Assignment 3 Theory Answers

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: Floating-point employs a sort of "sliding window" of precision appropriate to the scale of the number. This allows it to represent numbers from 1,000,000,000,000 to0.0000000000000001 with ease, and while maximizing precision (the number of digits) at both ends of the scale.

The *mantissa* represents the precision bits of the number. It is composed of an implicit leading bit (left of the radix point) and the fraction bits (to the right of the radix point).

All fractional bits are significant. So relative precision can be seen as

Single Precision :
2^–23 Equivalent to 23 × log(2) ≈ 23 × 0.3 ≈ 6. Thus 6 decimal digits of precision.

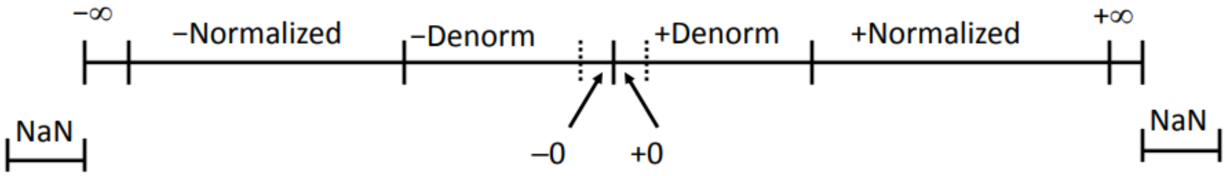
Double Precision:
2^ –52 Equivalent to 52 × log(2)≈ 52 × 0.3 ≈ 16.Thus 16 decimal digits of precision.

The range of mantissa for a 32-bit number is 1.0 to (2.0 − 2 ^–23).

There is loss of accuracy when storing irrational numbers or real numbers(like pi). Double precision format is better than single precision for storing these numbers. As double precision format allocated more bits for mantissa.

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

Normal numbers fall into general range of floating point format. Sub normal numbers fall into less than the smallest normal number.



Single Precision:

Sub-Normal numbers

Exponent is zero
Mantissa is non zero

Note: If Mantissa is zero then it represents decimal numbers +0,-0

Normal numbers

Exponent in 1-254   (0,255 value is reserved)
Mantissa is anything (including zeros)

Note: Exponent value of 255 and mantissa value of 0 are used to represent +Infinity, -Infinity.
Exponent value of 255 and mantissa value non-zero represents NaN.

Double Precision:

Sub-Normal numbers

Exponent is zero
Mantissa is non zero

Note: If Mantissa is zero then it represents decimal numbers +0,-0

Normal numbers

Exponent in 1-2046   (0,255 value is reserved)
Mantissa is anything (including zeros)

Note: Exponent value of 2047 and mantissa value of 0 are used to represent +Infinity, -Infinity.
Exponent value of 2047 and mantissa value non-zero represents NaN.

Thus, we can get the ranges for 32 bit numbers as

Largest Sub normal: The largest subnormal 0.99999988 × 2 ^−126 ≈ 1.175494e^38
0   0 0 0 0 0 0 0 0   1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Smallest Sub normal: The smallest subnormal 2 ^−149 is closer to zero ≈ 1.401298e^-45
0   0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

Due to the presence of the subnormal numbers, there are 223 numbers within the range

[0.0, 1.0 × 2 −126).
Thus we can handle underflows, and the gap between zero and smallest number is brought down.

Floating Point Range

	Sub-normalized	**Normalized**	**Approximate Decimal**
Single Precision	$\pm 2^{-149}$ to $(1-2^{-23})\times 2^{-126}$	$\pm 2^{-126}$ to $(2-2^{-23})\times 2^{127}$	$\pm \approx 10^{-44.85}$ to $\approx 10^{38.53}$
Double Precision	$\pm 2^{-1074}$ to $(1-2^{-52})\times 2^{-1022}$	$\pm 2^{-1022}$ to $(2-2^{-52})\times 2^{1023}$	$\pm \approx 10^{-323.3}$ to $\approx 10^{308.3}$

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.

IEEE 754 specifies the following rounding modes:

- **round to nearest** -- Ties round to the nearest even digit in the required position, which means If the number exactly fall midway then ties the value with even (LSB 0). This is default and common method.

  15.5 → 16 and 18.5 →18

- **round to nearest** -- Ties round away from zero which means positive numbers are rounded off to closest number above and negative numbers are rounded off to closest number below.It is the default mode for binary floating-point and the recommended default for decimal.

  15.5→ 16 and -11.5 → -12

- **round up --** regardless of value round towards +∞; negative results thus round toward zero.This is also called as ceiling.

  If 2 decimal places, 1.23→1.3 and -2.86 → -2.8

- **round down** – regardless of value round towards −∞; negative results thus round away from zero. This is also called as floor.

  If 2 decimal places, 1.23 → 1.2 and -2.86 → -2.9

- **round toward zero** –Truncation; it is similar to the common behavior of float-to-integer conversions

In decimal, −3.9 to −3 and 3.9 to 3.

In binary, 1.1101 → 1.11 and -1.001 → -1.00

References:

http://grouper.ieee.org/groups/754/

https://en.wikipedia.org/wiki/Floating-point_arithmetic

https://en.wikipedia.org/wiki/IEEE_754

https://blog.angularindepth.com/how-to-round-binary-fractions-625c8fa3a1af