UNIT 03

Digital Systems and Logic Design

Short Introduction of Unit

In this Chapter, we will discuss the Boolean functions, logic, digital logic and difference between analog and digital signals. We will also discuss several types of gates, their truth tables, and digital devices including half and full adders.

Q.1 Describe Analog and Digital Signals with their conversion. 09503001

Ans. Digital systems are the backbone of today's electronics and computing. They manipulate digital information in the form of binary digits, which are either 0 or 1 and are used in calculation devices such as calculators and computer, among others.

1. Analog signals

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

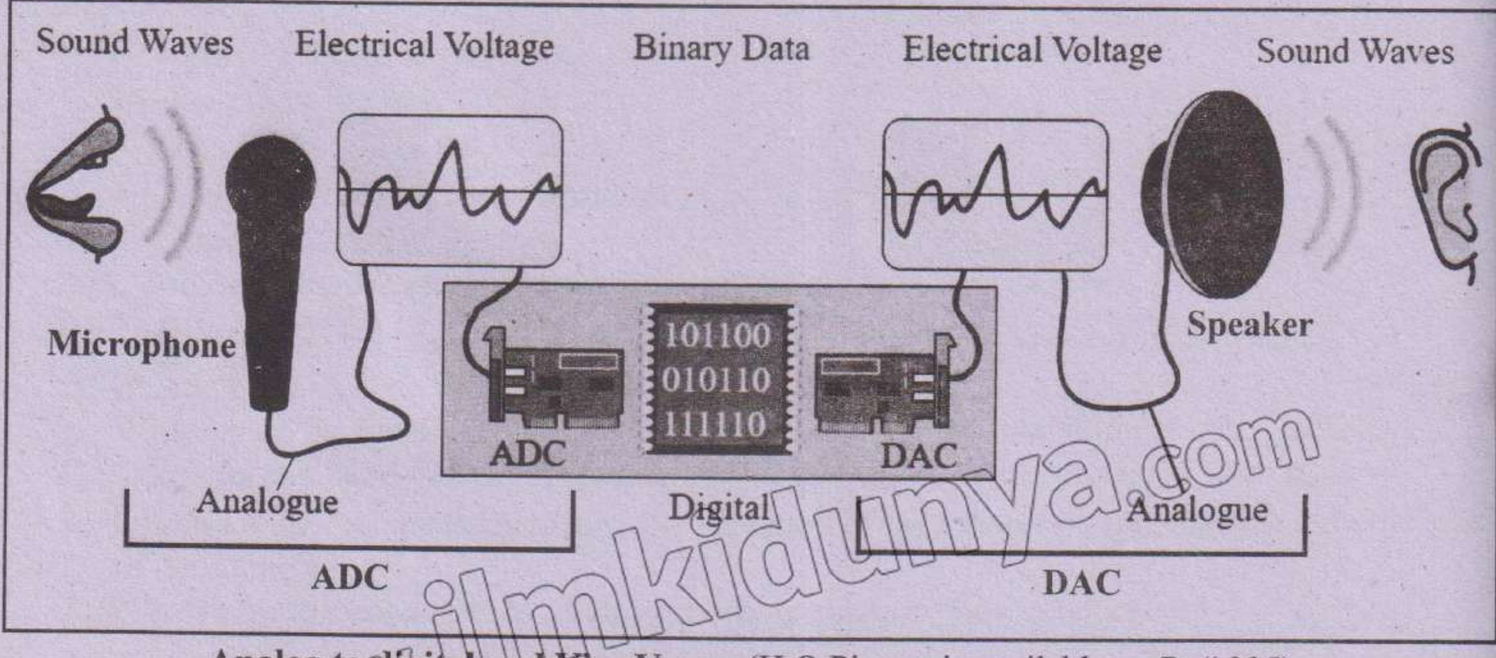
2. Digital signals

Digital signals are the signals which have only two values that are in the form '0' and '1 these are utilized in digital electronics and computing systems. Analog to digital converter (ADC) and digital to analog converting (DAC) are important operations in today's technological developments, enabling the transmission and control of signals.

Analog Signal	Digital Signal
Continuous	Discrete
Infinite possible values	Finite (0 or 1)
Example: Sound waves	Example: Binary data in computers

Analog to digital (ADC): ADC is the conversion of analog signals into digital signals, which are discrete and can be easily processed by computerized devices like computers and smart phone.

Digital to Analog Conversion (DAC): DAC is the conversion whereby analog signals are converted to digital signals, making it possible for human to perceive the information, for instance through speakers, as depicted in figures.



Analog to digital and Vice Versa (H.Q Picture is available on Pg# 235)

ADC and DAC Conversion

Digital to analog conversion, and vice versa, is critical since it enables data processing, storage, and transmission. Digital signals are much less affected by noise and signal degradation and are therefore better suited for transmitting and information over-long distance.

Example: Sound waves

Let us consider a situation where one person is speaking into a microphone while the other person is receiving sound through speakers as illustrated in the figure.

1. **Microphone (ADC):** When you speak into the microphone your voice produces sound waves (analog signals) that are captured by the system). This is done by converting the sound waves into digital form using an ADC with the microphone.
2. **Speakers (DAC):** At the receiver end, the digital signals are then converted back into analog signals with the help of DAC. The speakers then translate these analog signals back into sound waves to enable you hear to the other person's voice as if they were speaking directly to you.

Q.2 Describe Boolean Algebra and Logical Gates with the help of Corresponding Truth Tables. OR Explain the usage of Boolean functions in computers. 09503002

Ans. Boolean algebra is a branch of mathematics relate to logic and symbolic computation, using two values namely True and False. It is an essential branch of digital circuits since it is the basis for the analysis and design of circuits.

Boolean Functions and Expressions

Binary values are used to describe the relationship between variables in the Boolean function and Boolean expressions. The expressions are built using **AND**, **OR**, and other logic operations and can in several ways be reduced to optimize digital circuits.

Binary Variable sand Logic Operations

Binary variables that can have only have two values, 0 and 1. Logic operations are basic operations implemented in Boolean algebra for processing of these binary variables. The primary logic operations **AND**, **OR** and **NOT**.

1. AND Operation

AND is the basic logical operator which is used in Boolean algebra. It requires two binary inputs which will give a single binary output The symbolic used for the AND operation. The output of the AND operation is "1" only when both inputs are "1". Otherwise, the result is "0".

Example: A = 1 (True)

B = 0 (False)

The AND operation for these variables can be written mathematically as:

P = A.B

In this example: A = 1, B = 0

Therefore, then, the result P of the AND operation is 0 (false).

Truth Table

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

2. OR Operation

The OR operation is another basic logical operation in Boolean algebra. To be specific this is also a function tables two binary variables as input produces a single binary output. According to table, OR operation yield true (1) output when at least of '1' of the inputs in true (1). The output is 0 only when both inputs are '0'.

Example

Consider two binary variables:

$$A = 1 \text{ (true)}$$
$$B = 0 \text{ (false)}$$

The OR operation for these variables can be written mathematically as:

$$P = A + B$$

In this example

$$A = 1, B = 0$$

Therefore, result P of the OR gate will be 1.

Truth Table

A truth table is useful for better understanding of how the OR operation is organized and what the result of the OR's application is for all variants of the input variables. Below is the truth table for the OR operation.

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Did you know?

In binary logic, 1 + 1 does not equal 2 but equals 1 in logical operation. This is because the OR operation returns a value of 1 if any or both of the inputs to this operator are 1.

3. 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:

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

This signifies that if you have A = 1 (true), the result of NOT operation is going to be 0 (false).

Truth Table

A	NOT (P)
0	1
1	0

Q.3 How Are Boolean Functions Constructed? Provide a Detailed Explanation. 09503003

OR Describe how to construct a truth table for a Boolean expression with an example.

Ans. Boolean functions are algebraic statements that describe the relationship between binary variables and logical operations. These functions are particularly important for digital logic design and are employed in formation of various digital circuits, which are the basis of current computers, mobile phone and even simple calculator.

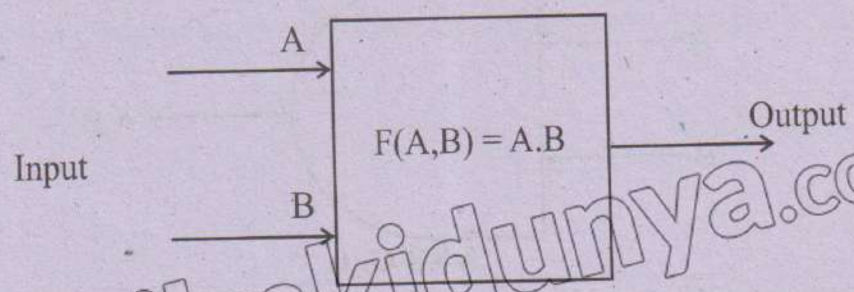
Understanding Boolean Function

A Boolean function is a function which has a one or more binary inputs and products 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 1: Simple Boolean Function

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

$$F(A, B) = A.B$$



Simple Boolean Function

The diagram shown above demonstrates a basic digital circuit, which is an AND gate. The box symbolizes the AND function F(A, 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 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

Example 2: Now, let us construct a more complex Boolean function with three inputs, A, B and C.

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

This function uses AND, OR and NOT at the same time. The truth table for this function is as follows:

Explanation

- The parameters A, B and C are included in the following example as the input columns.
- The results of AND operation between two variable A and B are presented in the column A.B.
- The column $\overline{A}$ standing for the NOT operation of A.
- Every value in the column $\overline{A}$. C displays the result of AND operation between the values in the fifth column and the third column.
- The final column F(A, B, C) shows the output of the Boolean function $(A.B) + (\overline{A} .C)$

A	B	C	A.B	$\bar{A}$	$\bar{A}$.C	F(A, B, C)
0	0	0	0	1	0	0
0	0	1	0	1	1	1
0	1	0	0	1	0	0
0	1	1	0	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	1
1	1	1	1	0	0	1

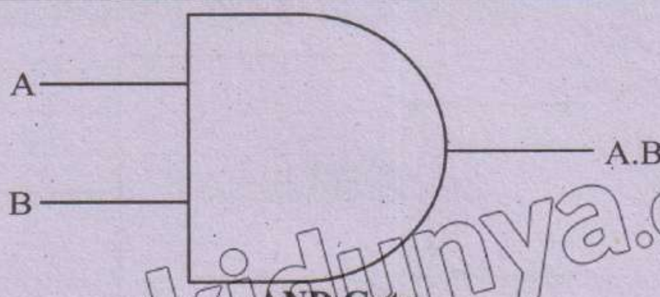
Did you know?

George Boole, a Mathematician who invented Boolean algebra was born in Lincoln, England in the year 1815.

Q.4 Discuss Logic Gates and their Functions. 09503004

Ans. Logic gates are physical devices in electronic circuits that perform Boolean operations. Each type of logic gate corresponds to a basic Boolean operation. Example of the logic gates are:

AND Gate: Implements the AND function. It outputs true only when both inputs are True (1)

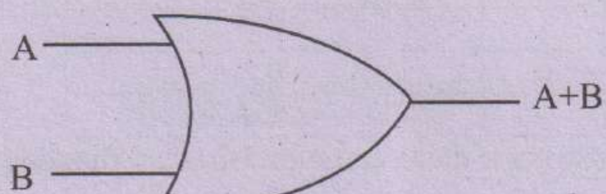


AND Gate

Imagine a simple electronic circuit with an AND gate. If you press two switches (both must be ON), a light bulb will turn on.

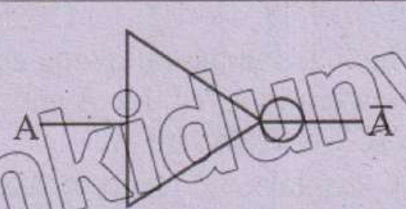
- Switch 1: ON (True)
- Switch 2: ON (True)
- Light bulb: ON (True) because both is switches are ON.
- If either switch is OFF, the light bulb will be OFF.

OR Gate: Implements the OR function. It outputs true when at least one input is true.



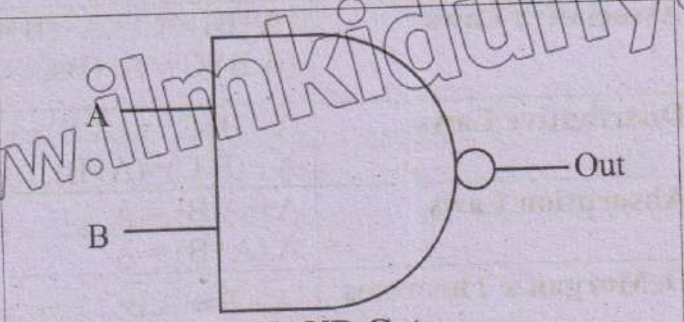
OR Gate

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



NOT Gate

NAND Gate: This gate is achieved when an AND gate is combined with a NOT gate. It generates true when at least one of the inputs in false. In other words, it is inverse of the AND gate, as presented in figure.

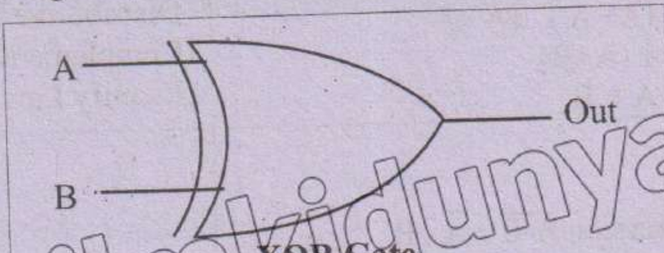


NAND Gate

Example: Imagine a safety system where an alarm should go on if either one of two sensors detects an issue.

- Sensor 1: No issue (False)
- Sensor 2: Issue detected (True)
- Alarm: ON (True) because one sensor detects an issue.

XOR Gate: The XOR (Exclusive OR) gate outputs true only when exactly one of the inputs is true. It differs from the OR gate in that it does not output true when both inputs are true. It is shown in Figure.



XOR Gate

Example: Imagine a scenario where you can either play video games or do homework, but not both at the same time.

- Play video games: Yes (True)
- Do homework: No (False)
- Allowed? Yes (True) because only one activity is being done.

Q.5 Mention and describe fundamental Boolean Algebraic rules with the help of examples. 09503005

OR

Describe the concept of duality in Boolean algebra and provide an example to illustrate it.

Ans. Simplification of Boolean function is a particularly important process in designing an efficient digital circuit. Such simplified functions required fewer gates making them compact in size, energy efficient and faster than the complicated ones. Simplification means applying of some Boolean algebra rules to make the functions less complicated.

1.	**Identity Laws**	$A + 0 = A$ $A.1 = A$
2.	**Null Laws**	$A + 1 = 1$ $A.0 = 0$
3.	**Idempotent Laws**	$A + A = A$ $A.A = A$
4.	**Complement Laws**	$A + \overline{A} = 1$ $A.\overline{A} = 0$

5.	**Commutative Laws**	$A + B = B+A$ $A.B = B.A$
6.	**Associative Laws**	$(A+B) + C = A+(B+C)$ $(A.B).C = A. (B.C)$
7.	**Distributive Laws**	$A. (B+C) = (A.B)+ (A.C)$ $A+ (B.C)=(A+B). (A+C)$
8.	**Absorption Laws**	$A+(A.B) = A$ $A.(A+B) = A$
9.	**DeMorgan's Theorems**	$\overline{A+B} = \overline{A}.\overline{B}$ $\overline{A.B} = \overline{A} + \overline{B}$
10.	**Double Negation Law**	$\overline{\overline{A}} = A$

Simplification Examples

Example 1 Simplify the expression $A+\overline{A}.B$ **Solution** $A+\overline{A}.B = (A+\overline{A}).(A+B$ $= 1.(A+B)$ $= A + B$	(Distributive Law) (Complement Law) (Identify Law)
Example 2 Simplify the expression $\overline{A.B}+\overline{A}.B$ **Solution** $\overline{A.B}+\overline{A}.B = \overline{A}+\overline{B}+\overline{A}.B$ $= (\overline{A}+\overline{B})$ $= \overline{A}+\overline{B}$	**(De Morgan's Theorem)** (Since $\overline{A}$ is already present in $\overline{A}.\overline{B}$, we can use absorption law.
Example 3 Simplify the expression $(A+B).(A+\overline{B})$ **Solution** $(A+B).(A+\overline{B}) = A.(A+\overline{B})+B.(A+\overline{B})$ $= A+A.\overline{B}+B.\overline{B}$ $= A+A.B$ $= A.(1+B)$ $= A.1$ $= A$	(Distributive Law) (Absorption Law) (Identity Law) (Distributive Law) (Null Law) (Identify Law)
Example 4 Simplify the expression $\overline{\overline{A}+B}.(A+\overline{B})$ **Solution** $\overline{\overline{A}+B}.(A+\overline{B}) = (A.\overline{B}).(A+\overline{B})$ $= A.\overline{B}.A+A.\overline{B}.\overline{B}$ $= A.\overline{B}+A.\overline{B}$ $= A.\overline{B}$	(De Morgan's Theorem) (Distributive Law) (Idempotent Law) (Identity Law)

Q.6 What are the Applications of Digital Logic? OR 09503006

Compare and contrast half-adders and full-address, including their truth tables, Boolean expression, and circuit diagrams.

Ans. Digital logic an essential aspect for the functioning of several modern electric systems, such as computers, smart phones, and other digital gadgets. Digital logic optimize in many ways in order to create and enhance circuits meant to perform various tasks. Two important applications of digital logic are the design of adder circuits and the use of Karnaugh maps for function simplification.

Half-adder and Full-adder Circuits

Adder circuits are widely used in the digital circuits to perform arithmetic calculations. There are two general forms of adder circuits known as half-adders and full adders.

1. Half-adder Circuits

A half adder is a basic circuitry unit that performs addition of two single-bit binary digits. It has two inputs, usually denoted as A and B, and two outputs: the sum (S) and the carry (C).

Truth Table for Half-adder

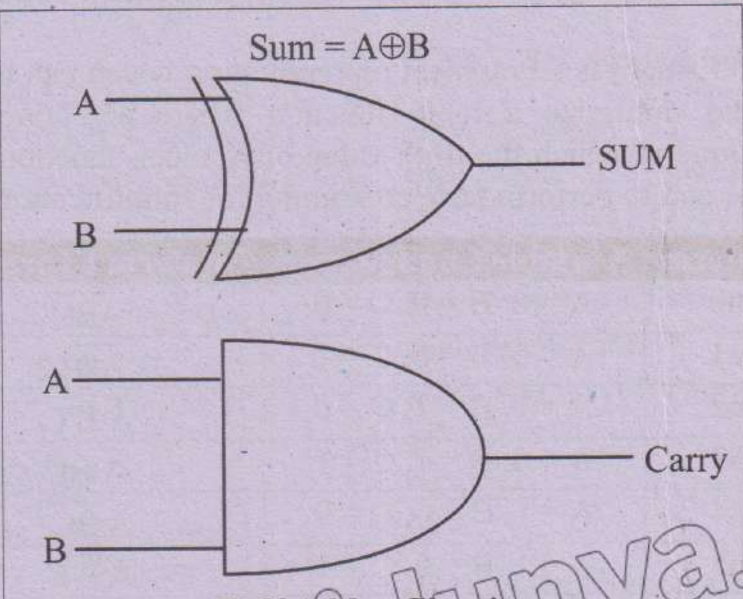
A	B	Sum (S)	Carry(C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Boolean Expressions for Half-adder:

$$S = A \oplus B$$
$$C = A.B$$

In this case the symbol $\oplus$ represents the XOR operation. The sum output is high when only one of the inputs is high, while the carry output is high when both inputs are high.

Boolean Expressions



Half Adder Circuits

2. Full-adder Circuits

A full-adder is a more complex circuit that adds three single-bit binary numbers. Two bits that belong to sum and a carry bit from a previous addition. It has three inputs, denoted as A, B, and

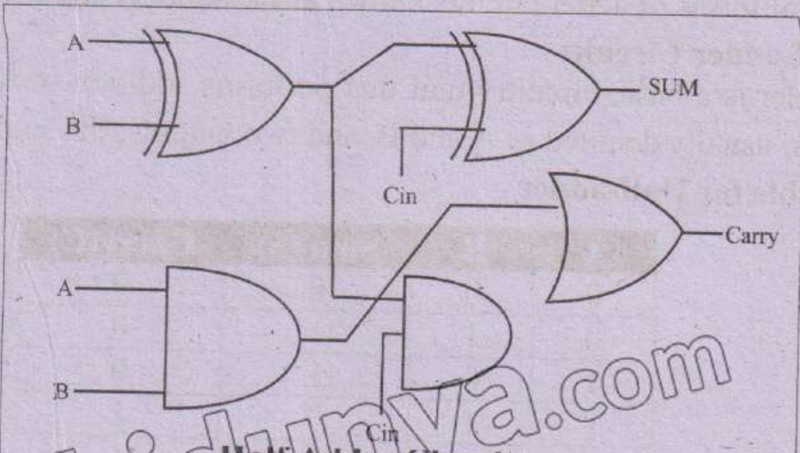
C_{in} (carry input), and two outputs: called the sum (S) and the carry (Cout) with both being integer values.

A	B	C_{in}	Sum (S)	Carry (C_{out})
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Boolean Expressions

Sum = $A \oplus B \oplus Cin$

Carry = $(A.B) + (Cin.(A \oplus B))$



Half Adder Circuits

The sum output is high if the number of high inputs is odd whereas the carry output is high if the number of high inputs is at least 2.

Q.7 What do you know about Karnaugh Map (K-Map)? Describe its structure. 09503007

OR

How do Karnaugh maps simplify Boolean expression? Provide a detailed example with steps.

Ans. 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.

Minterm	Variable Combination	Minterm Expression
m0	A = 0, B = 0, C = 0	$\overline{ABC}$
m1	A = 0, B = 0, C = 1	$\overline{AB}\,C$
m2	A = 0, B = 1, C = 0	$\overline{A}\,B\,\overline{C}$
m3	A = 0, B = 1, C = 1	$\overline{A}\,BC$
m4	A = 1, B = 0, C = 0	$A\overline{BC}$
m5	A = 1, B = 0, C = 1	$A\overline{B}C$
m6	A = 1, B = 1, C = 0	$AB\,\overline{C}$
m7	A = 1, B = 1, C = 1	ABC

Minterms foe A,B and C.

A K-map is a matrix where each square is a cell, which corresponds to a positioned combination. These cells are filed with '1' or '0' in reference to the truth table of the Boolean function. The size of the K-map depends on the number of variables:

- 2 Variables: 2×2 grid
- 2 Variables: 2×4 grid
- 4 Variables: 4×4 grid
- 5 Variables: 4×8 grid (less common for manual simplification)

Every cell in the K-map represent a minterm, and the cells in each row of the K-map differ by only one bit at any particular position, following the gray code sequence.

Topic Wise Short Questions(Additional)

Digital Systems

Q.1 What are the digital systems? 09503008

Ans. Digital systems are the basis of the present-day electronics and computing. They process digital data in form of '0' and '1'.

Q.2 What do you know about Analog systems? 09503009

Ans. Analog signals are continuous time varying signal.

Q.3 How ADC is processed? 09503010

Ans. ADC (Analog to Digital Converter) is the process of converting the devices for example continuous signals into discrete signals that can be processed by digital devices for example computers and smart phones.

Q.4 How DAC is processed? 09503011

Ans. DAC (Digital to Analog Converter) converts the digital signal back to the analog signal.

Q.5 Define Digital Logic. 09503012

Ans. Digital logic is the basis of all digital systems. This is the technique we use to process digital information in the form of binary numbers.

Boolean Algebra

Q.6 What is Boolean Algebra? 09503013

Ans. Boolean algebra is a sub-discipline of mathematics based on operations involving binary variables.

Q.7 What do you know about AND Operation? 09503014

Ans. In the case of AND operation the output is 1 only when both input values are 1. Otherwise, the output is 0.

Q.8 What do you know about OR Operation? 09503015

Ans. In an OR gate, the result is 0 only when both the input values are 0. Otherwise, the output is 1.

Q.9 What do you know about NOT Operation? 09503016

Ans. The NOT operation the simplest logical operation in Boolean algebra, which accept a single binary inputs and gives its opposite as the outputs.

A	NOT (A)
0	1
1	0

Q.10 What are the Boolean Functions? 09503017

Ans. Boolean functions are mathematical expressions that represent logical operations involving binary variables.

Example: AND, OR, NOT

Q.11 What are the basic logic gates and their truth tables? 09503018

Ans. AND Gate: Outputs 1 when all inputs are 1.

OR Gate: Returns 1 if at least one input is 1.

NOT Gate: Returns the complement of the input.

These basic gates serve as the foundation for NAND, NOR, XOR, and XNOR gates.

Digital Circuits

Q.12 What is the crucial element of digital circuit? 09503019

Ans. A crucial element of digital circuit design is the logical diagram, which represents the structure of the circuit by showing connections between logic gates.

Q.13 Why we use adder circuits? 09503020

Ans. Adder circuits are widely used in the digital electronic systems with the principal application in arithmetic operations.

Q.14 Differentiate between Half and Full adder. 09503021

Ans. A half-adder is a digital circuit used to compute the addition of two single-bit binary numbers. **A full-adder** is a more complex circuit that adds three single-bit numbers: two main bits and a carry bit from a previous addition.

K-Map

Q.15 Define K-map. 09503022

Ans. A Karnaugh map (K-map) is a graphic aid that is employed in simplification of Boolean expressions and minimizing logic functions without the used for comple algebraic operations.

Q.16 What is a multiplexer, and how does it work in digital circuits? 095030

Ans. A multiplexer (MUX) chooses one c many input signals and sends it to the outpu based on control signals.

Q.17 Explain the word propagation delay in terms of digital logic gates. 09503024

Ans. Propagation delay is the time it takes for an input change to result in an outpu change in a logic gate.It influences th overall performance of digital circuits.

Q.18 What do you know about minterm? 09503025

Ans. In Boolean algebra, a minterm is a particular product term whereby ever variable of the function is present in either i its true form or its complement. Each minterm corresponds to one and only one se of variable values that makes the Boolean function equal to true or 1.

Topic Wise Multiple Choice Questions (Additional)

Choose the correct option:

Digital Systems

1. Digital system depends on? 09503026

(a) 1 1 (b) 0 0
(c) 1 0 (d) None of These

2. In digital systems 0 represents?

(a) OF (b) OFF
(c) ON (d) All of These

3.In digital systems 1 represents? 09503027

(a) OF (b) OFF
(c) ON (d) All of These

4. Bit stands for_____? 09503028

(a) Binary Digit
(b) Binary Integer
(c) Binary Value
(d) Binary Number

5. What does ADC represents? 09503029

(a) Analog Digital Conversion
(b) Analog Digital Conversation
(c) Analog Digital Convenient
(d) All of These

6.What does DAC represents? 09503030

(a) Digital Analog Conversion
(b) Analog Digital Conversation
(c) Analog Digital Convenient
(d) All of These

7. Which signals are in discrete form? 09503031

(a) Analog
(b) Digital
(c) Both a,b
(d) None of these

8. Which signals are in Continuous form? 09503032

(a) Analog (b) Digital
(c) Both a,b (d) None of these

9. Which one is the example of Analog Signal? 09503033

(a) Sound Wave (b) Binary
(c) 1 0 (d) None of these

10. **Which one is the example of Digital Signal?** 09503034
(a) Sound Wave (b) Binary
(c) 1 1 (d) 0 0

Boolean Algebra

11. **The output of AND operation is 1 only when both inputs are?** 09503035
(a) 0 (b) 1
(c) 1 0 (d) 0 0

12. **OR operation yields 1 output when at least of ____ of the input is true.** 09503036
(a) 0 (b) 1
(c) 1 0 (d) 0 0

Control Unit

13. **ALU stands for______.** 09503037
(a) Arithmetic & Logic Unit
(b) Arithmetic & Local Unit
(c) Arithmetic & Legal Unit
(d) None of these

14. **CU stands for __________.** 09503038
(a) Control Unit (b) Central Unit
(c) Circuitry Unit (d) None of these

15. **Basic Arithmetic Operations like Addition, Subtraction are performed by?** 09503039
(a) ALU (b) CU
(c) UDP (d) All of these

16. **Comparison of two or more values are performed in?** 09503040
(a) CU (b) LU
(c) UDP (d) All of these

17. **In K-map, K stands for_____?** 09503041
(a) Knowledge (b) Karnaugh
(c) Both a,b (d) None of these

Answer Key

1	c	2	b	3	c	4	a	5	a	6	a	7	b	8	a
9	a	10	b	11	b	12	b	13	a	14	a	15	a	16	b
17	b														

Solved Exercise

Q.1 Choose the correct option.

1. **Which of the following Boolean expressions represents the OR operation?** 09503042
(a) A.B (b) A+B
(c) A (d) A⊕B

2. **What is the dual of Boolean expression A.0 = 0?** 09503043
(a) A+1 = 1 (b) A+0 = A
(c) A.1 = A (d) A. 0 = 0

3. **Which logic gate outputs true only if both inputs are true?** 09503044
(a) OR gate (b) AND gate
(c) XOR gate (d) NOT gate

4. **In a half-adder circuit, the carry is generated by which operation?** 09503045
(a) XOR operation
(b) AND operation
(c) OR operation
(d) NOT operation

5. **What is the decimal equivalent of the binary number 1101?** 09503046
(a) 11 (b) 12
(c) 13 (d) 14

Answer Key

1	b	2	a	3	b	4	b	5	c

Short Questions

1. Define a Boolean function and give an example. 09503047

Ans. See Short Question No. 10

2. What is the significance of the truth table in digital logic? 09503048

Ans. The truth table is an important tool in digital logic because it provides a systematic method for representing the behavior of a logic circuit or Boolean function for all possible input combinations. A truth table is useful for better understanding of how the OR operation is organized and what the result of the OR's application is for all variants of the input variables.

3. Explain the difference between analog and digital signals. 09503049

Ans. See Short Question No. 1,2

4. Describe the function of a NOT gate with its truth table. 09503050

Ans. See Short Question No. 9

5. What is the purpose of a Karnaugh map in simplifying Boolean expressions? 09503051

Ans. See Short Question No. 15

Long Questions

1. **Explain the usage of Boolean functions in computers.** 09503052
   Ans. Long Question No. 2
2. **Describe how to construct a truth table for a Boolean expression with an example.** 09503053
   Ans. Long Question No. 3
3. **Describe the concept of duality in Boolean algebra and provide an example to illustrate it.** 09503054
   Ans. Long Question No. 5
4. **Compare and contrast half-adders and full-address, including their truth tables, Boolean expression, and circuit diagrams.** 09503055
   Ans. Long Question No. 6
5. **How do Karnaugh maps simplify Boolean expression? Provide a detailed example with steps.** 09503056
   Ans. Long Question No. 7
6. **Design a 4-bit binary adder using both half-adders and full-adders. Explain each step with truth tables, Boolean expressions, and circuit diagrams.** 09503057

Ans. Half-adder Circuits

A half adder is a basic circuitry unit that performs addition of two single-bit binary digits. It has two inputs, usually denoted as A and B, and two outputs: the sum (S) and the carry (C).

Truth Table for Half-adder

A	B	Sum (S)	Carry(C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

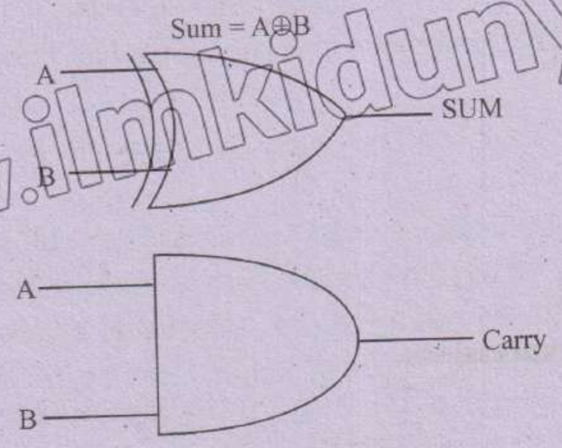
Boolean Expressions for Half-adder:

$S = A \oplus B$

$C = A.B$

In this case the symbol $\oplus$ represents the XOR operation. The sum output is high when only one of the inputs is high, while the carry output is high when both inputs are high.

Boolean Expressions



Half Adder Circuits

Full-adder Circuits

A full-adder is a more complex circuit that adds three single-bit binary numbers: Two bits that belong to sum and a carry bit from a previous addition. It has three inputs, denoted as A, B, and C_{in} (carry input), and two outputs: called the sum (S) and the carry (Cout) with both being integer values.

A	B	C_{in}	Sum (S)	Carry (C_{out})
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Boolean Expressions:

Sum = $A \oplus B \oplus Cin$

Carry = $(A.B) + (Cin.(A \oplus B)$

The sum output is high if the number of high inputs is odd whereas the carry output is high if the number of high inputs is at least 2.

7. **Simplify the following Boolean function using Boolean algebra rules:** 09503058

$F(A, B) = A.B + A.\overline{B}$

Ans.

$F(A, B) = A \cdot (B+1)$ Distributive law

$B+1=1$ Null Law

$F(A, B) = A \cdot 1$

$F(A, B) = A$ Identity Law

$F(A, B) = A$ (Answer)

8. Use De-Morgan's laws to simplify the following function: 09503059

F(A, B,C) = $\overline{A+B}+\overline{AC}$

Ans. De-Morgan's law states:

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

$\overline{A \cdot B}=\overline{A}+\overline{B}$

F (A, B, C) =(A·B) +AC

F (A, B, C) =(A·B) +(A·C)

F (A, B, C) =A·B+A·C (Result)

9. Simplify the following expressions. 09503060

(a) F = $\overline{A+B}$**. (A+B)**

Ans.

$X \cdot \overline{X}=0$ Null Law

Using Complement Law

So, F=0

(b) $(A+\overline{B}).(\overline{A}+B)$

Ans. $F=(A+\overline{B})\cdot(\overline{A}+B)$

$F=A\cdot\overline{A}+A\cdot B+\overline{B}\cdot\overline{A}+\overline{B}\cdot B$

$A\cdot\overline{A}=0$ (Complement Law)

$\overline{B}\cdot B=0$ (Complement Law)

$F=0+A\cdot B+\overline{B}\cdot\overline{A}+0$

$F=A\cdot B+\overline{A}\cdot\overline{B}$

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

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

$F=A+(\overline{A}\cdot\overline{B})+(\overline{A}\cdot C)$ **Distributive law**

$F=A+\overline{B}+\overline{A}\cdot C$ **Absorption Law**

(d) $\overline{A+B}$ **+A.B**

Ans. $F=\overline{A}\cdot\overline{B}+A\cdot B$ by using **De-Morgan's law**

(e) (A.B) + $(\overline{A}.\overline{B})$

Ans. $F=(A.B)+(\overline{A}.\overline{B})$

$F=A\odot B$

OR $F=A\oplus B$

Activities

Activity-1

09503061

Consider what do you do with your cell phone or calculator on daily basis. Can you distinguish activities that require logical choices, like entering a password to unlock your smart phone or solving a math problem? Ask your group members how Boolean functions may be utilized in the background.

Ans. Class Work\ Lab Work \ Practical Work.

Activity-2

09503062

Let's make learning these logical functions fun with an activity!

1. **AND Adventure:** Form pairs and give each pair two conditions they need to meet to win a prize (like both wearing a specific color shirt).
2. **OR Options:** Make a list of fun activities. If at least one activity is possible, the class gets extra playtime.
3. **NOT Negatives:** Ask true/false questions and have students shout the opposite answer. For example, "Is the sky green?" Students should shout "No!" (NOT True).
4. Construct a basic circuit using a breadboard, a battery, and LED lights to represent AND gate. Connect two switch which will serve as, inputs A and B. In this experiment the LED will light up only when both switches are pressed.

Ans. Class Work\ Lab Work \ Practical Work.

Activity-3

09503063

Construct a digital circuit that includes both half-adders and full-adders to add two 4-it binary numbers. Create the truth tables. Boolean expressions. And circuit diagrams for each steps.

Ans. Class Work\ Lab Work \ Practical Work

Hint: Check Long Question No.6