COMPUTER SCIENCE

Computer Organization and Architecture

Floating Point Representation

Lecture_01

Vijay Agarwal sir

TOPICS TO BE COVERED

01 Floating Point Representation

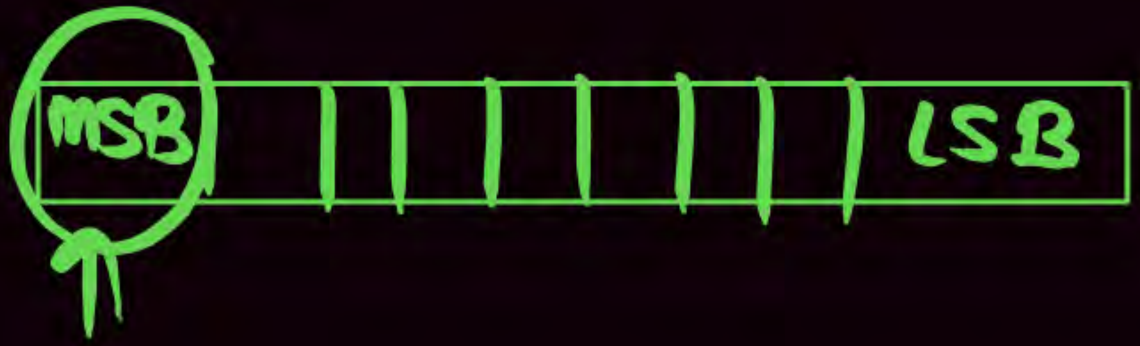
1. Introduction of COA.
2. Machine Instruction & Addressing Mode.
3. Floating Point Representation.

- ✓ Number System
- ✓ Floating Point Representation

Number System:

Sign
- 0 [+ve]
- 1 [-ve].

MSB



1. Magnitude Format
   [Signed & Unsigned]
   (+ve & -ve) (Only +ve)
2. Complement Format
   [1's Complement, 2's Complement]

I Magnitude Format

- unsigned: $(0 \text{ to } 2^{n}-1)$
- Signed: $-(2^{n-1}-1) \text{ to } +(2^{n-1}-1)$

II Complement Format

- 1's Complement: $(-2^{n-1}-1) \text{ to } +(2^{n-1}-1)$
- 2's Complement: $(-2^{n-1} \text{ to } +2^{n-1}-1)$

n bit number. I Magnitude Format

eg 4 bit

1

① Unsigned Range = 0 to $2^n - 1$

0000 → 0

eg 4 bit Unsigned Range = 0 to $2^4 - 1 \Rightarrow 0$ to 15

1111 → 15

2

② Signed Range = $-(2^{n-1} - 1)$ to $+(2^{n-1} - 1)$

4 bit Signed Range = $-2^{4-1} - 1$ to $+2^{4-1} - 1 \Rightarrow -7$ to $+7$

n bit number. (II) Complement format.

eg 4 bit

(1) 1's Complement Range = $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$

eg 4 bit 1's Complement Range = $-(2^{4-1}-1)$ to $+(2^{4-1}-1) \Rightarrow -7$ to $+7$

(2) 2 Complement Range = $-(2^{n-1})$ to $+(2^{n-1}-1)$

eg 4 bit 2's Complement = $-(2^{4-1})$ to $+(2^{4-1}-1) \Rightarrow -8$ to $+7$

DATA FORMAT

- **Fixed Point Data**
  - **Magnitude Format**
    - **Unsigned Format** [Only the + Number]
      - n bit range 0 to $2^n - 1$
      - 4bit range 0 to $2^4 - 1 \Rightarrow$ 0 to 15
    - **Signed Magnitude Format** [+ve and –ve]

MSB				LSB

      - MSB: Sign value
      - $-(2^{n-1} - 1)$ to $+(2^{n-1} - 1)$
      - 4bit range
      - Range[–7 to +7]
  - **Complement Format**
    - **1's complement Format** Both [+ve and –ve]
      - $-(2^{n-1} - 1)$ to $+(2^{n-1} - 1)$
      - 4bit range
      - [–7 to +7]
      - 2 Representations for 0
      - +0: 0000
      - –0: 1111
    - **2's complement Format** [+ve and –ve]
      - Range $[-(2^{n-1})$ to $+(2^{n-1} - 1)]$
      - [–8 to +7]
      - 0: 0000
- **Floating Point Data**
  - Single Precision Data (32 bit)
  - Double Precision (64 bit)

Q How to Find Number is Negitive ?

Soln If MSB bit [Sign] bit is

0 : +ve Number

1 : -ve Number.

MSB						LSB

Sign bit
- 0: +ve
- 1: -ve

4 bit Binary	Unsigned Data	Signed Magn. Data	1's Complement Data	2's Complement Data
0000	0	$\boxed{+0}$	$\boxed{+0}$	+0
0001	1	+1	+1	+1
0010	2	+2	+2	+2
0011	3	+3	+3	+3
0100	4	+4	+4	+4
0101	5	+5	+5	+5
0110	6	+6	+6	+6
0111	7	+7	+7	+7
1000	8	$\boxed{-0}$	-7	-8
1001	9	-1	-6	-7
1010	10	-2	-5	-6
1011	11	-3	-4	-5
1100	12	-4	-3	-4
1101	13	-5	-2	-3
1110	14	-6	-1	-2
1111	15	-7	$\boxed{-0}$	-1

Redundant Representations for '0'

How to Find 1's complement & 2's complement:

1's Complement:

0 $\xrightarrow{\text{Converted into}}$ 1

1 ⟶ 0

2's complement

1's complement
+ 1

	1's Complement	2's Complement
eg 1000 ⇒	0111 [-7]	0111 + 1 = 1000 ⇒ 1000 ⇒ [-8]
eg 1001 ⇒	0110 [-6]	0110 + 1 = 0111 ⇒ 0111 [-7].
eg 1101 ⇒	0010 [-2]	0010 + 1 = 0011 ⇒ 0011 [-3].
eg 1011 ⇒	0100 [-4]	0100 + 1 = 0101 ⇒ 0101 [-5].

Q) In Computer System WHY 2's Complement are Used to Represent Negitive Number, even we have Signed & 1's Complement also?

Soln

Signed Magnitude.

+0 :	0000
−0 :	1000

1's Complement

+0 :	0000
−0 :	1111

Redundant Representation of '0' in Signed & 1's Complement

So 2's Complement are Used in Computer System.

n bit 2's Complement Range $= -(2^{n-1})$ to $+(2^{n-1}-1)$

eg 4 bit 2's Complement Range $= -(2^{4-1})$ to $+(2^{4-1}-1)$

$= -8$ to $+7$.

Floating Point Representation

WHY Floating point Representation ?

To Represent very-very Large Number = (986512247....) Nearly: [∞]

To Represent very-very small Number = (0.............) Nearly: [...0]

16 bit Number Range $= -(2^{16-1})$ to $+(2^{16-1}-1) \Rightarrow -2^{15}$ to $+2^{15}-1$

$\Rightarrow \boxed{-32k \text{ to } +(32k-1)}$

we Want to Represent

51,000 its Not possible With 16 bit Data.

Floating-Point Representation

Principles

- With a fixed-point notation it is possible to represent a range of positive and negative integers centered on or near 0.
- By assuming a fixed binary or radix point, this format allows the representation of numbers with a fractional component as well
- Limitations:
  - Very large numbers cannot be represented nor can very small fractions
  - The fractional part of the quotient in a division of two large numbers could be lost

Floating-Point Representation

16 bit fixed point data format then

Range $= -2^{16-1}$ to $+ (2^{16-1} - 1)$

$\Rightarrow \quad \boxed{-(2^{15}) \text{ to } + (2^{15} - 1)}$

If we want to store 61,000 then we cannot store

Because range [–32k to + 32 k –1]

So floating point representation is to represent very large data and very small fraction and consume less memory

Floating point used to represent
$\left\{ \begin{array}{l} +8.5641000000000\ldots \; [\Rightarrow \infty] \\ +0.0000000007892 \quad \Rightarrow [\Rightarrow 0] \end{array} \right.$

Floating Point Representation

S	E	M

S[sign] — 0 [+ve]
S[sign] — 1 [-ve]

E|BE : Biased Exponent | Exponent

M : Mantissa.

$$\underbrace{\pm}_{\text{sign}} \; 0.\underbrace{xxxxxxx}_{\text{mantissa}} \times 2^{e}$$

e or AE

e : exponent. (Actual exponent)

$$\boxed{E = e + bias}$$

$$\boxed{BE = AE + bias}$$

Floating-Point Representation

S	E	M

S: sign bit — 0 +ve, 1 -ve

E: exponent

M: Mantissa

S

$$+/-\ \ \underbrace{x.\dots\dots}_{M} \times 2^{e}$$

e: exponent
Actual exponent [AE]

$. \dots\dots\dots \times 2^{e}$

6.5 in Binary $\Rightarrow$ 110.1

Number System

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
128	64	32	16	8	4	2	1	.	$1/2$	$1/4$	$1/8$	$1/16$	$1/32$	$1/64$
									(.5)	(.25)	(.125)	(0.0625)	. . .	.

① $6.5 \Rightarrow 110.1$

② $5.0 \Rightarrow 101.0$

16 8 4 2 1 . (.5) (.25) (.125) (0.0625)

(3) 11.5 $\Rightarrow$ 1011.1

(4) 13.25 $\Rightarrow$ 1101.01

(5) 16.625 $\Rightarrow$ 10000.101

(6) 6.37 $\Rightarrow$ 110.011

(7) 19.25 $\Rightarrow$ 10011.01

16 8 4 2 1

(8) $4.75 \Rightarrow 100.11$

(9) $11.625 \Rightarrow 1011.101$

(10) $37 \Rightarrow \overset{32}{1}\overset{16}{0}\overset{8}{0}\overset{4}{1}\overset{2}{0}\overset{1}{1}$

(11) $.37 \Rightarrow 0.011\ldots$

(12) $13.75 \Rightarrow 1101.11$

(13) $29.625 \Rightarrow 11101.101$

$.625 \Rightarrow 0.101$

$$\begin{array}{ll} 1 & 0.50 \\ 0 & 0.00 \\ 1 & 0.125 \end{array}$$

$+(6.5)$

$\boxed{110.1}$

$$\begin{array}{cccccc} 2^3 & 2^2 & 2^1 & 2^0 & . \; 2^{-1} & 2^{-2} \\ 8 & 4 & 2 & 1 & . \; (.5) & (.25) \\ 1 & 1 & 0 & . & 1 & \end{array}$$

110.1 ✓

0110.1
00110.1
000110.1
0000110.1

Not

110.1

Q $\boxed{+(6.5)}$

$+ \quad 110.1$

$\boxed{+/- \; 0.xxxx \times 2^{e}}$

$\boxed{+0.1101 \times 2^{+3}}$

S	E	M
1bit	2bit	4bit

v.v.v.v.v.v.v. Imp.

$\boxed{6.5} \Rightarrow 110.1$

$\Rightarrow \underline{0.1101 \times 2^{+3}}$

CROSS CHECK

$$\begin{array}{cccc} 2^{-1} & 2^{-2} & 2^{-3} & 2^{-4} \end{array}$$

$$[0.1101] \times 2^{+3}$$

$$[0.2^{-1} + 2^{-2} + 2^{-4}] \times 2^{+3}$$

$$\Rightarrow (2^{-1} \times 2^{+3}) + (2^{-2} \times 2^{+3}) + (2^{-4} \times 2^{+3})$$

$$\Rightarrow 2^{+2} + 2^{+1} + 2^{-1}$$

$$\Rightarrow 4 + 2 + 0.5 = (6.5)$$

$$= \boxed{6.5}$$

Another Method

to check 2^{+ve}

or

2^{-ve}.

110.1

Q $+(6.5)$

$+\ 110.1$

$$+/-\ 0.xxxx \times 2^{e}$$

$$+0.1101 \times 2^{+3}$$

2nd Technique

Databits side

Right Alignment : 2^{+ve}

Left Alignment : 2^{-ve}

S	E	M
1bit	2bit	4bit

V.V.V.V.V.V.Imp.

$\boxed{6.5} \Rightarrow 110.1$

$\Rightarrow 0.1101 \times 2^{+3}$

CROSS CHECK

$$[0.\overset{2^{-1}}{1}\overset{2^{-2}}{1}\overset{2^{-3}}{0}\overset{2^{-4}}{1}] \times 2^{+3}$$

$$[0.2^{-1}+2^{-2}+2^{-4}] \times 2^{+3}$$

$$\Rightarrow (2^{-1}\times 2^{+3}) + (2^{-2}\times 2^{+3}) + (2^{-4}\times 2^{+3})$$

$$\Rightarrow 2^{+2} + 2^{+1} + 2^{-1}$$

$$\Rightarrow 4+2+0.5 = (6.5)$$

$\Rightarrow \boxed{6.5}$

Q1 $+(6.5)$

S	E	M
1 bit	2 bit	4 bit

$+110.1$

$+0.\underbrace{1101}_{\text{Mantissa}} \times 2^{+3}$

Sign = 0

M = 1101

$e = +3 \Rightarrow [11]$

S	e	M
0	11	1101

Ans

Q.2 $+(4.5)$

$+(100.1)$

$\Rightarrow +0.1001 \times 2^{+3}$

$S: 0$

$e = 3\ (11)$

$M: 1001$

1bit	2bit	4bit
S	e	M

S	e(2bit)	M(4bit)
0	1 1	1001

Ans

Q.3 $+(4.75)$

$+(100.11)$

$\Rightarrow + 0.10011 \times 2^{+3}$

$S = 0$

$e = 3$ [11]

$M = 10011$

S	e(2bit	Mantissa.
1bit	2bit	5bit

Sign(1bit)	e(2bit)	Mantissa (5bit)
0	11	10011

Q. 1 $+6.5$

$6.5 = (110.1)_2$

$$\underbrace{0.}_{S}\underbrace{1101}_{M} \times \underbrace{2^3}_{2^e}$$

$S = 0\ (+)$

$M = 1101$

$e = 3 = (11)_2$

S	e	M
0	11	1101

Very. Imp

$6.5 = 110.1$

$= .1101 \times 2^3$

$= [.2^{-1} + 2^{-2} + 2^{-4}] \times 2^3$

$= [2^2 + 2^1 + 2^{-1}]$

$= 6.5$

Q. 2 $+4.5$

100.1

0.1001×2^3

$S = 0$ (+ve)

$M = 1001$

$e = 3$ [11]

S	e	M
0	11	1101

Q. 3 $+4.75$

100.11

$.10011 \times 2^{3}$

$S = 0$

$M: 10011$

$e = 3 \Rightarrow (11)_2$

S	e	M
0	11	10011

NOTE:

Mantissa alignment process is used to adjust the decimal point; in this process right alignment increments the exponent and left alignment decrements the exponent.

$2^{+\text{shift}}$ power(+) = Right alignment ⇒ Increment the exponent

$2^{-\text{shift}}$ power (-) = Left alignment ⇒ Decrease the exponent

Right Alignment

6.5

110.1

2^{+3}

$\Rightarrow .1101 \times 2^3$

$\Rightarrow [.2^{-1} + 2^{-2} + 2^{-4}] \times 2^3$

$\Rightarrow 2^2 + 2^1 + 2^{-1}$

$\Rightarrow 4 + 2 + 0.5$

$\Rightarrow$ 6.5 Ans

Left Alignment

Data: $0.000000000101 \times 2^{+5}$

$1.01 \times 2^{-8} \times 2^{+5}$

$[1.01 \times 2^{+5-8}]$

$+1.01 \times 2^{-3}$

(Align to use upto 8 times)

Left

Q WHY 'E' Required ?

'E' : Bias Exponent

1bit 4bit 4bit

s	e	m

Q $+0.001010$

$\Rightarrow +0.\underbrace{1010}_{M} \times 2^{-2}$

$\boxed{S: 0 \ [+ve]}$

M : 1010

$\boxed{e = -2}$ $\boxed{e = -ve}$ → No Provision to tell that exponent is Negitive.

Number is positive [∵ S=0] but exponent is Negitive.

→ signbit(0)

How to Deal with this Negitive exponent.

if exponent is negitive

then taking 2's complement

Q $+0.001010$

$\Rightarrow +0.\underbrace{1010}_{M} \times 2^{-2}$

1 bit	4 bit	4 bit
s	e	m

$\boxed{S : 0 \ [+ve]}$

$M : 1010$

$\boxed{e = -2}$

2's complement.

$0010 \Rightarrow 1101$
$+1$
1110

$\boxed{e = 1110}$

	e (4 bit)	4 bit (m)
0	1110	1010

Ans

$\boxed{e = -2}$

2's complement

Q But My Question is How we can say

$e = \underline{1110} \Rightarrow e = 14$

or

or

$e = 1110$ 2's Complement of -2

So Here 'e' create Confusion

if another example.

with having $e = 14$

Q $\boxed{+0.1010 \times 2^{+14}}$

s	e	m
1bit	4bit	4bit

$S = 0$

$M : 1010$

$e = +14 \Rightarrow 1110$

s(1bit)	e(4bit)	m(4bit)
0	1110	1010

Ans

$$\boxed{e = +14}$$

Q.1 WHY Biasing (Biased Exponent(BE)/E) is Required ?

Q.2 How bias value is decided/selected ?

Q $+0.001010$

$\Rightarrow +0.\underbrace{1010}_{M} \times 2^{-2}$

1bit	4bit	4bit
s	e	m

Soln

$\boxed{S : 0 \ [+ve]}$

M : 1010

$\boxed{e = -2}$

2's complement.

$0010 \Rightarrow 1101$
$+1$
1110

$\boxed{e = 1110}$

$\boxed{e = -2}$

	e(4bit)	4bit(m)
0	1110	1010

Ans

Q But My Question is

How we can say

$e = \underline{1110} \Rightarrow e = 14$

or

or

$e = 1110$ 2's Complement of -2

So Here 'e' create Confusion

Q $+0.1010 \times 2^{+14}$

s	e	m
1bit	4bit	4bit

Sol^n 1

$s = 0$

$m : 1010$

$e = +14 \Rightarrow 1110$

s(1bit)	e(4bit)	m(4bit)
0	1110	1010

$$\boxed{e = +14}$$

Sol^n 1 If we take Actual exponent [e] then there is No Provision to Represent the Negitive exponent. Bcz Sign bit is telling Number +ve [0] or −ve [if S=1].

If we take 2's complement of Negitive exponent then it Creates Ambiguity. Like in Previous 2 example

$$\left.\begin{array}{l} -2 : 1110 \\ +14 : 1110 \end{array}\right\} \text{Create Ambiguity.}$$

So the solution is instead of writing in 2's complement

Used Bias exponent [E/BE]

$$\boxed{E = e + bias}$$

OR

$$\boxed{BE = AE + bias}$$

$$\boxed{0.\underbrace{\text{- - -}}_{\text{mantiss}} \times 2^{e(AE)}}$$

Soln 2

All Number (-ve or +ve)

biasing : To Convert a Number into '0' or +ve Number.

How bias value Select : ?

n bit 2's Complement Range $= -(2^{n-1})$ to $+(2^{n-1}-1)$

4 bit 2's Complement $= -(2^{4-1})$ to $+(2^{4-1}-1)$

$= \boxed{-8 \text{ to } +7}$

How bias is Select ?

if Exponent = k bit

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

if exponent is k bit then bias = 2^{k-1}

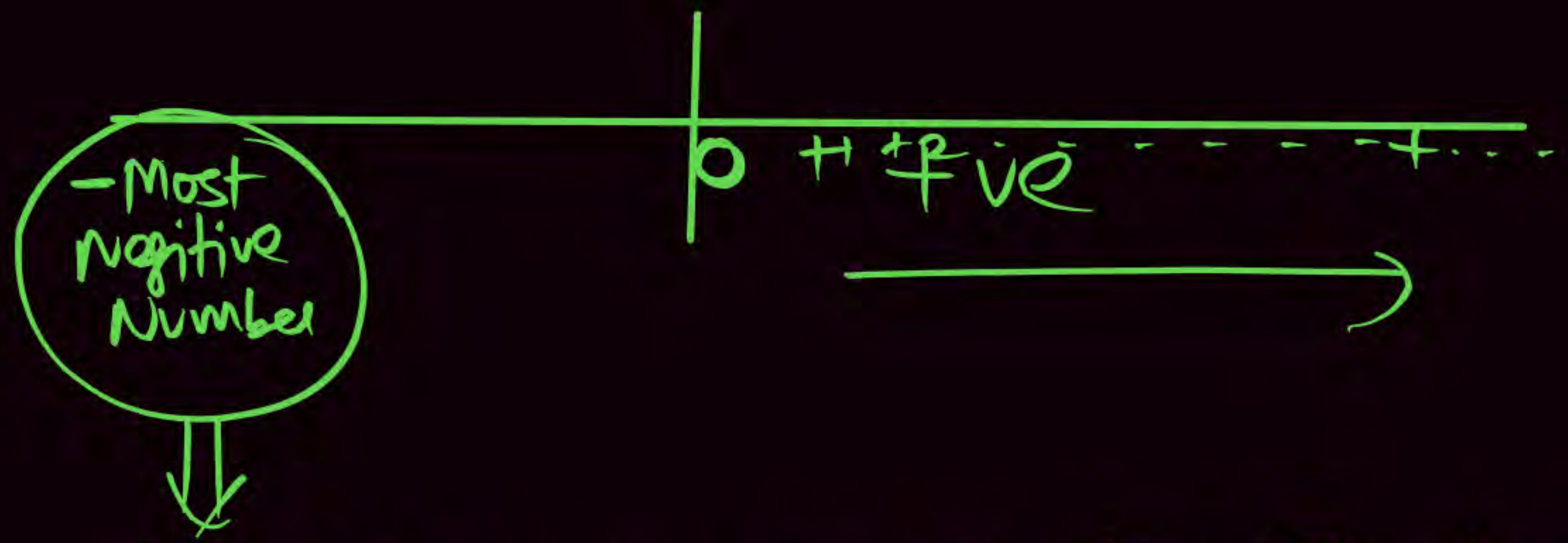
if exponent k bit then 2's Complement = $\boxed{(-2^{k-1})}$ to $(+2^{k-1}-1)$

exponent 4 bit 2's Complement Range = -2^{4-1} to $+2^{4-1}-1$

$= \boxed{-8 \text{ to } +7}$

Note

In order to Convert ALL Number (Negitive & Positive) Into Positive Number take Most Highest Negitive Number & ADD as a bias.



Add as bias so we get '0' or +ve Number

Exponent = k bit

Exponent = 4 bit

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

$$\Rightarrow 2^{4-1}$$

$$\boxed{\text{bias} = 8}$$

2's complement $= (-2^{4-1})$ to $+(2^{4-1}-1)$

$= -8$ to $+7$

AE (e)	BE [E] (e + bias)
−8	−8+8 = 0
−7	−7+8 = 1
−6	= 2
−5	= 3
−4	= 4
⋮	
0	0+8 = 8
1	1+8 = 9
⋮	
+7	7+8 = +15

Q. 4 $+0.00101$

0.101×2^{-2}

$M = 101$

$E = -2$

$S = 0$

S	E(4bit)	M(5 bit)
0	1110	10100
	E	M

$E = -2 = (1110)_2$ 2's complement

Biasing: is method in which we convert the negative number into the positive number

Bit	Bit	Bit
S	E	M

S = Sign

E/BE = Exponent or

BE = bias exponent

M = Mantissa

E = e + bias

$\boxed{\text{Bias} = 2^{K-1}}$ where K is exponent bits

Example

If K = 4 bits

Exponent = 4 bit then

bias = $2^{K-1} = 2^{4-1} = 8$

Exponent (k) = 4 bit

bias = $2^{4-1} = 8$

$E = e + \text{bias}$

$\boxed{E = e + 8}$

1 Bit	4 Bit	
S	E	M

x bit

$\boxed{Bias = 2^{K-1}} = 2^{4-1}$

bias = 8

E = e + bias

E = e + 8

E = 4 bit

or

Excess 8 code

$\boxed{2^{K-1} = 8}$

$2^{K-1} = 23$

K - 1 = 3

K = 4

E = 4 bit

Excess 8 code

$\Downarrow$

$\boxed{\text{bias} = 8}$

e [original exponent]		Stored exponent [BE] E E = e + bias ⇒ e + 8
-8	-8+8	0
-7	-7+8	1
-6	-6+8	2
-5	-5+8	3
-4	-4+8	4
-3	-3+8	5
-2	-2+8	6
-1	-1+8	7
0	0+8	8
1	1+8	9
2	2+8	10
3	3+8	11
4	4+8	12
5	5+8	13
6	6+8	14
7	7+8	15

Exponent = k bit

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

Excess - 8 : bias = 8

Excess - 16 : bias = 16

Excess - 32 : bias = 32

Excess - 64 : bias = 64.

Excess : 8

$2^{k-1} = 8$

$2^{k-1} = 2^3$

$k - 1 = 3$

$\boxed{k = 4}$ Exponent = 4 bit

$bias = 2^{4-1} = 8$

Exponent = k bit

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

Excess - 8 : bias = 8

Excess - 16 : bias = 16

Excess - 32 : bias = 32

Excess - 64 : bias = 64.

$\boxed{\text{Excess : 16}}$

$2^{k-1} = 2^4$

$2^{k-1} = 2^4$

$k - 1 = 4$

$\boxed{k = 5}$ Exponent = 4 bit

$bias = 2^{5-1} = \boxed{bias = 16}$

Exponent = k bit

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

Excess - 8 : bias = 8

Excess - 16 : bias = 16

Excess - 32 : bias = 32

Excess - 64 : bias = 64.

$\boxed{\text{Excess : } 32}$

$$2^{k-1} = 2^5$$

$$2^{k-1} = 2^5$$

$$k-1 = 5$$

$\boxed{k = 6}$ Exponent = 6 bit

$$bias = 2^{6-1} = \boxed{bias = 32}$$

Q.

From previous question

$\boxed{0.00101}$

0.101×2^{-2}

$M = 101$

$Bias = 2^{5-1}$

$Bias = 16$

$\boxed{e = -2}$

$\boxed{E = e + bias}$

$E = -2 + 16$

$\boxed{E = 14}$

$E = (01110)_2$

Formula: $(-1)^S \times 0.M \times 2^e$

$(-1)^0 \times 0.101 \times 2^{E-bias}$

1 bit	5 bit	4 bit
S	E	M

$bias = 2^{5-1} \Rightarrow 2^4 \Rightarrow \boxed{bias = 16}$

$E = e + bias$

1 bit	5 bit	4 bit
0	01110	1010

Ans

$0.101 \times 2^{14-16} = 0.101 \times 2^{-2}$

0.000101 Ans

S	E	M
1 bit	x bit	y bit

Normalized Mantissa

Explicit Normalized

Syntax

$$\boxed{0.\underbrace{1\ldots\ldots}_{M} \times 2^{e}}$$

Formula to get number

[value formula]

$(-1)^{S} \times 0.M \times 2^{e}$

$$\boxed{(-1)^{S} \times 0.M \times 2^{E-\text{bias}}}$$

Implicit Normalized

Syntax

$$\boxed{1.\underbrace{\ldots\ldots}_{M} \times 2^{e}}$$

Formula to get number

[value formula]

$(-1)^{S} \times 1.M \times 2^{e}$

$$\boxed{(-1)^{S} \times 1.M \times 2^{E-\text{bias}}}$$

Explicit

0.1 After the point,

Immediate first bit should be 1

Example

(101.11)

0.10111×2^3

$M = 10111,$

$e = 3$

$E = e + bias$

Implicit

Before the point 1 means 1. ......

Example

(101.11)

1.0111×2^2

$M = 0111,$

$e = 2$

$E = e + bias$

Q. 1 +(4.875) format

Then do explicit and implicit normalization

1 bit	4 bit	5 bit
0	1011	10011

Explicit

(+4.875)

100.111

0.100111×2^3

$M = 100111$

$e = 3$, bias $= 2^{4-1}$

$E = 3 + 8$

$E = 11$

$E = 1011$

Value Formula: $(-1)^S \times 0.M \times 2^e$

$(-1)^0 \times 0.10011 \times 2^{11-8}$

0.10011×2^3

100.11

4.75

(Not getting very accurate)

Implicit

$(+4.875)$

100.111

1.00111×2^2

$M = 00111$

$e = 2$, bias $= 2^{4-1}$

$E = 2 + 8$

$E = 10$

$E = 1010$

1 bit	4 bit	5 bit
0	1010	00111

Value Formula: $(-1)^S \times 1.M \times 2^e$

$(-1)^0 \times 1.00111 \times 2^{10-8}$

1.00111×2^2

100.111

4.875

(Getting very accurate)

Q. Consider a 16 bit register used to store floating point number. Mantissa is Explicit normalized signed fraction number. Exponent is in Excess-32 form then what is 16-bit for $-(29.75)_{10}$ in the register?

Solution

1 bit	6 bit	9 bit
S	E	M

-29.75

-11101.11

0.1110111×2^5

M: 1110111

$e = 5$

$\text{bias} = 2^{6-1}$

$\text{bias} = 32$

$E = 5 + 32 = 37 = (100101)_2$

S(1 bit)	E(6 bit)	M(9 bit)
1	100101	111011100

Q.

$+21.75$

Implicit ?

1 bit	7 bit	8 bit
S	E	M

10101.11

1.010111×2^4

$M = 010111$

$e = 4$, bias $= 2^{7-1}$

$E = 4 + 64$

$E = 68 = (1000100)_2$

Value Formula:

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

$(-1)^0 \times 1.010111 \times 2^{68-64}$

1.010111×2^4

$10101.11 = (21.75)_{10}$

Ans

S(1bit)	E(7bit)	M(8 bit)
0	1000100	01011100

Hexadecimal $= (445C)_{16}$

THANK YOU!