ECE621 Computer Organization

Faculty
Professor Ziqiang Patrick Huang (ziqiang.huang@uwaterloo.ca)
Office hours: TBD

Lectures
[redacted]

Teaching Assistant
Shubham Ranjan (shubham.ranjan@uwaterloo.ca)
Office hours: TBD

Webpage
https://uwaterloo.ca/scholar/z399huan/ece621-computer-organization
We will use LEARN for lecture notes, assignments, projects and grades

Synopsis
This is a graduate course on computer architecture focusing on quantitative methods for cost and performance design tradeoffs. This course covers the fundamentals of classical and modern general processor design. This includes organization, performance, instruction-sets, pipelining, caches, virtual memory, I/O, superscalar, out-of-order execution, speculative execution, multithreaded processors, multiprocessors, cache coherency, memory consistency and synchronization techniques; and special-purpose architectures.

Prerequisites
No formal course requirements are necessary; however, students are expected to be familiar with the basics of instruction-set architectures, assembly language programming, pipelines and caches.

Textbook
Hennessy and Patterson. “Computer Architecture: A Quantitative Approach” 6th Edition, 2018.

Grading
Reading responses: 10%
Project: 30%
Midterm: 20%
Final: 40%

Note: Late submissions incur a 20% penaltywhen < 24 hours late, incur a 50% penalty when 24-48 hours late, and receive no credit when > 48 hours late.

Reading Responses

Students will prepare an insightful critique of the assigned papers due at the beginning of class. These responses should take the form of a constructive paper review, including (1) summary, (2) strengths,(3) weaknesses, and (4) future directions. These responses should be no longer than one page per paper. Responses should be done individually and will be evaluated pass/not pass.

Project

Students can work in groups of maximum two people.The group will design and implement a cycle- accurate MIPS processor architecture. Further details on the specifics of the architecture and compiler suite used will be provided in class.The group will demonstrate their implementation. Further details on the demonstration will be provided later, but expect there to be intermediate demonstration requirements.

Academic Policy

The discussion of ideas and problem solving strategies is an integral part of the learning experience, but cheating and plagiarism is not. Practically, you violate academic integrity when (1) you obtain solutions and code from others, or (2) you provide solutions and code to others. A student is expected to know what constitutes academic integrity to avoid committing an academic offence, and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about “rules” for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean. For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline. For typical penalties check Guidelines for the Assessment of Penalties.