BINARY NUMBER

NUMBER SYSTEM

Number systems are the technique to represent numbers in the computer system architecture, every value that you are saving or getting into/from computer memory has a defined number system.

Computer architecture supports following number systems.

- **Binary number system**
- **Octal number system**
- **Decimal number system**
- **Hexadecimal (hex) number system**

BINARY NUMBER SYSTEM

A Binary number system has only two digits that are **0 and 1**. Every number (value) represents with 0 and 1 in this number system. The base of binary number system is 2, because it has only two digits.

OCTAL NUMBER SYSTEM

Octal number system has only eight (8) digits from **0 to 7**. Every number (value) represents with 0,1,2,3,4,5,6 and 7 in this number system. The base of octal number system is 8, because it has only 8 digits.

DECIMAL NUMBER SYSTEM

Decimal number system has only ten (10) digits from **0 to 9**. Every number (value) represents with 0,1,2,3,4,5,6, 7,8 and 9 in this number system. The base of decimal number system is 10, because it has only 10 digits.

HEXADECIMAL NUMBER SYSTEM

A Hexadecimal number system has sixteen (16) alphanumeric values from **0 to 9** and **A to F**. Every number (value) represents with 0,1,2,3,4,5,6, 7,8,9,A,B,C,D,E and F in this number system. The base of hexadecimal number system is 16, because it has 16 alphanumeric values. Here A is 10, B is 11, C is 12, D is 14, E is 15 and F is 16.

Number system	Base(Radix)	Used digits	Example
Binary	2	0,1	$(11110000)_2$
Octal	8	0,1,2,3,4,5,6,7	$(360)_8$
Decimal	10	0,1,2,3,4,5,6,7,8,9	$(240)_{10}$
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F	$(F0)_{16}$

CONVERSIONS

DECIMAL TO OTHER

1. DECIMAL TO BINARY

Decimal Number System to Other Base

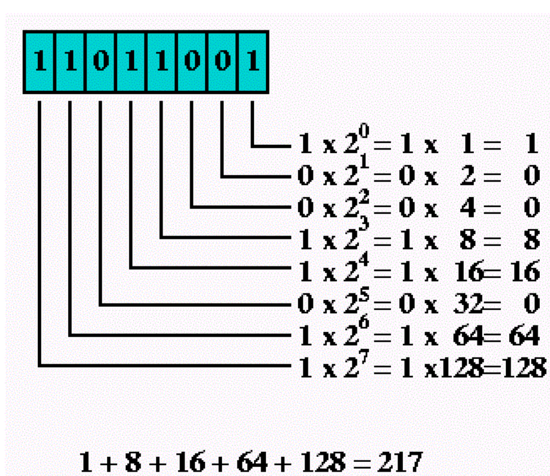
To convert Number system from **Decimal Number System** to **Any Other Base** is quite easy; you have to follow just two steps:
A) Divide the Number (Decimal Number) by the base of target base system (in which you want to convert the number: Binary (2), octal (8) and Hexadecimal (16)).
B) Write the remainder from step 1 as a Least Signification Bit (LSB) to Step last as a Most Significant Bit (MSB).

<table>
<tr><td>Decimal to Binary Conversion</td><td>Result</td></tr>
<tr><td>Decimal Number is : $(12345)_{10}$
<table>
<tr><td>2</td><td>12345</td><td>1</td><td>LSB</td></tr>
<tr><td>2</td><td>6172</td><td>0</td><td></td></tr>
<tr><td>2</td><td>3086</td><td>0</td><td></td></tr>
<tr><td>2</td><td>1543</td><td>1</td><td></td></tr>
<tr><td>2</td><td>771</td><td>1</td><td></td></tr>
<tr><td>2</td><td>385</td><td>1</td><td></td></tr>
<tr><td>2</td><td>192</td><td>0</td><td></td></tr>
<tr><td>2</td><td>96</td><td>0</td><td></td></tr>
<tr><td>2</td><td>48</td><td>0</td><td></td></tr>
<tr><td>2</td><td>24</td><td>0</td><td></td></tr>
<tr><td>2</td><td>12</td><td>0</td><td></td></tr>
<tr><td>2</td><td>6</td><td>0</td><td></td></tr>
<tr><td>2</td><td>3</td><td>1</td><td></td></tr>
<tr><td></td><td>1</td><td>1</td><td>MSB</td></tr>
</table>
</td><td>Binary Number is $(11000000111001)_2$</td></tr>
</table>

BINARY TO OTHER

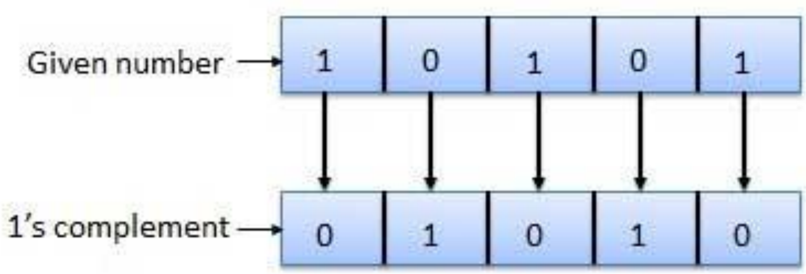
A) Multiply the digit with 2(with place value exponent). Eventually add all the multiplication becomes the Decimal number.

1. BINARY TO DECIMAL



1's complement

The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's. This is called as taking complement or 1's complement. Example of 1's Complement is as follows.



Binary Addition

It is a key for binary subtraction, multiplication, division. There are 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

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

0011010 + 001100 = 00100110

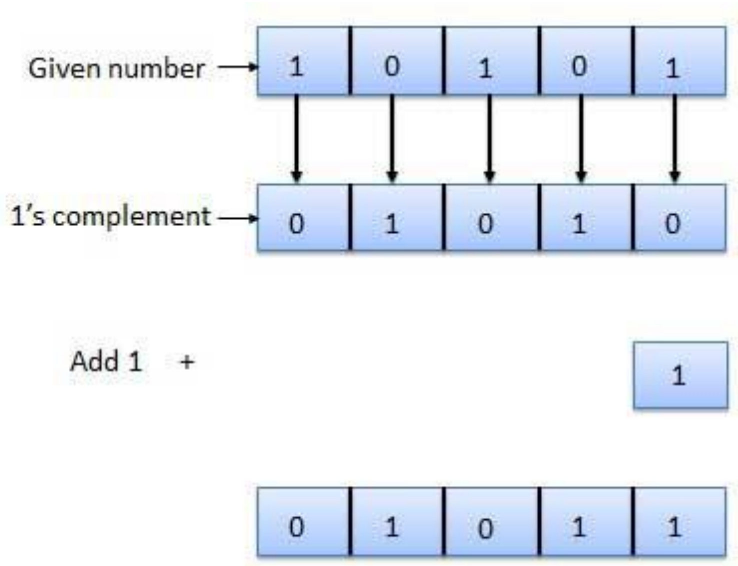
```
  1 1           carry
  0 0 1 1 0 1 0  = 26₁₀
 +0 0 0 1 1 0 0  = 12₁₀
 ---------------
  0 1 0 0 1 1 0  = 38₁₀
```

2's complement

The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.

2's complement = 1's complement + 1

Example of 2's Complement is as follows.



Rules of Binary Addition

- `0 + 0 = 0`
- `0 + 1 = 1`
- `1 + 0 = 1`
- `1 + 1 = 0`, and carry 1 to the next more significant bit

For example,

00011010

			1	*1*						*Carries*
	0	0	0	1	1	0	1	0	=	$26_{(base\ 10)}$
+	0	0	0	0	1	1	0	0	=	$12_{(base\ 10)}$
	0	0	1	0	0	1	1	0	=	$38_{(base\ 10)}$

00010011

		1	*1*	*1*	*1*	*1*				*carries*
	0	0	0	1	0	0	1	1	=	$19_{(base\ 10)}$
+	0	0	1	1	1	1	1	0	=	$62_{(base\ 10)}$
	0	1	0	1	0	0	0	1	=	$81_{(base\ 10)}$

Rules of Binary Multiplication

- `0 x 0 = 0`
- `0 x 1 = 0`
- `1 x 0 = 0`
- `1 x 1 = 1`, and no carry or borrow bits

For example,

`00101001 × 00000110 = 11110110`

```
            0   0   1   0   1   0   0   1   =   41(base 10)
          × 0   0   0   0   0   1   1   0   =    6(base 10)
   -------------------------------------
            0   0   0   0   0   0   0   0
        0   0   1   0   1   0   0   1
    0   0   1   0   1   0   0   1
   -------------------------------------
    0   0   1   1   1   1   0   1   1   0   =  246(base 10)
```

Binary Division

Binary division is the repeated process of subtraction, just as in decimal division.

For example,

`00101010 ÷ 00000110 =`

`00000111`

```
                                            1   1   1   =   7(base 10)
                 ____________________________
    1   1   0 )  0   0   ~~1~~  ¹0   1   0   1   0      =  42(base 10)
                   -     1   1   0                      =   6(base 10)
                 ----------------
                            1                              borrows
                       ~~1~~  ~~0~~  ¹0   1
                   -         1   1   0
                       ----------------
                               1   1   0
                   -           1   1   0
                           ---------------
                                       0
```

10000111 ÷ 00000101 = 00011011

```
                    1   1   0   1   1   =  27(base 10)
          ---------------------------
1  0  1 ) 1  0  0  ¹0   0   1   1   1   =  135(base 10)
        -    1  0   1                   =  5(base 10)
        -------------
             1  1  ¹0
          -  1  0   1
          -------------
                1   1
          -         0
          -------------
                1   1   1
          -     1   0   1
          -------------
                    1   0   1
                -   1   0   1
                -------------
                            0
```

Example − Division

101010 / 000110 = 000111

```
              1 1 1        = 7₁₀
         -------------
000110 ) 1¹0 1 0 1 0       = 42₁₀
        - 1 1 0            = 6₁₀
          -------
          1¹0 0 1
         - 1 1 0
          -------
             1 1 0
           - 1 1 0
           -------
                 0
```