UNIVERSITY OF BRITISH COLUMBIA
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

EECE 527 – Advanced Computer Architecture, Spring 2014

Fact Sheet and Syllabus

Instructor:
Tor Aamodt (aamodt@ece.ubc.ca)
Office: Kaiser 4043

Lectures: Thursdays 13:00 to 16:00, MacMillan 158

The Course:
This is a demanding course that focuses on the numerous ways chip architects translate an increasing number transistors into exciting products. Coursework includes assignments, various paper readings, a lecture presentation and a project. By the end of the course, students will be able to explain the details and tradeoffs in the microarchitecture of modern microprocessors, be able to select appropriate mechanisms for tackling processor design challenges and evaluate their impact using architecture level simulation methodologies.

Course Web Page: We will be using connect.ubc.ca throughout the course.

Textbook:
Readings will be based upon research papers that will be made available on UBC Connect. The following textbook is optional: *Modern Processor Design: Fundamentals of Superscalar Processors,* John Paul Shen, Mikko H. Lipasti, McGraw Hill.

Assignments: There will be roughly 3-4 programming assignments to familiarize you with the typical design methodology used by architects in industry (as well as academia). You will need to know C++ and possibly some scripting languages such as python or perl to complete these assignments.

Paper readings and presentation:
To augment the lectures, there will roughly 3 paper readings per week assigned from the research literature. Based on these, you submit paper summaries during the term. We will set aside some time each week to discuss the papers in lectures.

Student Lectures: Towards the middle of the semester students you will work in groups of 2-3 to present an in depth lecture on a specific topic related to computer architecture. Topics will be assigned.

Project: You will deepen your understanding of computer architecture by completing a project. For more information on the project see page 3.

Grading:
20% Paper reviews (3-4 per week)
15% Simulation Assignments
5% Student Lecture
50% Project
10% Final Exam

Academic Integrity:
All work submitted in this course must be entirely your own or your group's. Concepts/ideas from other sources must be clearly cited. When in doubt, cite related work. Where text is quoted from other sources it must appear in quotation marks and with a clear reference to the original work. If it is necessary (and usually it is not) to reproduce a figure from another source the source of the figure must be made clear in the figure caption (cited). Suspected violations must be reported to the Dean's Office for further action. All UBC students are expected to be aware of UBC's policies on academic integrity. For more information related specifically to plagiarism in graduate courses
http://www.grad.ubc.ca/faculty-staff/policies-procedures/plagiarism-course-work-academic-responses

Lecture topics from last offering (this year might vary)

Superscalar Organization
Register and Memory Dataflow
Complexity Effective Microarchitecture
Accelerator Architectures
Energy Efficient Architecture
Multicore Cache Coherence Protocols
Transactional Memory
DRAM and Memory Controllers
Interconnection Networks

Project

You are encouraged to try to propose a novel direction that might lead to publishable research. You may select a topic that fits within your graduate research provided it is related to computer architecture in some way. If you are at a loss for a topic, you may pick a prior published work and reproduce the results in that work. You can start working on projects right away but to ensure you have enough time you should not start any later than half way through the term.

The following rubric is a guideline and not a strict marking guideline but should give you an idea of what is expected.

Mark	**<u>Very Rough</u> Rubric for Project Marking** **Project reproduces prior work** <------------------------>	**Project is entirely novel**
100%	Good reproduction of existing work and significant additional insight beyond that in the original paper provided. **<u>OR</u>** Convincing (quantitative) refutation of existing work and strong quantitative based insight into why the idea does not work as originally proposed combined with sensitivity analysis to ensure the validity of this conclusion.	Positive outcome (idea "works") with in-depth analysis showing WHY the idea works and how sensitive your solution is to any important configuration parameters (you need to determine which parameters are important). **<u>OR</u>** Negative outcome (idea "doesn't work") for several variations of idea with in-depth analysis providing significant insight into the reasons for negative outcome and detailed sensitivity analysis and reasonable proposals for follow-up work.
85%	Reasonable reproduction of existing work with limited additional analysis. **<u>OR</u>** Plausible refutation of existing work with some corroborating quantitative data to back up this claim and sensitivity analysis.	Positive outcome for a novel idea (not published) with little analysis showing WHY idea works. **<u>OR</u>** Negative outcome with good analysis into reasons for negative outcome some ideas for improvements
70%	Simulator coded, some performance data but no correspondence with existing work and no quantitative insight into source of discrepancy.	Positive or negative outcome with limited data, no quantitative analysis to explain why idea works.
50%	Simulator coded, no performance data due to "bugs".	
0%	No final project report or source code submitted.	