



## Module 1: Information Representation I -- Number Systems

### Objectives:

- Learn how decimal numbers or numbers in base 10 are represented in other bases.
- Understand how data is organized.
- Understand the units use to measure the size of information and storage.
- Learn how to convert numbers in base 10 to other bases.
- Learn how unsigned (positive and zero) decimal numbers are represented in binary.

### Introduction

Before we focus on information representation, we need to know what a computer is and what it does. So, what is a **computer**? From:

1. The American Heritage® Dictionary: a **computer** is a device that computes, especially a programmable electronic machine that performs high-speed mathematical or logical operations or that assembles, **stores**, correlates, or otherwise **processes information**.
2. The Columbia Encyclopedia: a **computer** is a device capable of performing a series of arithmetic or logical operations. A **computer** is distinguished from a calculating machine, such as an electronic calculator, by being able to **store** a computer program (so that it can repeat its operations and make logical decisions), by the number and complexity of the operations it can perform, and by its ability to **process, store, and retrieve data** without human intervention.

In short, a computer is a device capable of **storing, retrieving, and processing information or data**. What types of information can a computer store/process? Information can roughly be categorized into 5 groups:

1. Numbers
2. Characters
3. Pictures
4. Sounds
5. Instructions or Computer Programs

This leads to a question how numbers, characters, pictures, sounds, or instructions can be **represented** such that they can be stored and processed by a computer. The answer is **binary numbers**. Therefore we will focus on **information representation** in a computer. Let us first explore how numbers information can be represented as binary numbers.

The number system that we accustom to and use in our everyday lives is decimal number system or base 10. On the other hand computers, by designed, store information as a sequence of 0s and 1s. The number system contains only 0 and 1 is called **binary number system** or base 2. So it is necessary for us to know how decimal numbers are represented in a computer system using

binary numbers. Before we find out how to represent decimal numbers using binary numbers, it is a good idea to explore numbers in different bases and understand how numbers in different bases are formed.

### Task 1: Base Number

As previously mentioned, we accustom to decimal numbers (base 10) but computers are designed to use binary numbers (base 2). In decimal or base 10 there are 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

**Activity 1.1:** How many digits are in *binary* or base 2? What are the digits?

**Activity 1.2:** How many digits are in *hexadecimal* or base 16? What are the digits? (Use A for 10, B for 11, and so on.)

**Activity 1.3:** In base 10 the next ten numbers after 9 are  $10_{10}$ ,  $11_{10}$ ,  $12_{10}$ ,  $13_{10}$ ,  $14_{10}$ ,  $15_{10}$ ,  $16_{10}$ ,  $17_{10}$ ,  $18_{10}$ , and  $19_{10}$ . Subscription 10,  $\square_{10}$ , indicates the number is decimal or in base 10. They are obtained from the combination of 1 and the digits 0 to 9. What are the next two numbers in base 2 after  $1_2$ ? **Observe that we use a subscription to denote the base that the number is in.**

**Activity 1.4:** What are the next two numbers in base 16 after  $F_{16}$  ( $15_{10}$ )?

**Activity 1.5:** What is the next number in base 2 after  $011_2$ ?

**Activity 1.6:** What is the next number in base 16 after  $0FF_{16}$ ?

**Activity 1.7:** Are the value  $10_{10}$  (10 in base 10), and  $10_2$  (10 in base 2) the same? Why or why not?

### Task 2: Data Organization

Computers store information as binary numbers, that is, a sequence of 0s and 1s. Since computers have limited size of storage, computer engineers need to have a way to measure the

size of information. The following terminology is used to measure information that represented in computers.

- *Bits* stand for **B**inary **d**igits. A **bit** is the fundamental unit of computer storage. Related bits are grouped to represent different types of information such as numbers, characters, pictures, sound, and instructions
- *Nibbles* – A **nibble** is a group of four bits. It is the size of a hexadecimal digit.
- *Bytes* – A **byte** is a group of 8 bits.
- *Words* – A **word** is a group of 16 bits or 2 bytes.

**Activity 2.1:** How many different combinations or numbers that 0s and 1s with 2 bits can form? (Hint: List all the possible combinations that binary numbers with 2 bits can form.)

**Activity 2.2:** How many different combinations or numbers that 0s and 1s with 3 bits can form?

**Activity 2.3:** How many different combinations or numbers that 0s and 1s with 4 bits can form?

**Activity 2.4:** In general, how many different combinations or numbers that 0s and 1s with  $n$  bits can form? That is to find a formula to determine how many numbers that  $n$  bits can form.

**Activity 2.5:** The size of storages or memory units of computers is usually measure in bytes. These units vary upon the size of storages. They can be measured as *kilobyte* (KB), *megabyte* (MB), *gigabyte* (GB), *terabyte* (TB) and *petabyte* (PB). A *kilobyte* is roughly one thousand bytes. But the actual number is 1,024 that is  $2^{10}$ . That means 10 bits can form exactly 1,024 binary numbers. We use kilobyte refer to 1,024 bytes because it is easy to reference. A megabyte is roughly one million bytes. What is the actual number of bytes in a megabyte? (Feel free to use any computation device.)

**Activity 2.6:** What is a gigabyte?

**Activity 2.7:** What is a terabyte?

### Task 3: Decimal Numbers

Let us first look at how to convert an unsigned or nonnegative binary number to a decimal number. If you recall from what you learned in elementary school how the position of each digit in a decimal number affects the value of the digit, this is called *positional notation*. For example,

$$2401_{10} = 2 \times 10^3 + 4 \times 10^2 + 0 \times 10^1 + 1 \times 10^0$$

Note that decimal numbers are number in base 10. We always start counting position from 0 and from right to left. Since 4 is in position 2, it has the value 400 which is 4 times 10 to the power of 2, that is  $4 \times 10^2$ . Similarly, since 1 is in position 0, it has the value 1 which is 1 times 10 to the power of 0. You can see that 10 refers to the base of the number 2401 that it represents, and 2 and 0 refer to the positions of digit 4 and 1, respectively.

An easier way to understand this is to view the position of digits and the corresponding positional values as a table as follows:

Position	...	4	3	2	1	0
Positional Value	...	$10^4$	$10^3$	$10^2$	$10^1$	$10^0$
	...	10000	1000	100	10	1
			2	4	0	1

$$2 \times 1000 + 4 \times 100 + 0 \times 10 + 1 \times 1 = 2401_{10}$$

Fortunately, we can use positional notation as the above example to convert binary numbers to decimal number or number in base 10.

For example, we want to convert a binary number  $10110_2$  to decimal. We, first, create a table that indicates positional values for base 2 to organize our conversion process by writing down the values of each position in ascending order from right to left. For example, in base 2, the positional values are: ... 256 128 64 32 16 8 4 2 1. These values are obtained as follows:

Position	...	8	7	6	5	4	3	2	1	0
Positional Value	...	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
	...	256	128	64	32	16	8	4	2	1

So, the table of positional values for base 2 looks as follows:

...	256	128	64	32	16	8	4	2	1

Then place each digit or bit of the given number from right to left. So, the table should look as follows:

...	256	128	64	32	16	8	4	2	1
					1	0	1	1	0

The table indicates the conversion of  $10110_2$  to decimal is

$$1 \times 16 + 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 = 16 + 4 + 2 = 22_{10}$$

Therefore  $10110_2 = 22_{10}$

Note that the converted decimal number is the sum of all positional values that have a corresponding 1 under them.

So, to convert a binary number to decimal number using position notation can be done as follows:

- 1) List enough positional values for base 2 from right to left.
- 2) Place each bit of a given binary number under each positional value from right to left.
- 3) The decimal number is the sum of all positional values that have a corresponding 1 under them.

You can use an application **BinHex** to practice changing values from binary (base 2) to decimal (base 10). The application **BinHex** can be downloaded from my webpage. Just follow the link: <http://cms.dt.uh.edu/faculty/ongards/links/links.php?id=1>



**Activity 3.1:** What decimal number does binary number  $110101_2$  represent?

**Activity 3.2:** What decimal number does  $2AC_{16}$  in base 16 represent? Follow the same process of converting a number in base 10 to base 2.

#### Task 4: Unsigned Binary Numbers

In Task 3 we use the positional notation to convert unsigned binary numbers to decimal numbers. In this task our focus is on representation of unsigned or nonnegative decimal numbers (integers) by binary numbers. In mathematics, a set of integers consists of negative integers, zero, and positive integers. So, in computer systems, there is a need to represent negative integers and positive integers including zero. Therefore we divide integers into *unsigned* and *signed* numbers. Unsigned numbers are only zero and positive integers. On the other hand, signed numbers are negative integers, zero, and positive integers or simply integers. So, we need number systems to represent these two groups of integers. There are four number systems that are used to represent integers: unsigned binary, signed and magnitude, 1's complement and 2's complement number systems. We will focus on only unsigned binary. **Note that unsigned binary is used to represent only positive integers and zero.**

There are two ways to convert a decimal number to a binary number.

1. Successive division by 2: Divide the given decimal number by 2 successively and collect the remainders in backward order. The process will be terminated if the quotient is 0.

For example: What is representation of  $79_{10}$  in binary?

$$\begin{array}{r}
 2 \overline{) 79} \\
 \underline{2 \phantom{0} 39} \phantom{0} 1 \\
 2 \phantom{0} \underline{19} \phantom{0} 1 \\
 2 \phantom{0} \phantom{0} \underline{9} \phantom{0} 1 \\
 2 \phantom{0} \phantom{0} \phantom{0} \underline{4} \phantom{0} 1 \\
 2 \phantom{0} \phantom{0} \phantom{0} \phantom{0} \underline{2} \phantom{0} 0 \\
 2 \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \underline{1} \phantom{0} 0 \\
 0 \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} 1
 \end{array}$$

Therefore  $79_{10} = 1001111_2$

2. Reverse positional notation: Use positional notation in reverse, that is, make use of the value in each position ( $2^0 = 1$ ,  $2^1 = 2$ ,  $2^2 = 4$ ,  $2^3 = 8$ ,  $2^4 = 16$ ,...) to determine which position is a 1. The following is the process of conversion:
- 1) Write down or create a table of the values of each position in ascending order from right to left. For example: ... **256 128 64 32 16 8 4 2 1** Make sure that the value of the left most number is larger than the decimal number to be converted. Note that the value **1** is  $2^0$ , **2** is  $2^1$ , **4** is  $2^2$ , **8** is  $2^3$ ,...
  - 2) Determine which position that has the largest value but less than or equal to the given decimal number and mark that position as a 1. For example, the number to be converted is **79**. Then the position **64** will be marked as a 1 since **79** is less than **128** but greater than **32**.
  - 3) Subtract that positional value from the given decimal number to obtain the remaining value for the next step. From the example in step 2, the positional value is **64**. Then the remaining value is  $79 - 64 = 15$
  - 4) Repeat the process with the value from the step 2. That means the position **8** will be marked as a **1**. Since **15** is less than **16** but greater than **8**, and the remaining value is  $15 - 8 = 7$
  - 5) The positions that are skipped will be mark as a 0. That means the position **32** and **16** will be marked as 0.
  - 6) The process will be terminated if the value in the third step is 0.

So, the number  $79_{10} = 1001111_2$

<b>64</b>	<b>32</b>	<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>
1	0	0	1	1	1	1

**Activity 4.1:** Convert the integer  $209_{10}$  to an unsigned binary using the successively division by 2.

**Activity 4.2:** Convert the integer  $209_{10}$  to an unsigned binary using the reversed-positional notation.

**Activity 4.3:** In a lot of cases, it is more convenient for programmers to deal with information in hexadecimal representations rather than binary representations. That means the programmers should know how to convert between binary and hexadecimal representation. Fortunately, the process of conversion from binary to hexadecimal and vice versa is relatively simple.

- To convert a binary representation to hexadecimal:
  1. Divide binary representation from right to left in to group of 4 bits (nibble).
  2. If the last group has less than 4 bits, pad 0s (to left) to make up 4 bits.
  3. Convert each group of 4 bits to a corresponding hexadecimal digit.

For example,  $1110100111011_2 = 0001\ 1101\ 0011\ 1011_2 = 1D3B_{16}$

- To convert a hexadecimal representation to binary, just convert each hex digit to corresponding binary representation.

For example,  $9C3F_{16} = 1001\ 1100\ 0011\ 1111_2$

The following table relates binary, decimal, and hexadecimal:

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

What is the Hex representation of the binary number  $1100101100_2$ ?

**Activity 4.4:** What is the binary representation of the hexadecimal number  $2A8DB9_{16}$ ?

---