COMP 122/L
Summer 2023

Floating Point using IEEE-754 (Answers)

For each of the following questions, it's recommended to follow the same steps laid out here: https://kyledewey.github.io/comp122-summer23/lecture/week_2/floating_point_interconversions.html

1.) Convert -7.5625 to binary32 (float) format.

- Step 1: Sign. Negative, so 1.
- Step 2: Integral to unsigned binary. 7 = 111
- Step 3: Fractional portion to binary.

Iteration	Calculation	>= 1.0?	Output Bit
1	0.5625 * 2 = 1.125	Yes	1
2	0.125 * 2 = 0.25	No	0
3	0.25 * 2 = 0.5	No	0
4	0.5 * 2 = 1.0	Yes	1

Overall (going from first to fourth iteration): 1001

- Step 4: Normalize via Adjusting Exponent

111.1001

Two moves left to start with a 1 (making 1.111001): exponent of 2

- Step 5: Bias the Exponent

2 + 127 = 129

- Step 6: Biased Exponent to Unsigned Binary. 129 = 1000 0001
- Step 7: Final Mantissa Bits.

(Portion to the right of the decimal point after step 4, padded with 0s to 23 places)

111 0010 0000 0000 0000 0000

- Step 8: Merge together (sign, then exponent, then mantissa):

1 (step 1) 1000 0001 (step 6) 111 0010 0000 0000 0000 0000 (step 7)
1100 0000 1111 0010 0000 0000 0000 0000

2.) Convert 0100 0001 1011 0110 0000 0000 0000 0000 to decimal.

- Step 1: Extract the sign bit. Leftmost bit is a 0.
- Step 2: Extract the exponent. 8 bits after sign bit: 1000 0011
- Step 3: Unbias the exponent. 1000 0011 = 131. 131 - 127 = 4.
- Step 4: Mantissa to decimal. Mantissa bits are the bits after the exponent, so:

011 0110 0000 0000 0000 0000

Going left to right: (0 * 2^-1) + (1 * 2^-2) + (1 * 2^-3) + (0 * 2^-4) + (1 * 2^-5) + (1 * 2^-6)

0.421875

- Step 5: Calculate the magnitude.

(1 + mantissa) * 2^(unbiased exponent)

(1 + 0.421875) * 2^(4) = 22.75

- Step 6: Factor in sign. Sign bit = 0 = positive, so 22.75