American University In Cairo
School of Sciences & Engineering
Department of Computer Science & Engineering
CSCE493005 - Selected Topics in CSCE (Fall 2022)

Project Document

In this project, you are going to develop a C++ program that simulates a very simple microprocessor incrementally adding features to it as the term progresses. This project will be implemented in ~3 phases.

Phase 1

In this phase, you will develop the microprocessor without concurrent execution and networking capabilities. The microprocessor, which we will call Simple Integer Machine (SIM), doesn't have registers but has two memory banks. The first is an instruction memory. It contains 1024 memory locations each capable of holding one instruction. The second, a data memory, also contains 1024 memory locations each capable of holding one integer value.

The input will be a text file containing the instructions. An instruction starts with a 3 character instruction word at the beginning, like ADD, followed by a single space, then a list of space-separated operands. The number of operands depends on the instruction. There exists a newline character after each instruction. You are supposed to read this text file at the beginning of the simulation and convert each instruction to your own representation that you store in the instruction memory. This representation should use C++ classes. After storing all instructions, you should start executing them.

Important Note: You are expected to output to the console a debug message each time you read an instruction or execute an instruction to be able to assess your code. Here is an example of such a message: "The SIM just added memory location 2 with value 10 & memory location 3 with value 20 and stored it in memory location 4 now with value 30". The more detailed your messages are the better.

The SIM can execute the following instructions:

Instruction	Format	Effect	Remarks & Addressing Modes
ADD	ADD in1 in2 out1	out1 := in1 + in2	in1, in2, & out1 are addresses.
NEG	NEG in1 out1	out1 := - in1	in1 & out1 are addresses.
MUL	MUL in1 in2 out1	out1 := in1 * in2	in1, in2, & out1 are addresses.
JPA	JPA a1	goto address a1	a1 must be a valid address within the instruction memory
JP0	JP0 in1 a1	if (in1 == 0) goto a1	in is an address & a1 is a valid address within the instruction memory
ASI	ASI in1 out1	out1 := in1	in1 is an integer value & out1 is an address.
LOE	LOE in1 in2 out1	if (in1 <= in2) then out :=1 else out1 := 0	in1, in2, & out1 are addresses.
HLT	HLT	stop the SIM	

You are required to submit the following items in this phase:

1. Class diagram showing the structure of the classes in your simulator
2. The C++ source code
3. The results of the sample tests that we will provide you.
4. Sample tests that run successfully on your simulator and their results only in the case that your program isn't able to pass our tests.

Grading

The grade of each phase is divided as follows:
75 % on the code implementation passing our tests **including the debug messages**
15 % on the class diagrams
10 % on the code quality and comments

Number of team members

Your team should consist of only 2 members.