

## **WEEK 4 – Number Systems**

### I. Definition:

- are systems in mathematics that are used to express numbers in various forms and are understood by computers.
- It is used for counting, measuring and performing arithmetic calculations.
- A representation of numbers by using digits or other symbols in a consistent manner.

### **1. Binary Number System (Base - 2)**

- Uses only two digits: 0 and 1.
- The numbers in this system have a base of 2.
- The binary number system does not deal with other numbers such as 2,3,4,5 and so on..

For example: 100012, 1111012, 10101012

### **2. Octal Number System (Base - 8)**

- Uses eight digits: 0,1,2,3,4,5,6 and 7
- The numbers in this system have a base of 8.

For example: 358, 238, 1418

### **3. Decimal Number System (Base - 10)**

- Uses ten digits: 0,1,2,3,4,5,6,7,8 and 9
- The numbers in this system have a base of 10.
- The decimal number system is the system that we generally use to represent numbers in real life. If any number is represented without a base, it means that its base is 10.

For example: 72310, 3210, 425710

### **4. Hexadecimal Number System (Base - 16)**

- Uses sixteen digits/alphabets: 0,1,2,3,4,5,6,7,8,9 and A,B,C,D,E,F
- The numbers in this system have a base of 16.
- A-F of the hexadecimal system means the numbers 10-15 of the decimal number system respectively.
- For example: 7B316, 6F16, 4B2A16

## **II. Integer Conversions**

### **Integer Conversions**

#### **1.1 Binary → Decimal**

Step 1: Assign each binary digit a power of 2 (right → left, LSB =  $2^0$ )

Step 2: Multiply each digit by its corresponding power of 2

Step 3: Add the results

Example:  $111101_2 \rightarrow$  Decimal

$$1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 32 + 16 + 8 + 4 + 0 + 1 = 61_{10}$$

#### **1.2 Binary → Octal**

Step 1: Group binary digits by 3 (right → left)

Step 2: Convert each group to its octal digit

Example:  $111101_2 \rightarrow$  Octal

Group: 111 101 → 7 5 →  $75_8$

### **1.3 Binary → Hexadecimal**

Step 1: Group binary digits by 4 (right → left)

Step 2: Convert each group to hex

Example:  $10110111_2 \rightarrow$  Hex

Group:  $1011\ 0111 \rightarrow B7_{16}$

### **1.4 Decimal → Binary**

Step 1: Divide the integer by 2

Step 2: Record remainder

Step 3: Repeat division with quotient until 0

Step 4: Read remainders from bottom to top

Example:  $75_{10} \rightarrow$  Binary

$$75 \div 2 = 37 \text{ R}1$$

$$37 \div 2 = 18 \text{ R}1$$

$$18 \div 2 = 9 \text{ R}0$$

$$9 \div 2 = 4 \text{ R}1$$

$$4 \div 2 = 2 \text{ R}0$$

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

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

Result:  $1001011_2$

### **1.5 Decimal → Octal**

Step 1: Divide the integer by 8

Step 2: Record remainder

Step 3: Repeat division until quotient = 0

Step 4: Read remainders bottom → top

Example:  $394_{10} \rightarrow$  Octal

$$394 \div 8 = 49 \text{ R}2$$

$$49 \div 8 = 6 \text{ R}1$$

$$6 \div 8 = 0 \text{ R}6$$

Result:  $612_8$

### **1.6 Decimal → Hexadecimal**

Step 1: Divide integer by 16

Step 2: Record remainder (use A-F for 10–15)

Step 3: Repeat until quotient = 0

Step 4: Read remainders bottom → top

Example:  $894_{10} \rightarrow$  Hex

$$894 \div 16 = 55 \text{ R}14 \rightarrow E$$

$$55 \div 16 = 3 \text{ R}7$$

$$3 \div 16 = 0 \text{ R}3$$

Result:  $37E_{16}$

### **1.7 Octal → Binary**

Step 1: Convert each octal digit into 3-bit binary

Step 2: Combine groups

Example:  $568_8 \rightarrow$  Binary

$5 \rightarrow 101, 6 \rightarrow 110, 8 \rightarrow 1000 \rightarrow 1011101000_2$

### **1.8 Octal → Decimal**

Step 1: Assign powers of 8 (right → left)

Step 2: Multiply each digit by its corresponding power of 8

Step 3: Sum results

Example:  $370_8 \rightarrow$  Decimal

$3 \times 8^2 + 7 \times 8^1 + 0 \times 8^0 = 192 + 56 + 0 = 248_{10}$

### **1.9 Octal → Hexadecimal**

Step 1: Convert octal → binary

Step 2: Group binary digits by 4 → convert to hex

Example:  $213_8 \rightarrow$  Hex

Octal → Binary:  $010\ 001\ 011 \rightarrow 10001011_2$

Binary → Hex:  $1000\ 1011 \rightarrow 8B_{16}$

### **1.10 Hexadecimal → Binary**

Step 1: Convert each hex digit to 4-bit binary

Step 2: Combine groups

Example:  $A9_{16} \rightarrow$  Binary

$A \rightarrow 1010, 9 \rightarrow 1001 \rightarrow 10101001_2$

### **1.11 Hexadecimal → Decimal**

Step 1: Assign powers of 16 (right → left)

Step 2: Multiply each hex digit by its corresponding power of 16

Step 3: Sum results

Example:  $23E_{16} \rightarrow$  Decimal

$2 \times 16^2 + 3 \times 16^1 + E(14) \times 16^0 = 512 + 48 + 14 = 574_{10}$

### **1.12 Hexadecimal → Octal**

Step 1: Hex → Binary (4 bits per hex)

Step 2: Binary → Octal (group 3 bits)

Example:  $AC_{16} \rightarrow$  Octal

Hex → Binary:  $A \rightarrow 1010, C \rightarrow 1100 \rightarrow 10101100_2$

Binary → Octal:  $101\ 011\ 00 \rightarrow 253_8$

## **Fractional Conversions (Decimal ↔ Binary fractions)**

### **2.1 Decimal Fraction → Binary Fraction**

Step 1: Multiply fraction by 2

Step 2: Take integer part as next binary digit

Step 3: Repeat with new fraction until 0 or desired precision

Example:  $0.625_{10} \rightarrow$  Binary

$0.625 \times 2 = 1.25 \rightarrow 1$

$0.25 \times 2 = 0.5 \rightarrow 0$

$$0.5 \times 2 = 1 \rightarrow 1$$

Result:  $0.101_2$

## 2.2 Binary Fraction → Decimal Fraction

Step 1: Multiply each binary digit after the point by  $2^{-n}$  ( $n$  = position after point)

Step 2: Sum results

Example:  $0.101_2 \rightarrow$  Decimal

$$1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 0.5 + 0 + 0.125 = 0.625_{10}$$

## III. Binary Arithmetic (added)

### Addition & Subtraction

Addition Steps:

Step 1: Align the binary numbers.

Step 2: Add digits column by column, starting from the LSB (rightmost bit).

Step 3: Carry over if the sum is greater than or equal to 2.

- Example – Addition:

$$1011_2 + 1101_2 = 11000_2$$

- Subtraction Steps:

Step 1: Align the binary numbers.

Step 2: Subtract digits column by column from LSB.

Step 3: Borrow 1 if the top digit is less than the bottom digit.

- Example – Subtraction:

$$10110_2 - 1101_2 = 1001_2$$

### Multiplication & Division

Multiplication Steps:

Step 1: Multiply binary numbers like decimal multiplication ( $1 \times 1 = 1$ ,  $1 \times 0 = 0$ ).

Step 2: Shift left for each new row (like multiplying by powers of 2).

Step 3: Add the partial products.

- Example – Multiplication:

$$101_2 \times 11_2 = 1111_2$$

- Division Steps:

Step 1: Divide like decimal long division.

Step 2: Subtract multiples of the divisor from the dividend.

Step 3: Bring down the next bit until complete.

- Example – Division:

$$1101_2 \div 10_2 = 110_2 \text{ remainder } 1_2$$

### Overflow & Underflow

**Overflow:** Occurs when a binary operation produces a result larger than the allotted bit size can store.

Example: Adding  $1111_2$  (4 bits) +  $0001_2$  =  $10000_2 \rightarrow$  overflow in 4-bit system.

**Underflow:** Occurs when a negative result cannot be represented in the allotted bit size (mainly in signed binary numbers).