

FYBSc: Electronic Science: Paper I: Fundamental of Digital Electronics

UNIT1: Basics of Digital Electronics

Introduction to Digital Electronics

- Electronic circuits are classified into two broad categories; Digital and Analog. Digital electronics involves quantities with discrete values and analog electronics involves quantities with continuous values. The word digital comes from the source word digit and digitus (the Latin word for finger), as fingers are used for discrete counting.

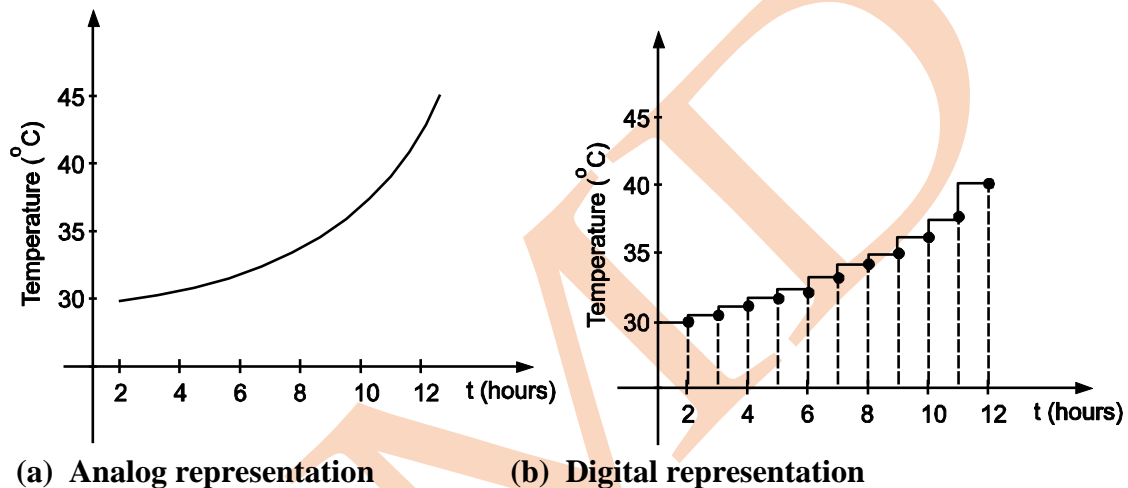


Fig. 1

An analog quantity is one having continuous value. A digital quantity is one having a discrete set of values. Most of the physical quantities appearing in nature are analog. For example, temperature, pressure of atmosphere. If we consider a temperature of a given day, it does not go from 40°C to 41°C, rather it takes all infinite values in between. However, instead of recording temperature on a continuous basis, if we take temperature reading after every hour we will get discrete values of the temperature. In Fig. 1.(a) the temperature is recorded continuously and as observed from it the temperature changes smoothly from 35°C to 45°C whereas in Fig. 1 (b) the temperature is recorded after an hour and it shows discrete values.

Advantages of Digital Data Over Analog Data:

It is found that digital data can be processed and transmitted more efficiently and reliably than analog data. Digital data can be stored more compactly and reproduced with greater accuracy and clarity than when it is in analog form. Besides this, some of

the more important reason for the widespread use of digital techniques and systems are:

- The basic building block used in digital circuits operate in one of the two states, known as ON and OFF which results in a very simple operation.
- There are only a few basic operations in digital circuits which are very easy to understand.
- The logic involved in digital techniques is based on Boolean algebra which is quite simple to understand.
- The large number of ready to use electronic circuits in the form of single chip known as IC's are available for performing various operations. They are highly reliable, accurate, small in size and the speed of operation is very high.
- Digital circuits are highly suitable for computers, calculators, telephone etc.
- The display of digital data is very convenient, and elegant.

Digital Systems/Electronics contains circuits and systems in which there are only two possible states.

These states are represented by two different voltage levels, usually referred to as High level and Low level.

The two states can also be represented by switch with open and closed state or with lamp in the ON and OFF state. Such two state systems are called a *binary system*.

The binary system uses symbols 1 and 0 called as *binary bits*, which is a contraction of the word's binary digit. In digital circuit, two different voltage levels are used to represent the two bits. Normally, 1 is represented by the higher voltage which is referred as high and a 0 is represented by the lower voltage level which is referred as low. The voltages used to represent a 1 and a 0 are called logic levels. In general, a voltage range is specified or assigned to represent logic levels rather than assigning a single voltage. Fig. 2 shows digital single representation.

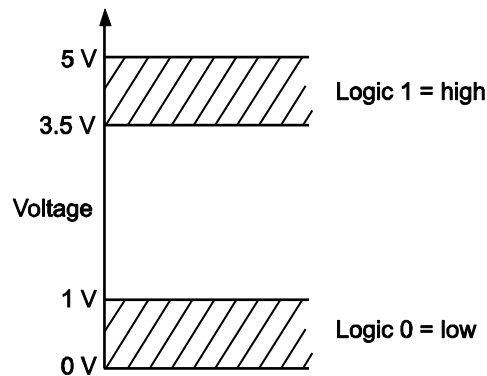


Fig. 2 : Voltage ranges for logic 1 and 0

As shown in Fig. 2, in a practical digital circuit a High can be 'any voltage between a specified minimum value and a specified maximum value', similarly a Low can be 'any voltage between a specified minimum and a specified maximum value'.

Introduction to Number Systems

After hearing the word number we immediately think of the most familiar decimal number system with its 10 digits 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. However, it is only customary and not necessary to use these symbols. Basically a *number system* is 'an ordered set of symbols known as digits', and the total number of symbols used in a system defines the type of number system. Some of the commonly used number systems are :

- 1] binary
- 2]octal
- 3] Decimal
- 4] hexadecimal number system.

Each number system has predefined rules for performing arithmetic operations like addition, subtraction, multiplication etc.

Decimal Number System

In decimal number system, an ordered set of ten symbols 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 are used to specify the quantities.

The symbols used are known as digits.

Since deci-stands for ten, decimal number system uses ten basic symbols.

The number of different basic symbols used in a system is known as *base* or *radix* of a number system.

Since decimal number system has 10 basic symbols, the base or **radix of this number system is 10.**

The **radix of binary system is 2.**

Octal system radix or base is 8 and that of **hexadecimal is radix or base 16.**

Table 1.1 shows the representation of different quantities in decimal number system, where black dots are used to represent the quantities.

We can use decimal digit 0 through 9 to represent quantities upto 9.

In order to represent 10 quantities, the decimal 10 is obtained by combining the second digit followed by the first.

Table 1.1: Decimal digits

Quantity	Symbol
None	0
•	1
•	2
••	3
•••	4
••••	5
•••••	6
••••••	7
•••••••	8
••••••••	9
•••••••••	10
••••••••••	11

Since decimal number system uses 10 symbols, we can express decimal number (integer or a whole number) in units, tens, hundreds and so on.

For example, the decimal number 298 decomposes into,

$$298 = 200 + 90 + 8$$

In powers of 10, this becomes

$$298 = 2 \times 10^2 + 9 \times 10^1 + 8 \times 10^0$$

Thus, the decimal number can be expressed as sum of powers of ten and in above example, the digit 2 has weight 200, the digit 9 has weight 90 and the digit 8 has weight 8.

As we go from right to the left, the weight also known as place value of decimal digit increases in terms of powers of 10. However, the decimal number may have an integer and a fractional part.

The integer and the fractional part are separated by a point known as the decimal point

e.g. decimal number 52.38, where 52 is an integer part and 38 is the fractional part of the given number. The weights assigned in case of mixed number are

$$\leftarrow 10^3 \ 10^2 \ 10^1 \ 10^0 \cdot 10^{-1} \ 10^{-2} \ 10^{-3}$$

Decimal point

Hence, the number 52.38 can be expressed as,

$$\begin{aligned} 52.38 &= 5 \times 10^1 + 2 \times 10^0 + 3 \times 10^{-1} + 8 \times 10^{-2} \\ &= 50 + 2 + \frac{3}{10} + \frac{8}{100} \\ &= 50 + 2 + 0.3 + 0.08 \\ &= 52.38 \end{aligned}$$

These systems are widely used in computers, logic circuits, microprocessors, etc.

Binary Number System

Decimal number system is not suitable for numerical operation and computation using electronic machine like computers, microprocessors etc. It is **very much essential to convert decimal numbers to binary numbers** that will solve this problem.

The **main advantage** of using binary number system is that, it minimises the number of lines in a computer system.

The **binary number system is a code that uses only two basic symbols i.e. 0 and 1.**

In digital electronics, '0' represents the low state and '1' the high state. **These symbols are known as bits.**

These states 1 and 0 are also referred as ON and OFF, HIGH or LOW, magnetised and demagnetised, white or black etc.

The binary number system has a base of two because the only two digits 0 and 1 are used.

All binary numbers consist of a string of 0s and 1s.

Examples are 10, 101, 1001 and which are read as one zero, one zero one, one zero zero one, to avoid confusion with decimal numbers.

Table 1.2 : Binary number system

Decimal number	Binary number
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

As we see from the table, only **two symbols are available**.

The group of four bits is known as nibble and group of eight bits is known as byte.
e.g. 1001, 1110 etc. are **nibbles** and 10101011, such group is called as **byte**.

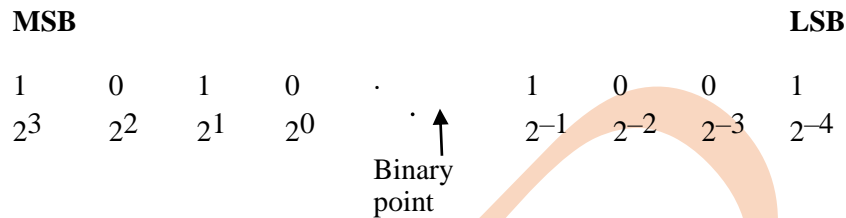
A binary number can be represented as,

$$N = d_n \times 2^n + d_{n-1} \times 2^{n-1} + \dots + d_1 \times 2^1 + d_0 \times 2^0 + d_{-1} \times 2^{-1} + \dots + d_{-m} \times 2^{-m}$$

where $d_n, d_{n-1} \dots d_{-(m-1)}, d_{-m}$ are binary symbols (0 or 1) and n, m are integers.

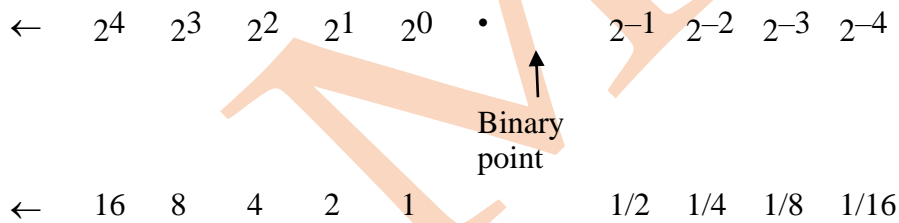
The decimal system binary system is also positionally weighted. However, the position value of each bit corresponds to some power of 2. In each binary number, the value increases by power of 2 starting with 0 to left of binary point and decreases to the right of the binary point starting with power of -1 . **The left most bit is known as Most Significant Bit (MSB) i. e. highest weight**
The right most bit is known as Least Significant Bit (LSB) i.e. lowest weight.

The position value (or weight) of each bit along with 8 bit 1010.1001 is shown below:



As seen from the above example, fourth bit to the left of binary point carries the maximum weight (i.e. has the highest value) and is called Most Significant Bit (MSB) or most significant digit. Similarly, the fourth bit to the right of the binary point is called Least Significant Bit (LSB) or least significant digit.

The position value of different bits are given by ascending power of 2 to the left of binary point and descending power of 2 to the right of the binary point. The different digit position in general form may be stated as given below :



Position value

If 2 bits are used, the highest count is upto 3.

If 3 bits are used it is upto 7,

for 4 bit highest count is 15, means in general we can say,

Highest decimal count = $2^n - 1$, where n is number of bits.

If $n = 5$, i.e. with 5 bits we can count from 0 through 31.

$$2^5 - 1 = 32 - 1$$

$$= 31 \text{ is highest count}$$

Hexadecimal Number System

Hexadecimal number system has **base of sixteen**.

It is extensively used in microprocessor work.

They are much shorter than binary numbers.

This makes them easy to write and remember.

Hexadecimal number represents **group of four bit binary numbers**.

It uses **16 distinct symbols 0 to 9 and A to F**.

Thus the symbols in hexadecimal number systems are,
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

A **subscript 16 or H** is used which indicates a hexadecimal number.

e.g. 42H or $(42)_{16}$
9CH or $(9C)_{16}$

Table : Hexadecimal number system

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

The hexadecimal number after F can be read as,

10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F,
20, 21, 29, 2A, 2B, 2C, 2D, 2E, 2F,
30, 31 so on ...

I.Binary to Decimal Conversion

Any binary number can be converted into its equivalent decimal number using the weights assigned to each bit position.

The rightmost bit (LSB) in a binary number has weight of $2^0 = 1$.

The weights increase by power of two for each bit from right to left.

e.g. Binary number 111 becomes,

Binary number : 1 1 1
Binary weight : 2^2 2^1 2^0
 $4 + 2 + 1 = 7$
111 = 7

Example 1: Convert the binary number 1111 to decimal.

Solution :

$$\begin{array}{r} \text{Binary number} \quad 1 \ 1 \ 1 \ 1 \\ \text{Binary weight} \quad 2^3 \ 2^2 \ 2^1 \ 2^0 \\ \text{Weight value} \quad 8 \ 4 \ 2 \ 1 \\ \\ = 1 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1 \\ = 8 + 4 + 2 + 1 \\ = (15)_D \text{ or } (15)_{10} \end{array}$$

It can be represented as

$$\begin{array}{l} (1111)_B = (15)_D \\ \text{or} \quad (1111)_2 = (15)_{10} \end{array}$$

The subscript **D** or **10** identifies decimal number and subscript **B** or **2** identifies binary number.

Example 2: Convert the binary number 110101 to decimal number.

Solution :

$$\begin{array}{r} \text{Binary number} \quad 1 \ 1 \ 0 \ 1 \ 0 \ 1 \\ \text{Binary weight} \quad 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \\ \text{Weight value} \quad 32 \ 16 \ 8 \ 4 \ 2 \ 1 \\ \\ = 1 \times 32 + 1 \times 16 + 0 + 1 \times 4 + 0 + 1 \times 1 \\ = 32 + 16 + 4 + 1 = (53)_{10} \end{array}$$

We can also use streamlined method by the following procedure,

$$\begin{array}{l} \text{Step 1} \quad 1 \ 1 \ 0 \ 1 \ 0 \ 1 \\ \text{Step 2} \quad 32 \ 16 \ 8 \ 4 \ 2 \ 1 \\ \text{Step 3} \quad 32 \ 16 \ 8 \ 4 \ 2 \ 1 \end{array}$$

In step 3 if zero appears in digit position, cross out the decimal weight for that position.

$$\begin{array}{l} \text{Step 4 :} \quad 32 + 16 + 4 + 1 = (53)_{10} \\ (110101)_2 = (53)_{10} \\ \text{or} \quad (110101)_B = (53)_D \end{array}$$

The binary numbers we have seen so far have been whole numbers. Fractional number can also be represented in a binary by placing bits to the right of binary point.

In general, binary number with fractions can be written as

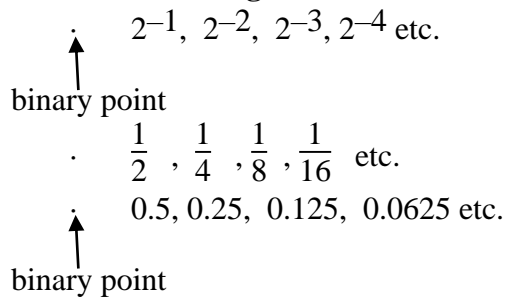
$$2^n \dots 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \ . \ 2^{-1} \ 2^{-2} \ 2^{-3} \ \dots \ 2^{-n}$$

↑

binary point

This indicates that **all the bits to the left of binary point have weights that are positive power of two.**

All bits to the right of binary point have weights that are negative power at two or fractional weight.



Example 3: Determine the decimal number represented by binary fraction 0.101.

Solution: First determine the weight of each bit and then sum the weight times the bit.

Binary weight	2^{-1}	2^{-2}	2^{-3}
Weight value	0.5	0.25	0.125
Binary number	0.1	0	1
	$= 1 \times 0.5 + 0 \times 0.25 + 1 \times 0.125$ $= 0.5 + 0 + 0.125$ $= 0.625$		

Example 4 : Determine the decimal value of the binary number 0.1011.

Solution :

Binary weight	2^{-1}	2^{-2}	2^{-3}	2^{-4}
Weight value	0.5	0.25	0.125	0.0625
Binary value	0.1	0	1	1
	$= 1 \times 0.5 + 0 \times 0.25 + 1 \times 0.125 + 1 \times 0.0625$ $= 0.5 + 0 + 0.125 + 0.0625$ $= 0.5 + 0.125 + 0.0625$ $= (0.6875)_{10}$			

Example 5 : Perform the following : $(11010.010)_2 = (?)_{10}$.

Solution : $11010.010 \leftarrow$ given number

Step :

- (i) Write the weight 16 8 4 2 1 0.5 0.125 0.0625
 - (ii) Discard the weight whose coefficients are zero
 $16 \ 8 \ 4 \ 2 \ 1 \ 0.5 \ 0.125 \ 0.0625$
 - (iii) Add the remaining weight $16 + 8 + 2 + 0.125 = 26.125$
- $\therefore (11010.010)_2 = (26.125)_{10}$

2.Decimal to Binary Conversion

Any decimal number can be converted into its equivalent binary number.

One way to convert a decimal number into binary equivalent is reverse process of binary to decimal conversion.

e.g. 9 can be written in the power of 2 as,

$$\begin{aligned}
 9 &= 2^3 + 2^2 + 2^1 + 2^0 \\
 &= 8 + 0 + 0 + 1 \\
 &= 1001
 \end{aligned}$$

Or, number 23 can be written as,

$$\begin{aligned}
 23 &= 2^4 + 2^3 + 2^2 + 2^1 + 2^0 \\
 &= 16 + 0 + 4 + 2 + 1 \\
 &= 10111
 \end{aligned}$$

A popular way to convert decimal to binary numbers is *double-dabble method* or *divide by two method*.

In this method, progressively divide the decimal number by 2 and write down the remainders after each division.

These remainders are taken in reverse order i.e. from bottom to top form the required binary number.

Example 1 : Convert the decimal number $(25)_{10}$ into equivalent binary number.

Solution :

	Quotient	Remainder
$\frac{25}{2}$	12	1 LSB
$\frac{12}{2}$	6	0
$\frac{6}{2}$	3	0
$\frac{3}{2}$	1	1
$\frac{1}{2}$	0	1 MSB

↑
Read

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

It may also be put in the following form :

$$\begin{aligned}
 25 \div 2 &= 12 + 1 \\
 12 \div 2 &= 6 + 0 \\
 6 \div 2 &= 3 + 0 \\
 3 \div 2 &= 1 + 1 \\
 1 \div 2 &= 0 + 1
 \end{aligned}$$

1 1 0 0 1

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

Example 2 : Convert the decimal number $(45)_{10}$ into equivalent binary number.

Solution :

	Quotient	Remainder
$\frac{45}{2}$	22	1 (LSB)
$\frac{22}{2}$	11	0
$\frac{11}{2}$	5	1
$\frac{5}{2}$	2	1
$\frac{2}{2}$	1	0
$\frac{1}{2}$	0	1 (MSB)

$$(45)_{10} = (101101)_2$$

3.Decimal Fractions to Binary Conversion

In this case, **multiply by two rule is used.**

We multiply bit by two and record the carry in the integer position, these are taken in the forward direction (top to bottom) i.e. top carry is MSB and bottom is LSB.

Example 1: Convert a decimal fraction 0.8125 into binary equivalent.

Solution:

$0.8125 \times 2 = 1.625 = 0.625$ with carry	1	MSB
$0.625 \times 2 = 1.25 = 0.25$ with carry	1	Read ↓ LSB
$0.25 \times 2 = 0.50 = 0.5$ with carry	0	
$0.5 \times 2 = 1.0 = 0$ with carry	1	

Multiply the decimal fraction by 2 until the fractional product is zero.

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

Example 2: Convert decimal number 25.6 into binary number.

Solution: First split the decimal number into an integer 25 and fraction of 0.6 and apply double-dabble to each part.

Quotient Remainder

$\frac{25}{2}$	12	1	LSB	
$\frac{12}{2}$	6	0	↑	
$\frac{6}{2}$	3	0	↑	Read
$\frac{3}{2}$	1	1	↑	
$\frac{1}{2}$	0	1	MSB	

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

and

$0.6 \times 2 = 1.2 = 0.2$	with carry	1	MSB	
$0.2 \times 2 = 0.4 = 0.4$	with carry	0	↓	
$0.4 \times 2 = 0.8 = 0.8$	with carry	0	↓	Read
$0.8 \times 2 = 1.6 = 0.6$	with carry	1	↓	
$0.6 \times 2 = 1.2 = 0.2$	with carry	1	LSB	

$(0.6)_{10} = (10011)_2$

In the fraction conversion, we stop the conversion process after getting five binary digits because in the last step, we had got the repetition of initial number i.e. 0.2.

$(25.6)_{10} = (11001 \cdot 10011)_2$

Example 3 : Perform the following : $(28.47)_{10} = (?)_2$.

Solution : For mixed numbers we have to perform conversion separately for integer part and fraction part.

2	28			
	14	0	↑	
	7	0	↑	
	3	1	↑	$28_{10} = 11100_2$
	1	1	↑	
	0	1	↑	

$0.47 \times 2 = 0.94 = 0.94$	with carry	0	
$0.94 \times 2 = 1.88 = 0.88$	with carry	1	

$$\begin{array}{l}
 0.88 \times 2 = 1.76 = 0.76 \quad \text{with carry } 1 \\
 \downarrow \\
 0.76 \times 2 = 1.52 = 0.52 \quad \text{with carry } 1 \\
 \therefore (0.47)_{10} = (0.0111)_2 \\
 \therefore (28.47)_{10} = (11100.0111)_2
 \end{array}$$

4.Hexadecimal to Binary Conversion

Hexadecimal number can be converted into binary number by converting each hexadecimal digit to 4 bit binary equivalent.

Example 1 : Convert $(9CA)_H$ to binary.

Solution :

$$\begin{array}{ccc}
 9 & C & A \\
 \downarrow & \downarrow & \downarrow \\
 1001 & 1100 & 1010 \\
 (9CA)_H & = & (1001\ 1100\ 1010)_2
 \end{array}$$

Example 2 : Convert $(D2E \cdot 8)_{16}$ to binary.

Solution :

$$\begin{array}{cccc}
 D & 2 & E & \cdot & 8 \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 1101 & 0010 & 1110 & \cdot & 1000 \\
 (D2E \cdot 8)_{16} & = & (1101\ 0010\ 1110 \cdot 1000)_2
 \end{array}$$

5.Binary to Hexadecimal Conversion

Binary number can be easily converted to hexadecimal by making a group of four bit starting from binary point. Then group is converted to equivalent hexadecimal.

Example 1: Convert the 1100101001010111 binary number to hexadecimal.

Solution :

$$\begin{array}{cccc}
 1100 & 1010 & 0101 & 0111 \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 C & A & 5 & 7 \\
 (1100101001010111)_2 & = & (CA57)_{16}
 \end{array}$$

Example 2: Convert $(10110110110011)_2$ to hexadecimal.

Solution :

$$\begin{array}{cccc}
 0010 & 1101 & 1011 & 0011 \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 2 & D & B & 3
 \end{array}$$

$$(101101101100011)_2 = (2DB3)_{16}$$

Example 3: Convert $11110101 \cdot 111$ to hexadecimal number.

Solution :

$$\begin{array}{ccc} 1111 & 0101 & \cdot & 1110 \\ \downarrow & \downarrow & & \downarrow \\ F & 5 & \cdot & E \end{array}$$

$$(1111\ 0101 \cdot 111)_2 = (F5 \cdot E)_{16}$$

6.Hexadecimal to Decimal Conversion

In the hexadecimal number system each digit position corresponds to power of 16.

The weights of the digit position in a hexadecimal number are as follows:

$$16^3 \ 16^2 \ 16^1 \ 16^0 \cdot 16^{-1} \ 16^{-2} \ 16^{-3}$$

Hexadecimal
point

Therefore, to convert from hexadecimal to decimal, multiply each hexadecimal digit by its weight and add the resulting products.

Example 1 : Convert $(E5)_{16}$ to decimal number.

Solution :

$$\begin{aligned} (E5)_{16} &= E \times 16^1 + 5 \times 16^0 \\ &= E \times 16 + 5 \times 1 \\ &= 14 \times 16 + 5 \times 1 \\ &= 224 + 5 \\ &= 229 \\ (E5)_{16} &= (229)_{10} \end{aligned}$$

Example 2 : Convert $(B2F \cdot 39)_{16}$ to decimal number.

Solution :

$$\begin{aligned} (B2F \cdot 39)_{16} &= B \times 16^2 + 2 \times 16^1 + F \times 16^0 + 3 \times 16^{-1} + 9 \times 16^{-2} \\ &= 11 \times 256 + 2 \times 16 + 15 \times 1 + 0.1875 + 0.0352 \\ &= (2863 \cdot 2227)_{10} \end{aligned}$$

Example 3 : $(A2.2)_{16} = (?)_{10}$.

Solution :

$$\begin{aligned} (A2.2)_{16} &= A \times 16^1 + 2 \times 16^0 + 2 \times 16^{-1} \\ &= 10 \times 16 + 2 \times 1 + 2 \times \frac{1}{16} \\ &= 160 + 2 + 0.125 \end{aligned}$$

$$= 162.125$$

$$\therefore (A2.2)_{16} = (162.125)_{10}$$

7. Decimal to Hexadecimal Conversion

The decimal number can be converted to hexadecimal number by successively dividing by 16.

Then the remainder is converted to the equivalent hexadecimal.

The remainder taken in reverse order, written from **the bottom to the top**, gives hexadecimal number.

Example 1 : Convert $(650)_{10}$ to hexadecimal by repeated division by 16.

Solution :

Successive division	Quotient	Remainder	Hexadecimal
$\frac{650}{16}$	40	10	A
$\frac{40}{16}$	2	8	8
$\frac{2}{16}$	0	2	2

$\therefore (650)_{10} = (28A)_{16}$

↑
Read

Example 2: Convert $(423)_{10}$ to hexadecimal.

Solution :

Successive division	Quotient	Remainder	Hexadecimal
$\frac{423}{16}$	26	7	7
$\frac{26}{16}$	1	10	A
$\frac{1}{16}$	0	1	1

$\therefore (423)_{10} = (1A7)_{16}$

↑
Read

Example 3 : $(43.4)_{10} = (?)_{16}$.

Solution :

$$\begin{array}{r|l}
 16 & 43 \\
 \hline
 & 2 \quad 11 \quad \rightarrow B \\
 & 0 \quad 2 \quad \rightarrow 2
 \end{array}$$

$$(43)_{10} = (2B)_{16}$$

$$0.4 \times 16 = 6.4 = 0.4 \text{ with carry } 6$$

$$0.4 \times 16 = 6.4 = 0.4 \text{ with carry } 6$$

$$0.4 \times 16 = 6.4 = 0.4 \text{ with carry } 6$$

$$\therefore (43.4)_{10} = (2B.666)_{16}$$

8.Hexadecimal Fraction

For converting the decimal fraction to hexadecimal, **multiply the decimal fraction by 16 and collect all the numbers to the left of the decimal point.**

Convert them into hexadecimal equivalent. Reading the number downwards gives the hexadecimal fraction.

Example 1: Convert 0.357 to hexadecimal equivalent.

Solution : 0.357 is original number.

	Carry	Hexadecimal	
$0.357 \times 16 = 5.712$	5	5	↓ Read
$0.712 \times 16 = 11.392$	11	B	
$0.392 \times 16 = 6.272$	6	6	

Reading the hexadecimal equivalent from top to the bottom.

$$(0.357)_{10} = (0.5B6)_{16}$$

Binary Coded Decimal (BCD)

The BCD code expresses each digit in a decimal number by its 4 bit binary equivalent i.e. nibble.

The 8 4 2 1 code is a type of binary coded decimal (BCD) code and is composed of four bit having binary weight ($2^3 \ 2^2 \ 2^1 \ 2^0$).

With four bit combination 16 numbers can be represented. But in 8421 code only ten of these are used as given in Table:

Table

8421 (BCD)	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

The six code combinations 1010, 1011, 1100, 1101, 1110 and 1111 are invalid in the 8 4 2 1 BCD code.

Example 1 : Convert each of the following decimal numbers into BCD.

Solution :

(i) 429

4 2 9
↓ ↓ ↓
0100 0010 1001

(ii) 9745

9 7 4 5
↓ ↓ ↓ ↓
1001 0111 0100 0101

(iii) 39.2

3 9 . 2
↓ ↓ ↓ ↓
0011 1001 . 0010

Example 2 : Find the decimal number represented by the following BCD code.

Solution : (i) 10001110

Make group of four bit

1000 0110
↓ ↓
8 6

(ii) 100101110101

1001 0111 0101
↓ ↓ ↓
9 7 5

Advantages and Disadvantages of BCD Code

Advantages :

Decimal numbers can be easily converted into BCD numbers and vice versa. Only we have to remember the binary equivalent of decimal 0 through 9. Since we convert only one digit at a time.

Disadvantages :

1. BCD numbers are not used in modern computers because of its limited value. A modern computer must be able to process alphanumeric. Therefore, computer uses binary numbers rather than BCD numbers.
2. The rules for binary addition are not applicable to entire 8421 BCD number, but only to the individual 4 bit group. For example, consider the addition of 12 and 8.

Decimal	Binary	BCD
12	1100	0001 0010
+ 08	+ 1000	+ 0000 1000
20	10100 ←	20 0001 1010

0001 1010 is not the BCD equivalent of 20. As 1010 does not exist in the BCD code.

One's and Two's complement

We know that in the decimal number system (+) sign is used to denote a positive number and a minus (−) sign for denoting a negative number.

Thus, a positive integer 5 can be represented by +5 and negative integer 5 by -5.

It is also customary to drop plus sign, and the absence of any sign means that the number has positive value. This representation of numbers is known as signed numbers.

Similar to signed decimal numbers we have signed binary numbers.

A signed binary number consists of both sign and magnitude information.

The sign indicate whether a number is positive or negative and the magnitude is the value of the number.

There are three forms in which signed integer numbers can be represented in binary, they are sign magnitude, 1's complement and 2's complement form.

Sign Magnitude Representation: Since, digital circuit can understand only two symbols 0 and 1, in the form of voltage levels assigned to it, therefore the same symbol is used to indicate the sign of the number also.

Normally an additional bit is used as the sign bit and is placed as the most significant bit.

A 0 is used to represent a positive number and a 1 to represent a negative number. Thus, when a binary number is represented in sign magnitude, the leftmost bit is the sign bit and the remaining bits are the magnitude bits.

The decimal number + 35 is expressed as an 8-bit signed binary number as

0 0 1 0 0 1 1
↑ ↑
Signed bit Magnitude bits

One's Complement Representation:

In a binary number, if **each 1 is replaced by 0 and each 0 by 1**, the resulting number is **known as one's complement** of the first number.

1's complement of a binary number is another binary number obtained by toggling all bits in it, i.e., transforming the 0 bit to 1 and the 1 bit to 0.

Examples:

1's complement of "0111" is "1000"

1's complement of "1100" is "0011"

In fact both the numbers are complement of each other. If one of these number is positive, then the other number will be negative with the same magnitude and vice-versa.

For example, the 1's complement of a binary number 0101 is 1010. Thus $(0101)_2$ represents $(+5)_{10}$ whereas $(1010)_2$ represents $(-5)_{10}$ in this representation,

using 8 bits, the decimal number -25 is expressed as the 1's complement of +25 (00011001) as 11100110.

Note that in **1's complement** representation also **MSB is 0 for positive numbers and 1 for negative numbers**.

In one's complement representation $(+7)_{10} = (0111)_2$ and $(-7)_{10} = (1000)_2$.

Two's Complement Representation :

The two's complement of binary number is obtained by adding 1 to 1's complement. For example, 2's complement of 0110 is 1010. Since 0110 represents $(+6)_{10}$. Therefore, 1010 represents $(-6)_{10}$ in 2's complement representation.

In this representation also, if the **MSB is 0, the number is positive, whereas if the MSB is 1 the number is negative**.

2's complement of a binary number is 1 added to the 1's complement of the binary number.

Examples:

2's complement of "0111" is "1001"

2's complement of "1100" is "0100"

Binary Addition and Subtraction

Basic Rules

There are four basic rules for adding binary digits :

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = (10)_2$$

Table : Rules of binary addition

Augend	Addend	Sum	Carry	Result
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	0	1	10

In the first three rows in above Table , there is no carry i.e. carry = 0, whereas in the fourth row a carry is produced i.e. carry = 1 and it is added to the next higher binary position as similar to decimal addition.

When there is a carry, we have a situation where three bits are being added (a bit in each of two numbers and a carry bit). The rules for this are as follows :

$$\text{Carry bits} \begin{cases} 1 + 0 + 0 = 01_2 & 1 \text{ with a carry of } 0 \\ 1 + 1 + 0 = 10_2 & 0 \text{ with a carry of } 1 \\ 1 + 0 + 1 = 10_2 & 0 \text{ with a carry of } 1 \\ 1 + 1 + 1 = 11_2 & 1 \text{ with a carry of } 1 \end{cases}$$

The subscript 2 (like 10_2) represents a binary number (8 for octal, 10 for decimal, 16 for hexadecimal).

Example 1 : (a)

$$\begin{array}{r} 01_2 \\ + 01_2 \\ \hline \end{array} \qquad \begin{array}{r} 0110 \\ + 0110 \\ \hline \end{array}$$

	102	02 ₁₀
(b)	112	31 ₀
	<u>+ 112</u>	<u>+ 310</u>
	1102	61 ₀
(c)	1112	71 ₀
	<u>+ 112</u>	<u>+ 310</u>
	10102	101 ₀
(d)	1011	
	<u>+ 1100</u>	
	10111	
	↑	
	carry	

(e) Add two numbers 0101 and 1111.

	(1)	(1)	(1)	← carry
	0	1	0	1
+ 1	1	1	1	1
	1	0	1	0
↑				
	□	□	□	□
	□□□□□□□□ carry			

For 8-bit arithmetic addition, least significant bit (LSB) is on the right and most significant bit (MSB) is on the left.

(MSB)	(LSB)
A7 A6 A5 A4 A3 A2 A1 A0	
B7 B6 B5 B4 B3 B2 B1 B0	

Example 2 : Add two 8-bit numbers 01010111 and 00110101.

Sol. :	(1)	(1)	(1)	(1)	(1)	(1)	← carry
	0	1	0	1	1	1	
+ 0	0	1	1	0	1	0	1
	1	0	0	1	1	0	0

Subtraction

There are four basic rules for subtracting binary digits.

0 - 0 = 0
1 - 1 = 0
1 - 0 = 1
10 ₂ - 1 = 1

Table : Rules of binary subtraction

Minuend	Subtrahend	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

While subtracting numbers, we sometime have to borrow from the next higher column. In Table 3.2 except in the second row, borrow = 0. When borrow = 1, as in the second row, this is to be subtracted from the next higher binary bit as it is done in decimal subtraction.

Example 1 : (a)

$$\begin{array}{r} 112 \\ - 102 \\ \hline 012 \end{array} \qquad \begin{array}{r} 310 \\ - 210 \\ \hline 110 \end{array}$$

(b) Subtract 011 from 101.
When 1 is borrowed, 0 is left

$$\begin{array}{r} 0 \quad \leftarrow \\ 1 \quad 10 \quad 1 \\ - 0 \quad 1 \quad 1 \\ \hline 0 \quad 1 \quad 0 \end{array} \quad \begin{array}{l} \text{Borrow 1 from next column making} \\ \text{10 in this column (10 - 1 = 1)} \\ \leftarrow (1 - 1 = 0) \end{array}$$

(c) Subtract the following :

$$\begin{array}{r} 1011 \\ - 0110 \\ \hline 0101 \end{array}$$

In columns 1 and 2, borrow = 0 and in column 3, it is 1. Therefore, in column 4, first subtract 0 from 1 and from this result obtained subtract the borrow bit.

Multiplication

The following are four rules for multiplying binary digit.

$$\begin{aligned} 0 \times 0 &= 0 \\ 0 \times 1 &= 0 \\ 1 \times 0 &= 0 \\ 1 \times 1 &= 1 \end{aligned}$$

Division in binary follows the same procedure as division in decimal.

Subtraction using 1's Complement Method

1's Complement

The 1's of binary number is found by simply changing all 1s to 0s and all 0s to 1s as illustrated by examples.

Binary number	1's complement
11010	00101
01001	10110
111100	000011
01011010	10100101

1's Complement Subtraction

To subtract a smaller number from a larger number, the 1's complement method is as follows :

1. Determine the 1's complement of the smaller number.
2. Add the 1's complement to the larger number.
3. Remove the carry and add it to the result. This is called end-around carry.

Example 1 : Subtract 1001 from 1100 using the 1's complement method.

Solution :

Direct subtraction

$$\begin{array}{r} 1100 \\ - 1001 \\ \hline 0011 \end{array}$$

1's complement method

$$\begin{array}{r} 1100 \\ + 0110 \rightarrow \text{1's complement of } 1001 \\ \hline \end{array}$$

$$\begin{array}{r} 1001 \\ \text{Add End around carry} \rightarrow \boxed{1} 0010 \\ \hline \end{array}$$

$$\begin{array}{r} \boxed{1} 0010 \\ \downarrow + 1 \\ \hline 0011 \text{ Final answer} \end{array}$$

Example 2 : Subtract 11001 from 11111 using 1's complement method.

Solution :

Direct subtraction

$$\begin{array}{r} 11111 \\ - 11001 \\ \hline 00110 \end{array}$$

1's complement method

$$\begin{array}{r} 11111 \\ + 00110 \rightarrow \text{1's complement of } 11001 \\ \hline \end{array}$$

$$\begin{array}{r} \boxed{1} 00101 \\ \downarrow + 1 \\ \hline 00110 \end{array}$$

Example 3 : Subtract 1000 from 0111 using 1's complement method.

Solution :

Direct subtraction

1's complement method

$$\begin{array}{r} 7 \\ - 8 \\ \hline - 1 \end{array}$$

$$\begin{array}{r} 0111 \\ - 1000 \\ \hline - 0001 \end{array}$$

$$\begin{array}{r} 0111 \\ 0111 \rightarrow 1\text{'s complement} \\ \hline 1110 \leftarrow \text{There is no carry,} \end{array}$$

For direct subtraction subtract smaller number from larger number and assign the sign of larger number.

\therefore result is -ve and is in 1's complement form.
 \therefore result is - 0001.

Subtraction using 2's Complement Method

2's Complement

The 2's complement of a binary number is found by adding 1 to 1's complement.

Example 1 : Binary number is 1010. 1's complement of 1010 is 0101.
 Add 1 to 0101

$$\begin{array}{r} \text{i.e. } 0101 \\ + 1 \\ \hline 0110 \end{array}$$

Example 2 : Binary number 10101 has 2's complement as,

$$\begin{array}{r} 01010 \text{ 1's complement of } 10101 \\ + 1 \text{ Add 1} \\ \hline 01011 \text{ 2's complement of } 10101 \end{array}$$

2's Complement Subtraction

To subtract a smaller number from larger one, 2's complement method is applied as follows :

1. Determine the 2's complement of the smaller number.
2. Add the 2's complement to the larger number.
3. Discard the carry (there is always a carry in this case).

Example 1 : Subtract 1101 from 1111 using 2's complement method.

Solution : Direct subtraction

$$\begin{array}{r} 1111 \\ - 1101 \\ \hline 0010 \end{array}$$

2's complement method

$$\begin{array}{r} 1111 \\ + 0011 \rightarrow 2\text{'s complement of } 1101 \\ \hline \end{array}$$

$$\boxed{1} \ 0010$$

↑
discard the carry

Final result is 0010.

Example 2 : Perform the following subtraction.

Solution : Direct method

$$\begin{array}{r}
 1110 \\
 - 1001 \\
 \hline
 0101
 \end{array}
 \qquad
 \begin{array}{r}
 1110 \\
 + 0111 \rightarrow \text{2's complement of } 1001 \\
 \hline
 \boxed{1} 0101 \\
 \uparrow \\
 \text{discard the carry} \text{ Result : } 0101
 \end{array}$$

Example 3 : Perform subtraction 11000 from 11111.

Solution : Direct method

$$\begin{array}{r}
 11111 \\
 - 11000 \\
 \hline
 00111
 \end{array}
 \qquad
 \begin{array}{r}
 11111 \\
 + 01000 \rightarrow \text{2's complement of } 11000 \\
 \hline
 \boxed{1} 00111 \\
 \uparrow \\
 \text{discard the carry} \text{ Result : } 00111
 \end{array}$$

$$\begin{array}{r}
 11111_2 \\
 - 11000_2 \\
 \hline
 00111_2
 \end{array}
 \qquad
 \begin{array}{r}
 31_{10} \\
 - 24_{10} \\
 \hline
 - 07_{10}
 \end{array}$$

Binary subtraction Decimal subtraction

ASCII (American Standard Code for Information Interchange)

Introduction:

Non-weighted binary codes do not follow the positional weighting principle, i.e., digit position within the number has not assigned fixed value. Examples of non-weighted code are gray code, excess-3 code and alphanumeric code etc.

Character Code : In communication, along with ordinary number, we need some character, punctuation symbols and also control signals. The codes, we have studied so far, is not enough for data communication. For overcoming this problem, binary symbols are used for representing numbers, alphabetic character and special symbols.

The binary codes which are used to represent number, alphabetic character and special symbols are called *alphanumeric codes*.

There are various codes which represent the alphanumeric data by using binary number. Some of them are :

- (i) ASCII
- (ii) EBCDIC.

ASCII (American Standard Code for Information Interchange)

The ASCII code is extensively used for data communication and in computers. It is a 7-bit code, so it can represent 2^7 or 128 different characters. It can represent decimal number 0 to 9, letters of alphabet both upper and lower case, special symbols and control instructions. Each symbol has 7-bit code which is made up of a 3-bit group followed by a 4-bit group. The numbers are represented by 8421 BCD code preceded by 011. For instance, decimal number 3 is represented by 0110011. The Table 1.7 shows code combination of alphabate and other symbols. For example, upper case 'B' is represented by 1000010 whereas lower case 'b' is represented by 1100010, '+' is represented by 0101011.

Example 1. : Solve the following :

(i) $(99)_{10} + (?)_{16}$

Solution : For conversion from decimal to hexadecimal, decimal number is divided by 16 (base of hexadecimal).

Successive division	Quotient	Remainder
$\frac{99}{16}$	6	3
$\frac{6}{16}$	0	6

□ read

Reading the remainder from bottom to the top.

$$(99)_{10} = (63)_{16} \text{ or } 63H$$

(iii) $(1011101)_2 = (?)_{16}$

Solution : Binary numbers can be converted into the equivalent hexadecimal number by making groups of four bit starting from LSB and moving towards MSB. Then replace each group of four bit by its hexadecimal representation.

Start from LSB. 1011101

0 1 0 1	1 1 0 1	
└───┘	└───┘	
↓	↓	
5	D	
□		

$$(1011101)_2 = (5D)_{16} \text{ or } 5DH$$

H stands for hexadecimal.

(iv) $(FA)_{16} = (?)_{10}$

Solution : To convert hexadecimal to decimal, multiply each hexadecimal digit by its weight and add the resulting product.

$$\begin{aligned}
 FA &= F \times 16^1 + A \times 16^0 \\
 &= 15 \times 16^1 + 10 \times 16^0 \\
 &= 240 + 10 \\
 &= 250
 \end{aligned}$$

$$(FA)_{16} = (250)_{10}$$

OR Another method can be used to evaluate a hexadecimal number in terms of its decimal equivalent, first convert the hexadecimal number to binary and then convert from binary to decimal by following way.

(FA)₁₆

First convert FA into binary form

F A

1111 1010

$$\begin{aligned} 11111010_2 &= 2^7 \times 1 + 2^6 \times 1 + 2^5 \times 1 + 2^4 \times 1 + 2^3 \times 1 + 2^2 \times 0 + 2^1 \times 1 + 2^0 \times 0 \\ &= 128 + 64 + 32 + 16 + 8 + 0 + 2 + 0 \\ &= 128 + 64 + 32 + 16 + 8 + 2 = 250_{10} \end{aligned}$$

$$(FA)_{16} = (250)_{10}$$

Example 2 : Convert the decimal number (64)₁₀ into : (i) binary, (ii) hexadecimal.

Solution : (i) Binary : To convert decimal to binary one of the method used is as follows. Determine set of the binary weight values whose sum is equal to the decimal number.

$$\begin{aligned} \square \quad 64_{10} &= 64 + 32 + 16 + 8 + 4 + 2 + 1 \\ &= 64 + 0 + 0 + 0 + 0 + 0 + 0 = 1000000 \end{aligned}$$

OR More systematic method of converting from decimal to binary is repeated division-by-2 process.

Successive division	Quotient	Remainder	
$\frac{64}{2}$	32	0	
$\frac{32}{2}$	16	0	
$\frac{16}{2}$	8	0	
$\frac{8}{2}$	4	0	
$\frac{4}{2}$	2	0	
$\frac{2}{2}$	1	0	\square
$\frac{1}{2}$	0	1	read

$$(64)_{10} = (1000000)_2$$

(iii) Hexadecimal : Repeated division of a decimal number by 16 will produce the equivalent hexadecimal number formed by the remainder of each division.

Successive division	Quotient	Remainder	
$\frac{64}{16}$	4	0	
$\frac{4}{16}$	0	4	read

$(64)_{10} = (40)_{16}$

Example 5 : Convert $(76)_{10}$ into (i) binary, (iii) hexadecimal.

Solution : (i) **Binary :** Convert into binary form by repeated division of 2.

Successive division	Quotient	Remainder	
$\frac{76}{2}$	38	0	
$\frac{38}{2}$	19	0	
$\frac{19}{2}$	9	1	
$\frac{9}{2}$	4	1	
$\frac{4}{2}$	2	0	
$\frac{2}{2}$	1	0	
$\frac{1}{2}$	0	1	□

Reading from bottom to the top,

$$(76)_{10} = (1001100)_2$$

(iii) **Hexadecimal :**

Successive division	Quotient	Remainder
$\frac{76}{16}$	4	12 => C
$\frac{4}{16}$	0	4

Reading the remainders from bottom to the top,

$$(76)_{10} = (4C)_{16}$$

Example 6 : Convert $(132.25)_{10}$ into binary, octal and hexadecimal number.

Solution : (i) First convert integer number to binary number by double-dabble method. i.e. repeated division-by-2 process.

Successive division	Quotient	Remainder
$\frac{132}{2}$	66	0
$\frac{66}{2}$	33	0
$\frac{33}{2}$	16	1
$\frac{16}{2}$	8	0
$\frac{8}{2}$	4	0
$\frac{4}{2}$	2	0
$\frac{2}{2}$	1	0
$\frac{1}{2}$	0	1

read

Read remainders from bottom to the top, $(132)_{10} = (10000100)_2$.

To convert fraction 0.25, multiply by 2 and record a carry in the integer position. The carries read downward are the binary fraction.

$$0.25 \times 2 = 0.50 = 0.5 \text{ with carry of } 0$$

$$0.50 \times 2 = 1.0 = 0.0 \text{ with carry of } 1 \quad \begin{array}{l} \text{read} \\ \downarrow \end{array}$$

$$(0.25)_{10} = (0.01)_2$$

$$\square (132.25)_{10} = (10000100.01)_2$$

(ii) To convert 132.25 to hexadecimal conversion.

First convert 132.

Successive division	Quotient	Remainder
$\frac{132}{16}$	8	4
$\frac{8}{16}$	0	8

\square read

Reading the remainders from bottom to the top,

$$(132)_{10} = (84)_{16}$$

To convert 0.25 multiply by 16.

$$\square 0.25 \times 16, \quad 4.00 \text{ with carry } 4 \quad \begin{array}{l} \text{read} \\ \downarrow \end{array}$$

Read the carry from top to the bottom.

$$(0.25)_{10} = (0.4)_{16}$$

$$\square (132.25)_{10} = (84.4)_{16}$$

(iii) To convert 132.25 to octal, first convert 132 by successive division of 8.

Successive division	Quotient	Remainder
$\frac{132}{8}$	16	4
$\frac{16}{8}$	2	0
$\frac{2}{8}$	0	2



Reading the remainders from bottom to the top,

$$(132)_{10} = (204)_8$$

To convert fraction 0.25,

$$0.25 \times 8 = 2.00 \quad \text{with carry 2} \quad \text{read}$$

$$0.00 \times 8 = 0.00 \quad \text{with carry 0}$$

$$(0.25)_{10} = (0.2)_8$$

$$\square \quad (132.25)_{10} = (204.2)_8$$

Example 7 : Convert the following :

(i) $(26.457)_{10} = (?)_2$, (ii) $(123.746)_{10} = (?)_8$.

Solution : (i) Double-dabble method can be used to convert decimal to binary number.

Let us convert decimal number 26 to its binary equivalent.

$$\frac{26}{2} = 13 \quad \text{with remainder of 0}$$

$$\frac{13}{2} = 6 \quad \text{with remainder of 1}$$

$$\frac{6}{2} = 3 \quad \text{with remainder of 0}$$

$$\frac{3}{2} = 1 \quad \text{with remainder of 1}$$

$$\frac{1}{2} = 0 \quad \text{with remainder of 1} \quad \text{Read}$$

The remainders taken in reverse order (from bottom to the top)

$$(26)_{10} = (11010)_2$$

As far as fractions are concerned, multiply by number 2 and record a carry in the integer position. The carries taken in forward order are the binary fraction.

Convert 0.457 to binary fraction,

$0.457 \times 2 = 0.914 = 0.914$	with carry	0	Read ↓
$0.914 \times 2 = 1.828 = 0.828$	with carry	1	
$0.828 \times 2 = 1.656 = 0.656$	with carry	1	
$0.656 \times 2 = 1.312 = 0.312$	with carry	1	
$0.312 \times 2 = 0.624 = 0.624$	with carry	0	
$0.624 \times 2 = 1.248 = 0.248$	with carry	1	

Taking the carries in forward order give binary fraction 0.011101. In this case, we stop the process after getting 6 binary digits. This answer gives an approximation. If more accuracy is needed, continue multiplying by 2 until you have as many digits as necessary.

(ii)

Example 8 : Convert $(8.5625)_{10}$ to binary.

Solution : For mixed decimal (number with integer and fractional part) we have to convert integer part and fractional part separately into binary

2	8		
	4	0	↑ $8_{10} = 1000_2$
	2	0	
	1	0	
	0	1	

Convert decimal fraction 0.5625 into binary

$0.5625 \times 2 = 1.125 = 0.125$	with carry	1	↓
$0.125 \times 2 = 0.250 = 0.250$	with carry	0	
$0.250 \times 2 = 0.500 = 0.500$	with carry	0	
$0.500 \times 2 = 1.000 = 0.000$	with carry	1	

$$(0.5625)_{10} = (0.1001)_2$$

$$\therefore (8.5625)_{10} = (1000.1001)_2$$

Example 9 : Convert $(143)_{10}$ into binary and hexadecimal .

Solution : $(143)_{10} = (X)_2 = (Y)_{16}$.

2	143		↑
	71	1	
	35	1	
	17	1	
	8	1	

4	0
2	0
1	0
0	1

 $(143)_{10} = (10001111)_2$

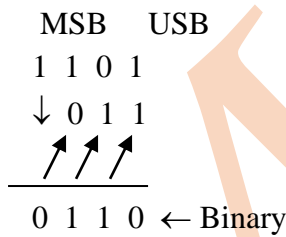
16	143	
8	15 = F	↑
0	8	

 $(143)_{10} = (8F)_{16}$

Example 10 : Perform the following :

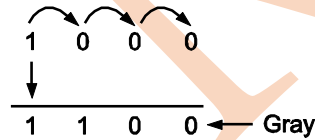
- (i) $(1101)_{\text{Gray}} = (?)_{\text{Binary}}$
(ii) $(1000)_{\text{Binary}} = (?)_{\text{Gray}}$

Solution : (i) $(1101)_{\text{Gray}} = (?)_{\text{Binary}}$



∴ $(1101)_{\text{Gray}} = (0110)_{\text{Binary}}$

(ii) $(1000)_{\text{Binary}} = (?)_{\text{Gray}}$



∴ $(1000)_{\text{Binary}} = (1100)_{\text{Gray}}$

(iii) $(291)_{10} = (?)_{\text{Excess-3}}$

2	9	1
+ 3	+ 3	+ 3
—	—	—
5	12	4
0101	1100	0100

$(291)_{10} = (0101 \ 1100 \ 0100)_{\text{Excess-3}}$

(iv) $(97)_{10} = (?)_{\text{BCD}}$

Decimal	BCD	
9	→ 1001	
7	→ 0111	∴ $(97)_{10} = (1001 \ 0111)_{\text{BCD}}$

Example 11: Perform the following :

- (i) $(11011)_2 = (?)_{10}$, (ii) $(9806)_{10} = (?)_{16}$, (iii) $(F8A)_{16} = (?)_2$, (iv) $(481)_{10} = (?)_2$.

Solution : (i) $(11011)_2 = (?)_{10}$

1 1 0 1 1

Step (i) : Write the binary weight.

Step (ii) : Discard the weight whose coefficients are zero.

16 8 4 2 1
Step (iii) : Add the remaining weights
 $= 16 + 8 + 2 + 1 = 27$

$\therefore (11011)_2 = (27)_{10}$
 (ii) $(9806)_{10} = (?)_{16}$

16	9806		
	612	14	→
	38	E	↑
	2	4	
	0	6	
		2	

$\therefore (9806)_{10} = (264E)_{16}$
 (iii) $(F8A)_{16} = (?)_2$

F → 1111
 8 → 1000
 A → 1010

$\therefore (F8A)_{16} = (1111\ 1000\ 1010)_2$

(iv) $(481)_{10} = (?)_2$

2	481		
	240	1	↑
	120	0	
	60	0	
	30	0	
	15	0	
	7	1	
	3	1	
	1	1	
	0	1	

$\therefore (481)_{10} = (111100001)_2$

Example 12 : Perform the following conversions :

- (i) $(0111)_G = (?)_{\text{Binary}}$. (ii) $(1100)_2 = (?)_{\text{Gray}}$ (iii) $(32)_{10} = (?)_{\text{Excess-3}}$, (iv) $(183)_{10} = (?)_{\text{BCD}}$.

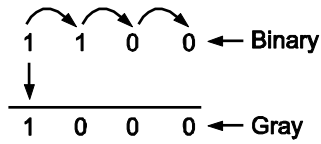
Solution : (i) $(0111)_{\text{Gray}} = (?)_{\text{Binary}}$
 Gray to binary conversion

0 1 1 1 ← Gray
 ↓ 0 1 0
 ↗ ↘ ↗ ↘

$\overline{0101} \leftarrow \text{Binary}$

$\therefore (0111)_G = (0101)_{\text{Binary}}$

(ii) $(1100)_2 = (?)_{\text{Gray}}$



$\therefore (1100)_2 = (1000)_{\text{Gray}}$

(iii) $(32)_{10} = (?)_{\text{Excess-3}}$

$$\begin{array}{r} 32 \\ + 3 \\ \hline 35 \end{array} \qquad \begin{array}{r} 2 \\ + 3 \\ \hline 5 \end{array}$$

$\therefore (32)_{10} = (65)_{\text{Excess-3}}$

(iv) $(183)_{10} = (?)_{\text{BCD}}$

1 \rightarrow 0001

8 \rightarrow 1000

3 \rightarrow 0011

$(183)_{10} = (0001\ 1000\ 0011)_{\text{BCD}}$

Example 13 : Perform the following :

(i) $(10110)_2 = (?)_{10}$.

Solution : Write the binary number

1 0 1 1 0

Write the weights below it

16 8 4 2 1

Discard the weights whose coefficients are zero

16 ~~8~~ 4 2 ~~1~~

Add the remaining weights

16 + 4 + 2 = 22

$\therefore (10110)_2 = (22)_{10}$

(ii) $(fB6)_{16} = (?)_2$

Solution :

f \rightarrow 1111₂

0 \rightarrow 0000₂

B \rightarrow 1011₂

6 = 0110₂

$\therefore (f0B6)_{16} = 1111\ 0000\ 1011\ 0110_2$

(iii) $(589)_{10} = (?)_{\text{BCD}}$.

Solution :

5₁₀ = 0101_{BCD}

8 = 1000_{BCD}

9 = 1001_{BCD}

$\therefore (589)_{10} = 0101\ 1000\ 1001_{\text{BCD}}$

AFRIMD