COL216 : Assignment 4

II Semester 2019-2020

Release Date: 31 January, 2020
Submission deadline : 11:55 pm, 14 February, 2020

General Instructions

1. You will be using Vivado for simulation and synthesis.
2. The assignment should be done in groups of two. Only one member from each group should submit the assignment on Moodle.
3. Both group members should understand the problem and contribute equally to the solution.
4. You will be awarded marks according to your design and the test cases you developed to evaluate the design. Extensive testing is expected as a part of the Assignment.

Problem

In this assignment, you will simulate and synthesize the processor designed in Assignment 3. Instead of reading the machine code from file/test-bench, machine code should be read from Memory. Make appropriate modifications to your design.

1. Create a Memory Module (see Memory Generation below)
2. Initialize machine code in Memory (see Memory Generation below)
3. Read one instruction from Memory at a time, execute it, and proceed to reading the next instruction from Memory. Halt the execution when you encounter an instruction with all ZEROes.
4. *Load* and *Store* instructions may require 2 cycles: 1 cycle to read the instruction from Memory and 1 cycle to load/store the data from/to Memory.

Memory Generation

In Vivado, go to the Project Manager window on the left side of the screen. Open *IP catalog*. You can also open IP Catalog through a drop down menu after clicking on Window in the main menu on the top of the screen. IP (Intellectual Property) refers to pre-designed modules available for use. The IP catalog lists various IPs category-wise. Select the category **"**Memories & Storage Elements**"** and sub-category **"**RAMs & ROMs**"**. Now choose **"**BlockRAM Memory Generator**"**. Specify width and depth of Memory. Initialize the BlockRAM with the machine code of instructions using the *coe* file.

In Other Options tab:

- Tick Load Init file and set the path of the *coe* file. Create a *coe* file and use its file path in Load Init file.
- Tick "Fill Remaining memory locations" with "0" to initialize uninitialized memory locations with zero.

Here is a link showing the *coe* file syntax:
https://www.xilinx.com/support/documentation/sw_manuals/xilinx11/cgn_r_coe_file_syntax.htm

Outputs

1. Print the lower 16 bits of the output register in the last instruction on the 7-segment display.

2. Along with the result, print the number of cycles taken to execute the program on the 7-segment display. You can use a switch to display one result at a time, i.e., the 7-segment LED displays the register contents when the switch is 0, and the cycle count otherwise.

Complete the following tasks.

1. Simulate the VHDL model in Vivado for correctness after providing test input programs.

The program memory is initialized with the specified *coe* file. All that needs to be done for simulation now is to generate the clock signal. This could be generated by using wait statements in the test bench or by directly using "Force clock" and "Force constant" in the simulator. Observe the waveforms of various signals as the program execution progresses. You can observe the data in Registers and address, data signals to/from Memory.

2. After you are satisfied with the simulation, synthesize the design. Use the clock available on the board.
3. Download the generated bit file into the FPGA and run the processor on the board. Further instructions will be posted on the FPGA demonstration.

Physical labs will be conducted for this assignment in the DHD lab.