CS130 Lab Exercise #11 – Multiplication of Floating-Point Numbers

Problem #1

Create an assembly language program to run on the LEGv8 simulator found in zyBooks Section 2.24 to multiply the binary representations of the following two decimal numbers. To multiply, use "shift & add"; do not use the LEGv8 MUL instruction. The numbers are:

`11.25 × -6.5`

Recommended: The following steps:

1. Convert the decimal number **`11.25`** into binary format (e.g. **`Integer.Fraction`**) .
2. "On Paper", shift the binary point to the right as required to form a single binary value.
3. Enter this re-formatted binary number into the Preset Register for Memory address **`4000`**.
4. Use the simulator Preset Register for Memory address **`4008`** to enter the number of logical shifts it would require if you had used the command **`LSL`**.
5. Enter the sign bit as a 1-bit number (0 for positive and 1 for negative) into Memory address **`4016`**.
6. Convert the (positive) decimal number **`6.5`** into binary format (e.g. **`Integer.Fraction`**) .
7. "On Paper", shift the binary point to the right as required to form a single binary value.
8. Enter this re-formatted binary number into the Preset Register for Memory address **`4024`**.
9. Use the simulator Preset Register for Memory address **`4032`** to enter the number of logical shifts it would require if you had used the command **`LSL`**.
10. Enter the sign bit as a 1-bit number (0 for positive and 1 for negative) into Memory address **`4040`**.
11. Using "Shift & Add" as was done in an earlier lab exercise, multiply these two numbers together.
12. Store the resulting product in Memory address **`4048`**.
13. Calculate the sum of the shift values stored in Memory addresses **`4008`** and **`4032`**.
14. Store this sum in Memory address **`4056`**.
15. "On Paper", using the product from Memory address **`4048`** and the exponent (i. e. the power of base 2) stored in Memory address **`4056`**, convert this binary number into decimal.
16. Exclusive OR the sign bits that are stored in Memory Addresses **`4016`** and **`4040`**; store in Memory address **`4064`**. This bit is the sign of the product.
17. Capture your magnificent program using the simulator's "Export" feature as a text file.
18. Copy your text file for Step (13) into Notepad (or equivalent).
19. Include in this text file the results of Steps (2), (6), and 13).
20. E-mail this text file to me named in the following format: Last Name Lab11 Part 1.Txt.
21. Of course, by now you should know that Last Name is your last name as it appears on the roster and that some of you may need to include a first name after it as well.

Problem #2

1. The following hexadecimal number has been created in IEEE double-precision format. Convert it into its decimal equivalent showing each of the steps required. Do not forget about the so-called "Hidden 1" in this hexadecimal representation. Number: **`40 26 80 00 00 00 00 00`**.
2. Using a text editor such as "Notepad", document your procedure – step-by-step.
3. The following hexadecimal number has been created in IEEE double-precision format. Convert it into its decimal equivalent showing each of the steps required. Do not forget about the so-called "Hidden 1" in this hexadecimal representation. Number: **`40 19 00 00 00 00 00 00`**.
4. Using a text editor such as "Notepad", document your procedure – step-by-step.
5. Multiply the decimal equivalents of these two numbers together using – say – the 'Calculator' provided with *Windows* or *Penjee*. A hand-held or phone-app calculator will do as well.
6. Convert this decimal number into IEEE double-precision format.
7. Using a text editor such as "Notepad", document your procedure – step-by-step.
8. E-mail this text file to me named in the following format: Last Name Lab11 Part 2.Txt.