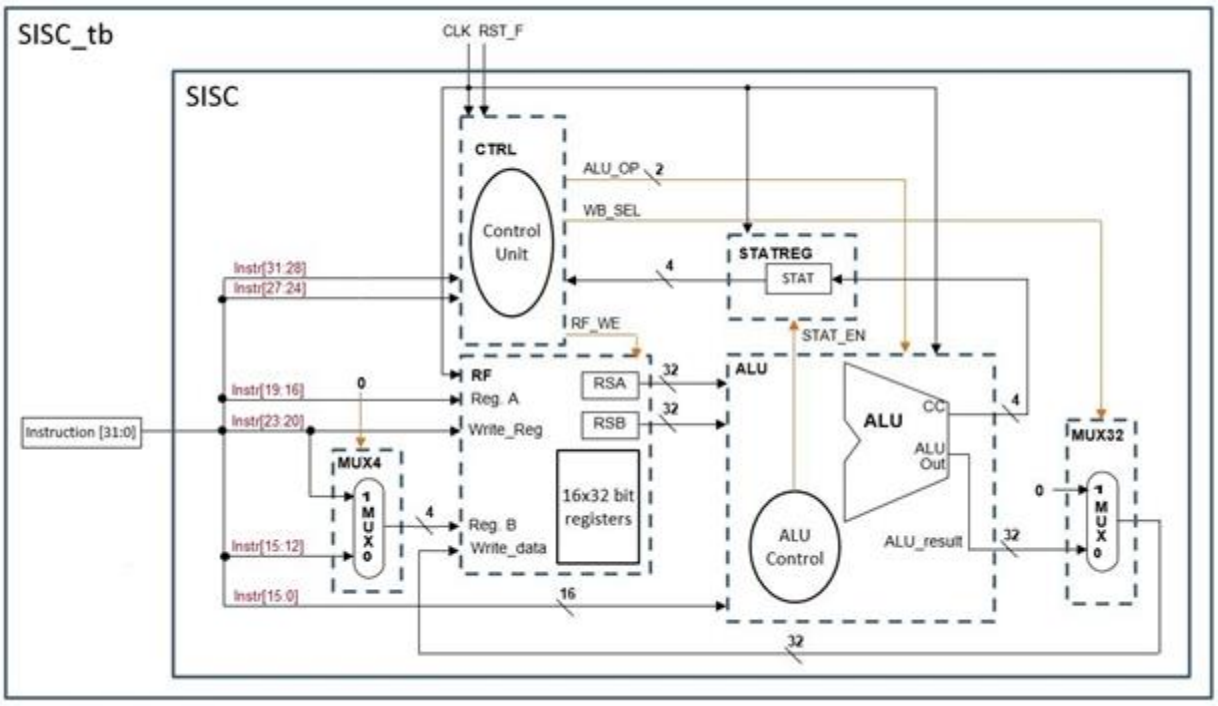
ECE:3350 Spring 2020 -
Computer Architecture and Organization
Simple Instruction Set Computer (SISC) Project

Part 1: Due Monday, March 2, at 8:30 am, dropbox closes March 9 at 8:30 am.

In this first part of the project you will implement the modules and datapath depicted below:



You are given:

- All of alu.v, rf.v, statreg.v, mux4.v, and mux32.v, which you may not modify.
- Detailed descriptions of each module's function and control lines, in block comment form, at the top of every Verilog file provided to you.
- A mostly empty ctrl.v, which contains descriptions for the inputs and outputs, as well as some parameters, reset processes, and other miscellaneous data.
- A mostly empty sisc.v, which contains descriptions for the inputs and outputs.

You are required to:

- Finish the implementation of ctrl.v as a finite state machine, so that the control lines shown above in orange operate correctly as described by the given files. Note that you are only to implement the control for the R type instructions and the ADD immediate instruction. Note that the state machine is already defined. You just have to generate the alu_op, wb_sel and rf_we control signal outputs.

- Finish the implementation of sisc.v which contains a module 'sisc' that instantiates each of the six modules shown above and connects their control and data signals. On the diagram above, any signal that goes from one module to another has to be defined as a wire of the appropriate size.
- You are given a Verilog file named 'sisc_tb_p1.v' which contains a module 'sisc_tb' that instantiates the sisc module, drives the CLK and RST_F signals as described below, and generates instructions that test each of the supported instructions for Part 1. In this file there are comments that denote what the contents of the register should be at a specific time.
- Once you have completed the sisc.v and ctrl.v files, run ModelSim. Don't forget to change the working directory to where you have located your files. Compile the files and them simulate the part 1 design. For verification of your design, you can either use the $monitor statement and them save the simulation transcript or take two screen shots of the Wave window to show the register contents as noted in the sisc_tb_p1.v file.
- Compress your project directory, including the 'work' directory and the transcript file or screenshots into a .zip file named 'project_p1.zip' and submit it to your "Verilog Project – Part 1" dropbox on ICON.

Part 1 Notes:

- Download the sisc_p1_files.zip file from Canvas and unzip the files to a sisc_p1_files directory.
- Read the documentation included with each file carefully before beginning to connect them.
- In the RF module, the outputs of the register file R[A] and R[B] are latched in the RSA and RSB registers on the positive edge of the clock.
- In the ALU module, the output of the ALU is latched in the ALUOut register on the positive edge of the clock.
- The sisc module should take in three inputs (IR, CLK, and RST_F) and have no outputs.
- The sisc_tb module should have no inputs or outputs.
- In the testbench file, the clock process runs at the rate of 1 cycle/10 time units. Since the default time unit in ModelSim is 1 ns, this gives us a 10 ns clock period. The RST_F reset line is active low and is held at 0 for 20 ns or 2 clock cycles before being set to logic 1. Generation of the first instruction is delayed until time = 35 ns so that it appears at IR on the second clock cycle after the reset line has been set to logic 1. All further instructions are delayed by 50 ns per instruction to allow all five cycles to execute in the control unit. At the time the instruction changes, the cntrl.v module should enter the DECODE state.
- The instructions your design should support by the end of Part 1 are NOP, ADD, ADD IMM, SUB, NOT, OR, AND, XOR, ROTR, ROTL, SHFR, and SHFL. These instructions all have an opcode of 4'b1000 (hex 8). All instructions have an MM field value of 4'b0000 except for the ADDI (add immediate) instruction which has an MM field value of 4'b1000. The HLT instruction is already implemented. Always include a HLT command at the end of your instructions to end execution.
- You may use any $monitor(…) statements you wish to test your design, but in the sisc.v file that you submit, please monitor the following signals: IR, R1 through R5, RD_SEL, ALU_OP, WB_SEL, RF_WE and STAT.

- To monitor a signal within an instantiated module, such as R1 and R2, use the dot operator. If you have instantiated module rf with the name my_rf, you can access R1 with the following syntax in the $monitor signal list, i.e. `my_rf.ram_array[1]`.

Submission Overview:

- Your .zip file should be named "Project_p1.zip" and contain:
  - A short description of your project implementation, and the names of your project team (no more than two per team).
  - Alu.v, mux4.v, mux32.v, rf.v, statreg.v, and sisc_tb_p1.v, exactly as they were provided.
  - Ctrl.v and sisc.v, completed by your team.
    - Note that sisc.v should contain the $monitor statement described above!
  - The 'work' directory.
  - Your simulation transcript

    OR
  - ModelSim screen captures.
- For a team of two, you only need to submit the project .zip file once, not for each team member. However, it is mandatory that the names of both team members be listed on the description file so both get credit for the work.

Part 1 Grading Rubric:

Implementation description	10 pts
Ctrl.v Implementation	15 pts
Sisc_tb.v Implementation	15 pts
Correct Execution of All Instructions	30 pts
Total	**70 pts**