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

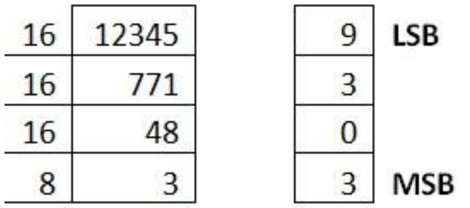
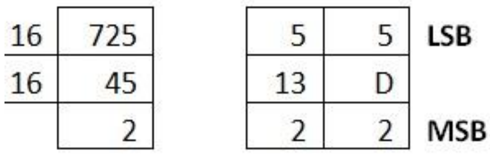
To convert Number system from **Decimal Number System** to **Any Other Base** is quite easy; you have to follow just two steps:
A) Divide the Number (Decimal Number) by the base of target base system (in which you want to convert the number: Binary (2), octal (8) and Hexadecimal (16)).
B) Write the remainder 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>

2. DECIMAL TO OCTAL

<table>
<tr><td>Decimal to Octal Conversion</td><td>Result</td></tr>
<tr><td>Decimal Number is : (12345)$_{10}$
<table>
<tr><td>8</td><td>12345</td><td>1</td><td>LSB</td></tr>
<tr><td>8</td><td>1543</td><td>7</td><td></td></tr>
<tr><td>8</td><td>192</td><td>0</td><td></td></tr>
<tr><td>8</td><td>24</td><td>0</td><td></td></tr>
<tr><td></td><td>3</td><td>3</td><td>MSB</td></tr>
</table>
</td><td>Octal Number is (30071)$_8$</td></tr>
</table>

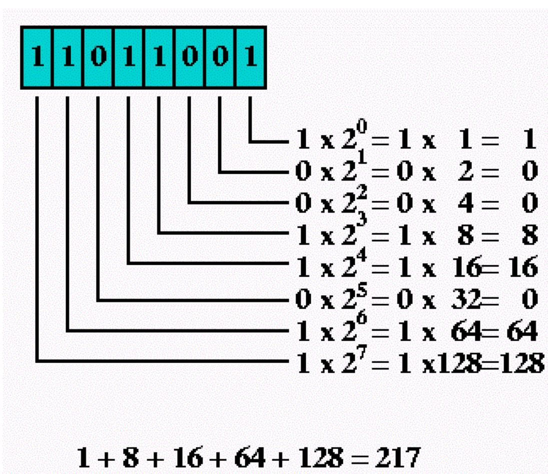
3. DECIMAL TO HEXADECIMAL

Decimal to Hexadecimal Conversion	Result
Example 1 Decimal Number is : **$(12345)_{10}$** 	Hexadecimal Number is **$(3039)_{16}$**
Example 2 Decimal Number is : **$(725)_{10}$** 	Hexadecimal Number is **$(2D5)_{16}$** **Convert** **10, 11, 12, 13, 14, 15** **to its equivalent...** **A, B, C, D, E, F**

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



2. BINARY TO OCTAL

An easy way to convert from binary to octal is to group binary digits into sets of three, starting with the least significant (rightmost) digits.

Binary: 11100101 =	11 100 101	
	011 100 101	Pad the most significant digits with zeros if necessary to complete a group of three.

Then, look up each group in a table:

Binary:	000	001	010	011	100	101	110	111

Octal:	0	1	2	3	4	5	6	7

Binary =	011	100	101	
Octal =	3	4	5	= 345 oct

3. BINARY TO HEXADECIMAL

An equally easy way to convert from binary to hexadecimal is to group binary digits into sets of four, starting with the least significant (rightmost) digits.

Binary: 11100101 = 1110   0101

Then, look up each group in a table:

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

Binary =	1110	0101	
Hexadecimal =	E	5	= E5 hex

OCTAL TO OTHER

1. OCTAL TO BINARY

Converting from octal to binary is as easy as converting from binary to octal. Simply look up each octal digit to obtain the equivalent group of three binary digits.

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

Octal =	3	4	5	
Binary =	011	100	101	= 011100101 binary

2. OCTAL TO HEXADECIMAL

When converting from octal to hexadecimal, it is often easier to first convert the octal number into binary and then from binary into hexadecimal. For example, to convert 345 octal into hex:

(from the previous example)

Octal =	3	4	5	
Binary =	011	100	101	= 011100101 binary

Drop any leading zeros or pad with leading zeros to get groups of four binary digits (bits):
Binary 011100101 = 1110   0101

Then, look up the groups in a table to convert to hexadecimal digits.

Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	0	1	2	3	4	5	6	7

Binary:	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal:	8	9	A	B	C	D	E	F

Binary =	1110	0101	
Hexadecimal =	E	5	= E5 hex

Therefore, through a two-step conversion process, octal 345 equals binary 011100101 equals hexadecimal E5.

3. OCTAL TO DECIMAL

The conversion can also be performed in the conventional mathematical way, by showing each digit place as an increasing power of 8.

345 octal = $(3 * 8^2) + (4 * 8^1) + (5 * 8^0) = (3 * 64) + (4 * 8) + (5 * 1) = 229$ decimal

OR

Converting octal to decimal can be done with repeated division.

1. Start the decimal result at 0.
2. Remove the most significant octal digit (leftmost) and add it to the result.
3. If all octal digits have been removed, you’re done. Stop.
4. Otherwise, multiply the result by 8.
5. Go to step 2.

Octal Digits	**Operation**	**Decimal Result**	**Operation**	**Decimal Result**
345	+3	3	× 8	24
45	+4	28	× 8	224
5	+5	229	done.	

⇨ $(345)_8 = (229)_{10}$

HEXADECIMAL TO OTHER

1. HEXADECIMAL TO BINARY

Converting from hexadecimal to binary is as easy as converting from binary to hexadecimal. Simply look up each hexadecimal digit to obtain the equivalent group of four binary digits.

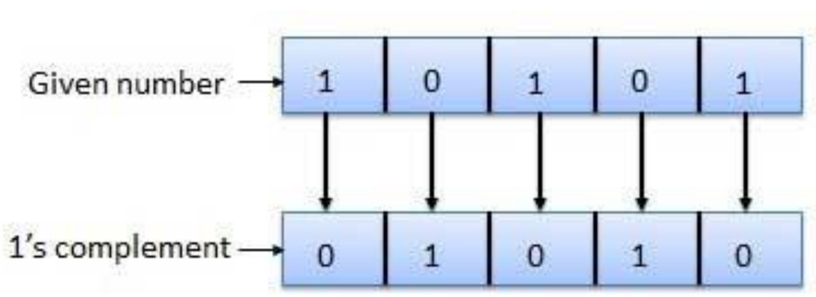
Hexadecimal:	0	1	2	3	4	5	6	7
Binary:	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal:	8	9	A	B	C	D	E	F
Binary:	1000	1001	1010	1011	1100	1101	1110	1111

Hexadecimal =	A	2	D	E	
Binary =	1010	0010	1101	1110	= 1010001011011110 binary

2. HEXADECIMAL TO OCTAL

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

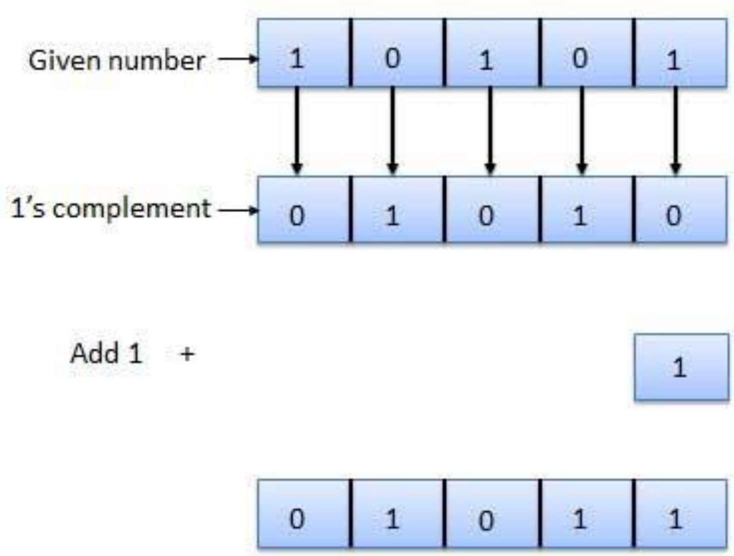
```
    1 1         carry
  0 0 1 1 0 1 0   = 26₁₀
+ 0 0 0 1 1 0 0   = 12₁₀
---------------
  0 1 0 0 1 1 0   = 38₁₀
```

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 + 00001100 = 00100110`

$$\begin{array}{ccccccccccl}
 & & & \mathit{1} & \mathit{1} & & & & & & \textit{Carries} \\
 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & = & 26_{(\text{base } 10)} \\
+ & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & = & 12_{(\text{base } 10)} \\
\hline
 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & = & 38_{(\text{base } 10)}
\end{array}$$

`00010011 + 00111110 = 01010001`

$$\begin{array}{ccccccccccl}
 & & \mathit{1} & \mathit{1} & \mathit{1} & \mathit{1} & \mathit{1} & & & & \textit{carries} \\
 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & = & 19_{(\text{base } 10)} \\
+ & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & = & 62_{(\text{base } 10)} \\
\hline
 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & = & 81_{(\text{base } 10)}
\end{array}$$

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`

$$\begin{array}{cccccccccccr}
 & & & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & = \; 41_{(\text{base } 10)} \\
 & & \times & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & = \; 6_{(\text{base } 10)} \\
\hline
 & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
 & & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & & \\
 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & & & \\
\hline
 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & = \; 246_{(\text{base } 10)}
\end{array}$$

Binary Division

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

For example,

`00101010 ÷ 00000110 =` $\quad 1 \;\; 1 \;\; 1 \; = \; 7_{(\text{base } 10)}$

00000111

1 1 0) 0 0 ~~1~~ [1]0 1 0 1 0 = 42(base 10)

\- 1 1 0 = 6(base 10)

1

~~1~~ ~~0~~ [1]0 1 *borrows*

\- 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~~ [1]0 0 1 1 1 = 135(base 10)

\- 1 0 1 = 5(base 10)

1 ~~1~~ [1]0

\- 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 = 7_{10}

000110) ~~1~~[1]0 1 0 1 0 = 42_{10}

-1 1 0 = 6_{10}

~~1~~[1]0 0 1

-1 1 0

1 1 0

-1 1 0

0