Lab7–Floating Point Practice
CPSC 2311- Fall 2021

## Introduction

The goal of this lab is to introduce/practice Floating Point conversion. We will also cover this concept in class this week along with working examples.

## Due:

Sunday, October 17, 2021, midnight.
Submit to Canvas

## Lab Instructions

While I know this would be much easier to do by hand. I have found grading to be easier when your answers are typed. Therefore, you **must** type your answers in RED on this document and submit the document through canvas as a **PDF**. Please read the entire document. Points will be deducted if you do not follow directions.

Part 1:
Watch the following videos pertaining to Floating Point conversion from decimal to binary and binary to decimal.

https://www.youtube.com/watch?v=tx-M_rqhuUA
https://www.youtube.com/watch?v=4DfXdJdaNYs

Part 2:

Following the instructions in the first video above. Convert the following floating-point number to binary.

76.48

Show your work. Also, explain what you are doing each step of the way. Your explanation does not have to be a long explanation. Only enough to let your TA know you understand what you are doing. If you do not show and explain your work, you will receive a 0 for the question.

First Convert left side (76) to binary:

76 = 38 * 2  remainder 0

38 = 19 * 2 remainder 0

19 =  9 * 2 remainder 1

9 = 4 * 2 remainder 1

4 = 2 * 2 remainder 0

2 = 1 * 2 remainder 0

1 = 0 * 2 remainder 1

76 is 1001100 in binary (we take the remainders from bottom to top)

Next, convert right side (0.48) to binary:

0.48 * 2 = 0.96

0.96 * 2 = 1.92

0.92 * 2 = 1.84

0.84 * 2 = 1.68

0.68 * 2 = 1.36

0.36 * 2 = 0.72

0.72 * 2 = 1.44

0.44 * 2 = 0.88

0.88 * 2 = 1.76

0.76 * 2 = 1.52

0.52 * 2 = 1.04

0.04 * 2 = 0.08

0.08 * 2 = 0.16

0.16 * 2 = 0.32

0.32 * 2 = 0.64

0.64 * 2 = 1.28

0.28 * 2 = 0.56

0.56 * 2 = 1.12

0.12 * 2 = 0.24

0.24 * 2 = 0.48. -> repeats after this

0.48 * 2 = 0.96

0.96 * 2 = 1.92

Take the ones place of each result from top to bottom to get binary bits.

Binary form of right-hand side is 01111010111000010100 repeating.

Put both binary sides together to get.

$1001100.\ \overline{01111010111000010100}$

Now put in scientific notation in binary (in terms of powers of 2).

Begin: 1001100. $\overline{01111010111000010100}$

Move binary point left 6 spaces (2^6): 1.001100$\overline{01111010111000010100}$

The mantissa will be everything after the binary point up to 23 bits or 00110001111010111000010

- This has been truncated because there was more than 23 bits.

Our exponent is 6 when we moved the binary point, so we need to add 6 to 127 (our bias) for the exponent section of the representation. This leaves us with 133, which we need to convert to binary. 133 in binary is 10000101, which is our exponent bits.

The number is positive, so our first sign bit is 0.

Putting together all bits in order of sign bit followed by exponent followed by mantissa we get:

0100 0010 1001 1000 1111 0101 1100 0010

This is 76.48 in binary.

Now convert the binary back to decimal, showing and explaining each step of the process. Again, your explanation does not have to be a long explanation. Only enough to let your TA know you understand what you are doing. If you do not show and explain your work, you will receive a 0 for the question.

- 0100 0010 1001 1000 1111 0101 1100 0010
    - is the binary number we are trying to convert back to decimal.
- The first bit will represent the sign.
    - It is 0 in this case so the number will be positive.
- The next 8 bits represent the exponent.
    - Those 8 bits are 1000 0101
- This leaves the mantissa which is the remaining bits or:
    - 001 1000 1111 0101 1100 0010

To convert back to decimal, we first need to find out the value of the exponent.

When the 8 bits stored in the register are converted to decimal, we get 133, however this is not the value of the exponent

We need to account for the bias and subtract 127.

133 – 127 = 6 so our exponent is 6. (e = 6)

Next, we need to find the value of our mantissa. For our mantissa, we can think of there being a leading 1. so we multiply each bit by 2 to a negative power and add them together.

Our mantissa is 001 1000 1111 0101 1100 0010 so its value is:

m = 2^-3 + 2^-4 +2^-8 + 2^-9 + 2^-10 + 2^-11 + 2^-13 + 2^-15 + 2^-16 + 2^-17 + 2^-22

m =  0.19499993324

Next we use the formula (-1)^sign bit * (1 + m) * 2^e to find the final decimal value.

(-1)^0 * (1 + 0.19499993324 ) * 2^6
1 * 1. 19499993324 * 64

Final decimal value = 76.4799957275

*This is slightly off from the original number due to the truncation of bits (more than 23 bits so some were left off in initial decimal to binary conversion)

Part 3:

Following the instructions in the above video. Convert the following floating-point numbers to binary.

-165.56

Show your work. Also, explain what you are doing each step of the way. Your explanation does not have to be a long explanation. Only enough to let your TA know you understand what you are doing. If you do not show and explain your work, you will receive a 0 for the question.

First, we need to convert the left side of the decimal point (165) to binary.

165 = 82 * 2 remainder 1
82 = 41 * 2 remainder 0
41 = 20 * 2 remainder 1
20 = 10 * 2 remainder 0
10 = 5 * 2 remainder 0
5 = 2 * 2 remainder 1
2 = 1 * 2 remainder 0
1 = 0 * 2 remainder 1

Going from bottom to top we take the remainder to get the binary form.
165 in binary form is 10100101

Now we need to convert the right-hand side (0.56).

0.56 * 2 = 1.12
0.12 * 2 = 0.24
0.24 * 2 = 0.48
0.48 * 2 = 0.96
0.96 * 2 = 1.92
0.92 * 2 = 1.84
0.84 * 2 = 1.68
0.68 * 2 = 1.36
0.36 * 2 = 0.72
0.72 * 2 = 1.44
0.44 * 2 = 0.88
0.88 * 2 = 1.76
0.76 * 2 = 1.52
0.52 * 2 = 1.04
0.04 * 2 = 0.08
0.08 * 2 = 0.16
0.16 * 2 = 0.32

0.32 * 2 = 0.64
0.64 * 2 = 1.28
0.28 * 2 = 0.56. -> repeat begins here so we can stop.

Taking the ones digit from top to bottom of each result gives us the binary form of 0.56 which is 10001111010111000010 repeating

When we put the left- and right-hand sides together we get 10100101.10001111010111000010 repeating

Next, we move the binary point to get the number in a notation that is like scientific notation but in terms of powers of 2.

We move the binary point 7 spaces left to get:
1.010010110001111010111000010

To get into IEEE floating point form we now need to complete 3 components, the sign, the exponent, and the mantissa.

Since the number is negative the sign bit will 1.

The exponent is 7 (the number of spaces we moved the binary point) plus 127 (the bias). The exponent is 134. When 134 is converted to binary we get 10000110

The mantissa is the first 23 bits after the binary point in the number above, or
m = 01001011000111101011100

Put all of these components together to get -165.56 in binary

1 10000110 01001011000111101011100 or

1100 0011 0010 0101 1000 1111 0101 1100

Now convert the binary back to decimal, showing and explaining each step of the process. Again, your explanation does not have to be a long explanation. Only enough to let your TA know you understand what you are doing. If you do not show and explain your work, you will receive a 0 for the question.

To convert back to decimal, we first need to take the 32 bit representation and break it down into its 3 components.

The sign (s) is the first bit or 1

The exponent is the following 8 bits or 10000110.

The mantissa or m is the following bits or 01001011000111101011100

To convert the exponent bits to its actual representation, we first convert it to decimal. So 10000110 is 134 but we need to subtract the bias or 127. So, the exponent or e = 7.

Next, we need to find the value of m. For our mantissa, we can think of there being a leading 1. so we multiply each bit by 2 to a negative power and add them together.

Our mantissa is 01001011000111101011100 so we can multiply as follows:
m = 2^-2 + 2^-5 + 2^-7 + 2^-8 + 2^-12 + 2^-13 + 2^-14 +2^-15+2^-17 +2^-19 + 2^-20 +2^-21
m = 0.29343748092

Finally, to get the number in decimal form, we plug each of these elements into the formula

(-1)^sign bit * (1 + m) * 2^e

Solve the formula.

(-1)^1 * (1 + 0.29343748092) * 2^7

-1 * (1.29343748092) * 128

Decimal form = -165.55999755776

*This is slightly off as the number cannot be perfectly represented in binary and therefore not perfectly converted back.

The following link is a nifty tool you can use to check your work. You should understand that sometime online tools like this one will round which could change the last one or two bits on the tool. So, if your answer has a different bit on the end that is perfectly fine. I am not saying this will be the case only letting you know this could happen.

https://evanw.github.io/float-toy/

Also, remember the discussion on the rounding that may occur when using the online converter.

Submission:

You should submit your document to Canvas. Please make sure your answers are in RED. If you do not, a substantial number of points will be deducted.