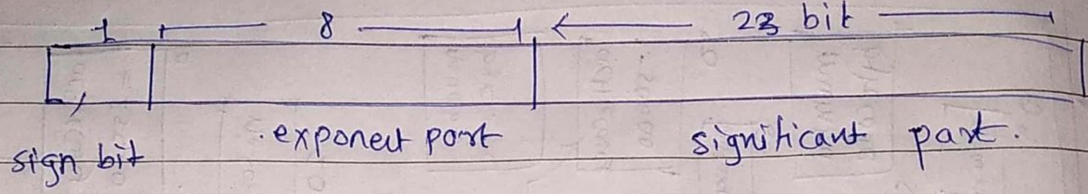
* floating point representation

Single precision format (32 bit)



Procedure :-

1. convert floating point to binary number
e.g.

1.56 → fractional part
1 ← intergral part

① convert integral part into binary
1 ⇒ $(0001)_2$

② fractional part :
multiply a number with by 2 untill we will get 1.00 value

∴ 0.56 × 2 = 1.12	0.72 × 2 = 1.44
0.12 × 2 = 0.24	0.44 × 2 = 1.88
0.24 × 2 = 0.48	0.88 × 2 = 1.76
0.48 × 2 = 0.96	0.76 × 2 = 1.52
0.96 × 2 = 1.92	0.52 × 2 = 1.04
0.92 × 2 = 1.84	0.04 × 2 = 0.08
0.84 × 2 = 1.68	0.08 × 2 = 0.16
0.68 × 2 = 1.36	0.16 × 2 = 0.32
0.36 × 2 = 0.72	0.32 × 2 = 0.64

Scanned by CamScanner

$0.64 \times 2 = 1.28$
$0.28 \times 2 = 0.56$
$0.56 \times 2 = 1.12$
$0.12 \times 2 = 0.24$
$0.24 \times 2 = 0.48$

skip theset bits because there is repeated part.

$\therefore\ 1.56 = (1.10001111\ 0111\ 1100\ 0010\ \cancel{100})_2$

2) make the converted binary number to normalize form
Normalization.
$1.\text{significant bit} \times 2^{\text{exponent}}$

so. $1.1000\ 1111\ 0111\ 110000 \times 2^0$

3) Add bias to exponent.
In floating number, no concept called 2's complement to store negative number.
$\therefore$ biasing concept is there to convert -ve to +ve.

$bias_n = 2^{n-1} - 1;$
Here we have to allocate 8 bits for exponent. so $n=8$

so. $2^7 - 1 = 127$

= Actual exponent + bias value
$= 0 + 127$
$= 127$. $\Rightarrow (0111\ 1111)_2$ (0111 1011).

exponent   significant

8 bit

| 0 | 0 | φ | φ | φ | φ | 0 | φ | φ | 1000 1111 0111 110000 |

+ve number

2) 2.34

2 ⇒ 0010.

.34 ⇒

$0.34 \times 2 = 0.68$	$0.64 = 1.28$
$0.68 \times 2 = 1.36$	$0.28 \times 2 = 0.56$
$0.36 \times 2 = 0.72$	$0.56 \times 2 = 1.12$
$0.72 \times 2 = 1.44$	$0.12 \times 2 = 0.24$
$0.44 \times 2 = 0.88$	$0.24 \times 2 = 0.48$
$0.88 \times 2 = 1.76$	$0.48 \times 2 = 0.96$
$0.76 \times 2 = 1.52$	$0.96 \times 2 = 1.92$
$0.52 \times 2 = 1.04$	$0.92 \times 2 = 1.84$
$0.04 \times 2 = 0.08$	$0.84 \times 2 = 1.68$ ← repeated
$0.08 \times 2 = 0.16$	
$0.16 \times 2 = 0.32$	
$0.32 \times 2 = 0.64$	

∴ 0010 . ~~01010111000000000000000~~

10. 0101011100001010000111

③ Normalization

$1.00101010111000010100 00111 \times 2^1$

④ biasing :-

$1 + 2^7 - 1 = 1 + 127$

= 128 = 1000 0000

0	1000 0000	1.0010 1011 0000 10100011 1
sign bit	exponent	significant

3) 3.65.

→ binary conversion

3 → 011

.65 → 0.65 × 2 = 1.30

0.30 × 2 = 0.60

0.60 × 2 = 1.20

0.20 × 2 = 0.40

0.40 × 2 = 0.80

0.80 × 2 = 1.60

0.60 × 2 = 1.20. ← stop: repeating the same sequence.

$\therefore (11.1010011)_2$

③ Normalization.

1.11010011×2^1

④ biasing

$1 + 2^7 - 1 = 1 + 128 - 1 = 128$

$= 1000\ 0000$

0	1000 0000	11010011001100110011001
1 bit	8 bit	23 bit.

Page No.
Date / /

(4) 4.103

⇒ (1) binary conversion.

4 → 100

.103 →

0.103 × 2 = 0.206
0.206 × 2 = 0.412
0.412 × 2 = 0.824
0.824 × 2 = 1.648
0.648 × 2 = 1.296
0.296 × 2 = 0.592
0.592 × 2 = 1.184
0.184 × 2 = 0.368
0.368 × 2 = 0.736
0.736 × 2 = 1.472
0.472 × 2 = 0.944
0.944 × 2 = 1.888
0.888 × 2 = 1.776

0.776 × 2 = 1.552
0.552 × 2 = 1.104.
0.104 × 2 = 0.208
0.208 × 2 = 0.416
0.416 × 2 = 0.832.
0.832 × 2 = 1.664.
0.664 × 2 = 1.328.
0.328 × 2 = 0.656
0.656 × 2 = 1.312.
0.312 × 2 = 0.624

$\boxed{100.0011010001011111000110 10}_2$

Normalization.

$1.00001101000101111100011 01 \times 2^2$

biasing

2 + 127 = 129 ⇒ $(1000\ 0001)_2$.

Representation.

0	10000001	00001101000101111000110 1

5) 5.97

⇒ binary conversion

5 ⇒ 101

.97 ⇒ 0.97 × 2 = 1.94

0.94 × 2 = 1.88

0.88 × 2 = 1.76

0.76 × 2 = 1.52

0.52 × 2 = 1.04

0.04 × 2 = 0.08

0.08 × 2 = 0.16

0.16 × 2 = 0.32

0.32 × 2 = 0.64

0.64 × 2 = 1.28

0.28 × 2 = 0.56

0.56 × 2 = 1.12

0.12 × 2 = 0.24

0.24 × 2 = 0.48

0.48 × 2 = 0.96

0.96 × 2 = 1.92

0.92 × 2 = 1.84

0.84 × 2 = 1.68

0.68 × 2 = 1.36

0.36 × 2 = 0.72

0.72 × 2 = 1.44

0.44 × 2 = 0.88

0.88 × 2 = 1.76

⇒ $(101.1111\,1000\,0101\,00011\,11101)_2$

Normalization

$(1.0111111000010100011\,1101)_2 \times 2^2$

biasing

$2 + 127 = 129 = (1000\;0001)_2$

0	1000 0001	0111 1110 0001 0100 0111 101