

# Assignment-1

## 1. Explain different Number Systems with examples.

A number system is a method of representing numbers using digits or symbols.

Different number systems are used in computers because computers understand only binary signals.

### (1) Decimal Number System (Base 10)

- ∅ Digits: 0–9
- ∅ Most commonly used in daily life.

Example:

$$245_{10} = (2 \times 10^2) + (4 \times 10^1) + (5 \times 10^0)$$

### (2) Binary Number System (Base 2)

- ∅ Digits: 0 and 1 only
- ∅ Computers use this system because electronic circuits work on ON/OFF (1/0) signals.

Example:

$$1101_2 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 13$$

### (3) Octal Number System (Base 8)

- ∅ Digits: 0–7
- ∅ Used as a shortcut for binary.

Example:

$$57_8 = 5 \times 8 + 7 \times 1 = 47$$

### (4) Hexadecimal Number System (Base 16)

- ∅ Digits: 0–9 and A–F  
(A=10, B=11, C=12, D=13, E=14, F=15)

Example:

$$2F_{16} = 2 \times 16 + 15 = 47$$

## 2. Converting Decimal to Binary, Octal, Hexadecimal

### (A) Decimal to Binary

Use division by 2 and write remainders.

Example: Convert 25 to Binary

$$25 \div 2 = 12 \text{ R}1$$

$$12 \div 2 = 6 \text{ R}0$$

$$6 \div 2 = 3 \text{ R}0$$

$$3 \div 2 = 1 \text{ R}1$$

$$1 \div 2 = 0 \text{ R}1$$

$$\text{Binary} = 11001_2$$

### (B) Decimal to Octal

Divide by 8.

Example: Convert 125

$$125 \div 8 = 15 \text{ R}5$$

$$15 \div 8 = 1 \text{ R}7$$

$$1 \div 8 = 0 \text{ R}1$$

$$\text{Octal} = 175_8$$

### (C) Decimal to Hexadecimal

Divide by 16.

Example: Convert 254

$$254 \div 16 = 15 \text{ R}14$$

$\rightarrow E$

$$15 \div 16 = 0 \text{ R}15$$

→

F

Hex = FE<sub>16</sub>

### 3. Perform Binary Addition & Subtraction

#### (A) Binary Addition Rules

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ (carry 1)}$$

Example

1011

+ 1101

-----

11000

#### (B) Binary Subtraction Rules

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1 \text{ (borrow 1)}$$

Example

1010

- 0111

-----

0011

## **4. Complements and Use to Represent Negative Numbers**

Complements are used for simplifying subtraction and representing negative numbers.

### **(A) 1's Complement**

- ⌚ Flip all bits (0

$\rightarrow 1$

, 1

$\rightarrow 0)$

Example:

1's complement of 1010 = 0101

### **(B) 2's Complement**

- ⌚ 1's complement + 1

Example:

Number = 0101 (5)

1's complement = 1010

+1 = 1011 (represents

-5)

Why Complements?

- ⌚ Used to represent negative numbers
- ⌚ Simplifies subtraction
- ⌚ CPU uses 2's complement for all arithmetic

## **5. What is a Logic Gate? Explain AND, OR, NOT with Truth Table &**

**Symbols.**

Logic Gates are digital circuits that perform logical operations on binary inputs.

### (A) AND Gate

- ∅ Output is 1 only if both inputs are 1.

Truth Table

A B Y

0 0 0

0 1 0

1 0 0

1 1 1

### (B) OR Gate

- ∅ Output is 1 if any one input is 1.

Truth Table

A B Y

0 0 0

A B Y

0 1 1

1 0 1

1 1 1

### (C) NOT Gate

- ∅ Also called Inverter

- ∅ Output is opposite of input.

Truth Table

A Y

0 1

1 0

## **6. Evaluation of Logical Expressions Using Basic Gates**

Logical expression uses AND, OR, NOT to produce results.

Example Expression:

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

With Parentheses:

1. First evaluate  $B \cdot C$  using AND
2. Add result to A using OR

Example Calculation

$$A=1, B=1, C=0$$

Step 1:

$$B \cdot C = 1 \cdot 0 = 0$$

Step 2:

$$A + 0 = 1$$

$$\rightarrow Y = 1$$

Without Parentheses

Follow operator precedence:

1. NOT
2. AND
3. OR

Example:

$$Y = A + B \cdot \text{NOT } C$$

Compute  $C'$ , then  $B \cdot C'$ , then OR with A.

## **7. NAND & NOR Gates + Universal Gate Property**

(A) NAND Gate

⦿ Output is 0 only when both inputs are 1.

⦿  $\text{NAND} = \text{NOT(AND)}$

Truth Table

A B Y

0 0 1

0 1 1

1 0 1

1 1 0

(B) NOR Gate

⦿ Output is 1 only when both inputs are 0.

⦿  $\text{NOR} = \text{NOT(OR)}$

Truth Table

A B Y

0 0 1

0 1 0

A B Y

1 0 0

1 1 0

Universal Gate Property

Both NAND and NOR can be used to build:

✓ AND

✓ OR

✓ NOT

✓ XOR

✓ All digital circuits

→ That is why they are called UNIVERSAL GATES.

## 8. Perform Conversions + Binary Arithmetic + Logic Expression Evaluation

### (A) Number System Conversions

Binary  $\leftrightarrow$  Octal

⦿ Group bits in 3

Binary  $\leftrightarrow$  Hex

⦿ Group bits in 4

Example:

Binary 110111 → Hex

= 0110 1111

→ 6F<sub>16</sub>

### (B) Binary Arithmetic

Example:

1010

+0111

----

10001

### (C) Logic Expression Evaluation

Expression:

$$Y = A \cdot B + C'$$

$$A=1, B=0, C=1$$

$$C' = 0$$

$$A \cdot B = 0$$

$$0+0=0$$

$$\mathrm{Y}=0$$