ARM Floating Point Assignment

**Problem-1**

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**

Ans:

A floating-point value in ARM is represented in IEEE754 arithmetic. Assuming a 32-bit value it has a structure as below:

Sign – bit 31

Exp – bit 23-30

Frac – bit 0 to 22

Exp is biased by 0x7F – so that small number are near 0 and large number are near 0xFF

Frac value plays role in determining the precision.

Ex. if Exp = 0x7F then number is between 1.0 to 2.0 (2.0 not included), let’s take different value of frac and see the value.

If frac = 00000000000000000000000b number is 1.0

If frac = 01000000000000000000000b number is 1.25

If frac = 10000000000000000000000b number is 1.5

If frac = 11000000000000000000000b number is 1.75

As we can see as the ones in frac moves from left to right the accuracy of value keeps increasing.

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

Ans: normal numbers: they have a regular format as discussed in Q1. Sign, exp, frac

Subnormal numbers: too small to represent in the above format, it consists, sign, frac(between 0 and 1) and exp=0. These are there to provide a smooth gap between the smallest number and 0.

s = sign, e = exp , f = frac

|  |  |  |
| --- | --- | --- |
| 0 < e < 255 | (-1)s x 2e-127 x 1.f | Normal numbers |
| e = 0, f != 0 | (-1)s x 2-126 x 0.f | Subnormal numbers |
| e = 0, f = 0 | (-1)s x 0.0 | Signed zero |
| s = 0, e=255, f = 0 | +∞ | Positive infinity |
| s = 1, e=255, f = 0 | -∞ | Negative infinity |
| s = u, e=255, f != 0 | NaN | Not a number |

**3.IEEE 754vv defines standards for rounding floating points numbers to a represent able 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.**

Ans: five types of rounding to zero

1. Round to nearest, ties to even – rounded to nearest value with even LSb (default for binary floating number)
2. Round to nearest, ties away from zero – rounded to nearest value with direction away from 0, i.e. for -ve it goes more -ve and for +ve goes more positive
3. Round toward 0 – rounded towards zero - truncation
4. Round toward +∞ - ceiling
5. Round toward -∞ - flooring

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Example** | **1.5** | **2.5** | **-1.5** | **-2.5** |
| **1. Ties to even** | 2.0 | 2.0 | -2.0 | -2.0 |
| **2. Ties away from 0** | 2.0 | 3.0 | -2.0 | -3.0 |
| **3. Rounded toward 0** | 1.0 | 2.0 | -1.0 | -2.0 |
| **4. Rounded toward +∞** | 2.0 | 3.0 | -1.0 | -2.0 |
| **5. Rounded toward -∞** | 1.0 | 2.0 | -2.0 | -3.0 |