Progressive Education Society's

Modern College of Arts, Science and Commerce, Shivajinagar, Pune – 5

(Autonomous College)

First Year of B. Sc. (2019 Course)

SEMESTER – I Paper -II

Course Code :19ScEleU102 **Course Name: Fundamentals of Digital Electronics**

Course Contents

Chapter 1	**Number Systems**	16 lectures
	Introduction to decimal, binary and hexadecimal number systems and their inter conversions, Unsigned and Signed binary number representations, Rules of binary addition and subtraction, Binary addition and subtraction, Subtraction using 1's and 2's complements, BCD code, Excess-3 code, Gray code, Alphanumeric representation in ASCII codes, Code conversion –binary to gray, gray to binary.	
Chapter 2	**Logic Gates**	7 lectures
	Positive and Negative Logic, OR, AND, NOT gates, NAND, NOR, EX-OR, EX-NOR gates (Symbol and truth table).	
Chapter 3	**Boolean Algebra**	12 lectures
	Boolean algebra and Boolean laws: Commutative, Associative, Distributive, AND, OR and Inversion laws, De Morgen's theorem, NAND, NOR as universal gate, K-map Basics, Min terms, Max terms, Boolean expression in SOP and POS form, Simplifications of Logic expressions using Boolean algebra rules and Karnaugh map (up to 4 variables), Implementation of Boolean expressions using basic gates.	
Chapter 4	**Experiential Learning**	1 lecture
	Group Discussion / Field Work / Mini Project.	

Text/ Reference Books:

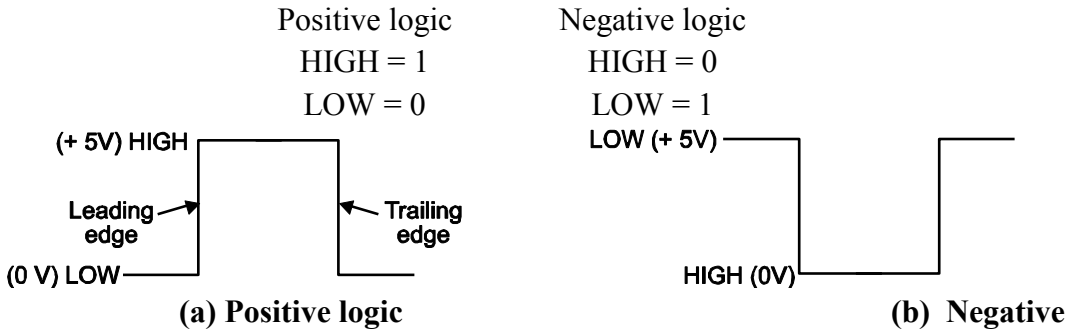
1. Digital Electronics: Jain R.P., Tata McGraw Hill
2. Digital Principles and Applications: Malvino Leach, Tata McGraw-Hill
3. Digital Fundamentals: Floyd T.M., Pearson Education

Chapter 2	**Logic Gates**	7 lectures
	Positive and Negative Logic, OR, AND, NOT gates, NAND, NOR, EX-OR, EX-NOR gates (Symbol and truth table).	

1. Positive and Negative Logic:

In a digital system there are two discrete levels HIGH (1) and LOW (0). If the higher of the two voltages represents a 1 and lower voltage represents 0, the system is called *positive logic* system. On the other hand, if lower voltage represents a 1 and higher voltage represents 0 we have *negative logic* system.

Suppose that +5V and 0V are our logic level voltages. We will designate +5V as HIGH (1) and 0V as LOW (0). So positive and negative logic can be defined as :



(a) Positive logic **(b) Negative logic**

Fig. : Digital signal representation

From Fig.(a) above , + 5V will be considered as HIGH level in positive logic and LOW level in negative logic system.

Similarly, voltage level 0V will be considered as LOW in the positive logic system and HIGH level in the negative logic system, as in Fig.(b).

2. Logic Gates

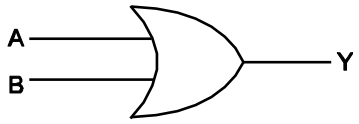
In a digital system, there are only few basic operations performed irrespective of the complexities of the system. These operations may be required to be performed number of times in a digital computer or digital control system.

The basic logic gates are AND, OR, NOT etc. These basic gates can be combined to perform other important logic operations like NAND, NOR and EX-NOR gates. So these are called as *derived gates*.

Any Boolean (or logic) expression can be realized by using the AND, OR and NOT gates. NAND, NOR operations can be derived from it. These operations have become very popular and are widely used because either NAND, NOR gates are sufficient for the realization of any logical expression. Because of this reason, NAND and NOR gates are known as universal gates.

3. Logic OR gate:

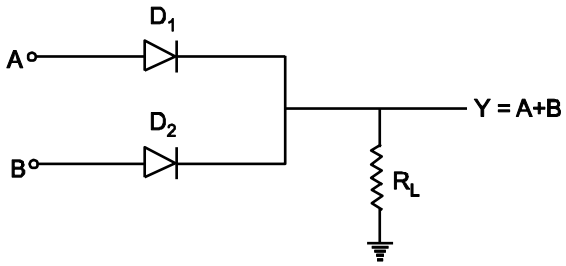
An OR gate performs logical addition, more commonly known as OR function. It has two or more inputs and one output. The logic symbol of OR gate is shown in Fig. (a).



(a) Symbol

The inputs of two input OR gates are labelled as A and B. Output is labelled as Y.

The operation of OR gate is such that a HIGH on the output is produced when any of the inputs are HIGH. Output is LOW only when all of the inputs are LOW. In certain situation, if we want a output HIGH, when one or more of its inputs is HIGH, OR gate can be used. Consider logic OR gate circuit using diodes as shown in fig. (b)



(b) Diode circuit of OR gate

Inputs		Output
A	**B**	**Y = A + B**
0	0	0
0	1	1
1	0	1
1	1	1

(c) Truth table

Fig. 1.6

Let us assume input voltage 1 (High state) or + 5V and 0 (Low state) is 0 V.

For different input combinations, we will have the following cases :

Case I : A and B both LOW i.e. A = 0 and B = 0. In this case, both diodes D_1 and D_2 are non-conducting, therefore, Y output is LOW. i.e. Y = 0.

Case II : A is LOW and B is HIGH i.e. A = 0 and B = 1.

The high B input voltage of the diode D_2 will conduct as it is under forward bias condition. The conducting current will flow through resistor R_L and high voltage will develop at the output. This will make Y = 1 (High logic). The diode D_1 is reversed bias.

Case III : A is HIGH and B is LOW i.e. A = 1 and B = 0.

The diode D_1 will conduct and D_2 is non-conducting. The conducting current in this case also flows through resistor R_L and HIGH voltage will develop at the output. This will make Y = 1 (High logic).

Case IV : A and B both are HIGH i.e. A = 1 and B = 1. Diodes D_1 and D_2 will conduct and therefore output is at high logic level i.e. Y = 1.

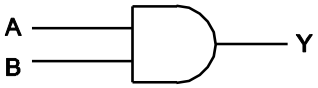
This overall operation of OR gate is indicated by Truth table in fig. (c)

4. Logic AND gate:

The AND gate performs logical multiplication, more commonly known as AND function.

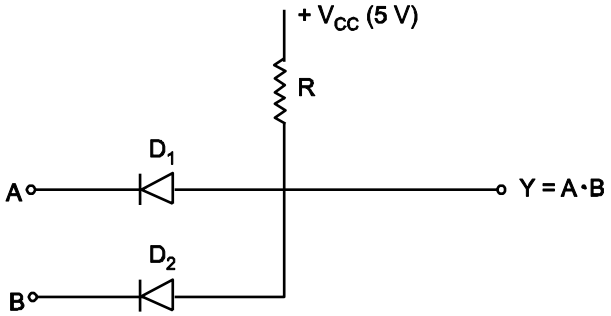
The AND gate is composed of two or more inputs and single output. The output of AND gate is high (1) only when all the inputs are high (1).

A logic gate can be used as diode, transistor, FET or combination of these elements.



(a) Symbol

The inputs of the two input AND gates are labelled as A and B and output is labelled as Y.



(b) Simplified circuit with diode

Inputs		Output
A	**B**	**Y**
0	0	0
0	1	0
1	0	0
1	1	1

(c) Truth table

Fig.

The AND gate can be designed using diodes. Fig. (b) shows two input AND gates using diodes. There are only four possible input cases.

Case I : A is low and B is low. With this situation both diodes D_1 and D_2 are forward bias by supply voltage and will conduct. Because of this the output voltage is ideally zero. This means Y is low. i.e. $Y = 0$

Case II : A is low and B is high since diode D_1 is forward bias pulling the output down to a low voltage. The diode D_2 is reversed bias. The output $Y = 0$.

Case III : A is high and B is low. The diode D_2 is forward bias and pulling the output down to a low voltage. The diode D_1 is reversed bias, therefore $Y = 0$.

Case IV : A is high and B is high with both inputs at + 5V. Both diodes are non-conducting because the voltage across each is zero. Therefore, current will not flow through R and output Y is at high (logic 1) i.e. $Y = 1$.

This overall operation of AND gate is indicated by Truth table in fig. (c)

5. Logic NOT gate:

The NOT gate is also called *inverter* as it inverts the input signal. The inverter (NOT circuit) performs a basic logic function called inversion or complementation. It is called NOT because its output is NOT the same as input. The purpose of NOT or inverter is to change one logic level to the opposite level. In terms of bits, it changes 1 to 0 and 0 to 1.

(a) **(b)**

Input A	**Output Y = $\overline{A}$**
0	1
1	0

(c) Truth table

Fig. 1.7 : NOT gate

Fig. 1.7 (a) and (b) shows standard symbols for NOT gate. As shown in Fig. 1.7, NOT gate has one input A and one output Y. Its logic expression is,

$$Y = \text{NOT } A$$

$$= \overline{A}$$

and is read as "Y equals NOT A" or Y equals complement of A.

If $A = 0$, $Y = \text{NOT } 0 = 1$

On the other hand,

if $A = 1$, $Y = \text{NOT } 1 = 0$

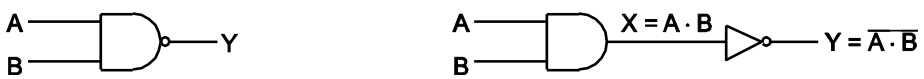
The presence of small circle, known as the bubble, always denotes inversion in digital circuits.

6. Logic NAND gate:

Any Boolean (or logic) expression can be realized by using the AND, OR and NOT gate as discussed before. From these three operations, two more operations have to be derived : the NAND and NOR operations. These operations are very popular and are widely used.

NAND and NOR gates are known as *universal gates* because it can perform any logical function of AND, OR and NOT gate. NAND or NOR gates are sufficient for the realization of any logical expression.

The NOT-AND operation is known as the NAND operation. It can be obtained by connecting a NOT gate in the output of an AND gate as it gives the output in the form of logic expression : $Y = \overline{A \cdot B}$



(a) Logic symbol **(b) NAND gate connection**

This gate gives an output 1 if its both inputs are not 1. In other words, it gives an output 1 if either A or B or both are 0.

Inputs		Outputs	
A	**B**	**AND X = A·B**	**NAND Y = $\overline{A \cdot B}$**
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

(c) Truth table

Fig. 1.10 : NAND gate

The truth table for two input NAND gates is given in Fig. 1.10 (c). It is just opposite of the truth table for AND gate.

The diode-transistor equivalent of a NAND gate is shown in Fig. 1.11.

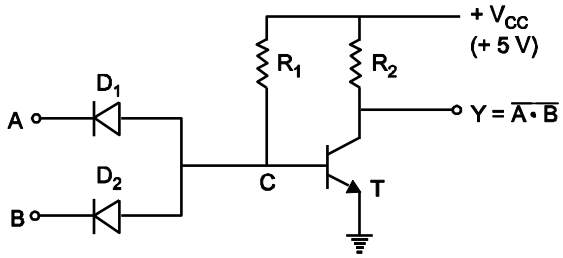


Fig. 1.11 : NAND equivalent circuit

It is seen that, point C would be driven to ground when either D_1 or D_2 or both D_1 and D_2 conduct.

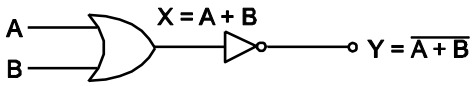
It represents input conditions from truth table. AB = 00, 01 and 10 transistor is cut-off and hence Y output goes to + V_{CC} or logic level 1.

When AB = 11 i.e. A = 1 and B = 1 the point C is at + 5V and transistor goes saturation, output becomes 0 V or logic level 0.

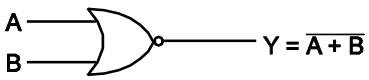
7. Logic NOR Gate:

It is a combination of NOT OR gate. It can be made by connecting an inverter to the output of OR gate.

Output of NOR gate will be complement of output of OR gate. The logical operation of NOR gate is such that a LOW output occurs when any of its inputs are HIGH. It have HIGH output only when all inputs are LOW.



(a) NOR gate connection



(b) Symbol

Inputs		Outputs	
A	**B**	**OR** $X = A + B$	**NOR** $Y = \overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

(c) Truth table

Fig. 1.12 : NOR gate

From Fig. 1.12 (c) inputs are labelled as A and B and output is labelled as Y.

$$\therefore \quad Y = \overline{A + B}$$

Note that this operation results in opposite to that of OR gate.

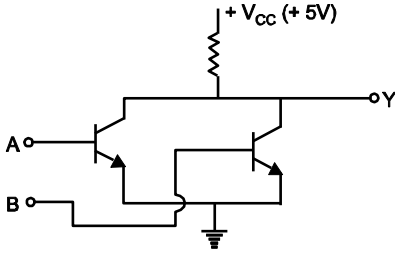
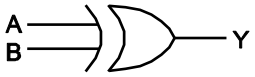


Fig. 1.13 : Equivalent circuit for NOR gate

A transistor equivalent circuit of NOR gate is shown in Fig. 1.13. As seen Y is 1 only when both transistors are cutoff i.e. A = 0 and B = 0. For any other input condition like, AB = 01, 10 and 11 both transistors saturate forcing point Y to goto LOW state.

8. Logic EX-OR gate:

The EX-OR operation is widely used in digital circuits. It is not a basic operation and can be performed using the basic gates AND, OR and NOT or universal gate NAND or NOR.



(a) Symbol

Inputs		Output
A	**B**	**Y = A ⊕ B**
0	0	0
0	1	1
1	0	1
1	1	0

(b) Truth table

Fig. 1.16

$$Y = \bar{A}\,B + A\bar{B}$$

i.e. $$Y = A \oplus B.$$

The circle around plus ⊕ sign is worth noting.

An EX-OR gate is special case of general OR gate. In general OR gate either input A or B or both must be 1 in order to get 1 at output. But in EX-OR case, the situation is different i.e. either input A or B, but not both should be 1 for getting 1 at the output.

This circuit is also called an *inequality comparator* or *detector* because it produces an output only when the two inputs are different.

EX-OR gate can be constructed using AND, OR and NOT gates.

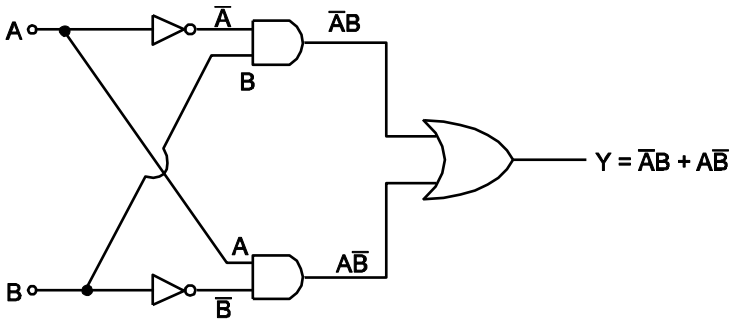


Fig. 1.17

The upper AND gate forms the product $\bar{A}\,B$ while the lower AND gate produces $A\bar{B}$, the output of the OR gate is $Y = \bar{A}\,B + A\bar{B}$

Case I : If A and B both are LOW, both AND gates have low output, therefore, the final output Y is LOW.

i.e. $A = B = 0$

$\bar{A} = \bar{B} = 1$

$\therefore \quad Y = 1.0 + 0.1$

$= 0 + 0$

$= 0$

Case II : If A = 0 (LOW) and B = 1 (HIGH), the upper AND gate has a HIGH output, so OR gate has HIGH output.

i.e. $A = 0$ and $B = 1$

$\bar{A} = 1$ and $\bar{B} = 0$

$Y = \bar{A}\,B + A\bar{B}$

$= 1.1 + 0.0$

$= 1 + 0$

$= 1$

Case III : Likewise A = 1 (HIGH) and B = 0 (LOW), Y output becomes 1 (HIGH).

Case IV : If A = 1 (HIGH) and B = 1 (HIGH), output Y becomes LOW.

i.e. $A = 1$ and $B = 1$

$\bar{A} = 0$ and $\bar{B} = 0$

$Y = 0.1 + 1.0$

$= 0 + 0$

$= 0$

9. Logic EX-NOR Gate:

Basically the "Exclusive-NOR" gate is a combination of the Exclusive-OR gate and the NOT gate but has a truth table similar to the standard NOR gate in that it has an output that is normally at logic level "1" and goes "LOW" to logic level "0" when ANY of its inputs are at logic level "1". The circuit symbol for EX-NOR gate s shown in fig. (a)

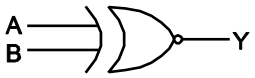


Fig. (a) EX-NOR gate

However, an output "1" is only obtained if BOTH of its inputs are at the same logic level, either binary "1" or "0". For example, "00" or "11".

This input combination would then give us the Boolean expression of:

$$Y = \overline{A \oplus B} = A.B + \overline{A}.\overline{B}$$

Exclusive-NOR Gate is the reverse or "Complementary" form of the Exclusive-OR gate.

Fig. (b) Shows truth table for EX-NOR gate.

Inputs		**Output**
A	**B**	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

(b) Truth table

Fig. 1.16

Hence The output of a two-input EX-NOR gate is a logic '1' when the inputs are like and a logic '0'when they are unlike.

As a summary the truth table of EXNOR gate by using equation

$$Y = \overline{A \oplus B} = A.B + \overline{A}.\overline{B}$$

Case I : If A and B both are LOW,

i.e. $A = B = 0$

$\overline{A} = \overline{B} = 1$

$\therefore \quad Y = A.B + \overline{A}.\overline{B}$

$\therefore \quad Y = 0.0 + \overline{0}.\overline{0}$

$= 0 + 1.1$

$= 1$

Case II : If A = 0 (LOW) and B = 1 (HIGH),

i.e. $A = 0$ and $B = 1$

$\bar{A} = 1$ and $\bar{B} = 0$

$\therefore \quad Y = A.B + \bar{A}.\bar{B}$

$\therefore \quad Y = 0.1 + \bar{0}.\bar{1}$

$= 0 + 0$

$= 0$

Case III : Likewise A = 1 (HIGH) and B = 0 (LOW), Output Y=0.

Case IV : If A = 1 (HIGH) and B = 1 (HIGH), output Y becomes LOW.

i.e. $A = 1$ and $B = 1$

$\bar{A} = 0$ and $\bar{B} = 0$

$\therefore \quad Y = A.B + \bar{A}.\bar{B}$

$\therefore \quad Y = 1.1 + \bar{1}.\bar{1}$

$= 1.1 + 0.0$

$= 1$

QUESTIONS

1. What is positive and negative logic level ? Explain with ideal pulse.
2. What is logic gate ? What are different types of basic gate ?
3. Describe following gates with symbol and truth table :
   (a) OR gate, (b) AND gate, (c) NOT gate, (d) NOR gate, (e) NAND gate.
4. Why NAND and NOR gates are called as universal gate ? Explain.
5. How basic gates can be constructed by using NAND and NOR gate ?
6. Explain EX-OR gates with symbol and truth table.
7. Explain the circuit diagram for basic NOT gate.
8. Draw and explain the NAND gate circuit with transistor. Write a truth table for it.