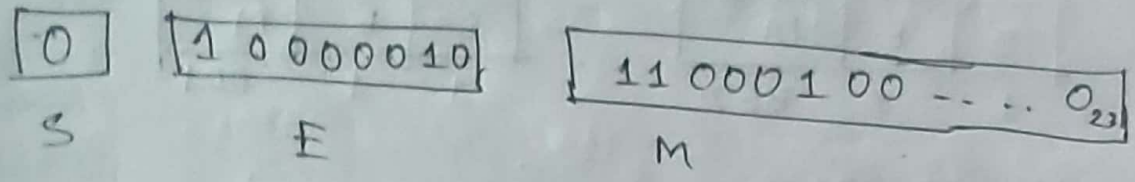
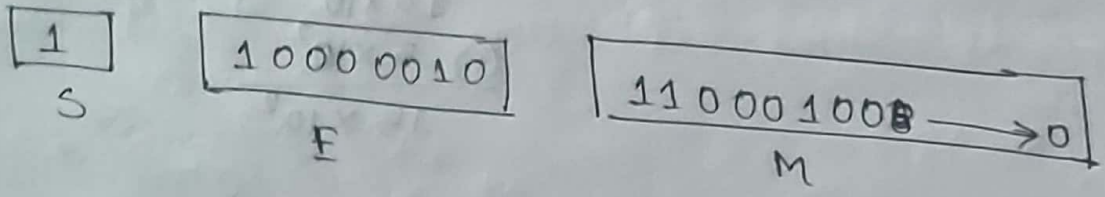
$(130)_{10} \rightarrow (10000010)_2$.



-14.125 হলে



Q: 14.125 → single precision means 32 bit.

Step 1: convert binary.

2: Normalization.

3: take the bias.

4: convert bias exponent to binary.

5: substitute into required formate.

$(14.125)_{10} \rightarrow (1110.001)$.

$1110.001 \rightarrow 1.110001 \times 2^3$ → positive value

$= (-1)^0 \times 1.110001 \times 2^3$ → True exponent

$(-1)^S \times 1.M \times 2^E$

single precision → 127. (bias exponent).

$2^{127+3} = 2^{130}$

~~$(-1)^0 \times 1.110001 \times$~~

→ Power ≥ (-) থাকবে problem এর solution.

Exponent ও (-) :

bias exponent use করব,

Biased exponent = True exponent + Bias.

BE = TE + Bias.

(127) → 8bit এর জন্য

(*) Support equal number of positive and negetive.

$2^8 = 256 = 255$ (যে 0 আছে) $= \frac{255}{2} = 127.$

ভগ্নাংশ avoid করব,

127 নিব। -128 নিলাম, 127 নেয়া যাবে না।

তাই (-127 and 127).

-128 to 128 নিলে space exceed করবে,

$2^{11} = 2048$ → 0 সহ 2047 $= \frac{2047}{2}$

$= 1023.5$

1023 ←

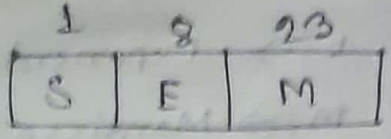
ভগ্নাংশ omit করে।

Scanned with CamScanner

S.E.M (digit of the number without 1 and (.))

$(-1)^S \times 1.M \times 2^E$. → Required normalization form.

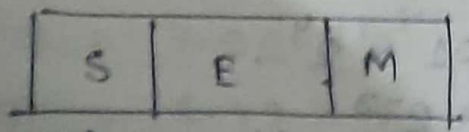
IEEE - 754 - 32 bit format.



IEEE - 754 - 64 bit format.

→ biased exponent.

$[2^{BE} = 2^{TE + \text{bias value}}]$



(bit)

101.101

$2^2 2^0 . 2^{-1} 2^{-2} 2^{-3}$

Lecture - 2

Floating point numbers

→ point store করা

→ ignore করা।

$72.0 = 7.20 \times 10^1$

$6.49 = 6.49 \times 10^0$

$.247 = 2.47 \times 10^{-1}$

$5.63 = 5.63 \times 10^0$.

} Normalized form.

Left side এ only one non zero digit থাকে। like 1.23, 2.46.

(position of the point) ↑ → (digit representation) করা।

E	M
0	563
2	649

E → Exponent (position of the point)

M → Mantissa. (digit in the number)

$0101.001 \longrightarrow 01.01001 \times 2^2$

$11111.101 \longrightarrow 1.111101 \times 2^4$.

$0.001.01 \longrightarrow 1.01 \times 2^{-3}$

$-10.01 \longrightarrow -1.001 \times 2^1$.

$(-1)^S$

E	M
1	1001

S = sign.

neg → sign = 1.

pos → sign = 0

↑001

1 omitted বা আগে থেকে same.

9 → +9 (0 1 00 1)

→ -9 1 0 1 1 0

+ 1

1 0 1 1 1 .

Fixed point numbers:

56.7
78.2
42.5
} point is position fixed.

Floating point numbers:

0.0158
1.275
23.5
} point is position fixed is,

Any number where the position of the of number is not fixed is called floating point number.

Convert the floating to fixed point (cause floating point number is processing time consuming in computer).

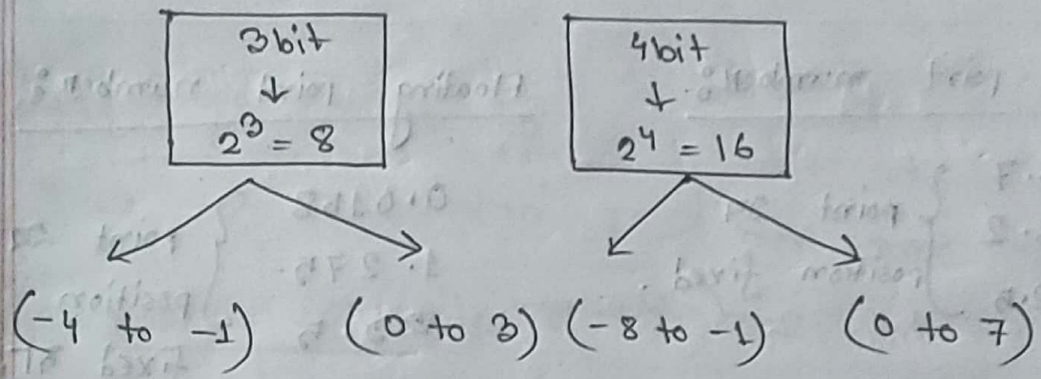
$73.2 = 7.32 \times 10^{1}$

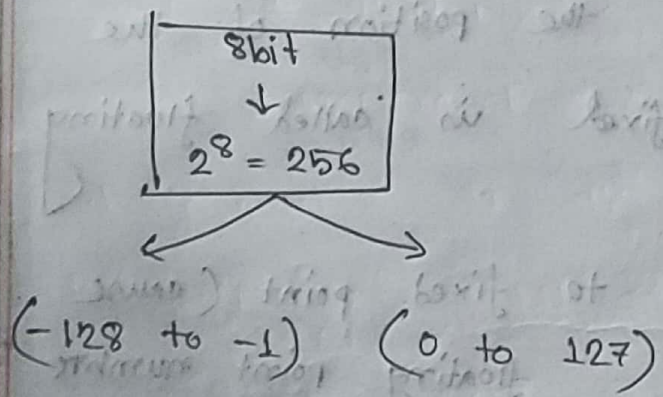
$649 = 6.49 \times 10^{2}$

$.847 = 8.47 \times 10^{-1}$

$5.63 = 5.63 \times 10^{0}$.

16	8	4	2	1
↑	↑	↑	↑	↑
2^4	2^3	2^2	2^1	2^0
(1	0	1	0)$_2$	→ (10)$_{10}$





7 → +7 (0 1 1 1)

→ -7 1 0 0 0
+ 1
1 0 0 1

101 → 2's complement.
because starts with 1.
2's complement করে
ans বের করতে হবে।

101 → 0 1 0
+ 1
0 1 1

011 means (+3).
So, main number (-3)

Unique combination of zero so problem solved in 2's complement method.

2's complement:

1's complement + 1.

positive number same থাকবে (0 to +3).

(-1 to -3) convert করব। 1's complement + 1

8 টা combination এ কোনটা নাই সেটা খুঁজে বের করব।

100 missing. 100 এর 1's complement = 011.

2's complement $\begin{array}{r} 011 \\ +1 \\ \hline 100 \end{array}$ Represents (-4)

So, (-0) problem solved so this method can be used.

Short cut:

From right side, copy the number as it is till you get first 1. After getting first 1, convert the remaining digit.

কারণ, 100 missing. 100 means (-0) as
1 represents neg sign and 00 represents 0.
(-)0 not possible because zero always positive.
Main failure of this method is (-0).
So, this method can not be used.

— x —

1's complement:

(1) আমি কি করব, (+3 থেকে 0 পর্যন্ত).
(+3 এর complement হবে -3).
2 টা combination এর মধ্যে কোনটা missing → খুঁজতে।
111 missing. 111 means এর 1's complement
means অর্থাৎ convert করা, convert করলে 000.
0 means sign and 00 means 0 এর corresponding
binary convert করার পর (-0) আর আসছে
convert করার আগে (-0) ছিল, Again (-0) problem
So 1's complement can not be used.

	1's complement	Sign magnitude	2's complement
+3	011	0 11	011
+2	010	0 10	010
+1	001	0 01	001
0	000	0 00	000
-1	110	1 01	111
-2	101	1 10	110
-3	100	1 11	101

Sign magnitude:

1st bit ↳ corresponding value.

sign represent করে (+ or −)

(+) এর জন্য 0 (−) এর জন্য 1.

zero কে positive ধরা হয়।

৪টা combination এর মধ্যে কোনটা missing check

Method : 2

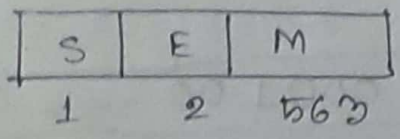
2^3 2^2 2^1 2^0.

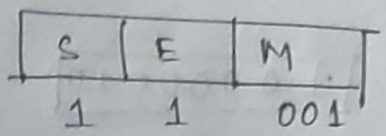
Method : 1

0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

Lecture – 4

$(-1)^S \times 1.M \times 2^E$





-1.001×2^1.

$= -10.01$.

$$\begin{array}{r} 10 \times 10^2 \\ (+)\ 5 \times 10^2 \\ \hline 15 \times 10^2 \end{array}$$

✱ bias কেন করি?

To make the calculation faster. exponent pos and neg হলে addition ও check and comparison time consuming and multiplication and division এ থাকবে time consuming

so bias imp.

(*) Single precision exponent এর জন্য 8 bit

মানে, $2^8 = 256 \rightarrow 255$ (যে 0 কে positive ধরব).

$\frac{255}{2} = 127.5$ (fraction ommit করব).

so, 127 so that it can support equal number of positive and negative.

(*) 0.00101.

$= 1.01 \times 10^{-3}$.

$(-1)^S \; 1.M \times 2^E$.

$M = 01$, $E = -3$.

D.P = 1023.

$2^{-3+1023} = 2^{1020}$

S	E	M
0	11111111100	01000000

$(-1)^0 (1.01)^{11110111100}$.

52

Q: 0101.001 → single precision.

$(-1)^0 \times 1.01001 \times 2^2$.

$(-1)^0 \quad 1.01001 \times 2^{129}$.

$(-1)^0 \quad 1.01001 \times 2^{10000001}$

Exponent 10 হলে বাম পাশে 6 টা zero নিব,

cz exponent bias value নিব।

bias value always positive

exponent কখনোই neg নিব না so exponent কখনোই 2's complement হবে না, 2's complement (1 neg number এ আমলে 1 আজ তাই pos, neg check করতে হয়),

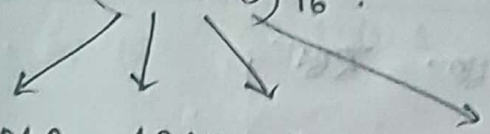
exponent always positive তাই 2's complement check → no need.

আমলে zero বসাব,

Book:

William Stalling → (Writer).

Q: $(2A3B)_{16}$.



0010 1010 0011 1011.

$(-1)^0 \times 1.01010\ 0011\ 1011 \times 2^{13}$.

$(-1)^0 \times 1.0101$.

0	10001100
S	

* Exponent এর জন্য single precision এ 8 bit and double precision এ 11 bit বরাদ্দ, অবগুলো bit represent করতে হবে, exponent মানে power. 2^{10} থাকলে $2^{00000010}$ হবে, আগেরটা 0 বসলে value র change হয় না.

Mantissa means point এর পরের digit.

Mantissa এর জন্য single precision এ 23 এবং double precision এ 52 bit assign.

এর bit represent করতে হবে।

1.01 হলে 1.0100 ⟶ 0 অর্থাৎ 0 বসানোর আগের change হয় না, so অর্থাৎ 0 বসবে।