**ASSIGNMENT-3**

**Ans1.** Floating point number approximates a real number to support a trade off between range and precision. In IEEE754 standard, a floating point number is represented as:

Floating Point Number = Significand \* Baseexponent

Precision is one of the primary attribute of floating point number. The smallest change that can be represented in floating point representation( or the maximum number of bits present in the number) is called as precision. The number of digits in fractional part of a number determines the precision of a number. **For example**:

A number 35634.7801 can be represented as

1. 3.56347801\*104 : Here, Precision is 9
2. 3.563478\*104 : Precision is 7

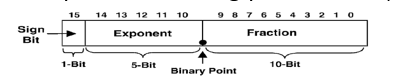
The **former number has a better precision than later** which proves that fractional part of a floating point number affects the precision of a number.

--------------------------------------------------------------------------------------------------------------------------------------

**Ans2.** As mentioned in the previous question, the floating point number is represented as:

Number = Significand\* Baseexp

For a 16-bit number, the bits are represented as:



A floating point number is represented using :

1. Sign bit
2. Exponent
3. Fractional part

Now in a **normalized form,** Significand/Mantissa(M) is chosen in such a way that 1<= M < B (base/radix).

**True value of a normalized number** is:-



Now, if the 5-bit exponent is 00000 and fraction is not equal to 0 then it is a **Subnormal number.** These are basically the floating point numbers having magnitude less than the smallest normal number.In this case, number is represented as:

Number = (-1)sign \* 2(-14) \* (0.0 + fraction)

**e.g.-** subnormal number **1000000110010101** is equal to **-2.413988113\*10-5**

If the 5 bit exponent is between (00001-11110) then it is a **Normalized number** and is represented as:

Number = (-1)sign \* 2(exponent – 15) \* (1.0 + fraction)

**e.g**- normalized number **1110000110010101** is equal to **-7.145\*102.**

The number line representation:

**🡨--------------Subnormal Numbers--------------🡪|🡨----------------Normal Numbers----------------🡪**

**↓**

**1.0 \* (base)exp\_min**

**--------------------------------------------------------------------------------------------------------------------------------------**

**Ans3.** The standard IEEE754 defines five rounding rules as described below:

1. **Round to nearest, ties to even** :- rounds to the nearest even value; even if the number falls midway. This is default for binary floating point numbers and also recommended for decimal numbers.
2. **Round to nearest, ties away from 0** :- rounds to the nearest value above( for positive numbers) and below( for negative numbers) that makes the resultant value away from zero. It is recommended for decimal floating numbers.
3. **Round towards 0** :- directed rounding towards 0.
4. **Round towards +inf**:- directed rounding towards positive infinity
5. **Round towards -inf** :- directed rounding towards negative infinity

The first two rules are round to nearest value while the other three are **Directed Roundings**.

Example of rounding of four different values according to each rule is given as:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rule** | **+1.5** | **+2.5** | **-1.5** | **-2.5** |
| Round to nearest, ties to even | +2.0 | +2.0 | -2.0 | -2.0 |
| Round to nearest, ties away from 0 | +2.0 | +3.0 | -2.0 | -3.0 |
| Round towards 0 | +1.0 | +2.0 | -1.0 | -2.0 |
| Round towards +inf | +2.0 | +3.0 | -1.0 | -2.0 |
| Round towards -inf | +1.0 | +2.0 | -2.0 | -3.0 |