

Introduction to computer

Chapter Four

Data Representation

- Data and instructions cannot be entered and processed directly into computers using human language.
- Any type of data be it numbers, letters, special symbols, sound or pictures must first be converted into machine-readable form i.e. binary form.

BIT

- Bits are the smallest units and can convey only two possible states 0 or 1;
- bit stands for binary digits;
- A bit is a single element in the computer, on a disk that stands for either “ON” indicating 1 or “OFF” indicating 0;
- In the computer “ON” is represented by the existence of current and “OFF” is represented by the non existence of current.

BYTE

- Bits can be organized into large units to make them represent more and meaningful information;
- This large unit is called a byte and is the basic “unit of data representation” in a computer system; The commonly used byte contains 8 bits;

WORD

- Word refers the number of bits that a computer process at a time or a transmission media transmits at a time
- Although bytes can store or transmit information, the process can even be faster if more than one byte is processed at a once; A combination of bytes, then form a “word”
- A word can contain one, two, three or four bytes based on the capacity of the computer;
- Word length is usually given in bits
- We say that a computer is an 8-bits, a 16 bit, a 32 bit or a 64 bit computer to indicate that the amount of data it can process at a time.

Computer Number System

- A number system is a set of symbols used to represent values or counting derived from a common base or radix.
- There are various number systems
- E.g. Decimal, Binary, octal-decimal, hexadecimal etc

- ***The Decimal Number System***

- The Decimal number system is based on the ten different digits (or symbols) 0,1,2,3,4,5,6,7,8,9.
- We say it is a *base ten number system*
- Though it is widely used, it is inconvenient for computer to represent data. So we need another number system.

The Binary number system

- Binary number system is based on the two different digits; 0 and 1;
- It is important to note that every decimal number system has its equivalent binary number;
- Operation on binary number system is also possible

<u><i>Binary</i></u>	<u><i>Decimal</i></u>
----------------------	-----------------------

0	0
---	---

1	1
---	---

10	2
----	---

11	3
----	---

100	4
-----	---

<u><i>Binary</i></u>	<u><i>Decimal</i></u>
----------------------	-----------------------

101	5
-----	---

110	6
-----	---

111	7
-----	---

1000	8
------	---

1001	9
------	---

- **Octal number System (base 8) (Oct)**

- ⇧ Consists of eight digits ranging from 0-7. the place value of octal numbers goes

- up in factors of eight from right to left.

- ⇧ Each position to the left of a digit increases by a power of 8.

- ⇧ Each position to the right of a digit decreases by a power of 8.

Example: $12403_8 = 3 \cdot 8_0 + 0 \cdot 8_1 + 4 \cdot 8_2 + 2 \cdot 8_3 + 1 \cdot 8_4$
 $= 3 \cdot 1 + 0 \cdot 8 + 4 \cdot 64 + 2 \cdot 512 + 1 \cdot 4096$
 $= 5379_{10}$

- **Hexadecimal number system (16) (hex)**

This is a base 16 number system that consists of sixteen digits ranging from 0-9 and letters A-F where A is equivalent to 10, B to 11 up to F which is equivalent to 15 in base ten system.

- The place value of hexadecimal numbers goes up in factors of sixteen. $Z = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$; $B = 16$

Conversion from one Base to Another

- Binary to Decimal Conversion
- The process of converting binary to decimal is quite simple.
- The process starts from multiplying the bits of binary number with its corresponding positional weights.
- And lastly, we add all those products.

- **Example 1: $(10110.001)_2$**

- We multiplied each bit of $(10110.001)_2$ with its respective positional weight, and last we add the products of all the bits with its weight.

- $(10110.001)_2 = (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$

+

$$(0 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3})$$

$$(10110.001)_2 = (1 \times 16) + (0 \times 8) + (1 \times 4) + (1 \times 2) + (0 \times 1) + (0 \times 1/2) + (0 \times 1/4) + (1 \times 1/8)$$

$$(10110.001)_2 = 16 + 0 + 4 + 2 + 0 + 0 + 0 + 0.125$$

$$(10110.001)_2 = (22.125)_{10}$$

Binary to Octal Conversion

1. In the first step, we have to make the pairs of three bits on both sides of the binary point. If there will be one or two bits left in a pair of three bits pair, we add the required number of zeros on extreme sides.
2. In the second step, we write the octal digits corresponding to each pair.

- **Example 1: $(111110101011.0011)_2$**

- 1. Firstly, we make pairs of three bits on both sides of the binary point.

111 110 101 011.001 1

- On the right side of the binary point, the last pair has only one bit. To make it a complete pair of three bits, we added two zeros on the extreme side.
- 111 110 101 011.001 100
- 2. Then, we wrote the octal digits, which correspond to each pair.

$$(111110101011.0011)_2 = (7653.14)_8$$

Binary to Hexadecimal Conversion

1. In the first step, we have to make the pairs of four bits on both sides of the binary point. If there will be one, two, or three bits left in a pair of four bits pair, we add the required number of zeros on extreme sides.
2. In the second step, we write the hexadecimal digits corresponding to each pair.

- **Example 1: $(11110101011.0011)_2$**

- 1. Firstly, we make pairs of four bits on both sides of the binary point.

111 1010 1011.0011

- On the left side of the binary point, the first pair has three bits. To make it a complete pair of four bits, add one zero on the extreme side.

0111 1010 1011.0011

- 2. Then, we write the hexadecimal digits, which correspond to each pair.

$(011110101011.0011)_2 = (7AB.3)_{16}$

- Decimal to Octal
- Decimal numbers can be converted to octal by repeated division of the number by 8 while recording the remainder. Let's take an example to see how this happens.

		Remainder	
8	473		
8	59	1	MSD
8	7	3	↑
	0	7	LSD

- Reading the remainders from bottom to top,
- $473_{10} = 731_8$

Decimal to Hexadecimal

- Decimal numbers can be converted to octal by repeated division of the number by 16 while recording the remainder. Let's take an example to see how this happens.

		Remainder	
16	423		
16	26	7	
16	1	A	
	0	1	

Reading the remainders from bottom to top we get,

423₁₀ = 1A7 **16**