1. Definition

The number system or the numeral system is the manner of writing and expressing numbers by using digits or other symbols. The number of digits or symbols used to represent a number is called base ***b***.

2. System of base b

- **Representation**

In a number system of base ***b*** a number ***N*** is written as follows:

$$N= (d_n d_{n-1} \ldots\ldots d_2 d_1 d_0)_{\boldsymbol{b}}$$

where d_i $(0 \leq d_i \leq b-1)$ are the digits used in base ***b***. The value of the index ***i*** represents the weight of the digit in the number.

- d_0 is the least significant digit.
- d_n is the most significant digit.

Example: $N=(31786)_{10}$

6 is the least significant digit, 3 is the most significant digit, the weight of digit 7 is 2.

- **Polynomial form**

In a number system of base ***b*** a number ***N*** can be decomposed according to the integer powers of its base **b**. This decomposition is called the polynomial form given as follows:

$$d_n \, x b^n + d_{n-1} \, x b^{n-1} + \ldots\ldots\ldots + d_1 \, x b^1 + d_0 \, x b^0$$

Example*:* write the following numbers in polynomial form

$(6375)_{10} = 6x10^3 + 3x10^2 + 7\,x10^1 + 5\,x10^0$

$(76)_8 = 7\,x8^1 + 6\,x8^0$

$(1101)_2 = 1\,x2^3 + 1\,x2^2 + 0\,x2^1 + 1\,x2^0$

There are infinitely many number systems, but the four most common are: decimal, binary, octal and hexadecimal.

2.1 Decimal (base 10)

The decimal system is the most used by humans. The base ***b=10*** and the digits are ***{0,1,2,3,4,5,6,7,8,9}*** . By convention, if the number is not between parentheses, it is a decimal number.

Example: *175 is a decimal number*

2.2 Binary (base 2)

The number system used by computers. The base ***b=2*** and the digits are ***{0,1}***

Example: $(1011001)_2$ is a binary number

2.3 octal (base 8)

Used some time ago (UNIX system, old PDP computers). The base ***b=8*** and the digits are ***{0, 1, 2, 3, 4, 5, 6,7}***

Example: *(461)*$_8$ is an octal number.

2.4 Hexadecimal (base 16)

Hexadecimal system is widely used in computing, especially with low-level programming such as *assembly language*. The base ***b=16*** and the digits are ***{0, 1, 2, 3, 4, 5, 6,7, 8, 9}*** + letters ***{ A=10, B=11, C=12, D=13, E= 14, F=15}***

Example: *(1F4)*$_{16}$ is an hexadecimal number.

3. Conversions

3.1 Conversion (decimal → base b)

Apply the method of successive divisions:

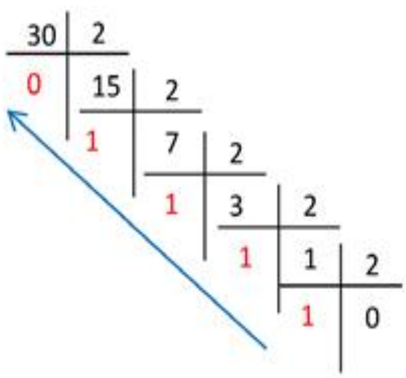
- Divide the number ***N*** by the base ***b*** and keep the remainder
- Divide the quotient obtained by the base ***b***
- Repeat the above steps until a zero quotient is obtained
- The result is obtained by writing (left to right) all the remainders, from bottom to top
- The first remainder corresponds to the *least significant digit* and the last to the *most significant digit*

Note:

In binary system we use the terms ***MSB*** (most significant bit) and ***LSB*** (least significant bit).

Examples :

- **(decimal → binary)**



30=(11110)$_2$

15=(1111)$_2$

- **(decimal → octal)**

25=(31)$_8$ 98=(142)$_8$

- **(decimal → hexa)**

29=(1D)$_{16}$ 79=(4F)$_{16}$

3.2 Conversion (base b → decimal)

The decimal value of a number N written in a base b, is obtained by its polynomial form described previously. It's equal to the sum of digits ***di*** times their power of ***b*** (b^i)

$$N= d_n \, x b^n + d_{n-1} \, x b^{n-1} + \ldots\ldots\ldots + d_1 \, x b^1 + d_0 \, x b^0$$

Examples: convert the following numbers into decimal

- **(binary → decimal)**

$(111101)_2 = 1 \, x 2^5 + 1 \, x 2^4 + 1 \, x 2^3 + 1 \, x 2^2 + 0 \, x 2^1 + 1 \, x 2^0 = 32+16+8+4+0+1=61$

$(10001)_2 = 17$

$(00010100)_2 = 20$

- **(octal → decimal)**

$(25)_8 = 21$ $\qquad$ $(133)_8 = 91$

- **(hexa → decimal)**

$(2E)_{16} = 2 \, x 16^1 + 14 \, x 16^0 = 32+14 = 46$ $\qquad$ $(15)_{16} = 21$

4. Fractional numbers conversion

A fractional number has two parts: whole number part (integer part) and decimal part (fractional part), these parts are separated by a dot (**.**) called the decimal(binary, octal…) point.

$$(d_n d_{n-1} \ldots\ldots d_1 d_0 \bullet d_{-1} d_{-2} \ldots\ldots d_{-m})_b$$

Example: 18.125 $\qquad$ $(1011.101)_2$ $\qquad$ $(35.14)_8$ $\qquad$ $(1B.7)_{16}$

4.1 Conversion base b →decimal

Use the polynomial form

$$d_n \, x b^n + d_{n-1} \, x b^{n-1} + \ldots\ldots + d_1 \, x b^1 + d_0 \, x b^{0} + d_{-1} \, \mathbf{x b^{-1}} + d_{-2} \, \mathbf{x b^{-2}} + \ldots\ldots + d_m \, \mathbf{x b^{-m}}$$

Example:

$(1011.01)_2 = 1 \, x 2^3 + 0 \, x 2^2 + 1 \, x 2^1 + 1 \, x 2^0 + 0 \, x 2^{-1} + 1 \, x 2^{-2} = 11 + 0.25 = 11.25$

$(76.4)_8 = 7 \, x 8^1 + 6 \, x 8^0 + 4 \, x 8^{-1} = 62 + 0.5 = 62.5$

$(2E.C)_8 = 2 \, x 16^1 + 14 \, x 16^0 + 12 \, x 16^{-1} = 46 + 0.75 = 46.75$

4.2 Conversion decimal→ base b

- Convert the integer part by successive divisions

- Multiply the fractional part by the base ***b***, iterate the multiplication on the fractional part of the obtained result until we get a fractional part equals to zero.

- Write out (left to right) the integer parts from the results of each multiplication taken from top to bottom.

***Example*:**

$9.125 = (1001.001)_2$

$0.125 \times 2 = \mathbf{0}.25$
$0.25 \times 2 = \mathbf{0}.5$
$0.5 \times 2 = \mathbf{1}$

$15.25 = (17.2)_8$

$26.75 = (1A.C)_{16}$

Note:

The conversion of fractional numbers is not always performed accurately (losing some precision).

Example: $17.8 = (10001.11001100\ldots..)_2$

5. Conversion from base *p* to base *q*

To convert a number N written in base p, into a base q, we pass through the intermediate base 10. But if p and q are written respectively in the form of a power of 2 (4, 8, 16, 32,….) we can go through the base 2 (easier and faster).

$\mathbf{8=2^3}$ each octal digit is represented on **3 bits**

$\mathbf{16=2^4}$ each hexadecimal digit is represented on **4 bits**

Table 2.1 : *Octal digits in Binary*

Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Table 2.2 : *Hexadecimal digits in Binary*

Hexadecimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

- **(octal → binary)**

Write each octal digit in its 3-bit binary equivalent according to Table 2.1

***Example*:** $(56)_8 = (101110)_2$

$(47.34)_8 = (100111.0111)_2$

- **(binary → octal)**

Grouping the bits (each group composed of 3 bits) from right to left for the integer part, and from left to right for the fractional part. Then substitute these groups according to table 2.1.

Example: $(100101)_2 = (45)_8$

$(10110)_2 = (26)_8$

$(110101.101011)_2 = (65.53)_8$

- **(hexa → binary)**

Write each hexadecimal digit in its 4-bit binary equivalent according to Table 2.2

Example: $(E6)_{16} = (11100110)_2$

$(1E.AB)_{16} = (11110.10101011)_2$

- **(binary → hexa)**

Grouping the bits (each group composed of 4 bits) from right to left for the integer part, and from left to right for the fractional part. Then substitute these groups according to table 2.2.

Example: $(1000101)_2 = (45)_{16}$

$(101110.10001001)_2 = (2E.89)_{16}$

- **(octal → hexa) / (hexa → octal)**

Use the intermediate base 10 or the intermediate base 2 (faster)

Example: $(54)_8 = (101100)_2 = (2C)_{16}$

$(2,74)_8 = (10.111100)_2 = (2.F)_{16}$

$(3A)_{16} = (111010)_2 = (72)_8$

$(F.A2)_{16} = (1111.10100010)_2 = (17.504)_8$

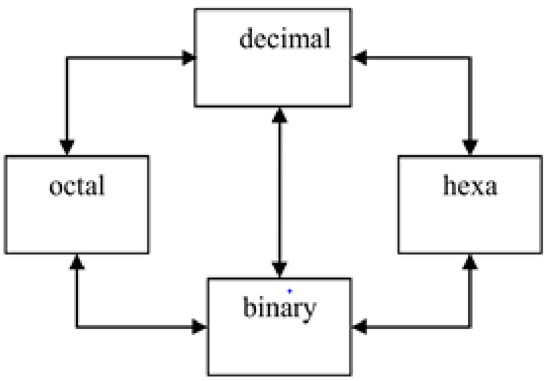


***Figure 2.1:** General conversion scheme*

Note :

It is very difficult to read and write a long binary numbers, we often used two more compact notations: octal or hexadecimal.

6. Arithmetic Operations

6.1 Binary Arithmetic

- ***Binary Addition***

0+0=0

0+1=1

1+0=1

1+1=0 with carry =1

Example: $(10100)_2$ + $(1101)_2$

$$
\begin{array}{cccccccc}
 & {\scriptstyle 1} & {\scriptstyle 1} & & & & & \\
 & & 1 & 0 & 1 & 0 & 0 \\
+ & & & 1 & 1 & 0 & 1 \\
\hline
 & 1 & 0 & 0 & 0 & 0 & 1
\end{array}
$$

- ***Binary Subtraction***

0-0=0

0-1=1 with borrow = 1

1-0=1

1-1=0

Example: $(1000)_2$ - $(10)_2$

$$
\begin{array}{ccccc}
 & 1 & 0 & 0 & 0 \\
- & {\scriptstyle 1} & {\scriptstyle 1} & 1 & 0 \\
\hline
 & 0 & 1 & 1 & 0
\end{array}
$$

- ***Binary Multiplication***

Binary multiplication is similar to decimal multiplication. It is simpler than decimal multiplication because only 0s and 1s are involved.

- ***Binary Division***

Binary division is similar to decimal division.

6.2 Octal Arithmetic

- ***Octal Addition***

Following octal addition table will help you to handle octal addition.

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

A

Sum

B

To use this table, simply follow the directions used in this example: Add 6_8 and 5_8. Locate 6 in the A column then locate the 5 in the B column. The point in 'sum' area where these two columns intersect is the 'sum' of two numbers.

$$6_8 + 5_8 = 13_8.$$

Example − Addition

$456_8 + 123_8 = 601_8$

```
 1 1      carry
 4 5 6   = 302₁₀
+1 2 3   =  83₁₀
--------
 6 0 1   = 385₁₀
```

- ***Octal Subtraction***

The subtraction of octal numbers follows the same rules as the subtraction of numbers in any other number system. The only variation is in borrowed number. In the decimal system, you borrow a group of 10_{10}. In the binary system, you borrow a group of 2_{10}. In the octal system you **borrow** a group of $\mathbf{8}_{10}$.

Example – Subtraction

Example:

$456_8 - 173_8 = 333_8$

```
   8       borrow
 ³4 5 6   = 302₁₀
 -1 7 3   = 123₁₀
--------
  2 6 3   = 179₁₀
```

6.3 Hexadecimal Arithmetic

- ***Hexadecimal Addition***

Following hexadecimal addition table will help you greatly to handle Hexadecimal addition.

X

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Sum

Y

To use this table, simply follow the directions used in this example: Add A_{16} and 5_{16}. Locate A in the X column then locate the 5 in the Y column. The point in 'sum' area where these two columns intersect is the sum of two numbers.

$$A_{16} + 5_{16} = F_{16}.$$

Example − Addition

$4A6_{16} + 1B3_{16} = 659_{16}$

```
  1        carry
  4 A 6  = 1190₁₀
+ 1 B 3  =  435₁₀
---------
  6 5 9  = 1625₁₀
```

- ***Hexadecimal Subtraction***

The subtraction of hexadecimal numbers follow the same rules as the subtraction of numbers in any other number system. The only variation is in borrowed number. In the decimal system, you borrow a group of 10_{10}. In the binary system, you borrow a group of 2_{10}. In the hexadecimal system you **borrow** a group of **16_{10}**.

Example - Subtraction

$4A6_{16} - 1B3_{16} = 2F3_{16}$

```
    16       borrow
  ³4 A 6  = 1190₁₀
  - 1 B 3  =  435₁₀
---------
   2 F 3  =  755₁₀
```