CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Floating Point Representation

Lecture No.- 01

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Control Unit

Topic Hardwired Control Unit

Topic Micro-Programmed Control Unit
→ Horizontal
→ vertical

Topic RISC vs CISC

Topics to be Covered

- **Topic** Floating-Point Numbers
- **Topic** Biased Exponent
- **Topic** Number Range
- **Topic** IEEE-754 Floating Point Representation
- **Topic** Denormalized Number

Need or requirement of Floating point representatⁿ

Fixed point representatⁿ :-

Given 8-bits storage

- → Range of number (unsigned) ⇒ 0 to $2^8 - 1$ | 0 to 255
- → —"— (signed) ⇒ -128 to $+127$
  2's comp.

Floating point representation provides a larger range of numbers as compared to fixed point representation.

Topic : Floating-Point Numbers

- The number is represented in format:

Sign	Exponent	Mantissa
S	E	M
1 bit		

- Mantissa is signed normalized (implicit/explicit) fraction number
- Exponent is stored in biased form.

$$S \begin{cases} 0 \Rightarrow +ve \\ 1 \Rightarrow -ve \end{cases}$$

Topic : Biased Exponent

S	E	M

→ arithmetic operations on floating point numbers become easy.

ex:- $(1.11 * 2^{2}) + (1.01 * 2^{-3})$ Vs $(1.11 * 2^{4}) + (1.01 * 2^{9})$

original exponent (e) + bias ⇒ stored Exponent (E)

Assume, E is represented ⇒ 4-bits

originally number range with 4-bits ⇒ -8 to $+7$ → Transform into unsigned number range.

⇓

0 to 15

original exponent (e)	stored exponent (E)
−8	0
−7	1
−6	2
−5	3
−4	4
−3	5
−2	6
−1	7
0	8
1	9
2	10
3	11
⋮	⋮
7	15

excess-8 code

$E = e + 8$

Here in this example 8 is bias.

if k-bits are used to represent E, then

$$\boxed{\text{bias} = 2^{k-1}}$$

Topic : Mantissa

101.11

Explicit normalization $\Rightarrow 0.\underline{1}0111 * 2^{3}$

should be one

$e = 3$

$E = 3 + \text{bias}$

$M = \text{Number after point}$

$= 10111$

Implicit Normalization $\Rightarrow \underline{1}.0111 * 2^{2}$

should be one

$e = 2$

$E = 2 + \text{bias}$

$M = 0111$

ex:-

1.0

Explicit normalization $\Rightarrow 0.10 * 2^1$

$e = 1$

$E = 1 + bias$

$M = 10$

Implicit Normalization $\Rightarrow 1.0 * 2^0$

$e = 0$

$E = 0 + bias$

$M = 0$

Topic : Value Formula

S	E	M

$$\text{Value}_{(\text{explicit})} = (-1)^{S} * 0.M * 2^{E-\text{bias}}$$

$$\text{Value}\ (\text{implicit}) = (-1)^{S} * 1.M * 2^{E-\text{bias}}$$

[MCQ]

S	$\overset{E}{e_6e_5e_4e_3e_2e_1e_0}$	M

7 bits

#Q. A certain well-known computer family represents the exponents of its floating-point numbers as "excess-64" integers; i.e., a typical exponent $e_6e_5e_4e_3e_2e_1e_0$ represents the number: → bias = 64

A) $e = -64 + \sum_{i=0}^{6} \cancel{2ie_i}\ 2^i e_i$

B) $e = -64 + \sum_{i=0}^{6} \cancel{2e_i}\ 2e_i$

C) $e = 64 - \sum_{i=0}^{6} \cancel{2ie_i}\ 2^i e_i$

D) $e = 64 - \sum_{i=0}^{6} \cancel{2e_i}\ 2e_i$

$e_6\ e_5\ e_4\ e_3\ e_2\ e_1\ e_0$

$\Downarrow$

decimal

$\Downarrow$

$$(e_6 * 2^6) + (e_5 * 2^5) + \cdots\cdots + (e_0 * 2^0)$$

$$= \sum_{i=0}^{6} e_i * 2^i$$

[NAT]

#Q. Consider a explicitly 16-bit register used to store floating point numbers. The mantissa is normalized signed fraction number. Exponent is represented in excess-32 form. What is the 16-bit value for $+(11.5)_{10}$ in this register?

$\longleftarrow 16 \longrightarrow$

S	E	M
1	6	9
0	100100	101110000

bias = 32 $\qquad 2^{k-1} \Rightarrow k = 6$

$+(11.5)_{10} \Rightarrow$ sign is positive $\Rightarrow S = 0$

$(11.5)_{10} = (1011.1)_2$

$(1011.1)_2 \Rightarrow$ explicit normalizat^n

$\Downarrow$

$0.10111 * 2^4$

$m = 10111$

$e = 4$

$E = 4 + 32 = (36)_{10} = (100100)_2$

[NAT]

#Q. What is the 4-digit hexadecimal value for $+(11.5)_{10}$ in above question's register?

0 100100 101110000

4 9 7 0

Ans $\Rightarrow (4970)_{16} = 0x4970 = 4970H$

[NAT]

Ans $= (4A5C)_{16}$

#Q. What is the 4-digit hexadecimal value for $+(37.75)_{10}$ in above question's register? (use implicit normalization)

S (1)	E (6)	M (9)
0	100101	001011100

4 A 5 C

$$(37.75)_{10} = (100101.11)_2$$

$\Downarrow$

implicit normalization

$\Downarrow$

$$1.00101111 * 2^5$$

$$e = 5,\ E = 5 + 32 = 37 = (100101)_2$$

$$M = 001011100$$

#Q. what is 4-digit hexa-decimal representation of $+(0.000101)_2$ in prev. question's register with explicit normalization?

Sol^n

$0.000101 \Rightarrow$ explicit normalization

$\Downarrow$

$0.101 * 2^{-3}$

$e = -3, \quad E = -3 + 32 = (29)_{10} = (011101)_2$

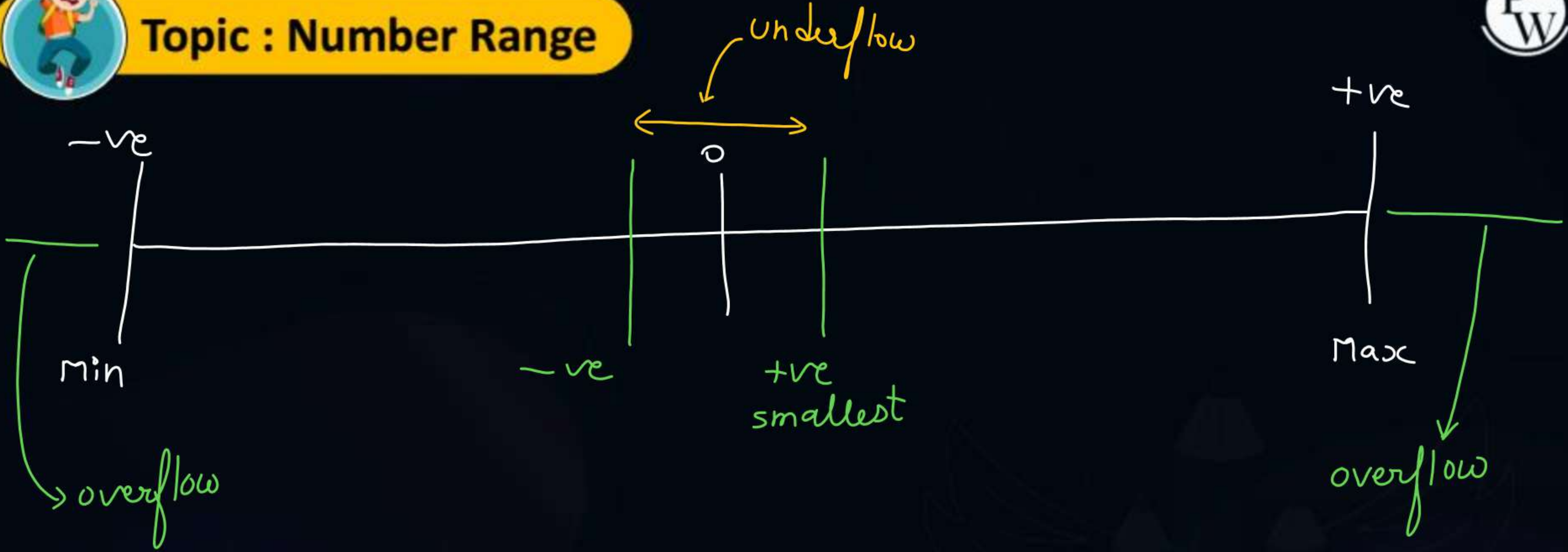
$M = 101$

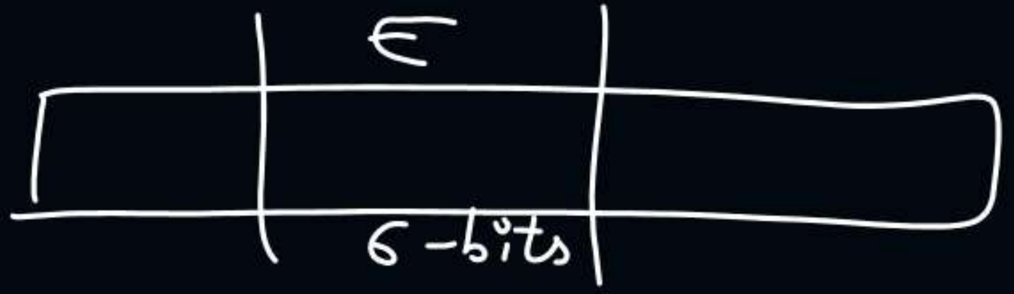
0	011101	101000000

3 B 4 0

Ans = $(3B40)_{16}$

Topic : Number Range



assume 

bias = 32

number after normalization $\Rightarrow 1.001 * 2^{32}$

$e = 32$

$E = 32 + 32 = 64$

Can not be stored in 6-bit E.

$\Downarrow$

overflow

	min	max
here $E \Rightarrow$	0	63
$e \Rightarrow$	-32	31

number after normalizat^n $= 1.0101 * 2^{-33}$

$e = -33$

$E = -33 + 32 = -1$

underflow

Topic : Bits in E and M

More bits are used for $E \Rightarrow$ Range of numbers will be larger

— " ——————————— $M \Rightarrow$ More precision or accuracy

Topic : Disadvantages of Conventional Representation

1. It can not store zero.
2. It suffers from underflow.

$\Rightarrow$ sol^n $\Rightarrow$ IEEE-754 floating point representat^n

Topic : IEEE-754 Floating Point Representation

- IEEE-754 Representation
  - Single Precision
  - Double Precision

32-bits

S	E	M
1	8	23

bias = 127

64-bits

S	E	M
1	11	52

bias = 1023

$$\left.\begin{array}{l} \text{if} \quad E = 00\ldots\ldots0 \\ \qquad\qquad \text{or} \\ \quad\;\; E = 111\ldots\ldots1 \end{array}\right\} \text{special case}$$

$$\left.\begin{array}{l} \text{if} \quad E \neq 00\ldots\ldots0 \\ \qquad\quad \text{and} \\ \quad\;\; E \neq 11\ldots\ldots1 \end{array}\right\} \begin{array}{l} \text{normal number} \\ \text{(always implicitly normalized)} \end{array}$$

[NAT]

#Q. The value of a float type variable is represented using the single- precision 32-bit floating point format IEEE-754 standard that uses 1bit for sign, 8 bits for biased exponent and 23 bits for mantissa. A float type variable X is assigned the decimal value of −27.625. The representation of X in hexadecimal notation is?

2 mins Summary

Topic Biased Exponent

Topic Normalized Mantissa

Topic Explicit vs Implicit Normalization

Topic IEEE-754 Floating Point Representation

Happy Learning

THANK - YOU