Logical Design

CS 221

Prof.Dr. Mohamed Osama Khozium

Simple Logic Gates
&
Boolean algebra

Module Outline

- Binary Logic and Basic Gates
- Boolean Algebra
- The Boolean expression for a logic circuit
- Implementation of a logic circuit using B.E
- Implementation of a logic circuit via truth table
- Converting a Boolean expression to a truth table
- Simplification of Boolean Functions

Binary Logic

- Deals with variables that take on only two discrete values and operations of mathematical logic that are applied to these variables
- The two values are known by different names:
  - HIGH and LOW
  - TRUE and FALSE
  - 0 and 1
- Variables may be denoted by any name but single letter character names such as *A*, *B*, *C*, *X*, *Y*, *Z*, etc. are common and easy to use

Logical Operations

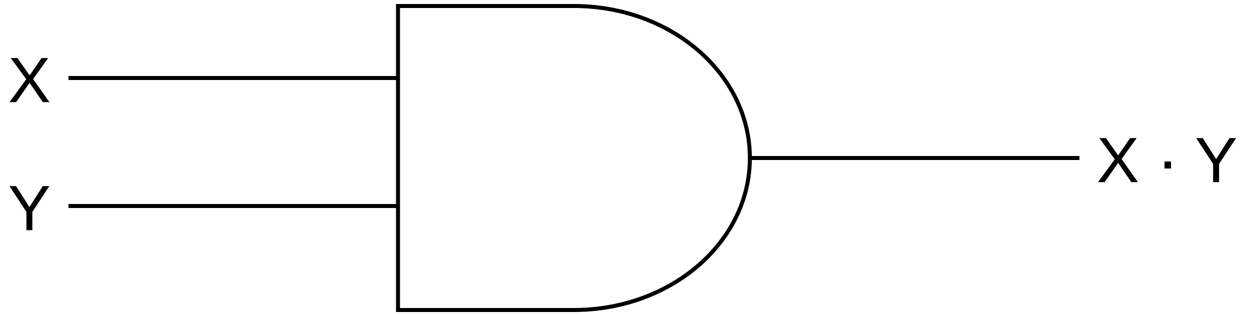
- AND
  - Represented by a dot $\cdot$ or sometimes $\wedge$ or $\cap$
  - Binary operator, i.e. operates on two variables at a time
- OR
  - Represented by a + or sometimes $\vee$ or $\cup$
  - Binary operator, i.e. operates on two variables at a time
- NOT
  - Represented by a bar over the variable, e.g. $\overline{A}$ or A'
  - Unary operator, i.e. operates on a single variable
  - Also referred to as COMPELMENT operator

Binary Logic and Basic Gates

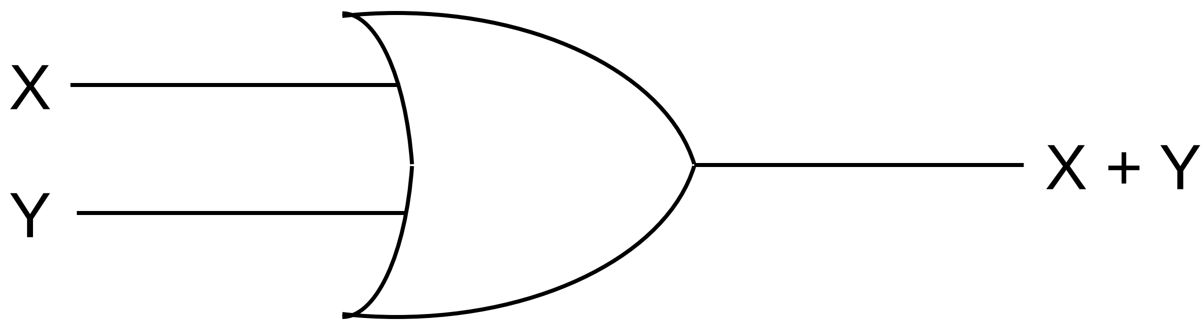
- Digital Circuits and Systems are complex
- However, the basic building blocks of all digital circuits and systems are logic gates
- Logic gates are therefore called the basic primitives of digital systems
- Logic gates in reality are implemented using electronic components such as transistors
- However, we are not concerned with their internal electronic properties but rather their external logic behavior

Standard Symbols

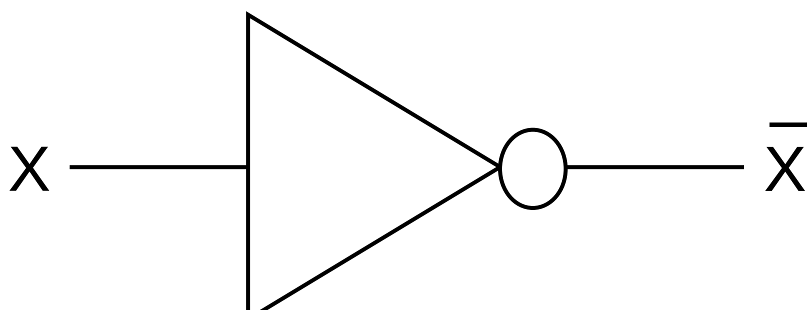
AND



OR



NOT



Truth Table

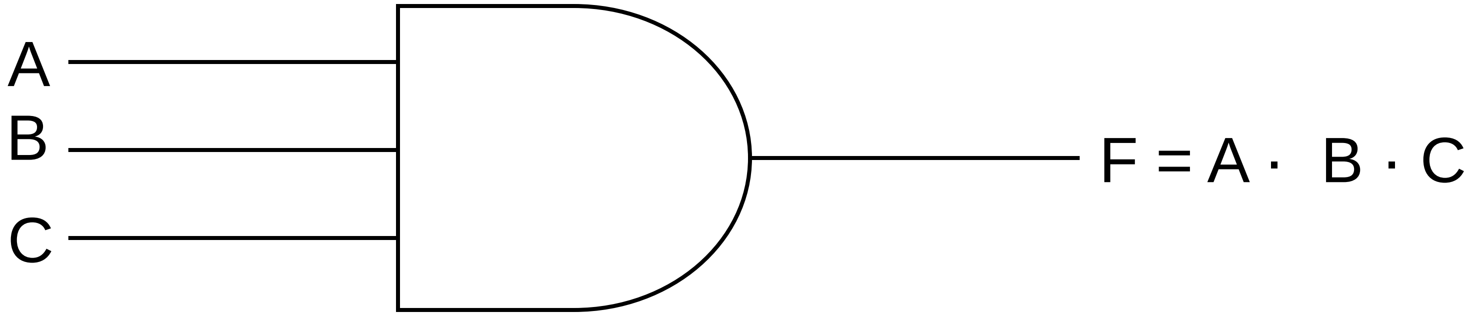
- A table of combinations of the binary variables showing the relationship between the values the variables take on and the values of the result of the operation

AND		
X	Y	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

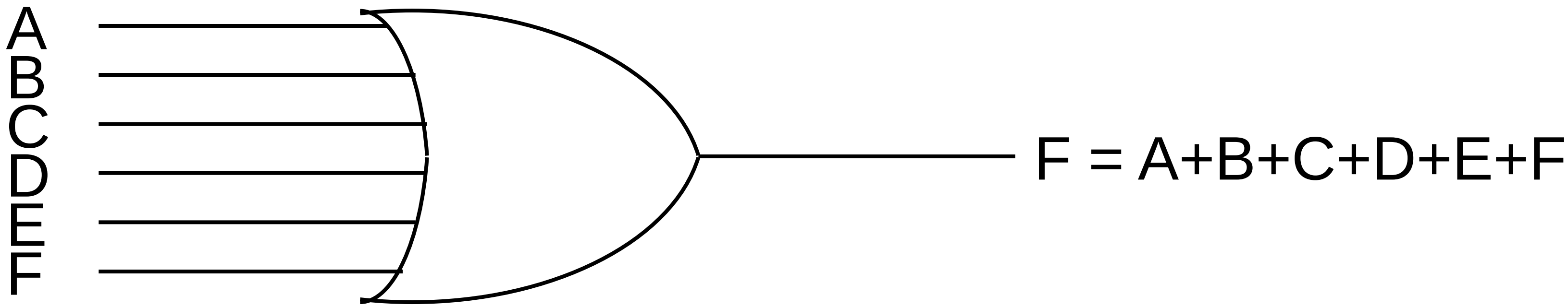
OR		
X	Y	$Z = X + Y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
X	$Z = \overline{X}$
0	1
1	0

Gates with More Than Two Inputs



3-Input AND gate



6-Input OR gate

Truth Table

AND			
X	Y	Z	F = X · Y · Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Boolean Algebra

- Defines rules of operations on Binary Logic and Logic Functions
- Sometimes similar to Binary Arithmetic but some times different – because binary logic variables can only take on two possible values 0 and 1
- A Boolean function consists of a binary variable denoting the function, an equals sign, and an algebraic expression formed by using binary variables. For example

$$F = X + Y'Z$$

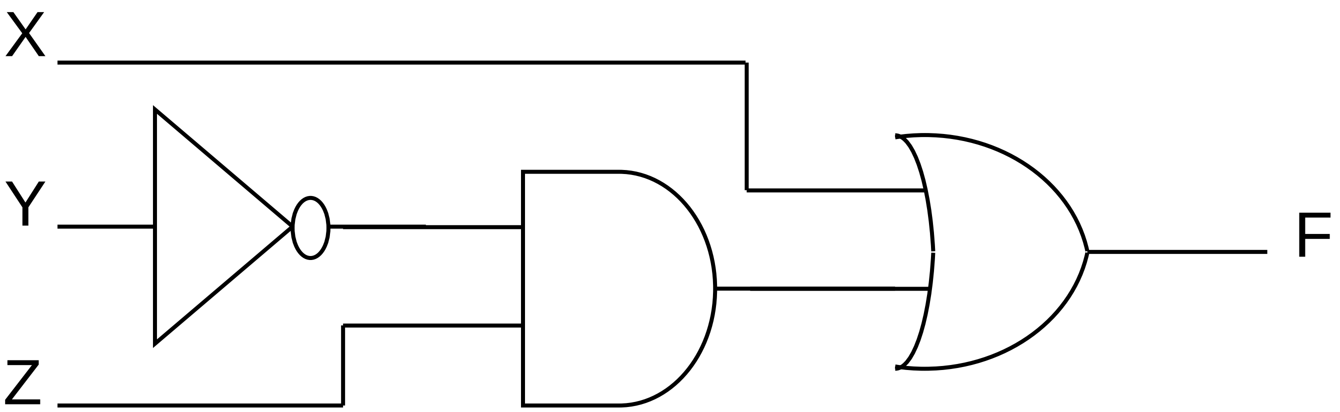
Truth Tables for Boolean Functions

- Truth Table can be constructed for every boolean function
- A boolean function of *n*-variables will have 2^n rows in its truth table
- These 2^n inputs are formed by counting from 0 to $2^n - 1$
- For example $G(W,X,Y,Z)$ will have 16 rows in the truth table and $F(A,B)$ will have 4 rows

Truth Table for F = X + Y'Z

X	Y	Z	Y'	Y'Z	X + Y'Z
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

Logic Circuit Diagram for F = X + Y'Z



Implementation of a logic circuit using a boolean expression

$$Y = AB(CD' + EF)$$

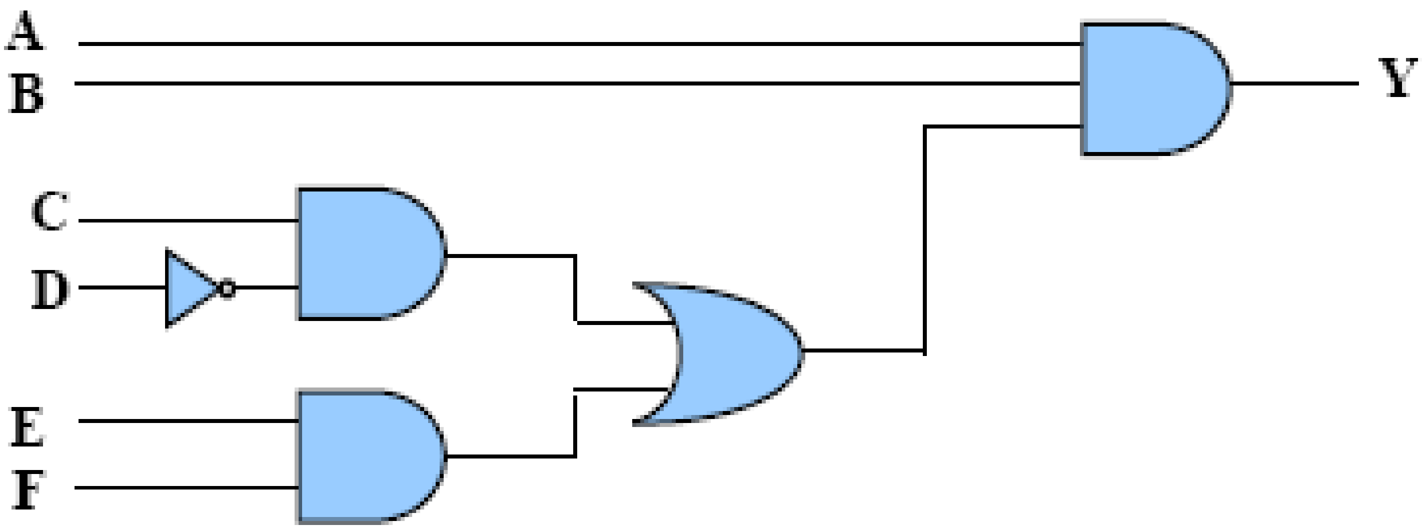
AND

NOT

OR

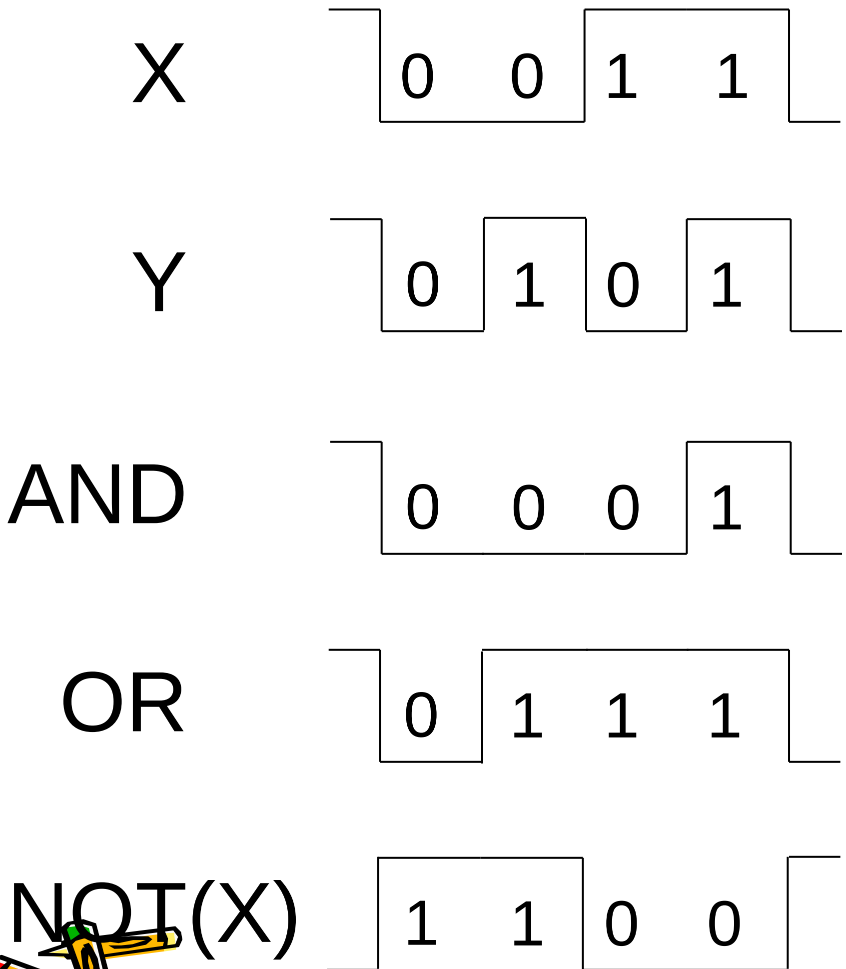
$$Y = A\,B\,(C\overline{D} + EF)$$

AND



$$Y = AB(CD' + EF)$$

Timing Diagram



- Horizontal axis represents time
- Vertical axis shows the value of the signal
- High voltage level represents 1 and low voltage level represents 0
- Timing diagrams are very important in digital systems design and verification

Basic Identities of Boolean Algebra

1. $X + 0 = X$
2. $X \cdot 1 = X$
3. $X + 1 = 1$
4. $X \cdot 0 = 0$
5. $X + X = X$
6. $X \cdot X = X$
7. $X + X' = 1$
8. $X \cdot X' = 0$
9. $(X')' = X$

Basic Identities of Boolean Algebra

Commutative

1. $X + Y = Y + X$
2. $XY = YX$

Associative

3. $X + (Y + Z) = (X + Y) + Z$
4. $X (YZ) = (XY)Z$

Distributive

5. $X(Y+Z) = XY+ XZ$
6. $X + YZ = (X+Y)(X+Z)$

DeMorgan's

7. $(X + Y)' = X'Y'$
8. $(XY)' = X' + Y'$

	OR
1	$x+1 = 1$
2	$x+x' = 1$
3	$x+x = x$
4	$x+0 = x$
5	$(x')' = x$
6	$x+y = y+x$
7	$x+(y+z) = (x+y)+z$
8	$x.(y+z) = x.y+x.z$
9	$(x+y)' = x'.y'$
10	$x+(x.y) = x$

	AND
1	$x.1 = x$
2	$x.x' = 0$
3	$x.x = x$
4	$x.0 = 0$
5	$(x')' = x$
6	$x.y = y.x$
7	$x.(y.z) = (x.y).z$
8	$x+y.z = (x+y).(x+z)$
9	$(x.y)' = x'+y'$
10	$x.(x+y) = x$

Algebraic Manipulation

- Using basic identities a boolean function can be simplified
- For example

$$F = X'YZ + X'YZ' + XZ$$

Algebraic Manipulation

- Using basic identities a boolean function can be simplified
- For example

$$
\begin{aligned}
F &= X'YZ + X'YZ' + XZ \\
&= X'Y(Z + Z') + XZ \\
&= X'Y \cdot 1 + XZ \\
&= X'Y + XZ
\end{aligned}
$$

Algebraic Manipulation Examples

1. $X + XY =$
2. $XY + XY' =$
3. $X + X'Y =$
4. $X (X + Y) =$
5. $(X + Y)(X + Y') =$

6. $X(X' + Y) =$

Algebraic Manipulation Examples

1. $X + XY = X(1+Y) = X$

Algebraic Manipulation Examples

2 . $XY + XY' = X(Y + Y') = X$

Algebraic Manipulation Examples

3 . $X + X'Y = (X + X')(X + Y) = X + Y$

Algebraic Manipulation Examples

4 . $X (X + Y) = X + XY = X (1 + Y) = X$

Algebraic Manipulation Examples

1. $X + XY = X(1+Y) = X$
2. $XY + XY' = X(Y + Y') = X$
3. $X + X'Y = (X + X')(X + Y) = X + Y$
4. $X(X + Y) = X + XY = X(1 + Y) = X$
5. $(X + Y)(X + Y')$
$$\begin{aligned} &= X + XY' + XY + YY' \\ &= X(1 + Y') + XY \\ &= X \cdot X + XY \\ &= X + XY \\ &= X(1 + Y) \\ &= X \end{aligned}$$
6. $X(X' + Y) = XX' + XY = XY$

Algebraic Manipulation Examples

5 . (X + Y)(X + Y')

$$= X + XY' + XY + YY'$$
$$= X(1 + Y') + XY$$
$$= X \cdot 1 + XY$$
$$= X + XY$$
$$= X(1 + Y)$$
$$= X$$

Algebraic Manipulation Examples

6 . $X(X' + Y) = XX' + XY = XY$

Algebraic Manipulation Examples

1. $X + XY = X(1+Y) = X$
2. $XY + XY' = X(Y + Y') = X$
3. $X + X'Y = (X + X')(X + Y) = X + Y$
4. $X(X + Y) = X + XY = X(1 + Y) = X$
5. $(X + Y)(X + Y')$
   $= X + XY' + XY + YY'$
   $= X(1 + Y') + XY$
   $= X \cdot 1 + XY$
   $= X + XY$
   $= X(1 + Y)$
   $= X$
6. $X(X' + Y) = XX' + XY = XY$

Example

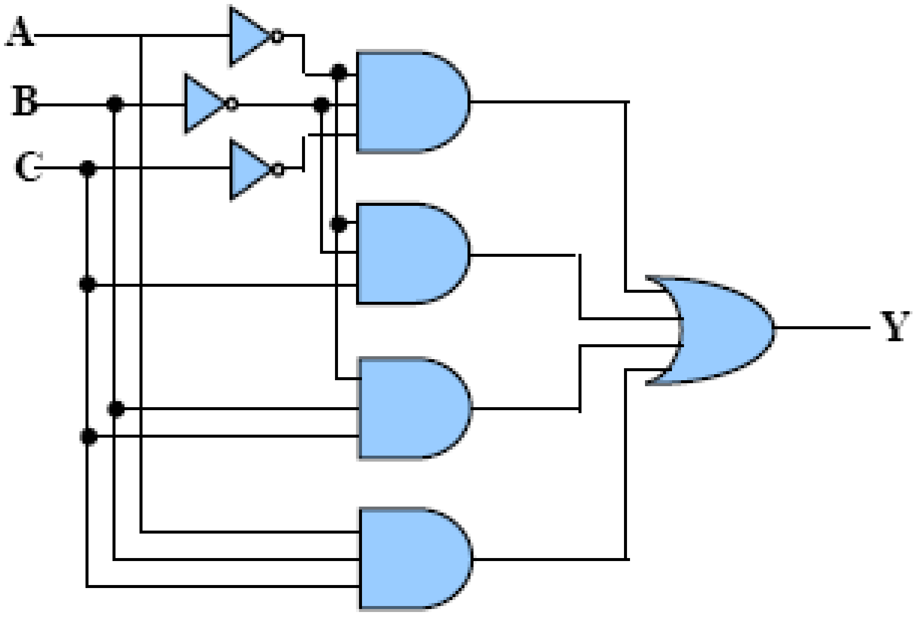
- Simplify the following Boolean expression and implement logic circuit for it

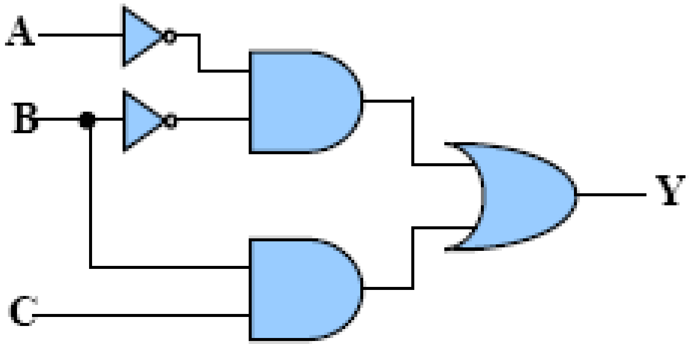
$$Y = A'B'C' + A'B'C + A'BC + ABC$$

$$Y = A'B'C' + A'B'C + A'BC + ABC$$

$$= A'B'(C'+C) + BC(A'+A)$$

$$= A'B' + BC$$


A
B
C
Y


A
B
C
Y

Combinational logic circuits

NOR & NAND

Basic Identities of Boolean Algebra

1. $X + Y = Y + X$
2. $XY = YX$

Commutative

3. $X + (Y + Z) = (X + Y) + Z$
4. $X (YZ) = (XY)Z$

Associative

5. $X(Y+Z) = XY+ XZ$
6. $X + YZ = (X+Y)(X+Z)$

Distributive

7. $(X + Y)' = X'Y'$
8. $(XY)' = X' + Y'$

DeMorgan's

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

$$(AB)' = A' + B'$$

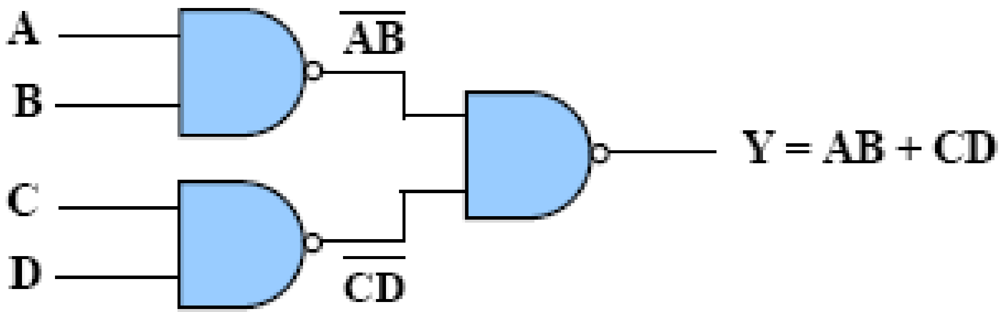
نظرية ديمورجان الأولى: $\overline{A + B} = \overline{A} \bullet \overline{B}$

نظرية ديمورجان الثانية: $\overline{A \bullet B} = \overline{A} + \overline{B}$

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

NAND — Negative-OR

Example



$$Y = \overline{(\overline{AB})(\overline{CD})}$$

From Demorgan Theorem

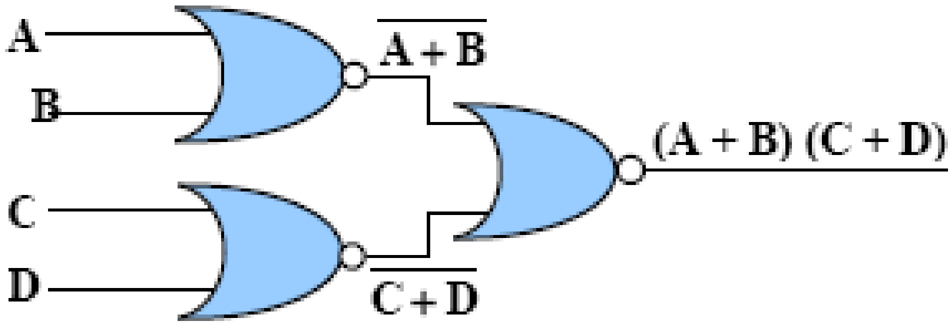
$$Y = \overline{\overline{AB}} + \overline{\overline{CD}}$$

Then Y = AB + CD

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

NOR

Negative-AND



$$Y = \overline{\overline{(A+B)} + \overline{(C+D)}}$$

From Demorgan Theorem

$$Y = \overline{\overline{(A+B)}} \bullet \overline{\overline{(C+D)}}$$

$$Y = (A+B) \bullet (C+D)$$

طبق نظريات ديمورجان على التعبير البوليني التالي:

$$Y = \overline{(A + \overline{B} + \overline{C}) \bullet (\overline{A} + B + \overline{C})}$$

$$Y = \overline{(A + \overline{B} + \overline{C}) \bullet (\overline{A} + B + \overline{C})}$$

$$= \overline{(A + \overline{B} + \overline{C})} + \overline{(\overline{A} + B + \overline{C})}$$

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

طبق نظريات ديمورجان على التعبير البوليني التالي:

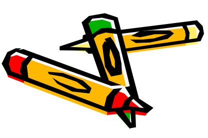
$$Y = \overline{(\overline{A} + B) + CD}$$

$$Y = \overline{(\overline{A} + B) + CD}$$

$$= \overline{(\overline{A} + B)}.\overline{CD}$$

$$= (\overline{\overline{A}}.\overline{B})(\overline{C} + \overline{D})$$

$$= A\overline{B}(\overline{C} + \overline{D})$$



Thank you


THANKS


H A K
T N S


THANKS