

Implementing the MIPS Processor with Pipelined Microarchitecture

Dates:

Section 1: Mon, 12 Apr, 8:30-12:20 in EA-Z04
Section 2: Wed, 7 Apr, 13:30-17:20 in EA-Z04
Section 3: Tue, 13 Apr, 13:30-17:20 in EA-Z04
Section 4: Mon, 12 Apr, 13:30-17:20 in EA-Z04
Section 5: Fri, 9 Apr, 8:30-12:20 in EA-Z04
Section 6: Fri, 9 Apr, 13:30-17:20 in EA-Z04

TAs:

Section 1: Zülal Bingöl, Ergün Batuhan Kaynak
Section 2: Zülal Bingöl, Kenan Çağrı Hırlak
Section 3: Alper Şahıstan, Kenan Çağrı Hırlak
Section 4: Ege Berkay Gülcan, Ergün Batuhan Kaynak
Section 5: Hüseyin Eren Çalık, Yusuf Dalva
Section 6: Ziya Erkoç, Yusuf Dalva

TA name (x No of labs) email address:

Alper Şahıstan (x1): alper.sahistan@bilkent.edu.tr
Ege Berkay Gülcan (x1): berkay.gulcan@bilkent.edu.tr
Ergün Batuhan Kaynak (x2): batuhan.kaynak@bilkent.edu.tr
Hüseyin Eren Çalık (x1): eren.calik@bilkent.edu.tr
Kenan Çağrı Hırlak (x2): cagri.hirlak@bilkent.edu.tr
Yusuf Dalva (x2): yusuf.dalva@bilkent.edu.tr
Ziya Erkoç (x1): ziya.erkoc@bilkent.edu.tr
Zülal Bingöl (x2): zulal.bingol@bilkent.edu.tr

Lab Attendance and Rotation Policy Reminder and Time for the Next Lab:

All labs will be done online and attendance is mandatory. You have to attend the online labs and submit the lab work by following the instructions of your TA. Before submission your TA will do a brief interview with you to understand your knowledge of the work you plan to submit. **Note that preliminary works are submitted before all labs and are graded only if you attend the lab and submit your lab work.**

Next Lab - Lab6 Days: Due to the lab rotation policy the first Lab6 session will be with Section 2, 3, 5 and 6 on the week starting with April 26, Monday. The lab days for Sections 1 & 4 are on the following week starting with May 3, Monday.

Purpose: In this lab you will use an online digital design tool, EDA Playground (<https://www.edaplayground.com/>), to implement and test the Pipelined MIPS Processor and implement a hazard unit to handle possible hazards. You will be provided a skeleton SystemVerilog code for the Pipelined MIPS processor and fill the necessary parts to make it work. To do this, you must first implement the pipeline registers, then implement a datapath to make the necessary connections between the pipelines, control signals and other hardware elements, e.g, forwarding muxes, adders, etc., and finally implement the hazard unit using the logic equations discussed in class. Implementing the Pipelined MIPS Processor will require you to fill-in some SystemVerilog modules in

the HDL model of the processor which is provided in the same folder as this one in Moodle. To test and prove correctness, you will simulate the microarchitecture on EDAPlayground.

Summary

Part 1 (50 points): Preliminary Work: Design Report with Pipeline hazards evaluation and preparing test programs in MIPS.

Part 2 (50 Points): Lab Work: Implementation and simulation of the MIPS-lite pipelined processor.

[REMINDER!] In this lab, we assume SystemVerilog knowledge, since you all took CS223 – Digital Design. If you are not familiar with these, please get used to them as soon as possible

Note try, study, and aim to complete lab part at home before coming to the online lab.

DUE DATE OF PART 1 (PRELIMINARY WORK): SAME FOR ALL SECTIONS

No late submission will be accepted.

- a. Please upload your programs of preliminary work to Moodle by 13:30 pm on Wednesday, April 7.
- b. Please note that the Moodle submission closes sharp at 13:30 pm and no late submissions will be accepted. You can make resubmissions before the system closes, so do not wait the last moment. Submit your work earlier and change your submitted work if necessary. Note that only the last submission will be graded.
- c. Do not send your work by email attachment they will not be processed. They have to be in the Moodle system to be processed.
- d. You will need to submit a report file (in pdf).
- e. For the report, use filename **StudentID_FirstName_LastName_SecNo_PRELIM_LabNo_Report.pdf** Only a PDF (pdf file) is accepted. Any other form of submission receives 0 (zero).

DUE DATE PART 2 (LAB WORK): (different for each section) YOUR LAB DAY

- a. You have to demonstrate (using Zoom) your lab work to your TA for grading. Do this by **12:00** in the morning lab and by **17:00** in the afternoon lab. Your TAs may give further instructions on this. If you wait idly and show your work last minute, your work may not be graded.
- b. At the conclusion of the demo for getting your grade, you will **upload your Lab Work Part 2** to the Moodle Assignment, for similarity testing by MOSS. See Part 6 below for details.
- c. Aim to finish all of your lab work before coming to the lab, but make sure that you upload your work after making sure that your work is analyzed by your TA and/or you are given the permission by your TA to upload.
- d. We use zoom to track your lab attendance.

If we suspect that there is cheating we will send the work with the names of the students to the university disciplinary committee.

Provide following five lines at the top of your submission for preliminary and lab work (make sure that you include the course no. CS224, important for ABET documentation).

CS224

Lab No.

Section No.

Your Full Name

Bilkent ID

Make sure that you identify what is what at the beginning of each program by using proper comments.

Part 1. Preliminary Work (50 points)

Important Notes

1. **Assume that all Inputs are Correct.** In all lab works, if it is not explicitly stated, you may assume that program inputs are correct.

Part 1.

In this part you will prepare a Design Report in PDF format named as:

StudentID_StudentFirstName_StudentLastName_SecNo_LabNo_REPORT.pdf

At the end of this lab, you will have implemented the pipelined MIPS architecture that can be seen in the file that is provided as **PipelineDatapath.PNG** (Notice that **there is an early branch prediction in this pipeline**. Hence, the branch resolution is done in the Decode stage.). Note also that there is no jump instruction implemented as well. Your Design Report should contain the following items:

- a) Cover page, with university name, department name, and course name and number at the top, "Design Report", Lab # (e.g. 5), Section #, and your name and ID# in the middle, and the date of your lab at the bottom.
- b) **[10 points]** The list of all hazards that can occur in this pipeline. For each hazard, give its type (data or control), its specific name ("compute-use" "load-use", "load-store", "branch" etc.), the pipeline stages that are affected.
- c) **[10 points]** For each hazard, give the solution (forwarding, stalling, flushing, combination of these), and explanation of what, when, how
- d) **[10 points]** Give the logic equations for each signal output by the hazard unit, as a function of the input signals that come to the hazard unit. This hazard unit should handle all the data and control hazards that can occur in your pipeline (listed in b) so that your pipelined processor computes correctly
- e) **[20 points]** Write small test programs, in MIPS assembly, that will show whether the pipelined processor is working or not. Each of your test programs should be designed to catch problems, if there are any, in the execution of MIPS instructions in your pipelined machine. Write:
 - A test program with no hazards (to verify that there are no problems with the connections in your pipeline etc.)
 - A test program that has one type of hazard, and another, and another...

In the end, have at least 4 test programs (testing at least 3 different hazards) with their machine code (in hex) and MIPS code (written in your report).

You can use the student-written assembler tool available online to help you quickly implement your test programs (<https://github.com/bilkentCraps/mips>). Remember that the goal of testing is to verify that all the instructions are fully working, and that all the instructions are working even in the presence of hazards.

Part 2. Lab Work (50 points)

For the lab work, you only need to submit resulting code file in the txt format.

- a) **[35 points]** You are given a skeleton SystemVerilog code for your pipelined MIPS processor in the file **PipelinedMIPSProcessorToFill.txt**. The places in the code that needs to be modified are shown with comment blocks above them. Fill them to implement the Pipelined Processor. You don't need to follow the skeleton code point by point. If you think your design is better, you are welcome to try it in your code, as long as your version of the code works, too. Note that in the skeleton code register file is initialized to 0 on reset. That effectively means all register values are set to 0 at the beginning.
- b) **[15 points]** Now make a SystemVerilog testbench file and using EDAPlayground, simulate your Pipelined Processor by executing the test programs you wrote at Part 1-e). However, during lab day we will provide you with our test programs and grade your work based on those. The top_mips module will be the top module of your design. That is, your testbench should instantiate that module. You should not remove existing input-output variables from that module as we will need at least those values for grading. Nevertheless, you are free to add more logic variables while testing your design. Present the results to your TA in the form of a waveform using waveform visualization functionality of EDA Playground. On the lab day, we might require you to add some logic variables to your top module so that we can see their values on the waveform.

Aim to complete lab part at home before coming to the online lab.

Part 6. Submit Your Code for MOSS Similarity Testing

1. Submit your Lab Work MIPS codes for similarity testing to Moodle.
2. You will upload one file. Use filename **Student-ID_FirstName_LastName_SecNo_LAB_LabNo.txt**
3. Only a NOTEPAD FILE (txt file) is accepted. No txt file upload means you get 0 from the lab. Please note that we have several students and efficiency is important.
4. *Even if you didn't finish, or didn't get the MIPS codes working, you must submit your code to the Moodle Assignment for similarity checking.*
5. Your codes will be compared against all the other codes in the class, by the MOSS program, to determine how similar it is (as an indication of plagiarism). So be sure that the code you submit is code that you actually wrote yourself!

Part 7. Lab Policies (Reminder)

1. As indicated in Lab1: Attendance is mandatory and the preliminary work is graded only if you submit your lab work with the observation/permission of your TA.
2. You can do the lab only in your section. Missing your section time and doing in another day is not allowed.

3. The questions asked by the TA will have an effect on your lab score.
 4. Lab score will be reduced to 0 if the code is not submitted for similarity testing, or if it is plagiarized. MOSS-testing will be done, to determine similarity rates. Trivial changes to code will not hide plagiarism from MOSS—the algorithm is quite sophisticated and powerful. Please also note that obviously you should not use any program available on the web, or in a book, etc. since MOSS will find it. The use of the ideas we discussed in the classroom is not a problem.
- Your lab attendance is tracked by the Zoom system. Please also see lab policies no. 1 item.