ECE4700J Computer Architecture

Summer 2022

Final Project

1. Logistics:

- This project is team-based.
- Please use the Piazza for Q&A.
- All reports and code (if available) MUST be submitted to the assignment of Canvas.
- Internet usage is allowed and encouraged. But copying code/materials from the internet is not allowed (you will get zero score and other penalties).
- No late submission is allowed for this project.
- Creativity is highly encouraged and credited for the project.
- Please read this document VERY carefully.

2. Project Description

This is a team-based project, you will work in a team of 5 or 6 students. Each team will pick one unique topic from the following pool, and the selection process will be announced separately.

2.1 Project Topic Pool

T1. Out of Order (OoO) Pipeline I

Implement a Scalar MIPS R10K Style (using map table and free list to implement register renaming) Out-of-Order pipeline for VeriSimpleV (The ISA we used in our lab4) using Verilog.

You only need to implement the pipeline, and you can use lab4 as the baseline. (You can just use the memory module we provide you in lab4). More details about this pipeline will be provided later.
Ref: 1. MIPS R10K Case Study (EECS470 WN21)

https://drive.google.com/file/d/1icZ03z12w_MNq6ft4AUwUrNQ6VO5Hf8b/view?usp=sharing

2. MIPS R10K: https://ieeexplore.ieee.org/document/491460

T2. Out of Order (OoO) Pipeline II

Implement a Scalar Intel P6 Style (using Tomasulo Algorithm + ROB) Out-of-Order pipeline for VeriSimpleV (The ISA we used in our lab4) using Verilog.

You only need to implement the pipeline, and you can use lab4 as the baseline. (You can just use the memory module we provide you in lab4). More details about this pipeline will be provided later.

Ref: 1. Tomasulo Algorithm in Textbook (*Computer Architecture: A Quantitive Approach*) Ch.3, 3.4-3.6

1. Intel P6 Microarchitecture Case Study (EECS470 WN21)
https://drive.google.com/file/d/1cVtQsRJsBT2aktJbcKJlaLLPD77rxMYF/view?usp=sharing

2. P6 Microarch: https://en.wikipedia.org/wiki/P6_(microarchitecture)

T3. SuperScalar Pipeline

Implement a Superscalar (3-way) 5-stage Pipeline for VeriSimpleV.

To increase performance, convert your pipeline into a 3-way super-scalar pipeline. Take considerations of all the bypassing/forwarding paths. Because super-scalar implementations increase the instruction bandwidth, this project will entail changes through the entire memory system.

T4. Deep Pipeline

Deep pipelining breaks the pipeline up into more than five stages, decreasing the cycle time of your processor. Lengthening the pipeline, however, will increase the difficulties that hazards cause in the system. Some components, such as the ALU, will have to be broken up into two stages (you will have to implement it using individual gates). To get full-credit for this part, you need to implement *at least* 8 stages: 2 IF, 1 DEC, 2 EXE, 2 MEM, and 1WB.

T5 Adv. Cache I

Implement a Non-blocking Write-Allocate Write-back L1 DCache with MSHRs (Miss status handling registers) using Verilog. You can use the memory module we provide you in lab4 (turn on the CACHE_MODE) as the DRAM memory you need to cache. You also need to implement a well-designed testbench for your cache and use benchmarks to profile it. More details will be provided later.

Ref. 1. Non-blocking Cache in Textbook (*Computer Architecture: A Quantitive Approach*) Ch.2, 2.3 "Fourth Optimization: Nonblocking Caches to Increase Cache Bandwidth"

T6. Adv. Cache II

Implement a Blocking L1 DCache with Data Prefetching and Victim Cache (victim buffer) using Verilog. You can use the memory module we provide you in lab4 (turn on the CACHE_MODE) as the DRAM memory you need to cache. You also need to implement a well-designed testbench for your cache and use benchmarks to profile it. More details will be provided later.

Ref. 1. Prefetching in Textbook (*Computer Architecture: A Quantitive Approach*) Ch2, 2.3 "Eighth Optimization: Hardware Prefetching of Instructions and Data to Reduce Miss Penalty or Miss Rate"

2. Victim Cache: https://ieeexplore.ieee.org/document/134547

T7. Adv. Cache III

Use small FIFO of 4 words to buffer out-going writes for either write-through or write-back caches. For write-through caches, a write-buffer allows several back-to-back writes to be queued without stalling the processor. For write-back caches, a write buffer allows a cache line to be temporarily stored in the write buffer while the replacement line is read in (allowing execution to continue). Note also, that stores should no longer block the pipeline, even if they miss in the cache.

T8. Gem 5 I

Implement an advanced cache system for the OoO core (O3CPU, https://www.gem5.org/documentation/general_docs/cpu_models/O3CPU) that is much different from the default cache (https://www.gem5.org/documentation/general_docs/memory_system/classic_caches/) in Gem5. Profile it using a benchmark.

T9. Gem 5 II

Implement an advanced branch predictor for the OoO core (O3CPU, https://www.gem5.org/documentation/general_docs/cpu_models/O3CPU) that is much different from the existing ones (e.g. TAGE()) in Gem5. Profile it using a benchmark. Branch prediction has been crucial to out-of-order processors' success, and was an area of significant improvement for several decades. There are already some advanced predictors available in gem5; check to see what's there first.

T10. Gem 5 III

Suppose you are designing the L1D and L2 data caches of a new processor, based on gem5's out-of-order O3CPU. For simplicity, you can choose from only three replacement policies for the caches: Random, LRU, NMRU. NMRU policy replaces a random block which is not the most recently used. Run

the simulations of your policies on workloads, and assume you have a 64KB L1D, and a 2MB L2. Assume both L1D and L2 policies have to be the same, and that you are using 8-way set associativity on both.

Other parameters: 64KB L1I + LRU, 2GHz frequency, 8-issue OOO core.

As for the second aspect, the hardware implementation team is always worried about hitting high frequencies with a complex policy. As the associativity increases, the hardware cost might also change. Thus, the implementation team suggests the following maximum associativity. Use the same capacity for all caches designs. Evaluate again how this can change your decisions.

	Random	NMRU	LRU
Associtivity	4-8	4-8	<4

In your report, describe the configurations you simulated, the results of your simulations, and your overall conclusion of how to architect the caches. Please include pretty looking graphs that help explain your conclusions. Please also do your best to explain why a certain policy is working better on various workloads (feel free to use microbenchmarks if you like). Any insights on whether/when more associativity or a better replacement policy matters more?

If time allows, you can choose to implement another policy LIP. LIP is an insertion policy which places incoming blocks at the LRU position instead of the MRU position. These slides from Moinuddin Qureshi do a good job explaining the LIP algorithm. You are encouraged to refer to the paper as well if needed.

References:

- https://polyarch.github.io/cs251a/hws/nmru-tut/
- http://www.jaleels.org/ajaleel/talks/DIP_ISCA2007.ppt

T11. Adv. Branch Predictor

Build an advanced instruction fetching system for your lab4 pipeline. This should include implementing the ICache and instruction prefetching, and implementing an advanced BranchPredictor. You can use the memory module we provide you in lab4 (turn on the CACHE_MODE) as the DRAM memory you need to cache. For the branch predictor, you must at least implement the BHT (Branch History Table) and the BTB (Branch Target Buffer). More details will be provided later.

Ref: 1. Prefetching in Textbook (*Computer Architecture: A Quantitative Approach*) Ch2, 2.3 "Eighth Optimization: Hardware Prefetching of Instructions and Data to Reduce Miss Penalty or Miss Rate"

T12. Open-ended Project I

Propose a research idea that falls in the scope of computer architecture and evaluate it using any means you like. You are free to combine it with ongoing research from your own studies, or with another course, provided the scope of the project implementation submitted for this course is sufficient.

T13. Open-ended Project II

Propose a plan to reproduce (part of) the main design of existing research work. You can reproduce any design (after approval) in the last 15 years of Top conferences (ISCA, MICRO, HPCA, ASPLOS) publication. You can choose any proper simulator/framework/language you like to reproduce it (e.g. Gem5, Chisel, Verilog, ...). You need to decide on how to implement the design and how to test/evaluate it.

Related proceedings

- International Symposium on Computer Architecture (ISCA)*
- International Symposium on High-Performance Computer Architecture (HPCA)*
- International Symposium on Microarchitecture (MICRO)*
- Architectural Support for Programming Languages and Operating Systems (ASPLOS)*
- International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)
- International Conference on Mobile Computing and Networking (MobiCom)
- Annual ACM IEEE Design Automation Conference (DAC)
- International Symposium on Low Power Electronics and Design (ISLPED)

3. Deliverables

- MDE Group **or** Mixed Group (MDE + non-MDE) Deliverables:

  a. Each team leader should submit your team's graduation thesis (together with Individual Contribution Report) to your course TA after you complete the thesis. **The templates and detailed instructions are available on canvas.**

  - Note that all senior students who complete their graduation thesis in the form of group work should submit an Individual Contribution Report (毕业论文单独工作报告) before graduation. The team leader needs to compile all these reports and attach them to your thesis.

  - You have to prepare slides for your oral defense. The 20-min final oral defense is scheduled to be held on Aug. 3rd. The exact time will be announced.

- Package your slides, implementation code, test benches and other necessary files into one zip file. If you use any tcl scripts, please also include those.
- Please follow the UEO announcement and checklist.
- Please follow the SJTU guidelines and templates for submitting your final thesis.

- Non-MDE Group Deliverables:
  - A well-presented project report that has the problem definition, your implementation details, any novelty, your evaluation methodology, detailed simulation results and evaluation results, and future work. No format requirement, but IEEE proceeding format is highly preferred. Please check the general guidelines below.
  - Each member needs to submit an Individual Contribution Report (uploaded on canvas). The team leader needs to compile all these reports and attach them to your final report.
  - Project presentation – A 20-min project presentation (15+5) will be held together with the MDE oral defense (tentatively).
  - Package your slides, implementation code, test benches and other necessary files into one zip file. If you use any tcl scripts, please also include those.

General Report Guidelines

- Abstract: Summarize the main contribution(s), and list any key findings.
- Introduction: Describe the problem your work addresses, why it is important, and overview the solution (if you are proposing one).
- Related Work: Overview the relationship of your work to prior work, with citations. It's of course okay if the work is not 100% novel.
- Methods: Detail the proposed design or methods (this may span one or more sections). This includes the design of any algorithms, hardware structures, interface strategies, codesign techniques, etc. Be sure to use drawings/figures/code-examples that can help explain the ideas.
- Methodology: Describe your approach for how you evaluated the work, and explain why it is fair or valid.
- Evaluation: Provide and analyze any quantitative results.
- Conclusions: Summarize the findings and main contributions, as well as any ideas for future work.
- Statement of Work: For each student in the group, describe the tasks that they performed.

- References: List all references cited.

General Presentation Guidelines

In this short presentation, you should motivate your audience (start with some background to connect with what we have learned), highlight the contributions of your project. You can even choose to interact with your audiences and prepare questions for discussions. There are no limitations on template to use or how many slides you need to have. The rule of thumb is usually one minute per slide. The following are several links and tips you might find helpful.

1. Top 5 Tips to a Successful Conference Presentation https://mitendicotthouse.org/top-5-tips-to-a-successful-conference-presentation/

2. How to give a technical presentation (how to give a scientific talk) https://homes.cs.washington.edu/~mernst/advice/giving-talk.html

3. Guidelines for Oral Presentations https://ocw.mit.edu/courses/biological-engineering/20-109-laboratory-fundamentals-in-biological-engineering-spring-2010/assignments/guidelines-for-oral-presentations/

A few tips for a good presentation:

- Well-defined scope
- Well balanced text vs. pictures
- Pick up only one aspect, clearly describe the novelty of the work
- Always think from your audiences' perspectives, in this case, your audiences will be your peers
- Proper references are required
- Good time management
- Being confident
- Eye contact is important

4. Important Dates

This is a 5-week project assignment; the intent is to allow you to plan and execute a significant, **open-ended** design exploration and mapping. You will not achieve the implementation goal or the course learning goals by trying to do this in one week. We give you timeline to help provide some structure, but the milestones are minimal and doing the minimum to hit the milestone each week will be insufficient to get you where you need to be at the end. We are giving you flexibility in planning and ordering rather than lock-step specifying exactly what you need to do each week. Also, please note that a team project is a great experience for exercising your ability for collaborating with peers. Thus, everyone needs to contribute to make the project a success. Please watch out for the following important steps.

Week	Item	Due date	
8	Form the team and select the team leader	June 28th 23:59 (Beijing Time)	ECE4700 Team Assignment
8	Rank the Project Topic Preferences	June 29th 23:59 (Beijing Time)	https://wenjuan.feishu.cn/m?t=skDzQqa2HDDi-2kc1
8	Announce the project topic selection results	June 30th 23:59 (Beijing Time)	
8	Kickoff Meeting Slot Registraion	June 30th 23:59 (Beijing Time)	ECE4700 Team Assignment
9	Kickoff Meetings	July 2nd 8:00pm - 12:00am (Beijing Time)	
10	Literature Review Slides Submission	July 13th 23:59 (Beijing Time)	Template on Canvas
11	Milestone Check	July 20th - July 22th	
13	Submit Thesis/Report to TA/Instructor (for initial check)	Aug 2nd 23:59 (Beijing Time)	Templates on Canvas
13	Final Presentation	Aug 3rd (Time TBD)	15min + 5 min (Q&A)
13	Submit source code & final slides	Aug 6th 23:59 (Beijing Time)	
13	(For MDE or Mixed group only) Thesis Submission to SJTU library	Aug 7th 23:59 (Beijing Time)	Instructions provided by UEO

13	(For non-MDE group only) Final report submitted	Aug 7th 23:59 (Beijing Time)	
14	Grade due date	Aug 9th	

- By **June 28th 23:59 (Beijing Time)**, form your team and select the team leader in this table ECE4700 Team Assignment
  - The group assignment is based on your own preferences but with the following guidelines.
    - We have students who are taking this course as MDE but some are not. This will lead to slight different requirements in terms of deliverables (in Section 3). To make the grading and judgement process easier, it is highly recommended to select your team members based on enrollment purposes (e.g. All MDE, All non-MDE). Please try to avoid the mixed group as much as possible.
    - We are planning to have **7** groups in total, where 4 groups will have 5 students each and 3 groups will have 6 students each. The assignment is based on your own choices. The team assembling process is shared ECE4700 Team Assignment . **For those who are not able to find a team, please let the instructor/TA know so you can be assigned randomly.**
    - A team leader is **required** and will be responsible for delivering the reports, dealing with team logistics and so on. **Leadership will be credited as extra bonus within a certain range based on the team performance**.
- By **June 29th 23:59 (Beijing Time)**, the team leader needs to indicate your intention for the project topic pool here. Eventually, each team will work on a unique topic. To make the selection, each team needs to rank the top 5 preferences. We will try to accomodate based on your preferences, but please note that top 1 preferences might not be ganranteed. Please also note that all projects are designed equally in terms of amout of work and efforts to spent, so if you are not able to select your top choice, it is totally ok. **To reduce the interrupt to the actual project execution, you won't be able to change your topic after the selection deadline unless being approved by the instructor/TA.**
- By **June 30th 23:59 (Beijing Time)**, each team needs to select a time slot for discussing more details about the project.
- By **June 30th 23:59 (Beijing Time)**, the teaching group will announce the project topic selection results. No more changes will be allowed unless approved by the instructor/TA.
- The kickoff meetings will be held on **July 2nd**. Each group will meet with the teaching group and discuss the details about the project including the technical challenges, deliverables, and any other questions or concerns.

- As part of a project like this, literature search and review is very important. The goal is to develop skills of searching literatures (within the scope of the selected project topic) using the library resources, identify the key information in literature and be able to interpret all findings into a short presentation. Critical thinking and paper critiques are highly encouraged. Here are useful resources for accessing campus library databases for literature search. Our class is in the field of computer science and engineering, so the most commonly used databases are IEEE Xplore and ACM Digital Library. But you are free to use Web of Science (which give you more results) too. As the delivery, each team needs to submit one set of slides that summarize the main contribution, techniques, results and how it can help with your project. A template has been uploaded on canvas. The slides will be due by **July 13th 23:59 (Beijing Time)**.
- To make sure that the project execution is on track and effective, milestone check sessions will be held between **July 20th** and **July 22th**. Each team leader (or representative) will need to briefly report your current progress and difficulties. Failure to do so will result in your project being incomplete.
- By **Aug 2nd 23:59 (Beijing Time)**, an initial draft of your thesis/report needs to be submitted to the TA/instructor on Canvas. Note that in this draft, you don't need to have everything ready, but please make sure you mark anything missing. High level comments and feedback will be provided.
- On August 3rd, your group **must** do a final presentation. Your presentation should be **15 minutes long, 20-minutes in total including questions.** Everybody in your group must present; your individual grade will include your presentation. Since you will present before your final report is due, you will not be expected to present a completed, fully functional, project. Good presentations (and write-ups, for that matter) will cover the specific sub-projects you chose to implement, and how they affected your processors performance. Detailed descriptions of your project datapath are not appropriate for a 15-minute presentation. However, high-level data paths might be appropriate. The presentation will be graded by the teaching group based on the quality of the results and the ability to answer all the questions (including clarification questions).
- Your source code and final version of the slides are expected to be submitted by **Aug 6th 23:59 (Beijing Time).**
- Your final version of the document (thesis or report) will be due by **Aug 7th 23:59 (Beijing Time)**.
- Grades will be released by Aug 9th.

5. Grading Policy

Factors	Percentage
Literature Review	5% towards your final course grade (will be graded separately)

Functionality Correctness & Results	40%
Defense Presentation (including presentation + Q&A)	25%
Report/Thesis	20%
Source Code	15%

- Late submission will result in 0 point for the corresponding part and deduction of 10% per day in the other part until all 100% is deducted. It can also potentially delay your graduation.

Bonus:

- The team leader will get **an extra bonus within a certain range (10% -15%) based on the team performance**.
- For creative implementation ideas, the whole team will get extra bonus (from 10% - 15%) towards the final lab grade.
- For excellent implementations, your work will be proposed to publish your results at top conferences/journals. The instructor/TA will help throughout the process.

6. Recommendations

- Pick the application based your interests, availability and your bandwidth. You will need to take the ownership of your topic once you make the selection.
- Start as soon as possible, and plan carefully.
- All team members need to contribute. This is very important to make the project a success.
- Gem5 Tutorial http://learning.gem5.org/tutorial/
- The deadline tracking board for this course is also available here. ECE4700 Deadline Tracking Board