Number System:- It is method of representing numbers on the numberline with the help of set of Rules and symbol.

Types of Number System

1. Binary
2. Decimal
3. Octal
4. Hexadecimal

Decimal Number System

- Base value 10
- Can go from 0–9 (10 digits)
- eg:- 10285

$(1\times10^4)+(0\times10^3)+(2\times10^2)+(8\times10^1)$
$+(5\times10^0)$

Binary Number System

- Base value 2
- It uses 2 digits (0 and 1)
- eg:- 0, 1, 10, 11, 100, 101, 110, 111, 1000, and 1001.

(0 to 9 in Binary)

Octal Number System

↳ Base value is 8.
↳ It uses 8 digits from 0-7.
↳ eg:- $(135)_{10}$ is $(207)_8$
$(215)_{10}$ is $(327)_8$

Hexa decimal Number System

↳ Base value is 16.
↳ Digits from 0-9 are taken and for 10-15 is represented as A-F.
↳ Eg:- $(255)_{10}$ can be written as $(FF)_{16}$ and $(1096)_{10}$ as $(448)_{16}$.

Conversion of Number System

Binary to Decimal

Base-2 to Base-10

$(1\ 0\ 1\ 0\ 1\ 0)_2 \longrightarrow (?)_{10}$

$2^5 + 2^4 + 2^3 + 2^1 + 2^0$

$32 + 0 + 8 + 2 + 0 = 42$

(Method - I)

Power of 2's

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	...
1	2	4	8	16	32	64	128	256	512	1024	...

Method - II

$(1\ 0\ 1\ 0\ 1\ 0)_2 \longrightarrow (?)_{10}$

$\boxed{0} \times \boxed{2} + 1$ → Initial value (0), Base value (2)

$= 1 \times 2 + 0$

$= 2 \times 2 + 1$

$= 5 \times 2 + \boxed{0}$ → Number (Binary)

$= 10 \times 2 + 1$

$= 21 \times 2 + 0$

$= 42$

$(1\ 2\ 3)_{10} \longrightarrow (?)_{10}$

$0 \times 10 + \boxed{1}$ → Number (Decimal)

$= 1 \times 10 + 2$

$= 12 \times 10 + 3$

$= \underline{\underline{123}}$

$(1\ 2\ 3)_{10} \xrightarrow{\boxed{\text{Reverse}}} (3\ 2\ 1)_{10}$

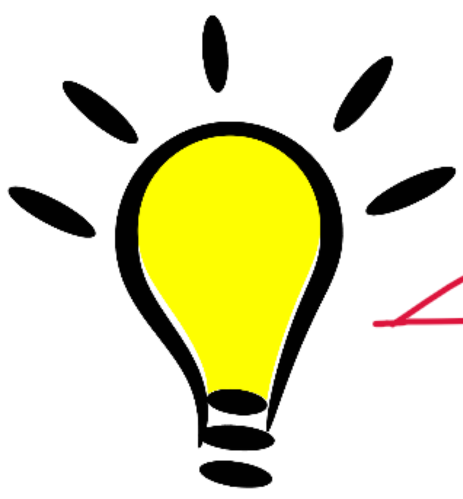
$0 \times 10 + 3$

$= 3 \times 10 + 2$

$= 32 \times 10 + 1$

$= \underline{\underline{321}}$

We can reverse numbers like this



Converting Base-10 to 2

$(42)_{10} \longrightarrow (?)_2$

$\longrightarrow (101010)_2$

Method - I

2	42	0
2	21	1
2	10	0
2	5	1
2	2	0
2	1	1
	0	

Method - II

$(42)_{10} \rightarrow$ Can be split into sum of 2^n

$42 = 32 + 8 + 2$

$\downarrow \quad \downarrow \quad \downarrow$

$2^5 \quad 2^3 \quad 2^1$

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

Decimal Range for n-bit binary number

How many n-bit binary numbers are possible? $\rightarrow 2^n$

$2^2 = 4$, $2^3 = 8$ numbers possible.

1st	0th
0	0
0	1
1	0
1	1

for $2^2 = 4$ numbers possible

for 3 bit binary, 8 numbers are possible.

2nd	1st	0th
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Smallest n-bit binary number

$\hookrightarrow \boxed{(0)_{10}}$

pos.	(n-1)th	(n-2)th	...	ith	...	2nd	1st	0th
bit	0	0		0		0	0	0
wgt.	2^{n-1}	2^{n-2}	...	2^i	...	2^2	2^1	2^0

$\Downarrow$

(largest n-bit binary number)

largest n-bit binary number :-

$$\underbrace{2^0 + 2^1 + 2^2 + \cdots + 2^{n-1}}_{\text{This forms a gp.}}$$

Here, $a = 1$, $r = 2$, $\# = n$

$$\frac{a(r^{\#} - 1)}{r - 1} = \frac{1(2^n - 1)}{2 - 1}$$

$$= \boxed{(2^n - 1)_{10}}$$

largest n bit binary

We know the limits of n-bit binary number which is.

$$2^n \left\{ \begin{array}{l} 0\,0\,0\,0 \cdots 0 \rightarrow (0)_{10} \\ \vdots \\ 1\,1\,1\,1 \cdots 1 \rightarrow (2^n-1) \end{array} \right.$$

$$\xleftarrow{\quad} n \xrightarrow{\quad}$$

least numbers of bits to represent a decimal number.

To deduce the least no of bits used, we know that

$0 \le x \le 2^n - 1$

$x \le 2^n - 1$

$2^n \ge x + 1$

taking $\log_2$ on both sides

we get,

$\log_2 2^n \ge \log_2(x+1)$

$$\boxed{n \ge \log_2(x+1)}$$

formula for least bits to represent decimal number.

for eg:- $x = 12$

$n \geq \log(12+1)$

$n \geq \log_2(13)$

$n \geq \underbrace{3.....}$

↓

ceil = 4 (least bits)

Complementary Number System

↳ Used to perform subtraction using addition.

$x + y$ → can be done

$x - y$ → cannot be done

↓

complement of $y = -y$

$\therefore x + (-y)$ → possible

for Base-2 they are two types of complement,

1's complement

↳ Two method to convert,

* Adding 1 to Binary
* Finding seek bit.

2's Complement

Method

Seek bit

$(12)_{10} \rightarrow (01\boxed{1}00)_2$

[1st flipped bit in a binary number]

1^s complement of $(12)_{10}$
$\hookrightarrow (10011)_2$

2^s complement of $(12)_{10}$
$\hookrightarrow$ Flip all the bits after the seek bit to get 2^s com.
$\hookrightarrow$ (10100) or (−12)
$\hookrightarrow$ [2^s complement will give −ve representation of number]

Substraction In Binary

$12 - 9 = 12 + (-9)$

$(12)_{10} = (01100)_2$

$(9)_{10} = (01001)_2 \xrightarrow{2^s} (10111)$

$$\begin{array}{r} (01100) \\ +\ (10111) \\ \hline (00011) \end{array}$$

$\Downarrow$

3

Binary Representation for Signed Integer

In signed binary representation, the left most bit is the sign-bit which indicate [(1) if number < 0] [(0) or ≥ 0.]

Eg:- $(\underline{1}0100)_2 \rightarrow (-ve)_{10}$

$(\underline{0}110)_2 \rightarrow (\geq 0)_{10}$

Converting signed integer from [Base-2 to Base-10]

$(1\,0100)_2 \rightarrow -16+0+4+0+0$
$= \underline{\underline{-12}}$

$(0110)_2 \rightarrow 4+2 = \underline{\underline{6}}$

$(-12) \rightarrow (\underline{1}10100)$
$\hookrightarrow -32+16+4 = -12$

[Also be represented as]

(-12)

$$[\underbrace{\boxed{11\ldots\ldots1}}^{27}\,10100]$$

32 (Size)

Decimal range for n-bit signed binary number

→ How many n-bit signed binary no. are possible.

$2^n \rightarrow$
- $\geq 0 \rightarrow 2^{n-1}$ ✓ [0 . . . .]
- $< 0 \rightarrow 2^{n-1}$ ✓ [1 . . . .]

Important

Smallest n-bit signed binary number given that the sign bit is zero?

pos.	(n-1)th	(n-2)th	.	...ith		.2nd	1st	0th
bit	0	0	.	0		0	0	0
wgt.	-2^{n-1}	2^{n-2}	.	...2^i		..2^2	2^1	2^0

$$\boxed{(0)_{10}}$$

if everything is zero.

↳ for largest no. with sign bit zero

for finding the largest no, we know,

$$\underbrace{2^0 + 2^1 + \dots + 2^{n-2}}_{GP}$$

first bit already filled

where, $a = 2^0 = 1$

$r = 2$

$\# = n-1$

$$\Rightarrow \frac{1(2^{n-1}-1)}{2-1} = \boxed{2^{n-1}}$$

largest no. with sign bit zero.

By this, we get the range for, n-bit signed binary number.

Range:-

$$2^{n-1}\begin{cases} \boxed{0}\ 000\ldots\ldots 0 \Rightarrow \boxed{0} \\ \vdots \\ \boxed{0}\ 111\ldots\ldots 1 \Rightarrow \boxed{2^{n-1}-1} \end{cases}$$

$[0, 2^{n-1}-1]$

Now similarly for n-bit signed binary number with sign bit as one:-

Smallest :- $\boxed{(-2^{n-1})_{10}}$

pos.	(n-1)th	(n-2)th	. . .	ith	. . .	2nd	1st	0th
bit	1	0	.	0		0	0	0
wgt.	-2^{n-1}	2^{n-2}	. . .	2^i	. . .	2^2	2^1	2^0

largest :-

pos.	(n-1)th	(n-2)th	. . .	ith	. . .	2nd	1st	0th
bit	1	1	.	1		1	1	1
wgt.	-2^{n-1}	2^{n-2}	. . .	2^i	. . .	2^2	2^1	2^0

$$= -\cancel{2^{n-1}} + (\cancel{2^{n-1}} - 1)$$

$$= \boxed{-1}$$

Range:- (Sign bit 1)

$$2^{n-1}\begin{cases} \boxed{1}\ 000\ldots 0 \Rightarrow -2^{n-1} \\ \vdots \\ \boxed{1}\ 000\ldots 1 \Rightarrow -1 \end{cases}$$

$[-2^{n-1}, -1]$

∴ Complete Range for Signed n-bit binary number, will be :-

2^n

- s.b = 1 → 2^{n-1}
  - $\boxed{1}\,000\ldots0 \Rightarrow -2^{n-1}$ ($-\infty$)
  - $\boxed{1}\,111\ldots1 \Rightarrow -1$
- s.b = 0 → 2^{n-1}
  - $\boxed{0}\,000\ldots0 \Rightarrow 0$
  - $\boxed{0}\,111\ldots1 \Rightarrow 2^{n-1}-1$ ($+\infty$)

Important

$$\boxed{\left[-2^{n-1},\ 2^{n-1}-1\right]}$$

for $n = 32$,

$$[\underbrace{-2^{31}}_{\text{MIN}},\ \underbrace{2^{31}-1}_{\text{MAX}}]$$

Conversion Hexadecimal to Binary

Base-16 to Base-2

eg:- $(FA7B92)_{16}$

1111 1010 0111 1011 1001 0010

$$\rightarrow (\underbrace{1111}_{F}\ \underbrace{1010}_{A}\ \underbrace{0111}_{7}\ \underbrace{1011}_{B}\ \underbrace{1001}_{9}\ \underbrace{0010}_{2})$$

from Base-2 to Base-16

eg:- $(111011)_2 \rightarrow (?)_{16}$

We can write,

$$(\underbrace{0011}_{3}\ \underbrace{1011}_{B})_2 \rightarrow \underline{\underline{(3B)_{16}}}$$