EC413 Computer Organization
Lab 9 – Pipelined CPU – Forwarding

OVERVIEW:

The purpose of Lab 9 is to learn more about the workings of a pipelined CPU. For this lab, you are given a pipelined CPU with a data forwarding unit. Unfortunately, someone has deleted the logic from the forwarding unit, as well as where it connects in the CPU. Your job is to reconnect the forwarding unit and implement the data forwarding logic. Figure 4.54 in the Hennessy and Patterson textbook will be helpful in understanding how this unit works, and where it connects. A detailed diagram of the CPU used in this lab is also included on the next page. Your final CPU should be able to deal with data forwarding for both R-type and I-type instructions and should work for reading from any register in either the Rs or Rt position. It should also be able to deal with hazards at either the ex/mem pipeline register or mem/wb pipeline register. In other words, your forwarding unit should work in all cases.

NB: To accomplish this, some wires from Lab 8 will need to be rerouted or routed to additional modules. Some of the inputs to your Lab 8 Hazard Detection Unit may shift to the forwarding unit. You may add muxes and wires in addition to the forwarding unit.

Below is a small MIPS program. Start by implementing a testbench for this program and deter- mine where the hazards are occurring (note that not all instructions need forwarding. You must determine which ones do). Then, work through which logic your forwarding unit needs to deal with each hazard. Your final design should be able to run this code with no hazards.

```
addi    R1, 423
addi    R2, 92
addi    R3, 13
addi    R4, 146
addi    R5, 5
add     R5, R1, R4
slt     R6, R3, R5
lw      R4, 0(R0)
sub     R7, R4, R6
sw      R7, 4(R0)
add     R8, R7, R2
```

TASKS:

1. Modify your hazard detection unit from lab 8 to handle only load-use hazards.
2. Add 1 ahead forwarding. [1 ahead Rt]
3. Add 2 ahead forwarding.
4. Make sure to account for branch instructions with input dependencies.
5. Add arbitration logic for deciding between 1 & 2 ahead forwarding.

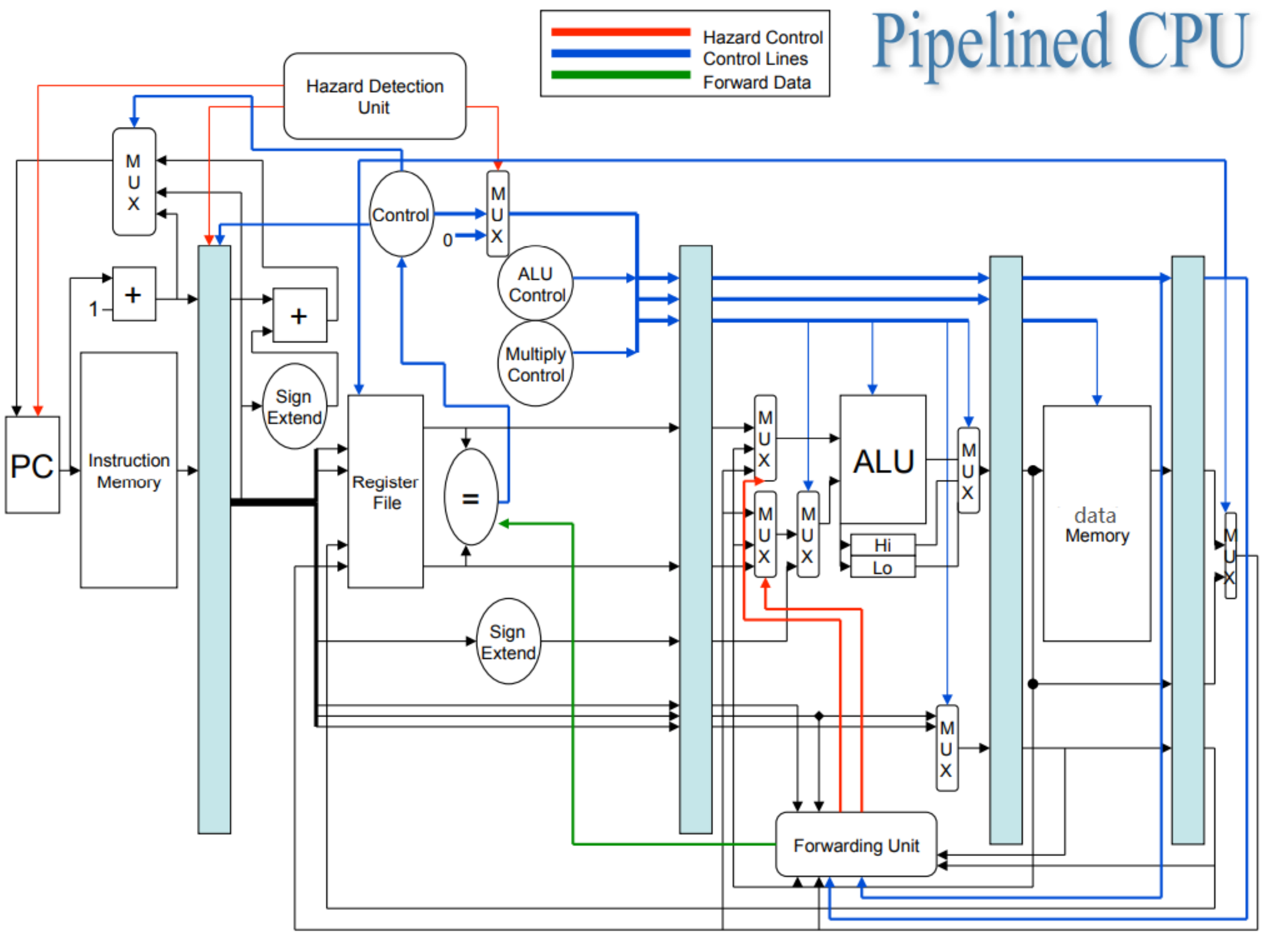
DELIVERABLES:

Submission on blackboard:

1. All your .v files
2. A brief report with:
   - A description of what you did for each of the above tasks.
   - Waveforms of how you tested your design, along with a brief explanation.

A demo to one of the TAs:

Be sure to submit all your code and report on blackboard before the demo
Both group members must be present during the demo
ALL GRADING WILL BE DONE DURING THE DEMO



GRADING GUIDELINES:

1. Ensure all code and brief report have been submitted to blackboard [**20 points**]
2. Demonstrate 1 ahead forwarding and 2 ahead forwarding has been added [**20 points**]
3. Demonstrate 1-ahead and 2-ahead forwarding arbitration logic has been added [**30 points**]
4. Demonstrate branch instructions with input dependencies dealt with [**20 points**]
5. Demonstrate overall understanding through short questions asked [**10 points**]