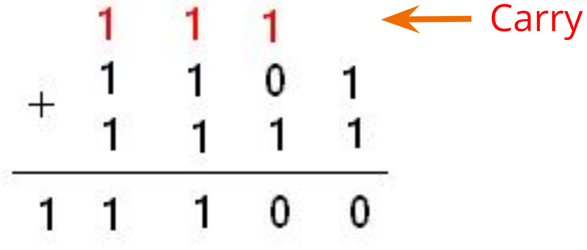
Arithmetic Operations

SCS 1205

Computer Systems

Binary Addition

Addition		Result	Carry
0 + 0	=	0	0
0 + 1	=	1	0
1 + 0	=	1	0
1 + 1	=	0	1



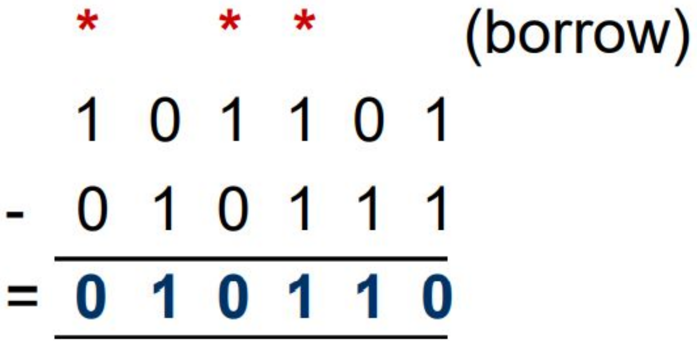
$$\begin{array}{rccccc} & & 1 & 1 & 1 & \\ & & 1 & 1 & 0 & 1 \\ + & & 1 & 1 & 1 & 1 \\ \hline & 1 & 1 & 1 & 0 & 0 \end{array}$$

Carry

Addition	Result	Carry
1 + 1 + 1	1	1

Binary Subtraction

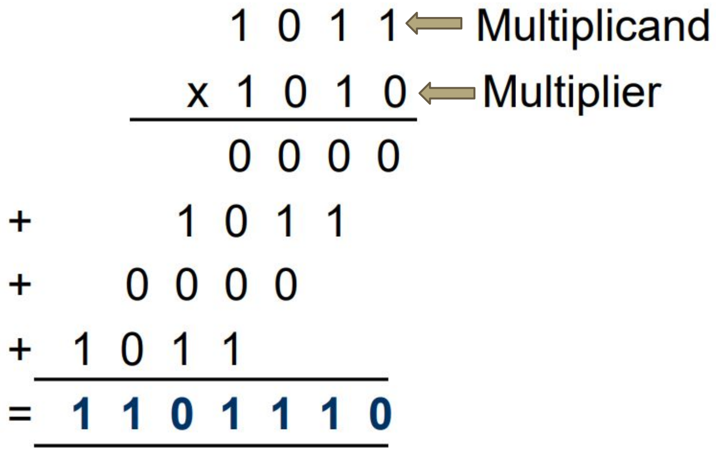
Subtraction		Result	
0 - 0	=	0	
0 - 1	=	1	* with borrow
1 - 0	=	1	
1 - 1	=	0	



```
    *       *   *           (borrow)
    1   0   1   1   0   1
-   0   1   0   1   1   1
= 0   1   0   1   1   0
```

Binary Multiplication

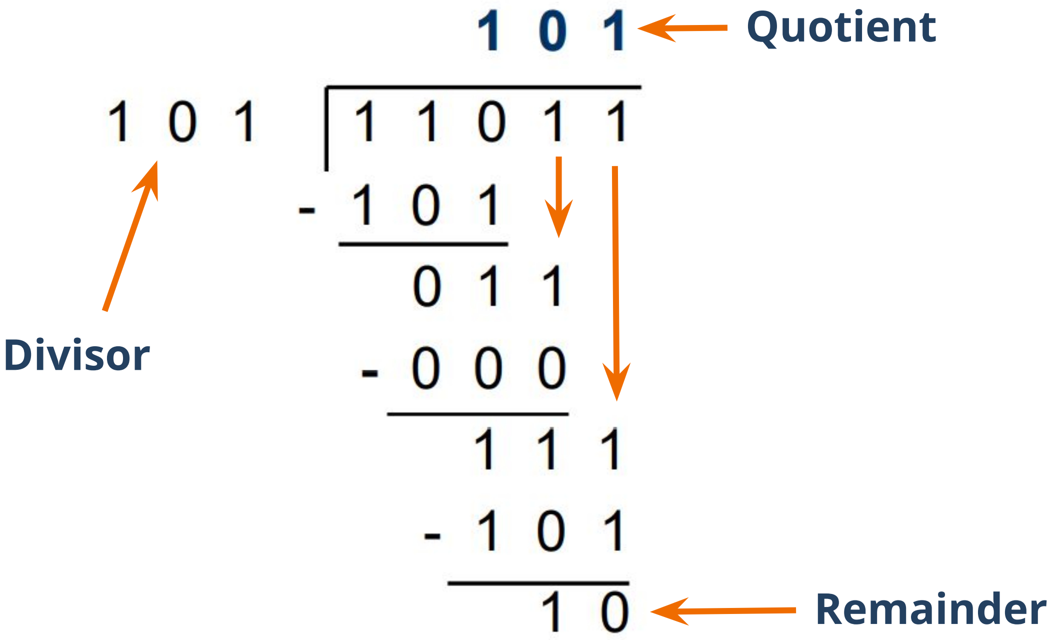
Multiplication		Result
0 * 0	=	0
0 * 1	=	0
1 * 0	=	0
1 * 1	=	1



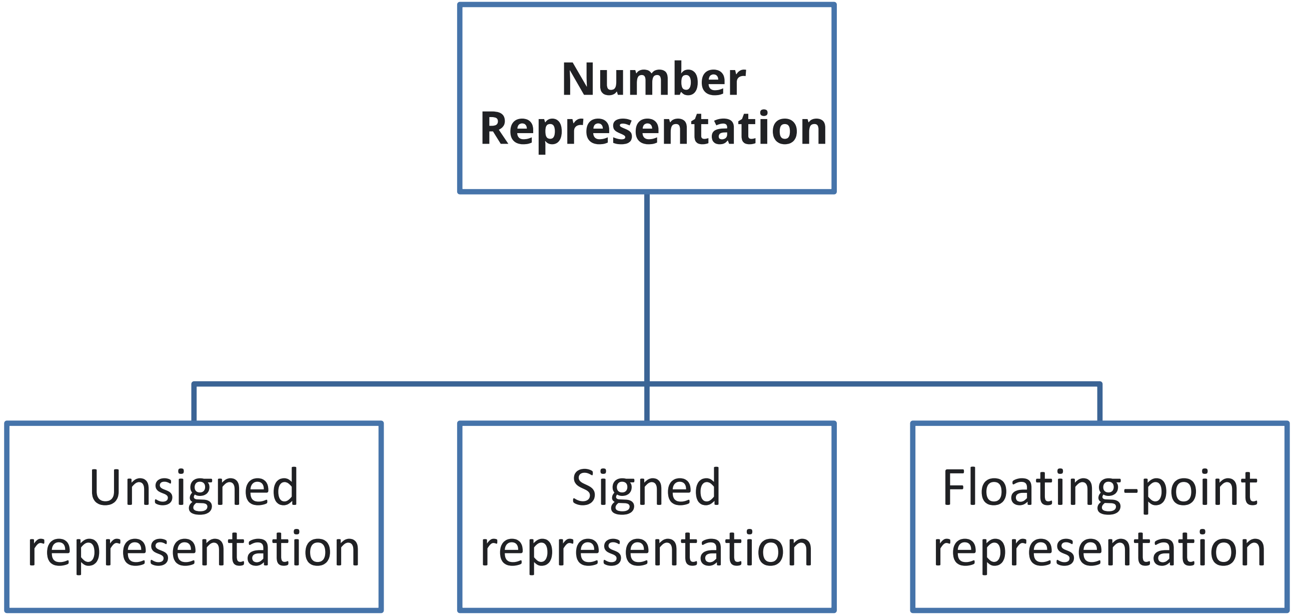
```
            1 0 1 1 ⟸ Multiplicand
          x 1 0 1 0 ⟸ Multiplier
        -----------
            0 0 0 0
+         1 0 1 1
+       0 0 0 0
+     1 0 1 1
     -------------
=     1 1 0 1 1 1 0
     -------------
```

Binary Division

- Decide whether it is 0 or 1



Number representation



Number Representation

Unsigned representation

Signed representation

Floating-point representation

Unsigned Binary Numbers

Magnitude is the bits in the pattern which store the size of the number

- All the bits are used for representing magnitude

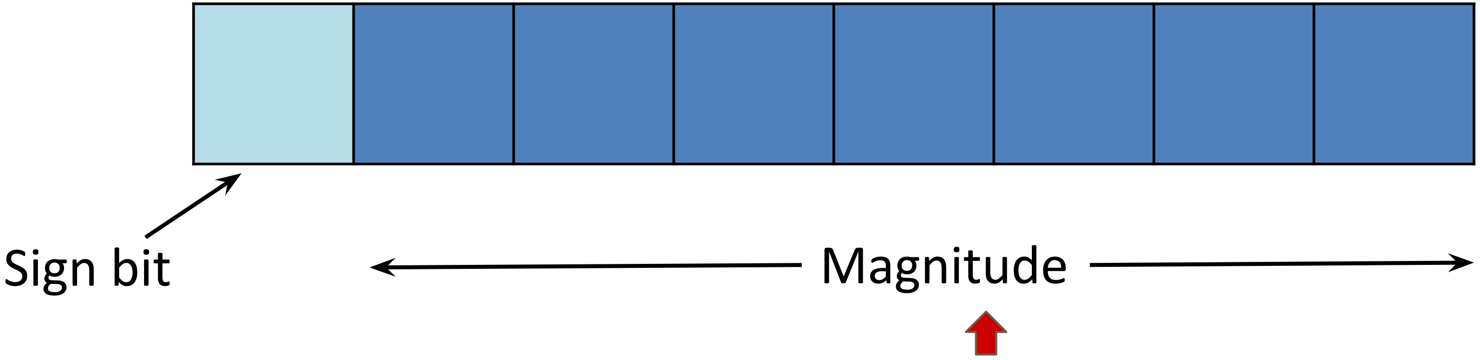
Smallest	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	= 0
Largest	1	1	1	1	1	1	1	1	
	128	64	32	16	8	4	2	1	= 255

Signed Binary Numbers

- Left most number is the **sign bit.**

  0 -> positive

  1 -> negative



(bits in the pattern store the size of the number)

Signed Binary Numbers

- The largest magnitude is **127**, which is approximately half of what is for unsigned binary numbers.

0	1	1	1	1	1	1	1	
	64	32	16	8	4	2	1	= 127

Problems in signed representation

- +0 and -0
- Need to consider both sign and magnitude in arithmetic

7 – 4 = ??

= 7+(-4)

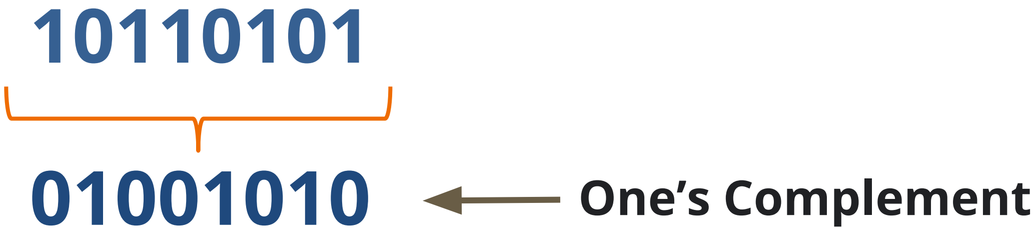
= 3

$$\begin{array}{rr} & 00000111 \\ + & 10000100 \\ \hline & 10001011 \\ \hline\hline \end{array} \quad = \quad -11$$

One’s Complement

A method which we can use to represent negative binary numbers in a signed binary number system.

- One’s complement of 1 is 0
- One’s complement of 0 is 1



10110101

01001010

One’s Complement

One's Complement

7 – 4 = ??

= 7+(-4)

00000100 → 4

11111011

one's complement value of 4

1111111 ← Carry

00000111 ← 7

+ 11111011 ← one's complement of 4

100000010

+ 1

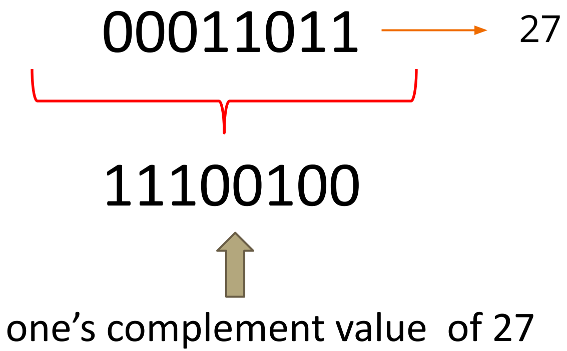
00000011 = **3**

- ❖ In one's complement if there is a carry value generated after the sign bit. it will be added to result.

One's Complement

115 – 27 = ??

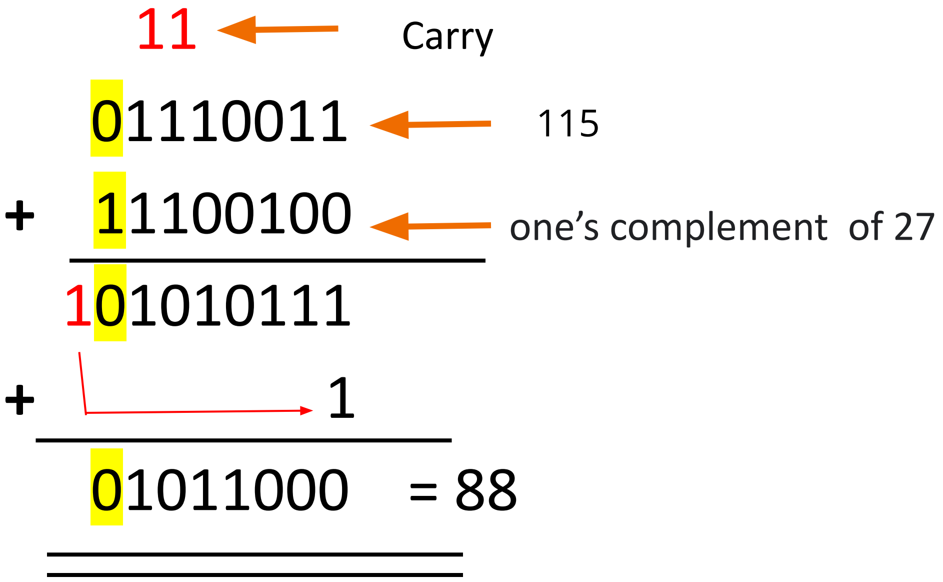
= 115+(-27)



00011011 → 27

11100100

one's complement value of 27



11 ← Carry

01110011 ← 115

+ 11100100 ← one's complement of 27

101010111

+ 1

01011000 = 88

Two's Complement

- 2's complement of a binary number is 1 added to the 1's complement of the binary number.

$$\begin{array}{r} 00000111 \\ 11111000 \\ +\qquad\quad 1 \\ \hline 11111001 \end{array}$$

00000111 ← Binary Number

11111000 ← One's complement

11111001 ← Two's complement

Two's Complement

7 – 4 = ??

= 7+(-4)

Binary → 00000100

1's complement → 11111011

+ 1

2's complement → 11111100

$$
\begin{array}{rrl}
 & 11111 & \text{Carry} \\
 & 00000111 & \\
+ & 11111100 & \\
\hline
 & 100000011 & = 3 \\
\hline\hline
\end{array}
$$

Two's Complement

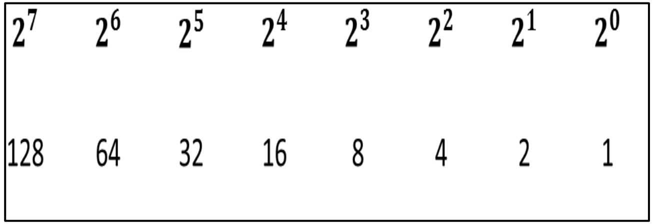
115 – 27 = ??

= 115+(-27)

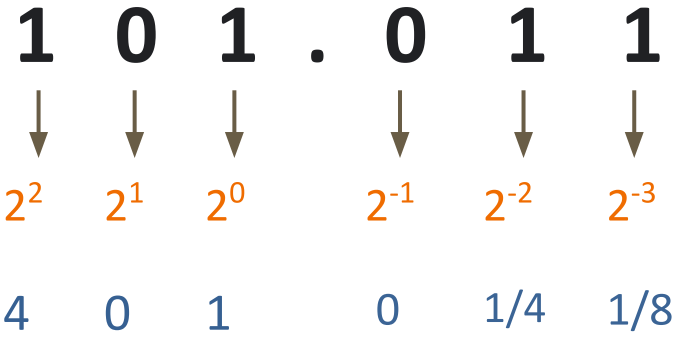
$$\begin{array}{rr} & \underbrace{00011011} \\ \text{1's complement} \longrightarrow & 11100100 \\ & +1 \\ \hline \text{2's complement} \longrightarrow & 11100101 \end{array}$$

$$\begin{array}{rrl} & 11\ \ \ 111 & \text{Carry} \\ & 01110011 & \\ + & 11100101 & \\ \hline & 101011000 & = 88 \\ \hline\hline \end{array}$$

Fractions in binary to decimal



2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1



1	0	1	.	0	1	1
2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
4	0	1		0	1/4	1/8

$$4 + 0 + 1 + \tfrac{1}{4} + \tfrac{1}{8} = 5\tfrac{3}{8}$$

$$101.011_2 = 5.375_{10}$$

Fractions : binary to decimal

1	1	1	.	0	1
↓	↓	↓		↓	↓
4	2	1		0	¼

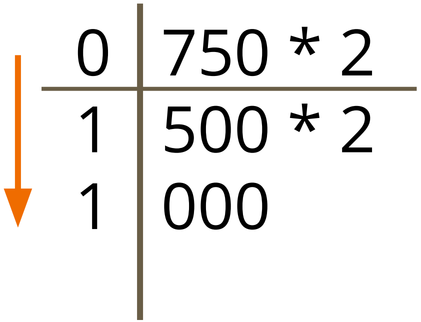
$$4 + 2 + 1 + {}^{1}/_{4} \;=\; 7\,{}^{1}/_{4}$$

$$111.01 = 7\,{}^{1}/_{4}$$

Fractions : decimal to binary

1. Multiply the number by the base (=2)
2. Take the integer on the left (0 or 1) as the coefficient.
3. Take the resultant fraction and repeat the division until resultant is zero

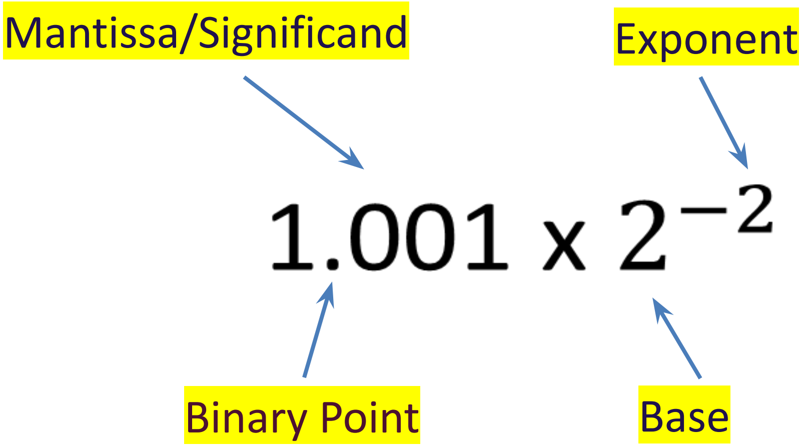
Eg : 0.750_{10}



0	750 * 2
1	500 * 2
1	000

$\mathbf{0.750_{10} = 0.11_2}$

Non-integer Representation (Float point)



Mantissa/Significand

Exponent

$$1.001 \text{ x } 2^{-2}$$

Binary Point

Base

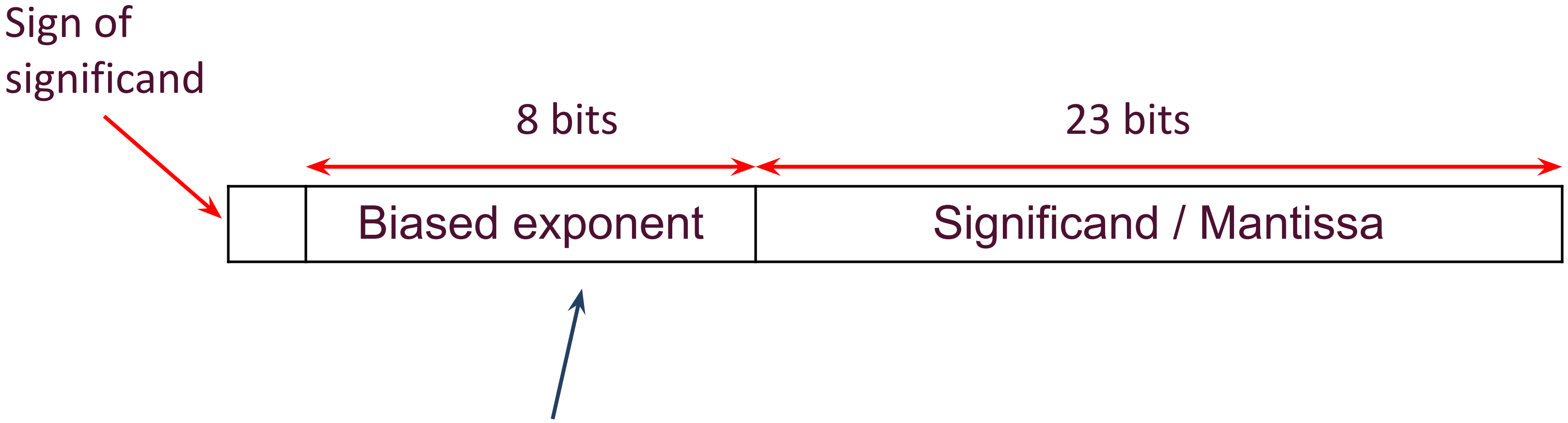
Scientific Notation

$$144.69 = 1.4469 \text{ x } 10^{2}$$

$$0.01001 = 1.001 \text{ x } 2^{-2}$$

Binary floating point representation

- 32-bit floating point representation
- Also called as single precision.



Binary representation = exponent + bias

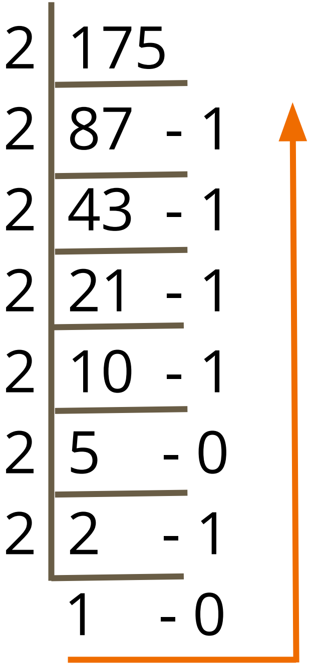
For 8 bits - > bias = 2^7-1 = **127**

Binary floating point representation

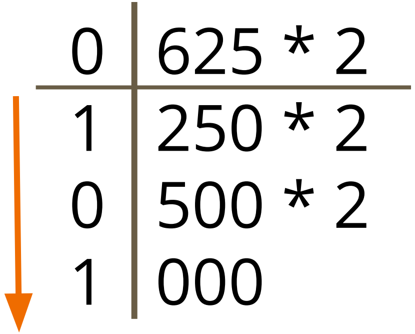
Represent 175.625 in 32-bit binary floating point number

Eg : Represent 175.625 in 32-bit binary floating point number

STEP 01 : Convert the given decimal number to binary



2	175	
2	87	- 1
2	43	- 1
2	21	- 1
2	10	- 1
2	5	- 0
2	2	- 1
	1	- 0



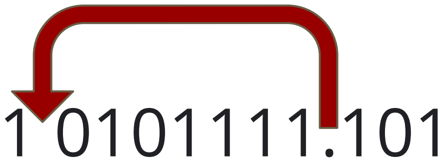
0	625 * 2
1	250 * 2
0	500 * 2
1	000

$$\mathbf{175.625_{10} = 10101111.101_2}$$

Eg : Represent 175.625 in 32-bit binary floating point number

STEP 02 : Represent binary number in the scientific notation

Binary number → 10101111.101_2



1 0101111.101

Exponent

$$10101111.101 = 1.\underbrace{0101111101}_{\text{Mantissa}} \times 2^{7}$$

Eg : Represent 175.625 in 32-bit binary floating point number

STEP 03 : Represent scientific notation in 32-bit binary floating number

i) Assign the sign bit

Exponent

$$1.\underbrace{0101111101}_{\text{Mantissa}} \times 2^{7}$$

0		

0; since 175.625 is a positive number

Eg : Represent 175.625 in 32-bit binary floating point number

STEP 03 : Represent scientific notation in 32-bit binary floating number

ii) Find the binary representation of biased exponent

$1.0101111101 \text{ x } 2^{7}$

(For a 8 bit exponent → bias = 127)

Biased exponent = bias + exponent

= 127 + 7

= 134

Binary representation of 134_{10} = 10000110_{2}

0	10000110	

Eg : Represent 175.625 in 32-bit binary floating point number

STEP 03 : Represent scientific notation in 32-bit binary floating number

iii) Assign the mantissa and fill the rest with zeros

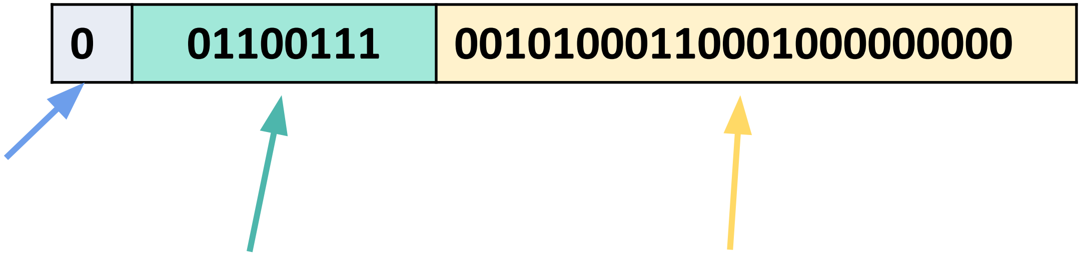
$$1.\underbrace{0101111101}_{\text{Mantissa}} \times 2^7$$

0	**1 0 0 0 0 1 1 0**	**0 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0**

Binary floating point representation

Eg : Represent the following 32-bit binary floating point number in scientific notation.

0	01100111	00101000110001000000000



Sign bit = 0;
positive

$01100111_2 = 103_{10}$
Bias adjustment = 103 - 127
= -24

Significand :

$= 1 + (0*2^{-1}) + (0*2^{-2}) + (1*2^{-3}) + (0*2^{-4}) + (1*2^{-5})+......$

$= 1 + 0 + 0 + ⅛ + 0 + 1/32 ....$

$= 1 + 0.1592407$

= 1.1592407

1.1592407×2^{-24} ← Simplify this to get the decimal scientific notation

Scientific Notation : $6.909613013 \times 10^{-8}$

Thank You!!