1. Explain Number System and types of number systems with examples.

A **Number System** is a method of representing numbers using symbols, rules and base values.

Computers and humans both use number systems to store, process and calculate values.

Types of Number Systems

(a) Decimal Number System

- Base: **10**
- Digits: 0–9
- Used in daily life.
- Example: $726 = 7\times10^2 + 2\times10^1 + 6\times10^0$

(b) Binary Number System

- Base: **2**
- Digits: 0 and 1
- Used by computers because electronic circuits understand ON/OFF states.
- Example: 1011 (Binary)
  $= 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 = 11$ (Decimal)

(c) Octal Number System

- Base: **8**
- Digits: 0–7
- Example: 45 (Octal)
  $= 4\times8^1 + 5\times8^0 = 37$ (Decimal)

(d) Hexadecimal Number System

- Base: **16**
- Digits: 0–9 and A–F
  (A=10, B=11…F=15)
- Example: 2A (Hex)
  $= 2\times16^1 + 10\times16^0 = 42$ (Decimal)

👉 **Conclusion:** Different number systems are used in computers for easier calculation, data representation and memory addressing.

2. Explain Decimal to Binary conversion with stepwise method.

Decimal → Binary conversion uses **Repeated Division by 2**.

Steps

1. Divide the decimal number by 2.
2. Record remainder (0 or 1).
3. Repeat division until quotient becomes 0.
4. Write remainders in **reverse order**.

Example: Convert 25 to binary

Step	Division	Quotient	Remainder
1	25 ÷ 2	12	1
2	12 ÷ 2	6	0
3	6 ÷ 2	3	0
4	3 ÷ 2	1	1
5	1 ÷ 2	0	1

Reverse the remainders → **11001**

👉 **$25_{10} = 11001_2$**

3. Explain Binary to Decimal conversion with example.

Binary → Decimal conversion uses **Positional Weight Method**.

Steps:

1. Multiply each bit from right to left by powers of 2.
2. Add all results.

Example: 1011_2

Bit	Weight	Value
1	2^3=8	8
0	2^2=4	0
1	2^1=2	2
1	2^0=1	1

Add: 8 + 0 + 2 + 1 = **11**

☞ **$1011_2 = 11_{10}$**

4. Describe Octal and Hexadecimal number system and conversion technique.

Octal System

- Base **8**
- Digits: 0–7

Binary → Octal Conversion Technique

- Group binary digits in **3 bits** from right.
- Convert each group.

Example:
101 111
= 5 7 → **57_8**

Hexadecimal System

- Base **16**
- Digits: 0–9, A–F

Binary → Hex Conversion Technique

- Group binary digits in **4 bits**.
- Convert each group.

Example:
1010 1101
= A D → **AD_{16}**

5. Explain Binary Addition and Subtraction rules with examples.

Binary Addition Rules

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 (Carry 1)

Example:
1011

- 0101
  = 10000

Binary Subtraction Rules

0 − 0 = 0

1 − 0 = 1

1 − 1 = 0

0 − 1 = 1 (Borrow 1 from next bit)

Example:
1010
− 0011
= 0111

6. Explain 1’s Complement and 2’s Complement methods to represent negative numbers.

1’s Complement

- Invert each bit (0→1, 1→0)

Example:
Number: 1010
1’s complement: **0101**

2’s Complement

- Take 1’s complement + 1

Example:
1010
1’s complement = 0101

- 1 → **0110**

☞ Used in computers to represent negative binary numbers.

7. Explain basic logic gates and their truth tables.

Logic gates perform logical operations.
Three basic gates:

AND Gate

Output = 1 only if both inputs are 1
Truth Table:

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

Output = 1 if any input is 1

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate

Single input — reverses the bit

A	Y
0	1
1	0

8. Explain Universal Gates (NAND & NOR) and why they are called Universal.

NAND Gate

- NOT + AND
  Truth table opposite of AND

NOR Gate

- NOT + OR
  Truth table opposite of OR

📌 **Why universal?**
Because **all other gates (AND, OR, NOT, XOR, XNOR)** can be constructed using only NAND or only NOR — hence called **Universal Gates.**

9. Evaluate Boolean expression using Boolean laws. (Example: A + AB)

Expression: **A + AB**

Applying **Absorption Law:**
A + AB = A (1 + B) = A × 1 = A

So result = **A**

10.Simplify Boolean expression using De-Morgan's theorem.

De-Morgan's Theorem:

1. (A + B)' = A'B'
2. (AB)' = A' + B'

Example:

(A + B)' Simplified:
= A'B'

11. Explain how Boolean expressions are implemented using logic gates.

- Boolean variables (A, B, C) represent input signals
- Boolean operators behave like gate functions
  Example:

$A \cdot B$ = AND gate
$A + B$ = OR gate
A' = NOT gate

Example:

Expression: $A + BC$
Implementation:

- AND gate for B and C
- OR gate between A and BC output

☞ Boolean expressions form logic circuits — used in ALU, CPU and digital electronics.

MEDIUM QUESTIONS (3–5 Marks Answers)

12. Convert following (any 3–4):

(a) Binary → Decimal

Method: Multiply each bit with powers of 2 and add.

Example: 1011_2

$= 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$
$= 8 + 0 + 2 + 1$
$= \mathbf{11_{10}}$

(b) Decimal → Binary

Method: Repeated division by 2 → reverse remainder.

Example: 19 →

19 ÷ 2 = 9 R1
9 ÷ 2 = 4 R1
4 ÷ 2 = 2 R0
2 ÷ 2 = 1 R0
1 ÷ 2 = 0 R1

Reverse remainders → **10011_2**

(c) Binary → Octal

Group 3 bits and convert.

Example: 110101_2

110 101
 6 5

= **65_8**

(d) Binary → Hexadecimal

Group 4 bits and convert.

Example: 11100101_2

1110 (E)
 0101 (5)

= **$E5_{16}$**

13. Write binary arithmetic addition rules.

Binary addition is performed digit-by-digit with carry rules:

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 and carry 1

Example:
 1011+0101
 = 10000

Binary addition is important in CPU arithmetic and ALU operations.

14. Write binary subtraction rules.

Binary subtraction uses borrowing:

$0 - 0 = 0$

$1 - 0 = 1$

$1 - 1 = 0$

$0 - 1 = 1$ (borrow 1 from next position)

Example:
 1010
 − 0011
 = 0111

Binary subtraction supports computer arithmetic operations.

15. Explain why computers use binary system.

Computers use binary system because:

✔ Electronic circuits understand only **two states**:

- ON (1)
- OFF (0)

✔ Binary values are reliable, easy to detect and fast for processing.

✔ Memory, CPU registers and logical circuits are based on binary switching.

☞ Therefore, binary number system is used for safe, noise-free and efficient computation inside computers.

16. Write truth table for XOR and XNOR gate.

XOR Gate Output is 1 when inputs are different

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

XNOR Gate Output is 1 when inputs are same

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

17. Compare AND, OR and NOT gates.

Feature	AND	OR	NOT
Inputs	2 or more	2 or more	Single
Output	1 only if all inputs are 1	1 if any input is 1	Complement
Symbol	D-shape with dot	Curved shape	Triangle + bubble

☞ AND performs multiplication, OR performs addition, NOT performs inversion.

18. Explain 1's complement method.

1's complement represents negative binary numbers by **bit inversion**.

Steps:

✔ Change 0 to 1

✔ Change 1 to 0

Example:

Binary: 1010

1's complement: **0101**

Used in representing negative values and subtraction operations.

19. Explain 2's complement method.

2's complement is the most widely used signed number representation.

Steps:

1. Take **1's complement**
2. Add **1**

Example:
Number: 1010
1's complement → 0101

- 1 → **0110**

Used by processors because subtraction becomes simpler using 2's complement.

20. Convert negative decimal number using complement representation.

Example: Represent −6 in binary (4 bit)

Step 1: Write +6 → 0110
Step 2: 1's complement → 1001
Step 3: Add 1 → 1010

So −6 = $\mathbf{1010_2}$

21.Reduce expression using Boolean laws.

Example expression: **A + AB**

Using Absorption Rule:
A + AB = A (1 + B) = A × 1 = **A**

Thus reduced expression = **A**

22.What is logical multiplication and logical addition?

✔ Logical Multiplication (AND)
Symbol: ·

Result is 1 only when **all inputs are 1**
Example: 1 · 1 = 1

✔ **Logical Addition (OR)**
Symbol: +
Result is 1 when **any input is 1**
Example: 1 + 0 = 1

23. What is decimal number system?

Decimal number system is the **base-10 system** used in everyday life.
It consists of **10 digits**: 0 to 9.
Each digit has a positional value based on powers of 10.

Example:
$347 = 3\times10^2 + 4\times10^1 + 7\times10^0$

24. What is binary number system?

Binary number system is a **base-2 system** used in computers.
It has only **two digits**: 0 and 1.
It represents ON/OFF states in digital circuits.

25. Define octal number system.

Octal number system is a **base-8 system**.
It uses digits **0 to 7**.
It is often used in digital electronics and memory representation.

26. Define hexadecimal number system.

Hexadecimal number system is **base-16**.
It uses digits **0–9 and A–F** where A=10 ... F=15.
It is used in computer memory addressing.

27. What is 2’s complement?

2’s complement is a method to represent **negative binary numbers**.
It is obtained by **adding 1 to the 1’s complement** of a number.

28. Write Boolean law A + A = ?

Answer: **A**

29. Write Boolean law A + A’ = ?

Answer: **1**

30. Draw truth table of NAND gate.

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

31. Write symbol of OR gate.

→ Curved input shape gate
(You can draw “≥1” symbol inside)

32. Give example of binary addition.

Example:
101+011
= 1000

33. Define logic gate.

Logic gate is an electronic circuit that performs a **logical operation** on binary inputs to produce a binary output.

34. Write symbol of XOR gate.

→ OR gate shape with **extra curved line** before input
(Symbolically = ⊕)

35. Define NOT gate.

NOT gate is a single input gate that **inverts** the input.
If input = 1, output = 0 and vice-versa.

⋆ **MCQ / 1-MARK ANSWERS**

36. Base of binary number system? → 2

37. Base of hexadecimal system? → 16

38. Binary representation of 13? → 1101_2

39. Boolean law: $A + 1 = ?$ → 1

40. Boolean law: $A \cdot 0 = ?$ → 0

41. Output of XOR gate when A=1, B=0? → 1

42. 1’s complement of 1010? → 0101

43. 2’s complement of 1010? → 0110

44. AND gate output when A=1, B=1? → 1

45. NOR gate is opposite of which gate? → OR