It takes at least five bits to count from 0 to 31. The following formula tells us how high we can count in decimal with n bits, beginning with zero.

Highest decimal number $= 2^n - 1$

With 2 bits we can count as ;

$2^2 - 1 = 4 - 1 = 3$ *i.e.*, from 0 to 3.

With 4 bits,

$2^4 - 1 = 16 - 1 = 15$ *i.e.*, from 0 to 15

With 6 bits,

$2^6 - 1 = 64 - 1 = 63$ *i.e.*, from 0 to 63

Collection of 4 bits is called a **Nibble.** *Collection of 8 bits is called a Byte.* Therefore a byte contains 2 nibble and byte is the basic unit of storage in computer.

2.3. BINARY TO DECIMAL CONVERSION

A binary number can be converted into decimal number using the weights assigned to it. The value of a given binary number in terms of decimal equivalent can be determined by adding the products of each bit and its weight. The right most bit is the least significant bit (LSB) and has a weight of 0 *i.e.*, $2^0 = 1$. The weight increases by power of two (2^0, 2^1, 2^2, 2^3, 2^4, ...) for each bit from right to left. The method can be well under stood from the following examples.

EXAMPLE 2.1 : Convert 1101 to Decimal.

Solution :

1	1	0	1
2^3	2^2	2^1	2^0

Weight Increase from Right to Left ⟵

$= 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3$

$= 1 + 0 + 4 + 8$

$= 1 + 4 + 8 = 13$

$\therefore$ $(1101)_2 = (13)_{10}$

Binary Decimal

EXAMPLE 2.2 : Convert $(110101)_2$ to Decimal.

Solution :

1	1	0	1	0	1
2^5	2^4	2^3	2^2	2^1	2^0

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

$= 1 + 0 + 4 + 0 + 16 + 32$

$= 1 + 4 + 16 + 32 = (53)_{10}$

$\therefore$ $(110101)_2 = (53)_{10}$

Binary Decimal

For differentiating between different number systems ; either corresponding number system may be specified along with number or small subscript at the end of number may be added signifying the number system e.g., $(1010)_2$ represents a binary number as indicated from base 2.

Example 2.3 : Determine the decimal equivalent of following binary number.

(a) $(110110)_2$ *(b)* $(1101101)_2$ *(c)* $(111110101)_2$

(d) $(100001000)_2$ *(e)* $(00000100)_2$

Solution :

(a) $(110110)_2$ $= 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 1 \times 2^5$

$= 0 + 2 + 4 + 0 + 16 + 32$

$= (54)_{10}$ **Ans.**

(b) $(1101101)_2$ $= 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 + 1 \times 2^6$

$= 1 + 0 + 4 + 8 + 0 + 32 + 64$

$= (109)_{10}$ **Ans.**

(c) $(111110101)_2$ $= 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 1 \times 2^5 + 1 \times 2^6$

$1 \times 2^7 + 1 \times 2^8$

$= 1 + 0 + 4 + 0 + 16 + 32 + 64 + 128 + 256$

$= (501)_{10}$ **Ans.**

(d) $(100001000)_2$ $= 0 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 0 \times 2^5 + 0 \times 2^6$

$+ 0 \times 2^7 + 1 \times 2^8$

$= 0 + 0 + 0 + 8 + 0 + 0 + 0 + 0 + 256$

$= (264)_{10}$ **Ans.**

(e) $(00000100)_2$ $= 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 0 \times 2^5 + 0 \times 2^6$

$+ 0 \times 2^7$

$= 0 + 0 + 4 + 0 + 0 + 0 + 0 + 0$

$= (4)_{10}$ **Ans.**

Therefore rules for converting from Binary to Decimal are as follow ;

1. Each digit of binary number is multiplied by 2 having power (position – 1). Here position is the position of binary digit in the number counted from right hand side.
2. All the products of multiplication are summed to get the decimal equivalent of the number.

The binary numbers converted so far are whole numbers. Fractions can also be represented in inary by placing bits to the right of the binary position in same way as in decimal number. The eights for binary numbering are ;

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

↓

Binary point

Thus, all the bits on the left hand side of binary point have positive powers of two and all bits to the right of binary point have weights that are negative power of two.

EXAMPLE 2.4 : Determine the decimal value of binary number $(101.0101)_2$

Solution :

$(101.0101)_2 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4}$

$= 1 + 0 + 4 + 0 + \frac{1}{4} + 0 + \frac{1}{16}$

$= 5 + 0.25 + 0.0625$

$= (5.3125)_{10}$ **Ans.**

EXAMPLE 2.5 : Determine the decimal number represented by following binary numbers.

(a) $(111.101)_2$ **(b) $(1100.1011)_2$** **(c) $(1011.10101)_2$**

(d) $(0.1011)_2$

Solution :

(a) $(111.101)_2$ $= 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

$= 1 + 2 + 4 + \frac{1}{2} + 0 + \frac{1}{8}$

$= 7 + 0.5 + 0.125$

$= (7.625)_{10}$ **Ans.**

(b) $(1100.1011)_2$ $= 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^{-1} + 0 \times 2^{-2} +$
$1 \times 2^{-3} + 1 \times 2^{-4}$

$= 0 + 0 + 4 + 8 + \frac{1}{2} + 0 + \frac{1}{8} + \frac{1}{16}$

$= 4 + 8 + 0.5 + 0.125 + 0.0625$

$= (12.6875)_{10}$ **Ans.**

(c) $(1011.10101)_2$ $= 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 1 \times 2^{-1} + 0 \times 2^{-2} +$
$1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5}$

$= 1 + 2 + 0 + 8 + \frac{1}{2} + 0 + \frac{1}{8} + 0 + \frac{1}{32}$

$= 11 + 0.5 + 0.125 + 0.03125$

$= (11.65625)_{10}$ **Ans.**

(d) $(0.1011)_2$ $= 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$

$= 0 + \frac{1}{2} + 0 + \frac{1}{8} + \frac{1}{16} = 0.5 + 0.125 + 0.0625$

$= (0.6875)_{10}$ **Ans.**

2.4. DECIMAL TO BINARY CONVERSION

Any decimal number can be converted into its binary equivalent number. For integer, the conversion is obtained by continuous division by 2 and keeping track of the remainder, while for fractional parts, the conversion is affected by continuous multiplication by 2 and keeping track of the integers generated.

For converting decimal number to binary, the decimal number is divided by 2 successively. At each stage, quotient and remainder are noted down. The quotient of one stage is divided by 2 at the next stage. The procedure is repeated till quotient becomes zero.

EXAMPLE 2.6 : Find the binary equivalent of decimal number 37.

Solution :

		Quotient	Remainder
Divide	37 by 2	18	1 (LSB)
Divide	18 by 2	9	0
Divide	9 by 2	4	1
Divide	4 by 2	2	0
Divide	2 by 2	1	0
Divide	1 by 2	0	1 (MSB)

∴ Binary number is $(100101)_2$

Check the Decimal equivalent

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

$= 1 + 0 + 4 + 0 + 0 + 32$

$= (37)_{10}$ **Ans.**

EXAMPLE 2.7 : Convert decimal number 25 to its binary equivalent.

Solution : $(25)_{10} = (\ ?)_2$

2	25		
2	12	1	(LSB)
2	6	0	
2	3	0	Bottom to top
2	1	1	
	0	1	(MSB)

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

CHECK : $(11001)_2 = 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 1 \times 2^4$

$= 1 + 0 + 0 + 8 + 16$

$= (25)_{10}$ **Ans.**

EXAMPLE 2.8 : Convert the following decimal numbers to binary.

(*a*) $(257)_{10}$ (*b*) $(99)_{10}$ (*c*) $(21)_{10}$ (*d*) $(65)_{10}$ (*e*) $(73)_{10}$ (*f*) $(59)_{10}$ (*g*) $(15)_{10}$ (*h*) $(28)_{10}$

Solution : (*a*) $(257)_{10} = (?)_2$

2	257		
2	128	1	(LSB)
2	64	0	
2	32	0	
2	16	0	
2	8	0	
2	4	0	
2	2	0	
2	1	0	
	0	1	(MSB)

∴ $(257)_{10} = (100000001)_2$ **Ans.**

(*b*) $(99)_{10} = (?)_2$

2	99		
2	49	1	(LSB)
2	24	1	
2	12	0	
2	6	0	
2	3	0	
2	1	1	
	0	1	(MSB)

∴ $(99)_{10} = (1100011)_2$ **Ans.**

(*c*) $(21)_{10} = (?)_2$

2	21		
2	10	1	(LSB)
2	5	0	
2	2	1	
2	1	0	
	0	1	(MSB)

∴ $(21)_{10} = (10101)_2$ **Ans.**

(*d*) $(65)_{10} = (\ ?)_2$

2	65		
2	32	1	(LSB)
2	16	0	
2	8	0	
2	4	0	
2	2	0	
2	1	0	
	0	1	(MSB)

∴ $(65)_{10} = (1000001)_2$ **Ans.**

(*e*) $(73)_{10} = (\ ?)_2$

2	73		
2	36	1	(LSB)
2	18	0	
2	9	0	
2	4	1	
2	2	0	
2	1	0	
	0	1	(MSB)

∴ $(73)_{10} = (1001001)_2$ **Ans.**

(*f*) $(59)_{10} = (\ ?)_2$

2	59		
2	29	1	(LSB)
2	14	1	
2	7	0	
2	3	1	
2	1	1	
	0	1	(MSB)

∴ $(59)_{10} = (111011)_2$ **Ans.**

(g) $(15)_{10} = (\ ?)_2$

2	15		
2	7	1	(LSB)
2	3	1	
2	1	1	
	0	1	(MSB)

$\therefore$ $(15)_{10} = (1111)_2$ **Ans.**

(h) $(28)_{10} = (\ ?)_2$

2	28		
2	14	0	(LSB)
2	7	0	
2	3	1	
2	1	1	
	0	1	(MSB)

$\therefore$ $(28)_{10} = (11100)_2$ **Ans.**

2.5. CONVERTING DECIMAL FRACTIONS TO BINARY

Till now we have converted whole decimal numbers to binary. Decimal numbers can be converted to binary by dividing it by 2 *i.e.*, base of binary numbers. For converting decimal fractions to binary instead of dividing, we have to multiply fraction part by 2 and obtaining the integer part, we have to multiply fraction part by 2 until the fractional product is zero. The carries generated by each multiplication form the binary number. The first carry produced is MSB. The procedure is as follows ;

EXAMPLE 2.9 : Convert $(0.6875)_{10}$ to binary.

Solution :

		Carry	
Step 1 :	$0.6875 \times 2 = 1.3750$	1	(MSB)
Step 2 :	$0.3750 \times 2 = 0.7500$	0	
Step 3 :	$0.7500 \times 2 = 1.5000$	1	
Step 4 :	$0.5000 \times 2 = 1.0000$	1	(LSB)

$\therefore$ $(0.6875)_{10} = (0.1011)_2$ **Ans.**

CHECK : $(0.1011)_2 = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$

$$= \frac{1}{2} + 0 + \frac{1}{8} + \frac{1}{16}$$

$$= 0.5 + 0.125 + 0.0625$$

$$= (0.6875)_{10}$$

EXAMPLE 2.10 : Convert the following decimal numbers into binary.

(*a*) 25.5_{10} (*b*) 10.625_{10} (*c*) 19.6_{10}

Solution :

(*a*) $(25.5)_{10} = (\ ?)_2$

Firstly take whole number part

2	25		
2	12	1	(LSB)
2	6	0	
2	3	0	
2	1	1	
	0	1	(MSB)

Now take fraction part

	Carry	
$0.5 \times 2 = 1.000$	1	(MSB)
$0.000 \times 2 = 0.000$		

$\therefore$ $(25.5)_{10} = (11001.1)_2$ **Ans.**

(*b*) $(10.625)_{10} = (\ ?)_2$

Firstly take whole number part

2	10		
2	5	0	(LSB)
2	2	1	
2	1	0	
	0	1	(MSB)

Now take fraction part

	Carry	
$0.625 \times 2 = 1.250$	1	(MSB)
$0.250 \times 2 = 0.500$	0	
$0.500 \times 2 = 1.000$	1	(LSB)

$\therefore$ **$(10.625)_{10} = (1010.101)_2$ Ans.**

(*c*) $(19.6)_{10} = (\ ?)_2$

Firstly take whole number part

2	19		
2	9	1	(LSB)
2	4	1	
2	2	0	
2	1	0	
	0	1	(MSB)

Now take fractional part

		Carry	
0.6×2	$= 1.2$	1	(MSB)
0.2×2	$= 0.4$	0	↓
0.4×2	$= 0.8$	0	↓
0.8×2	$= 1.6$	1	↓
0.6×2	$= 1.2$	1	(LSB)

and it will go on

∴ It is terminated at fifth bit

Hence $(19.6)_{10} = (10011.10011)_2$ **Ans.**

2.6. OCTAL NUMBER SYSTEM

The octal number system is composed of eight digits. These are

0, 1, 2, 3, 4, 5, 6, 7

To count above 7, we begin another column and start over

10, 11, 12, 13, 14, 15, 16, 17, 20, 21, 22, 23..........etc.

Counting in octal is same as that in decimal except, any number with 8 or 9 is omitted. The base or radix to an octal number is 8. Similar to Decimal and binary number system, it is also a positional system and has in general two part ; **Integer** and **fractional**. Table 2.3 shows first seventeen numbers along with their respective decimal and binary numbers.

TABLE 2.3

Octal Numbers and their equivalent

Decimal	Binary	Octal
0	00000	0
1	00001	1
2	00010	2
3	00011	3
4	00100	4
5	00101	5
6	00110	6
7	00111	7
8	01000	10
9	01001	11
10	01010	12
11	01011	13
12	01100	14
13	01101	15
14	01110	16
15	01111	17
16	10000	20
17	10001	21

From the above table, it is clear that digits 0 through 7 has the same meaning as they have in decimal system. After 7 the next digit in octal is 10 though 17 then 20 though 27 and so on. 8 and 9 are omitted in Octal Number System.

2.6.1. Octal to Decimal Conversion

Octal number system has a base of 8 and in Octal number, each digit corresponds to some power of 8. For converting an octal number into a decimal number, we make use of same equation. We used for converting binary to decimal with base 2. The various digits position in this system have weights as follows :

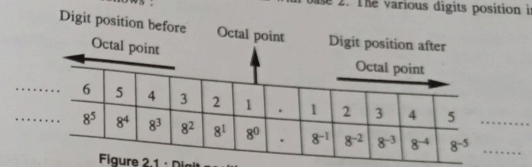

Figure 2.1 : Digit position and their respective weights

EXAMPLE 2.11 : Convert 1035_8 to decimal.

Solution : $1035_8 = 5 \times 8^0 + 3 \times 8^1 + 0 \times 8^2 + 1 \times 8^3$

$= 5 + 24 + 0 + 512$

$= (541)_{10}$ **Ans.**

EXAMPLE 2.12 : Convert the following octal numbers to decimal.

(a) 2374_8 **(b) 645_8** **(c) 5273_8** **(d) 75.5_8**

Solution :

(a) 2374_8 $= 4 \times 8^0 + 7 \times 8^1 + 3 \times 8^2 + 2 \times 8^3$

$= 4 + 56 + 3 \times 64 + 2 \times 512$

$= 4 + 56 + 192 + 1024$

$= 1276_{10}$

∴ $(2374)_8 = (1276)_{10}$ **Ans.**

(b) 645_8 $= 5 \times 8^0 + 4 \times 8^1 + 6 \times 8^2$

$= 5 \times 1 + 32 + 6 \times 64$

$= 5 + 32 + 384$

$= 421_{10}$

∴ $(645)_8 = (421)_{10}$ **Ans.**

(c) 5273_8 $= 3 \times 8^0 + 7 \times 8^1 + 2 \times 8^2 + 5 \times 8^3$

$= 3 \times 1 + 56 + 2 \times 64 + 5 \times 512$

$= 3 + 56 + 128 + 2560$

$= 2747_{10}$

$\therefore$ $(5273)_8 = (2747)_{10}$ **Ans.**

(d) 75.5_8 $= 5 \times 8^0 + 7 \times 8^1 + 5 \times 8^{-1}$

$= 5 \times 1 + 56 + \frac{5}{8}$

$= 61 + 0.625$

$= 61.625_{10}$

$\therefore$ $(75.5)_8 = (61.625)_{10}$ **Ans.**

2.6.2. Decimal to Octal Conversion

The conversion from decimal to octal is same as the conversion from decimal to binary. The only difference is that in conversion from decimal to binary there is division by 2 where as from decimal to octal, there is division by 8. In the same way, the fractional part is converted by using repeated multiplication with 8.

EXAMPLE 2.13 : Convert 562.3_{10} to octal

Solution :

8	562		
8	70	2	(LSB)
8	8	6	↑
8	1	0	↑
	0	1	(MSB)

$0.3 \times 8 = 2.4$	2	(MSB)
$0.4 \times 8 = 3.2$	3	↓
$0.2 \times 8 = 1.6$	1	↓
$0.6 \times 8 = 4.8$	4	↓
$0.8 \times 8 = 6.4$	6	↓
$0.4 \times 8 = 3.2$	3	(LSB)

upto 6 decimal places

$\therefore$ $(562.3)_{10} = (1062.231463)_8$ upto 6 decimal places. **Ans.**

EXAMPLE 2.14 : Convert the following decimal numbers to octal.

(a) $(359)_{10}$ (b) $(0.6875)_{10}$ (c) $(297)_{10}$ (d) $(7825)_{10}$ (e) $(0.225)_{10}$

Solution :

(a) $(359)_{10} = (\ ?)_8$

8	359		
8	44	7	(LSB)
8	5	4	↑
	0	5	(MSB)

$\therefore$ $(359)_{10} = (547)_8$ **Ans.**

(b) $(0.6875)_{10} = (?)_8$

0.6875_{10} $= 0.6875 \times 8 = 5.5000$ | 5 (MSB)

$0.5000 \times 8 = 4.0000$ | 4 (LSB)

$\therefore$ $(0.6875)_{10} = (0.54)_8$ **Ans.**

(c) $(297)_{10} = (\ ?)_8$

8	297		
8	37	1	(LSB)
8	4	5	↑
	0	4	(MSB)

$\therefore$ $(297)_{10} = (451)_8$ **Ans.**

(d) $(7825)_{10} = (?)_8$

8	7825		
8	978	1	(LSB)
8	122	2	↑
8	15	2	↑
8	1	7	↑
	0	1	(MSB)

$\therefore$ $(7825)_{10} = (17221)_8$ **Ans.**

(e) $(0.225)_{10} = (?)_8$

$(0.225)_{10} = 0.225 \times 8 = 1.80$	1	(MSB)
$0.80 \times 8 = 6.40$	6	↓
$0.40 \times 8 = 3.20$	3	↓
$0.20 \times 8 = 1.60$	1	↓
$0.60 \times 8 = 4.80$	4	↓
$0.80 \times 8 = 6.40$	6	(LSB)

upto 6 decimal places

$\therefore$ $(0.225)_{10} = (0.163146)_8$ upto 6 decimal places **Ans.**

2.6.3. Octal to Binary conversion

For converting octal number into binary number, each octal digit is converted into equivalent binary notation. It takes only one octal digit to represent three bits because all three bit binary numbers are required to represent the eight octal digits, it is very easy to convert from octal to binary and from binary to octal. Octal number system is used in digital system, especially for input/output applications. The table for representing each octal bit is as follows :

TABLE 2.4

Octal number & their Binary equivalent

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

To convert an octal number to binary, replace each octal digit by appropriate binary bits.

EXAMPLE 2.15 : Convert $(627)_8$ to binary.

Solution : $(627)_8 = (?)_2$

$(627)_8$ = 6 2 7

↓ ↓ ↓

110 010 111

$\therefore$ $(627)_8 = (110010111)_2$ **Ans.**

EXAMPLE 2.16 : Convert the following octal numbers to binary.

(*a*) $(474)_8$ (*b*) $(37.12)_8$ (*c*) $(43.52)_8$ (*d*) $(11.12)_8$

(*e*) $(5473.64)_8$ (*f*) $(170.6)_8$ (*g*) $(55337.3)_8$

Solution : (*a*) $(474)_8 = (?)_2$

$(474)_8$ = 4 7 4

100 111 100

$\therefore$ $(474)_8 = (100111100)_2$ **Ans.**

(*b*) $(37.12)_8 = (?)_2$

$(37.12)_8$ = 3 7 . 1 2

011 111 . 001 010

$\therefore$ $(37.12)_8 = (011111.001010)_2$ **Ans.**

(*c*) $(43.52)_8 = (?)_2$

$(43.52)_8$ = 4 3 . 5 2

100011 . 101 010

$\therefore$ $(43.52)_8 = (100011.101010)_2$ **Ans.**

(*d*) $(11.12)_8 = (?)_2$

$(11.12)_8$ = 1 1 . 1 2

001001 . 001 010

$\therefore$ $(11.12)_8 = (001001.001010)_2$ **Ans.**

(*e*) $(5473.64)_8 = (?)_2$

$(5473.64)_8$ = 5 4 7 3 . 6 4

101 100 111 011 . 110 100

$\therefore$ $(5473.64)_8 = (101100111011.110100)_2$ **Ans.**

(*f*) $(170.6)_8 = (?)_2$

$(170.6)_8$ = 1 7 0 . 6

001111 000 . 110

$\therefore$ $(170.6)_8 = (001111000.110)_2$

(*g*) $(55337.3)_8 = (?)_2$

$(55337.3)_8$ = 5 5 3 3 7 . 3

101101 011 011 111 . 011

$\therefore$ $(55337.3)_8 = (101101011011111.011)_2$ **Ans.**

EXAMPLE 2.17 : Convert $(37)_8$ to binary and check for answer.

Solution : $(37)_8 = (?)_2$

$(37)_8$ = 3 7

011111

$\therefore$ $(37)_8 = (011111)_2$ **Ans.**

CHECK : Now convert $(011111)_2$ to decimal

$$(011111)_2 = 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 0 \times 2^5$$
$$= 1 + 2 + 4 + 8 + 16$$
$$= (31)_{10} \text{ Ans.}$$

Now convert $(37)_8$ to decimal

Let $(37)_8 = (?)_{10}$

$$= 7 \times 8^0 + 3 \times 8^1$$
$$= 7 \times 1 + 24$$
$$= (31)_{10}$$

Which checks the result & verifies the answer.

2.6.4. Binary to Octal Conversion

This conversion is the reverse of octal to binary conversion. For converting a binary number into an octal, we divide the binary number into group of three bits each, starting at the LSB *i.e.*, binary point. The integer part of binary number is separated in groups of three bits starting from the binary point (*i.e., Decimal point*) and preceding to the left. The fractional part is also separated into groups of three bits starting from binary point and preceding to right. Each group is expressed as octal equivalent.

EXAMPLE 2.18 : Convert $(101111100)_2$ to octal number.

Solution : $(101111100)_2 = (?)_8$

$= 101\ 111\ 100$

$= 5\ \ 7\ \ 4$

$\therefore (101111100)_2 = (574)_8$

EXAMPLE 2.19 : Convert the following binary numbers to octal numbers.

(*a*) $(101011101.011)_2$ (*b*) $(110111011011.100)_2$ (*c*) $(01011011.011)_2$

(*d*) $(1101.011)_2$ (*e*) $(11.1)_2$

Solution :

(*a*) $(101011101.011)_2 = (?)_8$

$= 101\ 011\ 101\ .\ 011$

$5\ \ 3\ \ 5\ .\ 3$

$\therefore (101011101.11)_2 = (535.3)_8$ **Ans.**

(*b*) $(110111011011.100)_2 = (?)_8$

$= 110\ 111\ 011\ 011\ .\ 100$

$= 6\ \ 7\ \ 3\ \ 3\ .\ 4$

$\therefore (110111011011.100)_2 = (6733.4)_8$ **Ans.**

(*c*) $(01011011.011)_2 = (?)_8$

$= 001\ 011\ 011\ .\ 011$

$= 1\ \ 3\ \ 3\ .\ 3$

$\therefore (01011011.011)_2 = (1333.3)_8$ **Ans.**

(*d*) $(1101.011)_2 = (?)_8$

$= 001\ 101\ .\ 011$

$= 1\ \ 5\ .\ 3$

$\therefore (1101.011)_2 = (15.3)_8$ **Ans.**

(*e*) $(11.1)_2 = (?)_8$

$= 011\ .\ 100$

$= 3\ .\ 4$

$\therefore (11.1)_2 = (3.4)_8$ **Ans.**

2.7. HEXADECIMAL NUMBER SYSTEM

Hexadecimal number system is a very popular number system which is used in digital systems and computers.

TABLE 2.5

Hexadecimal Numbers & their equivalent binary & decimal numbers

DECIMAL	BINARY	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10
17	10001	11
18	10010	12
19	10011	13
20	10100	14
21	10101	15
22	10110	16
23	10111	17
24	11000	18
25	11001	19
26	11010	1A
27	11011	1B
28	11100	1C
29	11101	1D
30	11110	1E
31	11111	1F
32	100000	20

While using binary number system, we are faced with the problem while expressing very large numbers. It requires long sequence of 0's and 1's. Hexadecimal number system is used for expressing binary numbers concisely and by and large it is most commonly used number system. This number system is formed from a binary number by grouping bits in groups of four bits each, starting from binary point.

Hexadecimal system has a base of 16 *i.e.*, it is composed of 16 digits and characters. Ten digits and six alphabetic character make up the number system. A subscript 16 indicates a hexadecimal number. The number system is shown in Table 2.5.

From the above table, we observed that hexadecimal numbers are from 0 to 9, then A to F and then 10 to 19 and so on. As it is evident from table 10_{16} in hexadecimal is equivalent to 16_{10} in decimal number system. So $10_{16} \neq 10_{10}$.

2.7.1. Hexadecimal to binary conversion

For converting hexadecimal number to binary, we convert each hexadecimal digit to its 4 bit binary equivalent.

EXAMPLE 2.20 : Convert $C5E2F8_{16}$ to binary.

Solution : $(C5E2F8)_{16} = (?)_2$

= C 5 E 2 F 8

↓ ↓ ↓ ↓ ↓ ↓

= 1100 0101 1110 0010 1111 1000

$(C5E2F8)_{16}$ = $(110001011110001011111000)_2$ **Ans.**

EXAMPLE 2.21 : Convert the following hexadecimal numbers to their binary equivalent.

(*a*) $(5CB8)_{16}$ (*b*) $(4F2D)_{16}$ (*c*) $(6F3.42)_{16}$

Solution :

(*a*) $(5CB8)_{16}$ = $(?)_2$

= 5 C B 8

= 0101 1100 1011 1000

∴ $(5CB8)_{16}$ = $(101110010111000)_2$ **Ans.**

(*b*) $(4F2D)_{16}$ = $(?)_2$

= 4 F 2 D

= 0100 1111 0010 1101

∴ $(4F2D)_{16}$ = $(100111100101101)_2$ **Ans.**

(*c*) $(6F3.42)_{16}$ = (?)2

= 6 F 3 . 4 2

= 0110 1111 0011 . 0100 0010

∴ $(6F3.42)_{16}$ = $(11011110011.01000010)_2$ **Ans.**

2.7.2. Binary to Hexadecimal conversion

Binary numbers can be converted into the equivalent hexadecimal numbers by making groups of four bits starting from LSB and moving towards MSB for integer part and them replacing each group of four bits by its hexadecimal representation. For fractional part, same procedure is repeated from binary point and moving towards right.

EXAMPLE 2.22 : Convert the following binary numbers to equivalent hexadecimal numbers.

(*a*) 10100110101111 (*b*) 1000.000111110101101

Solution : $(10100110101111)_2$ = $(?)_{16}$

= 0010 1001 1010 1111

↓ ↓ ↓ ↓

2 9 A F

∴ $(10100110101111)_2$ = $(29AF)_{16}$ **Ans.**

(*b*) $(1000.000111110101101)_2$ = $(?)_{16}$

= 1000 . 0001 1111 0101 1010

↓ ↓ ↓ ↓ ↓

= 8 . 1 F 5 A

∴ $(1000.000111110101101)_2$ = $(8.1F5A)_{16}$ **Ans.**

2.7.3. Hexadecimal to Decimal Conversion

One way to convert hexadecimal numbers to its decimal equivalent is to first convert the number into Binary and then convert from Binary to Decimal. The following example illustrates the procedure.

EXAMPLE 2.23 : Convert the following hexadecimal numbers to decimal equivalent.

(*a*) $(4F)_{16}$ (*b*) $(CDE)_{16}$ (*c*) $(FFFF)_{16}$

Solution :

(*a*) $(4F)_{16}$ = $(?)_{10}$

= 4 F

↓ ↓

= 0100 1111

= $(1001111)_2$

Now, $(1001111)_2$ = $1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 0 \times 2^5 + 1 \times 2^6$

= $1 + 2 + 4 + 8 + 0 + 0 + 64 = (79)_{10}$

∴ $(4F)_{16}$ = $(79)_{10}$ **Ans.**

(b) $(CDE)_{16} = (?)_{10}$

$= $ C D E

$= 1100 \quad 1101 \quad 1110$

$= (110011011110)_2$

Now $(110011011110)_2 = 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 1 \times 2^6 + 1 \times 2^7 + 0 \times 2^8 + 0 \times 2^9 + 1 \times 2^{10} + 1 \times 2^{11}$

$= 0 + 2 + 4 + 8 + 16 + 0 + 64 + 128 + 0 + 0 + 1024 + 2048$

$= 3294$

$\therefore (CDE)_{16} = (3294)_{10}$ **Ans.**

(c) $(FFFF)_{16} = (?)_{10}$

$= $ F F F F

$= 1111 \quad 1111 \quad 1111 \quad 1111$

$= (1111111111111111)_2$

Now $(1111111111111111)_2 = 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 1 \times 2^5 + 1 \times 2^6 + 1 \times 2^7 + 1 \times 2^8 + 1 \times 2^9 + 1 \times 2^{10} + 1 \times 2^{11} + 1 \times 2^{12} + 1 \times 2^{13} + 1 \times 2^{14} + 1 \times 2^{15}$

$= 1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 + 1024 + 2048 + 4096 + 8192 + 16384 + 32768$

$= 65535$

$\therefore (FFFF)_{16} = (65535)_{10}$ **Ans.**

The another way of converting from hexadecimal to decimal is the direct method, where each digit is multiplied by its weight and then taking sum of products. The weights of hexadecimal number are increasing powers of 16 from right to left. *e.g.*,

....	6	5	4	3	2	1	.	1	2	3	4	5	6.....
......	16^5	16^4	16^3	16^2	16^1	16^0	.	16^{-1}	16^{-2}	16^{-3}	16^{-4}	16^{-5}	16^{-6}

EXAMPLE 2.24 : Convert the following hexadecimal numbers to decimal.

(a) $(E4)_{16}$ (b) $(B3F5)_{16}$ (c) $(FFFF)_{16}$ (d) $(3A.2F)_{16}$

Solution :

(a) $(E4)_{16} = (?)_{10}$

$= 4 \times 16^0 + E \times 16^1$

$= 4 \times 1 + 14 \times 16$ [E = 14]

$= 4 + 224$

$= (228)_{10}$

$\therefore (E4)_{16} = (228)_{10}$ **Ans.**

(b) $(B3F5)_{16} = (?)_{10}$

$= 5 \times 16^0 + F \times 16^1 + 3 \times 16^2 + B \times 16^3$ [F = 15, B = 11]

$= 5 \times 1 + 15 \times 16 + 3 \times 256 + 11 \times 4096$

$= 5 + 240 + 768 + 45056$

$= 46069$

$\therefore (B3F5)_{16} = (46069)_{10}$ **Ans.**

(c) $(FFFF)_{16} = (?)_{10}$

$= F \times 16^0 + F \times 16^1 + F \times 16^2 + F \times 16^3$

$= 15 \times 1 + 15 \times 16 + 15 \times 256 + 15 \times 4096$ [F = 15]

$= 15 + 240 + 3840 + 61440$

$= 65535$

$\therefore (FFFF)_{16} = (65535)_{10}$ **Ans.**

(d) $(3A.2F)_{16} = (?)_{10}$

$= A \times 16^0 + 3 \times 16^1 + 2 \times 16^{-1} + F \times 16^{-2}$

$= 10 \times 1 + 48 + \frac{2}{16} + \frac{15}{256}$

$= 58 + 0.1836$

$= 58.1836$

$\therefore (3A.2F)_{16} = (58.1836)_{10}$ **Ans.**

2.7.4. Decimal to Hexadecimal Conversion

For converting the decimal number to hexadecimal number system, the same technique is used as in Binary & Octal conversion. The decimal number has to be divided by the base of the number system where conversion is to be made. For binary conversion, decimal number is divided by 2, For hexadecimal conversion the decimal number is divided by its base *i.e.*, 16.

EXAMPLE 2.25 : Convert the following decimal numbers to their hexadecimal equivalent.

(a) $(94.5)_{10}$ (b) $(600.625)_{10}$ (c) $(10767)_{10}$ (d) $(37)_{10}$

Solution :

(a) $(94.5)_{10} = (?)_{16}$

16	94		
16	5	14	(E) **(LSB)**
	0	5	**(MSB)**

$0.5 \times 16 = 8.0$ | 8 **(MSB)**

$\therefore (94.5)_{10} = (5E.8)_{16}$

(b) $(600.625)_{10} = (?)_{16}$

16	600		
16	37	8	(LSB)
16	2	5	↑
	0	2	(MSB)

0.625 × 16 = 10.000 | A(10) (MSB)

∴ $(600.625)_{10} = (258.A)_{16}$ **Ans.**

(c) $(10767)_{10} = (?)_{16}$

16	10767		
16	672	F(15)	(LSB)
16	42	0	↑
16	2	A(10)	
	0	2	(MSB)

∴ $(10767)_{10} = (2A0F)_{16}$ **Ans.**

(d) $(37)_{10} = (?)_{16}$

16	37		
16	2	5	(LSB)
	0	2	(MSB)

∴ $(37)_{10} = (25)_{16}$ **Ans.**

2.7.5. Hexadecimal to Octal Conversion

Hexadecimal numbers can be converted into equivalent octal number by binary method. Firstly the hexadecimal number is converted into binary and then binary number so obtained can be converted into equivalent octal number by making grouping of three bits starting from LSB and moving towards MSB, for integer part of number and then replacing each group of three bits by its Octal representation. For fractional part, the grouping of three bits are made starting from the binary point.

EXAMPLE 2.26 : Convert the following hexadecimal numbers to octal numbers.

(a) $(B714.42)_{16}$ (b) $(4F.34)_{16}$ (c) $(0.BCBF)_{16}$ (d) $(BABA)_{16}$

Solution :

(a) $(B714.42)_{16} = (?)_8$

B 7 1 4 . 4 2

↓ ↓ ↓ ↓ ↓ ↓

1011 0111 0001 0100 . 0100 0010

= (1 011 011 100 010 100 .010 000 10)$_2$

= 001 011 011 100 010 100 . 010 000 100

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

1 3 3 4 2 4 . 2 0 4

∴ $(B714.42)_{16} = (133424.204)_8$ **Ans.**

(b) $(4F.34)_{16} = (?)_8$

4 F . 3 4

↓ ↓ ↓ ↓

0100 1111 . 0011 0100

= (01 001 111 . 001 101 00)$_2$

= 001 001 111 . 001 101 000

↓ ↓ ↓ ↓ ↓ ↓

1 1 7 . 1 5 0

∴ $(4F.34)_{16} = (117.15)_8$ **Ans.**

(c) $(0.BCBF)_{16} = (?)_8$

0 . B C B F

↓ ↓ ↓ ↓

0 . 1011 1100 1011 1111

= 0. 101 111 001 011 111 1

= 0 . 101 111 001 011 111 100

↓ ↓ ↓ ↓ ↓ ↓

0 . 5 7 1 3 7 4

∴ $(0.BCBF)_{16} = (0.571374)_8$ **Ans.**

(d) $(BABA)_{16} = (?)_8$

B A B A

↓ ↓ ↓ ↓

1011 1010 1011 1010

= 1 011 101 010 111 010

= 001 011 101 010 111 010

↓ ↓ ↓ ↓ ↓ ↓

1 3 5 2 7 2

∴ $(BABA)_{16} = (135272)_8$ **Ans.**

2.7.6. Octal to Hexadecimal Conversion

Octal to hexadecimal conversion is just reverse of hexadecimal to octal conversion. Octal number is first converted into binary and then four binary bits are grouped together starting from LSB and moving towards MSB for integer part and replacing each group of four bits by its hexadecimal representation. For fractional part, grouping of four bits are made starting from the binary point.

EXAMPLE 2.27 : Convert the following octal numbers to hexadecimal equivalent.

(a) $(247.36)_8$ (b) $(423.74)_8$ (c) $(2432)_8$

Solution :

(a) $(247.36)_8 = (?)_{16}$

$= (0\ 1010\ 0111\ .\ 0111\ 10)_2$

$= 0000\ \ 1010\ \ 0111\ .\ 0111\ \ 1000$

$\downarrow\ \ \downarrow\ \ \downarrow\ \ \ \downarrow\ \ \downarrow$

$0\ \ A\ \ 7\ .\ 7\ \ 8$

$= 0A7.78$

$\therefore (247.36)_8 = (A7.78)_{16}$ Ans.

(b) $(423.74)_8 = (?)_{16}$

$= (1\ 0001\ 0011\ .\ 1111\ 00)_2$

$= 0001\ \ 0001\ \ 0011\ .\ 1111\ \ 0000$

$\downarrow\ \ \downarrow\ \ \downarrow\ \ \ \downarrow\ \ \downarrow$

$1\ \ 1\ \ 3\ .\ F\ \ 0$

$= (113.F)_{16}$

$\therefore (423.74)_8 = (113.F)_{16}$ Ans.

(c) $(2432)_8 = (?)_{16}$

$= (0101\ 0001\ 1010)_2$

$= 0101\ \ 0001\ \ 1010$

$\downarrow\ \ \downarrow\ \ \downarrow$

$5\ \ 1\ \ A$

$= (51A)_{16}$

$\therefore (2432)_8 = (51A)_{16}$ Ans.

8. BINARY ARITHMETIC

Like decimal number system, all the arithmetic operations can be performed in case of binary ers. The basic arithmetic operations are;

ADDITION

SUBTRACTION

3. MULTIPLICATION
4. DIVISION

2.8.1. Addition

Binary addition is carried out in the same manner as decimal addition. The rules for adding binary bits are as follows :

TABLE 2.6

Binary Addition

NUMBERS		RESULT	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

First three additions result in a single bit, and addition of two 1's yield $(10)_2$ i.e. sum 0 with a carry 1 to next column as in decimal addition.

EXAMPLE :

```
   1   ← carry bit
  1011
+ 0010
------
  1101
```

In the first column, 1+0 = 1

In the 2nd column, 1+1 = 0 with a carry to next column

In the 3rd column, 1+0+0 = 1 carry from previous column

In the 4th column, 1+0 = 1

This is the final output

EXAMPLE 2.28 : Add the following binary numbers.

(a) 11001101 and 01011100

(b) 1011 and 1100

(c) 1010 and 1111

Solution : (a)

```
     111   ← carry bit
  11001101
  01011100
 ---------
 100101001
 ↑
 Carry
```

In the above example in 3rd column, 1 + 1 are added and result is 0 but a carry is given to next column. In 4th column there is addition of 1 + 1 and 1 carry bit.

∴ Sum is 1 + 1 + 1 where 1 + 1 makes 0 with carry of 1 and 1 is added to produce output as 1 and carry as 1.

```
                  1
              +   1
                  1
              -----
                 10
                  1          ──→   Carry bit
              -----
Next carry ←─    11          ──→   Output
(b)
               1011
               1100
             ------
Carry bit  ←─ 10111
(c)          ------
                11 ──→ Carry
               1010
               1111
             ------
Carry ←─      11001
             ------
```

EXAMPLE 2.29 : Add the binary numbers.

```
          01101010
          00001000
          10000001
          11111111
Solution : --------
          1 ──→ Secondary carry generated for previous column
carry row  1 1 1 1 1 1 1  ──→ Carry generated from previous
Ist row    0 1 1 0 1 0 1 0        column
2nd row    0 0 0 0 1 0 0 0
3rd row    1 0 0 0 0 0 0 1
4th row    1 1 1 1 1 1 1 1
          -----------------
Carry ──→ 11 1 1 1 0 0 1 0
          -----------------
```

∴ Sum = 111110010 Ans.

2.8.2. SUBTRACTION

Subtraction in binary numbers must be performed as in the decimal numbers. Rules for subtraction are given in following table 2.7.

TABLE 2.7

Binary Subtraction

A	B	DIFFERENCE	BORROW
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

While subtracting the number, we do require to borrow from the next higher column. A borrow is required in binary only when we try to subtract 1 from 0. In this case, when a 1 is borrowed from next higher column and difference is also 1.

EXAMPLE 2.30 : Subtract 3_{10} from 8_{10}

Solution : $8_{10} = 1000_2$ and $3_{10} = 0011_2$

```
∴       1000
      - 0011
      ------
         101     = (5)10  Ans.
      ------
```

EXAMPLE 2.31 : Perform the following subtraction.

```
           1011
         - 0110
         ------
Solution : 1011
         - 0110
         ------
           0101   Ans.
         ------
```

In column three, 0 – 1 = 1 and 1 is borrowed from 4th column

∴ In fourth column 0 – 0 = 0.

2.8.3. Multiplication

Multiplication in binary numbers is similar to that in decimal numbers. The rules for multiplication are in table 2.8.

TABLE 2.8

Binary Multiplication

A	B	OUTPUT
0	0	0
0	1	0
1	0	0
1	1	1

Multiplication involves forming the partial products, shifting each successive partial produ left by one place and adding all the products.

EXAMPLE 2.32 : Multiply 10111_2 by 101_2

```
Solution :   10111
           ×   101
           -------
             10111
            00000
           10111
           -------
           1110011₂
           -------
```

EXAMPLE 2.33 : Perform the following multiplication.

(a) 1101_2 by 101_2 (b) 1011_2 by 1001_2 (c) 1001_2 by 1101_2

Solution :

(a)

```
      1101
    × 101
    ------
      1101
     0000
    1101
    ------
   1000001₂   Ans.
```

(b)

```
      1011
    ×1001
    ------
      1011
     0000
    0000
   1011
   -------
   1100011₂   Ans.
```

(c)

```
      1001
    ×1101
    ------
      1001
     0000
    1001
   1001
   -------
   1110101₂   Ans.
```

2.8.4. Division

Division in binary numbers is carried out in same way as in decimal numbers. The rules for division are

$0 \div 1 = 0$

$1 \div 1 = 1$

Division by zero is meaningless.

Example : 2.34 : Divide 1110101 by 1001.

1001 ⟶ **Divisor**

1110101 ⟶ **Divident**


```
        1101 ──→ Quotient
  1001 )1110101
        1001
        ----
        1011
        1001
        ------
        001001
          1001
          ----
          0000 ──→ Remainder
```

$\therefore$ **$1110101_2 \div 1001_2 = 1101_2$ Ans.**

EXAMPLE 2.35 : Perform the following division.

(a) 11000_2 by 1000_2 (b) 1111000_2 by 100_2 (c) $11110_2 \div 110_2$

Solution :

(a) $11000_2 \div 1000_2$

```
          11
  1000 ) 11000
         1000
         ----
          1000
          1000
          ----
            ×
```

$\therefore$ **$11000_2 \div 1000_2 = 11_2$ Ans.**

(b) $1111000_2 \div 100_2$

```
        11110
  100 )1111000
       100
       ---
        111
        100
        ---
         110
         100
         ---
          100
          100
          ---
           000
           000
           ---
             0
```

$\therefore$ **$1111000_2 \div 100_2 = 11110_2$ Ans.**

(c) $11110_2 \div 110_2$

```
          11
110  ) 11110
       110
       ---
        110
        110
        ---
         ×
```

$\therefore$ $11110_2 \div 110_2 = 11_2$ **Ans.**

2.9. SIGN MAGNITUDE METHOD OF REPRESENTATION

Digital systems must be able to handle both positive and negative numbers. A ***signed binary number** consists of both sign and magnitude information.* The sign indicates whether a number is positive or negative and the magnitude is the value of the number. There are three forms in which signed number can be represented in binary

(*i*) Sign magnitude (*ii*) 1's complement (*iii*) 2's complement

The Sign Bit : The left-most bit in a signed binary number is the **sign bit**, which tells you whether the number is positive or negative.

If the **sign bit is a 0,** *the number* is **positive. If it is a 1,** *the number is* **negative.** *e.g.*

0	1	0	1	0	1	0	= + 42

↑ ← Magnitude bits →

Sign bit

↓

1	1	0	1	0	1	0	= – 42

← Magnitude bits →

In the sign-magnitude form, a negative number has the same magnitude bits as the corresponding positive number, but the sign bit is a 1 rather than a zero.

2.10. BINARY ARITHMETIC

2.10.1. One's Complement Representation

The 1's complement of a binary number is obtained by changing each 0 to 1 and each 1 to 0. Both the numbers complement each other. If one of these number is positive, the other will be negative with same magnitude and vice versa *e.g.*

Binary number	1's complement
101011	010100
110101	001010
010100	101011
101110000101	010001111010

EXAMPLE 2.37 : Find the 1's complement of following binary numbers.

(*a*) 1000100101 **(*b*) 110100010** **(*c*) 100000001** **(*d*) 1100111100010001**

Solution :

(*a*) 0111011010 (*b*) 001011101 (*c*) 0111111110 (*d*) 0011000011101110

2.10.2. Addition and subtraction using 1's complement method

Addition and subtraction can be performed using 1's complement method. While performing direct subtraction, it is not an easy task so the other way of performing it is 1's complement method. To **subtract a smaller number $(10110)_2$ from larger number $(11101)_2$, the method is :**

STEP 1 : Find the 1's complement of smaller number

Smaller number $= 10110_2$

1's complement $= 01001_2$

STEP 2 : Add 1's complement to the larger number

```
                 11101
1's complement   01001
                 -----
Carry  ←——   (1) 00110
```

STEP 3 : Remove the carry and add it to result.

```
  00110
      1
  -----
  00111  Ans.
```

Direct method :

```
  11101
  10110
  -----
  00111  Ans.
```

To subtract a larger number from smaller number, the method is as follows

STEP 1 : Find the 1's complement of larger number

```
    10001
  –  1101
  -------
       ?
```

Larger number $= 1101_2$

1's complement $= 0010_2$

STEP 2 : Add 1's complement to the smaller number

1001_2

0010_2

1011_2

STEP 3 : As there is no carry. Therefore complement the answer and write the opposite sign

i.e. – 0100 Answer

2.10.3. 2's Complement representation

If 1 is added to 1's complement of a number than it will obtain the 2's complement of the number.

e.g. : Obtain the 2's complement of 101101

1's complement = 010010

+ 1

$\therefore$ 2's complement = 010011

EXAMPLE 2.38 : Find the 2's complement of following numbers.

(a) **01100101** *(b)* **1101101** *(c)* **0010111** *(d)* **11011101**

Solution :

(a) 1's complement of 01100101 is 10011010

∴ its 2's complement is 10011010 + 1 = 10011011

(b) 1's complement of 1101101 is 0010010

∴ its 2's complement is 0010010 + 1 = 0010011

(c) 1's complement of 0010111 is 1101000

∴ its 2's complement is 1101000 + 1 = 1101001

(d) 1's complement of 11011101 is 00100010

∴ its 2's complement is 00100010 + 1 = 00100011

2.10.4. 2's Complement Subtraction

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

(1) **Find the 2's complement of smaller number.**
(2) **Add the 2's complement of larger number.**
(3) **Discard the carry.**

EXAMPLE 2.39 : Subtract 11010 from 11101 using 2's complement method.

Solution :

Direct method

```
 11101
 11010
 -----
 00011
```

2's Complement Method :

```
11010 = 00101
           +1
2's complement = 00110
        11101
        00110
Carry ← (1)00011 = 00011 Answer
```


To subtract a larger number from smaller one, the method is as follows ;

1. Find the 2's complement of larger number.
2. Add 2's complement to smaller number.
3. There is no carry. The result is in 2's complement form and negative.
4. Take the 2's complement of result and change the sign.

EXAMPLE 2.40 : Subtract 111101_2 from 110000_2 using 2's complement method.

Solution :

```
  110000
 -111101
2's complement of 111101 = 000010
                              +1
                          000011
Add it to smaller number
  110000
  000011
  ------
  110011
```

As there is no carry. Obtain the 2's complement and add negative sign.

```
= 001100
       1
- 001101 Ans.
```

2.11. FLOATING POINT NUMBERS

The floating point number system, based on scientific notation, is capable of representing very large and very small numbers without an increase in the number of bits and also for representing numbers that have both integer and fractional components.

A **floating point number** *(real number) consists of two parts plus a sign.* The **mantissa** is the part of a floating point number that represents the magnitude of the number. The **exponent** is the part of a floating point number that represent the number of places the decimal point (*or binary point*) is to be moved.

Binary floating point numbers are of three forms : **Single precision (32 bits), double precision (64 bits) and extended precision (80 bits)**

Single precision is the standard format and it has sign bit (S) as the left most bit, the exponent (E), includes the next eight bits and the mantissa or fractional part (F) includes the remaining 23 bits.

S	Exponent (E)	Mantissa (fraction, F)
1 bit	8 bits	23 bits

(32 Bits)

e.g. To illustrate how a binary number is expressed in floating point format, let's use 1011010010001 as an example.

$$1011010010001 = 1{\cdot}011010010001 \times 2^{12}$$

Assuming that this is a positive number, the sign bit (S) is 0.

2.12. BCD NUMBERS

BCD stands for **Binary Coded Decimal**. It is a method of representing binary numbers using binary digits. BCD is a weighted code. In weighted code, each successive digit from right to left, represents weights equal to some specified value. 8421 code is a type of BCD code and is composed of four bits representing the decimal digits 1 through 9. The designation 8421 indicates the binary weight of four bits ($2^3, 2^2, 2^1, 2^0$). BCD means that each decimal digit is represented by a binary code of four bits.

e.g. 2346 will be represented in BCD as follows ;

2 ↓ 0010 3 ↓ 0011 4 ↓ 0100 6 ↓ 0110

= 0010001101000110 **Ans.**

Each digit is represented individually.

e.g. $(35)_{10}$ is represented in binary as

2	35	
2	12	1
2	6	0
2	3	1
2	1	1
	0	1

(LSB) ↑ (MSB) $= (11101)_2$

But in BCD (35) = 3 ↓ 0011 5 ↓ 0101 = $(110101)_{BCD}$ **Ans.**

Which is a different value.

8421 code is similar to binary number upto 9. But after 9 it is different code.

EXAMPLE 2.41 : Convert the following numbers to BCD form.

(*a*) 473_{10} **(*b*) 312_{10}** (*c*) 257_{10} **(*d*) 112_{10}**

Solution : (*a*) $(473)_{10} = (?)_{BCD}$

4 ↓ 0100 7 ↓ 0111 3 ↓ 0011

= $(10001110011)_{BCD}$ **Ans.**

(*b*) $(312)_{10} = (?)_{BCD}$

= 3 ↓ 0011 1 ↓ 0001 2 ↓ 0010

= $(1100010010)_{BCD}$ **Ans.**

(*c*) $(257)_{10} = (?)_{BCD}$

= 2 ↓ 0010 5 ↓ 0101 7 ↓ 0111

= $(1001010111)_{BCD}$ **Ans.**

(*d*) $(112)_{10} = (?)_{BCD}$

= 1 ↓ 0001 1 ↓ 0001 2 ↓ 0010

= $(100010010)_{BCD}$ **Ans.**

EXAMPLE 2.42 : Convert the following BCD numbers to their decimal equivalent

(*a*) 10000110 **(*b*) 00110010.10010100** **(*c*) 10010010.0011**

Solution : (*a*) BCD ⟶ 1000 ↓ 8 0110 ↓ 6 = $(86)_{10}$ **Ans.**

(*b*) BCD ⟶ 0011 ↓ 3 0010 ↓ 2 . 1001 ↓ 9 0100 ↓ 4 = $(32.94)_{10}$ **Ans.**

(*c*) BCD ⟶ 1001 ↓ 9 0010 ↓ 2 . 0011 ↓ 3 = $(92.3)_{10}$ **Ans.**

2.13. BCD ADDITION

BCD is a four bit code and many applications require arithmetic operations to be performed. Addition of BCD is performed by adding two bits of binary, starting from LSB. The steps to add BCD numbers are as follows ;

1. Add the two numbers using rules of binary addition.
2. If a four bit sum is equal or less than 9, it is a valid BCD number.
3. If a four bit sum is greater than 9 or if a carry out of the group is generated, it is an invalid result. Add 6(0110) to the four bit sum. If a carry results when 6 is added, add the carry to the next four bit group.

Example : 2.43 : Add the following BCD numbers.

(*a*) 0010 and 0011 **(*b*) 00100011 and 00010101**

Solution : (*a*)

```
   0010
 + 0011
 ------
   0101   Ans.
```

(*b*)

```
   00100011
 + 00010101
 ----------
   00110111   Ans.
```

EXAMPLE 2.44 : Add the following BCD numbers.

(a) 1001 and 0100 (b) 1001 and 1000

(c) 00010110 and 00010101 (d) 01100111 and 01010011

Solution :

(a)

```
            1001
          + 0100
          + 1101  ──→  Invalid BCD number (because > 9)
Add 6       0110
0001        0011  ──→  Valid BCD number
 ↓           ↓
 1           3     = (13)10   Ans.
```

(b)

```
            1001
          + 1000
           10011  ──→  Invalid because of carry
        +   0110  ──→  Add 6
0001        1001
 ↓           ↓
 1           8     = (18)10   Ans.
```

(c)

```
                          0001  0110
                          0001  0101
                          0010  1011  ──→  Right four bits are invalid because > 9
                                           where as left four bits are valid
Add 6 to right     ──→        + 0110
four bits and carry       0011  0001  ──→  Valid BCD number
                           ↓     ↓
                           3     1   = (31)10
```

(d)

```
          0110  0111
      +   0101  0011
          1011  1010  ──→  Both groups are invalid > 9
      +   0110  0110       Add 6 to both groups
0001      0010  0000
 ↓         ↓     ↓
 1         2     0    = (120)10. Ans.
```