WEEK- 6

Chapter 2

Data Representation in Computer Systems

2.1 Introduction.

• ***<u>Bit</u>***: Is the most basic unit of information in a computer.

  – It is a state of “on” or “off” in a digital circuit.

  – Sometimes these states are “high” or “low” voltage instead of “on” or “off.”

• ***<u>Byte</u>***: Is a group of eight bits.

  – A byte is the smallest possible *addressable* unit of computer storage.

  – The term, “addressable,” means that a particular byte can be retrieved according to its location in memory.

• ***<u>Word</u>***: Is a contiguous group of bytes.

  – Words can be any number of bits or bytes.

  – Word sizes of 16, 32, or 64 bits are most common.

  – In a word-addressable system, a word is the smallest addressable unit of storage.

• A group of four bits is called a *nibble*.

Bytes, therefore, consist of two nibbles:

a “high-order nibble,” and a “low-order nibble”.

2.2 Decimal to Binary Conversions.

• Bytes store numbers when the position of each bit represents a power of 2.

– The binary system is also called the base-2 system.

– Our decimal system is the base-10 system. It uses powers of 10 for each position in a number.

– Any integer quantity can be represented exactly using any base (or *radix*).

Positional Numbering Systems

• The decimal number 947 in powers of 10 is:

$$9 \times 10^2 + 4 \times 10^1 + 7 \times 10^0$$

• The decimal number 5836.47 in powers of 10 is:

$$5 \times 10^3 + 8 \times 10^2 + 3 \times 10^1 + 6 \times 10^0 + 4 \times 10^{-1} + 7 \times 10^{-2}$$

• The binary number 11001 in powers of 2 is:

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 16 + 8 + 0 + 0 + 1 = 25$$

• When the radix of a number is something other than 10, the base is denoted by a subscript.

Sometimes, the subscript 10 is added for emphasis:

$$(11001)_2 = (25)_{10}$$

1 × 2^ 4 + 1 × 2^ 3 + 0 × 2 ^2 + 0 × 2 ^1 + 1 × 2^ 0

= 16 + 8 + 0 + 0 + 1

= 25

• Because binary numbers are the basis for all data representation in digital computer systems, it is important that you become proficient with this radix system.

• Your knowledge of the binary numbering system will enable you to understand the operation of all computer components as well as the design of instruction set architectures.

• Every integer value can be represented exactly using any radix system.

• You can use either of two methods for radix conversion:

❖ **The subtraction method.**

❖ **The division remainder method.**

• Another method of converting integers from decimal to some other radix uses division.

• This method is mechanical and easy.

• It employs the idea that successive division by a base is equivalent to successive subtraction by powers of the base.

• Fractional values of other radix systems have nonzero digits to the right of the *radix point*.

• Numerals to the right of a radix point represent negative powers of the radix:

$$
\begin{aligned}
0.47_{10} &= 4 \times 10^{-1} + 7 \times 10^{-2} \\
0.11_{2} &= 1 \times 2^{-1} + 1 \times 2^{-2} \\
&= \tfrac{1}{2} + \tfrac{1}{4} \\
&= 0.5 + 0.25 = 0.75
\end{aligned}
$$

• As with whole-number conversions, you can use either of two methods:

a subtraction method and an easy multiplication method.

• We always start with the largest value first, *n* -1, where *n* is our radix, and work our way along using larger negative exponents.

• **Converting 0.8125 to binary.**

• Using the multiplication method to convert the decimal 0.8125 to binary, we multiply by the radix 2.

– The first product carries into the unit's place.

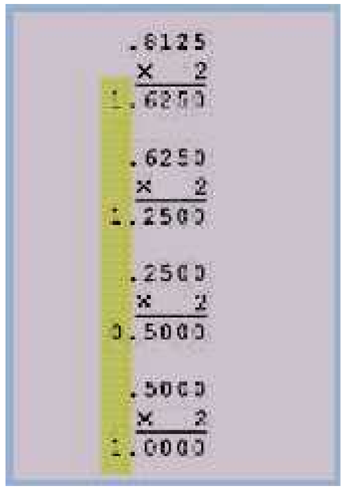
-Ignoring the value in the unit's place at each step, continue multiplying each fractional part by the radix.

```
  .8125
  x   2
 1.6250

  .6250
  x   2
 1.2500

  .2500
  x   2
 0.5000
```

- You are finished when the product is zero, or until you have reached the desired number of binary places.



– Our result, reading from top to bottom is:

$$(0.8125)_{10} = (0.1101)_2$$

This method also works with any base. Just use the target radix as the multiplier.

- The binary numbering system is the most important radix system for digital computers.
- However, it is difficult to read long strings of binary numbers and even a modestly-sized decimal number becomes a very long binary number.

For example:

$$(11010100011011)_2 = (13595)_{10}$$

- For compactness and ease of reading, binary values are usually expressed using the hexadecimal, or base-16, numbering system.
- The hexadecimal numbering system uses the numerals 0 through 9 and the letters A through F.
  - The decimal number 12 is $(B)_{16}$.
  - The decimal number 26 is $(1A)_{16}$.
- It is easy to convert between base 16 and base 2, because 16 = 2^4.
- Thus, to convert from binary to hexadecimal, all we need to do is group the binary digits into groups of four.

A group of four binary digits is called a hextet.

- Using groups of hextets, the binary number

$(11010100011011)_2 = (13595)_{10}$ in hexadecimal is:

0011	0101	0001	1011
3	5	1	B

Octal (base 8) values are derived from binary by using groups of three bits (8 = 2^3):

011	010	100	011	011
3	2	4	3	3

Octal was very useful when computers used six-bit words