Justin Hexem

AMATH 583

Homework 1

Problem 1: From my C++ file called Problem1.cpp, I have obtained precision values of 5.96046 x 10^(-8) and 1.11022 x 10^(-16) for SP and DP floats respectively.

Problem 2:

We can calculate the largest SP number by maximizing the 8-bit exponent and the 23-bit mantissa, and then setting the single sign bit to 0. To maximize the 8-bit exponent, we see that we cannot simply set all the bits to 1 because an exponent value of all ones is used for special cases. This means the maximum numerical value of the exponent must be 11111110, which after adjusting for the bias is an exponent of 127 in base 10. To maximize the mantissa, we can just set all the bits to 1, which gives us a mantissa of twenty-three 1s in a row. Adding on the hidden bit, we obtain a true mantissa of 1.11111111111111111111111. We see that we can write this number as 10 – 0.00000000000000000000001, which is 2 – 2^(-23) in base 10. Thus, the largest SP number in base 10 is (2 – 2^23) x 2^127, which is approximately 3.4028 x 10^38. In IEEE arithmetic, this becomes 01111111011111111111111111111111. Trivially we see that the smallest SP number is just the negative value of the largest number, which is -3.4028 x 10^38. In IEEE arithmetic, this becomes 11111111011111111111111111111111.

We can calculate the largest DP number by maximizing the 11-bit exponent and the 52-bit mantissa, and then setting the single sign bit to 0. To maximize the 11-bit exponent, we see that we cannot simply set all the bits to 1 because an exponent value of all ones is used for special cases. This means the maximum numerical value of the exponent must be 11111111110, which

after adjusting for the bias is an exponent of 1023 in base 10. To maximize the mantissa, we can just set all the bits to 1, which gives us a mantissa of fifty-two 1s in a row. Adding on the hidden bit, we obtain a true mantissa of 1.1111111111111111111111111111111111111111111111111111. We see that we can write this number as 10 – 0.0000000000000000000000000000000000000000000000000001, which is 2 – 2^(-52) in base 10. Thus, the largest DP number in base 10 is (2 – 2^52) x 2^1023, which is approximately 1.7977 x 10^308. In IEEE arithmetic, this becomes 0111111111101111111111111111111111111111111111111111111111111111. Trivially we see that the smallest DP number is just the negative value of the largest number, which is -1.7977 x 10^308. In IEEE arithmetic, this becomes 1111111111101111111111111111111111111111111111111111111111111111.

We can represent the smallest and largest floating-point numbers mathematically. We know the bias = 2^(k-1) – 1 where k is the number of exponent bits. Let m be the number of mantissa bits. Then the largest possible floating-point number is (2 – 2^m) * 2^bias. The smallest possible number then becomes (-1) * (2 – 2^m) * 2^bias.

Problem 3:

After running my C++ program called Problem3.cpp, multiplying 200*300*400*500 results in a value of -884901888. This effect is called overflow. We will now define what the concepts of overflow and underflow mean:

Overflow: overflow occurs when the magnitude of the value of an integer exceeds the largest possible value, in magnitude, that can be stored in 32 bits.

Underflow: underflow occurs when the magnitude of the value of an integer is smaller than the smallest possible value, in magnitude, that can be stored in 32 bits.

Overflow and underflow can also be described mathematically:

We will define the overflow boundary to be the largest possible value in magnitude that a floating-point number can be before overflow occurs. We see that this overflow value occurs when the mantissa is all ones, and the exponent is all ones except for the last bit. Let n be the total number of bits of the floating-point number, m represents the number of bits of the mantissa and let e be the number of exponent bits. Since the mantissa is all ones, then we may write the value of the mantissa as M = 2 – 2^m. The exponent of all ones except for a zero in the last bit can be written as E = e – 2 - bias where bias = 2^(e-1) – 1. Thus, we obtain the formula for the overflow boundary:

overflow boundary = M * 2^E = (2 – 2^m) * 2^(e – 2^(e-1) – 3).

This gives a mathematical definition of overflow as well. For a given number N, overflow will occur if |N| > overflow boundary.

We will define the underflow boundary to be the smallest possible value in magnitude that a floating-point number can be before underflow occurs. We see that this underflow value occurs when the mantissa is all zeros, and the exponent is all zeros except for the last bit. Let n be the total number of bits of the floating-point number, m represents the number of bits of the mantissa and let e be the number of exponent bits. Since the mantissa is all zeros, then we may write the value of the mantissa as M = 1 taking the hidden bit into account. The exponent of all zeros except for a one in the last bit can be written as E = 1 – bias where bias = 2^(e-1) – 1. Thus, we obtain the formula for the underflow boundary:

underflow boundary = M * 2^E = 1 * 2^(2 - 2^(e-1)).

This gives a mathematical definition of underflow as well. For a given number N, underflow will occur if $0 < |N| <$ underflow boundary.

Problem 4:

For SP floats, we know that there is 1 sign bit, 8 exponent bits, and 23 mantissa bits resulting in 32 total bits. We know we have a total number of possible bitstrings of 2^32. All of these are normalized IEEE floats except ones in which the 8 exponent bits are all zeros or all ones. Counting all the numbers that are not normalized, we see that there are 2 x 2^23 = 2^24 ways choose the bits for the sign and the mantissa, and two additional ways to choose whether the exponent is set to all zeros or all ones. This means there are 2 x 2^24 = 2^25 numbers that are not normalized. Thus, we can conclude that there are 2^32 – 2^25 = 4261412864 normalized SP floats.

For DP floats, we know that there is 1 sign bit, 11 exponent bits, and 52 mantissa bits resulting in 64 total bits. We know we have a total number of possible bitstrings of 2^64. All of these are normalized IEEE floats except ones in which the 11 exponent bits are all zeros or all ones. Counting all the numbers that are not normalized, we see that there are 2 x 2^52 = 2^53 ways choose the bits for the sign and the mantissa, and two additional ways to choose whether the exponent is set to all zeros or all ones. This means there are 2 x 2^53 = 2^54 numbers that are not normalized. Thus, we can conclude that there are 2^64 – 2^54 = 1.842873x 10^19 normalized DP floats.

We may now define the general formula to calculate the total number of normalized n-bit floats. Let n be the number of total bits that are used to store a given floating point value. Let m

represent the number of mantissa bits. Then we may define the following general formula to calculate the total number of floating-point values: total = 2^n – 2^(m+2). This is true because we calculate the total number of bit strings possible, which is 2^n, and we then subtract all the denormalized values and the values that are not numbers. This only occurs when the exponent is all zeros or all ones, so the number of bit strings possible where all the exponent bits are fixed as ones is 2^(m+2). The "+2" term comes from the sign bit either being positive or negative (first factor of 2) and the exponent bits either being all zeros or all ones (second factor of 2). Thus, we obtain our formula by subtracting all these special nonnumerical values by the total number of values (2^n – 2^(m+2).

6-bit floats: $s=1, k=3, n=2$ (Problem 5):

We will first calculate the sets of all possible values for the sign bit S, the exponent bits E, and the mantissa bits M. We know $E = e - \text{bias} = e_2e_1e_0 - (2^{3-1}-1) = e_2e_1e_0 - 3$ and $M = 1+f$ where $f = f_{n_1}f_{n_0}$. Trivially we see that $S_0 = s_0 = (+)$ and $S_1 = s_1 = (-)$.

e_2	e_1	e_0	2^2	2^1	2^0	4	2	1	e
0	0	0	= 0	0	0	= 0+0+0			= 0
0	0	1	= 0	0	1	= 0+0+1			= 1
0	1	0	= 0	1	0	= 0+2+0			= 2
0	1	1	= 0	1	1	= 0+2+1			= 3
1	0	0	= 1	0	0	= 4+0+0			= 4
1	0	1	= 1	0	1	= 4+0+1			= 5
1	1	0	= 1	1	0	= 4+2+0			= 6
1	1	1	= 1	1	1	= 4+2+1			= 7

Normalized Exponent Values:

$E_{001} = 1-3 = -2$
$E_{010} = 2-3 = -1$
$E_{011} = 3-3 = 0$
$E_{100} = 4-3 = 1$
$E_{101} = 5-3 = 2$
$E_{110} = 6-3 = 3$

f_{n_1}	f_{n_2}	2^{-1}	2^{-2}	$\frac{1}{2}$	$\frac{1}{4}$
0	0	= 0	0	= 0+0	= 0
0	1	= 0	1	= 0+$\frac{1}{4}$	= $\frac{1}{4}$
1	0	= 1	0	= $\frac{1}{2}$+0	= $\frac{1}{2}$
1	1	= 1	1	= $\frac{1}{2}$+$\frac{1}{4}$	= $\frac{3}{4}$

Mantissa Values:

$M_{00} = 1 + 0 = 1$
$M_{01} = 1 + \frac{1}{4} = \frac{5}{4}$
$M_{10} = 1 + \frac{1}{2} = \frac{3}{2}$
$M_{11} = 1 + \frac{3}{4} = \frac{7}{4}$

All Possible Normalized Numbers: $V = S \cdot M \cdot 2^E$

$M = M_{00}$:
$\pm M_{00} \cdot 2^{E_{001}} = \pm 1 \cdot 2^{-2} = \pm\frac{1}{4}$
$\pm M_{00} \cdot 2^{E_{010}} = \pm 1 \cdot 2^{-1} = \pm\frac{1}{2}$
$\pm M_{00} \cdot 2^{E_{011}} = \pm 1 \cdot 2^{0} = \pm 1$
$\pm M_{00} \cdot 2^{E_{100}} = \pm 1 \cdot 2^{1} = \pm 2$
$\pm M_{00} \cdot 2^{E_{101}} = \pm 1 \cdot 2^{2} = \pm 4$
$\pm M_{00} \cdot 2^{E_{110}} = \pm 1 \cdot 2^{3} = \pm 8$

$M = M_{01}$:
$\pm M_{01} \cdot 2^{E_{001}} = \pm\frac{5}{4} \cdot 2^{-2} = \pm\frac{5}{16}$
$\pm M_{01} \cdot 2^{E_{010}} = \pm\frac{5}{4} \cdot 2^{-1} = \pm\frac{5}{8}$
$\pm M_{01} \cdot 2^{E_{011}} = \pm\frac{5}{4} \cdot 2^{0} = \pm\frac{5}{4}$
$\pm M_{01} \cdot 2^{E_{100}} = \pm\frac{5}{4} \cdot 2^{1} = \pm\frac{5}{2}$
$\pm M_{01} \cdot 2^{E_{101}} = \pm\frac{5}{4} \cdot 2^{2} = \pm 5$
$\pm M_{01} \cdot 2^{E_{110}} = \pm\frac{5}{4} \cdot 2^{3} = \pm 10$

$M = M_{10}$:
$\pm M_{10} \cdot 2^{E_{001}} = \pm\frac{3}{2} \cdot 2^{-2} = \pm\frac{3}{8}$
$\pm M_{10} \cdot 2^{E_{010}} = \pm\frac{3}{2} \cdot 2^{-1} = \pm\frac{3}{4}$
$\pm M_{10} \cdot 2^{E_{011}} = \pm\frac{3}{2} \cdot 2^{0} = \pm\frac{3}{2}$
$\pm M_{10} \cdot 2^{E_{100}} = \pm\frac{3}{2} \cdot 2^{1} = \pm 3$
$\pm M_{10} \cdot 2^{E_{101}} = \pm\frac{3}{2} \cdot 2^{2} = \pm 6$
$\pm M_{10} \cdot 2^{E_{110}} = \pm\frac{3}{2} \cdot 2^{3} = \pm 12$

$M = M_{11}$:
$\pm M_{11} \cdot 2^{E_{001}} = \pm\frac{7}{4} \cdot 2^{-2} = \pm\frac{7}{16}$
$\pm M_{11} \cdot 2^{E_{010}} = \pm\frac{7}{4} \cdot 2^{-1} = \pm\frac{7}{8}$
$\pm M_{11} \cdot 2^{E_{011}} = \pm\frac{7}{4} \cdot 2^{0} = \pm\frac{7}{4}$
$\pm M_{11} \cdot 2^{E_{100}} = \pm\frac{7}{4} \cdot 2^{1} = \pm\frac{7}{2}$
$\pm M_{11} \cdot 2^{E_{101}} = \pm\frac{7}{4} \cdot 2^{2} = \pm 7$
$\pm M_{11} \cdot 2^{E_{110}} = \pm\frac{7}{4} \cdot 2^{3} = \pm 14$

All Possible Denormalized Numbers: $V = S \cdot M \cdot 2^E = S \cdot M \cdot 2^{-2}$

$M = f_{00}$: $\pm f_{00} \cdot 2^{-2} = \pm 0 \cdot \frac{1}{4} = \pm 0$

$M = f_{01}$: $\pm f_{01} \cdot 2^{-2} = \pm \frac{1}{4} \cdot \frac{1}{4} = \pm \frac{1}{16}$

$M = f_{10}$: $\pm f_{10} \cdot 2^{-2} = \pm \frac{1}{2} \cdot \frac{1}{4} = \pm \frac{1}{8}$

$M = f_{11}$: $\pm f_{11} \cdot 2^{-2} = \pm \frac{3}{4} \cdot \frac{1}{4} = \pm \frac{3}{16}$

$\therefore$ norms $= \{\pm\frac{1}{4}, \pm\frac{1}{2}, \pm 1, \pm 2, \pm 4, \pm 8, \pm\frac{5}{16}, \pm\frac{5}{8}, \pm\frac{5}{4}, \pm\frac{5}{2}, \pm 5, \pm 10,$
$\pm\frac{3}{8}, \pm\frac{3}{4}, \pm\frac{3}{2}, \pm 3, \pm 6, \pm 12, \pm\frac{7}{16}, \pm\frac{7}{8}, \pm\frac{7}{4}, \pm\frac{7}{2}, \pm 7, \pm 14\}$

denorms $= \{\pm 0, \pm\frac{1}{16}, \pm\frac{1}{8}, \pm\frac{3}{16}\}$

Now we will order these and plot them:

$-14, -12, -10, -8, -7, -6, -5, -4, -\frac{7}{2}, -3, -\frac{5}{2}, -2, -\frac{7}{4}, -\frac{5}{2}, -\frac{5}{4}, -1, -\frac{7}{8}, -\frac{3}{4}, -\frac{5}{8}, -\frac{1}{2}, -\frac{7}{16}, -\frac{3}{8}, -\frac{5}{16}, -\frac{1}{4}, -\frac{3}{16}, -\frac{1}{8}, -\frac{1}{16}, -0,$
$+0, \frac{1}{16}, \frac{1}{8}, \frac{3}{16}, \frac{1}{4}, \frac{5}{16}, \frac{3}{8}, \frac{7}{16}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1, \frac{5}{4}, \frac{3}{2}, \frac{7}{4}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4, 5, 6, 7, 8, 10, 12, 14$