Number System: Number system is technique to work with numbers. We are familiar with decimal number system having digit value 0-9. Other popular number systems include binary number system, octal number system, hexadecimal number system, etc.

Decimal Number System

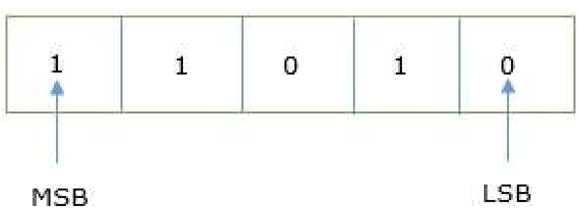
Decimal number system have base 10 with digit 0-9. Also known as positional value system i.e means that the value of digits will depend on its position

- Ex: In 734, value of 7 is 7 hundreds or 700 or 7 × 100 or 7×10^2
- In 207, value 0f 7 is 7 units or 7 or 7 × 1 or 7×10^0
- The weightage of each position are as under −

10^5	10^4	10^3	10^2	10^1	10^0

Binary Number System: Binary number system have base 2 with digit 0-1. Each binary digit is also called a bit. Binary number system is also based on positional value system, where each digit has a value expressed in powers of 2, as displayed here.

In any binary number, the rightmost digit is called least significant bit (LSB) and leftmost digit is called most significant bit (MSB).

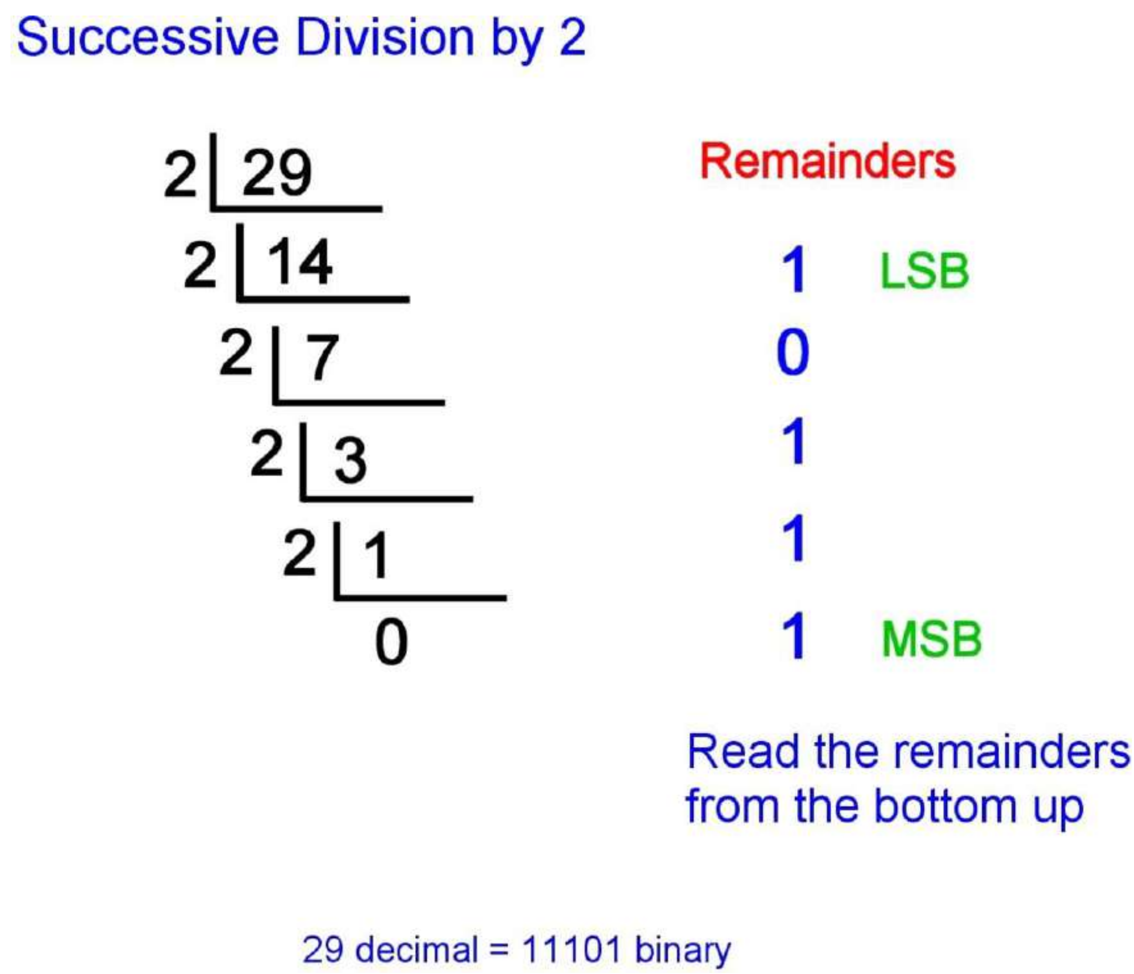


$11010_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

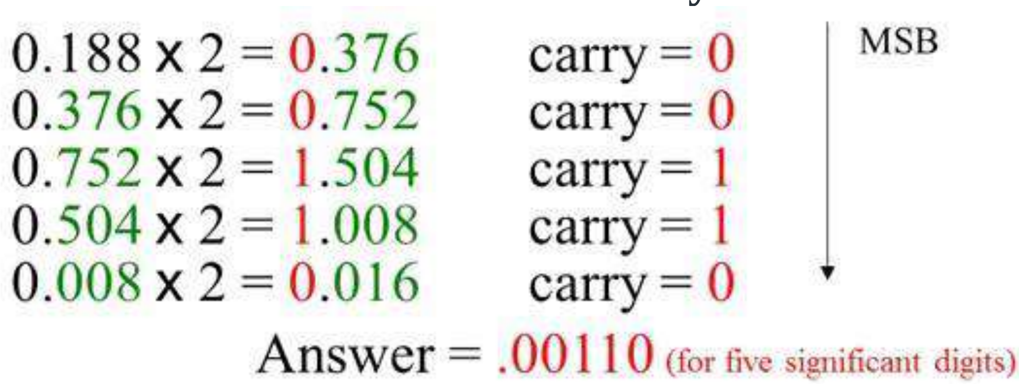
$= 16 + 8 + 0 + 2 + 0$

$= 26_{10}$

Decimal to binary conversion



Convert decimal fraction to binary number



Let's take an example for n = 4.47 k = 3

Step 1: Conversion of 4 to binary

1. 4/2 : Remainder = 0 : Quotient = 2
2. 2/2 : Remainder = 0 : Quotient = 1
3. 1/2 : Remainder = 1 : Quotient = 0

So equivalent binary of integral part of decimal is 100.

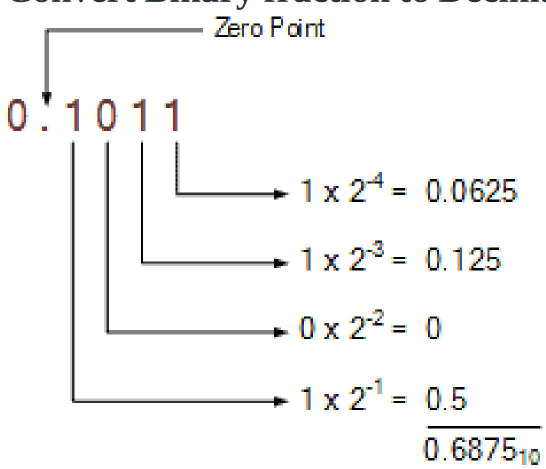
Step 2: Conversion of .47 to binary
1. 0.47 * 2 = 0.94, Integral part: 0
2. 0.94 * 2 = 1.88, Integral part: 1
3. 0.88 * 2 = 1.76, Integral part: 1

So equivalent binary of fractional part of decimal is .011

Step 3: Combined the result of step 1 and 2.

Final answer can be written as:
100 + .011 = 100.011

Convert Binary fraction to Decimal



Let's take an example for n = 110.101

Step 1: Conversion of 110 to decimal
=> $110_2 = (1*2^2) + (1*2^1) + (0*2^0)$
=> $110_2 = 4 + 2 + 0$
=> $110_2 = 6$
So equivalent decimal of binary integral is 6.

Step 2: Conversion of .101 to decimal
=> $0.101_2 = (1*1/2) + (0*1/2^2) + (1*1/2^3)$
=> $0.101_2 = 1*0.5 + 0*0.25 + 1*0.125$
=> $0.101_2 = 0.625$
So equivalent decimal of binary fractional is 0.625

Step 3: Add result of step 1 and 2.
=> 6 + 0.625 = 6.625

Octal Number System

Octal number system has eight digits – 0, 1, 2, 3, 4, 5, 6 and 7 i.e. base8. Octal number system is also a positional value system with where each digit has its value expressed in powers of 8, as shown here
–

8^5	8^4	8^3	8^2	8^1	8^0

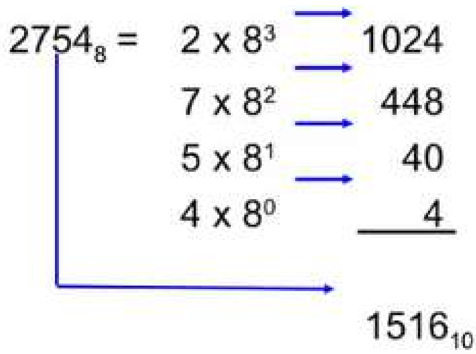
Conversion from Decimal to Octal number system

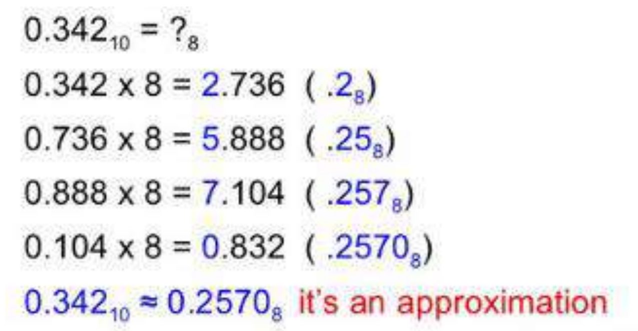
Convert decimal number 210 into octal number.

Since given number is decimal integer number, so by using above algorithm performing short division by 8 with remainder.

Division	Remainder (R)
210 / 8 = 26	2
26 / 8 = 3	2
3 / 8 = 0	3

Conversion from octal to decimal number system





$0.342_{10} = ?_8$

$0.342 \times 8 = 2.736 \quad (.2_8)$

$0.736 \times 8 = 5.888 \quad (.25_8)$

$0.888 \times 8 = 7.104 \quad (.257_8)$

$0.104 \times 8 = 0.832 \quad (.2570_8)$

$0.342_{10} \approx 0.2570_8$ it's an approximation

Convert octal number 7.12172 into decimal form

$= 7x8^0 + 1x8^{-1} + 2x8^{-2} + 1x8^{-3} + 7x8^{-4} + 2x8^{-5}$

= 7 + 0.125 + 0.03125 + 0.001953125 + 0.001708984375 + 0.00006103515624

= 10.1599...

= 10.16 (approx. value)

Hexadecimal Number System

Hexadecimal number system has 16 symbols It is base 16 which has only 16 symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and A, B, C, D, E, F. These A, B, C, D, E, F use as single digit in place of double digits, 10, 11, 12, 13, 14, 15 respectively.

Example − Convert decimal number 540 into hexadecimal number.

Division	Remainder (R)
540 / 16 = 33	12 = C

33 / 16 = 2	1
2 / 16 = 0	2
0 / 16 = 0	0

Now, write remainder from bottom to up (in reverse order), this will be 021C (or only 21C) which is equivalent hexadecimal number of decimal integer 540.

Example − Convert decimal fractional number 0.06640625 into hexadecimal number.

Since given number is decimal fractional number, so by using above algorithm performing short multiplication by 16 with integer part.

Multiplication	Resultant integer part
0.06640625 x 16=1.0625	1
0.0625 x 16 =1.0	1
0 x 16=0.0	0

Now, write these resultant integer part, this will be approximate 0.110 which is equivalent hexadecimal fractional number of decimal fractional 0.06640625.

Example − Convert Hexadecimal ABC into decimal.

C = 12 * (16 ^ 0) 12

B = 11 * (16 ^ 1) 176

A = 10 * (16 ^ 2) 2560

Then, we add the results.

2560 + 176 + 12 = 2748_{10} ***decimal***

Example-2 − Convert hexadecimal number 1F.01B into decimal number.

Since value of Symbols: B and F are 11 and 15 respectively. Therefore equivalent decimal number is,

= $(1F.01B)_{16}$
= $(1x16^{1}+15x16^{0} +0x16^{-1}+1x16^{-2}+11x16^{-3})_{10}$
= $(31.0065918)_{10}$

Convert binary number into octal

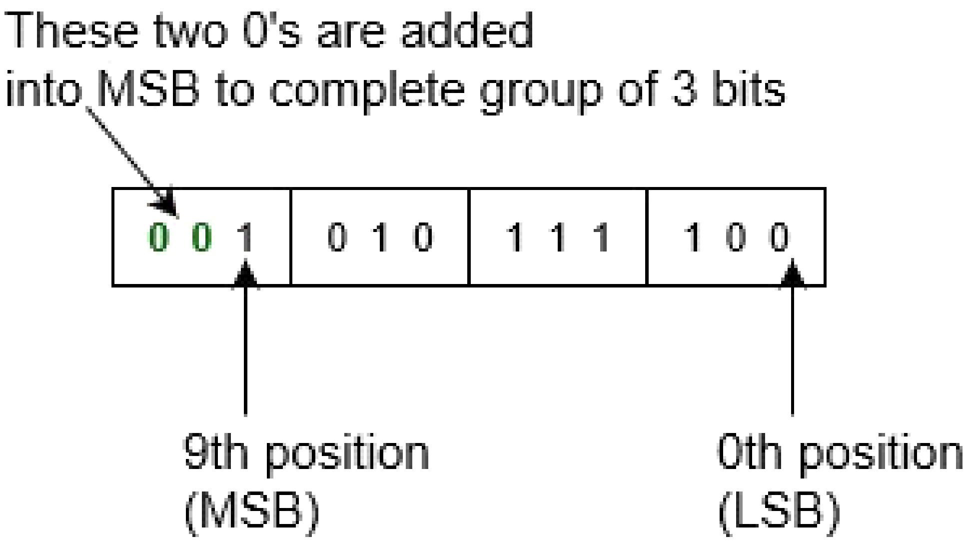
Example − Convert binary number 10010110 into octal number.

Method-1 First convert binary into decimal, convert resultant decimal into octal

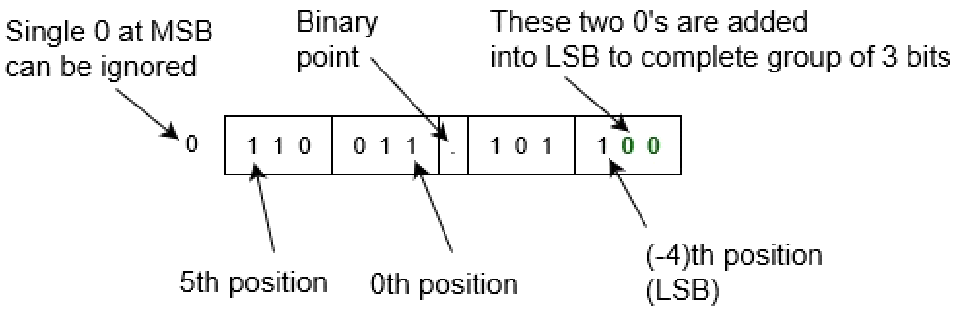
First convert this into decimal number
= $(10010110)_{2}$
= $1x2^{7}+0x2^{6}+0x2^{5}+1x2^{4}+0x2^{3}+1x2^{2}+1x2^{1}+0x2^{0}$
= 128+0+0+16+0+4+2+0
= $(150)_{10}$
Then, convert it into octal number
= $(150)_{10}$
= $2x8^{2}+2x8^{1}+6x8^{0}$
= $(226)_{8}$ which is answer.

Method-2 Grouping method:

- Take binary number
- Divide the binary digits into groups of three (starting from right) for integer part and start from left for fraction part.
- Convert each group of three binary digits to one octal digit.



Example-2 Convert binary number 0110 011.1011 into octal number. Since there is binary point here and fractional part. So,



= $(0110\ 011.1011)_2$
= $(0\ 110\ 011.\ 101\ 1)_2$
= $(110\ 011.\ 101\ 100)_2$
= $(6\ 3.\ 5\ 4)_8$
= $(63.54)_8$

Conversion from Octal to Binary number system

Example-1 Convert octal number 540 into binary number.

- Take Octal number as input

- Convert each digit of octal into binary.
- That will be output as binary number.

= $(540)_8$
= $(101\ 100\ 000)_2$
= $(101100000)_2$

Example-2 − Convert octal number 352.563 into binary number.

According to above algorithm, equivalent binary number will be,

= $(352.563)_8$
= $(011\ 101\ 010\ .\ 101\ 110\ 011)_2$
= $(011101010.101110011)_2$

Conversion from Binary to Hexadecimal number system

Example − Convert binary number 1101010 into hexadecimal number.

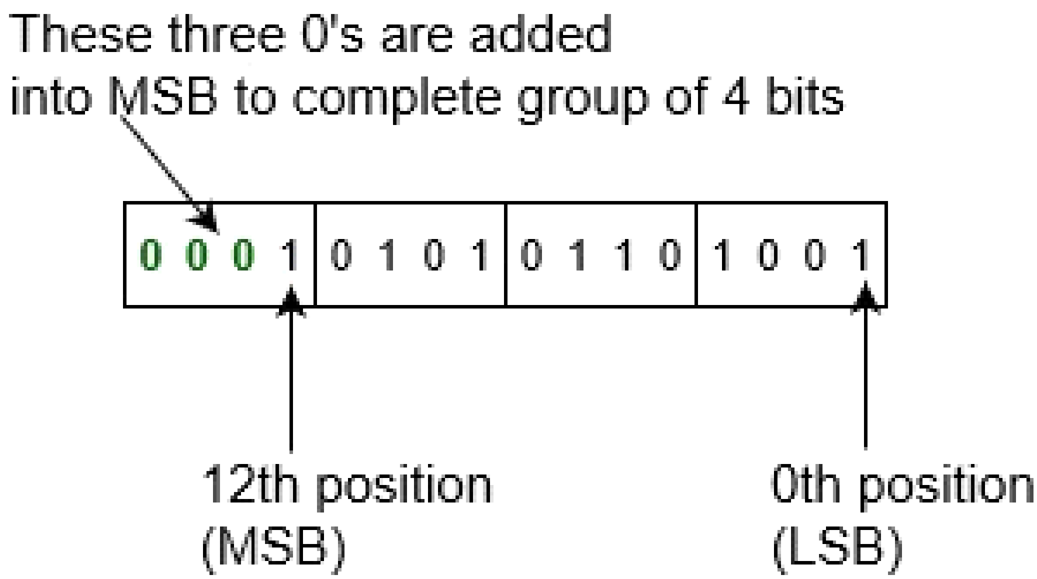
Method-1 First convert binary into decimal, convert resultant decimal into hexadecimal

First convert this into decimal number:
= $(1101010)_2$
= $1x2^6+1x2^5+0x2^4+1x2^3+0x2^2+1x2^1+0x2^0$
= 64+32+0+8+0+2+0
= $(106)_{10}$
Then, convert it into hexadecimal number
= $(106)_{10}$
= $6x16^1+10x16^0$
= $(6A)_{16}$

Method-2 use grouping method

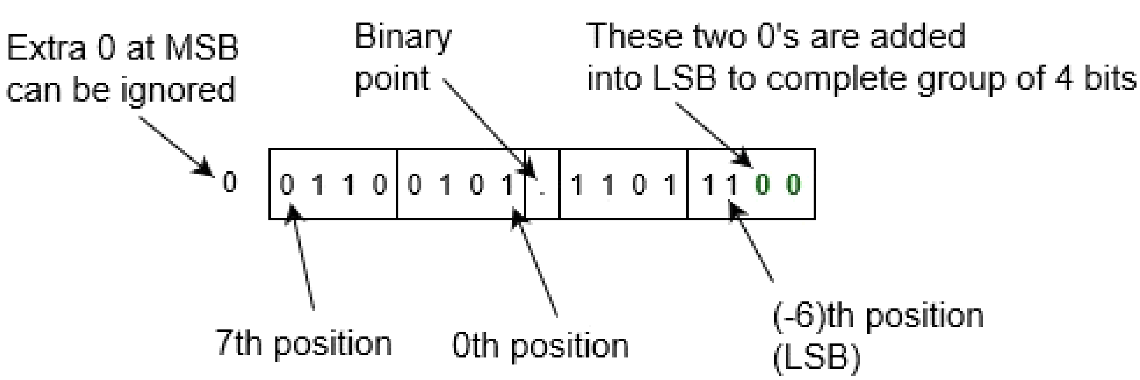
- Take binary number
- Divide the binary digits into groups of four (starting from right) for integer part and start from left for fraction part.
- Convert each group of four binary digits to one hexadecimal digit.

Example: Convert binary number 1010101101001 into hexadecimal number. Since there is no binary point here and no fractional part. So,



= $(1010101101001)_2$
= $(1\ 0101\ 0110\ 1001)_2$
= $(0001\ 0101\ 0110\ 1001)_2$
= $(1\ 5\ 6\ 9)_{16}$
= $(1569)_{16}$

Example-2 − Convert binary number 001100101.110111 into hexadecimal number



Therefore, Binary to hexadecimal is,

$= (001100101.110111)_2$
$= (0\ 0110\ 0101.\ 1101\ 1100)_2$
$= (0110\ 0101.\ 1101\ 1100)_2$
$= (6\ 5.\ D\ C)_{16}$
$= (65.DC)_{16}$

Convert hexadecimal into binary

Q: Convert $A2B_{16}$ to an equivalent binary number.

Method-1 convert hexa decimal into decimal,, convert resultant decimal into hexadecimal

Given hexadecimal number = $A2B_{16}$

First, convert the given hexadecimal to the equivalent decimal number.

$A2B_{16} = (A \times 16^2) + (2 \times 16^1) + (B \times 16^0)$

$= (A \times 256) + (2 \times 16) + (B \times 1)$

$= (10 \times 256) + 32 + 11$

$= 2560 + 43$

= 2603(Decimal number)

Now we have to convert 2603_{10} to binary

```
2 | 2603
2 |1301 -- 1
2 | 650 -- 1
2 | 325 -- 0
2 | 162 -- 1
2 | 81  -- 0
2 | 40 -- 1
2 | 20 -- 0
2 | 10 -- 0
2 | 5 -- 0
2 | 2 -- 1
2 | 1 -- 0
2 | 0 -- 1
```

The binary number obtained is 101000101011_2

Hence, $A2B_{16} = 101000101011_2$

Method-2 Grouping method

$A2B_{16}$

(10) (2) (11)

Convert into binary

(1010) (10) (1011) convert into group of 4 since $(2)^4 = 16$

(1010) (0010) (1011)

$(101000101011)_2$

Binary Addition

Four rules of binary addition.

Case	A + B	Sum	Carry
1	0 + 0	0	0
2	0 + 1	1	0
3	1 + 0	1	0
4	1 + 1	0	1

Addition Table

In fourth case, a binary addition is creating a sum of (1 + 1 = 10) i.e. 0 is written in the given column and a carry of 1 over to the next column.

Example − Addition

Addition Example

0011010 + 001100 = 00100110

```
  1 1          carry
 0 0 1 1 0 1 0   = 26₁₀
+0 0 0 1 1 0 0   = 12₁₀
---------------
 0 1 0 0 1 1 0   = 38₁₀
```

Example2:

	1				1	
		1	0	0	0	1
+		1	1	1	0	1
	1	0	1	1	1	0

Binary Subtraction

Four rules of binary substraction are as below

Case	A - B	Subtract	Borrow
1	0 - 0	0	0
2	1 - 0	1	0
3	1 - 1	0	0
4	0 - 1	0	1

Example − Subtraction

Subtraction Example

0011010 - 001100 =00001110

1 1	borrow
0 0 ~~1 1~~ 0 1 0	$= 26_{10}$
-0 0 0 1 1 0 0	$= 12_{10}$
0 0 0 1 1 1 0	$= 14_{10}$

Binary Multiplication

Case	A x B	Multiplication
1	0 x 0	0
2	0 x 1	0
3	1 x 0	0
4	1 x 1	1

Example − Multiplication

Multiplication Example

Example:

0011010 x 001100 = 100111000

```
     0011010    = 26₁₀
   x 0001100    = 12₁₀
  ------------
     0000000
    0000000
   0011010
  0011010
  ------------
  0100111000    = 312₁₀
```

Binary Division

Binary division is similar to decimal division. It is called as the long division procedure.

Example − Division

Division Example

101010 / 000110 = 000111

```
              111       = 7₁₀
         ----------
  000110 ) 101010       = 42₁₀
          -110          = 6₁₀
          -----
           1001
          -110
          -----
             110
            -110
            ----
               0
```

Boolean Algebra is used to analyze and simplify the digital (logic) circuits. It uses only the binary numbers i.e. 0 and 1. It is also called as Binary Algebra or logical Algebra. Boolean algebra was invented by George Boole in 1854.

Important Rule in Boolean Algebra

Rule in Boolean Algebra

1. Variable used can have only two values. Binary 1 for HIGH and Binary 0 for LOW.

2. Complement of a variable is represented by an overbar (-). Thus, complement of variable B is represented as B Bar. Thus if B = 0 then B Bar = 1 and B = 1 then B Bar = 0.

3. Logical OR of the variables is represented by a plus (+) sign between them. For example ORing of A, B, C is represented as A + B + C.

4. Logical AND of the two or more variable is represented by writing a dot between them such as A.B.C. Sometime the dot may be omitted like ABC.

Logic Gates

Logic gates are the basic building blocks of any digital system. It is an electronic circuit having one or more than one input and only one output. The relationship between the input and the output is based on a **certain logic**. Based on this, logic gates are named as AND gate, OR gate, NOT gate etc.

AND Gate

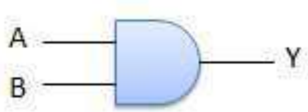
A circuit which performs an AND operation is shown in figure. It has n input (n >= 2) and one output.

Y = A AND B AND C .... N

Y = A.B.C .... N

Y = ABC .... N

Logic diagram



Truth Table

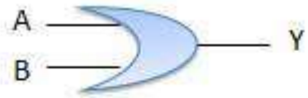
Inputs		Output
A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

A circuit which performs an OR operation is shown in figure. It has n input (n >= 2) and one output.

Y = A OR B OR C .... N
Y = A + B + C .... N

Logic diagram



Truth Table

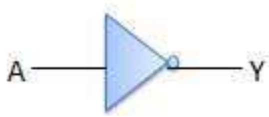
Inputs		Output
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate

NOT gate is also known as **Inverter**. It has one input A and one output Y.

Y = NOT A
Y = $\overline{A}$

Logic diagram



Truth Table

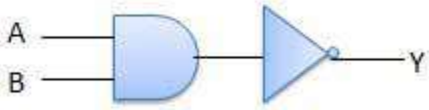
Inputs	Output
A	B
0	1
1	0

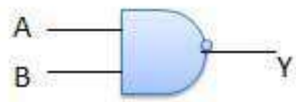
NAND Gate

A NOT-AND operation is known as NAND operation. It has n input (n >= 2) and one output.

Y = A NOT AND B NOT AND C .... N
Y = A NAND B NAND C .... N

Logic diagram





Truth Table

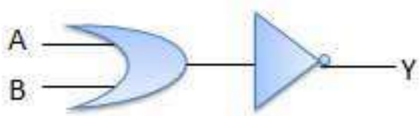
Inputs		Output
A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

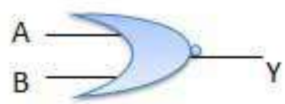
NOR Gate

A NOT-OR operation is known as NOR operation. It has n input (n >= 2) and one output.

Y = A NOT OR B NOT OR C .... N
Y = A NOR B NOR C .... N

Logic diagram





Truth Table

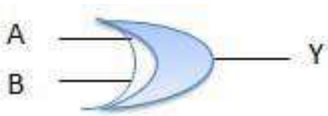
Inputs		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

XOR Gate

XOR or Ex-OR gate is a special type of gate. It can be used in the half adder, full adder and subtractor. The exclusive-OR gate is abbreviated as EX-OR gate or sometime as X-OR gate. It has n input (n >= 2) and one output.

Y = A XOR B XOR C .... N
Y = $A \oplus B \oplus C$ .... N
Y = $\overline{AB} + \overline{AB}$

Logic diagram



Truth Table

Inputs		Output
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

XNOR Gate

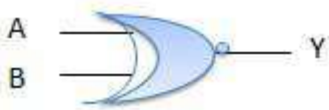
XNOR gate is a special type of gate. It can be used in the half adder, full adder and subtractor. The exclusive-NOR gate is abbreviated as EX-NOR gate or sometime as X-NOR gate. It has n input (n >= 2) and one output.

$$Y = A \text{ XOR } B \text{ XOR } C \ldots\ldots. N$$

$$Y = A \odot B \odot C \ldots\ldots. N$$

$$Y = \overline{A}\,\overline{B} + AB$$

Logic diagram



Truth Table

Inputs		Output
A	B	$A \odot B$
0	0	1
0	1	0
1	0	0
1	1	1

Page no:31,38,46 onwards on Decoder, Encoder & MUX