UNIT 23 DIGITAL ELECTRONICS

DIGITAL ELECTRONICS AND SIGNAL LEVELS:

Digital electronics is a field of electronics that focuses on processing and manipulating digital signals, represented by binary digits (0s and 1s). Unlike analog electronics, which deal with continuous signals, digital systems use discrete signals for precise control, computation, and data storage. This forms the backbone of modern computing, communication, and information processing technologies.

DIGITAL SIGNAL LEVELS:

In digital electronics, signal levels are represented in various ways. Here are common digital signal levels:

1. Low Level (0):

► This represents the binary digit 0.
► In terms of voltage, it is associated with a lower voltage level.
► Often referred to as "low" or "logic 0."

2. High Level (1):

► This represents the binary digit 1.
► In terms of voltage, it is associated with a higher voltage level.
► Often referred to as "high" or "logic 1."

3. Threshold Level:

► The threshold level is the voltage level that separates low and high states.
► Signals below this threshold are interpreted as 0, and
Signals above it are interpreted as 1.
► The threshold level helps define the noise margin and ensure reliable signal interpretation.

4. Logic Levels:

► Depending on the technology and the system, different logic levels might be used.

5. Swing or Voltage Range:

► The difference in voltage levels between low and high states is often referred to as the swing or voltage range. For example, a voltage ranging from -5V to +5V will have a swing or voltage range of 10V.
► A larger voltage swing can enhance noise immunity and signal reliability.

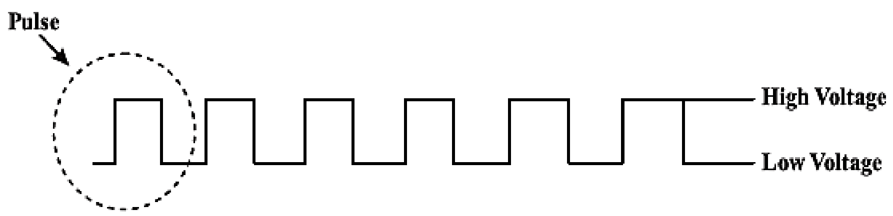
UNDERSTANDING DIGITAL SIGNAL LEVELS

In digital systems, signals operate in two states:

High State ('1'): Represents an active or "on" condition, typically associated with a closed circuit or a voltage level near the maximum specified value.

Low State ('0'): Represents an inactive or "off" condition, linked to an open circuit or a voltage level close to the minimum specified value.

Digital waveforms alternate between these states, forming a sequence of pulses known as a **pulse train**. These waveforms define the operation of digital circuits, enabling data transmission and processing.



UNIT 23 DIGITAL ELECTRONICS

LOGIC GATES:

Logic gates are small electronic switches that perform fundamental logical operations, enabling decision-making in computers and digital devices. They are the building blocks of digital circuits and play a vital role in information processing.

TYPES OF LOGIC GATES:

Logic gates typically have one output and one or more inputs. They are classified into the following categories:

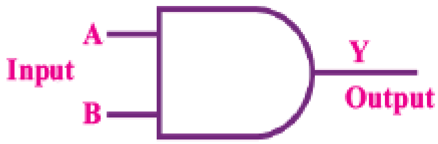
Basic Gates: AND, OR, and NOT gates form the foundation of digital logic.

Universal Gates: NAND and NOR gates can be used to construct any other logic gate, making them essential in circuit design.

Special Gates: XOR (Exclusive OR) and XNOR (Exclusive NOR) gates are used in specific applications, such as arithmetic operations and error detection.

AND Gate:

An AND gate produces a high (1) output only when all its inputs are high (1). In other words, the output of an AND gate is 1 if and only if all inputs are 1; otherwise, the output remains low (0).



The Boolean expression for an AND gate is given as follows:

1. For a two-input AND gate:

$$Y = A \cdot B$$

Y = A . B		
A	**B**	**Y**
0	0	0
0	1	0
1	0	0
1	1	1

2. For a three-input AND gate:

$$Y = A \cdot B \cdot C$$

Y = A · B · C			
A	**B**	**C**	**Y**
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

3. For a four-input AND gate:

$$Y = A \cdot B \cdot C \cdot D$$

Four Input $Y = A.B.C.D$				
A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

OR gate:

An OR gate is a fundamental digital logic gate that outputs a high signal (1) if any of its inputs are high (1). The '+' symbol represents the Boolean expression for an OR gate. For example, the output Y is given by Y = A + B for two inputs A and B.

1. **Two-Input OR Gate:**

Boolean Expression: **Y = A + B**

$Y = A + B$		
A	**B**	**Y**
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**1**

2. **Three-Input OR Gate:**

Boolean Expression: Y = A + B + C

$Y = A + B + C$			
A	**B**	**C**	**Y**
0	**0**	**0**	**0**
0	**0**	**1**	**1**
0	**1**	**0**	**1**
0	**1**	**1**	**1**
1	**0**	**0**	**1**
1	**0**	**1**	**1**
1	**1**	**0**	**1**
1	**1**	**1**	**1**

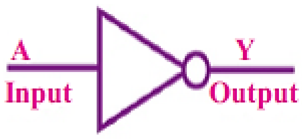
3. Four-Input OR Gate:

Boolean Expression: Y = A + B + C + D

Four Input $Y = A+B+C+D$				
A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

NOT gate:

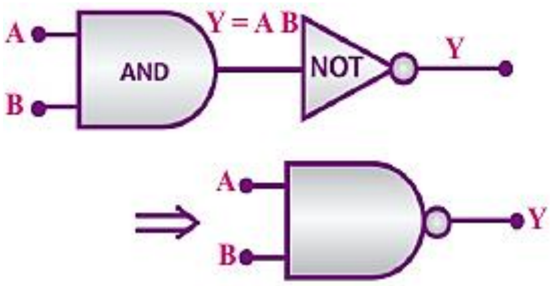
It is used to perform the inversion operation in digital circuits, for example, the output of a NOT gate attains state 1 if and only if the input does not attain state 1.



$Y = \bar{A}$	
0	**1**
1	**0**

NAND gate:

A NAND gate is a combination of an AND gate followed by a NOT gate, which produces the complement of the AND gate's output. The Boolean expressions and truth table for the NAND gate are given as follows:



1. **Two-input NAND gate: The Boolean** expression is $Y = \overline{A \cdot B}$

$Y = \overline{A \cdot B}$			
A	***B***	**A·B**	***Y***
0	**0**	**0**	**1**
0	**1**	**0**	**1**
1	**0**	**0**	**1**
1	**1**	**1**	**0**

2. **Three-input NAND gate: The Boolean expression is** $Y = \overline{A \cdot B \cdot C}$

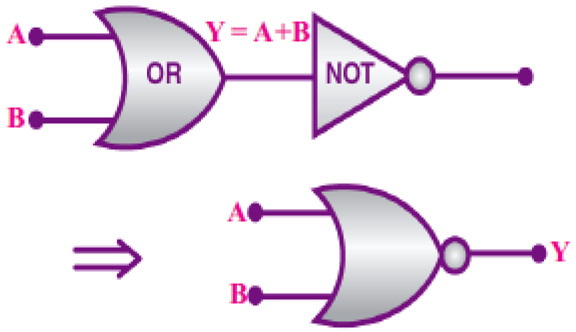
$Y = \overline{A \cdot B \cdot C}$				
A	**B**	**C**	$A \cdot B \cdot C$	**Y**
0	**0**	**0**	**0**	**1**
0	**0**	**1**	**0**	**1**
0	**1**	**0**	**0**	**1**
0	**1**	**1**	**0**	**1**
1	**0**	**0**	**0**	**1**
1	**0**	**1**	**0**	**1**
1	**1**	**0**	**0**	**1**
1	**1**	**1**	**1**	**0**

3. Four-input NAND gate: The Boolean expression is $Y = \overline{A \cdot B \cdot C \cdot D}$

Four Input $Y = \overline{A.B.C.D}$				
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

NOR gate:

A NOR gate is a combination of an OR gate followed by a NOT gate, producing the complement of the OR gate's output. The Boolean expressions and truth table (No: 23.5) for the NOR gate are given as follows:



1. Two-input NOR gate: The Boolean expression is $Y = \overline{A + B}$

$Y = \overline{A + B}$			
A	**B**	**A+B**	$Y = \overline{A + B}$
0	**0**	**0**	**1**
0	**1**	**1**	**0**
1	**0**	**1**	**0**
1	**1**	**1**	**0**

2. Three-input NOR gate: The Boolean expression is $Y = \overline{A + B + C}$

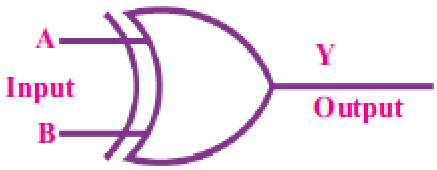
$Y = \overline{A + B + C}$				
A	**B**	**C**	**A+B+C**	$Y = \overline{A + B + C}$
0	**0**	**0**	**0**	**1**
0	**0**	**1**	**1**	**0**
0	**1**	**0**	**1**	**0**
0	**1**	**1**	**1**	**0**
1	**0**	**0**	**1**	**0**
1	**0**	**1**	**1**	**0**
1	**1**	**0**	**1**	**0**
1	**1**	**1**	**1**	**0**

3. Four-input NOR gate: The Boolean expression is $Y = \overline{A + B + C + D}$

Four Input $Y = \overline{A + B + C + D}$				
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

XOR (Exclusive OR) gate:

An XOR (Exclusive OR) gate is a digital logic gate that outputs true (1) only when the number of true inputs is odd. In simple terms, it produces 1 when the inputs are different and 0 when they are the same.



Boolean Expression:

1. Two-input XOR gate: $\boldsymbol{Y = A \oplus B = (A.\overline{B}) + (\overline{A}.B)}$

$\boldsymbol{Y = A \oplus B}$		
A	**B**	**Y**
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**0**

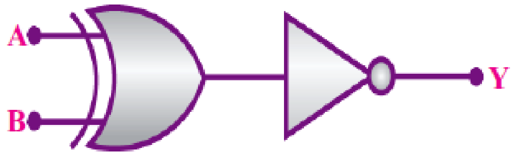
2. Three-input XOR gate:

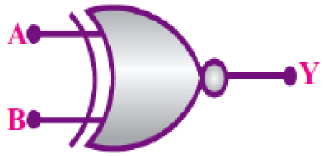
$$\boldsymbol{Y = A \oplus B \oplus C = A.\overline{B}.\overline{C} + \overline{A}.B.\overline{C} + \overline{A}.\overline{B}.C + A.B.C}$$

$\boldsymbol{Y = A \oplus B \oplus C}$			
A	**B**	**C**	**Y**
0	**0**	**0**	**0**
0	**0**	**1**	**1**
0	**1**	**0**	**1**
0	**1**	**1**	**0**
1	**0**	**0**	**1**
1	**0**	**1**	**0**
1	**1**	**0**	**0**
1	**1**	**1**	**1**

XNOR Gate (Exclusive-NOR Gate)

In the XNOR gate, the output is in state 1 when both inputs are the same, that is, both 0 or 1.





1 Two-input XNOR gate

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

Inputs		Output
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

2 Three-input XNOR gate

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

Three Input $Y = ((\overline{A}).\ (\overline{B})\ \overline{C}) + (A.B.C)$			
A	*B*	*C*	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

3. Four input

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

Four Input

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

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Demonstration of Logic Gates in Simple Circuits:

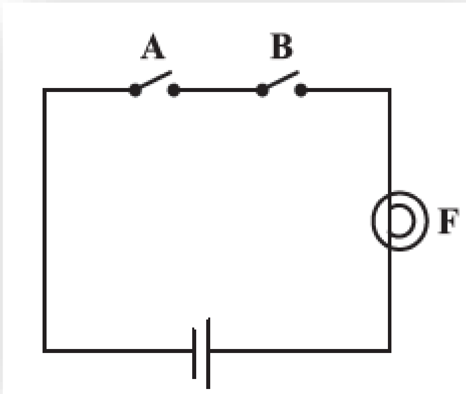
In this part, we are going to have some fun with simple circuits. By utilizing just 2 switches, a lamp and a battery, we explore the behaviors of a 2-input AND gate and a 2-input OR- gate. These hands-on examples offer a concrete insight into how logic gates process signals and make logical decisions, forming the basis for more complex digital systems.

I. AND Gate Demonstration:

In this demonstration, we will use the following components as shown in the circuit diagram:

- ► 2 switches (Switch A and Switch B)
- ► I lamp
- ► 1 battery

The presented circuit exhibits two switches (A & B) and a bulb (F) arranged in series with a cell. According to basic circuit theory, the bulb illuminates only when a closed circuit permits the flow of current.

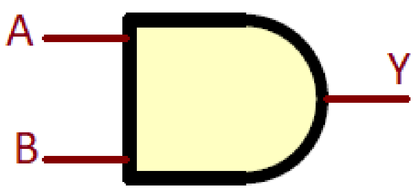


Behavior: The lamp will light up only when both Switch A and Switch B are in the ON position. If either or both Switches are OFF, the lamp remains off.

However, in the context of digital electronics, we approach the analysis differently. Instead of focusing on the physical flow of current, we interpret the circuit as making decisions, determining whether the output represents logic 1(the bulb is lit) or logic O (bulb not lit). This decision is contingent upon the logical states of the inputs (i.e., the switches), where an open switch corresponds to logic O, and a closed switch corresponds to logic 1. The truth table and the equivalent combination of the above circuit in logic gates are shown in the figure.

Logic AND Gate

Symbol



Boolean Expression

Y= A.B

Truth Table

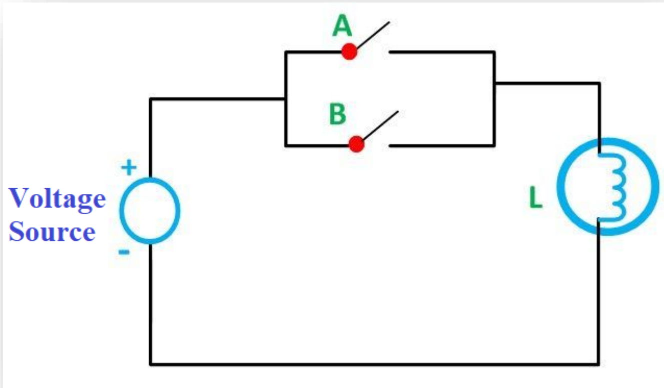
Logic Input		Logic Output
B	A	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate Demonstration:

In this demonstration, we will use following components as shown in the circuit diagram:

- ► 2 switches (Switch A and Switch B)
- ► I lamp
- ► 1 battery

Figure shows a simple circuit that contains two switches in parallel.



Behavior: The lamp will light up if either or both Switch A and Switch B are in the ON position. The lamp remains off only when both switches are OFF.

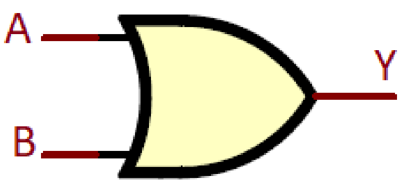
The truth table and the equivalent combination of the above circuit in logic gates is shown in fig In these circuits:

- ► For the AND gate, the output is HIGH (lamp on) only when both inputs are HIGH.
- ► For the OR gate, the output is HIGH (lamp on) when at least one input is HIGH.

These simple circuits demonstrate the basic principles of AND and OR gates, showcasing how they process signals based on logical operations.

Logic OR Gate

Symbol



Boolean Expression

Y= A+B

Truth Table

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