UG-CS-4 Sem: Jan-Jun 2021

CS4401 (Computer Architecture): Assignment 1

Submission deadline: **18-Jan-2021 17:30 hours** Weightage: 10%

[Your answers MUST be free from any kind of compile/runtime/simulation ERRORS]

(Roll no. and Course code to be **MANDATORILY** mentioned in every required FILE; upload single ZIP/RAR file)

Q.No. 1. Use of Logisim simulator to construct your 32-bit Arithmetic Logic Unit (ALU). Your ALU must support following twos complement arithmetic and logic operations —

- Arithmetic operations [Range extension, Negation, Addition, Subtraction, Multiplication (using Booth's algorithm), Division (using restoring approach)]
- Logic operations [AND, OR, NOT]

Generate 3-bit operation code to produce 8 combinations for 8 operations (NEG, ADD, SUB, MUL, DIV, AND, OR, NOT), which is to be used by control logic to control different components.

Input: two 32-bit binary numbers

Output: One 32/64-bit binary number, Flag bits (Carry, Overflow, Sign, Zero)

You must submit - (i) working design of ALU in the form of CIRC file (keep your roll number & course code in filename); (ii) test vector (in ASCII text file) for testing your ALU (keep your roll number & course code in filename).

Q.No. 2. Use of Logisim simulator to construct your 32-bit Floating-point Unit (FPU). Your FPU must support following arithmetic operations on real numbers represented using IEEE 754 FP standard —

- Addition, Subtraction, Multiplication, Division, where appropriate rounding logic is to be incorporated to produce most accurate result

Generate 2-bit operation code to produce 4 combinations for 4 operations (ADDF, SUBF, MULF, DIVF), which is to be used by control logic to control different components of FPU.

Input: two 32-bit binary real numbers (as per IEEE 754 FP standard)

Output: One 32-bit binary real number (as per IEEE 754 FP standard) to represent most accurate result, Flag bits (Overflow, Sign, Zero)

You must submit - (i) working design of FPU in the form of CIRC file (keep your roll number & course code in filename); (ii) test vector (in ASCII text file) for testing your FPU (keep your roll number & course code in filename).