MASTERS OF COMPUTER APPLICATIONS

SEMESTER 1

DISCRETE MATHEMATICS AND GRAPH THEORY

Unit 11

Logic Gates and Circuits

Table of Contents

1. INTRODUCTION

In digital circuits, a logic gate is a straightforward switching circuit that decides whether an input pulse may go to the output. Logic gates, which carry out the many logical operations needed by any digital circuit, are the fundamental components of a digital circuit. These have more than one input but only one output.

A logic gate's output is determined by the combination of inputs sent through it. Boolean algebra is used by logic gates to carry out logical operations. Almost every digital device we regularly use has logic gates. The architecture of modern phones, computers, tablets, and memory devices all make use of logic gates.

1.1. Objectives:

At studying this unit, you should be able to:

- *Understand basic logic Gates and Truth Tables.*
- *Apply logic gates to logic circuits.*

2. LOGIC GATES

Logic gate is an electronic circuit having one or more inputs and only one output and acts as a building block for digital circuits.

In other words, a logic gate is a digital gate that allows data to be transferred. Logic gates, use logic to determine whether or not to pass a signal. Logic gates, on the other hand, govern the flow of information based on a set of rules.

Logic gates are the basic building blocks of any digital system. Logic gates are electronic circuits having one or more than one input and only one output.

3. TYPES OF LOGIC GATES

The relationship between the input and the output is based on a certain logic. Based on this, The following types of logic gates are commonly used:

a) AND
b) OR
c) NOT
d) NOR
e) NAND
f) XOR
g) XNOR

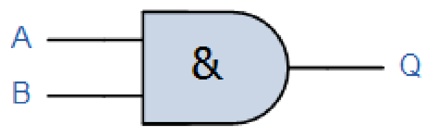
4. BASIC LOGIC GATES

4.1. AND Gate

An AND gate has a single output and two or more inputs. When all of the inputs are 1, the output of this gate is 1.

The AND gate's Boolean expression is **Q=A.B**, if there are two inputs A and B.

An AND gate's symbol and truth table are as follows:

Symbol	Truth Table		
 2-input AND Gate	A	B	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolean Expression Q = A.B	Read as A AND B gives Q		

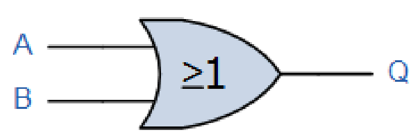
Note: The Boolean Expression for a two input AND gate can be written as: A.B or just simply AB without the decimal point.

4.2. OR Gate

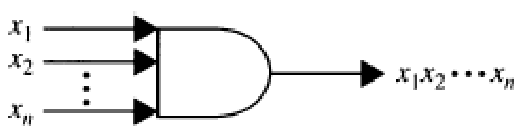
An OR gate has a single output and two or more inputs. For this Logic gate, the output will be 1 if at least one of the inputs is 1.

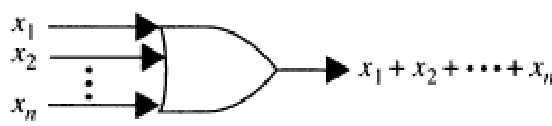
The OR gate's Boolean expression is **Q=A+B,** if there are two inputs A and B.

An OR gate's symbol and truth table are as follows:

Symbol	Truth Table		
 2-input OR Gate	A	B	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1
Boolean Expression Q = A+B	Read as A OR B gives Q		

Note



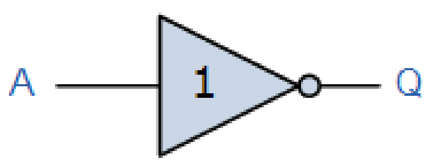


4.3. NOT Gate

The NOT gate is a basic one-input, one-output gate. When the input is 1, the output is 0, and vice versa. A NOT gate is sometimes called an inverter because of its feature.

If there is only one input A, the output may be calculated using the Boolean equation

The NOT gate's symbol and truth table are as follows:

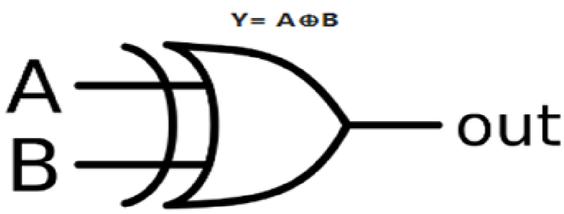
Symbol	Truth Table	
 Inverter or NOT Gate	A	Q
	0	1
	1	0
Boolean Expression Q = NOT A or $\overline{A}$	Read as inversion of A gives Q	

5. OTHER LOGICAL GATES

5.1. Ex-OR gate

The 'Exclusive-OR' gate is a circuit which will give a high output if either, but not both of its two inputs are high.

An encircled plus sign ($\oplus$) is used to show the Ex-OR operation.

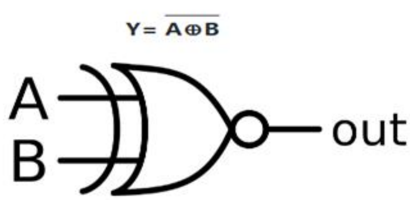


A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

5.2. Ex-NOR gate

The 'Exclusive-NOR' gate circuit does the opposite to the EX-OR gate. It will give a low output if either, but not both of its two inputs are high.

The symbol is an EX-OR gate with a small circle on the output. The small circle represents inversion.



XNOR Truth Table

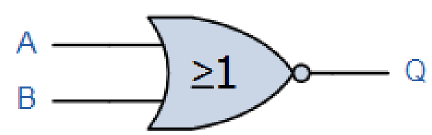
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

6. UNIVERSAL LOGIC GATES

6.1. NOR Gate

A NOR gate, sometimes known as a “NOT-OR” gate, consists of an OR gate followed by a NOT gate. This gate’s output is 1 only when all of its inputs are 0.

The Boolean statement for the NOR gate is Q=(A+B)’= (not (A or B)), if there are two inputs A and B. The NOR gate’s symbol and truth table are as follows:

Symbol	Truth Table		
 2-input NOR Gate	A	B	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	0
Boolean Expression Q = $\overline{A+B}$	Read as A OR B gives NOT-Q		

Note:

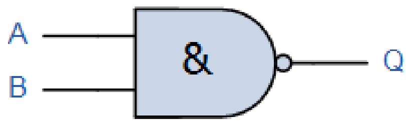
- NOR gate is actually a combination of two logic gates: OR gate followed by NOT gate.
- So its output is complement of the output of an OR gate.
- This gate can have minimum two inputs, output is always one.
- By using only NOR gates, we can realize all logic functions: AND, OR, NOT, Ex-OR, Ex-NOR, NAND. So this gate is also called universal gate.

6.2. NAND Gate

A NAND gate, sometimes known as a ‘NOT-AND’ gate, is essentially a Not gate followed by an AND gate. This gate’s output is 1 only if atleast one of the inputs is 0.

If there are two inputs A and B, the Boolean expression for the NAND gate is Y=(A.B)’

The Boolean statement for the NAND gate is Q=(A.B)'= (not (A and B)), if there are two inputs A and B. The NAND gate's symbol and truth table are as follows:

<table>
<tr><th>Symbol</th><th colspan="3">Truth Table</th></tr>
<tr><td rowspan="5">2-input NAND Gate</td><td>A</td><td>B</td><td>Q</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>Boolean Expression $Q = \overline{A.B}$</td><td colspan="3">Read as A AND B gives NOT-Q</td></tr>
</table>

Note

- NAND gate is actually a combination of two logic gates i.e. AND gate followed by NOT gate. So its output is complement of the output of an AND gate.
- This gate can have minimum two inputs, output is always one.
- By using only NAND gates, we can realize all logic functions: AND, OR, NOT, X-OR, X-NOR, NOR. So this gate is also called as universal gate.

6.3. NAND gates as NOT gate

A NOT produces complement of the input. It can have only one input, tie the inputs of a NAND gate together. Now it will work as a NOT gate.

Its output is

Y = (A.A)'

Y = (A)'

NOT (inverter)

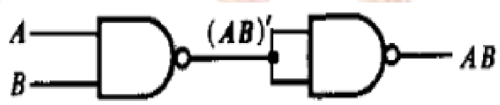
Input	Output
A	A'
0	1
1	0

6.4. NAND gates as AND gate

A NAND produces complement of AND gate. So, if the output of a NAND gate is inverted, overall output will be that of an AND gate.

Y = ((A.B)')'

Y = (A.B)



AND

Input		Output
A	B	F = A.B
0	0	0
0	1	0
1	0	0
1	1	1

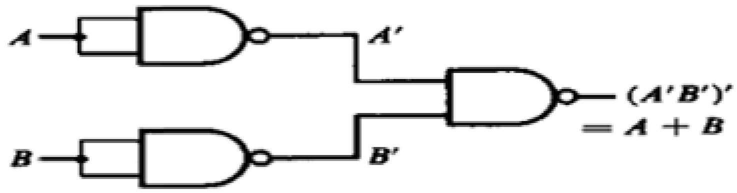
6.5. NAND gates as OR gate

From DeMorgan's theorems:

(A.B)' = A' + B'

(A’.B’)’ = A’’ + B’’ = A + B

So, give the inverted inputs to a NAND gate, obtain OR operation at output.



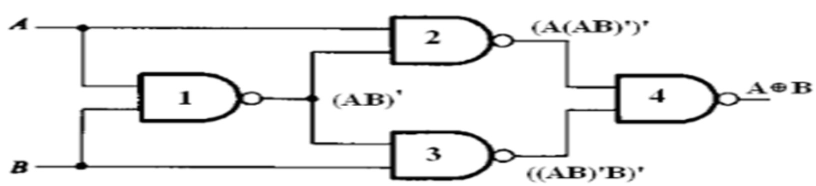
OR

A	B	X = A+B
0	0	0
0	1	1
1	0	1
1	1	1

6.6. NAND gates as Ex-OR gate

The output of a two input Ex-OR gate is shown by:

Y = A’B + AB’.



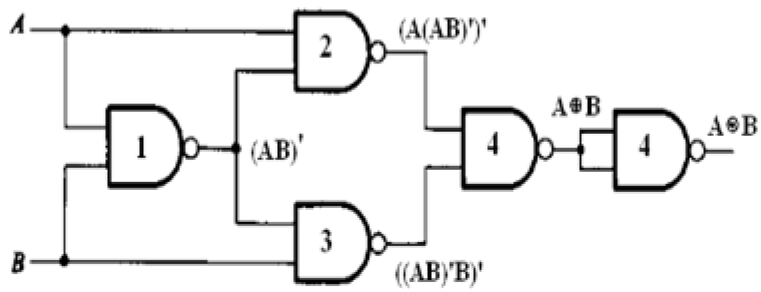
X-OR

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

6.7. NAND gates as Ex-NOR gate

Ex-NOR gate is actually Ex-OR gate followed by NOT gate. So give the output of Ex-OR gate to a NOT gate, overall output is that of an Ex-NOR gate.

Y = AB+ A’B’



X-NOR

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

6.8. NOR gates as NOT gate

A NOT produces complement of the input. It can have only one input, tie the inputs of a NOR gate together. Now it will work as a NOT gate.

Its output is

Y = (A+A)’

Y = (A)’

NOT

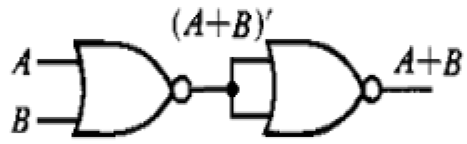
Input	Output
A	A’
0	1
1	0

6.9. NOR gates as OR gate

A NOR produces complement of OR gate. So, if the output of a NOR gate is inverted, overall output will be that of an OR gate.

Y = ((A+B)’)’

Y = (A+B)



OR

A	B	X = A+B
0	0	0
0	1	1
1	0	1
1	1	1

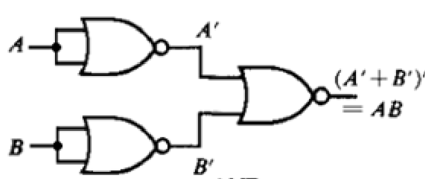
6.10. NOR gates as AND gate

From DeMorgan’s theorems:

(A+B)’ = A’B’

(A’+B’)’ = A’’B’’ = AB

So, give the inverted inputs to a NOR gate, obtain AND operation at output.



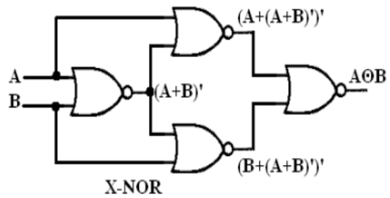
AND

Input		Output
A	B	F = A.B
0	0	0
0	1	0
1	0	0
1	1	1

6.11. NOR gates as Ex-NOR gate

The output of a two input Ex-NOR gate is shown by:

Y = AB + A’B’.

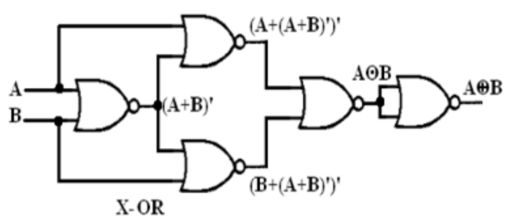


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

6.12. NOR gates as Ex-OR gate

Ex-OR gate is actually Ex-NOR gate followed by NOT gate. So give the output of Ex-NOR gate to a NOT gate, overall output is that of an Ex-OR gate.

Y = A’B+ AB’



A	B	A **XOR** B
0	0	0
0	1	1
1	0	1
1	1	0

7. CANONICAL FORMS

Any Boolean expression that can be expressed as a sum of Minterms or product of Maxterms are called canonical forms. This can be done in two ways:

1) Boolean Postulates and Theorems

2) Truth table

A Boolean function can be expressed canonically from a given truth table by forming

7.1. Minterm

Minterm for each combination of the variables that produces a 1 in the function and then taking the OR of all those terms.

7.2. Maxterm

Maxterm for each combination of the variables that produces a 0 in the function and then taking the AND of all those terms.

				Minterms		Maxterms
X	Y	Z		Product Terms		Sum Terms
0	0	0		$m_0 = \overline{X} \cdot \overline{Y} \cdot \overline{Z} = \min(\overline{X}, \overline{Y}, \overline{Z})$		$M_0 = X + Y + Z = \max(X, Y, Z)$
0	0	1		$m_1 = \overline{X} \cdot \overline{Y} \cdot Z = \min(\overline{X}, \overline{Y}, Z)$		$M_1 = X + Y + \overline{Z} = \max(X, Y, \overline{Z})$
0	1	0		$m_2 = \overline{X} \cdot Y \cdot \overline{Z} = \min(\overline{X}, Y, \overline{Z})$		$M_2 = X + \overline{Y} + Z = \max(X, \overline{Y}, Z)$
0	1	1		$m_3 = \overline{X} \cdot Y \cdot Z = \min(\overline{X}, Y, Z)$		$M_3 = X + \overline{Y} + \overline{Z} = \max(X, \overline{Y}, \overline{Z})$
1	0	0		$m_4 = X \cdot \overline{Y} \cdot \overline{Z} = \min(X, \overline{Y}, \overline{Z})$		$M_4 = \overline{X} + Y + Z = \max(\overline{X}, Y, Z)$
1	0	1		$m_5 = X \cdot \overline{Y} \cdot Z = \min(X, \overline{Y}, Z)$		$M_5 = \overline{X} + Y + \overline{Z} = \max(\overline{X}, Y, \overline{Z})$
1	1	0		$m_6 = X \cdot Y \cdot \overline{Z} = \min(X, Y, \overline{Z})$		$M_6 = \overline{X} + \overline{Y} + Z = \max(\overline{X}, \overline{Y}, Z)$
1	1	1		$m_7 = X \cdot Y \cdot Z = \min(X, Y, Z)$		$M_7 = \overline{X} + \overline{Y} + \overline{Z} = \max(\overline{X}, \overline{Y}, \overline{Z})$

7.3. Sum-of-products (SOP)

When two or more product terms are summed by Boolean addition, the resulting expression is a sum-of-products (SOP).

Ex:

a). AB + ABC

b). ABC + CDE + BCD

c). AB + BCD + AC

d). A + ABC + BCD

7.4. Product of Sums (POS)

When two or more sum terms are multiplied, the resulting expression is a product-of-sums (POS).

Ex:

a). (A + B)(A + B + C)

b). (A + B + C)(C + D + E)(B + C + D)

c). (A + B)(A + B + C)(A + C)

d). A(A + B + C)(B + C + D)

7.5. Standard Sum-of-products (SOP)

A standard SOP expression is one in which all the variables in the domain appear in each product term in the expression.

Ex:

ABCD + ABCD + ABCD

7.6. Standard Product of Sums (POS)

A standard POS expression is one in which all the variables in the domain appear in each sum term in the expression.

Example:

(A + B + C + D)(A + B + C + D)(A + B + C + D)

Note: Any boolean expression can be reduced into either standard SOP or standard POS using either Boolean postulates and theorems or using truth table.

Example:

Express the Boolean function $F = x + yz$ as a sum of Minterms/Canonical form using truth table.

Solution:

The truth table for the given Boolean expression is given as follows:

Now, from the truth table list out the minterms in which the

given expression truth value is 1.

$$F = x'yz + xy'z' + xy'z + xyz' + xyz = m_3 + m_4 + m_5 + m_6 + m_7 = \sum(3,4,5,6,7)$$

To find the product of maxterms from the truth table, it is

enough to find F' and list out the maxterms in which the given expression truth value is 1.

$$F' = (x + y + z).(x + y + z').(x + y' + z) = M_0 + M_1 + M_2 = \pi(0,1,2)$$

$F = x + yz$

x	y	z	Minterms	F	F'
0	0	0	$m_0 = x'y'z'$	0	1
0	0	1	$m_1 = x'y'z$	0	1
0	1	0	$m_2 = x'yz'$	0	1
0	1	1	$m_3 = x'yz$	1	0
1	0	0	$m_4 = xy'z'$	1	0
1	0	1	$m_5 = xy'z$	1	0
1	1	0	$m_6 = xyz'$	1	0
1	1	1	$m_7 = xyz$	1	0

8. SELF-ASSESSMENT QUESTIONS

1. Sketch the logic circuits that produce the following outputs:

$$(x + y)\bar{y}$$

2. Sketch the logic circuits that produce the following outputs:

$$(xy + xz + yz)$$

3. Sketch the logic circuits that produce the following outputs:

$$f(x, y) = xy + \bar{x}\,\bar{y}$$

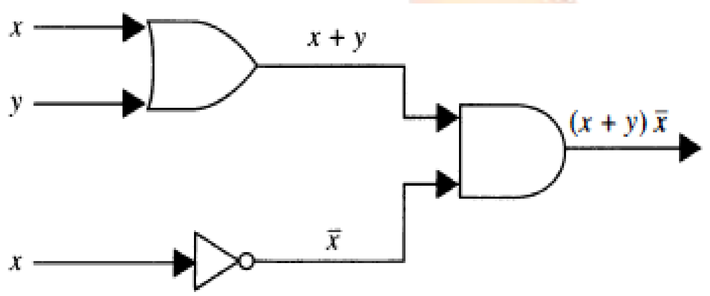
4. Sketch the logic circuits that produce the following outputs:

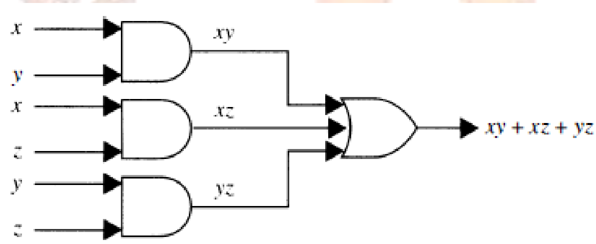
$$x + y + z)\ (\bar{x}\,\bar{y}\,\bar{z}$$

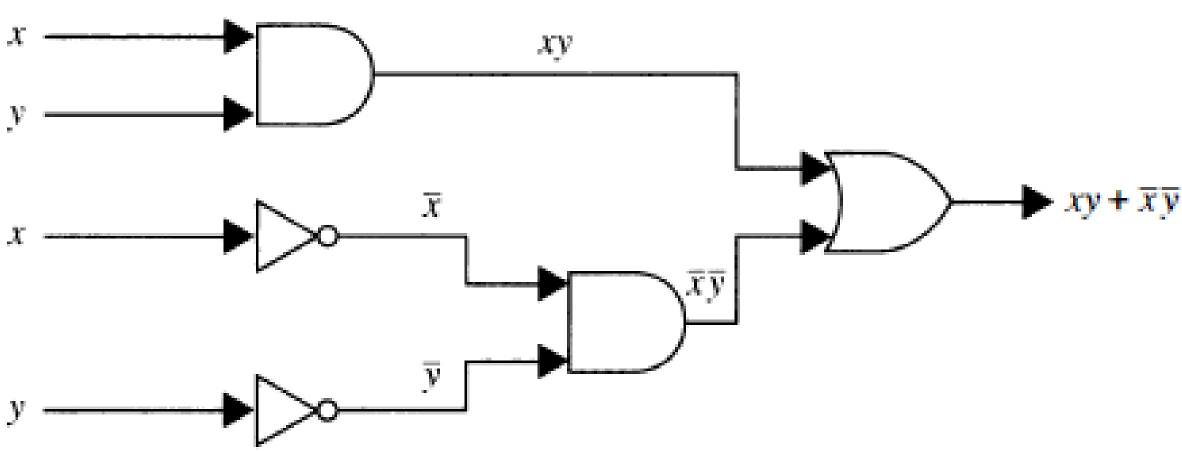
9. SUMMARY

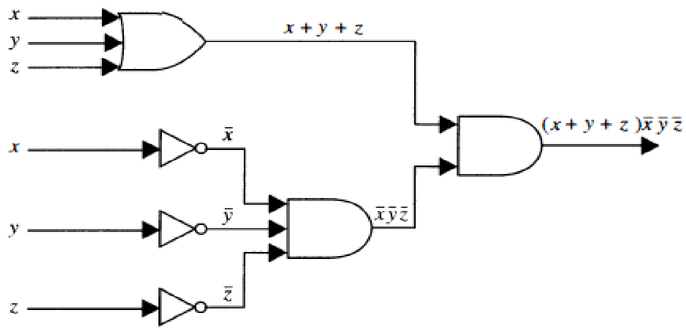
This chapter we studied about various types of Logical gates and implementing the concepts of logical gates to obtain various circuit.

10. ANSWERS TO SELF ASSESSMENT QUESTIONS

1. 

2. 

3. 

4. 

11. REFERENCES

1. "R.P. Jain, "Modern digital Electronics", Tata McGraw Hill, 4th edition, 2009.
2. Douglas Perry, "VHDL", Tata McGraw Hill, 4th edition, 2002.
3. W.H. Gothmann, "Digital Electronics- An introduction to theory and practice", PHI, 2nd edition, 2006. 2012.