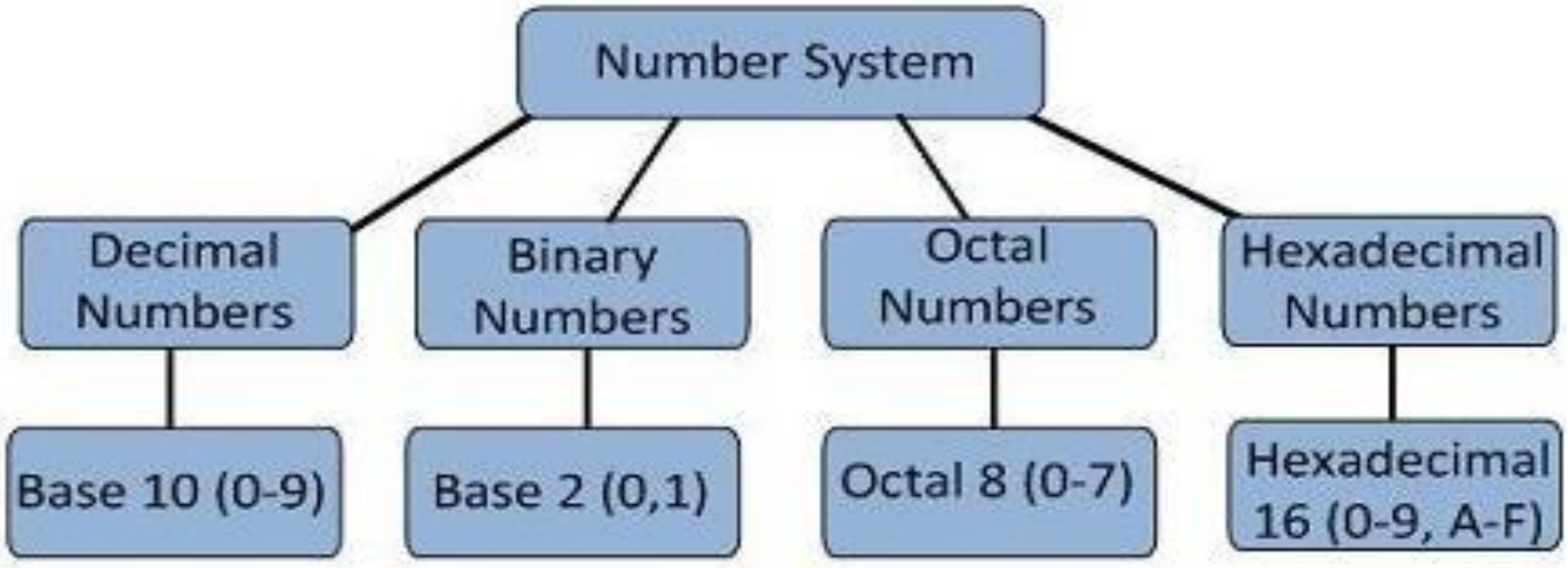


X1	X2	Z
0	0	0
0	1	1
1	0	1
1	1	0

Z = X1 Å X2

Number Systems

Some of the important types of number system are:



Decimal Numbers

In the **decimal** number system, each of the ten digits (0 through 9) represents a certain quantity.

The position of each digit in a decimal number indicates the magnitude of the quantity represented and can be assigned a weight. The weights for whole numbers are positive powers of ten that increase from right to left, beginning with 100 = 1.

$$\ldots\ 10^5\ 10^4\ 10^3\ 10^2\ 10^1\ 10^0$$

For fractional numbers, the weights are negative powers of ten that decrease from left to right beginning with 10-1.

$$10^2\ 10^1\ 10^0 . 10^{-1}\ 10^{-2}\ 10^{-3}\ \ldots$$

Decimal point

EXAMPLE 1: Express the decimal number 47 as a sum of the values of each digit.

Solution:

Decimal Number:	4	7
Decimal Weight:	10^1	10^0

Base

$$\mathbf{47} = (4 \times 10^1) + (7 \times 10^0)$$

$$= (4 \times 10) + (7 \times 1) = \mathbf{40 + 7}$$

Note: The digit 4 has a weight of 10, which is 10^1, as indicated by its position. The digit 7 has a weight of 1, which is 10^0, as indicated by its position.

EXAMPLE 2: Express the decimal number 568.23 as a sum of the values of each digit.

Solution:

Decimal Number:	5	6	8	.	2	3
Decimal Weight:	10^2	10^1	10^0	.	10^{-1}	10^{-2}

$$\mathbf{568.23} = (5 \times 10^2) + (6 \times 10^1) + (8 \times 10^0) + (2 \times 10^{-1}) + (3 \times 10^{-2})$$

$$= (5 \times 100) + (6 \times 10) + (8 \times 1) + (2 \times 0.1) + (3 \times 0.01)$$

$$= \mathbf{500 + 60 + 8 + 0.2 + 0.03}$$

Note: The whole number digit 5 has a weight of 100, which is 10^2, the digit 6 has a weight of 10, which is 10^1, the digit 8 has a weight of 1, which is 10^0, the fractional digit 2 has a weight of 0.1, which is 10^{-1}, and the fractional digit 3 has a weight of 0.01, which is 10^{-2}.

Binary Numbers:

The binary number system is another way to represent quantities. It is less complicated than the decimal system because the binary system has only two digits. The two binary digits (bits) are 1 and 0.The Weighting Structure of Binary Numbers:

A binary number is a weighted number. The right-most bit is the LSB (least significant bit) in a binary whole number. The weights increase from right to left by a power of two for each bit. The left-most bit is the MSB (most significant bit); its weight depends on the size of the binary number.

Binary Weight:	2^3	2^2	2^1	2^0
	8	4	2	1
Binary Number:	1	1	0	0
	↓ MSB			↓ LSB

$$\therefore (12)_{10} = (1100)_2$$

Binary Numbers:

Example 1

Binary Weight:	2^3	2^2	2^1	2^0
	8	4	2	1
Binary Number:	1	1	0	0
	↓			↓
	MSB			LSB

$$\therefore (12)_{10} = (1100)_2$$

Binary Numbers:

Example 2

Binary Weight:	2^6	2^5	2^4	2^3	2^2	2^1	20
	64	32	16	8	4	2	1
Binary Number:	1	1	0	1	1	0	1
	MSB						LSB

$$\mathbf{1101101} = 2^6 + 2^5 + 2^3 + 2^2 + 2^0$$

$$= 64 + 32 + 8 + 4 + 1 = \mathbf{109}$$

$$\therefore (\mathbf{1101101})_2 = (\mathbf{109})_{10}$$

Hexadecimal Numbers:

The **hexadecimal** number system has a base of sixteen.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,1A, 1B, 1C, 1D, 1E, 1F,20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B,2C, 2D, 2E, 2F, 30, 31, ….

Decimal	Binary	Hexadecimal
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

Hexadecimal Numbers:

Example 1

$$\underbrace{1100}_{C}\,\underbrace{1010}_{A}\,\underbrace{0101}_{5}\,\underbrace{0111}_{7} = \mathbf{CA57_{16}}$$

$$\underbrace{0011}_{3}\,\underbrace{1111}_{F}\,\underbrace{0001}_{1}\,\underbrace{0110}_{6}\,\underbrace{1001}_{9} = \mathbf{3F169_{16}}$$

Example 2

(a) $\overbrace{1}^{1}\,\overbrace{0000}^{0}\,\overbrace{1010}^{A}\,\overbrace{0100}^{4}$ → 1000010100100

(b) $\overbrace{1100}^{C}\,\overbrace{1111}^{F}\,\overbrace{1000}^{8}\,\overbrace{1110}^{E}$ → 1100111110001110

(c) $\overbrace{1001}^{9}\,\overbrace{0111}^{7}\,\overbrace{0100}^{4}\,\overbrace{0010}^{2}$ → 1001011101000010

Binary Coded Decimal (BCD)

Binary coded decimal (BCD) is a way to express each of the decimal digits with a binary code. There are **only ten code** groups in the BCD system.

(a) $\overbrace{0011}^{3}\overbrace{0101}^{5}$

(b) $\overbrace{1001}^{9}\overbrace{1000}^{8}$

(c) $\overbrace{0001}^{1}\overbrace{0111}^{7}\overbrace{0000}^{0}$

(d) $\overbrace{0010}^{2}\overbrace{0100}^{4}\overbrace{0110}^{6}\overbrace{1001}^{9}$

(a) $\underbrace{1000}_{8}\underbrace{0110}_{6}$

(b) $\underbrace{0011}_{3}\underbrace{0101}_{5}\underbrace{0001}_{1}$

Thanks