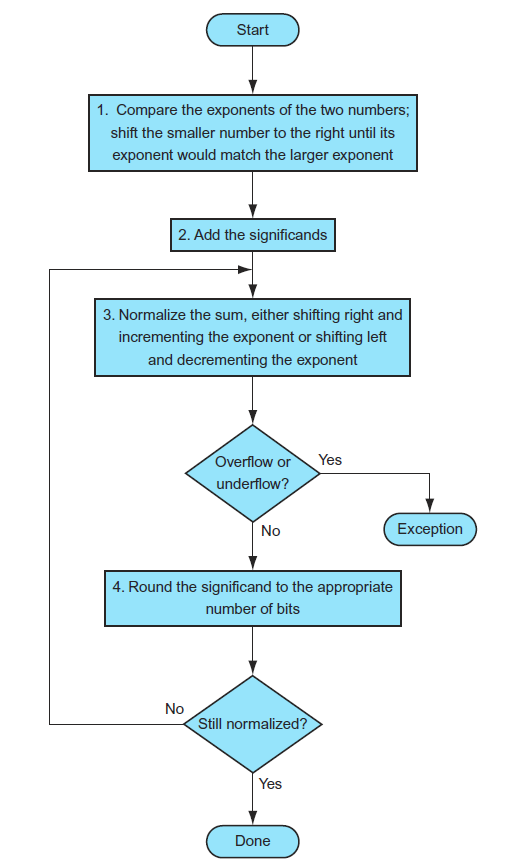
**HW#9 (CSC390) Due:** 04/07/2018 by 11:30PM

**Q1.** Let’s consider a 10-bit binary floating-point number system, in which -like IEEE 754 number system- 1 sign bit, 4 Exponent bits and 5 fraction bits are used to represent a floating-point number. Two such floating point numbers, as shown on figure 2, are added following the flow chart of Figure 1. The circuit that implements the flow chart is shown in figure 2.

Note that the Exponent = actual exponent + Bias, where Bias = 2n-1 – 1; n = number of bits in Exponent and like the IEEE 754 system, the decimal equivalent of the binary floating point number is represented by: (-1)Sx(1+Fraction)x2(Exponent – Bias) (**ref:** Section 3.5 of your Text)

1. Convert the two binary floating-point numbers to their corresponding decimal equivalent numbers.
2. Follow the flow chart of figure 1 and tabulate the values you would get after every step of the iterative process. Indicate the selection line values for the three Multiplexers (as shown in step – 1) to choose the proper values of the Exponent and Fractions. Verify that the adder circuit produces correct results.

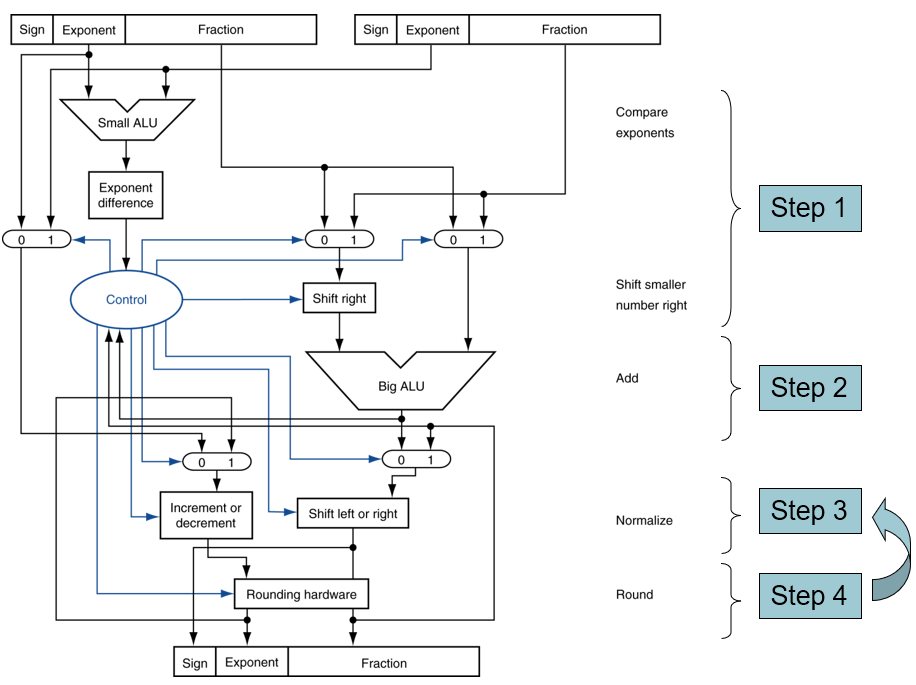


**Figure 1**

**10-bit Binary floating point numbers:**

**0 0 1 1 0 1 1 0 1 0**

**1 1 1 0 1 1 1 1 0 1**



**Figure 2**

**Q2.** Now use the flow chart of Figure 3 to perform the floating-point multiplication of the Binary floating-point numbers given in Question 1 (figure 2). Clearly show the values you would get in every step of the iterative process.

Q2. (Continue)

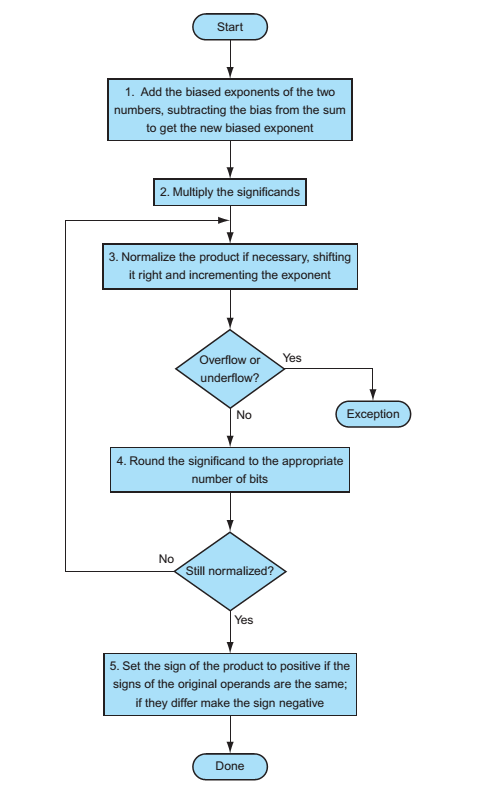


Figure 3

**Q3.** Write a MIPS assembly program to calculate the area of a rectangle using the floating point data (length and width of the rectangle) supplied from the keyboard. Your program should also display the results on the MIPS editor.

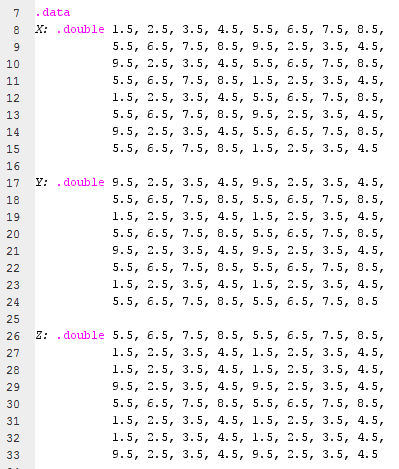
Hints: use “syscall” function discussed in our last class and example codes posted on the blackboard.

**Q4**. The following code (as shown in fig. 7, which is also posted on the blackboard) performs the matrix operation, X=X+Y\*Z, and display the results on the MIPS editor as shown in Figure 8. Please note that X, Y, and Z are square matrixes of size **4x4** and the elements of the matrixes are double precision floating point number.

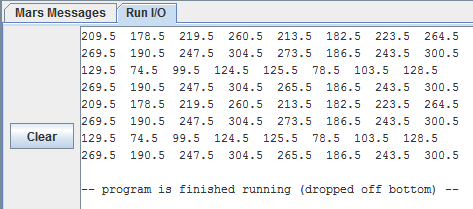
1. Now, modify the code in such a way that it would perform:

X = X+Y\*Z; where X, Y, and Z are **8x8 matrixes** (as shown in fig 4) and each element is represented by double precision number. Your program should also display the result as shown in figure 5.

(Note that results are verified using Matlab)



**Figure 4**

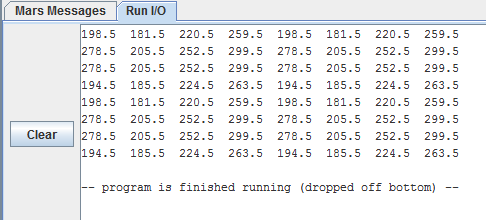


**Figure 5**

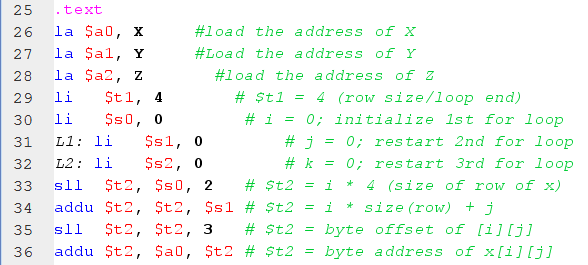
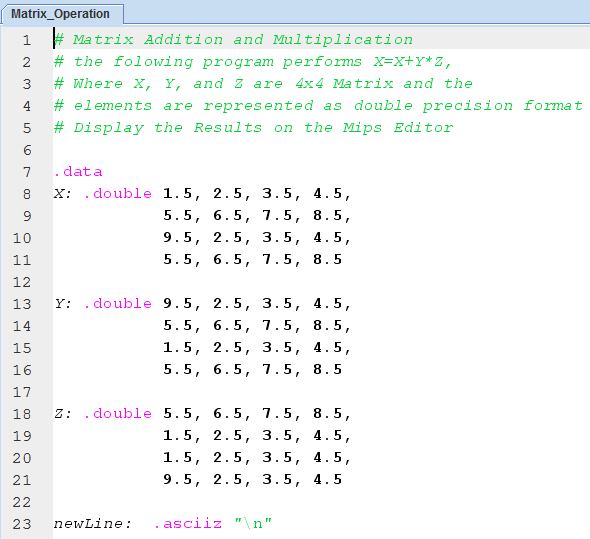
1. Also, modify the code in such a way that it would perform:

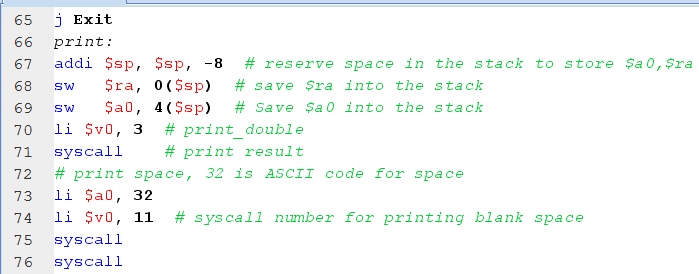
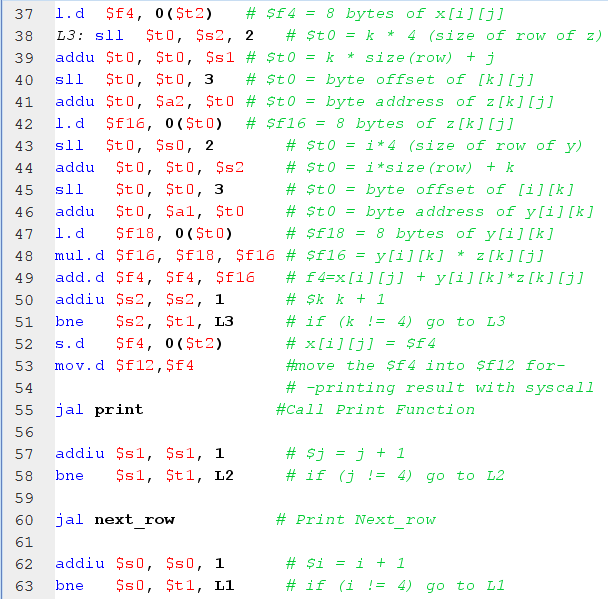
**Z = X\*Y-Z**; where X, Y, and Z are 8x8 matrixes (as shown in fig 4) and each element is represented by double precision number. Your program should also display the result as shown in figure 6.

(Note that this part would be more challenging than the first part. Please go through the pages 215-217 of your book. Pay attention to the Indexed operation and how the multiplication is done. You would need to use an extra instruction sub.d in addition to the add.d instruction. Add.d would be used to find each element of X\*Y and sub.d would be used for X\*Y-Z. Recall the Matrix multiplication operation. You may also need to use an extra floating point register to hold the X\*Y value temporarily)



**Figure 6**





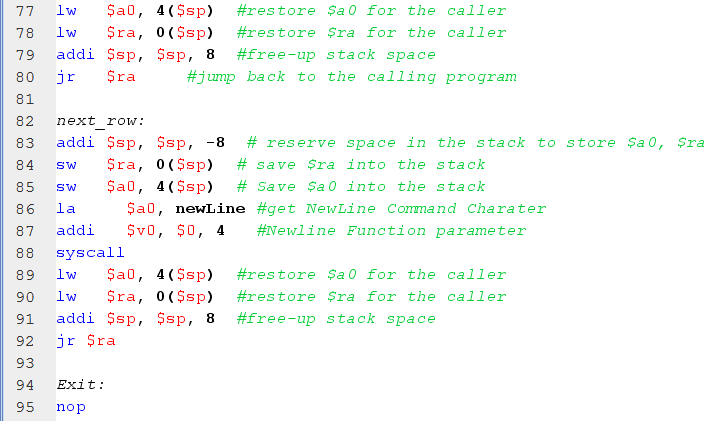


Figure 7

Results:

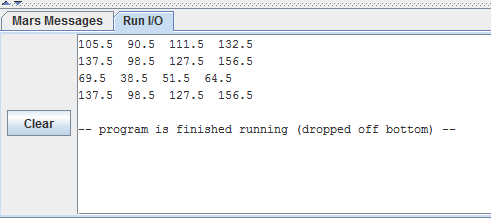


Figure 8