CHAPTER 3:

Digital Systems and Logic Design

SOLVED EXERCISE

Tick (✓) the correct answer.

1. Which of the following Boolean expressions represents the OR operation?

(a) A-B (b) A + B (c) A (d) $A \oplus B$

2. What is the dual of the Boolean expression A . 0 = 0?

(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?

(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?

(a) XOR operation (b) AND operation

(c) OR operation (d) NOT operation

5. What is the-decimal equivalent of the binary number 1101?

(a) 11 (b) 12 (c) 13 (d) 14

Answer Key:

1	2	3	4	5
b	a	b	b	c

Short Answer Questions

1. Define a Boolean function and give 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 phones and even simple calculator.

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

Ans: A truth table is a mathematical table used to evaluate the output of a digital logic circuit or a Boolean function, based on the inputs. The significance of truth tables in digital logic is:

- It evaluates output for all possible inputs.
- It verifies circuit behavior.
- It simplifies complex logic.

3. Explain the difference between analog and digital signals.

Ans:

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

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

Ans: A NOT operation performs the negative of the input variable i. e., it gives the opposite value. . If the input is one, the output is zero and if the input is zero, the output is one. This operation is important in digital logic design to generate more complex logic functions and verify the functionality of digital circuits.

A	NOT A
0	1
1	0

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

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.

Long Questions

1. Explain the usage of Boolean functions in computers.

Ans: There are many uses of Boolean functions in the computers for various operations. Here are some examples of their usage: -

- **Arithmetic Operations:** Boolean functions are used in Arithmetical Logic Units (ALUs) of CPUs to perform operations like addition, subtraction, multiplication etc.
- **Data Processing:** Boolean functions are used to process binary data in memory and storage devices, ensuring efficient data manipulation and retrieval.
- **Data encoding:** Boolean functions are used to encode data, such as text, images, and audio, into binary format.
- **Data compression:** Boolean functions are used in data compression algorithms to reduce the size of data.
- **Data encryption:** Boolean functions are used in data encryption algorithms
- **Control Logic:** Boolean functions are applied in computers to control various parts of the system's operation to function in co-ordinated manner.

Boolean functions are also present in our everyday devices like cell phones and calculators.

- **Cell Phones:** In cell phone processing, when you dial a number or press a button on a phone. A Boolean function evaluates these inputs as true or false and makes the necessary output.
- **Calculators:** Basic calculators use Boolean functions. When you feed it with numbers and the operations to be performed. Boolean logic is used to arrive at the right result.
- **Washing Machines:** Boolean functions determine cycles based on input conditions like water level, temperature, and selected mode.
- **Refrigerators:** Boolean logic regulates cooling by checking conditions like door status and internal temperature.

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

Ans: A truth table represents all possible input combinations for a Boolean expression and their corresponding outputs. It is used to analyze and simplify logical functions.

- Determine the number of variables (e.g., X,Y,Z or A,B,C).
- For n variables, there will be 2n rows in the table to cover all input combinations.
- Write down all possible combinations of 0's and 1's for the input variables in binary order (ascending from 0 to 2^{n-1}).
- Clearly specify the Boolean function or expression to evaluate.
- Compute the output of the Boolean expression for each input combination by applying logical operations step by step.
- Record the inputs and their corresponding outputs in the table.

Example of Truth Table

- Identify the Inputs: Inputs: A, B, C.
- Total combinations: $2^3=8$ rows (as there are 3 variables).
- Break Down the Function:

The given function is $F(A,B,C)=A\cdot B+A\cdot C$

Sub-expressions:

- $A\cdot B$: Logical AND between A and B.
- $A\cdot C$: Logical AND between A and C.
- $F(A,B,C)$: Logical OR of $A\cdot B$ and $A\cdot C$.

Create the Truth Table:

A	B	C	A.B	A.C	F(A,B,C)=A.B+A.C
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

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

Ans: *In Boolean algebra, duality is a fundamental concept that states that every Boolean expression has a dual counterpart. The dual of a Boolean expression is obtained by interchanging the following elements:*

- *AND (.) with OR (+)*
- *OR (+) with AND (.)*
- *0 (false) with 1 (true)*
- *1 (true) with 0 (false)*

Example:

Consider the following Boolean expression $F(A,B)=A\cdot(B+1)$

To find the dual, apply the duality rules:

- AND (.) with OR (+)
- OR (+) with AND (.)
- 1 (true) with 0 (false)

$$F_{dual}(A,B)=A+(B\cdot 0)$$

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

Ans: 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.

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

In this case the symbol ⊕ 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.

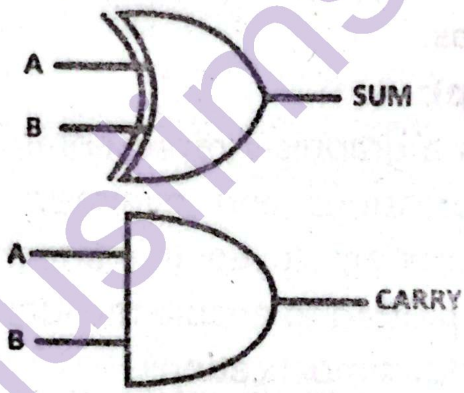
Boolean Expression:

- Sum (S)= $A \oplus B$
- Carry (C) =$A \cdot B$

Circuit Diagram:

For two single-bit binary numbers A and B, half adder produces two single-bit binary outputs S and C, where S is the Sum and C is the carry.

Sum = $A \oplus B$



Full-adder Circuits

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

Truth Table for Full-adder:

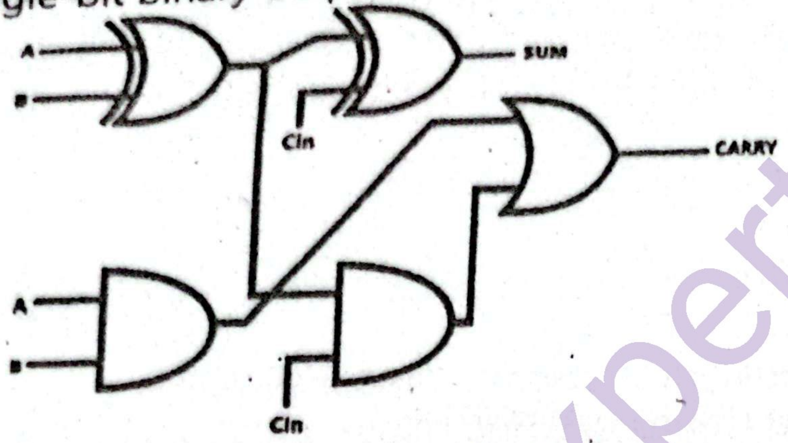
A	B	C_{in}	Sum (S)	*Carry* (C_{out})
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Boolean Expression:

- Sum = A ⊕ B ⊕ Cin
- Cout = (A . B) + (A . Cin) + (B . Cin)

Circuit Diagram:

For three single-bit binary numbers A, B, and D; the full adder circuit generates two single-bit binary outputs S (Sum), and C (Carry).



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

Ans: Karnaugh Map(K-Map):

A Karnaugh map (K-map) is a graphical representation which can be used t solve Boolean algebra expressions and minimize a logic function wher algebraic computations are not employed. It is a technique in which the trut value of Boolean function is plotted to enable the identification of patterns an to perform term combining for simplification.

Minterm	Variables Combination	Minterm Expression
m_0	A= 0, B- 0,C= 0	$\overline{A B C}$
m_1	A= 0, B= 0,C= 1	$\overline{A B}\ C$
m_2	A= 0, B= 1,C= 0	$\bar{A} B \bar{C}$
m_3	A=0, B= 1,C= 1	$\bar{A} B C$
m_4	A= 1, B= 0,C= 0	$A \overline{B C}$
'm5	A= 1, B= 0,C= 1	$A \bar{B} C$
m_6	A- 1, B- 1,C= 0	$A B \bar{C}$
m_7	A= 1, B= 1,C= 1	$A B C$

Structure of Karnaugh Maps

A K-map is a matrix where each square is a cell, which corresponds to a positioned combination. These cells are filled with T 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: 2x2 grid
- 3Variables: 2x4 grid
- 4 Variables: 4x4 grid least
- 5 Variables: 4x8 grid (less common for manual simplification)

Every cell in the K-map represents 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.

Creating Karnaugh Maps

- To create a K-map, follow these steps:
- Create a grid based on the number of variables that exists in the system.
- Let us complete the grid using the output values in the truth table.
- Arrange the 1 s in the grid in the largest possible groups of size 1,2,4,8 and so on. Every group must have one or more 1s, must be a power of two, and they must be in a continuous rows or columns.

Example: Simplifying a Boolean Expression with a K-map

To simplify the Boolean expression $A.\overline{B} + \bar{A}.B + A.B$ using a Karnaugh map (K- map):

1. Expression: $A.\overline{B} + \bar{A}.B + A.B$

Step-1: Draw the K-map Grid For two variables A and B:

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

Step 2: Fill in the K-map

For $A = 0$ and $B = 0$ F= $A.\overline{B} + \overline{A}.B + A.B$ $0.1+0.0+0.0 = 0$

For $A = 0$ and $B = 1$ F= $A.\overline{B} + \overline{A}.B + A.B$ $0.0+1.1+0.1 = 1$

For $A = 1$ and $B = 0$ F= $A.\overline{B} + \overline{A}.B + A.B$ $1.1+0.0+1.0 = 1$

For $A = 1$ and $B = 1$ F= $A.\overline{B} + \overline{A}.B + A.B$ $1.0+0.1+1.1 = 1$

Step 3: Group the I's in the K-map

Group adjacent 1's to simplify the expression

	$\overline{B}=0$	$B=1$
$\bar{A}=0$	0	1
$A=1$	[redacted]	

Group the two I's in the second column: $\overline{A}.B + A.B$

$\overline{A}.B + A.B = (\overline{A}+A).B = 1.B = B$

Group the two 1's in the second row. $A.\overline{B} + A.B$

$A.\overline{B} + A.B = (\overline{B}+B).A = 1.A = A$

Final Simplified Expression

F (A, B) = B + A

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.

Ans: Here's a step-by-step design of a 4-bit binary adder using half adders and full adders:

Step 1: Design a Half Adder

A half adder is a digital circuit that adds two single-bit binary numbers. It has two inputs (A and B) and two outputs (Sum and Carry).

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

In this case the symbol ⊕ 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.

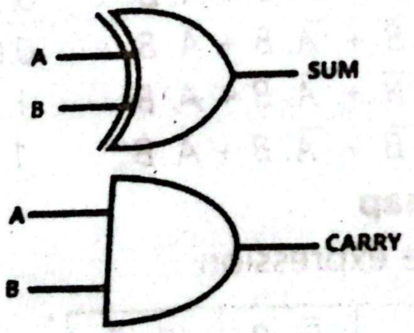
Boolean Expression:

- Sum (S)= $A \oplus B$
- Carry (C) = $A \cdot B$

Circuit Diagram

A half adder consists of an XOR gate and an AND gate.

Sum = $A \oplus B$



Step 2: Design a Full Adder

A full adder is a digital circuit that adds three single-bit binary numbers. It has three inputs (A, B, and C_{in}) and two outputs (Sum and C_{out}).

Truth Table for Full-adder:

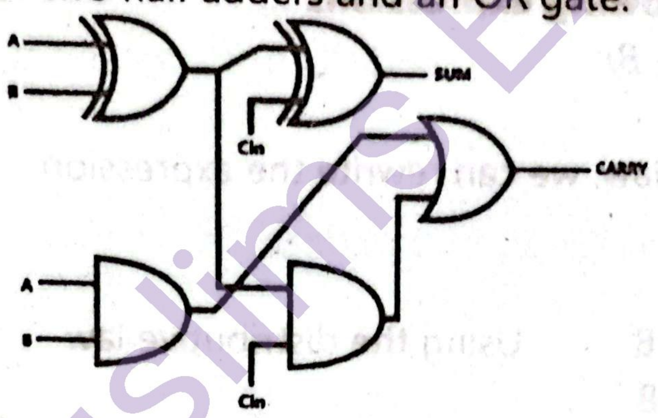
A	B	C_{in}	Sum (S)	*Carry* (C_{out})
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Boolean Expression:

- Sum = $A \oplus B \oplus C_{in}$
- $C_{out} = (A \cdot B) + (A \cdot C_{in}) + (B \cdot C_{in})$

Circuit Diagram

A full adder consists of two half adders and an OR gate.

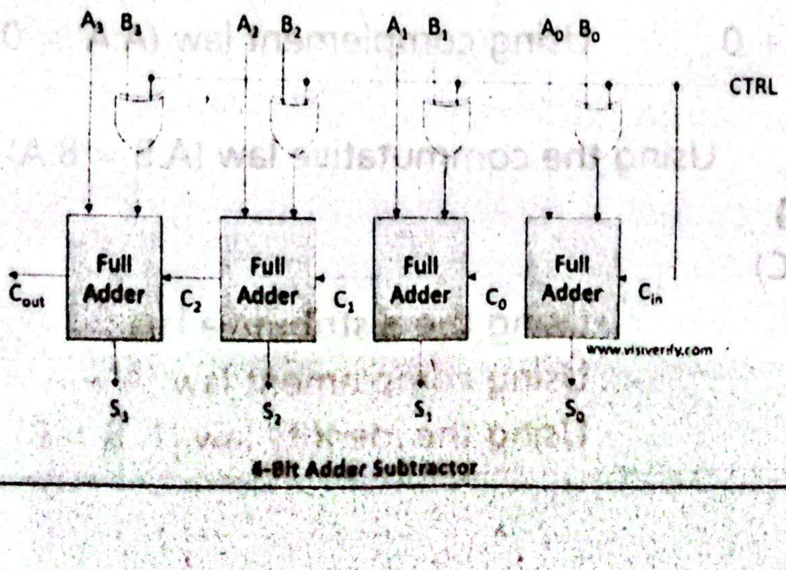


Step 3: Design a 4-Bit Binary Adder

A 4-bit binary adder adds two 4-bit binary numbers. It has eight inputs (A3, A2, A1, A0, B3, B2, B1, B0) and five outputs (S3, S2, S1, S0, Cout).

Circuit Diagram

A 4-bit binary adder consists of four full adders. The carry output of each full adder is connected to the carry input of the next full adder.
Here's the complete circuit diagram:



4-Bit Adder Subtractor

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

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

Ans: $A.B + A.\overline{B}$

$= A.(B + \bar{B})$ Using the distributive law

$= A.1$ Applying the complement law ($B + \bar{B} = 1$)

$= A$ Using the identity law ($A.1 = A$)

8. Use De Morgan's laws to simplify the following function: F (A, B, C) = A + B + AC

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

De Morgan's laws state that:

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

Using this law to the expression, we get:

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

9. Simplify the following expressions

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

Ans: (A+B)'.(A+B)

Using De Morgan's law, we can rewrite the expression:

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

$= (\bar{A}.\bar{B}).(A+B)$

$= (\bar{A}.\bar{B}).A + (\bar{A}.\bar{B}).B$ Using the distributive law

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

$= 0.(\bar{B}.A) + \overline{(A}.B).0$ Using the complement law (A'.A = 0) and (B'.B = 0), we get:

$= 0 + 0$

$= 0$

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

Ans: $(A + \bar{B}).(\bar{A}. + B)$

$= A.\bar{A}. + A.B + \bar{B}.\bar{A}. + \bar{B}.B$ Using the distributive law

$= 0 + A.B + \bar{B}.\bar{A} + 0$ Using complement law (A.A' = 0) and (B'.B = 0)

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

$= A.B + \bar{A}.\bar{B}$ Using the commutative law (A.B = B.A)

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

Ans: $A + \bar{A}.(\bar{B} + C)$

$A + \bar{A}.\bar{B} + \bar{A}.C$ Using the distributive law

$1.\bar{B} + \bar{A}.C$ Using complement law ($A + \bar{A} = 1$)

$\bar{B} + \bar{A}.C$ Using the identity law ($1.\bar{B} = \bar{B}$)

OR

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

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

The Absorption Law states:

$x + \bar{x}.y = x + y$

In this case, $x = A \; and \; y = (\bar{B} + C)$

Applying the absorption law:

$F = A + (\bar{B} + C)$

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

Ans: = $(\bar{A}.\bar{B})$+ A.B Using De Morgan's Law

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

(f) $(A.B) + (\bar{A}.\bar{B})$

Ans: $(A.B) + (\bar{A}.\bar{B})$

= $(A.B) + (\overline{A + B})$ Using De Morgan's Law

OR

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

$= \overline{((\bar{A}.\bar{B}).(A + B))}$ Using De Morgan's Law

$=(\overline{\bar{A}.A + \bar{B}.A + \bar{A}.B + \bar{B}.B})$ Using the distributive law

$= \overline{(0.A.\bar{B} + \bar{A}.B + 0)}$ Using identity law

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

$= \overline{(A.B)}$ Using absorption law

Additional Multiple Choice Questions (MCQs)

1. What binary values are used in digital circuits?

a) 2 and 3 b) 0 and 1 c) -1 and 1 d) 0 and 5

2. What is the typical voltage for binary '1' in a digital circuit?

a) 0 volts b) 3 volts c) 5 volts d) 10 volts

3. Which operation has an output of 1 only when both inputs are 1?

a) OR b) AND c) NOT d) XOR

4. What is the result of the AND operation when both inputs are 1?

a) 0 b) 1 c) -1 d) None of the above

5. What is the result of the OR operation when at least one of the inputs is 1?

a) 0 b) 1 c) -1 d) None of the above

6. What is the output of the NOT operation for input 0?

a) 0 b) 1 c) -1 d) Undefined

7. Which of the following is a Boolean function?

a) F(A,B)=A+B b) F(A,B)=A−B c) F(A,B)=A×B d) F(A,B)=A/B

8. What is the output of the OR operation for inputs A = 0 and B = 1?
a) 0 b) 1 c) 2 d) Undefined

9. What does the truth table for an AND operation show when both inpu are 0?
a) 1 b) 0 c) Undefined d) Both 0 and 1

10. Which operation combines inputs to always produce an output of any input is 1?
a) AND b) NOT c) OR d) NAND

11. Which device uses Boolean logic for operations like addition a subtraction?
a) Memory unit b) ALU c) Display screen d) Input device

12. What is an example of an analog signal?
a) A digital image b) A sound wave
c) A binary code d) A digital signal

13. What is the purpose of (ADC)?
a) To convert digital signals into analog signals
b) To convert analog signals into digital signals
c) To process digital signals d) To store analog signals

14. What is the purpose of (DAC)?
a) To convert analog signals into digital signals
b) To convert digital signals into analog signals
c) To process digital signals d) To store analog signals

15. What is digital logic based on?
a) Decimal numbers b) Binary numbers
c) Hexadecimal numbers d) Octal numbers

16. How are binary values represented in digital circuits?
a) By different frequencies b) By different voltage levels
c) By different currents d) By different temperatures

17. What is the purpose of the NOT operation?
a) To perform addition b) To perform subtraction
c) To negate the value of a binary variable d) To perform multiplication

18. What do logic diagrams primarily depict?
a) Physical layout of circuits
b) Working of a digital circuit through symbols
c) Function of power supplies
d) Configuration of mechanical components

19. What is the first step in creating a logic diagram?
a) Drawing the power supply b) Finding the logic gates needed
c) Testing the circuit d) Writing code for the circuit

20. **Which of the following is NOT necessary for designing a logic diagram?**
a) Boolean algebra knowledge b) Understanding logic gates
c) Knowledge of mechanical components d) Understanding circuit functions

21. **Why is Boolean algebra essential in digital circuit design?**
a) It connects physical components.
b) It defines the operations of the circuit.
c) It simplifies hardware design. d) Both b and c

22. **What ensures the correct operation of a logic circuit?**
a) Accurate physical connections
b) Correct Boolean function representation
c) Proper arrangement and connection of gates d) All of the above

23. **What do logic diagrams depict?**
a) The working of an analog circuit b) The working of a digital circuit
c) The working of a mechanical system
d) The working of an electrical system

24. **Why is knowledge of Boolean algebra crucial?**
a) To create analog circuits b) To create digital circuits
c) To study mechanical systems d) To study electrical systems

25. **What do logic gates represent in a logic diagram?**
a) Individual digital circuits b) Individual logic gates
c) Individual mechanical systems d) Individual electrical systems

26. **What is the primary function of adder circuits in digital circuits?**
a) To perform logical operations
b) To perform arithmetic calculations
c) To store data d) To transmit data

27. **Which of the following is a characteristic of a half-adder?**
a) Adds three single-bit binary numbers
b) Adds two single-bit binary digits
c) Performs subtraction d) Performs multiplication

28. **What is the purpose of a Karnaugh map (K-map)?**
a) To design digital circuits
b) To simplify Boolean expressions
c) To perform arithmetic calculations d) To store data

29. **How many variables can a 4x4 K-map represent?**
a) 2 variables b) 3 variables c) 4 variables d) 5 variables

30. **What are the outputs of a half-adder circuit?**
a) Sum and Product b) Sum and Carry
c) Carry and Difference d) Product and Difference

31. What is the Boolean expression for the sum output of a full-adder?

a) A•B+Cin
b) (A•B) + (Cin • (A⊕B))
c) A+B•A + B•A+B
d) A•B•Cin

32. How many variables does a 4x4 K-map represent?

a) 2
b) 3
c) 4
d) 5

33. What is the primary purpose of Karnaugh maps?

a) To construct truth tables
b) To minimize Boolean expressions
c) To test circuits
d) To perform arithmetic calculations

34. In the K-map method, what does each cell represent?

a) A logic gate
b) A minterm
c) A binary operation
d) A circuit diagram

35. Which condition makes the sum output of a half-adder high?

a) Both inputs are low
b) Both inputs are high
c) Only one input is high
d) Both inputs are equal

36. Which of the following expressions represents the carry output of a full-adder?

a) (A•B) + (Cin • (A⊕B))
b) A⊕B⊕Cin
c) A•Cin
d) A+B+Cin

37. What is the simplified Boolean expression for A·B+A·$\overline{B}$?

a) A
b) B
c) A•B
d) 1

Answers:

1	2	3	4	5	6	7	8	9	10	11	12
B	C	B	B	B	B	A	B	B	C	B	B
13	14	15	16	17	18	19	20	21	22	23	24
B	B	B	B	C	B	B	C	D	D	B	B
25	26	27	28	29	30	31	32	33	34	35	36
B	B	B	B	C	B	B	C	B	B	C	A
37											
A											

Topic Wise Additional Short Questions and Answers

3.1- Basics of Digital Systems:

1: What is an Analog Signal? Give examples.

Ans: Analog signals are signals that change 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: What is a digital signal?

Ans: 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, such as calculators and computers.

3: Why are ADC and DAC important?
Ans: ADC and DAC enable the conversion between Analog and Digital signals, making it possible to process, store, and transmit signals effectively.
4: What does ADC and DAC stand for?
Ans: ADC stands for Analog to Digital Converter.
DAC stands for Digital to Analog Converter.
5: What is ADC? Give an example of ADC in use.
Ans: 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 phones. A microphone converts sound waves (analog) into digital signals using an ADC.
6: What is DAC? Give an example of DAC in use.
Ans: DAC is the conversion whereby analog signals are converted to digital signals, making it possible for human to perceive the information, for instance, a speaker converts digital signals back into sound waves (analog) using a DAC.
7: Why are digital signals preferred for long-distance transmission?
Ans: Digital signals are less affected by noise and degradation, ensuring higher quality over long distances.
8: What is digital logic?
Ans: Digital logic is the basis of digital systems, using binary numbers (0 and 1) to represent and manipulate information.
9: What do digital logic circuits do?
Ans: They perform operations using binary values (0 and 1) and are essential to computers and electronic devices.
10: What are logic levels? Why are logic levels important?
Ans:

- Voltage levels that represent binary 0 and 1 in digital circuits.
- They switch devices on and off and define how digital circuits execute operations and process information.

3.2- Boolean Algebra and Logic Gates

11: What is Boolean algebra?
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.
12: What is AND operation?
Ans: 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 symbol '.' is used for the AND operation. The output of the AND operation is "1" only when both inputs are "1". Otherwise, the result is "0".
13. What is OR operation?
Ans: OR is the basic logical operator which is used in Boolean algebra. It requires two binary inputs which will give a single binary output. The symbol

'+' is used for the OR operation. The OR operation gives true (1) output wh[...] at least one of the inputs is true (1). The output is 0 only when both inputs a[...] '0'.

14. What is NOT operation?

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

15: Make a truth table of AND.

Ans:

X	Y	$X.Y$
0	0	0
0	1	0
1	0	0
1	1	1

16. Make a truth table of OR.

Ans:

X	Y	$X+Y$
0	0	0
0	1	1
1	0	1
1	1	1

17. Make a truth table of NOT.

Ans:

X	$\overline{Y}$
0	1
1	0

18. Give some examples of usage of Boolean functions in computers.

Ans: Here are some examples of their usage:

- Arithmetic Operations: Boolean functions are used in Arithmetical Logic Un[...] of CPUs to perform operations like addition, subtraction, multiplication, an[...] division.
- Data Processing: Boolean functions are used to process binary data in memo[...] and storage devices, ensuring efficient data manipulation and retrieval.
- Control Logic: Boolean functions are applied in computers to control variou[...] parts of the system's operation to function in co-ordinated manner.

19. What is the use of Boolean function in our everyday devices?

Ans: Boolean functions are also present in our everyday devices like cell phon[...] and calculators:

- Cell Phones: In cell phone processing, when you dial a number, or press a button on a phone, a Boolean function evaluates these inputs as true or false and makes the necessary output.
- Calculators: Basic calculators use Boolean functions. When you feed it with numbers and the operations to be performed, Boolean logic is used to arrive at the right result.

20. Who was George Boole?

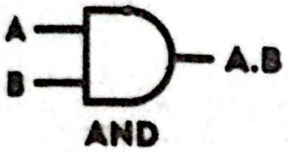
Ans: George Boole, a mathematician who invented Boolean algebra was born in Lincoln, England in the year 1815. His work laid the debate and the basis for future digital revolution and computer science as well as subsequent technologies of the future.

21. What are logic gates?

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

22. What is AND gate?

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



2 input AND Gate

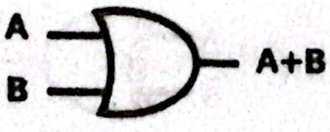
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

In 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 switches are ON.
- If either switch is OFF, the light bulb will be OFF.

23. What is OR gate?

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



2 input OR gate

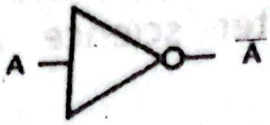
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

In a simple electronic circuit with an OR gate. If you press two switches (one is ON and other is OFF), a light bulb will turn on.

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

24. What is NOT gate?

Ans: Implements the NOT function. It outputs the opposite of the input.



2 input NOT gate

A	Ā
0	1
1	0

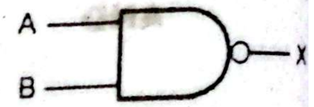
- Switch 1: ON (True) Then • Light bulb: OFF (False)
- Switch 2: OFF (False) Then • Light bulb: ON (True)

25. What is NAND gate?

Ans: This gate is achieved when an AND gate is combined with a NOT gate. In other words, it is t generates true when at least one of the inputs is false. In other words, it is the inverse of the AND 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.

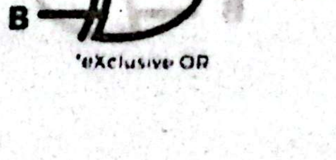
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

26. What is XOR gate?

Ans: 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.

Example:

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



2 input XOR gate

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Play video games: Yes (True)

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

3.3- Simplification of Boolean Functions

27. Why is simplification of Boolean functions important?

Ans: It helps in designing efficient digital circuits that require fewer gates, are compact, energy-efficient, and faster. Applying Boolean algebra rules to make the functions less complicated.

28. What does the Identity Law state?

Ans: $A + 0 = A$

$A \cdot 1 = A$

29. What are the Null Laws?

Ans: $A + 1 = 1$

$A \cdot 0 = 0$

30. What is the Idempotent Law?

Ans: $A + A = A$

$A \cdot A = A$

31. What are the Complement Laws?

Ans: $A + \bar{A} = 1$

$A \cdot \bar{A} = 0$

32. What does the Commutative Law state?

Ans: $A+B= B+A$

$A \cdot B= B \cdot A$

33. What are the Associative Laws?

Ans: $(A+B)+C=A+(B+C)$

$(A \cdot B) \cdot C=A \cdot (B \cdot C)$

34. What does the Distributive Law state?

Ans: $A \cdot (B+C)=(A \cdot B)+(A \cdot C)$

$A+(B \cdot C)=(A+B) \cdot (A+C)$

35. What are the Absorption Laws?

Ans: $A+(A \cdot B)=A$

$A \cdot (A+B)=A$.

36. State De Morgan's Theorems.

Ans: $\overline{A + B} = \bar{A} \bullet \bar{B}$

$\overline{A \bullet B} = \bar{A} + \bar{B}$

37. What is the Double Negation Law?

Ans: $\bar{\bar{A}} = A$

38. Simplify the expression $A + \bar{A} \bullet B$.

Ans:

$A + \bar{A} \bullet B$

$= (A + \bar{A}) \bullet (A + B)$ (Distributive Law)

$= 1 \bullet (A + B)$ (Complement Law)

$= A + B$ (Identity Law)

39. Simplify the expression $\overline{A \cdot B} + \bar{A} \cdot B$.

Ans:

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

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

$= (\bar{A} \cdot \bar{B})$

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

(De Morgan's Theorem) Since $\bar{A}$ is already present in $(\bar{A} \cdot \bar{B})$, we can use absorption law i.e. $\bar{A} + (\bar{A} \cdot B) = \bar{A}$

40. Simplify the expression $(A + B) \cdot (A + \bar{B})$

Ans:

$(A + B) \cdot (A + \bar{B})$

$= A \cdot (A+\bar{B}) + B \cdot (A+\bar{B})$ (Distributive Law)

$= A + A \cdot \bar{B} + B \cdot \bar{B}$ (Absorption Law)

$= A + A \cdot B$ (Identity Law)

$= A \cdot (1 + B)$ (Distributive Law)

$= A \cdot 1$ (Null Law)

$= A$ (Identity Law)

41. Simplify the expression $\overline{\bar{A} + B} \cdot (A + \bar{B})$

Ans:

$\overline{\bar{A} + B} \cdot (A + \bar{B})$

$= (A \cdot \bar{B}) \cdot (A + \bar{B})$ (De Morgan's Theorem)

$= A \cdot \bar{B} \cdot A + A \cdot \bar{B} \cdot \bar{B}$ (Distributive Law)

$= A \cdot \bar{B} + A \cdot \bar{B}$ (Idempotent Law)

$= A \cdot \bar{B}$ (Identity Law)

3.4- Creating Logic Diagrams

42. What do logic diagrams represent?

Ans: Logic diagrams represent the working of a digital circuit through symbols for individual logic gates.

43. What is the first step in creating a logic diagram?

Ans: Find out the logic gates needed for the Boolean function.

44. Why is knowledge of Boolean algebra important in digital circuit design?

Ans: It helps create and study digital circuits effectively.

45. What is required to connect the gates correctly in a circuit?

Ans: Inputs and outputs of the gates must be connected as defined by the circuit's function.

46. What two concepts are crucial for creating efficient digital systems?

Ans: Boolean algebra and logic gates.

3.5. Application of Digital Logic

47. What is digital logic used for in modern electronic systems?

Ans: Digital logic is used for designing circuits in devices like computers, smartphones, and digital gadgets.

48. What are two important applications of digital logic?

Ans: Design of adder circuits and use of Karnaugh maps for function simplification.

49. Define a half-adder circuit?

Ans: 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).

50. Make a truth table for Half Adder circuit.

Ans:

A	B	SUM= A⊕B	CARRY= A•B
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

51. Define a Full Adder circuit?

Ans: A full-adder is a complex circuit that adds three single-bit binary numbers: two bits that belong to the 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.

52. Write the Boolean expressions for the sum and carry in a half-adder circuit.

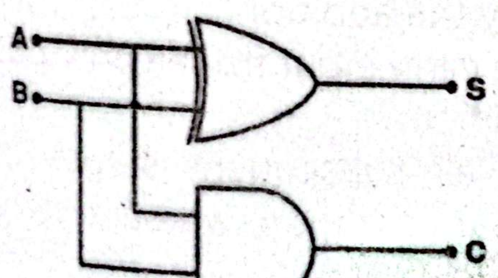
Ans: Sum (S) = A⊕B,
Carry (C) = A•B

53. Write the Boolean expressions for the sum and carry in a full-adder circuit.

Ans: Sum = A ⊕ B ⊕ Cin
Carry= (A•B) + (Cin • (A⊕B))

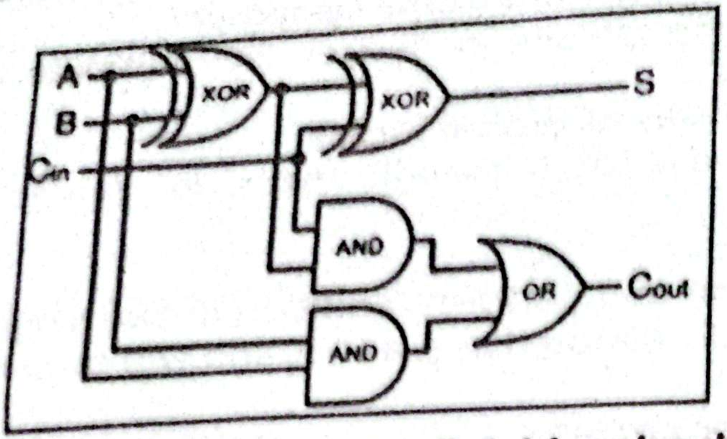
54. Draw circuit diagram of Half Adder.

Ans:



55. Draw a circuit diagram of full adder.

Ans:



56. Make a truth table of Full Adder circuit.

Ans:

A	B	C	SUM= A⊕B⊕Cin	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

57. What is the primary difference between a half-adder and a full-adder?

Ans:

Half Adder	Full Adder
• Adds two binary digits. • Two inputs (A and B). • Two outputs: • **Sum (S)**: Represents the sum of the two inputs. • **Carry (C)**: Represents the carry generated by the addition. • Cannot handle a carry input from a previous stage of addition. • Typically used for the least significant bit in binary addition, where no carry input exists.	• Adds three binary digits • Three inputs (A, B, and Carry-In). • Two outputs: • **Sum (S)**: Represents the sum of the three inputs. • **Carry (C)**: Represents the carry generated by the addition. • Can handle carry input from a previous stage of addition. • Used for all subsequent stages in multi-bit binary addition, where a carry input may exist.

58. What are adder circuits? Write down their types.

Ans: 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

59. What is a Karnaugh map (K-map) used for?

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.

60. Write down the structure of K. Map.

Ans: A K-map is a matrix where each square is a cell, which corresponds to a positioned combination. These cells are filled with '1' or '0' in reference to the truth table of the Boolean function.

61. How can we determine the size of the K. Map?

Ans: The size of the K-map depends on the number of variables:

- 2 Variables: 2x2 grid
- 3 Variables: 2x4 grid
- 4 Variables: 4x4 grid least
- 5 Variables: 4x8 grid (less common for manual simplification)

62. How does grouping work in K-maps?

Ans: Groups of 1s are formed in sizes of powers of two (1, 2, 4, 8, etc.), and they must be in continuous rows or columns.

63. What is a minterm in Boolean algebra? OR 5. How are minterms represented in Boolean algebra?

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

64. Write down the minterm where A = 1, B = 0 and C = 1 is written as A B C.

Ans: The minterm where A = 1, B = 0 and C = 1 is written as $A \bar{B} C$

65. Write down the steps to create the K. Map.

Ans: To create a K-map, follow these steps:

1. Create a grid based on the number of variables that exists in the system.
2. Let us complete the grid using the output values in the truth table.
3. Arrange the 1s in the grid in the largest possible groups of size 1, 2, 4, 8 and so on. Every group must have one or more 1s, must be a power of two, and they must be in a continuous rows or columns.

66. Write down the practical usage of K. Map.

Ans: Karnaugh maps are extensively used in digital circuit design to minimize the number of gates needed for a given function. This leads to circuits that are faster, cheaper, and consume less power.