EG 211 Computer Architecture Assignment – 1

Guidelines:

- To be done individually or in groups of 2 (not more than 2)
- You can choose to do the design in any language you wish, C, C++, python, Verilog etc
- What to submit: Submit your report (in pdf format) which contains roll numbers and names of teh students in the group, some explanation of the code, the program you chose, snapshots of result etc. Upload codes separately.
- The submission will be followed by a viva/demo
- When you submit the code, rename the filename of your code to <roll_numbers>_filename.< >
- All codes will run through a plagiarism check. Files found similar will get a 0 for the assignment. Repeat offence will attract Grade penalty on the overall grade
- Submit by Oct 4, 2023, 11:59pm on LMS under Assignment 1
- Marks: 20
- The following two tasks can be done in parallel by the two members in the team

1. Assembly programming
    a. The example program template.asm is provided to you. It takes 3 inputs
        i. Number of integers to sort
        ii. Starting address of inputs to store
        iii. Starting address of outputs (sorted integers) to store
        iv. Integers to sort

        Try this out first.
    b. Write a MIPS assembly program for any one of the following programs: (10 marks)
        - Sorting algorithm
        - Matrix Multiplication
        - Convolution
        - Simple encryption and decryption

    You could modify lines 49-54 in the template.asm to implement your own program. Test this program in MARS.

    c. Desired output: If you chose a sorting algorithm, the output should be sorted integers (either in ascending or descending order) stored in a certain address range. Take snapshots of the output from MARS.
2. Assembler
    - Write an assembler (in any language) that can read in the MIPS assembly program of question – 1. (10 marks)

- The assembler should read in the assembly code and generate a machine code. Verify the machine code generated by your assembler with the one generated by MARS

Hints:

- You are free to use any programming language
- Instruction and Data memory can be declared as arrays of limited size (need not be 2^{32})
- Memory can be word addressable (each memory location can store 32bits unlike byte-addressable memory used in MIPS where each memory location can store 8 bits)