HW3–Floating Point Practice
CPSC 2310- Fall 2024

CLEMSON
School of COMPUTING

Introduction

The goal of this homework is to introduce and give you practice with IEEE 754 Floating Point conversion.

Due:

Sunday, November 17, 2024 11:59 midnight

Submission: Upload the document to Canvas

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**. Since you are going to type your answer, all answers MUST BE IN RED. Do not consolidate the pages of this document. In other words, Part 2 and Part 3 should not be on the same page. Being able to scroll to separate pages for the different questions will make grading faster. Substantial points will be deducted if you do not follow directions.

Part 1:
Watch one of the two sets of videos pertaining to Floating Point conversion from decimal to binary and binary to decimal. I found both of these sets of videos to be helpful, so feel free to watch both sets.
https://www.youtube.com/watch?v=tx-M_rqhuUA
https://www.youtube.com/watch?v=4DfXdJdaNYs

or

https://www.youtube.com/watch?v=8afbTaA-gOQ
https://www.youtube.com/watch?v=LXF-wcoeT0o

I actually like both sets of these videos and both sets are similar in execution. However, the questions below are modeled from the 1st set of videos.

Part 2: 50 points

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

67.32

For easy of grading all your work **must be typed**.
Step 1:
Convert (67) the whole part of the number to binary. It does not matter which method you use, division or subtraction, however, you must show your work.

67.32
33 * 2 + **1** = 67
16 * 2 + **1** = 33
8 * 2 + **0** = 16
4 * 2 + **0** = 8
2 * 2 + **0** = 4
1 * 2 + **0** = 2
0 * 2 + **1** = 1
Read from bottom to top:
So answer is 1 000 011

Convert (.32) the fractional part of the number to binary using the method shown in the video. You must show your work.

1. 0.32 * 2 = **0**.64
2. 0.64 * 2 = **1**.28 — 010
3. 0.28 * 2 = **0**.56

4. 0.56 * 2 = **1**.12
5. 0.12 * 2 = **0**.24 — 100
6. 0.24 * 2 = **0**.48

7. 0.48 * 2 = **0**.96
8. 0.96 * 2 = **1**.92 — 011
9. 0.92 * 2 = **1**.84

10. 0.84 * 2 = **1**.68
11. 0.68 * 2 = **1**.36 — 110
12. 0.36 * 2 = **0**.72

13. 0.72 * 2 = **1**.44
14. 0.44 * 2 = **0**.88 — 101
15. 0.88 * 2 = **1**.76

16. 0.76 * 2 = **1**.52

17. 0.52 * 2 = **1**.04 — 110
18. 0.04 * 2 = **0**.08

19. 0.08 * 2 = **0**.16
20. 0.16 * 2 = **0**.32 — 000
21. 0.32 * 2 = **0**.64

22. 0.64 * 2 = **1**.28
23. 0.28 * 2 = **0**.56 — 10

Read from top to bottom
So the answer is 010 100 011 110 101 110 000 10 for 23 bits or 010 100 011 110 101 110 00 repeating to be more precise

Combine the binary representation of 67 with the binary for .32, the fractional part of the number.
_____**1 000 011. 010 100 011 110 101 110 000 10**____

Convert the above to scientific notation: __**1 . 000 011 010 100 011 110 101 110 000 10 x 2^6**__

Now you should have the information you need for the sign bit, exponent value and the mantissa. Fill them in

S: **0**
E: **6 + 127 = 133**
M: **000 011 010 100 011 110 101 11 (truncated to be 23 bits)**

Now show the IEEE Parts in binary:
For single precision this should be 1 bit for the sign 8 bits for the exponent and 23 bits for the mantissa.
Sign bit

0

Exponent bits

1	0	0	0	0	1	0	1

Mantissa bits

0	0	0	0	1	1	0	1	0	1	0	0	0	1	1	1	1	0	1	0	1	1	1

Part 3: 50 points

Following the instructions in the above video. Convert the following binary representation to a IEEE 754 floating point number.

Let A = (0**100 0010 0**110 1001 0100 0111 1010 1110)$_2$ a decimal number in the form of 1. ……. X 2^e

Show the above in the IEEE 754 form:
S = _0_$_2$ 1bit
Exp = ___1000 0100____$_2$8 bits

M = ___110 1001 0100 0111 1010 1110_____$_2$23 bits

Now calculate the exponent:

e = __4 + 128 - 127 = 132 - 127 = 5_____

Now calculate the value of the mantissa:

m = ___$2^{-1}+2^{-2}+2^{-4}+2^{-7}+2^{-9}+2^{-13}+2^{-14}+2^{-15}+2^{-16}+2^{-18}+2^{-20}+2^{-21}+2^{-22}$ = 0.8224999905 ____

Plug the values in to the following formula:

1^s x (1 + m) x 2^e = __2^0_x (1 +0.8224999905) x 2^5 = 58.31999969 = ~58.32_____

The following link is a nifty tool you can use to check your work. You should understand that sometime online tools like this 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/

Submission:
Upload your document to canvas.