WELCOME

TO

COMPUTER APPLICATIONS IN PHARMACY

UNIT 1

TOPIC :

- **Number system :** Binary number system, Decimal number system, Octal number system, Hexadecimal number systems, conversion decimal to binary, binary to decimal, octal to binary etc, binary addition, binary
subtraction–One's complement ,Two's complement method, binary
multiplication, binary division

NUMBER SYSTEMS

- A number system is a way to represent numbers using a consistent set of symbols. It defines how numbers are written and interpreted by computers and humans. It is essential in digital electronics and computing.

Types of Number Systems

Number System	Base	Digits Used	Example
Binary	2	0, 1	1011_2
Decimal	10	0 to 9	247_{10}
Octal	8	0 to 7	75_8
Hexadecimal	16	0 to 9 and A to F (10–15)	$2F_{16}$

Binary Number System

→ The Binary Number System is a positional numeral system that uses base 2, which means it uses only two digits – 0 and 1.
It is used extensively in mathematics and digital electronics, especially in computers, because electronic circuits have only two states: ON (1) and OFF (0).

Key Features

- **Base**: 2
- **Digits used**: Only **0** and **1**
- **Also called**: Base-2 system
- **Positional System**: Each digit has a weight depending on its position from right to left, based on powers of 2.

Decimal Number System

→ The Decimal Number System is the standard system used by humans for counting and calculations. It is a base-10 positional numeral system, meaning it uses ten digits (0 to 9), and the value of each digit depends on its position (or place value) in the number.

Key Features

Feature	Description
Base	10
Digits Used	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Positional System	Each digit's value depends on its position
Also Known As	Hindu-Arabic numeral system
Example	$425 = (4\times10^2) + (2\times10^1) + (5\times10^0) = 400 + 20 + 5 = 425$

How It Works (Positional Value)

- In a decimal number, each digit is multiplied by 10 raised to the power of its position, counting from right to left starting at 0.

Example: 2374

Position	**3**	**2**	**1**	**0**
Digit	2	3	7	4
Value	2×10^3	3×10^2	7×10^1	4×10^0
Result	2000	300	70	4

Total = 2000 + 300 + 70 + 4 = 2374

Octal Number System

- The Octal Number System is a positional numeral system that uses base 8, meaning it uses eight digits: 0 to 7.
  Each digit in an octal number represents a power of 8, depending on its position.

Key Features

Feature	Description
Base	8
Digits Used	0, 1, 2, 3, 4, 5, 6, 7
Positional System	Each digit has a value based on powers of 8
Example	$725_8 = (7\times8^2) + (2\times8^1) + (5\times8^0) = 469$

Hexadecimal Number System

→ The Hexadecimal Number System is a positional numeral system with a base of 16.

It uses sixteen symbols:

0–9 for values 0 to 9

A–F (or a–f) for values 10 to 15

Hexadecimal Digits

Hex Digit	Decimal Value	Binary Equivalent
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Why Use Hexadecimal?

- Represents large binary numbers in a compact, readable form.
- 1 hexadecimal digit = 4 binary digits (bits)
- Used in memory addresses, machine code, HTML colors, MAC addresses, and low-level programming.

Number System Conversions Table

Decimal	Binary	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Decimal to Binary Conversion

→ Decimal to Binary Conversion means converting a number from the base-10 system (which uses digits 0–9) into the base-2 system (which uses only 0 and 1).

→ This is an essential concept in digital electronics and computer science because computers operate using the binary (base-2) system.

Method: Repeated Division by 2

To convert a decimal number to binary:

1. **Divide** the number by 2.
2. **Record the remainder** (it will be 0 or 1).

3. **Divide the quotient by 2 again**, and repeat the process.
4. **Continue until the quotient is 0.**
5. **Write the remainders in reverse order** (from last to first) to get the binary number.

Example 1: Convert 25_{10} to Binary

Step 1: 25 ÷ 2 = 12 remainder 1

Step 2: 12 ÷ 2 = 6 remainder 0

Step 3: 6 ÷ 2 = 3 remainder 0

Step 4: 3 ÷ 2 = 1 remainder 1

Step 5: 1 ÷ 2 = 0 remainder 1

Write remainders in reverse: → 11001_2

Example 2: Convert 10_{10} to Binary

10 ÷ 2 = 5 remainder 0

5 ÷ 2 = 2 remainder 1

2 ÷ 2 = 1 remainder 0

1 ÷ 2 = 0 remainder 1

Answer = 1010_2

Binary to Decimal Conversion

→ Binary to Decimal Conversion means converting a number from the binary system (base-2) into the decimal system (base-10).
Binary numbers use only two digits: 0 and 1, while decimal uses ten digits: 0 to 9.

Method: Positional Value Method

- Each digit (bit) in a binary number has a positional value based on powers of 2, starting from the rightmost digit (least significant bit - LSB) to the leftmost digit (most significant bit - MSB).

General Formula:

If a binary number is written as:

$b_n b_{n-1} \ldots b_2 b_1 b_0$ (base 2)

Then:

Decimal = $(b_n \times 2^n) + (b_{n-1} \times 2^{n-1}) + \ldots + (b_1 \times 2^1) + (b_0 \times 2^0)$

Example 1: Convert 1011_2 to Decimal

Bit Position (from right)	**3**	**2**	**1**	**0**
Binary Digits	1	0	1	1
Power of 2	2^3	2^2	2^1	2^0
Calculation	8	0	2	1

= (1 × 8) + (0 × 4) + (1 × 2) + (1 × 1)

= 8 + 0 + 2 + 1 = **11_{10}**

Example 2: Convert 11001_2 to Decimal

= $(1\times2^4) + (1\times2^3) + (0\times2^2) + (0\times2^1) + (1\times2^0)$

= 16 + 8 + 0 + 0 + 1 = **25_{10}**

Octal to Binary Conversion

Octal to Binary conversion means converting a number from the **octal system (base 8)** to the **binary system (base 2)**.

- Octal system uses **digits 0–7**.
- Binary system uses **digits 0 and 1**.

1 Octal digit = 3 Binary digits (bits)

This is because:

- 8=238 = 2^38=23
- So, each octal digit directly converts into a **3-bit binary number**.

Step-by-Step Method

Steps**:**

1. Take each digit of the octal number (from left to right).
2. **Convert it into its 3-digit binary equivalent.**
3. Combine all the binary groups to get the full binary number.

Binary Equivalents of Octal Digits (0 to 7)

Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Example 1: Convert 345_8 to Binary

Step-by-step:

$3 \rightarrow 011$

$4 \rightarrow 100$

$5 \rightarrow 101$

$\rightarrow$ Binary = 011100101_2

Example 2: Convert 127_8 to Binary

$1 \rightarrow 001$

$2 \rightarrow 010$

$7 \rightarrow 111$

$\rightarrow$ Binary = 001010111_2

Binary to Octal Conversion

Binary to Octal conversion means converting a number from the **binary system (base-2)** to the **octal system (base-8)**.

- Binary uses digits: **0 and 1**
- Octal uses digits: **0 to 7**

Group binary digits in sets of 3 (from right to left)
Each group of **3 binary digits = 1 octal digit**
Because $2^3 = 8$

Step-by-Step Method

Steps:

- ❖ Start grouping the binary number from the right side into 3-digit groups.
  (Add leading 0s if necessary to complete a group of 3.)
- ❖ Convert each group of 3 binary digits into its octal equivalent.
- ❖ Combine the octal digits to get the final answer.

Binary to Octal Table

Binary	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Example 1: Convert 110101_2 to Octal

Step-by-step:

Step 1: Group binary into 3s from right:

110 101

Step 2: Convert each group:

110 → 6

101 → 5

Answer = 65_8

Example 2: Convert 1011101_2 to Octal

Step 1: Group:

001 011 101 (add one leading zero)

Step 2: Convert:

001 → 1

011 → 3

101 → 5

Answer = 135_8

Hexadecimal to Binary Conversion

Hexadecimal to Binary Conversion means converting a number from the **hexadecimal system (base-16)** to the **binary system (base-2)**.

- **Hexadecimal** uses 16 symbols:
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B (11), C (12), D (13), E (14), F (15)
- **Binary** uses only: **0 and 1**

1 Hexadecimal digit = 4 Binary digits (bits)

Because 16 = 2^4, each hex digit directly converts to 4-bit binary.

Hex to Binary Table

Hex	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Step-by-Step Method

1. Take **each hexadecimal digit**.
2. Replace it with its **4-digit binary equivalent** using the table above.
3. **Combine** all binary groups to get the final binary number.

Example 1: Convert $3F_{16}$ to Binary

Break into individual digits:
3 = 0011
F = 1111

Answer:
$3F_{16} = 00111111_2$

Example 2: Convert $A5_{16}$ to Binary

A = 1010
5 = 0101

Answer:
$A5_{16} = 10100101_2$

Binary to Hexadecimal Conversion

Binary to Hexadecimal conversion means converting a number from the **binary system (base-2)** into the **hexadecimal system (base-16)**.

- **Binary** uses digits: **0 and 1**
- **Hexadecimal** uses 16 symbols:
  0–9 and A (10), B (11), C (12), D (13), E (14), F (15)

1 Hexadecimal digit = 4 Binary digits (bits)

Because $16 = 2^4$, we convert **each group of 4 binary digits** into **1 hexadecimal digit**.

Hexadecimal Conversion Table

Binary	Hex
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Step-by-Step Method

1. **Start from the right**, group the binary digits into sets of **4 bits**.
2. If the leftmost group has **fewer than 4 bits**, add **leading zeros**.
3. Convert each 4-bit group into its **hexadecimal equivalent** using the table.
4. **Combine** all hex digits to form the final hexadecimal number.

Example 1: Convert 10101110_2 to Hex

Step 1: Group into 4 bits → 1010 1110
Step 2: Convert using table:

- 1010 → A
- 1110 → E

Answer: $\mathbf{AE_{16}}$

Example 2: Convert 111001_2 to Hex

Step 1: Add leading zeros: 0011 1001
Step 2: Convert:

- 0011 → 3
- 1001 → 9

Answer: $\mathbf{39_{16}}$

Binary Addition

→ Binary addition is a method of adding two binary numbers (base-2). It follows the same logic as decimal addition, but only uses 0 and 1 as digits.

Basic Binary Addition Rules

- Just like decimal addition has rules (e.g., 9+1 = 10), binary addition has four basic combinations:

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1
1	1	1	1 (if there's a carry from previous bit)

Example 1: Simple Binary Addition

Add: $1011_2 + 1101_2$

1011
+ 1101

11000

Step-by-step:

1 + 1 = 10 (0 carry 1)

1 + 0 + carry 1 = 10 (0 carry 1)

0 + 1 + carry 1 = 10 (0 carry 1)

1 + 1 + carry 1 = 11 (1 carry 1)

Final carry = 1 → Add to front

Answer: 11000_2

Binary Subtraction

→ Binary subtraction is the process of subtracting one binary number from another, using base-2 digits (only 0 and 1).
It is similar to decimal subtraction, but with different rules due to only having two digits.

Basic Binary Subtraction Rule

A (minuend)	B (subtrahend)	Difference	Borrow
0	0	0	0
1	0	1	0
1	1	0	0
0	1	1	1 (borrow from left bit)

Example 1: Subtract $1011_2 - 0101_2$

1011

- 0101

0110

Step-by-step:

1 – 1 = 0

1 − 0 = 1

0 – 1 → borrow → becomes 10 – 1 = 1

(Left 1 becomes 0 due to borrow)

1 − 0 = 1

Answer: **0110_2**

Binary Multiplication

→ Binary multiplication is the process of multiplying two binary numbers (base-2), using only 0 and 1.

→ It is similar to decimal multiplication, but follows simple rules based on binary logic.

Basic Binary Multiplication Rules

A	B	Product
0	0	0
0	1	0
1	0	0
1	1	1

Step-by-Step Binary Multiplication Process

- ❖ Multiply the bottom number with each digit of the top number, starting from the rightmost bit.
- ❖ For each new row (partial product), shift left by one position (just like adding zeros in decimal multiplication).
- ❖ Add all the partial products using binary addition.

Example 1: Multiply 101 × 11

```
     101   (5 in decimal)
  ×   11   (3 in decimal)
  --------
     101   ← 101 × 1
+   1010   ← 101 × 1 (shifted one position)
  --------
    1111   (15 in decimal)
```

Answer: 101 × 11 = $\mathbf{1111_2}$

Binary Division

→ Binary division is the process of dividing one binary number (called the dividend) by another (called the divisor) using base-2 arithmetic (only 0s and 1s).

→ Binary division is similar to long division in decimal, but simpler because the digits are only 0 and 1.

Binary Division Rules

Dividend Bit	Divisor Bit	Result
0 ÷ 1	0	
1 ÷ 1	1	
0 ÷ 0	**Undefined**	
1 ÷ 0	**Undefined**	

Steps of Binary Division (Long Division Method)

1. Compare the divisor with the leftmost bits of the dividend.
2. If the divisor is **smaller or equal**, subtract and write 1 in the quotient.
3. If it's **larger**, bring down the next bit of the dividend and write 0 in the quotient.
4. Repeat until all bits of the dividend are processed.
5. The remainder is what's left after the last subtraction.

Example 1: Divide $1010_2 \div 10_2$ (10 = 2 in decimal)

```
      101
    ________
 10 |1010
    -10      ← 10 × 1 = 10
    ----
     010
     -00     ← 10 × 0 = 00
     ----
      10
```

-10 ← 10 × 1 = 10

0

Quotient = $101_2 = 5$

Remainder = 0