|  |  |  |
| --- | --- | --- |
| |  | | --- | | HW3–Floating Point Practice  CPSC 2310- Fall 2024 | | Logo  Description automatically generated |

# 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 / 2 = 33 R1 … 33 / 2 = 16 R1 … 16 / 2 = 8 R0 … 8 / 2 = 4 … 4 / 2 = 2 R0 … 2 / 2 = 1 R0 … 1 / 2 = 0 R1 … (most significant bit is 1. 67 = 1000011.

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

.32 x 2 = .64 WP0 … .64 x 2 = 1.28 WP1 ... .28 x 2 = .56 WP0 … .56 x 2 = 1.12 WP1 … .12 x 2 = .24 WP0 … .24 x 2 = .48 WP0 … .48 x 2 = .96 WP0 … .96 x 2 = 1.92 WP1 … .92 x 2 = 1.84 WP1 … .84 x 2 = 1.68 WP1 … .68 x 2 = 1.36 WP1 … .36 x 2 = .72 WP0 … .72 x 2 = 1.44 WP1 … .44 x 2 = .88 WP0 … .88 x 2 = 1.76 WP1 … .76 x 2 = 1.52 WP1 … .52 x 2 = 1.04 WP1 … .04 x 2 = .08 WP0 … 0.08 x 2 = .16 WP0 … .16 x 2 = .32 WP0 … .32 x 2 = .64 WP0 … .64 x 2 = 1.28 WP1 … .28 x 2 = .56 WP0 … .32 = .01010001111010111000

Combine the binary representation of 67 with the binary for .32, the fractional part of the number.

1000011.01010001111010111000

Convert the above to scientific notation: 1. 00001101010001111010111 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

M: 00001101010001111010111

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 = (0100 0010 0110 1001 0100 0111 1010 1110)2 a decimal number in the form of 1. ……. X 2e

Show the above in the IEEE 754 form:

S = 02 1bit

Exp = 100001002 8 bits

M = 11010010100011110101110 223 bits

Now calculate the exponent:

e = 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 = .82666015625

Plug the values in to the following formula:

1s x (1 + m) x 2e = 1.82666015625 x 2^5

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.