

Solve this problem by modifying a copy of <http://www.ece.lsu.edu/ee3755/2001f/hw07.html> which can also be found in `/home/classes/ee3755/com/v/hw07.v`. See <http://www.ece.lsu.edu/ee3755/proc.html> for instructions on running the simulator. Alternate instructions can be found in Lesson 7 of the ModelSim Tutorial, linked to the references web page, <http://www.ece.lsu.edu/ee3755/ref.html>. The links are clickable when this assignment is viewed with Acrobat Reader. The ModelSim tutorial and other documentation can also be accessed from the Help menu on the ModelSim GUI (started by the command `vsim -gui`).

Copy the homework template, `/home/classes/ee3755/com/v/hw07.v`, into a subdirectory named `hw` in your class account.

Documentation on MIPS instructions can be found in Appendix A of the text (and elsewhere) and in <http://www.ece.lsu.edu/ee4720/mips32v2.pdf>.

Problem 1: Some of the `cpu` modules shown in class instantiate an `alu` module which is used to perform arithmetic operations, logical operations, and shifts. The `cpu` in the solution template however instantiates an `alu` that can do arithmetic and logical operations, but that cannot do shifts. Instead shifts are performed by a separate `shifter` module that the `cpu` module also instantiates. Modify the `cpu` module so that it can execute the `sll`, `srl`, `sllv`, and `srlv` instructions using the `shifter` module. The `alu` module is properly “connected” but `shifter` is not.

See the solution template for further information.

Problem 2: The `cpu` module in the solution template, like the modules presented in class, spends one cycle fetching each instruction. The processor does nothing else while fetching. This is a waste of time because while executing one instruction the processor has the address of the next instruction (in `npc`). Modify `cpu` so that while one instruction is executing the next instruction is fetched. This change should reduce the CPI by one.

See the solution template for further information.