Organization of Digital Computer Lab
EECS 112L/CSE 132L

Final Project - Pipeline ARM

prepared by: Pooria M.Yaghini

University of California, Irvine

February, 27, 2017

Due on March 19^{th}, 11pm.

1 Goal

As the final project for this course, you will speed up your single-cycle ARM processor by making its architecture pipeline.

1.1 Project Description

In this project, you will turn your previously designed single-cycle ARM processor into a five-stage pipeline. Then, synthesize it and measure the performance improvement pipeline architecture has provided. You need to re-design some part of your processor in multiple steps.

The very first change is to modify the datapath to reflect the pipeline stage using appropriate set of registers. The abstract micro-architecture of the corresponding processor is illustrated in Figure 1.

In the second step, you have to address the structural and data hazards. The data hazard is resolved through data forwarding, which a sample corresponding architecture is depicted in the Figure 2.

The Third step is to implement stalling feature, as it is shown for LDR instruction in Figure 3. This helps deal with control hazard, though it lowers the performance. Note the following point when implementing the stall.

- Turn off the enables on the earlier pipeline stages.
- Insert control and data values corresponding to a NOP into the "'downstream"' pipeline register.
- When the stall is over, re-enable the pipeline registers

1.2 Assignment Deliverables

Your submission should include the following:

- Block diagram of your processor design.
- SystemVerilog Testbench capable verifying the correct functionality of the processor using couple of simple and complicated test ARM program.

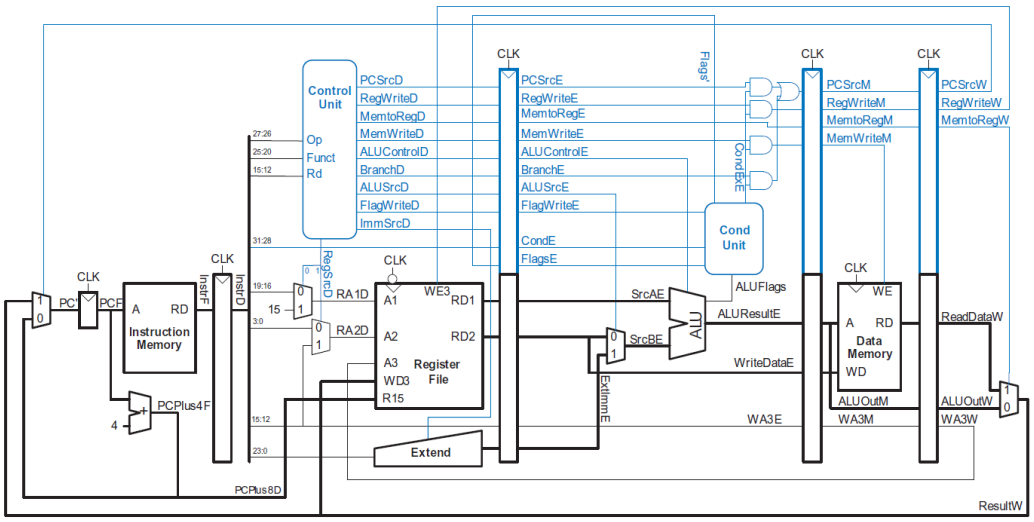

Figure 7.47 Pipelined processor with control

Figure 1: Pipeline ARM processor architecture

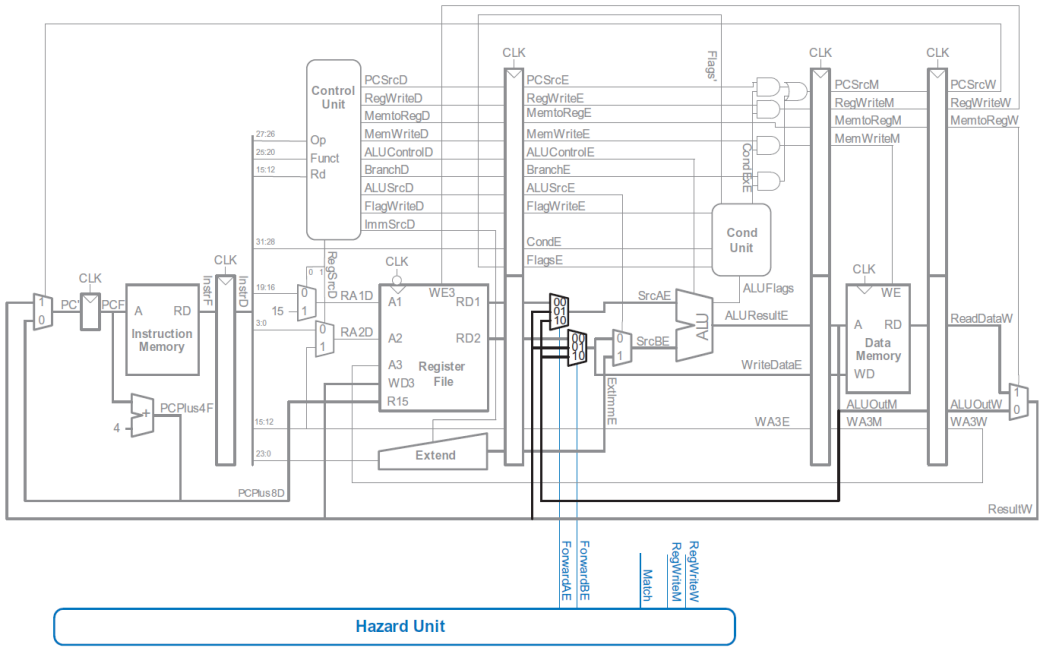

Figure 7.51 Pipelined processor with forwarding to solve hazards

Figure 2: Datapath with forwarding capability

- A comprehensive report of the Design architecture. In the report, include the information on how you have make sure the processor works fine, and if you have found any bugs in your design. Also, include the steps to run your testbench.
- In your report, describe after running the sample program, what was the expected/real output of the program.
- Synthesize your processor and report the Area, Power, and the maximum frequency your processor operates. Also, make sure you synthesis does not have any error, you can find it out by looking

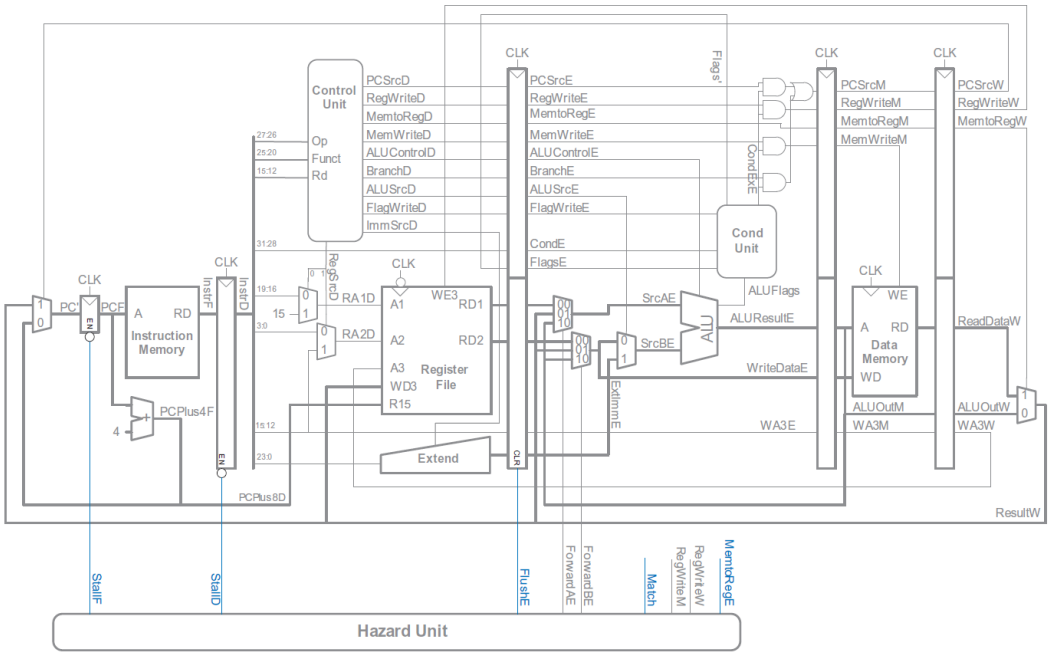

Figure 7.54 **Pipelined processor with stalls to solve** LDR **data hazard**

Figure 3: Datapath with stalling capability for LDR instruction

into analysis report in log directory and elaboration report (arm.elab.rpt) in reports directory.

IMPORTANT: only **ONE** submission per group is required and the group leader should be the submitter.

Note1: Compress all your files in "zip" or "tar" format and then submit the compressed file.

Note2: The compressed file should include **design, sim, verif, doc, syn** directories and the corresponding files inside each directory. Your report's tex (if prepared with Latex), drawing, and pdf files are expected to be included in **doc** directory. The synthesis files and reports should be located in **syn** directory. **ONLY** include the text files (*.rpt) and remove the other unnecessary files when submitting.

Note3: Remember to include your Group ID and the name and student ID of each group member in the report. When in doubt about your group ID, refer to the spreadsheet uploaded on the course website.

Note4: Assuming that the parent directory that you are working under that is named `final_project`, the following command will compress it for you:

```
$  tar cvf final_project.tar final_project
```