**Lab 11:**

**Floating-Point Arithmetic**

**Name (Print):** \_\_\_\_\_\_\_\_\_REZA SHISHEIE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ID\_\_\_2708062\_\_\_  
  
**Give brief answers to the following questions. You can edit this document and insert your answers after each question.**

**Due dates:**

**MW – Wed, May 2, beginning of class  
TTH – Tue, May 1, beginning of class**

**Circle one: MW or TTH**   
**Note**: All problems refer to the 16-bit, **modified**, IEEE 754, base-2 floating point format that we discussed in class unless otherwise stated. The decimal-to-float converters that are online are **not** the same.

1. (2 pts) Calculate the decimal equivalent of the base-2 floating-point number . (Remember that calculators are not permitted on exams.)  
     
   **Ans.**  
   11010.01 x 2^(-3) = .1101001 x 2^(-3+5) = .1101001 x 2^(2)

b = e + 128 🡪 b = 2+128 = 130

f = 1101001 = (1/2+1/4+1/16+1/128 )

number = (1/2+1/4+1/16+1/128 ) x 2^(2) = 2+1+1/4+1/32 = 3.25 + 0.03125 = 3.28125

1. (2 pts) Using the 16-bit floating point format that we discussed in class, how many different ways are there to represent the number 0? (Hint: Part of the answer is on Slide 6.)  
     
   **Ans.**one way : f has to be 0 to make the decimal number 0
2. (2 pts) Convert the modified IEEE 754 floating point representation 0x84EC to decimal. Show your work. (Hint: See slides for conversion algorithms.)  
     
   **Ans.**   
   0x84EC = 1000010011101100 = 10000100 1 1101100

f = 1101100

s = 1

b = 10000100 🡪 e = b-128 = 132 – 128 = 4

decimal = (1/2 + 1/4 + 1/16 + 1/32 ) x 2^(4) = 2+1+1/4+1/8 = 3+.25+.125 = 3.375

1. (2 pts) Convert the decimal 0.0058 to the modified IEEE 754 floating point representation. Show your work.  
     
   **Ans.**0.0058 = (1/64+1/32)/8 = 1/512 + 1/256 = (2^(-9) + 2^(-8))

(2^(-9) + 2^(-8))2^(0)

0.0058 = 0.00000001011111 = 0. 1011111x2^(-7)

f = 1011111

e = -7—> b = 01111001

s = 0

01111001 0 1011111

1. (2 pts) Use lab11.asm to calculate the sum of  
     
   0x8340 = 4  
   0x83C0 = −4  
     
   Convert the sum to decimal. Is the answer correct?  
     
   **Ans.**sum = 0x83AB = 10000011 1 0101000

b = 10000011 = 131

s = 1

f = 0.0101000 = 1\*2^(-2) + 1\*2^(-4) = ¼ + 1/16 = 5/16 = .3125

x = (-1)^1\*0.3125 \*2^(3) = -2.5

The answer is NOT correct since -4+4 =0 nut the result from micro is not correct.

1. (2 pts) Convert each of the following to the modified IEEE 754 floating point representation.  
     
   100352  
   100353  
     
   Are the answers correct? Explain.  
     
   **Ans.**100352 = 1 1000 1000 000 0000 = 0.110001\*2^(17)

e = 17 —> b = 128+17 = 144

X1 = 1001 0001 0110 0010 = 0x9162

100353 = 1 1000 1000 000 0001 = 0.1 1000 1000 0000 0001\*2^(17)

e = 17 —> b = 128+17 = 144

X2 = 1001 0001 0110 0010 = 0x9162

10352 is correct but 10353 is not since we truncated data

1. (10 pts) Note that the FloatMultiply routine in lab11.asm does not return the two sample multiplications in normalized form. Create a copy of lab11.asm and rename it to lab11\_normalized.asm. Create a new project called lab11\_normalized using this file. Add a subroutine called Normalize which normalizes the products of the two samples in the code. Demonstrate the correct normalized products in a Watch window and explain the subroutine.  
     
   **Student Name** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   **Instructor/TA signature** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_