# CSE340: Computer Architecture

## Assignment 3

Chapter 3

**Note**: <u>For the questions below, if necessary, Consider 5 digits while performing the decimal to binary conversion of the floating portion.</u>

### Question 01

Convert the following IEEE-754 number into a decimal number where 6 bits are allocated for the exponent field in the representation.

0x ABB9609

### Question 02

Perform the arithmetic operations using the Floating point format

a. 50.7869 + 79.83 - 29.58
b. 64.2486 * 49.1832

### Question 03

Subtract -4.0210 from 28.4810 using IEEE-754 single-precision floating-point representation. Check if the result has overflow or underflow or none.

Note: Consider 5 digits while performing the decimal to binary conversion of the floating portion.

### Question 04

a. Why is a bias added to the actual exponent in the IEEE 754 floating-point representation, and how does this affect the encoding of both positive and negative exponents?

b. How does optimized multiplication improve efficiency and performance compared to traditional long multiplication, especially in terms of speed and computational complexity?

## Question 05

| | Instruction name |
|---|---|
| 1 | mul |
| 2 | mulh |
| 3 | mulhu |
| 4 | mulhsu |

Explore these instructions from the slide first.
You **must** answer the following questions for the explanation of each of the instructions mentioned above:
(i) What do they do/ How do they work?
(ii) Explain their syntax.
(iii) Write an example for each of the instructions.

## Question 06

fle.d x5, f3, f5
(i) What x5 is an integer register?
(ii) f3 and f5 both are single precision numbers. True or False. If false, write the correct statement.
(iii) after running the instruction you see that x5 register contains 0. What can you comment on?

## Question 07

Write the necessary RISC-V code to compare two floating point registers f1 and f2. If f1 and f2 are equal jump to a label called "**jumpEqual**" else jump to another label called "**jumpNotEqual**"
Note: both the numbers stored in f1 and f2 registers are double-precision floating point numbers.

## Question 08

Write the necessary RISC-V code to compare two floating point registers f1 and f2. If f1 is greater than f2, jump to a label called “**jumpGreater**” else jump to another label called “**jumpNotGreater**” and divide the variable c with 16 and store that answer in c.

Note: both the numbers stored in f1 and f2 registers are double-precision floating point numbers and variable c is stored in x19.