

It is required to write an assembly program that performs floating point operations on single precision floating point numbers following the IEEE 754 standard without using any of the floating point arithmetic instructions, rounding the result to the nearest even as follows:

1. Write a procedure *fadd* that performs floating point addition of two single precision floating point numbers.
2. Write a procedure *fsub* that performs floating point subtraction of two single precision floating point numbers.
3. Write a procedure *fmul* that performs floating point multiplication of two single precision floating point numbers.
4. Ask the user to enter two single precision floating point numbers and read them.
5. Perform the three arithmetic operations on the two entered numbers: addition, subtraction, and multiplication, and print the results obtained by your implemented procedures and the results obtained using the existing floating point instructions.

Sample Execution of the Program:

Enter first floating point number: 5.67  
Enter second floating point number: -2.3  
Result of fadd = 3.3700001  
Result of add.s = 3.3700001  
Result of fsub = 7.9700003  
Result of sub.s = 7.9700003  
Result of fmul = -13.041  
Result of mul.s = -13.041

Enter first floating point number: 1.297e15  
Enter second floating point number: 7.014e13  
Result of fadd = 1.36714003E15  
Result of add.s = 1.36714003E15  
Result of fsub = 1.22685996E15  
Result of sub.s = 1.22685996E15  
Result of fmul = 9.097158E28  
Result of mul.s = 9.097158E28