Class: 9th

Subject: Computer

Unit 3: Digital System and Logic Design

Multiple Choice questions (MCQs)

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

(a) AB

(**b**) A + B ✅

(c) A

(d) AB

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

Important MCQs:

1. Which type of signal changes continuously over time?

(a) Digital

(b) Binary

(c) Analog ✅

(d) Logical

2. Which signal has only two discrete values: 0 and 1?

(a) Voltage

(b) Digital ✅

(c) Analog

(d) Pulse

3. What is the function of an Analog to Digital Converter (ADC)?

(a) Converts text into images

(b) Converts digital signals into analog

(**c**) Converts analog signals into digital ✅

(d) Amplifies analog signals

4. Digital signals are preferred for communication because they:

(a) Carry more energy

(b) Are less affected by noise ✅

(c) Are cheaper to produce

(d) Need no conversion

5. Which one of the following is an example of an analog signal?

(a) Data in a USB

(**b**) Sound waves ✅

(c) Binary code

(d) Light pulses in fiber optics

6. What does DAC stand for?

(a) Digital Amplifier Circuit

(**b**) Digital to Analog Converter ✅

(c) Data Access Code

(d) Direct Audio Control

7. In the communication system, which device is used at the receiving end to convert digital signals into analog signals?

(a) Microphone

(b) Amplifier

(**c**) Speaker ✅

(d) Transistor

8. What type of signal is produced when a person speaks into a microphone?

(a) Digital

(**b**) Analog ✅

(c) Binary

(d) Pulse

9. What happens to the sound wave when it passes through a microphone?

(a) It is amplified

(b) It is stored directly

(**c**) It is converted into binary data ✅

(d) It becomes louder

10. Why is conversion between analog and digital signals necessary?

(a) For designing circuits

(b) For faster typing

(**c**) For signal transmission and human understanding ✅

(d) For software programming

11. What voltage level usually represents logic '1' in digital circuits?

(a) 0 volts

(b) 3 volts

(**c**) 5 volts ✅

(d) 12 volts

12. Which logic gate produces output '1' only when both inputs are '1'?

(a) OR gate

(**b**) AND gate ✅

(c) NOT gate

(d) NAND gate

13. The output of an OR gate is '0' only when:

(a) Both inputs are 1

(b) One input is 1

(**c**) Both inputs are 0 ✅

(d) One input is 0

14. Boolean algebra uses which two logical values?

(a) High and Low

(**b**) True and False ✅

(c) On and Off

(d) 0 and 5 volts

15. In Boolean algebra, which symbol represents the OR operation?

(**a**) + ✅

(b) .

(c) ~

(d) &

16. What will be the output of AND operation if inputs are A=0 and B=1?

(a) 1

(**b**) 0 ✅

(c) Undefined

(d) Error

17. Which of the following is NOT a basic logic operation?

(a) AND

(b) OR

(c) NOT

(**d**) ADDITION ✅

18. The NOT operation in Boolean algebra:

(a) Combines two inputs

(**b**) Inverts the input value ✅

(c) Adds two inputs

(d) Outputs the same input

19. In digital systems, logic levels are used to:

(a) Represent numbers in decimal

(**b**) Control devices by switching ON and OFF ✅

(c) Measure temperature

(d) Amplify signals

20. What is the truth table used for in digital logic?

(a) To add binary numbers

(**b**) To represent all possible input-output combinations of a logic gate ✅

(c) To convert analog to digital

(d) To amplify signals

21. What does the NOT operation do to a binary input?

(a) Leaves it unchanged

(**b**) Converts 0 to 1 and 1 to 0 ✅

(c) Converts 0 to 0 and 1 to 1

(d) Converts 1 to 2

22. If A = 1, what will be the output of NOT A?

(a) 1

(**b**) 0 ✅

(c) 2

(d) Undefined

23. Which of the following is the correct truth table for the NOT operation?

(a) Input: 0 ⇒ Output: 0, Input: 1 ⇒ Output: 1

(b) Input: 0 ⇒ Output: 1, Input: 1 ⇒ Output: 0 ✅

(c) Input: 0 ⇒ Output: 1, Input: 1 ⇒ Output: 1

(d) Input: 0 ⇒ Output: 0, Input: 1 ⇒ Output: 0

24. Boolean functions are constructed using which basic logical operations?

(**a**) AND, OR, NOT ✅

(b) ADD, SUBTRACT, MULTIPLY

(c) AND, XOR, NAND

(d) ADD, DIVIDE, MOD

25. In digital logic design, Boolean functions are important because they:

(a) Represent decimal numbers

(**b**) Control how digital circuits operate and process data ✅

(c) Only work with analog signals

(d) Are used to increase voltage

26. Which logic gate outputs true only when both inputs are true?

(a) OR

(**b**) AND ✅

(c) NAND

(d) XOR

27. The OR gate outputs true when:

(a) Both inputs are false

(**b**) At least one input is true ✅

(c) Both inputs are true only

(d) None of the above

28. What is the output of a NOT gate if the input is 0?

(a) 0

(**b**) 1 ✅

(c) Cannot determine

(d) 2

29. NAND gate is the combination of which gates?

(a) AND and OR

(**b**) AND and NOT ✅

(c) OR and NOT

(d) XOR and NOT

30. When does an XOR gate output true?

(a) When both inputs are true

(**b**) When exactly one input is true ✅

(c) When both inputs are false

(d) When no input is true

31. In a safety alarm system using a NAND gate, the alarm goes ON when:

(a) Both sensors detect no issue

(**b**) At least one sensor detects an issue ✅

(c) Both sensors detect an issue

(d) None of the sensors detect an issue

32. The Boolean expression A + 0 equals:

(a) 0

(**b**) A ✅

(c) 1

(d) A + 1

33. The Boolean expression A * 1 equals:

(a) 0

(**b**) A ✅

(c) 1

(d) A + 1

34. According to the Null Law, what is the value of A * 0?

(a) A

(b) 1

(**c**) 0 ✅

(d) Cannot be determined

35. What does the Complement Law state about A + Ā?

(a) 0

(**b**) 1 ✅

(c) A

(d) Ā

36. The Commutative Law states that:

(**a**) A + B = B + A and A * B = B * A ✅

(b) A + B = A * B

(c) A + B = A

(d) None of these

37. Which law explains that (A + B) + C = A + (B + C)?

(a) Commutative Law

(**b**) Associative Law ✅

(c) Distributive Law

(d) Identity Law

38. Distributive Law can be written as:

(**a**) A + (B * C) = (A + B) * (A + C) ✅

(b) A * (B + C) = (A + B) + (A + C)

(c) A + B = B + A

(d) A * B = B * A

39. The Absorption Law states that:

(a) A + (A * B) = A ✅

(b) A * (A + B) = A + B

(c) A + 0 = A

(d) A * 1 = A

40. De Morgan's Theorem states that:

(a) (A + B)' = A' + B'

(**b**) (A + B)' = A' * B' ✅

(c) (A * B)' = A + B

(d) (A * B)' = A' + B

41. Logic diagrams use ________ to represent digital circuits.

(a) Variables

(**b**) Symbols of logic gates ✅

(c) Numbers

(d) Equations

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

(a) Connect inputs and outputs

(b) Arrange gates

(**c**) Find out the logic gates needed for the Boolean function ✅

(d) Test the circuit

43. Which of the following is an application of digital logic?

(**a**) Designing adder circuits ✅

(b) Using resistors in circuits

(c) Programming languages

(d) Electrical wiring

44. A half-adder has how many inputs and outputs?

(a) 1 input, 1 output

(**b**) 2 inputs, 2 outputs ✅

(c) 3 inputs, 1 output

(d) 2 inputs, 1 output

45. The sum output (S) of a half-adder is given by which Boolean expression?

(a) A.B

(b) A + B

(**c**) A XOR B ✅

(d) A NAND B

46. What is the carry output (C) expression for a half-adder?

(a) A + B

(**b**) A.B ✅

(c) A XOR B

(d) A NAND B

47. A full-adder circuit adds how many bits?

(a) 2 bits

(b) 3 bits ✅

(c) 1 bit

(d) 4 bits

48. In a Karnaugh map, what does each cell represent?

(a) Variable

(b) Minterm ✅

(c) Gate

(d) Output only

49. Which technique is used in Karnaugh maps to simplify Boolean expressions?

(a) Algebraic multiplication

(b) Grouping adjacent 1s to combine terms ✅

(c) Adding 0s

(d) Using NOT gates

50. When creating groups of 1s in a Karnaugh map, which of the following is true?

(a) Groups must contain an odd number of 1s.

(b) Groups can be formed from any scattered 1s.

(c) Groups must be in powers of 2 (such as 1, 2, 4, 8) and be adjacent. ✅

(d) Groups must include all 1s on the map, even if not adjacent.

51. What is the correct sequence to create a Karnaugh map?

(a) Draw the grid, connect inputs, then arrange the outputs.

(b) Write Boolean expression, create the grid, then fill the grid with outputs.

(c) Create the grid, fill it using output values from the truth table, then group the 1s to simplify. ✅

(d) Group 1s first, then create the grid and fill it later.

Exercise Short Questions:

1. Define a Boolean function and give an example.

Answer:

A Boolean function is a mathematical function that takes one or more binary inputs (0 or 1) and produces a binary output (0 or 1) based on logical operations like AND, OR, and NOT.

Example:

F(A, B) = A + B (This is an OR operation where output is 1 if either A or B is 1).

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

Answer:

A truth table lists all possible combinations of inputs for a digital circuit and shows the corresponding output for each combination. It is important because it helps to understand, analyze, and design digital circuits effectively by representing the behavior of logic gates or functions clearly.

3. Explain the difference between analog and digital signals.

Answer:

- **Analog signals** vary continuously over time and can take any value within a range (e.g., sound waves).
- **Digital signals** have discrete values, typically two levels (0 and 1), representing binary data used in digital circuits.

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

Answer:

A NOT gate is a logic gate that inverts its input. If the input is 0, the output is 1; if the input is 1, the output is 0.

Input (A)	Output (A')
0	1
1	0

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

Answer:

A Karnaugh map (K-map) is a graphical tool used to simplify Boolean expressions by organizing truth table values into a matrix. It helps to identify groups of 1s that can be combined to reduce the number of terms and variables, leading to simpler and more efficient digital circuit designs.

Important short questions:

1. What is a digital system?

A digital system is an electronic system that processes information in the form of binary digits (0 and 1). It is the foundation of modern electronics and computing devices like computers and calculators.

2. Define an analog signal with an example.

An analog signal is a continuous signal that varies smoothly over time and can take any value within a given range.

Example: Sound waves or the temperature of the human body.

3. What is a digital signal? Give an example.

A digital signal is a discrete signal that has only two possible values, usually represented as 0 and 1. These signals are used in digital electronics and computing.

Example: Binary data in computers.

4. How does an analog signal differ from a digital signal?

- Analog signals are continuous and can have infinite values.
- Digital signals are discrete and have only two values (0 or 1).
- Analog signals represent real-world information smoothly, while digital signals represent data in binary form.

5. What is the function of an Analog to Digital Converter (ADC)?

An ADC converts analog signals into digital signals. This conversion allows analog information (like sound) to be processed, stored, and transmitted by digital devices such as computers and smartphones.

6. Explain the role of a Digital to Analog Converter (DAC).

A DAC converts digital signals back into analog signals. This allows digital information to be perceived by humans in a natural form, such as sound through speakers.

7. Why is the conversion between analog and digital signals important?

The conversion is important because digital signals are less affected by noise and degradation, making them better for storing and transmitting information over long distances. Analog to digital conversion enables digital processing, while digital to analog conversion enables human perception of data.

8. Give an example of where ADC and DAC are used in daily life.

When you speak into a microphone, the sound waves (analog) are converted into digital signals by an ADC for transmission. At the receiver's end, these digital signals are converted back to analog by a DAC, allowing the sound to be played through speakers.

9. What are the advantages of digital signals over analog signals?

- Less affected by noise and distortion
- Easier to store and transmit over long distances
- More reliable and accurate for digital processing
- Can be easily compressed and encrypted

10. Describe the process of sound transmission using ADC and DAC with an example.

When a person speaks into a microphone, their voice produces analog sound waves. The ADC in the microphone converts these sound waves into digital signals. These digital signals are transmitted over networks with minimal quality loss. At the receiving end, a DAC converts the digital signals back to analog sound waves, which are played through speakers, allowing the listener to hear the voice clearly.

11. What is meant by digital logic?

Answer:

Digital logic is the use of binary values (0 and 1) to represent and manipulate information in digital systems. It forms the basis of digital circuits and computing devices.

12. How are binary values represented in digital circuits?

Answer:

Binary values are represented by different voltage levels, where a high voltage (e.g., 5V) represents binary '1' and a low voltage (e.g., 0V) represents binary '0'.

13. Define logic levels in digital systems.

Answer:

Logic levels are specific voltage ranges that represent binary values ‘0’ and ‘1’ in digital circuits, allowing devices to switch on or off and perform operations.

14. What is Boolean algebra?

Answer:

Boolean algebra is a branch of mathematics that deals with variables having two values, True and False, and the operations on these values used for designing and analyzing digital circuits.

15. Name the three basic logic operations in Boolean algebra.

Answer:

The three basic logic operations are AND, OR, and NOT.

16. Explain the AND operation with an example.

Answer:

AND operation requires two inputs and produces an output of ‘1’ only when both inputs are ‘1’.

Example: If A = 1 and B = 0, then A AND B = 0.

17. What does the OR operation do?

Answer:

OR operation produces an output of ‘1’ if at least one of the inputs is ‘1’. The output is ‘0’ only when all inputs are ‘0’.

18. Describe the NOT operation and its effect on a binary variable.

Answer:

NOT operation takes a single input and inverts it: if the input is ‘1’, output is ‘0’; if input is ‘0’, output is ‘1’.

19. What is a truth table?

Answer:

A truth table lists all possible input combinations for a logic operation and the corresponding output values, demonstrating how the operation works.

20. Why are truth tables important in digital logic design?

Answer:

Truth tables help in understanding, analyzing, and verifying the behavior of logic circuits by showing outputs for all input cases.

21. What is a Boolean function?

Answer:

A Boolean function is an algebraic statement that relates one or more binary inputs to a single binary output using logical operations.

22. How many values can inputs and outputs of a Boolean function have?

Inputs and outputs of a Boolean function can have only two values: 0 (False) or 1 (True).

23. Which basic logical operations are used to construct Boolean functions?

The basic logical operations used are AND, OR, and NOT.

24. Write the Boolean function for the AND operation with inputs A and B.

Answer:

The Boolean function for AND operation is $F(A, B) = A \cdot B$.

25. What will be the output of the Boolean function F(A, B) = A . B when both A and B are 0?

The output will be 0 because AND operation outputs 1 only when both inputs are 1.

26. How is the output determined in the Boolean function ?

F(A, B, C) = A . B + A .C?

The output is the OR of the AND of A and B, and the AND of A and C.

27. In the Boolean function F(A, B, C) = A . B + A .C , what operation does the '+' symbol represent?

The '+' symbol represents the OR operation.

28. What does the term 'truth table' mean in the context of Boolean functions?

A truth table lists all possible input combinations and their corresponding outputs for a Boolean function.

29. Name two uses of Boolean functions in computer systems.

Arithmetic operations in the Arithmetic Logic Unit (ALU).

Data processing and control logic in computer systems.

30. Why are Boolean functions important in the design of digital circuits?

Boolean functions are important because they allow the design and optimization of digital circuits that control and process binary data.

31. What is the function of an AND gate in digital circuits?

An AND gate performs the logical AND operation. It outputs true (1) only when both of its inputs are true (1). If either input is false (0), the output is false (0).

32. How does an OR gate differ from an AND gate in terms of output?

An OR gate outputs true (1) if at least one of its inputs is true (1). In contrast, an AND gate outputs true only when both inputs are true. Thus, OR gate is less strict for producing a true output than AND gate.

33. What is the output of a NOT gate when the input is 0?

A NOT gate outputs the opposite of its input. Therefore, if the input is 0, the output will be 1.

34. Explain the working of a NAND gate with respect to the AND gate.

A **NAND gate** is the inverse of an **AND gate**. It first performs an **AND operation** on its inputs, then outputs the opposite (NOT) of that result. So, a **NAND gate** outputs false (0) only when both inputs are true (1); otherwise, it outputs true (1).

35. When does an XOR gate output true, and how is it different from an OR gate?

Answer:

An **XOR** (Exclusive OR) **gate** outputs true (1) only when exactly one of the inputs is true (1). If both inputs are true or both are false, the output is false (0). This differs from an **OR gate**, which outputs true if at least one input is true, including when both are true.

36. What is the importance of simplifying Boolean functions in digital circuit design?

Answer:

Simplifying Boolean functions reduces the complexity of digital circuits. This leads to circuits with fewer logic gates, which makes them smaller in size, faster in operation, and more energy efficient. Simplified circuits are easier to design and cost-effective to produce.

37. State the Identity Laws in Boolean algebra.

Answer:

The Identity Laws state:

$A + 0 = A$

$A \cdot 1 = A$

These laws mean adding zero or multiplying by one leaves the variable unchanged.

38. Explain the Complement Laws with examples.

Answer:

Complement Laws state that:

$A + \bar{A} = 1$ (A OR NOT A equals 1)

$A \cdot \bar{A} = 0$ (A AND NOT A equals 0)

For example, if $A = 1$, then $\bar{A} = 0$, so $A + \bar{A} = 1 + 0 = 1$.

39. What is the purpose of De Morgan's Theorems in Boolean simplification?

Answer:

De Morgan's Theorems help transform expressions with AND and OR operations into equivalent expressions with OR and AND operations respectively, using NOT operations. This is useful for simplifying and implementing digital circuits with fewer gates.

40. Write the Distributive Laws of Boolean algebra.

Answer:

The Distributive Laws are:

A · (B + C) = (A · B) + (A · C)

A + (B · C) = (A + B) · (A + C)

41. Describe the process of creating a logic diagram from a Boolean function.

Answer:

To create a logic diagram:

- Identify the required logic gates from the Boolean function.
- Arrange these gates in a sequence that matches the operations in the function.
- Connect inputs and outputs correctly to show the flow of logic.

42. What are the key differences between a half-adder and a full-adder circuit?

Answer:

- Half-adder: Adds two single-bit inputs and produces sum and carry outputs.
- Full-adder: Adds three inputs (two bits plus carry from previous addition) and produces

sum and carry outputs.

43. Give the Boolean expressions for the sum and carry outputs of a half-adder.

Answer:

Sum (S) = $A \oplus B$ (A XOR B)

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

44. Explain the Associative Laws of Boolean algebra.

Answer:

Associative Laws allow grouping variables differently without changing the result:

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

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

45. What are the two main outputs of a half-adder, and what do they represent?

Answer:

- **Sum** (S): Represents the binary sum of the two inputs without considering carry.
- **Carry** (C): Represents the carry generated when

both inputs are 1, used in multi-bit addition.

Exercise Long Questions:

🌟 **Q1. Explain the usage of Boolean functions in computers.**

Answer:

Boolean functions form the foundation of digital electronics and computer systems. They use logical operations on binary variables (0 and 1) to perform various computational tasks.

- **Arithmetic Operations**: Boolean logic enables computers to perform binary addition, subtraction, and other calculations inside the Arithmetic Logic Unit (ALU).
- **Data Processing**: All data in computers is represented in binary. Boolean functions help manipulate this binary data efficiently, allowing tasks like searching, sorting, and decision making.
- **Control Systems**: Boolean expressions control how different components of a computer

operate, like enabling/disabling circuits or routing data paths.

- **Circuit Design**: Logic gates that implement Boolean functions are the building blocks of complex circuits like multiplexers, memory units, and processors.
- **Decision Making:** In programming, conditions and loops depend on Boolean logic to decide which instructions to execute.

Thus, Boolean functions allow computers to process information logically and perform complex tasks with simple binary inputs.

🌟 Q2. Describe how to construct a truth table for a Boolean expression with an example.

Answer:

A truth table systematically lists all possible input combinations of variables and their corresponding output values for a Boolean function.

Steps to construct:

1. **Determine variables**: Count the number of

variables in the expression (say n).

2. **List inputs**: Write all possible 2^n input combinations in binary.

3. **Evaluate outputs**: Calculate the function's output for each input using Boolean operations.

- **Example**: Construct truth table for

F =(A .B) + (A . C).

A	B	C	A·B	A·C	F = (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

This truth table clearly shows the output for every input combination.

🌟 Q3. Describe the concept of duality in Boolean algebra and provide an example to illustrate it.

Answer:

The duality principle in Boolean algebra states that every Boolean expression remains true if we interchange:

Answer:

- The duality principle in Boolean algebra states that every algebraic expression remains valid if we interchange the AND and OR operators and interchange the constants 0 and 1. This means each Boolean identity has a dual identity.
- This principle helps to generate new Boolean identities from existing ones by simply swapping AND ″ OR and 0 ″ 1.

Example:

- Original expression:
  $A + 0 = A$
- Dual expression (swap + with · and 0 with 1):
  $A \cdot 1 = A$

Both expressions are true and show the duality in Boolean algebra.

Another example:

- Original:
  $A + \overline{A} = 1$
- Dual:
  $A \cdot \overline{A} = 0$

Both are fundamental Boolean identities, demonstrating duality.

🌟 **Q4. Compare and contrast half-adders and full-adders, including their truth tables, Boolean expressions, and circuit diagrams.**

Answer:

Half-Adder:

Purpose: Adds two single-bit binary numbers (A and B).

- **Inputs**: Two bits (A, B).
- **Outputs**: Sum (S) and Carry (C).

- **Boolean expressions:**
  - Sum, $S = A \oplus B$ (XOR operation)
  - Carry, $C = A \cdot B$ (AND operation)

Truth table:

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

- **Circuit Diagram:**
  - One XOR gate for sum.
  - One AND gate for carry.

Full-Adder:

- **Purpose**: Adds three single-bit binary numbers: two bits (A, B) and a carry-in (Cin) from a previous addition.

- Inputs: Three bits (A, B, Cin).
- Outputs: Sum (S) and Carry-out (Cout).

Boolean expressions:

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

Truth table:

A	B	Cin	Sum (S)	Carry-out (Cout)
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

Circuit Diagram:

- Two XOR gates (for sum).
- Three AND gates and one OR gate (for

carry-out).

Summary:

Half-adders can only add two bits and do not consider carry-in, so they are used in the least significant bit addition. **Full-adders** add two bits along with a carry-in and are used in multi-bit binary addition chains.

🌟**Q5. How do Karnaugh maps simplify Boolean expressions? Provide a detailed example with steps.**

Answer:

Karnaugh Map (K-map) is a graphical tool used to simplify Boolean expressions by organizing truth table values into a grid format. It helps visualize and eliminate redundant terms in expressions, leading to minimal forms.

Steps to Simplify using K-map:

1. **Draw the K-map**: For n variables, draw a 2^n cell grid.

2. **Fill the K-map:** Mark cells with 1 for minterms

where the function is true.

3. **Group 1's:** Group adjacent 1's in sizes of powers of two (1, 2, 4, 8, ...).

4. **Write simplified expression**: For each group, write product terms where variables remain constant.

5. **Combine all terms**: The sum of these product terms is the simplified Boolean expression.

- **Example**:

Simplify:
$F(A, B, C) = \sum m(1, 3, 5, 7)$
(Minterms where function is 1)

A B C	F
000	0
001	1
010	0
011	1
100	0
101	1
110	0
111	1

Step 1: Draw 3-variable K-map

AB \ C	0	1
00	0	1
01	0	1
11	0	1
10	0	1

Step 2: Group 1's

- All 1's appear in column C=1 (vertical group of four 1's).

Step 3: Write simplified expression

- Since B and A change in group but C is always 1, the simplified function is:

F = C

🌟**Q6. Design a 4-bit binary adder using both half-adders and full-adders. Explain each step with truth tables, Boolean expressions, and circuit diagrams.**

Answer:

A **4-bit binary adder** adds two 4-bit binary numbers $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$, producing a 4-bit sum and a carry out.

Step 1: Add least significant bits (LSB) with Half-Adder

- Inputs: A_0, B_0
- Outputs: Sum S_0, Carry C_0

Step 2: Add next bits with Full-Adders

- For bits A_1, B_1, inputs are A_1, B_1, C_0
- Outputs: S_1, C_1

Repeat for A_2, B_2, C_1 and A_3, B_3, C_2.

Truth tables:

Half-Adder (LSB):

A0	B0	Sum (S0)	Carry (C0)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full-Adder (others):

A	B	Cin	Sum (S)	Cout
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:

- Half-Adder:
  Sum = $A \oplus B$
  Carry = $A \cdot B$
- Full-Adder:
  Sum = $A \oplus B \oplus Cin$
  Carry = $A \cdot B + B \cdot Cin + A \cdot Cin$

Circuit Diagram Overview:

- 1 Half-Adder for A_0, B_0
- 3 Full-Adders chained for bits 1 to 3
- Carry output of each adder goes to carry input of next higher bit

🌟**Q7. Simplify the following Boolean function using Boolean algebra rules:**

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

Answer:

Given:

$$F = A \cdot B + A \cdot B$$

Step 1: Apply the Distributive Law:

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

Step 2: Use Idempotent Law:

$$B + B = B$$

So,

$$F = A \cdot B$$

Simplified expression:

$$F = A \cdot B$$

🌟 Q8: Use De Morgan's laws to simplify the following function:

Given:

F(A, B, C) = A + B + AC

✳ **Important Note:**

This expression is already in simplified Sum of Products (SOP) form. De Morgan’s Theorem is used to simplify complemented Boolean expressions.

So we’ll demonstrate how to simplify the complement of this function using De Morgan’s Laws:

Step 1: Take the complement of the function

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

Step 2: Apply De Morgan’s Law

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

Step 3: Apply De Morgan’s Law again on $\overline{AC}$

$$\overline{AC} = \overline{A} + \overline{C}$$

Final Expression:

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

✅ This is the simplified **complement** of the function using De Morgan's laws. You can take complement again to return to original form if needed.

🌟 Q9: Simplify the following Boolean expressions

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

Start by expanding:

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

Since $\overline{B}B = 0$, the expression becomes:

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

Rewrite $\overline{A} + A\overline{B}$ using distribution:

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

Answer (a): $\overline{A} + \overline{B}$

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

Expand by distributive law:

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

Simplify terms:

- $A\overline{A} = 0$
- $\overline{B}B = 0$

So,

$$= 0 + AB + \overline{B}\overline{A} + 0 = AB + \overline{A}\overline{B}$$

This is the **XNOR** function.

Answer (b): $AB + \overline{A}, \overline{B}$

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

Apply distributive law:

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

Using the absorption law
$A + \overline{A}X = A + X$:

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

But since $A + \overline{B} + C$ contains the term A OR something else, it cannot be simplified further logically.

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

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

This is the same as in (b), but with reversed terms order:

$$\overline{AB} + AB$$

This is also the **XNOR** function.

Answer (d): $\overline{AB} + AB$

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

This is the same as (b) and (d):

$$AB + \overline{A}\,\overline{B}$$

Answer (e): $AB + \overline{A}, \overline{B}$

Summary of simplified expressions:

Expression	Simplified Form
(a)	$\overline{A} + \overline{B}$
(b)	$AB + \overline{A}, \overline{B}$ (XNOR)
(c)	$A + \overline{B} + C$
(d)	$\overline{AB} + AB$ (XNOR)
(e)	$AB + \overline{A}, \overline{B}$ (XNOR)

Note:

This chapter is designed to provide a solid foundation of knowledge, with the goal of deepening understanding and encouraging further exploration of the subject. The content has been carefully selected to support effective learning and inspire students to engage with the topic more deeply.

Author: Muhammad Asghar

Purpose: To contribute to education by offering insightful, valuable content that enhances learning and understanding.

Copyright & Usage Policy