Module -5

Logic gates

Learning objectives

1. To learn construction, working principle of basic and derived logic gates.
2. To understand the concept of Universal logic gates.
3. To study ANSI/IEEE standard symbols of logic gates.
4. To learn pulsed operations of logic gates.
5. To study simple applications of logic gates.

1. Introduction

Logic gates are the basic components in digital electronics. These gates are used to create digital circuits right from simple to complex logic circuit and even complex integrated circuits. Complex microprocessor or microcontroller ICs are constructed using many logic gates.

Logic gates are the fundamental building blocks of all digital systems. It has one or more inputs and one output with some logical relationship between them. Logic gate accepts binary signals i.e. True or False, ON or OFF, 1 or 0 and have an ability to make decisions. The state of the output is decided by the input states. All logic gates implements some Boolean function which correlates output with input through some logical operation. Logic gates are mainly designed with the electronic switches using diodes and transistors.

There are three basic gates ï AND, OR and NOT. NAND and NOR gates are derived gates are also known as universal logic gates. An XOR gate is inequality detector gate and can be used in comparator, adders, parity generators etc. The symbols of logic gates are of two types ï Distinctive shapes & rectangular shapes.

In this module, the logical operation, characteristics, switch analogy and truth tables of different logic gates are discussed. The original ANSI/IEEE distinctive shape symbols and new ANSI/IEEE standard outline symbols of logic gates are prominently introduced. Logic gates are commercially available in two basic logic families, such as IC 74XX series for TTL (Transistor Transistor Logic) and IC 40XX/45XX for CMOS (Complementary Metal Oxide Semiconductor) series. These all logic gates packaged as Small Scale Integration (SSI) ICs.

2. Logic Gates

Digital systems are constructed using logic gates. The logic gate is the most basic building block of any digital system capable of making decision including computers. Each one of the basic logic gates is a piece of hardware or an electronic circuit that can be used to implement some basic logic expression. It is an electronic circuit with one or many inputs and only one output.

With the help of logic gates it is possible to implement the most elementary logic expressions, also known as Boolean expressions. . The three basic logic gates are the OR gate, the AND gate and the NOT gate. Most logic gates have two inputs and one output.

At any given instance, every input or output terminal is in one of the two logic conditions

- **True or False** which describes Logic value,
- **1 or 0** provides binary value,
- **+5V** (+3.3V for new devices) or 0V provides voltage levels,
- **High or Low** describes the voltage levels and

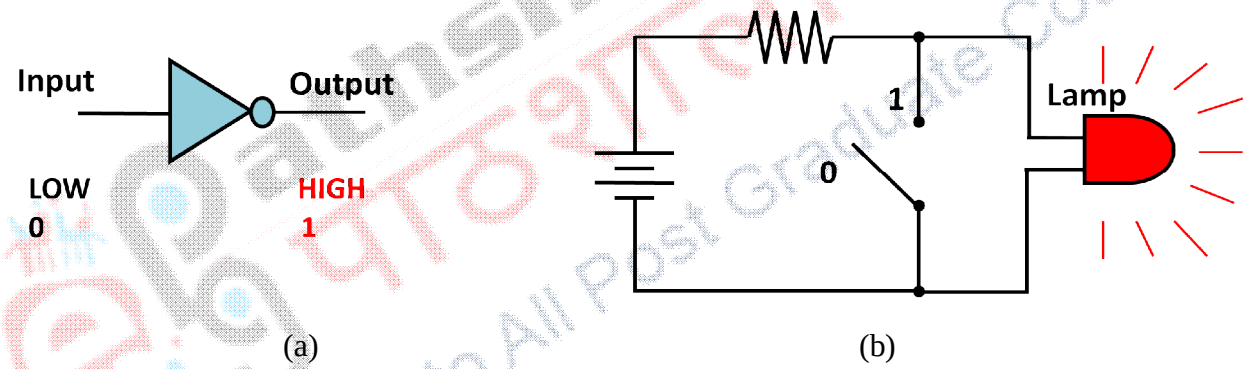
- **ON or OFF** describes the switch position.

The common use of logic gate is to act as switches (although they have no moving parts). Gate opens to pass on logic or close to shut to keep it off. This is why they are known as gates. Gates are classified as primary gates (NOT, AND, OR) and Secondary or derived gates (NAND, NOR ï Universal gates and XOR, XNOR - combinational gates). Let us now discuss these gates one by one.

2.1 NOT gate:

NOT gate has one-input and one-output. It is a logic circuit whose output is always the complement of the input. Fig. 1 indicates the logic symbol and truth table of NOT gate along with simple implementation using switches. The NOT gate is popularly known as **inverter**. It performs logical inversion or complementation.

That is, if A is the input to a NOT circuit, then its output Y is given by **Y = ⍰ or A'**. That is, a LOW input produces a HIGH output and vice versa.



(a) (b)

Input (Switch)	Output (Lamp)
0 (Open)	1 (ON)
1 (Closed)	0 (OFF)

(c)

NOT gate provides logical inversion.

Figure 1: NOT gate (a) Logic Symbol (b) Switch implementation (c) Truth table.

The purpose of inverter is to change one logic level to opposite level. The LOW level at input produces a HIGH level and vice versa. In terms of bits, it changes a 0 to a 1 and a 1 to 0.

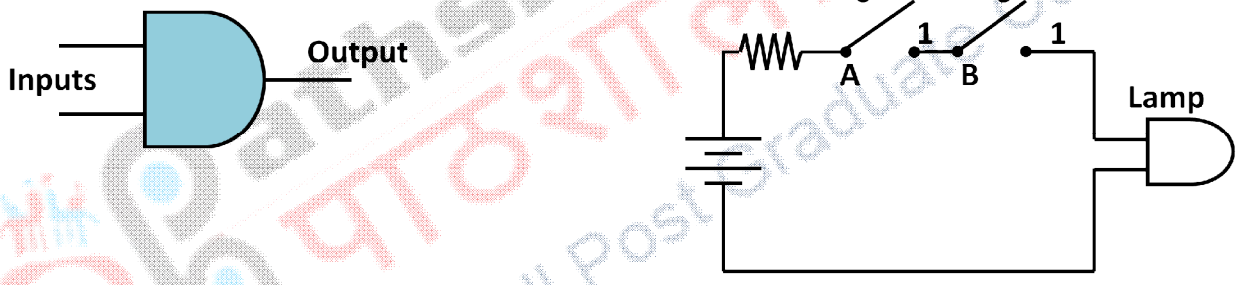
Let us now consider an electronic circuit consisting of Battery, Resistor, Switch and Lamp to implement simple NOT gate or inverter. When switch is open (logic ó0ô) then lamp is ónô (logic ó1ô) and when switch is closed (logic ó1ô) then lamp is óffô (logic ó0ô).

2.2 **AND gate :**

AND gate is a logic circuit having two or more inputs and one output. The AND gate performs logical multiplication i.e. AND function. Fig.2 indicates the logic symbol and truth table of two input AND gate along with simple implementation using switches.

The output of an AND gate is HIGH only when all of its inputs are in the HIGH state. In all other cases, the output is LOW. For AND gate, Y = A.B

The logical operation of AND gate can be expressed with the help of a table which includes all input combinations and corresponding outputs. Such table of input / output relations is known as truth table. In this case, there are 2 inputs to AND gates to provide 2^2 = 4 input combinations in the truth table. The truth table may be extended to any number of inputs. For 3 inputs, there are 8 combinations and 16 for 4 inputs.



(a)

(b)

Input A (Switch)	Input B (Switch)	Output Y (Lamp)
0 (Open)	0 (Open)	0 (OFF)
0 (Open)	1 (Closed)	0 (OFF)
1 (Closed)	0 (Open)	0 (OFF)
1 (Closed)	1 (Closed)	1 (ON)

(c)

The AND gate produces a logic 1, if and only if all the inputs are at logic 1.

Figure 2: AND gate (a) Logic Symbol (b) Switch implementation (c) Truth table.

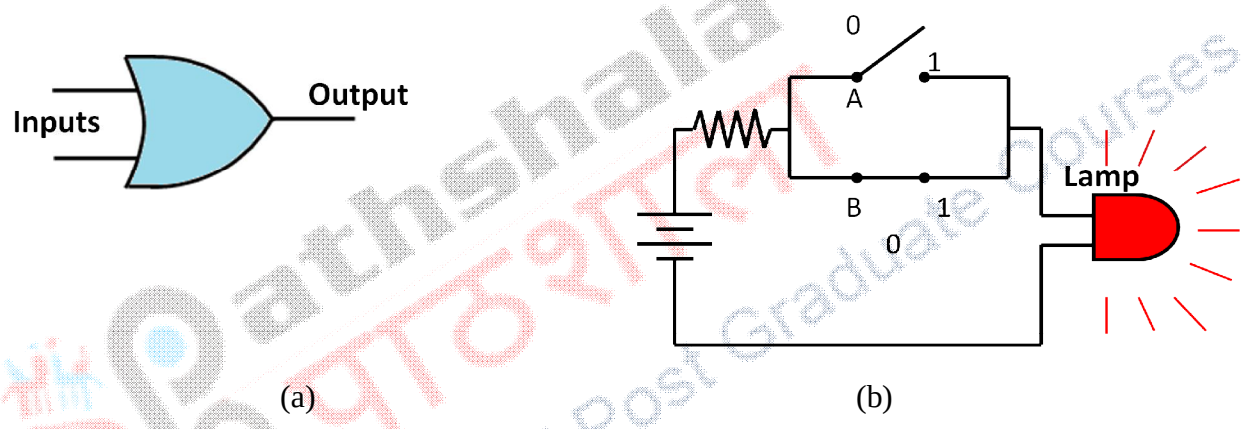
AND gate can be implemented using switches. Let us now consider an electronic circuit consisting of Battery, Resistor, Switches and Lamp to explain the AND function. If both switches are closed (logic ó1ô) then only lamp is ON (logic ó1ô). If any one or both switches are

open (logic‘0’) then lamp will be OFF (logic ‘0’). The AND gate produces a logic 1, if and only if all the inputs are at logic 1.

2.3 OR Gate

An OR gate is a logic circuit with two or more inputs and one output. The OR gate performs logical addition i.e. OR function. Fig. 3 shows the logic symbol and truth table of two input OR gate along with simple implementation using switches.

The output of an OR gate is HIGH only when all of its inputs are in the HIGH state. In all other cases, the output is LOW. For OR gate, **Y = A + B**



Input A (Switch)	Input B (Switch)	Output Y (Lamp)
0 (Open)	0 (Open)	0 (OFF)
0 (Open)	1 (Closed)	1 (ON)
1 (Closed)	0 (Open)	1 (ON)
1 (Closed)	1 (Closed)	1 (ON)

(c)

The OR gate produces a logic 1, if either one or both the inputs are at logic 1.

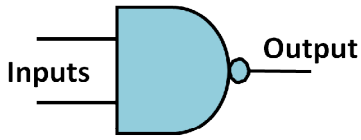
Figure 3: OR gate (a) Logic Symbol (b) Switch implementation (c) Truth table.

OR gate can be implemented using switches. Let us now consider an electronic circuit consisting of Battery, Resistor, Switches and Lamp to explain the OR function. If both switches are open (logic‘0’) then lamp will be OFF (logic ‘0’). If any one or both switches are closed (logic ‘1’) then the lamp is ON (logic ‘1’). The OR gate produces a logic 1, if either one or both the inputs are at logic 1. The truth table may be extended to any number of inputs.

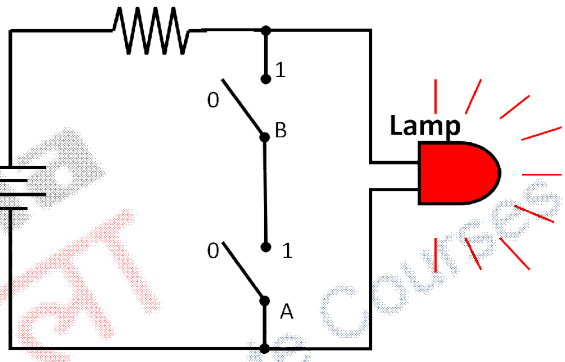
2.4 NAND gate

NAND gate is combination of AND and NOT gates. The NAND gate provides AND functions with inverted output. Fig.4 indicates the logic symbol and truth table of two input NAND gate along with simple implementation using switches.

The output of a NAND gate is a logic ó0ô when all its inputs are a logic ó1ô. For all other input combinations, the output is a logic ó1ô. NAND gate operation is logically expressed as **Y =** [illegible]



(a)



(b)

Input A (Switch)	Input B (Switch)	Output Y (Lamp)
0 (Open)	0 (Open)	1 (ON)
0 (Open)	1 (Closed)	1 (ON)
1 (Closed)	0 (Open)	1 (ON)
1 (Closed)	1 (Closed)	0 (OFF)

(c)

The NAND gate produces a logic 1, if either one or both the inputs are at logic 0 else the output is at logic 1.

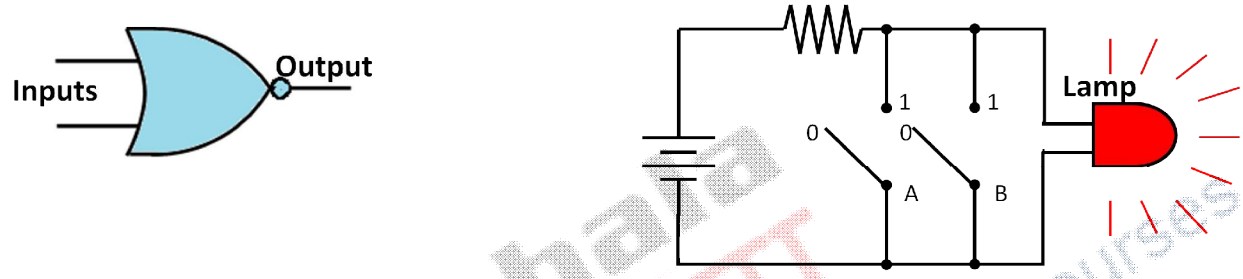
Figure 4: NAND gate (a) Logic Symbol (b) Switch implementation (c) Truth table.

NAND gate can be implemented using switches. Let us now consider an electronic circuit consisting of Battery, Resistor, Switches and Lamp to explain the NAND function. If any one or both the switches are open (logicõ0ô) then lamp will be ON (logic ó1ô). If both the switches are closed (logic ó1ô) then the lamp is ON (logic ó1ô). The NAND gate produces a logic 1, if either one of both the inputs are at logic 0 else the output is at logic 1. The truth table may be extended to any number of inputs.

2. 5 NOR gate

NOR gate is combination of OR and NOT gates. The NOR gate provides OR function with inverted output. Fig. 5 shows the logic symbol and truth table of two input NOR gate along with simple implementation using switches.

The output of a NOR gate is a logic ó1ô when all its inputs are logic ó0ô. For all other input combinations, the output is a logic ó0ô. NOR gate operation is logically expressed as **Y=**



(a) (b)

Input A (Switch)	Input B (Switch)	Output Y (Lamp)
0 (Open)	0 (Open)	1 (ON)
0 (Open)	1 (Closed)	0 (OFF)
1 (Closed)	0 (Open)	0 (OFF)
1 (Closed)	1 (Closed)	0 (OFF)

(c)

The NOR gate produces a logic 0, if either one or both the inputs are at logic 1 else the output is at logic 0.

Figure 5: NOR gate (a) Logic Symbol (b) Switch implementation (c) Truth table.

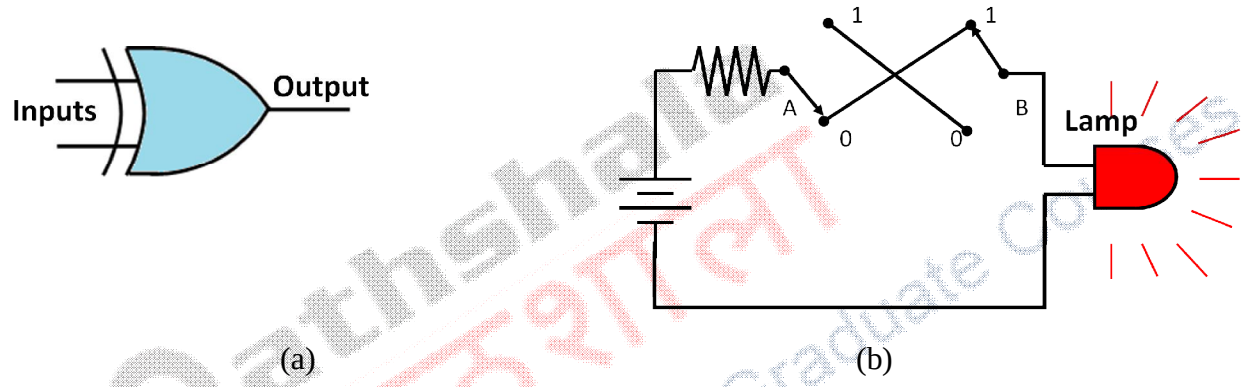
NOR gate can be implemented using switches. Let us now consider an electronic circuit consisting of Battery, Resistor, Switches and Lamp to explain the NOR function. If any one or both the switches are Closed (logicô1ô) then lamp will be OFF (logic ó0ô). If both the switches are open (logic ó0ô) then the lamp is ON (logic ó1ô). The NOR produces a 0, if either one or both the inputs are at logic 1 else the output is at logic 0. The truth table may be extended to any number of inputs.

2.6 XOR (Exclusive OR) gate

Exclusive OR gate is basically designed to exclude the condition of standard OR gate so as to generate real binary addition. An XOR gate is a two inputs and one output logic circuit. Fig.6 indicates the logic symbol and truth table of two input XOR gate along with simple implementation using switches.

The output of an XOR gate is at logic ó1ô when the inputs are dissimilar and at logic ó0ô when the inputs are similar. The logic equation for two input XOR gate is given by

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



(a) (b)

Input A	Input B	Output Y
0	0	0
0	1	1
1	0	1
1	1	0

(c)

The XOR gate produces a logic 1, only when the two inputs are at opposite levels .

Figure 6: XOR gate (a) Logic Symbol (b) Switch implementation (c) Truth table.

The output of XOR is in logic ó1ô, when one and only one of two inputs are at logic ó1ô. Similarly, the output of XOR is at logic ó0ô, when both the inputs are at logic ó0ô or both inputs are at logic ó1ô. As XOR gate generates logic ó1ô only when inputs are not equal, hence this gate is also known as **inequality detector** gate.

When more than two inputs are to be XORed then multiple two inputs XOR gates may be used. The output of a multiple-input XOR logic function is at logic ó1ô when the number of 1s in the input sequence is odd and at logic ó0ô when the number of 1s in the input sequence is even,

including zero. That is, an all 0s input sequence also produces a logic ó0ô at the output. Three or more inputs XOR do not exist.

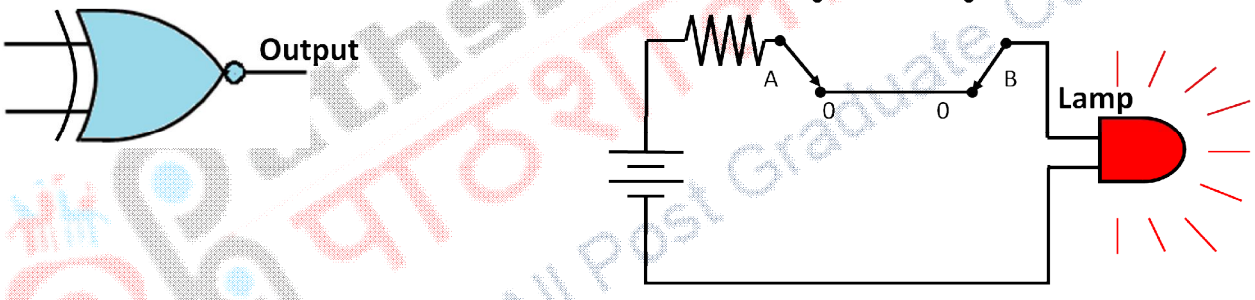
2.7 XNOR gate

XNOR is obtained by the combination of NOT and XOR gates. XNOR gate is a two inputs and one output XOR gate with active low output. Fig.7 indicates the logic symbol and truth table of two input XNOR gate along with simple implementation using switches.

The output of an XNOR gate is at logic ó1ô when the inputs are similar and at logic ó0ô when the inputs are dissimilar. The logic equation for two input XOR gate is given by

$$\mathbf{Y} = \overline{A\overline{B} + \overline{A}B}$$

The output of XNOR can be obtained by complementing the output column for XOR gate for corresponding input conditions.



(a) (b)

Input A	Input B	Output Y
0	0	1
0	1	0
1	0	0
1	1	1

The XNOR gate produces a logic 1, only when both the inputs are at the SAME logic levels.s

(c)

Figure 7: XNOR gate (a) Logic Symbol (b) Switch implementation (c) Truth table.

XNOR gate can be implemented using switches. Let us now consider an electronic circuit consisting of Battery, Resistor, Switches and Lamp to explain the XNOR function.

The output of XNOR is in logic ó1ô when both the inputs are at logic ó0ô or both inputs are at logic ó1ô. Similarly, the output of XOR is at logic ó0ô when one and only one of two inputs are at logic ó1ô. As XNOR gate generates logic ó1ô only when both the inputs are equal, hence this gate is also known as **equality detector** gate.

3. IEEE/ANSI Standard Symbols

American National Standards Institute (ANSI), the Institute of Electronics and Electronic Engineers (IEE) has developed a standard set of logic IEEE symbols for logic symbols. There are two sets of symbols commonly used for logic gates. Both are defined in ANSI/IEEE standard 91-1984 and its supplement in 1991. The distinctive shape symbols are based on traditional schematics. For more complex devices such as adders, subtractors, registers, counters , logic symbols used so far do not carry any useful information. The rectangular outline symbols were introduced as industry standard symbols. The following figure 8 shows IEEE/ANSI symbols for logic gates.

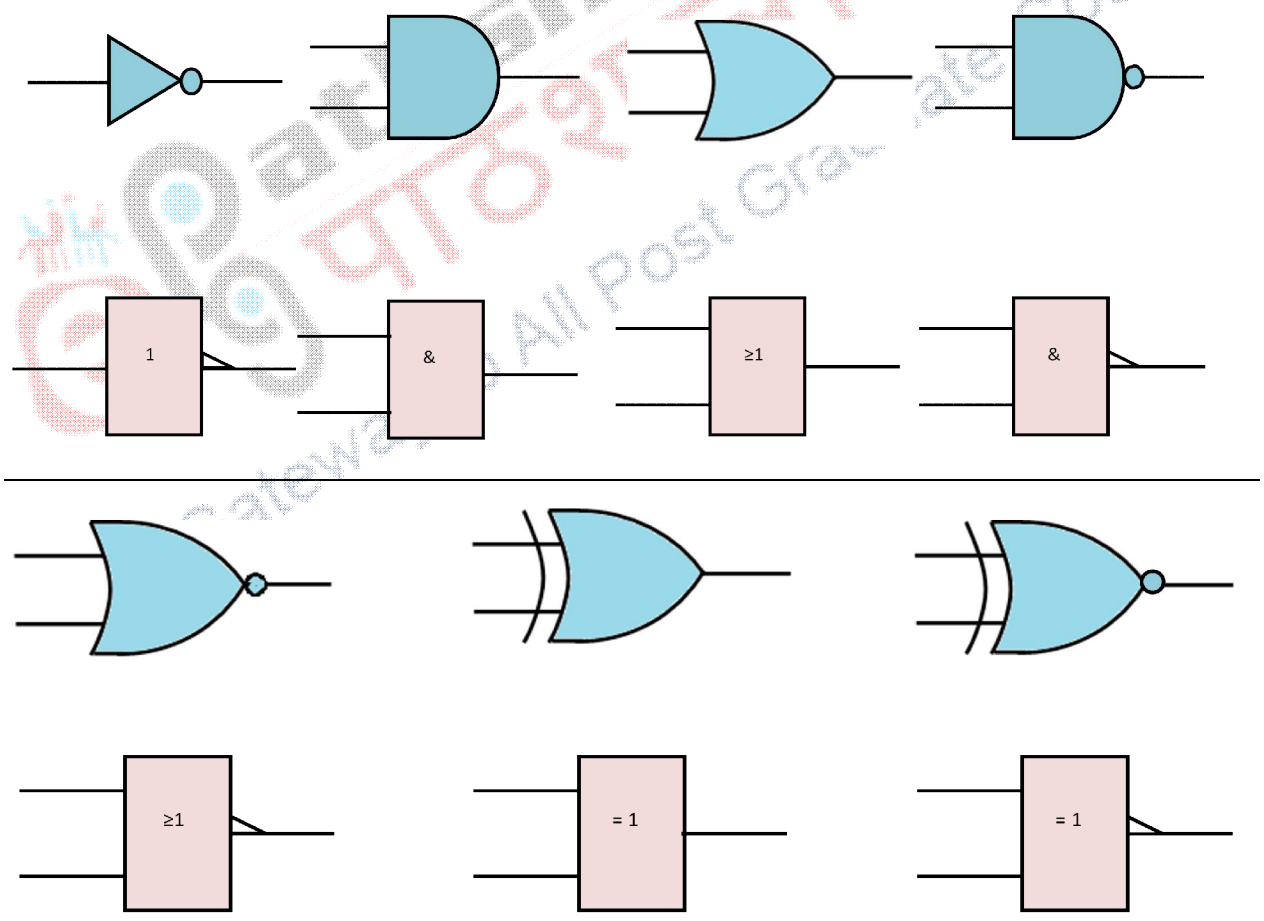


Figure 8: ANSI/IEEE symbols for logic gates

In these symbols ñbubbleò indicates active low and ñwedgeò is a negative polarity indicator in distinctive shape symbols and rectangular outline symbols respectively.

4. Universal Logic Gates

Any Boolean / logic expression can be realized using the AND, OR, and NOT gates. From these three primary gates, two derived gates NAND and NOR are usually realized. It is possible to construct basic gates namely NOT, AND, OR using combination of NAND gates or a combination of NOR gates . For this reason NAND and NOR gates are called as universal logic gates.

Let us now understand the importance of NAND and NOR logic gates. The NAND and NOR gates are referred to as Universal logic gates as each of these gates can be used to construct basic logic gates AND, OR & NOT gates.

4.1 NAND gate as universal logic gate

Let us consider now NAND gate as a universal logic gate. Figure 9 shows the implementation of basic logic gates using only NAND gates.

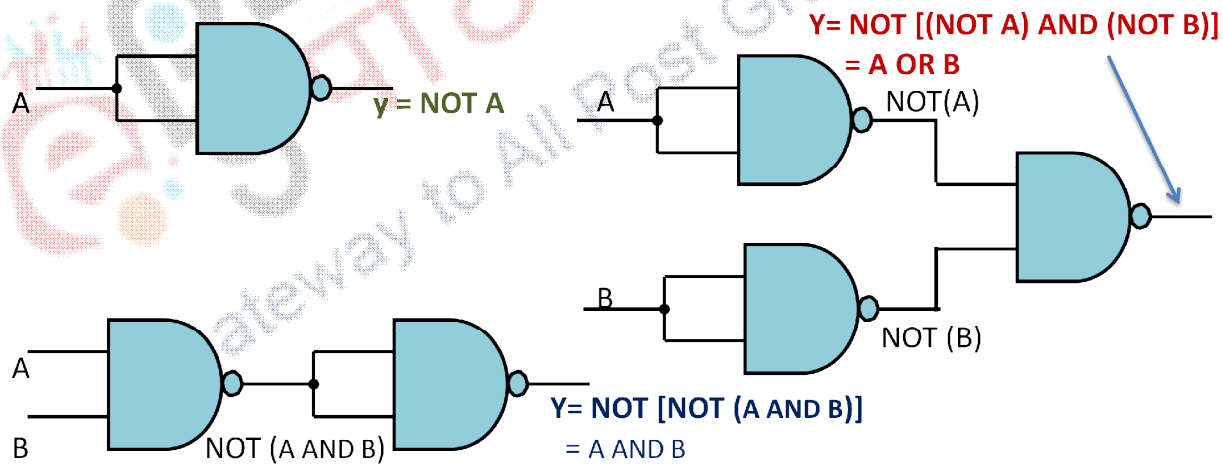


Figure 9: Construction of NOT, AND & OR gates using NAND gates

NAND gate can be used to as NOT gate by connecting all its inputs together and applying input to the common terminal. The output of NAND gate generates output as NOT A.

For constructing AND gate, two NAND gates are required. The first NAND gate provides complement of A AND B., whereas the second NAND gate acts like an inverter. The double complement will cancel each other to provide AND gate.

Three NAND gates are required to construct the OR gate. First level two NAND gates will act like an inverter. These two inverters will complement the inputs. The third NAND gate will perform ANDing of the complemented inputs first and then inverts the result. This provides final output as logical OR functions.

4.2 NOR Gate as universal logic gate

Let us consider now NOR gate as a universal logic gate. Figure 10 shows the implementation of basic logic gates using only NOR gates.

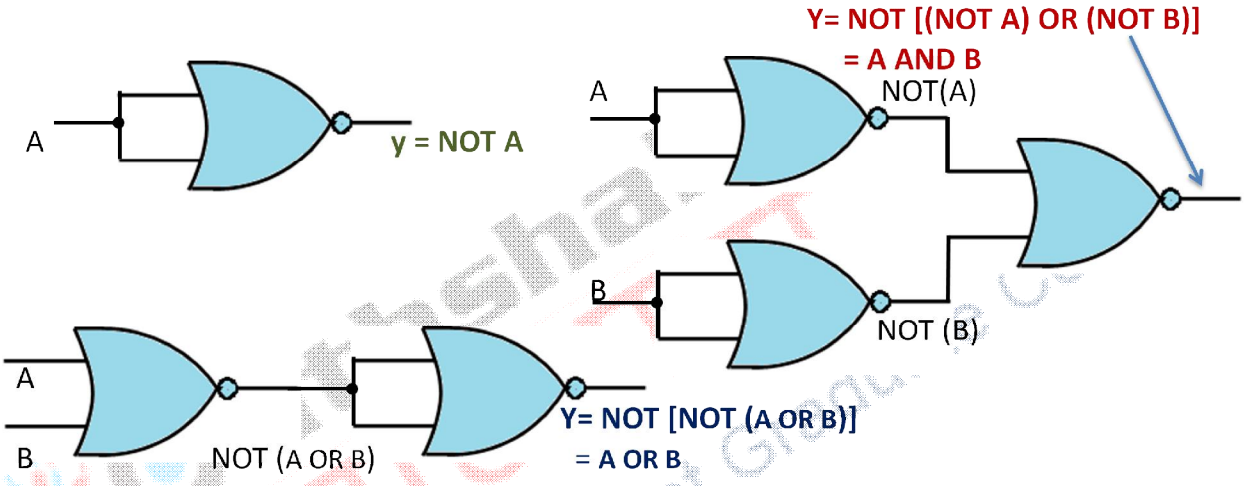


Figure 10: Construction of NOT, AND & OR gates using NOR gates

A NOR gate can be used to as NOT gate by connecting all its inputs together and applying input to the common terminal. The output of NOR gate generates output as NOT A i.e. complement of A.

The construction of OR gate requires two NOR gates. The first NOR gate provides complement of A OR B. Whereas the second NOR gate acts like an inverter. The double complement will cancel each other to provide action of OR gate.

Three NOR gates are required to construct the AND gate.

In the first level two NOR gates will act like an inverter. These two inverters will complement the inputs. The third NOR gate will perform ORing of the complemented inputs first and then inverts the result. This provides final output as logical AND functions.

5. Special logic gates

5.1 Gates with Open collector

In case of TTL logic family, some NAND require an external resistor, called the pull-up resistor. Pull up resistor us connected between the output and the DC power supply to get desired logic function. In case of MOS family open drain gates are used. These logic gates have capability of providing an ANDing operation when outputs without using an AND gate. This connection is is popularly known as wire-AND system. Figure 5.11 indicates Wired AND gate using open collector NAND gates. The output of wire-AND system is

Y = ABâ CDâEFô

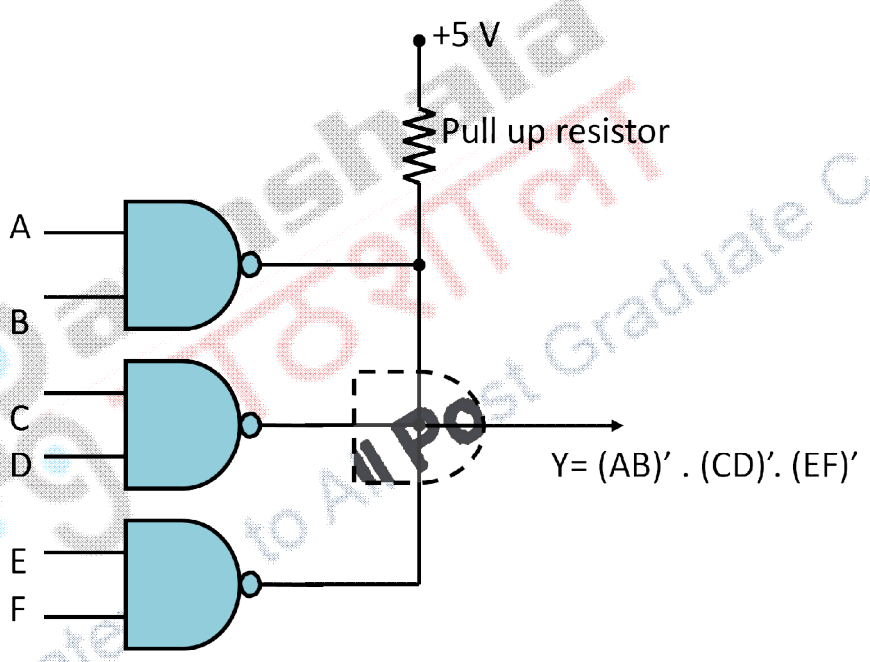


Figure 5.11: Wire-AND system using open collector NAND devices

The open collector outputs of several NAND gates can be connected together to form then Wired- AND function. In this example, three open collector NAND gate outputs are wired together as shown in this figure. One single pull up resistor is sufficient for this interconnection. Figure 5.12 shows similar circuit configuration for Wire-ANDing using open collector NOT gates.

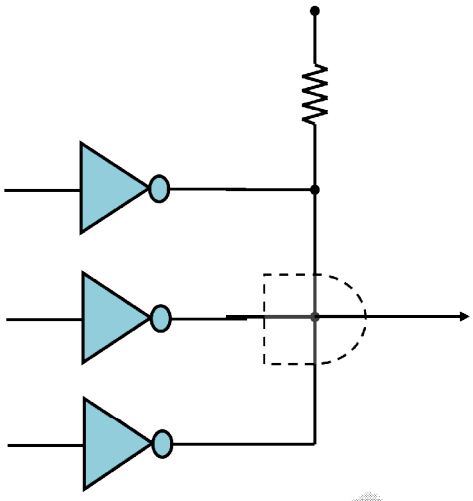


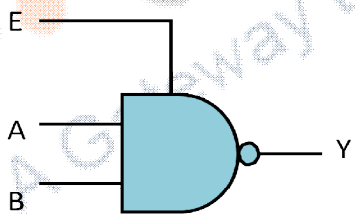
Figure 5.12: Wire-AND system using open collector NOT devices

The output Y of wire- AND ing of all the inputs is

Y= (A)ô. (B)ô (C)ô

5.2Tristate Gates

In digital electronics, tri-state (three state) logic allows three possible output states namely the logic ó1ô state, the logic ó0ô state and a high-impedance state. This logic gate allows to share same output line or lines. The high-impedance state is controlled by an external ENABLE input. The ENABLE input decides whether the tri-state gate is active (0 or 1) or in the high-impedance state.

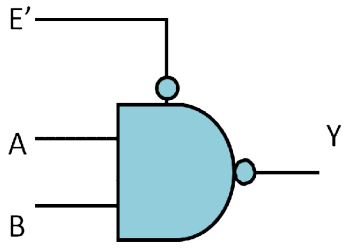


E	A	B	Y
0	0	0	Z
0	0	1	Z
0	1	0	Z
0	1	1	Z
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Figure 5.13: Tri-state gate with active HIGH enable

Fig. 5.13 show when Enable input is HIGH, the NAND function provides 0 or 1 depending upon the input combination. If enable is LOW, the NAND gate output will provide high-impedance state independent of the input combinations.

Let us now consider NAND gate with the active LOW enable input.



E	A	B	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	Z
1	0	1	Z
1	1	0	Z
1	1	1	Z

Figure 5.14: Tri-state gate with active LOW enable

The high-impedance state is controlled by an external ENABLE input. The ENABLE input decides whether the gate is active or in the high-impedance state. When active, it can be ó0ô or ó1ô depending upon input conditions. In this gate when Enable input is LOW, the NAND function provides 0 or 1 depending upon the input combination. If enable is HIGH, the NAND gate output will provide high-impedance state independent of the input combinations.

Let us consider the interconnections of tristate inverters together to form a common bus as shown in fig.5.15.

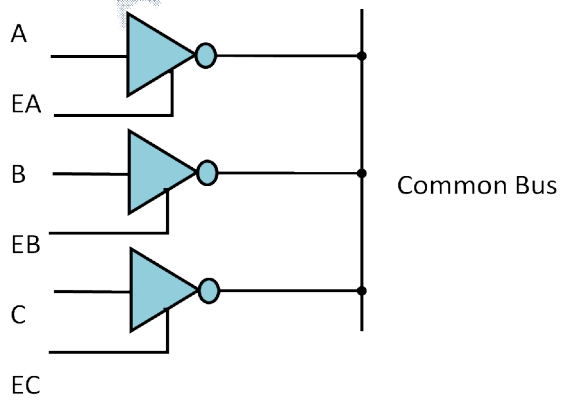


Figure 5.15: Tristate inverter

All not gates are having single input and single output with enable input. The outputs are connected together to form a common bus. Each NOT gate has a separate active HIGH Enable input. This arrangement allows only one of them to be enabled at a time. HIGH ENABLE INPUT activates the respective NOT gate with a condition that only one of them is enabled at a time.

5.3 Schmitt Gates

A standard NOT gate or NAND gate is usually made up from transistor as a switching device. Transistor cannot switch from one state to another instantly, there is some delay in switching between the states. Transistor is basically a current amplifier and can also operate in linear region. Any small changes at its input level may cause a change in its output level and can even switch ON or OFF several times if there is any noise present in the system. One way to overcome this problem is to use Schmitt inverter (NOT or NAND gates).

Schmitt gate is basically a comparator with hysteresis. It is designed to operate when its input signal goes above an ñUpper Threshold Point (UTP)ò, the output changes and goes to LOW level and remain in the same state until the input signal falls below the ñLower Threshold Point (LTP)ò. Once again the output changes and goes HIGH. In short, a Schmitt gate has built in Hysteresis to overcome the problem of noise.

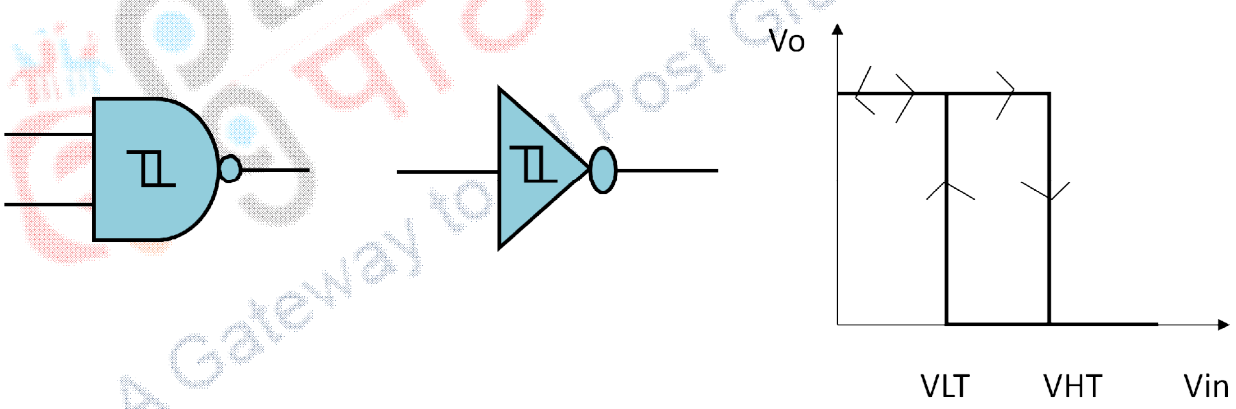


Figure 5.16: Schmitt Gates

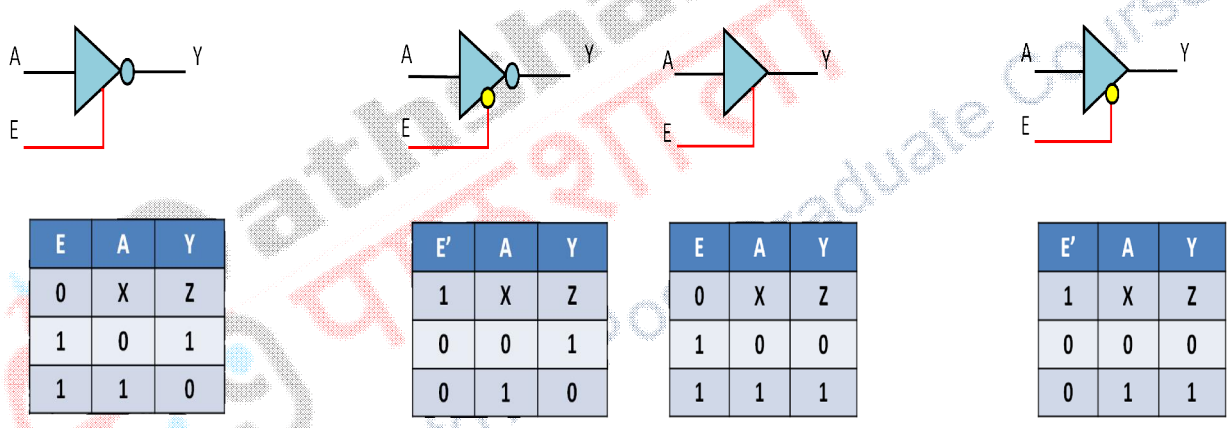
Fig. 5.16 shows NAND and NOT gate with built-in hysteresis. These gates are referred to as Schmitt gates. This gate interprets slowly changing input voltages according to two threshold voltages. Lower & upper threshold voltages are normally used for LOW to HIGH and HIGH to LOW transitions respectively. Schmitt gates are distinguished from conventional gates by the small ôhysteresisô symbol of the B ī H loop for a ferromagnetic material. Hysteresis increases noise immunity of the digital system.

This clean ON/OFF switching makes the Schmitt gate ideal for switching any slow-rising or slow falling input signal. An important application of Schmitt gate is in oscillators i.e. sine to square wave converter or a clock.

5.4 Buffers

A buffer has only one input and one output. Normal logic gates have a limited load-driving capability. A logic gate can be treated as signal amplifiers, regardless of what logic function they may perform. For this purpose, a special logic gate called a buffer is designed. A buffer has a larger load-driving capability than a normal logic gate. It could be an inverting or non inverting buffer with a single input, a single output and an Enable input. Fig. 5.17 shows the symbols and functional tables of inverting and noninverting buffers of the tri-state type.

Buffer is also popularly known as ñDriverò. Buffers are usually tri-state devices preferred for designing bus-oriented system..



E	A	Y
0	X	Z
1	0	1
1	1	0

E'	A	Y
1	X	Z
0	0	1
0	1	0

E	A	Y
0	X	Z
1	0	0
1	1	1

E'	A	Y
1	X	Z
0	0	0
0	1	1

Figure 5.17: Inverting and noninverting Buffers

A tristate buffer is a useful device that allows us to control when current passes through the device and when it does not. A tri-state buffer has two inputs: a data input A and a enable input E. When the enable input is active i.e. At logic 1 then the output is same as input. If enable input is 0 then the output is in tri-state. In many applications we need tri-state buffers with active LOW enable input.

5.5 Transceivers

It is kind of buffer gate capable of both transmit and receive. A transceiver is a combination of transmitter and receiver in a single IC package. It is capable of driving bidirectional systems. A *transceiver* is a bidirectional buffer capable of driving bidirectional system with additional direction control/ENABLE inputs. It allows flow of data in both directions, depending upon the

status of the control inputs. Transceivers can be popularly used to construct bus-oriented systems. Fig. 5.18 indicates the circuit symbols of inverting and non-inverting transceivers.

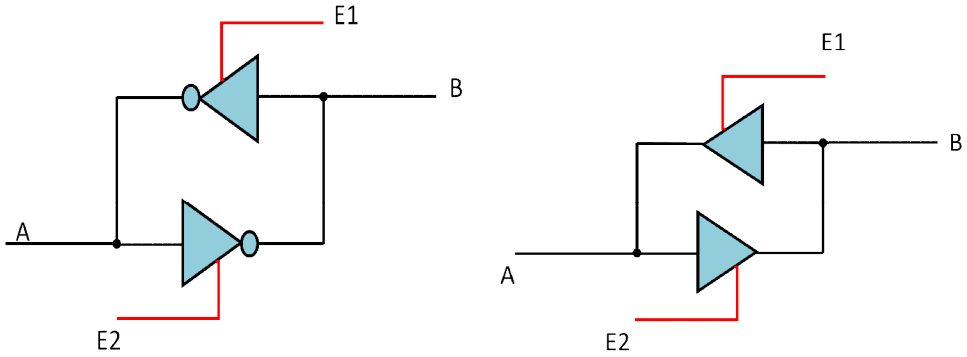


Figure 5.18 Inverting and non-inverting transceiver

In case of inverting transceiver, when active HIGH input is applied at Enable input E1, data is transferred from A to B in complemented form and when Enable input E2=1, then the data is transferred from B to A in complemented form. Where as in non-inverting transceiver, data is transmitted or received in un-complemented form.

5.6 AOI gate –

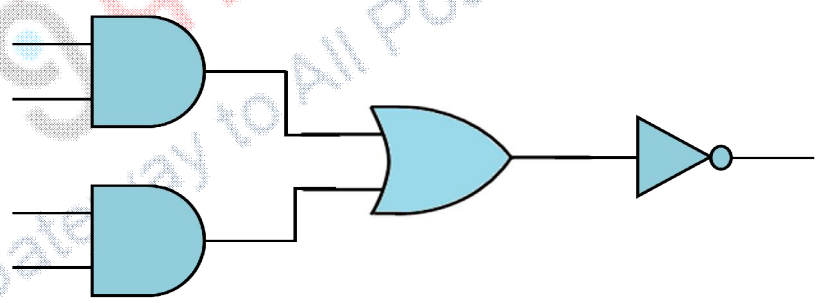


Fig. 5.19 AOI gate

Fig. 5.19 indicates the special arrangement of three basic logic gates, which is popularly known as AOI (AND-OR-Invert) gate. It is used in many applications of combinational systems.

6. Pulsed operation of logic gates

In many applications, the input to a gate are not stationary levels but changing dynamically between two voltage levels frequently. These types of changing voltage or logic levels give rise to Pulse waveform. Let us now study the response of different logic gates with pulse input waveform.

Pulsed operation of NOT gate

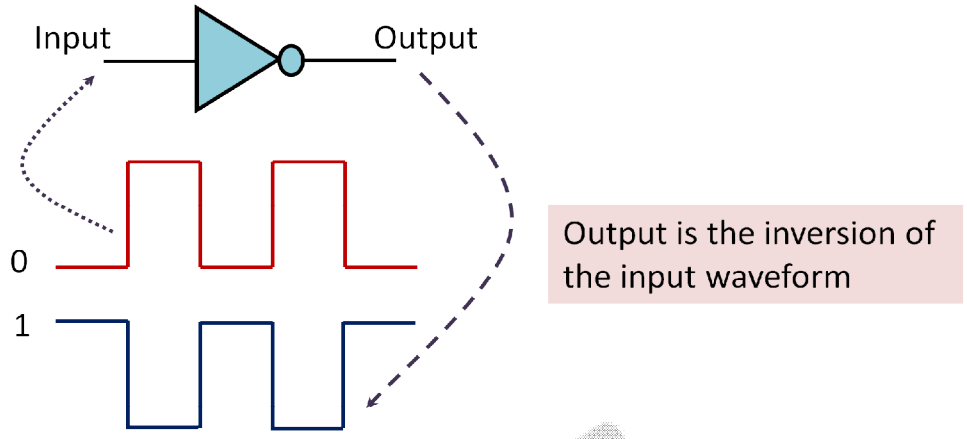


Figure 5.20: Response of NOT gate to Pulse input

Let us now study the response of NOT gate to the pulse input waveform. It is important to note that the all logic gates obey the truth table irrespective of the static or dynamic inputs. For a NOT gate if the input is at logic 0 then output will be in logic 1 and vice a versa. The output is inverted as compared to input.

Pulsed operation of AND gate

In order to study the pulse response, it is important to look at the input of the gate with respect to each other to determine the output level at any time. Let us consider two AND gates with one of the input is fixed and tied to either 0 or 1 and pulse waveform is applied to the other input as shown in fig. 5.21. When one of the inputs is 1, the pulse appears as it at output and output is 0 when one of the inputs is 0.

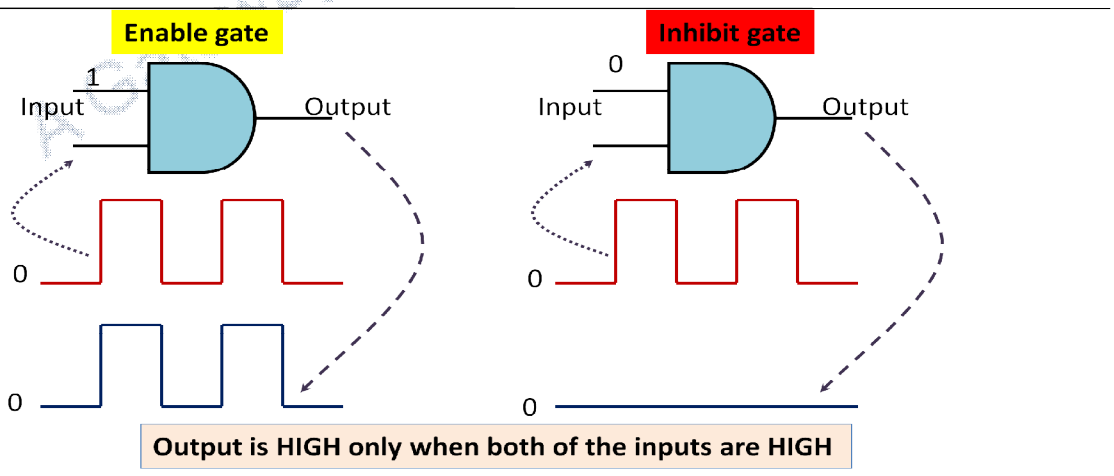


Figure 5.21: Response of AND gate to Pulse input

Pulsed operation of OR gate

Let us consider two OR gates with one of the input is fixed and tied to either 0 or 1 and pulse waveform is applied to the other input as shown in fig. 5.22. When one of the inputs is 0, the pulse appears as it at output and output is 1 when one of the inputs is 1.

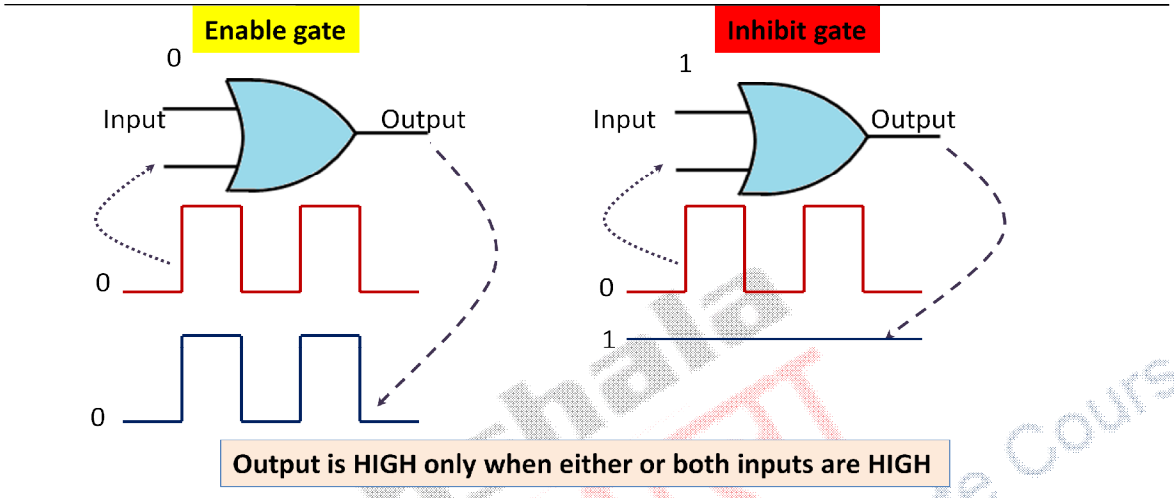


Figure 5.22: Response of OR gate to Pulse input

Pulsed operation of NAND gate

Let us consider two NAND gates with one of the input is fixed and tied to either 0 or 1 and pulse waveform is applied to the other input as shown in figure 5.23. When one of the inputs is 1, the pulse appears inverted at output and output is 1 when one of the inputs is 0.

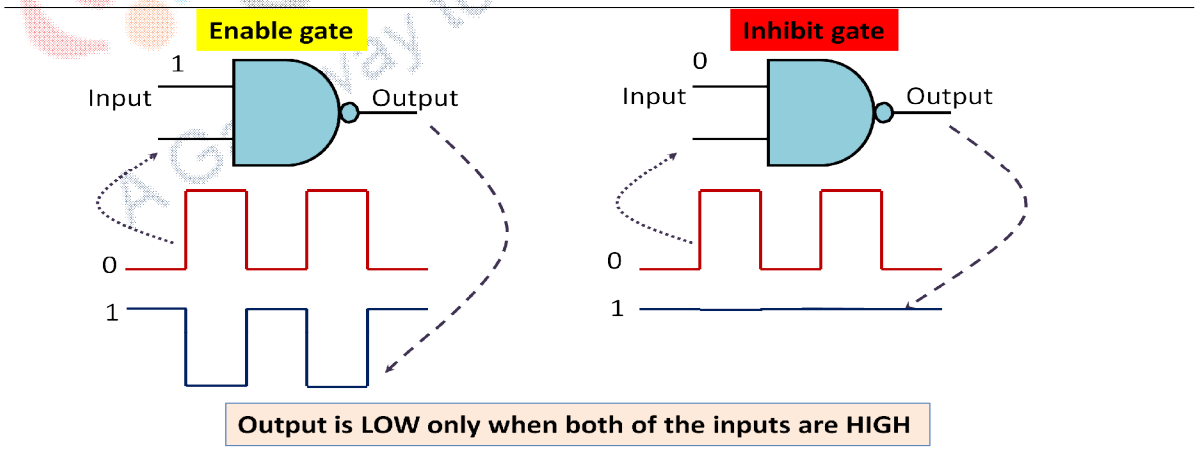


Figure 5.23: Response of NAND gate to Pulse input

Pulsed operation of NOR gate

Let us consider two NOR gates with one of the input is fixed and tied to either 0 or 1 and pulse waveform is applied to the other input as shown in fig. 5.24. When one of the inputs is 0, the pulse inverted at output and output is 0 when one of the inputs is 1.

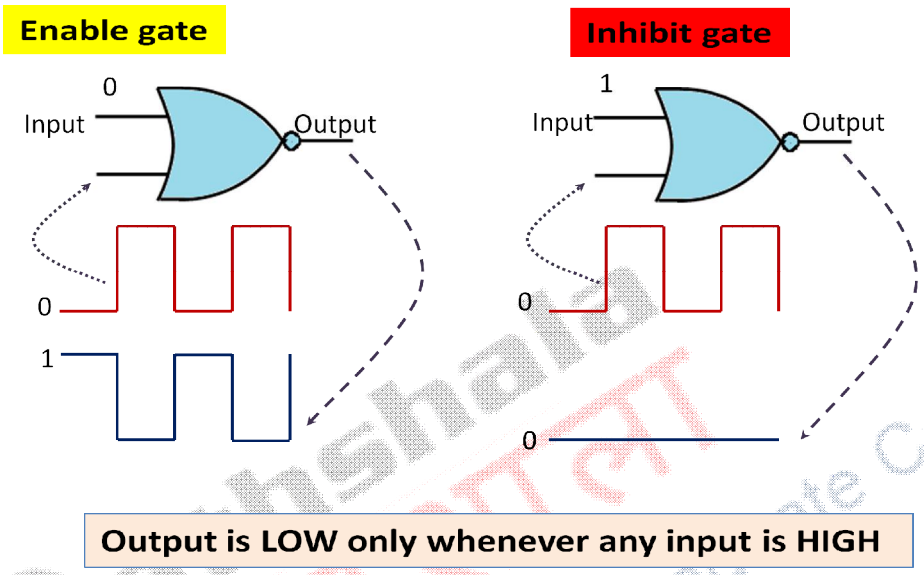


Figure 5.24: Response of NOR gate to Pulse input

5.7 Applications of logic gates

The applications of logic gates are countless. Logic gates are used in many household and industrial applications. Every digital gadget including PC, mobile phones, tablets, calculators, digital watches etc. Let us now list very simple applications of Logic gates. One must choose the right logic gate for the application.

1. Safety lock
2. Freezer Warning alarm
3. Enable gate for counter
4. Industrial plant control
5. Burglar Alarm etc.

5.8 Summary

Logic gate is an electronic circuit with one or many inputs and only one output. Various inputs and output can take only two values either ólowô (logic ó0ô) or óhighô (logic ó1ô). ***Truth table*** for a logic gate lists all combinations of input values and output values produced by each one.

- *Logic gates are the basic building blocks of all digital systems and gadgets.*
- *NOT produces complement of the input.*
- *AND gate output is HIGH(1) when all the inputs are at HIGH (1)*
- *OR gate output is HIGH(1) when any of the inputs is HIGH(1)*
- *NAND gate output is LOW (0) only when all the inputs are HIGH (1).*
- *NOR gate output is LOW (0) when any of the inputs is HIGH (1)*
- *XOR gate output is HIGH when the inputs are not the same.*
- *XNOR gate output is LOW when the inputs are not the same.*
- *AND and NOR gates are called as Universal logic gates.*
- *ANSI /IEEE support traditional distinctive shapes and rectangular outline symbols for logic gates.*
- *Logic gates can be used in two basic applications: Enable gate and Inhibit gate.*
- *Logic gates are used in many household and industrial applications.*