

Floating Point

How does RISC-V support numbers with fractions?

Scientific Notation is just a way to represent very large or very small numbers.

$$\Rightarrow 4500000 = \underbrace{4.5}_{\text{Coefficient}} \times \underbrace{10^6}_{\text{Base}} \quad \underbrace{6}_{\text{Exponent}}$$
$$\Rightarrow 0.00453 = \underbrace{4.53}_{\text{Coefficient}} \times \underbrace{10^{-3}}_{\text{Base}}$$

Decimal

$$\Rightarrow 5.64 \times 10^{33}$$
$$\Rightarrow -2.34 \times 10^{56}$$

Normalized.

$$\Rightarrow 109.64 \times 10^{33}$$
$$\Rightarrow 0.002 \times 10^{-4}$$

Not Normalized.

$$\Rightarrow +087.02 \times 10^9$$

In Binary,

$$\pm 1. \times \times \times_2 \times 2^{888}$$

IEEE-754 → defines floating point standard.

Single precision. (32-bit)

Double precision. (64-bit)

In case of double precision,

using more bits, you can represent a larger or a smaller number than single precision.

Normalized Number

Decimal Point (representing Decimal Numbers)

⇒ A decimal number is Normalized if :

- Only one digit before the decimal point.
- And that digit must be a non-zero number.

64.8×10^0 ✗ not a normalized number.

6.48×10^1 ✓ a normalized number.

>To normalize a number you need to shift the decimal point (.) left or right until you have a single non-zero digit before the decimal point.

⇒ If you shift left, the number of times you left shifted will be added with the exponent.

$$\Rightarrow 112.54 \times 10^{35}$$

$$\Rightarrow 1.1254 \times 10^{35+2} \Rightarrow 1.1254 \times 10^{37}$$

⇒ If you shift right, the number of times you right shifted will be subtracted from the exponent.

$$\Rightarrow 0.0065$$

$$\Rightarrow 0.0065 \times 10^0$$

$$\Rightarrow 6.5 \times 10^{0-3} = 6.5 \times 10^{-3}$$

Binary Point (representing Binary Numbers)

⇒ A binary number is Normalized if :

- Only one digit before the binary point.
- And that digit must be a non-zero number.

$11.00101 \times 2^{35} \times 10^0$ ✗ not a normalized number.

$1.100101 \times 2^{37} \times 10^0$ ✓ a normalized number.

To normalize a number you need to shift the binary point (.) left or right until you have a single non-zero digit before the binary point.

⇒ If you shift left, the number of times you left shifted will be added with the exponent.

$$\Rightarrow 1.10.111 \times 2^{35}$$

$$\Rightarrow 1.10111 \times 2^{35+2}$$

⇒ If you shift right, the number of times you right shifted will be subtracted from the exponent.

$$\Rightarrow 0.00110$$

$$\Rightarrow 0.00110 \times 2^0$$

$$\Rightarrow 1.10 \times 2^{0-3} = 1.10 \times 2^{-3}$$

IEEE Floating-Point Format

| | Sign Bit | Exponent | Fraction |
|-----------|----------|----------|----------|
| Single P. | 1 bit | 8 bits | 23 bits |
| Double P. | 1 " | 11 " | 52 " |

Single Precision (32 Bit)

| Sign Bit | Exponent | Fraction |
|----------|----------|----------|
| 1 | 8 | 23 |

Sign Bit = 0 ⇒ positive number

1 ⇒ negative

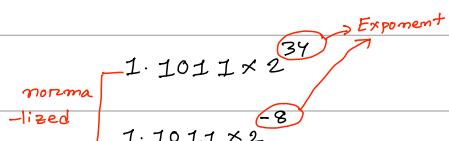
Exponent = It will be represented as unsigned number.

8 bit unsigned binary range = 0 to $2^8 - 1$
= 0 to 255

But, 0000 0000 and 1111 1111 are reserved, so the range for biased exponent is 1 to 254

If the size of biased exponent field is n bits, Bias = $2^{(n-1)} - 1$

Hence, for 8 bit biased exponent, bias = $2^7 - 1 = 127$



Biased Exponent = Actual exponent of the binary number + Bias

If you find the actual exponent within (-126 to +127) only then that number can be represented using Single Precision Format.

Ex: 1.1011×2^{34} ; Find the Biased Exponent of the given number in IEE 754 single precision format.

Solⁿ: Check if The number is in normalized format or not.

↪ If no. then normalize it and proceed.

$1.1011 \times 2^{34} \Rightarrow$ normalized number.

Actual Exponent of
the binary number

$$\text{Bias} = 2^{(m-1)} - 1 = 2^{8-1} - 1 = 127. \quad | \quad \therefore \text{Biased Exponent} = 34 + 127 = 161 = 1010\ 0001 \quad (\text{Ans})$$

Ex: 11.1011×2^{-8} ; Find the Biased Exponent of the given number in IEE 754 single precision format.

Solⁿ: 11.1011×2^{-8}
 $= 1.11011 \times 2^{-8-1} \quad \rightarrow$
 $= 1.11011 \times 2^{-9} \quad \text{Normalized}$

$$\text{Bias} = 2^{(m-1)} - 1 = 2^{8-1} - 1 = 127. \quad | \quad \therefore \text{Biased Exponent} = -9 + 127 = 118 = 0111\ 0110 \quad (\text{Ans})$$

Decimal to Floating Point Conversion:

(i) Convert the decimal number to binary number.

(ii) Normalize the binary number.

(iii) Find the biased exponent.

(iv) Sign Bit

(v) Find the fraction.

~~E~~: Convert 50.6749 to 32 bit IEEE-754 Floating point representation :-

$$50 = 11\ 0010$$

$$\cdot 6749 \times 2 = 1.3498$$

$$\cdot 6749 = 10\ 1011\ 0011\dots$$

$$\cdot 3498 \times 2 = 0.6006$$

(i) $50.6749 = (11\ 0010 \cdot 10\ 1011\ 0011\dots)$

$$\cdot 6006 \times 2 = 1.3002$$

$$= 11\ 0010 \cdot 10\ 1011\ 0011\dots \times 2^0$$

$$\cdot 3002 \times 2 = 0.7084$$

(ii) $= 1.1001\ 0101\ 0110\ 011\dots \times 2^5$ ← actual exponent

$$\cdot 7084 \times 2 = 1.5068$$

(iii) Bias = $2^{8-1} = 2^7 = 127$

$$\cdot 5068 \times 2 = 1.1036$$

$$\therefore \text{Biased exponent} = 5 + 127 = 132 = \underline{\underline{1000\ 0100}}$$

$$\cdot 1036 \times 2 = 0.3872$$

(iv) Positive number; sign bit = 0

$$\cdot 3872 \times 2 = 0.7744$$

(v) $\underbrace{1.1001\ 0101\ 0110\ 011\dots}_{\text{Fraction}} \times 2^5$

$$\cdot 7744 \times 2 = 1.5488$$

Fraction = 1001 0101 0110 011 00000000
rest of the bits
will be filled up by 0s.

$$\cdot 5488 \times 2 = 1.0026$$

(Biased)

| Sign Bit | Exponent | Fraction |
|----------|-----------|-----------------------------|
| 0 | 1000 0100 | 1001 0101 0110 011 00000000 |

$$50.6749 = 0100\ 0010\ 0100\ 1010\ 1011\ 0011\ 0000\ 0000$$

$$= 0x424AB300$$

Double Precision (64 Bit)

| (Biased) | | |
|----------|----------|----------|
| Sign Bit | Exponent | Fraction |
| 1 | 11 | 52 |

Sign Bit = 0 \Rightarrow positive number

1 \Rightarrow negative

Exponent = It will be represented as

unsigned number.

$$11 \text{ bit unsigned binary range} = 0 \text{ to } 2^{11}-1 \\ = 0 \text{ to } 2047$$

But, 000 0000 0000 and 111 1111 1111 are reserved, so the range for biased exponent is 1 to 2046

If the size of biased exponent field is m bits, Bias = $2^{m-1}-1$

Hence, for 11 bit biased exponent, bias = $2^{10}-1 = 1023$

Convert -0.0232 to 12 bit IEEE-754 floating point representation,

where biased component is 4 bits.

Soln:

$$(i) -0.0232 = -0.0000010$$

$$(ii) -0.0000010 = 1.0 \times 2^{-6}$$

actual exponent

$$(iii) \text{Bias} = 2^{4-1} = 7$$

$$\therefore \text{Biased Exponent} = -6 + 7 = 1 = 0001$$

(iv) Sign Bit = 1.

(v) Fraction = 000 0000

| | | |
|---|------|----------|
| 1 | 0001 | 000 0000 |
|---|------|----------|

$$-0.000232 = 1000 1000 0000$$

$$= 0x880 \text{ (Ans)}$$