COE 301 – Computer Organization

Assignment 4: Floating-Point Representation and Arithmetic

1. What is the decimal value of the following single-precision floating-point numbers?

a) **`1010 1101 0001 0100 0000 0000 0000 0000`** (binary)
b) **`0100 0110 1100 1000 0000 0000 0000 0000`** (binary)

2. Show the IEEE 754 binary representation for: `-75.4` in …

a) Single Precision
b) Double precision

3. *x* = **`1100 0110 1101 1000 0000 0000 0000 0000`** (binary) and
y = **`0011 1110 1110 0000 0000 0000 0000 0000`** (binary)
are single-precision floating-point numbers. Perform the following operations showing all work:

a) $x + y$
b) $x * y$

4. *x* = **`0101 1111 1011 1110 0100 0000 0000 0000`** (in binary) and
y = **`0011 1111 1111 1000 0000 0000 0000 0000`** (in binary) and
z = **`1101 1111 1011 1110 0100 0000 0000 0000`** (in binary)
represent single precision IEEE 754 floating-point numbers. Perform the following operations showing all work:

a) $x + y$
b) Result of (**a**) + z
c) Why is the result of (**b**) counterintuitive?

5. IA-32 offers an 80-bit extended precision option with a 1 bit sign, 16-bit exponent, and 63-bit fraction (64-bit significand including the implied 1 before the binary point). Assume that extended precision is similar to single and double precision.

a) What is the bias in the exponent?

b) What is the range (in absolute value) of normalized numbers that can be represented by the extended precision option?

6. Using the refined division hardware, show the **unsigned** division of:

Dividend = **`11011001`** (binary) `by` Divisor = **`00001010`** (binary)

The result of the division should be stored in the Remainder and Quotient registers. Eight iterations are required. Show your steps.

7. Using the refined **signed** multiplication algorithm, show the multiplication of:

Multiplicand = **`00101101`** `by` Multiplier = **`11010110 (signed)`**

The result of the multiplication should be a 16 bit signed number in HI and LO registers. Eight iterations are required because there are 8 bits in the multiplier. Show the steps.