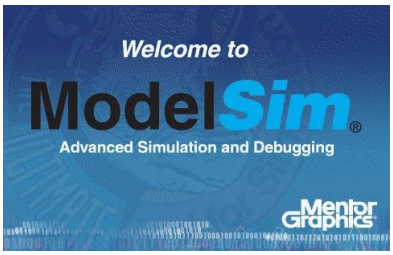

CSE 331 Computer Organization

Final Project – Single cycle MIPS with Structural Verilog

Due date: January 2, Wednesday 23:59 (Moodle)

You will design a MIPS processor supporting not only the previously implemented R-type instructions but also **`andi`**, **`ori`**, **`addiu`**, **`lw`**, **`sw`**, **`beq`** and **`j`** instructions as defined in the **MIPS Green Sheet**. You will need two more memory blocks other than the register block: data memory and instruction memory. Only the next PC calculating block, register block and memory blocks in your design will be behavioral but other than that, all your design must be structural.

Your MIPS module will execute the instructions inside the instruction memory one by one. The module named nextPC will compute the next PC value and write it to PC register at the clock edge. You have to use behavioral Verilog for that purpose.

Only structural Verilog is allowed, dataflow and behavioral Verilog is not allowed except for the nextPC, register and memory modules. This means you cannot use `assign`, `if-else`, `always`, `?:` and etc.

Use hierarchy in your project. For instance, you must design two memory modules, one control unit module and etc.

You will not support other instructions than mentioned above.

You have to simulate all instructions by yourself via Modelsim and put the results in your report as well as to your zip folder including all your project files to submit to Moodle.

You will initialize the instruction memory using readmemh and write down the resultant contents of the data memory and the register using writememh in your testbench Verilog.

You should write a report (20%) including:

1. Your schematic designs for all modules.
2. Your Verilog modules and their description.
3. Modelsim Simulation results.
4. If not compiling or partial working the explanation of which parts work which parts do not.

You will submit your report, your full project as a zip file to Moodle.

Rules:

1. Behavioral or Dataflow Verilog are not allowed.
2. Not compiling or not simulating solutions can at most get 25pts.
3. You have to use Quartus II tool referred in Moodle.
4. Each day of late submission will get 25 point loss.
5. The name of your top module should be **`mips32_single_cycle`**.
6. Design on top of your previous design with least modifications in your previous design.

Hint: Start with drawing schematic on paper for each module. Otherwisse you easily get lost. Do not hesitate to write 32 lines of logic expressions for each bit of one or two 32-bit numbers whenever required.

Honor code: It is not a group project. Do not take any code from Internet. Any cheating means at least -100 for both sides. Do not share your codes and design to any one in any circumstance. Be honest and uncorrupt not to win but because it is RIGHT!

Bonus: Any one who can show his/her project on an actual FPGA board will get 30pts Bonus. The solution is yours to design!