COMPUTER

---

3. Boolean Algebra and Digital Logic

3.1 Basics of Digital Systems

3.1.1 Analog and Digital Signals

1. **Analog Signal:** A smooth and continuous signal that changes over time.
   Example: Sound waves, body temperature, radio signals.
2. **Digital Signal:** A signal that has only two values: 0 and 1.
   Used in computers and digital electronics.
3. **ADC (Analog to Digital Conversion):** Changes analog signals into digital form.
   Used in microphones, phones, and computers.
4. **DAC (Digital to Analog Conversion):** Changes digital signals into analog form.
   Used in speakers and audio systems.

Key Point:

- **Digital signals are better for long-distance transmission** because they resist noise and errors.

3.1.2 Fundamentals of Digital Logic

5. **Digital Logic:** Uses 0 and 1 to control digital devices like computers.
6. **Logic Levels:**
   - High voltage = 1 (True)
   - Low voltage = 0 (False)

3.2 Boolean Algebra and Logic Gates

3.2.1 Boolean Functions and Expressions

7. **Boolean Algebra:** A form of mathematics using True/False (1/0) values.
8. **Boolean Function:** A formula that gives an output (1 or 0) based on logical operations like AND, OR, and NOT.

3.2.1.1 Logic Operations

9. **AND Operation:**

- Symbol: ·
- Output is 1 only if **both inputs are 1**.
- Truth Table:
  - 0 · 0 = 0
  - 0 · 1 = 0
  - 1 · 0 = 0
  - 1 · 1 = 1

10. **OR Operation:**
    - Symbol: +
    - Output is 1 if **any one or both inputs are 1**.
    - Truth Table:
      - 0 + 0 = 0
      - 0 + 1 = 1
      - 1 + 0 = 1
      - 1 + 1 = 1
11. **NOT Operation:**
    - Symbol: ¬A or Ā
    - Changes the input value.
    - Truth Table:
      - NOT 0 = 1
      - NOT 1 = 0

3.2.1.2 Building Boolean Functions

12. Boolean expressions can be built using:
    - **AND**, **OR**, and **NOT** operators
    - Example 1: F(A, B) = A · B
    - Example 2: F(A, B, C) = A · B + A · C

3.2.2 Logic Gates and Their Functions

13. **Logic Gates:** Physical parts in a digital circuit that perform logical operations.

Gate	Symbol	Output is 1 When...
AND	A · B	Both inputs are 1
OR	A + B	At least one input is 1
NOT	¬A	Input is 0
NAND	¬(A · B)	At least one input is 0 (inverse of AND)
XOR	A ⊕ B	Only one input is 1

3.3 Simplification of Boolean Functions

14. **Simplification:** Reducing Boolean expressions using rules to make circuits faster and simpler.

Basic Boolean Laws:

15. **Identity Laws:**

- A + 0 = A
- A · 1 = A

16. **Null Laws:**

- A + 1 = 1
- A · 0 = 0

17. **Idempotent Laws:**

- A + A = A
- A · A = A

18. **Complement Laws:**

- A + $\bar{A}$ = 1
- A · $\bar{A}$ = 0

19. **Commutative Laws:**

- A + B = B + A
- A · B = B · A

20. **Associative Laws:**

- (A + B) + C = A + (B + C)
- (A · B) · C = A · (B · C)

21. **Distributive Laws:**

- A · (B + C) = A·B + A·C
- A + (B · C) = (A + B)(A + C)

22. **Absorption Laws:**

- A + (A·B) = A
- A · (A + B) = A

23. **De Morgan's Theorems:**

- ¬(A + B) = ¬A · ¬B
- ¬(A · B) = ¬A + ¬B

24. **Double Negation:**

- ¬(¬A) = A

3.4 Logic Diagrams

25. **Logic Diagram:** A circuit diagram that shows how logic gates are connected to form a Boolean function.

Steps to create:

- Identify the required logic gates.
- Arrange them according to the function.
- Connect inputs and outputs correctly.

3.5 Application of Digital Logic

3.5.1 Adder Circuits

26. **Half-Adder:** Adds two single binary digits (A and B).

- Outputs: Sum (S), Carry (C)
- Sum = A ⊕ B
- Carry = A · B

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

27. **Full-Adder:** Adds three binary digits (A, B, Carry In).

- Outputs: Sum and Carry Out
- Sum = A ⊕ B ⊕ Cin

- Carry = (A·B) + (Cin·(A⊕B))

3.5.2 Karnaugh Map (K-Map)

28. **K-Map:** A visual method to simplify Boolean expressions.
29. **Minterm:** A product term where all variables appear once in either true or complemented form.
30. **K-Map Grid Sizes:**

- 2 variables → 2×2 grid
- 3 variables → 2×4 grid
- 4 variables → 4×4 grid

Steps to Simplify Using K-Map:

1. Draw the K-map based on variables.
2. Fill it using output values.
3. Group adjacent 1s (in groups of 1, 2, 4, 8, etc.).
4. Write the simplified Boolean expression.

✅ **Summary:**

- Digital systems use **0 and 1 (binary)** to process information.
- **Logic gates** perform operations like AND, OR, and NOT.
- **Boolean algebra** helps build and simplify logic functions.
- **Adder circuits** are used for performing binary arithmetic.
- **K-maps** are a powerful way to simplify logic expressions visually.
