In Floating point representation, we have three components

1.The Sign Bit

2.Exponent

3.Fractional Part

●

Precession is one the prime attribute of any Floating-Point Representation,

1.Does any of the above three components play a role in the defining the Precession of the number? If so which are the component or Components which play the role in defining precession and how? Explain this with example in your own words.

In floating point representation, we have the above 3 components (signed bit, exponent and fractional part). To determine sign of a number we need only one bit so this remains same across 32-bit and 64-bit representations. The exponent and fractional part vary for different representations. For 32 bit the exponent part is 8 bits and for 64 bits it is 11 bit. The fractional part also varies for 32 bit and 64 bit, it is 23 bit for 32 bit and 52 bits for 64 bits. Hence the parts that play role in defining precession are the exponent and fractional part.

2.What is Normal and Subnormal Values as per IEEE 754 standards explain this with the help of number line

Normal values are those which are represented as (-1)^s\*(1.x)\*(2^(Exponent-bias)). For normal numbers the value of exponent should not be zero.

Subnormal values are those which are represented as (-1)^s\*(0.x)\*(2^(-bias+1)). For subnormal numbers the exponent should be zero, the fractional part should be non-zero.

Subnormal

Normal

Normal

1

-1

0

3.IEEE 754 defines standards for rounding floating points numbers to a representable value. There are five methods defines by IEEE for this – Take time and understand what these five methods and explain it in your words using diagrams, illustrations of your own.

IEEE 754 defines 5 methods for rounding floating point numbers. The first two are nearest value rounding whereas the other three are directed rounding.

In nearest type of rounding the value output is chosen nearer of the possible values.

Round to nearest, ties to even: Round up to the nearest value , if the number is midway it is rounded to nearest value with lsb 0

Round to nearest, ties away from zero: Round up to the nearest value , if the number is midway it is rounded to nearest value with lsb 1

Round towards 0: directed towards 0

Round towards +infinity: directed towards +infinity

Round towards -infinity: directed towards -infinity

Example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rounding Mode | 11.5 | 12.5 | -11.5 | -12.5 |
| To nearest, ties to even | 12.0 | 12.0 | -12.0 | -12.0 |
| To nearest, ties away from zero | 12.0 | 13.0 | -12.0 | -13.0 |
| Toward 0 | 11.0 | 12.0 | -11.0 | -12.0 |
| Towards +infinty | 12.0 | 13.0 | -11.0 | -12.0 |
| Towards -infinty | 11.0 | 12.0 | -12.0 | -13.0 |