COE608: Computer Organization and Architecture
Mid Term Examination

Student Name: **Student #:**

i) Answer all the questions. **Total Marks: 35**
ii) Total time allowed is 50 minutes. Any notes or books are not allowed.
iii) Estimated time for each question is equivalent to the marks assigned to it.
iv) All the questions are not of equal difficulty. Read the questions carefully.

1. (a) Write the VHDL entity of a 3-bit wide Full Adder.

MARKS: 4

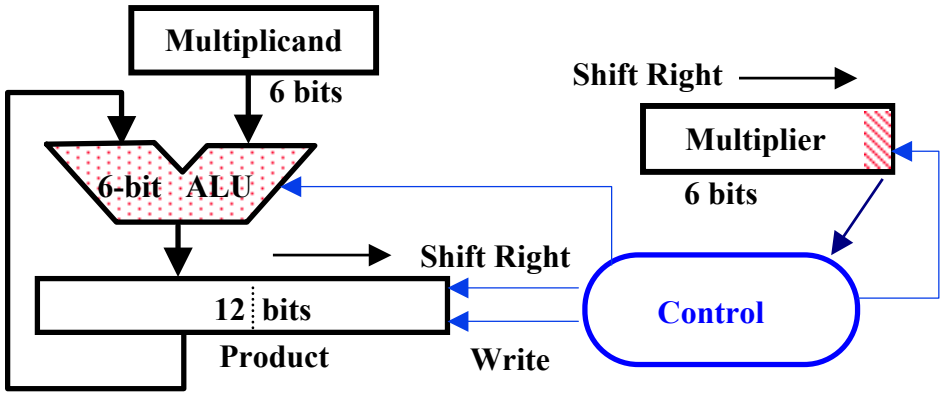
1. (b) What information about a VHDL design is represented in the Architecture?

MARKS: 2

2. Determine IEEE754 floating point representations, single precision and double precision of $(21.85)_{10}$. Show your complete work.

MARKS: 8 (4+4)

3. The block diagram of a 6-bit multiplier with optimal size ALU (adder) and registers is given below.



Assume that the Multiplicand and Multiplier registers are loaded with 6-bit numbers for multiplication and the product register is cleared initially as shown below. Two 6-bit unsigned binary numbers are to be multiplied using shift or add/shift operations. During this process, show the contents of all these registers that change. Determine the contents of these registers after an add/shift or just shift operation and fill up the following table after each operation.

MARKS: 10

Operation Shift and/or Add	Product	Multiplicand	Multiplier
Initial Values	0 0 0 0 0 0 0 0 0 0 0 0	0 1 1 0 1 1	0 1 0 1 0 1

4. (a) Consider the following C code:

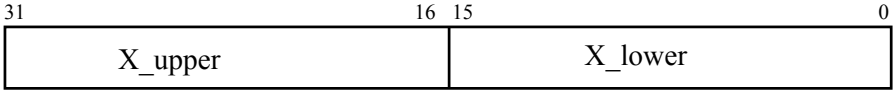
X[13] = m - n

i) For a load-store type 32-bit MIPS CPU, write an optimal assembly language instruction sequence for the above C code. Assume that variables **m** and **n** are held in registers $s1 and $s2 respectively while address of X[10] is in register $s0. You are free to use any of the temporary registers $t0-$t7.

ii) What is the size of the assembly code developed in part (a)?

MARKS: 6 (4+2)

4. (b) If X is a 32-bit memory address, divided into two 16-bit values X_upper & X_lower as shown below.



Typically follwoing instruction sequence is used to load the data at Address X into a register ($s0).

```
lui     $t0, X_upper
ori     $t0, $t0, X_lower
lw      $s0,  0($t0)
```

Consider the following alternate code that is more efficient.

```
lui     $t0, X_upper
lw      $s0,  X_lower($t0)
```

Is this code accurate? **YES** or **NO**

If YES, justify your answer.

If NO, identify the mistake and suggest any corrections.

MARKS: 5