Assignment 3

Chapter 3 (Arithmetic for Computers)

**Question 01**

Multiplicand = 13 (Decimal) and multiplier = 17 (Decimal)

Complete the multiplication following the optimized multiplication algorithm.

**Question 02**

Convert the following precise number into a decimal number where in the representation 7 bits are allocated for the exponent field.

0x ABB9609

**Question 03**

Convert the number 70.78955 into a floating point format where total bit length would be 36 and 9 bits will be allocated for the biased exponent field.

**Question 04**Perform the arithmetic operations using Floating point format

1. 50.7869 + 79.83 - 29.58
2. 64.2486 \* 49.1832

**Question 05**

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 10 decimal digits while converting from decimal to binary for the following questions.

**Question 06**

Multiply 0.0001012x2-85 and 10.12x2-90 using an 18 bit IEEE-754 floating-point representation where the size of the fraction field is 12 bits. Check if the result has overflow or underflow.

Note: Consider 10 decimal digits while converting from decimal to binary for the following questions.

**Question 07**

Suppose four single precision floating point numbers **g**,**h**,**i** and **j** are stored in memory. The memory locations are directly stored in register **X10**, **X11**, **X12**, **X13**. Write necessary code to store the result of   
 **( g + h ) - ( i + j )**

in the memory address that is stored in **X14.**

**Question 08**

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.