

* Number System & Bitwise Operator *

1] Introduction to Number System :-

- 1) It is just a system to represent numbers in different form with different base values.
- 2) We use decimal number system in our day to day life. Computer uses binary number system.
- 3) There are octal & hexadecimal number system as well. Hexadecimal number system used in colour codes & also in memory addresses in computer.

2] Decimal Number System :-

- 1] In decimal number system 0-9 digits are used. It has base 10.
- 2] Let's understand base 10 means like we can represent any number in the power of 10.

$$\text{e.g. } (3451)_{10} = 3 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 1 \times 10^0 \\ = 3451$$

So,

any decimal number can be represented in the powers of 10.

3] Binary Number System :-

1) It uses only 0 & 1

2) Base of binary is 2

3) e.g. →

$$(1001)_2 = (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$
$$= 8 + 0 + 0 + 1$$
$$= 9 \leftarrow \text{This is decimal number}$$

4] Conversion of Binary to Decimal :-

lets convert 1011 into decimal

$$(1011)_2 = (1 \times 2^3) * (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$$
$$= 8 + 0 + 2 + 1$$
$$= 11$$

we will see program for it later in loops Q.

5] Conversion of Decimal to Binary :-

lets take a number 11

$$\begin{array}{r} 2 | 11 \\ 2 | 5 \\ 2 | 2 \\ \hline 1 \end{array}$$

So, Binary of 11 is 1011

5] Binary addition & Binary Subtraction

a] Binary addition :-

$$\begin{array}{r} 0101 \rightarrow 5 \\ + 1001 \rightarrow 9 \\ \hline 1110 \rightarrow 14 \end{array}$$

Here, we can simply calculate decimal of any number, from 0101 , 1001 , 1110

\downarrow								
8	4	2	1	8	4	2	1	8
\downarrow								
0	7	4	0	8	9	7	1	8
$\underline{+4+0+1}$	$\underline{+4+2+1}$	$\underline{+4+2+1}$	$\underline{+4+2+1}$	$\underline{+0+0+1}$	$\underline{+0+0+1}$	$\underline{+0+0+1}$	$\underline{+0+0+1}$	$\underline{+4+2+0}$
		5			9			14

Here, $1+1=2$ & in binary of 2 is 10 so we take 0 first & took 1 **Carry**
also we can add as many as 0 in front of number

$$\begin{array}{r} 01110 \rightarrow 14 \\ + 01011 \rightarrow 11 \\ \hline 11001 \rightarrow 25 \end{array}$$

$$\begin{array}{r} 1111 \rightarrow 15 \\ + 1110 \rightarrow 14 \\ \hline 11110 \rightarrow 29 \end{array}$$

b] Binary Subtraction :-

- In binary subtraction we have to perform ~~subtraction~~ of given two numbers, like,
- $9 - 4 = 5$ same way $9 + (-4) = 5$ as well
- So we have to take negation of the second number
- $9 \rightarrow 1001 \rightarrow$
 $+(-4) \rightarrow +100$
 $\underline{\underline{-5}}$
- To make second number negative we have to use, 2's complement.
- 2's complement is, steps are,
 - swap the bits
 - add 1 in swapped bits
- We can add as many as zeros here as well

So, negative of 4 can be done like this
 binary of 4 is 100
 negation/swapping 0000011] step 1 (Here we complete it as 0 as it will be negative)
 adding 1 in it +1111001] step 2 (like this)
 1111100 (It will still remain negative)

Now, adding this 2's complement in 9

$$\begin{array}{r} 9 \rightarrow 100 \\ +(-4) \rightarrow +\underline{100} \\ \hline 5 \quad 0101 \rightarrow 5 \end{array}$$

In bigger picture,

0000000001001

11111111100

00000000101

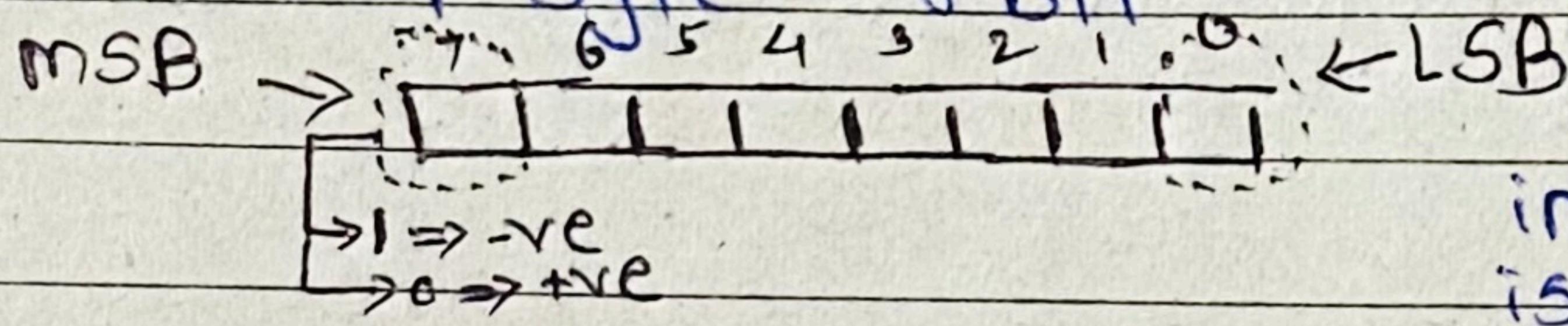
Q] Why 128 in byte is -128? ANSWER B

• int a = 128;

byte b = (byte) a; $\Rightarrow -128$

we explicitly convert int into byte but why it gave -128

• 1 byte = 8 bit



MSB = most significant bit
 LSB = least significant bit
 in MSB, if 1 is present it shows number is negative & if 0 is present it shows number is positive

• For 128, binary is,

100000000 i.e. 1|0|0|0|0|0|0|0

here MSB is 1 i.e. negative in the byte. but in int we have 4 byte or 32 bit memory in that case MSB is 0 & 128 becomes 0

• if we try to add 1 in 127 then this will happen

$$01111111 \rightarrow 127$$

$$+ 00000001 \rightarrow 1$$

$$10000000 \rightarrow 128 \text{ MSB here is 1 so it becomes negative}$$

128
represent
in
int

MSB
is
0

so
number
is
+ve