General Information

Instructor: Prof. Mark Jeffrey
Email: mcj@ece.utoronto.ca
Office: Pratt 484C Microsoft Teams
Office Hours: Tuesdays 3:00pm - 4:00pm, or by appointment

Teaching Assistants: Isidor Brkic, Alberto Delmas Lascorz, Viktor Karyofyllis, Gil Posluns, and Daniel Rozhko

Calendar Description


Overview

Prerequisite: ECE 243 or ECE 352

Prerequisites by topic: Computer Organization, C programming, Digital Logic

Lecture

Tuesday 3:00pm - 4:00pm Replaced with async video lecturettes on Quercus
Wednesday 12:00pm - 1:00pm Replaced with async video lecturettes on Quercus
Thursday 3:00pm - 4:00pm Synchronous Q&A on Blackboard Collaborate

Tutorials

TUT2 Monday 9:00am - 10:00am Synchronous on BB Collaborate
TUT1 Tuesday 10:00am - 11:00am Synchronous on BB Collaborate

Labs (approximately every other week)

PRA1 and PRA2: Thursday 9:00am - 12:00pm (Starting Sept 17)
Within this time window, set up a 30-minute Microsoft Teams appointment

Midterm: date and details TBD. Likely using CrowdMark or Quercus Assignment

A digital copy is also available at [http://go.utlib.ca/cat/12598461](http://go.utlib.ca/cat/12598461)

Technology: All course material—including lab handouts, videos, and lecture notes—will be posted or linked to on Quercus ([quatUTORONTO.CA](http://g.utoronto.ca)). We will use Blackboard Collaborate for synchronous sessions, Microsoft Teams for small appointments, and Piazza for text-based Q&A. Piazza will be monitored by
the instructor and TAs to answer student questions related to course material and assignments. Links to BB Collaborate and Piazza can be found as links from the Quercus page.

Questions: All questions (that are not of a personal nature, e.g., grade inquiries, missed coursework due to illness, etc) should be posted to Piazza. All questions that you email to the instructor or the TA will receive the response: "Please repost to Piazza", where both the question and the answer will reach its full audience. It is in everyone's interest that we maintain this policy; this is absolutely the most effective way to communicate.

Lecture Notes: Skeleton lecture notes will be posted on Quercus along with links to lecture videos. You are encouraged to further annotate the notes as you watch, and during Q&A sessions.

Acknowledgement: Lecture notes, labs, and syllabus are adapted from Prof. Natalie Enright Jerger.

**Course Description**

This course provides students with a solid understanding of fundamental architectural techniques used to design today's high-performance processors and systems.

Computer architecture is the science and art of selecting and interconnecting hardware components to create a computer that meets functional, performance and cost goals. This course qualitatively and quantitatively examines computer design tradeoffs. We will learn, for example, how uniprocessors execute many instructions concurrently and why state-of-the-art memory systems are nearly as complex as processors. We will also learn how VLSI technology has evolved and influenced processor design. At the end of this course you will be able to appreciate the technical rationale behind the clock-speed race in the 90s, reason for its demise, and the reasons why industry has moved to multi-core chips. Course topics include pipelining, superscalar, out-of-order execution, multithreading, caches, and multiprocessors. Some emphasis will be placed on hardware/software interaction to achieve good performance. Issues affecting the nexus of architecture, compilers and operating systems will be briefly touched upon.

Nearly all engineers, regardless of sub-specialty, utilize programming in the course of their job; to write high performance computer programs, it is necessary to understand the underlying hardware that those programs will run on. Two-thirds of this course will focus on high performance architectures and memory systems. The final third of the course will focus on multiprocessors. Significant challenges face today's computer hardware industry. At the forefront of these challenges is the multi-core revolution. The transition from single processor designs to multi-core design requires hardware and software designers knowledgeable about a range of issues in parallel computing including hardware.

**Course Objectives**

 Students who successfully fulfill the course requirements will have demonstrated:

- Competence in specialized engineering knowledge:
  - Ability to understand the design of a pipelined CPU and cache hierarchy
  - Ability to analyze and evaluate CPU and memory hierarchy performance
  - Understanding of trade-offs in modern CPU design including issues affecting superscalar and dynamically scheduled architectures
  - Understanding of hardware design of multiprocessors including cache coherence and synchronization
• Ability to select candidate engineering design solutions for further evaluation: an understanding of trade-offs between performance, power and area in microarchitectural components
• Demonstrate the ability to execute solution process to engineering problems: design processor components through laboratory exercises.
• Use of Engineering Tools: experience with complex simulation tools to study various microarchitectural features. Recognize the limitations of these tools.

Course Evaluation

Labs (5) 30%
Midterm (1) 30%
Final Exam 40%

Labs: There will be 5 lab assignments. Lab materials will be posted on Quercus at least 1 week before the lab date. Each lab will require modification to an architectural simulator to reinforce concepts discussed in class. These labs require knowledge of C/C++ programming. These labs are to be completed in groups of two. The workload for these labs is significant. Note: you do not need to be online for the full 3-hour lab, but you must schedule a recurring 30-minute appointment to be marked by your TA. For more details on lab policies and guidelines please see the Lab Policy handout in the course material.

Midterm: date and details TBD, likely through CrowdMark or Quercus Assignment

Calculators: Faculty-approved non-programmable

Exam Aids: Type X - Open book

Tutorials: Weekly tutorials supervised by TAs will reinforce course concepts through problems. A list of problems to be discussed will be posted on Quercus at least one week before the tutorial. Tutorials will also cover information regarding the labs.

Should you wish to go more in depth in the course material, I will periodically post research papers of historical significant or related to modern innovations. Students can break out in to groups in tutorial to discuss the reading and relate it to the lecture material. These readings are optional.

Learning amid a global pandemic:

As Professor Brandon Bayne of UNC has said: “Nobody signed up for this. Not for the sickness, not for the social distancing, not for the sudden end of our collective lives together on campus. Not for an online class, not for teaching remotely, not for learning from home, not for mastering new technologies, not for varied access to learning materials.”

We have adapted this course to support students during these difficult times. To enable more flexibility in your schedule, in-class lectures will be replaced with asynchronous recorded video lecturettes released weekly each Tuesday, with one weekly (recorded) synchronous Q&A session on Thursdays via Blackboard Collaborate (BBC). Weekly tutorials will remain synchronous (on BBC), to enable interactive questions with TAs. These will also be recorded for later viewing.

The instructor welcomes your constructive feedback as we navigate this semester together, e.g., on material, tools, bandwidth demands, and more. You can email Prof. Jeffrey directly or submit feedback anonymously through this form.
To get the most out of this class:

Watch lecture video content and attend lecture Q&As. Attending lecture Q&A has several benefits. First and foremost it gives you an opportunity to interact with the instructor and ask questions and provides additional material not included in the course textbook. Attending synchronous sessions also allows you to network with your peer group. After graduation, classmates can be valuable contacts for job opportunities.

Start labs early: the labs involve the modification of fairly complex software. You will be exposed a wide range of tools used in computer architecture evaluation. These labs are not designed to test your programming abilities but will still take substantial time. You want to focus on the labs concepts as they will help prepare you for exams. Both you and your partner are expected to contribute to each lab.

Attempt the practice problems in advance of tutorial. This will help you identify key areas of concern for discussion with the TA.

Take advantage of office hours. The time to figure out that you do not understand a course concept is not the day after the exam. Use office hours to clarify concepts that are unclear. Interacting with students during office hours also gives me a better idea of the how well lecture is being paced and if certain concepts need to be revisited. An additional benefit is that I will get to know you better; this can come in handy when looking for letters of recommendation or references for job applications and graduate school. Be sure to introduce yourself the first time we meet.

Notice of Video Recording and Sharing (download and re-use prohibited)

This course, including your participation, will be recorded on video and will be available to students in the course for viewing remotely and after each session. Course videos and materials belong to your instructor, the University, and/or other sources depending on the specific facts of each situation, and are protected by copyright. Do not download, copy, or share any course or student materials or videos without the explicit permission of the instructor. For questions about recording and use of videos in which you appear please contact your instructor.

Academic Integrity Policies

http://www.academicintegrity.utoronto.ca/

This course will use turnitin.com for assessments. We include the following required statement:

Normally, students will be required to submit their course assignments to Turnitin.com for a review of textual similarity and detection of possible plagiarism. In doing so, students will allow their material to be included as source documents in the Turnitin.com reference database, where they will be used solely for the purpose of detecting plagiarism. The terms that apply to the University’s use of the Turnitin.com service are described on the Turnitin.com web site.

Land Acknowledgement:

We wish to acknowledge this land on which the University of Toronto operates. For thousands of years it has been the traditional land of the Huron-Wendat, the Seneca, and most recently, the Mississaugas of the Credit River. Today, this meeting place is still the home to many Indigenous people from across Turtle Island and we are grateful to have the opportunity to work on this land.
Inclusivity Statement:

You belong here. All students, staff, and faculty at the University of Toronto have a right to learn, work, and create in a welcoming, respectful, inclusive, and safe environment. In this class we are all responsible for our language, actions, and interactions. Discriminatory speech or actions of any kind will not be permitted, and do not align with the values of our Faculty. As a class we will support each other’s learning by creating an inclusive learning environment, one which is based on mutual respect for the dignity and worth of every person.

If you experience or witness any form of discrimination, please reach out to the:

• Engineering Equity Diversity & Inclusion Action Group: https://www.engineering.utoronto.ca/about/equity-diversity-and-inclusion/,
• An academic advisor: https://undergrad.engineering.utoronto.ca/advising-and-wellness/academic-advising-2/upper-year-advising/,
• A U of T Equity Office: equity.hrandequity.utoronto.ca/ or
• Any FASE faculty or staff member that you feel comfortable approaching.

Mental Health and Wellness

As a university student, you may experience a range of health and/or mental health issues that may result in significant barriers to achieving your personal and academic goals. The University of Toronto offers a wide range of free and confidential services and programs that may be able to assist you. We encourage you to seek out these resources early and often.

If, at some point during the year, you find yourself feeling distressed and in need of more immediate support, visit the Feeling Distressed Webpage: http://www.studentlife.utoronto.ca/feeling-distressed, for more campus resources.

Immediate help is available 24/7 through Good2Talk, a post-secondary student helpline at 1-866-925-5454.

As a U of T Engineering student, you have an Academic Advisor (undergraduate students) or a Graduate Administrator (graduate students) who can support you by advising on personal matters that impact your academics. Other resources that you may find helpful are listed on the U of T Engineering Mental Health & Wellness webpage, and a small selection are also included here:

• Accessibility Services & the On-Location Advisor
• Graduate Engineering Council of Students’ Mental Wellness Commission
• Health & Wellness and the On-Location Health & Wellness Engineering Counsellor
• Inclusion & Transition Advisor
• U of T Engineering Learning Strategist and Academic Success
• Registrar’s Office
• SKULE Mental Wellness
• Scholarships & Financial Aid Office & Advisor

Statement on Accommodations:

The University of Toronto supports accommodations for students with diverse learning needs, which may be associated with mental health conditions, learning disabilities, autism spectrum, ADHD, mobility impairments, functional/fine motor impairments, concussion or head injury, blindness and low vision,
chronic health conditions, addictions, deafness and hearing loss, communication disorders and/or temporary disabilities, such as fractures and severe sprains, or recovery from an operation.

If you have a learning need requiring an accommodation the University of Toronto recommends that students register as soon as possible with Accessibility Services at [https://studentlife.utoronto.ca/service/accessibility-services-registration-and-documentation-requirements/](https://studentlife.utoronto.ca/service/accessibility-services-registration-and-documentation-requirements/).
Phone: 416-978-8060
Email: accessibility.services@utoronto.ca

**Weekly Lecture and Lab Schedule**

<table>
<thead>
<tr>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Week 1</strong></td>
</tr>
<tr>
<td>Introduction, Technology Trends, Performance</td>
</tr>
<tr>
<td><strong>Week 2</strong></td>
</tr>
<tr>
<td>Pipelining</td>
</tr>
<tr>
<td><strong>Week 3</strong></td>
</tr>
<tr>
<td>Data Hazards</td>
</tr>
<tr>
<td><strong>Week 4</strong></td>
</tr>
<tr>
<td>Control Hazards and Branch Prediction</td>
</tr>
<tr>
<td><strong>Week 5</strong></td>
</tr>
<tr>
<td>Instruction Level Parallelism</td>
</tr>
<tr>
<td><strong>Week 6</strong></td>
</tr>
<tr>
<td>Dynamic Scheduling</td>
</tr>
<tr>
<td><strong>Week 7</strong></td>
</tr>
<tr>
<td>Caches</td>
</tr>
<tr>
<td><strong>Week 8</strong></td>
</tr>
<tr>
<td>Cache Optimizations, Memory</td>
</tr>
<tr>
<td>Midterm: TBD</td>
</tr>
<tr>
<td><strong>Engineering Fall Study Break</strong></td>
</tr>
<tr>
<td><strong>Week 9</strong></td>
</tr>
<tr>
<td>Multiprocessors and Cache Coherence</td>
</tr>
<tr>
<td><strong>Week 10</strong></td>
</tr>
<tr>
<td>Synchronization and Memory Consistency</td>
</tr>
<tr>
<td><strong>Week 11</strong></td>
</tr>
<tr>
<td>Superscalar</td>
</tr>
<tr>
<td><strong>Week 12</strong></td>
</tr>
<tr>
<td>Multithreading, Exam Review</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lab</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sept 17</td>
<td>Lab 0 — Introduction</td>
</tr>
<tr>
<td>Oct 1</td>
<td>Lab 1 — Hazards   Due Fri Oct 9</td>
</tr>
<tr>
<td>Oct 15</td>
<td>Lab 2 — Branch Prediction Due Fri Oct 23</td>
</tr>
<tr>
<td>Oct 29</td>
<td>Lab 3 — Out of Order Execution Due Fri Nov 6</td>
</tr>
<tr>
<td>Nov 19</td>
<td>Lab 4 — Cache Design Due Fri Nov 27</td>
</tr>
<tr>
<td>Dec 3</td>
<td>Lab 5 — Coherence Protocols Due Wed Dec 9</td>
</tr>
</tbody>
</table>