

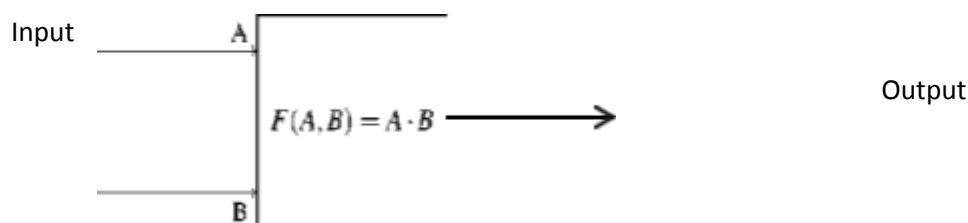
**Unit no 03: Digital Systems and Logic Designs****Exercise:****Give Short answers to the questions.****1. Define a Boolean function and give an example?**

A Boolean function is a function which has one or more binary inputs and produces a single binary output. The inputs and outputs can only have two values: False (represented by 0) and True (represented by 1). The construction of Boolean functions is done by employing the basic logical operations such as AND, OR and NOT, which connect the inputs to generate the correct output.

**Example : Simple Boolean Function**

Consider a Boolean function with two inputs, A and B. We can construct a function F that represents the AND operation:

$$F(A, B) = A \cdot B$$



The diagram shown above demonstrates a basic digital circuit, which is an AND gate. The box symbolizes the AND function  $F(A, B) = A \cdot B$ . This box has two inputs A and B. If both A and B are 1, the output will be 1. In any other case, the output will be 0. The input are shown at the entrance to the box, while the output is depicted at the exit of the block. The truth table for this function is as follows:

A	B	$F(A, B)$
0	0	0
0	1	0
1	0	0
1	1	1

**2. What is the significance of the truth table in Digital Logic?**

Significance/Importance of the Truth Table in Digital Logic:

A **truth table** is a crucial tool in digital logic design because it:

- Defines the Function Behavior Clearly**

It shows all possible combinations of input values and the corresponding output, making the logic of a digital circuit easy to understand.

- Helps in Circuit Design**

Engineers use truth tables to design combinational logic circuits by converting logical requirements into actual hardware (using gates like AND, OR, NOT).

**c. Basis for Boolean Expression Simplification**

It helps in writing the corresponding **Boolean expression**, which can then be simplified using techniques like Karnaugh maps or Boolean algebra.

**d. Error Detection and Debugging**

Designers can spot errors or incorrect outputs by comparing a circuit's actual behavior to its expected truth table.

**3. Explain the difference between analog and digital signals?****Analog signals:**

Analog signals are signals that change with time smoothly and continuously over time. They can have any value within a given range. Examples include voice signal (speaking), body's temperature and radio-wave signals.

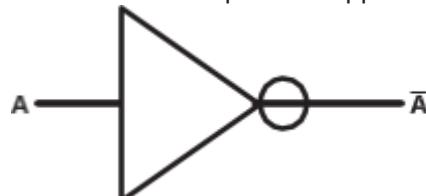
**Digital signals:**

Digital signals are the signals which have only two values that are in the form of '0' and '1'. These are utilized in digital electronics and computing systems.

Analog Signal	Digital Signal
<ul style="list-style-type: none"> <li>Continuous</li> <li>Infinite possible values</li> <li>Example: Sound waves</li> </ul>	<ul style="list-style-type: none"> <li>Discrete</li> <li>Finite (0 or 1)</li> <li>Example: Binary data in computers</li> </ul>

**4. Describe the function of a NOT gate with its truth table.**

**NOT Gate:** Implements the NOT function. It outputs the opposite of the input.

**NOT Operation:**

The NOT operation is one of the basic Boolean algebra operations which takes a single binary variable and simply negates its value. If the input is one, the output is zero and if the input is zero, the output is one.

Example: Consider a binary variable:

$$A = 1 \text{ (true)}$$

The NOT operation for this variable can be written mathematically as:

$$P = \overline{A} \quad \text{or} \quad P = \neg A$$

**Truth Table:** Below is the truth table for the NOT operation.

A	NOT A (P)
0	1

1	0
---	---

**5. What is the purpose of a Karnaugh map in simplifying Boolean expressions?**

A Karnaugh map (K-map) is a graphical representation which can be used to solve Boolean algebra expressions and minimize a logic function where algebraic computations are not employed. It is a technique in which the truth value of Boolean function is plotted to enable the identification of patterns and to perform term combining for simplification. A **Karnaugh map (K-Map)** is a visual tool used to **simplify Boolean expressions** without using complex Boolean algebra rules. Its main purpose is to **minimize logic expressions** by grouping adjacent 1s (or 0s) in a truth table, helping reduce the number of logic gates in a circuit.