CHAPTER 1
Decimal System (Base 10)

This is our basic system in which quantities large or small can be represented by use of symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 together with appropriate place values according to their positions.

Example :

	1	2	3	4 •	5	6	7
place →	10^3	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}
value →	1000	100	10	1	$\frac{1}{10}$	$\frac{1}{100}$	$\frac{1}{1000}$
Therefore	1	2	3	4 •	5	6	7
=	1×1000	2×100	3×10	4×1	$5 \times \frac{1}{10}$	$6 \times \frac{1}{100}$	$7 \times \frac{1}{1000}$
digit value → =	1000 +	200 +	30 +	4 +	0.5 +	0.06 +	0.0007
=	1234.567						

Binary system (base 2) ....

The only symbols used in binary system are 0 and 1 and the place values are powers of 2, i.e. the system has a base of **2**.The digits 0 and 1 are called **bits**, which is short for binary digits.

Converting a binary number to decimal number

<u>Example:</u>

Represent 110101_2 in decimal notation.

$$110101_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= 32 + 16 + 4 + 1$$

$$= 53_{10}$$

Converting a decimal number to binary number

The simplest way to do this is by repeated division by 2, noting the reminder at each stage. Continue dividing until a final zero quotient is obtained.

<u>Example:</u>

Represent 209 in decimal notation.

2	209	
2	104	1
2	52	0
2	26	0
2	13	0
2	6	1
2	3	0
2	1	1
	0	1

$\therefore\ 209_{10} = 11010001_2$

Converting a decimal fraction to binary fraction

<u>Example:</u>

Find the binary form of 5.578125.

Break the numbers into an integer and a fraction part. We find that
$5_{10} = 101_2$.

For the fractional part, $x_1 = x = 0.578125$, use the above algorithm

$$
\begin{array}{lll}
2x_1 = \underline{1}.15625 & x_2 = 0.15625 & a_1 = 1 \\
2x_2 = \underline{0}.3125 & x_3 = 0.3125 & a_2 = 0 \\
2x_3 = \underline{0}.625 & x_4 = 0.625 & a_3 = 0 \\
2x_4 = \underline{1}.25 & x_5 = 0.25 & a_4 = 1 \\
2x_5 = \underline{0}.5 & x_6 = 0.5 & a_5 = 0 \\
2x_6 = \underline{1}.0 & x_7 = 0. & a_6 = 1
\end{array}
\downarrow
$$

$\therefore\ 5.578125_{10} = 101.100101_2$

Addition in binary notation

<u>Example :</u>

Add 1101_2 and 111_2 using binary notation.

carry row

$$
\begin{array}{rccccc}
 & & 1 & 1 & 1 & \\
 & & 1 & 1 & 0 & 1_2 \\
+ & & & 1 & 1 & 1_2 \\
\hline
 & 1 & 0 & 1 & 0 & 0_2 \\
\hline
\end{array}
$$

Subtraction in binary notation

<u>Example:</u>

Subtract 1011_2 from 11000_2 using binary notation.

borrow row

$$
\begin{array}{rccccc}
 & & 2 & 1 & & \\
 & 0 & 0 & 2 & 2 & \\
 & 1 & 1 & 0 & 0 & 1_2 \\
- & & 1 & 1 & 1 & 1_2 \\
\hline
 & & 1 & 0 & 1 & 0_2 \\
\hline
\end{array}
$$

Multiplication in binary notation

<u>Example:</u>

$$
\begin{array}{rcccccc}
 & & & & 1 & 1 & 1_2 \\
\times & & & & 1 & 1 & 0_2 \\
\hline
 & & & & 0 & 0 & 0 \\
 & & & 1 & 1 & 1 & \\
 & & 1 & 1 & 1 & & \\
\hline
 & 1 & 0 & 1 & 0 & 1 & 0_2 \\
\hline
\end{array}
$$

Octal system (base 8) ....

This system uses the symbols 0,1,2,3,4,5,6,7 with place values that are powers of 8.

<u>Example:</u>

$$357.321_8 = 3 \times 8^2 + 5 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1} + 2 \times 8^{-2} + 1 \times 8^{-3}$$

$$= 3(64) + 5(8) + 7(1) + 3\left(\frac{1}{8}\right) + 2\left(\frac{1}{64}\right) + 1\left(\frac{1}{512}\right)$$

$$= 192 + 40 + 7 + \frac{3}{8} + \frac{2}{32} + \frac{1}{512}$$

$$= 239.4082_{10}$$

1.3.1 Converting a decimal number to a octal number

The way to do this is by repeated division by 8, noting the reminder at each stage. Continue dividing until a final zero quotient is obtained.

<u>Example:</u>

Represent 209 in octal notation.

8	209	
8	26	1
8	3	2
	0	3

$\therefore\ 209_{10} = 321_8$

Hexadecimal system (base 16) ....

This system has computer applications. The symbols here need to go up to an equivalent denary value of 15, so, after 9, letters of the alphabet are used as follows:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E ,F

where A, B, C, D, E, and F represent 10, 11, 12, 13, 14, and 15 respectively.

Example:

$$2A7.3E2 = 2\times16^{2} + 10\times16^{1} + 7\times16^{0} + 3\times16^{-1} + 14\times16^{-2} + 2\times16^{-3}$$

$$= 2\times256 + 10\times16 + 7\times1 + 3\times\frac{1}{16} + 14\times\frac{1}{256} + 2\times\frac{1}{4096}$$

$$= 679.243_{10}$$

Floating-point numbers ....

Fractional quantities are typically represented in computers using floating-point form. In this approach, the decimal floating-point representation of a number x is basically that given as

$$x = \pm m \times 10^n$$

where m is called the mantissa, $0.1 \le m < 1$

n is called the exponent, n is an interger.

Example:

A computer with a four decimal digits floating-point arithmetic means that the number of digits in $\overline{X}$ is limited to four. Hence, all the numbers are in the form

$$(\pm 0.xxxx)10^n$$

for example

$$1 \rightarrow +0.1000 \times 10^1$$

$$-0.005 \rightarrow -0.5000 \times 10^{-2}$$

$$-\frac{1}{3} \rightarrow -0.3333 \times 10^0$$

$$\pi \rightarrow +0.3142 \times 10^1$$