



## I. PRINCIPLE OF A BASE

The base is the number used to define a number system.

- The base of the decimal system is ten "10".
- The Binary Base is two "2".
- That of the Octal system is eight "8".
- The base of Hexadecimnal is sixteen "16"

Whatever the numerical base used, it follows the following relationship:

$$\sum_{i=0}^{i=n} (b_i a^i) = b_n a^n + \dots + b_5 a^5 + b_4 a^4 + b_3 a^3 + b_2 a^2 + b_1 a^1 + b_0 a^0$$

where :  $b_i$  : digit of the base of rank  $i$   
 and :  $a^i$  : power of base  $a$  with exponent of rank  $i$   
 $b_0$  is called the least significant digit.  
 And  $b_n$  the most significant digit.

The notation  $( )_b$  indicates that the number is written in base  $b$ .

### *Exemple : base 10*

$$1986 = (1 \times 10^3) + (9 \times 10^2) + (8 \times 10^1) + (6 \times 10^0)$$

## I.1 THE DECIMAL SYSTEM

The decimal system is the one in which we are most accustomed to writing and using.

Each number can have ten "10" different values :

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, therefore, the decimal system has the base 10.

Any number written in the decimal system satisfies the following rule

$$745 = 7 \times 100 + 4 \times 10 + 5 \times 1$$

$$745 = 7 \times 10 \times 10 + 4 \times 10 + 5 \times 1$$

$$745 = 7 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$$

Each digit of the number, must be multiplied by a power of 10, this is what we call the weight of the digit.

The exponent of this power is zero for the digit located furthest to the right, and increases by one for each one-digit move to the left.

$$12435 = 1 \times 10^4 + 2 \times 10^3 + 4 \times 10^2 + 3 \times 10^1 + 5 \times 10^0 .$$

This way of writing numbers is called the positional number system.

In our conventional system, we use powers of 10 to weight the value of numbers according to their position, However, it is possible to imagine other number systems based on a different integer.

## I.2 THE BINARY SYSTEM

It is a numbering system using base two ‘2’. We commonly call bit in French (from the English binary digit, or ‘binary digit’, the digits of positional binary numbering. These can only take two values (0 and 1).

## I.3 THE OCTAL SYSTEM

The octal number system is the base 8 number system, and uses the numbers 0 to 7.

Thus, a number expressed in base 8 could be presented as follows :  $(745)_8$

When we write a number, we must clearly specify the base in which we express it to remove any possible indeterminations. ( $745$  also exists in base 10).

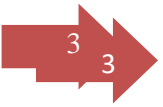
$(745)_8$ : Thus the number will be put in parentheses ( $745$  in our example) and subscripted with a number representing its base (8 is subscripted)

## I.4 THE HEXADECIMAL SYSTEM

It is a base 16 positional number system. It thus uses 16 symbols, the first ten digits (0,1,2,3,4,5,6,7,8,9) and the letters A to F for the following six:

|           |           |           |           |           |           |
|-----------|-----------|-----------|-----------|-----------|-----------|
| <b>10</b> | <b>11</b> | <b>12</b> | <b>13</b> | <b>14</b> | <b>15</b> |
| <b>A</b>  | <b>B</b>  | <b>C</b>  | <b>D</b>  | <b>E</b>  | <b>F</b>  |

The correspondence between base 2, base 10 and base 16 is indicated in the table below:



| Base 10 | Base 16 | Base 2 |
|---------|---------|--------|
| 0       | 0       | 0      |
| 1       | 1       | 1      |
| 2       | 2       | 10     |
| 3       | 3       | 11     |
| 4       | 4       | 100    |
| 5       | 5       | 101    |
| 6       | 6       | 110    |
| 7       | 7       | 111    |
| 8       | 8       | 1000   |
| 9       | 9       | 1001   |
| 10      | A       | 1010   |
| 11      | B       | 1011   |
| 12      | C       | 1100   |
| 13      | D       | 1101   |
| 14      | E       | 1110   |
| 15      | F       | 1111   |

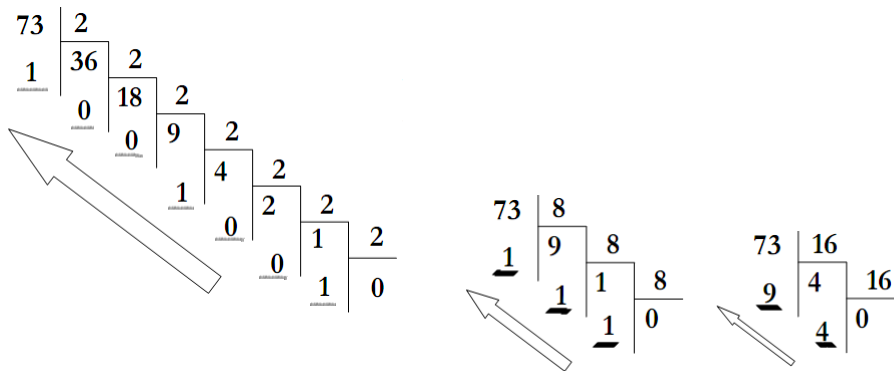
## II. CONVERSION FROM DECIMAL TO BASE B

The rule to follow is successive divisions :

- We divide the number by the base b
- Then the quotient by the base b
- So on until obtaining a quotient of zero or less than the base
- The sequence of remains corresponds to the symbols of the target base.
- We obtain the least significant figure first and the most significant figure last.



▪ *Exemple*



$$(73)_{10} = (1001001)_2 = (111)_8 = (49)_{16}$$

➤ **Conversion from base to the decimal base**

*Let N be a number represented in binary base by :  $N = (1010011101)_2$*

$$\begin{aligned} N &= (1 \cdot 2^9) + (0 \cdot 2^8) + (1 \cdot 2^7) + (0 \cdot 2^6) + (0 \cdot 2^5) + (1 \cdot 2^4) + (1 \cdot 2^3) + (1 \cdot 2^2) + (0 \cdot 2^1) + (1 \cdot 2^0) \\ &= 512 + 0 + 128 + 0 + 0 + 16 + 8 + 4 + 0 + 1 = (669)_{10} \end{aligned}$$

*Let N be a number represented in Octal base by :  $N = (251)_8$*

$$N = (2 \cdot 8^2) + (5 \cdot 8^1) + (1 \cdot 8^0) = 128 + 40 + 1 = (169)_8$$

*Let N be a number represented in hexadecimal base by :  $N = (1C9)_{16}$*

$$N = (1 \cdot 16^2) + (C \cdot 16^1) + (9 \cdot 16^0) = (1 \cdot 16^2) + (12 \cdot 16^1) + (9 \cdot 16^0) = 256 + 192 + 9 = (457)_{10}$$

### III. CODING FRACTIONAL NUMBERS

Coding a positif fractional number in base B:

$$(N)_B = a_n a_{n-1} a_{n-2} \dots a_1 a_0$$

To code a positive fractional number, add fractional part after a comma :

$$(N)_B = a_n a_{n-1} a_{n-2} \dots a_1 a_0 , b_1 b_2 \dots b_{m-1} b_m$$

The decimal value of such a number is given by calculating

$$a_n B^n + a_{n-1} B^{n-1} + \dots + a_1 B^1 + a_0 + b_1 B^{-1} + b_2 B^{-2} + \dots + b_{m-1} B^{-(m-1)} + b_m B^{-m}$$

**Exemple**

$$\blacklozenge 123,45 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

$$\begin{aligned} \blacklozenge (101,101)_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} \\ &\quad + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 4 + 1 + 0,5 + 0,125 = 5,625 \end{aligned}$$

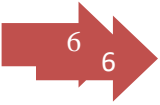
#### III.1 Conversion of a real decimal number to base B

**For the real part**

- Use the integer division method as for integers.

**For the fractional part**

- Multiply the fractional part by B .
- Note the integer part obtained.
- Repeat this operation with the fractional part of the result and so on.
- Stop when the fractional part is zero.



- Or when the desired precision is achieved.
  - Because we cannot always obtain a conversion to a finite number of digits for the fractional part.
- The fractional part in the base B is est la concatenation of the integer parts obtained in the order to their calculations.

**Exemple:** Conversion of  $(12.6875)_{10}$  in binary base.

◆ Conversion de 12 : donne  $(1100)_2$

◆ Conversion de 0,6875

$$\begin{array}{rcl}
 \text{◆ } 0,6875 \times 2 = 1,375 & = & \underline{1} + 0,375 \\
 0,375 \times 2 = 0,75 & = & \underline{0} + 0,75 \\
 0,75 \times 2 = 1,5 & = & \underline{1} + 0,5 \\
 0,5 \times 2 = 1 & = & \underline{1} + 0
 \end{array}$$

◆  $(12,6875)_{10} = (1100,1011)_2$

#### IV CONVERSION BETWEEN BASES.

In computer science, the binary, octal and hexadécimal bases are frequently used, all these bases being powers of 2,  $2^1$ ,  $2^3$ ,  $2^4$

There are some particularly simple conversions.

##### ➤ *Conversion Octal – Binary*

To convert a number from octal to binary, simply replace each digit constituting the octal number with its 3 digits binary equivalent.

#### Exemple

$$(742)_8 = (111\ 100\ 010)_2$$

$$(1\ 6\ 7\ 3)_8 = (001\ 110\ 111\ 011)_2 = (1\ 110\ 111\ 011)_2$$



➤ *Conversion Hexadecimal – Binary*

To convert a number from hexadecimal to binary, simply replace each digit constituting the hexadecimal number with its 4 digits binary equivalent.

**Exemple**

$$(A9)_{16} = (1010\ 1001\ 010)_2$$

$$(8B7F)_{16} = (1000\ 1011\ 0111\ 1111)_2$$

➤ *Conversion binary - Octal*

To convert a binary number to Octal, simply make groups of three digits (starting from the right).

**Exemple**

|              |                     |                         |                                 |
|--------------|---------------------|-------------------------|---------------------------------|
| Binary       | 1110010             | 11001100100             | 111100110010101011100101        |
| Grouping     | <b>001</b> /110/010 | <b>011</b> /001/100/100 | 111/100/110/010/101/011/100/101 |
| Pseudo Octal | 1 / 6 / 2           | 3 / 1 / 4 / 4           | 6 / 4 / 6 / 2 / 3 / 4 / 5       |
| Octal        | 162                 | 3144                    | 7462345                         |

➤ *Conversion hexadecimal - binary*

To convert a binary number to hexadecimal, simply make groups of four digitss (starting from the right).


**Exemple**

|                    |             |                |                                |
|--------------------|-------------|----------------|--------------------------------|
| Binary             | 110010      | 101001100100   | 110110110110101011100101       |
| Grouping           | 0011 / 0010 | 1010/0110/0100 | /1101/1011/0110/1010/1110/0101 |
| Pseudo hexadecimal | 3 / 2       | 10 / 6 / 4     | 13/11/6/10/14/5                |
| Hexadecimal        | 32          | A64            | DB6AE5                         |

## IV. ARITHMETICS OPERATIONS IN BASE B

- Arithmetics operations (addition, subtraction, multiplication, division) are carried out in any base B with the same rules in Base 10.
- A deduction or carryover appears when we reach or exceed the value B of the Base.

Arithmetic operations (addition, subtraction, multiplication, division) can be performed in any base B.

- With the same rules as for the base 10.
- Also detention (hold) but depending on the base.
- When we add two digits a and b in base B, if the sum of the decimal values a and b exceed or equal to B, then there is a detention.

### □ 1 The ADDITION

#### 1.1 Binary Addition :

- Adding two binary numbers is perfectly analogous to adding two decimal numbers.
- In fact, Binary addition is simpler since there are fewer cases to learn.
- We start by adding the digits of the lowest order, the digits of the second row are then added, and so on. The same rules apply to binary addition.







### 3. The Multiplication

#### 3-1 Binary Multiplication.

- We multiply Binary numbers in the same way as we multiply decimal numbers.
- The process is simpler because the digits of the multiplier are always 0 or 1, so we always multiply by 0 or 1.
- Binar multiplication is based on the following four rules :
  - $1 \times 1 = 1$
  - $1 \times 0 = 0$
  - $0 \times 1 = 0$
  - $0 \times 0 = 0$

#### Exemple

$$\begin{array}{r}
 \phantom{0000}1101 \\
 * \phantom{000}1011 \\
 \hline
 \phantom{0000}1101 \\
 \phantom{00}^11101. \\
 \phantom{00}^10000. \\
 \phantom{0}^1^1101.. \\
 \hline
 10001111
 \end{array}$$

#### 3.2 Multiplication in any base X (X=3,4,8,16,...)

- We multiply numbers in any base X in the same way as we multiply decimal numbers..
- When the multiplication of two digits equals or exceeds the base X, we convert the result to the base in which we work.



**Exemple:**

$$\begin{array}{r}
 11100111 \\
 - 101 \\
 \hline
 01000 \\
 - 101 \\
 \hline
 00111 \\
 - 101 \\
 \hline
 0101 \\
 - 101 \\
 \hline
 0001 \\
 - 000 \\
 \hline
 001
 \end{array}
 \quad
 \begin{array}{r}
 101 \\
 \hline
 101110
 \end{array}$$

**111100111 : 101 = 101110 reste 1.**