CAD7 Datapath Assembly

Assignment

To assemble and simulate the pipelined datapath of your microprocessor.

Description

The datapath of your microprocessor stores, accesses, and computes the operands of your instructions, and interfaces with the external modules. A good starting point for this exercise would to be to look at the data flow for each of your instructions and make sure that you have all the components necessary to do the desired computations. By now, you have designed most of the components needed to achieve the desired functionality. You may need to design tristate drivers and/or muxes. You must decide whether to locate modules on or off of the datapath. For example, you may decide to put the IR and PSR on the datapath. If so, you should design those and incorporate them into your datapath in cad7. Both the IR and PSR are excellent candidates for inclusion in the scan-chain. You must also commit to either a tristate bus or multiplexor-based design; this decision together with the placement of modules on the datapath dictates the number of buses on your datapath. A natural next step is to assemble all of the modules together (in both the schematic and the layout) and simulate for functional and timing correctness. This requires that you present correct control signals to all of the control points in the simulation (these are input signals, since you do not yet have the control logic).

Procedure

With the datapath complete, you should extract the load capacitance on each of your module outputs. (Other delays in your circuit should not change.) Replace the arbitrary load capacitance values used in earlier circuit simulations with these more accurate values, and resimulate your modules in Eldo. Change the rise and fall times in the schematic delay elements to reflect this accurate delay. Don't forget to perform delay simulation on any new blocks that you create for cad7.

Simulate the entire datapath in Modelsim (with the right delays) for functional verification. You will have to force control variables and register file select signals by yourself since you do not at present have the decoder. You will have a lot of control signals to force. You may find it simpler to begin coding a simple controller in verilog to provide the input stimulus, particularly if you have someone in your group that has written behavioral/functional verilog in the past.

Run at least one simulation each for all instructions (both Reg-Reg and Reg-Immediate). Pay careful attention to the write-back portion of your "execute" pipeline stage.

List and describe all of the control signals for the datapath. Your description should include functional specifics for your individual components. You may include a truth-table if you like. This will be needed for the next CAD assignment and will help the grader understand your verification.

Requirements

Please turn in the following in the directory **cad7**:

- Schematics of the entire datapath with all the delay modules added.
- Layout of the datapath. You do not need to wire vdd and ground pins together yet. You may find it more convenient to do this during final chip integration. However, all data and control signal wiring should be completed between datapath components.
- Modelsim simulation of the entire datapath with delays added. You should demonstrate a few examples of each arithmetic/logic operation, the ability to change program flow (branch, jump and jal) as well as important cases such as system reset. You should also include a minimal simulation of the test mode operation (scan chain), as well.
- List of all the control signals with descriptions. You may include a truth-table.
- Documentation is important, so make sure you comment your report well. Important considerations that went into the design, and extra features, if any, should be documented.
- Create a simple table in your README showing pre-CAD7 output timings for regfile, ALU, shifter and PC. Then include the output timings after real parasitics have been extracted from your complete datapath. Please reference Eldo config files AND schematics from your previous designs that were used to

attain these results as well as any new Eldo config files for blocks that were designed as part of CAD7.

- Reference Eldo config files and associated schematics that were used in obtaining delays for any new blocks that were designed in cad7. Note that it's not adequate to simulate simple CMOS transmission gate tristates alone - you must simulate with the proper driving gate to get true delays.
- Report files (DRC, LVS and PEX) for the datapath should be placed in the cad7 directory.